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 2

Студент:\_\_\_\_\_\_Волкова М.Д

Гр. № \_\_\_\_\_\_\_ [3540901/91501](https://vk.com/im?sel=c136)

Преподаватель: Антонов А.П.

Санкт-Петербург

2020

Оглавление

[Решение 1 4](#_Toc23606168)

[Решение 2 8](#_Toc23606169)

[Решение 3 12](#_Toc23606170)

[Решение 4 16](#_Toc23606171)

[Выводы 22](#_Toc23606172)

# Решение 1

Создание файла с исходным кодом

void foo (int \*d){

static int acc = 0;

int i = 0;

acc += d[i];

d[i] = acc;

}

Создание тестового файла

#include <stdio.h>

int main()

{

int d[1];

d[0] = 0;

foo(d);

int res = 0;

int pass;

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

if (res == d[0]) {

pass = 1;

} else {

pass = 0;

}

if (pass) {

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

return 0;

} else {

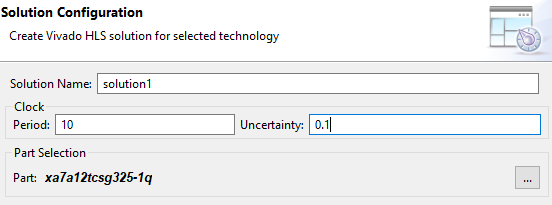
fprintf(stderr, "----------Fail!------------\n");

return 1;

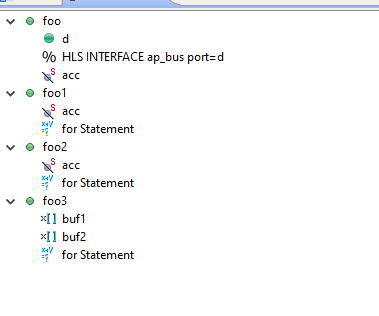
}

}

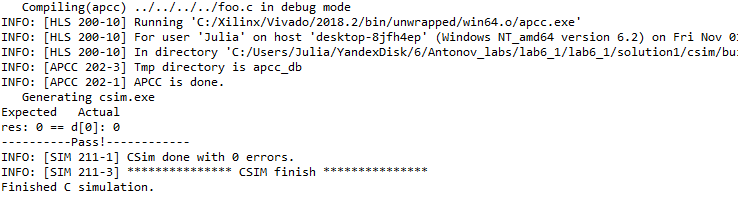
Создание решения с заданными параметрами



Задание интерфейса ap\_bus



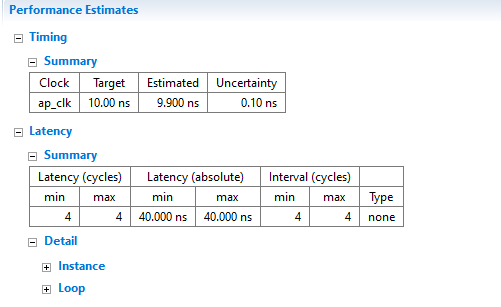
Моделирование



Результат моделирования успешный, заданное и полученное значения совпадают.

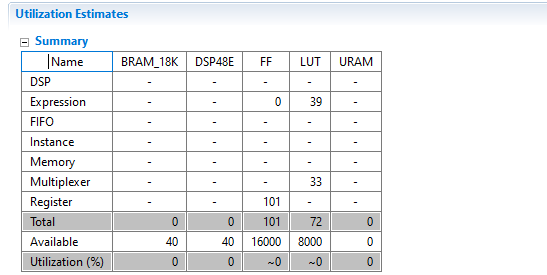
Синтез

Производительность



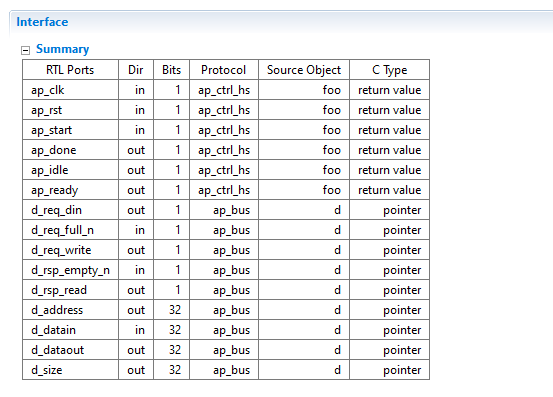
Полученная величина задержки укладывается в целевое значение.

