单片机控制与应用实验报告

姓名： 陈兴

班级： 八班

学号： 21160839

教学号： 63160827

实验三　步进电机原理及应用

1. 实验目的和要求

初步学习和掌握 MCS-51 的体系结构和汇编语言，了解 Keil 编程环境和程序下载工具的使用方法。

了解步进电机的工作原理，学习用单片机的步进电机控制系统的硬件设计方法，掌握定时器和中断系统的应用，熟悉单片机应用系统的设计与调试方法。

了解数码管输出的原理及编程方式。

2. 实验原理

我们使用的单片机系统的频率是 12M；步进电机转动一周需要 24 步。

本步进电机实验板，使用 FAN8200 作为驱动芯片。CPU 通过如下 4 个引脚与 FAN8200 相连.

本实验使用简单的双四拍工作模式即可，这也是 FAN8200 比较方便的工作方式。只要将 CE1 和 CE2 分别置为高，然后 IN1 和 IN2 按照预定的脉冲输出，即 01->11->10->00->01 这个循环构成一个方向旋转的输出脉冲，将此序列翻转，就是相反方向的输出脉冲。

计算想要设定的计数器初值 s，使用如下方程：

（2 定时器最大位数 － s）× 定时周期 =t

定时周期 = 12/CPU 晶振频率

得到的 s 需要分成高 8 位和低 8 位，分别放入计数器 THx 和 TLx 中（x 为 0 或 1）。

在本题中，快速旋转时转速为 60 转/分，即为 1 转/秒。我们使用的单片机系统的频率是 12M，步进电机转动一周需要 24 步，所以步进电机转动一步需要 1/24 秒，这样的话通过上面的公式我们就可以算出在快速旋转状态下需要被设定的计数器初值。即定时周期 =12/12X10-6=10-6.定时器最大位数为 16，所以公式可写为（216 － s）× 10-6 =1/24，可得 s=23870(十进制数)，转换为十六进制即为 5D3DH，所以需要将定时器的计数初值设为 5D3EH 来控制步进电机快速旋转。

本开发平台有 3 个数码管，使用串行方式连接在一起，具体电路参见实验原理。要想输出一个字形码，就需要从高位到低位依次向移位寄存器输出 8 个比特。移位寄存器的数据线和时钟线分别接到单片机的 P4.5 和 P4.4 管脚，可以使用 MCS-51 里面的位操作指令进行输出。连续输出 3 个字形，24 个 bit 之后，欲显示的字形将稳定地显示在数码管上，程序可以转而执行其他工作。

74HC164 是高速 CMOS 器件。74HC164 是 8 位边沿触发式移位寄存器，串行输入数据，然后并行输出。数据通过两个输入端（A 或 B）之一串行输入；任一输入端可以用作高电平使能端，控制另一输入端的数据输入。两个输入端或者连接在一起，或者把不用的输入端接高电平，一定不要悬空。

