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

Лабораторная №11

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Latency

Задание 1

Студенты:
Соболь В.
Темнова А.С.
Группа: 13541/3
Преподаватель:
Антонов А.П.

Санкт-Петербург
2019

Содержание

1. Задание

1. Создать проект lab11_1
2. Микросхема: xa7a12tcsg325-1q
3. Создать функцию, содержащую цикл по образцу

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

   N=16
4. Создать тест lab11_1_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
5. Исследование:
6. 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
7. 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

8. Сравнить два решения (solution_1a и solution_2a) и сделать выводы: зависимость от UNROLL; объяснить (посчитать) число циклов Latency, II...

9. 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

10. Сравнить два решения (solution_2a и solution_3a) и сделать выводы: зависимость от Unroll (factor 4 + ***exit check***) ; объяснить (посчитать) число циклов Latency, II...

11. 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

12. 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

Ниже приведен исходный код устройства и теста.

```
1 #include "lab11_1.h"
2 void lab11_1(int a[N], int b[N], int c[N])
3 {
4   int i;
5   Add:for (i = 0; i < N; i++)
6   {
7     a[i] = b[i] + c[i];
8   }
9 }
```

Рис. 2.1. Исходный код устройства

```
1 #define N 16
```

Рис. 2.2. Заголовочный файл

```c
#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");
  }
}
```

Рис. 2.3. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```
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
```

Рис. 3.1. Скрипт

4. Моделирование

Ниже приведены результаты моделирования.

