Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Лабораторная работа №14. Задание №3**

**Курс: «Проектирование реконфигурируемых гибридных вычислительных систем»**

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

(подпись)

Руководитель Антонов А.П.

(подпись)

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

Санкт – Петербург

2020

Оглавление

[1. Задание 3](#_Toc30376518)

[2. Исходный код 4](#_Toc30376519)

[3. Ход работы 7](#_Toc30376520)

[3.1. Решение 1а 7](#_Toc30376521)

[3.1.1. Моделирование 7](#_Toc30376522)

[3.1.2. Синтез 7](#_Toc30376523)

[3.1.3. C/RTL моделирование 9](#_Toc30376524)

[3.2. Решение 2а 10](#_Toc30376525)

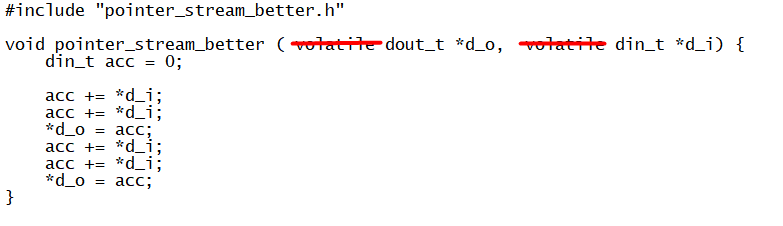
[3.2.1. Моделирование 10](#_Toc30376526)

[3.2.2. Синтез 10](#_Toc30376527)

[3.2.3. C/RTL моделирование 11](#_Toc30376528)

[4. Выводы 12](#_Toc30376529)

# Задание

* Создать проект lab14\_3
* Микросхема: xa7a12tcsg325-1q
* В папке source текст функции pointer\_stream\_better
  + *Познакомьтесь с ним (посмотрите в лекции часть Multi-Access Pointers)*
* Познакомьтесь с тестом.
* Исследование:
* Solution\_1а
  + Создать версию pointer\_stream\_ , в которой будет убран volatile
* 
  + Осуществить моделирование (при необходимости изменить тест) – обратить внимание на раздел тестирования в лекции
  + 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
  + Использовать исходную функцию pointer\_stream\_better
  + Осуществить моделирование – обратить внимание на раздел тестирования в лекции
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ПО УМОЛЧАНИЮ
  + 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

