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

Лабораторная работа 1

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Введение в Vivado HLS

Задание 1

Студент:\_\_\_\_\_\_Волкова М.Д

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

Преподаватель: Антонов А.П.

Санкт-Петербург

2020

Оглавление

[1. Задание 3](#_Toc20688026)

[2. Ход работы 4](#_Toc20688027)

[2.1. Решение 1 4](#_Toc20688028)

[2.1.1. Моделирование 5](#_Toc20688029)

[2.1.2. Синтез 6](#_Toc20688030)

[2.1.3. C/RTL моделирование 8](#_Toc20688031)

[2.2. Решение 2 9](#_Toc20688032)

[2.2.1. Моделирование 9](#_Toc20688033)

[2.2.2. С/RTL моделирование 11](#_Toc20688034)

[3. Выводы 13](#_Toc20688035)

# Задание

• Создать проект lab1\_1

• Подключить файл lab1\_1.c (папка source)

• Подключить тест lab1\_1\_test.c (папка source)

• 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

• 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: clock period 8; clock\_uncertainty 0.1

осуществить моделирование

осуществить синтез

привести в отчете:

• performance estimates=>summary

• utilization estimates=>summary

• Performance Profile

• 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

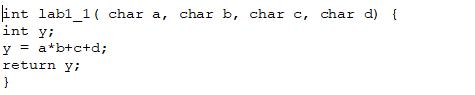
Объяснить отличие двух solutions

# Ход работы

## Решение 1

1. Создание проекта lab1\_1.
2. Подключение файлов lab1\_1.c, lab1\_1\_test.c.

lab1\_1.c:



lab1\_1\_test.c

#include <stdio.h>

int main()

{

int inA, inB, inC, inD;

int res;

// For adders

int refOut[3] = {270, 490, 1310};

int pass;

int i;

inA = 10;

inB = 20;

inC = 30;

inD = 40;

// Call the adder for 5 transactions

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

{

res = lab1\_1(inA, inB, inC, inD);

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

// Test the output against expected results

if (res == refOut[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

{

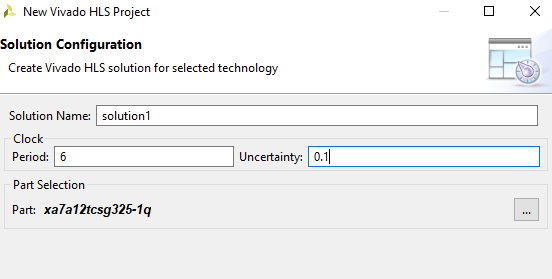
fprintf(stderr, "----------Fail!------------\n");

return 1;

}

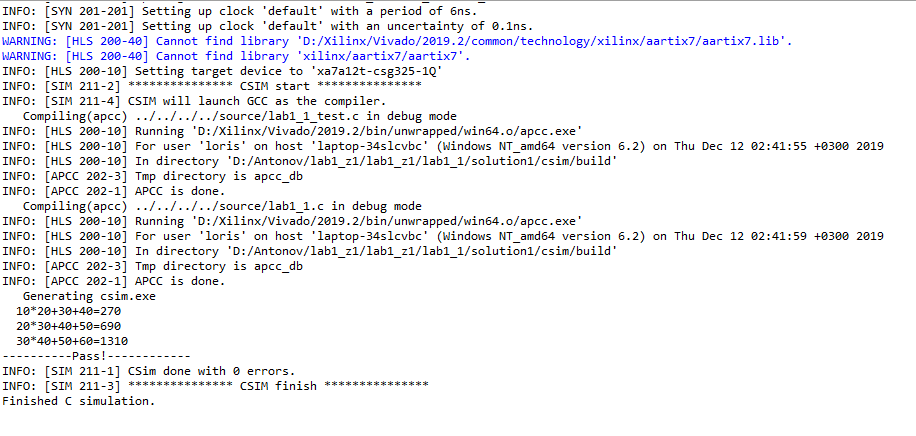
}

1. Конфигурирование решения.



### Моделирование

1. Результат моделирования заданного решения:

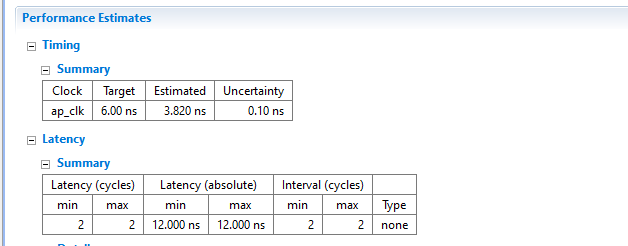


Моделирование выполнено без ошибок. Тест пройден успешно.

