**1.什么是嵌入式系统？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

**2.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。

**3．简述80C51单片机指令系统的寻址方式。**立即数；直接；寄存器；寄存器间接；相对寻址；变址；位寻址

**4.8051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。

**5.中断服务子程序与普通子程序有何异同之处？**当中断产生的时候进入中断服务程序，不需要调用；而普通子程序只有被调用了才能执行。

**6.80C51的中断与子程序调用有什么异同点？**相同：都是中断当前正在执行的程序，转去执行子程序或中断服务程序；都是由硬件自动地把断点地址压入堆栈，然后通过软件完成现场保护；执行完子程序或中断服务程序后，都要通过软件完成现场恢复，并通过执行返回指令，重新返回到断点处，继续往下执行程序；两者都可以实现嵌套，如中断嵌套和子程序嵌套。不同：中断请求信号可以由外部设备发出，是随机的 ，子程序调用是软件编排好的；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；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。

**7.在MCS51中，试叙述INT0中断的响应过程。**CPU每个周期及其采样INT0引脚信号一次，当有中断请求时，响应中断，由硬件生成长调用指令LCALL 0003H；当前程序计数器PC压入堆栈进行保护；将对应的中断源的中断矢量地址装入PC，转向中断服务程序，直至RETI为止；撤销中断请求，弹出断点处地址至PC；恢复源程序的断点执行，恢复中断触发源状态。

**8.在ARM中，试叙述IRQ异常的响应过程。**保存当前状态，PC->LR中；CPSR存入SPSR-irq；强制M[4：0]为10010B进入IRQ模式；将CPSR中断标志I禁止，F不变；将T标志清0，处于ARM状态；强制PC的值为特定的值，去执行异常响应程序；执行异常响应返回指令，返回被中断的程序继续执行。

**9.ARM处理器的工作模式有哪几种？**用户模式usr;快速中断模式fiq；外部中断模式irq；操作系统保护模式svc或管理模式；数据访问中止模式abt;处理未定义指令的未定义模式und

**总线类型**：内部、外部、系统、非系统总线

**冯诺依曼结构：**程序存储器与数据存储器合并在一起；**哈弗结构**：程序存储与数据存储分开

嵌入式系统扩展的外围接口可以有很多的选择（根据被控对象不同而选择不同的接口电路）

嵌入式处理器一般除了CPU之外还可以包括很多外部设备，甚至包括内存

**最基本系统的条件：**以某个处理器为核心的嵌入式系统中，处理器可以运行程序

**嵌入式最小系统的作用：**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

**嵌入式处理器分为四大类型：**嵌入式微控制器MCU；嵌入式微处理器MPU；嵌入式DSP处理器DSP；嵌入式片上系统SOC

**IP:** 知识产权或称为SIP（硅/半导体知识产权 )—指芯片电路的制作方式或电路图等方面的知识产权。

**ARM微处理器系列：**ARM7，ARM9 **嵌入式微处理器：**硬件系统核心部件

**1.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

**2.如何进行嵌入式处理器的选型？**根据具体应用领域选择MPU/MCU/DSP/ SOC中的一种；根据具体应用功能需求、性能指标、运行环境和成本预算等选择芯片的具体型号和配置参数；关注生产厂商是否提供相应内置硬件调试工具和评估板；对于32位及以上嵌入式处理器，关注是否有合适的嵌入式操作系统支持

**嵌入式操作系统的种类：按实时性能分类：嵌入式实时操作系统**：**功能**：对多个外部事件，尤其是异步事件进行实时处理（性能特征：多任务运行方式； 基于优先级的抢占式调度； 快速的任务上下文切换：**实时操作系统的重要特征之一**； 较短的中断延迟时间：中断延迟时间是指中断发生到开始执行中断处理程序的时间和嵌入式实时操作系统的重要特征之一；确定性）；典型的嵌入式实时操作系统VxWorks；pSOSystem；Nucleus；μC/OS等；**嵌入式非实时操作系统：**典型代表是嵌入式分时操作系统：也是一种多任务操作系统；任务调度采用时间片轮转方式（大的任务吞吐量）；典型的嵌入式软实时操作系统Window CE；Palm；EPOC；Embedded Linux等

**按购买方式分类 ：商用型：特点：**功能稳定、可靠；提供完整方便的开发调试工具；配套完善的技术支持和售后服务； 但往往价格昂贵**； 免费型 ：特点：**在价格方面具有优势；一般缺乏完善的集成开发调试工具；稳定性与服务方面存在挑战

**按源码是否开放分类：开放源码型；黑箱型**

**VxWorks基本特征：**高实时性、高稳定性的**微内核**（微内核结构，最小8kB；高稳定性；实时性）

**PalmOS：1992年，手写识别；**一款具有划时代意义的产品，带来了**“PDA”**和**“手持”**这两个计算机术语

**Windows CE的特点：**基于优先级的多线程**抢占式**调度管理方式，但实时性能仍欠佳，属于软实时操作系统

**EPOC：**是一个稳定的**无线操作系统**

**Embedded Linux：**Embedded Linux源码开放、免费，**同时提供强大的功能；**抢占式内核Linux版本：直接修改Linux内核代码中的调度算法机制（修改成抢占式内核：**一定的实时性） Linux的内核安全可靠性高**

**ARM体系结构：**ARM内核非常小；器件的功耗非常低

**ARM7 T D M I：T**：支持高密度16位的Thumb指令集；**D**支持片上调试；**M**内嵌硬件乘法器(Multiplier)；**I** Embeded-ICE,支持片上断点和调试点; **ARM7 E J:E**支持增强型DSP指令集;**J**支持新的JAVA

