Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab4\_Z2

Дисциплина: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Dataflow оптимизация

Выполнил студент гр. Шеметов С.А.

гр.5140901/21502

Руководитель, доцент Антонов А.П.

«\_\_» \_\_\_\_\_\_ 2023

Санкт-Петербург

2023

Оглавление

[1. Задание 4](#_Toc148840124)

[2. Исходный код функции 4](#_Toc148840125)

[3. Исходный код теста 5](#_Toc148840126)

[3.1. Моделирование 5](#_Toc148840127)

[4. Создание скрипта 7](#_Toc148840128)

[5. Сравнение результатов решений 8](#_Toc148840129)

[5.1. Сравнение временных параметров, производительности и аппаратных затрат 8](#_Toc148840130)

[5.2. Решение sol3 10](#_Toc148840131)

[5.2.1. Performance estimates 10](#_Toc148840132)

[5.2.2. Utilization estimates 11](#_Toc148840133)

[5.2.3. Результаты планирования (Schedule viewer) 11](#_Toc148840134)

[6. Решение 4 13](#_Toc148840135)

[6.1. Синтез 13](#_Toc148840136)

[6.1.1. Performance estimates 13](#_Toc148840137)

[6.1.2. Utilization estimates 14](#_Toc148840138)

[6.1.3. Результаты планирования (Schedule viewer) 14](#_Toc148840139)

[6.1.4. Dataflow viewer 15](#_Toc148840140)

[6.2. C/RTL моделирование 16](#_Toc148840141)

[6.2.1. Отчет по производительности 16](#_Toc148840142)

[6.2.2. Временная диаграмма 17](#_Toc148840143)

[7. Решение 5 17](#_Toc148840144)

[7.1. Синтез 17](#_Toc148840145)

[7.1.1. Performance estimates 17](#_Toc148840146)

[7.1.2. Utilization estimates 18](#_Toc148840147)

[7.1.3. Результаты планирования (Schedule viewer) 18](#_Toc148840148)

[7.1.4. Dataflow viewer 19](#_Toc148840149)

[7.2. C/RTL моделирование 20](#_Toc148840150)

[7.2.1. Отчет по производительности 20](#_Toc148840151)

[7.2.2. Временная диаграмма 21](#_Toc148840152)

[8. Сравнение решений 4 и 5 21](#_Toc148840153)

[9. Измерение времени выполнения 22](#_Toc148840154)

[9.1. Измерение времени выполнения на ПК 22](#_Toc148840155)

[9.2. Измерение времени выполнения на аппаратной реализации 26](#_Toc148840156)

[9.3. Сравнение результатов 27](#_Toc148840157)

[10. Вывод 28](#_Toc148840158)

1. Задание

* Выбрать одну из двух предложенных функций, подходящих для Dataflow оптимизации, и реализовать на языке С++ (lab4\_z2.cpp).
* Создать на языке С++ тест (lab4\_z2\_test.cpp) для проверки работы функции.
* Создать скрипт автоматизирующий процесс создания проекта и работы с ним.
* Создать, провести исследование и сравнительный анализ трех решений.
  + Микросхема: xa7a12tcsg325-1q
  + clock period 6, 10, 14 (решения sol1, sol2, sol3); clock\_uncertainty 1
  + Моделирование;
  + Синтез;
  + Выбрать лучший вариант.
* Создать решения 4 и 5 и сравнить.
  + осуществить синтез с Dataflow и параметром fifo для default\_channel (для sol4);
  + осуществить синтез с Dataflow и параметром ping-pong для default\_channel (для sol5);
  + C/RTL моделирование
* Измерить время выполнения на ПК и на аппаратной реализации.

2. Исходный код функции

Исходный код синтезируемой функции приведен на рисунке 1. Код заголовочного файла указан на рисунке 2.

*Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание*

*Рис.1. Исходный код функции* *lab4\_z2*

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Рис.2. Исходный код заголовочного файла lab4\_z2*

Функция принимает входной вектор, скаляр и два выходных вектора. Все векторы имеют размер N = 16. Затем в первом цикле Loop1 входной вектор умножается на скаляр. Полученный результат записывается во временные векторы tempA1 и tempA2. Во втором и третьем циклах вектора tempA1 и tempA2 умножаются на число и записываются в выходные массивы data\_out1 и data\_out2 соответственно.

