

第一讲 单片机基础知识

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解单片机的基础知识。
- 2、了解单片机的发展概况。
- 3、熟悉数制的转换。

教学重点:

单片机的含义及作用

教学难点:

数制间的转换

第一章单片机基础知识

§1-1 单片机概述

一、什么是单片机？

计算机（微型）的系统结构：如图 1 所示。

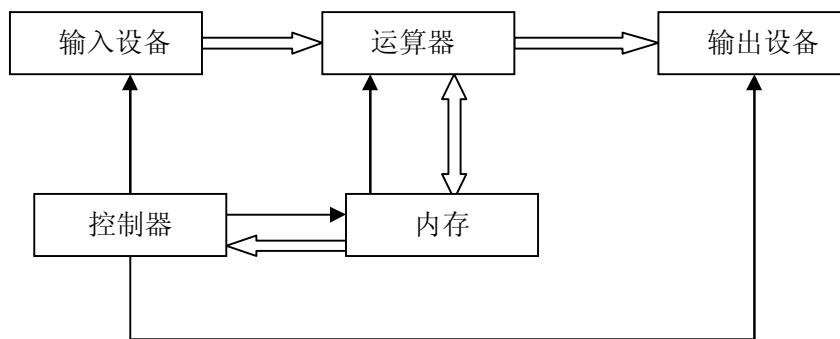


图 1 微机系统结构

多板机：

单板机：Z80

单片机：Intel MCS-51，体积小，功能强，可靠性高，价格低。以最小系统或单片机扩展系统出现在：家用电器，智能仪表，工业过程控制，航空，汽车等领域。

单片机特点：

- (1) 受集成度限制，片内存储器容量较小，一般内 ROM：8KB 以下；
内 RAM：256KB 以内。
- (2) 可靠性高

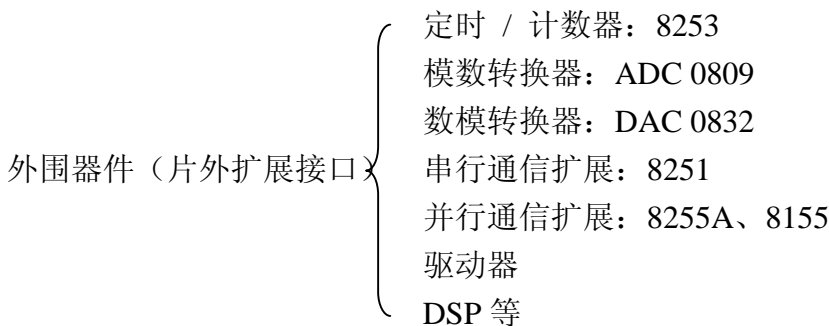
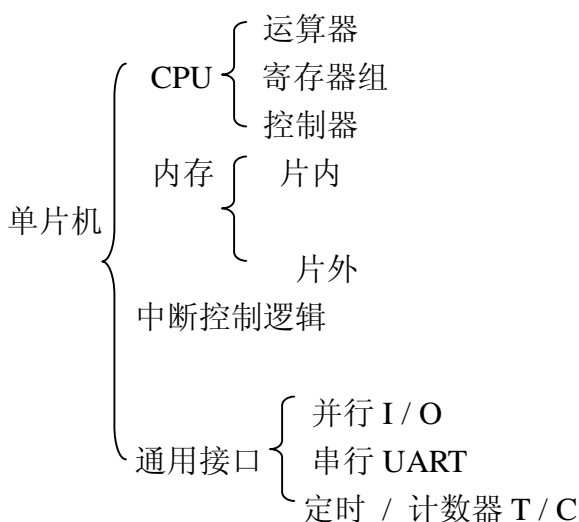
- (3) 易扩展
- (4) 控制功能强
- (5) 易于开发

单片机发展分四个阶段:

- 第一阶段 (74 年~76 年) 初级阶段: 仙童公司 F8 (8 位 CPU, 64KB)
- 第二阶段 (76 年~78 年) 低性能单片机: Intel 公司 MCS-48 (8 位 CPU)
- 第三阶段 (78 年~83 年) 高性能单片机: Intel 公司 MCS-51、Motorola 6801、Z8
- 第四阶段 (83 年~今) 新一代单片机 (单片微控制器): AT89C51(Atmel)

二、 单片机系统的组成

1、 硬件部分



2、 软件部分 (即程序)

需用户自己开发, 根据指令系统进行设计。某些功能硬件可以实现, 软件也可以实现。

硬件实现——速度快, 占 CPU 时间少; 但电路复杂、成本高。

软件实现——简化硬件电路设计, 可靠性高, 成体低, 占 CPU 时间少, 实时性差, 此

外，还需开发设备。

§1-2 计算机中数的表示及运算

计算机只识别和处理数字信息，数字是以二进制数表示的；它易于物理实现，同时，资料存储、传送和处理简单可靠；运算规则简单，使逻辑电路的设计、分析、综合、方便，使计算器具有逻辑性。

一、用数制及转换

1、各种进位计数及其表示方法

数字符号：0、1、2、……9 ——数码。数码的个数——基数。

进位规则：逢十进一

例如，十进制数，10 个数码；采用“逢十进一”

$$30681 = 3 \times 10^4 + 0 \times 10^3 + 6 \times 10^2 + 8 \times 10^1 + 1 \times 10^0$$

例如，二进制数，2 个数码，采用“逢二进一”

$$(11010100)_2 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

总之，N 进制数，N 个数码，“逢 N 进一”

2、数制之间的转换

任意进制之间相互转换，整数部分和小数部分必须分别进行，

十进制转换成二进制——短除取余法

		余数
2	45	
2	22	1
2	11	0
2	5	1
2	2	1
2	1	0
	0	1

十进制小数转换成二进制小数——乘 2 取整法。

二进制转换成十进制——展开求和法。

$$\begin{aligned} (101101)_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 0 + 8 + 4 + 0 + 1 \\ &= 45 \end{aligned}$$

二进制转换成八进制、十六进制与此类似。

二、机器数及其编码

1、机器数与真值

机器只认识二进制数：0、1。

这是因为，电路状态常有两个，如通、断；高电平、低电平；...可用0、1表示。这种0、1、0、1...1在机器中的表现形式——机器数。一般为8位。

机器数有：

{	无符号数：00000000B、.....11111111B 即 00H ~ FFH	
	带符号数：+1010110B、-1101001	————→ 真值
	01010110、11101001	————→ 机器数

2、 机器数的编码及运算

对带符号数而言，有原码、反码、补码之分，计算机内一般使用补码。

1) 原码

将数“数码化”，原数前“+”用0表示，原数前“-”用1表示，数值部分为该数本身，这样的机器数叫原码。

设 X——原数；则 $[X]_{原} = X (X \geq 0)$

$$[X]_{原} = 2^{n-1} - X (X < 0), n \text{ 为字长的位数。}$$

如， $[+3]_{原} = 00000011B$

$$[-3]_{原} = 2^7 - (-3) = 10000011B$$

0 有两种表示方法：00000000 ———→ +0

10000000 ———→ -0

原码最大、最小的表示：+127、-128

2) 反码

规定正数的反码等于原码；负数的反码是将原码的数值位各位取反。

$$[X]_{反} = X (X \geq 0)$$

$$[X]_{反} = (2^n - 1) + X (X < 0)$$

如， $[+4]_{反} = [+4]_{原} = 00000100 B$

$$[-4]_{反} = (2^8 - 1) + (-5) = 11111111 - 00000101 = 11111010 B$$

反码范围：-128 ~ +127

两个 0； +0 —— 00000000 B

-0 —— 11111111 B

3) 补码

补码的概念：现在是下午3点，手表停在12点，可正拨3点，也可倒拨9点。即是说-9的操作可用+3来实现，在12点里：3、-9互为补码。

运用补码可使减法变成加法。

规定：正数的补码等于原码。

负数的补码求法：1) 反码 + 1

$$2) \text{ 公式：} [X]_{补} = 2^n + X (X < 0)$$

如, 设 $X = -0101110\text{ B}$, 则 $[X]_{\text{原}} = 10101110\text{ B}$
 则 $[X]_{\text{补}} = [X]_{\text{反}} + 1 = 11010001 + 00000001 = 11010010\text{ B}$
 如, $[+6]_{\text{补}} = [+6]_{\text{原}} = 00000110\text{ B}$
 $[-6]_{\text{补}} = 2^8 + (-6) = 10000000 - 00000110 = 11111010\text{ B}$
 8 位补码的范围 $-128 \sim +127$ 。
 0 的个数: 只有一个, 即 00000000
 而 10000000 B 是 -128 的补码。
 原码、反码、补码对照表: 表 1-2 P₁₀

巩固新课:

小结:

- 1、计算机与单片机有何区别?
- 2、单片机主要有哪几部分组成?
- 3、真值与码值有何区别? 原码、反码、补码三者之间如何换算?

布置作业: P13 2、4

版书设计

一、单片机的概念 1、..... 2、..... 3、.....	二、单片机的组成 1、..... 2、...
三、 数制及转换 1、..... 2、.....	小结 作业:

第二讲 计算机中数的表示及运算

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、理解原码、反码、补码的概念
- 2、掌握真值、原码、反码及补码间的换算
- 3、熟悉 BCD 码和 ASCII 码

教学重点:

补码的概念及运算

BCD 码及 ASCII 码

教学难点:

补码的运算

复习:

原码、反码、补码三者之间有何关系?

4) 补码的运算

当 $X \geq 0$ 时, $[X]_{补} = [X]_{反} = [X]_{原}$

$[[X]_{补}]_{补} = [X]_{原}$

$[X]_{补} + [Y]_{补} = [X+Y]_{补}$

$[X-Y]_{补} = [X + (-Y)]_{补}$

例: 已知 $X=52$ $Y=38$ 求 $X-Y$

方法 1: 减法:

$$\begin{array}{r} X-Y = 52-38 = 14 \\ \quad 00110100 \\ -) 00100110 \\ \hline \quad 00001110 \end{array}$$

方法 2: 加法

$$\begin{aligned} X-Y &= [[X-Y]_{补}]_{补} = [[X]_{补} + [-Y]_{补}]_{补} \\ &= [[52]_{补} + [-38]_{补}]_{补} \\ &= [14]_{补} = 14 \end{aligned}$$

$$\begin{array}{r} [52]_{补}: \quad 00110100 \\ [-38]_{补}: +) 11011010 \\ \hline \quad \quad \quad 100001110 \end{array}$$

自然丢失

计算机在做算术运算时, 必需检查溢出, 以防止发生错误

5) 运算的溢出问题

资料字长 (位数) 有一定限制, 所以资料的表示应有一个范围。

如字长 8 位时; 补码范围 $-128 \sim +127$

若运算结果超出这个范围, 便溢出。

例:

$$\begin{array}{r} [98]_{补}: \quad 01100010 \\ \hline [25]_{补}: +) 00011001 \\ \hline [123]_{补}: \quad 01111011 \end{array} \quad \begin{array}{r} [85]_{补}: \quad 01010101 \\ \hline [47]_{补}: +) 00101111 \\ \hline [132]_{补}: \quad 10000100 \end{array}$$

未溢出	0 0	溢出	0 1
	Cs+1 Cs (未溢出)		Cs+1 Cs (溢出)
		错: 两个正数相加和为负数。	

$[-85]_{补}$: 1 0 1 0 1 0 1 1	$[-19]_{补}$: 0 1 0 1 0 1 0 1
$[-47]_{补}$: +) 1 1 0 1 0 0 0 1	$[-79]_{补}$: +) 1 0 1 1 0 0 0 1
$[-132]_{补}$: 1 0 1 1 1 1 1 0 0	$[-98]_{补}$: 1 1 0 0 1 1 1 1 0
溢出 1 0	未溢出 1 1
Cs+1 Cs	Cs+1 Cs

错: 两个负数相加和为正数。

可见: 结果正确(无溢出)时, $Cs+1 = Cs$

结果错误(溢出)时, $Cs+1 \neq Cs$

溢出判断: 溢出 = $Cs+1 \oplus Cs$ (即结果是 0 为无溢出; 1 为有溢出)

1、十进制数的编码

对机器: 二进制数方便,

对人 : 二进制数不直观, 习惯于十进制数。

在编程过程中, 有时需要采用十进制运算, 但机器不认识十进制数。

怎么办?

可以将十进制的字符用二进制数进行编码:

0	0000	9	1001	} 未用码
1	0001		1010	
2	0010		1011	
3	0011		1100	
4	0100		1101	
5	0101		1110	
6	0110		1111	
7	0111			
8	1000			

这叫做二进制数对十进制编码——BCD 码。

上述每 4 位二进制数表示一个十进制字符, 这 4 位中各位的权依次是:

8、4、2、1——8421 BCD 码。

BCD 码的运算:

例:

1 8		0 0 0 1 1 0 0 0
+) 3	BCD 码相加	+) 0 0 0 0 0 0 1 1
2 1	→	0 0 0 1 1 0 1 1
2 B		

非 BCD 码 (错) 怎么办?

需进行十进制调整:

	0 0 0 1 1 0 0 0	
+)	0 0 0 0 0 0 1 1	
	0 0 0 1 1 0 1 1	(个位大于 9, 应进 1)
+)	0 1 1 0	(加 6 使进 1)

————— 00100001 ————— (结果形如出 21)

(1) BCD 码加法规则

两个 BCD 数相加时，“某位”的和小于 10 则保持不变；
两个 BCD 数相加时，“某位”的和大于 9，则和数应加 6 修正。

(2) BCD 码减法规则

两个 BCD 数相减时，“某位”的差未发生借位，则差数保持不变；
两个 BCD 数相减时，“某位”发生了借位，其差应减 6 修正。
这里“某位”指 BCD 数中的“个位”、“十位”、“百位”、……

$\begin{array}{r} 0110\ 0101\ (65) \\ +) 1001\ 0111\ (97) \\ \hline 1111\ 1100 \\ +) 0110\ 0110\ (\text{加 } 66 \text{ 调整}) -) \\ \hline 10110\ 0010\ (162) \end{array}$	$\begin{array}{r} 0011\ 0100\ (34) \\ -) 0001\ 0101\ (15) \\ \hline 0001\ 1111 \\ 0110\ (\text{减 } 6 \text{ 调整}) \\ \hline 0001\ 1001\ (29) \end{array}$
---	--

四、 字符信息的表示

计算机能识别 0、1、0、1、……；这些 0、1、0、1、……有的代表数值，有的仅代表要处理的信息（如字母、标点符号、数字符号等文字符号），所以，计算机不仅要认识各种数字，还要能识别各种文字符号。人们事先已对各种文字符号进行二进制数编码：

如，美国信息交换标准码——ASCII 码，用一个字节表示一个字符。

低 7 位是字符的 ASCII 码值；最高位是通信时的校验位。

见 P328

巩固新课：

- 1、补码的作用？
- 2 为什么对 BCD 码要进行十进制调整？

布置作业： P₁₃： 5 、 6

版书设计

一、补码的运算 1、…… 2、……	二、运算溢出问题 1、…… 2、……	三、 十进制数的编码 1、…… 2、……
四、 字符信息的表示 1、…… 2、……	小结： 1、…… 2、…… 3、…… 4、……	作业：

第三讲 MCS-51 单片机结构原理

教学方法: 讲授法

教学目的

- 1、理解 MCS-51 单片机分类及特征
- 2、了解单片机内部组成
- 3、掌握 MCS-51 单片机引脚的使用

教学重点

单片机外部引脚的使用

教学难点

单片机外部引脚的功能

第二章 MCS-51 单片机结构原理

单片机硬件结构

- 内部结构
- 引脚功能
- 内存的配置
- CPU 时序
- I/O 接口

§2-1 概述

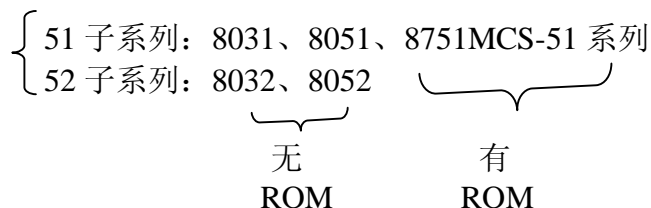
Intel MCS-51 系列单片机三个版本: 8031、8051、8751 (8 位机)

(表 2-1 P14 程序内存配置)

Intel MCS-96 系列机: 8096 (16 位机)

除此之外, Motorola 公司、Zilog 公司、Microchip相继推出产品, 各系列产品内部功能、单元组成、指令系统不尽相同。

Intel 公司单片机问世早, 系列齐全, 兼容性强, 所以得到广泛使用。



§2-2 MCS-51 单片机内部结构及引脚

一、MCS-51 单片机内部结构

P15 图 2-1

1、中央处理单元 (8 位)

数据处理、测试位, 置位, 复位 位操作

2、只读存储器 (4KB 或 8KB)

永久性存储应用程序, 掩模 ROM、EPROM、EEPROM

3、随机存取内存 (128B、128B SFR)

在程序运行时存储工作变量和资料

4、并行输入/输出口 (I/O) (32 条)

作系统总线、扩展外存、I/O 接口芯片

5、串行输入/输出口 (2 条)

串行通信、扩展 I/O 接口芯片

6、定时/计数器 (16 位、加 1 计数)

计满溢出、中断标志置位、向 CPU 提出中断请求, 与 CPU 之间独立工作

7、时钟电路

内振、外振。

8、中断系统

五源中断、2 级优先。

结构特点:

MCS-51 系列单片机为哈佛结构 (而非普林斯顿结构)

1) 内 ROM: 4KB

2) 内 RAM: 128B

3) 外 ROM: 64KB

4) 外 RAM: 64KB

5) I/O 线: 32 根 (4 埠, 每埠 8 根)

6) 定时/计数器: 2 个 16 位可编程定时/计数器

7) 串行口: 全双工, 2 根

8) 寄存器区: 工作寄存器区、在内 128B RAM 中, 分 4 个区,

9) 中断源: 5 源中断, 2 级优先

10) 堆栈: 最深 128B

11) 布尔处理机: 位处理机, 某位单独处理

12) 指令系统: 五大类, 111 条

二、MCS-51 单片机外部引脚

DIP	40 脚	} P17 图 2-3
方形封装	44 脚	

1、主电源引脚

V_{SS} 、 V_{CC}

2、外接晶振引脚

XTAL1 、 XTAL2

3、控制或复位引脚

RST / VPD 两个机器周期高电平, 单片机复位。

P0 ~ P3 口: 输出高电平

SP : 07H

SFR、PC: 清 0

不影响内 RAM 状态, 机器从 0 地址开始执行。

上电复位电路、电平方式开关复位电路如图 2-4 所示。

ALE / PROG : 地址锁存控制端

提供 $1/6 f_{osc}$ 振荡频率, 输入编程脉冲 EPROM

PSEN : 外部程序内存的读选通信号端。

EA / VPP : EA = 1 , 访问内部程序内存

当 PC 值超过内 ROM 范围 (0FFFH) 时, 自动转执行外部内存的程序
EA = 0 , 只访问外部程序内存。

对 8751 机, 可施加 21V 编程电源 (Vpp)

4、输入/输出引脚

P0 ~ P3: 四个 I/O 口, 每口 8 线, 共同 32 线。

小结:

1、MCS-51 系列机种类

2、MCS-51 内部结构有几部分? 特点?

3、MCS-51 机 DIP 有多少引脚? ALE、PSEN、EA 作用?

布置作业: P35 1

版书设计

<p>一、MCS-51 单片机内部结构 1、CPU..... 2、ROM..... 3、RAM</p>	<p>4、并行 I/O 口 5、串行 I/O 口..... 6、定时/计数器.....</p>	<p>7、时钟电路 8、中断系统.....</p>
<p>二、MCS-51 单片机内部结构 1、主电源 (VCC 、VSS) 2、外接晶振</p>	<p>3、 控制或复位 4、 输入/输出</p>	<p>小结: 1、..... 2、..... . 作业:</p>

第四讲 MCS-51 单片机存储器配置

教学方法: 讲授法

教学目的:

- 1、了解特殊功能寄存器名称
- 2、了解单片机位地址空间分配
- 3、掌握 MCS-51 单片机内部存储器的空间与分配

教学重点:

单片机外部引脚的使用

教学难点:

单片机外部引脚的功能

复习:

- 1、MCS-51 系列单片机有几种典型机型? 有何区别?
- 2、MCS-51 单片机有多少外部引脚? ALE、EA、PSEN、RET 的作用?
- 3、MCS-51 单片机的特点?

