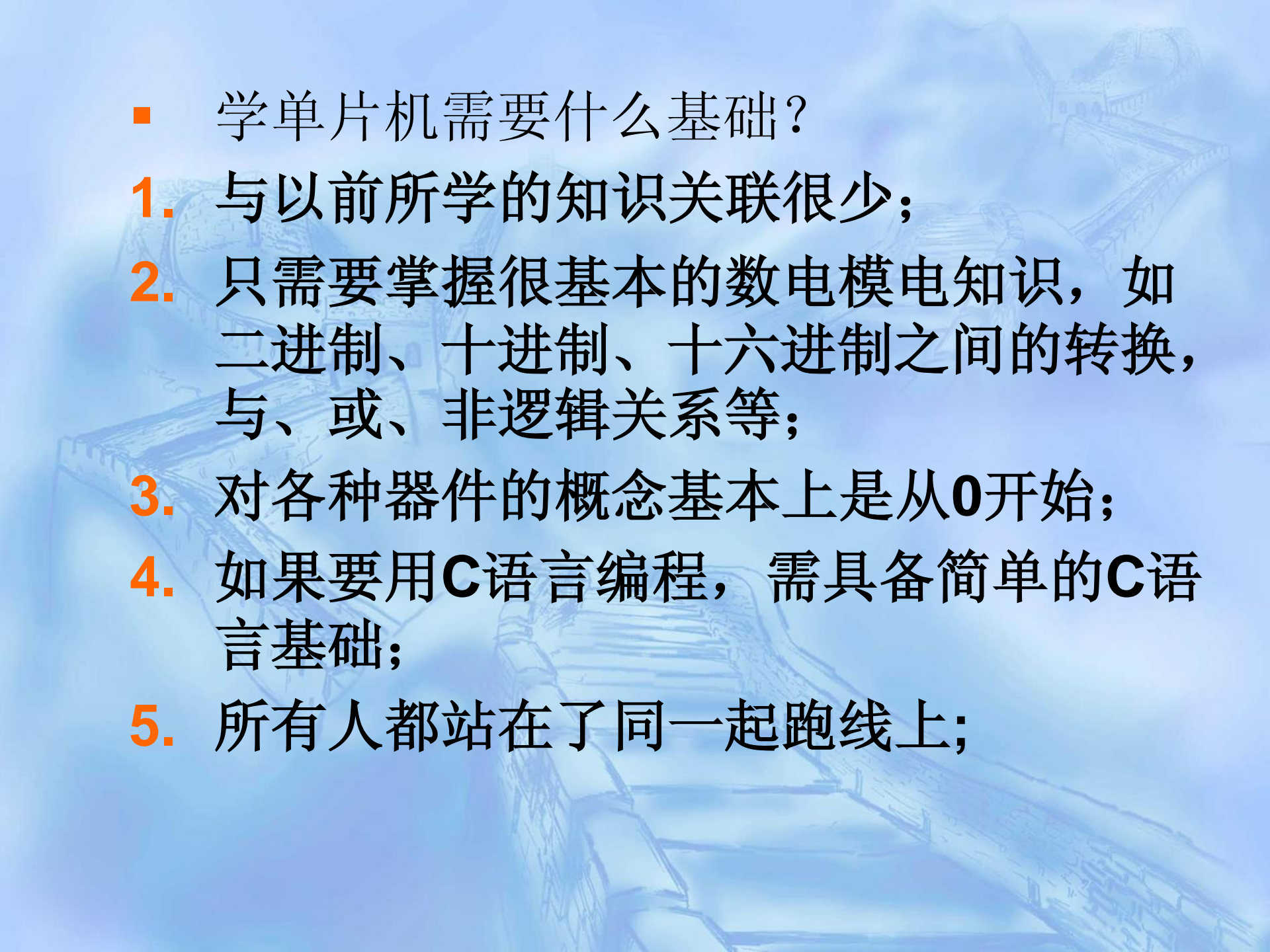


第一讲

- 学单片机需要什么基础？
- 什么是单片机、单片机能做什么、怎么开始学习单片机。
- 学单片机需要的一些预备知识
- **C51**基础知识。
- 了解**TX-1C**单片机学习板系统结构
- 如何用**TX-1C**单片机学习板在短时间内掌握**51**单片机和**C**语言编程。
- **KEIL**工程建立详细介绍及**KEIL**软件使用方法
- 从点亮实验板上的一个发光二极管开始

