САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа 1

Тема: Введение в Vivado HLS

Дисциплина: «Проектирование реконфигурируемых гибридных вычислительных систем»

Выполнил:

Дроздов Н. Д.

Группа: 3540901/02001

Преподаватель:

Антонов А. П.

Санкт-Петербург

2021 г.

Оглавление

[Список иллюстраций 3](#_Toc83738717)

[Ход работы 6](#_Toc83738718)

[Часть 1 (lab1\_z1) 6](#_Toc83738719)

[Задание 6](#_Toc83738720)

[Исходный код функции 9](#_Toc83738721)

[Исходный код теста 9](#_Toc83738722)

[Моделирование 10](#_Toc83738723)

[Первое решение (Solution 1) 11](#_Toc83738724)

[Второе решение (Solution 2) 15](#_Toc83738725)

[Вывод 20](#_Toc83738726)

[Пояснение различий в результатах планирования и распределения ресурсов 20](#_Toc83738727)

[Пояснение различий во временных диаграммах моделирования для двух решений 20](#_Toc83738728)

[Ответы на поставленные в задании вопросы 21](#_Toc83738729)

[Часть 2 (lab1\_z2) 21](#_Toc83738730)

[Задание 21](#_Toc83738731)

[Исходный код функции 24](#_Toc83738732)

[Исходный код теста 24](#_Toc83738733)

[Моделирование 25](#_Toc83738734)

[Первое решение (Solution 1) 26](#_Toc83738735)

[Второе решение (Solution\_2) 31](#_Toc83738736)

[Вывод 36](#_Toc83738737)

# Список иллюстраций

[Рисунок 1 – Создание проекта 5](#_Toc83738738)

[Рисунок 2 – Создание проекта 6](#_Toc83738739)

[Рисунок 3 – Создание проекта 7](#_Toc83738740)

[Рисунок 4 - исходный код функции 8](#_Toc83738741)

[Рисунок 5 – Создание проекта 9](#_Toc83738742)

[Рисунок 6 – Создание проекта 10](#_Toc83738743)

[Рисунок 7 - Оценка быстродействия для solution\_1 11](#_Toc83738744)

[Рисунок 8 - Аппаратные ресурсы для solution\_1 11](#_Toc83738745)

[Рисунок 9 - Планировщик для solution\_1 12](#_Toc83738746)

[Рисунок 10 - Распределение ресурсов по тактам для solution\_1 12](#_Toc83738747)

[Рисунок 11 - Результаты моделирования Си и RTL для solution\_1 13](#_Toc83738748)

[Рисунок 12 - Быстродействие при совместном моделировании solution\_1 13](#_Toc83738749)

[Рисунок 13 - Временная диаграмма solution\_1 14](#_Toc83738750)

[Рисунок 14 - Оценка быстродействия для solution\_2 15](#_Toc83738751)

[Рисунок 15 - Аппаратные ресурсы для solution\_2 16](#_Toc83738752)

[Рисунок 16 - Планировщик для solution\_2 16](#_Toc83738753)

[Рисунок 17 - Распределение ресурсов по тактам для solution\_2 17](#_Toc83738754)

[Рисунок 18 - Результаты моделирования Си и RTL для solution\_2 17](#_Toc83738755)

[Рисунок 19 - Быстродействие при совместном моделировании для solution\_2 18](#_Toc83738756)

[Рисунок 20 - Временная диаграмма для solution\_2 18](#_Toc83738757)

[Рисунок 21 - Сравнение двух решений 19](#_Toc83738758)

[Рисунок 22 – Создание проекта 20](#_Toc83738759)

[Рисунок 23 – Создание проекта 21](#_Toc83738760)

[Рисунок 24 – Создание проекта 22](#_Toc83738761)

[Рисунок 25 - Исходный код функции 23](#_Toc83738762)

[Рисунок 26 - Исходный код теста 24](#_Toc83738763)

[Рисунок 27 - Результаты моделирования Си кода 25](#_Toc83738764)

[Рисунок 28 - Оценка быстродействия для solution\_1 26](#_Toc83738765)

[Рисунок 29 - Аппаратные ресурсы для solution\_1 27](#_Toc83738766)

[Рисунок 30 - Планировщик для solution\_1 27](#_Toc83738767)

[Рисунок 31 - Performance Profile для solution1 28](#_Toc83738768)

[Рисунок 32 - Распределение ресурсов по тактам для solution\_1 28](#_Toc83738769)

[Рисунок 33 - Результаты моделирования Си и RTL для solution\_1 29](#_Toc83738770)

[Рисунок 34 - Быстродействие при совместном моделировании solution\_1 29](#_Toc83738771)

[Рисунок 35 - Временная диаграмма solution\_1 30](#_Toc83738772)

[Рисунок 36 - Оценка быстродействия для solution\_2 31](#_Toc83738773)

[Рисунок 37 - Аппаратные ресурсы для solution\_2 32](#_Toc83738774)

[Рисунок 38 - Планировщик для solution\_2 32](#_Toc83738775)

[Рисунок 39 - Распределение ресурсов по тактам для solution\_2 33](#_Toc83738776)

[Рисунок 40 - Результаты моделирования Си и RTL для solution\_2 33](#_Toc83738777)

[Рисунок 41 - Быстродействие при совместном моделировании для solution\_2 34](#_Toc83738778)

[Рисунок 42 - Временная диаграмма для solution\_2 34](#_Toc83738779)

[Рисунок 43 - Сравнение двух решений 35](#_Toc83738780)

# Ход работы

## Часть 1 (lab1\_z1)

### Задание

Создали проект lab1\_z1 (рабочай папка – lab1\_z1 с заданием)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – Создание проекта

Подключили файл lab1\_z1.c (папка source)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок – Создание проекта

Подключили тест lab1\_z1\_test.c (папка source)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок – Создание проекта

Модифицировали тест – число, добавляемое на каждой итерации теста = порядковому номеру в списке группы.

