

Санкт-Петербургский Политехнический Университет Петра Великого
Институт Компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Лабораторная работа 7 Задание 1

Предмет: «Проектирование реконфигурируемых гибридных вычислительных систем»

Тема: «Pipelining for Performance Demo Script»

Студент: Ерниязов Т.Е.
Гр. № 3540901/81501

Преподаватель: Антонов А.П.

Санкт-Петербург
2019

Оглавление

1.	Первое решение	3
1.1.	Открытие проекта	3
1.2.	Моделирование	6
1.3.	Производительность	6
1.4.	Использование ресурсов	7
2.	Второе решение.....	8
2.1.	Создание решения и применение директивы Pipeline.....	8
2.2.	Синтез	9
2.3.	Производительность.....	9
2.4.	Сравнение решений.....	9
2.5.	Использование ресурсов	10
3.	Третье решение	11
3.1.	Изменение директив.....	11
3.2.	Синтез	11
3.3.	Сравнение решений.....	12
4.	Выводы	12

1. Первое решение

1.1. Открытие проекта

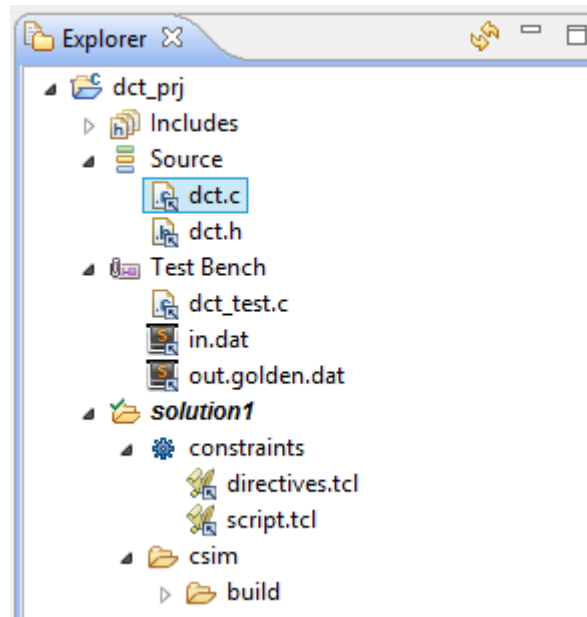


Рис. 1.1. Explorer opened project

Файл исходного кода dct.c:

```
#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);
}

```

Файл-хедер dct.h:

```

#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

```

Файл с тестами dct_test.c:

```

#include <stdio.h>
#include "dct.h"
// *****
int main() {
    short a[N], b[N], b_expected[N];

```

```

int retval = 0, i;
FILE *fp;

fp=fopen("in.dat", "r");
for (i=0; i<N; i++){
    int tmp;
    fscanf(fp, "%d", &tmp);
    a[i] = tmp;
}
fclose(fp);

fp=fopen("out.golden.dat", "r");
for (i=0; i<N; i++){
    int tmp;
    fscanf(fp, "%d", &tmp);
    b_expected[i] = tmp;
}
fclose(fp);

dct(a, b);

for (i = 0; i < N; ++i) {
    if(b[i] != b_expected[i]){
        printf("Incorrect output on sample %d. Expected %d, Received %d \n", i, b_expected[i], b[i]);
        retval = 2;
    }
}

#if 0 // Optionally write out computed values
fp=fopen("out.dat", "w");
for (i=0; i<N; i++){
    fprintf(fp, "%d\n", b[i]);
}
fclose(fp);
#endif

if(retval != (2)){
    printf(" *** ** \n");
    printf(" Results are good \n");
    printf(" *** ** \n");
} else {
    printf(" *** ** \n");
    printf(" BAD!! %d \n", retval);
    printf(" *** ** \n");
}
return retval;
}

```

Данный исходный файл использует дискретное косинусное преобразование (DCT). Функция реализует алгоритм 2D DCT, сначала обрабатывая каждую строку входного массива через 1D DCT, затем обрабатывая столбцы результирующего массива через тот же 1D DCT. Он вызывает функции `read_data`, `dct_2d` и `write_data`. Функция `read_data` состоит из двух циклов: `RD_Loop_Row` и `RD_Loop_Col`. Функция `write_data` состоит из двух циклов для выполнения записи результата. Функция `dct_2d`, вызывает функцию `dct_1d` и выполняет транспонирование. Наконец, функция `dct_1d`, использует `dct_coeff_table` и выполняет требуемую функцию, реализуя базовую итеративную форму алгоритма DCD 1D типа II.

