**Лабораторная работа по VGA**

**Формирование, вывод и анимация изображения на экране VGA монитора**

**Цель работы:** изучение использования VGA порта для вывода графической информации на экран монитора.

**1.Подготовка к лабораторной работе**

Для выполнения лабораторной работы необходимо изучить по материалам лекций и предложенной литературы:

1. Основные компоненты, используемые для вывода графической информации на экран монитора.
2. Принцип работы и назначение регистров и отдельных полей VGA порта. Эта информация содержится в описании процессорной системы «DE2-70 Media Computer». Включите ее в отчет.
3. Описание пиксельного и символьного буферов, используемых в процессорной системе. Данная информация содержится в описании процессорной системы «DE2-70 Media Computer».
4. Уясните пункты заданий методических указаний и напишите программные заготовки для их выполнения.

**Исходные файлы, используемые в работе**

- Файл **«**part\_1\_point\_4.c», содержит программу, которая делит экран на 4 полосы и позволяет закрасить одну из них выбранным цветом, номер полосы задается переключателями SW17-16, цвет задается переключателями SW15-0.

- Файл «part\_2\_point\_5.c», содержит программную заготовку, выводящую заданный текст с заданной начальной позиции (x; y) слева-направо.

- Файл «part\_4\_point\_1.c», содержит программу, рисующую прямую линию. Цвет линии задается при помощи 16 переключателей, после набора значения на переключателях необходимо нажать кнопку KEY1 (цвет задается 1 раз в начале программы).

Координаты начала и конца линии задаются с помощью переключателей. Первые 9 переключателей используются для задания координаты Х, остальные 9 переключателей – для задания координаты Y. При нажатии кнопки KEY3 на стенде, введенные на переключателях значения координат устанавливаются в качестве координат начала линии, при нажатии кнопки KEY2 введенные значения координат устанавливаются в качестве координат конца линии. Также содержит функцию, рисующую окружность, параметры окружности задаются в коде программы.

- Файл «part\_5\_point\_1.c», содержит программу, которая позволяет нарисовать на экране линию и перемещать её вправо по нажатию любой кнопки. После запуска программы задаются начальные координаты x и y (по 9 переключателей на x и y), после чего программа ждет нажатия любой кнопки (кроме KEY0). Далее точно таким же образом задаются конечные координаты x и y, для задания также необходимо нажать на любую кнопку. После этого происходит задание цвета линии, также с использованием переключателей, для выбора цвета нажать на любую кнопку. После этого рисуется линия и появляется возможность её перемещения в правую сторону.

**2. Порядок выполнения лабораторной работы**

**Часть 1. Использование пиксельного буфера для вывода цветного изображения на экран VGA монитора**

Для корректной компиляции заготовленных программ необходимо в меню Program Settings в поле Additional Compiler Flags указать ключ –std=c99.

1. Подключите разъем монитора к VGA разъему на стенде при выключенном питании стенда.
2. Включите питание стенда. Загрузите процессорную систему DE2-70 Media Computer в кристалл на стенде. Выполните подключение к системе Connect to system в меню Action > Connect to system. Сделайте снимок экрана, подключенного к процессорной системе монитора, используя мобильный телефон и поместите его в отчетные материалы.
3. Откройте вкладку Memory AMP. Вызовите правой клавишей мыши контекстно-зависимое меню. Используя команду Memory fill, заполните видеобуфер такими значениями, чтобы весь экран монитора закрасился черным цветом. Добавьте в отчет содержимое видеобуфера и снимок экрана.
4. Задайте адрес первого пикселя VGA с координатой (0,0) и запишите по нему значение, соответствующее белому цвету. Если пиксель не отобразился, то необходимо сместить изображение на мониторе (кнопками на самом мониторе). Аналогичным образом, закрасьте белым цветом соседний пиксел. Аналогичным образом закрасьте белым цветом всю первую строку экрана с координатой Y=0.
5. Определите диапазон адресов пиксельного буфера за пределами первой строки, который не отображается на экране монитора (невидимая зона). Экспериментально подтвердите свое предположение, меняя содержимое этой области памяти во вкладке Memory. Отразите в отчете размер невидимой зоны.
6. Повторите выполнение пункта 1.4 для второй строки. Используйте голубой цвет (или иной, в соответствии с вариантом).

Повторите выполнение пункта 1.4 для третьей строки. Используйте зеленый цвет (или иной в соответствии с вариантом).

Повторите для последней строки экрана, используя белый цвет (или иной в соответствии с вариантом). Включите в отчет снимок экрана.

7. Закрасьте экран в соответствии с вашим вариантом (Таблица 1). Также включите результат в отчет. Включите в отчет снимок экрана и содержимое графического буфера, соответствующее реализованной раскраске.

