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 5**

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

(подпись)

Руководитель Антонов А.П.

(подпись)

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

Санкт – Петербург

2019

Оглавление

[1. Задание 4](#_Toc30370349)

[2. Моделирование 6](#_Toc30370350)

[3. Первое решение 6](#_Toc30370351)

[3.1. Синтез 6](#_Toc30370352)

[4. Второе решение 8](#_Toc30370353)

[4.1. Синтез 9](#_Toc30370354)

[4.2. C\RTL моделирование 10](#_Toc30370355)

[5. Третье решение 11](#_Toc30370356)

[5.1. Синтез 11](#_Toc30370357)

[6. Выводы 13](#_Toc30370358)

1. Задание

* Создать проект lab11\_5
* Микросхема: xa7a12tcsg325-1q
* Познакомьтесь с исходным кодом функции loop\_var.c
* Познакомьтесь с исходным кодом теста loop\_var\_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
* Solution\_2а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить LOOP\_TRIPCOUNT (MIN 4 MAX 32 AVERAGE 18)
  + 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…
* Solution\_3а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить (В исходном коде: assert (width<64))
  + 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
  + Сравнить два решения (solution\_2a и solution\_3a) и сделать выводы: объяснить (посчитать) число циклов Latency, II…

Исходный текст подготовленной для синтеза функции и теста приведён ниже:

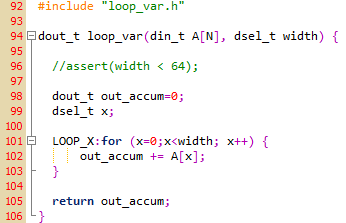


Рис. 1.1. Исходный код синтезируемой функции

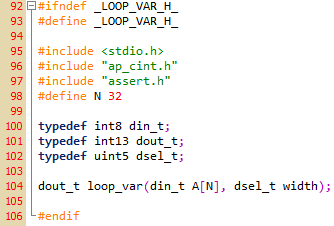


Рис. 1.2. Исходный код заголовочного файла

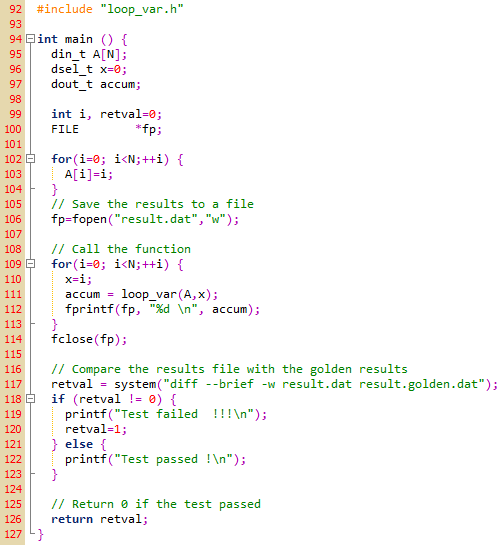


Рис. 1.3. Исходный код теста

1. Моделирование

При запуске моделирования можно увидеть, что тест успешно пройден:

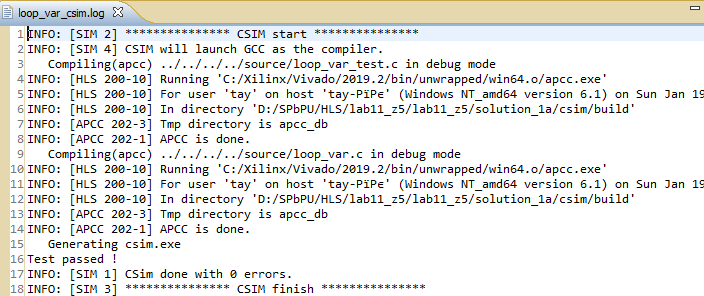


Рис. 2.1. Результаты моделирования

1. Первое решение
   1. Синтез

Приведем в отчете требуемые данные о проекте:

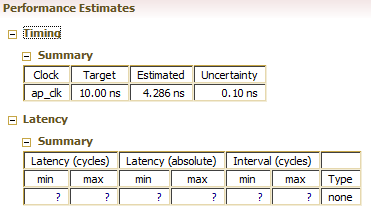


Рис. 3.1. Производительность

Здесь можно увидеть, что достигнутая задержка равна 4.286 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

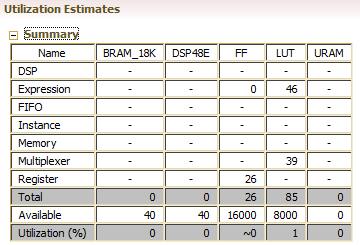


Рис. 3.2. Занимаемые ресурсы

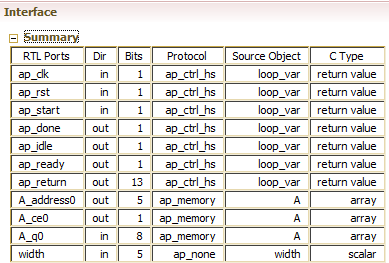


Рис. 3.3. Применяемые интерфейсы

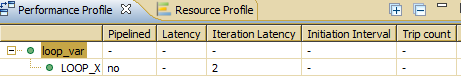


Рис. 3.4. Профиль производительности

На этом рисунке видно, что задержка получения первого выходного значения составляет 3 такта с момента старта, однако, невозможно сказать, через сколько тактов будут обработаны все данные и можно будет подавать новые.

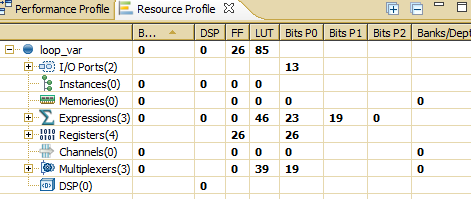


Рис. 3.5. Профиль использования ресурсов

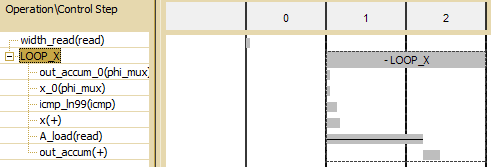


Рис. 3.6. Временная диаграмма

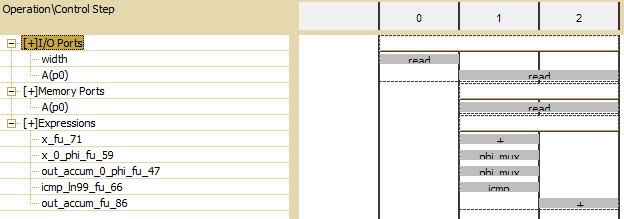


Рис. 3.7. Диаграмма использования ресурсов

1. Второе решение

Добавим директиву, задающую пределы, в которых находится число возможных итераций цикла.

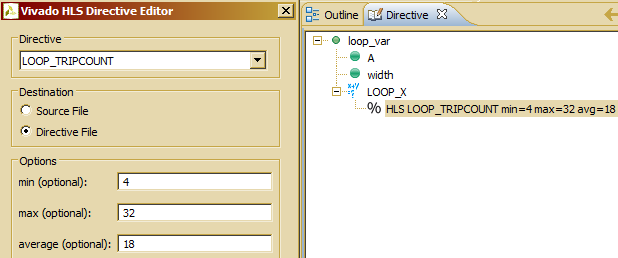


Рис. 4.1. Добавление директивы

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

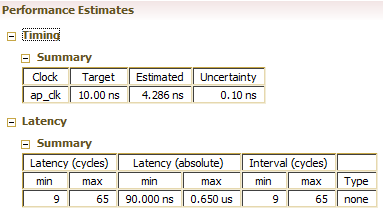


Рис. 4.2. Производительность

Здесь можно увидеть, что достигнутая задержка равна 4.286 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

