

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 3

Студенты:

Соболь В.

Темнова А.С.

Группа: 13541/3

Преподаватель:

Антонов А.П.

Санкт-Петербург
2019

Содержание

1. Задание	3
2. Исходный код	5
3. Скрипт	7
4. Моделирование	8
5. Решение 1a	9
5.1. Директивы	9
5.2. Синтез	9
6. Решение 2a	12
6.1. Директивы	12
6.2. Синтез	13
7. Решение 3a	16
7.1. Директивы	16
7.2. Синтез	16
8. Решение 4a	19
8.1. Директивы	19
8.2. Синтез	19
9. Вывод	22

1. Задание

1. Создать проект lab11_3
2. Микросхема: xa7a12tcsg325-1q
3. Познакомиться с исходным кодом функции loop_imperfect.c
4. Познакомиться с исходным кодом теста loop_imperfect_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
- преобразовать функцию из формы Unperfect в форму Perfect
- установить реализацию ПО УМОЛЧАНИЮ
- 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; объяснить (посчитать) число циклов Latency, И...
9. Solution_3a
- задать: clock period 10; clock_uncertainty 0.1
 - использовать функцию преобразованную в форму Perfect
 - установить реализацию Flattend для внутреннего цикла
 - 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 (интерес представляет количество и тип портов)
10. Сравнить два решения (solution_2a и solution_3a) и сделать выводы; объяснить (посчитать) число циклов Latency, И...
11. Solution_4a
- задать: clock period 10; clock_uncertainty 0.1
 - использовать функцию, преобразованную в форму Perfect
 - установить реализацию UNROLLED для внутреннего цикла
 - 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 (интерес представляет количество и тип портов)
- 12. Сравнить два решения (solution_3a и solution_4a) и сделать выводы; объяснить (посчитать) число циклов Latency, II...

2. Исходный код

Ниже приведен исходный код устройства и теста.

```

1 #include "loop_imperfect.h"
2
3 void loop_imperfect(din_t A[N], dout_t B[N]) {
4
5     int i, j;
6     dint_t acc;
7
8     LOOP_I: for(i=0; i < 20; i++){
9         acc = 0;
10        LOOP_J: for(j=0; j < 20; j++){
11            acc += A[j] * j;
12        }
13        if (i%2 == 0)
14            B[i] = acc / 20;
15        else
16            B[i] = 0;
17    }
18 }
19 }
```

Рис. 2.1. Исходный код устройства

```

1 #include "loop_imperfect.h"
2
3 void loop_imperfect(din_t A[N], dout_t B[N]) {
4
5     int i, j;
6     dint_t acc;
7
8     LOOP_I: for(i=0; i < 20; i++){
9         LOOP_J: for(j=0; j < 20; j++){
10             if(j == 0)
11             {
12                 acc = 0;
13             }
14             acc += A[j] * j;
15             if(j == 19)
16             {
17                 if (i%2 == 0)
18                     B[i] = acc / 20;
19             }
20             else
21                 B[i] = 0;
22         }
23     }
24 }
25
26 }

```

Рис. 2.2. Исходный код устройства, форма PERFECT

```

1 #ifndef _LOOP_IMPERFECT_H_
2 #define _LOOP_IMPERFECT_H_
3
4 #include "ap_cint.h"
5 #define N 20
6
7 typedef int5 din_t;
8 typedef int12 dint_t;
9 typedef int6 dout_t;
10
11 void loop_imperfect(din_t A[N], dout_t B[N]);
12
13 #endif

```

Рис. 2.3. Заголовочный файл

```

1 #include <stdio.h>
2 #include "loop_imperfect.h"
3
4 int main() {
5     din_t A[N];
6     dout_t B[N];
7
8     int i, retval = 0;
9     FILE *fp;
10
11     for (i = 0; i < N; ++i) {
12         A[i] = i;
13     }
14     // Save the results to a file
15     fp = fopen("result.dat", "w");
16
17     // Call the function
18     loop_imperfect(A, B);
19     for (i = 0; i < N; ++i) {
20         fprintf(fp, "%d\n", B[i]);
21     }
22     fclose(fp);
23
24     // Compare the results file with the golden results
25     retval = system("diff --brief -w result.dat result.golden.dat");
26     if (retval != 0) {
27         printf("Test failed!!!\n");
28         retval = 1;
29     } else {
30         printf("Test passed!\n");
31     }
32
33     // Return 0 if the test passed
34     return retval;
35 }

