

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

Лабораторная №14

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Указатели

Задание 5

**Студенты:**

Соболь В.

Темнова А.С.

Группа: 13541/3

**Преподаватель:**

Антонов А.П.

Санкт-Петербург  
2019

# Содержание

<b>1. Задание</b>	<b>3</b>
<b>2. Исходный код</b>	<b>5</b>
<b>3. Скрипт</b>	<b>6</b>
<b>4. Моделирование</b>	<b>7</b>
<b>5. Решение 1а</b>	<b>8</b>
5.1. Директивы . . . . .	8
5.2. Синтез . . . . .	8
5.3. C/RTL моделирование . . . . .	11
<b>6. Решение 2а</b>	<b>11</b>
6.1. Директивы . . . . .	11
6.2. Синтез . . . . .	11
6.3. C/RTL моделирование . . . . .	14
<b>7. Решение 3а</b>	<b>14</b>
7.1. Директивы . . . . .	14
7.2. Синтез . . . . .	14
7.3. C/RTL моделирование . . . . .	17
<b>8. Решение 4а</b>	<b>17</b>
8.1. Директивы . . . . .	17
8.2. Синтез . . . . .	18
8.3. C/RTL моделирование . . . . .	21
<b>9. Вывод</b>	<b>21</b>

# 1. Задание

1. Создать проект lab14\_5
2. Микросхема: xa7a12tcsg325-1q
3. В папке source текст функции pointer\_double

*Познакомьтесь с ним.*

4. Познакомьтесь с тестом.

5. Исследование:

6. Solution\_1a

- Осуществить моделирование
- 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

Обратить внимание на реализацию интерфейсов.

7. Solution\_2a

- задать: clock period 10; clock\_uncertainty 0.1
- установить реализацию КОНВЕЙЕР ДЛЯ 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
8. Сравнить два решения (solution\_1a и solution\_2a) и сделать выводы
9. Solution\_3a
- задать: clock period 10; clock\_uncertainty 0.1
  - установить реализацию INLINE ДЛЯ ФУНКЦИИ
  - 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\_1a и solution\_2a и solution\_3a) и сделать выводы
11. Solution\_4a
- задать: clock period 10; clock\_uncertainty 0.1
  - установить реализацию ПРЕДЛОЖИТЕ ДЛЯ УВЕЛИЧЕНИЯ ПРИЗВОДИТЕЛЬНОСТИ И ЗАДЕРЖКИ, + ОПТИМИЗАЦИЯ ИНТЕРФЕЙСА + УМЕНЬШЕНИЕ AREA
  - 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 с полученными ранее решениями: solution\_1a и solution\_2a и solution\_3a и сделать выводы

## 2. Исходный код

Ниже приведен исходный код устройства и теста.

```
1 #include "pointer_double.h"
2
3 data_t sub(data_t ptr[10], data_t size, data_t**flagPtr)
4 {
5     data_t x, i;
6
7     x = 0;
8     // Sum x if AND of local index and double-pointer index is true
9     LOOP: for (i=0; i<size; ++i)
10         if (**flagPtr & i)
11             x += *(ptr+i);
12     return x;
13 }
14
15 data_t pointer_double(data_t pos, data_t x, data_t* flag)
16 {
17     data_t array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
18     data_t* ptrFlag;
19     data_t i;
20
21     ptrFlag = flag;
22
23     // Write x into index position pos
24     if (pos >=0 & pos < 10)
25         *(array+pos) = x;
26
27     // Pass same index (as pos) as pointer to another function
28     return sub(array, 10, &ptrFlag);
29 }
```

Рис. 2.1. Исходный код устройства

```

1 #ifndef _POINTER_DOUBLE_H_
2 #define _POINTER_DOUBLE_H_
3
4 #include <stdio.h>
5
6
7 typedef int data_t;
8
9 data_t pointer_double(data_t pos, data_t x, data_t* flag);
10
11 #endif

