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 6

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

Санкт-Петербург
2019

Содержание

1. Задание

1. Создать проект lab14_6
2. Микросхема: xa7a12tcsg325-1q
3. В папке source текст функции pointer_multi

   Познакомьтесь с ним.
4. Познакомьтесь с тестом.
5. Исследование:
6. Solution_1a

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

     Обратить внимание на реализацию интерфейсов.
7. сделать выводы по работе с указателями.
8. Solution_2a – предложить другой вариант функции без использования указателей + создать тест

   - Осуществить моделирование
   - 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

9. Сравнить два решения (solution_1a и solution_2a) и сделать выводы

2. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```
1  open_project -reset lab14_6_ptr
2
3  add_files pointer_multi.c
4  add_files -tb pointer_multi_test.c
5  add_files -tb result.golden.dat
6  set_top pointer_multi
7
8  open_solution -reset solution_1a
9
10 set_part {xa7a12tcsg325-1q}
11 create_clock -period 10ns
12 set_clock_uncertainty 0.1
13
14 csim_design
15 csynth_design
16 cosim_design -trace_level all
17
18 open_project -reset lab14_6_no_ptr
19
20 add_files no_ptr.c
21 add_files -tb pointer_multi_test.c
22 add_files -tb result.golden.dat
23 set_top pointer_multi
24
25 open_solution -reset solution_2a
26
27 set_part {xa7a12tcsg325-1q}
28 create_clock -period 10ns
29 set_clock_uncertainty 0.1
30
31 csim_design
32 csynth_design
33 cosim_design -trace_level all
34
35 exit
```

Рис. 2.1. Скрипт

3. Решение 1а

3.1. Исходный код

Ниже приведен исходный код устройства и теста.

```
1 #include "pointer_multi.h"
2
3 dout_t pointer_multi (sel_t sel, din_t pos) {
4   static const dout_t a[8] = {1, 2, 3, 4, 5, 6, 7, 8};
5   static const dout_t b[8] = {8, 7, 6, 5, 4, 3, 2, 1};
6
7   dout_t* ptr;
8   if (sel)
9     ptr = a;
10  else
11    ptr = b;
12
13  return ptr[pos];
14 }
```

Рис. 3.1. Исходный код устройства

```
1 #ifndef _POINTER_MULTI_H_
2 #define _POINTER_MULTI_H_
3
4 #include <stdio.h>
5 #include <stdbool.h>
6
7 typedef unsigned char din_t;
8 typedef int dout_t;
9 typedef bool sel_t;
10
11 dout_t pointer_multi (bool sel, din_t pos);
12
13 #endif
```

Рис. 3.2. Заголовочный файл

```
#include "pointer_multi.h"

int main () {
  din_t idx=0;
  sel_t mem_sel=true;
  dout_t dout;

  int i, retval=0;
  FILE        *fp;

  // Save the results to a file
  fp=fopen("result.dat","w");

  // Call the function
  // Create Input Data
  for(i=0; i<8;++i) {
    dout=pointer_multi ( mem_sel, idx);
    fprintf(fp, "%d \n", dout);
    idx=idx+1;
    mem_sel=!mem_sel;
   }

