САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа lab5\_z1

Дисциплина:

«Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: Введение в Optimizing Structure for performance

Выполнил:

Дроздов Н.Д.

Группа: 3540901/02001

Преподаватель: А. П. Антонов

Санкт-Петербург

2021

**Оглавление**

[1. Задание 4](#_Toc90332202)

[2. Исходный код функции 4](#_Toc90332203)

[3. Исходный код теста 4](#_Toc90332204)

[4. Исходный код командного файла 5](#_Toc90332205)

[5. Результаты исследования (1) 6](#_Toc90332206)

[5.1 Solution1 с «resource -core RAM\_1P "lab5\_z1" d\_in» 6](#_Toc90332207)

[5.2 Solution2 с «resource -core RAM\_2P "lab5\_z1" d\_in» 7](#_Toc90332208)

[5.3 Solution3 с «resource -core RAM\_1P "lab5\_z1" d\_in» и «array\_partition -type block -factor 4 "lab5\_z1" d\_in». 9](#_Toc90332209)

[5.4 Solution4 с «resource -core RAM\_1P "lab5\_z1" d\_in» и «array\_reshape -type block -factor 4 "lab5\_z1" d\_in» 11](#_Toc90332210)

[5.5 Solution5 с «resource -core RAM\_2P "lab5\_z1" d\_in» и «array\_partition -type block -factor 2 "lab5\_z1" d\_in» 13](#_Toc90332211)

[5.6 Solution6 с «resource -core RAM\_2P "lab5\_z1" d\_in» и «array\_ reshape -type block -factor 2 "lab5\_z1" d\_in» 14](#_Toc90332212)

[5.7 Solution7 с «resource -core RAM\_2P "lab5\_z1" d\_in» и «array\_ partition -type block -factor 2 "lab5\_z1" d\_in» и «unroll -skip\_exit\_check -factor 2 "lab5\_z1/for\_label0" d\_in» 16](#_Toc90332213)

[6. Результаты сравнения всех исследований 17](#_Toc90332214)

[7. Результаты исследования (2) 18](#_Toc90332215)

[7.1 Запуск модифицированного теста на ПК 19](#_Toc90332216)

[8. Выводы 21](#_Toc90332217)

**Список иллюстраций**

[Рисунок 1- Исходный код Исходный код функции lab5\_z1 (файл lab5\_z1.c) 5](#_Toc90332156)

[Рисунок 2 - Исходный код теста (файл lab5\_z1\_test.c) 6](#_Toc90332157)

[Рисунок 3 - Исходный код командного файла для создания проекта (la5\_z1.tcl) 7](#_Toc90332158)

[Рисунок 4 - Временная диаграмма solution1 8](#_Toc90332159)

[Рисунок 5 – Временная диаграмма solution1\_1 8](#_Toc90332160)

[Рисунок 6 - Сравнение решений Solution1 и Solution2 9](#_Toc90332161)

[Рисунок 7 - Временная диаграмма solution2 9](#_Toc90332162)

[Рисунок 8 - Временная диаграмма solution2\_1 10](#_Toc90332163)

[Рисунок 9 - Сравнение решений Solution1, Solution2, Solution3 11](#_Toc90332164)

[Рисунок 10 - Временная диаграмма solution3 12](#_Toc90332165)

[Рисунок 11 - Временная диаграмма solution3\_1 12](#_Toc90332166)

[Рисунок 12 - Сравнение решений Solution1, Solution2, Solution3, Solution4 13](#_Toc90332167)

[Рисунок 13 - Временная диаграмма solution4 13](#_Toc90332168)

[Рисунок 14 - Временная диаграмма solution4\_1 14](#_Toc90332169)

[Рисунок 15 - Сравнение решений Solution3 и Solution5 14](#_Toc90332170)

[Рисунок 16 - Временная диаграмма solution5 15](#_Toc90332171)

[Рисунок 17 - Временная диаграмма solution5\_1 15](#_Toc90332172)

[Рисунок 18 - Сравнение решений Solution2, Solution4 и Solution6 16](#_Toc90332173)

[Рисунок 19 - Временная диаграмма Solution6 16](#_Toc90332174)

[Рисунок 20 - Временная диаграмма solution6\_1 17](#_Toc90332175)

[Рисунок 21 - Сравнение решений Solution4 и Solution7 17](#_Toc90332176)

[Рисунок 22 - Временная диаграмма Solution7 18](#_Toc90332177)

[Рисунок 23 - Временная диаграмма solution7\_1 18](#_Toc90332178)

[Рисунок 24 - Сравнение всех решений 19](#_Toc90332179)

[Рисунок 25 - Сравнение всех решений в виде таблицы и графика 19](#_Toc90332180)

[Рисунок 26 - Сравнение всех решений 20](#_Toc90332181)

[Рисунок 27 - Сравнение всех решений в виде таблицы и графика 20](#_Toc90332182)

[Рисунок 28 - Параметры ПК (Частота = 2.7 Гц) 21](#_Toc90332183)

[Рисунок 29 - Исходный код модифицированного теста для ПК 21](#_Toc90332184)

[Рисунок 30 - Временные показатели для модифицированного теста для foo\_m 22](#_Toc90332185)

1. Задание

Задание описано в файле «Задание lab5\_z1.docx», лежащее в рабочей папке.

1. Исходный код функции

Исходный код синтезируемой функции lab5\_z1 приведен на рисунке 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1- Исходный код Исходный код функции lab5\_z1 (файл lab5\_z1.c)

Функция принимает два аргумента массива типа int — вычисляет сумму отдельных элементов массивов и записывает результат в выходной массив.

