●.**MCS-51 外扩的程序存储器和数据存储器可以有相同的地址空间，但不会发生数据冲突，为什么？**不发生数据冲突的原因是：MCS-51 中访问程序存储器和数据存储器指令不一样；程序存储器访问指令为 MOVC；数据存储器访问指令为MOVX；选通信号不同，前者为/PSEN,后者为/WR 与/RD。

●**8051 在什么条件下可以响应中断？**由中断源发出中断申请；中断总控制位 EA=1，即 CPU 开中断；申请中断源的中断允许位为 1，即中断没有被屏蔽； 无同级或更高级中断正在服务；当片指令周期已经结束；若现行指令为 RETI 或访问IE 或 IP 指令时，该指令以及紧接着的另一条指令执行结束以后才响应。

●**外扩的程序存储器和外扩I/O端口地址重叠，80C51如何区分这些重叠地址？**

80C51单片机对片外数据存储器、片内数据存储器及程序存储器采用不同的指令，会产生不同的控制信号。片外数据存储器有读RD和写WR控制信号，程序存储器有读PSEN控制信号，因此，扩展时虽然数据线和地址线重复，但由不同的控制信号加以区别。片内数据存储器地址采用MOVC指令，不会产生RD和写WR控制信号。

●**外部中断INT0的中断响应过程？**中断采样：中断采样是针对外部中断请求信号进行的，可以直接置位TCON或SCON中的中断请求标志；2、中断查询：若查询到某中断标志为1，则按优先级的高低进行处理，即响应中断；3、中断响应：首先将当前程序计数器PC的内容压入堆栈进行保护，再将对应中断源的中断矢量地址装入PC，执行中断服务程序。运行直到遇到RETI指令为止，最后恢复原程序的断点地址执行，且恢复中断触发器原先状态。

●**程序存储器和数据存储器地址冲突，如何区分？**

不发生数据冲突的原因是：MCS-51中访问程序存储器和数据存储器指令不一样；程序存储器访问指令为MOVC；数据存储器访问指令为MOVX；选通信号不同，前者为/PSEN,后者为/WR与/RD。

●**中断响应的现场保护？**所谓现场是指中断发生时单片微机中存储单元、寄存器、特殊功能寄存器中的数据或标志位等。因此，在编写中断服务程序时必须考虑保护现场的功能。在80C51单片微机中，现场一般包括累加器A\工作寄存器R0~R7,以及程序状态字PSW等。保护的方法与子程序相同，可以有以下几种：1.通过堆栈操作指令PUSH direct。2.通过工作寄存器区的切换。3.通过单片微机内部存储器单元暂存。

●**ARM处理器异常中断响应过程？**当异常中断发生时，处理器挂起正常模式的执行，首先自动保存当前状态，即返回地址存入链接寄存器R14，当前程序状态寄存器CPSR存入SPSR中，然后进入相应的工作模式，把程序寄存器PC设置为一个特定的存储器地址，这一地址放在一个被称为中断向量表的特定的地址范围内，中断向量表的入口是一些跳转指令，跳转到专门处理某个异常或中断的子程序。

●**ARM的通用寄存器（存储器），R12、R13、R14？**在ARM状态下，通用寄存器包括R0~R15.这些寄存器又可以分为以下三类：1.未分组寄存器R0~R7。2.分组寄存器R8~R14。R12为写入寄存器，用作子程序间的中间结果寄存器，记录着IP。R13通常在ARM指令中用作堆栈指针，简称SP。R14用作子程序返回指针寄存器，称为链接寄存器（LR）：可以保存每一种运行模式下子程序的返回地址3.程序寄存器R15。

●**中断的执行与子程序相似点？**●**中断执行与子程序调用的差别？**

1.中断当前正在执行的程序。2.硬件把断点压栈，软件现场保护。3.通过软件恢复现场，重新返回到断点处，继续执行主程序。4.二者可嵌套，如中断嵌套和子程序嵌套。

1.中断请求由外设发出，是随机的；子程序调用是编排好的。2.中断响应后由固定的矢量地址转入中断服务程序，而子程序地址由软件设定。3.中断响应是受控的，其响应时间会受一些因素影响；子程序响应时间是固定的。

●**程序存储器的特点？**

1、存放经调试正确的应用程序和表格之类的固定常数；

2、采用16位的程序计数器 PC和16位的地址总线，可扩展的地址空间为64 KB；

3、64KB地址是空间连续且统一的。

●**、80C51 单片机的存储器的组织采用何种结构？ 存储器地址空间如何划分？各地址空间的地址范围和容量如何？在使用上有何特点？**采用哈佛结构，在物理上设计成程序存储器和数据存储器两个独立的空间；80C51 基本型单片机片内程序存储器为 4KB，地址范围是 0000H-0FFFH，用于存放程序或常数；片内数据存储器为 128 字节 RAM，地址范围是 00H-7FH，用于存放运算的中间结果、暂存数据和数据缓冲； 另外在80H-FFH 还配有 21 个 SFR。

●**、80C51 有几个中断源？各中断标志是如何产生的？又是如何复位的？CPU 响应各中断时，其中断入口地址是多少？** 5 个中断源，分别为外中断 INT0 和 INT1 、T0 和 T1 溢出中断、串口中断。电平方式触发的外中断标志与引脚信号一致； 边沿方式触发的外中断响应中断后由硬件自动 复位。T0 和 T1，CPU 响应中断时，由硬件自动复位。 RI 和 TI，由硬件置位。必须由软件复位。另外，所有能产生中断的标志位均可由软件 置位或复位。各中断入口地址：INT0 ―0003H， T0—000BH，INT1—0013H，T1—001BH，RI 和 TI—0023H。

●**80C51 单片机的 P0~P3 口在结构上有何不同？ 在使用上有何特点？**作为通用 I/O 口 时，P0-80H真正双向口，P1-90H、P2-A0H 和 P3-B0H都是准双向口（寻址范围都是X0～X7H）。P0 可以作为地址/数据总线，此时是一个真正的双向口；P2 口可以作为地址线的高 8 位；P3 口是双功能口，每条口线还具有不同的第二功能。 P0口的驱动能力为 8 个 TTL 负载，而 其 他 口 仅 可 驱 动 4个TTL负载。

●**80C51单片机的4个I／O口在使用上有哪些分工和特点?**

**1、P**0口是一个多功能的8位口，可按字节访问也可按位访问。 用作I／0口。相当于一个真正的双向口：输出锁存，输入缓冲，但输入时须先将口置l；每根口线可以独立定义为输入或输出。用作地址／数据复用总线。作数据总线使用时，输入／输出8位数据D0～D7；作地址总 线用时，输出低8位地址AO～A7。

2、P1口是一个8位口，可按字节访问也可按位访问，因此，P1口不仅可以8位一组进行 输入、输出操作，还可以逐位分别定义各口线为输入线或输出线。输入时有条件，即需要先输出1，将该口设为输入状态。一般作I／o口用。

3、P2口是一个多功能的8位口,可按字节访问也可按位访问。在单片机采用并行扩展方式时,P2口作为地址总线的高8位D8~D15。

4、P3口是个多功能的8位口,可按字节访问,也按可位访问。可作为I/0口使用,为准双向口。既可以字节操作,也可以位操作;既可以8位口操作,也可以逐位定义口线为输入线或输出线。

●**、80C51 单片机串行口有几种工作方式？简述其特点？**四种工作方式，由 SCON 中的SM0 、SM1 进行定义：方式 0：同步移位寄存器的输入输出方式，主要用于扩展并行输入或输出口，波特率固定。方式 1：10 位数据的异步通信口，波特率可变。方式 2：11 位数据的异步通信口，波特率固定。方式 3：11 位数据的异步通信口，波特率可变。

**●80C51单片机有哪几种工作方式?简单说明其应用场合和特点。**

80C51单片机共有复位、程序执行、低功耗以及编程和校验4种工作方式。

1复位方式:是单片机的初始化操作,其主要功能是把程序计数器Pc初始化为0000H,使单片机从000H单元开始执行程序。除了进入系统的正常初始化之外,当由于程序运行出错或操作错误使系统处于死锁状态时,为摆脱困境,可以按复位键重新启动,也可以通过监视定时器WDT来强迫复位。

2程序执行方式:是单片机的基本工作方式。

3低功耗方式:80C5l单片机有两种低功耗方式,即待机方式和掉电保护方式。用于降低功耗,提高可靠性。

4编程和校验方式:对于片内具有EPROM型程序存储器的87C51(87C52)和片内具有闪速存储器的89C51(89C52)、78E51(78E52)等单片机,可以通过编程来修改程序储器中的程序。

