

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

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab2_Z1

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.

«22» сентября 2021

Санкт-Петербург
2021

1. Задание

Текст задания находится в файле «Задание lab2_1.docx»

2. Исходный код функции

Исходный код синтезируемой функции представлен на рисунке 1.

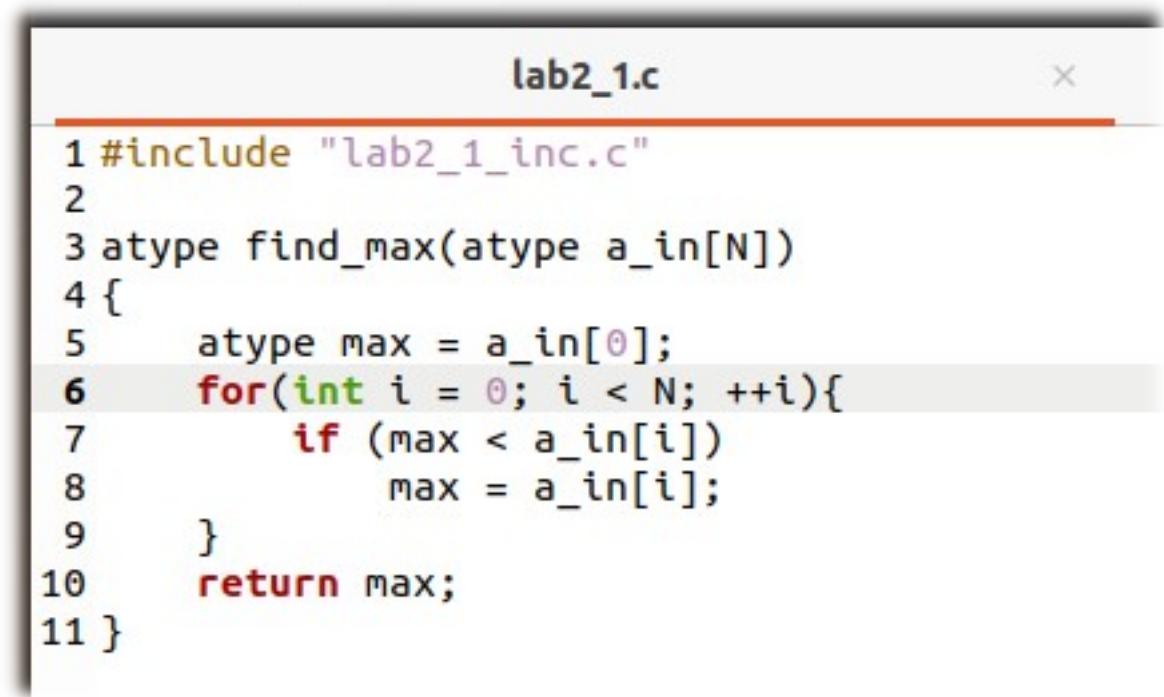


Рис. 1 Исходный код функции find_max

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 корректной функции.

