Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Лабораторная работа 9**

**Предмет: Проектирование реконфигурируемых гибридных вычислительных систем**

**Тема: Optimizing array structures**

**Задание 2**

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

(подпись)

Руководитель Антонов А.П.

(подпись)

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

Санкт – Петербург

2020

Оглавление

[1. Задание 5](#_Toc24036409)

[1. Часть 1 13](#_Toc24036410)

[1.1 Исходный код 13](#_Toc24036411)

[1.2 Решение 1а 14](#_Toc24036412)

[1.2.1 Моделирование 14](#_Toc24036413)

[1.2.2 Синтез 14](#_Toc24036414)

[1.3 Решение 2а 16](#_Toc24036415)

[1.3.1 Параметры решения 16](#_Toc24036416)

[1.3.2 Синтез 17](#_Toc24036417)

[1.3.3 Выводы по решению 19](#_Toc24036418)

[1.4 Решение 3а 19](#_Toc24036419)

[1.4.1 Параметры решения 19](#_Toc24036420)

[1.4.2 Синтез 20](#_Toc24036421)

[1.4.3 Выводы по решению 22](#_Toc24036422)

[1.5 Решение 4а 23](#_Toc24036423)

[1.5.1 Параметры решения 23](#_Toc24036424)

[1.5.2 Синтез 23](#_Toc24036425)

[1.5.3 Выводы по решению 26](#_Toc24036426)

[1.6 Решение 5а 26](#_Toc24036427)

[1.6.1 Параметры решения 26](#_Toc24036428)

[1.6.2 Синтез 27](#_Toc24036429)

[1.6.3 Выводы по решению 29](#_Toc24036430)

[1.7 Решение 6а 29](#_Toc24036431)

[1.7.1 Параметры решения 29](#_Toc24036432)

[1.7.2 Синтез 30](#_Toc24036433)

[1.7.3 Выводы по решению 32](#_Toc24036434)

[1.8 Решение 7а 32](#_Toc24036435)

[1.8.1 Параметры решения 32](#_Toc24036436)

[1.8.2 Синтез 33](#_Toc24036437)

[1.8.3 Выводы по решению 35](#_Toc24036438)

[1.9 Выводы по первой части 35](#_Toc24036439)

[2. Часть 2 36](#_Toc24036440)

[2.2 Решение 1b 37](#_Toc24036441)

[2.2.1 Моделирование 37](#_Toc24036442)

[2.2.2 Синтез 37](#_Toc24036443)

[2.3 Решение 2b 39](#_Toc24036444)

[2.3.1 Параметры решения 39](#_Toc24036445)

[2.3.2 Синтез 40](#_Toc24036446)

[2.3.3 Выводы по решению 42](#_Toc24036447)

[2.4 Решение 3b 42](#_Toc24036448)

[2.4.1 Параметры решения 42](#_Toc24036449)

[2.4.2 Синтез 43](#_Toc24036450)

[2.4.3 Выводы по решению 45](#_Toc24036451)

[2.5 Решение 4b 46](#_Toc24036452)

[2.5.1 Параметры решения 46](#_Toc24036453)

[2.5.2 Синтез 46](#_Toc24036454)

[2.5.3 Выводы по решению 49](#_Toc24036455)

[2.6 Решение 5b 50](#_Toc24036456)

[2.6.1 Параметры решения 50](#_Toc24036457)

[2.6.2 Синтез 50](#_Toc24036458)

[2.6.3 Выводы по решению 52](#_Toc24036459)

[2.7 Решение 6b 53](#_Toc24036460)

[2.7.1 Параметры решения 53](#_Toc24036461)

[2.7.2 Синтез 53](#_Toc24036462)

[2.7.3 Выводы по решению 55](#_Toc24036463)

[2.8 Решение 7b 56](#_Toc24036464)

[2.8.1 Параметры решения 56](#_Toc24036465)

[2.8.2 Синтез 56](#_Toc24036466)

[2.8.3 Выводы по решению 58](#_Toc24036467)

[2.9 Сравнение решений 59](#_Toc24036468)

[2.10 Решение 8b 59](#_Toc24036469)

[2.10.1 Параметры решения 59](#_Toc24036470)

[2.10.2 Синтез 60](#_Toc24036471)

[2.10.3 Выводы по решению 62](#_Toc24036472)

# 1. Задание

* Создать проект lab9\_2
* Микросхема: xa7a12tcsg325-1q

ЧАСТЬ 1

* Создать функцию

*foo\_a: входной массив short d\_in[N]; выходной массив short d\_out [N/4].*

*for (short i=0; i<N/4; i++){*

*d\_out[i] = d\_in[i]\*d\_in[i+8] + d\_in[i+4]\*d\_in[i+12];*

*}*

*N=16*

* Создать тест lab9\_2\_test.c для проверки функции. Осуществить моделирование (с выводом результатов в консоль)
* Исследование:
* Solution\_1а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию bram; RAM\_1P\_BRAM для входного (и выходного) массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Solution\_2а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного (и выходного) массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_1a и solution\_2a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_3а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_2P для входного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_2a и solution\_3a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_4а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного массива
  + установить array\_partition; block; factor =2 для входного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_3a и solution\_4a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_5а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_4a и solution\_5a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_6а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_2P для входного и выходного массивов
  + установить array\_partition; block; factor =2 для входного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_5a и solution\_6a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_7а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_2P для входного и выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_6a и solution\_7a) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…

Сделать сводную таблицу (*S\_x/Latency/II – номер решения/Latency/II)*

|  |  |  |
| --- | --- | --- |
|  | RAM\_1P | RAM\_2P |
| Без block | *S\_x/Latency/II* |  |
| block; factor =2 |  |  |
| block; factor =4 |  |  |

ЧАСТЬ 2

* Создать функцию

*foo\_b: входной массив short d\_in[N]; выходной массив short d\_out [N].*

*for (short i=0; i<N/4; i++){*

*d\_out[i] = d\_in[i]\*d\_in[i+4];*

*d\_out[i+1]= d\_in[i+8]\*d\_in[i+12];*

*d\_out[i+2]= d\_in[i]\*d\_in[i+12];*

*d\_out[i+3]= d\_in[i+4]\*d\_in[i+8];*

*}*

*N=16*

* Solution\_1b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и выходного массивов
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Solution\_2b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_1b и solution\_2b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_3b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + установить array\_partition; cyclic; factor =2 для выходного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_2b и solution\_3b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_4b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + установить array\_partition; cyclic; factor =4 для выходного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_3b и solution\_4b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_5b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и RAM\_2P для выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + установить array\_partition; cyclic; factor =1 для выходного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_3b и solution\_5b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_6b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и RAM\_2P для выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + установить array\_partition; cyclic; factor =2 для выходного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_5b и solution\_6b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…
* Solution\_7b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и RAM\_2P для выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + установить array\_partition; cyclic; factor =4 для выходного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_6b и solution\_7b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…

Сделать сводную таблицу (*S\_x/Latency/II – номер решения/Latency/II)*

|  |  |  |
| --- | --- | --- |
|  | RAM\_1P | RAM\_2P |
| Без cyclic | *S\_x/Latency/II* |  |
| cyclic; factor =2 |  |  |
| cyclic; factor =4 |  |  |

* Solution\_8b
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ap\_memory; RAM\_1P для входного и выходного массивов
  + установить array\_partition; block; factor =4 для входного массива
  + установить array\_partition; complete для выходного массива
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * 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
* Сравнить два решения (solution\_2b и solution\_8b) и сделать выводы: зависимость от типа интерфейса; объяснить количество использованных умножителей; объяснить (посчитать) число циклов Latency, II…

# 1. Часть 1

## 1.1 Исходный код

Исходный код функции

**#define** N 16

**void** **foo\_a** (**short** d\_in[N], **short** d\_out[N/4]) {

**for** (**short** i = 0; i < N/4; i++){

d\_out[i] = d\_in[i]\*d\_in[i+8] + d\_in[i+4]\*d\_in[i+12];

}

}

Исходный код теста

**#include** <stdio.h>

**#define** N 16

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

**short** d\_in[N], d\_out[N/4], d\_out\_exp[N/4];

**int** i;

**for** (i = 0; i < N; i++) {

d\_in[i] = i;

}

**for** (i = 0; i < N/4; i++) {

d\_out\_exp[i] = d\_in[i]\*d\_in[i+8] + d\_in[i+4]\*d\_in[i+12];

}

foo\_a(d\_in, d\_out);

