**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное автономное образовательное  
 учреждение высшего образования   
«Южный федеральный университет»**

**Институт высоких технологий и пьезотехники**

**Кафедра прикладной информатики и инноватики**

**Направление подготовки:   
09.03.03 "Прикладная информатика"**

**Отчет по Лабораторной работе №4**

**«Логические инструкции на языке ассемблера»**

**По дисциплине «Основы функционирования вычислительной техники»**

**Выполнил:**

студент 2 курса 6 группы

Куракин Н. А.

\_\_\_\_\_\_\_\_\_\_\_\_

*подпись*

**Проверил:**

Преподаватель Толмачев С.А.

Ростов-на-Дону

2024

# **ЛАБОРАТОРНАЯ РАБОТА №4**

**Тема:** «Логические инструкции на языке ассемблера».

**Цель работы:** изучить применение логических инструкций в

программах на языке ассемблера.

**Ход работы:**

Ознакомился с теоретической частью и приступил к выполнению задания. Необходимо было разработать программу, реализующую **17 задание:  
Дан массив из 6 байт. Посчитать количество единиц во всех разрядах, кратных пяти: 5, 10, … , 45.**

Внёс в сегмент данных исходные данные:

Массив mass размером в 6 байт;

А также переменную ones размером в слово (для того, чтобы в Dump был 00 между переменной и стеком). В неё будет помещен итог всех проверок массива.

Далее сегмент кода:

Первым делом загрузил сегмент данных и выставил адрес смещения массива.

Затем выставил счетчик для цикла на 9 (количество проверок) и обнулил dx (для повторного запуска).

Для первой проверки выставил единицу в пятый бит.

Создал метку **bytes**:

Загружается байт, сравнивается с битом, кратным 5

Если результат 0, перейти на следующую метку **moveset**, иначе прибавить 1 к DI.

Метка **moveset** (перемещение счетчика и создание нового)**:**

сохранение счетчика общих проверок и создание нового счетчика (сдвиг на 5 битов)

Перейти на метку **shift** (сдвиг)

Метка (и цикл) **shift**:

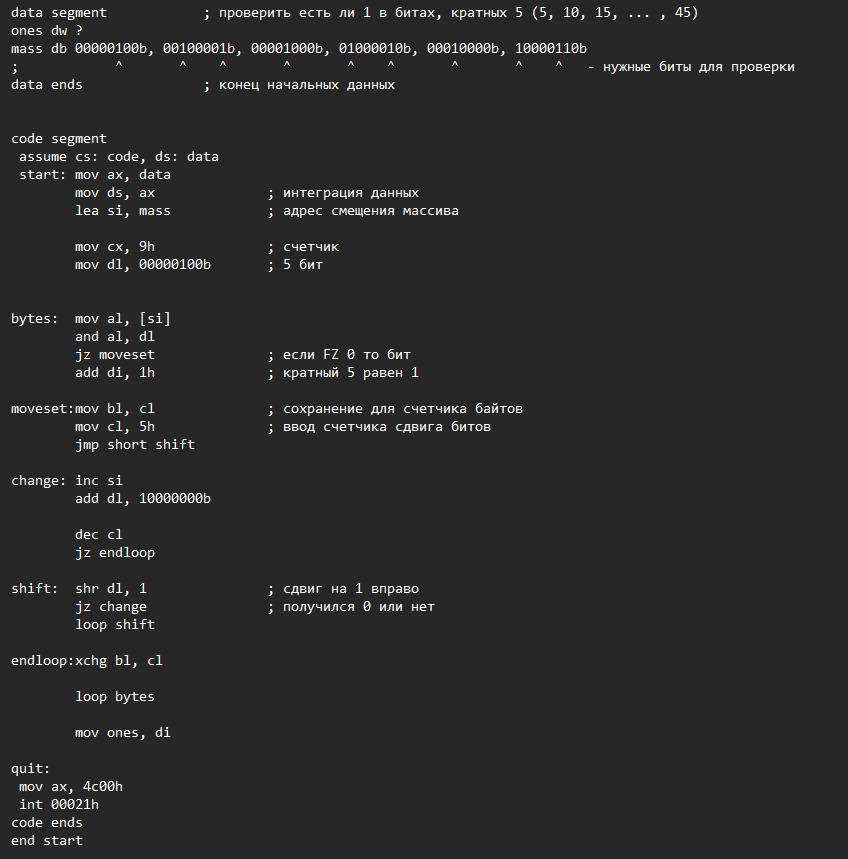
Сдвиг на 1 бит вправо, если получился 0, то перейти на метку **change** (смена байта). Иначе продолжить цикл

Метка **change:** добавляет к SI единицу, а затем в старший бит DL вставляет единицу. Отнимает от CL единицу (цикл **shift**), если получился 0, то перейти на метку **endloop**.

Метка **endloop:** меняет местами счетчики и делает цикл на **bytes**. Если CX=0, проходит дальше и переносит в переменную **ones** значение регистра DI.

Далее стандартный выход из программы

В результате полный код выглядит следующим образом (скрином):



Этот же код текстом:

data segment ; проверить есть ли 1 в битах, кратных 5 (5, 10, 15, ... , 45)

ones dw ?

mass db 00000100b, 00100001b, 00001000b, 01000010b, 00010000b, 10000110b

data ends ; конец начальных данных

code segment

assume cs: code, ds: data

start: mov ax, data

mov ds, ax ; интеграция данных

lea si, mass ; адрес смещения массива

mov cx, 9h ; счетчик

mov dl, 00000100b ; 5 бит

bytes: mov al, [si]

and al, dl

jz moveset ; если FZ 0 то бит

add di, 1h ; кратный 5 равен 1

moveset:mov bl, cl ; сохранение для счетчика байтов

mov cl, 5h ; ввод счетчика сдвига битов

jmp short shift

change: inc si

add dl, 10000000b

dec cl

jz endloop

shift: shr dl, 1 ; сдвиг на 1 вправо

jz change ; получился 0 или нет

loop shift

endloop:xchg bl, cl

loop bytes

mov ones, di

quit:

mov ax, 4c00h

int 00021h

code ends

end start

Выполнил проверку на нескольких вариантах, вот один из них (первое слово – итоговое значение значащих единиц; последующие байты – массив):

