第 2 章

MCS－51 单片机的基本结构及原理

本章将以 MCS－51 系列单片机为例，重点分析单片机的基本结构及原理。为能顺利地学习本章内容，首先应对单片机的相关概念以及常用术语有所了解。

1. 寄存器

寄存器由触发器构成，主要用于暂时存放各种输入、输出数据和运算结果。寄存器的寄存数据是由时钟脉冲 CP 的约定边沿（上升沿或下降沿）触发的。MCS－51 系列单片机中有多种寄存器，均被集成在片内。单片机内的寄存器有通用寄存器组、特殊功能寄存器、专用寄存器组等。

2. 锁存器

锁存器也属于寄存器，用于输入输出电路中，可将数字系统输出的数据锁好，供给外部设备使用；或将外部设备的数据锁存以供给数字系统使用。锁存器图形符号如图 2－1 所示，在 CP 为高电平时，输入数据存入锁存器，锁存后，输出不再随输入而变化；CP 为低电平时，数据保持在锁存器中，当 $\overline{E}$ 出现低电平时，锁存的数据可由 Q0～Q7 输出。管脚名称“E”上的一短横表示低电平有效。

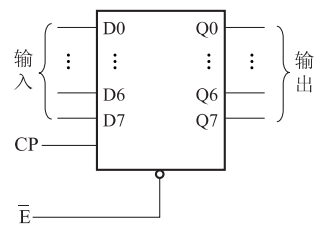

图 2－1　锁存器图形符号

3. 常用术语

（1）位（Bit）位是计算机中所能表示的最基本和最小的数据单位，由于在计算机中使用的都是二进制数，故位就是指一个二进制位。

（2）字节（Byte）相邻的 8 位二进制码称为一个字节，通常数据都以字节为单位存放，如 10110011 为一个字节，因 4 位二进制数可化成为 1 位十六进制数，故 2 位十六进制数就是一个字节。字节简写为“B”，1 KB 就是一千字节，实际字节为 1024。

（3）字长 字是计算机内部进行数据处理的基本单位，它由若干位二进制数码组成，通常与计算机内部的寄存器、运算器、数据总线的宽度（数据的位数）一致。如 MCS－51 系列单片机是 8 位机，说明每次内部数据处理字长也就是字所含的二进制码的位数是 8 位，所以称字长为 8 位。字长也是计算机一次可处理的二进制数位数，字长为字节的整数倍，有 8 位、16 位、32 位和 64 位机等。MCS－96 系列单片机是 16 位单片机，也就是说此种单片机

字长为16位，每次可处理16位，也即2个字节的二进制数。如十六进制数8100H为双字节数，在16位机内存放只需一个寄存器（包含16位），若存放在8位单片机中就必须有两个寄存器。

2.1 MCS－51系列单片机的内部结构

MCS－51是Intel公司20世纪80年代初推出的单片机系列，世界上许多著名的半导体厂商相继生产属于这一系列的兼容产品（如8031，8051，8751，89C51），使产品型号不断增加，品种不断丰富，功能不断加强，但它们都是以典型产品8051为核心，增加一定功能部件构成。本节介绍MCS－51单片机的内部总体结构。

MCS－51系列单片机是双列直插封装形式的集成器件，内部采用模块式的结构，包含了一个独立的微机硬件系统所应具有的各个功能部件和一些重要的功能扩展部件。其结构框图如图2－2所示。

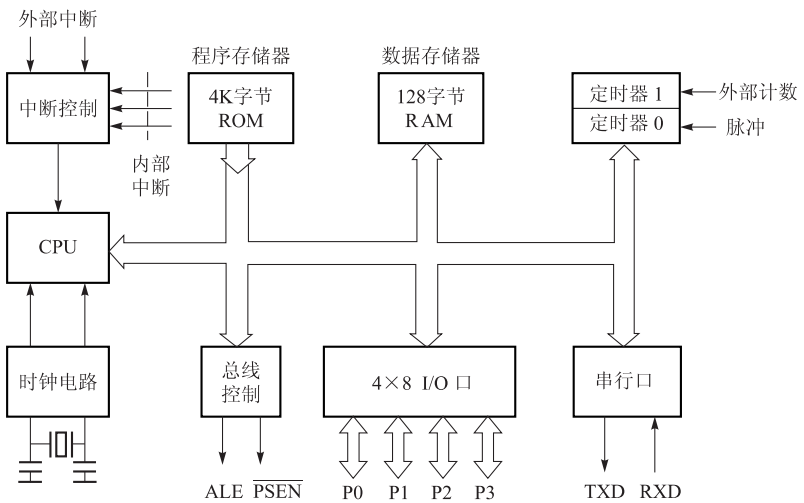

图2－2 MCS－51系列单片机结构框图

1. 微处理器（CPU）

结构框图中的一个重要功能部件是微处理器（CPU），也称中央处理器，一般由运算器和控制器组成。下面介绍它们的功能。

1）运算器

我们常讲计算机处理数据，“处理”的一个重要内容就是运算：一类是算术运算，一类

是逻辑运算。CPU 里完成这些运算的部件就是运算器，它还可以实现数据传送。

（1）运算器主要的单元和寄存器。

① 算术逻辑单元 ALU。

② 两个 8 位暂存器 TMP1 与 TMP2。

③ 8 位累加器 Acc，在指令系统中简写为“A”，经常使用，是最繁忙的寄存器。

④ 寄存器 B。

⑤ 程序状态字 8 位寄存器。

（2）运算器的具体功能。

① 加、减、乘、除算术运算。

② 增量（加 1）、减量（减 1）运算。

③ 十进制数调整。

④ 位的置“1”、置“0”和取反。

⑤ 与、或、异或等逻辑操作。

2）控制器

如果要进行运算，例如“6 +4”，事先应按 MCS – 51 系列单片机指令系统的编程规则编好“6 +4”的程序，存放于程序存储器中。计算机执行程序时，按程序的顺序取一个任务（指令），经寄存、译码、送入定时控制逻辑电路、产生定时信号和控制信号以完成这一任务；再取一个任务，再完成，不会有错，因为 CPU 内有个控制器，控制着整个单片机系统各种操作部件有序工作。一次一次取任务，这样会不会很慢呢？不会，控制器里的时钟发生器可产生一定序列的频率很高的脉冲，每秒钟可进行上万次、几十万次的操作。整个单片机便是在控制器发出的各种控制信号的控制下，统一协调地进行工作的。控制器包括时钟发生器，程序计数器 PC，指令寄存器，指令译码器，存储器的地址/数据传送控制，定时控制逻辑电路等。

程序计数器是控制器中重要的寄存器，简称 PC 或 PC 指针，用于存放指令在程序存储器中的存储地址。8051 的程序计数器有 16 位，但用户不可对它进行读写操作。CPU 根据它提供的存储地址取指令并执行。当取出指令后，PC 自动加“1”就得到下一个存储单元的地址。PC 的新地址值就叫 PC 当前值。如果在执行程序时得到转移指令、子程序调用/返回等指令时，CPU 将转移地址送到 PC。CPU 将从新地址开始执行程序。就像邮递员挨家挨户送信，送完一家，再送它的邻家，一个接一个，突然他接到通知，必须到另条街去送信，那邮递员就必须按命令转到另一条街再挨家挨户地送。

2. 片内程序存储器 ROM 与片内数据存储器

1）存储器概念和地址概念

要单片机完成的一些事情，必须先编好程序，这些用二进制码编成的程序通过键盘等输入设备，存放在存储器中；读/写的数据，如运算的中间结果、最终结果等也要放在存储器

中。所以，存储器像个仓库，只不过这个仓库不存放物品，而存放用 0，1 表示的程序和数据。

存储器也有很多存储单元，8051 单片机的一个存储单元可存放 8 位二进制数。CPU 对某个存储单元进行数据读写操作时，为了区别存储单元，需要给每个单元编号，这就是存储单元的地址。CPU 通过地址总线送出要寻找的存储单元地址。

2）分类

根据用途，存储器可分为程序存储器和数据存储器。

（1）程序存储器。单片机内部的程序存储器按字节存放指令和原始数据，主要有以下几类：

① ROM 型单片机。这种单片机程序存储器中的内容是给用户固化的专用程序，不可改写，如 8051。

② EPROM 型单片机。其内容可由用户通过编程器写入，也可通过紫外线擦除器擦除，如 8751。

③ Flash Memory 型单片机。内部含有快速的 Flash Memory 程序存储器，用户可用编程器对程序存储器反复擦除、写入，使用十分方便，如 89C51。

④ 无程序存储器的单片机。这种单片机内部没有程序存储器，必须外接 EPROM 程序存储器，如 8031。

（2）数据存储器。数据存储器是用来存放数据的存储器，MCS－51 系列单片机内部有 RAM 和特殊功能寄存器两种类型的数据存储器。

3）存储器容量

存储器的存储容量表示可存放二进制数的多少，存储的数据越多，存储的容量就越大。存储容量通常用存储单元与每个存储单元的位数来表示。如某芯片存储容量为 1 024 ×8，表明该芯片有 1 024 个存储单元，每个单元存放 8 位二进制数，也称芯片存储容量为 1 KB。MCS－51 系列单片机的内部程序存储器的存储容量一般为 4 K～32 KB，内部数据存储器的存储容量为 256 B。

MCS－51 芯片内的存储器有各自的地址空间，内外存储器的配置将在本章 2.4 节中详细讲述。

3. 并行输入/输出（I/O）端口

8051 单片机有 4 个并行输入/输出的端口 P0～P3，每个端口都可进行 8 位输入或输出操作，这些端口是单片机与外部设备或器件进行信息交换的主要通道，这种方式就是并行通信。并行通信速度快，适合于近距离通信。如 P1 口（8 位）是一个并行接口，作为输出口时，CPU 将一个 8 位数据写入 P1，这 8 位数据在 P1 口的 8 个引脚上并行地同时输出到外部设备。

MCS－51 芯片内的 4 个并行输入/输出口 P0～P3 的内部结构及使用将在本章 2.3 节中

详细讲述。

4. 定时/计数器

单片机内部有两个 16 位定时/计数器，它既可设置成计数方式，用于计数，又可设置成定时方式，实现定时，并以其定时或计数结果对单片机进行控制。

5. 中断源

MCS – 51 系列单片机的中断功能很强，以满足控制的需要。8051 共有 5 个中断源，包括外部中断源 2 个，内部中断源 3 个：定时/计数中断 2 个，串行中断 1 个。

6. 串行口

数据一位一位串行顺序传送，称为“串行通信”。8051 具有一个双工的串行接口，全双工的串行通信就是用两根线连接到发送器和接收器，其中一根用于发送数据，另一根用于接受数据，这样每根线只负担一个方向数据传送，这种通信适用于远距离通信。

7. 时钟电路

MCS – 51 系列单片机的内部有时钟电路，外接石英晶体和微调电容，可振荡产生 1. 2 ~ 12 MHz 的时钟频率，8051 的频率多数为 12 MHz，振荡周期为 1/12 μs，一个振荡脉冲称为一个拍节，用“P”来表示；振荡脉冲经过二分频后就是单片机的时钟信号，把时钟信号的周期称为状态，用“S”来表示。这样，一个时钟信号包含两个振荡脉冲，每两个振荡周期就组成一个状态周期，即 1/6 μs。状态周期是完成一种微机操作的周期。机器周期包含有 6 个状态周期，它是指完成一种基本操作的周期，故机器周期为 1 μs。振荡脉冲、状态周期、机器周期的关系如图 2 – 3 所示。

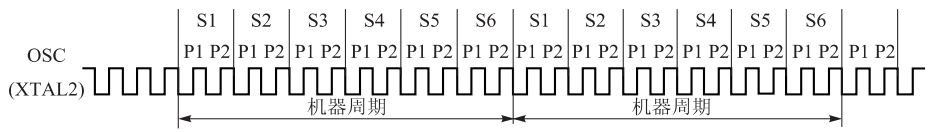

图 2 – 3　振荡脉冲、状态周期、机器周期的关系

8. 总线

上述这些部件都是通过总线连接起来，从而构成一个完整的单片机系统。单片机的总线按功能可分为地址总线（AB）、数据总线（DB）、控制总线（CB）三种。系统的地址信号、数据信号和控制信号都是通过相应的总线传送的。总线结构减少了单片机的连线和引脚，提高了集成度和可靠性。总线在图中可以有两种表示方法：

（1）用带箭头的空心线表示。

（2）用一根带小斜杠的线段表示，斜杠边的数字注明的是总线的根数。

存储器存储单元的数量应有相应的地址总线宽度，即地址总线根数，如现有存储单元为 8 个，就需有 3 根地址线，这样可形成 $2^3 = 8$ 个单元地址，所以存储器存储容量决定了与之

相连的地址总线根数。MCS－51 系列单片机的内部数据存储器有 256 个单元，故应有 8 根地址总线。每个存储单元含有的位数决定了与之相连的数据总线的根数，若一个存储单元可存 8 位二进制数，就必须有 8 根数据线。

2.2 MCS－51 单片机的引脚及功能

上一节我们重点学习了 MCS－51 系列单片机的内部总体结构，对 8051 的 CPU 和存储器有了基本了解。单片机发挥控制作用，其内部总是要和外界进行通信，输入或输出信息。单片机的引脚成为片内、片外联系的通道，用户只能使用引脚，即通过引脚组建控制系统，因此熟悉引脚是学习单片机的重要内容。本节重点讲述 MCS－51 单片机的引脚及功能。

8051 为 40 脚双列直插封装形的器件，其引脚图和逻辑符号如图 2－4 所示。

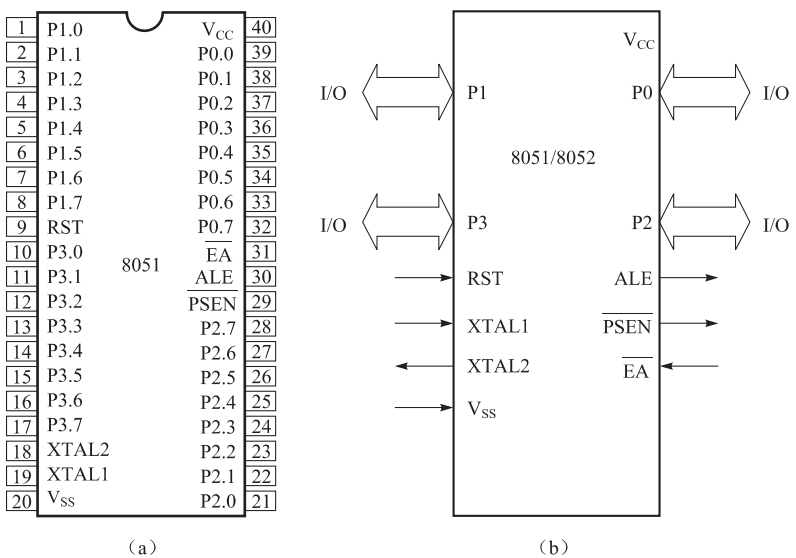

图 2－4　8051 的引脚图及逻辑符号

（a）8051 的引脚图；（b）8051 逻辑符号

根据集成器件引脚序列的有关规定，按图示正面视图方向，缺口在上方，左上方为第 1 管脚。按逆时针方向依次标号，图（a）所示为各管脚的编号及名称（复用引脚只标第一功能），图（b）所示为 8051 的逻辑符号，带箭头的空心线段表示总线，箭头方向表示信号流向，双向箭头表示既可输入，又可输出。

40 个引脚大致可分为电源、时钟、复位、I/O 口、控制总线几个部分，下面具体分析

它们的功能。

1. 电源引脚

① V_{CC}（40 脚）8051 工作电源接线，接 +5 V 直流。

② V_{SS}（20 脚）8051 接地端。

2. 时钟振荡电路引脚 XTAL1（19 脚）和 XTAL2（18 脚）

XTAL1 和 XTAL2 是时钟电路的引脚，时钟振荡电路的接法如图 2 – 5 所示，有图（a）和图（b）两种接法。图（a）是外接石英晶体和微调电容，与内部电路构成振荡电路，其振荡频率就是石英晶体固有频率，振荡信号送至内部时钟电路产生时钟脉冲信号。图（b）是 XTAL1 与 XTAL2 的另一种接法，XTAL1 接地，XTAL2 接外部时钟电路，由外部时钟电路向片内输入时钟脉冲信号。

