Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчет по лабораторной работе №7\_3**

**Курс: «Проектирование реконфигурируемых гибридных вычислительных систем»**

**Тема: Pipeline**

Выполнил студент гр. 3540901/81501 Селиверстов Я.А.

(подпись)

Руководитель Антонов А.П.

(подпись)

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Санкт – Петербург

2019

ОГЛАВЛЕНИЕ

[**1. Задание** 3](#_Toc27952822)

[1.2. Исходный код 4](#_Toc27952823)

[**2. Решение №1.** 5](#_Toc27952824)

[2.1 Моделирование 5](#_Toc27952825)

[2.2 Синтез 5](#_Toc27952826)

[2.3 C|RTL моделирование 7](#_Toc27952827)

[**3. Решение №2** 8](#_Toc27952828)

[3.1 Параметры второго решения 8](#_Toc27952829)

[3.2 Синтез 8](#_Toc27952830)

[3.3 C|RTL моделирование 10](#_Toc27952831)

[**4. Решение № 3.** 11](#_Toc27952832)

[4.1 Параметры третьего решения 11](#_Toc27952833)

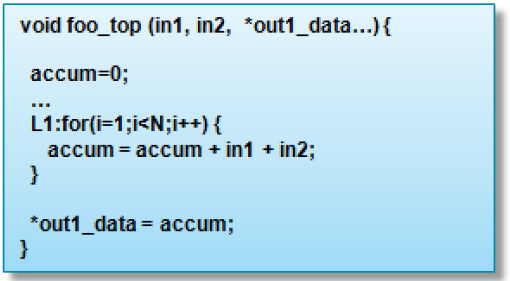
[4.2 Синтез 11](#_Toc27952834)

[4.3 C|RTL моделирование 13](#_Toc27952835)

[**5. Вывод** 14](#_Toc27952836)

# **1. Задание**

* Создать проект lab7\_3
* Микросхема: xa7a12tcsg325-1q
* Создать функцию на основе приведенного ниже слайда.



* Создать тест lab7\_3\_test.c для проверки функций выше.
  + осуществить моделирование (с выводом результатов в консоль)
* Сделать свой solution (для варианта без конвейеризации, с конвейеризацией, с конвейеризацией и rewind)
  + задать: clock period 10; clock\_uncertainty 0.1
  + осуществить синтез
    - привести в отчете:
      * performance estimates=>summary
      * utilization estimates=>summary
      * scheduler viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
      * resource viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
  + Осуществить C|RTL моделирование (для каждого варианта задания директивы)
    - Привести результаты из консоли
    - Открыть временную диаграмму (все сигналы)
      * Отобразить два цикла обработки на одном экране
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
* Выводы
  + Привести обобщенную таблицу зависимости utilization и performance от каждого варианта: без конвейеризации, с конвейеризацией, с конвейеризацией и rewind.
  + Объяснить отличие процедур обращения к элементам массива для каждого случая

## 1.2. Исходный код

Представим код программ lab7\_z3.c и lab7\_z3\_test.c на листинге 1 и 2.

**void** **foo** (**int** in1, **int** in2, **int** \*out\_data) {

**int** i;

**static int** accum = 0;

L1: **for**(i = 0; i < 10; i++) {

accum = accum + in1 + in2;

}

\*out\_data = accum;

}

Листинг 1.

**#include** <stdio.h>

**int** **main**() {

**int** in1 = 4;

**int** in2 = 6;

**int** out = 0; **int** \*o\_p = &out;

**int** exp\_out = 100;

foo(in1,in2,o\_p);

**printf**("Out %d == Exp %d\n", out, exp\_out);

**if** (out != exp\_out) {

**fprintf**(stdout, "-------ERROR-------\n");

**return** -1;

} **else** {

**fprintf**(stdout, "-------Test Pass-------\n");

**return** 0;

}

}

Листинг 2.

# **2. Решение №1.**

## 2.1 Моделирование

На рисунке 2.1. приведем результаты логи успешного моделировании.

