Санкт-Петербургский Политехнический Университет Петра Великого

Институт Компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа 4

Предмет: «Проектирование реконфигурируемых гибридных вычислительных систем»

**Тема: «Port-level IO protocols»**

Студент: Медведев М.А.

Онищук М.П.

Гр. № 3540901/81501,

3540901/81502

Преподаватель: Антонов А.П.

Санкт-Петербург

2019

Оглавление

[1. Задание 5](#_Toc26917356)

[2. Первое решение 7](#_Toc26917357)

[2.1. Моделирование 7](#_Toc26917358)

[2.2. Синтез 8](#_Toc26917359)

[2.3. C/RTL моделирование 10](#_Toc26917360)

[3. Второе решение 11](#_Toc26917361)

[3.1. Добавление директив 11](#_Toc26917362)

[3.2. Моделирование 12](#_Toc26917363)

[3.3. Производительность 12](#_Toc26917364)

[3.4. Использование ресурсов 12](#_Toc26917365)

[3.5. Интерфейсы 14](#_Toc26917366)

[3.6. С/RTL моделирование 15](#_Toc26917367)

[4. Выводы 15](#_Toc26917368)

1. Задание

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

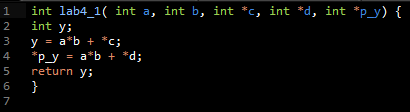


Рис. 1.1. Исходный код

Код теста:

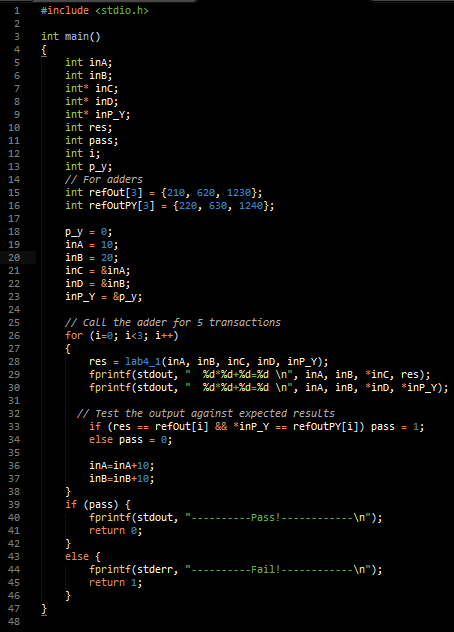


Рис. 1.2. Код теста

* Сделать solution1
  + задать: clock period 6; clock\_uncertainty 0.1
  + 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
    - Открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
* Сделать solution2
  + Задать протоколы
    - a: ap\_hs
    - b: ap\_ask
    - \*c: ap\_hs
    - \*d: ap\_vld
    - \*p\_y: ap\_ask
  + 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
    - Открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval

Объяснить отличие протоколов port\_level

1. Первое решение
   1. Моделирование

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

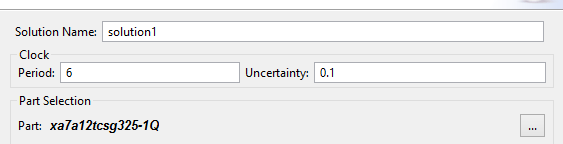


Рис. 2.1. Создание первого решения

При запуске моделирования можно увидеть, что тест успешно пройден:

