

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологии  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab2\_Z2

Дисциплина: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Введение в Vivado HLS Tool CLI Flow

Выполнил студент гр. 01502

С.С. Гаспарян

Руководитель, доцент

Антонов А.П.

«23» сентября 2021

Санкт-Петербург  
2021

## 1. Задание

Текст задания находится в файле «Задание lab2\_2.docx»

## 2. Исходный код функции

Исходный код синтезируемой функции представлен на рисунке 1.

```
1 #include "lab2_2_inc.c"
2
3 atype find_max(atype a_in[N])
4 {
5     atype max = a_in[0];
6     for(int i = 0; i < N; ++i){
7         if (max < a_in[i])
8             max = a_in[i];
9     }
10    return max;
11 }
```

Рис. 1 Исходный код функции find\_max

Функция принимает 1 аргумент — массив целых чисел и возвращает максимальное число в массиве.

## 3. Исходный код теста

Исходный код теста для проверки функции find\_max приведен на рисунке 2.

Тест обеспечивает проверку корректную работу функции.

## 4. Исходный код модифицированного теста

Исходный код модифицированного теста для проверки функции find\_max приведен на рисунке 3. Тест обеспечивает проверку производительности функции и функция была скомпилирована компилятором gcc-9.3.0. В таблице 1 представлены характеристики ПК:

Таблица 1:

CPU	Intel Core i5-6200U 2.30 GHz
Core	2
Threads	4
RAM	8 Gb

```

515 94154,19733,5945,76967,99888,84828,80777,117984,36944,81731,27623,123211,28724,81523,65849,99
516 78618,102949,120582,53872,72892,99549,100123,48457,106166,124517,73235,101806,83704,84845,431
517 104578,62695,12777,76655,19711,107131,80403,70231,61050,108026,79206,89774,61737,30820,74852,
518 5957,81198,79992,92426,52935,52303,26647,31289,62584,99883,18860,18477,70492,62034,82099
519 };
520
521 int main() {
522
523     int pass = 0;
524     int max_elem = 127789;
525     int res_max;
526
527
528     for(int i = 0; i < 32; ++i){
529         res_max = find_max(inArr_a);
530         if (res_max != max_elem){
531             fprintf(stdout, " expected %d != got %d ERROR\n", max_elem, res_max);
532             pass = 1;
533         }
534         else{
535             fprintf(stdout, " expected %d == got %d PASS \n", max_elem, res_max);
536         }
537     }
538
539     if (pass == 0)
540     {
541         fprintf(stdout, "-----Pass!-----\n");
542         return 0;
543     }
544     else
545     {
546         fprintf(stderr, "-----Fail!-----\n");
547         return 1;
548     }
549
550 }

