Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе № 7**

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

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

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

“23” марта 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Список иллюстраций: 2](#_Toc161238569)

[2. Цель упражнения: 3](#_Toc161238570)

[3. Алгоритм работы проекта: 3](#_Toc161238571)

[4. Решение: 4](#_Toc161238572)

[5. Вывод: 8](#_Toc161238573)

# Список иллюстраций:

[Рис. 1. Структура разрабатываемого устройства. 3](#_Toc162094218)

[Рис. 2. Переходы конечного автомата. 4](#_Toc162094219)

[Рис. 3. RTL Viewer устройства. 5](#_Toc162094220)

[Рис. 4. Результат тестирования. 6](#_Toc162094221)

[Рис. 5. Настройки Signal Tap II. 6](#_Toc162094222)

[Рис. 6. Signal Tap II после запуска. 6](#_Toc162094223)

[Рис. 7. RTL Viewer модуля с измененным интерфейсом. 7](#_Toc162094224)

[Рис. 8. Signal Tap II. Обновленные настройки. 8](#_Toc162094225)

[Рис. 9. Signal Tap II. 8](#_Toc162094226)

# Цель упражнения:

Пройти цикл проектирования в рамках пакетов Quartus и ModelSim, включая следующие этапы:

* Создание проекта.
* Разработка описания модулей с использованием конструкций расширения SystemVerilog.
* Разработка теста на языке SystemVerilog и моделирование.
* Отладка проекта.

# Алгоритм работы проекта:

Структура разрабатываемого устройства приведена на рисунке ниже.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рис. 1. Структура разрабатываемого устройства.

В состав устройства входят:

* Модуль master – ведущее устройство, формирует обращение к двум ведомым устройствам.
* Модули slave\_EVEN и slave\_ODD – ведомые устройства, управляемые мастером.
* Модуль bus\_MM - экземпляр интерфейса, обеспечивающий подключение мастера и ведомых устройств.

Выводы устройства (выделены зеленым цветом):

* CLK – вход тактового сигнала.
* reset – синхронный сброс всех устройств.
* D\_odd – восьмиразрядный выход.
* D\_even – восьмиразрядный выход.

Алгоритм работы разрабатываемого устройства определяется алгоритмами работы его модулей:

* Модуль master:
  + содержит КА Мура:
    - 3 состояния (граф переходов приведен ниже) - все переходы между состояниями безусловные:
      * начальное – initSM.
      * пустое – nop.
      * записи данных -wr1D.
    - формирует сигналы
      * address - Адреса (8 бит); комбинационный выход.
      * writedata - Данных (8 бит); комбинационный выход;
      * write - Разрешения записи; комбинационный выход;
  + содержит счетчик cnt (счетчик 8-ми разрядный), его значение используется для формирования адреса (address) и данных (writedata).

Изображение выглядит как текст, круг, Шрифт, линия

Автоматически созданное описание

Рис. 2. Переходы конечного автомата.

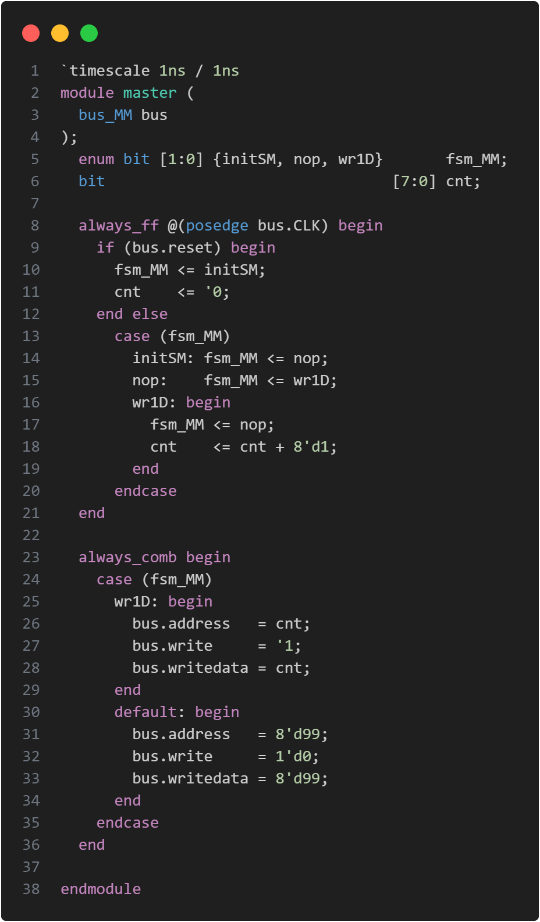
* Модуль slave\_EVEN - «ведомый\_четный»:
  + анализирует address и сигнал write и если: address четный и write = 1, то записывает в свой внутренний регистр значение шины данных (writedata);
  + на выходе – значение внутреннего регистра;
* Модуль slave\_ODD - «ведомый\_нечетный»:
  + анализирует address и сигнал write и если: address нечетный и write = 1, то записывает в свой внутренний регистр значение шины данных (writedata);
  + на выходе – значение внутреннего регистра;

