Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе № 11**

Дисциплина: Автоматизация проектирования дискретных  
устройств (на английском языке).

Выполнил студент гр. 5130901/10101 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Л. Симоновский (подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.П. Антонов (подпись)

“21” апреля 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Список иллюстраций: 2](#_Toc164603807)

[2. Задание: 3](#_Toc164603808)

[2.1. Цель задания: 3](#_Toc164603809)

[2.2. Структура проекта: 3](#_Toc164603810)

[3. Ход работы: 3](#_Toc164603811)

[4. Вывод: 13](#_Toc164603812)

# Список иллюстраций:

[Рис. 2.1. Структура проекта. 3](#_Toc164603813)

[Рис. 3.1. Создание проекта. 4](#_Toc164603814)

[Рис. 3.2. Модуль Clock\_Source. 4](#_Toc164603815)

[Рис. 3.3. Настройки On-Chip Memory. 4](#_Toc164603816)

[Рис. 3.4. Добавление процессора. 5](#_Toc164603817)

[Рис. 3.5. Отключение JTAG Debug. 5](#_Toc164603818)

[Рис. 3.6. Подключения модулей в проекте. 5](#_Toc164603819)

[Рис. 3.7. Настройка параметров в модуле nios2\_PD. 5](#_Toc164603820)

[Рис. 3.8. Модуль I/O для светодиодов. 6](#_Toc164603821)

[Рис. 3.9. Настройка подключений светодиодов к процессору. 6](#_Toc164603822)

[Рис. 3.10. Модуль I/O для переключателей. 7](#_Toc164603823)

[Рис. 3.11. Подключение SW к процессору. 7](#_Toc164603824)

[Рис. 3.12. Окно Address Map. 7](#_Toc164603825)

[Рис. 3.13. Предустановки системы. 8](#_Toc164603826)

[Рис. 3.14. Окно Messages. 8](#_Toc164603827)

[Рис. 3.15. RTL Viewer. 9](#_Toc164603828)

[Рис. 3.16. Входы-выходы в Pin Planner. 9](#_Toc164603829)

[Рис. 3.17. Создание проекта в Nios II. 9](#_Toc164603830)

[Рис. 3.18. Создание source файла. 10](#_Toc164603831)

[Рис. 3.19. Результат компиляции. 10](#_Toc164603832)

[Рис. 3.20. Измеренные настройки генерации. 11](#_Toc164603833)

[Рис. 3.21. Повторная компиляцию. 11](#_Toc164603834)

[Рис. 3.22. Изменение настроек вводов-выводов. 12](#_Toc164603835)

[Рис. 3.23. Pin Planner. 12](#_Toc164603836)

# Задание:

## Цель задания:

Введение в реализацию "системы внутри кристалла" представляет собой проект, основанный на использовании процессора NIOSII, включающий в себя следующие этапы:

* Начало работы с проектом в среде Quartus Prime (QP)
* Создание аппаратной части проекта с использованием инструмента Platform Designer (PD)
* Разработка программной части проекта в рамках среды NIOSII IDE
* Проверка функционирования проекта на платформе.

## Структура проекта:

Визуализация чисел от 0 до 255 в двоичном коде происходит на светодиодах LED1 до LED8 через процессор NIOSII. Этот процесс контролируется данными, поступающими с переключателей SW.:

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рис. 2.1. Структура проекта.

Конфигурация устройства включает в себя главный модуль и три вспомогательных: два модуля подчиненных my\_slave и один модуль my\_Dslave (стандартный подчиненный). Главный модуль получает данные через Conduit и через 8-битный интерфейс осуществляет адресное взаимодействие с одним из подчиненных, настраивая их соответственно либо записывая что-то в них. У каждого из подчиненных есть собственный Conduit, который обеспечивает возможность просмотра того, что было записано в него из главного модуля.

# Ход работы:

Выполним создание проекта со следующими настройками:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3.1. Создание проекта.

Откроем Platform Designer и выполним настройку созданного модуля clk:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рис. 3.2. Модуль Clock\_Source.

Установили синхронизацию для reset и частоту, равную частоте устройства.

Далее добавим модуль onchip\_mem, который будет хранить программу процессора. Тип памяти RAM, размер 16 Кб, остальные настройки приведены ниже:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис. 3.3. Настройки On-Chip Memory.

Добавим сам процессор:

Изображение выглядит как текст, программное обеспечение, число, Значок на компьютере

Автоматически созданное описание

Рис. 3.4. Добавление процессора.

Выключим JTAG debug т. к. на данный момент мы не будем это использовать:

Изображение выглядит как снимок экрана, текст, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 3.5. Отключение JTAG Debug.

Выполним подключения тактового сигнала, а также памяти к процессору:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3.6. Подключения модулей в проекте.

