Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологии

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab1\_z1

Дисциплина: Проектирование реконфигурируемых гибридных вычислительных систем

Выполнил студент Курякин Д.

Гр. 3540901/12001

Руководитель, доцент Антонов А.П.

«\_\_» \_\_\_\_\_\_ 2022

Санкт-Петербург

2022

Оглавление

[1. Задание 3](#_Toc116304427)

[2. Исходный код функции 3](#_Toc116304428)

[3. Исходный код теста 3](#_Toc116304429)

[3.1 Моделирование 3](#_Toc116304430)

[4. Первое решение — Solution1 4](#_Toc116304431)

[4.1 Исходные настройки 4](#_Toc116304432)

[4.2 Синтез 4](#_Toc116304433)

[5. Первое решение — Solution2 8](#_Toc116304434)

[5.1 Исходные настройки 8](#_Toc116304435)

[5.2 Синтез 8](#_Toc116304436)

[6. Сравнение Solution1 и Solution2 11](#_Toc116304437)

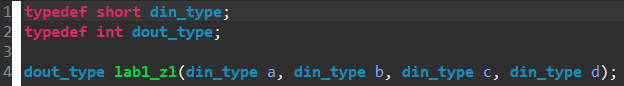
[7. Выводы 11](#_Toc116304438)

1. Задание

Текст задания находится в файле lab1\_z1.docx

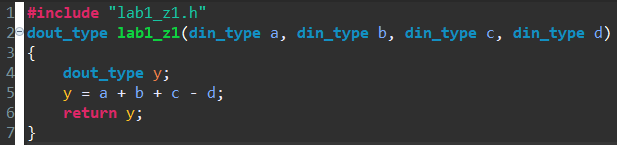
**2. Исходный код функции**

Исходный код заголовочного файла приведен на рисунке 1.



*Рис. 1 Исходный код функции lab1\_z1.h*

Исходный код синтезируемой функции приведен на рисунке 2.



*Рис. 2 Исходный код функции lab1\_z1.c*

Функция принимает 4 аргумента — скаляра считает значение по формуле и возвращает результат расчета.

**3. Исходный код теста**

Исходный код теста для проверки функции lab1\_z1 приведен на рисунке 3.

