# ЛАБОРАТОРНАЯ РАБОТА №3

# ИССЛЕДОВАНИЕ АРХИТЕКТУРЫ И СИСТЕМЫ КОМАНД 16-РАЗРЯДНОГО ПРОЦЕССОРА

Цель работы

Исследовать систему команд, архитектуру и основные блоки процессора Intel 8086 и взаимодействие этих блоков процессора при выполнении команд разных типов. Приобрести практические навыки написания ассемблерных программ и отладки их в эмуляторе микропроцессора экранным отладчиком типа emu8086.

Задачи

1. Изучить архитектуру МП 8086, состав регистров и работу процессора с использованием временных диаграмм;
2. Изучить основные директивы ассемблера и команды МП 8086;
3. Изучить функции BIOS и DOS и особенности использования их в ассемблерных программах;
4. Запустить эмулятор, выбрать шаблон в формате com и ввести в окне редактора пробную программу (Листинг 1);
5. Исследовать процесс выполнения программы (т.е. проследить изменение содержимого регистров, оперативной памяти и стека);
6. Пояснить смысл исследуемой программы в форме комментариев к каждой её ассемблерной строке;
7. Запустить эмулятор, выбрать шаблон в формате com и ввести в окне редактора пробную программу (Листинг 2);
8. Исследовать и пояснить изменения регистров при выполнении каждой из команд;
9. Рассчитать время выполнения программ.

Ход работы

Была рассмотрена программа №1 (Листинг 1). Она выполняет ряд арифметических и логических операций над значениями переменных и регистров общего назначения, затем с помощью специальных прерываний вывод надпись на экран (консоль). Она была запущена через эмулятор emu8086 в формате com (Рисунок 1), там же для неё были написаны комментарии.

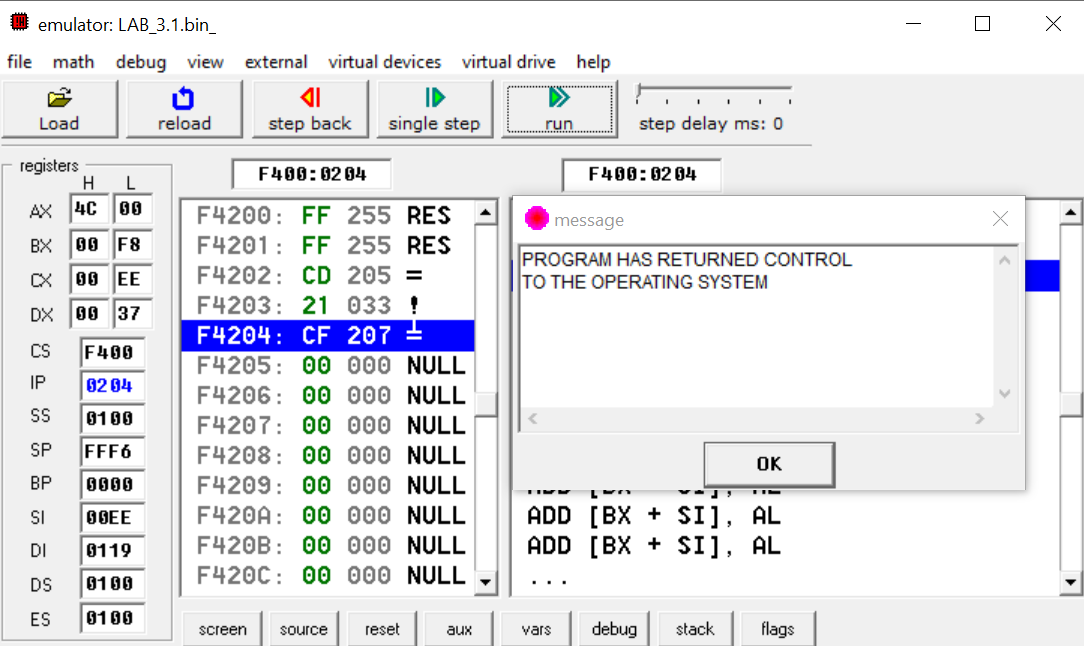


Рисунок 1 – Эмуляция программы 1

Похожие действия были выполнены для программы №2. Через специальный цикл она выводит на экран число 12345 (десятичная с.с.) в двоичной форме на экран, а затем ожидает ввода с клавиатуры, выводит строку и завершается. Программа 2 также была запущена в формате com, данные были выведены в консоль (Рисунок 2).