- 
- 学单片机需要什么基础？
 1. 与以前所学的知识关联很少；
 2. 只需要掌握很基本的数电模电知识，如二进制、十进制、十六进制之间的转换，与、或、非逻辑关系等；
 3. 对各种器件的概念基本上是从0开始；
 4. 如果要用C语言编程，需具备简单的C语言基础；
 5. 所有人都站在了同一起跑线上；

什么是单片机？

◆ 单板机

将**CPU**芯片、**存储器**芯片、**I/O接口**芯片和简单的**I/O设备**（小键盘、**LED显示器**）等装配在一块印刷电路板上，再配上**监控程序**（固化在**ROM**中），就构成了一台单板微型计算机（简称单板机）。

单板机

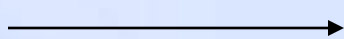


单板机的I/O设备简单，软件资源少，使用不方便。早期主要用于微型计算机原理的**教学及简单的测控系统**，现在已很少使用。

◆单片机

在一片集成电路芯片上集成微处理器、存储器、I/O接口电路，从而构成了单芯片微型计算机，即单片机。Intel公司推出了MCS-51系列单片机：集成8位CPU、4K字节ROM、128字节RAM、4个8位并口、1个全双工串行口、2个16位定时/计数器。寻址范围64K，并有控制功能较强的布尔处理器。

单板机



单片机



单片机能做什么

- 实物展示
- 用到单片机的项目经验介绍
 1. 手持粮库温度寻检设备
 2. 毕设答辩打分器
 3. 电话台灯
 4. 自动感应水龙头
- 凡是与控制或简单计算有关的电子设备都可以用单片机来实现，再根据具体实际情况选择不同性能的单片机，如：atmel,stc,pic,avr,凌阳，80C51，arm等

- 工业自动化：数据采集、测控技术。
- 智能仪器仪表：数字示波器、数字信号源、数字万用表、感应电流表等。
- 消费类电子产品：洗衣机、电冰箱、空调机、电视机、微波炉、手机、IC卡、汽车电子设备等。
- 通讯方面：调制解调器、程控交换技术、手机、小灵通等。
- 武器装备：飞机、军舰、坦克、导弹、航天飞机、鱼雷制导、智能武器等。
- 等等.....

怎么开始学习？

- 实践第一。
- 补充必要的理论知识，即缺什么补什么。
- 做工程项目积累经验。（可在网络上搜集题目，也可自己有什么想法大胆的去试验）

预备知识

- 电平特性
- 2进制与16进制的表示及转换
- 二进制数的逻辑运算
- 8051单片机介绍
- C51基础知识

关于电平特性

- 数字电路中只有两种电平：高和低
- （本课程中）定义单片机为TTL电平：

高 +5V 低 0V

- RS232电平：计算机的串口

高 -12V 低+12V

所以计算机与单片机之间通讯时需要加电平转换芯片max232（实验板上左下角）。



二进制

- 数字电路中的两种电平特性决定了它

0	0	6	110
1	1	7	111
2	10	8	1000
3	11	9	1001
4	100	10	1010
5	101	11	1011

十六进制

- 是二进制的简短表示形式。

十进制中的0-15分别表示为十六进制的0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F

- 熟练掌握二进制与十六进制之间的转换。

规律：一般把四个二进制数放在一起转换成一个十六进制数，转换时先把二进制数转换成十进制数，再把十进制数转换成十六进制数。

如：0001B -> 1 -> 1H

B表示为二进制，H为十六进制

1001B -> 9 -> 9H

1010B -> 10 -> AH

0010 1100 -> 44 -> 2CH

各种进位制的对应关系

十进制	二进制	十六进制	十进制	二进制	十六进制
0	0	0	9	1001	9
1	1	1	10	1010	A
2	10	2	11	1011	B
3	11	3	12	1100	C
4	100	4	13	1101	D
5	101	5	14	1110	E
6	110	6	15	1111	F
7	111	7	16	10000	10
8	1000	8			