```

Рис. 2.4. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```

1 open_project -reset lab11_3_imperfect
2
3 add_files loop_imperfect.c
4 add_files -tb loop_imperfect_test.c
5 add_files -tb result.golden.dat
6 set_top loop_imperfect
7
8 open_solution solution_1a -reset
9 set_part {xa7a12tcsg325-1q}
10 create_clock -period 10ns
11 set_clock_uncertainty 0.1
12
13 csim_design
14 csynth_design
15 # cosim_design -trace_level all
16
17 open_project -reset lab11_3_perfect
18
19 add_files loop_perfect.c
20 add_files -tb loop_imperfect_test.c
21 add_files -tb result.golden.dat
22 set_top loop_imperfect
23
24 set solutions [list 2a 3a 4a]
25
26 foreach sol $solutions {
27     open_solution solution_$sol -reset
28     set_part {xa7a12tcsg325-1q}
29     create_clock -period 10ns
30     set_clock_uncertainty 0.1
31
32     if {$sol == "3a"} {
33         set_directive_loop_flatten "loop_imperfect/LOOP_J"
34     }
35     if {$sol == "4a"} {
36         set_directive_unroll "loop_imperfect/LOOP_J"
37     }
38
39     csim_design
40     csynth_design
41     # cosim_design -trace_level all
42 }
43
44 exit

```

Рис. 3.1. Скрипт

4. Моделирование

Ниже приведены результаты моделирования.


```
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/1573141575889476931815
INFO: [APCC 202-1] APCC is done.
    Generating csim.exe
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 решении были установлены директивы, приведённые ниже.

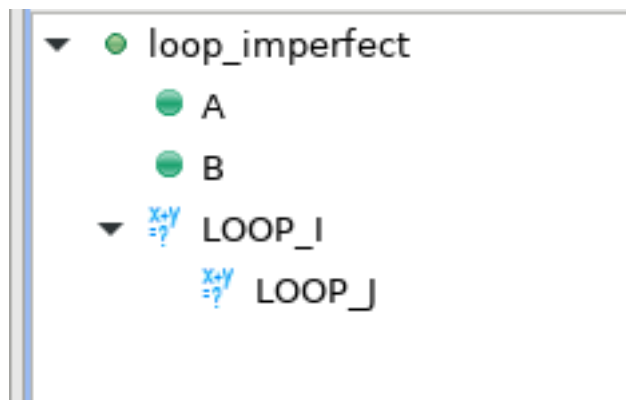


Рис. 5.1. Директивы

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	9.332	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
861	861	861	861	none

Рис. 5.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	-	0	118
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	90
Register	-	-	72	-
Total	0	2	72	208
Available	40	40	16000	8000
Utilization (%)	0	5	~0	2

Рис. 5.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
loop_imperfect	-	861	-	862	-
LOOP_I	no	860	43	-	20
LOOP_J	no	40	2	-	20

