

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе №11_1
Курс: «Проектирование реконфигурируемых гибридных
вычислительных систем»
Тема: « Задержка (Latency) »

Выполнил студент гр. 3540901/81501

Селиверстов Я.А.

(подпись)

Руководитель

Антонов А.П.

(подпись)

“ ” _____ 2019 г.

Санкт – Петербург
2019

ОГЛАВЛЕНИЕ

1. Задание.....	3
2. Исходный код.....	6
3. Моделирование	8
4. Исследование	9
4.1. Решение 1а	9
4.1.2.Синтез решения 1а	9
4.1.3. C/RTL моделирование	12
4.2. Решение 2а	13
4.2.2.Синтез решения 2а	14
4.2.3. C/RTL моделирование	17
4.3. Решение 3а	18
4.3.2.Синтез решения 3а	18
4.3.3. C/RTL моделирование	21
4.4. Решение 4а	22
4.4.2.Синтез решения 3а	22
4.4.3. C/RTL моделирование	25
Вывод.....	26

1. Задание

- Создать проект lab11_1
- Микросхема: ха7a12tcsg325-1q
- Создать функцию, содержащую цикл по образцу

```
Add: for(int i = 0; i < N; i++) {  
    a[i] = b[i] + c[i];  
}
```

N=16

- Создать тест lab11_1_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
- Исследование:
- Solution_1a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ПО УМОЛЧАНИЮ
 - осуществить синтез для:
 - привести в отчете:
 - performance estimates=>summary (timing, latency)
 - utilization estimates=>summary
 - performance Profile
 - Resource 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
 - Выполнить cosimulation и привести временную диаграмму
- Solution_2a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию UNROLL (без опций)
 - осуществить синтез
 - привести в отчете:
 - performance estimates=>summary (timing, latency)
 - utilization estimates=>summary

- performance Profile
 - Resource 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
- Выполнить cosimulation и привести временную диаграмму
- Сравнить два решения (solution_1a и solution_2a) и сделать выводы: зависимость от UNROLL; объяснить (посчитать) число циклов Latency, П...
- Solution_3a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию UNROLL (factor 4 + **exit check**)
 - осуществить синтез
 - привести в отчете:
 - performance estimates=>summary (timing, latency)
 - utilization estimates=>summary
 - performance Profile
 - Resource 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
 - Выполнить cosimulation и привести временную диаграмму
- Сравнить два решения (solution_2a и solution_3a) и сделать выводы: зависимость от Unroll (factor 4 + **exit check**) ; объяснить (посчитать) число циклов Latency, П...
- Solution_4a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию UNROLL (factor 4 без **exit check**)
 - осуществить синтез

- привести в отчете:
 - performance estimates=>summary (timing, latency)
 - utilization estimates=>summary
 - performance Profile
 - Resource 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
 - Выполнить cosimulation и привести временную диаграмму
- Сравнить два решения (solution_3a и solution_4a) и сделать выводы: зависимость от Unroll (factor 4 без *exit check*) ; объяснить (посчитать) число циклов Latency, II

2. Исходный код

Зададим следующий код устройства:

```
#include "lab11_1.h"
void lab11_1(int a[N], int b[N], int c[N])
{
    int i;
    Add:for (i = 0; i < N; i++)
    {
        a[i] = b[i] + c[i];
    }
}
```

Заголовочный файл определим как:

```
#define N 16
```

Код теста для проверки функции имеет вид:

```
#include <stdio.h>
#include "lab11_1.h"

int main()
{
    int a_actual[N], a_expected[N], b[N], c[N];
    int passed = 1;
    int i;
    for(i = 0; i < N; i++)
    {
        b[i] = i;
        c[i] = i * i * 3;
        a_expected[i] = b[i] + c[i];
    }

    lab11_1(a_actual, b, c);

    for(i = 0; i < N; i++)
    {
        printf("Expected_[%d]_actual_[%d]\n", a_expected[i], a_actual[i]);
        if(a_expected[i] != a_actual[i])
        {
            passed = 0;
        }
    }

    if(passed != 1) {
        printf("————Test_failed————\n");
    } else {
        printf("————Test_passed————\n");
    }
}
```

