

第五集

数码管外设讲解

§ 1.8 数码管外设

1.8.1 功能概述

说完了按键外设,接下来我们再来看下数码管外设,数码管是一种价格便宜、使用简单的半导体发光器件,在电器特别是家电领域应用极为广泛,我们可以通过对其不同的管脚输入相对的电流,变可以使其发亮,从而能够显示时间、日期和温度等。数码管可分为七段数码管和八段数码管,区别在于八段数码管比七段数码管多一个用于显示小数点的发光二极管。所谓八段数码管其实就是将八个发光二极管, a、b、c、d、e、f、g、dp 八段按一定的方式排列起来,利用不同的组合,来显示不同的阿拉伯数字,如图 1.37 所示。

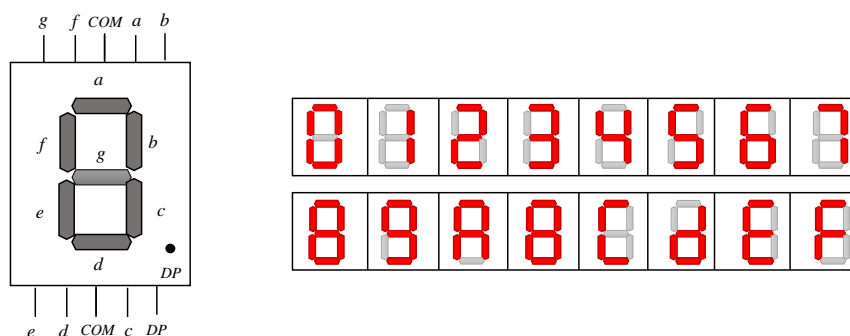


图 1.37 七段数字显示器及发光段组合图 (a) 显示器 (b) 段组合图

数码管按各发光二极管电极的连接方式分为共阳数码管和共阴数码管两种,如图1.38所示。

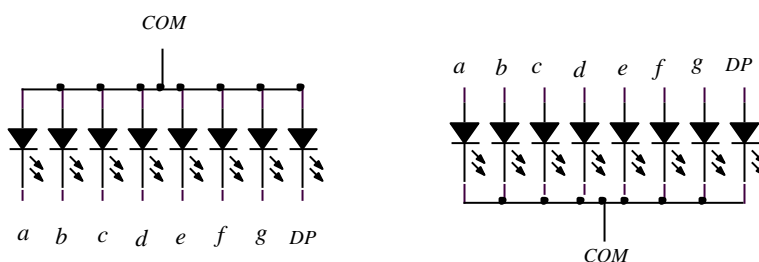


图 1.38 半导体数字显示器的内部接法 (a) 共阳极接法 (b) 共阴极接法

从图中我们可以看出,将八段数码管中的每个二极管的阴极并联在一起,组成公共阴极端。将八段数码管中的每个二极管的阳极并联在一起,组成公共阳极端。如果我们将共阴极管脚接地,那么此时不管哪个管脚输入高电平,对应发光二极管就被点亮。如果我们将共阳极接高电平,那么此时不管哪个管脚输入低电平,对应发光二极管就被点亮。我们的A4开发板采用的是共阴极连接方式,原理图如图 1.39 所示。

