《单片机系统及应用》作业题解

第一章、微型计算机与单片机基础知识（P19）

1、微型计算机的硬件由哪几部分组成？简述各部分的作用。

CPU（运算器，控制器），存储器，输入输出设备。

CPU：能够理解由二进制代码组成的指令和数据，不断地从内存中读取指令和数据，进行算术运算和逻辑运算，移动数据及控制输入输出设备。

存储器：存放计算机工作时的程序和数据。

输入输出设备：输入设备是将计算机运行需要的程序和数据输入到计算机中的设备；输出设备是将计算机运行的状态、中间结果和最终结果（如果需要）表现出来的设备。

2、为什么计算机能一步步自动执行存放在内存中的程序？

由于计算机中有一个被称为程序计数器（简称 PC）的部件，它存放下一条要执行的指令的地址，计算机执行程序时，只要按照 PC 的值（地址）去取指令就能一步步自动执行存放在内存中的程序。

3、微型计算机的内存储器有哪几种类型？它们各有什么特点？

ROM（Read Only Memory）和 RAM（Random Access Memory）；RAM 可读可写，为易失性的，即掉电后信息丢失。ROM 只能读不能写，为非易失性的，即掉电后信息不丢失。

4、为什么单片机应用的开发者喜欢选用以 Flash 作为程序存储器的单片机？

单片机采用的是 Flash 程序存储器，可以在线编程，其英文缩写为 ISP。不仅使用方便，而且快捷。

5、什么是堆栈？它有什么作用？

在计算机的内存中可以开辟这样的区域，这里的数据按照先进后出或后进先出的原则进行操作的，这个区域就是堆栈。它的作用是在调用子程序或执行中断服务程序时，为保证正常的返回而存储断点和断点现场信息。

6、CPU 与输入/输出设备之间的数据传送方式有几种？哪一种方式 CPU 效率最高？为什么？

CPU 与输入/输出设备之间的数据传送方式有 4 种：无条件传送方式，查询传送方式，中断传送方式和 DMA 传送方式。

DMA 传送方式 CPU 效率最高。前三种数据传送方式都是通过执行程序进行数据传送，一次只传送一个字节或一个字。而 DMA 传送方式是由 DMA 控制器采用周期窃取方式申请总线控制权，且由 DMA 控制器控制数据的传送（即由硬件实现数据传送）。

7、什么是中断？举一个日常生活中发生的类似计算机中断过程的例子。

中断是指 CPU 在执行一个程序时，发生一个事件（来自 CPU 内部部件或外部设备），CPU 中断现行程序的执行，转去处理这个事件，结束后返回原中断的程序继续执行的过程。

例如：政府机关平时都是按部就班地处理日常事务，一旦出现突发事件，必须停止日常事务的处理，并按应急预案处理突发事件。事件处理结束后又回到日常事务的处理。

8、在计算机内部采用什么表示信息？各举一个例子说明在计算机内部无符号整数、有符号整数、字符的具体表示。

采用二进制表示信息。

无符号整数：通常以字节的偶数倍表示，其所有位均为有效数据。如 16 位无符号整数：0000,0000,0110,0100 表示十进制 100。

有符号整数：通常用补码表示（其主要原因时补码运算时符合位可以数字一样采用运算）。如 16 位有符号整数：0000,0000,0110,0100 表示十进制 100。而 1111,1111,1001,1100 表示十进制-100

字符：通常指 ASCII 字符，8 位。在高级语言中，有一种数据类型为字符型变量（分为无符号字符型和有符号字符型），无符号字符型的数据表示范围为 0～255，有符号字符型的数据表示范围为-128～127。

9、分别写出字母 A、数字 1、回车动作的 ASCII 码，并总结出数字 0～9 和 26 个英文字母的 ASCII 记忆规律。

字母 A、数字 1、回车动作的 ASCII 码分别为：41H，31H，0DH。

数字 0～9 的 ASCII 记忆规律：0～9 的 ASCII 值分别为 30H～39H，即字符 0 的 ASCII 值为 30H，其它阿拉伯数字的字符按顺序递增。

小写 26 个英文字母的 ASCII 记忆规律：字符 a 的的 ASCII 值为 61H，后续字母按顺序递增。

大写 26 个英文字母的 ASCII 记忆规律：字符 A 的的 ASCII 值为 41H，后续字母按顺序递增。

第二章、MCS-51 系列单片机硬件结构（P42）

1、画出 80C51 单片机的内部结构图，说明它所含有的内部资源。

由 8 个部件组成，CPU，时钟电路，数据存储器，程序存储器，定时/计数器，并行接口，串行接口和中断系统。时钟电路直接与 CPU 连接，供给其时钟信号；CPU 与其它 6 个部件通过内部总线进行连接。

