

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

Лабораторная №6

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Сравнение протоколов Port-Level I/O

Задание 1

Студенты:

Соболь В.

Темнова А.С.

Группа: 13541/3

Преподаватель:

Антонов А.П.

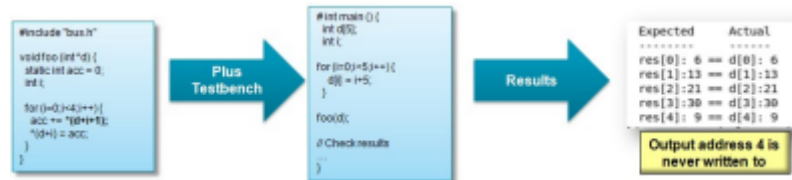
Санкт-Петербург
2019

Содержание

| | |
|------------------------------------|-----------|
| 1. Задание | 3 |
| 2. Исходный код | 4 |
| 3. Скрипт | 5 |
| 4. Решение 1 | 6 |
| 4.1. Директивы | 6 |
| 4.2. Моделирование | 6 |
| 4.3. Синтез | 7 |
| 5. Решение 2 | 7 |
| 5.1. Директивы | 7 |
| 5.2. Моделирование | 7 |
| 5.3. Синтез | 8 |
| 5.4. C/RTL моделирование | 11 |
| 6. Вывод | 11 |

1. Задание

1. Создать проект lab6_1
2. Микросхема: xa7a12tcsg325-1q
3. Создать Си код на основе слайда (функция foo)



4. Создать тест lab6_1_test.c на основе слайда выше.
5. Сделать solution1
 - задать: clock period 6; clock_uncertainty 0.1
 - осуществить моделирование (на основе слайда выше, с выводом результатов в консоль)
 - осуществить синтез (с настройками по умолчанию – интерфейс ap-fifo)
 - 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
6. Сделать solution2
 - Задать протокол
 - a: ap_bus

- 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
 - 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

7. Выводы

- Объяснить отличие протоколов

2. Исходный код

Ниже приведен исходный код устройства и теста, использованных в лабораторной работе.

```

1 void lab6_1(int* d){
2   static int acc = 0;
3   int i;
4   for(i = 0; i < 4; i++) {
5     acc += *(d + i + 1);
6     *(d + i) = acc;
7   }
8 }

```

Рис. 2.1. Исходный код устройства

```

1 #include <stdio.h>
2
3 int main() {
4     int pass = 1;
5     int i;
6     int d[5];
7     int expected[5] = {6, 13, 21, 30, 9};
8
9     for (i = 0; i < 5; i++) {
10         d[i] = i + 5;
11     }
12
13     lab6_1(d);
14
15     for (i = 0; i < 5; i++) {
16         fprintf(stdout, "%d:_Expeced_%d_Actual_%d\n", i, expected[i], d[i]);
17         if(expected[i] != d[i]) {
18             pass = 0;
19         }
20     }
21
22
23     if (pass)
24     {
25         fprintf(stdout, "—————Pass!—————\n");
26         return 0;
27     }
28     else
29     {
30         fprintf(stderr, "—————Fail!—————\n");
31         return 1;
32     }
33 }

```

Рис. 2.2. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```

1 open_project -reset lab6_1
2 add_files lab6_1.c
3 set_top lab6_1
4 add_files -tb lab6_1_test.c
5
6 open_solution solution1 -reset
7 set_part {xa7a12tcsg325-1q}
8 create_clock -period 6ns
9 set_clock_uncertainty 0.1
10
11 csim_design
12 # csynth_design
13 # cosim_design -trace_level all
14
15
16 open_solution solution2 -reset
17 set_part {xa7a12tcsg325-1q}
18 create_clock -period 6ns
19 set_clock_uncertainty 0.1
20
21 set_directive_interface -mode ap_bus lab6_1 d
22
23 csim_design
24 csynth_design
25 cosim_design -trace_level all
26
27
28 exit

```

Рис. 3.1. Скрипт

4. Решение 1

4.1. Директивы

В данном решения были установлены директивы, приведённые ниже.

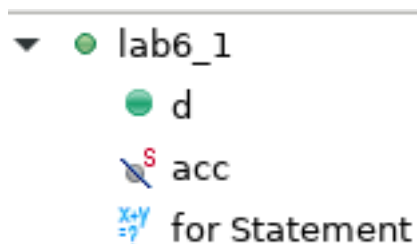


Рис. 4.1. Директивы

4.2. Моделирование

Ниже приведены результаты моделирования.

