Министерство науки и высшего образования Российской Федерации
Мытищинский филиал
Федерального государственного автономного образовательного учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ КОСМИЧЕСКИЙ

КАФЕДРА К-3

отчет

К ЛАБОРАТОРНОЙ РАБОТЕ

по ДИСЦИПЛИНЕ

«ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ»

Студент КЗ-76Б Чернов В.Д.
(Группа) (И.О.Фамилия)
Студент КЗ-76Б Сериков Д.Е.
(Группа) (И.О.Фамилия)

Преподаватель Н.В.Ефремов
(И.О.Фамилия)

2025 г.

Описание лабораторной работы

Тема:

Лабораторная работа по VGA; Формирование, вывод и анимация изображения на экране VGA монитора

Цель работы:

Изучение VGA порта для вывода графической информации на экран монитора.

Используемые файлы программ:

Файл «part_1_point_4.c», содержит программу, которая делит экран на 4 горизонтальные 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 текстовую строку с заданной начальной позиции (x; y) слева-направо.

Файл «part_4_point_1.c», содержит программу, изображающую прямую линию на экране VGA. Цвет линии задается при помощи 16 переключателей, после набора значения на переключателях необходимо нажать кнопку KEY1 (цвет задается 1 раз в начале выполнения программы). Координаты начала и конца линии задаются с помощью переключателей. Первые 9 переключателей используются для задания координаты X, 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 данная программа содержит функцию, рисующую окружность, параметры которой задаются в коде программы.

Файл «part_5_point_1.c», содержит программу, которая позволяет рисовать 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 линии x и y, для задания также необходимо нажать на любую кнопку. После этого происходит задание цвета линии, также с использованием переключателей, для выбора цвета надо нажать на любую кнопку. После этого рисуется линия и появляется возможность её перемещения в правую сторону.

Часть 1.

Использование графического 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.

Подключите VGA монитор к соответствующему разъему стенда DE 2-70 или DE 2-115 при выключенном питании стенда. Включите питание стенда. Реализуйте процессорную систему DE2-70 Media Computer или DE 2-115 Media Computer в стенде, загрузив соответствующий конфигурационный файл в кристалл ПЛИС стенда.

Выполните подключение к системе, выполнив команду Connect to system в меню Action. Сделайте снимок экрана, подключенного к процессорной системе VGA монитора, используя мобильный телефон и поместите его в отчетные материалы.

Рисунок 1.

Рисунок 2.

Откройте вкладку Memory AMP. Вызовите правой клавишей мыши контекстно-зависимое меню. Используя команду Memory fill, заполните графический видеобуфер такими значениями, чтобы весь экран монитора закрасился черным цветом. Экспериментально проверьте, входит ли в заполняемый диапазон памяти конечный адрес и отразите в отчете. Добавьте в отчет содержимое видеобуфера и снимок экрана.

Задайте во вкладке Memory адрес пикселя VGA экрана с координатой (0,0) и запишите по нему значение, соответствующее белому цвету. Если на экране пиксель не отобразился, то попробуйте сместить изображение, используя регулировочные колёсики на самом мониторе. Аналогичным образом закрасьте белым цветом соседний справа пиксель. Аналогичным образом закрасьте белым цветом всю строку экрана с координатой Y=0.

Определите диапазон адресов графического буфера за пределами нулевой 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 и диапазон адресов невидимой зоны.

Запустите программу, «part_1_point_4.c» и экспериментально определите выполняемые ею действия. Программа делит экран монитора на 4 горизонтальных полосы и позволяет закрасить одну из них выбранным цветом

Номер закрашиваемой полосы задается переключателями SW17-16, цвет полосы задается переключателями SW15-0. Меняя состояние переключателей, наблюдайте работу программы. Задавайте цвета: белый, голубой, красный, зеленый, черный и другие цвета радуги. Заполните в отчете таблицу соответствия цветов и числовых значений, используемых для их кодирования.

Рисунок 3. Цветные полосы.

