Санкт-Петербургский Политехнический Университет Петра Великого

Институт Компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа 3

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Block-Level I/O Protocols

Задание 1

Студент: Медведев М.А.

Онищук М.П.

Гр. № 3540901/81501,

3540901/81502

Преподаватель: Антонов А.П.

Санкт-Петербург

2019

Оглавление

[1. Задание 4](#_Toc25625164)

[2. Первое решение 6](#_Toc25625165)

[2.1. Моделирование 6](#_Toc25625166)

[2.2. Синтез 6](#_Toc25625167)

[2.3. C/RTL моделирование 8](#_Toc25625168)

[3. Второе решение 9](#_Toc25625169)

[3.1. Моделирование 9](#_Toc25625170)

[3.2. Синтез 10](#_Toc25625171)

[3.3. C\RTL моделирование 12](#_Toc25625172)

[4. Третье решение 13](#_Toc25625173)

[4.1. Моделирование 13](#_Toc25625174)

[4.2. Синтез 13](#_Toc25625175)

[4.3. C\RTL моделирование 15](#_Toc25625176)

[5. Выводы 16](#_Toc25625177)

1. Задание

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

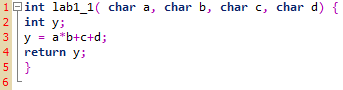


Рис. 1.1. Исходный код

Код теста:

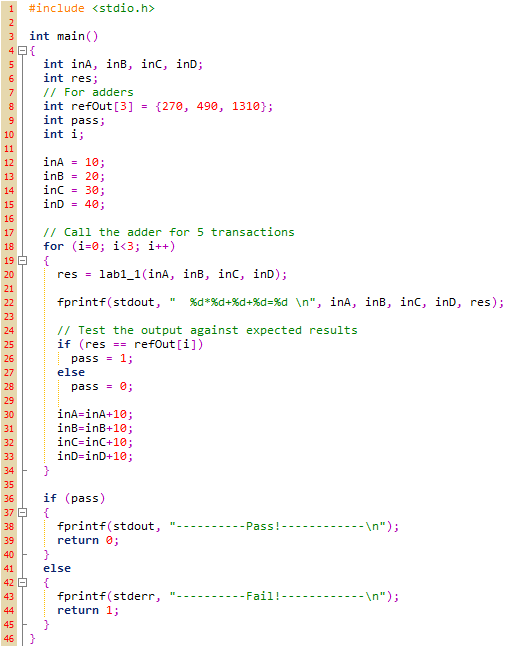


Рис. 1.2. Код теста

Создать 3 решения для синтеза устройства: задать clock period 6, 6 и 10, а также clock uncertain 0.1 во всех случаях.

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

* performance estimates=>summary
* utilization estimates=>summary
* Performance Profile
* interface estimates=>summary
  + объяснить какой интерфейс использован для блока (и какие сигналы входят) и для портов (и какие сигналы входят).
* 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 всех решений осуществить C|RTL моделирование:

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

Для решения №2 и №3 задать протокол (block-level): ap\_cntl\_chain и ap\_cntl\_none соответственно.

Написать выводы, где среди прочего объяснить отличие протоколов block\_level.

1. Первое решение
   1. Моделирование

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

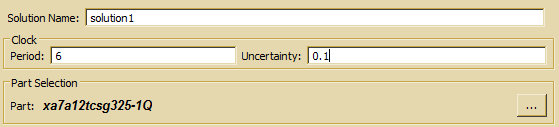


Рис. 2.1. Создание первого решения

При запуске моделирования можно увидеть, что тест успешно пройден:

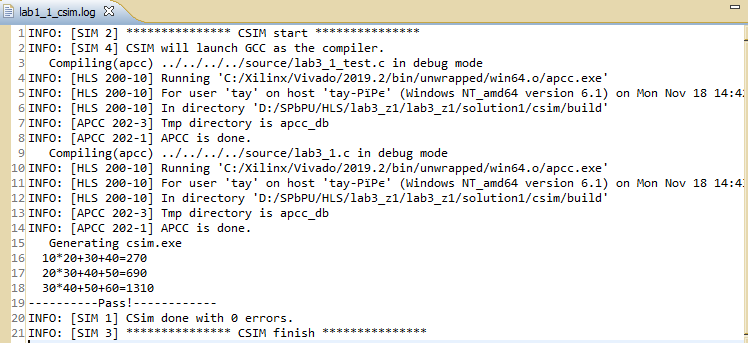


Рис. 2.2. Результаты моделирования

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

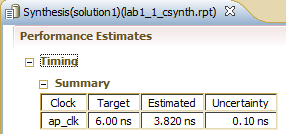


Рис. 2.3. Производительность

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

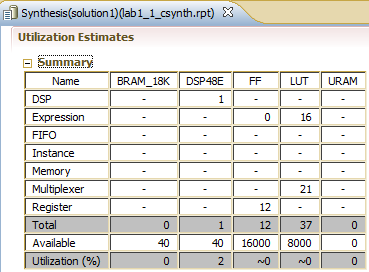


Рис. 2.4. Занимаемые ресурсы

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

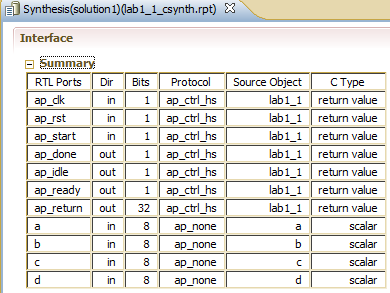


Рис. 2.5. Применяемые интерфейсы

На рисунке выше показаны интерфейсы, которые используются в синтезированном устройстве.

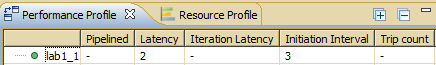


Рис. 2.6. Профиль производительности

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:

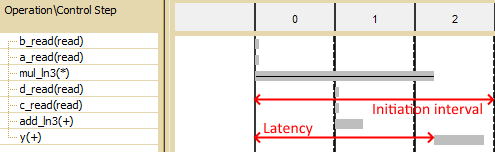


Рис. 2.7. Временная диаграмма

На первом такте происходит считывание операторов А и В, а также начинается их умножение. На втором считывание С и D, и 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:

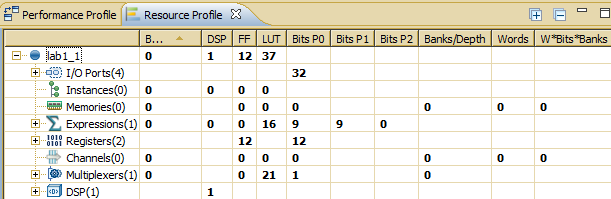


Рис. 2.8. Профиль ресурсов

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

* 1. C/RTL моделирование

При совместном моделировании (**Dump Trace = all**), программа отобразила те же самые, ожидаемые нами значения Latency и II:

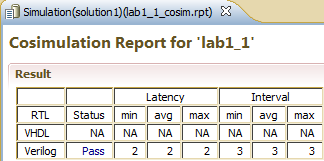


Рис.2.9. Результаты C\RTL моделирования

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

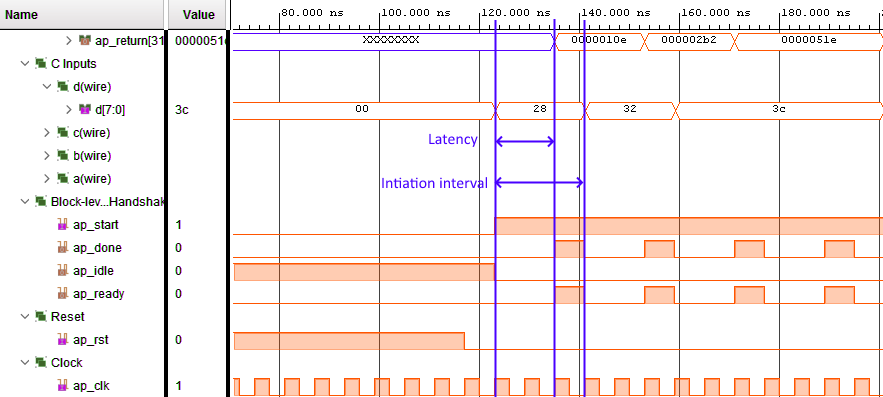


Рис. 2.10. Временная диаграмма совместного моделирования

1. Второе решение
   1. Моделирование

Создадим второе решение для данного проекта. Его настройки:

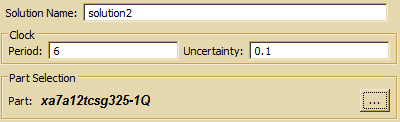


Рис. 3.1. Настройки нового решения

Добавим директиву, которая изменяет используемый block-level протокол.

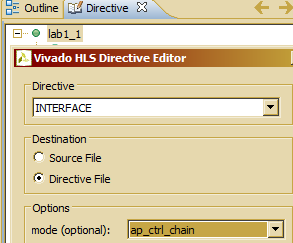


Рис. 3.2. Добавление директивы

При запуске моделирования можно увидеть, что тест успешно пройден:

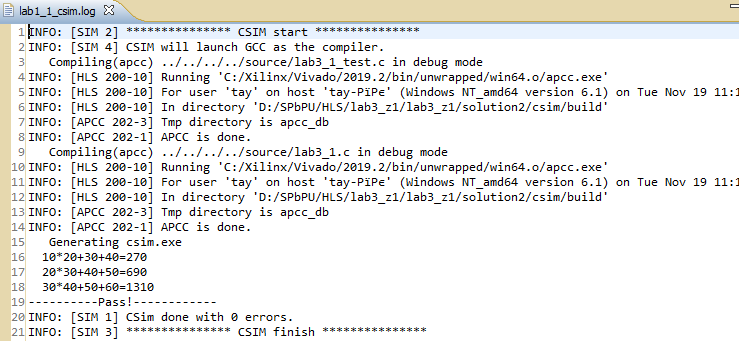


Рис. 3.3. Результаты моделирования

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

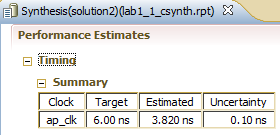


Рис. 3.4. Производительность

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

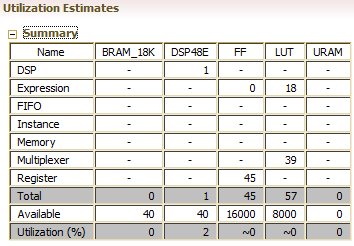


Рис. 3.5. Затрачиваемые ресурсы

Видно, что данный проект теперь займет на микросхеме 1 DSP блок, 45 регистров для хранения чисел, и 57 LUT.

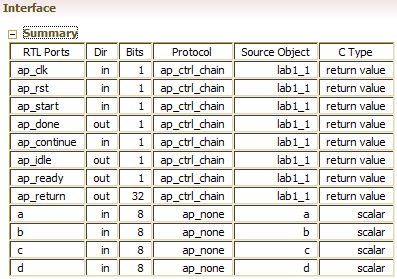


Рис. 3.6. Применяемые интерфейсы



Рис. 3.7. 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:

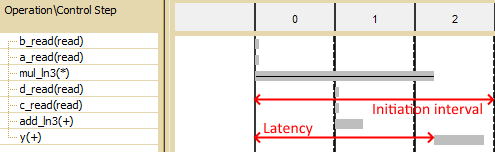


Рис. 3.8. Временная диаграмма

Последовательность работы соответствует решению 1.