# Исходный код

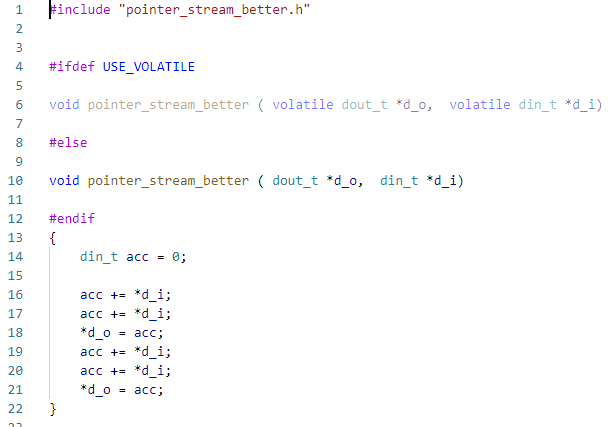


Рисунок . Исходный код устройства

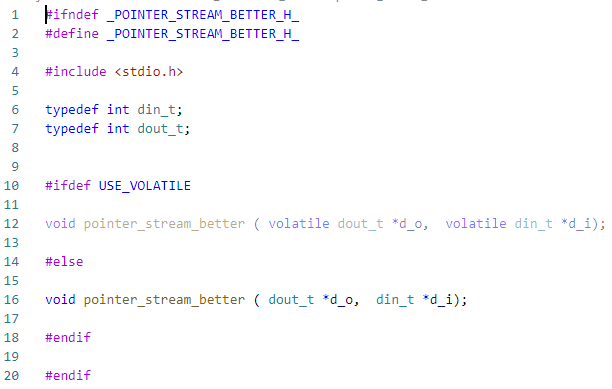


Рисунок . Заголовочный файл

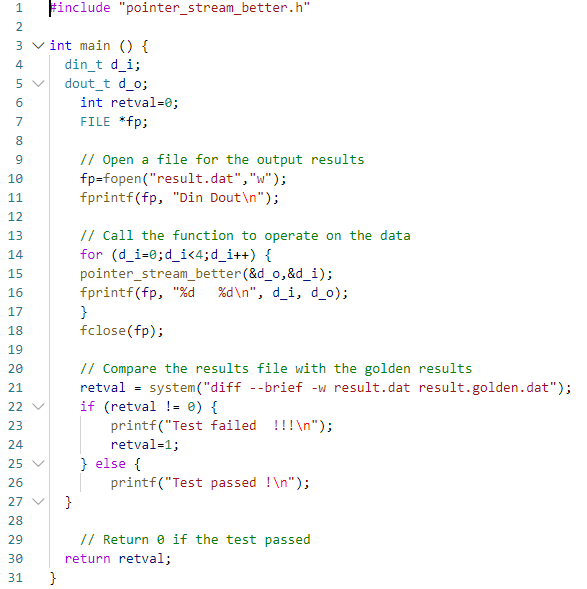


Рисунок . Тест

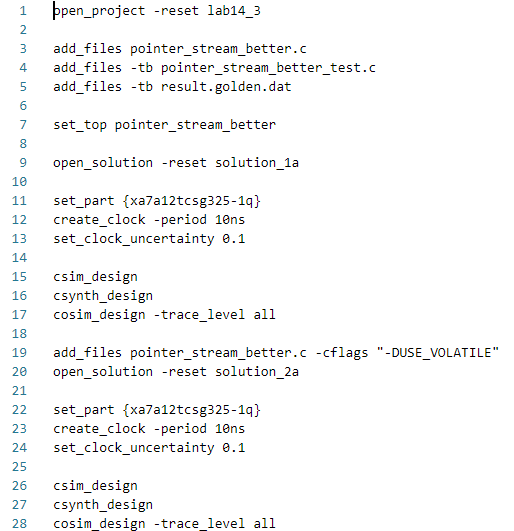


Рисунок . Скрипт

# Ход работы

## Решение 1а

## Моделирование

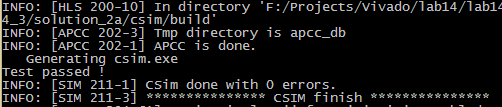


Рисунок . Результаты моделирования

Устройство работает корректно

## Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