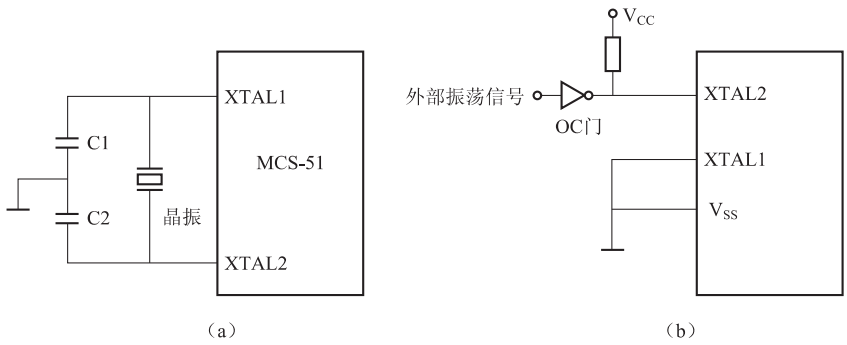

图 2 – 5　时钟振荡电路的接法

（a）内部时钟振荡电路；（b）外部时钟输入接线图

3. 复位引脚 RST（9 脚）

单片机在启动运行时都需要复位，复位可使 CPU 和系统中的其他部件都处于一个确定的初始状态，并从这个初始状态开始工作。当复位引脚 RST 上出现一定时间（约 2 个机器周期）的高电平时，系统就复位，内部寄存器处于初始状态；若保持高电平，单片机就处于循环复位；RST 从高电平变为低电平后，CPU 从初始化状态开始工作。复位以后，内部寄存器初始状态见本章第 4 节。

单片机的复位方式有两种：

（1）上电自动复位电路。上电自动复位电路如图 2 – 6 所示，其复位是利用 RC 充电来实现的。加电瞬间，V_{RST} = 5 V（高电平），电容充电，V_{RST} 下降，RC 越大则充电越慢，V_{RST} 下降越慢，保持一定时间高电平即可可靠复位。若复位电路失效，加电后 CPU 不能正常工作。

（2）人工复位。人工复位如图 2 – 7 所示，是将一个按钮开关并联于上电自动复位电

路，按一下按钮，在 RST 端出现一段时间的高电平，使单片机复位。

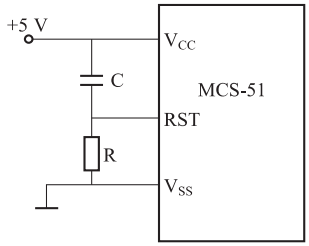

图 2－6　上电自动复位电路

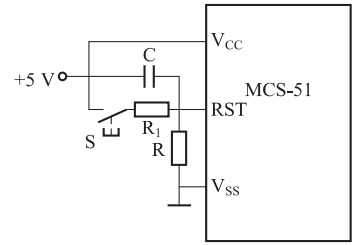

图 2－7　人工复位

关于单片机的具体复位电路将在后面的第 5 章中作详细介绍。

4. 控制信号线

（1）ALE（30 脚）。低 8 位地址锁存控制信号。在计算机系统中，为了减少 CPU 芯片引脚数目，常采用地址/数据分时复用同一引脚。MCS－51 系列单片机读写外部存储器时，P0 口先输出低 8 位地址信息，待地址信息稳定并可靠锁存后，P0 口作为数据总线使用，实现低位地址和数据的分时传送。因此，当这类 CPU 与外部存储器相连时，作为地址/数据分时复用引脚，需要通过锁存器，如 74LS373，与存储器地址线相连，同时 CPU 也必须提供地址锁存信号 ALE。

在访问外部程序存储器的周期内时，ALE 信号有效两次；而访问外部数据存储器的周期内，ALE 信号有效一次。

（2）$\overline{\text{PSEN}}$（29 脚）。外部程序存储器读选通信号，低电平有效。在访问外部程序存储器时，此引脚定时输出负脉冲作为读取外部程序存储器的信号，一个机器周期内两次有效，但访问内部 ROM 和外部 RAM 时，不会产生$\overline{\text{PSEN}}$信号。

（3）$\overline{\text{EA}}$（31 脚）。程序存储器控制信号。$\overline{\text{EA}}$ = 1，CPU 访问程序存储器有两种情况：

① 当访问地址在 0～4 KB 范围内时，CPU 访问片内程序存储器。

② 当访问地址超出 4 KB 时，CPU 自动访问外部程序存储器。

$\overline{\text{EA}}$ = 0，CPU 只访问外部程序存储器 ROM。

5. 并行 I/O 端口

（1）P0.0～P0.7。P0 口有 8 位双向口线。在读写外部存储器时，P0 口作为“低 8 位地址/数据”总线使用。

（2）P1.0～P1.7。P1 口有 8 位双向口线。P1.0，P1.1 引脚除了可作为一般 I/O 引脚使用外，还具有第二输入/输出功能：P1.0/ T2 为定时器 T2 的计数输入端或定时器 T2 的时钟输出端；P1.1/ T2EX 为定时器 T2 外部触发输入端。

（3）P2.0～P2.7。P2 口有 8 位双向口线。在读写外部存储器时，P2 口输出高 8 位地址

A15 ~ A8。

（4）P3.0 ~ P3.7。P3 口有 8 位双向口线。P3 口除了可作为一般 I/O 引脚使用外，还具有第二输入/输出功能。

6. 引脚的第二功能

由于工艺及标准化等原因，芯片的引脚数目是有限的，如 MCS –51 系列为 40 脚，但单片机为实现控制所需要的信号数目却远远超过此数。单片机通过复用引脚来解决这一矛盾，即一些引脚具有第二功能。如第 9，30，31 引脚和 P3 口的 8 位都具有第二功能。其中 P3 口的每一位的第二功能如下：

（1）P3.0/RXD　串行数据接收（输入）端。

（2）P3.1/TXD　串行数据发送（输出）端。

（3）P3.2/$\overline{\text{INT0}}$　外部中断 0 中断请求输入端。

（4）P3.3/$\overline{\text{INT1}}$　外部中断 1 中断请求输入端。

（5）P3.4/T0　定时器/计数器 0 的外部输入端。

（6）P3.5/T1　定时器/计数器 1 的外部输入端。

（7）P3.6/$\overline{\text{WR}}$　外部数据存储器“写”选通信号，低电平有效。

（8）P3.7/$\overline{\text{RD}}$　外部数据存储器“读”选通信号，低电平有效。

P3 口的第二功能的信号是单片机的重要控制信号，在实际使用时，总是按需要优先选用它的第二功能，剩下不用的才作为输入输出口线使用。

2.3　输入/输出（I/O）口内部结构及使用

MCS –51 系列单片机理论上有 4 个 8 位 I/O 口，即 P0 口、P1 口、P2 口和 P3 口等，等效电路如图 2 –8 所示。

2.3.1　P1 口内部结构及使用

P1 口的结构最简单，如图 2 –8（a）所示。用途也单一，除 P1.0，P1.1 引脚具有第二输入/输出功能外，P1.7 ~ P1.2 仅作为数据输入/输出引脚使用。

输出时，数据经内部总线、锁存器反相输出端 $\overline{Q}$，V2 管栅极和漏极到 P1 口引脚。例如，输出“1”电平时，在写锁存器时钟作用下，锁存器反相输出端 $\overline{Q}$ 为低电平，V2 管截止，漏极输出高电平，结果 P1 口对应引脚输出高电平；反之，输出“0”电平时，在写锁存器时钟作用下，锁存器反相输出端 $\overline{Q}$ 为高电平，V2 管导通，漏极输出低电平（漏源之间导通电阻很小，仅为几十欧到几百欧，而漏极等效上拉电阻一般为数十千欧，分压后，P1.X 引脚电位接近 0 V，输出低电平）。

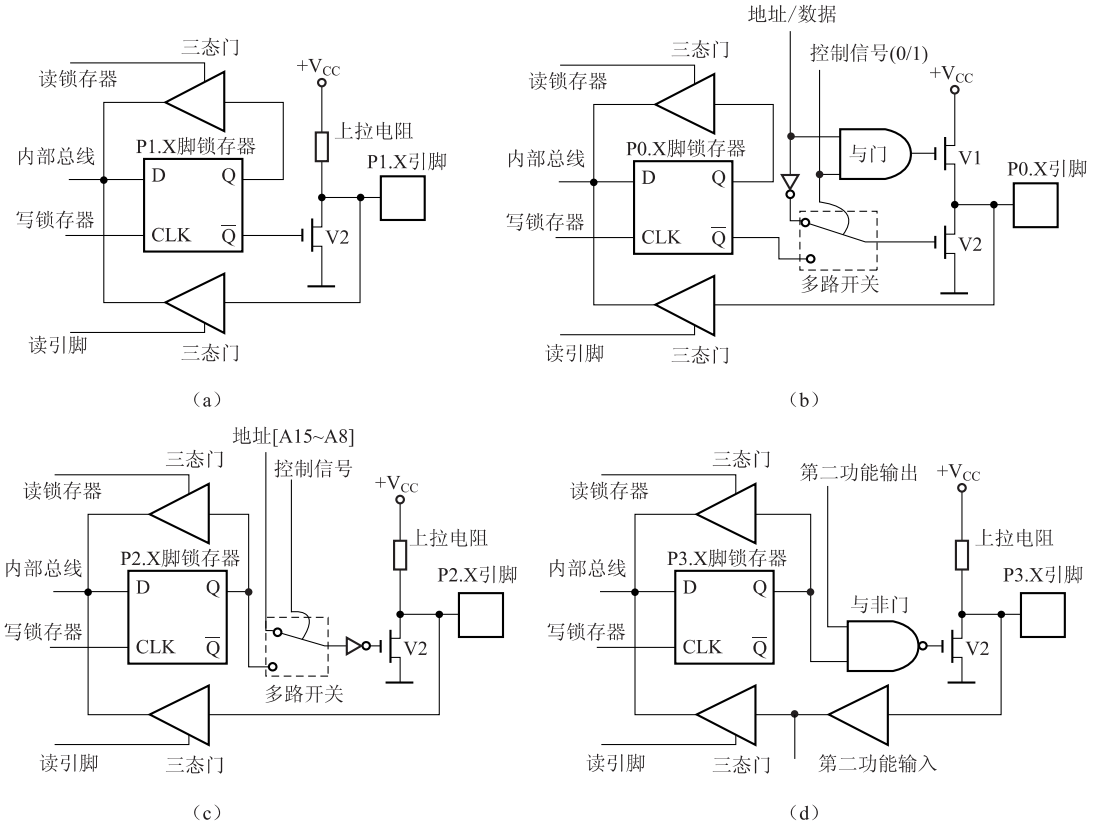

图 2－8　MCS－51 系列单片机 I/O 口等效电路

（a）P1 口；（b）P0 口；（c）P2 口；（d）P3 口

作为输入口时，必须先执行写端口指令，如：SETB P1. X 或 MOV P1，#0FFH 等，将 P1 口锁存器置 1，锁存器反相端 $\overline{Q}$ 输出低电平，使 V2 管截止；否则 P1. X 引脚有可能被钳位在低电平状态。例如，在图 2－9 所示电路中，当三极管基极输入低电平时，集电极输出高电平，但如果 P1. X 锁存器反相输出端 $\overline{Q}$ 为高电平，V2 管导通，P1. X 引脚将被钳位在低电平状态，读入一个错误信息。因此，当把 P1. X 引脚作为输入引脚使用时，必须确保 P1 口相应位锁存器为“1”，在读引脚信号作用下，输入信息经 P1. X 引脚、读引脚三态门电路到内部总线。

之所以安排读锁存器三态门，是为了防止当输出端驱动 NPN 三极管基极时，读引脚将获得错误信息，如图 2－10 所示。例如，当锁存器为“1”，P1. X 端输出高电平，三极管导通，但三极管导通后，BE 结电压仅为 0. 7 V 左右（硅管），输出端即被钳位在 0. 7 V，这样读引脚将得到“低电平”的错误信息，而实际上，P1. X 却输出高电平。

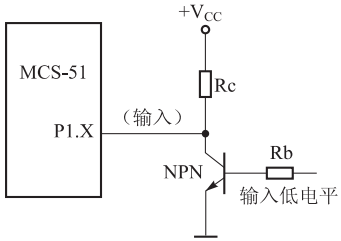

图 2 –9　P1. X 作为输入引脚的示意图

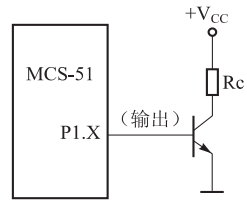

图 2 – 10　驱动三极管基极时 I/O 引脚被钳位

在增强型 MCS – 51 芯片中，P1. 0 和 P1. 1 引脚具有第二输入/输出功能，即除了可作为一般 I/O 引脚使用外，P1. 0 引脚还作为定时器 T2 的计数输入端或 T2 定时时钟输出端，而 P1. 1 引脚作为定时器 T2 外部触发输入端 T2EX。

2. 3. 2　P0 口内部结构及使用

P0 口内部结构如图 2. 8（ b）所示。对于内置了 ROM，EPROM，OTP ROM，Flash ROM 的 80C5X，87C5X，89C5X 芯片来说，当不使用外部存储器（ 包括程序存储器和数据存储器）时，可作为通用的输入/输出端口（ I/O ）使用；当需要扩展外部存储器时，P0 口是“地址/数据”总线。下面分别介绍 P0 口作为 I/O 端口和地址/数据总线使用时的工作过程和信号流向。

1. 作为 I/O 端口时

P0 口作为 I/O 端口使用时，多路开关控制信号为“0”（ 低电平），与门输出低电平，V1 管截止，同时多路开关转向锁存器反相输出端 $\overline{Q}$。输出时，写锁存器脉冲 CLK 有效，输出信号经内部总线→锁存器输入端 D→反相输出端 $\overline{Q}$→多路开关→V2 栅极→V2 漏极到输出端。由于 V1 管截止，所以作为输出口时，P0 口是漏极开路输出，类似于 OC 门。当驱动上拉电流负载时，需要外接上拉电阻，P0 口带有锁存器，因此具有输出锁存功能。P0 作为输入口时，与 P1 口类似，也必须先执行写端口指令，如：SETB P0. X 或 MOV P0，#0FFH，将 P0 口锁存器置“1”，$\overline{Q}$ 端输出低电平，使 V2 管截止（ 这时 V1，V2 均截止，P0. X 引脚悬空），否则 P0. X 引脚也有可能被钳位在低电平状态。在读引脚信号作用下，输入信息经 P0. X 引脚→读引脚三态门电路到内部总线。

2. 作为地址/数据总线时

在访问外部存储器时，P0 口作为地址/数据总线使用。这时，多路开关控制信号为“1”，与门解锁，与门输出信号电平由“地址/数据”线信号决定，同时，多路开关与反相器的输出端相连，地址信号经“地址/数据”线→反相器→V2 栅极→V2 漏极输出。例如，地址信号为“0”，与门输出低电平，V1 管截止，反相器输出高电平，V2 管导通，输出引脚的地址信号为低电平；反之，地址信号为“1”，与门输出高电平，V1 管导通，反相器输

出低电平，V2 管截止，输出引脚的地址信号为高电平。可见，在输出“地址/数据”信息时，V1，V2 交替导通，带负载能力很强，可以直接与存储器地址线相连，无需增加总线驱动器。

P0 口又可作为数据总线使用。在访问外部程序存储器时，P0 口输出低 8 位地址信息后，将变为数据总线，以便读指令码（输入）。在取指令期间，控制信号为“0”，V1 管截止，多路开关也跟着转向锁存器反相输出端 $\overline{Q}$，同时，CPU 自动将 0FFH 写入 P0 口锁存器，使 V2 管截止，在读引脚信号控制下，通过读引脚三态门电路将指令码读到内部总线。

如果该指令是输出数据，如“MOVX @ DPTR，A”（将累加器 A 中的内容通过 P0 口数据总线传送到外部 RAM 中），则多路开关控制信号为“1”，与门解锁，与输出地址信号类似，数据由地址/数据线→反相器→V2 栅极→V2 漏极输出。

如果该指令是输入数据（读外部数据存储器或程序存储器），如“MOVX A，@ DPTR”（将外部 RAM 某一存储单元内容通过 P0 口数据总线输入到累加器 A 中），则输入的数据仍通过读引脚三态门到内部总线，其过程类似于读指令码。