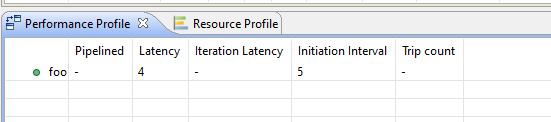
Использование ресурсов



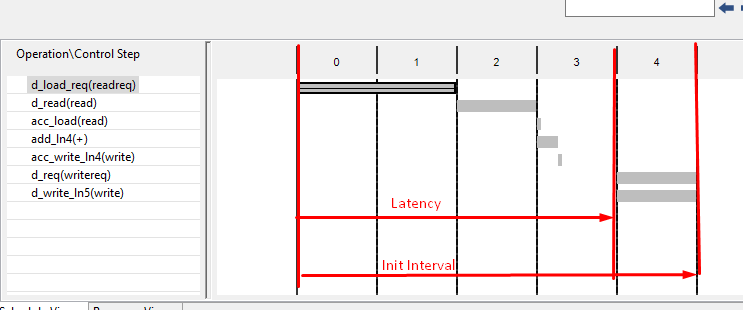
Данное решение потребует на микросхеме 101 регистр и 72 LUT.

Интерфейсы соответствуют заданному ap\_bus при создании решения:

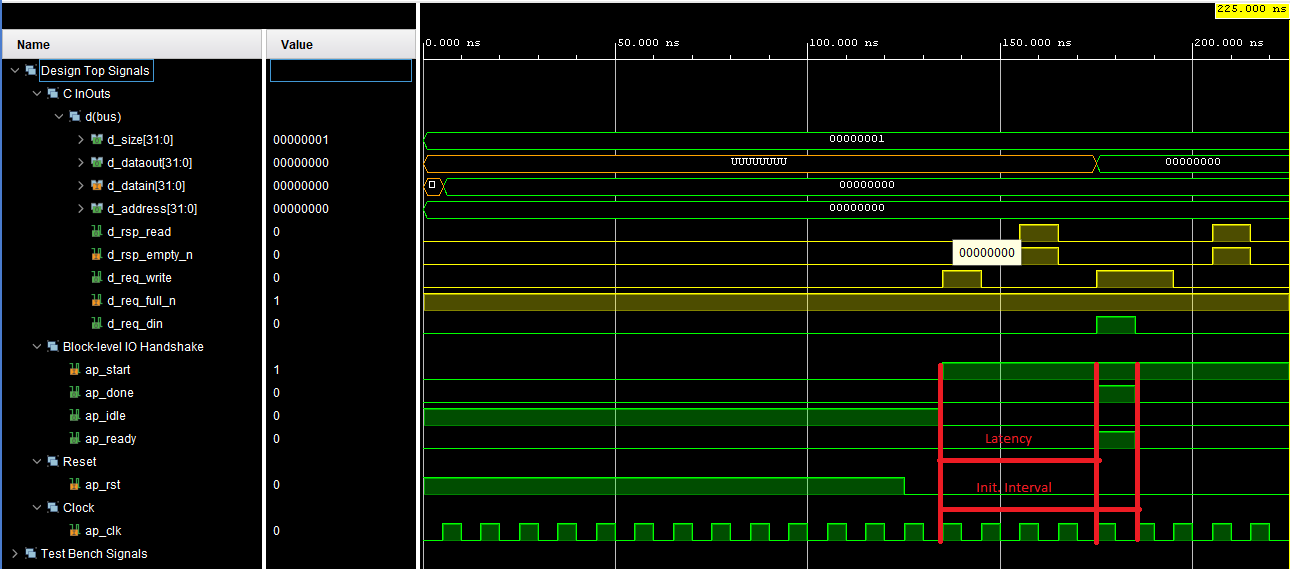




Задержка составляет 4 такта, интервал инициализации составляет 5 тактов.