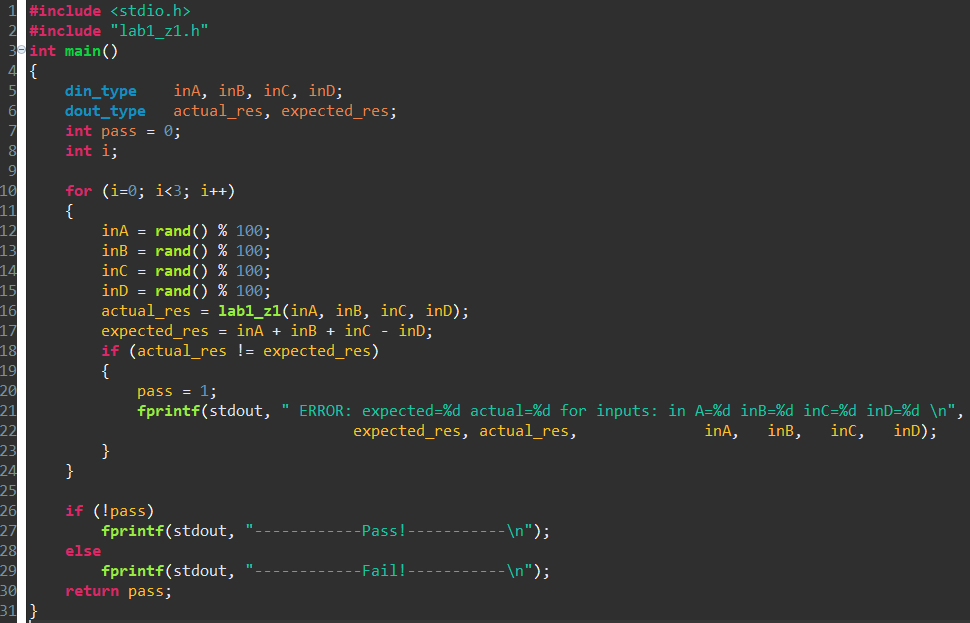
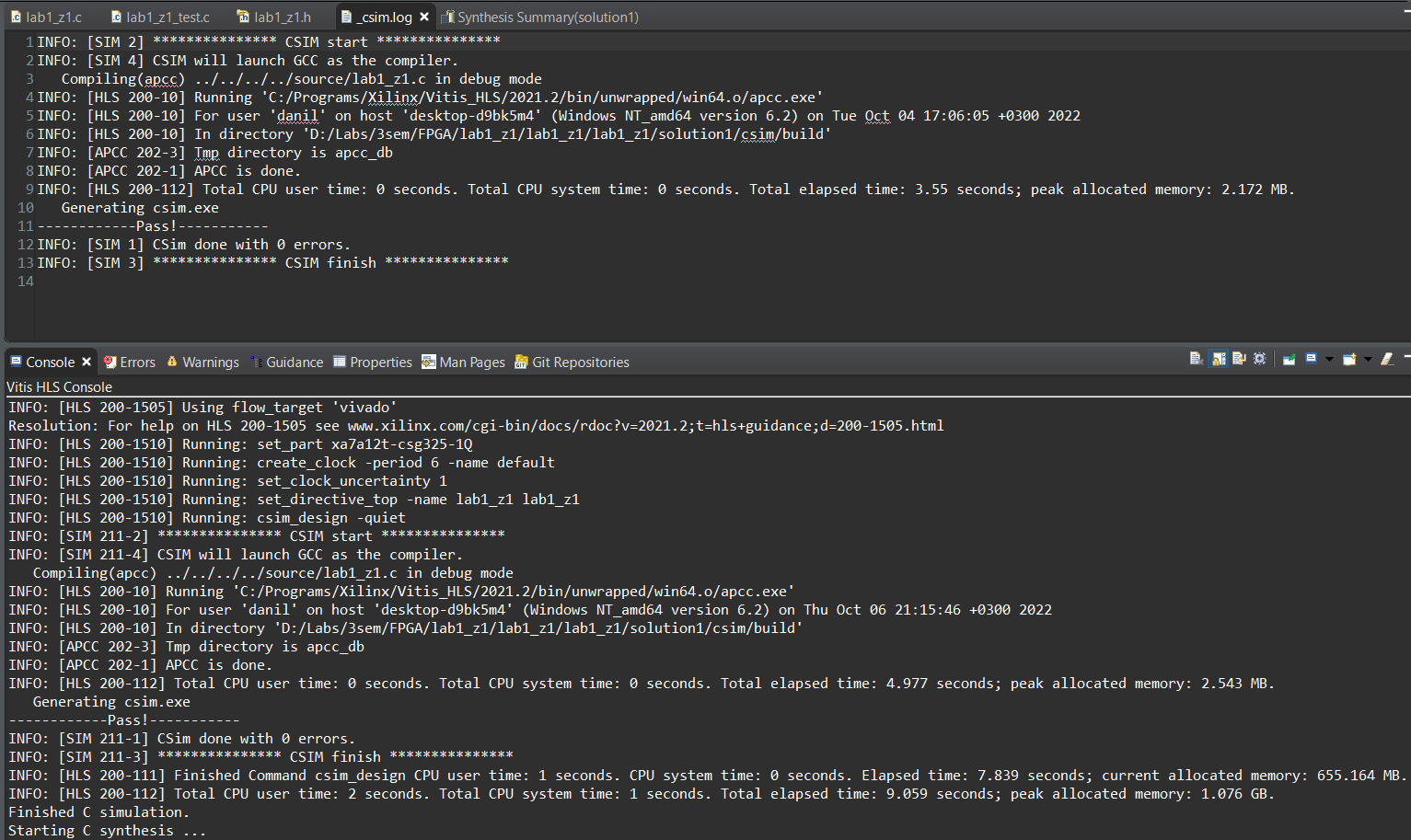


Рис. 3. Исходный код теста lab1\_z1\_test.c

Тест обеспечивает проверку корректной работы функции.

**3.1 Моделирование**

Результаты моделирования исходного кода синтезируемой функции приведены на рисунке 4.



*Рис. 4 Лог результата выполнения моделирования.*

Результаты моделирования показывают, что тест успешно пройден — так как условия теста выполняются (в консоли выведено сообщение -Pass!-).

