RST 引脚: 复位输入信号,高电平有效,有效时间应持续2个机器周期以上

储商。 80C31 的 EA 必接地,它无片内程序存储器。 ALE/PROG:地址锁存允许信号,輸出。 在访问片处存储器或 I/O 时,用于锁存低 8 位 在仍问为外行隐蔽或 VO 的, 用于现行低 o 拉 地址, 以实现低八位地址与数据隔离, 由于 ALE 以 1/6 振荡频率的固定速率输出, 可作为对外 输出的时钟或作为外部定时时钟. 在 EPROM 類四即時代表記名 編程期间作輸入輸入編程脉冲(PROG) PSEN:片外程序存储区读选通信号低有效在 从片外程序存储器取指期间在每个机器周期

从斤个在1分仔面随来自由川。在時 1 70 mm 回 777 mm 当其有效,程序存储器内容被送上 PO D. ALE 和 PSEN 都可驱动 8 个 LSTTL 负载。
●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

^{在前足科电话} ●程序计数器 PC:一个独立的计数器,是中央

爾宁取田的指字的地址, 內**容:指今地址**,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.16 位可以寻址 64K范围

顺序指令自动+1 条件转移指令自动+1 条件转移指令或无条件转移指令。程序计数器 将被置入转移的目标地址程序流向发生变化 调用指令或响应中断:将子程序的入口地址 者中断矢量地址送入PC程序流向发生变化

●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 的地址寄存器 访问片外数据存储器或 I/O 的指令为

MOVX A。他行间的或 MOVX @DPTRA写 既可以作为一个 16 位寄存器处理。或者两个位 位寄存器 其高 8 位用 DPH 表示。低 8 位用 DPL表示。也可以作为访问程序存储器时的。常 址寄存器,这时寻址程序存储器中的表格。常 业寄存器,这时守址作力,Linex数等单元,不是寻址指令。 数等单元,不是寻址指令。 JMP @A+DPTR JMP @A+DPTR

●PC 与 DPTR 不同之处:

1)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. 2)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

●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 ●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

1.算术逻辑运算单元 ALU

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:

ALD 有例了辅助: 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 ア 累加器 A

2.条加爾 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. 2.n. 字容易产生"瓶颈"现象.

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, B 数取自 B, 商数存放于 A, 余数存放于 B

● 4 程序状态字 PSW (PSW 7->PSW 0) 8 位寄存器,内容是算术逻辑运算单元(ALU)的

(1) P - 奇偶标志位(PSW.0) 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.

期致循的可靠性。 (2) OV - 溢出标志位(PSW.2) 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

0 3 RS1,RS0(4,3) - 工作寄存器组选择位 0 组地址:00H~07H 1 组地址:08H~0FH 2 组地址:01H~07H 3 组地址:18H~1FH (4)AC -辅助进位标志位(PSW.6)

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 加減法运算时,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. 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

(5) CY - 进位标志位 (PSW.7) 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(PSW.5) 开机时该位为"0".用户可根据需要,通过位操作指令置"1"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。

S1~S6.(**1 个状态对应 2 个振荡周期**) 一个机器周期有 12 个振荡脉冲周期.是单片 振荡脉冲频率为 12MHz,则机器周期为 1us (4)指令周期: 执行一条指令所需要的时间

版入时时/アビリード-00001 1171 12.7-7, 1.2.4 个机器周期. (5)ALE 周期中包含 3 个状态 6 个节拍。

应用对象:ARM920T、MCS-51、80C51

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

字储器.数据存储器:③片内数据存储器:④片

3 门边相杆储备地址空间 D片内片外统一的 64 KB 程序存储器地址空间;②片外 2568 数据存储器地址空间;③片外 64 KB 的数据存储器地址空间:

04 NO FISOKACHIBANUAL 오이. 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; 2568 的片内数据存储器寻址空间,包括 SFR

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.

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

系统复位后 PC 为 0000H,系统从 0000H 单元 开始取指执行程序

1003H~002DH 用于 5 个中断源的中断服务 呈序入口地址.(地址见 5 个中断源)

工作寄存器区:00H~1FH 位寻址区:20H~2FH 字节寻址区:30H~7FH

寄存器间接寻址;寄存器 DPTR,R1,R0;指令助

1.工作状态: 多功能 8 位口,字节访问地址: 80H,位访问地址:80H~87H.

2.PU 日的切能 (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.

(2) 地址/数据复用总线:作数据总线用时输出人/输出8位数据 DO~D7;作地址总线用时输出低8位地址 AO~A7.

(1) | / O 口; (2) 程序存储器或片外数据存储器的高 8 位地址

复位:单片机的初始化操作把 PC 初始化为0000H,单片机的初始化操作把 PC 初始化为10000H,单片机从0000H,单元开始执行。(1)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

(4)复证方式: 正电自动复证式键电平复证相 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,000H 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。

DRG 0000H; SJMP MAIN; 转主函数 两种低功耗方式:待机方式和掉电保护方式.

1.待机方式 1)IDL=1,80C51 进入待机方式 辰荡器仍运行,并向中断逻辑,串行口和定时器

尿汤斯切运行,开间中间定相,中门口和定时 计数器电路提供时钟,中断功能继续存在 可CPU提供时钟的电路被阻断CPU不工作。 BP.PC.PSW.ACC及通用寄存器冻结在原状态

2)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 新,PCON.0 位(IDL 位)被硬件自动清"0",单片

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

F XX (1) 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

指令中直接给出操作数,出现在指令中的操

即数前面必需加上前缀"#"

如:指令 MOV DPTR, # 1234H 寻址空间: 程序存储器

例·指令 MOV A 3AH

2.直接寻址 在指令中直接给出操作数的地址

1. 文即寻址

退出待机方式进入正常工作方式。

2.掉电保护方式 1)PD 位控制单片机进入掉电保护方式

2)特殊功能寄存器 SFR 区:128~255 3)堆栈区及堆栈指示器 SP:8 位寄存器

田对象 MCS96

数据存储器。 **个逻辑存储器地址空间**

●2.4.1 程序存储器

2.程序入口地址(中断)

3.片内数据存储器(4个区)

)片内数据存储器:0~12

4.片外数据存储区

●2.5.1 P0 □

2.P0 口的功能

●2.5.1 P2 □

1.工作状态: 输出高 8 位地址. 2.P2 口的功能:

●80C51 单片机的工作方式

.片内和片外程序存储器

3.寄存器寻址 3.학단교국교 例如:指令 INC RO 寻址空间: RO~R7、A、B、CY(bit)、DPTR 最大的时序定时单位.80C51的指令周期有

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 片内数据存储区的低128单元,只能采用R0

R1 为间址寄存器。 片外数据存储区的 64K 单元 使用 DPTR 作

寻址空间: 内部数据存储器低 128 字节、特

殊功能寄存器 直接寻址是访问特殊功能寄存器的唯一方法

●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(也称冯诺依曼结构) 为间址寄存器。 寻址空间: 1)内部数据存储器(@R0 @R1@SF 仅 PUSH POP); 2)外部数据存储器(@R0 @R1 业用对家.MC396 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.