2、简要说明 80C51 单片机 P3 各引脚的功能。

P3 的第一功能为普通的 8 位并行 I/O 接口；

P3 的第一功能，其各个引脚功能不同，主要作为控制总线使用；

P3.0：RXD

P3.1：TXD

P3.2：/INT0

P3.3：/INT1

P3.4：T0

P3.5：T1

P3.6：/WR

P3.7：/RD

3、说明 PSW 各位的含义，如果 PSW=91H，表示什么意思？

Cy：进位标志，当 Cy=1 时，表示运算结果的最高位向 Cy 产生了进位；

OV：溢出标志，当 OV =1 时，运算结果出现了实质性错误；

AC：辅助进位标志或半进位标志，当 AC=1 时，表示运算时 D3 向 D4 产生了进位；

P：奇偶标志，运算结果“1”的数量为奇数时 P=1，运算结果“1”的数量为偶数时 P=0；

RS1 和 RS0：它们的四种状态分别表示定义片内 RAM 00H～1FH 中对应的四个区域作为工作寄存器区。F0 和 F1：是由用户定义的标志位。

如果 PSW=91H，即 PSW=10010001B，说明 Cy=1，OV=0，AC=0，P=1，RS1RS0=10。

4、简述晶振周期、状态周期、机器周期和指令周期之间的关系。

晶振周期：是单片机的晶体震荡器（简称晶振）所提供的原始时钟周期。

状态周期：即时钟周期，它是 CPU 处理的最小时间单位。在 MCS-51X 系列单片机中，1 个时钟周期由 2 个晶振周期构成。

机器周期：机器周期由 12 个晶振周期构成，它是指令执行时间的度量单位。

指令周期：即指令的执行时间，在 MCS-51X 系列单片机中，指令周期为 1、2、4 个机器周期。

5、80C51 单片机的片内 RAM 和片内 ROM 各起什么作用？它们的地址范围各是多少？

片内 RAM：地址范围为 00H～FFH，作用是存储单片机处理的数据、运算的中间结果和结果，反映单片机各 SFR 的状态等。

片内 ROM：地址范围为 0000H～0FFFH，作用是存储单片机运行的程序和为操作方便制定的各种数据表等。

6、单片机的上电或复位后从哪处开始执行？应用程序为什么一般在该处放一条转移指令？

从 0000H 开始执行。由于 0003H、000BH、0013H、001BH、0023H 是单片机 5 个中断源之中断服务程序的入口地址，0000H～0003H 之间只留了三个字节单元，因此必须在该处放一条转移指令，跳到主程序所在位置去执行。

7、什么情况下需要使用 SP？

SP 是堆栈指针，其内容永远指向栈顶单元。

当调用子程序时，保存返回地址（即断点），当调用中断程序时，保存返回地址（断点）和断点的现场。

9、在最小系统应用模式时，P0 和 P3 可以作为什么使用？在总线扩展模式时，它们又起什么作用？

在最小系统应用模式时，P0 和 P3 均作为 I/O 接口使用；在总线扩展模式时，P0 作为 8 位 DB，低 8 位 AB 使用，P3 的/RD 和/WR 作为 CB 使用。

第四章、MCS-51 系列单片机的指令系统与汇编语言程序设计（P93）

2、在 MCS-51 中，操作数都可以存放在哪里？指令存放在哪里？

操作数可以存放在指令中（立即寻址），也可存放在寄存器中（寄存器寻址），也可存放在数据存储器中（除了立即寻址、寄存器寻址和相对寻址以外的其它寻址）。

注：还有一种操作数在程序存储器中，如常数。

3、说明下列指令中源操作数使用的寻址方式。

（1）MOV R0,#27；立即寻址

（2）MOVC A,@A+PC；变址寻址

（3）MOV @DPTR,A；寄存器寻址

（4）MOVC 70H,A；题目应改为 MOV 70H,A；寄存器寻址

（5）CLR P3.7；位寻址

（6）MOV 70H,A；寄存器寻址

（7）AJMP START；直接寻址

（8）SETB 20H；位寻址

（9）MOVC A,@DPTR+A；题目应改为 MOVC A,@A+DPTR；基址寻址

4、编程实现下列操作：

（1）将 R5 中的内容放入 A 中；　　MOV A，R5

（2）将以 R0 中的内容为地址的片内 RAM 单元中的内容放入 A 中；　　MOV A，@R0

（3）将 A 中内容放入片外 RAM 的 1000H 单元中；

```
MOV DPTR,#1000H
MOVX @DPTR,A
```

（4）将 P1.0 的信号传送给进位位；　　MOV C,P1.0 或 MOV C,90H

（5）（30H）◄——（R1）+（R2）。

首先，我们先看下一这样的操作：

（30H）◄——（R1）+（R2）和 30H ◄——（R1）+（R2）

很明显，它们之间一定存在差异。

30H◄——（R1）+（R2）是：R1 的内容与 R2 的内容求和，存放于 30H 单元中。

因此其程序如下：

```
MOV A，R1
ADD A，R2
MOV 30H，A
```

（30H）◄——（R1）+（R2）是 R1 的内容与 R2 的内容求和，存放于以 30H 单元中的内容为地址的单元中。

故其程序如下：

```
MOV A，R2
ADD A，R1
MOV R0，30H
MOV @R0，A
```

5、已知 R1 中的内容为 30H，30H 中的内容为 60H，40H 中的内容为 0FH，A 中的内容为 EFH，分析下列指令执行后的结果。

（1）MOV A，@R1　；A=60H，其它不变
（2）MOV @R1，A　；（30H）=EFH，其它不变
（3）MOV 40H，A　；（40H）=EFH，其它不变
（4）MOV R1，#0FH；R1=0FH，其它不变
（5）MOV R1，30H ；R1=60H，其它不变

6、编制程序实现将存放在片外 RAM 3000H 单元的 10 个数传送到片内 RAM 30H 开始的 10 个单元中。

```
MOV R7，#10
MOV DPTR，#3000H
MOV R0，#30H
LOOP:MOVX A，@DPTR
MOV @R0,A
INC DPTR
INC R0
DJNZ R7,LOOP
SJMP $
```

7、指出 MCS-51 系列单片机进行加法运算 85H+9DH 后的结果和进位标志位、奇偶标志位、溢出标志位的值。

答：　1000,0101
　　+1001,1101
　=1,0010,0010
Cy=1，OV=1，　P=0，（AC=1）

8、用移位指令实现累加器 A 中的数乘以 8。

方法一：

```
CLR C
RLC A
CLR C
RLC A
CLR C
RLC A
```

方法二：

```
RL A
ANL A，#0FEH
RL A
ANL A，#0FEH
RL A
ANL A，#0FEH
```

9、除法指令运算时除数和被除数需放在哪里？商和余数又在哪里？

除数和被除数放在 B 和 A 中，商和余数放在 A 和 B 中。

10、条件转移指令的转移地址范围是多少？无条件转移指令又如何？

条件转移指令都带 rel，是一个 8 位有符号数，故其转移地址范围为-128～127。

无条件转移指令：

短跳指令 AJMP add11，其转移地址范围为 2KB。该指令占 2 字节，即取指完成时，PC=PC+2，执行时，将 add11 放入 PC 的低 11 位中，PC 的高 5 位不变。

