第一章 绪论

第一节 单片机

单片机即单片机微型计算机，是将计算机主机(CPU、内存和 I/O 接口)集成在一小块硅片上的微型机。

第二节 单片机的历史与现状

第一阶段（1976~1978 年）：低性能单片机的探索阶段。以 Intel 公司的 MCS-48 为代表，采用了单片结构，即在一块芯片内含有 8 位 CPU、定时/计数器、并行 I/O 口、RAM 和 ROM 等。主要用于工业领域。

第二阶段（1978~1982 年）：高性能单片机阶段，这一类单片机带有串行 I/O 口，8 位数据线、16 位地址线可以寻址的范围达到 64K 字节、控制总线、较丰富的指令系统等。这类单片机的应用范围较广，并在不断的改进和发展。

第三阶段（1982~1990 年）：16 位单片机阶段。16 位单片机除 CPU 为 16 位外，片内 RAM 和 ROM 容量进一步增大，实时处理能力更强，体现了微控制器的特征。例如 Intel 公司的 MCS-96 主振频率为 12M，片内 RAM 为 232 字节，ROM 为 8K 字节，中断处理能力为 8 级，片内带有 10 位 A/D 转换器和高速输入/输出部件等。

第四阶段（1990 年~）：微控制器的全面发展阶段，各公司的产品在尽量兼容的同时，向高速、强运算能力、寻址范围大以及小型廉价方面发展。

第三节 单片机的应用领域

一、 单片机在仪器仪表中的应用
二、 单片机在机电一体化中的应用
三、 单片机在智能接口和多机系统中的应用
四、 单片机在生活中的应用

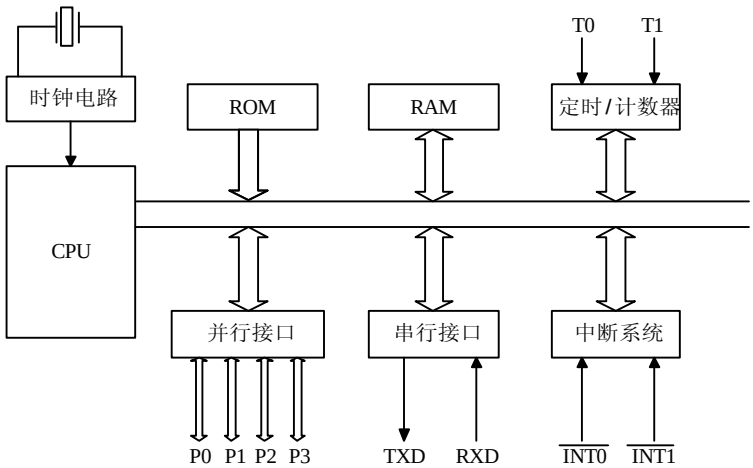
第二章 硬件结构

第一节 MCS-51 单片机及其演变

特点

（1）一个 8 位微处理器 CPU。
（2）数据存储器 RAM 和特殊功能寄存器 SFR。
（3）内部程序存储器 ROM。
（4）两个定时/计数器，用以对外部事件进行计数，也可用作定时器。
（5）四个 8 位可编程的 I/O（输入/输出）并行端口，每个端口既可做输入，也可做输出。
（6）一个串行端口，用于数据的串行通信。
（7）中断控制系统。
（8）内部时钟电路。

第二节 80C51 单片机的基本结构



1) 中央处理器（CPU）

中央处理器是单片机的核心，完成运算和控制功能。MCS-51 的 CPU 能处理 8 位二进制数或代码。

2) 内部数据存储器（内部 RAM）

8051 芯片中共有 256 个 RAM 单元，但其中后 128 单元被专用寄存器占用，能作为寄存器供用户使用的只是前 128 单元，用于存放可读写的数据。因此通常所说的内部数据存

储器就是指前 128 单元，简称内部 RAM。

3) 内部程序存储器（内部 ROM）

8051 共有 4 KB 掩膜 ROM，用于存放程序、原始数据或表格，因此，称之为程序存储器，简称内部 ROM。

4) 定时/计数器

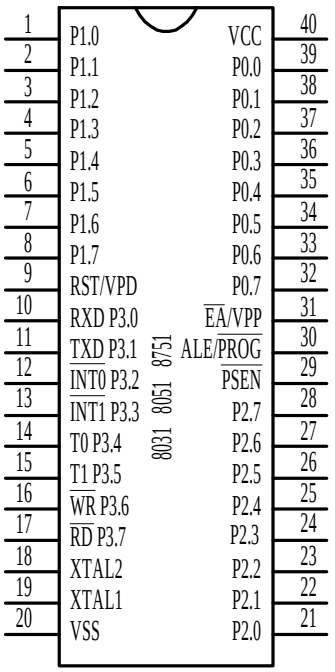
8051 共有两个 16 位的定时/计数器，以实现定时或计数功能，并以其定时或计数结果对计算机进行控制。

5) 并行 I/O 口

MCS-51 共有 4 个 8 位的 I/O 口（P0、P1、P2、P3），以实现数据的并行输入/输出。在实训中我们已经使用了 P1 口，通过 P1 口连接 8 个发光二极管。

第三节 80C51 单片机的引脚功能

MCS-51 是标准的 40 引脚双列直插式集成电路芯片，引脚排列请参见图



P0.0 ～ P0.7： P0 口 8 位双向口线。

P1.0 ～ P1.7 ： P1 口 8 位双向口线。

P2.0 ～ P2.7 ： P2 口 8 位双向口线。

P3.0 ～ P3.7 ： P3 口 8 位双向口线。

ALE：地址锁存控制信号。在系统扩展时，ALE 用于控制把 P0 口输出的低 8 位地址锁存起来，以实现低位地址和数据的隔离。此外，由于 ALE 是以晶振 1/6 的固定频率输出的正脉冲，因此,可作为外部时钟或外部定时脉冲使用。

PSEN：外部程序存储器读选通信号。在读外部 ROM 时，PSEN 有效（低电平），以实现外部 ROM 单元的读操作。

EA：访问程序存储控制信号。当信号为低电平时，对 ROM 的读操作限定在外部程序存储器；当信号为高电平时，对 ROM 的读操作是从内部程序存储器开始，并可延至外部程序存储器。

RST：复位信号。当输入的复位信号延续两个机器周期以上的高电平时即为有效，用以完成单片机的复位初始化操作。

XTAL1 和 XTAL2：外接晶体引线端。当使用芯片内部时钟时，此二引线端用于外接石英晶体和微调电容；当使用外部时钟时，用于接外部时钟脉冲信号。

VSS：地线。

VCC：+5 V 电源。

以上是 MCS-51 单片机芯片 40 条引脚的定义及简单功能说明，读者可以对照实训电路找到相应引脚，在电路中查看每个引脚的连接使用。

P3 口线的第二功能。P3 的 8 条口线都定义有第二功能

第四节 存储器结构

MCS-51 单片机的芯片内部有 RAM 和 ROM 两类存储器，即所谓的内部 RAM 和内部 ROM

MCS-51 内部程序存储器

MCS-51 的程序存储器用于存放编好的程序和表格常数。8051 片内有 4 KB 的 ROM，8751 片内有 4 KB 的 EPROM，8031 片内无程序存储器。MCS-51 的片外最多能扩展 64 KB 程序存储器，片内外的 ROM 是统一编址的。如端保持高电平，8051 的程序计数器 PC 在 0000H～0FFFH 地址范围内 (即前 4 KB 地址) 是执行片内 ROM 中的程序，当 PC 在 1000H～FFFFH 地址范围时，自动执行片外程序存储器中的程序；当保持低电平时，只能寻址外部程序存储器，片外存储器可以从 0000H 开始编址。

MCS-51 的程序存储器中有些单元具有特殊功能，使用时应予以注意。

其中一组特殊单元是 0000H～0002H。系统复位后，(PC)=0000H，单片机从 0000H 单元开始取指令执行程序。如果程序不从 0000H 单元开始，应在这三个单元中存放一条无条件转移指令，以便直接转去执行指定的程序。

还有一组特殊单元是 0003H～002AH，共 40 个单元。这 40 个单元被均匀地分为 5 段，作为 5 个中断源的中断地址区。其中：

0003H～000AH 外部中断 0 中断地址区
000BH～0012H 定时/计数器 0 中断地址区
0013H～001AH 外部中断 1 中断地址区
001BH～0022H 定时/计数器 1 中断地址区
0023H～002AH 串行中断地址区

中断响应后，按中断种类，自动转到各中断区的首地址去执行程序，因此在中断地址区中理应存放中断服务程序。但通常情况下，8 个单元难以存下一个完整的中断服务程序，因此通常也是从中断地址区首地址开始存放一条无条件转移指令，以便中断响应后，通过中断地址区，再转到中断服务程序的实际入口地址。

MCS-51 内部数据存储器

内部数据存储器低 128 单元

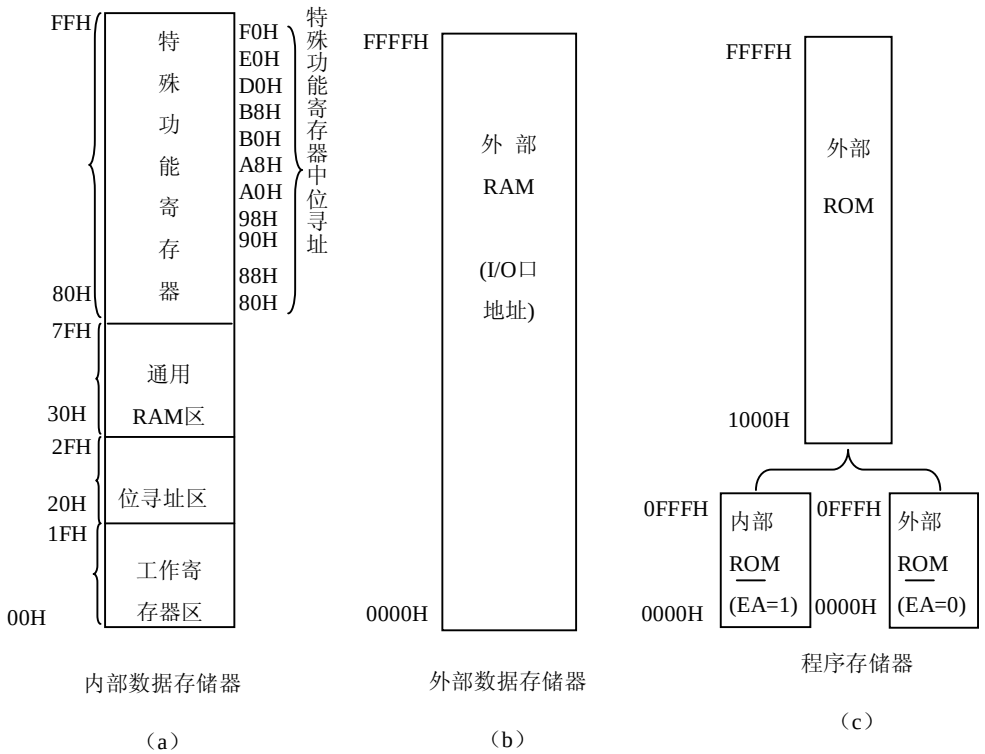
8051 的内部 RAM 共有 256 个单元，通常把这 256 个单元按其功能划分为两部分：低 128 单元（单元地址 00H～7FH）和高 128 单元（单元地址 80H～FFH）。如图所示为低 128 单元的配置图。

寄存器区

8051 共有 4 组寄存器，每组 8 个寄存单元（各为 8），各组都以 R0～R7 作寄存单元编号。寄存器常用于存放操作数中间结果等。由于它们的功能及使用不作预先规定，因此称之为通用寄存器，有时也叫工作寄存器。4 组通用寄存器占据内部 RAM 的 00H～1FH 单元地址。

在任一时刻，CPU 只能使用其中的一组寄存器，并且把正在使用的那组寄存器称之为当前寄存器组。到底是哪一组，由程序状态字寄存器 PSW 中 RS1、RS0 位的状态组合来决定。

通用寄存器为 CPU 提供了就近存储数据的便利，有利于提高单片机的运算速度。此外，使用通用寄存器还能提高程序编制的灵活性，因此，在单片机的应用编程中应充分



利用这些寄存器，以简化程序设计，提高程序运行速度。

位寻址区

内部 RAM 的 20H～2FH 单元，既可作为一般 RAM 单元使用，进行字节操作，也可以对单元中每一位进行位操作，因此把该区称之为位寻址区。位寻址区共有 16 个 RAM 单元，计 128 位，地址为 00H～7FH。MCS-51 具有布尔处理机功能，这个位寻址区可以构成布尔处理机的存储空间。这种位寻址能力是 MCS-51 的一个重要特点。

用户 RAM 区

在内部 RAM 低 128 单元中，通用寄存器占去 32 个单元，位寻址区占去 16 个单元，剩下 80 个单元，这就是供用户使用的一般 RAM 区，其单元地址为 30H～7FH。对用户 RAM 区的使用没有任何规定或限制，但在一般应用中常把堆栈开辟在此区中。

内部数据存储器高 128 单元

内部 RAM 的高 128 单元是供给专用寄存器使用的，其单元地址为 80H～FFH。因这些寄存器的功能已作专门规定，故称之为专用寄存器（Special Function Register），也可称为特殊功能寄存器。

第五节 特殊功能存储器 SFR

8051 共有 21 个专用寄存器，现把其中部分寄存器简单介绍如下：

程序计数器（PC—Program Counter）。在实训中，我们已经知道 PC 是一个 16 位的计数器，它的作用是控制程序的执行顺序。其内容为将要执行指令的地址，寻址范围达 64 KB。PC 有自动加 1 功能，从而实现程序的顺序执行。PC 没有地址，是不可寻址的，因此用户无法对它进行读写，但可以通过转移、调用、返回等指令改变其内容，以实现程序的转移。因地址不在 SFR（专用寄存器）之内，一般不计作专用寄存器。

累加器（ACC—Accumulator）。累加器为 8 位寄存器，是最常用的专用寄存器，功能较多，地位重要。它既可用于存放操作数，也可用来存放运算的中间结果。MCS-51 单片机中大部分单操作数指令的操作数就取自累加器，许多双操作数指令中的一个操作数也取自累加器。

B 寄存器。B 寄存器是一个 8 位寄存器，主要用于乘除运算。乘法运算时，B 存乘数。乘法操作后，乘积的高 8 位存于 B 中，除法运算时，B 存除数。除法操作后，余数存于 B 中。此外，B 寄存器也可作为一般数据寄存器使用。

程序状态字（PSW—Program Status Word）。程序状态字是一个 8 位寄存器，用于存放程序运行中的各种状态信息。其中有些位的状态是根据程序执行结果，由硬件自动设置的，而有些位的状态则使用软件方法设定。PSW 的位状态可以用专门指令进行测试，也可以用指令读出。一些条件转移指令将根据 PSW 有些位的状态，进行程序转移。PSW 的各位定义如下：

PSW 位地址	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
字节地址	CY	AC	F0	RS1	RS0	OV	F1	P

除 PSW.1 位保留未用外，其余各位的定义及使用如下：
CY（PSW.7）——进位标志位。CY 是 PSW 中最常用的标志位。其功能有二：一是存放算术运算的进位标志，在进行加或减运算时，如果操作结果的最高位有进位或借位时，CY 由硬件置“1”，否则清“0”；二是在位操作中，作累加位使用。位传送、位与位或等位操作，操作位之一固定是进位标志位。
AC（PSW.6）——辅助进位标志位。在进行加减运算中，当低 4 位向高 4 位进位或借位时，AC 由硬件置“1”，否则 AC 位被清“0”。在 BCD 码调整中也要用到 AC 位状态。
F0（PSW.5）——用户标志位。这是一个供用户定义的标志位，需要利用软件方法置位或复位，用以控制程序的转向。
RS1 和 RS0（PSW.4，PSW.3）——寄存器组选择位。它们被用于选择 CPU 当前使用的通用寄存器组。通用寄存器共有 4 组，其对应关系如下：
00：0 组　01：1 组　10：2 组　11：3 组
这两个选择位的状态是由软件设置的，被选中的寄存器组即为当前通用寄存器组。但当单片机上电或复位后，RS1 RS0=00。
OV（PSW.2）——溢出标志位。在带符号数加减运算中，OV=1 表示加减运算超出了累加器 A 所能表示的符号数有效范围（-128 ～ +127），即产生了溢出，因此运算结果是错误的，否则，OV=0 表示运算正确，即无溢出产生。
P（PSW.0）——奇偶标志位。表明累加器 A 中内容的奇偶性。如果 A 中有奇数个“1”，则 P 置“1”，否则置“0”。凡是改变累加器 A 中内容的指令均会影响 P 标志位。此标志位对串行通信中的数据传输有重要的意义。在串行通信中常采用奇偶校验的办法来校验数据传输的可靠性。

数据指针（DPTR）。数据指针为 16 位寄存器。编程时，DPTR 既可以按 16 位寄存器使用，也可以按两个 8 位寄存器分开使用，即：DPH　DPTR 高位字节，DPL　DPTR 低位字节。DPTR 通常在访问外部数据存储器时作地址指针使用。由于外部数据存储器的寻址范围为 64 KB，故把 DPTR 设计为 16 位。

