Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab1\_Z1

Дисциплина: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Введение в Vivado HLS

Выполнил студент гр. Шеметов С.А.

гр.5140901/21502

Руководитель, доцент Антонов А.П.

«\_\_» \_\_\_\_\_\_ 2023

Санкт-Петербург

2023

Оглавление

[1. Задание 3](#_Toc146319352)

[2. Исходный код функции 3](#_Toc146319353)

[3. Исходный код теста 3](#_Toc146319354)

[3.1. Моделирование 3](#_Toc146319355)

[4. Первое решение — Solution\_1 5](#_Toc146319356)

[4.1. Исходные настройки 5](#_Toc146319357)

[4.2. Синтез 5](#_Toc146319358)

[4.2.1. Оценка временных параметров 5](#_Toc146319359)

[4.2.2. Оценка производительности и аппаратных затрат 5](#_Toc146319360)

[4.2.3. Планировщик Schedule Viewer 6](#_Toc146319361)

[4.3. C/RTL моделирование 8](#_Toc146319362)

[4.3.1. Анализ результатов 8](#_Toc146319363)

[4.3.2. Отчет по производительности 8](#_Toc146319364)

[4.3.3. Временная диаграмма 8](#_Toc146319365)

[5. Второе решение — Solution\_2 9](#_Toc146319366)

[5.1. Исходные настройки 9](#_Toc146319367)

[5.2. Синтез 9](#_Toc146319368)

[5.2.1. Оценка временных параметров 9](#_Toc146319369)

[5.2.2. Оценка производительности и аппаратных затрат 9](#_Toc146319370)

[5.2.3. Планировщик Schedule Viewer 10](#_Toc146319371)

[5.3. C/RTL моделирование 10](#_Toc146319372)

[5.3.1. Анализ результатов 10](#_Toc146319373)

[5.3.2. Отчет по производительности 11](#_Toc146319374)

[5.3.3. Временная диаграмма 11](#_Toc146319375)

[6. Выводы 11](#_Toc146319376)

[6.1. Сравнение параметров двух решений 11](#_Toc146319377)

[6.2. Как определить минимальный период времени между моментами подачи новых данных на вход аппаратно реализованной функции для решения Solution1 и Solution2? 12](#_Toc146319378)

[6.3. Какой минимальный период времени между моментами поступления новых данных для решения Solution1 и Solution2? 12](#_Toc146319379)

[6.4. Какое решение более производительное? 13](#_Toc146319380)

1. Задание

* Разработать на языке **С++** описание функции, реализующей следующий алгоритм

y = inA + inB + inC **–** inD.

* Тип данных для inA, inB, inC, inD – short; для возвращаемого значения y – int;
* Разработать тест, обеспечивающий автоматическую проверку получаемых результатов моделирования разработанной функции.
* Создать, провести исследование и сравнительный анализ двух аппаратных реализаций разработанного на языке С++ описания функции.
  + Микросхема: xa7a12tcsg325-1q
  + clock period 6; clock\_uncertainty 1 (для решения Solution 1)
  + clock period 10; clock\_uncertainty 1 (для решения Solution 2)

2. Исходный код функции

Исходный код синтезируемой функции приведен на рисунке 1.

*Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, снимок экрана

Автоматически созданное описание*

*Рис.1. Исходный код фукнции lab1\_z1*

Функция принимает 4 аргумента — скаляра считает значение по формуле и возвращает результат расчета.

3. Исходный код теста

Исходный код теста для проверки функции lab1\_z1, исправленный в соответствии с заданием, приведен на рисунке 2. Тест обеспечивает проверку корректной работы функции.

