

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 5

Студенты:

Соболь В.

Темнова А.С.

Группа: 13541/3

Преподаватель:

Антонов А.П.

Санкт-Петербург
2019

Содержание

1. Задание	3
2. Исходный код	4
3. Скрипт	6
4. Моделирование	6
5. Решение 1a	7
5.1. Директивы	7
5.2. Синтез	7
6. Решение 2a	9
6.1. Директивы	9
6.2. Синтез	10
7. Решение 3a	12
7.1. Директивы	12
7.2. Синтез	12
8. Вывод	14

1. Задание

1. Создать проект lab11_5
2. Микросхема: xa7a12tcsg325-1q
3. Познакомьтесь с исходным кодом функции loop_var.c
4. Познакомьтесь с исходным кодом теста loop_var_test.c для проверки функции.
5. Осуществить моделирование (с выводом результатов в консоль)
6. Исследование:
7. 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

8. Solution_2a

- задать: clock period 10; clock_uncertainty 0.1
- установить LOOP_TRIPCOUNT (MIN 4 MAX 32 AVERAGE 18)
- 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
9. Сравнить два решения (solution_1a и solution_2a) и сделать выводы: объяснить (посчитать) число циклов Latency, П...
10. Solution_3a
- задать: clock period 10; clock_uncertainty 0.1
 - установить (В исходном коде: assert (width<64))
 - 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
11. Сравнить два решения (solution_2a и solution_3a) и сделать выводы: объяснить (посчитать) число циклов Latency, П...

2. Исходный код

Ниже приведен исходный код устройства и теста.

```

1 #include "loop_var.h"
2
3 dout_t loop_var(din_t A[N], dsel_t width) {
4     #ifdef ASSERT_WIDTH_64
5         assert(width < 64);
6     #endif
7
8     dout_t out_accum=0;
9     dsel_t x;
10
11     LOOP_X: for (x=0;x<width; x++) {
12         out_accum += A[x];
13     }
14
15     return out_accum;
16 }
```

Рис. 2.1. Исходный код устройства

```

1 #ifndef _LOOP_VAR_H_
2 #define _LOOP_VAR_H_
3
4 #include <stdio.h>
5 #include "ap_cint.h"
6 #include "assert.h"
7 #define N 32
8
9 typedef int8 din_t;
10 typedef int13 dout_t;
11 typedef uint5 dsel_t;
12
13 dout_t loop_var(din_t A[N], dsel_t width);
14
15 #endif

```

Рис. 2.2. Заголовочный файл

```

1 #include "loop_var.h"
2
3 int main () {
4     din_t A[N];
5     dsel_t x=0;
6     dout_t accum;
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     for(i=0; i<N;++i) {
19         x=i;
20         accum = loop_var(A,x);
21         fprintf(fp, "%d_\n", accum);
22     }
23     fclose(fp);
24
25     // Compare the results file with the golden results
26     retval = system("diff -brief -w result.dat result.golden.dat");
27     if (retval != 0) {
28         printf("Test_failed_!!!\n");
29         retval=1;
30     } else {
31         printf("Test_passed_!\n");
32     }
33
34     // Return 0 if the test passed
35     return retval;
36 }

```

Рис. 2.3. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```
1 open_project -reset lab11_5
2
3 add_files loop_var.c
4 add_files -tb loop_var_test.c
5 add_files -tb result.golden.dat
6
7 set_top loop_var
8
9 set solutions [list 1a 2a]
10
11 foreach sol $solutions {
12     open_solution solution_$sol -reset
13     set_part {xa7a12tcs325-1q}
14     create_clock -period 10ns
15     set_clock_uncertainty 0.1
16
17     if {$sol == "2a"} {
18         set_directive_loop_tripcount -max 32 -min 4 -avg 18 loop_var/LOOP_X
19     }
20
21     csim_design
22     csynth_design
23     # cosim_design -trace_level all
24 }
25
26
27 add_files loop_var.c -cflags "-DASSERT_WIDTH_64"
28
29 open_solution solution_3a -reset
30 set_part {xa7a12tcs325-1q}
31 create_clock -period 10ns
32 set_clock_uncertainty 0.1
33
34 csim_design
35 csynth_design
36 # cosim_design -trace_level all
37
38 exit
```

Рис. 3.1. Скрипт

4. Моделирование

Ниже приведены результаты моделирования.