●**80C51单片机的指令系统具有哪些特点?**

80C51单片机的指令系统容易理解和阅读。只要熟记代表33种功能的42种助记即可。有的功能如数据传送,可以有几种助记符,如MOV、MOVc、MOVX。而指令功能助记符与操作数各种寻址方式的结合,共构造出111种指令,而同一种指令所对应的操作码可以多至8种(如指令中Rn对应寄存器R0~R7)。80C51单片机的指令系统具有较强的控制操作类指令,容易实现“面向控制”的功能;具有位操作类指令,有较强的布尔变量处理能力。

**●访问特殊功能寄存器SFR,可使用哪些寻址方式?**

访问特殊功能寄存器SFR的惟一寻址方式是直接寻址方式。这时除了可以单元地址形式(如90H)给出外,还可以寄存器符号形式(如P1)给出。虽然特殊功能寄存器可以使用寄存器符号标志,但在指令代码中还是按地址进行编码的。

●**若访问外部RAM单元,可使用哪些寻址方式?**

访问外部RAM单元的惟一寻址方式是寄存器间接寻址方式。片外RAM的64 KB单元,使用DPTR作为间址寄存器,其形式为@DPTR,例如“MOVX A,@DPTR”的功能是把DPTR指定的片外RAM单元的内容送累加器A。片外RAM低256个单元,除了可使用DPTR作为间址寄存器外,也可使用:R0或R1作间址寄存器。例如“MOVX A,@R0'’即把R0指定的片外RAM单元的内容送累加器A。

**●若访问内部RAM单元,可使用哪些寻址方式?**

片内RAM的低128单元可以使用寄存器间接寻址方式,但只能采用R0或R1为间址寄存器,其形式为@Ri(i=0,1)。片内RAM的低128单元可以使用直接寻址方式,在指令中直接以单元地址形式给出。片内RAM的低128单元中的20H~2FH有128个可寻址位,还可以使用位寻址方式,对这128个位的寻址使用直接位地址表示。

●**若访问程序存储器,可使用哪些寻址方式?**

访问程序存储器可使用的寻址方式有立即寻址方式、变址寻址方式和相对寻址方式三种。立即寻址是指在指令中直接给出操作数。变址寻址方式只能对程序存储器进行寻址,或者说这是专门针对程序存储器的寻址方式。相对寻址方式是为实现程序的相对转移而设立的。三种寻址方式所得到的操作数或操作数地址都在程序存储器中。

●●●**单片微型计算机原理与接口技术**●●●

●**2.3.1中央控制器**是识别指令，并根据指令性质控制计算机各组成部件进行工作的部件，与运算器一起构成中央处理器

**功能：**控制指令的读出、译码和执行，对指令的执行过程进行定时控制，并根据执行结果决定是否分支转移。

**组成：**程序计数器PC、数据指针DPTR、指令寄存器IR、指令译码器、条件转移逻辑电路及定时控制逻辑电路。

●**程序计数器PC：**一个独立的计数器，是中央控制器中最基本的寄存器。

**内容：指令地址。**工作过程：PC变化的轨迹决定程序的流程。PC的宽度决定了程序存储器可直接寻址的范围。

顺序指令 条件转移指令或无条件转移指令

调用指令或响应中断 数据指针 DPTR

**16位特殊功能寄存器.**

**主要功能:用作片外数据存储器或I/O寻址用的地址寄存器**

访问片外数据存储器或I/O的指令为：

MOVX A，＠DPTR 读

MOVX ＠DPTR，A 写

既可以作为一个16位寄存器处理，或者两个8位寄存器，其高8位用DPH表示，低8位用DPL表示。

●**PC与DPTR不同之处：**

⑴ 与地址有关的16位寄存器

PC：程序存储器的地址； DPTR：数据存储器或I/O的地址。

PC的输出与ALE及PSEN有关；DPTR输出与ALE、WR、RD信号有关。

⑵ PC为16位寄存器，不可访问。DPTR为16位寄存器，也可作两个8位特殊功能寄存器，可访问。

●**3. 指令寄存器IR、指令译码器及控制逻辑**

IR:存放指令操作码的专用寄存器。

指令译码器对指令进行译码，译码结果送定时控制逻辑电路。

●**2.3.2 运算器**

**组成:**算术逻辑运算单元ALU、累加器A、暂存寄存器、B寄存器、程序状态标志寄存器PSW以及BCD码运算修正电路等

**1.算术逻辑运算单元ALU**

**ALU有两个输入：**

1)暂存器1的输入：2)暂存器 2或累加器A的输入

**ALU有两个输出:**

1）通过内部总线送回： 累加器A；

2) 标志位输出至程序状态字 PSW。

**2.  累加器A**

**主要功能：**存放操作数，暂存运算结果。单片机中大部分数据操作都要通过累加器A进行，容易产生“瓶颈”现象。

**3．B寄存器**

乘/除法指令中用作ALU的一个输入。

乘法的两个输入为A、B，运算结果，A中放积的低8位，B中放积的高8位。除法中，被除数取自A，除数取自B，商数存放于A，余数存放于B

●**4. 程序状态字PSW （Program Status Word）**

8位寄存器，内容是算术逻辑运算单元（ALU）的输出

**(1) P－奇偶标志位**

表示累加器A中值为1的个数的奇偶性：若累加器值为1的位数是奇数，P置位（奇校验）；否则P清除(偶校验) 。

如(A)=00001010，则P=0。

在串行通信中，常以传送奇偶校验位来检验传输数据的可靠性。

**(2) OV －溢出标志位**

指示运算结果是否溢出。

OV=1：运算结果超出了寄存器A所能表示的带符号数的范围（一128～＋127）。

**(3) RS1、RS0 －工作寄存器组选择位**

**(4)AC －辅助进位标志位**

加减法运算时，低4位向高4位数进位或借位时，AC将被硬件置位；否则，被清除。

在十进制调整指令DA中要用到AC标志位状态。

**(5) CY －进位标志位。**

在进行算术运算时，表示运算结果中高位是否有进位或借位，可以被硬件置位或清除。

**(6)F0 －用户标志位。**

开机时该位为“0”。

用户可根据需要，通过位操作指令置“l”或者清“0”。

●**时序定时单位: 节拍、状态、机器周期和指令周期。**

(1)节拍P: 振荡脉冲的周期称为节拍。

(2) 状态S : 一个状态S包含两个节拍。

(3)机器周期：宽度为6个状态，并依次表示为S1～S6。

一个机器周期有12个振荡脉冲周期。是单片机的最小时间单位。

(4)指令周期： 执行一条指令所需要的时间。是最大的时序定时单位。80C51的指令周期有1、2、4个机器周期。

●**单片机存储器的两种基本结构：**

**1. 普林斯顿(Princeton)结构：**程序和数据合用一个存储器空间

**2. 哈佛（Harvard）结构：**程序存储器和数据存储器截然分开，分别寻址的结构。

**4个物理存储器空间**

**·**程序存储器：①片内程序存储器； ②片外程序存储器。

·数据存储器：③片内数据存储器； ④片外数据存储器。

**3个逻辑存储器地址空间**

①片内、片外统一的 64 KB程序存储器地址空间；

②片内256B数据存储器地址空间；

③片外64 KB的数据存储器地址空间。

**三种基本寻址空间：**

· 64 KB的片内、外程序存储器寻址空间；

· 64 KB的片外数据存储器寻址空间；

· 256B的片内数据存储器寻址空间，包括SFR寻址空间。

●**2.4.1 程序存储器**

功能：存放程序和固定常数。

PC和地址总线为16位 ，可扩展的地址空间为64 KB。

**⒈片内和片外程序存储器**

EA引脚高电平，从片内程序存储器0000H开始执行；

当PC值超出4K，自动转向片外程序存储器空间执行。

EA引脚低电平，从片外程序存储器0000H开始执行。

**⒉ 程序入口地址（中断）**

系统复位后PC为0000H，系统从0000H单元开始取指，执行程序。0003H～002DH用于5个中断源的中断服务程序入口地址。

**⒉ P0口的功能**

(1) I／O口 ：输出锁存、输入缓冲，输入时需先将口置1；每根口线可以独立定义为输入或输出。替代功能：程序存储器、片外数据存储器低8位地址及8为数据

**⑵ 地址／数据复用总线：**作数据总线用时，输入／输出8位数据D0～D7；作地址总线用时，输出低8位地址A0～A7。

**3. P2口的工作状态是**输出高8位地址。

**P2口的**I／O口替代功能**：**程序存储器或片外数据存储器的高8位地址

●**3.1.1 寻址方式**

**寻址方式：指令中给出的寻找操作数或操作数地址的方法。**

**⒈ 立即寻址（程序存储器）**

在指令中直接给出操作数，出现在指令中的操作数称为立即数。立即数前面必需加上前缀“＃”。

如：指令 MOV DPTR，＃1234H

**2．直接寻址（内部数据存储器低128字节/特殊功能寄存器）**

在指令中直接给出操作数的地址

例：指令 MOV A，3AH

**直接寻址是访问特殊功能寄存器的唯一方法**

**3.寄存器内容是操作数**例如：指令 INC R0

**4．寄存器间接寻址**

**（内@SP仅PUSH POP/外@DPTR部数据存储器@R0@R1）**

寄存器内容是操作数地址。示形式：＠寄存器符号

例如：指令ANL A，@R1

**5. 相对寻址（程序存储器PC+偏移量）**

指令给出的操作数为程序转移的偏移量。在相对转移指令中，给出地址偏移量。目的地址＝转移指令所在地址＋转移指令字节数＋rel。如：指令JC 80H

**6．变址寻址（程序存储器@A+PC、@A+DPTR）**

以DPTR或PC为基址寄存器，累加器A为变址寄存器，以两者内容相加后形成的16位程序存储器地址作为操作数地址。称基址寄存器＋变址寄存器间接寻址。例如： MOVC A，＠A＋DPTR