通过以上分析，可以看出当 P0 口作为地址/数据总线使用时，在读指令码或输入数据前，CPU 自动向 P0 口锁存器写入 0FFH，破坏了 P0 口原来的状态。因此，不能再作为通用 I/O 端口，这点在系统硬件设计时务必注意，即程序中不能再含有以 P0 口作为操作数（包括源操作数和目的操作数）的指令。

2.3.3 P2 口内部结构及使用

P2 口的内部结构如图 2－8（c）所示，可以作为通用的 I/O 端口使用，也可以作为外部存储器高 8 位地址总线使用，在读写外部存储器期间，输出高 8 位（A15 ~ A8）地址信息。

1. 作为 I/O 端口时

没有外部程序存储器或虽然有外部数据存储器，但容量不大于 256 B，不需要高 8 位地址时（在这种情况下，不能通过数据地址寄存器 DPTR 读写外部数据存储器），P2 口可以作为 I/O 端口使用。这时，控制信号为“0”，多路开关转向锁存器同相输出端 Q，输出信号经内部总线→锁存器输出端 Q→反相器→V2 管栅极→V2 管漏极输出。由于 V2 管漏极带有上拉电阻，可以提供一定的上拉电流，负载能力约为 4 个 TTL 门电路。作为输入口前，同样需要向锁存器写入“1”，使反相器输出低电平，V2 管截止，即引脚悬空时为高电平，防止引脚被钳位在低电平。读引脚信号有效后，输入信息经读引脚三态门电路到内部数据总线。

2. 作为地址总线时

P2 口作为地址总线时，控制信号为“1”，多路开关转向地址线，地址信息经反相器→

V2 管栅极→V2 管漏极输出，由于 P2 口输出高 8 位地址，与 P0 口不同，无需分时使用，因此 P2 口上的地址信息（程序存储器的 A15 ~ A8）或数据地址寄存器高 8 位 DPH 保存时间长，无需锁存。关于这点可参看如图 2 – 15 所示的外部存储器读写时序。

2.3.4　P3 口内部结构及使用

P3 口内部结构如图 2 – 8（d）所示。不过 P3 口是个多功能口，它除了可作为一般的 I/O 口外，还具有第二功能。

作为 I/O 口时，第二功能输出控制信号为高电平，与非门等效为一个反相器，与 P2 口情况类似。此外，做第二功能输出时，CPU 会自动向锁存器写入“1”，打开与非门，这时与非门同样等效于一个反相器，第二功能输出信号经与非门→V2 管的栅极→漏极→P3. X 引脚；做第二功能输入时，第二功能输出控制端、锁存器输出端均为“1”，与非门输出低电平，V2 管截止，输入信号经引脚→缓冲器→第二功能输入。

从图 2 – 8 看出，作为第二功能输出引脚使用前并不需要做任何引脚切换设置，只要相应外设处于使能状态，对应 I/O 引脚就具有第二功能输出信号含义。例如，在“MOVX @ DPTR，A”指令执行期间，P3. 6 引脚自动输出外部数据存储器写控制信号$\overline{WR}$。而作为第二功能输入引脚使用前，也无需设置，只要相应引脚 I/O 口锁存器为 1（否则 I/O 口下拉 MOS 管导通，输入信号被钳位在 0 电平），则当对应外设处于使能状态时，就自动具有第二功能输入特性（当然这时仍可通过读引脚指令获取引脚的电平状态）。

2.3.5　I/O 口负载能力

由于 P1 ~ P3 口上拉电阻较大，约为 20 ~ 40 kΩ，属于“弱上拉”，因此 P1 ~ P3 口引脚输出高电平电流 I_{OH}很小（约为 30 ~ 60 μA）。而输出低电平时，下拉 MOS 管导通，可吸收 1. 6 ~ 15 mA 的灌电流，负载能力较强，即 P1 ~ P3 口负载能力为 3 ~ 4 个 TTL 门电路。

作为输出口驱动 NPN 三极管时，在最坏情况下（上拉电阻为 40 kΩ），如果三极管电流放大系数 β 取 100，则最大集电极电流 I_{Cmax}约为 10 mA。因此，P1 ~ P3 口可以直接驱动小功率 NPN 三极管，如图 2 – 11（a）所示。当驱动电流较大的中功率 NPN 三极管时，必须外接上拉电阻（但上拉电阻不得小于 3. 3 kΩ，否则输出低电平时灌电流会大于 1. 6 mA），如图 2 – 11（b）所示。

作为输出口驱动 PNP 三极管时，必须在 I/O 端口与三极管基极之间接串限流电阻（阻值大小与最大集电极输出电流有关），限制输出低电平时 I/O 口灌电流，如图 2 – 11（c）所示。MCS – 51 芯片 I/O 口输出级采用准双向结构，低电平驱动电流较大——多数增强型 MCS – 51 芯片 I/O 引脚能够吸收 1. 6 ~ 15 mA 的灌电流（但同一 I/O 口所有引脚灌电流总和 $\sum I_{OL}$，以及 I/O 引脚灌电流总和存在一个最大值，可从器件手册查到），当三极管电流放

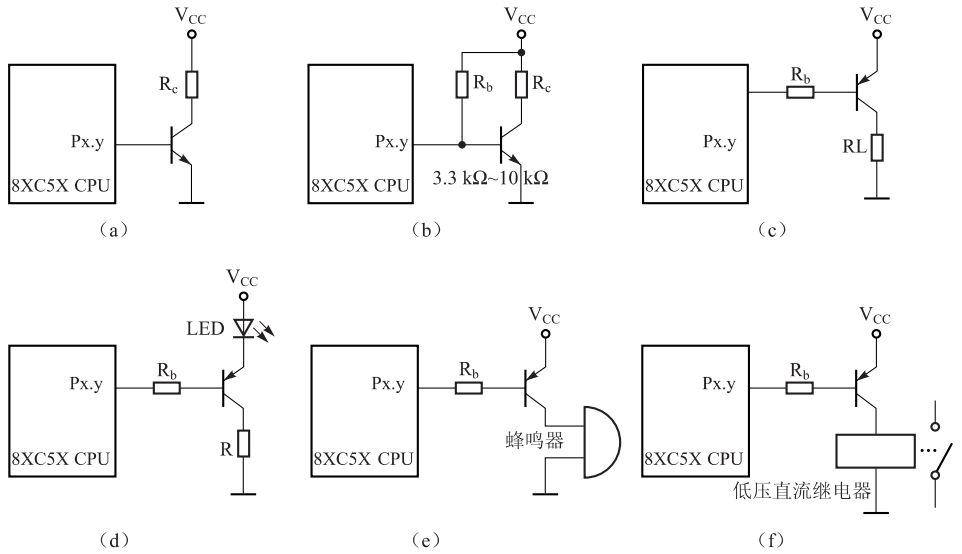

图 2－11　P1～P3 口驱动三极管电路

大系数 β 取 100 时，则最大集电极电流 I_{Cmax} 大于 160 mA，足可以驱动小型继电器。此时限流电阻 R_b 约为（5.0 V－0.7 V－0.4 V）/1.6 mA，即 2.7 kΩ（假设电源电压为 5.0 V）。因此，当需要驱动工作电流较大的 LED 发光二极管、蜂鸣器、小型继电器时，拟采用图 2－11（d）～（f）所示的驱动方式。

采用低电平有效驱动方式，除了驱动能力较强外，不输出时 Px. y 引脚锁存器输出高电平，I/O 口输出级下拉 N 沟 MOS 以及负载驱动管（PNP）均截止，功耗小；另一方面也避免了复位期间或复位后立即输出的弊端。

作为输出口使用时，P0 口漏极开路，当需要驱动上拉电流负载时，必须外接上拉电阻；输出低电平负载能力比 P1～P3 口强，可以吸收 3.2 mA 以上的灌电流，能驱动 8 个 TTL 门电路。

因 P1～P3 口上拉电阻较大，而 P0 口为漏极开路，因此作为输出口使用时 P0，P1～P3 口引脚均具有“线与”功能。

2.3.6　读锁存器和读引脚指令

当把 P0～P3 口作为输入引脚使用时，以 I/O 口作为源操作数的数据传送指令、算术及逻辑运算指令、位测试转移指令等属于读引脚指令。如：

```
MOV C，P1.0          ；将 P1.0 引脚状态读到位累加器 C 中。
MOV A，P1            ；将 P1 口的 P1.0～P1.7 引脚信号读到累加器 A 中。
```

```
ANL A，P1              ；将 P1 口的 P1.0 ~ P1.7 引脚信号与累加器 A 相与。
ADD A，P1              ；将 P1 口的 P1.0 ~ P1.7 引脚信号与累加器 A 相加。
JB P1.0，LOOP          ；P1.0 引脚信号为 1，则转移。
JNB P1.0，LOOP         ；P1.0 引脚信号为 0，则转移。
```

而所有的“读 – 改 – 写”指令均读 I/O 口锁存器，如：

```
JBC P1.0，LOOP         ；P1.0 锁存器为 1 转移，且将 P1.0 锁存器清零。
DEC P1
INC P1
CPL P1.0
```

2.4　存储器系统

多数单片机系统，包括 MCS – 51 系列单片机存储器组织方式与通用微机系统不同，程序存储器地址空间和数据存储器地址空间相互独立，并通过各自数据总线与 CPU 相连，以加快程序执行速度。而通用微机系统的程序存储器和数据存储器往往共用同一存储区，统一编址。

8XC5X 系列单片机的存储器由三部分组成，即程序存储器（包括片内程序存储器，大小与芯片型号有关，如 89C52 片内程序存储器容量为 8 KB，地址编码从 0000 ~ 1FFFH；89C54 片内程序存储器容量为 16 KB，地址编码从 0000 ~ 3FFFH；89C58 片内程序存储器容量为 32 KB，地址编码从 0000 ~ 7FFFH；外部程序存储器，地址编码从 0000 ~ FFFFH，共 64 KB）、片内数据存储器（包括内部 RAM 存储器 00 ~ FFH，共 256 字节；特殊功能寄存器）、外部数据存储器（0000 ~ FFFFH，共 64 KB），如图 2 – 12（a）所示。

对于 80C31，8XC51 芯片来说，片内数据存储器容量为 128 字节（00 ~ 7FH），如图 2 – 12（b）所示。

尽管数据存储器地址空间与程序存储器地址空间重叠，但不会造成混乱。原因是访问外部程序存储器时用$\overline{\text{PSEN}}$信号选通；而访问外部数据存储器时，由$\overline{\text{RD}}$信号（读）和$\overline{\text{WR}}$信号（写）选通。

数据存储器由片内数据存储器（内部 RAM）和外部数据存储器组成，尽管地址空间重叠，但也不会造成混乱。因为内部数据存储器通过 MOV 指令读写，使用内部数据总线，此时外部数据存储器读选通信号$\overline{\text{RD}}$、写选通信号$\overline{\text{WR}}$均无效；而外部数据存储器通过 MOVX 指令访问，并由$\overline{\text{RD}}$（读操作）或$\overline{\text{WR}}$信号（写操作）选通。

在 8XC32/8XC52/54/58 系列中，尽管高 128 字节内部 RAM 地址空间与特殊功能寄存器地址空间重叠，但使用不同寻址方式访问，也不会造成混乱。例如：

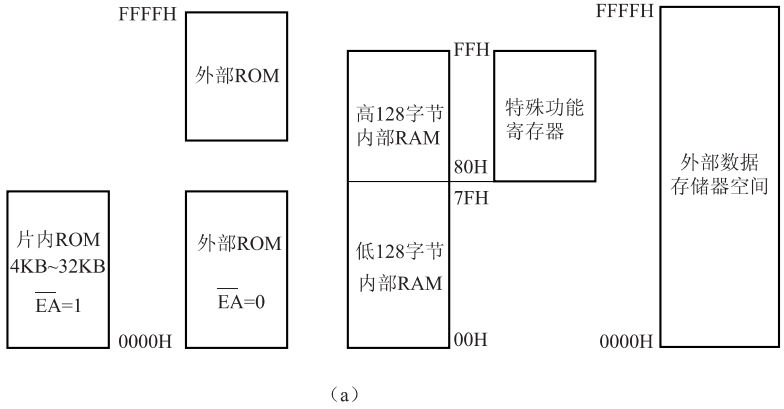

图 2－12　8XC5X/8XC3X 系列单片机存储器结构

```
MOV R0，#90H
MOV @R0，A    ；把累加器 A 内容送内部 RAM 90H 单元。MCS－51 规定，高 128 字
              ；节内部 RAM 只能用寄存器间接寻址方式访问。
MOV 90H，A    ；把累加器 A 内容送地址为90H 的特殊功能寄存器（即 P1 口）。MCS－51
              ；系列单片机规定特殊功能寄存器只能用直接寻址方式读写。
```

2.4.1　程序存储器

对于带有片内 ROM 的 MCS－51 系列单片机来说，片内程序存储器和外部程序存储器地址空间重叠。如果$\overline{EA}/V_{PP}$引脚为高电平，且程序计数器 PC 小于等于片内 ROM 的地址空间时，将从片内程序存储器取指令（在这种情况下，$\overline{PSEN}$信号无效）；而当 PC 超出片内 ROM 地址空间时，自动到外部程序存储器取指令，即在 P0 口输出低 8 位地址（A0～A7），在 P2

口输出高 8 位地址（A15 ~ A8）。当$\overline{EA}/V_{PP}$引脚为低电平时，一律从外部程序存储器取指令。因此对于不带 ROM 或 EPROM 的 80C31，80C32 CPU 来说，$\overline{EA}/V_{PP}$引脚一律接地。

在增强型 MCS –51 系列单片机中，大部分芯片均内置了不同容量的 EPROM（紫外光可擦除的只读存储器）、OTP EPROM（一次性编程的只读存储器，即没有擦除窗口的 EPROM）、F1ash ROM，一般无需使用外部程序存储器芯片，$\overline{EA}/V_{PP}$引脚一律通过 2.0 ~ 4.7 kΩ 电阻与电源 V_{CC}相连。

增强型 MCS –51 系列单片机保留的程序存储器地址空间如下。

系统复位	0000H
外部中断 0（$\overline{INT0}$）服务程序入口地址	0003H
定时器 0 中断服务程序入口地址	000BH
外部中断 1（$\overline{INT1}$）服务程序入口地址	0013H
定时器 1 中断服务程序入口地址	001BH
串行口中断服务程序入口地址	0023H
定时器 2 中断服务程序入口地址	002BH

复位后，程序计数器 PC 为 0000H，即从程序存储器的 0000H 单元读出第一条指令，因此可在 0000H 单元内放置一条跳转指令，如 LJMP XXXX（XXXX 表示主程序入口地址）。由于系统给每一中断服务程序预留了 8 个字节，因此，用户主程序一般存放在 0033H 单元以后。如：

```
ORG 0000H          ；伪指令 ORG 指示随后的指令码从 0000H 单元开始存放。
LJMP Main          ；在 0000H 单元放一条长跳转指令，共 3 个字节。
ORG 0003H
LJMP INT0          ；跳到外中断INT0服务程序的入口处。
…                  ；初始化其他中断入口地址。
ORG 50H            ；主程序代码从 50H 单元开始存放。
Main:              ；Main 是主程序入口地址标号。
```

2.4.2　片内数据存储器

片内数据存储器由内部 RAM 和特殊功能寄存器组成。对于 8XC51，8XC31 芯片来说，内部 RAM 的容量为 128 字节（00 ~ 7FH）；对于 8XC52/54/58 芯片来说，片内 RAM 容量为 256 字节（00 ~ 0FFH）。

1. 片内 RAM

8XC51，8XC31 芯片内部的 RAM 容量为 128 字节，根据用途可划分为工作寄存器区、位寻址区和用户数据存储器区（可作用户 RAM 和堆栈区），如表 2 – 1 所示。对于 8XC52/54/58 芯片来说，内部 RAM 的容量为 256 字节，分为 128 字节内部 RAM 和高 128 字节（80 ~ FFH）内部 RAM，可作为内部用户数据存储区和堆栈区，即 8XC52/54/58 芯片内部用