**for** (i = 0; i < N/4; i++) {

**printf**(" %d\*%d + %d\*%d = %d == %d\n", d\_in[i], d\_in[i+8], d\_in[i+4], d\_in[i+12], d\_out[i], d\_out\_exp[i]);

**if** (d\_out[i] != d\_out\_exp[i]) {

**printf**(" ----ERROR!----\n");

**return** -1;

}

}

**printf**(" ----Test pass!----\n");

**return** 0;

}

## 1.2 Решение 1а

### 1.2.1 Моделирование

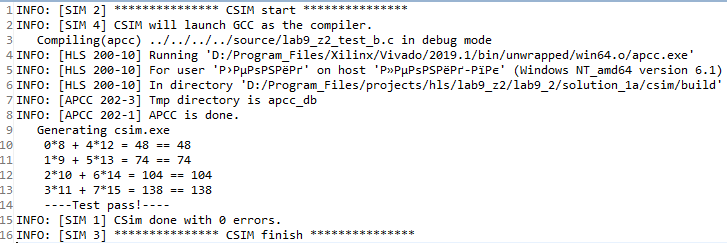


Рисунок 1. Логи моделирования

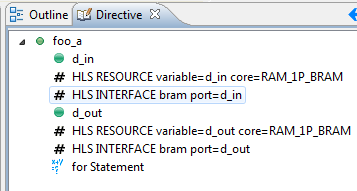


Рисунок 2. Примененные директивы

### 1.2.2 Синтез

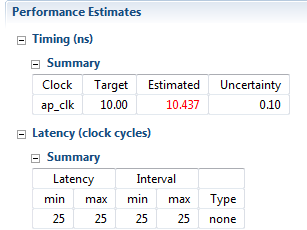


Рисунок 3. Performance estimates – summary

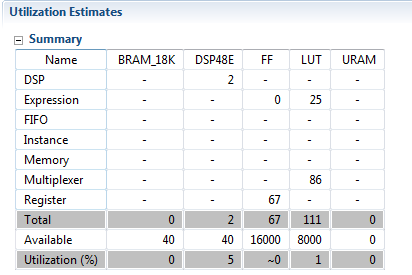


Рисунок 4. Utilization estimates – summary

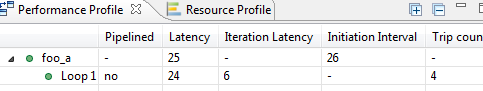


Рисунок 5. Performance profile

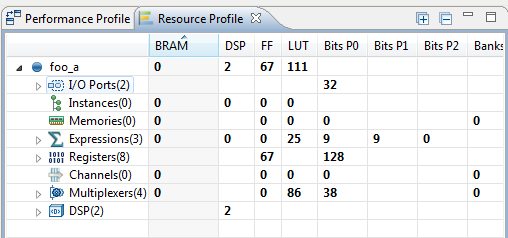


Рисунок 6. Resource profile

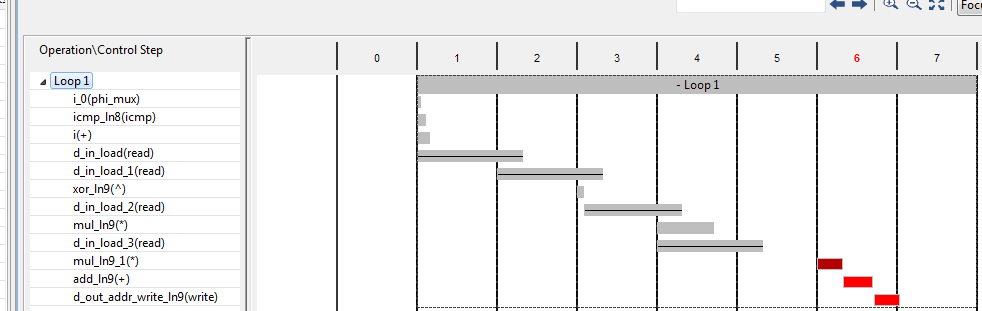


Рисунок 7. Schedule viewer



Рисунок 8. Resourse viewer

## 1.3 Решение 2а

### 1.3.1 Параметры решения

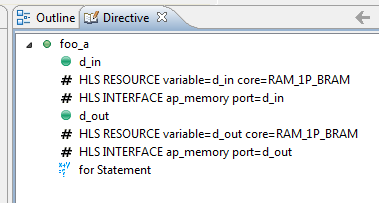


Рисунок 9. Примененные директивы

### 1.3.2 Синтез

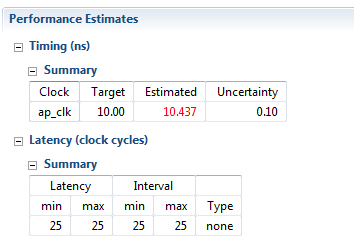


Рисунок 10. Performance estimates – summary

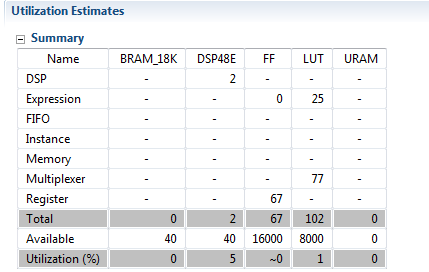


Рисунок 11. Utilization estimates – summary

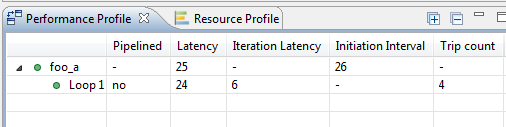


Рисунок 12. Performance profile

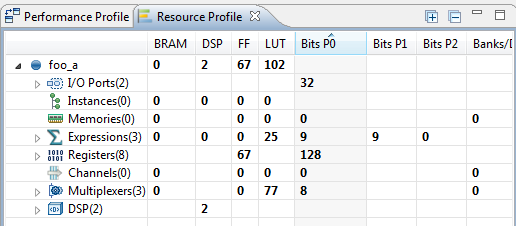


Рисунок 13. Resource profile

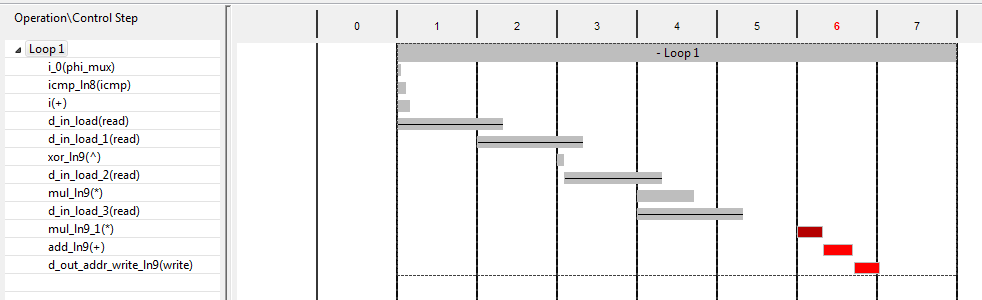


Рисунок 14. Schedule viewer

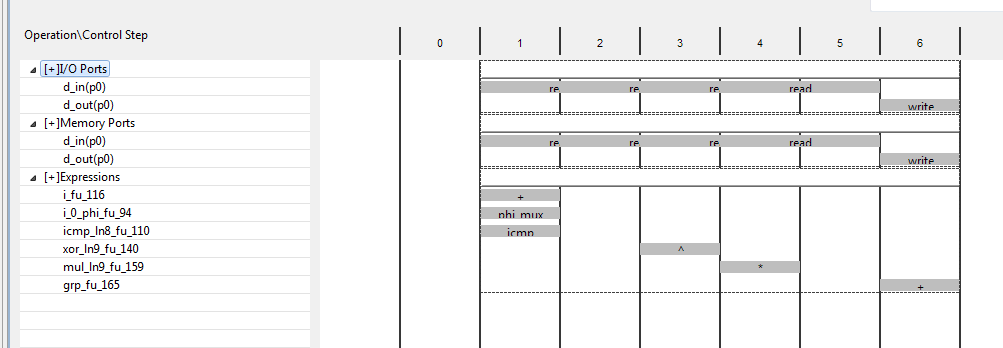


Рисунок 15. Resourse viewer

### 1.3.3 Выводы по решению

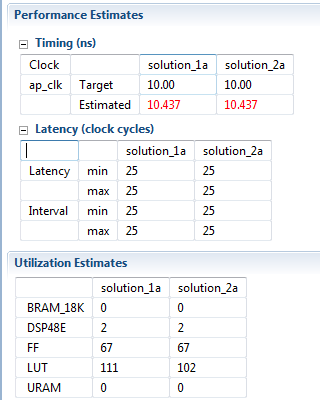


Рисунок 6. Сравнительный отчет

Разница между решениями лишь в разном количестве LUT. Все остальные параметры одинаковые.

## 1.4 Решение 3а

### 1.4.1 Параметры решения

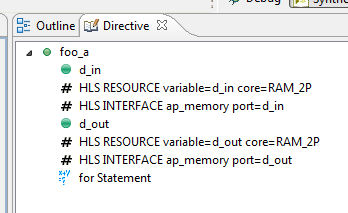


Рисунок 17. Примененные директивы

### 1.4.2 Синтез

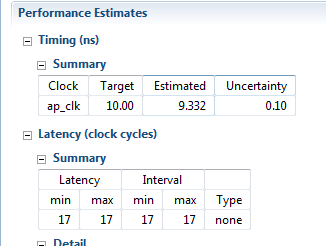


Рисунок 18. Performance estimates – summary

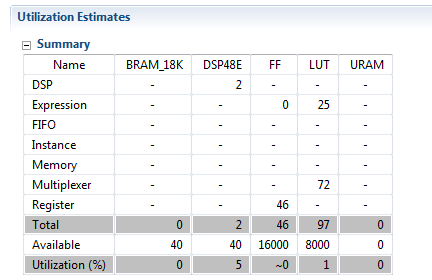


Рисунок 19. Utilization estimates – summary

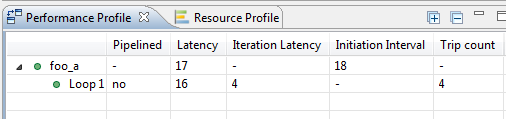


Рисунок 20. Performance profile

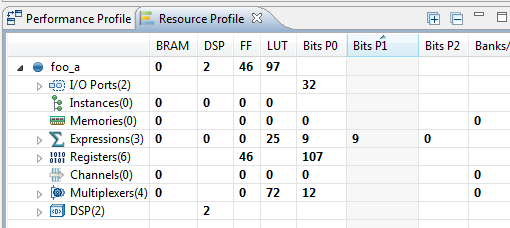


Рисунок 21. Resource profile

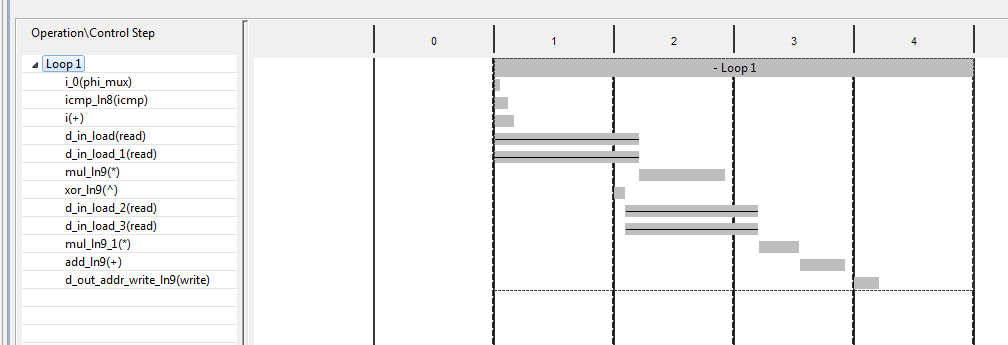


Рисунок 22. Schedule viewer

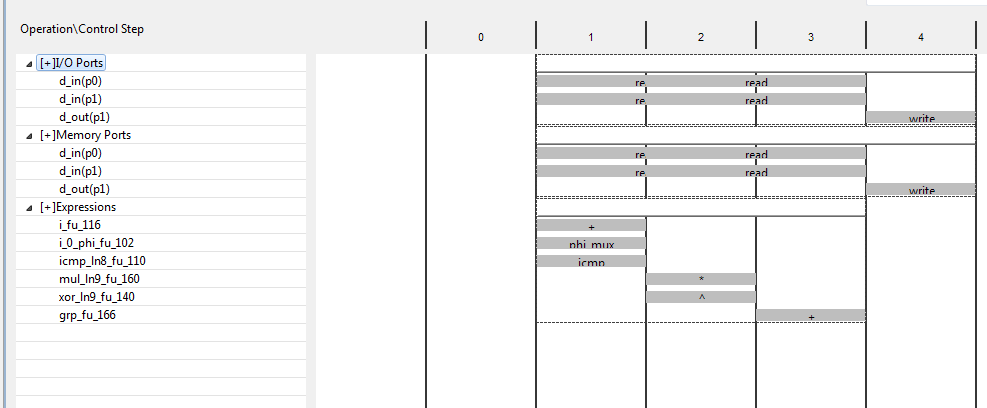


Рисунок 23. Resourse viewer

### 1.4.3 Выводы по решению

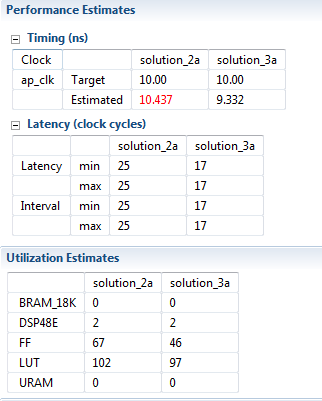


Рисунок 24. Сравнительный отчет

Решение 3а быстрее и экономичнее, чем 2а. Оно используется столько же умножителей, однако благодаря двухпортовой памяти можно производить по 2 чтения из входного массива, что сокращает выполнение 1 итерации с 6 до 4 тактов.