**4. Первое решение — Solution1**

**4.1 Исходные настройки**

При создании первого решения задаются следующие настройки:

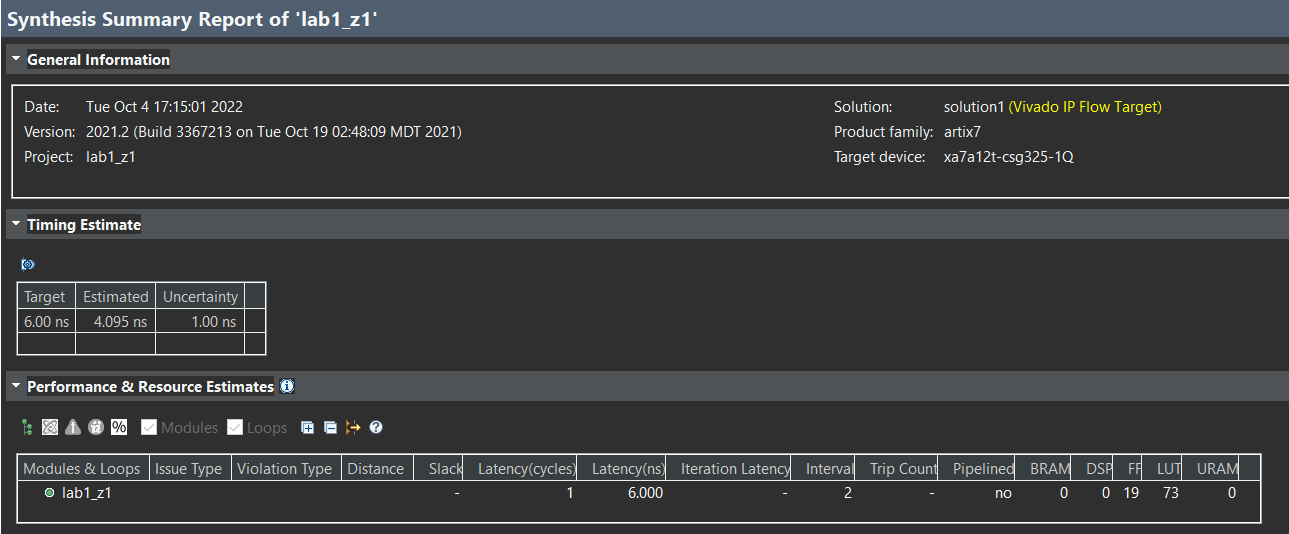
* clock period: 6,
* clock uncertain: 1,
* part: xa7a12tcsg325-1Q

**4.2 Синтез**

Результаты оценки быстродействия Timing/Summary и Latency/Summary представлены на рисунке 4.

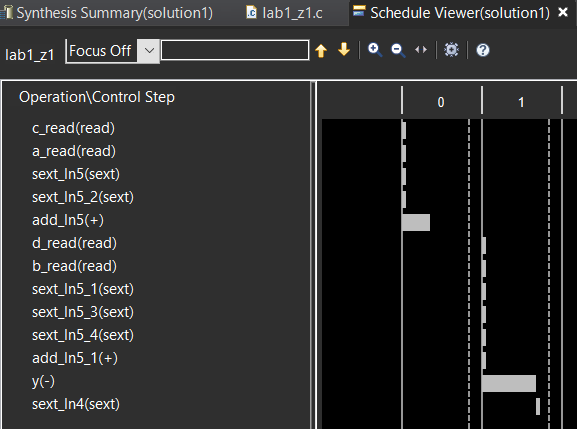
Вопросы:

* Target timing — заданная требуемая частота,
* Estimated timing — достигнутая частота равна 4.095 нс.
* Latency(cycle) — необходимое количество циклов для получения вывода и равно 1.
* Latency(absolute) — необходимое количество времени для получения вывода равное 6 нс.
* Interval – интервал, который показывает через какое количество тактов можно подать новые данные
* FF ­­­­– это количество триггеров для хранения данных.
* LUT ­­­­– это логические модули на основе мультиплексоров.
* Как оценить значение Interval в ns? (Estimated+Uncertainty)\*Interval
  + Наиболее точно это значение можно оценить, посчитав частоту, с которой мы можем подавать данные на функцию
* No в столбце Pipelined означает что функция является неконвейерезирована