```
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 ***************
```

Рис. 4.1. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

5. Решение 1a

5.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

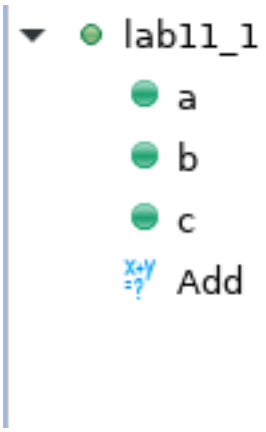


Рис. 5.1. Директивы

5.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

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

Рис. 5.2. Performance estimates

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

Рис. 5.3. Utilization estimates

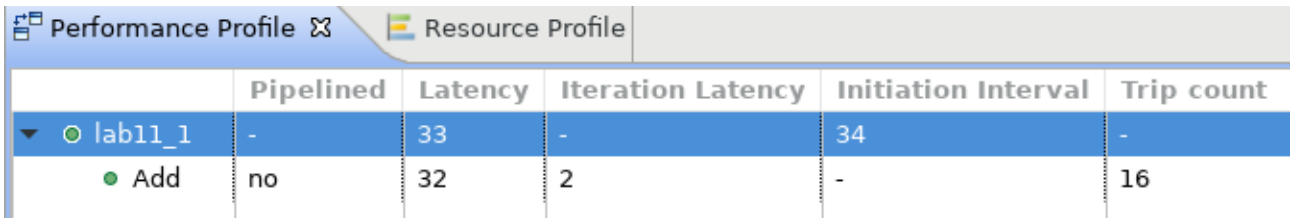


Рис. 5.4. Performance profile

Рис. 5.5. Scheduler viewer