```

Рис. 2.2. Заголовочный файл

```

1 #include "pointer_double.h"
2
3 int main () {
4     data_t data_o;
5
6     int i, retval=0;
7     FILE *fp;
8
9     // Save the results to a file
10    fp=fopen("result.dat", "w");
11
12    // Call the function for multiple transactions
13    for(i=0; i<10;++i) {
14        int flag = i;
15        data_o = pointer_double(i, -1, &flag);
16        fprintf(fp, "%d_\n", data_o);
17    }
18    fclose(fp);
19
20    // Compare the results file with the golden results
21    retval = system("diff --brief -w result.dat result.golden.dat");
22    if (retval != 0) {
23        printf("Test_failed_!!!\n");
24        retval=1;
25    } else {
26        printf("Test_passed_!\n");
27    }
28
29    // Return 0 if the test passed
30    return retval;
31 }

```

Рис. 2.3. Исходный код теста

### 3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```

1 open_project -reset lab14_5
2
3 add_files pointer_double.c
4 add_files -tb pointer_double_test.c
5 add_files -tb result.golden.dat
6 set_top pointer_double
7
8
9 set solutions [list 1a 2a 3a 4a]
10
11 foreach sol $solutions {
12     open_solution -reset solution_$sol
13
14     set_part {xa7a12tcsg325-1q}
15     create_clock -period 10ns
16     set_clock_uncertainty 0.1
17
18     if {$sol == "2a"} {
19         set_directive_pipeline "sub/LOOP"
20     }
21     if {$sol == "3a"} {
22         set_directive_inline sub
23     }
24     if {$sol == "4a"} {
25         set_directive_array_partition -type complete -dim 1 pointer_double array
26         set_directive_unroll "sub/LOOP"
27     }
28     csim_design
29     csynth_design
30     cosim_design -trace_level all
31 }
32
33 exit

```

Рис. 3.1. Скрипт

## 4. Моделирование

Ниже приведены результаты моделирования.

```

NFO: [HLS 200-10] In directory '/home/sobol/Downloads/labs_fr
NFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/19274215
NFO: [APCC 202-1] APCC is done.
    Generating csim.exe
est passed !
NFO: [SIM 211-1] CSim done with 0 errors.
NFO: [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.

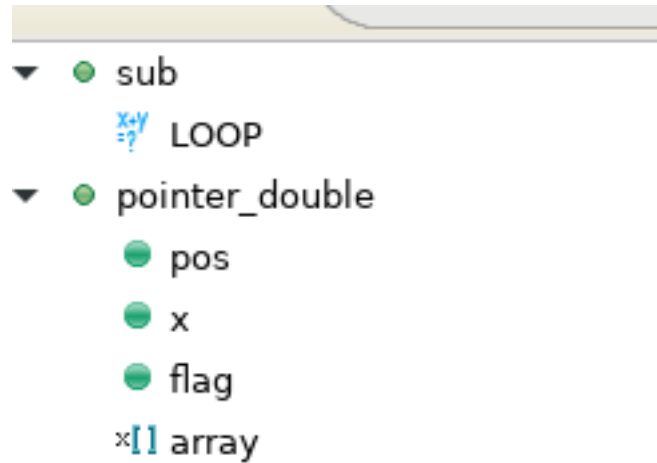


Рис. 5.1. Директивы

### 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	6.224	0.10

##### Latency (clock cycles)

##### Summary

Latency		Interval		
min	max	min	max	Type
26	26	26	26	none

Рис. 5.2. Performance estimates



## Utilization Estimates

### Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	139
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	209
Register	-	-	54	-
Total	2	0	54	348
Available	40	40	16000	8000
Utilization (%)	5	0	~0	4

Рис. 5.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ ● pointer_double	-	26	-	27	-
● LOOP	no	20	2	-	10

