

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

Отчет по лабораторной работе №6_2
Курс: «Проектирование реконфигурируемых гибридных
вычислительных систем»
Тема: «Port-level IO protocols»

Выполнил студент гр. 3540901/81501

Селиверстов Я.А.

(подпись)

Руководитель

Антонов А.П.

(подпись)

“ ____ ” _____ 2019 г.

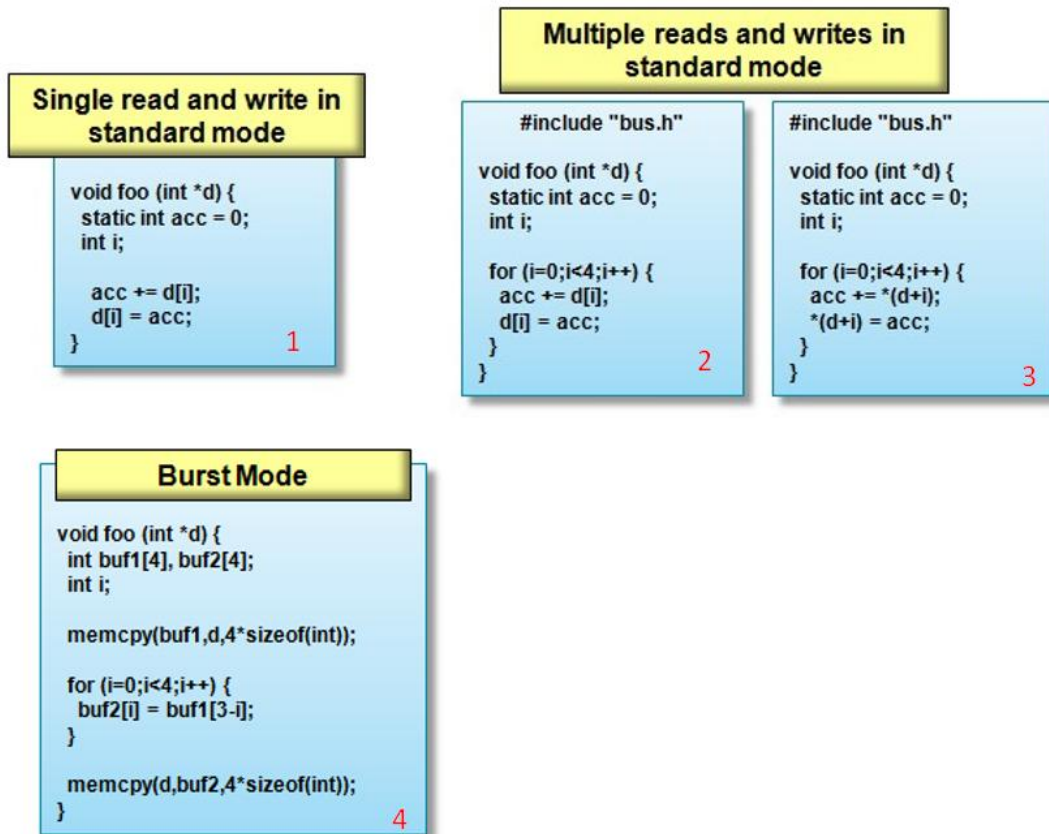
Санкт – Петербург
2019

ОГЛАВЛЕНИЕ

Задание	3
1. Первое решение.....	5
1.1. Исходный код программы и теста.....	5
1.2. Моделирование	6
1.3. Синтез.....	6
1.4. C RTL моделирование	8
2. Второе решение	10
2.1. Настройки второго решения	10
2.2. Моделирование	11
2.3. Синтез.....	11
2.4. C RTL моделирование	13
3. Третье решение	15
3.1. Настройки третьего решения.....	15
3.2. Моделирование	16
3.3. Синтез.....	16
3.4. C RTL моделирование	19
4. Четвертое решение	20
4.1. Настройки четвертого решения.....	20
4.2. Моделирование	21
4.3. Синтез.....	21
4.4. C RTL моделирование	23
4.Выводы.....	25

Задание

- Создать проект lab6_2
- Микросхема: xa7a12tcs325-1q
- Создать четыре функции на основе слайда (функция foo_1, foo_2, foo_3, foo_4).
При желании можно сделать 4 отдельные лабораторные работы lab6_2_1...lab6_2_4 но они все будут очень похожи.



- Создать тест lab6_2_test.c для проверки функций выше (это может быть один тест или разные тесты. Функция main д.б. одна, а в ней использовать проверяемый модуль. Д.б. вывод результатов в консоль.).
- Для каждой функции сделать свой solution
 - задать: clock period 10; clock_uncertainty 0.1
 - Задать протокол
 - a: ap_bus
 - осуществить моделирование (с выводом результатов в консоль)
 - осуществить синтез
 - привести в отчете:
 - performance estimates=>summary
 - utilization estimates=>summary
 - Performance Profile
 - interface estimates=>summary
 - объяснить какой интерфейс использован для блока (и какие сигналы входят) и для портов (и какие сигналы входят).
 - scheduler viewer (выполнить Zoom to Fit)

- На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
- resource viewer (выполнить Zoom to Fit)
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
- Осуществить C|RTL моделирование
 - Привести результаты из консоли
 - Открыть временную диаграмму (все сигналы)
 - Отобразить два цикла обработки на одном экране
 - На скриншоте показать Latency
 - На скриншоте показать Initiation Interval
- Выводы
 - Объяснить отличие процедур обращения к элементам массива для каждого случая

1. Первое решение

Clock

Period: Uncertainty:

Part Selection

Part: **xa7a12t-csg325-1Q**

Рисунок 1.1. Параметры первого решения

1.1. Исходный код программы и теста

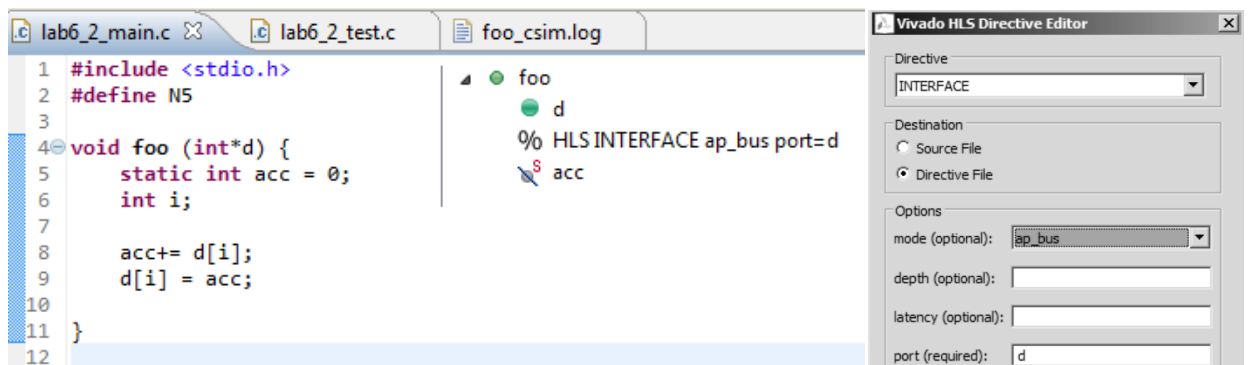


Рисунок 1.2. Исходный код синтезируемой функции

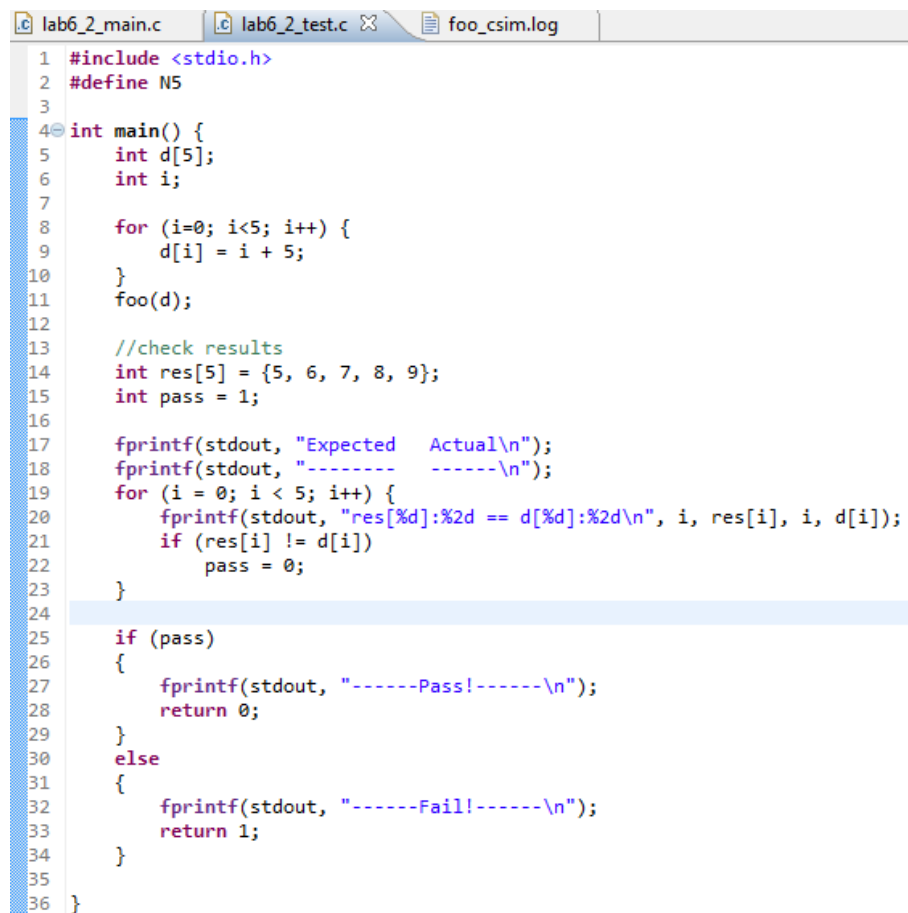


Рисунок 1.3. Исходный код теста

1.2. Моделирование



```
1 INFO: [SIM 2] ***** CSIM start *****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling(apcc) ../../../../../../Desktop/SPBGPU_Mag/Antonov/lab_6/lab6_z2/lab6_2_test.c in debug mode
4 INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
5 INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT_amd64 version 6.1) on Fri Jan 03 00:07:24 +0300 2020
6 INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab6_2/solution1/csim/build'
7 INFO: [APCC 202-3] Tmp directory is apcc_db
8 INFO: [APCC 202-1] APCC is done.
9   Generating csim.exe
10 Expected   Actual
11 -----
12 res[0]: 5 == d[0]: 5
13 res[1]: 6 == d[1]: 6
14 res[2]: 7 == d[2]: 7
15 res[3]: 8 == d[3]: 8
16 res[4]: 9 == d[4]: 9
17 -----Pass!-----
18 INFO: [SIM 1] CSim done with 0 errors.
19 INFO: [SIM 3] ***** CSIM finish *****
```

Рисунок 1.4. Результат успешного моделирования

Моделирование прошло успешно.

1.3. Синтез

Данные о проекте

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	9.900	0.10

Latency (clock cycles)

Summary

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

Рисунок 1.5. Performance estimates – summary

Здесь можно увидеть, что достигнутая задержка равна $9.900 + 0.1$, что укладывается в заданные нами требования к тактовой частоте.

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	39	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	48	-
Register	-	-	101	-	-
Total	0	0	101	87	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	~0	1	0

Рисунок 1.6. Utilization estimates – summary

Данный проект займет на микросхеме 101 регистр для хранения чисел и 87 LUT.

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_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Рисунок 1.7. Interface estimates – summary

Для расчета схемы требуется более одного такта. На рисунке 1.7 представлены интерфейсы, которые используются в синтезированном устройстве. Видно, что в схеме применяется протокол ap_bus. Порты d_address, d_datain, d_dataout и d_size 32-битные.

Performance Profile

Resource Profile

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	4	-	5	-

Рисунок 1.8. Performance Profile

На рисунке 1.8 видно, что задержка получения выходного значения составляет 4 такта с момента старта, а задержка после старта до готовности приема новых данных – 5.

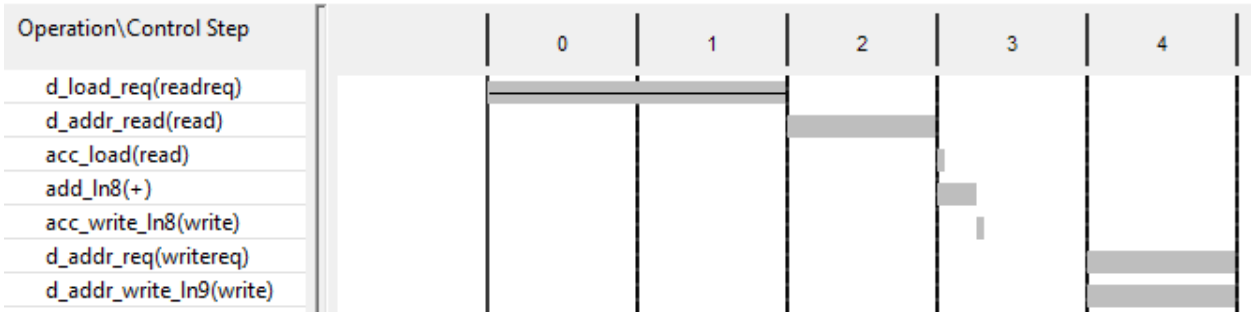


Рисунок 1.9. Schedule viewer

На рисунке 1.9 представлена диаграмма Schedule viewer. На ней видно, что величина Latency составляет 4 такта, а Iteration latency – 5 тактов.

Далее рассмотрим профиль ресурсов:

	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits*Banks
foo	0	0	101	87						
I/O Ports(1)					32					
Instances(0)	0	0	0	0						
Memories(0)	0		0	0	0			0	0	0
Expressions(1)	0	0	0	39	32	32	0			
Registers(4)			101		101					
Channels(0)	0		0	0	0			0	0	0
Multiplexers(2)	0		0	48	33			0		
DSP(0)		0								

Рисунок 1.10. Resource Profile.

Значения в Resource Profile на рис. 1.10 совпадают с результатами синтеза на рис. 1.6.

1.4. C|RTL моделирование

Cosimulation Report for 'foo'							
Result							
		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	4	4	4	NA	NA	NA

Рисунок 1.11. Cosimulation Report

При совместном моделировании, программа на рисунке 1.11 отобразила те же самые, ожидаемые нами значения Latency и П.

Покажем на рисунке 1.12 временную диаграмму совместного моделирования с отмеченными на ней Latency и П:

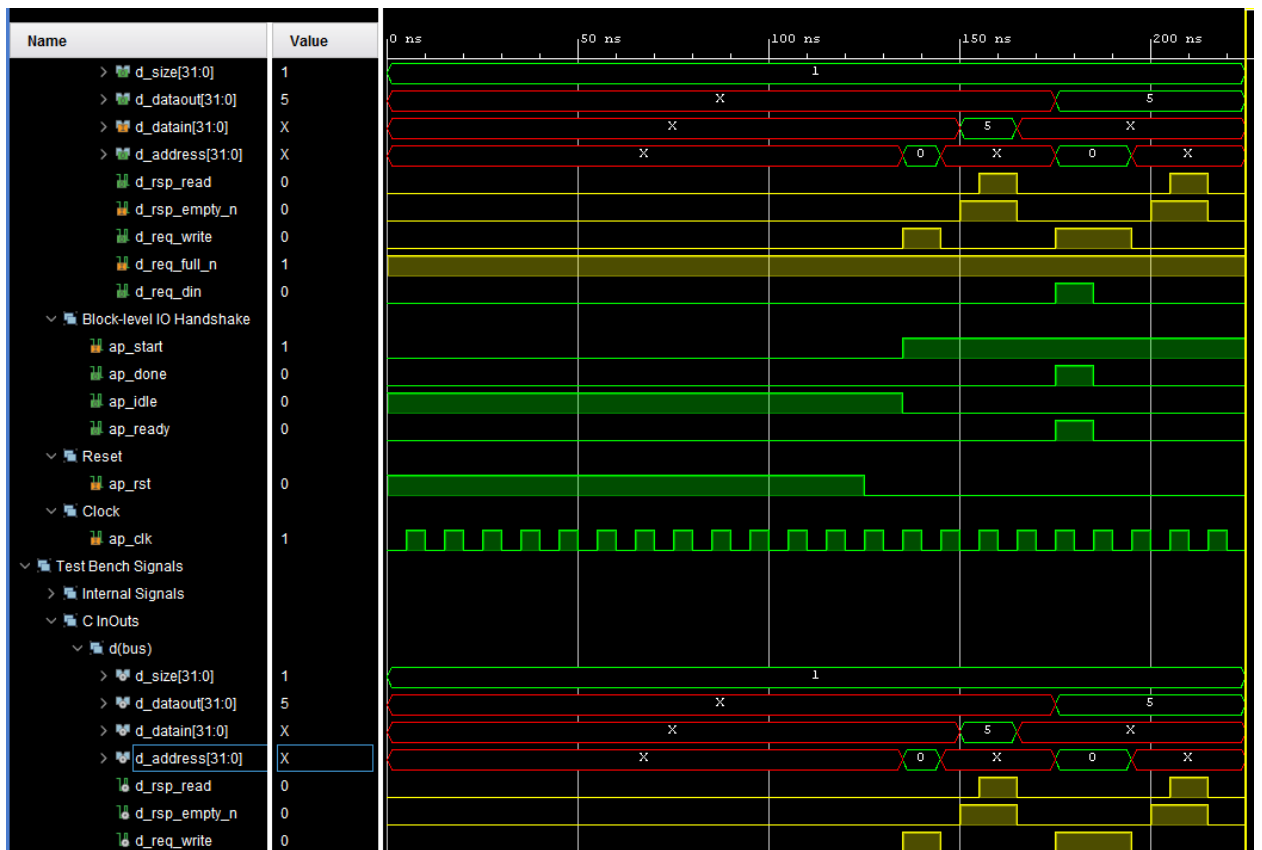
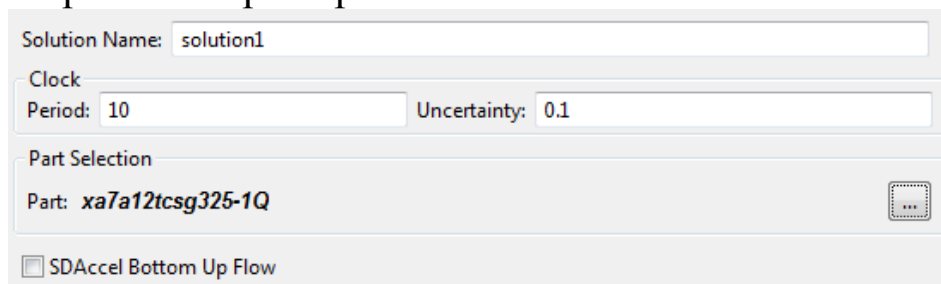


Рисунок 1.12. Design Top Signals

2. Второе решение

2.1. Настройки второго решения



Solution Name: solution1

Clock
Period: 10 Uncertainty: 0.1

Part Selection
Part: xa7a12tcsq325-1Q

☐ SDAccel Bottom Up Flow

Рисунок 2.0. Параметры второго решения

Программа, тест к ней и интерфейс (директива) представлены на рисунках 2.1 и 2.2.