。 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 5数)+rel_如:指令JC 80H___ 址空间:程序存储器(PC+偏移量

以DPIR 或PC 对基础寄行输,参加储备 A 安 业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 (只有下列三条)

例如: MOVC A,@A+DPTR MOVC A,@A+PC JMP @A+DPTR 寻址空间: 程序存储器(@A+PC @A+DPTR)

寻址空间: 1) 内部数据存储器中 128 个可寻址位:

如 MOV C2BH
2) 可位寻址的特殊功能寄存器位
●数据传送类指令 (8 种) 沒介绍 MOV
1) A 内容送外部数据存储器或 I/O 2)存储器数据传送指令(或查表指令)

「MOVC A,@A+PC MOVC A,@A+DPTR 3)交换指令组 XCH 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(RO) = 30H.(A) = 3FH,片内(30H) = BBH. 执行指令:XCH A,@R0 执行结果:(A) = BBH,(30H) = 3FH. **4)节交换指令组**

(1) 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(011010B), サムセンスリン 执行指令:XCHD A.@RO 执行结果: (20H)=01110110B=76H, (A)=00110101B=35H

2) SWAP A 客累加器 A 的高,低半字节交换 例:设(A) = 36H(0011 0110B)

が 抗行指令:SWAP A 执行结果:(A)=63H(0110 0011B) KCHD 和 SWAP 主要用于十六进制数或 BCD

5)操作指令组 PUSH direct 入栈指令: (SP) (SP) + 1,(SP) (direct) 出栈指令: (direct) (SP), (SP) (SP-1) 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(8 种) ADD ADDC,INC,DA,SUBB,DEC,MUL,DIV MUL AB A.低 8 位 B.高 8 位

DIV AR

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指令:RRC A(将累加器内容和进位位一起循环右移。0 移入 CY,CY 移到 a7) 循环左移指令:RLA(累加器的内容逐位循环右 一位,a7 移到 a0.不影响标志位) 持进位循环左移指令:RLC A(累加器的内容和

市进口调外在炒有字、NLC 八条加路的约合和进位位一层纸环左移一位。有移入进位位 CYCY的内容移到 a0) 累加器按位取反指令·CPL A(累加器的内容 逐位取反指导、仍存在A中不影响标志位) 累加器洁0指令·CLR A(对累加器进行清0.不

1 字节的高 3 位 a10~a8 和指令第 2 字节 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

字节和第3字节组成.目标地址为64KB空 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 累加器判零转移指令:**JZ** rel:(若(A)=0.则

C)=(PC+2)+rel;若(A)≠0,则(PC)=(PC)+2) JNZ rel;(若(A)≠0,则(PC)= (PC+2)+rel;若 数值比较转移指令:CJNE A, direct, rel (指令格

偏移量 rel.具有比较转移和数值大小比较的功

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 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

长河州省专记ALL add110元宗汗电洞州自 址为 add116 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).

ALALL 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) 空指令"NOP"不做任何操作"仅 PC加工" ●中断逐:能产生中断的外部和内部事件。

80C51 有两级中断优先级,5 个中断源: ◆ 两个外部中断源 INTO 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。

1)INT0:外部中断 0. (矢量地址 0003H) 当 IT0 = 0 时,低电平有效 当 IT0 = 1 时 下降沿有效 (2)INT1:外部中断 1. (矢量地址 0013H)

当 IT1=0 时,低电平有效 当 IT1=1 时,下降沿有效 2 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 80C51 有两个源.即

(1) TF0:T0 溢出中断 (矢量地址 000BH) (2) TF1:T1 溢出中断 (矢量地址 001BH) 3. **串行中断 (矢量地址 0023H)** 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. ●中断分许控制——F.客存器

Ε			Е	Е	Е	Е	E
Α			S	T	X	T	X
				1	1	0	0
A:中断允许总控制位;ES:串行口中断允许;							
T1/0:T1/0 中断允许位;EX1/0:INT1/0 中断允							
F位。							
A DECEMBER 1D STATES							

置 1 为高优先级,置 0 为低优先级 PS:串口; PT1/0:T1/T0; PX1/0:INT1 默认优先级:INT0>T0>INT1>T1>串口

●中断请求的撤除 1.硬件自动复位:定时器/外部中断 2.软件清除中断(调用中断服务函数):串行接收

/发达中断
●十新响应"LCALL"被锁条件
1)CPU 正在执行同级或高一级的中断服务程
序中2)查询时请求的机器局期不是执行当
前指令的最后一个周期3)当前正在执行 PETI 指令指令或对介对 EL PIO 该汇写操作器(
●中断响应对例产生级中断系统中,中断响应

器周期,中断响应时间为3个机器周期 2)中断响应时间最长为8个机器周期,如果 CPU正在执行 RETI 指令或者访问 IP、IE 指 PUL在执行作出指令或者切问PI、E指 多则等待时间不会多余之个机器周期而中断 系统规定把这几条指令执行完必须再执行一 条指令后才能响应中断如这条指令恰好指令 机器周期张的指令再加上执行长调用指令 CALL所需2个机器周期

●外部中斯源的扩展:80C51 只有两个外部中 一个部中间,原的扩展。60051 尺有两个个小部中 或进行扩展:1)可通过增加"OC 门"+ 软件来 扩展:2)定时器/计数器

●中断服务程序设计 例试编写设置外部中断 INTO 和串行接口中断为高优先级外部中断 INT1 为低优先级屏 蔽 T0/T1 中断请求程序

ORG 0000H; SJMP MAIN; ORG 0003H; LJMP INTOINT;设 INTO 中断向量 RG 0013H; LJMP INT1INT; ORG 0023H; LIMP SIOINT: ORG 0030H MAIN: *****

MOV IP.#00010001B: MOV IE.#10010101B ●6.2.1 TO、T1 特殊功能寄存器 1.TO、T1 方式寄存器——TMOD

D7-D4 为 T1 的; D3-D0 是 T0 的 GATE:=0 与 INTO/1 无关;=1 可以 C/T: =0 定时器;=1 计数器。 可以计数. M1/M0: 00 方式 0; 01 方式

THO:计数值高8位:TLO: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

方式 1:16 位计数器 |万式 2: 自动重装 8 位计数器 |方式 2: 自动重装 8 位计数器 |HO=TLO=计数值: **T1 常常工作在方式 2.作为** 串行口波特率发生器。 4)方式 3: T0 拆成 8 位计数器/定时器(TL0)和 8 位定时器(TH0); T1 停止工作。

にある期=12⋅版 あみ平(1036) 几器周期=12⋅振荡周期=12/fosc 需要定时时间= 2^13 – 当前计数值

LO 溢出时产生 TO 中断(设置 TF0=1 HO 溢出时产生 T1 中断(设置 TF1=1 对引脚下降沿计数时:TLO对 TO 脚计数 .TO、T1 控制寄存器——TCON

「F1/0: T1/0 溢出标志(=1 溢出); 「R1/0: T1/0 运行控制位(=1 运行);

IE1/0: INT1/0 中断标志 IT1/0: INT1/0 中断触发 ●6.4.1 定时器中断程序应用 根据定时时间t、定时器工作方式(确定 L)及 晶体振荡频率fasc计算TC 值(十进制数),转换 成二进制数 TCB,然后送入 THi、TLi。

対于TO、T1:TCB=TCH+TCL MOV THi,#TCH; MOV TLi,#TCL; L在P1.0 引脚上产生周期为 2ms 的方波输

出。已知晶体振荡器的频率为f_{cc}-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。 ED, 母陽 IMS & F1.0 3 1 個上的电子 & C 1) 解: 正时常数计算 振荡器的频率 fose = MHz 机器周期为 2 μs,方式 0 计数略长度 L: 3(2~13 = 8 192),定时时间 t = 1 ms = 0.001s 完計堂数·

