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

Отчет по лабораторной работе №4_1
Курс: «Проектирование реконфигурируемых гибридных вычислительных систем»
Тема: «Port-level IO protocols»

Выполнил студент гр. 3540901/81501 Селиверстов С.А.
(подпись)

Руководитель Антонов А.П.
(подпись)
“___” _____2019 г.

Санкт – Петербург
2019

ОГЛАВЛЕНИЕ

Задание

1) Создать проект;
2) Подключить файлы исходного кода устройства и тест.

Листинг. Исходный код функции и теста

```
int lab4_1( int a, int b, int *c, int *d, int *p_y) {
    int y;
    y = a*b + (*c);
    *p_y = a*b + (*d);
    return y;
}
```

```
#include <stdio.h>

int main()
{
    int res;
    // For adders
    int refOut[3] = {230, 640, 1250};
     int pOut[3] = {240, 650, 1260};
    int pass;
    int i;

    int inA = 10;
    int inB = 20;
      int inC = 30;
    int inD = 40;
    int inP = 1;

    // Call the adder for 5 transactions
    for (i=0; i<3; i++)
    {
        res = lab4_1(inA, inB, &inC, &inD, &inP);

        fprintf(stdout, "  %d*%d+%d=%d \n", inA, inB, inC, res);
    fprintf(stdout, "  %d*%d+%d=%d \n", inA, inB, inD, inP);

      // Test the output against expected results
        if (res == refOut[i] & inP == pOut[i])
            pass = 1;
        else
            pass = 0;

        inA=inA+10;
        inB=inB+10;
        inC=inC+10;
        inD=inD+10;
    }

    if (pass)
    {
        fprintf(stdout, "----------Pass!-----------\n");
        return 0;
    }
    else
    {
        fprintf(stderr, "----------Fail!-----------\n");
        return 1;
    }
}
```

Задание:

- Создать проект lab4_1
- Подключить файл lab4_1.c (папка source)
- Создать тест lab4_1_test.c на основе теста Подключить тест lab1_1_test.c
- Микросхема: xa7a12tcsg325-1q
- Сделать solution1
    - задать: clock period 6; clock_uncertainty 0.1
    - осуществить моделирование
    - 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
        - Открыть временную диаграмму (все сигналы)
            - Отобразить два цикла обработки на одном экране
                - На скриншоте показать Latency
                - На скриншоте показать Initiation Interval
- Сделать solution2
    - Задать протоколы
        - a: ap_hs
        - b: ap_ask
        - *c: ap_hs
        - *d: ap_vld
        - *p_y: ap_ask
    - осуществить моделирование
    - 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
    - Открыть временную диаграмму (все сигналы)
        - Отобразить два цикла обработки на одном экране
            - На скриншоте показать Latency
            - На скриншоте показать Initiation Interval

- Выводы
    - Объяснить отличие протоколов port_level

Решение №1

При создании решения зададим настройки: clock period 6, clock uncertain 0.1, устройство *xa7a12tcsg325-1q*.

Моделирование

При запуске моделирования можно увидеть, что тест успешно пройден:

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling(apcc) ../../../../lab4_1_test.c in debug mode
INFO: [HLS 200-10] Running
'D:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'AlexKomarov' on host 'desktop-kk6t6ei' (Windows
NT_amd64 version 6.2) on Thu Oct 31 15:44:34 +0300 2019
INFO: [HLS 200-10] In directory
'D:/xilinx_projects/lab4/lab4/solution1/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
   Compiling(apcc) ../../../../lab4_1.c in debug mode
INFO: [HLS 200-10] Running
'D:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'AlexKomarov' on host 'desktop-kk6t6ei' (Windows
NT_amd64 version 6.2) on Thu Oct 31 15:44:43 +0300 2019
INFO: [HLS 200-10] In directory
'D:/xilinx_projects/lab4/lab4/solution1/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
   Generating csim.exe
  10*20+30=230
  10*20+40=240
  20*30+40=640
  20*30+50=650
  30*40+50=1250
  30*40+60=1260
----------Pass!------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Синтез

Приведем в отчете требуемые данные о проекте:

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	6.00	5.690	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
3	3	3	3	none

Рис. 1. Информация о производительности

Здесь можно увидеть, что достигнутая задержка равна 5.690 + 0.1, что укладывается в заданные нами требования к тактовой частоте.