户 RAM 的容量多了 128 字节。但由于高 128 字节 RAM 的地址编码与特殊功能寄存器重叠，因此，只能通过寄存器间接寻址方式读写。

（1）工作寄存器区由 32 个字节组成，分为 4 个区，每区 8 个字节，分别用 R0 ~ R7 作为这 8 个字节的寄存器名。R0 的物理地址可能是 00H，也可能是 08H，10H 或 18H；同理，R1 的物理地址可能是 01H，也可能是 09H，11H 或 19H。任何时候只能选择 4 个工作寄存器区中的一个区作为当前工作寄存器区，当前工作寄存器区由程序状态字寄存器 PSW 的 b4，b3 位决定。具体情况如下：

PSW 寄存器 b4，b3 位	当前工作寄存器区	寄存器 R7 ~ R0 地址
00	0 区	07H ~ 00H
01	1 区	0FH ~ 08H
10	2 区	17H ~ 10H
11	3 区	1FH ~ 18H

复位后，PSW 的 b4，b3 位为 00，因此复位后将选择 0 区作为当前工作寄存器区。修改 PSW 的 b4，b3 位即可选择不同的工作寄存器区，这有利于快速保护现场，提高程序执行效率和中断的响应速度。

（2）20H ~ 2FH 单元，共 16 字节，属于位寻址区。该区域可以按字节读写，也可以按位读写。位地址从 20H 单元开始，共有 16 字节 ×8 位，即 128 个位地址（20H 单元 b0 位的位地址为 00H，20H 单元 b1 位的位地址为 01H，20H 单元 b2 位的位地址为 02H，依次类推，21H 单元 b0 位的位地址为 08H，2FH 单元 b7 位的位地址为 7FH），如表 2 – 1 所示。

表 2 – 1　内部 RAM 地址空间

									字节地址
高 128 字节内部 RAM									FF ~ 80H
用户 RAM 和堆栈区									7F ~ 30H
位寻址区（位地址）	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	2FH
	77H	76H	75H	74H	73H	72H	71H	70H	2EH
	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	2DH
	67H	66H	65H	64H	63H	62H	61H	60H	2CH
	5FH	5EH	5DH	5CH	5BH	5AH	59H	58H	2BH
	57H	56H	55H	54H	53H	52H	51H	50H	2AH
	4FH	4EH	4DH	4CH	4BH	4AH	49H	48H	29H
	47H	46H	45H	44H	43H	42H	41H	40H	28H
	3FH	3EH	3DH	3CH	3BH	3AH	39H	38H	27H

续表

									字节地址
高 128 字节内部 RAM									FF ~ 80H
用户 RAM 和堆栈区									7F ~ 30H
位寻址区（位地址）	37H	36H	35H	34H	33H	32H	31H	30H	26H
	2FH	2EH	2DH	2CH	2BH	2AH	29H	28H	25H
	27H	26H	25H	24H	23H	22H	21H	20H	24H
	1FH	1EH	1DH	1CH	1BH	1AH	19H	18H	23H
	17H	16H	15H	14H	13H	12H	11H	10H	22H
	0FH	0EH	0DH	0CH	0BH	0AH	09H	08H	21H
	07H	06H	05H	04H	03H	02H	01H	00H	20H
工作寄存器区	3 区（8 B）								1F ~ 18H
	2 区（8 B）								17 ~ 10H
	1 区（8 B）								0F ~ 08H
	0 区（8 B）								07 ~ 00H

如果系统中需要位操作，则最好保留 20H ~ 2FH 单元的部分或全部作为位存储区，以方便位操作。

（3）30H 单元以后可作为内部用户 RAM 区或堆栈区。对于 8XC31/8XC51 系列来说，从 30H ~ 7FH 尚有 80 B，可作用户内部 RAM 或堆栈区；对于 8XC32/8XC52/54/58 系列来说，从 30H ~ FFH 尚有 208 B，可作用户内部 RAM 或堆栈区。

复位后，堆栈指针 SP 指向 07H 单元，因此，一般需要修改，将 SP 设在 2FH 之上。

2. 特殊功能寄存器

由于单片机芯片内集成了一些常用的外围接口电路，如并行 I/O 端口、串行口、定时器/计数器、中断控制器等，因此这些外围接口电路中的控制寄存器、状态寄存器及数据寄存器也就位于芯片内，统称为特殊功能寄存器（Special Function Registers，SFR）。

MCS – 51 CPU 与通用微处理器不同，除了给外围接口电路寄存器，如定时/计数器控制寄存器 TCON 分配字节地址外，CPU 内的寄存器也有字节地址，如累加器 Acc 字节地址为 0E0H。增强型 MCS – 51 系列单片机内共有 32 个特殊功能寄存器（在标准 MCS – 51 系列单片机的基础上，增加了 6 个新的特殊功能寄存器），其地址分散在 80H ~ FFH 之间，如表 2 – 2 所示。

表 2－2　特殊功能寄存器地址映像

SFR 寄存器名	符号	位地址/位定义名								字节地址	复位后初值
		b7	b6	b5	b4	b3	b2	b1	b0		
* 累加器	Acc	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	E0H	00H
辅助功能寄存器	AUXR	–	–	–	–	–	–	–	A0	8E	xxxxxxx0B
辅助功能寄存器 1	AUXRl	–	–	–	–	GF2/WUPD	0	–	DPS	A2H	xxxx00x0B
* B 寄存器	B	F7	F6	F5	F4	F3	F2	F1	F0	F0H	00H
数据指针高 8 位	DPH									83H	00H
数据指针低 8 位	DPL									82H	00H
* 程序状态字	PSW	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	D0H	
		Cy	AC	F0	RS1	RS0	OV	–	P		
* 中断允许控制寄存器	IE	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	A8H	0x000000B
		EA	–	ET2	ES	ET1	EX1	ET0	EX0		
* 中断优先级控制寄存器	IP	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H	B8H	xx000000B
		–	–	PT2	PS	PT1	PX1	PT0	PX0		
中断优先级控制寄存器（高 8 位）	IPH	–	–	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	B7H	xx000000B
* I/O 端口 3（P3 口）	P3	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	B0H	FFH
		P3. 7	P3. 6	P3. 5	P3. 4	P3. 3	P3. 2	P3. 1	P3. 0		
* I/O 端口 2（P2 口）	P2	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	A0H	FFH
		P2. 7	P2. 6	P2. 5	P2. 4	P2. 3	P2. 2	P2. 1	P2. 0		
串行数据缓冲	SBUF									99H	不确定
* 串行控制	SCON	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	98H	00H
		SM0/FE	SM1	SM2	REN	TB8	RB8	T1	R1		
从地址寄存器	SADDR									A9H	00H
从地址掩蔽寄存器	SADEN									B9H	00H
* I/O 端口 1（P1 口）	P1	97H	96H	95H	94H	93H	92H	91H	90H	90H	FFH
		P1. 7	P1. 6	P1. 5	P1. 4	P1. 3	P1. 2	P1. 1	P1. 0		

续表

SFR 寄存器名	符号	位地址/位定义名								字节地址	复位后初值
		b7	b6	b5	b4	b3	b2	b1	B0		
时钟控制寄存器	CKCON	–	–	–	–	–	–	–	X2	8FH	xxxxxxx0B
定时器 T1 高 8 位	TH1									8DH	00H
定时器 T0 高 8 位	TH0									8CH	00H
定时器 T1 低 8 位	TL1									8BH	00H
定时器 T0 低 8 位	TL0									8AH	00H
定时/计数器方式控制寄存器	TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	89H	00H
* 定时/计数器控制	TCON	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	88H	00H
		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0		
定时器 T2 低 8 位	TL2									CCH	00H
定时器 T2 高 8 位	TH2									CDH	00H
定时器 T2 重装、捕获低 8 位	RCAP2L									CAH	00H
定时器 T2 重装、捕获高 8 位	RCAP2H									CBH	00H
* 定时/计数器 T2 控制寄存	T2CON	CFH	CEH	CDH	CCH	CBH	CAH	C9H	C8H	C8H	00H
		TF2	EXF2	RCLK	TCLK	EXE2N	TR2	C/$\overline{T2}$	CP/		
定时/计数器 T2 模式控制寄存器	T2MOD	–	–	–	–	–	–	T2OE	DCEN	C9H	xxxxxx00B
电源控制及波特率选择	PCON	SMOD1	SMOD0	–	POF	GF1	GF0	PD	IDL	87H	00xx0000
堆栈指针	SP									81H	07H
I/O 端口 0（P0 口）	P0	87H	86H	85H	84H	83H	82H	81H	80H	80H	FFH
		P0. 7	P0. 6	P0. 5	P0. 4	P0. 3	P0. 2	P0. 1	P0. 0		

说明：

- 带灰色背景寄存器或寄存器位为增强型 MCS－51 新增寄存器或寄存器位。
- 带＊号寄存器具有位寻址功能。
- 8XC5XX2 系列与 8XC5X 系列之间仅存在如下差异：

8XC5XX2 系列新增了时钟控制寄存器 CKCON，用于在运行中选择 CPU 工作模式。当 X2＝0 时，每机器周期包含 12 个时钟周期（与标准 MCS－51 相同）；而当 X2＝1 时，每机器周期包含 6 个时钟周期，即所谓的“X2”模式。

将 8XC5X 系列辅助功能寄存器 AUXR1 的通用标志位“GF2”作为“掉电唤醒”控制位 WUPD（即 Wake-up From Power Down 的简称）。在 8XC5XX2 系列中，当 WUPD（AUXR1.3）＝0 时，将禁止外中断唤醒掉电模式（复位后恢复到默认状态）。

- 寄存器中的保留位用“－”表示，不宜使用，初始化时只能写入 0。
- 从表 2－2 中可以看出：特殊功能寄存器地址分散在 80H～FFH 之间，对于没有相应寄存器与之对应的地址单元，不能对其进行读写操作。如果对空单元进行读操作，将得到一个不确定的值，即前后两次读出的数据不相同。写入时，数据将丢失，因为这些单元没有对应的物理存储器存放写入的数据。

下面对一些常见的特殊功能寄存器做一简单介绍。

（1）累加器 Acc。

CPU 内部特有的寄存器，常用于存放参加算术或逻辑运算的两个操作数中的一个及运算结果。例如：

```
ADD A，30H      ；在指令中，累加器 Acc 常简写为“A”
```

该指令的含义是以累加器 Acc 内容作为被加数，加数存放在内部 RAM 的 30H 单元中，相加后的结果，即和再存放到累加器 Acc 中。

由于早期的 CPU 没有乘法运算功能，乘法运算需要通过多次加法运算实现，而在多次加法运算中，寄存器 Acc 总是存放中间结果，相当于完成了累加的功能。因此，也就用“累加器”来称呼该寄存器。

（2）B 寄存器。

B 寄存器也是 CPU 内特有的一个寄存器，主要用于乘法和除法运算。在乘法运算中，被乘数放在累加器 Acc 中，乘数放在 B 寄存器中，积的高 8 位存放在 B 寄存器中，低 8 位放在累加器 Acc 中。如：

```
MUL   AB      ；BA←A×B
```

在除法运算中，被除数放在累加器 Acc 中，除数放在 B 寄存器中。运算后，商放在累加器 Acc 中，而余数放在 B 寄存器中。

（3）程序状态字寄存器 PSW。

程序状态字寄存器有时也称为“标志寄存器”，由一些标志位组成，用于存放指令运行

的状态。MCS – 51 系列单片机中 PSW 寄存器各位含义如表 2 – 3 所示。

表 2 – 3　MCS – 51 中 PSW 寄存器各位功能

b7	b6	b5	b4	b3	b2	b1	b0
Cy	AC	F0	RS1	RS0	OV	–	P

Cy：进位标志。在进行加法运算且当最高位（b7 位）有进位时，或执行减法运算且最高位有借位时，Cy 为 1；反之为 0。

AC：辅助进位标志。在进行加法运算且当 b3 位有进位，或执行减法运算且 b3 位有借位时，AC 为 1；反之为 0。

RS1，RS0：工作寄存器组选择位，前面已介绍过。

F0：用户标志位，可通过位操作指令将该位置 1 或清零。

PSW. 1：保留位。

OV：溢出标志。在计算机内，带符号数一律用补码表示。在 8 位二进制中，补码所能表示的范围是 – 128 ~ + 127，而当运算结果超出这一范围时，OV 标志为 1，即溢出；反之，为 0。

P：奇偶标志。该标志位始终体现累加器 Acc 中“1”的个数的奇偶性。如果累加器 Acc 中“1”的个数为奇数，则 P 位置 1；当累加器 A 中“1”的个数为偶数（包括 0 个）时，P 位为“0”。

例 2. 1　分析执行如下指令后，PSW 寄存器各标志位的状态。

```
MOV A，#10101101B   ；把立即数 0ADH 传送到累加器 A 中。由于立即数 0ADH 中共有
                    ；5 个“1”，因此该指令执行后，奇偶标志位 P 为“1”。
ADD A. #01111101B   ；0ADH 与 7DH 相加，结果存放在 A 中。
  10101101          ；173（无符号数），–83（带符号数）。
+ 01111101          ；125（无符号数），+125（带符号数）。
 1 00101010         ；作为无符号数时，和为 12AH（由于结果超出 FFH，前面的“1”
                    ；自然丢失，寄存器 A 的内容为 2AH），即 298；作为有符号数
                    ；时，和为 2AH，即 42。
```

由于 b7 位向前进位，因此 Cy 为“1”；b3 位也有进位，AC 位也为“1”；而作为带符号数时，结果为 42，没有超出 – 128 ~ + 127，OV 标志位为“0”；事实上，两个异号的数相加，结果不会溢出，OV 标志为“0”；而 A 中含有 3 个“1”，因此 P 标志位为“1”。

例 2. 2　分析执行如下指令后，PSW 寄存器各标志位的状态。

```
MOV A，#10101101B   ；把立即数 0ADH 送到累加器 A 中，由于立即数 0ADH 中共有 5
```

```
                                    ；个“1”，因此该指令执行后，奇偶标志位 P 为“1”。
ADD A，#10011101B    ；0ADH 与 9DH 相加，结果存放在 A 中。
  10101101           ；173（无符号数），－83（带符号数）。
+ 10011101           ；157（无符号数），－99（带符号数）。
1 01001010           ；作为无符号数时，和为 14AH（由于结果超出 FFH，前面的“1”
                     ；自然丢失，寄存器 A 的内容为 4AH），即 330；作为有符号数
                     ；时，和为－182。
```

由于 b7 位向前进位，因此 Cy 为“1”；b3 位也有进位，AC 位也为“1”。而作为带符号数时，结果为 4AH，即 +74，显然不对，因为两个负数相加结果为正数是不可能的。之所以出错，是因为 －83 加 －99 的结果为 －182，超出 －128 ~ +127，因此 OV 标志位为“1”。

两个同号数相加，结果可能溢出，而当 OV 标志位为“1”时，说明存放在目的操作数中的结果不正确！

不难发现：两个同号数相加，结果可能溢出；两个异号数相加，结果肯定不会溢出；两个同号数相减，结果肯定不会溢出；而两个异号数相减，结果可能溢出；而当溢出标志 OV 为 1 时，结果不正确。

（4）堆栈指针 SP。

在计算机内，需要一块具有“先进后出”（First In Last Out，FILO）特性的存储区，用于存放子程序调用（包括中断响应）时程序计数器 PC 的当前值，以及需要保存的 CPU 内各寄存器的值，即现场，以便子程序或中断服务程序执行结束后能正确返回主程序，这一存储区称为堆栈区。为了正确存取堆栈区内的数据，需要一个寄存器来指示最后进入堆栈的数据所在存储单元的地址，堆栈指针 SP 寄存器就是为此设计的。

在增强型 MCS－51 系列单片机中，SP 可以指向内部 RAM 中任一单元，且堆栈向上生长，即将数据压入堆栈后，SP 寄存器内容增大。假设 SP 当前值为 2FH，则入堆指令“PUSH B”（将寄存器 B 内容压入堆栈）的执行过程如图 2－13 所示。