Из двух предложенных реализаций самой производительной была выбрана вторая, так как в ней происходит меньше обращений к памяти. Значение data\_in[i] получается один раз за итерацию цикла Loop1. В другой реализации data\_in[i] необходимо получить в цикле Loop1, а также дважды за итерацию в цикле L1 в функции Split.

3. Исходный код теста

Исходный код теста для проверки функции lab4\_z2, исправленный в соответствии с заданием, приведен на рисунке 3. Тест обеспечивает: запуск функции 3 раза; заполненных входного массива и скаляра случайными числами; очистку выходных массивов между запусками функции; проверку правильности вычисленного результата и формирование признака успешного/неуспешного выполнения для каждого запуска функции.

3.1. Моделирование

Результаты моделирования исходного кода синтезируемой функции приведены на рисунке 4. Результаты моделирования показывают, что тест успешно пройден — так как условия теста выполняются (функция *main* возвращает 0) и результат расчета совпадает с ожидаемыми значениями.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

*Рис.3. Исходный код теста*

*Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание*

*Рис.4. Лог результата выполнения моделирования*

Для проверки правильности работы теста в него было внесено изменение в ожидаемый результат - expted\_ar1[j] = in\_ar[j] \* scale \* 22 + 1. Результат моделирования приведён на рисунке 5. Он показывает, что тест отрабатывает ошибку корректно.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Рис.5. Лог результата при ошибке в тесте*

4. Создание скрипта

Созданный скрипт для автоматизации работы с проектом представлен на рисунке 6.

Данный скрипт создаёт или открывает проект, предварительно очищая его, если он уже существует, командой *open\_project -reset lab4\_z2*. Затем добавляется файл с исходным кодом *add\_files ./source/lab4\_z2.cpp* и устанавливается функция верхнего уровня в иерархии *set\_top lab4\_z2*. Добавляется тестовый файл - *add\_files -tb ./source/lab4\_z2\_test.cpp*. Затем по очереди создаются три решения. Для каждого из них задаётся одинаковая микросхема, периоды 6, 10, 14 (решения sol1, sol2, sol3) и неопределённость 1 нс. Для каждого решения запускается Си моделирование (*csim\_design -clean*) и выполняется синтез (*csynth\_design*).

Запуск скрипта выполняется в командной строке Vitis HLS 2021.2 Command Prompt с помощью команды *Vitis\_hls -f lab3\_z4.tcl*.

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

*Рис.6. Скрипт для работы с проектом*

5. Сравнение результатов решений

5.1. Сравнение временных параметров, производительности и аппаратных затрат

В данной работе были реализованы три решения с разными периодами. Результаты сравнения приведены на рисунке 7.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

*Рис.7. Сравнение трёх решений*

Изображение выглядит как текст, число, снимок экрана, Шрифт

Автоматически созданное описание

*Рис.8. Электронная таблица для трёх решений*

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

*Рис.9. График для трёх решений*

Сравнение данных на рисунках 7–9 показывает, что с точки зрения производительности оптимальным вариантом является третье решение, так как оно затрачивает меньше всего времени. Оно меньше на 225 нс, чем решение 1, имеющее худшую производительность из трёх.

С точки зрения аппаратных затрат лучшим решением является третье, так как имеет наименьшее число FF триггеров – 165 и наименьшее число таблиц перекодировки LUT – 252.

Таким образом, если выбирать решение по критерию максимум производительности при наименьших аппаратных затратах, то лучше выбрать решение три, так оно имеет лучшие показатели.

5.2. Решение sol3

5.2.1. Performance estimates

Временные параметры и производительность решения 3 представлены на рисунке 10.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

*Рис.10. Performance estimates для sol3*

5.2.2. Utilization estimates

Аппаратные затраты для решения 3 представлены на рисунке 11.

