Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Лабораторная работа 11**

**Предмет: Проектирование реконфигурируемых гибридных вычислительных систем**

**Тема: Latency**

**Задание 4**

Выполнил студент гр. 3540901/81501 Селиверстов Я.А.

(подпись)

Руководитель Антонов А.П.

(подпись)

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Санкт – Петербург

2019

Оглавление

[1. Задание 4](#_Toc30365485)

[2. Моделирование 6](#_Toc30365486)

[3. Первое решение 6](#_Toc30365487)

[3.1. Синтез 6](#_Toc30365488)

[3.2. C\RTL моделирование 8](#_Toc30365489)

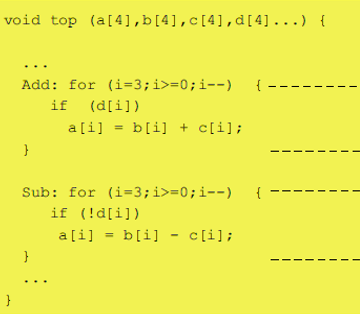
[4. Второе решение 9](#_Toc30365490)

[4.1. Синтез 9](#_Toc30365491)

[5. Выводы 11](#_Toc30365492)

1. Задание

* Создать проект lab11\_4
* Микросхема: xa7a12tcsg325-1q
* Создать функцию по образцу (иерархия функций)



* Создать тест lab11\_4\_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
* Исследование:
* Solution\_1а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ПО УМОЛЧАНИЮ
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource 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
  + Выполнить cosimulation и привести временную диаграмму
* Solution\_2а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию LOOP\_MERGE
  + осуществить синтез
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource 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
  + Выполнить cosimulation и привести временную диаграмму
  + Сравнить два решения (solution\_1a и solution\_2a) и сделать выводы: объяснить (посчитать) число циклов Latency, II…

Исходный текст подготовленной для синтеза функции и теста приведён ниже:

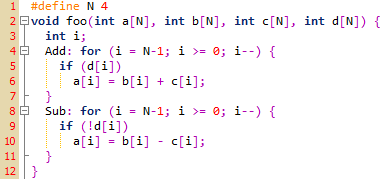


Рис. 1.1. Исходный код синтезируемой функции

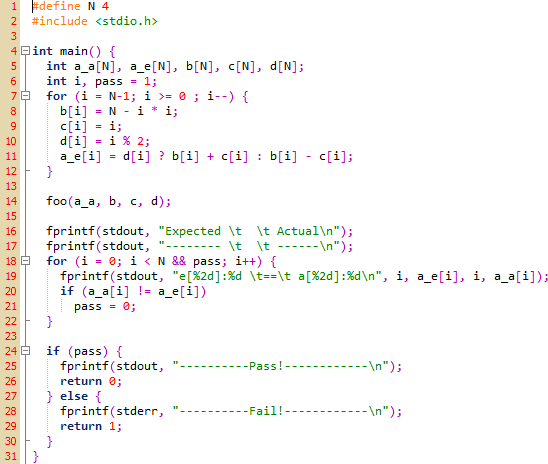


Рис. 1.2. Исходный код теста

1. Моделирование

При запуске моделирования можно увидеть, что тест успешно пройден:

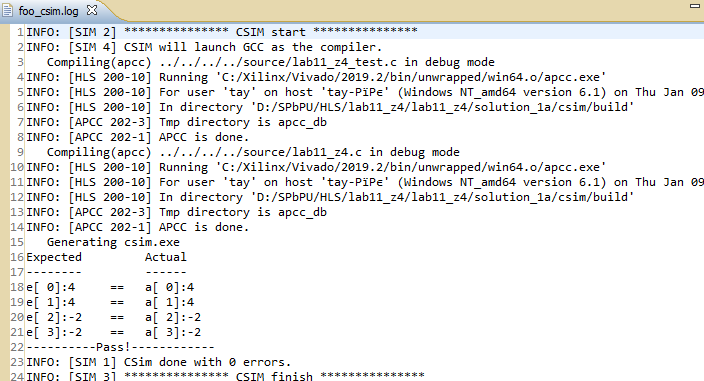


Рис. 2.1. Результаты моделирования

1. Первое решение
   1. Синтез

Приведем в отчете требуемые данные о проекте:

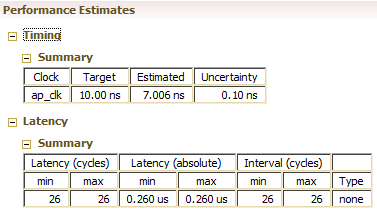


Рис. 3.1. Производительность

Здесь можно увидеть, что достигнутая задержка равна 7.006 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

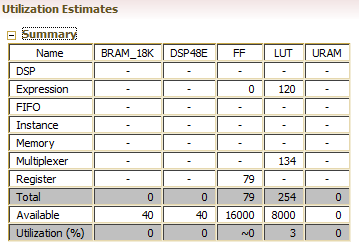


Рис. 3.2. Занимаемые ресурсы

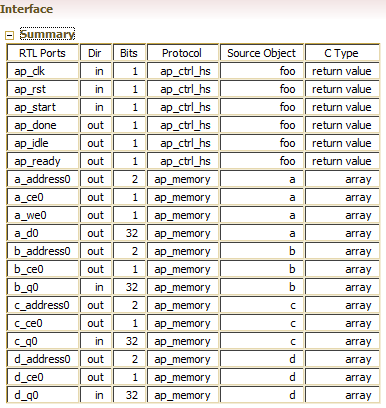


Рис. 3.3. Применяемые интерфейсы

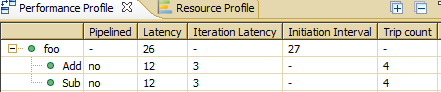


Рис. 3.4. Профиль производительности

На этом рисунке видно, что задержка получения первого выходного значения составляет 4 такта с момента старта (для всех – 26), а задержка после старта до готовности приема новых данных – 27:

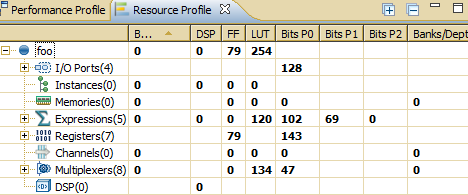


Рис. 3.5. Профиль использования ресурсов

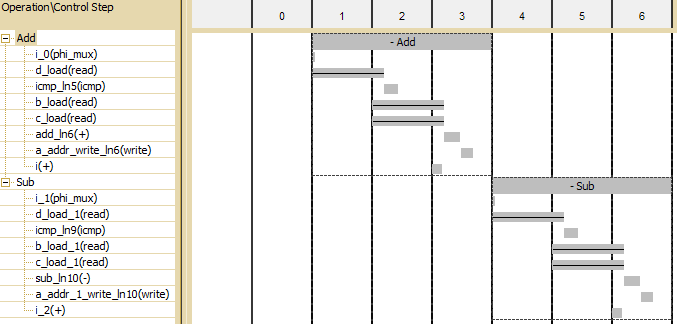


Рис. 3.6. Временная диаграмма

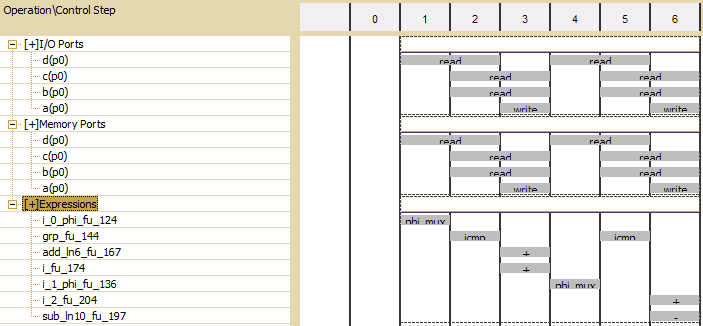


Рис. 3.7. Диаграмма использования ресурсов

* 1. C\RTL моделирование

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

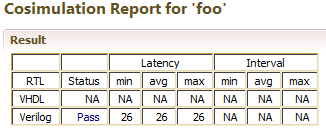


Рис. 3.8. C\RTL моделирование

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

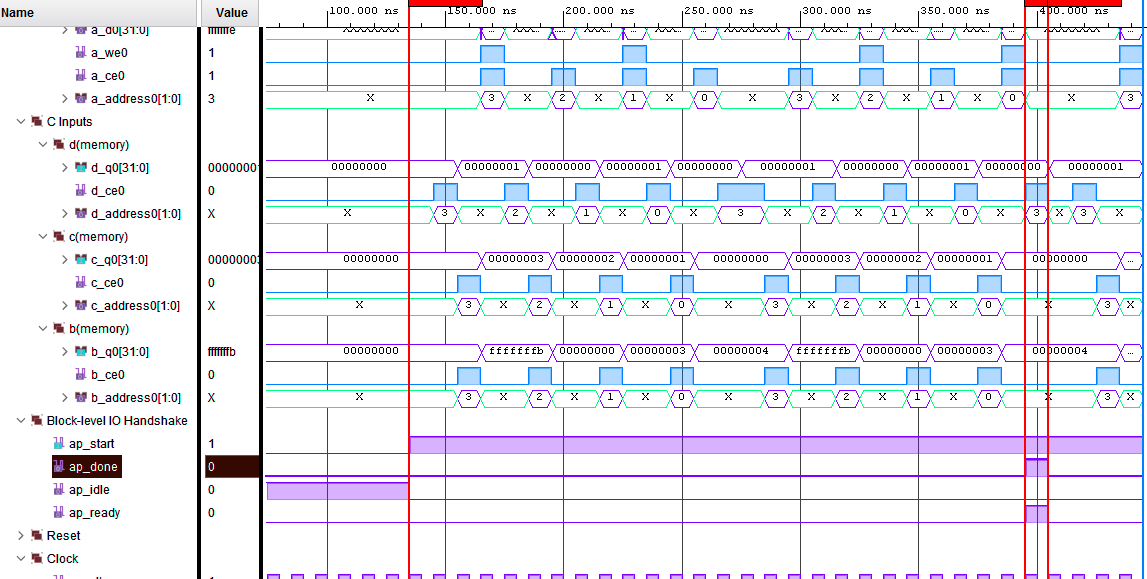


Рис. 3.9. Временная диаграмма совместного моделирования

1. Второе решение

Добавим директиву, которая управляет слиянием циклов.

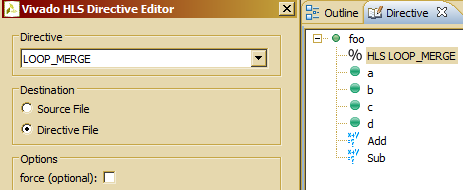


Рис. 4.1. Добавление директивы

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

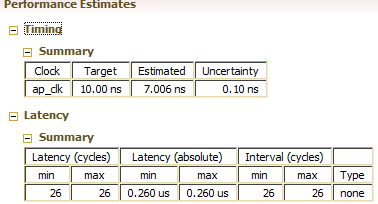


Рис. 4.2. Производительность

Здесь можно увидеть, что достигнутая задержка равна 7.006 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

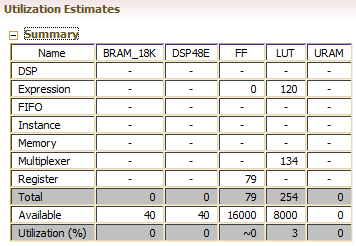


Рис. 4.3. Затрачиваемые ресурсы

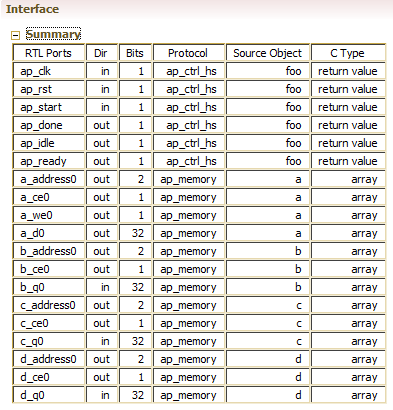


Рис. 4.4. Применяемые интерфейсы

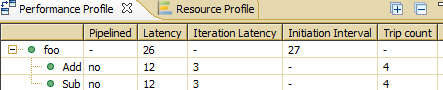


Рис. 4.5. Профиль производительности

На этом рисунке видно, что рассматриваемое решение полностью соответствует предыдущему.

1. Выводы

C помощью директивы Loop\_Merge можно выполнять объединение тел нескольких циклов в одно для избавления от дополнительных тактов, вызванных инициализацией. Однако, не смотря на ожидаемое поведение, для описанного устройства циклы не были объединены. Возможная причина – наличие конструкции ветвления в обоих циклах.