Скрипт для запуска программы с консоли имеет вид:

```
open_project -reset lab11_1

add_files lab11_1.c
add_files -tb lab11_1_test.c
set_top lab11_1

set solutions [list 1a 2a 3a 4a]

foreach sol $solutions {
    open_solution solution_$sol -reset
    set_part {xa7a12tcsg325-1q}
    create_clock -period 10ns
    set_clock_uncertainty 0.1

    if {$sol == "2a"} {
        set_directive_unroll "lab11_1/Add"
    }
    if {$sol == "3a"} {
        set_directive_unroll -factor 4 "lab11_1/Add"
    }
    if {$sol == "4a"} {
        set_directive_unroll -factor 4 -skip_exit_check "lab11_1/Add"
    }

    csim_design
    csynth_design
    cosim_design -trace_level all
}

exit
```

3. Моделирование

Результаты моделирования, подтверждающие корректность работы устройства, имеют вид:

```
INFO: [APCC 202-1] APCC is done.  
    Generating csim.exe  
Expected [0] actual [0]  
Expected [4] actual [4]  
Expected [14] actual [14]  
Expected [30] actual [30]  
Expected [52] actual [52]  
Expected [80] actual [80]  
Expected [114] actual [114]  
Expected [154] actual [154]  
Expected [200] actual [200]  
Expected [252] actual [252]  
Expected [310] actual [310]  
Expected [374] actual [374]  
Expected [444] actual [444]  
Expected [520] actual [520]  
Expected [602] actual [602]  
Expected [690] actual [690]  
-----Test passed-----  
INFO: [SIM 211-1] CSim done with 0 errors.  
INFO: [SIM 211-3] ***** CSIM finish *****
```


4. Исследование

4.1. Решение 1a

В соответствие с планом лабораторной работы устанавливаем:

- clock period = 10;
- clock uncertainty = 0.1;
- реализация ПО УМОЛЧАНИЮ

Директивы данного решения имеют вид:



4.1.2. Синтез решения 1a

Результаты оценки производительности имеют вид:

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.006	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
33	33	33	33	none

По данным результатам синтеза можно сделать вывод, что оценка производительности видно устройства соответствует заданным критериям.

Оценка использования имеет вид:

Utilization Estimates				
Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	65
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	30
Register	-	-	18	-
Total	0	0	18	95
Available	40	40	16000	8000
Utilization (%)	0	0	~0	1

Профиль производительности имеет вид:

Performance Profile					
Resource Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab11_1	-	33	-	34	-
Add	no	32	2	-	16

Данные планировщика просмотра имеет вид:

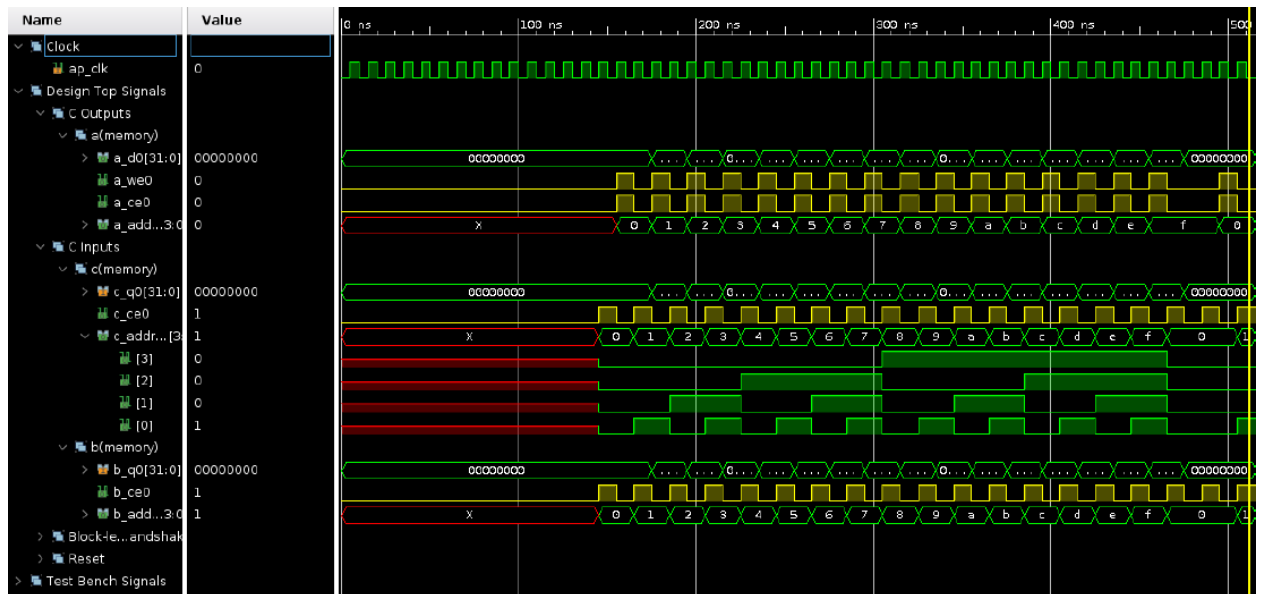
Operation\Control Step	0	1	2
Add		- Add	
i(phi_mux)			
exitcond(icmp)			
i_1(+)			
b_load(read)			
c_load(read)			
tmp_1(+)			
node_24(write)			