Рис. 5.4. Performance profile

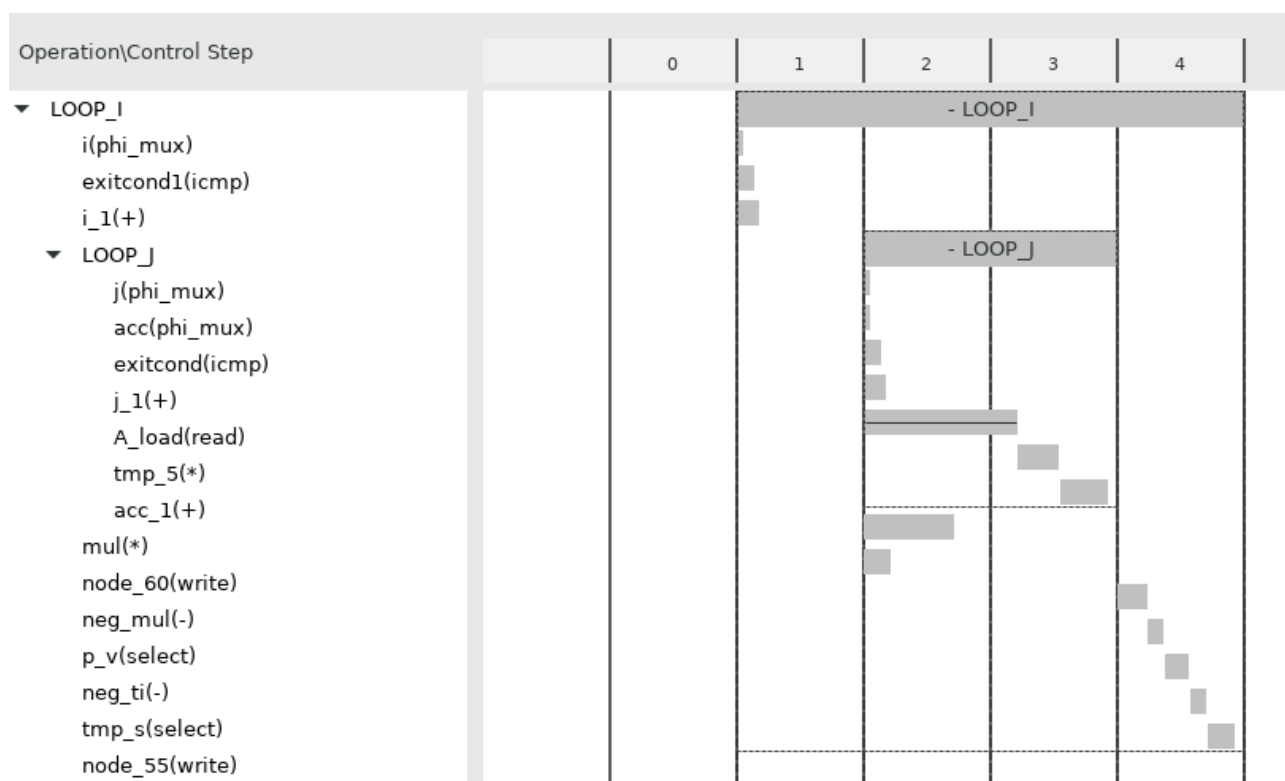


Рис. 5.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4
1	I/O Ports					
2	B(p0)			write		write
3	A(p0)			read		
4	Memory Ports					
5	B(p0)			write		write
6	A(p0)			read		
7	Expressions					
8	i_l_fu_135		+			
9	i_phi_fu_93		phi_mux			
10	exitcond1_fu_129		icmp			
11	j_l_fu_147			+		
12	acc_phi_fu_117			phi_mux		
13	j_phi_fu_105			phi_mux		
14	mul_fu_253			*		
15	exitcond_fu_141			icmp		
16	grp_fu_261				+	
17	neg_mul_fu_187					-
18	neg_ti_fu_229					-
19	p_v_fu_217					select
20	tmp_s_fu_239					select

Рис. 5.6. Resource viewer

Выполнение одного внутреннего цикла Loop j зависит от скорости чтения значения и нескольких небольших операций, т.е. пропускная способность = 2 откуда $LatencyJ = 20 * 2 = 40$. Для внешнего цикла значение $LatencyI = LatencyJ + 1$ общий подготовительный такт + 1 подготовительный такт для цикла J + 1 завершающий такт = $40 + 1 + 1 + 1 = 43$. Так как число итераций = 20, то $Latency = 20 * LatencyI + 1$ (подготовительный такт) = 861.