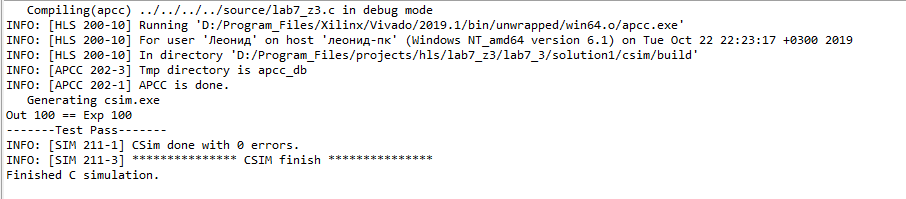


Рисунок 2.1. Логи моделирования

## 2.2 Синтез

Результаты синтеза с оценкой производительности и используемых ресурсов представлены на рисунках 2.2 и 2.3 соответственно

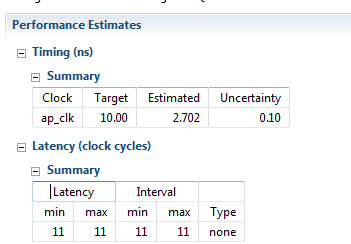


Рисунок 2.2. Performance estimates – summary

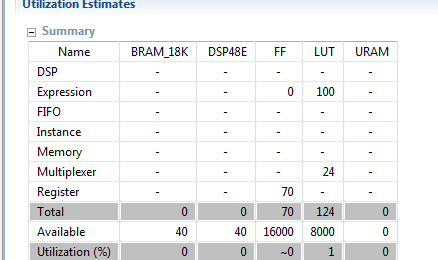


Рисунок 2.3. Utilization estimates – summary

Диаграмма операционного расписания с указанием Latency и диаграмма оперционного просмоторщика ресурсов приведены на рисунках 2.4. и 2.5.

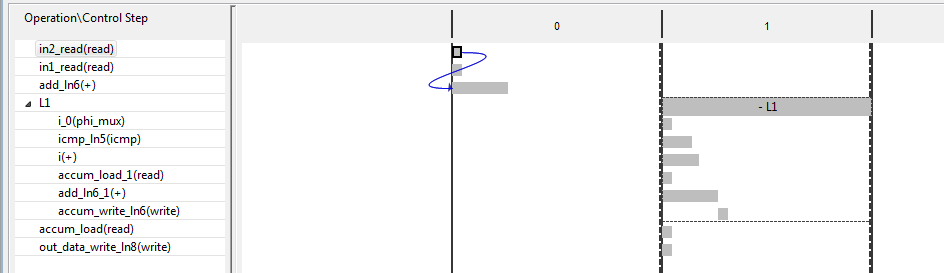


Рисунок 2.4. Schedule viewer

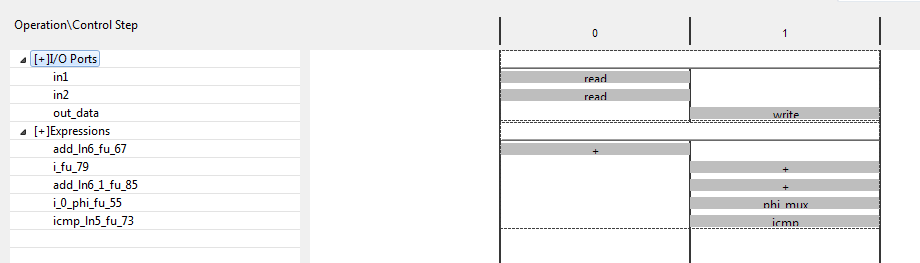


Рисунок 2.5. Resourse viewer

## 2.3 C|RTL моделирование

Результаты C|RTL приведены на рисунке 2.6.

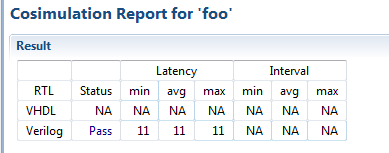


Рисунок 2.6. Отчет о моделировании

Временная диаграмма приведена на рисунке 2.7.