Осуществили моделирование

Создали solution1

Задали: clock period 6; clock\_uncertainty 0.1

Осуществили синтез

Создали solution 2

### Исходный код функции

Исходный код синтезируемой функции приведен на рисунке 4

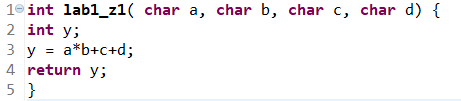


Рисунок - исходный код функции

В строке 1 объявляется функция с ее входными параметрами типа char. Далее на второй строке объявляется переменная y типа integer для хранения результата. В строке 3 происходит математические операции с числами и результат присваивается переменной y. В 4 строчке функция возвращает y.

### Исходный код теста

Исходный код теста для проверки функции lab1\_z (см. раздел 2), исправленный в соответствии с заданием, приведен на рисунке 5. Тест обеспечивает:

* Проверку результата функции **lab1\_z1**

В строках 5–15 рисунке 5 объявляются переменные и присваиваются к ним начальные значения. Переменная res объявленная на строке 6 будет в последствие хранить результат выполнения функции lab1\_z1. Массив refOut объявленный на строке 8 хранит предполагаемый результат для сравнения с фактическим.

Далее 18–32 строках используется массив с тремя итерациями где на каждой новой итерации к переменным inA, inB, inC, inD добавляется 1 (число варианта), и сравнивается с предполагаемым результатом, в случае несовпадения к переменной pass что объявлена на строке 9 прибавляется 1.

В строках 35–45 проверяется переменная pass, если она больше нуля, то возвращается 1 и строка о провале теста. Если ровна нулю, то возвращается 0 и строка об успешном завершение теста.

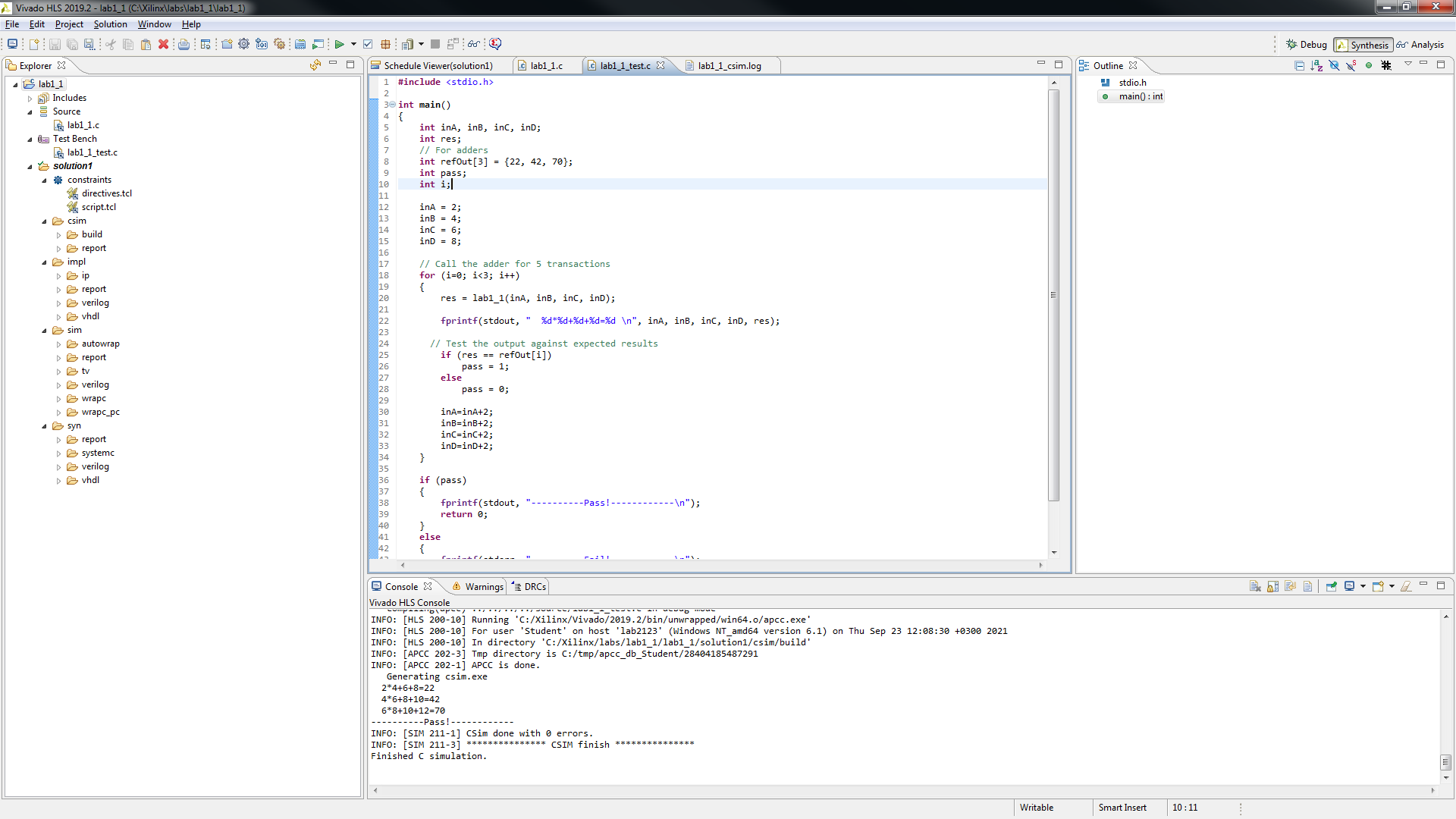


Рисунок – Создание проекта

### Моделирование

Результаты моделирования исходного кода синтезируемой функции приведены на рисунке 6. Результаты моделирования показывают, что тест успешно пройден – консоль оповестила об успешном прохождение теста. В случае если бы фактический результат не был равен ожидаемому функция main вернула бы 1, и консоль оповестила бы о не прохождении теста.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – Создание проекта

### Первое решение (Solution 1)

#### Исходные настройки:

* clock period: 6,
* clock uncertain: 0.1,
* part: xa7a12tcsg325-1q.

