

1.4.3 专用寄存器 SFR（名称及定义）

MCS51共21个专用寄存器，PC（程序计数器）在物理上是独立的，不属于内部RAM的SFR块；其余的专用寄存器都属于内部RAM的SFR块。

（1）程序计数器PC

PC是一个16位的专用寄存器，用来存放下一条要执行指令的地址（ROM的地址），可以寻址64KB的地址空间。

（2）累加器ACC

通常用助记符A，它是一个最常用的8位专用寄存器。

（3）B寄存器

B在乘、除法指令中有专门用途，与A配合使用，一般场合可用作普通RAM单元；

（4）程序状态字PSW

程序状态字寄存器PSW是一个8位寄存器。位于专用寄存器区，字节地址为D0H。PSW用于存放程序运行中的各种状态信息，可以进行位寻址。PSW各位的定义如下：

D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
Cy	AC	F0	RS1	RS0	OV	-	P

程序状态字PSW

CY：高位进位/借位标志；位累加器。

在进行加或减运算时，如果操作结果最高位有进位或借位时，Cy由硬件置“1”，否则清“0”。在进行位操作时，Cy又可以被认为是**位累加器**，它的作用相当于CPU中的累加器A。

AC：辅助进/借位标志（又称半进位）；

在进行加或减运算时，低四位向高四位产生进位或借位时，AC将由硬件置“1”，否则清“0”。AC位可用于BCD码调整时的判断位。

F0：用户定义标志位；

由用户置位或复位，它可作为用户自行定义的一个状态标记。

RS1、RS0：工作寄存器组选择控制位；

它用于选择内部RAM低128字节块中，4个工作寄存器组中的一个，每个寄存器组有8个8位工作寄存器（R0-R7），RS1和RS0可以用软件来置位或清零。

程序状态字PSW

RS1 RS0与工作寄存器组的对应关系

RS1	RS0	寄存器组	内部RAM地址
0	0	寄存器组0	00H~07H
0	1	寄存器组1	08H~0FH
1	0	寄存器组2	10H~17H
1	1	寄存器组3	18H~1FH

OV：为溢出标志；

用于带符号数计算，当进行算术运算时，如果产生溢出，则由硬件将OV位置1，否则清“0”。

当执行有符号数的加法指令**ADD**或减法指令**SUBB**时，当**D6**位有向**D7**位的进位或借位时 $D_{6Cy}=1$ 时，而**D7**位没有向**Cy**位的进位或借位 $D_{7Cy}=0$ 时，则**OV=1**；或 $D_{6Cy}=0$ ， $D_{7Cy}=1$ 则**OV=1**。所以溢出的逻辑表达式为： $OV = D_{6Cy} \oplus D_{7Cy}$

程序状态字PSW

OV：为溢出标志；

或者说：

当位6向位7有进位或借位，而位7不向Cy进位或借位时，OV=1；（不同时发生）

当位6不向位7进位或借位，而位7向Cy有进位或借位时，OV=1；（不同时发生）

产生溢出（OV=1），表示运算结果超出寄存器A所能表示的带符号数的范围（-128—+127）。

P：为奇偶校验标志；

指示累加器A中1的个数的奇偶性。

当累加器A内容中有奇数个“1”时，P置1；有偶数个“1”时，P置“0”。

改变累加器A中内容的指令，均会影响P标志位。

专用寄存器 SFR（名称及定义）

（5）堆栈指针 SP

- 堆栈指针SP是一个8位的专用寄存器，它总是指明堆栈顶部在内部RAM中的位置，可由软件设置初始值。系统复位后，SP初始化为07H，压栈时，SP首先自动加“1”，使得堆栈实际上由08H单元开始存放数据，但在实际应用中，SP一般被设置在30H～FFH的范围内。
- 堆栈存取数据时遵循“先进后出，后进先出”的原则，其中的存储单元在使用时是不能按字节任意访问的，通过专门的堆栈操作指令把数据压入或弹出堆栈。
- 堆栈的主要功能是为子程序和中断操作而设立。

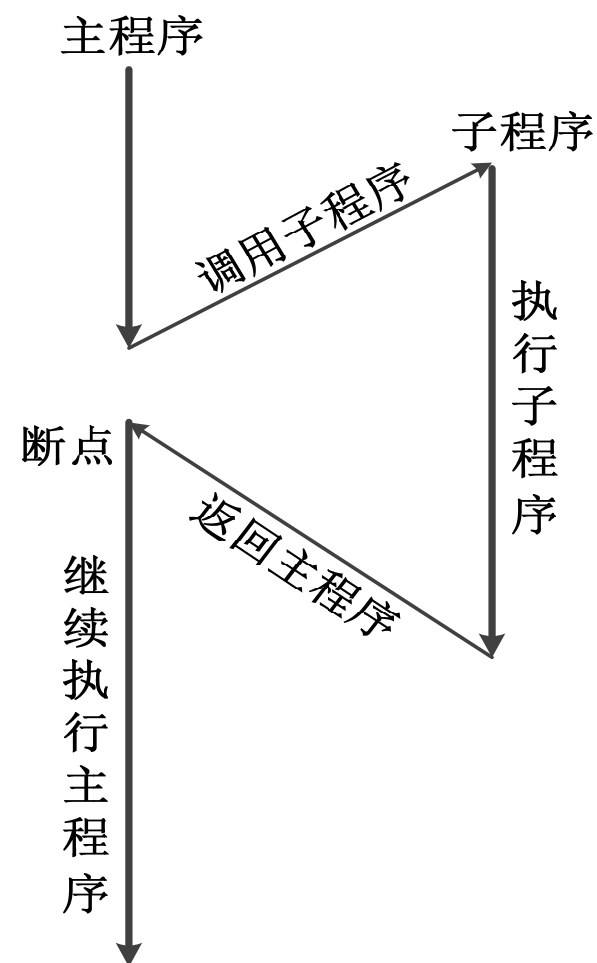
在响应中断或子程序调用时：PC自动入栈，但PSW并不自动入栈（需用软件操作使其压入、弹出堆栈）。

SP的内容可编程，从而能实现将堆栈再定位到内部RAM的任意位置；SP的初始值越小，堆栈的深度就可越深。

堆栈指针 SP

堆栈功能：保护断点

在调用子程序和执行中断服务程序的过程中，都要保护断点地址。所谓断点地址，就是在执行调用子程序或者转入中断服务程序时，主程序按顺序的下一条指令的地址。只有保护了断点地址，才能在子程序或者中断服务程序执行后保证返回到主程序的断点处继续执行。



堆栈指针 SP

堆栈功能：现场保护

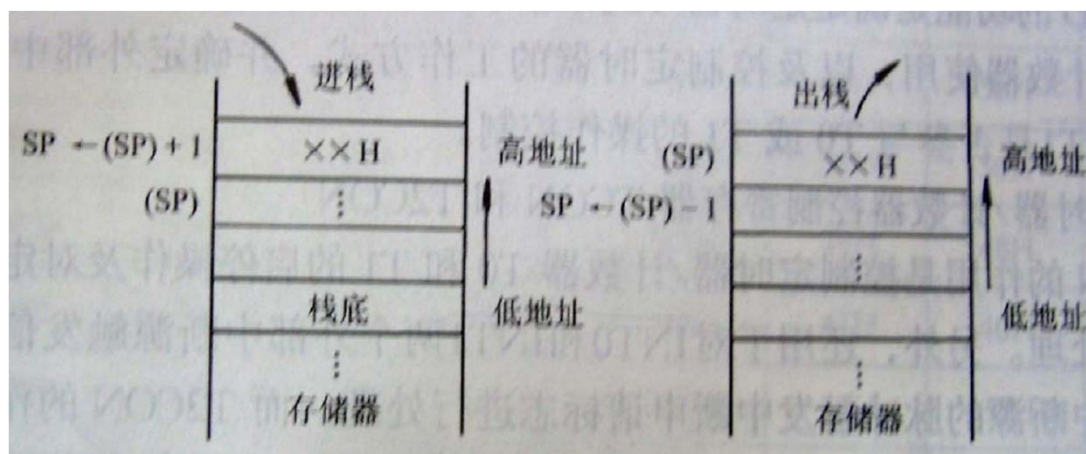
所谓现场，就是调用子程序或者进入中断服务程序前累加器ACC、工作寄存器、标志寄存器中的信息，这些信息是主程序执行的中间结果，如果在执行子程序或者中断服务程序的过程中要使用这些寄存器，将会破坏原来的内容。为此，进入子程序或者中断服务程序后，先要通过堆栈转存这些寄存器中的内容。这就是现场保护。

在返回主程序前，在把保存在堆栈中的现场信息送回对应的寄存器，称为现场恢复。

堆栈指针 SP

堆栈操作

堆栈有两种操作方式。
一种是数据压入（PUSH）堆栈，另一种是数据弹出（POP）堆栈。每次执行压入堆栈操作前，SP自动加1；执行弹出堆栈操作后，SP自动减1。



在调用子程序指令或者进入中断服务程序时，断点地址压入堆栈由硬件自动实现的；在执行返回主程序指令RET或者中断返回指令RETI时，断点地址自堆栈中弹出也是由硬件自动实现。现场保护和现场恢复需要由PUSH指令和POP指令完成。