**7．位寻址（内部数据存储器中128个可寻址位、特殊功能存储器可寻址位）** 寻址范围：

1）片内RAM位寻址区： 如 MOV C，2BH

●**数据传送类指令**

**指令助记符：**MOV、MOVX、MOVC、XCH、XCHD、SWAP、PUSH、POP

1）A内容送外部数据存储器或I/O

MOVX @Ri，A

MOVX @DPTR，A

2）存储器数据传送指令(或查表指令)

MOVC A，＠A＋PC

MOVC A，@A+DPTR

3）交换指令XCH组

XCH A，Rn； (A) (Rn) 、direct、@Ri

将累加器A与源操作数的字节内容互换。

例：设（R0）＝30H，（A）＝3FH，片内（30H）＝BBH。

执行指令 XCH A，@R0

**4）节交换指令组**

⑴ XCHD A，@Ri

Ri间接寻址单元的低4位与累加器A的低4位互换，而高4位不变。执行指令：XCHD A，＠R0

**⑵ SWAP A**

将累加器A的高、低半字节交换。XCHD和SWA

P主要用于十六进制数或BCD码的数位交换。

**5）操作指令组**

PUSH direct/POP direct

入栈指令： （SP） （SP）＋1，（SP） （direct）

出栈指令： （direct） （SP）， （SP） （SP-1）

●**算术运算类指令**

助记符8种：ADD、ADDC、INC、DA、SUBB、DEC、MUL、DIV

●**逻辑运算类指令**

助记符9种：ANL、ORL、XRL、RL、RLC、RR、RRC、CPL、CLR。

逻辑“与”助记符为**ANL**，用符号“∧”表示：

逻辑“或”助记符为**ORL**，用符号“∨”表示：

逻辑“异或”助记符为**XRL**，用符号“⊕”表示：

循环右移指令：**RR** A（累加器内容逐位循环右移一位，a0移到a7，不影响标志位）

带进位循环右移指令：**RRC** A（将累加器内容和进位位一起循环右移，a0移入CY，CY移到a7）

循环左移指令：**RL** A（累加器的内容逐位循环左移一位，a7移到a0。不影响标志位）

带进位循环左移指令：**RLC** A（累加器的内容和进位位一起循环左移一位，a7移入进位位**CY**，CY的内容移到a0）

累加器按位取反指令：**CPL** A（累加器的内容逐位取反，结果仍存在A中。不影响标志位）

累加器清0指令：**CLR** A（对累加器进行清0，不影响标志位）

●**控制程序转移类指令**

助记符12种：AJMP、LJMP、SJMP、JZ、JNZ、CJNE、DJNZ、ACALL、LCALL、RET、RETI、NOP

短转移指令：**SJMP** rel （目标地址是由当前PC值和（8位带符号）相对地址rel相加）

绝对转移指令：**AJMP** addr11（目标地址由指令第1字节的高3位a10～a8和指令第2字节的a7～a0所组成。以11位地址取代当前PC低11位，形成新的PC值。）

长转移指令：**LJMP** addr16（目标地址由指令第2字节和第3字节组成。目标地址为64KB空间）

间接转移指令：**JMP** @A+DPTR（目标地址是累加器A中的8位无符号数与数据指针DPTR的内容相加）

累加器判零转移指令：**JZ** rel；（若(A)=0，则(PC)=(PC+2)+rel；若(A)≠0，则(PC)=(PC)+2）

**JNZ** rel；（若(A)≠0，则(PC)= (PC+2)+rel；若(A)=0，则(PC)=(PC)+2）

数值比较转移指令：**CJNE** A，direct，rel（指令格式为： CJNE（操作数1），（操作数2），rel

数值比较指令的第1字节为操作码(或操作码+操作数1) ，第2字节为操作数2，第3字节为偏移量rel。具有比较转移和数值大小比较的功能。）

循环转移指令：**DJNZ** Rn，rel

DJNZ direct，rel

每执行一次本指令，先将操作数减1，判别是否为0。

◇不为0，转向目标地址；

◇为0，则结束循环程序，程序往下执行。

绝对调用指令：**ACALL** addr11（无条件地调用首址为addr11处的子程序。操作不影响标志位）

长调用指令：**LCALL** addr16（无条件地调用首址为addr16处的子程序。操作不影响标志位）

子程序返回指令：**RET**（表示结束子程序，返回ACALL或LCALL的下一条指令(即断点地址)，继续往下执行）

中断返回指令：**RETI**（中断服务程序返回，从断点处继续执行，清除内部相应的中断状态寄存器。中断服务程序必须以RETI为结束指令）

**●●什么是中断？**程序执行过程中，允许外部或内部事件通过硬件打断程序的执行，使其转向为处理外部或内部事件的中断服务程序中去，完成中断服务程序后，CPU继续原来被打断的程序，这样的过程称为中断服务响应过程。

**●单片机的中断系统主要应该解决3个问题,即:‘**

**①**当单片机内部或外部有中断申请时,能及时响应中断,中止正在执行的任务,转去处理中断服务子程序。中断服务处理以后,能正确回到原来的断点处继续处理原先的任务。

**②**当有多个中断源同时申,请中断时,单片机应能首先响应优先级高的中断源,能实现中断优先级的控制。‘

**③**当低优先级中断源正在享用中断服务时,若这时优先级比它高的中断源也申请中断,则要求单片机能中止低优先级中断源的服务程序,转去执行更高优先级中断源的服务程序,实现中断嵌套,并能逐级正确返回原断点处,继续处理原先的任务。

●**中断源：能产生中断的外部和内部事件。**

**80C51有5个中断源：**

**◆** 两个外部中断源INT0和INT1◆ 三个内部中断源

**1. 外部中断**

**两种信号触发方式：**

◆电平有效方式：若引脚上采样到有效的低电平，则向CPU提出中断请求；◆跳变有效方式：若引脚上采样到有效负跳变，则向CPU提出中断请求

**⑴INT0：外部中断0。**当IT0＝0 时，低电平有效；当IT0=1时，下降沿有效。

**⑵INT1：外部中断1。**当IT1＝0 时，低电平有效；当IT1=1 时，下降沿有效。

**⒉ 定时中断**

当计数器发生计数溢出时，表明设定的定时时间到或计数值已满，这时可以向CPU申请中断。

80C51有两个源，即：

⑴ TF0：T0溢出中断。

⑵ TF1：T1溢出中断。

**⒊ 串行中断**

每当串行口发送或接收一帧串行数据时，就产生一个中断请求。 RXD，TXD：串行中断。

●**中断矢量：**当CPU响应中断时，由硬件产生一个固定的地址，即矢量地址，由矢量地址指出每个中断源的中断服务程序的入口。

**中断允许寄存器IE。**EX01-外部。ET01-定计。ES-串行口。EA-总。**定时器／计数器控制寄存器** TCON：IT0(TCON.0)：INT0 中断申请触发方式控制位；IT1(TCON.2)：INT1 中断申请触发方式控制位；IF0(TCON.5)： T0计数溢出，硬件置位，软件清 0；IF1(TCON.7)： T1 计数溢出，硬件置位，软件清 0 ；IE0(TCON.1)：IE0=1 时，INT0 向 COU 申请中断；IE1(TCON.3)：IE1=1 时，INT1 向 COU 申请中断。

●**什么是单片机的中断优先级?中断优先级处理的原则是什么?**

在一个单片机系统中往往允许有多个中断源,通常给每个中断源规定了优先级别,称为“优先权"或“中断优先级’’。当单片机同时接收到两个或多个不同优先级的中断请求时,先响应高优先级的中断。如果同时接收到的是几个同一优先级的中断请求,则由内部的硬件查询序列确定它们的优先服务次序,当服务结束后,再响应级别较低的中断源。