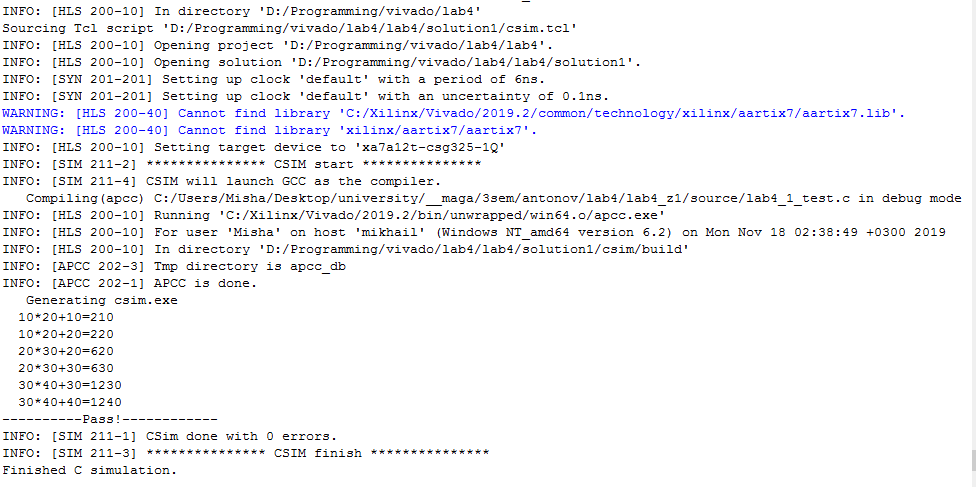


Рис. 2.2. Результаты моделирования

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

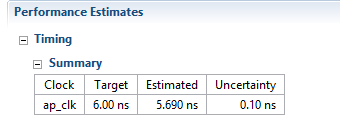


Рис. 2.3. Производительность

Здесь можно увидеть, что достигнутая задержка равна 5.690 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

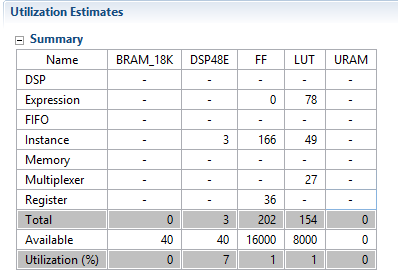


Рис. 2.4. Занимаемые ресурсы

Данный проект займет на микросхеме 1 DSP блок (в котором будут использованы и сумматоры и умножитель), 202 регистров для хранения чисел, и 154 LUT.

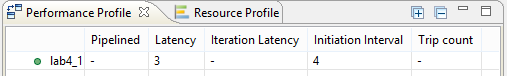


Рис. 2.5. Профиль производительности

На этом рисунке видно, что задержка получения выходного значения составляет 3 такта с момента старта, а задержка после старта до готовности приема новых данных – 4:

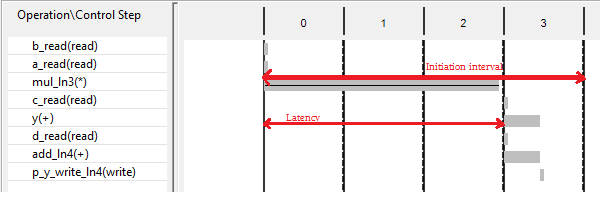


Рис. 2.6. Временная диаграмма

Интерфейс:

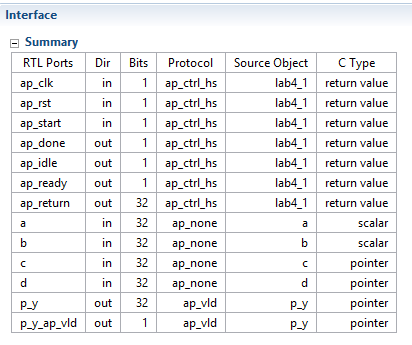


Рис. 2.7. Интерфейс

Конструкция имеет 6 портов данных.

Входные порты: a, b, c, d являются 32-битными входами и имеют входы / выходы, протокол ap\_none.

Конструкция имеет 32-битный выходной порт для возврата функции p\_y, p\_y\_ap\_vld – протокол по умолчанию для портов выхода.

Управляющие сигналы ap\_clk, ap\_rst и ap\_ \* автоматически добавляются в каждый дизайн по умолчанию. Ap\_start, ap\_done, ap\_idle и ap\_ready являются сигналами верхнего уровня, используемыми в качестве сигналов подтверждения связи, чтобы указать, когда проект способен принять следующую команду вычисления (ap\_ready), когда начинается следующее вычисление (ap\_start) и когда вычисление завершено (ap\_done).

* 1. C/RTL моделирование

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

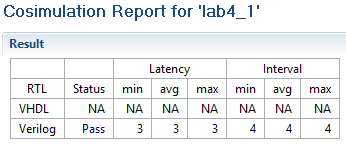


Рис.2.8. Результаты C\RTL моделирования

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

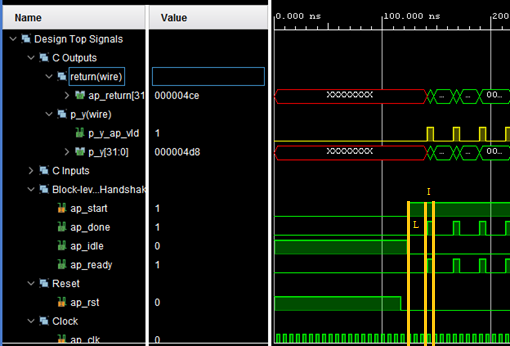


Рис. 2.9. Временная диаграмма совместного моделирования

На временной диаграмме отображены задержка и интервал инициализации.