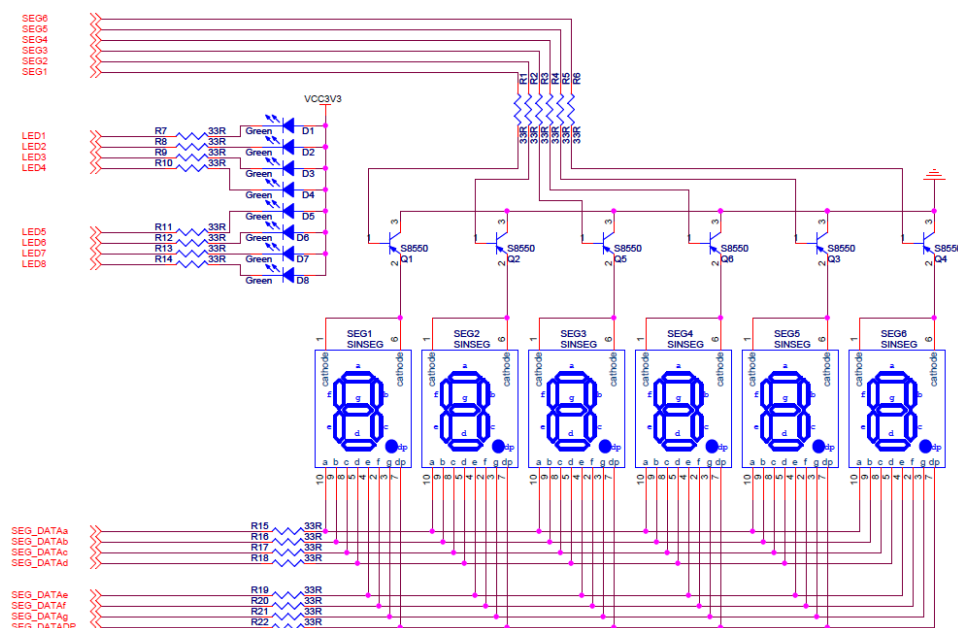


图 1.39 A4 开发板上的数码管原理图

从该图中可以看出，我们将6个数码管的a~g及小数点dp管脚并联在一起连接到FPGA中，作为数码管数据输入端，我们将6个数码管的公共端连接到FPGA中，作为数码管片选端，虽然不是名副其实的“片选”，但是还真达到了异曲同工之妙。

简单的介绍完了数码管，下面我们再来看下数码管例程将要实现一个怎样的功能，该例程主要实现的功能是让6个数码管同时显示一个数字。为了方便大家编写代码，我们将共阴极的数码管编码表，总结如表1.4所示。

表 1.4 数码管编码表

数字	编码	数字	编码	数字	编码	数字	编码
0	3FH	1	06H	2	5BH	3	4FH
4	66H	5	6DH	6	7DH	7	07H
8	7FH	9	6FH	A	77H	B	7CH
C	39H	D	5EH	E	79H	F	71H

1.8.2 设计说明

新建工程，命名为“A4_Segled1”，把这个工程放在专门的文件夹下，其他设置参考1.4 Quartus基础章节。新建Verilog源文件，命名为“A4_Segled1.v”，输入设计代码，综合编译后进行引脚分配，本例程的引脚分配如图1.40所示。

X Named: * Edit: [Icons]							
	Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
out	SEG_DATADP	Output	PIN_R16	5	B5_N0	PIN_R16	2.5 V
out	SEG_DATAa	Output	PIN_L16	5	B5_N0	PIN_L16	2.5 V
out	SEG_DATAb	Output	PIN_L15	5	B5_N0	PIN_L15	2.5 V
out	SEG_DATAc	Output	PIN_N16	5	B5_N0	PIN_N16	2.5 V
out	SEG_DATAd	Output	PIN_N15	5	B5_N0	PIN_N15	2.5 V
out	SEG_DATAe	Output	PIN_P16	5	B5_N0	PIN_P16	2.5 V
out	SEG_DATAf	Output	PIN_P15	5	B5_N0	PIN_P15	2.5 V
out	SEG_DATAg	Output	PIN_T15	4	B4_N0	PIN_T15	2.5 V
out	SEG_EN1	Output	PIN_C15	6	B6_N0	PIN_C15	2.5 V
out	SEG_EN2	Output	PIN_D14	7	B7_N0	PIN_D14	2.5 V
out	SEG_EN3	Output	PIN_C14	7	B7_N0	PIN_C14	2.5 V
out	SEG_EN4	Output	PIN_B14	7	B7_N0	PIN_B14	2.5 V
out	SEG_EN5	Output	PIN_A14	7	B7_N0	PIN_A14	2.5 V
out	SEG_EN6	Output	PIN_A15	7	B7_N0	PIN_A15	2.5 V
<<new node>>							