Рис. 5.4. Performance profile



Рис. 5.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4	C5	C6	C7
1	I/O Ports								
2	pos_r					read			
3	x					read			
4	flag						read		
5	ap_return							ret	
6	Memory Ports								
7	array(p1)	write	write	write	write	write		read	
8	array(p0)	write	write	write	write	write	write		
9	Expressions								
10	tmp_7_fu_250					l			
11	tmp_not_fu_244					icmp			
12	i_fu_270							+	
13	x_i_phi_fu_217							phi_mux	
14	i_i_phi_fu_229							phi_mux	
15	tmp_i_fu_276							&	
16	tmp_1_i_fu_281							icmp	
17	exitcond_i_fu_264							icmp	
18	x_1_fu_292								+
19	x_2_fu_298								select

Рис. 5.6. Resource viewer

## 5.3. C/RTL моделирование

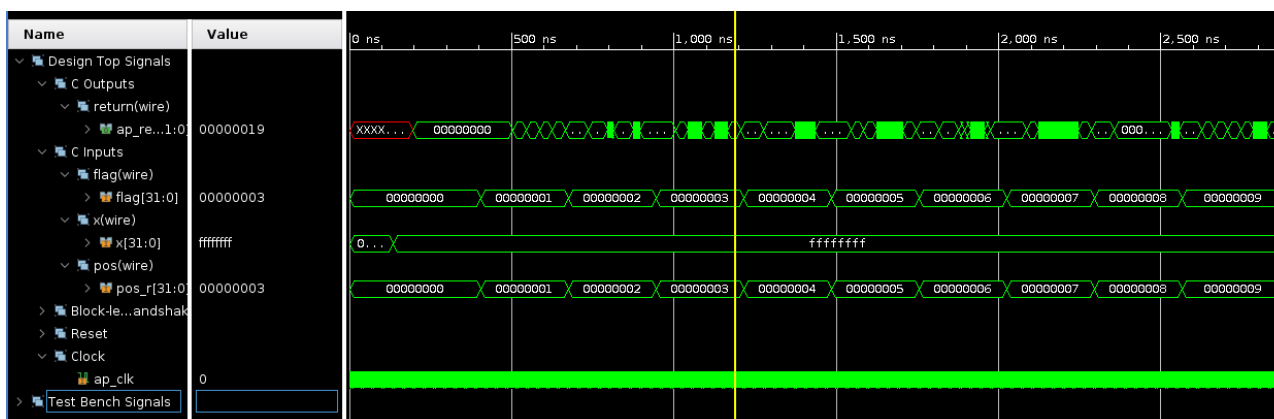


Рис. 5.7. Временная диаграмма

Исходя из временной диаграммы, цикл выполняется 10 раз, на каждую итерацию требуется 2 такта, в итоге  $\text{Latency} = 10 \cdot 2 = 20$ . Кроме цикла в программе присутствуют операции записи на которые требуется еще тактов, итоговое значение  $\text{Latency} = 20 + 6 = 26$ . Значения будут доступны на выходе еще через 1 такт  $\Pi = \text{Latency} + 1 = 27$ .

## 6. Решение 2а

### 6.1. Директивы

В данном решении были установлены директивы, приведённые ниже.

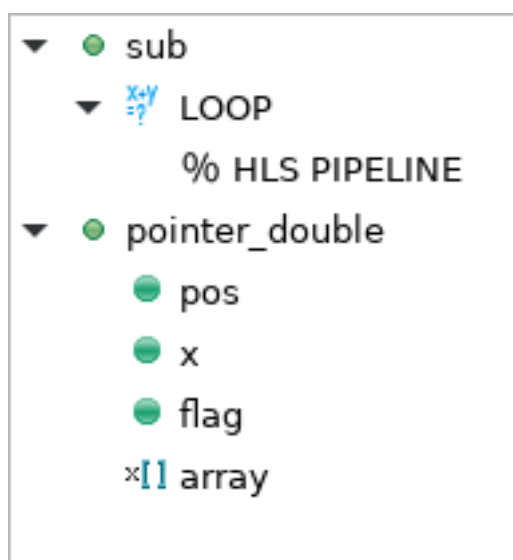


Рис. 6.1. Директивы

### 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	6.224	0.10

### Latency (clock cycles)

#### Summary

Latency		Interval		
min	max	min	max	Type
17	17	17	17	none

Рис. 6.2. Performance estimates

## Utilization Estimates

### Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	155
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	224
Register	-	-	53	-
Total	2	0	53	379
Available	40	40	16000	8000
Utilization (%)	5	0	~0	4

