

Задание lab1_z2:

- Разработать на языке Си описание функции, реализующей следующий алгоритм $outArr[i] = inArr[i] + inA + inB + inC$, где $i=0, 1, \dots (ROWS-1)$. $ROWS=3$
- 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: xa7a12tcs325-1q
 - clock period 6; clock_uncertainty 1 (для решения Solution 1)
 - clock period 10; clock_uncertainty 1 (для решения Solution 2)

Программа работы и комментарии.

1. **Создайте рабочие папки**

- 1.1. Создайте рабочую папку для проекта C:\Xilinx_trn\HLS2022\lab1_z2
- 1.2. Создайте папку для исходных кодов C:\Xilinx_trn\HLS2022\lab1_z2\source

2. **Создайте проект.**

- 2.1. проект lab1_z2 (рабочая папка – папка ...lab1_z2)
- 2.2. Микросхема: xa7a12tcs325-1q
- 2.3. clock period 6; clock_uncertainty 1

3. **Создайте описание функции и теста**

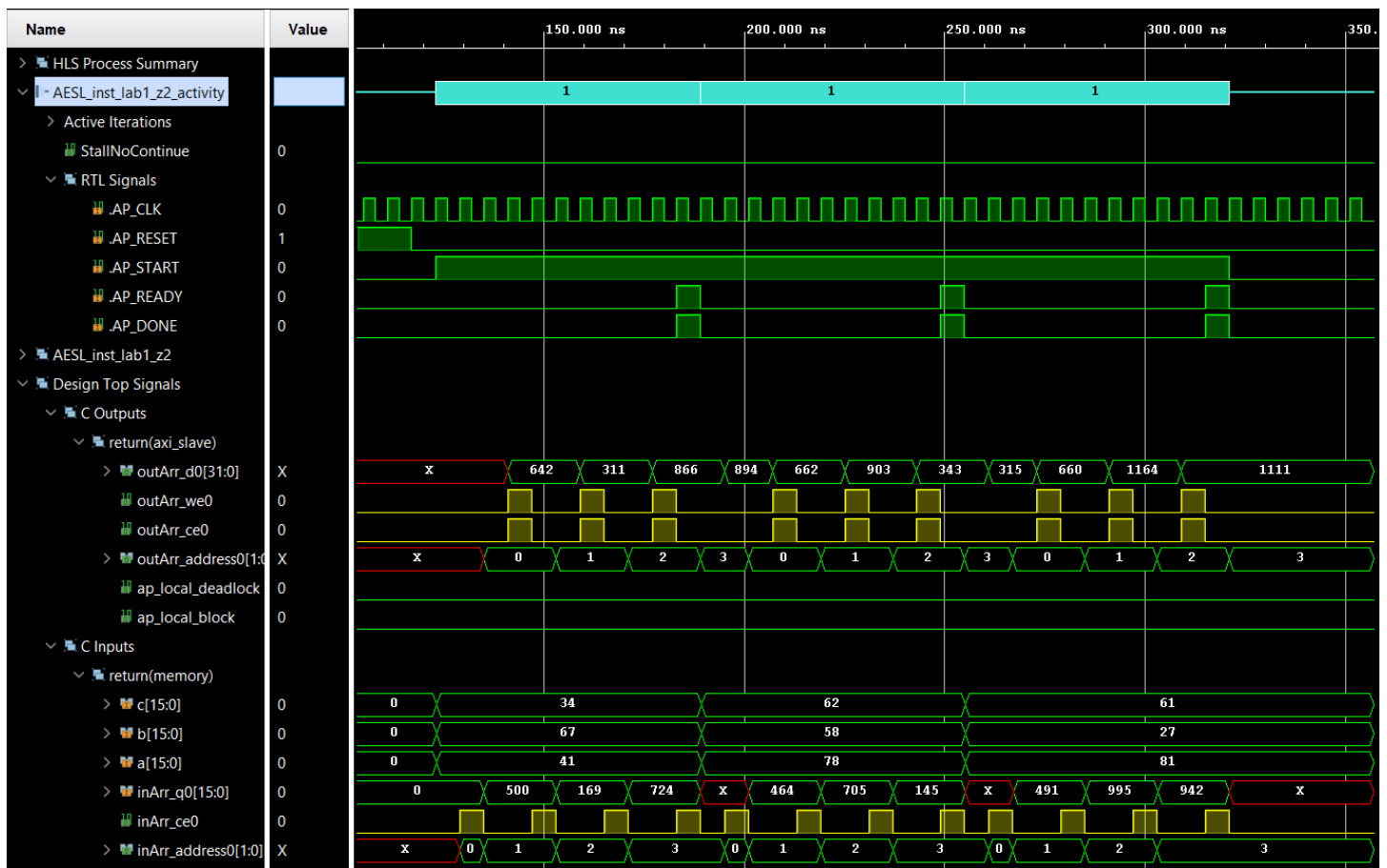
- 3.1. Создайте файл lab1_z2.h (пример файла в приложении) и сохраните в папке C:\Xilinx_trn\HLS2022\lab1_z2\source
- 3.2. Создайте файл с описанием функции - файл lab1_z2.c (пример файла в приложении) и сохраните в папке C:\Xilinx_trn\HLS2022\lab1_z2\source
- 3.3. Создайте файл с описанием теста - файл lab1_z2_test.c (пример файла в приложении) и сохраните в папке C:\Xilinx_trn\HLS2022\lab1_z2\source

