

# 第2章 单片机的结构原理与简单应用

(课时：8学时)

# 教学目的

- 了解单片机的内部结构与主要型号。
- 掌握单片机引脚信号功能定义。
- 掌握单片机的存储器空间分配及各I/O口的特点。
- 掌握单片机的复位电路、时钟电路及指令时序。
- 利用单片机的I/O口实现循环灯的控制。

## 学习重点和难点

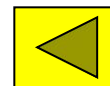
- 单片机的结构特点。
- 存储器配置与空间的分布。
- 程序状态寄存器(PSW)。
- 单片机的指令时序。

# 第2章 单片机的结构原理与 简单应用

- 2.1 单片机的结构与原理
- 2.2 单片机的存储器
- 2.3 单片机的输入/输出端口
- 2.4 时钟电路
- 2.5 复位电路
- 2.6 常用单片机功能简介
- 本章小结
- 习题

## 2.1 单片机的结构与原理

- 2.1.1 8051单片机的基本组成
- 2.1.2 8051单片机的内部结构
- 2.1.3 8051单片机的引脚及功能
- 2.1.4 单片机的工作原理



## 2.1.1 8051单片机的基本组成

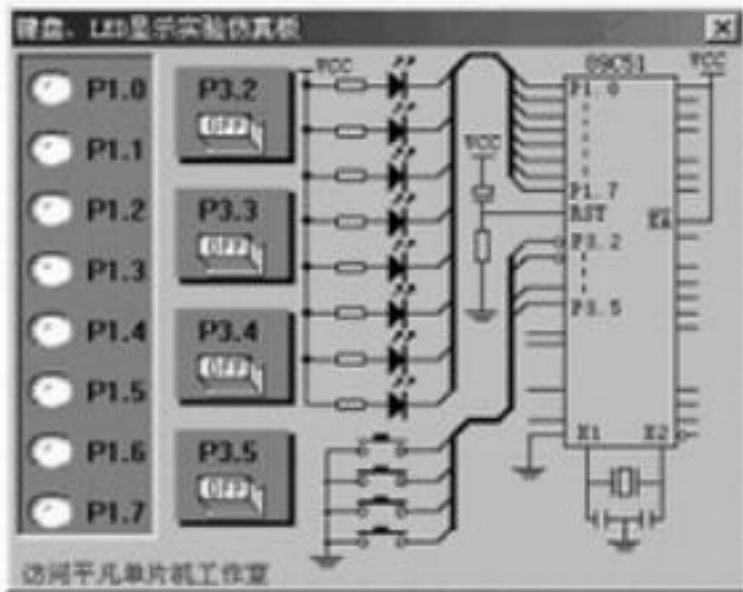
- **MCS-51**是美国Intel公司的8位高档单片机系列，也是我国目前应用最为广泛的一种单片机系列。**8051/80C51**是整个**MCS-51**系列单片机的核心，该系列其他型号的单片机都是在这—内核的基础上发展起来的。
- **MCS-51**单片机系列分为**51**和**52**子系列，并以芯片型号的末位数字加以标识。其中，**51**子系列是基本型，而**52**子系列是增强型。
- 单片机型号带有字母“**C**”的，表示该单片机采用的是**CHMOS**工艺，具有低功耗的特点。  
8051的功耗为630mW，而80C51的功耗只有120mW。



## 8031芯片实照



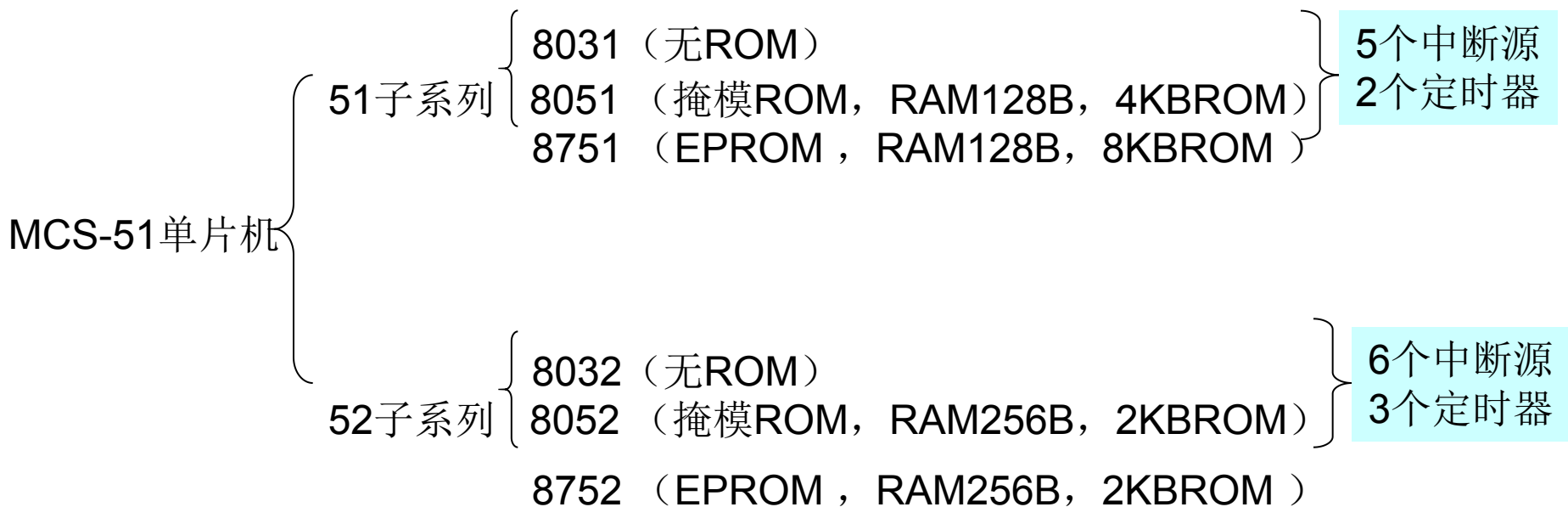
# 89C51



# 80C51单片机晶振

# MCS-51系列单片机技术参数表

| 子系列    | 片内ROM形式 |       |       | 片内ROM | 片内RAM | 寻址范围   | I/O特性 |     |     | 中断源 |
|--------|---------|-------|-------|-------|-------|--------|-------|-----|-----|-----|
|        | 无       | ROM   | EPROM |       |       |        | 定时器   | 并行口 | 串行口 |     |
| 51 子系列 | 8031    | 8051  | 8751  | 4KB   | 128B  | 2×64KB | 2×16  | 4×8 | 1   | 5   |
|        | 80C31   | 80C51 | 87C51 | 4KB   | 128B  | 2×64KB | 2×16  | 4×8 | 1   | 5   |
| 52 子系列 | 8032    | 8052  | 8752  | 8KB   | 256B  | 2×64KB | 3×16  | 4×8 | 1   | 6   |
|        | 80C32   | 80C52 | 87C52 | 8KB   | 256B  | 2×64KB | 3×16  | 4×8 | 1   | 6   |



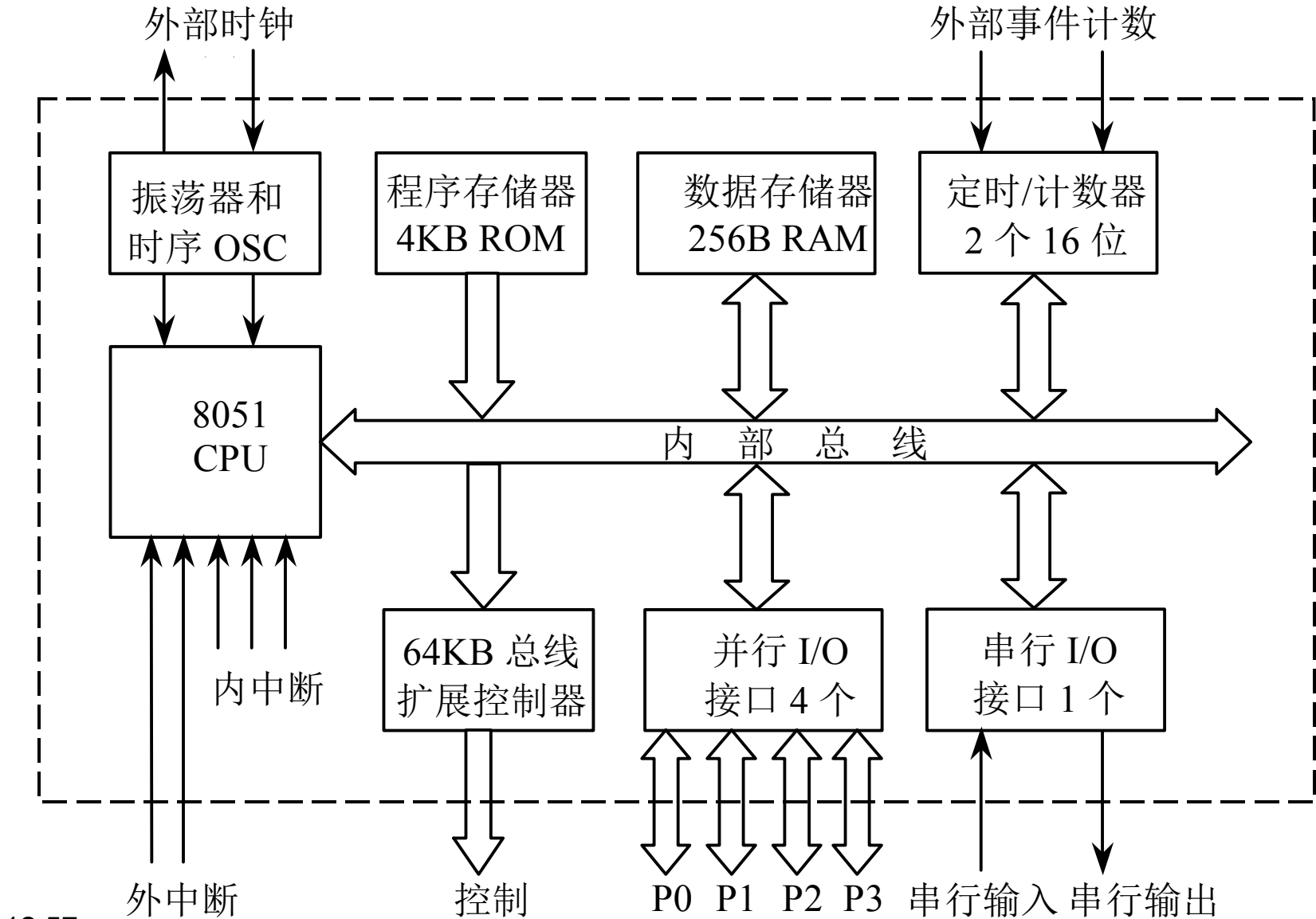
51、52子系列相同部分

- 指令系统
- 引脚

## 2.1.1 8051单片机的基本组成

- 8051单片机的结构框图如下图所示，可以看出，在一块芯片上集成了一个微型计算机的主要部件，它包括以下几部分：
  - 1个8位微处理器(CPU)。
  - 1个时钟电路。
  - 4KB程序存储器。
  - 256B数据存储器。
  - 2个16位定时/计数器。
  - 64KB扩展总线控制电路。
  - 4个8位并行I/O接口P0~P3。
  - 1个全双工串行I/O接口。
  - 5个中断源，其中包括2个优先级嵌套中断。

# 8051单片机结构框图



为了更方便的说明其工作原理将总体结构图分成6个部分来分别介绍：

- 1. CPU
  - 2. 存储器
  - 3. 并行I/O端口
  - 4. 中断系统
  - 5. 定时器/计数器
  - 6. 串行通信口
- 

本节先介绍前3部分，其他在后面各章节中分别介绍

让我们顺便看看串、并口方面的知识。

基本的通讯方式有并行通讯和串行通讯两种。一条信息的各位数据被同时传送的通讯方式称为并行通讯。并行通讯的特点是：各数据位同时传送，传送速度快、效率高，但有多少数据位就需多少根数据线，因此传送成本高，且只适用于近距离（相距数米）的通讯。一条信息的各位数据被逐位按顺序传送的通讯方式称为串行通讯。串通讯的特点是：数据位传送，传按位顺序进行，最少只需一根传输线即可完成，成本低但送速度慢。串行通讯的距离可以从几米到几千米。

根据信息的传送方向，串行通讯可以进一步分为单工、半双工和全双工三种。信息只能单向传送为单工；信息能双向传送但不能同时双向传送称为半双工；信息能够同时双向传送则称为全双工。

串行通讯又分为异步通讯和同步通讯两种方式。在单片机中，主要使用异步通讯方式。

**MCS\_51**单片机有一个全双工串行口。全双工的串行通讯只需要一根输出线和一根输入线。数据的输出又称发送数（**TXD**），数据的输入又称接收数据（**RXD**）。串行通讯中主要有两个技术问题，一个是数据传送、另一个是数据转换。数据传送主要解决传送中的标准、格式及工作方式等问题。数据转换是指数据的串并行转换。具体说，在发送端，要把并行数据转换为串行数据；而在接收端，却要把接收到的串行数据转换为并行数据。

# 单工、半双工和全双工的定义

如果在通信过程的任意时刻，信息只能由一方A传到另一方B，则称为单工。

如果在任意时刻，信息既可由A传到B，又能由B传A，但只能有一个方向上的传输存在，称为半双工传输。

如果在任意时刻，线路上存在A到B和B到A的双向信号传输，则称为全双工。

电话线就是二线全双工信道。 由于采用了回波抵消技术，双向的传输信号不致混淆不清。双工信道有时也将收、发信道分开，采用分离的线路或频带传输相反方向的信号，如回线传输。

|         |         |         |
|---------|---------|---------|
| ----->  | <-----  | ----->  |
| A-----B | A-----B | A-----B |
|         |         | <-----  |
| 单工      | 半双工     | 全双工     |

# 中央处理器（CPU）

中央处理器(CPU)是整个单片机的核心部件。

中央处理器(CPU) { 运算器  
                          控制器  
                          位处理器（布尔处理器）

**CPU的功能:** 负责控制、指挥和调度整个单元系统协调的工作,完成运算和控制输入/输出功能等操作

## 2.1.2 8051单片机的内部结构

**CPU即中央处理器，是单片机的核心部件，是计算机的控制指挥中心。同微型计算机CPU类似，8051内部CPU由运算器和控制器两部分组成。**

- **运算器电路以算术逻辑单元 (ALU, Arithmetic Logic Unit) 为核心，由暂存器1、暂存器2、累加器 (ACC, Accumulator)、寄存器B、程序状态寄存器 (PSW, Program Status Word) 及布尔处理机共同组成。它的主要任务是完成算术运算、逻辑运算、位运算和数据传送等操作，运算结果的状态由程序状态寄存器 (PSW) 保存。**
- **控制器电路包括程序计数器 (PC)、PC增1寄存器、指令寄存器 (IR)、指令译码器 (ID)、数据指针 (DPTR)、堆栈指针 (SP)、缓冲器及定时控制电路等。控制器电路完成指挥控制工作，协调单片机各部分正常工作。**

# 1. 运算器

(1) 运算器的组成

- 算数逻辑单元
- 累加器A
- 寄存器B
- 暂存器(TEMP)
- 程序状态寄存器PSW

(2) 功能：进行算术运算和逻辑运算,也可以对单字节、半字节（4位）等数据进行操作。

## 2. 控制器

控制器是控制单片机的神经中枢。

控制器

程序计数器PC

指令寄存器IR

指令译码ID

数据指针DPTR

堆栈指针SP

RAM地址寄存器

时钟发生器

定时控制逻辑

# 3. 位处理器（布尔处理机）

- ◆MCS-51CPU内有一个一位处理机子系统
- ◆是一个完整的位单片机，只是每次处理的数据只是一位
- ◆有自己累加器CY，数据存储器（可位寻址空间）
- ◆能完成各种逻辑运算
  - 逻辑与
  - 逻辑或
  - 逻辑非
  - 异或等
- ◆用途
  - 逻辑电路的仿真
  - 开关量的控制
  - 设置状态标志位等

# 存储器

存放运算  
中间或临  
时数据等  
可扩展至  
64KB

数据存储器RAM

通用存储器RAM, 128B/256B(8051/52)

特殊功能寄存器, 21/26个(8051/52)

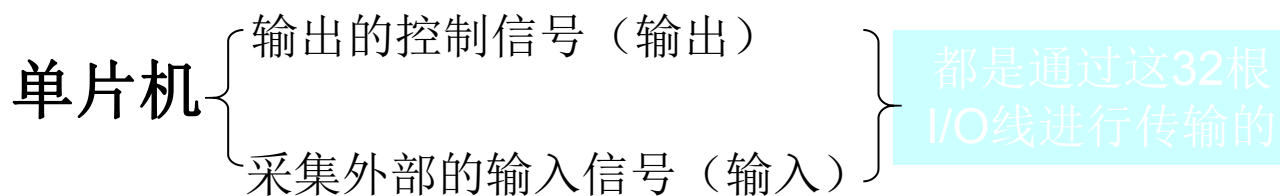
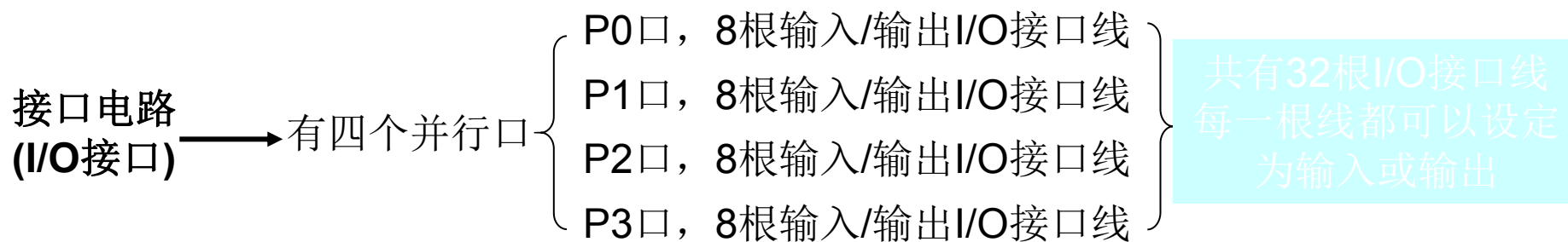
专用寄存器  
每个不得随  
便定义用途,  
如累加器A,  
定时器/计数  
器等。不能  
扩展

