Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе № 2\_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. Задача: 3](#_Toc158513791)

[2. Решение: 3](#_Toc158513792)

[2.1. Исходный код файла lab2\_3.v 3](#_Toc158513793)

[2.2. Структура из RTL viewer 5](#_Toc158513794)

[2.3. Исходный код теста 6](#_Toc158513795)

[2.4. Результаты моделирования 7](#_Toc158513796)

[3. Вывод: 7](#_Toc158513797)

# Задача:

* Описать на языке Verilog следующее устройство:



Рис. 1 – Схема устройства

**Выводы:**

1. ***Входы:***
2. *CLK* – тактовый сигнал.
3. *aRSTin* – вход асинхронного сброса (активный уровень для сброса – 1).
4. *[7:0] Din* – вход данных для управления ШИМ.
5. ***Выходы:***
   1. *PWM* – выход ШИМ.

**Модули:**

1. ***CNT*** – счетчик, создаваемый с помощью IP модуля LPM\_COUNTER:
2. Разрядность: 8 бит.
3. Двоичный счетчик на сложение.
4. Выход переноса (carry\_out).
5. Вход асинхронного сброса (clear).
6. ***CMP*** – модуль сравнения, создаваемый с помощью IP модуля LPM\_COMPARE:
7. Два входа по 8 бит.
8. a < b
9. Без знаковый.
10. Без конвейеризации.
11. ***RG*** – регистр, описываемый на Verilog в файле верхнего уровня:
12. arst – вход асинхронного сброса (активный уровень – 1) устанавливает 8`d128.
13. ***DFF*** – триггеры, описываемые на Verilog в файле верхнего уровня:
    1. aset – вход асинхронно устанавливает триггер в 1.

# Решение:

## Исходный код файла lab2\_3.v

С помощью кода на языке Verilog опишем устройство, представленной на рисунке 1. Листинг кода представлен ниже:

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

Автоматически созданное описание

Рис. 2 – Код файла lab2\_3.v

Данный код последовательно описывает модули, из которых состоит устройство, описанное в условии задания. Этот Verilog-модуль представляет собой устройство, состоящее из следующих компонентов:

1. DFF\_arst: Регистр с асинхронным сбросом, управляемый тактовым сигналом и внешним сигналом сброса.
2. RG: Регистр с асинхронным сбросом, инициализированный 128. Значение обновляется в зависимости от условий. Поскольку регистр должен иметь асинхронный сигнал сброса, поэтому always реагирует не только на фронт clk, но и на фронт сигнала сброса.
3. CNT: 8-битный счетчик, управляемый тактовым сигналом и асинхронным сигналом сброса.
4. CMP: Компаратор, сравнивающий значения из регистра и счетчика.
5. DFF\_out: Регистр с асинхронным сбросом для формирования выходного сигнала PWM. Если активен сброс, выход устанавливается в 1, иначе принимает значение сигнала компаратора.

\*Блоки CNT и CMP созданы средствами Quartus, а в файле с кодом выше происходит лишь их инициализация для дальнейшего использования.

## Структура из RTL viewer

Скомпилируем код, представленный выше (lab2\_3.v) и откроем схему получившегося устройства в RTL viewer:

Изображение выглядит как диаграмма, снимок экрана, План, линия

Автоматически созданное описание

Рис. 3 – Структура устройства из RTL viewer

Сравнив полученную схему (рис. 3) со схемой заданного устройства (рис. 1), можем увидеть, что они имеют аналогичную структуру.

При более подробном рассмотрении можно увидеть, что на входе 7-го разряда регистра Din возникает инверсия. Чтобы понять причину её возникновения рассмотрим отдельно CNT при помощи средств Technology Map Viewer. На картинке ниже можно увидеть инвертор (выделен красным цветом). Его появление связано с асинхронной установкой в регистр значения 8’d128. Возникновение сигнала arst устанавливает счетчик в 0, старший разряд инвертируется. Таким образом происходит запись числа 8’d128.

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Автоматически созданное описание

Рис. 4 – Расположение инвертора в Map Viewer

## Исходный код теста

Разработаем тест первого класса для рассматриваемого устройства. Листинг кода приведён ниже:

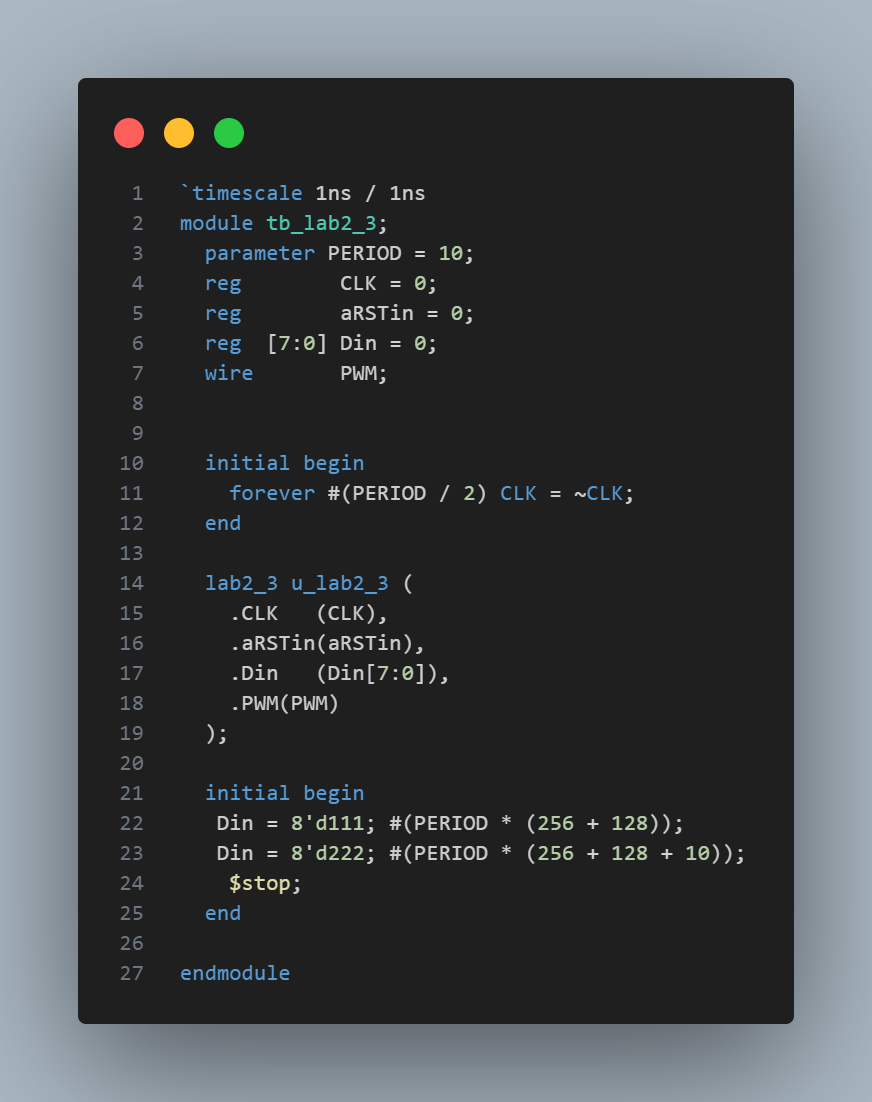


Рис. 5 – Код файла tb\_lab2\_3.v

Рассмотрим более детально основной цикл кода:

Изначально установим значение = 111 (рис. 6). Это значение запишется в регистр только после того, как счётчик достигнет максимального значения и выдаст сигнал carry out. Таким образом, чтобы подождать запись значения подождём 256 тактов.

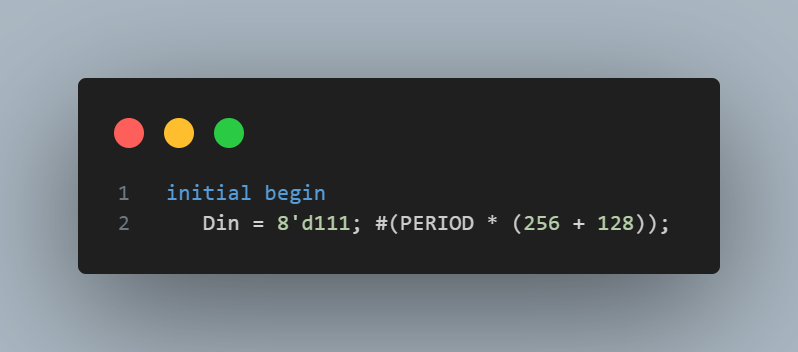


Рис. 6 – Задаем значение = 8’d111

После этого ждем еще 128 тактов и меняем значение на 2220, однако запись произойдет еще через 128 тактов. Смотрим на результат 256 + 10 тактов, чтоб увидеть, как выход сбросится опять до 0.

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Рис. 7 – Задаем значение = 8’d222

## Результаты моделирования

Запустим тестовый файл, используя средства Quartus Prime. Полученная waveform представлена на рисунке 8 ниже:

Изображение выглядит как Мультимедийное программное обеспечение, программное обеспечение, линия, Графическое программное обеспечение

Автоматически созданное описание

Рис. 8 – Результат моделирования

Из рисунка видно, что получившаяся временная диаграмма соответствует ожиданиям и ТЗ, что свидетельствует о корректном описании устройства.

# Вывод:

В ходе работы было разработано устройство в соответствии со схемой, предоставленной в условии (рис. 1). Результаты визуализации схемы в RTL Viewer подтвердили полное соответствие проектирования устройства требованиям задачи. Кроме того, анализ временной диаграммы подтвердил правильность функционирования устройства.

Активное применение функционала Quartus Prime, в том числе интеграция модулей через IP Catalog, значительно улучшило процесс разработки схемы. Этот подход позволил сократить затраты времени, которые ранее требовались на написание стандартных модулей, таких как LPM\_COUNTER и LPM\_MULT.