

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

Лабораторная работа 1_1

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Введение в Vivado HLS

Задание 1

Студент: Селиверстов Я.А.
Гр. № 3540901/81501

Преподаватель: Антонов А.П.

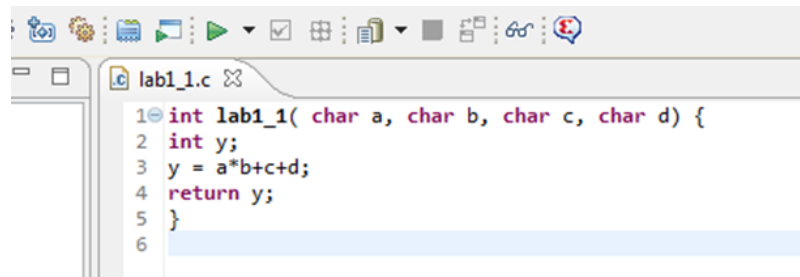
Санкт-Петербург
2019

Оглавление

| | |
|-------------------------------|----|
| 1. Задание | 3 |
| 2. Первое решение | 5 |
| 2.1. Моделирование | 5 |
| 2.2. Синтез | 5 |
| 2.3. C/RTL моделирование..... | 7 |
| 3. Второе решение..... | 8 |
| 3.1. Моделирование | 8 |
| 3.2. Синтез | 9 |
| 3.3. C\RTL моделирование..... | 11 |
| 4. Выводы..... | 13 |

1. Задание

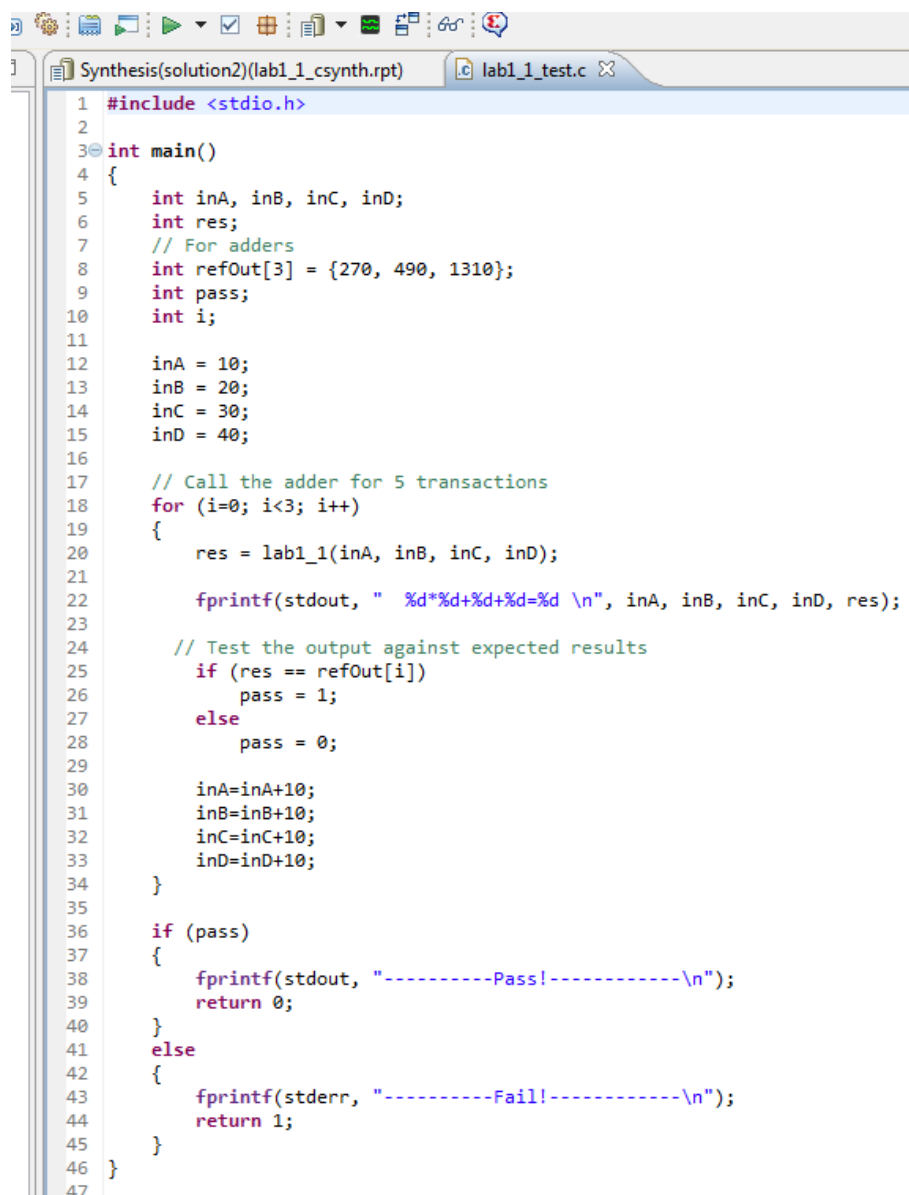
Создать проект, подключив готовые файлы исходного кода устройства и теста к нему. Исходный код:



```
1 int lab1_1( char a, char b, char c, char d) {
2   int y;
3   y = a*b+c+d;
4   return y;
5 }
6
```