二进制数的逻辑运算

1. “与”运算

“与”运算是实现“必须都有，否则就没有”这种逻辑关系的一种运算。

运算符为“ \cdot ”，其运算规则如下：

$$0 \cdot 0 = 0, \quad 0 \cdot 1 = 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1$$

2. “或” 运算

“或”运算是实现“只要其中之一有，就有”这种逻辑关系的一种运算，其运算符为“+”。“或”运算规则如下：

$$0+0=0, \quad 0+1=1+0=1, \quad 1+1=1$$

3. “非” 运算

“非” 运算是实现 “求反” 这种逻辑的一种运算，如变量A的 “非” 运算记作 \bar{A} 。其运算规则如下：

$$\bar{1} = 0, \bar{0} = 1$$

4. “异或”运算

“异或”运算是实现“必须不同，否则就没有”这种逻辑的一种运算，运算符为“ \oplus ”。其运算规则是：

$$0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$$



80C51系列介绍

80C51是**MCS-51**系列中的一个典型品种；其它厂商**以8051为基核**开发出的**CMOS**工艺单片机产品**统称为80C51系列**。当前常用的**80C51**系列单片机主要产品有：

- * **Intel**的：80C31、80C51、87C51，80C32、80C52、87C52等；
- * **ATMEL**的：89C51、89C52、89C2051等；
- * **Philips、华邦、Dallas、STC Siemens(Infineon)**等公司的许多产品。

补充：关于型号的介绍

80C51的引脚封装

总线型

P1.0	1	40	VCC
P1.1	2	39	P0.0
P1.2	3	38	P0.1
P1.3	4	37	P0.2
P1.4	5	36	P0.3
P1.5	6	35	P0.4
P1.6	7	34	P0.5
P1.7	8	33	P0.6
RST/V _{PD}	9	32	P0.7
P3.0/RXD	10	31	EA/V _{PP}
P3.1/TXD	11	30	ALE/ <u>PROG</u>
P3.2/ <u>INT0</u>	12	29	PSEN
P3.3/ <u>INT1</u>	13	28	P2.7
P3.4/T0	14	27	P2.6
P3.5/T1	15	26	P2.5
P3.6/ <u>WR</u>	16	25	P2.4
P3.7/ <u>RD</u>	17	24	P2.3
XTAL2	18	23	P2.2
XTAL1	19	22	P2.1
V _{SS}	20	21	P2.0

非总线型

RST	1	20	VCC
P3.0/RXD	2	19	P1.7
P3.1/TXD	3	18	P1.6
XTAL2	4	17	P1.5
XTAL1	5	16	P1.4
P3.2/ <u>INT0</u>	6	15	P1.3
P3.3/ <u>INT1</u>	7	14	P1.2
P3.4/T0	8	13	P1.1/AIN1
P3.5/T1	9	12	P1.0/AIN0
GND	10	11	P3.7

注：类似的还有Philips公司的
87LPC64，20引脚
8XC748/750/（751），24引脚
8X749（752），28引脚
8XC754，28引脚
等等

P3第二功能各引脚功能定义:

P3. 0: RXD串行口输入

P3. 1: TXD串行口输出

P3. 2: $\overline{\text{INT0}}$ 外部中断0输入

P3. 3: $\overline{\text{INT1}}$ 外部中断1输入

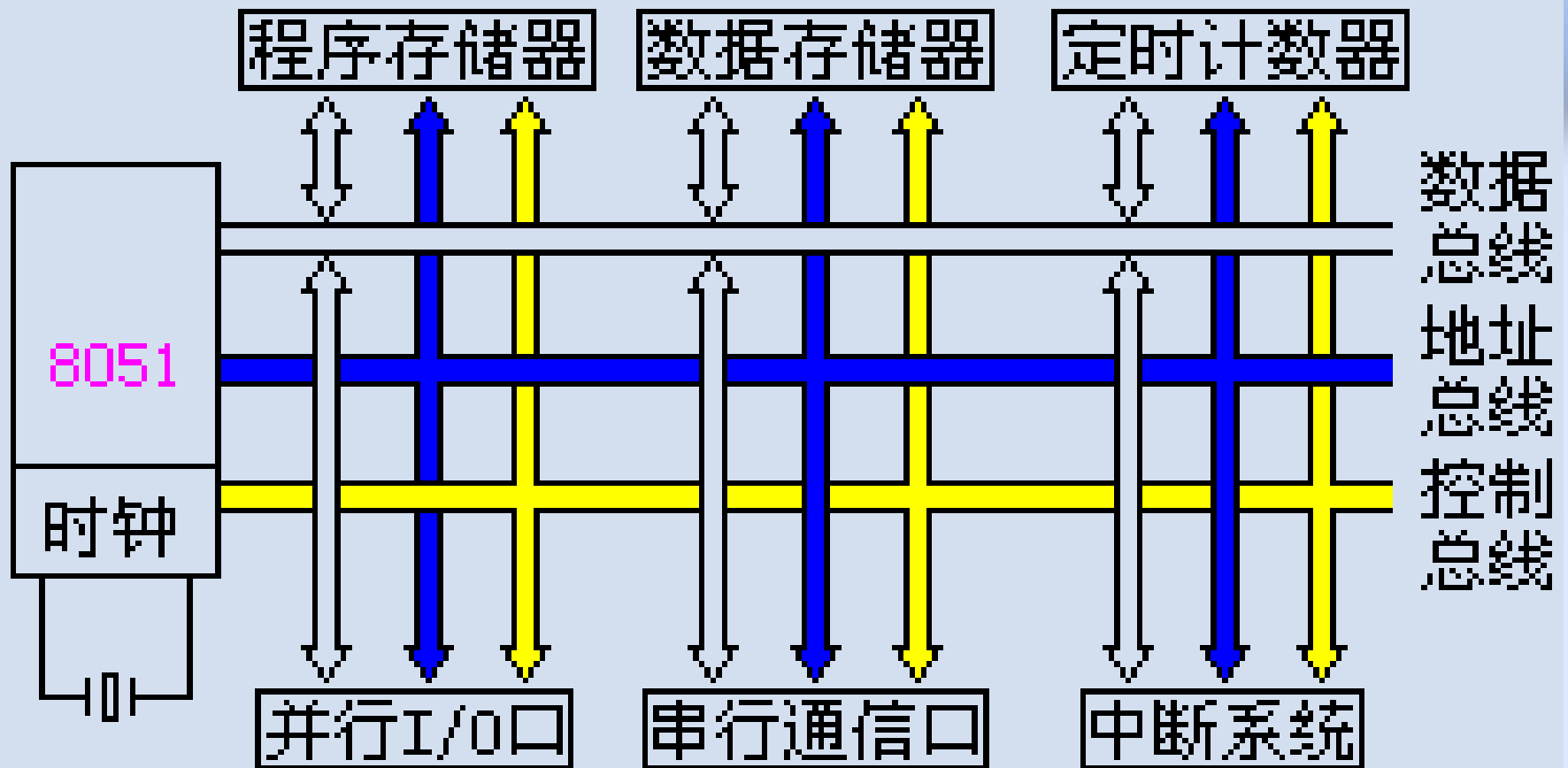
P3. 4: T0定时器0外部输入

P3. 5: T1定时器1外部输入

P3. 6: $\overline{\text{WR}}$ 外部写控制

P3. 7: $\overline{\text{RD}}$ 外部读控制

8051 内部结构



总线（**BUS**）是计算机各部件之间传送信息的公共通道。微机中有内部总线和外部总线两类。内部总线是**CPU**内部之间的连线。外部总线是指**CPU**与其它部件之间的连线。外部总线有三种：数据总线**DB**（**Data Bus**），地址总线 **AB**（**Address Bus**）和控制总线 **CB**（**Control Bus**）。

- **CPU**: 由运算和控制逻辑组成，同时还包括中断系统和部分外部特殊功能寄存器；
- **RAM**: 用以存放可以读写的数据，如运算的中间结果、最终结果以及欲显示的数据；
- **ROM**: 用以存放程序、一些原始数据和表格；
- **I/O口**: 四个8位并行I/O口，既可用作输入，也可用作输出；
- **T/C**: 两个定时/计数器，既可以工作在定时模式，也可以工作在计数模式；