1. Исходный код теста

Исходный код теста для проверки функции lab5\_z1 приведен на рисунке 2. Тест обеспечивает проверку корректной работы функций.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 - Исходный код теста (файл lab5\_z1\_test.c)

1. Исходный код командного файла

На рисунке 3 представлен текст команд для автоматизированного создания следующего варианта аппаратной реализации:

1. Для sol1 задается clock period 20; устанавливается директива «resource -core RAM\_1P» для входного массива d\_in.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 - Исходный код командного файла для создания проекта (la5\_z1.tcl)

1. Результаты исследования (1)

5.1 Solution1 с «resource -core RAM\_1P "lab5\_z1" d\_in»

На рисунке 4 представлена временная диаграмма solution1. На временной диаграмме имеются пустые промежутки у считывания данных. В этот момент можно брать уже следующие данные из массива.

Для того, чтобы решить этот вопрос, нужно добавить директиву конвейеризации. Временная диаграмма с добавлением «Pipeline» представлена на рисунке 5 для решения solution1\_1.

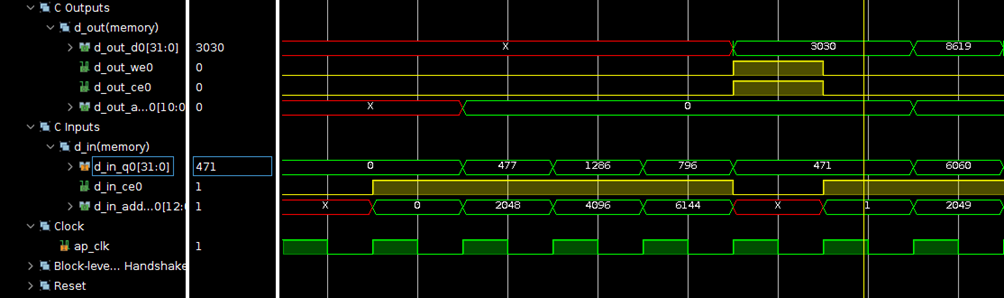


Рисунок 4 - Временная диаграмма solution1

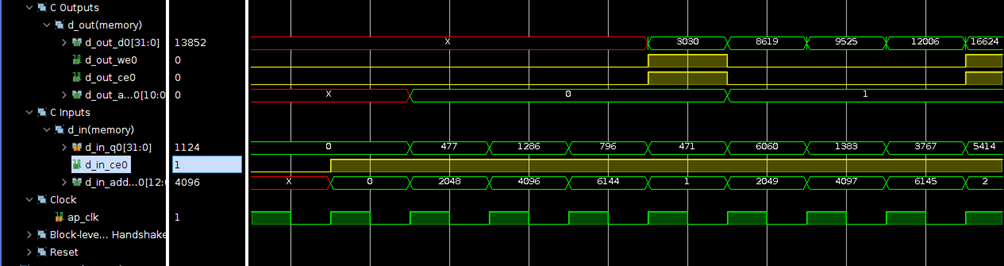


Рисунок 5 – Временная диаграмма solution1\_1

5.2 Solution2 с «resource -core RAM\_2P "lab5\_z1" d\_in»

На рисунке 6 приведено сравнение временных и аппаратных параметров.

Разделение входного массива на два порта уменьшает количество тактов до записи в выходной массив. Поэтому и уменьшается общее количество тактов.

На рисунке 7 приведена временная диаграмма Solution2. На временной диаграмме имеются пустые промежутки во время записи данных у считывания данных. В этот момент можно брать уже следующие данные из массива.

Для решения данного вопроса нужно добавить директиву конвейеризации. Временная диаграмма с добавлением «Pipeline» представлена на рисунке 8 для решения Solution2\_1.