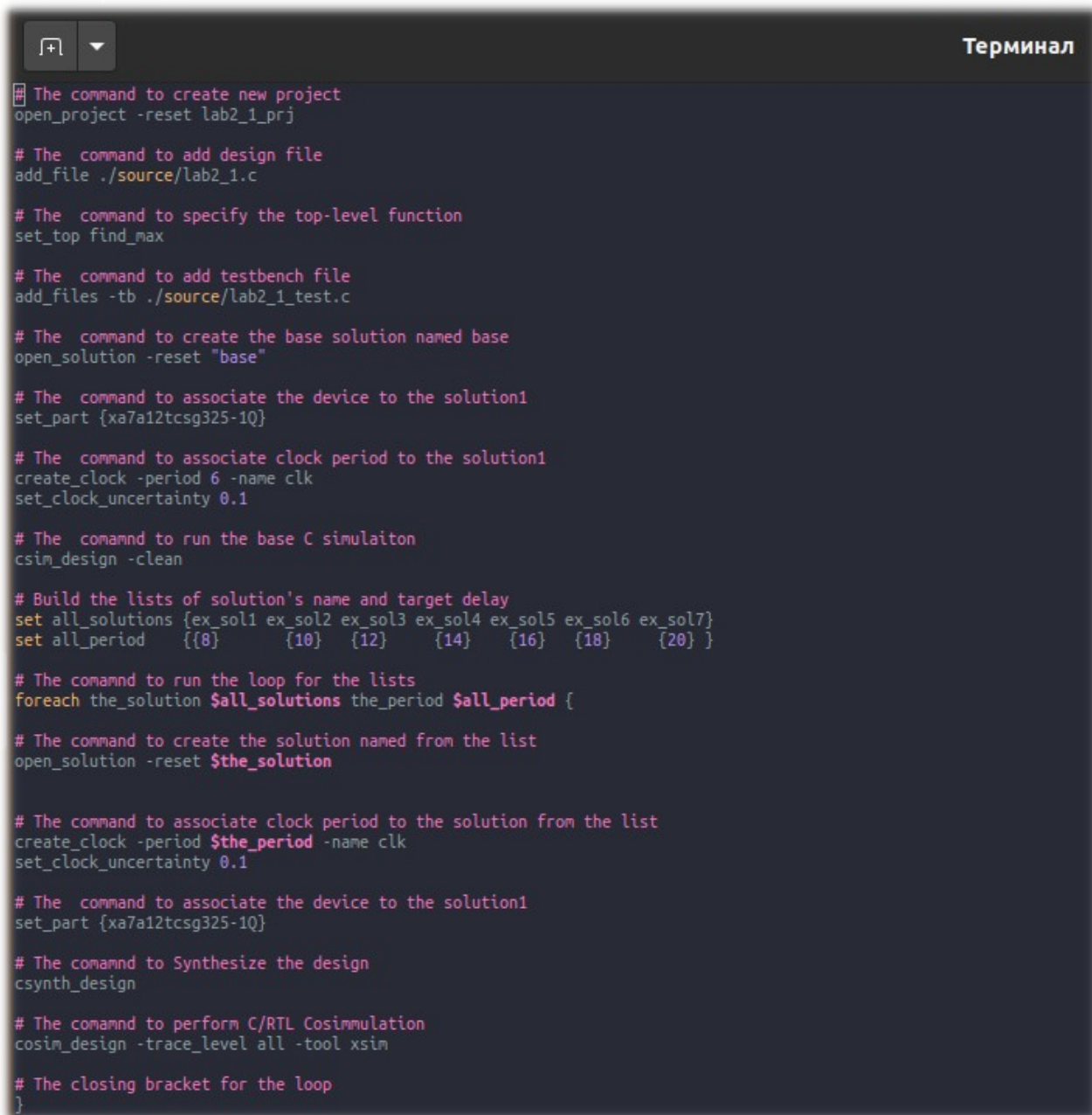
```
lab2_1.c      lab2_1_test.c      lab2_1_inc.c
1  int main() {
2      atype inArr_a[N] = {
3          58508, 20261, 59177, 44240, 57368, 29485, 27736, 37567, 42574, 35046, 63737, 19777, 17315, 61934, 59088, 60276, 32690,
4          21726, 53172, 32436, 14861, 16368, 36642, 45054, 60432, 23030, 45037, 2973, 16617, 2810, 38679, 42302, 23072,
5          65033, 53719, 47617, 28993, 15931, 52361, 6042, 18154, 50573, 58521, 35469, 14159, 19262, 62923, 46849, 8165,
6          50570, 46463, 23026, 34116, 17580, 2556, 29023, 7787, 14770, 31996, 57106, 17580, 5150, 33884, 40652, 37361,
7          54780, 55446, 829, 37888, 42282, 39574, 23220, 60032, 32570, 58689, 8667, 19009, 23264, 22693, 27175, 41012,
8          36333, 50201, 9603, 53913, 19934, 5803, 61701, 34704, 37799, 53282, 52285, 10126, 54343, 60114, 47487, 10776,
9          50036, 15494, 48664, 59495, 22245, 6359, 54003, 21992, 32226, 29847, 41002, 22667, 19717, 2652, 30856, 56051,
10         20030, 7636, 11616, 39965, 13439, 40494, 9144, 18415, 60954, 28606, 28542, 49772, 23196, 43206, 27725
11     };
12     int pass = 0;
13     int max_elem = 65033;
14     int res_max;
15
16     for(int i = 0; i < 2; ++i){
17         res_max = find_max(inArr_a);
18         if (res_max != max_elem){
19             fprintf(stdout, " expected %d != got %d ERROR\n", max_elem, res_max);
20             pass = 1;
21         }
22         else{
23             fprintf(stdout, " expected %d == got %d PASS \n", max_elem, res_max);
24         }
25     }
26
27     if (pass == 0)
28     {
29         fprintf(stdout, "-----Pass!-----\n");
30         return 0;
31     }
32     else
33     {
34         fprintf(stderr, "-----Fail!-----\n");
35     }
36 }
```