6. Решение 2а

6.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

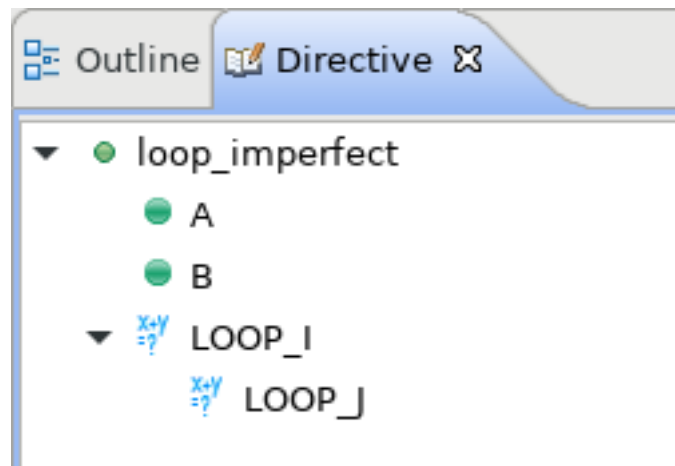


Рис. 6.1. Директивы

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	9.332	0.10

[-] Latency (clock cycles)

[-] Summary

Latency		Interval		
min	max	min	max	Type
1241	2041	1241	2041	none

Рис. 6.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	-	0	152
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	68
Register	-	-	98	-
Total	0	2	98	220
Available	40	40	16000	8000
Utilization (%)	0	5	~0	2

Рис. 6.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● loop_imperfect	-	1241~2041	-	1242 ~ 2042	-
▼ ● LOOP_I	no	1240 ~ 2040	62 ~ 102	-	20
● LOOP_J	no	60 ~ 100	3 ~ 5	-	20