课题 §2-3 MCS-51 单片机内存配置

一、MCS-51 机的内存结构 (如图 1 所示)

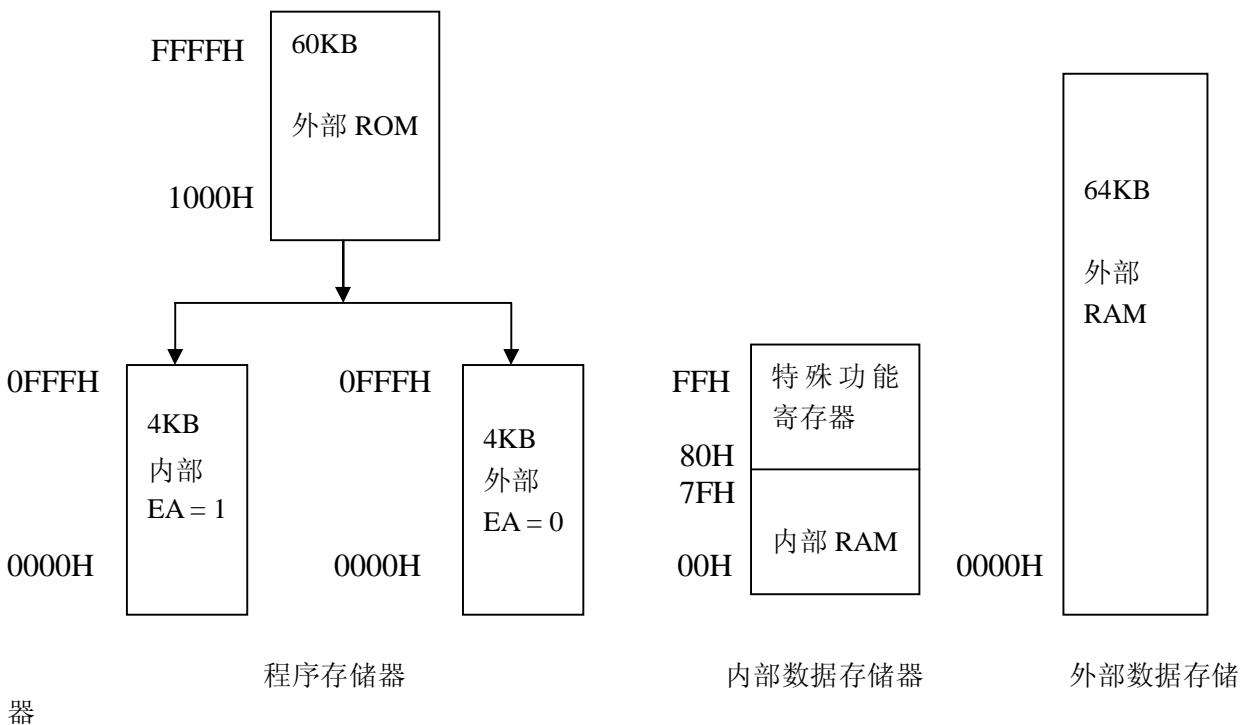


图 1 MCS-51 机的内存结构

物理上分为：4 个空间，片内 ROM、片外 ROM
 片内 RAM、片外 RAM
 逻辑上分为：3 个空间，程序内存（片内、外）统一编址 MOV C
 数据存储器（片内） MOV
 数据存储器（片外） MOVX

1、程序内存

寻址范围：0000H ~ FFFFH 容量 64KB

EA = 1，寻址内部 ROM；EA = 0，寻址外部 ROM

地址长度：16 位

作用：存放程序及程序运行时所需的常数。

七个具有特殊含义的单元是：0000H —— 系统复位，PC 指向此处；

0003H —— 外部中断 0 入口

000BH —— T0 溢出中断入口

0013H —— 外中断 1 入口

001BH —— T1 溢出中断入口

0023H —— 串口中断入口

002BH —— T2 溢出中断入口

2、内部数据存储器

物理上分为两大区：00H ~ 7FH 即 128B 内 RAM 和 SFR 区。如图 2 所示。

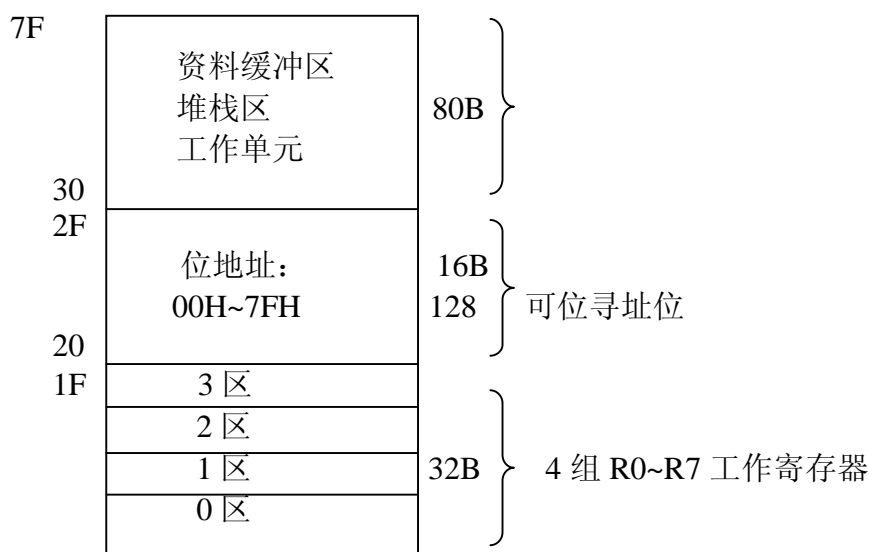


图 2 内部数据存储器

作用：作数据缓冲器用。

二、 殊功能寄存器SFR

寻址空间：80H ~ FFH ， 其中有 6 个双字节寄存器。

注意 PC 不在此范围内。

表 2-4 P22

表中：T2CO；TH2、TL2；RLDH、RLDL 属于 8032 / 8052。可位寻址的 SFR 为其地址能被 8 整除（表 2-4、表 2-5）

各寄存器的名称：

1、 算术运算寄存器

- (1) 累加器A (E0H)
- (2) B寄存器：乘、除法运算用
- (3) 程序状态字PSW寄存器：包含程序运行状态信息。

PSW	CY	AC	FO	RS1	RS0	OV	—	P
-----	----	----	----	-----	-----	----	---	---

CY —— 进位/借位标志；位累加器。

AC —— 辅助进/借位标志；用于十进制调整。

FO —— 用户定义标志位；软件置位/清零。

OV —— 溢出标志；硬件置位/清零。

P —— 奇偶标志；A中1的个数为奇数 P = 1；否则 P = 0。

RS1、RS0 —— 寄存器区选择控制位。

0 0 : 0区 R0 ~ R7

0 1 : 1区 R0 ~ R7

1 0 : 2区 R0 ~ R7

1 1 : 3区 R0 ~ R7

2、 指针寄存器

(1) 程序计数器PC

指明即将执行的下一条指令的地址，16位，寻址64KB范围，
复位时PC = 0000H

(2) 堆栈指针SP

指明栈顶元素的地址，8位，可软件设置初值，复位时SP = 07H

(3) 数据指针DPTR

@R0、@R1、@DPTR；指明访问的数据存储器的单元地址，16位，寻址范围64KB。 DPTR = DPH + DPL，也可单独使用。

3、 并行输入/输出端口

寄存器P0、P1、P2、P3实为相应端口锁存器。

4、 串行输入/输出端口

(1) 串行数据缓冲器 SBUF

是物理上独立的两个寄存器，共同使用一个地址。

(2) 串行控制/状态寄存器SCON

控制监视串行口的工作状态

- (3) 电源控制寄存器PCON
控制单片机的低功耗工作方式及波特率选择。

5、中断系统

- (1) 中断优先级寄存器IP: 2级优先, 可软件设定
- (2) 中断允许寄存器IE

6、定时/计数器

- (1) 定时器方式寄存器: TMOD
- (2) 定时器控制寄存器: TCON
- (3) 计数寄存器: TH0、TL0; TH1、TL1。可用于设定计数初值。

7、8052/8032增设专用寄存器

- (1) 定时器2控制寄存器T2CON; 控制、设置工作方式。
- (2) 计数寄存器: TH2、TL2
- (3) 定时器2捕获/重载寄存器: RCAP2H、RCAP2L
存放自动重载到TH2、TL2的数据。

三、位地址空间

- (1) 内部RAM 20H ~ 2FH共16个单元可按位寻址128位。
- (2) SFR 80H ~ FFH { 51子系列, 有21个寄存器, 83位;
52子系列, 有26个寄存器, 93位。

四、外部数据存储器

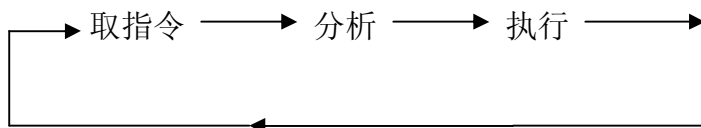
- (1) 容量最大扩展到64KB
- (2) 寻址范围: 0000H ~ FFFFH
- (3) 寻址方式: 间接寻址可用R0, R1及DPTR

§2-4 时序

时钟的基本概念

(投影胶片)

启动单片机后, 指令执行顺序



结合时序图说明单周期单字节指令在执行过程中ALE脉冲、取指操作、执行操作等在时间上的先后关系, 从而引出时序的概念:

单片机内的各种操作都是在一系列脉冲控制下进行的, 而各脉冲在时间上是有先后顺序的, 这种顺序就称为时序。

指令周期: 即从取指到执行完, 所需时间。

不同机器指令周期不一样; 即使相同机器, 不同的指令其指令周期也不一样。

机器周期: 机器的基本操作周期。

一个指令周期含若干机器周期(单、双、四周期)

状态周期: 一个机器周期分6个状态周期Si

每个状态周期含两个振荡周期, 即相位P1、P2。

振荡周期：由振荡时钟产生。

$$\text{振荡周期 } T_{osc} = 1/f_{osc}$$

$$\text{一个机器周期} = 12 \text{个振荡周期} = 12 \times 1/f_{osc} \text{。}$$

例如，若 $f_{osc} = 12\text{MHz}$ ，则一个机器周期 = $1 \mu\text{s}$ 。

时钟的产生：

XTAL1 (19)、XTAL (18)。

1、内部方式

与作为反馈元件的片外晶体或陶瓷谐振器一起组成一个自激振荡器。

(内振电路图) (外振电路图)

2、外部方式

CMOS工艺的8031，其XTAL1接外信号；XTAL2可悬空。

HMOS工艺的8031，其XTAL2接外信号；XTAL1接地。

小结：

MCS—51存储结构分配是怎样的？

SFR中各寄存器的名称？

布置作业：P35 2-5

版书设计

<p>一、MCS-51 单片机的内存结构</p> <p>1、片内 ROM....</p> <p>2、片外 ROM.....</p> <p>3、片内 RAM</p> <p>4、片外 RAM...</p>	<p>二、程序内存</p> <p>1、.....</p> <p>2、.....</p>	<p>三、内部数据存储器</p> <p>1、主电源 (VCC 、VSS)</p> <p>2、外接晶振</p>
<p>四、特殊功能寄存器</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p> <p>....</p>	<p>五、外部数据存储器</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>小结：</p> <p>1、.....</p> <p>2、.....</p> <p>作业：</p>

第五讲 并行I/O端口

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 8031 最小系统
- 2、掌握 P0、P1、P2、P3 口的使用
- 3、P0、P1、P2、P3 口的结构特点

教学重点:

P0、P1、P2、P3 口的使用

教学难点:

P0、P1、P2、P3 口的电路结构

复习: 1、MCS-51机的组成有几大部分?

2、外部引脚ALE、/PSEN、/EA、RST的功能是什么?

§ 2-5 并行I/O端口

四个端口、双向、每个口包含一个锁存器、一个输出驱动器和二个输入缓冲器。

一、P0 口

1、结构 (P0 口电路图)

2、通用 I/O 口

1) 读 (端口外数据 → 内部寄存器)

方式 1 (读锁存器) $\overrightarrow{Q} \xrightarrow{G2} \overrightarrow{D}$ 内部总线, 适于“读—修改—写”

方式 2 (读引脚): $P0 \cdot x \xrightarrow{G1} \overrightarrow{D}$ 内部总线。

作为通用 I/O 使用, 是一个准双向口:

“在输入数据时应先把口置 1, 使两个 FET 都截止, 引脚处于悬浮状态, 可作高阻抗输入”

2) 写 (片内数据 → 端口)

数据 → 锁存 → MUX → $P0 \cdot x$

3) 地址/数据总线口

控制 MUX

写: 地址/数据为 1, $P0 \cdot x$ ——高

地址/数据为 0, $P0 \cdot x$ ——低

读: 经缓冲器 G1 读入

4) 负载能力

可带 8 个 TTL 输入, 驱动 NMOS 时, 接上拉电阻。

总结特点:

(1) P0 口可作通用 I/O 口使用, 又可作地址/数据总线口;

(2) P0 既可按字节寻址, 又可按位寻址;

(3) P0 作为输入口使用时: 是准双向口;

(4) 作通用 I/O 口输出时: 是开漏输出;

(5) 作地址/数据总线口时, P0 是一真正双向口, 而作通用 I/O 口时, 只是一个准

双向口

二、P1 口

1、结构

(逻辑电路图)

只能作 I/O 口用, 且是一个准双向口。

内部已有上拉电阻，不是开漏输出口。

2、特点

- (1) 无地址/数据口功能
- (2) 可按字节寻址，也可按位寻址
- (3) 作 I/O 输入口时：是一准双向口，不是开漏输出。

三、P2 口

1、结构

2、特点

- (1) 当 P2 口作为通用 I/O 时，是一准双向口。
- (2) 从 P2 口输入数据时，先向锁存器写“1”。
- (3) 可位寻址，也可按字节寻址
- (4) 可输出地址高 8 位。

四、P3 口

1、结构图

2、特点

- (1) 作通用 I/O 时，“选择输出功能”应保持高电平，

(2) 工作于第二功能时，该位锁存器应置 1，

(3) 作输入口时，输出锁存器和选择输出功能端都应置 1

(4) 第二功能专用输入，取自输入通道第一缓冲器 (G1) 输出端，通用输入信号取自“读引脚”。

小结：

1、P0 口：地址低 8 位与数据线分时使用端口，

2、P1 口：按位可编址的输入输出端口，

3、P2 口：地址高 8 位输出口

4、P3 口：双功能口。若不用第二功能，也可作通用 I/O 口。

5、按三总线划分：

地址线：P0 低八位地址，P2 高八地址；

数据线：P0 输入输出 8 位数据；

控制线：P3 口的 8 位加上/PSEN、ALE 共同完成控制总线。

布置作业： P35 2-2, 2-4。

版书设计

一、P0 口 1、..... 2、.... 3、 4、	二、P1 口 1、..... 2、..... 3、..... 4、.	三、P2 口 1、..... 2、..... 3、..... 4、
四、P3 口 1、 2、... 3、 4、.	小结: 1、..... 2、..... 3、 4、. 作业:	

第六讲 MCS-51 指令系统及汇编语言程序设计

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 MCS-51 指令的分类
- 2、掌握 MCS-51 汇编语言指令格式
- 3、训练掌握六种寻址方式

教学重点:

六种寻址方式

教学难点:

间接寻址、基址+变址寻址

第三章 MCS-51 指令系统及汇编语言程序设计

§ 3-1 汇编语言与指令系统简介

一、汇编语言程序设计的意义

什么是程序？

完成某项特定任务的指令的集合。

计算机按程序一条一条地依次执行指令，从而完成指定任务。

要让计算机完成各项任务，就应设计各种程序。

程序设计语言：

机器语言

汇编语言

高级语言

汇编语言程序的每一条语句都与计算机的某一条指令对应，所以必需熟悉指令系统。

指令 = 操作码 + 操作数

操作码——表示了该指令所能执行的操作功能。

操作数——表示参加操作的数的本身或操作数所在的地址。

MCS-51 指令格式：

[标号：] 操作码助记符 [第一操作数][，第二操作数][；注释]

二、MCS-51 系列单片机的指令系统

111 条指令，共分五大类：

数据传送类；（29 条）

算术运算类；（24 条）

逻辑运算类；（24 条）

控制转移类；（17 条）

位操作类。（17 条）

指令中操作数的描述符号:

Rn —— 工作寄存器 R0 ~ R7

Ri —— 间接寻址寄存器 R0、R1

Direct —— 直接地址, 包括内部 128B RAM 单元地址、26 个 SFR 地址。

#data —— 8 位常数

#data 16 —— 16 位常数

addr 16 —— 16 位目的地址

addr 11 —— 11 位目的地址

rel —— 8 位带符号的偏移地址

DPTR —— 16 位外部数据指针寄存器

bit —— 可直接位寻址的位

A —— 累加器

B —— 寄存器 B

C —— 进、借位标志位, 或位累加器

@ —— 间接寄存器或基址寄存器的前缀

/ —— 指定位求反

(x) —— x 中的内容

((x)) —— x 中的地址中的内容

\$ —— 当前指令存放的地址

三、寻址方式

寻找操作数存放单元的地址的方式，共 6 种方式。

1、立即数寻址

所要找的操作数是一二进制数或十进制数，出现在指令中，用“#”作前缀

```
MOV A, #20H
```

2、寄存器寻址

操作数存放在工作寄存器 R0 ~ R7 中，或寄存器 B 中。

```
MOV A, R2
```

3、直接寻址

指令中直接给出操作数的地址。

```
MOV A, 30H
```

```
MOV 30H, DPH
```

4、寄存器间接寻址

指令中寄存器的内容作为操作数存放的地址，指令中间接寻址寄存器前用

“@”表示前缀。

举“两个抽屉，两把钥匙”的例子。

```
MOV R0, #30H
```

```
MOV A, @R0
```

```
MOV A, #20H
```

```
MOV R1, #40H
```

```
MOV @R1, A
```

5、变址寻址

操作数地址 = 变地址 + 基地址

基地址寄存器 DPTR 或 PC

变址寄存器 @A

该寻址方式常用于访问程序存储器，查表。

```
MOV A, @A + DPTR
```

6、相对寻址

把指令中给定的地址偏移量与本指令所在单元地址（PC 内容）相加得到真正有效的操作数所存放的地址。

举“李同学 20 岁，张同学比李同学大 3 岁”的例子。

JC 60H ; 设 (PC) = 2000H,

则当 C = 1 时,

转移的目的地址 = (PC) + 2 + 60H

巩固新课:

- 1、汇编语言有何种特点？指令格式？
- 2、MCS—51 系列机共有多少条指令？分几类？
- 3、MCS—51 系列汇编语言指令有几种寻址方式？

版书设计

<p>一、汇编语言程序设计的意义</p> <p>1、.....</p> <p>2、.....</p> <p>3、</p>	<p>二、MCS—51 单片机指令系统.....</p> <p>1、.....</p> <p>2、.....</p> <p>3、</p>
<p>三、寻址方式</p> <p>1、.....</p> <p>2、.....</p> <p>3、</p>	<p>4、.....</p> <p>5、.....</p> <p>6、</p> <p>小结:</p> <p>作业</p>

第七讲 数据传送类指令

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、熟悉数据传送的三种方式及特点
- 2、熟悉 ROM 数据传送 (查表)
- 3、熟练掌握 RAM 数据传送

教学重点:

RAM 数据传送

教学难点:

堆栈操作指令、查表指令

复习: 1、MCS-51 系列指令格式?

2、六种寻址方式分别是什么?

§ 3-2 数据传送类指令

一、内部 RAM 数据传送类

1、一般数据传送指令

回忆前述操作数描述符: A, direct, Rn, @Ri, #data, DPTR 等。

经排列组合可写出下列指令:

MOV A, Rn ; (~~A~~) — (Rn) 以下类似。

MOV A, direct

MOV A, @Ri

MOV A, #data

MOV Rn, A

MOV Rn, direct

MOV Rn, #data

MOV direct, A

MOV direct, Rn

MOV direct, direct

MOV direct, @Ri

MOV direct, #data

MOV @Ri, A

MOV @Ri, direct

MOV @Ri, #data

MOV DPTR, #data16 ; 唯一的 16 位数据传送指令。

上述指令不影响任何标志位, 但 PSW 的 P 位除外。

注意: MOV Rn, Rn

MOV @Ri, @Ri

MOV Rn, @Ri

MOV #data, A

等指令是非法指令。

2、栈操作指令

PUSH direct

POP direct

不影响任何标志位。

PUSH direct 指令执行中, 机器自动进行两步操作:

(1) $(SP) \leftarrow (SP) + 1$

(2) $((SP) \rightarrow (direct))$

例 1、设 $(SP) = 09H$, $(DPTR) = 0123H$, 分析:

执行 PUSH DPL

PUSH DPH

后, 各单元中的内容。

POP direct 指令执行中, 机器也自动进行两步操作:

(1) $(direct) \leftarrow ((SP))$

(2) $(SP) \leftarrow (SP) - 1$

例 2、设 (SP) = 0BH, (0BH) = 01H, (0AH) = 23H

执行 POP DPH

POP DPL

后, 各单元中的内容。

结论: 1) PUSH 与 POP 操作过程刚好相反;

2) 进、出栈规则:

先进后出, 后进先出。应注意指令书写先后顺序;

3) 可用于“保护现场, 恢复现场”。

3、字节交换指令

XCH A, Rn	}	整字节交换
XCH A, direct		
XCH A, @Ri		
XCHD A, @Ri	}	半字节交换
SWAP A		

不影响任何标志位。

二、外部 RAM 数据传送

MOVX A, @Ri

MOVX A, @DPTR

MOVX @Ri, A

MOVX @DPTR, A

执行过程中会使 /WR、/RD 有效。

例 3、试编写一程序段, 实现将外 RAM 0FAH 单元中的内容传送到外 RAM 04FFH 单元中。

解:

```

MOV DPTR, #04FFH
MOV R0, #0FAH
MOVX A, @R0
MOVX @DPTR, A

```

三、查表指令

与 ROM 之间的数据传送。

```
MOVC A, @A+DPTR
```

```
MOVC A, @A+PC
```

执行后会使/PSEN 有效。

MOVC 含义是传送常数。

以 DPTR 为基地址的指令，可在 ROM 的 64KB 范围内查表；

而以 PC 为基地址的指令只能在 (PC) + 1 为中心上、下 256B 范围内查表。

例 4、设 (A) = 一个 BCD 码常数，试用查表法获得其相应的 ASCII 码。

解法 I:

```
MOV DPTR, #TAB
```

```
MOVC A, @.A+DPTR
```

•

•

•

```
TAB: DB 30H
```

```
DB 31H
```

```
DB 32H, 33H, 34H, 35H
```

•••••

解法 II:

```
MOVC A, @A+PC
```

```
TAB: DB 30H, 31H, 32H, 33H
```

```
DB 34H, 35H, 36H, 37H
```

•••••

小结:

数据传送类指令有几种类型？

是否影响标志位？

版书设计

<p>一、内部 RAM 数据传送</p> <p>1、.....</p> <p>2、....</p> <p>3、</p>	<p>三、查表指令</p> <p>1、</p> <p>2、.....</p> <p>3、.....</p>
<p>二、外部 RAM 数据传送</p> <p>1、主电源（VCC 、VSS）</p> <p>2、外接晶振</p> <p>....</p>	<p>小结：</p> <p>1、.....</p> <p>2、.....</p> <p>.</p> <p>作业：</p>

第八讲 算术运算类指令

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、熟悉数据运算类指令的操作方式特点
- 2、理解二-十进制调整指令的含义
- 3、熟练掌握加法指令

教学重点:

加法、减法指令

教学难点:

二-十进制调整指令

复习:

direct、@Ri、@DPTR、Rn、#data、(x)、((x)) 的含义?

