

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

Лабораторная №7

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Конвейеризация вычислений

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	4
3. Скрипт	4
4. Решение 1	5
4.1. Директивы	5
4.2. Синтез	5
4.3. C/RTL моделирование	8
5. Решение 2	8
5.1. Директивы	8
5.2. Синтез	9
5.3. C/RTL моделирование	12
6. Решение 3	12
6.1. Директивы	12
6.2. Синтез	13
6.3. C/RTL моделирование	16
7. Вывод	16

1. Задание

1. Создать проект lab7_3
2. Микросхема: xa7a12tcsg325-1q
3. Создать функцию на основе приведенного ниже слайда.

```
void foo_top (in1, in2, *out1_data...){  
    accum=0;  
    ...  
    L1:for(i=1;i<N;i++) {  
        accum = accum + in1 + in2;  
    }  
  
    *out1_data = accum;  
}
```

4. Создать тест lab7_3_test.c для проверки функций выше.
 - осуществить моделирование (с выводом результатов в консоль)
5. Сделать свой solution (для варианта без конвейеризации, с конвейеризацией, с конвейеризацией и rewind)
 - задать: clock period 10; clock_uncertainty 0.1
 - осуществить синтез
 - привести в отчете:
 - * performance estimates=>summary
 - * utilization estimates=>summary
 - * 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 (для каждого варианта задания директивы)
 - Привести результаты из консоли
 - Открыть временную диаграмму (все сигналы)
 - * Отобразить два цикла обработки на одном экране
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
 - 6. Выводы
 - Привести обобщенную таблицу зависимости utilization и performance от каждого варианта: без конвейеризации, с конвейеризацией, с конвейеризацией и rewind.
 - Объяснить отличие процедур обращения к элементам массива для каждого случая

2. Исходный код

Ниже приведен исходный код устройства и теста.

```
1 void lab7_3 (int in1, int in2, int *out_data) {
2     int i;
3     static int acc = 0;
4
5     L1: for(i = 0; i < 20; i++) {
6         acc = acc + in1 + in2;
7     }
8     *out_data = acc;
9 }
```

Рис. 2.1. Исходный код устройства

```
1 #include <stdio.h>
2
3 int main() {
4
5     int in1 = 5;
6     int in2 = 8;
7
8     int out = 0; int *o_p = &out;
9     int exp_out = 260;
10
11     lab7_3(in1, in2, o_p);
12
13     printf("Out_%d==Exp_%d\n", out, exp_out);
14     if (out != exp_out) {
15         fprintf(stdout, "-----ERROR-----\n");
16         return -1;
17     } else {
18         fprintf(stdout, "-----Test_Pass-----\n");
19         return 0;
20     }
21 }
```