●**外部中断请求有哪两种触发方式?对跳变触发和电平触发信号有什么要求?如何选择和设置?**

外部中断请求有两种信号触发方式,即电平有效方式和跳变有效方式,可通过设置有关控制位进行定义。当设定为信号电平有效方式时,若INT0或INT1引脚上采样到有效的低电平,则提出中断请求;当设定为信号跳变有效方式时,若INT0或INT1引脚上采样到有效负跳变,则提出中断请求。

●**中断响应时间：**在单级中断系统中，中断响应时间：3→8个机器周期。

中断请求标志位查询占1个机器周期，而且是指令的最后一个机器周期，在这个机器周期结束后，CPU即响应中断，产生硬件长调用LCALL指令，执行这条长调用指令需要2个机器周期，中断响应时间为3个机器周期。

●**80C51单片机在什么情况下可响应中断?**

中断响应是有条件的,即:

·中断源申请中断;

·该中断源已被允许中断,且CPU也已允许中断;

·没有同级或高优先级中断在执行中断服务程序。

在接受中断申请时,如遇下列情况之一,硬件生成的长调用指令LCALL将被封锁:

**①**CPU正在执行同级或高一级的中断服务程序。因为当一个中断被响应时,其对应的中断优先级触发器被置1,封锁了同级和低级中断。

**②**查询中断请求的机器周期不是执行当前指令的最后一个周期。目的在于使当前指令执行完毕后,才能进行中断响应,以确保当前指令的完整执行。

**③**当前正在执行RETI指令或执行对IE、IP的读/写操作指令。80C51.中断系统的特性规定,在执行完这些指令之后,必须再继续执行一条指令,然后才能响应中断。

●**外部中断源的扩展：**80C51只有两个外部中断请求输入端INT0和INT1，可以通过两种方式进行扩展：

**1)** 可通过增加“OC门”+ 软件来扩展**；2)** 定时器/计数器

●**定时器／计数器的核心：**加 1计数器（计数∶是对 T0、T1 引脚上输入的负跳变（**高电平向低电平跳变**）进行计数增l**）方式寄存器:**TMOD，高四位t1:GATE,门控。C/T功能选择。M1M0方式选择。

●**80C51单片机内部设有几个定时器/计数器?简述各种工作方式的特点。**

80C51单片机内部设有2个16位定时器/计数器TO和T1。定时器/计数器有4种工作方式,其特点如下:.

**①方式O**是13位定时器/计数器。由THx高8位(作计数器)和TLx的低5位(32分频的定标器)构成, TLx的低5位 溢出时,向THx进位;THx溢出时,硬件置位TFx(可用于软件查询),并可以申请定时器中断。

**②方式1**是16位定时器/计数器。TLx的低8位溢出时向THx进位,THx溢出时,硬

件置位TFx(可用于软件查询),并可以申请定时器中断。

**③方式2**是定时常数自动重装载的8位定时器/计数器。TLx作为8位计数寄存器,

THx作为8位计数常数寄存器。当TLx计数溢出时,一方面将TFx置位,并申请中断;另一方面将THx的内容自动重新装入TLx中,继续计数。由于重新装入不影响THx的内容,所以可以多次连续再装入。方式2对定时控制特别有用。.

**④方式3**只适用于TO,T0被拆成两个独立的8位计数器TLO和TH0。TLO做8位计

数器,它占用了T0的GATE、INTO、启动/停止控制位TRO、TO引脚(P3.4)以及计数溢出标志位TF0和TO

**T0方式3时，T1 仅能作为波特率发生器或不用中断的地方**

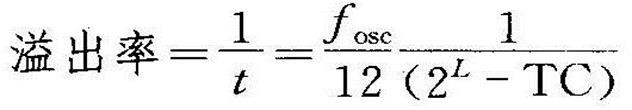
**T0 及 T1：**方式 0 L=13 213＝8 192，

方式 1 L＝16 216＝65 536，

方式 2 L＝8 28＝256

TC：定时器／计数器初值，即定时常数或计数常数。

●**定时时间的倒数即为溢出率，即：**



根据要求的定时时间 t、设定的定时器工作方式( 确定L) 及晶体振荡频率fosc，可计算出 TC 值(十进制数，TC=2^L-fosc✖t/12)，再将其转换成二进制数 TCB，然后再分别送入THi、TLi。

**对于 T0、T1：方式 0 时：**TCB=TCH＋TCL，TCH： 高 8 位，TCL：低 5 位、

MOV THi ， ＃ TCH ；送高 8 位

MOV TLi，＃TCL ；送低 5 位(高 3 位为0)

**方式 1 时**：TCB=TCH＋TCL，TCH ： 高 8 位，TCL： 低 8 位

MOV THi，＃TCH ；送高 8 位、

MOV TLi，＃TCL ；送低 8 位。

**方式 2 时**：TCB — 8 位重装载

MOV THi，#TCB ；送高 8 位、

MOV TLi，＃TCB；送低 8 位。

**8**0C51 串行口控制：串行口状态控制寄存器SCON、电源控制及波特率选择寄存器 PCON、串行数据寄存器。 SCON：TI(SCON.1)： 串行口发送中断源； RI(SCON.0)： 串行口接受中断源。

●**定时器/计数器做定时器使用时,定时时间与哪些因素有关?定时器/计数器做计数器使用时,外界输入计数频率最高为多少?**

定时器/计数器做定时器用时,定时器的定时时间与系统的振荡频率^sc,计数器的长度(如8位、13位或16位等)和定时初始值等有关。定时器/计数器做计数器用时,通过引脚T0(P3.4)和Tl(P3.5)对外部信号进行计数,由于检测一个1到O的跳变需要两个机器周期,故计数脉冲频率不能高于振荡脉冲频率的1/24。

●**如何实现通过定时器/计数器的计数功能达到扩大外部中断源的目的?**

可以利用定时器/计数器T0或T1的外部事件输入引脚TO、T1作为边沿触发的外部中断源。这时应设置定时器/计数器为计数器方式,而计数常数为满刻度值。外部输入的脉冲在负跳变时有效,计数器加1,由于计数常数已设为满刻度值,所以计数器加1后即溢出,向CPU申请中断。以定时器/计数器T0、T1的计数脉冲输入引脚TO、T1作为外部中断请求输入引脚,定时器/计数器T0、T1的中断矢量用做第三、笫四个扩展的外部中断矢量,定时器/计数器T0、T1的中断服务程序人口地址作为扩展的第三、笫四个外部中断服务人口地址,即实现了外部中断源的扩展。

●**计数器的应用**：当 TMOD 寄存器中 C/T 位设置为“1”时，作为计数器使用，可对来自单片微机引脚 T0 或 T1 上的负跳变脉冲进行计数，计数溢出时可申请中断，也可查询溢出标志位 TFx。运行中读定时器/计数器：80C51 可以随时读写计数寄存器 TLx 和 THx (x 为 0 或 1) ，用于实时显示计数值等。 办法是：先读 THx，后读 TLx，再重读 THx，若两次读得的 THx 值是一样的，则可以确定读入的数据是正确的；若两次读得的THx 值不一致，则必须重读。

SBUF 发送或接收一帧信息为 10 位：

(10 位 UART)1 位起始位(0)、8 位数据位(低位在前)和 l 位停止位

I2C 公用双总线结构：SDA 串行数据线双向、SCL 串行时钟线传输时钟信号

●**串行口的波特率发生器及波特率计算**

方式0时的波特率：振荡器的频率fosc/12

方式2时的波特率：图片%201.pngSMOD可以取0或1

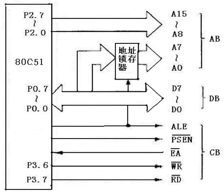
方式1和3时的波特率：由定时器T1和T2的溢出率和SMOD确定

图片%201.pngT1的溢出率与其工作方式相关

●**单片微机为什么需要 I/O 接口电路：**CPU 和外之间的数据传送：高速 CPU 与慢速外设的矛盾、外设的数据信号多种多样的、外设种类繁多、外设的数据传送有近距离的，也有远距离的。因此，必须在 CPU 与外设间通过接口电路进行协调。80C51 简单 I/O 的扩展：1、简单输出口的扩展：G = 0（锁存允许） CK 上跳变。 2、简单输入口的扩展D/A、A/D 转换器接口的扩展：8 位 D/A 转换器芯片 DAC0832：DAC0832 的单缓冲方式的接口电路和应用、DAC0832 的双缓冲方式的接口电路和应用●**I/O编址技术**：**独立编址**（优点是I/O地址空间和存储器地址空间相互独立，但需要专门设置一套I/O指令和控制信号，从而增加了系统的开销）、