程序存储器ROM, 4KB/8KB(8051/52)

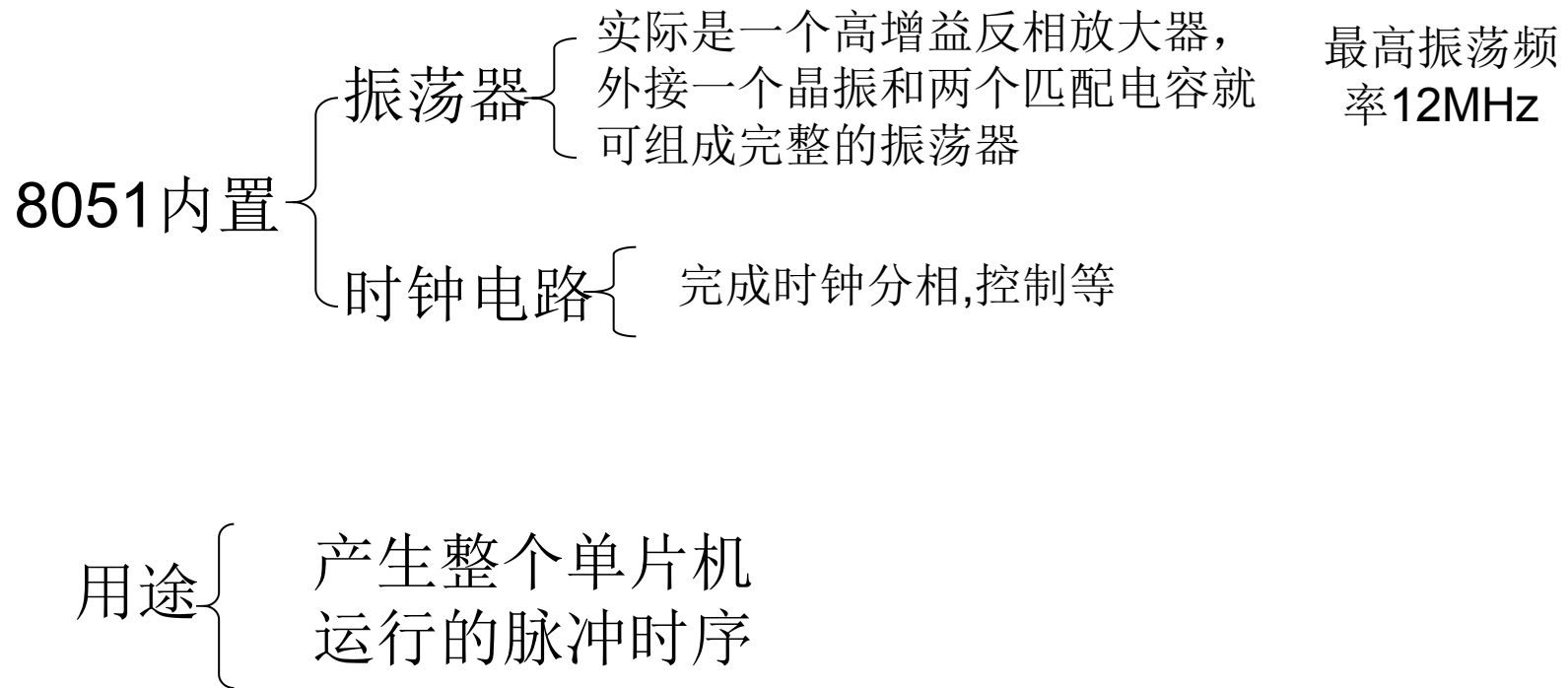
存放用户程  
序、常数及  
表格等。可  
扩展至64KB

存储器

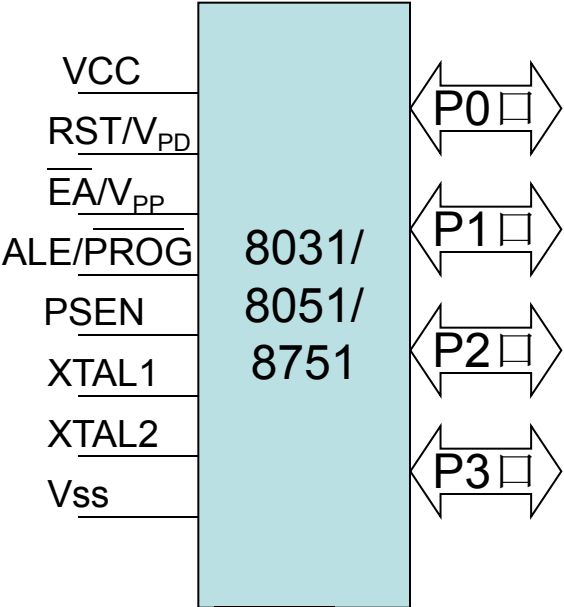
# 接口电路



# 时钟振荡电路



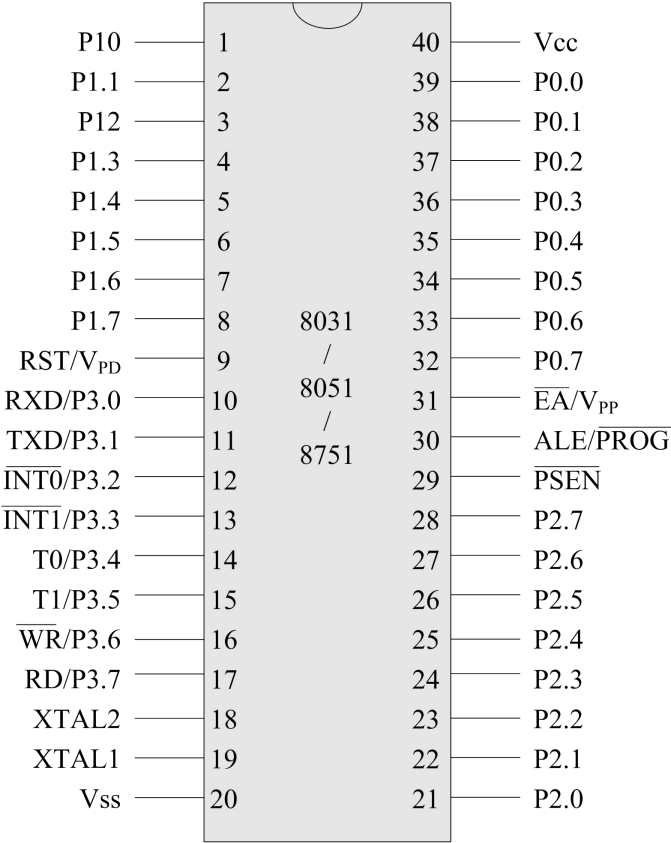
# 2.2 MCS-51的外部引脚及功能



MCS-51 单片机逻辑符号

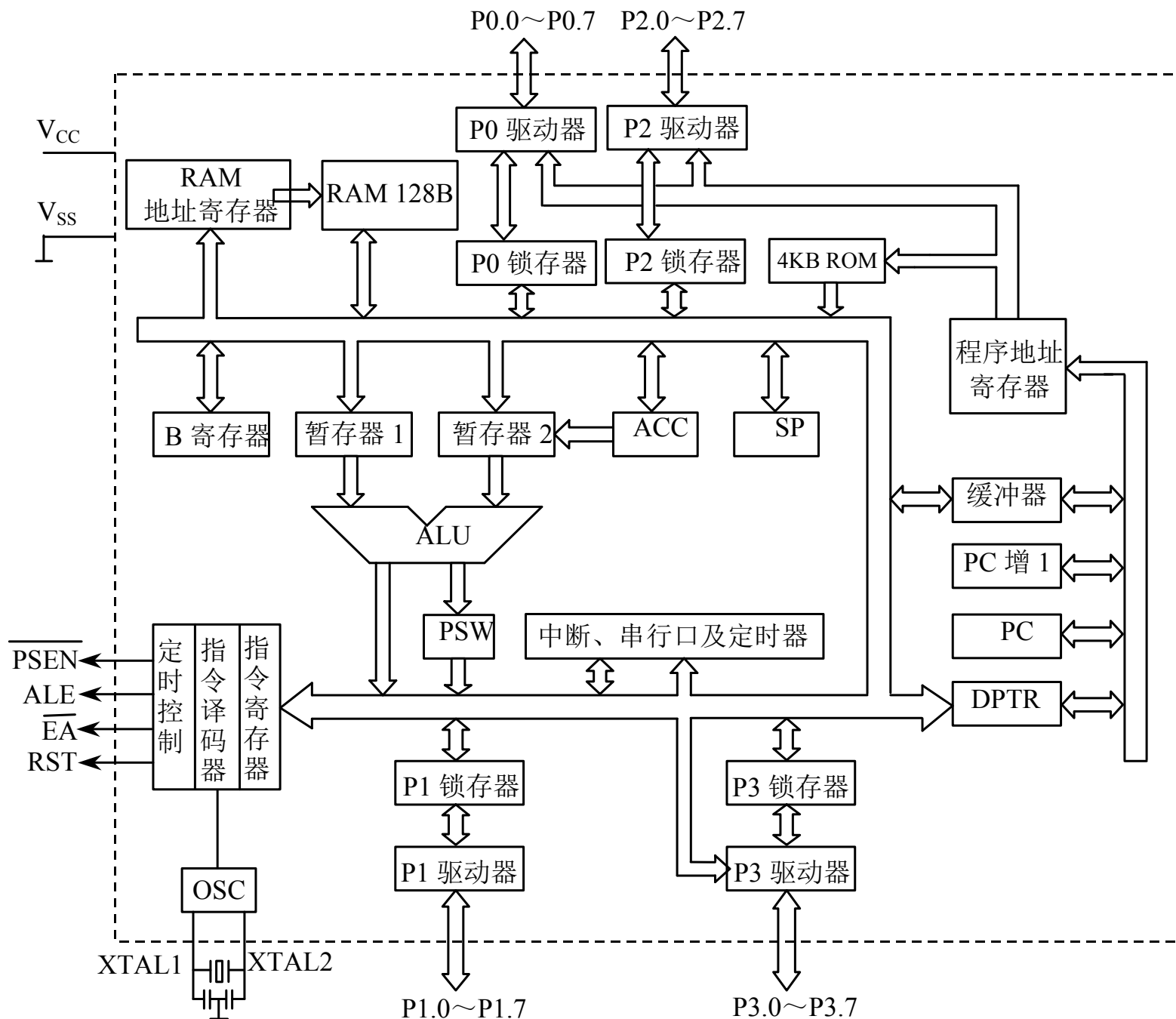
封装结构 { PLL 44脚  
                  双列直插式DIP 40脚

功能说明分三部分叙述 { I/O端口  
                                  控制引脚  
                                  电源与晶振引脚



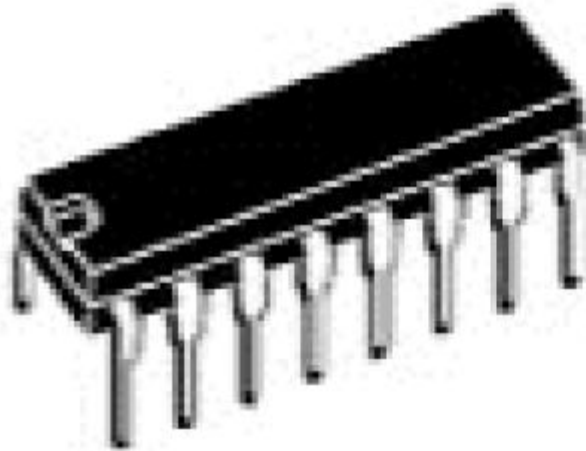
MCS-51 单片机引脚图

# 8051 单片机内部结构图



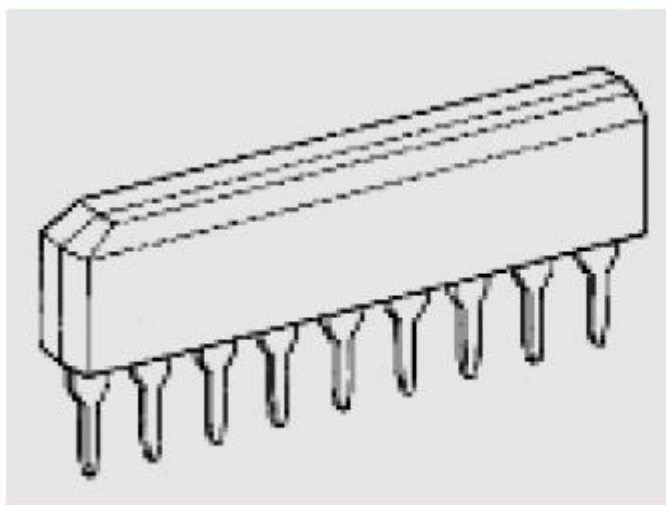
## 2.1.3 8051单片机的引脚及功能

- **MCS-51单片机的封装形式有两种，一种是双列直插式(DIP)封装，另一种是方形封装。**
- **8051的40个引脚可分为：**
  - 电源引脚2根
  - 时钟引脚2根
  - 控制引脚4根
  - I/O引脚32根
- **由于8051单片机是高性能的单片机，同时受到引脚数目的限制，所以有部分引脚具有第二功能。**



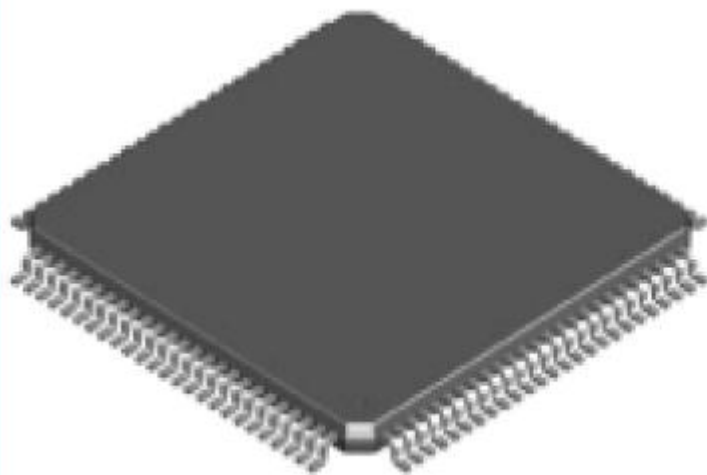
DIP  
Dual      Inline  
Package

双列直插封装



SIP  
Single In-Line  
Package

单列直插封装



TQFP 100L

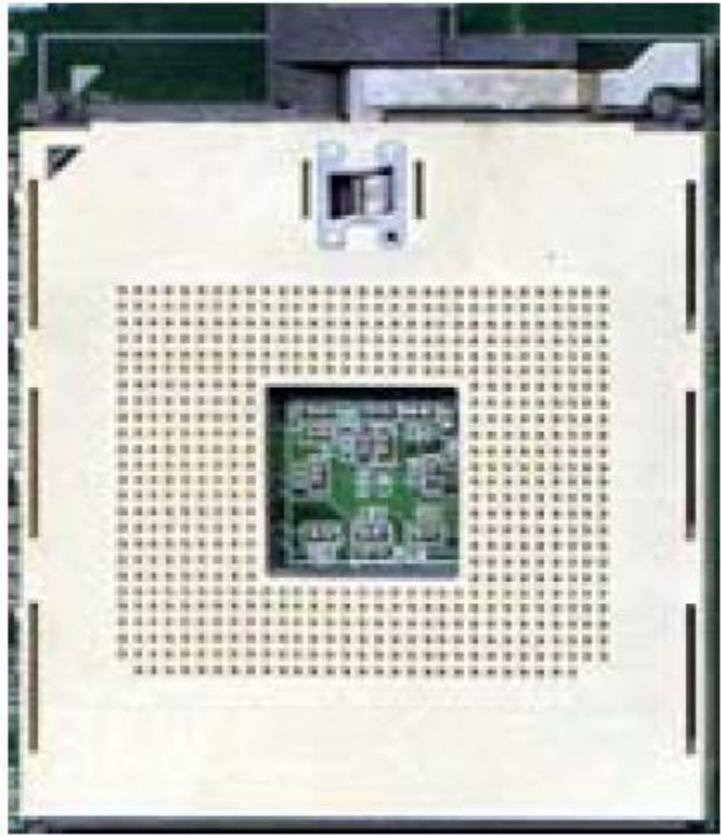
方形扁平封装



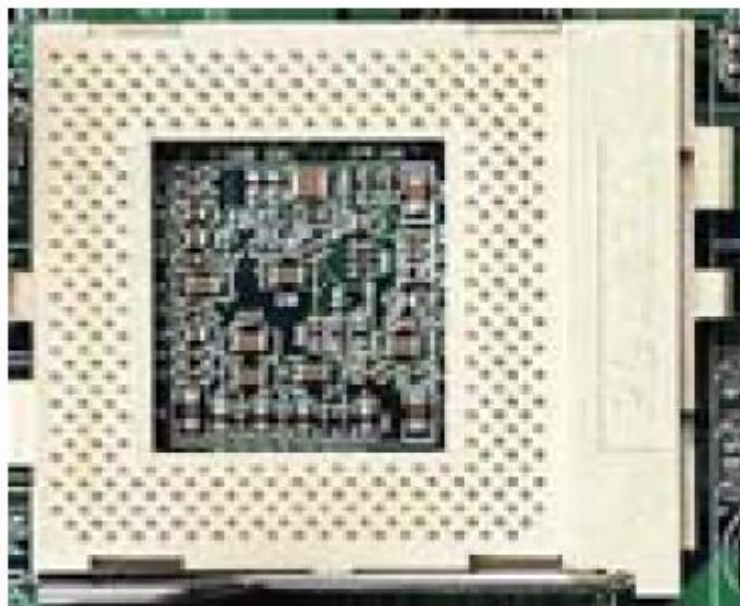
SO DIMM  
Small Outline  
Dual In-line  
Memory  
Module