Рис. 2.2. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```

1 open_project -reset lab7_3
2 add_files lab7_3.c
3 set_top lab7_3
4 add_files -tb lab7_3_test.c
5
6 set solutions [list 1 2 3]
7
8 foreach sol $solutions {
9     open_solution solution_$sol -reset
10    set_part {xa7a12tcsg325-1q}
11    create_clock -period 10ns
12    set_clock_uncertainty 0.1
13
14    if {$sol == 2} {
15        set_directive_pipeline "lab7_3/L1"
16    }
17    if {$sol == 3} {
18        set_directive_pipeline -rewind "lab7_3/L1"
19    }
20
21    csim_design
22    csynth_design
23    cosim_design -trace_level all
24 }
25
26 exit

```

Рис. 3.1. Скрипт

4. Решение 1

4.1. Директивы

В данном решении были установлены директивы, приведённые ниже.

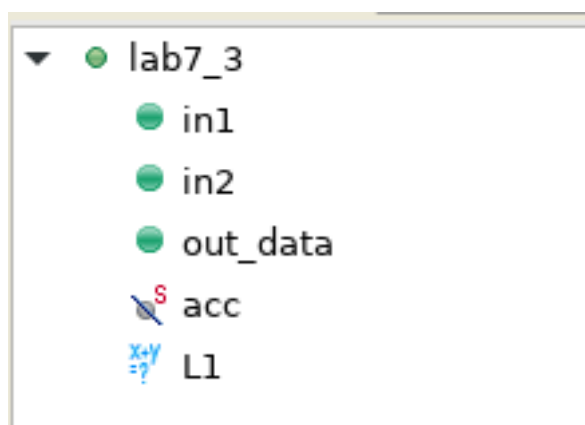


Рис. 4.1. Директивы

4.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	2.702	0.10

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
21	21	21	21	none

Рис. 4.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	104
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	24
Register	-	-	71	-
Total	0	0	71	128
Available	40	40	16000	8000
Utilization (%)	0	0	~0	1

Рис. 4.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab7_3	-	21	-	22	-
L1	no	20	1	-	20

Рис. 4.4. Performance profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	lab7_3	return value
ap_rst	in	1	ap_ctrl_hs	lab7_3	return value
ap_start	in	1	ap_ctrl_hs	lab7_3	return value
ap_done	out	1	ap_ctrl_hs	lab7_3	return value
ap_idle	out	1	ap_ctrl_hs	lab7_3	return value
ap_ready	out	1	ap_ctrl_hs	lab7_3	return value
in1	in	32	ap_none	in1	scalar
in2	in	32	ap_none	in2	scalar
out_data	out	32	ap_vld	out_data	pointer
out_data_ap_vld	out	1	ap_vld	out_data	pointer

