



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

---

Институт Информационных Технологий  
Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4**

«Основы языка команд Tcl»

по дисциплине

«Схемотехника устройств компьютерных систем»

Выполнил студент группы  
ИВБО-08-22

Стецюк В.В.

Принял ассистент кафедры ВТ

Дуксин Н. А.

Практическая работа выполнена

« \_\_ » \_\_\_\_\_ 2024 г.

«Зачтено»

« \_\_ » \_\_\_\_\_ 2024 г.

Москва 2024

## **АННОТАЦИЯ**

Данная работа включает в себя 9 рисунков, 16 листингов. Количество страниц в работе — 33.

# СОДЕРЖАНИЕ

|   |    |
|---|----|
| 1 ПОСТАНОВКА ЗАДАЧИ .....   | 4  |
| 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ .....                                   | 5  |
| 2.1 Исходный код файлов проекта .....                                 | 5  |
| 2.2 Файл с описанной программой на языке Tcl и результаты работы..... | 21 |
| ЗАКЛЮЧЕНИЕ .....  | 30 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....                                | 31 |

# 1 ПОСТАНОВКА ЗАДАЧИ

Сформировать набор файлов для создания проекта: Файлы на языке Verilog, содержащие модули для описания устройства, заданного вариантом, а также модули для верификации RTL-модели, файлы проектных ограничений для размещения проекта на ПЛИС, в набор файлов должны входить: конфигурация для работы устройства на частоте 100 МГц и для работы устройства на частоте 200 МГц. Сформировать файл с описанной программой на языке TCL, в которой: создать новый проект для чипа «xc7a100tcsg324-1». создать в рамках проекта наборы файлов «designs» для каждого отдельно оформленного теста, создать два набора файлов «constrs» проектных ограничений, добавить в проект в набор файлов «source set» с именем «sources\_1» файлы, содержащие модули для описания устройства, заданного вариантом, добавить в проект в каждый набор файлов «simulation set» необходимые файлы для тестирования в рамках набора, добавить в проект в каждый набор файлов «constraints set» требуемые файлы проектных ограничений, запуск процесса симуляции последовательно для каждого из набора «simulation set», запустить синтез и имплементацию последовательно для каждого набора «constraints set», для каждого варианта имплементации получить отчёт о временных задержках («Timing Summary Report»), сохранить данные отчёта в соответствующие файлы. Запустить на исполнение разработанный скрипт. Составить отчёт.

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

### 2.1 Исходный код файлов проекта

Индивидуальный вариант:  $a - b \ll c * d$ . Исходный код проекта был взят из третьей практической работы, где и проведено его описание. Результат представлен на Листингах 2.1 – 2.4.

*Листинг 2.1 – Модуль basis*

```
`timescale 1ns / 1ps

module basis(
    input clk,
    input btn_c,
    input btn_rst,
    input [15:0] SW,
    output reg [31:0] shift_register,
    output reg [7:0] an_mask,
    output reg error
);

reg [3:0] state;
reg [31:0] a_reg, b_reg, d_reg;
reg [15:0] c_reg;

initial begin
    shift_register = 0;
    state <= 0;
    an_mask = 8'b00000000;
end

always @(posedge clk) begin
    if (btn_rst)
        state <= 0;
    else
        case (state)
            4'b0: begin
                a_reg <= 0;
                b_reg <= 0;
                c_reg <= 0;
                an_mask <= 8'b11110000;
                shift_register <= 0;
                error <= 0;
                state <= 1;
            end
            4'd1: begin
                shift_register <= {SW};
                if (btn_c) begin
                    a_reg <= {SW};
                    state <= 2;
                end
            end
            4'd2: begin
                shift_register <= {SW};
                if (btn_c) begin
```

### Продолжение Листинга 2.1

```
        b_reg <= {SW};
        state <= 3;

        end

    end

    4'd3: begin
        shift_register <= {SW};
        if (btn_c) begin
            c_reg <= {SW};
            state <= 4;
        end
    end

    4'd4: begin
        shift_register <= {SW};
        if (btn_c) begin
            d_reg <= {SW};
            an_mask <= 8'b00000000;
            state <= 5;
        end
    end

    4'd5: begin
        shift_register <= a_reg - b_reg << c_reg * d_reg
        state <= 6;
    end

    end

endcase
end

endmodule
```

### Листинг 2.2 – Модуль SevenSegLED

```
`timescale 1ns / 1ps
`timescale 1ns / 1ps

module SevenSegLED (
    input clk,
    input RST,
    input [31:0] NUMBER,
    input [7:0] AN_MASK,
    output [7:0] AN,
    output reg [6:0] SEG
);
    reg [7:0] AN_REG = 0;
    assign AN = AN_REG | AN_MASK;
    reg [2:0] digit_counter;
    wire [3:0] NUMBER_SPLITTER [0:7];

    genvar i;
    generate
        for (i = 0; i < 8; i = i + 1)
            begin
                assign NUMBER_SPLITTER[i] = NUMBER[((i+1)*4-1)-:4];
            end
    endgenerate

    initial begin
        digit_counter = 0;
    end

    always @(posedge clk or posedge RST)
```

### *Продолжение Листинга 2.2*

```
        digit_counter <= RST ? 0 : digit_counter + 4'b1;

always @(digit_counter)
begin
    case (NUMBER_SPLITTER[digit_counter])
        4'h0: SEG <= 7'b1000000;
        4'h1: SEG <= 7'b1111001;
        4'h2: SEG <= 7'b0100100;
        4'h3: SEG <= 7'b0110000;
        4'h4: SEG <= 7'b0011001;
        4'h5: SEG <= 7'b0010010;
        4'h6: SEG <= 7'b0000010;
        4'h7: SEG <= 7'b1111000;
        4'h8: SEG <= 7'b0000000;
        4'h9: SEG <= 7'b0010000;
        4'ha: SEG <= 7'b0001000;
        4'hb: SEG <= 7'b0000011;
        4'hc: SEG <= 7'b1000110;
        4'hd: SEG <= 7'b0100001;
        4'he: SEG <= 7'b0000110;
        4'hf: SEG <= 7'b0001110;
        default: SEG <= 7'b1111111;
    endcase

    case (digit_counter[2:0])
        3'd0: AN_REG <= 8'b11111110;
        3'd1: AN_REG <= 8'b11111101;
        3'd2: AN_REG <= 8'b11111011;
        3'd3: AN_REG <= 8'b11110111;
        3'd4: AN_REG <= 8'b11101111;
        3'd5: AN_REG <= 8'b11011111;
        3'd6: AN_REG <= 8'b10111111;
        3'd7: AN_REG <= 8'b01111111;
        default: AN_REG <= 8'b11111111;
    endcase
end
end
endmodule
```

### *Листинг 2.3 – Модуль фильтра дребезга контактов - filter*

```
`timescale 1ns / 1ps

module filter #(size = 3) (
    input CLK, CLOCK_ENABLE, IN_SIGNAL,
    output reg OUT_SIGNAL, OUT_SIGNAL_ENABLE
);
reg [1:0] IN_SIGNAL_SYNC;
reg [size-1:0] counter;
initial begin
    IN_SIGNAL_SYNC = 0; counter = 0;
    OUT_SIGNAL = 0; OUT_SIGNAL_ENABLE = 0;
end
always @(posedge CLK) begin
    IN_SIGNAL_SYNC <= {IN_SIGNAL_SYNC[0], IN_SIGNAL};
    counter <= (IN_SIGNAL_SYNC[1] ^ OUT_SIGNAL) ?
    {size{1'd0}} : (CLOCK_ENABLE ? counter + 1 : counter);
    if (&(counter) & CLOCK_ENABLE)
        OUT_SIGNAL <= IN_SIGNAL_SYNC[1];
    OUT_SIGNAL_ENABLE <= &(counter) & CLOCK_ENABLE & IN_SIGNAL_SYNC[1];
end
endmodule
```

