Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчет по лабораторной работе №6\_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](#_Toc27673155)

[2. Первое решение 5](#_Toc27673156)

[2.1. Исходный код программы и теста 5](#_Toc27673157)

[2.2. Моделирование 6](#_Toc27673158)

[2.3. Синтез 6](#_Toc27673159)

[2.4. C|RTL моделирование 8](#_Toc27673160)

[3. Второе решение 10](#_Toc27673161)

[3.1. Настройки второго решения 10](#_Toc27673162)

[3.2. Моделирование 10](#_Toc27673163)

[3.3. Синтез 11](#_Toc27673164)

[3.4. C|RTL моделирование 13](#_Toc27673165)

[4.Выводы 14](#_Toc27673166)

# Задание

* Создать проект lab6\_1
* Микросхема: xa7a12tcsg325-1q
* Создать Си код на основе слайда (функция foo)
* Создать тест lab6\_1\_test.c на основе слайда выше.
* Сделать solution1
  + задать: clock period 6; clock\_uncertainty 0.1
  + осуществить моделирование (на основе слайда выше, с выводом результатов в консоль)
  + осуществить синтез (с настройками по умолчанию – интерфейс ap-fifo)
    - 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
    - Привести результаты из консоли
    - 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
    - a: ap\_bus
  + 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
    - Привести результаты из консоли
    - 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

# 2. Первое решение

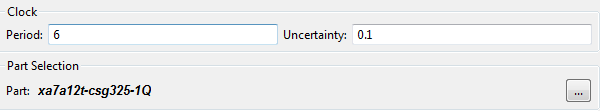


Рисунок 1.1. Параметры первого решения

## 2.1. Исходный код программы и теста

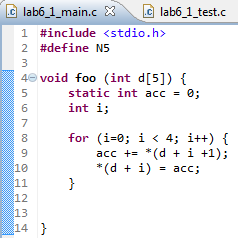


Рисунок 1.2. Исходный код синтезируемой функции

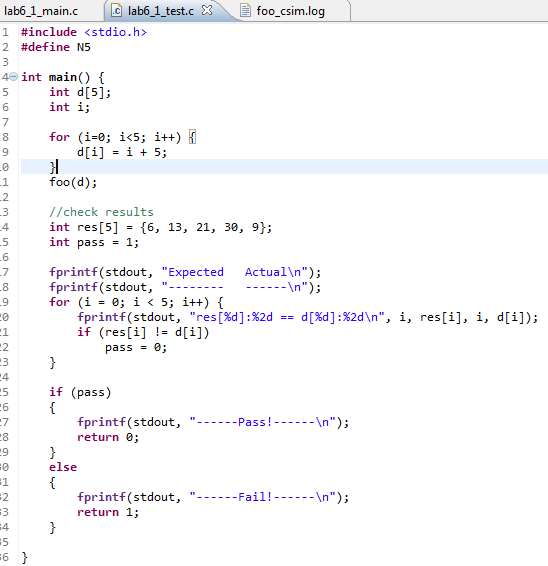


Рисунок 1.3. Исходный код теста

## 2.2. Моделирование

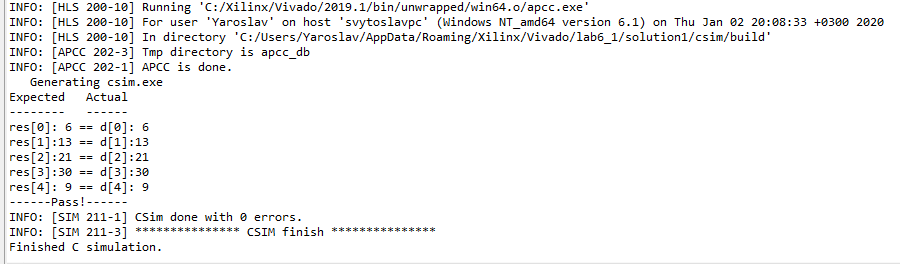


Рисунок 1.4. Результат успешного моделирования

Моделирование прошло успешно.

