



电子电路

综合设计

实例

14个电子电路小系统设计实例

硬件电路
软件VHDL程序
程序注解和说明

集萃

王振红 张斯伟 编著



化学工业出版社

TN702/183

2008



电子电路 综合设计 实例

14个电子电路小系统设计实例
硬件电路
软件VHDL程序
程序注解和说明

集萃

王振红 张斯伟 编著



化学工业出版社

· 北京 ·

本书共列举 14 个电子电路小系统设计实例, 包括数码显示电路, 键盘控制电路, 数模 D/A 转换控制电路, 数控式可逆步进调压直流稳压电源, 低频数字式相位测量仪, 多路数据采集系统, 测量放大器, 功率放大器, 开关型稳压电源, 程控滤波器, 信号发生器等。每个小系统有硬件电路、软件 VHDL 程序, 程序有注释和说明, 并且每个小系统都是经过实验验证过的。

本书可作为高等学校本科和工程专科电子、电气信息类专业电子技术综合实验和课程设计的教材, 参加全国大学生电子设计竞赛赛前训练教材, 也可供成人和职业教育相关专业学生或电气、电子技术工程人员使用。

图书在版编目 (CIP) 数据

电子电路综合设计实例集萃 / 王振红, 张斯伟编著.
北京: 化学工业出版社, 2008. 2
ISBN 978-7-122-01968-4

I. 电… II. ①王…②张… III. 电子电路-电路设计
IV. TN702

中国版本图书馆 CIP 数据核字 (2008) 第 009462 号

责任编辑: 宋 辉

文字编辑: 徐卿华

责任校对: 宋 夏

装帧设计: 尹琳琳

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 刷: 大厂聚鑫印刷有限责任公司

装 订: 三河市延风装订厂

720mm×1000mm 1/16 印张 13 $\frac{3}{4}$ 字数 295 千字

2008 年 3 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 30.00 元

版权所有 违者必究

近年来，全国大学生电子设计竞赛、“挑战杯”全国大学生课外学术科技作品竞赛不断开展，各个高等学校里经常举行各种各样的学生科技活动，大力进行实验教学改革，增设综合性、设计性实验，提高实验教学质量，其目的是培养学生的实践创新意识，提高实践创新能力。

如何通过各类竞赛、学生科技活动提高学生的实践创新能力？首先要使学生对实践感兴趣，在综合性、设计性实验中，学生能做出来就会有兴趣；其次，增加题目数量，使其熟练程度不断提高；之后根据生产和生活实际的需要实现创新作品。即兴趣—提高—创新，不断循环往复，学生的实践创新能力就会不断得到提高。

本书通过电子小系统实践提高学生对实践的兴趣，引导学生由单元电路实践到电子小系统实践的过渡，从而减少学生实践的难度。

本书具有以下几个特点。

① 覆盖了模拟电子技术基础、数字电子技术基础、可编程器件基本知识等内容，是综合电子电路小系统设计。

② 综合电子电路小系统设计，既有硬件电路，又有软件程序。使用的是 VHDL 语言和 MAX + plus II 软件。

③ 列举的每个综合电子电路小系统内容详尽，并且经过实验检验。

本书可作为高等院校相关专业电子技术综合实验和课程设计教材，也可作为参加全国大学生电子设计竞赛前的训练教材。

由于编著者水平有限，书中不妥之处，敬请读者批评指正。

编著者

2008 年 1 月于北方工业大学

实例 1 数码显示电路

1.1 显示原理	1
1.2 驱动 8 位数码管显示电路框图	2
1.3 模块及模块功能	3
1.3.1 时钟脉冲计数器模块	3
1.3.2 3 线-8 线译码器模块	3
1.3.3 八选一数据选择模块	4
1.3.4 七段译码器模块	5
1.3.5 驱动 8 位数码管显示的整体电路	6

