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

Лабораторная работа 4

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Port-Level IO Protocols

Задание 1

Студент:\_\_\_\_\_\_Волкова М.Д

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

Преподаватель: Антонов А.П.

Санкт-Петербург

2020

Оглавление

[Задание 4](#_Toc27017705)

[Ход работы 6](#_Toc27017706)

[Решение 1 6](#_Toc27017707)

[Решение 2 10](#_Toc27017708)

[Вывод 14](#_Toc27017709)

# Задание

* Создать проект lab4\_1
* Подключить файл lab4\_1.c (папка source)
* Создать тест lab4\_1\_test.c на основе теста Подключить тест lab1\_1\_test.c
* Микросхема: xa7a12tcsg325-1q
* Сделать solution1
  + задать: clock period 6; clock\_uncertainty 0.1
  + 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
    - Открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
* Сделать solution2
  + Задать протоколы
    - a: ap\_hs
    - b: ap\_ask
    - \*c: ap\_hs
    - \*d: ap\_vld
    - \*p\_y: ap\_ask
  + 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
    - 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 port\_level

# Ход работы

## Решение 1

Создание и конфигурирование решения.

Создание тест – файла

#include <stdio.h>

int main()

{

int res;

// For adders

int refOut[3] = {230, 640, 1250};

int pOut[3] = {240, 650, 1260};

int pass;

int i;

int inA = 10;

int inB = 20;

int inC = 30;

int inD = 40;

int inP = 1;

// Call the adder for 5 transactions

for (i=0; i<3; i++)

{

res = lab4\_1(inA, inB, &inC, &inD, &inP);

fprintf(stdout, " %d\*%d+%d=%d \n", inA, inB, inC, res);

fprintf(stdout, " %d\*%d+%d=%d \n", inA, inB, inD, inP);

// Test the output against expected results

if (res == refOut[i] & inP == pOut[i])

pass = 1;

else

pass = 0;

inA=inA+10;

inB=inB+10;

inC=inC+10;

inD=inD+10;

}

if (pass)

{

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

return 0;

}

else

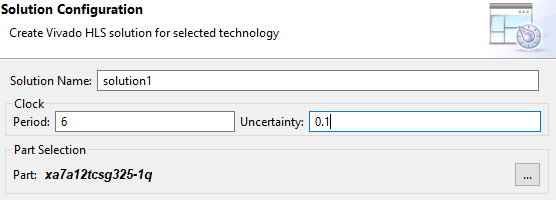
{

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

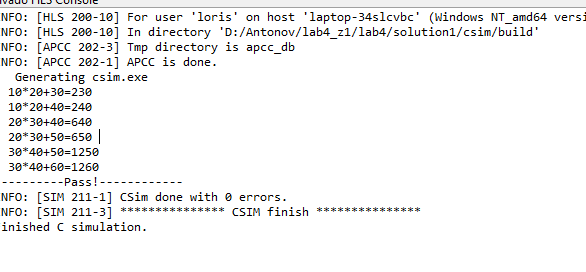
return 1;

}

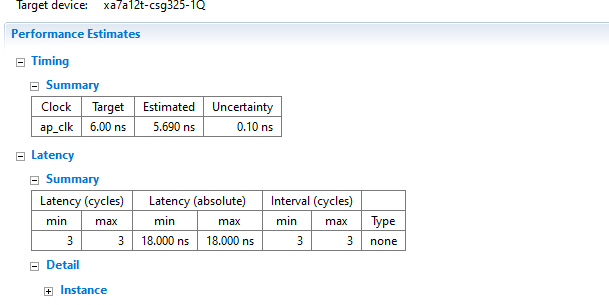
}



Моделирование

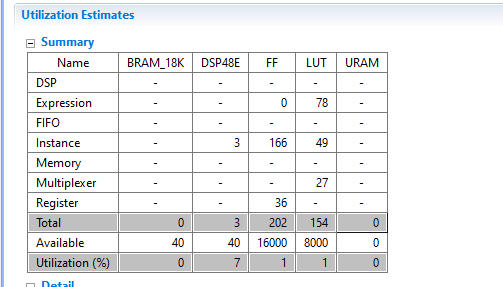


Производительность



На изображении видно, что полученная величина задержки укладывается в целевое значение.