*Листинг 2.4 – Модуль делителя частоты – clk\_div*

```
`timescale 1ns / 1ps
module clk_div #(DIV = 2) (
    input clk,
    output reg clk_d);
wire [$clog2(DIV/2)-1:0] cnt;
counter #(.STEP(1), .MODULE(DIV/2)) cntr(
    .clk(clk),
    .reset(1'd0),
    .dir(1'd1),
    .enable(1'd1),
    .cnt(cnt));
initial clk_d = 0;
always@(posedge clk)
    if (cnt == 0)
        clk_d = ~clk_d;
endmodule
```

*Листинг 2.5 – Модуль счетчика – counter*

```
`timescale 1ns / 1ps
module counter #(MODULE = 4, STEP = 1) (
    input clk,
    input reset,
    input enable,
    input dir,
    output reg [$clog2(MODULE)-1:0] cnt);
initial cnt = 0;
always@(posedge clk)
begin
    if (enable)
    begin
        if (reset)
            cnt = 0;
        else if (dir)
            cnt = (cnt + STEP) % MODULE;
        else
            cnt = (MODULE + (cnt - STEP)) % MODULE;
    end
end
endmodule
```

Далее будут представлены тестовые модули, разработанные в третьей практической работе, где они также были описаны и верифицированы, на Листингах 2.6 - 2.9.

*Листинг 2.6 – Тестовый модуль делителя частоты – test\_clk\_div*

```
`timescale 1ns / 1ps

module test_clk_div();

    reg clk;
    initial clk = 0;
    always #5 clk <= ~clk;
```



### *Продолжение Листинга 2.6*

```
wire clk_div_out;
clk_div clk_div1 (
    .clk(clk),
    .clk_div(clk_div_out)
);
integer clk_div_period = 200_000;
realtime t_begin, t_end;
initial
begin
    @(posedge clk_div_out);
    t_begin = $realtime;
    @(posedge clk_div_out);
    t_end = $realtime;

    $display("Ожидаемый период сигнала: %0d", clk_div_period);
    $write("Фактический период сигнала: ");
    $write((t_end - t_begin));

    if ( (t_end - t_begin) == clk_div_period )
        $display("\nТест пройден.");
    else
        $display("\nТест НЕ пройден.");
end

endmodule
```

### *Листинг 2.7 – Тестовый модуль фильтра дребезга – test\_filter*

```
`timescale 1ns / 1ps

module test_filter();

reg clk;
initial clk = 0;
always #5 clk <= ~clk;

localparam PRESS = 1, RELEASE = 0;
reg IN_SIGNAL; initial IN_SIGNAL = 0;
wire OUT_SIGNAL_ENABLE;

filter #(5) btn_c_filter(
    .CLK(clk),
    .CLOCK_ENABLE(1),
    .IN_SIGNAL(IN_SIGNAL),
    .OUT_SIGNAL_ENABLE(OUT_SIGNAL_ENABLE)
);

localparam TEST_COUNT = 3;
reg [0:TEST_COUNT-1] test_register;
initial
begin
    test_register = {TEST_COUNT{1'b0}};

    test_filter_1();
    test_filter_2();
    test_filter_3();
    test_show_stats();
end

task test_filter_1;
```

### Продолжение Листинга 2.7

```
    reg test_result;
begin
    $display("\n[%0t]: Тест 1. Реакция фильтра дребезга на сигнал высокого
уровня на шине физ. манипулятора.", $time);
    $display("[%0t]: (время удержания сигнала соответствует требуемому)",
$time);
    send_signal_to_filter(PRESS, 32);
    @(posedge clk) test_result <= (OUT_SIGNAL_ENABLE == 1'b1);
    send_signal_to_filter(RELEASE, 32);
    test_info(1, test_result);
end
endtask

task test_filter_2;
reg test_result;
begin
    $display("\n[%0t]: Тест 2. Реакция фильтра дребезга на сигнал высокого
уровня на шине физ. манипулятора.", $time);
    $display("[%0t]: (время удержания сигнала меньше требуемого)", $time);
    send_signal_to_filter(PRESS, 16);
    @(posedge clk); test_result = (OUT_SIGNAL_ENABLE == 1'b0);
    test_info(2, test_result);
end
endtask

task test_filter_3;
reg test_result;
begin
    $display("\n[%0t]: Тест 3. Реакция фильтра дребезга на сигнал низкого уровня
на шине физ. манипулятора.", $time);
    send_signal_to_filter(RELEASE, 32);
    @(posedge clk); test_result = (OUT_SIGNAL_ENABLE == 1'b0);
    test_info(3, test_result);
end
endtask

task test_info;
input integer test_number;
input test_result;
begin
    test_register[test_number-1] = test_result;
    if (test_result)
        $display("[%0t]: Тест %0d пройден.", $time, test_number);
    else
        $display("[%0t]: Тест %0d НЕ пройден.", $time, test_number);
end
endtask

task test_show_stats;
integer i, test_counter;
begin
    $display("\nРезультаты тестирования:");
    test_counter = 0;

    for (i = 0; i < TEST_COUNT; i = i + 1)
    begin
        if (test_register[i])
            $display("Тест %2d пройден.", i+1);
        else
            $display("Тест %2d НЕ пройден.", i+1);
        test_counter = test_counter + (test_register[i] ? 1 : 0);
    end
end
```

### Продолжение Листинга 2.7

```
        $display("Пройдено тестов: %0d/%0d", test_counter, TEST_COUNT);
    end
endtask

task send_signal_to_filter;
input signal_in;
input [6:0] ticks;
begin
    @(posedge clk);
    IN_SIGNAL <= signal_in;
    $display("[%0t]: Сигнал %b подан на линию.", $time, signal_in);

    repeat(ticks + 2)
        @(posedge clk);

    IN_SIGNAL <= 0;
    $display("[%0t]: Сигнал %b убран с линии, подан сигнал 0", $time,
signal_in);
end
endtask

task test_filter;
input signal_in;
input [6:0] ticks;
output reg test_result;
begin

end
endtask

task push_c_button;
begin
    //$display("Дребезг на кнопке C");

    $srandom(33985);
    repeat($urandom_range(150,0))
        begin
            IN_SIGNAL = $random;
            #3;
        end
    IN_SIGNAL = 1;

    #200;

    repeat($urandom_range(150,0))
        begin
            IN_SIGNAL = $random;
            #3;
        end
    IN_SIGNAL = 0;

    #200;
end
endtask

endmodule
```

*Листинг 2.8 – Тестовый модуль индикатора – test\_seven\_seg\_led*

```
`timescale 1ns / 1ps

module test_sev_seg_led();

reg clk; initial clk = 0;
always #10 clk <= ~clk;

localparam AN_COUNT = 8;
localparam CATH_COUNT = 7;

localparam DIGIT_SIZE = 4;
localparam DIGIT_COUNT = 16;

reg CE, RESET;
reg [AN_COUNT*DIGIT_SIZE-1:0] NUMBER;
reg [AN_COUNT-1:0] AN_MASK;
initial
begin
    CE = 1;
    RESET = 0;
    NUMBER = {(AN_COUNT*DIGIT_SIZE){1'b0}};
end

wire [AN_COUNT-1:0] AN;
wire [CATH_COUNT-1:0] CATH;

SevenSegLED uut (
    .clk(clk),
    .RST(RESET),
    .NUMBER(NUMBER),
    .AN_MASK(AN_MASK),
    .AN(AN),
    .SEG(CATH)
);

initial
begin
    test_seven_segments(8'b00101100);
    test_show_stats();
end

function [6:0] get_cath_mask;
    input [3:0] number;
begin
    case (number)
        4'h0: get_cath_mask = 7'b1000000;
        4'h1: get_cath_mask = 7'b1111001;
        4'h2: get_cath_mask = 7'b0100100;
        4'h3: get_cath_mask = 7'b0110000;
        4'h4: get_cath_mask = 7'b0011001;
        4'h5: get_cath_mask = 7'b0010010;
        4'h6: get_cath_mask = 7'b0000010;
        4'h7: get_cath_mask = 7'b1111000;
        4'h8: get_cath_mask = 7'b0000000;
        4'h9: get_cath_mask = 7'b0010000;
        4'ha: get_cath_mask = 7'b0001000;
        4'hb: get_cath_mask = 7'b0000011;
        4'hc: get_cath_mask = 7'b1000110;
        4'hd: get_cath_mask = 7'b0100001;
        4'he: get_cath_mask = 7'b0000110;
        4'hf: get_cath_mask = 7'b0001110;
        default: get_cath_mask = 7'b1111111;
    endcase
end
```

