Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по лабораторной работе № 6**

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

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

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

“13” марта 2024 г.

Санкт-Петербург

2024

Оглавление

[1. Список иллюстраций: 2](#_Toc161144763)

[2. Цель упражнения: 3](#_Toc161144764)

[3. Алгоритм работы проекта: 3](#_Toc161144765)

[4. Решение: 3](#_Toc161144766)

[5. Вывод: 6](#_Toc161144767)

# Список иллюстраций:

[Рис. 4.1. RTL схема разработанного АЛУ. 4](#_Toc161144768)

[Рис. 4.2. Waveform для тестовой последовательности. 5](#_Toc161144769)

[Рис. 4.3. Результат запуска в консоли. 5](#_Toc161144770)

[Рис. 4.4. Настройки Signal Tap II. 5](#_Toc161144771)

[Рис. 4.5. ISSP операция ADD. 5](#_Toc161144772)

[Рис. 4.6. ISSP операция SUB. 6](#_Toc161144773)

[Рис. 4.7. ISSP операция MUL. 6](#_Toc161144774)

[Рис. 4.8. ISSP операция DIV. 6](#_Toc161144775)

[Рис. 4.9. ISSP операция OP\_18. 6](#_Toc161144776)

[Рис. 4.10. Результат в Signal Tap II. 6](#_Toc161144777)

# 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:

АЛУ (арифметико-логическое устройство) с параметризированной разрядностью (width=8, по умолчанию). Реализуется как комбинационная схема. Выполняет знаковые операции:

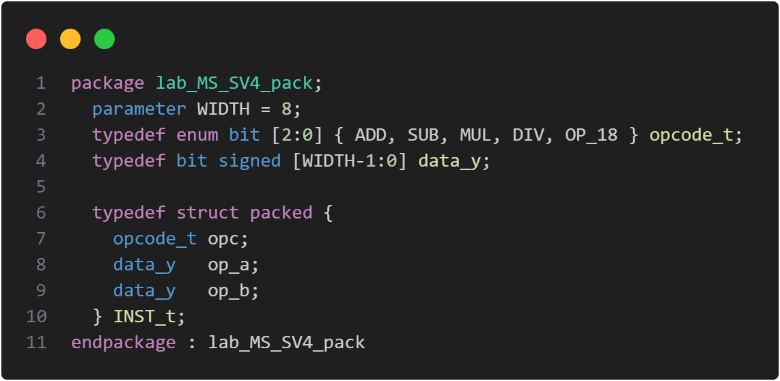
* ADD – сложение.
* SUB – вычитание.
* MUL – умножение.
* DIV – деление.
* OP\_18 – среднее арифметическое.

Выводы устройства:

* ops – вход кода операции
* op\_a – вход операнда А: знаковый (разрядность width).
* op\_b – вход операнда B: знаковый (разрядность width).
* ALU\_out – выход результата: знаковый (разрядность width).
* CLK – вход тактового сигнала, для синхронизации ISSPE и SignalTapII.

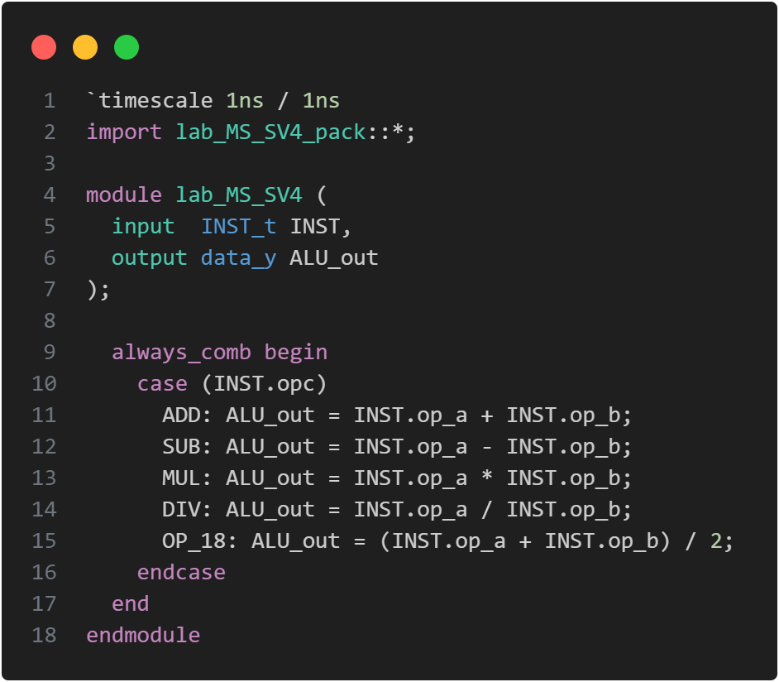
# Решение:

Первым делом необходимо создать пакет для типов данных, используемых в проекте:



Здесь мы видим перечисление кодов операций и общий тип данных для входов и выходов. Эти типы объединены в структуру.

На основе этого пакета создадим модуль АЛУ:



В качестве входных данных принимается структура INST\_t, созданная ранее, а выходным параметром является выход АЛУ, который аналогично имеет ранее созданный тип данных.