Использование ресурсов

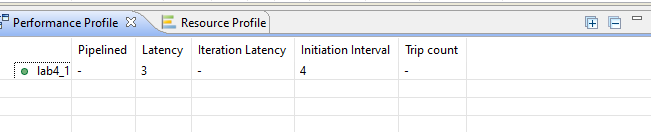


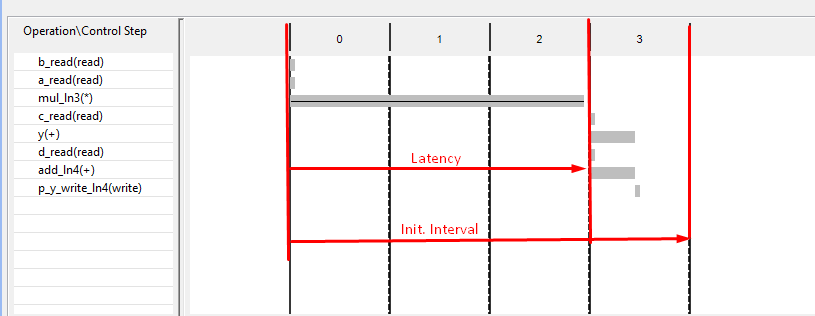
Данный проект будет занимать на микросхеме:

4 DSP блока, где будут задействованы сумматор и умножитель.

202 регистра для хранения и считывания данных (чисел).

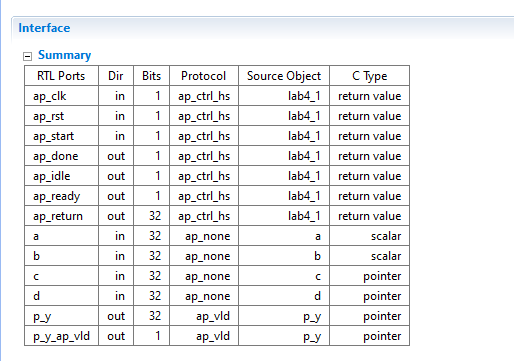
154 LUT.





На изображениях, приведенных выше видно, что задержка до получения результата составляет 3 такта, а интервал инициализации – 4 такта.

Интерфейс



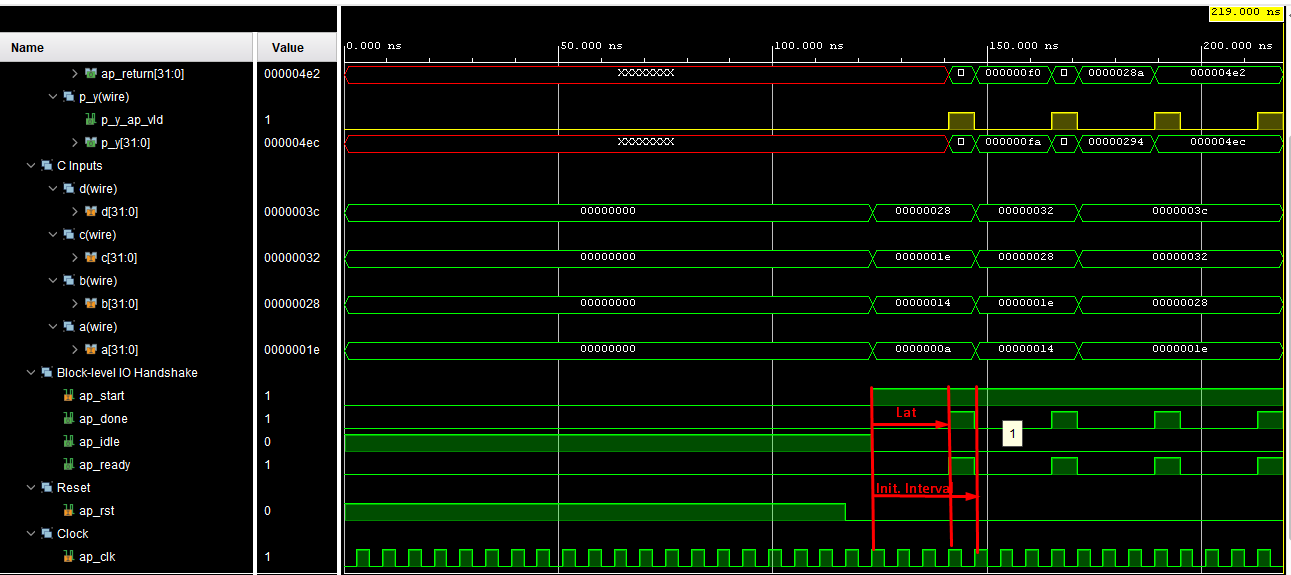
Конструкция имеет 6 портов данных.

Входные порты: a, b, c, d являются 32-битными входами и имеют входы / выходы, протокол ap\_none.

Конструкция имеет 32-битный выходной порт для возврата функции p\_y, p\_y\_ap\_vld – протокол по умолчанию для портов выхода.