*Рис. 5 Оценка быстродействия для solution1*

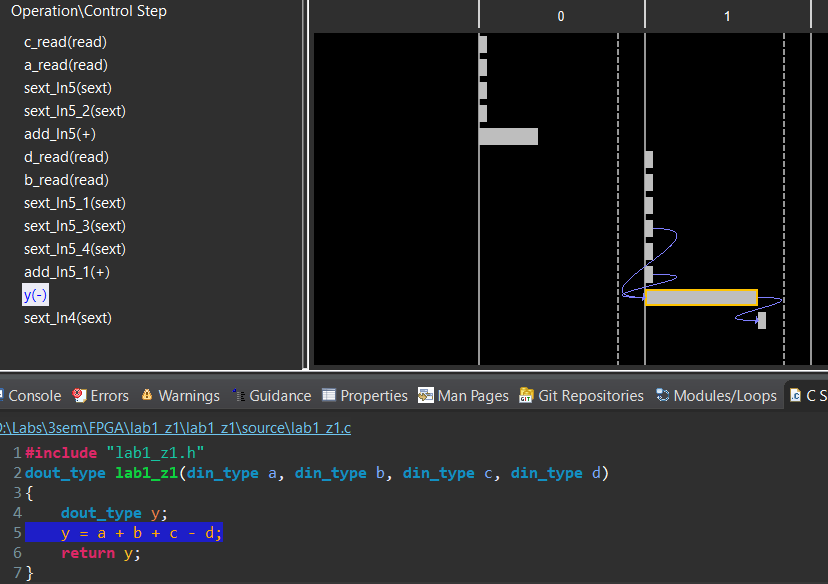
После этого было открыто окно планировщика (Schedule). Планировщик представлен на рис 6.



*Рис. 6 Планировщик Schedule для solution1*

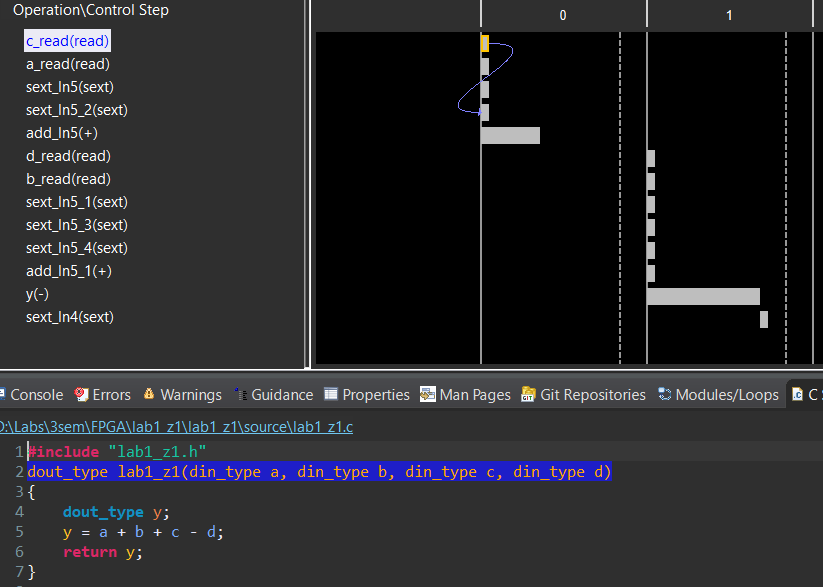
Вопросы:

* Почему представлено два столбца 0 и 1? Ответ: потому что Interval (такт) = 2
* Что изображено в столбцах 0 и 1? Ответ: В 0 фиксируются входные данные, формируется их сумма. В 1 мы получаем выходные данные
* Какие операции будут выполняться при реализации функции? Ответ: Сложение a и b, сложение (a+b)+c, вычитание (a+b+c)-d.
* Сопоставьте операцию y(-) с исходным кодом – приведите снимок экрана.



*Рис. 7 Сопоставление операции y(-)* *для solution1*

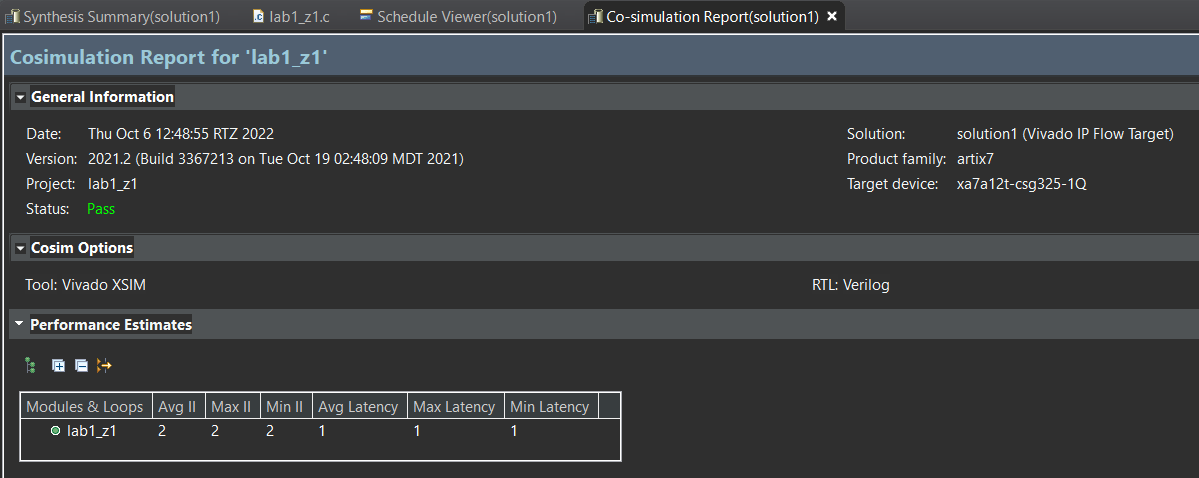
* Сопоставьте операцию c\_read(read) с исходным кодом.