时钟 (CLK) 每次由低变高时，数据右移一位，输入到 Q0，Q0 是两个数据输入端（A 和 B）的逻辑与，它将上升时钟沿之前保持一个建立时间的长度。

3. 实验内容

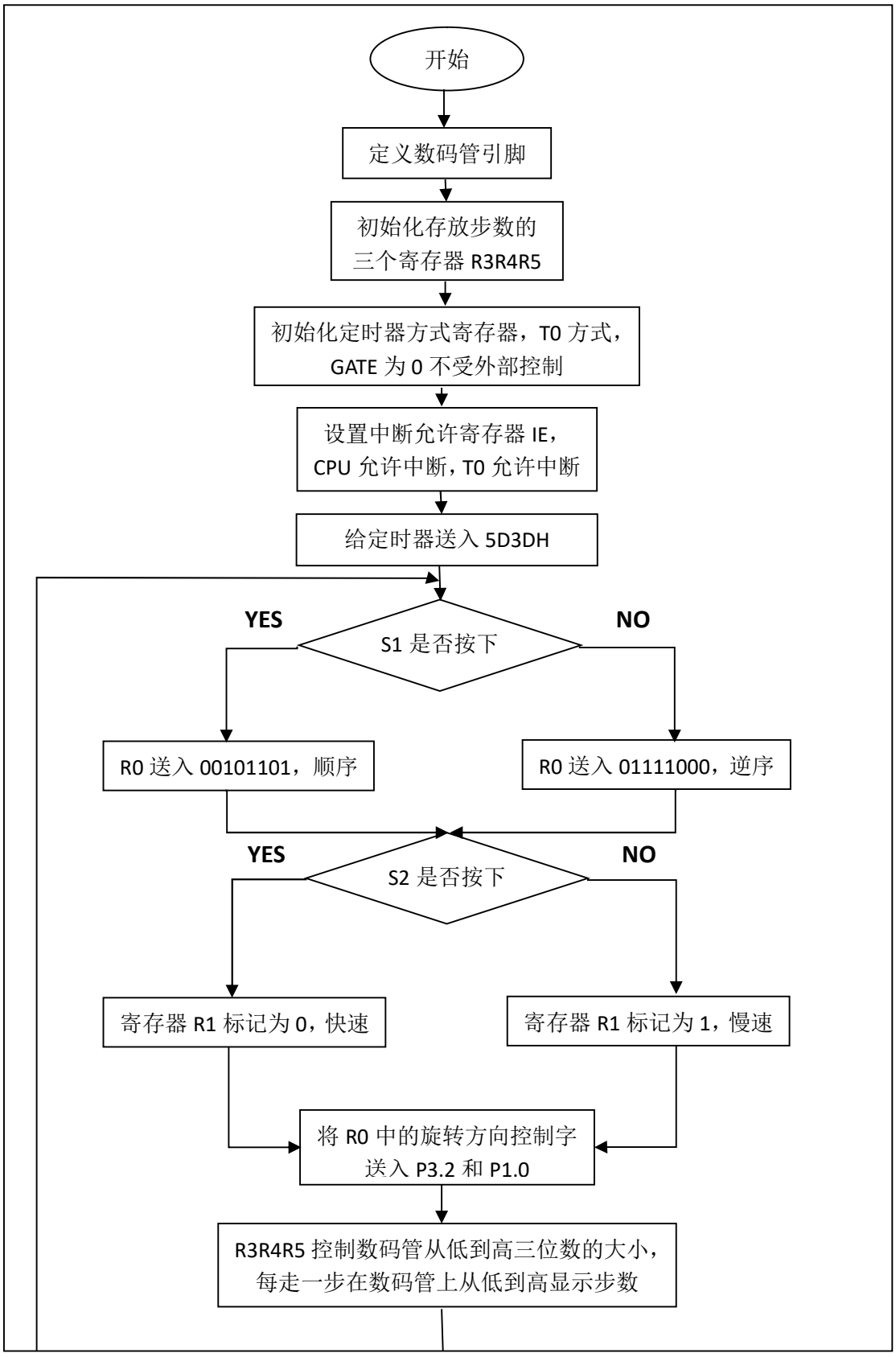
编制 MCS-51 程序使步进电机按照规定的转速和方向进行旋转，并将已转动的步数显示在数码管上。

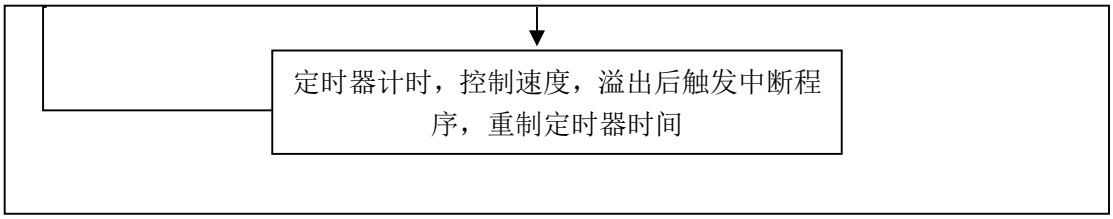
步进电机的转速分为两档，当按下 S1 开关时，进行快速旋转，速度为 60 转/分。当松开开关时，进行慢速旋转，速度为 10 转/分。当按下 S2 开关时，按照顺时针旋转；当松开