Рис. 6.4. Performance profile

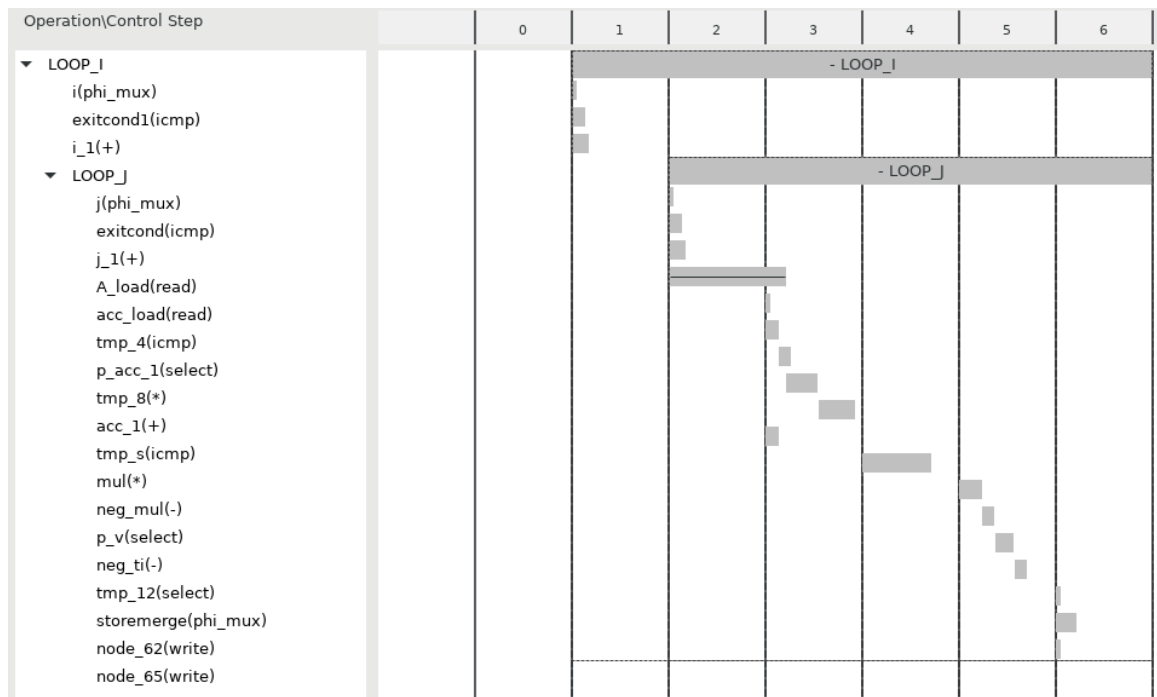


Рис. 6.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4	C5	C6
1	I/O Ports							
2	A(p0)			read				
3	B(p0)							write
4	Memory Ports							
5	A(p0)			read				
6	B(p0)							write
7	Expressions							
8	i_phi_fu_91		phi_mux					
9	i_1_fu_132		+					
10	exitcond1_fu_126		icmp					
11	j_phi_fu_102			phi_mux				
12	j_1_fu_149			+				
13	exitcond_fu_143			icmp				
14	p_acc_1_fu_169				select			
15	grp_fu_269				+			
16	tmp_4_fu_163				icmp			
17	tmp_s_fu_185				icmp			
18	mul_fu_277					*		
19	neg_mul_fu_206						-	
20	neg_ti_fu_247						-	
21	tmp_12_fu_257						select	
22	p_v_fu_235						select	
23	storemerge_phi_fu_114							phi_mux

Рис. 6.6. Resource viewer

Как видно по полученным результатам, после преобразования в форму PERFECT, производительность значительно ухудшилась.

7. Решение За

7.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

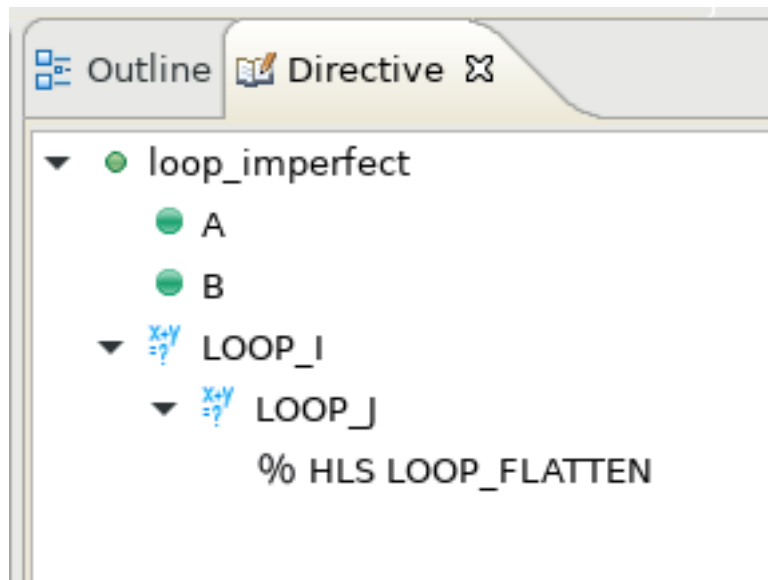


Рис. 7.1. Директивы

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	9.332	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
1201	2001	1201	2001	none

Рис. 7.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	-	0	180
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	74
Register	-	-	116	-
Total	0	2	116	254
Available	40	40	16000	8000
Utilization (%)	0	5	~0	3

Рис. 7.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● loop_imperfect	-	1201~2001	-	1202 ~ 2002	-
● LOOP_I_LOOP_J	no	1200 ~ 2000	3 ~ 5	-	400

