

## 第一章

### 1.什么是单片机？单片机和通用微机相比有何特点？

答：单片机又称为单片微计算机，它的结构特点是将微型计算机的基本功能部件（如中央处理器 CPU、存储器、输入接口、输出接口、定时/计数器及终端系统等）全部集成在一个半导体芯片上。虽然单片机只是一个芯片，但无论从组成还是从逻辑功能上来看，都具有微机系统的定义。与通用的微型计算机相比，单片机体积小巧，可以嵌入到应用系统中作为指挥决策中心，使应用系统实现智能化

### 3.举例说明单片机的主要应用领域。

答：单片机广泛应用于仪器仪表、家用电器、医用设备、航空航天、专用设备的智能化管理及过程控制等领域

## 第二章

### 2.1. MCS-51 单片机内部包含哪些主要功能部件？它们的作用是什么？

(1) CPU :8 位的 CPU，包括了运算器和控制器两大部分。

控制器:是对取自程序存储器中的指令进行译码,在规定的时刻发出各种操作所需要的控制信号,完成指令所规定的功能；

运算器：根据控制发来的信号，执行算术逻辑运算操作；

(2)存储器：包括程序存储和数据存储器；

(3)接口部分：定时器计数器：2 个 16 位定时器/计数器，可对机器周期计数，也可对外部输入脉冲计数；

(4) 中断系统：可响应三个内部中断源和两个外部中断源的中断请求；

(5) 输入输出接口：4 个 8 位并行口和一个全双工串行口。

2. MCS-51 单片机外部引脚的名称是什么？各有什么功能？

答：(1) 电源及晶振引脚

VCC(40 脚)：+5V 电源引脚

VSS(20 脚)：接地引脚

XTAL1(19 脚)：外接晶振引脚（内置放大器输入端）

XTAL2(18 脚)：外接晶振引脚（内置放大器输出端）

(2) 控制引脚

RST/ $V_{DD}$ (9) 为复位/ 备用电源引脚

ALE/ $\overline{\text{PROG}}$ (30) 为地址锁存使能输出/ 编程脉冲输入

$\overline{\text{PSEN}}$ (29)：输出访问片外程序存储器读选通信号

$\overline{\text{EA}}$ / VPP (31)：外部 ROM 允许访问/ 编程电源输入

(3) 并行 I/O 口引脚

P0.0~P0.7 (39~32 脚) ——P0 口；

P1.0~P1.7 (1~8 脚) ——P1 口；

P2.0~P2.7 (21~28 脚) ——P2 口；

P3.0~P3.7 (10~17 脚) ——P3 口。

5. 通常的微机系统存储器结构有哪几种？MCS-51 单片机存储器属于哪一类？

CS-51 可寻址多大空间？

答：通常的微机系统存储器结构有两种结构，即哈佛结构和冯诺依曼结构(也叫普林斯顿结构)，MCS-51 单片机存储器属于，MCS-51 可寻址空间是两个 64KB，即 64KB 的程序存储空间和 64KB 的数据存储空间。