*Продолжение Листинга 2.8*

```
        endcase
    end
endfunction
function [7:0] get_an_mask;
    input [2:0] an_number;
begin
    case (an_number)
        3'd0: get_an_mask = 8'b11111110;
        3'd1: get_an_mask = 8'b11111101;
        3'd2: get_an_mask = 8'b11111011;
        3'd3: get_an_mask = 8'b11110111;
        3'd4: get_an_mask = 8'b11101111;
        3'd5: get_an_mask = 8'b11011111;
        3'd6: get_an_mask = 8'b10111111;
        3'd7: get_an_mask = 8'b01111111;
        default: get_an_mask = 8'b11111111;
    endcase
end
endfunction

reg [AN_COUNT-1:0] test_an_register;
reg [DIGIT_COUNT-1:0] test_digit_register;
reg test_an_mask_register;

task test_seven_segments;
    input [AN_COUNT-1:0] mask_value;
reg [3:0] i;
reg [3:0] number;

begin
    $display("\n[%0t]: Тест отображения цифр на индикаторах, принципа работы
динамической индикации и анодной маски.", $time);
    test_an_register = {AN_COUNT{1'b1}};
    test_digit_register = {DIGIT_COUNT{1'b1}};
    test_an_mask_register = 1'b1;

    AN_MASK = mask_value;
    $display("Битовая маска (AN_MASK): %b", AN_MASK);

    wait(uut.digit_counter == AN_COUNT-1);
    @(posedge clk);

    number = 0;
    repeat(DIGIT_COUNT)
    begin
        // Подача числа на входную шину
        for (i = 0; i < AN_COUNT; i = i + 1)
            NUMBER[ ((i+1)*4)-1 -: 4 ] <= number;

        @(posedge clk);
        $display("\n[%0t]: Тест для цифры: %h", $time, number);
        for (i = 0; i < AN_COUNT; i = i + 1)
        begin
            $display("Текущий анод: %d", i);

            test_digit_register[number] <= CATH == get_cath_mask(number);
            $display("Ожидаемые сигналы на линии катодов (CATH): %b",
get_cath_mask(number));
            $display("Фактические сигналы на линии катодов (CATH): %b", CATH);

            test_an_register[number] <= uut.AN_REG == get_an_mask(i);
```

### Продолжение Листинга 2.8

```
        $display("Ожидаемые сигналы на линии анодов (ДО применения анодной  
маски): %b", get_an_mask(i));  
        $display("Фактические сигналы на линии анодов (ДО применения анодной  
маски): %b", uut.AN_REG);  
function [7:0] get_an_mask;  
    test_an_mask_register <= AN == (get_an_mask(i) | AN_MASK);  
    $display("Ожидаемые сигналы на линии анодов (ПОСЛЕ применения  
анодной маски): %b", get_an_mask(i) | AN_MASK);  
    $display("Фактические сигналы на линии анодов (ПОСЛЕ применения  
анодной маски): %b", AN);  
  
    if (i != AN_COUNT-1)  
        @(posedge clk);  
    end  
    number = number + 1;  
end  
endtask  
task test_show_stats;  
localparam TEST_COUNT = 3;  
integer test_counter, i;  
begin  
    test_counter = 0;  
    $display("\n[%0t]: Результаты тестирования:", $time);  
    // Отображение цифры  
    if (&(test_digit_register))  
    begin  
        $display("1. Тест на отображение пройден успешно для всех возможных  
вариантов цифр.");  
        test_counter = test_counter + 1;  
    end  
    else begin  
        $display("1. Тест на отображение цифр НЕ пройден");  
        for (i = 0; i < DIGIT_COUNT; i = i + 1)  
            if (!test_digit_register[i])  
                $display("Ошибка отображения цифры %d", i);  
    end  
    // Динамическая индикация  
    if (&(test_an_register))  
    begin  
        test_counter = test_counter + 1;  
        $display("2. Тест работы динамической индикации пройден успешно.");  
    end  
    else begin  
        $display("2. Тест работы динамической индикации НЕ пройден.", i);  
        for (i = 0; i < AN_COUNT; i = i + 1)  
            if (!test_an_register[i])  
                $display("Ошибка на индикаторе %0d.", i);  
    end  
    // Анодная маска  
    if (test_an_mask_register)  
    begin  
        $display("3. Тест анодной маски пройден успешно.");  
        test_counter = test_counter + 1;  
    end  
    else  
        $display("3. Тест анодной маски НЕ пройден.");  
  
    $display("Пройдено тестов: %0d/%0d.", test_counter, TEST_COUNT);  
end  
endtask  
endmodule
```

*Листинг 2.9 – Тестовый модуль конечного автомата – test\_basis*

```
`timescale 1ns / 1ps

module test_basis();
reg clk;
initial clk = 0;
always #5 clk <= ~clk;

reg btn_c, btn_rst;
reg [31:0] SW;
wire [31:0] shift_register;
wire [7:0] an_mask;
wire error;

initial begin
    btn_c = 0;
    btn_rst = 0;
end

basis uut(
    .clk(clk),
    .btn_c(btn_c),
    .btn_rst(btn_rst),
    .SW(SW),
    .shift_register(shift_register),
    .an_mask(an_mask),
    .error(error)
);

localparam TEST_COUNT = 3;
reg [0:TEST_COUNT-1] test_register;
initial
begin
    test_register = {TEST_COUNT{1'b0}}; test_op_1();
    send_rst();
    test_op_2(); send_rst();
    test_op_3();
    test_show_stats();
end

task test_op_1;
reg test_result;
begin
    $display("\n[%0t]: Тест 1. a = 1001, b = 0001, c = 0003, d = 0001.", $time);
    send_signal_to_op(16'h1001);

    send_signal_to_op(16'h0001);

    send_signal_to_op(16'h0003);

    send_signal_to_op(16'h0001);
    repeat(5)
        @(posedge clk);
    test_result = ((shift_register == 32'h000037cd) && (error == 1'b0));
    test_info(1, test_result); end
endtask

task test_op_2;
reg test_result;
begin
    $display("\n[%0t]: Тест 2. a = 5400, b = 4400, c = 0002, d = 0001.", $time);
```

*Продолжение Листинга 2.9*

```
send_signal_to_op(16'h5400);

send_signal_to_op(16'h4400);

send_signal_to_op(16'h0002);

send_signal_to_op(16'h0001);
repeat(5)
    @(posedge clk);
test_result = (shift_register == 16'hffff && error == 1'b1);
test_info(2, test_result);
end
endtask

task test_op_3;
reg test_result;
begin
    $display("\n[%0t]: Тест 3. a = 3219, b = 1119, c = 0001, d = 0002.", $time);
    send_signal_to_op(16'h3219);

    send_signal_to_op(16'h1119);

    send_signal_to_op(16'h0001);

    send_signal_to_op(16'h0002);

    repeat(5)
        @(posedge clk);
    test_result = (shift_register == 32'h000000c8 && error == 1'b0);
    test_info(3, test_result);
end
endtask

task test_info;
input integer test_number;
input test_result;
begin
    test_register[test_number-1] = test_result;
    if (test_result)
        $display("[%0t]: Тест %0d пройден.", $time, test_number);
    else
        $display("[%0t]: Тест %0d НЕ пройден.", $time, test_number);
end
endtask

task test_show_stats; integer i, test_counter;
begin
    $display("\nРезультаты тестирования:");
    test_counter = 0;

    for (i = 0; i < TEST_COUNT; i = i + 1)
    begin
        if (test_register[i])
            $display("Тест %2d пройден.", i+1);
        else
            $display("Тест %2d НЕ пройден.", i+1);
        test_counter = test_counter + (test_register[i] ? 1 : 0);
    end
    $display("Пройдено тестов: %0d/%0d", test_counter, TEST_COUNT);
end
endtask
```