  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. 3.3. Исходный код теста

3.2. Моделирование

Ниже приведены результаты моделирования.

```
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. 3.4. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

3.3. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	4.627	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
1	1	1	1	none

Рис. 3.5. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	4
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	0	-	8	2
Multiplexer	-	-	-	15
Register	-	-	2	-
Total	0	0	10	21
Available	40	40	16000	8000
Utilization (%)	0	0	~0	~0

Рис. 3.6. Utilization estimates

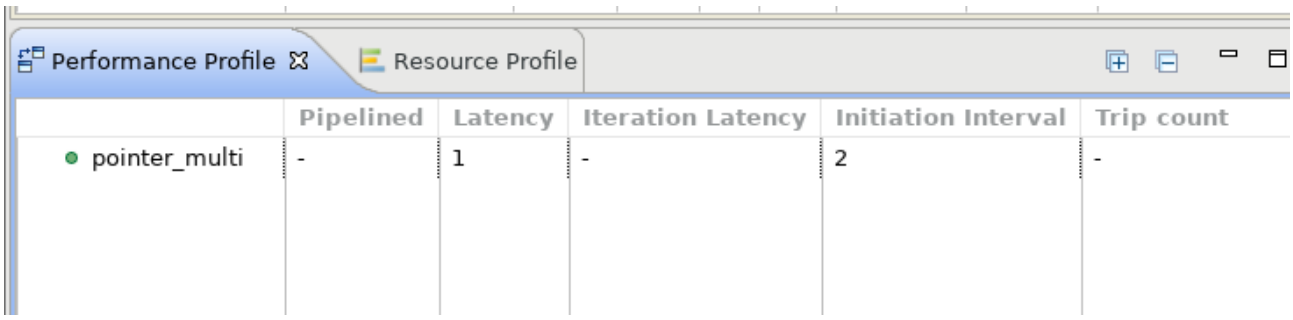

Рис. 3.7. Performance profile

Рис. 3.8. Scheduler viewer

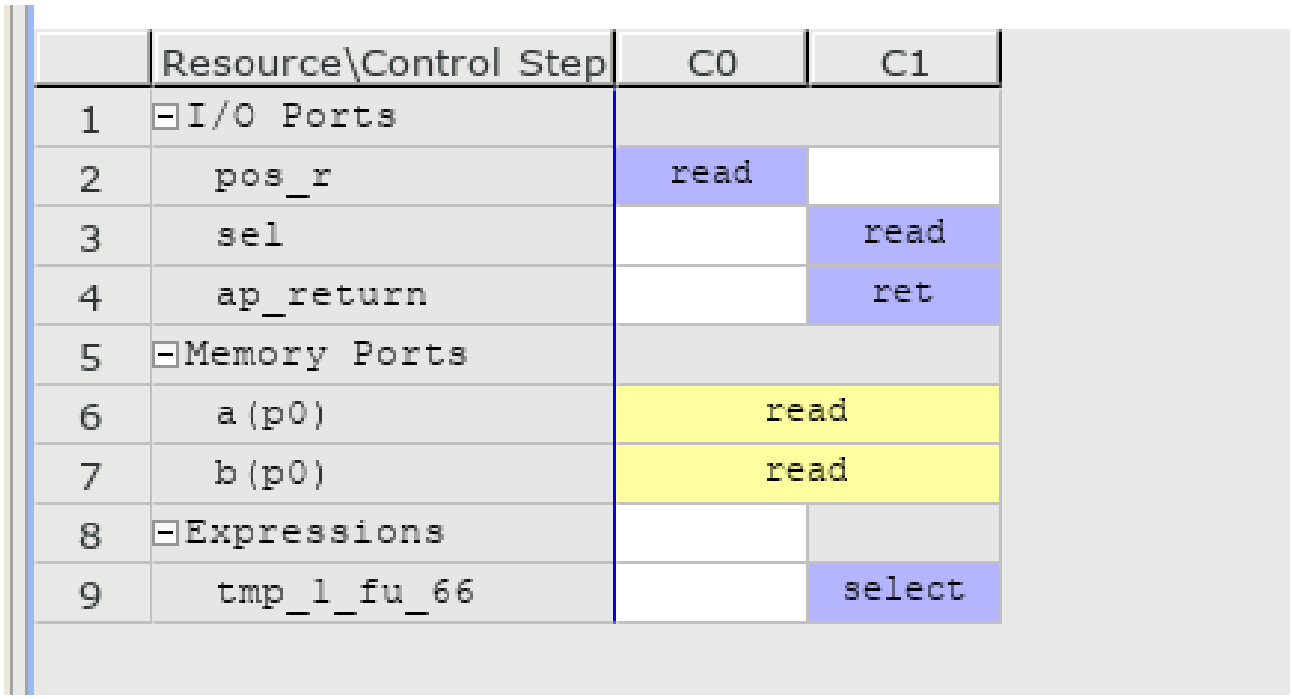

Рис. 3.9. Resource viewer

3.4. C/RTL моделирование

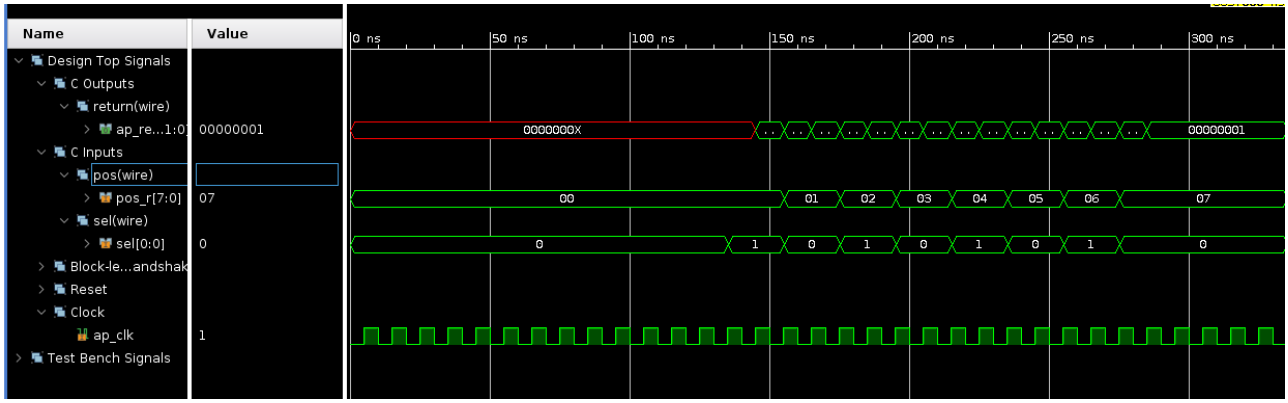

Рис. 3.10. Временная диаграмма

Как видно по результатам, задержка в данном решении составляет 1 такт.

4. Решение 2a

4.1. Исходный код

Ниже приведен исходный код устройства.