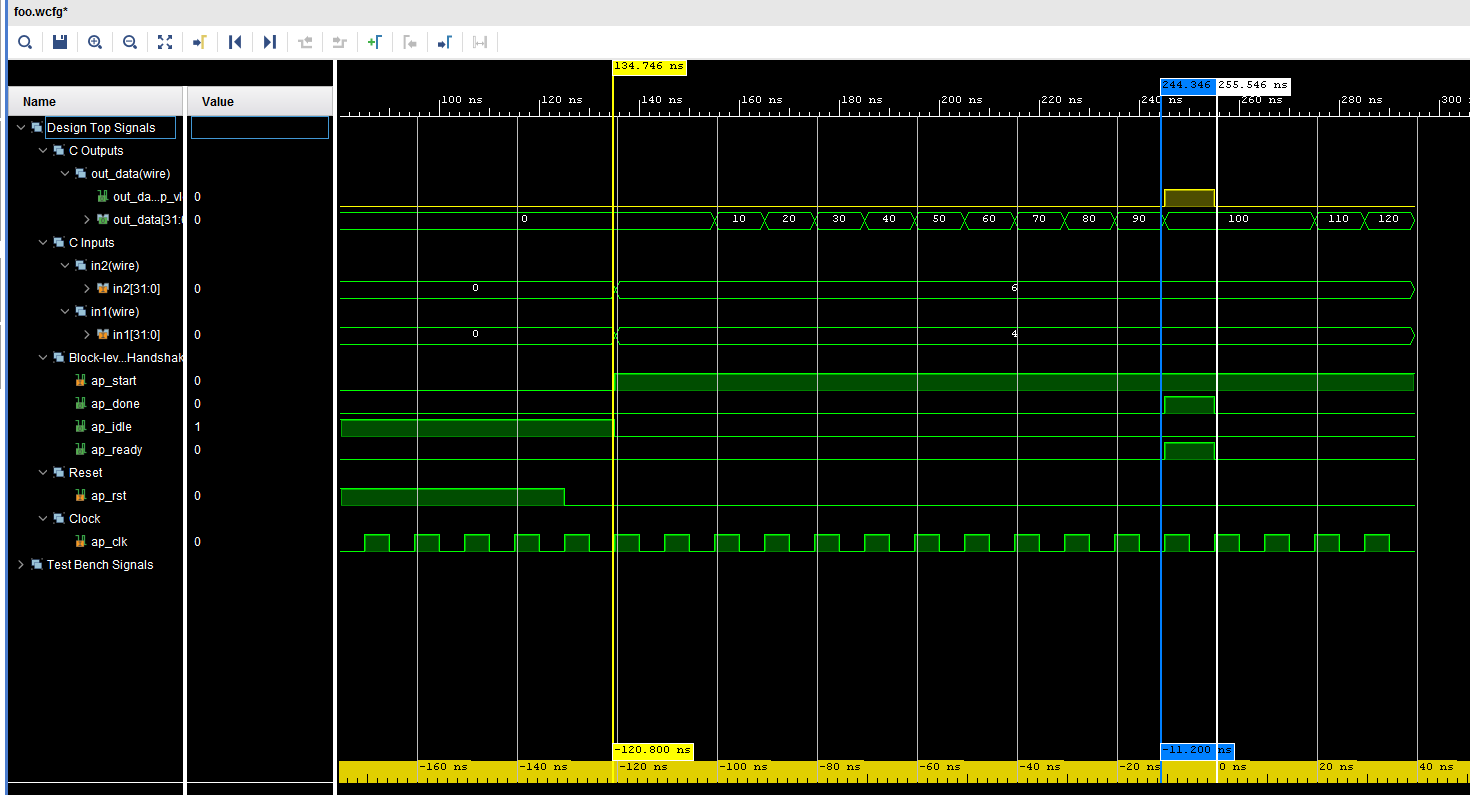


Рисунок 2.7. Временная диаграмма

# **3. Решение №2**

## 3.1 Параметры второго решения

Пропишем директиву PIPELINE примененную к циклу, рисунок 3.1.

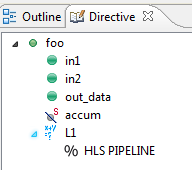


Рисунок 3.1. Директива PIPELINE примененная к циклу

## 3.2 Синтез

Результаты синтеза с оценкой производительности и используемых ресурсов представлены на рисунках 3.2 и 3.3 соответственно.

