Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчет по лабораторной работе №3\_1**

**Курс: «Проектирование реконфигурируемых гибридных**

**вычислительных систем»**

**Тема: «Port-level IO protocols»**

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

(подпись)

Руководитель Антонов А.П.

(подпись)

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

Санкт – Петербург

2019

Оглавление

[1. Задание 3](#_Toc26466596)

[2. Первое решение 5](#_Toc26466597)

[2.1 Исходный код функции и программы 5](#_Toc26466598)

[2.2 Моделирование 6](#_Toc26466599)

[2.3 Синтез 6](#_Toc26466600)

[2.4 C|RTL моделирование 8](#_Toc26466601)

[3. Второе решение 10](#_Toc26466602)

[3.1 Моделирование 10](#_Toc26466603)

[3.2 Синтез 10](#_Toc26466604)

[3.3 C|RTL моделирование 13](#_Toc26466605)

[4. Третье решение 14](#_Toc26466606)

[4.1 Моделирование 14](#_Toc26466607)

[4.2 Синтез 14](#_Toc26466608)

[4.3 C|RTL моделирование 17](#_Toc26466609)

[5.Выводы 17](#_Toc26466610)

# 1. Задание

* Создать проект lab2\_1
* Подключить файл lab2\_1.c (папка source)
* Подключить тест lab2\_1\_test.c (папка source)
* Микросхема: xa7a12tcsg325-1q
* 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 протокол (block-level): ap\_cntl\_chain
  + 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 solution3
  + Задать протокол (block-level): ap\_cntl\_none
  + задать: clock period 10; 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
    - Проверить происходит или нет моделирование, объяснить почему.
    - Если моделирование происходит, то открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
* Выводы
  + Объяснить отличие протоколов block\_level

# 2. Первое решение

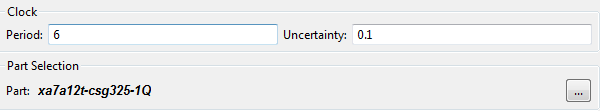
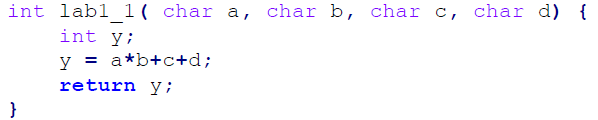
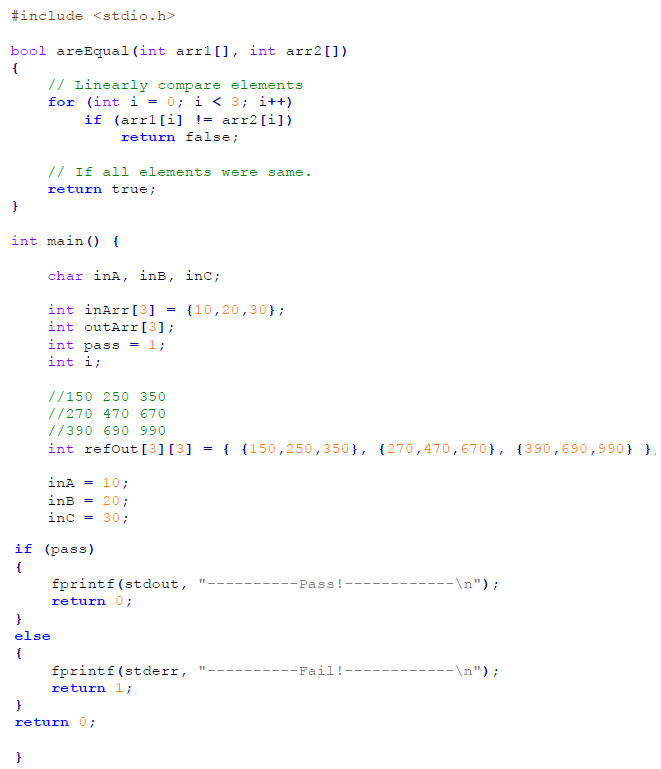


Рисунок 2. Параметры первого решения

## 2.1 Исходный код функции и программы





## 2.2 Моделирование

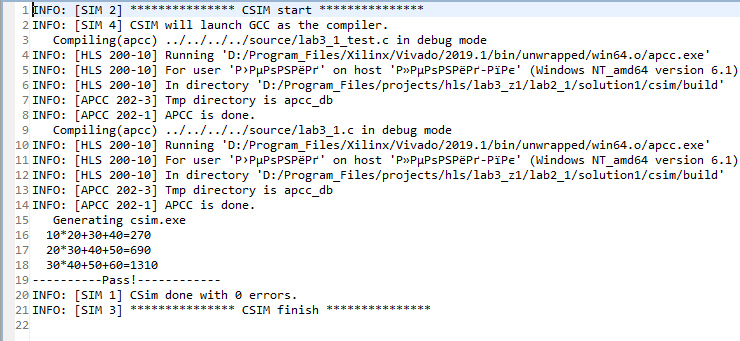


Рисунок .2 Результат успешного моделирования

Моделирование второго решения прошло успешно.