	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_1_fu_84		+	
11	i_phi_fu_71		phi_mux	
12	exitcond_fu_78		icmp	
13	tmp_1_fu_96			+

Рис. 5.6. Resource viewer

5.3. C/RTL моделирование

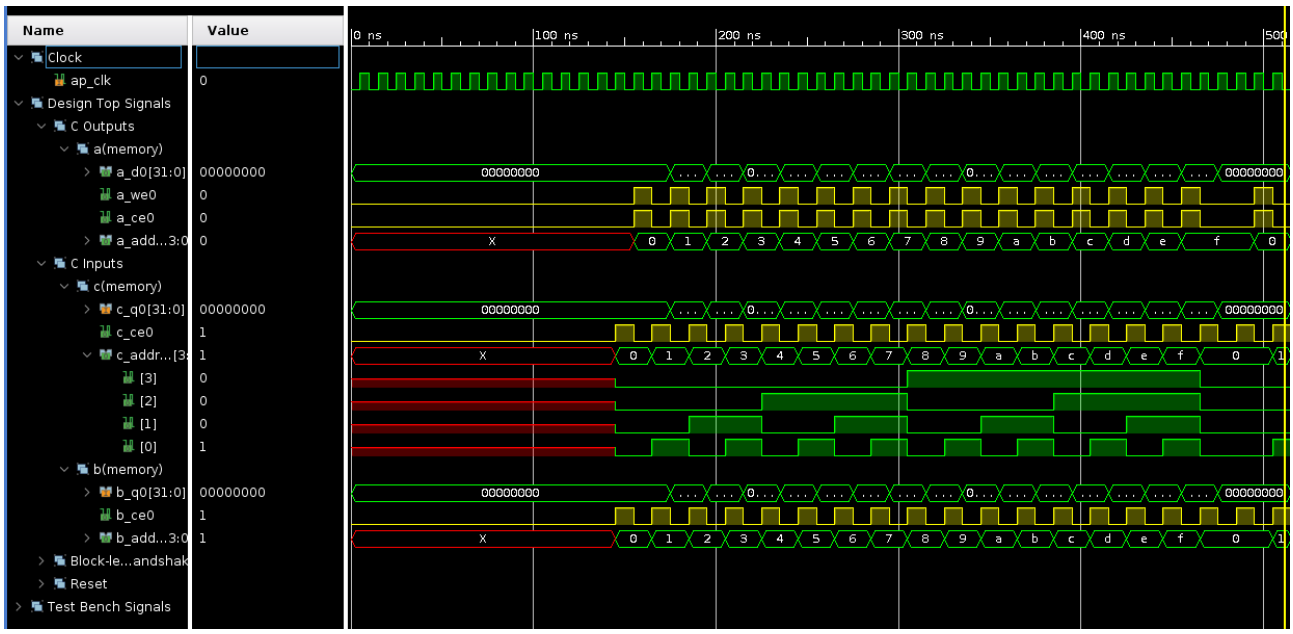


Рис. 5.7. Временная диаграмма

Как видно из диаграммы, выполнение одного цикла требует 2 такта (всего 16 циклов) и 1 такт для инициализации, в итоге Latency = 16*2 + 1 = 33 Данные будут готовы на

выходе еще через 1 такт II = 33+1 = 34 По умолчанию интерфейсы реализованы как ap_memory.

6. Решение 2a

6.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