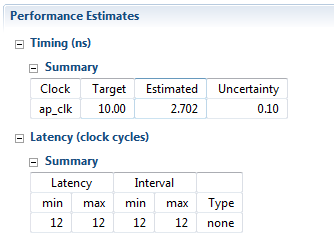


Рисунок 3.2. Performance estimates – summary

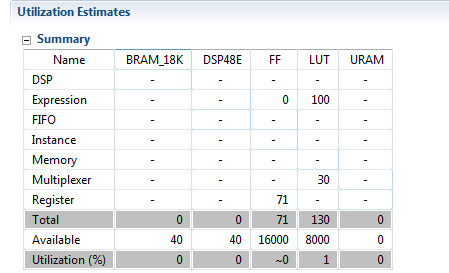


Рисунок 3.3. Utilization estimates – summary

Диаграмма операционного расписания с указанием Latency и диаграмма оперционного просмоторщика ресурсов приведены на рисунках 3.4 и 3.5.

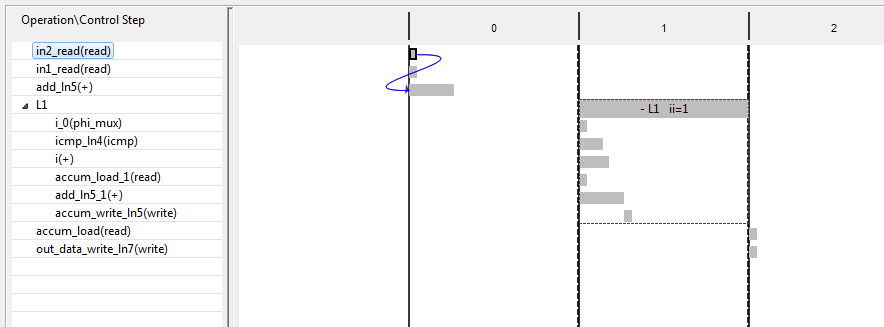


Рисунок 3.4. Schedule viewer

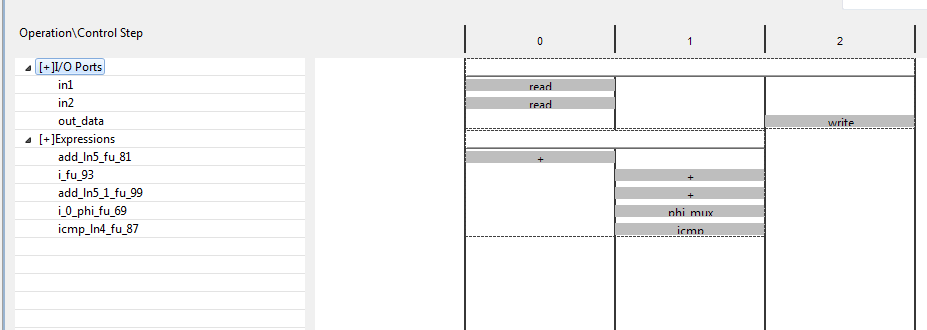


Рисунок 3.5. Resource viewer

## 3.3 C|RTL моделирование

Результаты C|RTL приведены на рисунке 3.6.

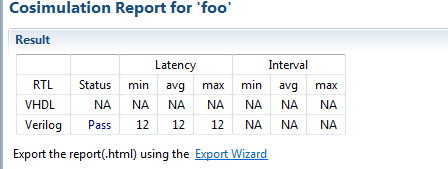


Рисунок 3.6. Отчет о моделировании

Временная диаграмма приведена на рисунке 3.7.