C/RTL моделирование



На временной диаграмме отображен интервал инициализации и задержка.

# Решение 2

Создание исходного файла

void foo1 (int \*d){

static int acc = 0;

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

acc += d[i];

d[i] = acc;

}

}

Создание тестового файла

#include <stdio.h>

int main()

{

int d[4] = {0, 1, 2, 3};;

foo1(d);

int res[4] = {0, 1, 3, 6};

int pass;

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

for (int i = 0; i < 4; 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 {

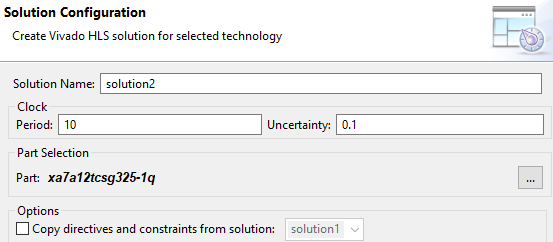
fprintf(stderr, "----------Fail!------------\n");

return 1;

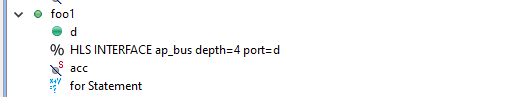
}

}

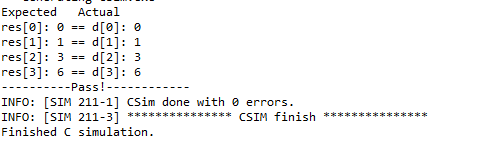
Настройка решения



Установка интерфейса

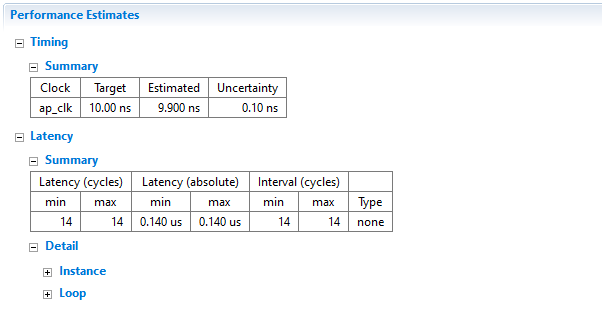


Моделирование



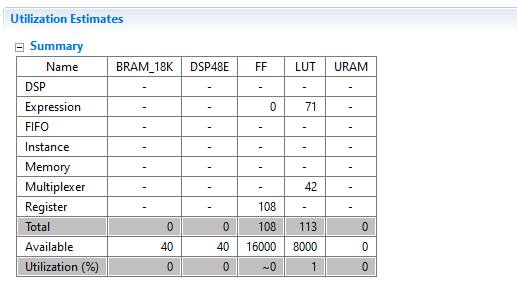
Синтез

Производительность

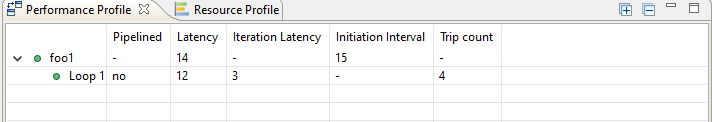


Полученное значение задержки укладывается в заданное и соответствует решению 1.