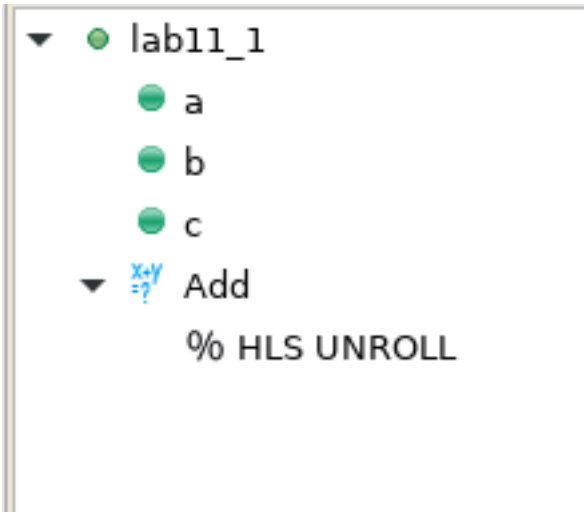


Рис. 6.1. Директивы

6.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

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

Рис. 6.2. Performance estimates

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	40	16000	8000
Utilization (%)	0	0	~0	4

Рис. 6.3. Utilization estimates

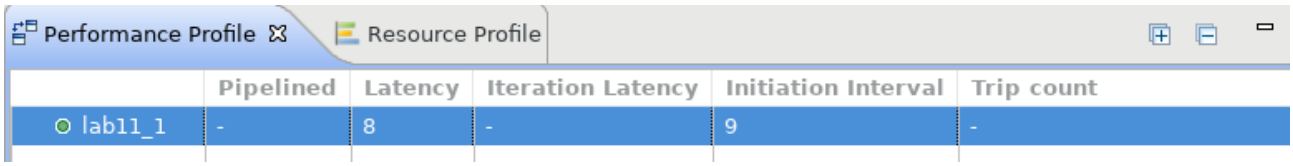


Рис. 6.4. Performance profile

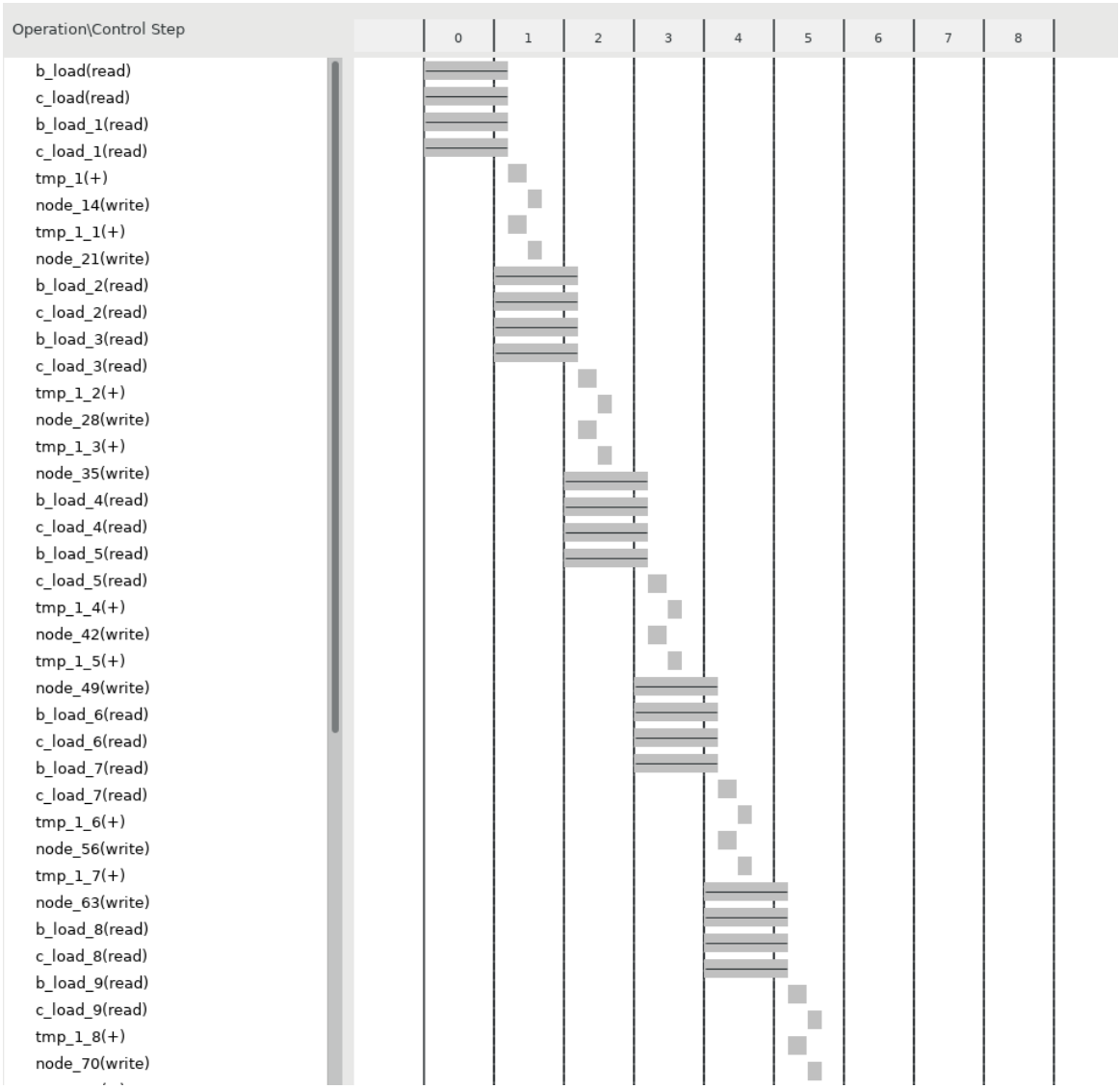


Рис. 6.5. Scheduler viewer

	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		+	+	+	+	+	+	+	+

Рис. 6.6. Resource viewer

6.3. C/RTL моделирование