**ARM核”并不是芯片:**ARM核＋RAM、ROM、片内外设等组合在一起才能构成现实的芯片

**RISC（Reduced Instruction Set Computer） CISC（Complex Instruction Set Computer）**

**RISC 的主要特征:**只有 **LOAD / STORE** 指令访存

**1.CISC与RISC的对比，RISC采用哪些特殊技术**

1.指令数量很多、较少，通常少于100；2.有些指令执行时间很长，如整块的存储器内容拷贝，或将多个寄存器的内容拷贝到存贮器、没有较长执行时间的指令；3.编码长度可变，1-15字节、编码长度固定，通常为4个字节；4.寻址方式多样、简单寻址；5.可以对存储器和寄存器进行算术和逻辑操作、只能对寄存器对行算术和逻辑操作，Load/Store体系结构；6.难以用优化编译器生成高效的目标代码程序 、**采用优化编译技术，生成高效的目标代码程序**

**存储格式（大端和小端格式的概念）大端：**字数据的高字节存储在低地址中；**小端：**字数据高字节存储在高地址中

**冯·诺依曼体系结构（ARM7）哈佛体系结构（ARM9以后）**

**处理器模式：特权模式**：系统、快中断、中断、管理、中止、未定义；**异常模式**：快中断、中断、管理、中止、未定义

**寄存器组：31个通用寄存器、6个状态寄存器（ARM状态下的寄存器组、Thumb状态下的寄存器组、**

**程序状态寄存器：**1个当前程序状态寄存器(CPSR)； 5个备份程序状态保存寄存器(SPSR,异常处理使用**)**

**备份的程序状态保存寄存器功能：** 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**）**

**异常与中断处理：**1.异常的响应过程：将下一条指令的地址存入相应连接寄存器LR；将CPSR复制到相应的SPSR中； 根据异常类型，强制设置CPSR的运行模式；强制PC从相关的异常向量地址取下一条指令执行。2.**异常的返回过程**：将连接寄存器LR的值减去相应的偏移量送到PC中；将SPSR复制到相应的CPSR中；若设置了中断禁止位，则清除

（**复位**异常处理程序不需要返回）

**JTAG调试接口：边界扫描技术：**通过边界扫描寄存器单元，实现对芯片输入输出信号的观察和控制；边界扫描寄存器对芯片是透明的，正常的运行不会受到任何影响；不占用ARM芯片上的任何硬件资源；比较方便和容易发现程序中存在的问题

**ARM指令的助记符格式：<opcode>：**操作码，决定了指令的具体操作；**{<cond>}**：决定指令执行的条件，为可选项； **{S}**：决定指令的执行是否影响CPSR寄存器的值，为可选项； **<Rd>**：是目的寄存器； **<Rn>**：是存放第一个操作数寄存器，当仅需要一个源操作数时可省略； **<op2>**：表示第二个操作数，可以是立即数，也可以是表示操作数的寄存器，为可选项。

**寄存器移位寻址：5种移位操作：**LSL逻辑左移；LSR逻辑右移；ASR算术右移；ROR循环右移；RRX带扩展的循环右移。

**SBC 带借位减法指令：**借位时清除进位标志，执行非操作，因此需要自动取反

**RSC 带借位的反向减法指令：**借位时清除进位标志，执行非操作，因此需要自动取反

**EOR 逻辑异或指令：如何实现，以8位字节为例请举例说明：**反转R0中的位0和1 ：EOR     R0, R0, #3  ； 将R1的低4位取反： EOR     R1,R1,#0x0F；R2=R1∧R0 ：EOR     R2,R1,R0；将R5和0x01进行逻辑异或，结果保存到R0，并根据执行结果设置标志位：EORS     R0，R5，＃0x01；

**TEQ 相等测试指令:** TEQ R0, #5; 判断R0的值是否和5相等（**如何判断？）:**将Rn的值与operand的值按位作逻辑“异或”操作，根据操作的结果更新CPSR中的相应条件标志位，当两个数据相等时：EQ有效，否则：NE有效

**跳转指令:** B 跳转指令; BL 带返回的跳转指令; BLX 带返回和状态切换的跳转指令; BX 带状态切换的跳转指令

**LDR 字数据加载指令:**把addr中的字数据加载到Rd中；同时将合成的有效地址写回到基址寄存器,如：LDR R0, [R1, R2,LSL#5]!;

**STR 字数据存储指令：**把Rd中的字数据保存到addr所表示的内存地址中；同时将合成的有效地址写回到基址寄存器：STR R0, [R1, #5]!;（**寻址方式同LDR**）

**LDM 批量数据加载指令：type字段：**指定了地址加还是减：IA ：每次传送后地址加1；IB ：每次传送前地址加1 ；DA：每次传送后地址减1；DB：每次传送前地址减1

**堆栈寻址:**FD：满递减堆栈; ED：空递减堆栈；FA：满递增堆栈；EA：空递增堆栈

**ARM汇编语言程序设计(格式)：**定义一个只读的代码段：AREA EXAMPLEI,CODE,READONLY;程序入口：ENTRY;源程序结束：END

**ENTRY**：语法格式：ENTRY伪操作用于指定汇编程序的入口点（在一个源文件里最多只能有一个ENTRY（可以没有）；在一个完整的汇编程序中至少要有一个ENTRY；当有多个ENTRY时，程序的真正入口点由链接器指定）

示例：**AREA example，CODE，READONLY**

**ENTRY ；应用程序的入口点**

**IMPORT**：语法格式：**IMPORT 标号{[WEAK]}**（使用的标号在其它的源文件中定义；但要在当前源文件中引用）

示例：**AREA Init，CODE，READONLY**

**IMPORT Main ….**

**END**

**ARM高级语言程序设计: C程序中内嵌汇编指令:**在ARM的C程序中可用**关键词\_ \_asm**加入一段汇编程序

**在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。

**嵌入式软件的交叉开发环境：**交叉开发环境是指用于嵌入式软件开发的所有工具软件的集合，一般包括**：**文本编辑器、交叉编译器、交叉调试器、仿真器、下载器等；**交叉开发环境由宿主机和目标机组成，宿主机与目标机之间在物理连接的基础上建立起逻辑连接。**