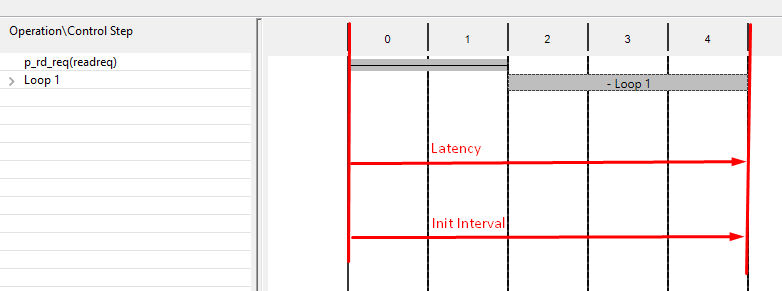
Использование ресурсов



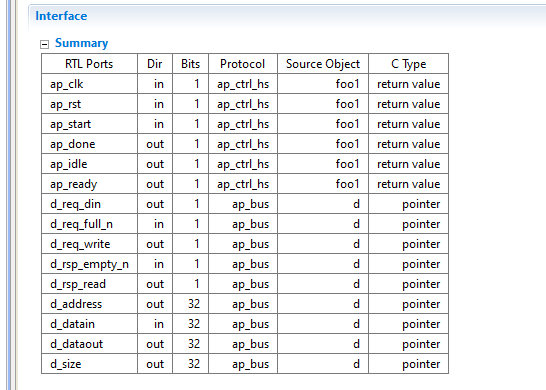
Данное решение потребует на микросхеме 108 регистров и 113 элементов LUT. По сравнению с решением 1 значительно увеличилось число LUT.



Задержка на одной итерации занимает 3 такта, для всего цикла – 12 тактов (так как производится 4 итерации). Интервал инициализации занимает 15 тактов.



Интерфейс



Интерфейс соответствуют заданному при настройке проекта ap\_bus.

C/RTL моделирование



На временной диаграмме отображена задержка и интервал инициализации.

# Решение 3

Создание исходного файла

void foo2 (int \*d){

static int acc = 0;

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

acc += \*(d+i);

\*(d+i) = acc;

}

}

Создание тестового файла

#include <stdio.h>

int main()

{

int d[4] = {0, 1, 2, 3};;

foo2(d);

int res[4] = {0, 1, 3, 6};

int pass;

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

for (int i = 0; i < 4; 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 {

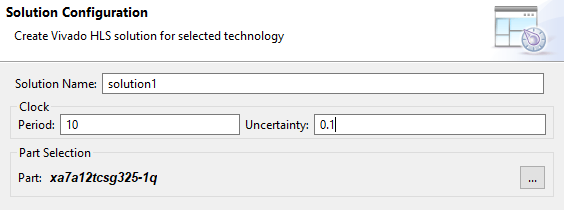
fprintf(stderr, "----------Fail!------------\n");

return 1;

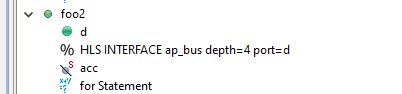
}

}

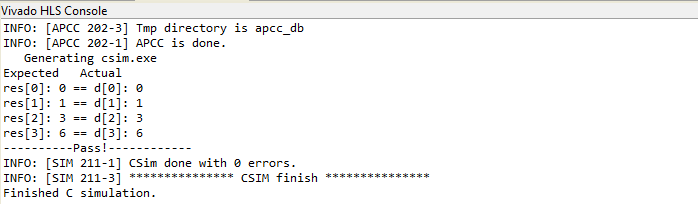
Настройка решения



Настройка интерфейса



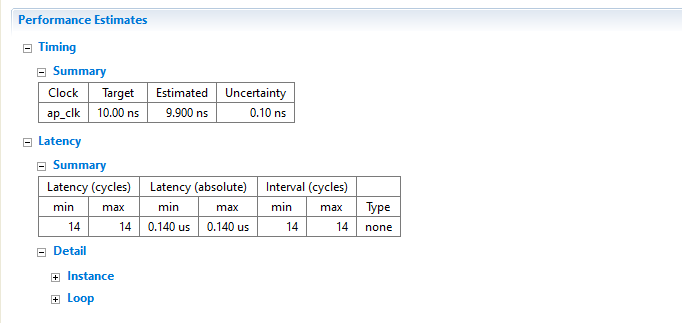
Моделирование



Моделирование пройдено успешно.