### *Продолжение Листинга 2.9*

```
    task send_rst;
begin
    @(posedge clk);
    @(posedge clk) btn_rst <= 1;
    @(posedge clk) btn_rst <= 0;
end
endtask

task send_signal_to_op;
input [15:0] in_signal;
begin
    @(posedge clk);
    SW <= in_signal;
    $display("[%0t]: Сигнал %h подан на линию.", $time, in_signal); @(posedge
clk) btn_c <= 1;
    @(posedge clk) btn_c <= 0;
end
endtask

endmodule
```

Далее представлен модуль верхнего уровня на Листинге 2.10

### *Листинг 2.10 – Модуль верхнего уровня – upper*

```
`timescale 1ns / 1ps

module upper(
    input clk,
    input btn_c,
    input btn_rst,
    input [15:0] SW,
    output [7:0] AN,
    output [6:0] SEG
);

wire [31:0] shift_register;
wire [7:0] an_mask;
wire error;

reg CLOCK_ENABLE = 0;
always @(posedge clk)
    CLOCK_ENABLE <= ~CLOCK_ENABLE;

wire btn_c_out, btn_c_out_enable;
filter #(.size(7)) btn_c_filter (
    .CLK(clk),
    .CLOCK_ENABLE(CLOCK_ENABLE),
    .IN_SIGNAL(btn_c),
    .OUT_SIGNAL(btn_c_out),
    .OUT_SIGNAL_ENABLE(btn_c_out_enable)
);

wire btn_rst_out, btn_rst_out_enable;
filter #(.size(7)) btn_RST_filter (
    .CLK(clk),
    .CLOCK_ENABLE(CLOCK_ENABLE),
    .IN_SIGNAL(btn_rst),
    .OUT_SIGNAL(btn_rst_out),
    .OUT_SIGNAL_ENABLE(btn_rst_out_enable)
```

### *Продолжение Листинга 2.10*

```
);  
basis basis1(  
    .clk(clk),  
    .btn_c(btn_c_out_enable),  
    .btn_rst(btn_rst_out_enable),  
    .SW(SW),  
    .shift_register(shift_register),  
    .an_mask(an_mask),  
    .error(error)  
);  
clk_div #(.DIV(20)) clk_div1 (  
    .clk(clk),  
    .clk_d(clk_div_out)  
);  
clk_div #(.DIV(10)) clk_div2 (  
    .clk(clk),  
    .clk_d(clk_div_out_10)  
);  
SevenSegLED seg (  
    .clk(clk_div_out),  
    .RST(btn_rst_out_enable),  
    .NUMBER(shift_register),  
    .AN_MASK(an_mask),  
    .AN(AN),  
    .SEG(SEG)  
);  
endmodule
```

На Листинге 2.11 представлен модуль для взаимодействия с *vio*, также разработанный в предыдущей работе

### *Листинг 2.11 – Модуль для взаимодействия с *vio* – main*

```
`timescale 1ns / 1ps  
  
module main(  
    input clk,  
    output [7:0] AN,  
    output [6:0] SEG  
);  
  
wire btn_c, btn_rst;  
wire [15:0] SW;  
upper uut(  
    .clk(clk),  
    .btn_c(btn_c),  
    .btn_rst(btn_rst),  
    .SW(SW),  
    .AN(AN),  
    .SEG(SEG)  
);  
vio_0 vio(  
    .clk(clk),  
    .probe_in0(AN),  
    .probe_in1(SEG),  
    .probe_out0(btn_c),  
    .probe_out1(btn_res),  
    .probe_out2(SW)  
);  
endmodule
```

Далее представлены файлы проектных ограничений, два из которых предназначены для работы без `vio` и два для работы с `vio`. Также учтено, что в работу должны входить две конфигурации для работы устройства на частоте 100 и 200 МГц. Файлы проектных ограничений представлены на Листингах 2.12 – 2.15.

*Листинг 2.12 – Конфигурация файла проектных ограничений с частотой 100 и работы с `vio`*

```
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN E3} [get_ports {clk}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J17} [get_ports {AN[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J18} [get_ports {AN[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T9} [get_ports {AN[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J14} [get_ports {AN[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P14} [get_ports {AN[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T14} [get_ports {AN[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K2} [get_ports {AN[6]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN U13} [get_ports {AN[7]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T10} [get_ports {SEG[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN R10} [get_ports {SEG[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K16} [get_ports {SEG[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K13} [get_ports {SEG[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P15} [get_ports {SEG[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T11} [get_ports {SEG[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN L18} [get_ports {SEG[6]}]
```

*Листинг 2.13 – Конфигурация файла проектных ограничений с частотой 200 и работы с `vio`*

```
create_clock -add -name sys_clk_pin -period 5.00 -waveform {0 4} [get_ports {clk}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN E3} [get_ports {clk}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J17} [get_ports {AN[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J18} [get_ports {AN[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T9} [get_ports {AN[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J14} [get_ports {AN[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P14} [get_ports {AN[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T14} [get_ports {AN[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K2} [get_ports {AN[6]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN U13} [get_ports {AN[7]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T10} [get_ports {SEG[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN R10} [get_ports {SEG[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K16} [get_ports {SEG[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K13} [get_ports {SEG[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P15} [get_ports {SEG[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T11} [get_ports {SEG[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN L18} [get_ports {SEG[6]}]
```

*Листинг 2.14–Конфигурация файла проектных ограничений с частотой 100 и работы без vio*

```
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN E3} [get_ports {clk}]

set_property -dict {PACKAGE_PIN J15 IOSTANDARD LVCMOS33} [get_ports {SW[0]}]
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33} [get_ports {SW[1]}]
set_property -dict {PACKAGE_PIN M13 IOSTANDARD LVCMOS33} [get_ports {SW[2]}]
set_property -dict {PACKAGE_PIN R15 IOSTANDARD LVCMOS33} [get_ports {SW[3]}]
set_property -dict {PACKAGE_PIN R17 IOSTANDARD LVCMOS33} [get_ports {SW[4]}]
set_property -dict {PACKAGE_PIN T18 IOSTANDARD LVCMOS33} [get_ports {SW[5]}]
set_property -dict {PACKAGE_PIN U18 IOSTANDARD LVCMOS33} [get_ports {SW[6]}]
set_property -dict {PACKAGE_PIN R13 IOSTANDARD LVCMOS33} [get_ports {SW[7]}]
set_property -dict {PACKAGE_PIN T8 IOSTANDARD LVCMOS33} [get_ports {SW[8]}]
set_property -dict {PACKAGE_PIN U8 IOSTANDARD LVCMOS33} [get_ports {SW[9]}]
set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVCMOS33} [get_ports {SW[10]}]
set_property -dict {PACKAGE_PIN T13 IOSTANDARD LVCMOS33} [get_ports {SW[11]}]
set_property -dict {PACKAGE_PIN H6 IOSTANDARD LVCMOS33} [get_ports {SW[12]}]
set_property -dict {PACKAGE_PIN U12 IOSTANDARD LVCMOS33} [get_ports {SW[13]}]
set_property -dict {PACKAGE_PIN U11 IOSTANDARD LVCMOS33} [get_ports {SW[14]}]
set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS33} [get_ports {SW[15]}]

set_property -dict {PACKAGE_PIN N17 IOSTANDARD LVCMOS33} [get_ports {btn_c}]
set_property -dict {PACKAGE_PIN M17 IOSTANDARD LVCMOS33} [get_ports {btn_rst}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J17} [get_ports {AN[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J18} [get_ports {AN[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T9} [get_ports {AN[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J14} [get_ports {AN[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P14} [get_ports {AN[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T14} [get_ports {AN[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K2} [get_ports {AN[6]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN U13} [get_ports {AN[7]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T10} [get_ports {SEG[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN R10} [get_ports {SEG[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K16} [get_ports {SEG[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K13} [get_ports {SEG[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P15} [get_ports {SEG[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T11} [get_ports {SEG[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN L18} [get_ports {SEG[6]}]
```