新授课: § 3-3 算术运算类指令

包括: 加、减、乘、除; 加一、减一。

一、加法指令

ADD A, Rn ; (A) ← (A) + (Rn) 以下类同。

ADD A, direct

ADD A, @Ri

ADD A, #data

无符号数相加时: 若 C = 1, 说明有溢出 (其值 > 255)。

带符号数相加时: 若 OV = D7c ⊕ D6c = 1, 说明有溢出。

ADDC A, Rn ; (A) ← (A) + (Rn) + (C), 以下类同。

ADDC A, direct

ADDC A, @Ri

ADDC A, #data

上述四条指令多用于多字节数相加。

INC A ; ~~(A)~~ (A) + 1 , 以下类同。

INC Rn

INC direct

INC @Ri

INC DPTR

例 1、设 (R0) = 7FH; (7EH) = 40H

执行: INC @R0

INC R0

INC @R0 后,

(R0) = 7FH; (7EH) = 00H; (7FH) = 41H.

DA A ; 二——十进制调整指令。

执行过程中, CPU 能根据加法运算后, 累加器中的值和 PSW 中的 AC 及 C 标志位的状况自动选择一个修正值 (00H、06H、60H、66H) 与原运算结果相加, 进行二——十进制调整。

选择修正值的规则:

(A3 ~ 0) > 9 时或 (AC) = 1 时, (A3 ← 0) (A3 ~ 0) + 6

(A7 ~ 4) > 9 或 (C) = 1 时, (A7 ← 4) (A7 ~ 4) + 6

例 2、设 (A) = 56H 为 56 的压缩的 BCD 码数, (R3) = 67H, (CY) = 1

执行 ADDC A, R3

DA A

结果为: 124

注意: 1) DA 指令只能跟在加法指令后面使用;

2) 调整前参与运算的两数是 BCD 码数;

3) DA 指令不能与减法指令配对使用, 但可以实现对 A 中压缩 BCD 减一操作。

例 3、设 (A) = 30H (压缩 BCD 码数), 执行:

```
ADD A, #99H
```

```
DA A
```

后，便实现了 $30 - 1 = 29$ 的操作。

例 4、两个 4 位 BCD 码相加，一个存放在 (31H) (30H)；另一个存放在 (33H) (32H)；和数拟回存在 (31H) (30H) 中，试编程实现之。

```
解: MOV R0, #30H
     MOV R1, #32H
     MOV A, @R0
     ADD A, @R1
     DA A
     MOV @R0, A
     INC R0
     INC R1
     MOV A, @R0
     ADDC A, @R1
     DA A
     MOV @R0, A
```

二、减法指令

SUBB A, Rn ; $\leftarrow (A) - (Rn) - (C)$ ，以下类同。

SUBB A, direct

SUBB A, @Ri

SUBB A, #data

注意：减法之前先清零 C。

DEC A ; $(A) \leftarrow (A) - 1$ ，以下类同。

DEC Rn

DEC @Ri

DEC direct

例 5、设 (R0) = 7FH，在内 RAM 中，(7EH) = 00H，(7FH) = 40H

执行: DEC @R0

DEC R0

DEC @R0

结果为: (R0) = 7EH, (7EH) = 0FFH, (7FH) = 3FH。

三、乘法和除法指令

乘法:

MUL AB ; (A) × (B), 积的低 8 位在 A 中, 积的高 8 位在 B 中;
C 总为 0。

除法:

DIV AB ; (A) ÷ (B), 商在 A 中, 余数在 B 中。

若 (B) = 0, 则结果不定, (OV) = 1, (C) = 0。

例 6、试将 A 中的二进制数转换为 3 位 BCD 码, 其中, 百位数存放于 31H 单元, 十位数和个位数压缩后存于 30H 单元中。

解: MOV B, #100

DIV AB

MOV 31H, A

MOV A, #10

XCH A, B

DIV AB

SWAP A

ADD A, B

MOV 30H, A

小结: 算术运算类指令都有哪些类型?

版书设计

一、加法指令 1、.... 2、..... 3、 4、...	二、减法指令 1、..... 2、.....	三、乘法指令 1、主电源（VCC、VSS） 2、外接晶振
四、除法指令 1、..... 2、..... 3、.....	小结: 1、..... 2、..... 3、 作业:	

第九讲 逻辑操作类指令（一）

教学方法: 讲授法

教学时数: 2 学时

教学目的:

- 1、了解控制 转移类指令的作用
- 2、掌握无条件 转移指令的应用
- 3、掌握有条件转移指令的用法

教学重点:

无条件 转移指令的应用

教学难点:

转移地址的确定

复习: 111 条指令，共分几大类？ 数据传送类指令又分成多少类？

§ 3-4 逻辑操作类指令

共分两大类：单字节逻辑操作，双字节逻辑操作，共 24 条。

一、单字节逻辑操作指令

CLR A ; (A) ← 0

CPL A A 中 8 位按位求反。

循环左移、右移指令：

RL A

RLC A

RR A

RRC A

注：左移一位相当于乘 2；右移一位相当于除 2。

二、双字节逻辑操作指令

“与操作”：

ANL A, Rn ; ~~(A)~~ (A) ∧ (Rn)，以下类同。

ANL A, direct

ANL A, @Ri
 ANL A, #data
 ANL direct, A
 ANL direct, #data

例 1、(P1) = 35H, 使其高 4 位输出 0, 低 4 位不变。

解; ANL P1, #0FH

此做法称为“屏蔽”位。

“或操作”:

ORL A, Rn ; $(A) \vee (Rn)$, 以下类同。
 ORL A, direct
 ORL A, @Ri
 ORL A, #data
 ORL direct, A
 ORL direct, #data

例 2、将 A 中的低 3 位送入 P1 中, 并且保持 P1 中高 5 位不变。

ANL A, #07H

ANL P1, #0F8H

ORL P1, A ; (P1) = P₁₇P₁₆P₁₅P₁₄P₁₃A₂A₁A₀

这称为“数位组合”。

“异或操作”:

XRL A, Rn ; $(A) \oplus (Rn)$, 以下类同。
 XRL A, direct
 XRL A, @Ri
 XRL A, #data
 XRL direct, A
 XRL direct, #data

例 3、设 (P1) = 0B4H = 10110100B, 执行:

XRL P1, #00110001B

结果按#00110001取反, 即:

(P1) = 10000101B = 85H

这称为“指定位取反”。

在上述 ANL、ORL、XRL 操作中, 用于端口操作时, 无论 P0 ~ P3 是第一, 还是第二操作数, 都遵循“读—修改—写”端口锁存器的操作。

小结:

- 1、逻辑操作分为哪两类操作?
- 2、循环移位指令每执行一次, 移几位?
- 3、ANL、ORL、XRL 指令有哪些使用技巧?

版书设计

<p>一、条件转移指令</p> <p>1、.....</p> <p>2、.....</p> <p>3、</p>	<p>二、调用指令</p> <p>1、.....</p> <p>2、.....</p>	<p>三、返回指令</p> <p>1、</p> <p>2、</p>
<p>小结:</p> <p>1、.....</p> <p>2、.....</p>	<p>作业:</p>	

第十讲 控制转移类指令

教学方法: 讲授法

教学

教学目的:

- 1、了解控制转移类指令的种类
- 2、掌握无条件转移指令的特点及应用
- 3、掌握调用指令的特点及应用

教学重点:

各类指令操作功能

教学难点:

循环、移位指令

复习: 逻辑操作指令, 单字节: CLR, CPL, RL, RLC, RR, RRC

双字节: ANL, ORL, XRL。

§ 3-5 控制转移类指令

作用: 改变程序计数器 PC 的值, 从而改变程序执行方向。

分为四大类: 无条件转移指令; 条件转移指令; 调用指令; 返回指令。

一、无条件转移指令

LJMP addr16

AJMP addr11

SJMP rel

JMP @A + DPTR

LJMP addr16 ; 长跳转

转移目的地址 $\xrightarrow{\text{addr16}}$ (PC); 0000H ~ FFFFH, 64KB

AJMP addr11 ; 绝对转移

转移目的地址的形成: 先 $\xleftarrow{\text{PC}}$ (PC) + 2;

后 PC15 ~ 11 不变, $\xleftarrow{\text{addr10}} \text{PC10} \sim 0$ addr10 ~ 0

$$64\text{KB} = 2^{16} = 2^5 \times 2^{11} = 32 \times 2\text{KB}$$

转移目的地址与 (PC) + 2 在同一个 2KB 范围内。

SJMP rel ; 短转移, 相对寻址。

转移目的地址 = (PC) + 2 + rel,

所以 rel = 转移目的地址 - (PC) - 2

但, 实际使用中常写成 SJMP addr16, 汇编时会自动转换出 rel。

JMP @A + DPTR ; 间接转移, 散转移指令。

转移目的地址 = (@A) + (DPTR)

本指令不影响标志位, 不改变 @A 及 DPTR 中的内容。常用于多分支程序结构中, 可在程序运行过程中动态地决定程序分支走向。

例 1、设 A 中为键值, 试编写按键值处理相应事件的程序段。

解: MOV DPTR, #KYEG

 MOV B, #03H

 MUL AB

 JMP @A + DPTR

 .

 KYEG: LJMP KYEG0

 LJMP KYEG1

 .

画图比较 LJMP、AJMP、SJMP、JMP 转移的起点和范围。

二、调用指令

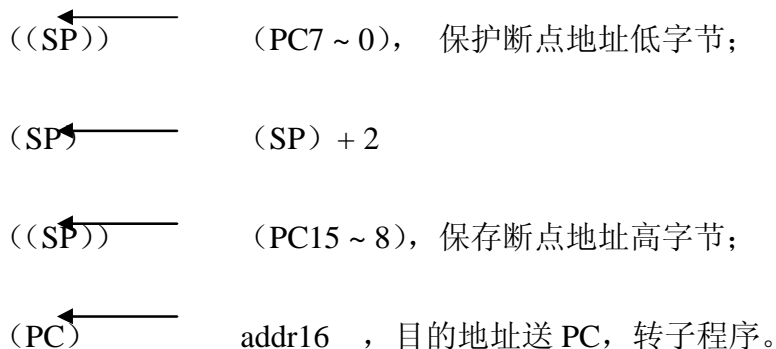
LCALL addr16 ; 长调用

ACALL addr11 ; 绝对调用

LCALL addr16 ; 转移范围 64KB, 不影响标志位。执行中自动完成如下过程:

 (PC) ← (PC) + 3

 (SP) ← (SP) + 1



例 2、设 $(SP) = 07H$, $(PC) = 2100H$, 子程序首地址为 $3456H$, 执行:

```

LCALL 3456H

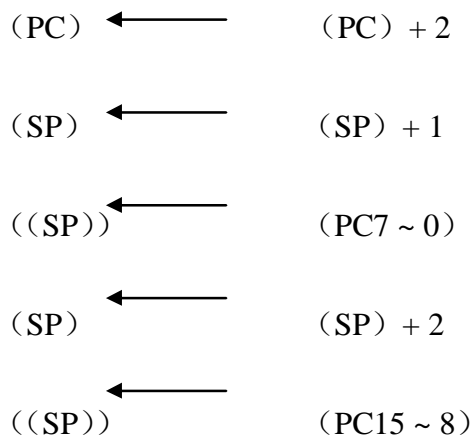
MOV A, 20H

.....
    
```

画出执行过程示意图。

执行结果: $(SP) = 09H$, $(09H) = 21H$, $(08H) = 03H$, $(PC) = 3456H$

ACALL addr11 ; 转移范围与 $(PC) + 2$ 在同一个 2KB 内。不影响任何标志位, 执行中机器自动完成下列过程:



(PC10 ←) addr10 ~ 0

三、返回指令

从子程序返回主程序。

RET ; 调用子程序返回;

RETI ; 中断子程序返回。

画图比较两种返回指令含义上的异同点, 结论: **RET** 返回地址事先已知, 而

RETI 的返回地址在程序执行中产生的, 不固定。

不影响标志位, 但 **PSW** 不能恢复到中断前的状态。

其机器自动操作过程如下:

(PC15 ←) ((SP))
 (SP) ← (SP) - 1
 (PC7 ←) ((SP))
 (SP) ← (SP) - 2

例 3、设 (SP) = 0BH, (0AH) = 23H, (0BH) = 01H

执行: **RET**

结果: (SP) = 09H, (PC) = 0123H (返回主程序)

空操作指令

NOP ; 空操作

不执行任何操作，仅仅使(PC)+1，继续执行下条指令，不影响标志位，

在ROM中占一个字节。用于延时调整。

小结:

- 1、长转移与长调用指令有何区别？
- 2、长转移、短转移及相对转移的跳转范围有什么不同？

作业：P96 3.4 3.5

版书设计

一、无条件转移指令 1、..... 2、..... 3、	二、调用指令 1、..... 2、.....
三、返回指令 1、..... 2、.....	小结： 1、..... 2、..... .作业：

第十一讲 控制转移类指令

教学方法: 讲授法

教学目的:

- 1、了解控制转移类指令的分类
- 2、掌握条件转移指令的特点及应用

教学重点:

条件转移指令的特点及应用

教学难点:

条件转移指令的特点及应用

复习: 1、控制转移指令是如何分类的?

2、无条件转移指令、调用指令分是哪几个?

二、条件转移指令

实现按照一定条件决定转移的方向。分三类。

1、判零转移

JZ rel

JNZ rel

JZ rel ; 若 $(A) = 0$, 则转移, 否则顺序执行。

JNZ rel ; 若 $(A) \neq 0$, 则转移, 否则顺序执行。

转移目的地址 = $(PC) + 2 + rel$

不影响任何标志位。

例 1、将外 RAM 的一个数据块（首地址为 DATA1）传送到内部数据 RAM（首地址为 DATA2），遇到传送的数据为零时停止传送，试编程。

```
解：          MOV  R0, #DATA2

              MOV  DPTR, #DATA1

LOOP1:  MOVX  A, @DPTR

              JZ   LOOP2

              MOV  @R0, A

              INC  R0

              INC  DPTR

              SJMP LOOP1

LOOP2:  SJMP  LOOP2
```

2、比较转移指令

功能：比较二个字节中的值，若不等，则转移。

```
CINE  A, #data, rel
```

```
CJNE  A, direct, rel
```


CJNE @Ri, #data, rel

CJNE Rn, #data, rel

该类指令具有比较和判断双重功能，比较的本质是做减法运算，用第一操作数内容减去第二操作数内容，但差值不回存。

转移目的地址 = (PC) + 3 + rel

若第一操作数内容小于第二操作数内容，则 (C) = 1，否则 (C) = 0。

该类指令可产生三支程序：

即，相等分支；大于分支；小于分支。

例 2、设 P1 口的 P1.0 ~ P1.3 为准备就绪信号输入端，当该四位为全 1 时，说明各项工作已准备好，单片机可顺序执行，否则，循环等待。

解： MOV A, P1

 ANL A, #0FH

 CJNE A, #0FH, WAIT ; P1.0 ~ P1.3 不为全 1 时，返回 WAIT

 MOV A, R2

3、循环转移指令

DJNZ Rn, rel ; (二字节指令)

DINZ direct, rel ; (三字节指令)

本指令也为双功能指令，即减 1 操作和判断转移操作。

第一操作数内容减 1 后，若差值不为零，则转移；否则顺序执行。

转移目的地址 = (PC) + 2 或 3 + rel

例 3、将 8031 内部 RAM 的 40H ~ 4FH 单元置初值 #A0H ~ #AFH。

```
解:      MOV  R0, #40H

          MOV  R2, #10H

          MOV  A, #0A0H

LOOP:    MOV  @R0, A

          INC  R0

          INC  A

          DJNZ R2, LOOP

          .....
```

小结: 1、无条件转移指令共有几条?

2、CJNE 指令与 DJNZ 指令有何区别？

作业：作业：P96 3.6 3.7

版书设计

一、判零转移 1、.... 2、..... 3、	二、比较转移指令 1、..... 2、.....
三、循环转移指令 1、..... 2、.....	小结： 1、..... 2、..... .作业：

第十二讲 位操作指令

教学方法: 讲授法

教学时数: 2 学时

教学目的:

- 1、了解位传送、位控制转移指令的功能
- 2、掌握位操作指令的使用

教学重点:

位传送、位控制转移

教学难点:

布尔条件判断转移指令

复习: 1、控制转移指令分为几类?

2、LJMP、AJMP、SJMP、JMP 属于哪一类?

LCALL、ACALL 属于哪一类?

3、无条件转移指令的转移条件分别是什么?

4、可按位寻址的空间是怎样的?

§ 3-6 位操作指令

包括: 位传送指令、条件转移指令、位运算指令。

位操作由单片机内布尔处理器来完成。

位地址的四种表示:

- 1) 使用直接位地址表示: 如 20H、30H、33H 等;
- 2) 使用位寄存器名来表示; 如 C、OV、F0 等;
- 3) 用字节寄存器名后加位数来表示: 如 PSW.4、P0.5、ACC.3 等;

4) 字节地址加位数来表示: 如 20.0、30.4、50.7 等。

一、位传送指令

MOV C, bit

MOV bit, C

功能: (C) \longleftrightarrow (bit);

二、位状态控制指令

CLR bit ; (bit) \longleftarrow 0

SETB bit ; (bit) \longleftarrow 1

CPL bit ; (bit) \longleftarrow (/ bit)

例 1、编程通过 P10 线连续输出 256 个宽度为 5 个机器周期长的方波。

解: MOV R0, #00H

 CLR P10

 LOOP: CPL P10

 NOP

 NOP

DJNZ R0, LOOP

三、位逻辑操作指令

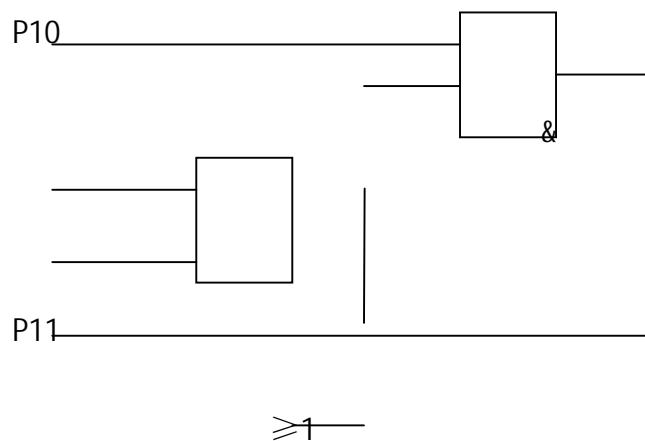
ANL C, bit ; (C) ← (C) ∧ (bit)

ANL C, /bit ; (C) ← (C) ∧ (/bit)

ORL C, bit ; (C) ← (C) ∨ (bit)

ORL C, /bit ; (C) ← (C) ∨ (/bit)

例 2、用软件实现下图所示的 P10 ~ P13 间的逻辑运算。



P12

P13

解: MOV C, P11

 ORL C, P12

```
ANL  C, P10
```

```
MOV  P13, C
```

四、布尔条件转移指令

有 5 条，分别对 C 和直接位地址进行测试，并根据其状态执行转移。

1、判布尔累加器转移

JC rel ; (C) = 1, 转移, 否则顺序执行。

JNC rel ; (C) = 0, 转移, 否则顺序执行。

不影响标志。转移地址 : (PC) ← (PC) + rel

例 3、比较内部 RAM 的 30H 和 40H 单元中的二个无符号数的大小，将大数存入 20H 单元，小数存入 21H 单元，若二数相等，则使内 RAM 的第 127 位置 1。

```
解:      MOV  A, 30H

          CJNE A, 40H, LOOP

          SETB 7FH

          SJMP $

LOOP1:   JC   LOOP2

          MOV  20H, A

          MOV  21H, 40H
```

```
                SJMP $  
  
LOOP2:  MOV  20H, 40H  
  
                MOV  21H, A  
  
                SJMP $
```

2、判位变量转移

JB bit, rel ; (bit) = 1, 则转移, 否则顺序执行。

JBC bit, rel ; (bit) = 1, 则转移, 否则顺序执行, 且无论
(bit) 是否等于 1, 均使该位清零。

JNB bit, rel ; (bit) = 0, 则转移, 否则顺序执行。

不影响标志。

例 4、试判断 A 中的正负, 若为正数, 存入 20H 单元; 若为负数则存入 21H 单元。

解:

```
                JB  ACC7, LOOP  
  
                MOV  20H, A  
  
                SJMP $  
  
LOOP:  MOV  21H, A  
  
                SJMP $
```


小结:

- 1、位操作指令分为几类?
- 2、位累加器 C 与字节累加器 A 有何区别?

