САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа 3\_z1

Тема: Введение в Port-Level I/O Protocols

Дисциплина: «Проектирование реконфигурируемых гибридных вычислительных систем»

Выполнил:

Дроздов Н. Д.

Группа: 3540901/02001

Преподаватель:

Антонов А. П.

Санкт-Петербург

2021 г.

**Оглавление**

[1. Задание 4](#_Toc85117515)

[2. Исходный код функции 6](#_Toc85117516)

[3. Исходный код теста 6](#_Toc85117517)

[4. Исходный код командного файла 8](#_Toc85117518)

[5. Исходный код модифицированного теста 9](#_Toc85117519)

[6. Результаты исследования и сравнение решений 10](#_Toc85117520)

[7. Анализ результатов 12](#_Toc85117521)

[8. Результат модифицированного теста 13](#_Toc85117522)

[9. Выводы 14](#_Toc85117523)

**Список иллюстраций**

[Рисунок 1 Исходный код функции (файл lab3\_z1.c) 6](#_Toc85117531)

[Рисунок 2 - Исходный код теста 7](#_Toc85117532)

[Рисунок 3 Исходный код командного файла для создания проекта (la3\_z1.tcl) 9](#_Toc85117533)

[Рисунок 4 - Параметры ПК (Частота = 2.7 Гц) 10](#_Toc85117534)

[Рисунок 5 - Сравнение полученных решений 11](#_Toc85117535)

[Рисунок 6 - Ошибки при синтезировании третьего решения sol3 11](#_Toc85117536)

[Рисунок 7 - Таблица данных для всех решений 12](#_Toc85117537)

[Рисунок 8 График зависимости данных для всех решений 12](#_Toc85117538)

[Рисунок 9 - Schedule Viewer для sol1 13](#_Toc85117539)

[Рисунок 10 - Schedule Viewer для sol2 13](#_Toc85117540)

[Рисунок 11 - Временные показатели для модифицированного теста 14](#_Toc85117541)

1. Задание

* Создать на языке Си функцию:
  + Последовательно считывающую элементы массива, заполненного случайными целыми числами (тип int), имеющими значения от **0 до M-1**. Размер массива - **N** элементов.
  + Функция подсчитывает количество появлений чисел от 0 до **M-1** во входном массиве и записывает полученные значения в соответствующие элементы (с 0 до **M-1)** выходного массива, имеющего **M** элементов типа int.

*Т. е. по входному массиву размером N, заполненного случайными целыми числами от* **0 до M-1** *строится гистограмма в выходном массиве размером M элементов.*

* + В файле lab3\_z1.h должны быть определены **M** и **N** и тип данных data\_sc, имеющий тип int.

Например, можно примерно так (возможны любые другие варианты):

Изображение выглядит как текст

Автоматически созданное описание

* Создать на языке Си тест для проверки работы функции. Тест должен обеспечивать
  + Запуск функции 2 раза
    - каждый раз с новым начальным значением генератора случайных чисел.
    - Перед следующим запуском функции не забудьте обнулить формируемый функцией выходной массив.
  + Формирование исходного входного массива, заполненного случайными числами.
    - N=8192, M=8192
  + проверку правильности полученного функцией результата (сравнение полученного функцией массива и ожидаемого массива, вычисленного в тесте с использованием другого, не такого как в функции, алгоритма построения гистограммы) и формирование признака успешного/неуспешного выполнения для каждого запуска функции.
* Отладить функцию и тест (при неправильном результате в любом из запусков функции должен сообщать об ошибке).
* Создать скрипт автоматизирующий процесс:
  + Создания проекта lab3\_z1,
  + Назначения функции lab3\_z1
  + Подключения файла lab3\_z1.c (папка source ),
  + Подключения файла lab3\_z1\_test.c (папка source),
  + Создания решении
    - sol1, для которого:
      * Port Level I/O interface по умолчанию
      * Микросхема: xa7a12tcsg325-1Q
      * Период тактового сигнала: 6нс, uncertainty 0.1нс.
      * Си моделирование
      * Синтез
      * С/RTL cosimulation (с опцией полной трассировки)
    - sol2, для которого:
      * Port Level I/O interface для входного массива ap**\_fifo**
      * Микросхема: xa7a12tcsg325-1Q
      * Период тактового сигнала: 6нс, uncertainty 0.1нс.
      * Синтез
      * С/RTL cosimulation (с опцией полной трассировки)
* Отладить и проверить работу созданного скрипта.
* После выполнения скрипта открыть GUI
* Убедиться, что созданы все решения
* Используя средства HLS сравнить полученные решения
  + использованные интерфейсы
  + аппаратные затраты
  + результаты планирования (Schedule viewer и Performance profile)
* В GUI Vivado HLS cоздайте решение sol3, в котором установите Port Level I/O interface для выходного массива ap**\_fifo**
* Осуществите синтез и объясните полученный результат.