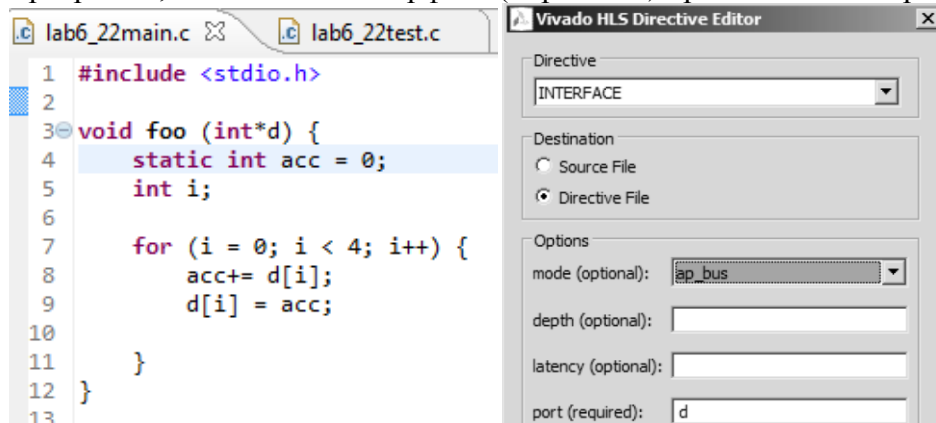


Рисунок 2.1 Исходная программа.

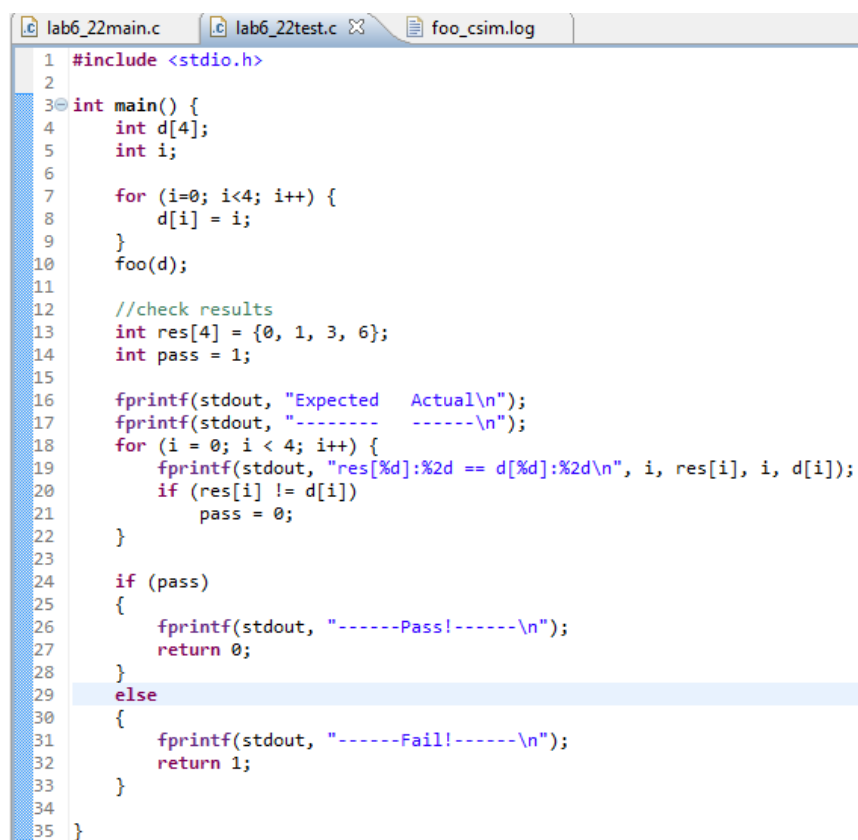
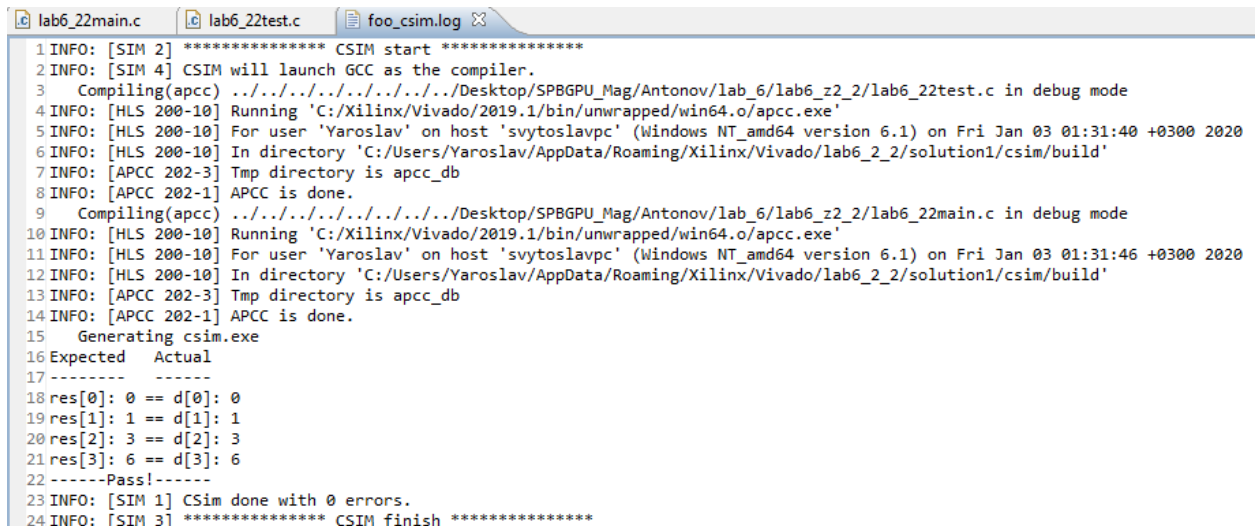


Рисунок 2.2. Тест к программе.

2.2. Моделирование



```
1 INFO: [SIM 2] ***** CSIM start *****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling(apcc) ../../../../../../Desktop/SPBGPU_Mag/Antonov/lab_6/lab6_z2_2/lab6_22test.c in debug mode
4 INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
5 INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT amd64 version 6.1) on Fri Jan 03 01:31:40 +0300 2020
6 INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab6_2_2/solution1/csim/build'
7 INFO: [APCC 202-3] Tmp directory is apcc_db
8 INFO: [APCC 202-1] APCC is done.
9   Compiling(apcc) ../../../../../../Desktop/SPBGPU_Mag/Antonov/lab_6/lab6_z2_2/lab6_22main.c in debug mode
10 INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
11 INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT amd64 version 6.1) on Fri Jan 03 01:31:46 +0300 2020
12 INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab6_2_2/solution1/csim/build'
13 INFO: [APCC 202-3] Tmp directory is apcc_db
14 INFO: [APCC 202-1] APCC is done.
15   Generating csim.exe
16 Expected  Actual
17 -----
18 res[0]: 0 == d[0]: 0
19 res[1]: 1 == d[1]: 1
20 res[2]: 3 == d[2]: 3
21 res[3]: 6 == d[3]: 6
22 -----Pass!-----
23 INFO: [SIM 1] CSim done with 0 errors.
24 INFO: [SIM 3] ***** CSIM finish *****
```

Рисунок 2.3 Результат успешного моделирования

Моделирование второго решения также прошло успешно.

2.3. Синтез

Performance Estimates

[-] Timing (ns)

[-] Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	9.900	0.10

[-] Latency (clock cycles)

[-] Summary

Latency		Interval		
min	max	min	max	Type
14	14	14	14	none

Рисунок 2.4. Performance estimates – summary

На рисунке 2.4. можно увидеть, что достигнутая задержка равна $9.900 + 0.1$, укладывается в заданные требования тактовой частоты. А величина Latency 14.

Использование ресурсов:

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	71	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	42	-
Register	-	-	108	-	-
Total	0	0	108	113	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	~0	1	0

Рисунок 2.5. Utilization estimates – summary

На рисунке 2.5. видно, что данный проект теперь займет на микросхеме 108 регистров для хранения чисел, и 113 LUT.