专用寄存器 SFR（名称及定义）

（6）数据指针 DPTR

DPTR是一个16位专用寄存器，指明访问的数据存储器的单元地址，它由两个8位的寄存器DPH与DPL组成，某些情况下，DPH、DPL可以单独使用。

DPTR主要用来存放16位的地址，当对64KB外部数据存储器空间寻址时，作为间址寄存器使用，而在访问程序存储器时，DPTR作为基址寄存器使用。

访问外部数据存储器：

```
MOVX  A, @DPTR  
MOVX  @DPTR, A;
```

访问程序存储器（实现“远程查表”）：

```
MOVC  A, @A+DPTR（基址+变址寻址）。
```

（7）I/O端口P0~P3

专用寄存器P0~P3分别是I/O端口P0~P3的锁存器。

在MCS-51中，将I/O端口，当作一般的专用寄存器来使用，采用MOV指令，可以用直接寻址方式访问。

专用寄存器 SFR（名称及定义）

（8）数据缓冲器 SBUF

串行口内部的数据缓冲器SBUF，在物理上由两个独立的寄存器组成，一个是**发送缓冲器**，另一个是**接收缓冲器**，共同使用一个逻辑地址（99H）。

写入SBUF的数据，存放在**发送缓冲器**，用于串行发送；

从SBUF读取的数据，取自**接收缓冲器**，读取的是刚接收到的数据。

（9）计数寄存器 TH0、TL0、TH1、TL1

51子系列单片机中有二个16位的定时器/计数器，即T0和T1。它们各自包含两个独立的8位计数寄存器：TH0、TL0，TH1、TL1。其中TH0、TL0分别是T0的高8位与低8位加法计数器，TH1、TL1分别是T1的高8位与低8位加法计数器。不能将它们作为一个16位的寄存器进行寻址访问。

52子系列单片机还有T2定时器/计数器（TH2、TL2），并在SFR寄存器区中设立了同样相应的定时器/计数器、控制方式寄存器、控制寄存器等。

专用寄存器 SFR（名称及定义）

（10）其它控制寄存器

IE：中断允许寄存器；

IP：中断优先级控制寄存器；

TMOD：定时器模式控制寄存器；

TCN：定时器控制寄存器；

SCN：串行控制/状态寄存器；

PCN：电源控制寄存器；

1.5 振荡器、时钟电路和CPU时序

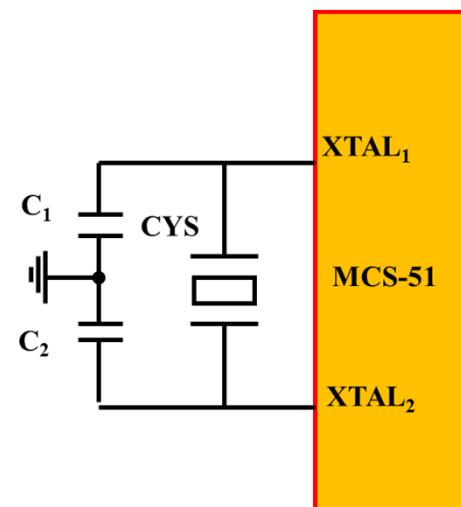
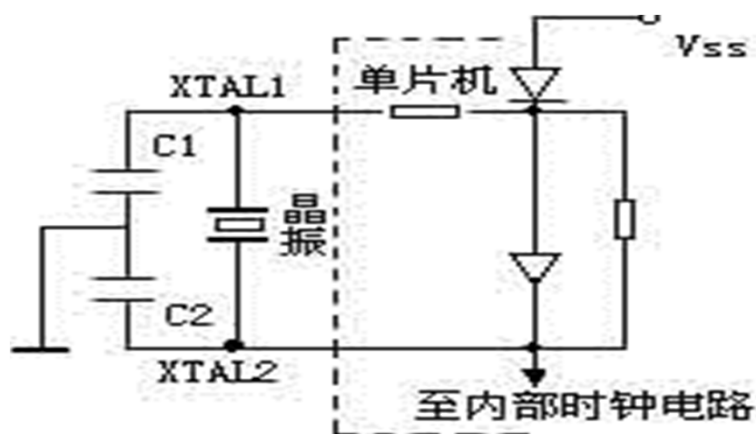
1.5.1 单片机的时钟电路

单片机时钟电路通常有两种形式：

(1) 内部时钟方式

MCS-51单片机片内有一个用于构成振荡器的高增益反相放大器，引脚 $XTAL_1$ 和 $XTAL_2$ 分别是此放大器的输入端和输出端。把放大器与作为反馈元件的晶体振荡器或陶瓷谐振器连接，就构成了内部自激振荡器并产生振荡时钟脉冲。

当外接晶振时， C_1 和 C_2 值通常选择30pF；外接陶瓷谐振器时， C_1 和 C_2 的典型值约为47pF。在设计印刷电路板时，晶体或陶瓷谐振器和电容应尽可能安装在单片机芯片附近，以减少寄生电容。 C_1 、 C_2 对频率有微调作用，振荡频率范围是1.2MHz~12MHz。

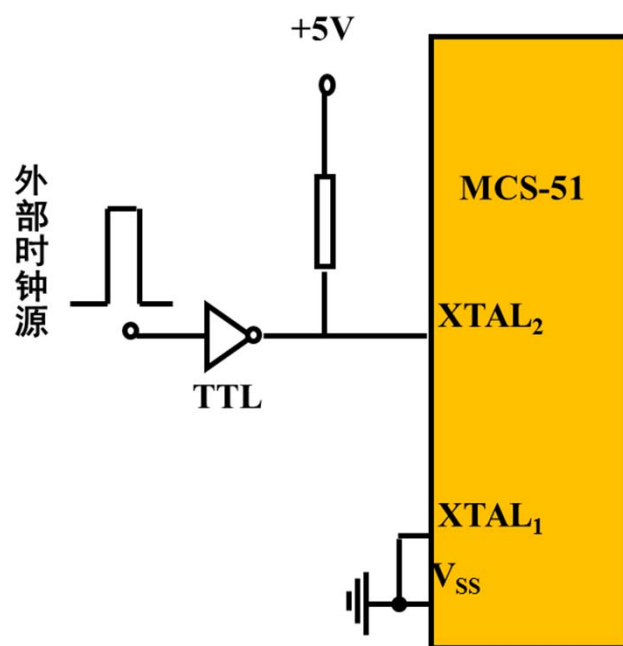


1.5 振荡器、时钟电路和CPU时序

1.5.1 单片机的时钟电路

(2) 外部时钟方式

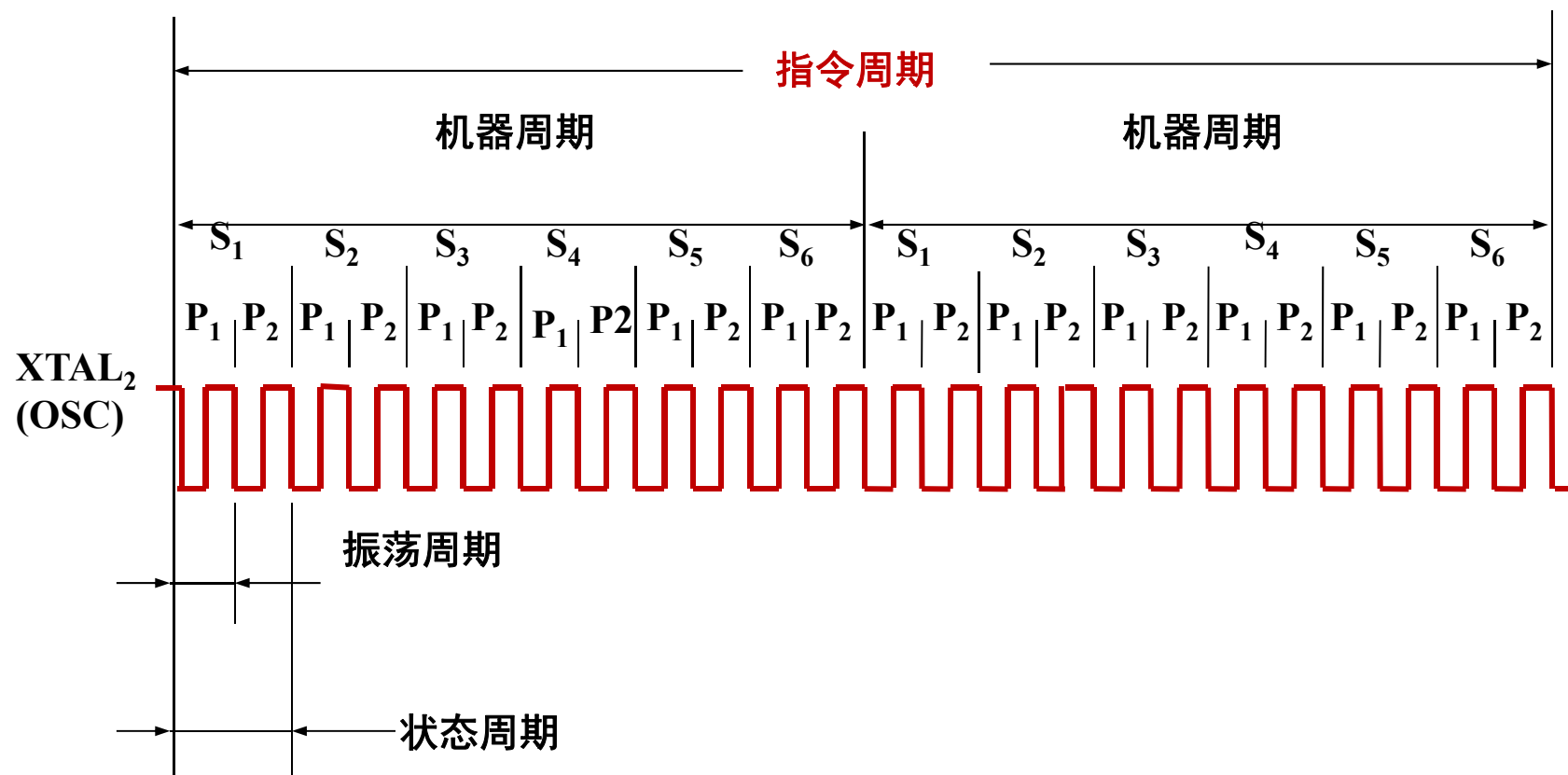
外部时钟方式就是把外部已有的时钟信号引入单片机内部



时钟的基本概念：

时钟是单片机的“心脏”，每跳动一下，整个单片机的各个电路就同步的动作一下，要求稳定而准确

1.5.2 振荡周期、状态周期、机器周期和指令周期



MCS-51单片机各种周期的相互关系

- **振荡周期**：为单片机提供时钟信号的振荡源的周期（ $T_{osc}=1/f_{osc}$ ）；
- **状态周期**：是振荡源信号经二分频后形成的时钟脉冲信号；
- **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；
- **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。

