

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	4
5	Структура тестового окружения	5
6	Временные диаграммы	6
7	Вывод	6

# 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 3
'define END 12
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;

always @ (clk) sck = clk;

always @ (negedge clk) begin
    if(counter == 15) begin
        counter = 0;
        cs = 1;
    end else begin
        cs = 0;
        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(
    sdo,
    clk,
    sw,
    cs,
    sck,
    led
);

    input sdo; //for PmodALS
    input clk;
    input sw;

    output cs; //for PmodALS
    output sck; //for PmodALS
    output led;

    wire[7:0] data;

    reg[15:0] led = 16'hFFFF;

    reader read (
        .clk(clk),
        .sdo(sdo),
        .data(data),
        .cs(cs),
        .sck(sck)
    );

    always @(negedge clk) begin
        if( sw == 0 ) begin
            if( data > 0 )
                led <= 16'h0000;
            else
                led <= 16'hFFFF;

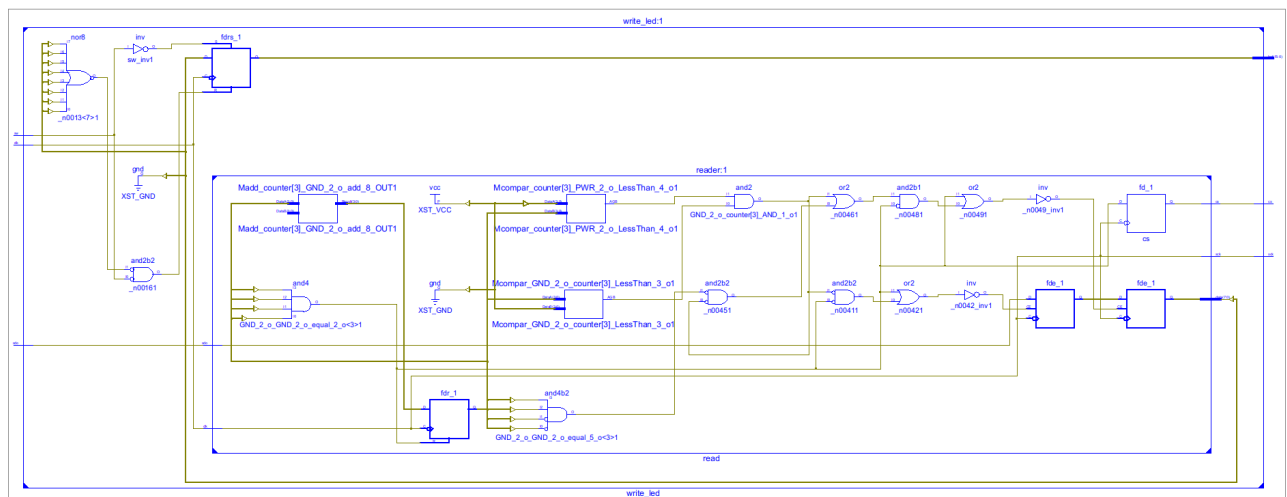
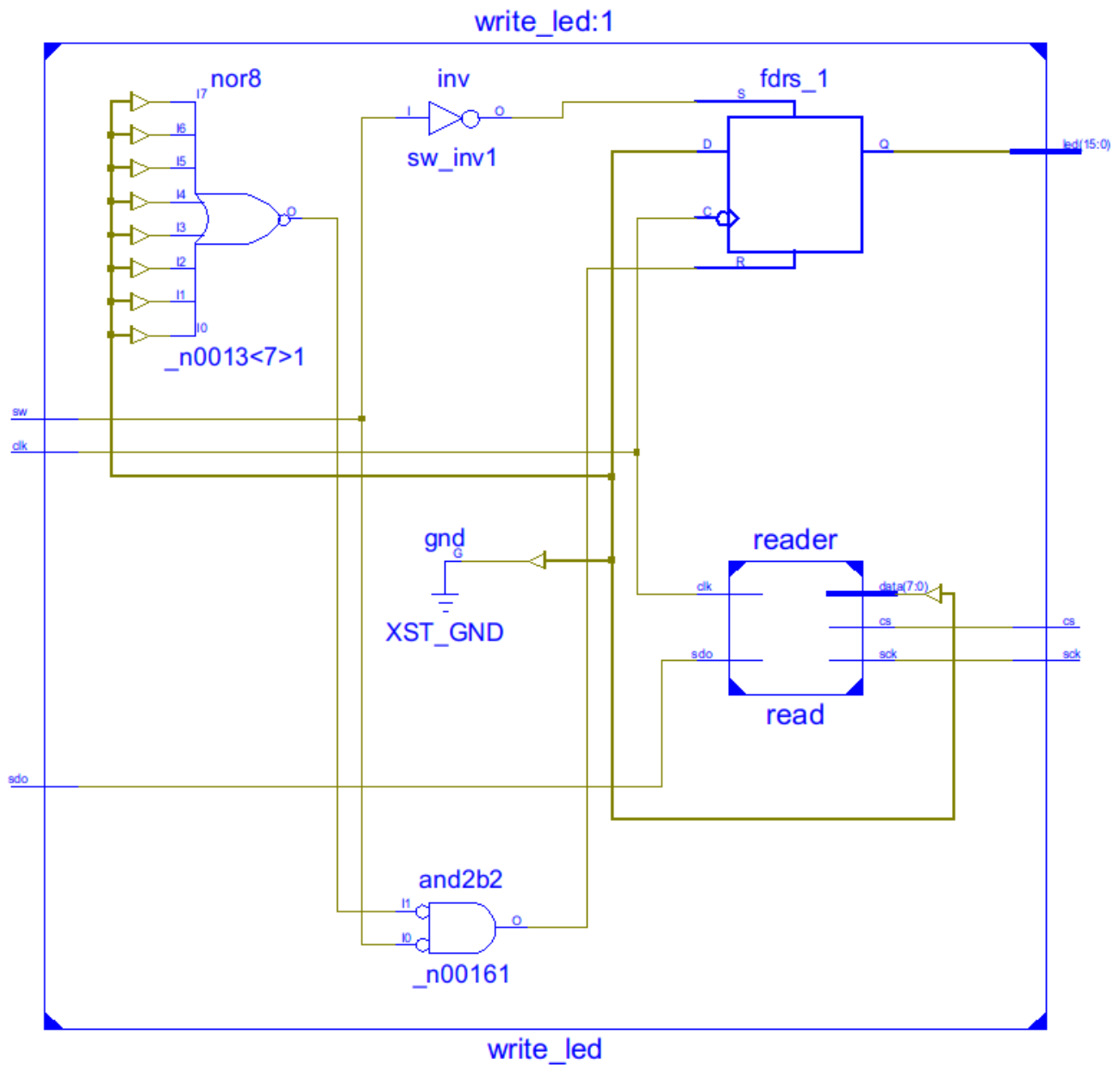
            end else begin
                led <= data;
            end;
        end;
    end;

