

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

кафедра «Информационные системы»

Отчёт
по лабораторной работе №5
по дисциплине «Технические средства информационных систем»

Выполнил:
ст.гр. ИС/б-20-2-о
Филозоф А.Н.

Принял:
Минкин С.И.

Севастополь
2022 г.

Исследование методов ввода-вывода в персональных компьютерах**5.1. Цель работы**

Изучить способы функционирования клавиатуры и подключения ее 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 МП8086 процедур ввода-вывода с использованием функций BIOS.

5.2. Постановка задачи

Изучить принцип устройства компьютерной клавиатуры и кодирования формируемых символов, а также основные функции BIOS, позволяющие обрабатывать состояния клавиатуры (выполняется в процессе домашней подготовки к лабораторной работе). Изучить принцип устройства жидкокристаллических мониторов и управления пикселями (выполняется во время домашней подготовке к работе). Изучить основные функции BIOS, позволяющие упрощать программировать задачи работы с клавиатурой и дисплеем (выполняется во время домашней подготовке к работе). 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.

5.3. Ход выполнения работы

В процессе выполнения работы была написана программа, представленная в листинге 1.

Листинг 1 — Текст программы

```
.data
chr db 'F'
x1 dw 10 ; col
y1 dw 10 ; row
x2 dw 50
y2 dw 20
color db 6

.code
begin:
    mov ax, @data
    mov ds, ax
    mov es, ax

    call InputInt
    mov x2, ax
    call InputInt
    mov y2, ax

    mov ah, 0 ; 0 - установить видеорежим
    mov al, 13h ; Видеорежим = 13h (графика, 320x200)
    int 10h ; Прерывание.

    mov cx, x1 ; устанавливаем координату X
    mov dx, y1 ; устанавливаем координату Y
    mov ah, 0Ch ; Номер функции установки точки
                ; CX - строка (Y) ; DX - столбец (X)
    xor bh, bh ; видеостраница - 0
    mov al, color ; устанавливаем цвет

c1:
    int 10h ; вызываем прерывание и ставим точку
    cmp dx, y2 ; сравниваем со значением y2
    jne lp ; если не равно - goto LP
    cmp cx, x2 ; если равно - сравниваем с X2
    jne lp2 ; не равно - goto lp2
    jmp ex ; иначе - выходим из цикла (т.к. половину прямоугольника мы нарисовали)
lp:
    inc dx ; увеличиваем координату
    jmp c1
lp2:
    inc cx
    jmp c1

ex:
; аналогичный цикл на достроение 2 части прямоугольника
c2:
    int 10h
    cmp dx, y1
    jne lp3
    cmp cx, x1
    jne lp4
    jmp ex2
lp3:
    dec dx
    jmp c2
lp4:
    dec cx
    jmp c2
ex2:
    mov ah, 1
    int 21h
```

```

    mov ax, 4c00h
    int 21h
InputInt proc

    mov ah,0ah
    xor di,di
    mov dx,offset buff ; адрес буфера
    int 21h ; принимаем строку
    mov dl,0ah
    mov ah,02
    int 21h ; выводим перевода строки

; обрабатываем содержимое буфера
    mov si,offset buff+2 ; берем адрес начала строки
    cmp byte ptr [si],"-" ; если первый символ минус
    jnz ii1
    mov di,1 ; устанавливаем флаг
    inc si ; и пропускаем его
ii1:
    xor ax,ax
    mov bx,10 ; основание cc
ii2:
    mov cl,[si] ; берем символ из буфера
    cmp cl,0dh ; проверяем не последний ли он
    jz endin

; если символ не последний, то проверяем его на правильность
    cmp cl,'0' ; если введен неверный символ <0
    jb er
    cmp cl,'9' ; если введен неверный символ >9
    ja er

    sub cl,'0' ; делаем из символа число
    mul bx ; умножаем на 10
    add ax,cx ; прибавляем к остальным
    inc si ; указатель на следующий символ
    jmp ii2 ; повторяем

er: ; если была ошибка, то выводим сообщение об этом и выходим
    mov dx, offset error
    mov ah,09
    int 21h
    int 20h

; все символы из буфера обработаны число находится в ax
endin:
    cmp di,1 ; если установлен флаг, то
    jnz ii3
    neg ax ; делаем число отрицательным
ii3:
    ret

error db "incorrect number$"
buff db 6,7 Dup(?)
InputInt endp
end begin

```

После написания кода программы программа была запущена в среде для эмуляции 16-разрядного процессора. Рисунок 1 содержит содержимое экрана буфера — результат работы программы при вводе значений 80 и 100.