#### Синтез 1-го решения: оценка быстродействия

Результаты оценки быстродействия Timing/Summary и Latency/Summary приведены на рисунке 7.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Оценка быстродействия для solution\_1

Target – планируемое время на один такт.

* Estimated – оценочное время.
* Latency (cycle) – количество тактов latency за один цикл.
* Latency (absolute) – время затраченное на latency.

#### Синтез 1-го решения: оценка аппаратных ресурсов

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на рисунке 8

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Аппаратные ресурсы для solution\_1

Для реализации данного устройства были использованы DSP48E – встроенные умножители в количестве 1, FF-триггеры в количестве 12 шт. и LUT – таблицы перекодировки для реализации логических функций в количестве 37 шт.

#### Синтез 1-го решения: планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 9

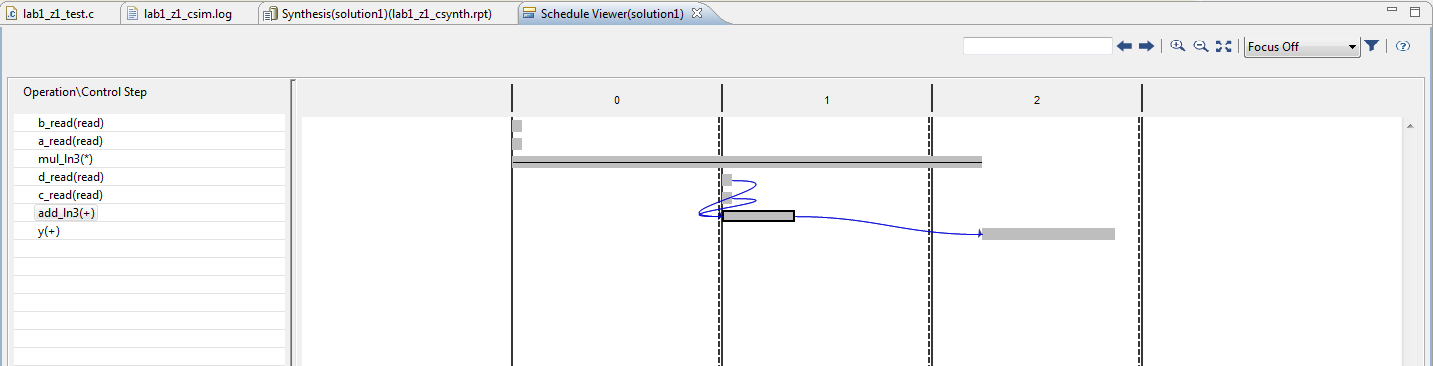


Рисунок - Планировщик для solution\_1

В планировщике видно, что чтение a и b происходят в нулевом такте, чтение c и d во втором такте, следовательно, у нас latency – 2 такта, а initiation interval – 3 такта.

#### Синтез 1-го решения: распределение ресурсов Resource Viewer

Распределение ресурсов по тактам приведено на рисунке 10.

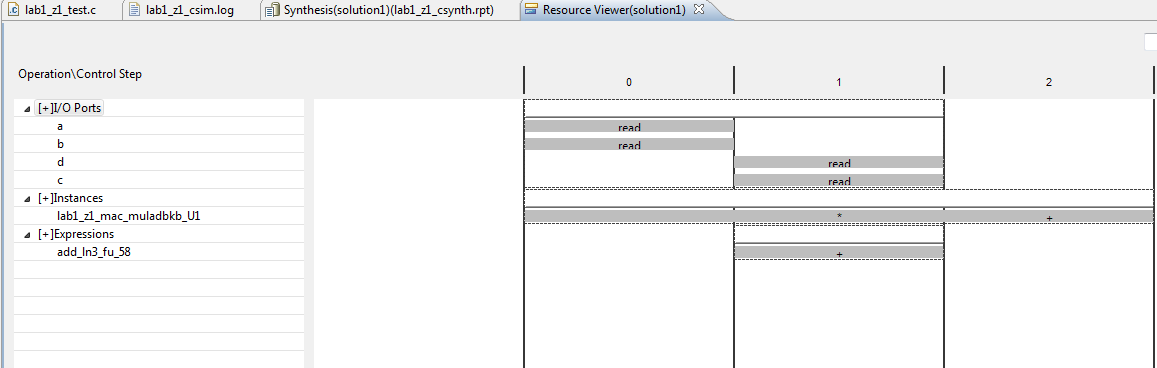


Рисунок - Распределение ресурсов по тактам для solution\_1

На нулевом такте считываются a и b, на первом такте считываются c и d и производится умножение a и b и сложение с c. На втором такте производится сложение с d.

#### Си и RTL моделирование: анализ результатов

Результаты, приведенные на рисунке 11, показывают, что синтезированное описание работает в соответствии с ожиданием – в информации указанно что при сравнение ожидаемого результата с фактическим ошибок не обнаружено, функция main вернула 0, следовательно тест успешно пройден.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Результаты моделирования Си и RTL для solution\_1

#### Си и RTL моделирование: отчет по быстродействию

Отчет по быстродействию, полученный при совместном Си и RTL моделировании приведен на рисунке 12. Из отчёта следует что за три такта выполнения одной итерации, два такта занимает latency.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Быстродействие при совместном моделировании solution\_1

#### Си и RTL моделирование: временная диаграмма

Временная диаграмма совместного Си|RTL моделирования приведены на рисунке 13.

Изображение выглядит как текст, электроника, дисплей

Автоматически созданное описание

Рисунок - Временная диаграмма solution\_1

На временной диаграмме solution 1 видно, что каждая итерация протекает в три такта. Первые два такта каждой итерации происходит считывание и вычисление на третий такт данные записываются и устройство готово считывать новые данные.