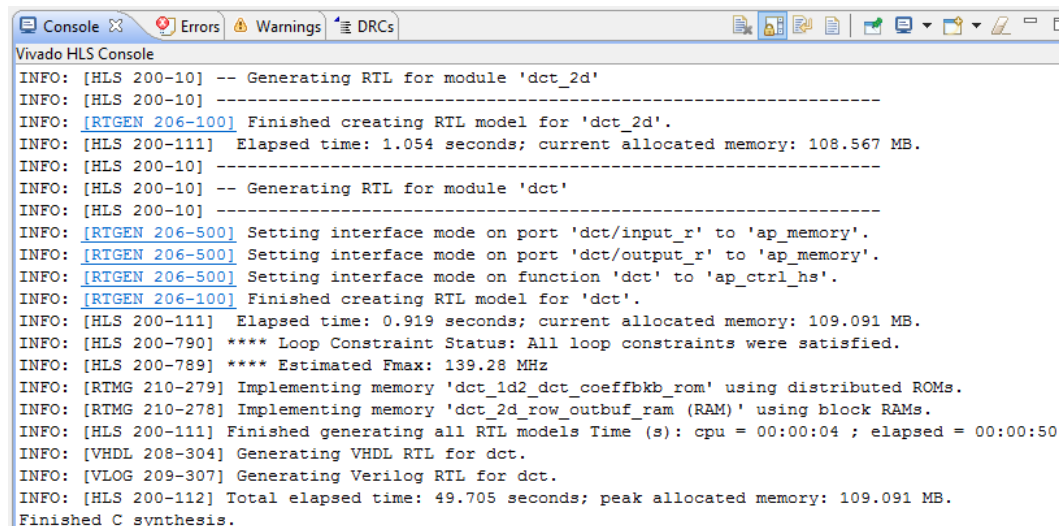
1.2. Моделирование



```
Console Errors Warnings DRCs
Vivado HLS Console
Compiling(apcc) ../../../../dct_test.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 version 6.2) on Mon Dec 09 0
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab7/lab7_z1/
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Compiling(apcc) ../../../../dct.c in debug mode
INFO: [HLS 200-10] Running 'C:/Xilinx/Vivado/2019.2/bin/unwrapped/win64.o/apcc.exe'
INFO: [HLS 200-10] For user 'Misha' on host 'mikhail' (Windows NT_amd64 version 6.2) on Mon Dec 09 0
INFO: [HLS 200-10] In directory 'C:/Users/Misha/Desktop/university/___maga/3sem/antonov/lab7/lab7_z1/
INFO: [APCC 202-3] Tmp directory is apcc_db
INFO: [APCC 202-1] APCC is done.
Generating csim.exe
*** **
Results are good
*** **
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.
```

Рис. 1.2. Successful result of modeling

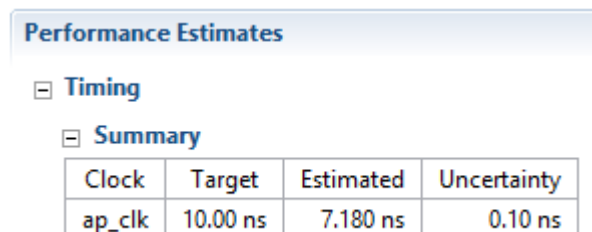
1.3. Синтез



```
Console Errors Warnings DRCs
Vivado HLS Console
INFO: [HLS 200-10] -- Generating RTL for module 'dct_2d'
INFO: [HLS 200-10] -----
INFO: [RTGEN 206-100] Finished creating RTL model for 'dct_2d'.
INFO: [HLS 200-111] Elapsed time: 1.054 seconds; current allocated memory: 108.567 MB.
INFO: [HLS 200-10] -----
INFO: [HLS 200-10] -- Generating RTL for module 'dct'
INFO: [HLS 200-10] -----
INFO: [RTGEN 206-500] Setting interface mode on port 'dct/input_r' to 'ap_memory'.
INFO: [RTGEN 206-500] Setting interface mode on port 'dct/output_r' to 'ap_memory'.
INFO: [RTGEN 206-500] Setting interface mode on function 'dct' to 'ap_ctrl_hs'.
INFO: [RTGEN 206-100] Finished creating RTL model for 'dct'.
INFO: [HLS 200-111] Elapsed time: 0.919 seconds; current allocated memory: 109.091 MB.
INFO: [HLS 200-790] **** Loop Constraint Status: All loop constraints were satisfied.
INFO: [HLS 200-789] **** Estimated Fmax: 139.28 MHz
INFO: [RTMG 210-279] Implementing memory 'dct_1d2_dct_coeffbkb_rom' using distributed ROMs.
INFO: [RTMG 210-278] Implementing memory 'dct_2d_row_outbuf_ram (RAM)' using block RAMs.
INFO: [HLS 200-111] Finished generating all RTL models Time (s): cpu = 00:00:04 ; elapsed = 00:00:50
INFO: [VHDL 208-304] Generating VHDL RTL for dct.
INFO: [VLOG 209-307] Generating Verilog RTL for dct.
INFO: [HLS 200-112] Total elapsed time: 49.705 seconds; peak allocated memory: 109.091 MB.
Finished C synthesis.
```