*Листинг 2.15–Конфигурация файла проектных ограничений с частотой 200 и работы без vio*

```
create_clock -add -name sys_clk_pin -period 5.00 -waveform {0 4} [get_ports {clk}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN E3} [get_ports {clk}]

set_property -dict {PACKAGE_PIN J15 IOSTANDARD LVCMOS33} [get_ports {SW[0]}]
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33} [get_ports {SW[1]}]
set_property -dict {PACKAGE_PIN M13 IOSTANDARD LVCMOS33} [get_ports {SW[2]}]
set_property -dict {PACKAGE_PIN R15 IOSTANDARD LVCMOS33} [get_ports {SW[3]}]
set_property -dict {PACKAGE_PIN R17 IOSTANDARD LVCMOS33} [get_ports {SW[4]}]
set_property -dict {PACKAGE_PIN T18 IOSTANDARD LVCMOS33} [get_ports {SW[5]}]
set_property -dict {PACKAGE_PIN U18 IOSTANDARD LVCMOS33} [get_ports {SW[6]}]
set_property -dict {PACKAGE_PIN R13 IOSTANDARD LVCMOS33} [get_ports {SW[7]}]
set_property -dict {PACKAGE_PIN T8 IOSTANDARD LVCMOS33} [get_ports {SW[8]}]
set_property -dict {PACKAGE_PIN U8 IOSTANDARD LVCMOS33} [get_ports {SW[9]}]
set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVCMOS33} [get_ports {SW[10]}]
set_property -dict {PACKAGE_PIN T13 IOSTANDARD LVCMOS33} [get_ports {SW[11]}]
set_property -dict {PACKAGE_PIN H6 IOSTANDARD LVCMOS33} [get_ports {SW[12]}]
```

### *Продолжение Листинга 2.15*

```
set_property -dict {PACKAGE_PIN U12 IOSTANDARD LVCMOS33} [get_ports {SW[13]}]
set_property -dict {PACKAGE_PIN U11 IOSTANDARD LVCMOS33} [get_ports {SW[14]}]
set_property -dict {PACKAGE_PIN V10 IOSTANDARD LVCMOS33} [get_ports {SW[15]}]

set_property -dict {PACKAGE_PIN N17 IOSTANDARD LVCMOS33} [get_ports {btn_c}]
set_property -dict {PACKAGE_PIN M17 IOSTANDARD LVCMOS33} [get_ports {btn_rst}]

set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J17} [get_ports {AN[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J18} [get_ports {AN[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T9} [get_ports {AN[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN J14} [get_ports {AN[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P14} [get_ports {AN[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T14} [get_ports {AN[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K2} [get_ports {AN[6]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN U13} [get_ports {AN[7]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T10} [get_ports {SEG[0]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN R10} [get_ports {SEG[1]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K16} [get_ports {SEG[2]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN K13} [get_ports {SEG[3]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN P15} [get_ports {SEG[4]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN T11} [get_ports {SEG[5]}]
set_property -dict {IOSTANDARD LVCMOS33 PACKAGE_PIN L18} [get_ports {SEG[6]}]
```

## **2.2 Файл с описанной программой на языке Tcl и результаты работы**

Если проект с именем Prac\_4 открыт, то он закрывается. Далее создается новый проект с этим же именем, в который копируются все исходные модули, включая тестовые файлы и файлы проектных ограничений. Создание и настройка сета симуляции для каждого тестового модуля, которая включает создание сетов симуляций, добавление модулей дизайна и тестовых модулей в наборы. Удаление сета симуляции по умолчанию. Создание сетов для файлов проектных ограничений. Удаление сета проектных ограничений по умолчанию и добавление файлов проектных ограничений в проект. Создание IP-ядра VIO. Симуляция на разных наборах тестов и имплементация на разных наборах файлов проектных ограничений. Код программы на языке Tcl представлен на Листинге 2.16.

### *Листинг 2.16 – Код программы на языке Tcl*

```
set project_name Prac_4
set project_found [llength [get_projects $project_name] ]
if {$project_found > 0} close_project

set origin_dir [file dirname [info script]]
cd $origin_dir
```

### Продолжение Листинга 2.16

```
set path $origin_dir/$project_name/$project_name

# Создание проекта
create_project $project_name $project_name -force -part xc7a100tcsq324-1

# Добавление файлов дизайна
file mkdir $path.srscs/designs
set design_file_list [list seven_seg_light.v basis.v clk_div.v upper.v filter.v
main.v counter.v ]
foreach s $design_file_list {
    file copy $origin_dir/${s} ${path}.srscs/designs/${s}
    add_files -fileset sources_1 -norecurse ${path}.srscs/designs/${s}
}
# Добавление файлов тестового окружения
file mkdir $path.srscs/testbenches
set test_name_list [ list test_light test_basis test_clk_div test_filter ]
# Формирование ассоциативного массива формата (тестовый модуль - список файлов
дизайнов для тестирования)
set test_set_designs(test_light) [ list seven_seg_light.v ]
set test_set_designs(test_basis) [ list basis.v ]
set test_set_designs(test_clk_div) [ list clk_div.v counter.v ]
set test_set_designs(test_filter) [ list filter.v ]

# Создание и настройка сета симуляции для каждого тестового модуля
foreach s $test_name_list {

    # Создание сета симуляции
    set set_name ${s}_set
    file mkdir ${path}.srscs/testbenches/${set_name}
    create_fileset -simset $set_name
    set_property SOURCE_SET {} [ get_filesets $set_name ]

    # Добавления модулей дизайна в сет
    set designs $test_set_designs(${s})
    foreach design $designs {
        add_files -fileset $set_name -norecurse ${path}.srscs/designs/$design
    }

    # Добавление тестового модуля в сет
    file copy $origin_dir/${s}.v ${path}.srscs/testbenches/${set_name}/${s}.v
    add_files -fileset $set_name ${path}.srscs/testbenches/${set_name}/${s}.v
    set_property TOP ${s} [get_filesets ${set_name}]
}

# Удаление сета симуляции по умолчанию
current_fileset -simset [ get_filesets test_light_set ]
delete_fileset [ get_filesets sim_1 ]
file delete -force $path.srscs/sim_1

# Создание сетов для файлов проектных ограничений
create_fileset -constrset fsm_with_vio_100_set
create_fileset -constrset fsm_with_vio_200_set
create_fileset -constrset fsm_100_set
create_fileset -constrset fsm_200_set

# Ассоциативный массив с указанием модуля верхнего уровня
# для каждого сета проектных ограничений

array set constrset_upper_module {
    fsm_with_vio_100_set main
    fsm_with_vio_200_set main
}
```