### Второе решение (Solution 2)

#### Исходные настройки

* clock period: 8,
* clock uncertain: 0.1,
* part: xa7a12tcsg325-1q.

#### Синтез 2-го решения: Оценка быстродействия

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на рисунке 14.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Оценка быстродействия для solution\_2

Target – планируемое время на один такт.

Estimated – оценочное время.

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

Latency (absolute) – время затраченное на latency.

#### Синтез 2-го решения Оценка аппаратных ресурсов

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на рисунке 15

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Аппаратные ресурсы для solution\_2

Для реализации данного устройства были использованы DSP48E – встроенные умножители в количестве 1 и LUT – таблицы перекодировки для реализации логических функций в количестве 16 шт.

#### Синтез 2-го решения: Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 16.

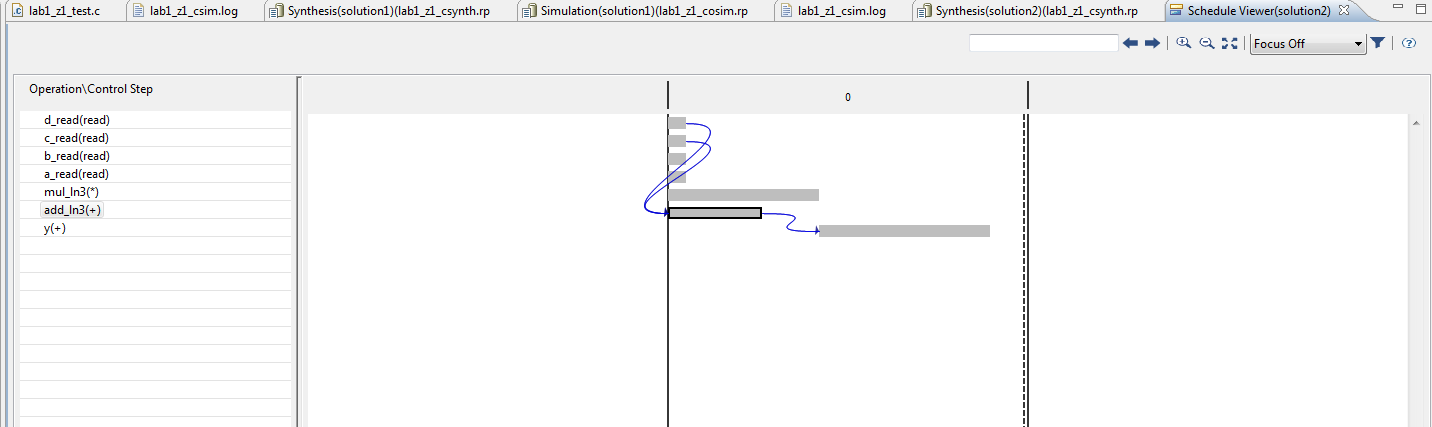


Рисунок - Планировщик для solution\_2

В планировщике видно, что все происходит за один такт.

#### Синтез 2-го решения: Распределение ресурсов Resource Viewer

Распределение ресурсов по тактам приведено на рисунке 17

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Распределение ресурсов по тактам для solution\_2

Все операции происходят за один такт.

#### Си и RTL моделирование: анализ результатов

Результаты, приведенные на рисунке 18, показывают, что синтезированное описание работает в соответствии с ожиданием – в информации указанно что при сравнение ожидаемого результата с фактическим ошибок не обнаружено, функция main вернула 0, следовательно тест успешно пройден.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Результаты моделирования Си и RTL для solution\_2

#### Си и RTL моделирование: отчет по быстродействию

Отчет по быстродействию, полученный при совместном Си и RTL моделировании приведен на рисунке 19. Из отчёта следует что задержка отсутствует.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Быстродействие при совместном моделировании для solution\_2

#### Си и RTL моделирование: временная диаграмма

Временная диаграмма совместного Си|RTL моделирования приведены на рисунке 20.