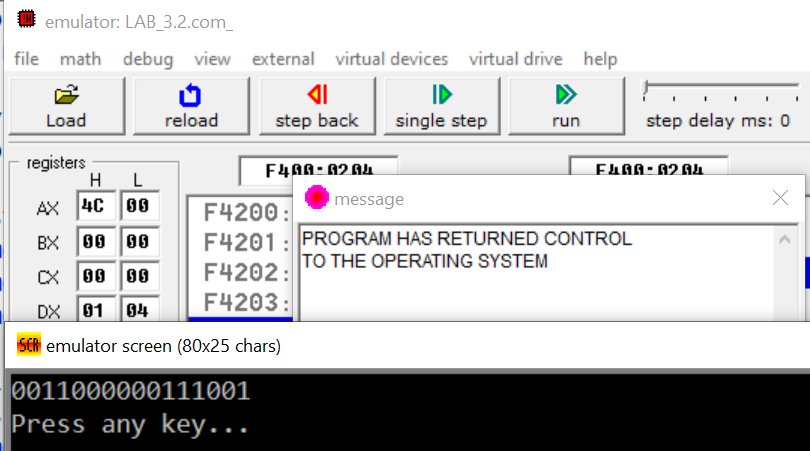


Рисунок 2 – Эмуляция программы 2

Код программы

Листинг 1 – Пробная программа 1

mov ax, 0255 ; сохранить значение 255 в регистр AX

inc ax ; увеличить значение AX на 1

add ax, alpha ; добавить значение alpha к AX

nop ; ничего не делать

mov bx, ax ; сохранить значение AX в регистре BX

dec bx ; уменьшить значение BX на 1

sub bx, beta ; вычесть значение beta из BX

mov dx, bx ; сохранить значение BX в регистре DX

sub dx, 10 ; вычесть 10 из значения DX

xchg ax, dx ; поменять значения AX и DX местами

push bx ; поместить значение BX в стек

push ax ; поместить значение AX в стек

pop cx ; извлечь значение из стека в CX

mov si, cx ; сохранить значение CX в регистре SI

mov di, dx ; сохранить значение DX в регистре DI

mov [0150h], cx ; сохранить значение CX в ячейке памяти 0150h

shl ax, 2 ; сдвинуть биты в AX на 2 позиции влево

mov dx, offset hello ; сохранить смещение строки hello в DX

mov ax, 0900h ; функция DOS - вывод строки

int 21h ; вызвать прерывание 21h (вывести строку hello)

mov ax, 4c00h ; функция DOS - передача управления ОС

int 21h ; вызвать прерывание 21h (вернуть управление ОС)

ret ; завершить программу

; определение переменных

alpha dw 25

beta dw 32

hello db 'Привет кафедра ИС!$'

Листинг 2 – Пробная программа 2

org 100h ; Программа начинается с адреса 100h

jmp start ; Безусловный переход на метку start

; -- Данные ------------------------------------------------------------

v dw 12345

pak db 13, 10, 'Press any key...$'

; ----------------------------------------------------------------------

start:

mov bx, [v] ; BX = v

mov ah, 2 ; Функция DOS 02h - вывод символа

mov cx, 16 ; Инициализация счётчика цикла

lp:

shl bx, 1 ; Сдвиг BX на 1 бит влево

mov dl, '0' ; dl = '0'

jnc print ; Переход, если выдвинутый бит равен 0

inc dl ; dl = dl + 1 = '1'

print:

int 21h ; Обращение к функции DOS 02h

loop lp ; Команда цикла

mov ah, 9 ; \

mov dx, offset pak ; > Вывод строки 'Press any key...'

int 21h ; /

mov ah, 8 ; \

int 21h ; / Ввод символа без эха

mov ax, 4C00h ; \

int 21h ; / Завершение программы

Вывод

В ходе выполнения данной лабораторной работы были изучены принципы функционирования 16-разрядного процессора Intel 8086. Были получены навыки изучения и комментирования ассемблерного кода для 16-разрядных процессоров.