 $TC = 2^{L} - \frac{f_{ost} \times t}{2} = 8192 - \frac{6 \times 10^{6} \times 10^{-3}}{2} = 8192 - 500 = 7692$

CB=000 11110000 01100B; CH=1111000B(F0H); TCL=01100B(0CH)

と)歯件 ORG 0000H; AJMP MAIN; ORG 000BH; TO 中 断矢量 AJMP INQP; ORG 0030H; MAIN:MOV TMOD, # 00H;设 TO 为定时器方 式 0; MOV THO,# 0F0H;写定时常数(定时 ms) MOV TL0,#0CH; SETB TR0;启动 T0; ETB ET0;允许 T0 中断; SETB EA ;开放 CPU

SEIB E IO. 允许 I O 中断: SEIB E A. 汁放 CPU 中断 AMP \$定時中断等待: ORG 2000 H:TO 中断服务程序 IMOP:MOV THO. # FOH: 重写定时常数: MOV TI.O. # OCH: CPL P1.O. P1.O. 变反輸出: RETI: 使用 11 的方式 1 框架同上, Z需将 L 改为 16. 重新计算 TCB. 注的工人的特定设为 10, 重新订算 105, 使中断矢量、THi、Ti 产生两个方波,一个 200us,另一个 00us,同时使用串行口,用定时器/计数器

作**为波特率发生器。** 解:这时TO采用方式3工作,其中,TLO产生

100µs 定时,由 P1.0 输出方波 1.THO 产生 200 us 定时,由 P1.1 输出方波 2,T1 设置为方式 2, 作波特率发生器用. foce = 9.216 MHz. $TC = 2^{L} - f_{osc} \times t/12$

斯入; AJMP ITLO; ORG 00IBH; THO 的中断 MAIN: MOV SP. # 60H;设栈指针

MAIN: MOV SP,#60H;设栈指针 MOV TMOD,#23H;设T0为方式3,T1为2 MOV TL0,#083H;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 400) TH1,#0F6H;设 TH1 初值

FTR TRO: 启动 TIO ETB TR1;启动 THO ETB ETO:允许 TL0中断 CPU 中断开放

RG 0200H TLO:MOV TLO,#0B3H;重装定时常数 CPL P1.0;输出方波 1(200µs)

ITH0:MOV TH0, #66H:重装定时常数

CPL P1.1;制出力液 2(400μS)							
●串行	宁口控	制寄存	7器—	—sco	ON(9	位数技	居)
S	S	S	R	T	R	T	R
M	M	M	E.	В	В	- 1	- 1
- 0	1	_ 2_,	N.	- 8	8_		_ //5
SM0/	上串行	日土1	F万式	选择1	¥3, ∏	里红	見友
	加请来	标志。	RI:串	行口书	そ収円	断话:	永 标

SM0/1:00 方式 0;01 方式 1;10 方式 2;1 。 =0 接收数据帧/地址帧;=1 只接收地 M2:

址帧(第九位数据为 1) REN:=0 不允许接收:=1 允许接收 RBN: 古 2.3 中要发送的第 9 位数据 RBN: 方式 2.3 中接收到的第 9 位数据 RI: SCPU从串口接收到的第 9 位数据 自动把本标志置 1.表示 CPU 已接收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,否则即使有新的数据也不接

收) TI执行 MOV SBUFA 指令后串口自动发送 数据位、当发送完成后、硬件自动把 T 标志置 1.表示串口已空刷。可以发送下一个数据此时 应把 Ti 清 (1数件)并发送新数据。 ●串行口方式 0一同步移位寄存器方式 ☆本件即程

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 L**ED:** MOV SBUFA:串行输出 JNB TI,\$;输出等待 CLR TI;软件清中断标志 ACALL DELAY:轮录间隔 AIMP I FD·循环 DELAY:MOV R6,#200H

DIN7 R6 & 延时子程序 · RFT ●串行口方式 1——双机通信 例:试编写双机通信程序 甲乙双机均为串行口方式 1,并以定时器 T1 的方式 2 为波特率发

的数据快内容 通过串行口传至乙机

)数据水目、不地組成ではから 数据的传送以中断方式进行。 JRG 0000H; SJMP TRANS; ORG 0023H;串行 コ中断入口、AJMP SINT; ORG 0030H;

MOV TL1,#0FAH; MOV TH1,#0FAH; SETB EA; LR ES: MOV PCON,#00H; CLR TI; MOV

SBUF,76H; **Wait3:** JNB TI,WAIT3; CLR TI;MOV SBUF,75H; **WAIT4:** JNB TI,WAIT4; CLR TI;SETB ES; MOV

BUF,78H; MOV DPL,77H; CLR TI; MOV BUF,A; SJMP \$; ORG 0200H;

SINT: PUSH DPL; PUSH DPH; PUSH A; INC 77H; MOV A,77H; JNZ JP1: INC 78H;

P1: MOV A,78H; CJNE A,76H,END1; MOV A,77H; CJNE A,75H,END1; CLR ES:

ESCOM: POP A; POP DPH; POP DPL; RETI; END1: MOV DPH,78H; MOV DPL,77H; MOVX

)乙机接收 机通过 RXD 引脚接收甲级发来的数据.接收

波特率与甲机一样 接收的第 1、2 字节是数 据块的首地址 第 3、4 字节是数据块的末地

业域 1.第5字节开始是数据接收到的数据以 比存入数据块首地址开始的存储器中。 DRG 0000H; SJMP RECEIVE; ORG 0023H;

OFAH; MOV TH1,#OFAH; SETB EA; SETB ES;

CLR TI; MOV SCON,#50H; CLR 20H; MOV 70H,#78H; SJMP \$; ORG 0200H;

RSINT: PUSH DPL; PUSH DPH; PUSH ACC; MOV A.RO: PUSH ACC; JB 20H,DATA; MOV

0.70H: MOA A SRIJE: MOV @R0 A: DEC 70E

CLR RI; MOV A.#74H; CJNE A.70H.RETURN;

RETURN: POP ACC; MOV RO,A; POP ACC;

POP DPH; POP DPL;RETI; DATA: MOV DPH,78H; MOV DPL,77H; MOV

A,SBUF; MOVX @DPTR,A; CLR RI; INC 77H; MOV A,77H; JNŽ DATA1; INC 78H;

DATA1: MOV A,76H; CJNE A,78H,RETURN;

MOV A,75H; CJNE A,77H,RETURN; CLR ES;

●试编写串行接口以工作方式2发送数据的

MOV SBUF,A;数据写入发送缓冲器,启动发送

● PCON 寄存器 (D7)SMOD:=0 波特率不加倍:=1 波特率加倍

方式 0: 波特率=fosc/12 方式 1: 波特率=(2^SMOD)/32*T1 的溢出率 1)定时器工作在方式 0:

溢出率=(f_{osc}/12)*[1/(2^13-TC)]。 2)定时器工作在方式 1:

温田学−(f_{ost}/12/F[1/(2/16-10)]。 3)定时器工作在方式 2: 盗出率=(f_{ost}/12)*[1/(2/8-TH1)]。 **方式 2:** SMOD=0 波特率=f_{ost}/64

SMOD=1 波特率=fosc/32

- ┍ッチセニルス ਤ0000ps お迷年だり凶 30H; り16 个字符发送出去,要有奇校验位。 11.0592MHZ) DRG 0000H; SJMP MAIN; ORG 0030H;

