Санкт-Петербургский Политехнический Университет Петра Великого

Институт Компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа 13 Задание 2

Предмет: «Проектирование реконфигурируемых гибридных вычислительных систем»

**Тема: «Сравнение типов данных»**

Студент: Медведев М.А.

Онищук М.П.

Гр. № 3540901/81501,

3540901/81502

Преподаватель: Антонов А.П.

Санкт-Петербург

2019

Оглавление

[1. Задание 3](#_Toc27506480)

[2. Исходный код 5](#_Toc27506481)

[3. Первое решение 6](#_Toc27506482)

[3.1. Моделирование 6](#_Toc27506483)

[4. Второе решение 7](#_Toc27506484)

[4.1. Моделирование 7](#_Toc27506485)

[4.2. Синтез 7](#_Toc27506486)

[4.3. Использование ресурсов 7](#_Toc27506487)

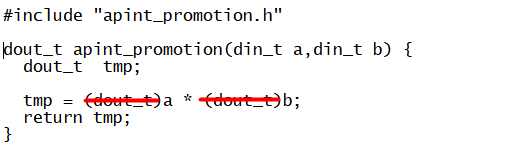
[5. Выводы 8](#_Toc27506488)

1. Задание

* Создать проект lab13\_2
* Микросхема: xa7a12tcsg325-1q
* В папке source текст функции apint\_promotion

*Познакомьтесь с ним.*

* Познакомьтесь с тестом.
* Исследование:
* Solution\_1а
  + Создать версию apint\_ , в которой будет убран Кастинг



* + Осуществить моделирование (при необходимости изменить тест)
  + 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 apint\_promotion
  + Осуществить моделирование
  + 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

1. Исходный код

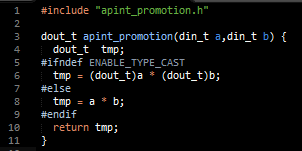


Рис. 2.1. Source code

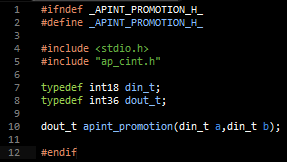


Рис. 2.2. Source code - header

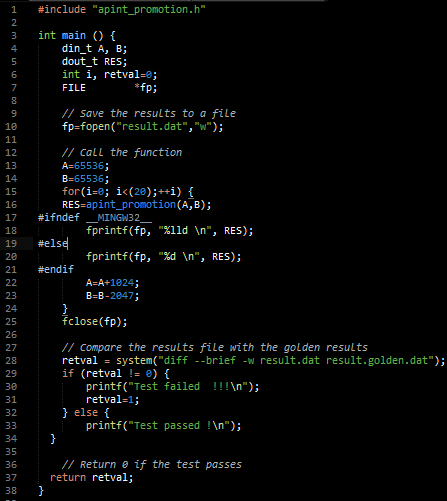


Рис. 2.1.3. Test code

1. Первое решение
   1. Моделирование

По результатам моделирования видно, что устройство работает не корректно.

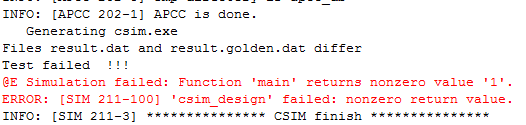


Рис. 3.1. Modeling result

1. Второе решение
   1. Моделирование

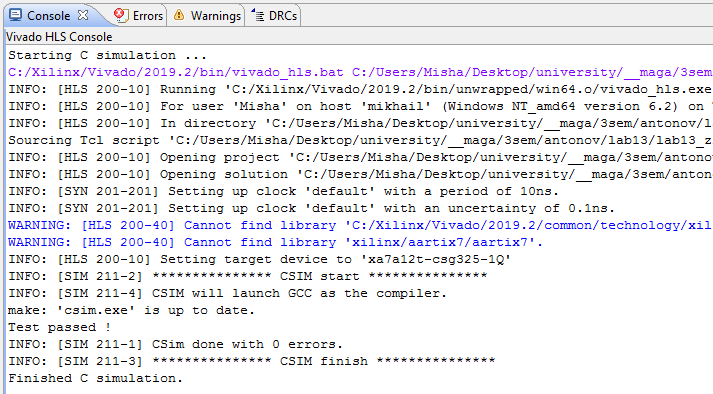


Рис. 4.1. Performance estimates

* 1. Синтез

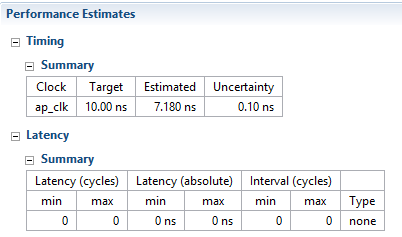


Рис. 4.2. Performance estimates

* 1. Использование ресурсов

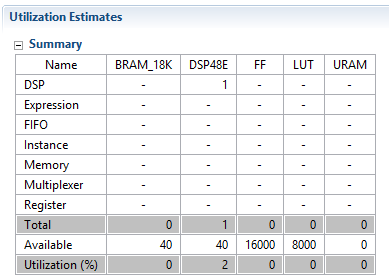


Рис. 4.3.1. Utilization estimates

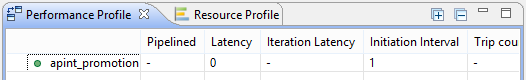


Рис. 4.3.2. Performance profile

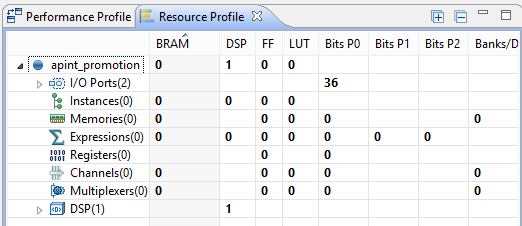


Рис. 4.3.3. Resource profile

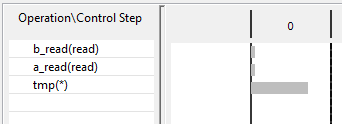


Рис. 2.4.4. Schedule viewer

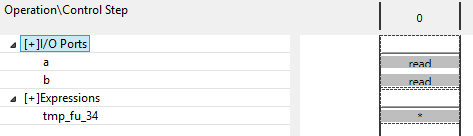


Рис. 2.4.5. Resource viewer

1. Выводы

В ходе выполнения работы была продемонстрирована важность порядка преобразования типов. В первом решении, операция умножения выполняется для двух операндов с типом d\_in, который значительно меньше, чем d\_out. Во время умножения происходит переполнение, и результат оказывается некорректным. Во втором решении, умножение выполняется над операндами с типом d\_out, так как произведено явное преобразование типов. Этот тип более вместительный, и переполнения не происходит.