时，按照逆时针旋转。

本程序要求使用定时器中断来实现，不准使用程序延时的方式。

4. 实验过程





代码如下：

```
ORG 0000H
    LJMP START
ORG 000BH
    LJMP ZHONGDUAN
ORG 0040H
TAB:
    DB 0C0H,0F9H,0A4H,0B0H,099H,092H,082H,0F8H,080H,090H
START:
    P4 EQU 0C0H
    P4SW EQU 0BBH
    MOV DPTR,#TAB
    MOV P4SW,#030H

    MOV R3,#0H
    MOV R4,#0H
    MOV R5,#0H

    MOV TMOD,#01H
    MOV IE,#10000010B

    MOV TH0,#5DH
    MOV TL0,#3DH
    SETB TR0

START2:
S2SHUN:
    JB P3.7,S2NI
    MOV R0,#00101101B
    LJMP SPEED
S2NI:
    MOV R0,#01111000B
SPEED:
    JB P3.6,LOW0
FAST:
    MOV R1,#0H
    LJMP OUTPUT
LOW0:
```

```
    MOV R1,#1H

OUTPUT:
    MOV R2,#4
    SETB P1.1
    SETB P1.4
LOOP1:
    MOV A,R0
    RLC A
    MOV P3.2,C
    RLC A
    MOV P1.0,C
    MOV R0,A
    LCALL DISPLAY
    LCALL RECORDTIME
    DJNZ R2,LOOP1

    LJMP START2

DISPLAY:
    MOV A,R3
    LCALL DISPLAYPLUS
    MOV A,R4
    LCALL DISPLAYPLUS
    MOV A,R5
    LCALL DISPLAYPLUS
    CJNE R3,#9H,NO1
    MOV R3,#0H
    CJNE R4,#9H,NO2
    MOV R4,#0H
    CJNE R5,#9H,NO3
    MOV R5,#0H
NO1:
    INC R3
    LJMP FINISH
NO2:
    INC R4
    LJMP FINISH
NO3:
    INC R5
    LJMP FINISH
FINISH:
RET
```

```
RECORDTIME:
    CJNE R1,#1H,FAST1
    MOV R6,#6H
LOW1:
     MOV R7,#0H
LOOP3:
    CJNE R7,#1H,LOOP3
     DJNZ R6,LOW1
     LJMP OUT
FAST1:
    MOV R7,#0H
LOOP4:
    CJNE R7,#1H,LOOP4
OUT:
RET

DISPLAYPLUS:
    MOVC A,@A+DPTR
    MOV R6,#8H
LOOP2:
    CLR P4.4
    RLC A
    MOV P4.5,C
    SETB P4.4
    DJNZ R6,LOOP2
RET

ZHONGDUAN:
    MOV TH0,#5DH
    MOV TL0,#3DH
    SETB TR0
    MOV R7,#1H
RETI
END
```

5. 思考题

1) 如何改变步进电机的转向？

答：通过反向 IN1 和 IN2 的输入即可，如将 01->11->10->00->01 改为：00->10->11->01->00。

实验四　LED 点阵显示屏

1. 实验目的和要求

了解 LED 点阵显示的基本原理和实现方法。掌握点阵汉字库的编码和从标准字库中提取汉字编码的方法。

2. 实验原理

高亮度 LED 发光管构成点阵，通过编程控制可以显示中英文字符、图形及视频动态图形。所显示字符的点阵数据可以自行编写（即直接点阵画图），也可从标准字库（如 ASC16、HZ16）中提取。后者需要正确掌握字库的编码方法和字符定位的计算。

