**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационной безопасности**

отчет

**по лабораторной работе №5**

**по дисциплине «Электроника и схемотехника»**

**Тема: «Коммуникация по интерфейсу SPI»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3363 |  | Минко Д.А. |
| Студент гр. 3363 |  | Овсейчик Н.И. |
| Студент гр. 3363 |  | Гончаренко О.Д. |
| Преподаватель |  | Рыбин В.Г. |

Санкт-Петербург

2024

**Цель работы**

Создать два проекта для ведущего и ведомого устройств (для удобства последующей прошивки). В каждом из них описать отдельно модуль SPI и поместить его в оболочку, которая будет передавать сигналы в модуль и выводить передаваемые данные на светодиоды.

**Ход работы**

1. В проекте для ведущей платы разработан модуль SPI, реализующий управление состояниями передачи и ожидания, а также контроль успешности передачи пакетов. Этот модуль интегрирован в оболочку, обеспечивающую инкрементирование значений и их передачу с использованием кнопок.
2. В отдельном проекте создан модуль SPI для ведомой платы, который также помещен в оболочку. Данная оболочка реализует передачу значений регистра на светодиоды.
3. Для ведущего и ведомого устройств выполнена отдельная разводка. Схемы разводки идентичны, за исключением того, что на ведомом устройстве добавлены кнопки: одна для увеличения двоичного значения числа, другая для выполнения передачи. На ведущем устройстве значение не сбрасывается автоматически и требует ручного обнуления.
4. При конфигурации устройств необходимо установить режим «Embedded Flash Mode» на обеих платах, чтобы обеспечить сохранение прошивки. Платы дополнительно соединены общей линией GND и питающим пином на 5V.

**Вывод**

В ходе выполнения лабораторной работы были разработаны два SPI-модуля: для Master-устройства и для Slave-устройства. Оба модуля интегрированы в оболочки, обеспечивающие передачу двоичного значения от ведущего устройства к ведомому.

ПРИЛОЖЕНИЕ 1 – ИСХОДНЫЙ КОД ПРОГРАММЫ

1. Основной модуль master\_control

module master\_control

(

input clk,

input increment,

input transmit,

output [5:0] led,

output SCLK,

output MOSI,

input MISO,

output SS

);

reg [7:0] data\_out;

wire [7:0] data\_in;

reg [19:0] start\_counter;

wire start;

reg [19:0] inc\_counter;

spi\_master m (clk, start, data\_out, data\_in, SCLK, MOSI, MISO, SS);

always @(posedge clk)

begin

if (!increment)

begin

if (!inc\_counter[19])

begin

inc\_counter = inc\_counter + 1;

if (inc\_counter[19])

data\_out = data\_out + 1;

end

end

else

begin

inc\_counter <= 0;

end

end

always @(posedge clk)

begin

if (!transmit)

begin

if (!start\_counter[19])

start\_counter <= start\_counter + 1;

end

else

begin

start\_counter <= 0;

end

end

assign led = ~data\_out[5:0];

assign start = start\_counter[19];

endmodule

1. Вспомогательный модуль spi\_master

module spi\_master

(

input clk,

input start,

input [7:0] data\_out,

output reg [7:0] data\_in,

output reg SCLK,

output reg MOSI,

input MISO,

output reg SS

);

reg [1:0] state;

reg [2:0] data\_counter;

initial

begin

data\_in <= 0;

state <= 0;

SS <= 1;

MOSI <= 0;

SCLK <= 0;

data\_counter <= 0;

end

always @(posedge clk)

begin

case (state)

0:

begin

if (start) // If not transmitting but commanded to start

begin

state <= 1;

SS <= 0;

SCLK <= 0;

data\_counter <= 7;

MOSI <= data\_out[7];

end

end

1: // If transmitting

begin

if (!SCLK) // If making a posedge

begin

data\_in[data\_counter] <= MISO;

if (!data\_counter)

state <= 2;

else

data\_counter <= data\_counter - 1'b1;

end

else

begin

MOSI <= data\_out[data\_counter];

end

SCLK <= ~SCLK;

end

2: // If packets sent

begin

SCLK <= 0;

state <= 3;

MOSI <= 0;

end

3: // If ending transmission

begin

SS <= 1;

state <= 0;

end

endcase

end

endmodule

1. Распиновка для master\_control

//Copyright (C)2014-2024 Gowin Semiconductor Corporation.