## 1.5 Решение 4а

### 1.5.1 Параметры решения

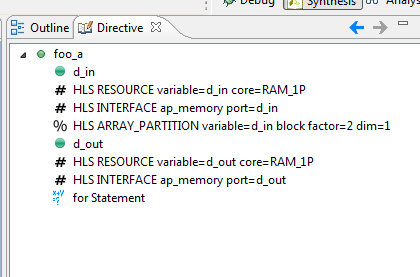


Рисунок 25. Примененные директивы

### 1.5.2 Синтез

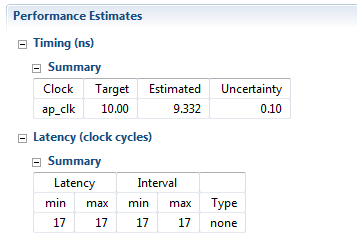


Рисунок 26. Performance estimates – summary

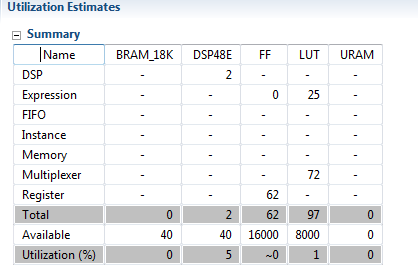


Рисунок 27. Utilization estimates – summary

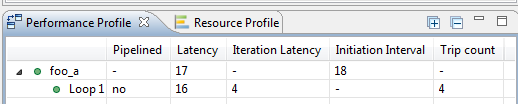


Рисунок 28. Performance profile

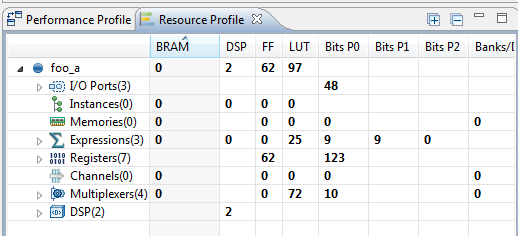


Рисунок 29. Resource profile

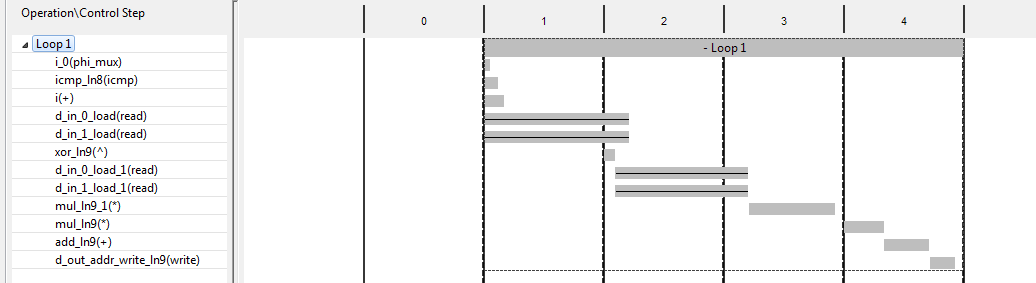


Рисунок 30. Schedule viewer

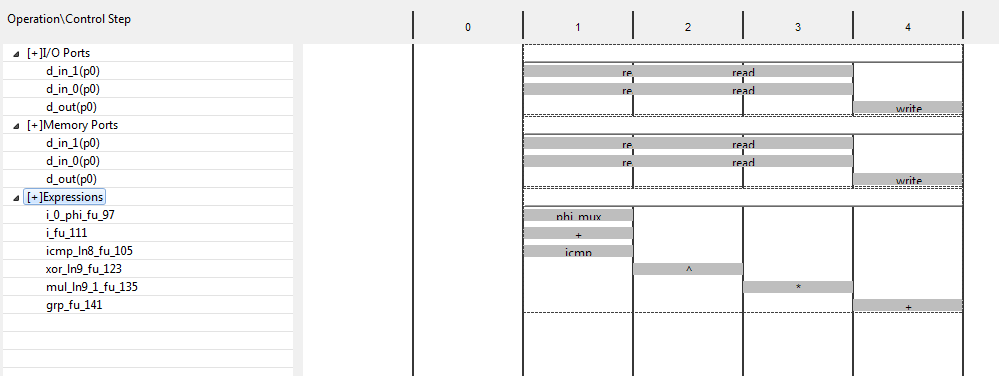


Рисунок 31. Resourse viewer

### 1.5.3 Выводы по решению

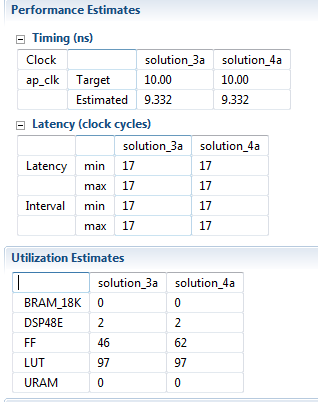


Рисунок 32. Сравнительный отчет

Решения 3а и 4а практически идентичны между собой, но у 4а имеется небольшое увеличение в используемых ресурсах.

## 1.6 Решение 5а

### 1.6.1 Параметры решения

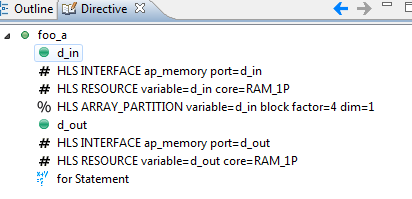


Рисунок 33. Примененные директивы

### 1.6.2 Синтез

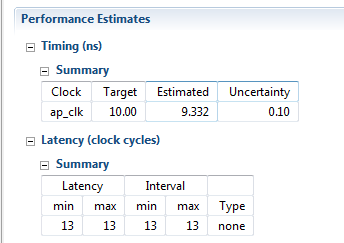


Рисунок 34. Performance estimates – summary

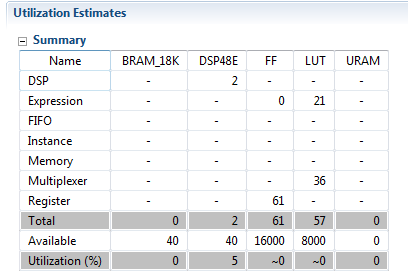


Рисунок 35. Utilization estimates – summary

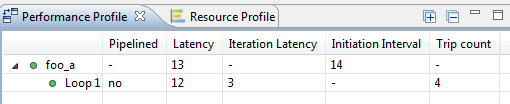


Рисунок 36. Performance profile

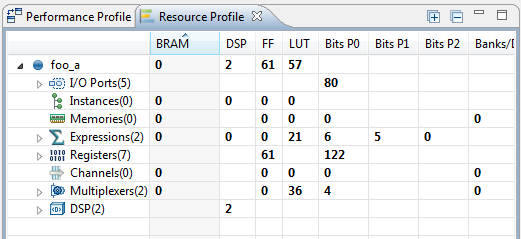


Рисунок 37. Resource profile

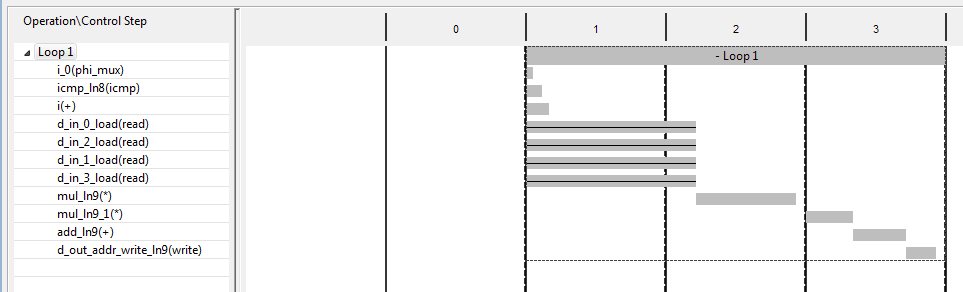


Рисунок 38. Schedule viewer

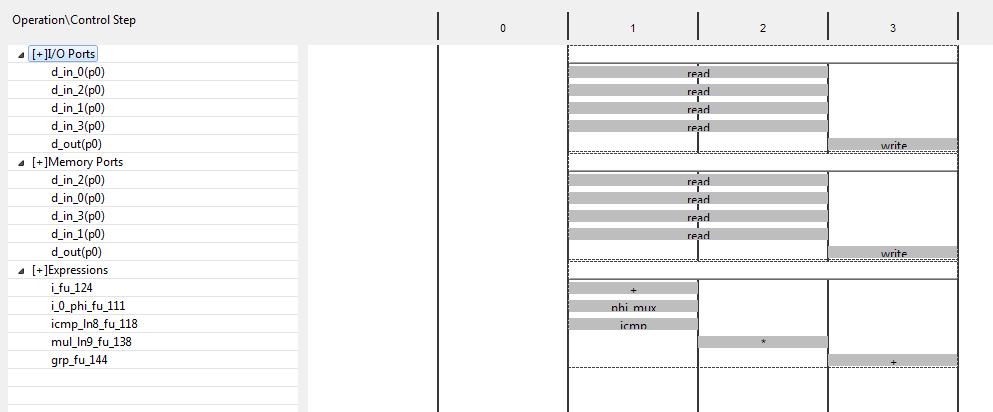


Рисунок 39. Resourse viewer

### 1.6.3 Выводы по решению

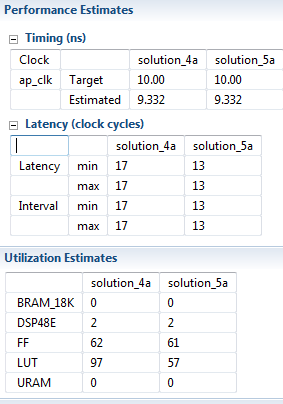


Рисунок 40. Сравнительный отчет

Использование ресурсов почти не изменилось. Умножителей по-прежнему 2. Затраты по сократились на 4 такта. Это объясняется тем, что теперь используется 4 экземпляра однопортовой памяти и все 4 операции чтения проходят одновременно.

