

Санкт-Петербургский Политехнический Университет Петра Великого
Институт Компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Лабораторная работа 9 Задание 2

Предмет: «Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: «Оптимизация работы с массивами»

Студент: Ерниязов Т.Е.
Гр. № 3540901/81502

Преподаватель: Антонов А.П.

Санкт-Петербург
2019

Оглавление

1. Задание	7
2. Часть 1	16
2.1. Исходный код	16
2.2. Моделирование	17
2.3. Решение 1a	17
2.3.1. Директивы	17
2.3.2. Синтез	17
2.3.3. Использование ресурсов	18
2.3.4. Анализ решения	20
2.4. Решение a2	20
2.4.1. Директивы	20
2.4.2. Синтез	20
2.4.3. Использование ресурсов	21
2.4.4. Анализ решения	22
2.5. Решение a3	22
2.5.1. Директивы	22
2.5.2. Синтез	23
2.5.3. Использование ресурсов	23
2.5.4. Анализ решения	25
2.6. Решение a4	25
2.6.1. Директивы	25
2.6.2. Синтез	25
2.6.3. Использование ресурсов	26
2.6.4. Анализ решения	27

2.7. Решение а5.....	27
2.7.1. Директивы	27
2.7.2. Синтез	28
2.7.3. Использование ресурсов.....	28
2.7.4. Анализ решения.....	30
2.8. Решение а6.....	30
2.8.1. Директивы	30
2.8.2. Синтез	30
2.8.3. Использование ресурсов.....	31
2.8.4. Анализ решения.....	33
2.9. Решение а7.....	33
2.9.1. Директивы	33
2.9.2. Синтез	33
2.9.3. Использование ресурсов.....	34
2.9.4. Анализ решения.....	35
2.10. Выводы	36
3. Часть 2.....	37
3.1. Исходный код.....	37
3.2. Моделирование	38
3.3. Решение 1b	38
3.3.1. Директивы	38
3.3.2. Синтез	38
3.3.3. Использование ресурсов.....	39
3.3.4. Анализ решения.....	40
3.4. Решение 2b	41

3.4.1.	Директивы	41
3.4.2.	Синтез	41
3.4.3.	Использование ресурсов.....	41
3.4.4.	Анализ решения.....	43
3.5.	Решение 3b	44
3.5.1.	Директивы	44
3.5.2.	Синтез	44
3.5.3.	Использование ресурсов.....	44
3.5.4.	Анализ решения.....	46
3.6.	Решение 4b	47
3.6.1.	Директивы	47
3.6.2.	Синтез	47
3.6.3.	Использование ресурсов.....	47
3.6.4.	Анализ решения.....	50
3.7.	Решение 5b	50
3.7.1.	Директивы	50
3.7.2.	Синтез	50
3.7.3.	Использование ресурсов.....	50
3.7.4.	Анализ решения.....	53
3.8.	Решение 6b	53
3.8.1.	Директивы	53
3.8.2.	Синтез	53
3.8.3.	Использование ресурсов.....	54
3.8.4.	Анализ решения.....	56
3.9.	Решение 7b	56

3.9.1.	Директивы	56
3.9.2.	Синтез	56
3.9.3.	Использование ресурсов.....	57
3.9.4.	Анализ решения.....	59
3.10.	Решение 8b	60
3.10.1.	Директивы	60
3.10.2.	Синтез	60
3.10.3.	Использование ресурсов.....	60
3.10.4.	Анализ решения.....	62
3.11.	Выводы	63

1. Задание

- Создать проект lab9_2
- Микросхема: ха7a12tcsг325-1q
ЧАСТЬ 1
- Создать функцию
foo_a: входной массив short d_in[N]; выходной массив short d_out [N/4].
for (short i=0; i<N/4; i++){
*d_out[i] = d_in[i]*d_in[i+8] + d_in[i+4]*d_in[i+12];*
}
N=16
- Создать тест lab9_2_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
- Исследование:
- Solution_1a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию bram; RAM_1P_BRAM для входного (и выходного) массива
 - 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
- Solution_2a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ар_memory; RAM_1P для входного (и выходного) массива
 - 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
- Сравнить два решения (solution_1a и solution_2a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...
- Solution_3a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_2P для входного массива
 - 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
- Сравнить два решения (solution_2a и solution_3a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...
- Solution_4a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного массива
 - установить array_partition; block; factor =2 для входного массива
 - 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
- Сравнить два решения (solution_3a и solution_4a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...
- Solution_5a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного и выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - 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
-
- Сравнить два решения (solution_4a и solution_5a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...
- Solution_6a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_2P для входного и выходного массивов
 - установить array_partition; block; factor =2 для входного массива
 - 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
-
- Сравнить два решения (solution_5a и solution_6a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...
- Solution_7a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_2P для входного и выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - 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
- Сравнить два решения (solution_6a и solution_7a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...

Сделать сводную таблицу ($S_x/Latency/P$ – номер решения/Latency/P)

	RAM_1P	RAM_2P
Без block	$S_x/Latency/P$	
block; factor =2		

block; factor =4

ЧАСТЬ 2

- Создать функцию

foo_b: входной массив short d_in[N]; выходной массив short d_out [N].

for (short i=0; i<N/4; i++){

*d_out[i] = d_in[i]*d_in[i+4];*

*d_out[i+1]= d_in[i+8]*d_in[i+12];*

*d_out[i+2]= d_in[i]*d_in[i+12];*

*d_out[i+3]= d_in[i+4]*d_in[i+8];*

}

N=16

- Solution_1b

- задать: clock period 10; clock_uncertainty 0.1
- установить реализацию ap_memory; RAM_1P для входного и выходного массивов
- 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

- Solution_2b

- задать: clock period 10; clock_uncertainty 0.1
- установить реализацию ap_memory; RAM_1P для входного и выходного массивов
- установить array_partition; block; factor =4 для входного массива
- 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
- Сравнить два решения (solution_1b и solution_2b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II...
- Solution_3b
- задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного и выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - установить array_partition; cyclic; factor =2 для выходного массива
 - 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
- Сравнить два решения (solution_2b и solution_3b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II...
- Solution_4b
- задать: clock period 10; clock_uncertainty 0.1

- установить реализацию ap_memory; RAM_1P для входного и выходного массивов
- установить array_partition; block; factor =4 для входного массива
- установить array_partition; cyclic; factor =4 для выходного массива
- 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
- Сравнить два решения (solution_3b и solution_4b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II...
- Solution_5b
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного и RAM_2P для выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - установить array_partition; cyclic; factor =1 для выходного массива
 - 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

- Сравнить два решения (solution_3b и solution_5b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II...

- Solution_6b
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного и RAM_2P для выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - установить array_partition; cyclic; factor =2 для выходного массива
 - 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

- Сравнить два решения (solution_5b и solution_6b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II...

- Solution_7b
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного и RAM_2P для выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - установить array_partition; cyclic; factor =4 для выходного массива
 - 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
- Сравнить два решения (solution_6b и solution_7b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...

Сделать сводную таблицу ($S_x/Latency/P$ – номер решения/Latency/P)

	RAM_1P	RAM_2P
Без cyclic	$S_x/Latency/P$	
cyclic; factor =2		
cyclic; factor =4		

- Solution_8b
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию ap_memory; RAM_1P для входного и выходного массивов
 - установить array_partition; block; factor =4 для входного массива
 - установить array_partition; complete для выходного массива
 - 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
- Сравнить два решения (solution_2b и solution_8b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, П...

2. Часть 1.

2.1. Исходный код

Исходный код функции:

```
1  #define N 16
2
3  void foo(short d_in[N], short d_out[N / 4]) {
4      for (short i = 0; i < N / 4; i++) {
5          d_out[i] = d_in[i] * d_in[i+8] + d_in[i + 4] * d_in[i + 12];
6      }
7  }
```

Рис. 2.1.1. source code

Исходный код теста:

```
1  #include "stdio.h"
2  #define N 16
3
4  void generate(short scale, short d_in[N], short d_out[N/4]) {
5      short i;
6      for (i = 0; i < N; ++i) {
7          d_in[i] = (i + 1) * scale;
8      }
9
10     for (i = 0; i < N/4; i++) {
11         d_out[i] = d_in[i] * d_in[i + 8] + d_in[i + 4] * d_in[i + 12];
12     }
13 }
14
15 int equals(short a[N/4], short b[N/4]) {
16     for (int i = 0; i < N/4; ++i) {
17         if (a[i] != b[i]) {
18             fprintf(stdout, "%d != %d", a[i], b[i]);
19             return 0;
20         }
21     }
22     return 1;
23 }
24
25 int main() {
26     short d_in[N];
27     short d_out[N/4];
28     short expected[N/4];
29     for (int i = 1; i < N/4; ++i) {
30         generate(i, d_in, expected);
31
32         foo(d_in, d_out);
33
34         if (!equals(d_out, expected)) {
35             fprintf(stdout, "____Fail!____\n");
36             return 1;
37         }
38     }
39     fprintf(stdout, "____Pass!____\n");
40     return 0;
41 }
```

