Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и технологии

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab3\_z2

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)

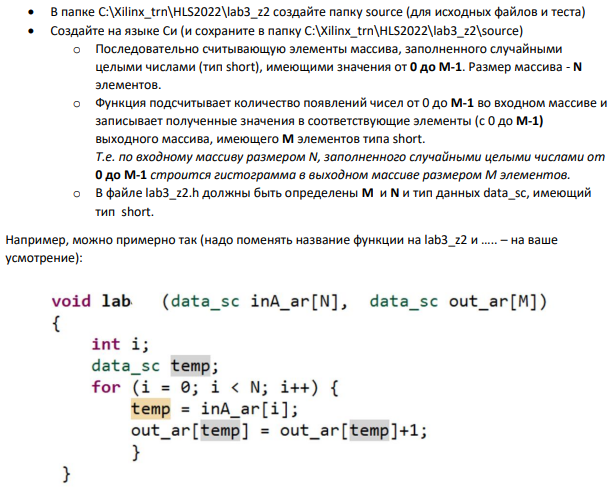
[5.1 Исходные настройки 7](#_Toc116916482)

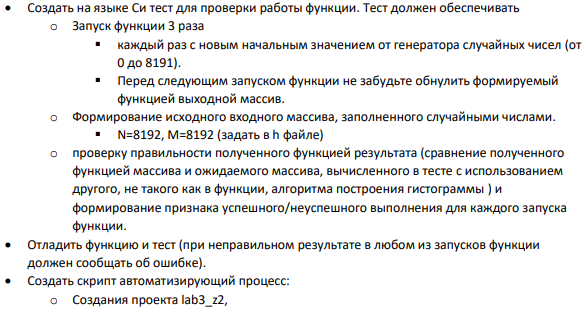
[5.2 Синтез 8](#_Toc116916483)

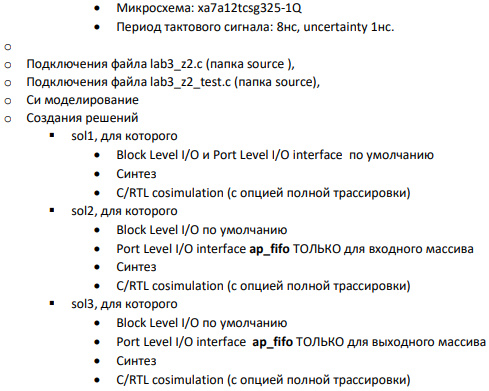
[6. Сравнение Solution1 и Solution2 9](#_Toc116916484)

[7. Выводы 10](#_Toc116916485)

1. Задание







**2. Исходный код функции**

Исходный код заголовочного файла приведен ниже.

|  |
| --- |
| **#include** <stdio.h>  **#define** **N** 8192  **#define** **M** 8192  **typedef** **short** **data\_sc**;  **typedef** **short** **out\_ar**;  **void** **lab3\_z2** **(data\_s**c inA\_ar[N], **data\_s**c out\_ar[M]); |

Исходный код синтезируемой функции приведен ниже.

|  |
| --- |
| **#include** "lab3\_z2.h"  **void** **lab3\_z2**(**data\_sc** inA\_ar[N], **data\_sc** out\_ar[M])  {  **int** i;  **data\_sc** temp;  **for** (i = 0; i < N; i++)  {  temp = inA\_ar[i];  out\_ar[temp] = out\_ar[temp] + 1;  }  } |

**3. Исходный код теста**

Исходный код теста для проверки функции lab3\_z2 приведен ниже.