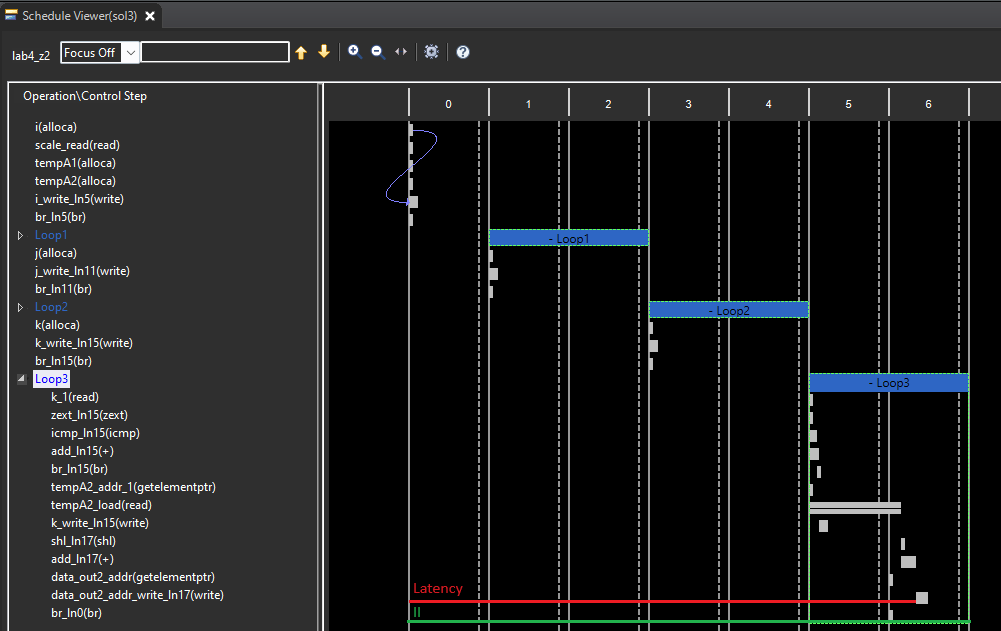
Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

*Рис.11. Utilization estimates для sol3*

5.2.3. Результаты планирования (Schedule viewer)

Результаты планирования для третьего решения представлены на рисунках 12–15.



*Рис.12. Результаты планирования для sol3*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

*Рис.13. Результаты планирования для sol3. Цикл Loop1*

Изображение выглядит как снимок экрана, Графическое программное обеспечение, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

*Рис.14. Результаты планирования для sol3. Цикл Loop2*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

*Рис.15. Результаты планирования для sol3. Цикл Loop3*

По результатам третьего решения видно, что Latency и II совпадают со значениями, полученными на рисунке 7. Первый цикл начинается со второго такта, между каждым циклом есть один промежуточный такт, а каждый цикл длится 2 такта. Таким образом Latency = 2 такта \* 16 \* 3 цикла + 1 такт (нулевой) + 2 такта (между циклами) = 99 тактов. II = Latency + 1 = 100 тактов.

6. Решение 4

6.1. Синтез

На основе решения 3 было добавлено решение 4, в котором осуществляется синтез с Dataflow и параметром FIFO для default\_channel. Решение представлено на рисунке 16.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Рис.16. Создание sol4 в скрипте проекта*

6.1.1. Performance estimates

Временные параметры и производительность решения 4 представлены на рисунке 17. Как можно видеть, Estimated timing уменьшилось (10.622 нс для sol4 и 12.774 нс для sol3), значения Latency и II также уменьшились (51 такт и 50 тактов для sol4 / 99 тактов и 100 тактов для sol3). Также в последней графе изменился Type на dataflow.

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание

*Рис.17. Performance estimates для sol4*

6.1.2. Utilization estimates

Аппаратные затраты для решения 4 представлены на рисунке 18. Как видно, аппаратные затраты увеличились почти вдвое (295 FF и 456 LUT для sol4 / 165 FF и 252 LUT для sol3).

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

*Рис.18. Utilization estimates для sol4*

6.1.3. Результаты планирования (Schedule viewer)

Результаты планирования для четвёртого решения представлены на рисунках 19–22.

