

Университет ИТМО
Факультет программной инженерии и компьютерной техники
Кафедра вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА № 1 ПО ДИСЦИПЛИНЕ
"СХЕМОТЕХНИКА ЭВМ"

ВАРИАНТ: 2

Выполнил: Айтуганов Д. А.

Чебыкин И. Б.

Группа: Р3301

Проверяющий: Баевских А. Н.

Содержание

1	Цели работы	2
2	Задача	2
3	Структура модулей	2
4	Структурная схема RTL-модели	6
5	Структура тестового окружения	8
6	Временные диаграммы	10
7	Вывод	11

1 Цели работы

1. Знакомство с основами проектирования цифровых устройств с использованием языка структурно-функционального описания аппаратуры Verilog HDL.
2. Освоение работы с базовыми дискретными элементами ввода/вывода: светодиоды, позиционные переключатели, семисегментные индикаторы.
3. Знакомство с маршрутом проектирования цифровых схем для ПЛИС.

2 Задача

Реализовать модуль управления трехцветным светодиодом LD17 отладочной платы Nexys-4 DDR.

Светодиод должен в непрерывном режиме отображать следующую цветовую анимацию:

1. Плавно, в течении 1 с, загорается красный светодиод и остается в активном состоянии.
2. Плавно, в течении 1 с, загорается синий светодиод и остается в активном состоянии.
3. Плавно, в течении 1 с, загорается зеленый светодиод и остается в активном состоянии.
4. Плавно, в течении 1 с, снимается индикация с зеленого светодиода.
5. Плавно, в течении 1 с, снимается индикация с синего светодиода.
6. Плавно, в течении 1 с, снимается индикация с красного светодиода.
7. Шаги 1-7 повторяются в циклическом режиме.

Плавною индикацию необходимо реализовать с использованием широтно- импульсной модуляции (ШИМ) сигнала, подаваемого на светодиод определенного цвета. Изменяя скважность сигнала, возможно варьировать яркость светодиода.

3 Структура модулей

```
'timescale 1ns / 1ps
'define TPS (100000)
'define NUM_OF_PIECES 100
'define PIECE ('TPS / 'NUM_OF_PIECES)
module pwm(
    clk,
    reset,
    border,
    out
);

input clk;
input reset;
input border;
output out;

wire clk;
wire reset;
wire signed [31:0] border;
reg out;
```

```

reg signed [31:0] steps_counter = 0;

always @ (posedge clk) begin
    if(reset || steps_counter >= 'PIECE)
        steps_counter = 0;
    else
        steps_counter = steps_counter + 1;

    if(steps_counter < border)
        out = 1;
    else
        out = 0;

end

endmodule

```

Листинг 1: src/pwm.v

```

`timescale 1ns / 1ps
`define TPS (100000)
`define NUM_OF_PIECES 100
`define PIECE ('TPS / 'NUM_OF_PIECES)
module controller(
    clk,
    reset,
    red,
    blue,
    green
);

    input clk;
    input reset;

    output red;
    output blue;
    output green;

    wire clk;
    wire reset;
    wire red;
    wire blue;
    wire green;

    reg [31:0] seconds_counter = 0;
    reg signed [31:0] border = 0;

    pwm red_pwm (
        .clk(clk),
        .reset(reset),
        .border(border),
        .out(red)
    );

    pwm blue_pwm (
        .clk(clk),
        .reset(reset),
        .border(border - 'PIECE),
        .out(blue)
    );

    pwm green_pwm (
        .clk(clk),
        .reset(reset),
        .border(border - 2 * 'PIECE),
        .out(green)
    );
    always @ (posedge clk) begin
        if(reset || (seconds_counter >= 'TPS * 6)) begin
            border = 0;
            seconds_counter = 0;
        end else begin