Данные обзора ресурсов имеет вид:

	Resource\Control Step	C0	C1	C2
1	[-] I/O Ports			
2	c(p0)		read	
3	b(p0)		read	
4	a(p0)			write
5	[-] Memory Ports			
6	c(p0)		read	
7	b(p0)		read	
8	a(p0)			write
9	[-] Expressions			
10	i_l_fu_84		+	
11	i_phi_fu_71		phi_mux	
12	exitcond_fu_78		icmp	
13	tmp_1_fu_96			+

4.1.3. C/RTL моделирование

Временная диаграмма результатов C/RTL моделирования имеет вид:



По результатам временной диаграммы можно сделать вывод, что выполнение одного цикла требует 2 такта (всего 16 циклов) и 1 такт для инициализации, таким образом, $Latency = 16 * 2 + 1 = 33$. По умолчанию интерфейсы реализованы как ap_memory.

4.2. Решение 2a

В соответствие с планом лабораторной работы устанавливаем:

- clock period 10;
- clock_uncertainty 0.1
- установить реализацию UNROLL (без опций)

Директивы данного решения имеют вид:

```
▼ ● lab11_1
    ● a
    ● b
    ● c
    ▼  $x+y=?$  Add
        % HLS UNROLL
```

4.2.2.Синтез решения 2а

Результаты оценки производительности имеют вид:

Performance Estimates

▣ Timing (ns)

▣ Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.006	0.10

▣ Latency (clock cycles)

▣ Summary

Latency		Interval		
min	max	min	max	Type
8	8	8	8	none

По данным результатам синтеза можно сделать вывод, что оценка производительности видно устройства соответствует заданным критериям.

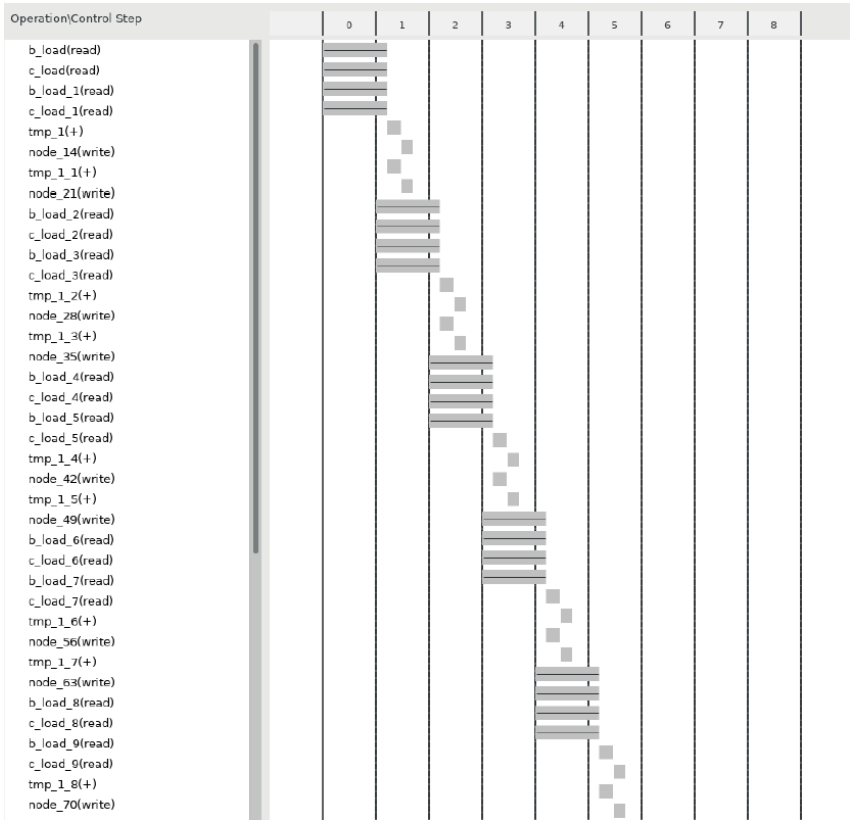
Оценка использования имеет вид:

Utilization Estimates				
Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	78
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	311
Register	-	-	9	-
Total	0	0	9	389
Available	40	4016000	8000	
Utilization (%)	0	0	~0	4

Профиль производительности имеет вид:

Performance Profile		Resource Profile				
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count	
lab11_1	-	8	-	9	-	

Данные планировщика просмотра имеет вид:

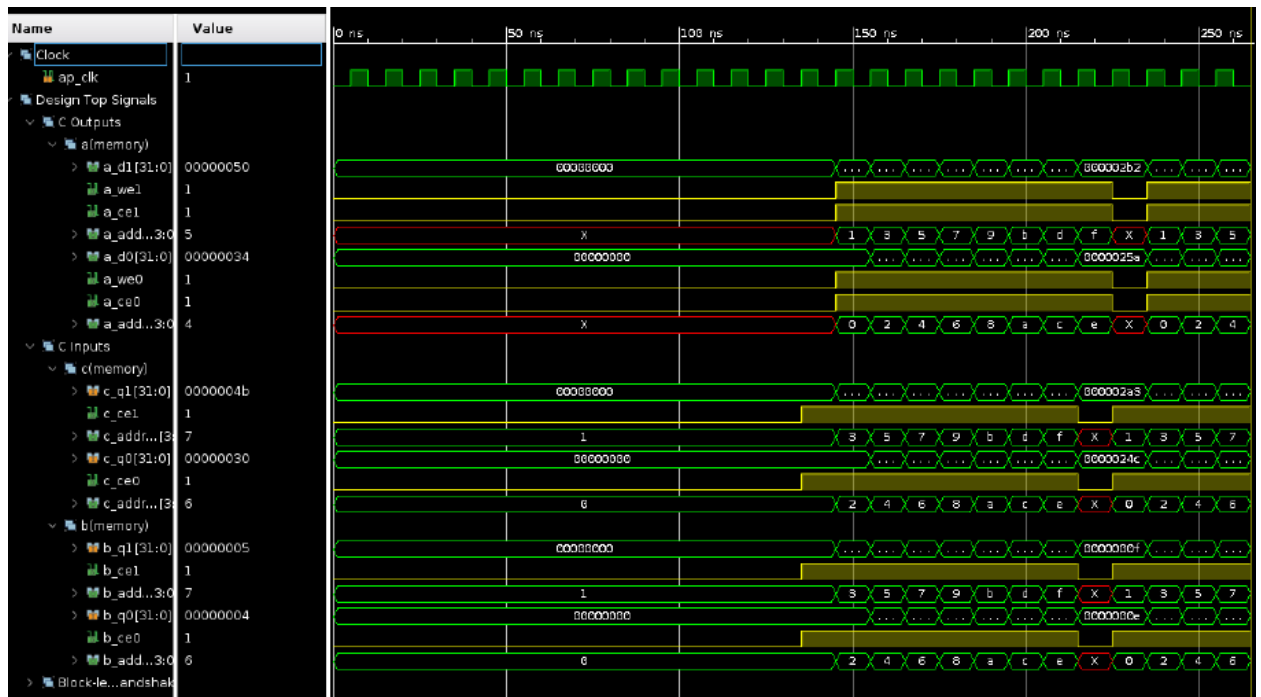


Данные обзора ресурсов имеет вид:

	Resource\Control Step	C0	C1	C2	C3	C4	C5	C6	C7	C8
1	⊞ I/O Ports									
2	b(p1)	read	read	read	read	read	read	read	read	
3	b(p0)	read	read	read	read	read	read	read	read	
4	c(p0)	read	read	read	read	read	read	read	read	
5	c(p1)	read	read	read	read	read	read	read	read	
6	a(p0)		write	write	write	write	write	write	write	write
7	a(p1)		write	write	write	write	write	write	write	write
8	⊞ Memory Ports									
9	c(p1)	read	read	read	read	read	read	read	read	
10	c(p0)	read	read	read	read	read	read	read	read	
11	b(p1)	read	read	read	read	read	read	read	read	
12	b(p0)	read	read	read	read	read	read	read	read	
13	a(p1)		write	write	write	write	write	write	write	write
14	a(p0)		write	write	write	write	write	write	write	write
15	⊞ Expressions									
16	grp_fu_503		+	+	+	+	+	+	+	+
17	grp_fu_510		+	+	+	+	+	+	+	+

