Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе № 2\_2**

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

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

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

“07” февраля 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Список иллюстраций: 2](#_Toc158119264)

[2. Задача: 3](#_Toc158119265)

[3. Решение: 3](#_Toc158119266)

[4. Вывод: 6](#_Toc158119267)

# Список иллюстраций:

[Рис. 2.1. Схема разрабатываемого устройства. 3](#_Toc158119170)

[Рис. 3.1. Код модуля верхнего уровня lab2\_2. 4](#_Toc158119171)

[Рис. 3.2. RTL Viewer модуля lab2\_2. 4](#_Toc158119172)

[Рис. 3.3. Код теста первого класса для модуля lab2\_2. 5](#_Toc158119173)

[Рис. 3.4. .do файл для запуска тестового модуля. 5](#_Toc158119174)

[Рис. 3.5. Результат моделирования. 5](#_Toc158119175)

# Задача:

На языке Verilog разработать устройство по следующей схеме:

Изображение выглядит как текст, диаграмма, Шрифт, План

Автоматически созданное описание

Рис. 2.1. Схема разрабатываемого устройства.

**Выводы:**

1. **Входы:**
   1. CLK – тактовый сигнал.
   2. aRSTin – вход асинхронного сброса (активный уровень для сброса – 1).
2. **Выходы:**
   1. [15:0] Dout – выход.

**Модули:**

1. **CNT** – счетчик, создаваемый с помощью IP модуля LPM\_COUNTER:
   1. Разрядность: 8 бит.
   2. Двоичный счетчик на сложение.
   3. Вход асинхронного сброса (clear).
2. **PWR** – модуль возведения в степень 2, создаваемый с помощью IP модуля LPM\_MULT:
   1. Два входа по 8 бит.
   2. Без знаковый.
   3. Без конвейеризации.
3. **RG** – регистр, описываемый на Verilog в файле верхнего уровня:
   1. arst – вход асинхронного сброса (активный уровень – 1).
4. **DFF** – триггеры, описываемые на Verilog в файле верхнего уровня:
   1. aset – вход асинхронно устанавливает триггер в 1.

# Решение:

Создадим модуль верхнего уровня на языке Verilog. Его код будет выглядеть следующим образом:

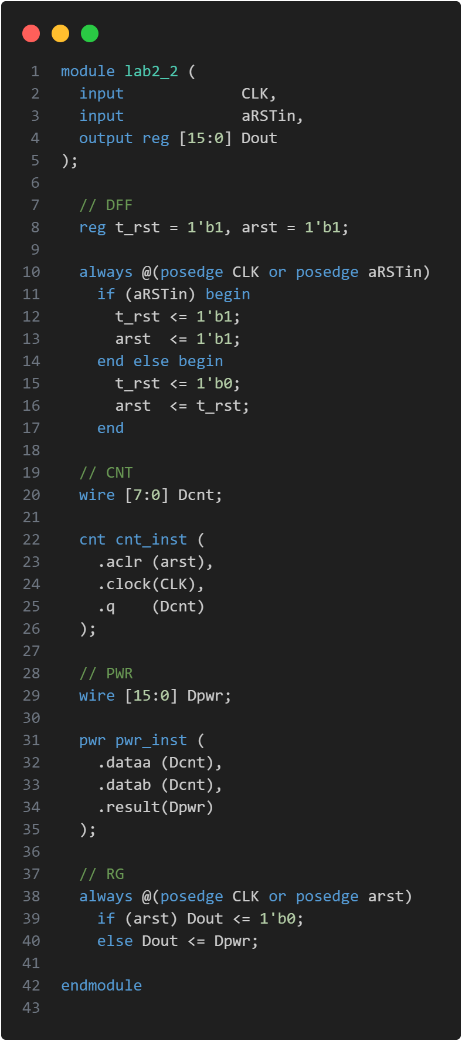


Рис. 3.1. Код модуля верхнего уровня lab2\_2.

Первый блок always описывает два триггера DFF для сигнала асинхронного сброса. Поскольку сброс асинхронный, always реагирует не только на фронт clk, но и на фронт сигнала сброса.

Блок CNT передает параметры IP-модулю счетчика, полученного из IP-модулей Quartus Prime.

Блок PWR, аналогично CNT передает модули IP-модулю умножителя.

Блок RG реализует регистр на выходе умножителя. Поскольку регистр должен иметь асинхронный сигнал сброса, поэтому always реагирует не только на фронт clk, но и на фронт сигнала сброса.

Проверим корректность разработанной схемы, используя RTL Viewer:

Изображение выглядит как диаграмма, линия, текст, График

Автоматически созданное описание

Рис. 3.2. RTL Viewer модуля lab2\_2.