Рис. 2.1.2 test code

2.2. Моделирование

По результатам моделирование, приведённым ниже, видно, что устройство проходит тесты.

```
INFO: [HLS 200-10] Setting target device to 'xa7a12t-csg325-1Q'
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
Compiling(apcc) ../../../../source.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 version 6.2) on Sat
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab9/'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
Pass!
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис. 2.2. Modeling results

2.3. Решение 1a

2.3.1. Директивы

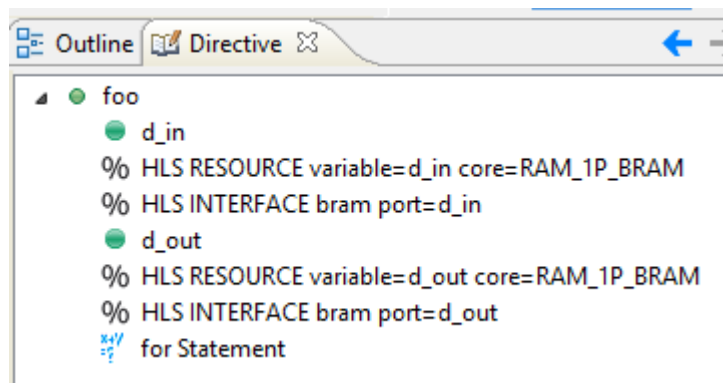


Рис. 2.3.1. Directives

2.3.2. Синтез

На изображении видно, что полученная задержка **НЕ** укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	10.437 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
25	25	0.261 us	0.261 us	25	25	none

Рис. 2.3.2. Performance estimates

2.3.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 111 LUT и 67 триггеров.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	25	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	86	-
Register	-	-	67	-	-
Total	0	2	67	111	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	1	0

Рис. 2.3.3.1. Utilization estimates

Performance Profile		Resource Profile				
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count	
foo	-	25	-	26	-	
Loop 1	no	24	6	-	4	

Рис. 2.3.3.2. Performance profile

Performance Profile		Resource Profile								
		BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Wor
foo		0	2	67	111					
I/O Ports(2)						32				
Instances(0)		0	0	0	0					
Memories(0)		0		0	0	0			0	0
Expressions(3)		0	0	0	25	9	9	0		
Registers(8)				67		128				
Channels(0)		0		0	0	0			0	0
Multiplexers(4)		0		0	86	38			0	
DSP(2)			2							

Рис. 2.3.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_A	out	32	bram	d_in	array
d_in_EN_A	out	1	bram	d_in	array
d_in_WEN_A	out	2	bram	d_in	array
d_in_Din_A	out	16	bram	d_in	array
d_in_Dout_A	in	16	bram	d_in	array
d_in_Clk_A	out	1	bram	d_in	array
d_in_Rst_A	out	1	bram	d_in	array
d_out_A	out	32	bram	d_out	array
d_out_EN_A	out	1	bram	d_out	array
d_out_WEN_A	out	2	bram	d_out	array
d_out_Din_A	out	16	bram	d_out	array
d_out_Dout_A	in	16	bram	d_out	array
d_out_Clk_A	out	1	bram	d_out	array
d_out_Rst_A	out	1	bram	d_out	array

Рис. 2.3.3.4. Interfaces

Ниже, на рисунке видны блоки, которые не укладываются во временной интервал.

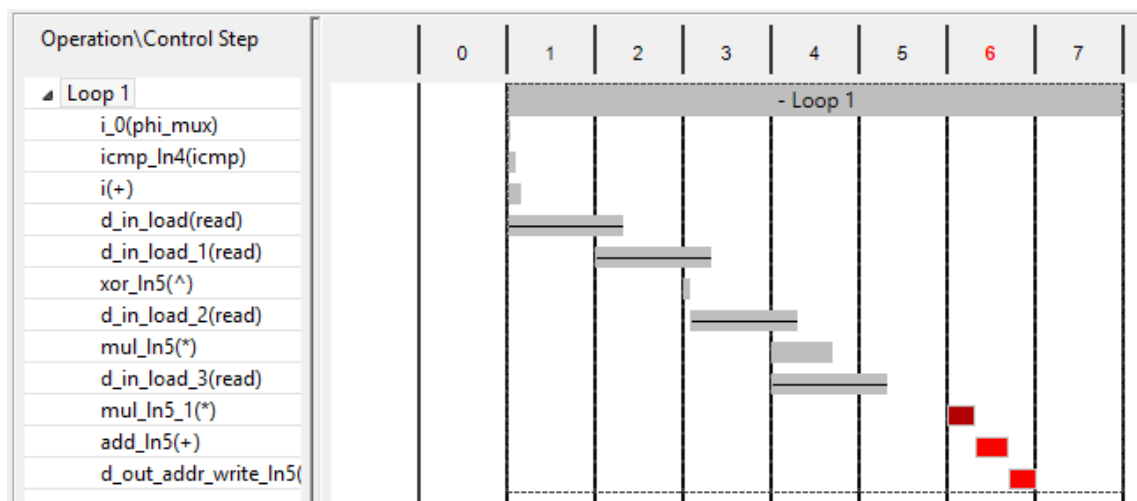


Рис. 2.3.3.5. Schedule viewer

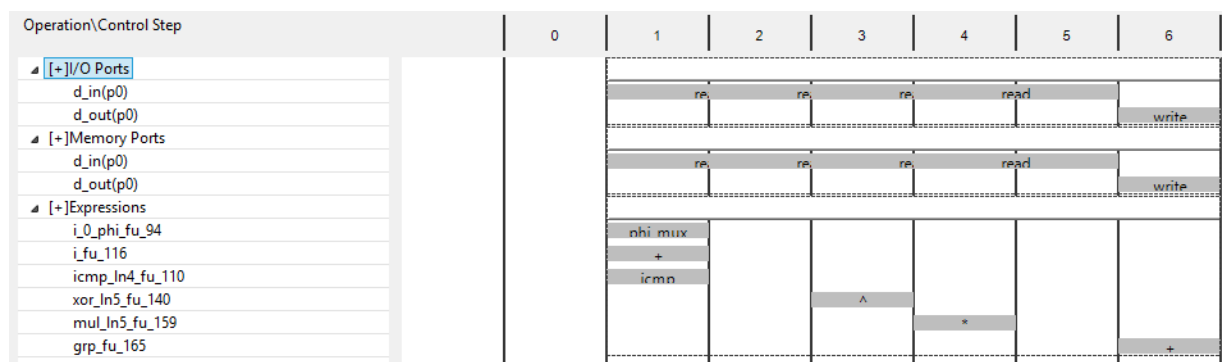


Рис. 2.3.3.5. Resource viewer

2.3.4. Анализ решения

На каждой итерации для двух операций умножения в устройстве используется 2 умножителя. 25 тактов задержки – это 1 начальный такт инициализации и 4 итерации цикла, каждый из которых занимает 6 тактов. П совпадает с задержкой, так как устройство не конвейеризировано и работает последовательно. Устройство не уложилось в заданные 10 нс периода тактовой частоты.

2.4.Решение а2

2.4.1. Директивы

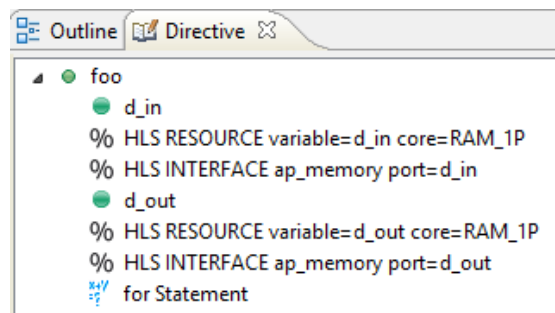


Рис.2.4.1. Directives

2.4.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
25	25	0.250 us	0.250 us	25	25	none

Рис. 2.4.2. Performance estimates

2.4.3. Использование ресурсов

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	25	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	77	-
Register	-	-	67	-	-
Total	0	2	67	102	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	1	0

Рис. 2.4.3.1. Utilization estimates

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 102 LUT и 67 триггеров.

Performance Profile				
Resource Profile				
	Pipelined	Latency	Iteration Latency	Initiation Inter
foo	-	25	-	26
Loop 1	no	24	6	-

Рис. 2.4.3.2. Performance profile

Resource Profile						
	BRAM	DSP	FF	LUT	Bits P0	Bits P1
foo	0	2	67	102		
I/O Ports(2)					32	
Instances(0)	0	0	0	0		
Memories(0)	0		0	0	0	
Expressions(3)	0	0	0	25	9	9
Registers(8)			67		128	
Channels(0)	0		0	0	0	
Multiplexers(3)	0		0	77	8	
DSP(2)		2				

Рис. 2.4.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_address0	out	4	ap_memory	d_in	array
d_in_ce0	out	1	ap_memory	d_in	array
d_in_q0	in	16	ap_memory	d_in	array
d_out_address0	out	2	ap_memory	d_out	array
d_out_ce0	out	1	ap_memory	d_out	array
d_out_we0	out	1	ap_memory	d_out	array
d_out_d0	out	16	ap_memory	d_out	array

Рис. 2.4.3.4. Interfaces



Рис. 2.4.3.5. Schedule viewer

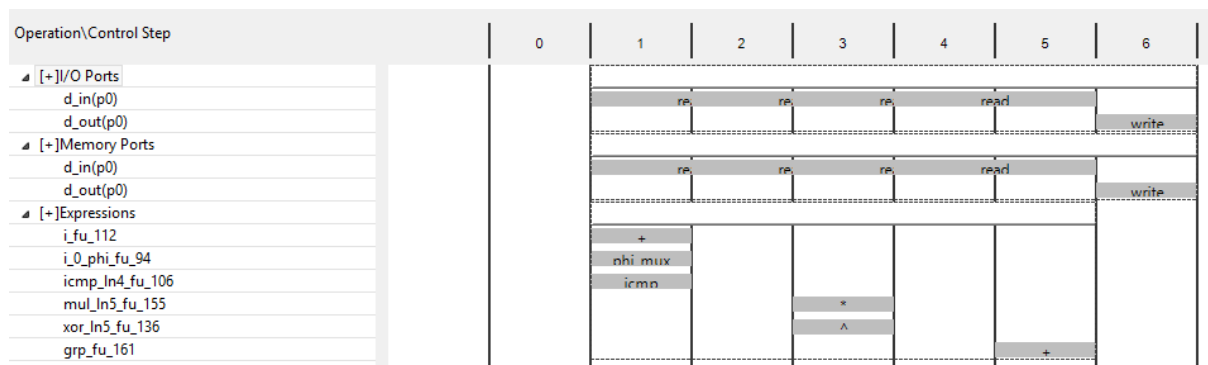


Рис. 2.4.3.6. Resource viewer

2.4.4. Анализ решения

Данное решение укладывается в установленный временной интервал. Это связано с другими используемыми элементами памяти.

2.5. Решение а3

2.5.1. Директивы

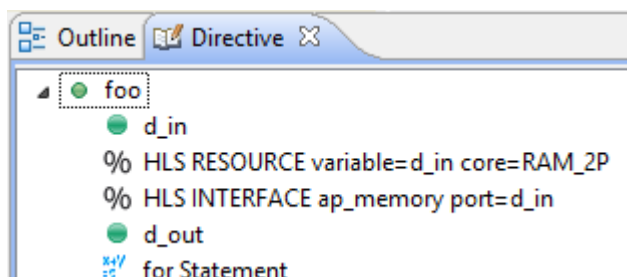


Рис. 2.5.1. Directives

2.5.2. Синтез

Ниже приведены оценки производительности. По ним видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
17	17	0.170 us	0.170 us	17	17	none

Рис. 2.5.2. Performance estimates

2.5.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 97 LUT и 46 триггеров.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	25	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	72	-
Register	-	-	46	-	-
Total	0	2	46	97	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	1	0

Рис. 2.5.3.1. Utilization estimates

Performance Profile		Resource Profile		
	Pipelined	Latency	Iteration Latency	Initiation Interval
foo	-	17	-	18
Loop 1	no	16	4	-

Рис. 2.5.3.2. Performance profile

Performance Profile		Resource Profile						
		BRAM	DSP	FF	LUT	Bits P0	Bits P1	B
foo		0	2	46	97			
I/O Ports(2)					32			
Instances(0)		0	0	0	0			
Memories(0)		0		0	0	0		
Expressions(3)		0	0	0	25	9	9	0
Registers(6)				46	107			
Channels(0)		0		0	0			
Multiplexers(4)		0		0	72	12		
DSP(2)			2					

Рис. 2.5.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_address0	out	4	ap_memory	d_in	array
d_in_ce0	out	1	ap_memory	d_in	array
d_in_q0	in	16	ap_memory	d_in	array
d_in_address1	out	4	ap_memory	d_in	array
d_in_ce1	out	1	ap_memory	d_in	array
d_in_q1	in	16	ap_memory	d_in	array
d_out_address0	out	2	ap_memory	d_out	array
d_out_ce0	out	1	ap_memory	d_out	array
d_out_we0	out	1	ap_memory	d_out	array
d_out_d0	out	16	ap_memory	d_out	array

Рис. 2.5.3.4. Interfaces

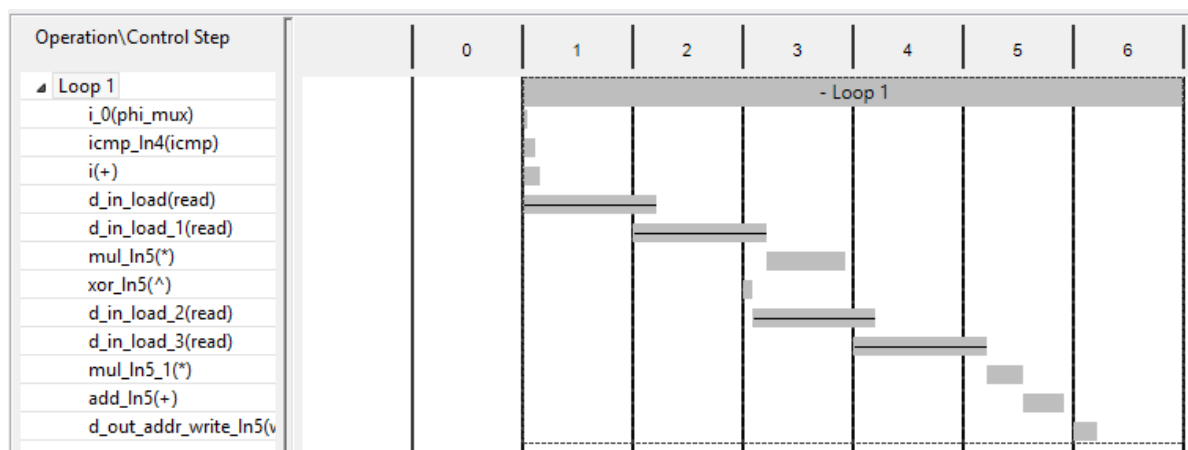


Рис. 2.5.3.5. Schedule viewer

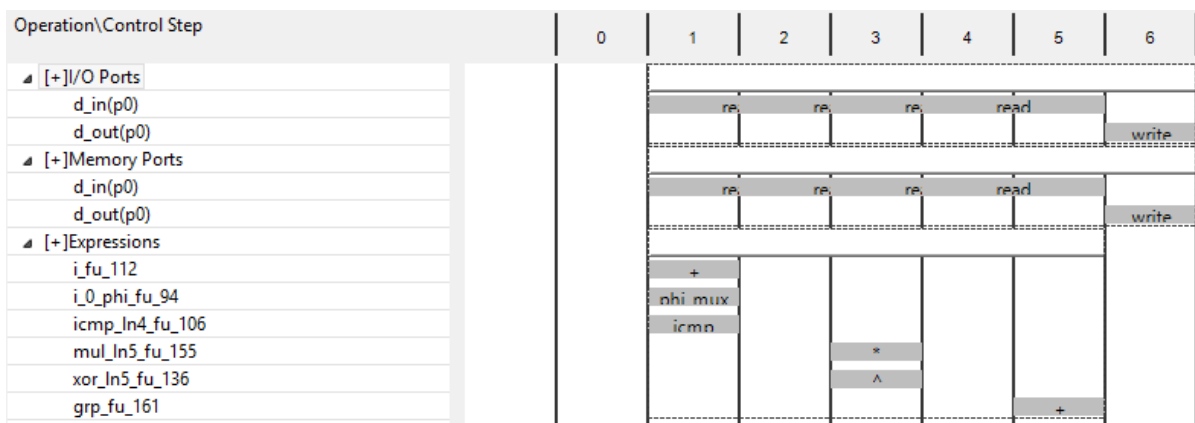


Рис. 2.5.3.6. Resource viewer

2.5.4. Анализ решения

Данное решение быстрее и экономичнее относительно предыдущего. В нём также используется 2 умножителя, но благодаря двухпортовой памяти можно производить по 2 чтения из входного массива, что сокращает выполнение 1 итерации с 6 до 4 тактов. Следовательно, latency и П меньше на 8 тактов (-2 такта на 4х итерациях).

2.6.Решение a4

2.6.1. Директивы

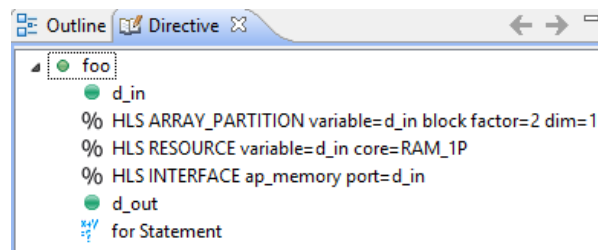


Рис. 2.6.1. Directives

2.6.2. Синтез

Ниже приведены оценки производительности. По ним видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
17	17	0.170 us	0.170 us	17	17	none

Рис. 2.6.2. Performance estimates

2.6.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 97 LUT и 62 триггера.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	25	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	72	-
Register	-	-	62	-	-
Total	0	2	62	97	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	1	0

Рис. 2.6.3.1. Utilization estimates

Performance Profile				
	Pipelined	Latency	Iteration Latency	Initiation Interval
foo	-	17	-	18
Loop 1	no	16	4	-

Рис. 2.6.3.2. Performance profile

Resource Profile									
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	
foo1	0	0	7	343					
I/O Ports(3)					96				
Instances(0)	0	0	0	0					
Memories(0)	0	0	0	0	0			0	
Expressions(4)	0	0	0	82	66	67	0		
Registers(3)			7	7					
Channels(0)	0		0	0	0			0	
Multiplexers(11)	0		0	261	91			0	
DSP(0)		0							