4.2.3. C/RTL моделирование

Временная диаграмма результатов C/RTL моделирования имеет вид:



При применении директивы UNROLL уменьшилось значение Задержки с 33 до 8, потому, что в предыдущем решении в проекте применялся один блок памяти для каждого массива, а в этом для каждого массива используется 2 блока памяти, что позволяет читать одновременно по 2 значения. Так как в цикле 16 итераций и читается по 2 значения одновременно, количество циклов уменьшается в 2 раза. Так как мы «развернули» оставшиеся 8 циклов в один каскад последовательных чтений/записи то результирующее значения Задержки стало 8.

4.3. Решение 3а

В соответствие с планом лабораторной работы устанавливаем:

- задать: clock period 10; clock_uncertainty 0.1
- установить реализацию UNROLL (factor 4 + *exit check*)

Директивы данного решения имеют вид:

```
▼ ● lab11_1
    ● a
    ● b
    ● c
    ▼ ⚙ Add
        % HLS UNROLL factor=4
```

4.3.2. Синтез решения 3а

Результаты оценки производительности имеют вид:

Performance Estimates

▣ Timing (ns)

▣ Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.006	0.10

▣ Latency (clock cycles)

▣ Summary

Latency		Interval		Type
min	max	min	max	
13	13	13	13	none

По данным результатам синтеза можно сделать вывод, что оценка производительности видно устройства соответствует заданным критериям.

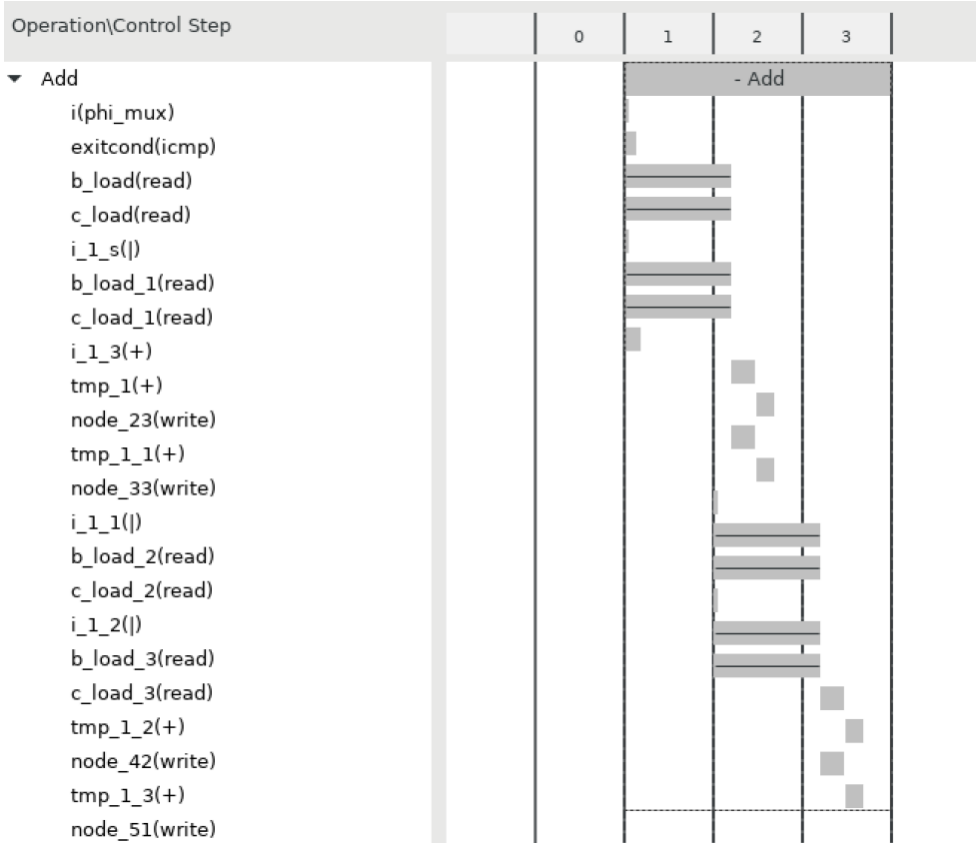
Оценка использования имеет вид:

Utilization Estimates				
Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	137
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	126
Register	-	-	31	-
Total	0	0	31	263
Available	40	4016000	8000	
Utilization (%)	0	0	~0	3

Профиль производительности имеет вид:

Performance Profile					
Resource Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab11_1	-	13	-	14	-
Add	no	12	3	-	4

Данные планировщика просмотра имеет вид:

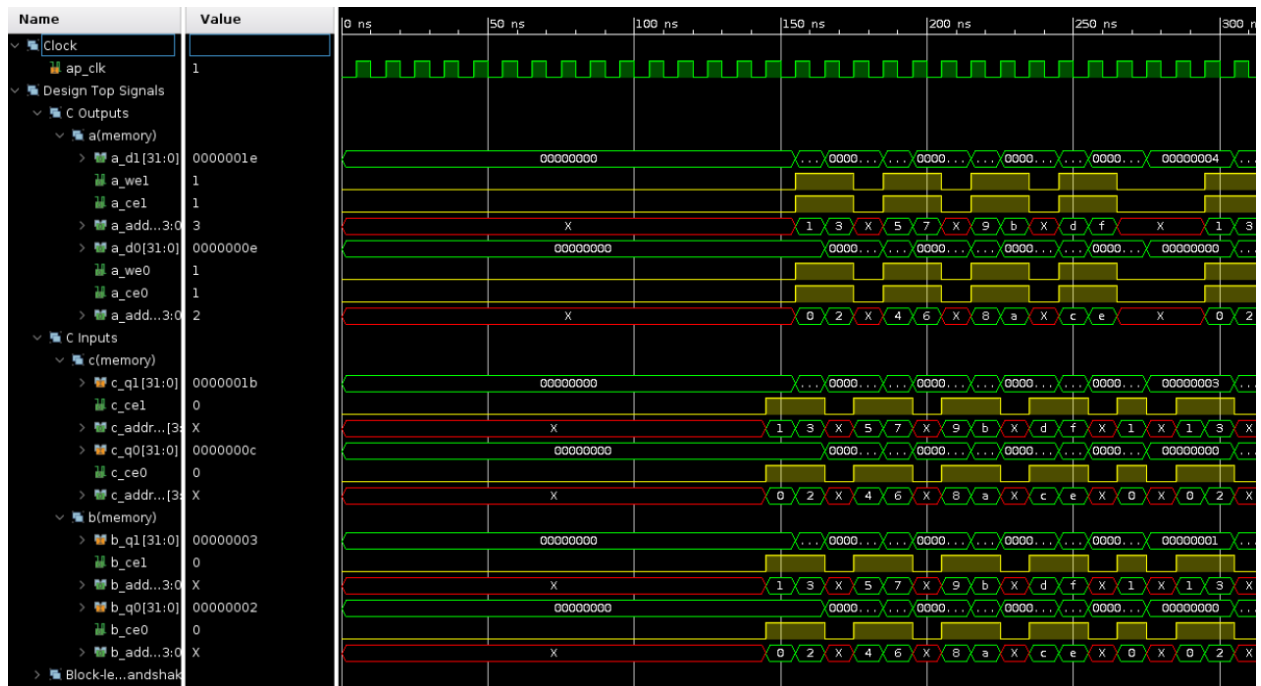


Данные обзора ресурсов имеет вид:

	Resource\Control Step	C0	C1	C2	C3
1	<input type="checkbox"/> I/O Ports				
2	c (p1)		read	read	
3	b (p1)		read	read	
4	b (p0)		read	read	
5	c (p0)		read	read	
6	a (p0)			write	write
7	a (p1)			write	write
8	<input type="checkbox"/> Memory Ports				
9	c (p0)		read	read	
10	c (p1)		read	read	
11	b (p1)		read	read	
12	b (p0)		read	read	
13	a (p1)			write	write
14	a (p0)			write	write
15	<input type="checkbox"/> Expressions				
16	i_1_3_fu_210		+		
17	i_phi_fu_161		phi_mux		
18	i_1_s_fu_198				
19	exitcond_fu_182		icmp		
20	grp_fu_168			+	+
21	grp_fu_175			+	+
22	i_1_2_fu_227				
23	i_1_1_fu_216				

4.3.3. C/RTL моделирование

Временная диаграмма результатов C/RTL моделирования имеет вид:



В данном случае использовался параметр factor 4, таким образом, в проекте был синтезирован цикл на 4 итерации в каждом по 4 чтения/записи который требует 3 такта на выполнение + 1 подготовительный такт – Задержка = $4 \cdot 3 + 1 = 13$, $\Pi = \text{Задержка} + 1 = 14$.