6. 片内 RAM 中低 128 个单元划分为哪三个主要部分？各部分的主要功能是什么？

答：片内 RAM 中低 128 个单元划分为三个部分：

①工作寄存器区 (00H-1FH)，四组，每组 8 个，可作用工作寄存器切换使用；

②可位寻址区 (20H-2FH)，16B，位地址为 00H-7FH，用作为按位寻址的空间；

③用户 RAM 区 (30H-7FH)，80B，用作普通 RAM 单元或堆栈。

7. 程序状态字寄存器 PSW 各位的定义是什么？

答：程序状态字寄存器 PSW 各位的定义如下：

PSW. 7	PSW. 6	PSW. 5	PSW. 4	PSW. 3	PSW. 2	PSW. 1	PSW. 0
CY	AC	F0	RS1	RS0	OV	F1	P

PSW. 7: 进/借位标志 CY，加法有进位时置 1，减法有借位时置 1；

PSW. 6: 辅助进位标志 AC，加法运算低四位向高上四位有进位时置 1；

PSW. 5、PSW. 1: 用户标志位 F0 和用户标志位 F1，

保存用户的位数据；

PSW. 4、PSW. 3: 工作寄存器选择控制位 RS1 和 RS0，00 至 11 分别选择四组工作之一作为当前工作寄存器

PSW. 2 : 溢出标志位 OV，有符号数加、减运算结果有溢出或乘除上结果异常(乘法运算结果大于 255 即乘积在 BA 中，或除法运算除数为 0)时置 1

PSW. 0: 奇偶标志位 P，累加器 A 中 1 的个数为奇数时置 1。

8. 什么是时钟周期？什么是机器周期？什么是指令周期？当振荡频率为 12MHz 时，一个机器周期为多少微秒？

答：时钟周期又叫振荡周期或拍，用 P 表示，是 MCS-51 单片机中最小的时间单位，在一个时钟周期内，CPU 完成一个最基本的动作。

机器周期：由 12 个时钟周期构成，完成一个基本操作

指令周期：是执行一条指令所需的时间，根据指令的复杂性，可由 1~4 个机器周期构成。

当振荡频率为 12MHz 时，一个机器周期为 1 微秒。

9. P0、P1、P2 和 P3 口的结构和功能分别是什么？

答：P0 口的每一位由 1 个锁存器、2 个三态缓冲器、1 个输出控制电路（非门 X、与门、电子开关 MUX、输出驱动电路构成，其功能既可以作为通用 I/O 口实现输入/输出功能，也可作为单片机地址线的低 8 位和数据线实现外部扩展功能。在用作输入输出口时，需外接上拉电阻。

P1 口的每一位由 1 个锁存器、1 个场效应管驱动器 V 和 2 个三态门缓冲器

---

构成，其作用是用作输入输出口

P2 口的每一位由 1 个锁存器、2 个三态缓冲器、1 个输出控制单元、1 个输出驱动单元构成，其功能是用作输入输出口，或地址总线的高 8 位。

P3 口的每一位由 1 个锁存器、2 个三态缓冲器、1 个第二功能控制单元、1 个输出驱动单元构成，其作用是用作输入输出口或第二功能。

10. 80C51 单片机引脚 ALE 的作用是什么？当 80C51 不外接存储器时，ALE 上的输出的脉冲频率是多少？

答：80C51 单片机引脚 ALE 的作用是对外部存储寻址时锁存 P0 口输出的低 8 位地址，当 80C51 不外接存储器时，ALE 上的输出的脉冲频率是  $f_{osc}/6$ 。

### 第三章

11. 编制程序，将内部 RAM 中 M1、M2、M3 和 M4 单元中的无符号数 x1、x2、x3 和 x4 相加，并把和存入 R0 和 R1 (R0 中为高 8 位) 中。

解：

```
M1    EQU    30H
M2    EQU    40H
M3    EQU    45H
M4    EQU    4FH
      MOV     A, M1      ;取第一个数
      ADD     A, M2      ;与第二个数相加
      JNC     NEXT1     ;如果无进位，则转移至第三个相加
```

---

```
      MOV     R0, #1     ;有进位，高 8 位置 1
NEXT1: ADD     A, M3      ;与第三个数相加
      JNC     NEXT2     ;没有进位，则转至第四个数相加
      INC     R0         ;有进位，高 8 位加 1
NEXT2: ADD     A, M4      ;与第四个数相加
      JNC     NEXT3     ;没有进位，则转至结束
      INC     R0         ;有进位，高 8 位再加 1
NEXT3: MOV     R1, A      ;低 8 位保存到 R1
      SJMP    $          ;暂停
      END
```



## 第四章

1. C 语言的优点是什么？C 程序的主要结构特点是什么？

答：C 语言是一种高级语言，学习比低级容易，不需要具体组织、分配存储器资源和处理端口数据，可以直接驱动单片机的所有资源。

C 程序以函数为单位，由一个主函数和若干个其他函数构成，主函数是程序的入口，其他函数由主函数直接或间接调用。程序可以由一个文件或多个文件组成。文件类型包括头文件和 C 语言源文件，也可以是汇编语言文件，C 程序可与汇编语言混合编程。