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	78	-
FIFO	-	-	-	-	-
Instance	-	3	166	49	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	27	-
Register	-	-	36	-	-
Total	0	3	202	154	0
Available	40	40	16000	8000	0
Utilization (%)	0	7	1	1	0

Рис. 2. Занимаемые ресурсы

Данный проект займет на микросхеме 3 DSP блока, 202 регистра для хранения чисел, и 154 LUT.

Performance Profile | Resource Profile

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab4_1	-	3	-	4	-

Рис. 3. Профиль производительности

На этом рисунке видно, что задержка получения выходного значения составляет 3 такта с момента старта, а задержка после старта до готовности приема новых данных – 4. Покажем эти интервалы на временной диаграмме:

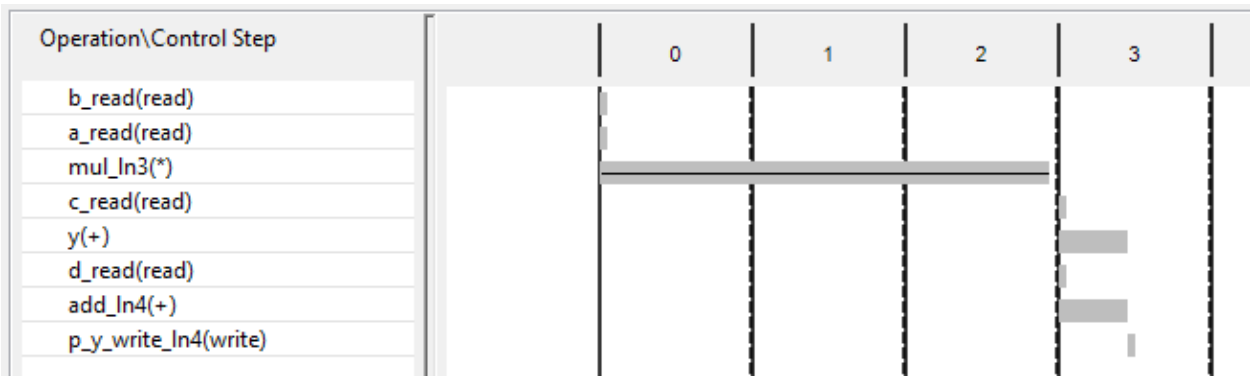


Рис. 4. Временная диаграмма

Здесь мы видим весь процесс получения результата. На первом такте происходит считывание операторов A и B, а также начинается их умножение. На четвертом такте начинается считывание C и D, а также их сложение. Таким образом суммарная задержка latency = 3, а со следующего 4-го такта можно подавать следующие данные (II = 4).

Таблица интерфейсов:

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	lab4_1	return value
ap_rst	in	1	ap_ctrl_hs	lab4_1	return value
ap_start	in	1	ap_ctrl_hs	lab4_1	return value
ap_done	out	1	ap_ctrl_hs	lab4_1	return value
ap_idle	out	1	ap_ctrl_hs	lab4_1	return value
ap_ready	out	1	ap_ctrl_hs	lab4_1	return value
ap_return	out	32	ap_ctrl_hs	lab4_1	return value
a	in	32	ap_none	a	scalar
b	in	32	ap_none	b	scalar
c	in	32	ap_none	c	pointer
d	in	32	ap_none	d	pointer
p_y	out	32	ap_vld	p_y	pointer
p_y_ap_vld	out	1	ap_vld	p_y	pointer