## 2.3. Синтез

Данные о проекте

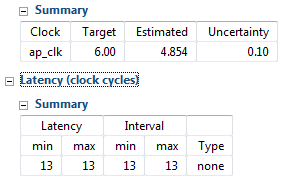


Рисунок 1.5. Performance estimates – summary

Здесь можно увидеть, что достигнутая задержка равна 4.854 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

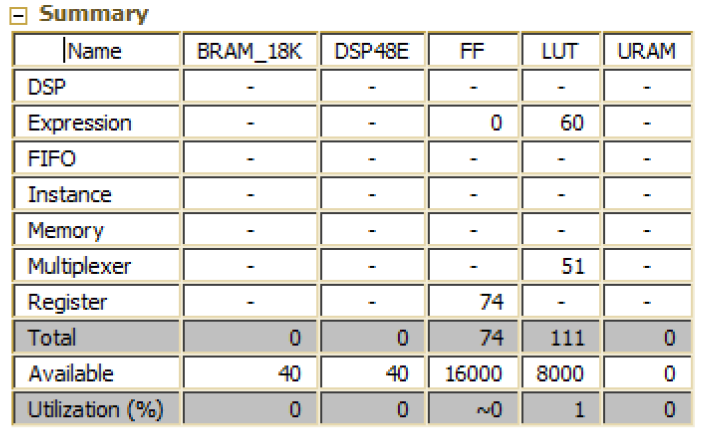


Рисунок 1.6. Utilization estimates – summary

Данный проект займет на микросхеме 74 регистра для хранения чисел и 111 LUT.

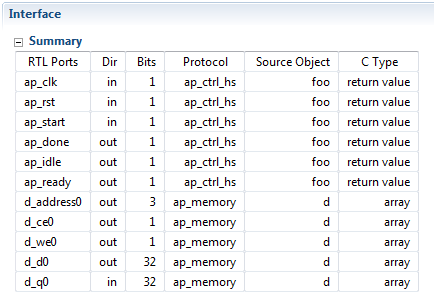


Рисунок 1.7. Interface estimates – summary

Для расчета схемы требуется более одного такта. На рисунке 1.7 представлены интерфейсы, которые используются в синтезированном устройстве. Видно, что в схеме применяется протокол ap\_memory. Порты d\_d0 и d\_q0 32-битные.

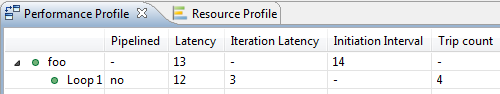


Рисунок 1.8. Performance Profile

На рисунке 1.8 видно, что задержка получения первого выходного значения составляет 3 такта с момента старта (всех данных – 13), а задержка после старта до готовности приема новых данных – 14.

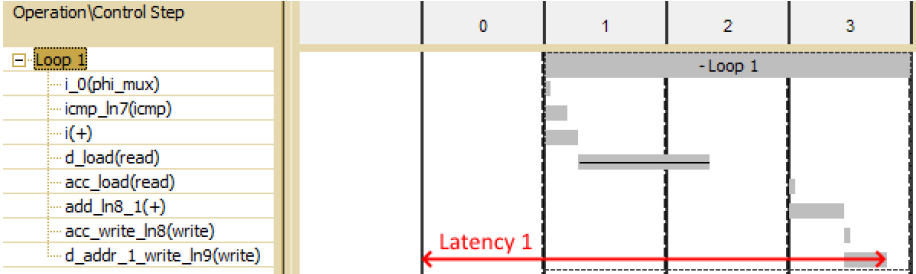


Рисунок 1.9. Schedule viewer

На рисунке 1.9 в первом такте происходит подготовка к запуску цикла, вероятно. Далее на первом такте каждой итерации цикла инициализация переменной-счетчика, проверка условия завершения, увеличение счѐтчика и начало считывания данных из массива d. На втором – завершение считывания данных из массива d. На третьем – чтение сохраненного значения аккумулятора, прибавление к нему значение элемента, прочитанного из d, запись результата обратно в аккумулятор и в массив d.