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_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Рисунок 2.6. Interface Summary.

Полученные значения Interface Summary на рисунке 2.6. совпадают с прошлым решением, представленным на рисунке 1.7. На рисунке 2.6 также представлены интерфейсы, которые используются в синтезированном устройстве. Видно, что в схеме применяется те же протоколы ap_bus. Порты d_adress, d_datain, d_dataout и d_size 32-битные.

Performance Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	14	-	15	-
Loop 1	no	12	3	-	4

Рисунок 2.7. Performance Profile

На рисунке 2.7. видно, что задержка получения первого выходного значения составляет 3 такта с момента старта, для остальных -14, а задержка после старта до готовности приема новых данных – 15:

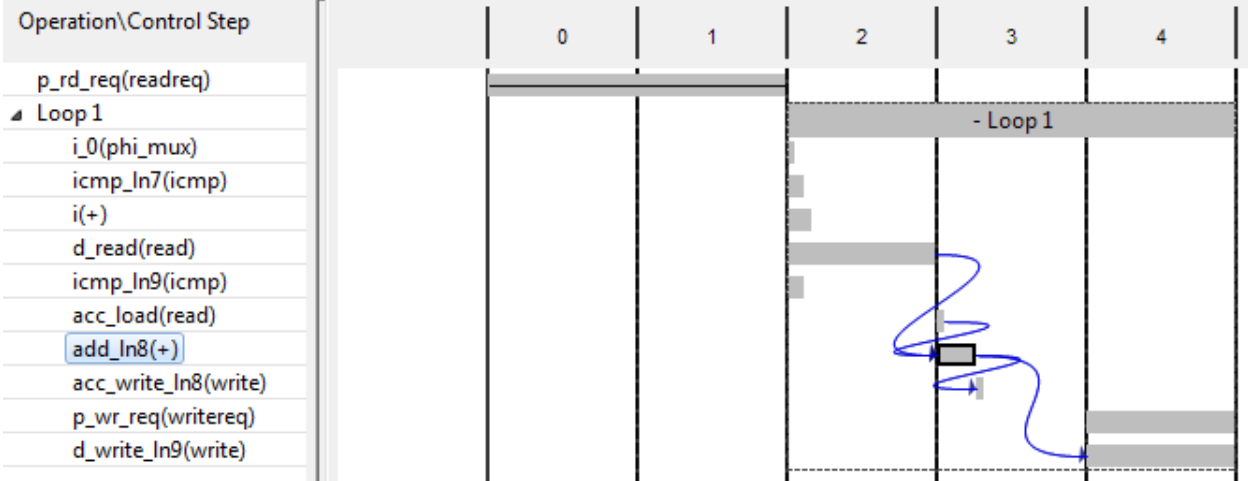


Рисунок 2.8. Schedule viewer

В отличие от предыдущего решения у нас присутствует цикл. За счет этого величина latency увеличилась до 14.

Рассмотрим профиль ресурсов:

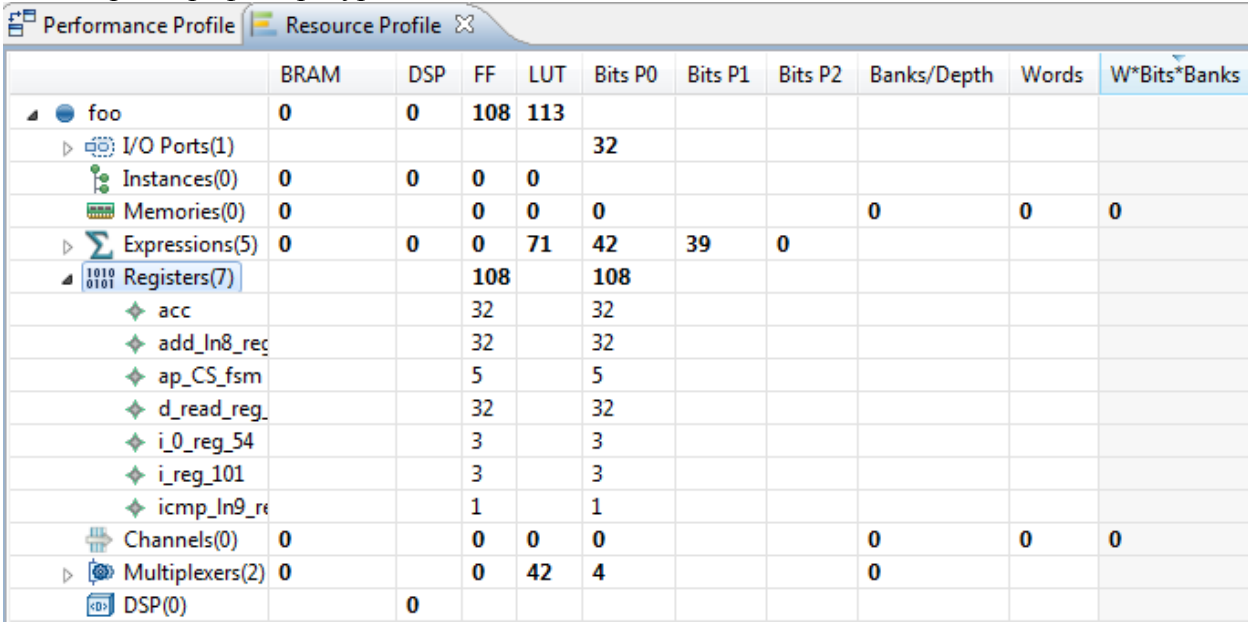


Рисунок 2.9. Resource Profile

Здесь мы также видим отличия, согласно общему отчету о затраченных ресурсах.

2.4. C|RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency

Cosimulation Report for 'foo'

Result

RTL	Status	Latency			Interval		
		min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	14	14	14	NA	NA	NA

Рисунок 2.10. Отчет о моделировании

Покажем временную диаграмму моделирования на рисунке 2.11.

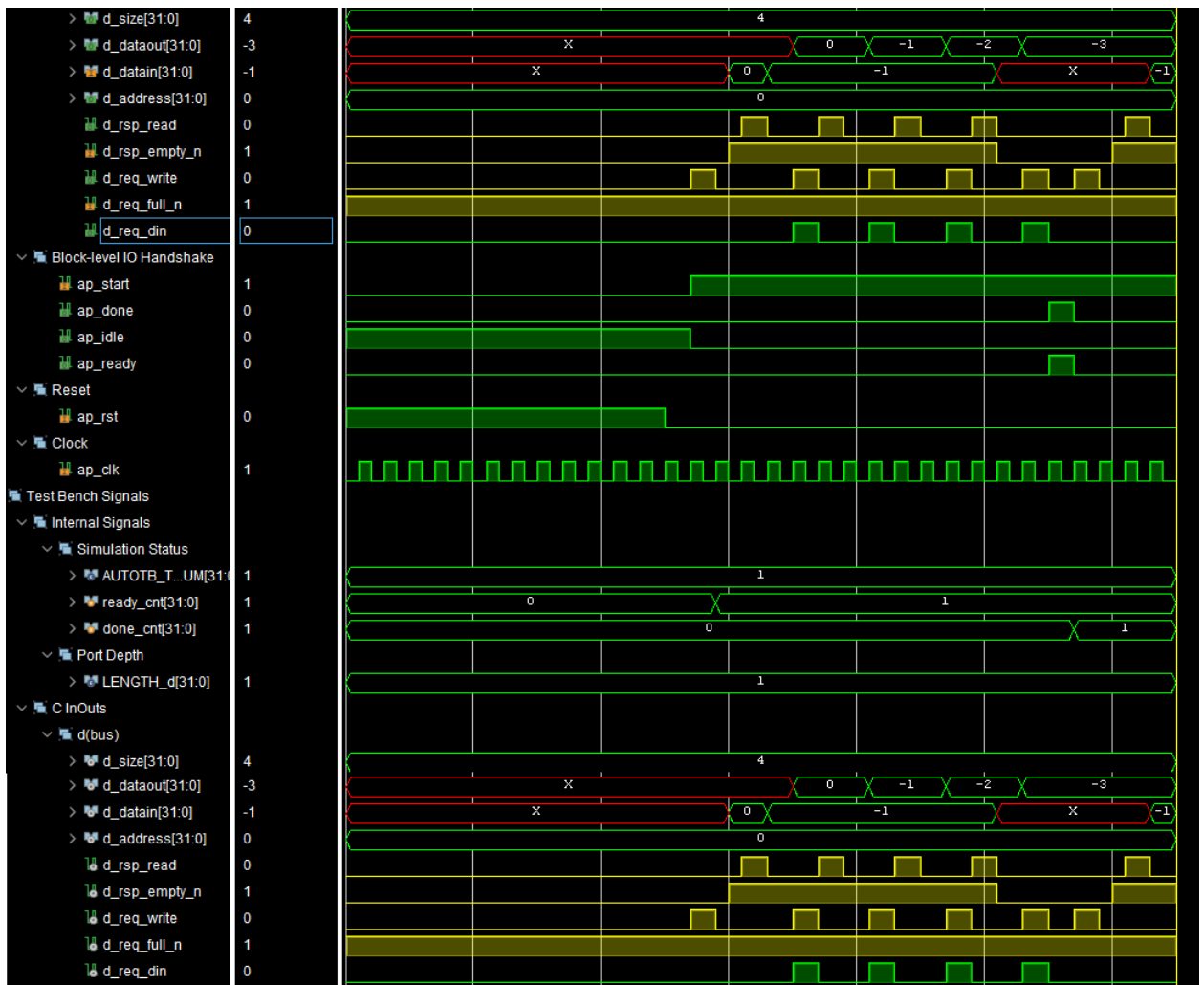


Рисунок 2.11. Design Top Signals

Здесь также видны отличия во времени выполнения итераций, задержка составила 14 в отличие от предыдущего решения, в котором она была равна 4.