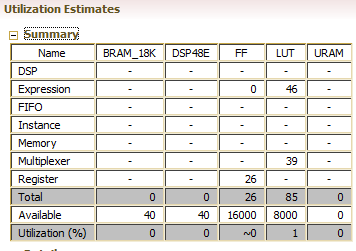


Рис. 4.3. Затрачиваемые ресурсы

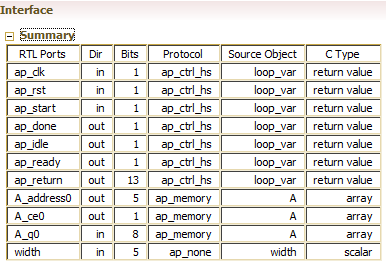


Рис. 4.4. Применяемые интерфейсы

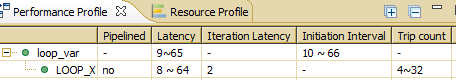


Рис. 4.5. Профиль производительности

На этом рисунке видно, что задержка получения результатов с момента старта – от 9 до 65 тактов, а задержка после старта до готовности приема новых данных – от 10 до 66:

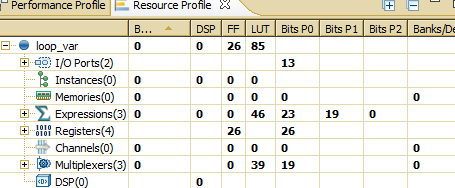


Рис. 4.6. Профиль использования ресурсов

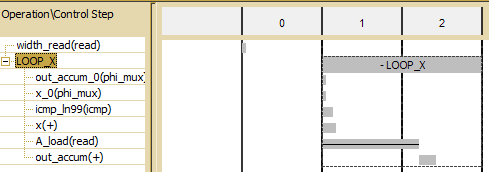


Рис. 4.7. Временная диаграмма

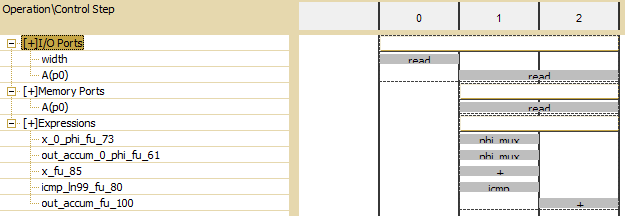


Рис. 4.8. Диаграмма использования ресурсов

* 1. C\RTL моделирование

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

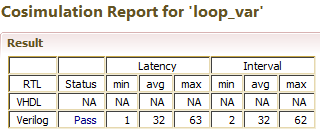


Рис. 4.9. C\RTL моделирование

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

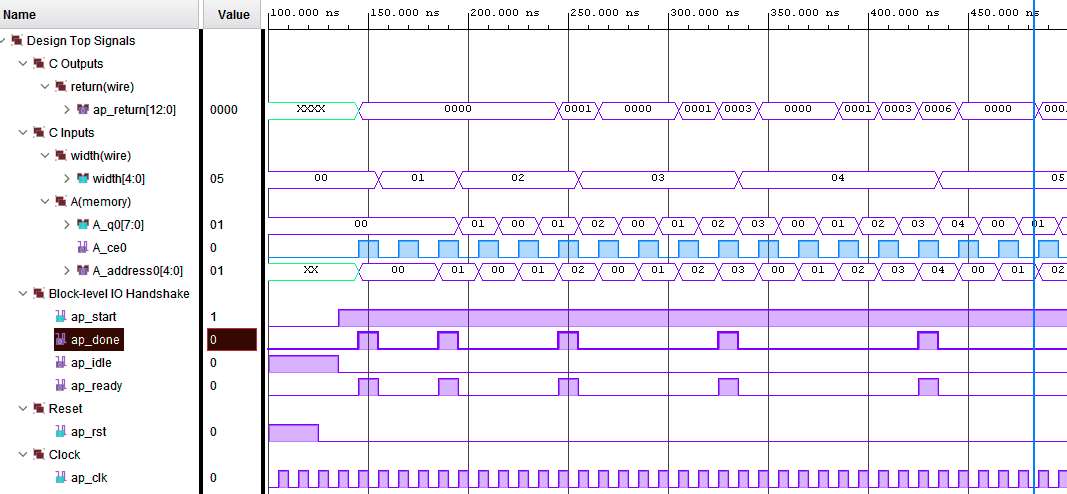


Рис. 4.10. Временная диаграмма совместного моделирования

Отчетливо видно, что устройство имеет различное время работы (сигнал ap\_done) в зависимости от входных данных.