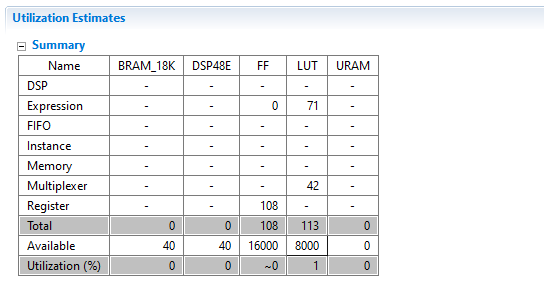
Синтез

Производительность

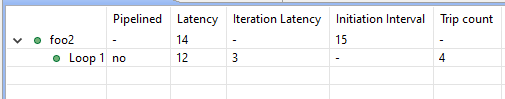


Полученная величина задержки укладывается в заданное значение и соответствует предыдущим решениям.

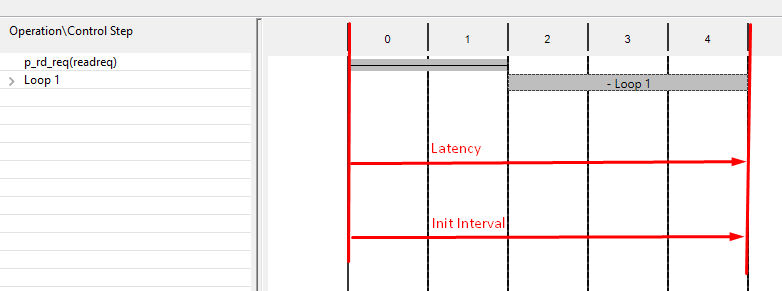
Использование ресурсов



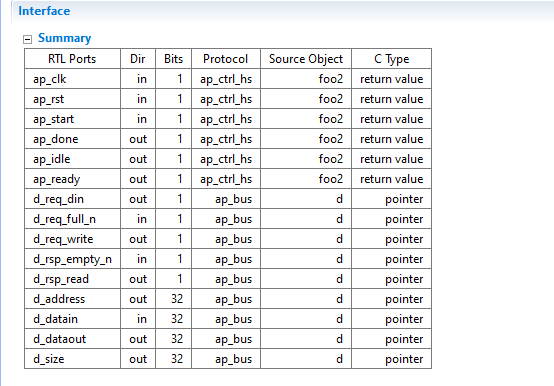
Данное решение потребует на микросхеме 108 регистров и 113 элементов LUT. Полученные значения соответствуют значениям в решении 2 так как оба решения определяют множественную запись и чтение.



Величина задержки для одной итерации составляет 3 такта, количество итераций – 4, полная величина задержки равна 14 тактам, интервал инициализации составляет 15 тактов.