SIMM30  
Single In-line  
Memory  
Module



Socket 603  
Foster



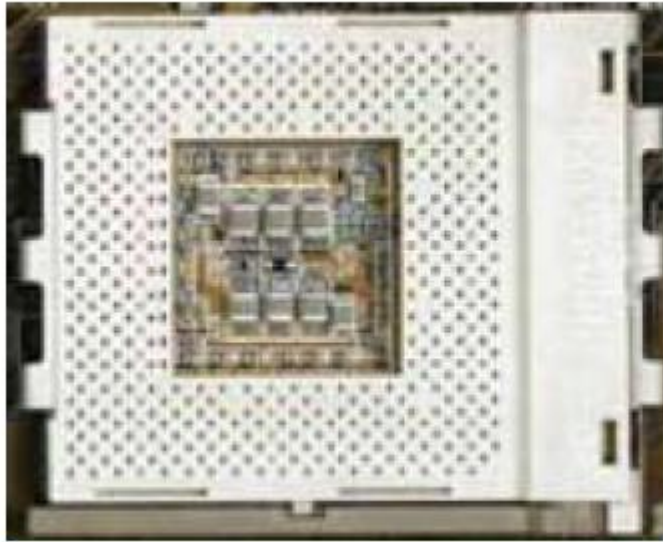
SOCKET 370  
For intel 370  
pin PGA  
Pentium III &  
Celeron CPU



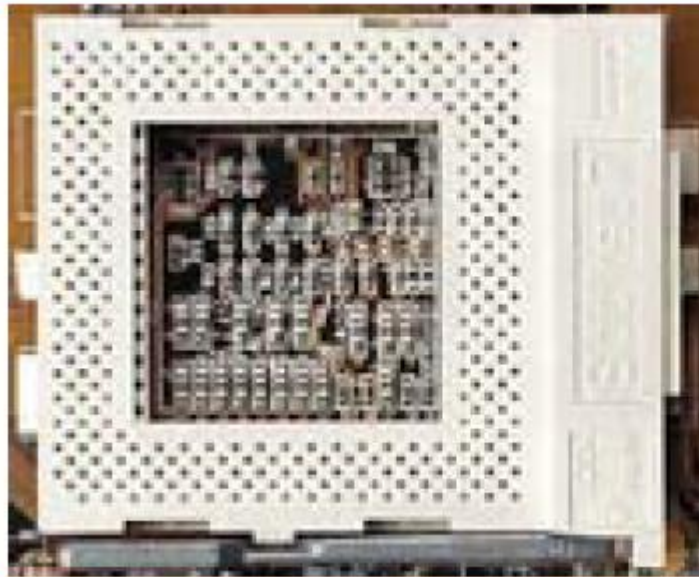
PCI 64bit 3.3V  
Peripheral  
Component  
Interconnect



SIMM72  
Single In-line  
Memory  
Module



SOCKET  
462/SOCKET A  
For PGA AMD  
Athlon &  
Duron CPU



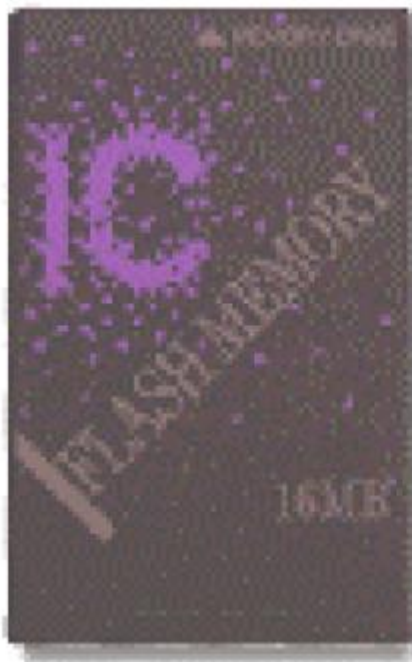
SOCKET 7  
For intel  
Pentium &  
MMX Pentium  
CPU



SLOT 1  
For intel  
Pentium II  
Pentium III &  
Celeron CPU



SLOT A  
For AMD  
Athlon CPU

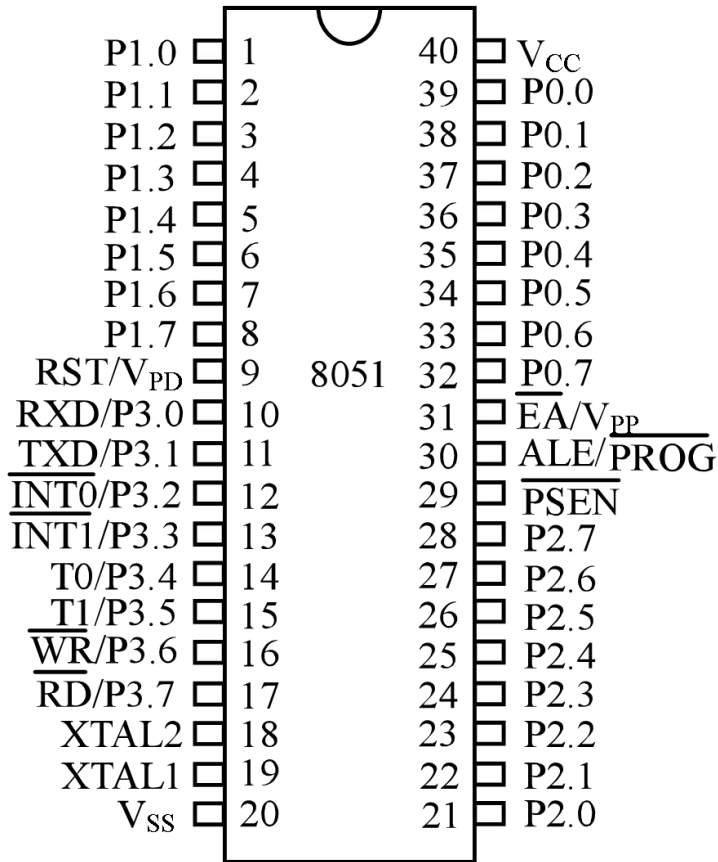


PCMCIA

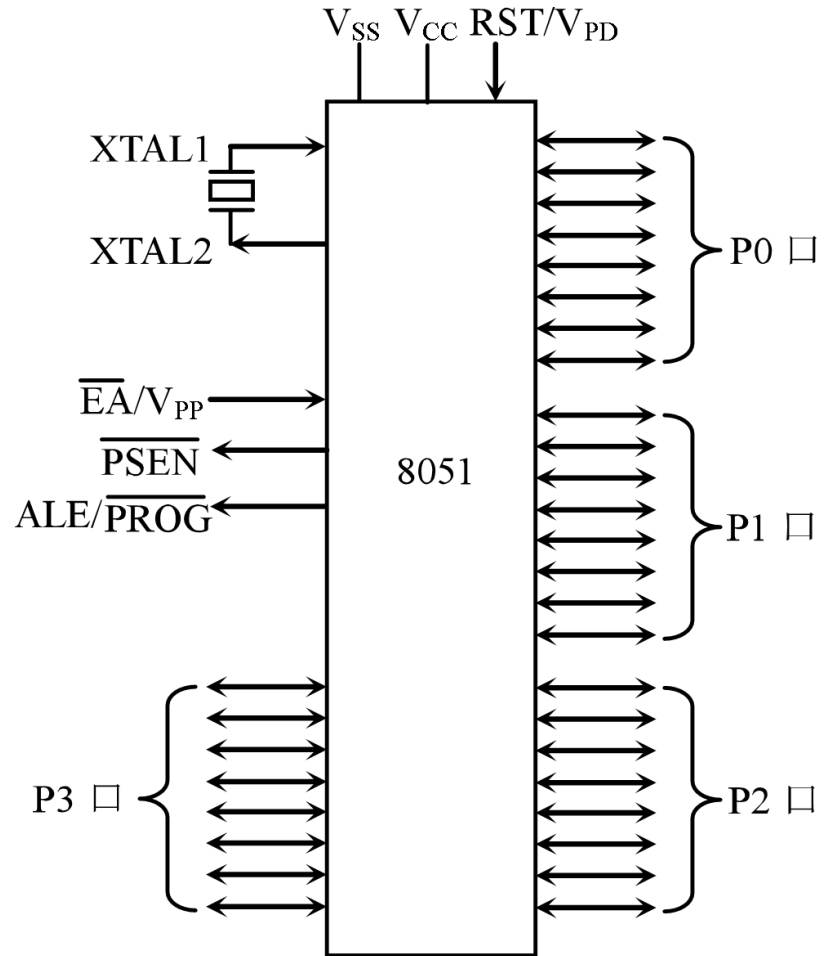


SIMM72  
Single In-line  
Memory  
Module

# 8051单片机引脚图



DIP引脚图



逻辑符号

## 8051引脚的功能描述（一）

- 电源引脚(2根)
  - $V_{CC}$  (40脚)： 电源端， 接+5V电源。
  - $V_{SS}$  (20脚)： 接地端。
- 时钟引脚(2根)
  - XTAL1 (19脚)： 接外部晶振和微调电容的一端。采用外部时钟电路时， 对HMOS型工艺的单片机， 此引脚应接地； 对CHMOS型而言， 此引脚应接外部时钟的输入端。
  - XTAL2 (18脚)： 接外部晶振和微调电容的另一端。使用外部时钟时， 对HMOS型工艺的单片机， 此引脚应接外部时钟的输入端； 对CHMOS型而言， 此引脚悬空。

## 8051引脚的功能描述 （二）

- 控制引脚(4根)

- **RST/V<sub>PD</sub>(9脚)**: 复位信号/备用电源输入引脚。

当**RST**引脚保持两个机器周期的高电平后，就可以使**8051**完成复位操作。该引脚的第二功能是**V<sub>PD</sub>**，即备用电源的输入端，具有掉电保护功能。若在该引脚接**+5V**备用电源，在使用中若主电源**VCC**掉电，可保护片内**RAM**中的信息不丢失。

- **ALE/PROG (30脚)**: 地址锁存允许信号输出/编程脉冲输入引脚。当**CPU**访问片外存储器时，**ALE**输出信号控制锁存**P0**口输出的低**8**位地址，从而实现**P0**口数据与低位地址的分时复用。当**8051**上电正常工作后，自动在**ALE**端输出频率为**fosc/6**的脉冲序列(**fosc**代表振荡器的频率)。

该引脚的第二功能**PROG**是对**8751**内部**4KB EPROM**编程写入时，作为编程脉冲的输入端。

- **EA/V<sub>PP</sub>(31脚)**: 外部程序存储器地址允许输入端/编程电压输入端。

当**EA**接高电平时，**CPU**执行片内**ROM**指令，但当**PC**值超过**0FFFFH**时，将自动转去执行片外**ROM**指令；当**EA**接低电平时，**CPU**只执行片外**ROM**指令。对于**8031**，由于其无片内**ROM**，故其**EA**必须接低电平。

该引脚的第二功能**V<sub>PP</sub>**是对**8751**片内**EPROM**编程写入时，作为**21V**编程电压的输入端。

- **PSEN(29脚)**: 片外**ROM**读选通信号端。

在读片外**ROM**时，**PSEN**有效，为低电平，以实现片外**ROM**的读操作。

## 8051引脚的功能描述 （三）

- I/O引脚( $4 \times 8 = 32$ 根)
- P0.0~P0.7 (39~32脚): P0口的8位双向I/O口线。  
P0口既可作地址/数据总线使用, 又可作通用的I/O口使用。当CPU访问片外存储器时, P0口分时先作低8位地址总线, 后作双向数据总线, 此时, P0口就不能再作I/O口使用了。
- P1.0~P1.7 (1~8脚): P1口的8位准双向I/O口线。  
P1口作为通用的I/O口使用。
- P2.0~P2.7 (21~28脚): P2口的8位准双向I/O口线。  
P2口既可作为通用的I/O口使用, 也可作为片外存储器的高8位地址总线, 与P0口配合, 组成16位片外存储器单元地址。
- P3.0~P3.7 (10~17脚): P3口的8位准双向I/O口线。  
P3口除了作为通用的I/O口使用之外, 每个引脚还具有第二功能。

## 2.1.4 单片机的工作原理

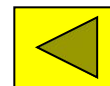
- 单片机执行程序的过程，就是逐条执行指令的过程。单片机每执行一条指令都可分为三个阶段，即取指令——分析指令——执行指令。
- 取指令的任务是根据程序计数器PC中的值从程序存储器读出下一条要执行的指令，送到指令寄存器。分析指令的任务是将指令寄存器中的指令操作码取出后进行译码，分析其指令性质。如指令要求操作数，则寻找操作数地址。接下来，就是执行指令。
- 单片机中的程序一般事先都已通过写入器(编程器)固化在片内或片外程序存储器中，因而一开机即可执行指令。

## 2.2 单片机的存储器

2.2.1 程序存储器ROM

2.2.2 数据存储器RAM

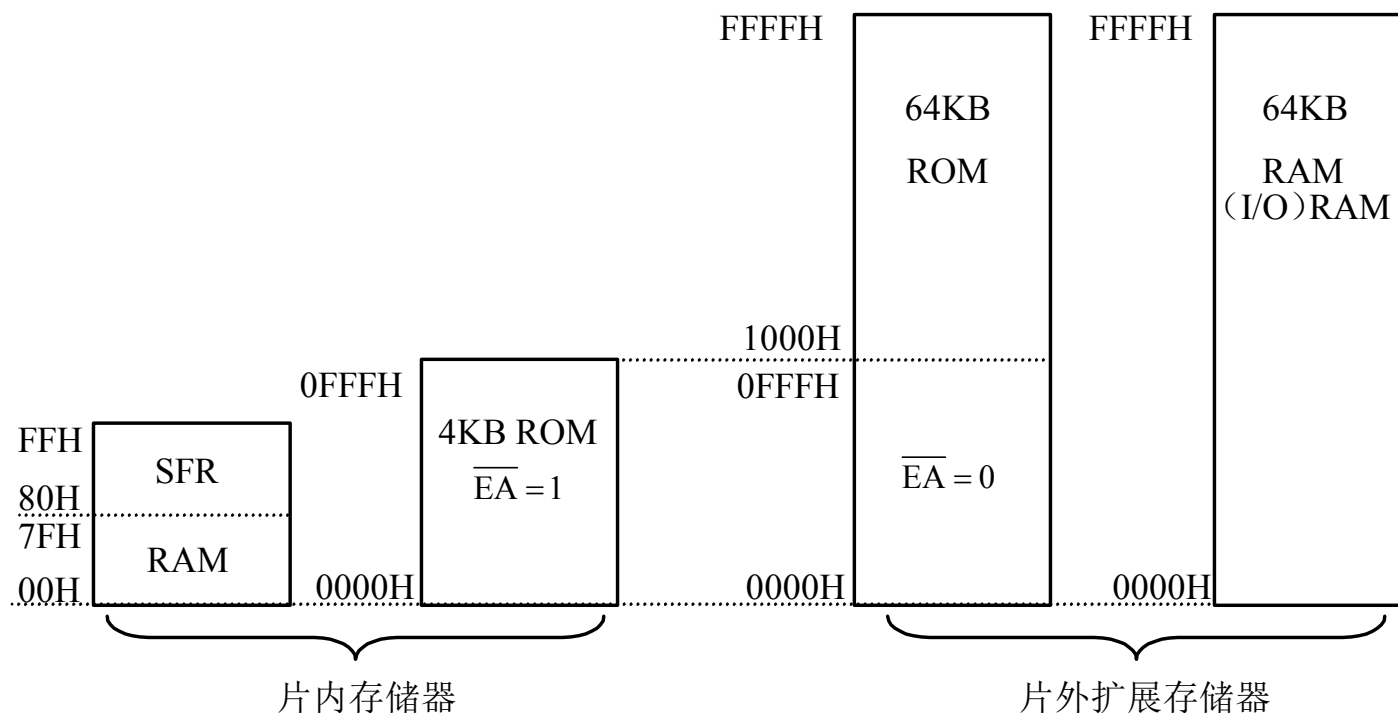
2.2.3 数据存储器数据读写应用实例



## 8051的存储器结构

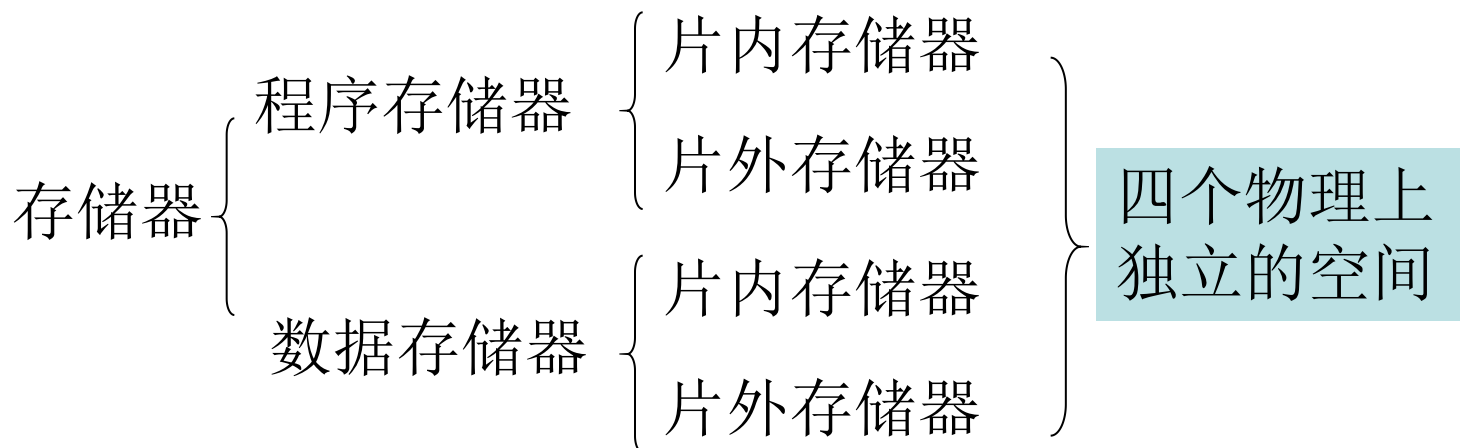
- **8051的存储器结构与常见的微型计算机的配置方式不同，它把程序存储器和数据存储器分开，有各自的寻址系统、控制信号和功能。**
- **8051的存储器在物理结构上分为片内数据存储器、片内程序存储器、片外数据存储器 and 片外程序存储器4个存储空间。但从用户使用的角度看，8051的存储器分为3个逻辑空间。**