```

INFO: [HLS 200-10] On os "Arch Linux"
INFO: [HLS 200-10] In directory '/home/sobol/Downloads/labs_from_8/lab11_z5/source/la
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/1662431575892380963243
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 *****
INFO: [HLS 200-10] Analyzing design file loop_var.gl

```

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.

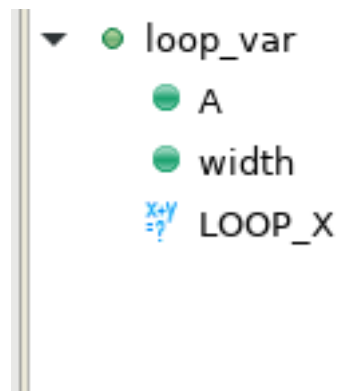


Рис. 5.1. Директивы

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	4.286	0.10

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
?	?	?	?	none

Рис. 5.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	46
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	39
Register	-	-	26	-
Total	0	0	26	85
Available	40	40	16000	8000
Utilization (%)	0	0	~0	1

Рис. 5.3. Utilization estimates

Performance Profile		Resource Profile				
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count	
loop_var	-	-	-	-	-	
LOOP_X	no	-	2	-	-	

Рис. 5.4. Performance profile

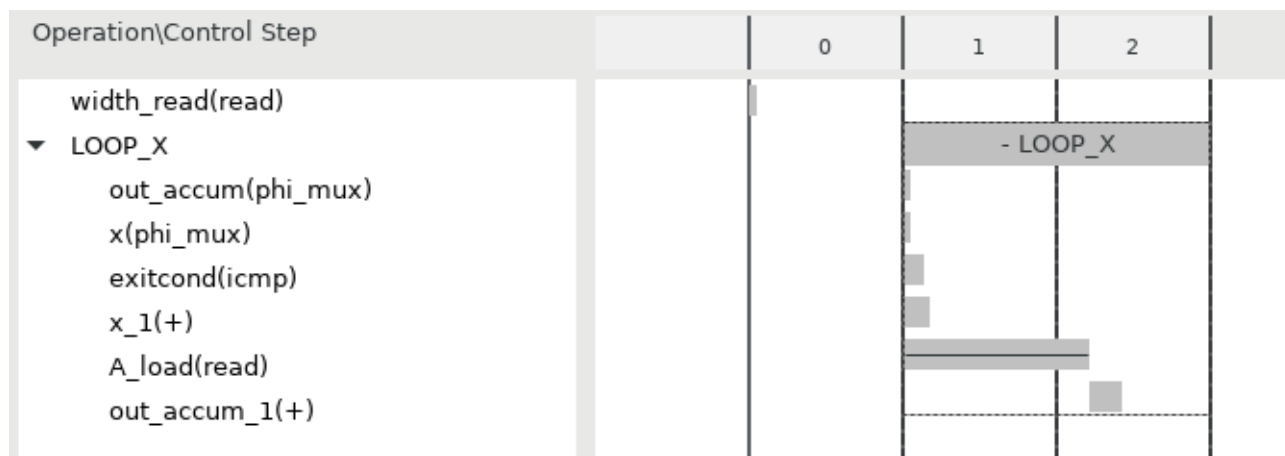


Рис. 5.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	I/O Ports			
2	width	read		
3	A(p0)		read	
4	ap_return		ret	
5	Memory Ports			
6	A(p0)		read	
7	Expressions			
8	x_phi_fu_59		phi_mux	
9	out_accum_phi_fu_47		phi_mux	
10	x_1_fu_71		+	
11	exitcond_fu_66		icmp	
12	out_accum_1_fu_86			+

Рис. 5.6. Resource viewer