布置作业: 题目抄在黑板上。

版书设计

一、位传送指令 1、片内 ROM.... 2、片外 ROM..... 3、片内 RAM 4、片外 RAM...	二、位状态控制指令 1、..... 2、.....	三、位逻辑操作指令 1、主电源 (VCC 、VSS) 2、外接晶振
四、布尔条件转移指令 1、..... 2、..... 3、.....	小结: 1、..... 2、..... 作业:	

第十三讲 汇编语言及汇编过程

教学方法: 讲授法

教学时数: 2 学时

教学目的:

- 1、理解源程序、目标代码、编辑、汇编等含义
- 2、了解汇编过程
- 3、掌握伪指令的使用

教学重点:

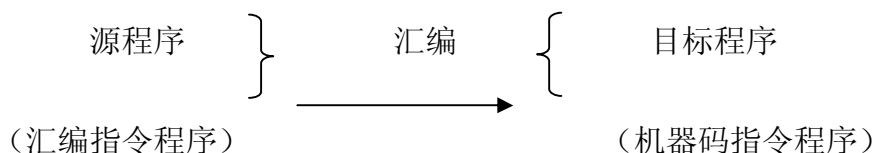
伪指令的使用

教学难点:

汇编过程

§ 3-7 汇编语言及汇编过程

一、汇编程序功能



汇编指令与机器码指令有一一对应的关系。

汇编程序是一种翻译程序，将源程序翻译成目标程序。

二、汇编程序的汇编过程

汇编有两种方法：手工汇编、机器汇编。

1、手工汇编：

第一次汇编：确定地址，翻译成各条机器码，字符标号原样写出；

第二次汇编：标号代真，将字符标号用所计算出的具体地址值或偏移量代换。

源程序	地址	目标程序	
		第一次汇编	第二次汇编
ORG 1000H			
START: MOV R0, BUFFER-1	1000	A8 <u>2F</u>	A82F
MOV R2, #00H	1002	7A00	7A00
MOV A, @R0	1004	E6	E6
MOV R3, A	1005	FB	FB
INC R3	1006	0B	0B
SJMP NEXT	1007	80NEXT	80 <u>05</u>
LOOP: INC R0	1009	08	08
CJNE @R0, #44H, NEXT	100A	B644NEXT	B644 <u>01</u>
INC R2	100D	0A	0A
NEXT: DJNZ R3, LOOP	100E	DBLOOP	DB <u>F9</u>

```
MOV RESULT, R2          1010    8A2A    8A2A
SJMP $                  1012    80FE    80FE

BUFFER DATA 30H

RESULT DATA 2AH

END
```

2、机器汇编

两次扫描过程。

第一次扫描：检查语法错误，确定符号名字；

建立使用的全部符号名字表；

每一符号名字后跟一对应值（地址或数）。

第二次扫描：是在第一次扫描基础上，将符号地址转换成真地址（代真）；

利用操作码表将助记符转换成相应的目标码。

三、伪指令

伪指令是告诉汇编程序，如何汇编源程序的指令。

伪指令既不控制机器的操作，也不能被汇编成机器代码，故称为伪指令。

1、起始地址伪指令 ORG

ORG addr16

用于规定目标程序段或数据块的起始地址，设置在程序开始处。

2、汇编结束伪指令 END

告诉汇编程序，对源程序的汇编到此结束。一个程序中只出现一次，在末尾。

3、赋值伪指令 EQU

告诉汇编程序，将汇编语句操作数的值赋予本语句的标号。

格式： 标号名称 EQU 数值或汇编符号

“标号名称”在源程序中可以作数值使用，也可以作数据地址、位地址使用。

先定义后使用，放在程序开头。

4、定义字节伪指令 DB

告诉汇编程序从指定的地址单元开始，定义若干字节存储单元并赋初值。

格式： [标号:] DB 字节数据或字节数据表

5、定义字伪指令 DW

从指定地址开始，定义若干个 16 个位数据，高八位存入低地址；低八位存入高地址。

例、 ORG 1000H

 PIOI: DW 7654H, 40H, 12, 'AB'

6、数据地址赋值伪指令 DATA

将表达式指定的数据地址赋予规定的字符名称

格式: 字符名称 DATA 表达式

注: 该指令与 EQU 指令相似, 只是, 可先使用后定义, 放于程序开头、结尾均可。

7、定义空间伪指令 DS

从指定地址开始, 保留由表达式指定的若干字节空间作为备用空间。

格式: [标号:] DS 表达式

例 2、 ORG 1000H

 DS 0AH

 DB 71H, 11H, 11H ; 从 100BH 开始存放 71H、11H、11H。

注: DB、DW、DS 只能用于程序存储器; 而不能用于数据存储器。

8、位地址赋值伪指令 BIT

将位地址赋予规定的字符名称。

格式: 字符名称 BIT 位地址

例 3、 X1 BIT P12

相当于 X1 EQU 92H

小结:

- 1、为什么要对汇编程序进行汇编?
- 2、机器汇编的过程是怎样的?
- 3、常用伪指令的功能是什么?

布置作业：题目抄在黑板上。

版书设计

<p>一、汇编程序功能</p> <p>1、...</p>	<p>二、汇编程序的汇编过程</p> <p>1、.....</p> <p>2、.....</p>	<p>三、伪指令</p> <p>1、主电源（VCC 、VSS）</p> <p>2、外接晶振</p>
<p>3、.....</p> <p>4、.....</p> <p>5、.....</p> <p>....</p>	<p>小结:</p> <p>1、.....</p> <p>2、.....</p> <p>作业:</p>	

第十四讲 汇编程序的基本结构（一）

教学目的:

- 1、掌握汇编语言的三种结构
- 2、掌握顺序结构、分支结构程序的编写方法

教学重点:

顺序结构、分支结构

教学难点:

分支结构、分支条件的确定

§ 3-8 汇编语言程序的基本结构及应用举例

程序编写做到:

占用存储空间少;

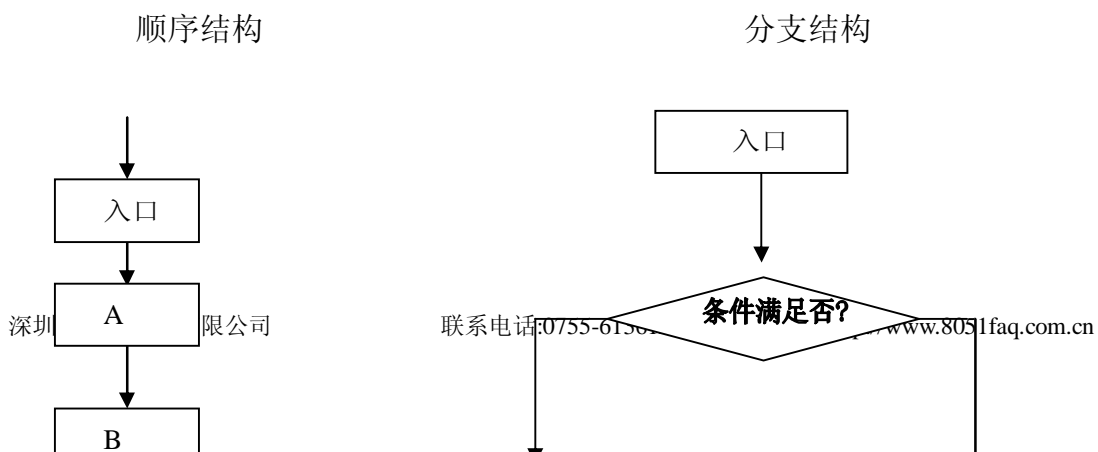
运行时间短;

程序的编制、调试及排错所需时间短;

结构清晰, 易读、易于移植。

按结构化程序设计思想, 任何复杂程序都可用顺序结构、分支结构、循环结构等构成。

如图 1 所示。



N

Y

图 (a)

图 (b)

循环结构

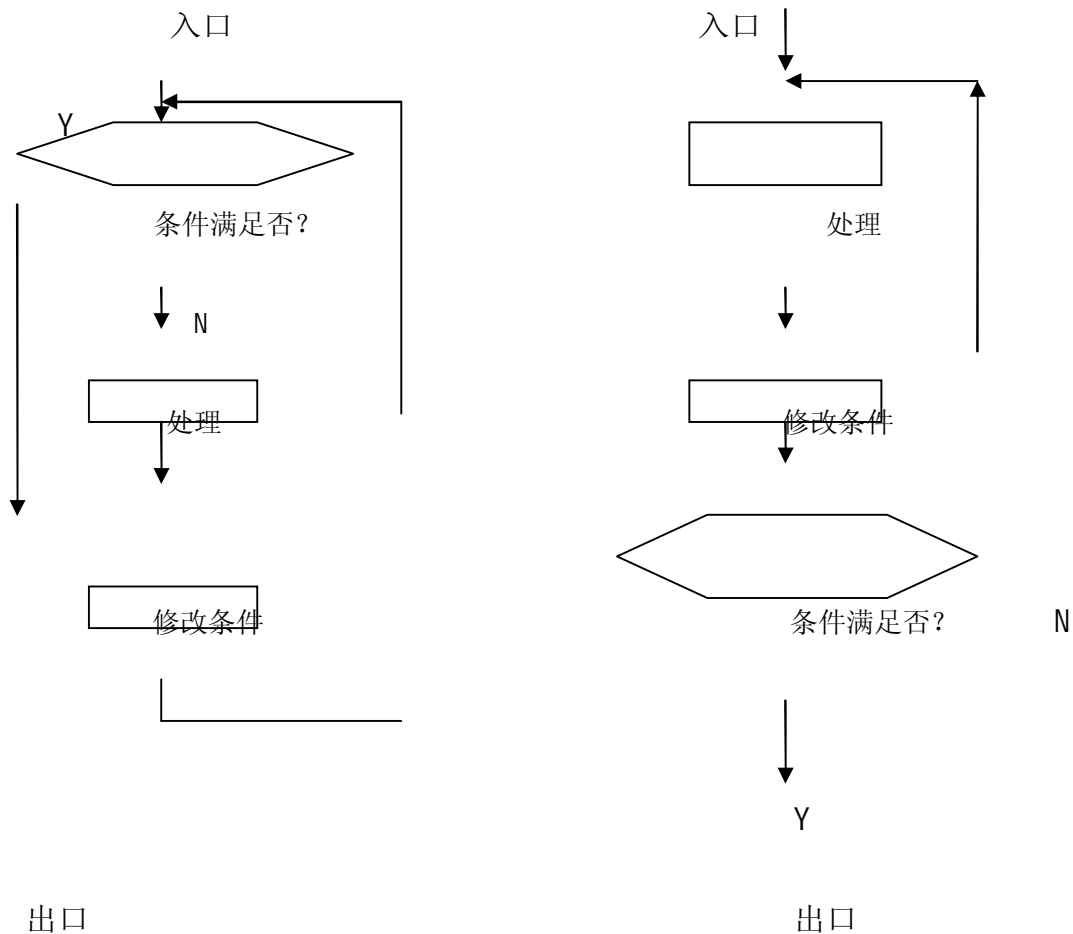


图 (c)

图 1 常用程序结构

一、顺序程序举例

例 1、设在外 RAM 的 60H 单元存有 1 个字节代码，要求将其分解成两个 4 位字段，高 4 位存入原单元的低 4 位，其低 4 位存入 61H 单元的低 4 位，且要求这两个单元的高 4 位均为 0，试编制完整程序。


```
INC R0
```

```
MOVX @R0, A
```

```
END
```

二、分支结构与分支程序设计

结构：根据不同的条件，进行相应的处理。

通常用条件转移指令形成简单分支结构。

如：判 (A) = Z 或 NZ ，转移

判 (CY) = 1 或 0 ，转移

判 (bit) =1 或 0 ，转移

CJNE 比较不相等转移

例 3、设 a 存放在累加器 A 中，b 存放在寄存器 B 中，要求按下式计算 Y 值，并将结果 Y 存于累加器 A 中，试编写程序。

$$Y = \begin{cases} a-b & (a \geq 0) \\ a+b & (a < 0) \end{cases}$$

解：本题关键是判 a 是正数，还是负数；由 ACC7 便知。

```
ORG 1000H
```

```
BR: JB ACC7, MINUS
```

```
CLR C  
  
SUBB A, B  
  
SJMP DONE  
  
MINUS: ADD A, B  
  
DONE: SJMP $  
  
END
```

例 4、设有两个 16 位无符号数 NA, NB 分别存放在 8031 单片机内部 RAM 的 40H、41H 及 50H、51H 单元中, 当 $NA > NB$ 时, 将内部 RAM 的 42H 单元清 0; 否则, 将该单元置成全 1, 试编程。

解法 I: 因为无 16 位数的比较指令, 所以, 只能用 8 位数的比较指令。

(画出流程框图)

```
ORG 2000H  
  
CMP: MOV A, 50H  
  
CJNE A, 40H, CMP1  
  
MOV A, 51H  
  
CJNE A, 41H, CMP1  
  
SJMP NHIGHE
```

```
CMP1: JC  HIGHE
```

```
NHIGHE: MOV  42H, #OFFH
```

```
SJMP  DONE
```

```
HIGHE: MOV  42H, #00H
```

```
DONE: SJMP  $
```

```
END
```

上述程序中多次用到 SJMP 语句，该语句为无条件转移语句。无条件语句应尽量少用，这样可使程序结构紧凑而易读，易理解。

解法 II：先假设 $NA > NB$ ，再来判断是否 $NA \leq NB$

```
ORG  3000H
```

```
CMP2: MOV  R0, #00H
```

```
MOV  A, 50H
```

```
CJNE A, 40H, CMP3
```

```
MOV  A, 51H
```

```
CJNE A, 41H, CMP3
```

```
SJMP  NHIGHE
```

```
CMP3: JC  HIGHE
```

NHIGHE: MOV R0, #OFFH ; 不大于标志

HIGHE: MOV 42H, R0

SJMP \$

END

小结:

- 1、汇编程序有哪三种基本结构?
- 2、构成分支程序,常用到哪些判断语句?

布置作业: 抄题目在黑板上。

版书设计

<p>一、顺序结构</p> <p>1、....</p> <p>2、.....</p> <p>3、</p>	<p>二、分支结构</p> <p>1、.....</p> <p>2、.....</p>	
<p>三、分支程序设计</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>四、举例</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>小结:</p> <p>1、.....</p> <p>2、.....</p> <p>.作业:</p>

第十五讲 汇编语言的基本结构 (二)

教学目的:

- 1、掌握循环程序结构
- 2、了解循环结束判断的两种方法
- 3、能编写单间的单循环、双循环的程序

教学重点:

单循环程序编制

教学难点:

循环程序的结构构思

- 复习:**
- 1、循环转移指令 DJNZ 的功能是怎样的?
 - 2、汇编程序的在三种基本结构是什么?

顺序程序 —— 每条指令只执行一次;

分支程序 —— 根据不同条件, 会跳过一些指令, 而转去执行另一些指令

共同点: 每条指令至多执行一次。但是, 实际中有时要求某程序段多次重复执行, 需要采用循环结构。

§ 3-8 汇编语言程序的基本结构及应用举例

循环结构不但使程序简练, 而且大大节省存储空间。

循环程序包含四部分:

初始化部分

循环处理部分 (主体)

循环控制部分 (修改地址指针、修改变量、检测循环结束条件)

循环结束部分 (对结果分析、处理, 存放结果)

循环有: 单循环、多重循环。

循环次数已知, 可用计数器控制循环次数;

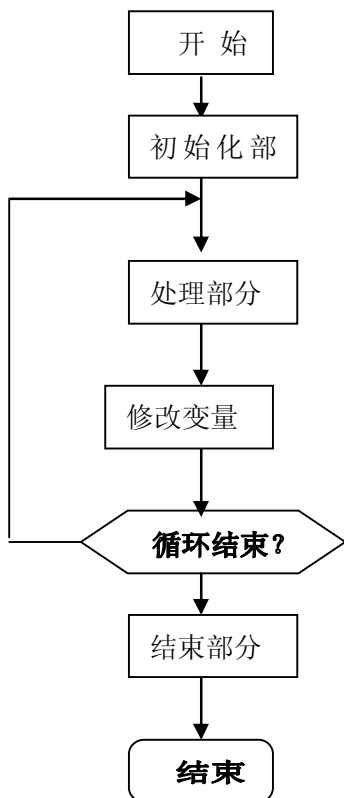
循环次数未知, 按问题条件控制循环是否结束。

一、单循环程序

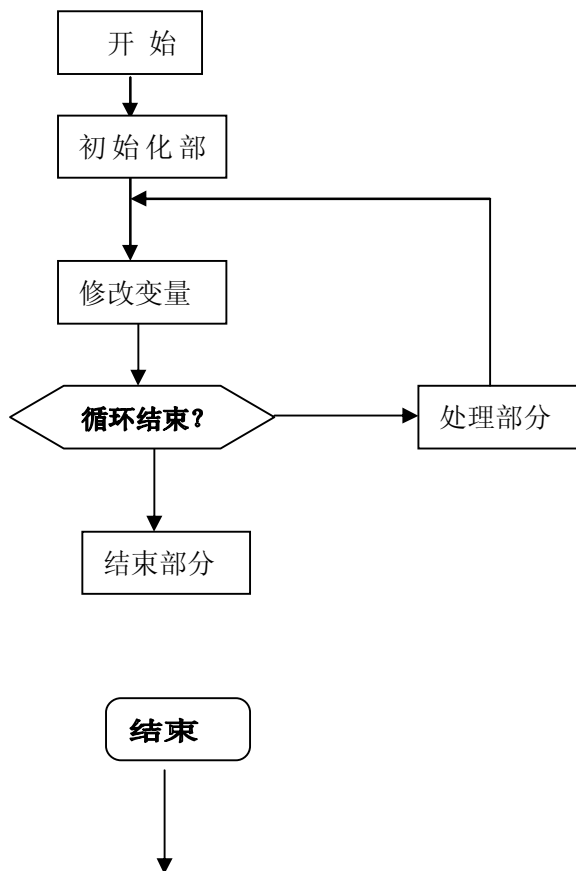
1、循环次数是已知的程序

例 1、已知片外 RAM 的 10H 单元存放 8 位二进制数，要求将其转移成相应的 ASCII 码，并以高位在前，低位在后的顺序，依次存放到片外 RAM 以 11H 为首地址的连续单元中，试编程。

先处理，后判断：



先判断，后处理：



解：先将中间单元置成 30H，然后判欲转换位是否为 1，

若是，则将中间单元内容加 1；否则，中间单元内容保持不变。

通过左移指令实现由高到低的顺序进行转换。

```
ORG 1000H

START: MOV R2, #08H ; 循环计数初值（循环次数已知）
      MOV R0, #10H ; 地址指针初值
      MOVX A, @R0 ; 取数
      MOV B, A ; 暂存 B 中
LOOP:  MOV A, #30H ; 将中间单元（A）置成 30H
      JNB B.7, NA ; 判断转换的二进制位为 0 否？
           ; 若是转 NA
      INC A ; 若为 1，则（A）内容加 1，成为
           ; 1 的 ASCII 码“31H”
NA:    INC R0 ; 修改地址指针
      MOVX @R0, A ; 存放转换的结果
      MOV A, B
      RL A, B ; 作好准备，判断下一位
      MOV B, A ; 暂存
      DJNZ R2, LOOP ; 判断转换结束否？未完继续
      SJMP $
      END
```

(2) 循环次数未知的程序

例 2、设用户用键盘输入长度不超过 100 字节的字符串放在 8031 单片机外部 RAM 以 20H 为首地址的连续单元，该字符串用回车符 CR（‘CR’ = 0DH）作为结束标志，要求统计此字符串的长度并存入内部 RAM 的 1FH 单元中。

解：从首单元开始取数，每取一数判断其是否为‘CR’，是则结束。

```

        ORG 1000H

STADA DATA 20H

SLANG DATA 1FH

CMCR2: MOV R0, #STADA-1

        MOV B, #OFFH

CRLOP: INC R0

        INC B

        MOVX A, @R0

        CJNE A, #0DH, CRLOP

        MOV SLANG, B

        SJMP $

        END
    
```

2、多重循环设计

循环体中还包含着一个或多个循环结构，即双重或多重循环。

例 3、设 8031 使用 12MHz 晶振，试设计延迟 100ms 的延时程序。

解：延时程序的延迟时间就是该程序的执行时间，通常采用 MOV 和 DJNZ 二指令。

$$T = 12 / f_{osc} = 12 / (12 \times 10^6) = 1\mu s$$

```

        ORG 1000H

DELAY: MOV R2, #CTS           ; T = 1us
        LOOPS: MOV B, #CTR     ; T = 1us
        LOOPR: DJNZ B, LOOPR   ; 2T = 2us
                DJNZ R2, LOOPS ; 2T = 2us

        END
    
```

