|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Московский технологический университет" **МИРЭА** | |
| **Институт информационных технологий (ИТ)** | |
| Кафедра вычислительной техники (ВТ) | |

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **О выполнении практического задания №2** | | |
| **по дисциплине** | | |
| **«**Моделирование**»**  *Вариант 20* | | |
|  | | |
| Выполнил студент группы ИВБО-03-14 | *Сафронов И. ?.* | |
| Преподаватель | *Борисенко Н.В.* | |

Москва 2017г

Введение

Техническое задание**.**

Спроектировать устройство выводящее свое состояние на семисегментные и на LED индикаторы. Устройство должно поочередно отобразить все биты из таблицы варианта, переключение состояния осуществляется нажатием на кнопку, подключаемую ко входу устройства.

Реализовать на языке Verilog тестовое окружение и провести верификацию спроектированных моделей при помощи симулятора iSim из состава САПР Xilinx ISE Design Suite.

Провести апробацию моделей при помощи отладочной платы Digilent Nexys 4 на ПЛИС Xilinx Artix 7 XC7A100T-1CSG324. Комбинации на входах комбинационных схем должны задаваться при помощи движковых переключателей отладочной платы, комбинации на выходах комбинационных схем должны отображаться светодиодами отладочной платы.

Таблица варианта**.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вар\Вх.** | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **24** | 4 | 7 | E | C | 6 | D | B | 7 | C | 5 | E | C | 0 | 2 | A | F |

Табл.1 Вариант задания.

Описание устройства

Назначение входов и выходов

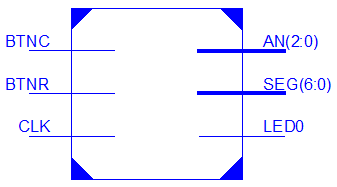


Рис. 1 Проектируемое устройство

Назначение входов:

* BTNC – сигнал перехода в следующее состояние
* BTNR – сигнал сброса состояния устройства
* CLK – синхросигнал

Назначение выходов:

* AN[2:0] – шина выбора семисегментного индикатора
* SEG[6:0] – шина значения семисегментного индикатора
* LED0 – сигнал активации LED индикатора

Реализация устройства на языке Verilog

Верхний модуль «TOP.v»

`timescale 1ns / 1ps

module Top(

input CLK,

input BTNC,

input BTNR,

output [6:0] SEG,

output [2:0] AN,

output reg LED0

);

wire BTN\_IN;

wire [3:0] EL;

wire [3:0] BIT\_4;

reg [3:0] NOM;

reg [1:0] BIT;

Filter CL1(CLK, BTNC, BTNR, BTN\_IN);

EL\_Decoder CL2(NOM, EL);

Dysplay CL3(CLK, BTNR, NOM, BIT\_4, EL, AN, SEG);

assign BIT\_4[1:0] = BIT;

assign BIT\_4[3:2] = 0;

always @ (posedge CLK, posedge BTNR) begin

if(BTNR) begin

NOM <= 0;

BIT <= 0;

end else begin

if(BTN\_IN) begin

BIT <= BIT + 1;

if(BIT == 3)

NOM <= NOM + 1;

end

end

case (BIT)

2'b00: LED0 <= EL[0];

2'b01: LED0 <= EL[1];

2'b10: LED0 <= EL[2];

2'b11: LED0 <= EL[3];

endcase

end

endmodule

Фильтр дребезга контактов «Filter.v»

`timescale 1ns / 1ps

module Filter(

input CLK,

input BTN\_IN,

input RST,

output reg BTN\_CEO

);

parameter [3:0] CNTR\_WIDTH = 4;

reg [CNTR\_WIDTH - 1:0] FLTR\_CNT;

reg BTN\_D, BTN\_S1, BTN\_S2;

always @ (posedge CLK, posedge RST) begin

if(RST)

FLTR\_CNT <= 0;

else if(BTN\_S1 == BTN\_S2)

FLTR\_CNT <= 0;

else

FLTR\_CNT <= FLTR\_CNT + 1;

end

always @ (posedge CLK, posedge RST) begin

if(RST) begin

BTN\_D <= 1'b0;

BTN\_S1 <= 1'b0;

end else begin

BTN\_D <= BTN\_IN;

BTN\_S1 <= BTN\_D;

end

end

always @ (posedge CLK, posedge RST) begin

if(RST)

BTN\_S2 <= 1'b0;

else if(&(FLTR\_CNT))

