●**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.

●**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.

1. **RISC的特点?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)

●**CPSR与SPSR的关系?**在处理器所有的运行模式下均可以访问当前的程序状态寄存器(CPSR),每一种异常模式下又都有一个专用的程序状态保存寄存器(SPSR),当异常发生时,SPSR用于保存当前程序状态寄存器CPSR的状态,以便从异常退出时,由SPSR来恢复CPSR,从而进行异常处理.

●**大端,小端模式?**ARM体系结构可以用两种方法存储字数据,称之为大端格式和小端格式.**大端格式:**字数据的高字节存储在低地址中,而字数据的低字节则存放在高地址中.**小端格式**:与大端储存格式相反,在小端储存格式中,低地址中存放的是字数据的低字节,高地址存放的是字数据的高字节.

●**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.

●**ARM的运行模式及特点?**用户(usr):正常程序执行模式.系统(sys):运行操作系统的特权任务.3.快中断(FIq):支持高速数据传输及通道处理.中断(irq):用于通用的中断处理.管理(svc):操作系统保护模式.中止(abt):用于支持虚拟内存和/或存储器保护.未定义(und):支持硬件协处理器的软件仿真.

●**中断的执行与子程序相似点?**

**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.

●**中断执行与子程序调用的差别?**

**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.

●●**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.

**P0口:**多功能8位口,字节访问地址: 80H,位访问地址:80H～87H.

**⒉ P0口的功能**

**⑴ 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.

**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.

**P2口的工作状态是**输出高8位地址.

**P2口的功能:**

⑴ I／O口

⑵ 程序存储器或片外数据存储器的高8位地址

●**80C51单片机的工作方式**

**复位:**单片机的初始化操作,把PC初始化为0000H,单片机从0000H单元开始执行.

**⑴ 复位信号**

RST引脚: 高电平有效,有效时间应持续2个机器周期以上

**(2)复位方式:** 上电自动复位,按键电平复位和外部脉冲复位

**程序执行**

单片机的基本工作方式.

复位后PC＝0000H,程序执行从0000H开始.

一般在0000H开始的单元中存放一条无条件转移指令,跳转到实际主程序入口去执行.

**低功耗**

**两种低功耗方式:待机方式和掉电保护方式.**

**⒈ 待机方式**

⑴ IDL=1,80C51进入待机方式

振荡器仍运行,并向中断逻辑,串行口和定时器/计数器电路提供时钟,中断功能继续存在 .

向CPU提供时钟的电路被阻断,CPU不工作,SP,PC,PSW,ACC及通用寄存器冻结在原状态.

⑵ 采用中断方式或硬件复位来退出待机方式.

若产生一个外部中断请求信号,单片机响应中断,PCON.0位(IDL位)被硬件自动清“0”, 单片机退出待机方式进入正常工作方式.

**⒉ 掉电保护方式**

 ⑴ PD位控制单片机进入掉电保护方式

当80C51检测到电源故障时,进行信息保护,把PCON.1位置“1”,进入掉电保护方式.

单片机一切工作停止,只有内部RAM单元内容被保护.

(2) 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则从进入待机方式的下一条指令开始重新执行程序.

**编程和校验.**

**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**

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**

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**

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**

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**

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**

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.

例:设(R0)＝20H,(A)＝36H(00110110B),(20H)＝75H(0111010lB).

执行指令:XCHD A,＠R0

**⑵ SWAP A**

将累加器A的高,低半字节交换

例:设(A)＝36H(0011 0110B)

执行指令: SWAP A

XCHD和SWAP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)

例:中断响应时(SP)＝30H,DPTR的内容为0123H,执行入栈指令:

PUSH DPL ;DPL内容入栈

PUSH DPH ;DPH内容入栈.

●**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:**ANL**,符号“∧”表示:

逻辑“或”助记符:**ORL**, “∨”表示:

逻辑“异或”助记符为**XRL**, “⊕”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)

**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.

●**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**

●●●●**嵌入式设计及应用**●●●●

●**RISC 的设计准则:**

1.指令集:选用使用频率较高的一些 简单指令 复杂指令的功能由简单指令来组合一个时钟周期 内完成一条指令

2.流水线:指令的处理过程被拆分成更小的,能够被流水线执行的单元.

3.寄存器:拥有多个通用寄存器

4.LOAD/STORE结构:实现寄存器与外存之间的数据传送.

●**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

在开始执行代码时,应该处于ARM状态.

●**状态寄存器:**

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.

**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

●**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 不相等

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

●**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.

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**

●**ARM指令分类说明**

**1.数据处理指令**

**2.跳转指令:**

**B 转移指令**

功能:跳转到目的地址.

跳转范围:当前指令的±32M字节地址内(ARM指令为字对齐,最低2位地址固定为0).

**BL 带链接的转移指令**

功能:PC拷贝到链接寄存器,然后跳转到指定地址.

跳转范围:当前指令的±32M字节地址内.