实例 2 键盘控制电路

2.1 键盘控制电路的功能	8
2.2 键盘控制电路及工作原理	8
2.2.1 键盘控制电路	8
2.2.2 工作原理	8
2.3 键盘程序设计思想与源程序	9
2.3.1 键盘程序设计思想	9
2.3.2 键盘源程序	9

实例 3 用 8×8 行共阴、列共阳双色点阵发光器件显示汉字

3.1 设计要求	17
3.2 设计步骤	17
3.3 器件及硬件电路	17
3.3.1 器件	17
3.3.2 硬件电路	19
3.4 设计软件的思路及源程序	21
3.4.1 静态显示一个汉字 (王)	21
3.4.2 一屏一屏地显示汉字 (王、正、王)	24
3.4.3 滚动地显示汉字 (王、正、王)	29

实例 4 数模 D/A 转换控制电路

4.1 DAC0832 转换器	37
4.2 数模 D/A 转换电路	

4.3 可编程器件控制的数模 D/A 转换电路	39
4.3.1 电路	39
4.3.2 电路的源程序设计思想	40
4.3.3 电路的源程序	40
实例 5 模数 A/D 转换器 ADC0809 的应用	
5.1 ADC0809 转换器	42
5.2 模数 A/D 转换电路	44
5.3 可编程器件控制的模数 A/D 转换电路	45
5.3.1 ADC0809 的工作时序图	45
5.3.2 可编程器件控制的模数 A/D 转换硬件电路	45
5.3.3 可编程器件控制的模数 A/D 转换软件设计	47
5.3.4 用数码管显示模数 A/D 转换器的输入电压	49
5.3.5 ADC0809 转换模拟输入负电压电路	56
实例 6 数控式可逆步进调压直流稳压电源	
6.1 数控式可逆步进调压直流稳压电源设计要求	58
6.2 数控式可逆步进调压直流稳压电源的原理及硬件电路	58
6.3 数控式可逆步进调压直流稳压电源的软件设计思想及源程序	60
6.3.1 软件设计思想	60
6.3.2 数控式可逆步进调压直流稳压电源源程序	60
实例 7 数控式直流电流源	
7.1 数控式直流电流源设计指标及框图	64
7.2 数控式直流电流源硬件电路图	64
7.3 软件设计思想及源程序	66
实例 8 低频数字式相位测量仪	
8.1 低频数字式相位测量仪设计指标及框图	70
8.2 移相网络	70
8.3 相位测量	72
8.3.1 信号处理电路	72
8.3.2 相位测量电路	72
实例 9 多路数据采集系统	
9.1 设计内容	87
9.2 现场模拟信号产生器	88

9.2.1	自制一正弦波信号发生器	88
9.2.2	频率电压变换器	88
9.2.3	加法器	88
9.2.4	输出直流电压与频率和电阻的关系	89
9.3	8路数据采集器	90
9.4	主控器	94
9.4.1	键盘	95
9.4.2	显示控制模块	104
9.4.3	串并数据转换	106
9.4.4	数据转换	107
9.4.5	数据显示控制	110

实例 10 测量放大器

10.1	测量放大器系统	113
10.2	桥式电路	114
10.3	信号变换放大器	114
10.4	直流电压放大器	115
10.5	程控的直流电压放大器	117
10.5.1	程控的直流电压放大器可编程器件内部模块	117
10.5.2	键盘模块	117
10.5.3	二-十进制数转换十进制数	127
10.5.4	十进制数转换二进制数	128

实例 11 功率放大器

11.1	设计任务	130
11.2	功率放大器	131
11.3	前置放大器	131
11.4	系统测试	132
11.5	自制稳压电源	133
11.6	集成功率放大器	134
11.6.1	集成功率放大器 TDA1521	134
11.6.2	集成功率放大器 TDA2030	136