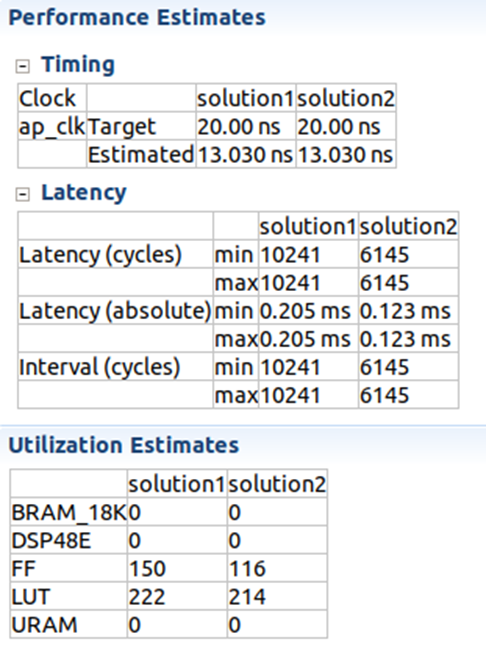


Рисунок 6 - Сравнение решений Solution1 и Solution2

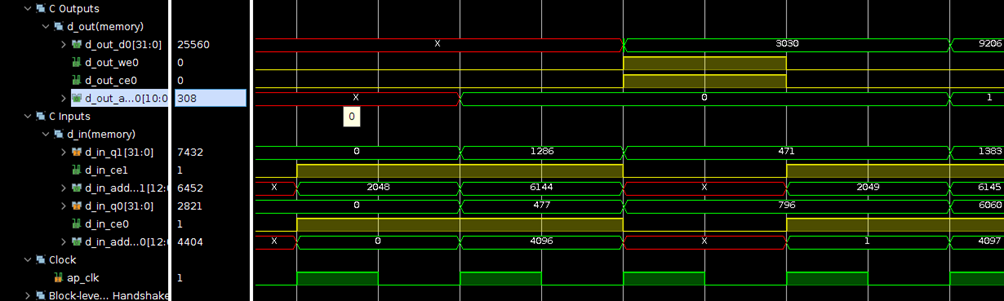


Рисунок 7 - Временная диаграмма solution2

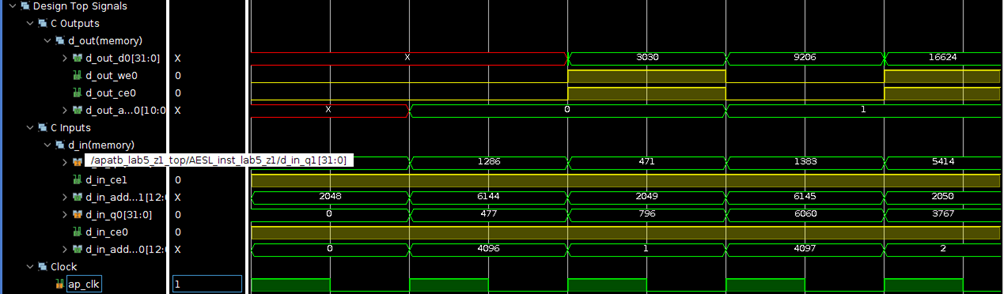


Рисунок 8 - Временная диаграмма solution2\_1

5.3 Solution3 с «resource -core RAM\_1P "lab5\_z1" d\_in» и «array\_partition -type block -factor 4 "lab5\_z1" d\_in».

На рисунке 9 приведено сравнение временных и аппаратных параметров.

Разделение входного массива одиночного порта на четыре подмассива уменьшает количество тактов до записи в выходной массив, и выходной массив теперь записывается одним тактом в два порта. В результате уменьшается общее количество тактов.

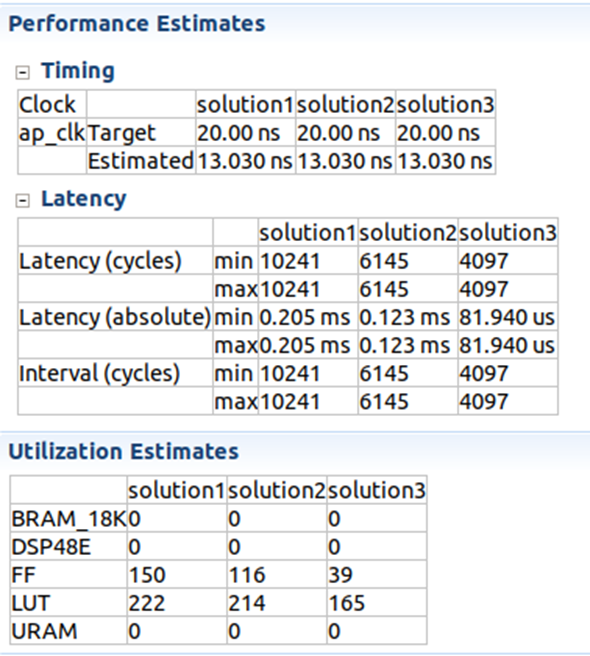


Рисунок 9 - Сравнение решений Solution1, Solution2, Solution3

На рисунке 10 представлена временная диаграмма solution3.

На временной диаграмме имеются пустые промежутки во время записи данных у считывания данных. В этот момент можно брать уже следующие данные из массива. Для того, чтобы решить данный вопрос, нужно добавить директиву конвейеризации.

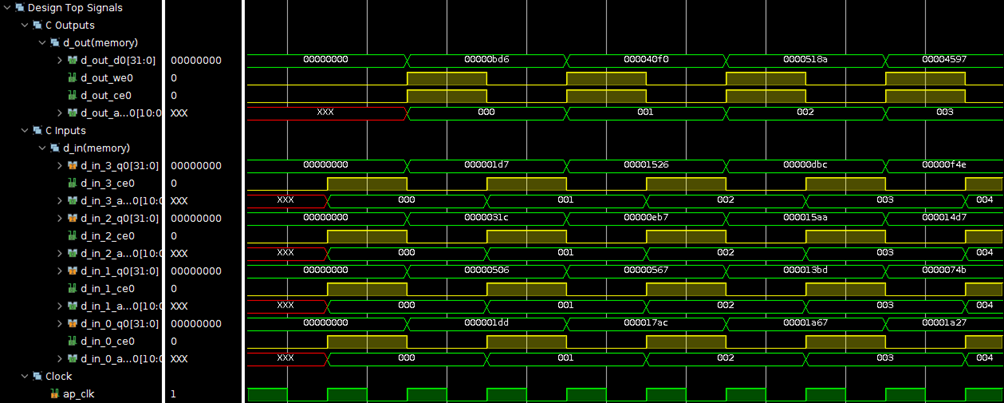


Рисунок 10 - Временная диаграмма solution3

Временная диаграмма с добавлением «Pipeline» представлена на рисунке 11 для решения Solution3\_1.