3. Третье решение

3.1. Настройки третьего решения

Solution Name: solution1

Clock
Period: 10 Uncertainty: 0.1

Part Selection
Part: **x7a12tcs9325-1Q**

☐ SDAccel Bottom Up Flow

Рисунок 3.0. Параметры второго решения

Программа, тест к ней и интерфейс (директива) представлены на рисунках 3.1 и 3.2.

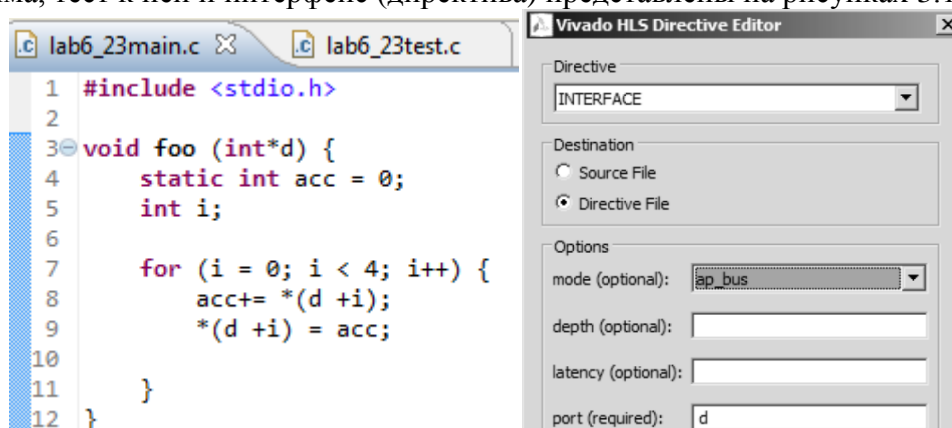


Рисунок 3.1 Исходная программа.

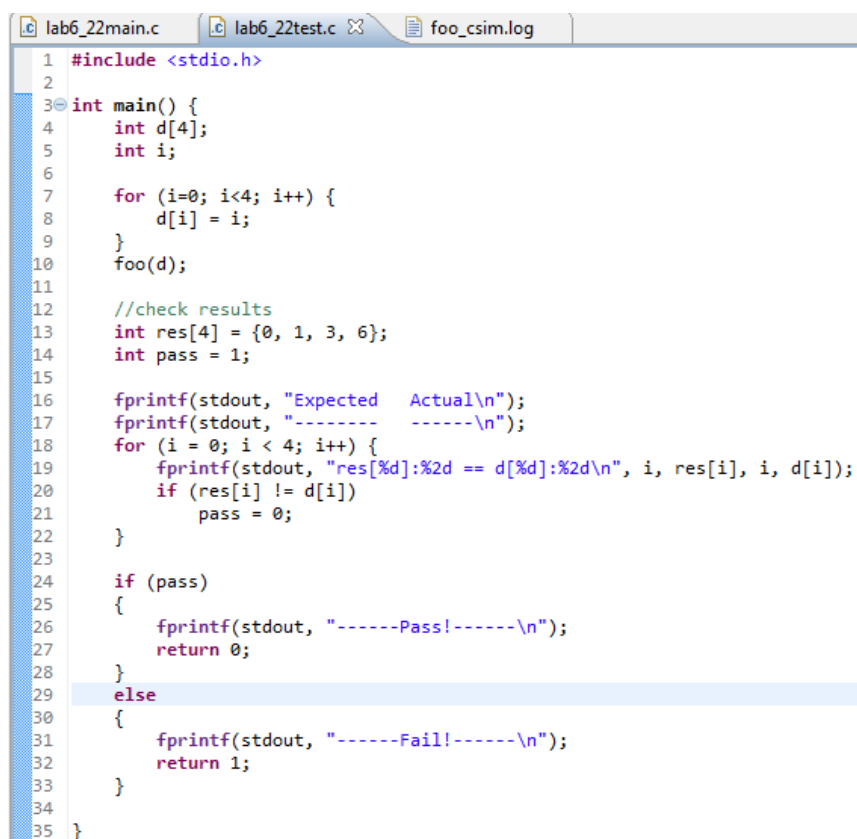
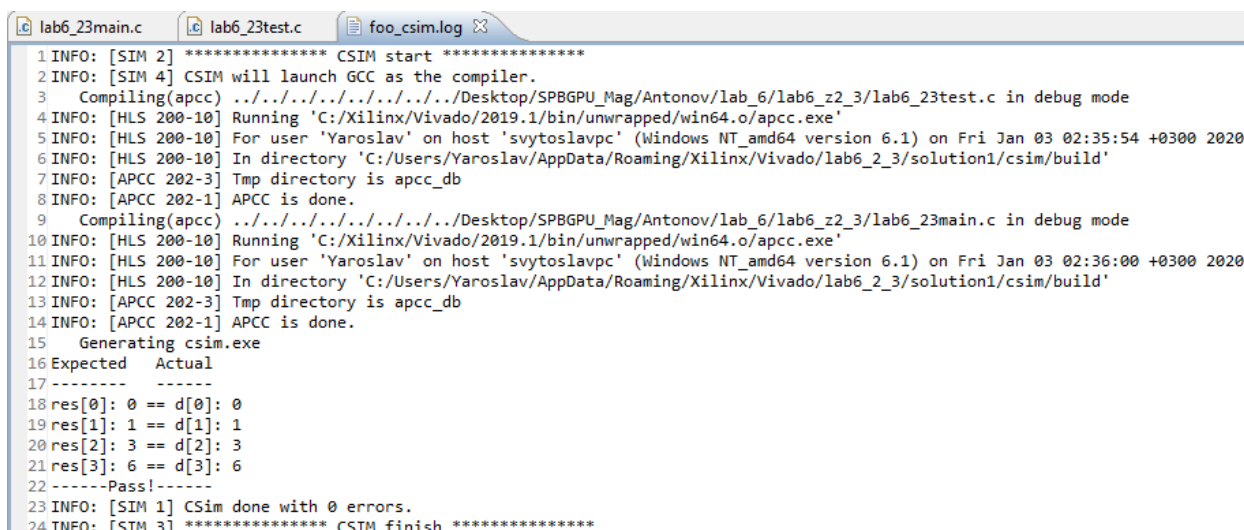


Рисунок 3.2. Тест к программе.

3.2. Моделирование



```

1 INFO: [SIM 2] ***** CSIM start *****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling(apcc) ../../../../../../Desktop/SPBGPU_Mag/Antonov/lab_6/lab6_z2_3/lab6_23test.c in debug mode
4 INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
5 INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT_amd64 version 6.1) on Fri Jan 03 02:35:54 +0300 2020
6 INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab6_2_3/solution1/csim/build'
7 INFO: [APCC 202-3] Tmp directory is apcc_db
8 INFO: [APCC 202-1] APCC is done.
9   Compiling(apcc) ../../../../../../Desktop/SPBGPU_Mag/Antonov/lab_6/lab6_z2_3/lab6_23main.c in debug mode
10 INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
11 INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT_amd64 version 6.1) on Fri Jan 03 02:36:00 +0300 2020
12 INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab6_2_3/solution1/csim/build'
13 INFO: [APCC 202-3] Tmp directory is apcc_db
14 INFO: [APCC 202-1] APCC is done.
15   Generating csim.exe
16 Expected Actual
17 -----
18 res[0]: 0 == d[0]: 0
19 res[1]: 1 == d[1]: 1
20 res[2]: 3 == d[2]: 3
21 res[3]: 6 == d[3]: 6
22 -----Pass!-----
23 INFO: [SIM 1] CSim done with 0 errors.
24 INFO: [SIM 3] ***** CSIM finish *****
  
```

Рисунок 3.3 Результат успешного моделирования

Моделирование второго решения также прошло успешно.

3.3. Синтез

Performance Estimates

[-] Timing (ns)

[-] Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	9.900	0.10

[-] Latency (clock cycles)

[-] Summary

Latency		Interval		
min	max	min	max	Type
14	14	14	14	none

Рисунок 3.4. Performance estimates – summary

На рисунке 3.4. можно увидеть, что достигнутая задержка равна $9.900 + 0.1$, укладывается в заданные требования тактовой частоты и совпадает со вторым решением . Величина Latency равная 14, также совпадает со вторым решением .