长跳指令 LJMP add16，add16 指的是实际的物理地址，故其转移地址范围为 64KB。执行时，将 add16 直接放入 PC 中。

相对转移指令 SJMP rel，rel 是一个 8 位有符号数，故其转移地址范围为-128～127。

11、什么是伪指令？它与指令系统中的指令有何区别？

它是不能被 CPU 执行的，只是为汇编程序提供必要的信息的指令。

区别：CPU 可执行指令，不能执行伪指令；

伪指令只用于编译汇编程序。

12、阅读下面的程序，指出程序实现的功能。

```
        ORG 0000H
        AJMP START
        ORG 0100H
START：MOV R0,#50
        MOV R1,#60
        MOV A,@R0
        ADD A,@R1
        MOV @R0,A
        INC R0
        INC R1
        MOV A,@R0
        ADDC A,@R1
        MOV @R0,A
        SJMP $
END
```

程序功能：该程序实现 16 位数的加法。片内单元 51H 和 50H 分别存放一个 16 位数据的高 8 位和低 8 位；61H 和 60H 分别存放另一个 16 位数据的高 8 位和低 8 位。相加结果存放在 51H 和 50H 中（51H 存放高 8 位）。

13、散转指令适合在什么程序结构应用？在散转指令中，累加器 A 存放的是什么？转移地址是如何形成的？

散转指令：JMP @A+DPTR

适合于多分支程序的应用。DPTR 存放基地址，累加器 A 存放位移量。A+DPTR 形成下一条要执行的指令地址。

14、保护断点和保护现场的含义是什么？保护断点是怎么实现的？什么时候需要保护现场？如何实现？

这里的保护断点是指子程序调用时的，即执行 LCALL 或 ACALL 指令时由系统自动完成。

保护现场也是只子程序调用时，子程序和主程序都使用相同的寄存器或某存储单元中的数据，而这些

容器中的数据并不是子程序使用的（只是要用到这些容器）。这样需要在子程序的开始进行现场保护，以保证返回主程序时，能够保证主程序的正确运行。

保护断点的实现：LCALL 或 ACALL 指令中有将 PC 的内容（即断点）入栈的微操作。

只有在子程序中使用了主程序使用的寄存器或某存储单元，并且这些容器中的数据并不是子程序使用的。

在子程序的开始时，将这些容器的内容入栈。

15、主程序与子程序之间传递参数有几种方法？

三种方法。①用累加器或寄存器传递参数；②通过指针寄存器传递参数；③使用堆栈传递参数。

16、阅读下面的程序，指出程序实现的功能。

```
        ORG 0000H
        AJMP START
        ORG 0100H
START：MOV R0,#30H
        MOV R7,#10
        MOV R6,#0
LOOP：  MOV A,@R0
        CLR C
        SUBB A,R6；A=A-R6-Cy
        JC NEXT
        MOV A,@R0
        MOV R6,A
NEXT：INC R0
        DJNZ R7,LOOP
        MOV 40H,R6
        SJMP $
END
```

程序功能：从 30H 单元开始存放 11 个数，该程序实现后数减去前数，结果存放于片内单元 40H 中。

17、设在片内 RAM 的 30H 开始的 5 个单元中存放 5 个数，编程将它们转换成对应的 ASCII 码，存放于片内 RAM 的 40H 开始的 10 个单元中.

```
        ORG 0000H
        AJMP START
        ORG 0100H
START：MOV R0,#30H
        MOV R7,#5
        MOV R1,#40H
        MOV DPTR,#TAB
LOOP：  MOV A,@R0
        ANL A,#0F0H
        SWAP A
        MOVC A,@A+DPTR
```

```
        MOV @R1,A
        INC R1
        MOV A,@R0
        ANL A,#0FH
        MOVC A,@A+DPTR
        MOV @R1,A
        INC R1
        INC R0
        DJNZ R7,LOOP
        SJMP $
TAB:    30H,31H,32H,33H,34H,35H,36H,37H,38H,39H,41H,42H,43H,44H,45H,46H
END
```

18、设计实现由 80C51 控制两列发光二极管循环亮灭的应用系统，每列有 8 个发光二极管。要求编程实现一列发光二极管从上到下顺序点亮，另一列发光二极管从下到上顺序点亮。循环往复。

P0 口接 8 个发光二极管，形成一列，P0.0 在上，低电平点亮。

P1 口接 8 个发光二极管，形成一列，P1.0 在上，低电平点亮。

```
        ORG 0000H
        AJMP START
        ORG 0100H
START：MOV R7,#8
        MOV R0,#0FEH
        MOV R1,#7FH
LOOP：MOV A,R0
        MOV P0,A
        RL A
        MOV R0,A
        MOV A,R1
        MOV P1,A
        RR A
        MOV R1,A
        CALL DELAY
        DJNZ R7,LOOP
        SJMP START
DELAY：MOV R6,#50
DELAY1：MOV R5,#50
DELAY2：DJNZ R5,DELAY2
         DJNZ R6,DELAY1
         RET
END
```