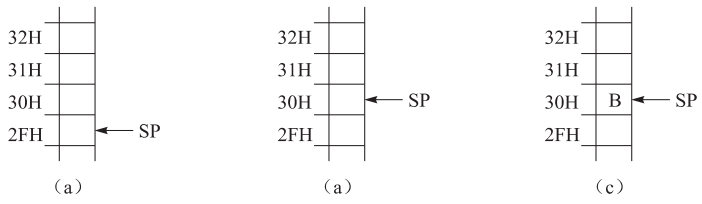

图 2－13　“PUSH B”指令的执行过程

（a）PUSH B 指令执行前；（b）SP 加 1；（c）寄存器 B 存入 SP 指定的单元中

将数据从堆栈中弹出时，SP 减小。例如，将保存在堆栈中的信息弹到寄存器 B 的操作

过程如图 2 – 14 所示。

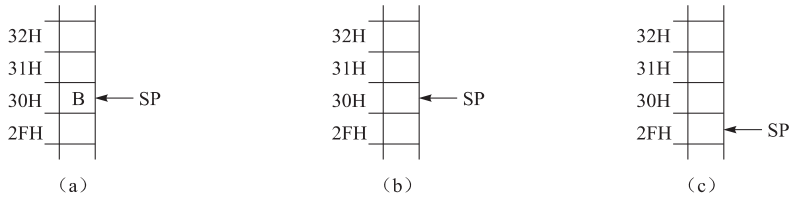

图 2 – 14　“POP B”指令的执行过程

（a）POP B 指令执行前；（b）将 SP 指定单元内容传送到寄存器 B 中；（c）SP 减 1

数据入栈的操作过程为：先将 SP 加 1（SP←SP + 1），然后将要入栈的数据存放在 SP 指定的存储单元中。将数据从堆栈中弹出时，先将 SP 寄存器指定的存储单元内容传送到 POP 指令给定的寄存器或内部 RAM 单元中，然后 SP 减 1（SP←SP – 1）。

可以看出堆栈的底部是固定的，而堆栈的顶部则随着数据入栈和出栈而上下浮动。

系统复位后，PSW 的 b4，b3 位为 0，0，即选择了工作寄存器区中的 0 区作为当前工作寄存器区，SP 寄存器的初值为 07H。当有数据进入堆栈时，将从 08H 单元开始存放是不允许的。因为 08 ~ 1FH 属于工作寄存器区，不宜占用；20 ~ 2FH 是位地址区，也需要部分或全部保留。因此，必须通过数据传送指令重新设置 SP 的初值，将堆栈底部设在 30 ~ 7FH（对于只有 128 字节内部 RAM 的 8XC31/8XC51）或 80 ~ FFH（对于具有 256 字节内部 RAM 的 8XC32/8XC52/54/58）之间。如：

```
MOV SP, #5FH        ; 将堆栈设在 60H 单元之后。
```

CPU 内 30H ~ FFH 单元既可以作为堆栈区，同时也是用户数据存储区。数量有限，必须充分利用，因此，将堆栈底部设在何处，必须认真考虑。随着入栈数据的增多，对于仅有低 128 字节内部 RAM 的 80C31，80C51 来说，当 SP 超出 7FH 时发生上溢，这将出现不可预料的后果。因此，在设置 SP 初值时，必须考虑堆栈最大深度。子程序或中断嵌套层数越多，所需的堆栈深度就越大。为了避免堆栈顶部进入用户数据存储区，造成混乱，一般将堆栈设在用户数据存储区之上，如在某一应用系统中，需要 32 个字节作为用户数据存储区（如 30 ~ 4FH），则初始化时将堆栈底部设在 50H，即堆栈深度为 48 个字节（50 ~ 7FH）。

```
MOV SP, #4FH        ; SP 初值为 4FH。
```

对于具有高 128 字节的 8XC32，8XC52/54/58 等 CPU 来说，最好将堆栈区设在 80 ~ 0FFH 之间的高 128 字节内部 RAM 中，而将具有直接寻址功能的低 128 字节内部 RAM 作为用户数据区，以便可用多种寻址方式存取用户数据。当然，SP 也不允许超出 0FFH，否则同样发生上溢。例如，预计某系统所需最大堆栈深度为 32 字节，可通过如下指令将栈底设在 0E0H 处。

```
MOV SP, #0DFH      ; SP 初值为 0DFH。
```

涉及入栈出栈操作的指令有：

```
PUSH direct        ; 将内部 RAM 单元压入堆栈。
POP direct         ; 从堆栈中将数据弹入内部 RAM 单元。
```

（5）数据指针 DPTR。

数据指针 DPTR 是一个 16 位的专用寄存器，由 DPH（数据指针高 8 位）和 DPL（数据指针低 8 位）组成，用于存放外部数据存储器单元地址。由于 DPTR 是 16 位的寄存器，因此通过 DPTR 寄存器间接寻址方式可以访问 0000 ~ FFFFH 全部 64 KB 的外部数据存储器空间。例如，可用如下指令将累加器 A 的内容传送到外部数据存储器的 107FH 单元中。

```
MOV DPTR, #107FH   ; 将外部数据寄存器地址 107FH 以立即数方式送到 DPTR 寄
                   ; 存器。
MOVX @ DPTR, A     ; 将累加器 A 的内容传送到 DPIR 寄存器内容指定的外部数据
                   ; 存储器单元中。
```

为了方便外部 RAM 之间的数据块传送，增强型 MCS－51 单片机采用双数据指针，由辅助功能寄存器 1（AUXR1）控制，该寄存器各位含义如表 2－4 所示。

表 2－4　辅助功能寄存器各位含义（AUXR1）

AUXR1（字节为 0A2H）

b7	b6	b5	b4	b3	b2	b1	b0
－	－	－	－	GF2	0	－	DPS

其中：

GF2——可作为用户标志位。

DPS——数据指针切换位。当 DPS＝0 时，DPTR 寄存器对应物理指针 DPTR0；当 DPS＝1 时，DPTR 寄存器对应物理指针 DPTR1。

b2 位恒为 0，且不能写入。这样就可以通过 INC AUXR1 指令快速切换数据指针，而不影响 GF2 标志（由于 b2 位不能写入，加 1 操作时，b1 向 b2 进位将自动丢失，影响不到 b7 ~ b3 位）。

（6）辅助功能寄存器 AUXR。

在增强型 MCS－51 单片机中，新增了辅助功能寄存器 AUXR，该寄存器各位含义如表 2－5 所示。

表 2 –5　辅助功能寄存器（AUXR）各位含义

AUXR（字节为 8EH）

b7	b6	b5	b4	b3	b2	b1	b0
–	–	–	–	–	–	–	A0

用于禁止/允许地址锁存信号 ALE 输出。当 A0 为 0 时，允许 ALE 输出；而当 A0 为 1 时，除了执行“MOVX”或“MOVC”指令外，禁止 ALE 输出，降低了电磁辐射量。当使用片内程序存储器时，调试结束后，最好通过“ORL AUXR，#01H”指令关闭 ALE 输出（但在仿真调试时，不宜关闭，原因是多数仿真机依靠 ALE 工作）。

（7）I/O 端口寄存器。

P0，P1，P2，P3 口寄存器实际上就是 P0 ~ P3 口对应的 I/O 端口锁存器，用于锁存通过端口输出的数据。

在增强型 MCS –51 单片机中，特殊功能寄存器分别隶属于 CPU 内不同的单元电路，具体如下：CPU 单元包含的寄存器有 Acc，B，SP，PSW，DPTR，AUXR、AUXR1 和程序计数器 PC。

PC 是一个 16 位的地址寄存器，用于存放当前指令码在程序存储器中的地址，但 PC 不属于特殊功能寄存器，它没有物理地址。

另外，MCS –51 系列单片机有些特别，CPU 单元内的寄存器，如累加器 A、寄存器 B、堆栈指针 SP、程序状态字 PSW 等均有字节地址。

但多数 CPU，如 PIC 系列单片机 CPU，Motorola 的 68 系列 CPU，以及 Intel 公司 X86 通用微处理器内，这些寄存器均没有字节地址，只能使用寄存器寻址方式读写。

定时/计数器单元包含的寄存器：TMOD，TCON，T2CON，T2MOD，TH0 与 TL0（分别是定时器 T0 的高 8 位和低 8 位）、TH1 与 TL1（分别是定时器 T1 的高 8 位和低 8 位）、TH2 与 TL2（分别是定时器 T2 的高 8 位和低 8 位），以及定时器 T2 的重装/捕捉寄存器 RCAPL2，RCAPH2。

并行 I/O 端口寄存器有 P0 ~ P3。

中断单元电路内的寄存器有 IE，IP，IPH。

串行通信单元电路内的寄存器有 SCON，SBUF，PCON，SADDR，SADEN。

3. 内部数据存储器之间的数据传送及寻址方式

（1）00 ~ 7FH 前 128 字节内部 RAM 存储器，可使用直接寻址方式或寄存器间接寻址方式读写。如：

```
MOV 30H，40H      ；内部 RAM 40H 单元内容送 30H 单元。
MOV 30H，#35H     ；将立即数 35H 写入内部 RAM 30H 单元。
```

所谓直接寻址，就是在指令中直接给出内部 RAM 单元的地址编码。

MOV @R0，#35H　；寄存器间接寻址方式。

该指令将立即数35H写入由R0寄存器内容指定的内部RAM单元中。如果该指令执行前，R0内容为30H，则上述两条指令执行后，结果相同，均把立即数35H写入内部RAM 30H单元中。所谓寄存器间接寻址，就是将内部RAM地址存放在寄存器R0或R1中。

（2）对于特殊功能寄存器只能使用直接寻址方式访问。如：

MOV P1，#35H　；立即数35H送P1口锁存器。

MOV B，#35H　；立即数35H送CPU内寄存器B。

对于特殊功能寄存器来说，用直接寻址和寄存器名寻址没有区别。汇编时，汇编程序将通过查表方式将寄存器名换成直接地址。但必须注意，引用特殊功能寄存器名仅是为了提高程序的可读性，与寄存器寻址方式不同。例如：

MOV Acc，#35H　；用特殊功能寄存器名，在本质上目的操作数属于直接寻址，该指令操作码为：75H，E0H，35H。其中75H是操作码，E0H是Acc的地址，35H是立即数。

MOV A，#35H　；寄存器寻址，操作码为74H，35H。同样，35H也是立即数。

（3）高128字节（80～FFH）内部RAM的访问。对于具有256字节内部RAM的MCS－51芯片来说，高128字节内部RAM地址空间与特殊功能寄存器的地址重叠，读写时需通过不同的寻址方式加以区别。规定用寄存器间接寻址方式访问高128字节（80～FFH）内部RAM，用直接寻址方式访问特殊功能寄存器。如在MCS－51中，“MOV 90H，#35H”指令的含义是将立即数35H写入P1口锁存器中，与“MOV P1，#35H”含义相同，而不是把立即数35H写入内部RAM的90H单元中。将立即数35H传送到内部RAM的90H单元中只能通过如下指令进行：

MOV R0，#90H　；将内部RAM地址90H写入R0寄存器中。

MOV @R0，#35H　；将立即数35H写入由R0指定的内部RAM单元中。

（4）位寻址区。MCS－51系列单片机既是8位机，同时也是一个功能完善的一位机。作为一位机时，它有自己的CPU、位存储区（位于内部RAM的20～2FH单元）、位寄存器，将进位标志Cy作为“位累加器”，以及完整的位操作指令，包括置1、清零、非（取反）、与、或、传送、测试转移等。

对于位存储器（20～2FH单元中的128个位），只能使用直接寻址方式确定操作数所在的存储单元。如：

MOV C，23H　；位传送指令，即将位地址23H单元（对应24H字节单元的b3位）；内容传送到位累加器C中。

CLR 23H　；位清零指令，即将位地址23H单元清零。

SETB 23H　；位置1指令，即将位地址23H单元置1。

CPL 23H　；位取反操作，即将位地址23H单元内容取反。

```
ORL C，23H      ；或运算，23H 位单元与位累加器 C 相或，结果存放在位累加器
                ；C 中。
ANL C，23H      ；与运算，23H 位单元与位累加器 C 相与，结果存放在位累加器
                ；C 中。
```

对于具有位地址的特殊功能寄存器中的位，除了使用位地址寻址外，还可以使用“位定义名”或“寄存器名．位”表示。例如将程序状态字寄存器 PSW 中的 b3 位置 0 可以用：

```
CLR D3H     ；位地址方式。
CLR RS0     ；位定义名方式。作为一个良好的习惯，建议使用位定义名方式。
CLR PSW.3   ；“寄存器名．位”方式。
```

尽管书写形式不同，但汇编时，汇编程序均自动转换为“位地址”方式，因此这三条指令完全等效，不过使用“位定义名”和“寄存器名．位”方式直观，容易理解。

“寄存器名．位”表示方式不仅适用具有位地址的特殊功能寄存器，也适用于具有位地址的内部 RAM 单元。如果使用了“BYTEBIT DATA 20H”伪指令定义了 BYTEBIT 变量，则如下三条指令也完全等效。

```
MOV C，BYTEBIT.1
MOV C，20H.1
MOV C，01H
```

2.4.3　外部数据存储器

通过 P0，P2 口最多可以连接 64KB 的外部数据存储器。有关外部数据存储器的连接及读写方式参阅“外存储器连接”部分。

2.5　操 作 时 序

计算机的操作就是逐条执行已编好且存储在存储器中的指令，计算机的工作就是不断重复“取指→译码→执行”过程。所有这些操作都按一定顺序进行，因此在计算机内需要向 CPU 输入精确、稳定的时钟信号，以实现定时。

MCS –51 系列单片机一个机器周期由 12 个振荡周期组成，分为 6 个状态，分别称为 S1，S2，S3，S4，S5，S6，每个状态都包含 P1，P2 两相，如图 2 –15 所示。

振荡周期，也就是时钟周期，它是输入时钟信号频率 f_{OSC}的倒数。如果时钟信号或晶体振荡器的频率为 12 MHz，则振荡周期 T = 1/12 MHz = 83 ns。

状态周期，即 CPU 从一个状态转换到另一状态所需的时间称为状态周期，一个状态周期由一个或一个以上的时钟周期组成。在 MCS –51 系列单片机中，一个状态周期由两个时

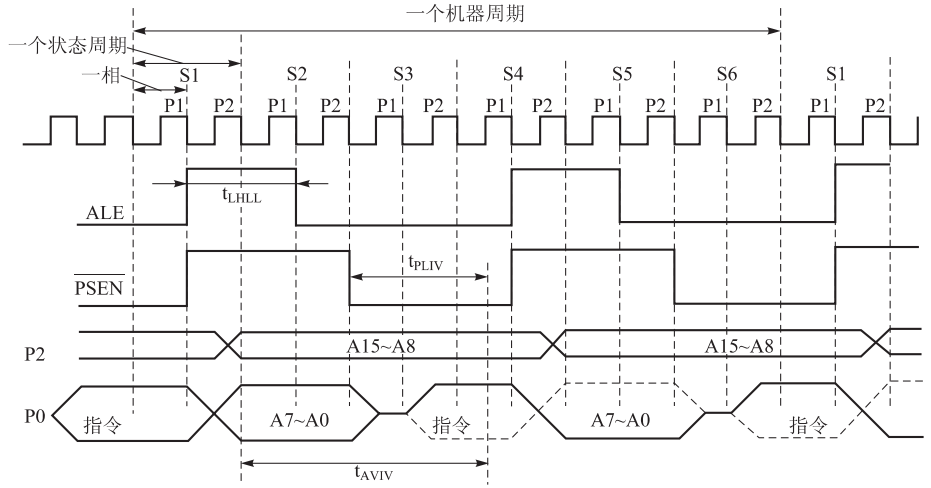

图 2－15　MCS－51 单片机外部程序存储器读时序

钟周期组成。

机器周期指的是计算机完成一次完整的、基本的操作所需要的时间，MCS－51 系列单片机的一个机器周期由 6 个状态周期组成，共 12 个振荡周期。

执行一条指令所需的时间称为指令周期，指令周期往往由一个或一个以上的机器周期组成，指令周期的长短与指令所执行的操作有关。简单指令，如 INC A 等仅需一个机器周期；而复杂指令，如乘法、除法指令则需要几个机器周期。

2.5.1　对外部程序存储器的读操作时序

MCS－51 系列单片机对外部程序存储器的读操作时序如图 2－15 所示。S1P2 相开始后，地址锁存信号 ALE 有效，经过一个振荡周期 T 的延迟后，在 S2P1 开始时刻，P0，P2 口分别送出低 8 位地址信息和高 8 位地址信息（当前指令码所在的程序存储器单元地址），再经过一个振荡周期，待 P0 口地址信息稳定后，ALE 由高电平变为低电平，将 P0 口输出的低 8 位地址信息（A7～A0）锁存在 74LS373 锁存器中。因此，ALE 信号有效时间（ALE 信号脉冲宽度为 t_{LHLL}）为 2T。