--

Рис. 6.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
pointer_double	-	17	-	18	-
LOOP	yes	10	2	1	10

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	pos_r					read				
3	x					read				
4	flag						read			
5	ap_return									ret
6	Memory Ports									
7	array(p0)	write	write	write	write	write	write			
8	array(p1)	write	write	write	write	write		read		
9	Expressions									
10	tmp_7_fu_260					l				
11	tmp_not_fu_254					icmp				
12	i_fu_280							+		
13	i_i_phi_fu_239							phi_mux		
14	tmp_i_fu_286							&		
15	exitcond_i_fu_274							icmp		
16	tmp_l_i_fu_291							icmp		
17	x_l_fu_302								+	
18	x_i_phi_fu_227								phi_mux	
19	x_2_fu_308								select	

Рис. 6.6. Resource viewer

## 6.3. C/RTL моделирование

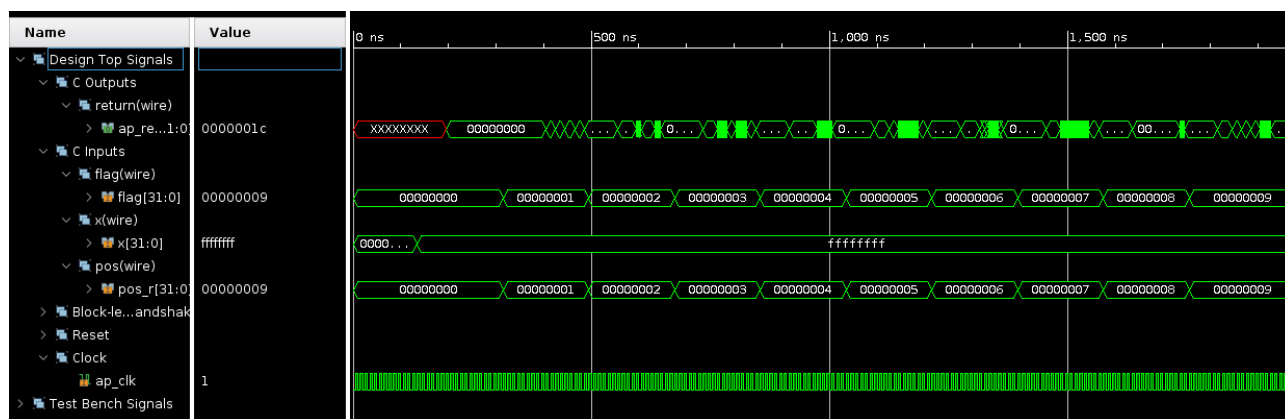


Рис. 6.7. Временная диаграмма

Как видно из временной диаграммы, значение Latency уменьшилось с 26 до 17 за цикл. Это связано с тем что после применения директивы конвейеризации модуль не ждет окончания выполнения всего цикла, а сразу читает следующие данные и подает в функцию.

## 7. Решение За

### 7.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

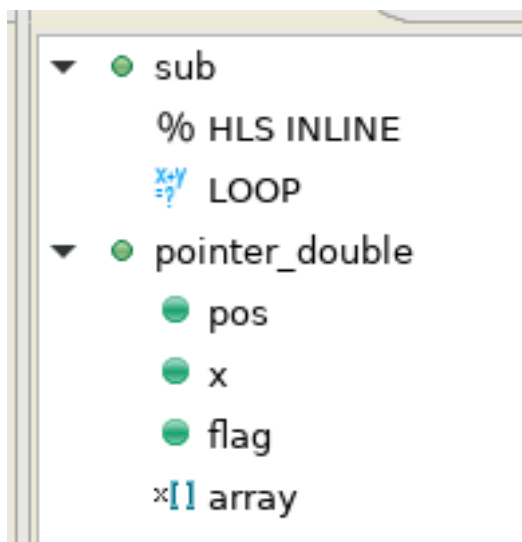


Рис. 7.1. Директивы

### 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	6.224	0.10

### Latency (clock cycles)

#### Summary

Latency		Interval		Type
min	max	min	max	
26	26	26	26	none

Рис. 7.2. Performance estimates

## Utilization Estimates

### Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	139
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	209
Register	-	-	54	-
Total	2	0	54	348
Available	40	40	16000	8000
Utilization (%)	5	0	~0	4