4.4. Решение 4a

В соответствие с планом лабораторной работы устанавливаем:

- clock period 10;
- clock uncertainty 0.1
- установить реализацию UNROLL (factor 4 без *exit check*)

Директивы данного решения имеют вид:

```
▼ ● lab11_1
    ● a
    ● b
    ● c
    ▼  $x+y=?$  Add
        % HLS UNROLL skip_exit_check factor=4
```

4.4.2. Синтез решения 3a

Результаты оценки производительности имеют вид:

Performance Estimates

▣ Timing (ns)

▣ Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	7.006	0.10

▣ Latency (clock cycles)

▣ Summary

Latency		Interval		
min	max	min	max	Type
13	13	13	13	none

По данным результатам синтеза можно сделать вывод, что оценка производительности видно устройства соответствует заданным критериям.

Оценка использования имеет вид:

Utilization Estimates

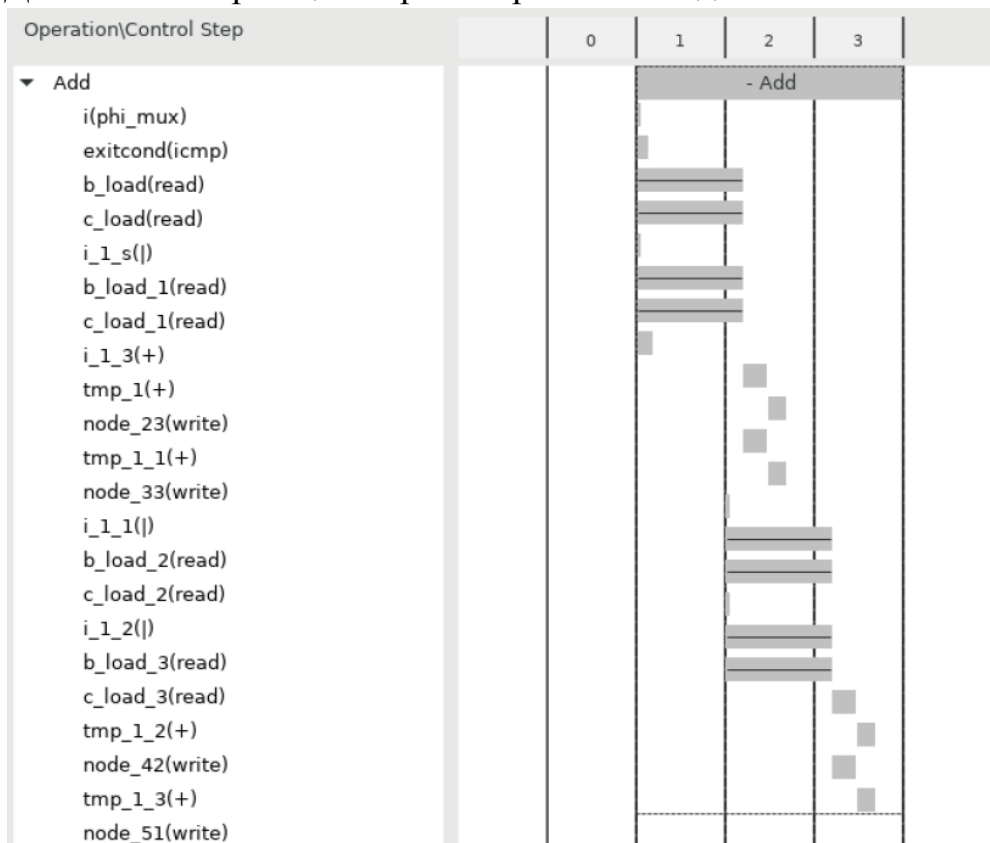
Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	137
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	126
Register	-	-	31	-
Total	0	0	31	263
Available	40	40	16000	8000
Utilization (%)	0	0	~0	3

