TH0 溢出时产生 T1 中断（设置 TF1=1)

**RI**:当 CPU 从串口接收到完整的一帧后;

INC DPTR；DJNZ R2 LOOP;**(结束)**

### ARM 的通用寄存器(存储器),

乘/除法指令中用作 ALU 的一个输入.

**复位:**单片机的初始化操作,把 PC 初始化

例:设(R0)＝30H,(A)＝3FH,片内(30H)＝

若对引脚下降沿计数时；TL0 对 T0 脚计 硬件自动把本标志置 1,表示 CPU 已接收 **4. 将连续存放在外部数据存贮器**

**R12,R13,R14?**在 ARM 状态下,通用寄存

乘法的两个输入为 A,B,运算结果,A 中放

为 0000H,单片机从 0000H 单元开始执

BBH.

数；T1 停止工作。

### 程序：

到一个字符,此时应把数据及时取出

(MOV A , SBUF),并把 RI 清 0(软件,否则

### 1000H 开始的 10 个无符号数求和，结果为双字节，存放在 1100H（低位），

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**

积的低 8 位,B 中放积的高 8 位. 行.

除法中,被除数取自 A,除数取自 B,商数存 **⑴ 复位信号**

执行指令 XCH A,@R0

### 4)节交换指令组

**利用 T0 定时，每隔 2ms 使连接在 P1.0** 即使有新的数据也不接收).

### 1101H（高位）中。

**寄存器** R8~R14.R12 为写入寄存器,用作

放于 A,余数存放于 B

RST 引脚: 高电平有效,有效时间应持续 ⑴ XCHD A,@Ri

### 引脚上的 LED 闪烁一次。(12MHZ)

**TI:**执行 MOV SBUF,A 指令后,串口自动发 MOV R6,#0(低位)**;**MOV R7,#0(高位)**；**

子程序间的中间结果寄存器,记录着

### 4. 程序状态字 PSW (Program

2 个机器周期以上

Ri 间接寻址单元的低 4 位与累加器 A 的

**(查询方式)**ORG 0000H**；**SJMP MAIN**；**

送数据位,当发送完成后,硬件自动把 TI

MOV DPTR,#1000H**；**MOV R2,#10**；**

IP.R13 通常在 ARM 指令中用作堆栈指

### Status Word)

**(2)复位方式:** 上电自动复位,按键电平复 低 4 位互换,而高 4 位不变.

ORG 0030H

标志置 1,表示串口已空闲,可以发送下一 **LOOP:** MOVX A,@DPTR**；**ADD A,R6**；**

针,简称 SP.R14 用作子程序返回指针寄

8 位寄存器,内容是算术逻辑运算单元

位和外部脉冲复位

例:设(R0)＝20H,(A)＝

**MAIN:**MOV TMOD,#00000000B**；**

个数据.此时应把 TI 清 0(软件)并发送新

JNC NORMAL**；**INC R7**；**

存器,称为链接寄存器(LR):可以保存每一

(ALU)的输出

### 程序执行

36H(00110110B),(20H)＝

MOV TH0,#11100000B**；**MOV

数据。

**NORMAL:**MOV R6,A**；**DJNZ R2,LOOP**；** 种运行模式下子程序的返回地址 **3.程序**

### P－奇偶标志位

单片机的基本工作方式.

75H(0111010lB).

TL0,#00011000B**；**SETB TR0**；**

**PCON 寄存器：**(D7)SMOD:=0 波特率不 MOV DPTR,#1100H**；**MOV A,R6**；**

**寄存器** R15.

表示累加器 A 中值为 1 的个数的奇偶性:

复位后 PC＝0000H,程序执行从 0000H

执行指令:XCHD A,＠R0

**LOOP:** JNB TF0, LOOP**；**CLR TF0；

加倍；=1 波特率加倍。

MOVX @DPTR,A**;**MOV DPTR,#1101H**；**

* **ARM 的运行模式及特点?**用户(usr):正

若累加器值为 1 的位数是奇数,P 置位(奇 开始.

### ⑵ SWAP A

MOV TH0,#11100000B**；**MOV

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

**方式 0：**波特率=fosc/12

MOV A,R7**；**MOVX @DPTR,A**；**

常程序执行模式.系统(sys):运行操作系统 校验);否则 P 清除(偶校验) .

一般在 0000H 开始的单元中存放一条无

将累加器 A 的高,低半字节交换

TL0,#00011000B**；**CPL P1.0**；**SJMP

**方式 1：**波特率=（2^SMOD）\*T1 溢出

的特权任务.3.快中断(FIq):支持高速数据

如(A)=00001010,则 P=0.

条件转移指令,跳转到实际主程序入口去

例:设(A)＝36H(0011 0110B)

LOOP**；程序结束**

频率。

### 外扩的程序存储器和外扩I/O 端口地址重

传输及通道处理.中断(irq):用于通用的中 在串行通信中,常以传送奇偶校验位来检 执行.

执行指令: SWAP A

**(中断方式)**ORG 0000H**；**SJMP MAIN**；**

溢出频率=(fosc/12)\*[1/(2^8-TH1)]。

### 叠,80C51如何区分这些重叠地 址?

断处理.管理(svc):操作系统保护模式.中

验传输数据的可靠性.

**低功耗**

XCHD 和 SWAP 主要用于十六进制数或

ORG 000BH**；**SJMP T0ISR**；**ORG 0030H**; 方式 2：**SMOD=0 波特率=fosc/64

80C51 单片机对片外数据存储器,片内数 止(abt):用于支持虚拟内存和/或存储器

### OV －溢出标志位

**两种低功耗方式:待机方式和掉电保护方** BCD 码的数位交换.

**MAIN:** MOV TMOD,#00000000B**；** MOV TH0,#11100000B**；**MOV

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

SMOD=1 波特率=fosc/32

**方式 3：**同方式 1

据存储器及程序存储器采用不同的指令, 保护.未定义(und):支持硬件协处理器的会产生不同的控制信号.片外数据存储器 软件仿真.

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

**式.**

### ⒈ 待机方式

**5)操作指令组**

PUSH direct

TL0,#00011000B**；** SETB TR0**；**

**例程：**

有读 RD 和写 WR 控制信号,程序存储器

### 中断的执行与子程序相似点?

示的带符号数的范围(一 128～＋127).

⑴ IDL=1,80C51 进入待机方式

POP direct

SETB EA**；**SETB ET0

### 以 9600bps 的速率把片内 30H 开始的

有读 PSEN 控制信号,因此,扩展时虽然数 **1.**中断当前正在执行的程序.**2.**硬件把断

### RS1,RS0 －工作寄存器组选择位

振荡器仍运行,并向中断逻辑,串行口和

入栈指令: (SP) (SP)＋1,(SP) (direct)

**LOOP:** SJMP LOOP

### 16 个字符发送出去，要有奇校验位。

据线和地址线重复,但由不同的控制信号 点压栈,软件现场保护.**3.**通过软件恢复现 **(4)AC －辅助进位标志位**

定时器/计数器电路提供时钟,中断功能

出栈指令: (direct) (SP), (SP) (SP-1)

**T0ISR:**MOV TH0,#11100000B**；**

### (11.0592MHZ)

加以区别.片内数据存储器地址采用

场,重新返回到断点处,继续执行主程序.**4.** 加减法运算时,低 4 位向高 4 位数进位或 继续存在 .