Рассмотрим профиль ресурсов:

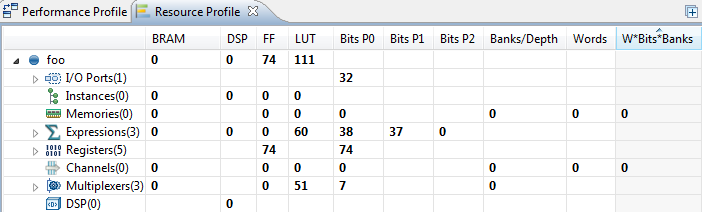


Рисунок 1.10. Resource Profile.

Значения в Resource Profile на рис. 1.10 совпадает с результатами синтеза на рис. 1.6.

## 2.4. C|RTL моделирование

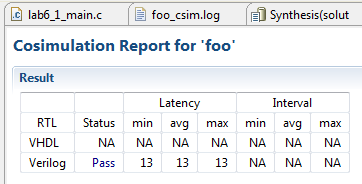


Рисунок 1.11. Cosimulation Report

При совместном моделировании, программа на рисунке 1.11 отобразила те же самые, ожидаемые нами значения Latency и II.

Покажем на рисунке 1.12 временную диаграмму совместного моделирования с отмеченными на ней Latency и II:

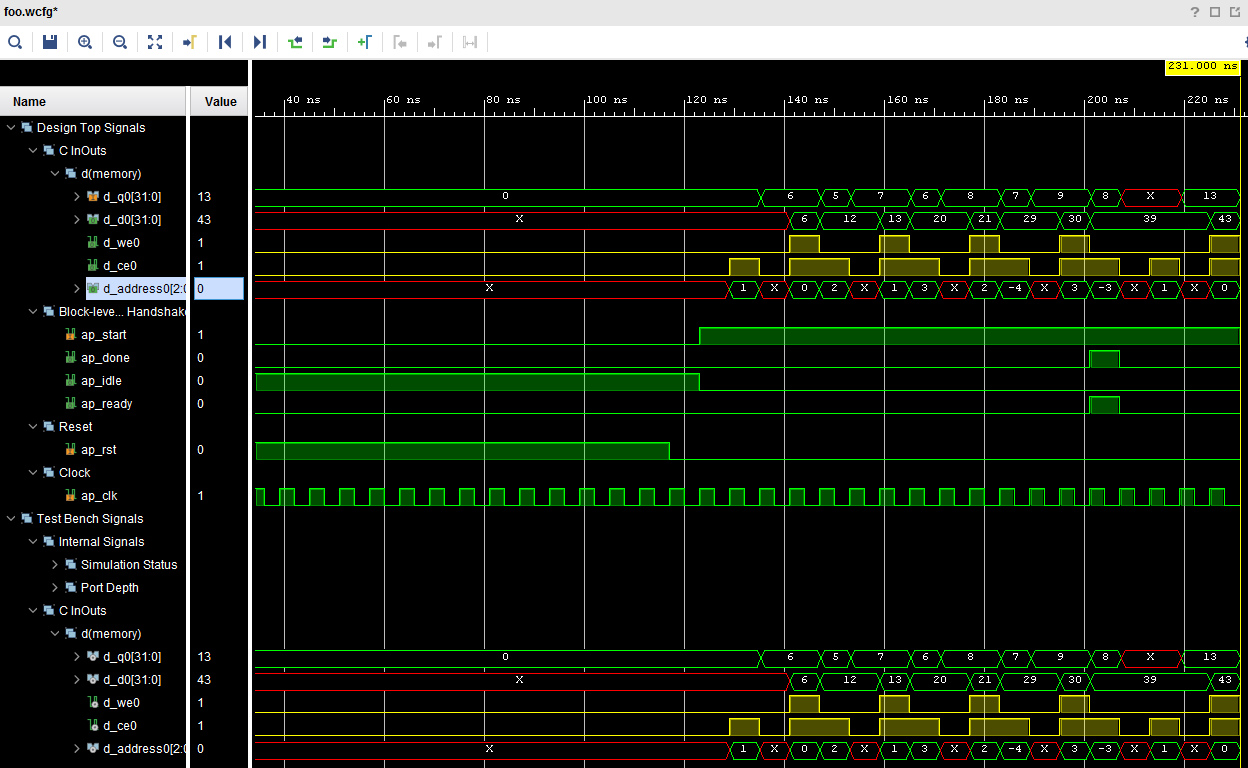


Рисунок 1.12. Design Top Signals