ALE 下降沿过后，再经过一个振荡周期，即在 S2P2 结束时刻，P0 地址信息消失，因此 ALE 无效后，P0 口地址信息保存时间等于 T。

外部程序存储器读选通信号$\overline{\text{PSEN}}$在 S2P2 相结束后，开始有效，以选通外部程序存储器芯片（该信号一般接 EPROM 芯片的输出允许端$\overline{\text{OE}}$），并保持到 S4P1 相结束时刻，即$\overline{\text{PSEN}}$有效时间为 3T。在$\overline{\text{PSEN}}$由低电平变为高电平前，CPU 读取 P0 口引脚的信息（指令码），而不管程序存储器芯片是否已将数据送到 P0 口引脚。因此，程序存储器芯片的速度必须足够

快，否则不能及时将数据输出到 P0 口引脚。

MCS – 51 系列单片机在一个机器周期内，可以从程序存储器中读出两个字节的指令码，从 S5P1 相开始，P0，P2 口分别输出下一个存储单元的地址信息，重复取指过程。因此，在访问外部程序存储器时，一个机器周期内，ALE 信号及$\overline{\text{PSEN}}$信号出现两次，即在一个机器周期内，可以读取两个字节的指令码。对于单字节指令来说，第一次读出指令码后，PC 保持不变，并自动丢弃在下一周期的 S1P1 时刻读出的指令码；对于双字节单周期指令来说，第一次读操作获得指令第一操作码后，PC 自动加 1，在下一机器周期的 S1P1 时刻读出指令第二操作码。

可见，为了保证在$\overline{\text{PSEN}}$上升沿前程序存储器将数据输出到 P0 引脚，必须确保：

（1）存储器地址有效到数据输出有效的时间 t_{ACC}必须小于等于 CPU 地址有效到采样 P0 口数据时间 t_{AVIV}，即要求 $t_{ACC} \leqslant t_{AVIV} = 5T - t$（地址有效到$\overline{\text{PSEN}}$无效的时间为 5T，但 CPU 在$\overline{\text{PSEN}}$无效前采样 P0 口数据，t 即为采样数据到$\overline{\text{PSEN}}$无效的间隔，t 的大小可从 MCS – 51 系列单片机技术手册中查到）。

（2）存储器$\overline{\text{OE}}$有效到数据输出有效的时间 t_{OE}必须小于等于$\overline{\text{PSEN}}$有效到 CPU 采样 P0 口数据时间 t_{PLIV}，即要求 $t_{OE} \leqslant t_{PLIV} = 3T - t$。

当由单片存储器芯片组成程序存储器时，片选信号$\overline{\text{CE}}$接地，一直处于有效状态，不用考虑存储器片选信号有效到数据输出有效的时间 t_{CE}；而当程序存储器由两个或两个以上的芯片组成时，片选信号$\overline{\text{CE}}$由高位地址译码产生，片选信号$\overline{\text{CE}}$有效到采样 P0 口数据的时间等于 t_{AVIV}。但存储器芯片技术参数中的 $t_{CE} \leqslant t_{ACC}$，因此只要满足 $t_{ACC} \leqslant t_{AVIV}$，即满足 $t_{CE} \leqslant t_{AVIV}$，而不用考虑 t_{CE}问题。

2.5.2　外部数据存储器读写时序

在 MCS – 51 系列单片机中，通过 MOVX 指令读写外部数据存储器，包括：

```
MOVX A，@Ri        ；将 Ri 寄存器指定的外部 RAM 单元（低 8 位外部数据存储器
                   ；地址存放在 Ri 寄存器中，寻址范围为 00H ~ FFH）内容传送
                   ；到累加器 A 中（读操作）。
MOVX @Ri，A        ；将累加器 A 中的内容传送到由 Ri 寄存器指定的外部 RAM 单
                   ；元中（写操作）。
MOVX A，@DPTR      ；将 DPTR 寄存器指定的外部 RAM 单元（16 位外部数据存储
                   ；器地址存放在外部数据地址寄存器 DPTR 中，寻址范围为
                   ；0000H ~ FFFFH）内容传送到累加器 A 中（读操作）。
MOVX @DPTR，A      ；将累加器 A 中的内容传送到由 DPTR 寄存器指定的外部 RAM
                   ；单元中（写操作）。
```

在读写外部数据存储器时，分别由$\overline{RD}$和$\overline{WR}$信号选通外部数据存储器，操作时序如图2－16所示。

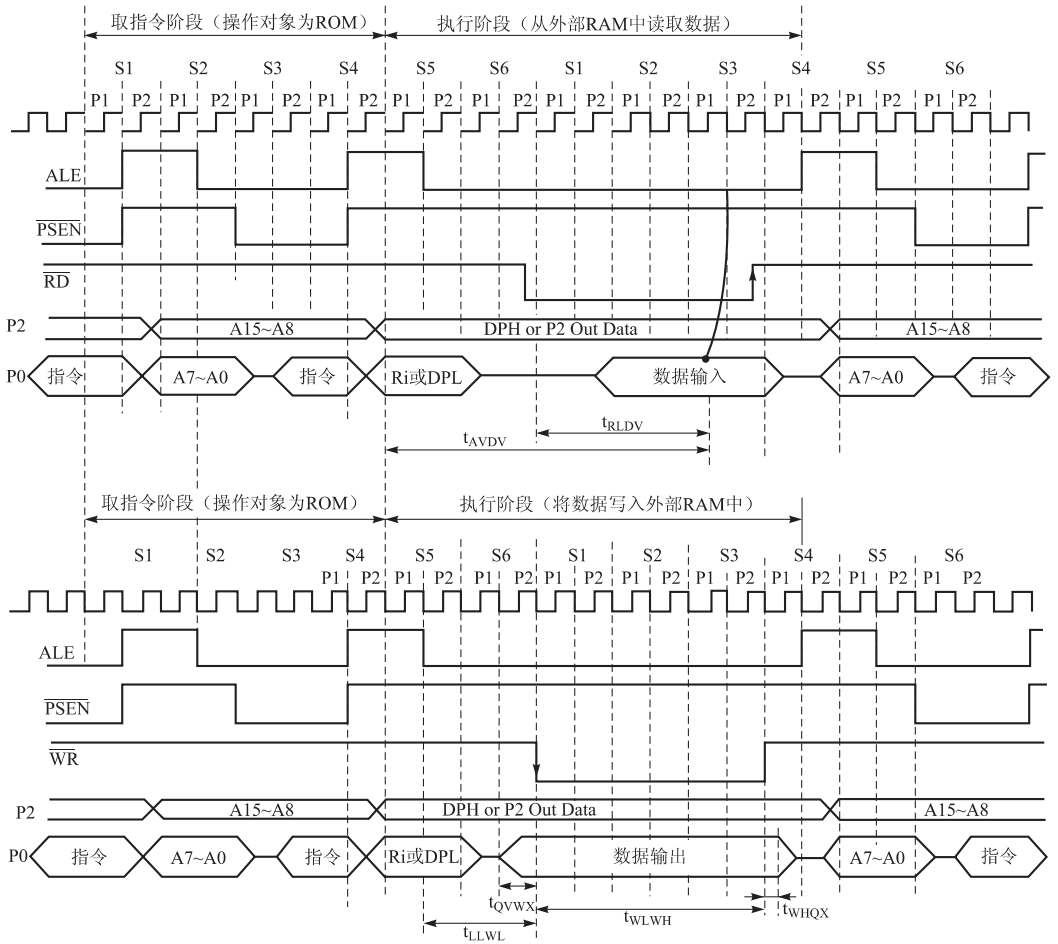

图2－16　MCS－51外部数据存储器读、写时序

1. 外部数据存储器读操作时序

“MOVX A，@DATR”或“MOVX A，@Ri”指令属于单字节双周期指令，操作时序如图2－16所示。在第一个机器周期内，取出指令码，操作时序与程序存储器读时序相同，在第一个机器周期的S5P1相后，进入指令的执行阶段。对于MOVX A，@DPTR指令来说，外部数据存储器低8位地址存放在DPL寄存器中，这时P0口输出的信息就是DPL寄存器内容；高8位地址存放在DPH寄存器中，P2口输出DPH寄存器内容。对于MOVX A，@Ri指令来说，外部数据存储器低8位地址存放在Ri寄存器中，通过P0口输出，而P2口输出

内容是 P2 口锁存器的内容，保持不变。

在第一个机器周期的 S5P1 结束时刻，ALE 由高电平变为低电平，将 P0 口输出的外部数据存储器低 8 位地址信息锁存在锁存器中。

在第一个机器周期的 S6P2 结束时刻，外部数据存储器的读选通信号$\overline{RD}$有效（该信号接外部数据存储器的输出允许端$\overline{OE}$），并保持到第二个机器周期的 S3P2 结束时刻，因此$\overline{RD}$有效时间为 6T。CPU 在第二个机器周期的 S3P1 结束前，读 P0 口引脚，将外部数据存储器数据总线上的信息传到 CPU 内部数据总线。因此，外部数据存储器必须在 S3P1 结束前将数据送到数据总线上，即 P0 口引脚，以便 CPU 在读外部数据存储器控制信号$\overline{RD}$上升沿来到前读出数据总线上的信息。

在扩展输入口电路中，控制输入芯片数据总线与 CPU 数据总线连通的三态门控制信号$\overline{OE}$由 CPU 高位地址译码输出信号和 CPU 外部数据存储器读控制信号$\overline{RD}$经过或门得到，$\overline{RD}$信号延迟时间约为两个门电路的延迟时间（20 ns），完全满足要求。

可见外部数据存储器地址有效到读引脚数据时间 $t_{AVDV} = 9T - X$；$\overline{RD}$有效到读引脚数据时间 $t_{RLDV} = 5T - X$；$\overline{RD}$无效到 ALE 有效的时间为 1T。

因此，MCS –51 与外部数据存储器连接时，必须保证：

（1）$t_{ACC} \leqslant t_{AVDV} = 9T - X$（地址有效到 CPU 读外部 RAM 数据时间，X 的大小可从 MCS –51 单片机技术手册中查到）。

（2）$t_{OE} \leqslant t_{RLDV} = 5T - X$。

（3）t_{OHZ}（$\overline{OE}$无效到存储器芯片数据总线变为高阻态的时间）小于等于 1 T。

（4）地址有效到写操作结束时间 $t_{AIV} \leqslant 10$ T。

（5）数据有效到写操作结束时间 $t_{DIV} \leqslant 7$ T。

一般采用静态 RAM（SRAM）作为外部数据存储器，而 SRAM 读写速度比 EPROM 快，且读写外部 RAM 时间比读外部 ROM 长。因此，当时钟信号频率小于等于 12 MHz 时，所有静态 RAM 均满足要求。

2. 外部数据存储器写操作时序

外部数据写操作时序与读操作时序相似，外部数据存储器地址信息锁存后，在第一个机器周期的 S6P1 结束时，写数据就出现在 P0 口引脚上；在 S6P2 结束时刻外部数据存储器写选通信号$\overline{WR}$有效（该信号接外部数据存储器芯片的写允许信号$\overline{WE}$），启动写操作（写数据有效到$\overline{WR}$有效时间 $t_{QVWX} = T - X$），$\overline{WR}$保持时间 $t_{WLWH} = 6$ T。因此，地址有效到写操作结束时间为 10 T（地址有效到$\overline{WR}$有效时间 + $\overline{WR}$有效时间）。

而$\overline{WR}$无效后数据维持时间 $t_{WHQX} = T - X$，小于一个时钟周期，当时钟频率高时，t_{WHQX} 将小于一个门电路的延迟时间，因此对外部数据存储器（包括扩展 I/O 口）写入时，只能利用 CPU 外部数据存储器写选通信号$\overline{WR}$的下降沿（前沿）作为外部 RAM、扩展输出口芯片数据输入锁存信号，一般不能利用$\overline{WR}$的上升沿（后沿）作为数据输入锁存信号。原因是

锁存信号由 CPU 高位地址译码输出信号和 CPU 外部数据存储器写控制信号$\overline{WR}$经过或非门得到，当$\overline{WR}$信号传送到输出口扩展芯片（如 74LS273）的锁存信号 CLK 引脚时存在延迟（延迟时间与经过门电路的级数、PCB 板上$\overline{WR}$信号连线长度有关），$\overline{WR}$上升沿来到时，CPU 数据总线上的输出数据很可能已无效。

2.5.3 6 时钟/机器周期模式下的时序

8XC5XX2，89C6XX2 芯片每机器周期包含的时钟周期由时钟选择寄存器 CKCON 的 X2 位和位于 Flash ROM 保密块中的时钟配置位 FX2 控制，如表 2－6 所示。这样通过修改时钟选择寄存器 CKCON 的 X2 位或保密块中的时钟选择位 FX2 来选择“6 时钟”或“12 时钟”运行模式。

表 2－6 时钟配置

FX2 位状态（位于 Flash ROM 保密字节内）	X2 位状态（CKCON. O）	CPU 时钟
擦除（未编程）	0（默认）	12 时钟
擦除（未编程）	1	6 时钟
编程	X（无效）	6 时钟

从表 2－6 可以看出位于 Flash ROM 保密字节内的系统时钟配置位 FX2 比 CKCON 寄存器内的 X2 位优先，即当 FX2 位被编程（可通过并行编程器编程或擦除）后，X2 位无效，系统运行在“6 时钟”模式。

当 FX2 位未被编程时，将 CKCON 寄存器的 X2 位置 1 时，系统由 12 时钟/机器周期模式切换到 6 时钟/机器周期模式。在这种情况下，时序图中各信号出现顺序不变，但时间间隔与 12 时钟/机器周期标准模式相比将减小一半，指令执行时间只有原来的 1/2。因此，在 6 时钟/机器周期模式下，扩展外部存储器或 I/O 端口时，必须注意外部存储器芯片存取速率能否满足要求，否则必须降低时钟频率。

习 题 2

1 8051 单片机内部包含哪些主要功能部件？

2 微处理器又称什么？由几部分组成？运算器具有哪些功能？

3 MCS－51 系列单片机有多少管脚？作为 I/O 口的管脚有哪些？说明 RST，XTAL1，XTAL2 的作用。说明 P3 口各管脚的第二功能。

4 简述$\overline{EA}$引脚有什么作用？8031 的$\overline{EA}$引脚应如何处理，为什么？

5　PC 是什么寄存器？它有什么作用？它是 8 位还是 16 位寄存器？

6　DPTR 是什么寄存器？它由哪两个特殊功能寄存器组成？其主要作用是什么？

7　单片机的振荡周期、状态周期、机器周期有什么关系？当晶振频率为 6 MHz 时，单片机的振荡周期、状态周期、机器周期各为多少？

8　单片机的复位有哪几种方法？复位后，对内部 RAM 有何影响？寄存器 A，B，PSW，SP，PC，DPTR 等的状态如何？

9　画出 8051 的复位电路，并说明复位原理。

10　简述 P1 口的内部结构。为什么将 P1 口引脚作为输入引脚使用前，一定要向 P1 口锁存器相应位写入“1”？

11　MCS – 51 系列单片机的存储器有哪几类？在使用时如何区别？

12　片内数据存储器简称什么？8051 片内数据存储器的地址空间在什么范围，又分为几个部分？

13　如何确定工作寄存器所在的区域？

14　可以位寻址的内部 RAM 区域是什么范围？可以位寻址的特殊功能寄存器有何特性？

15　写出位地址为 07H，10H，23H，50H 和 90H，0A3H，0F1H，0E6H 所在的内部 RAM 单元或特殊功能寄存器单元。

16　PSW 是什么寄存器？它的各个位含义如何？若（PSW）= 10H，则当前 R0 ~ R7 在内部 RAM 的哪些单元？

第 3 章

MCS－51 单片机的指令系统

在前面各章学习中，我们已经对单片微型计算机的内部结构和工作原理有了一个基本了解，在此基础上，本章将进一步介绍 MCS－51 单片机指令系统的指令格式、分类和寻址方式，并重点阐述 MCS－51 指令系统中每条指令的功能和特点，以及汇编语言程序设计的一些基本概念及注意事项。

3.1 指 令 格 式

单片机要执行某种操作，用户必须按照格式编写指令，单片机才能识别并准确操作。指令的编码规则称为指令格式。

3.1.1 指令的格式

1. 指令的一般格式

MCS－51 单片机指令的一般格式为：操作码　操作数。例如指令：74H 30H。

（1）操作码：用来表示执行什么样的操作。例如传送、加、减等。MCS－51 系列单片机的操作码为 8 位二进制的机器码，在指令中为第一字节。用机器码写成的指令是机器指令，也称为指令代码。