## 2.3 Синтез

Данные о проекте



Рисунок 2. Performance estimates – summary

Здесь можно увидеть, что достигнутая задержка равна 3.820 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

Использование ресурсов

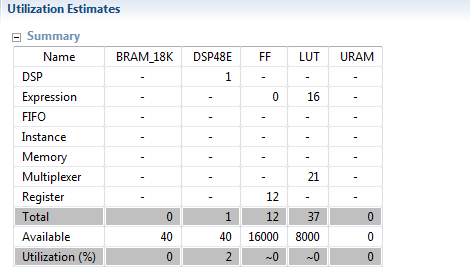


Рисунок 2. Utilization estimates – summary

Данный проект займет на микросхеме 1 DSP блок (в котором будут использованы и сумматоры и умножитель), 12 регистров для хранения чисел, и 37 LUT.

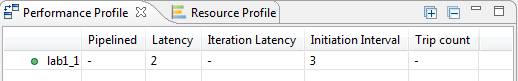


Рисунок 2.a Performance Profile

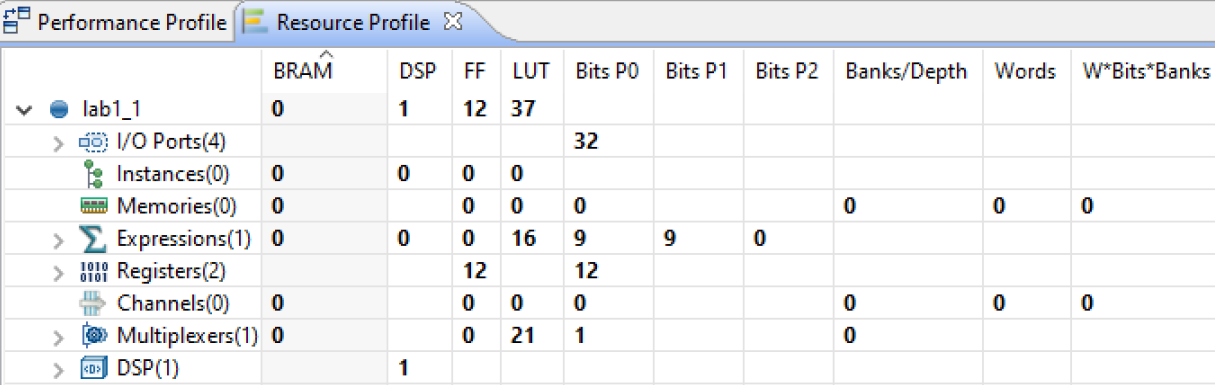


Рисунок 2.b Performance Profile

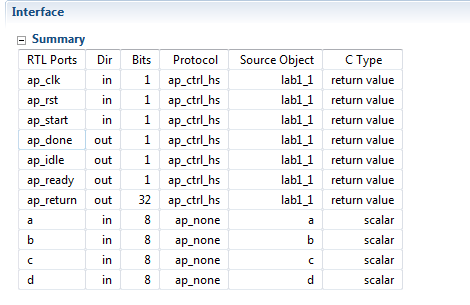


Рисунок 2.6 Interface estimates – summary

Для расчета схемы требуется более одного такта, поэтому в схему были добавлены ap\_clk и ap\_rst. Оба являются однобитовыми входами. Протокол управления вводом / выводом на уровне блоков был добавлен для управления RTL. Порты: ap\_start, ap\_done, ap\_idle и ap\_ready. Входные порты: a, b, c, d являются 8-битными входами и имеют входы / выходы, протокол ap\_none.

Конструкция имеет 32-битный выходной порт для возврата функции ap\_return.

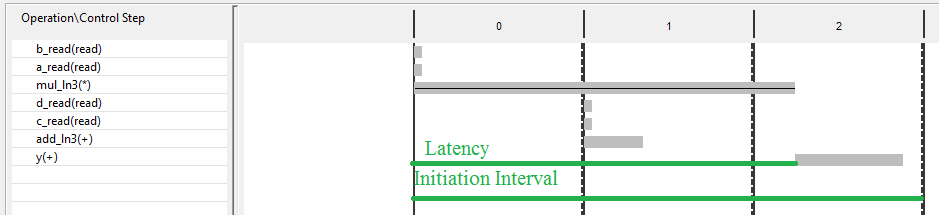


Рисунок 2. Schedule viewer

На рисунке 2.7 видно, что задержка получения выходного значения составляет 2 такта с момента старта, а задержка после старта до готовности приема новых данных – 3. Покажем эти интервалы на временной диаграмме. Здесь мы видим весь процесс получения результата. На первом такте происходит считывание операторов А и В, а также начинается их умножение. На втором такте начинается считывание C и D, а также их сложение. Таким образом суммарная задержка latency = 2, а со следующего 3-го такта можно подавать следующие данные (II = 3).

