

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

Лабораторная №10

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Директива DATA_PACK

Задание 1

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	6
4. Моделирование	7
5. Решение 1а	8
5.1. Директивы	8
5.2. Синтез	9
5.3. C/RTL моделирование	13
6. Решение 2а	13
6.1. Директивы	13
6.2. Синтез	14
6.3. C/RTL моделирование	19
7. Решение 3а	19
7.1. Директивы	19
7.2. Синтез	20
7.3. C/RTL моделирование	25
8. Решение 4а	25
8.1. Директивы	25
8.2. Синтез	26
8.3. C/RTL моделирование	31
9. Вывод	31

1. Задание

1. Создать проект lab10_1
2. Микросхема: xa7a12tcsg325-1q
3. Познакомиться с исходным кодом struct_port.c
4. Познакомиться с исходным кодом struct_port_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 (интерес представляет количество и тип портов)

7. Solution_2a

- задать: clock period 10; clock_uncertainty 0.1
- установить реализацию DATA_PACK
- 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 (интерес представляет количество и тип портов)
8. Сравнить два решения (solution_1a и solution_2a) и сделать выводы: зависимость от DATA_PACK; объяснить (посчитать) число циклов Latency, II...
9. Solution_3a
- задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию DATA_PACK with struct_level
 - 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 (интерес представляет количество и тип портов)
10. Сравнить два решения (solution_2a и solution_3a) и сделать выводы: зависимость от типа интерфейса; объяснить (посчитать) число циклов Latency, II...
11. Solution_4a
- задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию DATA_PACK with field_level
 - 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 (интерес представляет количество и тип портов)
12. Сравнить два решения (solution_3a и solution_4a) и сделать выводы: зависимость от типа интерфейса; объяснить (посчитать) число циклов Latency, II...

2. Исходный код

Ниже приведен исходный код устройства и теста.

```

1 #include "struct_port.h"
2
3 data_t struct_port(
4     data_t i_val,
5     data_t *i_pt,
6     data_t *o_pt
7 ) {
8
9     data_t o_val;
10    int i;
11
12    // Transfer pass-by-value structs
13    o_val.A = i_val.A+2;
14    for (i=0; i<4; i++) {
15        o_val.B[i] = i_val.B[i]+2;
16    }
17
18    // Transfer pointer structs
19    o_pt->A = i_pt->A+3;
20    for (i=0; i<4; i++) {
21        o_pt->B[i] = i_pt->B[i]+3;
22    }
23
24    return o_val;
25 }
```

Рис. 2.1. Исходный код устройства

```

1 #ifndef _STRUCT_PORT_H_
2 #define _STRUCT_PORT_H_
3
4 #include <stdio.h>
5
6 typedef struct {
7     unsigned short A;
8     unsigned char B[4];
9 } data_t;
10
11 data_t struct_port(data_t i_val, data_t *i_pt, data_t *o_pt);
12
13 #endif
```

Рис. 2.2. Заголовочный файл

```

1 #include "struct_port.h"
2
3 int main () {
4     data_t d_ival, d_ipt;
5     data_t d_oval, d_opt;
6
7     int i, retval=0;
8     FILE      *fp;
9
10    // Create input data
11    d_ival.A = 19;
12    d_ipt.A = 29;
13    for (i=0; i<4; i++) {
14        d_ival.B[i] = i+10;
15        d_ipt.B[i] = i+20;
16    }
17
18    // Call the function to operate on the data
19    d_oval = struct_port(d_ival, &d_ipt, &d_opt);
20
21    // Save the results to a file
22    fp=fopen("result.dat","w");
23    fprintf(fp, "Din_Dout\n");
24
25    fprintf(fp, "%d_%d\n", d_oval.A, d_opt.A);
26    for (i=0; i<4; i++) {
27        fprintf(fp, "%d_%d\n", d_oval.B[i], d_opt.B[i]);
28    }
29    fclose(fp);
30
31    // Compare the results file with the golden results
32    retval = system("diff --brief -w result.dat result.golden.dat");
33    if (retval != 0) {
34        printf("Test_failed_!!!\n");
35        retval=1;
36    } else {
37        printf("Test_passed_!\n");
38    }
39
40    // Return 0 if the test passed
41    return retval;
42 }

```

Рис. 2.3. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```

1 open_project -reset lab10_1
2
3 add_files struct_port.c
4 add_files -tb struct_port_test.c
5 add_files -tb result.golden.dat
6 set_top struct_port
7
8 set solutions [list 1a 2a 3a 4a]
9
10 foreach sol $solutions {
11     open_solution solution_$sol -reset
12     set_part {xa7a12tcs325-1q}
13     create_clock -period 10ns
14     set_clock_uncertainty 0.1
15
16     if {$sol == "2a"} {
17         set_directive_data_pack struct_port i_val
18         set_directive_data_pack struct_port i_pt
19         set_directive_data_pack struct_port o_pt
20     }
21     if {$sol == "3a"} {
22         set_directive_data_pack -byte_pad struct_level struct_port i_val
23         set_directive_data_pack -byte_pad struct_level struct_port i_pt
24         set_directive_data_pack -byte_pad struct_level struct_port o_pt
25     }
26     if {$sol == "4a"} {
27         set_directive_data_pack -byte_pad field_level struct_port i_val
28         set_directive_data_pack -byte_pad field_level struct_port i_pt
29         set_directive_data_pack -byte_pad field_level struct_port o_pt
30     }
31
32     csim_design
33     csynth_design
34     cosim_design -trace_level all
35 }
36
37 exit

```

Рис. 3.1. Скрипт

4. Моделирование

Ниже приведены результаты моделирования.

```

INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
  Compiling(apcc) ../../../../struct_port_test.c in debug mode
INFO: [HLS 200-10] Running '/opt/Xilinx/Vivado/2018.2/bin/unwrapped/lnx64.o/apcc'
INFO: [HLS 200-10] For user 'sobol' on host 'gadolinium.local' (Linux_x86_64 version 5.3.12-arch1-1)
12:19:08 MSK 2019
INFO: [HLS 200-10] On os "Arch Linux"
INFO: [HLS 200-10] In directory '/home/sobol/Downloads/labs_from_8/lab10_z1/source/lab10_1/solution_1'
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/1443291575883148925223
INFO: [APCC 202-1] APCC is done.
  Compiling(apcc) ../../../../struct_port.c in debug mode
INFO: [HLS 200-10] Running '/opt/Xilinx/Vivado/2018.2/bin/unwrapped/lnx64.o/apcc'
INFO: [HLS 200-10] For user 'sobol' on host 'gadolinium.local' (Linux_x86_64 version 5.3.12-arch1-1)
12:19:14 MSK 2019
INFO: [HLS 200-10] On os "Arch Linux"
INFO: [HLS 200-10] In directory '/home/sobol/Downloads/labs_from_8/lab10_z1/source/lab10_1/solution_1'
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/1443841575883154086776
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.

