

- Создать проект lab5_z1
- Микросхема: xa7a12tcs325-1q
- Для всех решений задать: clock period 20 ; clock_uncertainty не задавать (по умолчанию будет 12.5%)

- Создать на языке C++ функцию (N=512, din_type – short, dout_type - short),

```

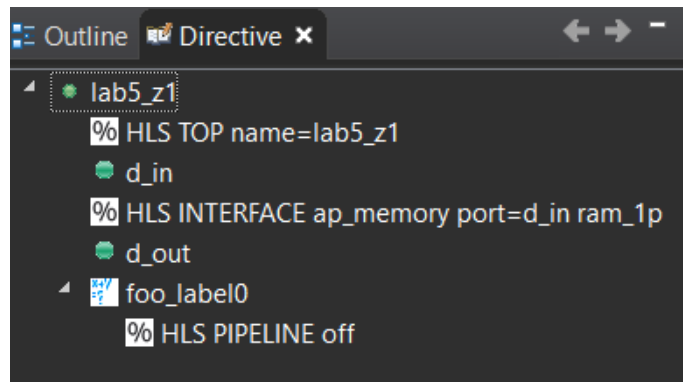
1  #include "lab5_z1.h"
2  void lab5_z1 (din_type d_in[N], dout_type d_out[N/4]){
3      int i;
4      int temp, temp_4, temp_24, temp_34;
5      foo_label0: for( i = 0; i < N/4; i++){
6          temp    = d_in[i];
7          temp_4   = d_in[i+N/4];
8          temp_24  = d_in[i+N/2];
9          temp_34  = d_in[i+3*N/4];
10         d_out[i] = (temp + temp_4) + (temp_24 + temp_34);
11     }
12 }

```

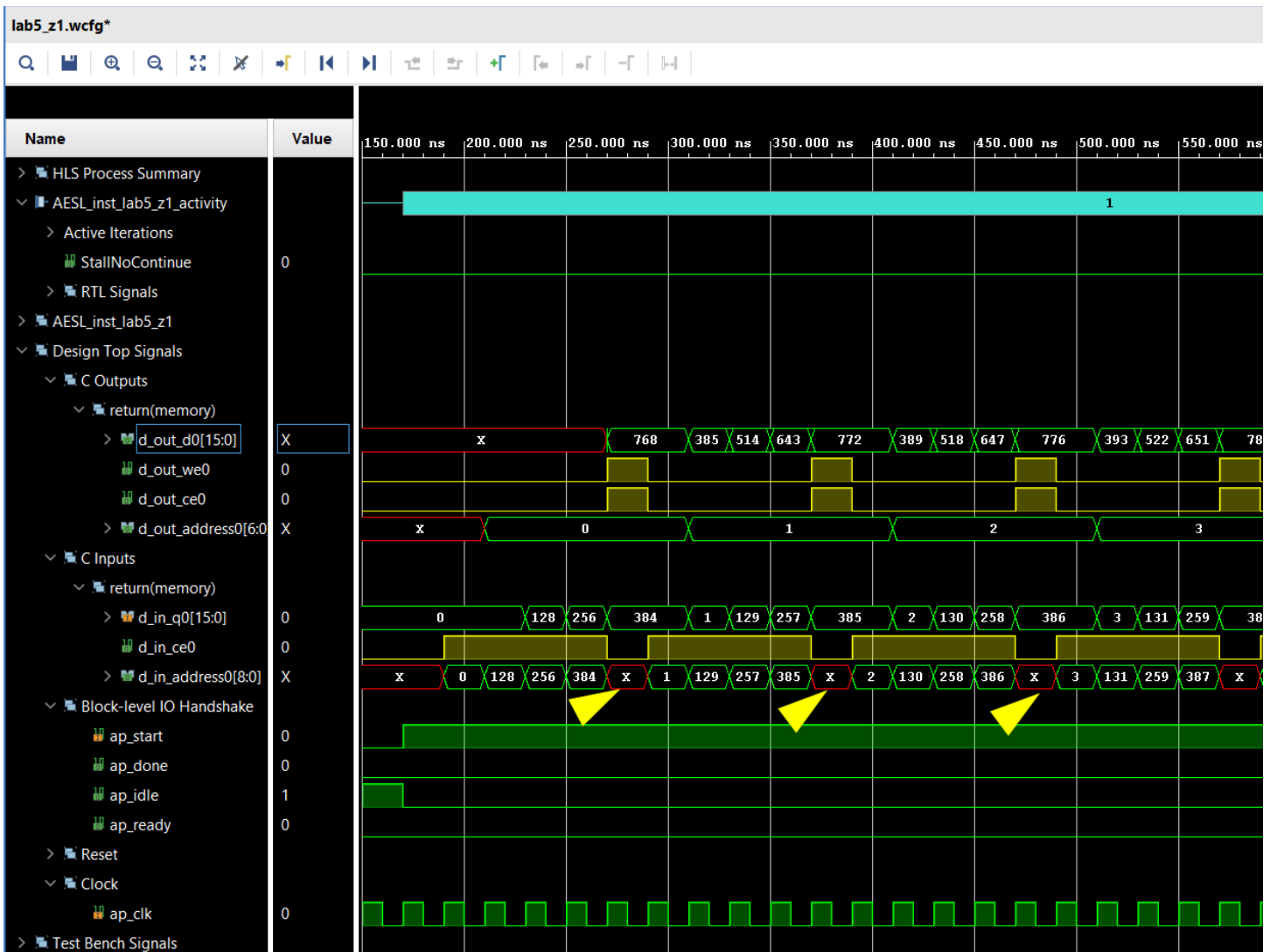
- Создать тест lab5_z1_test.cpp для проверки функции (не менее трех запусков функции) . Осуществить моделирование (с выводом результатов в консоль)

Исследование:

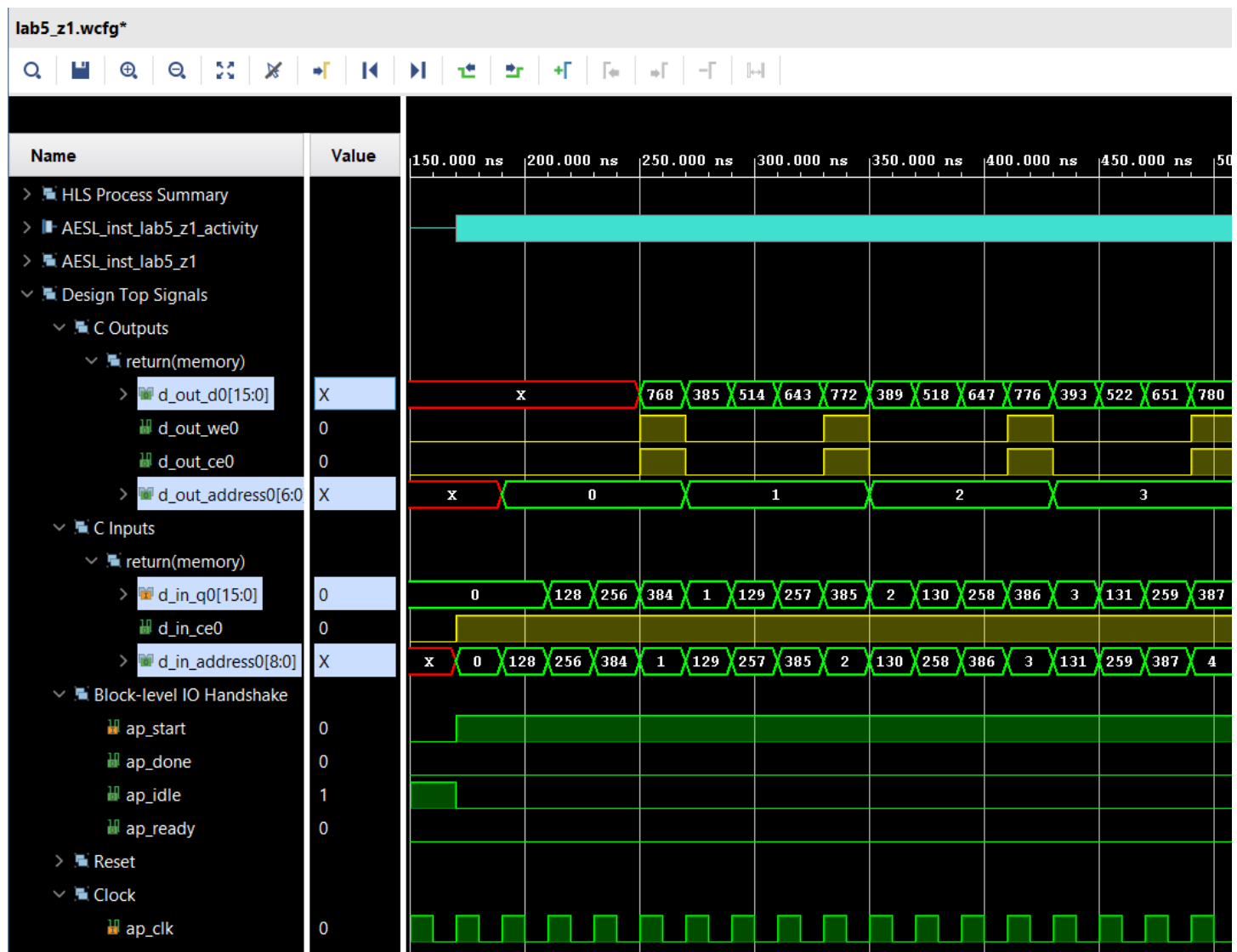
- Solution1
 - установите
 - RAM_1P для входного массива
 - Выключите конвейеризацию для цикла



