САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа lab2\_z2

Дисциплина:

«Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: Введение в Vivado HLS

Выполнил:

Бараев Д. Р.

Группа: 3540901/02001

Преподаватель: А. П. Антонов

Санкт-Петербург

2021

**Оглавление**

[1. Задание 4](#_Toc84508885)

[2. Исходный код функции 5](#_Toc84508886)

[3. Исходный код теста 6](#_Toc84508887)

[4. Исходный код модифицированного теста 7](#_Toc84508888)

[5. Исходный код командного файла 8](#_Toc84508889)

[6. Сравнение полученных решений и анализ результатов 9](#_Toc84508890)

[6.1 Электронная таблица и график для решений 10](#_Toc84508891)

[6.2 Результат модифицированного теста 11](#_Toc84508892)

[7. Выводы 12](#_Toc84508893)

**Список иллюстраций**

[Рисунок 1 Исходный код функции (файл lab2\_2.c) 5](#_Toc84508866)

[Рисунок 2 - Исходный код теста 6](#_Toc84508867)

[Рисунок 3 - Исходный код модифицированного теста 7](#_Toc84508868)

[Рисунок 4 - Параметры ПК 8](#_Toc84508869)

[Рисунок 5 Исходный код командного файла для создания проекта (lab2\_2.tcl) 9](#_Toc84508870)

[Рисунок 6 - Оценка быстродействия 10](#_Toc84508871)

[Рисунок 7 - Аппаратные ресурсы 10](#_Toc84508872)

[Рисунок 8 - Таблица данных для всех решений 11](#_Toc84508873)

[Рисунок 9 – График зависимости данных для всех решений 11](#_Toc84508874)

[Рисунок 10 - Результат теста производительности 12](#_Toc84508875)

1. Задание

* Используются исходные коды и результаты исследования lab2\_z1 (исходный код синтезируемой функции, .inc файл, тест скопировать в папку lab2\_z2\source и переименовать в lab2\_z2.xxx)
* Входной массив
* Размер массива - 8192 слова
* Тип данных (для студентов с четным номером в списке – int; с нечетным – long long)
* На базе скопированного 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
* Создать скрипт (на базе скрипта из lab2\_z1) для:
* Создания проекта lab2\_2,
* Подключения файла lab2\_2.c (папка source),
* Подключения файла lab2\_2\_test.c (папка source) – это скопированный из lab2\_z1 тест,
* Си моделирование.
* Создания решений - ex\_sol[2:1]
* Для ex\_sol1 задается clock period (при котором в lab2\_z1 производительность была максимальной); clock\_uncertainty 0.1
* Для ex\_sol2 задается clock period (при котором в lab2\_z1 аппаратные затраты были минимальными); clock\_uncertainty 0.1
* Для каждого решения реализуется синтез.
* Зафиксировать для каждого из решений
* Время выполнения: Latency (ns) – подсчитывается путем умножения Latency (cycles) на период тактового сигнала Estimated;
* Использованные ресурсы (если значения какого-либо ресурса остаются неизменными для всех решений, то такой ресурс не следует отображать на временной диаграмме)

1. Исходный код функции

Исходный код синтезируемой функции приведен на рисунке Рисунок 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 Исходный код функции (файл lab2\_2.c)

Функция принимает 1 аргумент — массив целых чисел и возвращает максимальное число в массиве.

На стр.2 объявляется функция с возвращаемым типом данных «atype». Далее на стр. 3 объявляется переменная max типа «atype», куда записывается первый элемент массива для дальнейшего сравнения. В стр. 4 заводится цикл, в котором сравнивается переменная max с элементом массива равным итерации цикла. И в случае, если элемент массива больше max, то в max записывается этот элемент, тем самым реализуя алгоритм нахождения максимального числа в массиве. Далее в стр. 8 return возвращает max.

1. Исходный код теста

Исходный код теста для проверки функции lab2\_2 приведен на рисунке Рисунок 2. Тест обеспечивает проверку корректной функции.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2 - Исходный код теста

1. Исходный код модифицированного теста

На рисунке 3 представлен исходный код модифицированного теста для проверки функции lab2\_2. Тест обеспечивает проверку производительности функции (Компилятор gcc-9.3.0).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 - Исходный код модифицированного теста