### Синтез

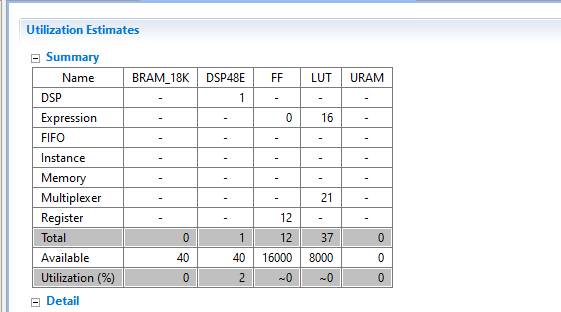
1. Выполним команду Solution-> Run C Synthesis -> Active solution

Производительность



Достигнутая задержка (estimated) равна 3,820 + (погрешность - uncertainty) 0,10. Величина задержки укладывается в заданные требования к тактовой частоте.

Занимаемые ресурсы



Данный проект будет занимать на микросхеме:

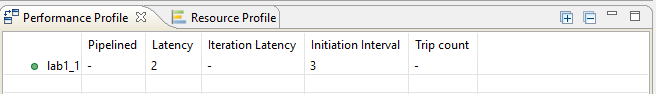
1 DSP блок.

DSP блок – цифровой процессор обработки сигналов, специализированный микропроцессор, особенностью работы которого является поточный характер обработки больших объемов данных в реальном масштабе времени и, обычно, с интенсивным обменом данных с другими внешними устройствами. (Нужен для преобразования сигналов, представленных в виде цифр, как правило, в режиме реального времени). В данном блоке будут использованы сумматор и умножитель.

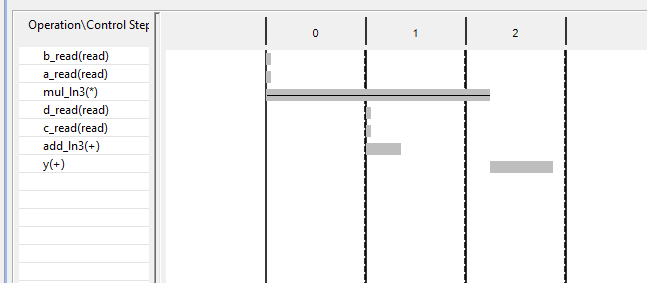
12 регистров для хранения и считывания данных (чисел).

37 LUT

1. Перейдем на вкладку Analysis.



Задержка времени со старта до момента получения значений составляет 2 такта. Задержка времени с момента старта до готовности инициализации (получения новых значений) – 3 такта. Временная диаграмма:

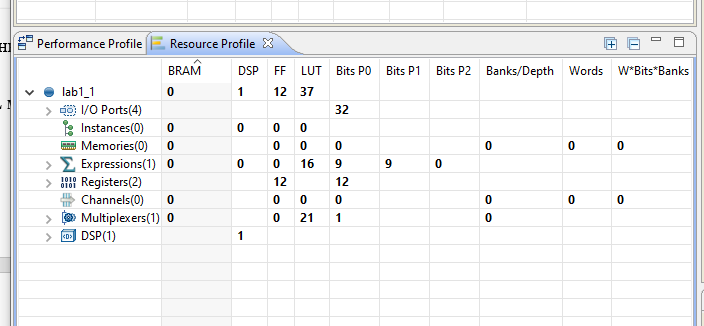


Получение результата происходит следующим образом:

* Первый такт
  + Считывание параметра b
  + Считывание параметра a
  + Умножение параметров a и b
* Второй такт
  + Считывание параметра d
  + Считывание параметра c
  + Сложение параметров d и c
* Третий такт
  + Результаты сложения и умножения складываются -> ответ

На следующем такте возможно поступление новых данных.