图 1.40 数码管外设管脚分配完成图

接下来,对工程进行全编译,不仅要让刚刚添加的引脚分配生效,也要生成可以下载到FPGA芯片中的配置文件。

1.8.3 源码解析

本实例代码中设计了 8 个输出端口 SEG_DATAa, SEG_DATAb, SEG_DATAc, SEG_DATAd,SEG_DATAe, SEG_DATAf, SEG_DATAg,SEG_DATADP分别对应的是数码管的 a, b, c, d, e, f, g, DP 八个数据位,即控制数码管显示什么数字。设计了6个输出端口 SEG_EN1, SEG_EN2, SEG_EN3, SEG_EN4, SEG_EN5, SEG_EN6, 分别对应的是六个数码管上的片选端,即控制数码管的亮和灭。我们通过编写相应代码下载到 FPGA 中, FPGA 管脚输出数据直接到数码管上, 便可实现点亮数码管功能。功能框图 1.41 所示。

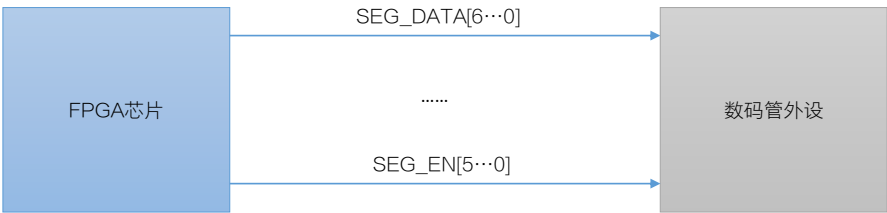


图 1.41 数码管的逻辑设计功能框图

下面我们给出该实例的 Verilog 代码,如代码 1.8 代码 1.7 所示。

代码 1.8 数码管外设的 Verilog 代码

```
1 module A4_Segled1 //模块的开始
2 (
3     //数码管数据引脚
4     SEG_DATAa,SEG_DATAb,SEG_DATAc, SEG_DATAd, SEG_DATAe,
5     SEG_DATAf,SEG_DATAg, SEG_DATADP,
6     //数码管使能引脚
7     SEG_EN1,SEG_EN2,SEG_EN3,SEG_EN4,SEG_EN5, SEG_EN6
8 );
```

```

9
10 //将数码管数据位和数码管使能位声明为输出默认类型 wire
11 output SEG_DATAa,SEG_DATAb,SEG_DATAc,SEG_DATAd,SEG_DATAe,
12        SEG_DATAf,SEG_DATAg,SEG_DATADP;
13 output SEG_EN1,SEG_EN2,SEG_EN3,SEG_EN4,SEG_EN5,SEG_EN6;
14 //ouput [6:0] SEG_DATA; SEG_DATA[0]等价于 SEG_DATAa .....
15        SEG_DATA[7]等价于 SEG_DATAg
16 //ouput [5:0] SEG_EN; SEG_EN[0]等价于 SEG_EN1 ..... SEG_EN[5]等价于 SEG_EN6
17
18 //数码管显示 0~F 对应段选输出
19 parameter SEG_NUM0 = 8'hbf, //数字 0
20            SEG_NUM1 = 8'h86, //数字 1
21            SEG_NUM2 = 8'hdb, //数字 2
22            SEG_NUM3 = 8'hcf, //数字 3
23            SEG_NUM4 = 8'he6, //数字 4
24            SEG_NUM5 = 8'hed, //数字 5
25            SEG_NUM6 = 8'hfd, //数字 6
26            SEG_NUM7 = 8'h87, //数字 7
27            SEG_NUM8 = 8'hff, //数字 8
28            SEG_NUM9 = 8'hcf, //数字 9
29            SEG_NUMA = 8'hf7, //数字 A
30            SEG_NUMB = 8'hfc, //数字 B
31            SEG_NUMC = 8'hb9, //数字 C
32            SEG_NUMD = 8'hde, //数字 D
33            SEG_NUME = 8'hf9, //数字 E
34            SEG_NUMF = 8'hf1; //数字 F
35
36 //给数码管数据位赋值,也可以这样写 SEG_DATA[7:0] = SEG_NUM8;
37 这样写是有前提的,要这样声明 output [7:0] SEG_DATA;
38 assign {SEG_DATADP,SEG_DATAg,SEG_DATAf,SEG_DATAe,
39        SEG_DATAd,SEG_DATAc,SEG_DATAb,SEG_DATAa} = SEG_NUM8;
40 //给数码管使能位赋值也可以这样写 SEG[5:0] = 6'b000000; 同上
41 assign {SEG_EN6,SEG_EN5,SEG_EN4,SEG_EN3,SEG_EN2,SEG_EN1} = 6'b000000;
42 endmodule //模块的结束