Рис. 1.3. Successful result of synthesis

1.4. Производительность



Performance Estimates			
Timing			
Summary			
Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	7.180 ns	0.10 ns

Рис. 1.4. Performance estimates

Полученная величина задержки укладывается в заданное значение.

1.5. Использование ресурсов

Utilization Estimates				
Summary				
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	156
FIFO	-	-	-	-
Instance	3	1	209	677
Memory	2	-	0	0
Multiplexer	-	-	-	149
Register	-	-	69	-
Total	5	1	278	982
Available	40	40	16000	8000
Utilization (%)	12	2	1	12

Рис. 1.5.1. Utilization estimates

Module Hierarchy									
	Negative Slack	BRAM	DSP	FF	LUT	Latency	Interval	Pipeline type	
dct	-	5	1	278	982	3959	3960	none	
dct_2d	-	3	1	209	677	3668	3668	none	
dct_1d2	-	0	1	117	204	209	209	none	

Рис. 1.5.2. Module hierarchy

Resource Profile										
	BRAM	DSP	FF	LUT	Bits P0	Bits P1	Bits P2	Banks/Depth	Words	W*Bits*Banks
dct	5	1	278	982						
I/O Ports(2)					32					
Instances(1)	3	1	209	677						
Memories(2)	2		0	0	32			2	128	2048
Expressions(12)	0	0	0	156	60	52	0			
Registers(16)			69		83					
Channels(0)	0		0	0	0			0	0	0
Multiplexers(10)	0		0	149	32			0		
DSP(1)		0								

Рис. 1.5.3 Resource profile

Данное решение потребует на микросхеме 982 элемента LUT, 278 регистров, 1 DSP48E и 5 блоков RAM_18K.

Performance Profile						
	Pipelined	Latency	Iteration Latency	Initiation Interval	Trip count	
dct	-	3959	-	3960	-	
RD_Loop_Row	no	144	18	-	8	
RD_Loop_Col	no	16	2	-	8	
WR_Loop_Row	no	144	18	-	8	
WR_Loop_Col	no	16	2	-	8	

Рис. 1.5.4. Performance Profile

Задержка составляет 3959 тактов, интервал инициализации – 3960.