3.1. Моделирование

Результаты моделирования исходного кода синтезируемой функции приведены на рисунке 3. Результаты моделирования показывают, что тест успешно пройден — так как условия теста выполняются (функция *main* возвращает 0) и результат расчета совпадает с ожидаемыми значениями.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

*Рис.2. Исходный код теста*

*Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание*

*Рис.3. Лог результата выполнения моделирования*

Для проверки правильности работы теста в него было внесено изменение в ожидаемый результат - expected\_res = inA + inB + inC + inD. Результат моделирования приведён на рисунке 4. Он показывает, что тест отрабатывает ошибку корректно.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

*Рис.4. Лог результата при ошибке в тесте*

4. Первое решение — Solution\_1

4.1. Исходные настройки

При создании первого решения задаются следующие настройки:

* clock period: 6,
* clock uncertain: 1,
* part: xa7a12tcsg325-1q

4.2. Синтез

4.2.1. Оценка временных параметров

Результаты оценки временных параметров представлены на рисунке 5.

Target timing — заданная требуемая частота равная 6 нс.

Estimated timing — достигнутая частота равна 4.095 нс.

Uncertainty – неопределенность, которая гарантирует некоторый запас времени для неизвестных задержек из-за целевого устройства.

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

*Рис.5. Оценка временных параметров для solution\_1*

4.2.2. Оценка производительности и аппаратных затрат

Результаты оценки производительности и аппаратных затрат представлены на рисунке 6.

Latency(cycle) — необходимое количество циклов для получения вывода, равно 1.

Latency(absolute) — необходимое количество времени для получения вывода, равно 1 такт \* 6 нс = 6 нс.

Interval - число тактов до того, как функция сможет принять новые входные данные, равно 2.

Для реализации синтезируемой функции были использованы следующие ресурсы: 19 FF модулей (триггеров) для хранения данных, 73 модулей LUT (таблиц перекодировок).

Pipelined – указывает была ли выполнена конвейеризация. В данном случае нет.

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

Автоматически созданное описание

*Рис.6. Оценка производительности и аппаратных затрат для solution\_1*

4.2.3. Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунках 7-11.

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

Автоматически созданное описаниеРис.7. Планировщик для solution\_1*

Изображение выглядит как Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение, Редактирование

Автоматически созданное описание

*Рис.8. Планировщик для solution\_1. Операция y(-)*

Изображение выглядит как Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение, Редактирование

Автоматически созданное описание

*Рис.9. Планировщик для solution\_1. Операция c\_read(read)*

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

Автоматически созданное описание*

*Рис.10. Планировщик для solution\_1. Операция add\_In6(+)*

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

Автоматически созданное описание*

*Рис.11. Планировщик для solution\_1. Операция add\_In6\_1(+)*

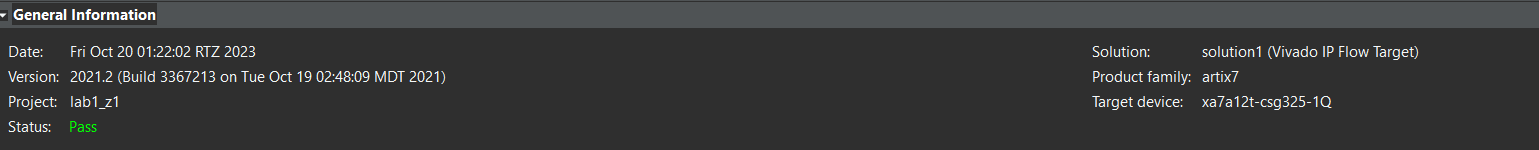
На рисунке 7 представлено два столбца, которые отображают 2 такта. На первом такте происходит чтение переменных *а* и *с* (рисунок 9) и их сложение с помощь операции *add\_In6(+)* (рисунок 10). На втором такте происходит чтение переменных *b* и *d*; *b* прибавляется к сумме *a+c*, полученной на предыдущем такте (рисунок 11) и вычисляется результат *y* – разница между суммой трёх переменных и *d* (рисунок 8).

Значение Latency совпадает с результатами Performance & Resource Estimates и равно 1 такту, также совпадает и Initiation Interval (II), и он равен 2 тактам.

4.3. C/RTL моделирование

4.3.1. Анализ результатов

Результаты, приведенные на рисунке 12, показывают, что синтезированное описание работает в соответствии с ожиданием — так как в отчёте моделирования отображается статус Pass.