- осуществите синтез.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить).



Вопрос: Как убрать «пустой» такт (отмечены желтыми стрелками)? Создайте решение solution1_1, в котором не будет этого такта – см. рисунок ниже.



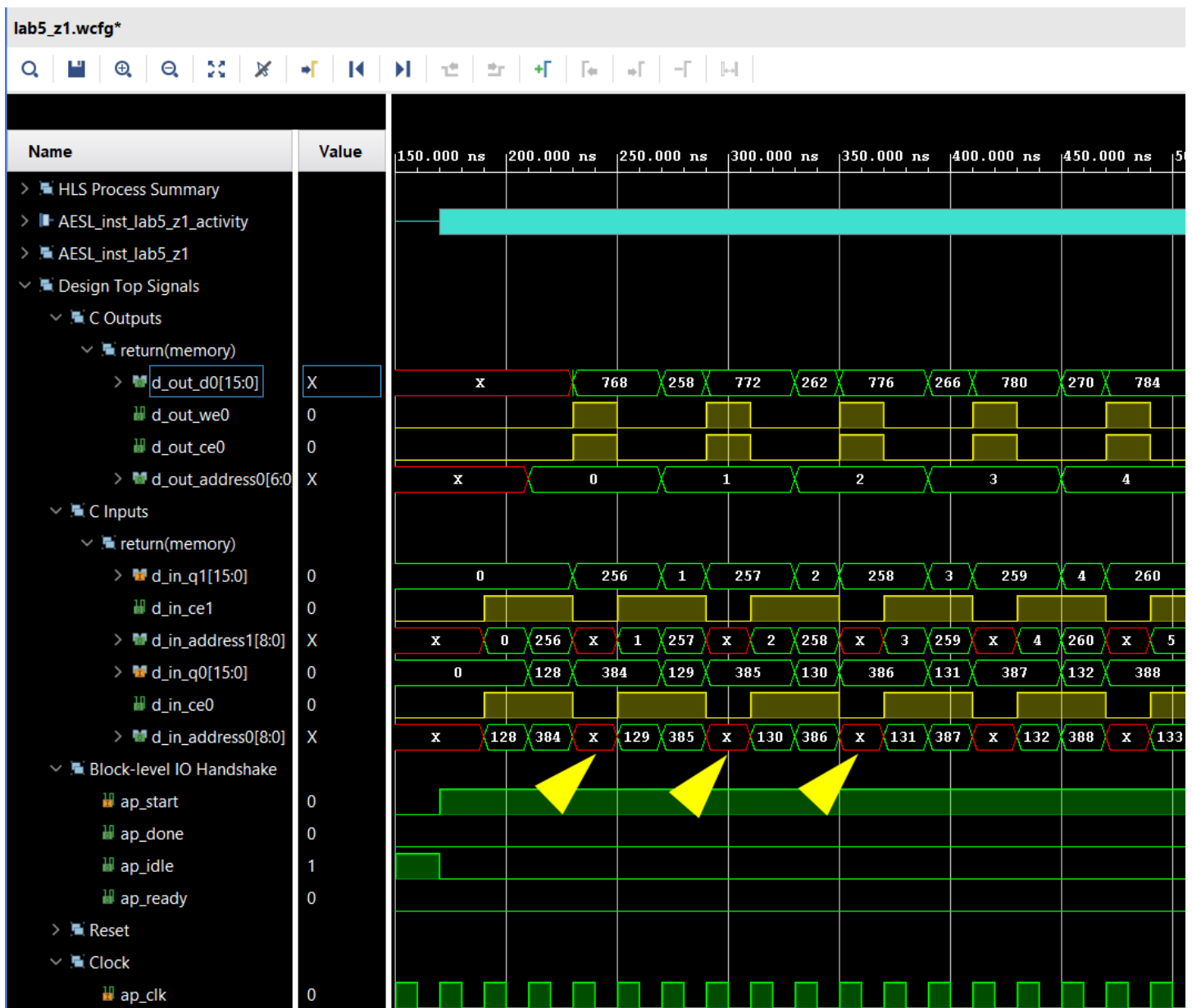
- Solution2
 - установите: RAM_2P для входного массива
 - Выключите конвейеризацию для цикла

```

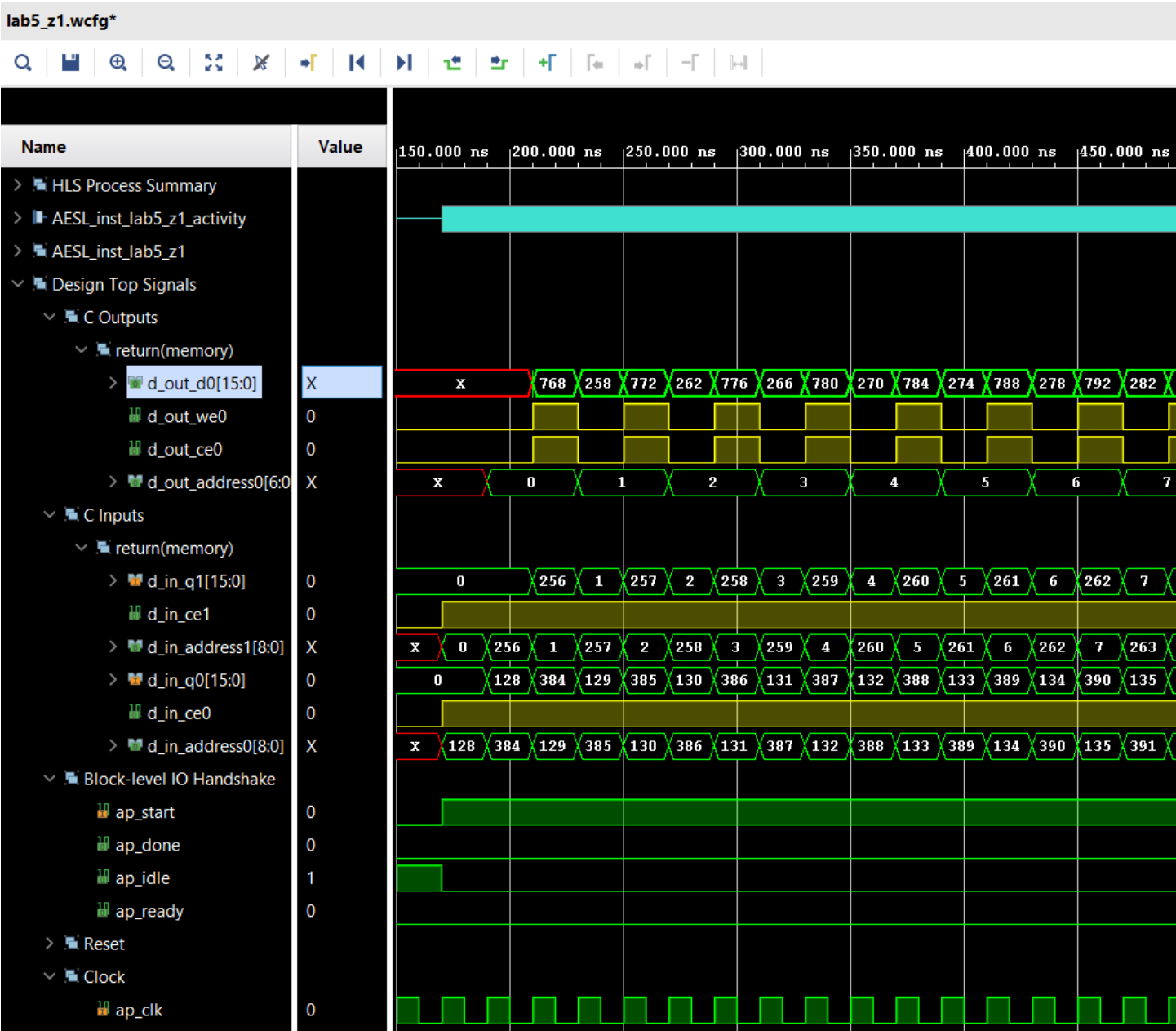
lab5_z1
  % HLS TOP name=lab5_z1
  d_in
  % HLS INTERFACE mode=ap_memory port=d_in storage_type=ram_2p
  d_out
  foo_label0
  % HLS PIPELINE off