Рис. 1.1. Исходный код

Код теста:



```
1 #include <stdio.h>
2
3 int main()
4 {
5     int inA, inB, inC, inD;
6     int res;
7     // For adders
8     int refOut[3] = {270, 490, 1310};
9     int pass;
10    int i;
11
12    inA = 10;
13    inB = 20;
14    inC = 30;
15    inD = 40;
16
17    // Call the adder for 5 transactions
18    for (i=0; i<3; i++)
19    {
20        res = lab1_1(inA, inB, inC, inD);
21
22        fprintf(stdout, " %d*%d+%d+%d=%d \n", inA, inB, inC, inD, res);
23
24        // Test the output against expected results
25        if (res == refOut[i])
26            pass = 1;
27        else
28            pass = 0;
29
30        inA=inA+10;
31        inB=inB+10;
32        inC=inC+10;
33        inD=inD+10;
34    }
35
36    if (pass)
37    {
38        fprintf(stdout, "-----Pass!-----\n");
39        return 0;
40    }
41    else
42    {
43        fprintf(stderr, "-----Fail!-----\n");
44        return 1;
45    }
46 }
47
```

Рис. 1.2. Код теста

Создать 2 решения для синтеза устройства: задать clock period 6 и clock period 8, а также clock uncertain 0.1 в обоих случаях.

Для обоих решений осуществить моделирование и синтез, привести в данном 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

Также для обоих решений осуществить C|RTL моделирование:

- Отобразить два цикла обработки на одном экране
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval

Написать выводы, где среди прочего объяснить отличие двух решений.

2. Первое решение

2.1. Моделирование

При создании решения зададим настройки: clock period 6, clock uncertain 0.1, part xa7a12tcsg325-1q.