*Рис. 8 Сопоставление операции c\_read(read) для solution1*

* Какие аргументы функции будут просуммированы оператором add\_In6(+)? Ответ: a и c.
* Какие аргументы функции будут просуммированы оператором add\_In6\_1(+)? Ответ: a+c и b

После этого была выполнена процедура Co-simulation.



*Рис. 9 Результаты Co-simulation для solution1*

Вопросы:

* Что означает Avg, Max, Min? Ответ: Средний, максимальный и минимальный Iteration Interval
* Почему столбы Avr II, Max II, Min II содержат одинаковые результаты? Ответ: потому что число шагов выполнения алгоритма не зависит от исходных данных.

После этого была получена временная диаграмма, представленная на рисунке ниже:



*Рис. 10 Временная диаграмма для solution1*

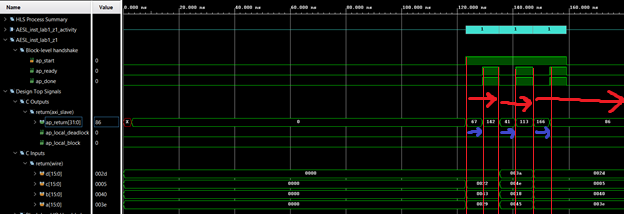
Вопросы:

* Сколько циклов запуска аппаратной реализации созданной функции приведено на временной диаграмме? Ответ: 3
* В какие моменты времени на шине ap\_return[31:0] отображаются правильные выходные значения? (отметьте их на временной диаграмме).



*Рис. 11 Временная диаграмма для solution1*

* Изобразите на приведенной вами временной диаграмме промежуток времени, определяемый как Latency и определяемый как Iteration Interval (II) для всех циклов запуска созданной функции(синий – Latency, красный - Iteration Interval II).



