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

Лабораторная №13

Предмет: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Сравнение типов данных

Задание 2

Студенты:

Соболь В.

Темнова А.С.

<u>Группа: 13541/3</u>

Преподаватель:

Антонов А.П.

Содержание

| 6. | Вывод | 9 |
|------------|---------------------------------------|---|
| | 5.1. Моделирование | 7 |
| 5 . | Решение 2а | 7 |
| 4. | Решение 1a 4.1. Моделирование | 6 |
| 3. | Скрипт | 6 |
| 2. | Исходный код | 4 |
| 1. | Задание | 3 |

1. Задание

- 1. Создать проект lab13 2
- 2. Микросхема: ха7а12tcsg325-1q
- 3. В папке source текст функции apint_promotion Познакомьтесь с ним.
- 4. Познакомьтесь с тестом.
- 5. Исследование:
- 6. Solution 1a
 - Создать версию apint , в которой будет убран Кастинг

```
#include "apint_promotion.h"
|dout_t apint_promotion(din_t a,din_t b) {
    dout_t tmp;

    tmp = (dout_t)a * (dout_t)b;
    return tmp;
}
```

- 7. Осуществить моделирование (при необходимости изменить тест)
- 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 apint promotion

- 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
- 13. Сравнить два решения (solution_1a и solution_2a) и сделать выводы

2. Исходный код

Ниже приведен исходный код устройства и теста.

```
1 #include "apint_promotion.h"
3
  dout_t apint_promotion(din_t a, din_t b) {
4
    dout t tmp;
    #ifdef ENABLE TYPE CAST
6
    tmp = (dout t)a * (dout t)b;
    \#else
8
      tmp = a * b;
9
    #endif
10
    return tmp;
11 }
```

Рис. 2.1. Исходный код устройства

```
#ifndef _APINT_PROMOTION_H_
#define _APINT_PROMOTION_H_

#include <stdio.h>
#include "ap_cint.h"

typedef int18 din_t;
typedef int36 dout_t;

dout_t apint_promotion(din_t a,din_t b);

#endif
```

Рис. 2.2. Заголовочный файл

```
1 #include "apint promotion.h"
3
  int main () {
     din t A, B;
4
5
     dout_t RES;
6
     int i, retval=0;
7
     FILE
                   *fp;
8
9
     // Save the results to a file
10
     fp=fopen("result.dat", "w");
11
12
     // Call the function
13
     A = 65536;
14
     B = 65536;
15
     for (i=0; i<(20);++i)
16
       RES=apint promotion (A,B);
  #ifndef MINGW32
17
18
            fprintf(fp, "\%lld \ \ \ \ RES);
19
  #else
20
            fprintf(fp, "\%I64d \ \ \ \ ", RES);
21
  #endif
22
       A = A + 1024;
23
       B=B-2047;
24
25
     fclose (fp);
26
27
     // Compare the results file with the golden results
28
     retval = system("diff_—brief_-w_result.dat_result.golden.dat");
29
     if (retval != 0) {
30
       printf("Test_failed__!!!\n");
31
       retval=1;
     } else {
32
33
       printf("Test_passed_!\n");
34
35
36
     // Return 0 if the test passes
37
     return retval;
38
```

Рис. 2.3. Исходный код теста

3. Скрипт

Ниже приводится скрипт, для автоматизации выполнения лабораторной работы.

```
open project -reset lab13 2
2
3
4
  add_files apint_promotion.c
  add files -tb apint promotion test.c
5
  add files -tb result.golden.dat
8
  set top apint promotion
10 # open solution solution 1a -reset
11 | \# \text{ set\_part } \{ xa7a12tcsg325 - 1q \}
12 # create_clock -period 10 ns
13 # set_clock_uncertainty 0.1
14
15
16 # csim design
17 # csynth design
18 # cosim design -trace level all
19
20
21
  add files apint promotion.c -cflags "-w_-DENABLE TYPE CAST"
22
23 open solution solution 2a -reset
24 | \text{set part } \{ \text{xa7a12tcsg325} - 1 \text{q} \}
25 create clock -period 10ns
26
  set_clock_uncertainty 0.1
27
28
29 csim design
30 csynth design
31 # cosim design -trace level all
```

Рис. 3.1. Скрипт

4. Решение 1а

4.1. Моделирование

Ниже приведены результаты моделирования.

Рис. 4.1. Результаты моделирования

По результатам моделирования видно, что устройство работает не корректно.

5. Решение 2а

5.1. Моделирование

Ниже приведены результаты моделирования.

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.

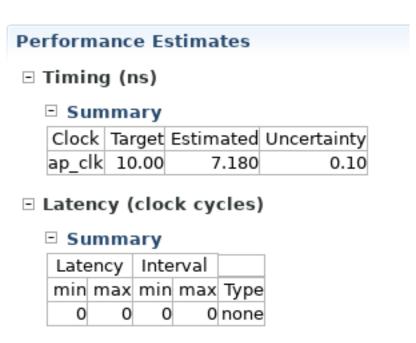


Рис. 5.2. Performance estimates

Utilization Estimates

∃ Summary

| Name | BRAM_ | 18K | DSP48E | FF | LUT |
|-----------------|-------|-----|--------|-------|------|
| DSP | - | |] | | - |
| Expression | - | | - | - | - |
| FIFO | - | | - | - | - |
| Instance | - | | - | - | - |
| Memory | - | | - | - | - |
| Multiplexer | - | | - | - | - |
| Register | - | | - | - | - |
| Total | | 0 |] | . 0 | 0 |
| Available | | 40 | 40 | 16000 | 8000 |
| Utilization (%) | | 0 | 2 | 0 | 0 |

Рис. 5.3. Utilization estimates

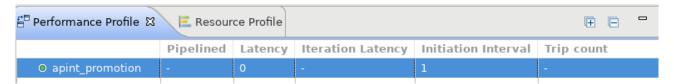


Рис. 5.4. Performance profile



Рис. 5.5. Scheduler viewer

| | Resource\Control S | C0 |
|---|--------------------|------|
| 1 | ⊡I/O Ports | |
| 2 | b | read |
| 3 | ap_return | ret |
| 4 | a | read |
| 5 | ⊡Expressions | |
| 6 | tmp_2_fu_34 | * |
| | | |

Рис. 5.6. Resource viewer

6. Вывод

В данной работе продемонстрирована важность порядка преобразования типов. В первом решении, операция умножения выполняется для двух операндов с типом d_in , который значительно меньше, чем d_out . Во время умножения происходит переполнение, и результат оказывается некорректным.

Во втором решении, умножение выполняется над операндами с типом d_{out} , так как произведено явное преобразование типов. Это тип более вместительный, и переполнения не происходит.