## 存储空间分布图



- 片内外统一寻址的**64KB**程序存储器空间，地址范围为**0000H~FFFFH**。
- **64KB**的片外数据存储器空间，地址范围也为**0000H~FFFFH**。
- **256B**的片内数据存储器空间，地址范围为**00H~FFH**。

# MCS-51的存储器配置



程序存储器和数据存储器的寻址空间是分开的，属于哈佛存储结构。

## 2.2.1 程序存储器ROM

- 程序存储器ROM用来存放程序、常数或表格等。
- 在8051中，其存储空间分布如下：
  - 片内有4KB的ROM存储单元，地址为0000H~0FFFH。
  - 片外最多可扩展至64KB的ROM，地址为1000H~FFFFH。
  - 片内外ROM统一编址。
- 当EA引脚接高电平时，CPU将首先访问片内ROM，当指令地址超过0FFFH时，自动转向片外ROM取指令。
- 当EA引脚接低电平时，CPU只访问片外ROM。片外ROM的地址从0000H开始编址。对于8031，由于其片内无ROM，所以使用时必须使EA接低电平，以便能够从片外扩展的EPROM中取指令。

## 2.3.1 程序存储器配置

$\overline{EA}$ : 片内程序存储器屏蔽控制端

当 $\overline{EA}=0$ 时，单片机只执行片外程序存储器的程序，起始点从片外ROM的0000H开始。

当 $\overline{EA}=1$ 时，单片机先执行片内的ROM，当ROM的地址超过0FFFH（51子系列单片机超过7FFH时），接着执行片外的ROM程序，最大可寻址范围是64KB。

具体片外的ROM多少取决于实有的物理扩展程序存储器的大小。

配置方式

## 2.2.1 程序存储器ROM

- 在程序存储器中，以下6个单元具有特殊含义。
- 0000H: 单片机复位后的程序入口地址。
- 0003H: 外部中断0的中断服务程序入口地址。
- 000BH: 定时器0的中断服务程序入口地址。
- 0013H: 外部中断1的中断服务程序入口地址。
- 001BH: 定时器1的中断服务程序入口地址。
- 0023H: 串行口的中断服务程序入口地址。

- 由于这6个中断向量地址的存在，所以在写程序时，这些地址不要占用。一般在0000H地址只写一条跳转指令，从0030H开始写主程序，如：

```
ORG 0000H
```

```
LJMP MAIN
```

```
...
```

```
ORG 0030H
```

```
MAIN:      ...      ; 开始写主程序
```

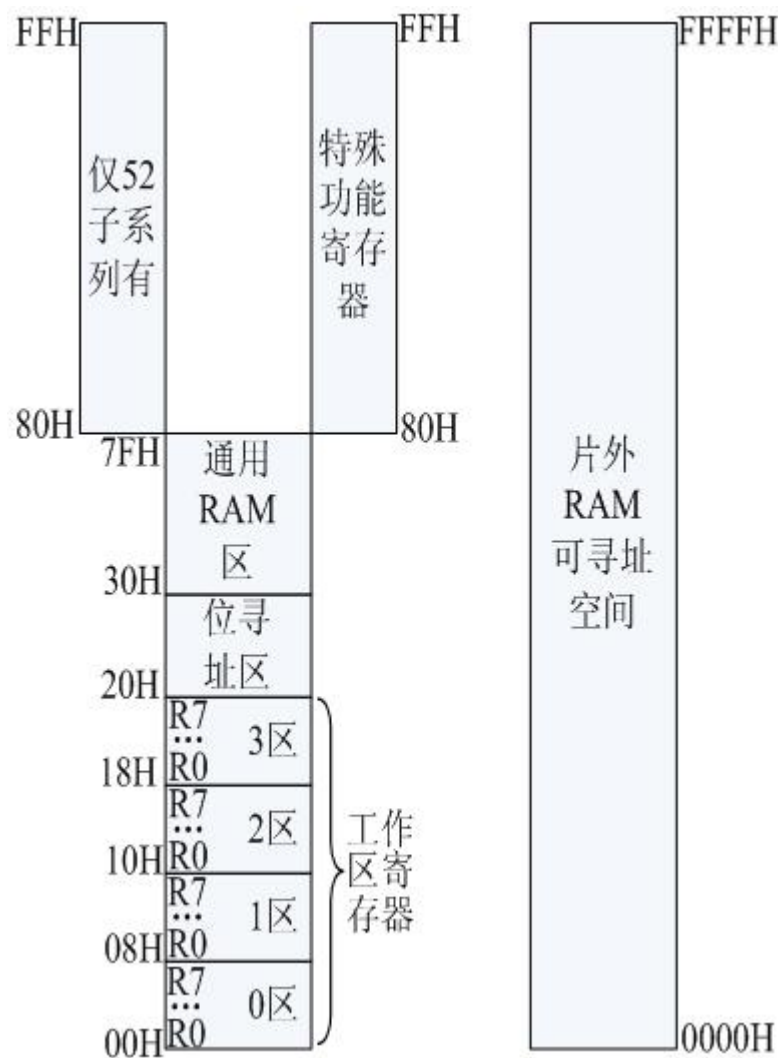
## 2.2.2 数据存储器RAM

- 数据存储器RAM主要用来存放运算的中间结果和数据等。
- 在8051中，其存储空间分布如下：
  - 片外RAM最多可扩展至64KB存储单元，地址范围为0000H~FFFFH。
  - 片内RAM为256B存储单元，地址范围为00H~FFH。
- 片内RAM地址空间共有256B，又分为两个部分：
  - 低128B (00H~7FH) 为真正的RAM区，如下图所示。
  - 高128B (80H~FFH) 为特殊功能寄存器 (SFR) 区，如下表所示。

# 数据存储器配置

数据存储器用于存放运算的中间结果、数据暂存及数据缓冲等。

数据存储器**RAM**也包括：一个片内数据存储器和一个片外数据存储器可寻址空间。



数据存储器配置图

|                     |                  |    |    |    |    |    |    |    |                      |
|---------------------|------------------|----|----|----|----|----|----|----|----------------------|
| 7FH                 |                  |    |    |    |    |    |    |    | 用户RAM区<br>(堆栈、数据缓冲区) |
| 30H                 |                  |    |    |    |    |    |    |    |                      |
| 2FH                 | 7F               | 7E | 7D | 7C | 7B | 7A | 79 | 78 | 位寻址区                 |
| 2EH                 | 77               | 76 | 75 | 74 | 73 | 72 | 71 | 70 |                      |
| 2DH                 | 6F               | 6E | 6D | 6C | 6B | 6A | 69 | 68 |                      |
| 2CH                 | 67               | 66 | 65 | 64 | 63 | 62 | 61 | 60 |                      |
| 2BH                 | 5F               | 5E | 5D | 5C | 5B | 5A | 59 | 58 |                      |
| 2AH                 | 57               | 56 | 55 | 54 | 53 | 52 | 51 | 50 |                      |
| 29H                 | 4F               | 4E | 4D | 4C | 4B | 4A | 49 | 48 |                      |
| 28H                 | 47               | 46 | 45 | 44 | 43 | 42 | 41 | 40 |                      |
| 27H                 | 3F               | 3E | 3D | 3C | 3B | 3A | 39 | 38 |                      |
| 26H                 | 37               | 36 | 35 | 34 | 33 | 32 | 31 | 30 |                      |
| 25H                 | 2F               | 2E | 2D | 2C | 2B | 2A | 29 | 28 |                      |
| 24H                 | 27               | 26 | 25 | 24 | 23 | 22 | 21 | 20 |                      |
| 23H                 | 1F               | 1E | 1D | 1C | 1B | 1A | 19 | 18 |                      |
| 22H                 | 17               | 16 | 15 | 14 | 13 | 12 | 11 | 10 |                      |
| 21H                 | 0F               | 0E | 0D | 0C | 0B | 0A | 09 | 08 |                      |
| 20H                 | 07               | 06 | 05 | 04 | 03 | 02 | 01 | 00 |                      |
| 1FH                 | 第3组工作寄存器 (R0~R7) |    |    |    |    |    |    |    | 工作寄存器区               |
| 18H                 |                  |    |    |    |    |    |    |    |                      |
| 17H                 | 第2组工作寄存器 (R0~R7) |    |    |    |    |    |    |    |                      |
| 10H                 |                  |    |    |    |    |    |    |    |                      |
| 0FH                 | 第1组工作寄存器 (R0~R7) |    |    |    |    |    |    |    |                      |
| 08H                 |                  |    |    |    |    |    |    |    |                      |
| 07H<br>12:57<br>00H | 第0组工作寄存器 (R0~R7) |    |    |    |    |    |    |    |                      |

片内RAM地址空间

片内  
RAM  
地址空间

## 1.片内数据存储器

工作寄存器区 (00H-1FH)

位寻址区 (20H-2FH)

通用区 ( 30H-7FH ) 直接寻址和  
间接寻址

特殊功能寄存器区(80H-FFH) 直接寻址  
通用区(80H-FFH): 间接寻址

仅52子系  
列有

# 位寻址区位地址分配表

| 字节地址 | 位地址            |                |                |                |                |                |                |                |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|      | D <sub>7</sub> | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
| 2FH  | 7FH            | 7EH            | 7DH            | 7CH            | 7BH            | 7AH            | 79H            | 78H            |
| 2EH  | 77H            | 76H            | 75H            | 74H            | 73H            | 72H            | 71H            | 70H            |
| 2DH  | 6FH            | 6EH            | 6DH            | 6CH            | 6BH            | 6AH            | 69H            | 68H            |
| 2CH  | 67H            | 66H            | 65H            | 64H            | 63H            | 62H            | 61H            | 60H            |
| 2BH  | 5FH            | 5EH            | 5DH            | 5CH            | 5BH            | 5AH            | 59H            | 58H            |
| 2AH  | 57H            | 56H            | 55H            | 54H            | 53H            | 52H            | 51H            | 50H            |
| 29H  | 4FH            | 4EH            | 4DH            | 4CH            | 4BH            | 4AH            | 49H            | 48H            |
| 28H  | 47H            | 46H            | 45H            | 44H            | 43H            | 42H            | 41H            | 40H            |
| 27H  | 3FH            | 3EH            | 3DH            | 3CH            | 3BH            | 3AH            | 39H            | 38H            |
| 26H  | 37H            | 36H            | 35H            | 34H            | 33H            | 32H            | 31H            | 30H            |
| 25H  | 2FH            | 2EH            | 2DH            | 2CH            | 2BH            | 2AH            | 29H            | 28H            |
| 24H  | 27H            | 26H            | 25H            | 24H            | 23H            | 22H            | 21H            | 20H            |
| 23H  | 1FH            | 1EH            | 1DH            | 1CH            | 1BH            | 1AH            | 19H            | 18H            |
| 22H  | 17H            | 16H            | 15H            | 14H            | 13H            | 12H            | 11H            | 10H            |
| 21H  | 0FH            | 0EH            | 0DH            | 0CH            | 0BH            | 0AH            | 09H            | 08H            |
| 20H  | 07H            | 06H            | 05H            | 04H            | 03H            | 02H            | 01H            | 00H            |

# 特殊功能寄存器（SFR）

特殊功能寄存器的地址离散的分布在80H~FFH的空间中。

**51子系列单片机有21个特殊功能寄存器。52子系列比51子系列多了一个定时器/计数器T2，增加了5个特殊功能寄存器。**

| 符 号    | 名 称                 | 地 址     |
|--------|---------------------|---------|
| ACC    | 累加器                 | 0E0H    |
| B      | B寄存器                | 0F0H    |
| PSW    | 程序状态字               | 0D0H    |
| SP     | 堆栈指针                | 81H     |
| DPTR   | 数据指针(包括高位DPH和低位DPL) | 83H(高位) |
|        |                     | 82H(低位) |
| P0     | P0口锁存寄存器            | 80H     |
| P1     | P1口锁存寄存器            | 90H     |
| P2     | P2口锁存寄存器            | 0A0H    |
| P3     | P3口锁存寄存器            | 0B0H    |
| IP     | 中断优先级控制寄存器          | 0B8H    |
| IE     | 中断允许控制寄存器           | 0A8H    |
| TMOD   | 定时/计数器工作方式、状态寄存器    | 89H     |
| T2CON* | 定时/计数器2控制寄存器        | 0C8H    |
| TCON   | 定时/计数器控制寄存器         | 88H     |

| 符 号     | 名 称               | 地 址  |
|---------|-------------------|------|
| TH0     | 定时/计数器0(高字节)      | 8CH  |
| TL0     | 定时/计数器0(低字节)      | 8AH  |
| TH1     | 定时/计数器1(高字节)      | 8DH  |
| TL1     | 定时/计数器1(低字节)      | 8BH  |
| TH2*    | 定时/计数器2(高字节)      | 0CDH |
| TL2*    | 定时/计数器2(低字节)      | 0CCH |
| RCAP2H* | 定时/计数器2记录寄存器(高字节) | 0CBH |
| RCAP2L* | 定时/计数器2记录寄存器(低字节) | 0CAH |
| SCON    | 串行口控制寄存器          | 98H  |
| SBUF    | 串行数据缓冲器           | 99H  |
| PCON    | 电源控制寄存器           | 97H  |

这些特殊功能寄存器，可以以字节寻址，部分也可以位寻址。可位寻址的寄存器，是该寄存器的地址是8的整倍数。有11个可位寻址寄存器，如累加器（0E0H）、串行口控制寄存器（98H）等。