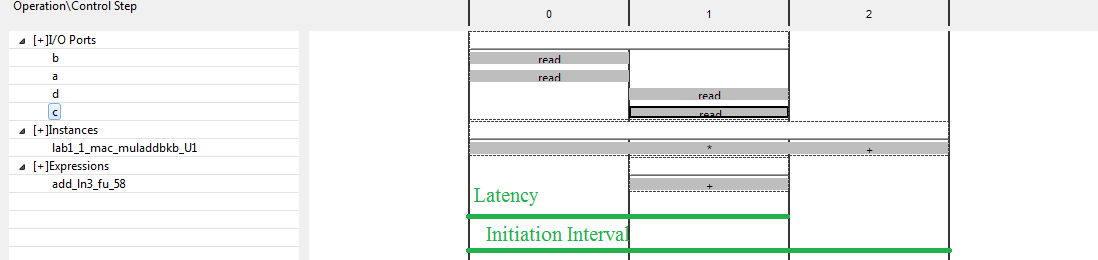


Рисунок 2. Resource viewer

## 2.4 C|RTL моделирование

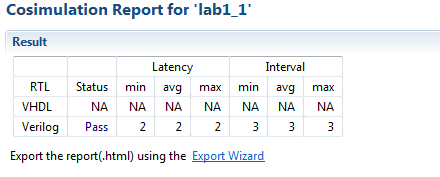


Рисунок 2. Отчет о моделировании

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

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

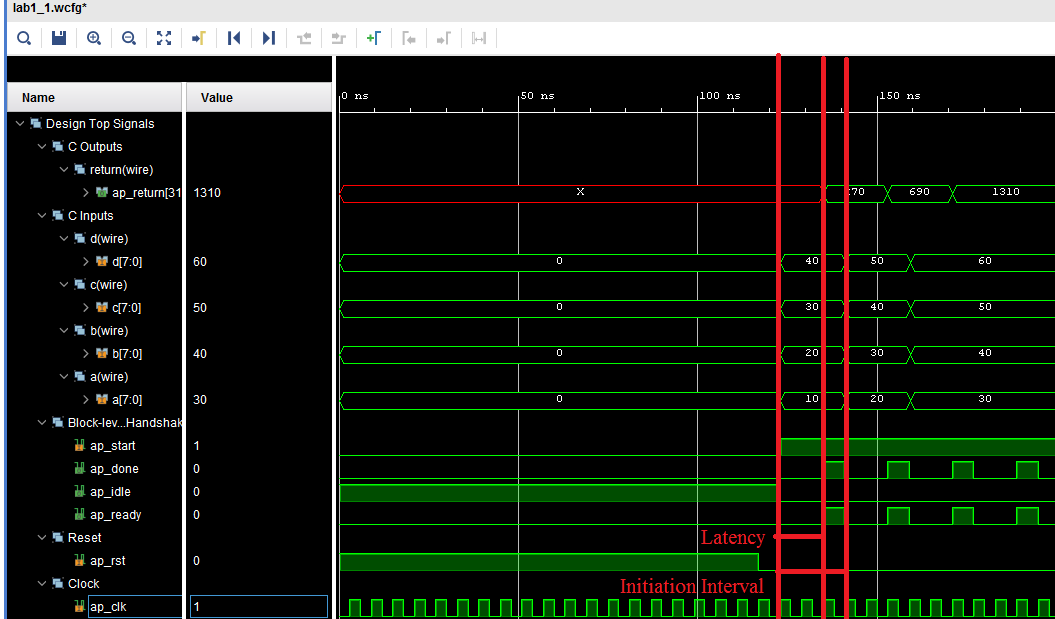


Рисунок 2. Временная диаграмма

# 3. Второе решение

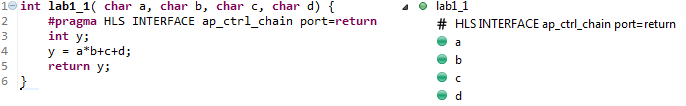


Рисунок 3.1Листинг кода и директивы

Во-втором задании добавлен протокол ap\_ctrl\_chain. Данный протокол является протоколом ввода-вывода на уровне блоков для цепочки управления. Этот протокол ввода / вывода в основном используется для объединения конвейерных блоков.