По умолчанию для порта A реализован интерфейс ap_memory с шиной данных 8бит и шиной адреса 5бит (32 адреса). Значение Latency неизвестно т.к. неизвестно количество итераций цикла, однако известно что выполнение одной итерации цикла занимает 2 такта.

6. Решение 2а

6.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

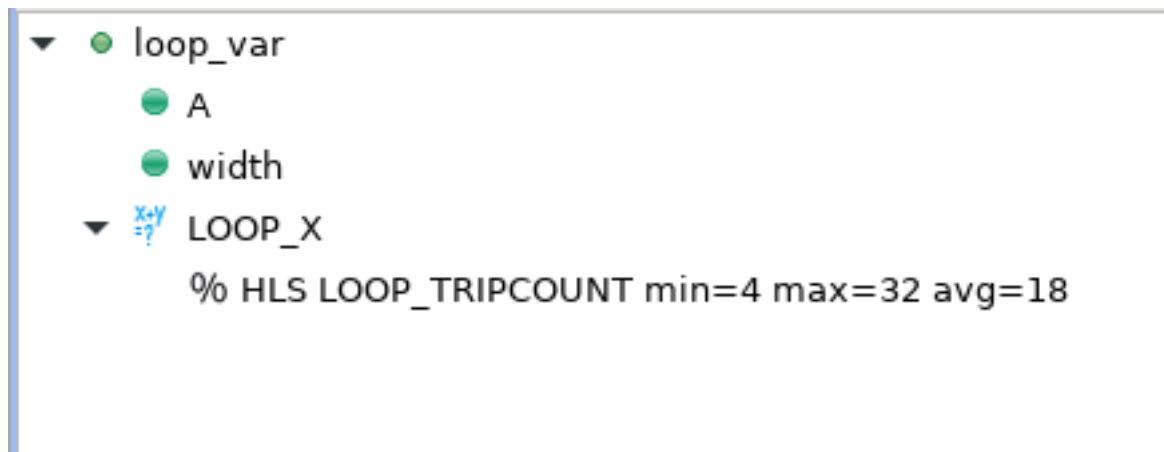


Рис. 6.1. Директивы

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	4.286	0.10

☐ **Latency (clock cycles)**

☐ **Summary**

Latency		Interval		
min	max	min	max	Type
9	65	9	65	none

Рис. 6.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	46
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	39
Register	-	-	26	-
Total	0	0	26	85
Available	40	40	16000	8000
Utilization (%)	0	0	~0	1

Рис. 6.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● loop_var	-	9~65	-	10 ~ 66	-
○ LOOP_X	no	8 ~ 64	2	-	4~32

Рис. 6.4. Performance profile

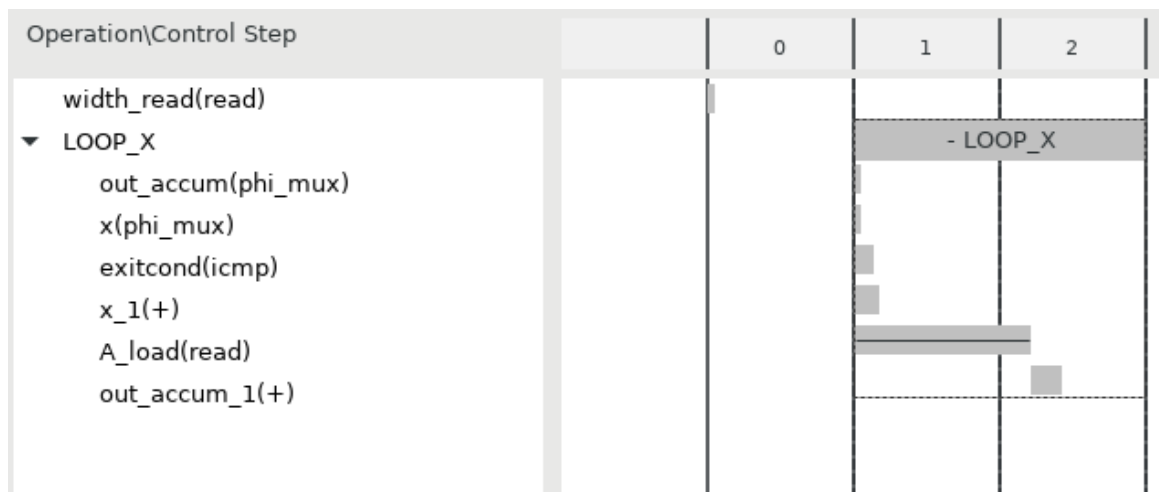


Рис. 6.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	I/O Ports			
2	width	read		
3	A(p0)		read	
4	ap_return		ret	
5	Memory Ports			
6	A(p0)		read	
7	Expressions			
8	out_accum_phi_fu_61		phi_mux	
9	x_phi_fu_73		phi_mux	
10	x_l_fu_85		+	
11	exitcond_fu_80		icmp	
12	out_accum_l_fu_100			+

Рис. 6.6. Resource viewer

В данном случае ничего не изменилось с точки зрения структуры и реализации, однако теперь известно в каком диапазоне находится число итераций цикла и можно рассчитать примерные Latency и II.

7. Решение За