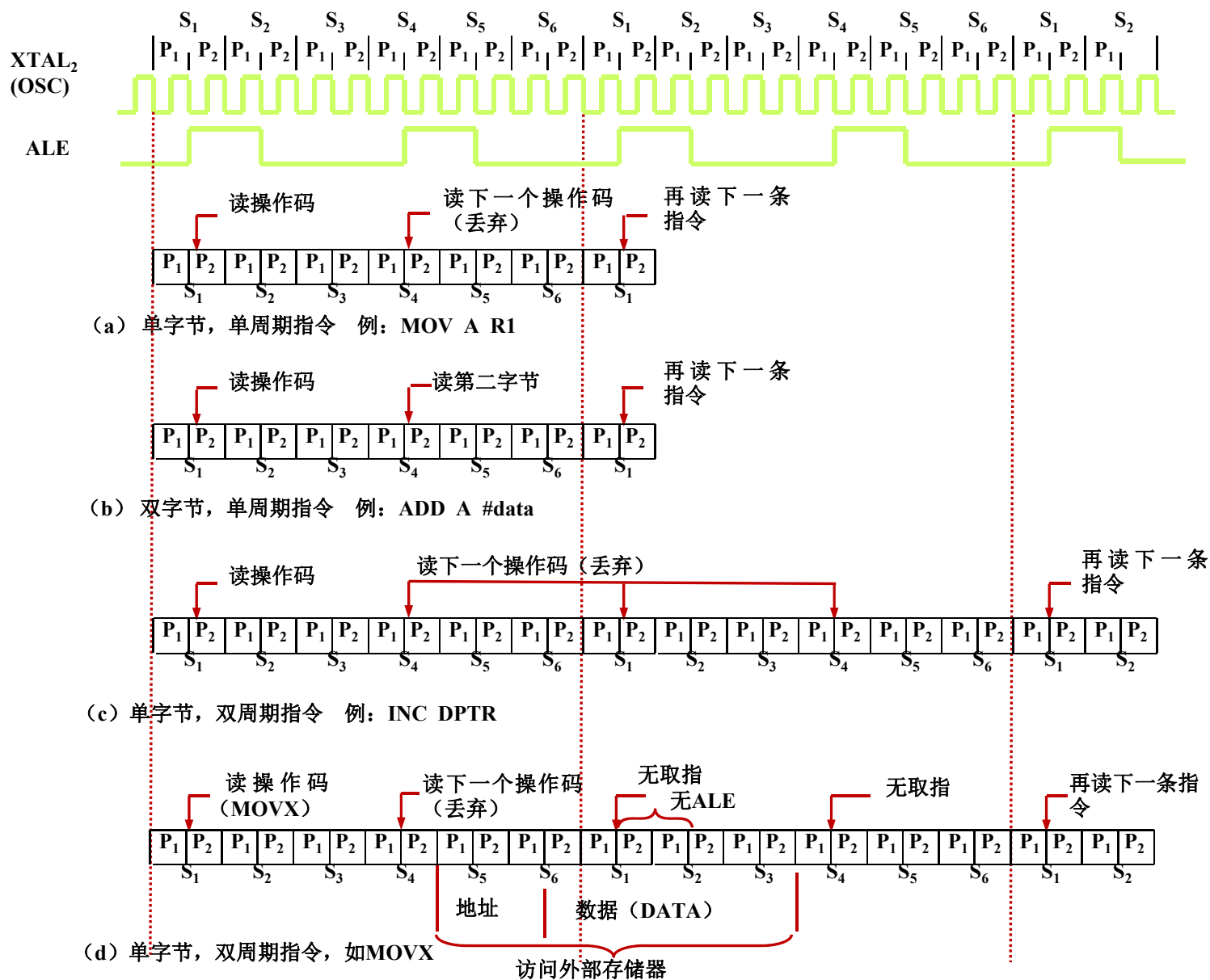
例如：若 MCS-51 单片机外接晶振为 12MHz 时，则单片机的四个周期的具体值为：

- 振荡周期 = $1/12\text{MHz} = 1/12\mu\text{s} = 0.0833\mu\text{s}$
- 状态周期 = $1/6\mu\text{s} = 0.167\mu\text{s}$
- 机器周期 = $1\mu\text{s}$
- 指令周期 = $1\sim 4\mu\text{s}$

CPU完成一个基本操作所需要的时间为机器周期，这个基本操作，如取指、读/写数据等。每12个**振荡周期**为1个**机器周期**。

一个机器周期分为6个状态：
S1~S6，每个状态分为两个节拍：
P1和P2。

1.5.3 MCS51单片机指令的取指和执行时序



1.5.3 MCS51单片机指令的取指和执行时序

时序的定义：

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

当ALE（ALE信号为振荡频率6分频）正跳变时，对应单片机进行一次读指令操作。一个机器周期二次出现，在S1P2和S2P1及S4P2和S5P1期间，有效宽度为一个状态。

(1) 单字节单周期指令：MOV A , R1

只需进行一次读指令操作（指令只有一个字节），当第二个ALE有效时，由于PC没有加1，读出的还是原指令。属于一次无效操作。

(2) 双字节单周期指令：ADD A , #data

ALE两次读操作都有效，第一次读操作码（指令第一字节），第二次读立即数（指令第二字节）。

(3) 单字节双周期指令：INC DPTR

两个机器周期共进行四次读指令操作，但其后三次的读操作都是无效的。

1.5.3 MCS51单片机指令的取指和执行时序

(4) 单字节双周期指令：MOVX（访问外部数据存储器的指令）

在第一个机器周期S5开始时，送出外部RAM地址，随后读/写数据，读/写期间在ALE端不输出有效信号，在第二个机器周期，即外部RAM已被寻址和选通后，也不产生取指操作。

1.5.4 MCS51典型复位电路及复位状态

复位操作

复位是单片机的初始化操作，其作用是使CPU和系统中其它部件都处于一个确定的初始状态，并从这个状态开始工作。RST端保持2个机器周期（24个振荡周期）的高电平，就可完成一次复位。

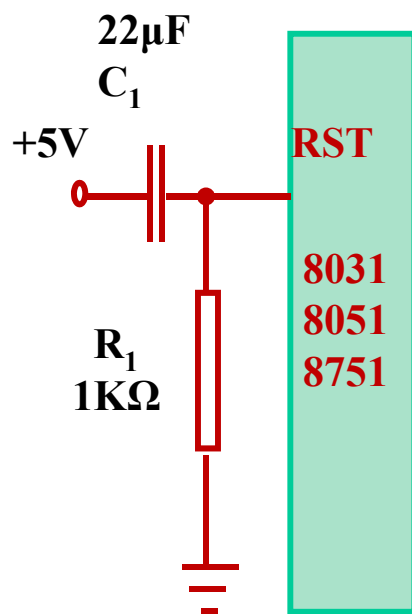
复位后片内各专用寄存器的状态

表 2-7 复位时片内各寄存器的状态

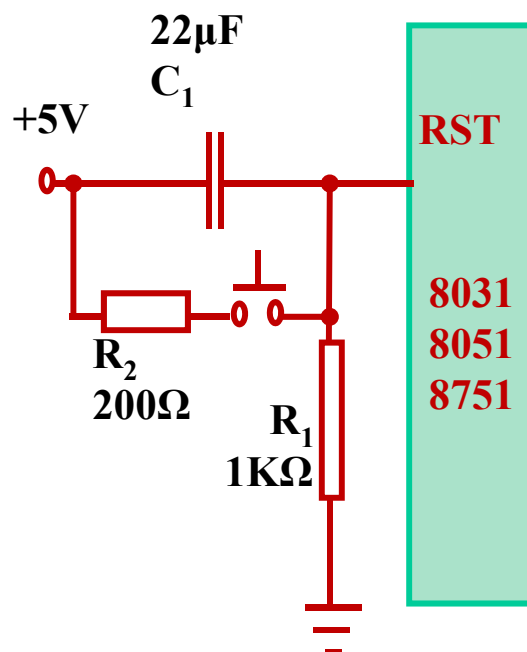
寄 存 器	复 位 状 态	寄 存 器	复 位 状 态
PC	0000H	DP1H	00H
Acc	00H	DP1L	00H
PSW	00H	TMOD	00H
B	00H	TCON	00H
SP	07H	TH0、TH1	00H
DPTR	0000H	TL0、TL1	00H
P0 ~ P3	FFH	SCON	00H
IP	× × 00 0000 B	SBUF	× × × × × × × × B
IE	0 × 00 0000 B	PCON	0 × × × 0 0 0 0 B
DP0H	00H	AUXR	× × × 0 0 × × 0 B
DP0L	00H	AUXR1	× × × × × × × 0 B
WDTRST	× × × × × × × × B		

注：X表示状态不确定

复位电路：单片机复位电路包括片内、片外两部分。外部复位电路就是为内部复位电路提供两个机器周期以上的高电平而设计的。MCS-51单片机通常采用上电自动复位和按键手动复位两种方式。

