

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

Отчет по лабораторной работе №11_2
Курс: «Проектирование реконфигурируемых гибридных
вычислительных систем»
Тема: « Задержка (Latency) »

Выполнил студент гр. 3540901/81501

Селиверстов С.А.

(подпись)

Руководитель

Антонов А.П.

(подпись)

“ ” _____ 2019 г.

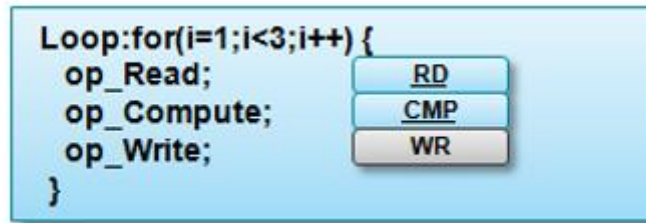
Санкт – Петербург
2019

ОГЛАВЛЕНИЕ

1. Задание.....	3
2. Исходный код.....	7
3. Моделирование	9
4. Исследование	10
4.1. Решение 1а	10
4.1.2.Синтез решения 1а	10
4.1.3. C/RTL моделирование	12
4.2. Решение 2а	12
4.2.2.Синтез решения 2а	13
4.3. Решение 3а	13
4.3.2.Синтез решения 3а	13
4.4. Решение 4а	14
4.4.2.Синтез решения 4а	14
4.5. Решение 5а	14
4.5.2.Синтез решения 5а	14
Вывод.....	15

1. Задание

- Создать проект lab11_2
- Микросхема: ха7a12tcsg325-1q
- Создать функцию по образцу (иерархия функций)



Op_read: t_in=d_in[i]

Op_compute: t_r=t_in*t_in;

Op_write: d_out[i]=t_r;

- Создать тест lab11_2_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
- Исследование:
- Solution_1a
 - 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
- Solution_2a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию LATENCY для одной итерации

```
Loop_A: for (i=0; i<N; i++)  
{
```

```
#pragma HLS latency
  ..Loop Body...
}
```

- 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
- Сравнить два решения (solution_1a и solution_2a) и сделать выводы: зависимость от LATENCY ; объяснить (посчитать) число циклов Latency, П...
- Solution_3a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию LATENCY для всего цикла

```
#pragma HLS latency

Loop_A: for (i=0; i<N; i++)
{
  ..Loop Body...
}
```

- 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
- Сравнить два решения (solution_2a и solution_3a) и сделать выводы: зависимость от LATENCY ; объяснить (посчитать) число циклов Latency, П...
- Solution_4a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию LATENCY с опцией min=5
 - 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
- Сравнить два решения (solution_3a и solution_4a) и сделать выводы: зависимость от LATENCY; объяснить (посчитать) число циклов Latency, П
- Solution_5a
 - задать: clock period 10; clock_uncertainty 0.1
 - установить реализацию LATENCY с опцией max=8
 - 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
-
- Сравнить два решения (solution_3a и solution_5a) и сделать выводы: зависимость от LATENCY; объяснить (посчитать) число циклов Latency, П

2. Исходный код

Зададим следующий код устройства:

```
#include "lab11_2.h"
void lab11_2(int d_in[N], int d_out[N])
{
    int i;
    int t_in, t_r;
    Loop: for(i = 0; i < N; i++)
    {
        t_in = d_in[i];
        t_r = t_in * t_in;
        d_out[i] = t_r;
    }
}
```

Заголовочный файл определим как:

```
#define N 16
```

Код теста для проверки функции имеет вид:

```
#include "lab11_2.h"
#include <stdio.h>

int main()
{
    int d_in[N], d_actual[N], d_expected[N];
    int passed = 1;
    int i;
    for(i = 0; i < N; i++)
    {
        d_in[i] = i;
        d_expected[i] = i * i;
    }

    lab11_2(d_in, d_actual);

    for(i = 0; i < N; i++)
    {
        printf("Expected_[%d]_actual_[%d]\n", d_expected[i], d_actual[i]);
        if(d_expected[i] != d_actual[i])
        {
            passed = 1;
        }
    }
    if(passed != 1){
        printf("————Test_failed————\n");
    } else {
        printf("————Test_passed————\n");
    }
}
```