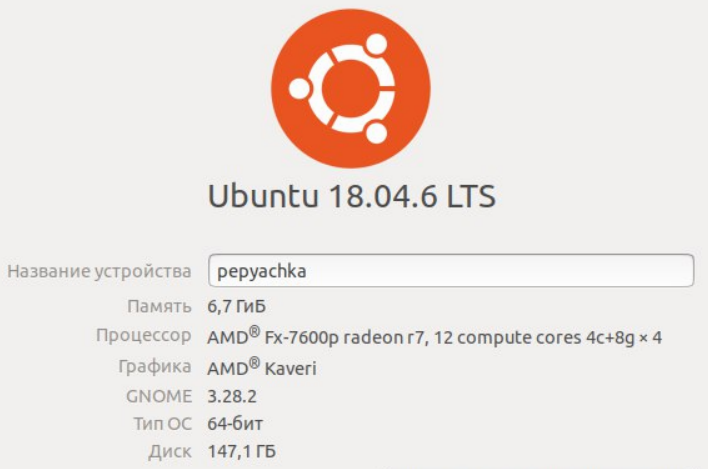


Рисунок 4 - Параметры ПК

1. Исходный код командного файла

На Рисунке 5 представлен текст команд для автоматизации создания вариантов аппаратной реализации:

1. Для ex\_sol1 задается clock period 6; clock\_uncertainty 0.1
2. Для ex\_sol2 задается clock period 20; clock\_uncertainty 0.1

В коде командного файла рисунке 5 на стр. 2 создаётся проект под именем «lab2\_2». На стр. 5 к проекту подключается файл lab2\_2.c. На стр. 8 задаётся файл верхнего уровня. На стр. 11 подключается файл исходного теста. На стр. 15 создаётся solution "ex\_sol1". На стр. 18 выбирается тип микросхемы. В строках 21–22 добавляется тактовый период в 6 наносекунд и погрешность 0.1. На стр.25 запускается симуляция. Командой «vivado\_hls -f lab2\_2.tcl” создаем проект.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 Исходный код командного файла для создания проекта (lab2\_2.tcl)

1. Сравнение полученных решений и анализ результатов

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на Рисунке 6.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6 - Оценка быстродействия

Target – планируемое время на один такт.

Estimated – оценочное время.

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

Latency (absolute) – время затраченное на latency.

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на Рисунке 7.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 7 - Аппаратные ресурсы

* 1. Электронная таблица и график для решений

На Рисунке 8 приведена таблица с параметрами для всех решений.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8 - Таблица данных для всех решений

На Рисунке 9 представлен график данных для сравнения всех решений.

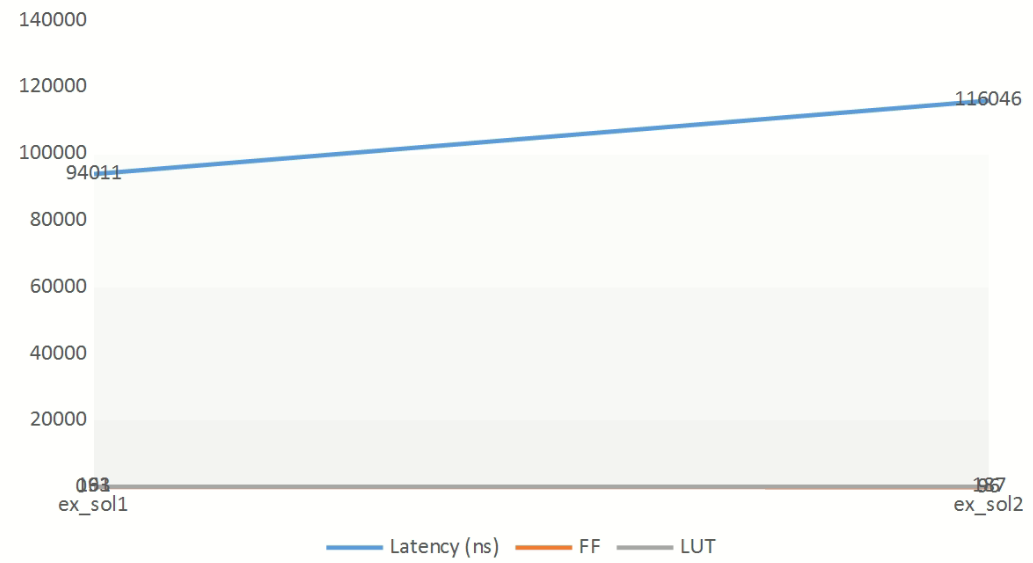


Рисунок 9 – График зависимости данных для всех решений

У них одинаковое затраченное (Estimated) время 7,082 нс и одинаковое количество тактов Latency 16386. Также одинаковое количество затраченных аппаратных ресурсов — FF и LUT.

* 1. Результат модифицированного теста

На рисунке 8 представлен результат модифицированного теста. Среднее время теста составило 60128 нс.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 - Результат теста производительности

Как видно из результатов, расчет на ПК оказался быстрее, чем на компьютерах в лаборатории. (60128 нс < 94022 нс).

1. Выводы

В ходе данной лабораторной работы была изучена возможность создания проекта и решений для проекта в Vivado HLS с помощью командной строки. Для автоматизированного создания проекта был создан скрипт, в котором было прописано создание проекта и решений с различными временными параметрами. Был произведен анализ теста производительности синтезируемой функции на микросхеме и на ПК.