ФГБОУ ВО “Чувашский государственный университет им. И. Н. Ульянова” Факультет: ИВТ

Кафедра: Вычислительной техники

Предмет: [Электронно-вычислительная машина](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE-%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0) и периферийные устройства

Лабораторная работа №5

Непривилегированные команды процессоров x86

Вариант №6

Выполнил: студент группы ИВТ-41-20

Галкин Дмитрий

Проверил: доцент Андреева А.А.

Чебоксары 2022 г.

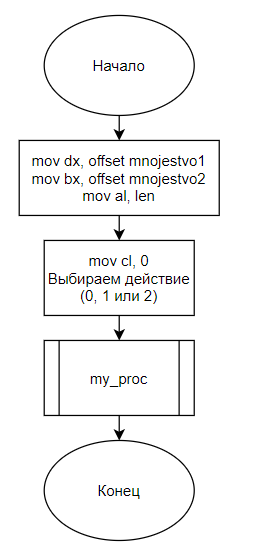
# Цель работы

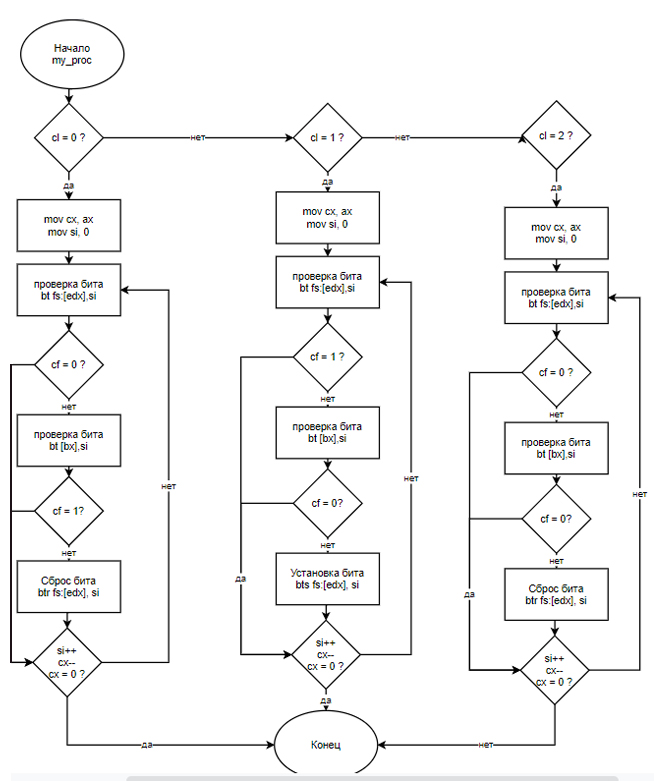
Познакомиться с особенностями реального режима микропроцессоров x86.

**Текст задания**

Даны два двоичных множества одинаковой длина, представленные в виде битовых строк. Написать процедуру, производящую в зависимости от значения регистра CL=0 – произведение, CL = 1- объединение, следующие действия: CL=2 – разность первого и второго множеств. Адрес первого множества FS:DX, второго – GS:BX, длина – AX. Результат записывается на место первого множества.