实验用的 LED 点阵显示屏为 16*16 点阵。

行和列分别使用两个移位寄存器作为输出。

当移位寄存器输出的第 i 行为 0，第 j 列为 1 时点亮点(i,j)。

为了能够显示出一个点阵字型，需要进行循环扫描，也就是每一次只点亮一行，然后在列上输出该列对应的 16 个点阵值。

输出一行后暂停一段时间，输出下一行。为了达到较好的显示效果，整屏总的扫描时间不高于 40ms。

上述过程中行列可以互换。

实验中使用的移位寄存器是 74HC595，它是一个同时具有串行移位和输出锁存驱动功能的器件。

74HC595 是具有 8 位移位寄存器和一个存储器，三态输出功能。 移位寄存器和存储器是分别的时钟。

数据在 SRCK（移位寄存器时钟输入）的上升沿输入到移位寄存器中，在 RCK（存储器时钟输入）的上升沿输入到存储寄存器中去。

移位寄存器有一个串行移位输入（行 Dx（P00）、列 Dy(P03)），和一个串行输出（QH），和一个异步的低电平复位，存储寄存器有一个并行 8 位的，具备三态的总线输出，当使能（P02 和 P07 为低电平）时，存储寄存器的数据输出到总线。

在控制 74HC595 时，首先将数据放到串行输入的 SI 端，然后在串行时钟 SRCK 上产生一个脉冲，即可输出一个 bit，重复以上步骤 16 次，输出所有列值。

然后给存储器时钟 RCK 一个脉冲，将串行数据锁存起来。将使能端　　输出低电平，驱动到 LED 点阵上。

行的输出每次只移位一次，并重新锁存即可。

其他信息见给定的参考资料。

3. 实验内容

了解 16*16 点阵电路的原理，编写汇编语言程序。

编写一行汉字字符（至少三个字）的显示程序。

能够从左到右（或从右到左）循环显示（要求显示过程中字的大小与屏幕尺寸相适应）。

4. 实验过程

①生成字符的对应表：

“大”

DB 00H,00H,04H,01H,04H,01H,04H,02H,04H,04H,04H,08H,04H,30H,04H,0C0H

DB 0FFH,00H,04H,0C0H,04H,30H,04H,08H,04H,04H,04H,02H,04H,01H,04H,01H

“帅”