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

*Рис.19. Результаты планирования для sol4*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

*Рис.20. Результаты планирования для sol4. Цикл Loop1*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

*Рис.21. Результаты планирования для sol4. Цикл Loop2*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

*Рис.22. Результаты планирования для sol4. Цикл Loop3*

По результатам четвёртого решения видно, что циклы были оптимизированы и на рисунке 19 представлены как функции.

6.1.4. Dataflow viewer

Результаты Dataflow viewer для решения 4 представлены на рисунке 23.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Рис.23. Dataflow viewer для sol4*

Как видно на рисунке входной вектор data\_in и скаляр scale сначала попадают в первый цикл Loop 1, затем данные направляются в остальные два цикла, которые формируют два выходных вектора data\_out.

6.2. C/RTL моделирование

6.2.1. Отчет по производительности

Отчеты по производительности для решения 4, полученный при моделировании cosim, приведен на рисунке 24. Из отчётов следует, что Latency и II совпадают со значениями, полученными при синтезе, показанными на рисунке 17.

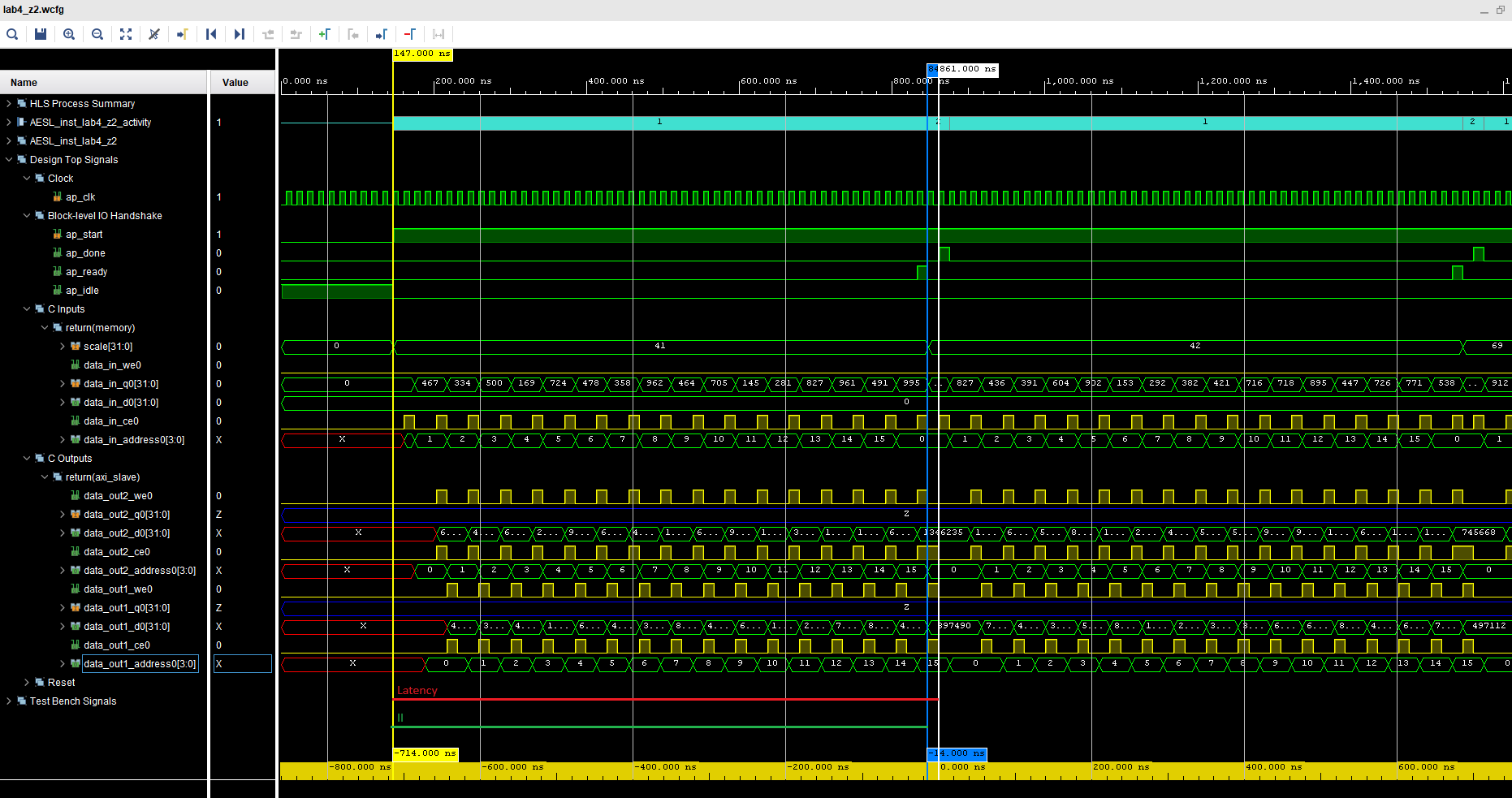
Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.24. Отчет по производительности C/RTL для sol4*

