Университет ИТМО

Кафедра вычислительной техники

Отчёт по лабораторной работе № 1
по дисциплине: "Схемотехника ЭВМ"
Вариант №5

Студенты:

Преподаватель:

Санкт-Петербург
2016 г.

Содержание

Цели работы

1.

2.

Листинг

```
module switches(
    input           sw0,
    input           sw1,
    output  [1:0]   sw_state
);
    assign sw_state = { sw0, sw1 };
endmodule
```

```
// Module chooses time for animation speed.
module sw_mode(
    input               clk,
    input               rst,
    input       [1:0]   sw_state,

    output  reg[26:0]   tm_value,
    output              count_en
);


    parameter TM_STATE_1 = 100000000;
    parameter TM_STATE_2 = 50000000;
    parameter TM_STATE_3 = 20000000;

    assign count_en = sw_state[0] | sw_state[1];

    always @( posedge clk ) begin
        if( rst )
            tm_value <= 0;
        else
            case( sw_state )
                0: tm_value <= 0;
                1: tm_value <= TM_STATE_1;
                2: tm_value <= TM_STATE_2;
                3: tm_value <= TM_STATE_3;
            endcase
     end

endmodule
```

```
// Module counts time of changing frame
// and sends fc signal to frame_counter module.
// If fc == 1 then frame_counter is incremeted
// and new frame goes on LEDs. Otherwise, nothing
// is happened.
module time_mode(

    input           clk,
    input           rst,
    input   [26:0]  tm_value,
    input           count_en,
    output          fc
);


    reg [26:0] tm_counter;

    assign fc =  count_en & ( tm_counter ==  tm_value ? 1 : 0 );

    always @( posedge clk ) begin
        if( rst )
            tm_counter <= 0;
        else if( tm_counter == tm_value )
            tm_counter <= 0;
        else
            tm_counter <= tm_counter + 27'd1;
    end

endmodule
```

```
// On fc refresh frame.
module frame_counter(
    input           clk,
    input           rst,
    input           fc,
    output [4:0]    fm_no
);

    reg [4:0] counter = 0;
    assign fm_no = counter;

    always @( posedge clk ) begin
        if( rst )
            counter <= 0;
        else if( fc == 1 )
            counter <= counter + 5'b00001;
    end

endmodule
```

```
// Input: frame counter
// Return: next frame
module next_frame (
    input           clk,
    input           rst,
    input   [4:0]   fm_no,
    output  [15:0]  led
);

    reg [15:0] frame;

    assign led = frame;

    always@( posedge clk )
        if( rst )
            frame <= 0;
        else
            case( fm_no )
                0: frame <= 16'b0000000000000000;
                1: frame <= 16'b1000000000000000;
                2: frame <= 16'b1100000000000000;
                3: frame <= 16'b1110000000000000;
                4: frame <= 16'b1111000000000000;
                5: frame <= 16'b1111100000000000;
                6: frame <= 16'b1111110000000000;
                7: frame <= 16'b1111111000000000;
                8: frame <= 16'b1111111100000000;
                9: frame <= 16'b1111111110000000;
                10: frame <= 16'b1111111111000000;
                11: frame <= 16'b1111111111100000;
                12: frame <= 16'b1111111111110000;
                13: frame <= 16'b1111111111111000;
                14: frame <= 16'b1111111111111100;
                15: frame <= 16'b1111111111111110;
                16: frame <= 16'b1111111111111111;
                17: frame <= 16'b1111111111111110;
                18: frame <= 16'b1111111111111100;
                19: frame <= 16'b1111111111111000;
                20: frame <= 16'b1111111111110000;
                21: frame <= 16'b1111111111100000;
                22: frame <= 16'b1111111111000000;
                23: frame <= 16'b1111111110000000;
                24: frame <= 16'b1111111100000000;
                25: frame <= 16'b1111111000000000;
                26: frame <= 16'b1111110000000000;
                27: frame <= 16'b1111100000000000;
                28: frame <= 16'b1111000000000000;
                29: frame <= 16'b1110000000000000;
                30: frame <= 16'b1100000000000000;
                31: frame <= 16'b1000000000000000;
            endcase

endmodule
```

```
`timescale 1ns / 1ps
// Common block
module animation(
    input           clk,
    input           rst,
    input           sw0,
    input           sw1,

    output [15:0]   led
);

    wire [26:0] time_val;
    wire fc;
    wire [4:0]fm_no;
    wire [1:0]sw_state;
    wire count_en;

    switches switches(
        .sw0(sw0),
        .sw1(sw1),

        .sw_state(sw_state)
    );

    sw_mode sw_mode(
        .clk(clk),
        .rst(rst),
        .sw_state(sw_state),

        .tm_value(time_val),
        .count_en(count_en)
    );

    time_mode time_mode(
        .clk(clk),
        .rst(rst),
        .tm_value(time_val),
        .count_en(count_en),

        .fc(fc)
    );

    frame_counter frame_counter(
        .clk(clk),
        .rst(rst),
        .fc(fc),

        .fm_no(fm_no)
    );

    next_frame next_frame(
        .clk(clk),
        .rst(rst),
        .fm_no(fm_no),

        .led(led)
    );

endmodule
```

Вывод