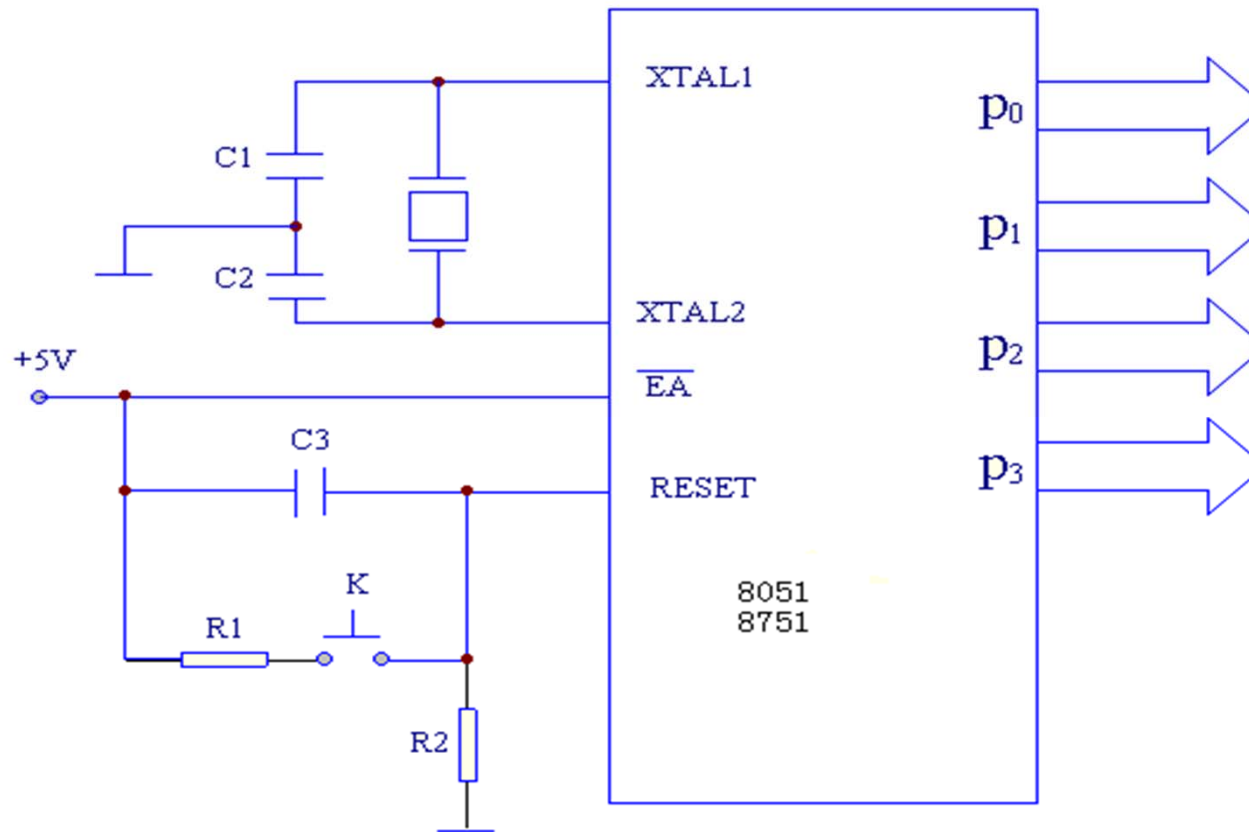


(a) 上电复位电路



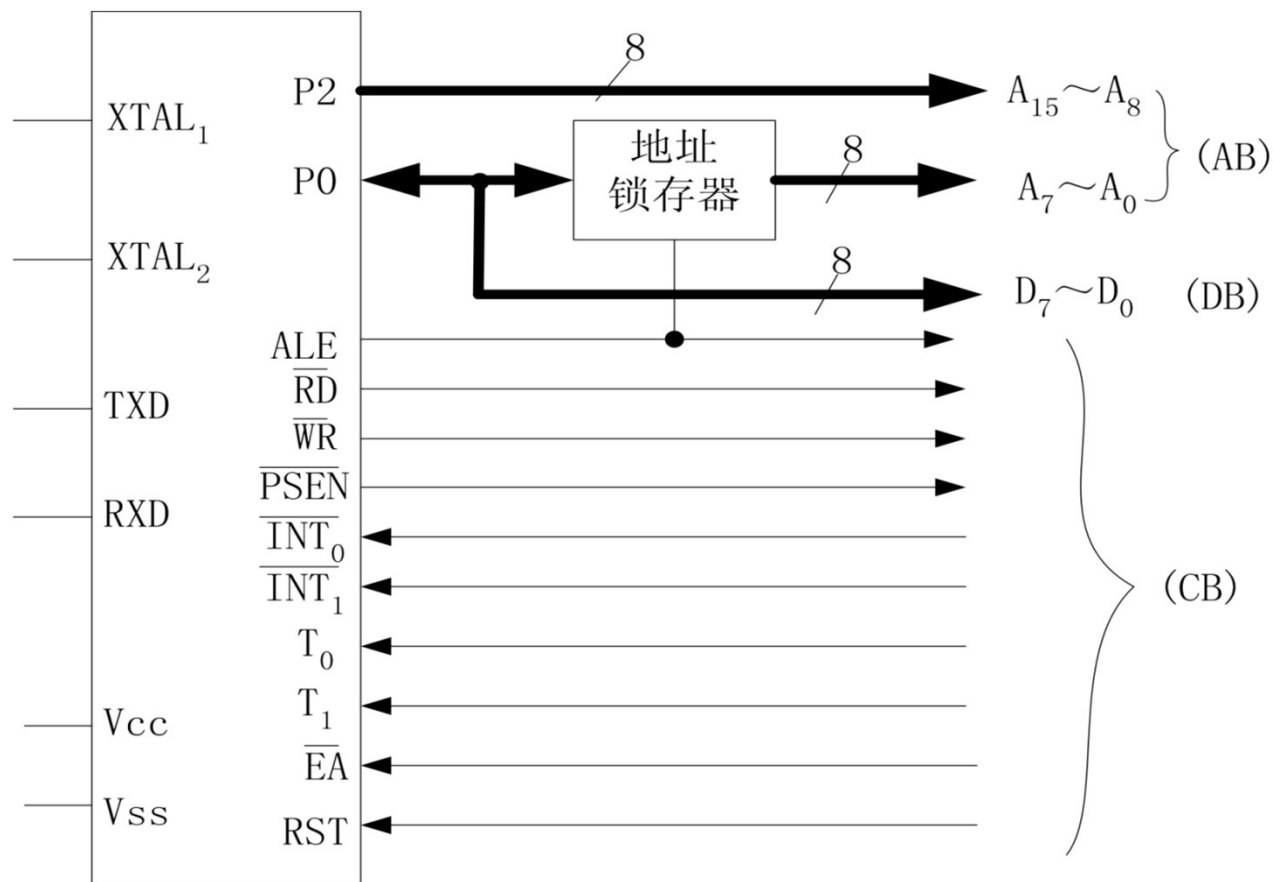
(b) 按键电平复位电路

示例：MCS-51单片机最小应用系统（8051/8751）



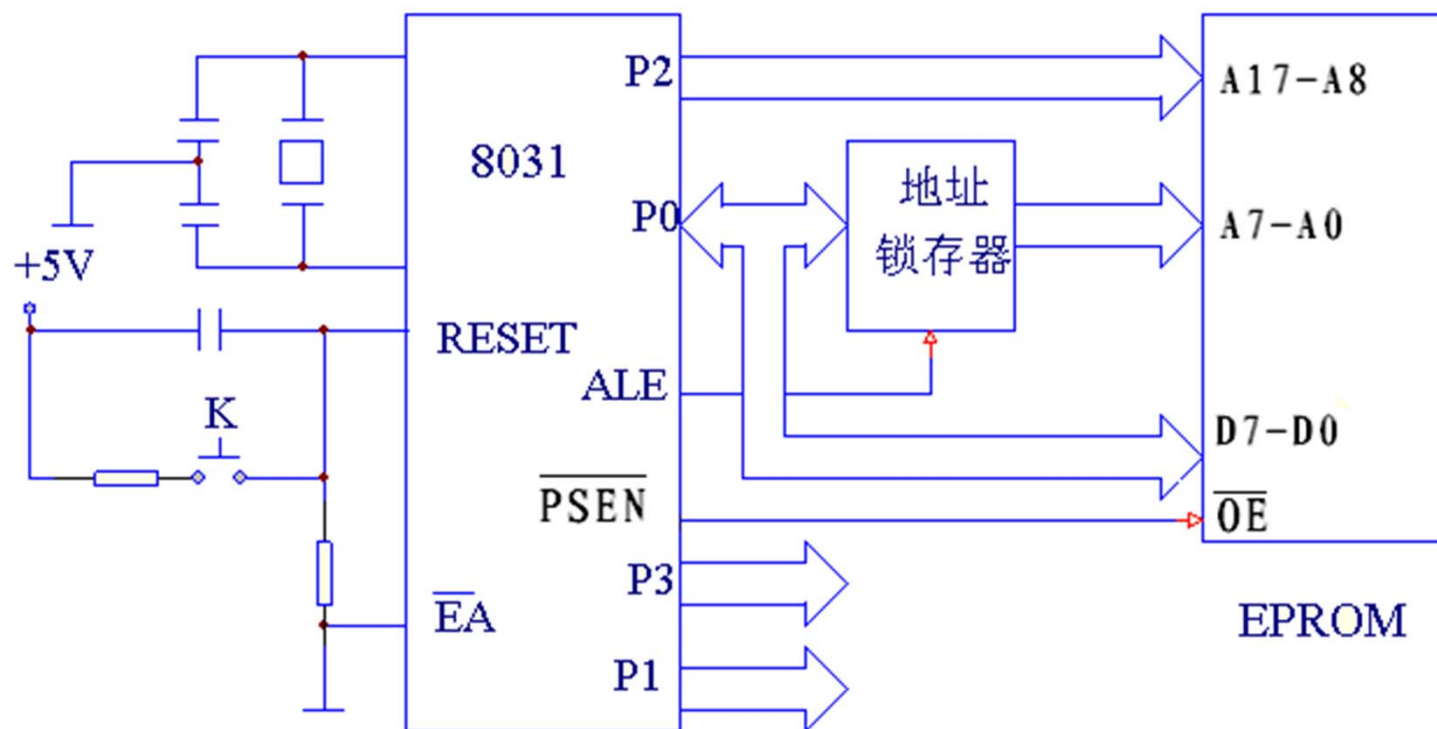
8051/8751最小应用系统

示例： MCS-51单片机的片外总线结构



MCS51单片机总线引脚结构

示例： MCS-51单片机最小应用系统（8031）

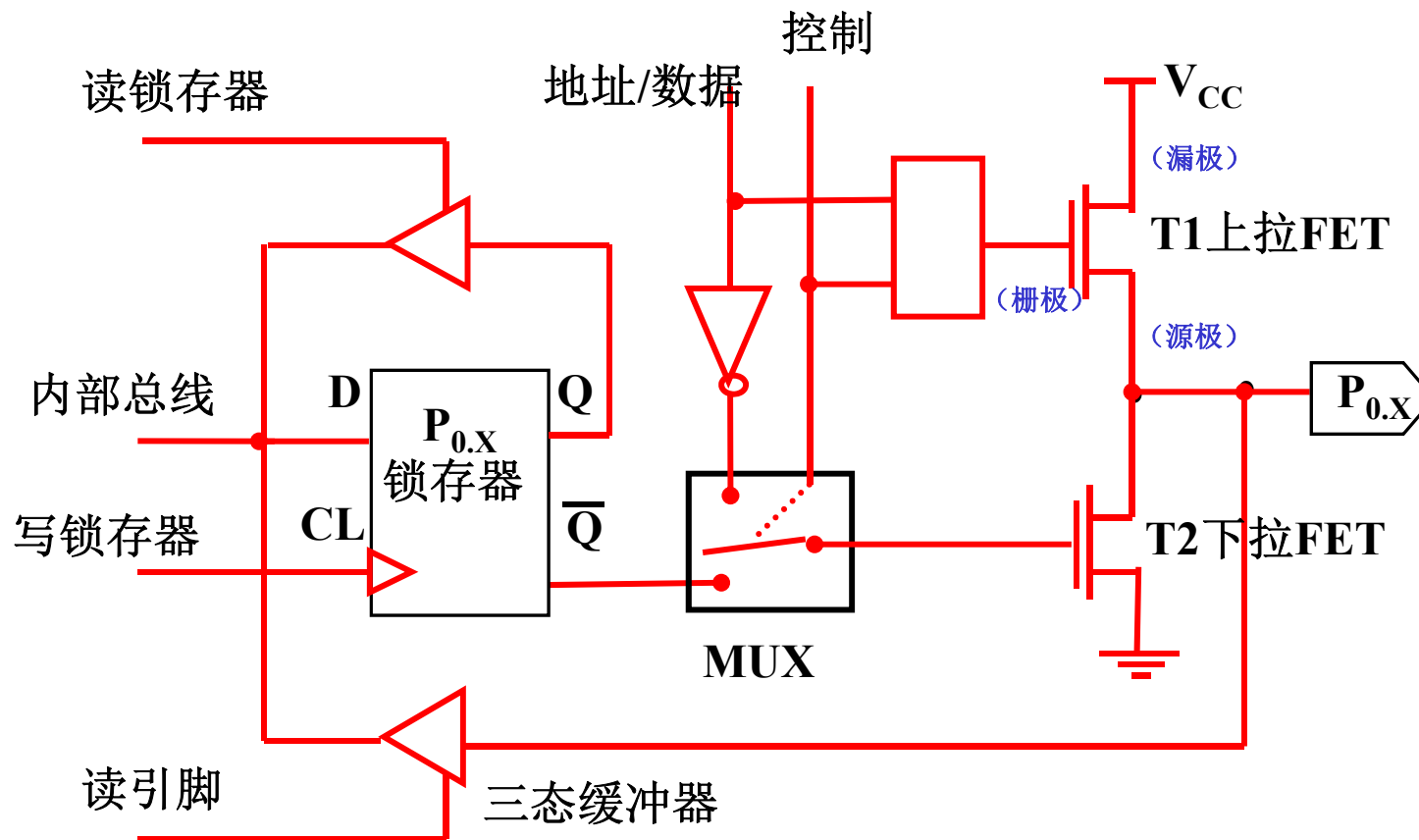


8031最小应用系统

1.6 输入/输出（I/O）端口

- MCS-51单片机有4个准双向（作为I/O输入时，需要先把口位锁存器置1，使输出下拉FET截止，故称为准双向口）并行的8位I/O口P0～P3，P0口为三态双向口，可驱动8个TTL电路输入，P1、P2、P3内部带有上拉网络，其负载能力为4个TTL电路输入。
- 每个口对应：1个口锁存器、一个输出控制电路、一个输出驱动器、二个输入三态缓冲器（同向）。