6.2.2. Временная диаграмма

На рисунке 25 приведена временная диаграмма совместного C/RTL моделирования для четвертого решения. Она отображает два запуска функции. Как можно видеть Latency занимает 51 такт, а II – 50 тактов, что соответствует рисунку 24.



*Рис.25. Временная диаграмма C/RTL для sol4*

7. Решение 5

7.1. Синтез

На основе решения 3 было добавлено решение 5, в котором осуществляется синтез с Dataflow и параметром ping-pong для default\_channel. Решение представлено на рисунке 26.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Рис.26. Создание sol5 в скрипте проекта*

7.1.1. Performance estimates

Временные параметры и производительность решения 5 представлены на рисунке 27. Как можно видеть, Estimated timing совпадает с sol3, а значения Latency и II уменьшились (67 тактов и 34 такта для sol5 / 99 тактов и 100 тактов для sol3). Также в последней графе изменился Type на dataflow.

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание

*Рис.27. Performance estimates для sol5*

7.1.2. Utilization estimates

Аппаратные затраты для решения 5 представлены на рисунке 28. Как видно, число триггеров значительно уменьшилось, а число таблиц перекодировки немного увеличилось (44 FF и 293 LUT для sol4 / 165 FF и 252 LUT для sol3).

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

*Рис.28. Utilization estimates для sol5*

7.1.3. Результаты планирования (Schedule viewer)

Результаты планирования для решения 5 представлены на рисунках 29–32.

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

*Рис.29. Результаты планирования для sol5*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

*Рис.30. Результаты планирования для sol5. Цикл Loop1*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

*Рис.31. Результаты планирования для sol5. Цикл Loop2*

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

*Рис.32. Результаты планирования для sol5. Цикл Loop3*

По результатам пятого решения видно, что циклы были оптимизированы и на рисунке 29 представлены как функции.

7.1.4. Dataflow viewer

Результаты Dataflow viewer для решения 5 представлены на рисунке 33.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Рис.33. Dataflow viewer для sol5*

Как видно на рисунке входной вектор data\_in и скаляр scale сначала попадают в первый цикл Loop 1, затем данные направляются в остальные два цикла, которые формируют два выходных вектора data\_out.

7.2. C/RTL моделирование

7.2.1. Отчет по производительности

Отчеты по производительности для решения 5, полученный при моделировании cosim, приведен на рисунке 34. Из отчётов следует, что Latency и II совпадают со значениями, полученными при синтезе, показанными на рисунке 27.