*Таблица 1. Варианты цветов.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вариант** | **Номер строки** | | | |
| **1** | **2** | **3** | **Последняя** |
| **1** | Белый | зеленый | красный | синий |
| **2** | Белый | синий | красный | белый |
| **3** | Белый | красный | зеленый | синий |
| **4** | Белый | белый | синий | зеленый |
| **5** | Белый | зеленый | белый | синий |
| **6** | Белый | синий | зеленый | красный |
| **7** | Белый | зеленый | красный | белый |
| **8** | Белый | красный | синий | Зеленый |
| **9** | Белый | красный | зеленый | синий |

8. Запустите программу, «part\_1\_point\_4.c» и экспериментально определите выполняемые ею действия. Меняя состояние переключателей, наблюдайте работу программы. Задавайте цвета: белый, голубой, красный, зеленый, черный и другие цвета радуги. Заполните в отчете таблицу соответствия цветов и численных значений, используемых для их кодирования. Используйте файл «Таблица\_цветов».

9. Используя вкладку Memory AMP, наблюдайте содержимое программно доступных регистров контроллера VGA порта (0x10003020). Уясните назначения этих регистров и их отдельных полей. Поместите в отчет наблюдаемые значения, и ваши комментарии.

10. Заполните буфер заднего плана таким образом, чтобы экран монитора закрашивался черным цветом. Используйте статическую память начиная с адреса 0х08000000. Выполните переключение отображаемых кадров. Наблюдайте изменение цвета экрана монитора и регистров контроллера VGA порта. Выполните повторное переключение кадров. Зафиксируйте в отчете выполняемые действия и наблюдаемые результаты.

**Часть 2. Использование символьного буфера для вывода текстовой информации на экран**

1. Используя вкладку Memory AMP, определите в ОП местоположение символьного буфера. Запишите в начало символьного буфера ASCII код заглавного символа, с которого начинается Ваша фамилия. Используйте латинский алфавит и таблицу ASCII кодов, приведенную в файле «Таблица ASCII». Наблюдайте выводимый символ на экране монитора. Запишите в буфер новое значение так, чтобы в соседней позиции вывелась вторая буква вашей фамилии.
2. Определите адрес в символьном буфере, соответствующий последнему символу первой строки. Выведите в этой позиции последнюю букву вашей фамилии.
3. Определите область ОП символьного буфера за пределами первой строки, которая не отображается на экране VGA. Экспериментально подтвердите свое предположение, записывая в неё произвольные символы. Отразите в отчете.
4. Запишите в символьный буфер новые значение так, чтобы заглавная буква вашего имени появилась в начале второй строки, в конце второй строки- последняя буква имени, в левом нижнем углу экрана - заглавная буква вашего отчества, в правом нижнем углу - строчная последняя буква вашего отчества. Сделайте снимок экрана и поместите его в отчет.
5. Используя символьный буфер, выведите на экран VGA вашу фамилию и имя, используя латинский алфавит, в соответствии с заданным в варианте способом, начиная с указанной позиции. Напишите программу, выполняющую заполнение символьного буфера, в соответствии с заданным вариантом (Таблица 2), используя заготовку «part\_2\_point\_5.c», перед написанием создайте копию заготовки. Поместите текст отлаженной вами программы в отчет и снимок экрана, подтверждающий успешное выполнение этого пункта задания.

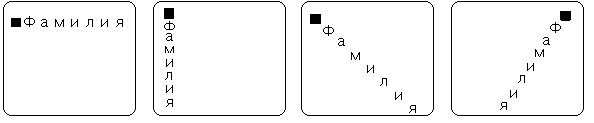
*Таблица 2. Варианты вывода текста.*

|  |  |  |
| --- | --- | --- |
| Вариант | Способ вывода | Начальные координаты (x; y) |
| 1 | Справа-налево (д) | (65; 12) |
| 2 | Снизу по диагонали направо (з) | (12; 49) |
| 3 | Слева-направо (а) | (9; 12) |
| 4 | Сверху по диагонали налево (г) | (38; 27) |
| 5 | Сверху по диагонали направо (в) | (33; 15) |
| 6 | Снизу-вверх (е) | (1; 11) |
| 7 | Снизу по диагонали налево (ж) | (66; 48) |
| 8 | Сверху-вниз (б) | (61; 47) |
| 9 | Сверху по диагонали налево(г) | (63; 49) |

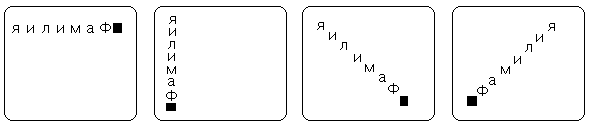
1. Ниже приведены способы вывода строки с вашей фамилией. Закрашенный символ означает начальную позицию, задаваемую предыдущей таблицей.

*Рисунок 1. Направление вывода текста.*

а б в г



д е ж з



7\*. Модифицируйте программу из предыдущего пункта так, чтобы строка с вашей фамилией перемещалась по экрану в заданном таблицей направлении.