} 内 外
 } 循 循
 } 环 环

内循环延时：

$$(1 + 2 \times \text{CTR}) T = 500\mu\text{s} \text{ (假设)}$$

则 $\text{CTR} = 250$

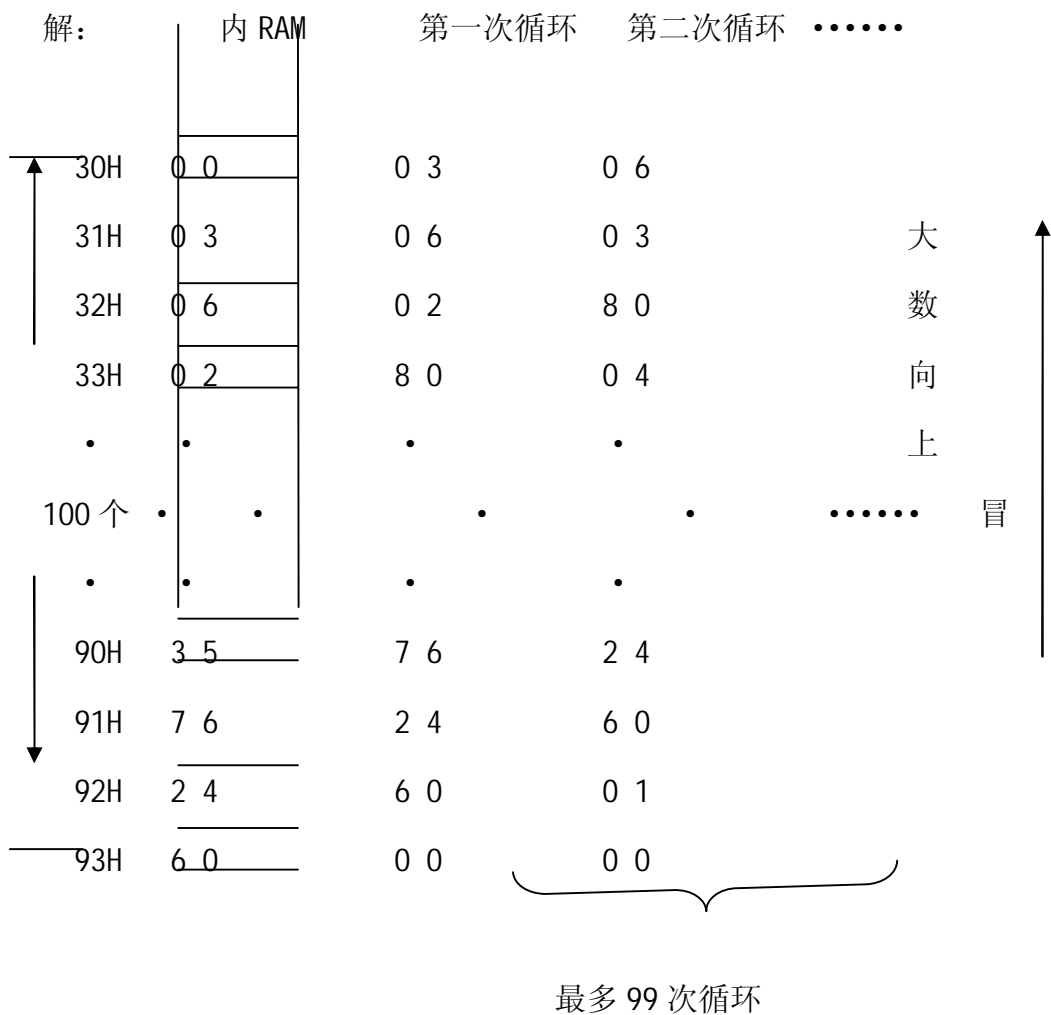
实际延时: $[1 + 2 \times 250] \times 1\mu\text{s} = 501\mu\text{s}$

外循环延时: $T + (501 + 2T) \times \text{CTS} = 100\text{ms} = 100\,000\mu\text{s}$

所以, $\text{CTS} = 198.8$ 取 199

实际延时: $[1 + (501 + 2) \times 199] = 1000.98\text{ms}$

例 4、设在 8031 内部 RAM 中存一无符号数的数组，其长度为 100，起始地址是 30H，要求将它们从大到小排序，排序后仍存放在原区域中，试编者按程。



这就是所畏的“冒泡法”。

实际上大多情况，用不到 99 次循环，排序就结束。为了提高排序速度，程序中可设一交换标志位，如 10H 位，

每次循环中：若有交换则 SETB 10H

若无交换则 CLR 10H

每次循环结束时，测 10H 位，判断排序是否结束。

```
ORG 1000H
```

```
BUBBLE: MOV R0, #30H
```

```
MOV B, #64H
```

```
CLR 10H
```

```
DEC B ; 长度计数
```

```
LOOP: MOV A, @R0
```

```
MOV 20H, A ; 暂存，为交换作准备
```

```
INC R0
```

```
MOV 21H, @R0
```

```
CJNE A, 21H, BUEU ; 若 (20H) ≠ (21H) 转移
```

```
BUEU: JNC BUNEXT ; (20H) ≥ (21H) 转移
```

```
MOV A, @R0 ; 若 (20H) < (21H) 则交换
```

```
MOV @R0, 20H
```

```

DEC R0 ; 使 R0 退格指向小地址

MOV @R0, A

INC R0 ; 恢复 R0 指向大地址

SETB 10H ; 置交换标志

BUNEXT: DJNZ B, LOOP

JB 10H, BUBBLE ; 判断标志位为 1 否? 若为 1, 则继续

END
    
```

小结:

- 1、 循环程序包括几部分内容?
- 2、 何谓单循环, 多重循环?
- 3、 何谓循环程序的先判断, 后处理; 先处理, 后判断?

布置作业: 题目抄在黑板上

版书设计

<p>一、循环结构流程图</p> <p>1、</p> <p>2、</p> <p>3、</p>	<p>二、单循环程序</p> <p>1、</p> <p>2、</p>	
<p>三、循环次数未知的程序</p> <p>1、</p> <p>2、</p> <p>3、</p>	<p>四、举例</p> <p>1、</p> <p>2、</p> <p>3、</p>	<p>小结:</p> <p>1、</p> <p>2、</p> <p>.作业:</p>

第十六讲 中断系统

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、掌握计算机中断概念
- 2、掌握单片机中断系统的组成
- 3、熟练运用中断控制

教学重点:

五个中断源, 两级管理

教学难点:

中断的响应过程

第四章 MCS-51 的片内接口

§ 4-1 中断系统

一、概述

什么叫中断?

举例: 同学正在教室写作业, 忽然被人叫出去, 回来后, 继续写作业。

引入计算机中断的概念。

中断的作用:

处理断电保存, 解决快速 CPU 与慢速外设之间的矛盾等。

对 MCS-51 单片机的中断系统用一句话讲叫: “五源中断, 两级管理”

二、中断请求源 (“五源中断”)

五个中断源:	入口地址
外部中断 0 (/INT0)	0003H
T0 溢出中断	000BH
外部中断 1 (/INT1)	0013H
T1 溢出中断	001BH
串口中断	0023H

有了中断请求, 如何通知 CPU?

通过中断请求标志位来通知 CPU。

(投影显示 MCS-51 单片机中断控制系统逻辑电路图)

外部中断源、定时/计数器的中断请求标志位分布在 TCON 中;

串口中断标志位分布在 **SCON** 中。

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
------	-----	-----	-----	-----	-----	-----	-----	-----

TF1—— T1 的溢出中断标志。

硬件置 1，硬件清 0（也可软件清 0）。

TF0—— T0 的溢出中断标志。（同 **TF1**，只是针对 T0 的）

IE1 —— 外部中断 1（/INT1）请求标志。

外部有中断请求时，硬件使 **IE1** 置 1，硬件清 0。

IE0 —— 外部中断 0（/INT0）请求标志。

IT1 —— 外部中断 1（/INT1）触发类型控制位。

IT1 = 0，低电平触发。

IT1 = 1，下降沿触发。

IT0 —— 外中断 0（/INT0）触发类型控制位，用法同 **IT1**。

SCON						TI	RI
------	--	--	--	--	--	----	----

TI —— 串口发送中断标志位。

发送完数据，硬件使 **TI** 置 1，软件清 0（CLR TI）

RI —— 串行口接收中断标志位。

硬件置 1，软件清 0。

三、 中断控制（两级管理）

1、中断屏蔽

在中断源与 CPU 之间有一级控制，类似开关，其中第一级为一个总开关，第二级为五个分开关，由 **IE** 控制。

IE	EA			ES	ET1	EX1	ET0	EX0
----	----	--	--	----	-----	-----	-----	-----

- EA —— 总控制位
- ES —— 串口控制位
- ET1 —— T1 中断控制位
- EX1 —— /INT1 控制位
- ET0 —— T0 中断控制位
- EX0 —— /INT0 控制位



若为“1”，允许（开关接通）
 若为“0”，不允许（开关断开）
 例如，SETB EA
 CLR IE.7

2、中断优先级

为什么要有中断优先级？

CPU 同一时间只能响应一个中断请求。

若同时来了两个或两个以上中断请求，就必须有先有后。为此将 5 个中断源分成高级、低级两个级别，高级优先，由 IP 控制。

IP

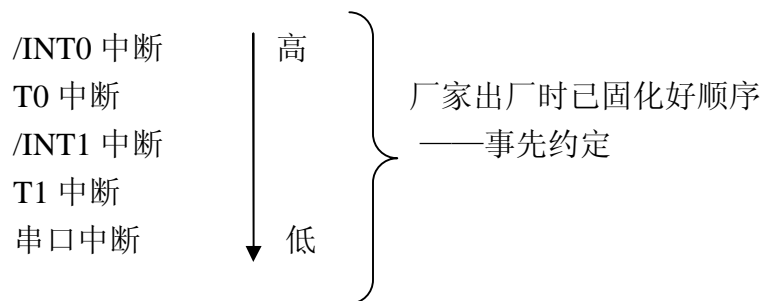
			PS	PT1	PX1	PT0	PX0
--	--	--	----	-----	-----	-----	-----

以上各位与 IE 的低五位相对应，为“1”时为高级。初始化编程时，由软件确定。

例如，SETB PT0 或 SETB IP1

CLR PX0 等。

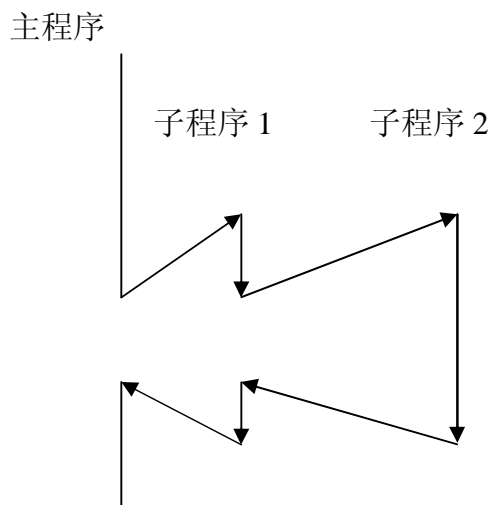
同一级中的 5 个中断源的优先顺序是：



中断优先原则：（概括为四句话）

- 1、低级不中断高级
- 2、高级不睬低级
- 3、同级不能打断
- 4、同级、同时中断，事先约定。

中断嵌套的概念：



四、MCS-51 中断的响应过程

CPU 每个机器周期都需要顺序检查每个中断源，当检测到有中断请求时，能否响应，还要看下述情况是否存在：

- (1) CPU 正处理相同级别或更高级别的中断；
- (2) 正在执行指令，还未到最后一个机器周期；
- (3) 正在执行的指令是 RETI 或访问 IP、IE 指令，则执行完上述指令后，再执行一条指令后，才会响应新中断。

响应过程：（假设已使某中断请求标志置 1）

- (1) 先使相应优先级状态触发器置 1；
- (2) 执行一个硬件子程序的调用，
 - 1) 硬件清零相应中断请求标志（TI、RI 除外）
 - 2) 将当前 PC 内容压入堆栈——保护断点；
 - 3) 将中断服务子程序入口地址送 PC——转移。

返回过程：（RETI 执行后）

- (1) 使相应优先级状态触发器清 0。
- (2) 从堆栈中弹出栈顶的两个字节内容送 PC——恢复断点。
- (3) CPU 接着中断处继续执行原程序。

注意：1) 保护断点与保护现场以及恢复断点与恢复现场的区别。

2) 外部中断响应时间在 3 ~ 8 个机器周期之间。

利用外部中断实现单步操作。

小结：

1、MCS-51 单片机中断系统有几个中断源？分别是什么？事先约定的优先顺序是怎样的？

2、如何进行中断允许控制？如何进行中断优先级控制？

3、中断优先的规则是什么？

版书设计

一、中断概述	二、中断请求源	四、 中断控制
1、	1、	1、
2、	2、	2、
3、	3、	3、
4、	4、	

四、MCS-51 中断响应过程 1、 2、 3、	小结: 1、 2、 3、 作业:
--	--	-----------------

第十七讲 定时/计数器

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 MCS-51 单片机内定时/计数器系统组成
- 2、掌握定时/计数器工作方式的控制
- 3、能编制简单的应用程序

教学重点:

定时/计数器的工作方式及模式

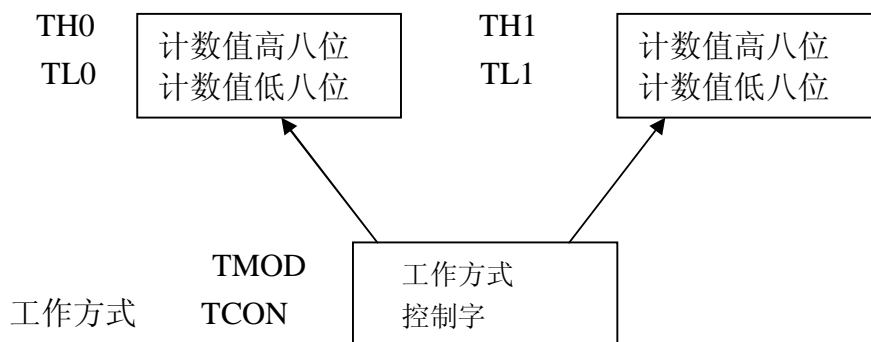
教学难点:

方式及模式确定

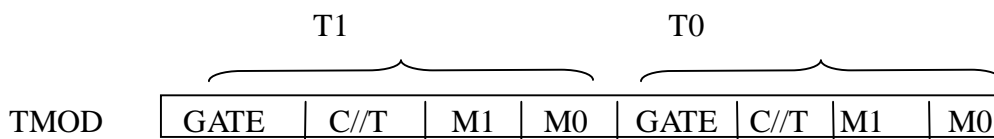
§ 4-2 定时 / 计数器

一、定时 / 计数器的结构

T0、T1 均为 16 位加 1 计数器。



1、工作方式控制寄存器 TMOD



GATE —— 门控位。

GATE = 0 启动不受 /INT0 或 /INT1 的控制；

GATE = 1 启动受 /INT0 或 /INT1 的控制。

C//T —— 外部计数器 / 定时器方式选择位

C//T = 0 定时方式；

C//T = 1 计数方式。

M1M2 —— 工作模式选择位。

M1	M0	模式	说明
0	0	0	13 位定时/计数器 高八位 TH (7 ~ 0) + 低五位 TL (4 ~ 0)
0	1	1	16 位定时/计数器 TH (7 ~ 0) + TL (7 ~ 0)
1	0	2	8 位计数初值自动重装 TL (7 ~ 0) ← TH (7 ~ 0)
1	1	3	T0 运行, 而 T1 停止工作, 8 位定时/计数。

2、定时 / 计数器控制寄存器 TCON

TCON		TR1		TR0				
------	--	-----	--	-----	--	--	--	--

TR0 —— 定时 / 计数器 0 运行控制位。

软件置位, 软件复位。

与 GATE 有关, 分两种情况:

GATE = 0 时, 若 TR0 = 1, 开启 T0 计数工作;
若 TR0 = 0, 停止 T0 计数。

GATE = 1 时, 若 TR0 = 1 且 /INT0 = 1 时, 开启 T0 计数;
若 TR0 = 1 但 /INT0 = 0, 则不能开启 T0 计数。
若 TR0 = 0, 停止 T0 计数。

TR1 —— 定时 / 计数器 1 运行控制位。

用法与 TR1 类似。

二、定时 / 计数器的四种工作模式

1、模式 0

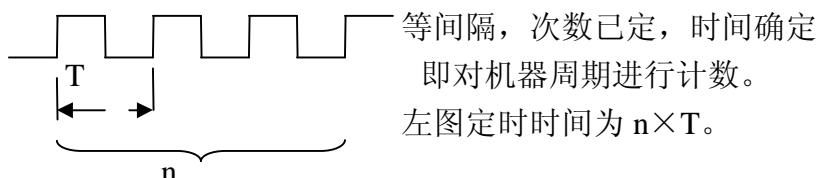
T1、T0 的等效逻辑结构 (显示投影胶片)

计数寄存器 TL_i 低 5 位 + TH_i 8 位

C//T = 0 —— 定时

C//T = 1 —— 对外计数。

定时: $f_{osc} / 12 = 1 / (12 / f_{osc}) = 1 / T$



计数: 脉冲不等间隔。



— — — — — 确认一次负跳变需两个机器周期，
所以，计数频率最高为 $f_{osc} / 24$ 。

GATE = 0 时，A = “1” $\left\{ \begin{array}{l} B = \text{“1” (TRi = 1 时), 启动计数} \\ B = \text{“0” (TRi = 0 时), 停止计数。} \end{array} \right.$

GATA = 1 时， $\left\{ \begin{array}{l} A = \text{“1” (/INTXi = “1”)} \\ A = \text{“0” (/INTXi = “0”)} \end{array} \right. \left\{ \begin{array}{l} B = \text{“1” (TRi = 1 时) 启动,} \\ B = \text{“0” (TRi = 0 时) 停止。} \end{array} \right.$

2、模式 1

与模式 0 相似。

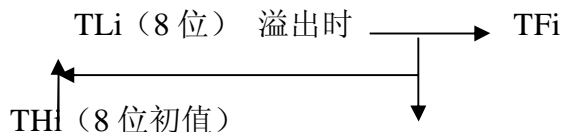
与模式 0 的区别：计数位数不同。

计数寄存器：THi（高 8 位）+ TLi（低 8 位）

3、模式 2

与模式 0、1 的区别：1) 计数位数不同；
2) 初值自动重装。

计数寄存器：



4、模式 3

T0 定时/计数，而 T1 停止计数，但可作波特率发生器。T0 分成两独立定时/计数器 TL0 和 TH0。

TL0 使用 C/T、GATE、TR0、/INT0、TF0 定时/计数，
TH0 使用 TR1、TF1 因此，只能用于定时。

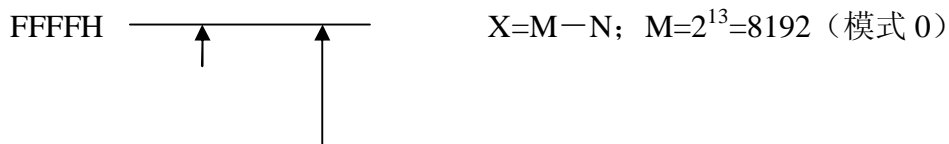
(展示投影胶片。)

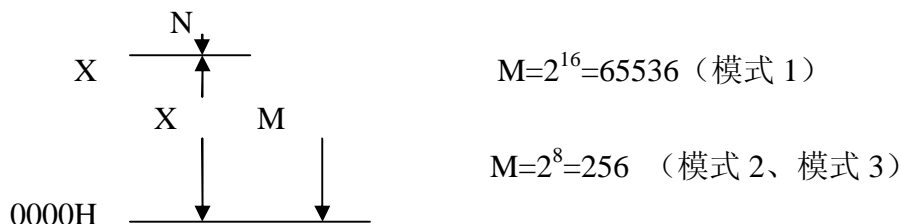
模式 3 时，T1 可定时为模式 0、1、2 的定时/计数，但不可中断，所以一般只作串口波特率发生器用。(投影展示图 4-8)

三、定时/计数器的应用

编程前确定参数：

- (1) 定时/计数器—— T0、T1 选择其一，
- (2) 工作方式——C/T 及 GATA，
- (3) 计数初值——加 1 计数、16 位。





(4) 工作模式——M1、M0

编制初始化程序:

1) 写 TMOD; 2) 确定 IE、IP; 3) 写计数初值; 4) 启动计数 (TRi)

例 1、设计一个能产生 $t=1ms$ 的周期信号发生器, 试编程。

解: 选 T0; C/T=0, GATE=0

$$N = t / T = t / (12 \times (1/fosc)) = 500$$

所以, $X = M - 500$, 问: M 取多少?

模式 0、模式 1 均可, 取模式 0, $M=2^{13}=8192$

$$X = 8192 - 500 = 7692 = 1E0CH = 0001\mathbf{1110\ 000}\ \mathbf{0\ 1100}B,$$

先将低五位放入 TL0 中, 再将剩余的数从右向左数出八位放入 TH0 中:

TH0 ← 0F0H

TL0 ← 0CH

程序:

```

    ORG 0000H
    LJMP MAIN
    ORG 000BH
    LJMP INSE1
    ORG 1000H
MAIN: MOV SP, #60H
      MOV TL0, #0CH
      MOV TH0, #0F0H
      MOV TMOD, #00H
      SETB TR0; IE
      SETB ET0; TCON
      SETB EA
      SJMP $

INSE1: MOV TL0, #0CH
       MOV TH0, #0F0H
       CPL P10
    
```


RETI

- 小结:** 1、MCS-51 单片机内有几个定时计数器? 如何计数?
 2、T0、T1 有几种工作方式?
 3、编程应用前要事先确定的参数有几个?