Исследование времени выполнения на ПК

* **Используются исходные коды и результаты исследования, проведенного ранее.**
* На базе использованного выше Си теста создать отдельный, модернизированный, тест для проверки времени выполнения синтезируемой функции на ПК:
  + добавить в тест операторы измерения **времени выполнения** синтезируемой функции (например, как-то так: <https://solarianprogrammer.com/2019/04/17/c17-programming-measuring-execution-time-delaying-program/> ).
  + Увеличить количество запусков синтезируемой функции до 32. Для каждого запуска измерить время, найти среднее значение и вывести как результат.
  + Точность измерения времени (наносекунды).
  + Провести исследование времени выполнения синтезируемой функции на Вашем ПК
    - Осуществить компиляцию модернизированного теста и запустить его как отдельное приложение
    - В отчете привести:
      * Параметры Вашего ПК: тип процессора, частота работы процессора, объем ОЗУ
      * результаты измерения времени выполнения

1. Исходный код функции

Исходный код синтезируемой функции приведен на рисунке Рисунок 1

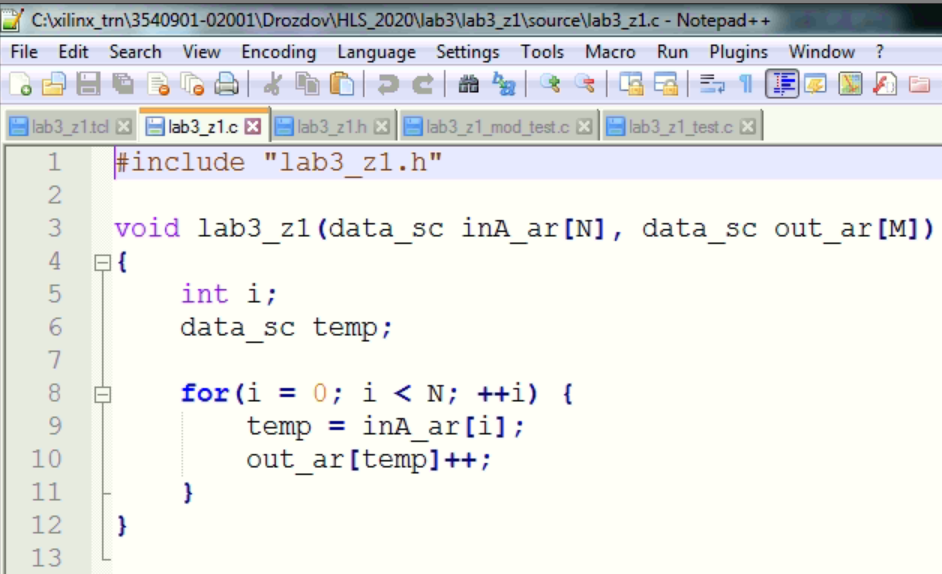


Рисунок 1 Исходный код функции (файл lab3\_z1.c)

Функция принимает два аргумента массива типа int — строит гистограмму входного массива и записывает в выходной массив.