MAIN: MOV SCON #11000010B: MOV

H1 #0FDH: MOV TI 1 #0FDH: SFTB TR1: MOV R0 #30H: MOV R2 #16:

方式 3:同方式 1

●例程:以 9600bps 的速率把片内 30H 开始

CON,#00H; MOV TMOD,#00100000B; MOV

溢出率=(f.../12)*[1/(2^16-TC)]。

RECEIVE: MOV TMOD #20H-MOV TI 1

JMP RSINT: ORG 0030H:

T 🖏 DRG 0023H; AJMP SPINT:

PUSH ACC; SETB EA;开中断 SETB PSW.4;切换寄存器工作组

MOV C.P.奇偶标志位 P 送 TB8

CLR TI;清除发送中断请求标志 MOV A @RO:取数据 署奇偶标志位

MOV SBUT,ASMSJACKS《伊藤/A INC RO、数据地址指针加 1 CLR OAFH:恢复现场 POP ACC; POP PSW; SETB OAFH CLR PSW 4.切换寄存器工作组 RETI

SPINT:CLR EA; 关中断

USH PSW:保护现场

FTR 20H-

RANS: MOV TMOD,#20H;置 T1 为方式 2

ON,#40H; MOV SBUF,78H;

WAIT1: JNB TI,WAIT1; CLR TI;MOV

WAIT2: JNB TI,WAIT2; CLR TI;MOV

主器,波特率为 2400。 坡特率的计算:这里使用 6MHz 晶振,以定时器 1 的方式 2 制定波特率,此时定时器 T1 相当 个8位的计数器

+算定时器 T1 的计数初值: 2^{SMOD} 波特率 = $\frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12} \times \frac{1}{2^8 - TH1}$

MAIN:MOV SCON,#11010010B;MOV PCON,#00H; MOV TMOD,#00100000B; MOV TH1 #0F4H: MOV TI 1 #0F4H: SETE TR1: SETB EA: SETB ES: MOV R0.#30H: SJMP リール及込 将以片内数据存储区的 78H 及 77H 的内容; 首地址、以 76H 及 75H 的内容减 1 为末地; COMISR: PUSH PSW: PUSH ACC: CLR RI:

MOV A.SBUF: JB P.BÍT1: JNB RB8.OK; SJÍMP 別外部のドラスのシャリーロマエンル。 例: (78H)=20H (77H)=00H: 首地址高位 (76H)=20H (75H)=20H: 末地址高位 対数据块首、末地址的传送以查询方式进行。 報報的体送、以本版主選出行。 BIT1: INB RB8 FXIT:

LOOP: INB THEOOP: CLR TI: MOV A @RO

ORG 0000H; SJMP MAIN; ORG 0030H; MAIN:MOV SCON.#11000010B:MOV

MOV R0 #30H: MOV R2 #16:

MOV C,P; CPL C; MOV TB8,C; MOV SBUF,A; INC R0; DJNZ R2,LOOP; SJMP \$ (结束)

(以上是查询方式)(下面是查询+子程序调用)

PCON #00H·MOV TMOD #00100000R· MOV

H1,#0FDH; MOV TL1,#0FDH; SETB TR1;

LOOP:MOV A,@RO; LCALL SENDCHAR;

SENDCHAR:WAIT:JNB TI,WAIT;CLR TI; MOV

C.P. CPL C: MOV TB8.C: MOV SBUF.A; RET

●从串口接收数据.存放在片内 30H-3FH 的单元内,并检查偶校验位.波特率为 2400

OK:MOV @RO.A; INC RO: EXIT:POP ACC; POP PSW; RETI(结束) ● ● ●程序例子 ● ●

1.把在 ROM 中 300H~3FFH 单元中最大的数 找出来存放在内部 RAM 的 30H 单元,遇零则 MOV DPTR #300H: MOV R2 #0: MOV R #0:

LOOP:CLR A; MOVC A,@A+DPTR; JZ ZERO; CINE A.B.NOTSAME; SJMP NEXTLOOP: NOTSAME: JC NEXTLOOP: (A<B) XCH A.B: (A>B

NEXTLOOP: INC DPTR; DJNZ R2,LOOP; ZERO: MOV 30H.B; (结束) 2.把内部 RAM 的 30H~7FH 单元分别和外部 RAM 的 50H~9FH 单元相加送入外部 RAM 的 1030H~107FH 单元 MOV R0,#30H; MOV R1,#50H; MOV DPTR.#1030H: MOV R2.#50H:

LOOP:MOV P2,#00H; MOVX A,@R1; ADD A,@R0; MOVX @DPTR,A; INC R0; INC R1; INC DPTR; DJNZ R2,LOOP; 3.把外部 RAM 中从 1234H 开使的 200 个字

节送至外部 RAM 的 5678H 开始的单元中 MOV DPTR,#1234H; MOV R7,#56;MOV R6.#78: MOV R2.#200:

LOOP:MOVX A,@DPTR; PUSH DPH; PUSH DPL; MOV DPH .R7; MOV DPL.R6; MOVX DDPTR,A; INC DPTR; MOV R7,DPH; MOV 6,DPL; POP DPL;POP DPH; INC DPTR; DJNZ

ROLDEL; POP DEL; POP DER; NDIO DER; NDIO ARE LOOP (结束)
4.将连续存放在外部数据存贮器 1000H 开始的10 个无符号数求和结果为双字节,存放在1100H (低位)、1101H (高位)中。
MOV R6#0(低位); MOV R7.#0(高位); MOV PTR #1000H: MOV R2 #10 LOOP: MOVX A.@DPTR: ADD A.R6: JNC

NORMAL; INC R7; NORMAL:MOV R6,A; DJNZ R2,LOOP; MOV

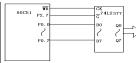
DPTR,#1100H; MOV A,R6; MOVX @DPTR,A; MOV DPTR,#1101H;MOV A,R7; MOVX



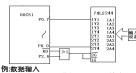
) I/O 口的数据传送指令 輸出数据的指令: 输入数据的指令有 MOV DATA, Px 2) I/O 口的位操作指令 立传送指令 MOV Px.y, C 立清 0 指令 CLR Px.y 位置 1 指令 SETB Px.y SEID PX.y IB PX.y, rel 位取反 CPL Px.y 位取入1转移 JB Px.y, rel 位为1转移 JNB Px.y, rel 位为1转移并清零 JBC Px (3) I/O 口其它操作指令 Px.y, rel

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 | E本計 対加 1指令 INC Px 減 1指令 DEC Px 減 1集件转移指令 DJNZ Px, rel 数値比较转移指令 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,其余了 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 @DPTRA:輸出数据

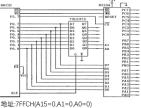


2. 简单输入口的扩展 数据输入需采用8位三态门控制电路。



MOV DPTR,#0BFFFH;指向 74LS244 口地址 MOVX A,@DPTR;读入数据 三杰:低电平状态、高电平状态、高阴抗

●8255A 与 80C51 的接口及应用



PA th til-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 口輸出 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//INC DPTF MOVX @DPTR.A MOV DPTR.#7FFEH:从 C 口输出//INC DPTF