布置作业: P141 9

版书设计

<p>一、定时 / 计数器的结构</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p> <p>4、.....</p>	<p>二、定时 / 计数器的四种工作模式</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p> <p>4、.....</p>
<p>三、定时/计数器的应用</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>小结:</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p> <p>作业:</p>

第十八讲 计数/定时器的应用

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、GATE 的用法
- 2、掌握定时/计数参数的计算方法
- 3、掌握定时/计数器工作模式的确定

教学重点:

定时/计数器的初

教学难点:

GATE 的用法

复习: 1、定时 / 计数器的控制位有哪些? 分别在何处?

2、计数初值如何计算?

讲授新课:

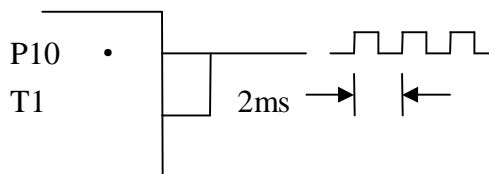
§ 4-2 定时 / 计数器应用

TMOD	GATE	C//T	M1	M0	GATE	C//T	M1	M0
------	------	------	----	----	------	------	----	----

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
------	-----	-----	-----	-----	-----	-----	-----	-----

例 2、设 $f_{osc} = 6\text{MHz}$ ，利用单片机内定时/计数器及 P10 口线输出 1000 个脉冲，脉冲周期为 2ms，试编程。

解:



$$T = 12 \times 1 / f_{osc} = 2\mu\text{s}$$

选取 T0 定时; T1 计数。

设 T0 采用中断方式产生周期为 2ms 方波，T1 对该方波计数，当输出至第 1000 个脉冲时，使 TF1 置 1。

在主程序中用查询方法，检测到 TF1 变 1 时，关掉 T0，停止输出方波。

T0、T1 参数的确定:

T0 模式 0、定时: 脉宽为脉冲周期的一半

$$\text{所以, } X = 2^{13} - 1\text{ms} / 2\mu\text{s} = 0001\ 1110\ 0000\ 1100\text{B}$$

$$\text{TH0} = 0\text{F0H}$$

$$\text{TL0} = 0\text{CH}$$

T1 模式 1、计数: $N = 1000$

则 $X = 65536 - 1000 = 64536 = 0FC18H$

(若选模式 0 也可以, 此时 $X = 7192 = 1C18H$)

程序:

```

ORG 0000H
LJMP MAIN
ORG 000BH
LJMP TOS
ORG 1000H
MAIN: MOV TMOD, #50H; T0 定时, 模式 0; T1 计数, 模式 1
      MOV TL0, #0CH
      MOV TH0, #0F0H
      MOV TL1, #18H
      MOV TH1, #0FCH
      SETB TR1
      SETB TR0
      SETB ET0
      SETB EA
WAIT: JNB TF1, WAIT; 查询 1000 个脉冲计够没有? 没有等待。
      CLR EA
      CLR ET0
      ANL TCON, #0FH ; 停 T0、T1
      SJMP $
TOS:  MOV TL0, #0CH
      MOV TH0, #0F0H
      CPL P10
      RETI
      END

```

例 3、脉冲参数测量——GATE 功能的使用。

电路连接如下图所示。

脉冲高电平(计数)长度值存于 21H、20H 中,

脉冲低电平长度存于 23H、22H 中。

解: 复习 GATE 的用法:

GATE = 0 时, $TR_i = 1$, 即可启动 T_i 定时 / 计数

GATE = 1 时, $TR_i = 1$, 且 $\overline{INT}_i = 1$, 才启动定时 / 计数。

```
ORG 0000H
LJMP MAIN
ORG 2000H
MAIN: MOV TMOD, #99H ; T0、T1 均工作在定时,
; 模式 1, GATE=1
MOV A, #00H ; T0、T1 赋计数初值 00H, 定时最长时间
; 为 0000 ~ 65536
MOV TL0, A
MOV TH0, A
MOV TL1, A
MOV TH1, A
TEST0: JB P32, TEST0 ; 检测是否到 a 点
SETB TR0 ; 到 a 点, TR0 = 1, 做好取计时值准备。
TEST1: JNB P32, TEST1 ; 检测是否到 1 点
SETB TR1 ; 到 1 点 T0 计时; TR1 = 1, 做好 T1 计时准备。
TEST2: JB P32, TEST2 ; 检测是否到 2 点
CLR TR0 ; 到 2 点, 停止 T0 计时, T1 开始计时。
MOV 20H, TH0 ; 保存 T0 计时结果
MOV 21H, TL0
TEST3: JB P33, TEST3 ; 检测是否到 3 点
CLR TR1 ; 到 3 点, 停止 T1 计数
MOV 22H, TH1 ; 保存 T1 计数结果
MOV 23H, TL1
LCALL DISP
SJMP $
```

- 小结: 1、定时/计数的四种方式是怎样的?
2、如何计算计数初值? 如何编程送入计数初值?
3、GATE 的用法是怎样的?

布置作业: P141 10、11

版书设计

<p>一、例 1</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>二、例 2</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p> <p>.</p>	<p>三、例 3</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p> <p>小结.</p> <p>作业:</p>
---	--	---

第十九讲 单片机串行接口及其应用

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解单片机的基础知识。
- 2、了解单片机的发展概况。
- 3、熟悉数制的转换。

教学重点:

单片机的含义及作用

教学难点:

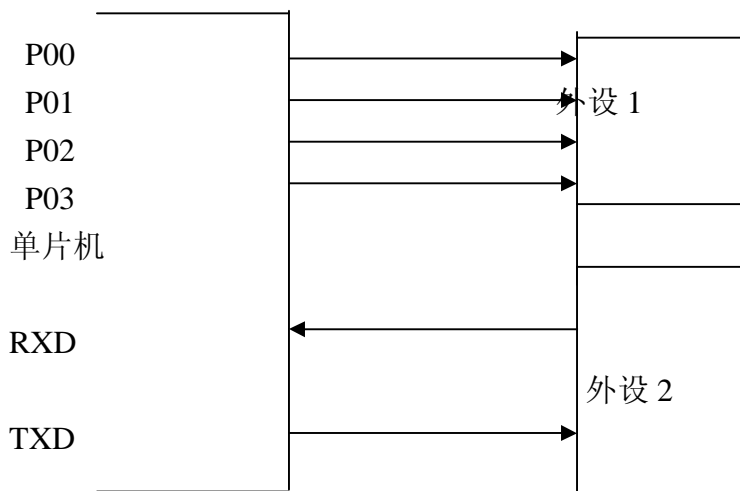
掌握逻辑分析仪的使用方法

§ 4-3 MCS-51 单片机串行接口

一、串行通信概述

1、什么叫串行通信？

并行、串行举生活中的例子（排横队行走，排纵队行走）说明；引出并行通信，串行通信的概念。



2、同步通信、异步通信

提问：数字电路中移位寄存器是怎样进行移位的？

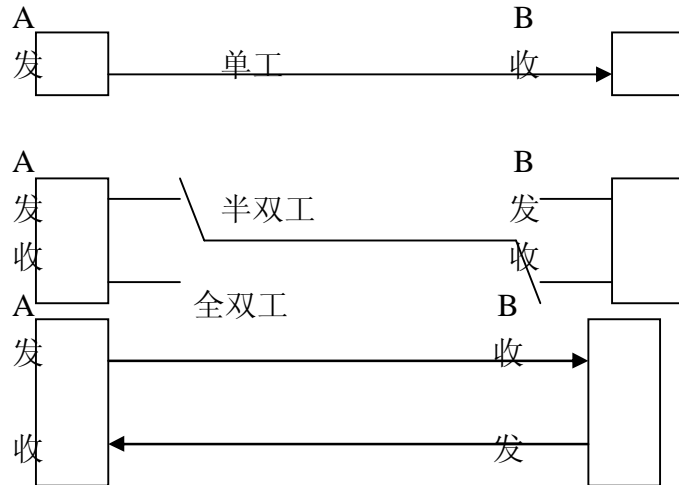
同步——发送设备时钟等于接收设备时钟。

同步字符 1	同步字符 2	数据 1	数据 2	...	数据 n	校验字符	校验字符
--------	--------	------	------	-----	------	------	------

异步——发送时钟不一定等于接收时钟。

空闲位	起始位	5 ~ 8 位数据	奇偶校验位	停止位	空闲位
-----	-----	-----------	-------	-----	-----

3、串行通信方向



4、波特率

即串行通信速率。 b/s 、 bps

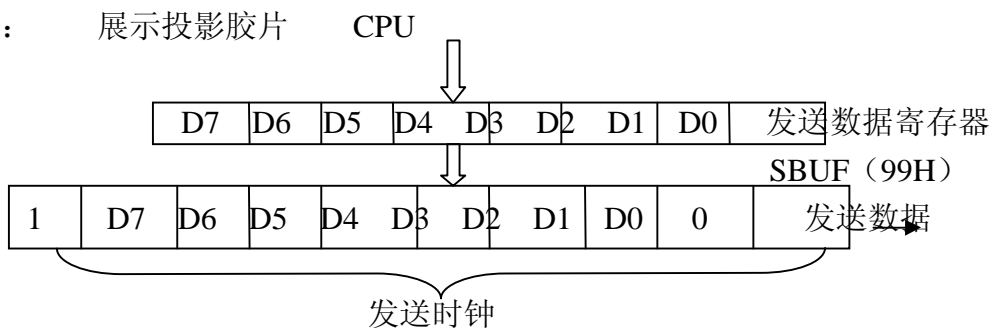
举例、设有一帧信息，1 个起始位、8 个数据位、1 个停止位，传输速率为 240 个字符。求波特率。

解：

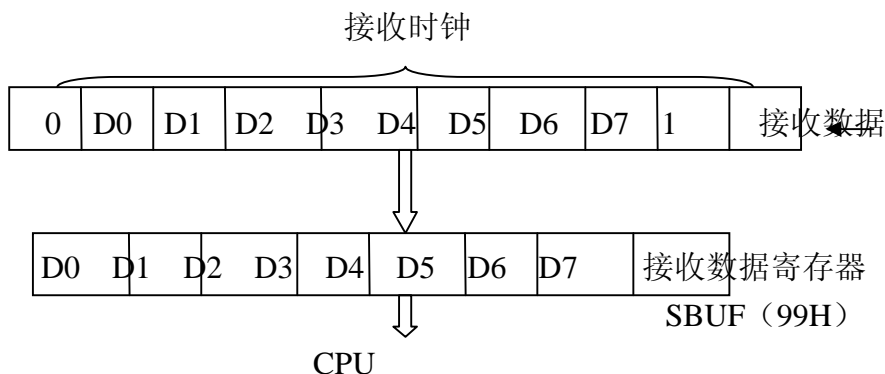
$$(1+8+1) \times 240 = 2400 \text{ b/s} = 2400 \text{ 波特。}$$

5、串行通信接口

发送： 展示投影胶片



接收： 展示投影胶片



二、MCS-51 机串行接口

单片机内有：通用异步接收/发送器 UART

全双工，4 种工作方式，波特率可编程设置，可中断。

1、串口的组成

从编程角度讲来看主要由以下寄存器组成。

PCON	SMOD							
------	------	--	--	--	--	--	--	--

SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
------	-----	-----	-----	-----	-----	-----	----	----

SBUF	D7	D6	D5	D4	D3	D2	D1	D0	} SBUF 发 SBUF 收
------	----	----	----	----	----	----	----	----	--------------------

SM0、SM1：工作方式选择位（后面再讲）

SM2：多机通信控制位，常与 RB8 配合，决定是否激活 RI

REN：允许接收

TB8：发送的第九位数

RB8：接收的第九位数

TI }：中断标志

RI }

SMOD：波特系数选择位，0 —— $2^0 = 1$

1 —— $2^1 = 2$

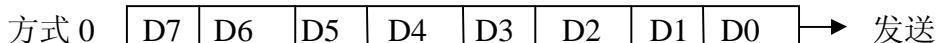
2、串行口的工作方式

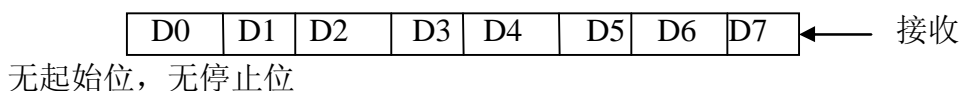
(1) 方式 0：同步移位寄存器方式，波特率固定为 $f_{osc} / 12$

RXD —— 接收发送数据

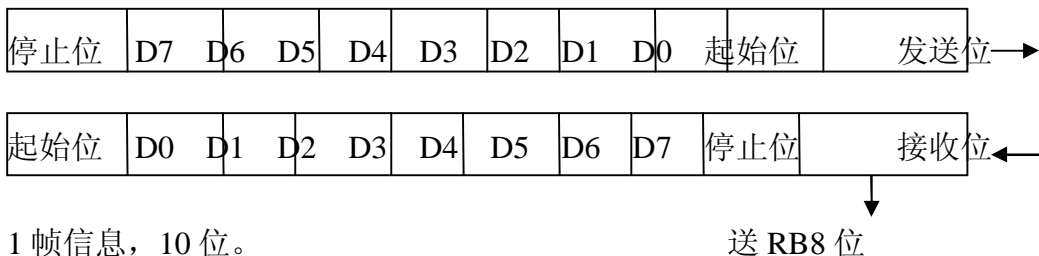
TXD —— 产生同步移位脉冲

接收/发送完，置位 RI / TI ，（要求 SM2 = 0）





(2) 方式 1: 8 位 UART, 波特率为 $(2^{SMOD} \times T1 \text{ 的溢出率}) / 32$



1 帧信息, 10 位。

波特率可变: $2^{SMOD} / 32 \times (T1 \text{ 的溢出率})$

发送完置位 TI。

接收完数据置位 RI。

置位 RI 是有条件的。即:

$REN = 1, RI = 0$ 且 $SM2 = 0$ 或 $SM2 = 1$ 但是接收到的停止位为 1。

此时, 数据装载 SBUF, 停止位进入 RB8, RI 置 1。

(3) 方式 2、方式 3 : 9 位 UART, 多机通信。



方式 2 波特率: (固定) $2^{SMOD} / 64 \times fosc$

方式 3 波特率: $2^{SMOD} / 32 \times (T1 \text{ 溢出率})$

发送完数据置位 TI。

接收到有效数据完毕, 置位 RI。

有效数据条件: $REN = 1, RI = 0$ 且 $SM2 = 0$ 或接收到第 9 位数据为 1。

此时, 数据装载 SBUF, 第 9 位数据 (TB8) → RB8, RI 置 1。

3、波特率的设置

方式 0、方式 2 固定。

方式 1, 方式 3 可变。波特率 = $2^{SMOD} / 32 \times (T1 \text{ 的溢出率})$

T1 溢出率 = 单位时间内溢出次数 = $1 / (T1 \text{ 的定时时间})$

而 T1 的定时时间 t 就是 T1 溢出一次所用的时间。此情况下，一般设 T1 工作在模式 2 (8 位自动重装初值)。

$$N = 2^8 - t/T, \quad t = (2^8 - N) T = (2^8 - N) \times 12 / f_{osc}$$

所以, T1 溢出率 = $1/t = f_{osc} / 12 (2^8 - N)$,

故, 波特率 = $2SMOD / 32 \times f_{osc} / 12 (256 - N)$ 。

若已知波特率, 则可求出 T1 的计数初值:

$$y = 256 - 2^{SMOD} \times f_{osc} / (\text{波特率} \times 32 \times 12)$$

巩固新课:

例、若 $f_{osc} = 6\text{MHz}$, 波特率为 2400 波特, 设 $SMOD = 1$, 则定时/计数器 T1 的计数初值为多少? 并进行初始化编程。

解: $y = 256 - 2^{SMOD} \times f_{osc} / (2400 \times 32 \times 12) = 242.98 \approx 243 = F3H$

同理, $f_{osc} = 11.0592\text{MHz}$, 波特率为 2400, 设 $SMOD = 0$, 则

$$y = F4H$$

初始化编程:

```
MOV  TMOD, #20H
MOV  PCON, #80H
MOV  TH1, #0F3H
MOV  TL1, #0F3H
SETB TR1
MOV  SCON, #50H
```

小结:

- 1、什么叫串行通信?
- 2、UART 叫什么? (通用异步接收/发送器)
- 3、单工、半双工、全双工?
- 4、串口工作用到哪些寄存器? (PCON、SCON、SBUF)
- 5、串口有几种工作方式?
- 6、波特率如何设置? (T1 的计数初值如何确定)

布置作业: P141 7、9、10

版书设计

<p>一、 串行通信概述</p> <p>1、....</p> <p>2、.....</p> <p>3、</p> <p>4、...</p>	<p>二、MCS-51 机串行接口</p> <p>1、.....</p> <p>2、.....</p>
--	---

<p>巩固新课 1、..... 2、..... 3、.....</p>	<p>小结: 1、..... 2、.....</p> <p>作业:</p>
--	---

第二十讲 MCS-51 存储器扩展技术

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解单片机最小系统
- 2、了解存储器的分类及应用
- 3、掌握存储器扩展方法

教学重点:

存储器扩展方法

教学难点:

地址分配

单片机内资源少,容量小,在进行较复杂过程的控制时,它自身的功能远远不能满足需要。为此,应扩展其功能。

MCS-51 单片机的扩展性能较强,根据需要,可扩展:

ROM、RAM; 定时 / 计数器; 并行 I/O 口、串行口; 中断系统扩展等。

本章主讲 ROM、RAM 的扩展; 并口、串口的扩展; 以及常用通道接口技术; 并介绍单片机与 A/D 转换、D/A 转换芯片的接口技术。

一、5-1 MCS-51 单片机最小系统

1、8051/8751 硬件最小系统

对于片内有 ROM 型单片机,其自身可以构成最小系统

图

该系统的资源如下:

4KB ROM, 256B RAM;

五源中断系统;

两个十六位加一定时 / 计数器;

一个全双工串行 UART;

四个并行 I/O 口。

2、8031 硬件最小系统

8031 单片机片内无 ROM，若要正常工作，必需外配 ROM。外接 ROM 后，P3 口、P2 口、P0 口均被占用只剩下 P1 口作 I/O 口用，其它功能不变。

图

图中：/E ——三态门控制端

G ——低电平锁存

二、存储器的扩展

1、三总线的连接

1、数据线的连接

P0 口的八位线承担此任，此时不用外接上拉电阻。

2、地址线的连接

P0 口承担地址低八位线，A0 ~ A7；

P2 口承担地址高八位线。A8 ~ A15。

注意：P0 口线地址 / 数据分时复用，需用地址锁存器 74LS373 锁存地址。

3、控制线的连接

对存储器来讲控制线无非是：芯片的选通控制、读写控制。

单片机与外部器件数据交换要遵循两个重要原则：

一是，地址唯一性，一个单元一个地址。

二是，同一时刻，CPU 只能访问一个地址，即只能与一个单元交换数据。

不交换时，外部器件处于锁闭状态，对总线呈浮空状态。

选通：CPU 与器件交换数据或信息，需先发出选通信号/CE 或/CS，以便选中芯片。

读 / 写：CPU 向外部设备发出的读/写控制命令。

EPROM: /OE ← /PSEN

SRAM: /WE ← /WR

/OE ← /RD

2) 存储器地址编码

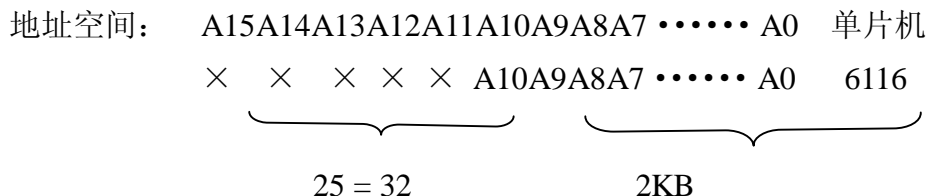
SRAM6116: “16” —— $2K \times 8b = 2KB \longrightarrow 2^1 \times 2^{10} = 2^{11}$

即 6116 有 11 根地址线。

地址空间: A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

最低地址: 0 0 0 0 0 0 0 0 0 0 0 0000H 6116 本身 }
最高地址: 1 1 1 1 1 1 1 1 1 1 1 1111H 的地址空间 }

MCS-51 单片机寻址范围: 64KB $\longrightarrow 2^6 \times 2^{10} = 2^{16}$ 即 16 位地址线



上式中: “×” 表示 0 或 1。

即单片机地址空间中包含有 32 个 2KB。某片 6116 占据的是哪 2KB 不能确定——地址浮动。

只有限定 A15 A11 的取值才能确定 6116 在系统中的地址范围。

如, P2.6 = 0, 选中 6116 的 /CS 线。设 P2.7 P2.5 P2.4 P2.3 假定全为 1 则: 6116 地址范围是 B800H ~ BFFFH。

同理, P27 P25 P24 P23 假定全为 0

则: 6116 地址范围是 0000H ~ 07FFH;

再设 P27 选中 6116 (设 P26 P25 P24 P23 全为 1)

则: 地址范围是 7800H ~ 7FFFH

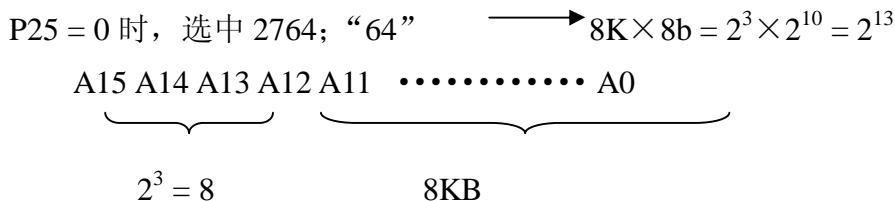
可见: 存储器芯片在系统中地址分布由两个因素决定:

- 一是, 芯片本身的地址线 (与容量有关)
- 二是, 芯片选通信号的获得方式。

扩展存储器时, 总是让单片机低位地址与存储器芯片地址线相接; 而让单片机剩余的高位地址线 (在 P2 口) 常作为片选信号线。

这种方法对 SRAM、EPROM、扩展 I/O 芯片、外设同样适用。

若上图中 2764 的接线中:



若取 P27 P26 均为 1;

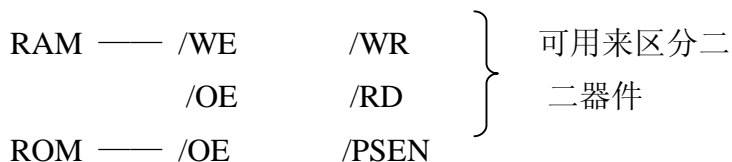
则 2764 在本系统内地址范围: C000H ~ DFFFH

若 6116 与 2764 都用 A13 (P25) 选中可否?