## 1.7 Решение 6а

### 1.7.1 Параметры решения

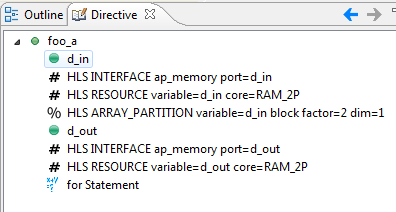


Рисунок 41. Примененные директивы

### 1.7.2 Синтез

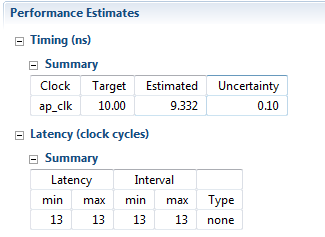


Рисунок 42. Performance estimates – summary

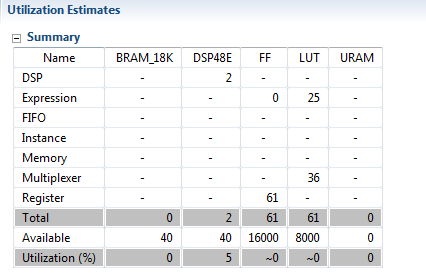


Рисунок 43. Utilization estimates – summary

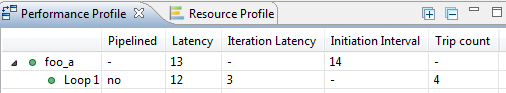


Рисунок 44. Performance profile

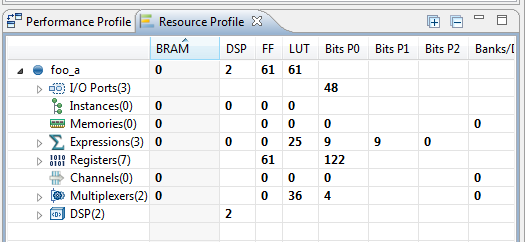


Рисунок 45. Resource profile

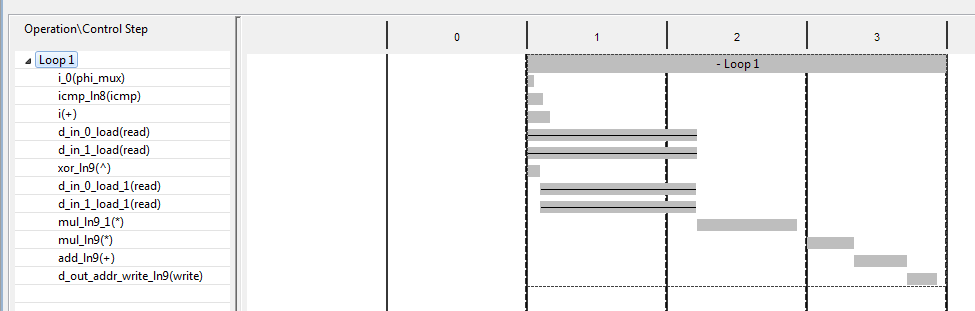


Рисунок 46. Schedule viewer

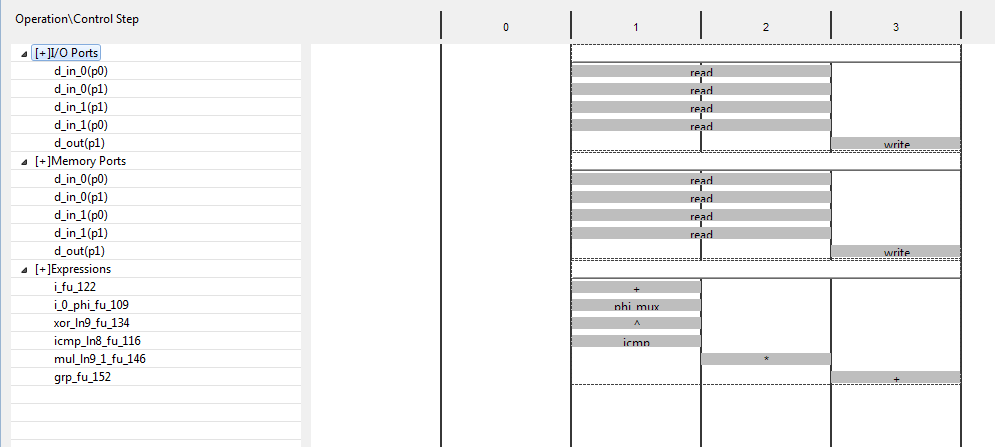


Рисунок 47. Resourse viewer

### 1.7.3 Выводы по решению

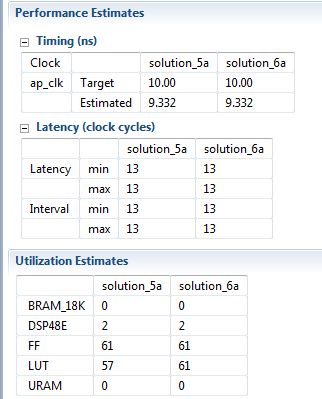


Рисунок 48. Сравнительный отчет

По затратам оба решения похожи. Задержка и интервал инициализации оказались одинаковыми. Использование 4 однопортовых памяти или 2 двухпортовых приводит к практически одинаковым результатам.

## 1.8 Решение 7а

### 1.8.1 Параметры решения

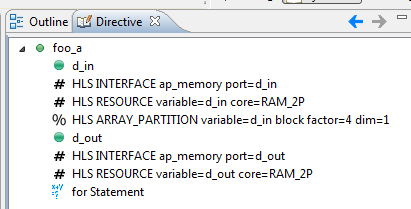


Рисунок 49. Примененные директивы

### 1.8.2 Синтез

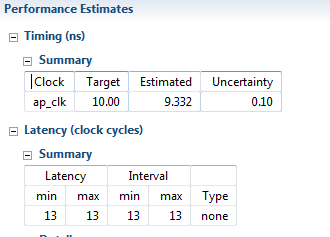


Рисунок 50. Performance estimates – summary

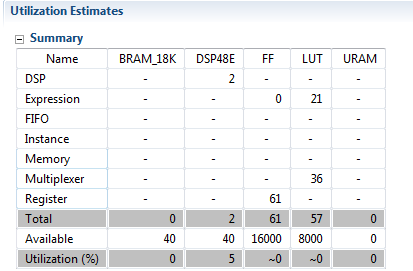


Рисунок 51. Utilization estimates – summary

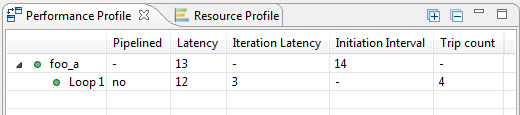


Рисунок 52. Performance profile

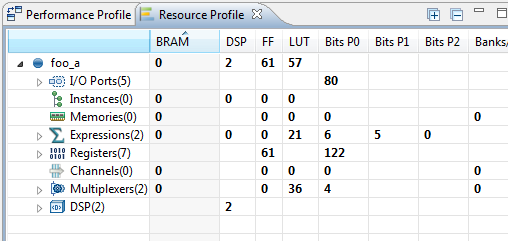


Рисунок 53. Resource profile

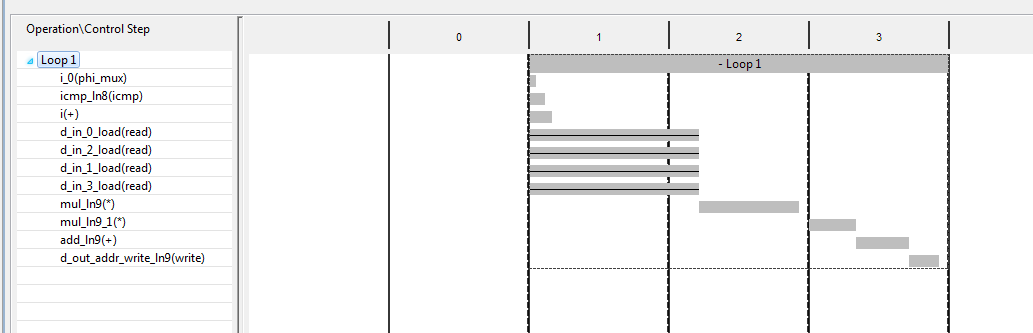


Рисунок 54. Schedule viewer

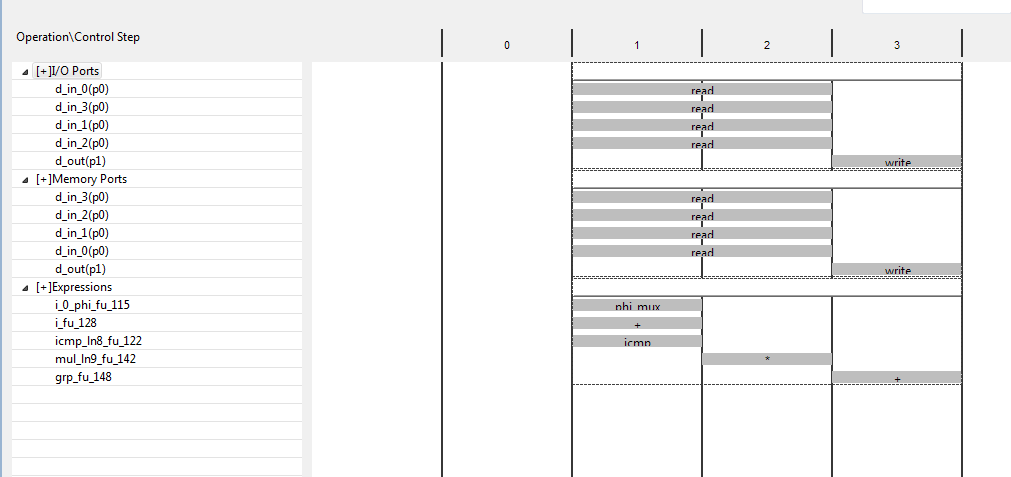


Рисунок 55. Resourse viewer

### 1.8.3 Выводы по решению

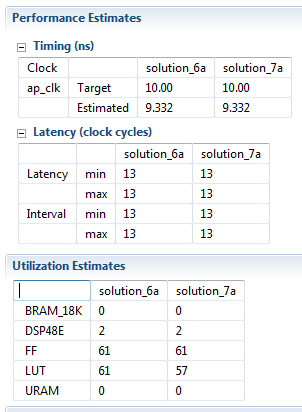


Рисунок 56. Сравнительный отчет

Решения 6а и 7а практически идентичны между собой как по времени выполнения так и по используемым ресурсам. Для данного устройства нет необходимости иметь 4 двухпортовых памяти.

## 1.9 Выводы по первой части

|  |  |  |  |
| --- | --- | --- | --- |
|  | RAM\_1P | RAM\_2P | BRAM\_1P |
| Без block | 2a/25/25 | 3a/17/17 | 1а/25/25 |
| block; factor =2 | 4а/17/17 | 6a/13/13 |  |
| block; factor =4 | 5a/13/13 | 7a/13/13 |  |

Наименьшие задержки получили те решения, где есть возможность осуществлять 4 операции чтения массива. Решение, которое имеет больше 4 чтений (7а), реализуется как 5а, так как оно избыточно. Такая избыточность может понадобится если использовать конвейеризацию.

# 2. Часть 2

2.1 Исходный код

Исходный код функции

**#define** N 16

**void** **foo\_b**(**short** d\_in[N], **short** d\_out[N]) {

**for** (**short** i=0; i<N/4; i++){

d\_out[i] = d\_in[i]\*d\_in[i+4];

d\_out[i+1]= d\_in[i+8]\*d\_in[i+12];

d\_out[i+2]= d\_in[i]\*d\_in[i+12];

d\_out[i+3]= d\_in[i+4]\*d\_in[i+8];

}

}

Исходный код теста

**#include** <stdio.h>

**#define** N 16

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

**short** d\_in[N], d\_out[N/4 + 3], d\_out\_exp[N/4 + 3];

**int** i;

**for** (i = 0; i < N; i++) {

d\_in[i] = i;

}

**for** (i = 0; i < N/4; i++){

d\_out\_exp[i] = d\_in[i]\*d\_in[i+4];

d\_out\_exp[i+1]= d\_in[i+8]\*d\_in[i+12];

d\_out\_exp[i+2]= d\_in[i]\*d\_in[i+12];

d\_out\_exp[i+3]= d\_in[i+4]\*d\_in[i+8];

}

foo\_b(d\_in, d\_out);