Рис. 5. Таблица интерфейсов

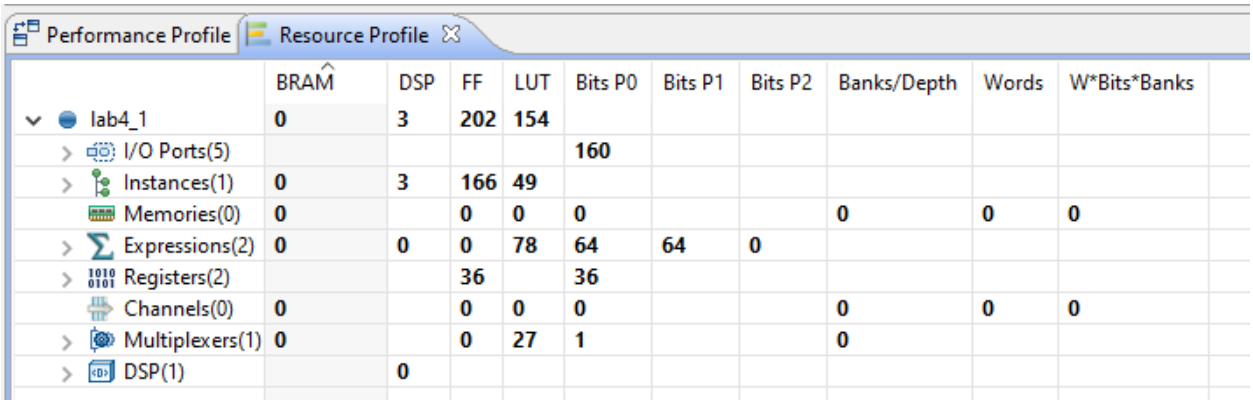
Для расчета схемы требуется более одного такта, поэтому в схему были добавлены ap_clk и ap_rst. Оба являются однобитовыми входами. Протокол управления вводом / выводом на уровне блоков был добавлен для управления RTL. Порты: ap_start, ap_done, ap_idle и ap_ready. Конструкция имеет 7 портов данных.

Входные порты: a, b, c, d являются 32-битными входами и имеют входы / выходы, протокол ap_none.

Конструкция имеет 32-битный выходной порт для возврата функции ap_return.

Также имеются порты: p_y – порт выхода для указателя и его однобитовый порт управления по протоколу wire handshake protocol.

Приведем профиль ресурсов:



Performance Profile | Resource Profile

	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits*Banks
lab4_1	0	3	202	154						
I/O Ports(5)					160					
Instances(1)	0	3	166	49						
Memories(0)	0		0	0	0			0	0	0
Expressions(2)	0	0	0	78	64	64	0			
Registers(2)			36		36					
Channels(0)	0		0	0	0			0	0	0
Multiplexers(1)	0		0	27	1			0		
DSP(1)		0								

Рис. 6. Профиль ресурсов

Здесь можно увидеть те же числа, что и в отчете синтезатора.

C/RTL моделирование

При совместном моделировании, программа отобразила те же самые, ожидаемые нами значения Latency и II:

Cosimulation Report for 'lab4_1'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	3	3	3	4	4	4

Export the report(.html) using the Export Wizard

Рис. 7. Результаты C/RTL моделирования

Покажем временную диаграмму совместного моделирования с отмеченными на ней Latency и II:

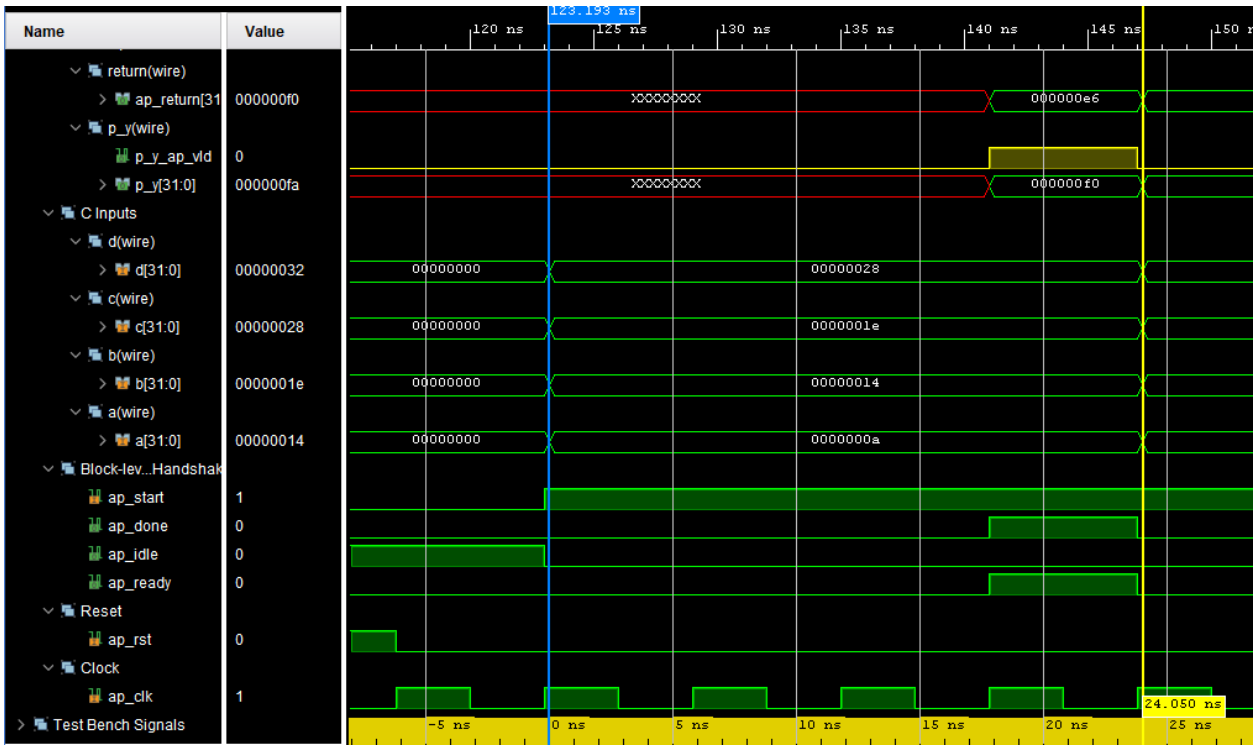


Рис. 8. Временная диаграмма совместного моделирования

Решение №2

Добавление протокола:

```
1  ############################################################
2  ## This file is generated automatically by Vivado HLS.
3  ## Please DO NOT edit it.
4  ## Copyright (C) 1986-2019 Xilinx, Inc. All Rights Reserved.
5  ############################################################
6  set_directive_interface -mode ap_hs "lab4_1" a
7  set_directive_interface -mode ap_ack "lab4_1" b
8  set_directive_interface -mode ap_hs "lab4_1" c
9  set_directive_interface -mode ap_vld "lab4_1" d
10 set_directive_interface -mode ap_ack "lab4_1" p_y
11
```

Рис. 9. Директивы

ap_hs, ap_ack и ap_vld – это wire handshake протоколы ввода/вывода.

ap_hs – протокол специализирующий валидацию значения на входе и потверждение значения на выходе.

ap_ack – задает подтверждение на входном/выходном порте.

ap_vld – задает валидацию на выходном/входном порте.

Значения считываются тогда, когда ap_vld имеет высокое значение, а когда считывание закончено – ap_ack принимает высокое значение.