例:把 C 口的第5位 PC5 置为 1 AOV DPTR #7FFFH MOV A,#00001011; PC5 置位

MOVX @DPTRA C 口控制字 D7:特征位 0; D6~D4:无定义; D3~D0:位选择; D0:1 置位 0 复位

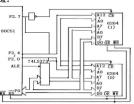
□ ● 単片机与 8255A 的接口电路,PA 口作输出 □ 接 8 个 LED 发光二极管,PB 作输出口,接 8 个按键开关,PC 口不用,都工作在方式 0,要实 现"按下任意键,对应的 LED 发光",相应的程序 读 PB 口开关状态,送 PA 口输出控制 LED,循

MOV DPTR,#0FF7FH:指向 8255A 的控制口 MOV A,#82H;工作方式控制字

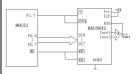
MOVX @DPTR,A;向控制口写控制字,PA 输出,PB 输入 LOOP:MOV DPTR.#0FF7DH:指向 8255A 的

MOVX A,@DPTR;读 PB 口按键状态 MOV DPTR,#0FF7CH;指向 8255A 的 PA 口 MOVX @DPTR,A;从 PA 口輸出,驱动 LED 发光

SIMIP LOOP ● 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



●DACC0832 单缓冲方式



DEC RO: XCH AV@RO: INC RO

JB 00H,PX :有交换则再比较一遍

将 2000H 单元开始的一批数据传送到从

MOV R6,DPL;目的数据区首址存入寄存器

的 30H 中. MOV DPTR,#2000H;源数据区首址

LP: POP DPH:取数据区地址指针 POP DPT, MOVX A,@DPTR;取源数INC DPTR; PUSH DPL; 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;

●设计—个简单系统,当一个键按下时就

LED BIT P1 0:ORG 0000H: SIMP MAIN: ORG

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; SIMP MAIN; ORG 000BH; SIMP INT1_ISR; ORG 0030H

MAIN: MOV SP.#20H; SETB ITO:SETB EA:

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

OH,RO; POP DPL; POP DPH; POP ACC; POP

●利用 T0 定时,每隔 2ms 使连接在 P1.0 引

脚上的 LED 闪烁一次.(12MHz) (查询方式)ORG 0000H; SJMP MAIN; ORG

MAIN:MOV TMOD,#00000000B; MOV

LOOP: SJMP LOOP TOISR:MOV TH0,#11100000B;MOV

●设计一实时时钟,能显示"时:分:秒:十分

秒"。以 BCD 码依次存放在 30H、31H、 32H、33H。(晶振=6MHZ)

HOUR EQU 30H; MIN EQU 31H; SEC EQU 32H: TENSEC EQU 33H: ORG 0000H: SIMP

MAIN:MOV SP #40H: MOV TMOD #01H:

LOOP: LCALL DISP(显示); SJMP LOOP; TOISR: PUSH PSW: PUSH ACC: MOV

MOV SEC A: CINE A #60H DONE: MOV

HO,#3CH; MOV TLO,#0B0H; INC TENSEC

MOV A,TENSEC; CJNE A,#0AH,DONE; MOV TENSEC,#0; INC SEC; MOV A,SEC; DA A;

SEC,#0; INC MIN; MOV A,MIN; DA A; MOV

MIN,A; CJN A,#60H,DONE; MOV MIN,#0; INC HOUR; MOV A,HOUR; DA A; MOV HOUR,A;

MAIN; ORG 000BH; SJMP TOISR; ORG 0030H

MOV TH0,#3CH; MOV TL0,#0B0H; SETB TR0;

1.0 #00011000B: CPL P1.0:RFTI

FTR FA: SFTR FTO:

TH0,#11100000B; MOV TL0,#00011000B;

SETB ETO; ···; LCALL DISP; ···; SJMP MAIN;

MAIN:SETB ITO; SETB EA; SETB ETO; SJMP

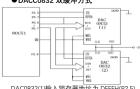
KEY BI P0.0;LED BIT P1.0;ORG 0000H

0003H; SJMP INTO_ISR; ORG 0030H

PX2: MOV A,@R0;A < DX+1 DJNZ R3.PX1:比较 9 次

LP: MOV A,R0; MOVX @DPTR,A;送出模拟量 INC R0; SJMP LP

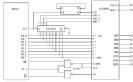
●DACC0832 双缓冲方式



DAC0832(1)输入锁存器地址为DFFFH(P2.5) DAC0832(2)输入锁存器地址为BFFFH(P2.6) 二片 0832 的 DAC 寄存器地址:7FFFH(P2.7) 若第一片的数据在 R0 中.第二片的数据在 R1 中此时送数据程序为

点亮 LED 灯,再次按下时熄灭 LED 灯,采 用查询方式实现。 MOV DPTR,#0DFFFH;将数据送 DAC(1)锁存器 MOV A,R0; MOVX @DPTR.A; MOV DPTR.#0BFFFH;将数据送 DAC(2)锁存器 MAINJB KEY.MAIN:CPL LED:SJMPMAIN:END MOV A.R1: MOVX @DPTR.A: MOV DPTR #7FFFH: MOVX @DPTR A

●ADC0809 与 80C51 连接图



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.

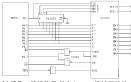
ORG 0000H; SJMP MAIN; ORG 0030H MAIN: MOV R1,#20H

MOV DPTR,#/FF8H;指向通道 0 地址 MOV P7,#/08H;共需转换 8 个通道 LOOP: MOVX @DPTR,A启动 A/D 转换 LCALL D128us;延时等6 A/D 转换值 MOVX A,@DPTR,读入 A/D 转换值 MOV @R1.A

INC DPTR;指向下一通道地址 DJNZ R7,LOOP;8 个通道未转换完,则继续。

THO,#11100000B, MOV TEO,#00011000B, P1.0; SJMP LOOP; (中断方式)ORG 0000H; SJMP MAIN; ORG 000BH; SJMP TOISR; ORG 0030H; MAIN: MOV TMOD,#00000000B; MOV D128µs: MOV R6,#200H; NOP; DJNZ R6,&; 延时 128µs 子程序 RET: H0,#11100000B; MOV TL0,#00011000B; SETB TR0: SETB EA: SETB ET0

●ADC0809 与 80C51 中断方式接口



例:采集 8 路模拟量,并存入 20H 地址开始的 内部 RAM 中。 ORG 0000H: SIMP MAIN: ORG 0003H: 9

部中断 0 入口地址; LJMP INTDATA; ORG 0100H;数据采集程序

UIUUI: 放低米条柱庁 MAIN: MOV RO, #20日: 数据缓冲区首址 MOV R2, #8:8通道计数器 MOV DFTR: #7FBH:指向0通道 START:CIR F0 清中断发生标志 MOVX @DPTR4:启动 A70 (P27=0, /WR=(SETB:T10: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

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

SETB 00H; DX>DX+1 量交换标志位, 20H.0=1 XCH A ,@R0; DX 与 DX+1 交换

为不合格产品,则被剔除掉。 设数据组为 X1,X2,···X10 共 10 个。 ORG 0000H

PX1: INC RO; MOV R1,A;

JBB A,@R0; DX - DX+1

JC PX2;DX<DX+1 则转 PX2,不交换

LOOP: JNB FO, LOOP:中断发生标志是否为 0 DJNZ R2,START:8 个通道转换是否结束

SIVIF IVIAIIN
INTDATA:MOVX A,@DPTR;读数据(P2.7=0,
RD!=0),硬件撤销中斯 MOV @RO.A:存数据
INC RO; INC DPTR:指向下一通道; SETB FO;置
中断发生标志; RETI MILEHIGH EQU 31H; MILELOW EQU 30H; DRG 0000H; SJMP MAIN; ORG 000BH; SJMP 0ISR; ORG 0030H;