**BX 带状态切换的转移指令**

**功能:**跳转到目标地址;处理器工作状态切换.

**目标地址:**寄存器Rn和0xFFFFFFFE相与的结果.Rn的第0位拷贝到CPSR中T位,位[31∶1]移入PC.

**跳转范围:**当前指令的±32M字节地址内.

**BLX 带链接和状态切换的转移指令**

**功能:**跳转到目标地址;PC值保存到LR寄存器;处理器工作状态切换.

**3.Load/Store指令**

**A.单寄存器的存取指令(LDR,STR)**

**LDR:**从内存中读取单个字或字节数据存于寄存器;

**LDR Rd,[Rn,Rm] ;**将内存中的地址为Rn+Rm的字数据装入寄存器Rd中

**LDR Rd,[Rn,Rm] ! ;**将内存中的地址为Rn+Rm的字数据装入寄存器Rd中,并将新地址Rn+Rm写入Rn

LDR{cond}T <Rd>,<addr>;

T:若指令有T,即使处理器是在特权模式下,也将存储系统访问看成是在用户模式的.

**STR:**将寄存器中的单个字或字节数据保存到内存.

**STRH:**半字数据存储指令

STR{cond}H <Rd>,<addr>;

**STRT:**用户模式的字数据存储指令

STR{cond}T <Rd>,<addr>;

**B.多寄存器存取指令(LDM,STM)**

LDM:批量数据加载指令:

LDM {cond}{<type>} Rn{!}, reglist {^}

指令中,type字段有以下几种:

IA 每次传送后地址加一

IB 每次传送前地址加一

DA 每次传送过后地址减一

DB 每次传送前地址减一

FD 满递减堆栈

ED 空递减堆栈

FA 满递增堆栈

EA 空递增堆栈

**C.单寄存器交换指令(SWP)**

**D.SWP 字数据交换指令**

**SWP{cond} Rd,Rm,[Rn];**

功能:从Rn所表示的内存装载一个字并把这个字放置到目的寄存器Rd中,然后把寄存器Rm的内容存储到同一内存地址中,即Rd=[Rn],[Rn]=Rm,其中Rm,Rn均为寄存器.

例:SWP R0,R1,[R2];将R2所表示的内存单元中的字数据装载到R0,然后将R1中的字数据保存到R2所表示的内存单元中.

**4.程序状态寄存器指令**

MRS指令:对状态寄存器CPSR和SPSR进行读操作.CPSR:了解当前处理器的工作状态.SPSR:了解进入异常前的处理器状态.

MRS指令格式

MRS{cond} Rd,CPSR/SPSR

**5.协处理器指令**

**6.软件中断指令**

●**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

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

●**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



●●●●●●●●●●●●

**可能是实验题**

**根据定时时间t、定时器工作方式(确定L)及晶体振荡频率fosc,计算TC值(十进制数),转换成二进制数TCB,然后送入THi、TLi。**

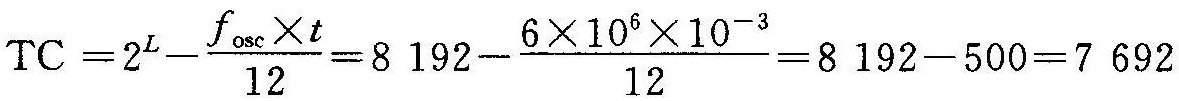
对于T0、T1:TCB=TCH＋TCL

MOV THi,＃TCH;

MOV TLi,＃TCL;

**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:



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;

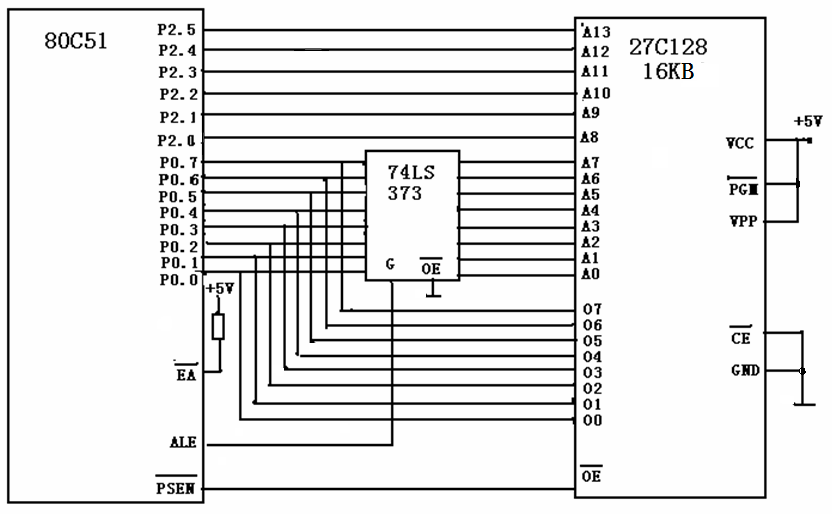
ORG 2000 H;T0中断服务程序

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



⑴ **I/O口的数据传送指令**

输出数据的指令:

MOV Px, DATA

输入数据的指令有:

MOV DATA, Px

⑵ **I/O口的位操作指令**

位传送指令 MOV Px.y, C

位清0指令 CLR Px.y

位置1指令 SETB Px.y

位取反 CPL Px.y

位为1转移 JB Px.y, rel

位为0转移 JNB Px.y, rel

位为1转移并清零 JBC Px.y, rel

⑶ **I/O口其它操作指令**

逻辑与指令 ANL Px, A

逻辑和指令 ORL Px, A

逻辑异与指令 XRL Px, A

加1指令 INC Px

减1指令 DEC Px

减1条件转移指令 DJNZ Px, rel

数值比较转移指令 CJNE A, Px, rel

**例:执行指令 MOV P1,#7FH**

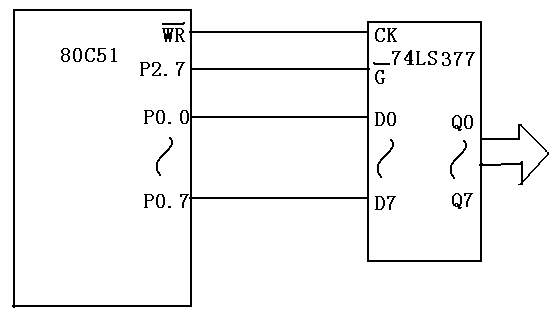
执行结果为:P1.7引脚输出为低电平,其余7个引脚都输出高电平

**例:将一个数据字节从74LS377输出**

MOV DPTR,#7FFFH;地址指针指向74LS377

MOV A,#DATA;将输出数据送A

MOVX @DPTR,A;输出数据



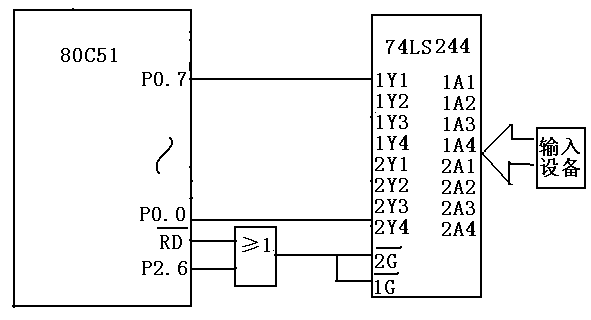
**⒉ 简单输入口的扩展**

数据输入需采用8位三态门控制电路。

**例:数据输入**

MOV DPTR,#0BFFFH;指向74LS244口地址

MOVX A,@DPTR;读入数据



**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。

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:产生锯齿波**

DAC0832的地址为7FFFH(P2.7=0)

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN:MOV DPTR,#7FFFH ;DAC寄存器地址

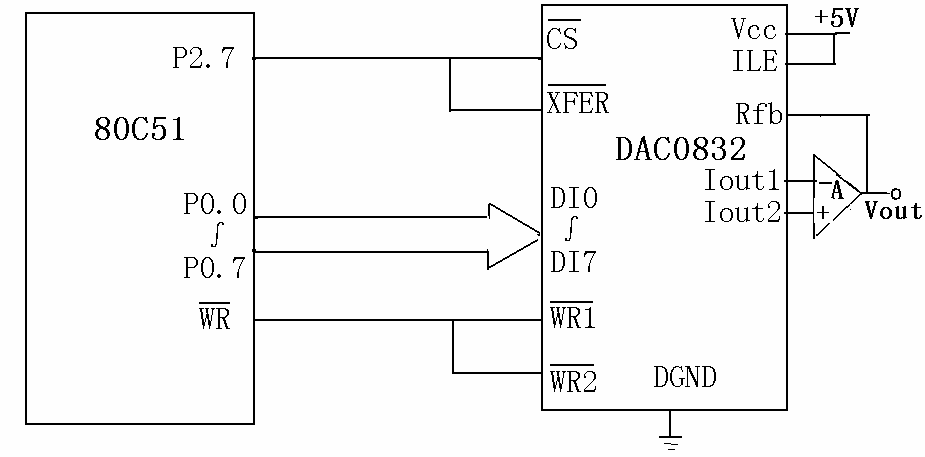
MOV R0,#0;转换初值

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①LCALL 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

