САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа lab1\_z1

Дисциплина:

«Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: Введение в Vivado HLS

Выполнил:

Дроздов Н. Д.

Группа: 3540901/02001

Преподаватель:

Антонов А. П.

Санкт-Петербург

2021

**Оглавление**

[1. Задание 5](#_Toc83211047)

[2. Исходный код функции 6](#_Toc83211048)

[3. Исходный код теста 6](#_Toc83211049)

[3.1 Моделирование 8](#_Toc83211050)

[4. Первое решение - Solution\_1 8](#_Toc83211051)

[4.1. Исходные настройки 8](#_Toc83211052)

[4.2. Синтез 8](#_Toc83211053)

[4.2.1 Оценка быстродействия 8](#_Toc83211054)

[4.2.2 Оценка аппаратных ресурсов 9](#_Toc83211055)

[4.2.3 Планировщик Schedule Viewer 10](#_Toc83211056)

[4.2.4 Распределение ресурсов Resource Viewer 10](#_Toc83211057)

[4.3 Си и RTL моделирование 10](#_Toc83211058)

[4.3.1 Анализ результатов 10](#_Toc83211059)

[4.3.2 Отчет по быстродействию 11](#_Toc83211060)

[4.3.3 Временная диаграмма 11](#_Toc83211061)

[5. Второе решение - Solution\_2 12](#_Toc83211062)

[5.1 Исходные настройки 12](#_Toc83211063)

[5.2 Синтез 12](#_Toc83211064)

[5.2.1 Оценка быстродействия 12](#_Toc83211065)

[5.2.2 Оценка аппаратных ресурсов 12](#_Toc83211066)

[5.2.3 Планировщик Schedule Viewer 13](#_Toc83211067)

[5.2.4 Распределение ресурсов Resource Viewer 13](#_Toc83211068)

[5.3 Си и RTL моделирование 14](#_Toc83211069)

[5.3.1 Анализ результатов 14](#_Toc83211070)

[5.3.2 Отчет по быстродействию 14](#_Toc83211071)

[5.3.3 Временная диаграмма 14](#_Toc83211072)

[6 Выводы 15](#_Toc83211073)

[Сравнение параметров двух решений 15](#_Toc83211074)

[Пояснение различий в результатах планирования и распределения ресурсов 16](#_Toc83211075)

[Пояснение различий во временных диаграммах моделирования для двух решений 16](#_Toc83211076)

[Ответы на поставленные в задании вопросы 16](#_Toc83211077)

**Список иллюстраций**

[Рисунок 1 Исходный код функции 6](#_Toc51272014)

[Рисунок 2 Исходный код теста 7](#_Toc51272015)

[Рисунок 3 Результаты моделирования Си кода 8](#_Toc51272016)

[Рисунок 4 Оценка быстродействия для solution\_1 9](#_Toc51272017)

[Рисунок 5 Аппаратные ресурсы для solution\_1 10](#_Toc51272018)

[Рисунок 6 Планировщик для solution\_1 10](#_Toc51272019)

[Рисунок 7 Распределение ресурсов по тактам для solution\_1 11](#_Toc51272020)

[Рисунок 8 Результаты моделирования Си и RTL для solution\_1 11](#_Toc51272021)

[Рисунок 9 Быстродействие при совместном моделировании solution\_1 12](#_Toc51272022)

[Рисунок 10 Временная диаграмма solution\_1 12](#_Toc51272023)

[Рисунок 11 Оценка быстродействия для solution\_2 13](#_Toc51272024)

[Рисунок 12 Аппаратные ресурсы для solution\_2 14](#_Toc51272025)

[Рисунок 13 Планировщик для solution\_2 14](#_Toc51272026)

[Рисунок 14 Распределение ресурсов по тактам для solution\_2 15](#_Toc51272027)

[Рисунок 15 Результаты моделирования Си и RTL для solution\_2 15](#_Toc51272028)

[Рисунок 16 Быстродействие при совместном моделировании для solution\_2 16](#_Toc51272029)

[Рисунок 17 Временная диаграмма для solution\_2 16](#_Toc51272030)

[Рисунок 18 Сравнение двух решений 17](#_Toc51272031)

1. Задание

* Создать проект lab1\_z1 (рабочая папка – папка lab1\_z1 с заданием)

Изображение выглядит как текст

Автоматически созданное описание

* Подключить файл lab1\_z1.c (папка source)