**for** (i = 0; i < N/4 + 3; i++) {

**printf**(" %d == %d\n", d\_out[i], d\_out\_exp[i]);

**if** (d\_out[i] != d\_out\_exp[i]) {

**printf**(" ----ERROR!----\n");

**return** -1;

}

}

**printf**(" ----Test pass!----\n");

**return** 0;

}

## 2.2 Решение 1b

### 2.2.1 Моделирование

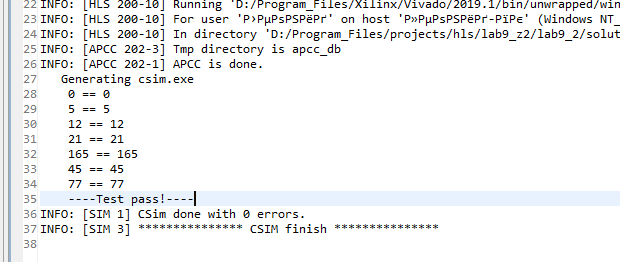


Рисунок 57. Логи моделирования

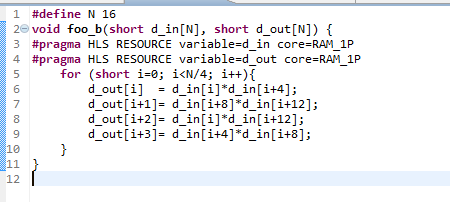


Рисунок 58. Примененные директивы

### 2.2.2 Синтез

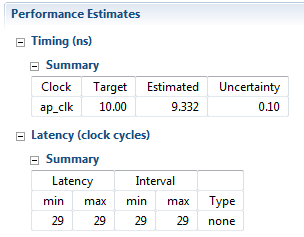


Рисунок 59. Performance estimates – summary

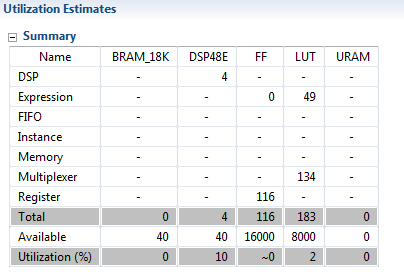


Рисунок 60. Utilization estimates – summary

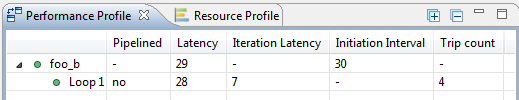


Рисунок 61. Performance profile

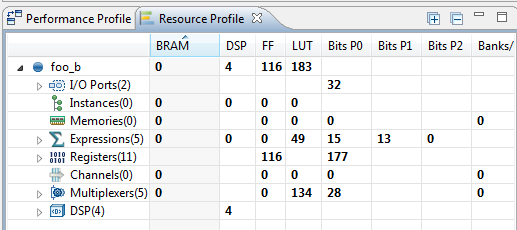


Рисунок 62. Resource profile

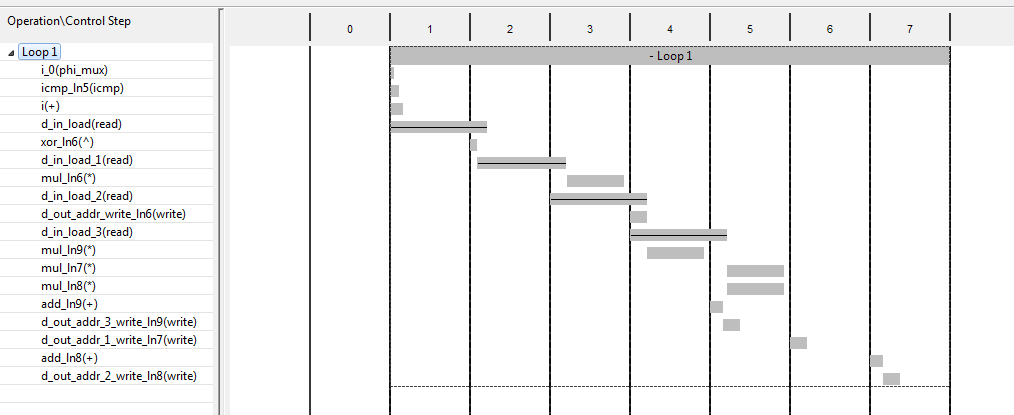


Рисунок 63. Schedule viewer

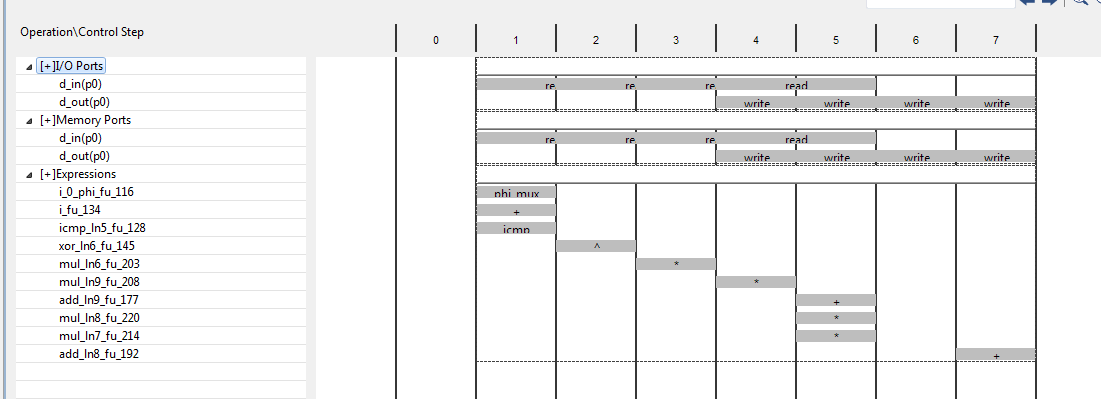


Рисунок 64. Resourse viewer

## 2.3 Решение 2b

### 2.3.1 Параметры решения

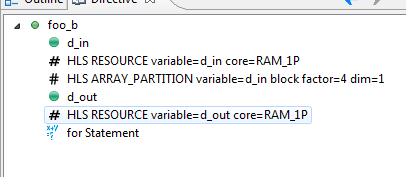


Рисунок 65. Примененные директивы

### 2.3.2 Синтез

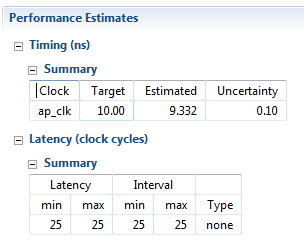


Рисунок 66. Performance estimates – summary

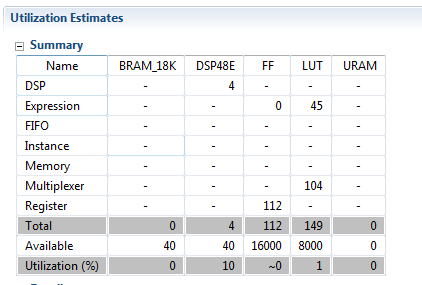


Рисунок 67. Utilization estimates – summary

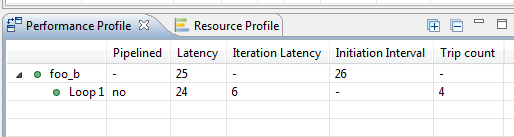


Рисунок 68. Performance profile

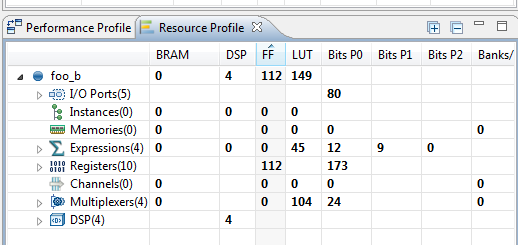


Рисунок 69. Resource profile



Рисунок 70. Schedule viewer

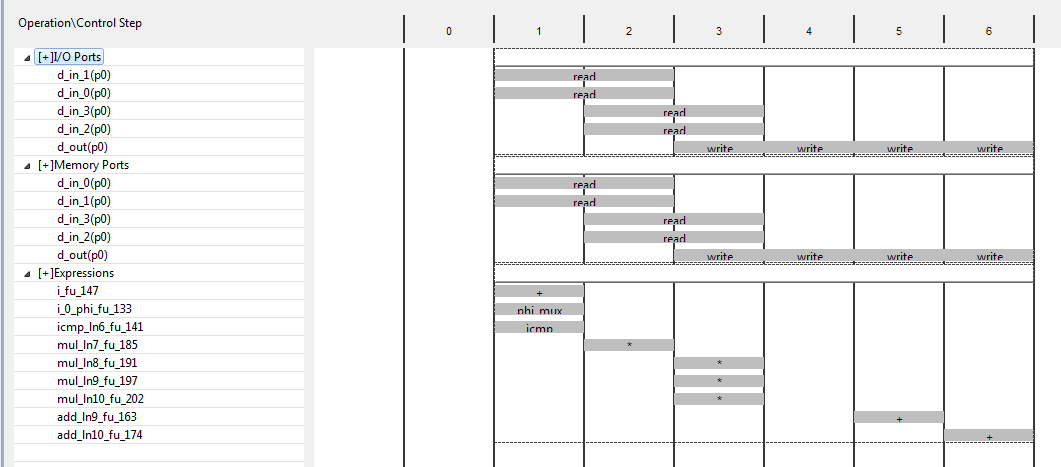


Рисунок 71. Resourse viewer

### 2.3.3 Выводы по решению

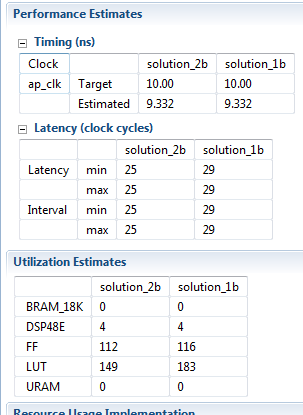


Рисунок 72. Сравнительный отчет

Количество умножителей не изменилось, однако за счет возможности параллельного чтения сократился параметр Iteration Latency – c 7 до 6 тактов.