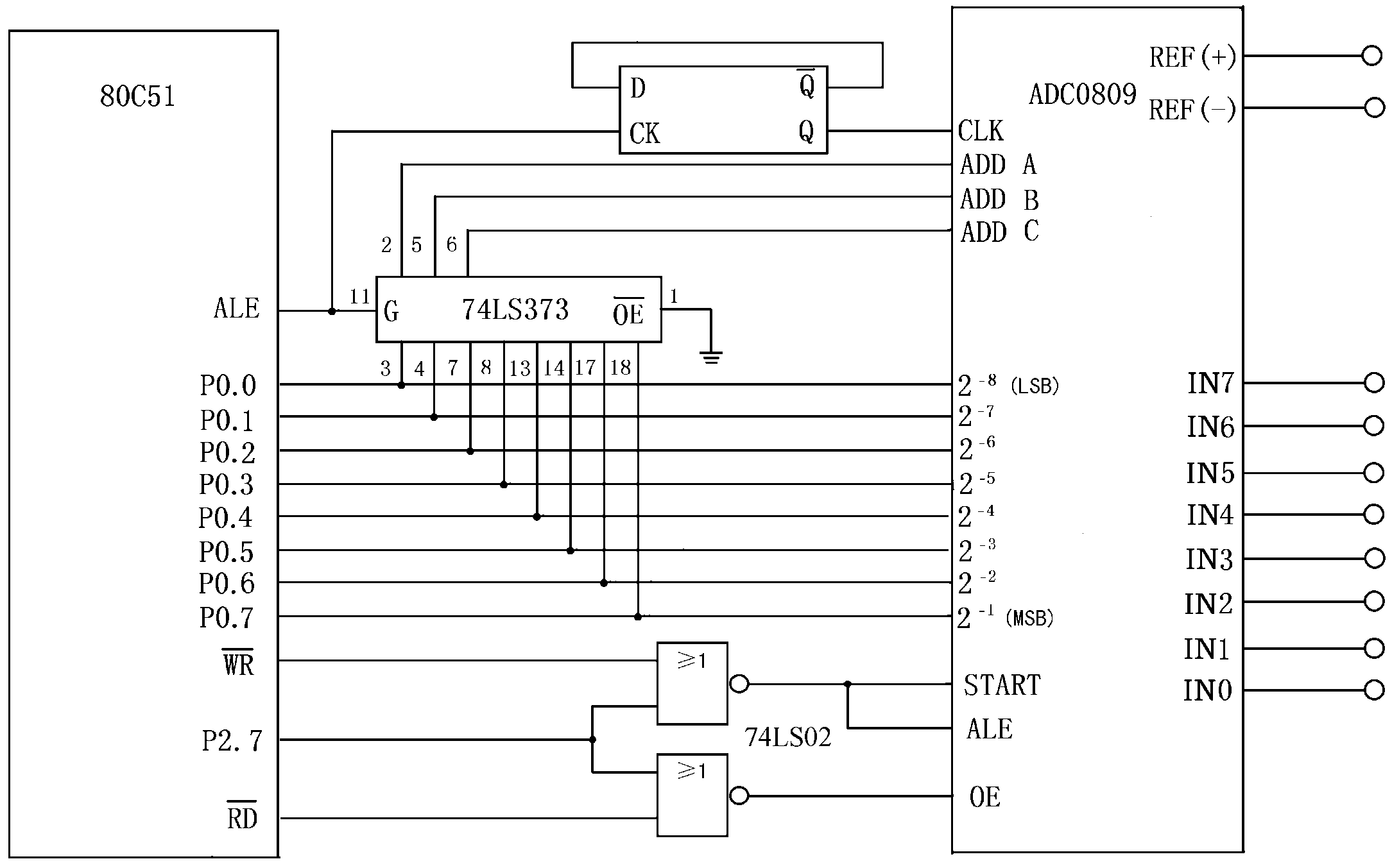
INC DPTR;指向下一通道地址

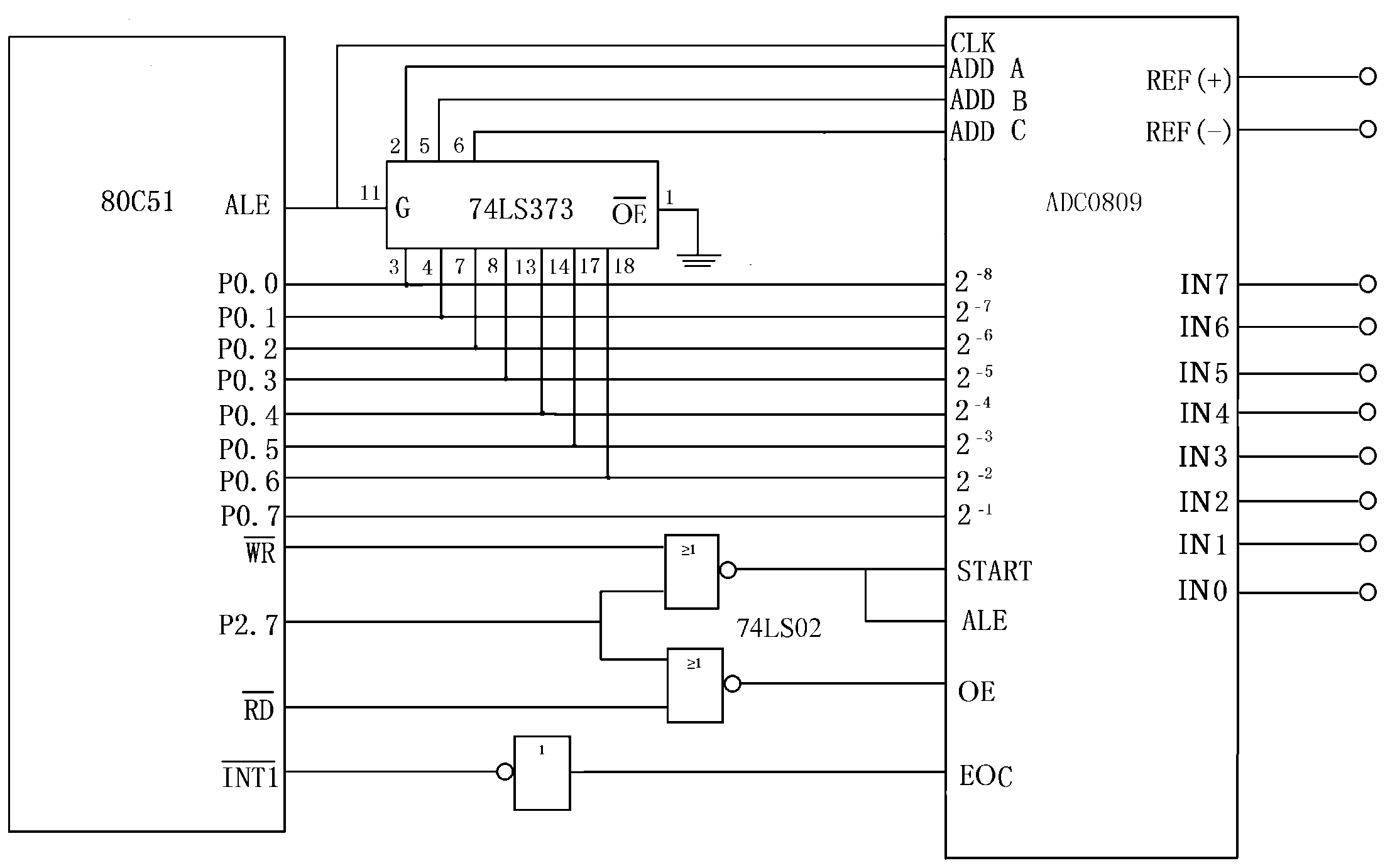
INC R1

DJNZ R7,LOOP 8个通道未转换完,则继续。

D128μs:… ;延时128μs 子程序

RET





**例: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非=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

**在一组测量数据中,挑选出大于标准m的数值作为合格的产品,而那些小于m 的数值作为不合格产品,则被剔除掉。**

设数据组为X1,X2,…,X10共10个。

ORG 0000H

PX:CLR 00 ;设交换过标志

MOV R3,#09H;10个数据比较,第一次;比较两个数据,比较次数为(n-1)次