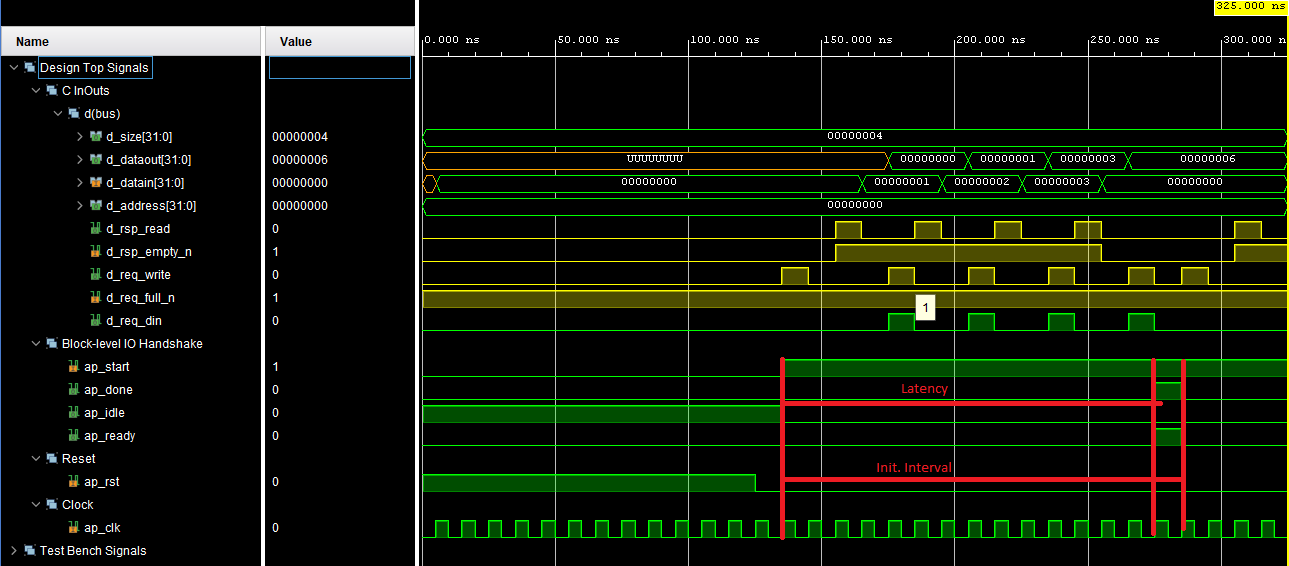


Интерфейс



Интерфейс соответствует заданному ap\_bus при конфигурировании решения.

C/RTL моделирование



На временной диаграмме отображены задержка и интервал инициализации.

# Решение 4

Создание исходного файла

void foo3 (int \*d){

int buf1[4], buf2[4];

int i;

memcpy(buf1, d, 4\*sizeof(int));

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

buf2[i] = buf1[3-i];

}

memcpy(d, buf2, 4\*sizeof(int));

}

void \* memcpy( void \* destptr, const void \* srcptr, size\_t num );

Функция memcpy копирует num байтов первого блока памяти, на который ссылается указатель srcptr, во второй блок памяти, на который ссылается указатель destptr.

Тип данных объектов, на которые указывают как srcptr так и destptr не имеют никакого значения. Так как эта функция работает с бинарными данными.

Функция не проверяет, есть ли символ завершения в srcptr, она всегда копирует количество байтов, указанное в num.

Чтобы избежать переполнения блока памяти destptr, размер destptr должен быть не менее num байтов. Однако, может возникнуть ситуация, когда destptr и srcptr пересекутся. Поэтому, для перекрытия блоков памяти, функция memmove является более безопасным подходом.

Параметры:

Destptr - Указатель на блок памяти назначения (куда будут копироваться байты данных), имеет тип данных void.

Srcptr - Указатель на блок памяти источник (т. е., откуда будут копироваться байты данных), имеет тип данных void.

Num - Количество копируемых байтов.

Возвращаемое значение - Указатель на блок памяти назначения.

Создание тестового файла

#include <stdio.h>

int main()

{

int d[4] = {0, 1, 2, 3};;

foo3(d);

int res[4] = {3, 2, 1, 0};

int pass;

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

for (int i = 0; i < 4; 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 {

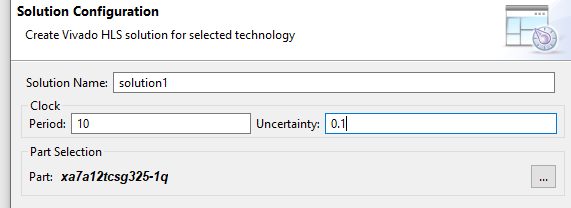
fprintf(stderr, "----------Fail!------------\n");

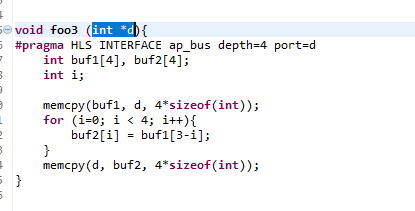
return 1;

}

}

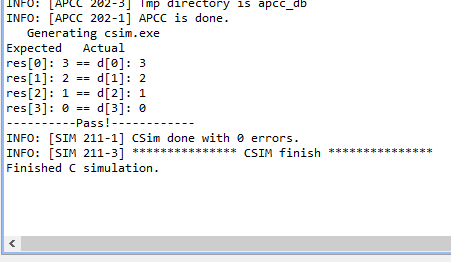
Настройка решения





В этот раз директива интерфейса вставлена прямо в исходный код, иначе синтез не будет проходить из за использования memcpy без инициализированного интерфейса шины.