**物理连接**是逻辑连接的基础；**逻辑连接**指宿主机与目标机间按某种通信协议建立起来的通信连接，目前逐步形成了一些通信协议的标准。

**物理上的连接：**宿主机的PC机等硬件与目标机的嵌入式硬件的连接

HMOS**(高性能金属氧化物半导体,描述了集成电路中MOS管的结构,形成栅极，高精度)**制造工艺的MCS-51单片机都采用40引脚**双列直插（DIP）封装方式**。

**CHMOS(HMOS高精度、CMOS低功耗)**制造工艺的80C51/80C31芯片，除DIP封装外，还采用方形封装（PLCC）工艺。

1.电源：VCC、GND；2.时钟：XTAL1、XTAL2；3.外部ROM读选通：PSEN；外部ROM访问控制：EA/VPP；地址锁存：ALE/PROG；复位：RST；4. I/O口（既可输入也可输出）； P0: 地址、数据；P1: 用户使用；P2: 地址；P3: 兼具二功能

**低电平有效：**在对外部程序存贮器取指令时，PSEN每个机器周期产生两次有效，并将程序存储器的内容送往P0口（数据总线）

**EA/Vpp：访问外部程序存储器控制信号(Pin31)：**当EA为低电平时，CPU仅执行外部程序存贮器中的程序（**对于8031**，由于其内部无程序存贮器，**EA必须接地**，这样才能只选择外部程序存贮器）。

**ALE/PROG：地址锁存允许信号（Pin30）：**在**访问外部存贮器**时，ALE的输出用于锁存地址的低8位；在**非访问外部存贮器期间**，ALE以振 荡频率的1/6固定速率输出，能作外部时钟或定时用；在**访问外部数据存贮器期间**，则以1/12 fosc频率输出ALE脉冲。对于EPROM型单片机（8751），在EPROM编程期间，此引脚用于输入编程脉冲PROG。

**RST：**复位信号输入端/内部RAM备用电源输入端 (Pin9) 高电平有效。**连续2个机器周期的高电平复位该芯片。**

**输入/输出引脚（4个端口，共32条）：**当访问外部存贮器时，它是**复用总线**，作为数据总线D0～D7；作为地址总线的低八位（A0～A7）

**P2口（Pin21～28）：**8位准双向并行I/O口；负载能力：驱动4个LSTTL**。在访问外部存贮器时，作为地址总线高8位。P3口（Pin10～17）：**8位准双向并行I/O口。 负载能力：驱动4个LSTTL。**可提供以下特殊功能（也称第二功能）**

**80C51**包含**128字节的数据存贮器（RAM）**

**ALU**：对传送到CPU的数据进行算术/逻辑运算。 运行一条指令的时间：单周期，双周期或四周期**（机器周期）**

**程序状态字PSW**——8位,存放当前指令执行后的状态。（为下条或以后指令的执行提供状态条件。许多指令的执行结果将影响PSW的某些对应的状态标志位。）重要特点:**可编程**，即通过软件改变PSW的状态标志。（**注意奇偶标志位P的特殊性**）**P**：PSW.0 位地址D0，奇偶标志，**累加器ACC中**1的个数为奇数时则P=1，否则P=0（即：偶校验）

**堆栈指针SP——8位**：**堆栈**：一组特殊的存储单元，通常是在存储器中开辟的一个存储区域。堆栈元素的压入和弹出遵循**后进先出**的原则，堆栈操作总是对栈顶（向上）进行的。**压栈（PUSH）**时SP先自动加“1”，然后将欲进栈的数据压入SP所指示的栈顶单元；**出栈(POP)**时将所指示的栈顶单元内数据弹出，然后SP自动减1，因而SP总是指向堆栈栈顶。例：**设（SP）＝32H，片内RAM的30H~32H单元中的内容分别为20H，23H，01H，执行下列指令的结果怎样？POP DPH ；（（SP））＝（32H）＝01H→DPH、（SP）－1＝32H－1＝31H→SP， POP DPL；（（SP））＝（31H）＝23H→DPL、（SP）－1＝31H－1＝30H→SP**

**数据指针DPTR**——16位地址寄存器：用于寻址64K地址空间中任一地址单元，既可寻址数据存贮器，也可寻址程序存贮器的表格等单元。常用于调用（CALL）、转移以及访问外部数据存储器和查表等指令。可分成两个独立的8位数据指针：DPH,DPL。

**程序计数器PC**——16位（**无地址**）存放程序地址（CPU**即将**执行的指令的地址）； 控制程序顺序运行或跳转。PC是51系统中**唯一没有地址的寄存器。**

**振荡器**的工作可以由PD位（特殊功能寄存器**PCON**中的一位）控制。当PD置1时，振荡器停止工作，系统进入低功耗工作状态。 在由多片单片微机组成的系统中，为了各单片微机之间时钟信号的同步，**应当引入唯一的公用外部脉冲信号作为各单片微机的振荡脉冲。**

