

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 2

Студент: Ерниязов Т.Е.  
Гр. № 3540901/81501  
Преподаватель: Антонов А.П.

Санкт-Петербург  
2019

## Оглавление

Задание .....	3
Ход работы.....	4
Решение 1 .....	4
Моделирование.....	5
Синтез.....	6
C/RTL моделирование .....	8
Решение 2 .....	9
Моделирование.....	9
Синтез.....	10
C/RTL моделирование .....	12
Выводы .....	13

## Задание

Создать проект lab1\_2

Подключить файл lab1\_2.c (папка source)

Написать тест lab1\_2\_test.c по аналогии с тестом 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
- 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 10; 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\_2.
2. Подключение файлов lab1\_2.c, lab1\_2\_test.c.

#### Lab1\_2.c

```
void lab1_2 (int in[3], char a, char b, char c, int out[3]) {
    int x,y;
    for(int i = 0; i < 3; i++) {
        x = in[i];
        y = a*x + b + c;
        out[i] = y;
    }
}
```

#### Lab1\_2\_test.c

```
#include <stdio.h>

int main()
{
    int inA, inB, inC;
    int out[3];
    // For adders
    int refOut[3][3] = {{6,7,8}, {12,14,16}, {24,28,32}};
    int pass;
    int i;

    inA = 1;
    inB = 2;
    inC = 3;
    int in[3] = {1, 2, 3};
    int res[3];

    // Call the adder for 5 transactions
    for (i=0; i<3; i++)
    {
        lab1_2(in, inA, inB, inC, out);

        for (int j=0; j<3; j++)
        {
            fprintf(stdout, "  %d*1+%d+%d=%d \n", inA, inB, inC, out[j]);
        }

        // Test the output against expected results
        if (equals(out, refOut[i]) == 1)
```

```

        pass = 1;
    else
        pass = 0;

    inA=inA*2;
    inB=inB*2;
    inC=inC*2;

}

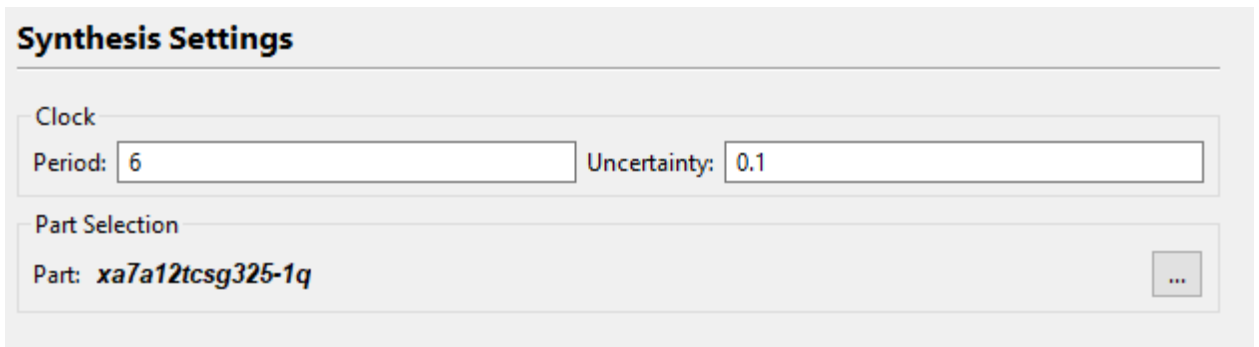
if (pass)
{
    fprintf(stdout, "-----Pass!-----\n");
    return 0;
}
else
{
    fprintf(stderr, "-----Fail!-----\n");
    return 1;
}
}

int equals(int a[], int b[])
{
    int res = 1;

for (int i = 0; i<sizeof(a)/sizeof(int); i++)
    {
        if (a[i] == b[i])
        {
            res = 1;
        }
        else return 0;
    }
    return res;
}

```

### 3. Конфигурирование решения.



**Synthesis Settings**

**Clock**

Period:  Uncertainty:

**Part Selection**

Part: **xa7a12tcsg325-1q**

## Моделирование

### 4. Результат моделирования заданного решения.

```

1 warning generated.
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Compiling(apcc) ../../../../source/lab1_2.c in debug mode
INFO: [HLS 200-10] Running 'D:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'loris' on host 'laptop-34slcvbc' (windows NT_amd64 version 6.2) on Thu Dec 12 02:59:10 +0300 2019
INFO: [HLS 200-10] In directory 'D:/Antonov/lab1_z2/lab1_2/solution1/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
1*1+2+3=6
1*1+2+3=7
1*1+2+3=8
2*1+4+6=12
2*1+4+6=14
2*1+4+6=16
4*1+8+12=24
4*1+8+12=28
4*1+8+12=32
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.

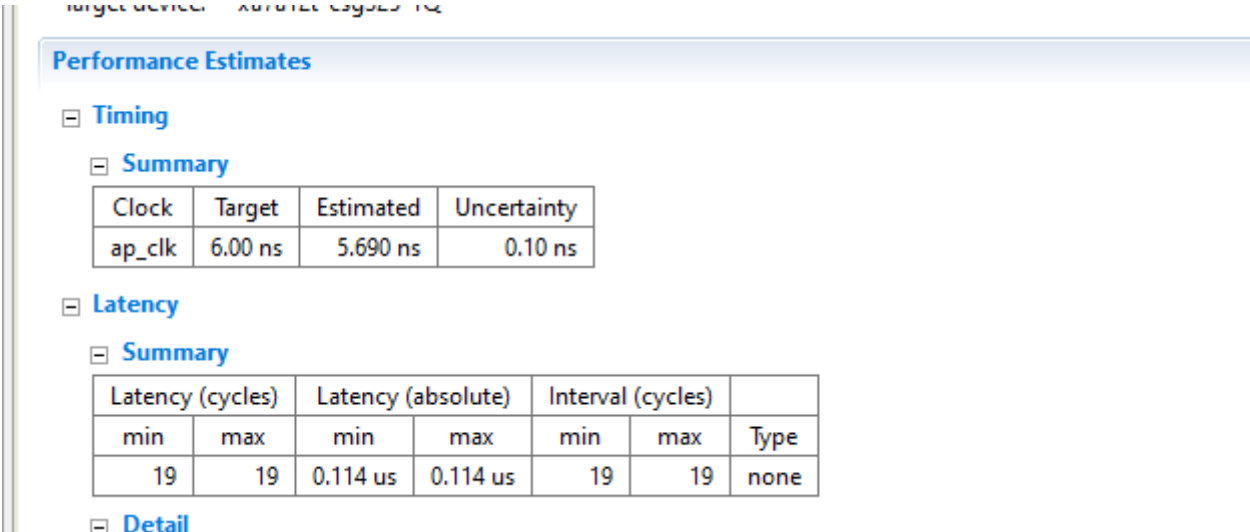
```

Моделирование выполнено без ошибок. Тест пройден успешно.

## Синтез

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

Производительность



**Performance Estimates**

☒ **Timing**

☒ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	6.00 ns	5.690 ns	0.10 ns

☒ **Latency**

☒ **Summary**

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
19	19	0.114 us	0.114 us	19	19	none

☒ **Detail**

Достигнутая задержка (estimated) равна 5,690 + (погрешность - uncertainty) 0,10. Величина задержки укладывается в заданные требования к тактовой частоте, но является значительной для заданных ограничений.

Занимаемые ресурсы

### Utilization Estimates

#### Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	73	-
FIFO	-	-	-	-	-
Instance	-	2	166	49	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	50	-
Register	-	-	141	-	-
Total	0	2	307	172	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	1	2	0

Данный проект будет занимать на микросхеме:

#### 4 DSP блока.

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.

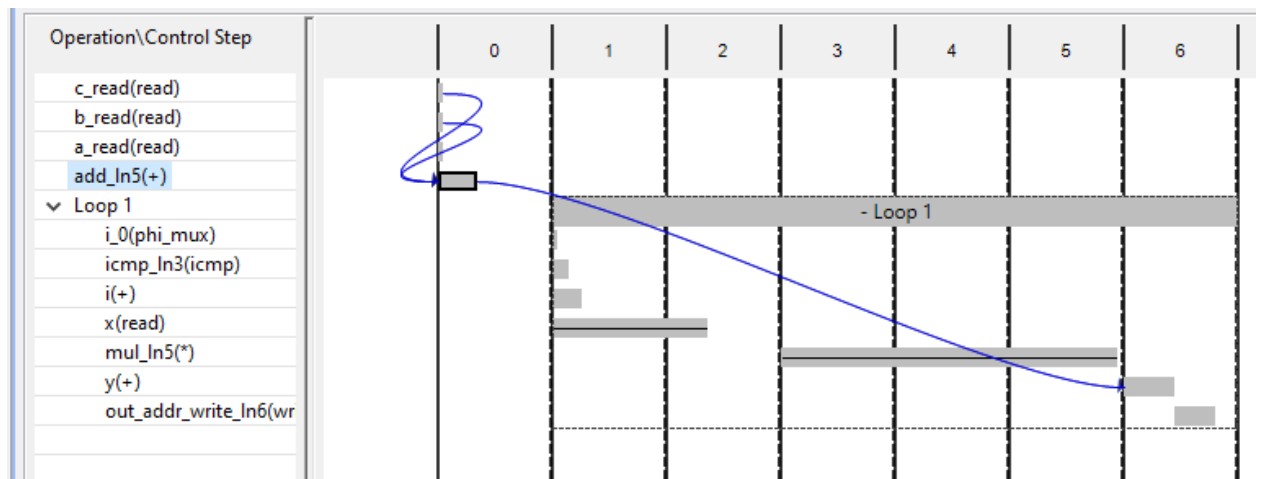
307 регистров для хранения и считывания данных (чисел).