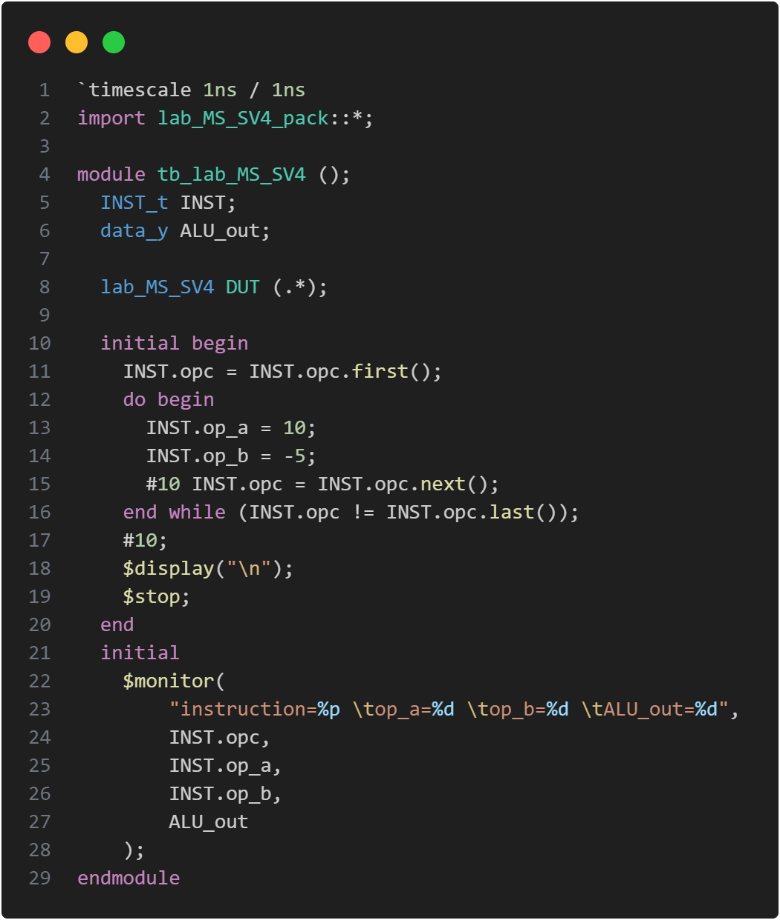
Выполним компиляцию и убедимся в том, что была создана именно комбинационная схема:

Изображение выглядит как диаграмма, текст, линия, План

Автоматически созданное описание

Рис. 4.1. RTL схема разработанного АЛУ.

Проверим корректность разработанного модуля, используя тест первого класса:



Этот тест проходит по всем состояниям и подает на вход 10 и -5, чтоб проверить, что операции корректно переключаются. Запустим его и получим следующий результат:

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

Автоматически созданное описание

Рис. 4.2. Waveform для тестовой последовательности.

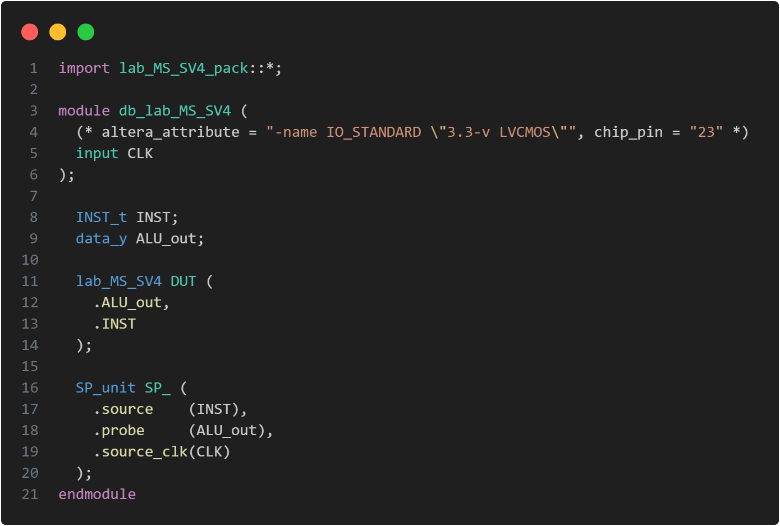
Изображение выглядит как текст, Цвет электрик, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 4.3. Результат запуска в консоли.

Как мы видим все операции работают корректно, в соответствии с заданием.

Разработаем модуль для отладки программы на плате:



Для ввода значений в модуль будем использовать ISSP, там же будем смотреть результат. Дополнительно добавим Signal Tap II, в котором будем получать значения по изменению типа операции и получать результат в виде сегментов:

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рис. 4.4. Настройки Signal Tap II.

Запишем наш проект на плату и начнем тестирование:



Рис. 4.5. ISSP операция ADD.

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.6. ISSP операция SUB.

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.7. ISSP операция MUL.

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.8. ISSP операция DIV.

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.9. ISSP операция OP\_18.

Изображение выглядит как красный, Карминный цвет

Автоматически созданное описание

Рис. 4.10. Результат в Signal Tap II.

Как видно по рисункам, приведенным выше, устройство работает корректно в соответствии с заданием, что было продемонстрированно преподавателю практики.

# 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.