BTN\_S2 <= BTN\_S1;

end

always @ (posedge CLK, posedge RST) begin

if(RST)

BTN\_CEO <= 1'b0;

else

BTN\_CEO <= &(FLTR\_CNT) & BTN\_S1;

end

endmodule

Блок преобразования числа для вывода «Decoder.v»

`timescale 1ns / 1ps

module SEG\_Decoder(

input [3:0] NUM,

output reg [6:0] SEG

);

always @ (NUM)

case (NUM)

4'b0000: SEG <= 7'b0001000;

4'b0001: SEG <= 7'b1101101;

4'b0010: SEG <= 7'b0100010;

4'b0011: SEG <= 7'b0100100;

4'b0100: SEG <= 7'b1000101;

4'b0101: SEG <= 7'b0010100;

4'b0110: SEG <= 7'b0010000;

4'b0111: SEG <= 7'b0101101;

4'b1000: SEG <= 7'b0000000;

4'b1001: SEG <= 7'b0000100;

4'b1010: SEG <= 7'b0000001;

4'b1011: SEG <= 7'b1010000;

4'b1100: SEG <= 7'b0011010;

4'b1101: SEG <= 7'b1100000;

4'b1110: SEG <= 7'b0010010;

4'b1111: SEG <= 7'b0010011;

default: SEG <= 7'b1111111;

endcase

endmodule

Делитель частоты «Divider.v»

`timescale 1ns / 1ps

module Divider(

input CLK,

input RST,

output CEO

);

parameter [9:0] N = 10'd100;

reg [9:0] STATE;

always @ (posedge CLK, posedge RST) begin

if(RST)

STATE <= 0;

else begin

STATE <= STATE + 1;

if(STATE == N - 1)

STATE <= 0;

end

end

assign CEO = &(~STATE);

endmodule

Блок вывода чисел на семисегментник «Dysplay.v»

`timescale 1ns / 1ps

module Dysplay(

input CLK,

input RST,

input [3:0] NUM1,

input [3:0] NUM2,

input [3:0] NUM3,

output reg [2:0] AN,

output [6:0] SEG

);

reg [1:0] NUM;

reg [3:0] ToDysplay;

Divider CL1(CLK, RST, CEO);

SEG\_Decoder CL2(ToDysplay, SEG);

always @ (posedge CEO, posedge RST) begin

if(RST)

NUM <= 0;

else begin

case(NUM)

2'b00: begin

NUM <= 2'b01;

AN <= 3'b011;

ToDysplay <= NUM1;

end

2'b01: begin

NUM <= 2'b10;

AN <= 3'b101;

ToDysplay <= NUM2;

end

2'b10: begin

NUM <= 2'b00;

AN <= 3'b110;

ToDysplay <= NUM3;

end

endcase

end

end

endmodule

Блок преобразования номера по заданию «EL\_Decoder.v»

`timescale 1ns / 1ps

module EL\_Decoder(

input [3:0] NOM,

output reg [3:0] Y

);

always @ (NOM)

case(NOM)

4'h0: Y <= 4'hF;

4'h1: Y <= 4'hA;

4'h2: Y <= 4'h2;

4'h3: Y <= 4'h0;

4'h4: Y <= 4'hC;

4'h5: Y <= 4'hE;

4'h6: Y <= 4'h5;

4'h7: Y <= 4'hC;

4'h8: Y <= 4'h7;

4'h9: Y <= 4'hB;

4'hA: Y <= 4'hD;

4'hB: Y <= 4'h6;

4'hC: Y <= 4'hC;

4'hD: Y <= 4'hE;

4'hE: Y <= 4'h7;

4'hF: Y <= 4'h4;

endcase

endmodule

Тестирование устройства в симуляторе ISim

Тестовый модуль «Test.v»

`timescale 1ns / 1ps

module Test;

// Inputs

reg CLK;

reg BTNC;

reg BTNR;

// Outputs

wire [6:0] SEG;

wire [2:0] AN;

wire LED0;

// Instantiate the Unit Under Test (UUT)

Top uut (

.CLK(CLK),

.BTNC(BTNC),

.BTNR(BTNR),

.SEG(SEG),

.AN(AN),

.LED0(LED0)

);

always begin

CLK = 1'b0; #25;

CLK = 1'b1; #25;

end

always begin

BTNC = 1;

#20000;

BTNC = 0;

#20000;

end

initial begin

BTNR = 1; #100;

BTNR = 0;

end

endmodule

Временная диаграмма