（2）操作数：表示参与操作的数据或数据的存储地址。不同类型的指令，操作数的个数是不一样的，可以有 3 个、2 个、1 个等。在具有多个操作数的指令中，把它们分别称为第一操作数、第二操作数等。如果操作数是一个直接参加操作的数据，这种操作数称为立即数；而大部分操作数存放于寄存器或数据存储器的某个存储单元，操作数字段仅指出操作数所在的寄存器或存储器地址。

2. 常用指令格式

编写指令时，要记住各种由“0”和“1”二进制数组成的代码和他们的含义是很困难的，既容易出错，又不易检查。所以常用的指令格式是以助记符表示的符号指令，也称汇编语言，由标号、操作码助记符、操作数和注释 4 个字段组成，格式如下：

［标号：］操作码助记符［操作数 1］［，操作数 2］［；注释］

其中，方括号内的项为任选项，需要此项时，指令中不写方括号；两操作数之间应以逗号分开。

例如指令：　　START：　　MOV A，　　#79H　　；A←79H

（1）标号：标号是用户定义的符号，由以字母开始的 1 ~ 8 个字符（字母或数字）组成，它代表指令的符号地址，通常在程序分支、转移等所需要的地方加上一个标号，并不是每条指令都必须有标号。当将指令转换成机器指令时，指令第一字节（也称首字节）的存储单元地址值赋给该标号。

（2）操作码助记符：助记符是一些代表操作含义的英文缩写，一般由 2 ~ 5 个英文字母组成，如“MOV”表示“传送”、“ADD”表示“加”等。操作码助记符对应的机器码是指令的第一字节，也是指令不可缺少的部分。

（3）操作数：与机器指令格式中的操作数相似。

（4）注释：注释是对本指令或本段程序的功能说明，便于对程序的阅读理解，在转换成机器指令时不予考虑。注释的前面需加分号“;”。

单片机识别机器指令，编程人员使用符号指令，机器指令与符号指令之间有一一对应的关系，绝没有重复。各种指令的机器码不需要记忆，编程人员可查阅机器指令与符号指令的映射表将符号指令译成机器指令，这个过程称为汇编；但更多的是用专门的软件来完成汇编过程。

3.1.2　指令的分类

MCS – 51 指令系统有 33 种操作功能。指令助记符与寻址方式组合，得到 111 种指令。分类如下：

1. 按字节数分类

（1）单字节指令，有 49 条。

（2）双字节指令，有 45 条。

（3）3 字节指令，有 17 条。

2. 按指令执行时间分类

（1）单周期指令，有 64 条。

（2）双周期指令，有 45 条。

（3）四周期指令，乘、除各有 1 条。

3. 按功能分类

（1）数据传送指令，有 28 条。

这类指令主要用于单片机片内 RAM 和特殊功能寄存器 SFR 之间传送数据，也可以用于单片机片内和片外存储单元之间传送数据。数据传送指令是把源地址中操作数传送到目的地

址（或目的寄存器）的指令，在该指令执行后源地址中的操作数不被破坏。源操作数有 8 位和 16 位之分，前者称为 8 位数传送指令，后者叫做 16 位数传送指令。

交换指令也属于数据传送指令，是把两个地址单元中内容相互交换。因此，这类指令中的操作数或操作数地址是互为“源操作数”和“目的操作数”的。

（2）算术运算指令，有 24 条。

算术运算指令用于对两个操作数进行加、减、乘、除等算术运算。在两个操作数中，一个应放在累加器 A 中，另一个可以在某个寄存器或片内 RAM 单元中，也可以放在指令码的第二和第三字节中。指令执行后，运算结果便可保留在累加器 A 中，运算中产生的进位标志、奇偶标志和溢出标志等皆可保留在 PSW 中。参加运算的两数可以是 8 位的，也可以是 16 位的。

（3）逻辑运算和环移指令，有 25 条。

这类指令包括逻辑运算和环移两类指令。逻辑操作指令用于对两个操作数进行逻辑乘、逻辑加、逻辑取反和异或等操作，大多数指令也需要把两个操作数中的一个预先放入累加器 A，操作结果也在累加器 A 中。环移指令可以对累加器 A 中的数进行环移。环移指令有左环移和右环移之分，也有带进位位 Cy 和不带进位位 Cy 之分。

（4）位操作指令，有 12 条。

位操作指令又称布尔变量操作指令，共分为位传送、位置位、位运算和位控制转移指令等 4 类。其中，位传送、位置位和位运算指令的操作数不是以字节为单位进行操作的，而是以字节中某位为单位进行的；位控制转移指令不是以检测某个字节的结果为条件而转移的，而是以检测字节中的某一位的状态来转移的。

（5）控制转移指令，有 22 条。

控制转移指令分为条件转移、无条件转移、调用和返回等指令。这类指令的共同特点是可以改变程序执行的流向，或者是使 CPU 转移到另一处执行，或者是继续顺序地执行。无论是哪一类指令，执行后都会改变程序计数器 PC 中的值。

3.1.3 指令的存放空间

指令是单片机执行某种操作的命令，用户若要单片机完成一件事情，必须先编写指令，再转成机器码形式，从键盘等设备输入到程序存储器存放。存放在程序存储器的哪个空间，用户应首先给程序计数器 PC 一个首地址，指令就从这个首地址开始存放，一个字节存放到一个单元后，程序计数器 PC 自动加 1，指令的下一字节存放在 PC 当前值的地址单元。

3.1.4 指令常用的缩写符号说明

在描述 MCS－51 指令系统的功能时，经常使用一些缩写符号，各符号的含义如下：

（1）A：累加器 Acc。常用 Acc 表示其地址，用 A 表示其名称。

（2）AB：累加器 Acc 和寄存器 B 组成的寄存器对。通常在乘、除法指令中出现。

（3）Rn：n = 0 ~ 7，选定的当前工作寄存器，范围为 R0 ~ R7。

（4）Ri：i = 0 或 1，工作寄存器 R0 或 R1。

（5）@：间接寻址符号，简称间址符，常与 Ri 配合用，如@ R1，表示指令对 R1 寄存器间接寻址。

（6）#data：8 位立即数，“#”表示后面的 data 是立即数而不是直接地址。

（7）direct：表示片内 RAM 存储单元的 8 位直接地址，立即数和直接地址可用二进制码表示，后缀为“B”；也可用十六进制码表示，后缀为“H”；如果是以字母开头的十六进制数，在其前面应加一个“0”。如二进制码 10101000B 也可转成十六进制码 A8H，但必须写成“0A8H”。

（8）@ DPTR：表示以 DPTR 为数据指针的间接寻址，用于对外部 64K RAM/ROM 寻址。

（9）rel：以补码形式表示的 8 位地址偏移量，范围为 –128 ~ +127。

（10）Bit：位地址。

（11）$：当前指令的地址。

指令中还经常使用到下列符号，含义如下：

+　加；

–　减；

*　乘；

/　除；

∧　逻辑与；

∨　逻辑或；

⊕　逻辑异或；

<　小于；

>　大于；

< >　不等于；

←→　取代或替换；

（X）　表示 X 寄存器或 X 地址单元中的内容；

（（X））　表示以 X 寄存器或 X 地址单元中的内容为地址的存储单元中的内容；

（$\overline{X}$）　将 X 寄存器的内容取反；

rrr　机器指令三位值由工作寄存器 Rn 确定，R0 ~ R7 对应的 rrr 分别为 000 ~ 111。

例 3 – 1　已知数据存储器各单元内容如图 3 – 1 所示，说明（50H），（A），（$\overline{50H}$），（（50H））各为多少？

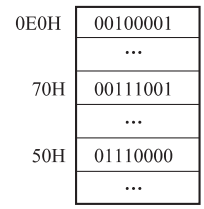

图 3 – 1　例 3 – 1 示意图

解：（50H）：表示地址为50H存储单元里的内容01110000B。

（A）：表示累加器A中的内容，因A的地址为0E0H，所以（A）为00100001B。

（$\overline{50H}$）：表示地址为50H存储单元里的内容取反，为10001111B。

（（50H））：以50H存储单元的内容70H为地址的存储单元内的内容，为00111001B。

3.2 寻址方式

指令包含操作码和操作数，有些指令直接给出参与运算和操作的数，但更多指令只是以各种方式给出操作数所在的地址。单片机通过地址信息寻找操作数的方式，称为寻址方式。寻址方式越多，表明计算机的功能越强，灵活性越大。

在MCS－51单片机中，操作数的存放范围是很宽的，可以放在片外ROM/RAM中，也可以放在片内ROM/RAM以及特殊功能寄存器SFR中。为了适应这一操作数范围内的寻址，MCS－51的指令系统共使用了7种寻址方式，它们是立即寻址、直接寻址、寄存器寻址、寄存器间接寻址、变址寻址、相对寻址和位寻址。

3.2.1 立即寻址

指令中的操作数只是数据，而不是地址，这样的操作数就称为立即数，立即数直接参与操作，这种寻址方式称为立即寻址。其实立即寻址并没有寻址过程。这类指令的立即数大多数是一个字节的8位二进制数。指令中，操作数前有“#”符号，据此可以判定是立即寻址，并有相应的操作码。例如指令：

MOV A，　#64H　；A←64H

这条指令的功能是把数据64H（立即数）送到累加器A中，在MCS－51指令系统中有相应操作码为74H，立即寻址的示意如图3－2所示，设程序计数器PC＝70H。

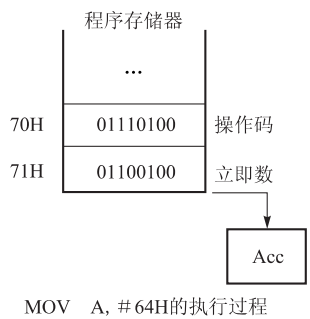

图3－2　立即寻址的示意图

3.2.2 直接寻址

指令中直接给出操作数所在地址的寻址方式称为直接寻址。

例如指令：MOV A，64H　；A←（64H）

这条指令的功能是把内部数据存储器地址为64H的存储单元的内容送至累加器A，这个指令的操作码为0E5H，直接寻址的示意图如图3－3所示，设程序计数器PC＝38H。由图可知，地址为64H的存储单元内容为87H，则指令执行结果是将87H送到Acc。

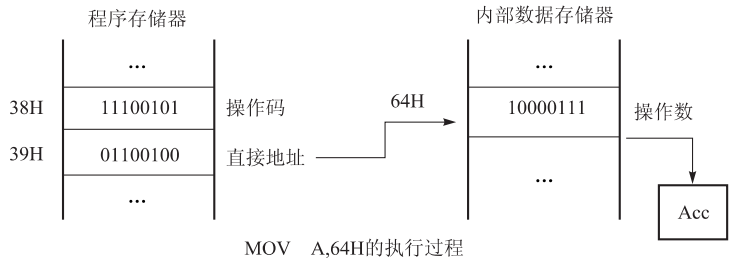

图 3 – 3　直接寻址的示意图

应注意直接寻址方式与立即寻址方式的区别：虽然两者指令的第二字节相同，但在助记符指令中相差一个“#”号，由于符号指令与机器指令有一一对应的关系，故操作码不同，所以执行结果不一样，而在操作码中含有寻址方式的信息。

3. 2. 3　寄存器寻址

由指令指出某一个寄存器的内容作为操作数，这种寻址方式称为寄存器寻址。寄存器寻址对所选的工作寄存区 R0 ~ R7 进行操作，操作码字节的低 3 位 000 ~ 111 指明了所用的寄存器。例如指令：

MOV A，R7　　；A←（R7）

这条指令的功能是把寄存器 R7 的内容送入累加器 A 中，但数据存储器的 4 个工作寄存器区都有寄存器 R7，并都有相应地址，怎样确定 R7 的地址呢？首先由程序状态字寄存器（PSW）中的 D3 位和 D4 位决定寄存器区，再与操作码低 3 位组合（n = 7，为 111）形成地址 0FH（与表 2 – 2 符合），就是工作寄存器区 1 区中的 R7 寄存器。寄存器寻址方式执行过程如图 3 – 4 所示。

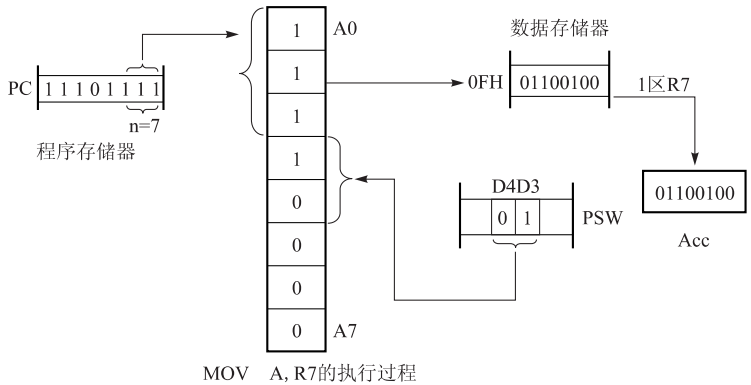

图 3 – 4　寄存器寻址方式

3.2.4 寄存器间接寻址

由指令指出某一个寄存器的内容作为操作数的地址，这种寻址方式称为寄存器间接寻址。特别注意：寄存器的内容不是操作数，而是操作数所在的存储器地址。例如指令：

MOV A, @Ri

这条指令中，i=0 指 R0；i=1，则指 R1。

指令功能是将以寄存器 R0 或 R1 内容为地址的存储单元中的数据送入 A。执行这条指令首先应找到 Ri 地址，Ri 的 8 位地址 A7～A0 中：A7A6A5=000，A2A1=00，A0 由 i 决定，A4A3 由程序状态字寄存器 PSW 中 D4D3 位决定，其作用实际是选定工作寄存器区域。寄存器间接寻址执行过程如图 3-5 所示，图中 i=1，将以工作寄存器区 1 的 R1 内容 64H 为地址的存储单元的数据 51H 送入 A。

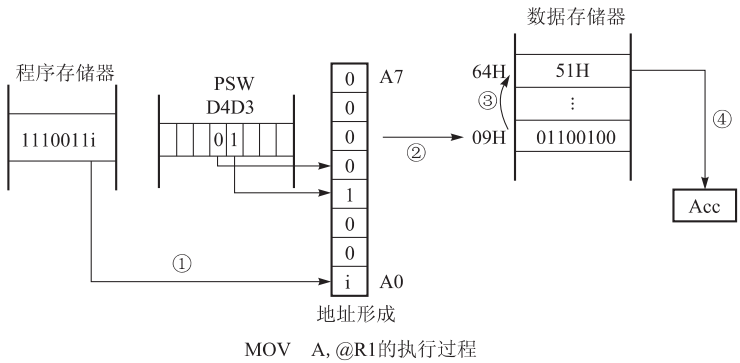

图 3-5 寄存器间接寻址方式

3.2.5 变址寻址

变址寻址也称基地址寄存器加变址寄存器间接寻址，它是以程序计数器 PC 或数据指针 DPTR 作为基地址寄存器，以累加器 A 作为变址寄存器，把两者内容相加形成操作数的地址。这种寻址方式用于读取程序存储器中常数表，或访问外部数据存储器。例如指令：

MOVC A, @A+DPTR ；A←((A)+(DPTR))

这条指令的功能是把 DPTR 的内容作为基地址，DPTR 是 16 位寄存器，其高 8 位在 DPH 中，低 8 位在 DPL 中。把累加器 A 的内容作为地址偏移量，两者相加后得到 16 位地址，把该地址对应的程序存储器 ROM 单元中的内容送到 A 中，A 中原数据自动擦除。寻址过程示意图如图 3-6 所示。

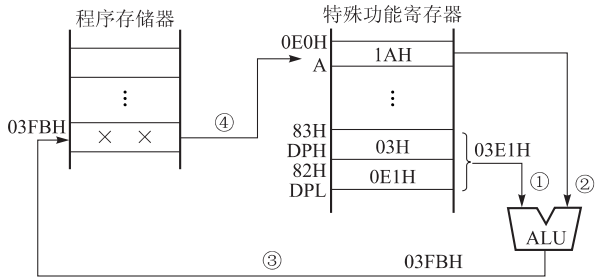

图 3 – 6　变址寻址方式

3.2.6　位寻址

MCS – 51 指令系统有一些指令是用于位的操作。位操作指令能对内部 RAM 中的位寻址区和某些有位地址的特殊功能寄存器进行位操作，也就是说可对位地址空间的每个位进行位变量传送、状态控制、逻辑运算等操作，常用的位处理器是程序状态字寄存器的 Cy 位。例如指令：