例:中断响应时(SP)＝30H,DPTR 的内容为

MOV TL0,#00011000B**；**CPL P1.0**；**

ORG 0000H；SJMP MAIN；ORG 0030H; MOVC 指令,不会产生 RD 和写 WR 控制

二者可嵌套,如中断嵌套和子程序嵌套.

借位时,AC 将被硬件置位;否则,被清除.

向 CPU 提供时钟的电路被阻断,CPU 不 0123H,执行入栈指令:

RETI**；程序结束**

**MAIN:**MOV SCON,#11000010B；MOV

信号.

### 中断执行与子程序调用的差别?

在十进制调整指令 DA 中要用到 AC 标

工作,SP,PC,PSW,ACC 及通用寄存器冻结 ●**算术运算类指令**

**设计一实时时钟，能显示“时：分：秒：** PCON,#00H；MOV

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

* **INT0 的中断响应过程?**中断采样:中断

**1.**中断请求由外设发出,是随机的;子程序 志位状态.

在原状态.

助记符

### 十分秒”。以 BCD 码依次存放在 30H、

TMOD,#00100000B；MOV

采样是针对外部中断请求信号进行的,可 调用是编排好的.**2.**中断响应后由固定的

### CY －进位标志位.

⑵ 采用中断方式或硬件复位来退出待机 种:ADD,ADDC,INC,DA,SUBB,DEC,MUL,DI

### 31H、32H、33H。(晶振=6MHZ)

TH1,#0FDH；MOV TL1,#0FDH；SETB

以直接置位 TCON 或 SCON 中的中断请 矢量地址转入中断服务程序,而子程序地 在进行算术运算时,表示运算结果中高位 方式. V

HOUR EQU 30H**；**MIN EQU 31H**；**SEC

TR1；MOV R0,#30H；MOV R2,#16；

求标志;2,中断查询:若查询到某中断标志 址由软件设定.**3.**中断响应是受控的,其响 是否有进位或借位,可以被硬件置位或清

若产生一个外部中断请求信号,单片机 ●**逻辑运算类指令**

EQU 32H**；**TENSEC EQU 33H**；**ORG

**LOOP:**JNB TI,LOOP；CLR TI；MOV

为 **1,**则按优先级的高低进行处理,即响应 应时间会受一些因素影响;子程序响应时 除.

响应中断,PCON.0 位(IDL 位)被硬件自动

助记符 9 种:ANL, ORL, XRL, RL, RLC, RR,

0000H**；**SJMP MAIN**；**ORG 000BH**；**

A,@R0；MOV C,P；CPL C；MOV

中断;**3,**中断响应:首先将当前程序计数器 间是固定的.

### F0 －用户标志位.

清“0”, 单片机退出待机方式进入正常工

RRC, CPL, CLR.

SJMP T0ISR**；**ORG 0030H**；**

TB8,C；MOV SBUF,A；INC R0；DJNZ

PC 的内容压入堆栈进行保护,再将对应

### 程序存储器的特点 ?

开机时该位为“0”.

作方式.

逻辑“与”助记符:**ANL**,符号“∧”表示:

**MAIN:**MOV SP,#40H **;**MOVTMOD,#01H**;** R2,LOOP；SJMP $（**结束**）（**以上是查询** 中断源的中断矢量地址装入 PC,执行中

**1,**存放经调试正确的应用程序和表格之

用户可根据需要,通过位操作指令置“l”或 **⒉ 掉电保护方式**

逻辑“或”助记符:**ORL**, “∨”表示:

MOV TH0,#3CH**；**MOV TL0,#0B0H**；**

**方式**）（**下面是查询+子程序调用**）

断服务程序.运行直到遇到 RETI 指令为

类的固定常数;

者清“0”.

⑴ PD 位控制单片机进入掉电保护方式 逻辑“异或”助记符为 **XRL**, “⊕”表示:

SETB TR0**；**SETB EA**；**SETB ET0**；**

ORG 0000H**；**SJMP MAIN**;**ORG 0030H**；** 止,最后恢复原程序的断点地址执行,且恢 **2,**采用 16 位的程序计数器PC 和 16 位的地

### 时序定时单位: 节拍,状态,机器周期和

当 80C51 检测到电源故障时,进行信息 循环右移指令:**RR** A(累加器内容逐位循

**LOOP:** LCALL DISP(显示)**；** SJMP LOOP**; MAIN:**MOV SCON,#11000010B**；**

复中断触发器原先状态.

址总线,可扩 展的地址 空间 为64 KB;

### 指令周期.

保护,把 PCON.1 位置“1”,进入掉电保护

环右移一位,a0 移到 a7,不影响标志位)

**T0ISR:**PUSH PSW**；**PUSH ACC**；**

MOV PCON,#00H**；**

1、**RISC 的特点?指令集:**减少了指令种

**3,**64KB 地址是空间连续且统一的.

(1)节拍 P:振荡脉冲的周期称为节拍.

方式.

带进位循环右移指令:**RRC** A(将累加器

MOV TH0,#3CH**；**MOV TL0,#0B0H**；** INC TENSEC**；**MOV A,TENSEC**；**

CJNE A,#0AH,DONE**；**MOV TENSEC,#0**;**

MOV TMOD,#00100000B**；**MOV TH1,#0FDH**；**MOV TL1,#0FDH**；** SETB TR1**；**MOV R0,#30H**；**MOV

类,每条指令的长度是固定的,指令格式和 ●●**单片微型计算机原理与接口技术**●● 寻址模式相当少,一个周期就可以执行一 ●**2.3.1 中央控制器**是识别指令,并根据

条指令;**2,流水线:**指令的处理过程被拆分 指令性质控制计算机各组成部件进行工

(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.

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

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

INC SEC**；**MOV A,SEC**；**DA A**；**

R2,#16**；**

成几个更小的.能够被流水线并行执行的 作的部件,与运算器一起构成中央处理器 一个机器周期有 12 个振荡脉冲周期.是

当 Vcc 恢复正常后,只要硬件复位信号维 循环左移一位,a7 移到 a0.不影响标志位)

MOV SEC,A**；**CJNE A,#60H,DONE**；**

**LOOP:**MOV A,@R0**；**LCALL

单元,允许流水线在当前指令译码器阶段 **功能:**控制指令的读出,译码和执行,对指

单片机的最小时间单位.