Изображение выглядит как стол

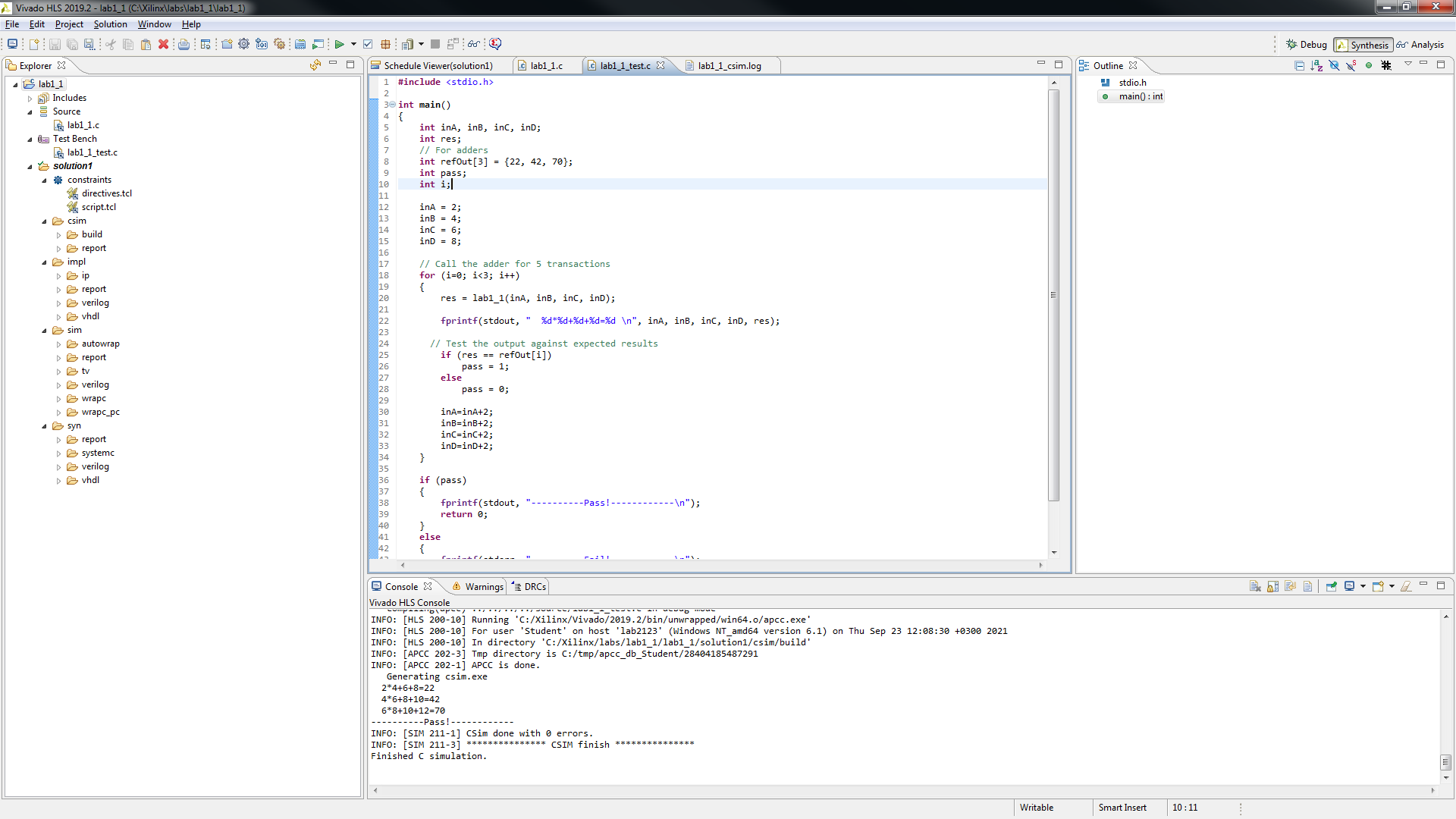
Автоматически созданное описание

* Подключить тест lab1\_z1\_test.c (папка source)

Изображение выглядит как стол

Автоматически созданное описание

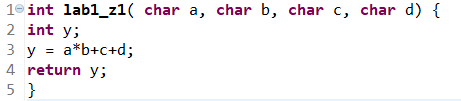
* Модифицировать тест – число, добавляемое на каждой итерации теста = порядковому номеру в списке группы.



* Осуществить моделирование
* Сделать solution1
* Задать: clock period 6; clock\_uncertainty 0.1
* Осуществить синтез
* Привести и пояснить в отчете:
* Таблицу performance estimates => Timing/summary и Latency/Summary
* Таблицу utilization estimates => summary
* Performance Profile
* Scheduler viewer (выполнить Zoom to Fit)
* На скриншоте показать Latency
* На скриншоте показать Initiation Interval
* Resource viewer (выполнить Zoom to Fit и развернуть все строки)
* Осуществить C|RTL моделирование (Dump trace – ALL)
* Привести скриншот Cosimulation report и дать пояснения
* Открыть временную диаграмму
* Отобразить все сигналы раздела Design Top Signal
* Отобразить 2 цикла работы на одном экране:
* На скриншоте показать Latency
* На скриншоте показать Initiation Interval
* Сделать solution2
* Задать: clock period 8; clock\_uncertainty 0.1
* Выполнить всю программу для Solution1
* В разделе Выводы отчета привести
* Сравнение Timing, Latency, Utilization Estimation (используя Compare reports) и пояснить отличия
* Пояснения отличий scheduler viewer и resource viewer двух решений
* Пояснения отличий временных диаграмм двух решений
* Ответить на вопрос: какое из двух решений сформирует на выходе результат за меньшее время (в ns)? Пояснить почему Вы так считаете, и в чем причина.