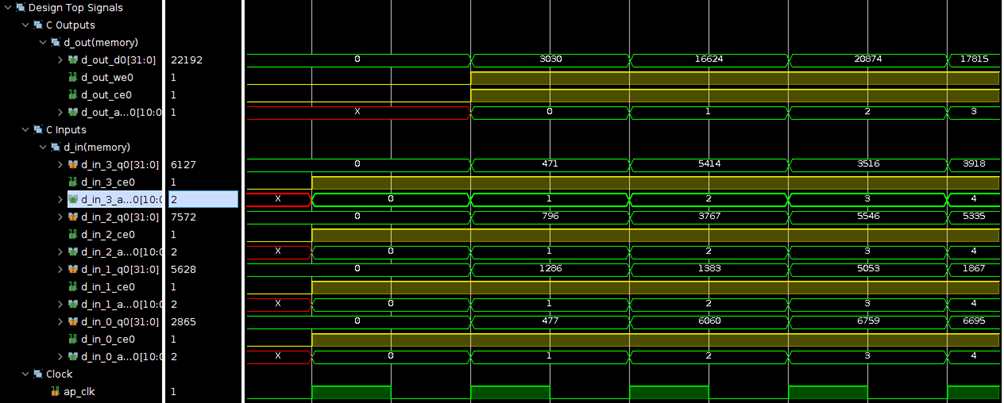


Рисунок 11 - Временная диаграмма solution3\_1

* 1. Solution4 с «resource -core RAM\_1P "lab5\_z1" d\_in» и «array\_reshape -type block -factor 4 "lab5\_z1" d\_in»

На рисунке 12 представлено сравнение временных и аппаратных параметров. Разделение входного массива упаковка одного порта в многоразрядное значение уменьшает количество тактов до записи в выходной массив, и выходной массив теперь записывается одним тактом в два порта. В результате уменьшается общее количество тактов.

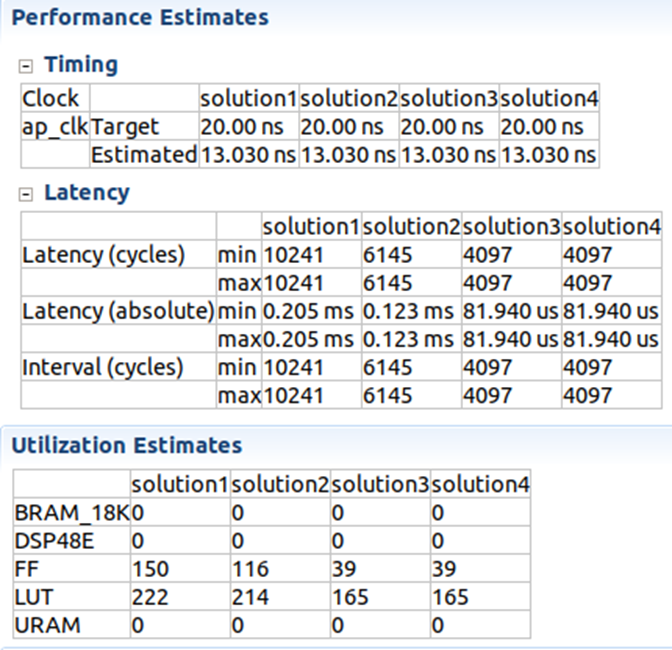


Рисунок 12 - Сравнение решений Solution1, Solution2, Solution3, Solution4

На рисунке 13 представлена временная диаграмма solution4. На временной диаграмме имеются пустые промежутки во время записи данных у считывания данных. В этот момент можно брать уже следующие данные из массива. Для того, чтобы решить этот вопрос, нужно добавить директиву конвейеризации.

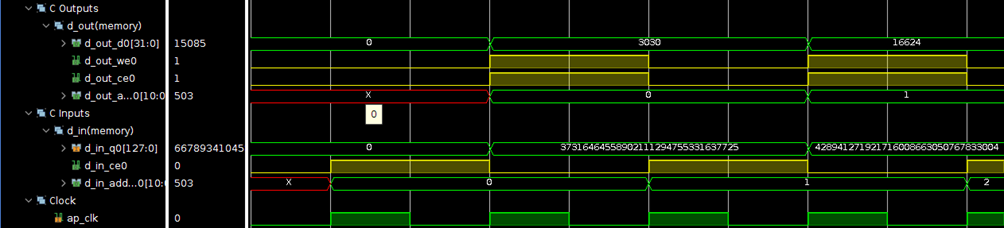


Рисунок 13 - Временная диаграмма solution4

Временная диаграмма с добавлением «Pipeline» представлена на рисунке 14 для решения solution4\_1.

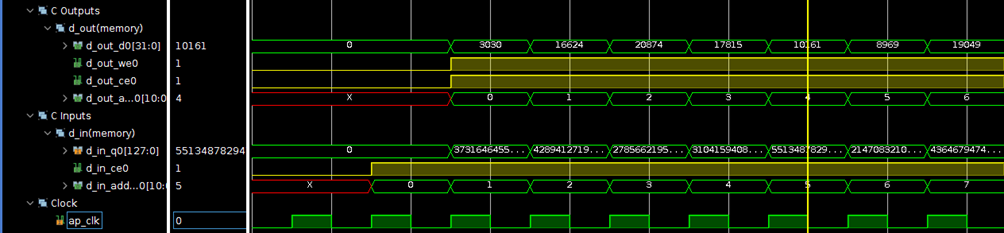


Рисунок 14 - Временная диаграмма solution4\_1

* 1. Solution5 с «resource -core RAM\_2P "lab5\_z1" d\_in» и «array\_partition -type block -factor 2 "lab5\_z1" d\_in»

На рисунке 15 представлено сравнение временных и аппаратных параметров.

