----------------------------------------------\*/

原代码1:：上板程序（显示单位输入+输出结果）检测111

16位拨码开关输入X，1秒CLK时钟

----------------------------------------------\*/

module sequential\_detector(clk, reset, din, y,dout);

//输入输出端口定义

input clk, reset;

input [15:0] din;

output y,dout;

//内部寄存器及连线定义

reg [2 : 0] state;

wire y,dout;

reg d;

wire [15:0] dd;

reg [15:0] dd\_mid;

parameter update\_interval = 50\_000\_000;

integer selcnt;

reg clk100;

//状态编码

parameter idle = 3'b000, s1 = 3'b001, s2 = 3'b010, s3 = 3'b011,

s4 = 3'b100, s5 = 3'b101, s6 = 3'b110, s7 = 3'b111;

//状态机实现

always @(posedge clk) //100Hz

begin

if (selcnt == update\_interval)

begin

selcnt <= 0;

clk100 <=~clk100;

end

else

selcnt<=selcnt+1;

end

/\*

always@(posedge clk100 or posedge reset )

begin

if(reset)

begin

dd\_mid <= din;

end

else

begin

dd\_mid = {dd\_mid[14 : 0], dd\_mid[15]};

d = dd\_mid[15]; //将最高位输入到序列检测器中

end

end

assign dd = dd\_mid;

assign dout = d;

\*/

always@(posedge clk100 or posedge reset)

begin

if(reset)

begin

state <= idle;

dd\_mid <= din;

end

else

begin

dd\_mid = {dd\_mid[14 : 0], dd\_mid[15]};

d = dd\_mid[15]; //将最高位输入到序列检测器中

case(state)

idle: begin

if(d == 1) state <= s1;

else state <= idle;

end

s1: begin

if(d == 1) state <= s2;

else state <= idle;

end

s2: begin

if(d == 1) state <= s3;

else state <= idle;

end

s3: begin

if(d == 1) state <= s3;

else state <= idle;

end

default: state <= idle;

endcase

end

end

//用组合逻辑实现输出

// assign y = (state == s3 ) ? 1 : 0; // 莫尔机

assign y = (state == s3 && dout == 1 ) ? 1 : 0; //米利机

endmodule

================ xdc ==============

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[15]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[14]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[13]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[12]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[11]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[10]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[9]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[8]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {din[0]}]

set\_property PACKAGE\_PIN R2 [get\_ports {din[15]}]

set\_property PACKAGE\_PIN T1 [get\_ports {din[14]}]

set\_property PACKAGE\_PIN U1 [get\_ports {din[13]}]

set\_property PACKAGE\_PIN W2 [get\_ports {din[12]}]

set\_property PACKAGE\_PIN R3 [get\_ports {din[11]}]

set\_property PACKAGE\_PIN T2 [get\_ports {din[10]}]

set\_property PACKAGE\_PIN T3 [get\_ports {din[9]}]

set\_property PACKAGE\_PIN V2 [get\_ports {din[8]}]

set\_property PACKAGE\_PIN W13 [get\_ports {din[7]}]

set\_property PACKAGE\_PIN W14 [get\_ports {din[6]}]

set\_property PACKAGE\_PIN V15 [get\_ports {din[5]}]

set\_property PACKAGE\_PIN W15 [get\_ports {din[4]}]

set\_property PACKAGE\_PIN W17 [get\_ports {din[3]}]

set\_property PACKAGE\_PIN W16 [get\_ports {din[2]}]

set\_property PACKAGE\_PIN V16 [get\_ports {din[1]}]

set\_property PACKAGE\_PIN V17 [get\_ports {din[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]

set\_property PACKAGE\_PIN W5 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports reset]

set\_property PACKAGE\_PIN U18 [get\_ports reset]

set\_property IOSTANDARD LVCMOS33 [get\_ports y]

set\_property PACKAGE\_PIN L1 [get\_ports y]

set\_property IOSTANDARD LVCMOS33 [get\_ports dout]

set\_property PACKAGE\_PIN U16 [get\_ports dout]

----------------------------------------------\*/

----------------------------------------------\*/

原代码2：上板程序（显示单位输入+输出结果）检测111

单一拨码开关输入X，2秒CLK时钟

----------------------------------------------\*/

`timescale 1ns / 1ps

module sequential\_detector(clk, reset, din, y,dout);

//输入输出端口定义

input clk, reset;

input [15:0] din;

output y,dout;

//内部寄存器及连线定义

reg [2 : 0] state;

wire y;

wire dout;

reg d;

parameter update\_interval = 100\_000\_000;

integer selcnt;

reg clk100;

//状态编码

parameter idle = 3'b000, s1 = 3'b001, s2 = 3'b010, s3 = 3'b011,

s4 = 3'b100, s5 = 3'b101, s6 = 3'b110, s7 = 3'b111;

//状态机实现

always @(posedge clk) //100Hz

begin

if (selcnt == update\_interval)

begin

selcnt <= 0;

clk100 <=~clk100;

end

else

selcnt<=selcnt+1;

end

assign dout= din[15];

always@(posedge clk100)

begin

d = din[15];

case(state)

idle: begin

if(d == 1) state <= s1;

else state <= idle;

end

s1: begin

if(d == 1) state <= s2;

else state <= idle;

end

s2: begin

if(d == 1) state <= s3;

else state <= idle;

end

s3: begin

if(d == 1) state <= s3;

else state <= idle;

end

default: state <= idle;

endcase

end

//用组合逻辑实现输出

// assign y = (state == s3 ) ? 1 : 0; // 莫尔机

assign y = (state == s3 && dout == 1 ) ? 1 : 0; //米利机

endmodule