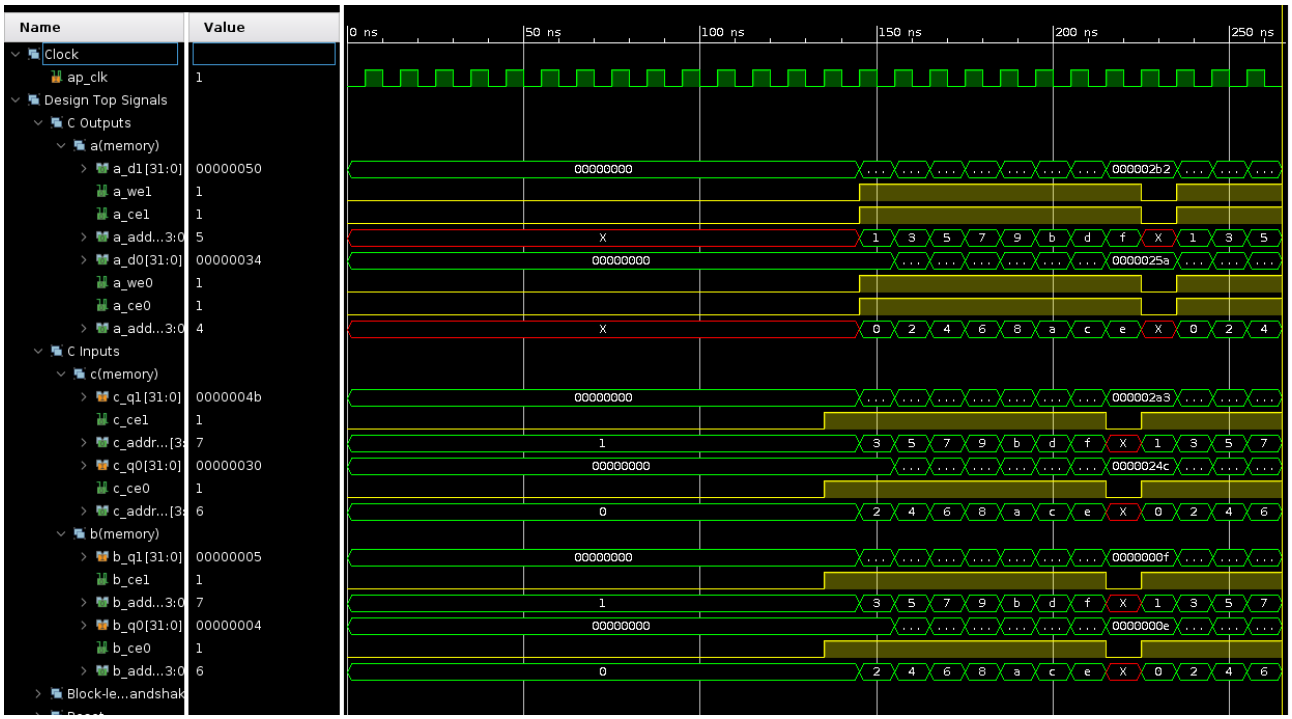


Рис. 6.7. Временная диаграмма

При применении директивы UNROLL, цикл был «развернут» в связи с чем уменьшилось значение Latency с 33 до 8 Это связано с тем, что в предыдущем решении в проекте применялся один блок памяти для каждого массива, в данном же случае, для каждого массива используется 2 блока памяти что позволяет читать одновременно по 2 значения. Поскольку в цикле 16 итераций и одновременно читается по 2 значения, количество циклов уменьшается в 2 раза. И поскольку мы «развернули» оставшиеся 8 циклов в один каскад последовательный чтений/записи – результирующее значения Latency стало 8.

