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](#_Toc161238569)

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

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

[4. Решение: 3](#_Toc161238572)

[5. Вывод: 6](#_Toc161238573)

# Список иллюстраций:

[Рис. 4.1. RTL схема разработанного АЛУ. 4](#_Toc161238581)

[Рис. 4.2. Waveform для тестовой последовательности. 5](#_Toc161238582)

[Рис. 4.3. Результат запуска в консоли. 5](#_Toc161238583)

[Рис. 4.4. Настройки Signal Tap II. 5](#_Toc161238584)

[Рис. 4.5. ISSP операция ADD. 6](#_Toc161238585)

[Рис. 4.6. ISSP операция SUB. 6](#_Toc161238586)

[Рис. 4.7. ISSP операция MUL. 6](#_Toc161238587)

[Рис. 4.8. ISSP операция DIV. 6](#_Toc161238588)

[Рис. 4.9. ISSP операция OP\_18. 6](#_Toc161238589)

[Рис. 4.10. Результат в Signal Tap II. 6](#_Toc161238590)

# 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:

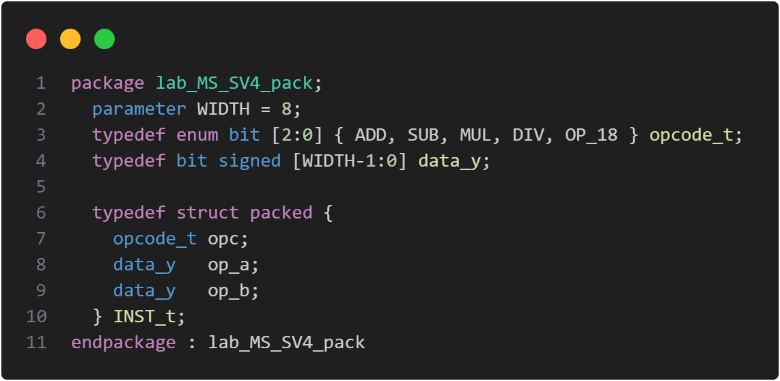
* ADD – сложение.
* SUB – вычитание.
* MUL – умножение.
* DIV – деление.
* OP\_18 – среднее арифметическое.

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.

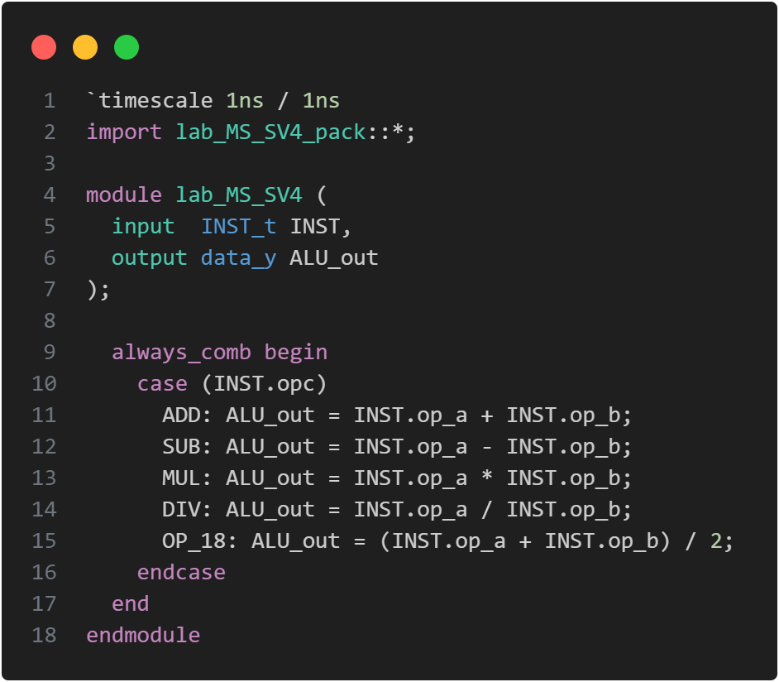
# Решение:

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.

