Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе № 3**

Дисциплина: Автоматизация проектирования дискретных устройств (на английском языке)

Выполнил студент гр. 5130901/10101 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Т. Непомнящий

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Федотов

(подпись)

Санкт-Петербург

2023

**Оглавление**

[1. Задание 3](#_Toc159170090)

[2. Ход решения 3](#_Toc159170091)

[2.1. Создание модуля на языке Verilog 3](#_Toc159170092)

[2.2. Создание теста 4](#_Toc159170093)

[2.3. Моделирование проекта 5](#_Toc159170094)

[2.4. Создание модуля отладки 6](#_Toc159170095)

[2.5. Настройка SignalTap 8](#_Toc159170096)

[2.6. Отладка устройства 9](#_Toc159170097)

[2.7. Тестирование на плате 11](#_Toc159170098)

[3. Вывод 12](#_Toc159170099)

# Задание

Необходимо пройти цикл проектирования и отладки проекта в рамках пакета Quartus, включая использование таких средств отладки как:

✓ In-System Sources & Probes (ISSP)

✓ SignalTap II embedded logic analyzer

# Ход решения

## Создание модуля на языке Verilog

На языке Verilog было описано устройство, формирующее семисегментные коды.

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

Автоматически созданное описание

Рис. 1 – Код файла lab3\_1.v (1)

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

Автоматически созданное описание

Рис. 2 – Код файла lab3\_1.v (2)

Проведём Анализ и синтез проекта, чтобы выявить наличие предупреждение и ошибок:

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описание

Рис. 4 – Код файла lab3\_1.v (2)

Получили предупреждение, касающееся постоянных значений на выходах DIG. Оно будет исправлено далее, при модифицировании файла lab3\_1.v.

## Создание теста

Был разработан тест первого рода на языке Verilog, код которого представлен на рисунках 4–5 ниже:

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

Автоматически созданное описание

Рис. 4 – Код файла tb\_lab3\_1.v (1)

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

Автоматически созданное описание

Рис. 5 – Код файла tb\_lab3\_1.v (2)

## Моделирование проекта

Запустим симуляцию проекта средствами ModelSim и выведем wave:



Рис. 6 – Симуляция исходных файлов проекта в ModelSim (1)

Теперь добавим на wave элементы div\_cnt, cout и Counter и запустим симуляцию заново:

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рис. 7 – Симуляция исходных файлов проекта в ModelSim (2)

Можно заметить, что вначале сигнал cout правильный (div\_cnt имеет значение 4, в соответствии со значением параметра в файле tb\_Lab3\_1.v). Затем div\_cnt продолжает считать вместо того, чтобы начинать отсчет с 1. Это свидетельствует об ошибке в файле. Для её решения необходимо модернизировать блок кода, отвечающий за Clock Divider, чтобы добавить сброс счётчика:

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

Автоматически созданное описание

Рис. 8 – Исправление ошибки в блоке Clock Divider файла lab3\_1.v

После внесения изменений моделирование показывает, что устройство работает корректно.

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Рис. 9 – Результат моделирования после внесения исправлений

В окне Memory Data можем увидеть, что массив памяти инициализировался значениями из файла ss\_ascii.txt.

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

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рис. 10 – Значения массива tb\_mem

## Создание модуля отладки

Создадим SP\_unit с использованием IP-модуля Intel FPGA In-System Source & Probes:

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

Автоматически созданное описание

Рис. 11 – Модуль SP\_unit

Для отладки устройства необходимо создать два модуля при помощи IP-каталогов: Intel FPGA In-System Source & Probes и ALTPLL. Проверим, что IP-адреса добавлены в проект:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 12 – IP-адреса

После создания компонентов необходимо подключить их при помощи файла dl\_Lab3\_1.