### Продолжение Листинга 2.16

```
fsm_100_set upper
fsm_200_set upper
}
# Удаление сета проектных ограничений по умолчанию
set_property constrset fsm_with_vio_100_set [get_runs synth_1]
set_property constrset fsm_with_vio_100_set [get_runs impl_1]
delete_fileset [ get_filesets constrs_1 ]
file delete -force $path.srcs/constrs_1

# Добавление файлов проектных ограничений в проект
file mkdir $path.srcs/constrs/fsm_with_vio
file mkdir $path.srcs/constrs/fsm
file copy $origin_dir/c_fsm_100.xdc $path.srcs/constrs/fsm/c_fsm_100.xdc
file copy $origin_dir/c_fsm_200.xdc $path.srcs/constrs/fsm/c_fsm_200.xdc
file copy $origin_dir/c_vio_100.xdc
$path.srcs/constrs/fsm_with_vio/c_vio_100.xdc
file copy $origin_dir/c_vio_200.xdc
$path.srcs/constrs/fsm_with_vio/c_vio_200.xdc
add_files -fileset fsm_100_set -norecurse $path.srcs/constrs/fsm/c_fsm_100.xdc
add_files -fileset fsm_200_set -norecurse $path.srcs/constrs/fsm/c_fsm_200.xdc
add_files -fileset fsm_with_vio_100_set -norecurse
$path.srcs/constrs/fsm_with_vio/c_vio_100.xdc
add_files -fileset fsm_with_vio_200_set -norecurse
$path.srcs/constrs/fsm_with_vio/c_vio_200.xdc

# Создание IP-ядра VIO
create_ip -name vio -vendor xilinx.com -library ip -version 3.0 -module_name
vio_0
set_property -dict [list \
    CONFIG.C_NUM_PROBE_IN {2} \
    CONFIG.C_NUM_PROBE_OUT {3} \
    CONFIG.C_PROBE_IN0_WIDTH {8} \
    CONFIG.C_PROBE_IN1_WIDTH {7} \
    CONFIG.C_PROBE_OUT2_WIDTH {16} \
] [get_ips vio_0]

generate_target {instantiation_template} [get_files
"$path.srcs/sources_1/ip/vio_0/vio_0.xci"]

update_compile_order -fileset sources_1
generate_target all [get_files "$path.srcs/sources_1/ip/vio_0/vio_0.xci"]

catch { config_ip_cache -export [get_ips -all vio_0] }
export_ip_user_files -of_objects [get_files
"$path.srcs/sources_1/ip/vio_0/vio_0.xci"] -no_script -sync -force -quiet
create_ip_run [get_files -of_objects [get_fileset sources_1]
"$path.srcs/sources_1/ip/vio_0/vio_0.xci"]
launch_runs vio_0_synth_1 -jobs 16
wait_on_runs vio_0_synth_1
export_simulation -of_objects [get_files
"$path.srcs/sources_1/ip/vio_0/vio_0.xci"] -directory
"$path.ip_user_files/sim_scripts" -ip_user_files_dir "$path.ip_user_files" -
ipstatic_source_dir "$path.ip_user_files/ipstatic" -lib_map_path [list
{modelsim="$path.cache/compile_simlib/modelsim"}
{questa="$path.cache/compile_simlib/questa"}
{riviera="$path.cache/compile_simlib/riviera"}
{activehdl="$path.cache/compile_simlib/activehdl"}] -use_ip_compiled_libs -force
-quiet

# Симуляция на разных наборах (simulation sets)
foreach t_set [ get_filesets test* ] {
    current_fileset -simset $t_set
```

*Продолжение Листинга 2.16*

```
    file mkdir $origin_dir/$project_name/sim_output/${t_set}
# Команда сбрасывает время симуляции
    set_property -name xsim.simulate.runtime -value 0 -objects [get_filesets
${t_set}]

    launch_simulation

# Запуск симуляции на 210000ns и перенаправление вывода из TCL-консоли в файл
    restart
    run 210000ns > "$origin_dir/$project_name/sim_output/${t_set}/sim_output.txt"

    close_sim
}
# Имплементация на разных наборах (constraints set)
foreach c_set [ get_filesets fsm* ] {
    set_property constrset $c_set [get_runs synth_1]
    set_property constrset $c_set [get_runs impl_1]
    set_property TOP $constrset_upper_module($c_set) [get_fileset sources_1]
    reset_run synth_1
    launch_runs synth_1 -jobs 16
    wait_on_run synth_1
    reset_run impl_1
    launch_runs impl_1 -jobs 16

    wait_on_run impl_1
    open_run impl_1
    report_timing_summary -file
"$origin_dir/$project_name/timing_summary_${c_set}"
}
```



Результатом работы данного кода стало создание проекта, также программой были проведены тесты, результаты которых были записаны в директории `sim_output` (Рисунок 2.1). Программа провела все тесты и записала итоговый результат в соответствующие файлы (Рисунок 2.2 – 2.5). Были проведены синтез и имплементация на каждом наборе файлов проектных ограничений. Результаты были записаны во временную сводку (Рисунок 2.6 – 2.9).

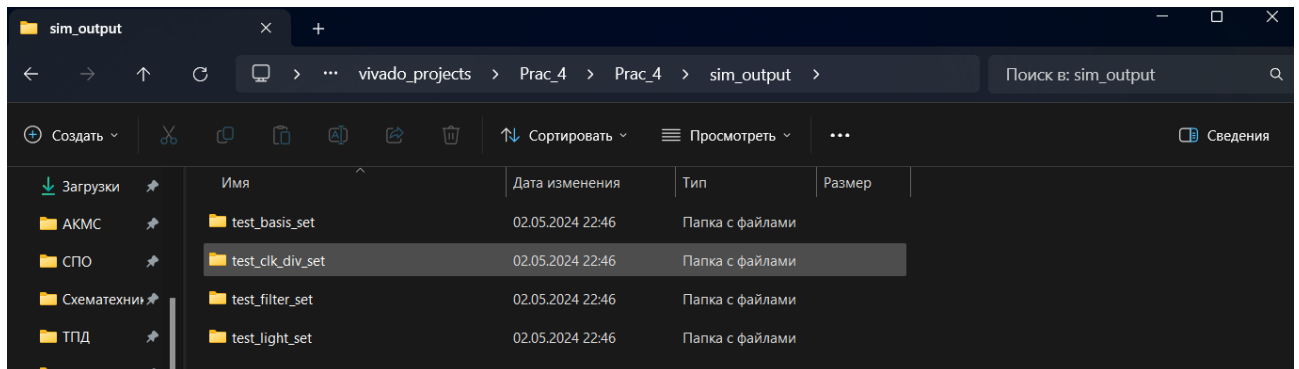


Рисунок 2.1 – Директория `sim_output`

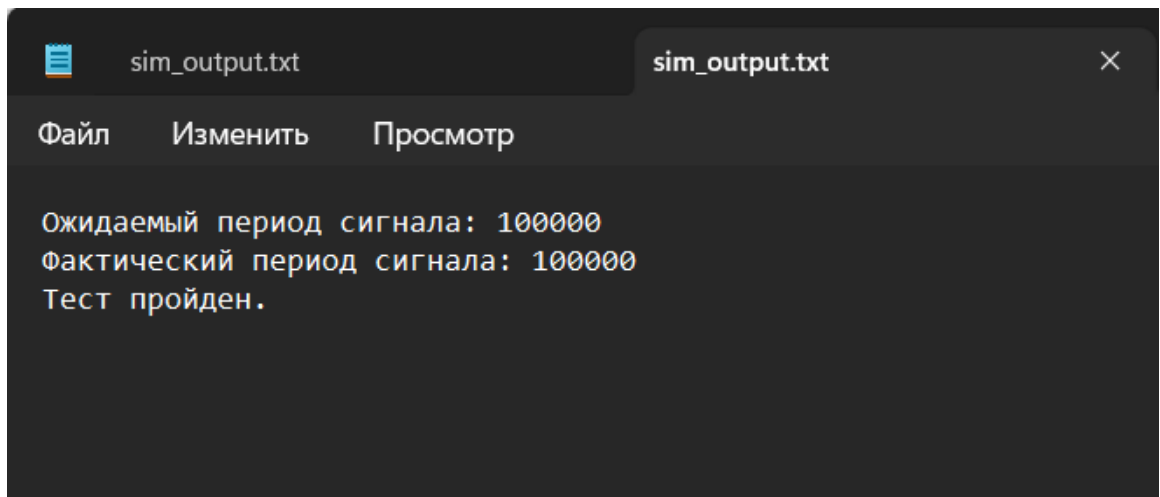
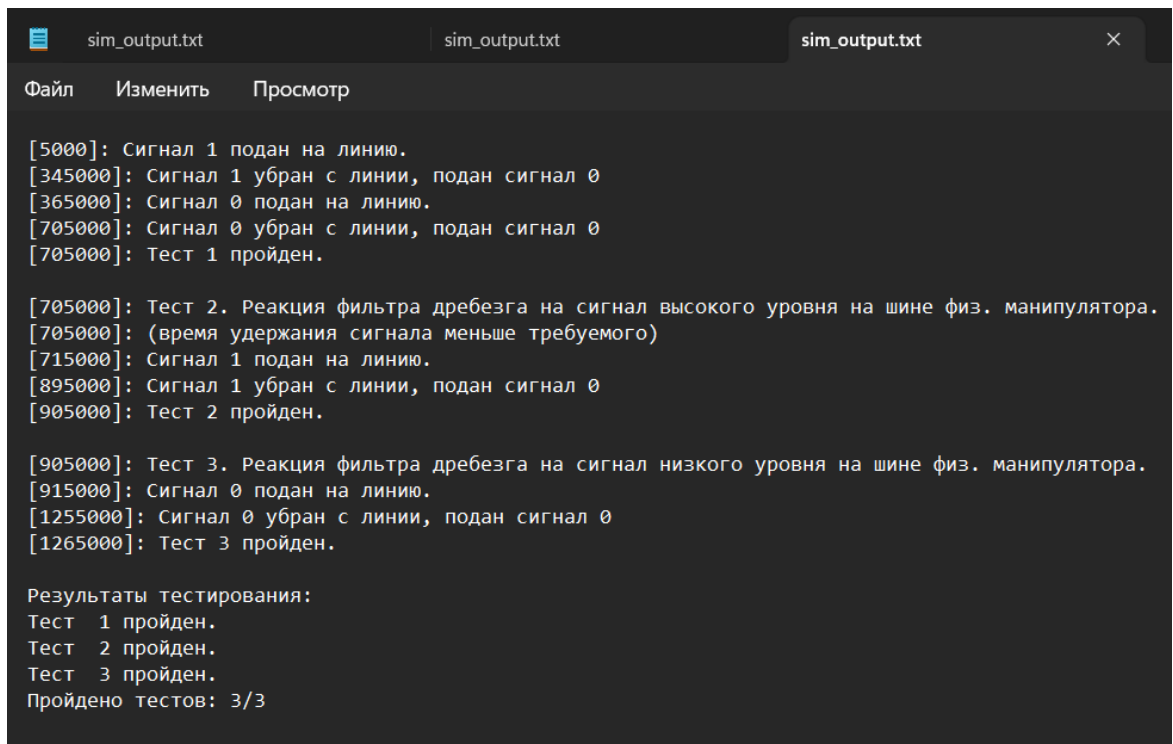