持 10ms,就能使单片机退出掉电保护方

带进位循环左移指令:**RLC** A(累加器的内

MOV SEC,#0**；**INC MIN**；**MOV A,MIN**；** SENDCHAR**；**INC R0**；**DJNZ R2,LOOP**；** 去取其下一条指令;**3,寄存器**:RISC 拥有更 令的执行过程进行定时控制,并根据执行 (4)指令周期: 执行一条指令所需要的时 式,CPU 则从进入待机方式的下一条指令 容和进位位一起循环左移一位,a7 移入进

DA A**；**MOV MIN,A**；**CJN A,#60H,DONE SJMP $**；**

多的通用寄存器,每个寄存器都可以存放 结果决定是否分支转移.

间.是最大的时序定时单位.80C51 的指令 开始重新执行程序.

位位 **CY**,CY 的内容移到 a0)

MOV MIN,#0**；**INC HOUR**；**MOV

**SENDCHAR:WAIT:**JNB TI,WAIT**;**CLR TI**；** 数据或地址,寄存器可为所有的数据操作 **组成:**程序计数器 PC,数据指针 DPTR,指

周期有 1,2,4 个机器周期.

### 编程和校验.

累加器按位取反指令:**CPL** A(累加器的

A,HOUR**；**DA A**；**MOV HOUR,A**；**

MOV C,P**；**CPL C**；**MOV TB8,C**；**

提供快速的局部储存访问;**4,Load-Store** 令寄存器 IR,指令译码器,条件转移逻辑

### ●单片机存储器的两种基本结构:

* + 1. **寻址方式**

内容逐位取反,结果仍存在 A 中.不影响

CJNE A,#24H,DONE**；**MOV HOUR,#0**；**

MOV SBUF,A**；**RET（**结束**）

**结构**:处理器只处理寄存器中的数据.独立 电路及定时控制逻辑电路.

### 普林斯顿(Princeton)结构:程序和数

**寻址方式:指令中给出的寻找操作数或操** 标志位)

**DONE:**POP ACC**；**POP PSW**；**RETI

**从串口接收数据，存放在片内 30H-3FH** 的 load 和 store 指令用来完成数据在寄

* **程序计数器 PC:**一个独立的计数器,是

据合用一个存储器空间

### 作数地址的方法.

累加器清 0 指令:**CLR** A(对累加器进行清

### 汽车轮子周长 3 米，轮子每转一圈，即

**的单元内，并检查偶校验位，波特率为**

存器和外部存储器之间的传送(**5,**充分利

中央控制器中最基本的寄存器.

* + - 1. **哈佛(Harvard)结构:**程序存储器和数

### ⒈ 立即寻址

0,不影响标志位)

### 往 T0 引脚发送一个脉冲，汽车行驶的

**2400bps。(11.0592MHZ)**

用 VLSI 芯片的面积;**6,**提高计算机运行速 **内容:指令地址.**工作过程:PC 变化的轨迹

据存储器截然分开,分别寻址的结构.

在指令中直接给出操作数,出现在指令

### 控制程序转移类指令

**里程数以压缩的 BCD 码存放在 30H**

ORG 0000H**；**SJMP MAIN**;**ORG 0023H**；** 度;**7,**便于设计,可靠性高;**8,**有效支持高级 决定程序的流程.PC 的宽度决定了程序

### 4 个物理存储器空间

中的操作数称为立即数.

助记符 12 种:AJMP, LJMP, SJMP, JZ, JNZ,

### （高位）、31H（低位）单元内。(晶振

**=12MHZ)**

MILEHIGH EQU 31H**；**MILELOW EQU 30H**；**ORG 0000H**；**SJMP MAIN**；**ORG

000BH**；**SJMP T0ISR**；**ORG 0030H**；**

**MAIN:** MOV SP,#40H**；**

SJMP COMISR**；**ORG 0030H**； MAIN:**MOV SCON,#11010010B**；** MOV PCON,#00H**；**

MOV TMOD,#00100000B**；**MOV TH1,#0F4H**；** MOV TL1,#0F4H**；**SETB TR1**；**SETB EA**；**SETB ES**；**

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

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.

**·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

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

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

MOV TMOD,#00000100B**;**

MOV R0,#30H**；**SJMP $**；**

前程序状态寄存器 CPSR 的状态,以便从

**主要功能:用作片外数据存储器或 I/O 寻** 空间;**③**片外 64 KB 的数据存储器地址空 **方法**

指令第 1 字节的高 3 位 a10～a8 和指令

MOV TL0,#00010011B ;SETB TR0**；** SETB EA**；**SETB ET0**；**

**LOOP:** LCALL DISP(显示)**；**SJMP LOOP**;**

**COMISR:**PUSH PSW**；**PUSH ACC**；** CLR RI**；**MOV A,SBUF**；**JB P,BIT1**；** JNB RB8,OK**；**SJMP EXIT**；**

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

### 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

间.

### 三种基本寻址空间:

* 64 KB 的片内,外程序存储器寻址空间;

### 3.寄存器内容是操作数

例如:指令 INC R0

### 4．寄存器间接寻址

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

**T0ISR:**PUSH PSW**；**PUSH ACC**；**

**BIT1:**JNB RB8,EXIT**；**

两种方法存储字数据,称之为大端格式和

MOVX ＠DPTR,A 写

* + 64 KB 的片外数据存储器寻址空间;

寄存器内容是操作数地址.示形式:＠寄存 令第 2 字节和第 3 字节组成.目标地址为

MOV TH0,#11110101B**；**MOV

**OK:**MOV @R0,A**；**INC R0**；**

小端格式.**大端格式:**字数据的高字节存储 既可以作为一个 16 位寄存器处理,或者

* 256B 的片内数据存储器寻址空间,包括 器符号

64KB 空间)

TL0,#00010011B**；**MOV A,MILELOW**；**

**EXIT:**POP ACC**；**POP PSW**；**RETI**(结束)**

在低地址中,而字数据的低字节则存放在 两个 8 位寄存器,其高 8 位用 DPH 表示,

SFR 寻址空间.

例如:指令 ANL A,@R1

间接转移指令:**JMP** @A+DPTR(目标地址

INC A**；**DA A**；**MOV MILELOW,A**；**

**●**●●程序例子●●●

高地址中.**小端格式**:与大端储存格式相

低 8 位用 DPL 表示.

### 2.4.1 程序存储器

**5. 相对寻址**

是累加器 A 中的 8 位无符号数与数据指

MOV A,MILEHIGH**；**ADDCA,#0**；**

**1.把在 ROM 中 300H~3FFH 单元中最大** 反,在小端储存格式中,低地址中存放的是 ●**PC 与 DPTR 不同之处:**

功能:存放程序和固定常数.

