САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа lab2\_z1

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](#_Toc84122634)

[2. Исходный код функции 6](#_Toc84122635)

[3. Исходный код теста 7](#_Toc84122636)

[4. Исходный код командного файла 7](#_Toc84122637)

[5. Solution №1 10](#_Toc84122638)

[5.1 Задание Solution №1 10](#_Toc84122639)

[6. Сравнение полученных решений для Solution №1 10](#_Toc84122640)

[7. Анализ результатов 11](#_Toc84122641)

[8. Solution №2 12](#_Toc84122642)

[8.1. Задание Solution №2 12](#_Toc84122643)

[9. Сравнение полученных решений для Solution №2 12](#_Toc84122644)

[10. Анализ результатов 13](#_Toc84122645)

[11. Выводы 14](#_Toc84122646)

**Список иллюстраций**

[Рисунок 1 Исходный код функции (файл lab2\_1.c) 6](#_Toc84122607)

[Рисунок 2 - Исходный код теста 7](#_Toc84122608)

[Рисунок 3 Исходный код командного файла для создания проекта (lab2\_1.tcl) 9](#_Toc84122609)

[Рисунок 4 - Содержимое файла lab2\_1\_inc.c 10](#_Toc84122610)

[Рисунок 5 Оценка быстродействия 10](#_Toc84122611)

[Рисунок 6 Аппаратные ресурсы 11](#_Toc84122612)

[Рисунок 7 - Таблица данных для всех решений 11](#_Toc84122613)

[Рисунок 8 График зависимости данных для всех решений 12](#_Toc84122614)

[Рисунок 9 - Содержимое файла lab2\_1\_inc.c 12](#_Toc84122615)

[Рисунок 10 - Оценка быстродействия 13](#_Toc84122616)

[Рисунок 11 - Аппаратные ресурсы 13](#_Toc84122617)

[Рисунок 12 - Таблица данных для всех решений 14](#_Toc84122618)

[Рисунок 13 – График зависимости данных для всех решений 14](#_Toc84122619)

1. Задание

* Создать на языке Си функцию (синтезируемую функцию) поиска максимума (студенты с четными номерами используют оператор сравнения; студенты с нечетными номерами сравнение осуществляют путем вычитания и анализа знака результата) в одномерном массиве
  + На вход функции поступает массив
    - Тип элементов массива (atype) задается typedef в .inc файле, подключаемом к описанию функции и теста (при отладке можно определить atype как char).
    - Размер массива задается с помощью #define в .inc файле, подключаемом к описанию функции и теста. (при отладке можно определить как 16 слов)
    - Возвращаемое значение (возвращается с помощью return) имеет тип atype.
    - Создать на языке Си тест для проверки синтезируемой функции. Тест должен обеспечивать:
    - Заполнение входного (для синтезируемой функции) массива случайными данными и поиск максимального значения (для сравнения со значением, получаемым с синтезируемой функции),
    - запуск синтезируемой функции 2 раза (на одном и том же наборе входных данных),
    - проверку правильности вычисленного результата и формирование признака успешного/неуспешного выполнения для каждого запуска функции,
    - Отладить функцию и тест.
    - Создать скрипт, автоматизирующий процесс создания вариантов аппаратной реализации, включающий этапы:
    - Создания проекта lab2\_1,
    - Подключения файла lab2\_1.c (папка source ),
    - Подключения файла lab2\_1\_test.c (папка source),
    - Создания базового решения ex\_sol0, для которого
* задается микросхема: xa7a12tcsg325-1q,
* задается clock period 6; clock\_uncertainty 0.1,
* выполняется Си моделирование.
* создания семи решений - ex\_sol[7:1]
* Для каждого решения задается микросхема: xa7a12tcsg325-1q
* Для каждого решения отдельное требование к периоду тактового сигнала
* Для ex\_sol1 задается clock period 8; clock\_uncertainty 0.1
* Для ex\_sol2 задается clock period 10; clock\_uncertainty 0.1
* Для ex\_sol3 задается clock period 12; clock\_uncertainty 0.1
* Для ex\_sol4 задается clock period 14; clock\_uncertainty 0.1
* Для ex\_sol5 задается clock period 16; clock\_uncertainty 0.1
* Для ex\_sol6 задается clock period 18; clock\_uncertainty 0.1
* Для ex\_sol7 задается clock period 20; clock\_uncertainty 0.1
* Для каждого реализуется синтез.
* Для каждого решения реализуется моделирование cosim.
* Отладить и проверить работу созданного скрипта.
* Провести два исследования:
* Исследование 1:
* В atype задайте как int
* Размер массива задайте равным 128 слов
* После выполнения скрипта надо открыть GUI
* используя средства HLS сравнить полученные решения.
* Составить электронную таблицу для сравнения решений (перенести в нее данные из HLS) и построить график, в котором должны быть отражены: 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)
* Исследование 2:
* В atype задайте как long long
* Размер массива задайте равным 128 слов
* После выполнения скрипта надо открыть GUI
* используя средства HLS сравнить полученные решения.
* Составить электронную таблицу для сравнения решений (перенести в нее данные из HLS) и построить график, в котором должны быть отражены: 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\_1.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\_1 приведен на рисунке Рисунок 2. Тест обеспечивает проверку корректной функции.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 - Исходный код теста