```

INFO: [APCC 202-1] APCC is done.
    Generating csim.exe
0: Expeced 6 Actual 6
1: Expeced 13 Actual 13
2: Expeced 21 Actual 21
3: Expeced 30 Actual 30
4: Expeced 9 Actual 9
-----Pass!-----
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

При синтезе возникает ошибка, приведённая ниже.

```

ERROR: [SYNCHK 200-61] lab6_1.c:5: unsupported memory access on variable 'd' which is (or contains) an array with unknown size at compile time.
INFO: [SYNCHK 200-10] 1 error(s), 0 warning(s).
ERROR: [HLS 200-70] Synthesizability check failed.

```

Рис. 4.3. Ошибка синтеза

5. Решение 2

5.1. Директивы

В данном решении были установлены директивы, приведённые ниже.

```

▼ ● lab6_1
    ● d
    % HLS INTERFACE ap_bus port=d
    S acc
    X+Y=? for Statement

```

Рис. 5.1. Директивы

5.2. Моделирование

Ниже приведены результаты моделирования.

```

INFO: [APCC 202-1] APCC is done.
    Generating csim.exe
0: Expeced 6 Actual 6
1: Expeced 13 Actual 13
2: Expeced 21 Actual 21
3: Expeced 30 Actual 30
4: Expeced 9 Actual 9
-----Pass!-----
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****

```

Рис. 5.2. Результаты моделирования

По результатам моделирования видно, что устройство работает корректно.

5.3. Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

Performance Estimates

▣ Timing (ns)

▣ Summary

| Clock | Target | Estimated | Uncertainty |
|--------|--------|-----------|-------------|
| ap_clk | 6.00 | 5.900 | 0.10 |

▣ Latency (clock cycles)

▣ Summary

| Latency | | Interval | | Type |
|---------|-----|----------|-----|------|
| min | max | min | max | |
| 25 | 25 | 25 | 25 | none |

Рис. 5.3. Performance estimates

Utilization Estimates

Summary

| Name | BRAM_18K | DSP48E | FF | LUT |
|------------------------|-----------|-----------|--------------|-------------|
| DSP | - | - | - | - |
| Expression | - | - | 0 | 60 |
| FIFO | - | - | - | - |
| Instance | - | - | - | - |
| Memory | - | - | - | - |
| Multiplexer | - | - | - | 65 |
| Register | - | - | 109 | - |
| Total | 0 | 0 | 109 | 125 |
| Available | 40 | 40 | 16000 | 8000 |
| Utilization (%) | 0 | 0 | ~0 | 1 |

Рис. 5.4. Utilization estimates

| Performance Profile | | Resource Profile | | | |
|---------------------|-----------|------------------|-------------------|---------------------|------------|
| | Pipelined | Latency | Iteration Latency | Initiation Interval | Trip count |
| lab6_1 | - | 25 | - | 26 | - |
| Loop 1 | no | 24 | 6 | - | 4 |

Рис. 5.5. Performance profile

Interface

Summary

| RTL Ports | Dir | Bits | Protocol | Source Object | C Type |
|---------------|-----|------|------------|---------------|--------------|
| ap_clk | in | 1 | ap_ctrl_hs | lab6_1 | return value |
| ap_rst | in | 1 | ap_ctrl_hs | lab6_1 | return value |
| ap_start | in | 1 | ap_ctrl_hs | lab6_1 | return value |
| ap_done | out | 1 | ap_ctrl_hs | lab6_1 | return value |
| ap_idle | out | 1 | ap_ctrl_hs | lab6_1 | return value |
| ap_ready | out | 1 | ap_ctrl_hs | lab6_1 | 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 |

Рис. 5.6. Interface estimates

По списку сигналов в проекте видно, что для заданного порта установлен протокол ap_bus. Также видно, что для этого протокола требуются дополнительные сигналы.

