Санкт-Петербургский Политехнический Университет Петра Великого

Институт Компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа 6

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Port-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](#_Toc26466107)

[2. Первое решение 6](#_Toc26466108)

[2.1. Моделирование 6](#_Toc26466109)

[2.2. Синтез 7](#_Toc26466110)

[2.3. C/RTL моделирование 9](#_Toc26466111)

[3. Второе решение 10](#_Toc26466112)

[3.1. Моделирование 10](#_Toc26466113)

[3.2. Синтез 11](#_Toc26466114)

[3.3. C\RTL моделирование 13](#_Toc26466115)

[4. Выводы 14](#_Toc26466116)

1. Задание

* Создать проект lab6\_1
* Микросхема: xa7a12tcsg325-1q
* Создать Си код на основе слайда (функция foo)
* Создать тест lab6\_1\_test.c на основе слайда выше.
* Сделать solution1
  + задать: clock period 6; clock\_uncertainty 0.1
  + осуществить моделирование (на основе слайда выше, с выводом результатов в консоль)
  + осуществить синтез (с настройками по умолчанию – интерфейс ap-fifo)
    - привести в 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
  + Осуществить C|RTL моделирование
    - Привести результаты из консоли
    - Открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
* Сделать solution2
  + Задать протокол
    - a: ap\_bus
  + осуществить моделирование
  + осуществить синтез
    - привести в 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
  + Осуществить C|RTL моделирование
    - Привести результаты из консоли
    - Открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
* Выводы

Объяснить отличие протоколов.

1. Первое решение
   1. Моделирование

Перед началом работы была создана функция и тест для неё следующего вида:

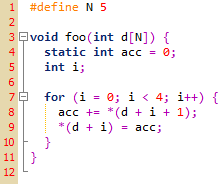


Рис. 2.1. Исходный код синтезируемой функции

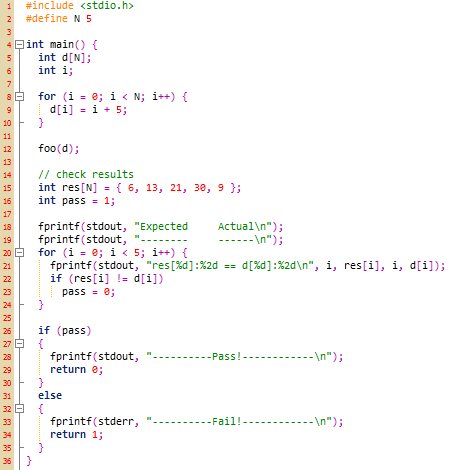


Рис. 2.2. Исходный код теста

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

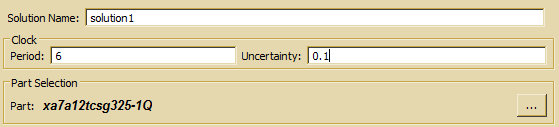


Рис. 2.3. Создание первого решения

При запуске моделирования можно увидеть, что тест успешно пройден:

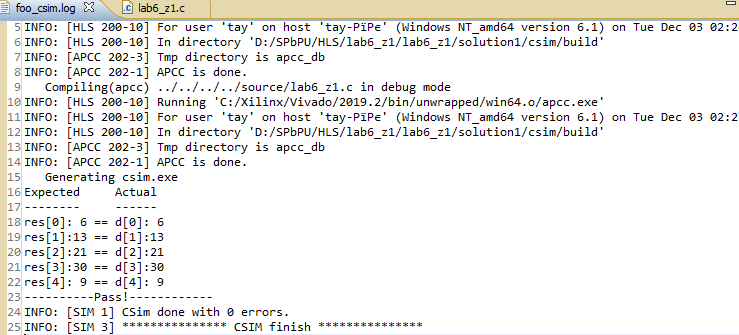


Рис. 2.4. Результаты моделирования

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

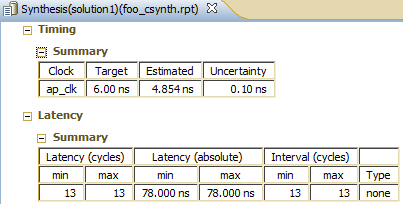


Рис. 2.5. Производительность

Здесь можно увидеть, что достигнутая задержка равна 4.854 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