**ALE信号**：状态时钟经过3分频之后，产生ALE引脚上的信号输出。

**时序定时单位：时序**是用定时单位来说明的。80C51的时序定时单位共有4个：节拍、状态、机器周期和指令周期。

**机器周期**是单片微机的最小时间单位，机器周期是指令执行中每一步操作所需要的时间，一般以CPU中完成一个运算操作所需的时间作为机器周期的基本时间，其**长度是均匀**的，而各种指令的功能不同，因而**各指令执行时所需的机器周期数是可变的。**当振荡脉冲频率为12MHz时，一个机器周期为1μs，当振荡脉冲频率为6MHz时，一个机器周期为2μs。

**指令周期：**执行一条指令所需要的时间称为指令周期。它是**最大的时序定时单位**。一个机器周期包含编号为S1P1（状态1、相位1）～S6P2（状态6、相位2）共12个振荡周期。**一般，算术/逻辑运算发生在相位1（P1）周期，而内部寄存器之间的传送操作发生在相位2（P2）周期。**以**机器周期为单位**，有三种情况：单周期、双周期和四周期。当振荡脉冲频率为12MHz时，8051一条指令执行的时间最短为1μs，最长为4μs。

**复位的作用：**可以使系统处于确定的初始状态**；复位的条件：RST引脚连续两个机器周期的高电平** 。**若使用频率为6MHz的晶振，则复位信号应持续4 μs以上。**  **复位后：PC：0000H(**单片机从0000H地址开始执行程序)；**P0~P3口：0FFH；SP：07H**

**输入/输出（I/O）口：**作普通I/O口使用时，相当于**准双向口、**输出：写入P0口锁存器、输入：读入数据前应**先置1；**作地址/数据复用总线时，是**真正的双向口**

**P1口**：**准**双向I/O口，当它们作输出方式时，各口Q0管将通过饱和或截止使输出端为0或1。但当**用作输入时**，必须使Q0截止，即**先置相应锁存器为1**，这就是所谓**准双向I/O口的特点**。输入时：口锁存器**0**——引脚电平为低，不能正确读入数据。**1——可正确读入数据**。 **P2口：**8位**准**双向I/O口。 **P3口：**第一功能：普通的**准双向**I/O口；第二功能：**其口锁存器必须置为1状态**，此时与非门的状态由第二输出功能控制线的状态确定，反映第二功能输出的电平状态。

**在物理上设有4个存储器空间：**程序存储器：片内（外）程序存储器；数据存储器：片内（外）数据存储器；**在逻辑上设有3个存储器地址空间：**片内、片外统一的 64 KB程序存储器地址空间；片内256（80C52 为384）B 的数据存储地址空间；片外64 KB的数据存储器地址空间。**在访问这3个不同的逻辑空间时，应选用不同形式的指令。**

**逻辑上的三个空间又可细分为五类**：程序存贮器；工作寄存器区和数据缓冲区、位寻址区、特殊功能寄存器这**三个属于内部数据存储**；外部扩展的数据存贮器和I/O空间

**程序存贮器：**当EA=0时，低4K地址指向片外；当EA=1时，低4K地址(0000H～0FFFH)指向片内。**8031**无片内程序存贮器，其**EA必须为0。**

软件上，有一类指令对程序存贮器（其中存放的表格）进行处理**(读操作)：MOVC** A，@A+PC ； A←((A)+(PC))；**MOVC** A，@A+DPTR； A←((A)+(DPTR))

**程序存储器中从0003H～002DH单元被保留用于中断源的中断服务程序的入口地址：**7个特定地址被保留**：**复位0000H；外部中断0 0003H；计时器T0溢出 000BH；外部中断1 0013H；计时器T1溢出 001BH；串行口中断 0023H；计时器 T2/T2EX 002BH（对于8052、8032）

**低128字节中位寻址区的位地址与字节地址的转换举例：**23H.6 (字节地址) =(23H-20H)\*8+6=30=1EH(位地址)；61H(位地址)＝6\*2+20H=12+20H=0CH+20H=2CH，所以字节地址为：2CH. 1；5DH(位地址)=5\*2+20H=10+20H=0AH+20H=2AH(**所以字节地址为：2AH.D，但.D显然不合理，“.”后只能为0～7，通过推算可得2BH.5(因为D-8=5，2AH+1=2BH)**

MOV  **A** ，87H 自动翻译成**字节**寻址;MOV **C** ，87H 自动翻译成**位**寻址

**MOVX @DPTR，A；（写）; MOVX A，@DPTR；（读）**

**复位方式:1.复位操作:**其主要功能是把程序计数器PC初始化为0000H，使单片微机从0000H单元开始执行程序。**⒉ 复位信号及其产生 ：**RST引脚是复位信号的输入端。复位信号是高电平有效，其有效时间**应持续 2个机器周期(即24个振荡周期) 以上。**

**程序执行方式：程序执行方式是单片机的基本工作方式。**一般在0000H开始的单元中存放一条无条件转移指令，以便跳转到实际主程序的入口去执行。

**退出待机方式：**在中断服务程序中安排一**条RETI指令**，就可以使单片机恢复正常工作，从设置待机方式指令的下一条指令开始继续执行程序。**只能依靠复位退出掉电保护方式：**当VCC恢复正常后，**只要硬件复位信号维持10ms**，就能使单片机退出掉电保护方式，CPU则从进入该方式的下一条指令开始重新执行程序。