MOV R0,#50H;10个单元无符号数存放首址

MOV A,@R0

PX1:INC R0

MOV R1,A

SUBB A , @R0 ; DX－DX+1

MOV A , R1

JC PX2;DX<DX+1则转PX2,不交换

SETB 00H;DX>DX+1量交换标志位,20H.0=1

XCH A ,@R0; DX与DX+1交换

DEC R0

XCH AV@R0

INC R0

PX2:MOV A , @R0 ;A< DX+1

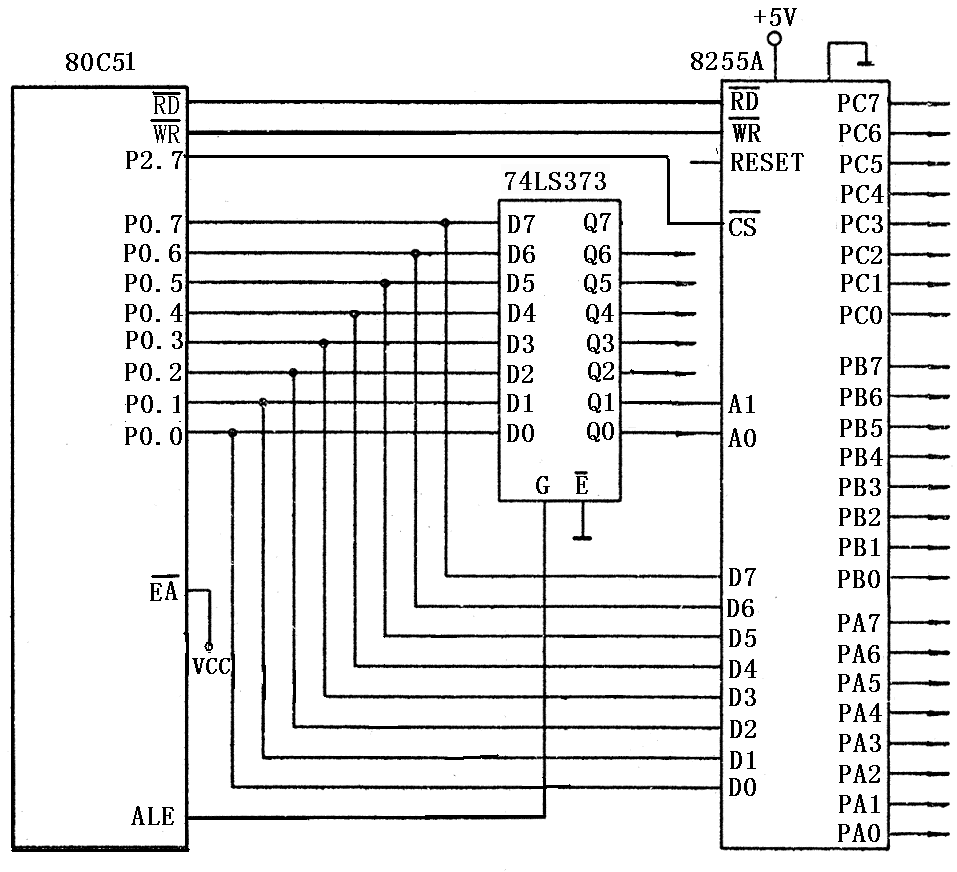
DJNZ R3 , PX1 ;比较9次

JB 00H , PX ;有交换则再比较一遍

END0:SJMP END0

●●●●●●●●●●●●●●●

**8255A与80C51的接口及应用**



PA地址：7FFCH(A15=0，A1=0，A0=0)

PB地址：7FFDH(A15=0，A1=0，A0=1)

PC地址：7FFEH(A15=0，A1=1，A0=0)