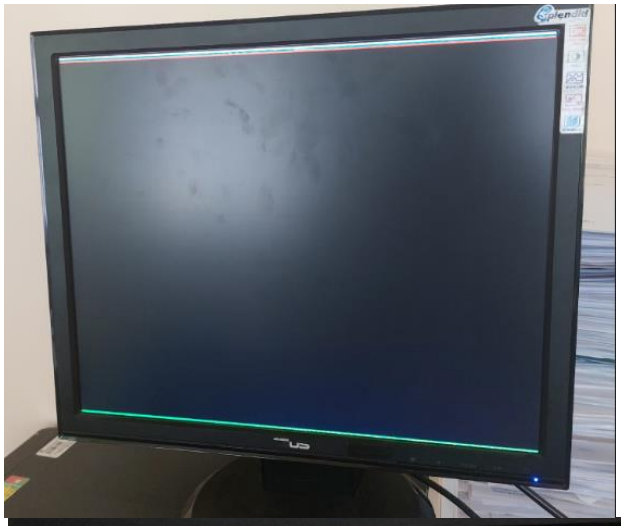
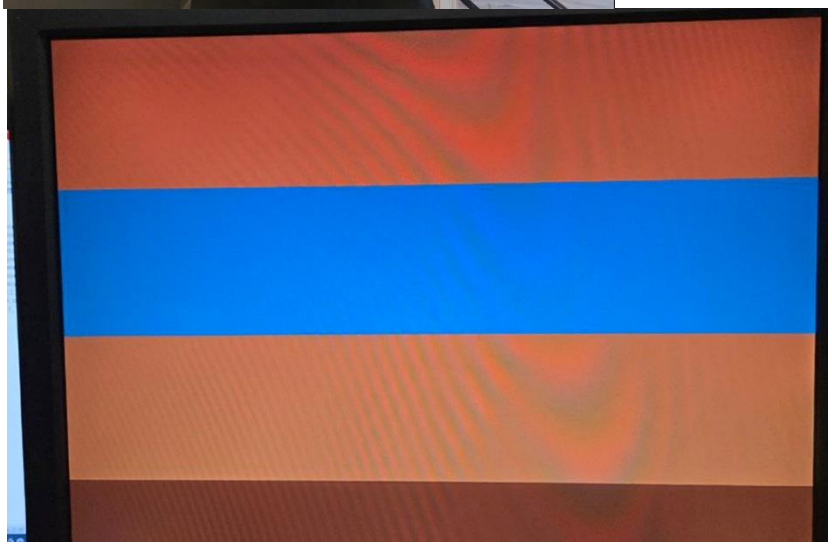


Рисунок 4. Тоже цветные полосы



	R	G	B
Цвет	0xFFFFF	0xFFFFFF	0xFFFFF
Красный	0x1f	0x0	0x0
Оранжевый	0x1f	0x18	0x0
Желтый	0x1f	0x3f	0x0
Зеленый	0x0	0x1f	0x0
Голубой	0x7	0x20	0x1f
Синий	0x0	0x0	0x1f
Фиолетовый	0x0	0x1f	0x1f
Коричневый	0x17	0x1c	0x1
Черный	0x00	0x00	0x00
Белый	0x1f	0x3f	0x1f

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

Заполните буфер заднего плана таким образом, чтобы экран монитора закрашивался заданным в таблице 1 цветом. Используйте для него статическую память за пределами основного буфера. Выполните переключение отображаемых кадров. Наблюдайте изменение цвета экрана монитора и регистров контроллера VGA порта. Выполните повторное переключение отображаемых кадров. Зафиксируйте в отчете выполняемые действия и наблюдаемые результаты. На одном кадре должны быть видны закрашенные полосы экрана в соответствии с вариантом, на другом одноцветный фон.

Проверьте экспериментально, можно ли после того, как выполнено переключение на буфер заднего плана изменить местоположение буфера переднего плана, задав его новый адрес в поле адреса буфера заднего плана. Свои наблюдения отразите в отчете.

Часть 2.