4. **Осуществите Си моделирование созданной функции**

- 4.1. Запустите тест и убедитесь в правильности работы созданного описания функции (приведите снимок экрана с результатами моделирования)
- 4.2. Убедитесь в правильности работы созданного теста - убедитесь, что тест отрабатывает ошибку (приведите снимок экрана с результатами моделирования)

5. **Проведите синтез и анализ решения Solution1**

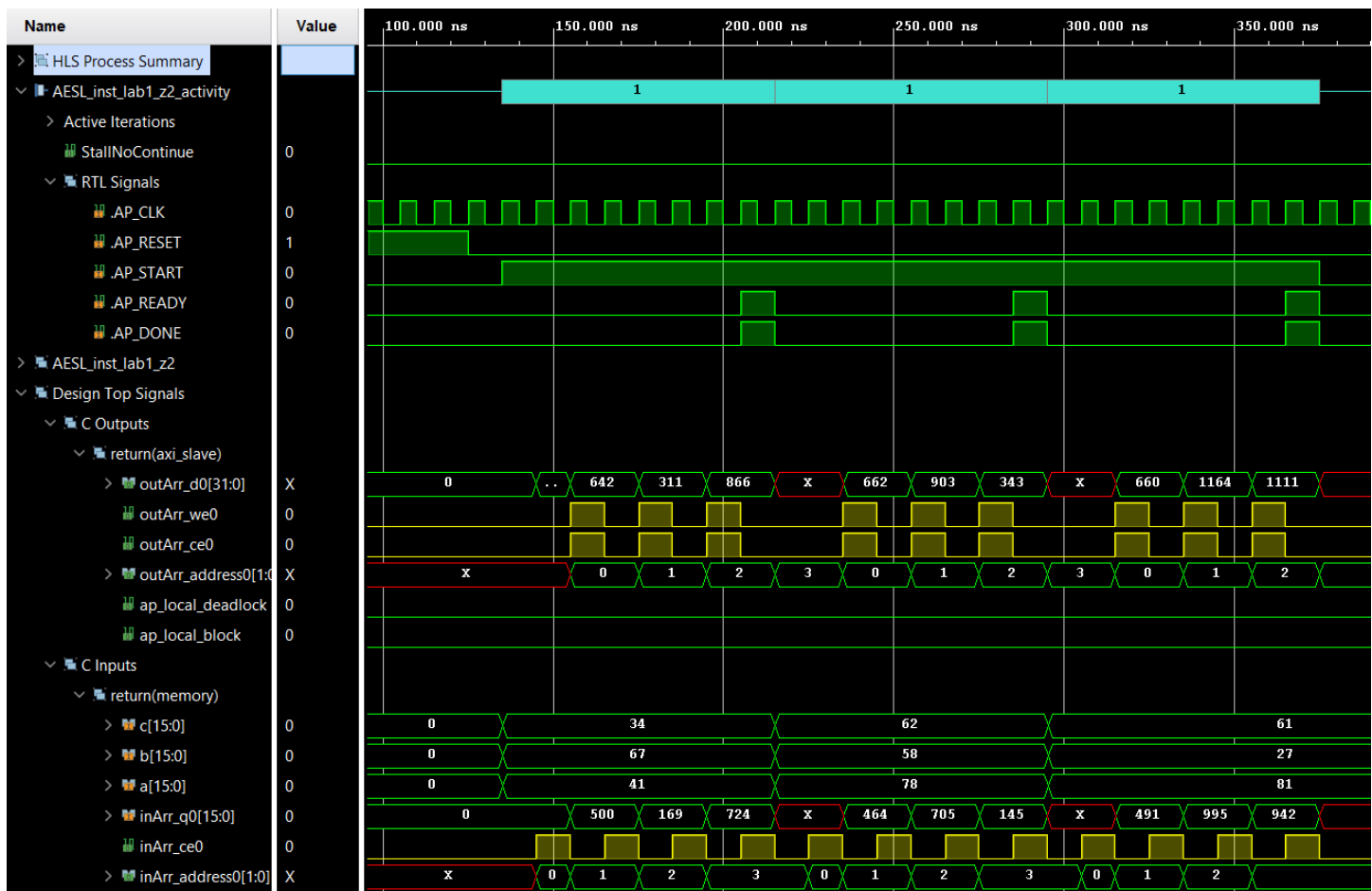
- 5.1. Проведите анализ результатов синтеза
 - 5.1.1. Оценка временных параметров (приведите снимок экрана)
 - 5.1.2. Оценка производительности и аппаратных затрат (приведите снимок экрана)
 - 5.1.3. Окно планировщика (Schedule Viewer) - приведите снимок экрана
- 5.2. Выполните и проведите анализ результатов Co-simulation
 - 5.2.1. Убедитесь в том, что тест прошел успешно (ожидаемые результаты соответствуют полученным при моделировании) – статус Pass (приведите снимок экрана)
 - 5.2.2. Зафиксируйте (приведите снимок экрана) оценки производительности и сравните их с оценками, полученными при синтезе.
 - 5.2.3. Получите и проведите анализ временных диаграмм
 - 5.2.3.1. Разверните временную диаграмму и приведите снимок экрана с полученной временной диаграммой