## 3.1 Моделирование

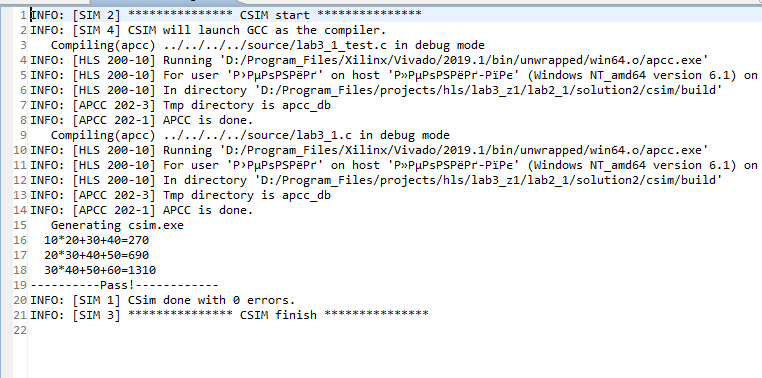


Рисунок 3.2 Результат успешного моделирования

Моделирование второго решения также прошло успешно.

## 3.2 Синтез

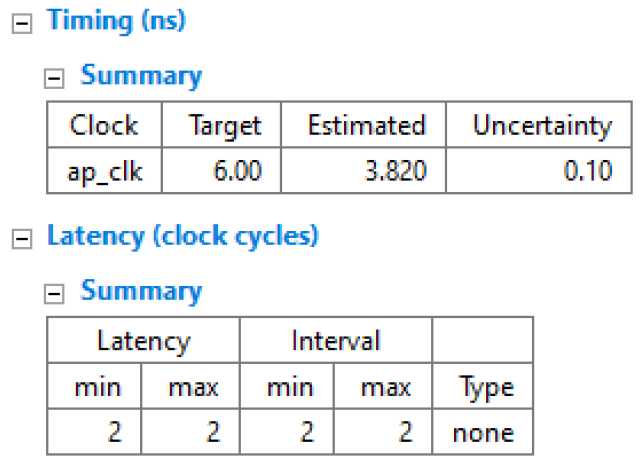


Рисунок 3.3 Performance estimates – summary

Значения соответствуют тем, которые были определены в решении 1.

Использование ресурсов

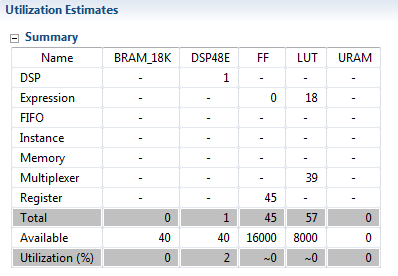


Рисунок 3.4 Utilization estimates – summary

Данный проект будет занимать на микросхеме: 1 DSP блок, где будут задействованы сумматор и умножитель, 45 регистров для хранения и считывания данных (чисел), 57LUT. По сравнению с предыдущим решением выросло на 33 количество используемых регистров и на 20 количество LUT.

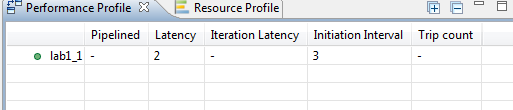


Рисунок 3.5a Performance Profile

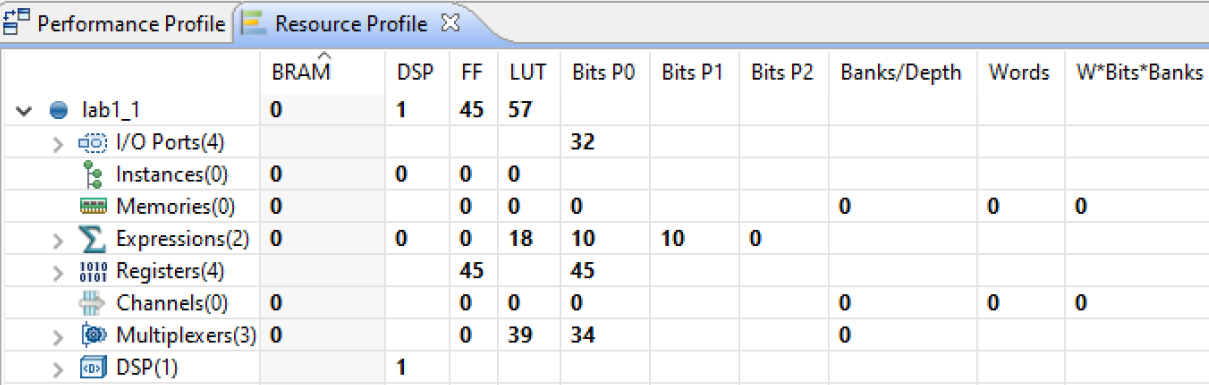


Рисунок 3.5b Resource Profile

Здесь мы также видим отличия, согласно общему отчету о затраченных ресурсах.

