Санкт-Петербургский Политехнический Университет Петра Великого

Институт Компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа 1\_1

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Введение в Vivado HLS

Задание 1

Студент: Селиверстов С.А.

Гр. № 3540901/81501

Преподаватель: Антонов А.П.

Санкт-Петербург

2019

Оглавление

[1. Задание 3](#_Toc21441187)

[2. Первое решение 5](#_Toc21441188)

[2.1. Моделирование 5](#_Toc21441189)

[2.2. Синтез 5](#_Toc21441190)

[2.3. C/RTL моделирование 7](#_Toc21441191)

[3. Второе решение 8](#_Toc21441192)

[3.1. Моделирование 8](#_Toc21441193)

[3.2. Синтез 9](#_Toc21441194)

[3.3. C\RTL моделирование 11](#_Toc21441195)

[4. Выводы 13](#_Toc21441196)

1. Задание

Создать проект, подключив готовые файлы исходного кода устройства и теста к нему. Исходный код:

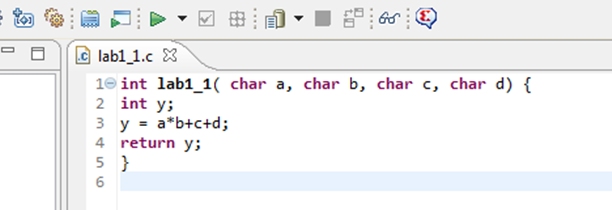


Рис. 1.1. Исходный код

Код теста:

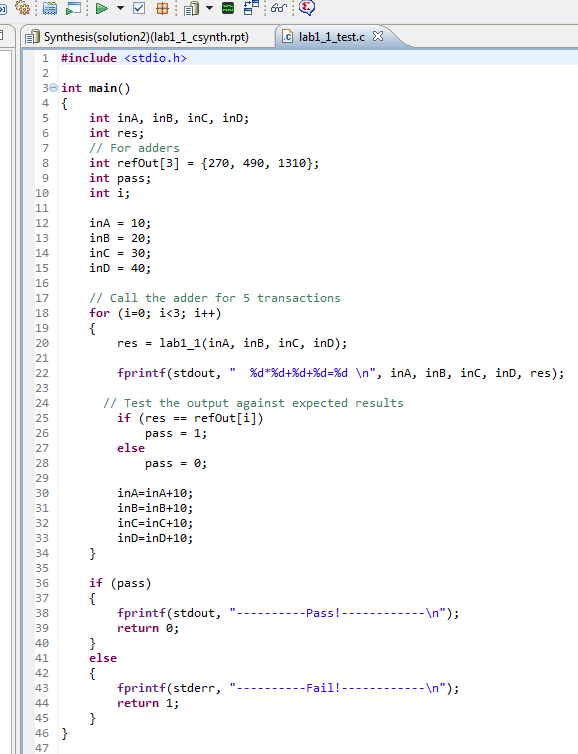


Рис. 1.2. Код теста

Создать 2 решения для синтеза устройства: задать clock period 6 и clock period 8, а также clock uncertain 0.1 в обоих случаях.

Для обоих решений осуществить моделирование и синтез, привести в данном отчете:

* performance estimates=>summary
* 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)
  + На скриншоте показать Latency
  + На скриншоте показать Initiation Interval

Также для обоих решений осуществить C|RTL моделирование:

* Отобразить два цикла обработки на одном экране
  + На скриншоте показать Latency
  + На скриншоте показать Initiation Interval

Написать выводы, где среди прочего объяснить отличие двух решений.

1. Первое решение
   1. Моделирование

При создании решения зададим настройки: clock period 6, clock uncertain 0.1, part xa7a12tcsg325-1q.

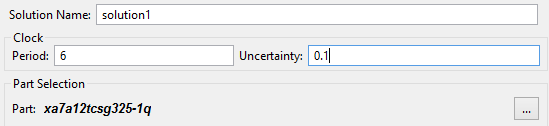


Рис. 2.1. Создание первого решения

При запуске моделирования можно увидеть, что тест успешно пройден:

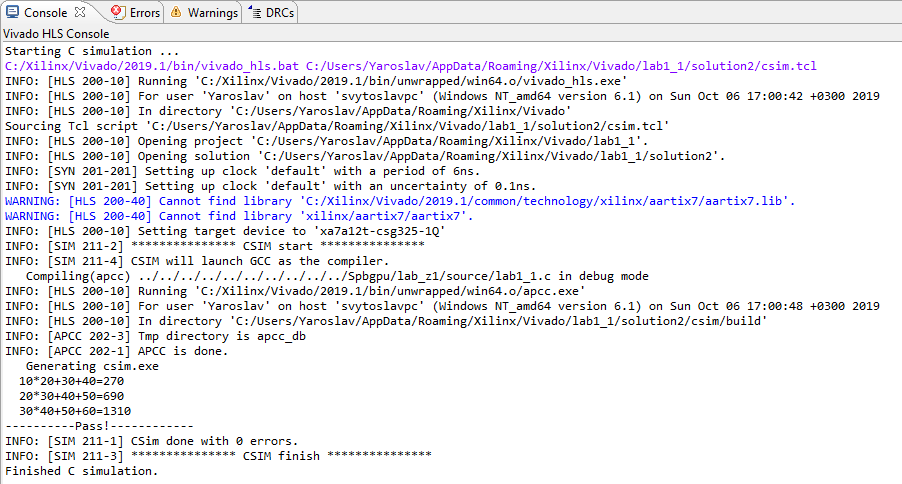


Рис. 2.2. Логи моделирования

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

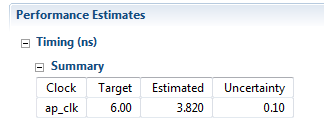


Рис. 2.3. Производительность

Здесь можно увидеть, что достигнутая задержка равна 3.820 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

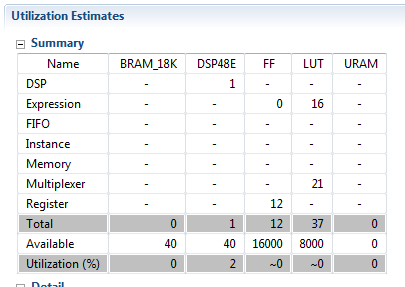


Рис. 2.4. Занимаемые ресурсы

Данный проект займет на микросхеме 1 DSP блок (в котором будут использованы и сумматоры и умножитель), 12 триггеров для хранения чисел, и 37 LUT.

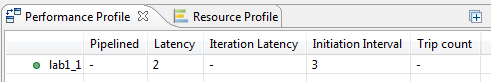


Рис. 2.5. Профиль производительности

На этом рисунке видно, что задержка получения выходного значения составляет 2 такта с момента старта, а задержка после старта до готовности приема новых данных – 3. Покажем эти интервалы на временной диаграмме:

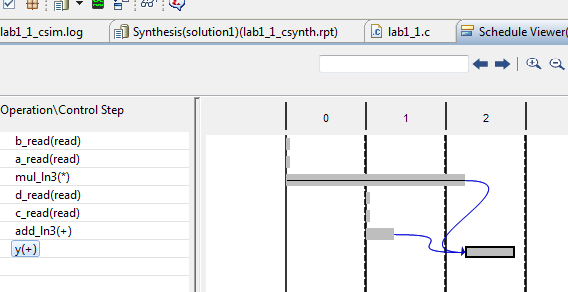


Рис. 2.6. Временная диаграмма

Здесь мы видим весь процесс получения результата. На первом такте происходит считывание операторов А и В, а также начинается их умножение. На втором считывание С и D, а также их сложение. На третьем результаты умножения и сложения складываются и получается ответ (latency = 2), а со следующего 4-го такта можно подавать следующие данные (II = 3).

Наконец покажем профиль ресурсов:

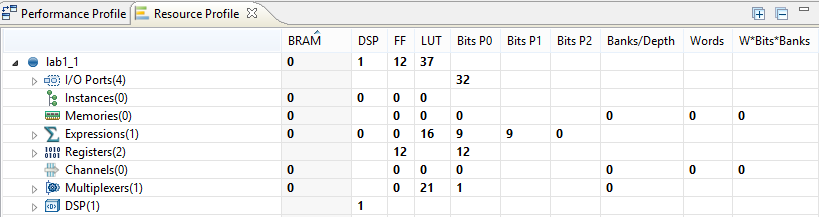


Рис. 2.7. Профиль ресурсов

Здесь можно увидеть те же числа, что и в отчете синтезатора.

* 1. C/RTL моделирование

При совместном моделировании, программа отобразила те же самые, ожидаемые нами значения Latency и II:

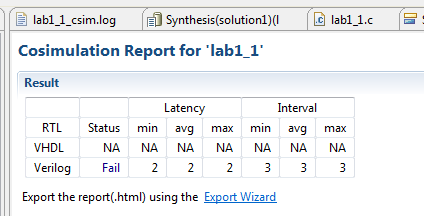


Рис.2.8. Результаты C\RTL моделирования

Покажем временную диаграмму совместного моделирования с отмеченными на ней Latency и II:

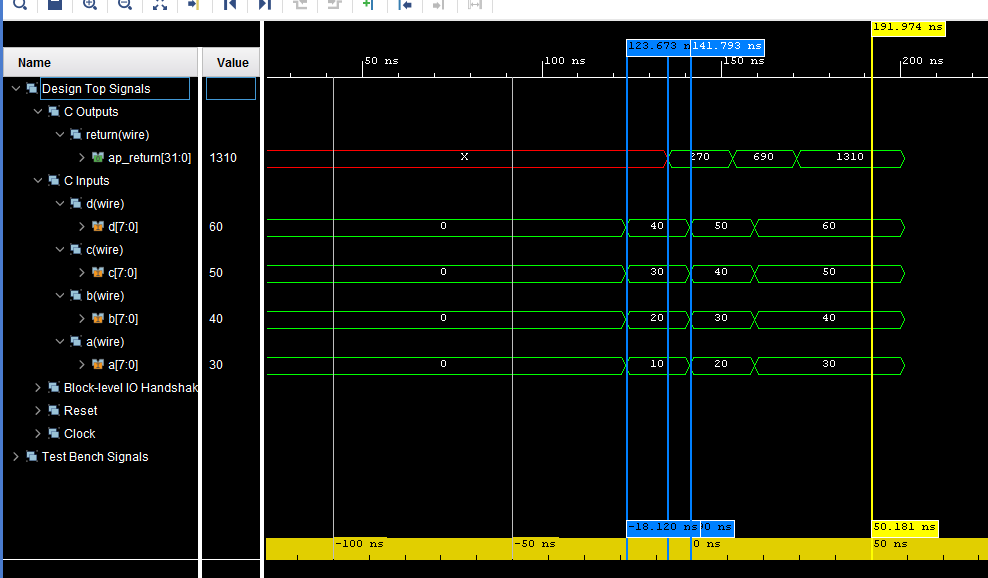


Рис. 2.9. Временная диаграмма совместного моделирования

1. Второе решение
   1. Моделирование

Создадим второе решение для данного проекта. Его настройки:

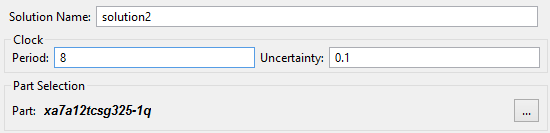


Рис. 3.1. Настройки нового решения

Настройки этого решения изменены согласно заданию.

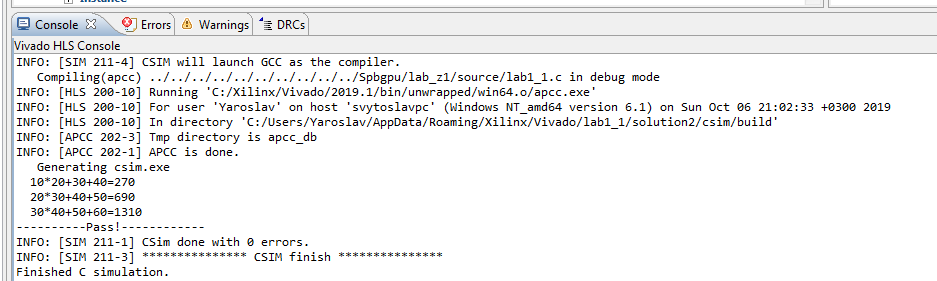


Рис. 3.2. Логи моделирования

Моделирования второго решения также прошло успешно.

* 1. Синтез

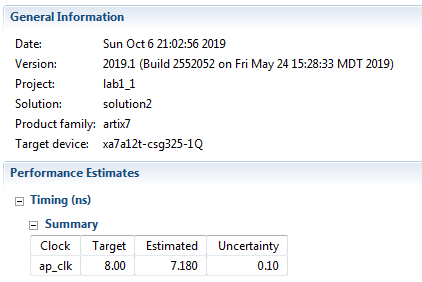


Рис. 3.3. Производительность

На этом рисунке видно, что данное решение также укладывается в установленный период тактовой частоты, однако задержки сильно отличаются, что означает, что изменения периода тактовой частоты повлекло за собой существенные изменения устройства.

На примере затрачиваемых на решение ресурсов это также хорошо видно:

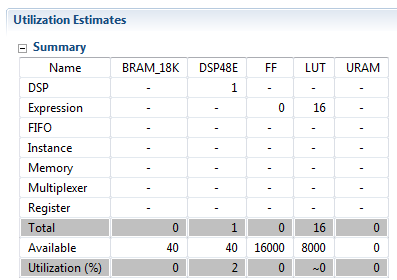


Рис. 3.4. Затрачиваемые ресурсы

В сравнении с предыдущим решением схема избавилась от всех триггеров и большинства LUT.