**Часть 3. Использование встроенного шрифта для генерации символов на экране VGA**

1. Напишите программу на ассемблере или Си, которая будет заполнять

символьный буфер последовательными значениями так, чтобы в строке с координатой У, выводились символы, которым соответствует ASCII код, старшая тетрада которого совпадает с порядковым номером строки. То есть на нулевой строке (верхней) коды в диапазоне 0x00 – 0x0F, на первой строке

0x10 – 0x1F, на второй строке 0x20 – 0x2F и так далее. На строке с номером 0xF - 0xF0 – 0xFF. Проанализируйте выводимые на экран символы. Сделайте снимок экрана и поместите его в отчет. Сравните символы, отображаемые на экране с представленными в таблице ASCII, приведенную в файле «Таблица ASCII». Обратите внимание на то, как отображаются на экране VGA символы, коды которых начинаются с единичного бита. Сделайте заключение о поддержке символов кириллицы.

2. В файлах проекта процессорной системы DE 2 -115 Media Computer есть файл «altera\_up\_video\_char\_mode\_rom.v» , который на языке Verylog описывает модуль постоянной памяти, используемый для хранения шрифта применяемого алфавита. Файл «altera\_up\_video\_char\_mode\_rom\_128.mif» используется для инициализации этого модуля постоянной памяти, после загрузки процессорной системы в кристалл ПЛИС.

3. Откройте файл «altera\_up\_video\_char\_mode\_rom\_128.mif», находящейся в папке с лабораторной работой и сопоставьте содержимое файла с таблицей ASCII.

4. Следующим шагом необходимо загрузить руссифицированную систему, файлы которой находятся в папке с материалами лабораторной работы (DE2\_XXX\_Media\_Computer\_RUS). Для этого необходимо произвести отключение от загруженной системы в приложении «AMP», после чего во вкладке «Settings->System settings» выбрать «Custom system» и загрузить системные файлы из папки с названием соответствующего стенда, в директории «vhdl», и выбрать соответствующие файлы, в названии файла «.sof» должно присутствовать словосочетание «Time\_limited». Загрузите систему.

5. Повторите выполнение пункта 3.1 для расширенной таблицы. Поместите в отчет снимок экрана, подтверждающий успешное выполнение этого пункта задания.

6\*. Модифицируйте систему DE2\_XXX\_Media\_Computer\_MOD, изменив файл altera\_up\_video\_char\_mode\_rom\_128.mif, находящийся в папке с модифицированной системой (DE2\_XXX\_Media\_Computer\_MOD/vhdl/nios\_system/synthesis/submodules), начиная с адреса 8192 и добавьте свой символ в файл. Скомпилируйте файлы. Загрузите систему, используя новые файлы.

7\*. Повторите вывод таблицы и добавьте результат в отчет.

**Часть 4. Вывод прямых линий на экран монитора**

1. Запустите программу, «part\_4\_point\_1.c» и экспериментально определите выполняемые ею действия. Меняя состояние переключателей, наблюдайте работу программы.
2. Используя программу последовательно нарисуйте:

Горизонтальную линию;

Вертикальную линию;

Линию под углом 45◦.

Пронаблюдайте как линия выводится на экран. Выведите линию под углом менее 45◦. И сравните результат с линией под наклоном 45◦. Как угол влияет на изображение линии на экране. Запишите результат в отчет.

3. Отредактируйте программу (уберите комментарий с функции draw\_circle()), ознакомьтесь с работой программы.

4. Используя функции из программы «part\_4\_point\_1.c» реализуйте вывод фигуры в соответствии с вариантом (Таблица 3).

*Выводить изображение в центре экрана.*

*Таблица 3*

|  |  |
| --- | --- |
| Вариант | Фигура |
| 1 | Квадрат (со стороной 100) |
| 2 | Равносторонний треугольник (со стороной 100) |
| 3 | Прямоугольник (со сторонами 50 и 100) |
| 4 | Равнобедренный прямоугольный треугольник (с гипотенузой 100) |
| 5 | Ромб (со сторонами 100) |
| 6 | Окружность с перпендикулярными диаметрами внутри (радиус 50) |
| 7 | Четверть круга (радиус 50) |
| 8 | Полукруг (с радиусом 50) |
| 9 | Квадрат, вписанный в окружность (радиусом 50) |

**Часть 5. Анимация**

1. Запустите программу «part\_5\_point\_1.c», экспериментально определите выполняемые ею действия.
2. Изучите код программы. Модернизируйте программу таким образом, чтобы кроме исходного перемещения происходило перемещение в другом направлении в соответствии с вариантом (Таблица 4). Назначьте перемещение в заданном направлении на любую свободную кнопку. Добавьте результат в отчет.

*Таблица 4*

|  |  |  |
| --- | --- | --- |
| Вариант | Перемещение | Начальные координаты (x; y) |
| 1 | Влево | (130; 162) |
| 2 | Вниз | (142; 155) |
| 3 | Вверх | (99; 120) |
| 4 | Влево | (138; 127) |
| 5 | Вниз | (133; 151) |
| 6 | Вверх | (122; 131) |
| 7 | Влево | (96; 118) |
| 8 | Вниз | (161; 147) |
| 9 | Вверх | (138; 149) |

**3. Отчетные материалы.**

Отчет должен включать в себя:

1. Цель работы.

2. Краткие сведения из теоретической части.

3. Информацию по выполнению пунктов задания.

1. Краткое заключение.