```

- осуществите синтез.
- Сравните результаты (schedule viewer, II, аппаратные затраты, реализацию памяти) с Solution1 и поясните отличия.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить).



Вопрос: Как убрать «пустой» такт (отмечены желтыми стрелками)? Создайте решение solution2_1, в котором не будет этого такта – см. рисунок ниже.



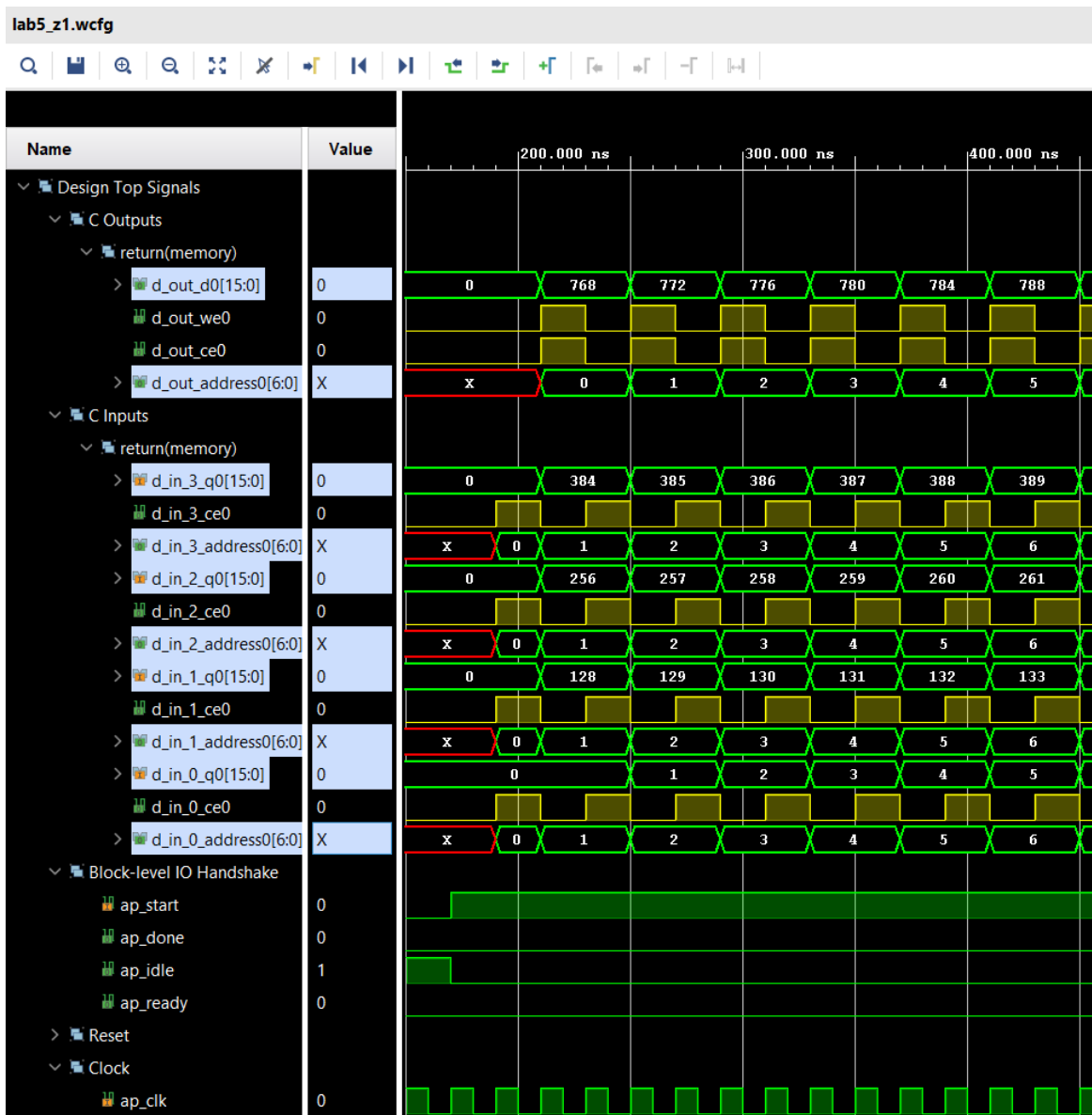
- Solution_3
 - Установите RAM_1P;
 - Установите array_partition block, factor =4 для входного массива
 - Выключите конвейеризацию для цикла

```

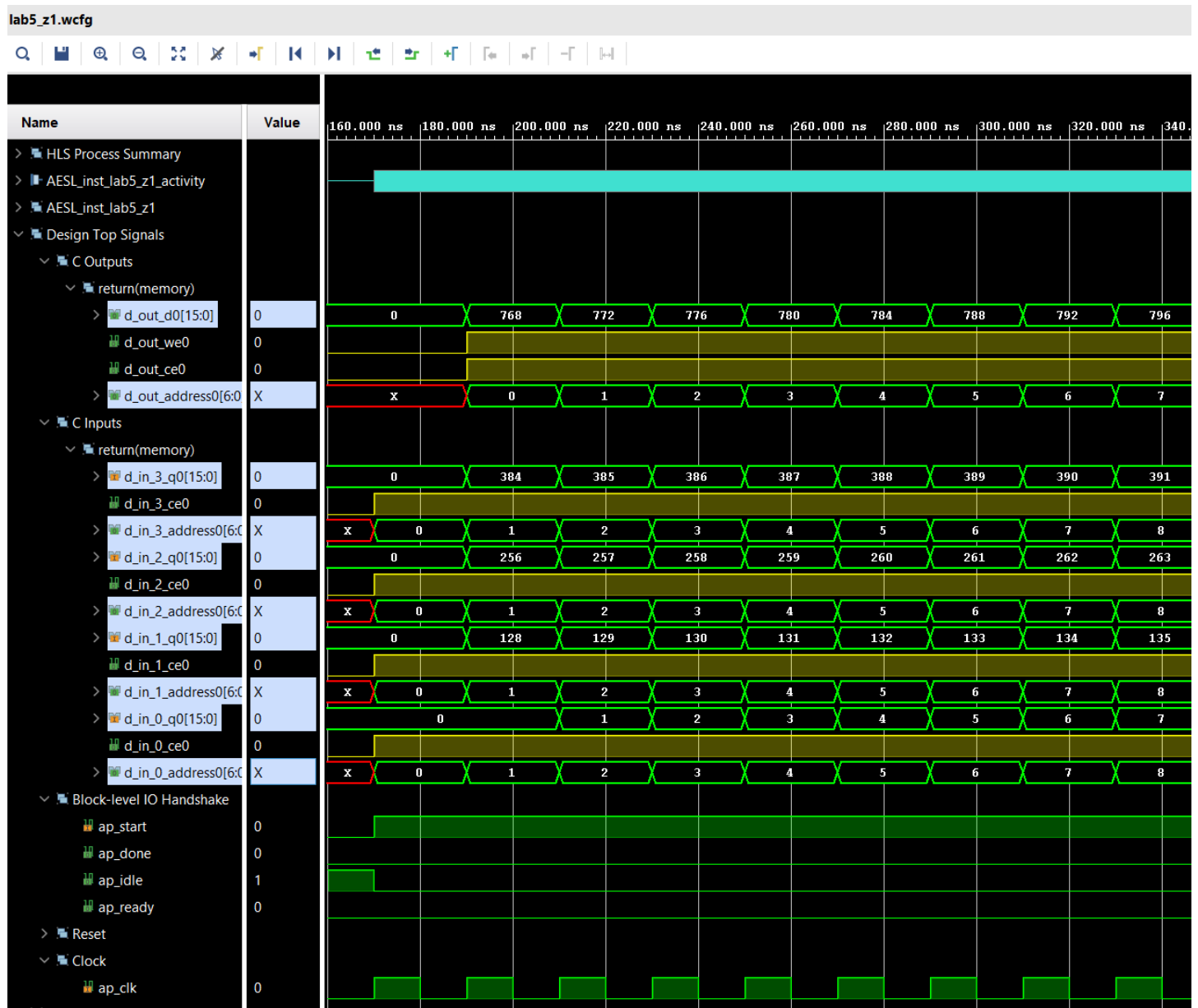
lab5_z1
  % HLS TOP name=lab5_z1
  d_in
  % HLS ARRAY_PARTITION block dim=1 factor=4 variable=d_in
  % HLS INTERFACE ap_memory port=d_in ram_1p
  d_out
  foo_label0
    % HLS PIPELINE off