Рис. 2.6.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_0_address0	out	3	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	3	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_out_address0	out	2	ap_memory	d_out	array
d_out_ce0	out	1	ap_memory	d_out	array
d_out_we0	out	1	ap_memory	d_out	array
d_out_d0	out	16	ap_memory	d_out	array

Рис. 2.6.3.4. Interfaces

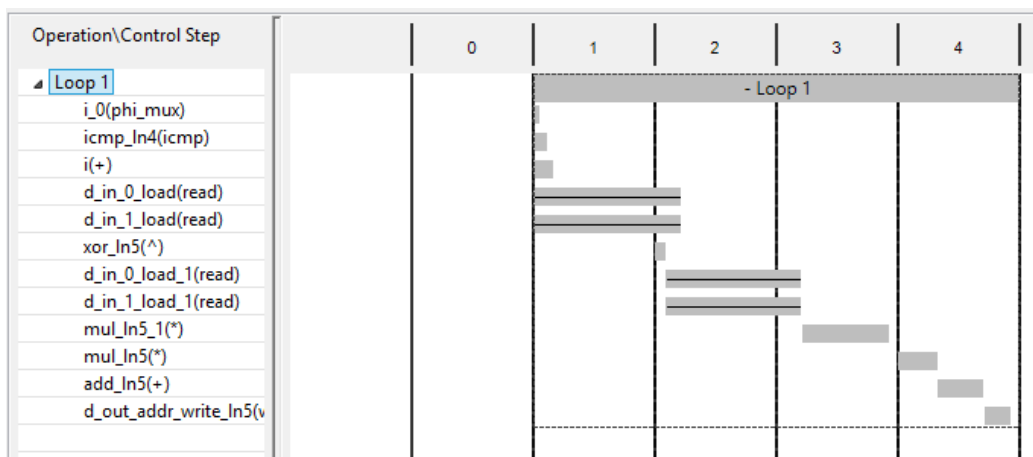


Рис. 2.6.3.5. Schedule viewer

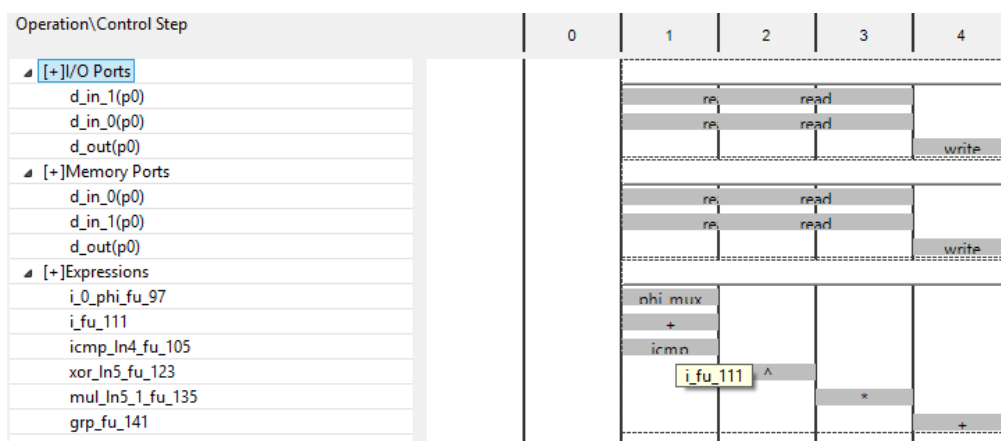


Рис. 2.6.3.6. Schedule viewer

2.6.4. Анализ решения

В данном решение отсутствует двухпортовая память и используется 2 экземпляра однопортовой памяти. На результирующие характеристики устройства это не повлияло.

2.7.Решение а5

2.7.1. Директивы

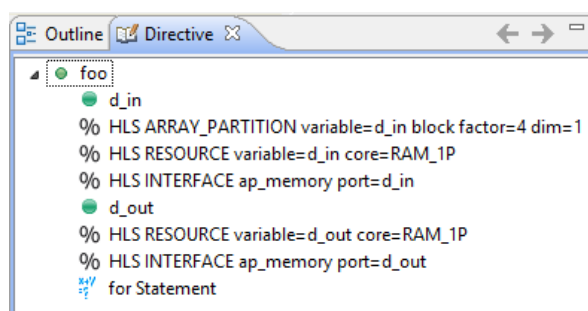


Рис. 2.7.1. Directives

2.7.2. Синтез

Ниже приведены оценки производительности. По ним видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
13	13	0.130 us	0.130 us	13	13	none

Рис. 2.7.2. Performance estimates

2.7.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 57 LUT и 61 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	21	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	36	-
Register	-	-	61	-	-
Total	0	2	61	57	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	~0	0

Рис. 2.7.3.1. Utilization estimates

Performance Profile		Resource Profile		
	Pipelined	Latency	Iteration Latency	Initiation Interval
foo	-	13	-	14
Loop 1	no	12	3	-

Рис. 2.7.3.2. Performance profile

Performance Profile		Resource Profile					
		BRAM	DSP	FF	LUT	Bits P0	Bits P1
foo		0	2	61	57		
I/O Ports(5)						80	
Instances(0)		0	0	0	0		
Memories(0)		0	0	0	0		
Expressions(2)		0	0	0	21	6	5
Registers(7)				61	122		
Channels(0)		0		0	0		
Multiplexers(2)		0		0	36	4	
DSP(2)			2				

Рис. 2.7.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_address0	out	2	ap_memory	d_out	array
d_out_ce0	out	1	ap_memory	d_out	array
d_out_we0	out	1	ap_memory	d_out	array
d_out_d0	out	16	ap_memory	d_out	array

Рис. 2.7.3.4. Interfaces

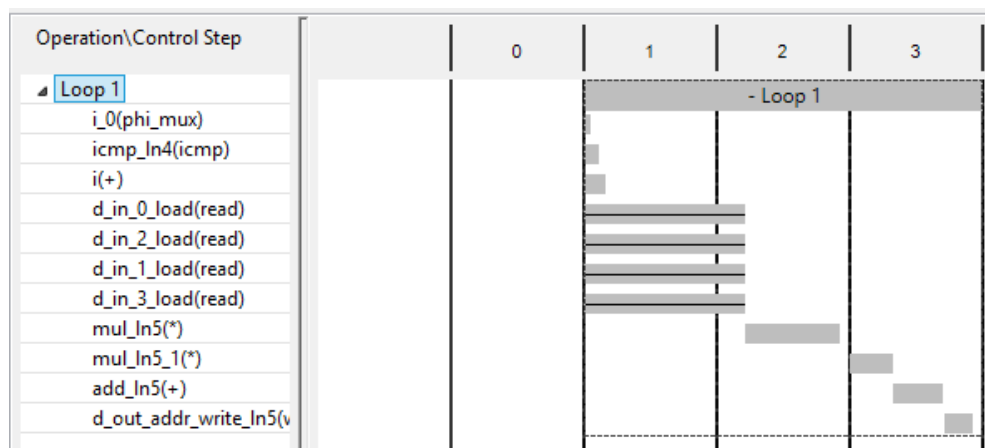


Рис. 2.7.3.5. Schedule viewer

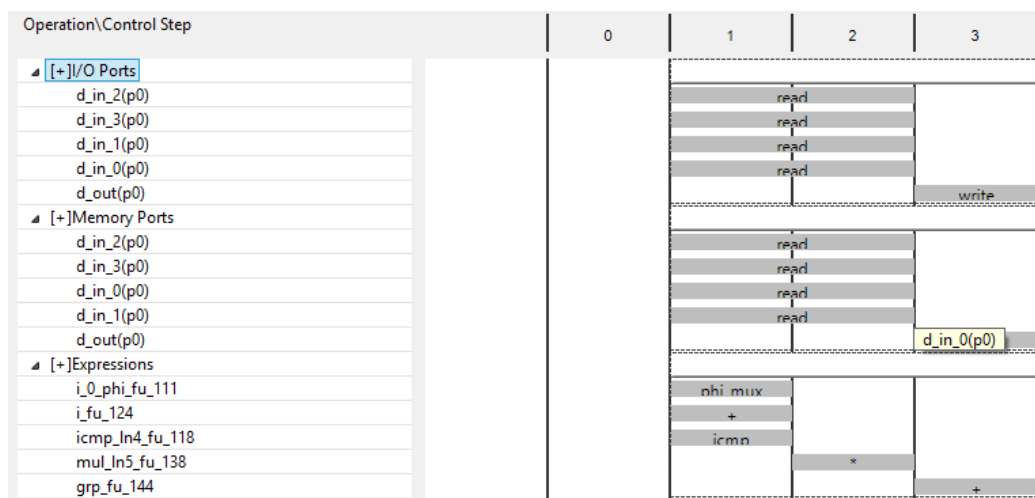


Рис. 2.7.3.6. Schedule viewer

2.7.4. Анализ решения

В данном решение latency и П сократились на 4 такта. Это можно объяснить, что используется 4 экземпляра однопортовой памяти и все 4 операции чтения проходят одновременно. Это сокращает выполнение одной итерации на 1, что и создает улучшение в задержке на 4 такта.

2.8.Решение а6

2.8.1. Директивы

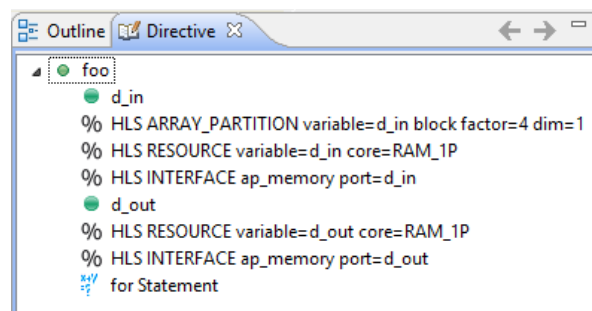


Рис. 2.8.1. Directives

2.8.2. Синтез

Ниже приведены оценки производительности. По ним видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
13	13	0.130 us	0.130 us	13	13	none

Рис. 2.8.2. Performance estimates

2.8.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 61 LUT и 61 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	25	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	36	-
Register	-	-	61	-	-
Total	0	2	61	61	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	~0	0

Рис. 2.8.3.1. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	13	-	14	-
Loop 1	no	12	3	-	4

Рис. 2.8.3.2. Performance profile

Performance Profile		Resource Profile						
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Bank
foo	0	2	61	61				
I/O Ports(3)					48			
Instances(0)	0	0	0	0				
Memories(0)	0	0	0	0	0			0
Expressions(3)	0	0	0	25	9	9	0	
Registers(7)			61		122			
Channels(0)	0		0	0	0			0
Multiplexers(2)	0		0	36	4			0
DSP(2)		2						

Рис. 2.8.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_0_address0	out	3	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_0_address1	out	3	ap_memory	d_in_0	array
d_in_0_ce1	out	1	ap_memory	d_in_0	array
d_in_0_q1	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	3	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_1_address1	out	3	ap_memory	d_in_1	array
d_in_1_ce1	out	1	ap_memory	d_in_1	array
d_in_1_q1	in	16	ap_memory	d_in_1	array
d_out_address1	out	2	ap_memory	d_out	array
d_out_ce1	out	1	ap_memory	d_out	array
d_out_we1	out	1	ap_memory	d_out	array
d_out_d1	out	16	ap_memory	d_out	array

Рис. 2.8.3.4. Interfaces

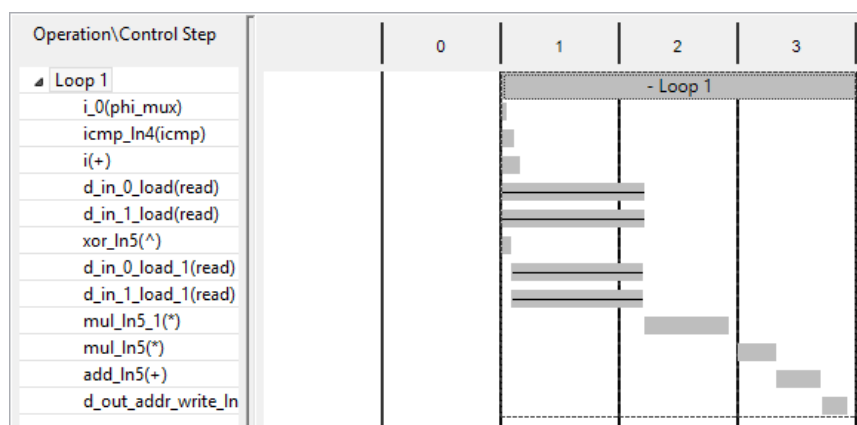


Рис. 2.8.3.5. Schedule viewer

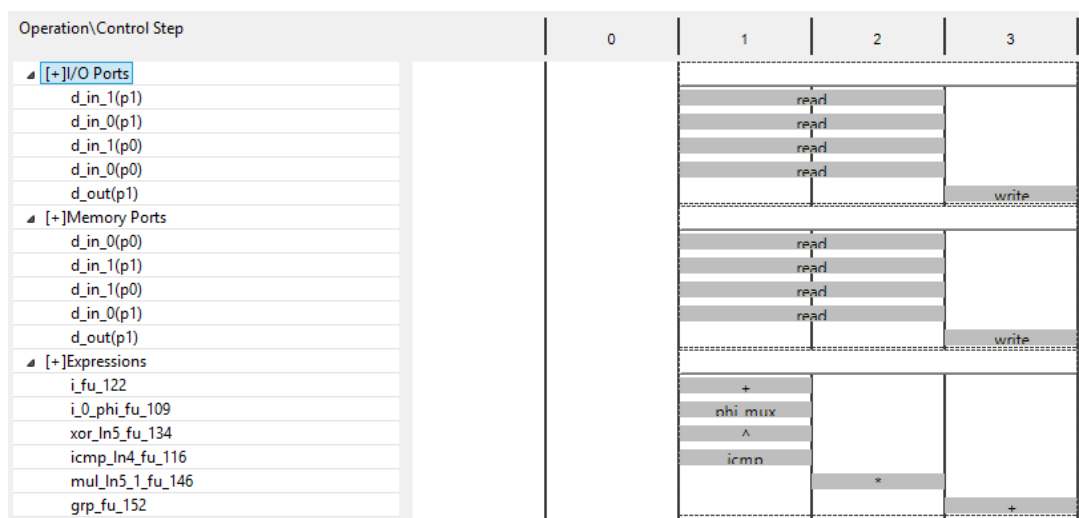


Рис. 2.8.3.6. Schedule viewer

2.8.4. Анализ решения

В данном решении latency и П оказались одинаковыми. Использовать 4 однопортовых памяти или 2 двухпортовых оказалось практически одинаковыми. Разницу можно заметить только в том, что в решении ба есть ещё одна небольшая операция получения новых индексов для чтения из памяти. Однако, это операция очень быстрая, и все 4 чтения происходят практически одновременно.

2.9. Решение a7

2.9.1. Директивы

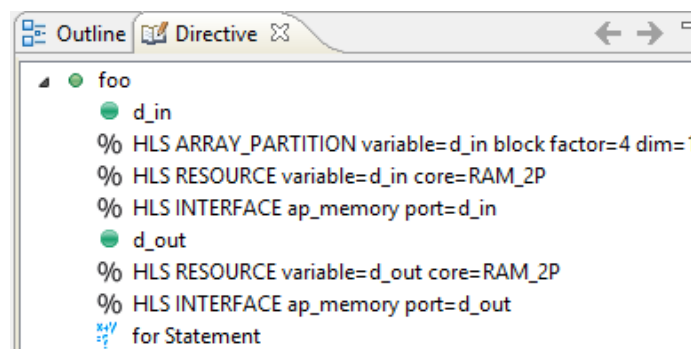


Рис. 2.9.1. Directives

2.9.2. Синтез

Ниже приведены оценки производительности. По ним видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
13	13	0.130 us	0.130 us	13	13	none

Рис. 2.9.2. Performance estimates

2.9.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 2 блока DSP48E, 57 LUT и 61 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	2	-	-	-
Expression	-	-	0	21	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	36	-
Register	-	-	61	-	-
Total	0	2	61	57	0
Available	40	40	16000	8000	0
Utilization (%)	0	5	~0	~0	0

Рис. 2.9.3.1. Utilization estimates

Performance Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	13	-	14	-
Loop 1	no	12	3	-	4

Рис. 2.9.3.2. Performance profile

Resource Profile							
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2
foo	0	2	61	57			
I/O Ports(5)					80		
Instances(0)	0	0	0	0			
Memories(0)	0		0	0	0		0
Expressions(2)	0	0	0	21	6	5	0
Registers(7)			61		122		
Channels(0)	0		0	0	0		0
Multiplexers(2)	0		0	36	4		0
DSP(2)		2					

Рис. 2.9.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo	return value
ap_rst	in	1	ap_ctrl_hs	foo	return value
ap_start	in	1	ap_ctrl_hs	foo	return value
ap_done	out	1	ap_ctrl_hs	foo	return value
ap_idle	out	1	ap_ctrl_hs	foo	return value
ap_ready	out	1	ap_ctrl_hs	foo	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_address1	out	2	ap_memory	d_out	array
d_out_ce1	out	1	ap_memory	d_out	array
d_out_we1	out	1	ap_memory	d_out	array
d_out_d1	out	16	ap_memory	d_out	array

Рис. 2.9.3.4. Interfaces

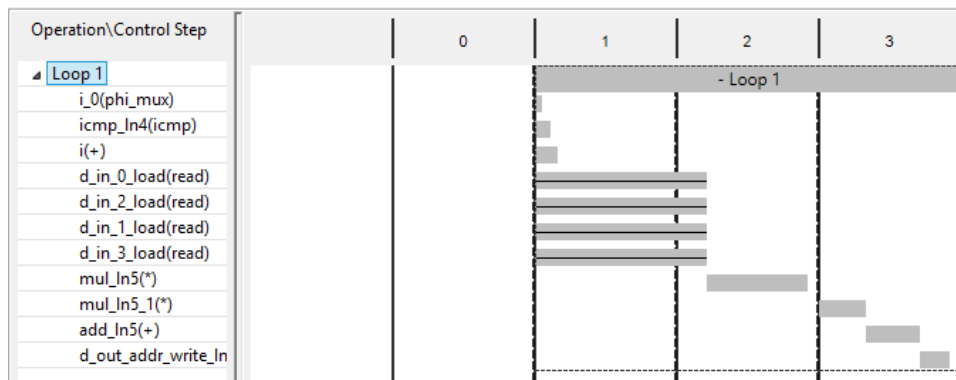


Рис. 2.9.3.5. Schedule viewer

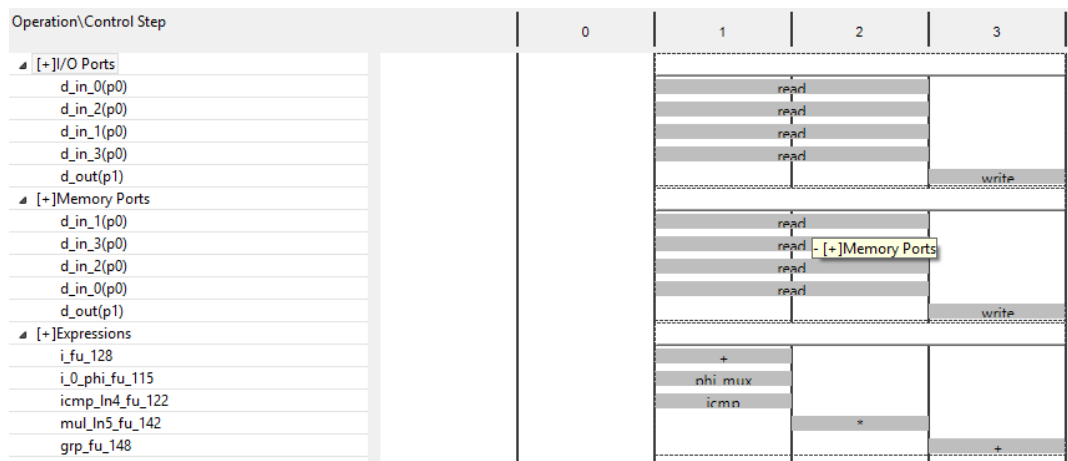


Рис. 2.9.3.6. Schedule viewer

2.9.4. Анализ решения

Видно, что задержки остались прежними. Данное решение аналогично решению 5а по всем параметрам. Это означает, что для данного устройства нет необходимости

иметь 4 двухпортовых памяти, и программа автоматически создала 4 однопортовых, как более экономичное решение.

2.10. Выводы

Ниже представлена таблица сравнения всех решений первой части.

Performance Estimates								
⊟ Timing								
Clock		solution1	solution2	solution3	solution4	solution5	solution6	solution7
ap_clk	Target	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns
	Estimated	10.437 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns
⊟ Latency								
		solution1	solution2	solution3	solution4	solution5	solution6	solution7
Latency (cycles)	min	25	25	17	17	13	13	13
	max	25	25	17	17	13	13	13
Latency (absolute)	min	0.261 us	0.250 us	0.170 us	0.170 us	0.130 us	0.130 us	0.130 us
	max	0.261 us	0.250 us	0.170 us	0.170 us	0.130 us	0.130 us	0.130 us
Interval (cycles)	min	25	25	17	17	13	13	13
	max	25	25	17	17	13	13	13

Наименьшие задержки получили те решения, где есть возможность осуществлять более 4 чтений массива памяти. Решение, которые имеет больше 4 чтений (7а), реализуется, как решение (5а), так как оно избыточно, ведь в устройстве на каждой итерации всего 4 операции чтения. Такая избыточность может понадобиться только, если использовать конвейеризацию, а пока 4 итерации проходят последовательно, этого не требуется.

3. Часть 2

3.1. Исходный код

Исходный код функции:

```
1  #include "source.h"
2
3  void foo_part2(short d_in[N], short d_out[N/4 + 3]) {
4      for (short i = 0; i < N/4; i++) {
5          d_out[i] = d_in[i] * d_in[i + 4];
6          d_out[i + 1] = d_in[i + 8] * d_in[i + 12];
7          d_out[i + 2] = d_in[i] * d_in[i + 12];
8          d_out[i + 3] = d_in[i + 4] * d_in[i + 8];
9      }
10 }
```

Рис. 3.1.1. source code

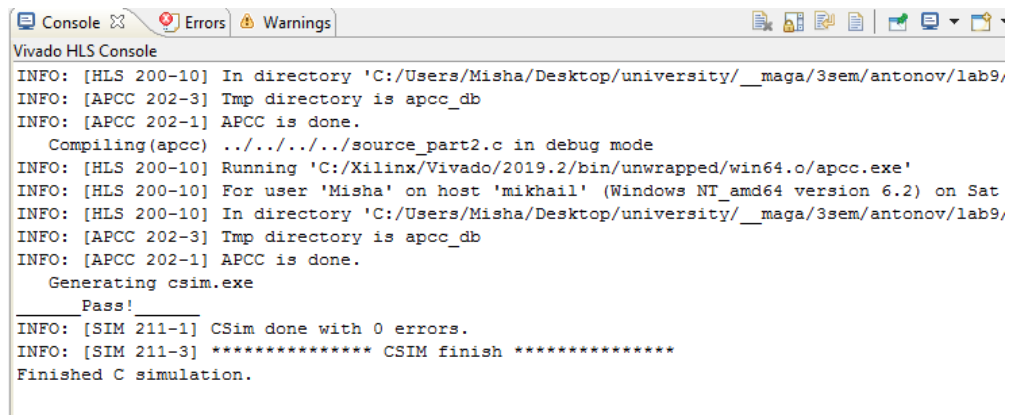
Исходный код теста:

```
1  #include "stdio.h"
2  #include "source.h"
3
4  void generate(short scale, short d_in[N], short d_out[N/4 + 3]) {
5      short i;
6      for (i = 0; i < N; ++i) {
7          d_in[i] = (i + 1) * scale;
8      }
9
10     for (i = 0; i < N/4; i++) {
11         d_out[i] = d_in[i] * d_in[i + 4];
12         d_out[i + 1] = d_in[i + 8] * d_in[i + 12];
13         d_out[i + 2] = d_in[i] * d_in[i + 12];
14         d_out[i + 3] = d_in[i + 4] * d_in[i + 8];
15     }
16 }
17
18 int equals(short a[N/4 + 3], short b[N/4 + 3]) {
19     for (int i = 0; i < N/4 + 3; ++i) {
20         if (a[i] != b[i]) {
21             fprintf(stdout, "%d != %d", a[i], b[i]);
22             return 0;
23         }
24     }
25     return 1;
26 }
27
28 int main() {
29     short d_in[N];
30     short d_out[N/4 + 3];
31     short expected[N/4 + 3];
32     for (int i = 1; i < 4; ++i) {
33         generate(i, d_in, expected);
34
35         foo_part2(d_in, d_out);
36
37         if (!equals(d_out, expected)) {
38             fprintf(stdout, "____Fail!____\n");
39             return 1;
40         }
41     }
42     fprintf(stdout, "____Pass!____\n");
43     return 0;
44 }
```

Рис. 3.1.2 test code

3.2. Моделирование

По результатам моделирование, приведённым ниже, видно, что устройство проходит тесты.



