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

Лабораторная №14

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Указатели

Задание 4

Студенты:
Соболь В.
Темнова А.С.
Группа: 13541/3
Преподаватель:
Антонов А.П.

Санкт-Петербург
2019

Содержание

1. Задание

1. Создать проект lab14_3
2. Микросхема: xa7a12tcsg325-1q
3. В папке source текст функции pointer_stream_good

   Познакомьтесь с ним (посмотрите в лекции часть Multi-Access Pointers)
4. Познакомьтесь с тестом.
5. Исследование:
6. Solution_1a

   - Создать версию pointer_stream_ , в которой будет убран volatile

```
#include "pointer_stream_good.h"

void pointer_stream_good ( volatile dout_t *d_o,  volatile din_t *d_i) {
    din_t acc = 0;

    acc += *d_i;
    acc += *(d_i+1);
    *d_o = acc;
    acc += *(d_i+2);
    acc += *(d_i+3);
    *(d_o+1) = acc;
}
```

7. Осуществить моделирование (при необходимости изменить тест) – обратить внимание на раздел тестирования в лекции
8. задать: clock period 10; clock_uncertainty 0.1
9. установить реализацию ПО УМОЛЧАНИЮ
10. 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
11. Выполнить cosimulation и привести временную диаграмму
12. Solution_2a

- Использовать исходную функцию pointer_stream_good
- Осуществить моделирование – обратить внимание на раздел тестирования в лекции
- задать: clock period 10; clock_uncertainty 0.1
- установить реализацию ПО УМОЛЧАНИЮ
- осуществить синтез
  - привести в отчете:
    * performance estimates=>summary (timing, latency)
    * utilization estimates=>summary
    * performance Profile
    * Resource profile
    * scheduler viewer (выполнить Zoom to Fit)
      · На скриншоте показать Latency
      · На скриншоте показать Initiation Interval
    * resource viewer (выполнить Zoom to Fit)
      · На скриншоте показать Latency
      · На скриншоте показать Initiation Interval
- Выполнить cosimulation и привести временную диаграмму

13. Сравнить два решения (solution_1a и solution_2a) и сделать выводы
14. Сравнить с решениями (solution_1a и solution_2a) для pointer_stream_better (предыдущее задание) и сделать выводы.
15.

2. Исходный код

Ниже приведен исходный код устройства и теста.

```
1 #include "pointer_stream_good.h"
2
3
4 #ifdef USE_VOLATILE
5
6 void pointer_stream_good ( volatile dout_t *d_o,  volatile din_t *d_i)
7
8 #else
9
10 void pointer_stream_good ( dout_t *d_o,  din_t *d_i)
11
12 #endif
13
14 {
15     din_t acc = 0;
16
17     acc += *d_i;
18     acc += *(d_i+1);
19     *d_o = acc;
20     acc += *(d_i+2);
21     acc += *(d_i+3);
22     *(d_o+1) = acc;
23 }
```

Рис. 2.1. Исходный код устройства

```
1 #ifndef _POINTER_STREAM_GOOD_H_
2 #define _POINTER_STREAM_GOOD_H_
3
4 #include <stdio.h>
5
6 typedef int din_t;
7 typedef int dout_t;
8
9 #ifdef USE_VOLATILE
10
11 void pointer_stream_good ( volatile dout_t *d_o,  volatile din_t *d_i);
12
13 #else
14
15 void pointer_stream_good ( dout_t *d_o,  din_t *d_i);
16
17 #endif
18
19 #endif
```

Рис. 2.2. Заголовочный файл

```
#include "pointer_stream_good.h"

int main () {
  din_t d_i[4];
  dout_t d_o[4];
  int i, retval=0;
  FILE        *fp;

  // Create input data
  for (i=0;i<4;i++) {
    d_i[i] = i;
  }

  // Call the function to operate on the data
  pointer_stream_good(d_o,d_i);

  // Save the results to a file
  fp=fopen("result.dat","w");
  fprintf(fp, "Din Dout\n");
  for (i=0;i<4;i++) {
    if (i<2)
      fprintf(fp, "%d   %d\n", d_i[i], d_o[i]);
    else
      fprintf(fp, "%d \n", d_i[i]);
  }
  fclose(fp);

  // Compare the results file with the golden results
  retval = system("diff --brief -w result.dat result.golden.dat");
  if (retval != 0) {
    printf("Test failed  !!!\n");
    retval=1;
  } else {
    printf("Test passed !\n");
  }

  // Return 0 if the test passed
  return retval;
}
```