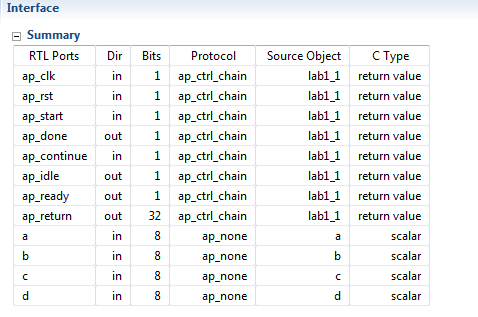


Рисунок 3.6 Interface estimates – summary

По сравнению с решением 1 появился протокол ap\_ctrl\_chain. Порты: ap\_start, ap\_done, ap\_idle, ap\_ready, ap\_clk, ap\_rst, ap\_continue (активен, когда ap\_done завершается для следующей транзакции; дает возможность останавливать дальнейшую обработку при отсутствии возможности обработки новых данных).

Входные порты: a, b, c, d являются 8-битными входами и имеют входы / выходы, протокол ap\_none.

Конструкция имеет 32-битный выходной порт для возврата функции ap\_return.

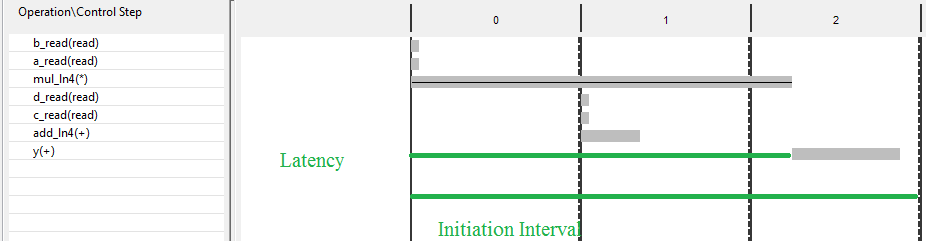


Рисунок 3.7 Schedule viewer

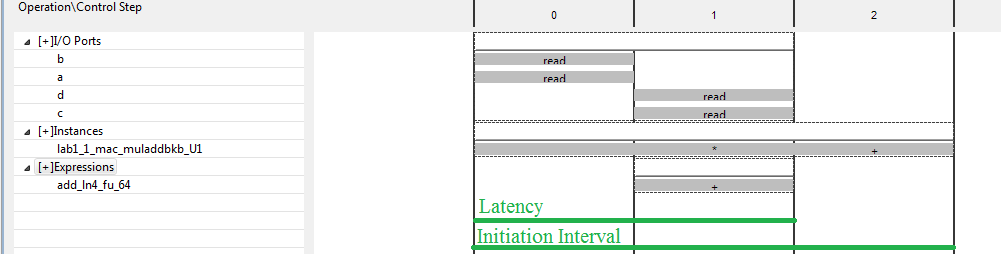


Рисунок 3.8 Resource viewer

## 3.3 C|RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency

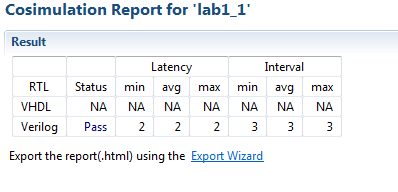