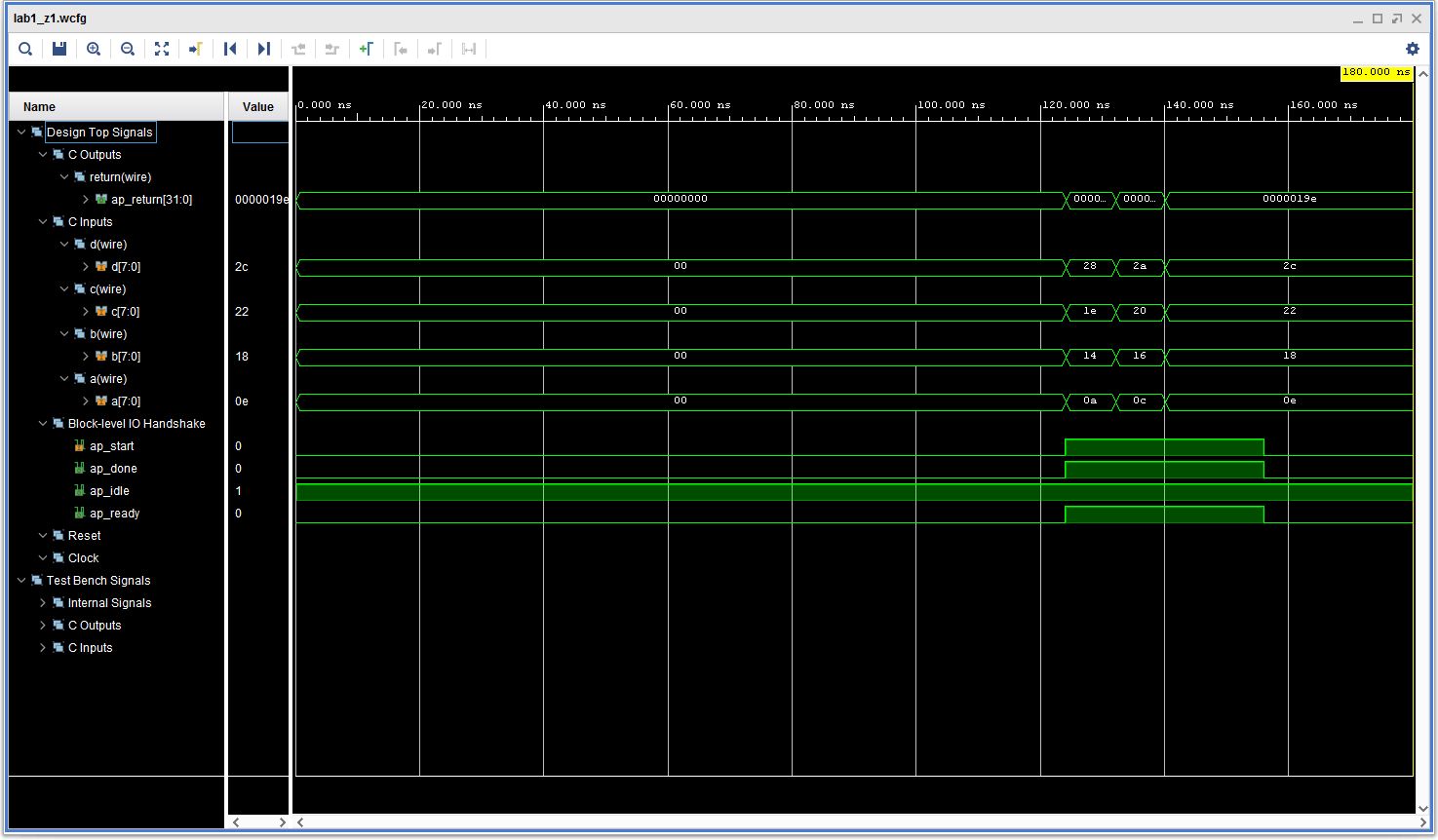


Рисунок - Временная диаграмма для solution\_2

На временной диаграмме для solution\_2 видно, что задержка отсутствует. Чтение и вычисление происходят в один такт.

## Вывод

Сравнение параметров двух решений

Результаты сравнения параметров двух решений приведены на рисунке 21.

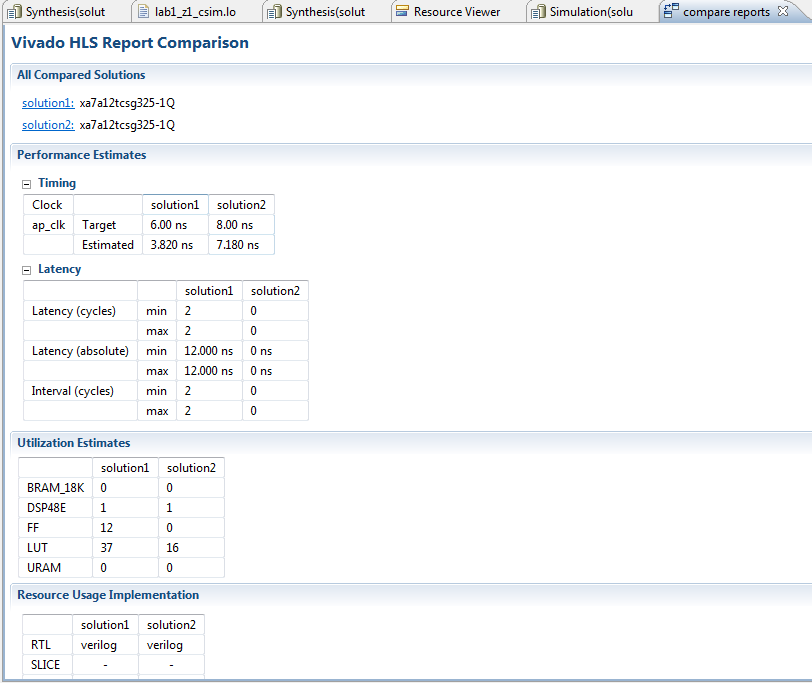


Рисунок - Сравнение двух решений

Анализ приведенных на рисунке 21 данных показывает, что: во solution2 отсутствуют latency. Так же solution2 имеет меньше аппаратных затрат.

### Пояснение различий в результатах планирования и распределения ресурсов

В solution2 в результатах планирования и распределения показывает всю работу в один такт в отличие от solution1, где каждый цикл работы занимает три такта.

### Пояснение различий во временных диаграммах моделирования для двух решений

Во временной диаграмме solution2 отсутствует тактовый сигнал. Также на временной диаграмме solution2 видно, что чтение и вычисления данных происходит за один такт в отличие от solution1, где задержка занимает два такта и только на третий такт выдаётся результат.

### Ответы на поставленные в задании вопросы

Из двух решений быстрее даст результат решение с периодом 8 ns, так как в этом случае задержка отсутствует и вычисления происходят быстрее (это можно увидеть и на временной диаграмме).

## Часть 2 (lab1\_z2)

### Задание

Создали проект lab1\_z2 (рабочая папка – папка lab1\_z2 с заданием)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – Создание проекта

Подключили файл lab1\_z2.c (папка source)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок – Создание проекта

Подключили тест lab1\_z2\_test.c (папка source)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок – Создание проекта

Модифицировали тест – число, добавляемое на каждой итерации теста = порядковому номеру в списке группы.

Осуществили моделирование

Создали solution1

Задали: clock period 6; clock\_uncertainty 0.1

Осуществили синтез

Создали solution 2

### Исходный код функции

Исходный код синтезируемой функции приведен на рисунке 25

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Исходный код функции

В строке 1 Рисунок 1 объявляется функция с ее входными параметрами типа char и массивами int. Далее на второй строке объявляется переменные x, y типа integer для хранения результата. В строках 3–7 в цикле происходят математические операции с входными данными. В строке 4 переменной х присваивается значение массива in а в строке пять к y присваивается результат математических операций. В строке 6 в массив out записывается y.