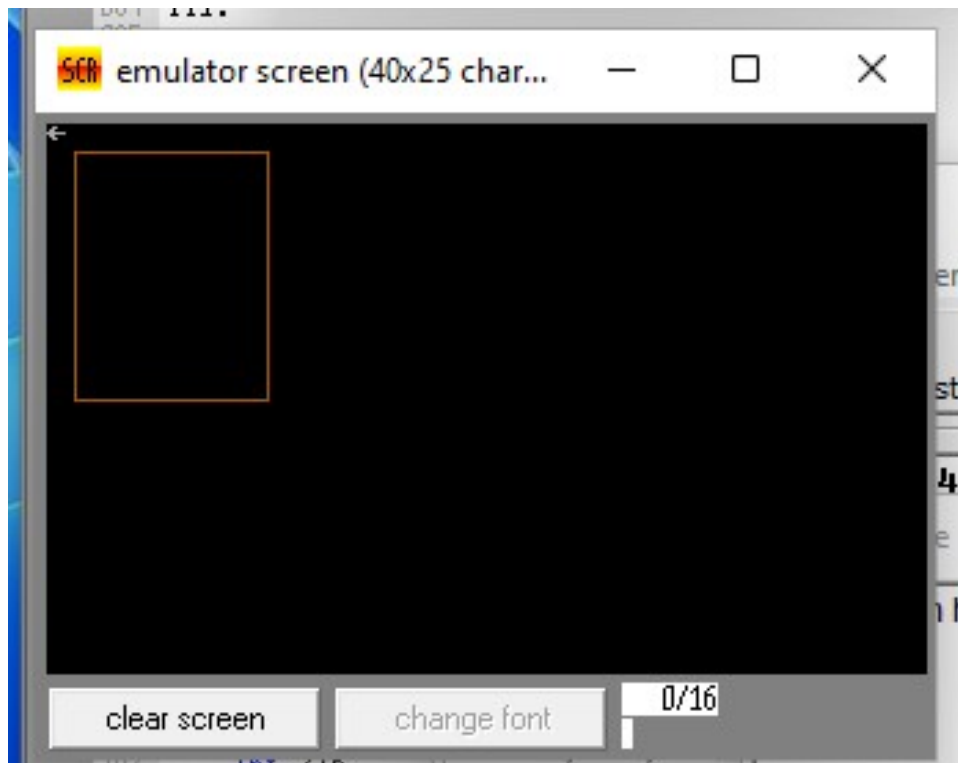


Рисунок 1 — Результат выполнения программы

Вывод

В ходе выполнения данной лабораторной работы были изучены способы функционирования клавиатуры и подключения ее к процессору, принципы отображения цифровой информации в жидкокристаллических дисплеях, методы программирования ввода-вывода данных. Были исследованы особенности функционирования 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.

Ответы на контрольные вопросы

1. ...

2. При нажатии байт скэн-кода содержит число в диапазоне от 1 до 83 (в стандартной клавиатуре XT). При отпускании генерируется скэн-код на 128 (80H) больше, чем скэн-код при нажатии клавиши. Например, при нажатии клавиши Z скэн-код 44, а при отпускании $172 = 44 + 128$. В клавиатуре AT при отжатии клавиши сначала посылается байт F0h, а затем код клавиши.

3. При нажатии на клавишу замыкаются контакты одной из ячеек матрицы кнопок. Установленный в клавиатуре контроллер получает скэн-код нажатой клавиши. Эти коды однозначно идентифицируют каждую клавишу и не могут быть переназначены. После обнаружения нажатия контроллер сравнивает скэн-код с картой символов в своей постоянной памяти. Если клавиша не отпускается в течение времени более 0.5 с, то клавиатура генерирует повторные коды нажатой клавиши.

4. ...

5. ...

6. ...

7. Сигнал запроса прерывания int 09h подаётся системным контроллером клавиатуры на процессор.

8. ...

9. Для работы с клавиатурой определены следующие функции: 00 — прочитывать текущий символ и записать его ASCII-код в регистр AL; 01 — проверить готовность.

10. Видеосистема компьютера состоит из видеоадаптера (видеокарты) и монитора (дисплея). Растровая развертка представляет собой набор непрерывных горизонтальных линий, последовательно заполняющих весь экран, то есть весь экран сканируется последовательно строка за строкой.

11. Качество монитора определяют следующие параметры: размер экрана, размер пикселя, частота кадров, технология изготовления матрицы.

12. Принцип действия и многослойная структура всех LCD TFT-дисплеев примерно одинаковы. Свет от лампы подсветки (неоновая или светодиоды проходит через первый поляризатор и попадает в слой жидких кристаллов, над которыми размещены пластины миниатюрных конденсаторов, управляемых тонкопленочными транзисторами. Транзистор с конденсатором создает электрическое поле, которое формирует ориентацию жидких кристаллов. Пройдя такую структуру, свет меняет свою поляризацию и будет или полностью поглощен вторым поляризационным фильтром (экран становится черным), или не будет поглощаться (экран – белый), или поглощение будет частичным (одна из градаций яркости цвета).

13. Цвет изображения определяют цветные фильтры. Каждый пиксель матрицы состоит из трех субпикселей – красного, зеленого и голубого.

14. Для взаимодействия с дисплеем определены следующие функции: 00 — переключить режим работы видеоадаптера на режим, установленный в AL; 05h — изменить активную страницу видеопамати на указанную в AL; 0Bh – установить цветовую палитру; 0Ch – вывод пикселя указанного цвета и находящегося по указанным координатам. Для начала выполнения функции необходимо инициировать прерывание 10h.