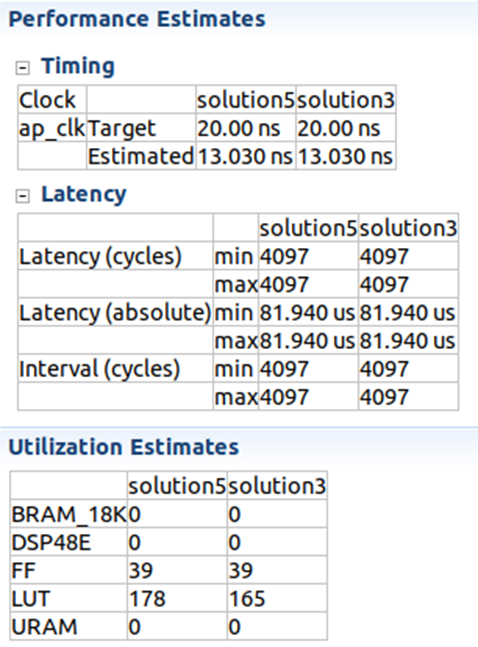


Рисунок 15 - Сравнение решений Solution3 и Solution5

На рисунке 16 представлена временная диаграмма Solution5. На временной диаграмме имеются пустые промежутки во время записи данных у считывания данных. В этот момент можно брать уже следующие данные из массива.

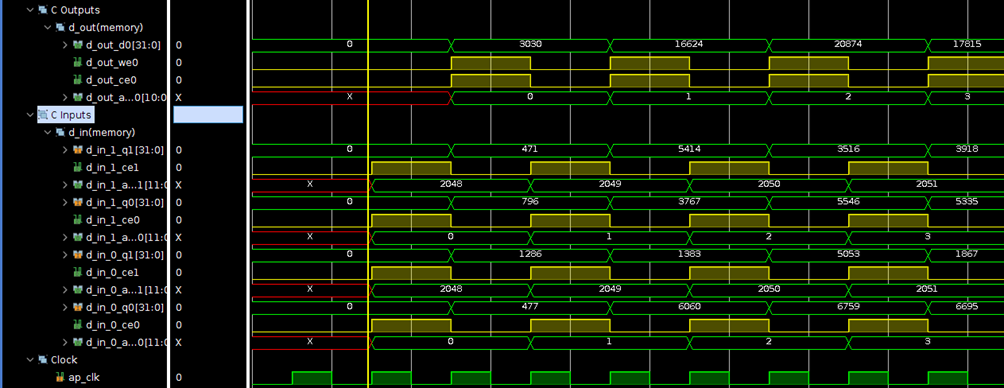


Рисунок 16 - Временная диаграмма solution5

Для того, чтобы решить этот вопрос, нужно добавить директиву конвейеризации. Временная диаграмма с добавлением «pipeline» представлена на рисунке 17 для решения solution5\_1. Если изменить block до 4, то производительность не изменится.

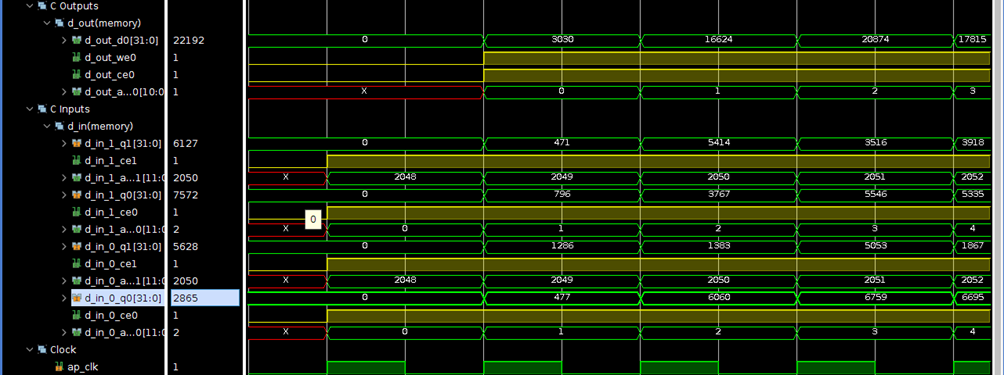


Рисунок 17 - Временная диаграмма solution5\_1

* 1. Solution6 с «resource -core RAM\_2P "lab5\_z1" d\_in» и «array\_ reshape -type block -factor 2 "lab5\_z1" d\_in»

На рисунке 18 представлено сравнение временных и аппаратных параметров.

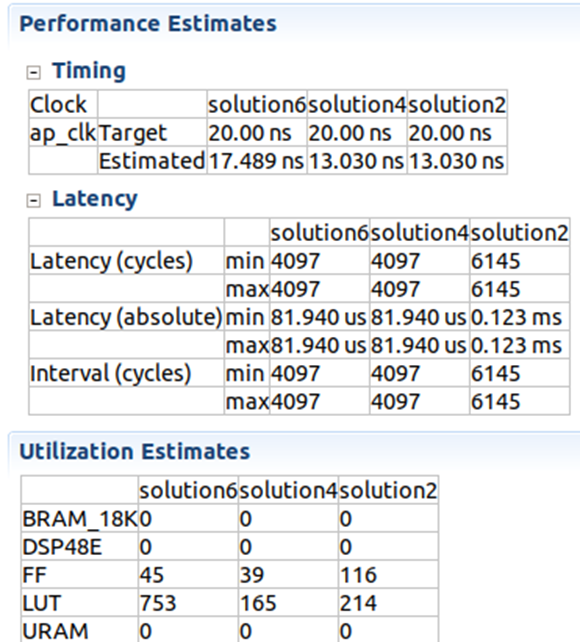


Рисунок 18 - Сравнение решений Solution2, Solution4 и Solution6

На рисунке 19 представлена временная диаграмма Solution6. На временной диаграмме имеются пустые промежутки во время записи данных у считывания данных. В этот момент можно брать уже следующие данные из массива.