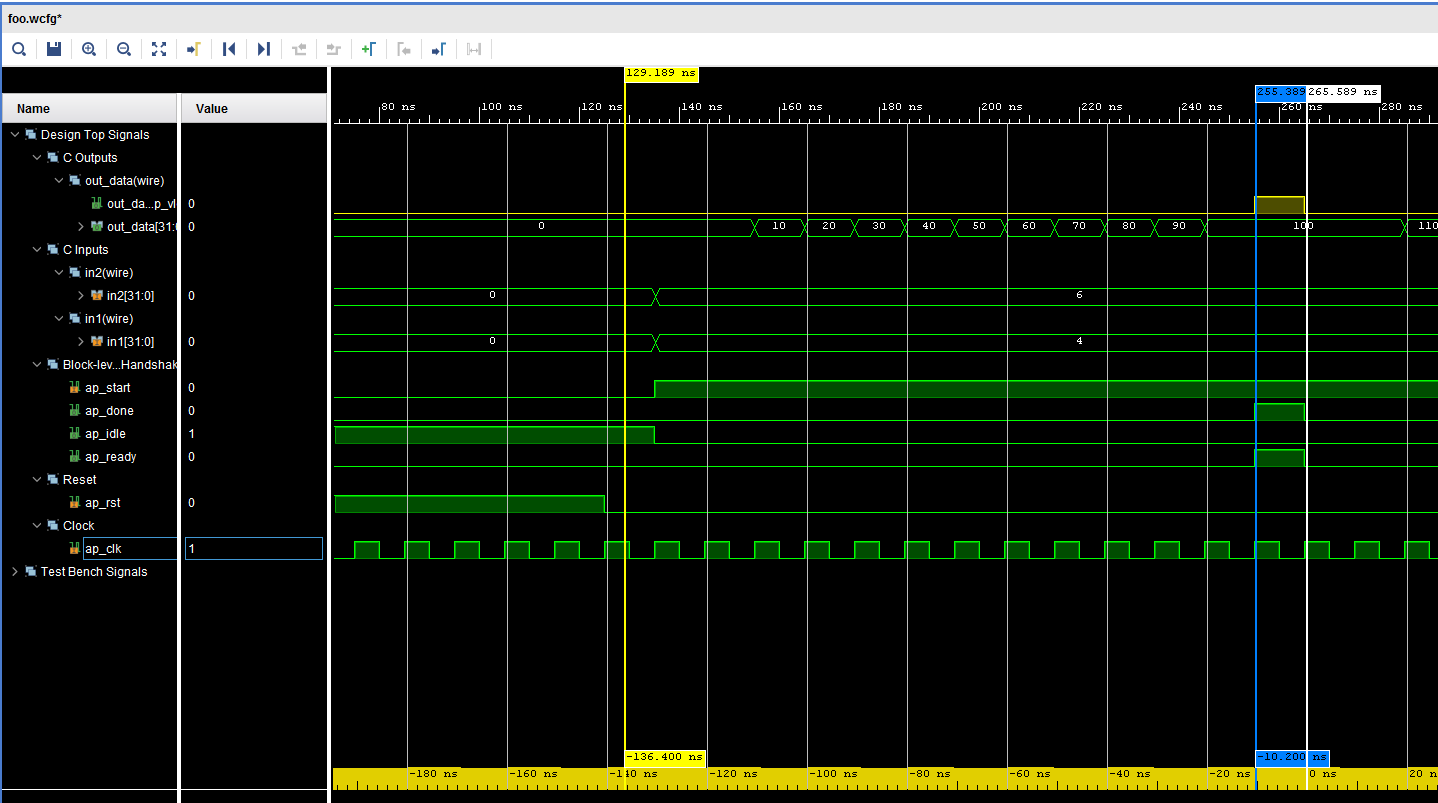


Рисунок 3.7. Временная диаграмма

# **4. Решение № 3.**

## 4.1 Параметры третьего решения

Пропишем директиву PIPELINE внешнего цикла, рисунок 4.1.

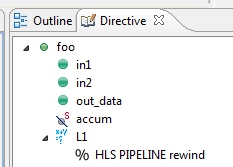


Рисунок 4.1. Директива PIPELINE примененная внутри внешнего цикла

## 

## 4.2 Синтез

Результаты синтеза с оценкой производительности и используемых ресурсов представлены на рисунках 4.2. и 4.3 соответственно.