Скрипт для запуска программы с консоли имеет вид:

```
open_project -reset lab11_2

add_files lab11_2.c
add_files -tb lab11_2_test.c
set_top lab11_2

set solutions [list 1a 2a 3a 4a 5a]

foreach sol $solutions {
    open_solution solution_$sol -reset
    set_part {xa7a12tcsg325-1q}
    create_clock -period 10ns
    set_clock_uncertainty 0.1

    if {$sol == "2a"} {
        set_directive_latency "lab11_2/Loop"
    }
    if {$sol == "3a"} {
        set_directive_latency "lab11_2"
    }
    if {$sol == "4a"} {
        set_directive_latency -min 5 "lab11_2"
    }
    if {$sol == "5a"} {
        set_directive_latency -max 8 "lab11_2"
    }

    csim_design
    csynth_design
    cosim_design -trace_level all
}

exit
```


3. Моделирование

Результаты моделирования, подтверждающие корректность работы устройства, имеют вид:

```
INFO: [APCC 202-1] APCC is done.  
    Generating csim.exe  
Expected [0] actual [0]  
Expected [1] actual [1]  
Expected [4] actual [4]  
Expected [9] actual [9]  
Expected [16] actual [16]  
Expected [25] actual [25]  
Expected [36] actual [36]  
Expected [49] actual [49]  
Expected [64] actual [64]  
Expected [81] actual [81]  
Expected [100] actual [100]  
Expected [121] actual [121]  
Expected [144] actual [144]  
Expected [169] actual [169]  
Expected [196] actual [196]  
Expected [225] actual [225]  
-----Test passed-----  
INFO: [SIM 211-1] CSim done with 0 errors.  
INFO: [SIM 211-3] ***** CSIM finish *****
```

4. Исследование

4.1. Решение 1a

В соответствие с планом лабораторной работы устанавливаем:

- clock period = 10;
- clock uncertainty = 0.1;
- реализация ПО УМОЛЧАНИЮ

Директивы данного решения имеют вид:

▼ ● lab11_2
 ● d_in
 ● d_out
 x-y
 =? Loop

4.1.2.Синтез решения 1a

Результаты оценки производительности имеют вид:

Performance Estimates

▣ Timing (ns)

▣ Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.470	0.10

▣ Latency (clock cycles)

▣ Summary

Latency		Interval		
min	max	min	max	Type
65	65	65	65	none

По данным результатам синтеза можно сделать вывод, что оценка производительности видно устройства соответствует заданным критериям.

Оценка использования имеет вид:

Utilization Estimates				
Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	3	0	47
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	-	-	-	-
Multiplexer	-	-	-	42
Register	-	-	84	-
Total	0	3	84	89
Available	40	40	16000	8000
Utilization (%)	0	7	~0	1

Профиль производительности имеет вид:

Performance Profile					
Resource Profile					
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count
lab11_2	-	65	-	66	-
Loop	no	64	4	-	16

Данные планировщика просмотра имеет вид:

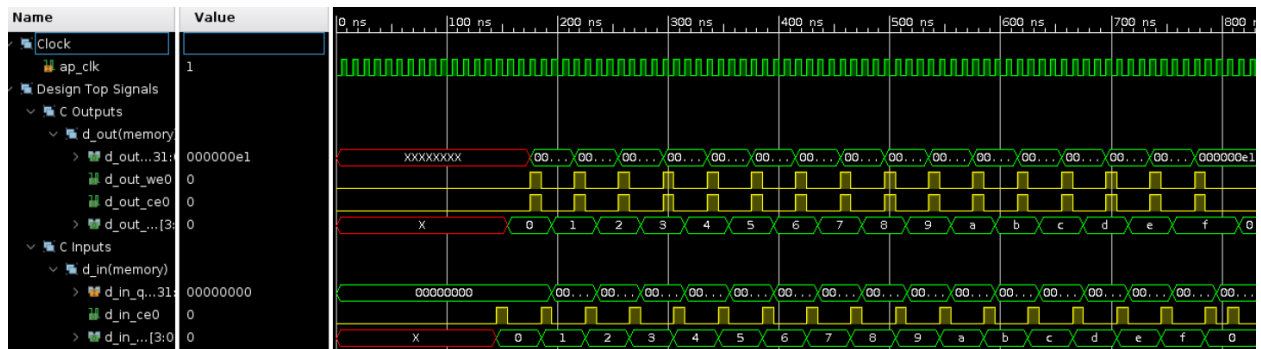
Operation\Control Step	0	1	2	3	4
Loop	- Loop				
i(phi_mux)					
exitcond(icmp)					
i_1(+)					
t_in(read)					
t_r(*)					
node_20(write)					

Данные обзора ресурсов имеет вид:

	Resource\Control Step	C0	C1	C2	C3	C4
1	I/O Ports					
2	d_in(p0)		read			
3	d_out(p0)					write
4	Memory Ports					
5	d_in(p0)		read			
6	d_out(p0)					write
7	Expressions					
8	i_1_fu_69		+			
9	i_phi_fu_56		phi_mux			
10	exitcond_fu_63		icmp			
11	t_r_fu_80				*	

4.1.3. C/RTL моделирование

Временная диаграмма результатов C/RTL моделирования имеет вид:



По результатам временной диаграммы можно сделать вывод, что выполнение одного цикла требует 4 такта (16 итераций цикла) и плюс 1 такт для инициализации, таким образом, $Latency = 4 * 16 + 1 = 65$, $\Pi = 66$ тактов.

4.2. Решение 2а

В соответствие с планом лабораторной работы устанавливаем:

- clock period 10;
- clock_uncertainty 0.1
- установить реализацию LATENCY для одной итерации
 - Loop_A: for (i=0; i<N; i++)
 - {
 - #pragma HLS latency
 - ..Loop Body...
 - }

Директивы данного решения имеют вид:

- lab11_2
 - d_in
 - d_out
 - x+y
=? Loop

% HLS LATENCY

4.2.2. Синтез решения 2а

Результаты синтеза идентичны Решению 1а

4.3. Решение 3а

В соответствие с планом лабораторной работы устанавливаем:

- clock period 10;
- clock uncertainty 0.1
- установить реализацию LATENCY для всего цикла
 - #pragma HLS latency
 - Loop_A: for (i=0; i<N; i++)
 - {
 - ..Loop Body...
 - }

Директивы данного решения имеют вид:

```
▼ ● lab11_2
    % HLS LATENCY
    ● d_in
    ● d_out
    X+Y
    =? Loop
```

4.3.2. Синтез решения 3а

Результаты синтеза идентичны Решению 1а

4.4. Решение 4a

В соответствие с планом лабораторной работы устанавливаем:

- clock period 10;
- clock_uncertainty 0.1
- реализацию LATENCY с опцией min=5

Директивы данного решения имеют вид:

```
▼ ● lab11_2
    % HLS LATENCY min=5
    ● d_in
    ● d_out
    x=y
    =? Loop
```

4.4.2.Синтез решения 4a

Результаты синтеза идентичны Решению 1a

4.5. Решение 5a

В соответствие с планом лабораторной работы устанавливаем:

- clock period 10;
- clock uncertainty 0.1
- реализацию LATENCY с опцией max=8

Директивы данного решения имеют вид:

```
▼ ● lab11_2
    % HLS LATENCY max=8
    ● d_in
    ● d_out
    x=y
    =? Loop
```

4.5.2.Синтез решения 5a

Результаты синтеза идентичны Решению 1a

Вывод

При проведении исследований для данной функции никаких отличий при применении директивы LATENCY выявлено не было.