```
Vivado HLS Console
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab9,
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
  Compiling(apcc) ../../../../source_part2.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 version 6.2) on Sat
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab9,
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
  Generating csim.exe
  Pass!
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис. 3.2. Modeling results

3.3. Решение 1b

3.3.1. Директивы

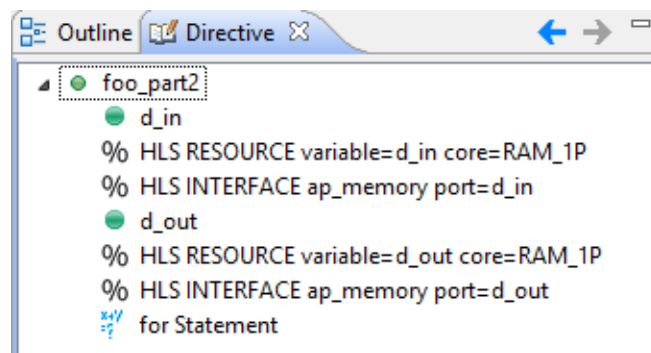


Рис. 3.3.1. Directives

3.3.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
29	29	0.290 us	0.290 us	29	29	none

Рис. 3.3.2. Performance estimates

3.3.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 183 LUT и 116 триггеров.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	49	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	134	-
Register	-	-	116	-	-
Total	0	4	116	183	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	~0	2	0

Рис. 3.3.3.1. Utilization estimates

Performance Profile		Resource Profile				
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count	
foo_part2	-	29	-	30	-	
Loop 1	no	28	7	-	4	

Рис. 3.3.3.2. Performance profile

Performance Profile		Resource Profile						
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth
foo_part2	0	4	116	183				
I/O Ports(2)					32			
Instances(0)	0	0	0	0				
Memories(0)	0		0	0	0			0
Expressions(5)	0	0	0	49	15	13	0	
Registers(11)			116		177			
Channels(0)	0		0	0	0			0
Multiplexers(5)	0		0	134	27			0
DSP(4)		4						

Рис. 3.3.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_address0	out	4	ap_memory	d_in	array
d_in_ce0	out	1	ap_memory	d_in	array
d_in_q0	in	16	ap_memory	d_in	array
d_out_address0	out	3	ap_memory	d_out	array
d_out_ce0	out	1	ap_memory	d_out	array
d_out_we0	out	1	ap_memory	d_out	array
d_out_d0	out	16	ap_memory	d_out	array

Рис. 3.3.3.4. Interfaces

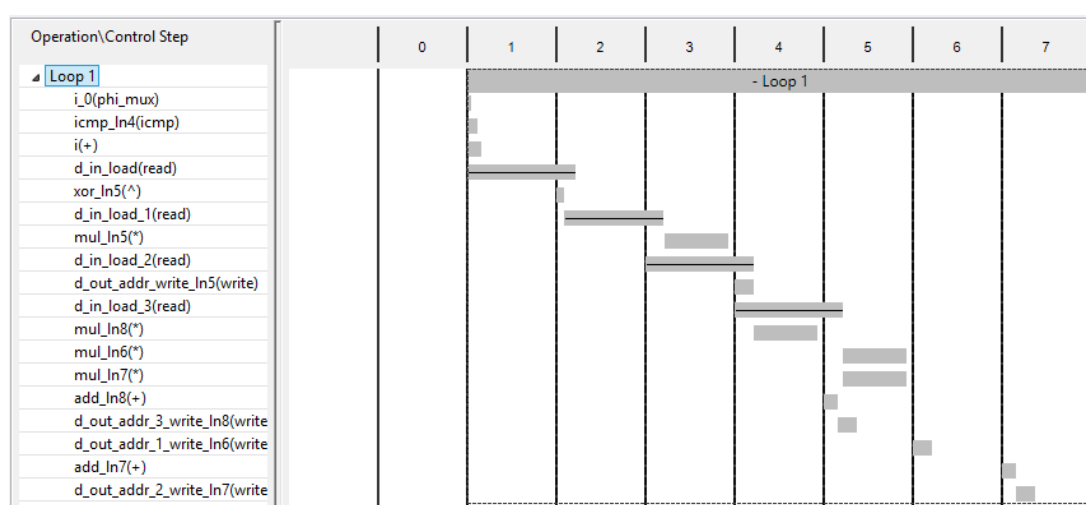


Рис. 3.3.3.5. Schedule viewer

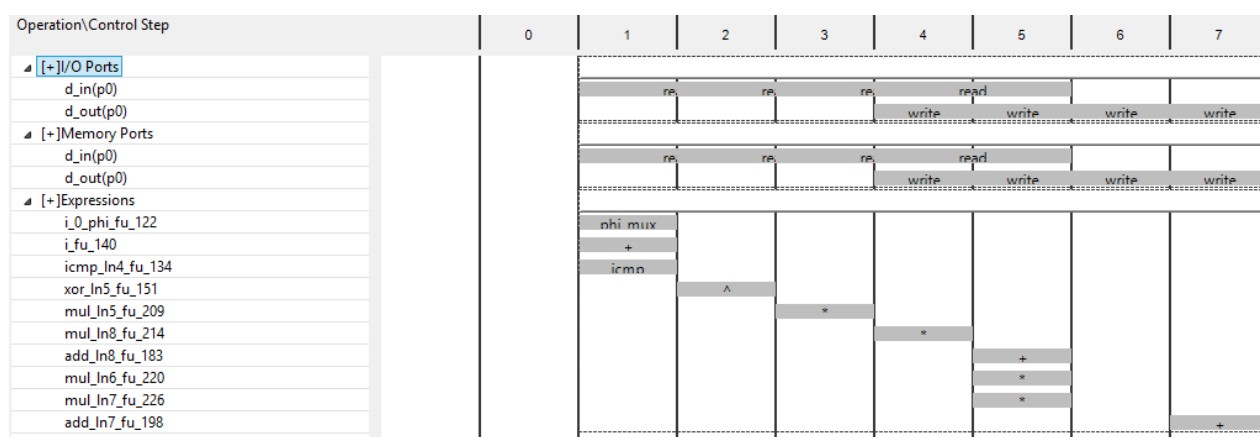


Рис. 3.3.3.5. Resource viewer

3.3.4. Анализ решения

Устройство использует 4 умножителя и имеет большие задержки, потому что 4 чтения выполняются последовательно.

3.4.Решение 2b

3.4.1. Директивы

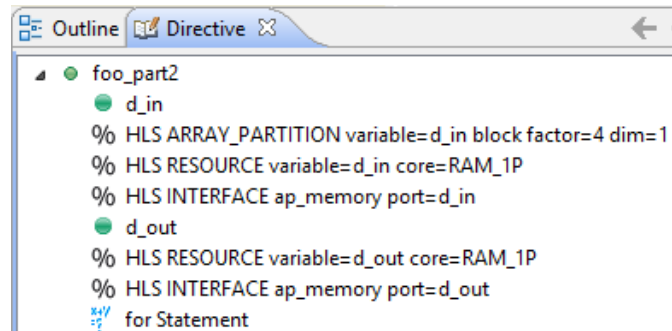


Рис. 3.4.1. Directives

3.4.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
25	25	0.250 us	0.250 us	25	25	none

Detail

Рис. 3.4.2. Performance estimates

3.4.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 149 LUT и 112 триггеров.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	45	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	104	-
Register	-	-	112	-	-
Total	0	4	112	149	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	~0	1	0

Рис. 3.4.3.1. Utilization estimates

Performance Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo_part2	-	25	-	26	-
Loop 1	no	24	6	-	4

Рис. 3.4.3.2. Performance profile

Resource Profile								
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth
foo_part2	0	4	112	149				
I/O Ports(5)					80			
Instances(0)	0	0	0	0				
Memories(0)	0		0	0	0			0
Expressions(4)	0	0	0	45	12	9	0	
Registers(10)			112		173			
Channels(0)	0		0	0	0			0
Multiplexers(4)	0		0	104	23			0
DSP(4)		4						

Рис. 3.4.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_address0	out	3	ap_memory	d_out	array
d_out_ce0	out	1	ap_memory	d_out	array
d_out_we0	out	1	ap_memory	d_out	array
d_out_d0	out	16	ap_memory	d_out	array

Рис. 3.4.3.4. Interfaces

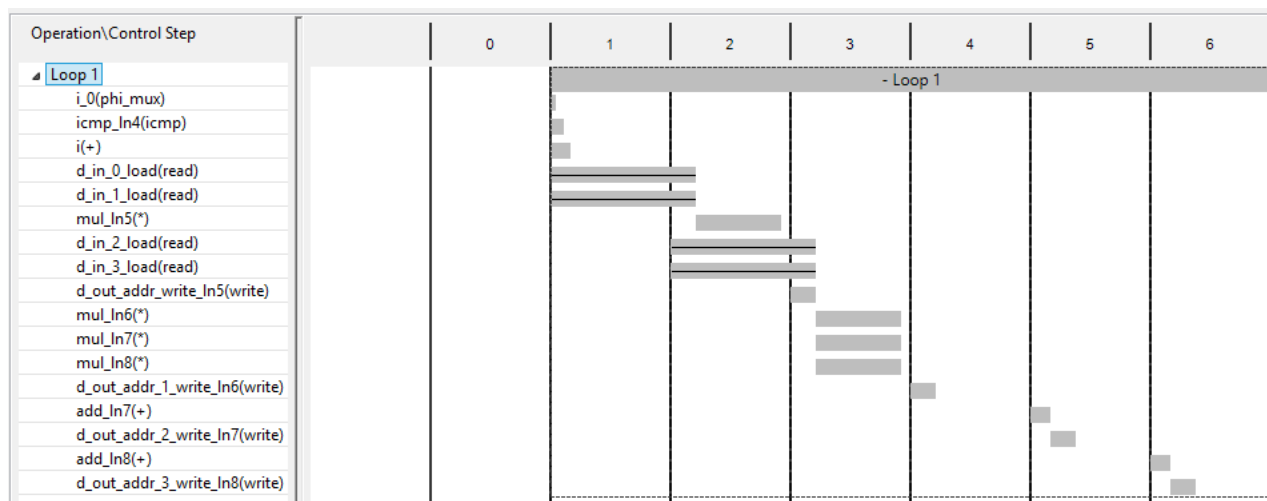


Рис. 3.4.3.5. Schedule viewer

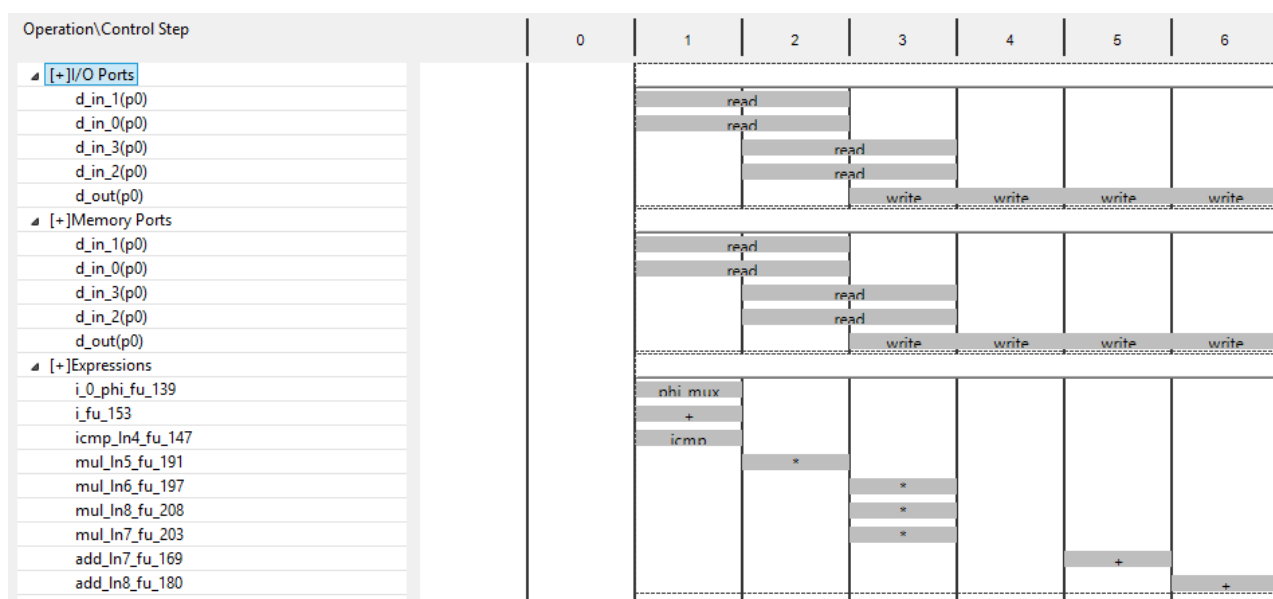


Рис. 3.4.3.5. Resource viewer

3.4.4. Анализ решения

В данном решении используется такое же количество умножителей, как и в предыдущем, однако за счёт возможности двух параллельных чтений сократились задержки (цикл за 6 тактов, а не за 7).