```
1 #include "pointer_multi.h"
2
3 dout_t pointer_multi (sel_t sel, din_t pos) {
4   if (sel){
5     return  1 + pos;
6   } else {
7     return 8 - pos;
8   }
9 }
```

Рис. 4.1. Исходный код устройства

Исходный код заголовочного файла и теста соответствуют предыдущему решению.

4.2. Моделирование

Ниже приведены результаты моделирования.

```
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. 4.2. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

4.3. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	3.485	0.10

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
0	0	0	0	none

Рис. 4.3. Performance estimates

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	42
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	-	-
Total	0	0	0	42
Available	40	40	16000	8000
Utilization (%)	0	0	0	~0

Detail

Рис. 4.4. Utilization estimates

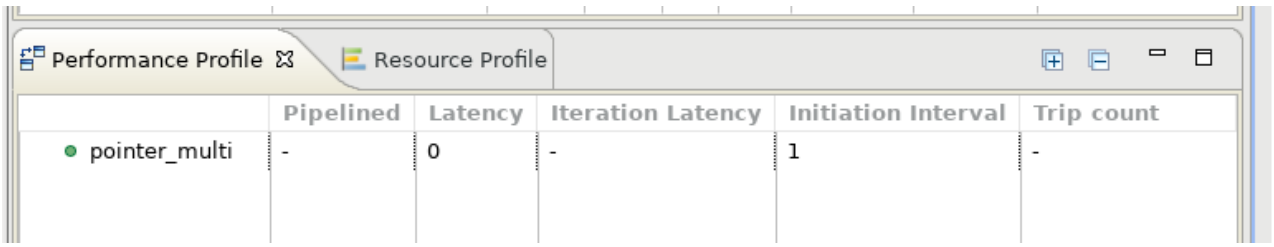

Рис. 4.5. Performance profile

Рис. 4.6. Scheduler viewer

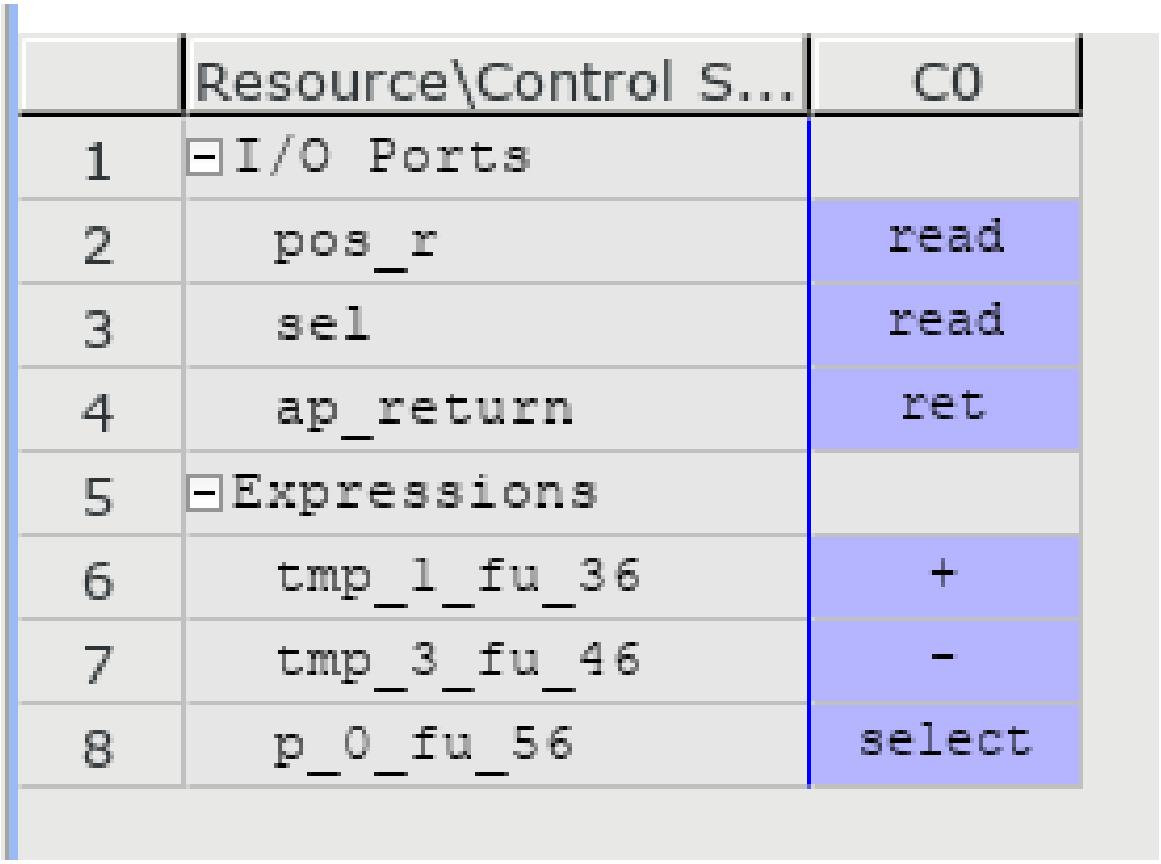

Рис. 4.7. Resource viewer

4.4. C/RTL моделирование

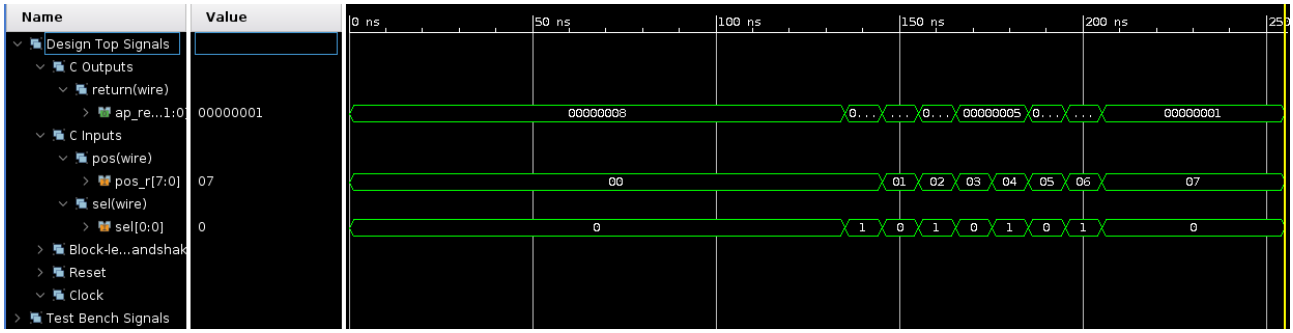

Рис. 4.8. Временная диаграмма

Как видно по результатам, задержка в данном решении отсутствует. Также, в сравнении с предыдущим решением, значительно снизились затраты ресурсов.

5. Вывод

В ходе данной лабораторной работы было выяснено, что по возможности следует избегать работы с массивами и указателями, так как это влечёт за собой дополнительные расходы.