**1.80C51单片微机在片内集成了哪些主要逻辑功能部件？各部件最主要功能是什么？** 80C51单片机在片内主要包含中央处理器CPU(算术逻辑单元ALU及控制器等)、只读存储器ROM、读／写存储器RAM、定时器／计数器、并行I／O口Po～P3、串行口、中断系统以及定时控制逻辑电路等，各部分通过内部总线相连。 1．中央处理器(CPU)：单片机中的中央处理器和通用微处理器基本相同，是单片机的最核心部分，主要完成运算和控制功能，又增设了“面向控制”的处理功能，增强了实时性。2．内部程序存储器：程序存储器用于存放程序和表格、原始数据等。3．内部数据存储器(RAM)：在单片机中，用读／写存储器(RAM)来存储程序在运行期间的工作变量和数据。4．I／O口：单片机提供了功能强、使用灵活的I／O引脚，用于检测与控制。有些I／O引脚还具有多种功能，比如可以作为数据总线的数据线、地址总线的地址线或控制总线的控制线等。有的单片机I／0引脚的驱动能力增大。5．串行I／O口：用以实现与某些终端设备进行串行通信，或与一些特殊功能的器件相连的能力，甚至用多个单片机相连构成多机系统。6.定时器／计数器：定时器／计数器可以编程实现定时和计数功能。7．中断系统：80C51单片机的中断功能较强，具有内、外共5 个中断源，具有两个中断优先级。8．定时电路及元件：单片机内部设有定时电路，只需外接振荡元件。近年来有些单片机将振荡元件也集成到芯片内部。单片机整个工作是在时钟信号的驱动下，按照严格的时序有规律地一个节拍一个节拍地执行各种操作。

**2. 80C51片内数据存储器低128单元划分为哪3个主要部分？各部分主要功能是什么？寄存器区**。共4组寄存器，每组8个存储单元，各组以R0～R7作为单元编号。常用于保存操作数及中间结果等。：R0～R7也称为“通用工作寄存器”，占用00H～1FH共32个单元地址。**位寻址区**。单元地址为20H~2FH，既可作为一般RAM单元使用，按字节进行操作，也可对单元中的每一位进行位操作。因此，称为“位寻址区”。寻址区共有16个RAM单元，共计128位，位地址为OOH～7FH。**用户RAM区**。在内部RAM低128单元中，除去前面两个区，剩下80个单元，单元地址为30H～7FH。在用户RAM区内可以设置堆栈区。

**80C51单片机的寻址方式有哪几种？**80C51 单片机的寻址方式有七种。即：寄存器寻址、直接寻址、寄存器间接寻址、立即寻址、基址寄存器、加变址寄存器变址寻址、相对寻址和位寻址。**其中：(1)访问特殊功能寄存器SFR有**直接寻址和位寻址方式。 **(2)访问内部RAM单元有**直接寻址、寄存器间接寻址和位寻址方式。**(3)访问外部RAM单元有**寄存器间接寻址。 **(4)访问外部程序存储器有**立即寻址、变址寻址和相对寻址方式。

**执行任何一条指令都需要使用操作数、**在立即数前面必需加上前缀**“＃”。**

**直接寻址方式只能给出8位地址，限于片内RAM：**低128单元，在指令中直接以单元地址形式给出；特殊功能寄存器，除了以单元地址形式给出外，还可以以寄存器符号形式给出。**直接寻址是访问特殊功能寄存器的唯一方法**

**寄存器寻址：如：**指令INC R0**（用于计数等）：**把寄存器R0的内容加1，再送回R0中。由于操作数在R0中，指定了R0，也就得到了操作数。**寄存器寻址方式的寻址范围包括：4**个寄存器组共32个通用寄存器；部分特殊功能寄存器

**寄存器间接寻址（寻地址及地址内内容）：**在寄存器间接寻址中，在寄存器的名称前面加前缀“＠”。片内RAM的低128单元，**只能采用R0或R1为间址寄存器。其形式为@Ri（i＝0，1）；**片外RAM 的64 KB单元，使用**DPTR**作为间址寄存器。例如：MOVX A，＠DPTR。

**变址寻址方式**只能对**程序存储器**进行寻址。**位寻址：MOV C，2BH：**功能是把位寻址区的2BH位状态送累加位C。

**为什么在12MHz晶振条件下，80C51的指令执行时间分别为1*μs*、2*μs*、4*μs？***因为80C51的指令周期根据指令的不同，可包含一、二、四个机器周期

16位常数在指令的第二、第三字节中（第二字节为**高位字节DPH**，第三字节为**低位字节DPL**）**MOV DPTR，＃1234H**

**执行结果：（DPH）＝12H，（DPL）＝34H**

**算术运算类指令：**包括加、减、乘、除四则运算，还有带借位减法、比较指令。加法类指令包括加法、带进位的加法、加1以及二进制和十进制调整。**ALU仅执行无符号二进制整数的算术运算。对于带符号数则要进行其它处理。**

**逻辑“与”运算指令**用做清除或屏蔽某些位**。逻辑或**运算指令用做指定位强迫置位。给某些位置1，合并两个数中的“1”。**“异或”：相同为0，相反为1**

**根据目的地址计算偏移量：**rel=(目标地址－当前PC地址) 的低8位，而高8位必须是00H或FFH，否则超出短转移范围，溢出出错。例：2100H 80 FE  **HERE:SJMP HERE** rel:2100H－2102H=FFFEH。高8位是FFH,表示负跳,偏移量为FEH (－2)。**用于程序结束或中断等待**