**统一编址**（把系统中的I/O和存储器统一进行编址，注：接口中的寄存器（端口）与存储器中的存储单元同等对待。80C51使用统一编址方式）

●**I/O数传送控制方式**：无条件传送、程序查询、程序中断

●**系统扩展原理：单片微机是通过芯片引脚进行系统扩展，**并行：利用三组总线结构(AB、DB、CB) 

**（1）地址总线（AB）**P0口提供低8位地址A0～A7， P2口提供高8位地址A8～A15**（2）数据总线（DB）**P0口提供：D0～D7。与外部交换的所有数据、指令、信息，除少数可直接通过P1口外，全部通过P0口传送，同一时间只能有一个是有效的数据传送通道，由地址线控制各个芯片的片选线来选择**（3）控制总线（CB）**片外系统扩展用控制线；片外信号对单片微机的控制线

● ALE● ：读片外程序存储器中的数据

● ：选择片内或片外程序存储器

0：只访问外部程序存储器1：先访问内部程序存储器

● 、：片外数据存储器和I/O端口的写、读

●**并行扩展方式的编址技术**

片选：找到所在的芯片。 字选：找到该片的存储单元或I/O端口

(**1) 线选法：**利用最高几位空余的地址线中一根(如P2.7) 作为某一片存储器芯片或I/O接口芯片的“片选”。**(2) 译码法**：用译码器对空余的高位地址线进行译码，而译码器的输出作为“片选” 控制线。**⑶ 可编程阵列逻辑器件**：PAL、GAL **⑷ 输入／输出线作为译码线：**利用空余的I/O口线(如P1.0) ，或系统扩展的I/O口线作为芯片的片选线

●**程序存储器的扩展：**

**1. 片外并行数据存储器扩展**时的总线功能和读、写操作时序，片外数据存储器读、写操作的指令：

MOVX A，＠Ri ； 片外RAM→（A）， 读（RD）

MOVX ＠Ri，A ；（A）→片外RAM ， 写（WD）

MOVX A，＠DPTR；片外RAM→（A）， 读（RD）

MOVX ＠DPTR，A；（A）→片外RAM， 写（WD）

●使用8KBX8位的RAM芯片，用译码法扩展 64KBX8位的外部数据存储器，需要(8)片存储芯片，共需使用 (16)条地址线，其中 (13)条用千存储单元选择，( 3)条用千芯片选择 。

●三态缓冲器的三态分别是（低电平）、（高电平）和（ 高阻抗）

●80C51单片机系统整个存储空间由4部分组成，分别为(256)个地址单元的内部（数据）存储器,(4kb)个地址单元的内部（程序）存储器,(64kb)个地址单元的外部（数据）存储器，(60kb)个地址单元的外部（程序 ）存储器.

●在80C51单片机系统中,为外扩展存储器准备了( 16)条地址线,其中低位地址线由(PO口)提供,高位地址线由(P2口)提供。

在80C51 单片机系统中,存储器并行外扩展涉及的控制信号有(ALE)、 (\YJR)、(RD)、(PSEN)和(CE), 其中用于分离低 8位地址和数据的控制信号是(A LE) , 它的频率是晶振频率的(6 )分之一 。

●起止地址为OOOOH~3FFF H的外扩展存储器芯片的容量是 (1 6 KB)。 若外扩展存储器芯片的容量为2KB, 起始地址3000H, 则终止地址应为(37FFH)。

●与微型机相比， 单片机必须具有足够容量的程序存储器是因为它没有（外存）。

●在存储器扩展中，无论是线选法还是译码法，最终都是为扩展芯片的（片选）引脚端提供信号。有一片80C51和一片2716组成的单片机最小系统。若 2716 片选信号 CE 接地，则该存储芯片连接共需(11)条地址线。 除数据线外， 系统中连接的信号线只有(PSEN)和(ALE)。

●**单片微机为什么需要I/O接口电路**

CPU和外设之间的数据传送：

**①** 高速CPU与慢速外设的矛盾

**②** 外设的数据信号多种多样的

**③** 外设种类繁多

**④** 外设的数据传送有近距离的，也有远距离的

**在数据的I/O传送中，接口电路主要功能：**

1速度协调 ② 三态缓冲 ③ 数据转换

●●●**嵌入式设计及应用**●●●

●**什么是嵌入式系统？其特点是什么？**嵌入式系统是以应用为中心，以计算机技术为基础，软硬件可裁剪， 适应应用系统对功能、可靠性、成本、体积和功耗等严格要求的专用计算机系统。IEEE:嵌入式系统是用于控制、监视或辅助操作机器和设备的装置。特点：小型化与有限资源、与应用密切相关、系统软、硬件协同设计、需要交叉开发环境和调试工具

●●●**嵌入式系统的特点：**

1、小型化与有限资源 2、与应用密切相关（专用性）

3、系统软硬件协同设计（一体化）4、需要交叉开发环境调试工具

●●**嵌入式系统的组成：**嵌入式计算机系统是整个嵌入式系统的核心，由硬件层、中间层、系统软件层和应用软件层组成。

●**嵌入式最小系统有哪些组成部分，为什么要研究嵌入式最小系统？**处理器、内存、时钟、调试接口、电源和复位；构建嵌入式系统的第一步，保证嵌入式处理器可以运行；实际嵌入式系统的基础

●**RISC的特点1、**指令集：减少了指令种类，每条指令的长度是固定的，指令格式和寻址模式相当少，一个周期就可以执行一条指令；2、流水线：指令的处理过程被拆分成几个更小的。能够被流水线并行执行的单元，允许流水线在当前指令译码器阶段去取其下一条指令；3、寄存器：RISC拥有更多的通用寄存器，每个寄存器都可以存放数据或地址，寄存器可为所有的数据操作提供快速的局部储存访问；4、Load-Store结构：处理器只处理寄存器中的数据。独立的load和store指令用来完成数据在寄存器和外部存储器之间的传送（5、充分利用VLSI芯片的面积；6、提高计算机运行速度；7、便于设计，可靠性高；8、有效支持高级语言程序）

●**RISC架构的优点：**

1.充分利用VLSI芯片的面积2.提高计算机运行速度：指令；通用寄存器；流水线3.便于设计、可靠性高4.有效支持高级语言程序

●**ARM内核RISC架构的特别技术：**

1.所有的指令可根据前面的执行结果决定是否被执行

2.可用加载/存储指令批量传输数据

3.可在一条数据处理指令中同时完成逻辑处理和移位处理

4.在循环处理中使用地址的自动增减来提高运行效率

●**ARM的流水线技术：**

指令的二级流水线结构：若取指和执行阶段时间上完全重叠，指令周期减半，速度提高1倍

ARM7:三级流水（预取，译码，执行）

ARM9:五级流水（预取，译码，执行，访存，写回）

ARM10:六级流水（预取，发射，译码，执行，访存，写回）

ARM11:八级流水（预取，预取，发射，译码，转换，执行，访存，写回）

●**ARM处理器运行模式：**（用户 系统 快中断 中断 管理 中止 未定义）

●**ARM处理器支持两种指令集**

**32位的ARM指令集：**处理32位数据时性能较高

**16位的Thumb指令集：**ARM指令集的功能子集。

**特点：**密度高，所占存储空间较小；可以使用32位的寻址空间、32位寄存器、32位位移与逻辑运算、32位的内存存取访问功能。

●**从编程角度看，ARM微处理器的工作状态有两种：**

**ARM状态，**此时处理器执行32位的字对齐的ARM指令；

**Thumb状态，**此时处理器执行16位的、半字对齐的Thumb指令。

可以相互调用，只要遵循一定的调用规则

●**Thumb指令与ARM指令的时间效率和空间效率关系为:**

存储空间约为ARM代码的60％～70％

存储器为32位时ARM代码比Thumb代码快约40％

存储器为16位时Thumb比ARM代码快约40～50％

使用Thumb代码，存储器的功耗会降低约30％

●**状态切换方法**

ARM指令集和Thumb指令集均有切换处理器状态的指令，并可在两种工作状态之间切换，切换不影响处理器的运行模式和寄存器内容

在开始加电启动时，应该处于ARM状态。

**通用寄存器（R0~R15）,分组寄存器（R8~R14）程序计数器R15（PC）**

●**状态寄存器R16:**

当前程序状态寄存器 CPSR (Current Program Status Register)，可以在任何工作模式下被访问；程序状态备份寄存器SPSR (Saved Program Status Register) ，只有在异常模式下，才能被访问；（各种异常模式专有的）当异常发生时，SPSR用于保存当前程序状态寄存器CPSR的状态，从异常退出时，用于恢复CPSR。

