

### Lab3\_z3

Исследование и сравнительный анализ времени выполнения функции lab3\_z2 на ПК и времени выполнения ее аппаратных реализаций lab3\_z2 (sol2).

Рабочая папка – папка проекта lab3\_z2!

Исследования проводятся для случаев (параметры задаются в lab3\_z2.h файле) :

1.  $M = N = 4096$
2.  $M = N = 8192$
3.  $M = N = 16384$

#### Измерение времени выполнения на ПК

- Используются исходные коды функции lab3\_z2.
- На базе теста lab3\_z2\_test.c следует создать отдельный, модернизированный, тест lab3\_z2\_testSW.c (сохранить в папке C:\Xilinx\_trn\HLS2022\lab3\_z2\source) для проверки времени выполнения функции lab3\_z2 на ПК:
  - добавить в тест операторы измерения времени выполнения синтезируемой функции (например, как-то так: <https://solarianprogrammer.com/2019/04/17/c17-programming-measuring-execution-time-delaying-program/> ).
  - Увеличить количество запусков синтезируемой функции до 32. Для каждого запуска нужно измерить время выполнения (Точность измерения времени – наносекунды);
- Следует осуществить компиляцию модернизированного теста и запускать его как отдельное приложение
- Следует провести измерение времени выполнения синтезируемой функции на Вашем ПК **для каждого** из случаев 1, 2, 3 (см. выше)
  - Необходимо найти и зафиксировать максимальное, минимальное, среднее значения времени выполнения среди 32 запусков.

#### Измерение времени выполнения на аппаратной реализации

- Используются исходные коды функции lab3\_z2.
- Используются исходные коды теста lab3\_z2\_test.c (C:\Xilinx\_trn\HLS2022\lab3\_z2\source)
- Для решения sol2 следует осуществить синтез для случаев 1, 2, 3 (см. выше) и для каждого случая зафиксировать: II, Estimated period, время выполнения = II \* Estimated period

#### Сравнительный анализ

- Составить xls таблицу и построить два графика (по оси X – случаи 1, 2, 3 по У – время выполнения функции на ПК и аппаратного)
- Оформить отчет, который должен включать
  - Задание
  - Раздел с описанием исходного кода функции
  - Раздел с описанием модернизированного теста
    - Следует указать компилятор, используемый для компиляции.
  - Результаты измерения времени выполнения на ПК

- Следует указать: тип процессора, базовую частоту работы, максимальную частоту работы, объем ОЗУ.
- Результаты измерения времени выполнения на аппаратной реализации
- Раздел с анализом результатов
- Выводы

Архив должен включать всю рабочую папку проекта (включая модернизированный тест и **скомпилированные приложения – папка ..\source**), отчет

#### ПРИМЕР КОДА ДЛЯ ИЗМЕРЕНИЯ ВРЕМЕНИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

```

34     struct timespec t0, t1;
35     double acc_time = 0.0;
36
37     for (int i = 0; i < 32; ++i){
38         set_zero(out_ar);
39         for(int j = 0; j < N; j++){
40             inA_ar[j] = rand() % (M - 1);
41         }
42         if(clock_gettime(CLOCK_REALTIME, &t0) != 0) {
43             perror("Error in calling clock_gettime\n");
44             exit(EXIT_FAILURE);
45         }
46         lab3_z1(inA_ar, out_ar);
47         if(clock_gettime(CLOCK_REALTIME, &t1) != 0) {
48             perror("Error in calling clock_gettime\n");
49             exit(EXIT_FAILURE);
50         }
51         double diff_time = (((double)(t1.tv_sec - t0.tv_sec))*1000000000.0) + (double)(t1.tv_nsec - t0.tv_nsec);
52         acc_time += diff_time;
53         double temp_avg_time = acc_time / (i + 1); // take average time
54         printf("Elapsed time: %.41f nanoseconds\n", temp_avg_time);
55
56         pass = cmp_arr(inA_ar, out_ar);
57     }
58 }

```