- 5.2.3.1.1. На скриншоте следует показать Iteration Latency
- 5.2.3.1.2. На скриншоте следует показать trip count
- 5.2.3.1.3. На скриншоте следует показать loop latency
- 5.2.3.1.4. На скриншоте следует показать Latency
- 5.2.3.1.5. На скриншоте следует показать Initiation Interval

6. Создайте, проведите синтез и анализ решения Solution2

- 6.1. Измените период тактового сигнала – сделайте его равным 10нс.
- 6.2. Проведите анализ результатов синтеза
 - 6.2.1. Оценка временных параметров (приведите снимок экрана)
 - 6.2.2. Оценка производительности и аппаратных затрат (приведите снимок экрана)
 - 6.2.3. Окно планировщика (Schedule Viewer) - приведите снимок экрана
- 6.3. Выполните и проведите анализ результатов Co-simulation
 - 6.3.1. Убедитесь в том, что тест прошел успешно (ожидаемые результаты соответствуют полученным при моделировании) – статус Pass (приведите снимок экрана)
 - 6.3.2. Зафиксируйте (приведите снимок экрана) оценки производительности и сравните их с оценками, полученными при синтезе.
 - 6.3.3. Получите и проведите анализ временных диаграмм
 - 6.3.3.1. Разверните временную диаграмму и приведите снимок экрана с полученной временной диаграммой