1. Исходный код функции

Исходный код синтезируемой функции приведен на рисунке Рисунок 1



*Рисунок 1 - 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.

1. Исходный код теста

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

* Проверку результата функции **lab1\_z1**

В строках 5–15 Рисунок 2 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.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 2 Исходный код теста*

1. Моделирование

Результаты моделирования исходного кода синтезируемой функции приведены на Рисунок 3. 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 3 Результаты моделирования Си кода*

1. Первое решение - Solution\_1
2. Исходные настройки

При создании первого решения задаются следующие настройки:

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

1. Синтез
   * 1. Оценка быстродействия

Результаты оценки быстродействия Timing/Summary и Latency/Summary приведены на Рисунок 4.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 4 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.
  + 1. Оценка аппаратных ресурсов

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на Рисунок 5

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 5 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.

* + 1. Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на Рис. 6

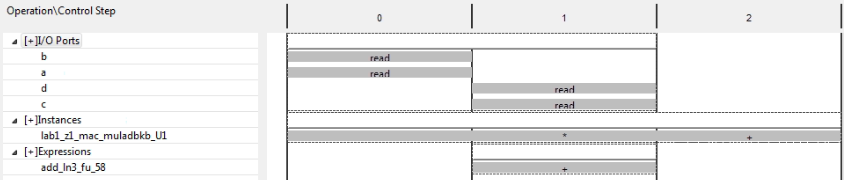


*Рисунок 6 Планировщик для solution\_1*

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.

* + 1. Распределение ресурсов Resource Viewer

Распределение ресурсов по тактам приведено на Рисунок 7



*Рисунок 7 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.

* 1. Си и RTL моделирование
     1. Анализ результатов

Результаты, приведенные на Рисунок 8, 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 8 Результаты моделирования Си и RTL для solution\_1*

* + 1. Отчет по быстродействию

Отчет по быстродействию, полученный при совместном Си и RTL моделировании приведен на Рисунок 9. Из отчёта следует что за три такта выполнения одной итерации, два такта занимает latency.

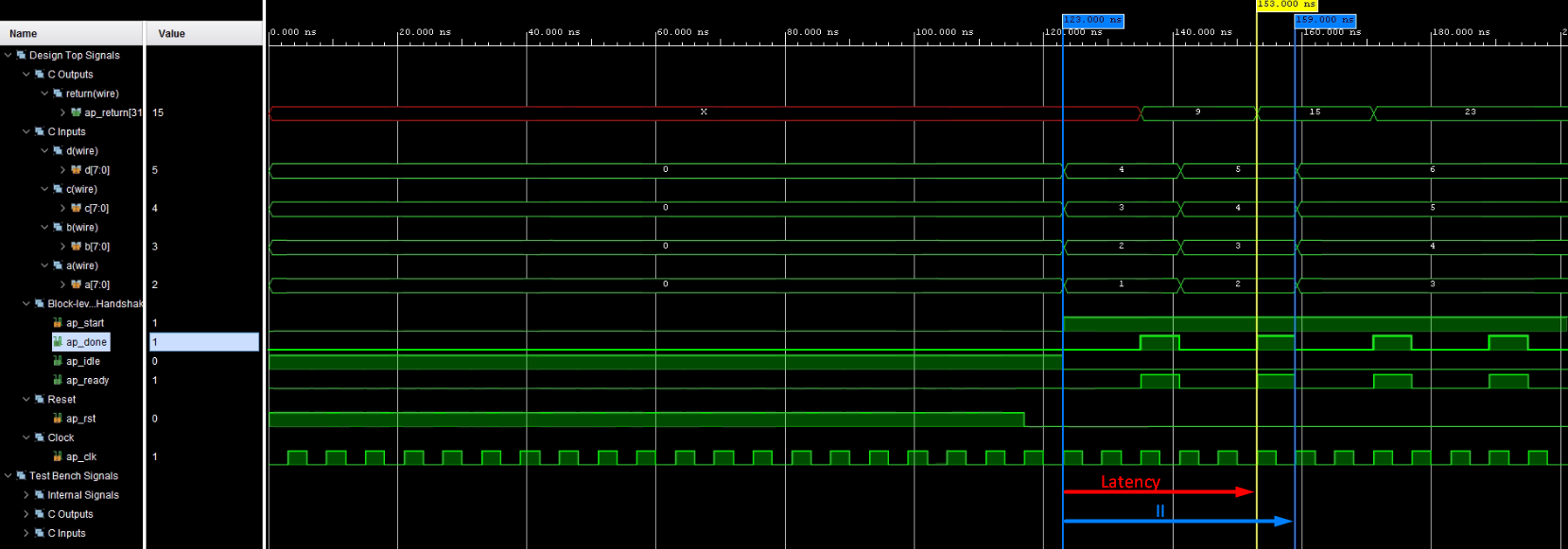
Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 9 Быстродействие при совместном моделировании solution\_1*