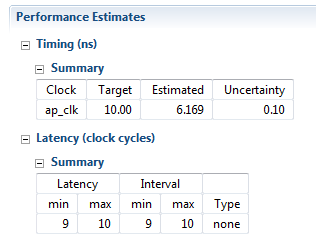


Рисунок 4.2. Performance estimates – summary

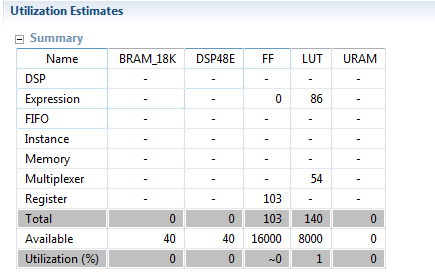


Рисунок 4.3. Utilization estimates – summary

Диаграмма операционного расписания с указанием Latency и диаграмма оперционного просмоторщика ресурсов приведены на рисунках 4.4. и 4.5.

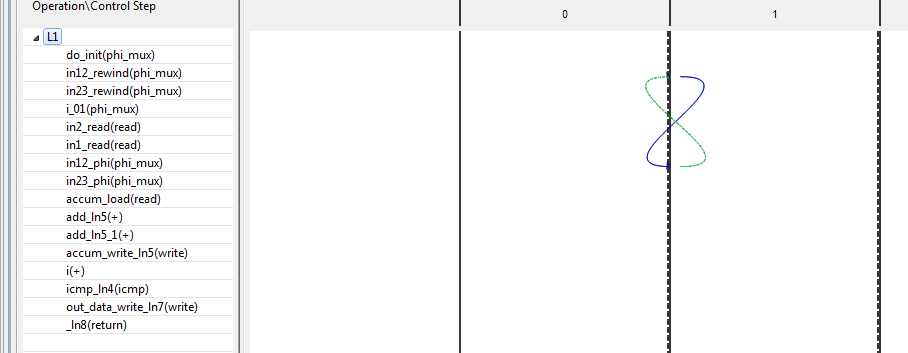


Рисунок 4.4. Schedule viewer

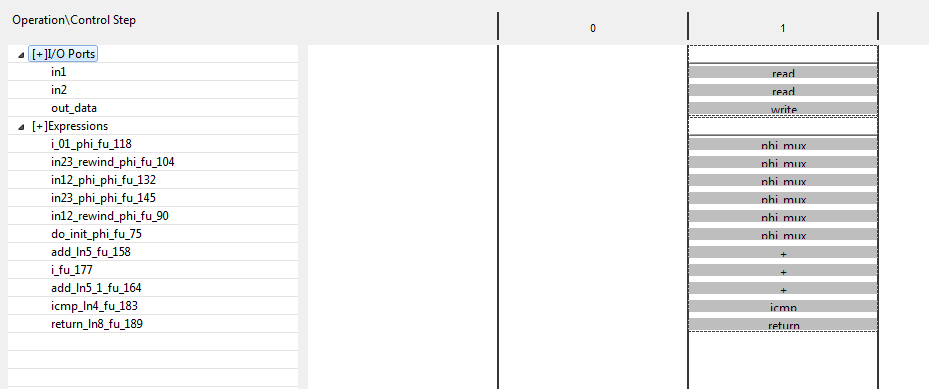


Рисунок 4.5. Resource viewer

## 4.3 C|RTL моделирование

Результаты C|RTL приведены на рисунке 4.6.

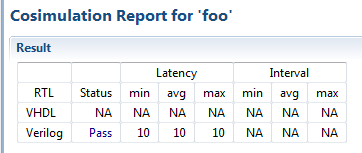


Рисунок 4.6. Отчет о моделировании

Временная диаграмма приведена на рисунке 4.7.