3.5.Решение 3b

3.5.1. Директивы

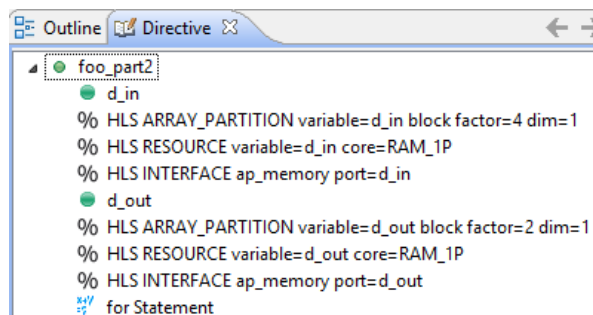


Рис. 3.5.1. Directives

3.5.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
25	25	0.250 us	0.250 us	25	25	none

Рис. 3.5.2. Performance estimates

3.5.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 214 LUT и 159 триггеров.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	68	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	146	-
Register	-	-	159	-	-
Total	0	4	159	214	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	~0	2	0

Рис. 3.5.3.1. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo_part2	-	25	-	26	-
Loop 1	no	24	6	-	4

Рис. 3.5.3.2. Performance profile

Performance Profile		Resource Profile							
		BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth
foo_part2		0	4	159	214				
I/O Ports(6)						96			
Instances(0)		0	0	0	0				
Memories(0)		0		0	0	0			0
Expressions(7)		0	0	0	68	17	16	0	
Registers(21)				159		220			
Channels(0)		0		0	0	0			0
Multiplexers(6)		0		0	146	40			0
DSP(4)			4						

Рис. 3.5.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_0_address0	out	2	ap_memory	d_out_0	array
d_out_0_ce0	out	1	ap_memory	d_out_0	array
d_out_0_we0	out	1	ap_memory	d_out_0	array
d_out_0_d0	out	16	ap_memory	d_out_0	array
d_out_1_address0	out	2	ap_memory	d_out_1	array
d_out_1_ce0	out	1	ap_memory	d_out_1	array
d_out_1_we0	out	1	ap_memory	d_out_1	array
d_out_1_d0	out	16	ap_memory	d_out_1	array

Рис. 3.5.3.4. Interfaces



Рис. 3.5.3.5. Schedule viewer

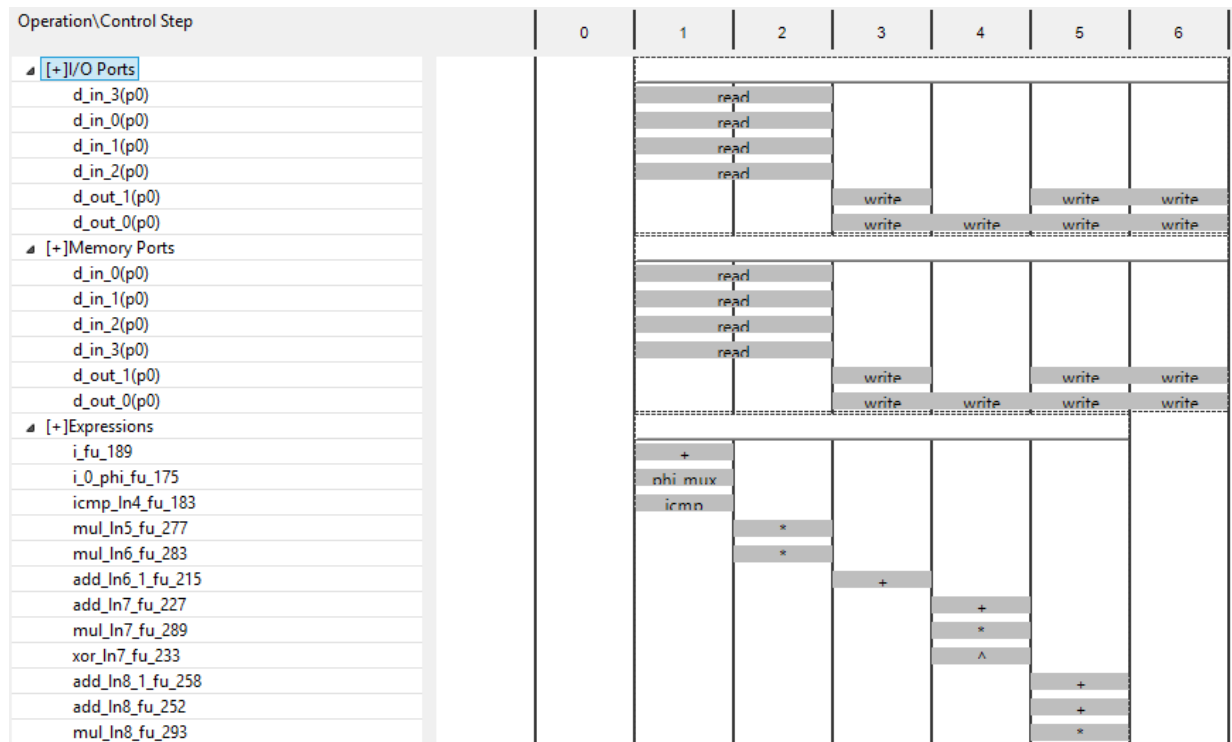


Рис. 3.5.3.5. Resource viewer

3.5.4. Анализ решения

В данном решении 4 чтения выполняются одновременно, однако итерация всё равно выполняется 6 тактов. Это связано с тем, что устройство не может одновременно писать 4 значения в выходной массив.

3.6.Решение 4b

3.6.1. Директивы

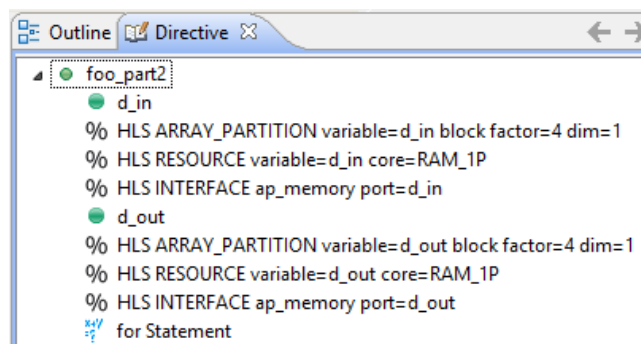


Рис. 3.6.1. Directives

3.6.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
29	29	0.290 us	0.290 us	29	29	none

Рис. 3.6.2. Performance estimates

3.6.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 225 LUT и 161 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	64	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	161	-
Register	-	-	161	-	-
Total	0	4	161	225	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	1	2	0

Рис. 3.6.3.1. Utilization estimates

Performance Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo_part2	-	29	-	30	-
Loop 1	no	28	7	-	4

Рис. 3.6.3.2. Performance profile

Resource Profile							
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2
foo_part2	0	4	161	225			
I/O Ports(8)					128		
Instances(0)	0	0	0	0			
Memories(0)	0		0	0	0		
Expressions(7)	0	0	0	64	18	14	0
Registers(26)			161		348		
Channels(0)	0		0	0	0		
Multiplexers(8)	0		0	161	55		
DSP(4)		4					

Рис. 3.6.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_0_address0	out	1	ap_memory	d_out_0	array
d_out_0_ce0	out	1	ap_memory	d_out_0	array
d_out_0_we0	out	1	ap_memory	d_out_0	array
d_out_0_d0	out	16	ap_memory	d_out_0	array
d_out_1_address0	out	1	ap_memory	d_out_1	array
d_out_1_ce0	out	1	ap_memory	d_out_1	array

Рис. 3.6.3.4. Interfaces

d_out_0_ce0	out	1	ap_memory	d_out_0	array
d_out_0_we0	out	1	ap_memory	d_out_0	array
d_out_0_d0	out	16	ap_memory	d_out_0	array
d_out_1_address0	out	1	ap_memory	d_out_1	array
d_out_1_ce0	out	1	ap_memory	d_out_1	array
d_out_1_we0	out	1	ap_memory	d_out_1	array
d_out_1_d0	out	16	ap_memory	d_out_1	array
d_out_2_address0	out	1	ap_memory	d_out_2	array
d_out_2_ce0	out	1	ap_memory	d_out_2	array
d_out_2_we0	out	1	ap_memory	d_out_2	array
d_out_2_d0	out	16	ap_memory	d_out_2	array
d_out_3_address0	out	1	ap_memory	d_out_3	array
d_out_3_ce0	out	1	ap_memory	d_out_3	array
d_out_3_we0	out	1	ap_memory	d_out_3	array
d_out_3_d0	out	16	ap_memory	d_out_3	array

Рис. 3.6.3.5. Interfaces continue

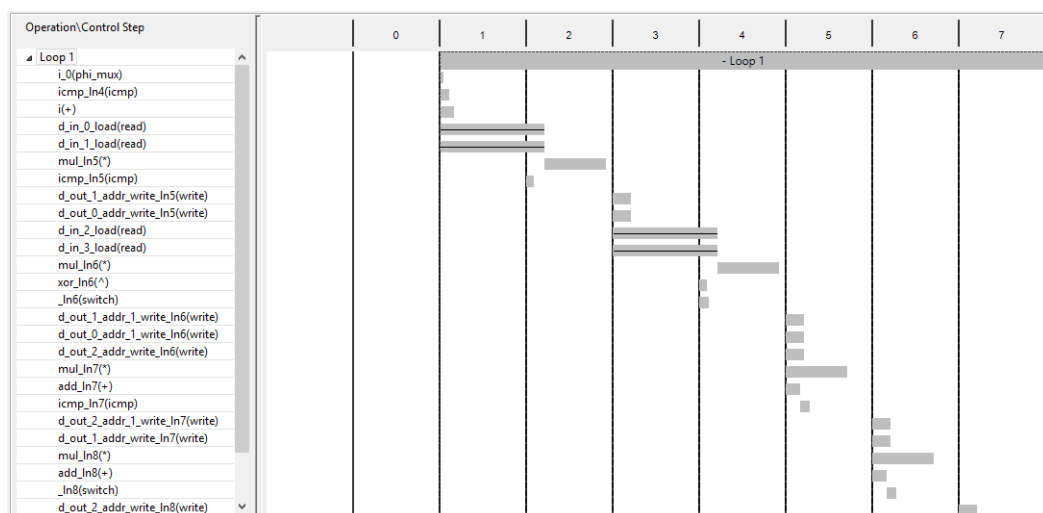


Рис. 3.6.3.6. Schedule viewer

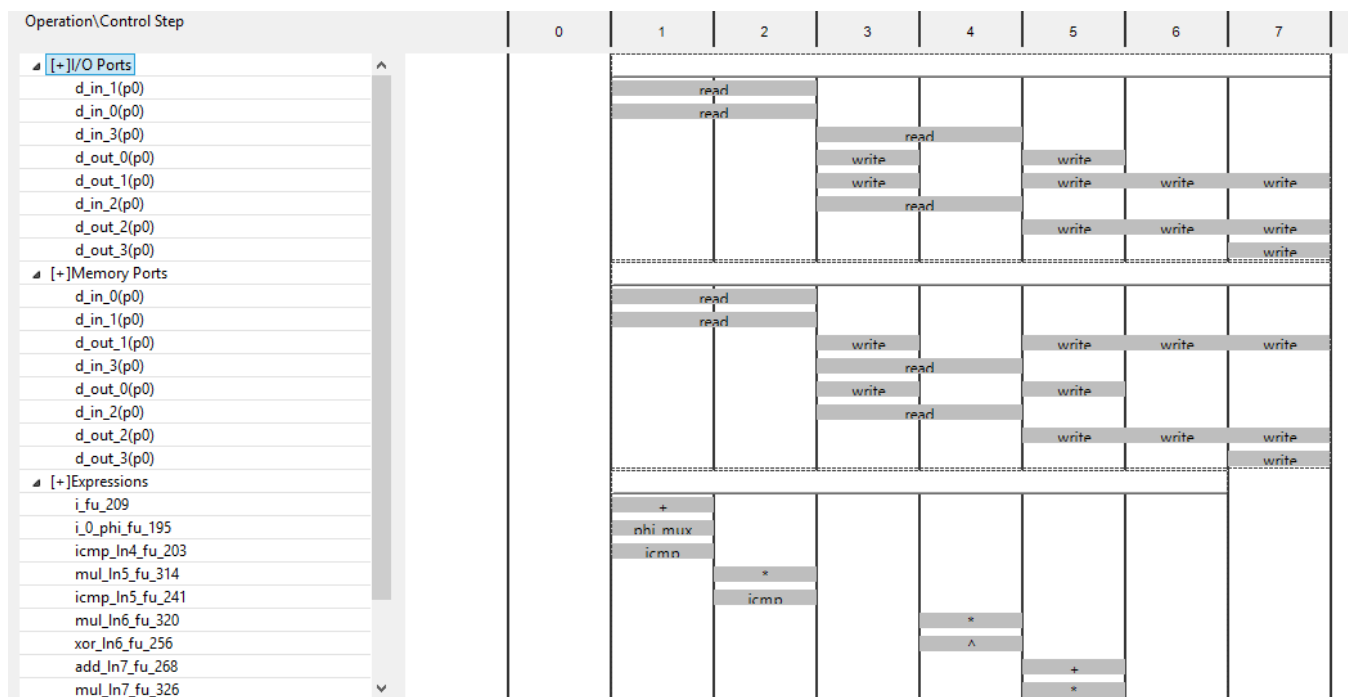


Рис. 3.6.3.7. Resource viewer

3.6.4. Анализ решения

В данном решении всё стало хуже, при возможности на запись 4 памяти. В устройстве используется очень много перезаписей в регистры, а не сразу в выходные линии, что не дает ни делать 4 чтения, ни записывать 4 значения на выходы.

