

数码管静态控制

该例子使用了数码管和时钟电路，具体电路见“3.1.6 七段数码管 (SMG) 显示电路”和“3.1.4 时钟电路(CLK)”

数码管按驱动方式可以分为静态驱动显示和动态驱动显示。静态显示也就是每一位数码管的 7 段笔画、小数点 (可以共称 8 段，称为段码或字码) 和控制端 (W，称为位码) 都由单独的信号控制；动态显示是指不同位的数码管的对应段码连接在一起，控制端由不同信号控制。

我们现在把数码管 SMG4 ~ SMG1 的位码 (W4 ~ W1) 都对应连接在一起，而他们的段码本来硬件上就连在一起，这样我们就可以把他们看着一个静态驱动显示的数码管来看待，只不过静态数码管是由 4 个相同的数码管组成。下面我们就用这个静态数码管通过 CPLD 来做数码管静态显示实验

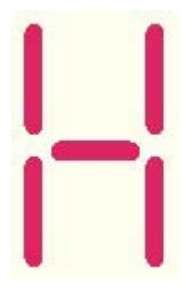
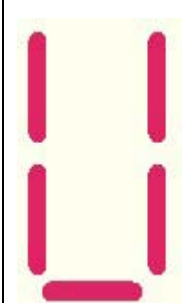
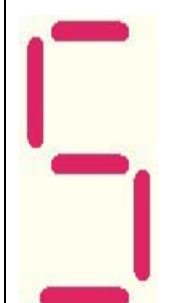
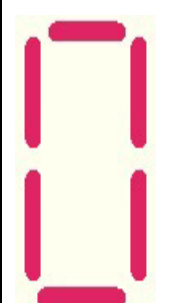

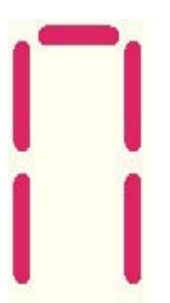
功能描述：通过 CPLD 来控制 1 个静态数码管 (4 个看成一个，如果还有其他数码管，可以同样控制)，让他们显示 H，过 0.76S ($2^{23}/11059200S$) 显示 U，同样分别经过 0.76S 各显示 S、O、O、N，连后熄灭 2.27S，以后重复这个循环。

字符	DP	G	F	E	D	C	B	A
H	0	1	1	1	0	1	1	0
U	0	0	1	1	1	1	1	0
S	0	1	1	0	1	1	0	1
O	0	0	1	1	1	1	1	1
O	0	0	1	1	1	1	1	1
N	0	0	1	1	0	1	1	1

所以 H 的字码为 01110110，U 的字码为 00111110，S 的字码为

01101101 , O 的字码为 00111111 , N 的字码为 00110111

由于七段数码管字形太简单 ,所有上面字符用简单的替代 ,关系如下 :

H	U	S	O	O	N
					

源程序 :(GUIDE 光盘\sample\STATICSMG\STATICSMG.v)

```
//静态数码管显示程序 , STATICSMG.v
//DOWNLOAD FROM WWW.HUSOON.COM
module STATICSMG(SMG, W,CLK);    //定义模块结构
    output [8:1] SMG;            //定义数码管段输出引脚,SMG=dp g f e d c b a
    output [4:1] W;              //定义数码管选择输出引脚 , W=W4 W3 W2 W1
    input CLK;                   //定义输入时钟引脚,CLK=11.0592MHz

    reg [8:1] SMG_REG;           //定义数码管段输出寄存器
    reg [36:0] COUNT;            //定义计数器寄存器

    always @(posedge CLK)        //定义 CLK 信号下降沿触发
    begin
        COUNT=COUNT+1;         //计数器值加 1
    end

    assign W=4'b0000;
    //定义四个数码管的位码都选中 , 相当他们连在一起 , 就可以看着是一个静态数码管

    always @(COUNT[26:23])      //定义显示数据触发事件
    begin
        case (COUNT[26:23])    //选择扫描显示数据

            8'h0: SMG_REG = 8'b01110110; //显示 H
            8'h1: SMG_REG = 8'b00111110; //显示 U
            8'h2: SMG_REG = 8'b01101101; //显示 S
            8'h3: SMG_REG = 8'b00111111; //显示 O
            8'h4: SMG_REG = 8'b00111111; //显示 O
            8'h5: SMG_REG = 8'b00110111; //显示 N
            default:SMG_REG= 8'b00000000; //关掉所有数码管
        endcase
    end
endmodule
```

```

endcase

end

assign SMG=SMG_REG;           //输出数码管译码结果
endmodule

```

操作：在 QUARTUS 中建立工程，并用上面的语句建立 verilog-HDL 文件，保存、编译，连后选定芯片 EPM7128SLC84-15, 并按下表指定管脚:

	To	Location	General Function
1	SMG[1]	PIN_6	I/O
2	SMG[2]	PIN_79	I/O
3	SMG[3]	PIN_81	I/O
4	SMG[4]	PIN_8	I/O
5	SMG[5]	PIN_9	I/O
6	SMG[6]	PIN_4	I/O
7	SMG[7]	PIN_80	I/O
8	SMG[8]	PIN_5	I/O
9	W[1]	PIN_76	I/O
10	W[2]	PIN_77	I/O
11	W[3]	PIN_10	I/O
12	W[4]	PIN_11	I/O

再编译、仿真、下载，并把排针 JP1 对应脚用跳冒插上，并把 J2 上选择 11.0592MHz 的频率，你将看到实验结果

注意：

1. 这里两个“0”(指 HUS00N 中的“0”)是连在一起的，所以可能感觉不到出现两个“0”，但你从时间上可以感觉出，“0”出现的时间是其他字符的两倍

2. 为了便于数组命名,我们把字码定义为 $SMG[8..1]$, 他们对应 dp g f e d c b a3.

3. 这里看上去字符是动的,感觉怎么是静态控制呢?但这和静态显示和动态显示的定义不是一个概念。静态显示也就是每一位数码管的 7 段笔画、小数点(可以共称 8 段,称为段码或字码)和控制端(W,称为位码)都由单独的信号控制;动态显示是指不同位的数码管的对应段码连接再一起,控制端由不同信号控制。