Рис. 7.4. Performance profile

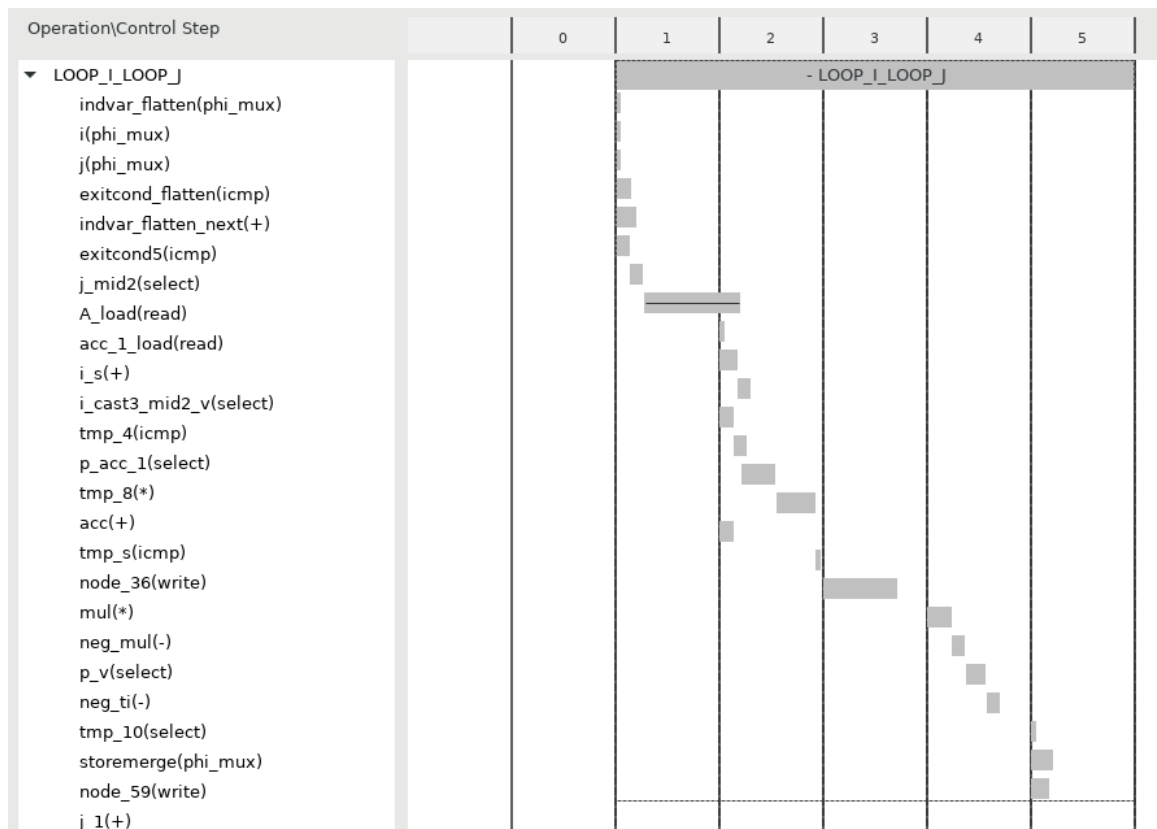


Рис. 7.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4	C5
1	I/O Ports						
2	A(p0)		read				
3	B(p0)						write
4	Memory Ports						
5	A(p0)		read				
6	B(p0)						write
7	Expressions						
8	indvar_flatten_next_fu_138		+				
9	indvar_flatten_phi_fu_90		phi_mux				
10	j_phi_fu_113		phi_mux				
11	i_phi_fu_101		phi_mux				
12	j_mid2_fu_150		select				
13	exitcond5_fu_144		icmp				
14	exitcond_flatten_fu_132		icmp				
15	i_s_fu_166			+			
16	i_cast3_mid2_v_fu_172			select			
17	p_acc_1_fu_192			select			
18	grp_fu_293			+			
19	tmp_s_fu_207			icmp			
20	tmp_4_fu_187			icmp			
21	mul_fu_302				*		
22	neg_ti_fu_271					-	
23	neg_mul_fu_238					-	
24	tmp_10_fu_281					select	
25	p_v_fu_260					select	
26	j_1_fu_288						+
27	storemerge_phi_fu_124						phi_mux

Рис. 7.6. Resource viewer

Видно, что два вложенных цикла превратились в один, с числом итераций 400, но так как длительность одной итерации составляет от 3 до 5 тактов, то общая производительность хуже, чем у первого решения.

По сравнению с предыдущим решением, производительность повысилась, т.к. были убраны задержки на инициализацию внутреннего цикла (3 такта на каждой итерации).