7. Решение 3a

7.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

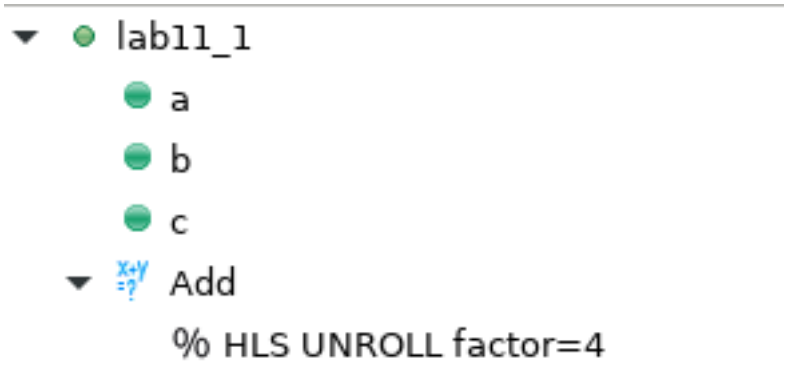


Рис. 7.1. Директивы

7.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

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

Рис. 7.2. Performance estimates

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

Рис. 7.3. Utilization estimates

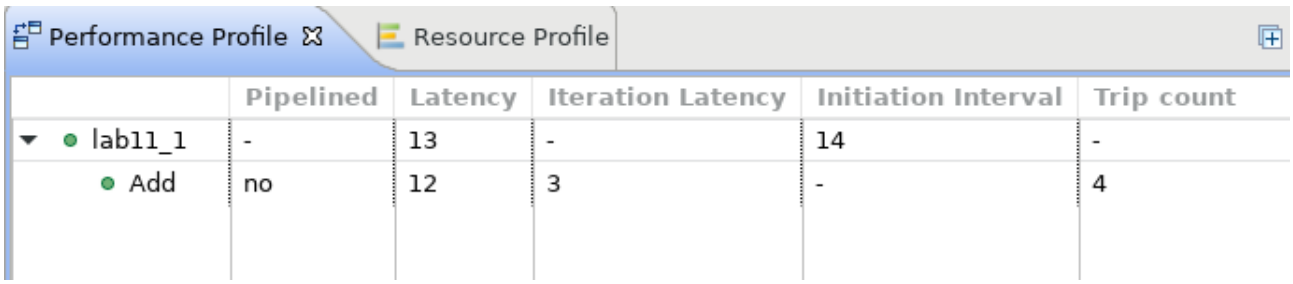


Рис. 7.4. Performance profile

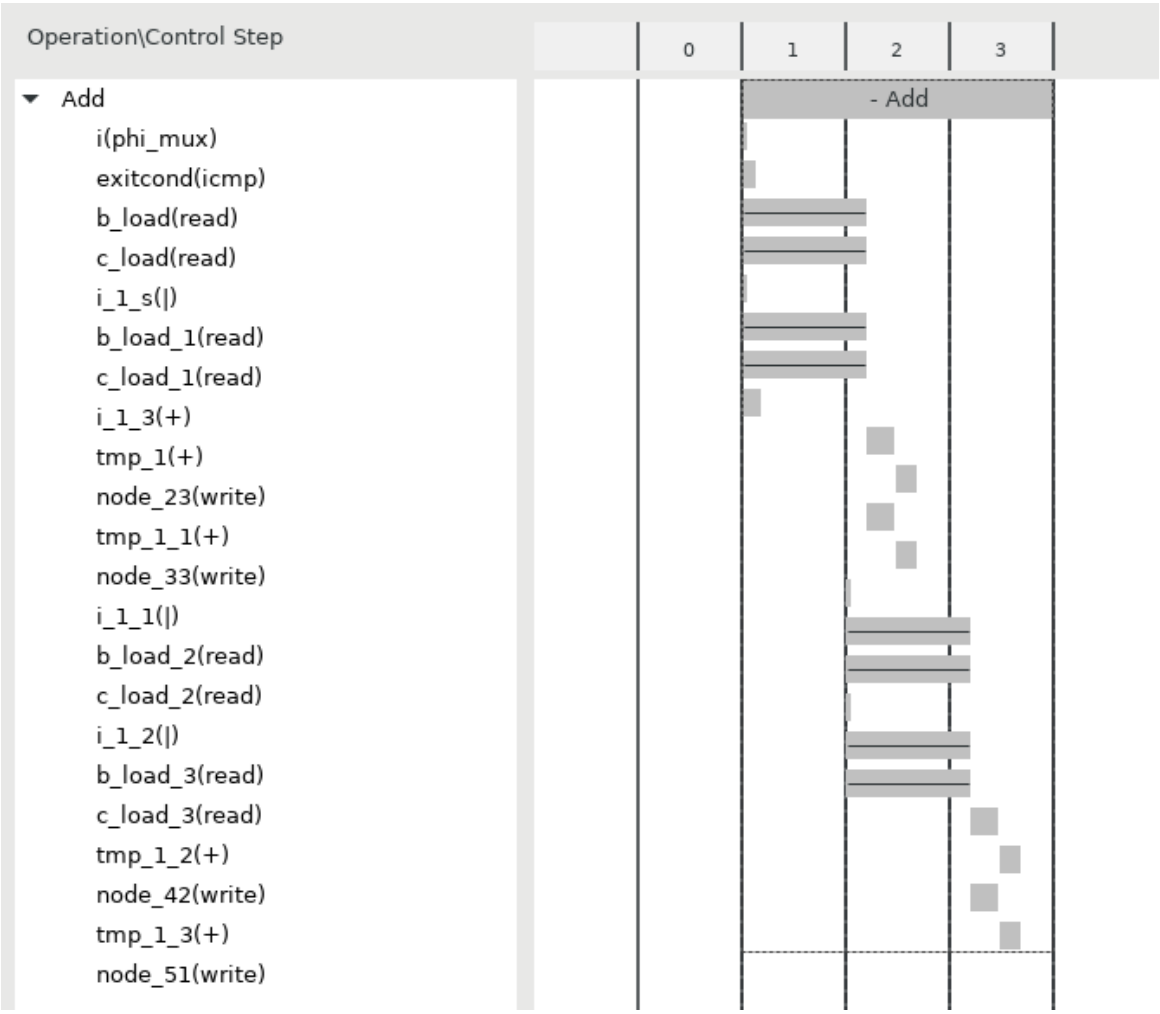


Рис. 7.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3