堆栈指针（SP—Stack Pointer）。堆栈是一个特殊的存储区，用来暂存数据和地址，它是按“先进后出”的原则存取数据的。堆栈共有两种操作：进栈和出栈。由于 MCS-51 单片机的堆栈设在内部 RAM 中，因此 SP 是一个 8 位寄存器。系统复位后，SP 的内容为 07H，

从而复位后堆栈实际上是从 08H 单元开始的。但 08H～1FH 单元分别属于工作寄存器 1～3 区，如程序要用到这些区，最好把 SP 值改为 1FH 或更大的值。

对专用寄存器的字节寻址问题作如下几点说明：

(1)21 个可字节寻址的专用寄存器是不连续地分散在内部 RAM 高 128 单元之中，尽管还余有许多空闲地址，但用户并不能使用。

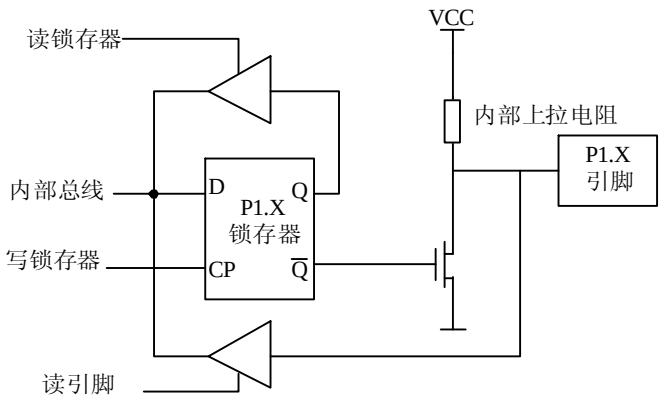
(2) 程序计数器 PC 不占据 RAM 单元，它在物理上是独立的，因此是不可寻址的寄存器。

(3) 对专用寄存器只能使用直接寻址方式，书写时既可使用寄存器符号，也可使用寄存器。

第六节 输入输出端口

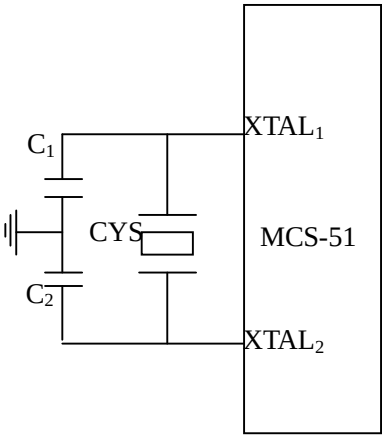
单片机芯片内还有一项主要内容就是并行 I/O 口。MCS-51 共有 4 个 8 位的并行 I/O 口，分别记作 P0、P1、P2、P3。每个口都包含一个锁存器、一个输出驱动器和输入缓冲器。实际上，它们已被归入专用寄存器之列，并且具有字节寻址和位寻址功能。

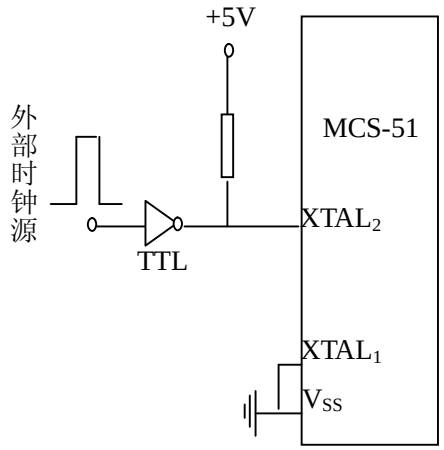
在访问片外扩展存储器时，低 8 位地址和数据由 P0 口分时传送，高 8 位地址由 P2 口传送。在无片外扩展存储器的系统中，这 4 个口的每一位均可作为双向的 I/O 端口使用。



第七节 时钟电路

在 MCS-51 芯片内部有一个高增益反相放大器，其输入端为芯片引脚 XTAL1，其输出端为引脚 XTAL2 。而在芯片的外部，XTAL1 和 XTAL2 之间跨接晶体振荡器和微调电容，从而构成一个稳定的自激振荡器，这就是单片机的时钟电路。



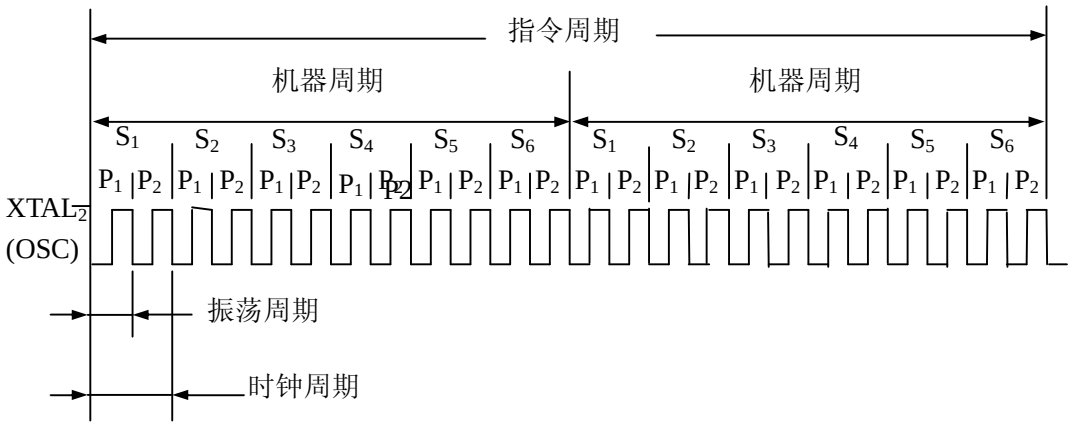


1．振荡周期：为单片机提供时钟信号的振荡源的周期。

2．时钟周期：是振荡源信号经二分频后形成的时钟脉冲信号。

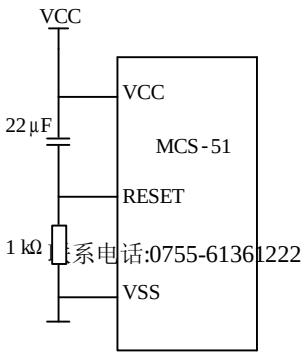
3．机器周期：通常将完成一个基本操作所需的时间称为机器周期。

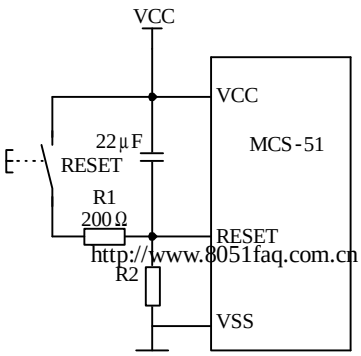
4．指令周期：是指 CPU 执行一条指令所需要的时间。一个指令周期通常含有 1～4 个机器周期。



第八节 复位电路

单片机复位是使 CPU 和系统中的其他功能部件都处在一个确定的初始状态，并从这个状态开始工作，例如复位后 PC=0000H，使单片机从第一个





单元取指令。实训中已经看出，无论是在单片机刚开始接上电源时，还是断电后或者发生故障后都要复位，所以我们必须弄清楚 MCS-51 型单片机复位的条件、复位电路和复位后状态。

第三章 MCS-51 指令系统

第一节 指令格式

采用助记符表示的汇编语言指令格式如下：

标号：	操作码	操作数或操作数地址	；注释

标号是程序员根据编程需要给指令设定的符号地址，可有可无；标号由 1~8 个字符组成，第一个字符必须是英文字，不能是数字或其它符号；标号后必须用冒号。
操作码表示指令的操作种类，如 MOV 表示数据传送操作，ADD 表示加法操作等。
操作数或操作数地址表示参加运算的数据或数据的有效地址。操作数一般有以下几种形式：没有操作数项，操作数隐含在操作码中，如 RET 指令；只有一个操作数，如 CPL A 指令；有两个操作数，如 MOV A,#00H 指令，操作数之间以逗号相隔；有三个操作数，如 CJNE A,#00H,NEXT 指令，操作数之间也以逗号相隔。
注释是对指令的解释说明，用以提高程序的可读性；注释前必须加分号。

第二节 寻址方式

寻找操作数地址的方式称为寻址方式。

1. 寄存器寻址

寄存器寻址是指将操作数存放于寄存器中，寄存器包括工作寄存器 R0~R7、累加器 A、通用寄存器 B、地址寄存器 DPTR 等。例如，指令 MOV R1,A 的操作是把累加器 A 中的数据传送到寄存器 R1 中，其操作数存放在累加器 A 中，所以寻址方式为寄存器寻址。

如果程序状态寄存器 PSW 的 RS1RS0=01（选中第二组工作寄存器，对应地址为 08H～0FH），设累加器 A 的内容为 20H，则执行 MOV R1，A 指令后，内部 RAM 09H 单元的值就变为 20H。

2. 直接寻址

直接寻址是指把存放操作数的内存单元的地址直接写在指令中。在 MCS-51 单片机中，可以直接寻址的存储器主要有内部 RAM 区和特殊功能寄存器 SFR 区。

例如，指令 MOV A，3AH 执行的操作是将内部 RAM 中地址为 3AH 的单元内容传送到累加器 A 中，其操作数 3AH 就是存放数据的单元地址，因此该指令是直接寻址。

3. 立即数寻址

立即数寻址是指将操作数直接写在指令中。

例如，指令 MOV A，#3AH 执行的操作是将立即数 3AH 送到累加器 A 中，该指令就是立即数寻址。

4. 寄存器间接寻址

寄存器间接寻址是指将存放操作数的内存单元的地址放在寄存器中，指令中只给出该寄存器。执行指令时，首先根据寄存器的内容，找到所需要的操作数地址，再由该地址找到操作数并完成相应操作。

在 MCS-51 指令系统中，用于寄存器间接寻址的寄存器有 R0、R1 和 DPTR，称为寄存器间接寻址寄存器。

设 R0=3AH，内部 RAM 3AH 中的值是 65H，则指令 MOV A，@R0 的执行结果是累加器 A 的值为 65H。

5. 变址寻址

变址寻址是指将基址寄存器与变址寄存器的内容相加，结果作为操作数的地址。DPTR 或 PC 是基址寄存器，累加器 A 是变址寄存器。该类寻址方式主要用于查表操作。

例如，指令 MOVC A，@A+DPTR 执行的操作是将累加器 A 和基址寄存器 DPTR 的内容相加，相加结果作为操作数存放的地址，再将操作数取出来送到累加器 A 中。

设累加器 A=02H，DPTR=0300H，外部 ROM 中，0302H 单元的内容是 55H，则指令 MOVC A，@A+DPTR 的执行结果是累加器 A 的内容为 55H。

6. 相对寻址

相对寻址是指程序计数器 PC 的当前内容与指令中的操作数相加，其结果作为跳转指令的转移地址（也称目的地址）。该类寻址方式主要用于跳转指令。

例如，指令 SJMP 54H 执行的操作是将 PC 当前的内容与 54H 相加，结果再送回 PC 中，成为下一条将要执行指令的地址。

设指令 SJMP 54H 的机器码 80H 54H 存放在 2000H 处，当执行到该指令时，先从 2000H 和 2001H 单元取出指令，PC 自动变为 2002H；再把 PC 的内容与操作数 54H 相加，形成目标地址 2056H，再送回 PC，使得程序跳转到 2056H 单元继续执行。

7. 位寻址

位寻址是指按位进行的寻址操作，而上述介绍的指令都是按字节进行的寻址操

作。MCS-51 单片机中，操作数不仅可以按字节为单位进行操作，也可以按位进行操作。当我们把某一位作为操作数时，这个操作数的地址称为位地址。
位寻址区包括专门安排在内部 RAM 中的两个区域：一是内部 RAM 的位寻址区，地址范围是 20H~2FH，共 16 个 RAM 单元，位地址为 00H~7FH；二是特殊功能寄存器 SFR 中有 11 个寄存器可以位寻址，参见有关章节中位地址定义。

第三节 数据操作和指令类型

MCS-51 单片机指令系统包括 111 条指令，按功能可以划分为以下 5 类
数据传送指令（29 条）
算术运算指令（24 条）
逻辑运算指令（24 条）
控制转移指令（17 条）
位操作指令（17 条）

第四节 数据传送指令

数据传送指令是 MCS-51 单片机汇编语言程序设计中使用最频繁的指令，包括内部 RAM、寄存器、外部 RAM 以及程序存储器之间的数据传送。

数据传送操作是指把数据从源地址传送到目的地址，源地址内容不变。

1. 以累加器 A 为目的操作数的指令

```
MOV   A,   #data     ;  A← #data
MOV   A,   Rn        ;  n=0~7,  A← (Rn)
MOV   A,   @Ri       ;  i=0,1   ,  A← ((Ri))
MOV   A,   direct    ;  A← (Rn)   direct 为内部 RAM 或 SFR 地址
```

2. 以 Rn 为目的操作数的指令

```
MOV   Rn,   A        ; Rn ← (A) ,    n=0~7
MOV   Rn ,  direct   ; Rn ← (direct)
MOV   Rn ,  #data    ; Rn ← #data
```

3.以直接地址为目的操作数的指令

```
MOV   direct,  A        ; direct← (A)
MOV   direct,  Rn       ;  direct← (Rn) ,        n=0~7
MOV   direct,  @Ri      ;  direct← (( Ri ) ) ,   i=0,1
MOV   direct,  direct   ; direct← (direct)
```

MOV direct, #data ；direct← ＃data

4. 以寄存器间接地址为目的操作数指令

MOV @Ri， A ； ((Ri))← (A) ， i=0,1
MOV @Ri ， direct ； ((Ri))← (direct)
MOV @Ri ，＃data ； ((Ri))← ＃data

字节交换指令

XCH A， Rn ； (A) ← →(R**i**)
XCH A， direct ； (A) ← →(direct)
XCH A， @Ri ； (A) ← →(R**i**)

半字节交换指令

XCHD A, @Ri ；(A)0-3 ← →((Ri)) 0-3

累加器 A 与外部数据传输指令

MOVX A, @DPTR ； A← ((DPTR)) 地址范围 64K
MOVX A, @Ri ； A← ((R**i**)) 地址范围 0～255
MOVX @DPTR, A ； (DPTR) ←(A)
MOVX @Ri , A ； (R**i**) ←(A)

查表指令

1）MOVC A, @A+DPTR ； A ←((A)+(DPTR))
2）MOVC A, @A+PC ； A ←((A)+(pc))

第五节 算术运算指令

加法指令（Addtion）

ADD A, Rn ； A←（A）+（Rn）
ADD A, @Ri ；A←（A）+（(Ri)）
ADD A, direct ；A←（A)+(direct)
ADD A, ＃data ；A←（A)+#data

带进位加法指令

ADDC A, Rn ； A←(A)+(Rn)+(Cy)
ADDC A, @Ri ；A←(A)+((Ri)) +(Cy)
ADDC A, direct ；A←（A)+(direct) +(Cy)

ADDC A, ＃data ；A←（A)+#data +(Cy)

加 1 指令

INC A ；A←(A)+1
INC Ri ；R**i**←（A)+1
INC direct ；direct←（direct)+1
INC @Ri ；(R**i**)←((R**i**))+1
INC DPTR ；DPTR←(DPTR)+1

十进制调整指令

DA A

带借位减法指令（Subtraction）

SUBB A, Rn ； A←（A)-(Rn)-(Cy）
SUBB A, @Ri ；A←（A)-((Ri)) -(Cy）
SUBB A, direct ；A←（A)-(direct) -(Cy）
SUBB A, ＃data ；A←（A)-#data -(Cy）

减 1 指令（Decrease）

DEC A ；A←(A)－1
DEC Ri ；R**i**←（A)－1
DEC direct ；direct←（direct)－1
DEC @Ri ；(R**i**)←((R**i**))－1

乘法指令（Multiplication）

MUL AB

除法指令（Division）

DIV AB

第六节 逻辑运算指令

简单逻辑操作指令

CLR A ； A← “0”
CPL A ； A← A
SWAP A ； A0~3 A4~7

左循环指令 （Rotate Accumulator Left）
RL A

带进位左循环指令（Rotate Accumulator Left through Carry flag）
RLC A

右循环指令（Rotate Accumulator Right）
RR A

带进位右循环指令（Rotate A Right with C）
RRC A

逻辑与指令
ANL A, Rn
ANL A, direct
ANL A, ＃data
ANL A, @Ri
ANL direct, A
ANL direct, ＃data

逻辑或指令
ORL A, Rn
ORL A, direct
ORL A, ＃data
ORL A, @Ri
ORL direct, A
ORL direct, ＃data

逻辑异或指令
XRL A, Rn
XRL A, direct
XRL A, ＃data
XRL A, @Ri
XRL direct, A
XRL direct, ＃data

第七节 位操作指令

数据位传送指令

MOV C, bit ；bit 可直接寻址位 C←(bit)

MOV bit，C ；C 进位位 (bit) ← C

位变量修改指令

CLR C ；将 C＝0

CLR bit

CPL C ；将 C 求反再存入 C

CPL bit ；将 bit 求反再存入 bit

SETB C ；将 C＝1

SETB bit ； (bit) ← 1

位变量逻辑指令

ANL C, bit ANL C, bit ORL C, bit ORL C, bit

第八章 控制转移指令

跳转指令

短跳指令 AJMP addr11

PC←addr11，跳转范围 2k

长跳指令 LJMP addr16

PC←addr16，跳转范围 64k

间接跳转指令 JMP @A+DPTR

PC←（A）＋（DPTR)