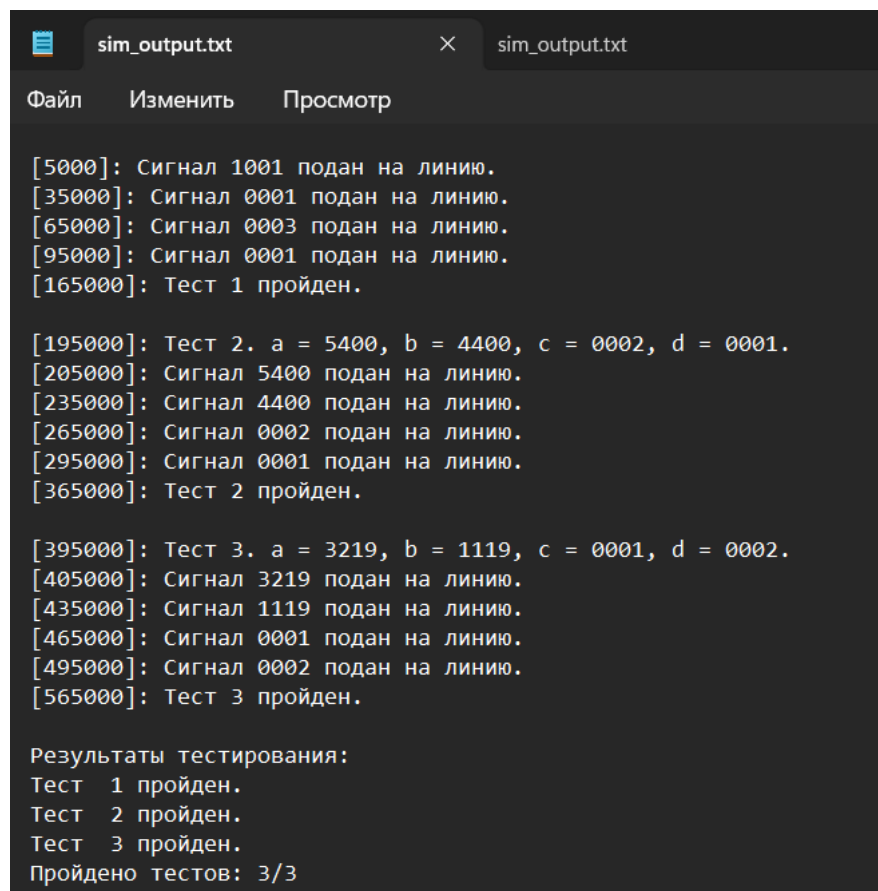
Рисунок 2.2 – Файл результата симуляции делителя частоты



The screenshot shows a text editor window with three tabs, all named 'sim\_output.txt'. The window has a menu bar with 'Файл', 'Изменить', and 'Просмотр'. The text content is as follows:

```
[5000]: Сигнал 1 подан на линию.  
[345000]: Сигнал 1 убран с линии, подан сигнал 0  
[365000]: Сигнал 0 подан на линию.  
[705000]: Сигнал 0 убран с линии, подан сигнал 0  
[705000]: Тест 1 пройден.  
  
[705000]: Тест 2. Реакция фильтра дребезга на сигнал высокого уровня на шине физ. манипулятора.  
[705000]: (время удержания сигнала меньше требуемого)  
[715000]: Сигнал 1 подан на линию.  
[895000]: Сигнал 1 убран с линии, подан сигнал 0  
[905000]: Тест 2 пройден.  
  
[905000]: Тест 3. Реакция фильтра дребезга на сигнал низкого уровня на шине физ. манипулятора.  
[915000]: Сигнал 0 подан на линию.  
[1255000]: Сигнал 0 убран с линии, подан сигнал 0  
[1265000]: Тест 3 пройден.  
  
Результаты тестирования:  
Тест 1 пройден.  
Тест 2 пройден.  
Тест 3 пройден.  
Пройдено тестов: 3/3
```

Рисунок 2.3 – Файл результата симуляции фильтра дребезга



The screenshot shows a text editor window with two tabs, both named 'sim\_output.txt'. The window has a menu bar with 'Файл', 'Изменить', and 'Просмотр'. The text content is as follows:

```
[5000]: Сигнал 1001 подан на линию.  
[35000]: Сигнал 0001 подан на линию.  
[65000]: Сигнал 0003 подан на линию.  
[95000]: Сигнал 0001 подан на линию.  
[165000]: Тест 1 пройден.  
  
[195000]: Тест 2. a = 5400, b = 4400, c = 0002, d = 0001.  
[205000]: Сигнал 5400 подан на линию.  
[235000]: Сигнал 4400 подан на линию.  
[265000]: Сигнал 0002 подан на линию.  
[295000]: Сигнал 0001 подан на линию.  
[365000]: Тест 2 пройден.  
  
[395000]: Тест 3. a = 3219, b = 1119, c = 0001, d = 0002.  
[405000]: Сигнал 3219 подан на линию.  
[435000]: Сигнал 1119 подан на линию.  
[465000]: Сигнал 0001 подан на линию.  
[495000]: Сигнал 0002 подан на линию.  
[565000]: Тест 3 пройден.  
  
Результаты тестирования:  
Тест 1 пройден.  
Тест 2 пройден.  
Тест 3 пройден.  
Пройдено тестов: 3/3
```

