САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа lab3\_z2

Дисциплина:

«Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: Введение в Port-Level I/O Protocols

Выполнил:

Бараев Д. Р.

Группа: 3540901/02001

Преподаватель: А. П. Антонов

Санкт-Петербург

2021

**Оглавление**

[1. Задание 4](#_Toc85233553)

[2. Исходный код функции 5](#_Toc85233554)

[3. Исходный код теста 6](#_Toc85233555)

[4. Исходный код командного файла 7](#_Toc85233556)

[5. Результаты исследования и сравнение решений 9](#_Toc85233557)

[6. Анализ результатов 10](#_Toc85233558)

[7. Выводы 11](#_Toc85233559)

**Список иллюстраций**

[Рисунок 1 - Исходный код функции (файл lab3\_z2.h) 5](#_Toc85233533)

[Рисунок 2 - Исходный код функции с обычным считыванием (lab3\_z2s.c) 5](#_Toc85233534)

[Рисунок 3 - Исходный код синтезируемой функции с memcpy считыванием (lab3\_z2b.c) 6](#_Toc85233535)

[Рисунок 4 - Исходный код теста 7](#_Toc85233536)

[Рисунок 5 Исходный код командного файла для создания проекта (la3\_z2.tcl) 8](#_Toc85233537)

[Рисунок 6 - Сравнение полученных решений 9](#_Toc85233538)

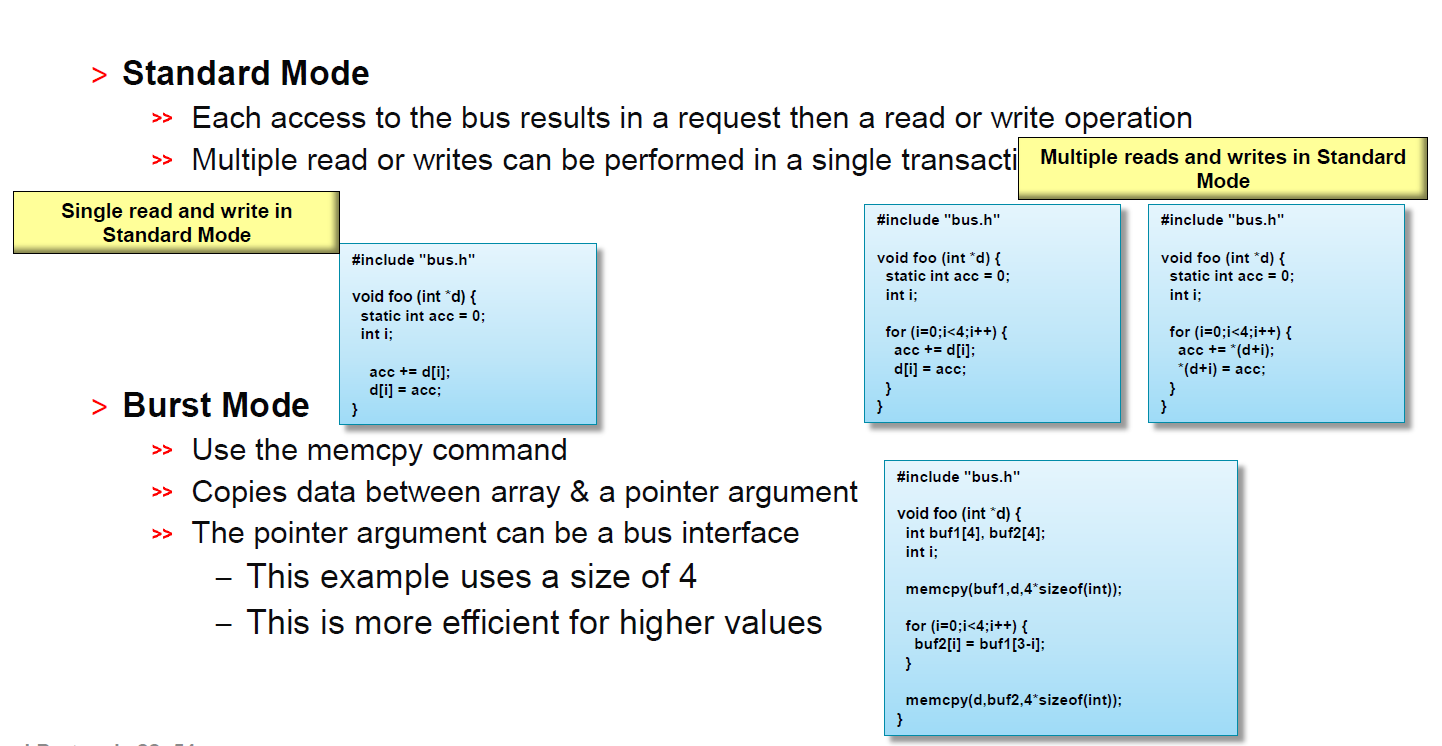
[Рисунок 7 - Таблица данных для всех решений 10](#_Toc85233539)

[Рисунок 8 График зависимости данных для всех решений 11](#_Toc85233540)

1. Задание

* Создать на языке Си функцию,
  + Считывающую элементы массива D\_I, элементы массива имеют тип int (32 бита), размер массива 256 элемента, в функцию передается массив.
  + Считывающую элементы массива С\_I, элементы массива имеют тип int (32 бита), размер массива 16 элементов, в функцию передается указатель.
  + произведение i го элемента массива D\_I и j-го (от 0 до 15) элемента массива С\_I записывается в i элемент массива D\_O
  + Сделать ДВЕ реализации функции
    - Первая – обычное считывание (lab3\_z2s.c)
    - Вторая – считывание С\_I в буфер с помощью memcpy (lab3\_z2b.c)

