

Лабораторная работа №2

«Исследование архитектуры и системы команд 16-разрядного процессора»

2.1 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.

2.2 Постановка задачи

Требуется изучить архитектуру МП 8086, состав регистров и работу процессора с использованием временных диаграмм (выполняется в процессе домашней подготовки к лабораторной работе). Изучить основные директивы ассемблера и команды МП 8086. Изучить функции BIOS и DOS и особенности использования их в ассемблерных программах. Запустить эмулятор, выбрать шаблон в формате com и ввести в окне редактора пробную программу, приведенную в приложении А. Исследовать процесс выполнения программы (т.е. проследить изменение содержимого регистров, оперативной памяти и стека). Пояснить в форме комментариев к каждой ассемблерной строке исследуемой программы. Запустить эмулятор, выбрать шаблон в формате com и ввести в окне редактора пробную программу, приведенную в приложении Б. Исследовать и пояснить изменения регистров при выполнении каждой из команд. Рассчитать время выполнения программ.

2.3 Ход работы

Были изучены архитектура МП 8086, состав регистров, работа процессора, основные директивы ассемблера и команды МП 8086. Изучены функции BIOS и DOS и их особенности использования в ассемблерных программах.

Был запущен эмулятор emu8086 и выбран шаблон в формате com. В окно редактора был добавлен код программы предложенной в методических указаниях программы. Были добавлены комментарии. Код программы с комментариями представлен в листинге 1.

Листинг 1 – Код первой программы с комментариями

```

mov ax,0255    ; записать в AX значение 255                ; + 4
inc ax         ; увеличить значение AX на 1                ; + 2
add ax, alpha  ; прибавить к AX значение alpha             ; + 16
nop           ; ничего не делать в течение цикла          ; + 1
mov bx,ax      ; записать в BX содержимое AX               ; + 2
dec bx         ; уменьшить значение в BX на 1              ; + 2
sub bx, beta   ; вычесть из значения в BX значение в beta  ; + 16
mov dx, bx     ; записать в DX значение в BX               ; + 2
sub dx,10      ; вычесть из DX число 10                    ; + 4
xchg ax,dx     ; поменять местами значения регистров AX и DX ; + 3
push bx        ; переместить в стек значение BX            ; + 3
push ax        ; переместить в стек значение AX            ; + 3
pop cx         ; взять значение из стека и поместить в CX   ; + 8
mov si,cx      ; записать в регистр SI значение CX         ; + 2
mov di,dx      ; записать в DI значение DX                 ; + 2
mov 0150h,cx   ; записать содержимое CX в ячейку памяти по адресу 0150h ; + 8
shl ax,2       ; сдвинуть биты в AX на 2 позиции влево    ; + 8 + 2 / 2
mov dx, offset hello ; записать в регистр DX значение отступа hello ; + 4
                    ; от начала сегмента
mov ax,0900h   ; записать в AX число 0900h                ; + 4
int 21h        ; вызвать прерывание 21h                    ; + 51
mov ax,4c00h   ; записать в AX число 4c00h                ; + 4
int 21h        ; вызвать прерывание 21h                    ; + 51
ret           ; вернуть управление операционной системе

alpha dw 25
beta dw 32
hello db 'Privet kafedra IS!$'

```

Время выполнения программы было высчитано следующим образом. Количество тактов при выполнении каждой команды было сложено, получено значение 201 такт. Так как частота тактового генератора равна 5МГц, то время выполнения программы равно $201 * \frac{1}{5 * 10^9} = 40,2 * 10^{-9} \text{с}$.

Аналогично была запущена и проанализирована вторая программа из методических указаний. Были добавлены комментарии. Код программы представлен в листинге 2.

Листинг 2 –Код второй программы с комментариями

```

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

```

Выводы

При выполнении данной работы было получено понимание работы 16 разрядного процессора Intel8086. Также были получены навыки комментирования и вычисления времени выполнения программы, написанной на языке ассемблера.