Санкт-Петербургский Политехнический Университет Петра Великого  
Институт Компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

Лабораторная работа 6

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Port-Level IO Protocols

Задание 1

Студент: Ерниязов Т.Е.

Гр. № 3540901/81501

Преподаватель: Антонов А.П.

Санкт-Петербург

2019

Оглавление

[Задание 3](#_Toc22318352)

[Ход работы 5](#_Toc22318353)

[Решение 1 5](#_Toc22318354)

[Решение 2 9](#_Toc22318355)

[Выводы 14](#_Toc22318356)

# 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

# Ход работы

## Решение 1

Исходные файлы

#define DCT\_SIZE 5

void foo(int d[DCT\_SIZE]){

static int acc = 0;

int i;

for (i = 0; i < 4; i++) {

acc += d[i+1];

d[i] = acc;

}

}

Тест

#include <stdio.h>

int main()

{

int d[5];

int i;

for (i = 0; i < 5; i++) {

d[i] = i + 5;

}

foo(d);

int res[5] = {6, 13, 21, 30, 9};

int pass;

fprintf(stdout, "Expected Actual\n");

for (i = 0; i < 5; i++) {

fprintf(stdout, "res[%d]: %d == d[%d]: %d\n", i, res[i], i, d[i]);

if (res[i] == d[i]) {

pass = 1;

} else {

pass = 0;

break;

}

}

if (pass) {

fprintf(stdout, "----------Pass!------------\n");

return 0;

} else {

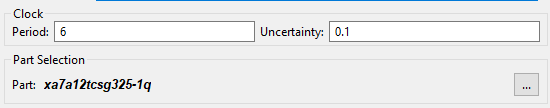
fprintf(stderr, "----------Fail!------------\n");

return 1;

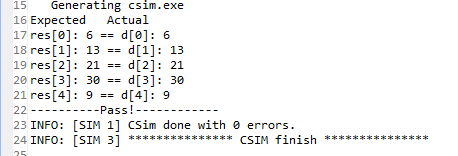
}

}

Настройка решения

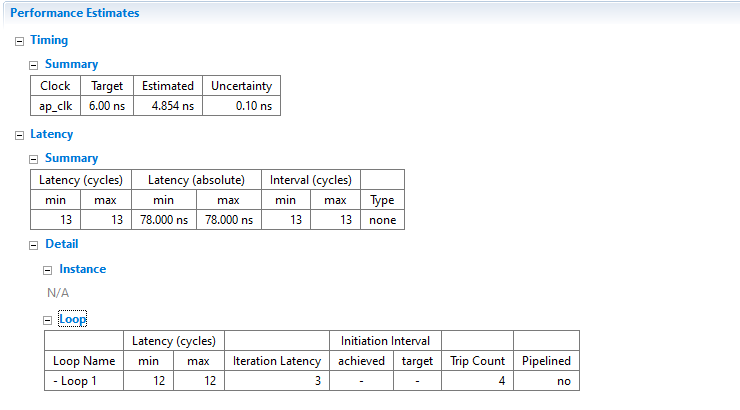


Моделирование

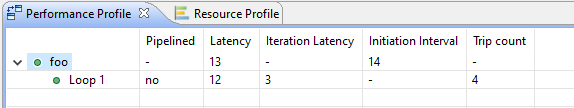


Моделирование пройдено без ошибок.

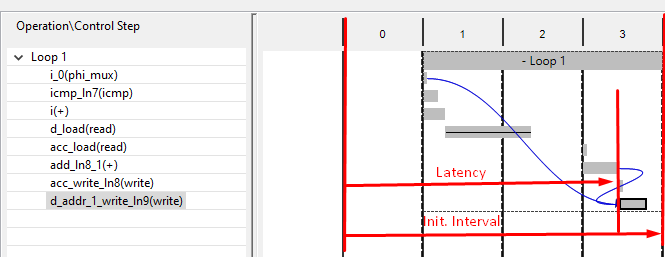
Симуляция с настройками и интерфейсом по умолчанию:



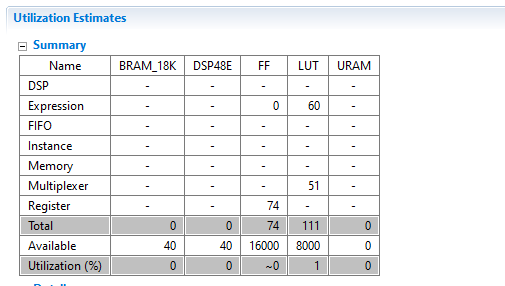
Полученное значение задержки укладывается в заданное целевое значение.