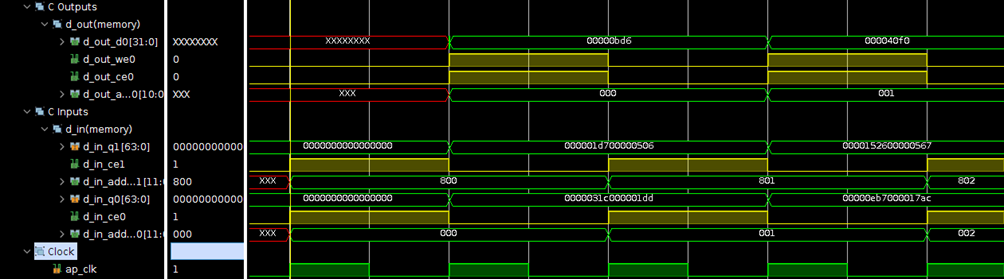


Рисунок 19 - Временная диаграмма Solution6

Для решения данного вопроса нужно добавить директиву конвейеризации. Временная диаграмма с добавлением «Pipeline» представлена на рисунке 20 для решения Solution6\_1. Если изменить block до 4, то производительность не изменится.

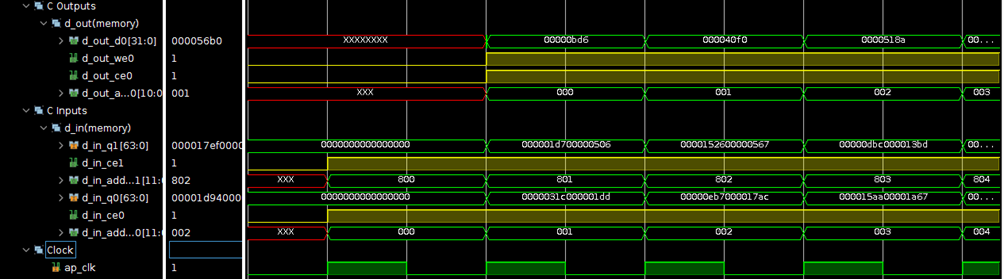


Рисунок 20 - Временная диаграмма solution6\_1

* 1. Solution7 с «resource -core RAM\_2P "lab5\_z1" d\_in» и «array\_ partition -type block -factor 2 "lab5\_z1" d\_in» и «unroll -skip\_exit\_check -factor 2 "lab5\_z1/for\_label0" d\_in»

На рисунке 21 представлено сравнение временных и аппаратных параметров.

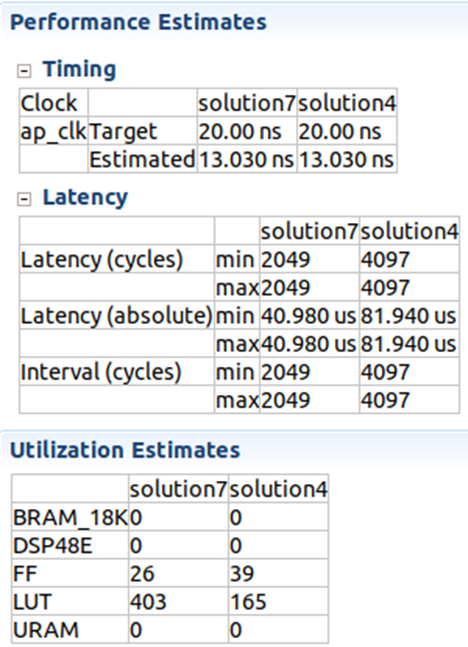


Рисунок 21 - Сравнение решений Solution4 и Solution7

На рисунке 22 представлена временная диаграмма Solution7.