1. Исходный код командного файла

На рисунке 3 представлен текст команд для автоматизации создания вариантов аппаратной реализации:

1. Для ex\_sol1 задается clock period 8; clock\_uncertainty 0.1
2. Для ex\_sol2 задается clock period 10; clock\_uncertainty 0.1
3. Для ex\_sol3 задается clock period 12; clock\_uncertainty 0.1
4. Для ex\_sol4 задается clock period 14; clock\_uncertainty 0.1
5. Для ex\_sol5 задается clock period 16; clock\_uncertainty 0.1
6. Для ex\_sol6 задается clock period 18; clock\_uncertainty 0.1
7. Для ex\_sol7 задается clock period 20; clock\_uncertainty 0.1
8. Для ex\_sol0 задается clock period 6; clock\_uncertainty 0.1

В коде командного файла рисунке 3 на стр. 2 создаётся проект под именем «lab2\_1». На стр.5 к проекту подключается файл lab2\_1.c. На стр.8 задаётся файл верхнего уровня. На стр. 11 подключается файл теста. На стр.14 создаётся базовый solution ‘base’. На стр. 17 выбирается тип микросхемы. В строках 20–21 добавляется тактовый период в 6 наносекунд и погрешность 0.1. На стр.25 запускается симуляция. В строках 28–29 создаются два списка ‘all\_solutions’ который содержит названия решений и ‘all\_period’ который содержит задержку для каждого решения. В строках 33–55 запускается цикл, на каждой итерации которого создаётся solution с разным тактовым периодом и погрешностью 0.1, а также на каждой итерации запускается синтез и симуляция проекта. Командой «vivado\_hls -f lab2\_1.tcl” создаем проект.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 Исходный код командного файла для создания проекта (lab2\_1.tcl)

1. Solution №1
   1. Задание Solution №1

На рисунке 4 представлены данные в файле lab2\_1\_inc.c для исследования №1 — размер массива 128 машинных слов и тип данных int.



Рисунок 4 - Содержимое файла lab2\_1\_inc.c

1. Сравнение полученных решений для Solution №1

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на Рисунке 5.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5 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 6.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6 Аппаратные ресурсы

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

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 все решения имеют одинаковое Latency(ns) и используют одинаковое количество аппаратных ресурсов — FF и LUT. Только решение с заданным Timing = 6ns имеет наименьший Estimated и большее количество аппаратных ресурсов 85 FF и 144 LUT, поэтому выделить лучшее решения нельзя.

1. Solution №2
   1. Задание Solution №2

На рисунке 9 представлены данные в файле lab2\_1\_inc.c для исследования №1 — размер массива 128 машинных слов и тип данных long long.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 - Содержимое файла lab2\_1\_inc.c

1. Сравнение полученных решений для Solution №2

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на рисунке 10.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 10 - 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 11.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 11 - Аппаратные ресурсы

1. Анализ результатов

На Рисунке 12 приведена таблица с параметрами для всех решений.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 12 - Таблица данных для всех решений

На Рисунке 13 представлен график данных для сравнения всех решений.

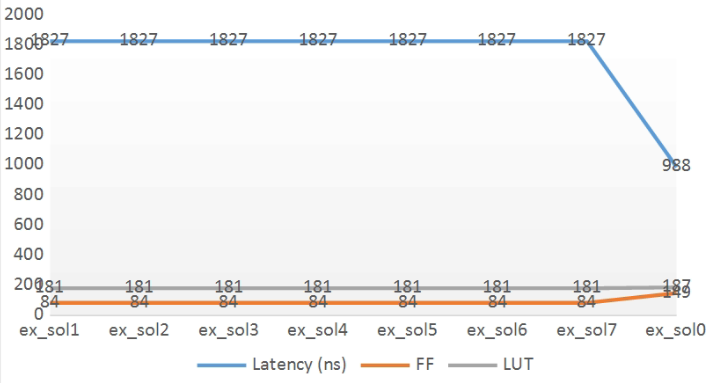


Рисунок 13 – График зависимости данных для всех решений

По полученному графику видно, что почти все решения имеют одинаковое Latency (ns) и используют одинаковое количество аппаратных ресурсов (FF и LUT), и только решение с заданным Timing = 6ns имеет наименьший Estimated и большее количество аппаратных ресурсов - 149 FF и 187 LUT. Исходя из полученных результатов можно сделать вывод, что наилучшего решения для данной функции нет.

1. Выводы

В ходе данной лабораторной работы была изучена возможность создания проекта и решений для проекта в Vivado HLS с помощью командной строки.

В работе было написано устройство, которое находит максимальный элемент в массиве и тест для проверки этого устройства. Так же был написан скрипт, который выполнял команды для создания 7 решений с разными тактовыми сигналами, скрипт запускался помощью «command prompt». Синтезируемая функция была протестирована с разными типами данными, такими как int и long long. В работе сравнивались все решения и выбирались оптимальные по двум критериям «аппаратные затраты» и «время задержки».