Рис. 2.3. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```
open_project -reset lab14_4

add_files pointer_stream_good.c
add_files -tb pointer_stream_good_test.c
add_files -tb result.golden.dat

set_top pointer_stream_good


open_solution -reset solution_1a

set_part {xa7a12tcsg325-1q}
create_clock -period 10ns
set_clock_uncertainty 0.1

set_directive_interface -depth 4 -mode ap_fifo "pointer_stream_good" d_i
set_directive_interface -depth 2 -mode ap_fifo "pointer_stream_good" d_o

csim_design
csynth_design
cosim_design -trace_level all

add_files pointer_stream_good.c -cflags "-DUSE_VOLATILE"
open_solution -reset solution_2a

set_part {xa7a12tcsg325-1q}
create_clock -period 10ns
set_clock_uncertainty 0.1

set_directive_interface -depth 4 -mode ap_fifo "pointer_stream_good" d_i
set_directive_interface -depth 2 -mode ap_fifo "pointer_stream_good" d_o

csim_design
csynth_design
cosim_design -trace_level all
```

Рис. 3.1. Скрипт

4. Решение 1a

4.1. Моделирование

Ниже приведены результаты моделирования.

```
INFO: [HLS 200-10] In directory '/home/sobol/Downloads/labs_fro
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/18744615
INFO: [APCC 202-1] APCC is done.