可以。此时 6116: D800H ~ DFFFH

2764: C000H ~ DB00H

可见，6116 与 2764 在 0800H ~ DFFFH 范围内地址重叠，这是不是违反交换原则呢？不！因为，6116 是 SRAM，2764 是 EPROM。除地址和选通信号外，还有读/写控制信号起作用。



- 小结：1) CPU 与外设数据交换的原则是什么？
 2) 决定存储器芯片在系统中地址范围的两个因素是什么？

布置作业： P220 5.2

版书设计：

<p>一、8051/8751 硬件最小系统</p> <p>1、</p> <p>2、</p> <p>3、</p> <p>4、 ...</p>	<p>二、存储器扩展</p> <p>1、</p> <p>2、</p>
<p>巩固新课</p> <p>1、</p> <p>2、</p> <p>3、</p> <p>....</p>	<p>小结：</p> <p>1、</p> <p>2、</p> <p>作业：</p>

第二十一讲 可编程并行 I/O 接口芯片 8255A

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 I/O 接口的特点及应用
- 2、掌握 8255A 的结构
- 3、掌握 8255A 的应用

教学重点:

8255A 的结构及应用

教学难点:

8255A 的结构及应用

复习: 并行 I/O 口扩展常用哪三种方法?

§ 5-4 可编程并行 I/O 接口芯片 8255A

一、8255A 的结构和操作系统

1、8255A 的组成及引脚

引脚图: P163 图 5-17

组成: P162 图 1-16

分三部分:

(1) 总线接口部分

/CS —— 片选线

A1、A0 —— 端口选择线 (选片内四个端口寄存器) } 输入

/RD —— 读信号线

/WR —— 写信号线

(2) 内部逻辑部分

PA

PC7 ~ PC4

PB

PC3 ~ PC0

} A 组控制电路

} B 组控制电路

} 控制 8255A 工作方式

} 二者合一成为端口控制寄存器。

(3) 外设接口部分

可由编程决定三个端口的功能

	输入	输出	其它
A 口	8 位锁存 / 缓冲	8 位锁存	双向
B 口	8 位锁存 / 缓冲	8 位缓冲	
C 口	8 位锁存 / 缓冲	8 位缓冲	可分成两组分别作 A 口、B 口的选通联络线

2、8255A 的端口操作

见 P163 表 5-9

A1	A0	选中
0	0	PA 口
0	1	PB 口
1	0	PC 口
1	1	控制寄存器

二、8255A 的工作方式及方式选择

1、8255A 的工作方式

(1) 方式 0 —— 基本输入/输出方式

A 口、B 口、C 口均有此方式，无选通，是单片机与外部设备之间的直接数据通道。

(2) 方式 1 —— 选通输入/输出方式

仅 PA 口、PB 口有此方式，

PC 口中若干位作联络信号线，联络信号的组合情况见 P164 图 5-18。

各联络信号线的意义：

/STB —— 输入选通信号，外设发来。

IBF —— 输入缓冲器满信号，发给外设（通知外设数据未被取走，暂不能接收新数据）

INTR —— 中断请求信号，外部设备发给单片机

INTE —— 中断允许信号

/OBF —— 输出缓冲器满信号，发给外设（单片机将数据已送到指定口，外部设备可以取走）

/ACK —— 外设响应信号，由外部设备发来（数据已送到外部设备）

(3) 方式 3 —— 双向方式

仅 PA 口有此方式。PC3 ~ PC7 作联络线

此时, PB 口可以是方式 0; 也可以是方式 1 (PC0 ~ PC1 作联络线)。

2、8255A 的方式控制字

用编程方法向 8255A 的控制口写控制字, 可决定它的工作方式。

有两个控制字:

(1) 方式选择控制字

1	D6	D5	D4	D3	D2	D1	D0
---	----	----	----	----	----	----	----

“1” —— 方式控制标志位

D6、D5 —— 决定 A 组的工作方式, 0 0 —— 方式 0

0 1 —— 方式 1

1 × —— 方式 2

D4 —— A 口的传输方向, 1 —— 入, 0 —— 出。

D3 —— PC7 ~ PC4 的传输方向, 1 —— 入, 0 —— 出。

D2 —— 决定 B 组的工作方式, 0 —— 方式 0, 1 —— 方式 1。

D1 —— B 口的传输方向, 1 —— 入, 0 —— 出。

D0 —— PC3 ~ PC0 传输方向, 1 —— 入, 0 —— 出。

(2) PC 口置位/复位控制字

0	D6	D5	D4	D3	D2	D1	D0
---	----	----	----	----	----	----	----

“0” —— 标志位。

D6、D5 —— 不使用位。

D3、D2、D1 —— 位选择位, 000 ~ 111 分别对应 PC7 ~ PC0。

D0 —— 位状态位, 1 —— 置位, 0 —— 复位。

三、8031 和 8255A 的连接及应用

单片机	8255A
A0	A0
A1	A1
A15 ~ A2 中任一线	/CS
P0	D7 ~ D0
/RD	/RD
/WR	/WR

RST

RESET

应用:

电路图

上图中, 地址线连接采用线选法。

8255A 的地址分布: A15 A14 A13 ... A8 A7 ... A1 A0

	0	0	0	...	0	0	...	0	0	0000H
		.			.			0	1	0001H
		.			.			1	0	0002H
		.			.			1	1	0003H
或	0	1	1	...	1	1	...	0	0	7FFCH—PA
		.			.			0	1	7FFDH—PB
		.			.			1	0	7FFE—PC
		.			.			1	1	7FFF—控制

例、设单片机 8031 与微型打印机之间的数据传送采用查询方式。要求将存放在 8031 单片机内 RAM 中以 30H 为首地址的 64 个连续单元中的内容打印输出, 试编程。

解: 因为 PC0 连接 BUSY, 所以, PC3 ~ PC0 为输入

又因 PC7 连接 /DATA STROBE, 所以 PC7 ~ PC4 为输出,

STROBE —— 表示重复的意思。

/DATA STROBE —— 数据选通信号。作用是通知打印机, 8255A 要给它传数。

PA 口输出, PB 口未用。

故 8255A 的控制字可设为: 1 0 0 0 0 0 0 1B = 81H

PA 口地址: 7FFCH

PB 口地址: 7FFDH

PC 口地址: 7FFE—PC

控制口: 7FFF—控制

ORG 1000H

```

PRINT: MOV DPTR, #7FFFH ; 控制口地址
      MOV A, #81H ; 控制字
      MOVX @DPTR, A ; 写入控制字
      MOV R1, #30H ; 数据指针
    
```

```

MOV R2, #40H ; 64 个数
NEXT: MOV DPTR, #7FFEh ; PC 口地址
MOV A, #80H ; 使 PC7 为高电平
MOVX @DPTR, A ; 输出/DATA STR OBE 为高电平使
; 无效, 不准备送数
WAIT: MOVX A, @DPTR ; 查询打印机状态
JB ACC.0, WAIT ; 若 PC0 即 BUSY = 1 忙, 则等待
MOV DPTR, #7FFCh ; 若 BUSY = 0 空闲, 则指向 PA 口
MOV A, @R1 ; 输出数据
MOVX @DPTR, A
MOV DPTR, #7FFEh ; 指向 PC 口
MOV A, #00H ; 8255A 输出/DATA STR OBE 信号
; 通知打印机, 给它传数。

MOVX @DPTR, A
ACALL PDELAY ; 调延时子程序, 以形成一个宽度
; 定时值的负脉冲为

INC R1
DJNZ R2, NEXT ; 判断打印输出完成否?
SJMP $
.
.
.
PDELAY: (延时程序略)
END

```

小结: 1、8255A 有几个并行 I/O 口? 几种工作方式?

2、工作方式控制字、置位/复位控制字是如何定义的?

布置作业: 抄题目在黑板上。

版书设计:

8255A 的结构和操作系统 1、 2、 3、 ...	二、 8255A 的工作方式及方式选择 1、 2、
三、 8031 和 8255A 的连接及应用 1、 2、 3、	小结: 1、 2、 作业:

第二十二讲 可编程并行 RAM / IO 扩展接口芯片 8155

教学方法: 讲授法

授课时数: 2 学时

教学目的:

1、进一步熟悉 I/O 接口的特点及应用

2、掌握 8155 的结构

3、掌握 8155 的应用

教学重点:

8155 的结构及应用

教学难点:

8155 的结构及应用

复习: 可编程并行接口芯片 8255A 有几个 I/O 口?

有几种工作方式? 由谁来确定?

§ 5-5 可编程 RAM / IO 扩展接口芯片 8155

包括: 256B 静态 RAM;

2 个 8 位和 1 个 6 位可编程并行 I/O 口;

1 个 14 位定时器 (减 1 计数);

1 个 8 位地址锁存器;

一些控制逻辑电路等。

一、引脚

AD0 ~ AD7 —— 地址/数据线。

传送地址、数据、命令、状态等。

ALE —— 地址锁存 (输入)

IO / M —— IO 口/RAM 选择

0: 选内 RAM

1: 选内 IO 口

/CE —— 片选线

/RD、/WR —— 读、写控制

PA7 ~ PA0 —— A 口输入/输出线

PB7 ~ PB0 —— B 口输入/输出线

PC5 ~ PC0 —— C 口输入/输出线

TIMERIN —— 定时器输入（输入定时器所需时钟）

TIMEROUT —— 定时器输出（输出所产生的方波脉冲）

二、8155 的 RAM 和 I/O 口地址编码

包括：内 RAM —— 256B 对应 256 个地址

命令/状态寄存器地址 —— 1 个地址

PA 口地址 —— 1 个地址

PB 口地址 —— 1 个地址

PC 口地址 —— 1 个地址

定时器低 8 位地址 —— 1 个地址

定时器高 8 位地址 —— 1 个地址

} 由 AD7 ~ AD0 及
IO / M 决定。

三、8155 的工作方式与基本操作

有三种基本操作：

1、作单片机片外 256B 数据存储器

IO / M = 0，与其它数据存储器统一编址。用 MOVX 访问。

2、作扩展 I/O 口使用

IO / M = 1，PA 口、PB 口、PC 口，可通过编程决定如何使用。

命令寄存器（命令控制字） —— I/O 口工作方式

状态标志寄存器 —— PA 口、PB 口状态标志。

I/O 口工作方式有四种：

A 口、B 口 基本 I/O 口， C 口输入；

A 口、B 口 基本 I/O 口， C 口输出；

A 口选通 I/O、B 口基本 I/O、C 口作联络线；

A 口、B 口选通 I/O、C 口作联络线。

状态标志寄存器：

BF —— 缓冲器满标志；

INTR —— 端口中断请求标志；

INTE —— 端口中断允许标志；

TIMER —— 定时器中断请求。

命令寄存器，只写不读

状态标志寄存器，只读不写

} 二者使用同一地址

3、作定时器扩展使用

可以通过编程决定输出 4 种信号，即有四种工作台方式（P172 图 5-27）

- 单方波； 连续方波；
- 单脉冲； 连续脉冲。

由两个 8 位寄存器，决定 14 位定时器计数常数及四种工作方式。

由命令寄存器的最高两位对定时器进行四种控制。

四、单片机与 8155 的接口及操作软件

图 5-28 (P173)

如图连线后地址分布是：内 RAM 地址 7E00H ~ 7EFFH

- I/O 口地址：
- 命令状态口 7F00H
- PA 口 7F01H
- PB 口 7F02H
- PC 口 7F03H
- 定时器低 8 位 7F04H
- 定时器高 8 位 7F05H

操作软件：

```
MOV DPTR, #RAM 或 I/O 口地址
MOV A, #欲写内容或命令字
MOV @DPTR, A
```

布置作业：P220 5-11 抄题目在黑板上。

版书设计：

<p>一、8155 的结构和操作系统</p> <p>1、....</p> <p>2、.....</p> <p>3、...</p>	<p>二、8155 的工作方式及方式选择</p> <p>1、.....</p> <p>2、.....</p>
--	--

<p>三、8031 和 8155 的连接及应用</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>...</p> <p>作业:</p>
--	-----------------------

第二十三讲 键盘及其接口

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、熟悉键盘接口的特点及应用
- 2、掌握键盘的结构
- 3、掌握扫描法检测按键的编程方法

教学重点:

键盘的结构

教学难点:

扫描法检测按键

复习: 1、可编程并行接口芯片 8155 有几个 I/O 口?

2、几种工作方式? 由谁来确定?

一、按键操作存在的问题 —— 键抖动

键在电路中的连接如图 8-15 所示。当操作键时，其一对触点闭合或断开，引起 A 点电压的变化。A 点电压就用来向单片机输入键的通断状态。

由于机械触点的弹性作用，触点在闭合和断开瞬间的电接触情况不稳定，造成了电压信号的抖动现象，如图 8-15 所示。键的抖动时间一般为 5~10ms。这种现象会引起单片机对于一次键操作进行多次处理，因此须设法消除键接通或断时的抖动现象。去抖动的方法有硬件和软件两种。硬件去抖动和软件去抖动。

1、硬件消除抖动

2、软件去抖动

采用软件去抖动的方法是在单片机检测到有键按下时执行一个 10~20ms 的延时程序后再次检查该键电平是否仍保持闭合状态。如保持闭合状态，则确认为有键按下，否则从头检测。这样就能消除键的抖动影响。

二、按键分类

1、独立式键盘的结构

独立式键盘的结构如图 1 所示，这是最简单的键盘结构形式，每个按键的电路是独立

的，都有单独一根数据线输出键的通断状态。
单片机一条 I/O 口线对应一个按键。

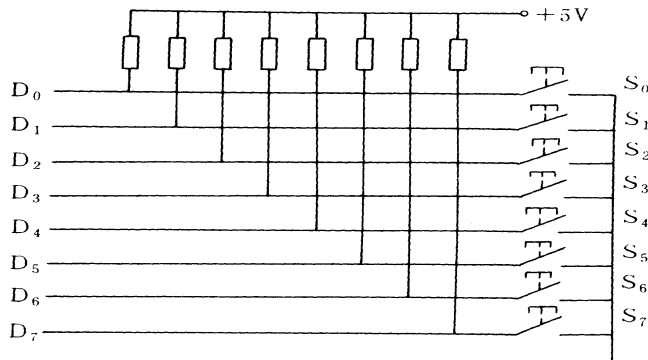


图 1 独立式键盘的结构

2、矩阵式键盘

若干 I/O 口线作行线；若干 I/O 口线作列线，在每个行列交点设置按键组成。如图 2 所示。

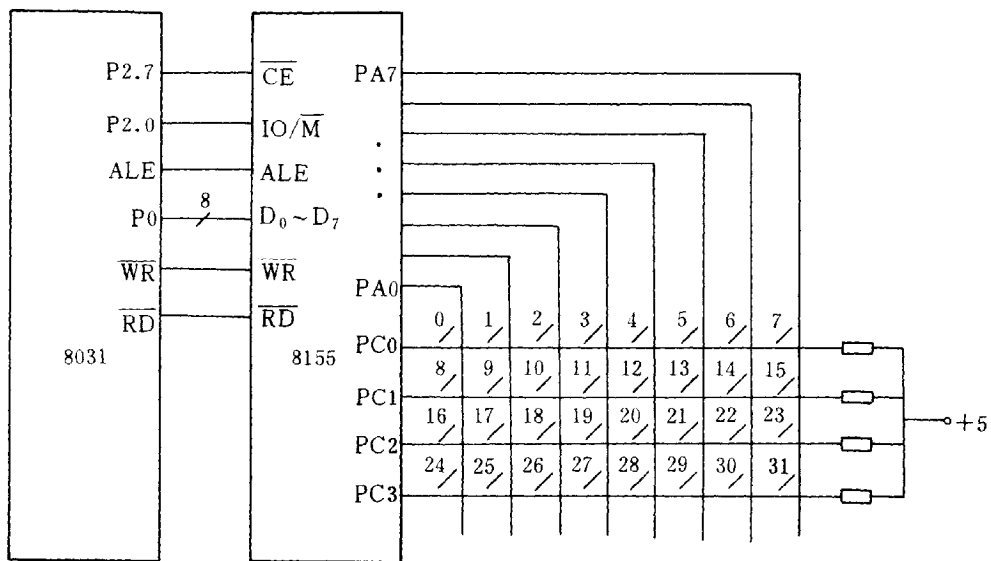


图 8-22 行列式键盘与 8031 的连接电路

图 2 矩阵式键盘结构

三、按键监测方法

- 1、监测有无键按下
- 2、按键的识别（哪一个键被按下）

扫描法 —— 有键按下时，读入行或列值，则为 0 值的行列交点的键便是。

下面以图 2 为例讲解扫描法的程序。

键扫描子程序的程序清单如下：

```

KEY1:  ACALL  KS1           ; 调用判断有无键闭合子程序
        JNZ   LK1           ; 有键闭合时, (A) ≠ 0, 转消抖动
        AJMP  KEY1         ; 无键闭合时, 转起始
LK1:   ACALL  D12MS        ; 调用 12ms 延时程序
        ACALL  KS1         ; 再次判断有无键闭合
        JNZ   LK2           ; 确认有键闭合, 转逐列逐行扫描
        AJMP  KEY1         ; 否则转起始
LK2:   MOV    R2, 70FEH    ; 送首列扫描字
        MOV    R4, 7001H   ; 送首列号
LK3:   MOV    DPTR, #7F01H ; 选中 8155PA 口
        MOV    A, R2
        MOVX  @DPTR, A     ; 送列扫描字
        INC   DPTR
        INC   DPTR
        MOVX  A, @DPTR     ; 读 8155PC 口
        JB   ACC.0, LONE   ; 第 0 行无键闭合, 转查第 1 行
        MOV  A, #00H       ; 第 0 行有键闭合, 该行首键号送 A
        AJMP LKP           ; 转去求键号
LONE:  JB   ACC.1, LTWO    ; 第 1 行无键闭合, 转查第 2 行
        MOV  A, #08H       ; 第 1 行有键闭合, 该行首键号送 A
        AJMP LKP
LTWO:  JB   ACC.2, LTHR    ; 第 2 行无键闭合, 转查第 3 行
        MOV  A, #10H       ; 第 2 行有键闭合, 该行首键号送 A
        AJMP LKP
LTHR:  JB   ACC.3, NEXT    ; 第 3 行无键闭合, 改查下 1 列
        MOV  A, #18H       ; 第 3 行有键闭合, 该行首键号送 A
LKP:   ADD   A, R4         ; 键号一行首键号+列号
        PUSH ACC           ; 键号进栈保护

```

```

LK2:  ACALL  K51
      JNZ   LK3      ; 等待键释放
      POP  ACC      ; 键释放, 键号送 A
      RET                ; 键扫描结束, (A) = 键号
NEXT:  INC   R4      ; 指向下一列
      MOV  A, R2     ; 判断 8 列扫描完否
      JNB ACC.7, KEY1 ; 8 列扫描完, 未查到闭合键, 转开始
      RL   A        ; 未扫描完, 列扫描字左移一位
      MOV  R2, A
      AJMP LK4      ; 转下一列扫描
K51:   MOV  DPTR, #7F01H ; 判断有无键闭合子程序
      MOV  A, #00H
      MOVX @DPTR, A   ; 从 PA 口送出全 0
      INC  DPTR
      INC  DPTR
    
```

布置作业: P220 5-11

版书设计:

<p>一、8155 的结构和操作系统</p> <p>1、.....</p> <p>2、.....</p> <p>3、...</p>	<p>二、8155 的工作方式及方式选择</p> <p>1、.....</p> <p>2、.....</p>
<p>三、8031 和 8155 的连接及应用</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>... 作业:</p>

第二十四讲 LED 显示接口

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、熟悉键盘接口的特点及应用
- 2、掌握键盘的结构
- 3、掌握扫描法检测按键的编程方法

教学重点:

键盘的结构

教学难点:

扫描法检测按键

复习: 1、可编程并行接口芯片 8155 有几个 I/O 口?

3、几种工作方式? 由谁来确定?