# Решение:

Выполним описание интерфейса, для разрабатываемых модулей:

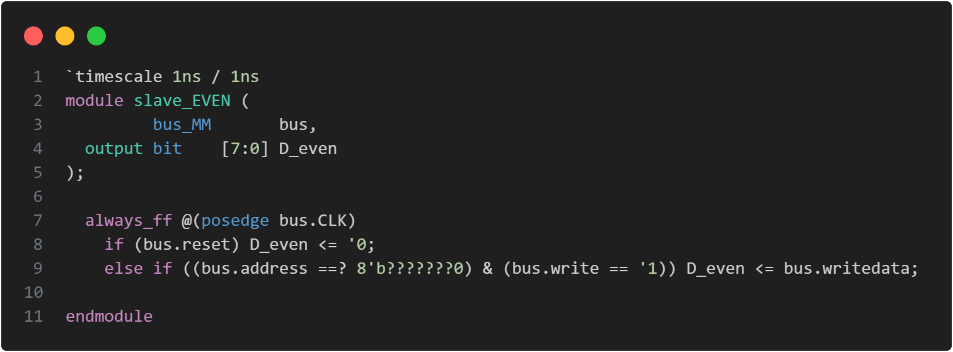


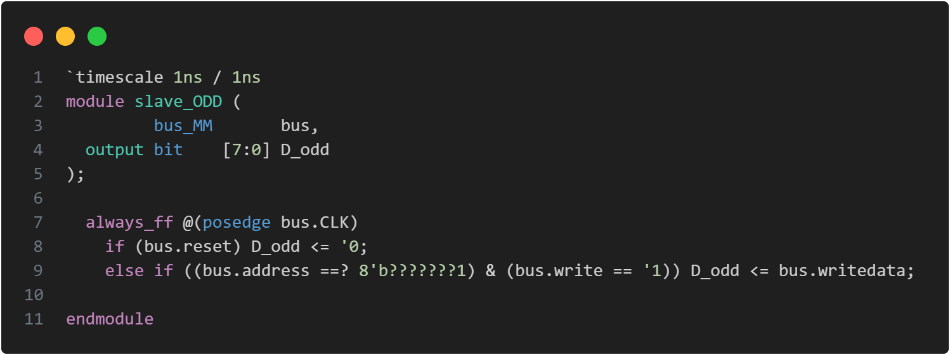
Используя интерфейс, создадим описание «мастера»:



В данном модуле реализован конечный автомат Мура в соответствии с заданием.

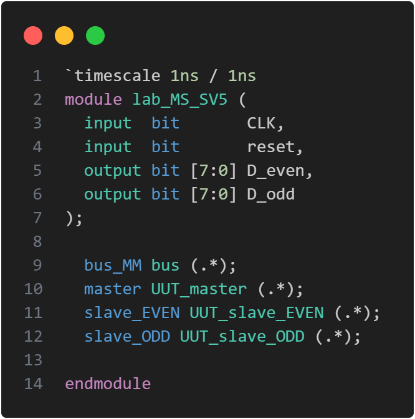
Теперь разработаем модули slave\_ODD и slave\_EVEN, используя разработанный интерфейс:





Как мы видим запись в EVEN происходит по сигналу write и адресу, оканчивающемуся на 0, а в ODD, наоборот, адрес должен кончаться на 1.

Теперь разработаем описание верхнего уровня:



RTL Viewer разработанной схемы выглядит следующим образом:

Изображение выглядит как диаграмма, текст, линия, снимок экрана

Автоматически созданное описание

Рис. 3. RTL Viewer устройства.

Разработаем тест первого класса для устройства:

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

Автоматически созданное описание

Запустим тест и посмотрим, корректно ли работает разработанное устройство:

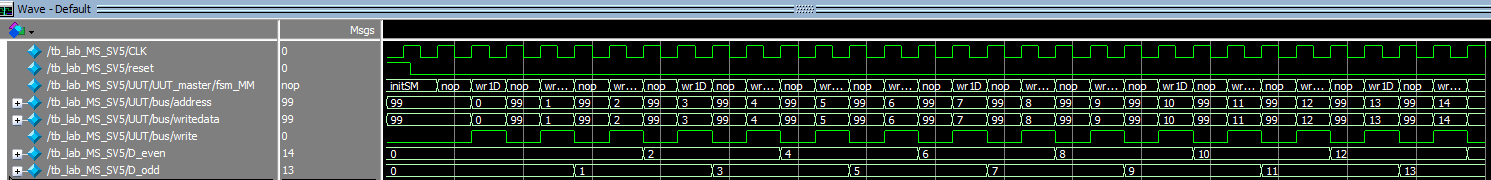
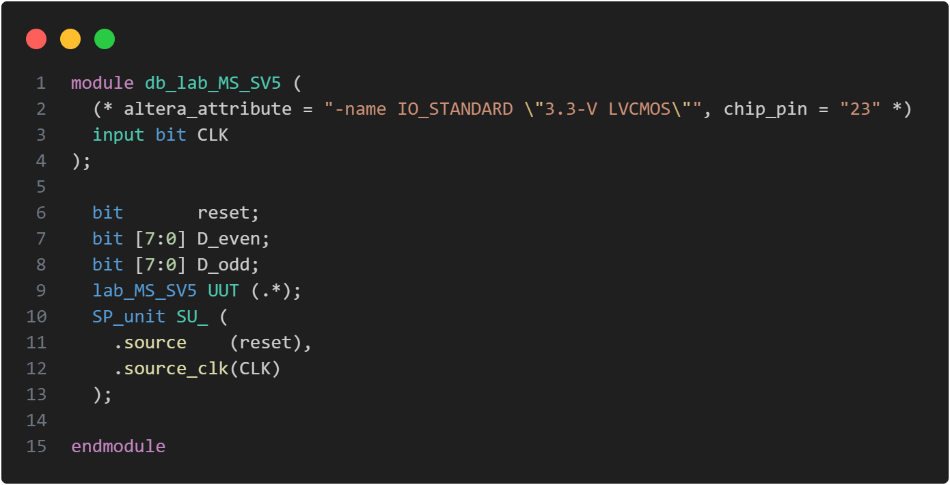