7.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

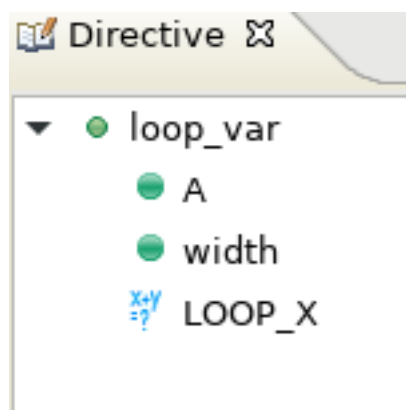


Рис. 7.1. Директивы

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	4.286	0.10

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
?	?	?	?	none

Рис. 7.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	46
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	39
Register	-	-	26	-
Total	0	0	26	85
Available	40	40	16000	8000
Utilization (%)	0	0	~0	1

Рис. 7.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
loop_var	-	-	-	-	-
LOOP_X	no	-	2	-	-

Рис. 7.4. Performance profile

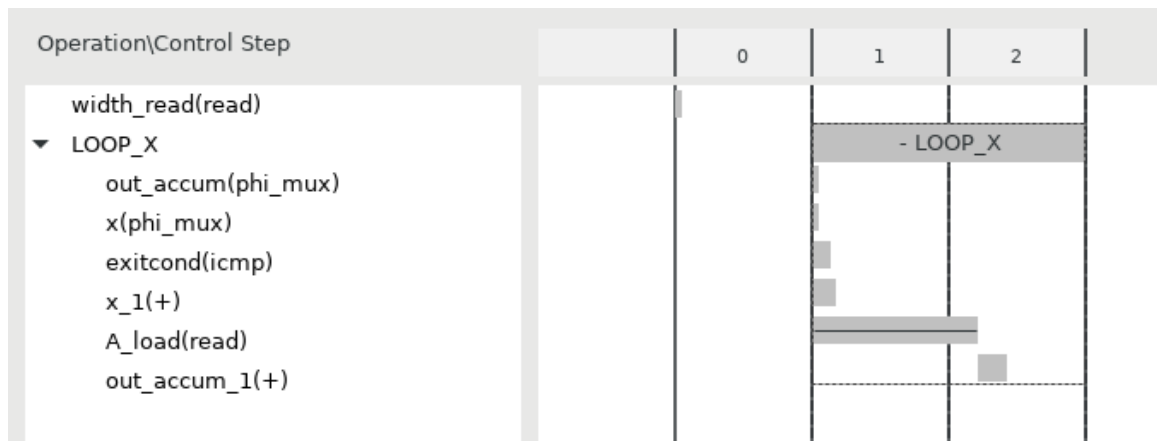


Рис. 7.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	<input type="checkbox"/> I/O Ports			
2	width	read		
3	ap_return		ret	
4	A(p0)		read	
5	<input type="checkbox"/> Memory Ports			
6	A(p0)		read	
7	<input type="checkbox"/> Expressions			
8	x_l_fu_71		+	
9	x_phi_fu_59		phi_mux	
10	out_accum_phi_fu_47		phi_mux	
11	exitcond_fu_66		icmp	
12	out_accum_l_fu_86			+

Рис. 7.6. Resource viewer

Результат не отличается от первого решения.

8. Вывод

В ходе работы была исследована возможность реализации циклов с неизвестным заранее количеством итераций. Для решения проблемы имеются 2 средства: директива `loop_count` и функция `assert`.

Вторая, как выяснилось в ходе исследований, не работает т.к. результат получается, как в случае с решением без применения дополнительных действий.