172 LUT

6. Перейдем на вкладку Analysis.

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab1_2	-	19	-	20	-
Loop 1	no	18	6	-	3

Задержка времени со старта до момента получения результата составляет 19 тактов. Задержка времени до готовности получения новых данных – 20 тактов. Временная диаграмма:



Получение результата происходит следующим образом:

- Первый такт
  - Считывание параметра c
  - Считывание параметра b
  - Считывание параметра a
  - Инициализация параметра
- Второй – седьмой такты
  - Выполнение действий в цикле

## 7. Профиль ресурсов

Performance Profile		Resource Profile								
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits^Banks
lab1_2	0	2	307	172						
> I/O Ports(5)					88					
> Instances(1)	0	2	166	49						
> Memories(0)	0		0	0	0			0	0	0
> Expressions(4)	0	0	0	73	45	44	0			
> Registers(8)			141		203					
> Channels(0)	0		0	0	0			0	0	0
> Multiplexers(2)	0		0	50	3			0		
> DSP(1)		0								

Значения в отчете по ресурсам идентичные значениям в пункте 5.

## C/RTL моделирование

### 8. Результат выполнения.



```

Compiling module work.g004
Built simulation snapshot lab1_2

***** Webtalk v2019.2 (64-bit)
**** SW Build 2708876 on Wed Nov  6 21:40:23 MST 2019
**** IP Build 2700528 on Thu Nov  7 00:09:20 MST 2019
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

source D:/Antonov/lab1_2/lab1_2/lab1_2/solution1/sim/verilog/xsim.dir/lab1_2/webtalk/xsim_webtalk.tcl -notrace
INFO: [Common 17-186] 'D:/Antonov/lab1_2/lab1_2/lab1_2/solution1/sim/verilog/xsim.dir/lab1_2/webtalk/usage_statistics_ext_xsim.xml' has been successfully
INFO: [Common 17-206] Exiting Webtalk at Thu Dec 12 03:02:19 2019...

***** xsim v2019.2 (64-bit)
**** SW Build 2708876 on Wed Nov  6 21:40:23 MST 2019
**** IP Build 2700528 on Thu Nov  7 00:09:20 MST 2019
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

start_gui
INFO: [Common 17-206] Exiting xsim at Thu Dec 12 03:04:45 2019...
INFO: [COSIM 212-316] Starting C post checking ...
1*1+2+3=6
1*1+2+3=7
1*1+2+3=8
2*1+4+6=12
2*1+4+6=14
2*1+4+6=16
4*1+8+12=24
4*1+8+12=28
4*1+8+12=32
-----Pass!-----
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
Finished C/RTL cosimulation.