控制寄存器地址;7FFFH(A15=0，A1=1，A0=1)

**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。**

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

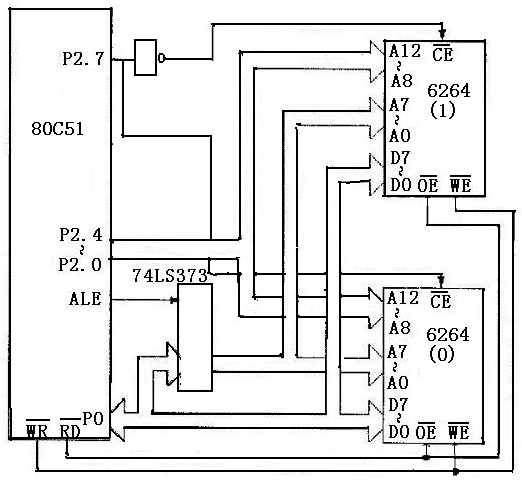
MOVX @DPTR，A

MOV DPTR，#7FFEH ;从C口输出

MOVX @DPTR，A

**2. 片外并行数据存储器的扩展**

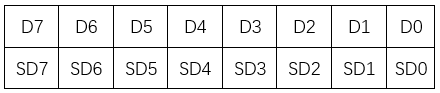
**例：用两片6264扩展16KB片外并行数据存储器,采用线选法寻址：用一根口线P2.7来寻址。**



●●●●●●●●●●●●●

**假卷子2.0**

1. **填空**
2. 若单片机使用的频率为6MHz的晶振,那么时钟周期为**0.167us**,机器的周期为**2us**.
3. 单片机的特殊功能寄存器区位于片内从**80H**到ffh的地址范围,对特殊功能寄存器的访问只能采用**变址**寻址的方式.
4. 当T0工作在方式3时,将T0分为两个独立的**8**位定时/计数器,此时T1的功能受到了限制,只能作为不需要中断功能的**波特率**发生器使用.
5. MCS-51的外部中断有触发方式,一种是**下降沿**,另一种是低电平触发.
6. 8051单片机的工作寄存器分成**4**个组,每个组**8**个字节.
7. ARM微处理器共有37个32位寄存器,其中**31**个为通用寄存器,**6**个为状态寄存器.
8. ARM状态下,SP寄存器指的是**R13**,LR寄存器指的是**R14**,PC寄存器指的是**R15**.
9. ARM体系结构可用两种方法储存子数据,其分别为**大端格式**和**小端格式**.
10. S3C410中,其地址空间有**8**个BANK,每个BANK最大的容量是**126Mb**.
11. 8051单片机外部中断请求信号有电平方式和**边沿触发方式**,在电平方式下,当采集到INT0,INT1的有效信号为**低电平**激活外部中断.
12. 填空题
13. **串行口的发送缓冲器和接收缓冲器使用同一个特殊功能寄存器名SBUF,并具有相同的字节地址99H,他们是同一个寄存器吗？串行口的读,写SBUF指令分别操作的是哪个缓冲器？**答：串行数据寄存器SBUF包含在物理上试隔离的两个8位寄存器:发送书记寄存器和接受数据寄存器，但是它们共用一个地址99H，格式如下



读sbuf(mov a,sbuf):接收数据寄存器;

写sbuf(mov sbuf,a):发送数据寄存器.

1. **ARM协处理器指令包括哪三类,请描述它们的功能.**答:**(1)**ARM处理器用于初始化协处理器的数据操作指令(CDP)**(2)**协处理器寄存器和内存单元之间的数据传送指令(LDC,STC)**(3)**ARM处理器寄存器和协处理器寄存器之间的数据传送指令(MCR,MRC)
2. **ARM920T体系结构支持哪两种方法储存字数据？答：大端格式:**字数据的高字节存储在低地址,而字数据的低字节则存储在高地址中:**小端格式:**与大端存储格式相反,在小段存储格式中，低地址中存放的是字数据的低字节，高地址存放的是字数据的高字节(小端格式是ARM处理器的默认格式)**。**
3. **简述嵌入式操作系统的基本特点.答: (1)**小型化与有限资源**(2)**与应用米歇相关**(3)**系统软,硬件协同设计**(4)**需要交叉开发环境和调试工具
4. **程序分析题**

1.下各条指令其源操作数的寻址方式是什么？各条指令单独执行后,A中的结果是什么？设(60H)=35H, (A)=19H, R0=30H, (30H)=0FH.

(1)MOV A ,#48H; 寻址方式: **立即寻址** (A)= **48H**

(2)MOV A,#60H; 寻址方式: **直接寻址** (A)= **5EH**

2.执行下列程序后,（A）=**40H**,(B)= **01H**.

MOV A,#0AH

MOV B,#20H

MUL AB

3.请分析下面程序执行后的操作结果,(A)= **40H**,(R0)= **60H**.