Рис. 4.5. Interface estimates

По списку сигналов в проекте видно, что для заданного порта установлен протокол ap_bus. Также видно, что для этого протокола требуются дополнительные сигналы.



Рис. 4.6. Scheduler viewer

	Resource\Control Step	C0	C1
1	I/O Ports		
2	in2	read	
3	in1	read	
4	out_data		write
5	Expressions		
6	tmp1_fu_67	+	
7	i_phi_fu_55		phi_mux
8	i_l_fu_79		+
9	tmp_1_fu_85		+
10	exitcond_fu_73		icmp

Рис. 4.7. Resource viewer

4.3. C/RTL моделирование

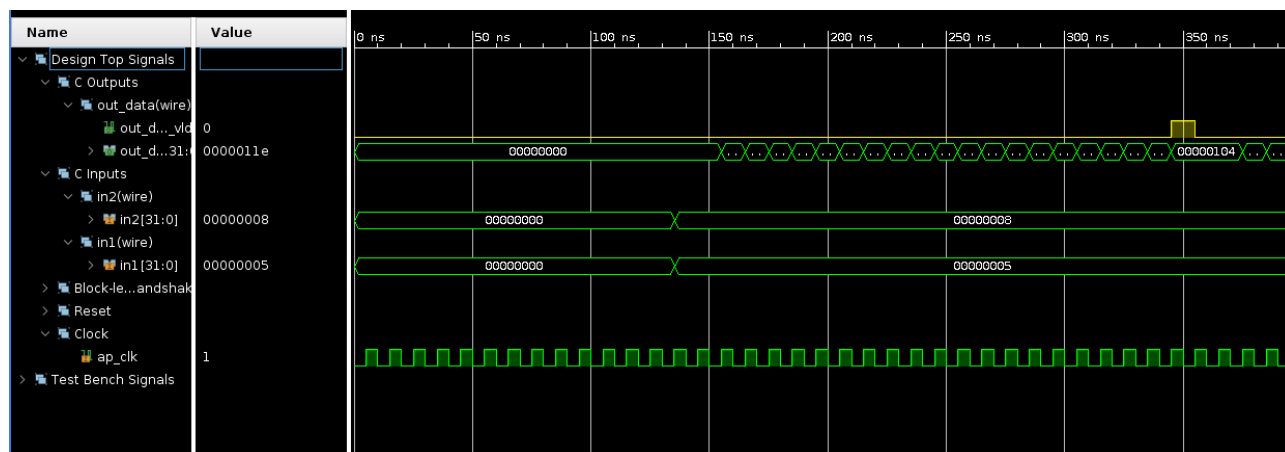


Рис. 4.8. Временная диаграмма

По временной диаграмме видно, что latency составляет 21 такт, а Π – 22 такта.

5. Решение 2

5.1. Директивы

В данном решении были установлены директивы, приведённые ниже.

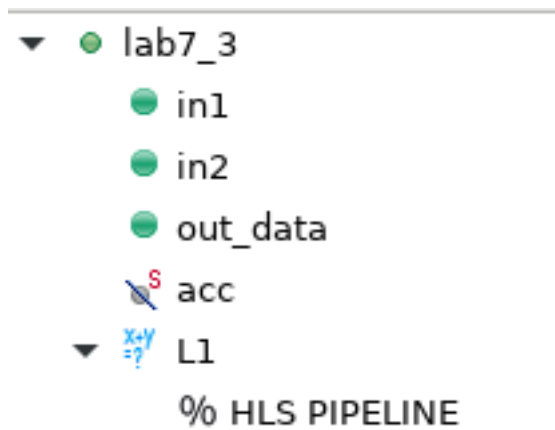


Рис. 5.1. Директивы

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	2.702	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
22	22	22	22	none

Рис. 5.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	104
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	30
Register	-	-	72	-
Total	0	0	72	134
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
lab7_3	-	22	-	23	-
L1	yes	20	1	1	20

Рис. 5.4. Performance profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	lab7_3	return value
ap_rst	in	1	ap_ctrl_hs	lab7_3	return value
ap_start	in	1	ap_ctrl_hs	lab7_3	return value
ap_done	out	1	ap_ctrl_hs	lab7_3	return value
ap_idle	out	1	ap_ctrl_hs	lab7_3	return value
ap_ready	out	1	ap_ctrl_hs	lab7_3	return value
in1	in	32	ap_none	in1	scalar
in2	in	32	ap_none	in2	scalar
out_data	out	32	ap_vld	out_data	pointer
out_data_ap_vld	out	1	ap_vld	out_data	pointer

Рис. 5.5. Interface estimates