- 系统需要访问外部ROM或RAM（I/O口）时，由P0、P2输出地址，P0分时传送数据；
- P0~P3均可用作普通I/O口（为准双向口，且P0口线须外接上拉电阻）。

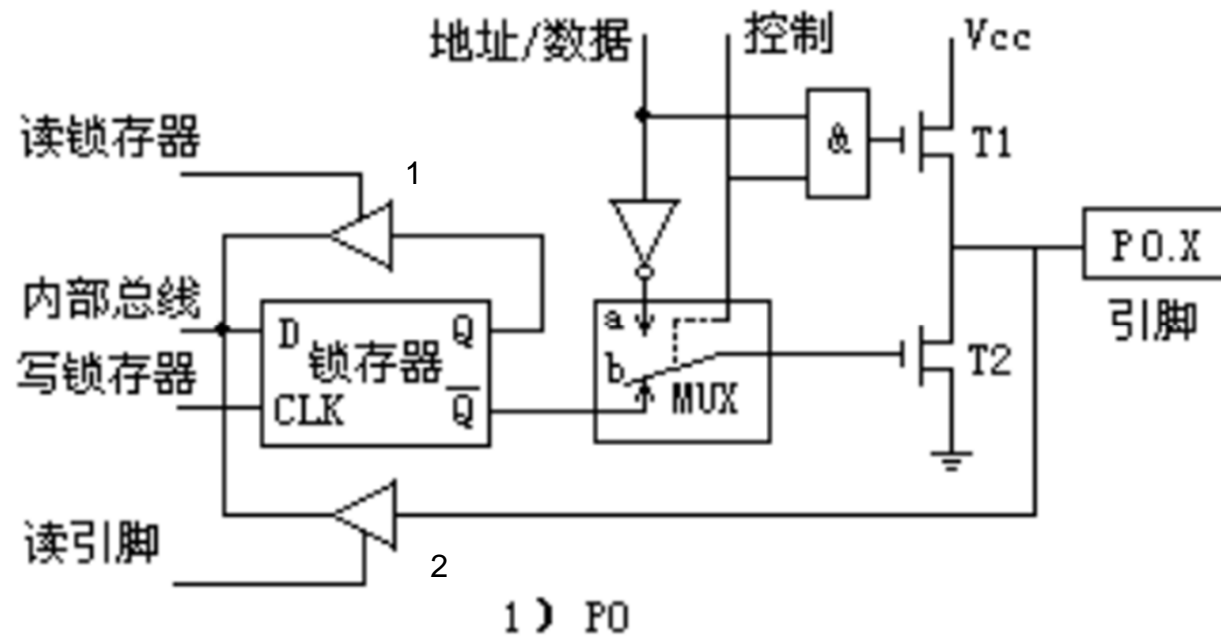
1.6.1 P0口



P0口的一位结构图

P0口结构与功能

P0口由1个输出锁存器、2个三态输入缓冲器、1个输出驱动电路和1个输出控制电路组成。输出驱动电路由一对FET（场效应管）组成，其工作状态受输出控制电路的控制；输出控制电路由一个与门电路，1个反相器和1个多路开关MUX组成。



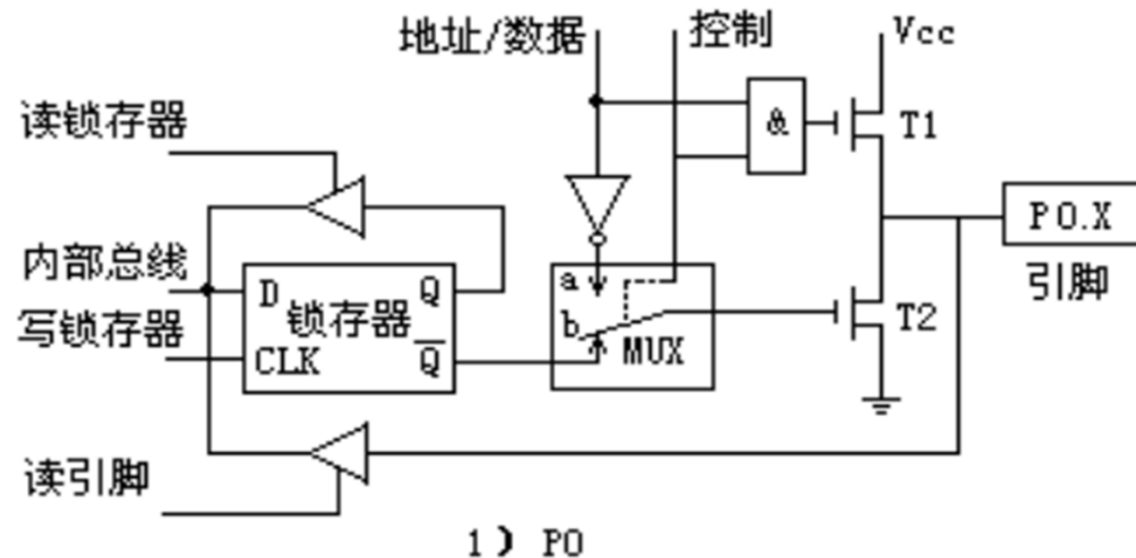
P0口结构与功能

多路开关MUX的位置由CPU发出的控制信号决定。

(1) 当P0口做一般I / O端口使用时：

CPU内部发出控制电平“0”信号，封锁与门，T1截止，同时多路开关MUX把输出锁存器 \overline{Q} 端与T2的栅极接通。

内部数据总线上的信息，由写脉冲锁存至输出锁存器，P0口引脚输出电平与Q端同步，在P0引脚上出现的数据（经过2次反相），正好是内部总线上的数据，输出驱动级是漏极开路电路，P0口作一般I/O口使用时须外接上拉电阻。