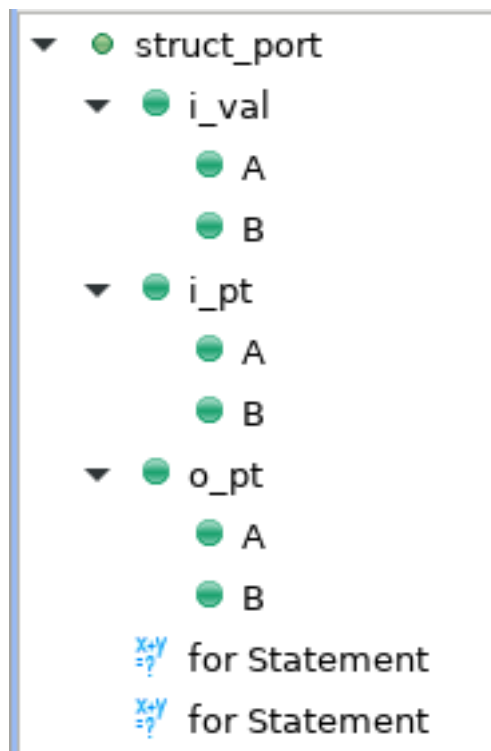


Рис. 5.1. Директивы

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	6.419	0.10

Latency (clock cycles)

Summary

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

Рис. 5.2. Performance estimates

Utilization Estimates				
▣ Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	118
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	116
Register	-	-	71	-
Total	0	0	71	234
Available	40	40	16000	8000
Utilization (%)	0	0	~0	2

Рис. 5.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● struct_port	-	19	-	20	-
● Loop 1	no	8	2	-	4
● Loop 2	no	8	2	-	4

Рис. 5.4. Performance profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	struct_port	return value
ap_rst	in	1	ap_ctrl_hs	struct_port	return value
ap_start	in	1	ap_ctrl_hs	struct_port	return value
ap_done	out	1	ap_ctrl_hs	struct_port	return value
ap_idle	out	1	ap_ctrl_hs	struct_port	return value
ap_ready	out	1	ap_ctrl_hs	struct_port	return value
agg_result_A	out	16	ap_vld	agg_result_A	pointer
agg_result_A_ap_vld	out	1	ap_vld	agg_result_A	pointer
agg_result_B_address0	out	2	ap_memory	agg_result_B	array
agg_result_B_ce0	out	1	ap_memory	agg_result_B	array
agg_result_B_we0	out	1	ap_memory	agg_result_B	array
agg_result_B_d0	out	8	ap_memory	agg_result_B	array
agg_result_B_address1	out	2	ap_memory	agg_result_B	array
agg_result_B_ce1	out	1	ap_memory	agg_result_B	array
agg_result_B_we1	out	1	ap_memory	agg_result_B	array
agg_result_B_d1	out	8	ap_memory	agg_result_B	array
i_val_A	in	16	ap_none	i_val_A	scalar
i_val_B_address0	out	2	ap_memory	i_val_B	array
i_val_B_ce0	out	1	ap_memory	i_val_B	array
i_val_B_q0	in	8	ap_memory	i_val_B	array
i_pt_A	in	16	ap_none	i_pt_A	pointer
i_pt_B_address0	out	2	ap_memory	i_pt_B	array
i_pt_B_ce0	out	1	ap_memory	i_pt_B	array
i_pt_B_q0	in	8	ap_memory	i_pt_B	array
o_pt_A	out	16	ap_vld	o_pt_A	pointer
o_pt_A_ap_vld	out	1	ap_vld	o_pt_A	pointer
o_pt_B_address0	out	2	ap_memory	o_pt_B	array
o_pt_B_ce0	out	1	ap_memory	o_pt_B	array
o_pt_B_we0	out	1	ap_memory	o_pt_B	array
o_pt_B_d0	out	8	ap_memory	o_pt_B	array