在位寻址时，某特殊功能寄存器的字节地址，就是该特殊功能寄存器最低位（D0）的位地址，其它位地址依次递增。

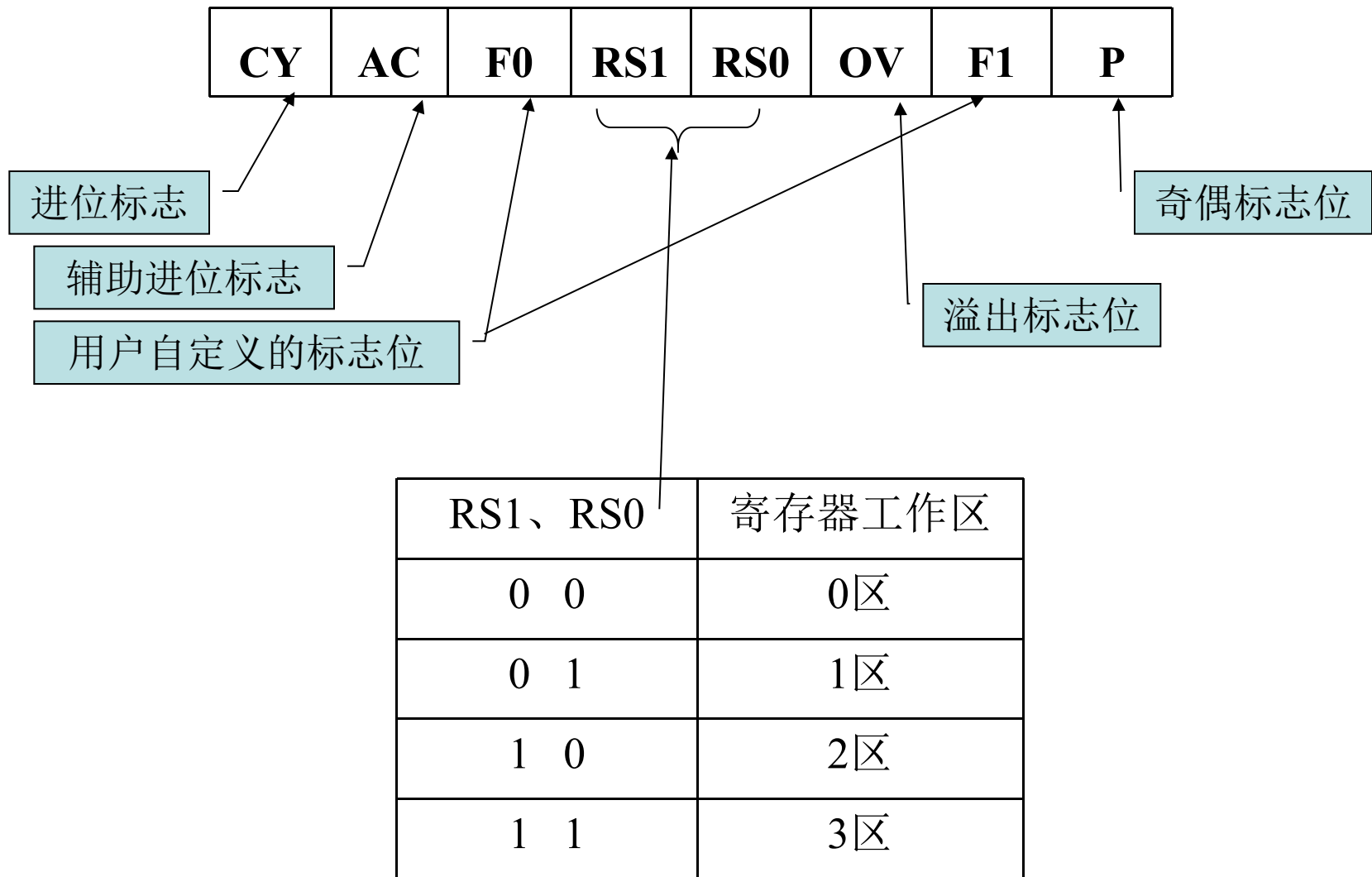
这些特殊功能寄存器的功能如下所述。

1. 累加器**ACC** (**0E0H**) { 助记符是**A**进行位操作时常用符号**ACC**。  
如累加器的**D0**位，表示为“**ACC.0**”。

2. **B**寄存器 { 一般寄存器使用  
在乘除指令中，专门的用途 { 乘法：两个操作数一个是**A**，另一个必是**B**。其结果放在**AB**寄存器对中。  
除法：被除数是**A**，除数是**B**，商数存放于**A**，余数存放于**B**。

3. 程序状态寄存器 (**PSW**) 程序状态寄存器是一个**8**位寄存器它包含了程序状态信息和一些可控制位

|           |           |           |            |            |           |           |          |
|-----------|-----------|-----------|------------|------------|-----------|-----------|----------|
| <b>CY</b> | <b>AC</b> | <b>F0</b> | <b>RS1</b> | <b>RS0</b> | <b>OV</b> | <b>F1</b> | <b>P</b> |
|-----------|-----------|-----------|------------|------------|-----------|-----------|----------|



- **PSW**程序状态寄存器位定义
- **CY (PEW.7)** 进位标志。

在执行某些算术和逻辑指令时，可以被硬件或软件置位或清除。（见3.3节，**MCS-51**指令集）。在布尔处理机中，它被认为是位累加器。它的重要性相当于字节处理中的累加器。

- **AC (PSW.6)** 辅助进位标志位。

在加减运算中，当低4位向高4位有进位或借位时，**AC**由硬件置位，否则**AC**位被清零。在**BCD**码运算时要十进制调整，也要用到**AC**位状态进行判断。

- **F0 (PSW.5)** 用户定义的标志位。

用户可根据需要用软件方法对该位进行置位或复位，以控制程序的流程。

- **F0 (PSW.5)** 用户定义的标志位。用户可根据需要用软件方法对该位进行置位或复位，以控制程序的流程。表2.6 寄存器工作区选择控制表
- **RS1、RS0 (PSW.4、PSW.3)** 选择当前工作区控制位。可用软件对它们置“1”或置“0”，以选择或确定当前工作寄存器区。**RS1、RS0**与寄存器区的关系见2.5表

表2.6 寄存器工作区选择控制表

| RS1、RS0 | 寄存器工作区 |
|---------|--------|
| 0 0     | 0区     |
| 0 1     | 1区     |
| 1 0     | 2区     |
| 1 1     | 3区     |

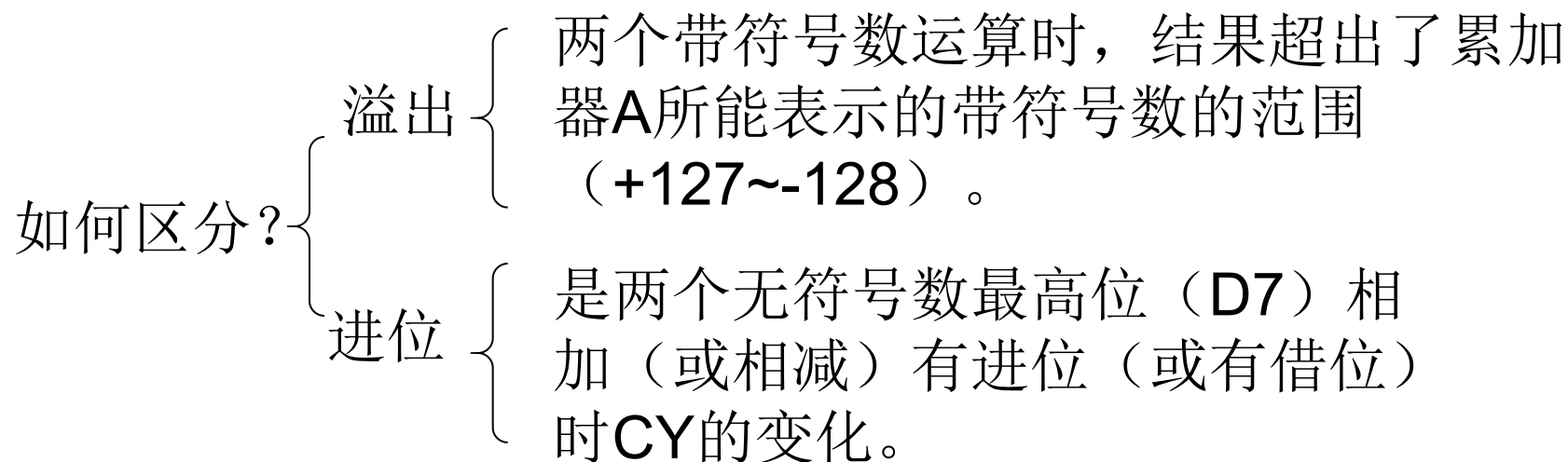
- 0V (PSW. 2) 溢出标志位

当执行算术指令时，反映带符号数的运算结果是否溢出，溢出时由硬件置OV=“1”，否则OV=“0”。溢出和进位是两种不同的概念。对8位运算而言，溢出是指两个带符号数运算时，结果超出了累加器A所能表示的带符号数的范围（-128~+127）。而进位是两个无符号数最高位（D7）相加（或相减）有进位（或有借位）时CY的变化。（参阅3.3 MCS-51指令集）。还有无符号数乘法指令MUL的执行结果也会影响溢出标志位。置于累加器A和寄存器B的两个乘数的积超过255时，OV=“1”，否则OV=“0”。此积的高8位放在B内，低8位放在A内。因此OV=“0”只意味着乘积结果，只从A中取得即可。否则要从BA寄存器对中取得乘积。除法指令DIV也会影响溢出标志位。当除数为0时，OV=“1”，否则OV=“0”。

- **F1 (PSW.1)** 同**F0**
- **P (PSW.0)** 奇偶标志位

执行每条指令都由硬件来置位或清零，以表示累加器**A**中为1位的个数的奇偶性。若累加器**A**中1的个数为奇数，则**P**=“1”，否则**P**=“0”。此标志位对串行通信中的数据传输校验有重要意义。常用**P**作为发送一个符号的奇偶校验位，以增加通信的可靠性。

溢出时由硬件置OV=“1”，否则OV=“0”。



## 4. 堆栈指针SP（81H）

8位的特殊功能寄存器

特点：指数据只允许在其一端进出的一段存储空间。

SP { 数据写入：入栈或压栈  
数据读出：出栈或弹栈 } 原则：先入后出，后进先出

类型 { 指针指向栈顶    MCS-51  
指针指向栈底

一般设置SP=30H或以上的空间

堆栈正落在工作寄存器1区，避免在切换工作寄存器区时发生冲突。

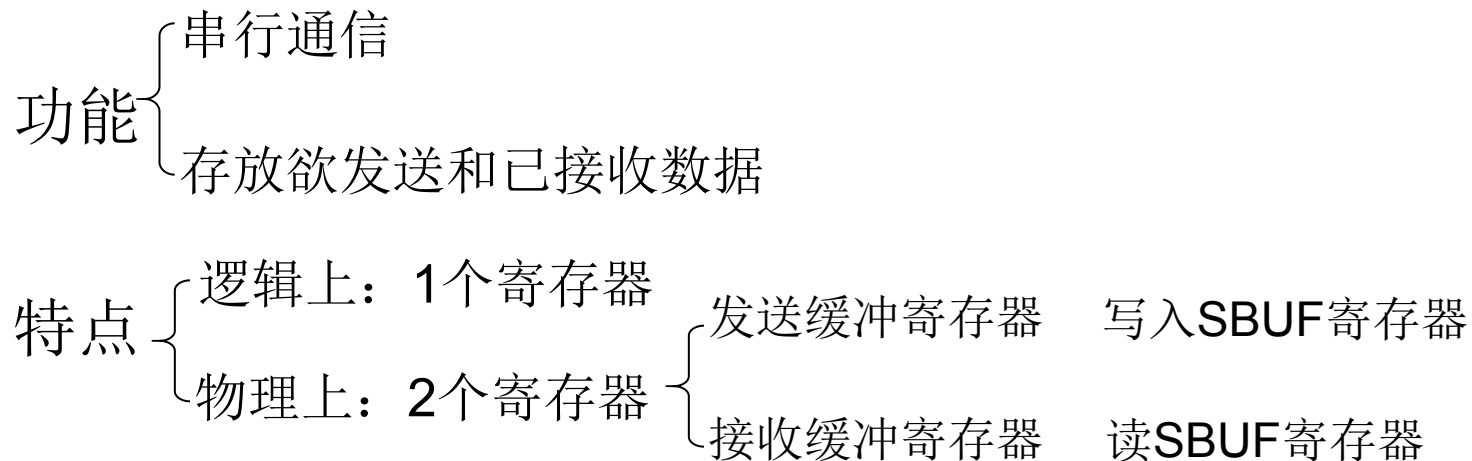
## 5. 数据指针寄存器DPTR (83H、82H)

数据指针DPTR { 16位专用寄存器  
高位字节: DPH, 低位字节: DPL  
2个独立的8位寄存器DPH和DPL使用

6. P0~P3端口寄存器 { I/O端口P0~P3的8位锁存器  
(80H,90H,0A0H,0B0H) 均为可位寻址寄存器

7. 定时器/计数器T0,T1和T2 { 51子系列: T0,T1 一个定时器/计数器是由2个独立的8位寄存器组成的16位寄存器  
52子系列: T0,T1,T2

## 8. 串行数据缓冲器SBUF（99H）



- 其它IP、TMOD、T2CON、SCON和PCON等控制寄存器，将在后面有关章节中介绍。

9. 程序计数器PC { 16位程序地址寄存器  
存放下一条要执行指令的地址  
寻址范围：0000H~0FFFFH

SFR 中位地址分布表

| SFR  | MSB  | 位地址/位定义 |      |      |      |      |      | LSB  | 字节地址 |
|------|------|---------|------|------|------|------|------|------|------|
| B    | F7   | F6      | F5   | F4   | F3   | F2   | F1   | F0   | F0H  |
|      |      |         |      |      |      |      |      |      |      |
| ACC  | E7   | E6      | E5   | E4   | E3   | E2   | E1   | E0   | E0H  |
|      |      |         |      |      |      |      |      |      |      |
| PSW  | D7   | D6      | D5   | D4   | D3   | D2   | D1   | D0   | D0H  |
|      | CY   | AC      | F0   | RS1  | RS0  | OV   | –    | P    |      |
| IP   | BF   | BE      | BD   | BC   | BB   | BA   | B9   | B8   | B8H  |
|      | –    | –       | –    | PS   | PT1  | PX1  | PT0  | PX0  |      |
| P3   | B7   | B6      | B5   | B4   | B3   | B2   | B1   | B0   | B0H  |
|      | P3.7 | P3.6    | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |      |
| IE   | AF   | AE      | AD   | AC   | AB   | AA   | A9   | A8   | A8H  |
|      | EA   | –       | –    | ES   | ET1  | EX1  | ET0  | EX0  |      |
| P2   | A7   | A6      | A5   | A4   | A3   | A2   | A1   | A0   | A0H  |
|      | P2.7 | P2.6    | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |      |
| SCON | 9F   | 9E      | 9D   | 9C   | 9B   | 9A   | 99   | 98   | 98H  |
|      | SM0  | SM1     | SM2  | REN  | TB8  | RB8  | TI   | RI   |      |
| P1   | 97   | 96      | 95   | 94   | 93   | 92   | 91   | 90   | 90H  |
|      | P1.7 | P1.6    | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |      |
| TCON | 8F   | 8E      | 8D   | 8C   | 8B   | 8A   | 89   | 88   | 88H  |
|      | TF1  | TR1     | TF0  | TR0  | IE1  | IT1  | IE0  | IT0  |      |
| P0   | 87   | 86      | 85   | 84   | 83   | 82   | 81   | 80   | 80H  |
|      | P0.7 | P0.6    | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |      |

## 2.片外数据存储器

片外数据存储器可寻址空间是指**MCS-51**单片机对片外扩展数据存储器的最大寻址能力。**51**单片机是**64KB**。

片外扩展的数据存储器与片内数据存储器不是统一编址的，逻辑上、物理都是独立的两个空间。

在数据传送上，访问片内数存器用**MOV**指令，访问片外数据存存器用**MOVB**指令。

## 2.2.3 数据存储器数据读写应用实例

### 【程序1】 工作寄存器的读写

| 地址           | 机器码             | 程 序                  | 注 释                 | 检查结果     |
|--------------|-----------------|----------------------|---------------------|----------|
|              |                 | <b>ORG 0000H</b>     | ； 程序从ROM区0000H处开始存放 |          |
| <b>0000H</b> | <b>78 11</b>    | <b>MOV R0, #11H</b>  | ； 将立即数11H送入寄存器R0中   | ( H)=11H |
| <b>0002H</b> | <b>79 22</b>    | <b>MOV R1, #22H</b>  | ； 将立即数22H送入寄存器R1中   | ( H)=22H |
| <b>0004H</b> | <b>7A 33</b>    | <b>MOV R2, #33H</b>  | ； 将立即数33H送入寄存器R2中   | ( H)=33H |
| <b>0006H</b> | <b>7B 44</b>    | <b>MOV R3, #44H</b>  | ； 将立即数44H送入寄存器R3中   | ( H)=44H |
| <b>0008H</b> | <b>75 D0 10</b> | <b>MOV PSW, #10H</b> | ； 使当前工作寄存器组为第2组     | ( H)=10H |
| <b>000BH</b> | <b>78 55</b>    | <b>MOV R0, #55H</b>  | ； 将立即数55H送入寄存器R0中   | ( H)=55H |
| <b>000DH</b> | <b>79 66</b>    | <b>MOV R1, #66H</b>  | ； 将立即数66H送入寄存器R1中   | ( H)=66H |
| <b>000FH</b> | <b>7A 77</b>    | <b>MOV R2, #77H</b>  | ； 将立即数77H送入寄存器R2中   | ( H)=77H |
| <b>0011H</b> | <b>7B 88</b>    | <b>MOV R3, #88H</b>  | ； 将立即数88H送入寄存器R3中   | ( H)=88H |
|              |                 | <b>END</b>           | ； 程序结束              |          |

## 【程序2】 位寻址区的读写

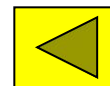
| 地址           | 机器码          | 程 序              | 注 释                 | 检查结果              |
|--------------|--------------|------------------|---------------------|-------------------|
|              |              | <b>ORG 0100H</b> | ； 程序从ROM区0100H处开始存放 |                   |
| <b>0100H</b> | <b>D2 00</b> | <b>SETB 00H</b>  | ； 将位地址为00H的位置1      | ( H)=1            |
| <b>0102H</b> | <b>D2 01</b> | <b>SETB 01H</b>  | ； 将位地址为01H的位置1      | ( H)=1            |
| <b>0104H</b> | <b>D2 02</b> | <b>SETB 02H</b>  | ； 将位地址为02H的位置1      | ( H)=1            |
| <b>0106H</b> | <b>D2 03</b> | <b>SETB 03H</b>  | ； 将位地址为03H的位置1      | ( H)=1            |
| <b>0108H</b> | <b>C2 04</b> | <b>CLR 04H</b>   | ； 将位地址为04H的位清0      | ( H)=0            |
| <b>010AH</b> | <b>C2 05</b> | <b>CLR 05H</b>   | ； 将位地址为05H的位清0      | ( H)=0            |
| <b>010CH</b> | <b>C2 06</b> | <b>CLR 06H</b>   | ； 将位地址为06H的位清0      | ( H)=0            |
| <b>010EH</b> | <b>C2 07</b> | <b>CLR 07H</b>   | ； 将位地址为07H的位清0      | ( H)=0<br>(20H )= |
| <b>0110H</b> | <b>C2 90</b> | <b>CLR 90H</b>   | ； 将P1.0位清0          | ( H)=0            |
| <b>0112H</b> | <b>D2 90</b> | <b>SETB 90H</b>  | ； 将P1.0位置1          | ( H)=1            |
| <b>0114H</b> | <b>C2 91</b> | <b>CLR 91H</b>   | ； 将P1.1位清0          | ( H)=0            |
| <b>0116H</b> | <b>D2 91</b> | <b>SETB 91H</b>  | ； 将P1.1位置1          | ( H)=1            |
|              |              | <b>END</b>       | ； 程序结束              |                   |