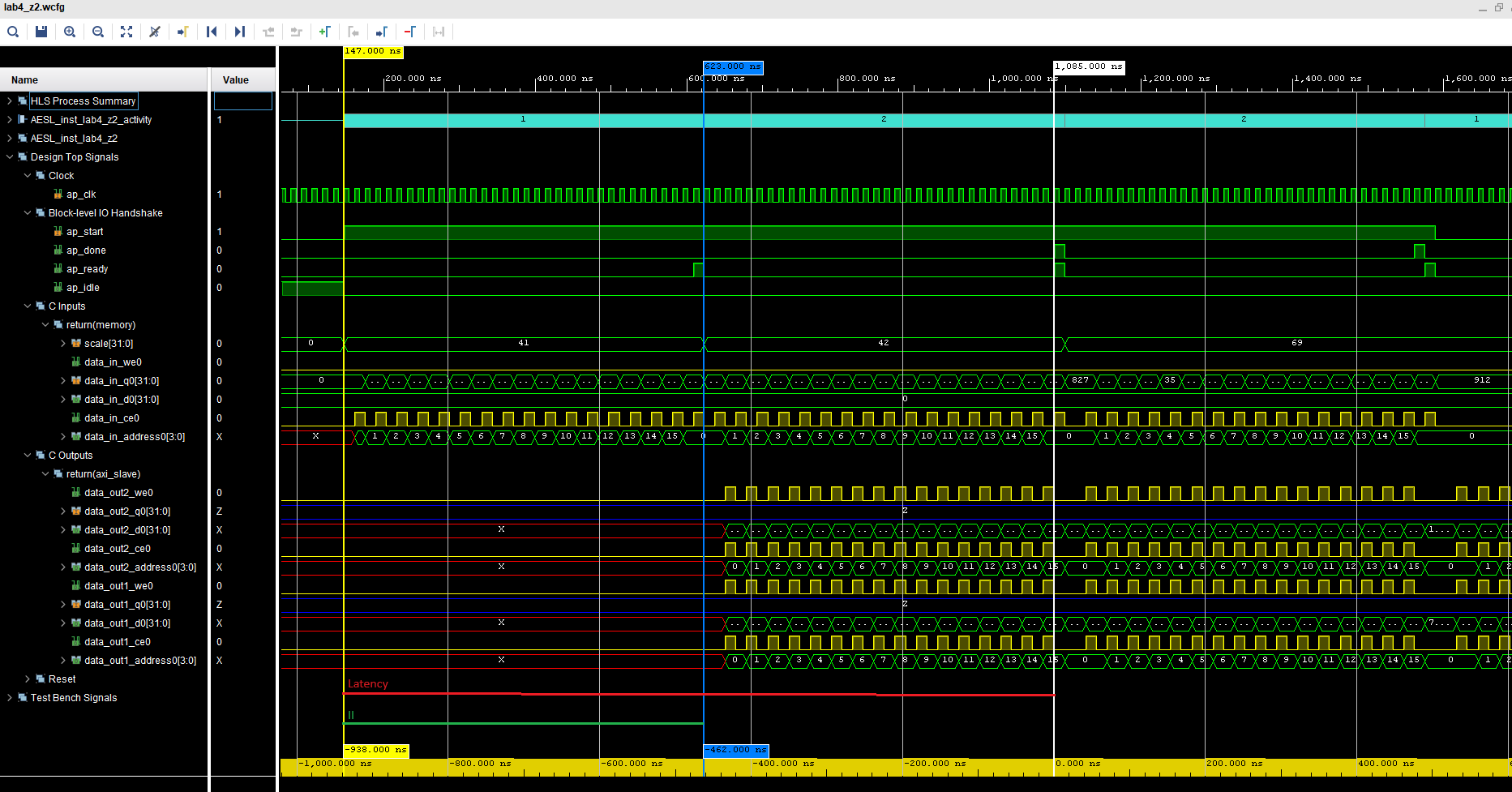
Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.34. Отчет по производительности C/RTL для sol5*

7.2.2. Временная диаграмма

На рисунке 35 приведена временная диаграмма совместного C/RTL моделирования для пятого решения. Она отображает два запуска функции. Как можно видеть Latency занимает 67 тактов, а II – 34 такта, что соответствует рисунку 34.



*Рис.35. Временная диаграмма C/RTL для sol4*

8. Сравнение решений 4 и 5

Были реализованы два решения с разными параметрами Dataflow. Результаты сравнения приведены на рисунке 36.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

*Рис.36. Сравнение двух решений*

С точки зрения производительности оптимальным решением является четвёртое, так как оно затрачивает меньше времени (542 нс для sol4 и 856 нс для sol5). Это обусловлено использованием, что параметр fifo более эффективен при работе с данными, которые читаются и записываются в последовательном порядке. Однако данное решение требует больших аппаратных затрат.

С точки зрения аппаратных затрат пятое решение является оптимальным, так как оно имеет меньшее число триггеров и таблиц перекодировок.