第五章、单片机的 C 语言程序设计（P124）

1、MCS-51 单片机汇编语言与 C51 语言各有什么优缺点？

通常，汇编语言编制的程序操作直接、简捷、程序紧凑、执行效率高。但可移植性和可读性较差，在程序规模较大时，开发工作量大。

利用 C51 编制程序，编程效率高，可移植性和可读性好，便于模块化设计。通常占用资源多，执行效率不如汇编语言程序。

2、在 C51 程序中，unsigned char 和 char 各表示什么数据类型？取值范围各是多少？为什么要尽可能使用它们？

unsigned char 是无符号字符型，值域：0～255

char 是有符号字符型，值域：-128～127

节省片内资源占用，提高 CPU 效率。

3、在 C51 程序中，为什么很少使用长整形和浮点型的数据类型？

主要原因是单片机的资源太少，如果可以，尽可能使用其它数据类型，以减少资源占用和提高运算速度。

4、在 C51 程序中有哪几种数据类型是 ANSI C 语言所没有的？它们各有什么用途？

bit,sbit,sfr 和 sfr16 四种数据类型。

Bit：定义位变量的数据类型，在位寻址区。

Sbit：定义特殊功能位变量的数据类型，在可位寻址的 SFR 中。

Sfr：定义 8 位特殊功能寄存器变量的数据类型。

sfr16：定义 16 位特殊功能寄存器变量的数据类型。

5、C51 为什么要定义变量的存储类型？C51 有哪几种存储类型？哪种类型的变量所占空间最小且处理速度最快？

定义变量存储类型的目的是尽可能最大程度地提高单片机应用系统的工作效率。

C51 的存储类型有：data、bdata、idata、pdata、xdata 和 code 共六种。

处理速度最快的存储类型是 data 和 bdata，占空间最小存储类型是 bdata。因此占空间最小且处理速度最快的存储类型是 bdata。

6、C51 存储模式有哪些？含义是什么？

C51 存储模式有 small,compact,large。

small 存储模式也称小模式，小模式定义的变量和参数均安排在片内 RAM 中，为 data 数据存储类型。

compact 存储模式也称紧凑模式，此模式定义的变量和参数均安排在片外 RAM 中，且可以按页访问，为 pdata 数据存储类型。

large 存储模式也称大模式，此模式定义的变量和参数均安排在片外 RAM 中，为 xdata 数据存储类型。

7、如果不定义程序中变量的存储类型，而选择了 SMALL 存储类型，这时程序中所使用变量的存储类型是什么？

程序中所有的变量和参数均为 data 数据存储类型。

8、举例说明 C51 实现对某一位进行操作的方法。

数据类型定义：bit x；定义 x 为位变量，且该变量被分配在位寻址区。定义后可以使用 C51 语句进行相应操作。

数据类型定义：Sbit PP=P1^0；定义 PP 为特殊位变量，该变量被分配在可位寻址的 SFR 中，即定义某个特殊国内位。定义后可以使用 C51 语句进行相应操作。

9、写出下列变量最适合的变量声明：

（1）存放在片内 RAM 中取值范围为 0～200 整数的变量 a0。

（2）存放在片外 RAM 中取值为 0～500 整数的变量 a1。

（3）存放在片内 RAM 的地址 0x30～0x7F 中取值为-100～+100 整数的变量 a3。

（4）特殊功能寄存器 P0。

（5）P1 的最高位。

```
（1）unsigned char idata a0
（2）int xdata a1
（3）char data a3
（4）sfr P0=0x80
（5）sbit P17=P1^7
```

10、举例说明用宏定义定义一个常数的的好处。

①可用标识符替代较长的数据，减少数据或字符串输入的工作量

②用易于理解的标识符替代不易记忆的具体数据，便于理解和维护程序

③易于程序修改和升级，当修改数据时，只需要修改宏定义即可

11、条件编译预处理指令有何用途？

C51 的条件编译指令是为满足可移植性而设置的。条件编译预处理指令可以实现程序的部分语句根据条件进行编译，即条件满足则进行编译，否则不编译。

12、定义一个中断函数，函数的功能是对来自单片机/INT1 引脚的信号进行计数。

```
Main()
{
int x=0;
EA=1;
EX1=1;
IT1=1;
while(1);
}

Int_1() interrupt 2
```

```
{
x++;
}
```

13、阅读下列程序，给程序加注释，说明程序实现的功能，画出流程图。用汇编语言编制实现同样功能的程序。

```
#include<absacc.h>
#define ADDR 0x0040// ADDR=0x0040
#define max XBYTE[0x0045]// max 为片外 RAM 0x0045 单元
void main(void)
{
  char n;
  for(n=0;n<4;n++)
  {
    if(XBYTE[ADDR+n+1]<XBYTE[ADDR+n])//如果前面的数据大于后面的
      {
       max=XBYTE[ADDR+n];
       XBYTE[ADDR+n]=XBYTE[ADDR+n+1];
       XBYTE[ADDR+n+1]=Max; //将两个数据交换位置，即把小的数据放在前面
       }
    else
       max=XBYTE[ADDR+n+1];//把相邻两个数据中大的放在 max 中
  }
}
```