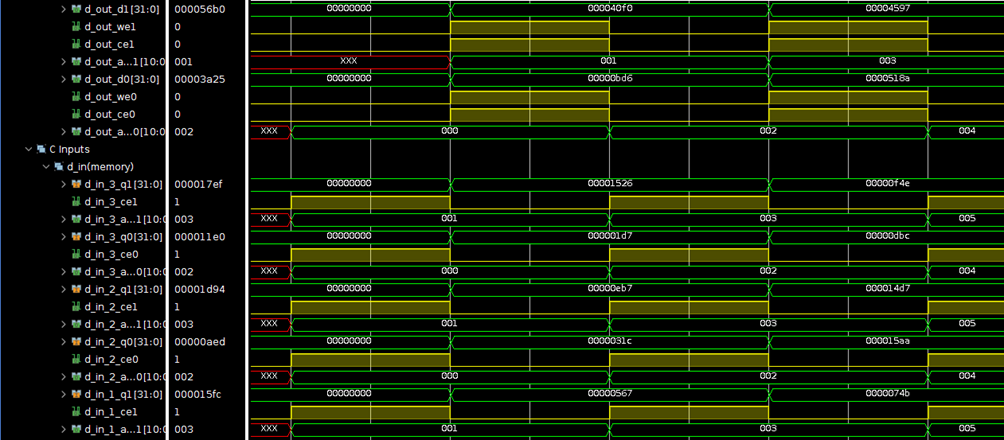


Рисунок 22 - Временная диаграмма Solution7

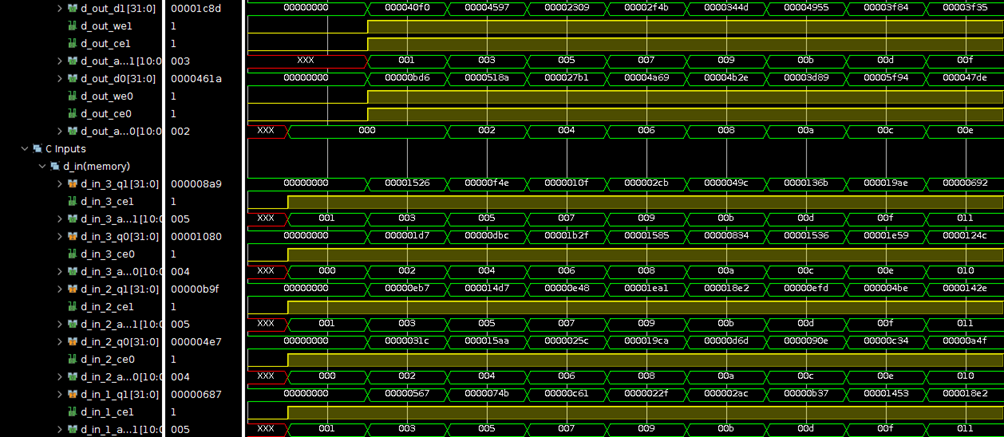


Рисунок 23 - Временная диаграмма solution7\_1

На временной диаграмме имеются пустые промежутки во время записи данных у считывания данных. В этот момент можно брать уже следующие данные из массива.

Чтобы решить этот вопрос, нужно добавить директиву конвейеризации. Временная диаграмма с добавлением «Pipeline» представлена на рисунке 23 для решения Solution7\_1.

1. Результаты сравнения всех исследований

На рисунке 24 представлено сравнение отчетов всех решений. На рисунке 25 представлено сравнение всех решений с расчетом Latency (нс) и график зависимости производительности с аппаратной частью от порядка решения.

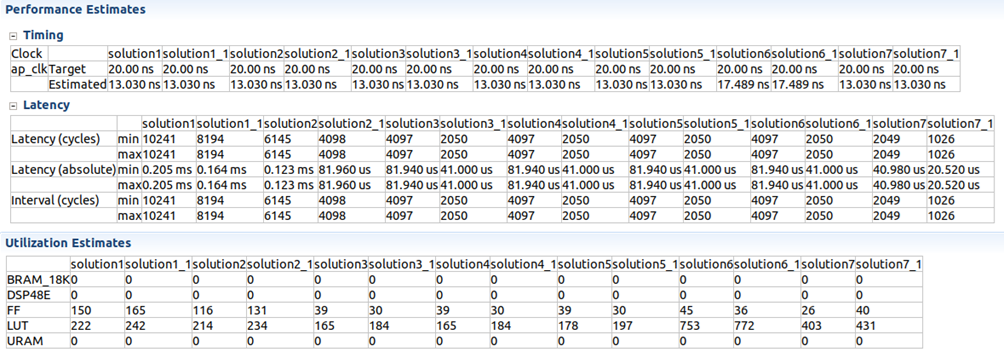


Рисунок 24 - Сравнение всех решений

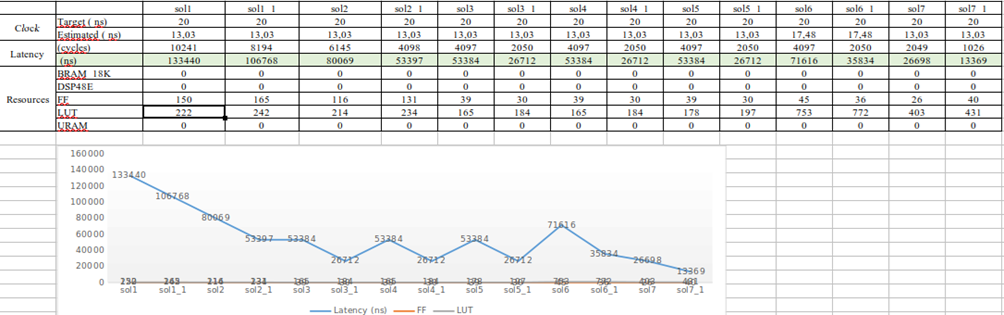


Рисунок 25 - Сравнение всех решений в виде таблицы и графика

Исходя из результатов, можно сделать вывод, что solution3\_1 и solution4\_1 имеют лучший результат по производительности и аппаратным ресурсам.

1. Результаты исследования (2)

На рисунке 26 приведено сравнение отчетов для solution1, solution7 и solution7\_1 для clock period = {4, 12, 20} и clock\_uncertainty = 1, после увеличения N = 131072.

На рисунке 27 приведено сравнение решений в виде графика и таблицы с Latency (нс). По результатам видно, что решением с лучшими временными показателями является solution7\_1\_3.