### Исходный код теста

Исходный код теста для проверки функции lab1\_z2 (см. раздел 2), исправленный в соответствии с заданием, приведен на рисунке 26.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Исходный код теста

### Моделирование

Результаты моделирования исходного кода синтезируемой функции приведены на рисунке 27. Результаты моделирования показывают, что тест успешно пройден – консоль оповестила об успешном прохождение теста.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Результаты моделирования Си кода

### Первое решение (Solution 1)

#### Исходные настройки

* clock period: 6,
* clock uncertain: 0.1,
* part: xa7a12tcsg325-1q.

#### Синтез 1-го решения: оценка быстродействия

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на рисунке 28.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Оценка быстродействия для solution\_1

Target – планируемое время на один такт.

Estimated – оценочное время.

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

Latency (absolute) – время затраченное на latency.

Loop Latency – задержки во всём цикле.

Iteration Latency – задержка в одной итерации

Trip count – количество итераций.

#### Синтез 1-го решения: оценка аппаратных ресурсов

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на рисунке 29

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Аппаратные ресурсы для solution\_1

#### Синтез 1-го решения: планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 30.

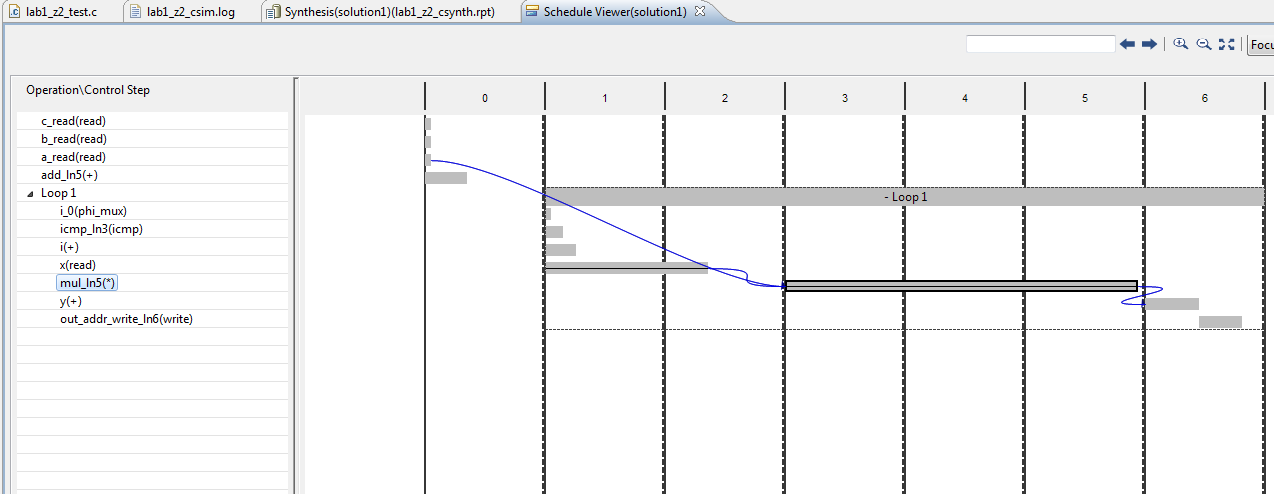


Рисунок - Планировщик для solution\_1

В планировщике видно, что чтение a, b, c и массива In происходит на нулевом такте. На первом такте запускается итерация цикла, там же считывается переменная цикла и x. На втором такте чтение x заканчивается. 3, 4, 5 такт производятся вычисление, 6 такте в y записывается полученный результат и запись y в массив out. В итоге вся итерация протекает в 6 тактов.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 31 - Performance Profile для solution1

#### Синтез 1-го решения: распределение ресурсов Resource Viewer

Распределение ресурсов по тактам приведено на рисунке 32

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Распределение ресурсов по тактам для solution\_1

На нулевом такте считывается a, b, c. На первом такте происходит прибавления к переменной массива. И на первом и втором такте происходит чтение из массива in. На 3, 4, 5 тактах происходит операция умножения. На 6 такте происходит операция сложения и запись результата в массив out.

#### Си и RTL моделирование: анализ результатов

Результаты, приведенные на рисунке 33, показывают, что синтезированное описание работает в соответствии с ожиданием – в информации указанно что при сравнение ожидаемого результата с фактическим ошибок не обнаружено, функция main вернула 0, следовательно тест успешно пройден.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Результаты моделирования Си и RTL для solution\_1

#### Си и RTL моделирование: отчет по быстродействию

Отчет по быстродействию, полученный при совместном Си и RTL моделировании приведен на рисунке 34.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Быстродействие при совместном моделировании solution\_1

#### Си и RTL моделирование: временная диаграмма

Временная диаграмма совместного Си|RTL моделирования приведены на рисунке 35.