*Рис. 12 Временная диаграмма для solution1*

**5. Первое решение — Solution2**

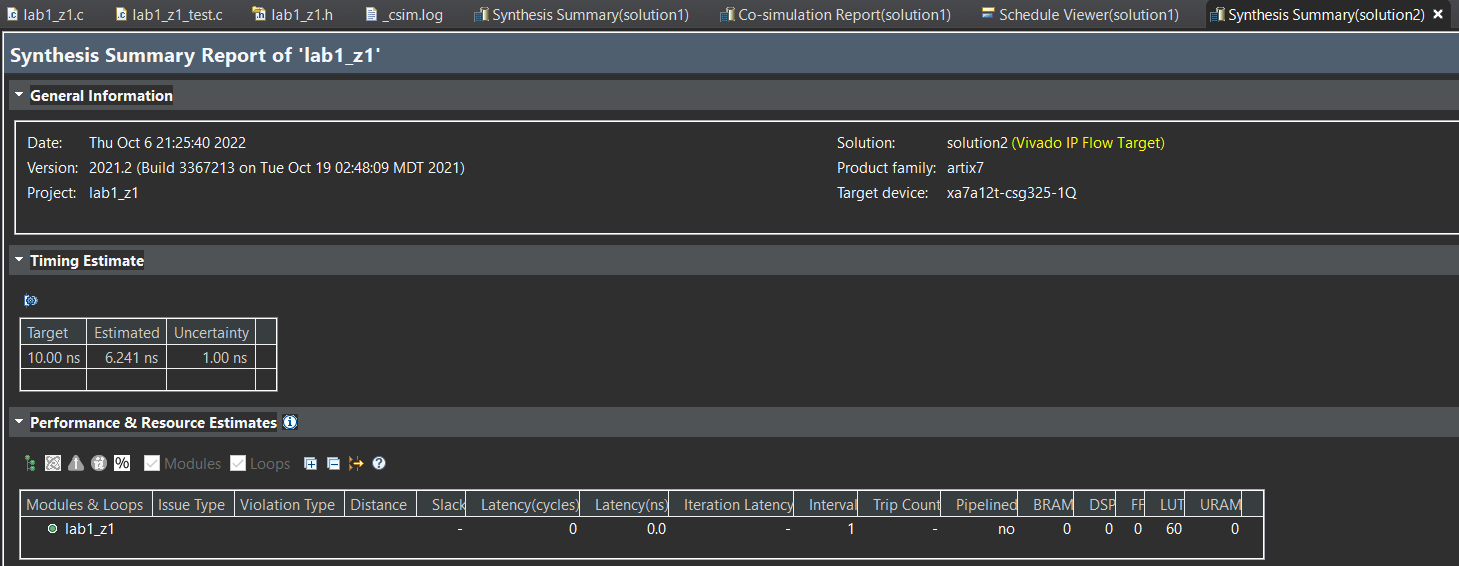
**5.1 Исходные настройки**

При создании второго решения задаются следующие настройки:

* clock period: 10,
* clock uncertain: 1,
* part: xa7a12tcsg325-1Q

**5.2 Синтез**

Было создано новое решение и проделаны все те же самые шаги что и с Solution1. Сначала был запущен синтез. В настройках был изменён Clock Period с 6 на 10. Результаты синтеза приведены ниже.

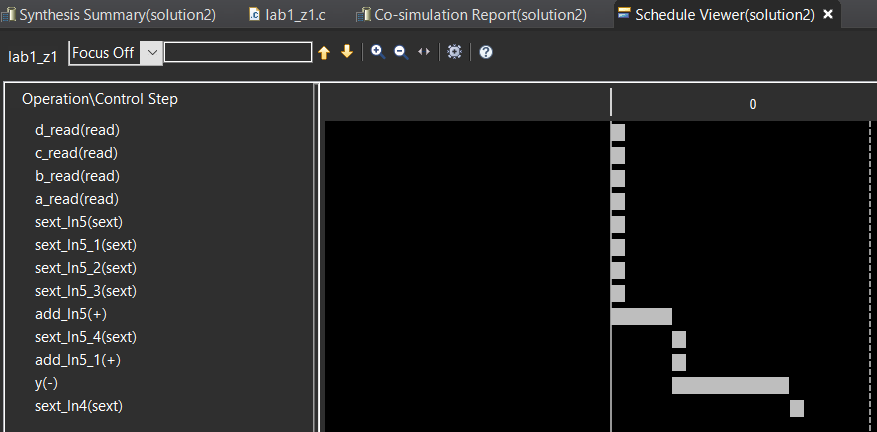


*Рис. 13 Оценка быстродействия для solution2*

Вопросы:

* Почему Latency (cycles) = 0? Ответ: потому что как только были поданы данные, тут же, в течении этого же периода этого тактового сигнала будет получен результат. (Комбинационная схема)
* Сколько триггеров было использовано при реализации функции? Ответ: 0

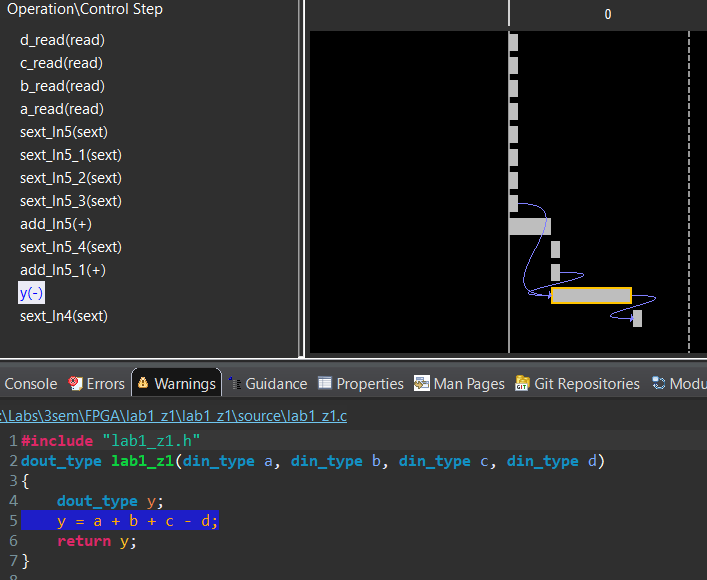
Окно планировщика выглядит представлено ниже.