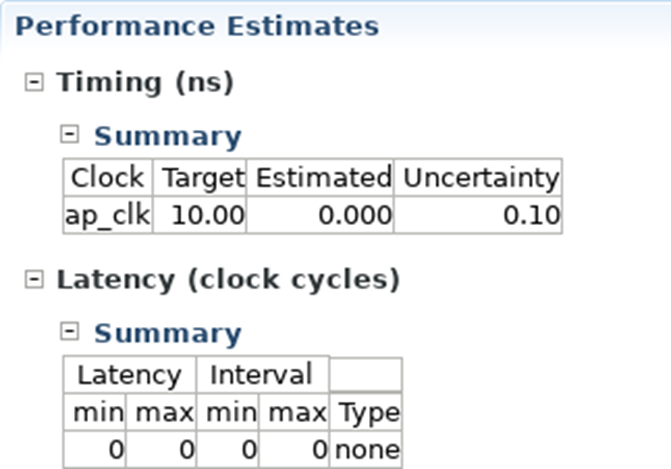


Рисунок .Performance estimates

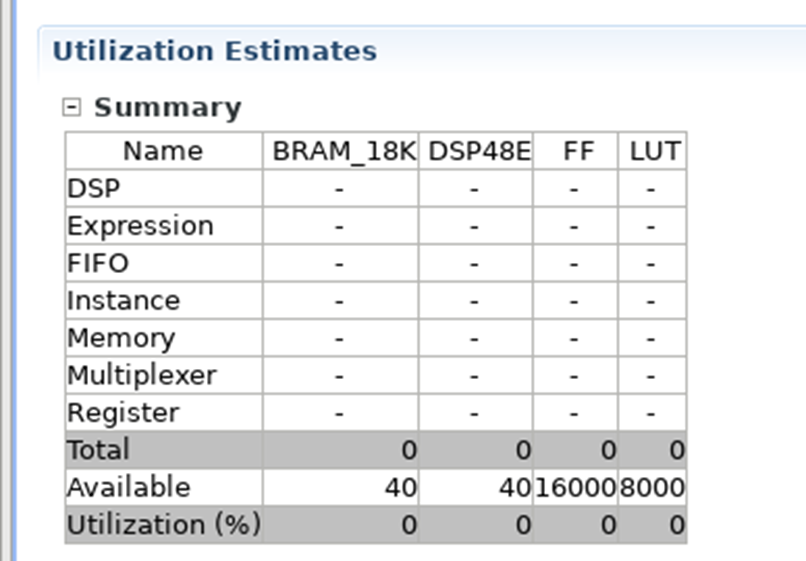


Рисунок . Utilization estimates

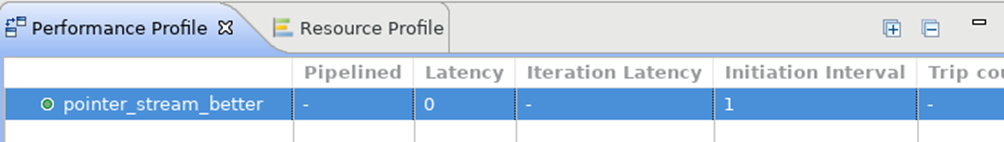


Рисунок . Performance profile



Рисунок . Scheduler viewer

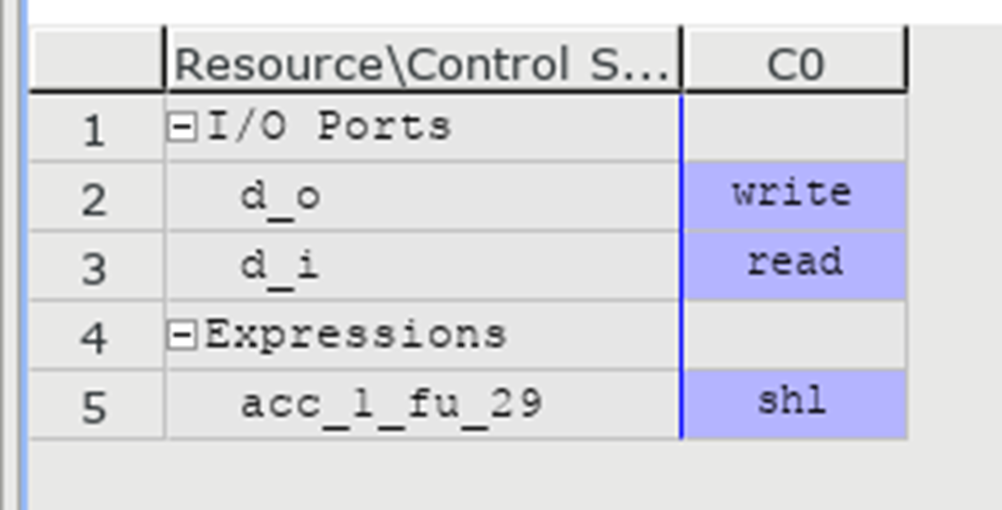


Рисунок . Resource viewer

## C/RTL моделирование

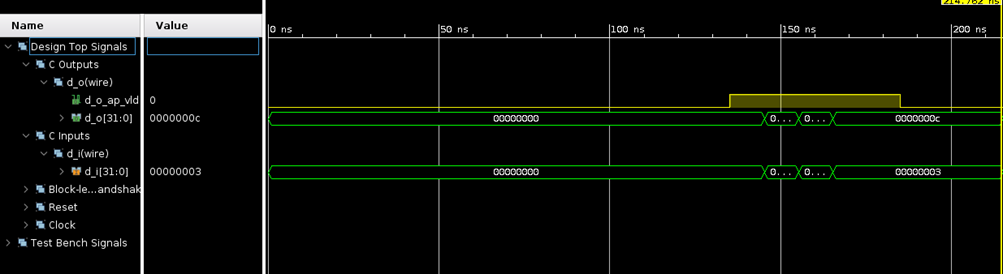


Рисунок . Временная диаграмма

Как видно по результатам выше, полученное решение выполняется за один такт.