Рис. 7.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
▼ pointer_double	-	26	-	27	-
○ LOOP	no	20	2	-	10

Рис. 7.4. Performance profile



Рис. 7.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4	C5	C6	C7
1	I/O Ports								
2	x					read			
3	pos_r					read			
4	flag						read		
5	ap_return							ret	
6	Memory Ports								
7	array(p0)	write	write	write	write	write	write		
8	array(p1)	write	write	write	write	write		read	
9	Expressions								
10	tmp_3_fu_250								
11	tmp_not_fu_244					icmp			
12	i_fu_270							+	
13	i_0_i_phi_fu_229							phi_mux	
14	x_0_i_phi_fu_217							phi_mux	
15	tmp_5_fu_276							&	
16	tmp_6_fu_281							icmp	
17	exitcond_i_fu_264							icmp	
18	x_1_fu_292								+
19	x_2_fu_298								select

Рис. 7.6. Resource viewer



## 7.3. C/RTL моделирование

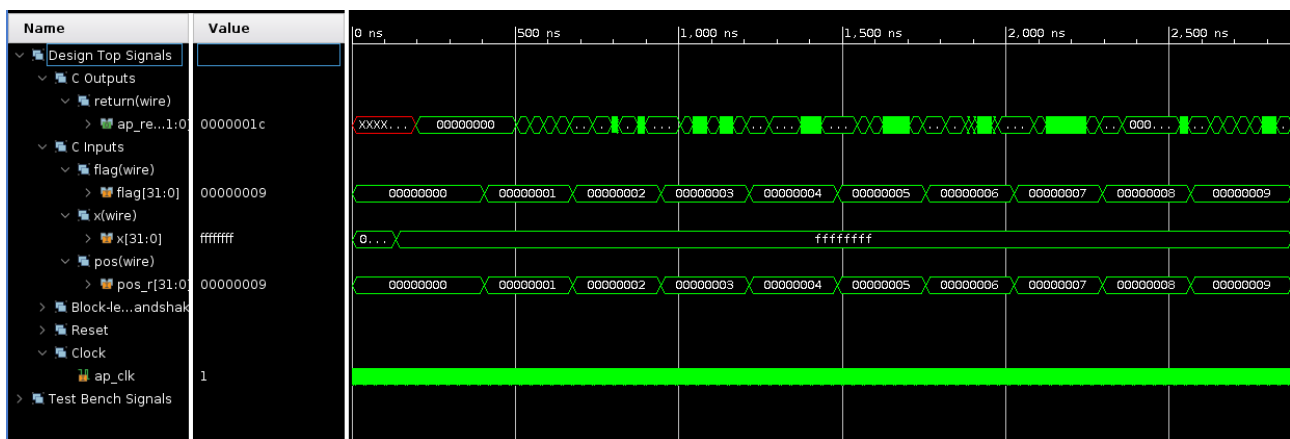


Рис. 7.7. Временная диаграмма

Как видно по результатам, после применения директивы `INLINE` решение соответствует решению 1 (по-умолчанию), то есть `inlining` не влияет на производительность данного устройства.

## 8. Решение 4a

Исходя из временных диаграмм полученных ранее, можно сделать вывод, что дольше всего выполняются команды записи и чтения в массив `array`. Для уменьшения временных задержек можно применить директиву `ARRAY_PARTITION` для массива `array` и директиву `UNROLL` для цикла.

### 8.1. Директивы

В данном решении были установлены директивы, приведённые ниже.