1	⊟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	⊟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	⊟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			|	

Рис. 7.6. Resource viewer

7.3. C/RTL моделирование

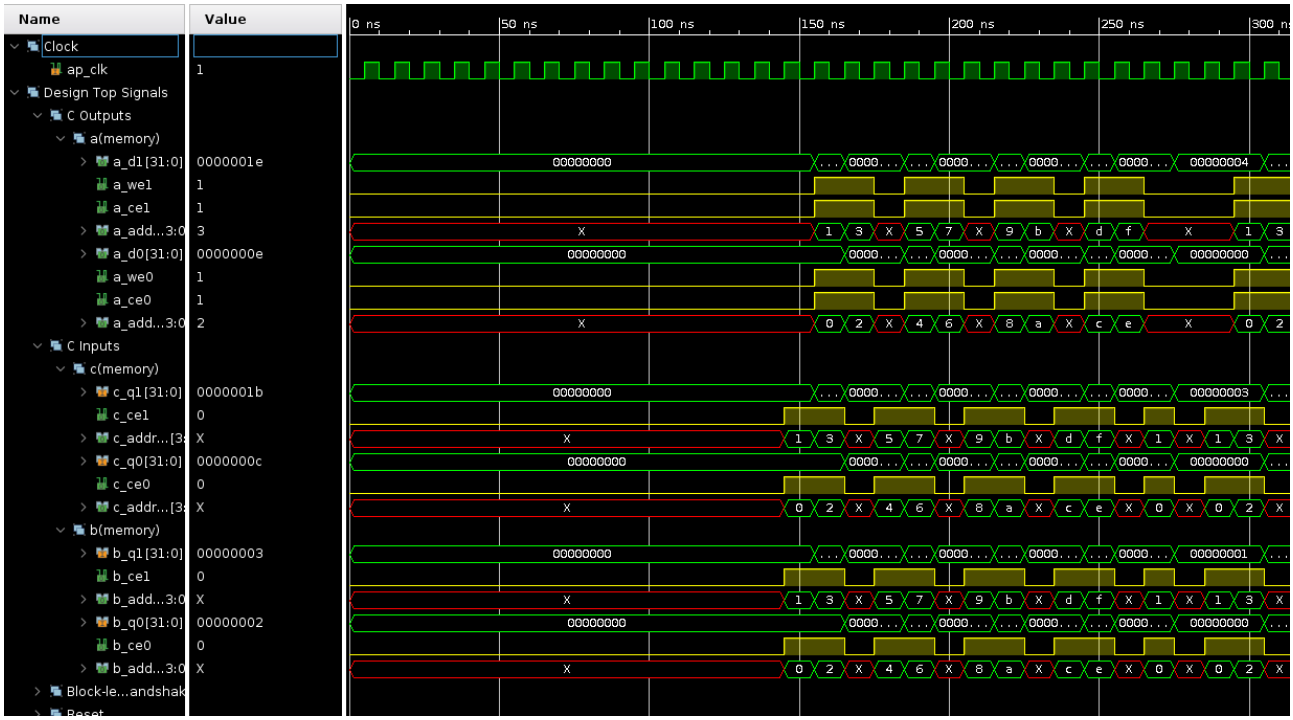


Рис. 7.7. Временная диаграмма

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 – Latency = 4*3 + 1 = 13, II = Latency + 1 = 14.

8. Решение 4a

8.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

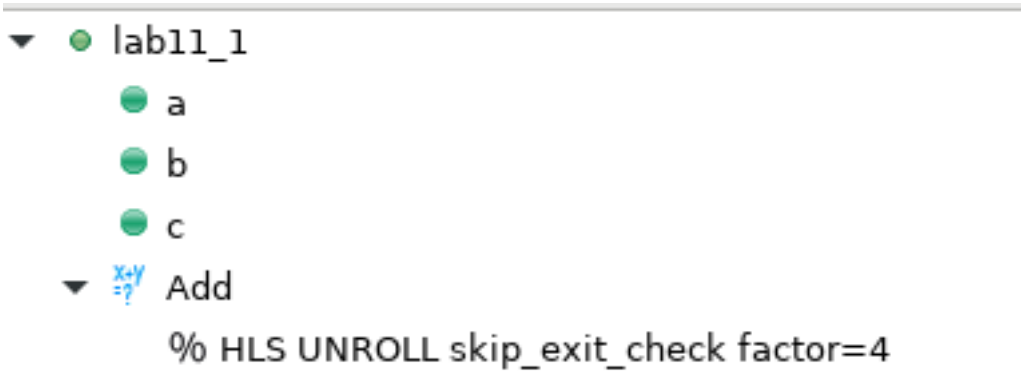


Рис. 8.1. Директивы

8.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

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