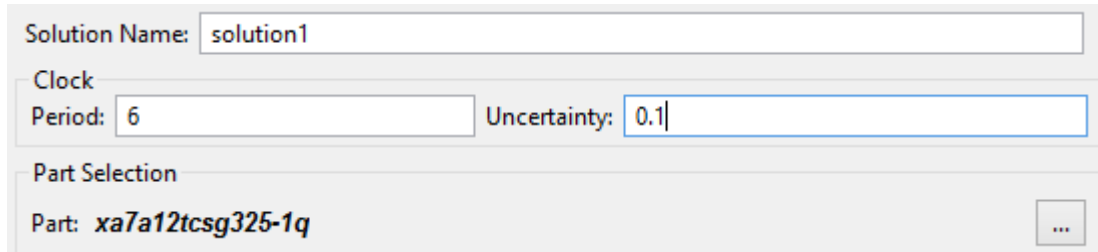


Рис. 2.1. Создание первого решения

При запуске моделирования можно увидеть, что тест успешно пройден:

```
Console Errors Warnings DRCs
Vivado HLS Console
Starting C simulation ...
C:/Xilinx/Vivado/2019.1/bin/vivado_hls.bat C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab1_1/solution2/csim.tcl
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/vivado_hls.exe'
INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT amd64 version 6.1) on Sun Oct 06 17:00:42 +0300 2019
INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado'
Sourcing Tcl script 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab1_1/solution2/csim.tcl'
INFO: [HLS 200-10] Opening project 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab1_1'.
INFO: [HLS 200-10] Opening solution 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab1_1/solution2'.
INFO: [SYN 201-201] Setting up clock 'default' with a period of 6ns.
INFO: [SYN 201-201] Setting up clock 'default' with an uncertainty of 0.1ns.
WARNING: [HLS 200-40] Cannot find library 'C:/Xilinx/Vivado/2019.1/common/technology/xilinx/aartix7/aartix7.lib'.
WARNING: [HLS 200-40] Cannot find library 'xilinx/aartix7/aartix7'.
INFO: [HLS 200-10] Setting target device to 'xa7a12t-csg325-1Q'
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
Compiling(apcc) ../../../../../../../../../../Spbgpu/lab_z1/source/lab1_1.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT amd64 version 6.1) on Sun Oct 06 17:00:48 +0300 2019
INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab1_1/solution2/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
10*20+30+40=270
20*30+40+50=690
30*40+50+60=1310
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис. 2.2. Логи моделирования