MAIN: MOV SP #40H·MOV MOD,#00000100B;MOV TL0,#00010011B;

ETB TRO: SETB EA: SETB ETO: L**OOP:** LCALL DISP(显示); SJMP LOOP; TOISR: PUSH PSW: PUSH ACC: MOV H0,#11110101B; MOV TL0,#00010011B; MOV A MILELOW: INC. A: DA A: MOV

MILELOW,A; MOV A,MILEHIGH; ADDCA,#0; DA A; MOV MILEHIGH, A; POP ACC; POP

●设计数据采集程序,每隔 200us 读取 P1 口的数据,依次存放在外部 RAM 的 1000H-10FFH 的内存単元(晶振=12MHZ) PTRH EQU 30H; PTRL EQU 31H; ORG 0000H SJMP MAIN; ORG 000BH; SJMP T0ISR; ORG

MAIN:MOV TMOD,#00000010B; MOV TH0,#56; MOV TL0,#56; SETB TR0; SETB EA;

DPH; POP ACC; POP PSW; RETI 3000H 开始的单元中,数据长度在内部 RAM

ORG 0000H: SIMP MAIN: ORG 0030H MAIN: MOV R2,#10H; MOV R0,#30H; MOV PUSH DPL:源数据区首址压栈保护 PUSH DPHMOV DPTR,#3000H;目的数据区 DPTR #8000H

LOOP: MOV A,@R0; MOVX @DPTR,A; INC R0; INC DPTR; DJNZ R2,LOOP; SJMP \$; END ●在片外 RAM 找出最大数

LP: MOVX A,@DPTR; CJNE A,20H,CHK; SJMP

●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

LOOP: MOV P1,A; LCALL DELAY; CPL A; SJMF

DELAY: MOV R1,#100 LOOP1: MOV R2.#100 LOOP2: MOV R3,#50; DJNZ R3,\$; DJNZ R2,LOOP2: DJNZ R1,LOOP1; RET; END ●编写串口 Echo 程序,即串口接收到的任 何字符马上转发出去。

位自动重载计数 MOV TL1.#244;F=11.0592MHz,波特率 4800 MOV TH1 #244: SETR TR1 MOV SCON.#11000010B;串口工作方式3

LOOP:JNB RI,\$;MOV A,SBUF; CLR RI;CLR TI; MOV SBUF,A; JNB TI,\$; CLR TI; SJMP LOOP;

●编写一串字符串"hello world "的程序,采用奇校验。

START:MOV DPTR,#0030H; MOV R0,#0; MOV R7,#12; MOV TMOD,#20H; MOV L1,#244; MOV TH1,#244; SETB TR1; MOV CON #11000010B

●嵌入式最小系统 电源电路 时钟电路 存储电路 调试接口 复位电路 嵌入式微型处理器芯片 接口电路

LOOP: JNB TF0, LOOP; CLR TF0; MOV TH0,#11100000B; MOV TL0,#00011000B; CPL MCU;3)嵌入式 DSP 处理器 MPU;4)嵌入式片

●RISC 的设计准则:

周期内元成一条指令 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

1.充分利用 VISI 芯片的面积 2.提高计算机运行速度:指令格式少:采用较多通用寄存器:适合流水线工作 3.便于设计可靠性高 4.有效支持高级语言程序 ●CISC 与 RISC 对比

2] 列行时间 CISC 有的指令执行时间很长,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储器, RISC 没有较长执行时间的指令.

3)编码长度 CISC 编码长度可变,1-15 字节; RISC 编码长度 固定,通常为 4 个字节.

5)条FF CISC 可以对存储器和寄存器进行算术和逻辑操作 操作; RISC 只能对寄存器对行算术和逻辑操作 Load/Store 体系结构

於代的程序 ◆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 推: 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 会出微波速度电离二径流移

目 可谓純米提為這行效學 ◆ ARM 的遺水线技术: 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咸半速度提高1倍 ARM7三级流水(预取, 译码,执行, 访存,写回) 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, ¬,;;,, ARM11:八级流水(预取,预取,发射,译码,转换,

续执行,并获得总线控制权 ● DMA 控制器处理的四种情况: ●ARM 的 7 种处理器运行模式

上理器模式	说明
月户(usr)	正常程序执行模式
系统(sys)	运行操作系统的特 权任务
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)	操作系统的保护模式
P止(abt)	用于支持虚拟内存 和/或存储器保护
未定义(und)	支持硬件协处理的 软件仿真

后六种:特权模式 后五种:异常模式 ●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

16 位的 Thumb 指令集:ARM 指令集的功能子

株式: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 ARM 状态,此时处理器执行 32 位的字对准的

ΔRM 指令 Thumb 状态,此时处理器执行 16 位的,半字对 准的 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

ARM 代码快约 40~50%;4)使用 Thumb 代码, 存储器的功耗会降低约 30%

●状态切换方法 ■ 不必り挟力法 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.
●通用寄存器

●通用寄存器 1)未分组寄存器 R0~R7: 所有运行模式下, 都指向同一个物理寄存器。 2)分组寄存器 R8~R14: 每一次访问的物理寄存器和处理器当前的运行模式有关。

3程序计数器 R15 ARM 状态未两位为 0, 高 30 位记录 PC 值 Thumb 状态未位位 0, 高 31 位记录 PC 值

□和IDMの水念水世址 0. 高 31 址记束 P. 值 单程序状态 春存器 1 个当前程序状态寄存器 CPSR 可以在任何工作模式下散访问5 个程序状态备价寄存器 SPSR 只有在异常模式下才能被访问(各种异常模式下有的)与岸常发生时 SPSR 用于保存当前程序状态等存器 CPSR 的状态,从异常退即时 田子体卷 (228)

出財 用于恢复 CPSR 場的程序状态保存等存得功能包括: 1)保存 ALU 中的当前操作信息 2)控制 允许和 廃止中断 3)恢复 近延陽的运行模式、 ● 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: 1)的 - R7 是相同的、3)CPSR 和所有的 SPSP 是 相同的、3)Thumb 状态下的 SP 对应于 ARM 相同的、5)Thumb 状态下的 SP 对应于 ARM 下的 R13; 4)Thumb 状态下的 LP 对应于 ARM 状态下的 R14; 5)Thumb 状态下的程序 计数器对应于 ARM 状态下的 R15

计数器对应于 AKM 办芯 ●ARM 的异常中断处理 异常中断类型

指今袻取中 数据访问中山 保留 普诵中断请求(IRC

下一条指令的地址存入相应连接寄存器)将 CPSR 复制到相应的 SPSR 中;3)根据 1)将下一条相字即是 LR:2)将 CPSR 复制到相应的 SPSR 中,3月85年2 常类型,强制设置 CPSR 的运行模式,4)强制 PC 常类型,强制设置 CPSR 的运行模式,4)强制 PC

从相关的并常问查地址取下一条指令执行 ●异常的返回过程(复位不需要返回) 1)将连接寄存器 IR 的值减去相应的偏移量送 到 PC 中; 2)将 SPSR 复制到相应的 CPSR 中; 3)若设置了中断茶止位则清除

▼20介71個技术 1)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:2)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;

下占用 ARM 芯片上的任何硬件资源: 4)比 较方便和容易发现程序中的问题 1)保护现场; 2)模式切换; 3)获得中断服务子程序地址; 4)多个中断请求处理; 5)中断返回,恢

