

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

**Отчет по лабораторной работе №9\_1**  
**Курс: «Проектирование реконфигурируемых гибридных**  
**вычислительных систем»**  
**Тема: Optimizing array structures**

Выполнил студент гр. 3540901/81501

Селиверстов Я.А.

(подпись)

Руководитель

Антонов А.П.

(подпись)

“    ” \_\_\_\_\_ 2019 г.

Санкт – Петербург  
2019

# ОГЛАВЛЕНИЕ

<b>1. Задание .....</b>	<b>3</b>
1.1 Код программы и теста.....	3
Задание: .....	5
<b>2.Решение №1.....</b>	<b>6</b>
2.1. Синтез.....	6
<b>3. Второе решение.....</b>	<b>7</b>
3.1.Директива.....	7
3.2 Синтез.....	8
3.3. Вывод логов.....	9
<b>4. Решение №3.....</b>	<b>10</b>
4.1 Параметры третьего решения .....	10
4.2 Синтез.....	11
<b>5. Вывод.....</b>	<b>12</b>

# 1. Задание

## 1.1 Код программы и теста

```
#include "dct.h"
void dct_1d(dct_data_t src[DCT_SIZE], dct_data_t dst[DCT_SIZE])
{
    unsigned int k, n;
    int tmp;
    const dct_data_t dct_coeff_table[DCT_SIZE][DCT_SIZE] = {
#include "dct_coeff_table.txt"
    };
DCT_Outer_Loop:
    for (k = 0; k < DCT_SIZE; k++) {
DCT_Inner_Loop:
        for(n = 0, tmp = 0; n < DCT_SIZE; n++) {
            int coeff = (int)dct_coeff_table[k][n];
            tmp += src[n] * coeff;
        }
        dst[k] = DESCALE(tmp, CONST_BITS);
    }
}
void dct_2d(dct_data_t in_block[DCT_SIZE][DCT_SIZE],
            dct_data_t out_block[DCT_SIZE][DCT_SIZE])
{
    dct_data_t row_outbuf[DCT_SIZE][DCT_SIZE];
    dct_data_t col_outbuf[DCT_SIZE][DCT_SIZE],
col_inbuf[DCT_SIZE][DCT_SIZE];
    unsigned i, j;
    // DCT rows
Row_DCT_Loop:
    for(i = 0; i < DCT_SIZE; i++) {
        dct_1d(in_block[i], row_outbuf[i]);
    }
    // Transpose data in order to re-use 1D DCT code
Xpose_Row_Outer_Loop:
    for (j = 0; j < DCT_SIZE; j++)
Xpose_Row_Inner_Loop:
        for(i = 0; i < DCT_SIZE; i++)
            col_inbuf[j][i] = row_outbuf[i][j];
    // DCT columns
Col_DCT_Loop:
    for (i = 0; i < DCT_SIZE; i++) {
        dct_1d(col_inbuf[i], col_outbuf[i]);
    }
    // Transpose data back into natural order
Xpose_Col_Outer_Loop:
    for (j = 0; j < DCT_SIZE; j++)
Xpose_Col_Inner_Loop:
```

```

        for(i = 0; i < DCT_SIZE; i++)
            out_block[j][i] = col_outbuf[i][j];
    }
void read_data(short input[N], short buf[DCT_SIZE][DCT_SIZE])
{
    int r, c;
RD_Loop_Row:
    for (r = 0; r < DCT_SIZE; r++) {
RD_Loop_Col:
        for (c = 0; c < DCT_SIZE; c++)
            buf[r][c] = input[r * DCT_SIZE + c];
    }
}
void write_data(short buf[DCT_SIZE][DCT_SIZE], short output[N])
{
    int r, c;
WR_Loop_Row:
    for (r = 0; r < DCT_SIZE; r++) {
WR_Loop_Col:
        for (c = 0; c < DCT_SIZE; c++)
            output[r * DCT_SIZE + c] = buf[r][c];
    }
}
void dct(short input[N], short output[N])
{
    short buf_2d_in[DCT_SIZE][DCT_SIZE];
    short buf_2d_out[DCT_SIZE][DCT_SIZE];
    // Read input data. Fill the internal buffer.
    read_data(input, buf_2d_in);

    dct_2d(buf_2d_in, buf_2d_out);

    // Write out the results.
    write_data(buf_2d_out, output);
}

#ifndef __DCT_H__
#define __DCT_H__

#define DW 16
#define N 1024/DW
#define NUM_TRANS 16

typedef short dct_data_t;

#define DCT_SIZE 8 /* defines the input matrix as 8x8 */
#define CONST_BITS 13
#define DESCALE(x,n) (((x) + (1 << ((n)-1))) >> n)
void dct(short input[N], short output[N]);
#endif // __DCT_H__ not defined