### 4.2. C 语言的变量定义包含哪些因素，为何这样考虑？

【存储种类】   数据类型   【存储类型】   变量名

存储种类用于说明变量的作用范围：

1、auto（自动型）——变量的作用范围在定义它的函数体或语句块内。执行结束后，变量所占内存即被释放。

2、extern（外部型）——在一个源文件中被定义为外部型的变量，在其它源文件中需要通过 extern 说明方可使用。

3、static（静态型）——利用 static 可使变量定义所在的函数或语句块执行结束后，其分配的内存单元继续保留。

数据类型除了与包括标准 C 的字符型、整型、浮点型、普通指针，还有 C51 扩充的 3 种数据类型：bit、sfr 或 sfr16、sbit

存储类型体现了变量的存放区域。51 系列单片机共有 6 个存储类型（分布在 3 个逻辑存储空间中）。片内 RAM,data，片内可位寻址,bdata、片内 RAM 间接寻址,idata、片外 RAM 页寻址,pdata、片外 RAM,xdata、ROM 区,code。定义变量时要指出它所在的存储空间，编译模式决定了存储器类型缺省时变量所在的存储空间。

3. C51 与汇编语言的特点各有哪些？怎样实现两者的优势互补？

答：C51 是结构化语言，代码紧凑；接近自然语言，程序可读性强，易于调试、维护；库函数丰富，编程工作量小，可使产品开发周期短；具有机器级控制能力，功能很强，适合于嵌入式系统开发；汇编指令无关，易于掌握，上手快。

汇编语言优点是编写的程序代码精炼、执行速度快，在相同功能下，汇编语言程序可能比 C 语言程序效率高。缺点是对程序员要求高，必须对单片机的硬件

---

结构非常熟悉才能编程，不便于编写比较复杂的程序。

可根据需要对要求时间性很强的代码用汇编语言编写，其它部分用 C 语言编写，两者混合编程就可以实现优势互补。

4. 指出下面程序的语法错误：

```
#include<reg51.h>
main() {
    a=C;
    int a=7,C
    delay(10)
    void delay();{
    cgar i;
    for(i=0; i<=255; " ++" );
}
```

答：

```
#include<reg51.h>
main() {
    a=C;           //a 和 C 必须先定义才可使用
    int a=7,C      //缺分号，应该提在使用前说明
    delay(10)      //延时子程序必须先定义，或先有函数原型说明
    void delay();{
        //作为函数定义，有函数定义不能嵌套的问题，
        //即不能在一个函数内定义另一个函数，而且“{”前不应有分号
        //作为函数原型说明，应该放在函数调用之前，且其后不应该接函数体
        cgar i;      //字符型应该是 unsigned char
        for(i=0; i<=255; " ++" );// " ++" 应改成 i++
    }
    //主函数没有结束，缺"}"
```

5. 定义变量 a, b, c, 其中 a 为内部 RAM 的可位寻址区的字符变量, b 为外部数据存储器区浮点型变量, c 为指向 int 型 xdata 区的指针。

答:     char bdata a;  
          float xdata b;  
          int xdata \*c;

6. 编程将 8051 的内部数据存储器 20H 单元和 35H 单元的数据相乘, 结果存到外部数据存储器中(任意位置)。

<p>解: 方法一: 用嵌入式汇编语言实现</p> <pre>#include&lt;reg51.h&gt; void main() {  #pragma    asm    MOV  A,20H    MOV  B,35H    MUL  AB  </pre> <hr/> <pre>   MOV  DPTR,#1234H    MOVX @DPTR,A    INC  DPTR    MOV  A,B    MOVX @DPTR,A    #pragma    endasm }</pre>	<p>方法二: 单用 C 语言编程实现</p> <pre>#include&lt;reg51.h&gt; #include&lt;math.h&gt; int movdata(char); void main() {    unsigned int xdata x;    unsigned char *ptr,a,b;    ptr=0x25;    a=*ptr;    ptr=0x30;    b=*ptr;    x=a*b; }</pre>
--	--



7. 8051 的片内数据存储器 25H 单元中存放有一个 0~10 的整数，编程求其平方根(精确到 5 位有效数字)，将平方根放到 30H 单元为首址的内存中。

方法二：单用 C 语言编程实现

```
//MAIN.C
#include<reg51.h>
#include<math.h>
int movdata(char);
void main()
{
    char n;
    char *ptr;
    float *ptr2;
    float f;
    ptr=0x25;
    n=*ptr;
    f=sqrt(n);
    ptr2=0x30;
    *ptr2=f;
}
```