Использование ресурсов:

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	71	-
FIFO	-	-	-	-	-
Instance	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	42	-
Register	-	-	108	-	-
Total	0	0	108	113	0
Available	40	40	16000	8000	0
Utilization (%)	0	0	~0	1	0

Рисунок 3.5. Utilization estimates – summary

На рисунке 3.5. видно, что данный проект теперь займет на микросхеме 108 регистров для хранения чисел, и 113 LUT. Затрачиваемые ресурсы также совпадают со вторым решением.

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_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Рисунок 3.6. Interface Summary.

Полученные значения Interface Summary на рисунке 3.6. совпадают с прошлыми решениями, представленным на рисунке 1.7. и 2.6. На рисунке 3.6 также представлены интерфейсы, которые используются в синтезированном устройстве. Видно, что в схеме применяется те же протоколы ap_bus. Порты d_adress, d_datain, d_dataout и d_size 32-битные.

Performance Profile		Resource Profile			
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	14	-	15	-
Loop 1	no	12	3	-	4

Рисунок 3.7. Performance Profile

Performance Profile совпадает на рисунке 3.7 совпадает с предыдущим решением. Из рисунка видно, что задержка получения первого выходного значения составляет 3 такта с момента старта, для остальных -14, а задержка после старта до готовности приема новых данных – 15.

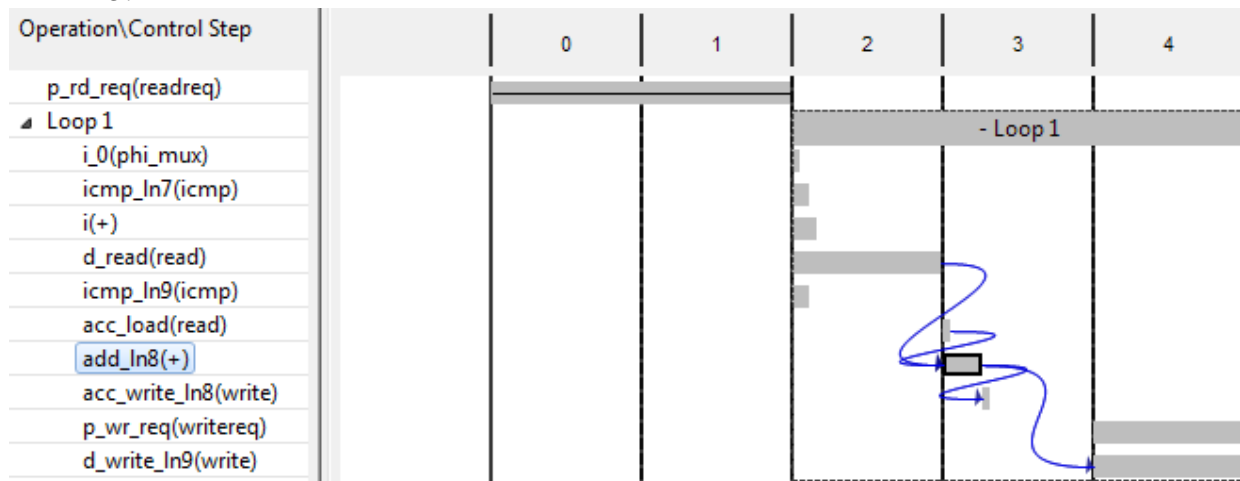


Рисунок 3.8. Schedule viewer

Schedule viewer также совпадает с предыдущим решением.

Величина latency также составила 14.

Профиль ресурсов также совпадает со вторым решением.

Performance Profile		Resource Profile								
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits*Banks
foo	0	0	108	113						
I/O Ports(1)					32					
Instances(0)	0	0	0	0						
Memories(0)	0		0	0	0			0	0	0
Expressions(5)	0	0	0	71	42	39	0			
Registers(7)			108		108					
acc			32		32					
add_ln8_reg			32		32					
ap_CS_fsm			5		5					
d_read_reg			32		32					
i_0_reg_54			3		3					
i_reg_101			3		3					
icmp_ln9_re			1		1					
Channels(0)	0		0	0	0			0	0	0
Multiplexers(2)	0		0	42	4			0		
DSP(0)		0								

Рисунок 3.9. Resource Profile

Здесь мы также не видим отличий от второго решения.

3.4. C|RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency, которые также соответствуют второму решению.

lab6_22main.c lab6_22test.c foo_csim.log

Cosimulation Report for 'foo'

Result

RTL	Status	Latency			Interval		
		min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	14	14	14	NA	NA	NA

Рисунок 3.10. Отчет о моделировании

Покажем временную диаграмму моделирования на рисунке 3.11.

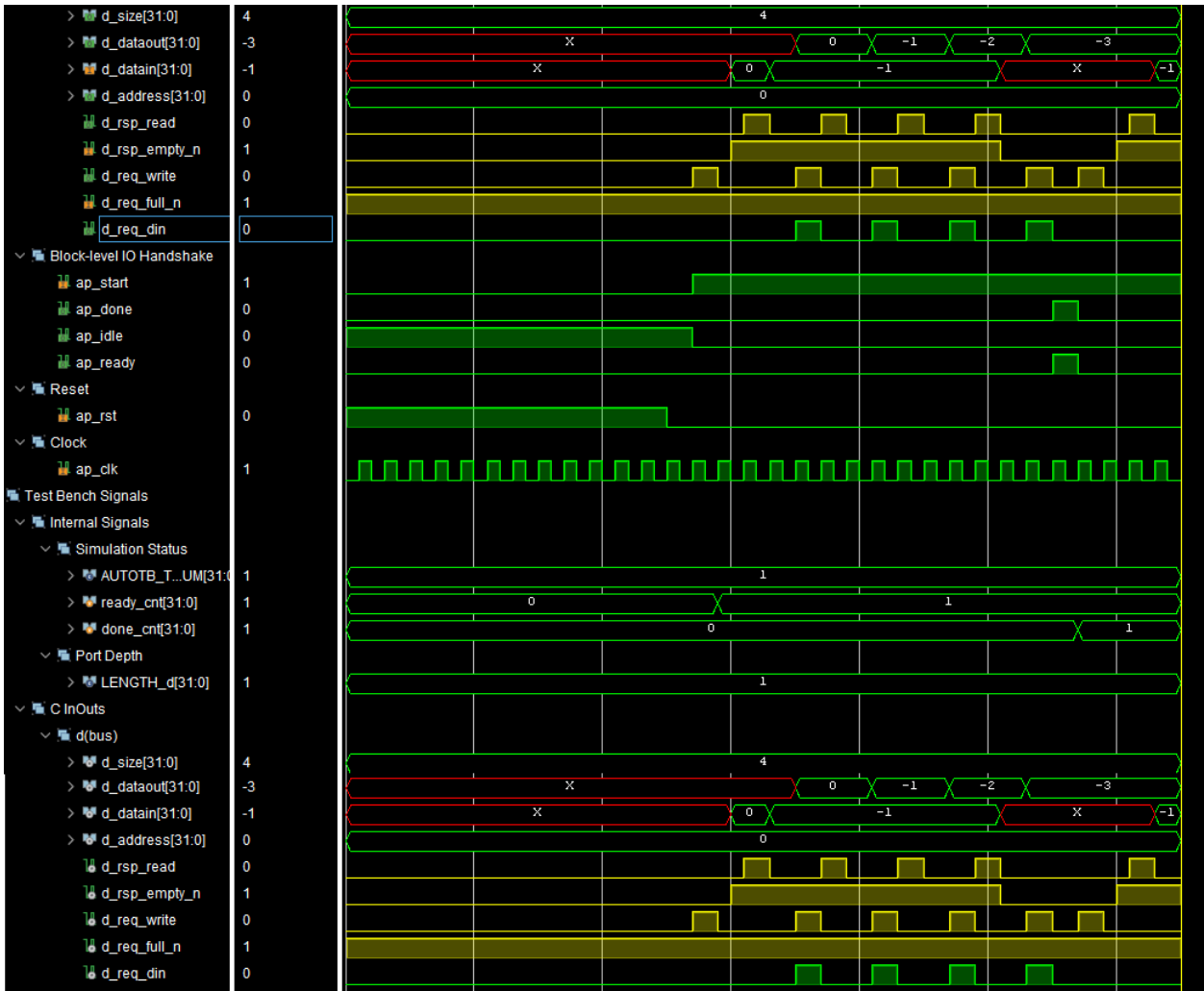


Рисунок 3.11. Design Top Signals

Здесь также не видны отличия во времени выполнения итераций от второго решения. Задержка составила также 14, отличие есть только с первым решением, в котором она была равна 4.

4. Четвертое решение

4.1. Настройки четвертого решения

Solution Name: solution1

Clock

Period: 10 Uncertainty: 0.1

Part Selection

Part: xa7a12tcsg325-1Q

☐ SDAccel Bottom Up Flow

Рисунок 4.0. Параметры второго решения

Программа, тест к ней и интерфейс (директива) представлены на рисунках 4.1 и 4.2.