## 【程序3】一般RAM区和特殊功能寄存器区的读和写

| 地址           | 机器码             | 程 序                     | 注 释                                      | 检查结果                                 |
|--------------|-----------------|-------------------------|------------------------------------------|--------------------------------------|
|              |                 | <b>ORG 0200H</b>        | ； 程序从ROM区0200H存放                         |                                      |
| <b>0200H</b> | <b>75 30 99</b> | <b>MOV 30H, #99H</b>    | ； 将立即数99H送到30H单元中                        | <b>(30H)=<br/>(PC)=</b>              |
| <b>0203H</b> | <b>75 45 AA</b> | <b>MOV 45H, #0AAH</b>   | ； 将立即数AAH送到45H单元中                        | <b>(45H)=<br/>(PC)=</b>              |
| <b>0206H</b> | <b>75 81 50</b> | <b>MOV SP, #50H</b>     | ； 将立即数50H送到堆栈指针SP中                       | <b>(<br/>H)=50H<br/>(PC)=</b>        |
| <b>0209H</b> | <b>74 60</b>    | <b>MOV A, #60H</b>      | ； 将立即数60H送到累加器A中                         | <b>(<br/>H)=60H<br/>(PC)=</b>        |
| <b>020BH</b> | <b>75 90 55</b> | <b>MOV P1, #55H</b>     | ； 将立即数55H送到P1口中                          | <b>(<br/>H)=55H<br/>(PC)=</b>        |
| <b>020EH</b> | <b>75 D0 90</b> | <b>MOV PSW, #90H</b>    | ； 将立即数90H送到PSW中，<br>使当前工作寄存器组为第2组且将CY位置1 | <b>(<br/>H)=90H<br/>(PC)=</b>        |
| <b>0211H</b> | <b>90 12 34</b> | <b>MOV DPTR, #1234H</b> | ； 将立即数1234H送到数据指针DPTR中                   | <b>(<br/>H)=12H<br/>(<br/>H)=34H</b> |

## 2.3 单片机的输入/输出端口

- 2.3.1 输入/输出端口结构
- 2.3.2 输入/输出端口负载能力
- 2.3.3 P1口输出功能应用实例
- 2.3.4 P3口输入功能应用实例



## 2.3.1 输入/输出端口结构

- 8051单片机有4个8位并行I/O端口，称为P0、P1、P2和P3口，每个端口都各有8条I/O口线，每条I/O口线都能独立地用作输入或输出。
- 在无片外扩展存储器的系统中，这四个I/O口都可以作为通用I/O口使用。
- 在有片外扩展存储器的系统中，P2口送出高8位地址，P0口分时送出低8位地址和8位数据。

# I/O端口电路与电气特性

4个I/O端口 {  
P0是三态双向口  
P1~P3口是准双向口  
P2及P3口还有不同的第二功能电路

4个I/O端口的共性 {  
1. 每个端口的8位线有相同电路结构  
2. 每位I/O线均有一个输出锁存器、一个驱动器和一个输入缓冲器。

# I/O端口内部电路结构（引脚）

1.P0口 { 三态双向、双功能的8位并行口  
字节地址为（80H），可位寻址，位地址为（80H~ 87H）

(1) 位电路的组成，如图所示。

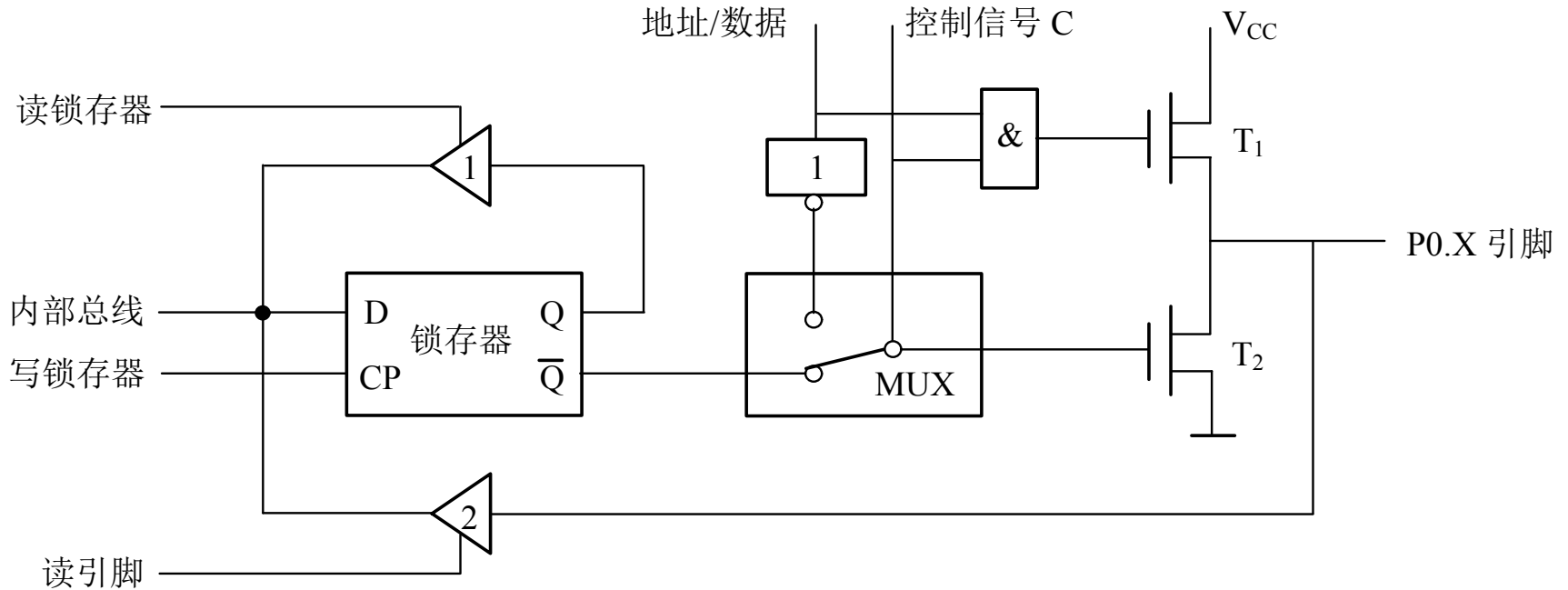
(2) P0口作为通用I/O接口

(3) 地址/数据分时复用功能

P0口作为地址/数据复用线时，才是真正的三态双向接口。

## 1. P0口

- P0口某一位的结构图如下图所示，它由一个输出锁存器、两个三态输入缓冲器、一个转换开关MUX、一个输出驱动电路(T1和T2)和一个与门及一个非门组成。



## (1) P0口用作通用I/O口

**MUX与锁存器的Q端接通，与门输出为0，T1截止，输出驱动级就工作在需外接上拉电阻的漏极开路方式。**

### ① P0口用作输出口

**CPU在执行输出指令时，内部数据总线的数据在“写锁存器”信号的作用下，由D端进入锁存器，取反后出现在Q端，再经过T2反向，则P0.X引脚上的数据就是内部总线的数据。由于T2为漏极开路输出，故此时必须外接上拉电阻。**

### ② P0口用作输入口

**数据可以读自端口的锁存器，也可以读自端口的引脚，这要看输入操作执行的是“读锁存器”指令还是“读引脚”指令。**

- **方式1：读引脚。CPU在执行“MOV”类输入指令时(如：MOV A, P0)，内部产生的操作信号是“读引脚”。P0.X引脚上的数据经过缓冲器2读入到内部总线。注意，在读引脚时，必须先向电路中的锁存器写入1，使T2截止，P0.X引脚处于悬浮状态，可作为高阻抗输入。**
- **方式2：读锁存器。CPU在执行“读-改-写”类输入指令时(如：ANL P0, A)，内部产生的操作信号是“读锁存器”，锁存器中的数据经过缓冲器1送到内部总线，然后与A的内容进行逻辑“与”，结果送回P0的端口锁存器并出现在引脚。除了MOV类指令外，其他的读口操作指令都属于这种情况。**

## (2) P0口用作地址/数据总线

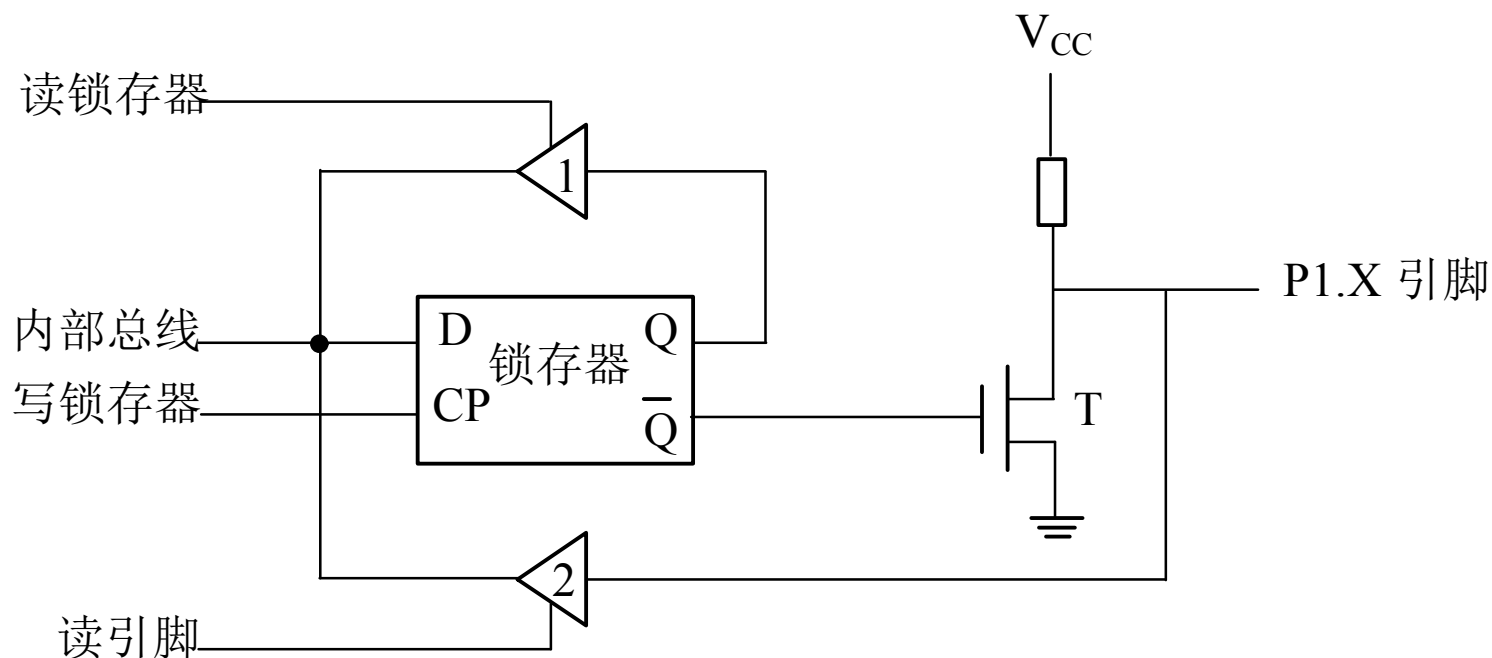
- **MUX将地址/数据线与T2接通，同时与门输出有效。**
- **若地址/数据线为1，则T1导通，T2截止，P0口输出为1；反之T1截止，T2导通，P0口输出为0。**
- **当数据从P0口输入时，读引脚使三态缓冲器2打开，端口上的数据经缓冲器2送到内部总线。**

### (3) P0口小结

- ① P0口既可作地址/数据总线使用，也可作通用I/O口使用。  
当P0口作地址/数据总线使用时，就不能再作通用I/O口使用了。
- ② P0口作输出口使用时，输出级属漏极开路，必须外接上拉电阻，才有高电平输出。
- ③ P0口作输入口读引脚时，应先向锁存器写1，使T2截止，不影响输入电平。

## 2. P1口

- **P1口**是唯一的单功能口，仅能作为通用**I/O**口使用。由于在其输出端接有上拉电阻，故可以直接输出而无需外接上拉电阻。
- 同**P0**口一样，当作输入口时，必须先向锁存器写“1”，使场效应管**T**截止。



P1.X作I/O口的输出线 { 当CPU对P1.X锁存器写入高电平“1”时，Q=“0”，场效应管T1截止，P1.X引脚输出高电平。

当CPU对P1.X锁存器写入低电平“0”时，Q=“1”，场效应管T1导通，P1.X引脚输出低电平。注意输出高电平不要带较重的负载。

P1.X作I/O口的输入线 { 软件首先对P1.X锁存器写高电平“1”，使场效应管T1截止，P1.X引脚呈高电平“1”。

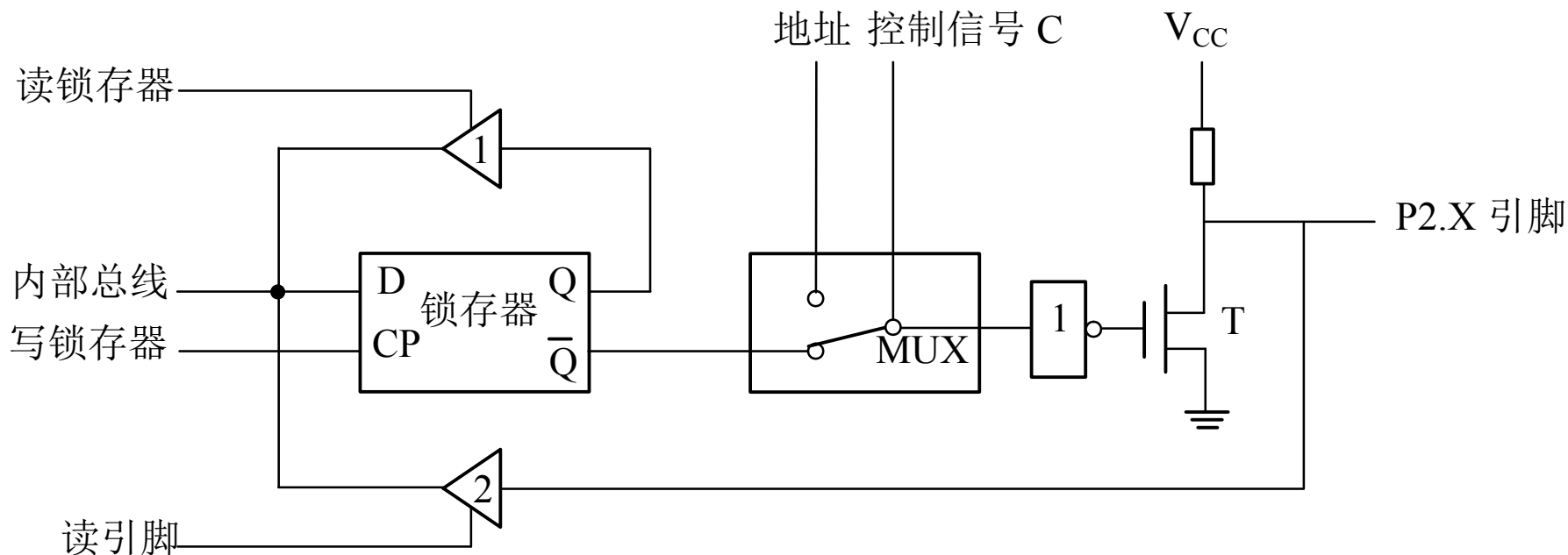
很微弱的电流就可把P1.X引脚拉为低电平，所以P1.X引脚的电平是随外电路驱动的电平变化而变化