```

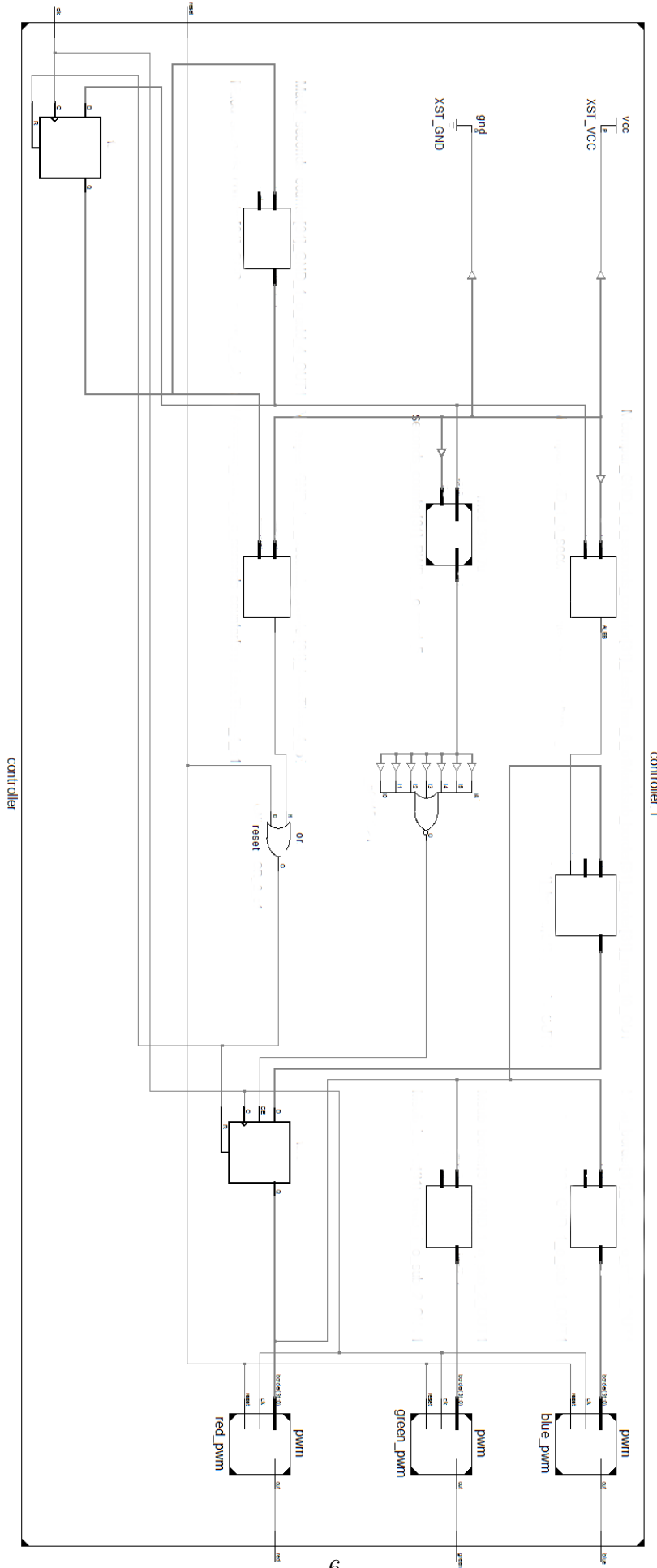
```
seconds_counter = seconds_counter + 1;
if(seconds_counter % 'NUM_OF_PIECES' == 0) begin
    if(seconds_counter >= 'TPS * 3')
        border = border - 1;
    else
        border = border + 1;
end;
end;

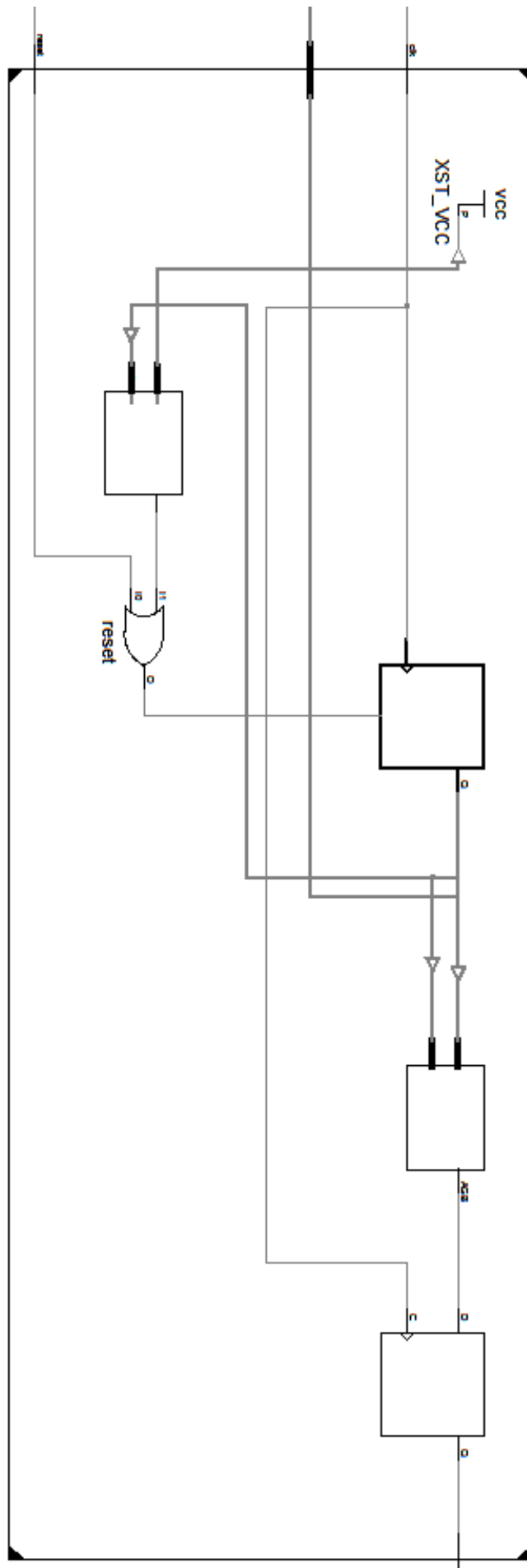
end;

endmodule
```