指令给出的操作数为程序转移的偏移量. 针 DPTR 的内容相加)

DA A**；**MOV MILEHIGH,A**；**

### 的数找出来存放在内部 RAM 的 30H 单

字数据的低字节,高地址存放的是字数据 ⑴ 与地址有关的 16 位寄存器

PC 和地址总线为 16 位 ,可扩展的地址

在相对转移指令中,给出地址偏移量.

累加器判零转移指令:

POP ACC**；**POP PSW**；**RETI**（结束）**

### 元，遇零则停止。

的高字节.

PC:程序存储器的地址; DPTR:数据存储

空间为 64 KB.

目的地址＝(转移指令所在地址＋转移 **JZ** rel;(若(A)=0,则(PC)=(PC+2)+rel;若

### 设计数据采集程序，每隔 200us 读取

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

MOV DPTR,#300H**；**MOV R2,#0**；**

* **程序存储器和数据存储器地址冲突,如** 器或 I/O 的地址.

### ⒈片内和片外程序存储器

指令字节数)＋rel

(A)≠0,则(PC)=(PC)+2)

### P1 口的数据，依次存放在外部 RAM 的

MOV B,#0**；**

**何区分?**不发生数据冲突的**原因**是:MCS-

PC 的输出与 ALE 及 PSEN 有关;DPTR

EA 引脚高电平,从片内程序存储器

如:指令 JC 80H

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

### 1000H-10FFH 的内存单元。(晶振

**LOOP:**CLR A**；**MOVC A,@A+DPTR**；**

51 中访问程序存储器和数据存储器指令 输出与 ALE,WR,RD 信号有关.

0000H 开始执行;

### 6．变址寻址

(A)=0,则(PC)=(PC)+2)

### =12MHZ)

JZ ZERO**；**CJNE A,B,NOTSAME**；**

不一样;程序存储器访问指令为 MOVC;

⑵ PC 为 16 位寄存器,不可访问.DPTR 为

当 PC 值超出 4K,自动转向片外程序存

以 DPTR 或 PC 为基址寄存器,累加器

数值比较转移指令:**CJNE** A, direct, rel (指

PTRH EQU 30H**；**PTRL EQU 31H**；**

SJMP NEXTLOOP**；**

数据存储器访问指令为 MOVX;选通信号 16 位寄存器,也可作两个 8 位特殊功能

储器空间执行.

A 为变址寄存器,以两者内容相加后形成

令格式为: CJNE(操作数 1),(操作数 2),rel

ORG 0000H**；**SJMP MAIN**；** ORG

**NOTSAME:** JC NEXTLOOP(A<B)**；**

不同,前者为/PSEN,后者为/WR 与/RD.

寄存器,可访问.

EA 引脚低电平,从片外程序存储器

的 16 位程序存储器地址作为操作数地

数值比较指令的第 1 字节为操作码(或操

000BH**；**SJMP T0ISR**；**ORG 0030H**；**

**MAIN:**MOV TMOD,#00000010B**；**

XCH A,B(A>B)**；**

**NEXTLOOP：**INC DPTR**；**

* **中断响应的现场保护?**所谓现场是指中 ●**3. 指令寄存器 IR,指令译码器及控制**

断发生时单片微机中存储单元,寄存器,特 **逻辑**

0000H 开始执行.

### ⒉ 程序入口地址(中断)

址.

称基址寄存器＋变址寄存器间接寻址.

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

MOV TH0,#56**；**MOV TL0,#56**；**SETB TR0**；**SETB EA**；**SETB ET0**；**

DJNZ R2,LOOP**；**

**ZERO:** MOV 30H,B**；（结束）**

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

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

系统复位后 PC 为 0000H,系统从 0000H

单元开始取指,执行程序.

例 如 : MOVC A,＠A＋DPTR

### 7．位寻址

大小比较的功能.)

循环转移指令:**DJNZ** Rn,rel

MOV PTRH,#10H**；**MOV PTRL,#00H**；**

1. **把内部 RAM 的 30H~7FH 单元分别** 场的功能.在 80C51 单片微机中,现场一

送定时控制逻辑电路.

0003H～002DH 用于 5 个中断源的中断

寻址范围:

DJNZ direct,rel

SJMP $**;**

**T0ISR:**PUSH PSW**；**PUSH ACC**；**PUSH

### 和外部 RAM 的 50H~9FH 单元相加送入外部 RAM 的 1030H~107FH 单元

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

### 2.3.2 运算器

**组成:**算术逻辑运算单元 ALU,累加器 A,

服务程序入口地址.

**P0 口:**多功能 8 位口,字节访问地址: 80H,

1)片内 RAM 位寻址区: 如 MOV C,2BH

每执行一次本指令,先将操作数减 1,判别是否为 0.

DPH**；**PUSH DPL**；**MOV TH0,#56**；**

MOV R0,#30H**；**MOV R1,#50H**；**

程序相同,可以有以下几种:**1.**通过堆栈操 暂存寄存器,B 寄存器,程序状态标志寄存 位访问地址:80H～87H.

### 数据传送类指令

◇不为 0,转向目标地址;

MOV TL0,#56**；**MOV A,P1**；**MOV

MOV DPTR,#1030H**；**MOV R2,#50H**；**

作指令 PUSH direct.**2.**通过工作寄存器区 器 PSW 以及 BCD 码运算修正电路等

### ⒉ P0 口的功能

**指令助记**

◇为 0,则结束循环程序,程序往下执行.

DPH,PTRH**；**MOV DPL,PTRL**；**MOVX

**LOOP:**MOV P2,#00H**;**MOVX A,@R1**；**

的切换.**3.**通过单片微机内部存储器单元

### 1.算术逻辑运算单元 ALU

**⑴ I／O 口 :**输出锁存,输入缓冲,输入时

**符:**MOV,MOVX,MOVC,XCH,XCHD,SWAP 绝对调用指令:**ACALL** addr11(无条件地

@DPTR,A**；**INC PTRL**；**POP DPL**；**POP

ADD A,@R0**；**MOVX @DPTR,A**;**INC

暂存.

### ALU 有两个输入:

需先将口置 1;每根口线可以独立定义为

,PUSH,POP

调用首址为 addr11 处的子程序.操作不

DPH**；**POP ACC**；**POP PSW**；**RETI **结束**

R0**;**INC R1**;**INC DPTR**;**DJNZ R2,LOOP**；**

### ARM 处理器异常中断响应过程?当异

1)暂存器 1 的输入: 2)暂存器 2 或累加器 输入或输出.

1)A 内容送外部数据存储器或 I/O

影响标志位)

### ●●串口部分：SCON，PCON，SBUF SCON：

1. **把外部 RAM 中从 1234H 开使的**

**200 个字节送至外部 RAM 的 5678H 开**

常中断发生时,处理器挂起正常模式的执 A 的输入

行,首先自动保存当前状态,即返回地址存 **ALU 有两个输出:**

**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

MOVX @Ri,A MOVX @DPTR,A

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

**SM0/1(D7 D6 组合):** 00 方式 0；01 方 **始的单元中去。**

入链接寄存器 R14,当前程序状态寄存器

1)通过内部总线送回累加器 A;

总线用时,输出低 8 位地址 A0～A7.