MOV A,#60H

MOV R0,#40H

MOV @R0,A

MOV 41H,R0

XCH A,R0

4.设在31H单元存有#23H,执行下面程序后（41H）=**01H** （42H）=**10H**

MOV SP,#60H

MOV A,#10H

MOV B,#01H

PUSH A

PUSH B

POP A

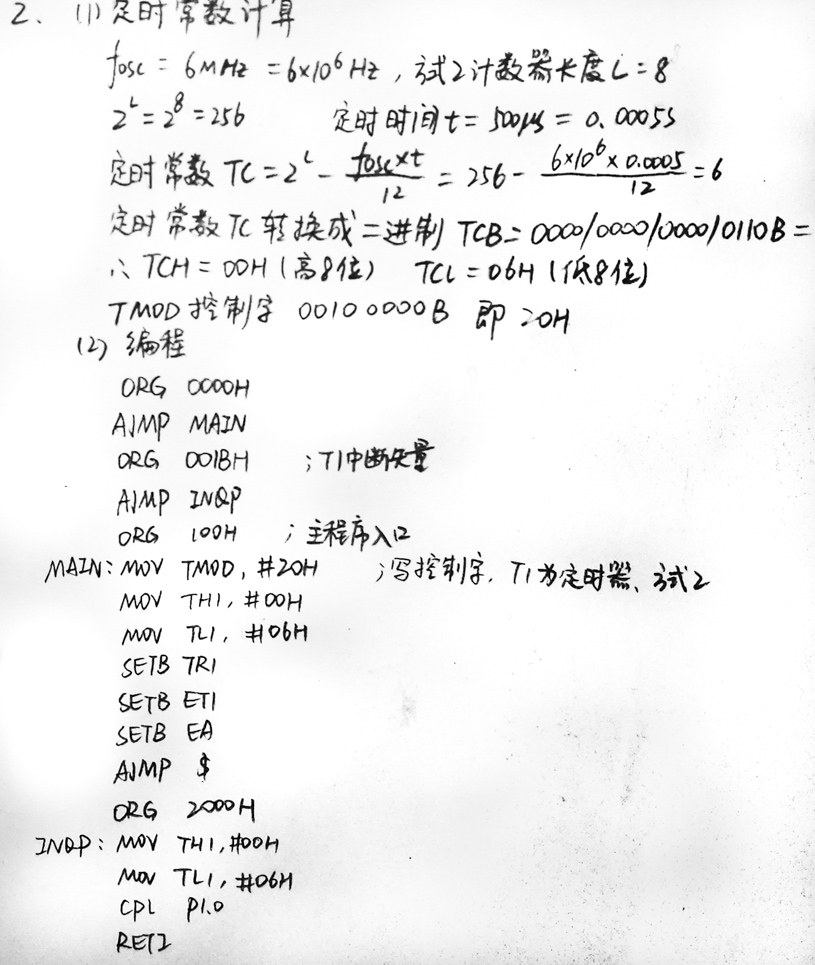
POP B

1. 假设R0的内容为0x104,寄存器R1,R2的内容分别是0x01与0x10,储存器所有单元初始内容为0.连续执行下述指令后,储存器中内容变为非0的地址为**R0**和**R1**,其内容分别为**0x8010**和**0x10**.

STMIB R0!,{R1,R2}

LDMIA R0!{R1,R2}

1. **编程题**
2. 编写3字节二进制加法程序,程序入口为:加数:22H,21H,20H三字节,22H为最高位；被加数:32H,31H,30H三字节,32H为最高位；程序出口为23H,22H,21H,20H四字节,23H为最高位.
3. 已知晶振频率为6MHz,在p1.0引脚上输入周期为500微妙的等宽矩形波,若T1中断,工作方式2,试写出中断初始化程序.