实例 12 开关型稳压电源

12.1	脉冲宽度调制电路 MIC2194	140
12.2	MC34060 控制的串联型开关稳压电源	141

实例 13 程控滤波器

13.1 设计要求	143
13.2 设计框图	143
13.3 程控放大器	144
13.3.1 程控放大器 0~40dB	144
13.3.2 程控放大器 0~60dB	146
13.4 程控低通滤波器	148
13.5 程控高通滤波器	151
13.6 程控滤波器的 FPGA 控制核心	152
13.6.1 程控滤波器的 FPGA 控制核心内部模块	152
13.6.2 键盘模块和源程序	154
13.6.3 控制模块和源程序	169
13.6.4 低通滤波器控制模块和源程序	170
13.6.5 高通滤波器控制模块和源程序	174
13.6.6 程控放大器 0~40dB 控制模块和源程序	178
13.6.7 分频器模块和源程序	179

实例 14 信号发生器

14.1 信号发生器的功能及其内部接线	181
14.2 信号发生器的可编程器件内部结构	181
14.2.1 键盘模块 KEYBOARD	181
14.2.2 采集键盘数据模块 KEYBOARD_CH	192
14.2.3 键盘数据转换模块 ADD	193
14.2.4 除法计算模块 DIVIDA	194
14.2.5 频率计算和频率步进模块 FRD_DIV	195
14.2.6 正弦波、方波、锯齿波产生模块	197
14.2.7 占空比可调的方波产生模块	201
14.2.8 波形选择模块	204
14.2.9 频率计及其显示模块	206
参考文献	211

实例 1

数码显示电路

1.1 显示原理

八段数码显示管如图 1.1 所示，八段数码管每一段为一发光二极管，共有 a~g 以及小数点 dp 八个发光二极管。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。这样把共阴极管脚接地，此时哪个管脚输入高电平，对应发光二极管就被点亮。

CL5461AS 数码管管脚图如图 1.2 所示，它将 4 个数码显示管的 a~g 及小数点 dp 管脚并联在一起，作为数码管数据输入端。分别引出各个数码管的阴极 A1~A4。

只要在 A1~A4 管脚上轮流加低电平，其频率大于 40Hz，可实现 4 个数码管同时被点亮的视觉效果。在点亮不同数码管的同时输入不同的数据，即可在数码管上同时显示 4 位不同的数字。例如，4 个数码管要显示数字 9876。第 1 个数码管 A1 加低电

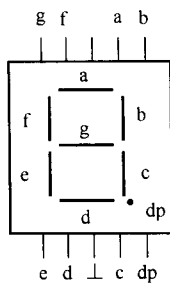


图 1.1 八段数码显示管

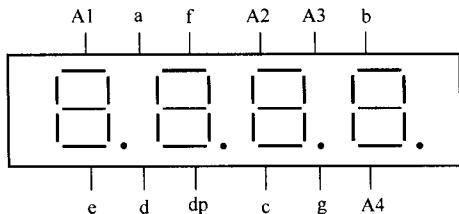


图 1.2 CL5461AS 数码管管脚图

平，其余 A2、A3、A4 加高电平，同时数码管输入和 9 对应的数据；第 2 个数码管 A2 加低电平，其余 A1、A3、A4 加高电平，同时数码管输入和 8 对应的数据；第 3 个数码管 A3 加低电平，其余 A1、A2、A4 加高电平，同时数码管输入和 7 对应的数据；第 4 个数码管 A4 加低电平，其余 A1、A2、A3 加高电平，同时数码管输入和 6 对应的数据。周而复始重复上述过程，4 个数码管就显示数字 9876。