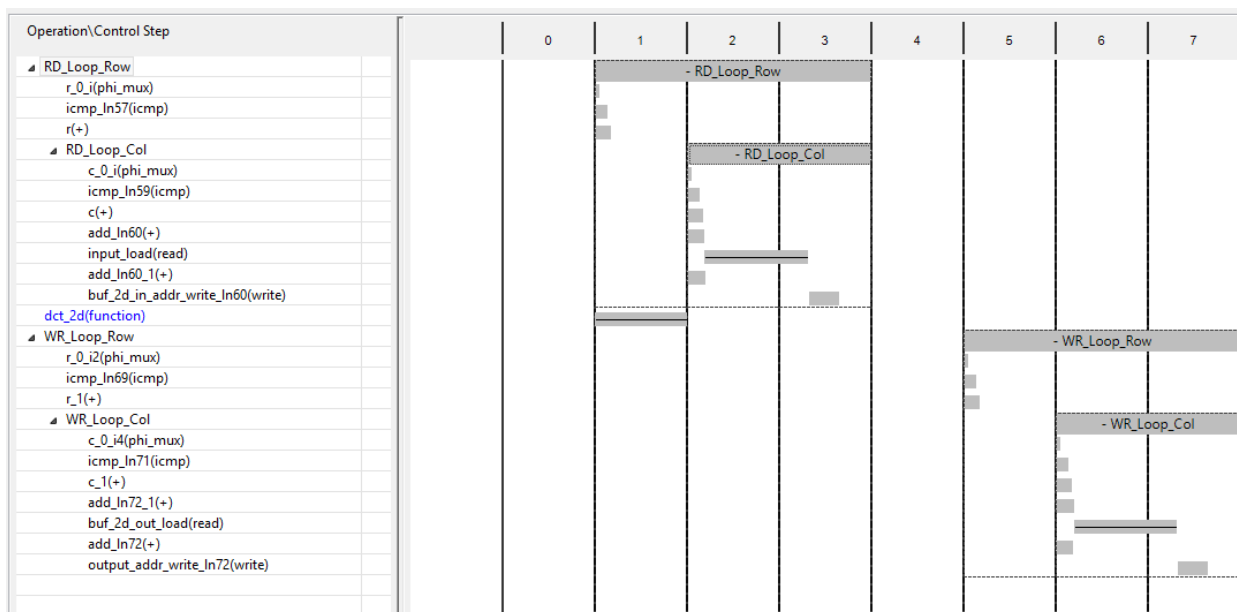


Рис. 1.5.5. Operation/Control Step

Максимальная величина задержки для dct_2d равна 3668. Максимальная величина задержки для dct_1d2 равна 209.

2. Второе решение

2.1. Создание решения и применение директивы Pipeline

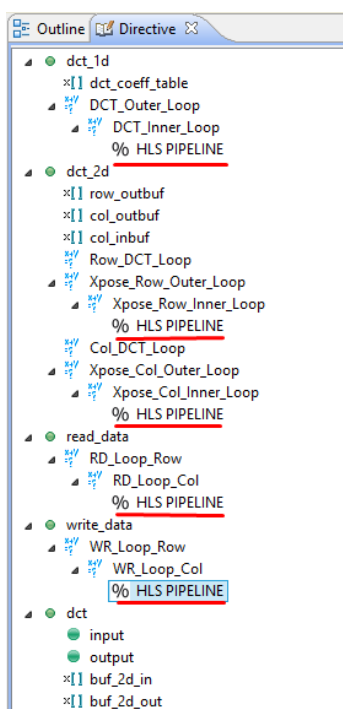


Рис.2.1 Directive Pipeline

2.2. Синтез

```
Vivado HLS Console
INFO: [RTGEN 206-100] Generating core module 'dct_mac_muladd_15cud': 1 instance(s).
INFO: [RTGEN 206-100] Finished creating RTL model for 'dct_1d2'.
INFO: [HLS 200-111] Elapsed time: 0.368 seconds; current allocated memory: 108.598 MB.
INFO: [HLS 200-10] -----
INFO: [HLS 200-10] -- Generating RTL for module 'dct_2d'
INFO: [HLS 200-10] -----
INFO: [RTGEN 206-100] Finished creating RTL model for 'dct_2d'.
INFO: [HLS 200-111] Elapsed time: 0.598 seconds; current allocated memory: 109.292 MB.
INFO: [HLS 200-10] -----
INFO: [HLS 200-10] -- Generating RTL for module 'dct'
INFO: [HLS 200-10] -----
INFO: [RTGEN 206-500] Setting interface mode on port 'dct/input_r' to 'ap_memory'.
INFO: [RTGEN 206-500] Setting interface mode on port 'dct/output_r' to 'ap_memory'.
INFO: [RTGEN 206-500] Setting interface mode on function 'dct' to 'ap_ctrl_hs'.
INFO: [RTGEN 206-100] Finished creating RTL model for 'dct'.
INFO: [HLS 200-111] Elapsed time: 0.643 seconds; current allocated memory: 109.982 MB.
INFO: [HLS 200-790] **** Loop Constraint Status: All loop constraints were satisfied.
INFO: [HLS 200-789] **** Estimated Fmax: 124.15 MHz
INFO: [RTMG 210-279] Implementing memory 'dct_1d2_dct_coeffbkb_rom' using distributed RO
INFO: [RTMG 210-278] Implementing memory 'dct_2d_row_outbuf_ram (RAM)' using block RAMs.
INFO: [HLS 200-111] Finished generating all RTL models Time (s): cpu = 00:00:05 ; elapse
INFO: [VHDL 208-304] Generating VHDL RTL for dct.
INFO: [VLOG 209-307] Generating Verilog RTL for dct.
INFO: [HLS 200-112] Total elapsed time: 20.795 seconds; peak allocated memory: 109.982 M
Finished C synthesis.
```

Рис. 2.2. Successful result of synthesis

2.3. Производительность

Performance Estimates			
Timing			
Summary			
Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.054 ns	0.10 ns

Рис. 2.3 Performance estimates

Величина задержки укладывается в заданное значение.

2.4. Сравнение решений

Performance Estimates			
Timing			
Clock		solution1	solution2
ap_clk	Target	10.00 ns	10.00 ns
	Estimated	7.180 ns	8.054 ns
Latency			
		solution1	solution2
Latency (cycles)	min	3959	1851
	max	3959	1851
Latency (absolute)	min	39.590 us	18.510 us
	max	39.590 us	18.510 us
Interval (cycles)	min	3959	1851
	max	3959	1851

Рис. 2.4 Comparing solutions

На изображении выше видно значительное уменьшение количества тактов в задержке получения решения и интервале инициализации.