Моделирование

Создадим второе решение для данного проекта. Его настройки аналогичны настройкам первого.

```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling(apcc) ../../../../lab4_1_test.c in debug mode
INFO: [HLS 200-10] Running
'D:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'AlexKomarov' on host 'desktop-kk6t6ei' (Windows
NT_amd64 version 6.2) on Thu Oct 31 16:33:05 +0300 2019
INFO: [HLS 200-10] In directory
'D:/xilinx_projects/lab4/lab4/solution2/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
   Compiling(apcc) ../../../../lab4_1.c in debug mode
INFO: [HLS 200-10] Running
'D:/Xilinx/Vivado/2019.1/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'AlexKomarov' on host 'desktop-kk6t6ei' (Windows
NT_amd64 version 6.2) on Thu Oct 31 16:33:10 +0300 2019
INFO: [HLS 200-10] In directory
'D:/xilinx_projects/lab4/lab4/solution2/csim/build'
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
   Generating csim.exe
  10*20+30=230
  10*20+40=240
```

```
  20*30+40=640
  20*30+50=650
  30*40+50=1250
  30*40+60=1260
----------Pass!------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
  30*40+50+60=1310
----------Pass!------------
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Моделирование второго решения также прошло успешно.

Синтез

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	6.00	5.690	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
3	3	3	3	none

Рис. 11. Производительность

Значения задержки соответствуют значениям из предыдущего решения.

Использование ресурсов:

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	80	-
FIFO	-	-	-	-	-
Instance	-	3	166	49	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	90	-
Register	-	-	101	-	-
Total	0	3	267	219	0
Available	40	40	16000	8000	0
Utilization (%)	0	7	1	2	0

Рис. 12. Затрачиваемые ресурсы

Данный проект будет занимать на микросхеме:

- 3 DSP блока

- 267 регистров для хранения и считывания данных (чисел)
- 219 LUT

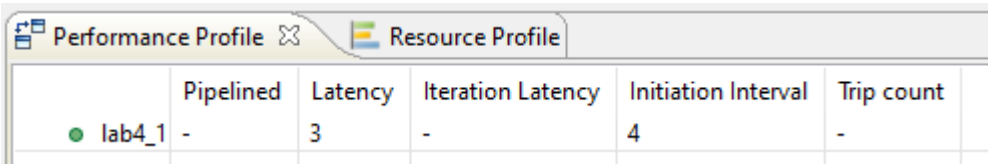


Рис. 13. Профиль производительности

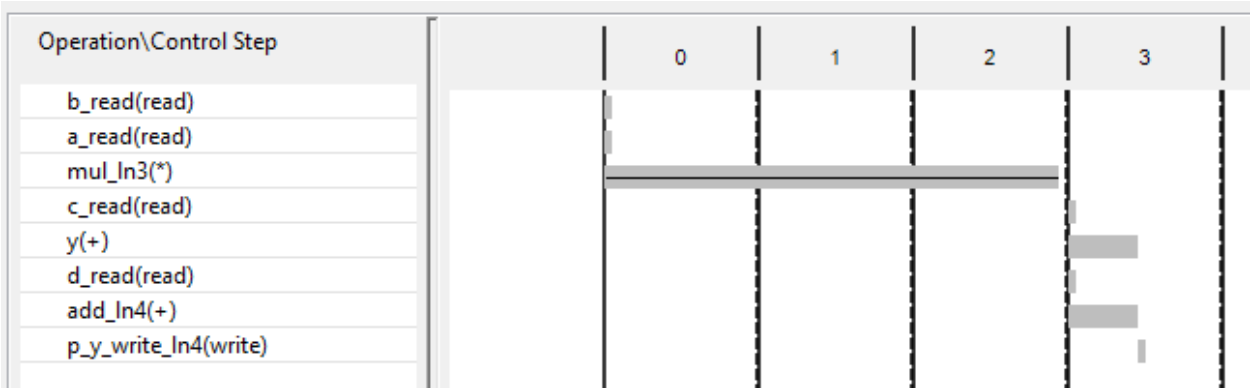


Рис. 14. Временная диаграмма