相对转移指令 SJMP rel

条件转移指令

JZ rel ；（A）＝0，转移

JNZ rel ；（A）≠0，转移

JC rel ；如 C=1, 转移

JNC rel ；如 C=0, 转移

JB bit，rel ；如 bit=1, 转移

JNB bit，rel ；如 bit=0, 转移

JBC bit，rel ；如 bit=1, 转移并 bit＝0

比较不相等转移指令

CJNE A，#data，rel；(A)=#data，继续 C←0 (A)＞#data，转 C←0 (A)＜#data，转 C←1

特点：只有＜时， C←1

CJNE A，direct，rel
CJNE Rn，#data，rel
CJNE @Ri，#data，rel

减 1 不为 0 转移指令
DJNZ Rn，rel ；
DJNZ direct，rel

例：延时子程序

```
delay： MOV R7, #03H
delay0： MOV R6, #19H
delay1： DJNZ R6, delay1
DJNZ R7, delay0
RET
```

调用子程序指令
短调用指令 ACALL addr11
长调用指令 LCALL addr16
子程序返回指令 RET
中断返回指令 RETI
空操作指令 NOP

第四章 定时器/计数器

第一节 概述

第二节 结构和工作原理

实质是计数器，脉冲每一次下降沿，计数寄存器数值将加 1。
计数的脉冲如果来源于单片机内部的晶振,由于其周期极为准确，这时称为定时器。
计数的脉冲如果来源于单片机外部的引脚，由于其周期一般不准确，这时称为计数器。

定时/计数器方式寄存器 TMOD
(1) M1 和 M0：方式选择位。
(2) c/T ：功能选择位。时，设置为定时器工作方式；时，设置为计数器工作方式。
(3) GATE：门控位。当 GATE=0 时，软件控制位 TR0 或 TR1 置 1 即可启动定时器；当

GATE=1 时，软件控制位 TR0 或 TR1 须置 1，同时还须（P3.2）或（P3.3）为高电平方可启动定时器，即允许外中断、启动定时器。

定时器/计数器控制寄存器 TCON

(1) TCON.7　TF1：定时器 1 溢出标志位。当定时器 1 计满数产生溢出时，由硬件自动置 TF1=1。在中断允许时，向 CPU 发出定时器 1 的中断请求，进入中断服务程序后，由硬件自动清 0。在中断屏蔽时，TF1 可作查询测试用，此时只能由软件清 0。

(2) TCON.6　TR1：定时器 1 运行控制位。由软件置 1 或清 0 来启动或关闭定时器 1。当 GATE=1，且为高电平时，TR1 置 1 启动定时器 1；当 GATE=0 时，TR1 置 1 即可启动定时器 1。

(3) TCON.5　TF0：定时器 0 溢出标志位。其功能及操作情况同 TF1。

(4) TCON.4　TR0：定时器 0 运行控制位。其功能及操作情况同 TR1。

(5) TCON.3　IE1：外部中断 1（　　）请求标志位。

(6)TCON.2　IT1：外部中断 1 触发方式选择位。

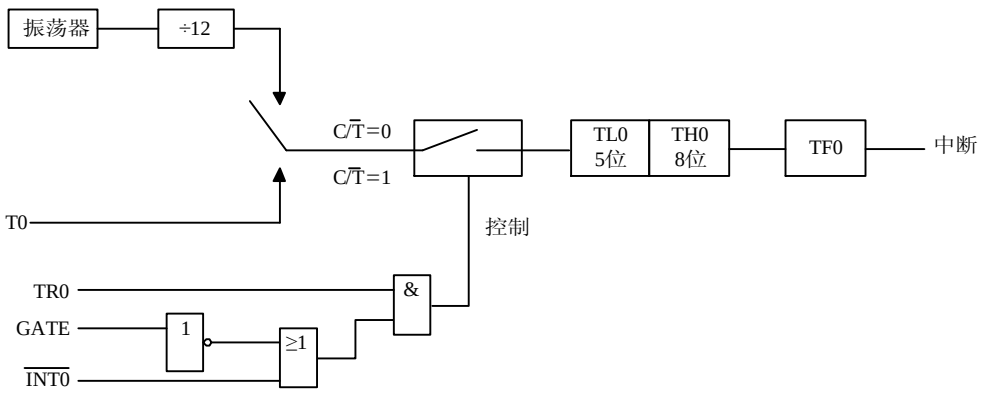
(7)TCON.1　IE0：外部中断 0　（　　）请求标志位。

(8) TCON.0　IT0：外部中断 0 触发方式选择位。

第三节　定时/计数器的工作方式

1．方式 0

方式 0 构成一个 13 位定时/计数器。图是定时器 0 在方式 0 时的逻辑电路结构，定时器 1 的结构和操作与定时器 0 完全相同。



2．方式 1

定时器工作于方式 1 时。

由图可知，方式 1 构成一个 16 位定时/计数器，其结构与操作几乎完全与方式 0 相同，惟一差别是二者计数位数不同。

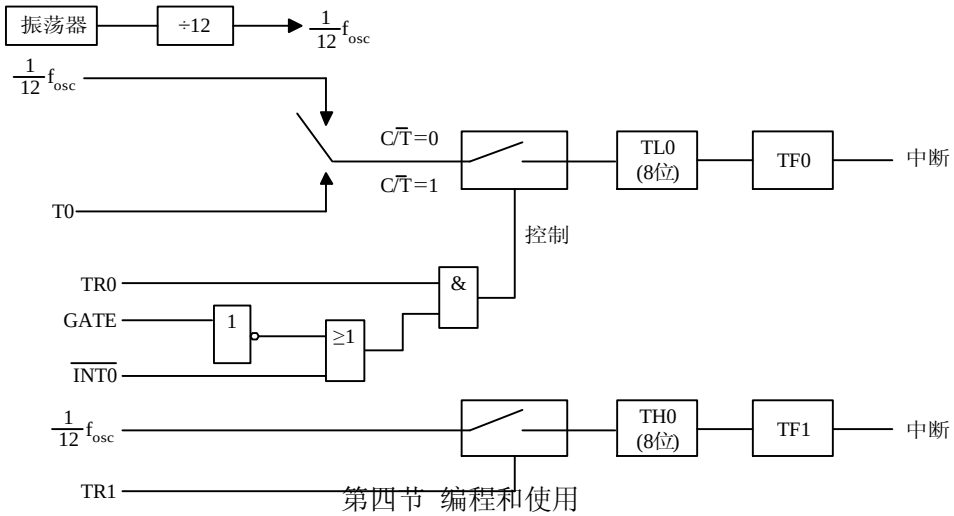
3．方式 2

定时/计数器工作于方式 2 时，。

由图可知，方式 2 中，16 位加法计数器的 TH0 和 TL0 具有不同功能，其中，TL0 是 8 位计数器，TH0 是重置初值的 8 位缓冲器。

4．方式 3

定时/计数器工作于方式 3 时，其逻辑结构图如图所示。



第四节 编程和使用

1．计数器初值的计算

把计数器计满为零所需要的计数值设定为 C，计数初值设定为 TC，由此可得到公式：TC=M-C 式中，M 为计数器模值，该值和计数器工作方式有关。在方式 0 时 M 为 213；在方式 1 时 M 为 216；在方式 2 和方式 3 时 M 为 28。

2．定时器初值的计算

在定时器模式下，计数器由单片机主脉冲经 12 分频后计数。因此，定时器定时时间 T 的公式：T=（M-TC）T 计数，上式也可写成：TC=M-T/ T 计数。式中，M 为模值，和定时器的工作方式有关；T 计数是单片机振荡周期 TCLK 的 12 倍；TC 为定时器的定时初值。

第五章 串行通讯基础

第一节 串行通信基础

在计算机系统中，CPU 和外部通信有两种通信方式：并行通信和串行通信。并行通信，即数据的各位同时传送；串行通信，即数据一位一位顺序传送。

串行通信的分类

按照串行数据的时钟控制方式，串行通信可分为同步通信和异步通信两类。

1. 异步通信（Asynchronous Communication）

在异步通信中，数据通常是以字符为单位组成字符帧传送的。字符帧由发送端一帧一帧地发送，每一帧数据均是低位在前，高位在后，通过传输线被接收端一帧一帧地接收。发送端和接收端可以由各自独立的时钟来控制数据的发送和接收，这两个时钟彼此独立，互不同步。

在异步通信中，接收端是依靠字符帧格式来判断发送端是何时开始发送，何时结束发送的。字符帧格式是异步通信的一个重要指标。

字符帧（Character Frame）

字符帧也叫数据帧，由起始位、数据位、奇偶校验位和停止位等 4 部分组成，

波特率（baud rate）

异步通信的另一个重要指标为波特率。

波特率为每秒钟传送二进制数码的位数，也叫比特数，单位为 b/s，即位/秒。波特率用于表征数据传输的速度，波特率越高，数据传输速度越快。但波特率和字符的实际传输速率不同，字符的实际传输速率是每秒内所传字符帧的帧数，和字符帧格式有关。

2. 同步通信（Synchronous Communication）

同步通信是一种连续串行传送数据的通信方式，一次通信只传输一帧信息。这里的信息帧和异步通信的字符帧不同，通常有若干个数据字符，如图 8.4 所示。图 8.4(a)为单同步字符帧结构，图 8.4(b)为双同步字符帧结构，但它们均由同步字符、数据字符和校验字符 CRC 三部分组成。在同步通信中，同步字符可以采用统一的标准格式，也可以由用户约定。

第二节 串行通讯口的工作方式

1.数据缓冲器 SBUF

发送 SBUF 和接收 SBUF 共用一个地址 99H 。

1）发送 SBUF 存放待发送的 8 位数据，写入 SBUF 将同时启动发送。

发送指令：MOV SBUF，A

2）接收 SBUF 存放已接收成功的 8 位数据，供 CPU 读取。

读取串行口接收数据指令：MOV A，SBUF

2.串行口控制/状态寄存器 SCON(98H)

SM0，SM1：选择串行口 4 种工作方式。

SM2：多机控制位，用于多机通讯。
REN：允许接收控制位，REN=1，允许接收；REN=0，禁止接收。
TB8 发送的第 9 位数据位，可用作校验位和地址/数据标识位
RB8：接收的第 9 位数据位或停止位
TI：发送中断标志，发送一帧结束，TI=1，必须软件清零
RI：接收中断标志，接收一帧结束，RI=1，必须软件清零

3.节电控制寄存器 PCON
SMOD(PCON.7)：波特率加倍控制位。
SMOD=1，波特率加倍， SMOD=0，则不加倍。

串行接口的工作方式
SM0，SM1 选择四种工作方式。
(1) 方式 0：同步移位寄存器方式。用于扩展并行 I/O 接口。
1.一帧 8 位，无起始位和停止位。
2.RXD：数据输入/输出端。TXD：同步脉冲输出端，每个脉冲对应一个数据位。
3.波特率 B = fosc/12 如： fosc=12MHz， B=1MHz，每位数据占 1μs。
4.发送过程：写入 SBUF，启动发送，一帧发送结束，TI=1。接收过程：REN=1 且 RI=0，启动接收，一帧接收完毕，RI=1。

(2) 方式 1：8 位数据异步通讯方式。
1.一帧 10 位：8 位数据位，1 个起始位(0)，1 个停止位(1)。
2.RXD：接收数据端。 TXD：发送数据端。
3.波特率：用 T1 作为波特率发生器，B=(2SMOD/32)×T1 溢出率。
4.发送：写入 SBUF，同时启动发送，一帧发送结束，TI=1。接收：REN=1，允许接收。接收完一帧，若 RI=0 且停止位为 1 (或 SM2=0)，将接收数据装入 SBUF，停止位装入 RB8，并使 RI=1；否则丢弃接收数据，不置位 RI。

(3) 方式 2 和方式 3：9 位数据异步通讯方式。
1.一帧为 11 位：9 位数据位，1 个起始位(0)，1 个停止位(1)。 第 9 位数据位在 TB8/RB8 中，常用作校验位和多机通讯标识位。
2.RXD：接收数据端，TXD：发送数据端。
3.波特率： 方式 2：B=(2SMOD/64)×fosc 。
方式 3：B=(2SMOD/32)×T1 溢出率 。
4.发送：先装入 TB8，写入 SBUF 并启动发送，发送结束，TI=1。接收：REN=1，允许接收。接收完一帧，若 RI=0 且第 9 位为 1 (或 SM2=0)，将接收数据装入接收 SBUF，第 9 位装入 RB8，使 RI=1；否则丢弃接收数据，不置位 RI。

第三节 波特率的设置方法

方式 0 为固定波特率：B=fosc/12
方式 2 可选两种波特率： B=(2SMOD /64)×fosc
方式 1、3 为可变波特率，用 T1 作波特率发生器。
B=(2SMOD/32)×T1 溢出率
T1 为方式 2 的时间常数： X = 28 - t/T
溢出时间： t= (28 -X)T = (28 -X)×12/ fosc
T1 溢出率=1/t= fosc /[12×(2n -X)]
波特率 B=(2SMOD /32)×fosc/[12×(28-X)]
串行口方式 1、3，根据波特率选择 T1 工作方式，计算时间常数。
T1 选方式 2： TH1=X= 28-fosc/12×2SMOD/(32×B)
T1 选方式 1 用于低波特率，需考虑 T1 重装时间常数时间。

第六章 中断系统

第一节 概述

中断是通过硬件来改变 CPU 的运行方向的。计算机在执行程序的过程中，当出现 CPU 以外的某种情况时，由服务对象向 CPU 发出中断请求信号，要求 CPU 暂时中断当前程序的执行而转去执行相应的处理程序，待处理程序执行完毕后，再继续执行原来被中断的程序。这种程序在执行过程中由于外界的原因而被中间打断的情况称为“中断”。

与中断有关的寄存器有 4 个，分别为中断源寄存器 TCON 和 SCON、中断允许控制寄存器 IE 和中断优先级控制寄存器 IP；中断源有 5 个，分别为外部中断 0 请求 、外部中断 1 请求、定时器 0 溢出中断请求 TF0、定时器 1 溢出中断请求 TF1 和串行中断请求 RI 或 TI。5 个中断源的排列顺序由中断优先级控制寄存器 IP 和顺序查询逻辑电路共同决定，5 个中断源分别对应 5 个固定的中断入口地址。

第二节 中断源与中断申请标志

中断源
(1)外部中断 0 请求，由 P3.2 脚输入。通过 IT0 脚（TCON.0）来决定是低电平有效还是下跳变有效。一旦输入信号有效，就向 CPU 申请中断，并建立 IE0 标志。
(2)外部中断 1 请求，由 P3.3 脚输入。通过 IT1 脚 TCON.2）来决定是低电平有效还是下跳变有效。一旦输入信号有效，就向 CPU 申请中断，并建立 IE1 标志。

(3)TF0：定时器 T0 溢出中断请求。当定时器 0 产生溢出时，定时器 0 中断请求标志位（TCON.5）置位（由硬件自动执行），请求中断处理。
(4)TF1：定时器 1 溢出中断请求。当定时器 1 产生溢出时，定时器 1 中断请求标志位（TCON.7）置位（由硬件自动执行），请求中断处理。
(5) RI 或 TI：串行中断请求。当接收或发送完一串行帧时，内部串行口中断请求标志位 RI（SCON.0）或 TI（SCON.1）置位（由硬件自动执行），请求中断。

中断标志
TCON 寄存器中的中断标志
TCON 为定时器 0 和定时器 1 的控制寄存器，同时也锁存定时器 0 和定时器 1 的溢出中断标志及外部中断和的中断标志等。与中断有关位如下：
(1) TCON.7 TF1：定时器 1 的溢出中断标志。T1 被启动计数后，从初值做加 1 计数，计满溢出后由硬件置位 TF1，同时向 CPU 发出中断请求，此标志一直保持到 CPU 响应中断后才由硬件自动清 0。也可由软件查询该标志，并由软件清 0。
(2) TCON.5 TF0：定时器 0 溢出中断标志。其操作功能与 TF1 相同。
(3) TCON.3 IE1：中断标志。IE1 = 1，外部中断 1 向 CPU 申请中断。
(4) TCON.2 IT1：中断触发方式控制位。当 IT1 = 0 时，外部中断 1 控制为电平触发方式。
(5) TCON.1 IE0：中断标志。其操作功能与 IE1 相同。
(6) TCON.0 IT0：中断触发方式控制位。其操作功能与 IT1 相同。

SCON 寄存器中的中断标志
SCON 是串行口控制寄存器，其低两位 TI 和 RI 锁存串行口的发送中断标志和接收中断标志。
(1) SCON.1 TI：串行发送中断标志。CPU 将数据写入发送缓冲器 SBUF 时，就启动发送，每发送完一个串行帧，硬件将使 TI 置位。但 CPU 响应中断时并不清除 TI，必须由软件清除。