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

Автоматически созданное описание

Рис. 13 – Код файла dl\_Lab3\_1

После подключения файла можем убедиться, что устройство собрано верно, открыв RTL Viewer:

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рис. 14 – RTL Viewer

Выполним полную компиляцию проекта:

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

Рис. 15 – Полная компиляция проекта

После компиляции посмотрим на временные характеристики модели:

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рис. 16 – Временные характеристики устройства

## Настройка SignalTap

Выполним предварительную настройку она SignalTap перед запуском симуляции:

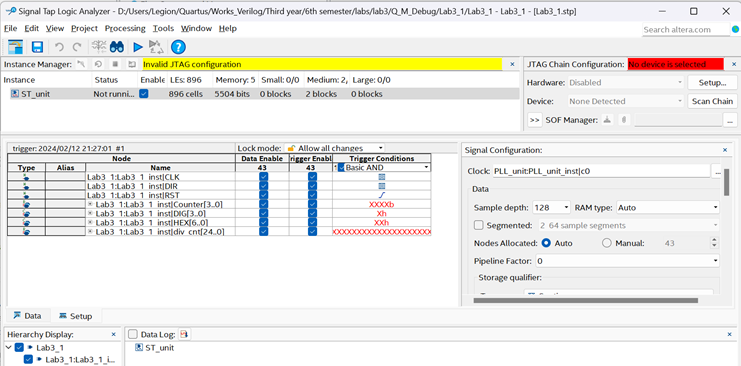


Рис. 16 – Предварительная настройка окна SignalTap

Для задания значений шины данных используем мнемоническую таблицу:

Изображение выглядит как стол

Автоматически созданное описание

Рис. 17 – Значения мнемонической таблицы

Зададим следующие настройки:

Изображение выглядит как стол

Автоматически созданное описание

Рис. 18 – рp tab

Запустим полную компиляцию и проверим временные характеристики в отчёте компиляции. Они не нарушают ограничений.

## Отладка устройства

В окне пакета Quartus выполним команду: Tools=> In System Sources and Probes Editor. После выполнения всех настроек выполним Processing => Continuously Read Probe Data.

Изображение выглядит как текст, линия, число, Шрифт

Автоматически созданное описание

Рис. 19 – Диаграмма из ISSPE

На временной диаграмме видим, что выходы HEX и DIG отображаются верно при активном уровне сигнала Reset. Изменим сигнал Reset на 1.

Изображение выглядит как текст, линия, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 20 – Диаграмма из ISSPE

Значения на выходе HEX не удовлетворяют ожидаемой последовательности значений: 3F; 06; 5B; 4F; 66; 6D; 7D; 07; 7F; 6F … Следовательно, устройство работает неправильно.

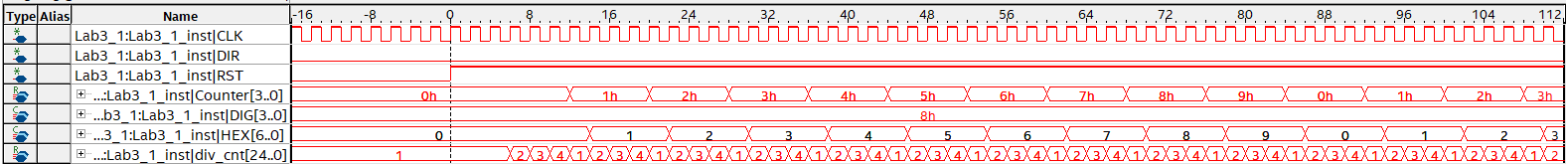


Рис. 21 – Диаграмма из Signal Tap

Из приведенной выше диаграммы видим, что делитель тактового сигнала изменяется слишком быстро. Необходимо исправить ошибку в описании модуля. Для этого необходимо задать частоту в параметр.

Выполним полную компиляцию. В отчете о компиляции видим, что устройство удовлетворяет временным ограничениям.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 22 – Временные характеристики устройства

Снова проверим временную диаграмму ISSPE:

Изображение выглядит как текст, линия, число, Шрифт

Автоматически созданное описаниеРис. 23 – Диаграмма из ISSPE

Изображение выглядит как текст, линия, Шрифт, число

Автоматически созданное описаниеРис. 24 – Диаграмма из ISSPE

Из приведенных выше диаграмм видим, что устройство работает верно во всех режимах.