P0口结构与功能

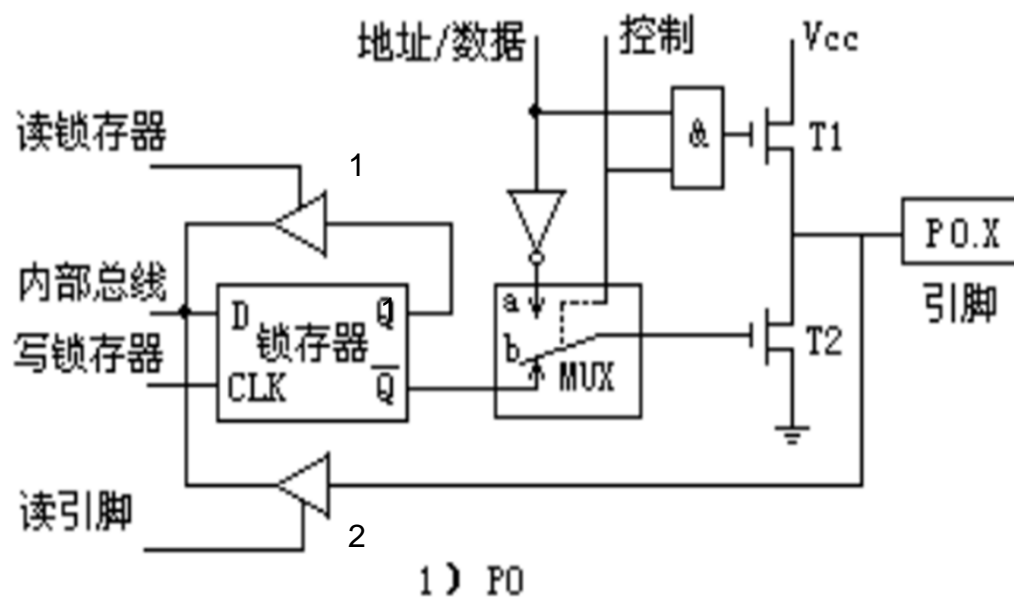
(1) 当P0口做一般I / O端口使用时:

执行一条读端口输入的指令时，产生读引脚信号，将下面的缓冲器2打开，端口引脚上的数据，经缓冲器2读入内部数据总线。

缓冲器1并不能直接读取端口引脚上的数据，而是读取输出锁存器Q端的数据，Q端数据与引脚上的数据是一致的，结构上这样的安排是为了适应“**读-修改-写**”一类指令的需要。

端口进行输入操作前，应先向端口输出锁存器写入“1”，使 $\overline{Q}=0$ ，则输出级的两个FET管均截止，引脚处于悬空状态，变为高阻抗输入。否则，端口引脚电平会被钳在“0”电平上，输入数据不能正确读入。（故称为“准双向口”）

系统复位时，锁存器初始化为1，保证复位后，直接将端口用作输入功能；而访问外部存储器时，CPU会在必要时自动向P0口写FFH。



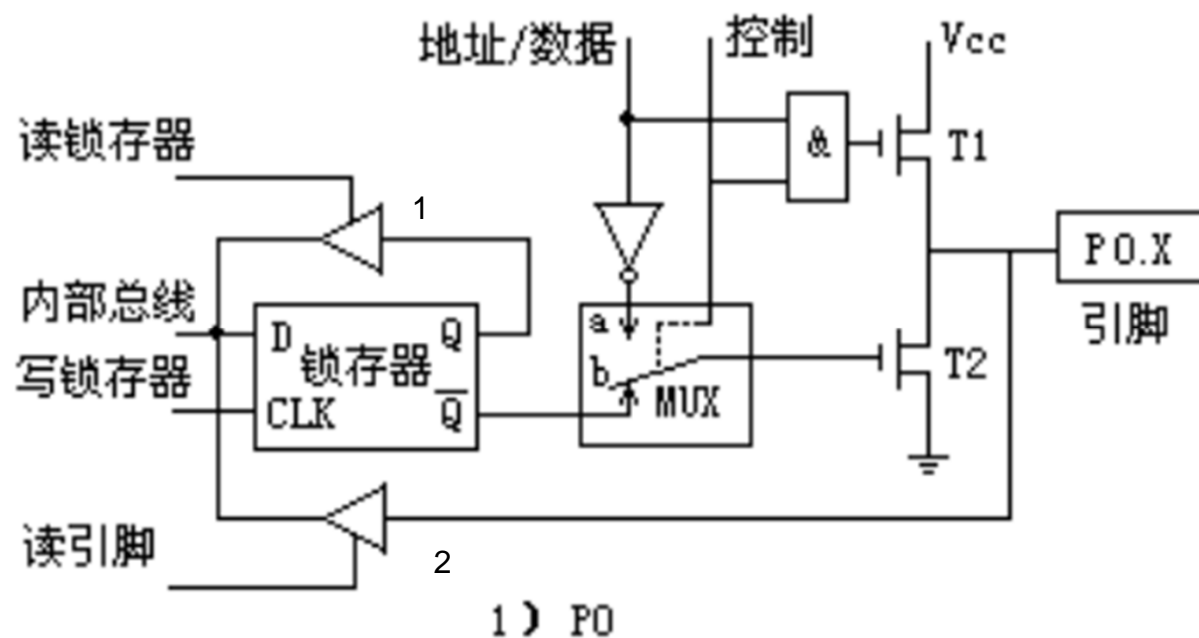
P0口结构与功能

(2) 当P0口做地址/数据总线时:

CPU内部发出控制电平“1”信号，打开与门，多路开关MUX打向上方，把非门输出端和T2的栅极接通。此时可分为两种情况：

在CPU执行输出指令时，P0口引脚分时输出地址/数据信息。

在CPU执行输入指令时，P0口输入数据，此时输入的数据是从引脚通过输入缓冲器2，进入内部总线。

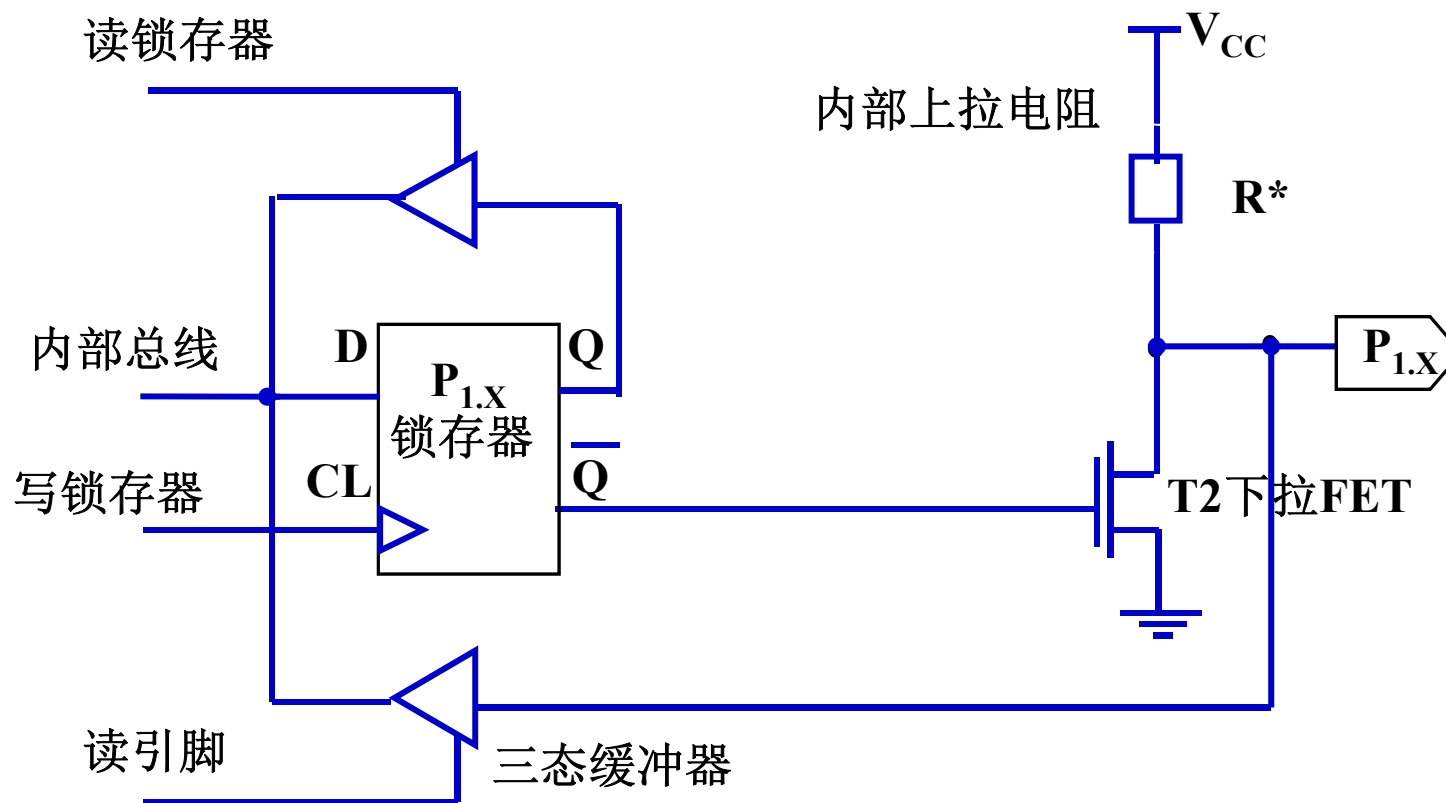


P0口特点总结

- (1) P0口可作通用I / O口使用，又可作地址/数据总线口；
- (2) P0既可按字节寻址，又可按位寻址；
- (3) 作通用I / O 口输出时：是开漏输出；
- (4) 作地址/数据总线口时，P0是一真正双向口；