**间接转移指令JMP @A+DPTR：当DPTR的值固定，而给A赋以不同的值，即可实现程序的多分支转移。如实现键盘译码散转功能**

**中断服务程序必须以RETI为结束指令。**

**高级语言：**特点：直观、易学、易懂、通用性强、易于移植到不同类型的机器中。但是，**计算机不能直接识别高级语言**，需将其转换为机器语言才能被识别和执行。

在指令语句中，标号位于一个语句的开头位置，由字母和数符组成，**字母打头，冒号“：”结束。**

操作数与操作码之间用空格“”分隔，若有两个操作数，这两个操作数之间**必须用逗号“，”分开**

用**EQU语句**给一个标号赋值以后，在整个源程序中该标号的值是**固定的，不能更改。**

**补码=原码取反+1，例：求双字节补码程序：**首先对低字节数取补，然后判其结果是否为全“0”。若为“0”，则高字节数取补；否则，高位字节数取反。

循环的执行过程是**从内向外逐层**展开的。内层执行完全部循环后，外层则完成一次循环，逐次类推。

**中断矢量：**矢量中断：当CPU响应中断时，**由硬件**直接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设备的**中断服务程序的入口。**

**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。

**串行口控制寄存器SCON：**TI(SCON.1)： 串行口发送中断源； RI(SCON.0)： 串行口接受中断源。

**中断响应过程和响应时间：**过程：中断采样、中断查询、中断响应；时间：一般，在单级中断系统中，中断的响应时间最短为3个机器周期，最长为8个机器周期

**80C51的单步操作：**重要特性，即执行中断返回指令RETI后，必须至少执行一条其它指令后，才能响应新的中断。

**硬件上**，把按键产生的脉冲连到 INT0，作为外部中断0的中断请求信号，并把电路设计为不按键时为低电平，按一次键产生一个正脉冲。**软件上**，编制外部中断0的中断服务程序。

**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0 13位定时器/计数器：**TLx低五位溢出，向THx进位；计数溢出时，TFx置位

**方式2 定时常数自动重装载的8位定时器/计数器：**溢出时，TFx置位、重装载；TLX为计数寄存器；THX为计数常数寄存器

**方式3** TL0占用了T0的GATE、INT0、TR0、T0引脚 以及TF0和T0的中断矢量等；TH0占用了T1的TR1、TF1和T1的中断矢量。 **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)，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。

**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。**

**单工方式：**信号在信道中只能沿**一个方向**传送；**半双工方式**：通信双方均**具有发送和接受能力**，信道也**具备双向传输性能**，但是不能同时既发送信息又接受信息，**只能沿一个方向**传送信息；**全双工方式**：**可同时发送和接受信息**

**在二进制**的情况下，波特率（表示每秒传输离散信号事件的个数，或每秒信号电平的变化次数）与比特率**数值相等**

**80C51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（1）

**串行口的应用：**串行口的波特率发生器及波特率计算、方式0的编程和应用（应用方式0外扩并行I/O口）、方式1的编程和应用、方式2和3的编程和应用

**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**

**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：**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

**LED显示电路：**静（动）态显示电路**——分时选通**

**1、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。

**2、80C51单片机寻址方式所对应的寄存器或存储器空间如何？**

这些寻址方式所对应的寄存器和存储空间如下表所示。

**3、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。

**4、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、P1、P2和P3都是准双向口。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。 另外，P0口的驱动能力为8个TTL负载，而其它口仅可驱动4个TTL 负载。