```

这里需要我们注意的是:

```
{SEG_EN6,SEG_EN5,SEG_EN4,SEG_EN3,SEG_EN2,SEG_EN1} = 6'b000000;
```

```
{SEG_EN1,SEG_EN2,SEG_EN3,SEG_EN4,SEG_EN5,SEG_EN6} = 6'b000000;
```

以上两句如果都赋值 6'b000000 是没有区别的,如果都赋值 6'b100000;区别就呈现出来了。

第一个句子是把 SEG_EN6 给赋值为 1, 而第二个句子是 SEG_EN1 给赋值为 1。

1.8.4 板级调试

最后,我们将 Quartus 编译生成的 A4_Segled1.sof 文件下载到开发板中,接着我们就能看

到数码管将会被点亮，显示 8，如图 1.42 所示。

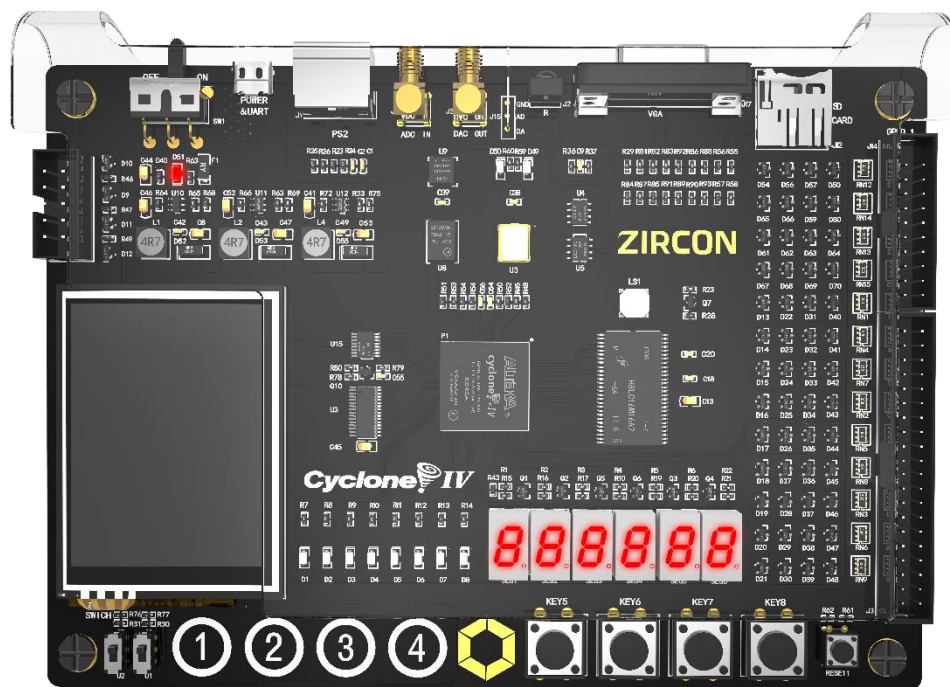


图 1.42 数码管外设板级调试图