第三节 中断控制

IE 寄存器中断的开放和禁止标志
(1) IE.7 EA：总中断允许控制位。EA = 1，开放所有中断，各中断源的允许和禁止可通过相应的中断允许位单独加以控制；EA = 0，禁止所有中断。
(2) IE.4 ES：串行口中断允许位。ES = 1，允许串行口中断；ES = 0，禁止串行口中断。
(3) IE.3 ET1：定时器 1 中断允许位。ET1 = 1，允许定时器 1 中断；ET1 = 0，禁止定时器 1 中断。
(4) IE.2 EX1：外部中断 1（）中断允许位。EX1 = 1，允许外部中断 1 中断；EX1 = 0，

禁止外部中断 1 中断。
(5) IE.1 ET0：定时器 0 中断允许位。ET0 = 1，允许定时器 0 中断；ET0 = 0，禁止定时器 0 中断。
(6) IE.0 EX0: 外部中断 0（）中断允许位。EX0 = 1，允许外部中断 0 中断；EX0 = 0，禁止外部中断 0 中断。
8051 单片机系统复位后，IE 中各中断允许位均被清 0，即禁止所有中断。

IP 寄存器中断优先级标志
8051 单片机有两个中断优先级，每个中断源都可以通过编程确定为高优先级中断或低
（1）IP.4 PS：串行口中断优先控制位。PS = 1，设定串行口为高优先级中断；PS = 0，设定串行口为低优先级中断。
（2）IP.3 PT1：定时器 T1 中断优先控制位。PT1 = 1，设定定时器 T1 中断为高优先级中断；PT1 = 0，设定定时器 T1 中断为低优先级中断。
（3）IP.2 PX1：外部中断 1 中断优先控制位。PX1 = 1，设定外部中断 1 为高优先级中断；PX1 = 0，设定外部中断 1 为低优先级中断。
（4）IP.1 PT0：定时器 T0 中断优先控制位。PT0 = 1，设定定时器 T0 中断为高优先级中断；PT0 = 0，设定定时器 T0 中断为低优先级中断。
（5）IP.0 PX0：外部中断 0 中断优先控制位。PX0 = 1，设定外部中断 0 为高优先级中断；PX0 = 0，设定外部中断 0 为低优先级中断。
当系统复位后，IP 低 5 位全部清 0，所有中断源均设定为低优先级中断。
如果几个同一优先级的中断源同时向 CPU 申请中断，CPU 通过内部硬件查询逻辑，按自然优先级顺序确定先响应哪个中断请求。自然优先级由硬件形成，排列如下：

中断源	同级自然优先级
外部中断 0	最高级
定时器 T0 中断	
外部中断 1	
定时器 T1 中断	
串行口中断	最低级

第四节 中断响应

中断处理过程可分为中断响应、中断处理和中断返回三个阶段。

中断响应
中断响应是 CPU 对中断源中断请求的响应，包括保护断点和将程序转向中断服务程序的入口地址（通常称矢量地址）。

中断响应过程

中断响应过程包括保护断点和将程序转向中断服务程序的入口地址。首先，中断系统通过硬件自动生成长调用指令（LACLL），该指令将自动把断点地址压入堆栈保护（不保护累加器 A、状态寄存器 PSW 和其它寄存器的内容），然后，将对应的中断入口地址装入程序计数器 PC（由硬件自动执行），使程序转向该中断入口地址，执行中断服务程序。MCS-51 系列单片机各中断源的入口地址由硬件事先设定，分配如下：

中断源	入口地址
外部中断 0	0003H
定时器 T0 中断	000BH
外部中断 1	0013H
定时器 T1 中断	001BH
串行口中断	0023H

使用时，通常在这些中断入口地址处存放一条绝对跳转指令，使程序跳转到用户安排的中断服务程序的起始地址上去。

中断返回

中断返回是指中断服务完后，计算机返回原来断开的位置（即断点），继续执行原来的程序。中断返回由中断返回指令 RETI 来实现。该指令的功能是把断点地址从堆栈中弹出，送回到程序计数器 PC，此外，还通知中断系统已完成中断处理，并同时清除优先级状态触发器。特别要注意不能用“RET”指令代替“RETI”指令。

中断请求的撤除

CPU 响应中断请求后即进入中断服务程序，在中断返回前，应撤除该中断请求，否则，会重复引起中断而导致错误。MCS-51 各中断源中断请求撤消的方法各不相同，分别为：

1）定时器中断请求的撤除

对于定时器 0 或 1 溢出中断，CPU 在响应中断后即由硬件自动清除其中断标志位 TF0 或 TF1，无需采取其它措施。

2）串行口中断请求的撤除

对于串行口中断，CPU 在响应中断后，硬件不能自动清除中断请求标志位 TI、RI，必须在中断服务程序中用软件将其清除。

3）外部中断请求的撤除

外部中断可分为边沿触发型和电平触发型。

对于边沿触发的外部中断 0 或 1，CPU 在响应中断后由硬件自动清除其中断标志位 IE0 或 IE1，无需采取其它措施。

第六章 MCS_51 单片机扩展存储器的设计

第一节 概述

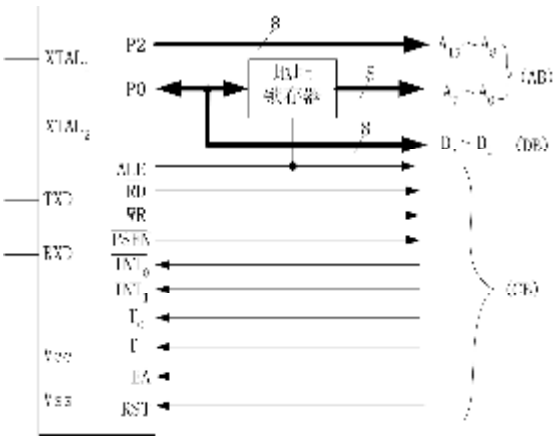
对于没有内部 ROM 的单片机或者当程序较长、片内 ROM 容量不够时，用户必须在单片机外部扩展程序存储器。MCS-51 单片机片外有 16 条地址线，即 P0 口和 P2 口，因此最大寻址范围为 64 KB（0000H～FFFFH）。

第二节 系统总线及总线构造

MCS-51 系列单片机片外引脚可以构成如图所示的三总线结构：

地址总线（AB）
数据总线（DB）
控制总线（CB）

所有外部芯片都通过这三组总线进行扩展。



第三节 程序存储器 EPROM 的扩展

扩展程序存储器常用的芯片是 EPROM（Erasable Programmable Read Only Memory）型（紫外线可擦除型）， 如 2716（2K×8）、2732（4K×8）、2764（8K×8）、27128（16K×8）、27256（32K×8）、27512（64K×8）等。另外，还有+5 V 电可擦除 EEPROM，如 2816（2K×8）、2864（8K×8）等等。

紫外线擦除电可编程只读存储器 EPROM 是国内用得较多的程序存储器。EPROM 芯片上有一个玻璃窗口，在紫外线照射下，存储器中的各位信息均变 1，即处于擦除状态。擦除干净的 EPROM 可以通过编程器将应用程序固化到芯片中。

如果程序总量不超过 4 KB，一般选用具有内部 ROM 的单片机。8051 内部 ROM 只能由厂家将程序一次性固化，不适合小批量用户和程序调试时使用，因此选用 8751、8951 的用户较多。如果程序超过 4 KB，用户一般不会选用 8751、8951，而是直接选用 8031，利用外部扩展存储器来存放程序。

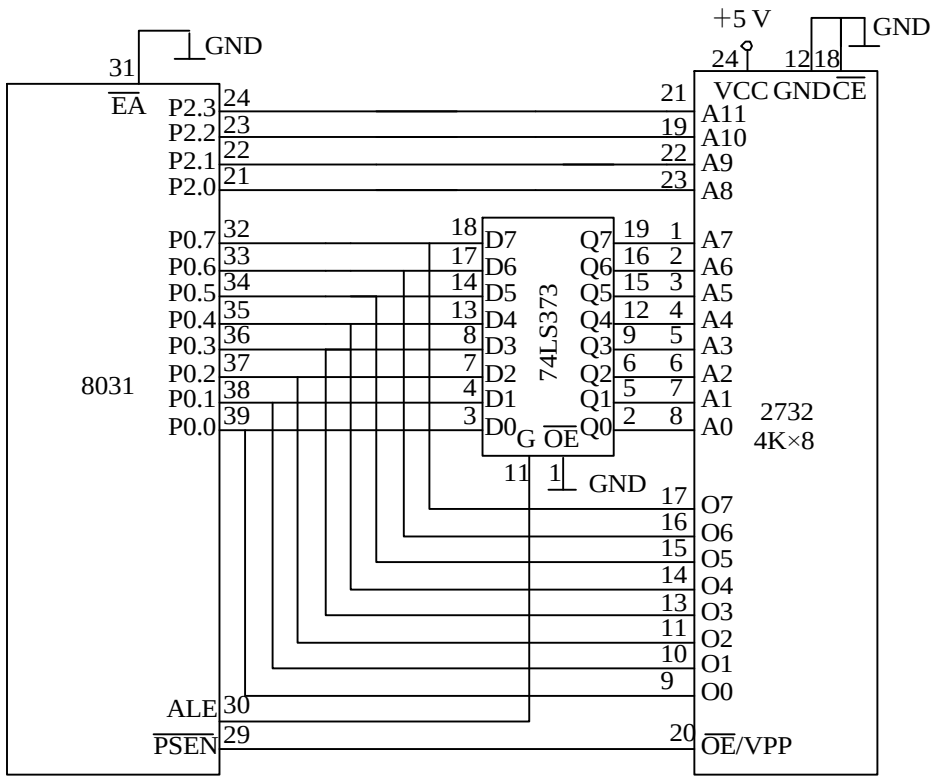
实例：在 8031 单片机上扩展 4 KB EPROM 程序存储器。

选择芯片

本例要求选用 8031 单片机，内部无 ROM 区，无论程序长短都必须扩展程序存储器（目前较少这样使用，但扩展方法比较典型、实用）。在选择程序存储器芯片时，首先必须满足程序容量，其次在价格合理情况下尽量选用容量大的芯片。这样做的话，使用的芯片少，从而接线简单，芯片存储容量大，程序调整余量也大。如估计程序总长 3 KB 左右，最好是扩展一片 4 KB 的 EPROM 2732，而不是选用 2 片 2716（2 KB）。在单片机应用系统硬件设计中应注意，尽量减少芯片使用个数，使得电路结构简单，提高可靠性，这也是 8951 比 8031 使用更加广泛的原因之一。

硬件电路图

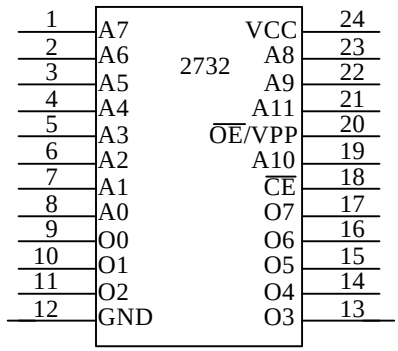
8031 单片机扩展一片 2732 程序存储器电路如图所示。



芯片说明

①74LS373。74LS373 是带三态缓冲输出的 8D 锁存器，由于片机的三总线结构中，数据线与地址线的低 8 位共用 P0 口，因此必须用地址锁存器将地址信号和数据信号区分开。74LS373 的锁存控制端 G 直接与单片机的锁存控制信号 ALE 相连，在 ALE 的下降沿锁存低 8 位地址。

②EPROM 2732。EPROM 2732 的容量为 4 K×8 位。4 K 表示有 4×1024（$2^2\times2^{10}=2^{12}$）个存储单元，8 位表示每个单元存储数据的宽度是 8 位。前者确定了地址线的位数是 12 位（A0～A11），后者确定了数据线的位数是 8 位（O0～O7）。目前，除了串行存储器之外，一般情况下，我们使用的都是 8 位数据存储器。2732 采用单一+5 V 供电，最大静态工作电流为 100 mA，维持电流为 35 mA，读出时间最大为 250 ns。2732 的封装形式为 DIP24，管脚如图所示。



其中，A0～A11 为地址线；O0～O7 为数据线； 为片选线；OE/VPP 为输出允许/编程高压。 除了 12 条地址线和 8 条数据线之外，CE 为片选线，低电平有效。也就是说，只有当 CE 为低电平时，2732 才被选中，否则，2732 不工作。OE/VPP 为双功能管脚，当 2732 用作程序存储器时，其功能是允许读数据出来；当对 EPROM 编程（也称为固化程序）时，该管脚用于高电压输入，不同生产厂家的芯片编程电压也有所不同。当我们把它作为程序存储器使用时，不必关心其编程电压。

连线说明：

① 地址线。单片机扩展片外存储器时，地址是由 P0 和 P2 口提供的。图 6.2 中，2732 的 12 条地址线（A0～A11）中，低 8 位 A0～A7 通过锁存器 74LS373 与 P0 口连接，高 4 位 A8～A11 直接与 P2 口的 P2.0～P2.3 连接，P2 口本身有锁存功能。注意，锁存器的锁存使能端 G 必须和单片机的 ALE 管脚相连。

② 数据线。2732 的 8 位数据线直接与单片机的 P0 口相连。因此，P0 口是一个分时复用的地址/数据线。

③ 控制线。CPU 执行 2732 中存放的程序指令时，取指阶段就是对 2732 进行读操作。注意，CPU 对 EPROM 只能进行读操作，不能进行写操作。CPU 对 2732 的读操作控制都是通过控制线实现的。2732 控制线的连接有以下几条：

CE ：直接接地。由于系统中只扩展了一个程序存储器芯片，因此，2732 的片选端直接接地，表示 2732 一直被选中。若同时扩展多片，需通过译码器来完成片选工作。

OE ：接 8031 的读选通信号端。在访问片外程序存储器时，只要端出现负脉冲，即可从 2732 中读出程序。

扩展程序存储器地址范围的确定

单片机扩展存储器的关键是搞清楚扩展芯片的地址范围，8031 最大可以扩展 64 KB（0000H～FFFFH）。决定存储器芯片地址范围的因素有两个：一个是片选端的连接方法，一个是存储器芯片的地址线与单片机地址线的连接。在确定地址范围时，必须保证片选端为低电平。

本例中，2732 的地址范围为 0000H~0FFFH：

EPROM 的使用

存储器扩展电路是单片机应用系统的功能扩展部分，只有当应用系统的软件设计完成了，才能把程序通过特定的编程工具（一般称为编程器或 EPROM 固化器）固化到 2732 中，然后再将 2732 插到用户板的插座上（扩展程序存储器一定要焊插座）。

第四节 数据存储器 RAM 的扩展

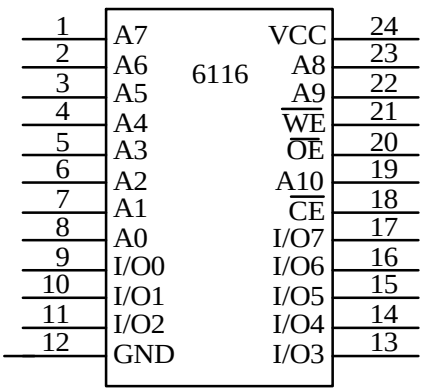
RAM 是用来存放各种数据的，MCS-51 系列 8 位单片机内部有 128 B RAM 存储器，CPU 对内部 RAM 具有丰富的操作指令。但是，当单片机用于实时数据采集或处理大批量数据时，仅靠片内提供的 RAM 是远远不够的。此时，我们可以利用单片机的扩展功能，扩展外部数据存储器。

常用的外部数据存储器有静态 RAM（Static Random Access Memory—SRAM）和动态 RAM（Dynamic Random Access Memory—DRAM）两种。前者读/写速度高，一般都是 8 位宽度，易于扩展，且大多数与相同容量的 EPROM 引脚兼容，有利于印刷板电路设计，使用方便；缺点是集成度低，成本高，功耗大。后者集成度高，成本低，功耗相对较低；缺点是需要增加一个刷新电路，附加另外的成本。

MCS-51 单片机扩展片外数据存储器的地址线也是由 P0 口和 P2 口提供的，因此最大寻址范围为 64 KB（0000H～FFFFH）。

一般情况下，SRAM 用于仅需要小于 64 KB 数据存储器的小系统，DRAM 经常用于需要大于 64 KB 的大系统。

实例：在一单片机应用系统中扩展 2 KB 静态 RAM。



芯片选择

单片机扩展数据存储器常用的静态 RAM 芯片有 6116（2K×8 位）、6264（8K×8 位）、62256（32K×8 位）等。根据题目容量的要求，我们选用 SRAM 6116。