## 2.4 Решение 3b

### 2.4.1 Параметры решения

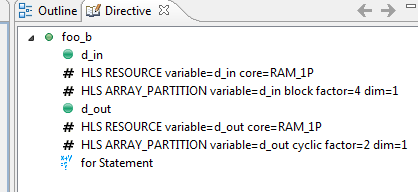


Рисунок 73. Примененные директивы

### 2.4.2 Синтез

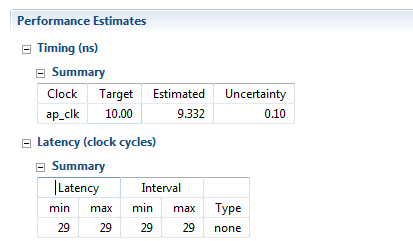


Рисунок 74. Performance estimates – summary

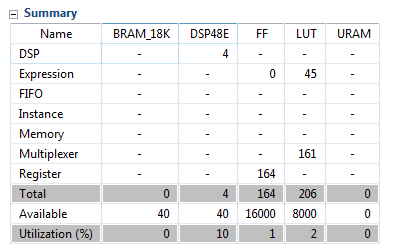


Рисунок 75. Utilization estimates – summary

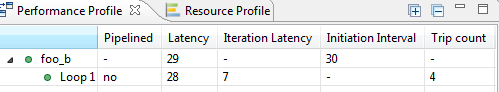


Рисунок 76. Performance profile

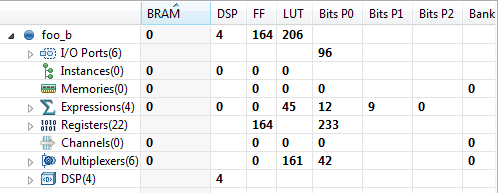


Рисунок 77. Resource profile

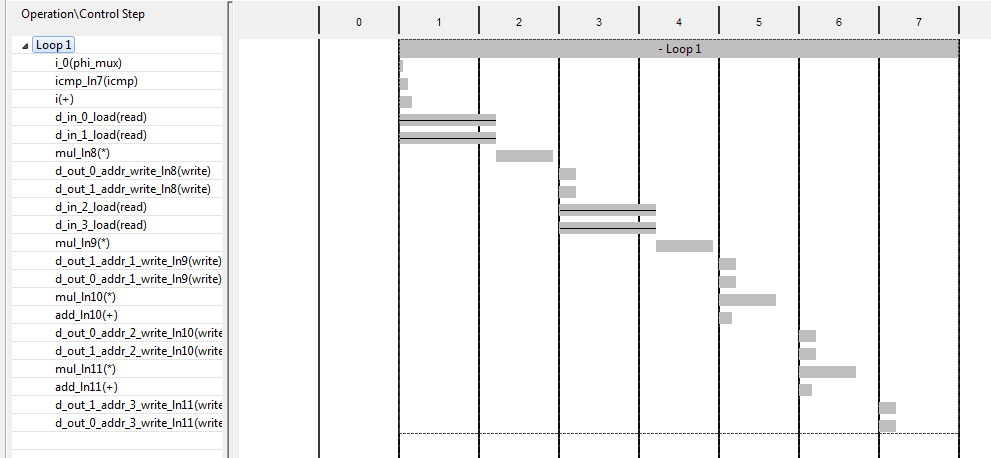


Рисунок 78. Schedule viewer

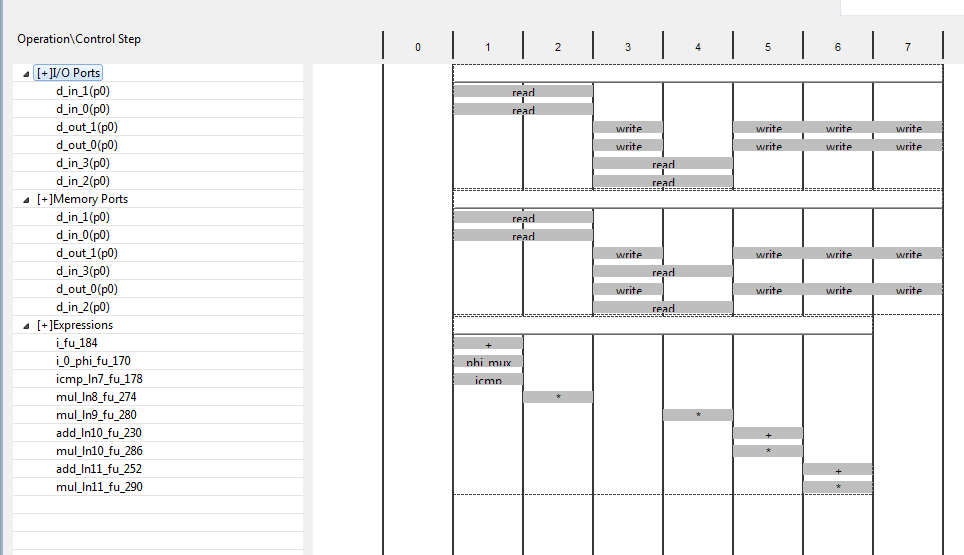


Рисунок 79. Resourse viewer

### 2.4.3 Выводы по решению

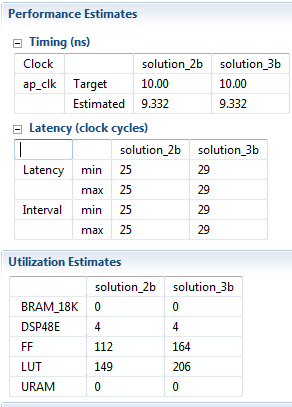


Рисунок 80. Сравнительный отчет

Добавления циклического разбиения выходного массива ухудшило решение по всем показателям, и стало примерно похоже на первое решение.

## 2.5 Решение 4b

### 2.5.1 Параметры решения

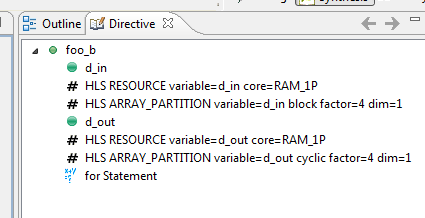


Рисунок 81. Примененные директивы

### 2.5.2 Синтез

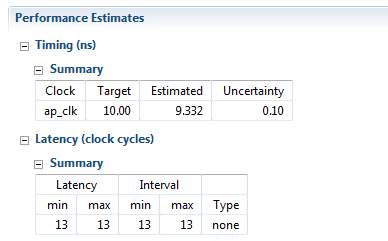


Рисунок 82. Performance estimates – summary

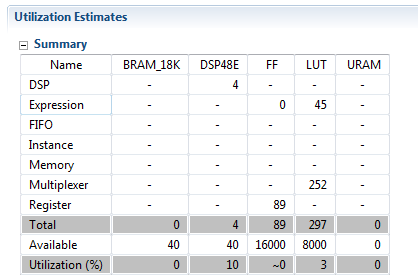


Рисунок 83. Utilization estimates – summary

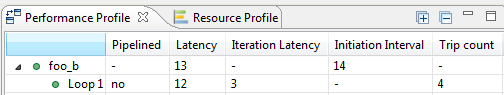


Рисунок 84. Performance profile

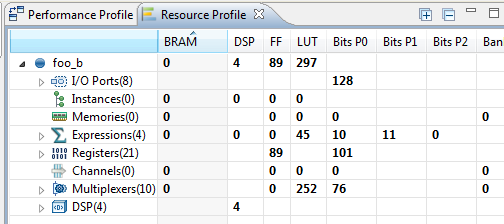


Рисунок 85. Resource profile

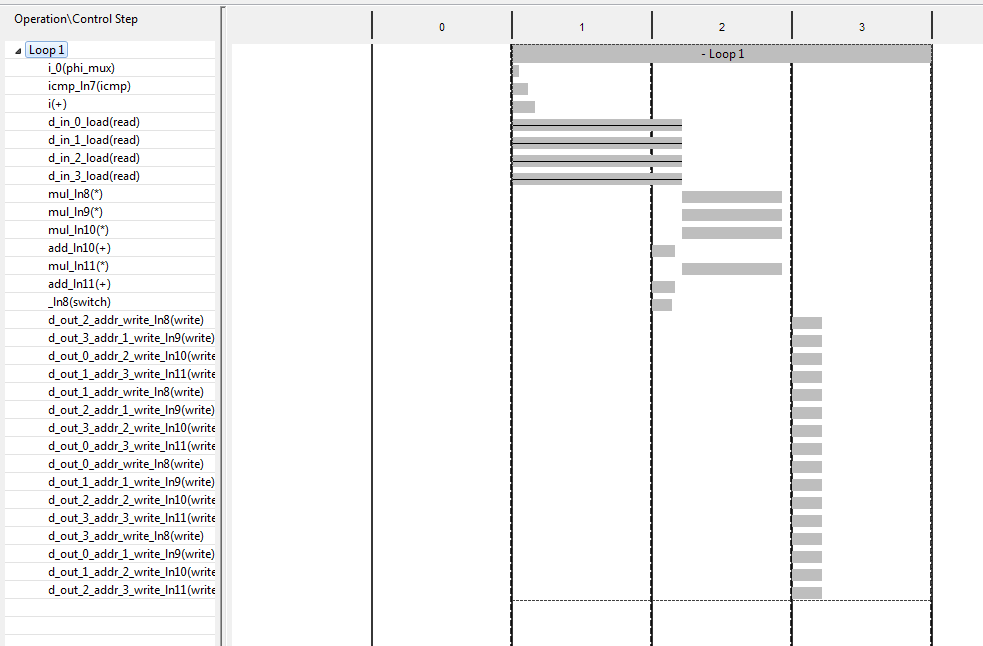


Рисунок 86. Schedule viewer

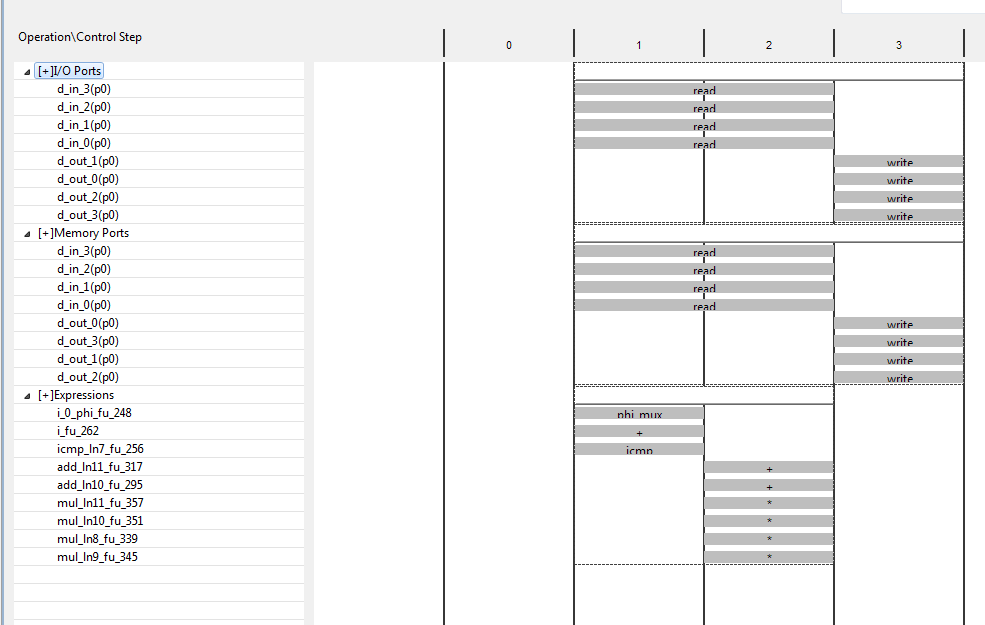


Рисунок 87. Resourse viewer

### 2.5.3 Выводы по решению

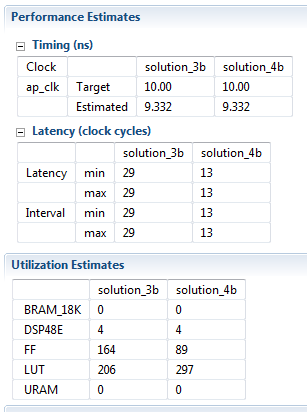


Рисунок 88. Сравнительный отчет

Увеличение количества доступной на запись памяти заметно снизило задержку выполнения не только по сравнению с решением 3b, но так же и решением 2b.Теперь вывод ответа происходит практически одновременно для всех значений, в то время как раньше он чередовался с операциями вычисления.