```

- осуществите синтез.
- Сравните результаты (schedule viewer, II, аппаратные затраты, реализацию памяти) с Solution1, 2 и поясните отличия.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить).



Вопрос: Как сделать так, чтобы данные считывались на каждом такте и результат формировался на каждом такте (см. рисунок ниже)? Создайте решение solution3_1, в котором будет реализована приведенная ниже временная диаграмма.



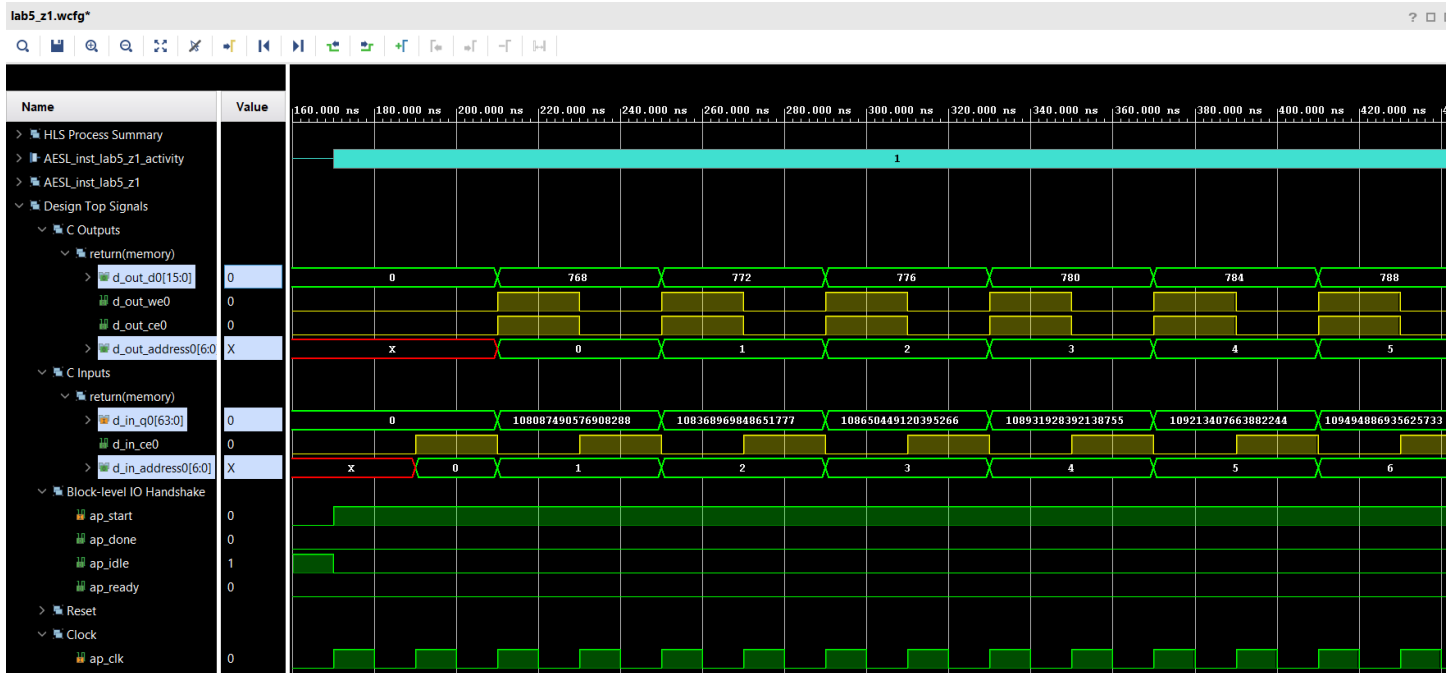
- Solution_4
 - Установите RAM_1P;
 - Установите array_reshape, block, factor =4 для входного массива
 - Выключите конвейеризацию для цикла

```

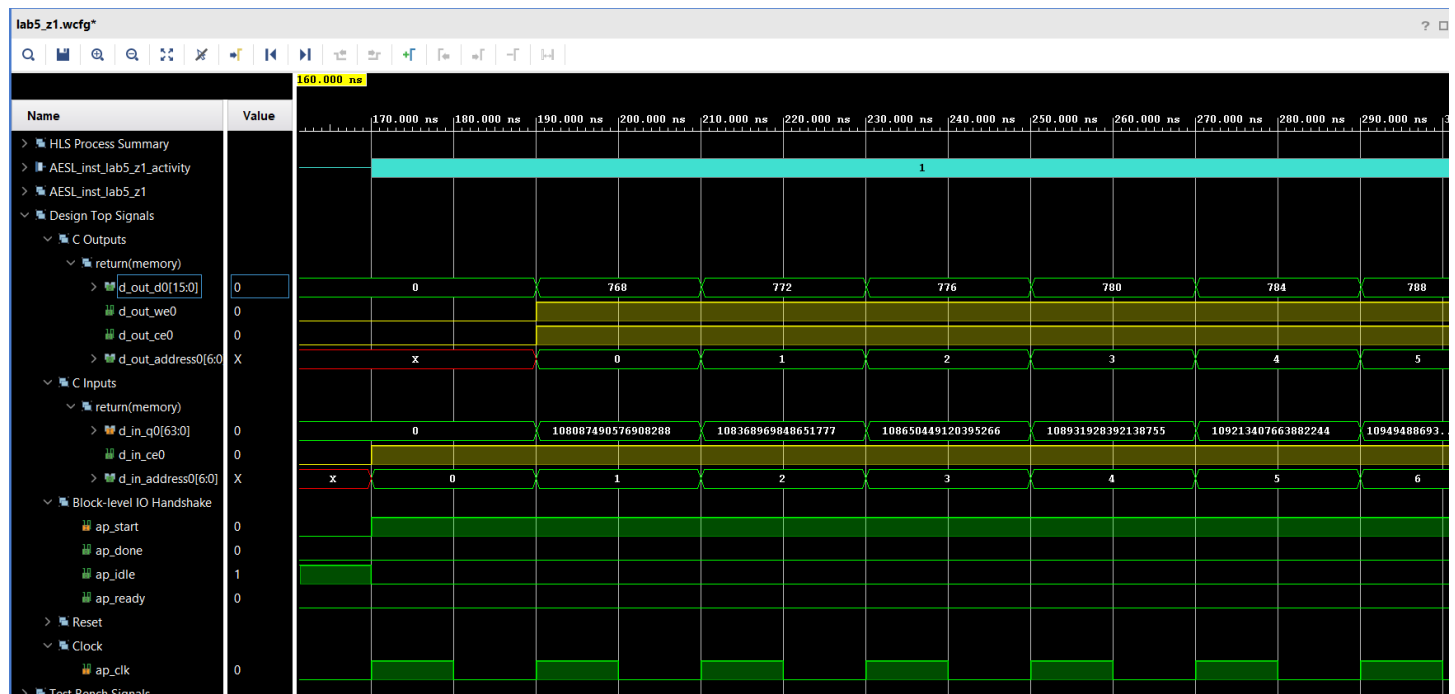
Outline Directive X
lab5_z1
  % HLS TOP name=lab5_z1
  d_in
  % HLS INTERFACE ap_memory port=d_in ram_1p
  % HLS ARRAY_RESHAPE dim=1 factor=4 type=block variable=d_in
  d_out
  foo_label0
    % HLS PIPELINE off