Рис 2. Исходный код теста lab2_1_test

4. Командный файл

На рисунке 3 представлен текст команд для автоматизации создания вариантов аппаратной реализации:

- Для ex_sol1 задается clock period 8; clock_uncertainty 0.1
- Для ex_sol2 задается clock period 10; clock_uncertainty 0.1
- Для ex_sol3 задается clock period 12; clock_uncertainty 0.1
- Для ex_sol4 задается clock period 14; clock_uncertainty 0.1
- Для ex_sol5 задается clock period 16; clock_uncertainty 0.1
- Для ex_sol6 задается clock period 18; clock_uncertainty 0.1
- Для ex_sol7 задается clock period 20; clock_uncertainty 0.1
- Для ex_sol0 задается clock period 6; clock_uncertainty 0.1



```
# The command to create new project
open_project -reset lab2_1_prj

# The command to add design file
add_file ./source/lab2_1.c

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

# The command to add testbench file
add_files -tb ./source/lab2_1_test.c

# The command to create the base solution named base
open_solution -reset "base"

# The command to associate the device to the solution1
set_part {xa7a12tcsg325-1Q}

# The command to associate clock period to the solution1
create_clock -period 6 -name clk
set_clock_uncertainty 0.1

# The command to run the base C simulation
csim_design -clean

# Build the lists of solution's name and target delay
set all_solutions {ex_sol1 ex_sol2 ex_sol3 ex_sol4 ex_sol5 ex_sol6 ex_sol7}
set all_period {{8} {10} {12} {14} {16} {18} {20}}

# The command to run the loop for the lists
foreach the_solution $all_solutions the_period $all_period {

# The command to create the solution named from the list
open_solution -reset $the_solution

# The command to associate clock period to the solution from the list
create_clock -period $the_period -name clk
set_clock_uncertainty 0.1

# The command to associate the device to the solution1
set_part {xa7a12tcsg325-1Q}

# The command to Synthesize the design
csynth_design

# The command to perform C/RTL Cosimulation
cosim_design -trace_level all -tool xsim

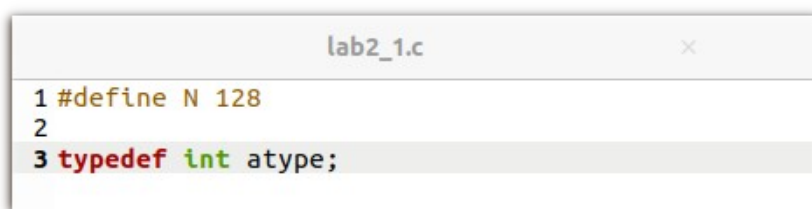
# The closing bracket for the loop
}
```

Рис. 3 скрипт с командами для создания проекта

5. Исследование №1

5.1 Задание исследования №1

На рисунке 4 представлены данные в файле lab2_1_inc.c для исследования №1 — размер массива 128 машинных слов и тип данных int.



```
lab2_1.c
1 #define N 128
2
3 typedef int atype;
```

Рис 4. Содержимое файла lab2_1_inc.c

5.2 Сравнение решений для исследования №1

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

Performance Estimates									
Timing									
Clock		base	ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7
clk	Target	6.00 ns	8.00 ns	10.00 ns	12.00 ns	14.00 ns	16.00 ns	18.00 ns	20.00 ns
	Estimated	3.257 ns	6.491 ns	6.491 ns	6.491 ns	6.491 ns	6.491 ns	6.491 ns	6.491 ns
Latency									
		base	ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7
Latency (cycles)	min	386	258	258	258	258	258	258	258
	max	386	258	258	258	258	258	258	258
Latency (absolute)	min	2.316 us	2.064 us	2.580 us	3.096 us	3.612 us	4.128 us	4.644 us	5.160 us
	max	2.316 us	2.064 us	2.580 us	3.096 us	3.612 us	4.128 us	4.644 us	5.160 us
Interval (cycles)	min	386	258	258	258	258	258	258	258
	max	386	258	258	258	258	258	258	258
Utilization Estimates									
		base	ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7
BRAM_18K0		0	0	0	0	0	0	0	
DSP48E		0	0	0	0	0	0	0	
FF		85	52	52	52	52	52	52	
LUT		144	138	138	138	138	138	138	
URAM		0	0	0	0	0	0	0	