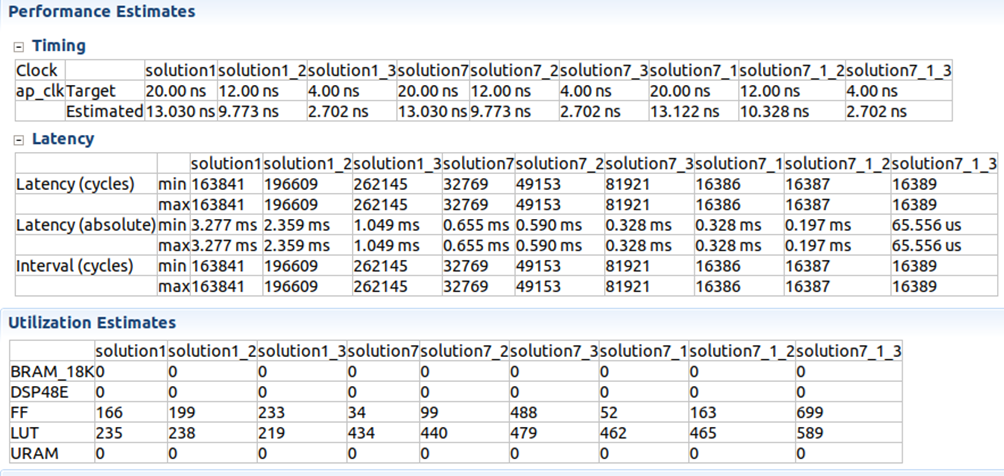


Рисунок 26 - Сравнение всех решений

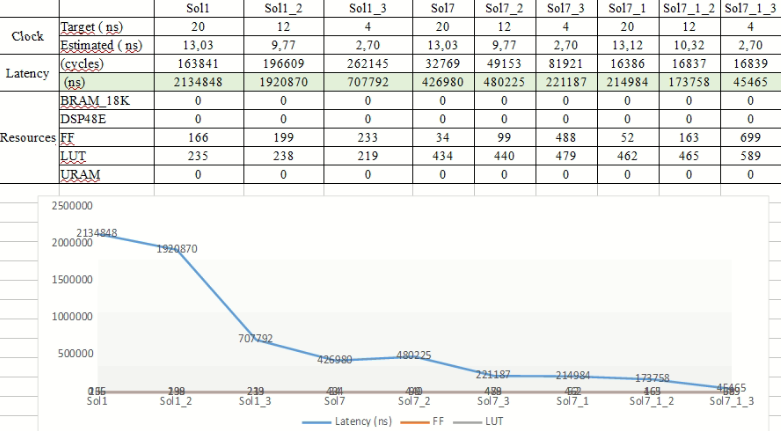


Рисунок 27 - Сравнение всех решений в виде таблицы и графика

* 1. Запуск модифицированного теста на ПК

На рисунке 28 представлен исходный код модифицированного теста для ПК. Тест обеспечивает проверку производительности функции на ПК (Компилятор gcc-9.3.0).



Рисунок 28 - Параметры ПК (Частота = 2.7 Гц)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 29 - Исходный код модифицированного теста для ПК

На рисунке 30 представлены результаты запуска функции на ПК.

Среднее время выполнения функции составило 148360 нс, что в разы медленнее, чем лучшее аппаратное решения.

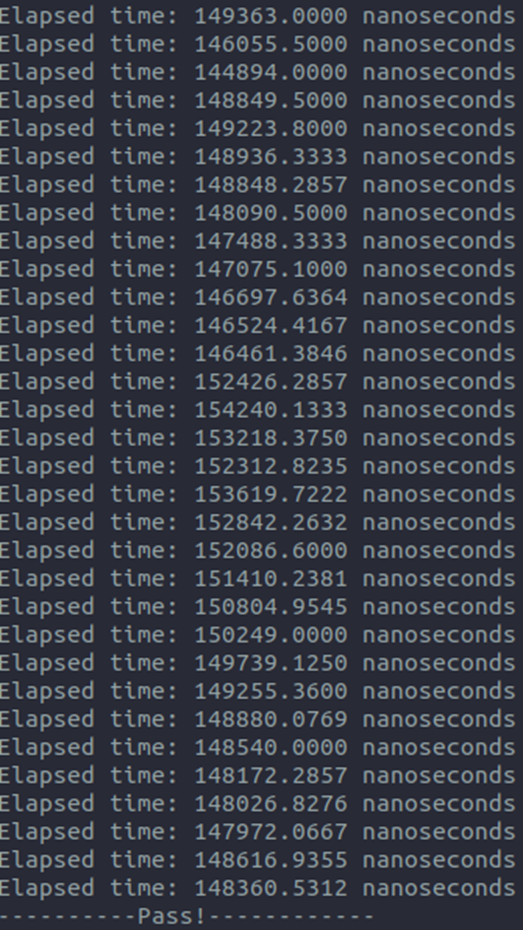


Рисунок 30 - Временные показатели для модифицированного теста для foo\_m

1. Выводы

В ходе данной лабораторной работы была изучена возможность добавления директив по оптимизации работы с массивами для синтезируемой функции. Был произведен сравнительный анализ между решением без добавления и с добавлением различных директив с одним и двумя портами для входного массива, раскруткой цикла и разбиением массива на блоки разного размера. Также было произведено сравнение временных показателей между решением полученным Vivado HLS и программным решением на ПК. Из результатов видно, что решение, полученное на ПК медленнее, чем лучшее решение, полученное синтезированием в Vivado HLS.