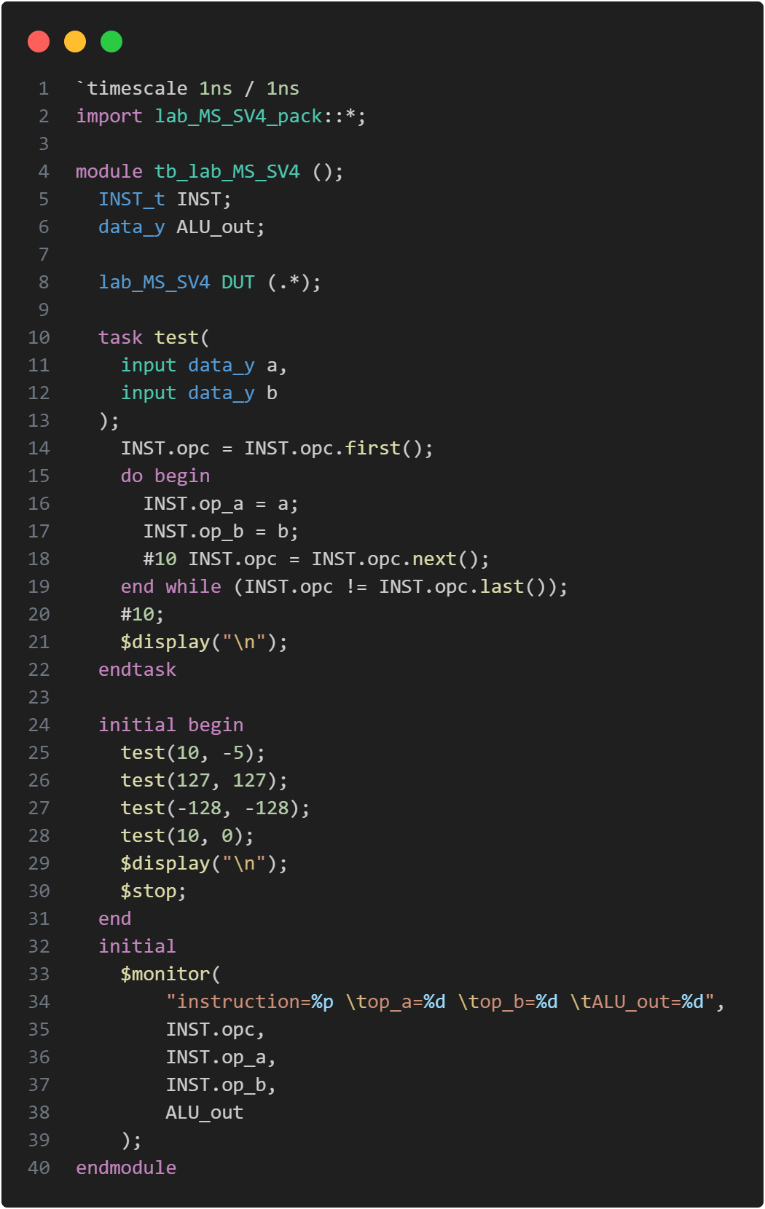
Выполним компиляцию и убедимся в том, что была создана именно комбинационная схема:

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 операциям и подает на вход различные комбинации цифр, чтоб проверить, что операции корректно переключаются. Запустим его и получим следующий результат:

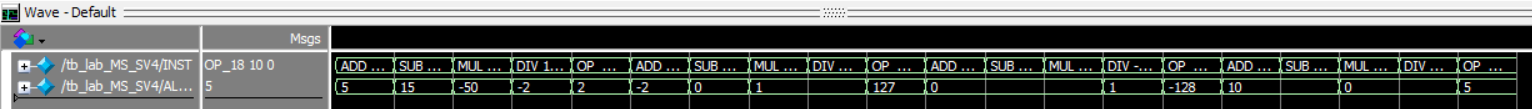


Рис. 4.2. Waveform для тестовой последовательности.