1. Второе решение
   1. Добавление директив

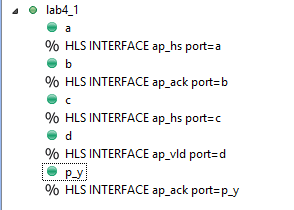


Рис.3.1 Директивы

* ap\_hs, ap\_ack и ap\_vld – это wire handshake протоколы I/O.
* ap\_hs – протокол специализирующий валидацию значения на входе и подтверждения значения на выходе.
* ap\_ack – задает подтверждение на I/O порте.
* ap\_vld – задает валидацию на I/O порте.

Значения считываются тогда, когда ap\_vld имеет высокое значение, а когда считывание закончено – ap\_ack принимает высокое значение.

* 1. Моделирование

Создадим второе решение для данного проекта. Его настройки:

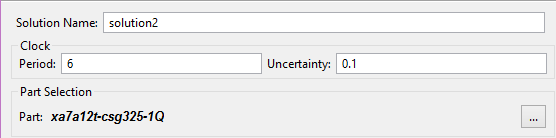


Рис. 3.2.1. Настройки нового решения

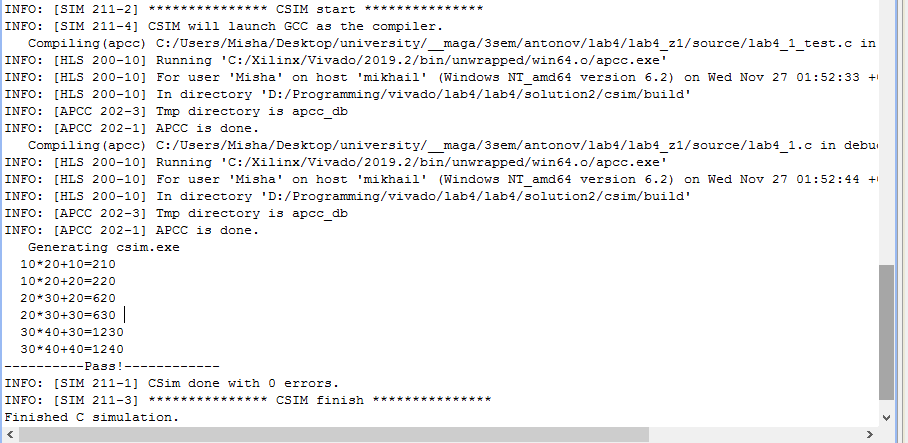


Рис. 3.2.2. Логи моделирования

Моделирования второго решения также прошло успешно.

* 1. Производительность

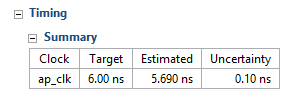


Рис. 3.3. Производительность

* 1. Использование ресурсов

Значения задержки советуют значениям из предыдущего решения.

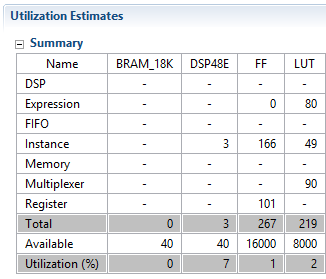


Рис. 3.4.1 Затрачиваемые ресурсы

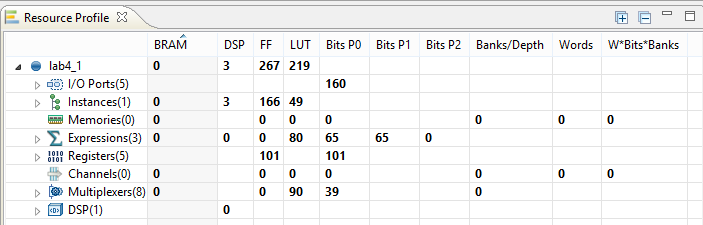


Рис. 3.4.2 Затрачиваемые ресурсы

Данный проект будет занимать на микросхеме:

* 3 DSP
* 267 регистров для хранения и считывания данных
* 219 LUT



Рис. 3.4.3. Module hierrarchy

На изображении выше видно, что задержка получения результата составляет 3 такта, а интервал инициализации – 4.

