EDA实验七

1181002038 张璐矾

实验七

1. 实验目的：
   1. 学会数控分频器的设计与使用
   2. 熟练使用原理图，文本输入等设计方法设计数字系统
   3. 进一步巩固系统和开发软件的使用
2. 实验原理
   1. 设计一个8路跑马灯系统，8个LED等轮流被点亮，通识通过数控分频器控制LED轮流被点亮的频率
   2. 设置3-8译码器的使能按键，当G1,G1A,G2B分别为‘1’‘0’，‘0’时实现跑马灯的效果。3-8译码器可以调用宏模块74LS138或者Verilog文本输入
3. 实验内容
   1. 实验设计
      1. 系统包含5个模块：
         1. 顶层模块：pmd
         2. 分频器模块：divi\_250hz
            1. 输入时钟clk,输出时钟250Hz。实验台的时钟为50MHz,需要将为250Hz的时钟。计算公式:
         3. 数控分频器模块：divclk
            1. 输入时钟clk为250hz,data\_in为预置值，根据预置值来改变计数的多少，从而实现不同的分频比，计算公式：
         4. 3位二进制计数器模块：cnt3
            1. 由于3-8译码器需要8个输出，因此计数器需要从0开始记到7，直接往上加，不用考虑超出的情况。这是因为3位二进制数最大就是7，再大变成0。
         5. 3-8译码器模块：ymq，之前的实验实现过。LED等是0亮1不亮，有两种跑马灯效果：逐渐灭和逐渐亮，并且逐渐灭的视觉效果比逐渐亮的闪烁频率更快，其实速度是一样的。
   2. 代码
      1. 顶层模块：pmd：

module pmd(clk,data\_in,l1,l2,l3,l4,l5,l6,l7,l8);

input clk;

input [3:0]data\_in;

output l1,l2,l3,l4,l5,l6,l7,l8;

wire divi\_1hz,div\_clk;

wire [2:0]q;

divi\_250hz u1(

.clk(clk),

.divi\_1hz(divi\_1hz));

divclk u2(

.data\_in(data\_in),

.clk\_in(divi\_1hz),

.div\_clk(div\_clk));

cnt3 u3(

.clk(div\_clk),

.q(q));

ymq u4(

.in(q),

.l1(l1),

.l2(l2),

.l3(l3),

.l4(l4),

.l5(l5),

.l6(l6),

.l7(l7),

.l8(l8));

endmodule

* + 1. 分频器模块：divi\_250hz：

module divi\_250hz(clk,divi\_1hz);

input clk;

output reg divi\_1hz;

reg[24:0] count\_reg1;

always@(posedge clk)

if(count\_reg1==99999)

begin

divi\_1hz<=~divi\_1hz;

count\_reg1<=0;

end

else

begin

count\_reg1<=count\_reg1+1;

end

endmodule

* + 1. 数控分频器模块：divclk：

module divclk(data\_in,clk\_in,div\_clk);

input clk\_in;

input [3:0]data\_in;

output reg div\_clk;

reg [3:0] count\_c;

always@(posedge clk\_in)

begin

if(count\_c==(data\_in/2)-1)

begin

div\_clk<=~div\_clk;

count\_c<=0;

end

else

count\_c<=count\_c+1;

end

endmodule

* + 1. 3位二进制模块cnt3：

module cnt3(clk,q);

input clk;

output reg[2:0]q;

always@(posedge clk)

begin

q=q+1;

end

endmodule

* + 1. 3-8译码器模块：

module ymq(in,l1,l2,l3,l4,l5,l6,l7,l8);

input [2:0]in;

output l1,l2,l3,l4,l5,l6,l7,l8;

reg [7:0]out;

assign l1=out[0];

assign l2=out[1];

assign l3=out[2];

assign l4=out[3];

assign l5=out[4];

assign l6=out[5];

assign l7=out[6];

assign l8=out[7];

always@(\*)

case(in)

//3'b000:out=8'b00000001;//逐渐灭

//3'b001:out=8'b00000010;

//3'b010:out=8'b00000100;

//3'b011:out=8'b00001000;

//3'b100:out=8'b00010000;

//3'b101:out=8'b00100000;

//3'b110:out=8'b01000000;

//3'b111:out=8'b10000000;

3'b000:out=8'b11111110;//逐渐亮

3'b001:out=8'b11111101;

3'b010:out=8'b11111011;

3'b011:out=8'b11110111;

3'b100:out=8'b11101111;

3'b101:out=8'b11011111;

3'b110:out=8'b10111111;