**SPSR功能包括：**

保存ALU中的当前操作信息

控制允许和禁止中断

设置处理器的运行模式**。**

●**ARM状态下寄存器的子集：Thumb状态下寄存器和ARM状态下的寄存器组的关系：**

R0~R7是相同的

CPSR和所有的SPSP是相同的

Thumb状态下的SP对应于ARM状态下的R13

Thumb状态下的LP对应于ARM状态下的R14

Thumb状态下的程序计数器对应于ARM状态下的R15

**ARM的存储管理：**（大端格式高字节存储在底地址中，小端格式相反）

ARM920T微处理器核采用哈佛结构，将数据和指令分开存储分别使用各自的寻址方式和MMU。

●**S3C2410A的DMA控制器**

**采用DMA方式进行数据传输的具体过程如下：**

1.外设向DMA控制器发出DMA请求。

2.DMA控制器向CPU发出总线请求信号

3.CPU执行完总线周期，向DMA控制器发出相应的回答信号；

4.CPU将控制总线、地址总线及数据总线交给DMA控制器控制；

5.DMA控制器向外部设备发出DMA请求回答信号；

6.在DMA控制器控制下进行DMA传送；

7.数据传送完，DMA控制器通过中断请求线发出中断信号。CPU接收到中断信号后，转入中断处理程序进行处理；

8.中断处理结束后，CPU返回断点继续执行，并获得总线控制权

●**ARM指令系统及编程技术**

**ARM指令的一般格式：**

<opcode>{<cond>}{S} <Rd>,<Rn>{,<operand2 >}

其中< >内的项是必须的， { }内的项是可选的。

opcode：指令助记符

cond：执行条件

S：是否影响CPSR寄存器的值

Rd：目标寄存器

Rn：第1个操作数的寄存器

operand2：第2个操作数

●**ARM指令的条件域<cond>**

**cond CPSR中标志位 含义**

EQ Z置位 相等

NE Z清零 不想等

CS C置位 无符号数大于或等于

CC C清零 无符号数等于

MI N置位 负数

PL N清零 正数或零

VS V置位 溢出

VC V清零 未溢出

HI C置位Z清零 无符号数大于

LS C清零Z置位 无符号数小于或等于

GE N等于V 带符号数大于或等于

LT N不等于V 带符号数小于

GT Z清零且N等于V 带符号数大于

LE Z置位或N不等于V 带符号数小于或等于

AL 忽略 无条件执行

●**ARM寻址方式：**

**1.寄存器寻址2.立即数寻址**

**3.寄存器移位寻址**5种移位操作：

（1）LSL逻辑左移； （2）LSR逻辑右移

（3）ASR算术右移； （4）ROR循环右移；

（5）RRX带扩展的循环右移（标志位）。

例： MOV R0, R1, LSL #5; R0=R1逻辑左移5位

**4.寄存器间接寻址5.基址变址寻址6.多寄存器寻址**

一条指令实现一组寄存器值的传送，连续的寄存器用“-”连接，否则用“,”分隔

例：LDMIA R0, {R1-R5};

功能：R1=[R0], R2=[R0+4], R3=[R0+8]，R4=[R0+12], R5=[R0+16]

从以R0为起始地址的存储单元中取出5个字的数据送到R1到R5寄存器中。

**7.堆栈寻址 8.块拷贝寻址 9.相对寻址**

**●Load/Store指令**（用于寄存器和内存间数据传送，load把内存中数据装载到寄存器）

ldr <rd>， <adr>

LDR：后者内存的子数据装载到目标寄存器，同时还可以把合成的有效地址写回基址寄存器。

LDRB:同ldr，但只读取8位字节数据，并将目标寄存器高24位清零。

LDRBT:当做一般用户模式下的内存操作。

LDRH:读取16位，Rd高16位清零。

LDRSB:同ldrb，目标寄存器高24位改成该字节数据的符号位。

LDRT:同ldr，用户模式下。后者有效地址须对齐，否则从内存中读取的数值循环右移。

STR:把寄存器rd中子数据保存到adr所示内存。同ldr。

STRB:rd中低八位存到rd，其他同STR。

STRBT:同strb，但处于用户模式下。

STRH:rd中低16位保存到addr，addr所示地址半字对齐。

STRT:同str，处于用户模式下。

LDM:批量数据加载，从一片连续的内存单元读取数据到各个寄存器，起始地址为基址寄存器rn的值。IA传送后地址加一，IB传送前地址加一，DA传送后地址减一，DB传送前地址减一。

STM:将寄存器值传入一片连续内存地址。用法同LDM。FD传送后加一，ED 传送前加一，FA传送后减一，EA传送前减一。

SWP:R0,R1,[R2] r2的内存单元的子数据至r0，r1子数据至r2。

SWPB:（上式），r2内存单元的一个字节数据至r0低八位，r1低八位至r2所示内存单元。:

●**C 程序中内嵌汇编指令:**

ARM 的 C 程序中可用关键词\_ \_asm 加入一段汇编程序

C 语言中嵌入 ARM 汇编时需注意的几个问题:

汇编中不能再使用“;”作为注释行的开头，而应使用; C 语言中的“/\*\*/”或者“//”进行注释;不能使用一个物理寄存器去改变一个 C 变量;对于内嵌的汇编代码用到的寄存器，编译器在编译时;会自动加入保存和恢复这些寄存器的代码而不用用户去管理;不支持 ADR、ADRL 伪指令;不能用PC 寄存器返回当前指令的地址;不支持内存分配的伪操作

●**汇编、C 以及 C++的相互调用**:

C 程序调用汇编程序 :extern void strcopy(char \*d,const char \*s); 使用 extern 关键词表明 strcopy

函数是本文件之外定义的

汇编程序调用 C 程序://C 程序 g()返回 5 个整数的和；IMPORT g 使用伪操作 IMPORT 声明 C 程序 g();BL g 调用 C 程序 g()

C++程序调用 C 程序:extern “C” void cfunc(s\*) 使

用关键字 extern 声明被调用的 C 程序；

汇编程序调用 C ++程序：//被汇编程序调用的 C++

程序；extern “C” void cppfunc(s \* p) 被调用的 C++ 程序使用关键词 extern；BL cppfunc 调用 C++程序。

●**嵌入式系统来发的基本模式**:（传统模式）1需求分析2.软硬件分别设计开发调试测试3.系统集成，集成测试4.系统正确结束5出现错误，对软硬件分别验证修改6返回3。

**弱点**:虽然考虑了软硬件接口问题 但由于分开开发，各自部分的修改和缺陷容易导致系统集成出现错误，且难以定位，即使定位错误，对他们的修改往往也会涉及整个软件结构或硬件配置的改动。

**协同开发模式**:1.需求分析2.软硬件协同设计3.软硬件实现4软硬件协同测试和验证。

●**定时器 中断程序**

**这时T0采用方式3工作，其中，TL0产生100μs定时，由 P1．0输出方波1；TH0产生200μs定时，由Pl．1输出方波2；T1设置为方式2，作波特率发生器用。fOCC＝9．216 MHz**。

ORG 0000H

AJMP MAIN

ORG 000BH ；TL0的中断入口

AJMP ITL0

ORG 00lBH；TH0的中断入口

AJMP ITH0

ORG 0100H

MAIN： MOV SP，＃ 60H；设栈指针

MOV TMOD，＃ 23H 设T0为方式3,TI为2

MOV TL0，＃0B3H设TL0初值(100μs定时

MOV TH0，＃66H设TH0初值(200μs定时)

MOV TL1，＃0F6H设TL1初值(波特率为2400)

MOV TH1，＃ 0F6H 设 TH1初值

SETB TR0 ；启动 TL0

SETB TR1 ；启动TH0

SETB ET0 ；允许TL0中断

SETB ET1 ；允许TH0中断

SETB EA ；CPU中断开放

AJMP ＄

ORG 0200H

ITL0：MOV TL0，＃ 0B3H；重装定时常数

CPL P1．0；输出方波1(200μs)

RETI

ITH0：MOV TH0，＃66H；重装定时常数

CPL P1．1；输出方波2(400μs)