6116 的管脚与 EPROM 2716 管脚兼容，管脚如图所示

硬件电路

单片机与 6116 的硬件连接如图所示。

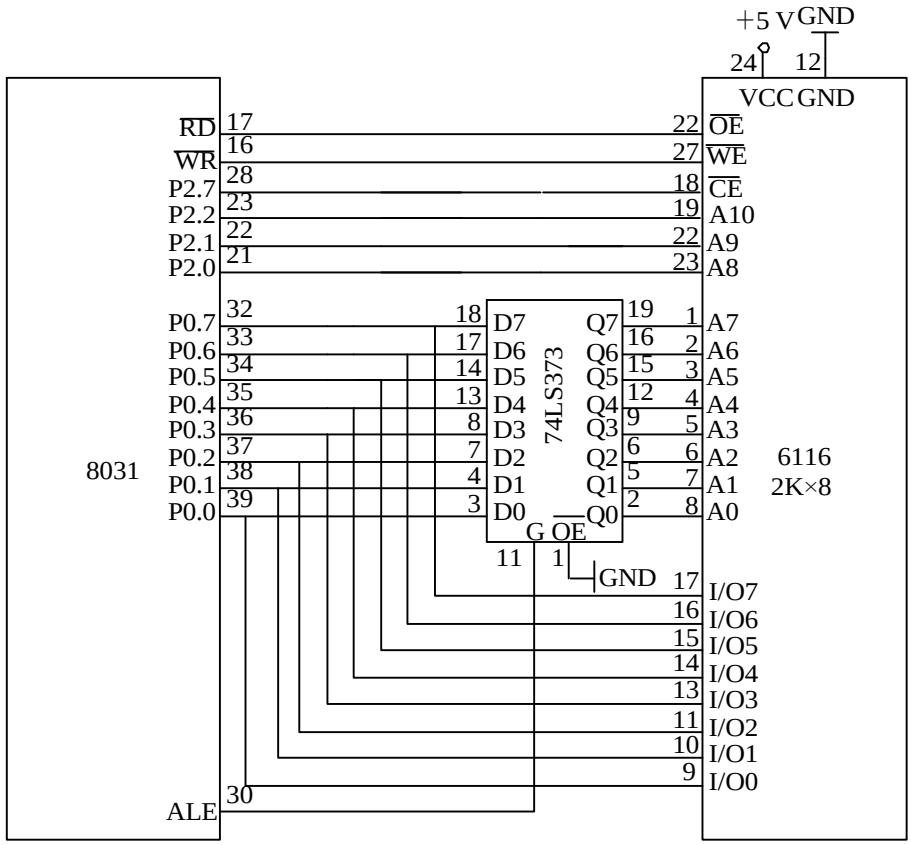
连线说明

6116 与单片机的连线如下：

地址线：A0～A10 连接单片机地址总线的 A0～A10，即 P0.0～P0.7、P2.0、P2.1、P2.2 共 11 根。

数据线：I/O0～I/O7 连接单片机的数据线，即 P0.0～P0.7。

控制线：CE 片选端连接单片机的 P2.7，即单片机地址总线的最高位 A15；OE 读允许线连接单片机的读数据存储器控制线 RD；写允许线 WE 连接单片机的写数据存储器控制线 WR。



片外 RAM 地址范围的确定及使用

按照上图的连线，片选端直接与某一地址线 P2.7 相连，这种扩展方法称为线选法。显然，只有 P2.7=0，才能够选中该片 6116，故其地址范围确定如下：

如果与 6116 无关的管脚取 0，那么，6116 的地址范围是 0000H～07FFH；如果与 6116 无关的管脚取 1，那么，6116 的地址范围是 7800H～7FFFH。

单片机对 RAM 的读写除了可以如下指令：

MOVX @DPTR,A ；64 KB 内写入数据

MOVX A,@DPTR ；64 KB 内读取数据

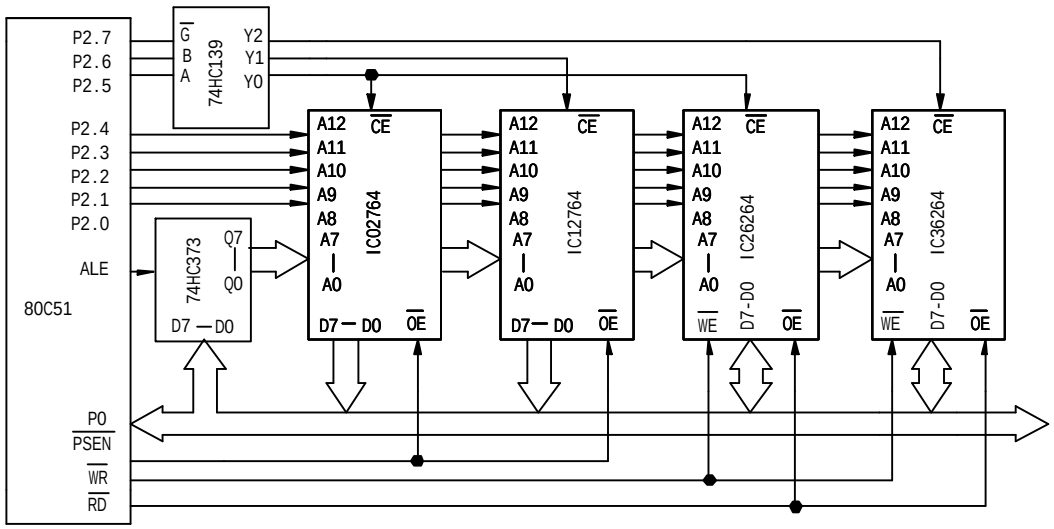
外，还可以使用以下对低 256 B 的读写指令：

MOVX @Ri,A ；低 256 B 内写入数据

MOVXA，@Ri ；低 256 B 内读取数据

第五节 综合扩展

综合扩展实例



第七章 MCS_51 扩展 I/O 接口的设计

第一节 概述

51 系列单片机内部有 4 个双向的并行 I/O 端口：P0～P3，共占 32 根引脚。P0 口的每一位可以驱动 8 个 TTL 负载，P1～P3 口的负载能力为三个 TTL 负载。有关 4 个端口的结构及详细说明，在前面的有关章节中已作过介绍，这里不再赘述。

在无片外存储器扩展的系统中，这 4 个端口都可以作为准双向通用 I/O 口使用。我们知道，在具有片外扩展存储器的系统中， P0 口分时地作为低 8 位地址线和数据线，P2 口作为高 8 位地址线。这时，P0 口和部分或全部的 P2 口无法再作通用 I/O 口。

P3 口具有第二功能，在应用系统中也常被使用。因此在大多数的应用系统中，真正能够提供给用户使用的只有 P1 和部分 P2、P3 口。

综上所述，MCS-51 单片机的 I/O 端口通常需要扩充，以便和更多的外设（例如显示器、键盘）进行联系。

在 51 单片机中扩展的 I/O 口采用与片外数据存储器相同的寻址方法,所有扩展的 I/O 口，以及通过扩展 I/O 口连接的外设都与片外 RAM 统一编址，因此，对片外 I/O 口的输入/输出指令就是访问片外 RAM 的指令。

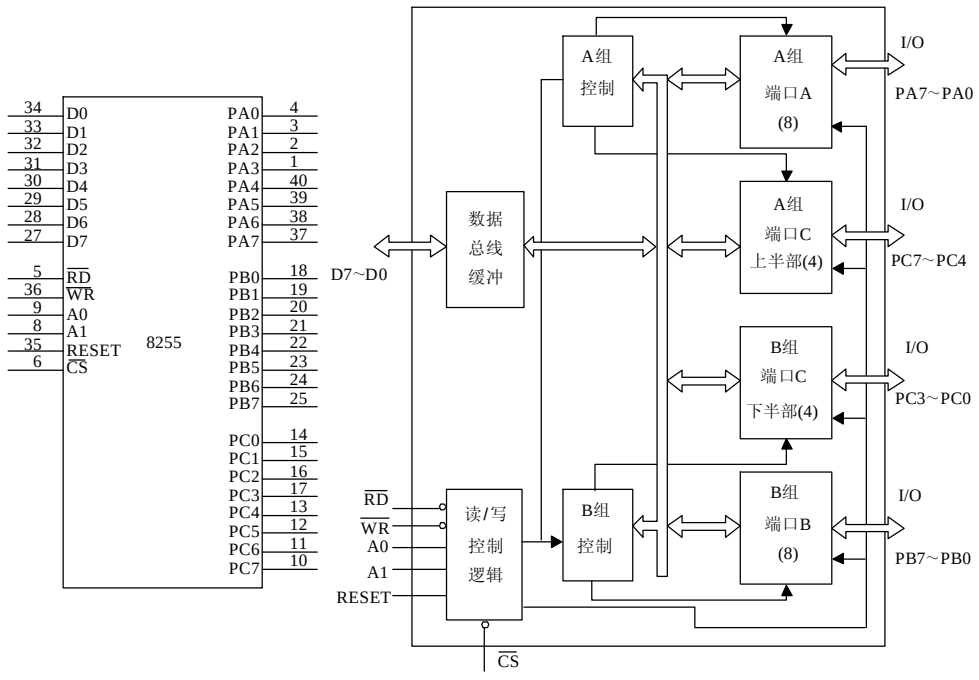
第二节 MCS_51 与 8255A 的接口设计

所谓可编程的接口芯片是指其功能可由微处理机的指令来加以改变的接口芯片，利用编程的方法，可以使一个接口芯片执行不同的接口功能。目前，各生产厂家已提供了很多系列的可编程接口，MCS-51 单片机常用的两种接口芯片是 8255 以及 8155。

8255 和 MCS-51 相连，可以为外设提供三个 8 位的 I/O 端口：A 口、B 口和 C 口，三个端口的功能完全由编程来决定。

1. 8255 的内部结构和引脚排列

下图 7 为 8255 的内部结构和引脚图。



(1)A 口、B 口和 C 口。A 口、B 口和 C 口均为 8 位 I/O 数据口，但结构上略有差别。A 口由一个 8 位的数据输出缓冲/锁存器和一个 8 位的数据输入缓冲/锁存器组成。B 口由

一个 8 位的数据输出缓冲/锁存器和一个 8 位的数据输入缓冲器组成。三个端口都可以和外设相连，分别传送外设的输入/输出数据或控制信息。

(2) A、B 组控制电路。这是两组根据 CPU 的命令字控制 8255 工作方式的电路。A 组控制 A 口及 C 口的高 4 位，B 组控制 B 口及 C 口的低 4 位。

(3) 数据缓冲器。这是一个双向三态 8 位的驱动口，用于和单片机的数据总线相连，传送数据或控制信息。

(4) 读/写控制逻辑。这部分电路接收 MCS-51 送来的读/写命令和选口地址，用于控制对 8255 的读/写。

(5) 数据线(8 条)：D0～D7 为数据总线，用于传送 CPU 和 8255 之间的数据、命令和状态字。

(6) 控制线和寻址线(6 条)。

RESET：复位信号，输入高电平有效。一般和单片机的复位相连，复位后，8255 所有内部寄存器清 0，所有口都为输入方式。

WR 和 RD： 读/写信号线，输入，低电平有效。当为 0 时（必为 1），所选的 8255 处于读状态，8255 送出信息到 CPU。反之亦然。

(6)CS：片选线，输入，低电平有效。

(7) A0、A1：地址输入线。当=0，芯片被选中时，这两位的 4 种组合 00、01、10、11 分别用于选择 A、B、C 口和控制寄存器。

(8)I/O 口线(24 条)：PA0～PA7、PB0～PB7、PC0～PC7 为 24 条双向三态 I/O 总线，分别与 A、B、C 口相对应，用于 8255 和外设之间传送数据。

(9) 电源线(2 条)：VCC 为+5 V，GND 为地线。

2. 8255 的控制字

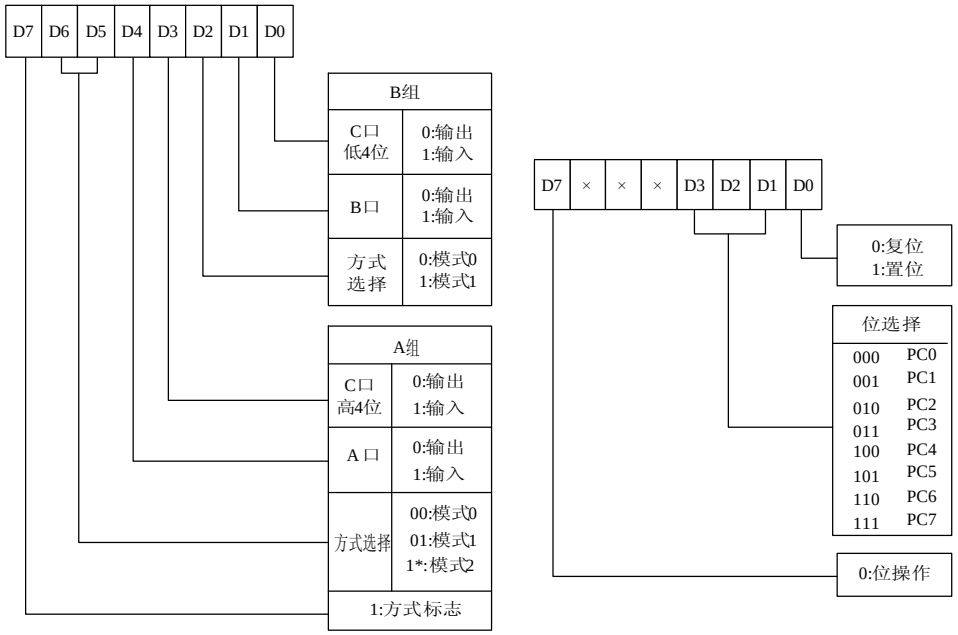
8255 的三个端口具体工作在什么方式下，是通过 CPU 对控制口的写入控制字来决定的。8255 有两个控制字：方式选择控制字和 C 口置/复位控制字。用户通过程序把这两个控制字送到 8255 的控制寄存器（A0A1=11），这两个控制字以 D7 来作为标志。

1) 方式选择控制字

方式选择控制字的格式和定义如下图所示。

2) C 口置/复位控制字

C 口置/复位控制字的格式和定义如下图所示。C 口具有位操作功能，把一个置/复位控制字送入 8255 的控制寄存器，就能将 C 口的某一位置 1 或清 0 而不影响其它位的状态。



3. 8255 的工作方式

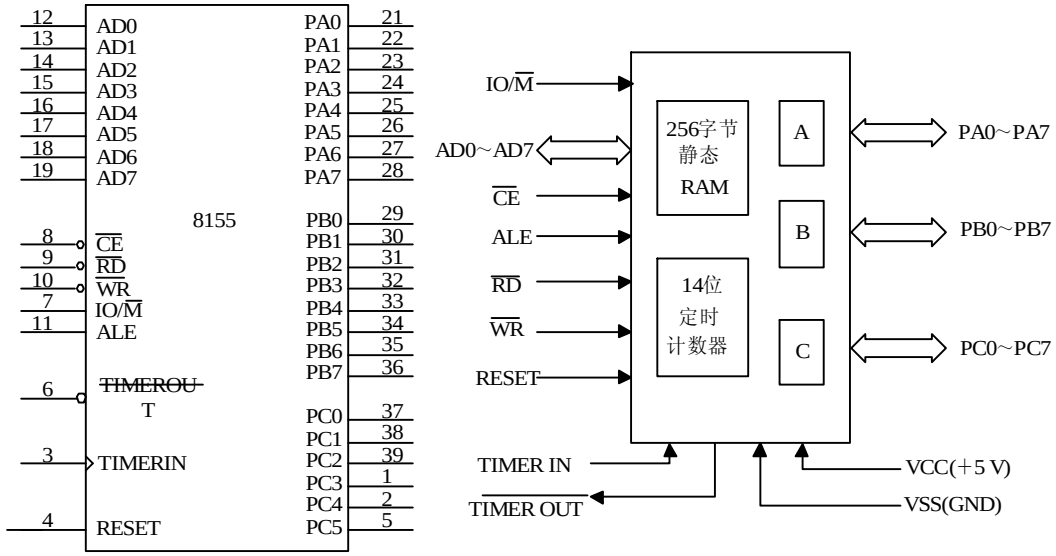
8255 有三种工作方式：方式 0、方式 1、方式 2。方式的选择是通过上述写控制字的方法来完成的。

第三节 MCS_51 与 8155H 的接口设计

另一种可编程的接口芯片 8155，Intel 公司研制的 8155 不仅具有两个 8 位的 I/O 端口(A 口、B 口)和一个 6 位的 I/O 端口(C 口)，而且还可以提供 256 B 的静态 RAM 存储器和一个 14 位的定时/计数器。8155 和单片机的接口非常简单，目前被广泛应用。

1. 8155 的结构和引脚

8155 有 40 个引脚，采用双列直插封装，其引脚图和组成框图如下图所示。



我们对 8155 的引脚分类说明如下：

(1) 地址/数据线 AD0～AD7（8 条）：是低 8 位地址线和数据线的共用输入总线，常和 51 单片机的 P0 口相连，用于分时传送地址数据信息，当 ALE=1 时，传送的是地址。

(2) I/O 口总线（22 条）：PA0～PA7、PB0～PB7 分别为 A、B 口线，用于和外设之间传递数据；PC0～PC5 为 C 端口线，既可与外设传送数据，也可以作为 A、B 口的控制联络线。

(3) 控制总线（8 条）：

RESET：复位线，通常与单片机的复位端相连，复位后，8155 的 3 个端口都为输入方式。

WR, RD：读/写线，控制 8155 的读、写操作。

ALE：地址锁存线，高电平有效。它常和单片机的 ALE 端相连，在 ALE 的下降沿将单片机 P0 口输出的低 8 位地址信息锁存到 8155 内部的地址锁存器中。因此，单片机的 P0 口和 8155 连接时，无需外接锁存器。