Использование символьного буфера для вывода текстовой информации на экран VGA

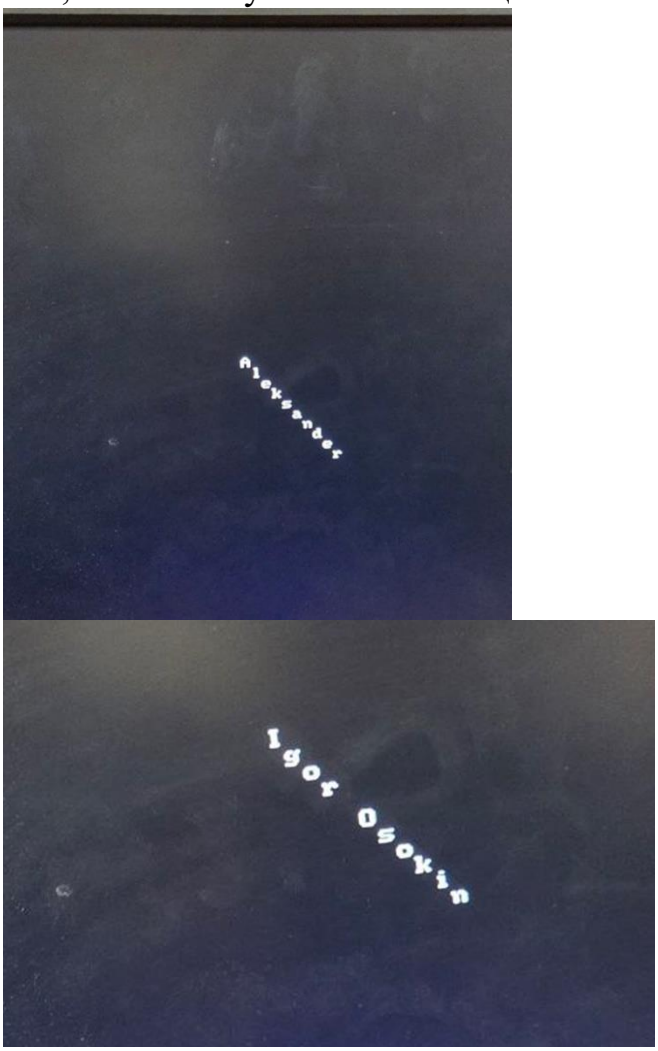
Используя вкладку Memory AMP, наблюдайте содержимое символьного буфера. Зафиксируйте в отчете свои наблюдения. Запишите в начало буфера ASCII код заглавного символа, с которого начинается ваша фамилия. Используйте латинский алфавит и таблицу ASCII кодов, приведенную в

приложении. Наблюдайте выводимый символ на экране монитора. Запишите в буфер новое значение так, чтобы в соседней позиции строки вывелась вторая строчная буква вашей фамилии.

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 размер и диапазон адресов невидимой части символьного буфера.

Запишите в символьный буфер новые значение так, чтобы заглавная буква вашего имени появилась в начале второй строки, в следующей позиции - вторая буква, а в конце второй строки- последняя буква имени, в последней строке экрана в его левом нижнем углу - заглавная буква вашего отчества, затем вторая буква, а в конце строки в правом нижнем углу - строчная последняя буква вашего отчества. Сделайте снимок экрана и поместите его в отчет.

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.



Часть 3.

Экспериментальное определение встроенного шрифта для генерации символов на экране VGA и его редактирование

Напишите программу на ассемблере или Си, которая будет заполнять символьный буфер последовательными значениями так, чтобы в строке экрана с координатой y, выводились символы, которым соответствует ASCII код, старшая тетрада которого совпадает с порядковым номером строки.

В файлах проекта процессорной системы DE 2 -115 Media Computer есть файл altera_up_video_char_mode_rom.v, который на языке Verilog описывает модуль постоянной памяти, используемый для хранения шрифта, применяемого в процессорной системе. Одноименный файл с расширением .mif требуется для инициализации этого модуля постоянной памяти, после загрузки процессорной системы в кристалл ПЛИС.

Используя приложение QUARTUS II найдите этот файл, откройте его и уясните его содержимое. Сопоставьте его бинарное содержимое с видом выводимых на экран VGA символов. Поместите в отчет бинарное представление первых трех, четырех латинских букв вашей фамилии, изображение которых не совпадает с русскоязычными аналогами. Например, для фамилии Ефремов такими символами являются f, r, m, v.

Для этих символов латиницы создайте изображения их русскоязычных аналогов. Для рассмотренного примера это будут символы ф, р, м, в. Используйте для этого матрицу 8*8 точек. Чтобы обеспечить минимальный интервал между текстовыми строками при выводе их на экран VGA, нижнюю битовую строку

символа оставьте нулевой. Поместите изображения созданных вами символов кириллицы в отчет.