8. 将外部 RAM 10H~15H 单元的内容传送到内部 RAM 10H~15H 单元。

方法二：单用 C 语言编程实现

//MAIN.C

---

```
#include<reg51.h>
int movdata(char);
void main()
{
    char n=6;
    char *ptr1=0x10;
    char xdata *ptr2;
    ptr2=0x20;
    while(n--){
        *ptr2++=*ptr1++;
    }
}
```

9. 内部 RAM 20H、21H 和 22H、23H 单元分别存放着两个无符号的 16 位数，将其中的大数置于 24H 和 25H 单元。

解：方法一：

```
#include<reg51.h>
```

```
void main()
```

```
{
    unsigned int *ptr;    //设置一个内部 RAM 指针
    unsigned int x,y,z;
    ptr=0x20;            //指向 0x20 单元
    x=*ptr;              //取第一个数
    ptr=0x22;            //指向 0x22 单元
    y=*ptr;              //取第二个数
    z=(x>y)?x:y;         //将两数中的较大者赋给 z
    ptr=0x24;            //指向地址为 0x24 的目标单元
    *ptr=z;              //将大数存入目标单元
}
```

## 第五章

### 1. 什么是中断、中断源、中断优先级和中断嵌套？

答：中断是指单片机内部有一个中断管理系统，它对内部的定时器事件、串行通信的发送和接收及外部事件（如键盘按键动作）等进行自动的检测判断。当 CPU 正在处理某件事情（例如正在执行主程序）的时候，外部或内部发生的某一事件（如某个引脚上电平的变化，一个脉冲沿的发生或计数器的计数溢出等）请求 CPU 迅速处理，于是，中断管理系统会置位相应标志通知 CPU 暂时中止当前的工作，迅速转去处理所发生的事件。处理完该事件后，再回到原来被中止的地方，继续原来的工作，这样的过程称为中断。

引发中断的事件称为中断源。

将中断事件按轻重缓急分若干级别叫中断优先级。

允许中断优先级高的中断源中断正在执行的低优先级的中断服务程序叫中断嵌套。

### 2. 什么叫中断源？MCS-51 有哪些中断源？各有什么特点？它们的中断向量地址分别是多少？

答：中断源即引发中断的事件。

MCS-51 单片机有 5 个中断源，它们是外部中断 0，定时器 T0，外部中断 1，定时器 T1，串行口。

外部中断源是由引脚的触发信号引起的中断，定时器中断源是由于定时器计数器的溢出引发的中断，串行口是由于串行通信的发送或接收引发的中断。

外部中断 0，定时器 T0，外部中断 1，定时器 T1，串行口五个中断源的中断向量地址依次为：0003H，000BH，0013H，001BH，0023H。

## 5.3，5.4. 89C51 单片机中响应中断是有条件的，请说出这些条件是什么？并

描述中断响应的全过程。

条件：（1）有中断源发中断请求

（2）EA1=1

（3）此中断源允许位位 1

不存在一些特殊情况：

（1）当前指令不是 RETI 或访问 IE，IP 的指令

（2）CPU 没有处理同级中断或更高级的中断。

硬件生成一条 LCALL 指令。

过程：

- (1) 将相应的中断优先级触发器置 1；
- (2) 硬件清除相应的中断请求标志，串行口的发送和接收中断除外；
- (3) 执行一条硬件 LCALL 指令，即把程序计数器 PC 的内容压入堆栈保存，再将相应的中断服务程序的入口地址送入 PC；
- (4) 将中断优先级触发器复位，返回断点继续执行。

5. 编写出外部中断 1 为下跳沿触发的中断初始化程序。

解：

```
void Int1_init() {  
    IT1=1;  
    EA=1; EX1=1;          //IE=0x84; // IE|=0x84;  
}
```



## 第六章

1. MCS-51 系列的 8051 单片机内有几个定时/计数器？每个定时/计数器有几种工作方式？如何选择？

答：MCS-51 系列的 8051 单片机内有 2 个定时/计数器，即 T0 和 T1，每个都可以编程为定时器或计数器，T0 有四种工作方式（方式 0—13 位、方式 1—16 位、方式 2—可自动装入初值的 8 位、方式 3—两个 8 位），T1 有三种工作方式（与 T0 相同的前三种），通过对 TMOD 的设置选择，其高四位选择 T1，低四位选择 T0。