## Решение 2а

## Моделирование

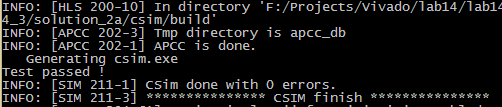


Рисунок . Результаты моделирования

Устройство работает корректно

## Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

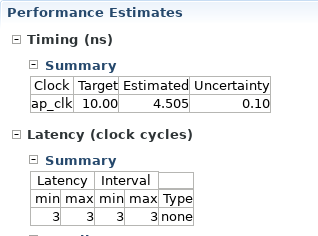


Рисунок . Performance estimates

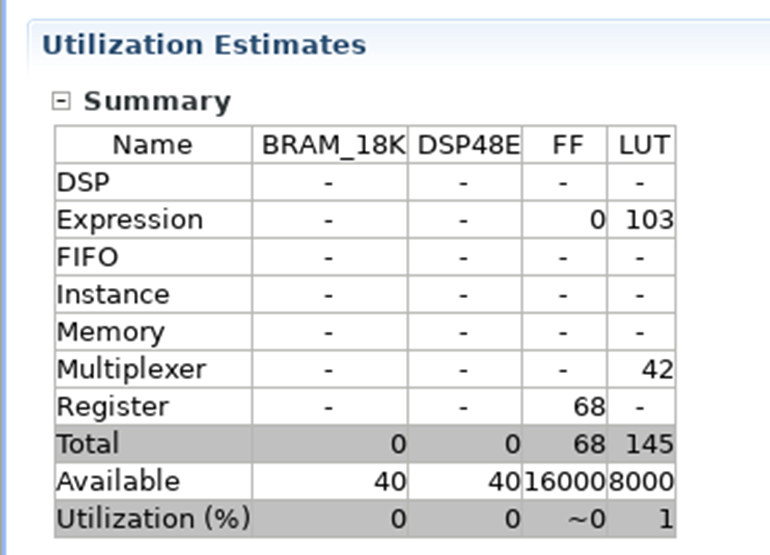


Рисунок . Utilization estimates

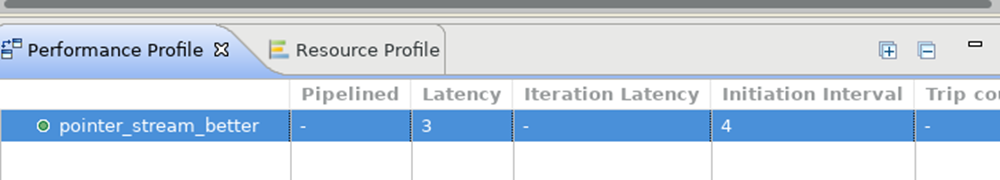


Рисунок . Performance profile

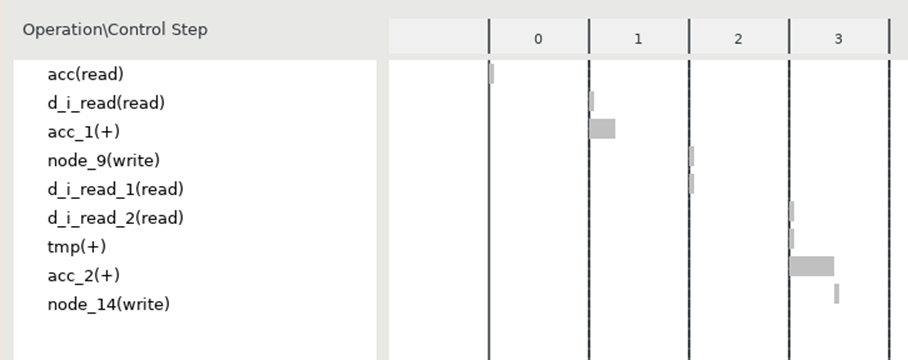


Рисунок . Scheduler viewer

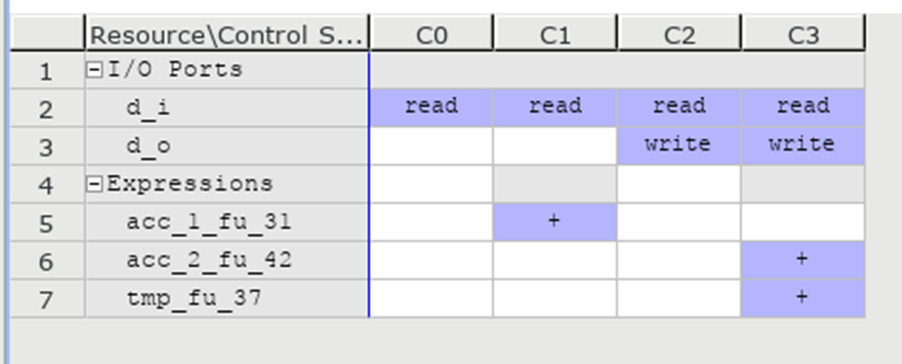


Рисунок . Resource viewer

## C/RTL моделирование

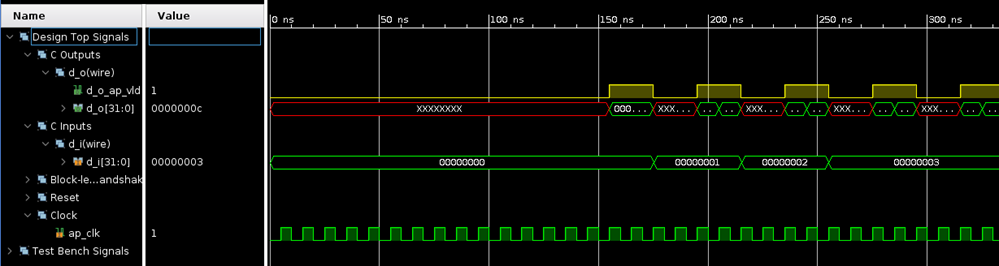


Рисунок . Временная диаграмма

Как видим, при использовании ключевого слова volatile производительность проекта снизилась. Это связано с тем, что отключена оптимизация операций и теперь у на выходе значения появляются 2 раза, а не один раз в самом конце, как в предыдущем решении.

# Выводы

Ключевое слово volatile используется когда требуется полный контроль над выполнением операций для избегания моментов когда компилятор оптимизирует исходный код, тем самым убирает ненужные строки кода которые по-другому работают на аппаратуре нежели на процессоре.