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)

“21” февраля 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Список иллюстраций: 2](#_Toc159392787)

[2. Цель упражнения: 3](#_Toc159392788)

[3. Алгоритм работы проекта: 3](#_Toc159392789)

[4. Решение: 3](#_Toc159392790)

[5. Вывод: 11](#_Toc159392791)

# Список иллюстраций:

[Рис. 3.1. Структура разрабатываемого проекта. 3](#_Toc159392792)

[Рис. 4.1. RTL Viewer. 5](#_Toc159392793)

[Рис. 4.2. Результат моделирования. 6](#_Toc159392794)

[Рис. 4.3. Настройки ISSP. 7](#_Toc159392795)

[Рис. 4.4. Настройки Signal Tap II. 8](#_Toc159392796)

[Рис. 4.5. Значения в ISSP. 8](#_Toc159392797)

[Рис. 4.6. Значения в ISSP. 8](#_Toc159392798)

[Рис. 4.7. Значения в ISSP. 9](#_Toc159392799)

[Рис. 4.8. Значения в ISSP. 9](#_Toc159392800)

[Рис. 4.9. Результат измерений в Signal Tap II. 9](#_Toc159392801)

[Рис. 4.10. Новые настройки Signal Tap II. 10](#_Toc159392802)

[Рис. 4.11. Значения в ISSP. 10](#_Toc159392803)

[Рис. 4.12. Значения в ISSP. 10](#_Toc159392804)

[Рис. 4.13. Результат измерений в Signal Tap II. 11](#_Toc159392805)

# 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:

* Два экземпляра (cntA и cntB) 10-разрядного счетчика (cnt\_10bits) на сложение, по модулю, заданному на входах cntA\_Module и cntB\_Module, формируют:
  + текущие значения: cntA[9:0], cntB[9:0]
  + сигналы переноса (по достижению заданного модуля счета): CoutA, CoutB
* Компаратор (cpm\_eq) на равенство сравнивает значения двух счетчиков и формирует выходной сигнал cntA\_EQ\_cntB.
* Конечный автомат (fsm) анализирует сигналы переноса двух счетчиков и формирует сигналы:
  + AeqB – равное количество сигналов переноса от счетчика А и счетчика В.
  + AmB – сигналы переноса от счетчика А появляются чаще.
  + BmA – сигналы переноса от счетчика В появляются чаще.
* Сигнал асинхронного сброса (rst\_n) обеспечивает сброс всех устройств с памятью.

Структура проекта приведена на рисунке ниже:

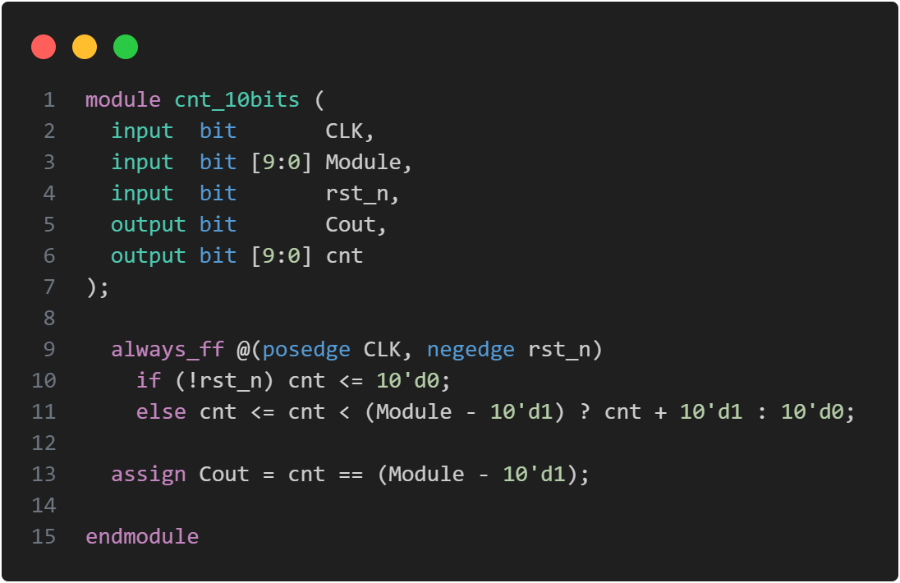
Изображение выглядит как текст, диаграмма, снимок экрана, План

Автоматически созданное описание

Рис. 3.1. Структура разрабатываемого проекта.