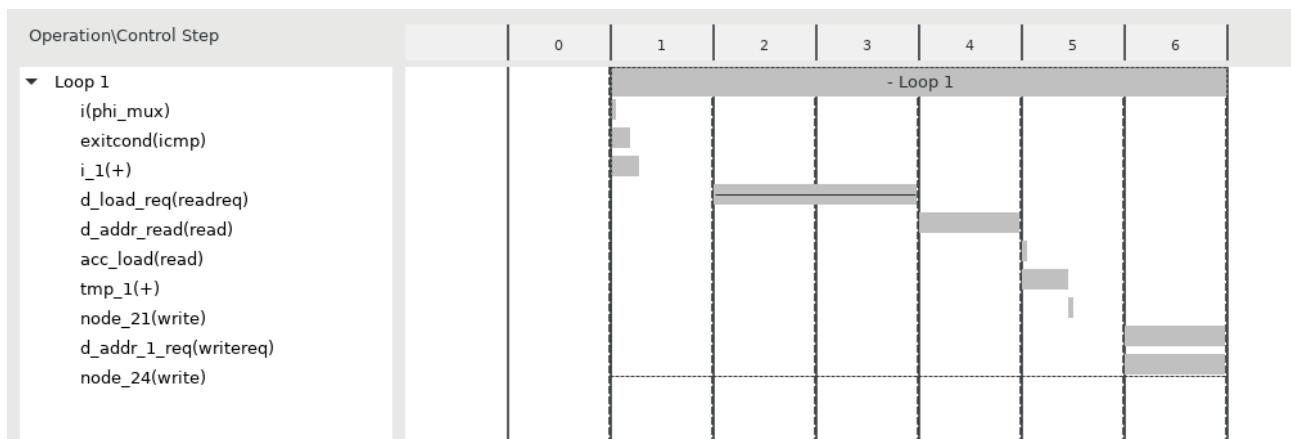


Рис. 5.7. Scheduler viewer

| | Resource\Control Step | C0 | C1 | C2 | C3 | C4 | C5 | C6 |
|---|-----------------------|----|---------|----|----|----|----|----|
| 1 | I/O Ports | | | | | | | |
| 2 | d | | | | | | | |
| 3 | Expressions | | | | | | | |
| 4 | i_l_fu_76 | | + | | | | | |
| 5 | i_phi_fu_62 | | phi_mux | | | | | |
| 6 | exitcond_fu_70 | | icmp | | | | | |
| 7 | tmp_l_fu_96 | | | | | | + | |

Рис. 5.8. Resource viewer

5.4. C/RTL моделирование

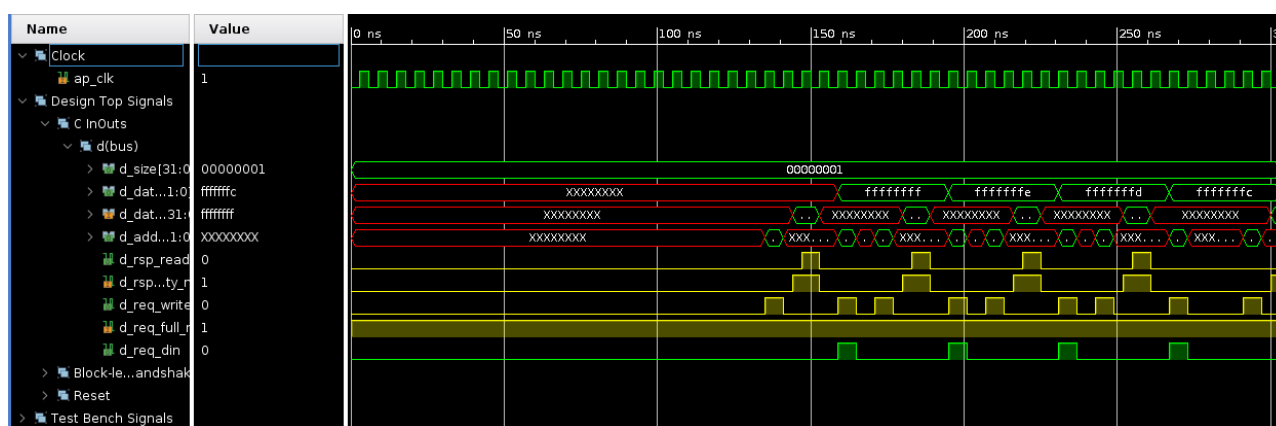


Рис. 5.9. Временная диаграмма

По временной диаграмме видно, что latency составляет 25 тактов, а П – 26 тактов.

6. Вывод

В данной лабораторной работе были рассмотрены протоколы Port-Level I/O. Выяснено, что для арифметических операций с указателем необходим протокол ap_bus.