//All rights reserved.

//File Title: Physical Constraints file

//Tool Version: V1.9.9.03 Education (64-bit)

//Part Number: GW1NR-LV9QN88PC6/I5

//Device: GW1NR-9

//Device Version: C

//Created Time: Sun 09 22 13:27:35 2024

IO\_LOC "SS" 28;

IO\_PORT "SS" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "MOSI" 26;

IO\_PORT "MOSI" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "SCLK" 27;

IO\_PORT "SCLK" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[5]" 16;

IO\_PORT "led[5]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[4]" 15;

IO\_PORT "led[4]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[3]" 14;

IO\_PORT "led[3]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[2]" 13;

IO\_PORT "led[2]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[1]" 11;

IO\_PORT "led[1]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[0]" 10;

IO\_PORT "led[0]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "transmit" 4;

IO\_PORT "transmit" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;

IO\_LOC "increment" 3;

IO\_PORT "increment" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;

IO\_LOC "clk" 52;

IO\_PORT "clk" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;

IO\_LOC "MISO" 25;

IO\_PORT "MISO" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;

1. Основной модуль slave\_control

module slave\_control

(

output [5:0] led,

input SCLK,

input MOSI,

output MISO,

input SS

);

reg [7:0] data\_out;

wire [7:0] data\_in;

spi\_slave s (data\_out, data\_in, SCLK, MOSI, MISO, SS);

assign led = ~data\_in[5:0];

endmodule

1. Вспомогательный модуль spi\_slave

module spi\_slave

(

input [7:0] data\_out,

output reg [7:0] data\_in,

input SCLK,

input MOSI,

output MISO,

input SS

);

reg state;

reg [2:0] data\_counter;

initial

begin

data\_in <= 0;

state <= 0;

data\_counter <= 0;

end

always @(posedge SCLK)

begin

if (!SS)

begin

if (!state)

begin

state = 1;

data\_counter = 7;

end

data\_in = { data\_in[6:0], MOSI };

if (data\_counter == 0) state = 0;

data\_counter = data\_counter - 1;

end

end

assign MISO = state;

endmodule

1. Распиновка для slave\_control

//Copyright (C)2014-2024 Gowin Semiconductor Corporation.

//All rights reserved.

//File Title: Physical Constraints file

//Tool Version: V1.9.9.03 Education (64-bit)

//Part Number: GW1NR-LV9QN88PC6/I5

//Device: GW1NR-9

//Device Version: C

//Created Time: Sun 09 22 13:47:32 2024

IO\_LOC "MISO" 25;

IO\_PORT "MISO" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[5]" 16;

IO\_PORT "led[5]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[4]" 15;

IO\_PORT "led[4]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[3]" 14;

IO\_PORT "led[3]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[2]" 13;

IO\_PORT "led[2]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[1]" 11;

IO\_PORT "led[1]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "led[0]" 10;

IO\_PORT "led[0]" IO\_TYPE=LVCMOS18 PULL\_MODE=UP DRIVE=8 BANK\_VCCIO=1.8;

IO\_LOC "SS" 28;

IO\_PORT "SS" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;

IO\_LOC "MOSI" 26;

IO\_PORT "MOSI" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;

IO\_LOC "SCLK" 27;

IO\_PORT "SCLK" IO\_TYPE=LVCMOS18 PULL\_MODE=UP BANK\_VCCIO=1.8;