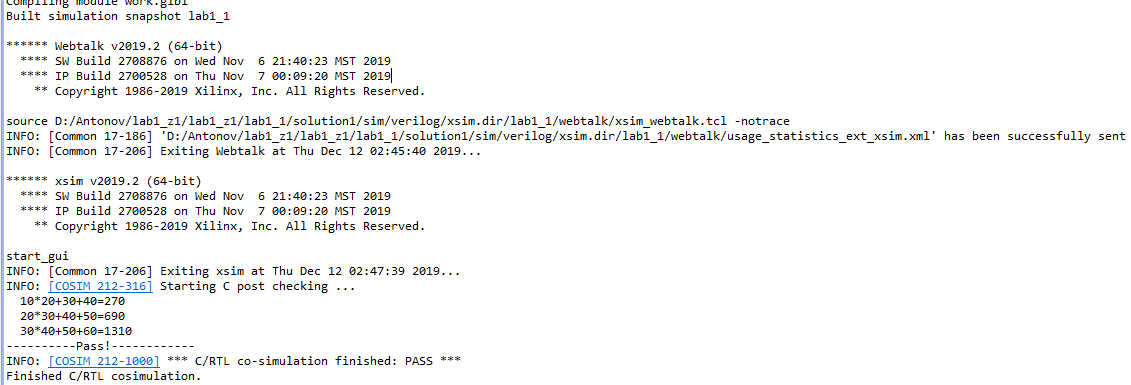
1. Профиль ресурсов



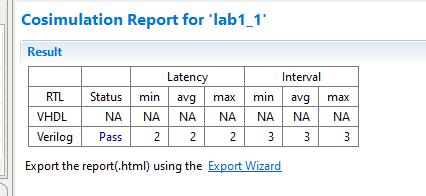
Значения в отчете аналогичны значениям в пункте 5.

### C/RTL моделирование

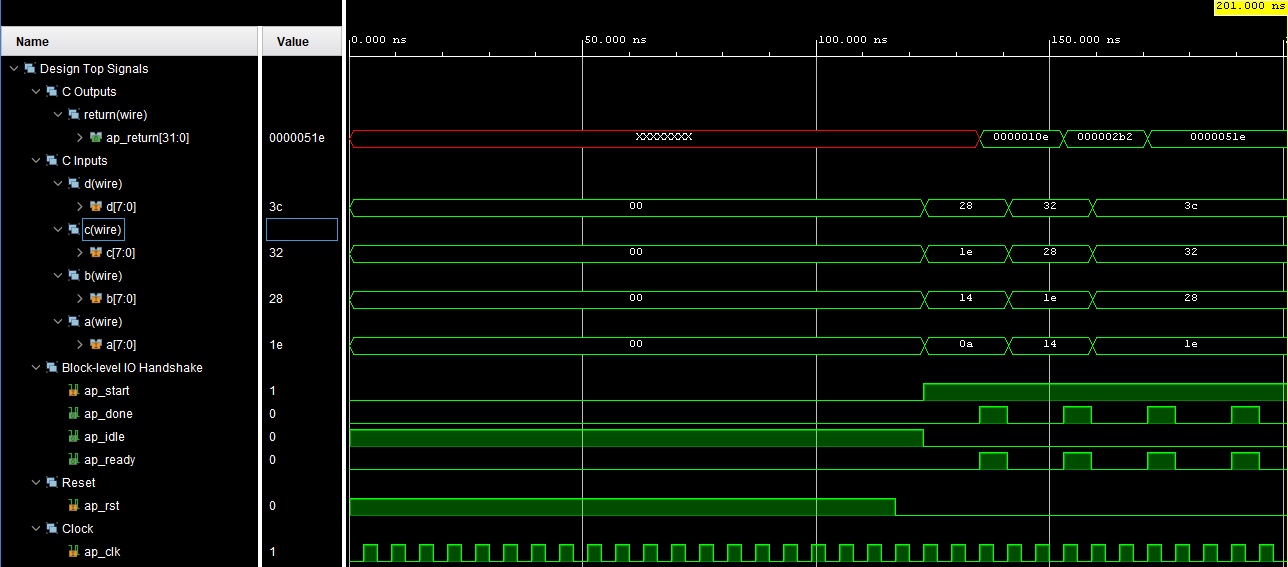
1. Результат выполнения



1. Отчет о выполнении



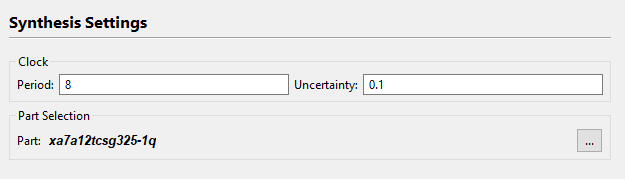
Результаты идентичны полученным в пункте 6.