```

Рис. 2 Исходный код теста lab2\_2\_test

```

522 int main() {
523     int pass = 0;
524     int max_elem = 127789;
525     int res_max;
526     struct timespec t0, t1;
527     double acc_time = 0.0;
528
529     for(int i = 0; i < 32; ++i){
530         if(clock_gettime(CLOCK_REALTIME, &t0) != 0) {
531             perror("Error in calling clock_gettime\n");
532             exit(EXIT_FAILURE);
533         }
534         res_max = find_max(inArr_a);
535         if(clock_gettime(CLOCK_REALTIME, &t1) != 0) {
536             perror("Error in calling clock_gettime\n");
537             exit(EXIT_FAILURE);
538         }
539         double diff_time = (((double)(t1.tv_sec - t0.tv_sec))*1000000000.0) + (double)(t1.tv_nsec - t0.tv_nsec);
540         acc_time += diff_time;
541         double temp_avg_time = acc_time / (i + 1); // take average time
542         printf("Elapsed time: %.4lf nanoseconds\n", temp_avg_time);
543
544         if (res_max != max_elem){
545             fprintf(stdout, "Complete expected %d != got %d ERROR\n", max_elem, res_max);
546             pass = 1;
547         }
548         else{
549             fprintf(stdout, "Complete expected %d == got %d PASS \n", max_elem, res_max);
550         }
551     }
552
553     if (pass == 0)
554     {
555         fprintf(stdout, "-----Pass!-----\n");
556         return 0;

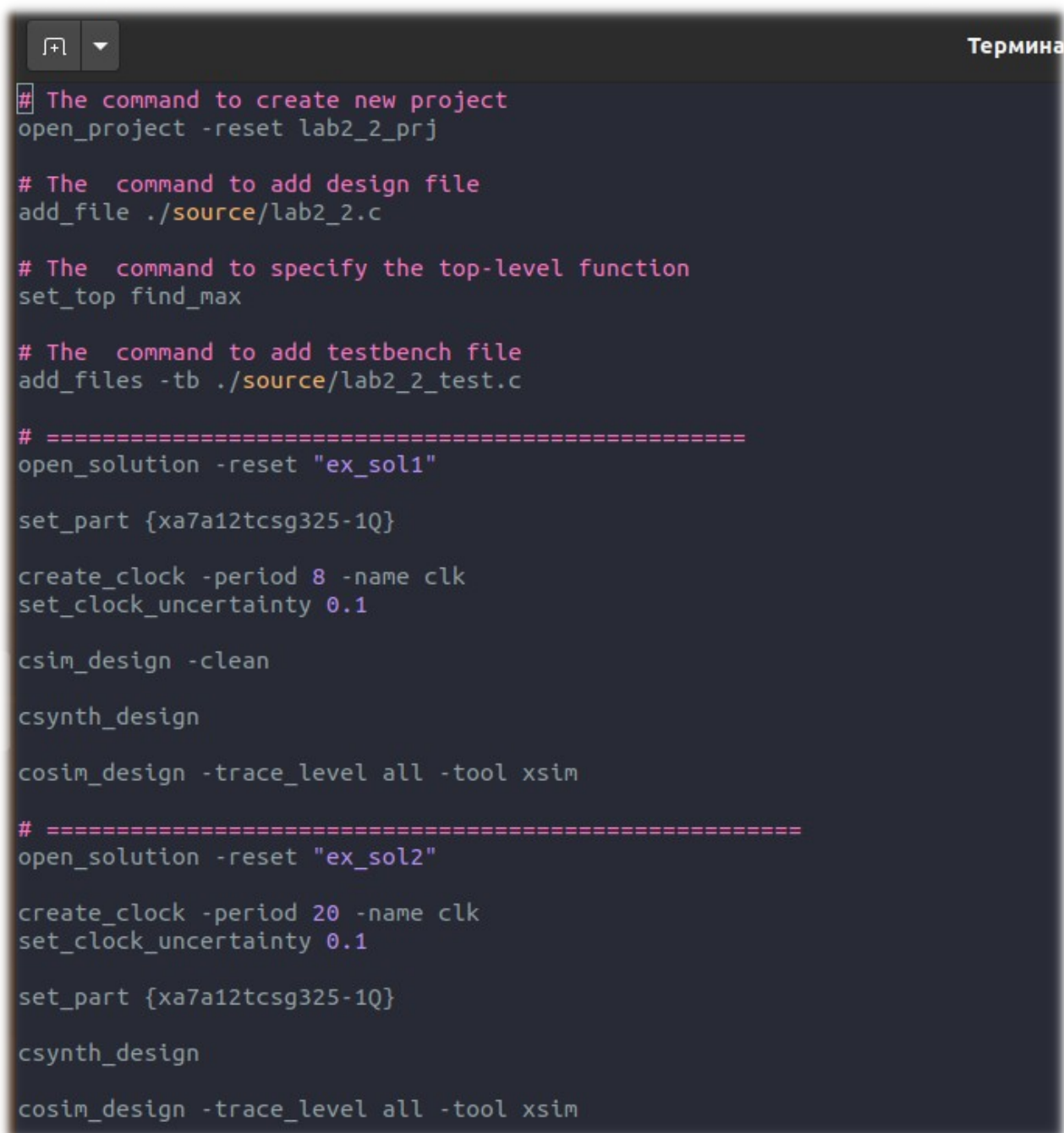
```

Рис. 3 Исходный код модифицированного теста

## 5. Командный файл

На рисунке 4 представлен текст команд для автоматизации создания вариантов аппаратной реализации:

- a. Для ex\_sol1 задается clock period 6; clock\_uncertainty 0.1
- b. Для ex\_sol2 задается clock period 20; clock\_uncertainty 0.1



```
# The command to create new project
open_project -reset lab2_2_prj

# The command to add design file
add_file ./source/lab2_2.c

# The command to specify the top-level function
set_top find_max

# The command to add testbench file
add_files -tb ./source/lab2_2_test.c

# =====
open_solution -reset "ex_sol1"

set_part {xa7a12tcsg325-1Q}

create_clock -period 8 -name clk
set_clock_uncertainty 0.1

csim_design -clean

csynth_design

cosim_design -trace_level all -tool xsim

# =====
open_solution -reset "ex_sol2"

create_clock -period 20 -name clk
set_clock_uncertainty 0.1

set_part {xa7a12tcsg325-1Q}

csynth_design

cosim_design -trace_level all -tool xsim
```

Рис. 4 Скрипт с командами для создания проекта

## 6. Результаты исследования

### 6.1 Сравнение решений

На рисунке 5 представлено сравнение решений из Vivado HLS GUI по аппаратным ресурсам и временным параметрам.