Рисунок 3.9 Отчет о моделировании

Покажем временную диаграмму совместного моделирования

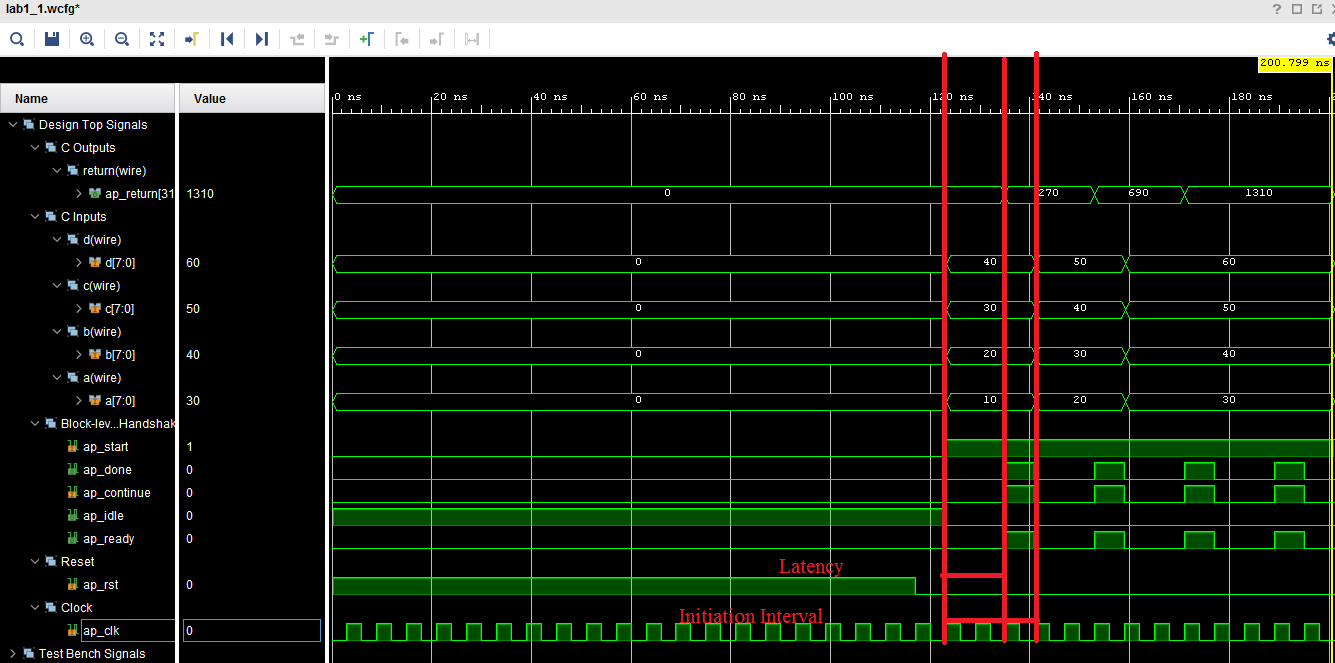


Рисунок 3.10 Временная диаграмма

# 4. Третье решение

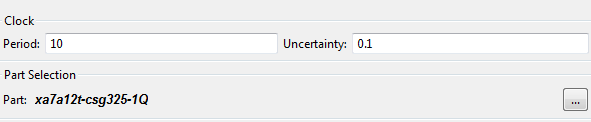


Рисунок 4.1 Настройки третьего решения

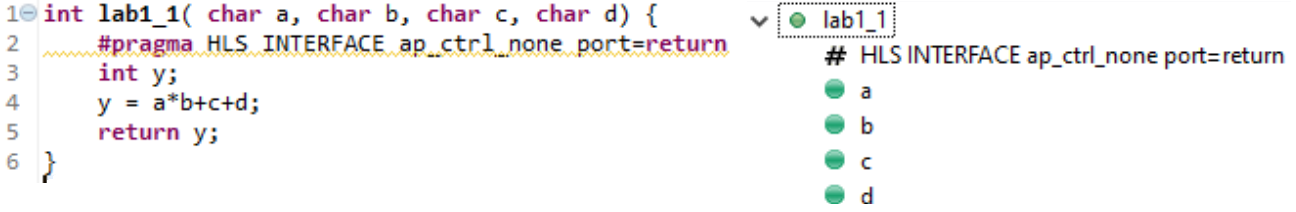


Рисунок 4.2 Листинг кода и директива настройки.

## 4.1 Моделирование

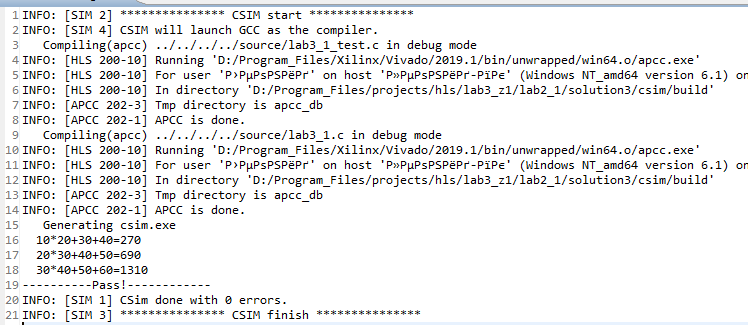


Рисунок 4.3 Результат успешного моделирования

## 4.2 Синтез

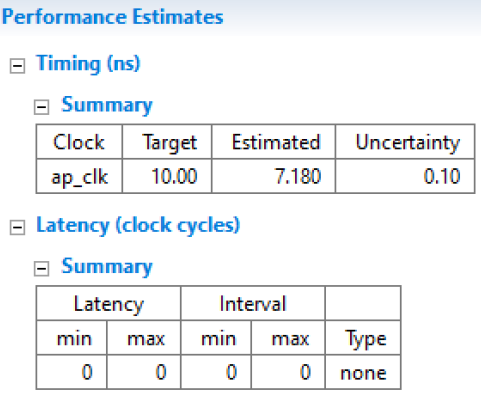


Рисунок 4.4 Performance estimates – summary

По сравнению с предыдущими решениями значение задержки изменилось из-за того, что были заданы другие конфигурации решения. Величина полученной задержки соответствует заданному значению.