   Generating csim.exe
Test passed !
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] *************** CSIM finish ***************
```

Рис. 4.1. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

4.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.413	0.10

Latency (clock cycles)

Summary

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

Рис. 4.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	119
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	66
Register	-	-	101	-
Total	0	0	101	185
Available	40	40	16000	8000
Utilization (%)	0	0	~0	2

Рис. 4.3. Utilization estimates

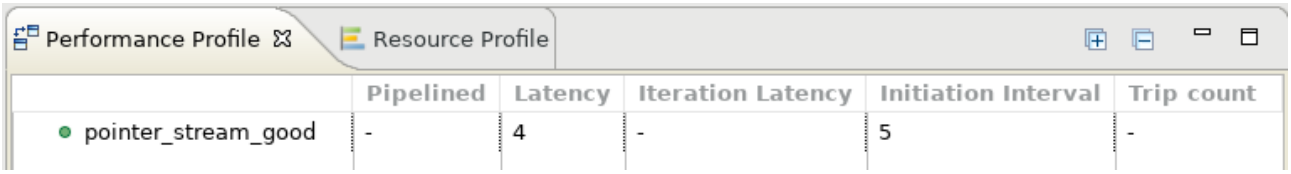

Рис. 4.4. Performance profile

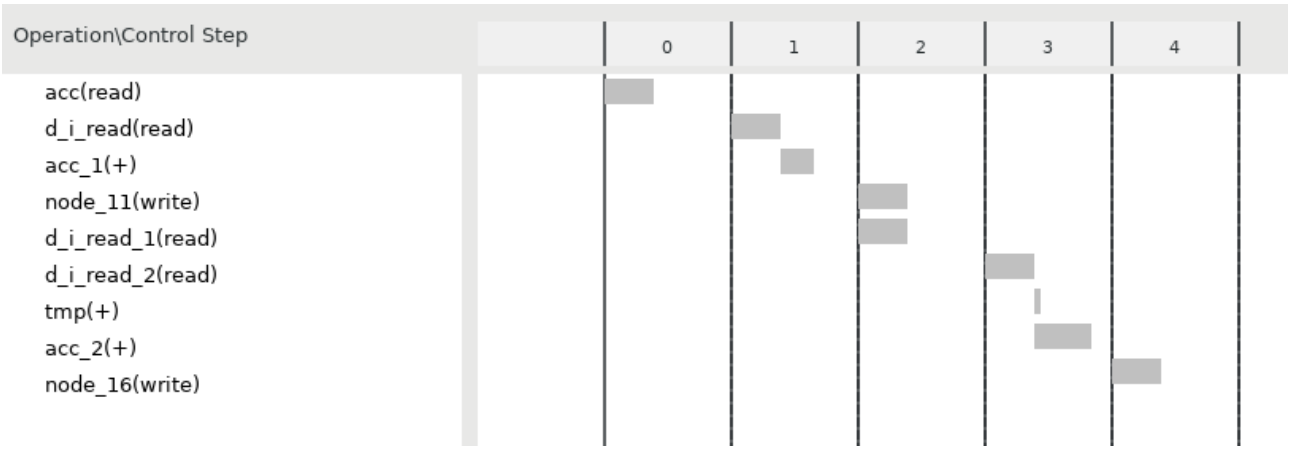

Рис. 4.5. Scheduler viewer

	Resource\Control S...	C0	C1	C2	C3	C4
1	⊟I/O Ports					
2	d_i	read	read	read	read	
3	d_o			write		write
4	⊟Expressions					
5	acc_1_fu_46		+			
6	acc_2_fu_57				+	
7	tmp_fu_52				+	

Рис. 4.6. Resource viewer

4.3. C/RTL моделирование

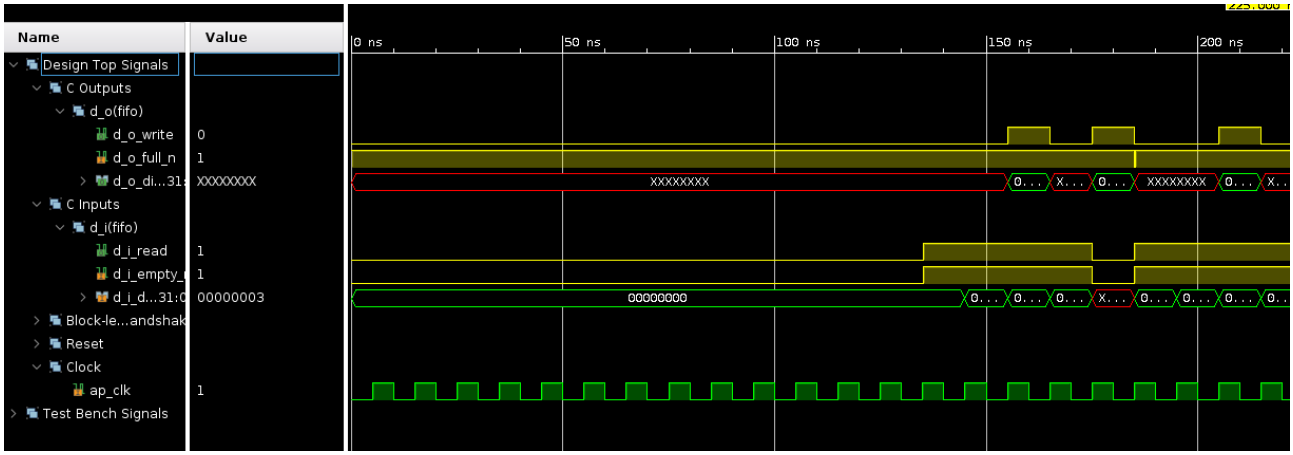

Рис. 4.7. Временная диаграмма

В данном решении, latency составляет 4 такта, а II равен 5 тактов.

5. Решение 2a

5.1. Моделирование

Ниже приведены результаты моделирования.