2.2. Синтез

Приведем в отчете требуемые данные о проекте:

| Performance Estimates | | | |
|-----------------------|--------|-----------|-------------|
| Timing (ns) | | | |
| Summary | | | |
| Clock | Target | Estimated | Uncertainty |
| ap_clk | 6.00 | 3.820 | 0.10 |

Рис. 2.3. Производительность

Здесь можно увидеть, что достигнутая задержка равна $3.820 + 0.1$, что укладывается в заданные нами требования к тактовой частоте.

| Utilization Estimates | | | | | |
|-----------------------|----------|--------|-------|------|------|
| Summary | | | | | |
| Name | BRAM_18K | DSP48E | FF | LUT | URAM |
| DSP | - | 1 | - | - | - |
| Expression | - | - | 0 | 16 | - |
| FIFO | - | - | - | - | - |
| Instance | - | - | - | - | - |
| Memory | - | - | - | - | - |
| Multiplexer | - | - | - | 21 | - |
| Register | - | - | 12 | - | - |
| Total | 0 | 1 | 12 | 37 | 0 |
| Available | 40 | 40 | 16000 | 8000 | 0 |
| Utilization (%) | 0 | 2 | ~0 | ~0 | 0 |
| Detail | | | | | |

Рис. 2.4. Занимаемые ресурсы

Данный проект займет на микросхеме 1 DSP блок (в котором будут использованы и сумматоры и умножитель), 12 триггеров для хранения чисел, и 37 LUT.

| Performance Profile | | Resource Profile | | | |
|---------------------|-----------|------------------|-------------------|---------------------|------------|
| | Pipelined | Latency | Iteration Latency | Initiation Interval | Trip count |
| lab1_1 | - | 2 | - | 3 | - |

Рис. 2.5. Профиль производительности

На этом рисунке видно, что задержка получения выходного значения составляет 2 такта с момента старта, а задержка после старта до готовности приема новых данных – 3. Покажем эти интервалы на временной диаграмме:

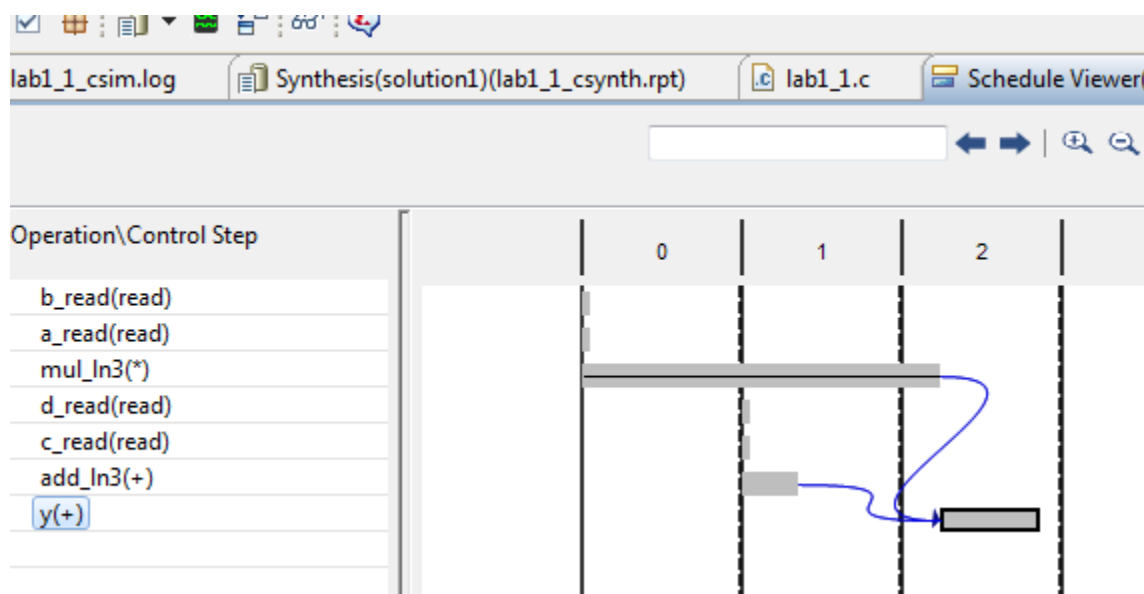