```

## Задание:

- Launch the Vivado® HLS tool.  
Open the provided **dct\_prj** Vivado HLS tool project located at:  
*C:\training\optimize\_array\_performance\demo\dct\_prj*
- Access and review the source files (*dct.c* and *dct.h*) from the Explorer pane.
- Run C synthesis.
- Review the Synthesis report.
- Create a new solution named *solution2* (select **Project** > **New Solution**).
- Accept the default settings and click **Finish**.
- Apply the **PIPELINE** directive on *DCT\_Outer\_Loop* of the *dct\_1d* function (shown below).
- Similarly, apply the **PIPELINE** directive to the following loops:
  - *Xpose\_Row\_Inner\_Loop* of the *dct\_2d* function
  - *Xpose\_Col\_Inner\_Loop* of the *dct\_2d* function
  - *RD\_Loop\_Col* of the *read\_data* function
- *WR\_Loop\_Col* of the *write\_data* function
- Compare the results of two solutions (*solution1* and *solution2*).
- What is the worst-case latency of the design?
- Go to the Utilization Estimates section and note the number of DSP48E and block RAMs used to implement *solution2*.
- Select the **Console** tab and review the synthesis information.
- Create a new solution named *solution3*.
- Accept the default settings and click **Finish**.
- Apply the **ARRAY\_PARTITION** directive to *buf\_2d\_in* of the *dct* function as shown in the figure below.
- Similarly, apply the **ARRAY\_PARTITION** directive *col\_inbuf* of the *dct\_2d* function.
- Run C synthesis.
- Compare the results of the two solutions (*solution2* and *solution3*).

## 2.Решение №1.

### 2.1. Синтез

Результаты синтеза с оценкой производительности и используемых ресурсов представлены на рисунках 2.1 и 2.2 соответственно.

Performance Estimates

[-] Timing (ns)

[-] Summary

Clock	Target	Estimated	Uncertainty
ap_clk	6.00	5.790	1.25

[-] Latency (clock cycles)

[-] Summary

Latency		Interval		
min	max	min	max	Type
4215	4215	4215	4215	none

Рисунок 2.1. Отчет о синтезе

Utilization Estimates	
	solution1
BRAM_18K	5
DSP48E	1
FF	346
LUT	994
URAM	0

Рисунок 2.2. Отчет об используемых ресурсах

Величина Latency составила 4215.

Количество ресурсов в проекте составило 5 блоков BRAM\_18K и один блок DSP48E.

## 3. Второе решение

### 3.1. Директива

Пропишем директиву PIPELINE, рисунок 3.1.

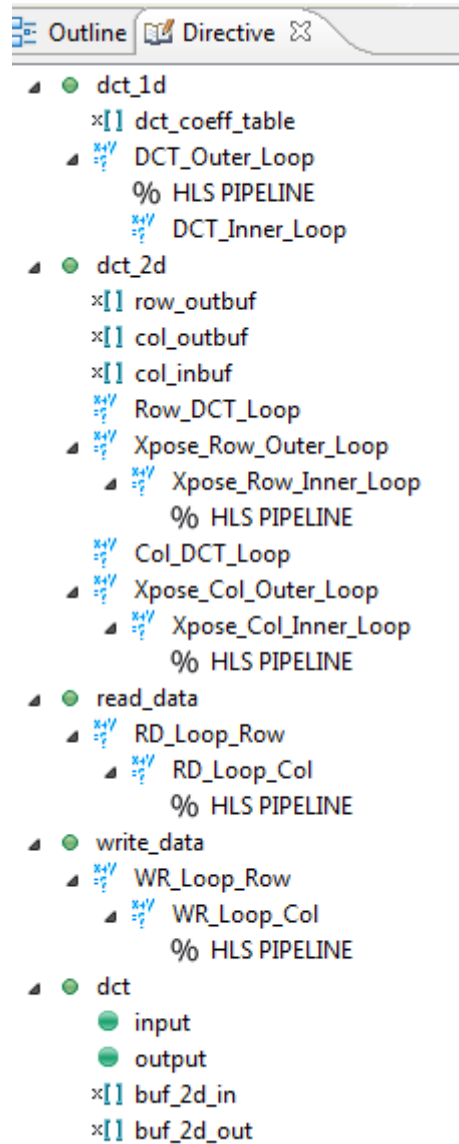


Рисунок 3.1. Параметры второго решения

## 3.2 Синтез

Результаты синтеза с оценкой производительности и сравнительном анализе используемых ресурсов первого и второго решения представлены на рисунках 3.2 и 3.3 соответственно.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	6.00	5.790	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
899	899	899	899	none

Рисунок 3.1. Отчет о синтезе

Величина Latency составила 899.