Рис. 5.5. Interface estimates

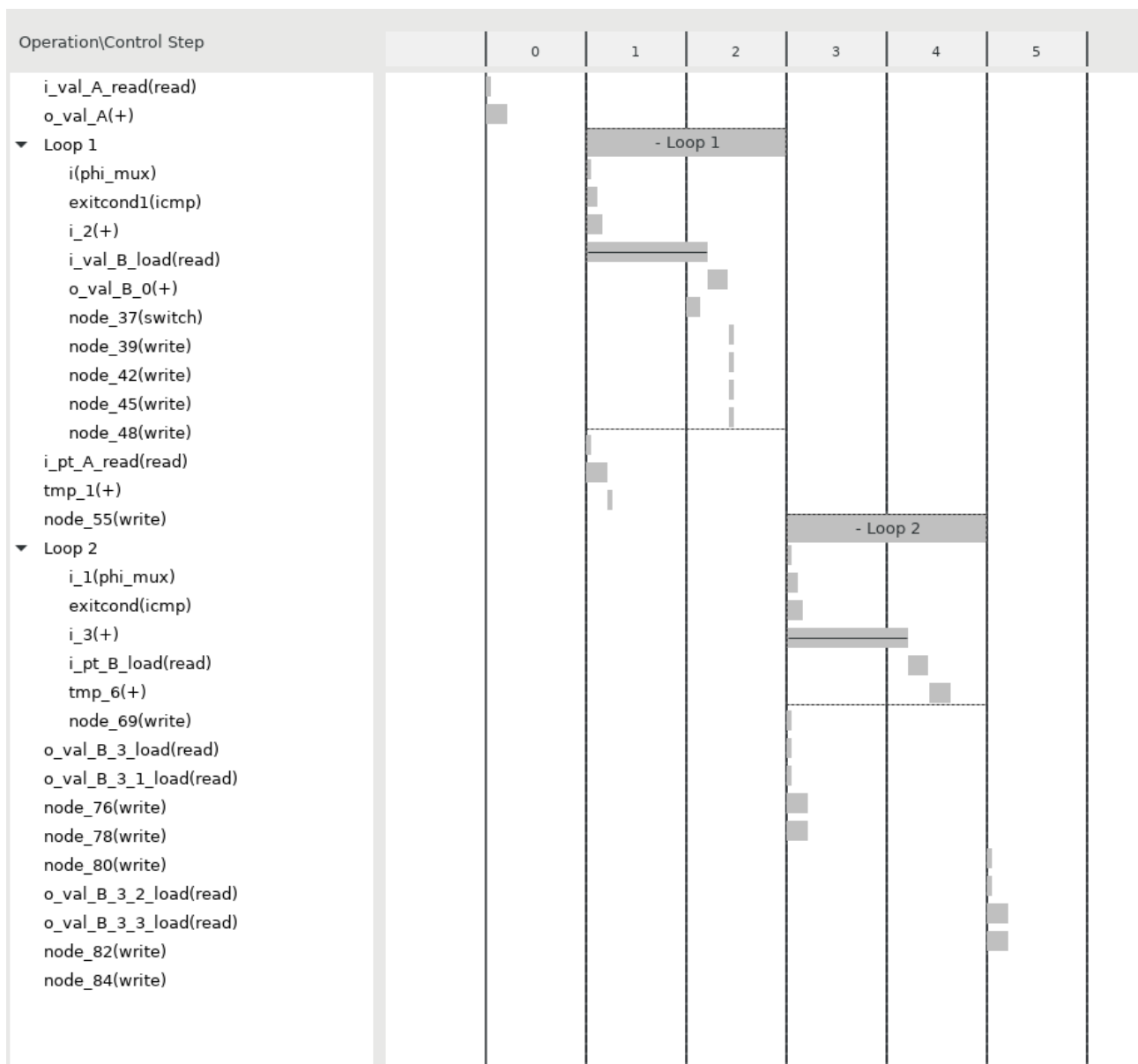


Рис. 5.6. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4	C5
1	I/O Ports						
2	i_val_A	read					
3	o_pt_A		write				
4	i_pt_A		read				
5	i_val_B(p0)		read				
6	agg_result_B(p1)				write		write
7	agg_result_B(p0)				write		write
8	i_pt_B(p0)				read		
9	agg_result_A				write		
10	o_pt_B(p0)					write	
11	Memory Ports						
12	i_val_B(p0)		read				
13	agg_result_B(p1)				write		write
14	agg_result_B(p0)				write		write
15	i_pt_B(p0)				read		
16	o_pt_B(p0)					write	
17	Expressions						
18	o_val_A_fu_210	+					
19	i_phi_fu_192		phi_mux				
20	i_2_fu_222		+				
21	tmp_1_fu_237		+				
22	exitcond1_fu_216		icmp				
23	o_val_B_0_fu_244			+			
24	i_1_phi_fu_203				phi_mux		
25	i_3_fu_276				+		
26	exitcond_fu_270				icmp		
27	tmp_6_fu_295					+	

Рис. 5.7. Resource viewer

5.3. C/RTL моделирование

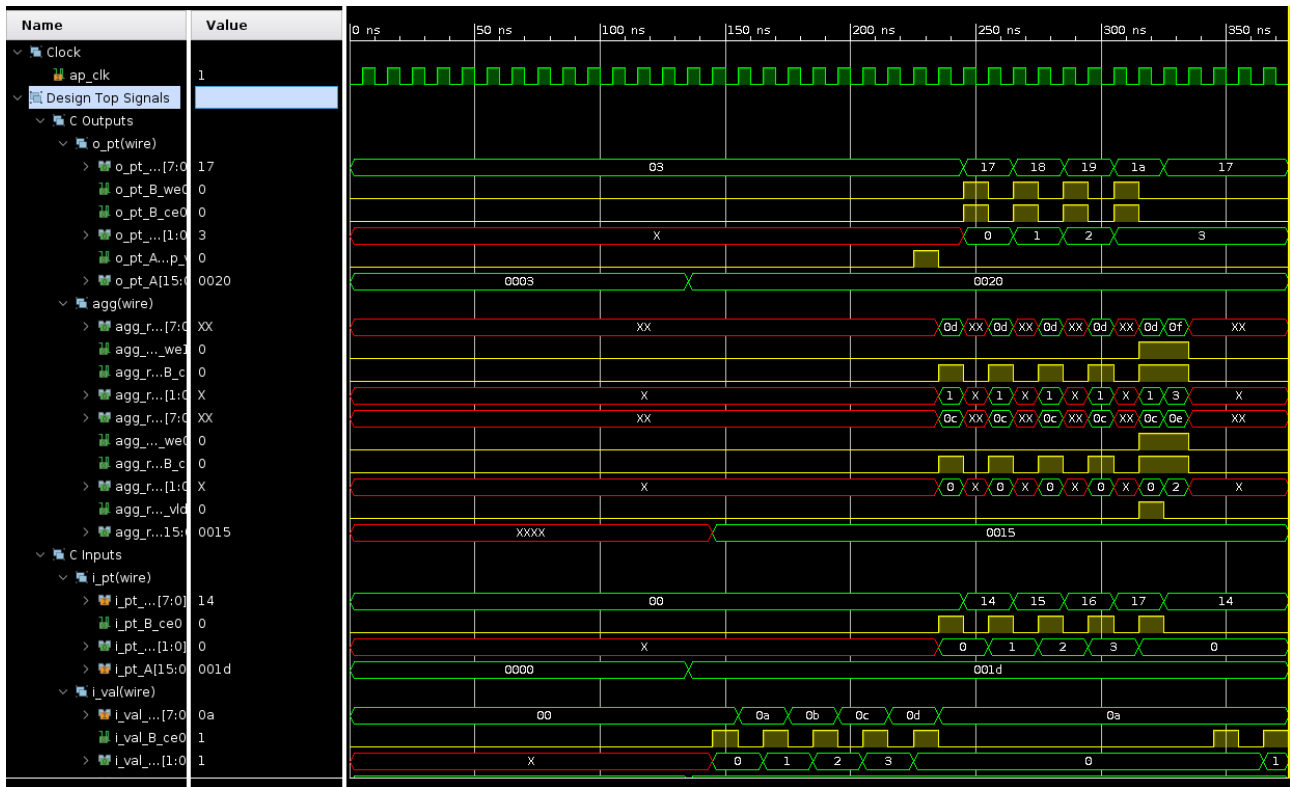


Рис. 5.8. Временная диаграмма

К портам были применены стандартные интерфейсы – `ap_memory`. В функции присутствуют 2 цикла по 4 итерации. Для выполнения итерации первого цикла требуется 2 такта, для выполнения второго – 2 такта. Также нужен 1 такт на подготовку данных и 2 такта для записи результатов, в итоге имеем $\text{Latency} = 4 \cdot 2 + 4 \cdot 2 + 1 + 2 = 19$. Еще через 1 такт данные будут готовы на выходе $\Pi = 20$.

6. Решение 2a

6.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

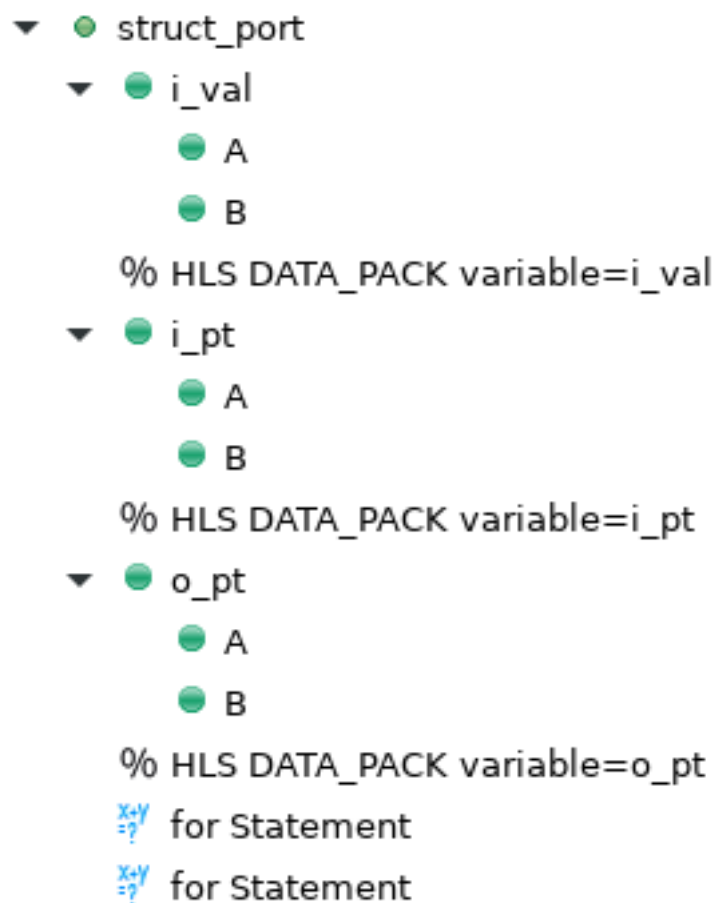


Рис. 6.1. Директивы

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	8.331	0.10

☐ Latency (clock cycles)