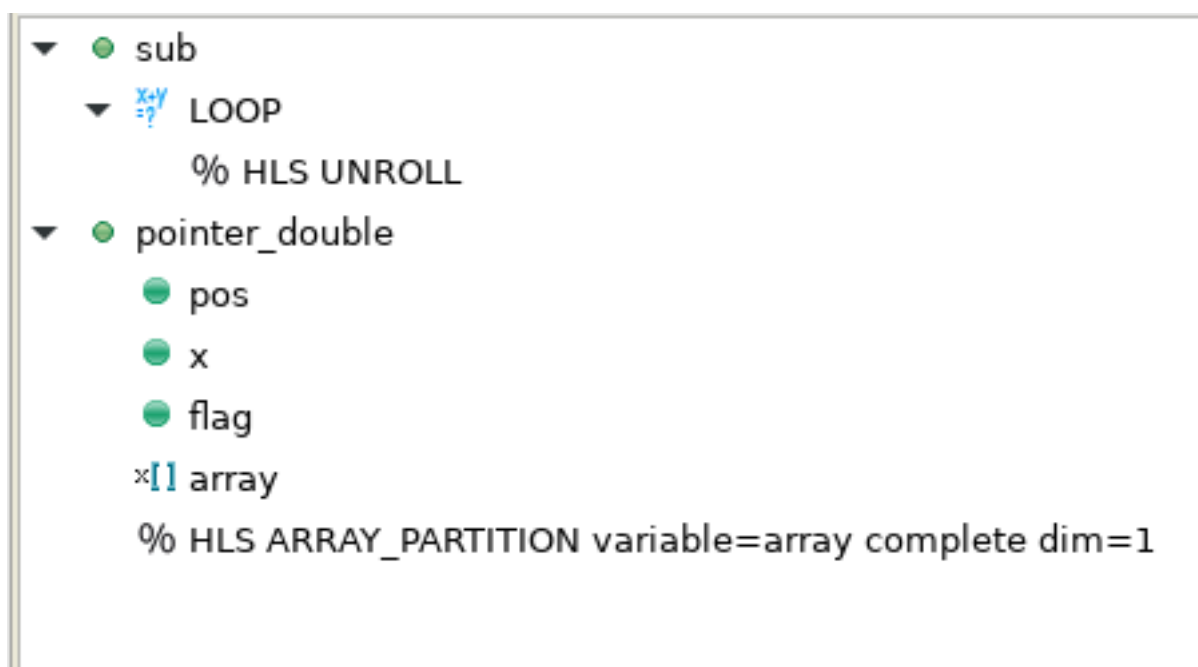


Рис. 8.1. Директивы

## 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	9.514	0.10

#### ▣ Latency (clock cycles)

##### ▣ Summary

Latency		Interval		
min	max	min	max	Type
4	4	4	4	none

Рис. 8.2. Performance estimates

### Utilization Estimates

#### ▣ Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	669
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	114
Register	-	-	395	-
Total	0	0	395	783
Available	40	40	16000	8000
Utilization (%)	0	0	2	9

— — — —

Рис. 8.3. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
pointer_double	-	4	-	5	-