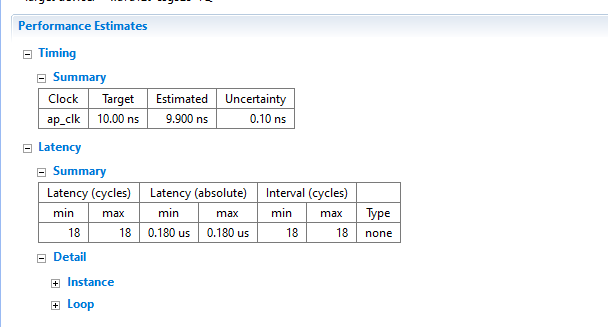
Моделирование



Моделирование выполнено успешно

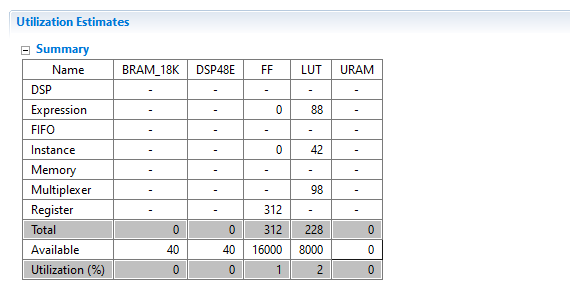
Синтез

Производительность

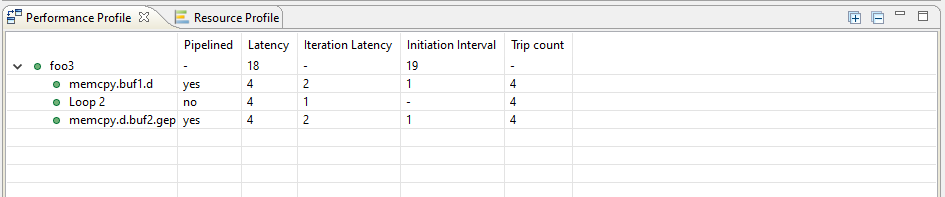


Полученная величина задержки укладывается в заданное значение и соответствует предыдущим решениям.

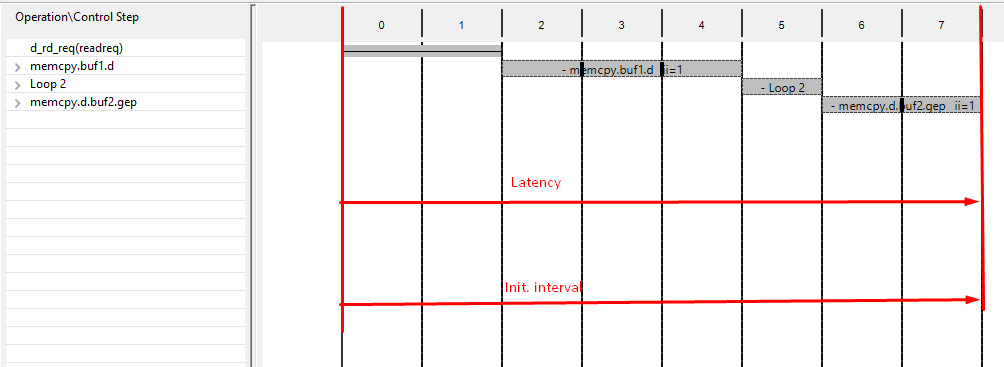
Использование ресурсов



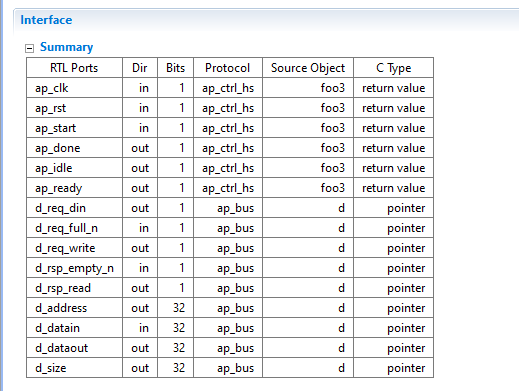
Данное решение требует 312 регистров и 228 элементов LUT, что значительно больше по сравнению с предыдущими решениями. Число требуемых регистров и элементов LUT возросло примерно в 3 раза по сравнению с предыдущими решениями.



