Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологии

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab2\_z0

Дисциплина: Проектирование реконфигурируемых гибридных вычислительных систем

Выполнил студент Курякин Д.

Гр. 3540901/12001

Руководитель, доцент Антонов А.П.

«\_\_» \_\_\_\_\_\_ 2022

Санкт-Петербург

2022

Оглавление

[1. Задание 3](#_Toc116916474)

[2. Исходный код функции 3](#_Toc116916475)

[3. Исходный код теста 3](#_Toc116916476)

[3.1 Моделирование 4](#_Toc116916477)

[4. Первое решение — Solution1 5](#_Toc116916478)

[4.1 Исходные настройки 5](#_Toc116916479)

[4.2 Синтез 5](#_Toc116916480)

[5. Первое решение — Solution2 7](#_Toc116916481)

[5.1 Исходные настройки 7](#_Toc116916482)

[5.2 Синтез 8](#_Toc116916483)

[6. Сравнение Solution1 и Solution2 9](#_Toc116916484)

[7. Выводы 10](#_Toc116916485)

1. Задание

* Разработать на языке Си описание функции, реализующей следующий алгоритм outArr[i] = inArr[i] + inA + inB + inC, где i=0, 1, .. (ROWS-1). ROWS=3
* Тип данных для inArr[i] , inA, inB, inC, inD – short; для outArr[i] – int;
* Разработать тест, обеспечивающий автоматическую проверку получаемых результатов моделирования
* разработанной функции.
* Создать, провести исследование и сравнительный анализ двух аппаратных реализаций разработанного на
* языке Си описания функции.
  + Микросхема: xa7a12tcsg325-1q
  + clock period 6; clock\_uncertainty 1 (для решения Solution 1)
  + clock period 10; clock\_uncertainty 1 (для решения Solution 2)

**2. Исходный код функции**

Исходный код заголовочного файла приведен ниже.

|  |
| --- |
| **#define** **ROWS** 3  **typedef** **short** **din\_type**;  **typedef** **int** **dout\_type**;  // Prototype of top level function for C-synthesis  **void** **lab2\_z0** (**din\_type** in\_a[ROWS], **din\_type** in\_b[ROWS], **dout\_type** res[ROWS]); |

Исходный код синтезируемой функции приведен ниже.

|  |
| --- |
| **#include** "lab2\_z0.h"  **void** **lab2\_z0** (**din\_type** in\_a[ROWS], **din\_type** in\_b[ROWS], **dout\_type** res[ROWS])  {  **din\_type** tmp\_a, tmp\_b;  **dout\_type** tmp\_res;  **for**(**int** i = 0; i < ROWS; i++)  {  **#pragma** HLS PIPELINE off  tmp\_a = in\_a[i];  tmp\_b = in\_b[i];  tmp\_res = tmp\_a + tmp\_b;  res[i] = tmp\_res;  }  } |

Функция принимает на вход массив, 3 аргумента — скаляра и выходной массив. Затем считает значение по формуле и возвращает результат расчета в принимаемый массив.

**3. Исходный код теста**

Исходный код теста для проверки функции lab1\_z2 приведен ниже.

|  |
| --- |
| **#include** <stdio.h>  **#include** "lab2\_z0.h"  //function for comparing expected and actual results in arrays  **int** **arentEqual**(**dout\_type** expectedArr[ROWS], **dout\_type** actualArr[ROWS])  {  // Linearly compare elements  **for** (**int** i = 0; i < ROWS; i++)  **if** (expectedArr[i] != actualArr[i])  {  **fprintf**(stdout, " ERROR: expected=%d actual=%d for ROW: %d \n",  expectedArr[i], actualArr[i], i);  **return** 1;  }  // If all elements were the same.  **return** 0;  }  **int** **main**() {  **din\_type** inArr\_a[ROWS], inArr\_b[ROWS];  **dout\_type** expectedArr[ROWS], actualArr[ROWS];  **int** pass = 0;  // Calling the function for 3 times and the results comparing  **for** (**int** i = 0; i < 3; i++)  {  // initial settings  **for** (**int** j=0; j<ROWS; j++)  {  inArr\_a [j] = **rand**() % 1000;  inArr\_b [j] = inArr\_a [j] + **rand**() % 1000;  actualArr[j]= 0;  }  //function invocation and getting actual results  **lab2\_z0**(inArr\_a, inArr\_b, actualArr);  //expected results evaluation  **for** (**int** i=0; i<ROWS; i++)  expectedArr[i] = inArr\_a [i] + inArr\_b [i];  // Compare the actual results against the expected results  **if** (**arentEqual**(expectedArr, actualArr))  pass = 1;  }  **if** (!pass)  **fprintf**(stdout, "----------Pass!------------ \n");  **else**  **fprintf**(stderr, "----------Fail!------------ \n");  **return** pass;  }; |

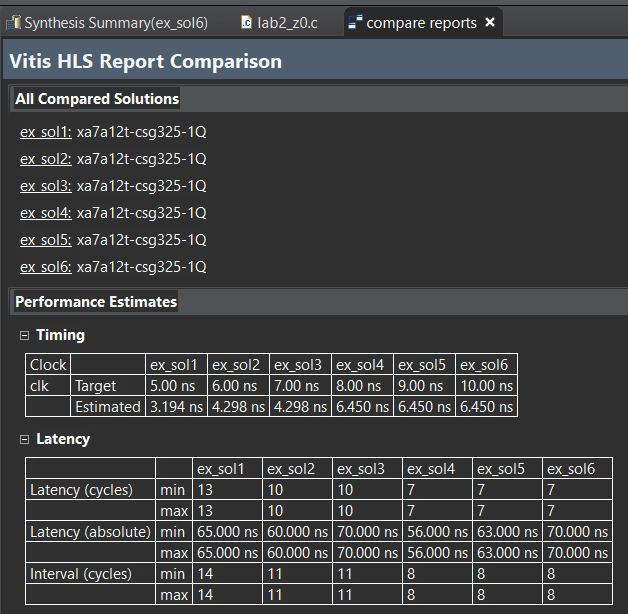
Тест обеспечивает проверку корректной работы функции.

В листинге приведен tcl скрипт.

|  |
| --- |
| open\_project -reset lab2\_z0\_ex  add\_files ./source/lab2\_z0.c  set\_top lab2\_z0  add\_files -tb ./source/lab2\_z0\_test.c  open\_solution -reset "ex\_sol1"  set\_part {xa7a12tcsg325-1Q}  create\_clock -period 5 -name clk  set\_clock\_uncertainty 1  csynth\_design  csim\_design -clean  set all\_solutions {ex\_sol2 ex\_sol3 ex\_sol4 ex\_sol5 ex\_sol6}  set all\_period {{6} {7} {8} {9} {10}}  foreach the\_solution $all\_solutions the\_period $all\_period {  open\_solution -reset $the\_solution  create\_clock -period $the\_period -name clk  set\_clock\_uncertainty 1  set\_part {xa7a12tcsg325-1Q}  csynth\_design  cosim\_design -trace\_level all -tool xsim  } |

**6. Сравнение решений**

Был запущен tcl скрипт. После выполнения скрипта была выполнена команда Compare Reports. После которой получаем результаты сравнения решений.



*Рис. 1 Сравнение решений*

*Рис. 2 Диаграмма сравнения решений*

**7. Выводы**

В данной работе были получен опыт работы с автоматической сборкой проекта.