1. Исходный код теста

Исходный код теста для проверки функции lab3\_z1 приведен на рисунке 2. Тест обеспечивает проверку корректной функции.

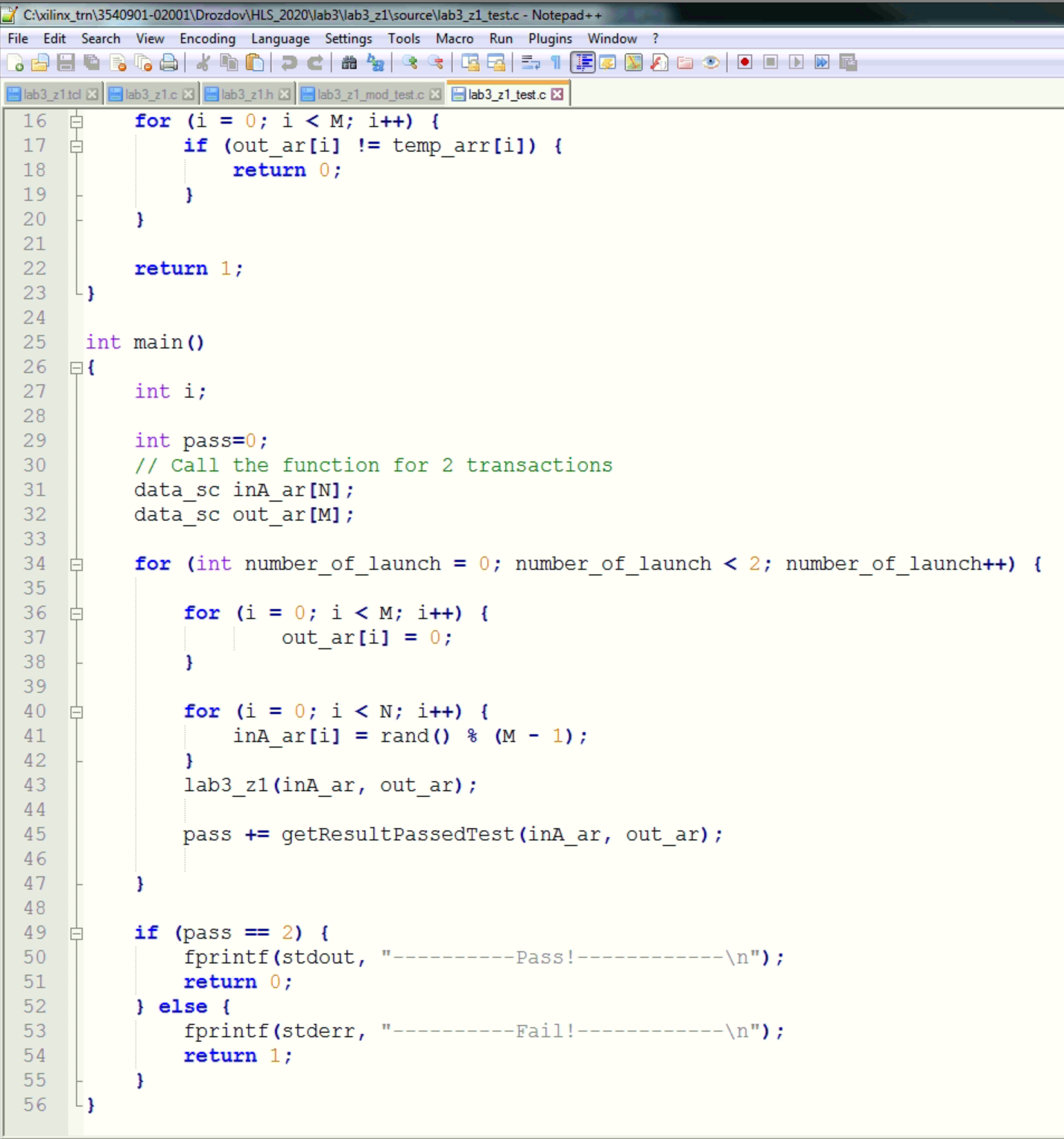


Рисунок 2 - Исходный код теста

1. Исходный код командного файла

На рисунке 3 и 4 представлен текст команд для автоматизированного создания следующих вариантов аппаратной реализаций:

1. Для sol1 задается clock period 6: clock uncertainty 0.1
2. Для sol2 задается clock period 6. clock uncertainty 0.1 и I/O protocol interface для входного массива inA\_arr — ap\_fifo.