Рис. 4. Результат тестирования.

Как мы можем заметить, все работает корректно.

Теперь разработаем модуль для отладки устройства на плате:



Используя SP, будем подавать reset, а данные снимать будем с использованием SignalTap II:

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рис. 5. Настройки Signal Tap II.

Запишем модуль на плату, включим Signal Tap II и переключим reset в 0. Тогда получим следующий результат:

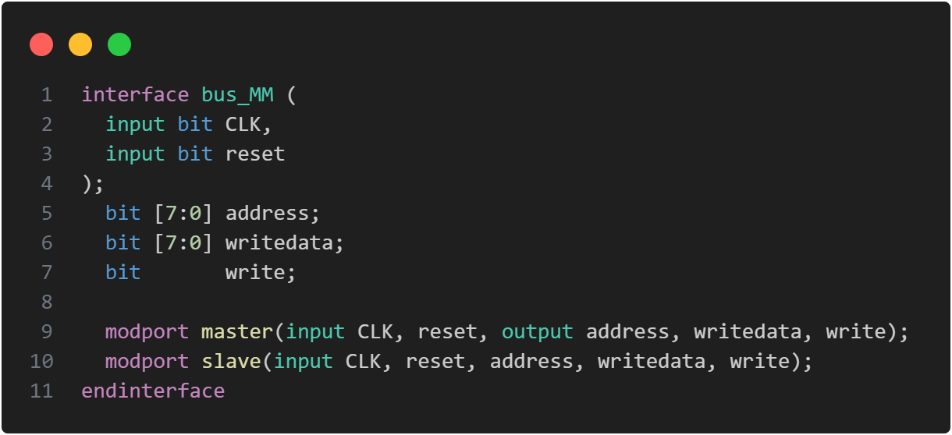
Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рис. 6. Signal Tap II после запуска.

Как мы видим, после установки reset в 0 мы перешли из стартового состояния в цикл из состояний nop и wr1D. D\_EVEN и D\_ODD меняются в соответствии с заданием, что свидетельствует о корректности разработанного устройства.

Теперь немного изменим интерфейс:



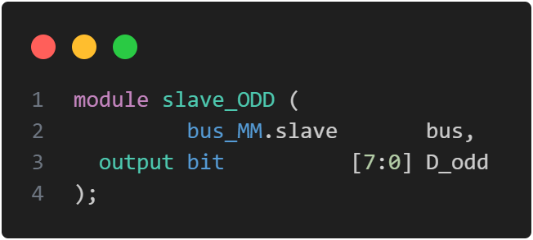
Вместо того, чтоб оставлять входы и выходы inout явно зададим что есть что. Из-за этого чуть-чуть изменится объявления интерфейсов в различных модулях:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание



Более ничего не поменялось в коде. Выполним компиляцию и посмотрим на RTL Viewer:

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

Рис. 7. RTL Viewer модуля с измененным интерфейсом.

Как мы видим, появились дополнительные элементы, они позволяют настроить направление данных, то, что мы и задавали в интерфейсе.

Теперь чуть изменим настройки для SignalTap II:

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рис. 8. Signal Tap II. Обновленные настройки.

Запишем на плату разработанный ранее отладочный модуль и посмотрим на результат в Signal Tap II:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис. 9. Signal Tap II.

Как мы видим, мы зафиксировали значение 23, что и требовалось по заданию.

# Вывод:

В ходе лабораторной работы успешно пройден цикл проектирования, начиная с создания проекта и разработки модулей с использованием расширений SystemVerilog. Использование SystemVerilog предоставило широкий спектр новых возможностей по сравнению с Verilog, облегчая процесс разработки и улучшая читаемость кода.

Отладка проекта осуществлялась с помощью инструментов In-System Sources and Probes Editor и SignalTap II, что значительно повысило эффективность процесса. Эти инструменты позволили быстро выявить и исправить ошибки, что является ключевым аспектом при работе с любым проектом.