## Решение 2

### Моделирование

1. Создание и конфигурирование решения.

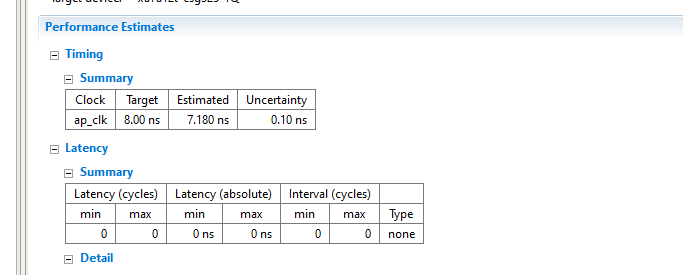


Исходные файлы соответствуют файлам решения 1.

### Синтез

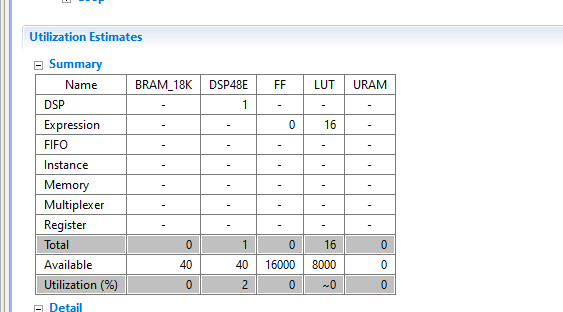
1. Выполним команду Solution-> Run C Synthesis -> Active solution

Производительность



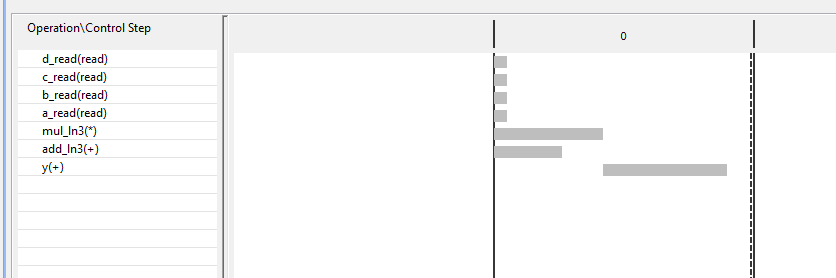
Несмотря на то, что полученное решение укладывается в заданный период тактовой частоты, величина полученной задержки значительно отличается по отношению к первому решению. Результаты свидетельствуют о том, что измененные параметры влекут за собой изменение устройства.

Затрачиваемые ресурсы



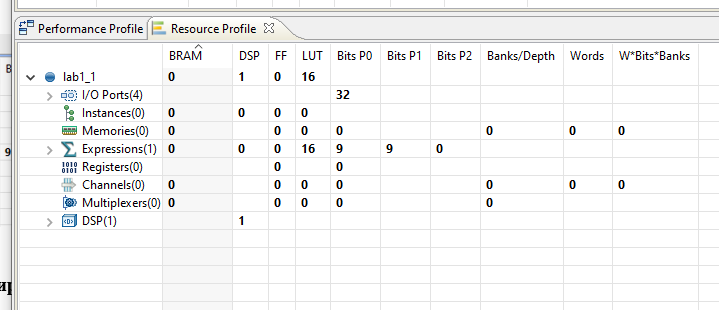
По сравнению с решением 1, в схеме отсутствуют регистры и мультиплексоры.

1. Перейдем на вкладку Analysis.



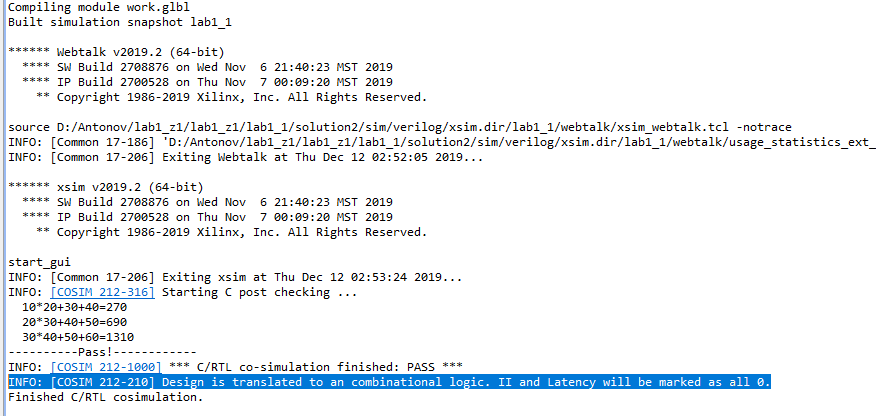
На изображении видно, что до получения ответа требуется 0 тактов, а готовность получения новых данных наступает уже через 1 такт. По сравнению с диаграммой решения 1 произошли изменения: чтения всех параметров происходит одновременно, после этого сразу производятся действия умножения и сложения, сразу после произведения умножения считанных параметров осуществляется переход к сложению и получению конечного результата.