Задержка одной итерации цикла составляет 1 такт, для всего цикла – 4 такта. Задержки для выполнения memcpy по 2 такта. Интервал инициализации составляет 19 тактов.

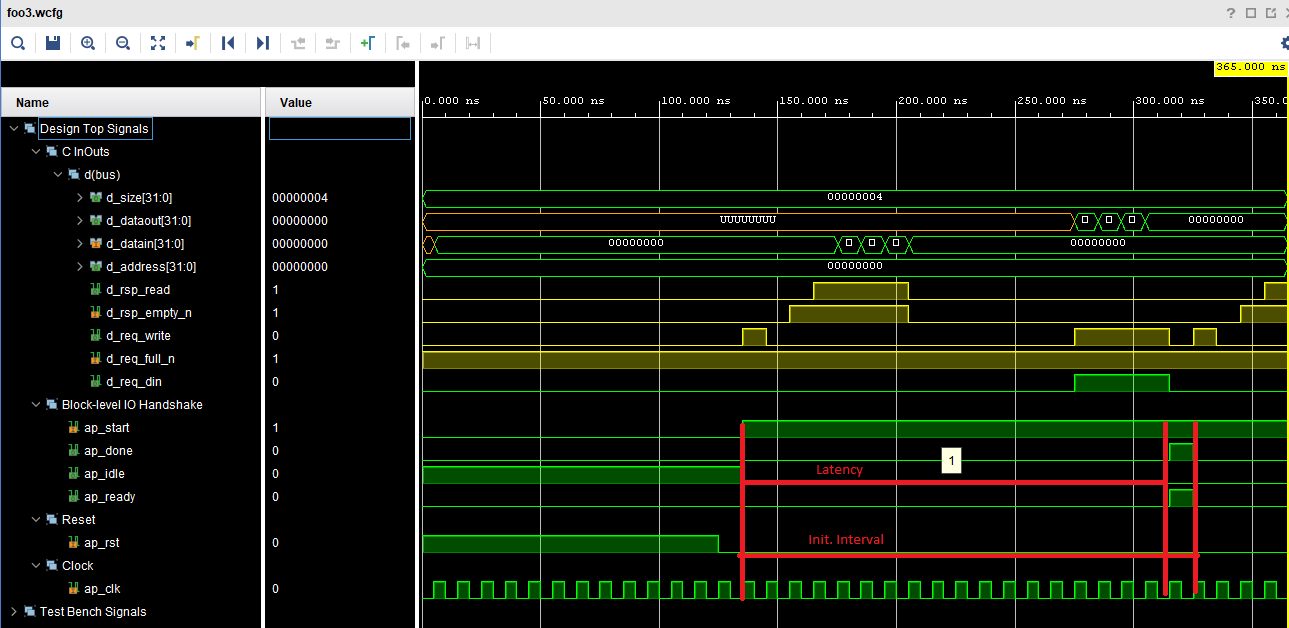


Интерфейс



Интерфейс соответствует заданному в начале решения ap\_bus.

C/RTL моделирование



На временной диаграмме отображена задержка и интервал инициализации.

# Выводы

В ходе работы были построены четыре решения, одно с одиночным вариантом чтения и записи, два – с множественным и одно с режимом потокового обмена. Был использован интерфейс ap\_bus. Применялись следующие порты: D\_datain – входные данные, D\_req\_full\_n - активный низкий уровень сигнала указывает что мост полный, D-rsp\_empty\_n – указывает на готовность принятия данных, D\_dataout – выходные данные и другие, например, определяющие адрес и размер, готовность чтения и записи. Временные диаграммы и анализ решений приведены в отчете.