Использование ресурсов:

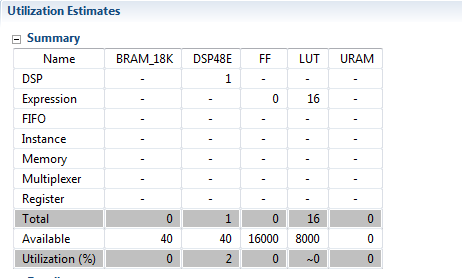


Рисунок 4.5 Utilization estimates – summary

Данный проект будет занимать на микросхеме: 1 DSP блок, где будут задействованы сумматор и умножитель. Количество используемых LUT в выражениях сократилось на 2 LUT и составило 16 LUT.

По сравнению с предыдущими решениями использование регистров и мультиплексеров полностью отсутствует.

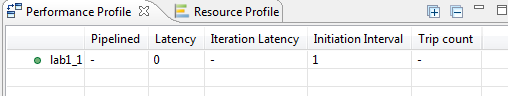


Рисунок 4.6a Performance Profile

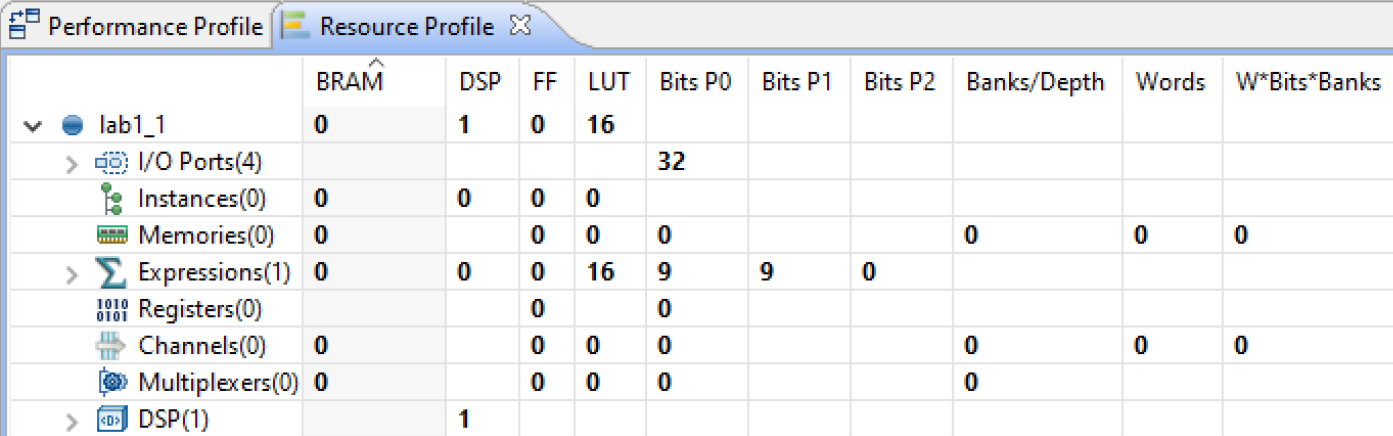


Рисунок 4.6b. Resource Profile

Здесь мы также видим отличия, согласно общему отчету о затраченных ресурсах.

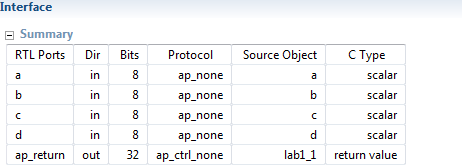


Рисунок 4.7 Interface estimates – summary

По сравнению с предыдущими решениями в данном решении отсутствуют ap\_start, ap\_done, ap\_idle, ap\_ready, ap\_clk, ap\_rst, ap\_continue RTL порты.

Входные порты: a, b, c, d являются 8-битными входами и имеют входы / выходы, протокол ap\_none.

Конструкция имеет 32-битный выходной порт для возврата функции ap\_return.

Заданный протокол ap\_ctrl\_none: No block-level I/O protocol. Когда используется протокол интерфейса ap\_ctrl\_none, никакие протоколы ввода-вывода уровня блока не используются

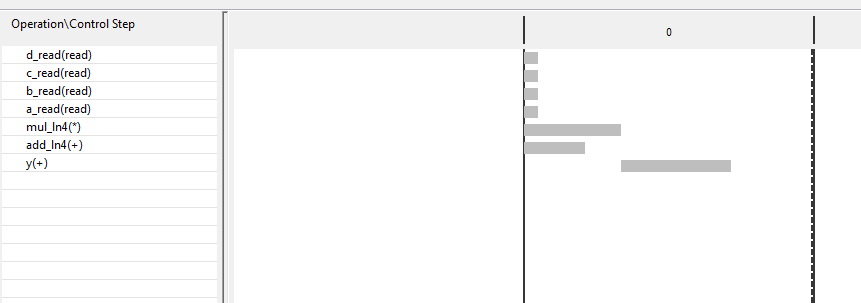


Рисунок 4.8 Schedule viewer

На данном изображении видно, что задержка получения результата отсутствует, а интервал инициализации составляет 1 такт.