Performance Estimates			
Timing			
Clock		ex_sol1	ex_sol2
clk	Target	6.00 ns	20.00 ns
	Estimated	3.257 ns	6.491 ns
Latency			
		ex_sol1	ex_sol2
Latency (cycles)	min	24578	16386
	max	24578	16386
Latency (absolute)	min	0.147 ms	0.328 ms
	max	0.147 ms	0.328 ms
Interval (cycles)	min	24578	16386
	max	24578	16386
Utilization Estimates			
		ex_sol1	ex_sol2
BRAM_18K0		0	
DSP48E	0	0	
FF	97	64	
LUT	150	144	
URAM	0	0	

Рис. 5 Сравнение решений

## 6.2 Электронная таблица и график для решений

На рисунке 6 представлена таблица с параметрами для все решений. На рисунке 7 представлен график для сравнения всех решений.

		ex_sol1	ex_sol2
Clock	Target ( ns)	6	20
	Estimated ( ns)	3,26	6,49
Latency	(cycles)	16386	16386
	(ns)	53369	106362
Resources	BRAM_18K	0	0
	DSP48E	0	0
	FF	97	64
	LUT	150	144
	URAM	0	0

Рис. 6. Таблица данных для всех решений

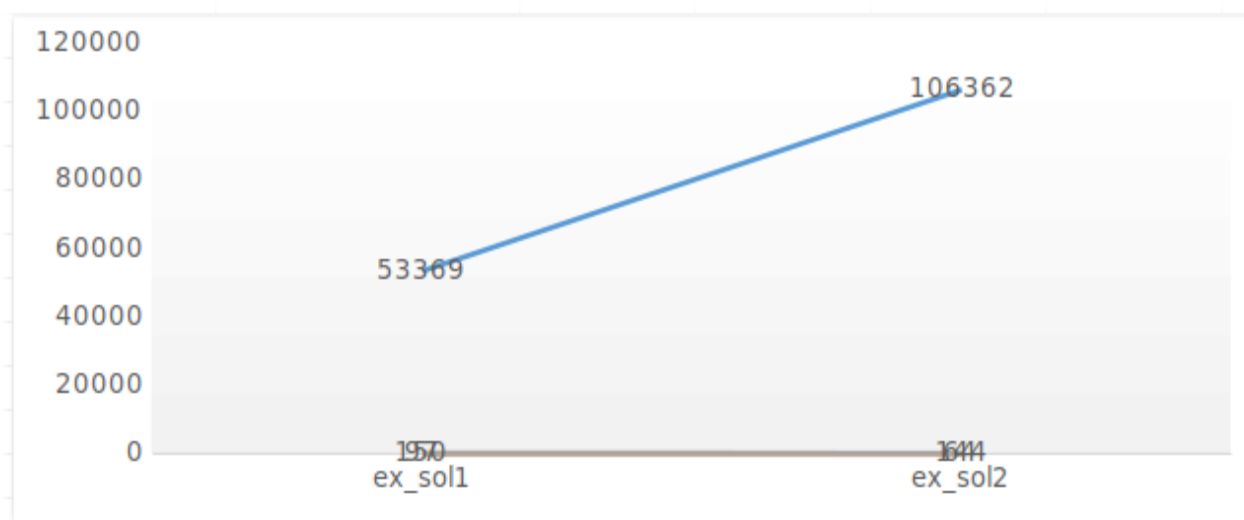


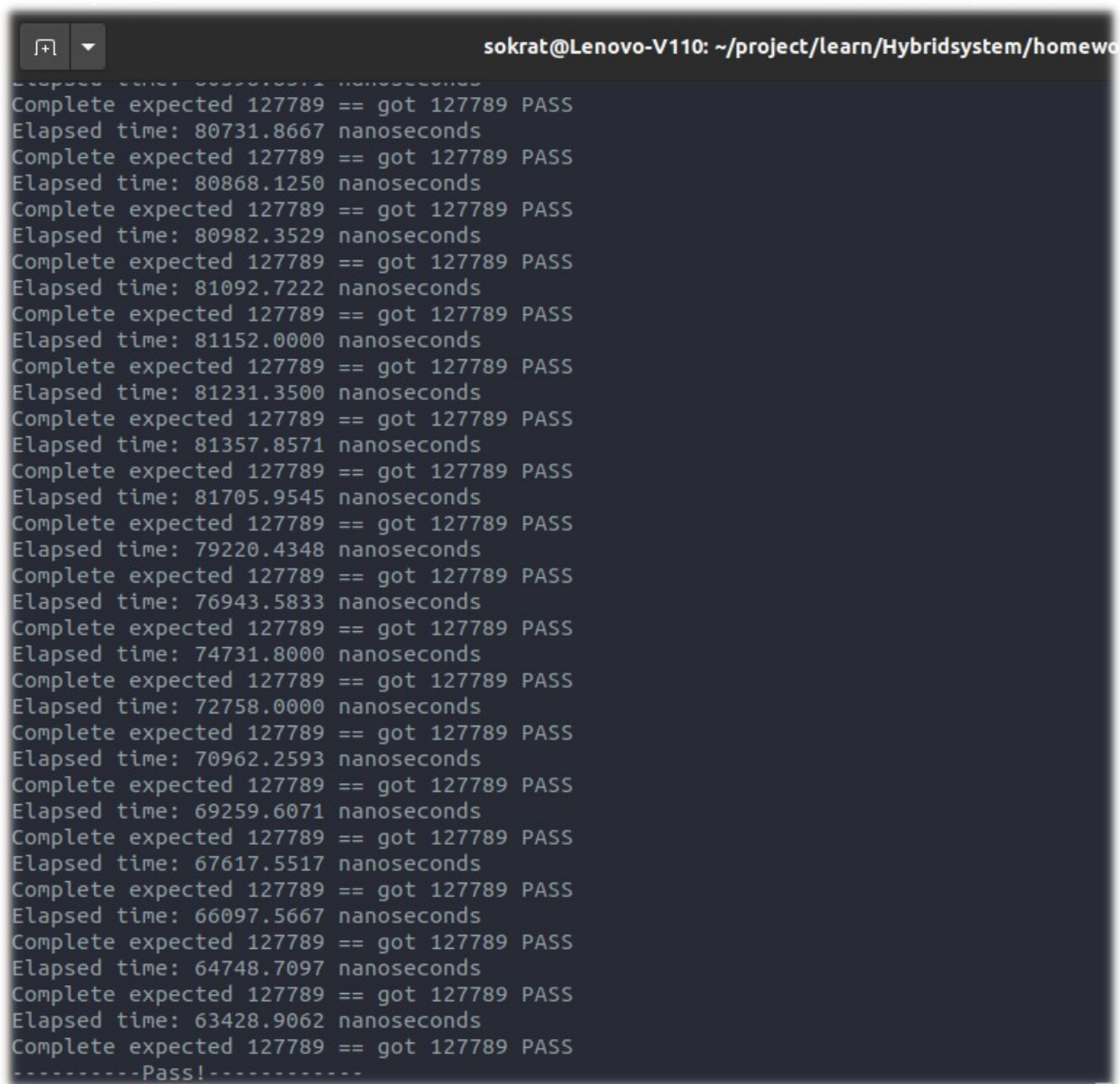
Рис. 7 График данных для всех решений

Как видно из таблицы и графика все решения являются одинаковыми. У них одинаковое затраченное(Estimated) время 6,49 нс и одинаковое количество тактов Latency 16386. Также одинаковое количество затраченных аппаратных ресурсов — FF и LUT.



### 6.3 Тест производительности

На рисунке 8 представлен результат модифицированного теста. Среднее время теста составило 63428 нс.

A screenshot of a terminal window with a dark background. The title bar at the top reads 'sokrat@Lenovo-V110: ~/project/learn/Hybridsystem/homewo'. The terminal displays a series of test results. Each line consists of a status message, an expected value, a comparison operator, a got value, and a result. The status messages are 'Complete' and 'Elapsed time:'. The expected and got values are consistently 127789. The results are 'PASS'. The elapsed times are listed in nanoseconds and decrease from 80731.8667 to 63428.9062. The final line is '-----Pass!-----'.

```
sokrat@Lenovo-V110: ~/project/learn/Hybridsystem/homewo
Complete expected 127789 == got 127789 PASS
Elapsed time: 80731.8667 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 80868.1250 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 80982.3529 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 81092.7222 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 81152.0000 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 81231.3500 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 81357.8571 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 81705.9545 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 79220.4348 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 76943.5833 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 74731.8000 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 72758.0000 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 70962.2593 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 69259.6071 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 67617.5517 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 66097.5667 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 64748.7097 nanoseconds
Complete expected 127789 == got 127789 PASS
Elapsed time: 63428.9062 nanoseconds
Complete expected 127789 == got 127789 PASS
-----Pass!-----
```

Рис. 8 Результат теста производительности

Как видно из рисунка 8 результат расчета на ПК оказался гораздо медленнее, чем на плате разница 63426 нс.

### Вывод

В данной работе была изучена возможность создания проекта и решений для проекта в Vivado HLS с помощью командной строки. Для автоматизированного создания проекта был создан скрипт в котором было прописано создание



проекта и решений с различными временными параметрами. Был произведен анализ теста производительности синтезируемой функции на микросхеме и на ПК.