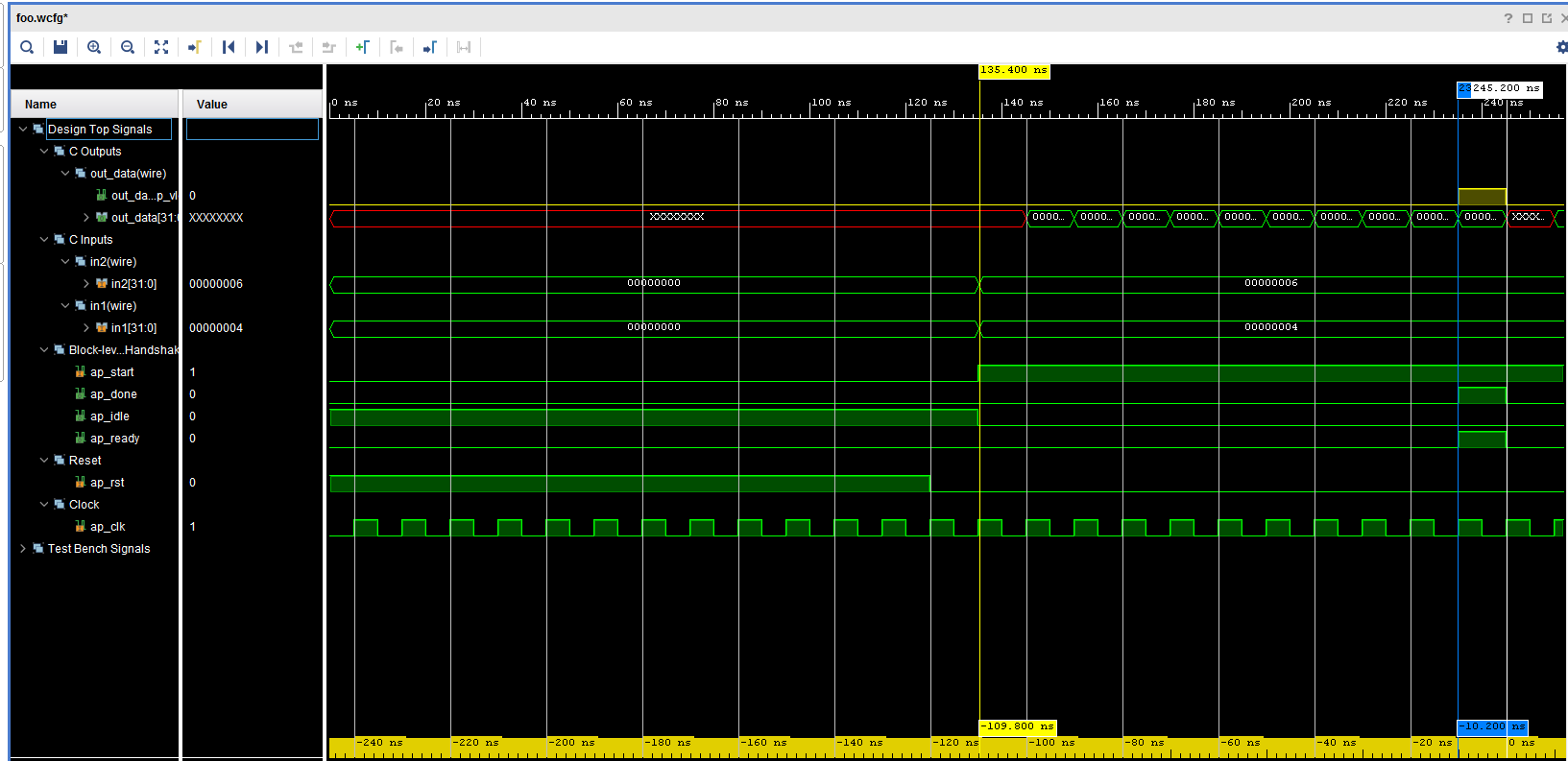


Рисунок 4.7. Временная диаграмма

# **5. Вывод**

Функция «rewind» позволяет сократить время выполнения за счет того, что она начинает выполнение цикла верхнего уровня сразу по его завершению что и видно при сравнении решений 2 и 3. Хотя конкретно в данном случае применение директивы PIPELINE лишь ухудшило стандартное решение.

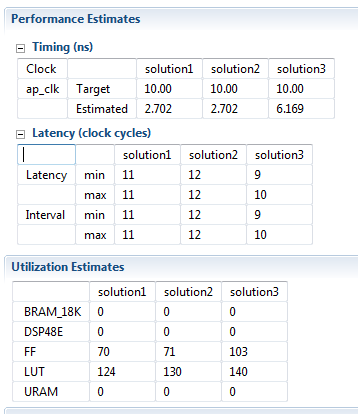


Рисунок 5.1 Сравнительный отчет решений