По списку сигналов в проекте видно, что для заданного порта установлен протокол ap_bus. Также видно, что для этого протокола требуются дополнительные сигналы.

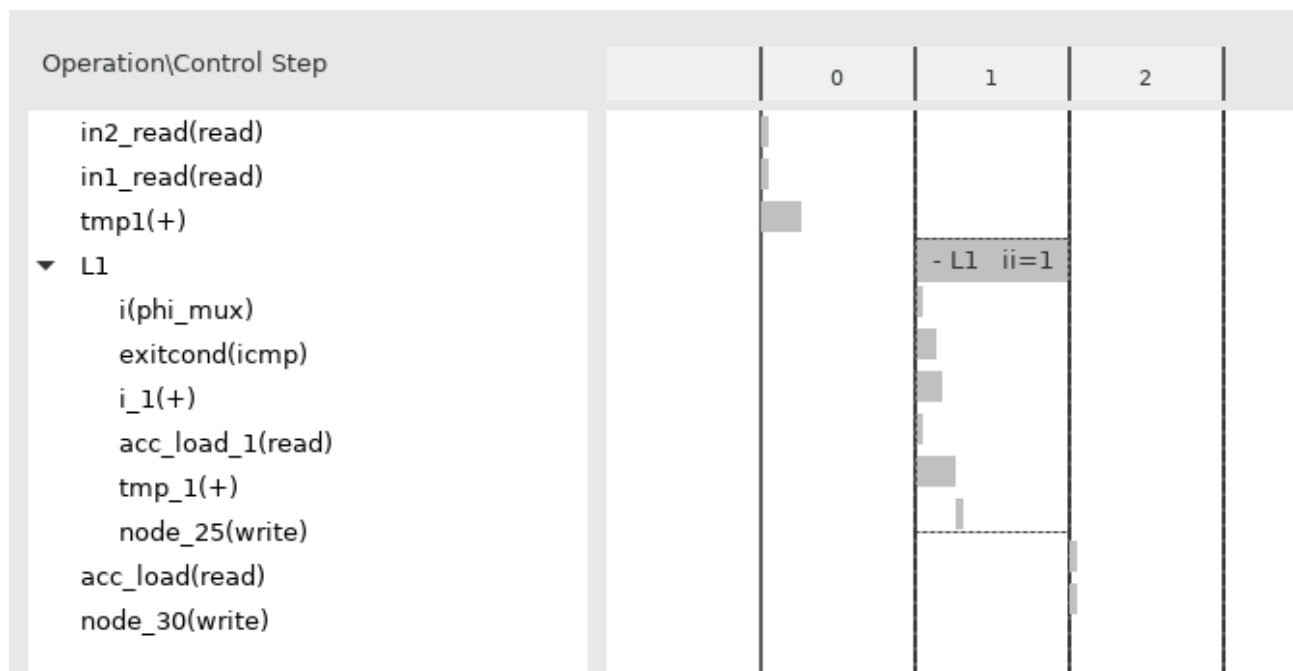


Рис. 5.6. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	I/O Ports			
2	in1	read		
3	in2	read		
4	out_data			write
5	Expressions			
6	tmp1_fu_81	+		
7	i_l_fu_93		+	
8	tmp_l_fu_99		+	
9	i_phi_fu_69		phi_mux	
10	exitcond_fu_87		icmp	

Рис. 5.7. Resource viewer

5.3. C/RTL моделирование

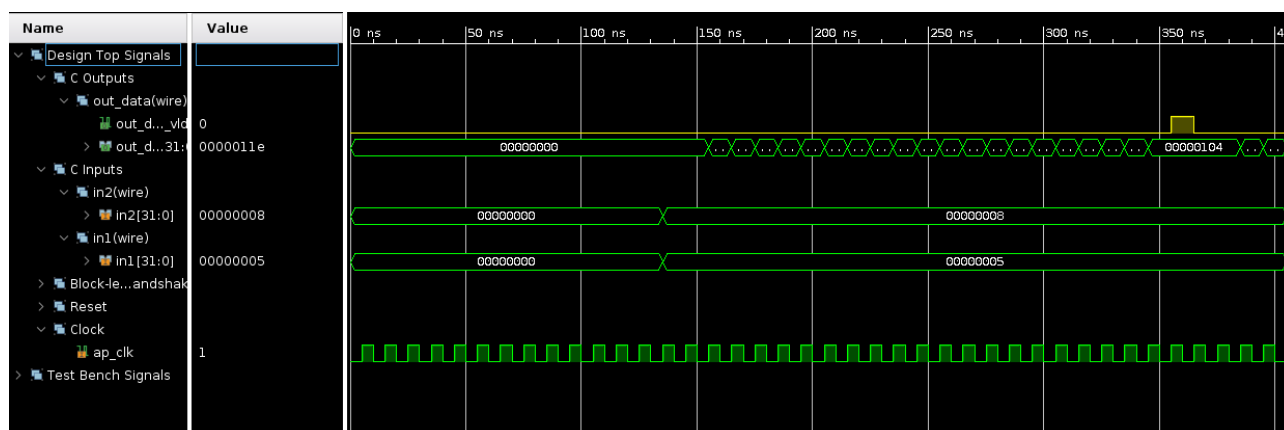


Рис. 5.8. Временная диаграмма

По временной диаграмме видно, что latency составляет 22 такта, а П – 23 такта.

6. Решение 3

6.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

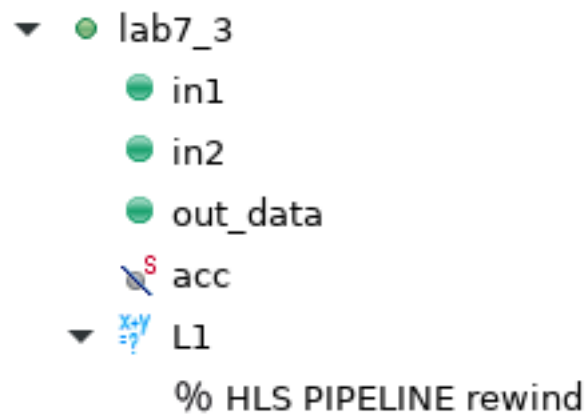


Рис. 6.1. Директивы

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	6.169	0.10

▣ Latency (clock cycles)

▣ Summary

Latency		Interval		
min	max	min	max	Type
19	20	19	20	none

Рис. 6.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	90
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	54
Register	-	-	104	-
Total	0	0	104	144
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
lab7_3	-	19~20	-	20 ~ 21	-
L1	yes	19	1	1	20

Рис. 6.4. Performance profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	lab7_3	return value
ap_rst	in	1	ap_ctrl_hs	lab7_3	return value
ap_start	in	1	ap_ctrl_hs	lab7_3	return value
ap_done	out	1	ap_ctrl_hs	lab7_3	return value
ap_idle	out	1	ap_ctrl_hs	lab7_3	return value
ap_ready	out	1	ap_ctrl_hs	lab7_3	return value
in1	in	32	ap_none	in1	scalar
in2	in	32	ap_none	in2	scalar
out_data	out	32	ap_vld	out_data	pointer
out_data_ap_vld	out	1	ap_vld	out_data	pointer

Рис. 6.5. Interface estimates