 *Рис.12. Результат моделирования C/RTL для solution\_1*

4.3.2. Отчет по производительности

Отчет по производительности, полученный при совместном Си и RTL моделировании приведен на рисунке 13. Из отчета следует, что получены те же Latency и II, что и при синтезе, показанные на рисунке 6.

*Изображение выглядит как текст, снимок экрана, Шрифт, линия

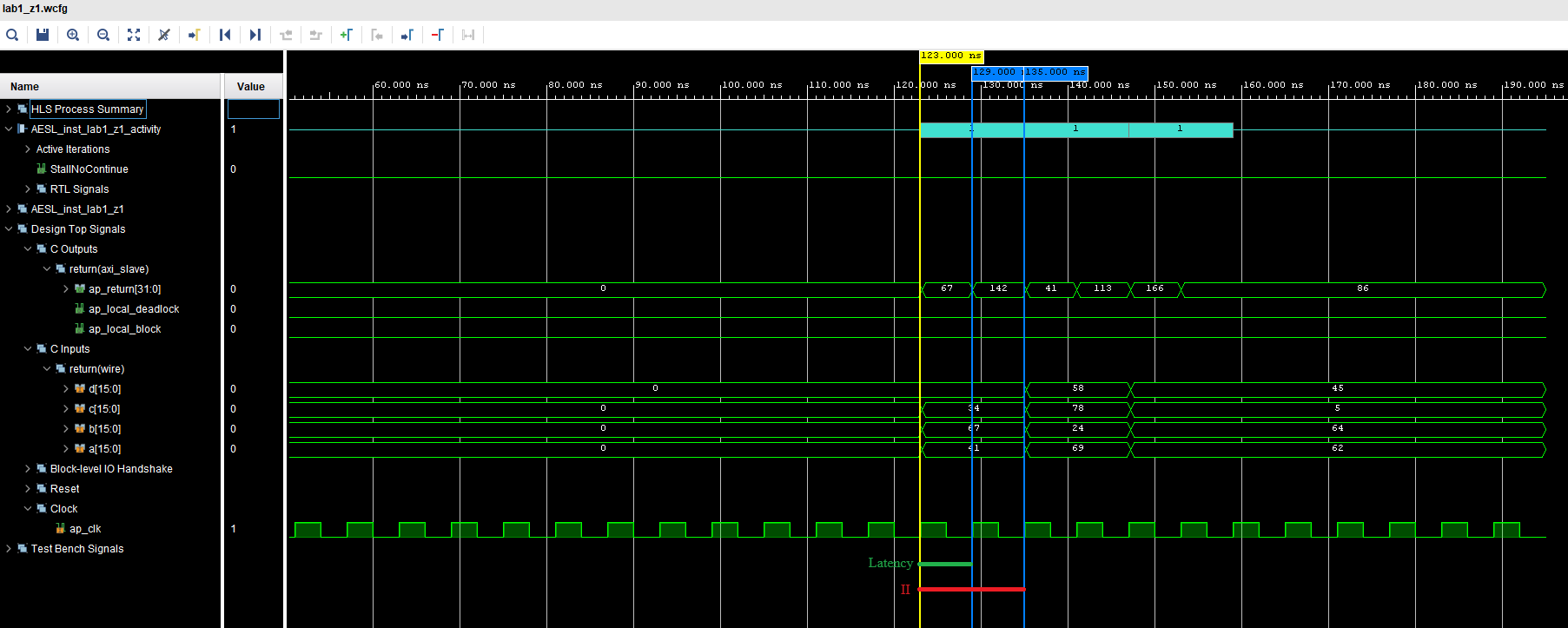
Автоматически созданное описание*

*Рис.13. Отчет по производительности C/RTL для solution\_1*

Avg, Max, Min – среднее, максимальное и минимальное число тактов для Latency/II. Значения в данных столбцах зависит от наличия ветвления в циклах. Так как в данной программе ветвления отсутствуют, все значения одинаковые.

4.3.3. Временная диаграмма

Временная диаграмма совместного C/RTL моделирования приведены на рисунке 14. На временной диаграмме приведено 3 цикла запуска аппаратной реализации созданной функции. На шине ap\_return[31:0] отображаются правильные выходные значения после Latency. Latency занимает 1 такт (6 нс) на строке Clock (ap\_clk), а II занимает 2 такта (12 нс).