1.2 驱动 8 位数码管显示电路框图

用 CPLD 设计一个驱动 8 位数码管显示电路。8 位数码管管脚图如图 1.3 所示。

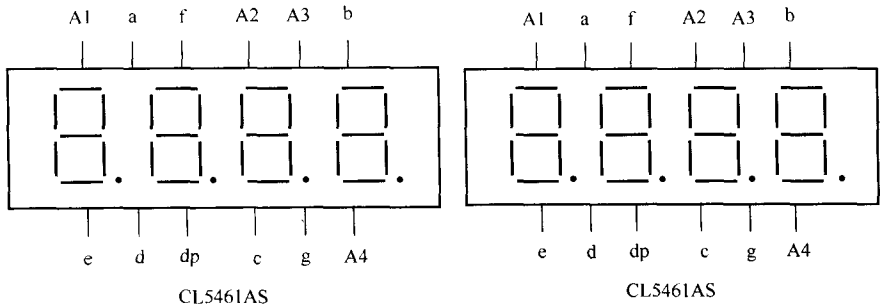


图 1.3 8 位数码管管脚图

用两个 CL5461AS 数码管接成一个 8 位数码管显示，将两个 CL5461AS 数码管的 a~g 及小数点 dp 管脚并联在一起，两个 CL5461AS 数码管的阴极 A1~A4 定义为 Vss0、Vss1、Vss2、Vss3、Vss4、Vss5、Vss6、Vss7。

用 CPLD 设计一个驱动 8 位数码管显示电路的框图如图 1.4 所示。

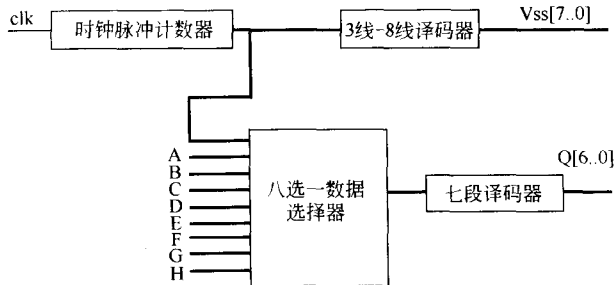


图 1.4 驱动 8 位数码管显示电路框图

时钟脉冲计数器的输出同时作为 3 线-8 线译码器、八选一数据选择器地址码的输入。时钟脉冲计数器的输出经过 3 线-8 线译码器译码其输出信号接到 8 位数码管的阴极 Vss0、Vss1、Vss2、Vss3、Vss4、Vss5、Vss6、Vss7 端。要显示的数据信息 A~H 中哪一个，通过八选一数据选择器的地址码来选择，选择出的数据信息经七段译码器译码接数码管的 a~g 管脚。这样 8 个数码管就可以轮流显示 8 个数字，如果时钟脉冲频率合适，可实现 8 个数码管同时被点亮的视觉效果。

1.3 模块及模块功能

1.3.1 时钟脉冲计数器模块

时钟脉冲计数器模块 CN8 如图 1.5 所示。CN8 模块输入信号是时钟脉冲 clk，其频率大于 40Hz，每遇到一个时钟脉冲 clk 上升沿，内部累加器便加 1，再把累加器所得结果以二进制数的形式输出。要显示 8 位数字，所以用 3 位二进制数作为输出。输出信号为 cout [2..0]。

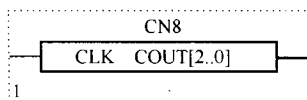


图 1.5 时钟脉冲计数器模块 CN8

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity cn8 is
    port(clk:in std_logic;
         cout:out std_logic_vector(2 downto 0));
end cn8;

architecture rtl of cn8 is
    signal q: std_logic_vector(2 downto 0);
    begin
        process(clk)
        begin
            process(clk)
            begin
                if (clk'levent and clk='1' ) then
                    if (q=7) then
                        q<="000";
                    else
                        q<=q+1;
                    end if;
                end if;
            end process;
            cout<=q;
        end process;
    end rtl;

```

1.3.2 3线-8线译码器模块

3线-8线译码器模块 DECODER3_8 如图 1.6 所示。DECODER3_8 模块

的输入端是 A [2..0]，接收时钟脉冲计数器 CN8 模块的输出信号，经过译码后输出信号 Q [7..0] 分别接 8 个数码管的阴极 Vss7、Vss6、Vss5、Vss4、Vss3、Vss2、Vss1、Vss0，使对应的数码管的阴极为低电平，对应的数码管被点亮。要显示 8 位数字，需要 8 个输出端，所以做成 3 线-8 线译码器。

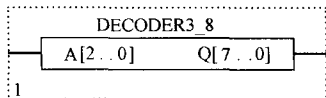


图 1.6 3 线-8 线译码器模块 DECODER3_8