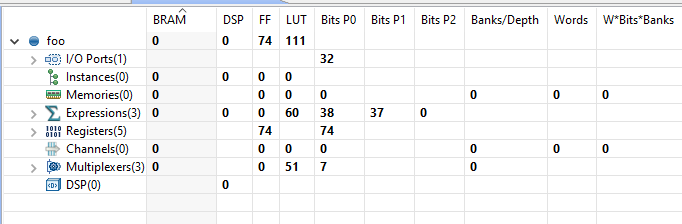
Период задержки до получения результатов составляет 13 тактов, интервал инициализации составляет 14 тактов. Всего на одну итерацию приходится 3 такта.



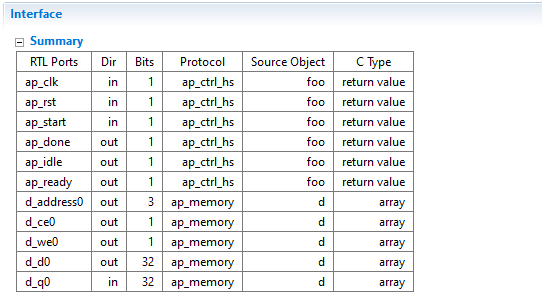
Использование ресурсов



Данное решение потребует на микросхеме 74 регистра и 111 LUT.