8. Решение 4а

8.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

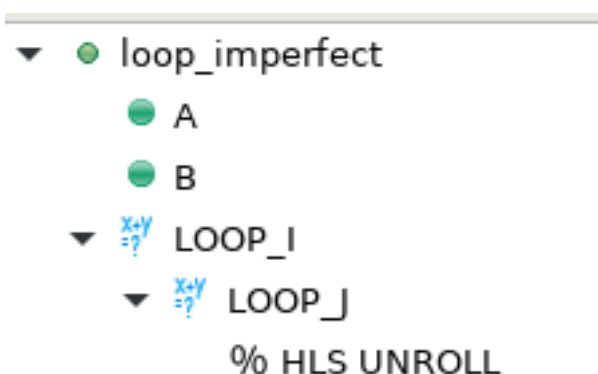


Рис. 8.1. Директивы

8.2. Синтез

По оценке производительности видно, что устройство соответствует **НЕ** заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	11.000	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
261	301	261	301	none

Рис. 8.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	4	-	-
Expression	-	-	0	478
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	218
Register	-	-	229	-
Total	0	4	229	696
Available	40	40	16000	8000
Utilization (%)	0	10	1	8

Рис. 8.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
loop_imperfect	-	261~301	-	262 ~ 302	-
LOOP_I	no	260 ~ 300	13 ~ 15	-	20