DB 00H,00H,00H,00H,0FH,0F0H,08H,08H,08H,10H,0FFH,0FFH,08H,00H,08H,00H

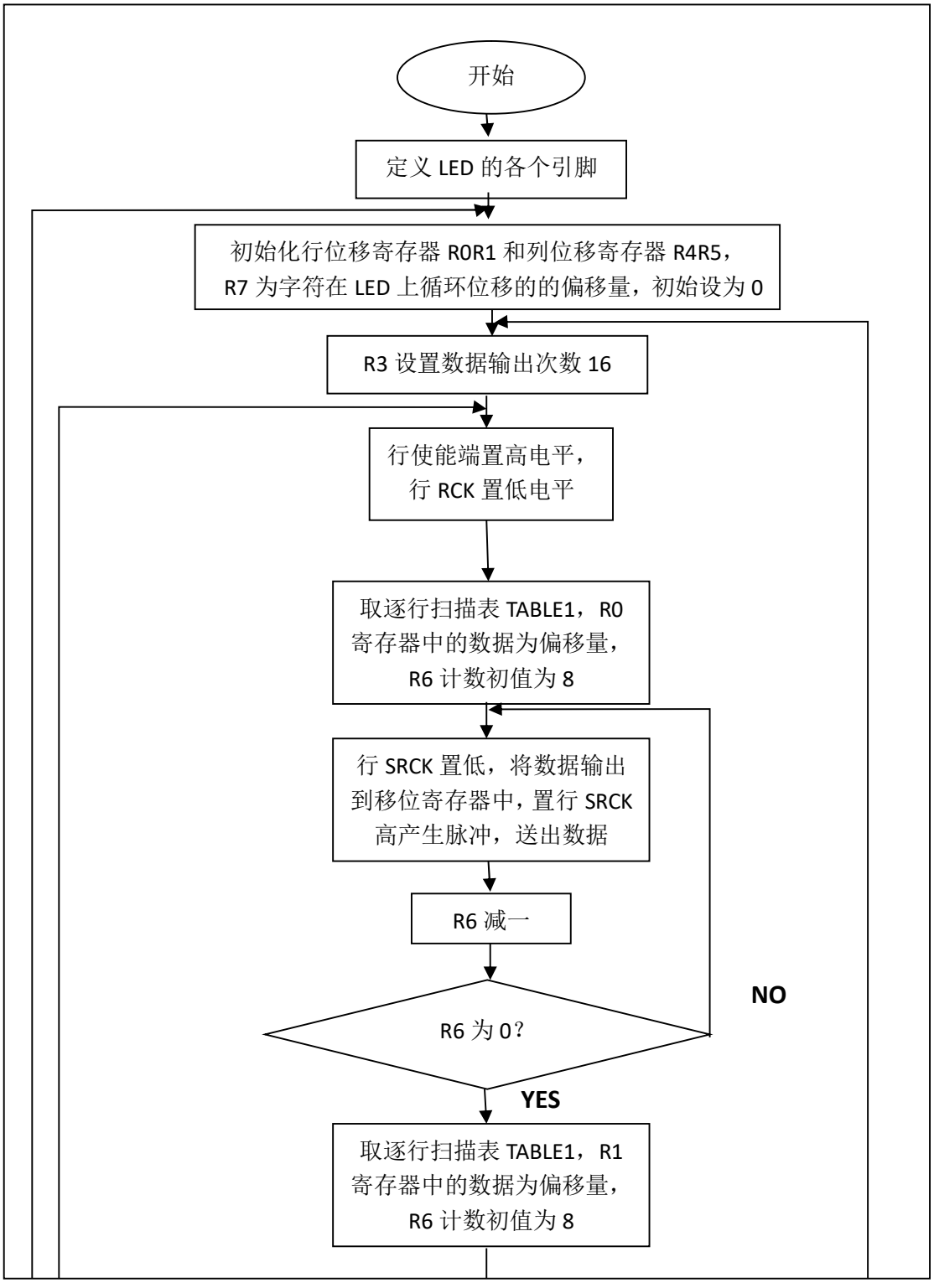
DB 0FH,0F8H,00H,00H,00H,00H,0FFH,0F0H,00H,0CH,00H,02H,3FH,0E1H,00H,00H

