МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №5

по дисциплине “Технические средства информационных систем”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

«Информационные системы»

Минкин С. И.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №5**

**«ИССЛЕДОВАНИЕ МЕТОДОВ ВВОДА-ВЫВОДА ДАННЫХ В ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ»**

**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 МП 8086 процедур ввода-вывода с использованием функций BIOS.

**Постановка задачи**

1. 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).

2. Изучить принцип устройства жидкокристаллических мониторов и управления пикселами (выполняется во время домашней подготовке к работе).

3. Изучить основные функции BIOS, позволяющие упрощать программировать задачи работы с клавиатурой и дисплеем (выполняется во время домашней подготовке к работе).

4. 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. Модифицировать приведенную в примере программу, позволяющей: а) изменять размер отображаемого прямоугольника; б) изменение цвета фигуры.

6. Произвести отладку разработанных программ в пошаговом режиме и проследить за изменениями содержимого регистров.

7. Сформулировать выводы по работе.

**Ход работы**

Для начала выполнения работы была запущена заранее подготовленная программа (листинг 1), выводящая на экран монитора прямоугольник.

Листинг 1 – Код программы из набора примеров

; Название программы

name "vga"

; ---Эта программа рисует маленький прямоугольник при помощи видеоадаптера---

org 100h ; Директива, позволяющая начать адресацию со смещения 100h из-за резервирования DOS 256 байт (100h)

jmp code ; безусловный переход к метке кода

; Размеры прямоугольника:

; ширина(width): 10 пикселей

; высота(height): 5 пикселей

w equ 10 ; объявление константы w

h equ 5 ; объявление константы h

; Установить видео режим 13h в регистр al - 320x200 (размер экрана в пикселях)

code: mov ah, 0 ; установка значения 0013h в регистр аккумулятора для задания

mov al, 13h ; графического типа информации (256 цветов, 320х200 пикселей)

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

; Нарисовать верхнюю линию:

mov cx, 100+w ; столбец (сдвиг на 100+w пикселей вправо)

mov dx, 20 ; строка (сдвиг на 20 пикселей вниз)

mov al, 15 ; белый цвет

u1: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec cx ; сдвиг координаты X на один пиксель назад (влево)

cmp cx, 100 ; сравнить координату X с изначальным сдвигом по горизонтали

jae u1 ; перейти к метке u1, если условие сверху верно

; или если координата X = 0

; Нарисовать нижнюю линию:

mov cx, 100+w ; столбец (сдвиг на 100+w пикселей вправо)

mov dx, 20+h ; строка (сдвиг на 20+h пикселей вниз)

mov al, 15 ; белый цвет

u2: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec cx ; сдвиг координаты X на один пиксель назад (влево)

cmp cx, 100 ; сравнить координату X с изначальным сдвигом по горизонтали

ja u2 ; перейти к метке u2, если условие сверху верно

; Нарисовать левую линию:

mov cx, 100 ; столбец (сдвиг на 100 пикселей вправо)

mov dx, 20+h ; строка (сдвиг на 20+h пикселей вниз)

mov al, 15 ; белый цвет

u3: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec dx ; сдвиг координаты Y на один пиксель назад (вверх)

cmp dx, 20 ; сравнить координату Y с изначальным сдвигом по вертикали

ja u3 ; перейти к метке u3, если операнды верхней команды равны

; Нарисовать правую линию:

mov cx, 100+w ; столбец (сдвиг на 100+w пикселей вправо)

mov dx, 20+h ; строка (сдвиг на 20+h пикселей вниз)

mov al, 15 ; белый цвет

u4: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec dx ; сдвиг координаты Y на один пиксель назад (вверх)

cmp dx, 20 ; сравнить координату Y с изначальным сдвигом по вертикали

ja u4 ; перейти к метке u4, если операнды верхней команды равны

; Остановить экран для совместимости с DOS:

; Ожидание нажатия клавиши

mov ah,00 ; установка значения 0000h в регистр аккумулятора для задания

; чтения текущего символа

int 16h ; код прерывания 16h для доступа к клавиатурным функциям BIOS

; Возврат к текстовому режиму:

mov ah,00 ; установка значения 0003h в регистр аккумулятора для задания

mov al,03 ; текстового типа информации (16 цветов, экран 80х25, символы 8х8)

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

ret ; завершение программы

На рисунке 1 представлены результаты выполнения программы.



Рисунок 1 – Прямоугольник, нарисованный программой из набора примеров

Затем, программа была модифицирована (листинг 2). Размер фигуры был изменён с 10х5 пикселей на 92х92 пикселей. Также, верхняя сторона прямоугольника стала красной, правая сторона – зелёной, нижняя сторона – голубой, а левая – фиолетовой (рисунок 2).

Листинг 2 – Код модифицированной программы из набора примеров

; Название программы

name "vga2"

; ---Эта программа рисует маленький прямоугольник при помощи видеоадаптера---

org 100h ; Директива, позволяющая начать адресацию со смещения 100h из-за резервирования DOS 256 байт (100h)

jmp code ; безусловный переход к метке кода

; Размеры прямоугольника:

; ширина(width): 92 пикселя

; высота(height): 92 пикселя

w equ 92 ; объявление константы w

h equ 92 ; объявление константы h

; Установить видео режим 13h в регистр al - 320x200 (размер экрана в пикселях)

code: mov ah, 0 ; установка значения 0013h в регистр аккумулятора для задания

mov al, 13h ; графического типа информации (256 цветов, 320х200 пикселей)