Наконец покажем профиль ресурсов:

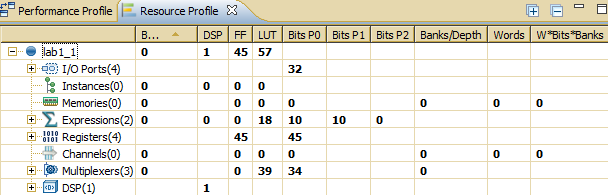


Рис. 3.9. Профиль ресурсов

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

* 1. C\RTL моделирование

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

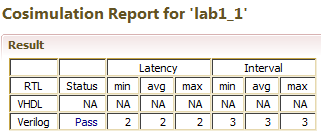


Рис. 3.10. C\RTL моделирование

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

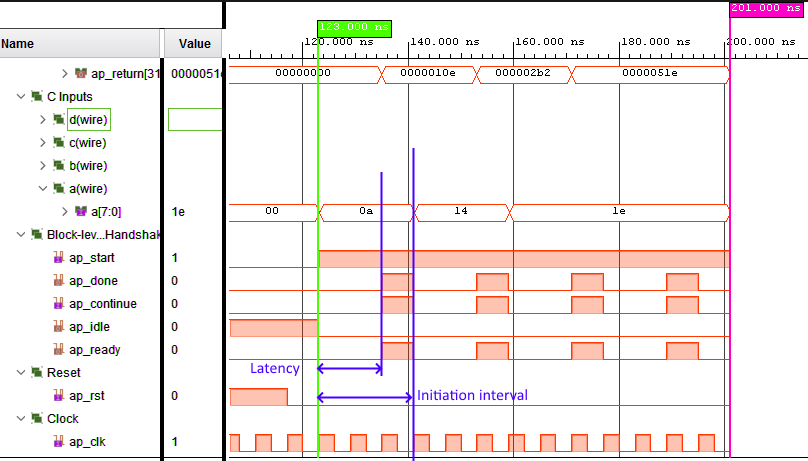


Рис. 3.11. Временная диаграмма совместного моделирования

Здесь видно соответствие результатам решения 1, однако, имеется дополнительный сигнал ap\_continue.

1. Третье решение
   1. Моделирование

Создадим третье решение для данного проекта. Его настройки:

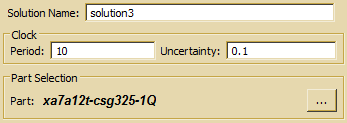


Рис. 4.1. Настройки нового решения

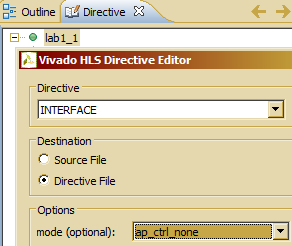


Рис. 4.2. Добавление директивы

Настройки этого решения изменены согласно заданию.

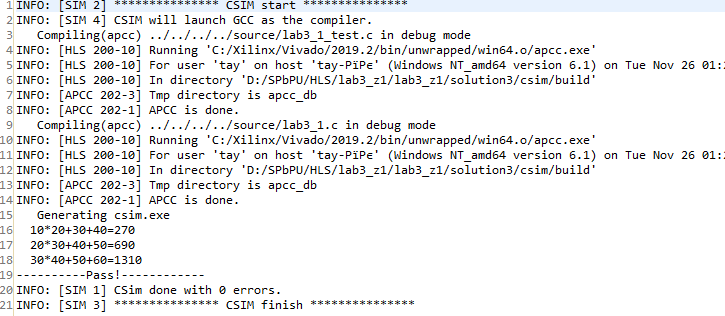


Рис. 4.3. Логи моделирования

Моделирования второго решения также прошло успешно.