Рис. 2.6. Временная диаграмма

Здесь мы видим весь процесс получения результата. На первом такте происходит считывание операторов A и B, а также начинается их умножение. На втором считывание C и D, а также их сложение. На третьем результаты умножения и сложения складываются и получается ответ ($\text{latency} = 2$), а со следующего 4-го такта можно подавать следующие данные ($\Pi = 3$).

Наконец покажем профиль ресурсов:

| | BRAM | DSP | FF | LUT | Bits P0 | Bits P1 | Bits P2 | Banks/Depth | Words | W*Bits*Banks |
|-----------------|------|-----|----|-----|---------|---------|---------|-------------|-------|--------------|
| lab1_1 | 0 | 1 | 12 | 37 | | | | | | |
| I/O Ports(4) | | | | | 32 | | | | | |
| Instances(0) | 0 | 0 | 0 | 0 | | | | | | |
| Memories(0) | 0 | | 0 | 0 | 0 | | 0 | | 0 | 0 |
| Expressions(1) | 0 | 0 | 0 | 16 | 9 | 9 | 0 | | | |
| Registers(2) | | | 12 | | 12 | | | | | |
| Channels(0) | 0 | | 0 | 0 | 0 | | 0 | | 0 | 0 |
| Multiplexers(1) | 0 | | 0 | 21 | 1 | | 0 | | | |
| DSP(1) | | 1 | | | | | | | | |

Рис. 2.7. Профиль ресурсов

Здесь можно увидеть те же числа, что и в отчете синтезатора.

2.3. C/RTL моделирование

При совместном моделировании, программа отобразила те же самые, ожидаемые нами значения Latency и Π :

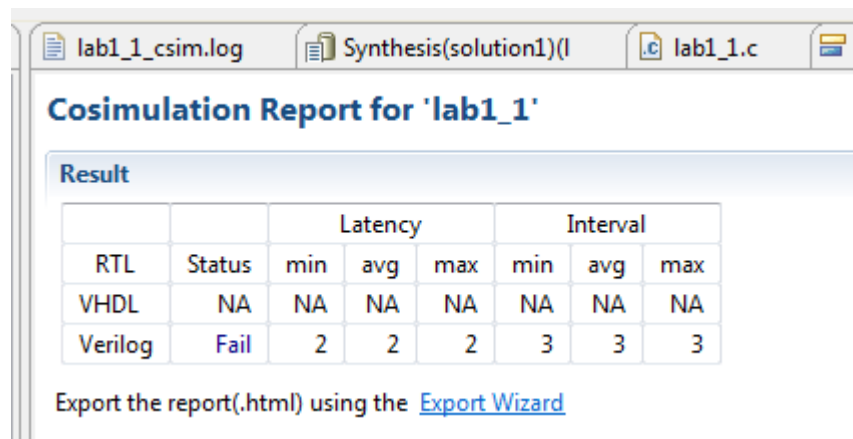


Рис.2.8. Результаты C\RTL моделирования

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

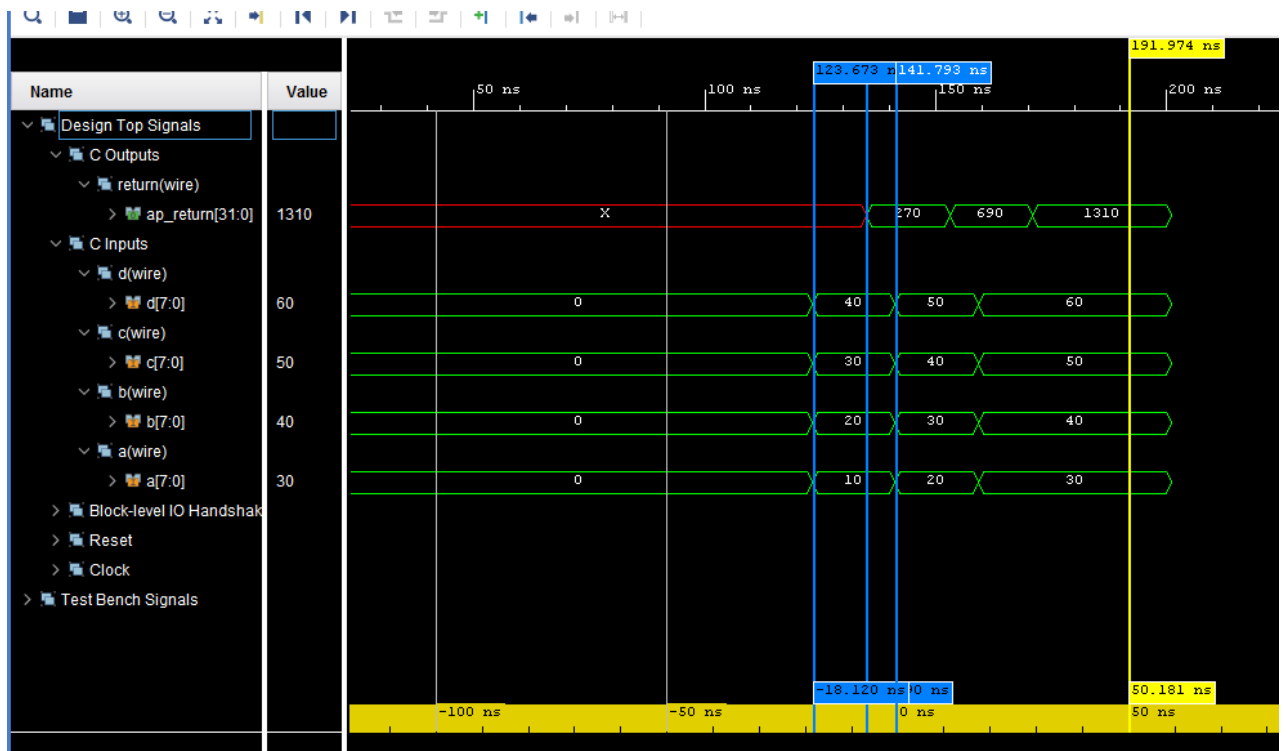


Рис. 2.9. Временная диаграмма совместного моделирования

3. Второе решение

3.1. Моделирование

Создадим второе решение для данного проекта. Его настройки:

Solution Name:

Clock
 Period: Uncertainty:

Part Selection
 Part:

Рис. 3.1. Настройки нового решения

Настройки этого решения изменены согласно заданию.