1. Третье решение

Раскомментируем строку «assert(width < 64)».

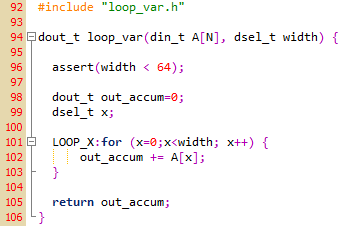


Рис. 5.1. Измененный исходный текст функции

* 1. Синтез

Приведем в отчете требуемые данные о проекте:

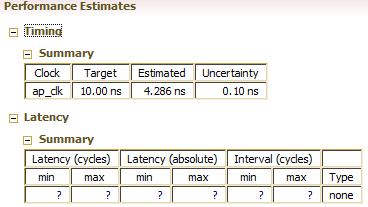


Рис. 5.2. Производительность

Здесь можно увидеть, что достигнутая задержка равна 4.286 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

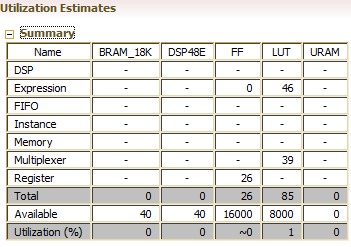


Рис. 5.3. Затрачиваемые ресурсы

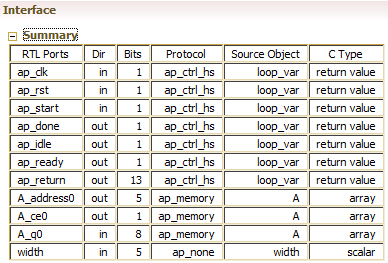


Рис. 5.4. Применяемые интерфейсы

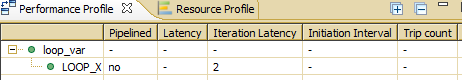


Рис. 5.5. Профиль производительности

На этом рисунке видно, что задержка получения первого выходного значения составляет 3 такта с момента старта, однако, невозможно сказать, через сколько тактов будут обработаны все данные и можно будет подавать новые.

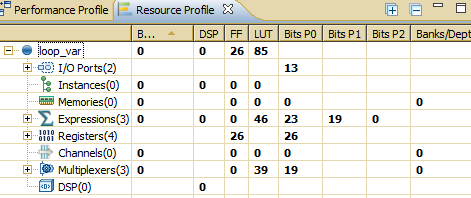


Рис. 5.6. Профиль использования ресурсов

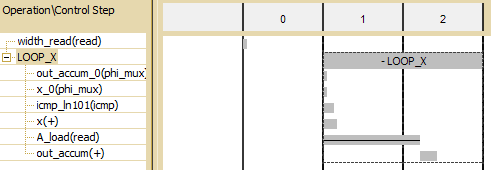


Рис. 5.7. Временная диаграмма

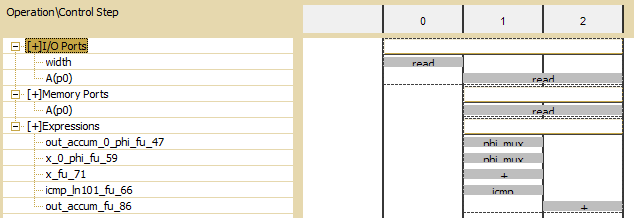


Рис. 5.8. Диаграмма использования ресурсов

Рассматриваемое решение полностью совпадает с первым решением.

1. Выводы

В данной работе была рассмотрена возможность реализации циклов с неизвестным заранее количеством итераций. Для решения проблемы имеются следующие средства: директива Loop\_Count и функция Assert. Применение последней не возымела эффекта в данной работе.