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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Л. Симоновский (подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Федотов (подпись)

“14” февраля 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Список иллюстраций: 2](#_Toc158821551)

[2. Задача: 3](#_Toc158821552)

[3. Решение: 3](#_Toc158821553)

[4. Вывод: 9](#_Toc158821554)

# Список иллюстраций:

[Рис. 3.1. Результат тестирования моудля. 4](#_Toc158821909)

[Рис. 3.2. Результаты моделирования с внутренними сигналами. 4](#_Toc158821910)

[Рис. 3.3. Результат тестирования после исправления багов. 5](#_Toc158821911)

[Рис. 3.4. RTL-Viewer модуля отладки. 6](#_Toc158821912)

[Рис. 3.5. Настройка SignalTap II. 6](#_Toc158821913)

[Рис. 3.6. Slow 1200mV 85C Model Fmax Summary. 6](#_Toc158821914)

[Рис. 3.7. In-System Source and Probes. 7](#_Toc158821915)

[Рис. 3.8. In-System Source and Probes. Работа. DIR = 0. 7](#_Toc158821916)

[Рис. 3.9. In-System Source and Probes. Работа. DIR = 1. 7](#_Toc158821917)

[Рис. 3.10. SignalTap II. Захват при div\_cnt = 2500000. 7](#_Toc158821918)

[Рис. 3.11. Измененные настройки SignalTap II. 7](#_Toc158821919)

[Рис. 3.12. SignalTap II. Захват 16 сегментов по 8 измерений. DIR = 0. 7](#_Toc158821920)

[Рис. 3.13. SignalTap II. Захват 16 сегментов по 8 измерений. DIR = 1. 8](#_Toc158821921)

[Рис. 3.14. Модуль реализующий проект. 8](#_Toc158821922)

[Рис. 3.15. RTL Viewer итогового проекта. 8](#_Toc158821923)

[Рис. 3.16. Timing Analyzer report. 9](#_Toc158821924)

# Задача:

Изучить:

* Как создавать устройства в Quartus Prime.
* Как моделировать созданное устройство, используя модели.
* Как отладить созданное устройство используя ISSP и SignalTap II.
* Как реализовывать созданные устройства на плате.

# Решение:

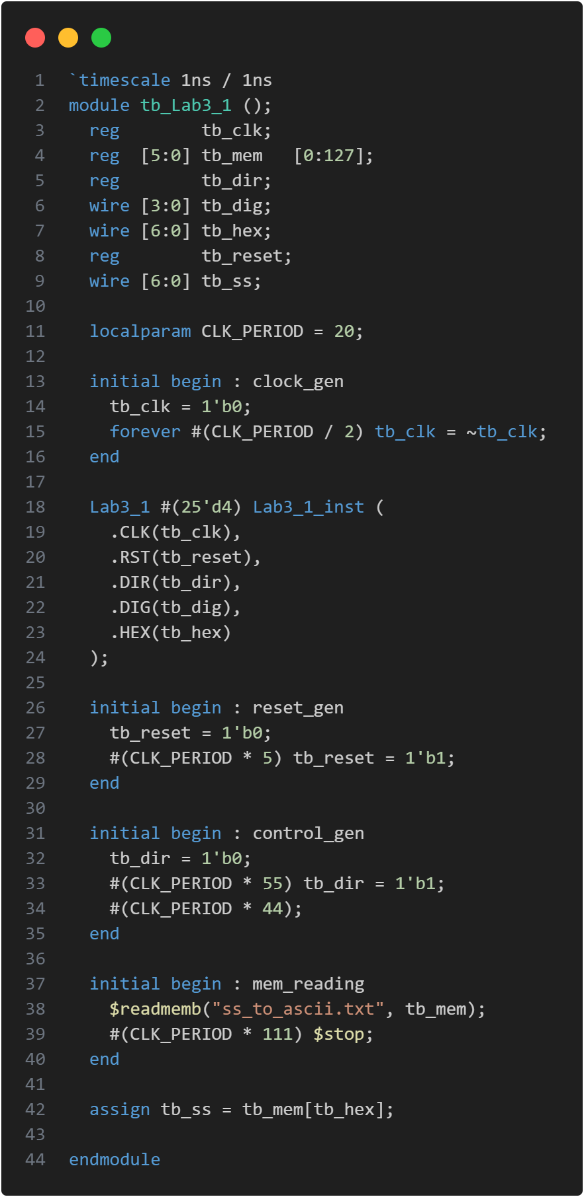
Возьмем код устройства из приложения к лабораторной:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Данное устройство реализует счетчик от 0 до 9 на семисегментном индикаторе.