3.7.Решение 5b

3.7.1. Директивы

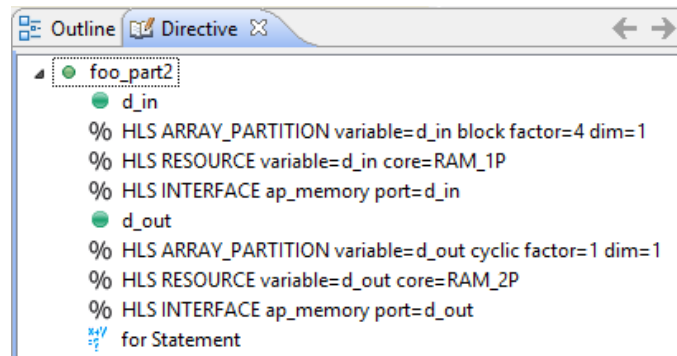


Рис. 3.7.1. Directives

3.7.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
25	25	0.250 us	0.250 us	25	25	none

Рис. 3.7.2. Performance estimates

3.7.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 149 LUT и 112 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	45	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	104	-
Register	-	-	112	-	-
Total	0	4	112	149	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	~0	1	0

Рис. 3.7.3.1. Utilization estimates

Performance Profile					
Resource Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo_part2	-	25	-	26	-
Loop 1	no	24	6	-	4

Рис. 3.7.3.2. Performance profile

Performance Profile							
Resource Profile							
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2
foo_part2	0	4	112	149			
I/O Ports(5)					80		
Instances(0)	0	0	0	0			
Memories(0)	0		0	0	0		
Expressions(4)	0	0	0	45	12	9	0
Registers(10)			112		173		
Channels(0)	0		0	0	0		
Multiplexers(4)	0		0	104	23		
DSP(4)		4					

Рис. 3.7.3.3. Resource profile

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_address1	out	3	ap_memory	d_out	array
d_out_ce1	out	1	ap_memory	d_out	array
d_out_we1	out	1	ap_memory	d_out	array
d_out_d1	out	16	ap_memory	d_out	array

Рис. 3.7.3.4. Interfaces



Рис. 3.7.3.5. Schedule viewer

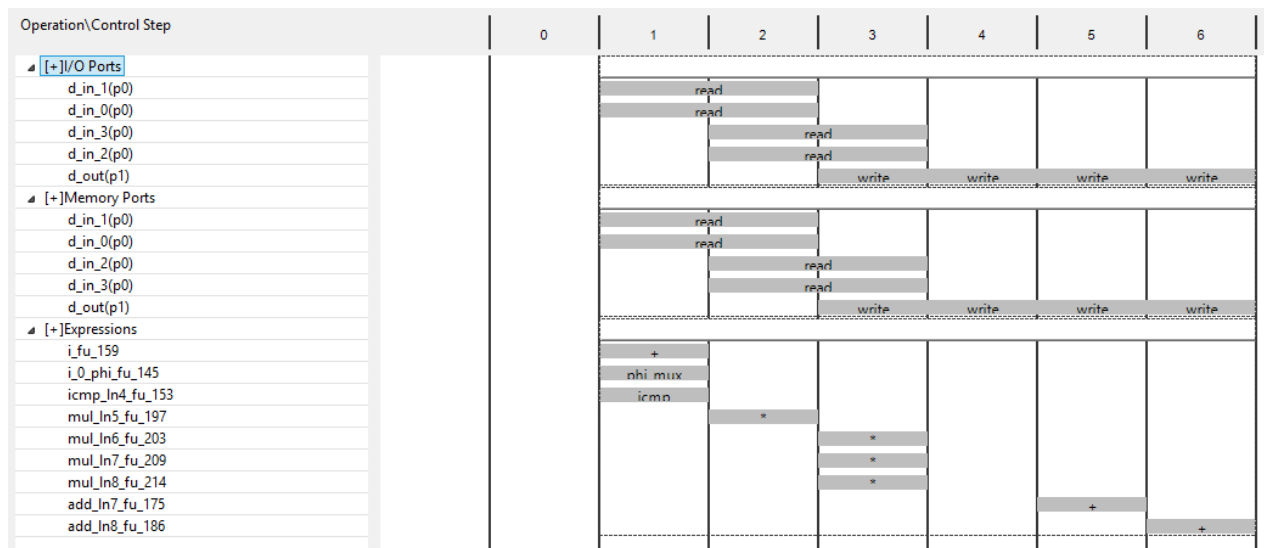


Рис. 3.7.3.6. Resource viewer

3.7.4. Анализ решения

В данном решении была использована одна двух портовая память, и это вышло эффективнее, чем 4 одно-портовых. Но одновременно 2 записи всё также не производятся.

3.8. Решение 6b

3.8.1. Директивы

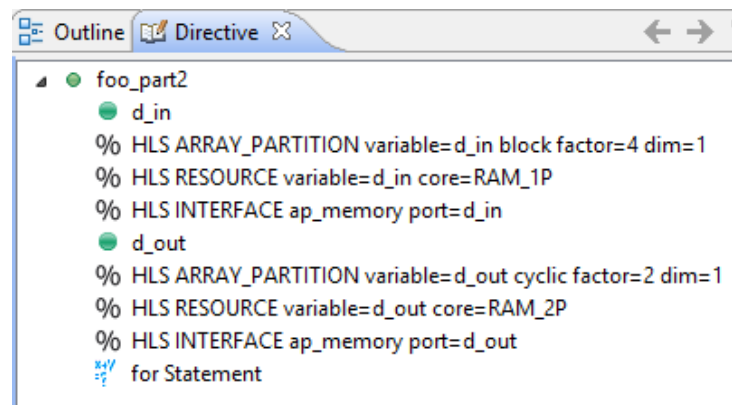


Рис. 3.8.1. Directives

3.8.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
29	29	0.290 us	0.290 us	29	29	none

Рис. 3.8.2. Performance estimates

3.8.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 206 LUT и 164 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	45	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	161	-
Register	-	-	164	-	-
Total	0	4	164	206	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	1	2	0

Рис. 3.8.3.1. Utilization estimates

Performance Profile					
Resource Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo_part2	-	29	-	30	-
Loop 1	no	28	7	-	4

Рис. 3.8.3.2. Performance profile

Resource Profile								
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth
foo_part2	0	4	164	206				
I/O Ports(6)					96			
Instances(0)	0	0	0	0				
Memories(0)	0		0	0	0			0
Expressions(4)	0	0	0	45	12	9	0	
Registers(22)			164		225			
Channels(0)	0		0	0	0			0
Multiplexers(6)	0		0	161	40			0
DSP(4)		4						

Рис. 3.8.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_0_address1	out	2	ap_memory	d_out_0	array
d_out_0_ce1	out	1	ap_memory	d_out_0	array
d_out_0_we1	out	1	ap_memory	d_out_0	array
d_out_0_d1	out	16	ap_memory	d_out_0	array
d_out_1_address1	out	2	ap_memory	d_out_1	array
d_out_1_ce1	out	1	ap_memory	d_out_1	array
d_out_1_we1	out	1	ap_memory	d_out_1	array
d_out_1_d1	out	16	ap_memory	d_out_1	array

Рис. 3.8.3.4. Interfaces

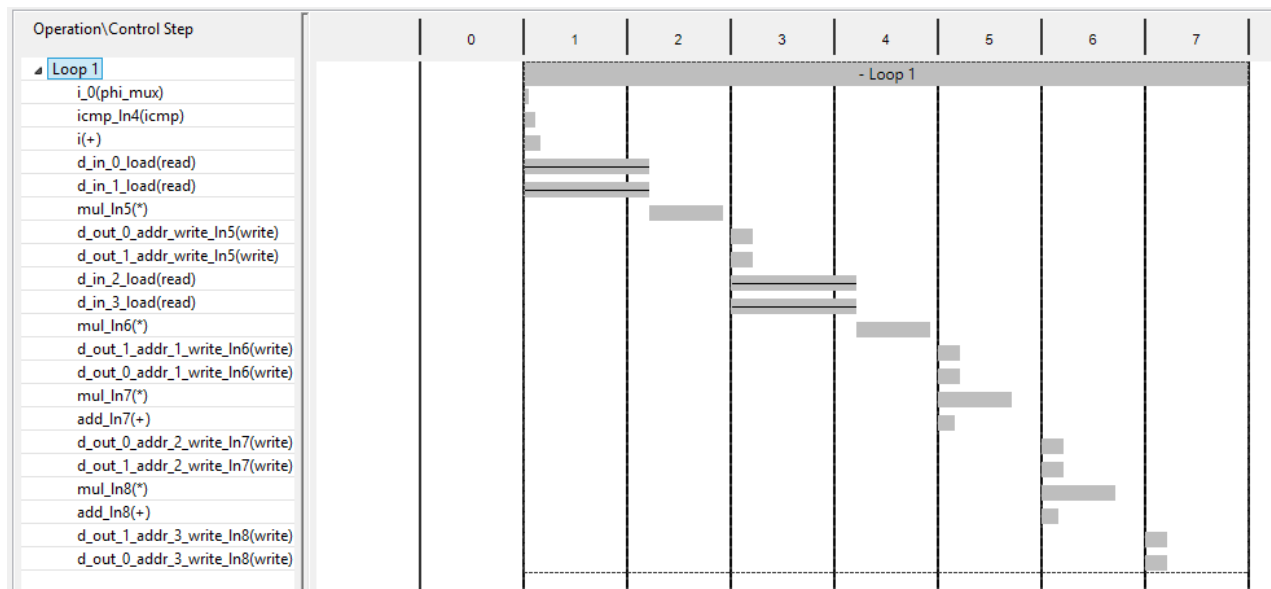


Рис. 3.8.3.5. Schedule viewer

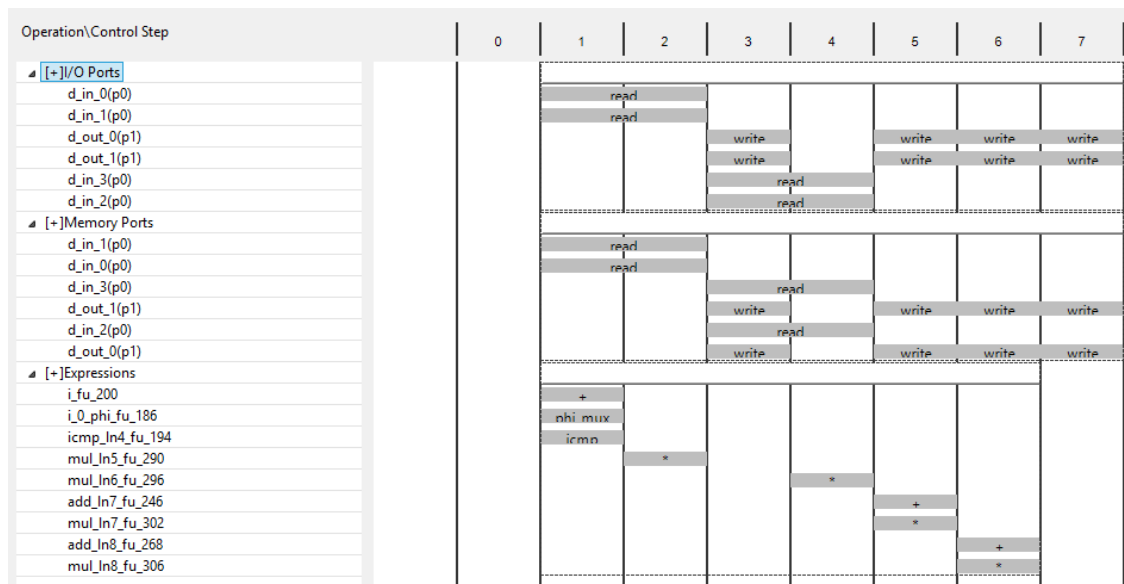


Рис. 3.8.3.6. Resource viewer

3.8.4. Анализ решения

В данном решении 4 чтения выполняются одновременно, а также присутствует одновременная запись в выходной массив, но задержки остались такими же.

3.9. Решение 7b

3.9.1. Директивы

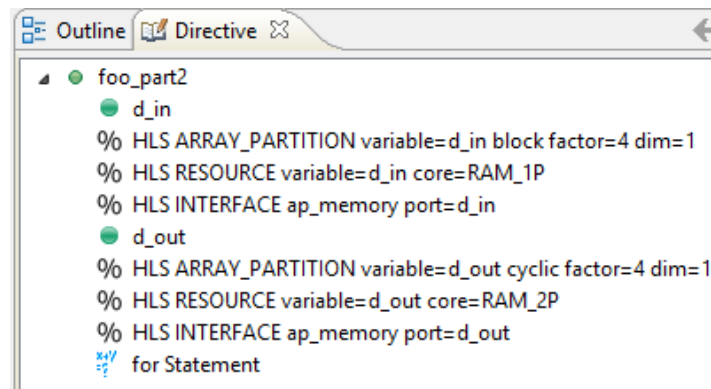


Рис. 3.9.1. Directives

3.9.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		Type
min	max	min	max	min	max	
13	13	0.130 us	0.130 us	13	13	none

Рис. 3.9.2. Performance estimates

3.9.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 297 LUT и 89 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	45	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	252	-
Register	-	-	89	-	-
Total	0	4	89	297	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	~0	3	0

Рис. 3.9.3.1. Utilization estimates

Performance Profile		Resource Profile				
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count	
foo_part2	-	13	-	14	-	
Loop 1	no	12	3	-	4	

Рис. 3.9.3.2. Performance profile

Performance Profile		Resource Profile						
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth
foo_part2	0	4	89	297				
I/O Ports(8)					128			
Instances(0)	0	0	0	0				
Memories(0)	0	0	0	0	0			0
Expressions(4)	0	0	0	45	10	11	0	
Registers(21)			89		89			
Channels(0)	0		0	0	0			0
Multiplexers(10)	0		0	252	72			0
DSP(4)		4						

Рис. 3.9.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_0_address1	out	1	ap_memory	d_out_0	array
d_out_0_ce1	out	1	ap_memory	d_out_0	array
d_out_0_we1	out	1	ap_memory	d_out_0	array
d_out_0_d1	out	16	ap_memory	d_out_0	array
d_out_1_address1	out	1	ap_memory	d_out_1	array
d_out_1_ce1	out	1	ap_memory	d_out_1	array

Рис. 3.9.3.4. Interfaces

d_out_1_ce1	out	1	ap_memory	d_out_1	array
d_out_1_we1	out	1	ap_memory	d_out_1	array
d_out_1_d1	out	16	ap_memory	d_out_1	array
d_out_2_address1	out	1	ap_memory	d_out_2	array
d_out_2_ce1	out	1	ap_memory	d_out_2	array
d_out_2_we1	out	1	ap_memory	d_out_2	array
d_out_2_d1	out	16	ap_memory	d_out_2	array
d_out_3_address1	out	1	ap_memory	d_out_3	array
d_out_3_ce1	out	1	ap_memory	d_out_3	array
d_out_3_we1	out	1	ap_memory	d_out_3	array
d_out_3_d1	out	16	ap_memory	d_out_3	array