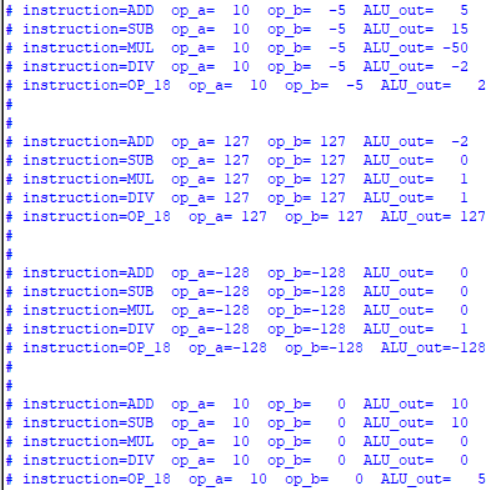
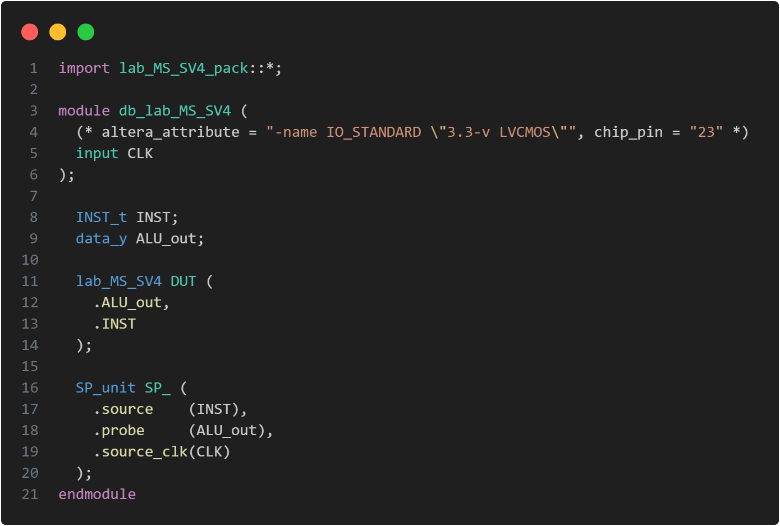


Рис. 4.3. Результат запуска в консоли.

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:

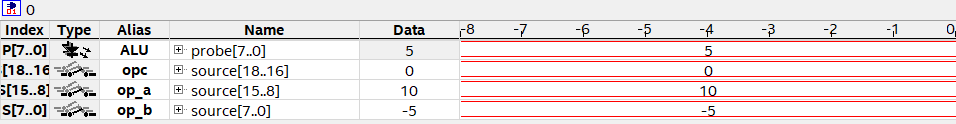


Рис. 4.5. ISSP операция ADD.

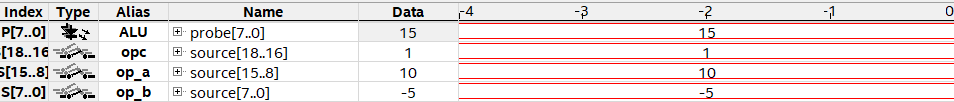


Рис. 4.6. ISSP операция SUB.

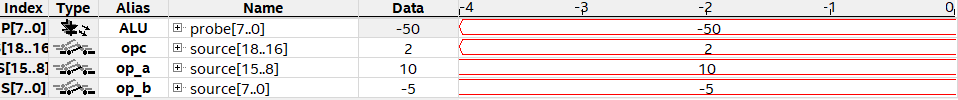


Рис. 4.7. ISSP операция MUL.

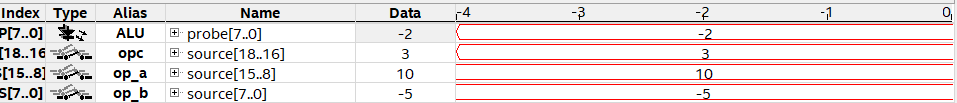


Рис. 4.8. ISSP операция DIV.

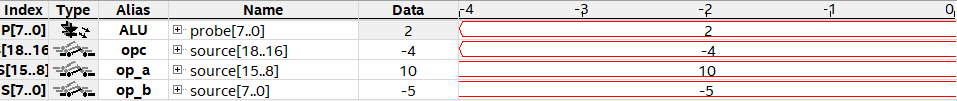


Рис. 4.9. ISSP операция OP\_18.

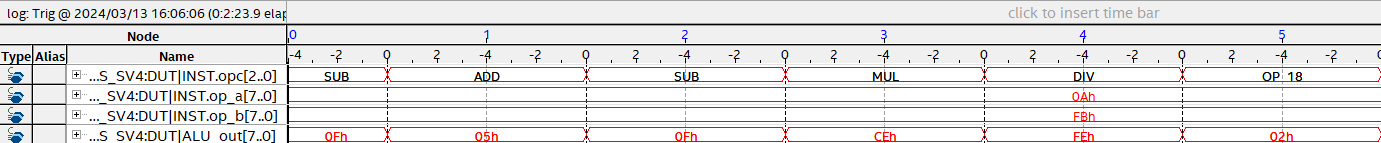


Рис. 4.10. Результат в Signal Tap II.

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.