* + 1. Временная диаграмма

Временная диаграмма совместного Си|RTL моделирования приведены на Рисунок 10.



*Рисунок 10 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.

1. Второе решение - Solution\_2
   1. Исходные настройки

При создании первого решения задаются следующие настройки:

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

Результаты оценки быстродействия Timing/summary и Latency/Summary приведены на Рисунок 11.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 11 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.

* + 1. Оценка аппаратных ресурсов

Результаты оценки аппаратных ресурсов, требуемых для реализации синтезируемой функции, приведены на Рисунок 12

Изображение выглядит как стол

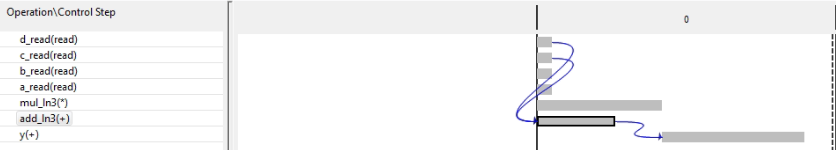
Автоматически созданное описание

*Рисунок 12 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.

* + 1. Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на Рис. 13



*Рисунок 13 Планировщик для solution\_2*

В планировщике видно, что все происходит за один такт.

* + 1. Распределение ресурсов Resource Viewer

Распределение ресурсов по тактам приведено на Рисунок 14

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 14 Распределение ресурсов по тактам для solution\_2*

Все операции происходят за один такт.

* 1. Си и RTL моделирование
     1. Анализ результатов

Результаты, приведенные на Рисунок 15, 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 15 Результаты моделирования Си и RTL для solution\_2*

* + 1. Отчет по быстродействию

Отчет по быстродействию, полученный при совместном Си и RTL моделировании приведен на Рисунок 16. Из отчёта следует что задержка отсутствует.

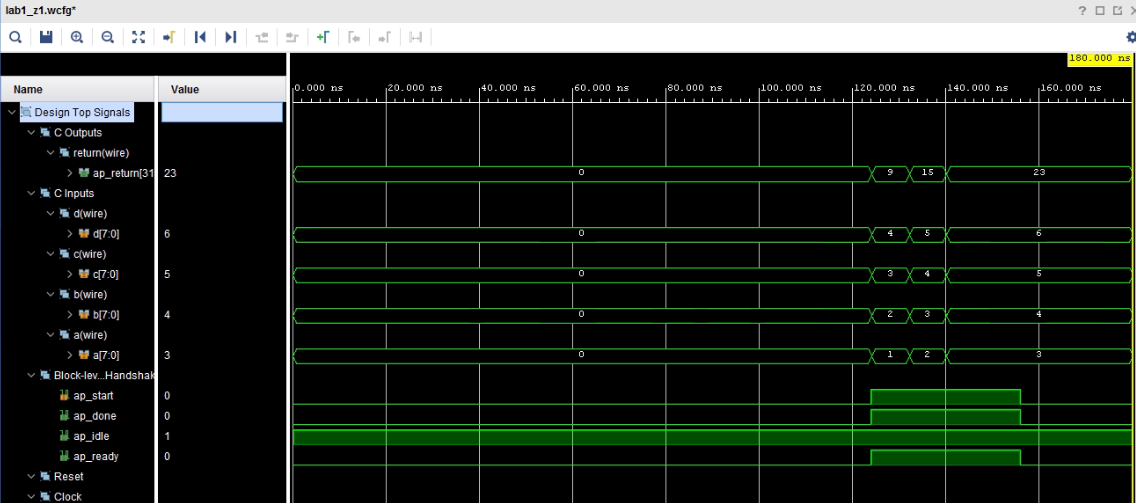
Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 16 Быстродействие при совместном моделировании для solution\_2*

* + 1. Временная диаграмма

Временная диаграмма совместного Си|RTL моделирования приведены на Рисунок 17.



*Рисунок 17 Временная диаграмма для solution\_2*

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

1. Выводы

Сравнение параметров двух решений

Результаты сравнения параметров двух решений приведены на Рисунок 18.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 18 Сравнение двух решений*

Анализ приведенных на Рис. 18 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).