☐ Summary

Latency		Interval		
min	max	min	max	Type
6	6	6	6	none

Рис. 6.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	508
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	90
Register	-	-	54	-
Total	0	0	54	598
Available	40	40	16000	8000
Utilization (%)	0	0	~0	7

Рис. 6.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● struct_port	-	6	-	7	-
● Loop 1	no	4	1	-	4

Рис. 6.4. Performance profile

Interface

□ Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	struct_port	return value
ap_rst	in	1	ap_ctrl_hs	struct_port	return value
ap_start	in	1	ap_ctrl_hs	struct_port	return value
ap_done	out	1	ap_ctrl_hs	struct_port	return value
ap_idle	out	1	ap_ctrl_hs	struct_port	return value
ap_ready	out	1	ap_ctrl_hs	struct_port	return value
agg_result_A	out	16	ap_vld	agg_result_A	pointer
agg_result_A_ap_vld	out	1	ap_vld	agg_result_A	pointer
agg_result_B_address0	out	2	ap_memory	agg_result_B	array
agg_result_B_ce0	out	1	ap_memory	agg_result_B	array
agg_result_B_we0	out	1	ap_memory	agg_result_B	array
agg_result_B_d0	out	8	ap_memory	agg_result_B	array
agg_result_B_address1	out	2	ap_memory	agg_result_B	array
agg_result_B_ce1	out	1	ap_memory	agg_result_B	array
agg_result_B_we1	out	1	ap_memory	agg_result_B	array
agg_result_B_d1	out	8	ap_memory	agg_result_B	array
i_val	in	48	ap_none	i_val	scalar
i_pt	in	48	ap_none	i_pt	pointer
o_pt	out	48	ap_vld	o_pt	pointer
o_pt_ap_vld	out	1	ap_vld	o_pt	pointer

Рис. 6.5. Interface estimates

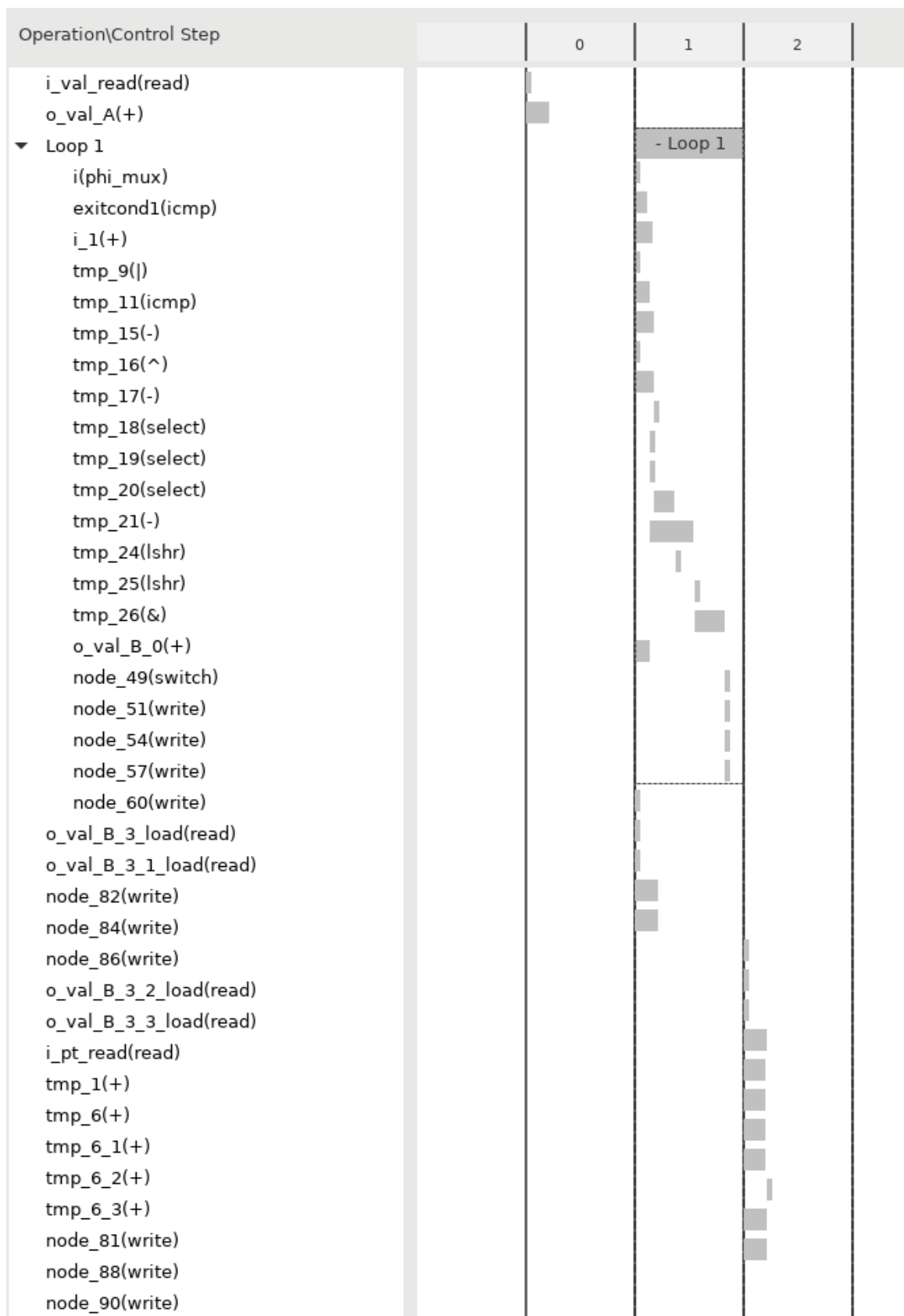


Рис. 6.6. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	I/O Ports			
2	i_val	read		
3	agg_result_A		write	
4	agg_result_B(p0)		write	write
5	agg_result_B(p1)		write	write
6	o_pt			write
7	i_pt			read
8	Memory Ports			
9	agg_result_B(p1)		write	write
10	agg_result_B(p0)		write	write
11	Expressions			
12	o_val_A_fu_190	+		
13	i_l_fu_202		+	
14	o_val_B_0_fu_336		+	
15	i_phi_fu_179		phi_mux	
16	tmp_17_fu_270		-	
17	tmp_21_fu_300		-	
18	tmp_15_fu_258		-	
19	tmp_19_fu_284		select	
20	tmp_20_fu_292		select	
21	tmp_18_fu_276		select	
22	tmp_9_fu_229			
23	tmp_24_fu_314		lshr	
24	tmp_25_fu_320		lshr	
25	tmp_26_fu_326		&	
26	tmp_16_fu_264		^	
27	tmp_11_fu_235		icmp	
28	exitcond1_fu_196		icmp	
29	tmp_6_2_fu_430			+
30	tmp_6_3_fu_446			+
31	tmp_1_fu_382			+
32	tmp_6_1_fu_414			+
33	tmp_6_fu_398			+