●S3C2410A 的 DMA 控制器(4 个 采用 DMA 方式进行数据传输的具体过程如

F: L分设向 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 销来回答信号: 6.在 DMA 控 部设有发出UNIA用水凹含信号。6在UNIA 控制 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息 与CPU 接收到中断信息后,转入中断处理程序 进行处理:8中断处理结束后。CPU 返回断点缘

1)源和目的都在系统总线上;2)源在系统总线 上,目的在外围总线上;3)源在外围总线上,目 的在系统总线上;4)源和目的都在外围总线上

● ARM F 振発 を ARM 7 TDM (后缀含义 ARM 7 TDM (后缀含义) I:Embeded - ICE,支持片上断点和调试点 1:内嵌硬件乘法器(Multiplier) 『文付九工师』、 :支持高密度 16 位的 Thumb 指令集. RM7 E J (后缀含义)

支持新的 101/0 支持增强型 DSP 指令集 ● ARM 指令系统及编程技术 ARM 指令的一般格式:

是否影响 CPSR 寄存器的值

d:目标寄存器 n:第1个操作数的寄存器 p2:第2个操作数

● ARM 指令的条件域<cond> cond CPSR 中标志位 含义

CPSR 中标志位 含义 乙置位 相等 乙 清空位 相相等等数大于或等于 乙 清空位 无符号号数等于 入 置位 无数或零 V 置位 溢出 V 清零 未溢出 C 置位 Z 清零 ラ V消零 禾溢出 C 置位 Z 清零 无符号数大于 C 清零 Z 置位 无符号数小于或等于 N 等于V 带符号数小于或等于 N 不等于V 带符号数小于 Z 清零且 N 等于V 带符号数大于 Z 置位或 N 不等于V 带符号数小于或

MOV R0,#0XFF00 MOV R1.R2 寄存器移位寻址

种移位操作 (2)LSR 逻辑右移 (4)ROR 循环右移; L)LSL 逻辑左移 B)ASR 算术右移)RRX 带扩展的循环右移. I: MOV RO, R1, LSL #5;R0=R1 逻辑左移 5 位

寄存器间接寻址 .DR R0,[R2]:将 R2 指向的存储单元内容保存

基址变址寻址 STR R1,[R0,#-4];先 R0=R0-4,然后把 R1 的值 寄存到 R0.

"仍对了想 ;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 字器用"-"连接,否则用","分隔 列:LDMIA RO.(R1-R5);

●ARM 指令分类说明 数据处理指令 2.跳转指令: 1)B 跳转指令

格式:B {<cond>} <addr> 功能:跳转到目的地址.跳转范围:当前指令的±

例: B exit; 程序跳转到标号 exit 处 带返回的转移指令

B式:BL (<cond>) cond>) cond>) caddr> 功能:跳转到指定地址,同时 LR(R14)=PC(R15) 列: BL func; 调用子程序 func

MOV R15, R14;子程序返回 BX 带状态切换的转移指令

A式:BX < Kn>; 力能:跳转到目标地址:目标地址处的指令可以 是 ARM 指令也可以是 Thumb 指令:目标地址 寄存器 Rn 和 0xFFFFFFFE 相与的结果. LDR RO, exit; exit 处的地址装入 RO 中 BX RO: 跳转到 exit 处

I)BLX 带返回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, 苦目标地址处为Thumb指令,则状态从ARM 刃换到 Thumb 状态 列: BLX T16; 跳转到 T16 处执行

T16 后面指令为 Thumb 指令

3.Load(内存-寄存器)/Store(寄存器-内存)指 A.单寄存器的存取指令(LDR,STR)

合成的有效地址写回到基址寄存器 DR Rd [Rn.Rm] :将内存中的地址为 Rn+Rm 内字数据装入寄存器 Rd 中

入基址寄存器 .DR Rd.[Rn].Rm:将内存地址为 Rn 的字装入 ₹d.并将新的地址_Rn+Rm 写入 Rn

DR{<cond>}T < Rd>, <addr>, :若指令有 T,即使处理器是在特权模式下,也

)STR 字数据存储指令 格式: STR{<cond>} <Rd>, <addr>; 功能: 把 Rd 中的字数据保存到 addr 所表; 为存地址中;同时将合成的有效地址写回到基 址寄存器

知 5 (1 mg) 例:STR R0, [R1, #5]!:把 R0 中的字数据保存到 以 R1+5 为地址的内存中;R1=R1+5 寻址方式同 LDR STRH:半字数据存储指令

TR{cond}H <Rd>,<addr>; TRT:用户模式的字数据存储指令 3.多寄存器存取指令(LDM.STM)

格式: LDM {<cond>} {<type>} <Rn> {!},

<regs>{^};
d邮:从一片连续的内存单元读取数据到各个 寄存器中:一般用于多个寄存器数据的出栈

type 字段: 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 ED:空递减堆栈 FA:满递增堆栈

EA:空递增堆栈 ^}:当 regs 中不包含 PC 时,用于指示指令所用的寄存器为用户模式下的寄存器,否则指示 旨令执行时,将 SPSR 的值复制到 CPSR 中 STM 批量数据存储指令 格式: STM {<cond>} {<type>} <Rn>{!},

<regs>{^};
幼能:将各个寄存器的值存入一片连续的内 20th. 19 日 | 可付額的III 伊人一片连续的内存单元中一般用于多个寄存器数据的入栈 ^}} 指示指令所用的寄存器为用户模式下的寄存器

其它参数用法同 LDM 指令 C.单寄存器交换指令(SWP) I)SWP 字数据交换指令

/3WF 子数/3ススロマ 春式: SWF(<0nd>> <Rd>- <Rm>.[<Rn>]; 功能:从 Rn 所表示的内存装載一个字并把这 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 RO.然后将 R1 中的字数据保存到 R2 所表示的内存单元中。

) SWPB 字节数据交换指令 SWP{<cond>}B <Rd>, <op1>, [<op2>]; 功能:Rd(低 8 位)=[op2](一字节), [op2]=op1

1版 8 (1) 1 **1.程序状态寄存器指令** 1.)MRS 状态->通用寄存器的数据传送指令 格式: MRS{-cond>} < Rn>, CPSR/SPSR; (2013) MRS(-cond>} < Rn>, CPSR/SPSR; 情式: MR3(Scolid 27 SR12 SR25) 517 功能: 将程序状态寄存器的内容传送到目标 寄存器 Rd 中;当进入中断服务程序或进程切换

时,可用来保存当前寄存器的值 例:MRS RO, CPSR; 状态寄存器 CPSR 的值存 入寄存器 RO 中 NMSR 通用->状态寄存器的数据传送指令 MSR {<cond>} CPSR/SPSR <field>, <op1>; 力能将Rd 的值传送到程序状态寄存器中,当 是出中断服务程序或进程切换时,可用来恢复 大态寄存器的值

I: MSR CPSR_f, RO;修改 CPSR 的条件标志域 MSR CPSR_fsxc, #5; CPSR 的值修改为 5

)协处理器寄存器和内存单元之间的数据传 类指令(LDCSTC)

)ARM 处理器寄存器和协处理器寄存器直接 的数据传送指令(MCR,MRC) 5.软件中断指令

ISWI 软件中断指令 2)BKPT 断点中断指令 ●**交叉开发环境** 交叉开发环境由宿主机(计算机)和目标机(嵌

人式系统组成 人式系统组成 L)宿主机(Host): 是用于开发嵌入式系统的计算机一般为 PC 机(或者工作站),具备丰富的 次硬件资源,为嵌入式软件的开发提供全过程

)目标机(Target): 即所开发的嵌入式系统,是 -,,a-xx,xx,rarget,, xx/n/m/友的嵌入式系统,是 嵌入式软件的运行环境,其硬件软件是为特定 立用定制的.