Рис. 8.4. Performance profile

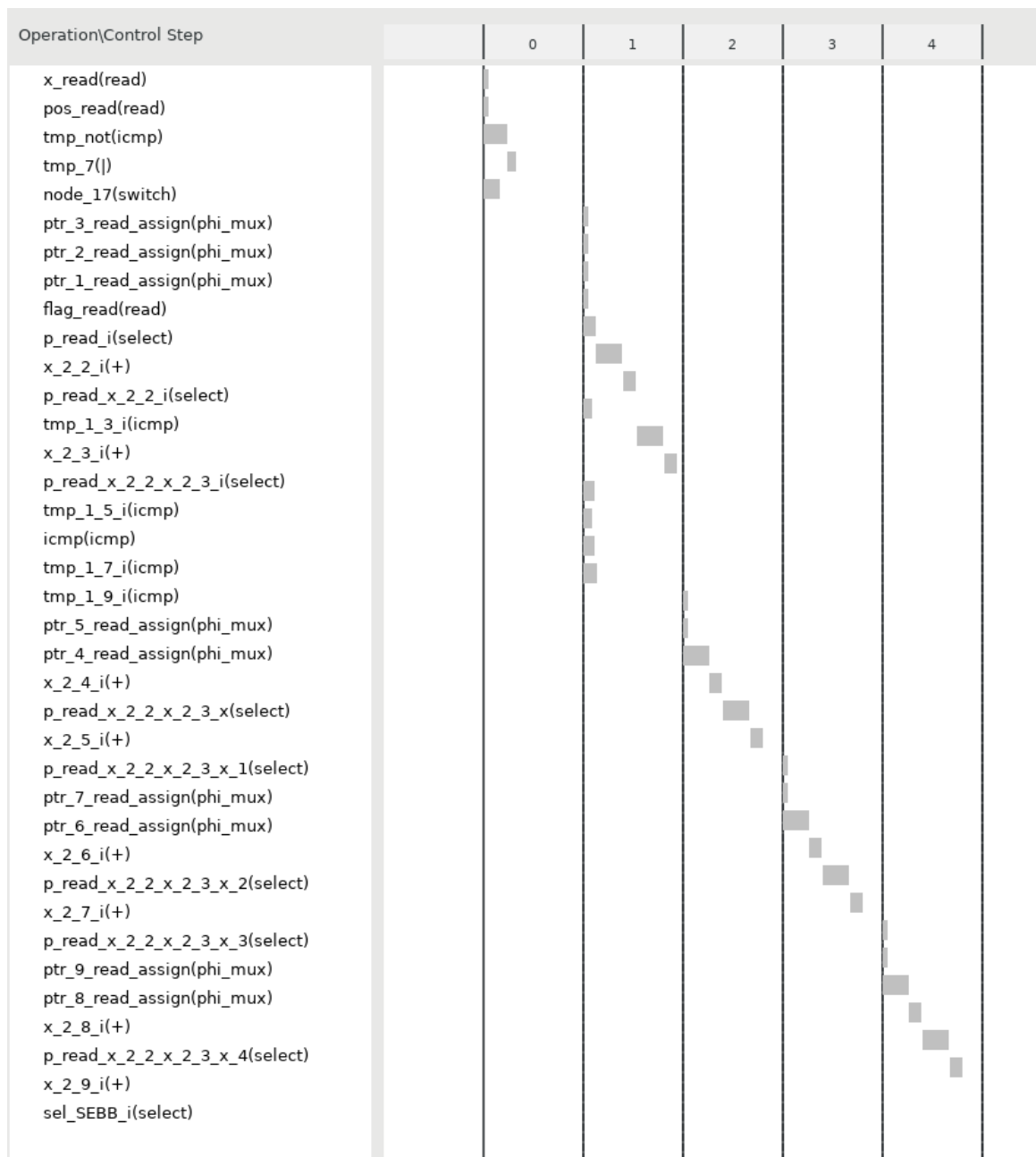


Рис. 8.5. Scheduler viewer

	Resource\Control Step	C0	C1	C2	C3	C4
1	I/O Ports					
2	pos_r	read				
3	x	read				
4	flag		read			
5	ap_return					ret
6	Expressions					
7	tmp_7_fu_446					
8	tmp_not_fu_440	icmp				
9	x_2_2_i_fu_484		+			
10	x_2_3_i_fu_504		+			
11	ptr_1_read_assign_phi_fu_170		phi_mux			
12	ptr_3_read_assign_phi_fu_94		phi_mux			
13	ptr_2_read_assign_phi_fu_132		phi_mux			
14	p_read_i_fu_468		select			
15	p_read_x_2_2_x_2_3_i_fu_510		select			
16	p_read_x_2_2_i_fu_490		select			
17	tmp_1_9_i_fu_598		icmp			
18	tmp_1_5_i_fu_544		icmp			
19	tmp_1_3_i_fu_498		icmp			
20	icmp_fu_560		icmp			
21	tmp_1_7_i_fu_566		icmp			
22	x_2_5_i_fu_615			+		
23	x_2_4_i_fu_604			+		
24	ptr_4_read_assign_phi_fu_246			phi_mux		
25	ptr_5_read_assign_phi_fu_208			phi_mux		
26	p_read_x_2_2_x_2_3_x_1_fu_621			select		
27	p_read_x_2_2_x_2_3_x_fu_609			select		
28	x_2_6_i_fu_628				+	
29	x_2_7_i_fu_639				+	
30	ptr_6_read_assign_phi_fu_322				phi_mux	
31	ptr_7_read_assign_phi_fu_284				phi_mux	
32	p_read_x_2_2_x_2_3_x_3_fu_645				select	
33	p_read_x_2_2_x_2_3_x_2_fu_633				select	
34	x_2_8_i_fu_652					+
35	x_2_9_i_fu_663					+
36	ptr_9_read_assign_phi_fu_360					phi_mux
37	ptr_8_read_assign_phi_fu_398					phi_mux
38	sel_SEBB_i_fu_669					select
39	p_read_x_2_2_x_2_3_x_4_fu_657					select