*Рис.14. Временная диаграмма C/RTL для solution\_1*

5. Второе решение — Solution\_2

5.1. Исходные настройки

При создании второго решения задаются следующие настройки:

* clock period: 10,
* clock uncertain: 1,
* part: xa7a12tcsg325-1q

5.2. Синтез

5.2.1. Оценка временных параметров

Результаты оценки временных параметров представлены на рисунке 15.

Из рисунка видно, что заданная ожидаемая частота (Target) равно 10 нс, а достигнутая (Estimated) равна 6.241 нс.

*Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описание*

*Рис.15. Оценка временных параметров для solution\_2*

5.2.2. Оценка производительности и аппаратных затрат

Результаты оценки производительности и аппаратных затрат представлены на рисунке 16.

*Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение

Автоматически созданное описаниеРис.16. Оценка производительности и аппаратных затрат для solution\_2*

Latency(cycle) равно 0 тактам и Latency(absolute) 0 нс. Interval равен 1. Для реализации синтезируемой функции были использованы следующие ресурсы: 0 FF модулей (триггеров) для хранения данных, 60 модулей LUT – таблиц перекодировок. Нулевые значения были получен в связи с тем, что период был увеличен и все вычисления уложились в него.

5.2.3. Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 17.

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение, текст

Автоматически созданное описание

*Рис.17. Планировщик для solution\_2*

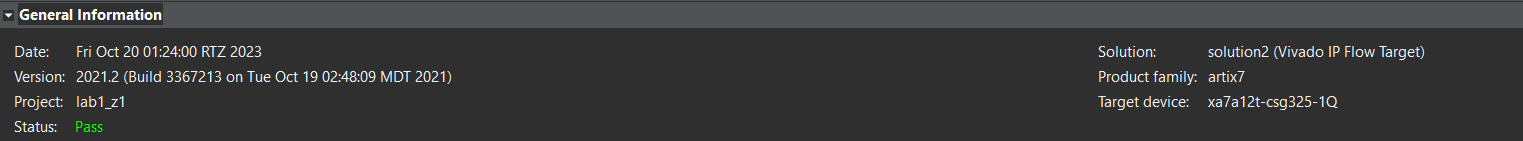
Как видно на рисунке 17 планировщик содержит 1 столбец. Это означает, что все операции – чтение четырёх переменных, сложение *a* и *c*, добавление к сумме *b* и вычитание из суммы *d* – выполняется за один период.

Значение Latency совпадает с результатами Performance & Resource Estimates и равно 0 тактов, также совпадает и Initiation Interval (II), и он равен 1 такту.

5.3. C/RTL моделирование

5.3.1. Анализ результатов

Результаты, приведенные на рисунке 18, показывают, что синтезированное описание работает в соответствии с ожиданием — так как в отчёте моделирования отображается статус Pass.



*Рис.18. Результат моделирования C/RTL для solution\_2*

5.3.2. Отчет по производительности

Отчет по производительности, полученный при совместном Си и RTL моделировании приведен на рисунке 19. Из отчета следует, что получены те же Latency и II, что и при синтезе, показанные на рисунке 16.