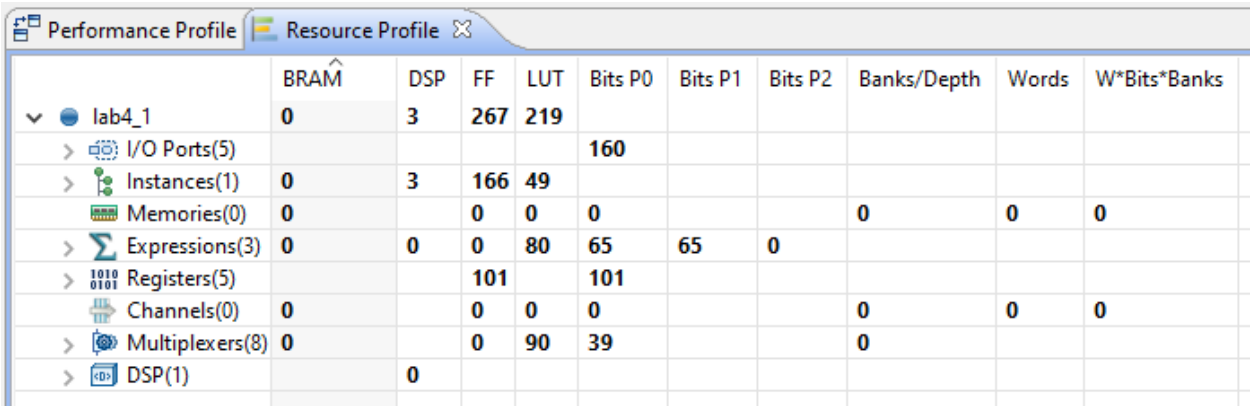


Рис. 15. Профиль ресурсов

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

Интерфейсы:

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	lab4_1	return value
ap_rst	in	1	ap_ctrl_hs	lab4_1	return value
ap_start	in	1	ap_ctrl_hs	lab4_1	return value
ap_done	out	1	ap_ctrl_hs	lab4_1	return value
ap_idle	out	1	ap_ctrl_hs	lab4_1	return value
ap_ready	out	1	ap_ctrl_hs	lab4_1	return value
ap_return	out	32	ap_ctrl_hs	lab4_1	return value
a	in	32	ap_hs	a	scalar
a_ap_vld	in	1	ap_hs	a	scalar
a_ap_ack	out	1	ap_hs	a	scalar
b	in	32	ap_ack	b	scalar
b_ap_ack	out	1	ap_ack	b	scalar
c	in	32	ap_hs	c	pointer
c_ap_vld	in	1	ap_hs	c	pointer
c_ap_ack	out	1	ap_hs	c	pointer
d	in	32	ap_vld	d	pointer
d_ap_vld	in	1	ap_vld	d	pointer
p_y	out	32	ap_ack	p_y	pointer
p_y_ap_ack	in	1	ap_ack	p_y	pointer

По сравнению с решением 1 появились следующие портовые протоколы ввода/вывода:

Помимо порта ввода для “a”, есть порты валидации для начала считывания и подтверждения по завершению считывания.

Помимо порта ввода для “b”, есть порт подтверждения по завершению считывания.

Помимо порта ввода для “c”, есть порты валидации для начала считывания и подтверждения по завершению считывания.

Помимо порта ввода для “d”, есть порт валидации для начала считывания.

Помимо порта вывода для “p_y”, есть порт приема подтверждения считывания.

C/RTL моделирование

При осуществлении совместного моделирования программа показала те же результаты для задержки.

Cosimulation Report for 'lab4_1'

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	3	3	3	4	4	4

Export the report(.html) using the Export Wizard

Рис. 16. C/RTL моделирование

Покажем временную диаграмму совместного моделирования:

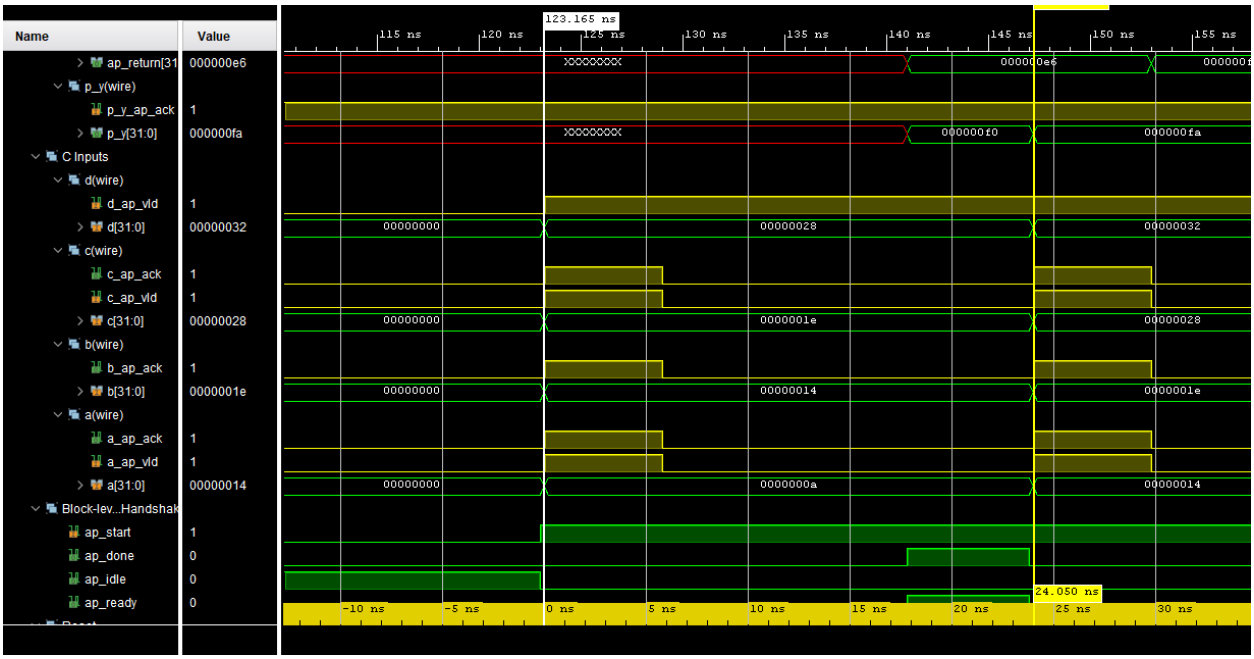


Рис. 17. Временная диаграмма совместного моделирования

Выводы

Существуют следующие типы port-level wire handshake протоколов: ap_vld, ap_ack и ap_hs. Они могут быть заданы портов ввода/вывода. Все порты для подтверждений валидаций являются однобитными. По умолчания для входов используется протокол ap_none, а для выходов – ap_vld.