*Рис. 14 Оценка быстродействия для solution2*

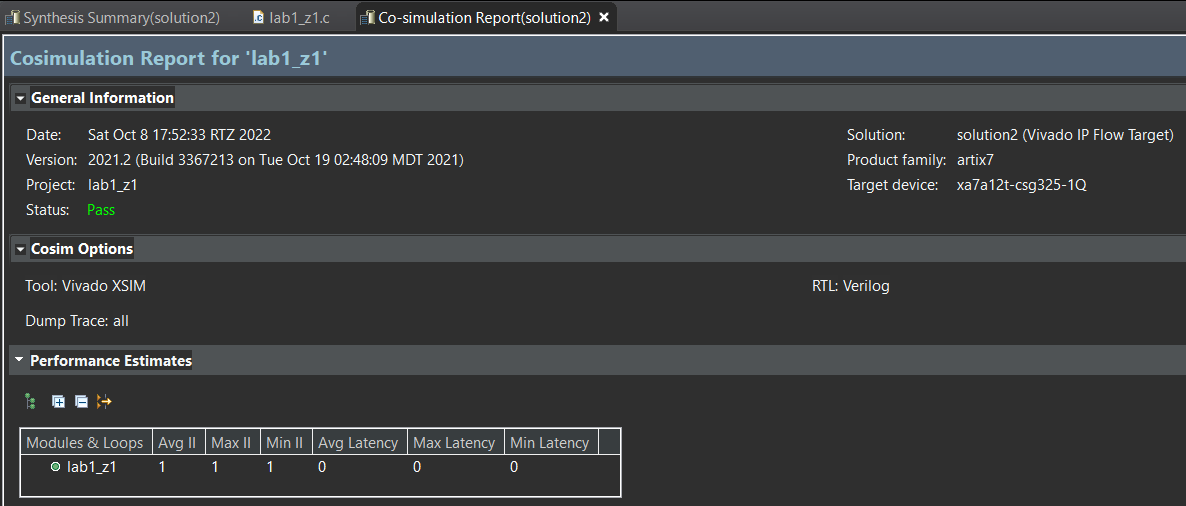
Вопросы:

* Почему представлен только один столбец 0? Ответ: Все процессы (считывание, сложение, вычитание) производятся в рамках одного периода
* Сопоставьте операцию y(-) с исходным кодом – приведите снимок экрана.



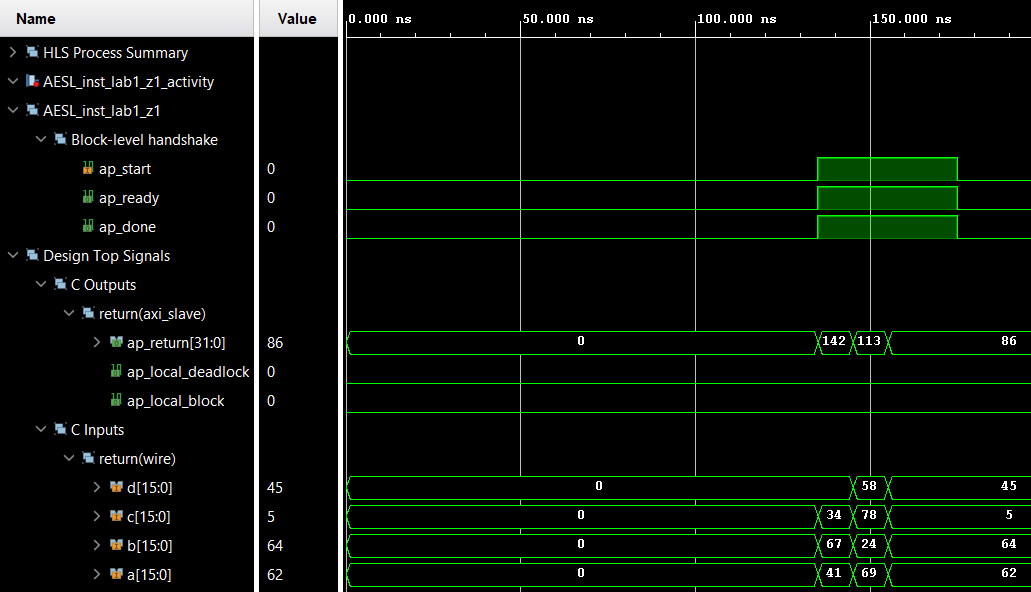
*Рис. 15 Планировщик Schedule для solution2*

Далее выполнена Co-simulation. Результаты показаны ниже.