Рис. 6.7. Resource viewer

6.3. C/RTL моделирование

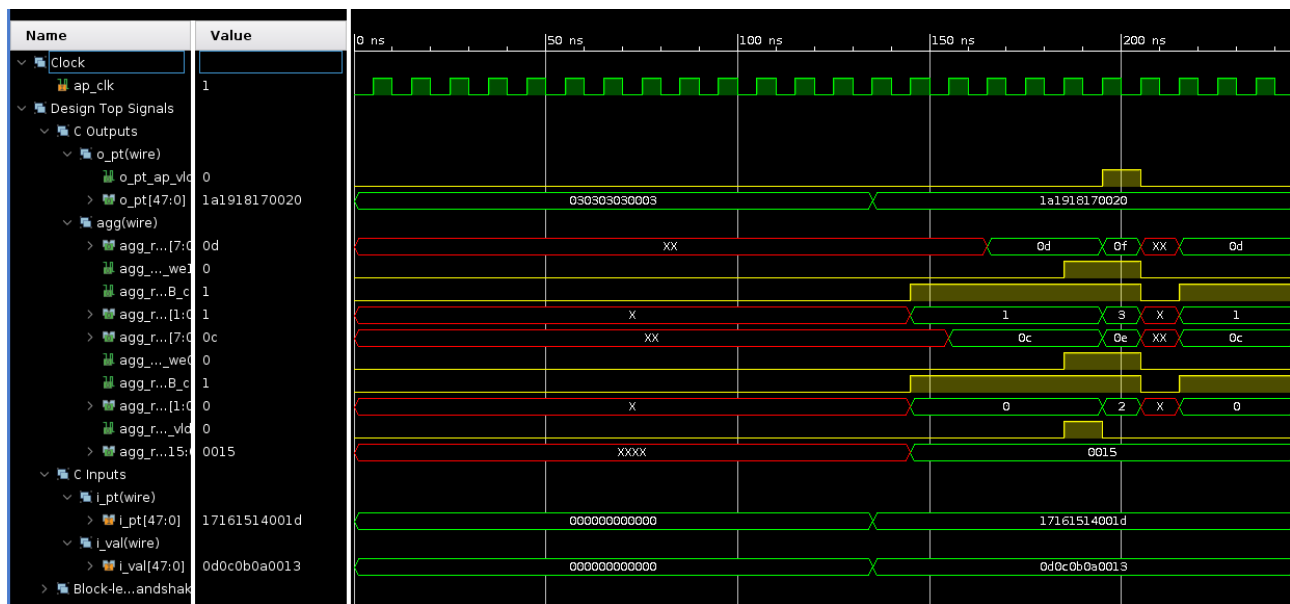


Рис. 6.8. Временная диаграмма

Как видно, директива успешно применилась к аргументами функции сделав из них один порт длиной 48 бит, это позволило получить параллельный доступ ко всем элементам структуры. Однако структуру `o_val` развернуть не удалось т.к. мы не можем применить директиву к `return`. В итоге не получилось распараллелить первый цикл и значение $\text{Latency} = 1 \cdot 4 + 1$ на подготовку + 1 на запись = 6 тактов. $\text{Initiation interval} = \text{Latency} + 1 = 7$ тактов.

7. Решение За

7.1. Директивы

В данном решении были установлены директивы, приведённые ниже.

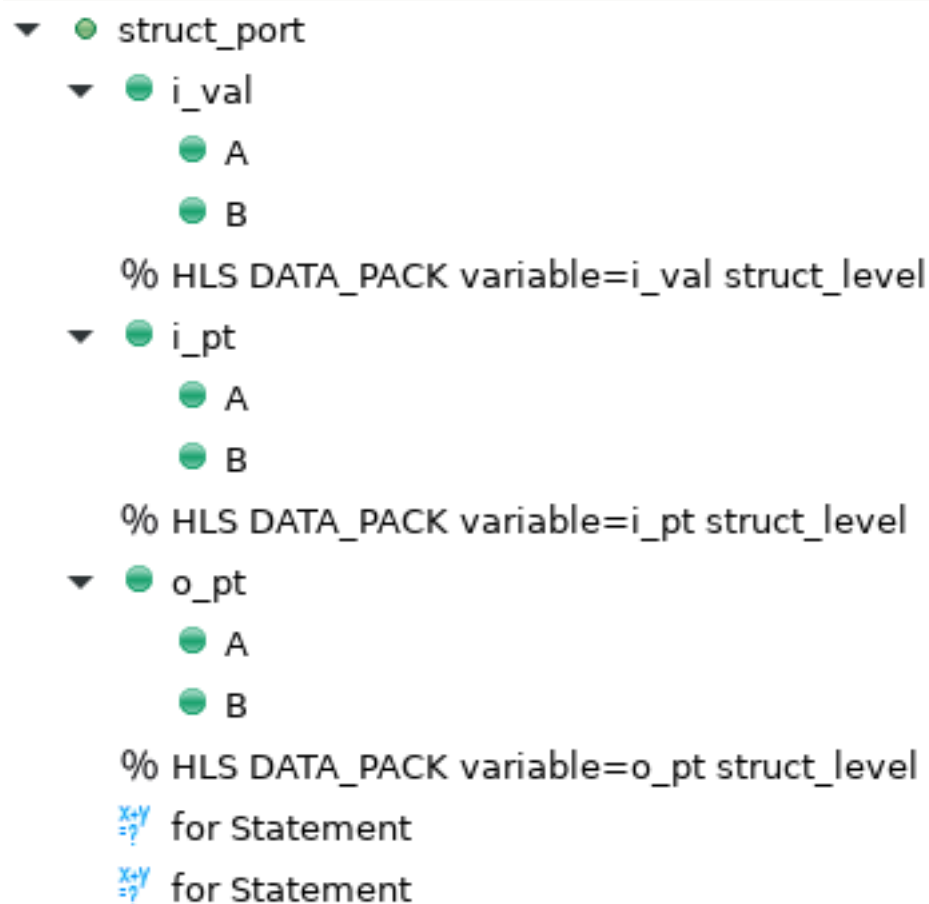


Рис. 7.1. Директивы

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	8.331	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
6	6	6	6	none

Рис. 7.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	508
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	90
Register	-	-	54	-
Total	0	0	54	598
Available	40	40	16000	8000
Utilization (%)	0	0	~0	7

Рис. 7.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● struct_port	-	6	-	7	-
○ Loop 1	no	4	1	-	4

Рис. 7.4. Performance profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	struct_port	return value
ap_rst	in	1	ap_ctrl_hs	struct_port	return value
ap_start	in	1	ap_ctrl_hs	struct_port	return value
ap_done	out	1	ap_ctrl_hs	struct_port	return value
ap_idle	out	1	ap_ctrl_hs	struct_port	return value
ap_ready	out	1	ap_ctrl_hs	struct_port	return value
agg_result_A	out	16	ap_vld	agg_result_A	pointer
agg_result_A_ap_vld	out	1	ap_vld	agg_result_A	pointer
agg_result_B_address0	out	2	ap_memory	agg_result_B	array
agg_result_B_ce0	out	1	ap_memory	agg_result_B	array
agg_result_B_we0	out	1	ap_memory	agg_result_B	array
agg_result_B_d0	out	8	ap_memory	agg_result_B	array
agg_result_B_address1	out	2	ap_memory	agg_result_B	array
agg_result_B_ce1	out	1	ap_memory	agg_result_B	array
agg_result_B_we1	out	1	ap_memory	agg_result_B	array
agg_result_B_d1	out	8	ap_memory	agg_result_B	array
i_val	in	48	ap_none	i_val	scalar
i_pt	in	48	ap_none	i_pt	pointer
o_pt	out	48	ap_vld	o_pt	pointer
o_pt_ap_vld	out	1	ap_vld	o_pt	pointer