По списку сигналов в проекте видно, что для заданного порта установлен протокол ap_bus. Также видно, что для этого протокола требуются дополнительные сигналы.

Operation\Control Step		0	1	
▼ L1				
do_init(phi_mux)				
in12_rewind(phi_mux)				
in23_rewind(phi_mux)				
i1(phi_mux)				
in2_read(read)				
in1_read(read)				
in12_phi(phi_mux)				
in23_phi(phi_mux)				
acc_load(read)				
tmp1(+)				
tmp_1(+)				
node_30(write)				
i(+)				
exitcond icmp)				
node_37(write)				
node_38(return)				

Рис. 6.6. Scheduler viewer

	Resource\Control Step	C0	C1
1	I/O Ports		
2	in2		read
3	in1		read
4	out_data		write
5	Expressions		
6	tmp_1_fu_164		+
7	i_fu_177		+
8	tmp1_fu_158		+
9	in23_phi_phi_fu_145		phi_mux
10	i1_phi_fu_118		phi_mux
11	in23_rewind_phi_fu_104		phi_mux
12	in12_rewind_phi_fu_90		phi_mux
13	in12_phi_phi_fu_132		phi_mux
14	do_init_phi_fu_75		phi_mux
15	exitcond_fu_183		icmp
16	StgValue_32_fu_189		return

Рис. 6.7. Resource viewer

6.3. C/RTL моделирование

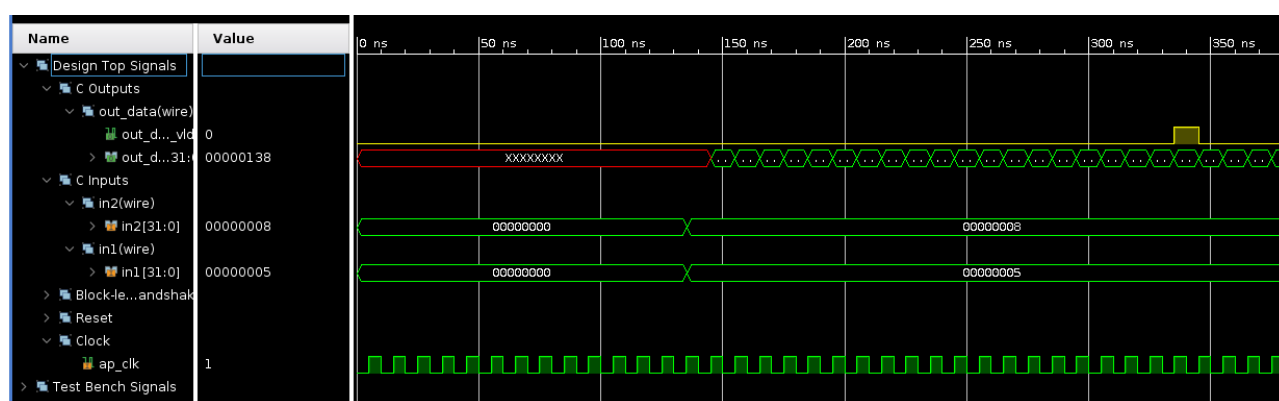


Рис. 6.8. Временная диаграмма

По временной диаграмме видно, что latency составляет 20 тактов, а Π – 21 такт.

7. Вывод

Ниже приведены сравнения производительности и использования ресурсов, в зависимости от решения.

Performance Estimates

▣ Timing (ns)

Clock		solution_1	solution_2	solution_3
ap_clk	Target	10.00	10.00	10.00
	Estimated	2.702	2.702	6.169

▣ Latency (clock cycles)

		solution_1	solution_2	solution_3
Latency	min	21	22	19
	max	21	22	20
Interval	min	21	22	19
	max	21	22	20

Рис. 7.1. Сравнение производительности

Utilization Estimates

	solution_1	solution_2	solution_3
BRAM_18K0	0	0	0
DSP48E	0	0	0
FF	71	72	104
LUT	128	134	144

Рис. 7.2. Сравнение используемых ресурсов

Также можно сказать, что во всех решения реализация доступа к элементам массива не отличается.