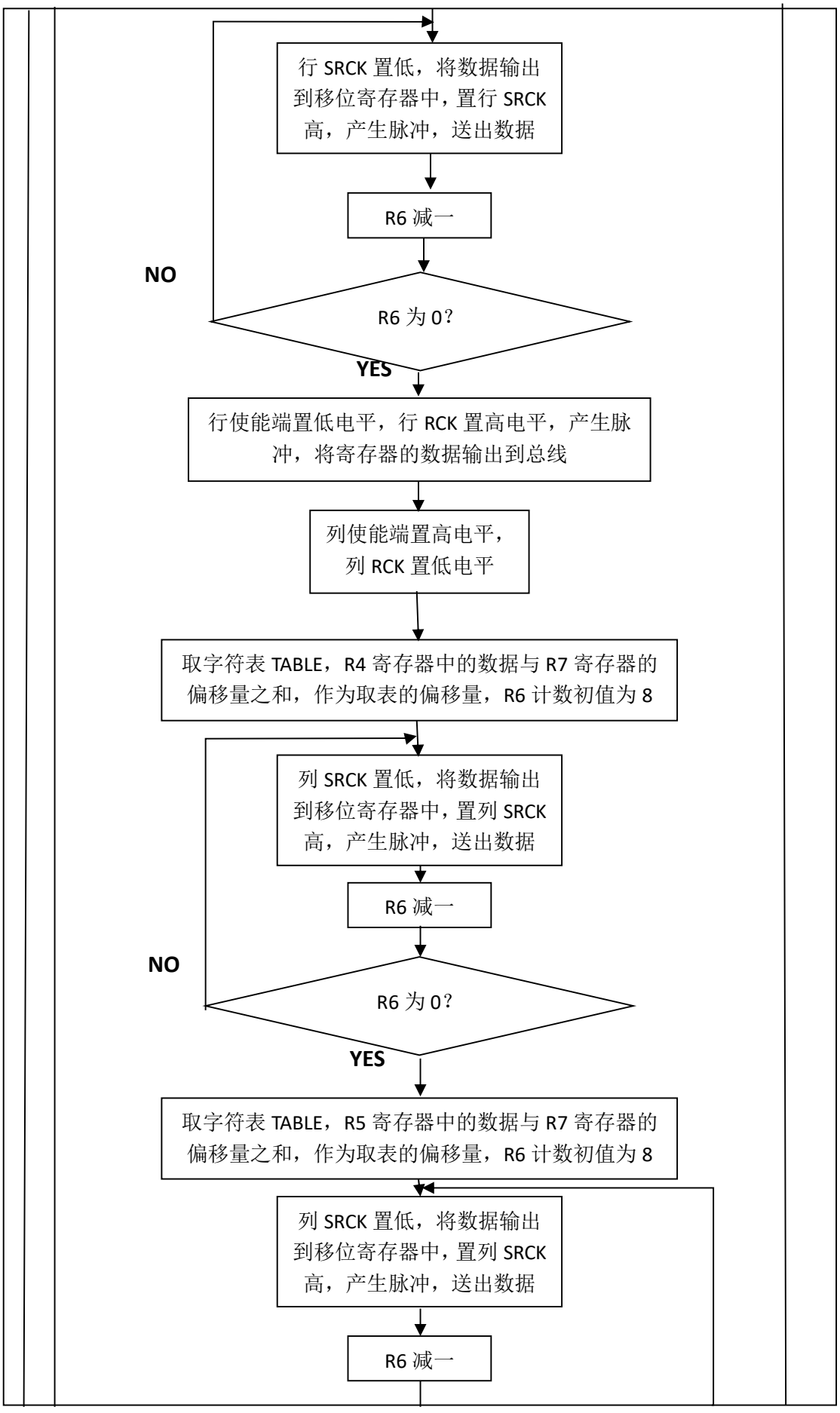
“比”