## 2.6 Решение 5b

### 2.6.1 Параметры решения

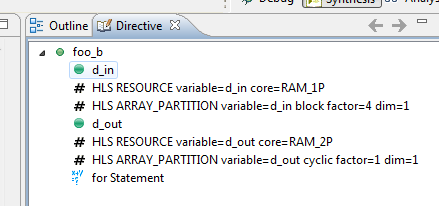


Рисунок 89. Примененные директивы

### 2.6.2 Синтез

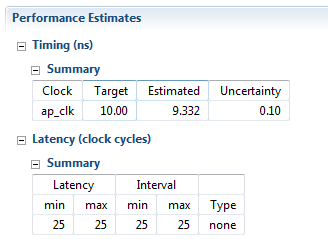


Рисунок 90. Performance estimates – summary

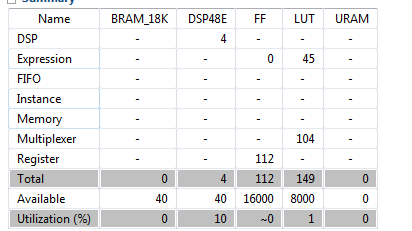


Рисунок 91. Utilization estimates – summary

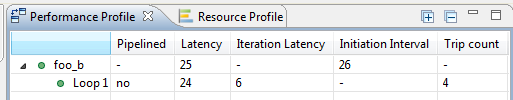


Рисунок 92. Performance profile

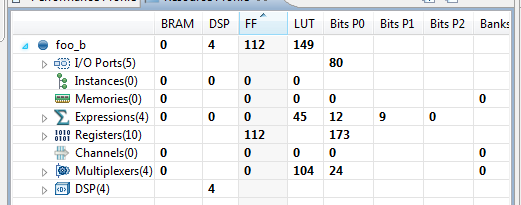


Рисунок 93. Resource profile

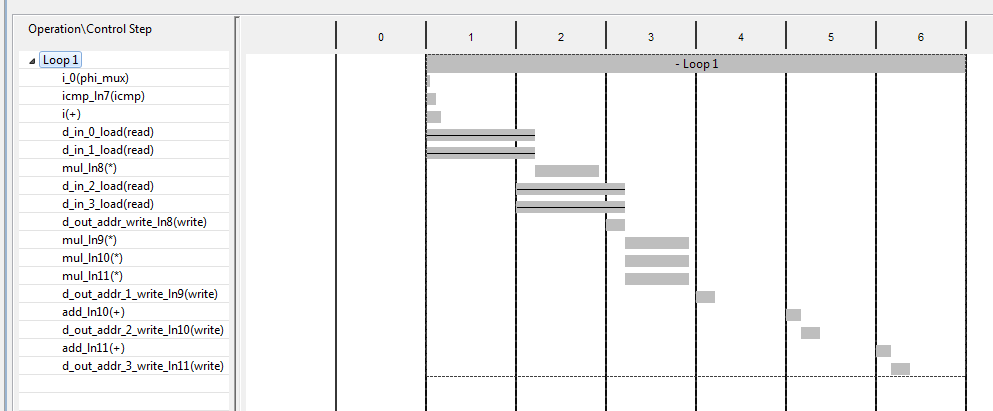


Рисунок 94. Schedule viewer

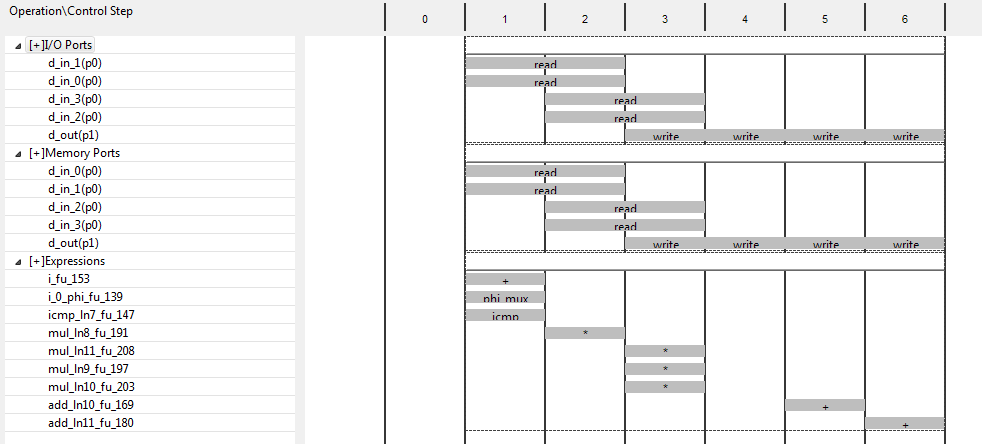


Рисунок 95. Resourse viewer

### 2.6.3 Выводы по решению

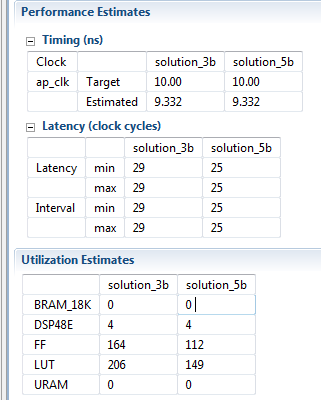


Рисунок 96. Сравнительный отчет

Добавление дополнительного порта для вывода с одновременным уменьшением разбиения массива позволило сократить время выполнения и уменьшить количество используемых ресурсов. Но по сути решение 5b стало аналогично решению 2b, дизайн которого немного проще.

## 2.7 Решение 6b

### 2.7.1 Параметры решения

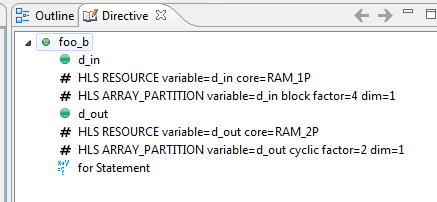


Рисунок 97. Примененные директивы

### 2.7.2 Синтез

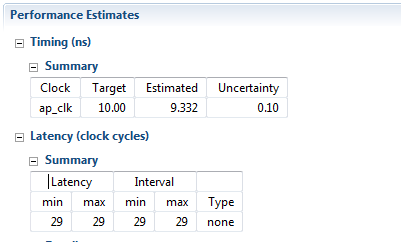


Рисунок 98. Performance estimates – summary

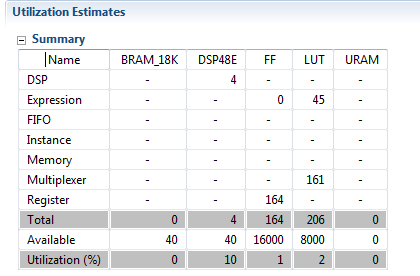


Рисунок 99. Utilization estimates – summary

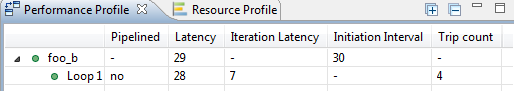


Рисунок 100. Performance profile

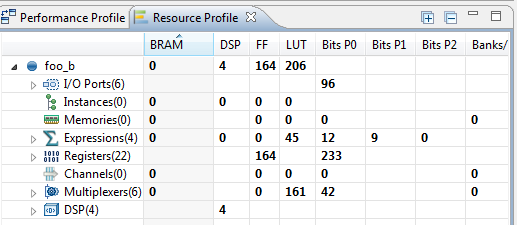


Рисунок 101. Resource profile

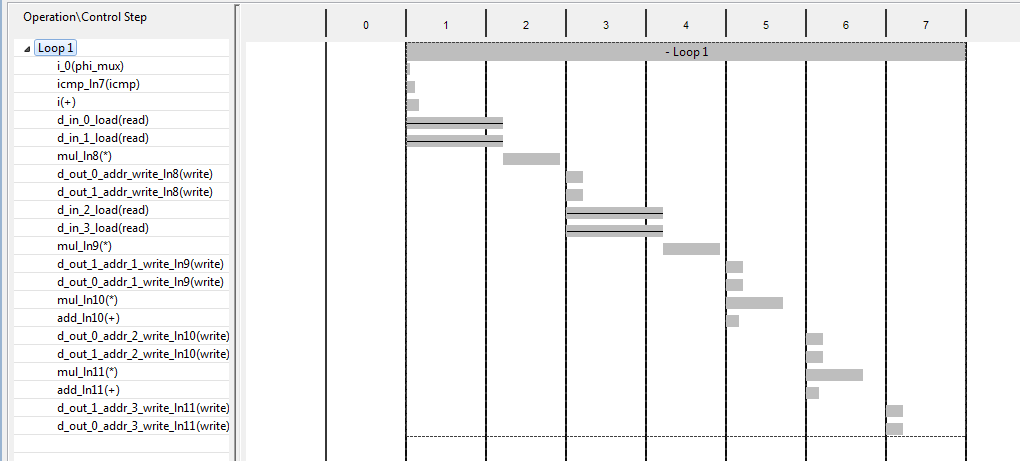


Рисунок 102. Schedule viewer

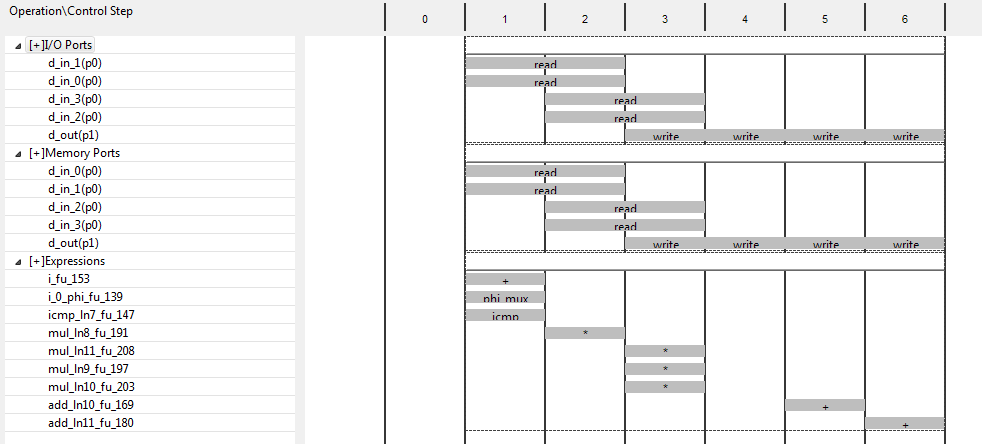


Рисунок 103. Resourse viewer

### 2.7.3 Выводы по решению

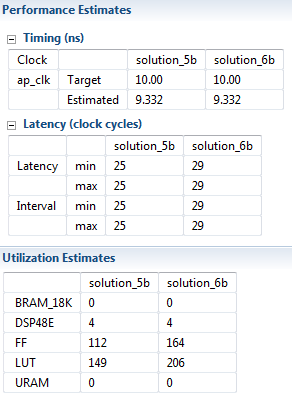


Рисунок 104. Сравнительный отчет

Задержки увеличились с увеличением памяти доступной для записи, т.к. операции умножения теперь происходят вперемешку с операциями записи, в то время как раньше все умножение происходило за 1 такт до начала вывода.