CS：片选线，低电平有效。

IO/M：RAM 或 I/O 口的选择线。当=0 时，选中 8155 的 256 B RAM；当=1 时，选中 8155 片内 3 个 I/O 端口以及命令/状态寄存器和定时/计数器。

TIMERIN、TIMEROUT：定时/计数器的脉冲输入、输出线。TIMERIN 是脉冲输入线，其输入脉冲对 8155 内部的 14 位定时/计数器减 1；为输出线，当计数器计满回 0 时，8155 从该线输出脉冲或方波，波形形状由计数器的工作方式决定。

2. 作片外 RAM 使用

当 CE=0，IO/M=0 时，8155 只能做片外 RAM 使用，共 256 B。其寻址范围由以及 AD0～AD7 的接法决定，这和前面讲到的片外 RAM 扩展时讨论的完全相同。当系统同时扩展片外 RAM 芯片时，要注意二者的统一编址。对这 256 B RAM 的操作使用片外 RAM 的读/写指令“MOVX”。

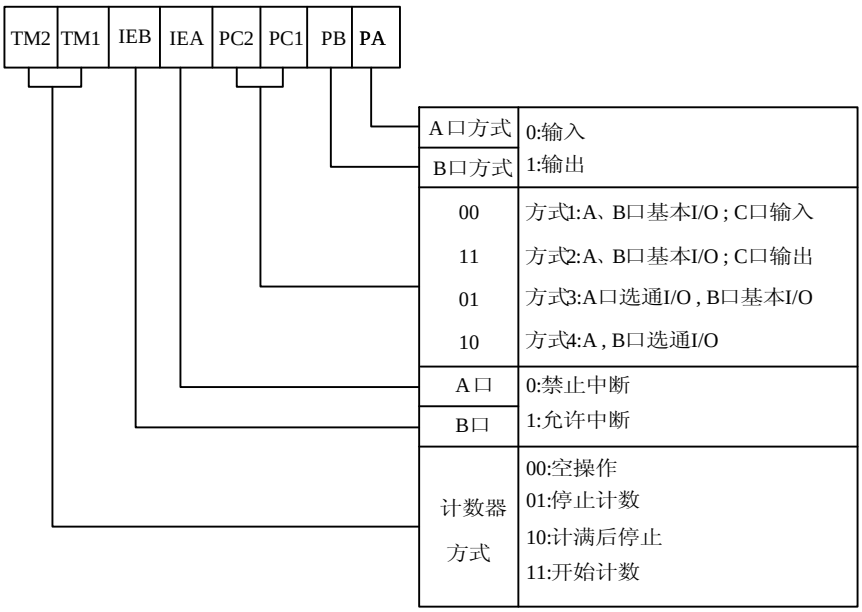
4. 作扩展 I/O 口使用

当 CE=0，IO/M=1 时，此时可以对 8155 片内 3 个 I/O 端口以及命令/状态寄存器和定时/计数器进行操作。与 I/O 端口和计数器使用有关的内部寄存器共有 6 个，需要三位地址来区分。

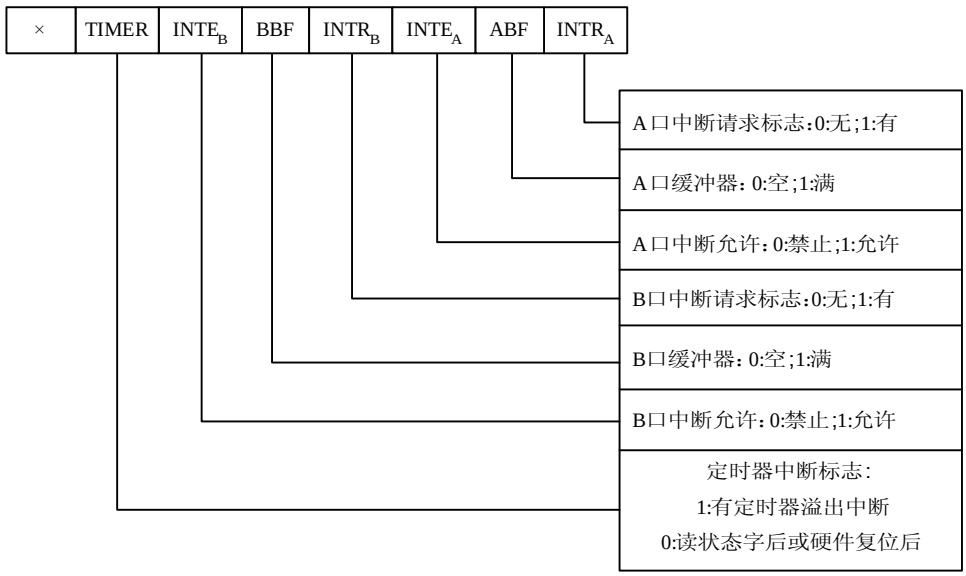
1) 命令/状态寄存器

和接口芯片 8255 一样，芯片 8155 I/O 口的工作方式的确定也是通过对 8155 的命令寄存器写入控制字来实现的。8155 控制字的格式如下图所示。

命令寄存器只能写入不能读出，也就是说，控制字只能通过指令 MOVX @DPTR, A 或 MOVX @Ri, A 写入命令寄存器。



状态寄存器中存放有状态字，状态字反映了 8155 的工作情况，状态字的各位定义如下图所示。



状态寄存器和命令寄存器是同一地址，状态寄存器只能读出不能写入，也就是说，状态字只能通过指令 MOVX A ，@DPTR 或 MOVX A，@Ri 来读出，以此来了解 8155 的工作状态。

2) 计数器高、低 8 位寄存器

关于计数器高、低 8 位寄存器的使用，我们将在后面讲到定时器使用时再作介绍。

4. I/O 口的工作方式

当使用 8155 的三个 I/O 端口时，它们可以工作于不同的方式，工作方式的选择取决于写入的控制字，如图 6.21 所示。其中，A、B 口可以工作于基本 I/O 方式或选通 I/O 方式，C 口可工作于基本 I/O 方式，也可以作为 A、B 选通方式时的控制联络线。

5. 定时/计数器使用

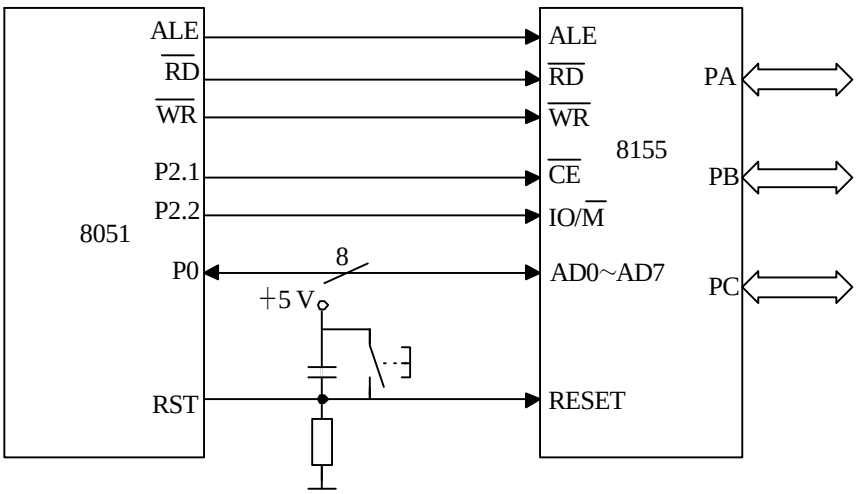
8155 的可编程定时/计数器是一个 14 位的减法计数器，在 TIMERIN 端输入计数脉冲，计满时由 TIMEROUT 输出脉冲或方波，输出方式由定时器高 8 位寄存器中的 M2、M1 两位来决定。当 TIMERIN 接外脉冲时为计数方式，接系统时钟时为定时方式，实际使用时一定要注意芯片允许的最高计数频率！

定时/计数器的初始值和输出方式由高、低 8 位寄存器的内容决定，初始值 14 位，其

余两位定义输出方式。

6. MCS-51 单片机和 8155 的接口

MCS-51 和 8155 的接口非常简单，因为 8155 内部有一个 8 位地址锁存器，故无需外接锁存器。在二者的连接中，8155 的地址译码即片选端可以采用线选法、全译码等方法，这和 8255 类似。在整个单片机应用系统中要考虑与片外 RAM 及其它接口芯片的统一编址。



此时，8155 内部 RAM 的地址范围为：0000H～00FFH，8155 各端口的地址（设无关位为 0，这些地址都不是惟一的）为：

命令/状态口 0400H
A 口 0401H
B 口 0402H
C 口 0403H
定时器低字节 0404H
定时器高字节 0405H

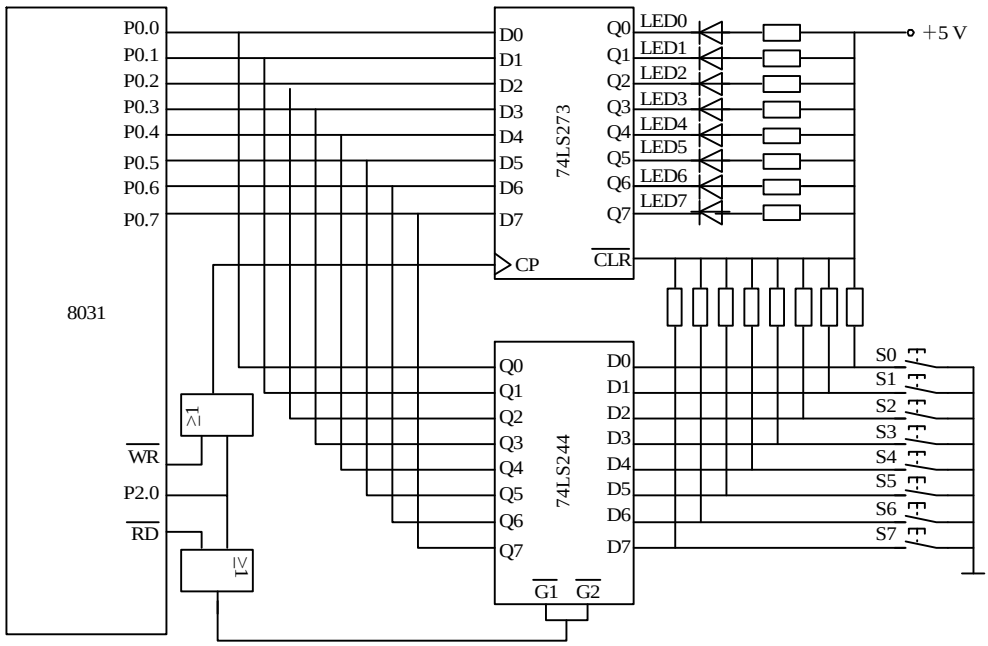
第四节 简单的 I/O 口扩展

扩展实例

简单的 I/O 口扩展通常是采用 TTL 或 CMOS 电路锁存器、三态门等作为扩展芯片，通过 P0 口来实现扩展的一种方案。它具有电路简单、成本低、配置灵活的特点。下图为采用 74LS244 作为扩展输入、74LS273 作为扩展输出的简单 I/O 口扩展。

芯片及连线说明

图中电路中采用的芯片为 TTL 电路 74LS244、74LS273。其中，74LS244 为 8 缓冲线驱动器(三态输出),、为低电平有效的使能端。当二者之一为高电平时，输出为三态。74LS273 为 8D 触发器，为低电平有效的清除端。当=0 时，输出全为 0 且与其它输入端无关；CP 端是时钟信号，当 CP 由低电平向高电平跳变时刻，D 端输入数据传送到 Q 输出端。P0 口作为双向 8 位数据线，既能够从 74LS244 输入数据，又能够从 74LS273 输出数据。输入控制信号由 P2.0 和相“或”后形成。当二者都为 0 时，74LS244 的控制端有效，选通



74LS244，外部的信息输入到 P0 数据总线上。当与 74LS244 相连的按键都没有按下时，输入全为 1，若按下某键，则所在线输入为 0。输出控制信号由 P2.0 和相“或”后形成。当二者都为 0 后，74LS273 的控制端有效，选通 74LS273，P0 上的数据锁存到 74LS273 的输出端，控制发光二极管 LED，当某线输出为 0 时，相应的 LED 发光。

I/O 口地址确定

因为 74LS244 和 74LS273 都是在 P2.0 为 0 时被选通的，所以二者的口地址都为 FEFFH（这个地址不是惟一的，只要保证 P2.0=0，其它地址位无关）。但是由于分别由和

控制，因而两个信号不可能同时为 0（执行输入指令，如 MOVX A，@DPTR 或 MOVX A，@Ri 时，有效；执行输出指令，如 MOVX @DPTR，A 或 MOVX @Ri，A 时，有效），所以逻辑上二者不会发生冲突。

第八章 MCS_51 与键盘、显示器的接口设计

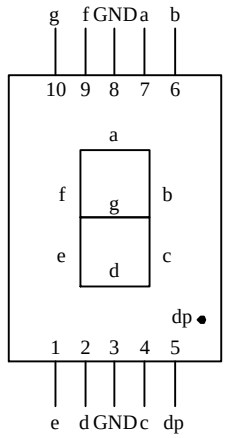
第一节 LED 接口原理

常用的 LED 显示器有 LED 状态显示器（俗称发光二极管）、LED 七段显示器（俗称数码管）和 LED 十六段显示器。发光二极管可显示两种状态，用于系统状态显示；数码管用于数字显示；LED 十六段显示器用于字符显示。

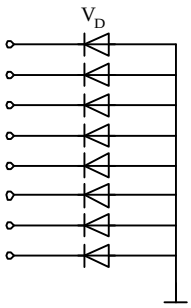
1. 数码管简介

1) *数码管结构*

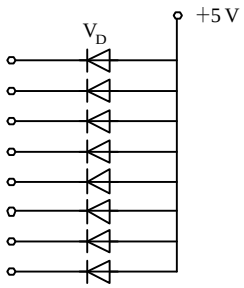
数码管由 8 个发光二极管（以下简称字段）构成，通过不同的组合可用来显示数字 0 ~9、字符 A ~ F、H、L、P、R、U、Y、符号“-”及小数点“.”。数码管的外形结构如下图所示。数码管又分为共阴极和共阳极两种结构。



(a)



(b)



(c)

2) *数码管工作原理*

共阳极数码管的 8 个发光二极管的阳极（二极管正端）连接在一起。通常，公共阳极接高电平（一般接电源），其它管脚接段驱动电路输出端。当某段驱动电路的输出端为

低电平时，则该端所连接的字段导通并点亮。根据发光字段的不同组合可显示出各种数字或字符。此时，要求段驱动电路能吸收额定的段导通电流，还需根据外接电源及额定段导通电流来确定相应的限流电阻。

共阴极数码管的 8 个发光二极管的阴极（二极管负端）连接在一起。通常，公共阴极接低电平（一般接地），其它管脚接段驱动电路输出端。当某段驱动电路的输出端为高电平时，则该端所连接的字段导通并点亮，根据发光字段的不同组合可显示出各种数字或字符。此时，要求段驱动电路能提供额定的段导通电流，还需根据外接电源及额定段导通电流来确定相应的限流电阻。

3) *数码管字形编码*

要使数码管显示出相应的数字或字符，必须使段数据口输出相应的字形编码。对照图 7.10（a），字型码各位定义为：数据线 D0 与 a 字段对应，D1 与 b 字段对应……，依此类推。如使用共阳极数码管，数据为 0 表示对应字段亮，数据为 1 表示对应字段暗；如使用共阴极数码管，数据为 0 表示对应字段暗，数据为 1 表示对应字段亮。如要显示“0”，共阳极数码管的字型编码应为：11000000B（即 C0H）；共阴极数码管的字型编码应为：00111111B（即 3FH）。依此类推。

2．静态显示接口

静态显示是指数码管显示某一字符时，相应的发光二极管恒定导通或恒定截止。这种显示方式的各位数码管相互独立，公共端恒定接地（共阴极）或接正电源（共阳极）。每个数码管的 8 个字段分别与一个 8 位 I/O 口地址相连，I/O 口只要有段码输出，相应字符即显示出来，并保持不变，直到 I/O 口输出新的段码。采用静态显示方式，较小的电流即可获得较高的亮度，且占用 CPU 时间少，编程简单，显示便于监测和控制，但其占用的口线多，硬件电路复杂，成本高，只适合于显示位数较少的场合。

3．动态显示接口

动态显示是一位一位地轮流点亮各位数码管，这种逐位点亮显示器的方式称为位扫描。通常，各位数码管的段选线相应并联在一起，由一个 8 位的 I/O 口控制；各位的位选线（公共阴极或阳极）由另外的 I/O 口线控制。动态方式显示时，各数码管分时轮流选通，要使其稳定显示，必须采用扫描方式，即在某一时刻只选通一位数码管，并送出相应的段码，在另一时刻选通另一位数码管，并送出相应的段码。依此规律循环，即可使各位数码管显示将要显示的字符。虽然这些字符是在不同的时刻分别显示，但由于人眼存在视觉暂留效应，只要每位显示间隔足够短就可以给人以同时显示的感觉。

采用动态显示方式比较节省 I/O 口，硬件电路也较静态显示方式简单，但其亮度不如静态显示方式，而且在显示位数较多时，CPU 要依次扫描，占用 CPU 较多的时间。