RETI

例. 飞读

RDTIME： MOV A，TH0；读TH0

MOV R0，TL0 ；读TL0并存入R0

CJNE A，TH0，RDTIME ；再读TH0

MOV R1，A ；存TH0在R1中

RET

●**并行输出 流水灯程序**

MOV SCON，#00H ；设串行口为方式0

CLR ES ；禁止串行口中断

MOV A，#80H；先显示最左边发光二极管

LED: MOV SBUF，A ；串行输出

JNB TI，$ ；输出等待

CLR TI ；软件清中断标志

ACALL DELAY ；轮显间隔

RR A ；发光右移

AJMP LED ；循环

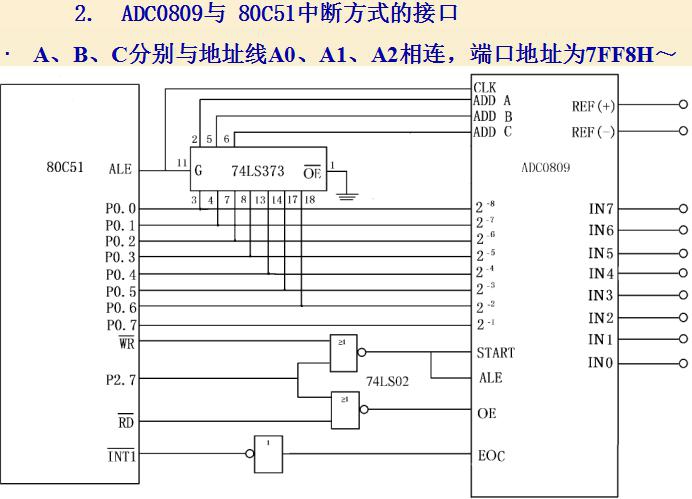
DELAY：MOV R6,#200H

NOP

DJNZ R6,& ；延时子程序

RET

●**将2000H单元开始的一批数据传送到从3000H开始的单元中，数据长度在内部RAM的30H中。**

MOV DPTR，#2000H；源数据区首址

PUSH DPL ；源数据区首址压栈保护

PUSH DPH

MOV DPTR，#3000H；目的数据区首址

MOV R6，DPL ；目的数据区首址存入寄存器

MOV R7，DPH

LP： POP DPH；取数据区地址指针

POP DPL

MOVX A，@DPTR ；取源数

INC DPTR

PUSH DPL

PUSH DPH

MOV DPL，R6 ；取目的数据区地址指针

MOV DPH，R7

MOVX @DPTR，A ；存入目的数据区

INC DPTR

MOV R6，DPL

MOV R7，DPH

DJNZ 30H，LP ；若数据块未移完，则继续

POP DPH

POP DPL

SJMP ＄

●**单片机与8255A的接口电路，PA口作输出口，接8个LED发光二极管，PB作输出口，接8个按键开关，PC口不用，都工作在方式0，要实现“按下任意键，对应的LED发光”，相应的程序如下：**

读PB口开关状态，送PA口输出控制LED，循环

MOV DPTR,#0FF7FH;指向8255A的控制口

MOV A,#82H; 工作方式控制字

MOVX @DPTR,A;向控制口写控制字，PA输出，PB输入

LOOP:MOV DPTR,#0FF7DH;指向8255A的PB口

MOVX A,@DPTR;读PB口按键状态

MOV DPTR,#0FF7CH;指向8255A的PA口

MOVX @DPTR,A;从PA口输出，驱动LED发光

SJMP LOOP

●**试编写串行接口以工作方式2发送数据的中断** ORG 0023H

AJMP SPINT

 SPINT：CLR EA ；关中断

PUSH PSW ；保护现场

PUSH ACC

SETB EA ；开中断

SETB PSW.4 ；切换寄存器工作组

CLR TI ；清除发送中断请求标志

MOV A，＠R0 ；取数据，置奇偶标志位

MOV C，P ；奇偶标志位P送TB8

MOV TB8，C

MOV SBUF，A ；数据写入发送缓冲器，启动发送·

CLR 0AFH ；恢复现场

POP ACC

POP PSW

SETB 0AFH

CLR PSW .4 ；切换寄存器工作组

RETI

●**采用延时等待A/D转换结束方式，分别对8路模拟信号轮流采样一次，并依次把结果存入数据存储器。**

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN：MOV R1，#20H

MOV DPTR，#7FF8H；指向通道0地址

MOV R7，#08H ；共需转换8个通道

LOOP：MOVX @DPTR，A；启动A/D转换①

LCALL D128μs ；延时等待A/D转换结束②

MOVX A，@DPTR 读入A/D转换值③

MOV @R1，A

INC DPTR ；指向下一通道地址

INC R1

DJNZ R7，LOOP ；8个通道未转换完则继续

D128μs： …；延时128μs 子程序

RET

●**中断方式AD采样 采集8路模拟量，并存入20H地址开始的内部RAM中**

ORG 0000H

SJMP MAIN

ORG 0003H ；外部中断0入口地址

LJMP INTDATA

ORG 0100H ；数据采集程序

MAIN：MOV R0，＃20H ；数据缓冲区首址

MOV R2，＃8 ；8通道计数器

MOV DPTR，＃7FF8H ；指向0通道

START：CLR F0 ；清中断发生标志

MOVX ＠DPTR，A ；启动A/D (P2.7=0, /WR=0) ①

SETB IT0 ；置外部中断0为边沿触发

SETB EX0 ；允许外部中断0

SETB EA ；开中断

LOOP：JNB F0，LOOP ；中断发生标志是否为0 ②

DJNZ R2，START ；8个通道转换是否结束

SJMP MAIN

INTDATA：MOVX A，＠DPTR ；读数据 (P2.7=0, RD=0)，；硬件撤销中断③

MOV ＠R0，A ；存数据

INC R0

INC DPTR ；指向下一通道

SETB F0 ；置中断发生标志

RETI

# 7、完成某种操作可以采用几条指令构成的指令 序列实现，试写出完成以下每种操作的指令序列。

（1）将R0 的内容传送到R1；（2）内部 RAM 单元 60H 的内容传送到寄存器R2；（3）外部 RAM 单元

1000H 的内容传送到内部RAM 单元 60H ；（4）外部 RAM 单元 1000H 的内容传送到寄存器 R2 ；

（5）外部 RAM 单元 1000H 的内容传送到外部RAM单元 2000H 。

1MOV A,R0

MOV R1,A

2MOV R2,60H

3MOV DPTR,#1000H MOVX A,@DPTR MOV 60H,A

4MOV DPTR,#1000H MOVX A,@DPTR MOV R2,A

5MOV DPTR,#1000H MOVX A,@DPTR MOV DPTR,#2000H MOVX @DPTR, A

# 若（R1）=30H，（A）=40H，（30H）=60H，（40H）

**=08H。试分析执行下列程序段后上述各单元内容的变化。**

MOV A ，@R1 （R1）=7FH

MOV @R1 ，40H （A）=60H

MOV 40H ，A （30H）=08H

MOV R1 ，#7FH （40H）=60H

# ●若（A）=E8H，（R0）=40H，（R1）=20H，（R4）=3AH，（40H）=2CH，（20H）=0FH，试写出下列各指令独立执行后有关寄存器和存储单元的内容？若该指令影响标志位，试指出 CY、AC、和 OV 的值。

（1）MOV A，@R0 （A）=2CH

（2）ANL 40H ，#0FH （40H）=0CH

（3）ADD A ，R4 （A）=22H，（CY）=1，（AC）=1，（OV）=0

（4）SWAP A （A）=8EH

（5） DEC @R1 （20H）=0EH，P=1

（6）XCHD A，@R1 （A）=EFH，（20）=08H

# ●若（50H）=40H，写出执行以下程序段后累加器 A、寄存器 R0 及内部 RAM 的 40H、41H、42H 单元中的内容各为多少？

MOV A，50H

MOV R0，A

MOV A，#00H

MOV @R0，A

MOV A，#3BH

MOV 41H，A

MOV 42H，41H

答：（A）=3BH，（R0）=40H，（40H）=00H，（41H ）

=3BH，（42H）=3BH。

**●试分析以下程序段的执行结果。**

**结果如下:**

MOV SP,#60tt ;(SP)=60H

NOV A,#88tt ;(A)=88H

MOV B.#0FFH ;(B)=FFH

PUSH ACC ;(SP)=61H,(61H)=88H

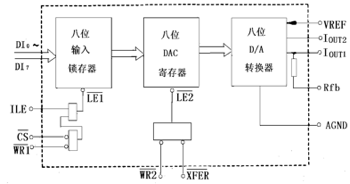
PUSH B ;(SP)=62H,(62H)=FFH

POP ACC ;(A)=FFH,(SP)=61H

POP B ;(B)=88H,(SP)=60H

程序段的执行结果:累加器A和寄存器B的内容通过堆栈进行了交换。

注意:80C51单片机的堆栈是按照先进后出的原则进行管理的。

**DAC 0832逻辑结构图**

●**已知(A)=7AH,(RO)=30H,(30H)=A5 H,(PSW)** **=80H。请填写各条指令单独执行后的结果。**

结果如下:

(1) XCH A,R0 ;(A)=30H,(R0):7AH

(2) XCH A,30H ;(A)=A5H,(30H)=7AH,(PSW)=81H

(3) XCH A,@R0 ;(A)=R5H,(30H)=7AH,(PSW)=81H

(4) XCHD A,@R0 ;(A)=75H,(30H)=AAH,(PSW)=81H

(5) SWAP A ;(A)=A7H

(6) ADD A,R0 ;(A)=AAH,(PSW)=04H

(7) ADD A,30H ;(A)=1FH,(PSW)=81H

(8) ADD A,#30H ;(A)=AAH,(PSW)=04H

(9) ADDC A,30H ;(A)=20H,(PSW)=01H

(10) SUBB A,30H ;(A)=D4H,(PSW)=84H

(11) SUBB A,#30H ;(A)=49H,(PSW)=01H

●**例：使用74LS164的并行输出端接8只发光二极管，利用它的串入并出功能，把发光二极管从左向右依次点亮，并不断循环**

MOV SCON，#00H ；设串行口为方式0

CLR ES ；禁止串行口中断

MOV A，#80H ；先显示最左边发光二极管

LED: MOV SBUF，A ；串行输出

JNB TI，$ ；输出等待

CLR TI ；软件清中断标志

ACALL DELAY ；轮显间隔

RR A ；发光右移

AJMP LED ；循环

DELAY：…… ；延时子程序

RET

●**例：通过串行口发送带奇偶校验位的数据块ASCII码由7位组成，其最高位可作为奇偶校验位用。数据块通过串行口发送和接收，采用8位异步通信，波特率为1200，fosc=11.0592MHz。从内部RAM单元20H～3FH中取出ASCII码加上奇偶校验位之后发出。设串行口为方式1，T1为方式2作为串行口的波特率发生器**。

∵SMOD＝0 ∴TH1＝232＝E8H

ORG 0000H

MOV TMOD，#20H ；设T1为方式2

MOV TL1，#0E8H ；T1定时常数

MOV TH1，#0E8H

MOV SCON，#01000000B ；设串行口为方式1

MOV R0，#20H ；设发送数据区首址

MOV R7，#32 ；发送32个ASCII码数据

LOOP：MOV A，@R0 ；取ASCII码数据

ACALLSP-OUT ；调用串行口发送子程序

INC R0 ；未发送完，则继续

DJNZ R7，LOOP

…

串行口发送子程序

SP-OUT：MOV C，P ；设奇校验位

CPL C

MOV ACC.7，C

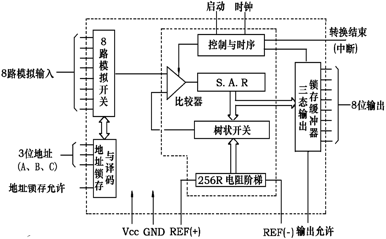
MOV SBUF，A ；带校验位发送

JNB TI，$ ；发送等待

CLR TI

RET

**ADC0809的内部逻辑结构**



●**例： 试编写串行接口以工作方式2发送数据的中断服务程序**

ORG 0023H

AJMP SPINT

SPINT：CLR EA ；关中断

PUSH PSW ；保护现场

PUSH ACC

SETB EA ；开中断

SETB PSW.4 ；切换寄存器工作组

CLR TI ；清除发送中断请求标志

MOV A，＠R0 ；取数据，置奇偶标志位

MOV C，P ；奇偶标志位P送TB8

MOV TB8，C

MOV SBUF，A ；数据写入发送缓冲器，启动发送

INC R0 ；数据地址指针加1

CLR 0AFH ；恢复现场

POP ACC

POP PSW

SETB 0AFH

CLR PSW .4 ；切换寄存器工作组

RETI ；中断返回

●**例：产生锯齿波由上图可知，输入寄存器和DAC寄存器同时受控锁存，DAC0832的地址为7FFFH(P2.7=0)**

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN： MOV DPTR，#7FFFH ；输入输入寄存器和DAC寄存器地址 MOV R0，#0 ；转换初值

LP： MOV A，R0

MOVX @DPTR，A ；送出模拟量

INC R0

SJMP LP

●**例：采用延时等待A/D转换结束方式，分别对8路模拟信号轮流采样一次，并依次把结果存入数据存储器。**

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN： MOV R1，#20H

MOV DPTR，#7FF8H ；指向通道0地址

MOV R7，#08H ；共需转换8个通道

LOOP： MOVX @DPTR，A ；启动A/D转换①

LCALL D128μs ；延时等待A/D转换结束②

MOVX A，@DPTR ；读入A/D转换值③

MOV @R1，A

INC DPTR ；指向下一通道地址

INC R1

DJNZ R7，LOOP ；8个通道未转换完则继续

D128μs： … ；延时128μs 子程序

RET

●**例 要求在Pl.O引脚上产生周期为2 ms 的方波输出。已知晶体振荡器的频率为fosc = 6MHz。可使用T0作定时器，设为方式0, 设定1ms的定时，每隔1ms使Pl.0引脚上的电平变反。**

解： (1)定时常数计算振荡器的频率fosc = 6MHz, 机器周期为2μs, 方式0计数器长度L =13(2^13=8192, 定时时间t=1ms=0.001s，定时常数：TC为7692 = lEOCH, 转换成二进制数TCB，取低13位，其中高8位TCH = FOH, 低5位为TCL =0CH

计数长度为1EOCH=76 92, 定时为(8192 — 7692) X 2μs = 0.001s

(2)编程

ORG 0000H

AJMP MAIN

ORG 000BH o/ ;T0中断矢量

AJMP INQP

ORG 0030 B

MAIN: MOV TMOD, # OOH ；写控制字，设T0为定时器

MOV TH0'# 0F0H ；写定时常数（定时lms）

MOV TL0, #0CH ，

SETB TR0 ；启动T0

SETB ET0 ；允许T0中断

SETB EA ；开放CPU中断

AJMP ＄ ；定时中断等待

ORG 2000 H ;TO中断服务程序

INQP:MOV TH0, #0F0H ；重写定时常数

MOV TL0,#0CH

CPL Pl.0 ; Pl.0变反输出

RETI ；中断返回

●**例：采用延时等待A/D 转换结束方式，分别对8 路模拟**

**信号轮流采样一次，并依次把结果存入数据存储器。**

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN：MOV R1，#20H

MOV DPTR，#7FF8H ；指向通道0 地址

MOV R7，#08H ；共需转换8 个通道

LOOP：MOVX @DPTR，A；启动A/D 转换①

LCALL D128μs ；延时等待A/D 转换结束②

MOVX A，@DPTR ；读入A/D 转换值③

MOV @R1，A

INC DPTR ；指向下一通道地址

INC R1

DJNZ R7，LOOP ；8 个通道未转换完，则继续。

D128μs： … ；延时128

μs 子程序

RET

●**例：采集8 路模拟量，并存入20H 地址开始的内部RAM中。**

ORG 0000H

SJMP MAIN

ORG 0003H ；外部中断0 入口地

址

LJMP INTDATA

ORG 0100H ；数据采集程序

MAIN：MOV R0，＃20H ；数据缓冲区首址

MOV R2，＃8 ；8 通道计数器

MOV DPTR，＃7FF8H ；指向0 通道

START：CLR F0 ；清中断发生标志

MOVX ＠DPTR，A ；启动A/D (P2.7=0, /WR=0)

①

SETB IT0 ；置外部中

断0 为边沿触发

SETB EX0 ；允许外部中断0

SETB EA ；开中断

LOOP：JNB F0，LOOP ；中断发生标志是否

为0 ②

DJNZ R2，START ；8 个通道转换是否结束

SJMP MAIN

INTDATA：MOVX A，＠DPTR ；读数据(P2.7=0,

RD=0)， 硬件撤销中断③

MOV ＠R0，A ；存数据

INC R0

INC DPTR ；指向下一通道

SETB F0 ；置中断发生标志

RETI

●**例：对8255A 各口作如下设置：A 口方式0，B 口方式0，**

**从A 口输入，从B 口、C 口输出。**

工作方式控制字为10010000，即90H。

初始化程序段：

MOV A，#90H ；设A 口、B 口为方式0 ；A 口输入，

B 口、C 口输出

MOV DPTR，#7FFFH

MOVX @DPTR，A

MOV DPTR，#7FFCH ；从A 口输入

MOVX A，@DPTR

MOV DPTR，#7FFDH ；从B 口输出

MOVX @DPTR，A

MOV DPTR，#7FFEH ；从C 口输出

MOVX @DPTR，A

例:把C 口的第5 位pc 端置为1

MOV DPTR,#7FFFH

MOV A,#00001011 ; PC5 置位

MOVX @DPTR,A

● **(1) int SubXY(int x, int y)**

**｛ return x-y;｝**

● **(2) void SubXY(int x, int y, int z)**

**｛ z = x-y;｝**

(1)AREA SubXY ,CODE,READONLY

EXPORT SubXV

SUB RO, RO, R1；结果通过RO返回

MOV PC, LR；返回

END

(2)AREA SubXY ,CODE,READONL V

EXPORT SubXY

SUB R2, RO, R1

LDR R3, =Z ; R3执行变量z

STA R2, [R3] ；更新变量z

MOV PC, LR ；返回

END