Рис. 8.6. Resource viewer

### 8.3. C/RTL моделирование

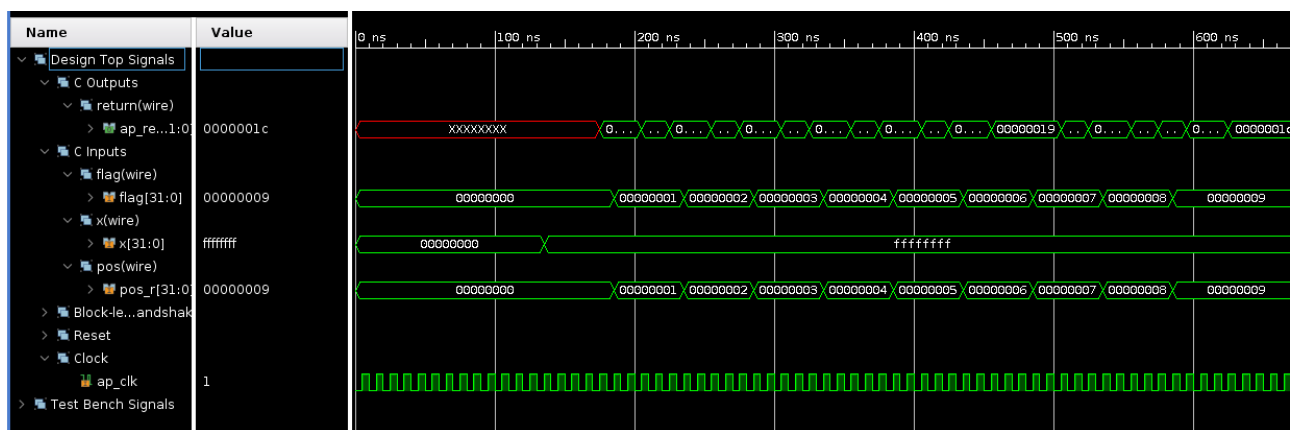


Рис. 8.7. Временная диаграмма

По результатам видно, что в данном решении достигнута наибольшая производительность по сравнению с остальными.

## 9. Вывод

Ниже приведено сравнение решений.

Performance Estimates					
Timing (ns)					
Clock		solution_1a	solution_2a	solution_3a	solution_4a
ap_clk	Target	10.00	10.00	10.00	10.00
	Estimated	6.224	6.224	6.224	9.514
Latency (clock cycles)					
		solution_1a	solution_2a	solution_3a	solution_4a
Latency	min	26	17	26	4
	max	26	17	26	4
Interval	min	26	17	26	4
	max	26	17	26	4

Рис. 9.1. Сравнение производительности

Utilization Estimates				
	solution_1a	solution_2a	solution_3a	solution_4a
BRAM_18K	2	2	2	0
DSP48E	0	0	0	0
FF	54	53	54	395
LUT	348	379	348	783

Рис. 9.2. Сравнение использования ресурсов

По результатам можно сделать вывод, что применённые директивы позволяют улучшить производительность.