# ЛАБОРАТОРНАЯ РАБОТА №6

# ИССЛЕДОВАНИЕ МЕТОДОВ ВВОДА-ВЫВОДА ДАННЫХ В ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ

Цель работы

Изучить способы функционирования клавиатуры и подключения её к процессору, принципы отображения цифровой информации в жидкокристаллических дисплеях, методы программирования ввода-вывода данных. Исследовать особенности функционирования микропроцессора при реализации ассемблерных функций ввода данных с клавиатуры и вывода их на экран монитора. Приобрести практические навыки программирования на языке ассемблера МП 8086 процедур ввода-вывода с использованием функций BIOS.

Задачи

1. Изучить принцип устройства компьютерной клавиатуры и кодирования формируемых символов, основные функции BIOS, позволяющие обрабатывать состояния клавиатуры, принцип устройства жидкокристаллических мониторов и управления пикселями, основные функции BIOS, позволяющие упрощать программирование задач работы с клавиатурой и дисплеем;
2. Запустить в отладчике emu8086 программу вывода на экран VGA-монитора прямоугольника (emu8086\examples\0\_sample\_vga\_graphics.asm) и исследовать работу процессора при выполнении этой программы. Составить подробный алгоритм работы этой программы;
3. Модифицировать приведённую в примере программу, чтобы позволить ей:
   1. изменение размера отображаемого прямоугольника;
   2. изменение цвета фигуры;
4. Произвести отладку разработанных программ в пошаговом режиме и проследить за изменением содержимого регистров;

Ход работы

Была открыта программа 0\_sample\_vga\_graphics, содержащая пример работы с графикой в формате VGA через ассемблерный код. В результате её компиляции и запуска через эмулятор было открыто графическое окно, в котором был выведен прямоугольник (Рисунок 1).



Рисунок 1 – Вывод прямоугольника по умолчанию

Алгоритм программы несложный: режим работы видеоадаптера устанавливается на графический цветной. Затем по очереди выводятся 4 линии прямоугольника, отступы и длина линий задаются регистрами CX и DX, вывод пикселей происходит через циклы. В конце программа ожидает ввода символа, переходит в текстовый режим графики и завершается.

Далее программа была модифицирована таким образом, чтобы перед выводом пользователю было предложено ввести значения высоты, ширины и цвета прямоугольника. При вводе значений 100, 100 и 3 был выведен крупный квадрат голубого цвета (Рисунок 2).

Данное изображение соответствует желаемому (3 – голубой цвет). Программа была отлажена для различных входных значений, в том числе некорректных данных.