作通用I / O口时，只是一个准双向口，各引脚需在片外接上拉电阻。

1.6.2 P1口



P1口的一位结构图

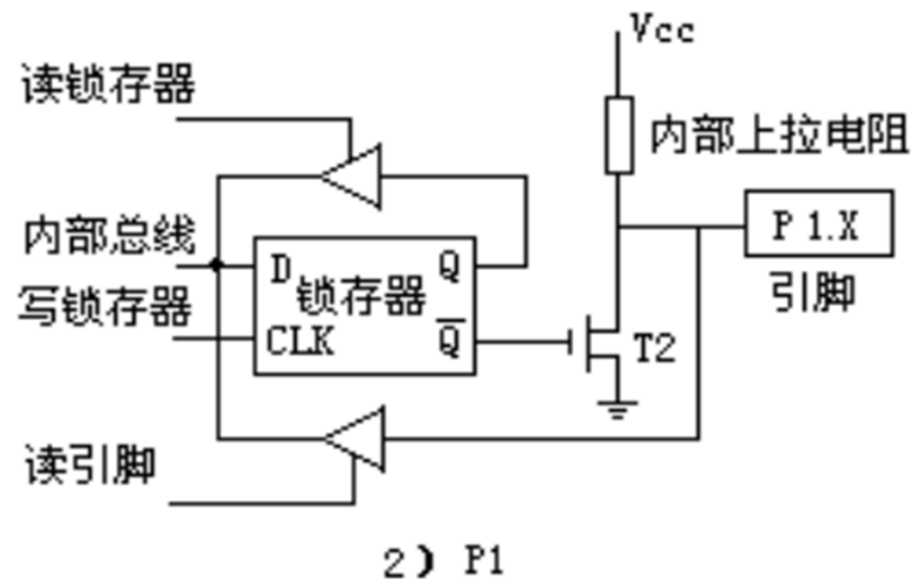
P1口结构与功能

P1口在结构上与P0口的区别是：没有多路开关MUX和控制电路部分；输出驱动电路部分与P0也不相同，只有一个FET场效应管，同时内部带上拉电阻，此电阻与电源相连，用来将I/O端口上拉成高电平。

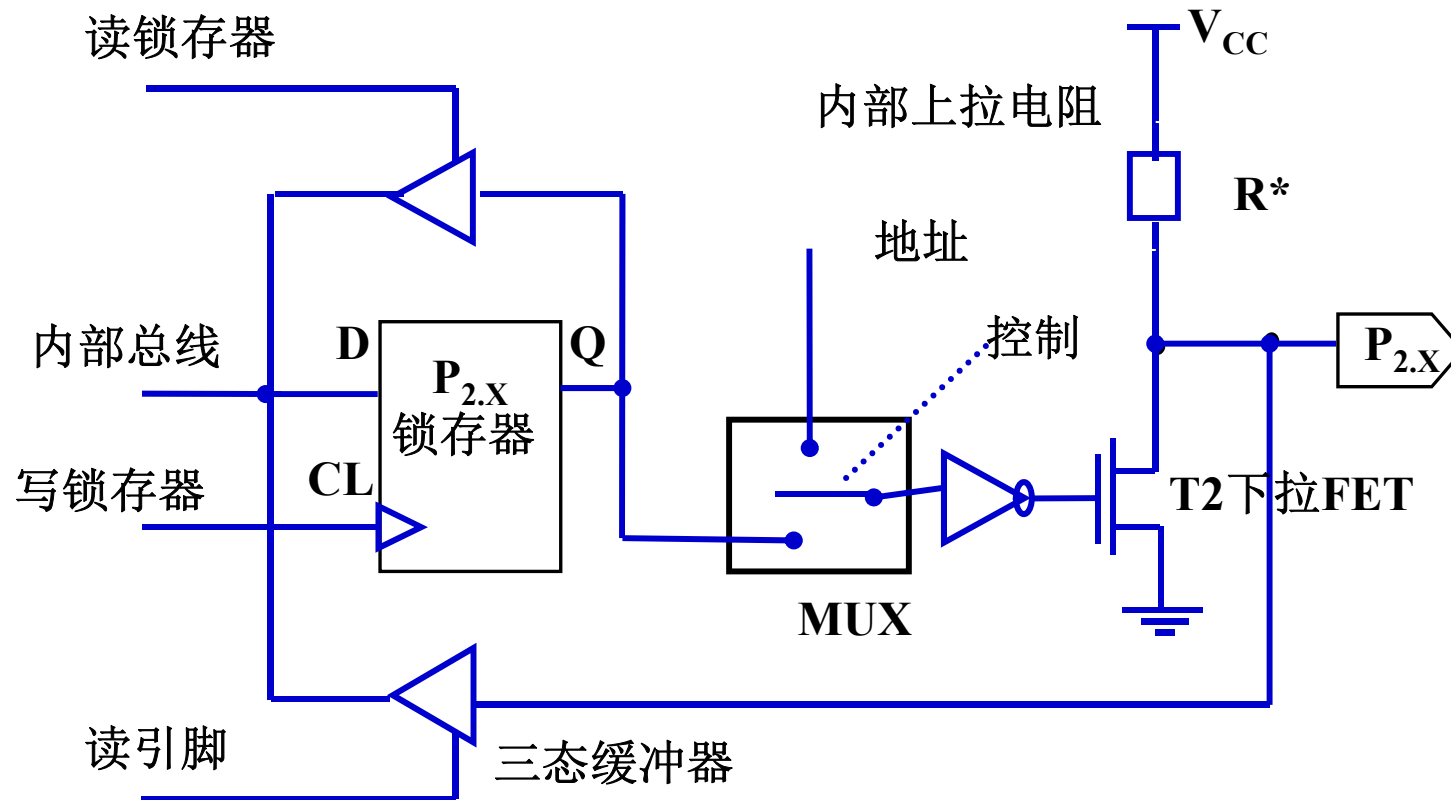
P1口可做通用双向I / O口使用，
是准双向口；

当P1口输出高电平时，不必再外接上拉电阻。当端口用做输入时，须先向对应的输出锁存器写入“1”，使FET截止。然后再读端口引脚。

对52子系列，P1.0、P1.1属双功能口结构，具有第二功能。



1.6.3 P2口



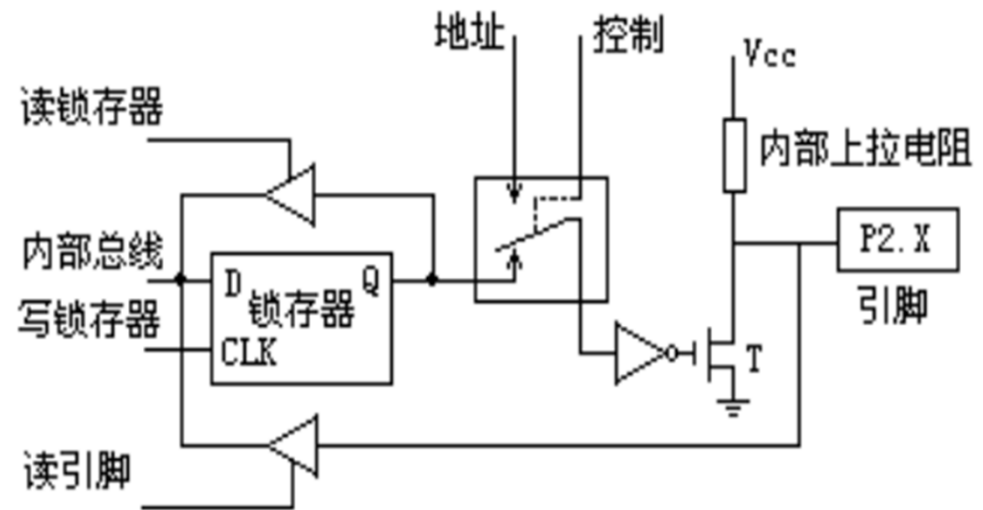
P2口的一位结构图

P2口结构与功能

P2口的位结构中上拉电阻的结构与P1相同，但比P1口多了一个输出转换多路控制部分。

P2口用作地址总线时，在CPU的控制下，转换开关MUX倒向内部地址线一端，此时P2口用于输出高8位的地址，即访问外部ROM、RAM（I/O口）时的高8位地址（PCH或DPH）。

P2口用作通用I/O口时，多路开关MUX倒向锁存器输出Q端，P2引脚的数据与内部总线数据相同。此时，P2口是一个准双向口。由输出转为输入时，需要先将口锁存器置1，使下拉FET截止；系统复位时，锁存器初始化为1，保证复位后直接将口线用作输入功能。