Протестируем это устройство тестом первого класса:



Как можно заметить, сначала выполняется счет вверх, потом вниз. При этом первые 5 периодов CLK подается сигнал сброса. Для удобства добавлен модуль, который переводит из семисегментного кода в HEX.

Выполним компиляцию и посмотрим на результат запуска этого тестового модуля:

Изображение выглядит как снимок экрана, электроника, линия, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 3.1. Результат тестирования моудля.

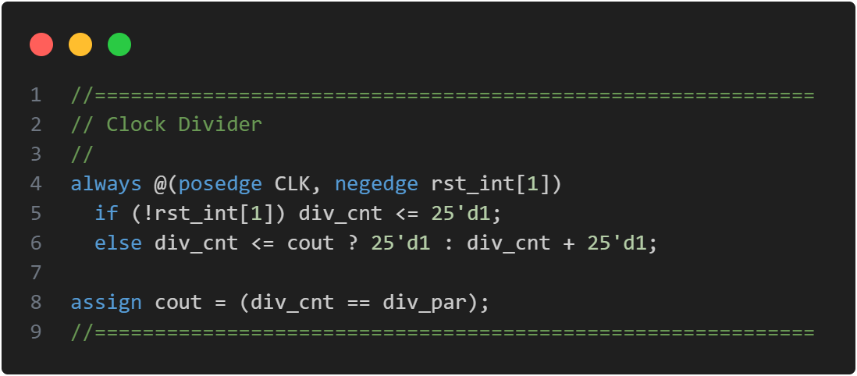
Как мы видим, полученный результат отличается от ожидаемого. Добавим внутренние сигналы тестируемого модуля и найдем ошибку:

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

Автоматически созданное описание

Рис. 3.2. Результаты моделирования с внутренними сигналами.

Как видно на этом скриншоте, счетчик-делитель вместо того, чтоб сброситься продолжает считать, поэтому в дальнейшем числа не меняются. Поправим эту ошибку, добавив в модуль счетчика сброс:



Повторим тестирование модуля:

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

Автоматически созданное описание

Рис. 3.3. Результат тестирования после исправления багов.

Как видим исправление этого модуля помогло и устройство работает корректно.

После моделирования устройство необходимо выполнить тестирование на плате. Разработаем для этого модуль, который позволит управлять всеми входами устройства с компьютера, а также отслеживать выходы:

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

Автоматически созданное описание

Используя ISSP мы будем управлять входами направления счета и сброса, а отслеживать значение, передаваемое на счетчик.

Также создаем PLL, который выдает частоту в 2 раза больше заданной, чтоб используя SignalTap II видеть тактовый сигнал, однако пока SignalTap II не настроен, db\_clk\_high исчезнет при компиляции:

Изображение выглядит как текст, диаграмма, План, снимок экрана

Автоматически созданное описание

Рис. 3.4. RTL-Viewer модуля отладки.

Выполним настройку SignalTap II следующим образом:

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

Автоматически созданное описание

Рис. 3.5. Настройка SignalTap II.

Добавим к проекту SDC файл и выполним полную компиляцию. Получим следующий Fmax для разработанного устройства:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 3.6. Slow 1200mV 85C Model Fmax Summary.

Здесь важно, чтоб clock\_in работал со скоростью минимум 25 МГц (частота платы). Как мы видим, это действительно выполняется.

Запишем разработанное устройство на плату для дальнейшей отладки.

Выполним настройку In-System Source and Probes:

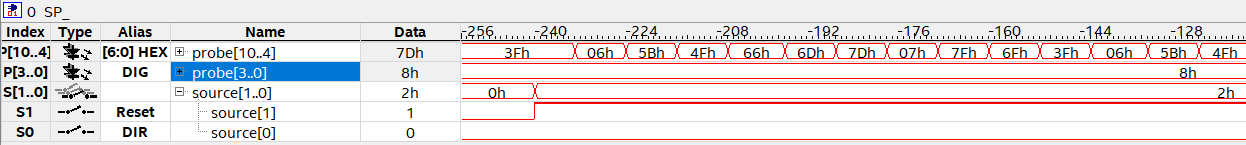


Рис. 3.7. In-System Source and Probes.