2. 如果采用的晶振频率为 3MHz，定时/计数器 T0 分别工作在方式 0、1 和 2 下，其最大的定时时间各为多少？

答：如果采用的晶振频率为 3MHz，机器周期为  $12 \times 1 / (3 \times 10^6) = 4\mu s$ ，由于定时/计数器 T0 工作在方式 0、1 和 2 时，其最大的计数次数为 8192、65536 和 256，所以，其最大定时时间分别是：方式 0 为  $8192 \times 4\mu s = 32.768ms$ 、方式 1 为  $65536 \times 4\mu s = 262.144ms$ 、方式 2 为  $256 \times 4\mu s = 1024\mu s$ 。

3. 定时/计数器 T0 作为计数器使用时，其计数频率不能超过晶振频率的多少？

答：由于定时/计数器 T0 作为计数器使用时，是对外部引脚输入的脉冲进行计数，CPU 在每个机器周期采样一次引脚，当前一次采样为高电平，后一次采样为低电平，则为一次有效计数脉冲，所以如果晶振频率为  $f_{osc}$ ，则其采样频率  $f_{osc}/12$ ，两次采样才能决定一次计数有效，所以计数频率不能超过  $f_{osc}/24$ 。

4. 简单说明定时/计数器在不同工作模式下的特点。

答：方式 0 为 13 位的定时/计数器，由 THx 的 8 位和 TLx 的低 5 位构成，方式 1 为 16 位的定时/计数器，由 THx 的 8 位和 TLx 的 8 位构成，方式 2 为 8 位的定时/计数器，TLx 为加 1 计数器，THx 为计数初值寄存器。方式 3 只能用于 T0，是将 T0 的低 8 位用作一个独立的定时/计数器，而高 8 位的 TH0 用作一个独立的定时器，并借用 T1 的 TR1 和 TF1 作为高 8 位定时器的启停控制位和溢出标志位。

5. 定时器工作在方式 2 时有何特点？适用于什么应用场合？

答：定时器工作在方式 2 时是一个可自动装入时间常数初值的 8 位定时/计数器，TLx 为加 1 计数器，THx 为计数初值寄存器。由于其恢复初值由硬件自动完成，所以当需要反复计数时，用方式 2 可以方便地实现精确的定时。

6. 一个定时器的定时时间有限，如何采用两个定时器的串行定时来实现较长时间的定时？

答：一个定时器的定时时间有限，可采用两个定时器的串行定时来实现较长时间的定时，比如，当  $f_{osc} = 12MHz$  时，单个定时/计数器采用方式 1 的最大定时时间为 65.536ms，此时若用另一个定时/计数器按方式 1 进行溢出次数计数，在定时器溢出中断时，给计数器发一个计数脉冲，且两者均为方式一，则两者串行可达到的定时时间为  $65536 \times 65.536ms = 4294967.296ms$ 。



8. 采用定时/计数器 T0 对外部脉冲进行计数，每计数 100 个脉冲，T0 切换为定时工作方式。定时 1ms 后，又转为计数方式，如此循环不止。假定 MCS-51 单片机的晶体振荡器的频率为 6MHz，要求 T0 工作在方式 1 状态，请编写出相应程序。

解：晶体振荡器的频率为 6MHz，则机器周期为  $12 \times 1 / (6 \times 10^6) = 2\mu s$ ，要定时 1ms，需计数次数为  $1000 / 2 = 500$  次

```
#include<reg51.h>
sbit P1_0=P1^0;
void main() {
    while(1) {
        TMOD=0x05;//T0 计数，方式一
        TH0=(65536-100)/256;//计数 100 次
        TL0=(65536-100)%256;
        TR0=1;
        while(!TF0);//等待计数 100 次的溢出
        TF0=0;      //溢出标志复位
        TMOD=0x01;//T0 定时，方式一
```

```
        TH0=(65536-500)/256;//计数 100 次
        TL0=(65536-500)%256;
        TR0=1;      //启动 T0
        while(!TF0); //等待定时时间到 1ms 的溢出
        TF0=0;      //溢出标志复位
    }
}
```