## 2.8 Решение 7b

### 2.8.1 Параметры решения

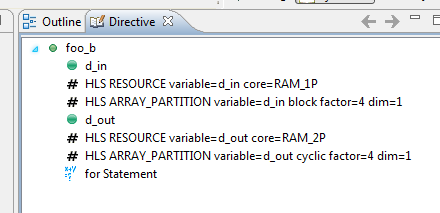


Рисунок 105. Примененные директивы

### 2.8.2 Синтез

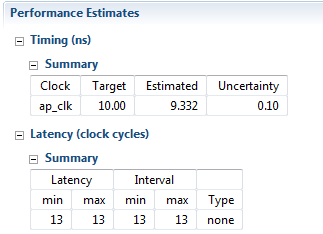


Рисунок 106. Performance estimates – summary

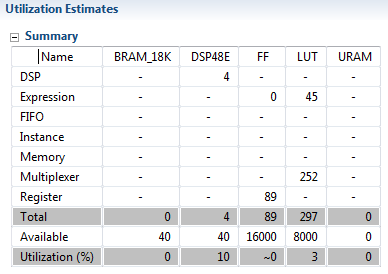


Рисунок 107. Utilization estimates – summary

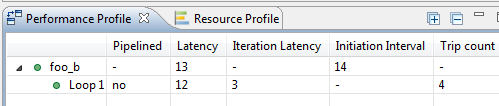


Рисунок 108. Performance profile

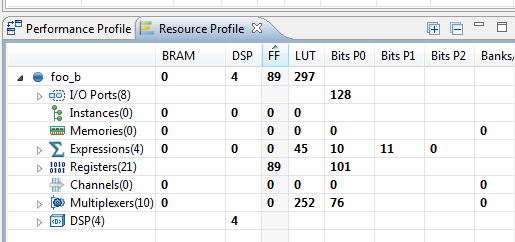


Рисунок 109. Resource profile

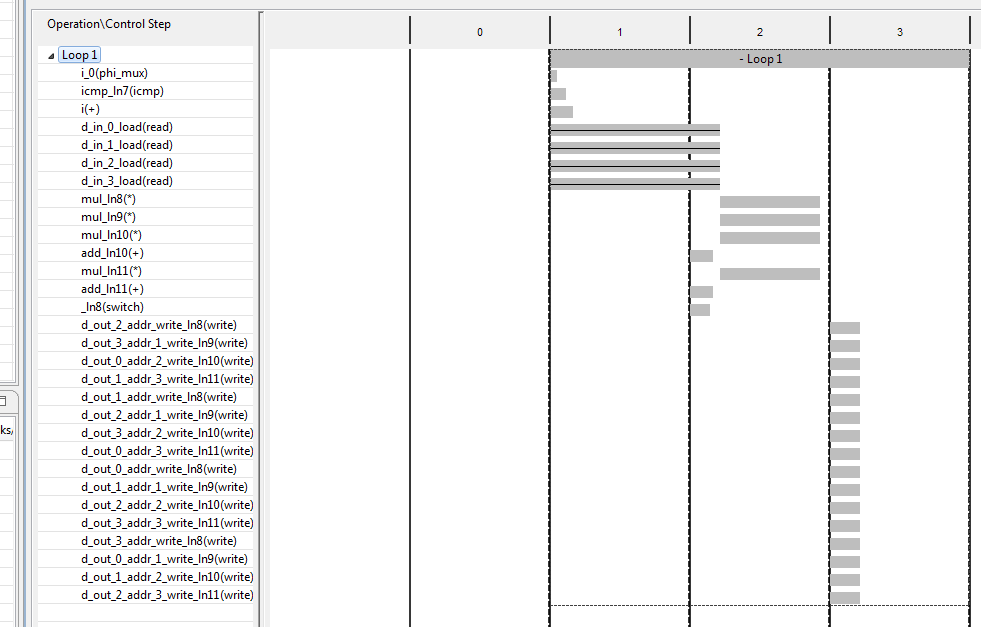


Рисунок 110. Schedule viewer

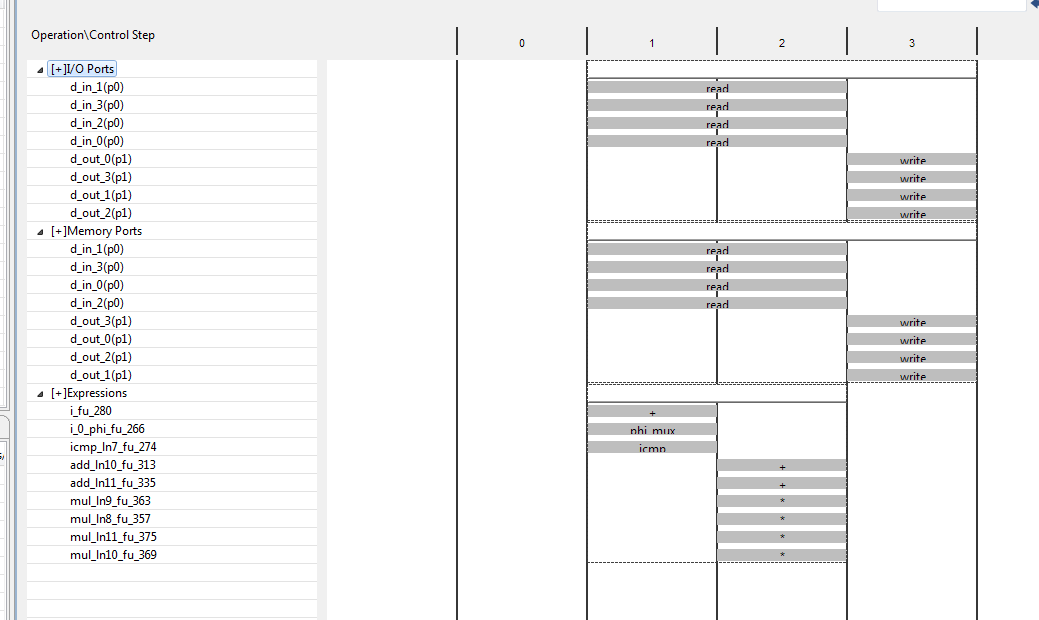


Рисунок 111. Resourse viewer

### 2.8.3 Выводы по решению

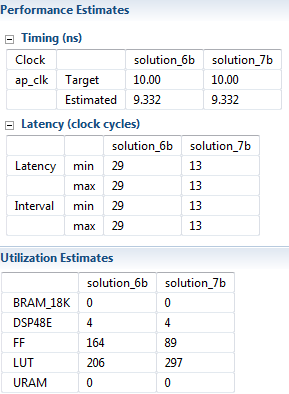


Рисунок 112. Сравнительный отчет

Увеличение количества доступной на запись памяти опять заметно снизило задержку выполнения по сравнению с решением 6b. Вывод ответа происходит практически одновременно для всех значений.

## 2.9 Сравнение решений

|  |  |  |
| --- | --- | --- |
|  | RAM\_1P | RAM\_2P |
| Без cyclic | 2b/25/25 | 5b/25/25 |
| cyclic; factor =2 | 3b/29/29 | 6b/29/29 |
| cyclic; factor =4 | 4b/13/13 | 7b/13/13 |

При разбиении выходной памяти мы не всегда получаем ожидаемый прирост производительности. По временным диаграммам видно, что запись в выходную память не всегда производится параллельно для нескольких ячеек.

## 2.10 Решение 8b

### 2.10.1 Параметры решения

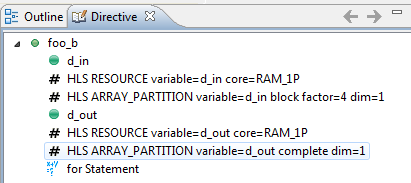


Рисунок 113. Примененные директивы

### 2.10.2 Синтез

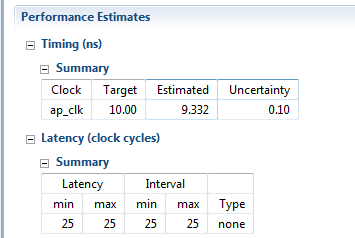


Рисунок 114. Performance estimates – summary

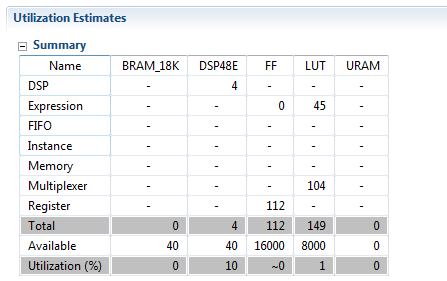


Рисунок 115. Utilization estimates – summary

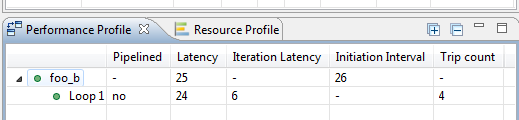


Рисунок 116. Performance profile

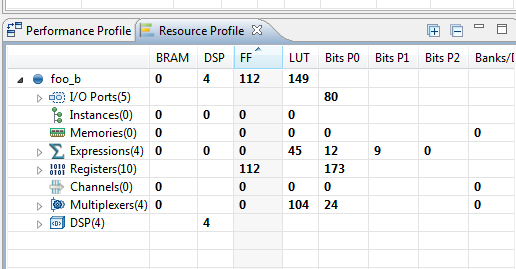


Рисунок 117. Resource profile

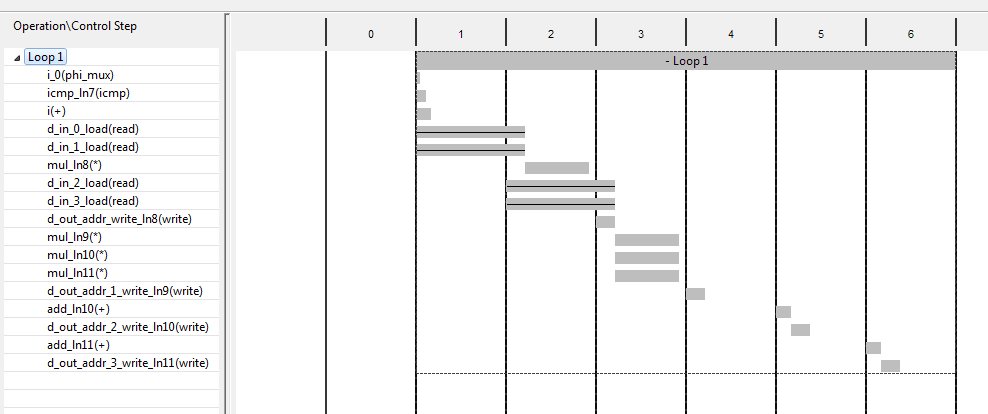


Рисунок 118. Schedule viewer

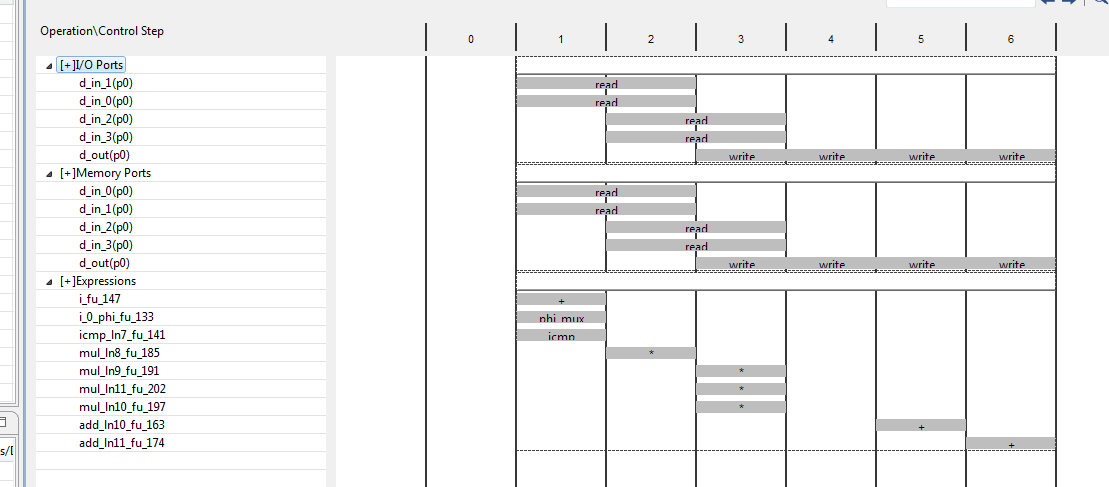


Рисунок 119. Resourse viewer

### 2.10.3 Выводы по решению

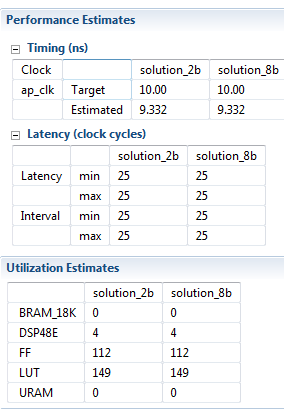


Рисунок 120. Сравнительный отчет

Решения получились идентичные друг другу. Во время синтеза программа проигнорировала разбиение выходного массива.



Рисунок 11. Логи синтеза