程序功能：这是冒泡排序的一次排序，把数据序列中最大的数排在该数据序列的最后位置。

汇编语言程序（无符号数）

```
DDR0 EQU 30H
ORG 0000H
AJMP ST
ORG 0100H
ST:  MOV DPTR,#40H
     MOV R0,#30H
     MOV R7,#5
ST1 :MOVX A,@DPTR
     MOV @R0,A
     INC R0
     INC DPTR
     DJNZ R7,ST1;以上程序是将片外数据串搬到片内 30H 开始的单元中
     MOV A,DDR0；以下是相邻两数比较，小数在前，大数在后
     CJNE A,DDR0+1,ST2
ST20:MOV A,DDR0+1
     CJNE A,DDR0+2,ST3
```

```
ST30:MOV A,DDR0+2
     CJNE A,DDR0+3,ST4
ST40:MOV A,DDR0+3
     CJNE A,DDR0+4,ST5
ST50:SJMP ST6
ST2: JC ST20
     MOV DDR0+5,A
     MOV DDR0,DDR0+1
     MOV DDR0+1,A
     SJMP ST20
ST3: JC ST30
     MOV DDR0+5,A
     MOV DDR0+1,DDR0+2
     MOV DDR0+2,A
     SJMP ST30
ST4: JC ST40
     MOV DDR0+5,A
     MOV DDR0+2,DDR0+3
     MOV DDR0+3,A
     SJMP ST40
ST5: JC ST50
     MOV DDR0+5,A
     MOV DDR0+3,DDR0+4
     MOV DDR0+4,A
     SJMP ST50
ST6: MOV DPTR,#40H；一次排序后，将结果搬回到片外从 0040H 开始的单元中
     MOV R0,#30H
     MOV R7,#5
ST7: MOV A,@R0
     MOVX @DPTR,A
     INC R0
     INC DPTR
     DJNZ R7,ST7
     SJMP $
END
```

14、阅读下列程序，给程序加注释，说明程序实现的功能，画出流程图。

```
#include<reg51.h>
#define ON 0
#define OFF 1
sbit lamp1=P0^0;
sbit lamp2=P0^1;
void initial();
void delay();
```

```
main()
{
  initial();
  delay();
  while(1)
  {
    lamp1=ON;
    delay();
    lamp1=OFF;
    delay();
    lamp2=ON;
    delay();
    lamp2=OFF;
    delay();
  }
}

void initial()
{
P0=1;
}
void delay()
{
int i=0;
while(i<30000)i++;
}
```

程序功能：P0.0 和 P0.1 管脚各接一个发光二极管（低电平点亮）。该程序使两个发光二极管一亮一灭。

15、设计一个单片机控制 LED 流水灯应用系统。具体要求为：设置 3 个开关 K0、K1、K2，当 K0 闭合时，8 个 LED 灯都亮；当 K1 闭合时，8 个 LED 灯先从左到右逐个点亮，再从右到左逐个点亮；当 K2 闭合时，8 个 LED 灯都灭。

解：

K0、K1、K2 均为按键开关，分别接 P1.0、P1.1、P1.2 管脚（另一端接地）；

8 个 LED 灯接 P0 口，从左到右的管脚为 P0.0～P0.7（低电平点亮）。

```
#include<reg51.h>
sbit K0=P1^0;
sbit K1=P1^1;
sbit K2=P1^2;
void delay();
main()
{
  while(K0==0)
```

```
{
  P0=0;
}
while(K1==0)
{
  P0=0xfe;
  delay();

  P0=0xfc;
  delay();

  P0=0xf8;
  delay();

  P0=0xf0;
  delay();

  P0=0xe0;
  delay();

  P0=0xc0;
  delay();

  P0=0x80;
  delay();

  P0=0x00;
  delay();
  P0=0xff;

  P0=0x7f;
  delay();

  P0=0x3f;
  delay();

  P0=0x1f;
  delay();

  P0=0x0f;
  delay();

  P0=0x07;
  delay();
```

```
    P0=0x03;
    delay();

    P0=0x01;
    delay();

    P0=0x00;
    delay();
  }
  while(K2==0)
  {
    P0=0xff;
  }
}

void delay()
{
int i=0;
while(i<30000)i++;
}
```

第六章、MCS-51 系列单片机的中断系统与定时/计数器（P154）

1、什么是中断？中断有什么用途？

所谓中断是

2、中断与调用子程序有什么区别？

共同点：都是中断原来的程序，转去执行另一个程序，结束后返回原来的断点继续执行原来的程序。

不同点：子程序调用是程序中预先设置好的，调用子程序时只保护断点；而中断请求是随机发生的，中断时不仅要保护断点，而且还要保护断点的现场。

3、80C51 能处理哪几种中断源？它的外部信号触发中断可接受什么样的信号？如何编程选择？

4、如果 80C51 的一个中断源发出中断请求，在什么条件下 CPU 才能响应？

这个中断源的中断允许位为“1”，中断总允许位为“1”，这个中断源的中断优先级要高于正在执行的程序的优先级。

5、如果没有给各中断源设置优先级，那么哪个中断源的优先级最高？哪个中断源的优先级最低？

优先级最高的是/INT0，优先级最低的是串口。

6、试编程设置/INT1 具有最高的优先级，T0 具有最低的优先级，不允许其它中断源中断。

```
IE=0x86;
IP=4;
```

7、中断标志位的含义是什么？为什么在 CPU 响应中断后需要将它们清零？哪些中断源的中断标志位需要使用软件将它们清零？

中断标志位为 1 时，表示这个中断源发出了中断请求。在 CPU 响应中断后，必须将这个中断标志位清零，否则会再次响应中断。RI 和 TI 需要使用软件清零。

8、80C51 程序存储器的 0000H～0030H 单元一般如何使用？

80C51 程序存储器的 0000H～0030H 单元中，系统做了如下分配：

0000H：主程序的首地址

0003H：外中断 0 的中断服务程序首地址

000BH：定时/计数器 0 的中断服务程序首地址

0013H：外中断 1 的中断服务程序首地址

001BH：定时/计数器 1 的中断服务程序首地址

0023H：串行接口的中断服务程序首地址

从这些首地址可以了解，两个首地址之间最大距离 8 字节，无法放置程序。通常的做法是在各个首地

址存放一条无条件转移指令，跳到对应的程序执行。

9、在什么情况下中断服务程序才能得以执行？在用 C51 编写中断服务程序时，如何区分不同中断源的中断服务程序？

只有在中断请求得到响应后，方可执行对应中断请求的中断服务程序。

用 C51 编写中断函数时，其函数名具有如下特征：中断函数名() interrupt m，这里的 m 被称为中断号，其中“0”表示外中断 0，“1”表示定时/计数器 0，“2”表示外中断 1，“3”表示定时/计数器 1，“4”表示串行接口。系统可根据 m 找到中断服务程序首地址。