Запустим его, выставим Reset в 1 (устройство начнет работу) и посмотрим на результат:

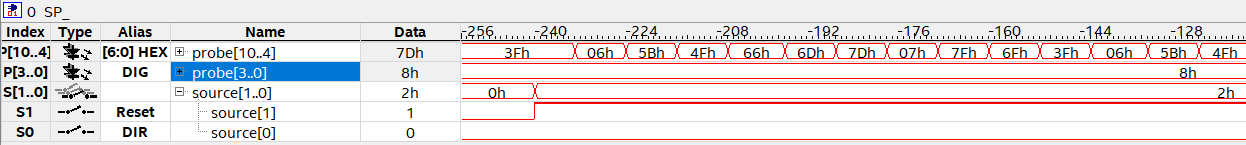


Рис. 3.8. In-System Source and Probes. Работа. DIR = 0.

Как мы видим, устройство корректно начало считать вверх, как и было при тестировании.

Изменим направление счета и посмотрим, что устройство корректно считает и в обратную сторону:

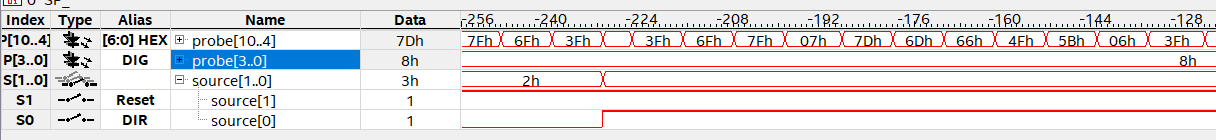


Рис. 3.9. In-System Source and Probes. Работа. DIR = 1.

Перейдем в SignalTap II и настроим таким образом, чтоб сигнал захватывался при div\_cnt = 2500000 т.е. параметр, переданный в модуль ранее (момент переключения на следующее значение). После этого выполним запуск и проучим следующий результат:

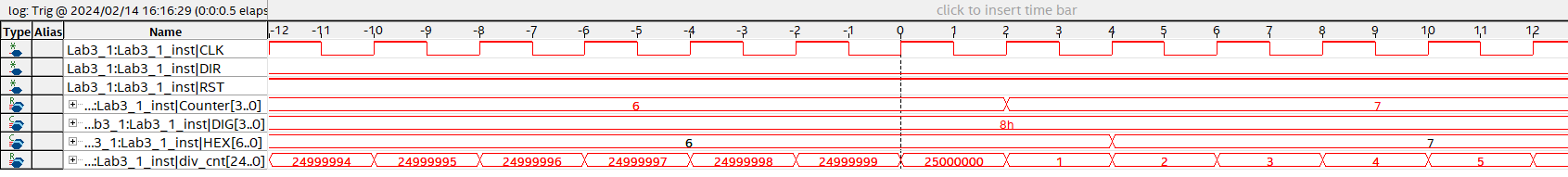


Рис. 3.10. SignalTap II. Захват при div\_cnt = 2500000.

Мы захватили момент перехода к следующей цифре, что можно заметить на скриншоте. Так же важно отметить, что благодаря PLL мы корректно наблюдаем сигнал CLK.

Теперь изменим настройки SignalTap II следующим образом:

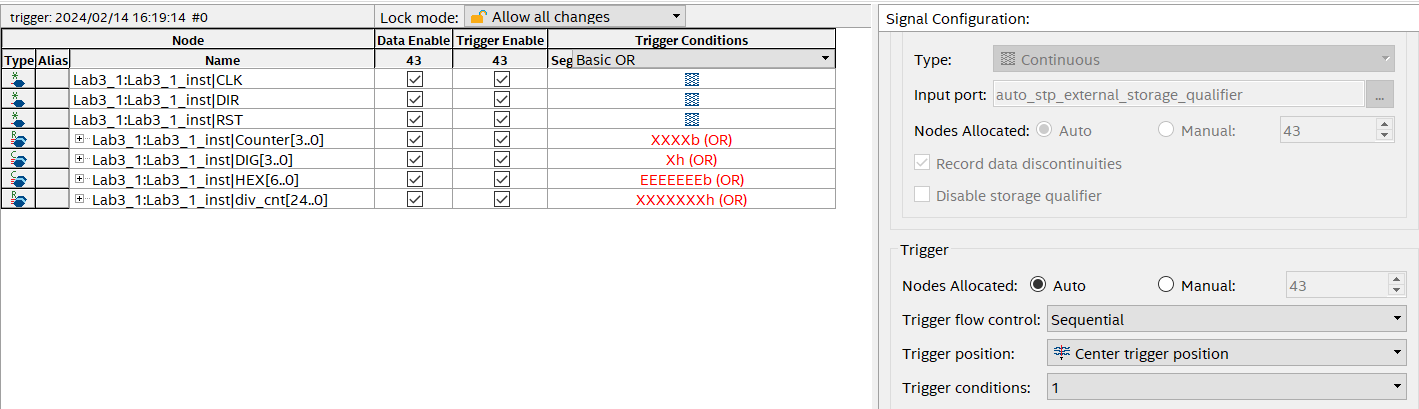


Рис. 3.11. Измененные настройки SignalTap II.

Они позволят нам захватывать именно изменения HEX выхода и так 16 раз, в каждом по 8 измерений, это удобно позволит отслеживать изменения на этом выходе.

Поставим Reset = 1, DIR = 0 и запустим захват данных на SignalTap II:

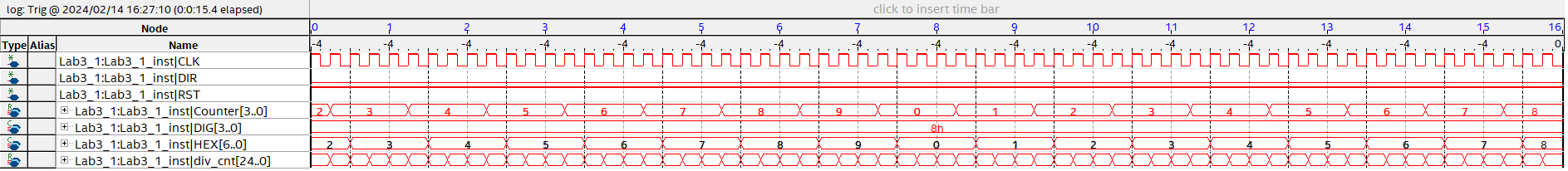


Рис. 3.12. SignalTap II. Захват 16 сегментов по 8 измерений. DIR = 0.

Как и ожидалось мы видим как значение растет на отладке.

Повторим измерения, установив DIR = 1:

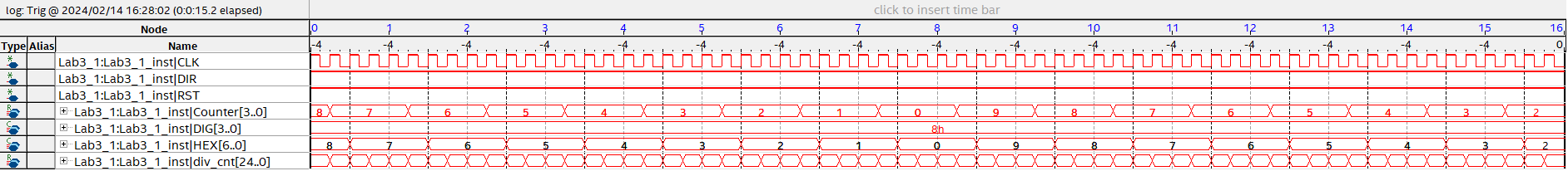


Рис. 3.13. SignalTap II. Захват 16 сегментов по 8 измерений. DIR = 1.

А теперь мы получаем направление счета вниз.