Рис. 3.9.3.5. Interfaces continue

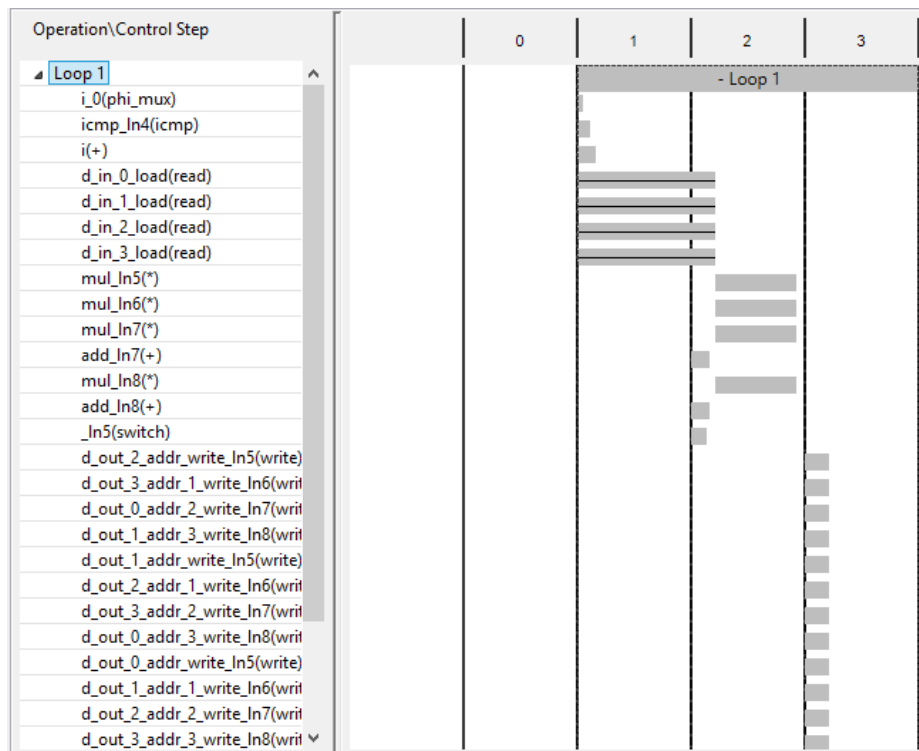


Рис. 3.9.3.6. Schedule viewer

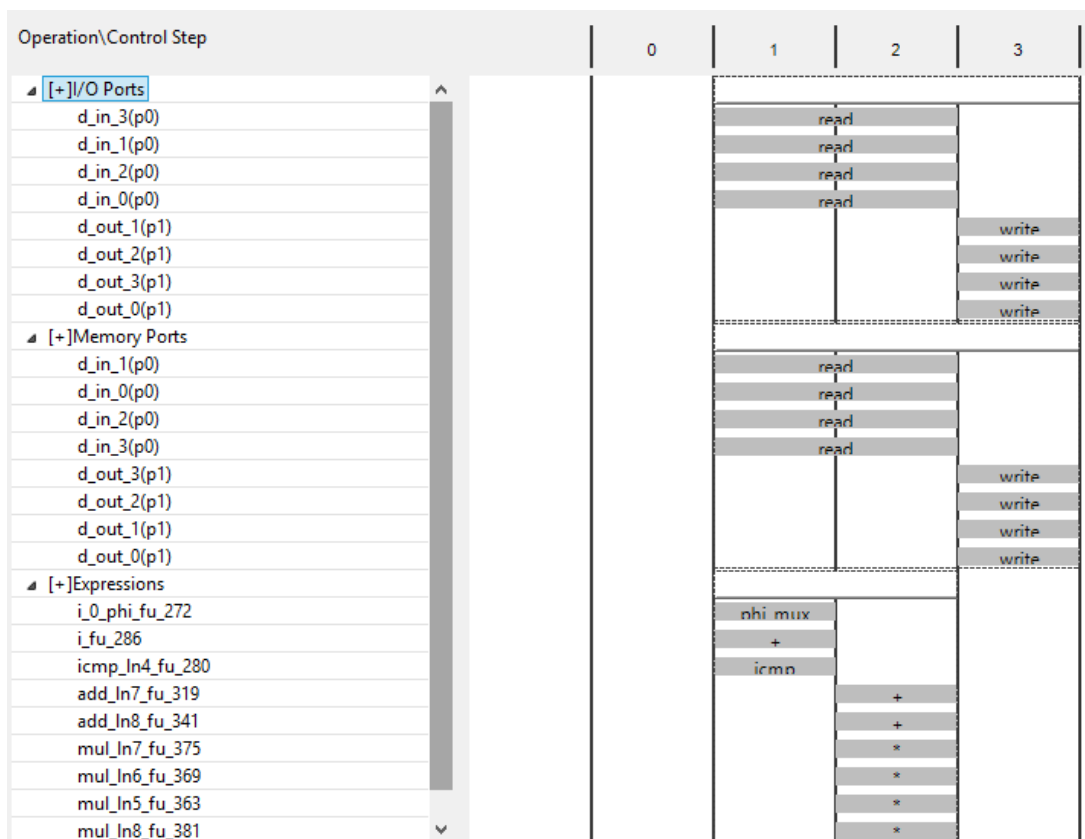


Рис. 3.9.3.7. Resource viewer

3.9.4. Анализ решения

Данное решение аналогично с RAM_1 хуже, чем предыдущее (с factor = 2).

3.10. Решение 8b

3.10.1. Директивы

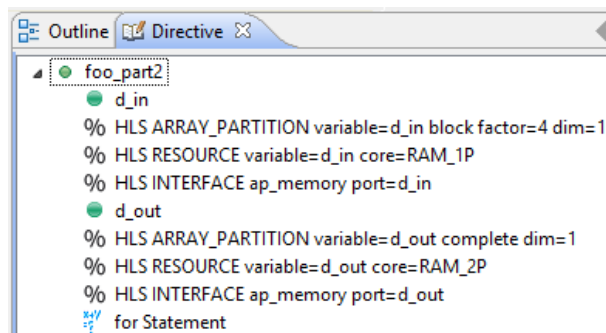


Рис. 3.10.1. Directives

3.10.2. Синтез

На изображении видно, что полученная задержка укладывается в заданное значение.

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	9.332 ns	0.10 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
25	25	0.250 us	0.250 us	25	25	none

Рис. 3.10.2. Performance estimates

3.10.3. Использование ресурсов

Оценка использования ресурсов показывает, что будут использованы 4 блока DSP48E, 149 LUT и 112 триггер.

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	4	-	-	-
Expression	-	-	0	45	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	104	-
Register	-	-	112	-	-
Total	0	4	112	149	0
Available	40	40	16000	8000	0
Utilization (%)	0	10	~0	1	0

Рис. 3.10.3.1. Utilization estimates

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo_part2	-	25	-	26	-
Loop 1	no	24	6	-	4

Рис. 3.10.3.2. Performance profile

Performance Profile		Resource Profile						
		BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2
foo_part2		0	4	112	149			
I/O Ports(5)						80		
Instances(0)		0	0	0	0			
Memories(0)		0		0	0	0		
Expressions(4)		0	0	0	45	12	9	0
Registers(10)				112		173		
Channels(0)		0		0	0	0		
Multiplexers(4)		0		0	104	23		
DSP(4)			4					

Рис. 3.10.3.3. Resource profile

Interface					
Summary					
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	foo_part2	return value
ap_rst	in	1	ap_ctrl_hs	foo_part2	return value
ap_start	in	1	ap_ctrl_hs	foo_part2	return value
ap_done	out	1	ap_ctrl_hs	foo_part2	return value
ap_idle	out	1	ap_ctrl_hs	foo_part2	return value
ap_ready	out	1	ap_ctrl_hs	foo_part2	return value
d_in_0_address0	out	2	ap_memory	d_in_0	array
d_in_0_ce0	out	1	ap_memory	d_in_0	array
d_in_0_q0	in	16	ap_memory	d_in_0	array
d_in_1_address0	out	2	ap_memory	d_in_1	array
d_in_1_ce0	out	1	ap_memory	d_in_1	array
d_in_1_q0	in	16	ap_memory	d_in_1	array
d_in_2_address0	out	2	ap_memory	d_in_2	array
d_in_2_ce0	out	1	ap_memory	d_in_2	array
d_in_2_q0	in	16	ap_memory	d_in_2	array
d_in_3_address0	out	2	ap_memory	d_in_3	array
d_in_3_ce0	out	1	ap_memory	d_in_3	array
d_in_3_q0	in	16	ap_memory	d_in_3	array
d_out_address1	out	3	ap_memory	d_out	array
d_out_ce1	out	1	ap_memory	d_out	array
d_out_we1	out	1	ap_memory	d_out	array
d_out_d1	out	16	ap_memory	d_out	array

Рис. 3.10.3.4. Interfaces



Рис. 3.10.3.5. Schedule viewer

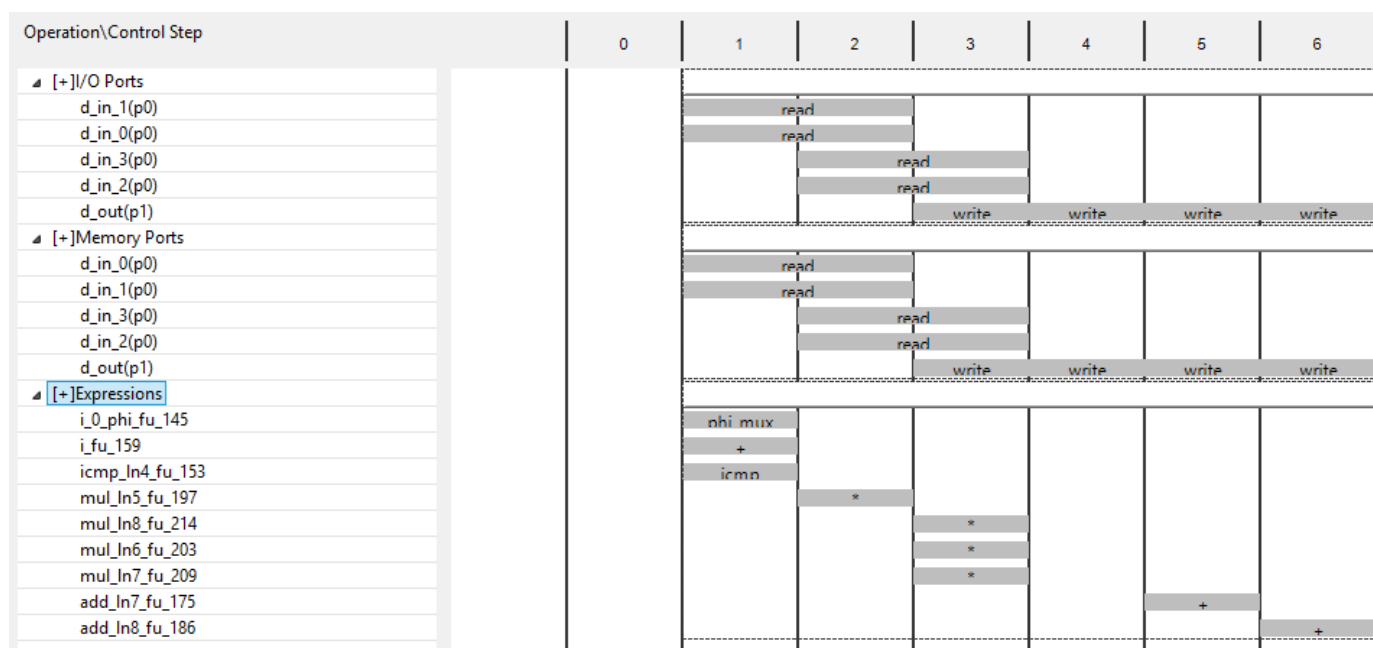


Рис. 3.10.3.6. Resource viewer

3.10.4. Анализ решения

В данном решении было проигнорировано разбиение выходного массива. Возможно, это связано с особенностью устройства, в общем случае это должно дать наивысшую скорость записи выходных данных.

3.11. Выводы

Ниже представлена таблица сравнения всех решений второй части.

Performance Estimates

Timing

Clock		solution1	solution2	solution3	solution4	solution5	solution6	solution7	solution8
ap_clk	Target	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns	10.00 ns
	Estimated	9.332 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns	9.332 ns

Latency

		solution1	solution2	solution3	solution4	solution5	solution6	solution7	solution8
Latency (cycles)	min	29	25	25	29	25	29	13	25
	max	29	25	25	29	25	29	13	25
Latency (absolute)	min	0.290 us	0.250 us	0.250 us	0.290 us	0.250 us	0.290 us	0.130 us	0.250 us
	max	0.290 us	0.250 us	0.250 us	0.290 us	0.250 us	0.290 us	0.130 us	0.250 us
Interval (cycles)	min	29	25	25	29	25	29	13	25
	max	29	25	25	29	25	29	13	25

По разбиению выходной памяти не всегда получается ожидаемый прирост производительности, а часто даже ухудшение. Согласно временным диаграммам, запись в выходную память почти никогда не производится параллельно для нескольких ячеек. Вероятно, это связано с особенностью устройства, где на выход подаются различные комбинации одних и тех же входов.