```

Vivado HLS Console
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
Compiling(apcc) ../../../../../../../../../../Spbgpu/lab_z1/source/lab1_1.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT_amd64 version 6.1) on Sun Oct 06 21:02:33 +0300 2019
INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab1_1/solution2/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
10*20+30+40=270
20*30+40+50=690
30*40+50+60=1310
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
  
```

Рис. 3.2. Логи моделирования

Моделирования второго решения также прошло успешно.

3.2. Синтез

| General Information | | | |
|---------------------|--|--|--|
| Date: | Sun Oct 6 21:02:56 2019 | | |
| Version: | 2019.1 (Build 2552052 on Fri May 24 15:28:33 MDT 2019) | | |
| Project: | lab1_1 | | |
| Solution: | solution2 | | |
| Product family: | artix7 | | |
| Target device: | xa7a12t-csq325-1Q | | |

| Performance Estimates | | | |
|-----------------------|--------|-----------|-------------|
| Timing (ns) | | | |
| Summary | | | |
| Clock | Target | Estimated | Uncertainty |
| ap_clk | 8.00 | 7.180 | 0.10 |

Рис. 3.3. Производительность

На этом рисунке видно, что данное решение также укладывается в установленный период тактовой частоты, однако задержки сильно отличаются, что

означает, что изменения периода тактовой частоты повлекло за собой существенные изменения устройства.

На примере затрачиваемых на решение ресурсов это также хорошо видно:

| Utilization Estimates | | | | | |
|-----------------------|----------|--------|-------|------|------|
| Summary | | | | | |
| Name | BRAM_18K | DSP48E | FF | LUT | URAM |
| DSP | - | 1 | - | - | - |
| Expression | - | - | 0 | 16 | - |
| FIFO | - | - | - | - | - |
| Instance | - | - | - | - | - |
| Memory | - | - | - | - | - |
| Multiplexer | - | - | - | - | - |
| Register | - | - | - | - | - |
| Total | 0 | 1 | 0 | 16 | 0 |
| Available | 40 | 40 | 16000 | 8000 | 0 |
| Utilization (%) | 0 | 2 | 0 | ~0 | 0 |

Рис. 3.4. Затрачиваемые ресурсы

В сравнении с предыдущим решением схема избавилась от всех триггеров и большинства LUT.

| Performance Profile | | | | | |
|---------------------|-----------|---------|-------------------|---------------------|------------|
| Resource Profile | | | | | |
| | Pipelined | Latency | Iteration Latency | Initiation Interval | Trip count |
| lab1_1 | - | 0 | - | 1 | - |

Рис. 3.5. Профиль производительности

Сразу видно, что задержка до получения ответа 0 тактов, а новые данные можно подавать уже через 1 такт. Покажем это на временной диаграмме:

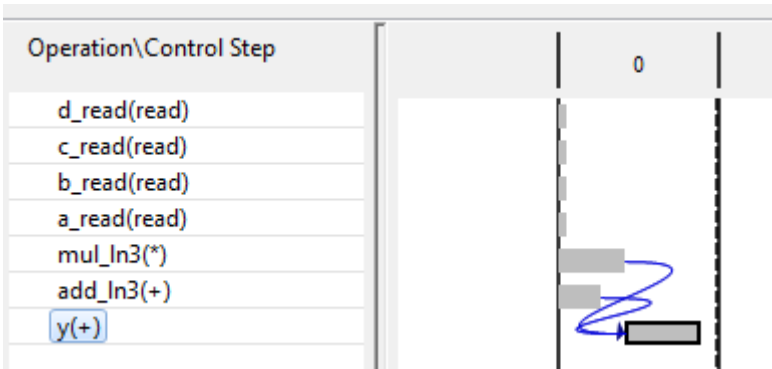


Рис. 3.6. Временная диаграмма