2)存储器数据传送指令(或查表指令)

响标志位)

式 1；10 方式 2；11 方式 3。

MOV DPTR,#1234H**；**MOV R7,#56**；**

CPSR 存入 SPSR 中,然后进入相应的工作 2) 标志位输出至程序状态字 PSW.

**P2 口的工作状态是**输出高 8 位地址.

MOVC A,＠A＋PC

子程序返回指令:**RET**(表示结束子程序,返

**SM2：**=0 接收数据帧/地址帧;=1 只接收地址帧(第九位数据为 1)

MOV R6,#78**；**MOV R2,#200**；**

**LOOP:**MOVX A,@DPTR**;**PUSH DPH**；**

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

### 累加器 A

**主要功能:**存放操作数,暂存运算结果.

### P2 口的功能:

⑴ I／O 口

MOVC A,@A+DPTR

3)交换指令 XCH 组

回 ACALL 或 LCALL 的下一条指令(即断点地址),继续往下执行)

### REN:=0 不允许接收;=1 允许接收。

PUSH DPL**;**MOV DPH ,R7**;**MOV DPL,R6**；** 中断向量表的特定的地址范围内,中断向

单片机中大部分数据操作都要通过累

⑵ 程序存储器或片外数据存储器的高 8

XCH A,Rn ; (A) (Rn) ,direct,@Ri

中断返回指令:**RETI**(中断服务程序返回,

**TB8:**方式 2,3 中要发送的第 9 位数据.

MOVX @DPTR,A**;**INC DPTR**;**MOV R7

量表的入口是一些跳转指令,跳转到专门 加器 A 进行,容易产生“瓶颈”现象.

位地址

将累加器 A 与源操作数的字节内容互

从断点处继续执行,清除内部相应的中断

**RB8:**方式 2,3 中接收到的第 9 位数据

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| * **中断部分**寄存器:TCON,SCON,IE,IP。中 断 请 求 源 ： INT0(P3.2),INT1(P3.3),T0/T1(P3.4/P3.5)， TX/RX(P3.1/P3.0)   **TCON:定时器控制寄存器** | | | | | | | |
| T F  1 | T R  1 | T F  0 | T R  0 | I E  1 | I T  1 | I E  0 | I T  0 |
| TF1/0：T1/0 溢出出(=1 出)  IE1/0:INT1/0 中断标志；IT1/0:INT1/0 中断触发方式。  **SCON:串行口控制寄存器**  S S S R T R T R M M M E B B I I 0 1 2 N 8 8  TI:串行口发送中断请求标志。R1:串行口接收中断请求标志。  **IE 中断允许寄存器：**  E E E E E E  A S T X T X 1 1 0 0  EA:中断允许总控制位；ES:串行口中断允 许 ；ET1/0:T1/0 中 断 允 许 位 ； EX1/EX0: INT1/0 中断允许位。  **IP 中断优先级寄存器：**  P P P P P S T X T X 1 1 0 0  置 1 为高优先级，清 0 为低优先级  PS:串口；PT1/0：T1/T0；PX1/0:INT1/0.  默认优先级 INT0>T0>INT1>T1>串口**中 断 服 务 入 口 地 址 ：**INT0:0003H T0:000BH；INT1:0013H；T1:001BH；串  行口：0023H。   * **程序：**   **设计一个简单系统，当一个键按下时就点亮 LED 灯，再次按下时熄灭 LED 灯， 采用查询方式实现。**  KEY BIT P0.0**;**LED BIT P1.0**;**ORG0000H**；**  **MAIN:** JB KEY,MAIN**；**CPL LED**；**SJMP MAIN**；**END  **采用中断方式实现：**  LED BIT P1.0；ORG 0000H**；** SJMP MAIN**；**ORG 0003H**；**  SJMP INTO\_ISR**；**ORG 0030H**；**  **MAIN:** SETB IT0**；**SETB EA；SETB ET0； SJMP MAIN  **INTO\_ISR:** CPL LED**；**RETI；END  **设计系统，当一个键按下时把片外 RAM 中 1000H ~ 104FH 的 0 的个数送到片内 RAM 中的 20H 单元。**  ORG 0000H**；**SJMP MAIN**；**ORG 000BH**；**SJMP INT1\_ISR**；**ORG 0030H **MAIN:** MOV SP,#20H**；**SETB IT0**；** SETB EA**；**SETB ET0**；…；**  LCALL DISP**；…；**SJMP MAIN**；**  **中断程序**  **INT1\_ISR:** PUSH PSW**；**PUSH ACC**；** PUSH DPH**；**PUSH DPL**；**SETB RS1**；** MOV DPTR,#1000H**；**MOV R0,#0**；** MOV R2, #50H**；**  **LOOP:**MOVX A,@DPTR**；**JNZ NEXT**；；** INC R0**；**  **NEXT:**INC DPTR**；**DJNZ R2,LOOP**；** MOV 20H,R0**；**POP DPL**；**POP DPH**；** POP ACC**；**POP PSW**；**RETI**；**END**。**   * **定时器:TMOD,TCON,TH1/0,TL1/0**   **TMOD 方式控制寄存器**  G C M M G C M M A / 1 0 A / 0 1 T T T T  E E  D7-D4 为 T1 的；D3-D0 是 T0 的。  GATE:=0 与 INT0/1 无关;=1 可以计数。  C/T:=0 定时器；=1 计数器。  M1/M0：**00 方式 0；01 方式 1；10 方式 2；11 方式 3**  **TCON：寄存器见中断部分**  TF1/0：=0 无 T1/0 中断；=1 有中断。  TR1/0:=0 停止 T1/0 计数;=1 开始计数。  **方式 0:13 位计数器(T0 为例子)**  .TH0:计数值高 8 位;TL0 计数值低 5 位  C/T=0 每过 1 个**机器周期**,计数值+1 C/T=1 T0 管脚每来一次下降沿计数值+1 振荡周期=1/振荡频率 fosc  **机器周期=12\*振荡周期=12/fosc**  **需要定时时间= 2^13 – 当前计数值方式 1:16 位计数器**  **方式 2:自动重装 8 位计数器：**  TH0=TL0=计数值；T1 常常工作在方式  2，作为串行口波特率发生器。  **方式 3：**T0 拆成：8 位计数器/定时器  （TL0）；8 位定时器（TH0）；  TL0 溢出时产生 T0 中断（设置 TF0=1） | | | | | | | |

DPH**；**MOV R6,DPL**;**POP DPL**;**POP DPH**;**

处理某个异常或中断的子程序.

### 3．B 寄存器

* + **80C51 单片机的工作方式** 换.

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.