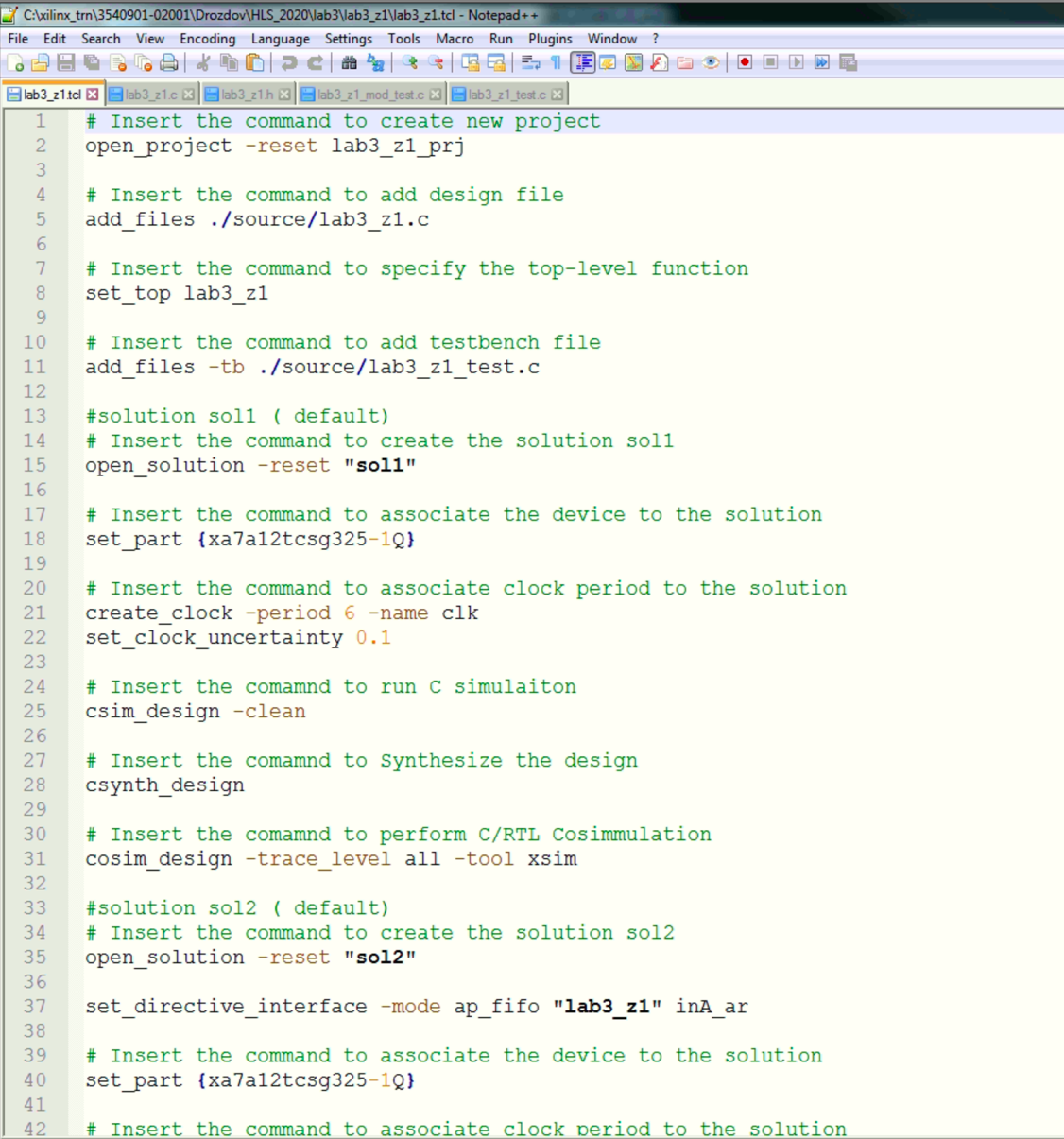


Рисунок 3 Исходный код командного файла для создания проекта (la3\_z1.tcl)