```

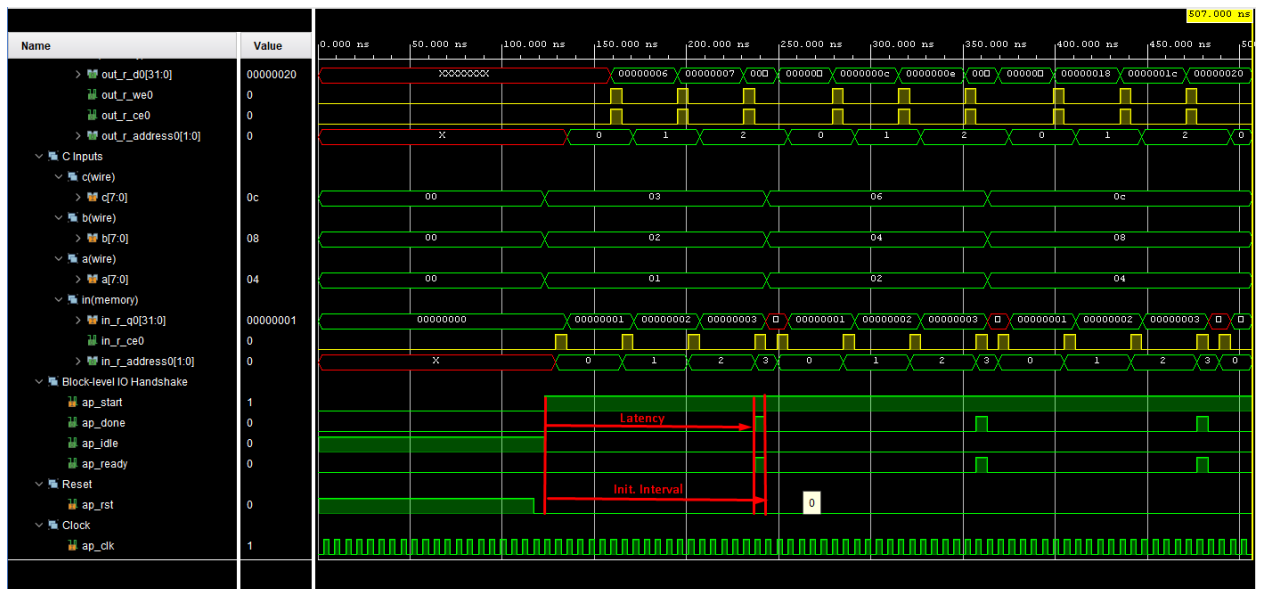
## 9. Отчет о выполнении

### Cosimulation Report for 'lab1\_2'

#### Result

RTL	Status	Latency			Interval		
		min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	19	19	19	20	20	20

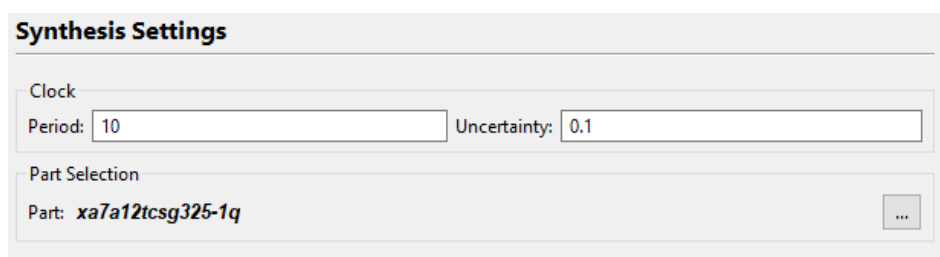
Результаты идентичны полученным в пункте 6.



## Решение 2

## Моделирование

## 1. Создание и конфигурирование решения.



**Synthesis Settings**

Clock

Period: 10      Uncertainty: 0.1

Part Selection

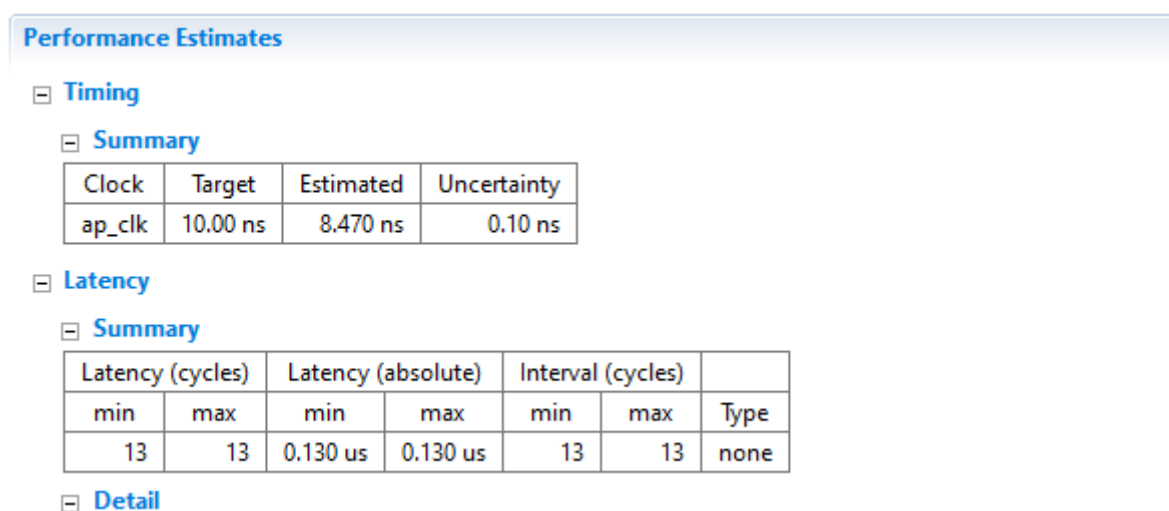
Part: xa7a12tcsq325-1q

Исходные файлы соответствуют файлам решения 1.