第二节 键盘接口原理

1．键的分类

按键按照结构原理可分为两类，一类是触点式开关按键，如机械式开关、导电橡胶式开关等；另一类是无触点式开关按键，如电气式按键，磁感应按键等。前者造价低，后者寿命长。目前，微机系统中最常见的是触点式开关按键。

2．输入原理

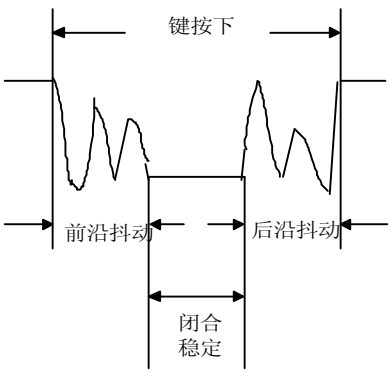
在单片机应用系统中，除了复位按键有专门的复位电路及专一的复位功能外，其它按键都是以开关状态来设置控制功能或输入数据的。当所设置的功能键或数字键按下时，计算机应用系统应完成该按键所设定的功能，键信息输入是与软件结构密切相关的过程。

对于一组键或一个键盘，总有一个接口电路与 CPU 相连。CPU 可以采用查询或中断方式了解有无将键输入，并检查是哪一个键按下，将该键号送入累加器 ACC，然后通过跳转指令转入执行该键的功能程序，执行完后再返回主程序

3．按键结构与特点

微机键盘通常使用机械触点式按键开关，其主要功能是把机械上的通断转换成为电气上的逻辑关系。也就是说，它能提供标准的 TTL 逻辑电平，以便与通用数字系统的逻辑电平相容。

机械式按键再按下或释放时，由于机械弹性作用的影响，通常伴随有一定时间的触点机械抖动，然后其触点才稳定下来。其抖动过程如下图所示，抖动时间的长短与开关的机械特性有关，一般为 5~10 ms。



在触点抖动期间检测按键的通与断状态，可能导致判断出错，即按键一次按下或释放被错误地认为是多次操作，这种情况是不允许出现的。

为了克服按键触点机械抖动所致的检测误判，必须采取去抖动措施。这一点可从硬件、软件两方面予以考虑。在键数较少时，可采用硬件去抖，而当键数较多时，采用软件去抖。

4. 按键编码

一组按键或键盘都要通过 I/O 口线查询按键的开关状态。根据键盘结构的不同，采用不同的编码。无论有无编码，以及采用什么编码，最后都要转换成为与累加器中数值相对应的键值，以实现按键功能程序的跳转。

5． 制键盘程序

一个完善的键盘控制程序应具备以下功能：

(1) 检测有无按键按下，并采取硬件或软件措施，消除键盘按键机械触点抖动的影响。

(2) 有可靠的逻辑处理办法。每次只处理一个按键，其间对任何按键的操作对系统不产生影响，且无论一次按键时间有多长，系统仅执行一次按键功能程序。

(3) 准确输出按键值（或键号），以满足跳转指令要求。

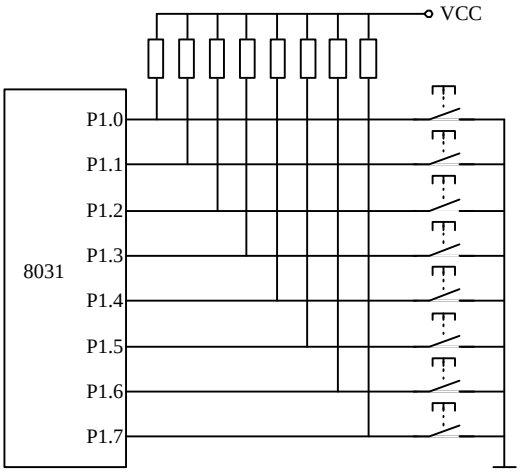
独立式按键

单片机控制系统中，往往只需要几个功能键，此时，可采用独立式按键结构。

1．独立式按键结构

独立式按键是直接用 I/O 口线构成的单个按键电路，其特点是每个按键单独占用一根 I/O 口线，每个按键的工作不会影响其它 I/O 口线的状态。独立式按键的典型应用如图 7.4 所示。

独立式按键电路配置灵活，软件结构简单，但每个按键必须占用一根 I/O 口线，因此，在按键较多时，I/O 口线浪费较大，不宜采用。



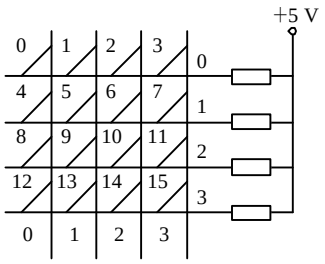
2．立式按键的软件结构

独立式按键的软件常采用查询式结构。先逐位查询每根 I/O 口线的输入状态，如某一根 I/O 口线输入为低电平，则可确认该 I/O 口线所对应的按键已按下，然后，再转向该键的功能处理程序。

矩阵式按键

单片机系统中，若使用按键较多时，通常采用矩阵式（也称行列式）键盘。

1. 矩阵式键盘的结构及原理



矩阵式键盘由行线和列线组成，按键位于行、列线的交叉点上，其结构如下图所示。

由图可知，一个 4×4 的行、列结构可以构成一个含有 16 个按键的键盘，显然，在按键数量较多时，矩阵式键盘较之独立式按键键盘要节省很多 I/O 口。

矩阵式键盘中，行、列线分别连接到按键开关的两端，行线通过上拉电阻接到＋5V 上。当无键按下时，行线处于高电平状态；当有键按下时，行、列线将导通，此时，行线电平将由与此行线相连的列线电平决定。这是识别按键是否按下的关键。然而，矩阵键盘中的行线、列线和多个键相连，各按键按下与否均影响该键所在行线和列线的电平，各按键间将相互影响，因此，必须将行线、列线信号配合起来作适当处理，才能确定闭合键的位置。

2．式键盘按键的识别

识别按键的方法很多，其中，最常见的方法是扫描法。

按键按下时，与此键相连的行线与列线导通，行线在无键按下时处在高电平。显然，如果让所有的列线也处在高电平，那么，按键按下与否不会引起行线电平的变化，因此，必须使所有列线处在低电平。只有这样，当有键按下时，该键所在的行电平才会由高电平变为低电平。CPU 根据行电平的变化，便能判定相应的行有键按下。

3．盘的编码

对于独立式按键键盘，因按键数量少，可根据实际需要灵活编码。对于矩阵式键盘，按键的位置由行号和列号惟一确定，因此可分别对行号和列号进行二进制编码，然后将两值合成一个字节，高 4 位是行号，低 4 位是列号。

4. 键盘的工作方式

对键盘的响应取决于键盘的工作方式，键盘的工作方式应根据实际应用系统中 CPU 的工作状况而定，其选取的原则是既要保证 CPU 能及时响应按键操作，又不要过多占用 CPU 的工作时间。通常，键盘的工作方式有三种，即编程扫描、定时扫描和中断扫描。

1) 编程扫描方式

编程扫描方式是利用 CPU 完成其它工作的空余时间，调用键盘扫描子程序来响应键盘输入的要求。在执行键功能程序时，CPU 不再响应键输入要求，直到 CPU 重新扫描键盘为止。

2) 定时扫描方式

定时扫描方式就是每隔一段时间对键盘扫描一次，它利用单片机内部的定时器产生一定时间（例如 10 ms）的定时，当定时时间到就产生定时器溢出中断。CPU 响应中断后对键盘进行扫描，并在有键按下时识别出该键，再执行该键的功能程序。

3) 中断扫描方式

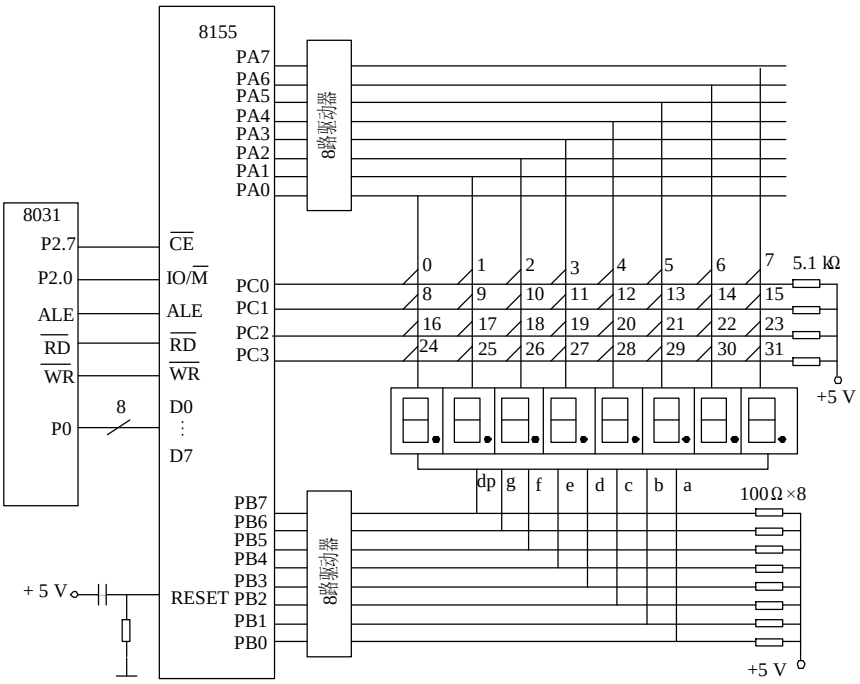
采用上述两种键盘扫描方式时，无论是否按键，CPU 都要定时扫描键盘，而单片机应用系统工作时，并非经常需要键盘输入，因此，CPU 经常处于空扫描状态。

为提高 CPU 工作效率，可采用中断扫描工作方式。其工作过程如下：当无键按下时，CPU 处理自己的工作，当有键按下时，产生中断请求，CPU 转去执行键盘扫描子程序，并识别键号。

第三节 典型的键盘、显示接口电路

在单片机应用系统中，键盘和显示器往往需同时使用，为节省 I/O 口线，可将键盘和显示电路做在一起，构成实用的键盘、显示电路。下图是用 8155 并行扩展 I/O 口构成的典型的键盘、显示接口电路。键盘、显示器共用一个接口电路的设计方法除上述方案外，

还可采用专用的键盘、显示器接口的芯片——8279。

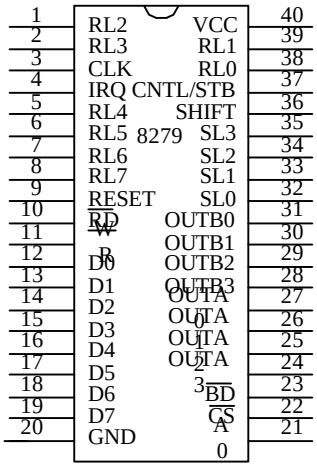


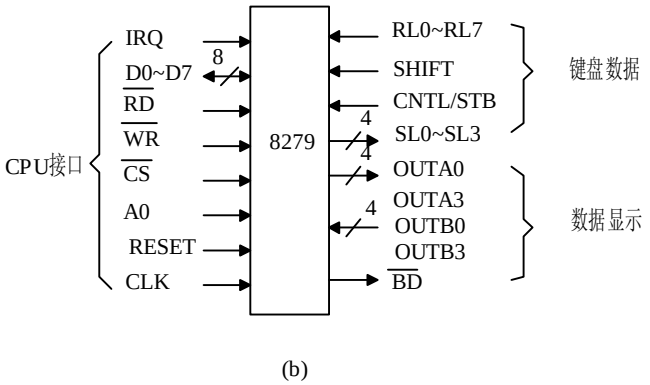
第四节可编程键盘/显示器接口——INTEL 8279

INTEL 8279 是一种可编程键盘/显示器接口芯片，它含有键盘输入和显示器输出两种功能。键盘输入时，它提供自动扫描，能与按键或传感器组成的矩阵相连，接收输入信息，它能自动消除开关抖动并能对多键同时按下提供保护。显示输出时，它有一个 16×8 位显示 RAM，其内容通过自动扫描，可由 8 或 16 位 LED 数码管显示。

8279 的引脚和功能

8279 的内部结构框图如下图所示。





(b)

D0～D7：数据总线，双向三态总线。

CLK：系统时钟输入端。

RESET：系统复位输入端，高电平有效。复位状态为：16 个字符显示；编码扫描键盘：

双键锁定；程序时钟编程为 31。

CS：片选输入端，低电平有效。

A0：数据选择输入端。A0=1 时，CPU 写入数据为命令字，读出状态字为状态字；A0=0 时，CPU 读、写均为数据。

WR、RE：读、写信号输入端，低电平有效。

IRQ：中断请求输出端，高电平有效。

SL0～SL3：扫描输出端，用于扫描键盘和显示器。可编程设定为编码（4 中选 1）或译码输出（16 选 1）。

RL0～RL7：回复线，它们是键盘或传感器的列信号输入端。

SHIFT：移位信号输入端，高电平有效。它是 8279 键盘数据的次高位（D6），通常用作键盘上、下档功能键。在传感器和选通方式中，SHIFT 无效。

CNTL/STB：控制/选通输入端，高电平有效。在键盘工作方式时，它是键盘数据的最高位，通常用作控制键。在选通输入方式时，它的上升沿可把来自 RL0~RL7 的数据存入 FIFO/传感器 RAM 中。在传感器方式时，它无效。

OUTA0～OUTA3：A 组显示信号输出端。

OUTB0～OUTB3：B 组显示信号输出端。

BD：显示熄灭输出端，低电平有效。它在数字切换显示或使用熄灭命令时关显示。

8279 的工作方式

8279 工作方式的确定是通过 CPU 对 8279 送入命令字实现的。当数据选择端 A0 置“1”时，CPU 对 8279 写入的数据为命令字，读出的数据为状态字。在叙述命令字、状态字前，先说明 8279 的几种工作方式。

1）键盘的工作方式

通过对键盘/显示方式命令字的设置，可置为双键互锁方式和 N 键巡回方式。

双键互锁

双键锁定是为两键同时按下提供的保护方法。若有两键或多个键同时按下，则无论这些键是以什么次序按下的，它只识别最后一个释放的键，并把该键值送入 FIFO/传感器 RAM 中。

N 键巡回

N 键巡回是为 N 个键同时按下时提供的保护方法。若有多个键同时按下时，键盘扫描能按按键先后顺序依次将键值送入 FIFO/传感器 RAM 中。

2) 显示器工作方式

通过对键盘/显示方式命令字和写显示 RAM 命令字的设置，显示数据写入显示缓冲器时可置为左端送入和右端送入两种方式。左端送入为依次填入方式，右端送入为移位方式。

8279 的命令格式和命令字

8279 共有 8 条命令字和一条状态字，分别控制其工作方式和工作状态。

第九章 MCS_51 与 A/D，D/A 接口设计

第一节 D/A 接口

D/A 转换器输入的是数字量，经转换后输出的是模拟量。有关 D/A 转换器的技术性能指标很多，例如绝对精度、相对精度、线性度、输出电压范围、温度系数、输入数字代码种类（二进制或 BCD 码）等。

1）**分辨率**

分辨率是 D/A 转换器对输入量变化敏感程度的描述，与输入数字量的位数有关。如果数字量的位数为 n，则 D/A 转换器的分辨率为 2^{-n}。这就意味着数/模转换器能对满刻度的 2^{-n} 输入量作出反应。

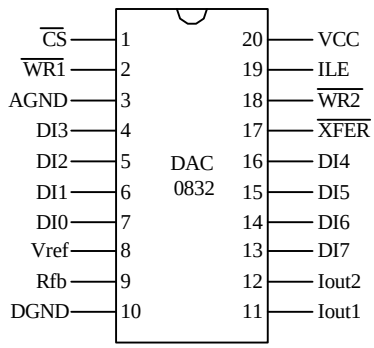
2）**建立时间**

建立时间是描述 D/A 转换速度快慢的一个参数，指从输入数字量变化到输出达到终值误差±（1/2）LSB（最低有效位）时所需的时间。通常以建立时间来表示转换速度. 转换器的输出形式为电流时，建立时间较短；输出形式为电压时，由于建立时间还要加上运算放大器的延迟时间，因此建立时间要长一点。但总的来说，D/A 转换速度远高于 A/D 转换速度，快速的 D/A 转换器的建立时间可达 1 μs。

3）**接口形式**

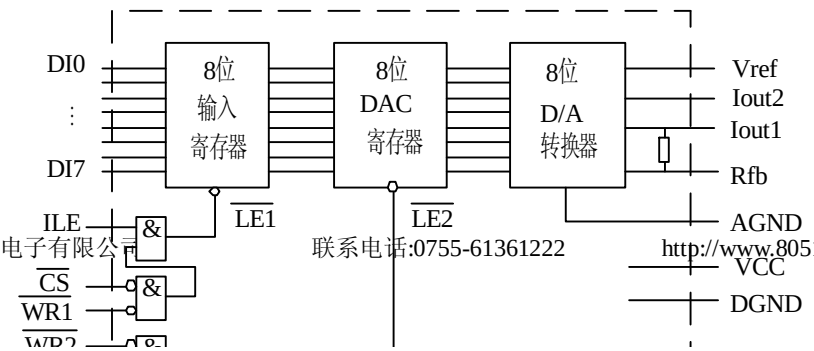
D/A 转换器与单片机接口方便与否，主要决定于转换器本身是否带数据锁存器。有两类 D/A 转换器，一类是不带锁存器的，另一类是带锁存器的。对于不带锁存器的 D/A 转换器，为了保存来自单片机的转换数据，接口时要另加锁存器，因此这类转换器必须在口线上；而带锁存器的 D/A 转换器，可以把它看作是一个输出口，因此可直接在数据总线上，而不需另加锁存器。