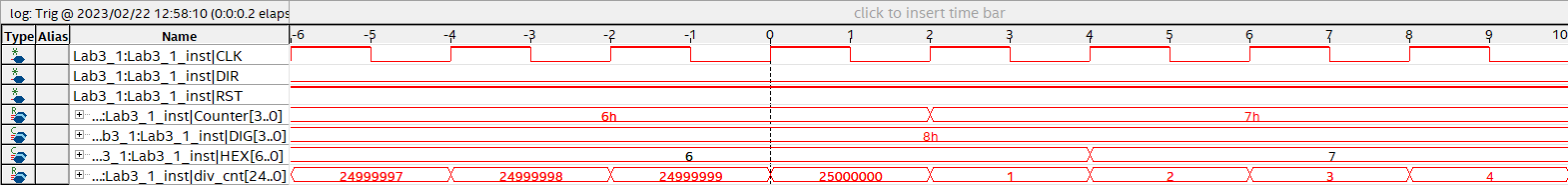
Изменим настройки Signal Tap:

Изображение выглядит как стол

Автоматически созданное описание

Рис. 25 – Вкладка Setup

Из диаграммы Signal Tap видим, что ошибка с делителем тактового сигнала была исправлена.

Рис. 26 – Диаграмма из Signal Tap

Изменим настройки Signal Tap:

Изображение выглядит как стол

Автоматически созданное описание

Рис. 27 – Настройки Signal Tap

Запустим полную компиляцию. В отчете временного анализа нарушений нет.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 28 – Временные характеристики устройства

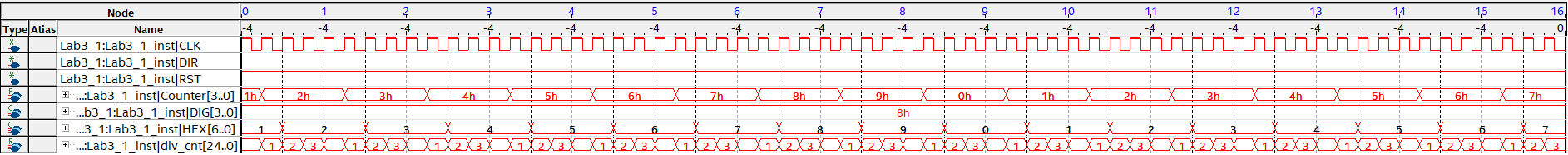


Рис. 29 – диаграмма из Signal Tap

Из приведенных выше диаграмм видно, что устройство работает верно. На шине HEX значения из мнемонической таблицы.

## Тестирование на плате

Для запуска проекта на плате необходимо добавит файл верхнего уровня impl\_Lab3\_1.v для подключения выводов тестового стенда.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 30 – Описание модуля на языке Verilog

После компиляции проверим устройство в RTL Viewer.

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

Рис. 31 – Структура устройства в RTL-viewer

Проверим также временные характеристики в отчете о компиляции.

Изображение выглядит как стол

Автоматически созданное описание

Рис. 32 – Временные характеристики устройства

Как и ожидалось, временных нарушений нет.

Далее был проведено тестирование на тестовом стенде, продемонстрировавшее, что устройство работает корректно.

# Вывод

В данной лабораторной работе был реализован полный цикл разработки и отладки проекта с использованием инструментов ISSP и SignalTap. В качестве основы проекта были применены компоненты Intel FPGA In-System Source & Probes и ALTPLL из библиотеки IP-ядер. Для облегчения моделирования устройства была задействована мнемоническая таблица.

Применение инструментов ISSP и SignalTap позволило:

* Ускорить процесс разработки за счет возможности отладки проекта на реальном FPGA без необходимости его перекомпиляции и загрузки.
* Повысить качество проектирования за счет возможности отслеживать сигналы в реальном времени и выявлять ошибки на ранних стадиях разработки.
* Снизить количество ошибок за счет возможности тестирования проекта в различных условиях.

Использование мнемонической таблицы:

* Упростило процесс моделирования устройства за счет возможности использования понятных и читаемых обозначений для сигналов.
* Снизило вероятность ошибок за счет возможности визуализировать сигналы в удобном для восприятия виде.

В результате использования данных инструментов и методов удалось значительно сократить время разработки и повысить качество проектирования.