Сохраните файл altera_up_video_char_mode_rom_128.mif с новым именем. Внесите изменения в созданный файл таким образом, чтобы стал возможен вывод всей или части вашей фамилии с использованием символов русского алфавита.

Выполните модификацию процессорной системы с целью изменения файла инициализации памяти знакогенератора. Используйте для этого файл, подготовленный при выполнении предыдущего пункта задания.

Повторите выполнение пункта 3.1 с использованием модифицированной процессорной системы и убедитесь, что теперь на экран VGA вместо трех, четырех символов латиницы выводятся их русскоязычные аналоги. Включите в отчет снимок экрана с выделенными позициями добавленных символов.

Повторно выполните программу из пункта 5 второй части задания, которая теперь выведет вашу фамилию на экран VGA заданным способом с использованием символов кириллицы. Наблюдайте работу программы. Поместите в отчет снимок экрана, подтверждающий успешное выполнение этого пункта задания.

Русификация процессорной системы DE 2-115 Media Computer

В процессорной системе DE 2-70 и DE 2-115 первоначально использовалось семь бит для кодирования символов в коде ASCII, что не позволяло работать с русскоязычными текстами. Следует напомнить, что символьный буфер в процессорной системе реализован на блочной памяти внутри кристалла (onchip memory). При выполнении лабораторной работы по ОЭВМ выполнялось тестирование этого компонента ОП, в результате которого был выявлен факт, что эта память имеет семибитную организацию. С целью русификации процессорная система была модифицирована.

Изменены следующие аппаратные модули: nios_system_VGA_Char_Buffer и altera_up_video_char_mode_rom_128. Первый из них описывает символьный буфер, реализованный внутри кристалла ПЛИС на ресурсах блочной памяти. Теперь этот буфер организован как восьмиразрядное ОЗУ с произвольным доступом. Второй модуль представляет собой ПЗУ знакогенератора. Так как для кодирования символов теперь используется восьмиразрядный код, его размер увеличен вдвое за счет добавления в ПЗУ восьмого адресного входа.

Теперь ПЗУ знакогенератора будет хранить изображения 256 символов. Так как каждый символ задается матрицей 8*8 пикселей, то есть восьмью байтами, то общий объем памяти знакогенератора составит 256*8 =2048 байт. Эта память также построена на блочной памяти внутри кристалла. Потребуется создать новый, расширенный файл инициализации ПЗУ знакогенератора. Таким образом, после модификации процессорная система сможет работать со 128 дополнительными символами, включая прописные и заглавные буквы русского алфавита.

Экспериментальное определение кодов русскоязычных символов в кодировках UTF -8 и ANSI.

Используя директиву. asciz, определите коды букв русского алфавита, формируемые после компиляции программы в приложении IMP, причем как прописных, так и заглавных. Для этого сегмент данных с текстовой строкой разместите отдельно от кода программы в начале статической памяти SRAM в процессорной системе. Текстовая строка содержит все буквы русского алфавита, причем как заглавные, так и прописные. После компиляции и загрузки сегмента данных в ОП найдите в начале статической памяти текстовую строку с буквами русского алфавита. Используйте вкладку Memory и побайтовое отображение ОП.

Наблюдайте ASCII коды отдельных букв кириллицы и заполните подготовленную таблицу. Рассмотрите варианты использования в текстовом редакторе, в котором создается или редактируется исходный файл с буквами латиницы, кодировок UTF-8 и ANSI.

Часть 4. Вывод прямых линий на экран монитора

Запустите программу «part_4_point_1.c» и экспериментально определите выполняемые ею действия. Меняйте состояние переключателей и наблюдайте работу программы. Удостоверьтесь в правильности работы программы, задавая на переключателях различные координаты начала и конца линии, меняйте цвет линии.