Рис. 5 Сравнение решений исследования №1

5.3 Электронная таблица и график исследования №1

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

		ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7	ex_sol0
Clock	Target (ns)	8	10	12	14	16	18	20	6
	Estimated (ns)	6,49	6,49	6,49	6,49	6,49	6,49	6,49	3,26
Latency	(cycles)	258	258	258	258	258	258	258	258
	(ns)	1675	1675	1675	1675	1675	1675	1675	840
Resources	BRAM_18K	0	0	0	0	0	0	0	0
	DSP48E	0	0	0	0	0	0	0	0
	FF	52	52	52	52	52	52	52	85
	LUT	138	138	138	138	138	138	138	144
	URAM	0	0	0	0	0	0	0	0

Рис. 6 Таблица данных для всех решений

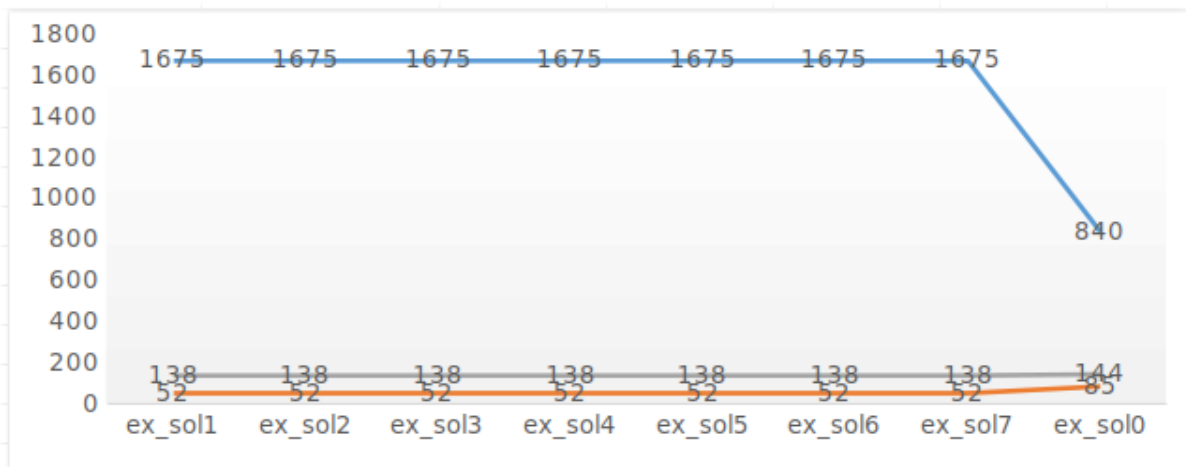


Рис. 7 График данных для всех решений

Как видно из рисунка 7 все решения имеют одинаковое Latency(ns) и используют одинаковое количество аппаратных ресурсов — FF и LUT. Только решение с заданным Timing = 6ns имеет наименьший Estimated и большее количество аппаратных ресурсов 85 FF и 144 LUT Следовательно наилучшего решения для данной функции нет.

6. Исследование №2

6.1 Задание исследования №2

На рисунке 8 представлены данные в файле lab2_1_inc.c для исследования №1 — размер массива 128 машинных слов и тип данных long long.

```
lab2_1.c
1 #define N 128
2
3 typedef long long atype;
```

Рис 8. Содержимое файла lab2_1_inc.c

6.2 Сравнение решений для исследования №2

На рисунке 9 представлено сравнение решений из Vivado HLS GUI по аппаратным ресурсам и временным параметрам.