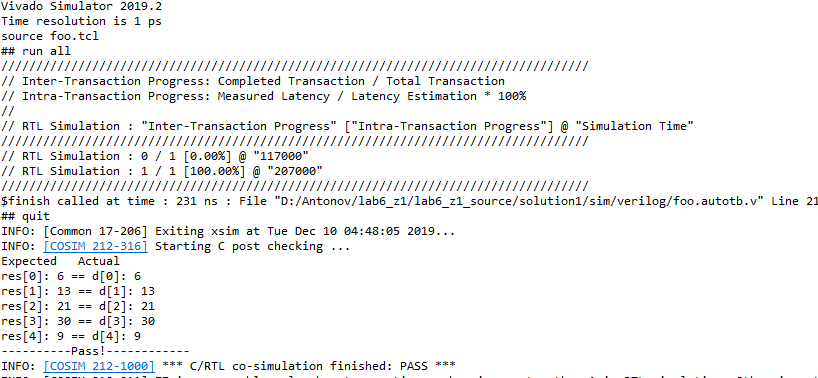


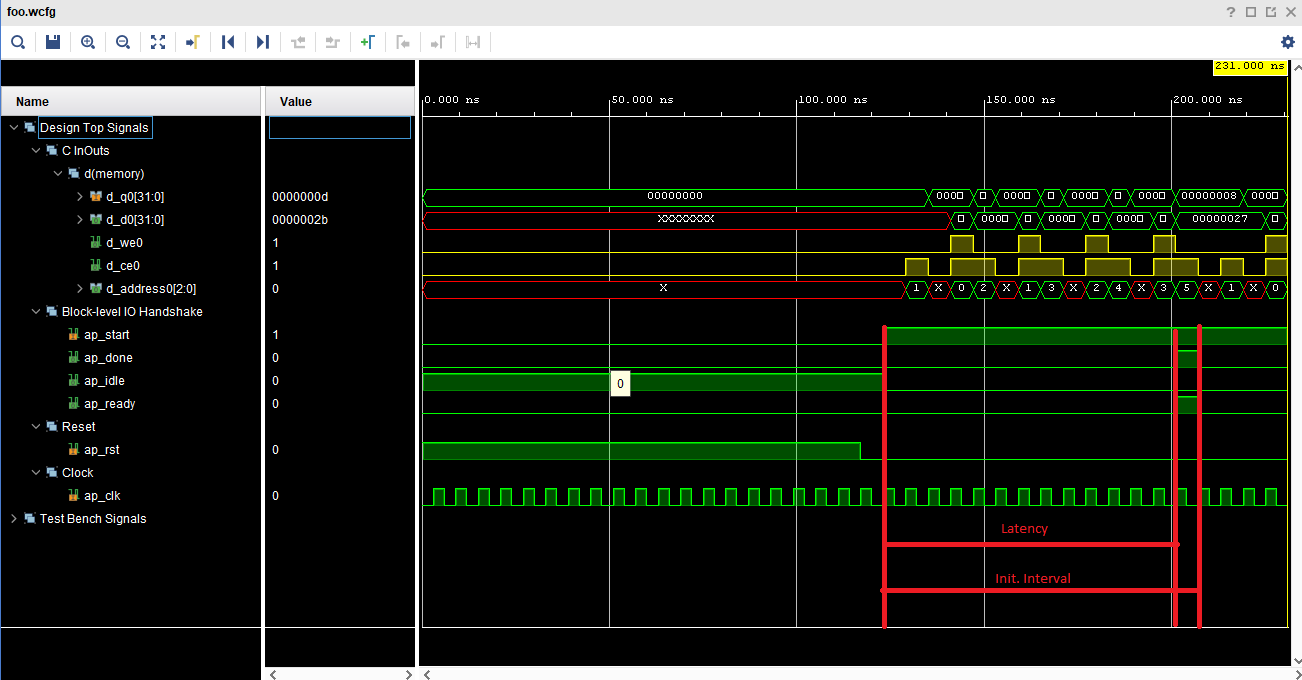
Интерфейс



Ap\_memory – интерфейс по умолчанию для аргументов – массивов. Может быть указан только для массивов.

C/RTL моделирование



На временной диаграмме отображена задержка и интервал инициализации. 

## Решение 2

Исходные файлы

#define DCT\_SIZE 5

void foo(int d[DCT\_SIZE]){

static int acc = 0;

int i;

for (i = 0; i < 4; i++) {

acc += d[i+1];

d[i] = acc;

}

}

Тест

#include <stdio.h>

int main()

{

int d[5];

int i;

for (i = 0; i < 5; i++) {

d[i] = i + 5;

}

foo(d);

int res[5] = {6, 13, 21, 30, 9};

int pass;

fprintf(stdout, "Expected Actual\n");

for (i = 0; i < 5; i++) {

fprintf(stdout, "res[%d]: %d == d[%d]: %d\n", i, res[i], i, d[i]);

if (res[i] == d[i]) {

pass = 1;

} else {

pass = 0;

break;

}

}

if (pass) {

fprintf(stdout, "----------Pass!------------\n");

return 0;

} else {

fprintf(stderr, "----------Fail!------------\n");

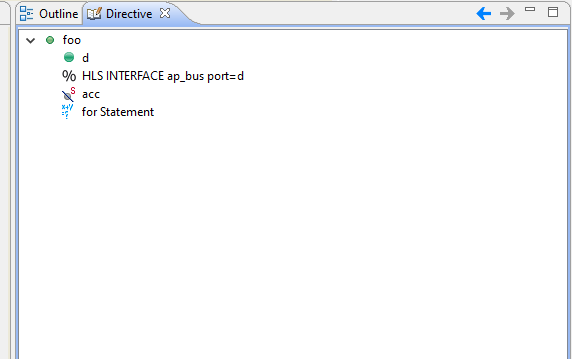
return 1;