一、常用的显示器件

- 1、CRT 显示器
- 2、LCD 显示器
- 3、LED 显示器

二、LED 显示器件

- 1、单个发光二极管
- 2、七段数码管

三、LED 显示方式

有共阴极和共阳极两种。

1) 字形码表的产生, P192 表 8-1

2) 显示方式:

静态显示

在静态显示方式下, 每一位显示器的字段控制线是独立的。当显示某一字时, 该位的各字段线和字位线的电平不变, 也就是各字段的亮灭状态不变。静态显示方式下 LED 显示器的电路连接方法是: 每位 LED 的字位控制线门共阴极点或共阳极点连在一起, 接地或接 +5V; 其字段控制线 (a~dP) 分别接到一个 8 位口。

动态显示

利用人眼的视觉暂留效应。分时显示不同的数码管。

四、 LED 显示器接口方法

1、 硬件译码法

举例：现要求将单片机内存 30H-33H 中存放的四个数分高低位分别用 8 个数码管显示出来。

采用 4 线---7 线译码芯片 MC14495 实现单片机与数码管之间的连接。如图 1 所示。

1) 硬件连接

2) 控制软件

2、 软件译码法

举例 1

利用 8255A 实现静态显示方式将 8031 单片机内部 RAM 区 30H—32H 内的三个数在数码管中显示出来。

(1) 硬件连接如图 2 所示。

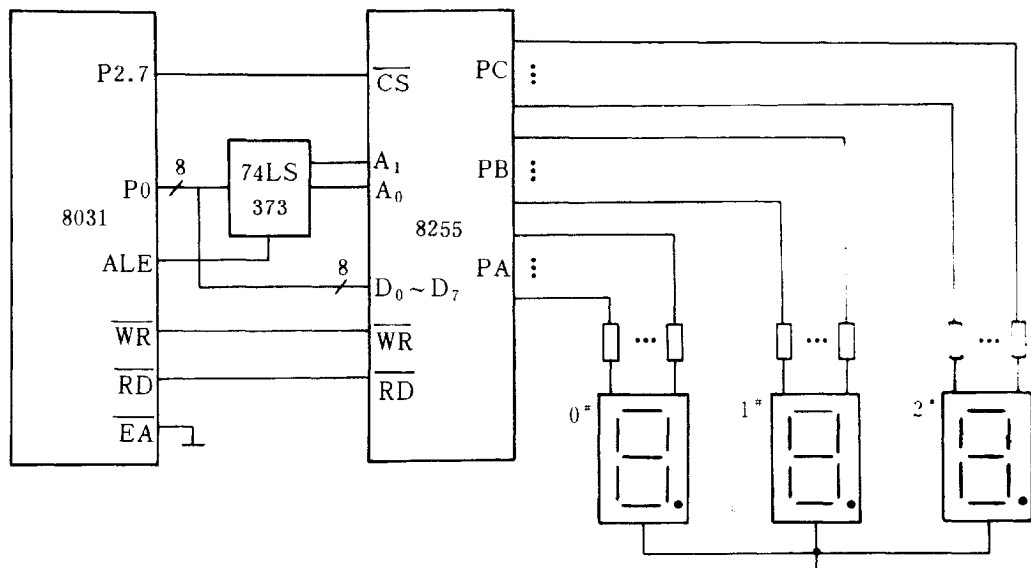


图 2

(2) 控制软件

现要求利用图 84 所示电路编一程序，将 8031 片内存储器 30H、31H、32H 单元的数值（十六进制数 0~F 之间）分别显示于 0#、1#、2# LED 显示器上。所编程序采用查表法。将对应于 0~F 这 16 个数码的字段码依次存放在以 TABLE 为起始地址的存储单元中。然后根据地址指针 RI 的内容查表得到对应的字段码，送相应的 UO 口输出。图 2 中 8255 作于方式 0 输出，A 口、B 口、C 口和控制口的地址为 7F00H~7F03H。程序如下：

```

DISP1: MOV     P2, #7FH
        MOV     R0, #03H      ; 指向 8255 控制口
        MOV     A, #80H
        MOVX    @R0, A        ; 写入方式控制字
        MOV     R0, #00H      ; 指向 8255 的 A 口
        MOV     R1, #30H      ; 显示单元指针
        MOV     DPTR, #TAB     ; 字段码表首址
LP:     MOV     A, @R1
        MOVC    A, @A+DPTR     ; 查字段码
        MOVX    @R0, A        ; 将字段码送 I/O 口
        INC     R0             ; 改 I/O 口地址
        INC     R1             ; 改显示单元地址
        CJNE    R1, 33H, LP    ; 判断是否结束
        RET
    
```

举例 2

利用 8155 实现动态显示方式，显示 2005 年 5 月 16 日。

(1) 硬件设计

连接如图 3 所示。图中采用了 8155 作为单片机与数码管的接口芯片。

同时考虑到有八个数码管，故采用了两片 7406 和三片 75452 作为驱动。

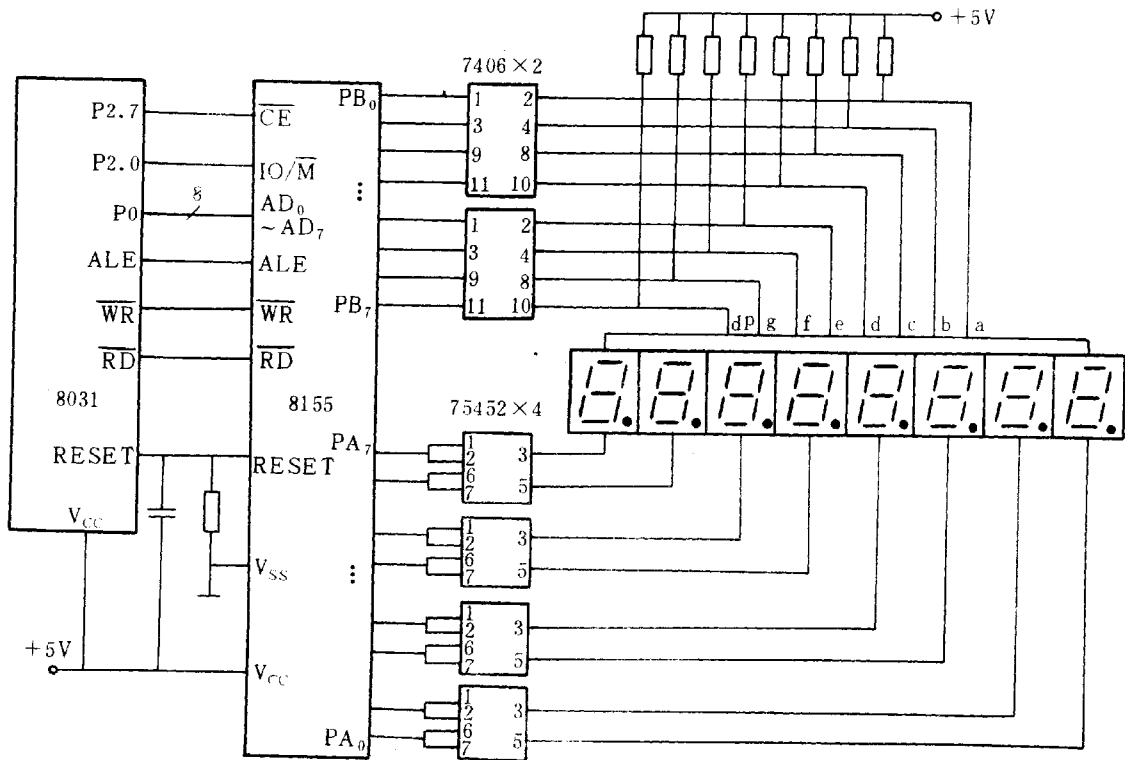


图 3

(2) 软件设计

小结：8155 有哪些功能？有几个并行 I/O 口？

如何去除按键抖动？按键查询有几种方法？

板书设计：

一、常用显示器 1、 2、 3、	二、LED 显示器显示方式 1、 2、
三、LED 显示方式 1、 2、 3、	四、LED 接口方法 1、 2、 3、

第二十五讲 D/A 转换接口

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 D/A 转换的作用
- 2、掌握 D/A 转换器的指标
- 3、掌握 0832 的结构及应用

教学重点:

0832 的结构及应用

教学难点:

0832 的结构及应用

复习: 1、LED 显示方式有哪几种?

- 4、软件译码和硬件译码法有何不同?

§ 5-7 D/A 转换接口

一、 D/A 转换的作用

二、 D/A 转换的重要指标

分辨率的定义: 最小输出电压与最大输出电压所对应的数字量之比。

如 10 位 D/A 转换器:

$$\text{分辨率} = 1 / (2^{10} - 1) = 1 / 1023 = 0.001$$

反映了 D/A 转换的灵敏度。

转换原理。

三、 8 位 D/A 转换器 DAC 0830 ~ 0832

- 8 位分辨率
- 具有两个输入数据寄存器 (8 位)
可单缓冲、双缓冲或直接输入, 实现多通道 D/A 同步转换输出。
- 须外接参考电压源
- 为电流输出型 D/A 转换器

1、DAC 0832 内部结构

P205 图 5-58

有 20 个引脚。

DI0 ~ DI7——数据输入线
 ILE —— 数据允许锁存
 /CS —— 输入寄存器选择
 /WR1 —— 输入寄存器的写选通
 /WR2 —— DAC 寄存器的写选通
 /XREF—— 数据传送信号
 Vref —— 基准电源输入
 Rfb —— 反信号输入（内有反馈电阻）
 Iout1、Iout2 —— 电流输出
 Vcc、AGND、DGND —— 电源、地

2、与 8031 单片机接口及操作软件

（1）单缓冲方式

P206 图 5-60

完成一次 D/A 转换：

```
MOV DPTR, #7FFFH ; DAC 0832 口地址
MOV A, #DATA ; 欲转换的数字量
MOVX @DPTR, A ; 转换
```

（2）双缓冲器同步方式

电路：图 5-61 （胶片）

图中：P2.5 选中 DAC 0832(1) }
 P2.6 选中 DAC 0832(2) } 输入锁存器
 P2.7 控制两芯片/XREF 同步转换

3、应用

例、利用单片机及 DAC 0832 产生阶梯波，DAC 0832 采用单缓冲方式，定时 1ms，增幅 10，10ms 一循环。

```
解：START: MOV A, #00H
            MOV DPTR, #7FFFH ; 转换器地址
            MOV R1, #0AH ; 10 个台阶（10ms）
LOOP: MOVX @DPTR, A ; 送欲转换数字量
            CALL DL1 ; 延时 1ms
            DJNZ R1, NEXT ; 10 个台阶未完继续增幅
            SJMP START
NEXT: ADD A, #10
            JMP LOOP
```

```

DL1: MOV 20H, #249
D1L: NOP
      NOP
      DJNZ 20H, D1L
      RET

```

如此，还可产生锯齿波，三角波等。

小结：

DAC 0832 的作用是什么？分辨率是多少位？

D/A 转换器的指标

D/A 转换器芯片 0832

布置作业： P220 5-10

版书设计：

<p>一、D/A 转换器的作用</p> <p>1、</p> <p>2、</p> <p>3、 ...</p>	<p>二、D/A 转换器的指标</p> <p>1、</p> <p>2、</p>
<p>三、D/A 转换器芯片 0832</p> <p>1、</p> <p>2、</p> <p>3、</p>	<p>小结：</p> <p>1、</p> <p>2、</p> <p>作业：</p>

第二十六讲 A/D 转换接口

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 A/D 转换的作用
- 2、掌握 A/D 转换器的指标
- 3、掌握 0809 的结构及应用

教学重点:

0809 的结构及应用

教学难点:

0809 的结构及应用

复习: 1、D/A 转换的作用和指标是什么?

2、常用的 D/A 转换器芯片有几种?

§ 5-8 A/D 转换器接口

一、常用 A/D 转换器的转换原理

1、逐次逼近式 图 5-66

2、双积分式 图 5-67

3、V/F 变换式

分辨率的定义: 输出数字量变化一个相邻数码所需要输入模拟电压的变化量。用公式表示为:

$$\text{满刻度电压}/2^n \quad (n \text{ 为位数})$$

二、ADC 0808 / 0809

8 路 8 位 A/D 转换器

- 采用逐次逼近式 A/D 转换原理 (时钟: 10 ~ 1280KHz)
- 模拟输入有 8 个通道 (任一通道模拟都可以转换为一个 8 位数字量输出)

1、ADC 0809 内部结构及引脚

图 5-72 (胶片)

图中 SAR 是逐次逼近寄存器。

引脚 (有 28 条):

IN0 ~ IN7—— 8 路模拟量输入通道的输入口;

- 2⁻¹ ~ 2⁻⁸ —— 8 位数字量输出口;
- START —— 启动输入口 } 启动转换
- ALE —— 通道地址锁存
- EOC —— 转换结束输出信号
- OE —— 输出允许
- CLK —— 时钟端
- ADDA、ADDB、ADDC —— 8 路模拟开关的 3 位地址选择端
- REF (+)、REF (-) —— 参电压输入端
- Vcc、GND —— 电源、地。

2、ADC 0809 与 8031 单片机接口设计

有查询方式、中断方式、等待延时方式。

片内无时钟，由单片机的 ALE (fosc/6) 经二分频后接入。

若 fosc = 6MHz，则接入时钟为 500KHz， 满足：10 ~ 1280KHz 的要求。

图 5-75 (胶片)

	P2.7 (片选)			地址码			输入通道		
	C	B	A	C	B	A			
7FF8H	0	×	×	×	×	0	0	0	IN0
		.							.
		.							.
		.							.
7FFFH	0	×	×	×	×	1	1	1	IN7

例、对通道 IN0，采样一次，转换，并将转换结果存到数据存储区。

解：

```

MAIN: MOV R1, #data ; 存数地址
      MOV DPTR, #7FF8H ; P2.7 = 0, 指向 IN0
      MOVX @DPTR, A ; 不论 A 中为何内容都启动转换
                        ; (使/WR 产生脉冲)
      MOV R6, #0AH ; 软件延时
DLAY: NOP
    
```

```

NOP
NOP
NOP
NOP
DJNZ R6, DLAY
MOVX A, @DPTR ; 读转换结果
MOV @R1 ,A ; 存储数据

```

小结: DAC 0832 的作用是什么? 分辨率是多少位?

ADC0809 的作用是什么? 有几个模拟通道?

布置作业: P220 5-10

版书设计:

<p>一、A/D 转换器的作用</p> <p>1、....</p> <p>2、.....</p> <p>3、...</p>	<p>二、A/D 转换器的指标</p> <p>1、.....</p> <p>2、.....</p>
<p>三、A/D 转换器芯片 0809</p> <p>1、.....</p> <p>2、.....</p> <p>3、.....</p>	<p>小结:</p> <p>1、.....</p> <p>2、.....</p> <p>作业:</p>

第二十七讲 A/D 转换接口设计

教学方法: 讲授法

授课时数: 2 学时

教学目的:

- 1、了解 A/D 转换的作用
- 2、掌握 A/D 转换器的指标
- 3、掌握 0809 的结构及应用

教学重点:

0809 的结构及应用

教学难点:

0809 的结构及应用

复习: 1、D/A 转换的作用和指标是什么?

2、常用的 D/A 转换器芯片有几种**组织教学:** 检查学生人数, 填写教学日志。

§ 5-8 A/D 转换接口的设计

一、硬件连线

图 P218 5-75

图中, $ALE = START = /(\overline{WR}+P2.7)$

$OE = /(\overline{RD}+P2.7)$

通道地址:	P27	P26	...	P20	P07	...	P0.3	P0.2	P0.1	P0.0
	A15	A14	...	A8	A7	...	A3	A2	A1	A0
7FFBH	0	×	...	×	×	...	×	0	0	0
			.						.	
			.						.	
			.						.	
7FFFH	0	×	...	×	×	...	×	1	1	1

二、软件设计方法

1、延时等待方式

分别 8 路模拟信号轮流采样一次, 并依次把结果转存到数据存储区的采样转换程序如下 (设数据区首地址 30H; 采样采用循环程序, R7 作计数器):

```
ORG 0000H
AJMP MAIN
```



```

        ORG 0100H
MAIN:   MOV R1, #30H
        MOV DPTR, #7FF8H
        MOV R7, #08H
LOOP:   MOVX @DPTR, A
        MOV R6, #0FH
DLAY:   NOP
        NOP
        NOP
        NOP
        DJNZ R6, DLAY
        MOVX A, @DPTR
        MOV @R1, A
        INC DPTR
        INC R1
        DJNZ R7, LOOP
        RET

```

2、 中断方式

合上 K1，启动转换后，经 t_{EOC} 时间，EOC 输出负脉冲波形，表示某通道转换已结束，单片机可以取走数据。

让八个通道轮流转换一遍。

```

        ORG 0000H
        AJMP MAIN2
        ORG 0013H
        AJMP INTR1
        ORG 0300H
MAIN2:   MOV R1, #30H
        SETB IT1           ; 下降沿触发/INT1 中断
        SETB EX1           ; 允许/INT1 中断
        SETB EA            ; 开中断
        MOV R7, #08H       ; 待转换的通道个数，即循环次数
        MOV DPTR, #7FF8H; 指向通道 0
        MOV @DPTR, A      ; 启动首次转换
WAIT:    CJNE R7, #00H, $   ; 等待中断
        CLR EX1            ; 8 个通道转换完成，结束
        SJMP $

```

```

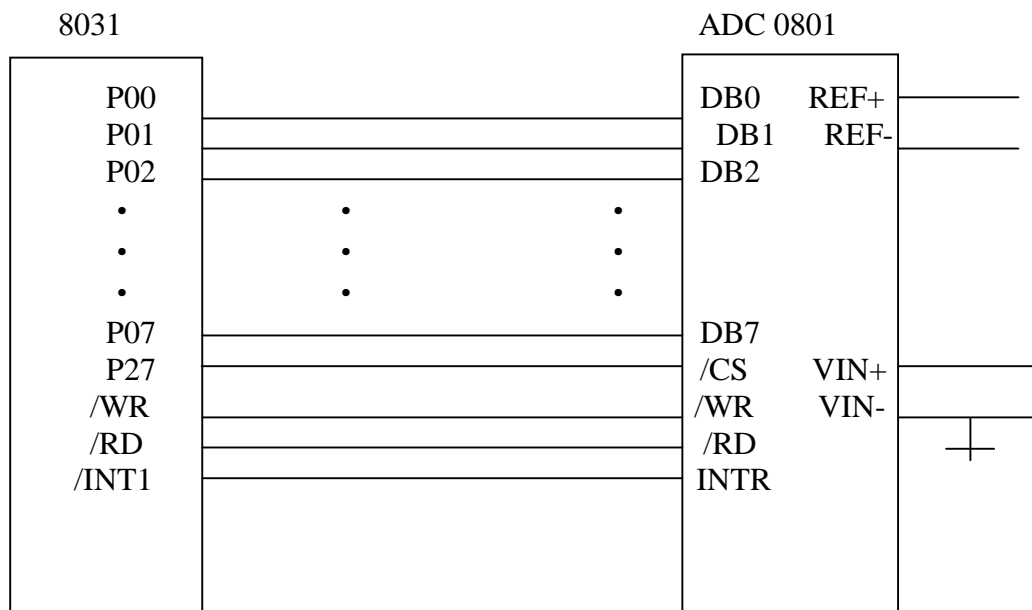
        ORG 0500H
INTR1:  MOVX A, @DPTR ; 读取转换结果
        MOV @R1, A   ; 存数
        INC DPTR     ; 通道号加 1
        INC R1
        DEC R7
        MOVX @DPTR, A ; 启动下一次转换
        RETI
        END
    
```

3、 查询方式

启动转换后，经 t_{OEC} 时间后，转换结束，EOC 输出负脉冲。

所以，启动转换后，延时 t_{OEC} 时间，检查 EOC 电平，若为 0 说明仍在转换中，若为高电平说明转换结束，书中只针对 IN0 通道模拟量转换一次而编程。

ADC 0801 ~ 0804 转换芯片与 ADC 0809 芯片的区别是：
ADC 0801 ~ 0804 是单通道模拟信号。



上图中，模拟通道地址是：7FFFH。
编程同 ADC 0809，只是仅有一个通道。

小结：

1、A/D 转换原理分两大类:

直接型 、 间接型;

按转换原理分 A/D 转换器常见三大种类:

逐次逼近式 (直接型)、双积分式 (间接型)、V/F 变换式。

2、ADC 0809

8 路模拟通道, 由 ADDA、ADDB、ADDC 决定, (START、ALE)

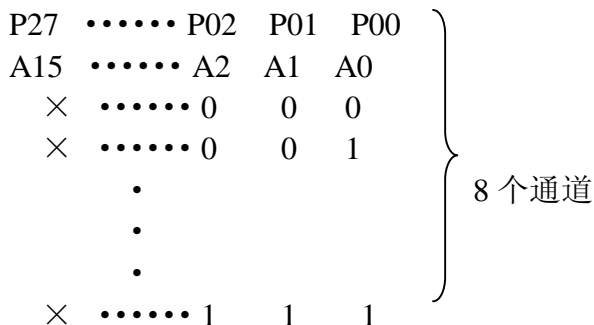
起动转换, (EOC、OE) 转换结束; 从起动转换开始, 经 tEOC 时间, EOC 输出负脉冲, 因此查询 EOC 的上升沿便可知道转换是否结束。

3、ADC 0801 ~ 0804 是单通道模拟量

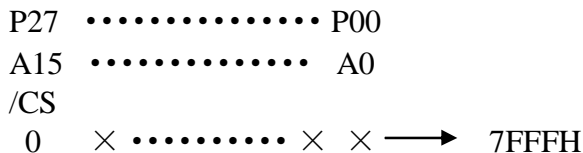
/WR 起动转换;

INTR 指出转换结束。

4、ADC 0809 通道地址



ADC 0801 ~ 0804 的地址:



布置作业: 抄题目在黑板上。

版书设计:

<p>一、硬件连线 1、....</p> <p>2、.....</p>	<p>二、软件设计方法</p> <p>1、.....</p> <p>2、.....</p>
<p>3、.....</p> <p>4、.....</p>	<p>小结:</p> <p>作业:</p>