Как мы видим, разработанное устроство полностью совпадает с примером.

Далее разработаем тесты первого класса для разработанного модуля:

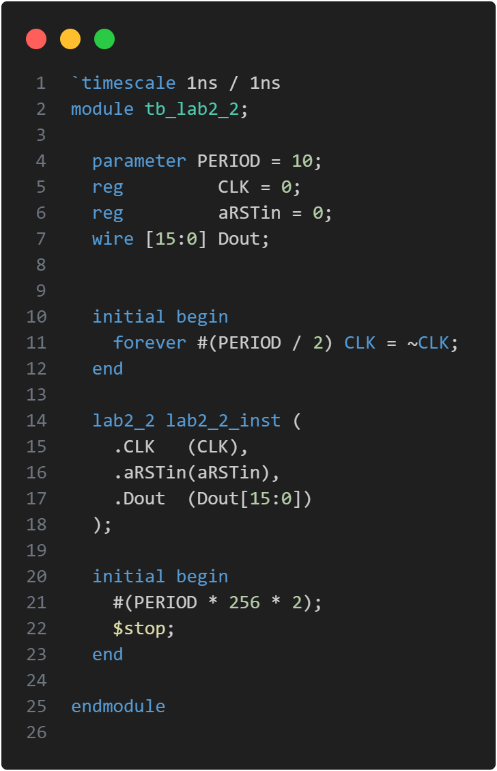


Рис. 3.3. Код теста первого класса для модуля lab2\_2.

Поскольку на один полный цикл счетчика (от 0 до 255 (т.к. раззрядность счетчика 8)) необходимо 256 тактов, для двух полных циклов необходимо времени. Именно столько мы и ждем перед остановкой.

Для запуска тестового модуля созададим следующий .do файл:

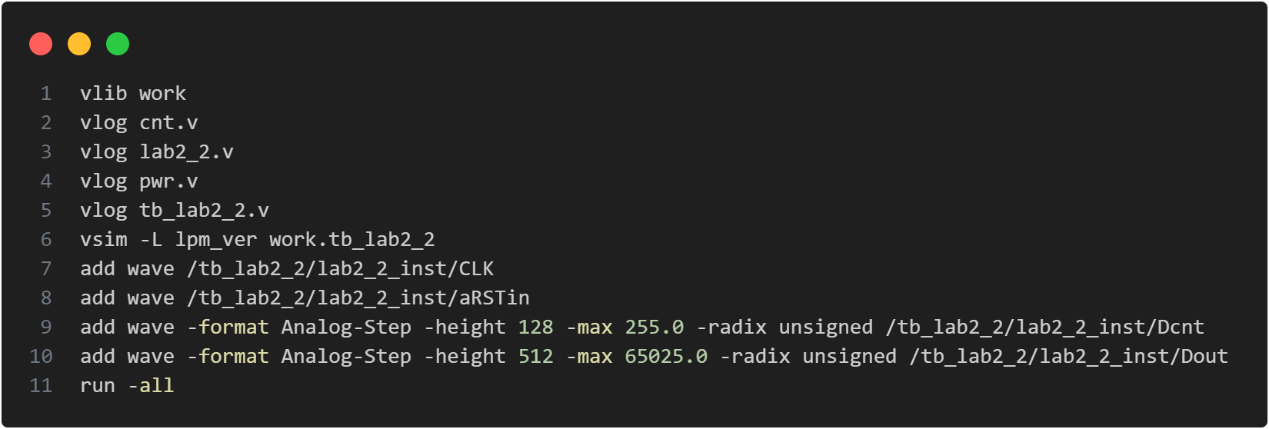


Рис. 3.4. .do файл для запуска тестового модуля.

В строках с 1 по 5 мы создаем проект и выполняем компиляцию всех требуемых модулей, далее с 6 по 10 мы запускам симуляцию и добавлем все необходимо на waveform, после чего мы запускаем проект.

Результат запуска выглядит следующим образом:

Изображение выглядит как снимок экрана, линия, График, Графическое программное обеспечение

Автоматически созданное описание

Рис. 3.5. Результат моделирования.

Как видно, получившаяся waveform полностью соответствует ожиданиям и техническому заданию, что свидетельствует о корректно разработанном устройстве.

# Вывод:

В ходе лабораторной работы было разработано устройство на языке Verilog в соответствии с схемой. Получившееся устройство полностью соответствует техническому заданию.

В процессе разработки использовались IP-модули из библиотеки Quartus Prime, стоит отметить, что они сильно ускорили процесс и помогли избавиться от написания стандартных модулей, дав возможность сосредоточиться на основном задании.