...............

WR 控制语气作序作幅新有碳 PSEN 控制语 导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。 ● INTO 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 查询若查询其早市断标志为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 最后恢复原程序的断点地址执行,且恢复中

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 绕指令的处理过程被折分成几个更小的能够 能激水线并行的单元允许流水线在书前 指令译码器的授某取其下一条指令3.套存 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 弦按性提供决查的局部储存访问。4.Load-Store 结构处理器 尺处理寄存器中的数据 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.完分利用记 定方的面积6.提高计算机运行速度7.便于设 计可参数查。2.6 为大转运和运车程6.

人物借取。子数据的画子卫行储住底地址中,则 字数据的低字节则序放在高地址中, 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

如任放印定于级级的硕士可 量程序存储器和数据存储器地址冲突,如何区分介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的间指令为MOVC。数据存储器访问指令为MOVC。数据存储器访问指令 48H 2)MOV A,60H; 寻址方式: **直接寻址** (A)= 35H 为 MOVX;选通信号不同,前者为/PSEN,后者为

●中断响应的现场保护?所谓现场是指中断发 MUL AB 存留中的数据处价总区等设置,在海与中旬的发现考虑保护现场的功能在80C51 单片微机中现场一般包括雾加器AX工作寄名 8R0-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 R14.当則程序が必合存態 CPSR 仔/入5PSR 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素的特定的地址范围内, 中断向量素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。 POP A

●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 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

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.中断执行与于程序调用的差别。 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

2.3.中断响应是受控的,其响应时间会受一些 |素影响;子程序响应时间是固定的. ●程序存储器的特点? ●程序仔值器的特点?
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.

••••••

-、填空 .若单片机使用的频率为 6MHz 的晶振.那? 时钟周期为 0.167us,机器的周期为 2us.

.单片机的特殊功能寄存器区位于片内人 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. 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

1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.

I.MCS-51的外部中断有触发方式,一种是**下降** ERROR: MOV A,#255; RET; 沿,另一种是低电平触发. 5.8051 单片机的工作寄存器分成 4 个组,每个

组8个字节 6.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。 ARM 状态下SP 寄存器指的是 R13LR 寄存

器指的是 R14.PC 寄存器指的是 R15. B.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. 9.S3C410 中,其地址空间有 **8**个 BANK,每个

10.8051 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

RΔNIK 最大的容量是 126Mb

INTO,INT1 的有效信号为低电平激活外部中断 11.在中断服务程序中现场保护和现场恢复期 间,中断系统处于**关中断**状态 12.MCS-51 是 8 位单片机 共有 4 个并行 1/0

址总线高 8 位,P3:相应引脚会输出控制信号 . 填空颗

L.串行口的发送缓冲器和接收缓冲器使用F -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



读sbuf(mov a,sbuf):接收数据寄存器 写shuf(mov shuf a):发送数据客存器

2.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

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,

1)MOV A ,#48H; 寻址方式: **立即寻址** (A)=

2.执行下列程序后,(A)=40H,(B)=01H.

.请分析下面程序执行后的操作结果,(A)=

MOV @R0,A; MOV 41H,R0 XCH A.R0 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; POP B I.假设 RO 的内容为 0x104,寄存器 R1,R2 的内

0x8010 和 0x10.

LDMIA R0!{R1,R2}

1)定时常数计算 $6 \times 10^6 \times 0.005$

TCB=0000 0000 0000 0110B TCH=00H(高 8 位) TCB=06H(低 8 位)

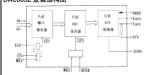
小编程 ORG 0000H; AJMP MAIN; ORG 001BH; AJMP INOP: ORG 0100H:

AIMP \$: ORG 2000H: INQP: MOV TH1,#00H; MOV TL1,#06H; CPL

P1.0: RETI 2.编写一段子程序, 二位 BCD 转二进制,入口 出口是 A, 若非法 BCD, 则 A 返回值:255

A,#10,NEXT1; LJMP ERROR; NEXT1: JNC ERROR; MOV B,#10; MUL AB; XCH A,R1; ANL A,#0FH; CJNE A,#10,NEXT2; LJMP

NEXT2: JNC ERROR; ADD A,R1; RET;



自动增减来提高运行效率

SETB ET0: MOV PTRH.#10H: MOV

PTRL #00H-S1MP \$ TOISR: PUSH PSW; PUSH ACC; PUSH DPH; PUSH DPL; MOV TH0.#56; MOV TL0.#56; MOV A,P1; MOV DPH,PTRH; MOV DPL,PTRL MOVX @DPTR.A: INC PTRL: POP DPL: POP

●编写块搬移程序

ORG 0000H; MOV R0,#10H; MOV DPTR,#1000H; MOV 20H,#0

CHK: JC LP1; MOV 20H,A LP1: INC DPTR: DJNZ RO,LP: SJMP \$: END

RG 0000H; SJMP MAIN; ORG 0030H MAIN: MOV A.#0

英独住庁如下: ORG 0000H; SJMP START; ORG 0030H **START:**MOV TMOD,#20H;T1 工作于方式 2, 8

SETR REN

0000H; SJMP START; ORG 0030H TABLE:DB "hello world!"

LOOP:CLR TI; MOV A,RO; MOVC A,@A+DPTF MOV C.P; CPL C; MOV TB8,C; MOV SBUF,A; JNB TI,\$; INC RO; DJNZ R7,LOOP; END ●●●●嵌入式系统及应用●●●●

●嵌入式处理器的种类 1)嵌入式微处理器 MPU:2)嵌入式微控制器

- NIGO H30KH (在内): 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

●RISC 架构的优点:

指令数量很多; RISC 较少,通常少于 100

3)编码长度

의氏 4)寻址方式 CISC 寻址方式多样; RISC 简单寻址.

的字数据装入寄存器 Rd 中,并将新地址

(I):表示当指令执行完毕后, 将最后的地址写

●RISC 的特点?1.指令集:减少了指令种类,每

心月的個秋6.提高计算机运行速度//搜十设计可靠性高.8有效支持高级语言程分 ● CPSR 与 SPSR 的关系存在处理器所有的运 符模式下均可以访问当前的程序状态存存器 (CPSR)每一种异常模式下又都有一个专用的 程序状态保存著存器(SPSR)当异常发生的 SPSR 用于保存当前程序状态者存器 CPSR 的

状态,以便从异常退出时,由 SPSR 来恢复 CPSK从而近行开帝处理。 ◆ARM920T体系结构支持哪两种方法储存 字数据?ARM体系结构可以用两种方法存储 字数据充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

(30H)=0FH

MOV A,#0AH;MOV B,#20H

40H.(R0)= 60H MOV A,#60H; MOV R0,#40H

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 RO 和 R1,其内容分别为

STMIB R0!.{R1.R2}

四、编程题 1.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.

TMOD 控制字:0010 0000B,即 20H

MAIN: MOV TMOD,#20H; MOV TH1,#00H MOV TL1,#06H; SETB TR1; SETB ET1; SETB EA;

SUBP: MOV R1,A; ANL A,#0FH; SWAP A; CINE

FRROR-

DAC0832 逻辑结构图