# Второе решение

## 3.1. Настройки второго решения

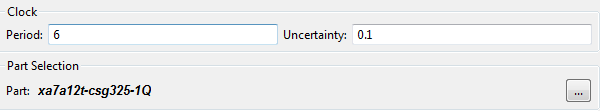


Рисунок 2.1. Параметры второго решения

Добавим директиву, которая изменяет используемый port-level протокол

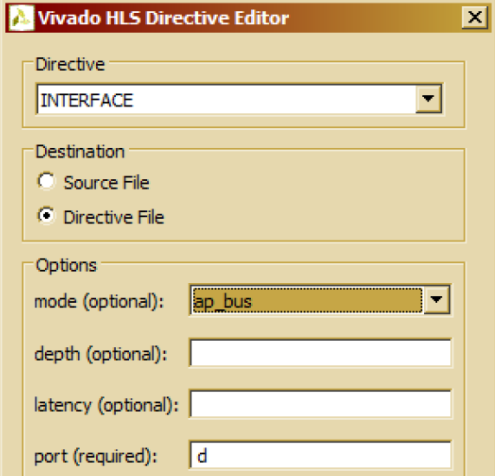


Рисунок 2.2. Directive Interface

## 3.2. Моделирование

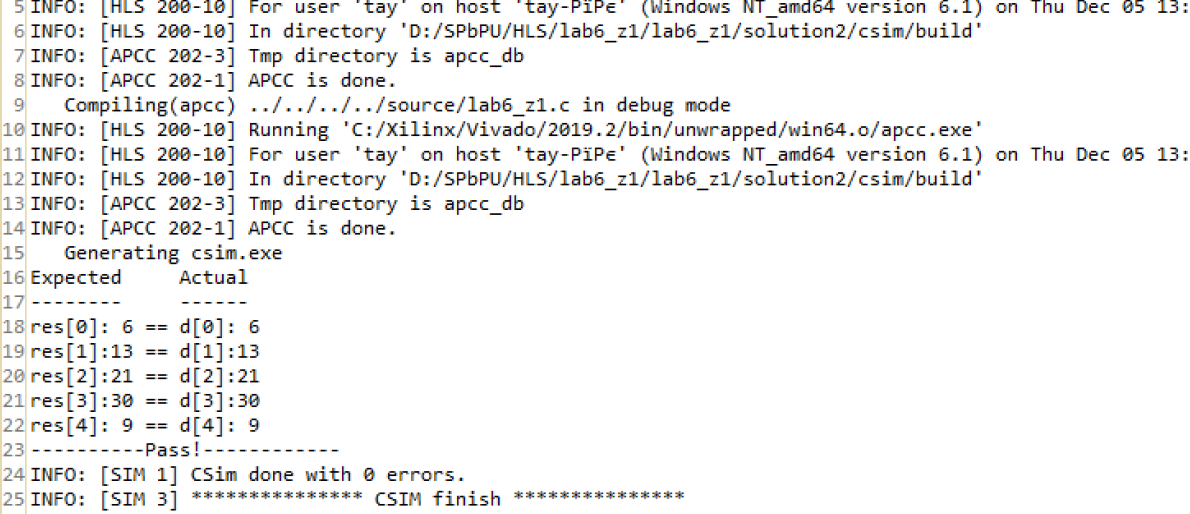


Рисунок 2.3 Результат успешного моделирования

Моделирование второго решения также прошло успешно.

## 3.3. Синтез

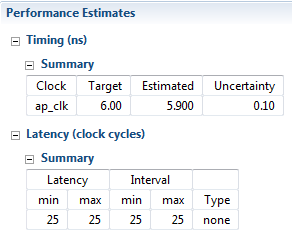


Рисунок 2.4. Performance estimates – summary

На рисунке 2.4. можно увидеть, что достигнутая задержка равна 5.900 + 0.1, укладывается в заданные требования тактовой частоты.