Рис. 7.5. Interface estimates

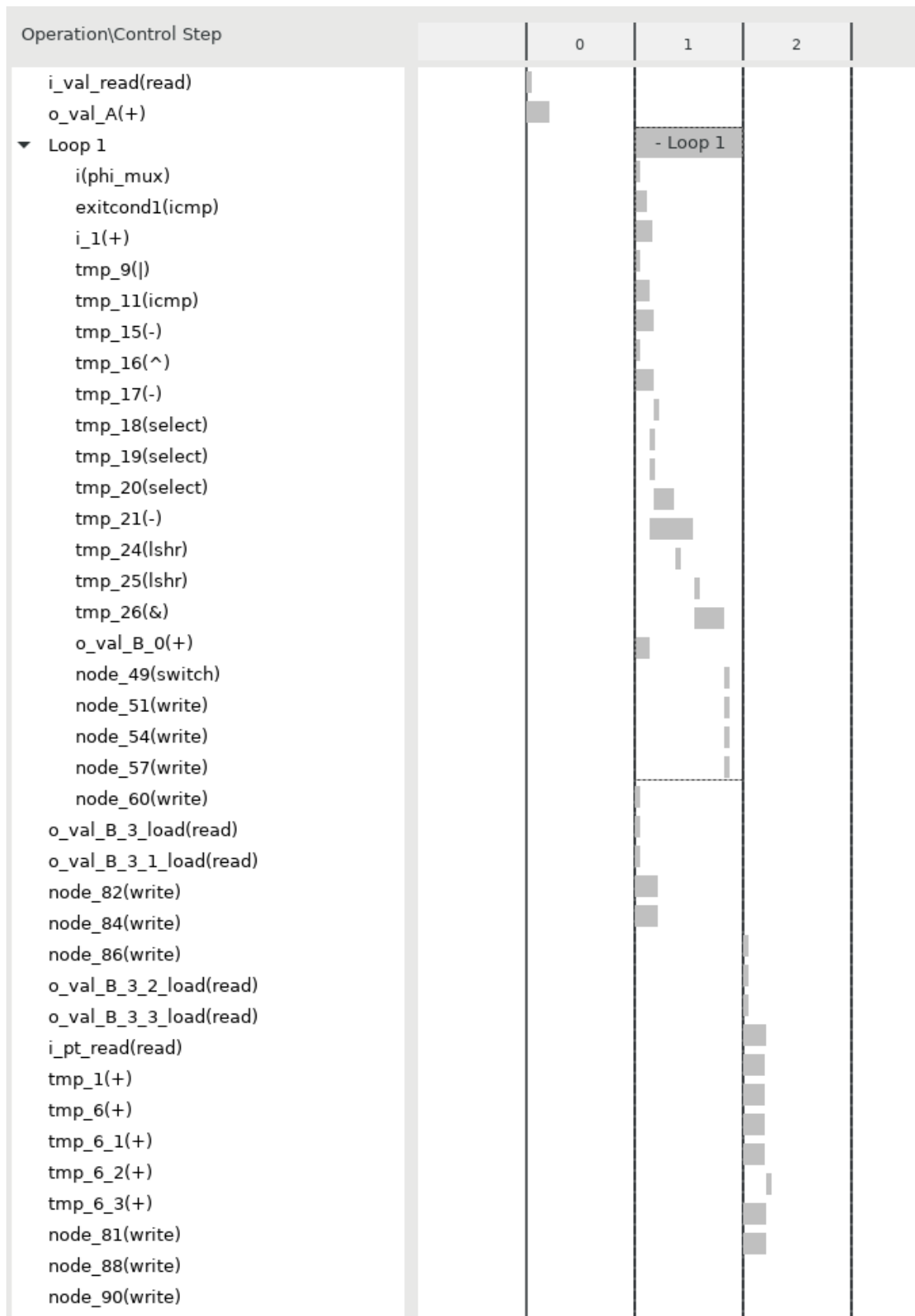


Рис. 7.6. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	I/O Ports			
2	i_val	read		
3	agg_result_B(p1)		write	write
4	agg_result_B(p0)		write	write
5	agg_result_A		write	
6	i_pt			read
7	o_pt			write
8	Memory Ports			
9	agg_result_B(p0)		write	write
10	agg_result_B(p1)		write	write
11	Expressions			
12	o_val_A_fu_190	+		
13	o_val_B_0_fu_336		+	
14	i_l_fu_202		+	
15	i_phi_fu_179		phi_mux	
16	tmp_15_fu_258		-	
17	tmp_21_fu_300		-	
18	tmp_17_fu_270		-	
19	tmp_25_fu_320		lshr	
20	tmp_24_fu_314		lshr	
21	tmp_18_fu_276		select	
22	tmp_20_fu_292		select	
23	tmp_19_fu_284		select	
24	tmp_9_fu_229			
25	tmp_26_fu_326		&	
26	tmp_16_fu_264		^	
27	tmp_11_fu_235		icmp	
28	exitcond1_fu_196		icmp	
29	tmp_1_fu_382			+
30	tmp_6_3_fu_446			+
31	tmp_6_1_fu_414			+
32	tmp_6_2_fu_430			+
33	tmp_6_fu_398			+