}

}

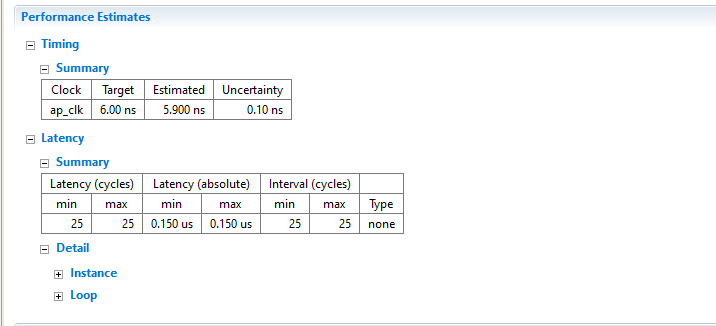
Моделирование

Задание интерфейса ap\_bus



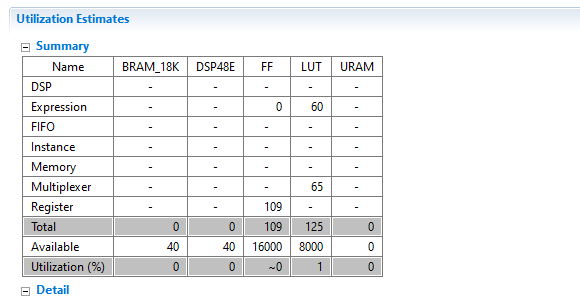
Синтез

Производительность

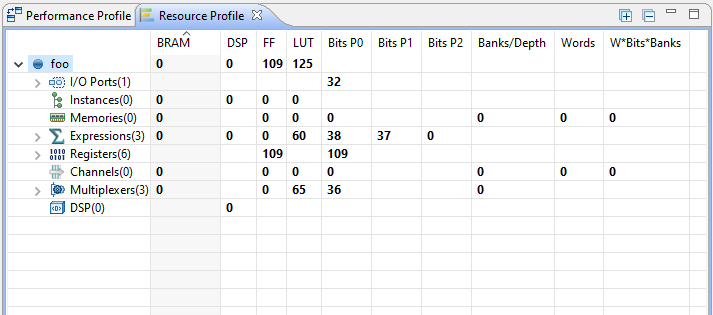


Полученная величина задержки укладывается в заданное значение.

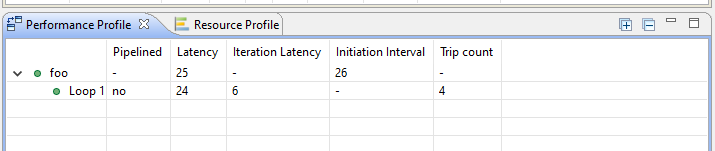
Использование ресурсов



Для данного решения на микросхеме будет задействовано 109 портов и 125 LUT.

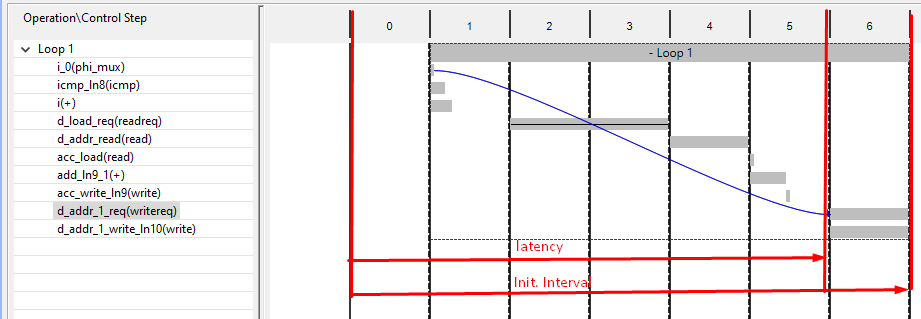


Данные, отображенные на рисунке соответствуют описанным выше.

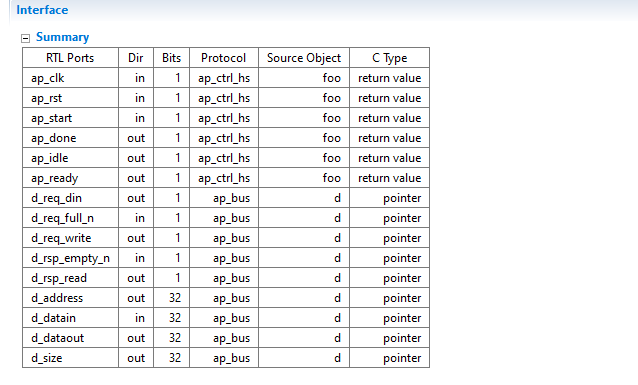


Задержка для каждой итерации составляет 6 тактов, всего происходит 4 повтора итераций. Полный цикл занимает 26 тактов.

Приведем диаграмму, определяющую последовательность операций для каждой итерации.