Конфигурирование процессора NIOS II требует дополнительных настроек. При включении или сбросе устройства происходит событие исключения, сигнализирующее о необходимости начать выполнение процессором с определенного адреса. Этот адрес содержит первоначальную инициализацию процессора. Поскольку сброс может быть выполнен в любой момент, необходимо учесть следующие параметры:

* Местоположение памяти для вектора сброса;
* Местоположение памяти для вектора исключений;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 3.7. Настройка параметров в модуле nios2\_PD.

Теперь добавим модуль для вывода значений на светодиоды:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3.8. Модуль I/O для светодиодов.

Подключим его к процессору:

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рис. 3.9. Настройка подключений светодиодов к процессору.

А также таким же образом подключим SW:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3.10. Модуль I/O для переключателей.

И подключим к процессору:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3.11. Подключение SW к процессору.

Выполним анализ получившейся системы.

Карта адресов приведена ниже:

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Автоматически созданное описание

Рис. 3.12. Окно Address Map.

Посмотрим на предустановки системы:

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рис. 3.13. Предустановки системы.

В окне Messages есть только 1 предупреждение, связанное с тем, что не подключён JTAG Debug модуль, однако это было сделано намеренно, поэтому на это предупреждение можем не обращать внимание:

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Автоматически созданное описание

Рис. 3.14. Окно Messages.

Подключим файлы к проекту и создадим модуль верхнего уровня:

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Выполним компиляцию посмотрим на получившийся результат в RTL Viewer:

Изображение выглядит как диаграмма, План, Параллельный, линия

Автоматически созданное описание

Рис. 3.15. RTL Viewer.

Можем увидеть, что полученная в RTL Viewer схема совпадает с той, что была задана по условию (в зелёном блоке отображается тот фрагмент системы, который был создан средствами PD).

Выполним назначение входов-выходов:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 3.16. Входы-выходы в Pin Planner.

Перейдем к созданию проекта для процессора. Создадим пустой проект в Nios II:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рис. 3.17. Создание проекта в Nios II.

Далее создадим .с файл с основным кодом проекта:

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рис. 3.18. Создание source файла.

Создадим следующий код:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Каждые пол секунды мы увеличиваем значение счетчика на 1, если sw = 1, иначе уменьшаем значение. Таким образом, используя переключатель мы можем контролировать направление счета.

Выполним сборку проекта:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис. 3.19. Результат компиляции.

Данный проект занимает 4572 байта, уменьшим объем проекта, выполним следующие настройки:

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рис. 3.20. Измеренные настройки генерации.

Выполним повторную компиляцию:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис. 3.21. Повторная компиляцию.

Как видим, проект стал занимать много меньше места.

Соберем проект для памяти и добавим в проект.

Проект был загружен на плату, светодиоды LED8-LED1 отображали последовательное увеличение счётчика от 0 до 255 при sw[0] в положении 1, а при переключении sw[0] в положение 0 счёт происходил в обратном порядке с тем же шагом от 255 до 0. Работа на стенде была продемонстрирована преподавателю.

Далее повторим то же самое, установив адреса статически в коде, а также:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

А также подключим все переключатели, для этого поправим модуль в platform designer:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3.22. Изменение настроек вводов-выводов.

Также добавим значения в Pin Planner.

Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

Рис. 3.23. Pin Planner.

Проект был загружен на плату, светодиоды LED8-LED1 отображали последовательное увеличение счётчика от 0 до 255 при sw[0] в положении 1, а при переключении sw[0] в положение 0 счёт происходил в обратном порядке с тем же шагом от 255 до 0. Работа на стенде была продемонстрирована преподавателю.

Далее добавим модуль счета по SW:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Этот код позволяет выставить по SW верхний предел счета. Устройство было продемонстрировано преподавателю.

# Вывод:

В процессе выполнения лабораторной работы мы осуществили создание и настройку системы на основе процессора NIOS II с применением программного обеспечения Quartus Prime и интегрированной среды разработки Eclipse. Реализация проекта включала разработку аппаратной составляющей с помощью Platform Designer (PD), написание программного кода в среде разработки Eclipse, где был создан файл для инициализации памяти программ процессора, а также осуществление проекта на плате.

При проверке проекта на плате светодиоды LED8-LED1 отображали пошаговое увеличение счётчика от 0 до 255 при положении sw[0] в единице, а при переключении sw[0] в ноль счёт происходил в обратном порядке с тем же шагом от 255 до 0. Это свидетельствует о правильной работе созданного устройства.

В рамках дополнительных заданий были внедрены различные функциональности, такие как изменение частоты переключения светодиодов, использование указателей для адресации данных и увеличение разрядности ввода-вывода для взаимодействия с переключателями.

Анализ результатов показал успешную работу системы на плате и правильное отображение данных на светодиодах в зависимости от положения переключателей.

Навыки работы с NIOS II, полученные в ходе выполнения этой лабораторной работы, окажутся востребованными при создании проектов в сфере встраиваемых систем, таких как системы управления, обработки сигналов, автоматизации и других.