```

library ieee;
use ieee.std_logic_1164.all;
entity decoder3_8 is
    port(a:in std_logic_vector(2 downto 0);
         q:out std_logic_vector(7 downto 0));
end decoder3_8;

architecture rtl of decoder3_8 is
begin
    process(a)
    begin
        case a is
            when "000" => q <= "11111110";
            when "001" => q <= "11111101";
            when "010" => q <= "11111011";
            when "011" => q <= "11110111";
            when "100" => q <= "11101111";
            when "101" => q <= "11011111";
            when "110" => q <= "10111111";
            when others => q <= "01111111";
        end case;
    end process;
end rtl;
    
```

实例

1

1.3.3 八选一数据选择模块

八选一数据选择模块 SEL81 如图 1.7 所示。SEL81 模块输入信号一个是数据选择模块 SEL81 的地址码 SEL [2..0]，另一部分是数据信息 A [3..0] ~ H [3..0]。地址码 SEL [2..0] 来自时钟脉冲计数器 CN8，由地址码 SEL [2..0] 决定输出哪个输入数据。输出信号是 Q [3..0]。

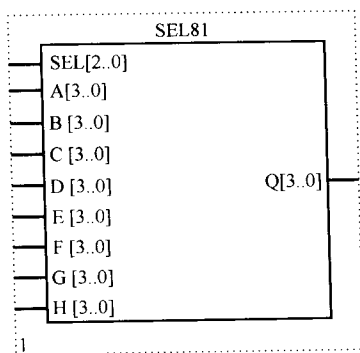


图 1.7 八选一数据选择模块 SEL81

```

library ieee;
use ieee.std_logic_1164.all;
entity sel81 is
    port(sel:in std_logic_vector(2 downto 0);
          a,b,c,d,e,f,g,h:in std_logic_vector(3 downto 0);
          q:out std_logic_vector(3 downto 0));
end sel81;

architecture rtl of sel81 is
begin
    process(a,b,c,d,e,f,g,h,sel)
        variable cout: std_logic_vector(3 downto 0);
    begin
        case (sel) is
            when "000" => cout:=a;
            when "001" => cout:=b;
            when "010" => cout:=c;
            when "011" => cout:=d;
            when "100" => cout:=e;
            when "101" => cout:=f;
            when "110" => cout:=g;
            when others=> cout:=h;
        end case;
        q<=cout;
    end process;
end rtl;

```

1.3.4 七段译码器模块

七段译码器模块 DISP 如图 1.8 所示。该模块将输入的 4 位二进制数转换为

数码显示管所对应的数字。例如，输入为 4 位二进制数 0000 时，使数码显示管显示 0，则要七段译码器输出为 0111111，即 g 段为 0，g 段发光二极管不亮，其他发光二极管被点亮，显示效果为 0。DISP 模块输入信号 D [3..0] 接到八选一数据选择模块的输出信号 Q [3..0]；七段译码器输出信号 Q [6..0] 接数码管的 a~g 管脚。

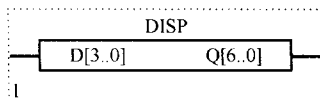


图 1.8 七段译码器模块 DISP

```

library ieee;
use ieee.std_logic_1164.all;
entity disp is
    port(d:in std_logic_vector(3 downto 0);
         q:out std_logic_vector(6 downto 0));
end disp;
architecture rtl of disp is
begin
    process(d)
    begin
        case d is
            when "0000" => q <= "0111111";
            when "0001" => q <= "0000110";
            when "0010" => q <= "1011011";
            when "0011" => q <= "1001111";
            when "0100" => q <= "1100110";
            when "0101" => q <= "1101101";
            when "0110" => q <= "1111101";
            when "0111" => q <= "0100111";
            when "1000" => q <= "1111111";
            when others => q <= "1101111";
        end case;
    end process;
end rtl;
    