软件读P1.X引脚时，CPU使“读引脚”=“1”，三态缓冲器2导通，将P1.X引脚的电平读入内部数据总线。

由于P1.X的输出驱动器是单管方式，所以P1口是准双向口。

### 3. P2口

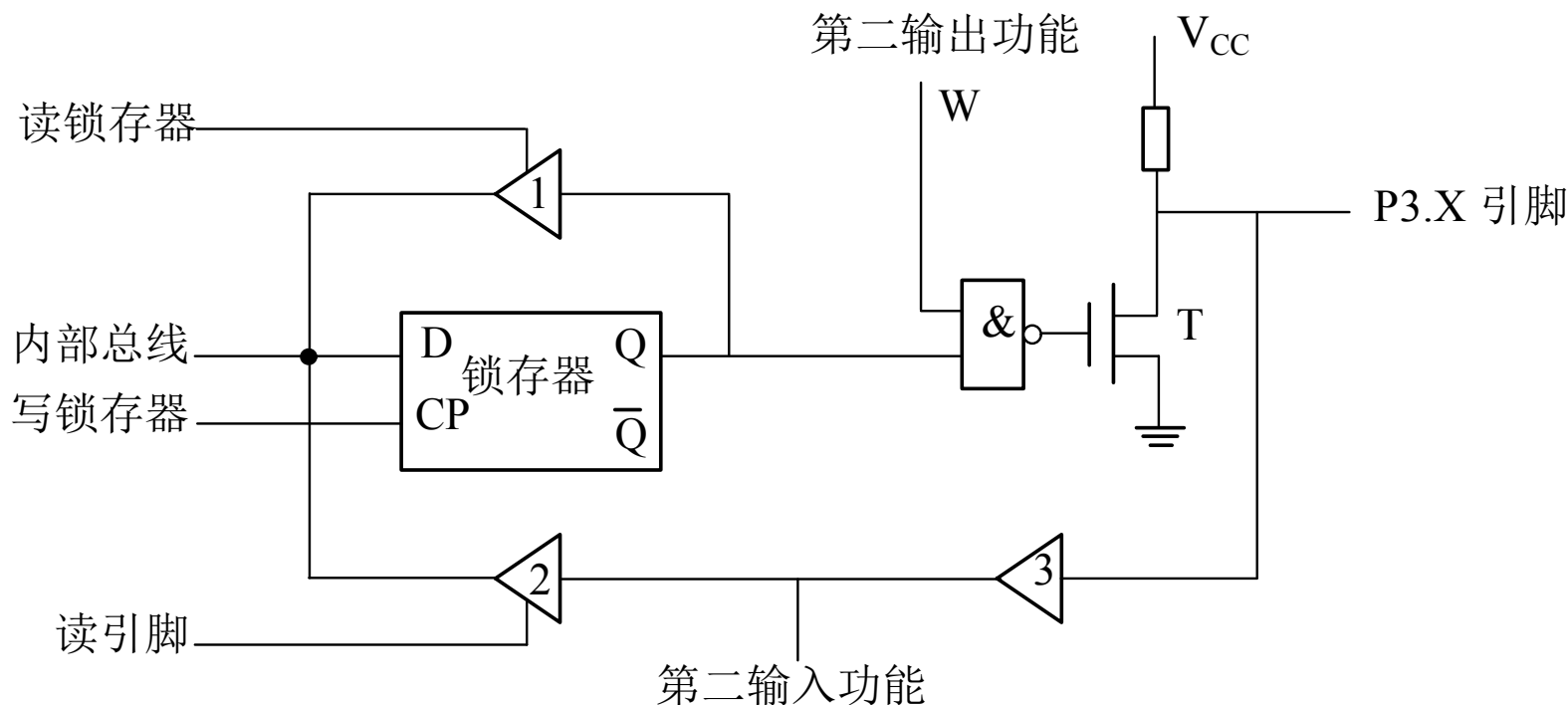
- 图中的控制信号C决定转换开关MUX的位置：当C=0时，MUX拨向下方，P0口为通用I/O口；当控制信号C=1时，MUX拨向上方，P0口作为地址总线使用。
- 在实际应用中，P2口通常作为高8位地址总线使用。



P2口的功能 { (1) P2.X作为通用I/O接口。  
(2) P2.X作为高8位地址输出线。

## 4. P3口

- **P3口用作通用I/O口时，第二输出功能信号 $W=1$ ，P3口的每一位都可定义为输入或输出，其工作原理同P1口类似。**
- **在真正的应用电路中，P3口的第二功能显得更为重要。**



P3口的功能 { (1) P3.X作为通用I/O接口。  
(2) P3.X作为第二功能

## P3口的第二功能

| 引脚          | 第二功能                   | 功能说明           |
|-------------|------------------------|----------------|
| <b>P3.0</b> | <b>RXD</b>             | 串行口输入          |
| <b>P3.1</b> | <b>TXD</b>             | 串行口输出          |
| <b>P3.2</b> | <b>INT0</b>            | 外部中断0输入        |
| <b>P3.3</b> | <b>INT1</b>            | 外部中断1输入        |
| <b>P3.4</b> | $\overline{\text{T0}}$ | 定时器/计数器0计数输入   |
| <b>P3.5</b> | $\overline{\text{T1}}$ | 定时器/计数器1计数输入   |
| <b>P3.6</b> | <b>WR</b>              | 片外RAM写选通信号(输出) |
| <b>P3.7</b> | <b>RD</b>              | 片外RAM读选通信号(输出) |

## 2.3.2 输入/输出端口负载能力

- P0口的输出级与P1~P3口的输出级在结构上不同，其输出级无上拉电阻，因此它们的负载能力和接口要求也不相同。
- P0口的每一位能驱动8个LSTTL负载。在作为通用I/O口使用时，输出驱动电路是开漏的，所以，驱动集电极开路(OC门)电路或漏级开路电路需外接上拉电阻。当作为地址/数据总线使用时(T1可以提供上拉电平)，口线不是开漏的，无需外接上拉电阻。
- P1~P3口的每一位能驱动4个LSTTL负载。它们的输出驱动电路有上拉电阻，所以可以方便地由集电极开路(OC门)电路或漏级开路电路所驱动，而无需外接上拉电阻。
- 对于80C51单片机(CHMOS)，端口只能提供几毫安的输出电流，故当作输出口去驱动一个普通晶体管的基极时，应在端口与晶体管基极间串联一个电阻，以限制高电平输出时的电流。

## 2.3.3 P1口输出功能应用实例

【例1】 P1口做输出口，控制八只发光二极管循环点亮(P1口输出低电平时发光二极管被点亮)。

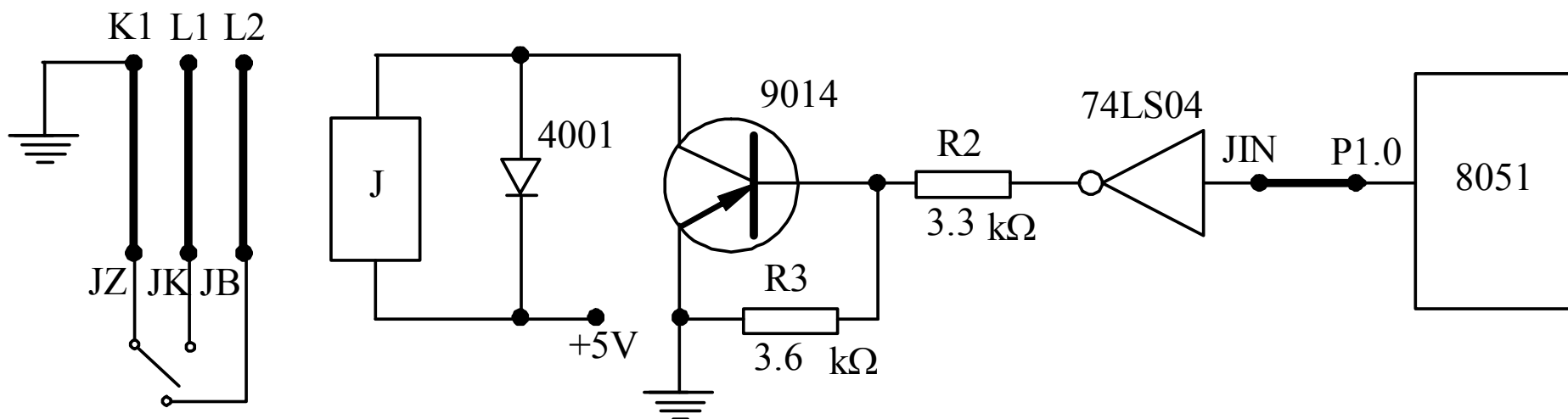
解： 由于发光二极管低电平点亮，所以，需要哪个发光二极管点亮，只需在相应的端口输出逻辑0即可。由于每个发光二极管点亮后要持续一段时间才熄灭，再使下个发光二极管点亮，因此需要编写延时子程序，供主程序反复调用。

本例中，延时子程序采用指令循环来实现。

| 地址    | 机器码      | 程序                   | 注释     |
|-------|----------|----------------------|--------|
|       |          | ORG 0000H            |        |
| 0000H | 02 10 00 | LJMP MAIN            |        |
|       |          | ORG 1000H            |        |
| 1000H | 74 FE    | MAIN: MOV A, #0FEH   |        |
| 1002H | F5 90    | LOOP: MOV P1, A      |        |
| 1004H | 12 10 0A | LCALL DELAY          | ;延时    |
| 1007H | 23       | RL A                 | ;左移位   |
| 1008H | 80 F8    | SJMP LOOP            | ;循环    |
| 100AH | 7F FF    | DELAY: MOV R7, #0FFH | ;延时子程序 |
| 100CH | 7E FA    | L1: MOV R6, #0FAH    |        |
| 100EH | DE FE    | DJNZ R6, \$          |        |
| 1010H | DF FA    | DJNZ R7, L1          |        |
| 1012H | 22       | RET                  |        |
| 12:57 |          | END                  |        |

**【例2】** 利用P1.0输出高低电平，控制继电器的开合，以实现对外部装置(如灯L1和L2)的控制。

解：将单片机的P1.0接继电器控制端JIN，继电器的JZ通过K1接地，常开触点JK接L1，常闭触点JB接L2。编制程序，使P1.0电平变化，高电平时继电器吸合，常开触点闭合，L1点亮，L2熄灭；低电平时继电器不工作，常闭触点闭合，L2点亮，L1熄灭。



## 参考程序：

| 地址    | 机器码      | 程序                  | 注释        |
|-------|----------|---------------------|-----------|
|       |          | ORG 0000H           |           |
| 0000H | 02 20 00 | LJMP MAIN           |           |
|       |          | ORG 2000H           |           |
| 2000H | C2 90    | MAIN: CLR P1.0      | ;P1.0送低电平 |
| 2002H | 12 20 0C | LCALL DELAY         | ;延时       |
| 2005H | D2 90    | SETB P1.0           | ;P1.0送高电平 |
| 2007H | 12 20 0C | LCALL DELAY         | ;延时       |
| 200AH | 80 F4    | SJMP MAIN           | ;循环       |
| 200CH | 7F 06    | DELAY: MOV R7, #06H | ;延时子程序    |
| 200EH | 7E FF    | L1: MOV R6, #0FFH   |           |
| 2010H | 7D FA    | L2: MOV R5, #0FAH   |           |
| 2012H | DD FE    | DJNZ R5, \$         |           |
| 2014H | DE FA    | DJNZ R6, L2         |           |
| 2016H | DF F6    | DJNZ R7, L1         |           |
| 2018H | 22       | RET                 |           |
|       |          | END                 |           |

## 2.3.4 P3口输入功能应用实例

**【例3】** P3口的P3.0连接一个开关，作为输入端；P1口的P1.0~P1.7连接八只发光二极管，作为输出端。要求用P3.0来控制P1输出的循环灯，即当P3.0输出高电平时，控制P1口的发光二极管左循环点亮；当P3.0输出低电平时，控制P1口的发光二极管右循环点亮(P1口输出低电平时发光二极管被点亮)。

解：在主程序中要对P3.0的状态进行判断。

如果P3.0为高电平，则使用循环左移指令。

如果P3.0为低电平，则使用循环右移指令。

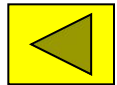
延时子程序同例1。

## 参考程序：

| 地址    | 机器码      | 程序                   | 注释     |
|-------|----------|----------------------|--------|
|       |          | ORG 0000H            |        |
| 0000H | 02 30 00 | LJMP MAIN            |        |
|       |          | ORG 3000H            |        |
| 3000H | 74 FE    | MAIN: MOV A, #0FEH   |        |
| 3002H | F5 90    | LOOP: MOV P1, A      |        |
| 3004H | 12 30 10 | LCALL DELAY          | ;延时    |
| 3007H | 20 B0 03 | JB P3.0, L1          |        |
| 300AH | 03       | RR A                 | ;右移位   |
| 300BH | 80 F5    | SJMP LOOP            | ;循环    |
| 300DH | 23       | L1: RL A             | ;左移位   |
| 300EH | 80 F2    | SJMP LOOP            | ;循环    |
| 3010H | 7F FF    | DELAY: MOV R7, #0FFH | ;延时子程序 |
| 3012H | 7E FA    | L2: MOV R6, #0FAH    |        |
| 3014H | DE FE    | DJNZ R6, \$          |        |
| 3016H | DF FA    | DJNZ R7, L2          |        |
| 3018H | 22       | RET                  |        |
| 12:57 |          | END                  |        |

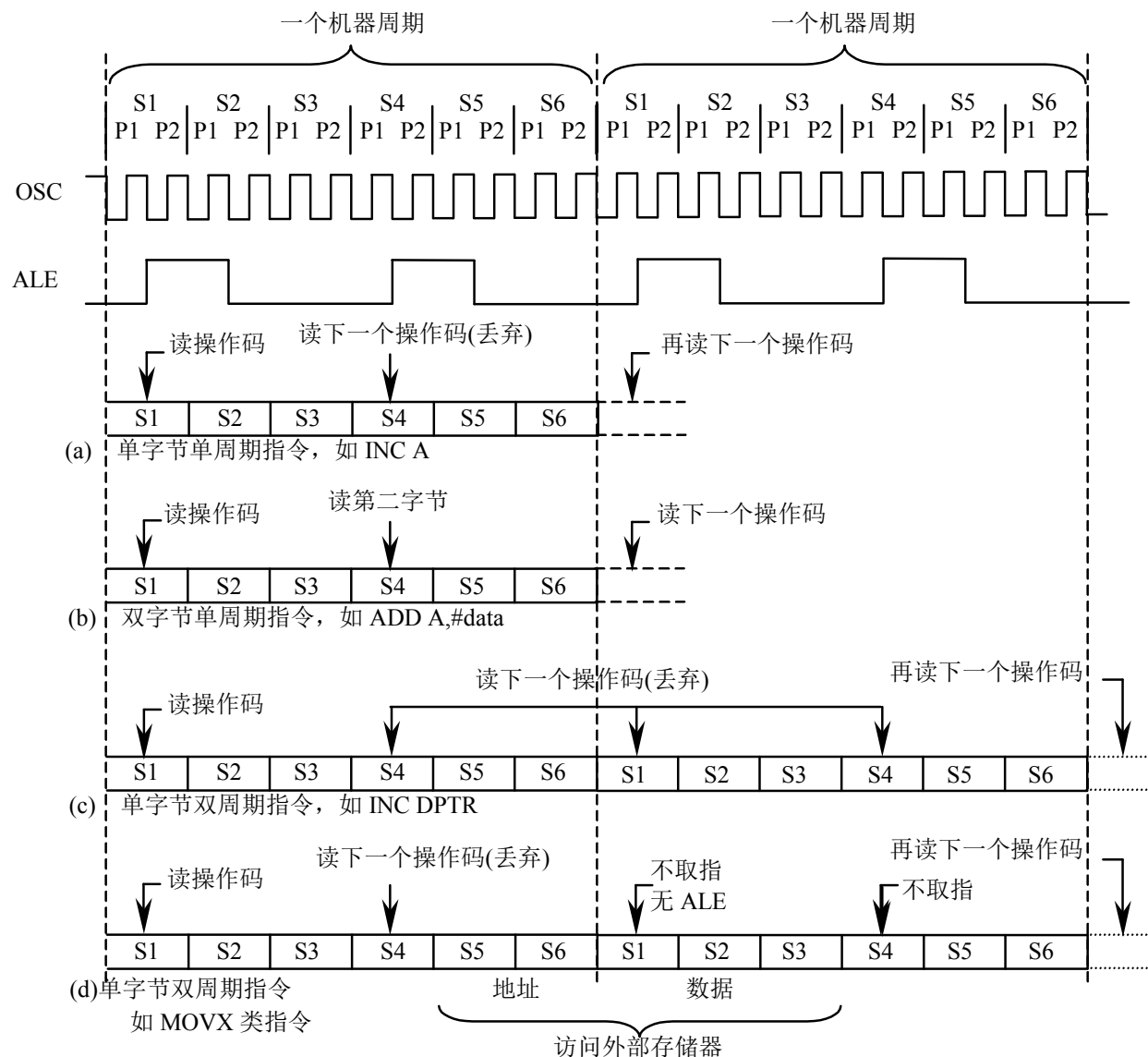
## 2.4 时钟电路

- 2.4.1 CPU时序
- 2.4.2 时钟电路



## 2.4.1 CPU时序

- **CPU时序就是CPU在执行指令时所需控制信号的时间顺序。**
- 单片机的时序定时单位从小到大依次为：时钟周期、状态周期、机器周期和指令周期。
- **MCS-51单片机共有111条指令，按照指令字节数和机器周期数可分为六类，即单字节单周期指令、单字节双周期指令、单字节四周期指令、双字节单周期指令、双字节双周期指令和三字节双周期指令。**



典型指令的取指、执行时序

时钟周期为单片机的振荡源的周期, 也称为振荡周期。

状态周期(或状态 S)是振荡周期的两倍, 它分为 P1 节拍和 P2 节拍。

一条指令的执行过程分作几个基本操作, 完成一个基本操作所需的时间称作机器周期。

执行一条指令所需的时间称为指令周期。

# 工作周期划分

根据各类指令的需要和一些特殊工作状态的需要，可将单片机的周期划分为**6**个状态，用**6**个周期状态触发器作为它们的标志。某一时期只有其中一个周期状态触发器为**1**，指名**CPU**现在所处的工作周期状态，为该阶段的工作提供时间标志与依据。

- 1、取指令周期**FT**
- 2、源周期**ST**
- 3、目的周期**DT**
- 4、执行周期**ET**
- 5、中断周期**IT**
- 6、DMA周期**DMAT**