10、阅读下列程序，给程序加注释，说明程序实现的功能，编写出实现该功能的 C51 程序。

```
      ORG 0000H
      AJMP START
      ORG 0013H
      AJMP INT_1
      ORG 0030H
START:MOV SP,#50H
      MOV P1, #0FFH
      MOV A,#01H
      SETB EA
      SETB EX1
      SETB IT1
      SJMP $
INT_1:MOV P1,A
      RL A
      RETI
      END
```

程序功能：

电路设计 P1 口接 8 个 LED 灯（高电平亮），外中断 1 的管脚接一个按键（其另一端接地）。

主程序预设使 8 个 LED 灯都亮。第一次按下键，P1 口的最低位 LED 灯亮（其它均灭），以后每有一次外中断 1 的中断请求，LED 亮灯逐步向高位移动（其它均灭），若 P1 口的最高位 LED 灯亮时（其它均灭），在来一个中断请求，P1 口的最低位 LED 灯亮（其它均灭），循环往复。

C51 程序：

```
#include<reg51.h>
Unsigned char x=0xfe;
main()
{
  EA=1;
  EX1=1;
  IT1=1;
  while(1);
}
INT_1() interrupt 2
{
```

```
  P1=x;
  x=x<<1;
  if(x==0)
    {
     x=0xfe;
     }
}
```

11、80C51 内部有几个定时/计数器？它们有什么用途？如何将它们设置为定时器使用？

有 2 个定时/计数器。

可以作为定时器或计数器使用。

通过软件（汇编语言或 C51）进行设置。如设置定时/计数器 0 为定时器(工作方式 0)，使 TMOD=0，再设初值。

12、单片机的定时/计数器内部的 TMOD 和 TCON 有什么用途？

TMOD：可确定 2 个定时/计数器处于何种工作方式，以及是定时器还是计数器。

TCON：其中的 IT0 和 IT1 是外中断 0 或 1 的外部信号触发方式（“1”为边沿触发，“0”为电平触发）；TR0/TR1 是定时/计数器 0/定时/计数器 1 的启动计数开关。

13、归纳总结定时/计数器的工作方式 1、工作方式 2、工作方式 3 的各自特点。

工作方式 1：

16 位计数器，THi 是计数器的高 8 位，TLi 是计数器的高 8 位，通常需要设初值。由 TRi 作为计数器启动开关，当计数器满溢出时，产生中断请求信号。

工作方式 2：

8 位计数器，TLi 是计数器，THi 是初值寄存器。当计数器满溢出时，产生中断请求信号，系统自动重装初值。

工作方式 3：

定时/计数器 0 被拆成 2 个独立的 8 位计数器，其中 TL0 既可为计数器也可为定时器使用，TR0 作为计数器启动开关，TF0 作为其中断请求信号。这时，TH0 就没有那么多“资源”可以利用了，只能作为定时器使用，TR1 作为计数器启动开关，TF1 作为其中断请求信号。

在定时/计数器 0 处于工作方式 3 时，由于 TR0、TR1、TF0、TF1 均被占用，故定时/计数器 1 只能作为时钟发生器使用。

14、80C51 的晶振频率为 12MHz，当它的定时/计数器作为定时器使用时，在工作方式 1 和工作方式 3 下，定时器的最大定时时间各是多少？

解：

fosc=12MHz，则机器周期为 1μs，作为计数器输入信号。

工作方式 1,16 位计数器，最大定时时间是 65536μs。

工作方式 1,8 位计数器，最大定时时间是 256μs。

15、阅读下列程序，给程序加注释，说明程序实现的功能，编写出实现该功能的 C51 程序。

```
  ORG 0000H
  AJMP START
  ORG 000BH
```

```
        AJMP TIME0
        ORG 0100H
START:MOV SP,#60H
        MOV P1，#0FFH
        MOV TMOD,#01H
        MOV TL0,#0A0H
        MOV TH0,#15H
        SETB EA
        SETB ET0
        SETB TR0
        SJMP $
TIME0:PUSH Acc
        PUSH PSW
        CPL P1.0
        MOV TL0,#0A0H
        MOV TH0,#15H
        POP PSW
        POP Acc
        RETI
        END
```

程序功能：

定时/计数器 0 作为定时器使用，且为工作方式 1；16 位计数器的初值是 A015H（40981），即 24555 个机器周期产生一次定时中断。每中断一次，P1.0 管脚输出电平取反一次。

C51 程序：

```
#include<reg51.h>
Sbit PP=P1^0;
main()
{
  SP=0x60;
  TMOD=1;
  TH0=24555/256;
  TL0=24555%256;
  EA=1;
  ET0=1;
  TR0=1;
  while(1);
}
TIMER0() interrupt 1
{
  PP=!PP;
}
```

16、单片机的晶振频率为 12MHz，分别编程实现在 P1.0 引脚输出 2KHz 方波的汇编语言程序和 C51 程序。

单片机的晶振频率为 12MHz，意味着机器周期为 1μs。

2KHz 方波的周期为 0.5ms=500μs，即 250μs 改变一次电平。

方法一：T0 工作方式 2，初值=6，即 250 次（250μs）中断一次。

```
ORG 0000H
AJMP START
ORG 000BH
AJMP T0_INT
ORG 0100H
START:SETB EA
SETB ET0
MOV TH0,#6
MOV TL0,#6
MOV TMOD,#2
SETB TR0
SJMP $

T0_INT: CPL P1.0
RETI
END
```