- 6.3.3.1.1. На скриншоте следует показывать Iteration Latency
- 6.3.3.1.2. На скриншоте следует показывать trip count
- 6.3.3.1.3. На скриншоте следует показывать loop latency
- 6.3.3.1.4. На скриншоте следует показывать Latency
- 6.3.3.1.5. На скриншоте следует показывать Initiation Interval

Вопрос:

Поясните отличия временных диаграмм решений **Solution 1** и **Solution 2**

7. Сравните решения **Solution 1** и **Solution 2** -

7.1. Зафиксируйте результаты сравнения (сделайте снимок экрана)

7.2. Сравните Timing, Latency, Utilization Estimation и поясните отличия

7.3. Вопросы:

7.3.1. Какое из двух решений имеет меньший II

7.3.1.1. В тактах?

7.3.1.2. В ns (число тактов надо умножить на estimated)?

7.3.2. какое из двух решений обеспечивает большее быстродействие (в ns)?

7.3.2.1. Насколько велико отличие?

7.3.3. Какое из двух решений требует меньших аппаратных затрат?

7.3.3.1. Насколько велико отличие?

7.3.4. Какое из двух решений Вам кажется более оптимальным? И почему?

Приложение

lab1_z2.h

```
1  #define ROWS 3
2
3  typedef short  din_type;
4  typedef int    dout_type;
5
6  // Prototype of top level function for C-synthesis
7  void lab1_z2( din_type inArr[ROWS], din_type a, din_type b, din_type c, dout_type outArr[ROWS]);
```

lab1_z2.c

```
1  #include "lab1_z2.h"
2
3  void lab1_z2( din_type inArr[ROWS], din_type a, din_type b, din_type c, dout_type outArr[ROWS])
4  {
5      din_type    x;
6      dout_type   y;
7      for(int i = 0; i < ROWS; i++)
8      {
9          #pragma HLS PIPELINE off
10         x = inArr[i];
11         y = x + a + b + c;
12         outArr[i] = y;
13     }
14 }
```

lab1_z2_test.c

```
1  #include <stdio.h>
2  #include "lab1_z2.h"
3
4  //function for comparing expected and actual results in arrays
5  int arentEqual(dout_type expectedArr[ROWS], dout_type actualArr[ROWS])
6  {
7      // Linearly compare elements
8      for (int i = 0; i < ROWS; i++)
9          if (expectedArr[i] != actualArr[i])
10             {
11                 fprintf(stdout, " ERROR: expected=%d  actual=%d  for ROW: %d  \n",
12                     expectedArr[i], actualArr[i], i);
13                 return 1;
14             }
15     // If all elements were the same.
16     return 0;
17 }
18
19 int main() {
20     din_type inA, inB, inC;
21     din_type inArr[ROWS];
22     dout_type expectedArr[ROWS], actualArr[ROWS];
23
24     int pass = 0;
25     // Calling the function for 3 times and the results comparing
26     for (int i = 0; i < 3; i++)
27     {
28         // initial settings
29         inA = rand() % 100;
30         inB = rand() % 100;
31         inC = rand() % 100;
32         for (int j=0; j<ROWS; j++)
33         {
34             inArr [j]      = rand() % 1000;
35             actualArr[j]   = 0;
36         }
37
38         //function invocation and getting actual results
39         lab1_z2(inArr, inA, inB, inC, actualArr);
40
41         //expected results evaluation
42         for (int i=0; i<ROWS; i++)
43             expectedArr[i] = inArr [i] + inA  + inB  + inC;
44
45         // Compare the actual results against the expected results
46         if (arentEqual(expectedArr, actualArr))
47             pass = 1;
48     }
49
50     if (!pass)
51         fprintf(stdout, "-----Pass!----- \n");
52     else
53         fprintf(stderr, "-----Fail!----- \n");
54
55     return pass;
56 };
```