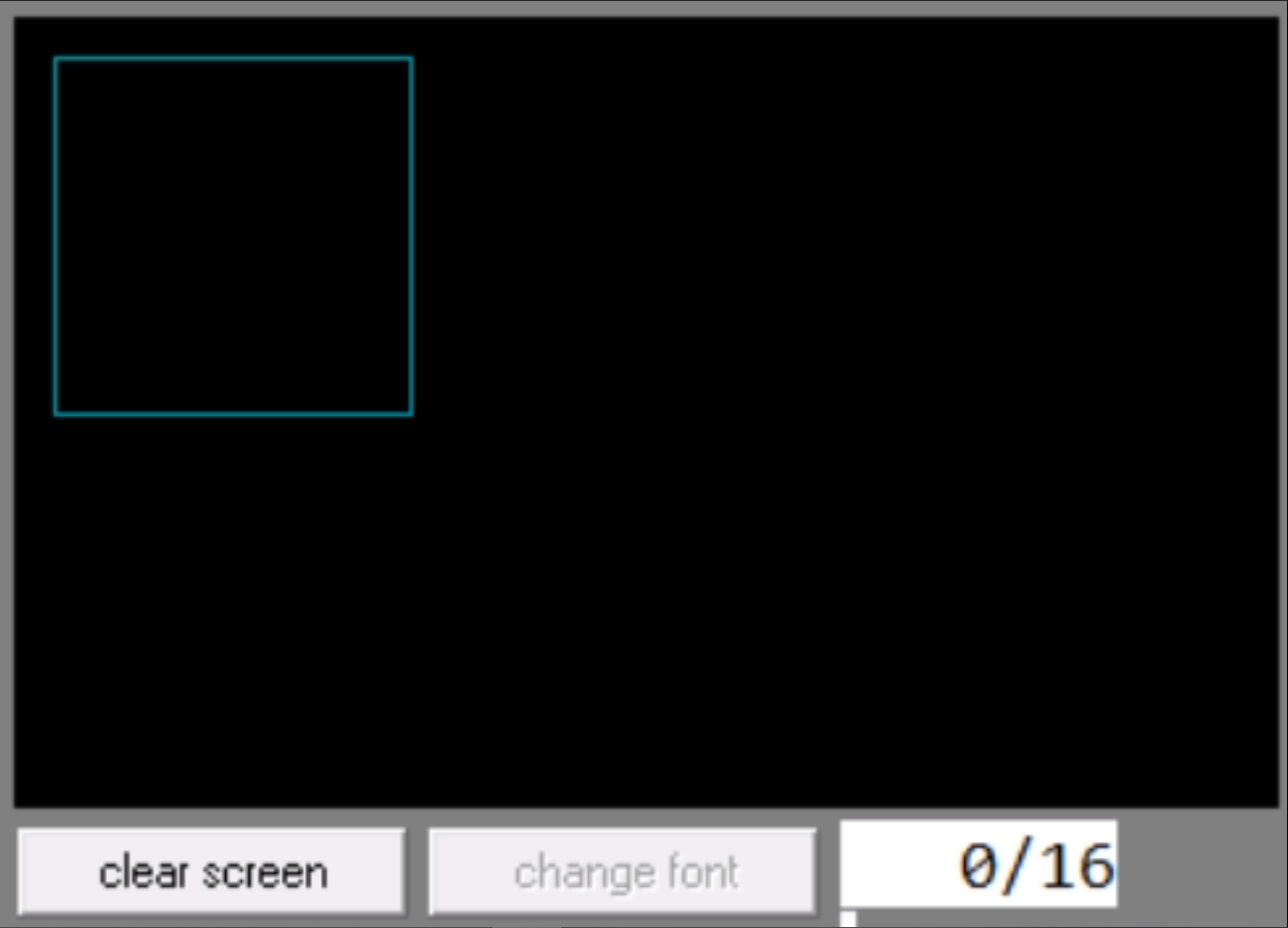


Рисунок 2 – Вывод прямоугольника по заданным параметрам

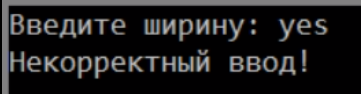


Рисунок 3 – Сообщение об ошибке для некорректного ввода

Код программы

Листинг 1 − 0\_sample\_vga\_graphics.asm (без модификаций)

name "vga"

; this program draws a tiny rectangle in vga mode.

org 100h

jmp code

; dimensions of the rectangle:

; width: 10 pixels

; height: 5 pixels

w equ 10

h equ 5

; set video mode 13h - 320x200

code: mov ah, 0

mov al, 13h

int 10h

; draw upper line:

mov cx, 100+w ; column

mov dx, 20 ; row

mov al, 15 ; white

u1: mov ah, 0ch ; put pixel

int 10h

dec cx

cmp cx, 100

jae u1

; draw bottom line:

mov cx, 100+w ; column

mov dx, 20+h ; row

mov al, 15 ; white

u2: mov ah, 0ch ; put pixel

int 10h

dec cx

cmp cx, 100

ja u2

; draw left line:

mov cx, 100 ; column

mov dx, 20+h ; row

mov al, 15 ; white

u3: mov ah, 0ch ; put pixel

int 10h

dec dx

cmp dx, 20

ja u3

; draw right line:

mov cx, 100+w ; column

mov dx, 20+h ; row

mov al, 15 ; white

u4: mov ah, 0ch ; put pixel

int 10h

dec dx

cmp dx, 20

ja u4

; pause the screen for dos compatibility:

;wait for keypress

mov ah,00

int 16h

; return to text mode:

mov ah,00

mov al,03 ;text mode 3

int 10h

ret

Листинг 2 – Модифицированная программа

org 100h

; Код программы

mov cx, 0

mov dx, offset enter\_x ; ввод ширины

mov ah, 09h ;

int 21h ;

call getInt ;

mov x2, ax ;

mov dx, offset enter\_y ; ввод высоты

mov ah, 09h ;

int 21h ;

call getInt ;

mov y2, ax ;

mov dx, offset enter\_col ; ввод цвета

mov ah, 9 ;

int 21h ;

call getInt ;

mov color, al ;

mov ah, 0 ; видеорежим = 13h (графика, 256, 320х200)

mov al, 13h ;

int 10h ;

mov cx, x1 ; текущая координата Х

mov dx, y1 ; текущая координата Y

mov ah, 0ch ; функция установки пикселя

mov bh, 0 ; видеостраница - 0

mov al, color ; установить цвет линий

c1:

int 10h ; установка пикселя

cmp dx, y2 ; заполнение по высоте (Y)

jne lp1\_y ;

cmp cx, x2 ; заполнение по ширине (X)

jne lp1\_x ;

jmp c1\_end ; выход из цикла

lp1\_y:

inc dx ; увеличить координату Y

jmp c1 ;

lp1\_x:

inc cx ; увеличить координату X

jmp c1 ;

c1\_end:

c2:

int 10h ; установка пикселя

cmp dx, y1 ; заполнение по высоте (Y)

jne lp2\_y ;

cmp cx, x1 ; заполнение по ширине (X)

jne lp2\_x ;

jmp c2\_end ; выход из цикла

lp2\_y:

dec dx ; увеличить координату Y

jmp c2 ;

lp2\_x:

dec cx ; увеличить координату X

jmp c2 ;

c2\_end:

mov ah, 01h ;

int 21h ; ожидание отклика

mov ax, 4c00h ;

int 21h ; конец программы

; Ввод целого числа

getInt proc

mov ah, 0ah ; команда считывания строки в буфер

mov dx, offset buff ; адрес буфера

int 21h ;

mov ah, 02h ; переход на новую строку

mov dl, 0dh ; символ абзаца

int 21h ;

mov dl, 0ah ; символ новой строки

int 21h ;

; обработка содержимого буфера

mov si, offset buff + 2 ; адрес начала строки

mov ax, 0 ; итоговое целое число

mov bx, 10 ; основание сc

check\_next:

mov cl, [si] ; рассматриваемый символ буфера

cmp cl, 0dh ; проверка окончания буфера

jz endin ;

cmp cl, '0' ; ошибка если символ <0

jb err

cmp cl, '9' ; ошибка если символ >9

ja err

sub cl, '0' ; преобразование символа в число

mul bx ; умножение ax на 10

add ax, cx ; прибавление нового разряда

inc si ; следующий символ

jmp check\_next ;

err:

mov dx, offset error ;

mov ah, 09h ;

int 21h ; вывод сообщения об ошибке

mov ax, 4c01h ;

int 21h ; завершение программы

endin:

ret ; конец обработки буфера

getInt endp

; Переменные

x1 dw 10 ; начальная координата x

y1 dw 10 ; начальная координата y

x2 dw 0 ; ширина прямоугольника

y2 dw 0 ; высота прямоугольника

color db 10 ; цвет линий

enter\_x db ’Введите ширину: $’ ;

enter\_y db ’Введите выысоту: $’ ;

enter\_col db ’Введите цвет: $’ ;

error db "Некорректный ввод! $" ; сообщение об ошибке

buff db 4, ?, 4 dup(?) ; буфер для строки ввода

Вывод

В ходе работы были изучены механизмы функционирования клавиатуры и подключения её к процессору, принципы отображения цифровой информации в жидкокристаллических дисплеях, методы программирования ввода-вывода данных. Были исследованы особенности работы микропроцессора при реализации ассемблерных функций ввода данных с клавиатуры и вывода их на экран монитора. Были приобретены практические навыки программирования на языке ассемблера МП 8086 процедур ввода-вывода с использованием функций BIOS.