**6、80C51单片机串行口有几种工作方式？如何选择？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。

**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 。

（1）MOV A,R0

MOV R1,A

（2）MOV R2,60H

（3）MOV DPTR,#1000H

MOVX A,@DPTR

MOV 60H,A

（4）MOV DPTR,#1000H

MOVX A,@DPTR

MOV R2,A

（5）MOV DPTR,#1000H

MOVX A,@DPTR

MOV DPTR,#2000H

MOVX @DPTR, A

**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，试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

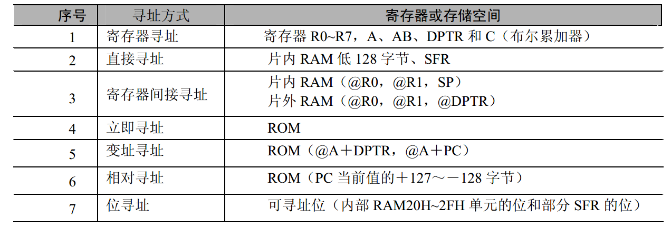
MOV @R0，A

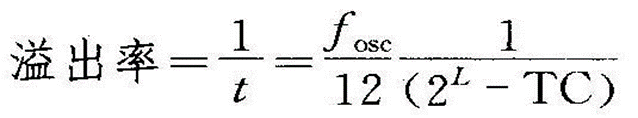
MOV A，#3BH

MOV 41H，A

MOV 42H，41H

答：（A）=3BH，（R0）=40H，（40H）=00H，（41H ）=3BH，（42H）=3BH。



****

**1.试分析以下程序段的执行结果。 2.已知(30H)=40H，(40H)=10 H，(10H)=00H，(P1)=CAH，请写出执行以下程序段后有关单元的内容。**

MOV SP，#60tt ；(SP)=60H MOV R0，#30H ；(RO)=30H

NOV A，#88tt ；(A)=88H MOV A，@R0 ；(A)=40H

MOV B．#0FFH ；(B)=FFH MOV Rl，A ；(R1)=40H

PUSH ACC ；(SP)=61H，(61H)=88H MOV B，@R1 ；(B)=1OH

PUSH B ；(SP)=62H，(62H)=FFH MOV @R1，P1 ；(40H)=CAH

POP ACC ；(A)=FFH，(SP)=61H MOV A，@RO ；(A)=40H

POP B ；(B)=88H，(SP)=60H MOV 10H，#20H ；(10H)=20H

程序段的执行结果：累加器A和寄存器 MOV 30H，10H ；(30H)=20H

B的内容通过堆栈进行了交换

**3.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。**

(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

**4.已知(R1)=20 H，(20 H)=AAH，请写出执行完下列程序段后A的内容。**

MOV A，#55H ；(A)=55H

ANL A，#0FFH ；(A)=55H

ORL 20H，A ；(20H)=FFH

XRL A，@R1 ；(A)=AAH

CPL A ；(A)=55H

执行完程序段后，A的内容为5 5 H。

**5.80C51单片机从内部RAM的31H单元开始存放一组8位带符号数，字节个数存放在30H中。请编写程序统计出其中正数、0和负数的数目，并把统计结果分别存入20H、21H和22H三个单元中。加上必要的伪指令，并对源程序加以注释。**

LENGTH EQU 30H ；数据长度

DATA\_ADR EQU 31H ；数据首地址

POS\_NUM EQU 20H ；正数个数

ZERO\_NUM EQU 21H ；0个数

NEG\_NUM EQU 22H ；负数个数

ORG 0000H

AJMP MAIN ，

ORG 0030H

MAIN： MOV POS\_NUM，#0 ；计数单元初始化为0

MOV ZERO\_NUM，#0

MOV NEG\_NUM，#0

MOV R1，#LENGTH ；数据长度

MOV R0，#DATA\_ADR ；数据首地址

LOOP： MOV A，@R0 ．

JB ACC．7，INC\_NEG ；符号位为1，该数为负数，跳转加1

CJNE A，#0，INC\_POS

INC ZERO\_NUM ；该数为0，0个数加1

AJMP L00P1

INC NEG：INC NE~\_NUM ；负数个数加1

AJMP LOOPl ．

INC\_POS：INC POS\_NUM ；该数为正数，正数个数加1

LOOPl： INC R0 ；判断统计是否结束

DJNZ R1，LOOP

END

例如：已知(30H)=08H，31H单元起存放数据为00H，80H，7EH，6DH，2FH，34H，

EDH，FFH。

执行结果：(20 H)=04 H，(21 H)=01H，(22H)=03 H。

**6.两个10位的无符号二一十进制数，分别从内部RAM的40 H单元和50H单元开始存放。请编程计算该两个数的和，并从内部RAM的60 H单元开始存放。加上必要的伪指令，并对源程序加以注释。**

10位的无符号二～十进制数，占5字节，每个字节存放一个压缩BCD码(2位)。

ORG 0000H MOV R0，20H

AJMP MAIN MOV @R0，A ；存和

0Re 0030H MOV R2，20H ；恢复和数地址

MAIN： MOV R7，#05H ；十位(5字节)计数 MOV R0，B ；恢复被加数地址

MOV R0，#40H ；被加数首址 INC R0 ；三个地址指针均加1

MOV R1，#50H ；加数首址 INC R1

MOV R2，#60H ；和数首址 INC R2

CLR C ；清C标志位 DJNZ R7，ADDB ；多字节加未结束，则循环

ADDB： MOV A，@R0 HERE： SJMP HERE

ADDC A，@R1 END

DA A ；二一十进制调整

MOV B，R0 ；保护被加数地址

MOV 20H, R2

注意：寄存器间接寻址只针对R0和R1，所以存和时不能使用指令“MOV@R2，A”。

例如：

40 H～44 H内容为78 H，10 H，10 H，10 H，10 H

5O H～54 H内容为42 H，10 H，10 H，10 H，10 H

1 0 1 0 1 0 1 0 7 8

即BCD数 +1 0 1 0 1 0 1 0 4 2

2 0 2 0 2 0 2 1 2 0

运行结果：6O H～64 H单元中的数为20 H，21 H，20 H，20 H，20 H。

**7.将外部RAM的40 H单元中的一个字节拆成2个ASCII码，分别存入内部数据存储器40 H和41 H单元中。试编写以子程序形式给出的转换程序，说明调用该子程序的入口条件和出口功能。加上必要的伪指令，并对源程序加以注释。**

子程序人口条件：准备拆为2个ASCII码的数存入外部RAM的40 H单元中。

子程序出口功能：完成外部RAM单元一个字节拆成2个ASCII码，分别存入内部数据存储器40 H和41 H单元中。

ORG 1 000H

B\_TO\_A：MOV DPTR，#40H ；外部RAM40H单元

MOV R0，#40H

MOVX A，@DPTR ；取数

PUSH A

ANL A，#0FH ；低4位转换为ASCII码

LCALL． CHANGE

MOV @RO，A

INC R0

POP A ’

SWAP A

ANL A，#0FH ；高4位转换为ASCII码

LCALL CHANGE

MOV @R0，A

RET

CHANGE：CJNE A，#0AH，NEXT ；转换子程序

NEXT： JNC NEXT2 ；≥0AH，转移

ADD A，#3 0H ；≤9，数字0-9转化为ASCII码

RET

NEXT2： ADD A，#37H ；字母A～F转化为ASCII码

RET

END

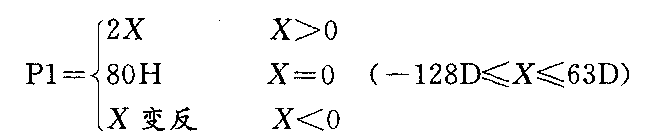
设外部(40 H)=12 H。

执行程序B\_TO\_A后：内部(40 H)=31 H，(41 H)=32 H。

设外部RAM(40 H)=ABH。

执行程序B\_TO\_A后,内部(40 H)=41 H，(41 H)=42 H。

**8.根据8100 H单元中的值X，决定P1口引脚输出为：**



加上必要的伪指令，并对源程序加以注释。 ．

ORG 0000H

SJMP BEGIN

ORG 0030H

BEGIN： MOV DPTR，#8100H

MOVX A，@DPTR

MOV R2，A

JB ACC.7，SMALLER；有符号数<0

SJMP UNSIGNED ；无符号数≥0

SMALLER：DEC A ；X<0，输出-X(先减1，再取反)