Управляющие сигналы ap\_clk, ap\_rst и ap\_ \* автоматически добавляются в каждый дизайн по умолчанию. Ap\_start, ap\_done, ap\_idle и ap\_ready являются сигналами верхнего уровня, используемыми в качестве сигналов подтверждения связи, чтобы указать, когда проект способен принять следующую команду вычисления (ap\_ready), когда начинается следующее вычисление (ap\_start) и когда вычисление завершено (ap\_done).

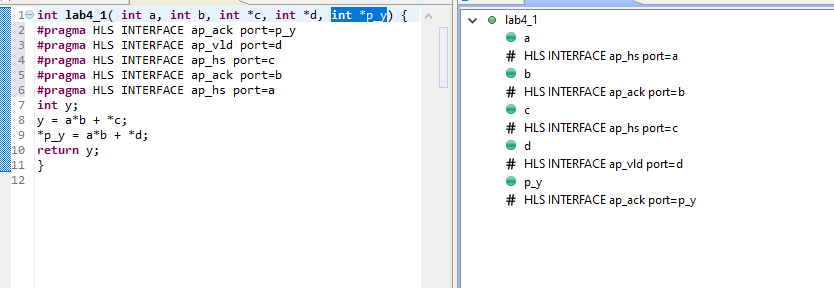
C/RTL моделирование. Временная диаграмма



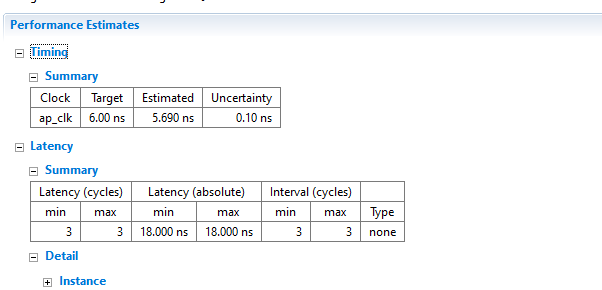
На временной диаграмме отображена задержка и интервал инициализации.

## Решение 2

Добавление директив

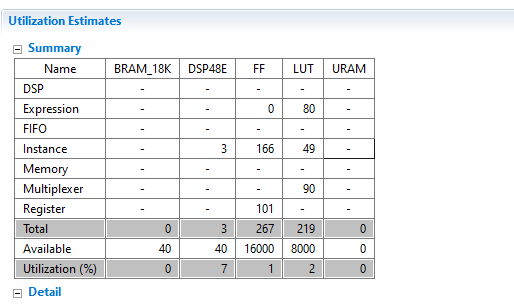


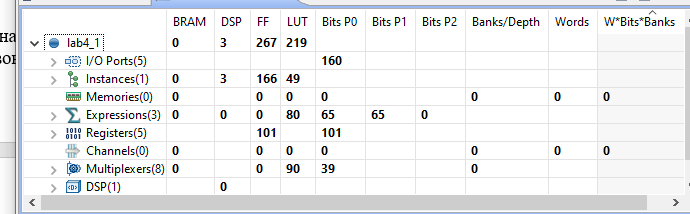
Производительность



По сравнению с предыдущим решением значение полученной задержки не изменилось.

Использование ресурсов



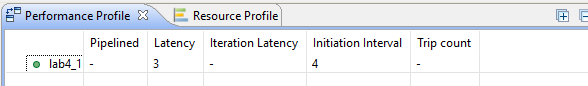


Данный проект будет занимать на микросхеме:

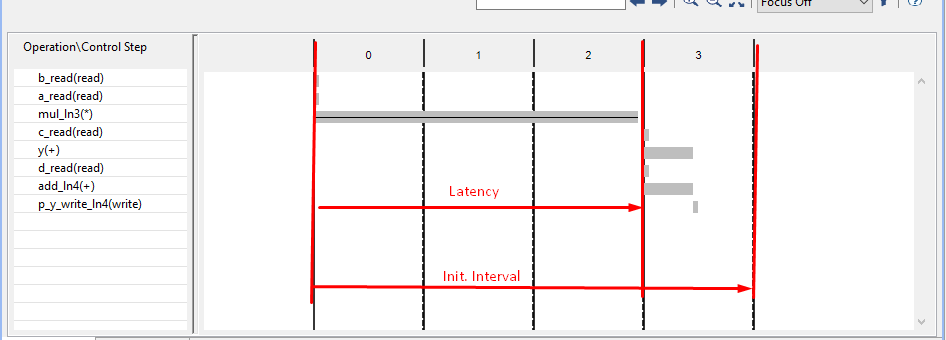
4 DSP блока, где будут задействованы сумматор и умножитель.