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;
    reg pf;
    reg [15:0] test_input = 16'b0001111011100000;
    reg [3:0] i;

    wire [15:0] led;
    wire sck;

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

    event sw_trigger;

    initial begin
        forever begin
            @ (sw_trigger);
            @ (posedge clk);
            sw = ~sw;
        end
    end

    initial begin
        // Initialize Inputs
        pf = 0;
        clk = 0;
        sw = 0;
        sdo = 0;
    end

    always begin
        #1 clk = ~clk;
    end

    always begin
        #100 sw <= ~sw;
    end

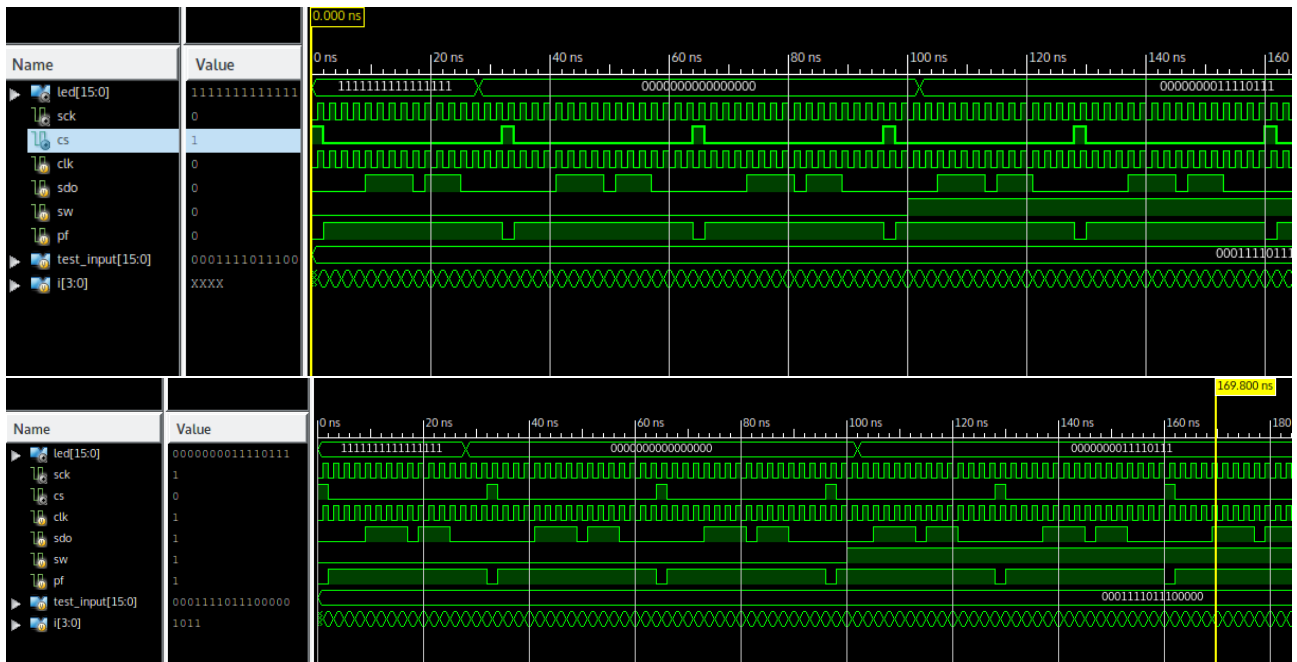
    always @(posedge cs)
        pf = 0;

    always @(negedge cs)
        pf = 1;

    always @(posedge sck) begin
        if(pf && i > 0) begin
            sdo = test_input[i];
            i = i - 1;
        end else
            i = 15;
    end
endmodule
```

Листинг 3: src/test.v

## 6 Временные диаграммы



## 7 Вывод

В ходе данной лабораторной работы мы ознакомились с принципами работы последовательных интерфейсов ввода/вывода на примере SPI, а также мы изучили основы работы с цифровыми датчиками и научились разрабатывать аппаратные контроллеры периферийных устройств.