方法二：T0 工作方式 1，初值=65536-250=65286=FF06H，即 250 次（250μs）中断一次。

```
ORG 0000H
AJMP START
ORG 000BH
AJMP T0_INT
ORG 0100H
START:SETB EA
SETB ET0
MOV TH0,#0FF6H
MOV TL0,#6
MOV TMOD,#1
SETB TR0
SJMP $

T0_INT: MOV TH0,#0FFH
MOV TL0,#6
CPL P1.0
RETI
END
```

方法三：T0 工作方式 0，初值=8192-250=7942=1F06H（用低 13 位），TH0=F8H，TL0=06H，即 250 次（250μs）中断一次。

```
ORG 0000H
AJMP START
ORG 000BH
AJMP T0_INT
ORG 0100H
START:SETB EA
SETB ET0
MOV TH0,#0F8H
MOV TL0,#6
MOV TMOD,#0
SETB TR0
SJMP $

T0_INT:MOV TH0,#0F8H
MOV TL0,#6
CPL P1.0
RETI
END
```

第七章、MCS-51 系列单片机的串行通信（P184）

1、什么是异步串行通信？与其它通信方式相比，它具有哪些优点？

2、什么是波特率？波特率为 9600 表示什么意思？

3、计算机内部的并行数据是怎样利用 UART 变成串行数据发出去的？

4、写出字符“B”在 8 个数位、1 个停止位、1 个奇偶校验位的字符帧格式。

5、51 系列单片机串行口有哪几种工作方式？说明这几种工作方式的特点。

6、如果要在单片机串行通信时使用奇偶校验可以选择哪几种工作方式？如果希望通信的波特率可变可采用哪种工作方式？

7、在单片机串行通信时，什么情况下要使用定时/计数器 1？这时 T1 应采用哪种工作方式？波特率又如何计算？

8、设置串口工作于方式 3，波特率为 9800bit/s,系统主频为 11.0592MHz，允许节省数据，串口开中断，编程实现上述要求的初始化程序；若将串口设置工作方式 1，应如何修改初始化程序？

9、说明如何利用 51 系列单片机串行口提供的功能实现奇偶校验？

10、阅读下列程序，说明程序完成的功能。并用 C51 写出同样功能的程序。

```
        MOV SON,#80H
        MOV PCON,#80H
        MOV R7,#20H
        MOV R0,#50H
START:MOV A,@R0
        MOV C,P
        MOV TB8,C
        MOV SBUF,A
```

```
WAIT: JBC TI,CONT
      AJMP WAIT
CONT: INC RO
      DJNZ R7, START
      RET
```

11、设计并实现一个单片机与 PC 点对点通信系统。具体要求为：单片机每 10 秒向 CPU 发送一个数据，PC 将接收到的数据在显示器上显示。写出设计说明书，内容包括设计思想、硬件电路图、程序流程图、程序清单、运行结果截图、存在问题及解决办法。

第八章、单片机应用中的人机接口（P244）

自编题：用逐行扫描法或行反转法求 P194 图 8.3 中的所有按键的键值。

K0： 11101110B=EEH，K1： 11011110B=DEH，K2： 10111110B=BEH，K3： 01111110B=7EH；
K4： 11101101B=EDH，K5： 11011101B=DDH，K6： 10111101B=BDH，K7： 01111101B=7DH；
K8： 11101011B=EBH，K9： 11011011B=DBH，K10：10111011B=BBH，K11：01111011B=7BH；
K12：11100111B=E7H，K13：11010111B=D7H，K14：10110111B=B7H，K15：01110111B=77H。

第九章、单片机应用中模拟量的输入输出（P278）

5、动手设计一个方波和三角波发生器的软硬件，电压变化的范围是 0～5V，频率自选。

使用 P273 的图 9.11，将图中放大器的正负极互换，即可形成输出 0～5V 的电压。或使参考电压为-5V 即可。

C51 程序生成方波：

```
#include<reg51.h>
#define PORT XBYTE[0xFF40]
delay()
{ unsigned char i=0;
 while(i<250)i++;}
Main()
{
 while(1)
 {PORT=255;
  delay();
  PORT=0;
  delay();
 }
}
```

C51 程序生成三角波：

```
#include<reg51.h>
#define PORT XBYTE[0xFF40]
unsigned char n=;
main()
{
 while(1)
 {
  for(n=0;n<=255;n++)
    PORT=n;
  for(n=255;n<=0;n--)
    PORT=n;
 }
}
```

6、使用 ADC0809 芯片，设计一个模拟量数据采集系统，画出电路图和程序流程图，写出程序。模拟输入电压的范围是 0～5V，要求使用 IN0 和 IN6 两个通道循环采集数据。

使用图 9.5 电路，将转换后的数字量存入片内 RAM30H 和 31H 中。

```
#include<reg51.h>
#include<absacc.h>
#define PORT XBYTE[0x00F40]
 bit i=0;//“1”时表示 ADC0809 芯片处于闲置状态，即可以进行转换
char j;//模拟量通道号
main()
{
 while(1)
 {
  j=0;
  PORT=j;//选择 IN0 并启动 A/D 转换
  while(i=0);
  i=0;
  j=6;
  PORT=j;//选择 IN60 并启动 A/D 转换
  while(i=0);
  i=0;
 }
}

INT-1() interrupt 2
{
 i=1;
 if(j==0)
   DBYTE[0x30]=PORT;
 Else
   DBYTE[0x31]=PORT;
}
```

第十章、存储器与并行接口扩展（P294）

1、其止地址为 0000H～5FFFH 的存储器的容量是多少？如果采用 SRAM6264 芯片，需要几片？

24K，3 片。

2、请说明当 MCS-51 系列单片机的/EA 引脚接高电平，CPU 取指令时，PC 的取值范围；当 MCS-51 系列单片机的/EA 引脚接地，CPU 取指令时，PC 的取值范围。

