

## 数码管显示实验

7 段数码管（不算小数点）的原理也很简单，它无非是由 7 个发光二极管组成。这 7 个发光二极管有一个公共端，必须接 GND（共阴极数码管）或者接 VCC（共阳极数码管）。对 7 个二极管的另一端进行控制，相应的就能控制他们的亮暗。不同的亮暗组合就产生了数字 0-9 的显示效果。若希望数码管显示某个数字，只要给数码管的 7 个段选接口送相应的译码信号即可。

如图 5.6 所示，两个共阴数码管 SM4205 的段选和位选与 CPLD 的 I/O 连接。

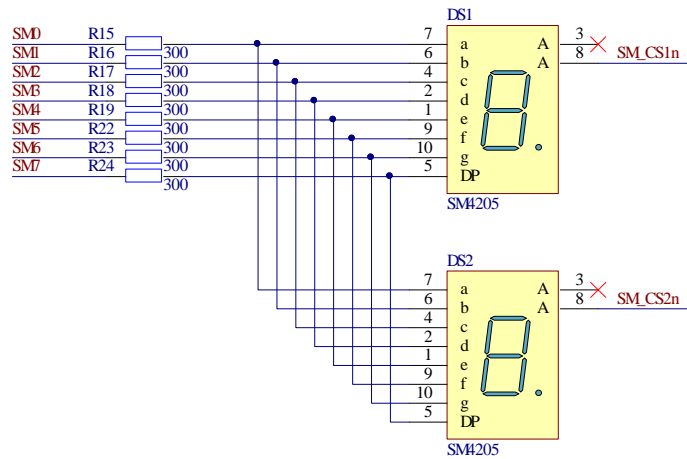


图 5.6 数码管接口

带小数点的共阴极数码管对应的段选如图 5.7 所示。

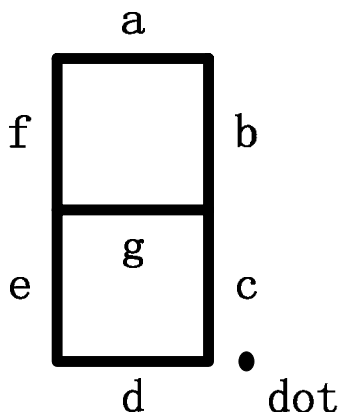


图 5.7 数码管段选

如表 5.6 所示，对 SM0-SM7 进行编码得到相应的显示数字或字符（不带小数点显示）。

表 5.6 数码管编码表

数字/字符	0	1	2	3	4	5	6	7
编码(16 进制)	3f	06	5b	4f	66	6d	7d	07
数字/字符	8	9	A	B	C	D	E	F
编码(16 进制)	7f	6f	77	7c	39	5e	79	71

该实验实现一个两位数码管同时从 0 到 F 循环递增的功能。其接口定义如表 5.7 所示。

表 5.7 数码管实验接口定义

信号名称	方向	描述
clk	input	时钟信号，50MHz
rst_n	input	复位信号，低电平有效
sm_cs1_n	output	数码管 1 片选信号（即共阴端），低电平有效
sm_cs2_n	output	数码管 2 片选信号（即共阴端），低电平有效
sm_db[6:0]	output	数码管段选信号

```

module led_seg7(
    clk, rst_n,
    sm_cs1_n, sm_cs2_n, sm_db
);

input clk;          // 50MHz
input rst_n;       // 复位信号，低有效

output sm_cs1_n, sm_cs2_n; //数码管片选信号，低有效
output[6:0] sm_db; //7 段数码管（不包括小数点）

reg[24:0] cnt; //计数器，最大可以计数到 2 的 25 次方*20ns=640ms

always @ (posedge clk or negedge rst_n)
    if(!rst_n) cnt <= 25'd0;
    else cnt <= cnt+1'b1; //循环计数

reg[3:0] num; //显示数值

always @ (posedge clk or negedge rst_n)
    if(!rst_n) num <= 4'd0;
    
```

```

else if(cnt == 24'hffffff) num <= num+1'b1; //每 (大约) 640ms 增一

//-----
--
/* 共阴极 : 不带小数点
    ;0, 1, 2, 3, 4, 5, 6, 7,
db    3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h
    ;8, 9, a, b, c, d, e, f, 灭
db    7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h, 00h*/
parameter seg0    = 7'h3f,
          seg1    = 7'h06,
          seg2    = 7'h5b,
          seg3    = 7'h4f,
          seg4    = 7'h66,
          seg5    = 7'h6d,
          seg6    = 7'h7d,
          seg7    = 7'h07,
          seg8    = 7'h7f,
          seg9    = 7'h6f,
          sega    = 7'h77,
          segb    = 7'h7c,
          segc    = 7'h39,
          segd    = 7'h5e,
          sege    = 7'h79,
          segf    = 7'h71;

reg[6:0] sm_dbr;          //7 段数码管 (不包括小数点)

always @ (num)
    case (num) //NUM 值显示在两个数码管上
        4'h0: sm_dbr <= seg0;

        4'h1: sm_dbr <= seg1;
        4'h2: sm_dbr <= seg2;
        4'h3: sm_dbr <= seg3;
        4'h4: sm_dbr <= seg4;
        4'h5: sm_dbr <= seg5;
        4'h6: sm_dbr <= seg6;
    endcase

```

```
4' h7: sm_dbr <= seg7;
4' h8: sm_dbr <= seg8;
4' h9: sm_dbr <= seg9;
4' ha: sm_dbr <= sega;
4' hb: sm_dbr <= segb;
4' hc: sm_dbr <= segc;
4' hd: sm_dbr <= segd;
4' he: sm_dbr <= sege;
4' hf: sm_dbr <= segf;
default: ;
endcase

assign sm_db = sm_dbr;
assign sm_cs1_n = 1'b0;    //数码管 1 常开
assign sm_cs2_n = 1'b0;    //数码管 2 常开

endmodule
```

设计中用了一个计数器进行定时，每个定时周期显示的数据递增，而相应的译码逻辑触发后，译码输出数据就发生变化，从而改变数码管显示的数值。

欢迎加入 EDN 网站 FPGA/CPLD 助学小组 <http://group.ednchina.com/1375/>

购买 BJ-EPM240 CPLD 学习板 <http://group.ednchina.com/1375/23842.aspx>

购买 SF-EP1C FPGA 开发板 <http://group.ednchina.com/1375/27650.aspx>

北航出版社将于 2010 年 3 月份前后出版《深入浅出玩转 FPGA》一书，欢迎各位网友到时购买，作为本视频和学习板/开发板的参考教材

特权

2009. 11