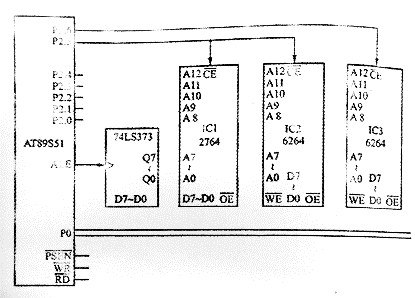
1. **综合应用题**

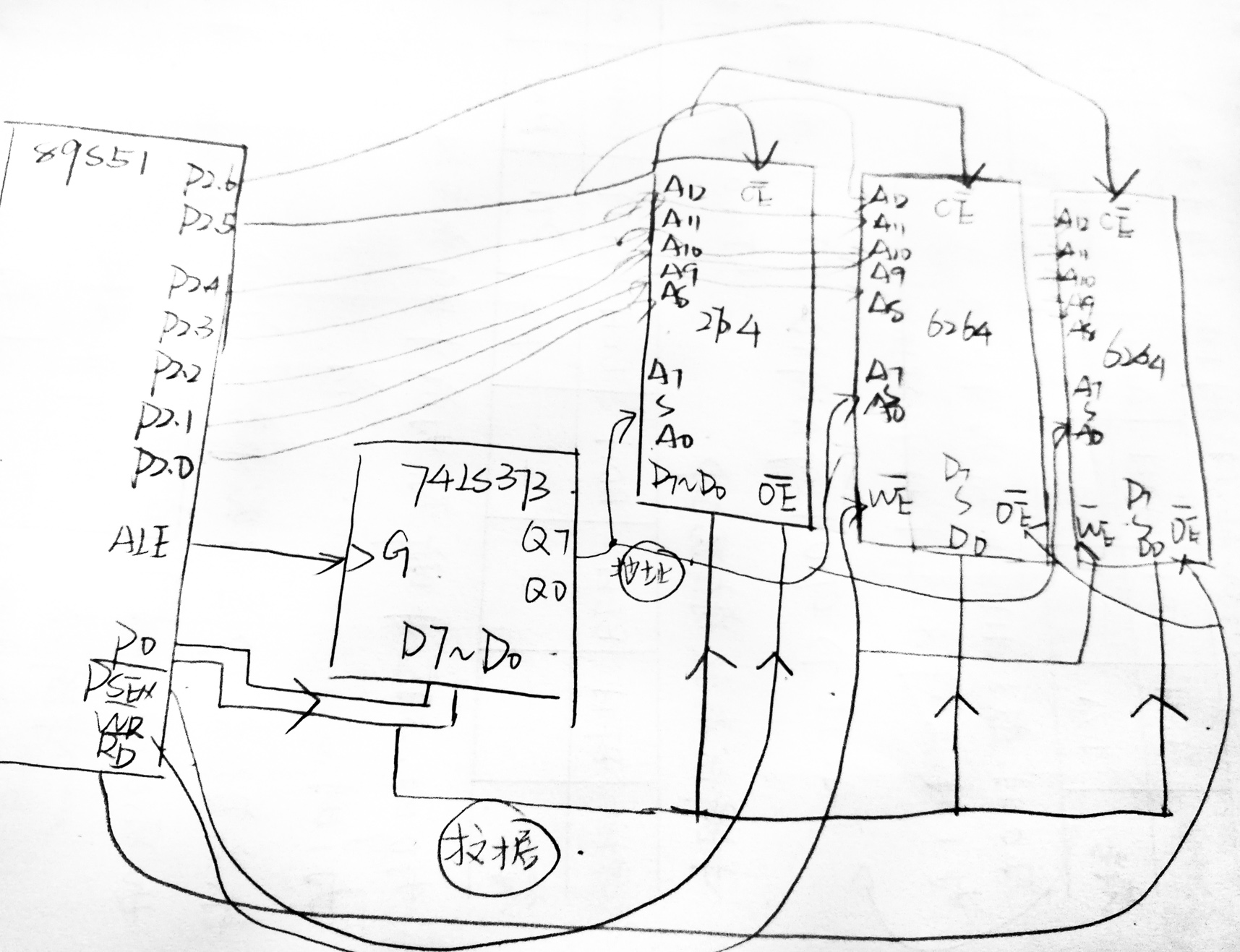
如图所示,现有AT89C51单片机,74LS373锁存器,一片264EPROM和两片6264RAM,请用它们组成一个单片机应用系统,要求如下:

（1）画出硬件电路连接图,并标注主要引脚

（2）指出该应用系统程序储存器空间和数据储存器空间鸽子的地址范围

（3）编写程序实现将外部RAM A000H~A00F单元内容传送到内部RAM的40H~4FH单元中.





**（1）编写块搬移程序**

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN:

MOV R2,#10H

MOV R0,#30H

MOV DPTR,#8000H

LOOP:

MOV A,@R0

MOVX @DPTR,A

INC R0

INC DPTR

DJNZ R2,LOOP

SJMP $

END

**2）在片外RAM找出最大数**

ORG 0000H

MOV R0,#10H

MOV DPTR,#1000H

MOV 20H,#0

LP:

MOVX A,@DPTR

CJNE A,20H,CHK

SJMP LP1

CHK:

JC LP1

MOV 20H,A

LP1:

INC DPTR

DJNZ R0,LP

SJMP $

END

（**3）采用多重循环方式编写延时1s的子程序，并调用该子程序使得P1.0口上的LED灯闪烁**

ORG 0000H

SJMP MAIN

ORG 0030H

MAIN:

MOV A,#0

LOOP:

MOV P1,A

LCALL DELAY

CPL A

SJMP LOOP

DELAY:

MOV R1,#100

LOOP1:

MOV R2,#100

LOOP2:

MOV R3,#50

DJNZ R3,$

DJNZ R2,LOOP2

DJNZ R1,LOOP1

RET

END

**（1）编写串口Echo程序，即串口接收到的任何字符马上转发出去。**

**实验程序如下：**

ORG 0000H

SJMP START

ORG 0030H

START:

MOV TMOD,#20H ;T1工作于方式2,8位自动重载计数

MOV TL1,#244 ;F=11.0592MHz，波特率4800

MOV TH1,#244

SETB TR1

MOV SCON,#11000010B ;串口工作方式3

SETB REN

LOOP:

JNB RI,$

MOV A,SBUF

CLR RI

CLR TI

MOV SBUF,A

JNB TI,$

CLR TI

SJMP LOOP

END

**（2）编写一串字符串“hello world ”的程序，采用奇校验。**

ORG 0000H

SJMP START

ORG 0030H

TABLE:DB "hello world!"

START:

MOV DPTR,#0030H

MOV R0,#0

MOV R7,#12

MOV TMOD,#20H

MOV TL1,#244

MOV TH1,#244

SETB TR1

MOV SCON,#11000010B

LOOP:

CLR TI

MOV A,R0

MOVC A,@A+DPTR

MOV C,P

CPL C

MOV TB8,C

MOV SBUF,A

JNB TI,$

INC R0

DJNZ R7, LOOP

END