```

1.3.5 驱动 8 位数码管显示的整体电路

将各个模块连接起来构成的整体电路如图 1.9 所示，可以实现用 CPLD 设计一个驱动 8 位数码管显示电路的功能。clk 是时钟脉冲输入信号，经过时钟脉冲计数器 CN8 模块，将信号以 3 位二进制数的形式输出，输出信号是 COUT [2..0]。时钟脉冲计数器 CN8 的输出同时作为 3 线-8 线译码器 DECODER3_8

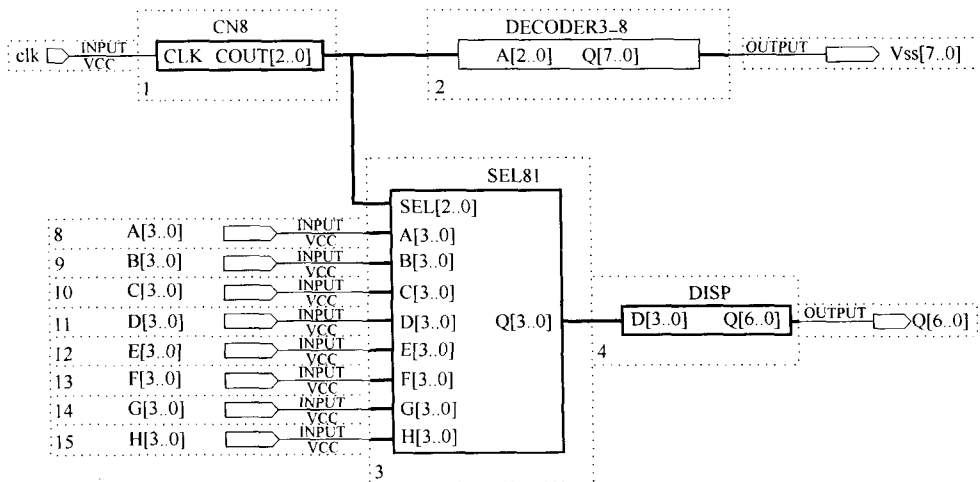


图 1.9 驱动 8 位数码管显示的整体电路

和八选一数据选择器 SEL81 地址码 SEL [2..0] 的输入。时钟脉冲计数器 CN8 的输出经过 3 线-8 线译码器 DECODER3_8 译码其输出信号 Vss [7..0] 接到 8 位数码管的阴极 Vss7、Vss6、Vss5、Vss4、Vss3、Vss2、Vss1、Vss0 端，决定点亮哪位数码管。同时，时钟脉冲计数器 CN8 模块输出的信号也进入数据选择器 SEL81 地址码 SEL [2..0] 的输入，进行输出数据的选择，其输出是 Q [3..0]。八选一数据选择模块 SEL81 模块的输出 Q [3..0] 再经过七段译码器 DISP 模块，将其翻译成可以用数码显示管显示的数据。七段译码器模块 DISP 的输出 Q [6..0] 分别经 300Ω 电阻接数码显示管的 a~g 管脚。八选一数据选择器模块的输入端可根据具体需要进行设计。

实例 2

键盘控制电路

2.1 键盘控制电路的功能

- ① 用两位数码管显示键值。
- ② 输出二十进制代码。
- ③ 防止键抖动。
- ④ 具有功能键（加、减、清零、送数功能键）。

2.2 键盘控制电路及工作原理

2.2.1 键盘控制电路

键盘控制电路如图 2.1 所示。键盘控制电路由键盘、数码管、可编程器件组成。键盘是 4×4 键盘，列线接的电阻是上拉电阻，阻值在 $1k\Omega$ 左右即可，作用为限流。

两位数码管是共阴极的。