**Общая схема задания**

****

****

**Текст подпрограммы**

*; Даны два двоичных множества одинаковой длины,*

*; представленные в виде битовых строк. Написать процедуру,*

*; производящую в зависимости от значения регистра CL следующие действия*

*; CL = 0 - Произведение*

*; CL = 1 - Объединение*

*; CL = 2 - разность первого и второго множества*

*; Адрес первого множества - FS:DX*

*; Адрес второго множества - GS:BX*

*; Длина - AX*

*; Результат записывается на место первого множества*

*; Как ассемблер расстовляет префиксы 66 и 67 и как процессор их выполняет*

*.386*

*DATA segment para use16*

*multe\_1 db 1100b ; Множество 1 (DX)*

*multe\_2 db 0110b ; Множество 2 (BX)*

*DATA ends*

*CODE segment para use16*

*assume CS:CODE, DS:DATA*

*start:*

*mov ax, DATA ; Загружаем данные из DATA*

*mov ds, ax*

*mov fs, ax*

*mov edx, 0*

*mov ebx, 0*

*mov dx, offset multe\_1 ; Загружаем Множество 1*

*mov bx, offset multe\_2 ; Загружаем Множество 2*

*mov ax, 0*

*mov al, 10*

*mov cl, 0*

*cmp cl, 0*

*je multiplication*

*cmp cl, 1*

*je addition*

*cmp cl, 2*

*je difference*

*jmp exit*

*;------------------------------------------------------------> Умножение*

*multiplication:*

*mov cx, ax*

*mov si, 0*

*start1:*

*bt fs:[edx], si ; Проверка бита*

*jnc loop1 ; cf = 0*

*bt gs:[ebx], si ; Проверка бита*

*jc loop1 ; cf = 1*

*btr fs:[edx], si ; Проверка и сброс бита*

*loop1:*

*inc si*

*loop start1*

*jmp exit*

*;-----------------------------------------------------------> Объединение*

*addition:*

*mov cx, ax*

*mov si, 0*

*start2:*

*bt fs:[edx], si ; Проверка бита*

*jc loop2 ; cf = 1*

*bt gs:[ebx], si ; Проверка бита*

*jnc loop2 ; cf = 0*

*btr fs:[edx], si ; Проверка и сброс бита*

*loop2:*

*inc si*

*loop start2*

*jmp exit*

*;---------------------------------------------------------> Разность*

*difference:*

*mov cx, ax*

*mov si, 0*

*start3:*

*bt fs:[edx], si ; Проверка бита*

*jnc loop3 ; cf = 0*

*bt gs:[ebx], si ; Проверка бита*

*jnc loop3 ; cf = 1*

*btr fs:[edx], si ; Проверка и сброс бита*

*loop3:*

*inc si*

*loop start3*

*;-----------------------------------------------------------> Выход из процедуры*

*exit:*

*mov ah, 4ch*

*int 21h*

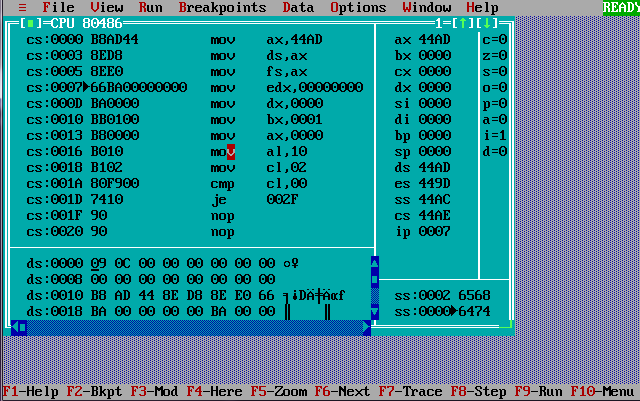
*;--------------------------------------------------------------------------*

*CODE ends*

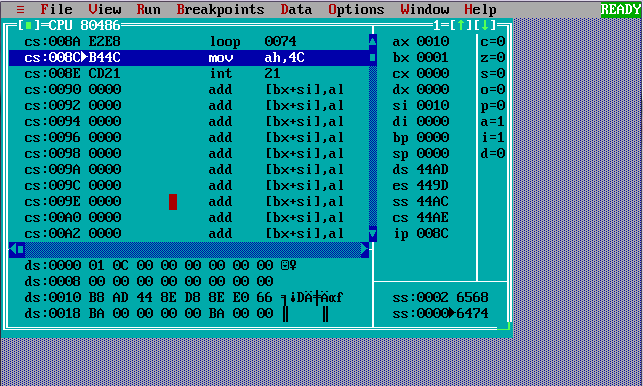
*end start*

**Пример работы программы**

Исходная строка



Результат работы программы



**Вывод**

Я ознакомился с особенностями реального режима микропроцессора x86.