Performance Estimates

Timing (ns)

Clock		solution1	solution2
ap_clk	Target	6.00	6.00
	Estimated	5.790	5.790

Latency (clock cycles)

		solution1	solution2
Latency	min	4215	899
	max	4215	899
Interval	min	4215	899
	max	4215	899

Utilization Estimates

	solution1	solution2
BRAM_18K	5	5
DSP48E	1	8
FF	346	1476
LUT	994	1735
URAM	0	0

Рисунок 2.3. Сравнительный отчет первого и второго решений

В проекте используется 5 блоков BRAM и 8 блоков DSP.



### 3.3. Вывод логов.

```
INFO: [SCHED 204-61] Pipelining loop 'DCT_Outer_Loop'.
WARNING: [SCHED 204-69] Unable to schedule 'load' operation ('src_load_5', dct.c:17) on array 'src' due to limited memory ports. Please consider using a memory core wit
INFO: [SCHED 204-61] Pipelining result : Target II = 1, Final II = 4, Depth = 8.
WARNING: [SCHED 204-21] Estimated clock period (5.79ns) exceeds the target (target clock period: 6ns, clock uncertainty: 1.25ns, effective delay budget: 4.75ns).
WARNING: [SCHED 204-21] The critical path in module 'dct_id2' consists of the following:
'mul' operation of DSP[109] ('mul_ln17_7', dct.c:17) [104] (2.84 ns)
'add' operation of DSP[109] ('add_ln19_5', dct.c:19) [109] (2.95 ns)
INFO: [SCHED 204-11] Finished scheduling.
INFO: [HLS 200-111] Elapsed time: 10.486 seconds; current allocated memory: 113.710 MB.
INFO: [BIND 205-100] Starting micro-architecture generation ...
INFO: [BIND 205-101] Performing variable lifetime analysis.
INFO: [BIND 205-101] Exploring resource sharing.
INFO: [BIND 205-101] Binding ...
INFO: [BIND 205-100] Finished micro-architecture generation.
INFO: [HLS 200-111] Elapsed time: 0.098 seconds; current allocated memory: 114.073 MB.
INFO: [HLS 200-10] -----
INFO: [HLS 200-42] -- Implementing module 'dct_2d'
INFO: [HLS 200-10] -----
INFO: [SCHED 204-11] Starting scheduling ...
INFO: [SCHED 204-61] Pipelining loop 'Xpose_Row_Outer_Loop_Xpose_Row_Inner_Loop'.
INFO: [SCHED 204-61] Pipelining result : Target II = 1, Final II = 1, Depth = 4.
INFO: [SCHED 204-61] Pipelining loop 'Xpose_Col_Outer_Loop_Xpose_Col_Inner_Loop'.
INFO: [SCHED 204-61] Pipelining result : Target II = 1, Final II = 1, Depth = 4.
INFO: [SCHED 204-11] Finished scheduling.
INFO: [HLS 200-111] Elapsed time: 0.19 seconds; current allocated memory: 114.304 MB.
INFO: [HLS 200-434] Only 2 loops out of a total 4 loops have been pipelined in this design.
INFO: [BIND 205-100] Starting micro-architecture generation ...
INFO: [BIND 205-101] Performing variable lifetime analysis.
INFO: [BIND 205-101] Exploring resource sharing.
INFO: [BIND 205-101] Binding ...
INFO: [BIND 205-100] Finished micro-architecture generation.
INFO: [HLS 200-111] Elapsed time: 0.116 seconds; current allocated memory: 114.647 MB.
INFO: [HLS 200-10] -----
INFO: [HLS 200-42] -- Implementing module 'dct'
INFO: [HLS 200-10] -----
INFO: [SCHED 204-11] Starting scheduling ...
INFO: [SCHED 204-61] Pipelining loop 'RD_Loop_Row_RD_Loop_Col'.
INFO: [SCHED 204-61] Pipelining result : Target II = 1, Final II = 1, Depth = 4.
INFO: [SCHED 204-61] Pipelining loop 'WR_Loop_Row_WR_Loop_Col'.
INFO: [SCHED 204-61] Pipelining result : Target II = 1, Final II = 1, Depth = 4.
INFO: [SCHED 204-11] Finished scheduling.
```

Рисунок 3.4. Окно Console

Как видно из сообщения, не удалось достичь требуемого значения Initiation interval для DCT\_Outer\_Loop в связи с нехваткой портов памяти. Для достижения требуемой пропускной способности нужно разбить массивы col\_inbuf и in\_block.