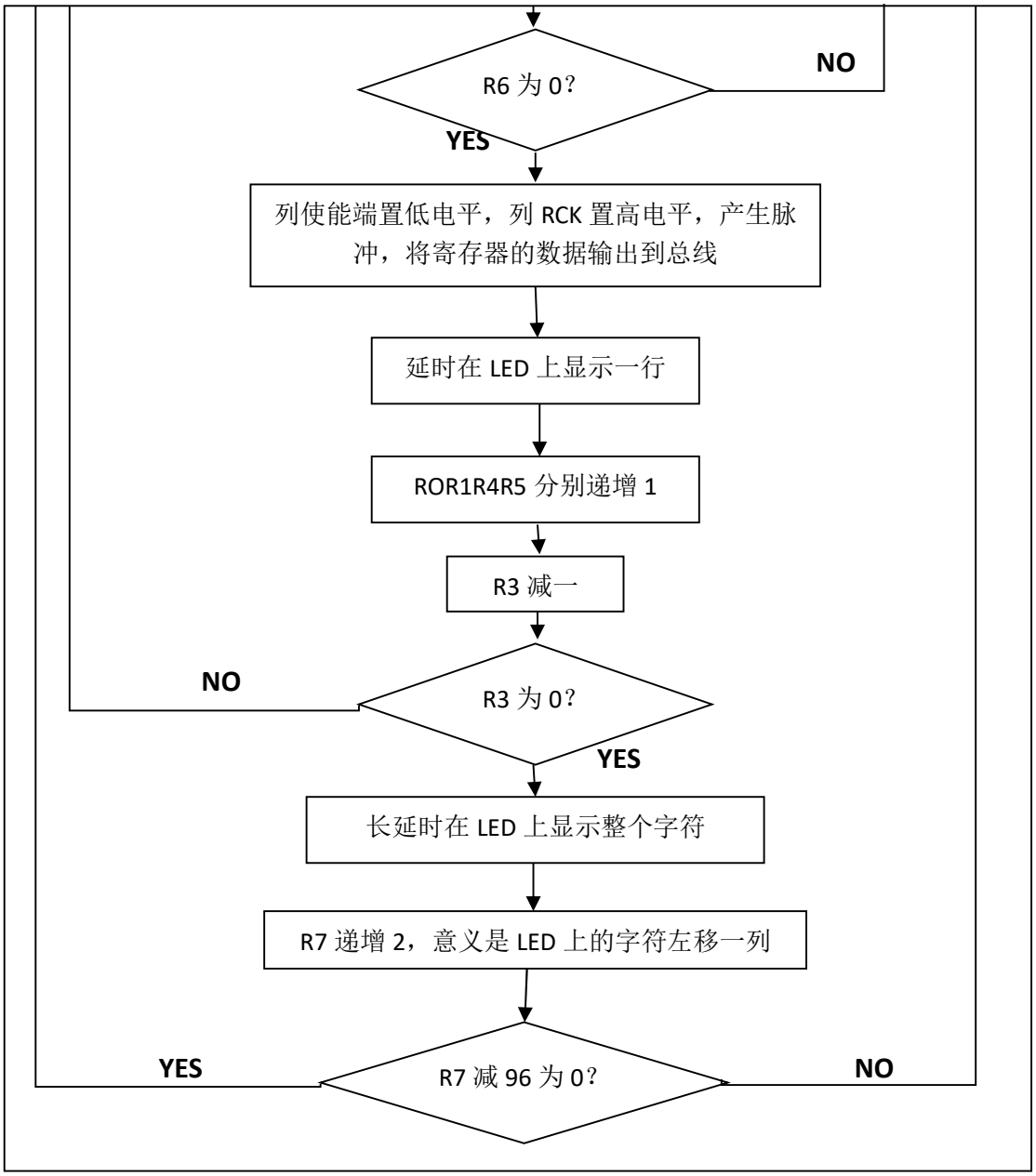
DB 00H,00H,00H,1EH,10H,02H,08H,02H,04H,02H,02H,02H,01H,02H,0FFH,0FCH

DB 00H,00H,02H,08H,02H,08H,02H,04H,02H,04H,7FH,0FEH,00H,00H,00H,00H

②流程图如下：




行 SRCK 置低，将数据输出到移位寄存器中，置行 SRCK 高，产生脉冲，送出数据
R6 减一
NO
R6 为 0？
YES
行使能端置低电平，行 RCK 置高电平，产生脉冲，将寄存器的数据输出到总线
列使能端置高电平，列 RCK 置低电平
取字符表 TABLE，R4 寄存器中的数据与 R7 寄存器的偏移量之和，作为取表的偏移量，R6 计数初值为 8
列 SRCK 置低，将数据输出到移位寄存器中，置列 SRCK 高，产生脉冲，送出数据
R6 减一
NO
R6 为 0？
YES
取字符表 TABLE，R5 寄存器中的数据与 R7 寄存器的偏移量之和，作为取表的偏移量，R6 计数初值为 8
列 SRCK 置低，将数据输出到移位寄存器中，置列 SRCK 高，产生脉冲，送出数据
R6 减一



