Verilog作业6

2211290 姚知言

1 作业要求

请使用Verilog编写模块，在龙芯实验箱或口袋实验板上完成数码管的数字显示，要求如下：

(1)数码管至少要显示两位不同的数，建议（2-4位）

(2)数码管初始显示0，然后拨一个拨码开关（或按一个按键）数码管显示+1计数。

(3)按reset按键数码管显示数字归零。

(4)其他功能同学们可以自行添加。（比如设置最大计数为60，数到59之后归零，然后用一个led灯表示溢出）

2 代码实现及说明

(1)设计源文件count.v

`timescale 1ns / 1ps

module count(

输入包括一个时钟clk，

turn即为+1按键，每按一次自增1

reset清零信号，使得显示数字归0

seg和sel位数码管输出和选通

led为溢出信号，当数码管数字为59时，再按动将显示数字清零，并点亮溢出信号。

input clk,

input turn,

input reset,

output reg [7:0] seg,

output reg [3:0] sel,

output reg led=1'b0

);

cnt用于计数，每当turn按钮上升沿，turn\_e=1，更新计数器。

clk\_div和slow\_clk用于设计分频器，把频率放慢50000倍以保证稳定显示。

reg [7:0] cnt = 8'd0;

reg p\_turn;

wire turn\_e = (~p\_turn & turn);

reg [15:0] clk\_div;

wire slow\_clk = (clk\_div == 16'd50000);

always @(posedge clk) begin

if(!reset) begin

reset清空计数，led等，并初始化sel。

cnt<= 8'd0;

led<= 1'b0;

sel<= 4'b1000;

end

clk\_div <= clk\_div + 1;

分频器实现。

if (clk\_div >= 16'd50000)

clk\_div <= 0;

p\_turn <= turn;

if (turn\_e) begin

if (cnt == 8'd59)begin

cnt <= 8'd0;

led <= 1'b1;

end

计数器实现。

else begin

cnt <= cnt + 1;

led <= 1'b0;

end

end

if (slow\_clk) begin

if (sel == 4'b1000) begin // 显示十进制个位

sel <= 4'b0001;

case (cnt % 10)

4'd0: seg <= 8'b0111\_1110; // 0

4'd1: seg <= 8'b0011\_0000; // 1

数码管的位切换及显示。

共使用四个数码管，前两位显示计数器的十进制表示的十位和个位，后两位显示计数器的十六进制表示的低位和高位。

4'd2: seg <= 8'b0110\_1101; // 2

4'd3: seg <= 8'b0111\_1001; // 3

4'd4: seg <= 8'b0011\_0011; // 4

4'd5: seg <= 8'b0101\_1011; // 5

4'd6: seg <= 8'b0101\_1111; // 6

4'd7: seg <= 8'b0111\_0000; // 7

4'd8: seg <= 8'b0111\_1111; // 8

4'd9: seg <= 8'b0111\_1011; // 9

default: seg <= 8'b1000\_0000; // .

endcase

end

else if(sel == 4'b0001) begin // 显示十进制十位

sel <= 4'b0010;

case (cnt / 10)

4'd0: seg <= 8'b0111\_1110; // 0

4'd1: seg <= 8'b0011\_0000; // 1

4'd2: seg <= 8'b0110\_1101; // 2

4'd3: seg <= 8'b0111\_1001; // 3

4'd4: seg <= 8'b0011\_0011; // 4

4'd5: seg <= 8'b0101\_1011; // 5

4'd6: seg <= 8'b0101\_1111; // 6

4'd7: seg <= 8'b0111\_0000; // 7

4'd8: seg <= 8'b0111\_1111; // 8

4'd9: seg <= 8'b0111\_1011; // 9

default: seg <= 8'b1000\_0000; // .

endcase

end

else if(sel == 4'b0010) begin // 显示十六进制低位

sel <= 4'b0100;

case (cnt % 5'd16)