MOV C，06H　；Cy←（06H）

C 累加器只有一位，指令是位寻址方式，操作码是 0A2H，06H 是位地址。查看表 2 – 1 可知，指令含义是将内部 RAM 的 20H 单元的 D6 位的内容送入累加器 C，其指令执行过程如图 3 – 7 所示。这条指令应区别于指令“MOV A，06H”，A 是 8 位累加器，意思是将 RAM 中 06H 单元的内容送入 A 中，操作码是 0E5H。

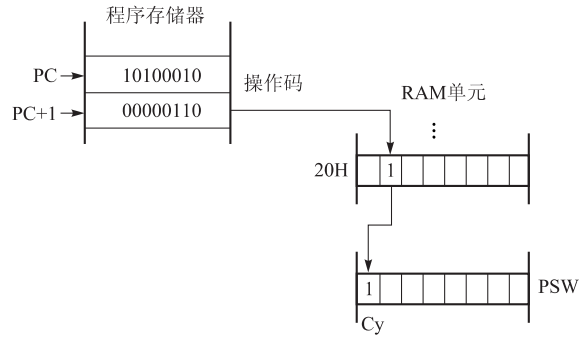

图 3 – 7　位寻址方式

3.2.7　相对寻址

相对寻址以程序计数器 PC 的当前值作为基地址，与指令中给定的相对偏移量 rel 相加，作为程序逻辑的转移地址。这种寻址方式用于相对转移指令中。

3.3　数据传送指令

在 MCS – 51 单片机中，数据传送是最基本和最主要的操作。数据传送操作可以在片内 RAM 和 SFR 内进行，也可以在累加器 A 和片外存储器之间进行。指令中必须指定传送数据

的源地址和目的地址，以便机器执行指令时把源地址中内容传送到目的地址中，但不改变源地址中内容。在这类指令中，除以累加器 A 为目的操作数寄存器指令会对奇偶标志位 P 有影响外，其余指令执行时均不会影响任何标志位。

MCS－51 单片机的数据传送指令共有 28 条，分为内部数据传送指令、外部数据传送指令、堆栈操作指令和数据交换指令等 4 类。本节将分类进行介绍，并给出相应的机器代码，以便相互对照加以理解。

3.3.1 内部数据传送指令（15 条）

1. 以 A 为目的操作数的指令

这类指令的格式为：　　MOV　　目的操作数，　　源操作数

1）指令与指令代码

指令	指令代码	操作
MOV A，Rn	11101rrr	A←（Rn）
MOV A，direct	11100101 direct	A←（direct）
MOV A，@Ri	1110011i	A←((Ri))
MOV A，#data	01110100 data	A←data

对 Rn 寻址的指令，其机器码字节的低三位为 rrr，对应于 8 个工作寄存器之一，当为 000 时，表示 R0；为 001 时，表示 R1；依次类推。

2）指令功能

这组指令的功能是把源操作数的内容送入累加器 A。

3）源操作数寻址方式

有立即寻址、直接寻址、寄存器寻址、寄存器间接寻址等寻址方式。

例 3－2　说明下列指令的功能及寻址方式。

```
MOV  A,  R6
MOV  A,  64H
MOV  A,  @R0
MOV  A,  #78H
```

解：

```
MOV A,   R6     ;   A←(R6),      寄存器寻址
MOV A,   64H    ;   A←(64H),     直接寻址
MOV A,   @R0    ;   A←((R0)),    寄存器间接寻址
MOV A,   #78H   ;   A←78H,       立即寻址
```

例 3－3　说明下列指令的错误。

```
MOV A, @R3   ;   A←((R3))
```

解：这条指令是错误的，@ 表示寄存器间接寻址，而用于寄存器间接寻址的只有 4 个工

作寄存器区的 R0 和 R1 寄存器。

4）指令字节

表面看来，这组指令由 3 部分组成：操作码、A 和源操作数，但操作码含有向 A 送数的信息，所以累加器 A 不占程序存储器的存储空间，故这个指令是两字节指令。一般来说，不占程序存储器的存储空间的还有各区的工作寄存器。

2. 以 Rn 为目的操作数的指令

1）指令与指令代码

指令	指令代码	操作
MOV Rn，A	11111rrr	Rn←（A）
MOV Rn，direct	10101rrr direct	Rn←（direct）
MOV Rn，#data	01111rrr data	Rn←data

2）功能

这组指令的功能是将源操作数的内容送入当前工作寄存器区的 R0 ~ R7 中的某一个寄存器。

3）源操作数寻址方式

有立即寻址、直接寻址、寄存器寻址。

例 3 –4　指出下列指令的功能和寻址方式。

```
MOV R2，#64H
MOV R3，64H
MOV R7，A
```

解：

```
MOV R2，#64H     ;    R2←64H，      立即寻址
MOV R3，64H      ;    R3←（64H），  直接寻址
MOV R7，A        ;    R7←（A），    寄存器寻址
```

例 3 –5　指出指令“MOV Rn，@Ri”的错误。

解：这条指令是错误的，在以 Rn 为目的操作数的数据传送中，无寄存器间接寻址的寻址方式。

例 3 –6　指出指令“MOV R1，R3”的错误。

解：这个指令是错误的，MCS –51 指令系统中无工作寄存器之间的直接传送指令。要实现（R3）→R1 必须有以下指令：

```
MOV A，R3
MOV R1，A
```

3. 以直接寻址的单元为目的操作数指令

1）指令与指令代码

指令	指令代码	操作
MOV direct，A	11110101 direct	direct←（A）

MOV direct，Rn	10001rrr direct	direct←（Rn）
MOV direct1，direct2	10000101 direct2 direct1	direct1←（direct2）
MOV direct，@Ri	1000011i direct	direct←（（Ri））
MOV direct，#data	01110101 direct，data	direct←data

应注意："MOV direct1，direct2"指令译成指令代码时，源地址（direct2）在前，目的地址 direct1 在后。

2）功能

这组指令的功能是把源操作数送入由直接地址指向的存储单元，直接寻址时，direct 可以是特殊功能寄存器的地址（见表 2－2）、内部 RAM 区的地址，因此允许对端口直接操作。例如指令："MOV P1，40H"的含义是将 40H 单元的内容送入 P1 端口，机器指令为"10000101 01000000 10010000"，是三字节指令。第一字节是操作码，第二字节是源操作数的地址，第三字节是目的操作数 P1 的地址。

3）源操作数寻址方式

有立即寻址、直接寻址、寄存器寻址、寄存器间接寻址。

例 3－7 若（64H）=70H，（A）=78H，说明执行指令"MOV 64H，A"的结果。

解：执行后结果为（64H）=78H，（A）=78H 不变。

例 3－8 设（30H）=40H，（R0）=36H，（36H）=52H，说明执行指令"MOV 30H，@R0"的结果。

解：执行后，（30H）=52H，但（R0）=36H 和（36H）=52H 不变。

例 3－9 若（30H）=35H，（40H）=45H，说明执行指令"MOV 30H，40H"的结果。

解：这条指令功能将 40H 单元内容送至 30H 的单元内，指令执行结果应是（30H）=45H，而（40H）=45H 不变。

4. 以寄存器间接寻址的单元为目的操作数指令

1）指令与指令代码

指令	指令代码	操作
MOV @Ri，A	1111011i	（Ri）←（A）
MOV @Ri，direct	1010011i direct	（Ri）←（direct）
MOV @Ri，#data	0111011i data	（Ri）←data

2）功能

本组指令是将源操作数送入以 R0 或 R1 内容为地址的存储单元中。

3）源操作数寻址方式

有立即寻址、直接寻址、寄存器寻址。

例 3－10 若（30H）=35H，（R1）=70H，说明执行指令"MOV@R1，30H"的结果。

解：70H 单元的内容为 35H，同时（30H）=35H，（R1）=70H 不变。

例 3 – 11　已知数据存储器的部分单元如图 3 – 8 所示，（A）= 0EEH，说明指令执行后各存储单元的情况。

① MOV　@R1，A

② MOV　@R0，40H

③ MOV　R1，　#40H

解：70H 中为 0EEH；

50H 中为 0FFH；

R0 中为 50H；

R1 中为 40H。

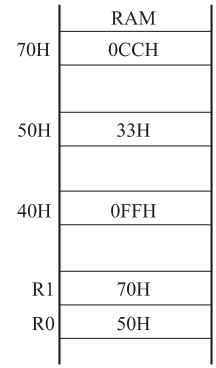

图 3 – 8　例题示意

例 3 – 12　设（70H）= 60H，（60H）= 20H，P1 口为输入口，当前的输入状态为 0B7H，执行下面指令，说明最后结果。

```
MOV RO，#70H          ；R0←70H
MOV A，@R0            ；A←60H
MOV R1，A             ；R1←60H
MOV B，@R1            ；B←20H
MOV @R0，P1           ；70H←0B7H
```

解：结果为：（70H）= 0B7H，（B）= 20H，（R1）= 60H，（R0）= 70H，（60H）= 20H。

3.3.2　外部数据传送指令（7 条）

1. 16 位数据传送指令

1）指令与指令代码

指令	指令代码	操作
MOV DPTR，#data16	10010000 dataH8 dataL8	DPTR←data16

2）功能

把 16 位常数送入 DPTR，16 位的数据指针 DPTR 由 DPH 和 DPL 组成，执行结果把高位立即数送入 DPH，低位立即数送入 DPL，本指令是三字节指令。

2. 外部 ROM 的字节传送指令

由于外部程序存储器只读不写，因此数据传送是单向的，即只从外部程序存储器中读出数据，并且只能向累加器 A 传送。这类指令共有两条，均属于变址寻址指令，因专门用于查表而又称为查表指令。

1）指令　MOVC A，　@A + PC

本指令的指令代码是 10000011，为一字节指令，当 CPU 读取本条指令后，PC 已自动加 1，指向下一条指令的首字节的地址。如果该指令存放单元为 1000H，则 PC 的当前值为 1001H。执行该指令时，把 A 的内容作为无符号数和 PC 当前内容相加后得到一个 16 位的地

址，并将该地址指出的程序存储器单元内容送到累加器 A。而 A 的内容从 00H ~ 0FFH 共 256 个，所以（A）和（PC）相加得到的地址只能在该查表指令以下 256 个单元地址内。“MOVC”中的“C”表示程序存储器。

例 3 – 13 设（A）= 30H，执行指令“1000H：MOVC A，@A + PC”的结果。

解：该指令将程序存储器中 1031H 单元内容送入累加器 A 中。

例 3 – 14 说明下面一段程序的执行情况。

```
8000H：MOV A，#20H
8002H：MOV A，@A + PC
8003H：MOV 60H，A
       ……
8023H：MOV 30H，#24H
```

解：执行情况如下：

① 执行第一条指令后，（A）= 20H。

② 执行第二条指令时，PC 当前值为下条指令首址 8002H + 1 = 8003H，将（A）+ 8003H = 20H + 8003H = 8023H。将程序存储器地址为 8023H 的内容送至 A。由于程序存储器 8023H 单元存放的指令“MOV 30H，#24H”的操作码为 75H，所以执行完后（A）= 75H。

③ 第三条指令将（A）→（60H），所以（60H）= 75H。

如果最后一条指令为“8023H：MOV 30H，24H”，则 8023H 单元存放的操作码将是 85H，那么（60H）= 85H。

2）指令 MOVC A，@A + DPTR

这条指令的指令代码是 10010011，它的操作是以 DPTR 作为基地址寄存器，由 A 中的无符号数与 DPTR 中的内容相加得到一个 16 位的程序存储器的地址，并将该地址的内容送到 A。

例 3 – 15 设（DPTR）= 8100H，（A）= 40H，说明执行下列指令后的结果。

```
MOVC A，  @A + DPTR
```

解：将程序存储器中 8140H 单元内容送入累加器 A。

该指令只与 A 的内容相关，与该指令存放的地址无关，因此表格大小和位置可在 64K 字节程序存储器中任意安排，只要在查表之前对 DPTR 和 A 赋值，就使一个表格可被各个程序所公用。

3. 外部 RAM 的字节传送指令

1）指令与指令代码

指令	指令代码	操作
MOVX A，@DPTR	11100000	((DPTR))→A
MOVX A，@Ri	1110001i	((Ri) + (P2))→A

MOVX	@ DPTR，A	11110000	（A）→（DPTR）
MOVX	@ Ri，A	1111001i	（A）→（Ri）+（P2）

2）功能

这组指令是将累加器 A 和外部扩展的 RAM 的数据传送，由于 DPTR 是 16 位地址指针，因此用 DPTR 间接寻址的指令寻址范围是 0 ~ 64KB；而 Ri 是 8 位寄存器，故用 Ri 间接寻址的指令寻址范围是外部 RAM 的低 256 单元。外部 RAM 的数据传送，只有通过累加器 A 来进行。在片外容量大于 256 个单元时，地址高 8 位由 P2 口输出，而 Ri 中的地址作为低 8 位地址经 P0 口输出，这些（Ri）与（P2）组合后，可对片外 0 ~ 64KB 范围寻址。

例 3 –16　将外部 RAM 30H 单元的内容送入内部 RAM 20H 的单元。

解：
```
MOV R0，#30H
MOVX A，@R0
MOV 20H，A
```

例 3 –17　将外部 RAM 3000H 单元的内容送入内部 RAM 20H 的单元。

解： 方法一：
```
MOV DPTR，  #3000H
MOVX A，  @DPTR
MOV 20H，A
```
方法二：
```
MOV P2，  #30H
MOV R1，  #00H
MOVX A，  @R1
MOV 20H，  A
```

3.3.3　堆栈操作指令（2 条）

在 MCS –51 内部 RAM 中可以设定一个后进先出的堆栈，地址为 30H ~ 7FH，堆栈指针 SP 中的内容总是堆栈区中最后一个进栈数据所在的存储单元地址。堆栈操作包括进栈和出栈两种。

1. 进栈指令与指令代码

指令	指令代码	操作
PUSH direct	11000000 direct	SP←（SP）+1，（SP）←（direct）

这条指令首先将堆栈指针 SP 的内容加 1，然后把直接地址里的内容传送到堆栈指针 SP 指出的内部 RAM 存储单元中。

例 3 –18　设（SP）=30H，（ACC）=60H，（B）=70H，执行下列指令后结果怎样？
```
PUSH ACC
PUSH B
```

解： 操作过程是：

PUSH ACC ；（SP）+1，31H→SP，（ACC）→31H

PUSH B ；（SP）+1，32H→SP，（B）→32H

结果为（31H）=60H，（32H）=70H，（SP）=32H。

2. 出栈指令与指令代码

指令	指令代码	操作
POP direct	11010000 direct	((SP))→direct，SP←（SP）-1

这条指令的功能是将堆栈指针 SP 指出的内部 RAM 单元的内容送入直接地址指出的存储单元中，堆栈指针 SP 减 1。出栈指令用于恢复 CPU 现场。

例 3-19 设（SP）=32H，（32H）=70H，（31H）=60H，执行下列指令后结果怎样？

POP DPH

POP DPL

解：操作过程是：

POP DPH ；((SP))→DPH，（SP）-1→SP

POP DPL ；((SP))→DPL，（SP）-1→SP

结果为（DPH）=70H，（DPL）=60H，所以，（DPTR）=7060H，（SP）=30H。

3.3.4 数据交换指令（4 条）

数据交换指令共有 4 条，其中字节交换指令 3 条，半字节交换指令 1 条。

1. 字节交换指令

1）指令与指令代码

指令	指令代码
XCH A，Rn	11001rrr
XCH A，direct	11000101 direct
XCH A，@Ri	1100011i

2）功能

将累加器 A 的内容和源操作数内容相互交换。

3）源操作数寻址方式

有寄存器寻址、直接寻址、寄存器间接寻址。

例 3-20 设（A）=80H，（R1）=74H，（74H）=60H，（60H）=50H，（80H）=40H，顺序执行下列指令后结果怎样？

XCH A，R1

XCH A，60H

XCH A，@R1

解：执行指令①结果：（A）=74H，（R1）=80H。