3) P2

P2口结构与功能

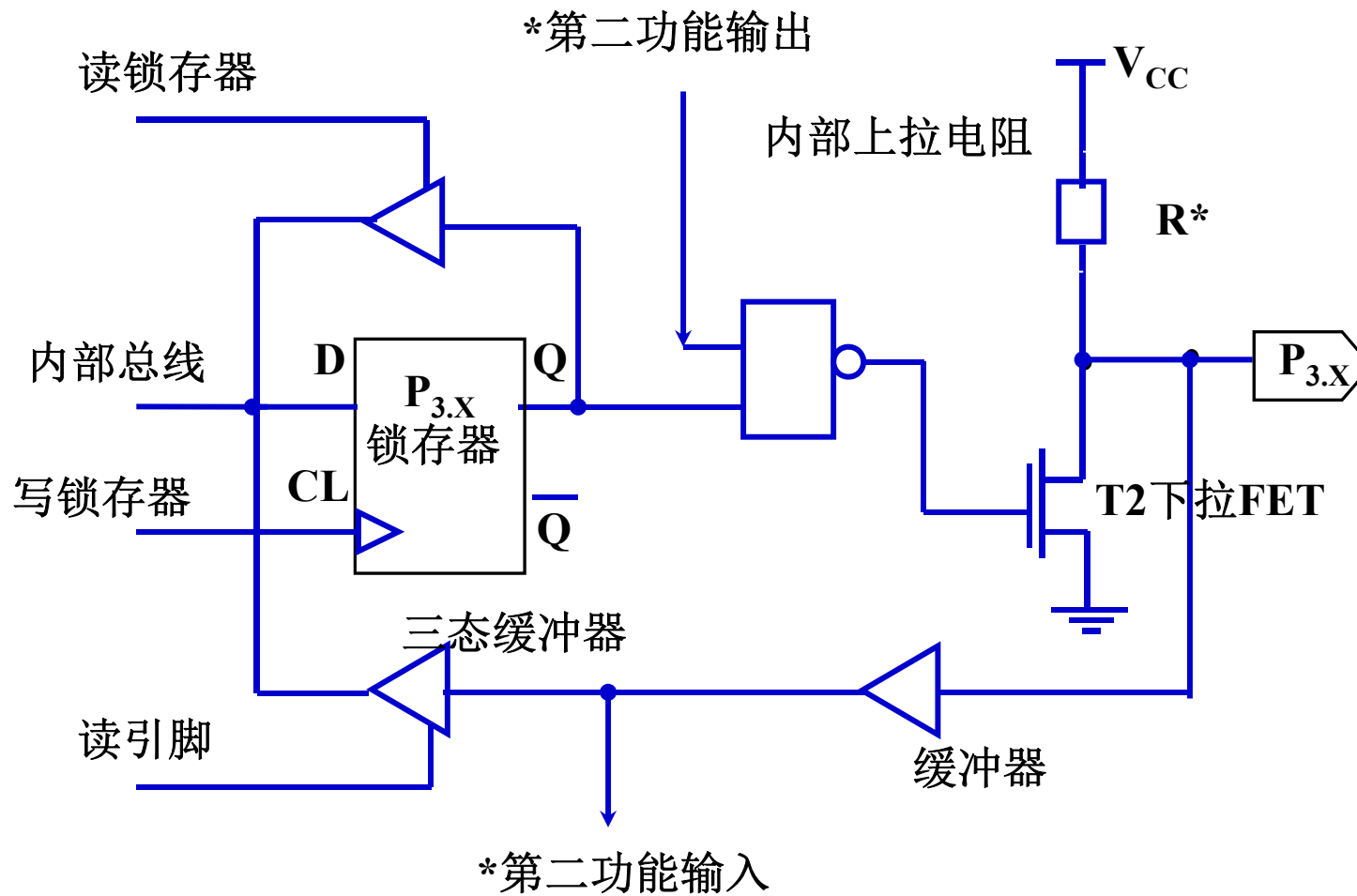
当P2口用作地址总线时：

(1) 当系统接外部程序存储器（ROM）时（8031必须外扩程序存储器），P2口将被占用，用于输出PCH（高8位地址）。由于访问外部ROM连续不断，P2口就不能再用作通用I/O口；

(2) 当系统不接外部ROM而接有外部RAM时，

- 若外部RAM容量为256B，则使用MOVE @Ri类指令，由P0口送出8位地址，P2口还可作为通用I/O口使用；
- 若外部RAM容量达64KB，则使用MOVE @DPTR类指令，由P0送低8位，P2送高8位地址，访问外部RAM周期结束后，P2锁存器的内容又会重现在P2引脚上；这样根据访问外部RAM频繁程度，P2口仍可在一定限度内用作通用I/O口；
- 当256B< 外部RAM容量 <64KB时，可考虑用软件的方法传送几位高位地址，仍使用MOVX @Ri指令，以节省部分或全部P2口线作通用I/O口使用。

1.6.4 P3口



P3口的一位结构图

P3口结构与功能

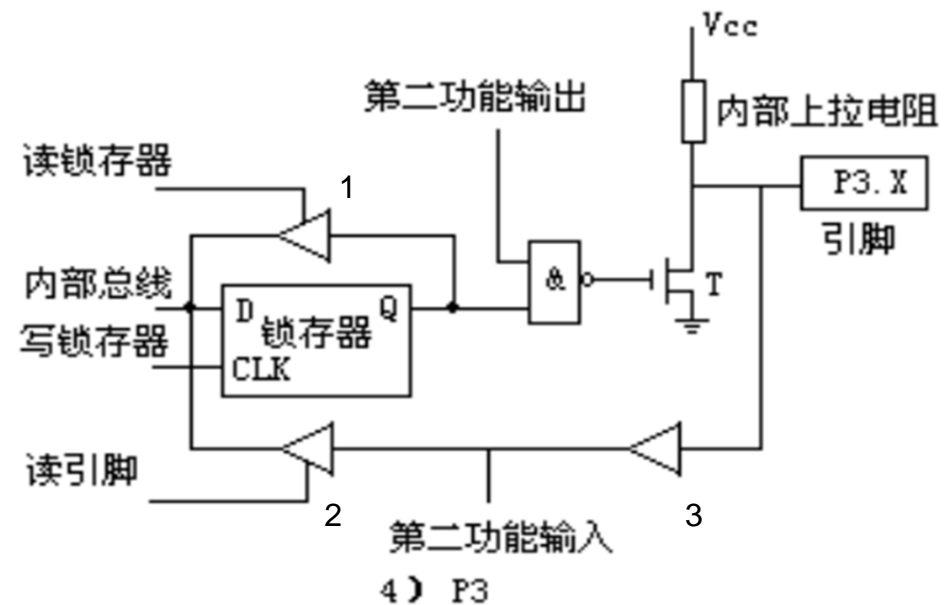
P3口与P1口的输出驱动部分及内部上拉电阻相同，比P1口多了一个第二功能控制部分的逻辑电路（由一个与非门和一个输入缓冲器组成）。

P3口用作通用I/O口时，第二功能输出线保持高电平时，打开与非门，锁存器Q端输出，可以通过与非门送至FET管，输出到引脚端。

输入时，引脚数据通过三态缓冲器3和2，在读引脚选通控制下进入内部总线。此时，锁存器Q端应置“1”，保持下拉FET截止。

P3口作第二功能输出时，锁存器Q端置“1”，打开与非门，第二功能输出端内容通过“与非门”和FET送至端口引脚。

作第二功能输入时，第二功能输出线和该位锁存器Q均置“1”，以保持FET截止；端口引脚的第二功能信号，通过缓冲器3送到第二功能输入端。



P3口的第二功能表

引脚	第二功能
P3.0	RxD（串口输入）
P3.1	TxD（串口输出）
P3.2	$\overline{\text{INT0}}$ （外部中断0输入）
P3.3	$\overline{\text{INT1}}$ （外部中断1输入）
P3.4	T0（定时器0计数输入）
P3.5	T1（定时器1计数输入）
P3.6	$\overline{\text{WR}}$ （外部数据存储器写选通）
P3.7	$\overline{\text{RD}}$ （外部数据存储器读选通）

I/O端口小结

P0口：地址低8位与数据线分时使用端口；

P1口：按位可编址的输入、输出端口；

P2口：地址高8位输出口（PC 或 DPTR高8位），或通用I/O口；

P3口：双功能口，若不用第二功能，也可作通用I/O口。

按三总线划分：

地址线：P0低8位地址，P2高8位地址；

数据线：P0输入、输出8位数据；

控制线：P3口的8位加上 PSEN、ALE 等共同组成控制总线。

并行I/O端口的读-修改-写操作

对于每个并行I/O口均有两种读写操作的方法：

对端口锁存器的读写操作和对引脚的读操作。在MCS-51系列单片机指令中，有些指令是读锁存器内容，有些指令是读引脚内容。

读锁存器指令，指的是从端口锁存器中读取一个值，进行运算，然后把运算结果重新写入锁存器中，这类指令称“读-修改-写”指令。通常这类指令的目的操作数为一个端口或端口的1位。

读引脚指令，指的是把端口引脚上的电平信号读到内部总线上。
如：

```
MOV P1, #0FFH
```

```
MOV A, P1
```


并行I/O端口的读-修改-写操作

例如，逻辑与指令（**ANL P0, A**），此指令的功能是先把**P0**口锁存器中的数据读入**CPU**，同累加器**A**中的数据按位进行逻辑与运算，然后把运算结果写入**P0**口。

下面这些指令都是读取锁存器中内容而不是读引脚上的数据。

ANL （逻辑与）	例： ANL P1, A
ORL （逻辑或）	例： ORL P2, A
XRL （逻辑异或）	例： XRL P3, A
MOV （传送）	例： MOV P1.2, C
CLR （清“0”）	例： CLR P1

这类指令还包括：**JBC, CPL, INC, DEC, DJNZ, SET**及位传送指令等。