可以相互调用,只要遵循一定的调用规则 (3)**ASR** 算术右移; (4)**ROR** 循环右移;

* **Thumb 指令与 ARM 指令的时间效率** (5)**RRX** 带扩展的循环右移.

ORG 0023H AJMP SPINT

# ⑴ I/O 口的数据传送指令

## 输出数据的指令:

址 MOV R7,#08H;共需转换 8 个通道

读sbuf(mov a,sbuf):接收数据寄存器;

写sbuf(mov sbuf,a):发送数据寄存器.

### 80C51 有 5 个中断源:

* 两个外部中断源 INT0 和 INT1
* 三个内部中断源

### 外部中断

**两种信号触发方式:**

**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

例: MOV R0, R1, LSL #5; R0=R1 逻辑左

移 5 位

**4.寄存器间接寻址 LDR** R0,[R2]:将 R2 指向的存储单元内容保存到 R0.

**5.基址寻址 STR** R1,[R0,#-4]:先 R0=R0-

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

## 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

## 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

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

◆电平有效方式:若引脚上采样到有效的 快约 40～50％

4 然后把 R1 的值寄存到 R0.

CLR TI ;清除发送中断请求标志

位 清 0 指 令 CLR Px.y

## MOV @R1,A

令(LDC,STC)**(3)**ARM 处理器寄存

低电平,则向 CPU 提出中断请求;

使用 Thumb 代码,存储器的功耗会降低

### 6.多寄存器寻址

MOV A,＠R0 ;取数据,置奇偶标志

位 置 1 指 令 SETB Px.y

## INC DPTR;指向下一通道地址

器和协处理器寄存器之间的数据

◆跳变有效方式:若引脚上采样到有效负 约 30％

一条指令实现一组寄存器值的传送,连续 位

## 位 取 反 CPL Px.y

INC R1

传送指令(MCR,MRC)

跳变,则向 CPU 提出中断请求.

### ⑴INT0:外部中断 0.

* **状态切换方法**

ARM 指令集和 Thumb 指令集均有切换

的寄存器用“-”连接,否则用“,”分隔

例:LDMIA R0!, {R1-R5};

MOV C,P ;奇偶标志位 P 送 TB8 MOV TB8,C

## 位 为 1 转 移 JB Px.y, rel

位 为 0 转 移 JNB Px.y, rel

DJNZ R7,LOOP 8 个通道未转换完, 则继续。

1. **ARM920T 体系结构支持哪两种方法储存字数据？答：大端**

当 IT0＝0 时,低电平有效; 当 IT0=1 时,下降沿有效.

处理器状态的指令,并可在两种工作状态 功能:R1=[R0], R2=[R0+4],

之间切换,切换不影响处理器的运行模式 R3=[R0+8],R4=[R0+12], R5=[R0+16]

MOV SBUF,A ;数据写入发送缓冲器,启动发送

## 位为 1 转移并清零 JBC Px.y, rel

⑶ **I/O 口其它操作指令**

## D128μs:… ;延时 128μs 子程序

RET

PA 地址：7FFCH(A15=0，A1=0，

A0=0)

**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

### 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

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

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.相对寻址**

INC R0 ;数据地址指针加 1 CLR 0AFH ;恢复现场

POP ACC

POP PSW

## 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

PB 地址：7FFDH(A15=0，A1=0，

A0=1)

PC 地址：7FFEH(A15=0，A1=1，

A0=0)

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

当计数器发生计数溢出时,表明设定的定 Program Status Register),可以在任何工时时间到或计数值已满,这时可以向 CPU 作模式下被访问;程序状态备份寄存器

### 9.块拷贝寻址

* **ARM 指令分类说明**

SETB 0AFH

CLR PSW .4 ;切换寄存器工作组

## 减 1 指 令 DEC Px

减 1 条件转移指令 DJNZ Px, rel

控制寄存器地址;7FFFH(A15=0，

A1=1，A0=1)

端格式是 ARM 处理器的默认格式)**。**

申请中断.

80C51 有两个源,即:

SPSR (Saved Program Status Register) ,

只有在异常模式下,才能被访问;(各种异

### 1.数据处理指令

**2.跳转指令:**

RETI

### 采用延时等待 A/D 转换结束方式,分

数值比较转移指令 CJNE A, Px, rel

**例：对 8255A 各口作如下设置：A**

1. **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

⑴ TF0:T0 溢出中断.

常模式专有的)当异常发生时,SPSR 用于

### B 转移指令

**别对 8 路模拟信号轮流采样一次,并依次 例:执行指令 MOV P1,#7FH**

# 口方式 0，B 口方式 0，从 A 口输

用米歇相关**(3)**系统软,硬件协同设

⑵ TF1:T1 溢出中断.

### ⒊ 串行中断

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:跳转到目的地址.

跳转范围:当前指令的±32M 字节地址内

### 把结果存入数据存储器.

ORG 0000H

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

**入，从 B 口、C 口输出。**

工作方式控制字为 10010000，即

计**(4)**需要交叉开发环境和调试工具。

每当串行口发送或接收一帧串行数据时, **功能包括:**

(ARM 指令为字对齐,最低 2 位地址固定

SJMP MAIN

# 例:将一个数据字节从 74LS377 输出

## 90H。

1. 假设 R0 的内容为 0x104,寄存器

就产生一个中断请求.

RXD,TXD:串行中断.

保存 ALU 中的当前操作信息控制允许和禁止中断

为 0).

### BL 带链接的转移指令

ORG 0030H MAIN:MOV R1,#20H

## MOV DPTR,#7FFFH;地址指针指向74LS377

MOV A，#90H ; 设 A 口 、 B 口为方式 0

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

* **中断矢量:**当 CPU 响应中断时,由硬件

设置处理器的运行模式**.**

功能:PC 拷贝到链接寄存器,然后跳转到

MOV DPTR,#7FF8H ;指向通道 0 地 MOV A,#DATA;将输出数据送 A

## ;A 口输入， B 口、 C 口输出

续执行下述指令后,储存器中内容

产生一个固定的地址,即矢量地址,由矢量 ●**ARM 状态下寄存器的子集.Thumb 状** 指定地址. 址

## MOVX @DPTR,A;输出数据

MOV DPTR，#7FFFH

变为非 0 的地址为 **R0** 和 **R1**,其内

地址指出每个中断源的中断服务程序的入口.

### 态下寄存器和 ARM 状态下的寄存器组的关系:

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

MOV R7,#08H ;共需转换 8 个通道

LOOP:MOVX @DPTR,A;启动 A/D 转换

## MOVX @DPTR，A

MOV DPTR，#7FFCH ;从 A 口输入

容分别为 **0x8010** 和 **0x10**. STMIB R0!,{R1,R2}

* **中断响应时间:**在单级中断系统中,中断 R0~R7 是相同的

**功能:**跳转到目标地址;处理器工作状态切 ①

## MOVX A，@DPTR

LDMIA R0!{R1,R2}

响应时间:3→8 个机器周期.