Если изучить эту диаграмму подробнее, то можно сказать, что считка всех 4-х операторов происходит одновременно, после чего сразу же начинаются умножение и

первое сложение. Второе сложение начинается сразу после завершения умножения. Такой алгоритм позволяет получить ответ уже до начала второго такта.

Также можно сказать, что применен другой алгоритм умножения, который не требует задержки в 1 такт, и вообще весь процесс получения ответа является комбинаторным, что и объясняет нулевое значение параметра latency.

| | BRAM | DSP | FF | LUT | Bits P0 | Bits P1 | Bits P2 | Banks/Depth | Words | W*Bits*Banks |
|-----------------|------|-----|----|-----|---------|---------|---------|-------------|-------|--------------|
| lab1_1 | 0 | 1 | 0 | 16 | 32 | | | | | |
| I/O Ports(4) | | | | | 32 | | | | | |
| Instances(0) | 0 | 0 | 0 | 0 | | | | | | |
| Memories(0) | 0 | | 0 | 0 | 0 | | | 0 | 0 | 0 |
| Expressions(1) | 0 | 0 | 0 | 16 | 9 | 9 | 0 | | | |
| Registers(0) | | | 0 | 0 | 0 | | | | | |
| Channels(0) | 0 | | 0 | 0 | 0 | | | 0 | 0 | 0 |
| Multiplexers(0) | 0 | | 0 | 0 | 0 | | | 0 | | |
| DSP(1) | | 1 | | | | | | | | |

Рис. 3.7. Профиль ресурсов

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

3.3. C\RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency, но II также стал нулевым:

| | | Latency | | | Interval | | |
|---------|--------|---------|-----|-----|----------|-----|-----|
| RTL | Status | min | avg | max | min | avg | max |
| VHDL | NA | NA | NA | NA | NA | NA | NA |
| Verilog | Pass | 0 | 0 | 0 | 0 | 0 | 0 |

Export the report(.html) using the [Export Wizard](#)

Рис. 3.8. C\RTL моделирование

Моделирование прошло успешно, однако программа предупредила, что оба значения выставлены в ноль, поскольку моделируемая схема – комбинаторная.

Покажем временную диаграмму совместного моделирования:

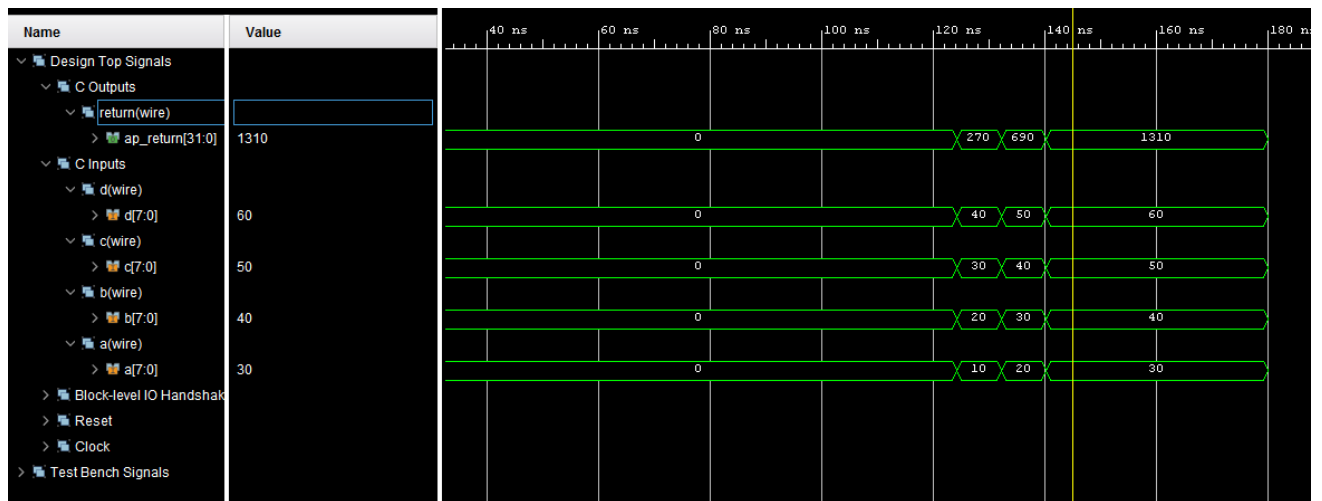


Рис. 3.9. Временная диаграмма совместного моделирования

Здесь видно отсутствия тактового сигнала и мгновенное (без учета задержек) получение ответа.

4. Выводы

В данной работе главным образом были рассмотрены различия, которые могут появиться при синтезе устройства с одним незначительным изменением периода тактовой частоты. В данном примере видно, что при большем периоде тактов, программа сумела вместить в 1 период чисто комбинаторную схему, в то время как при меньшем периоде это оказалось невозможным. Таким образом, получены 2 решения: первое – полный цикл выполнения 3 такта, а максимальная задержка обработки сигнала на такте составляет 3.820 нс, и второе – чисто комбинаторное решение, задержка в котором уже 7.180 нс, но все выполняется всего за 1 период тактового сигнала.