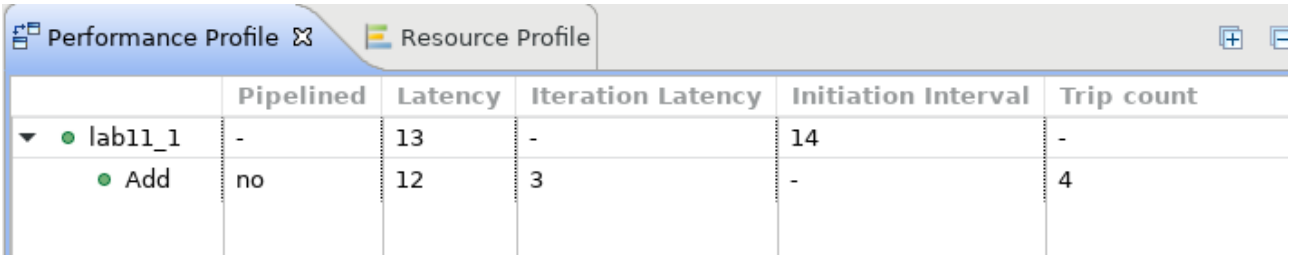
Рис. 8.2. Performance estimates

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

Рис. 8.3. Utilization estimates



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

Рис. 8.4. Performance profile

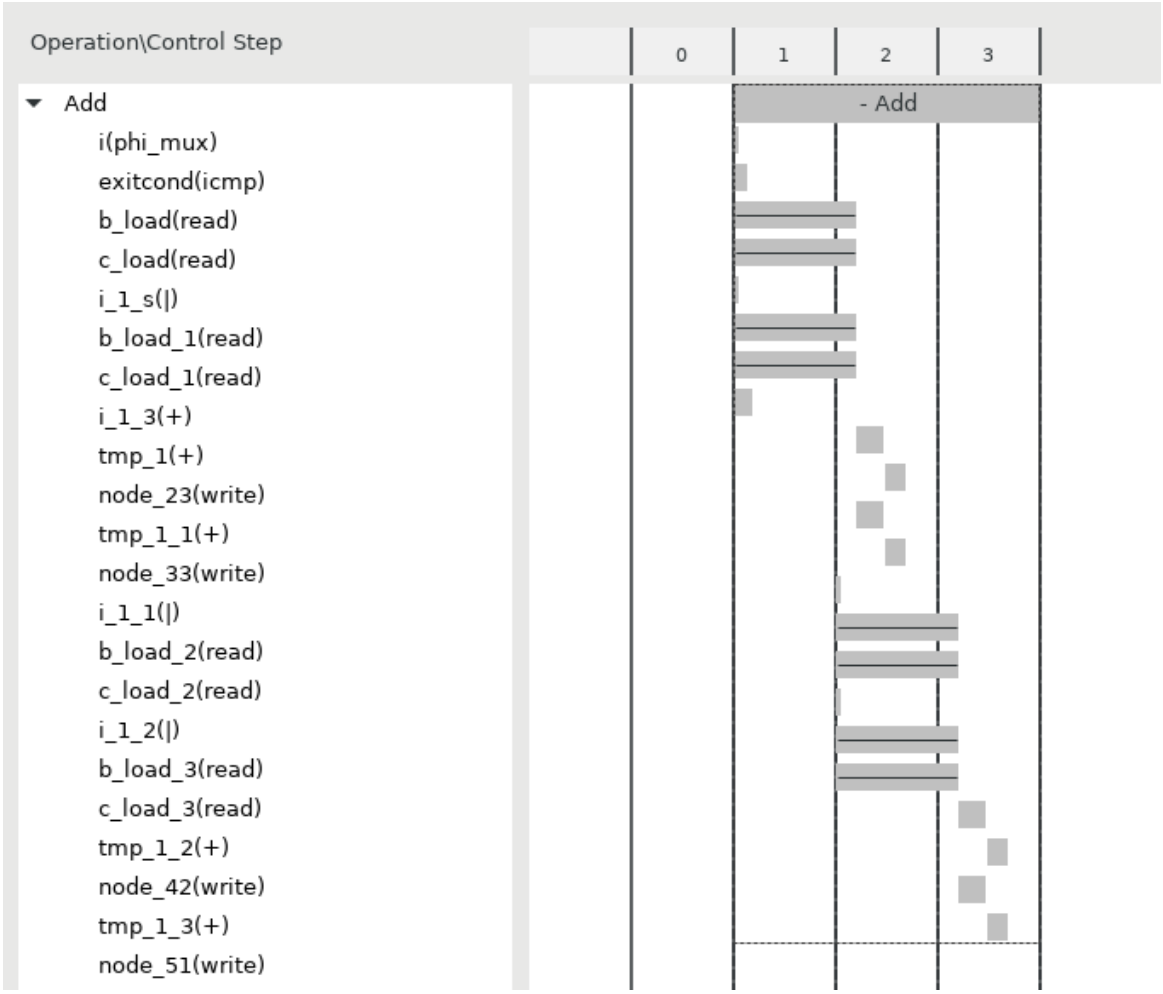


Рис. 8.5. Scheduler viewer

	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			|	

Рис. 8.6. Resource viewer

8.3. C/RTL моделирование

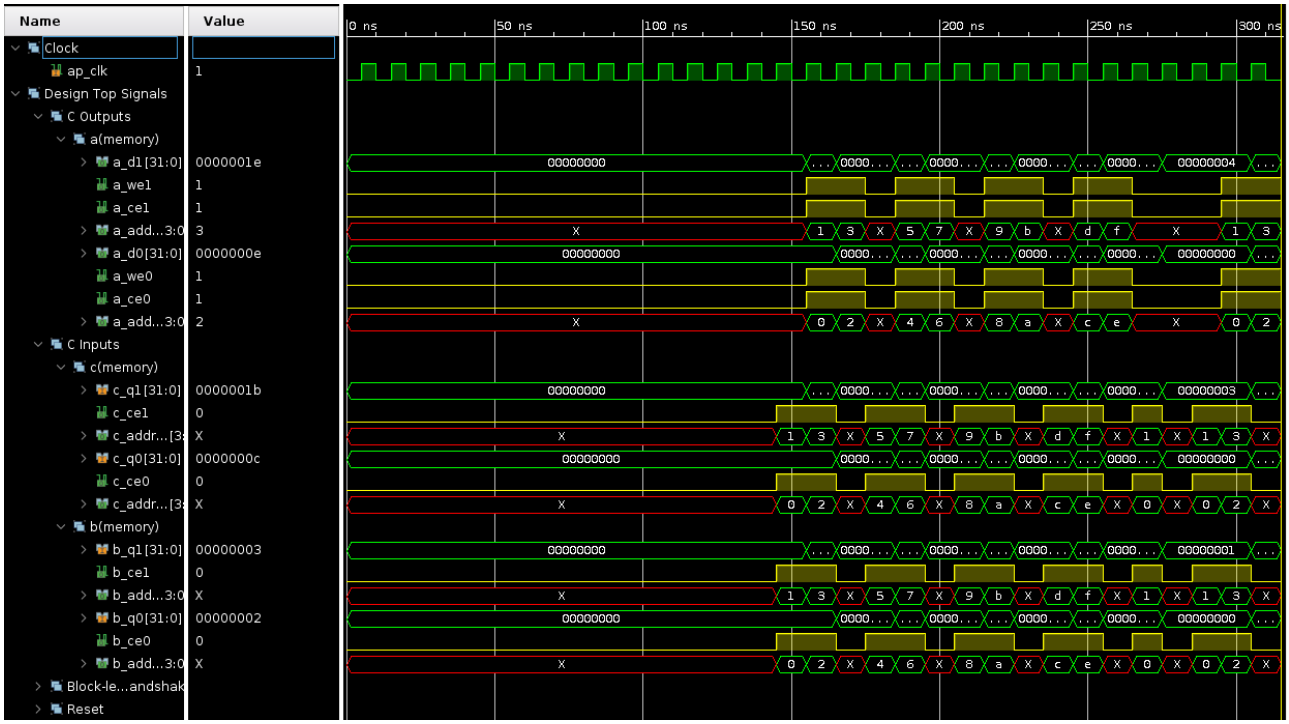


Рис. 8.7. Временная диаграмма

Данное решение полностью совпадает с предыдущим.

9. 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.