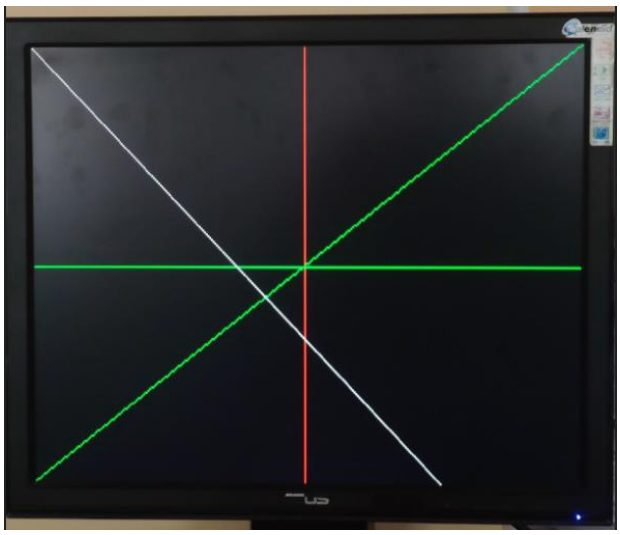
Используя программу, последовательно изобразите на экране:

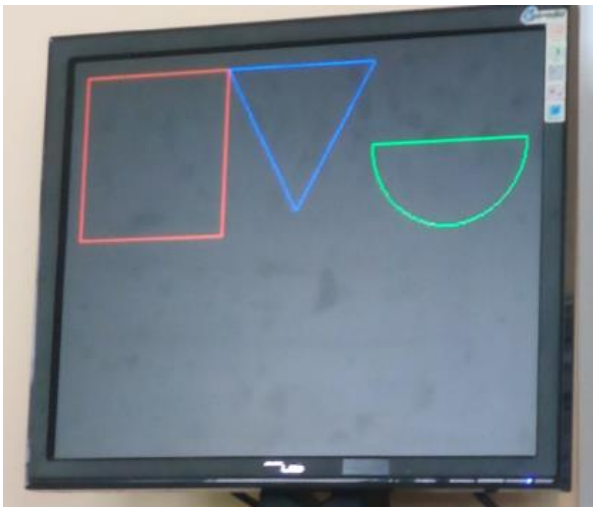
- Сквозную горизонтальную линию, проходящую через центр экрана;
- Сквозную вертикальную линию, проходящую через центр экрана;
- Линию под углом 45 градусов, исходящую из одного из углов экрана, проходящую через весь экран.

Наблюдайте, как прямая линия выводится на экран. Пробуйте изменить угол её наклона и наблюдайте, как угол влияет на изображение линии на экране. Поместите свои наблюдения в отчет.

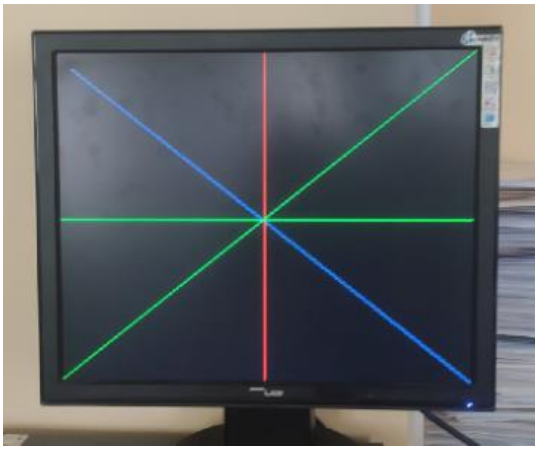
Выполните редактирование программы, удалив комментарии с функции draw_circle (). Наблюдайте работу измененной программы. Сделайте снимок экрана и поместите его в отчет.

Используя функции из программы «part_4_point_1.c» реализуйте вывод заданной фигуры в центре экрана в соответствии с заданным в таблице 3 вариантом. Числовое значение в таблице задает линейный размер фигуры в пикселях. Сделайте снимок экрана, подтверждающий успешное выполнение программы, и поместите его в отчетные материалы.





Модифицируйте программу из предыдущего пункта. Реализуйте задание координат начала и конца линии с помощью переключателей. Первые 9 переключателей используются для задания координаты X, 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.



Вывод:

Мы научились: вносить изменения в аппаратные компоненты используемой процессорной системы; генерировать файлы модифицированной системы, включая файл прошивки кристалла ПЛИС; загружать прошивку модифицированной процессорной системы в кристалл ПЛИС учебного стенда; исследовать её возможности.