Рис. 7.7. Resource viewer

7.3. C/RTL моделирование

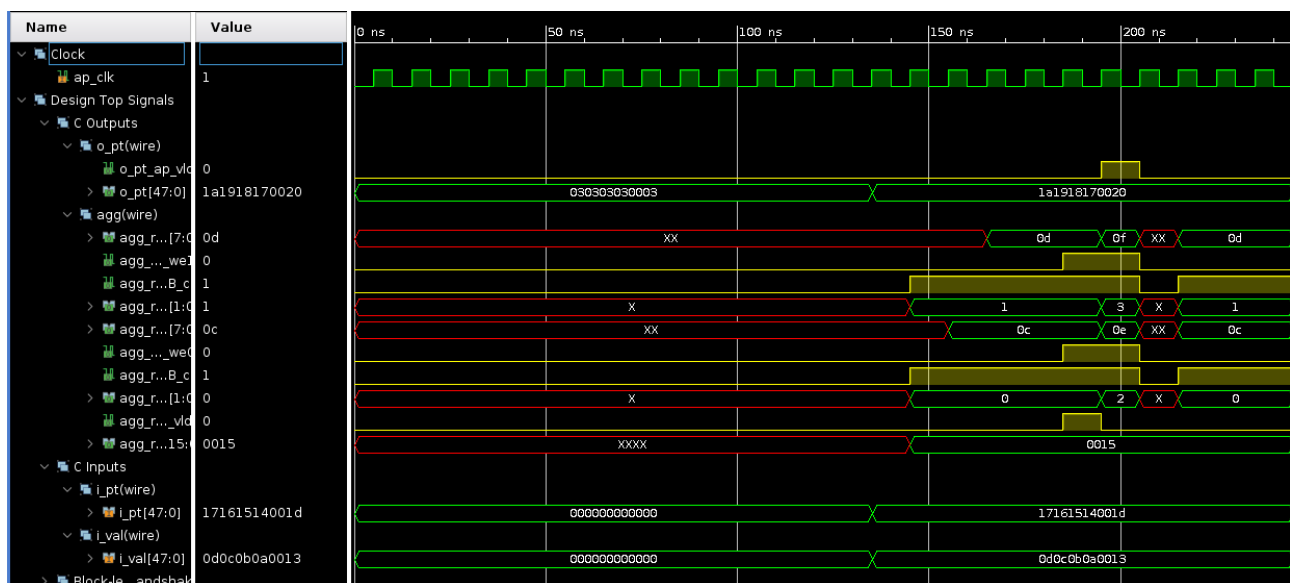


Рис. 7.8. Временная диаграмма

Данное решение полностью совпадает с предыдущим.

8. Решение 4а

8.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

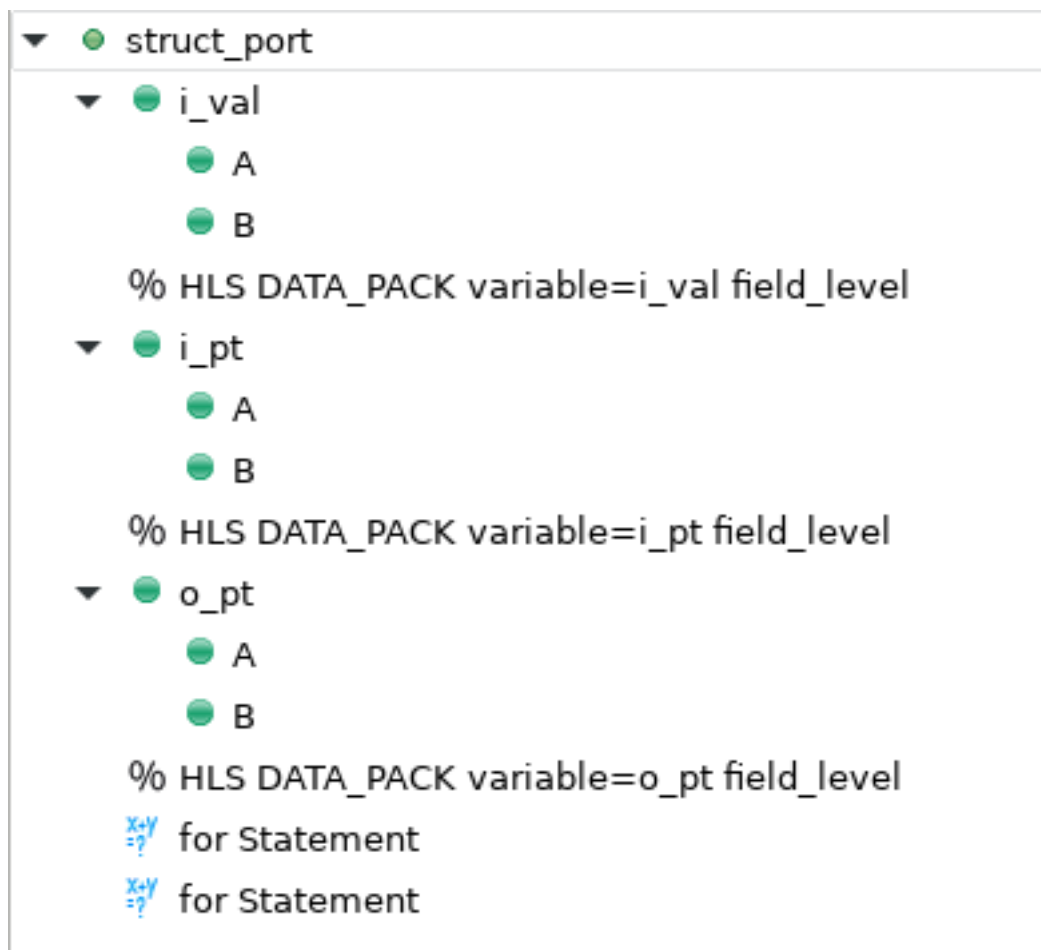


Рис. 8.1. Директивы

8.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.331	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
6	6	6	6	none

Рис. 8.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	508
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	90
Register	-	-	54	-
Total	0	0	54	598
Available	40	40	16000	8000
Utilization (%)	0	0	~0	7

Рис. 8.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● struct_port	-	6	-	7	-
○ Loop 1	no	4	1	-	4

Рис. 8.4. Performance profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	struct_port	return value
ap_rst	in	1	ap_ctrl_hs	struct_port	return value
ap_start	in	1	ap_ctrl_hs	struct_port	return value
ap_done	out	1	ap_ctrl_hs	struct_port	return value
ap_idle	out	1	ap_ctrl_hs	struct_port	return value
ap_ready	out	1	ap_ctrl_hs	struct_port	return value
agg_result_A	out	16	ap_vld	agg_result_A	pointer
agg_result_A_ap_vld	out	1	ap_vld	agg_result_A	pointer
agg_result_B_address0	out	2	ap_memory	agg_result_B	array
agg_result_B_ce0	out	1	ap_memory	agg_result_B	array
agg_result_B_we0	out	1	ap_memory	agg_result_B	array
agg_result_B_d0	out	8	ap_memory	agg_result_B	array
agg_result_B_address1	out	2	ap_memory	agg_result_B	array
agg_result_B_ce1	out	1	ap_memory	agg_result_B	array
agg_result_B_we1	out	1	ap_memory	agg_result_B	array
agg_result_B_d1	out	8	ap_memory	agg_result_B	array
i_val	in	48	ap_none	i_val	scalar
i_pt	in	48	ap_none	i_pt	pointer
o_pt	out	48	ap_vld	o_pt	pointer
o_pt_ap_vld	out	1	ap_vld	o_pt	pointer