Использование ресурсов:

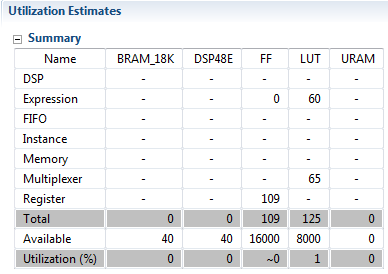


Рисунок 2.5. Utilization estimates – summary

На рисунке 2.5. видно, что данный проект теперь займет на микросхеме 109 регистров для хранения чисел, и 125 LUT.

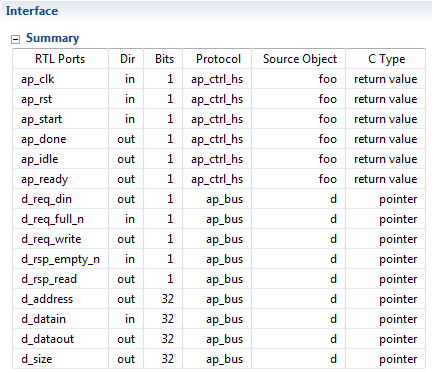


Рисунок 2.6. Interface Summary.

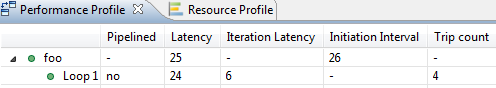


Рисунок 2.7. Performance Profile

На рисунке 2.7. видно, что задержка получения первого выходного значения составляет 7 тактов с момента старта, для остальных -25, а задержка после старта до готовности приема новых данных – 26:

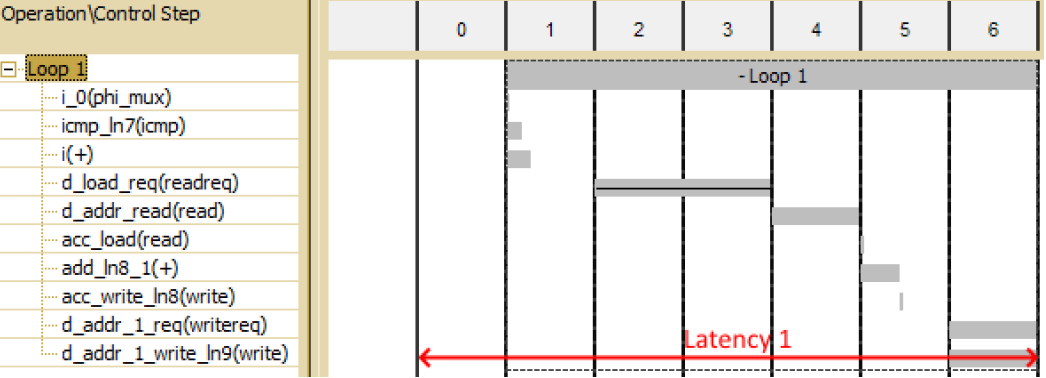


Рисунок 2.8. Schedule viewer

Последовательность работы близка к решению 1, однако, операции чтения и записи в массив d теперь выполняются отдельно от всех остальных операций, и на это тратится больше тактов.

Рассмотрим профиль ресурсов:

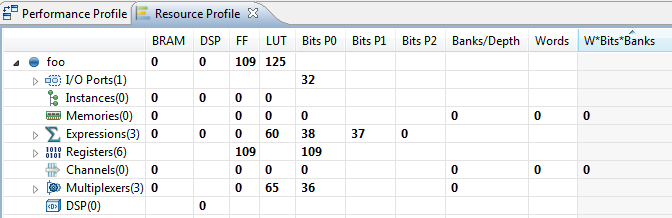


Рисунок 2.9. Resource Profile

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

## 3.4. C|RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency

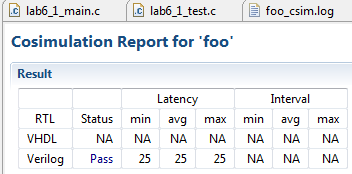


Рисунок 2.10. Отчет о моделировании

Покажем временную диаграмму совместного моделирования

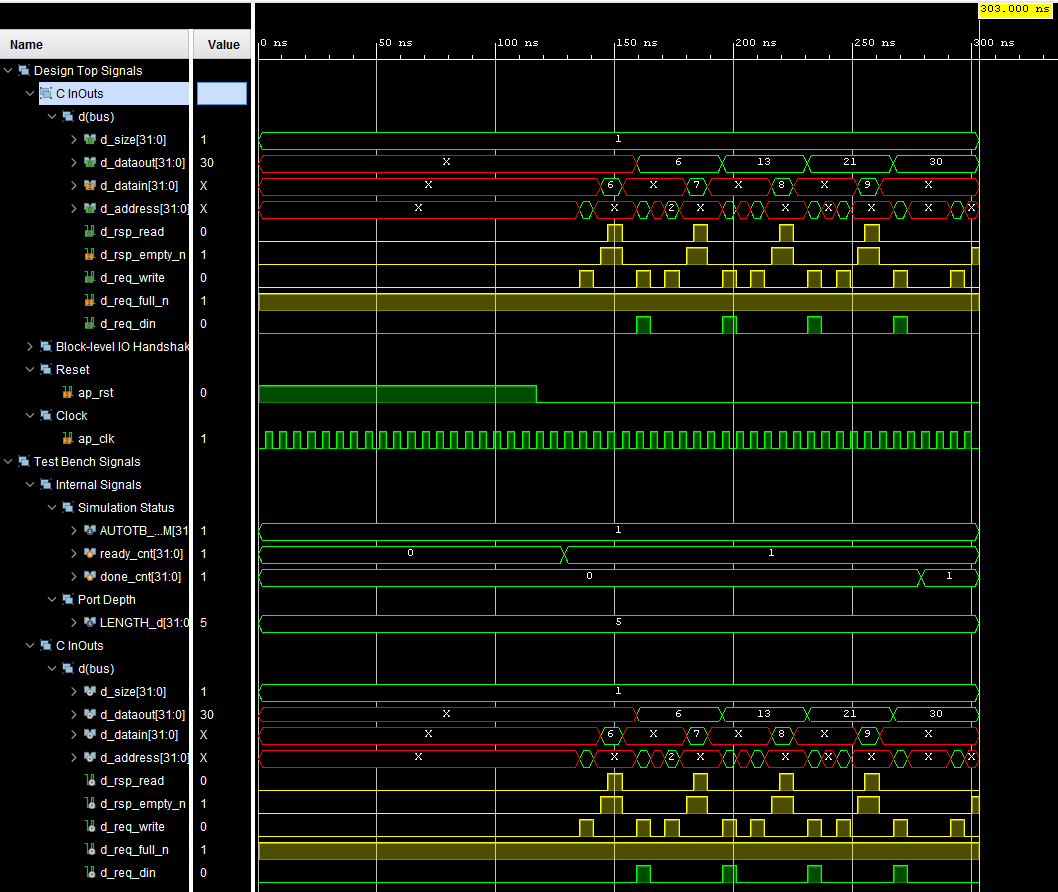


Рисунок 2.11. Design Top Signals

Здесь также видны отличия во времени выполнения итераций и протоколе работы с массивом D.

# 4.Выводы

В данной работе были рассмотрены различия, которые могут появиться при синтезе устройства с применением различных block-level протоколов (ap\_memory и ap\_bus). По-умолчанию для массивов используется протокол типа тип ap\_memory. Протокол типа ap\_bus реализует переменные указателя и передачи по ссылке в виде шины общего назначения.

В результате получены 2 решения: первое – полный цикл выполнения 6 тактов, а максимальная задержка обработки сигнала на такте составляет 4.854 нс, и второе – полный цикл выполнения тоже 6 тактов, но задержка уже 5.900 нс и используется протокол ap\_bus.