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

; Нарисовать верхнюю линию:

mov cx, 100+w ; столбец (сдвиг на 100+w пикселей вправо)

mov dx, 20 ; строка (сдвиг на 20 пикселей вниз)

mov al, 4 ; красный цвет

u1: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec cx ; сдвиг координаты X на один пиксель назад (влево)

cmp cx, 100 ; сравнить координату X с изначальным сдвигом по горизонтали

jae u1 ; перейти к метке u1, если условие сверху верно

; или если координата X = 0

; Нарисовать нижнюю линию:

mov cx, 100+w ; столбец (сдвиг на 100+w пикселей вправо)

mov dx, 20+h ; строка (сдвиг на 20+h пикселей вниз)

mov al, 11 ; зелёный цвет

u2: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec cx ; сдвиг координаты X на один пиксель назад (влево)

cmp cx, 100 ; сравнить координату X с изначальным сдвигом по горизонтали

ja u2 ; перейти к метке u2, если условие сверху верно

; Нарисовать левую линию:

mov cx, 100 ; столбец (сдвиг на 100 пикселей вправо)

mov dx, 20+h ; строка (сдвиг на 20+h пикселей вниз)

mov al, 5 ; голубой цвет

u3: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec dx ; сдвиг координаты Y на один пиксель назад (вверх)

cmp dx, 20 ; сравнить координату Y с изначальным сдвигом по вертикали

ja u3 ; перейти к метке u3, если операнды верхней команды равны

; Нарисовать правую линию:

mov cx, 100+w ; столбец (сдвиг на 100+w пикселей вправо)

mov dx, 20+h ; строка (сдвиг на 20+h пикселей вниз)

mov al, 10 ; фиолетовый цвет

u4: mov ah, 0ch ; вставить пиксель

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

dec dx ; сдвиг координаты Y на один пиксель назад (вверх)

cmp dx, 20 ; сравнить координату Y с изначальным сдвигом по вертикали

ja u4 ; перейти к метке u4, если операнды верхней команды равны

; Остановить экран для совместимости с DOS:

; Ожидание нажатия клавиши

mov ah,00 ; установка значения 0000h в регистр аккумулятора для задания

; чтения текущего символа

int 16h ; код прерывания 16h для доступа к клавиатурным функциям BIOS

; Возврат к текстовому режиму:

mov ah,00 ; установка значения 0003h в регистр аккумулятора для задания

mov al,03 ; текстового типа информации (16 цветов, экран 80х25, символы 8х8)

int 10h ; код прерывания 10h для доступа к функциям BIOS видеоадаптера

ret ; завершение программы

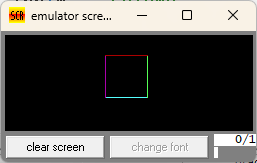


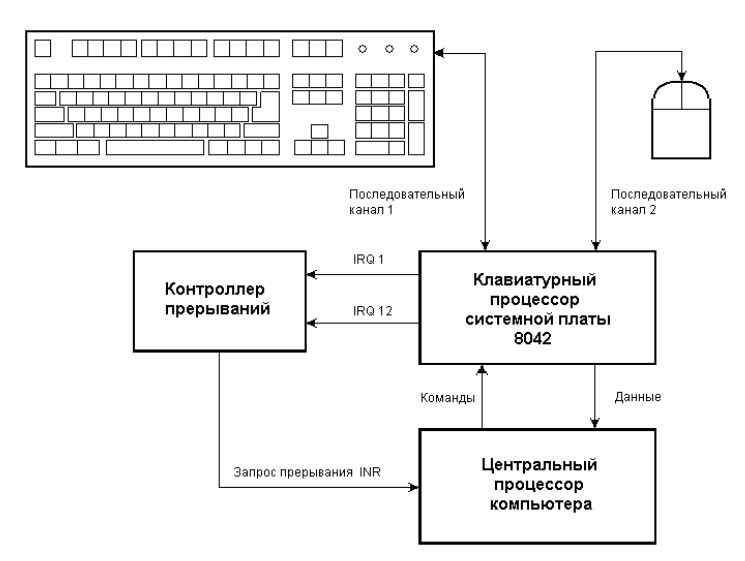
Рисунок 2 – Квадрат, нарисованный модифицированной программой из набора примеров

**Выводы**

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

**Контрольные вопросы**

1) Клавиатура - основное устройство ввода-вывода персонального компьютера (ПК).

****

2) Нажатие или отжатие клавиши сообщается ROM – BIOS посредством прерывания INT9, которое вызывает прикладную программу обработки прерываний. При этом читается состояние порта 96 (60H), через который осуществляется связь клавиатуры с BIOS.

3) Контроллер клавиатуры постоянно сканирует контакты клавиш; при обнаружении нажатой клавиши ее код фиксируется в буфере клавиатуры; если клавиша не отпускается в течение времени более 0.5 с, то клавиатура генерирует повторные коды нажатой клавиши; Каждый раз, когда нажимается или отпускается одна из клавиш, схема клавиатуры генерирует однобайтовое число, называемое скен -кодом, которое однозначно идентифицирует перемещение клавиши.

4) Работа по переводу скэн-кодов усложняется тем, что клавиатура IBM имеет несколько операций по изменению регистров, меняющих значение нажатых клавиш: Shift – С, Ctrl – C «break». Также меняется значение клавиши при нажатии клавиши Alt. Кроме этих регистровых клавиш есть еще CapsLock и NumLock.

5) При нажатии байт скэн-кода содержит число в диапазоне от 1 до 83 (в стандартной клавиатуре ХТ). 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 АТ при отжатии клавиши сначала посылается байт F0h, а затем код клавиши.