CPL A

MOV P1，A

SJMP OK

UNSIGNED：CJNE A，#00H，BIGGER ；不等于0即大于0

MOV P1，#80H ；X等于0，输出80H

SJMP OK

BIGGER： CLR C ；X大于0，输出A×2

RLC A ；A×2

MOV P1，A

OK： SJMP $

END

例如：输入55 H，P1口引脚输出AAH；输入00 H，P1口引脚输出80 H；输入F1(一1 5的补码)，P1口引脚输出0FH。

**9.求双字节补码程序：设对addrl，addrl＋1的双字节数取补后存入addr2和 addr2十1单元中，其中高位字节在高地址单元中。8位微机对双字节数取补需分两次进行。**

START：MOV R0，＃addrl ；原码低字节地址码送R0

MOV R1，＃addr2 ；补码低字节地址码送R1

MOV A，＠R0 ；原码低字节内容送A

CPL A ; 按位取反

INC A ；A内容取反加l，即取补

MOV ＠R1，A ；低字节补码存addr2单元

INC R0 ；指向原码高字节

INC R1 ；指向补码高字节

JZ LOOP1 ；当（A）＝0，转LOOP1

MOV A，＠R0 ；原码高字节送A

CPL A ；高字节内容取反

MOV ＠R1，A ；字节反码存(addr2十1)单元

SJMP LOOP2 ；转LOOP2，结束

LOOP1：MOV A，＠R0 ；原码高字节送A

CPL A ；对高字节数取反

INC A ；对高字节数取补

MOV @R1，A ；高字节补码存(addr2+1)单元

LOOP2：…

END ；结束

**1.什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。在80C51单片机中有高、低两个中断优先级，通过中断优先级寄存器IP来设定。在80C51单片机中存在同一优先级内由内部硬件查询序列确定的第二个优先级结构。

**2.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。①INT0(P3．2)—一外部中断O。当IT0(TCON．O)=0 时，信号触发方式为低电平有效；当IT0(TCON．O)=1时，信号触发方式为下降沿有效。②INTI(P3．3)——外部中断1。当ITI(TCON．2)=0时，信号触发方式为低电平有效；当ITl(TCON．2)=1时，信号触发方式为下降沿有效。

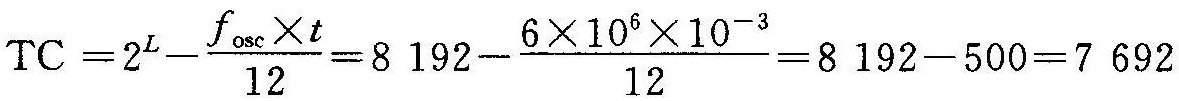
**3.80C51单片机有哪些中断源?对其中断请求如何进行控制?** 1．中断源：80C51．单片机中有5个中断源。80c52中增加了一个中断源--定时器／计数器T2中断，共有6个中断源。每一个中断源都能被单片机程控为高优先级或低优先级。 80c51单片机的5个中断源包括2个外部中断源和3个内部中断源。2个外部中断源为INT0和INT1，外部设备的中断请求信号、掉电等故障信号都可以从INT0或INT1引脚输入。3个内部中断源为定时器／计数器TO、T1的定时／计数溢出中断源和串行口发送或接收中断源。 2．中断的允许和禁止由中断允许寄存器IE控制

**例 ．要求在 P1.0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P1.0引脚上的电平变反

(1) 解：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(213＝8 192)，定时时间 t＝1ms＝0.001s

定时常数：

TC为7692＝1E0CH，

二进制数TCB＝0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 B，取低13位，其中高8位TCH＝F0H，低5位为TCL＝0CH.计数长度为1E0CH=7692，定时为(8192—7692)×2μs=0.001S

( 2) 编 程 ORG 0000H

AJMP MAIN

ORG 000BH ；T0中断矢量

AJMP INQP

ORG 0030H

MAIN： MOV TMOD，＃00H ；设T0为定时器方式0

MOV TH0，＃0F0H ；写定时常数(定时1ms)

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 ；T0中断服务程序

INQP：MOV TH0，＃0F0H ；重写定时常数

MOV TL0，＃0CH

CPL P1．0 ；P1．0变反输出

RETI ；中断返回

**在80C5 1单片机系统中，已知时钟频率为6 MHz，选用定时器T0设置为定时方式3，请编程使P1.0和P1.1引脚上分别输出周期为4 ms和800 μs的方波。加上必要的伪指令，并对源程序加以注释。**

机器周期为2μs，定时分别为2 ms和400μs。

计算：400 μs定时，400μs=(28--TC)×2 μs，TC=38 H。

程序如下：

ORG 0000H

0000 0130 AJMP MAIN

ORG 000BH ；定时器TO中断矢量 。

000B 2100 AJMP TIME

MAIN：

0030 7805 MOV R0，#05H

0032 758903 MOV ．TMOD，#03H ；T0方式3，定时器中断

0035 758A38 MOV TL0，#38H ；TLO定时400μs

0038 D28C SETB TR0 ；开启定时器TL0

003A C28E CLR, TRl

003C D2A9 SETB ET0 ；开定时器TLO中断

003E D2AF SETB EA

0040 80FE SJMP$ ；中断等待

ORG 01 00H ．

TIME：

0100 758A38 MOV TL0，#38H ；TL0定时400~s

0103 B291 CPL． P1.1 ；400μs 定时到，P1.1输出变反

0105 D804 DJNZ R0，RETURN

0107 7805 MOV R0，#05H

0109 B290 CP[． P1．0 ；400μs \* 5=2 ms到，P1.O输出变反

RETURN：

010B 32 RETI

END