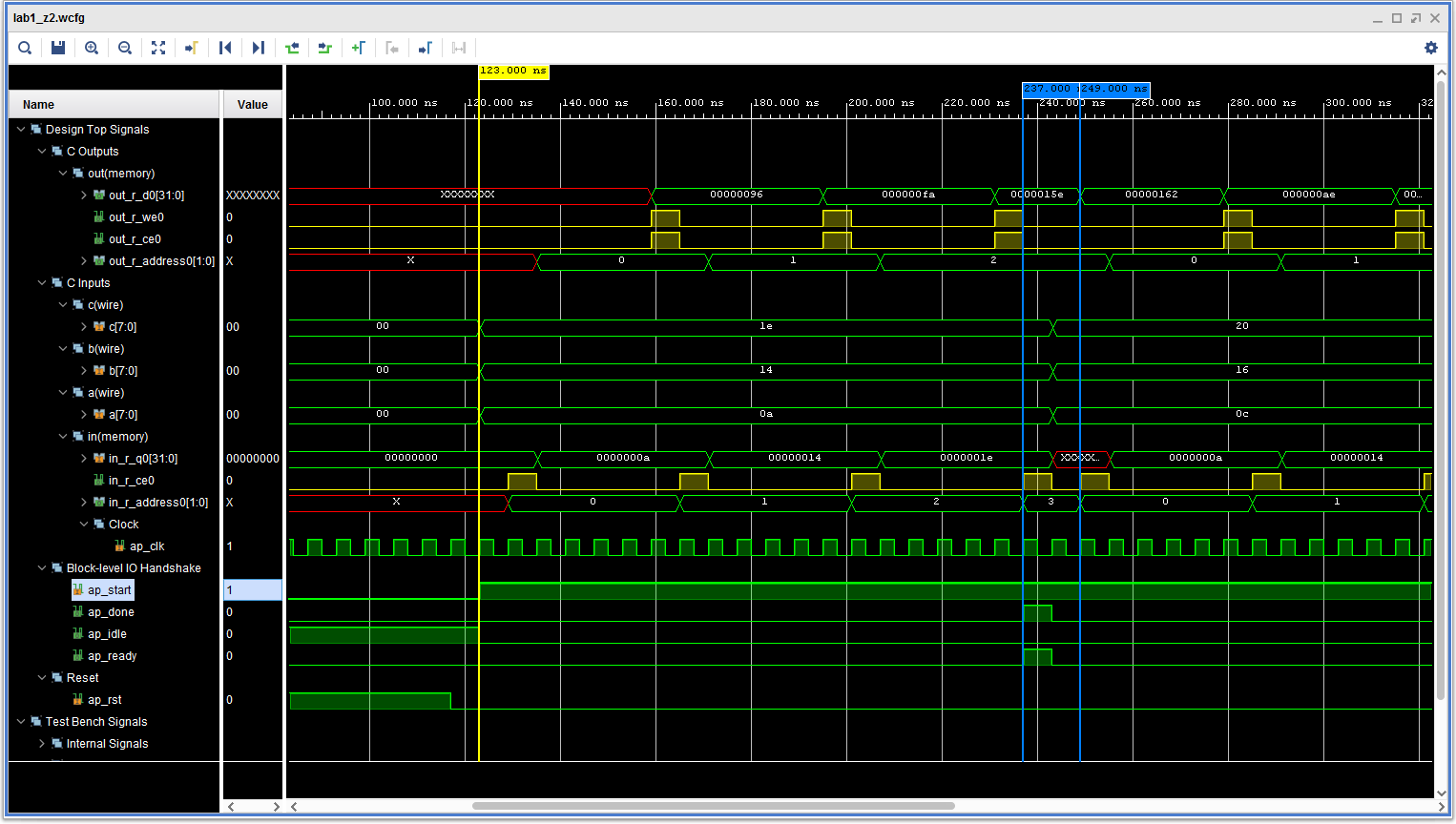


Рисунок - Временная диаграмма solution\_1

На временной диаграмме solution1 видно, что каждая итерация протекает 6 тактов. Весь цикл занимает 18 тактов плюс один такт на считывание a, b, c.

### Второе решение (Solution\_2)

#### Исходные настройки

* clock period: 12,
* clock uncertain: 0.1,
* part: xa7a12tcsg325-1q.

#### Синтез 2-го решения: оценка быстродействия

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на рисунке 36.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Оценка быстродействия для solution\_2

Target – планируемое время на один такт.

Estimated – оценочное время.

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

Latency (absolute) – время затраченное на latency.

Loop Latency – задержки во всём цикле.

Iteration Latency – задержка в одной итерации

Trip count – количество итераций.

#### Синтез 2-го решения: оценка аппаратных ресурсов

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на рисунке 37.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Аппаратные ресурсы для solution\_2

#### Синтез 2-го решения: планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 38.

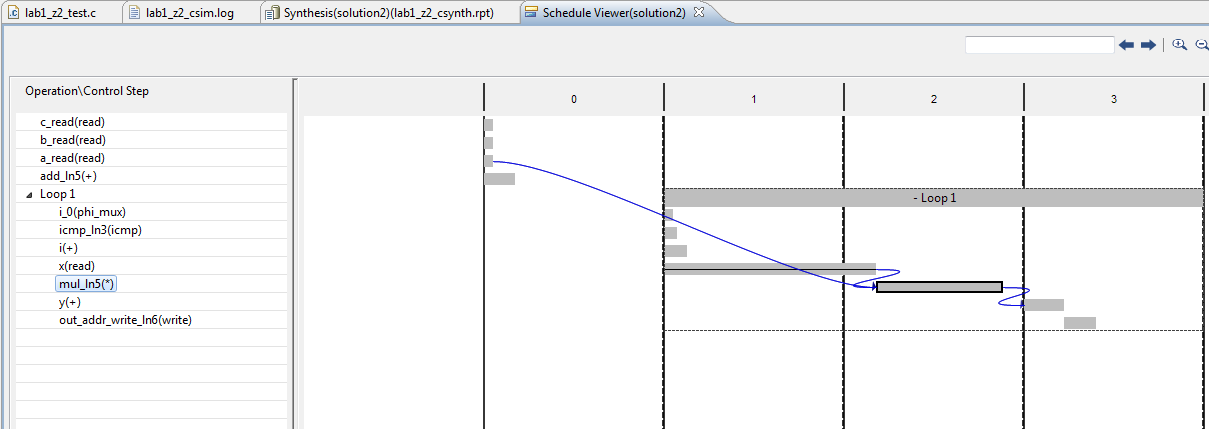


Рисунок - Планировщик для solution\_2

В планировщике видно, что на нулевом такте считывается a, b, c и производится операция сложения. На первом такте начинается итерация цикла – в переменную цикла записывается значение и считывается x. X считывается до начала второго такта. Так же на втором такте производится умножение и на третьем такте результат записывается в y и далее на том же такте записывается в массив out.