1. Исходный код модифицированного теста

На рисунке 4 представлен исходный код модифицированного теста для проверки функции lab3\_z1. Тест обеспечивает проверку производительности функции (Компилятор gcc-9.3.0).

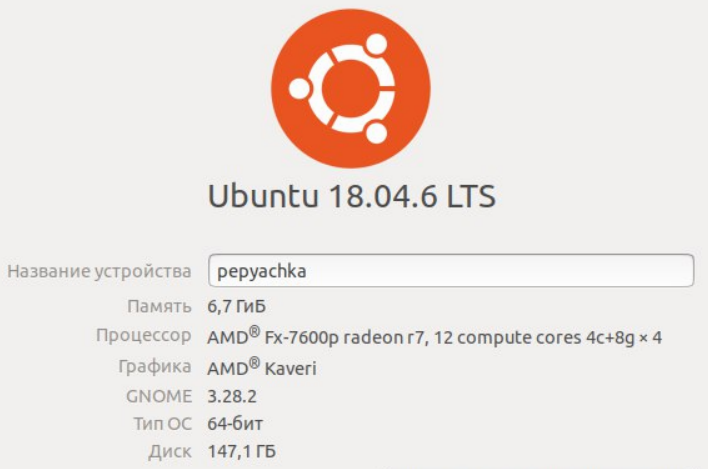


Рисунок 4 - Параметры ПК (Частота = 2.7 Гц)

1. Результаты исследования и сравнение решений

На рисунке 5 представлено сравнение из Vivado HLS GUI по аппаратным ресурсам, требуемых для реализации синтезируемой функции, и временным параметрам.

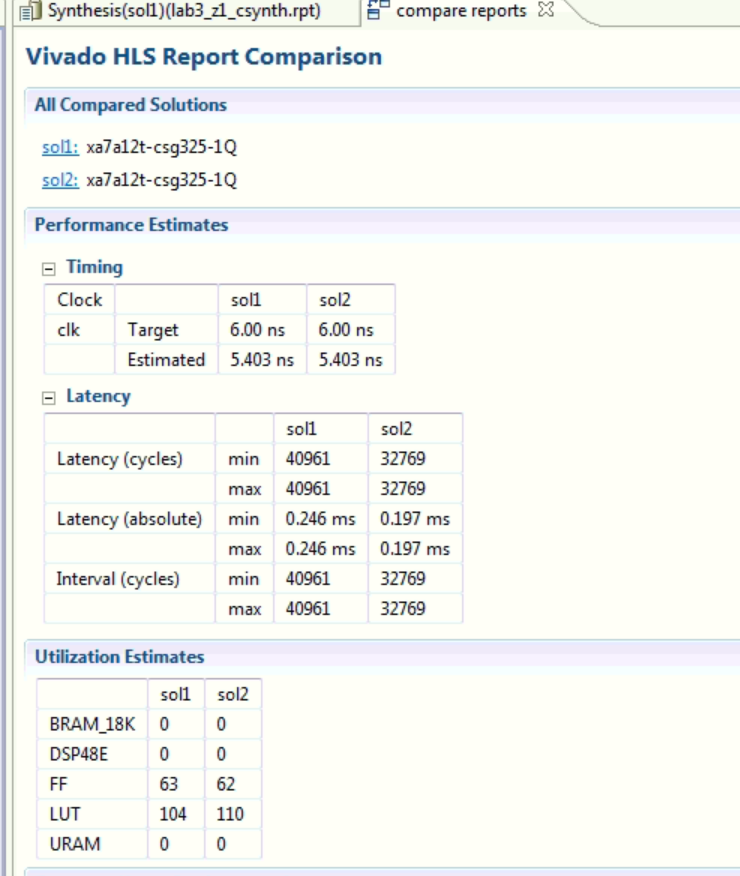


Рисунок 5 - Сравнение полученных решений

Target – планируемое время на один такт.