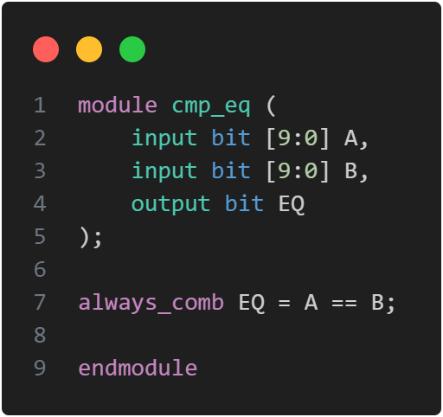
# Решение:

Выполним создание первого модуля – cnt\_10bits, используя средства System Verilog:

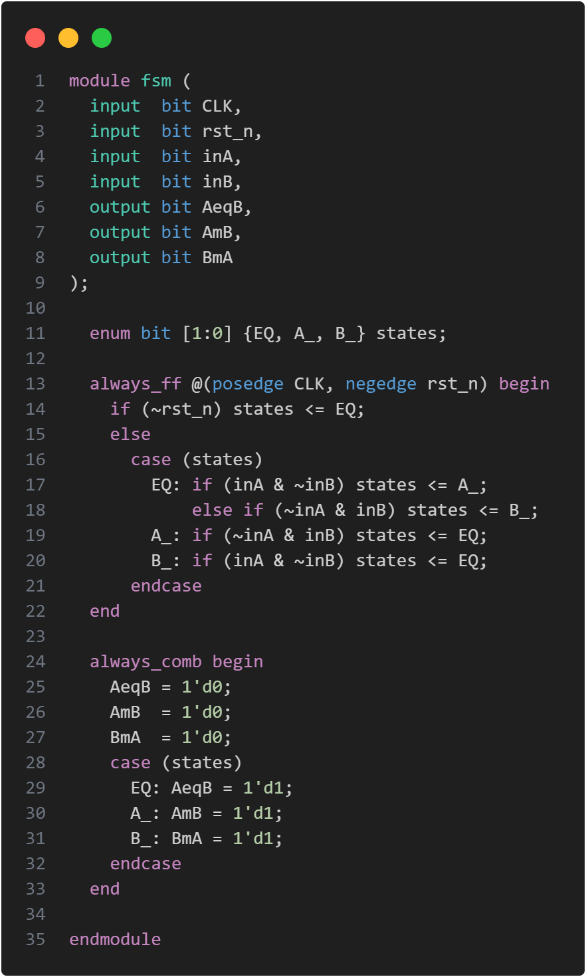


Это счетчик делитель с делителем, подаваемым на вход Module.

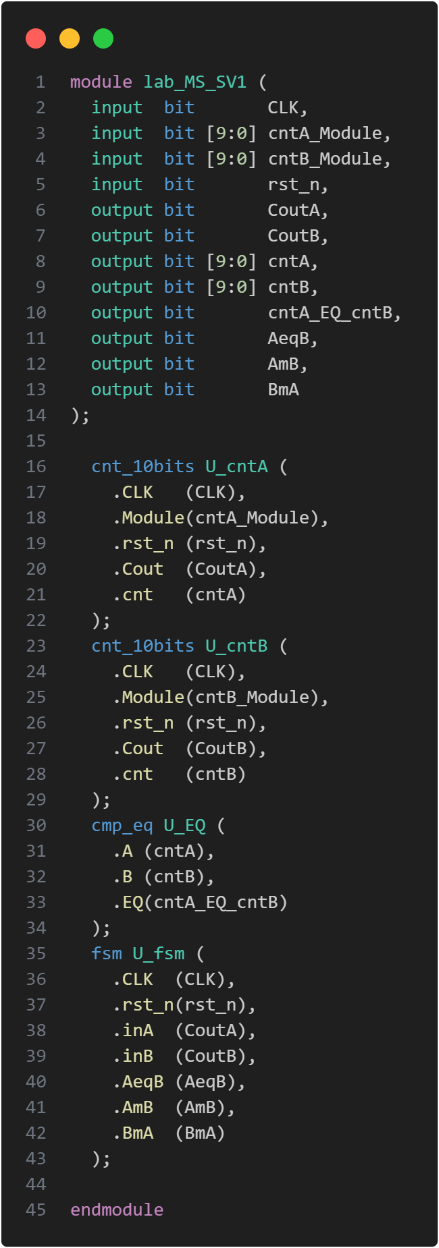
Далее реализуем модуль cmp\_eq, как комбинационную схему средствами System Verilog:



И последний модуль, необходимый для реализации – конечный автомат для фиксации, какие сигналы переноса появляются чаще – со счетчика A или B:



Теперь создадим модуль верхнего уровня, собрав все модули вместе:



Выполним полную компиляцию и проверим, что получившаяся схема соответствует поставленному заданию:

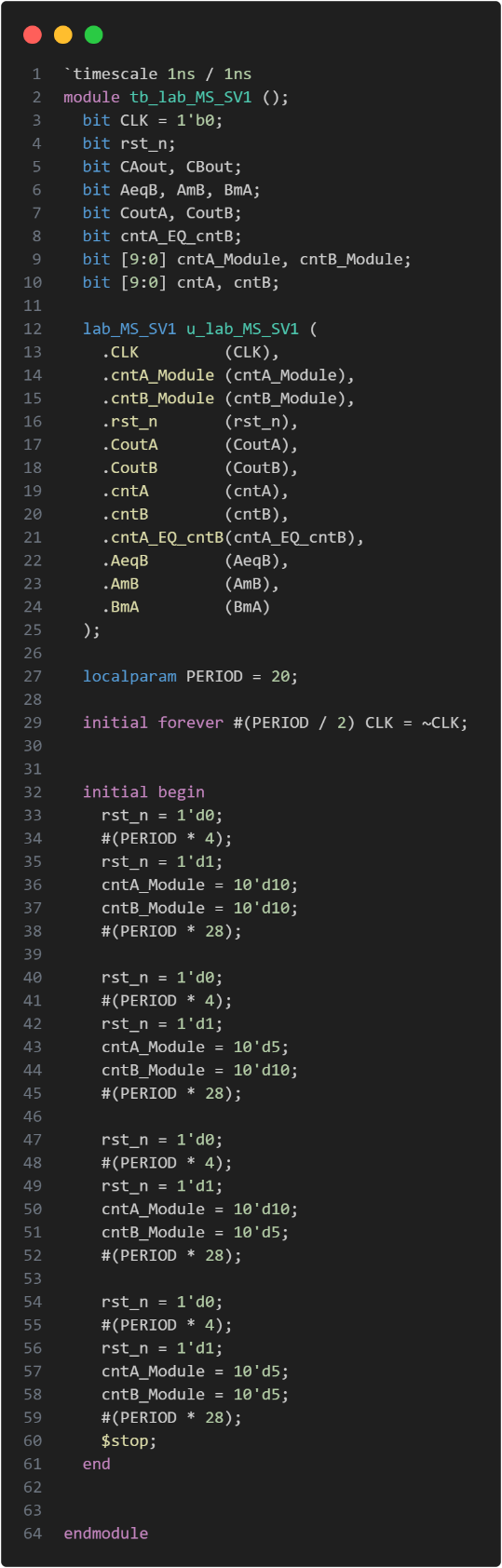
Изображение выглядит как текст, диаграмма, снимок экрана, План

Автоматически созданное описание

Рис. 4.1. RTL Viewer.

Как мы видим, получившаяся схема полностью идентична поставленной задаче, далее необходимо выполнить тестирование.

Разработаем тест первого уровня:



Выполним запуск этого теста, средствами Quartus и получим следующий результат:

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

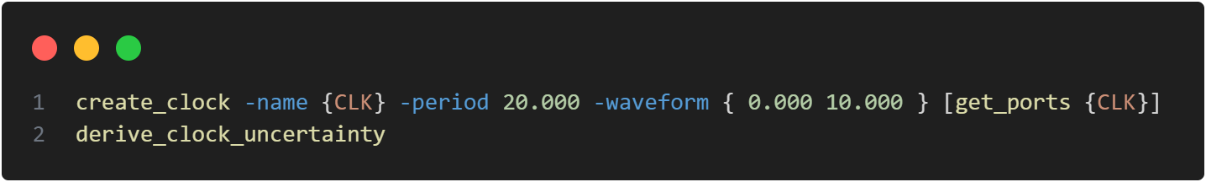
Автоматически созданное описание

Рис. 4.2. Результат моделирования.

Видим, что результат моделирования полностью соответствует ожиданиям, тогда перейдем к моделированию на плате, для этого создадим следующий модуль верхнего уровня:



Добавим SDC файл, чтоб задать требования тактовому сигналу, со следующим содержанием:



Для отладки на устройстве будем использовать ISSP, с помощью которого будем подавать сигнал сброса и значения для счетчиков и Signal Tap II, которым будем отслеживать результаты.

ISSP будет иметь следующие настройки:



Рис. 4.3. Настройки ISSP.

А настройки Signal Tap II выглядят следующим образом:

Изображение выглядит как текст, число, Шрифт, программное обеспечение

Автоматически созданное описание

Рис. 4.4. Настройки Signal Tap II.

Как можно заметить, захват происходит по фронту сигнала сброса. Используя ISSP, запишем на устройство следующие значения:



Рис. 4.5. Значения в ISSP.

После чего включим счетчики, подав 1 на rst\_n.

Теперь изменим значения в ISSP:

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.6. Значения в ISSP.

После этого опять подадим 1 на rst\_n, тем самым включим счетчики и чтоб Signal Tap II выполнил захват данных.

Опять поменяем значения в ISSP:

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.7. Значения в ISSP.

Опять переключаем значение rst\_n на 1, после чего повторяем изменение данных в ISSP:

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.8. Значения в ISSP.

После чего в 4 раз переключаем rst\_n на 1, тем самым получив 4 сегмента на Signal Tap II:

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.9. Результат измерений в Signal Tap II.

Как можно заметить, Рис. 4.2 и Рис. 4.9 полностью совпадают, что свидетельствует о корректности работы устройства на ПЛИС.

После этого поменяем настройки Signal Tap II следующим образом:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 4.10. Новые настройки Signal Tap II.

Таким образом будет выполняться захват 2 сегментов, вместо 4, как было ранее.

Запишем этот проект на плату и используя ISSP подадим следующие значения (в соответствии с вариантом 20 и 40):

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.11. Значения в ISSP.

Включаем устройство, переключением rst\_n в единицу, после чего опять меняем значения в ISSP:

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.12. Значения в ISSP.

На Signal Tap II получаем следующую диаграмму:

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.13. Результат измерений в Signal Tap II.

Как мы видим, она полностью соответствует ожиданиям. Сначала при подаче комбинации 40; 20 мы получаем B, а после изменения их местами, получаем A.

# 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.