Максимальное значение задержки для второго решения составляет 1851 такт.

2.5. Использование ресурсов

Utilization Estimates					
Summary					
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	210	-
FIFO	-	-	-	-	-
Instance	3	1	195	807	0
Memory	2	-	0	0	0
Multiplexer	-	-	-	209	-
Register	-	-	61	-	-
Total	5	1	256	1226	0
Available	40	40	16000	8000	0
Utilization (%)	12	2	1	15	0

Рис. 2.4. Utilization estimates

Данное решение потребует на микросхеме 1226 элементов LUT, 256 регистров и как и в первом решении 1 DSP48E и 5 блоков RAM_18K.

Максимальное значение задержки для dct_2d равно 1718. Максимальное значение задержки для dct_1d2 равно 97.

3. Третье решение

3.1. Изменение директив

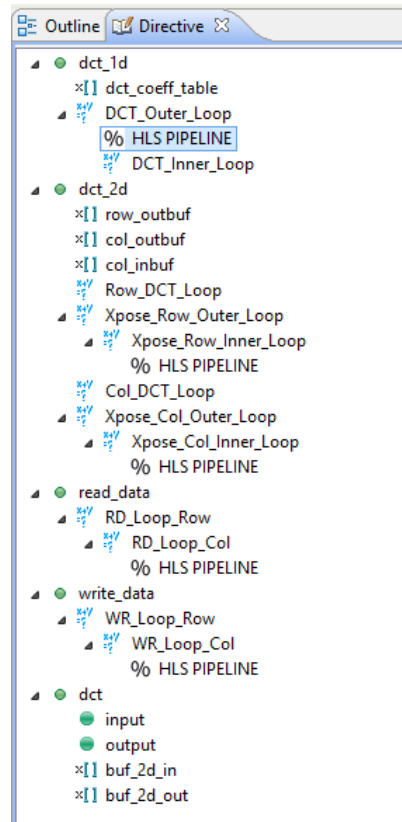


Рис. 3.1. Changing a directive

3.2. Синтез

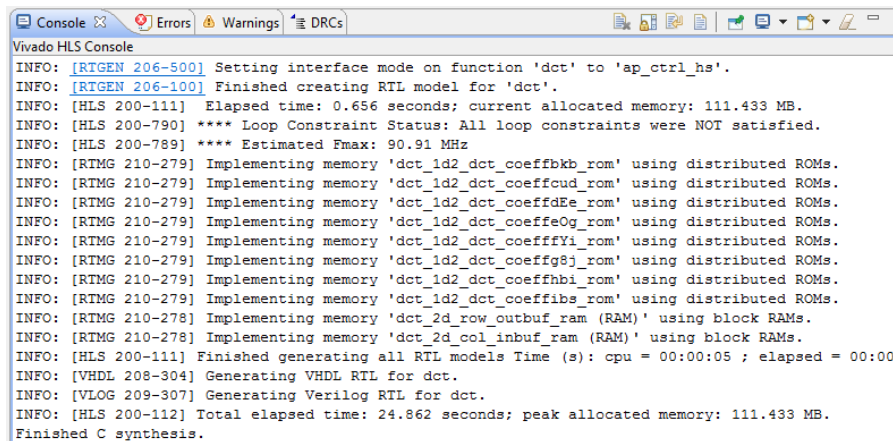


Рис. 3.2. Successful result of synthesis

3.3. Сравнение решений

Performance Estimates			
⊟ Timing			
Clock		solution3	solution2
ap_clk	Target	10.00 ns	10.00 ns
	Estimated	11.000 ns	8.054 ns
⊟ Latency			
		solution3	solution2
Latency (cycles)	min	875	1851
	max	875	1851
Latency (absolute)	min	9.625 us	18.510 us
	max	9.625 us	18.510 us
Interval (cycles)	min	875	1851
	max	875	1851

Рис. 3.2. Comparing solutions

На изображении выше видно, что время задержки третьего решения значительно выше, но при этом, количество тактов практически в 2,5 раза меньше.

4. Выводы

В ходе выполнения работы было получено три решения и проведено сравнение величин задержек в этих решениях. Директива PIPELINE встраивает конвейерные регистры в сгенерированный RTL, чтобы улучшить значение задержки. При правильном применении этой директивы есть возможность увеличить пропускную способность проекта.