CPSR 和所有的 SPSP 是相同的 换.

LCALL D128μs ;延时等待 A/D 转换

## MOV DPTR，#7FFDH ; 从 B 口 输

中断请求标志位查询占 1 个机器周期, Thumb 状态下的 SP 对应于 ARM 状态下 **目标地址:**寄存器 Rn 和 0xFFFFFFFE 相与 结束② 出

而且是指令的最后一个机器周期,在这个 的 R13

的结果.Rn 的第 0 位拷贝到 CPSR 中 T

MOVX A,@DPTR 读入 A/D 转换值

# 例:采集 8 路模拟量,并存入 20H 地址

## MOVX @DPTR，A

机器周期结束后,CPU 即响应中断,产生 Thumb 状态下的 LP 对应于 ARM 状态下 位,位[31∶1]移入 PC. ③

# 开始的内部 RAM 中。

## MOV DPTR，#7FFEH ; 从 C 口

硬件长调用 LCALL 指令,执行这条长调用 的 R14

指令需要 2 个机器周期,中断响应时间为 Thumb 状态下的程序计数器对应于

**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

MOV @R1,A

INC DPTR ;指向下一通道地址

# ⒉ 简单输入口的扩展

## ORG 0000H SJMP MAIN

输出

## MOVX @DPTR，A

3 个机器周期.

ARM 状态下的 R15

**功能:**跳转到目标地址;PC 值保存到 LR 寄

INC R1

## 数据输入需采用 8 位三态门控制电

ORG 0003H ;外部中断 0 入口地

**2. 片外并行数据存储器的扩展**

* **外部中断源的扩展:**80C51 只有两个外 ●**S3C2410A 的 DMA 控制器**

存器;处理器工作状态切换.

DJNZ R7,LOOP ;8 个通道未转换完则 路。 址

# 例：用两片 6264 扩展 16KB 片外并

部中断请求输入端 INT0 和 INT1,可以通过两种方式进行扩展:

### 可通过增加“OC 门”+ 软件来扩展;

1. **定时器/计数器**

●●●●**嵌入式设计及应用**●●●●

### RISC 的设计准则:

**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.

1. DMA 控制器向 CPU 发出总线请求信号
2. CPU 执行完总线周期,向 DMA 控制器

### 3.Load/Store 指令

**A.单寄存器的存取指令(LDR,STR) LDR Rd,[Rn]:** 把内存中地址为 Rn 的字数据存入 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

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

# 例:数据输入

## MOV DPTR,#0BFFFH;指向 74LS244

口地址

MOVX A,@DPTR;读入数据

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

**行数据存储器,采用线选法寻址：用一根口线 P2.7 来寻址。**

1.指令集:选用使用频率较高的一些 简单 发出相应的回答信号;

指令 复杂指令的功能由简单指令来组合 4.CPU 将控制总线,地址总线及数据总线

**LDR Rd,[Rn,Rm] ! ;**将内存中的地址为 SJMP MAIN

Rn+Rm 的字数据装入寄存器 Rd 中,并将 ORG 0003H ;外部中断 0 入口地址

## START:CLR F0 ;清中断发生标志

MOVX ＠DPTR,A;启动 A/D (P2.7=0,

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.

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;

新地址 Rn+Rm 写入 Rn

**LDR Rd,[Rn],Rm:**将内存地址为 Rn 的字装入 Rd,并将新的地址 Rn+Rm 写入 Rn

LJMP INTDATA

ORG 0100H ;数据采集程序MAIN:MOV R0,＃20H ;数据缓冲区首址

## /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

3.寄存器:拥有多个通用寄存器

6.在 DMA 控制器控制下进行 DMA 传送; **LDR{cond}T <Rd>,<addr>;**

MOV R2,＃8 ;8 通道计数器

## SETB EA ;开中断

**例:对 8255A 各口作如下设置:A 口方**

4.LOAD/STORE 结构:实现寄存器与外存之间的数据传送.

7.数据传送完,DMA 控制器通过中断请求 T:若指令有 T,即使处理器是在特权模式 MOV DPTR,＃7FF8H ;指向 0 通道线发出中断信号.CPU 接收到中断信号 下,也将存储系统访问看成是在用户模式 START:CLR F0 ;清中断发生标志

# 式 0,B 口方式 0,从 A 口输入,从 B

**口、C 口输出。**

## LOOP:JNB F0,LOOP;中断发生标志是否 为 0 ②

### RISC 架构的优点:

1.充分利用 VLSI 芯片的面积

后,转入中断处理程序进行处理; 的. MOVX ＠DPTR,A; 启 动 A/D (P2.7=0,

8.中断处理结束后,CPU 返回断点继续执 **STR:**将寄存器中的单个字或字节数据保 /WR=0) ①

## 工作方式控制字为 10010000,即

90H。

DJNZ R2,START;8 个通道转换是否结

束 SJMP MAIN

●●●●●●●●●●●●●

1.单片机的特殊功能寄存器区位于片

2.提高计算机运行速度:指令;通用寄存器; 行,并获得总线控制权

存到内存.

SETB IT0 ;置外部中断 0 为边沿触发

## MOV A,#90H;设 A 口、B 口为方式

INTDATA:MOVX A,＠DPTR;读数据

内从 80H 到 ffh 的地址范围,对特殊功

流水线

3.便于设计,可靠性高

### ARM 指令系统及编程技术

**ARM 指令的一般格式:**

**STRH:**半字数据存储指令

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

SETB EX0 ;允许外部中断 0 SETB EA ;开中断

## 0 ,A 口输入, B 口、 C 口输出

MOV DPTR,#7FFFH

(P2.7=0, RD 非=0),硬件撤销中断③

MOV ＠R0,A;存数据

能寄存器的访问只能采用变址寻址的

方式.

4.有效支持高级语言程序

<opcode>{<cond>}{S}<Rd>,<Rn>{,<op **STRT:**用户模式的字数据存储指令

LOOP:JNB F0,LOOP ;中断发生标志是 MOVX @DPTR,A

## INC R0

1. 当 T0 工作在方式 3 时,将 T0 分为

### 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

erand2 >}

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

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:

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

## 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

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**

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.

1. 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.

辑处理和移位处理

S:是否影响 CPSR 寄存器的值

IA 每次传送后地址加一

(P2.7=0, RD=0),;硬件撤销中断③

## MOVX @DPTR,A

**于 m 的数值作为不合格产品,则被剔** 3. 8051 单片机的工作寄存器分成 **4**

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:

Rd:目标寄存器

Rn:第 1 个操作数的寄存器

operand2:第 2 个操作数

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

MOV ＠R0,A ; 存 数 据INC R0

INC DPTR ;指向下一通道

# 例:产生锯齿波

## DAC0832 的地址为 7FFFH(P2.7=0)

ORG 0000H

**除掉。**

## 设数据组为 X1,X2,…,X10 共 10 个。

ORG 0000H

个组,每个组 **8** 个字节.