Листинг 2: src/controller.v

4 Структурная схема RTL-модели





5 Структура тестового окружения

```
'timescale 1ns / 1ps
module test;

    // Inputs
    reg clk;
    reg reset;

    // Outputs
    wire red;
    wire blue;
    wire green;

    // Instantiate the Unit Under Test (UUT)
    controller uut (
        .clk(clk),
        .reset(reset),
        .red(red),
        .blue(blue),
        .green(green)
    );

    event reset_trigger;

    initial begin
        forever begin
            @ (reset_trigger);
            @ (negedge clk);
            reset = 1;
            @ (negedge clk);
            reset = 0;
        end
    end

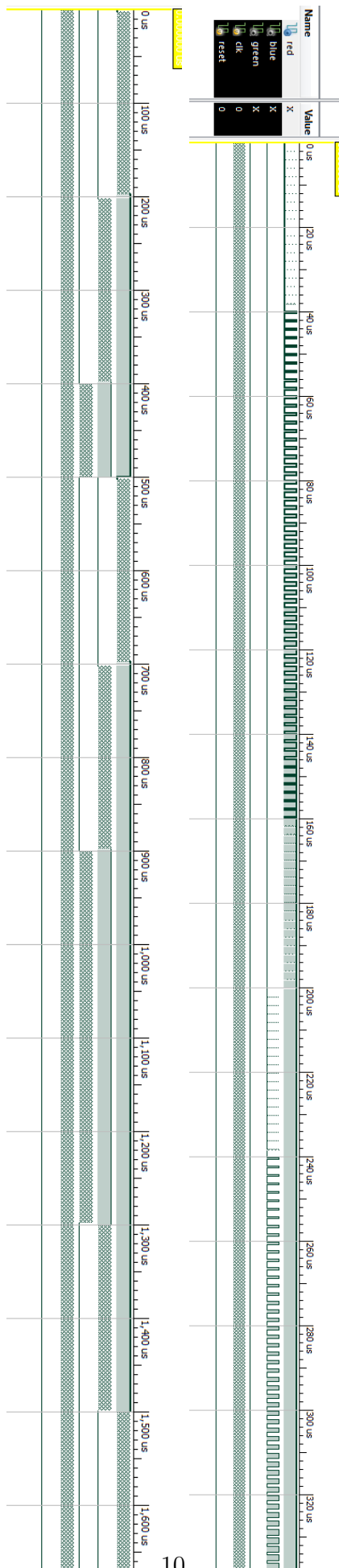
    initial begin
        // Initialize Inputs
        clk = 0;
        reset = 0;

        #500000 -> reset_trigger;
    end

    always begin
        #1 clk = ~clk;
    end
endmodule
```

Листинг 3: src/test.v

6 Временные диаграммы



7 Вывод

В ходе данной лабораторной работы, мы познакомились с основами проектирования цифровых устройств с использованием языка Verilog HDL в среде Xilinx ISE, освоили работу с базовыми дискретными элементами ввода/вывода на примере светодиодов.