4'd0: seg <= 8'b0111\_1110; // 0

4'd1: seg <= 8'b0011\_0000; // 1

4'd2: seg <= 8'b0110\_1101; // 2

4'd3: seg <= 8'b0111\_1001; // 3

4'd4: seg <= 8'b0011\_0011; // 4

4'd5: seg <= 8'b0101\_1011; // 5

4'd6: seg <= 8'b0101\_1111; // 6

4'd7: seg <= 8'b0111\_0000; // 7

4'd8: seg <= 8'b0111\_1111; // 8

4'd9: seg <= 8'b0111\_1011; // 9

4'd10: seg <= 8'b0111\_0111; // A

4'd11: seg <= 8'b0001\_1111; // B

4'd12: seg <= 8'b0100\_1110; // C

4'd13: seg <= 8'b0011\_1101; // D

4'd14: seg <= 8'b0001\_1111; // E

4'd15: seg <= 8'b0100\_0111; // F

default: seg <= 8'b0100\_1111; // .

endcase

end

else begin // 显示十六进制高位

sel <= 4'b1000;

case (cnt / 5'd16)

4'd0: seg <= 8'b0111\_1110; // 0

4'd1: seg <= 8'b0011\_0000; // 1

4'd2: seg <= 8'b0110\_1101; // 2

4'd3: seg <= 8'b0111\_1001; // 3

4'd4: seg <= 8'b0011\_0011; // 4

4'd5: seg <= 8'b0101\_1011; // 5

4'd6: seg <= 8'b0101\_1111; // 6

4'd7: seg <= 8'b0111\_0000; // 7

4'd8: seg <= 8'b0111\_1111; // 8

4'd9: seg <= 8'b0111\_1011; // 9

4'd10: seg <= 8'b0111\_0111; // A

4'd11: seg <= 8'b0001\_1111; // B

4'd12: seg <= 8'b0100\_1110; // C

4'd13: seg <= 8'b0011\_1101; // D

4'd14: seg <= 8'b0001\_1111; // E

4'd15: seg <= 8'b0100\_0111; // F

default: seg <= 8'b1000\_0000; // .

endcase

end

end

end

endmodule

(2)约束文件mycon.xdc

set\_property PACKAGE\_PIN R11 [get\_ports turn]

set\_property IOSTANDARD LVCMOS33 [get\_ports turn]//+1按钮

set\_property PACKAGE\_PIN P17 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk]//时钟

set\_property PACKAGE\_PIN P15 [get\_ports reset]

set\_property IOSTANDARD LVCMOS33 [get\_ports reset]//复位按钮

set\_property PACKAGE\_PIN K3 [get\_ports led]

set\_property IOSTANDARD LVCMOS33 [get\_ports led]//溢出led

set\_property PACKAGE\_PIN D5 [get\_ports {seg[7]}]

set\_property PACKAGE\_PIN B4 [get\_ports {seg[6]}]

set\_property PACKAGE\_PIN A4 [get\_ports {seg[5]}]

set\_property PACKAGE\_PIN A3 [get\_ports {seg[4]}]

set\_property PACKAGE\_PIN B1 [get\_ports {seg[3]}]

set\_property PACKAGE\_PIN A1 [get\_ports {seg[2]}]

set\_property PACKAGE\_PIN B3 [get\_ports {seg[1]}]

set\_property PACKAGE\_PIN B2 [get\_ports {seg[0]}]

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

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

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

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

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

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

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

set\_property IOSTANDARD LVCMOS33 [get\_ports {seg[0]}]//数码管段绑定

set\_property PACKAGE\_PIN C1 [get\_ports {sel[3]}]

set\_property PACKAGE\_PIN H1 [get\_ports {sel[2]}]

set\_property PACKAGE\_PIN G2 [get\_ports {sel[1]}]

set\_property PACKAGE\_PIN C2 [get\_ports {sel[0]}]

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

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

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

set\_property IOSTANDARD LVCMOS33 [get\_ports {sel[0]}]//数码管位绑定

3 上板测试

(1)初始上电状态，及reset清零后状态：计数器均为0，LED不亮。



(2)按动一次按钮S0，计数器变为1。



(3)持续按动S0，计数器持续增加，直到59。



十进制11

十六进制B



十进制42

十六进制2A



十进制59

十六进制3B

(4)计数器达到59，再按动，计数器清零，LED亮



(5)继续按动S0，重新计数。



(6)按动reset，计数器清零，LED不亮。



4 总结

在这次实验中，我尝试了使用实验板。实验板的使用和实验箱的使用基本相似，但也有一些不同的地方。例如，在实验板中，信号往往是高电平有效，而在实验箱中很多情况是低电平有效。

本次实验加深了我对硬件实验板结构的认识，也进一步熟悉了Verilog语法和分频器的意义。