* 1. Синтез

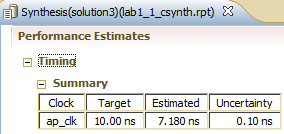


Рис. 4.4. Производительность

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:

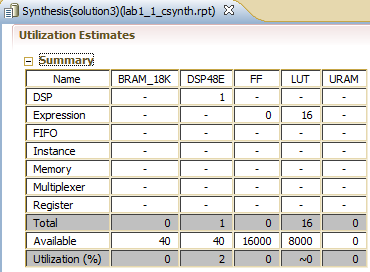


Рис. 4.5. Затрачиваемые ресурсы

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

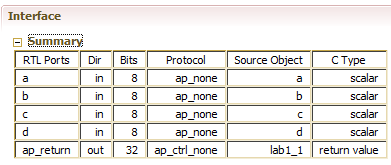


Рис. 4.6. Применяемые интерфейсы



Рис. 4.7. 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:

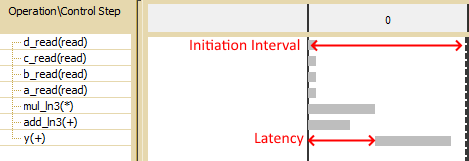


Рис. 4.8. Временная диаграмма

Считывание всех 4-х операторов происходит одновременно, после чего сразу же начинаются умножение и первое сложение. Второе сложение начинается сразу после завершения 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.

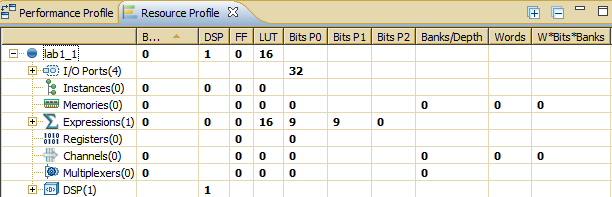


Рис. 4.9. Профиль ресурсов

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 Initiation Interval также стал нулевым:

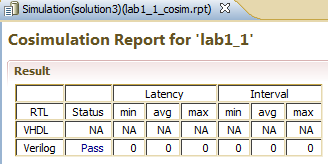


Рис. 4.10. C\RTL моделирование

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

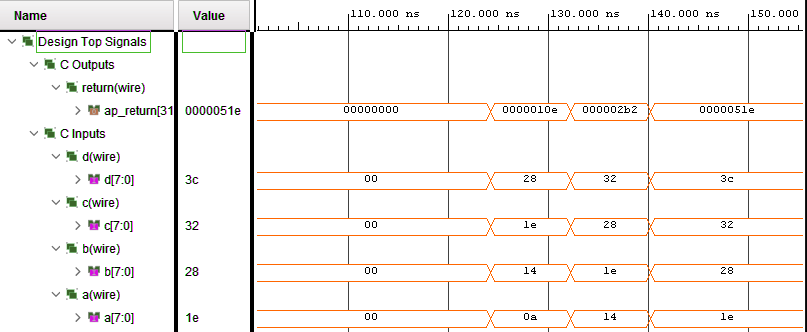


Рис. 4.11. Временная диаграмма совместного моделирования

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

1. Выводы

В данной работе были рассмотрены различия, которые могут появиться при синтезе устройства с незначительным изменением периода тактовой частоты и применением различных block-level протоколов. Протокол данного типа изменялся вручную для возвращаемого значения функции. По-умолчанию используется тип ap\_ctrl\_hs. Тип ap\_ctrl\_chain похож на ap\_ctrl\_hs, но имеет дополнительный входной порт ap\_continue. Режим ap\_ctrl\_none реализует моделирование без block-level протокола ввода-вывода – была получена чисто комбинаторная схема.

В результате получены 3 решения: первое и второе – полный цикл выполнения 3 такта, а максимальная задержка обработки сигнала на такте составляет 3.820 нс, и третье – чисто комбинаторное решение, задержка в котором уже 7.180 нс, но все выполняется всего за 1 период тактового сигнала.