268 регистров для хранения и считывания данных (чисел).

160 LUT.



На изображении выше видно, что задержка получения результата составляет 3 такта, а интервал инициализации – 4.



Интерфейс



По сравнению с предыдущим решением ap\_none изменился на заданные протоколы. Для портов входа и выхода используются следующие протоколы: ap\_hs, ap\_vld, ap\_ack.

Протоколы ввода-вывода на уровне порта ap\_hs обеспечивает наибольшую гибкость в процессе разработки.

Протокол ввода-вывода уровня порта ap\_hs предоставляет следующие сигналы:

- Порт данных

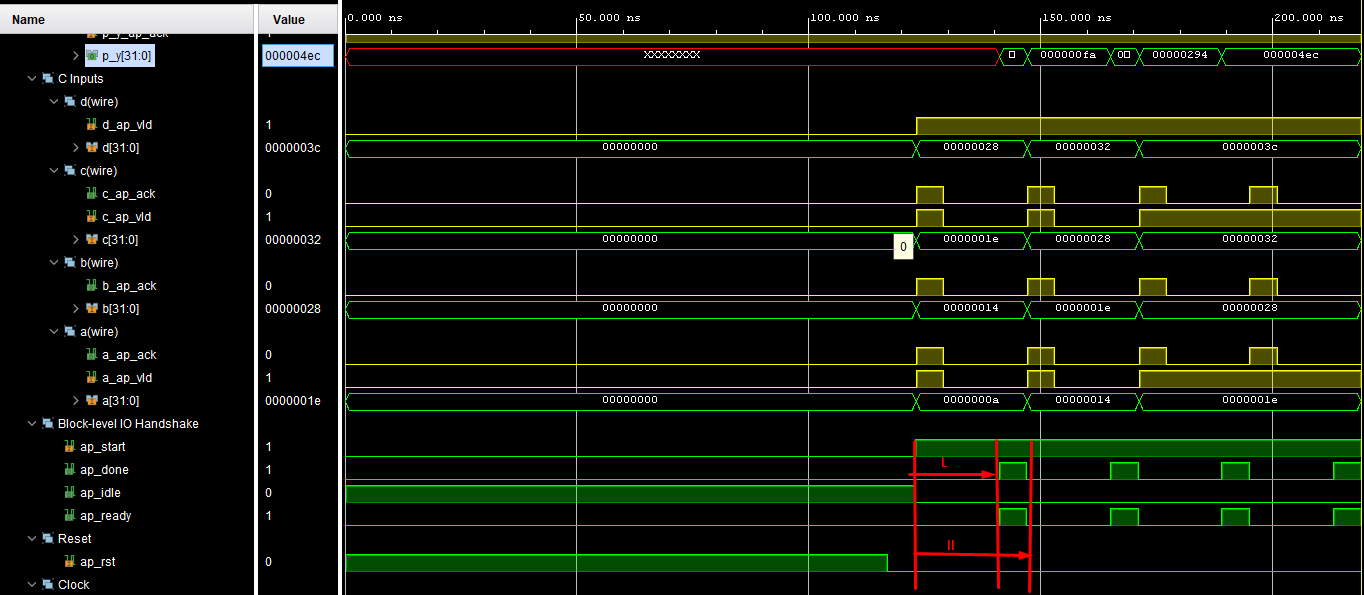
- Сигнал указания момента использования данных (ack)

- Действительный сигнал для указания, когда данные считываются (vld)

Протокол ввода-вывода уровня порта ap\_none является самым простым типом интерфейса и не имеет никаких других сигналов, связанных с ним. Ни входные, ни выходные сигналы данных не имеют связанных портов управления, которые указывают, когда данные считываются или записываются. Единственными портами в конструкции RTL являются порты, указанные в исходном коде. Интерфейс ap\_none не требует дополнительных аппаратных издержек.

C/RTL моделирование

На временной диаграмме отображена задержка и интервал инициализации.



# Вывод

Протоколы ввода-вывода на уровне порта ap\_hs обеспечивает наибольшую гибкость в процессе разработки.

Протокол ввода-вывода уровня порта ap\_hs предоставляет следующие сигналы:

- Порт данных

- Сигнал указания момента использования данных (ack)

- Действительный сигнал для указания, когда данные считываются (vld)

Протокол ввода-вывода уровня порта ap\_none является самым простым типом интерфейса и не имеет никаких других сигналов, связанных с ним. Ни входные, ни выходные сигналы данных не имеют связанных портов управления, которые указывают, когда данные считываются или записываются. Единственными портами в конструкции RTL являются порты, указанные в исходном коде. Интерфейс ap\_none не требует дополнительных аппаратных издержек.