典型 D/A 转换器芯片 DAC0832



DAC0832 是一个 8 位 D/A 转换器。单电源供电，从+5 V～+15 V 均可正常工作。基准电压的范围为±10 V；电流建立时间为 1 μs；CMOS 工艺，低功耗 20 mW。

DAC0832 转换器芯片为 20 引脚，双列直插式封装，其引脚排列图如图所示。DAC0832 内部结构框图如图所示。 该转换器由输入寄存器和 DAC 寄存器构成两级数据输入锁存。使用时，数据输入可以采用两级锁存（双锁存）形式，或单级锁存（一级锁存，一级直通）形式，或直接输入（两 级直通）形式。



此外，由三个与门电路组成寄存器输出控制逻辑电路，该逻辑电路的功能是进行数据锁存控制，当=0 时，输入数据被锁存；当=1 时，锁存器的输出跟随输入的数据。

D/A 转换电路是一个 R-2R T 型电阻网络，实现 8 位数据的转换。对各引脚信号说明如下：

(1) DI7～DI0：转换数据输入。

(2) CS：片选信号（输入），低电平有效。

(3) ILE：数据锁存允许信号（输入），高电平有效。

(4) WR：第 1 写信号（输入），低电平有效。

上述两个信号控制输入寄存器是数据直通方式还是数据锁存方式，当 ILE=1 和 WR1 =1=0 时，为输入寄存器直通方式；当 ILE=1 和 WR1 =1 时，为输入寄存器锁存方式。

(5) WR2 =1：第 2 写信号（输入），低电平有效。

(6) XFER：数据传送控制信号（输入），低电平有效。

上述两个信号控制 DAC 寄存器是数据直通方式还是数据锁存方式，当 WR2=0 和 XFER=0 时，为 DAC 寄存器直通方式；当 WR2=1 和 XFER=0 时，为 DAC 寄存器锁存方式。

(7) Iout1：电流输出 1。

(8) Iout2：电流输出 2。

DAC 转换器的特性之一是：Iout1+Iout2=常数。

(9) Rfb：反馈电阻端。

DAC 0832 是电流输出，为了取得电压输出，需在电压输出端接运算放大器，Rfb 即为运算放大器的反馈电阻端。运算放大器的接法如图 7.31 所示。

(10) Vref：基准电压，其电压可正可负，范围是-10 V～+10 V。

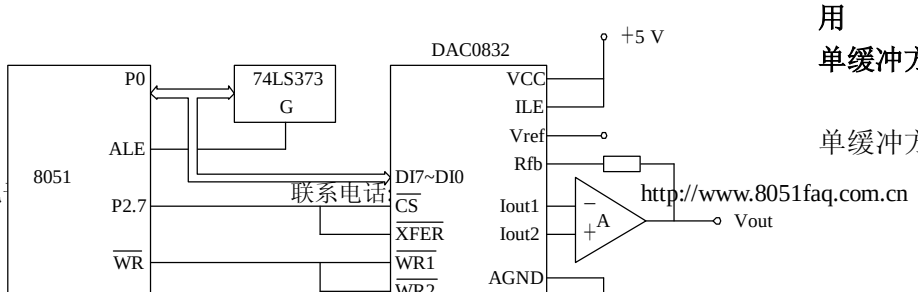
(11) DGND：数字地。

(12) AGND：模拟地。

单缓冲方式的接口与应用

1. 单缓冲方式连接

所谓单缓冲方



式就是使 DAC 0832 的两个输入寄存器中有一个处于直通方式，而另一个处于受控的锁存方式，或者说两个输入寄存器同时受控的方式。在实际应用中，如果只有一路模拟量输出，或虽有几路模拟量但并不要求同步输出时，就可采用单缓冲方式。

2. 单缓冲方式应用举例——产生锯齿波

在许多控制应用中，要求有一个线性增长的电压（锯齿来控制检测过程，移动记录笔或移动电子束等。对此可通过在 DAC0832 的输出端接运算放大器，由运算放大器产生锯齿波来实现，电路连接如图所示。图中的 DAC8032 工作于单缓冲方式，其中输入寄存器受控，而 DAC 寄存器直通。

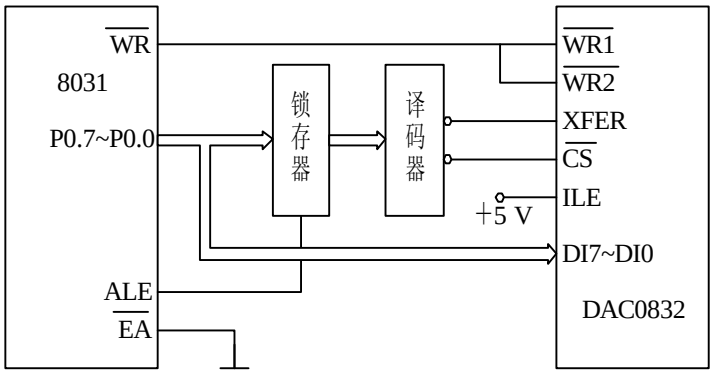
假定输入寄存器地址为 7FFFH，产生锯齿波的源程序清单如下：

```
        ORG     0200H
DASAW：MOV     DPTR，#7FFFH      ；输入寄存器地址，假定 P2.7 接
        MOV     A，#00H           ；转换初值
WW:     MOVX    @DPTR，A          ；D/A 转换
        INC     A
        NOP                       ；延时
        NOP
        NOP
        AJMP    WW
```

双缓冲方式的接口与应用

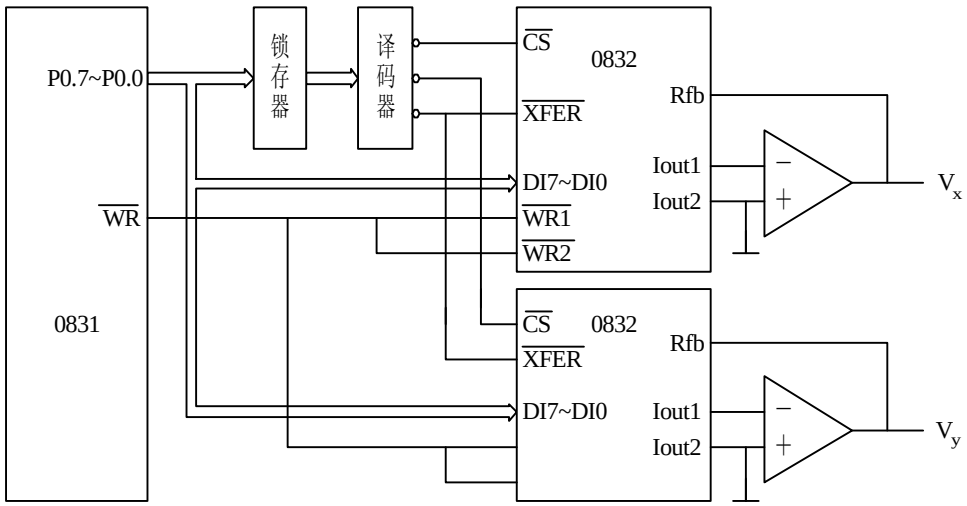
1. 双缓冲方式连接

所谓双缓冲方式，就是把 DAC0832 的两个锁存器都接成受控锁存方式。双缓冲 DAC0832 的连接如图所示。为了实现寄存器的可控，应当给寄存器分配一个地址，以便能按地址进行操作。图中采用地址译码输出分别接和来实现，然后再给和提供写选通信号，这样就完成了两个锁存器都可控的双缓冲接口方式。



2. 双缓冲方式应用举例

双缓冲方式用于多路 D/A 转换系统，以实现多路模拟信号同步输出的目的。例如使用单片机控制 X-Y 绘图仪。X-Y 绘图仪由 X、Y 两个方向的步进电机驱动，其中一个电机控制绘图笔沿 X 方向运动，另一个电机控制绘图笔沿 Y 方向运动，从而绘出图形。因此，对 X-Y 绘图仪的控制有两点基本要求：一是需要两路 D/A 转换器分别给 X 通道和 Y 通道提供模拟信号，二是两路模拟量要同步输出。



第二节 A/D 接口

A/D 转换器用于实现模拟量→数字量的转换，按转换原理可分为 4 种，即：计数式 A/D 转换器、双积分式 A/D 转换器、逐次逼近式 A/D 转换器和并行式 A/D 转换器。

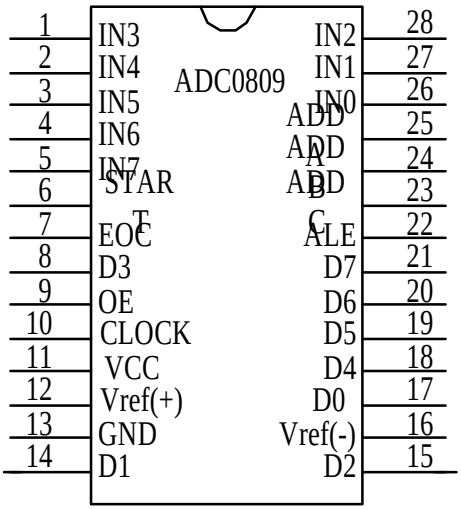
目前最常用的是双积分式 A/D 转换器和逐次逼近式 A/D 转换器。双积分式 A/D 转换

器的主要优点是转换精度高，抗干扰性能好，价格便宜。其缺点是转换速度较慢，因此，这种转换器主要用于速度要求不高的场合。另一种常用的 A/D 转换器是逐次逼近式的，逐次逼近式 A/D 转换器是一种速度较快，精度较高的转换器，其转换时间大约在几μs到几百μs之间。通常使用的逐次逼近式典型 A/D 转换器芯片有：

(1)ADC0801~ADC0805 型 8 位 MOS 型 A/D 转换器（美国国家半导体公司产品）。

(2)ADC0808 / 0809 型 8 位 MOS 型 A/D 转换器。

(3) ADC0816 / 0817。这类产品除输入通道数增加至 16 个以外，其它性能与 ADC0808 /0809 型基本相同。



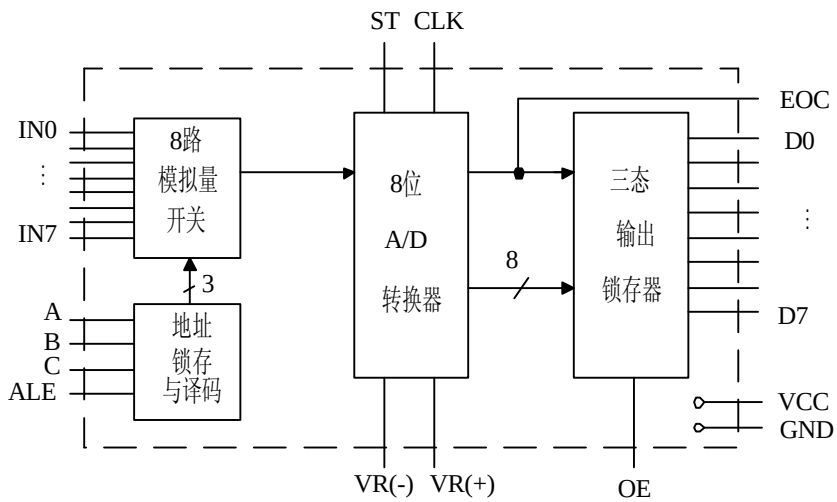
典型 A/D 转换器芯片 ADC0809

ADC0809 是典型的 8 位 8 通道逐次逼近式 A/D 转换器，CMOS 工艺。

1. ADC0809 的内部逻辑结构

ADC0809 内部逻辑结构如图所示。

图中，多路开关可选通 8 个模拟通道，允许 8 路模拟量分时输入，共用一个 A/D 转换器进行转换。地址锁存与译码电路完成对 A、B、C 三个地址位进行锁存和译码，其译码输出用于通道选择。



对 ADC0809 主要信号引脚的功能说明如下：

(1) IN7～IN0：模拟量输入通道。ADC0809 对输入模拟量的要求主要有：信号单极性，电压范围 0～5 V，若信号过小还需进行放大。另外，在 A/D 转换过程中，模拟量输入的值不应变化太快，因此，对变化速度快的模拟量，在输入前应增加采样保持电路。

(2) A、B、C：地址线。A 为低位地址，C 为高位地址，用于对模拟通道进行选择。图中为 ADDA、ADDB 和 ADDC。

(3) ALE：地址锁存允许信号。在对应 ALE 上跳沿，A、B、C 地址状态送入地址锁存器中。

(4)START：转换启动信号。START 上跳沿时，所有内部寄存器清 0；START 下跳沿时，开始进行 A/D 转换；在 A/D 转换期间，START 应保持低电平。

(5)D7~D0：数据输出线。其为三态缓冲输出形式，可以和单片机的数据线直接相连。

(6)OE：输出允许信号。其用于控制三态输出锁存器向单片机输出转换得到的数据。OE=0，输出数据线呈高电阻；OE=1，输出转换得到的数据。

(7)CLK：时钟信号。ADC0809 的内部没有时钟电路，所需时钟信号由外界提供，因此有时钟信号引脚。通常使用频率为 500kHz 的时钟信号。

(8)EOC：转换结束状态信号。EOC=0，正在进行转换；EOC=1，转换结束。该状态信号既可作为查询的状态标志，又可以作为中断请求信号使用。

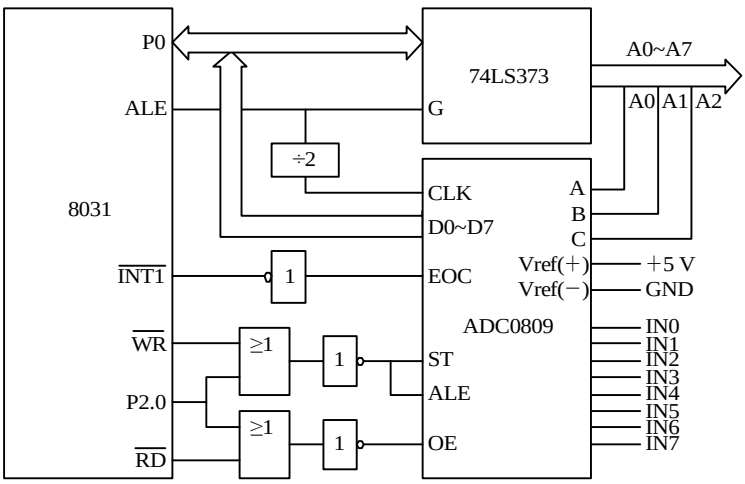
(9)VCC：+5 V 电源。

(10)Vref：参考电源。参考电压用来与输入的模拟信号进行比较，作为逐次逼近的基准。其典型值为+5 V（Vref (+) =+5 V，Vref(-) =0 V）

MCS-51 单片机与 ADC0809 接口

ADC0809 与 8031 单片机的一种连接如图所示。

电路连接主要涉及两个问题，一是 8 路模拟信号通道选择，二是 A/D 转换完成后转换数据的传送。



8 路模拟通道选择

A、B、C 分别接地址锁存器提供的低三位地址，只要把三位地址写入 0809 中的地址锁存器，就实现了模拟通道选择。对系统来说，地址锁存器是一个输出口，为了把三位地址写入，还要提供口地址。图 7.40 中使用的是线选法，口地址由 P2.0 确定，同时和相或取反后作为开始转换的选通信号。

转换数据的传送

A/D 转换后得到的是数字量的数据，这些数据应传送给单片机进行处理。数据传送的关键问题是如何确认 A/D 转换完成，因为只有确认数据转换完成后，才能进行传送。为此，可采用下述三种方式。

1) 定时传送方式

对于一种 A/D 转换器来说，转换时间作为一项技术指标是已知的和固定的。例如,ADC0809 转换时间为 128 μs，相当于 6 MHz 的 MCS-51 单片机 R 64 个机器周期。可据此设计一个延时子程序，A/D 转换启动后即调用这个延时子程序，延迟时间一到，转换肯定已经完成了，接着就可进行数据传送。

2) 查询方式

A/D 转换芯片有表明转换完成的状态信号，例如 ADC0809 的 EOC 端。因此，可以用查询方式，软件测试 EOC 的状态，即可确知转换是否完成，然后进行数据传送。

3) 中断方式

把表明转换完成的状态信号（EOC）作为中断请求信号，以中断方式进行数据传送。

在图中，EOC 信号经过反相器后送到单片机的 UMDJ，因此可以采用查询该引脚或中断的方式进行转换后数据的传送。

不管使用上述哪种方式，一旦确认转换完成，即可通过指令进行数据传送。 首先送出口地址，并以作选通信号，当信号有效时，OE 信号即有效，把转换数据送上数据总线，供单片机接收，即：

```
MOV     DPTR，#0000H        ；选中通道 0
MOVX    A，  @DPTR          ；信号有效，输出转换后的数据到 A 累加器
```