Рис. 8.5. Interface estimates

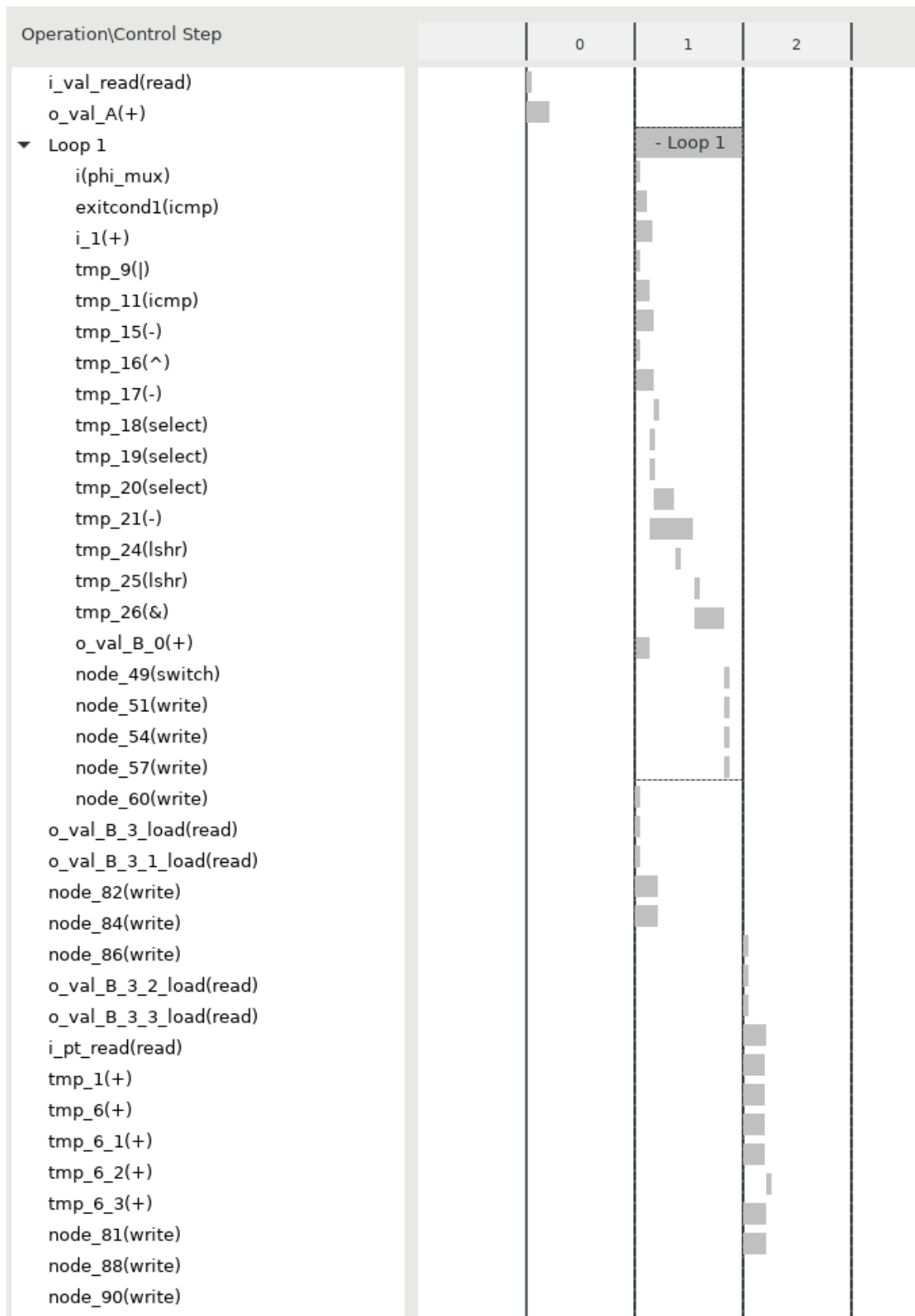


Рис. 8.6. Scheduler viewer

	Resource\Control Step	C0	C1	C2
1	I/O Ports			
2	i_val	read		
3	agg_result_B(p1)		write	write
4	agg_result_B(p0)		write	write
5	agg_result_A		write	
6	i_pt			read
7	o_pt			write
8	Memory Ports			
9	agg_result_B(p0)		write	write
10	agg_result_B(p1)		write	write
11	Expressions			
12	o_val_A_fu_190	+		
13	o_val_B_0_fu_336		+	
14	i_l_fu_202		+	
15	i_phi_fu_179		phi_mux	
16	tmp_15_fu_258		-	
17	tmp_21_fu_300		-	
18	tmp_17_fu_270		-	
19	tmp_25_fu_320		lshr	
20	tmp_24_fu_314		lshr	
21	tmp_18_fu_276		select	
22	tmp_20_fu_292		select	
23	tmp_19_fu_284		select	
24	tmp_9_fu_229			
25	tmp_26_fu_326		&	
26	tmp_16_fu_264		^	
27	tmp_11_fu_235		icmp	
28	exitcond1_fu_196		icmp	
29	tmp_1_fu_382			+
30	tmp_6_3_fu_446			+
31	tmp_6_1_fu_414			+
32	tmp_6_2_fu_430			+
33	tmp_6_fu_398			+

Рис. 8.7. Resource viewer

8.3. C/RTL моделирование

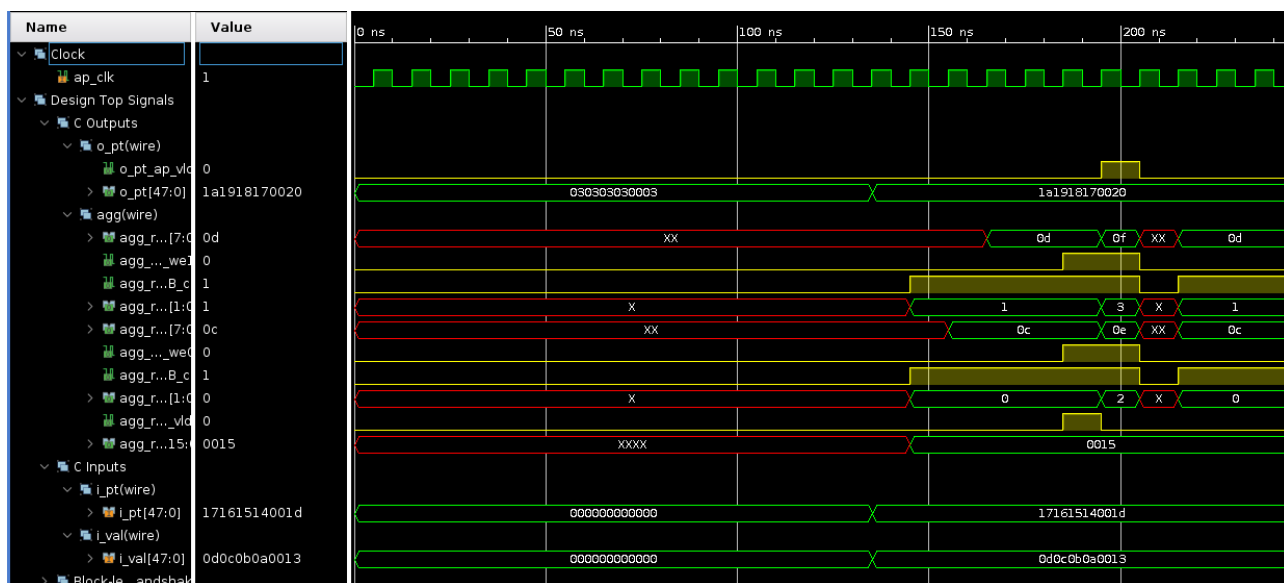


Рис. 8.8. Временная диаграмма

Данное решение полностью совпадает с предыдущим.

9. Вывод

В ходе работы было исследовано влияние директивы DATA_PACK на функции использующие структуры. Данная директива позволяет развернуть структуру в один порт соответствующей длины, однако это требует большого количества ресурсов. Изменения параметра byte_rad в данной лабораторной работе не возымело никакого эффекта в связи с тем, что исследуемая функция оказалась непоказательной.