Теперь создадим модуль, который будет полноценно реализовывать модуль на плате:

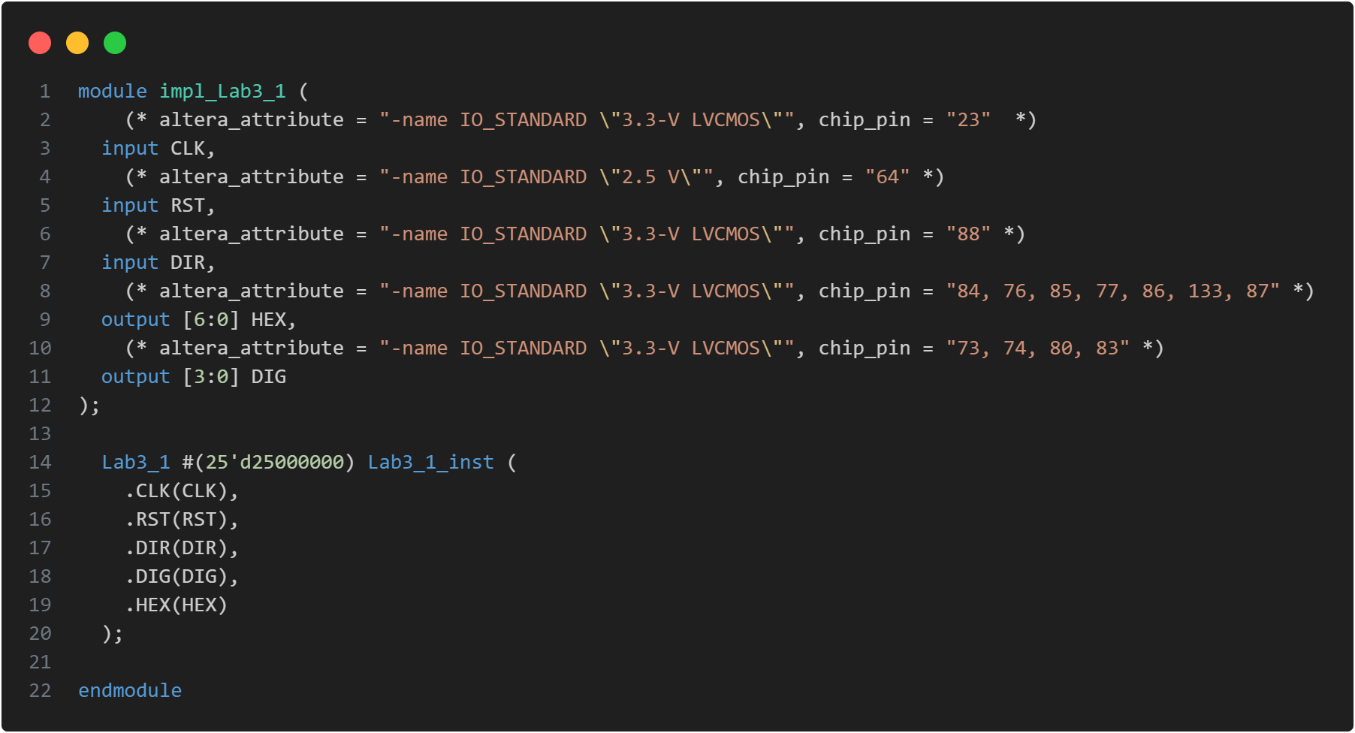


Рис. 3.14. Модуль реализующий проект.

Его RTL схема приведена ниже:

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рис. 3.15. RTL Viewer итогового проекта.

Как по нему видно, этот файл действительно только задает входы и выходы созданному модулю.

Также добавим SDC файл для временных требований, включим SignalTap II и запустим полную компиляцию.

Используя Timing Analyzer, убедимся, что все временные требования выполняются, а также что SignalTap II действительно выключился:

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

Автоматически созданное описание

Рис. 3.16. Timing Analyzer report.

Запишем полученный проект на плату. Данное устройство работает корректно и было продемонстрированно преподавателю.

# Вывод:

В ходе лабораторной работы были получены навыки по многоэтапному тестированию устройства, сначала используя моделирование (без платы) средствами ModelSim, после чего тестирование уже на плате, используя In-System Sources and Probes Editor и SignalTap II и последующую реализацию модуля, готового к полноценному использованию на плате.

Данные навыки помогут при разработке как маленьких проектов, так и больших, которые не так просто отлаживать. ModelSim, In-System Sources and Probes Editor и SignalTap II очень сильно ускоряют отладку, что несомненно важно при работе с любым проектом.