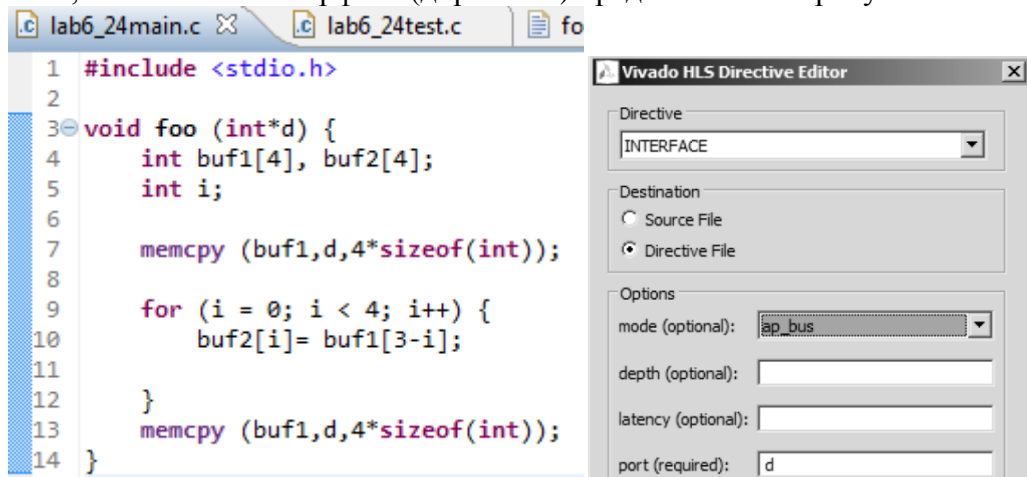


Рисунок 4.1. Исходная программа.

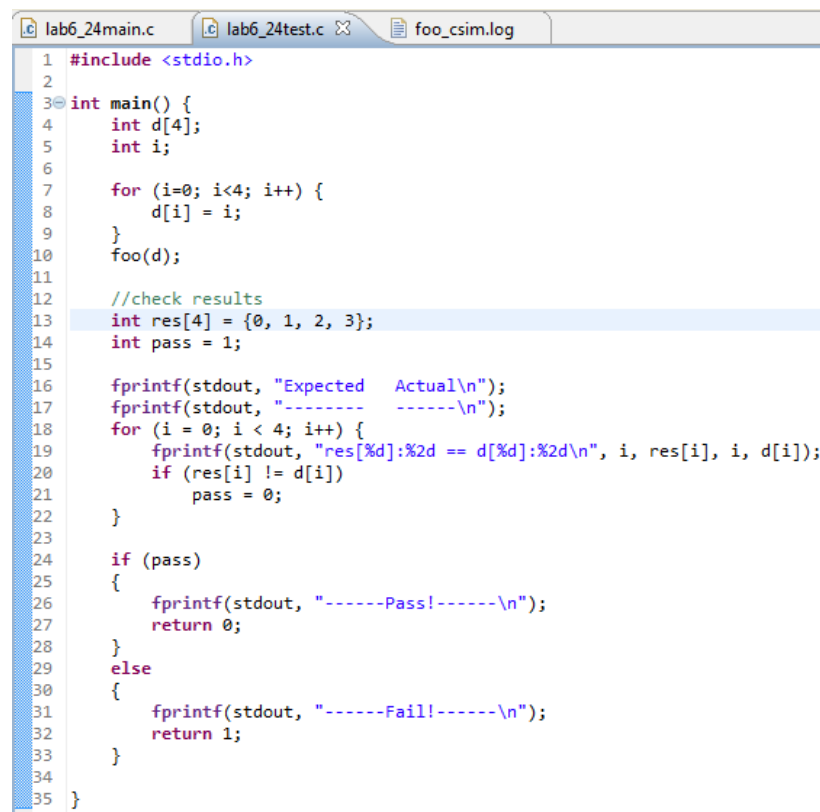
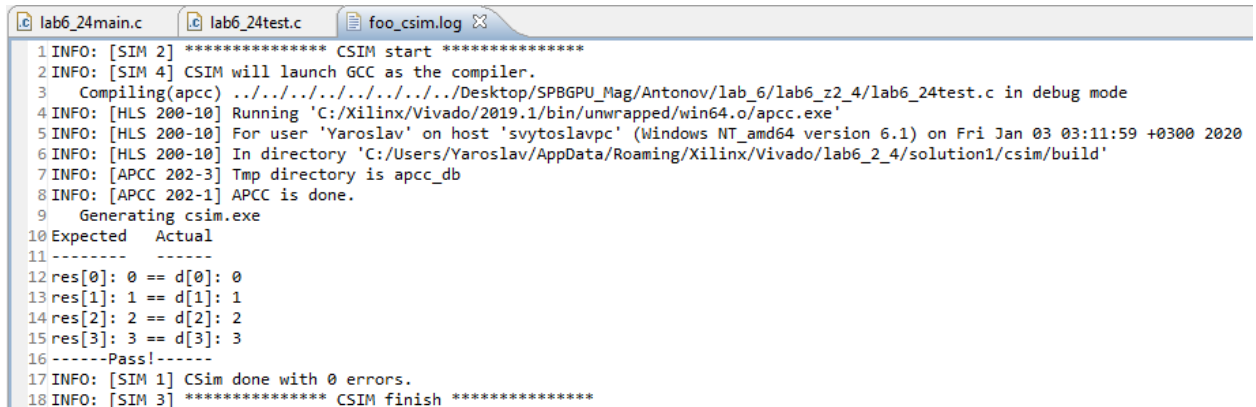


Рисунок 4.2. Тест к программе.

4.2. Моделирование

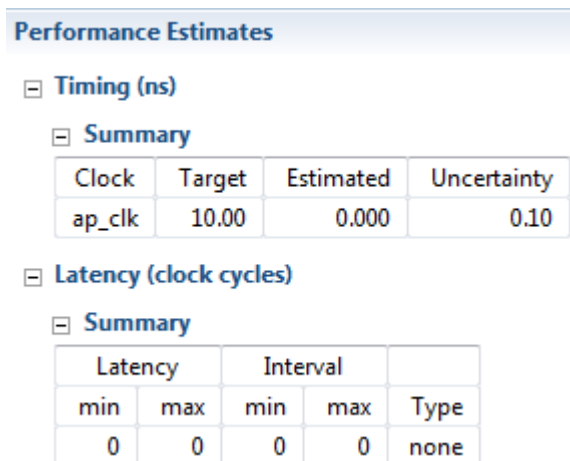


```
lab6_24main.c lab6_24test.c foo_csim.log
1 INFO: [SIM 2] ***** CSIM start *****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling(apcc) ../../../../../../Desktop/SPBGPU_Mag/Antonov/lab_6/lab6_z2_4/lab6_24test.c in debug mode
4 INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
5 INFO: [HLS 200-10] For user 'Yaroslav' on host 'svytoslavpc' (Windows NT_amd64 version 6.1) on Fri Jan 03 03:11:59 +0300 2020
6 INFO: [HLS 200-10] In directory 'C:/Users/Yaroslav/AppData/Roaming/Xilinx/Vivado/lab6_2_4/solution1/csim/build'
7 INFO: [APCC 202-3] Tmp directory is apcc_db
8 INFO: [APCC 202-1] APCC is done.
9   Generating csim.exe
10 Expected Actual
11 -----
12 res[0]: 0 == d[0]: 0
13 res[1]: 1 == d[1]: 1
14 res[2]: 2 == d[2]: 2
15 res[3]: 3 == d[3]: 3
16 -----Pass!-----
17 INFO: [SIM 1] CSim done with 0 errors.
18 INFO: [SIM 3] ***** CSIM finish *****
```

Рисунок 4.3 Результат успешного моделирования

Моделирование четвертого решения – операции копирования также прошло успешно.

4.3. Синтез



Performance Estimates				
Timing (ns)				
Summary				
Clock	Target	Estimated	Uncertainty	
ap_clk	10.00	0.000	0.10	
Latency (clock cycles)				
Summary				
Latency		Interval		
min	max	min	max	Type
0	0	0	0	none

Рисунок 4.4. Performance estimates – summary

На рисунке 4.4. можно увидеть, что достигнутая задержка $0.000 + 0.1$, и не совпадает ни с одним решением. Величина Latency составила 0, что также не совпадает ни с одним решением. Операция копирования прошла без задержек.

Использование ресурсов:

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

Рисунок 4.5. Utilization estimates – summary

На рисунке 4.5. видно, что данный проект теперь займет на микросхеме только 1 регистр для хранения чисел. Затрачиваемые ресурсы не совпадают ни с одним решением, так осуществляется процедура копирования.

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_req_din	out	1	ap_bus	d	pointer
d_req_full_n	in	1	ap_bus	d	pointer
d_req_write	out	1	ap_bus	d	pointer
d_rsp_empty_n	in	1	ap_bus	d	pointer
d_rsp_read	out	1	ap_bus	d	pointer
d_address	out	32	ap_bus	d	pointer
d_datain	in	32	ap_bus	d	pointer
d_dataout	out	32	ap_bus	d	pointer
d_size	out	32	ap_bus	d	pointer

Рисунок 4.6. Interface Summary.

Полученные значения Interface Summary на рисунке 4.6. совпадают с прошлыми решениями, представленным на рисунке 1.7, 2.6 и 3.6. На рисунке 4.6 также представлены интерфейсы, которые используются в синтезированном устройстве. Видно, что в схеме применяется те же протоколы ap_bus. Порты d_address, d_datain, d_dataout и d_size 32-битные.