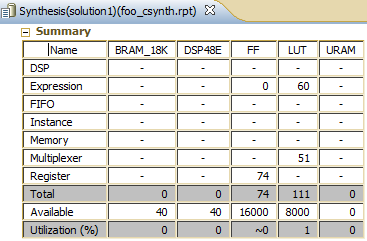


Рис. 2.6. Занимаемые ресурсы

Данный проект займет на микросхеме 74 регистра для хранения чисел, и 111 LUT.

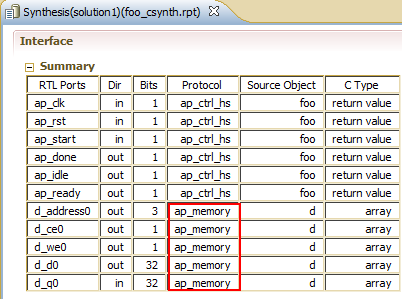


Рис. 2.7. Применяемые интерфейсы

На рисунке выше показаны интерфейсы, которые используются в синтезированном устройстве. Видно, что по-умолчанию применяется протокол **ap\_memory**.

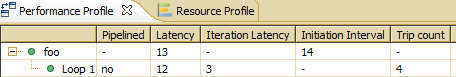


Рис. 2.8. Профиль производительности

На этом рисунке видно, что задержка получения первого выходного значения составляет 3 такта с момента старта (всех данных – 13), а задержка после старта до готовности приема новых данных – 14:

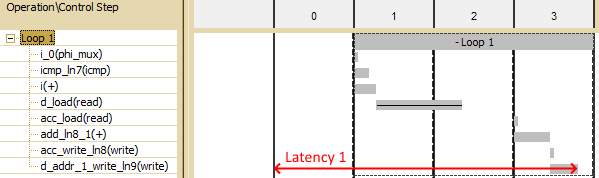


Рис. 2.9. Временная диаграмма

На первом такте происходит подготовка к запуску цикла, вероятно. Далее на первом такте каждой итерации цикла инициализация переменной-счетчика, проверка условия завершения, увеличение счётчика и начало считывания данных из массива D. На втором – завершение считывания данных из массива D. На третьем – чтение сохраненного значения аккумулятора, прибавление к нему значение элемента, прочитанного из D, запись результата обратно в аккумулятор и в массив D.

Покажем профиль ресурсов:

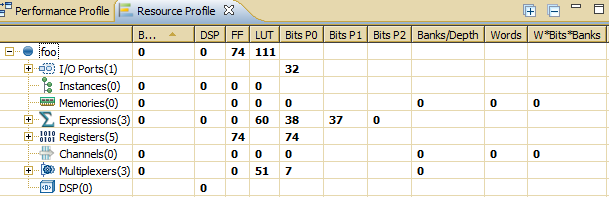


Рис. 2.10. Профиль ресурсов

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:

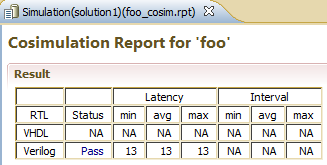


Рис.2.9. Результаты C\RTL моделирования

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

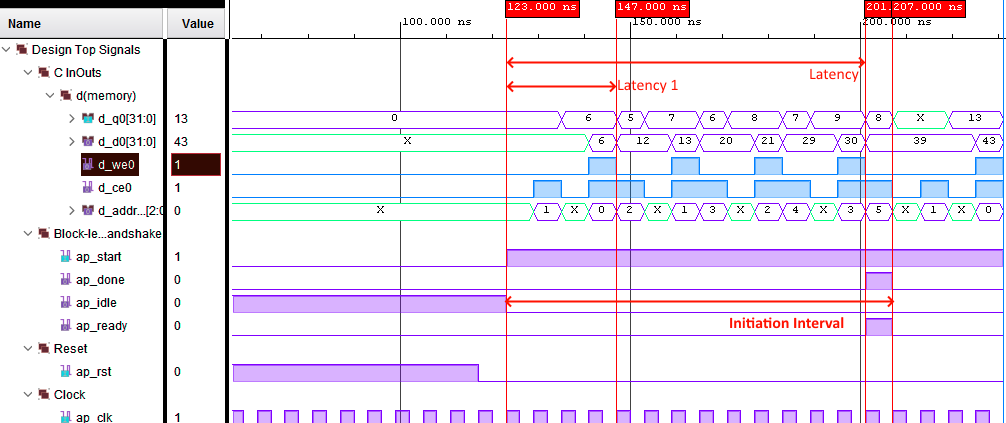


Рис. 2.10. Временная диаграмма совместного моделирования

1. Второе решение
   1. Моделирование

Создадим второе решение для данного проекта. Его настройки:

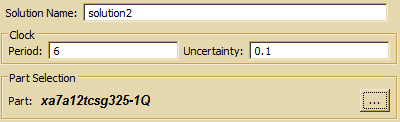


Рис. 3.1. Настройки нового решения

Добавим директиву, которая изменяет используемый port-level протокол.

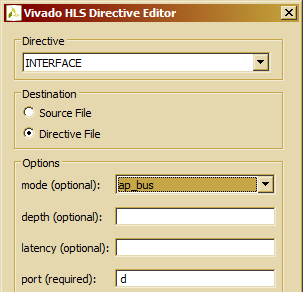


Рис. 3.2. Добавление директивы

При запуске моделирования можно увидеть, что тест успешно пройден:

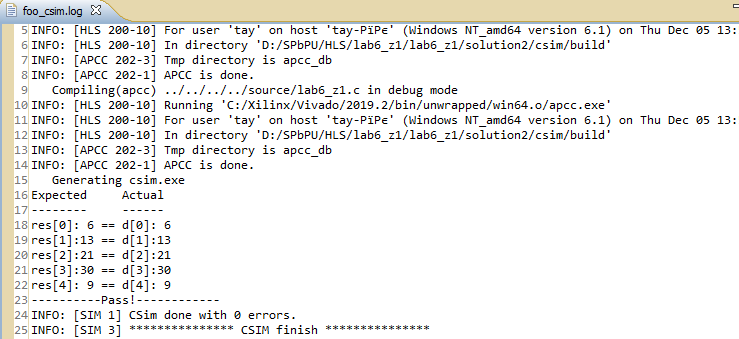


Рис. 3.3. Результаты моделирования

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

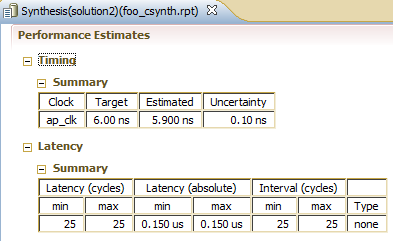


Рис. 3.4. Производительность

Здесь можно увидеть, что достигнутая задержка равна 5.900 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

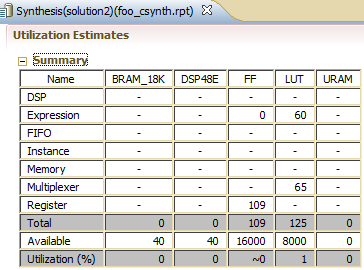


Рис. 3.5. Затрачиваемые ресурсы

Видно, что данный проект теперь займет на микросхеме 109 регистров для хранения чисел, и 125 LUT.

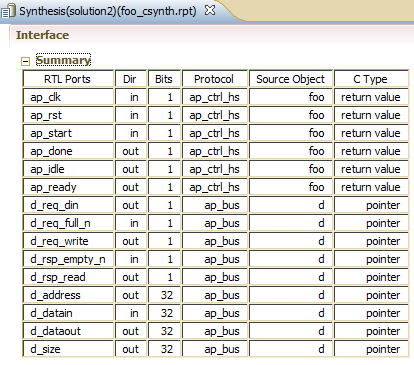


Рис. 3.6. Применяемые интерфейсы

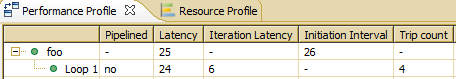


Рис. 3.7. Профиль производительности

На этом рисунке видно, что задержка получения первого выходного значения составляет 7 тактов с момента старта (25 для всех), а задержка после старта до готовности приема новых данных – 26:

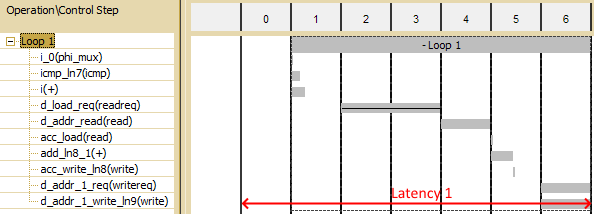


Рис. 3.8. Временная диаграмма

Последовательность работы в первом приближении соответствует решению 1, однако, видно, что операции чтения и записи в массив D теперь выполняются отдельно от всех остальных операций, на что тратится больше тактов.

Наконец покажем профиль ресурсов:

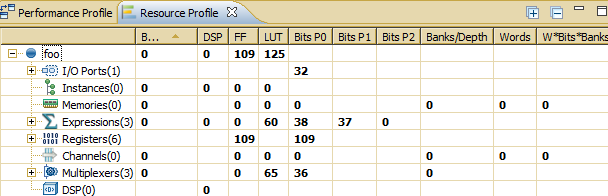


Рис. 3.9. Профиль ресурсов

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:

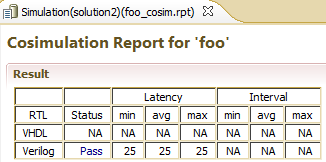


Рис. 3.10. C\RTL моделирование

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

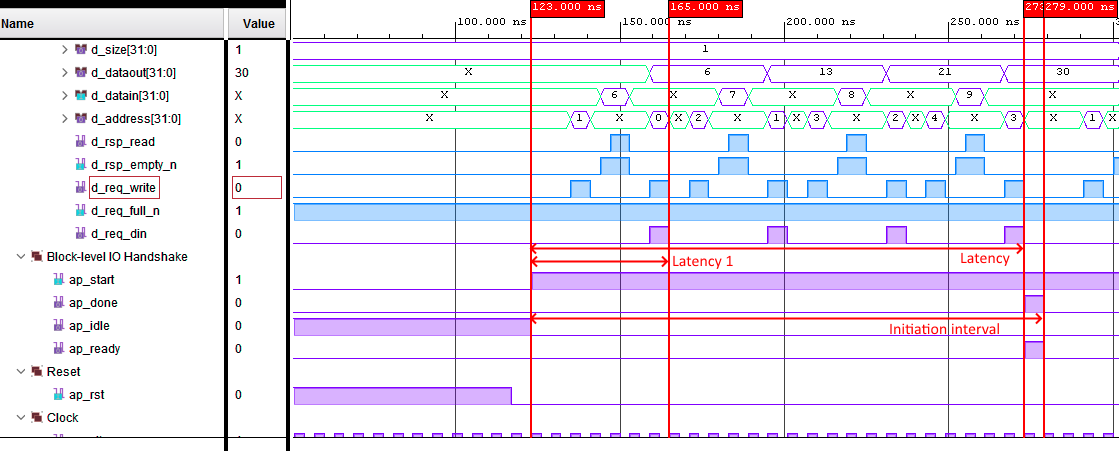


Рис. 3.11. Временная диаграмма совместного моделирования

Здесь также видны отличия во времени выполнения итераций и протоколе работы с массивом D.

1. Выводы

В данной работе были рассмотрены различия, которые могут появиться при синтезе устройства с применением различных block-level протоколов (ap\_memory и ap\_bus). По-умолчанию для массивов используется протокол типа тип ap\_memory. Протокол типа ap\_bus реализует переменные указателя и передачи по ссылке в виде шины общего назначения.

В результате получены 2 решения: первое – полный цикл выполнения 6 тактов, а максимальная задержка обработки сигнала на такте составляет 4.854 нс, и второе – полный цикл выполнения тоже 6 тактов, но задержка уже 5.900 нс и используется протокол ap\_bus.