可编程器件由键盘程序生成的电路模块表示。CLK_1K 是时钟输入信号，其频率可选 $256\text{Hz} \sim 1\text{kHz}$ ；CLK_40K 也是时钟输入信号，其频率可选 $1 \sim 40\text{kHz}$ ；KEY_LIE [3..0] 为列信号输入，接到键盘的列线上。KEY_HANG [3..0] 是行信号输出，接到键盘的行线上；DISP_DATA [6..0] 是数码管显示译码输出，DISP_SEL [1..0] 是数码管显示扫描输出，接到数码管上；DATA_P [7..0] 是二十进制数输出，START 是二十进制数输出标志。

2.2.2 工作原理

如图 2.1 所示，可编程器件的 KEY_HANG [3..0] 行信号输出端不停地循环输出“1110”、“1101”、“1011”、“0111”。当没有按键按下时，可编程器件的 KEY_LIE [3..0] 列信号输入端检测到的是“1111”。当有按键按下时，例如按下 1 键，此时 KEY_HANG [3..0] 行信号输出为“0111”，即 KEY_HANG [3..0] 的 3 管脚为“0”，可由电路看出，此时输入端 KEY_LIE

[3..0] 检测到的将是“0111”，KEY_LIE [3..0] 的 3 管脚为“0”。可以在编写程序时，将输出信号 KEY_HANG [3..0] 与输入信号 KEY_LIE [3..0] 同时判断，如刚刚举的例子，可以认为当数据“key_hang&key_lie”为“01110111”时，可以译成数据 1。同理可得其他按键的编码。编写程序时，根据不同的键值编码，译成不同的数据。

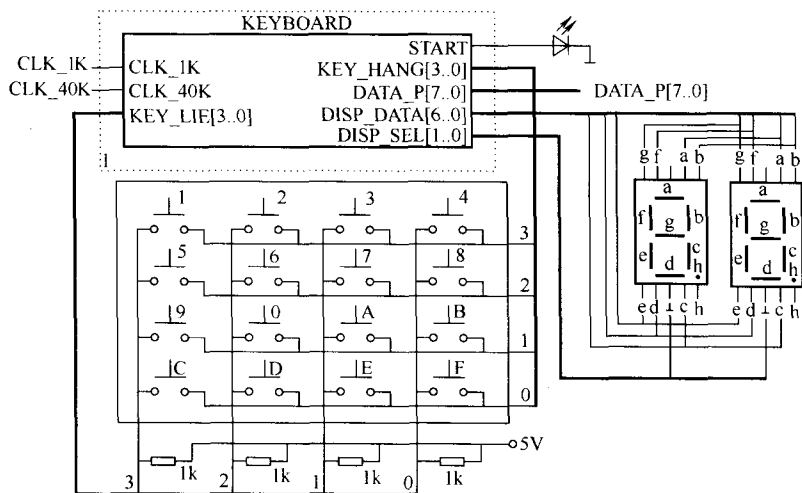


图 2.1 键盘控制电路

2.3 键盘程序设计思想与源程序

2.3.1 键盘程序设计思想

① 循环输出行信号“1110”、“1101”、“1011”、“0111”。检测列信号输入。将行、列信号相并“key_code<=key_hang_tmp&key_lie”。

② 译键值。将行、列相并的信号“key_code”译成“1、2、3、4、5、6、7、8、9”等数字，将“A、B、C、D、E、F”译成功能键，即加 1、加 10、减 1、减 10、清零、送数功能键。

③ 去抖动。在译每一个键值后，为了防止按键抖动，加了一个计数环节，一旦检测列信号后，译码，紧接着进入计数环节，此时键抖动不会进入其他环节，这样可以防止键的抖动。

2.3.2 键盘源程序

键盘源程序产生的电路模块如图 2.2 所示。