# 1 取指周期FT

取指所需的操作安排在**FT**中完成，是每条指令都必须经历的。在**FT**中完成的操作是与指令操作码无关的公共性操作，但取指周期结束后将转向那种工作周期，则与**FT**中取出的指令类型有关。

# 2 源周期ST

如需从主存中读取源操作数，则进入**ST**。在**ST**中将依据指令寄存器**IR**的源地址段信息进行操作，形成源地址，读取操作数。

## 3 目的周期DT

如需从主存中读取目的地址或目的操作数，则进入DT。在DT中将依据指令寄存器IR的目的地址段信息进行操作。

## 4 执行周期ET

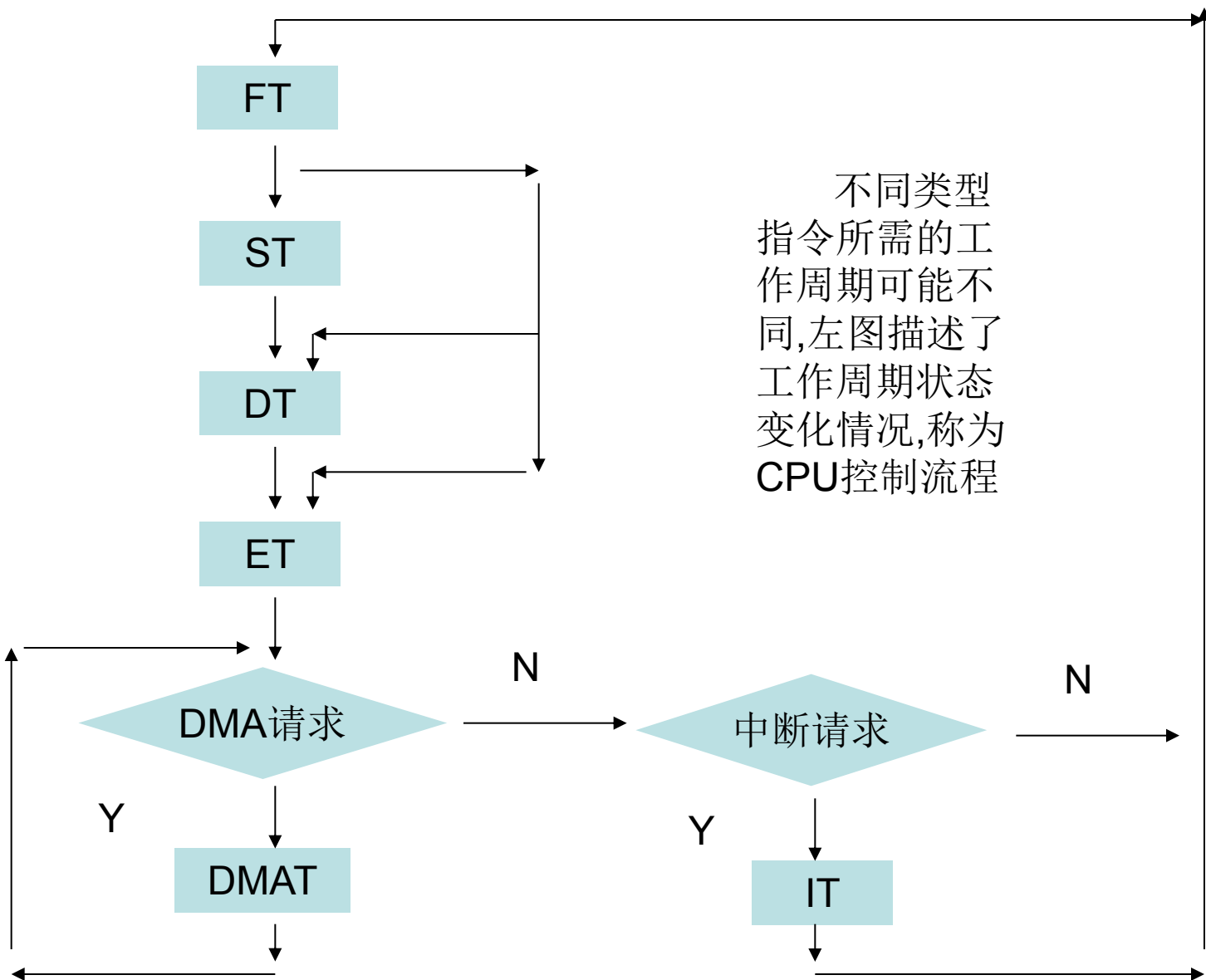
已取得操作数后，则进入ET，这也是各类指令都需要经历的最后个工作阶段。在ET中将依据IR中操作码执行相应操作，如传送、算术运算、逻辑运算、获得转移地址等。在ET中还要将后续指令地址送入地址寄存器中，为下一条指令周期读取新指令做好准备。

## 5 中断周期IT

取指所需的操作安排在**FT**中完成，是每条指令都必须经历的。在**FT**中完成的操作是与指令操作码无关的公共性操作，但取指周期结束后将转向哪种工作周期，则与**FT**中取出的指令类型有关。

## 6 DMA周期DMAT

如需从主存中读取源操作数，则进入**ST**。在**ST**中将依据指令寄存器**IR**的源地址段信息进行操作，形成源地址，读取操作数。



例1 双操作数指令,数均在存储器中,则经历的工作周期变化为:

FT→ST→~~DT~~→ET(FT...)

MOV @R<sub>i</sub>, @R<sub>j</sub>

例2 单操作数指令,数在主存中,则经历的工作周期变化为:

FT → DT →ET(FT...)

INC @R<sub>i</sub>

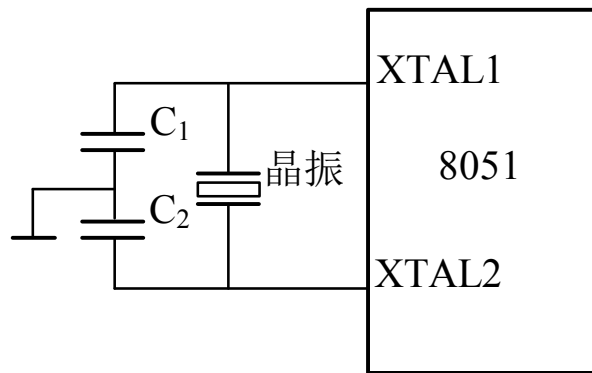
例3 双操作数指令,数均在CPU寄存器中,则经历的工作周期变化为:

FT→ET(FT...)

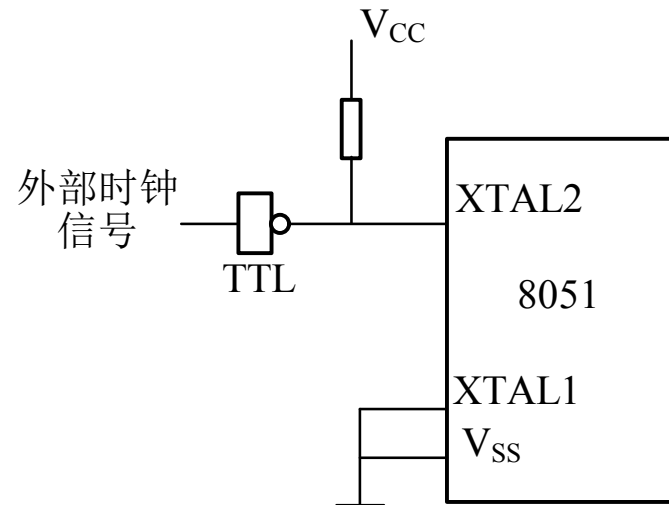
MOV A,R<sub>3</sub>

## 2.4.2 时钟电路

- 8051单片机的时钟信号通常由两种方式产生：一是内部振荡方式，二是外部时钟方式。



(a) 内部方式时钟电路



(b) 外部方式时钟电路

## 2.4.2 时钟电路

### 1. 内部振荡方式

在8051单片机内部有一个高增益的反相放大器，用于构成振荡器，反相放大器的输入端为XTAL1，输出端为XTAL2。

- 内部振荡方式是在XTAL1和XTAL2引脚两端跨接石英晶体振荡器和两个电容构成稳定的自激振荡电路。
- 电容C1和C2通常取30pF，对振荡频率有微调作用。晶振频率范围是1.2MHz~12MHz。

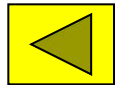
### 2. 外部时钟方式

外部时钟方式是把外部已有的时钟信号引入到单片机内。

- 8051：外部时钟由XTAL2输入，直接送入内部时钟电路，XTAL1接地；
- 80C51：外部时钟由XTAL1输入，XTAL2悬空。
- 外部时钟信号为高电平持续时间要大于20ns，且频率低于12MHz的方波。

## 2.5 复位电路

- 2.5.1 复位功能
- 2.5.2 复位电路



## 2.5.1 复位功能

- 复位是单片机的初始化操作。
- 复位功能是把PC初始化为0000H，使CPU从0000H单元开始执行程序；复位操作同时还对其他一些寄存器有影响，但内部RAM的数据是不变的。
- 除了进入系统的正常初始化之外，当由于程序运行出错或操作错误使系统处于死锁状态时，为摆脱困境，也需要按复位键重新启动。

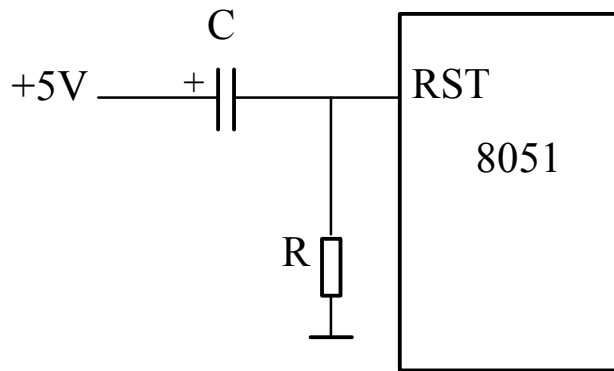
## 部分特殊功能寄存器的复位状态

| 寄存器             | 复位状态             | 寄存器             | 复位状态             |
|-----------------|------------------|-----------------|------------------|
| <b>PC</b>       | <b>0000H</b>     | <b>ACC</b>      | <b>00H</b>       |
| <b>B</b>        | <b>00H</b>       | <b>PSW</b>      | <b>00H</b>       |
| <b>SP</b>       | <b>07H</b>       | <b>DPTR</b>     | <b>0000H</b>     |
| <b>P0～P3</b>    | <b>0FFH</b>      | <b>IP</b>       | <b>×××00000B</b> |
| <b>IE</b>       | <b>0××00000B</b> | <b>TMOD</b>     | <b>00H</b>       |
| <b>TCON</b>     | <b>00H</b>       | <b>TL0, TL1</b> | <b>00H</b>       |
| <b>TH0, TH1</b> | <b>00H</b>       | <b>SCON</b>     | <b>00H</b>       |
| <b>SBUF</b>     | <b>不定</b>        | <b>PCON</b>     | <b>0×××0000B</b> |

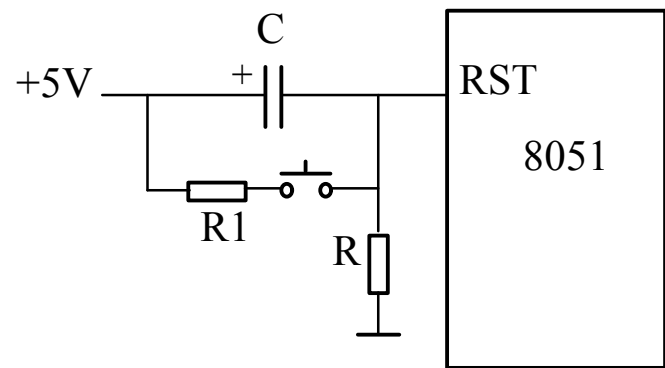
说明：表中符号×状为随机态。

## 2.5.2 复位电路

- 单片机的复位引脚RST出现2个机器周期以上的高电平时，单片机就执行复位操作。
- 常见的复位操作有上电自动复位和按键手动复位两种方式。
- 上电自动复位是通过外部复位电路的电容充电来实现的。
- 按键手动复位是通过复位端经电阻与电源 $V_{CC}$ 接通而实现的，它兼备上电复位功能。



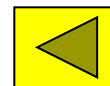
(a) 上电自动复位



(b) 按键手动复位

## 2.6 常用单片机功能简介

- 2.6.1 Atmel公司的AT89系列单片机
- 2.6.2 Motorola公司M68HC08系列单片机
- 2.6.3 其他系列单片机



## 2.6.1 Atmel公司的AT89系列单片机

- **AT89系列单片机不但具有一般MCS-51单片机的所有特性，而且其Flash程序存储器可以用电擦除方式瞬间擦除、改写，写入单片机内的程序还可以进行加密。**
- **Atmel公司的89系列单片机主要有AT89C51、AT89C2051和AT89C1051等型号。AT89C2051和AT89C1051算是AT89C51的精简版。**
- **AT89C51最为实用，它不但和8051指令、管脚完全兼容，而且其片内含有4KB Flash型程序存储器，性价比远高于8751。**
- **AT89C2051去掉了P0口和P2口，内部的Flash程序存储器也只有2K，封装形式也由40脚改为20脚。**
- **AT89C1051在AT89C2051的基础上，再次精简掉了串口等功能，程序存储器减小到1KB。**

## 2.6.2 Motorola公司M68HC08系列单片机

- **M68HC08系列单片机**，具有速度快、功能强、价格低、功耗低、指令系统丰富等特点，内部程序存储器采用成熟的Flash存储器技术。
- 编程速度快、可靠性高 。
- 单一电源电压供电——片内集成电荷泵，可由单一工作电压在片内产生编程电压。
- 支持在线编程 ——允许单片机内部运行的程序去改写Flash存储器内容，这样可代替外部电可擦除存储器芯片，减少外围部件，增加嵌入式系统开发的方便性。
- 高频噪声低，抗干扰能力强 。

## 2.6.3 其他系列单片机

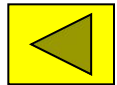
- **Atmel**公司的**AVR**单片机，是增强型**RISC**(精简指令集)、内载**Flash**的单片机。由于采用增强的**RISC**结构，使其具有高速处理能力，在一个时钟周期内可执行复杂的指令。**AVR**单片机工作电压为2.7~6.0V，可以实现耗电最优化。
- **MicroChip**公司的主要产品是**PIC16C**系列和**PIC17C**系列8位单片机，**CPU**采用**RISC**结构，分别仅有33、35、58条指令，采用**Harvard**双总线结构，运行速度快，工作电压低，低功耗，有较大的输入/输出直接驱动能力，价格低，能一次性编程，体积小，适用于用量大、档次低、价格敏感的产品。

# 本章小结

- Intel公司MCS-51系列单片机是我国目前应用最为广泛的单片机。8051/80C51是整个MCS-51系列单片机的核心，该系列其他型号的单片机都是在这—内核的基础上发展起来的。
- 8051单片机内部结构包括中央处理器、程序存储器、数据存储器、并行I/O接口、定时器/计数器、时钟电路、中断系统、串行口。中央处理器是单片机的核心部件，是计算机的控制指挥中心。
- 8051的程序存储器和数据存储器是各自独立的，各有各的寻址系统、控制信号和功能。在物理结构上可分为片内数据存储器、片内程序存储器、片外数据存储器 and 片外程序存储器4个存储空间。
- 片内RAM共256B，分为两大功能区，低128B为真正的RAM区；高128B为特殊功能寄存器(SFR)区。低128B RAM又分为工作寄存器区、位寻址区和用户RAM区。

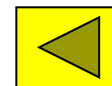
# 本章小结

- 8051单片机有P0、P1、P2和P3 4个8位并行I/O端口，每个端口各有8条I/O口线，每条I/O口线都能独立地用作输入或输出。各端口的功能不同，且结构上也有差异，通常P2口作为高8位地址线，P0口分时复用作为低8位地址线和8位数据线，P3口使用第二功能，P1口只能作为通用I/O口使用。P0口的输出级与P1~P3口的输出级在结构上不同，其输出级无上拉电阻，因此它们的负载能力和接口要求也不相同。
- 时序就是CPU在执行指令时所需控制信号的时间顺序，其单位有振荡周期、时钟周期、机器周期和指令周期。时钟信号产生方式有内部振荡方式和外部时钟方式两种。
- 复位是单片机的初始化操作，复位操作对PC和部分特殊功能寄存器有影响，但对内部RAM没有影响。



# 习 题

1. 请结合MCS-51系列单片机的结构框图，阐明其组成。
2. 综述MCS-51系列单片机各引脚的作用，并试分类。
3. 程序计数器的符号是什么？MCS-51系列单片机的程序计数器有几位？
4. 何谓程序状态字？它的符号是什么？它各位的含义是什么？
5. 何谓振荡周期、时钟周期、机器周期和指令周期？针对MCS-51系列单片机，如采用12MHz晶振，它们的周期各是什么值？
6. 引脚的作用是什么？在下列三种情况下，引脚各应接何种电平？
  - (1) 只有片内ROM；
  - (2) 有片内ROM和片外ROM；
  - (3) 有片内ROM和片外ROM，片外ROM所存为调试程序。
7. MCS-51系列单片机存储器的地址空间是如何划分的？各地址空间的地址范围和容量如何？
8. MCS-51怎样实现上电自动复位与按键手动复位？并请综述各专用寄存器复位后的状态。



**Q & A?**

**3QU!**