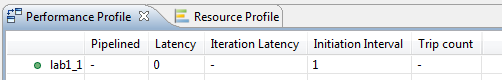


Рис. 3.5. Профиль производительности

Сразу видно, что задержка до получения ответа 0 тактов, а новые данные можно подавать уже через 1 такт. Покажем это на временной диаграмме:

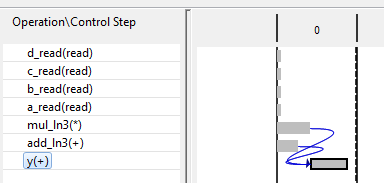


Рис. 3.6. Временная диаграмма

Если изучить эту диаграмму подробнее, то можно сказать, что считка всех 4-х операторов происходит одновременно, после чего сразу же начинаются умножение и первое сложение. Второе сложение начинается сразу после заверения умножения. Такой алгоритм позволяет получить ответ уже до начала второго такта.

Также можно сказать, что применен другой алгоритм умножения, который не требует задержки в 1 такт, и вообще весь процесс получения ответа является комбинаторным, что и объясняет нулевое значение параметра latency.

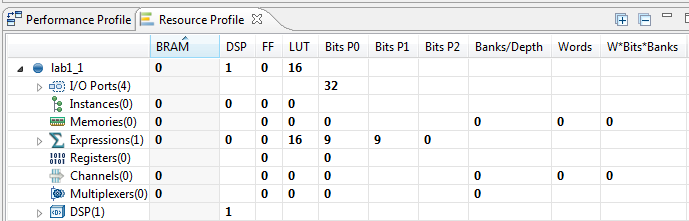


Рис. 3.7. Профиль ресурсов

Здесь мы также видим отличия, согласно общему отчету о затраченных ресурсах.

* 1. C\RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency, но II также стал нулевым:

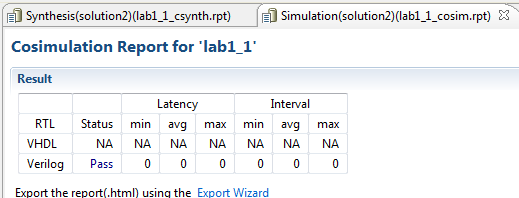


Рис. 3.8. C\RTL моделирование

Моделирование прошло успешно, однако программа предупредила, что оба значения выставлены в ноль, поскольку моделируемая схема – комбинаторная.

Покажем временную диаграмму совместного моделирования:

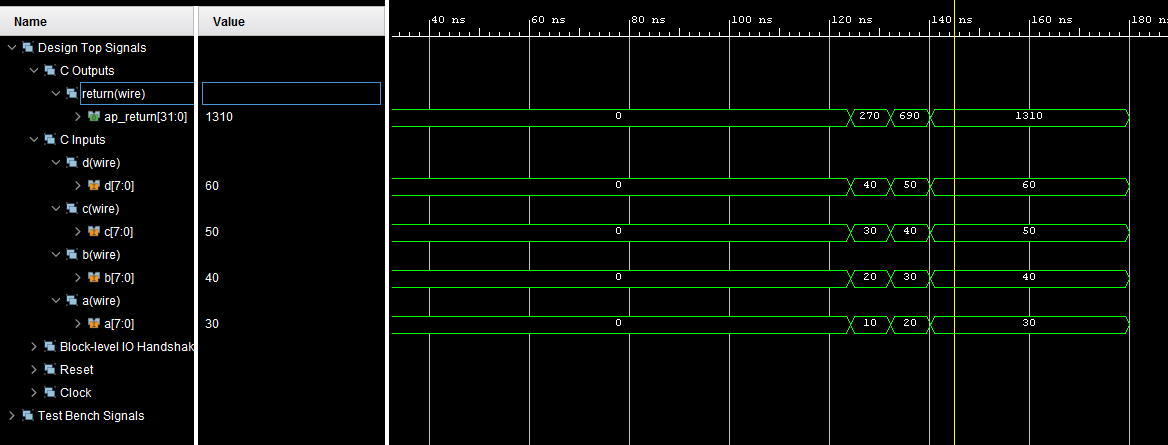


Рис. 3.9. Временная диаграмма совместного моделирования

Здесь видно отсутствия тактового сигнала и мгновенное (без учета задержек) получение ответа.

1. Выводы

В данной работу главным образом были рассмотрены различия, которые могут появится при синтезе устройства с один незначительным изменением периода тактовой частоты. В данном примере видно, что при больше периоде тактов, программа сумела вместить в 1 период чисто комбинаторную схему, в то время как при меньшем периоде это оказалось невозможным. Таким образом, получены 2 решения: первое – полный цикл выполнения 3 такта, а максимальная задержка обработки сигнала на такте составляет 3.820 нс, и второе – чисто комбинаторное решение, задержка в котором уже 7.180 нс, но все выполняется всего за 1 период тактового сигнала.