Performance Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
foo	-	0	-	1	-

Рисунок 4.7. Performance Profile

Performance Profile на рисунке 4.7 говорит о том, что задержка при копировании отсутствует, а арифметических операций не происходит.

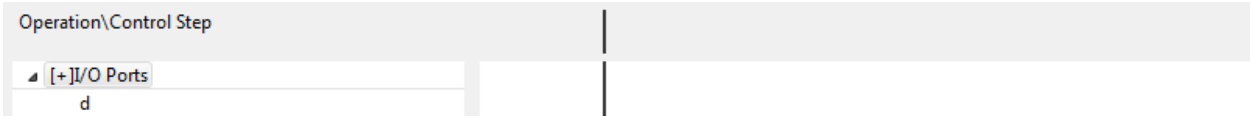


Рисунок 4.8. Schedule viewer

Schedule viewer также говорит о том, что процедура копирования происходит без задержек.

Профиль ресурсов также имеет ожидаемую структуру.

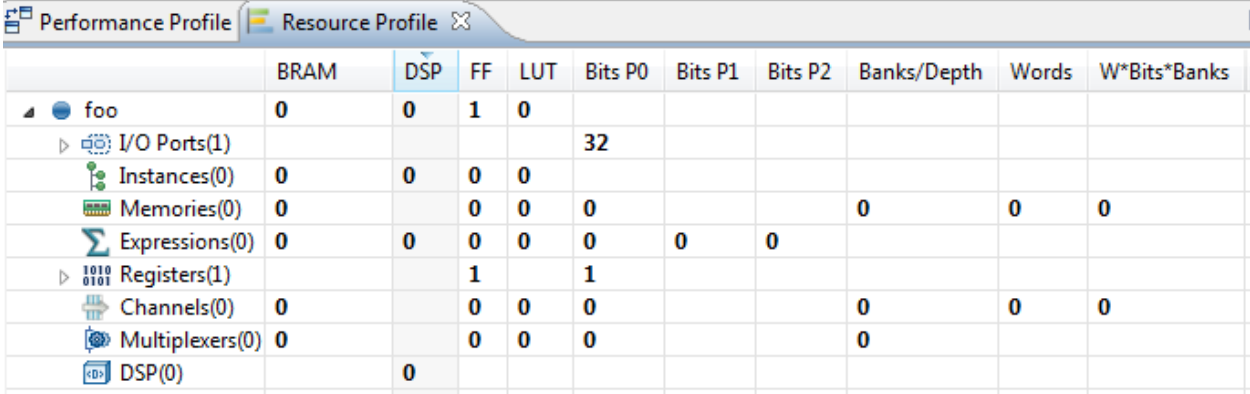


Рисунок 4.9. Resource Profile

4.4. C|RTL моделирование

При осуществлении совместного моделирования программа показала ожидаемые результаты Latency, которые также соответствуют второму решению.

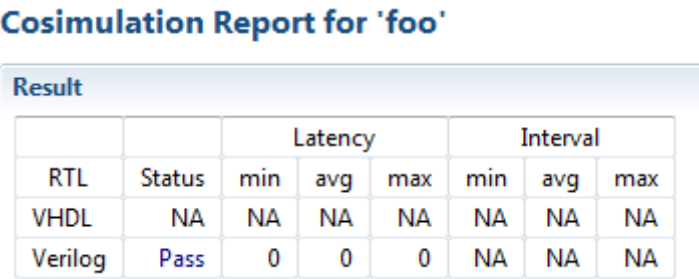


Рисунок 4.10. Отчет о моделировании

Покажем временную диаграмму моделирования на рисунке 4.11.

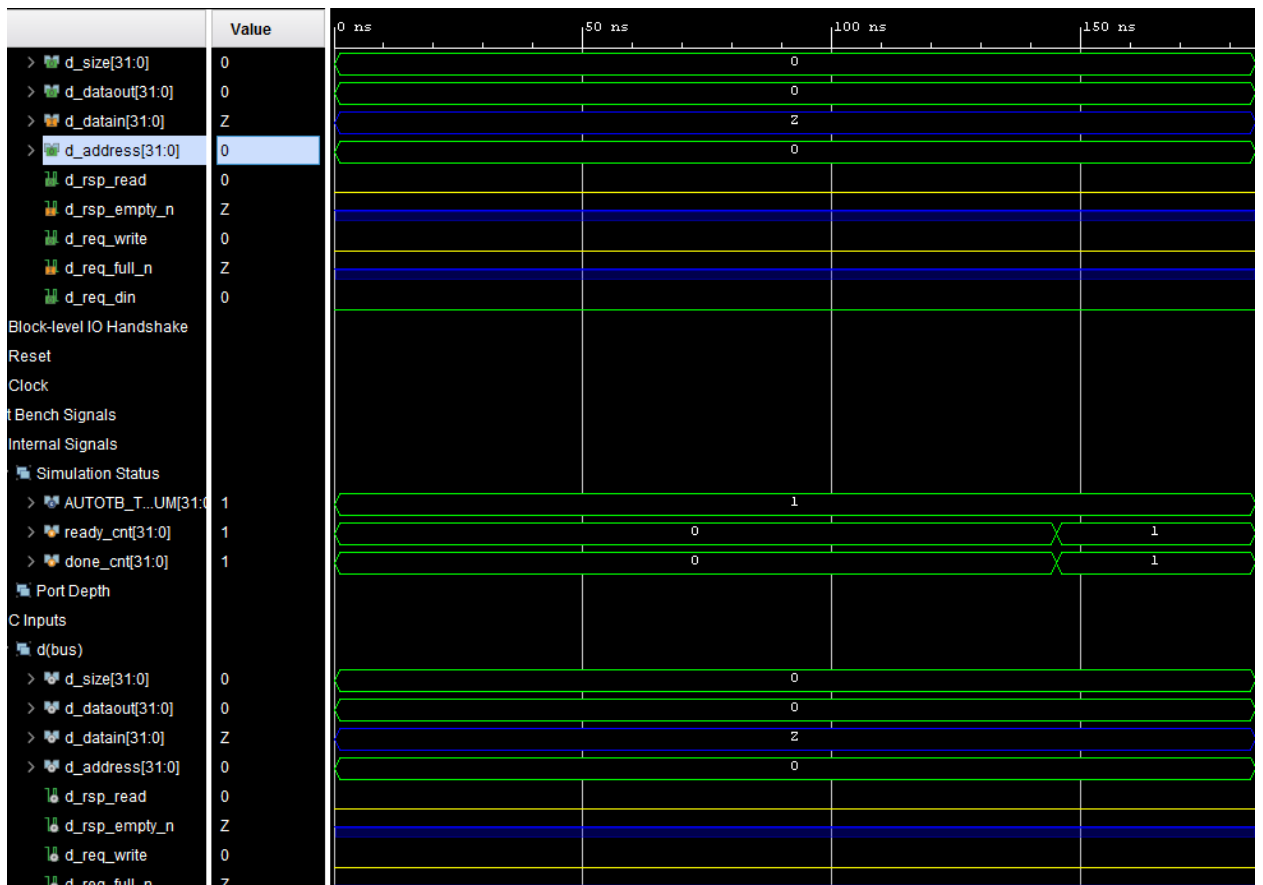


Рисунок 4.11. Design Top Signals

Здесь также видно, что операция копирования идет без задержек и иных арифметических операций не происходит.

4. Выводы

В данной работе были рассмотрены различия, которые могут появиться при синтезе устройства с применением протокола шины на уровне порта.

Инструмент Vivado HLS поддерживает протокол ввода-вывода шины:

- Протокол ввода-вывода - это протокол общей шины
- Протокол ввода / вывода не является отраслевым стандартом
- Протокол шины инструментов Vivado HLS позволяет подключаться к ядрам адаптера.

Протокол ввода / вывода шины поддерживает memсru

- Протокол ввода / вывода шины поддерживает функцию C memсru
- Обеспечивает высокопроизводительный интерфейс для пакетной передачи данных в стиле DMA

Протокол ввода / вывода шины поддерживает сложную арифметику указателей на вводе / выводе

- Указатели на ввод / вывод могут быть синтезированы в ar_fifo или ar_bus
- При использовании ar_fifo доступ должен быть последовательным
- Если используется арифметика указателя, порт должен использовать ar_bus

Протокол типа ar_bus реализует переменные указателя и передачи по ссылке в виде шины общего назначения.

В результате получены 4 решения: первое – одиночная операция чтения и записи массива в стандартном режиме. Величина latency составила 4 такта, а максимальная задержка обработки сигнала на такте составляет $9.900 + 0.1\text{нс}$; второе – множественная операция чтения и записи массива с аккумулярующим значением без дополнительных арифметических операций в стандартном режиме, полный цикл выполнения за 14 тактов, задержка составила также $9.900 + 0.1\text{нс}$; третье решение также - множественная операция чтения и записи массива с аккумулярующим значением с дополнительными арифметическими операциями в стандартном режиме, полный цикл выполнения также за 14 тактов, а максимальная задержка обработки сигнала на такте составила также $9.900 + 0.1\text{нс}$; четвертое решение - режим копирования, данный режим прошел без задержек и занял минимальное количество ресурсов.