9. Измерение времени выполнения

9.1. Измерение времени выполнения на ПК

Измерение времени выполнения синтезируемой функции было выполнено на ПК со следующими характеристиками:

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип процессора** | **Базовая частота работы** | **Максимальная частота работы** | **ОЗУ** |
| 12th Gen Intel(R) Core(TM) i5-12450H | 2 ГГц | 4,4 ГГц | 16 Гб |

На базе теста lab4\_z2\_test.cpp был создан отдельный, модернизированный, тест lab4\_z2\_testSW.cpp, который содержит операторы измерения времени выполнения и имеет большее число запусков равное 32. Код теста представлен на рисунке 37.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

*Рис.37. Код теста* *lab4\_z2\_testSW.cpp*

Для компиляции теста был установлен компилятор GCC версии 13.2.0. После этого была выполнена сама компиляции и запуск теста для трёх значений N – 4096. 8192, 16384 (рисунок 38). Результаты запуска представлены на рисунках 39–41.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

*Рис.38. Выполнение компиляции теста*

Изображение выглядит как текст, снимок экрана, черно-белый, монохромный

Автоматически созданное описание

*Рис.39. Результаты теста на ПК для N=4096*

Изображение выглядит как текст, снимок экрана, черно-белый, монохромный

Автоматически созданное описание

*Рис.40. Результаты теста на ПК для M=N=8192*

Изображение выглядит как текст, снимок экрана, меню, черно-белый

Автоматически созданное описание

*Рис.41. Результаты теста на ПК для M=N=16384*

9.2. Измерение времени выполнения на аппаратной реализации

Для измерения времени выполнения на аппаратной реализации был осуществлён синтез решения sol4 для трёх значений N – 4096. 8192, 16384. Результаты запуска представлены на рисунках 42–44.

Время выполнения = II \* Estimated period для каждого случая:

* для 4096: 32772 такта \* 11.727 нс = 384317.244 нс
* для 8192: 65540 такта \* 11.727 нс = 768587.58 нс
* для 16384: 131076 такта \* 11.727 нс = 1537128.25 нс

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.42. Результаты теста на аппаратной реализации для N=4096*

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.43. Результаты теста на аппаратной реализации для N=8192*

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.44. Результаты теста на аппаратной реализации для N=16384*

9.3. Сравнение результатов

На основе результатов, полученных при измерениях времени выполнения на ПК и аппаратной реализации, была составлена электронная таблица и построен график, в котором отражено соотношение размера массивов данных задачи (N) ко времени выполнения. Результаты представлены на рисунках 45–46.

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

*Рис.45. Электронная таблица для ПК и аппаратной реализации*

Изображение выглядит как снимок экрана, текст, линия, График

Автоматически созданное описание

*Рис.46. График для ПК и аппаратной реализации*

На графике можно видеть, что результаты аппаратной реализации растут экспоненциально, а на ПК более линейно. При этом на аппаратной реализации время выполнения в 10–20 раз больше. Более медленное выполнения аппаратной реализации связано с тем, что она имеет меньшую частоту по сравнению с ПК.

10. Вывод

В результате выполнения данной лабораторной работы были получены навыки работы с Dataflow оптимизацией. Для этого была выбрана наиболее производительная из двух предложенных реализаций программы, удовлетворяющая модели single-producer-consumer. Такой реализацие оказалась вторая, так как она имеет меньшее число обращений к data\_in[i] в циклах.

Для создания проекта и работы с ним был создан TCL скрипт, содержащий три решения с различным периодом. Из трёх решений было выбрано оптимальное по критерию максимум производительности при наименьших аппаратных затратах. Им оказалось решение 3. На основе этого решения были созданы sol4 и sol5, осуществляющие синтез с Dataflow.

Для обоих решений были заданы разные параметры fifo для default\_channel для решений 4 и ping-pong для default\_channel для решений 5. Они задают как будет выполняться конвейерная обработка потока данных. С точки зрения производительности лучшим оказалось 4 решение из-за наличия fifo.

Также было выполнено измерение времени выполнения программы на ПК и аппаратной реализации. Результаты показали, что аппаратная реализация значительно проигрывает в производительности.