3'b111:out=8'b01111111;

//default:out=8'b00000000;//全亮

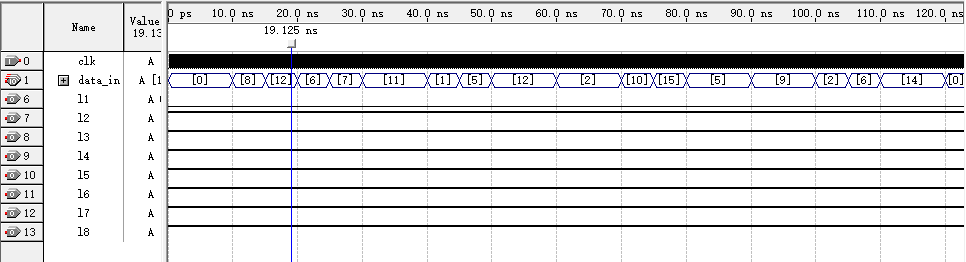
default:out=8'b11111111;//全灭

endcase

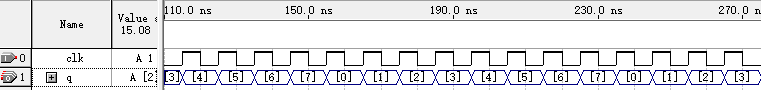
endmodule

* 1. 仿真波形
     1. Pmd

因为分出来的时间对于ln的信号来说太短了，所以一直是高电平。



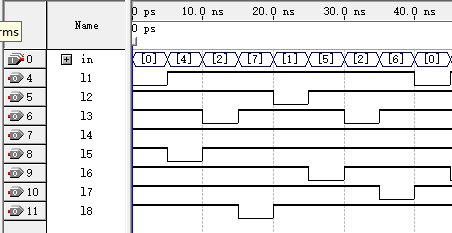
* + 1. cnt3



* + 1. ymq

仅l5=0

仅l6=0

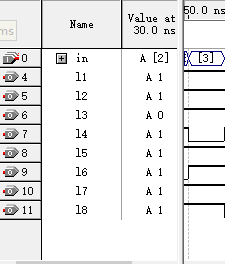


仅l7=0

仅l2=0

仅l3=0

仅l1=0



仅l4=0

* 1. RTL
     1. Pmd



* + 1. Divi\_250hz



* + 1. Divi\_clk



* + 1. Cnt3



* + 1. Ymq



* 1. 门电路
     1. Pmd



* + 1. Divi\_250hz



* + 1. Divi\_clk



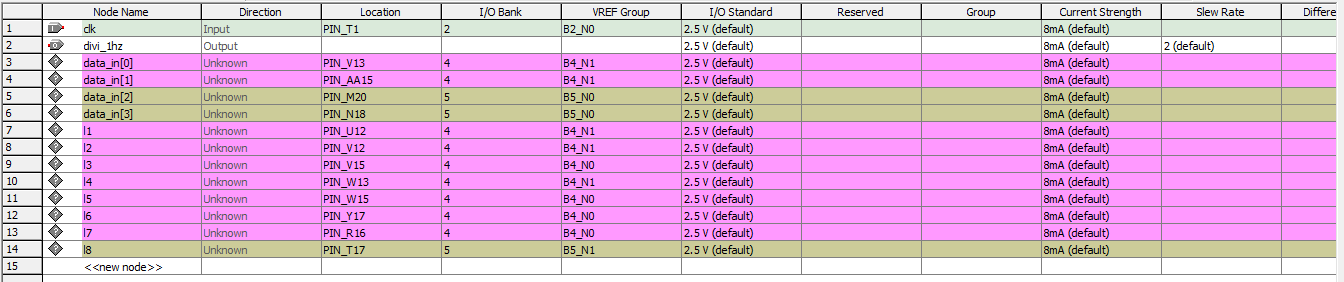
* + 1. Cnt3

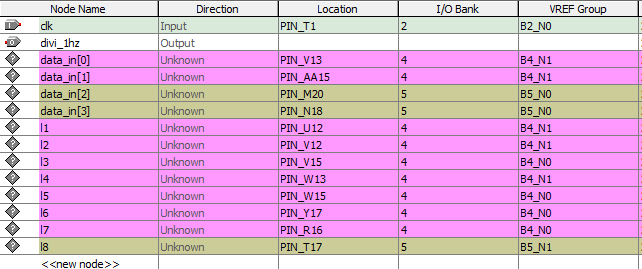


* + 1. Ymq



* 1. 引脚绑定





* 1. 实物图