#### Синтез 2-го решения: распределение ресурсов Resource Viewer

Распределение ресурсов по тактам приведено на рисунке 39.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Распределение ресурсов по тактам для solution\_2

В распределении ресурсов наглядно видно, что на нулевом такте происходит чтение a, b, c и сложение b и с. На первом такте считывается данные из массива in, к переменной цикла добавляется значение. Во втором такте так же считываются данные из массива In и происходит операция умножения. В третьем такте результат присваивается к y и далее y записывается в массив out.

#### Си и RTL моделирование: анализ результатов

Результаты, приведенные на рисунке 40, показывают, что синтезированное описание работает в соответствии с ожиданием – в информации указанно что при сравнение ожидаемого результата с фактическим ошибок не обнаружено, функция main вернула 0, следовательно тест успешно пройден.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Результаты моделирования Си и RTL для solution\_2

#### Си и RTL моделирование: отчет по быстродействию

Отчет по быстродействию, полученный при совместном Си и RTL моделировании приведен на рисунке 41. Из отчёта следует что за 11 тактов выполнения одной итерации, 10 тактов – latency.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок - Быстродействие при совместном моделировании для solution\_2

#### Си и RTL моделирование: временная диаграмма

Временная диаграмма совместного Си|RTL моделирования приведены на рисунке 42.

Изображение выглядит как текст, монитор, электроника, экран

Автоматически созданное описание

Рисунок - Временная диаграмма для solution\_2

На временной диаграмме solution2 видно, что каждая итерация протекает 3 такта. Весь цикл занимает 9 тактов плюс один такт на считывание a, b, c.

## Вывод

**Сравнение параметров двух решений**

Результаты сравнения параметров двух решений приведены на рисунке 43.

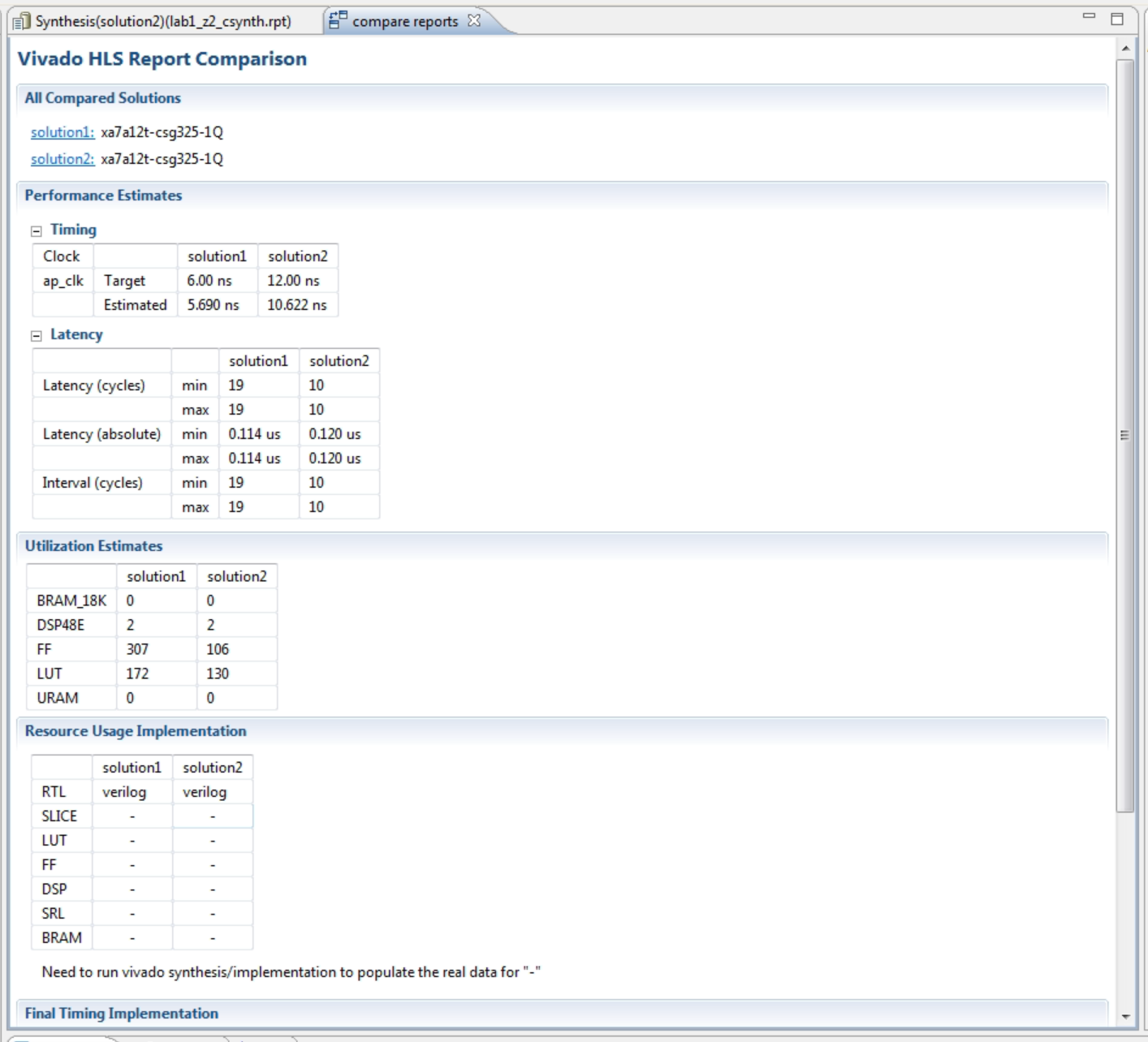


Рисунок - Сравнение двух решений

Один такт solution2 занимает 12 ns, а Solution1 – 6 ns. Предполагаемое время solution2 – составляет 10.622 ns, а solution2 – 5.690 ns. Latency всего цикла первого solution занимает 19 тактов по 6 таков на 3 итерации плюс такт на считывание. Latency solution2 всего цикла занимает 10 тактов, по три такта на каждую итерацию плюс один такт на считывание.