## 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**

指令的二级流水线结构:若取指和执行阶 ●**ARM 指令的条件域<cond>**

段时间上完全重叠,指令周期减半,速度提 **cond CPSR 中标志位 含义**

FD 满递减堆栈

ED 空递减堆栈

SETB F0 ;置中断发生标志

## RETI

SJMP MAIN

ORG 0030H

PX:CLR 00 ;设交换过标志

MOV R3,#09H;10 个数据比较,第一

个为状态寄存器.

1. ARM 状态下,SP 寄存器指的是

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.

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

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

## 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

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

**R13**,LR 寄存器指的是 **R14**,PC 寄存器指

## S3C410 中,其地址空间有 **8** 个

BANK,每个 BANK 最大的容量是

**126Mb**.

1. 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.

一 、 填 空 题

1. **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**

**特点:**密度高,所占存储空间较小;可以使 GT Z 清零且 N 等于 V 带符号数大于 中.

用 32 位的寻址空间,32 位寄存器,32 位 LE Z 置位或 N 不等于 V 带符号数小于 **4.程序状态寄存器指令**

## DEC R0

XCH AV@R0

**99H,他们是同一个寄存器吗？串**

**行口的读,写 SBUF 指令分别操作**

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:

或等于

AL 忽略 无条件执行

### 寻址方式:

1.寄存器寻址 MOV R1,R2

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.

# 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

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

**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:发送数据寄存

器和接受数据寄存器，但是它们

**ARM 状态,**此时处理器执行 32 位的字对 2.立即数寻址 MOV R0,#0XFF00

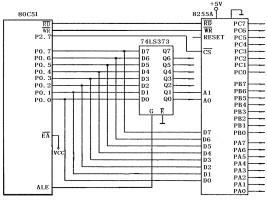
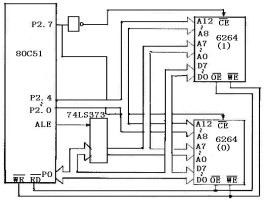
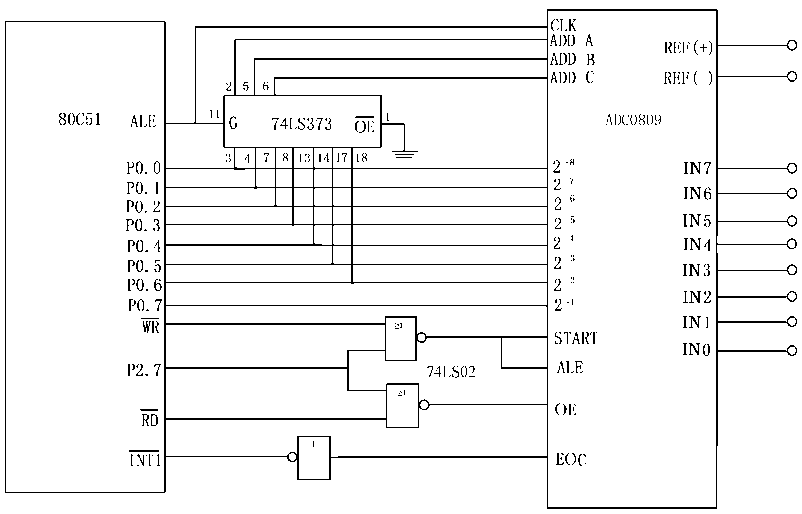
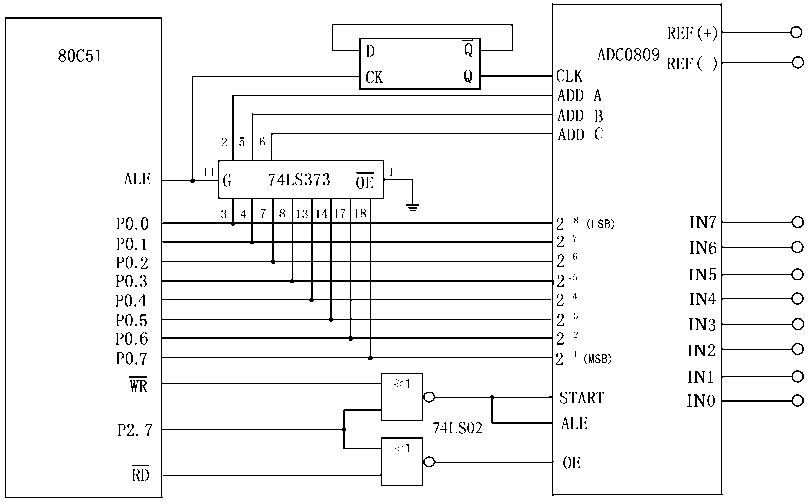
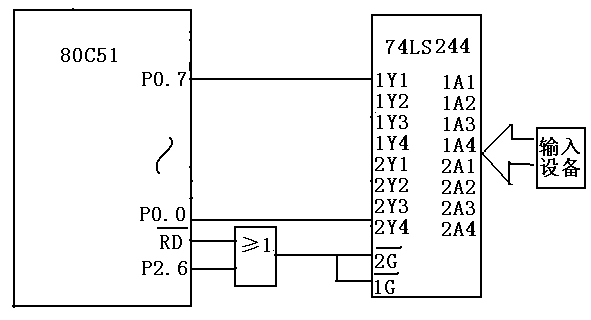
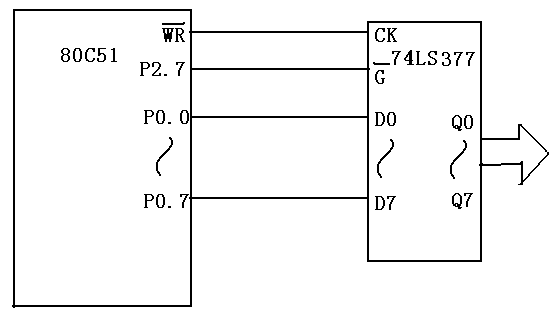
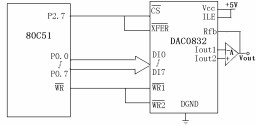
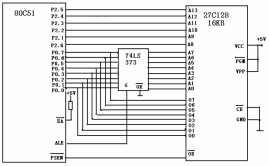
MRS 指令格式

## SJMP MAIN

END0:SJMP END0

共用一个地址 99H，格式如下

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.

### 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

MRS{cond} Rd,CPSR/SPSR

### 试编写串行接口以工作方式 2 发送数据的中断

ORG 0030H MAIN:MOV R1,#20H

MOV DPTR,#7FF8H ;指向通道 0 地

●●●●●●●●●●●●●●●

8255A 与 80C51 的接口及应用