Рисунок 2.4 – Файл результата симуляции конечного автомата

```

sim_output.txt  sim_output.txt  sim_output.txt  sim_c
Файл  Изменить  Просмотр
-----
Фактические сигналы на линии анодов (ПОСЛЕ применения анодной маски): 11101111
Текущий анод: 5
Ожидаемые сигналы на линии катодов (CATH): 0001110
Фактические сигналы на линии катодов (CATH): 0001110
Ожидаемые сигналы на линии анодов (ДО применения анодной маски): 11011111
Фактические сигналы на линии анодов (ДО применения анодной маски): 11011111
Ожидаемые сигналы на линии анодов (ПОСЛЕ применения анодной маски): 11111111
Фактические сигналы на линии анодов (ПОСЛЕ применения анодной маски): 11111111
Текущий анод: 6
Ожидаемые сигналы на линии катодов (CATH): 0001110
Фактические сигналы на линии катодов (CATH): 0001110
Ожидаемые сигналы на линии анодов (ДО применения анодной маски): 10111111
Фактические сигналы на линии анодов (ДО применения анодной маски): 10111111
Ожидаемые сигналы на линии анодов (ПОСЛЕ применения анодной маски): 10111111
Фактические сигналы на линии анодов (ПОСЛЕ применения анодной маски): 10111111
Текущий анод: 7
Ожидаемые сигналы на линии катодов (CATH): 0001110
Фактические сигналы на линии катодов (CATH): 0001110
Ожидаемые сигналы на линии анодов (ДО применения анодной маски): 01111111
Фактические сигналы на линии анодов (ДО применения анодной маски): 01111111
Ожидаемые сигналы на линии анодов (ПОСЛЕ применения анодной маски): 01111111
Фактические сигналы на линии анодов (ПОСЛЕ применения анодной маски): 01111111

[2710000]: Результаты тестирования:
1. Тест на отображение пройден успешно для всех возможных вариантов цифр.
2. Тест работы динамической индикации пройден успешно.
3. Тест анодной маски пройден успешно.
Пройдено тестов: 3/3.

```

**Рисунок 2.5 – Файл результата симуляции модуля управления индикаторами**

```

-----
| Design Timing Summary
| -----
-----
WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints  WHS(ns)  THS(ns)  THS Failing Endpoints  THS Total Endpoints
WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
-----
7.927        0.000           0           10        0.221        0.000           0           10
4.500        0.000           0           11
-----

All user specified timing constraints are met.

-----
| Clock Summary
| -----
-----
Clock      Waveform(ns)      Period(ns)      Frequency(MHz)
-----
sys_clk_pin {0.000 5.000}  10.000          100.000

```

**Рисунок 2.6 – Значения задержек по Setup и Hold для набора проектных ограничений без vio с частотой 100 МГц**

|  |               |                        |                      |         |         |                       |                     |
|--|---------------|------------------------|----------------------|---------|---------|-----------------------|---------------------|
| -----  |               |                        |                      |         |         |                       |                     |
| Design Timing Summary                          |               |                        |                      |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| WNS(ns)  | TNS(ns)       | TNS Failing Endpoints  | TNS Total Endpoints  | WHS(ns) | THS(ns) | THS Failing Endpoints | THS Total Endpoints |
| WPWS(ns)                                       | TPWS(ns)      | TPWS Failing Endpoints | TPWS Total Endpoints |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| 2.951  | 0.000         | 0                      | 10                   | 0.221   | 0.000   | 0                     | 10                  |
| 0.500  | 0.000         | 0                      | 11                   |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| All user specified timing constraints are met. |               |                        |                      |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| Clock Summary                                  |               |                        |                      |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| Clock  | Waveform(ns)  | Period(ns)             | Frequency(MHz)       |         |         |                       |                     |
| -----  |               |                        |                      |         |         |                       |                     |
| sys_clk_pin                                    | {0.000 4.000} | 5.000                  | 200.000              |         |         |                       |                     |

**Рисунок 2.7 – Значения задержек по Setup и Hold для набора проектных ограничений без v<sub>io</sub> с частотой 200 МГц**

-----

| Design Timing Summary

| -----

-----

| WNS(ns)  | TNS(ns)                | TNS Failing Endpoints | TNS Total Endpoints | WHS(ns) | THS(ns) | THS Failing Endpoints | THS Total Endpoints | WPWS(ns) |
|----------|------------------------|-----------------------|---------------------|---------|---------|-----------------------|---------------------|----------|
| TPWS(ns) | TPWS Failing Endpoints | TPWS Total Endpoints  |                     |         |         |                       |                     |          |
| -----    |                        |                       |                     |         |         |                       |                     |          |
| 5.180    | 0.000                  | 0                     | 2156                | 0.037   | 0.000   | 0                     | 2140                | 3.750    |
| 0.000    | 0                      | 1183                  |                     |         |         |                       |                     |          |
| -----    |                        |                       |                     |         |         |                       |                     |          |

All user specified timing constraints are met.

-----

| Clock Summary

| -----

-----

| Clock  | Waveform(ns)   | Period(ns) | Frequency(MHz) |
|--|----------------|------------|----------------|
| -----  |                |            |                |
| dbg_hub/inst/BSCANID.u_xsdbm_id/SWITCH_N_EXT_BSCAN.bscan_inst/SERIES7_BSCAN.bscan_inst/TCK | {0.000 16.500} | 33.000     | 30.303         |
| sys_clk_pin  | {0.000 5.000}  | 10.000     | 100.000        |

**Рисунок 2.8 – Значения задержек по Setup и Hold для набора проектных ограничений с v<sub>io</sub> с частотой 100 МГц**

-----

| Design Timing Summary

| -----

-----

| WNS(ns)  | TNS(ns)                | TNS Failing Endpoints | TNS Total Endpoints | WHS(ns) | THS(ns) | THS Failing Endpoints | THS Total Endpoints | WPWS(ns) |
|----------|------------------------|-----------------------|---------------------|---------|---------|-----------------------|---------------------|----------|
| TPWS(ns) | TPWS Failing Endpoints | TPWS Total Endpoints  |                     |         |         |                       |                     |          |
| -----    |                        |                       |                     |         |         |                       |                     |          |
| 0.549    | 0.000                  | 0                     | 2156                | 0.094   | 0.000   | 0                     |                     |          |
| 2140     | -0.250                 | -6.000                | 24                  | 1183    |         |                       |                     |          |
| -----    |                        |                       |                     |         |         |                       |                     |          |

Timing constraints are not met.

-----

| Clock Summary

| -----

-----

| Clock  | Waveform(ns)   | Period(ns) | Frequency(MHz) |
|--|----------------|------------|----------------|
| -----  |                |            |                |
| dbg_hub/inst/BSCANID.u_xsdbm_id/SWITCH_N_EXT_BSCAN.bscan_inst/SERIES7_BSCAN.bscan_inst/TCK | {0.000 16.500} | 33.000     | 30.303         |
| sys_clk_pin  | {0.000 4.000}  | 5.000      | 200.000        |

**Рисунок 2.9 – Значения задержек по Setup и Hold для набора проектных ограничений с v<sub>io</sub> с частотой 200 МГц**

Программа успешно завершила свою работу, результаты временных задержек были выведены в соответствующие файлы.

## **ЗАКЛЮЧЕНИЕ**

Таким образом, в данной практической работе был создан файл Tc1, в котором присутствует реализация создания нового проекта с исходными модулями, тестовыми модулями и файлом проектных ограничений, запуск симуляции, синтеза и имплементации на наборах тестов и файлов проектных ограничений. Синтез и имплементация была проведена без и с использованием IP-ядра VIO, с вариациями частот, равными 100 и 200 МГц. Полученные результаты были сохранены в соответствующие файлы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 4 — URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=405132>.
2. Смирнов С.С. Информатика [Электронный ресурс]: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов — М., МИРЭА — Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).
3. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.
4. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).
5. Практическая работа № 3 — URL: [https://online-edu.mirea.ru/pluginfile.php?file=%2F1225652%2Fassignsubmission\\_file%2Fsubmission\\_files%2F3439663%2FПрактика-3.pdf&forcedownload=1](https://online-edu.mirea.ru/pluginfile.php?file=%2F1225652%2Fassignsubmission_file%2Fsubmission_files%2F3439663%2FПрактика-3.pdf&forcedownload=1)