/EA 引脚接高电平，表示使用片内 ROM 和片外 ROM，PC 的取值范围：先片内后片外；/EA 引脚接地，表示只使用片外 ROM，PC 的取值范围：片外 64K。

4、试用 SRAM 6264 构建 32K 的外部数据存储器，地址范围：0000H～7FFFH。并设计与 MCS-51 系列单片机连接的硬件结构图。

SRAM 6264 为 8K*8 位的芯片，要构建 32K 的外部数据存储器，需要 4 片这样的芯片。地址范围：0000H～7FFFH，说明：

0#6264：0000H～1FFFH，即 000**0**,0000,0000,0000～000**1**,1111,1111,1111

1#6264：2000H～3FFFH，即 001**0**,0000,0000,0000～001**1**,1111,1111,1111

2#6264：4000H～5FFFH，即 010**0**,0000,0000,0000～010**1**,1111,1111,1111

3#6264：6000H～7FFFH，即 011**0**,0000,0000,0000～011**1**,1111,1111,1111

由上述所列可以看出：对每个芯片来讲，其最高 3 位地址是一样的。这样，这 3 位地址可以作为该芯片的地址译码输入生成片选信号。

由于 SRAM 6264 有两个片选信号 CS1 和/CS1，可以使其中一个常有效，另一个受控来解决片选问题。

方法一：CS1 常有效时。

0#6264：A15A14A13=000，则/CS1= A15+A14+A13

1#6264：A15A14A13=001，则/CS1= A15+A14+$\overline{A13}$

2#6264：A15A14A13=010，则/CS1= A15+$\overline{A14}$+A13

3#6264：A15A14A13=011，则/CS1= A15+$\overline{A14}$+$\overline{A13}$

方法二：/CS1 常有效时。

0#6264：A15A14A13=000，则 CS1=$\overline{A15}\cdot\overline{A14}\cdot\overline{A13}$

1#6264：A15A14A13=001，则 CS1=$\overline{A15}$ · $\overline{A14}$ · A13

2#6264：A15A14A13=010，则 CS1=$\overline{A15}$ · A14 · $\overline{A13}$

3#6264：A15A14A13=011，则 CS1=$\overline{A15}$ · A14 · A13

有了片选信号，就可以方便地画出与单片机连接的硬件结构图。

单片机端	中间部分	0#6264	1#6264	2#6264	3#6264
P0 口	数据总线直接接	D7～D0	D7～D0	D7～D0	D7～D0
P0 口	经锁存器输出低 8 位地址总线直接接	A7～A0	A7～A0	A7～A0	A7～A0
P2 口	低 5 位（P2.4～P2.0）地址总线直接接	A12～A8	A12～A8	A12～A8	A12～A8
ALE	接锁存器控制端 G				
P2 口	高 3 位（P2.7～P2.5）接译码器输入生成片选信号分别接对应的 SRAM 6264				
/RD		/OE	/OE	/OE	/OE
/WR		/WE	/WE	/WE	/WE

5、试用 EPROM2764 构建 32KB 的外部程序存储器，地址范围 1000H～8FFFH。并设计与 MCS-51 系列单片机连接的硬件结构图。

0#2764：0001,0000,0000,0000～0010,1111,1111,1111

1#2764：0011,0000,0000,0000～0100,1111,1111,1111

2#2764：0101,0000,0000,0000～0110,1111,1111,1111

3#2764：0111,0000,0000,0000～1000,1111,1111,1111

由此看出：共性在低 12 位，个性在高 4 位，可以得出如下结论：

高 4 位为 0001 或 0010 时均可选中 0#2764；

高 4 位为 0011 或 0100 时均可选中 1#2764；

高 4 位为 0101 或 0110 时均可选中 2#2764；

高 4 位为 0111 或 1000 时均可选中 3#2764；

对于 0#2764：或高 4 位为 0001 时有效，可选中 0#2764；或高 4 位为 0010 时有效，也可选中 0#2764。因此 0001 有效的表达式为 $\overline{A15}$ · $\overline{A14}$ · $\overline{A13}$ · A12，0010 有效的表达式为 $\overline{A15}$ · $\overline{A14}$ · A13 · $\overline{A12}$ 。两个表达式为或的关系，即：$\overline{A15}$ · $\overline{A14}$ · $\overline{A13}$ · A12+ $\overline{A15}$ · $\overline{A14}$ · A13 · $\overline{A12}$，因为 2764 的片选为低电平有效。

则 0#2764：/CE=/($\overline{A15}$ · $\overline{A14}$ · $\overline{A13}$ · A12+ $\overline{A15}$ · $\overline{A14}$ · A13 · $\overline{A12}$)

同理：1#2764：/CE=/($\overline{A15}$ · $\overline{A14}$ · A13 · A12+ $\overline{A15}$ · A14 · $\overline{A13}$ · $\overline{A12}$)

2#2764：/CE=/($\overline{A15}$ · A14 · $\overline{A13}$ · A12+ $\overline{A15}$ · A14 · A13 · $\overline{A12}$)

3#2764：/CE=/($\overline{A15}$ · A14 · A13 · A12+A15 · $\overline{A14}$ · $\overline{A13}$ · $\overline{A12}$)

7、决定 8255A 选口地址的引脚有哪些？作用是什么？

A1 和 A0 管脚，通常，A1 和 A0 分别与地址总线的 A1 和 A0 直接连接。

A1A0=00：选择 8255A 的 PA 口；

A1A0=01：选择 8255A 的 PB 口；

A1A0=10：选择 8255A 的 PC 口；

A1A0=11：选择 8255A 的控制寄存器。

9、请写出 8255A 的 A 口工作在方式 2，B 口工作在方式 1 时的控制字。

11XXX1XXB=C4H

10、请写出 8255A 的 A 口工作在方式 1，输入；B 口工作在方式 0，输出的控制字。

1011X00XB=B0H