```

- осуществите синтез.
- Сравните результаты (schedule viewer, II, аппаратные затраты, реализацию памяти) с Solution1, 2, 3 и поясните отличия.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить) и сравнение с решением solution3.



Вопрос: Как сделать так, чтобы данные считывались на каждом такте и результат формировался на каждом такте (см. рисунок ниже)? Создайте решение solution4_1, в котором будет реализована приведенная ниже временная диаграмма.



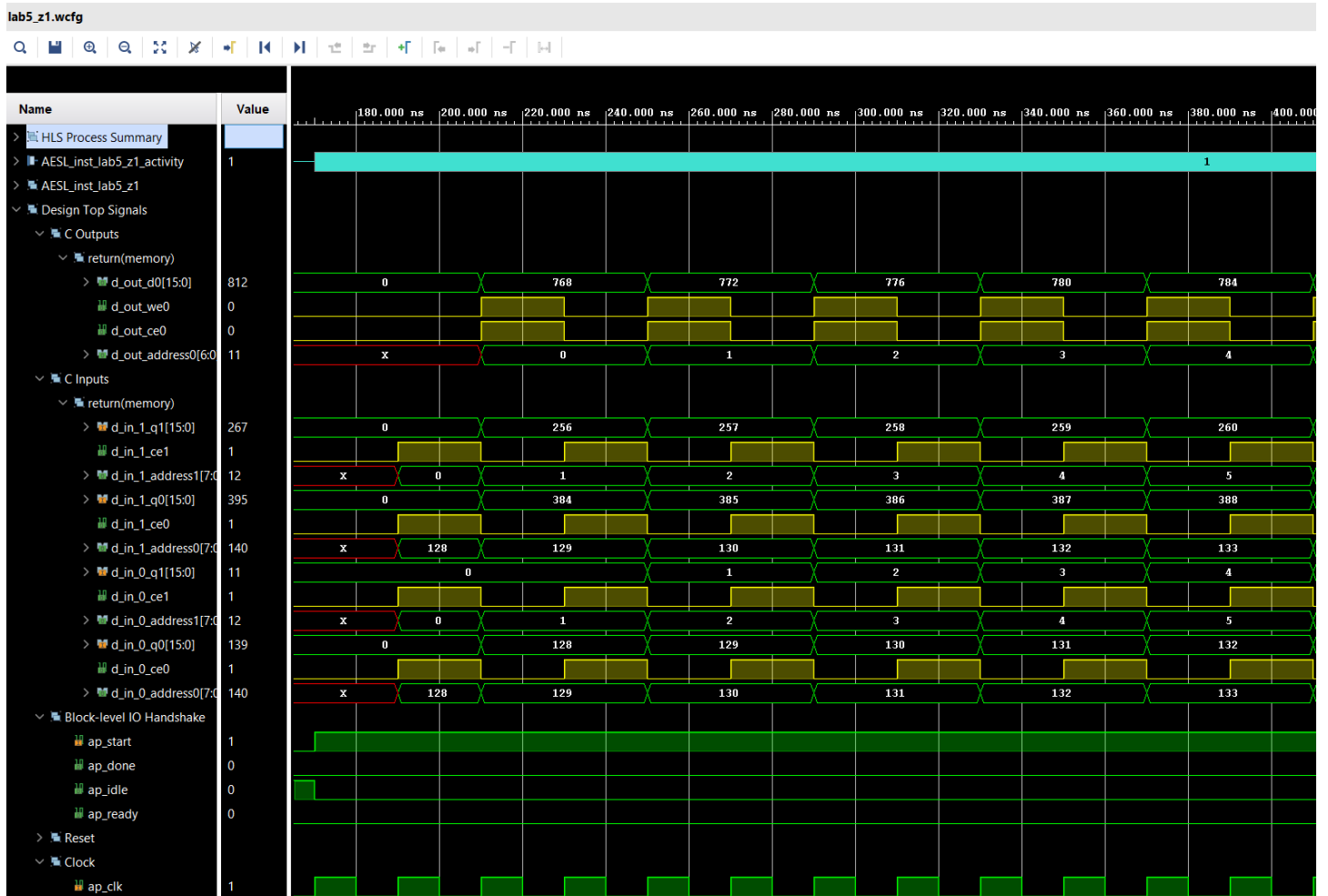
- Solution_5
 - Установите RAM_2P; block,
 - Установите array_partition block factor =2 для входного массива
 - Выключите конвейеризацию для цикла

```

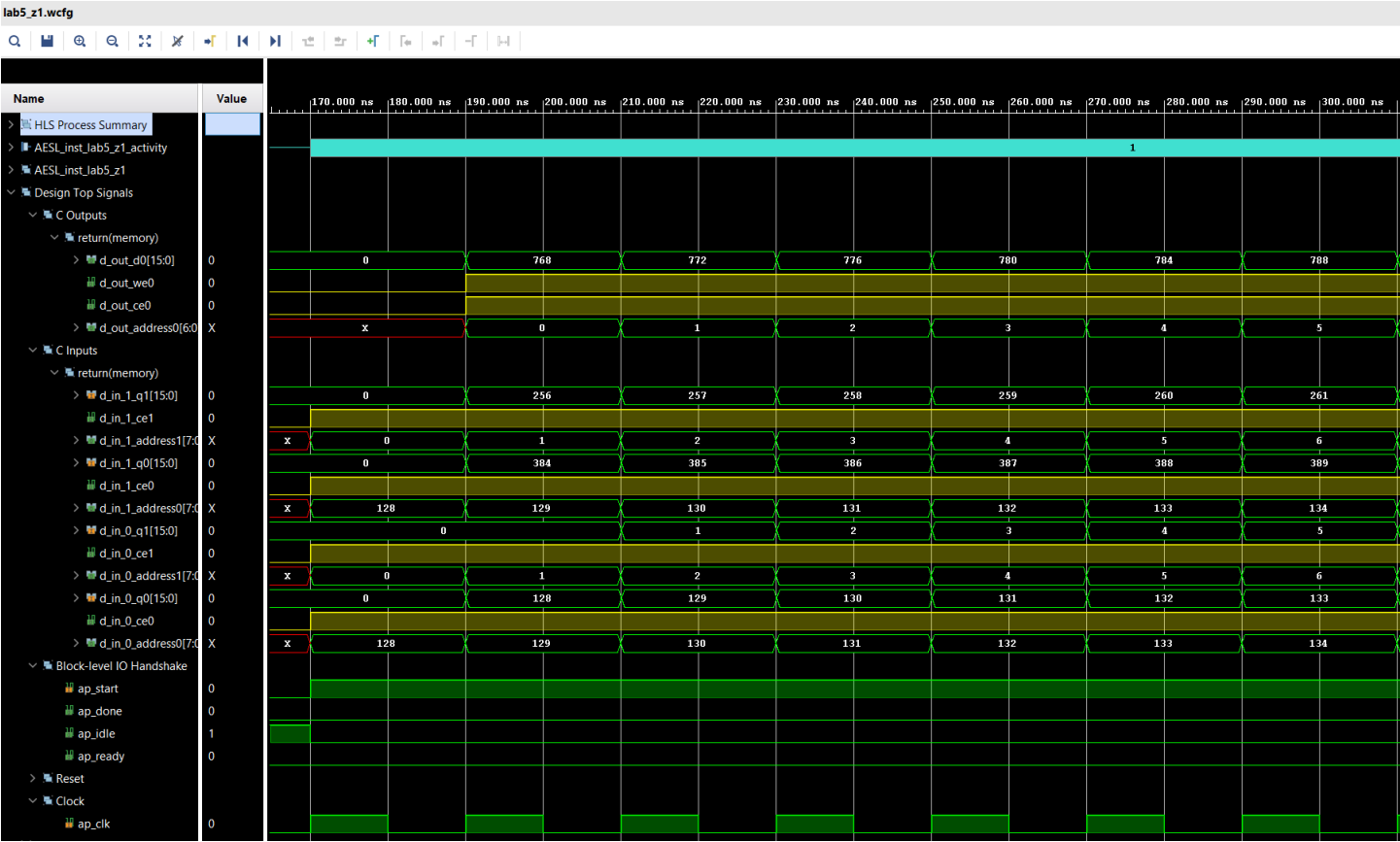
Outline Directive X
└─ lab5_z1
    ├── %HLS TOP name=lab5_z1
    ├── d_in
    ├── %HLS INTERFACE mode=ap_memory port=d_in storage_type=ram_2p
    ├── %HLS ARRAY_PARTITION dim=1 factor=2 type=block variable=d_in
    ├── d_out
    └─ foo_label0
        ├── %HLS PIPELINE off