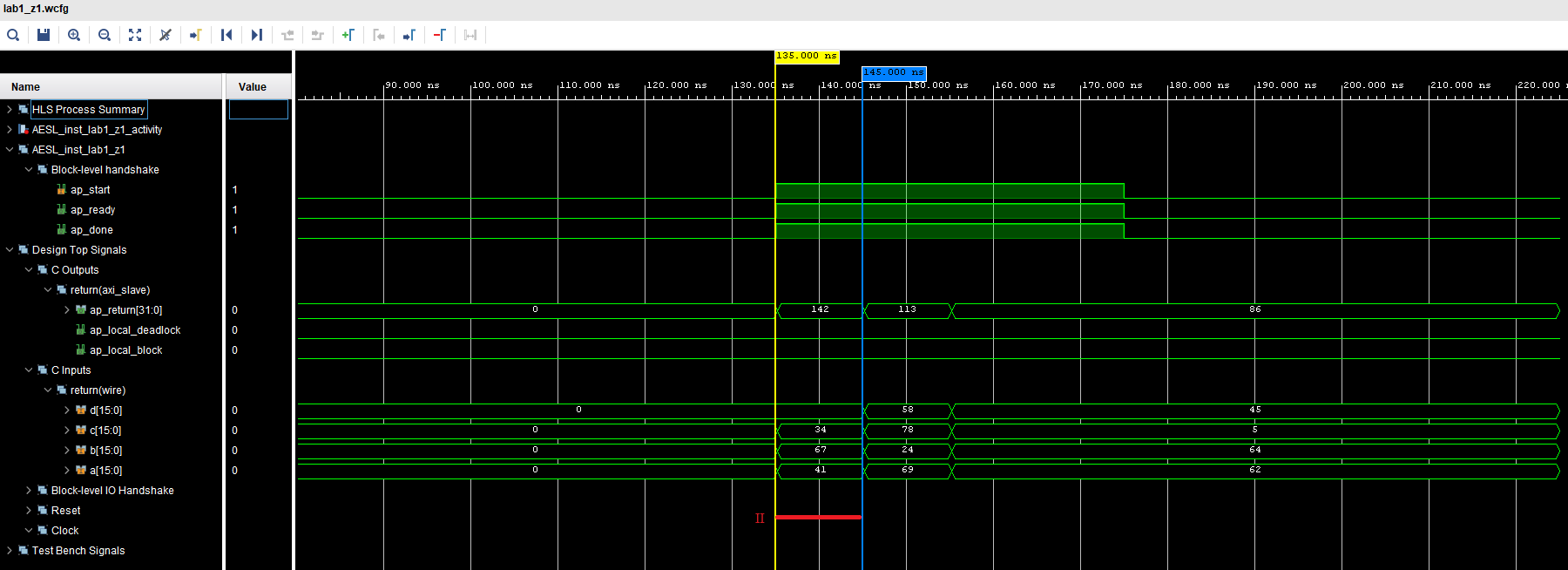
Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.19. Отчет по производительности C/RTL для solution\_2*

5.3.3. Временная диаграмма

Временная диаграмма совместного C/RTL моделирования приведены на рисунке 20. На временной диаграмме приведено 0 циклов запуска аппаратной реализации созданной функции. Так как в результате моделирования была порождена комбинационная схема, то тактовый сигнал clk отсутствует (тактов не требуется) и на шине ap\_return[31:0] отображаются правильные выходные значения одновременно с входными значениями. Latency равняется 0 нс, а II – 10 нс.



*Рис.20. Временная диаграмма C/RTL для solution\_2*

6. Выводы

6.1. Сравнение параметров двух решений

В данной работе были реализованы два решения с разной частотой, которые могут отличаться в зависимости от задаваемого параметра. Результаты сравнения параметров двух решений приведены на рисунке 21.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Рис.21. Сравнение двух решений*

Сравнение данных на рисунке 21 показывает, что при solution2 нет Latency, нет FF триггеров, на 13 меньше LUT и все операции выполняются за 1 такт. Но Estimated clock period на 2.146 нс выше, чем при solution1.

6.2. Как определить минимальный период времени между моментами подачи новых данных на вход аппаратно реализованной функции для решения Solution1 и Solution2?

Данный период вычисляется следующим образом: Interval(cycles) \* Target timing.

Для Solution1: 6 нс \* 2 такта = 12 нс.

Для Solution2: 10 нс \* 1 такт = 10 нс.

6.3. Какой минимальный период времени между моментами поступления новых данных для решения Solution1 и Solution2?

Данный период вычисляется следующим образом: Latency(cycles) \* Target timing.

Для Solution1: 6 нс \* 1 такт = 6 нс.

Для Solution2: 10 нс \* 0 тактов = 0 нс.

6.4. Какое решение более производительное?

Таким образом, наиболее производительным является второе решение, так как интервал между подачами новых данных меньше, поступают они сразу же и одновременно формируется выходное значение.