Estimated – оценочное время.

Latency (cycle) – количество тактов latency за один цикл.

Latency (absolute) – время затраченное на latency.

Во всех решениях используются FF-триггеры и LUT – таблицы перекодировки для реализации логических функций в количестве.

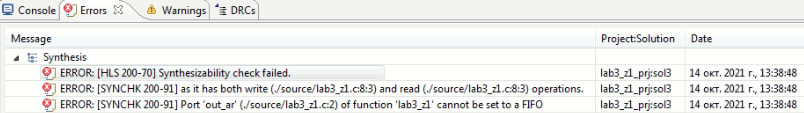


Рисунок 6 - Ошибки при синтезировании третьего решения sol3

После добавления третьего решения sol3 синтез прерывается из-за ошибки. Ошибка заключается в невозможности использования интерфейса ap\_fifo для выходного массива. Для выходного массива последовательность неизвестная, поэтому, при замене памяти на FIFO, синтез будет выдавать ошибки.

1. Анализ результатов

На Рисунке 7 представлена таблица с параметрами для всех решений.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 7 - Таблица данных для всех решений

На Рисунке 8 представлен график данных для сравнения всех решений.

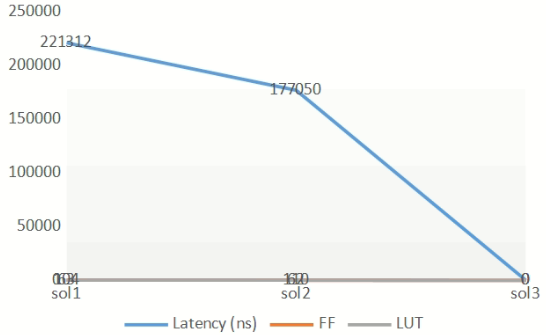


Рисунок 8 График зависимости данных для всех решений

Исходя из результатов, видно, что первое решение затрачивает больше времени и большее количество триггеров FF, а второе решение затрачивает больше модулей LUT. Разница по времени составляет 44262 нс.

Сделаем вывод, что лучшим является второе решение, так как мы получаем выигрыш по двум параметрам.

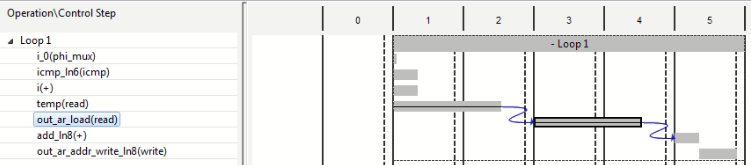


Рисунок 9 - Schedule Viewer для sol1

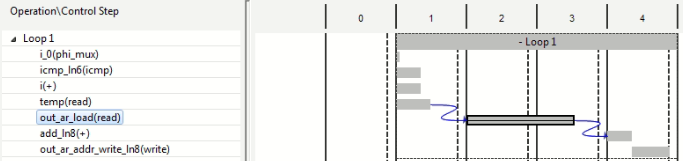


Рисунок 10 - Schedule Viewer для sol2

1. Результат модифицированного теста

На рисунке 10 представлен результат тестирования модифицированного теста. Среднее время после всех итераций составило 304.31 нс, что быстрее, чем решение, полученное синтезированием, на 170000 нс.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 - Временные показатели для модифицированного теста

1. Выводы

В данной работе была изучена возможность задания протоколов для I/O port-level у параметров функции. Был произведен сравнительный анализ между решением по умолчанию и заданным интерфейсом. Также было произведено сравнение временных показателей между решением полученным Vivado HLS и тестированием решения на ПК.