

Университет ИТМО
Факультет программной инженерии и компьютерной техники
Кафедра вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА № 2 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	3
5	Структура тестового окружения	3
6	Временные диаграммы	4
7	Вывод	4

1 Цели работы

1. Знакомство с принципами работы последовательных интерфейсов ввода/вывода: I2C, SPI.
2. Изучение основ разработки аппаратных контроллеров периферийных устройств.
3. Изучение основ работы с цифровыми датчиками.

2 Задача

Разработать контроллер датчика освещенности PmodALS.

Контроллер должен реализовывать следующие функции:

- Поддерживать обмен данными с датчиком освещенности посредством интерфейса SPI
- При значении переключателя $SW[0] = 0$ показывать факт наличия света в аудитории с помощью светодиодов: $LED[15:0] = 0xFFFF$ – свет выключен, $LED[15:0] = 0x0000$ – свет включен. Таким образом, все светодиоды должны загораться, когда свет в аудитории выключается, и выключаться – когда свет в аудитории есть.
- При значении переключателя $SW[0] = 1$ показывать на светодиодах текущее значение освещенности, считанное с датчика

3 Структура модулей

```
'timescale 1ns / 1ps
'define START 2
'define END 11
module reader(
    clk,
    sdo,
    data,
    cs,
    sck
);

input clk;
input sdo; //for PmodALS

output cs; //for PmodALS
output sck; //for PmodALS
output data;

reg sck;
reg cs;
reg[7:0] tmp = 0;
reg[7:0] data = 0;
reg[3:0] counter = 15;

// Зацикливается
//always sck <= clk;

always @ (negedge clk) begin
    if(counter == 15) begin
        counter = 0;
        cs <= 1;
    end else begin
        cs <= 0;
        //0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
        //0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
        if(counter > 'START && counter < 'END) begin
```

```

        tmp = tmp << 1;
        tmp[0] = sdo;
    end else if(counter == 'END)
        data = tmp;
        counter = counter + 1;
    end
end;

endmodule

```

Листинг 1: src/reader.v

```

`timescale 1ns / 1ps

module write_led(
    input clk,
    input sw,
    input [7:0] data,
    output reg[15:0] led = 16'hFFFF
);
    always @(negedge clk) begin
        if( sw == 0 ) begin
            if( data > 0 ) begin
                led <= 16'h0000;
            end;
        end else begin
            led <= data;
        end;
    end;
end;

endmodule

```

Листинг 2: src/write_led.v

4 Структурная схема RTL-модели

5 Структура тестового окружения

```

`timescale 1ns / 1ps

module test;

    // Inputs
    reg clk;
    reg sdo;
    reg sw;
    wire [7:0] data;

    reg cs;
    reg sck;
    reg [15:0] test_input = 16'b0001111111100000;
    reg [4:0] i = 15;

    wire [15:0] led;

    reader read (
        .clk(clk),
        .sdo(sdo),
        .data(data),
        .cs(cs),
        .sck(sck)
    );
    // Instantiate the Unit Under Test (UUT)
    write_led uut (
        .clk(clk),
        .sw(sw),
        .data(data),
        .led(led)
    );

    initial begin

```

```

    // Initialize Inputs
    clk = 0;
    sw = 0;
    //sw = 1;
    sdo = 0;
    cs = 0;
end

always begin
    #1 clk = ~clk;
end

always @(posedge clk) begin
    sdo = test_input[i];
    i = i - 1;
end
endmodule

```

Листинг 3: src/test.v

6 Временные диаграммы

7 Вывод