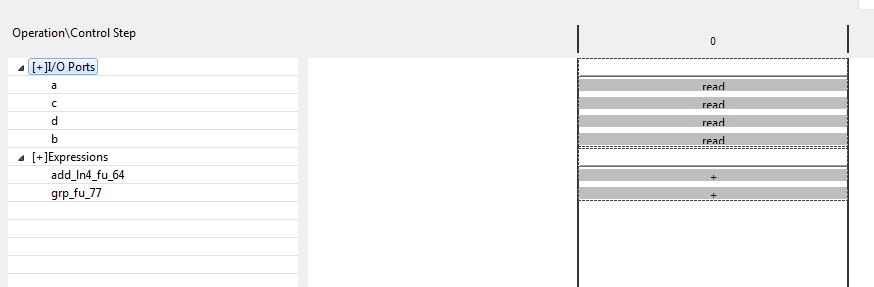


Рисунок 4.9 Resource viewer

## 4.3 C|RTL моделирование

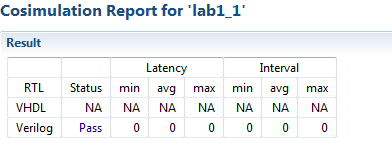


Рисунок 4.10 Отчет о моделировании

Покажем временную диаграмму совместного моделирования

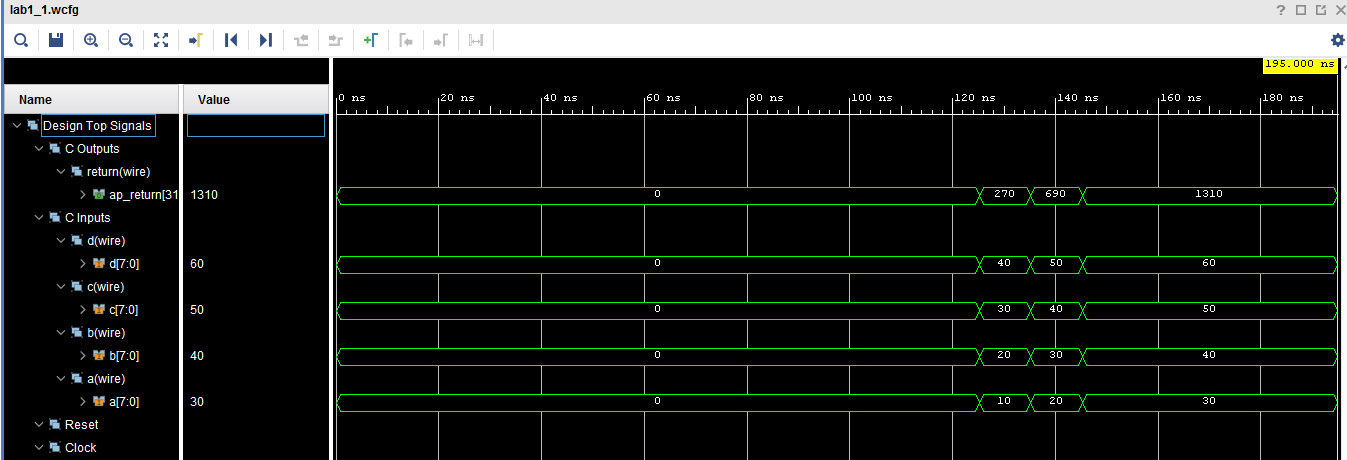


Рисунок 4. Временная диаграмма

# 5.Выводы

Существуют следующие типы протоколов: ap\_ctrl\_none, ap\_ctrl\_hs, и ap\_ctrl\_chain. Они могут быть заданы только для возвращаемого значения функции. ap\_ctrl\_hs задается как протокол по умолчанию. Протокол ap\_ctrl\_chain похож на ap\_ctrl\_hs, но имеет дополнительный входной порт ap\_continue. Если порт ap\_continue является логическим 0, когда функция завершается, блок остановит операцию и следующая транзакция не будет продолжена. Следующая транзакция будет выполняться только тогда, когда ap\_continues иммет значение 1. Режим ap\_ctrl\_none реализует моделирование без какого-либо блочного протокола ввода-вывода.

В данной работе рассмотрены следующие директивы блочного протокола.

1. ap\_ctrl\_none: нет протокола управления вводом / выводом на уровне блоков,
2. ap\_ctrl\_hs: стандартный протокол управления вводом-выводом на уровне блоков,
3. ap\_ctrl\_chain: протокол ввода-вывода для управления цепочками конвейерных блоков.