```

- осуществите синтез.
- Сравните результаты (schedule viewer, II, аппаратные затраты, реализацию памяти) с Solution3 , поясните отличия.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить) и сравнение с решением solution3.



Вопрос: Как сделать так, чтобы данные считывались на каждом такте и результат формировался на каждом такте (см. рисунок ниже)? Создайте решение solution5_1, в котором будет реализована приведенная ниже временная диаграмма.



Вопрос: если увеличить block до 4 изменится ли производительность? Объясните почему (при необходимости сделайте solution5_2).

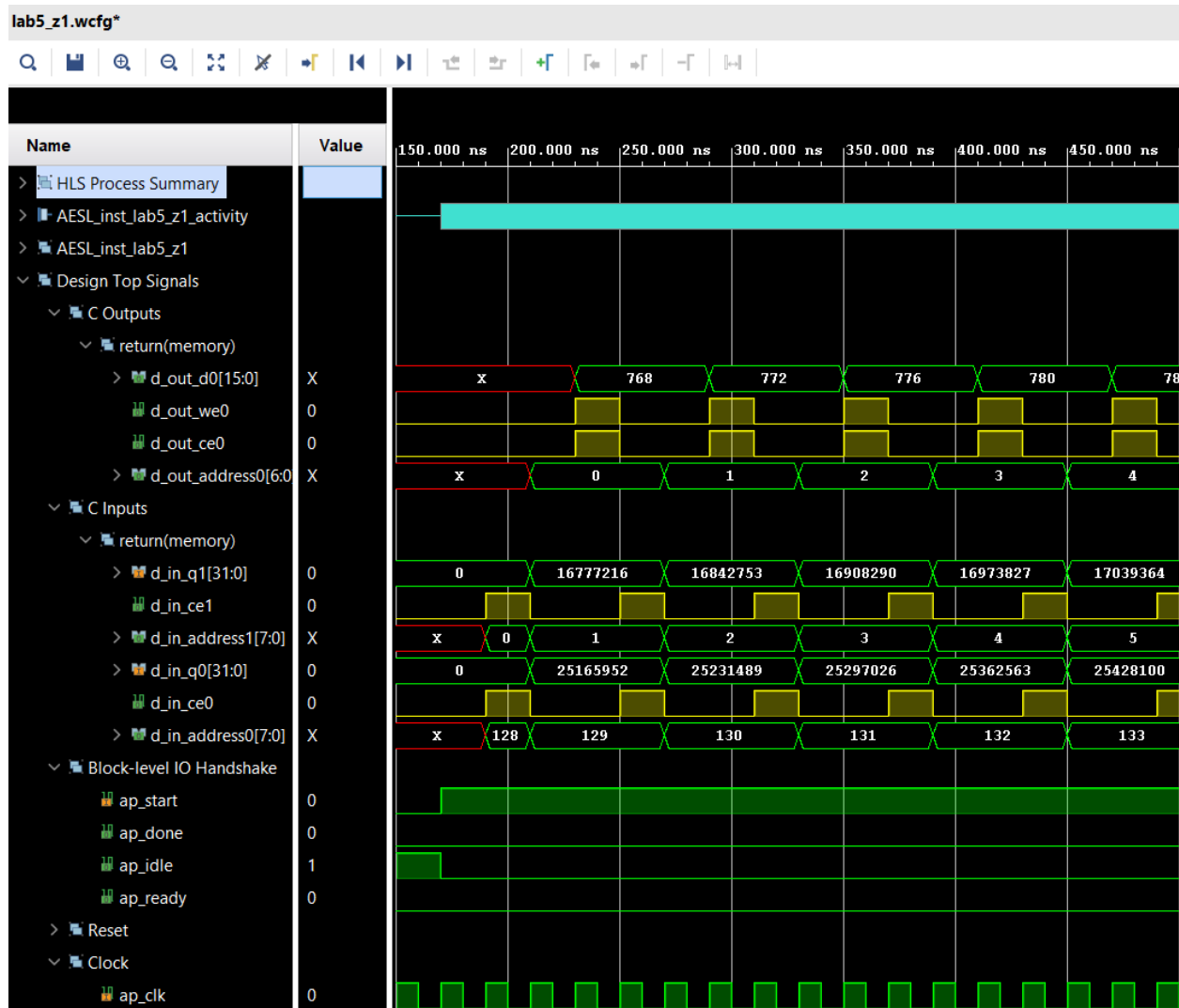
- Solution6
 - Установите RAM_2P;
 - Установите array_reshape block factor =2 для входного массива
 - Выключите конвейеризацию для цикла

```

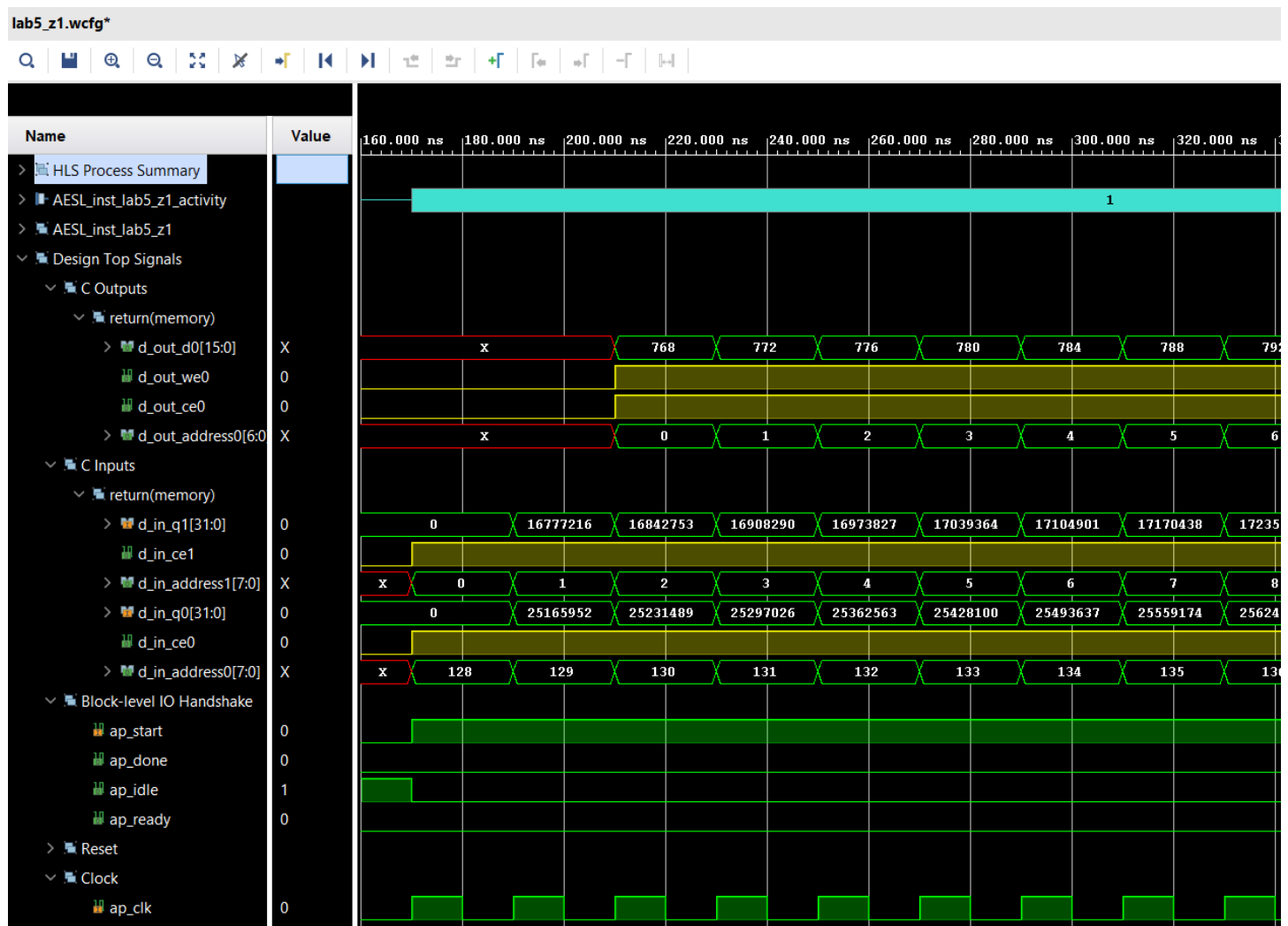
lab5_z1
  % HLS TOP name=lab5_z1
  d_in
  % HLS INTERFACE ap_memory port=d_in ram_2p
  % HLS ARRAY_RESHAPE dim=1 factor=2 type=block variable=d_in
  d_out
  foo_label0
  % HLS PIPELINE off

```

- осуществите синтез.
- Сравните результаты (schedule viewer, II, аппаратные затраты, реализацию памяти) с Solution2,4.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить) и сравнение с решением Solution2, 4.



Вопрос: Как сделать так, чтобы данные считывались на каждом такте и результат формировался на каждом такте (см. рисунок ниже)? Создайте решение solution6_1, в котором будет реализована приведенная ниже временная диаграмма.



Вопрос: если увеличить block до 4 изменится ли производительность? Объясните почему (при необходимости сделайте solution4_2).