## Синтез

## 2. Выполним команду Solution-> Run C Synthesis -> Active solution

### Производительность



**Performance Estimates**

[-] **Timing**

[-] **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.470 ns	0.10 ns

[-] **Latency**

[-] **Summary**

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
13	13	0.130 us	0.130 us	13	13	none

[-] **Detail**

Отношение величины полученной задержки к заданному значению значительно отличаются по отношению к первому решению. Результаты свидетельствуют о том, что измененные параметры влекут за собой изменение устройства.

### Затрачиваемые ресурсы

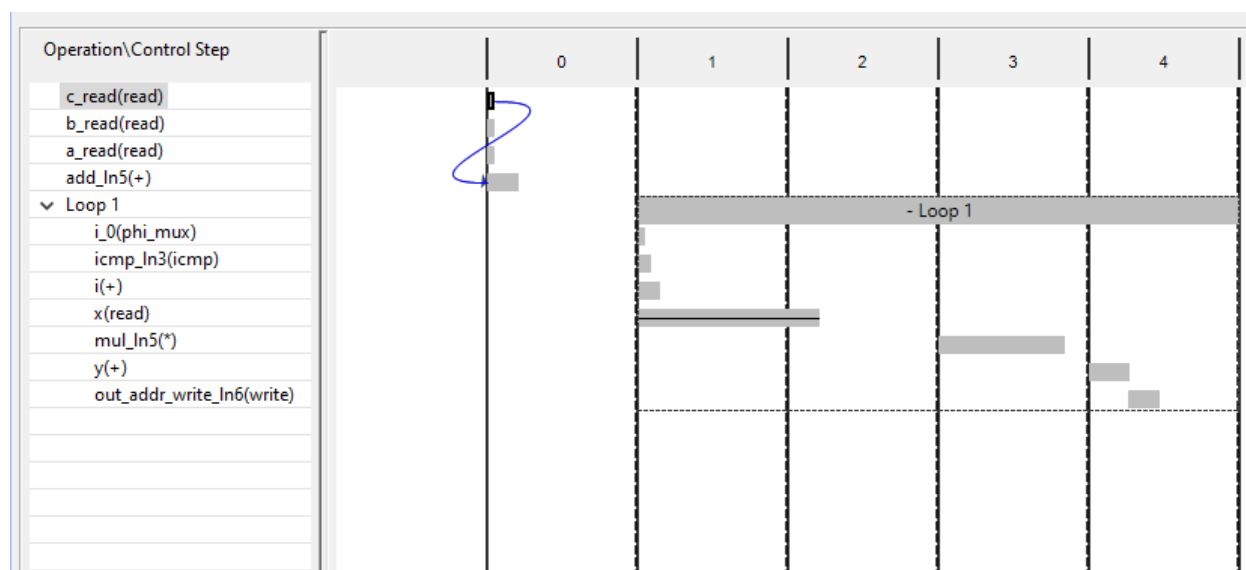
Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	2	0	94	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	42	-
Register	-	-	139	-	-
Total	0	2	139	136	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	1	0
Detail					

По сравнению с решением 1, использование ресурсов значительно меньше. Используется более чем в 2 раза меньше регистров и значительно меньше LUT.

3. Перейдем на вкладку Analysis.

Performance Profile					
Resource Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab1_2	-	13	-	14	-
Loop 1	no	12	4	-	3

На изображении видно, что до получения ответа требуется 13 тактов, а готовность получения новых данных наступает через 14 такт. Действия в цикле ограничиваются 4 тактами вместо 6.