По примеру (слайд 53 в лекции)



* Создать на языке Си тест для проверки работы функции. Тест должен обеспечивать
  + запуск функции минимум 2 раза,
    - На вход функции поступает массив (созданный с использованием генератора случайных чисел), элементы массива имеют тип int (32 бита).
  + проверку правильности вычисленного результата и формирование признака успешного/неуспешного выполнения для каждого запуска функции,
* Отладить функции и тест (при неправильном результате в любом из запусков функции должен сообщать об ошибке).
* Создать скрипт автоматизирующий процесс:
  + Создания проекта lab3\_z2,
  + Подключения файла lab3\_z2\_test.c (папка source),
    - задается микросхема: xa7a12tcsg325-1q,
    - задается clock period 6; clock\_uncertainty 0.1,
  + Создания решении (
    - sol1, для которого Block Level I/O interface по умолчанию и используется функция первого варианта
      * Подключения файла lab3\_z2s.c (папка source),
    - sol2, для которого Block Level I/O interface по умолчанию и используется функция второго варианта
      * Подключения файла lab3\_z2b.c (папка source),
* Отладить и проверить работу созданного скрипта.
* После выполнения скрипта открыть GUI
* Убедиться, что созданы все решения
* Используя средства HLS сравнить полученные решения
  + аппаратные затраты
  + результаты планирования
  + использованные интерфейсы
  + привести временные диаграммы и объяснить в чем их отличия.

1. Исходный код функции

Исходный код объявления синтезируемой функции и макроопределений приведен на Рисунке 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 - Исходный код функции (файл lab3\_z2.h)

Исходный код синтезируемой функции с обычным считыванием приведен на рисунке 2.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 - Исходный код функции с обычным считыванием (lab3\_z2s.c)

Исходный код синтезируемой функции с memcpy считыванием приведен на рисунке 3.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 - Исходный код синтезируемой функции с memcpy считыванием (lab3\_z2b.c)

1. Исходный код теста

Исходный код теста для проверки функции lab3\_z2 приведен на рисунке 4. Тест обеспечивает проверку корректной работы функции.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 - Исходный код теста

1. Исходный код командного файла

На рисунке 5 приведен текст команд для автоматизированного создания следующих вариантов аппаратной реализаций:

1. Для sol1 задается clock period 6: clock uncertainty 0.1;

подключается файл lab3\_z2s.c

1. Для sol2 задается clock period 6: clock uncertainty 0.1;

подключается файл lab3\_z2b.c

Также устанавливается директива интерфейса для указателя C\_I — ap\_bus с параметром depth = 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 Исходный код командного файла для создания проекта (la3\_z2.tcl)

1. Результаты исследования и сравнение решений

На рисунке 6 представлено сравнение из Vivado HLS GUI по аппаратным ресурсам, требуемых для реализации синтезируемой функции, и временным параметрам.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6 - Сравнение полученных решений

Target – планируемое время на один такт.

Estimated – оценочное время.

Latency (cycle) – количество тактов latency за один цикл.

Latency (absolute) – время затраченное на latency.

По результатам видно, что Estimated Time = 5,9 нс для двух решений, Latency(cycle) в количестве 17153 итерации для sol1 и 21269 итерации для sol2.

По аппаратным ресурсам видно, что и решение sol1 и sol2 требуют 3 DSP48E модуля. Также для решения sol1 требуется 274 триггера FF и 174 LUT, а для sol2 394 триггера FF и 277 LUT.

После добавления третьего решения sol3 синтез прерывается из-за ошибки. Ошибка заключается в невозможности использования интерфейса ap\_fifo для выходного массива. Для выходного массива последовательность неизвестная, поэтому, при замене памяти на FIFO, синтез будет выдавать ошибки.

1. Анализ результатов

На Рисунке 7 представлена таблица с параметрами для всех решений.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 7 - Таблица данных для всех решений

На Рисунке 8 представлен график данных для сравнения всех решений.

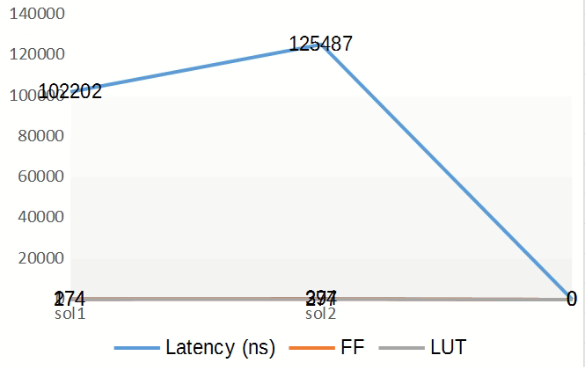


Рисунок 8 График зависимости данных для всех решений

Исходя из результатов, видно, что второе решение затрачивает большее время и

большее количество почти всех аппаратных ресурсов.

Сделаем вывод, что лучшим является первое решение, так как мы получаем выигрыш по двум параметрам.

1. Выводы

В данной работе была изучена возможность задания протоколов для I/O port-level у параметров функции. Был произведен сравнительный анализ между решением с обычным копированием данных и копированием с использованием функции memcpy из стандартной библиотеки. Как видно по результатам решением с memcpy является менее предпочтительным, так как проигрывает по все показателям по сравнению с решением без него.