Performance Estimates									
Timing									
Clock		base	ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7
clk	Target	6.00 ns	8.00 ns	10.00 ns	12.00 ns	14.00 ns	16.00 ns	18.00 ns	20.00 ns
	Estimated	3.825 ns	7.082 ns	7.082 ns	7.082 ns	7.082 ns	7.082 ns	7.082 ns	7.082 ns
Latency									
		base	ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7
Latency (cycles)	min	386	258	258	258	258	258	258	258
	max	386	258	258	258	258	258	258	258
Latency (absolute)	min	2.316 us	2.064 us	2.580 us	3.096 us	3.612 us	4.128 us	4.644 us	5.160 us
	max	2.316 us	2.064 us	2.580 us	3.096 us	3.612 us	4.128 us	4.644 us	5.160 us
Interval (cycles)	min	386	258	258	258	258	258	258	258
	max	386	258	258	258	258	258	258	258
Utilization Estimates									
		base	ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7
BRAM_18K0		0	0	0	0	0	0	0	0
DSP48E		0	0	0	0	0	0	0	0
FF		149	84	84	84	84	84	84	84
LUT		187	181	181	181	181	181	181	181
URAM		0	0	0	0	0	0	0	0

Рис. 9 Сравнение решений исследования №2

6.3 Электронная таблица и график исследования №2

На рисунке 10 представлена таблица с параметрами для все решений. На рисунке 11 представлен график для сравнения всех решений.

		ex_sol1	ex_sol2	ex_sol3	ex_sol4	ex_sol5	ex_sol6	ex_sol7	ex_sol0
Clock	Target (ns)	8	10	12	14	16	18	20	6
	Estimated (ns)	7,08	7,08	7,08	7,08	7,08	7,08	7,08	3,83
Latency	(cycles)	258	258	258	258	258	258	258	258
	(ns)	1827	1827	1827	1827	1827	1827	1827	987
Resources	BRAM_18K	0	0	0	0	0	0	0	0
	DSP48E	0	0	0	0	0	0	0	0
	FF	84	84	84	84	84	84	84	149
	LUT	181	181	181	181	181	181	181	187
	URAM	0	0	0	0	0	0	0	0

Рис. 10 Таблица данных для всех решений

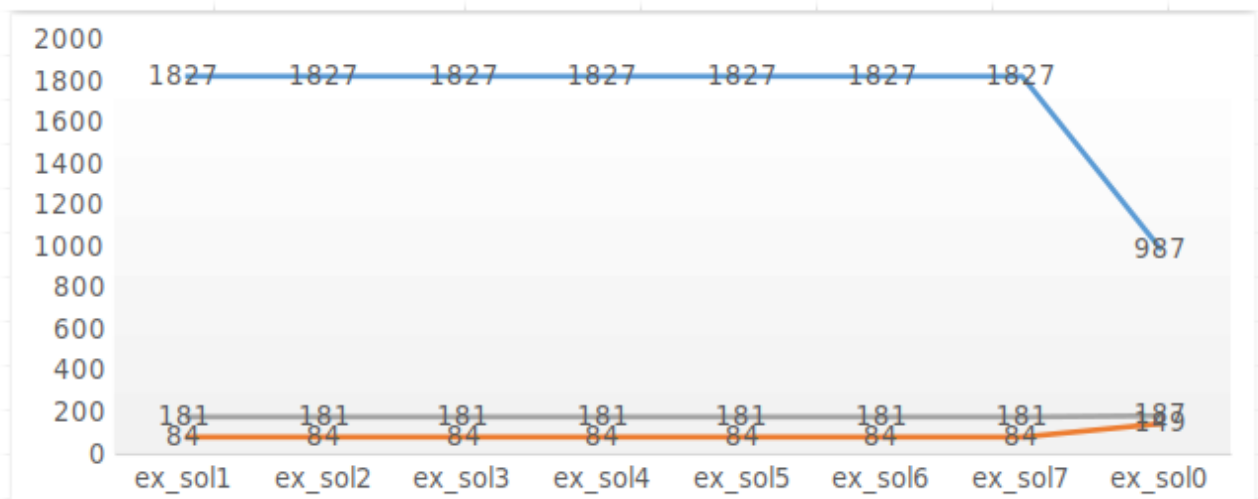


Рис. 11 График данных для всех решений

Как видно из рисунка 11 почти все решения имеют одинаковое Latency(ns) и используют одинаковое количество аппаратных ресурсов — FF и LUT. Только решение с заданным Timing = 6ns имеет наименьший Estimated и большее количество аппаратных ресурсов 149 FF и 187 LUT. Следовательно наилучшего решения для данной функции нет.

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. Синтезируемая функция была протестирована с разными типами данными, такими как int и long long. Было приведено сравнение решений.