|  |
| --- |
| **#include** "lab3\_z2.h"  **int** **arentEqual**(**data\_sc** expected\_arr[M], **data\_sc** actual\_arr[M])  {  **for** (**int** i = 0; i < M; i++)  **if** (expected\_arr[i] != actual\_arr[i])  {  **fprintf**(stdout, " ERROR: expected=%d actual=%d \n", expected\_arr[i], actual\_arr[i]);  **return** 1;  }  **return** 0;  }  **int** **main**() {  **data\_sc** in[N];  **data\_sc** out[M], expected\_ar[M];  **int** i, j;  **int** pass = 0;  **for** (i = 0; i < 3; i++)  {  **for** (j = 0; j < N; j++)  in[j] = **rand**() % M;  **for** (j = 0; j < M; j++){  out[j] = 0;  expected\_ar[j] = 0;  }  **lab3\_z2**(in, out);  **for** (j = 0; j < N; j++)  expected\_ar[in[j]]++;  pass = **arentEqual**(out, expected\_ar);  }  **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 lab3\_z2  add\_files ./source/lab3\_z2.c  set\_top lab3\_z2  add\_files -tb ./source/lab3\_z2\_test.c  open\_solution -reset sol1  create\_clock -period 8 -name clk  set\_clock\_uncertainty 1  set\_part {xa7a12tcsg325-1Q}  set\_directive\_interface -mode ap\_fifo lab3\_z2 inA\_ar  csim\_design -clean  csynth\_design  cosim\_design -trace\_level all -tool xsim |

**6. Сравнение производительности**

Было измерено время выполнения у программы из lab3\_z2 для компьютера и разрабатываемого устройства.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #include "lab3\_z2.h"  #include "lab3\_z2.c"  int arentEqual(data\_sc expected\_arr[M], data\_sc actual\_arr[M])  {  for (int i = 0; i < M; i++)  if (expected\_arr[i] != actual\_arr[i])  {  fprintf(stdout, " ERROR: expected=%d actual=%d \n", expected\_arr[i], actual\_arr[i]);  return 1;  }  return 0;  }  int main() {  data\_sc in[N];  data\_sc out[M], expected\_ar[M];  int i, j;  struct timespec t0, t1;  double all\_time = 0.0;  double max\_time = 0.0;  double min\_time = 0.0;  int pass = 0;  for (i = 0; i < 32; i++)  {  for (j = 0; j < N; j++)  in[j] = rand() % M;  for (j = 0; j < M; j++){  out[j] = 0;  expected\_ar[j] = 0;  }  if(clock\_gettime(CLOCK\_REALTIME, &t0) != 0) {  perror("Error in calling clock\_gettime");  exit(EXIT\_FAILURE);  }  lab3\_z2(in, out);  if(clock\_gettime(CLOCK\_REALTIME, &t1) != 0) {  perror("Error in calling clock\_gettime");  exit(EXIT\_FAILURE);  }  double diff = (double)(t1.tv\_sec - t0.tv\_sec) + ((double)(t1.tv\_nsec - t0.tv\_nsec));  all\_time += diff;  if(max\_time < diff || i == 0)  max\_time = diff;  if(min\_time > diff || i == 0)  min\_time = diff;  for (j = 0; j < N; j++)  expected\_ar[in[j]]++;  pass = arentEqual(out, expected\_ar);  }  printf("All time: %.0fns, max time: %.0fns, min time: %.0fns\n", all\_time, max\_time, min\_time);  if (!pass)  fprintf(stdout, "----------Pass!------------ \n");  else  fprintf(stderr, "----------Fail!------------ \n");  return pass;  }; |

Ниже представлены таблица и график с результатами.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 4096 | 8192 | 16384 |
| Clock | Target ( ns) | 8 | 8 | 8 |
| Estimated ( ns) | 6,757 | 6,757 | 6,757 |
| Iteration Interval | (cycles) | 32769 | 16385 | 65537 |
| (ns) | 254189 | 127098 | 508371 |
| Computer | All time (ns) | 215100 | 427100 | 876800 |
|  | Max time (ns) | 7800 | 14400 | 28600 |
|  | Min time (ns) | 6500 | 12900 | 26500 |

*Рис. 1 Диаграмма сравнения производительности*

**7. Выводы**

В данной работе были получен опыт работы с vitis hls.