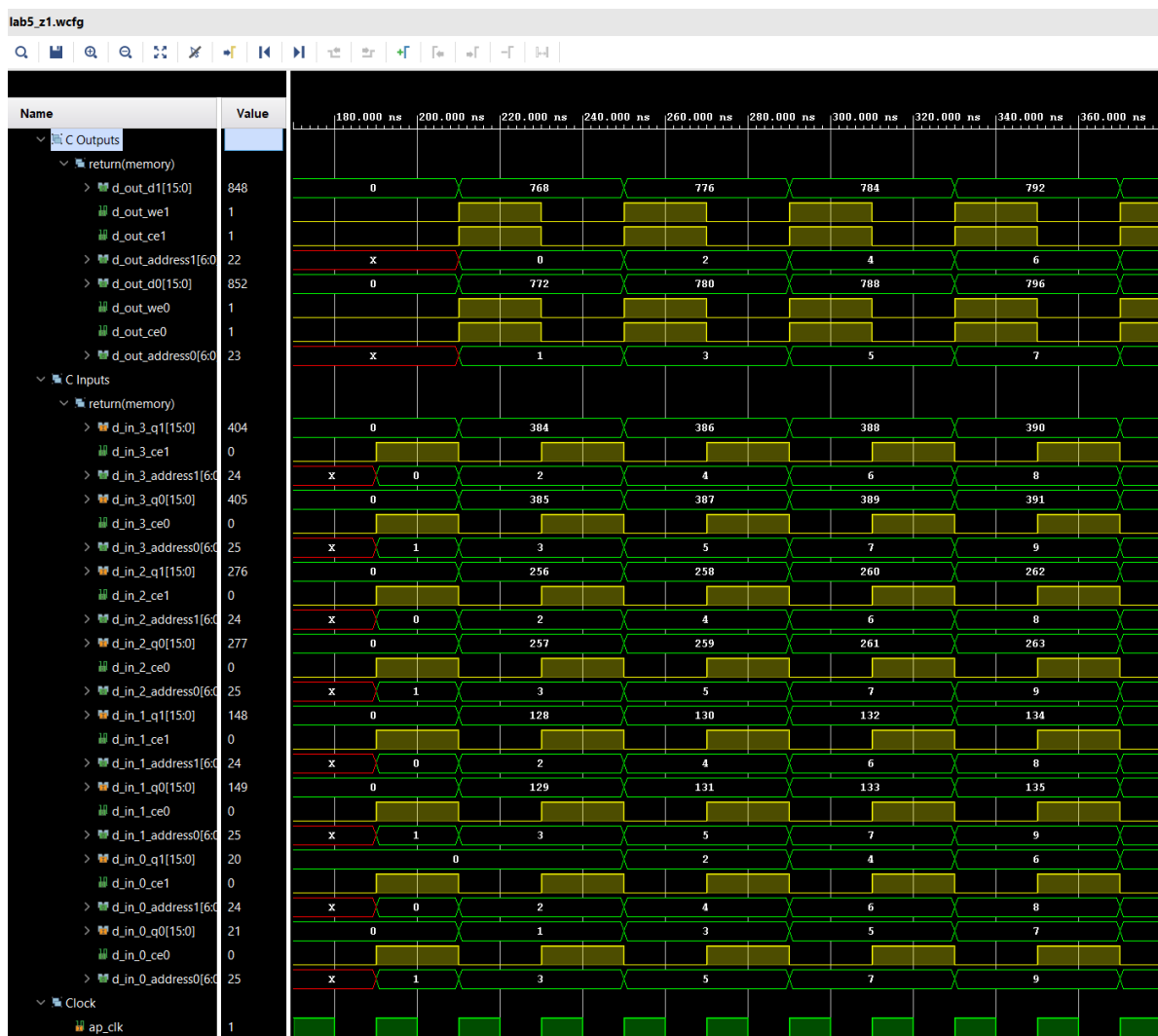
- Solution7

- Установите RAM_2P;
- Установите array_partition, block factor =4 для входного массива
- Установите unroll, factor=2 для цикла.
- Выключите конвейеризацию для цикла

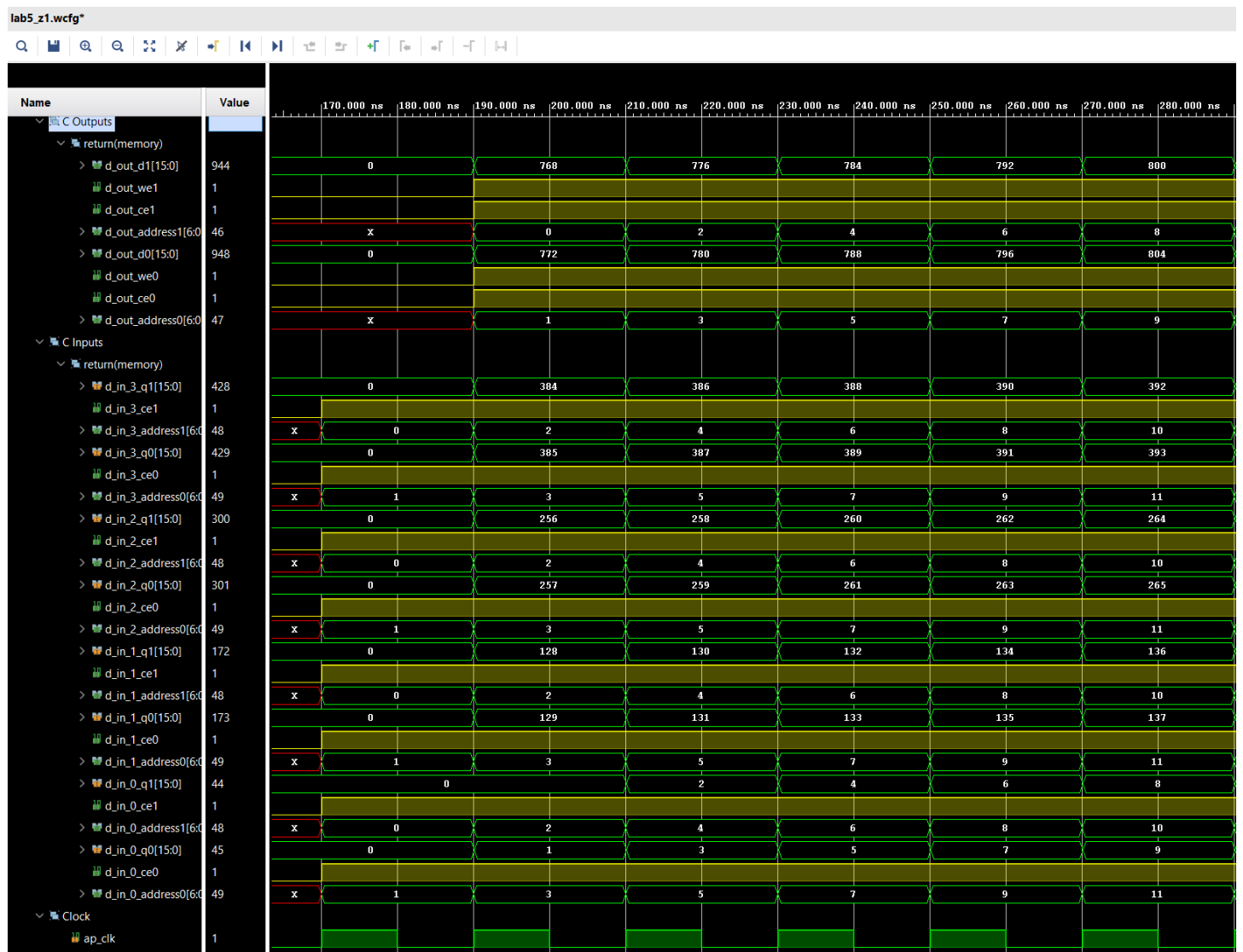
```

lab5_z1
% HLS TOP name=lab5_z1
d_in
% HLS INTERFACE ap_memory port=d_in ram_2p
% HLS ARRAY_PARTITION dim=1 factor=4 type=block variable=d_in
d_out
foo_label0
% HLS UNROLL factor=2
% HLS PIPELINE off
  
```

- осуществите синтез.
- Сравните результаты (schedule viewer, II, аппаратные затраты, реализацию памяти) с Solution4.
- Осуществите cosim. Приведите временную диаграмму (диаграмма должна быть похожа на приведенную ниже). Приведите ее анализ (что бы быть готовым ее пояснить) и сравнение с решением solution4.



Вопрос: Как сделать так, чтобы данные считывались на каждом такте и результат формировался на каждом такте (см. рисунок ниже)? Создайте решение solution7_1, в котором будет реализована приведенная ниже временная диаграмма.



- Сравните все решения между собой Solution1, 1_1, 2, 2_1, 7, 7_1
- Занесите данные в xls файл и постройте зависимости.
- **Выберите лучшее на Ваш взгляд решение.** Объясните выбор.

Измерение времени выполнения на ПК

- Используются исходные коды функции lab5_z1.cpp (**лучшее по быстродействию решение**)
- На базе теста lab5_z1_test.cpp **следует** создать отдельный, модернизированный, тест lab5_z1_testSW.cpp (сохранить в папке C:\Xilinx_trn\HLS2023\lab5_z1\source) для проверки времени выполнения функции lab5_z1 на ПК
- 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
 - N = 4096
 - N = 8192
 - N = 16384
- среди 32 запусков необходимо найти и зафиксировать максимальное, минимальное значения времени выполнения и медиану.

Измерение времени выполнения на аппаратной реализации

- Используются исходные коды функции lab5_z1.cpp (**лучшее по быстродействию решение**)
- **следует** осуществить синтез для случаев
 - N = 4096
 - N = 8192
 - N = 16384

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
 - N = 4096
 - N = 8192
 - N = 16384
 - по Y – время выполнения функции на ПК и аппаратной реализации
- Оформить отчет, который должен включать
 - Задание
 - Раздел с описанием исходного кода функции
 - Раздел с описанием теста
 - Раздел с описанием созданного командного файла
 - Раздел с описанием результатов сравнения решений (со снимками экрана)
 - Раздел с анализом результатов
 - Анализ и выбор оптимального (критерий максимальная производительность) решения
 - 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