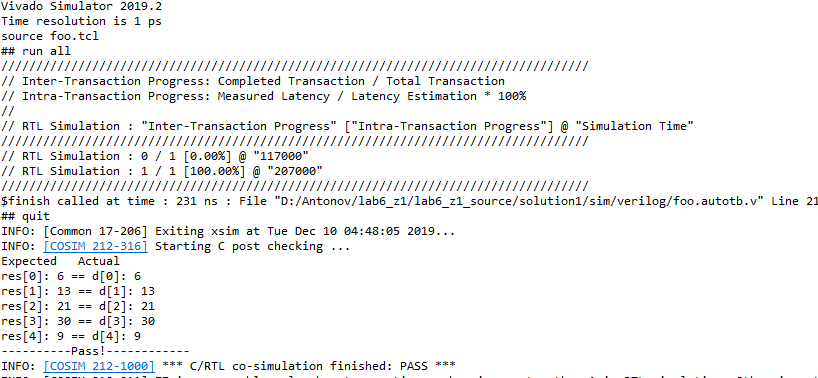


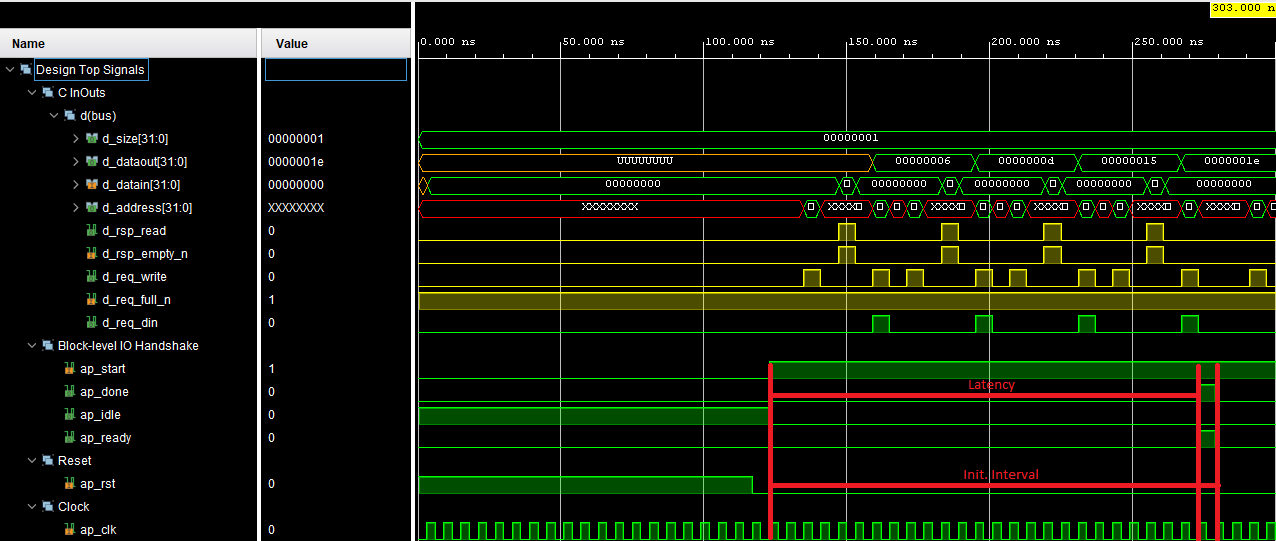
Интерфейс



Все порты с протоколами ap\_ctrl\_hs представляют собой порты block level I/O. Протокол ap\_bus реализует переменные указателя и передачи по ссылке в качестве шины общего назначения, аналогичной типичному интерфейсу DMA.

C/RTL моделирование



На временной диаграмме изображён интервал инициализации и задержка.

# Выводы

В первом варианте решения был задействован интерфейс по умолчанию ap\_memory, который предназначен к использованию только для аргументов – массивов. Во втором варианте был использован интерфейс ap\_bus. Протокол ap\_bus реализует переменные указателя и передачи по ссылке в качестве шины общего назначения, аналогичной типичному интерфейсу DMA.

Результат полученной величины задержки и количество тактов интервала инициализации с протоколом по умолчанию является меньшим. Таким образом первое решение является оптимальным.