## Общий отчет о затраченных ресурсах

Performance Profile		Resource Profile									
		BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits*Banks
lab1_2		0	2	139	136						
>	I/O Ports(5)					88					
	Instances(0)	0	0	0	0						
	Memories(0)	0		0	0	0			0	0	0
>	Expressions(5)	0	2	0	94	77	52	0			
>	Registers(8)			139		201					
	Channels(0)	0		0	0	0			0	0	0
>	Multiplexers(2)	0		0	42	3			0		
	DSP(0)		0								

## C/RTL моделирование

### 4. Результат выполнения

```
Compiling module lab1_2_top
Compiling module work.glbl
Built simulation snapshot lab1_2

***** Webtalk v2019.2 (64-bit)
**** SW Build 2708876 on Wed Nov  6 21:40:23 MST 2019
**** IP Build 2700528 on Thu Nov  7 00:09:20 MST 2019
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

source D:/Antonov/lab1_2/lab1_2/lab1_2/solution2/sim/verilog/xsim.dir/lab1_2/webtalk/xsim_webtalk.tcl -notrace
INFO: [Common 17-186] "D:/Antonov/lab1_2/lab1_2/lab1_2/solution2/sim/verilog/xsim.dir/lab1_2/webtalk/usage_statistics_ext_xsim.xml"
INFO: [Common 17-206] Exiting Webtalk at Thu Dec 12 03:08:04 2019...

***** xsim v2019.2 (64-bit)
**** SW Build 2708876 on Wed Nov  6 21:40:23 MST 2019
**** IP Build 2700528 on Thu Nov  7 00:09:20 MST 2019
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

start_gui
INFO: [Common 17-206] Exiting xsim at Thu Dec 12 03:10:21 2019...
INFO: [COSIM 212-316] Starting C post checking ...
1*1+2+3=6
1*1+2+3=7
1*1+2+3=8
2*1+4+6=12
2*1+4+6=14
2*1+4+6=16
4*1+8+12=24
4*1+8+12=28
4*1+8+12=32
-----Pass!-----
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
Finished C/RTL cosimulation.
```

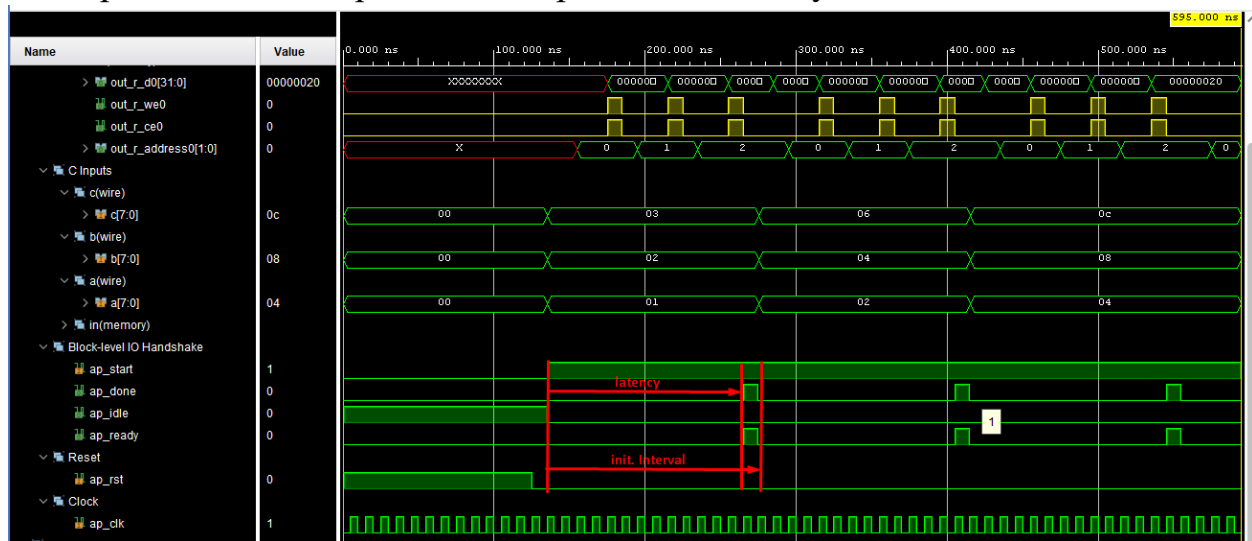
### 5. Отчет решения

## Cosimulation Report for 'lab1\_2'

### Result

		Latency			Interval		
		min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	13	13	13	14	14	14

На временной диаграмме изображены Latency, Initiation interval.



## **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 19 тактов, а максимальная задержка обработки сигнала на такте составляет 5.69 нс, и второе – решение, задержка в котором составила большую величину 8.47 нс, но при этом все действия выполняются за 13 тактов.