③代码如下：

```
ORG 000H
LJMP START
ORG 0040H
START:
    D_X EQU P0.0
    D_Y EQU P0.3
    CKX EQU P0.1
    CKY EQU P0.5
    CK_XL EQU P0.2
    CK_YL EQU P0.6
    EN_X EQU P0.7
    EN_Y EQU P0.4
```

```
FORMOVE:
    MOV R7,#0

SM:
    MOV R0,#0
    MOV R1,#1
    MOV R4, #0
    MOV R5, #1

    MOV R3,#16

SM16:
    SETB EN_X
    CLR CK_XL
    MOV DPTR,#TABLE1
    MOV A,R0
    MOVC A,@A+DPTR
    MOV R6,#8
YW1:
    CLR CKX
    RLC A
    MOV D_X,C
    SETB CKX
    DJNZ R6,YW1

    MOV A,R1
    MOVC A,@A+DPTR
    MOV R6,#8
YW0:
    CLR CKX
    RLC A
    MOV D_X,C
    SETB CKX
    DJNZ R6,YW0

    SETB CK_XL
    CLR EN_X
    SETB EN_Y
    CLR    CK_YL
    MOV DPTR,#TABLE

    MOV A,R4
    ADD A,R7
```

```
    MOVC A,@A+DPTR
    MOV R6,#8
YW3:
    CLR CKY
    RLC A
    CPL C
    MOV D_Y,C
    SETB CKY
    DJNZ R6,YW3

    MOV A,R5
    ADD A,R7
    MOVC A,@A+DPTR
    MOV R6,#8
YW2:
    CLR CKY
    RLC A
    CPL C
    MOV D_Y,C
    SETB CKY
    DJNZ R6,YW2

    SETB CK_YL
    CLR EN_Y
    LCALL DELAY

    INC R0
    INC R0
    INC R1
    INC R1
    INC R4
    INC R4
    INC R5
    INC R5

    DJNZ R3,SM16
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    INC R7
    INC R7
```

```
        MOV A,R7
        SUBB A,#96
        JZ    FORMOVE
        LJMP SM

DELAY:
        MOV R6,#0FFH
LOOPD1:
        MOV R2,#0FH
LOOPD2:
        DJNZ R2,LOOPD2
        DJNZ R6,LOOPD1
RET
TABLE:
DB 00H,00H,04H,01H,04H,01H,04H,02H,04H,04H,04H,08H,04H,30H,04H,0C0H
DB 0FFH,00H,04H,0C0H,04H,30H,04H,08H,04H,04H,04H,02H,04H,01H,04H,01H"

DB 00H,00H,00H,00H,0FH,0F0H,08H,08H,08H,10H,0FFH,0FFH,08H,00H,08H,00H
DB 0FH,0F8H,00H,00H,00H,00H,0FFH,0F0H,00H,0CH,00H,02H,3FH,0E1H,00H,00H

DB 00H,00H,00H,1EH,10H,02H,08H,02H,04H,02H,02H,02H,01H,02H,0FFH,0FCH
DB 00H,00H,02H,08H,02H,08H,02H,04H,02H,04H,7FH,0FEH,00H,00H,00H,00H

TABLE1:
DB 80H,00H
DB  40H,00H
DB  20H,00H
DB  10H,00H
DB  08H,00H
DB  04H,00H
DB  02H,00H
DB  01H,00H

DB  00H,80H
DB  00H,40H
DB  00H,20H
DB  00H,10H
DB  00H,08H
DB  00H,04H
DB  00H,02H
DB  00H,01H
END
```

6. 思考题

1）如何使用软件调整和控制 LED 点阵的亮度

答：可以通过控制行显示延时调整亮度。延时越短，LED 点阵越亮；延时越长，LED 点阵越暗。

2）如何尽量避免显示过程中的闪烁

答：增加每一屏显示次数。

3）如何将本实验的软硬件推广到多行多列的 LED 显示屏（如 64*1280）

答：可将 64*1280 看做由 4*80 也即 320 块 16*16 显示屏组成的，显示方法与本实验相同。