Рис. 8.4. Performance profile

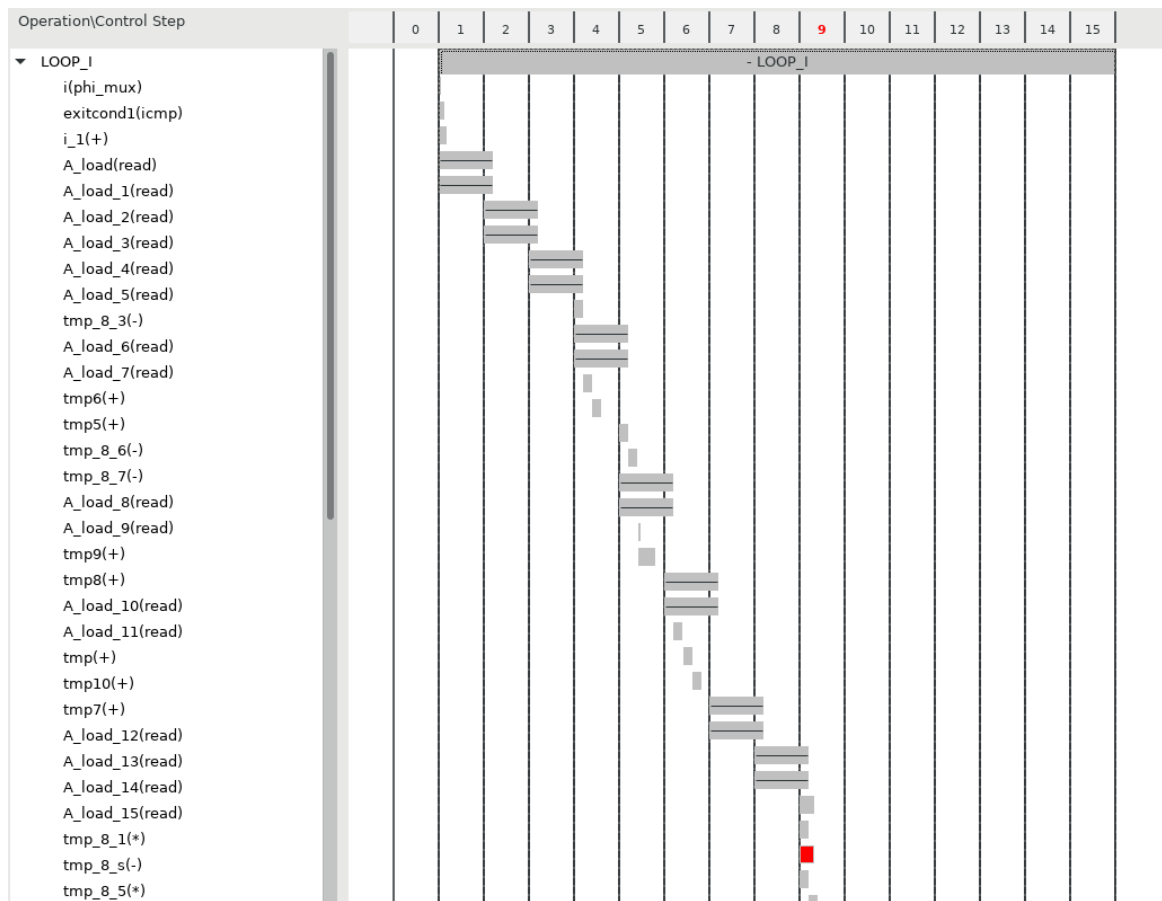


Рис. 8.5. Scheduler viewer

На рисунке выше видны элементы, задержка в которых превышает допустимую.

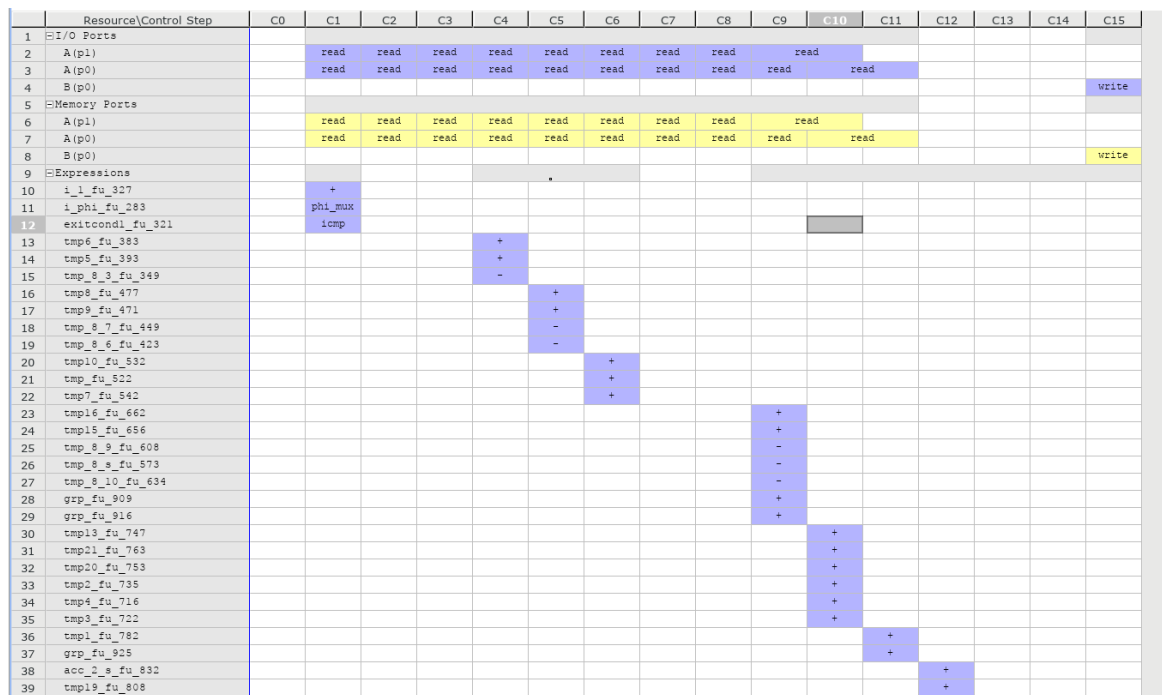


Рис. 8.6. Resource viewer

За счёт разворачивания внутреннего цикла, данное решение имеет лучшую оценочную производительность, хоть и не укладывается в отведённый временной интервал.

9. Вывод

Как видим, 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 получить «золотую середину» между пропускной способностью и требуемыми ресурсами. Для объединения циклов и оптимизации проекта при написании кода стоит учитывать что циклы лучше писать в виде `PERFECT` для максимальной оптимизации результата.