```
INFO: [HLS 200-10] In directory '/home/sobol/Downloads/labs_fr
INFO: [APCC 202-3] Tmp directory is /tmp/apcc_db_sobol/1880421
INFO: [APCC 202-1] APCC is done.
   Generating csim.exe
Test passed !
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] *************** CSIM finish ***************
```

Рис. 5.1. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

5.2. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.413	0.10

Latency (clock cycles)

Summary

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

Рис. 5.2. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	119
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	66
Register	-	-	101	-
Total	0	0	101	185
Available	40	40	16000	8000
Utilization (%)	0	0	~0	2

Рис. 5.3. Utilization estimates

Performance Profile | Resource Profile

	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip co
pointer_stream_good	-	4	-	5	-

Рис. 5.4. Performance profile

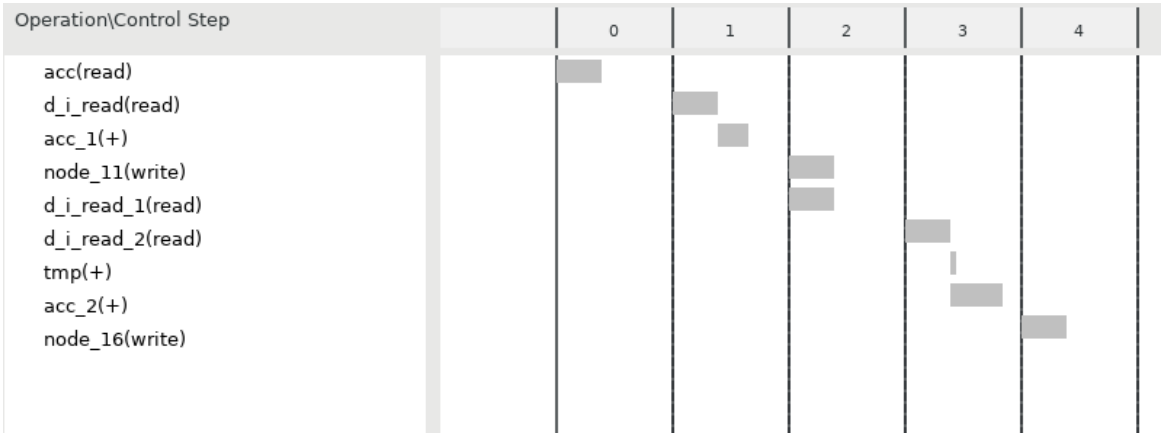

Рис. 5.5. Scheduler viewer

	Resource\Control S...	C0	C1	C2	C3	C4
1	I/O Ports					
2	d_i	read	read	read	read	
3	d_o			write		write
4	Expressions					
5	acc_1_fu_43		+			
6	tmp_fu_49				+	
7	acc_2_fu_54				+	

Рис. 5.6. Resource viewer

5.3. C/RTL моделирование

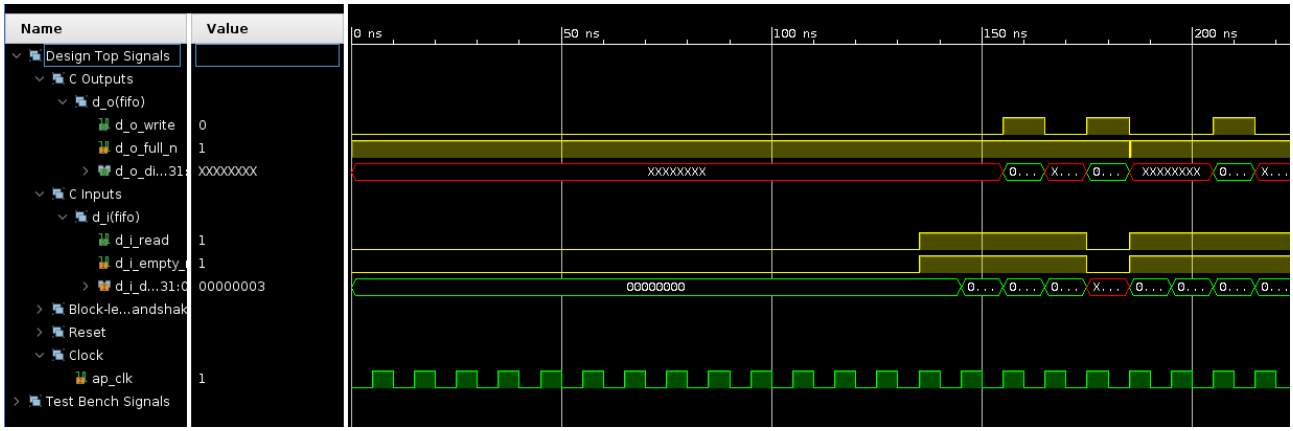

Рис. 5.7. Временная диаграмма

Результаты в данном решении полностью совпадают с предыдущим.

6. Вывод

В данной лабораторной работе использования ключевого слова volatile не влияет на результат.