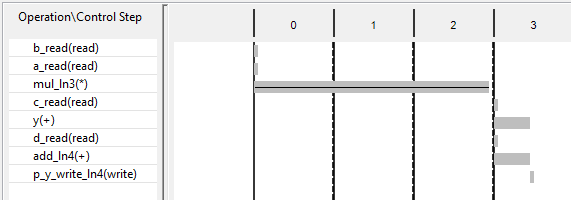


Рис. 3.4.3. Operation\Control step

* 1. Интерфейсы

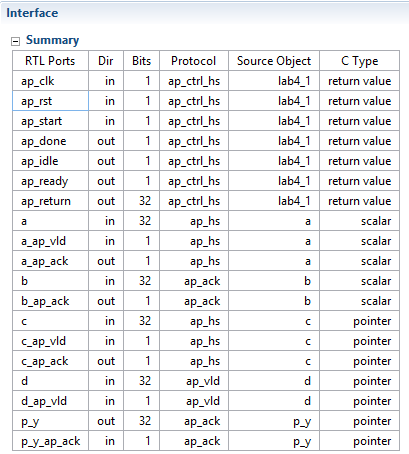


Рис. 3.5. Интерфейсы

По сравнению с предыдущим решением ap\_none изменился на заданные протоколы. Для портов входа и выхода используются следующие протоколы: ap\_hs, ap\_vld, ap\_ack.

Протоколы ввода-вывода на уровне порта ap\_hs обеспечивает наибольшую гибкость в процессе разработки.

Протокол ввода-вывода уровня порта ap\_hs предоставляет следующие сигналы:

- Порт данных

- Сигнал указания момента использования данных (ack)

- Действительный сигнал для указания, когда данные считываются (vld)

Протокол ввода-вывода уровня порта ap\_none является самым простым типом интерфейса и не имеет никаких других сигналов, связанных с ним. Ни входные, ни выходные сигналы данных не имеют связанных портов управления, которые указывают, когда данные считываются или записываются. Единственными портами в конструкции RTL являются порты, указанные в исходном коде. Интерфейс ap\_none не требует дополнительных аппаратных издержек.

* 1. С/RTL моделирование

На временной диаграмме отображены задержка и интервал инициализации.

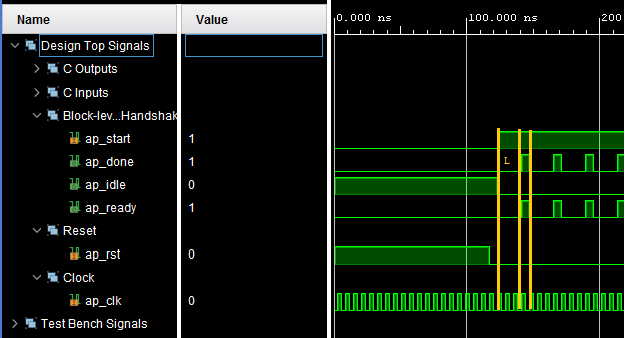


Рис. 3.6. Временная диаграмма совместного моделирования

1. Выводы

Существуют следующие типы port-level wire handshake протоколов:

* ap\_vld
* ap\_ack
* ap\_hs

Они могут быть заданы портов ввода/вывода. Все порты для подтверждений валидаций являются однобитными. По умолчания для входов используется протокол ap\_none, а для выходов – ap\_vld.

Протоколы ввода-вывода на уровне порта ap\_hs обеспечивает наибольшую гибкость в процессе разработки.

Протокол ввода-вывода уровня порта ap\_hs предоставляет следующие сигналы:

- Порт данных

- Сигнал указания момента использования данных (ack)

- Действительный сигнал для указания, когда данные считываются (vld)

Протокол ввода-вывода уровня порта ap\_none является самым простым типом интерфейса и не имеет никаких других сигналов, связанных с ним. Ни входные, ни выходные сигналы данных не имеют связанных портов управления, которые указывают, когда данные считываются или записываются. Единственными портами в конструкции RTL являются порты, указанные в исходном коде. Интерфейс ap\_none не требует дополнительных аппаратных издержек.