Общий отчет о затраченных ресурсах



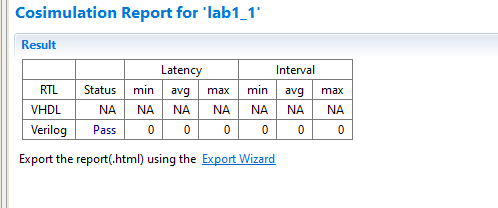
### С/RTL моделирование

1. Результат выполнения

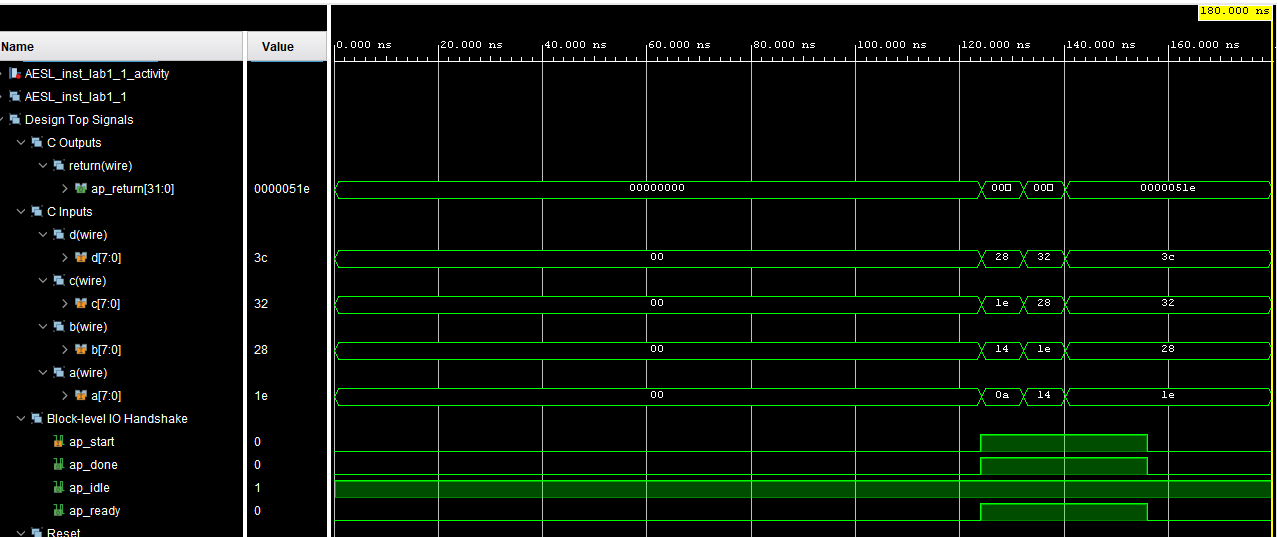


В ходе выполнения было получено информационное сообщение: «Дизайн переводится в комбинационную логику. II и Latency будут помечены как все 0.»

1. Отчет решения



По сравнению с предыдущим данное решение является комбинаторным.



Получение ответа происходит без задержек.

# Выводы

В ходе работы были рассмотрены два решения отличающиеся лишь параметром периода тактовой частоты. В ходе выполнения лабораторной работы был получен результат, что при большем периоде тактов, программа сумела вместить в 1 период чисто комбинаторную схему, в то время как при меньшем периоде это оказалось невозможным. Были получены 2 решения: первое – полный цикл выполнения составил 3 такта, а максимальная задержка обработки сигнала на такте составляет 3.82 нс, и второе –комбинаторное решение, задержка в котором составила значительно большую величину 7.18 нс, но при этом все действия выполняются всего за 1 период тактового сигнала.