*Рис. 16 Планировщик Schedule для solution2*

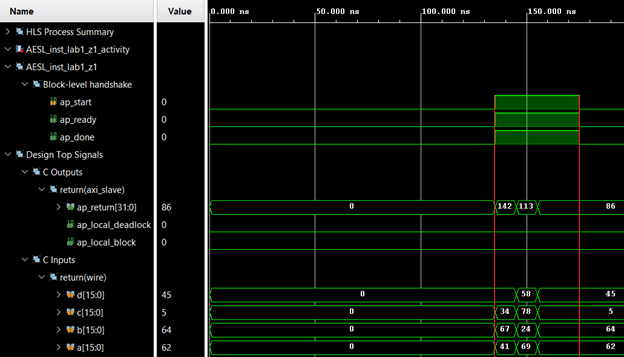
После этого была получена временная диаграмма, представленная на рисунке ниже:



*Рис. 17 Результаты Co-simulation для solution2*

Вопросы

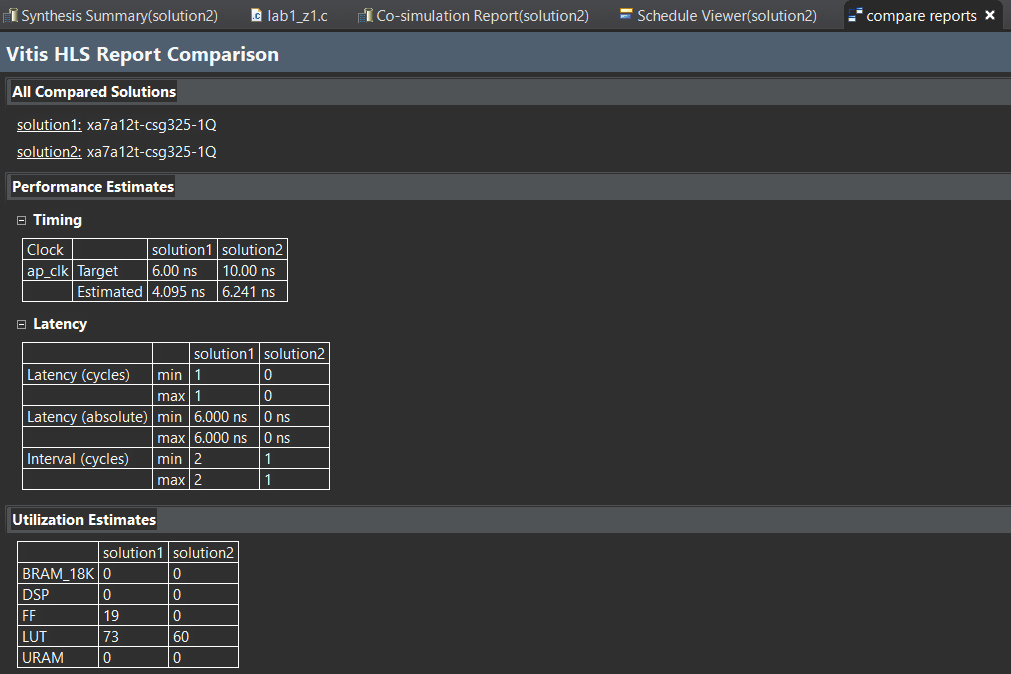
* Сколько циклов запуска аппаратной реализации созданной функции приведено на временной диаграмме? Ответ: 0
* В какие моменты времени на шине ap\_return[31:0] отображаются правильные выходные значения? Ответ: В течении 10нс. Однако по временной диаграмме кажется, что мгновенно, так как у нас поведенческое моделирование.
* Почему на временной диаграмме нет тактового сигнала ap\_clk, который был на временной диаграмме решения Solution1? Ответ: потому что Solution2 представляет собой комбинационную схему
* Изобразите на приведенной вами временной диаграмме промежуток времени, определяемый как Iteration Interval (II) для всех циклов запуска созданной функции.



*Рис. 18 Временная диаграмма для solution2*

**6. Сравнение Solution1 и Solution2**

Была выполнена команда Compare Reports. После которой получаем результаты сравнения двух решений.



*Рис. 19 Сравнение двух решений*

Вопросы:

* Как определить минимальный период времени между моментами подачи новых данных на вход аппаратно реализованной функции для решения Solution1 и Solution2? Ответ: (Estimated+Uncertanty)\*II
* Какой минимальный период времени между моментами поступления новых данных для решения Solution1 и Solution2?
  + Solution1 = (4.095+1.0) \*2 = 10.19 ns
  + Solution2 = (6.241+1) = 7.241 ns
* Какое решение более производительное?
  + Второе решение более производительное, так как благодаря тому, что используется комбинационная схема временные и аппаратные затраты данного решения меньше, чем у Solution1

**7. Выводы**

В данной работе были реализованы два решения с разной частотой, которые могут отличаться в зависимости от задаваемого параметра.