## 4. Решение №3.

### 4.1 Параметры третьего решения

Пропишем соответствующие директивы, рисунок 4.1.

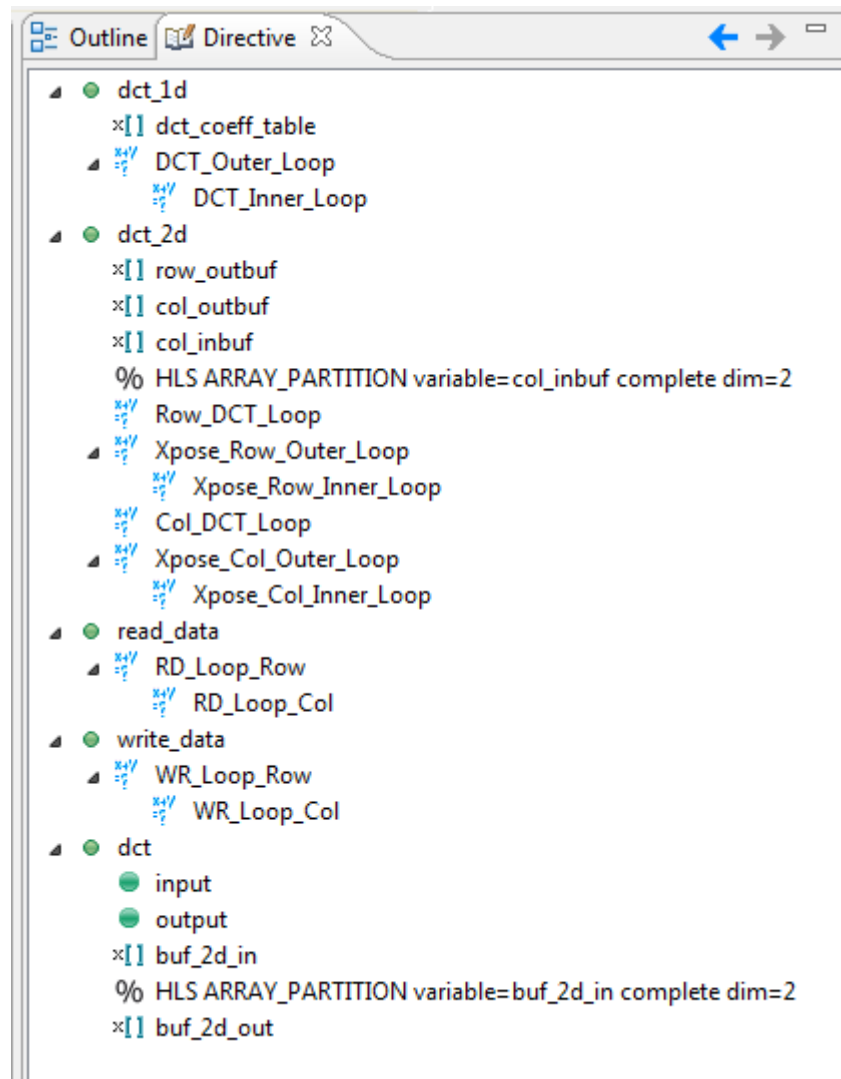


Рисунок 4.1. Параметры третьего решения

## 4.2 Синтез

Результаты сравнительного анализа результатов синтеза третьего решения с первым и вторым и сравнительными оценками производительности и используемых ресурсов представлены на рисунках 4.2.

Performance Estimates				
▢ Timing (ns)				
Clock		solution1	solution2	solution3
ap_clk	Target	6.00	6.00	6.00
	Estimated	5.790	5.790	5.790
▢ Latency (clock cycles)				
		solution1	solution2	solution3
Latency	min	4215	899	531
	max	4215	899	531
Interval	min	4215	899	531
	max	4215	899	531
Utilization Estimates				
		solution1	solution2	solution3
BRAM_18K		5	5	3
DSP48E		1	8	8
FF		346	1476	1839
LUT		994	1735	2143
URAM		0	0	0

Рисунок 4.2. Сравнительный отчет трех решений

Latency в третьем решении достигла наименьшей величины из трех решений и составила 531.

В проекте используется 3 блока BRAM и 8 блоков DSP.

## 5. Вывод

Значение Latency уменьшилось с 4215 до 531 при использовании директивы для разбиения массивов что поспособствовало успешному применению директивы PIPELINE и повышению пропускной способности проекта.