Профиль производительности имеет вид:

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab11_1	-	13	-	14	-
Add	no	12	3	-	4

Данные планировщика просмотра имеет вид:

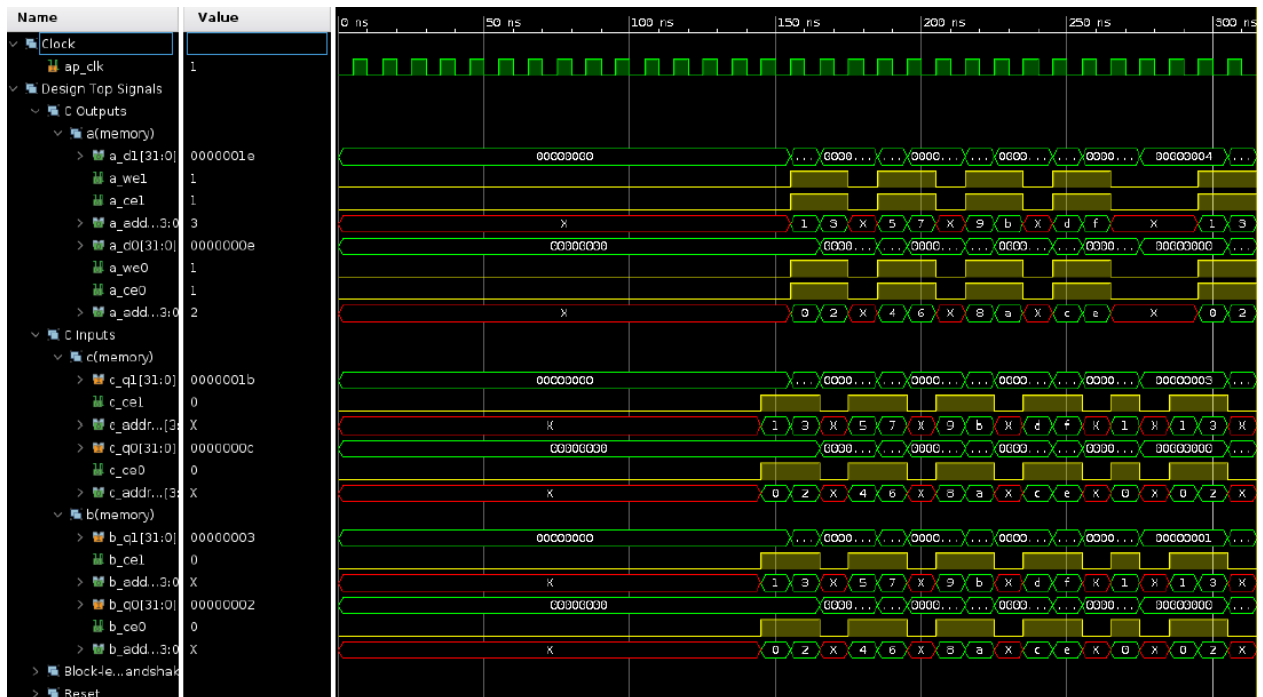


Данные обзора ресурсов имеет вид:

	Resource\Control Step	C0	C1	C2	C3
1	I/O Ports				
2	c (p0)		read	read	
3	b (p1)		read	read	
4	c (p1)		read	read	
5	b (p0)		read	read	
6	a (p1)			write	write
7	a (p0)			write	write
8	Memory Ports				
9	c (p0)		read	read	
10	b (p1)		read	read	
11	b (p0)		read	read	
12	c (p1)		read	read	
13	a (p1)			write	write
14	a (p0)			write	write
15	Expressions				
16	i_1_3_fu_210		+		
17	i_phi_fu_161		phi_mux		
18	i_1_s_fu_198				
19	exitcond_fu_182		icmp		
20	grp_fu_175			+	+
21	grp_fu_168			+	+
22	i_1_1_fu_216				
23	i_1_2_fu_227				

4.4.3. C/RTL моделирование

Временная диаграмма результатов C/RTL моделирования имеет вид:



Данное решение полностью совпадает с предыдущим.

Вывод

С помощью директивы UNROLL можно «развернуть» цикл для получения конвейера, однако, чем больше «глубина» такого конвейера, тем больше количество затраченных ресурсов. Для управления глубиной конвейера используется параметр `factor` что позволяет балансировать между пропускной способностью и требуемыми ресурсами.