- 五个中断源的中断控制系统；
- 一个全双工**UART**（通用异步接收发送器）的串行**I/O**口，用于实现单片机之间或单片机与微机之间的串行通信；
- 片内振荡器和时钟产生电路，石英晶体和微调电容需要外接。最高振荡频率取决于单片机型号及性能。

C51知识

- C语言是一种编译型程序设计语言，它兼顾了多种高级语言的特点，并具备汇编语言的功能。目前，使用C语言进行程序设计已经成为软件开发的一个主流。用C语言开发系统可以大大缩短开发周期，明显增强程序的可读性，便于改进、扩充和移植。而针对8051的C语言日趋成熟，成为了专业化的实用高级语言。

C-51的特点

C语言作为一种非常方便的语言而得到广泛的支持，很多硬件开发都用C语言编程，如：各种单片机、DSP、ARM等。

C语言程序本身不依赖于机器硬件系统，基本上不作修改就可将程序从不同的单片机中移植过来。

C提供了很多数学函数并支持浮点运算，开发效率高，故可缩短开发时间，增加程序可读性和可维护性。

C-51与ASM-51相比，有如下优点：

1. 对单片机的指令系统不要求了解，仅要求对8051 的存贮器结构有初步了解；
2. 寄存器分配、不同存贮器的寻址及数据类型等细节可由编译器管理；
3. 程序有规范的结构，可分成不同的函数，这种方式可使程序结构化；
4. 提供的库包含许多标准子程序，具有较强的数据处理能力；
5. 由于具有方便的模块化编程技术，使已编好程序可容易地移植；

C-51的数据类型

基本数据类型

类型	符号	关键字	所占位数	数的表示范围
整型	有	(signed) int	16	-32768~32767
		(signed) short	16	-32768~32767
		(signed) long	32	-2147483648~2147483647
	无	unsigned int	16	0~65535
		unsigned short int	16	0~65535
		unsigned long int	32	0~4294967295
实型	有	float	32	3.4e-38~3.4e38
	有	double	64	1.7e-308~1.7e308
字符型	有	char	8	-128~127
	无	unsigned char	8	0~255

C语言中的基本数据类型

类型	符号	关键字	所占位数	数的表示范围
整型	有	(signed) int	16	-32768~32767
		(signed) short	16	-32768~32767
		(signed) long	32	-2147483648~2147483647
	无	unsigned int	16	0~65535
		unsigned short int	16	0~65535
		unsigned long int	32	0~4294967295
实型	有	float	32	3.4e-38~3.4e38
	有	double	64	1.7e-308~1.7e308
字符型	有	char	8	-128~127
	无	unsigned char	8	0~255

C-51的数据类型扩充定义

sfr: 特殊功能寄存器声明

sfr16: sfr的16位数据声明

sbit: 特殊功能位声明

bit: 位变量声明

例: sfr SCON = 0X98;

sfr16 T2 = 0xCC;

sbit OV = PSW^2;

C-51数据的存储类型

例：

数据类型

变量名

char

var1;

bit

flags;

unsigned char

vextor[10];

int

www;

注意：变量名不能用C语言中的关键字表示。

C-51的包含的头文件

通常有: `reg51.h` `reg52.h` `math.h`
`ctype.h` `stdio.h` `stdlib.h` `absacc.h`

常用有: `reg51.h` `reg52.h`

（定义特殊功能寄存器和位寄存器）；

`math.h` （定义常用数学运算）；

C-51的运算符

与C语言基本相同:

+ - * / (加 减 乘 除)

> >= < <= (大于 大于等于 小于 小于等于)

== != (测试等于 测试不等于)

&& || ! (逻辑与 逻辑或 逻辑非)

>> << (位右移 位左移)

& | (按位与 按位或)

^ ~ (按位异或 按位取反)

C-51的基本语句

与标准C语言基本相同：

if

选择语言

while

循环语言

for

循环语言

switch/case

多分支选择语言

do-while

循环语言

中断服务程序

函数名 () interrupt **n** using **m**

{

函数内部实现

}

I/O口定义

```
sbit beep=P2^3;
```


单片机主要掌握以下几点

- 最小系统能够运行起来的必要条件。
 - 1.电源
 - 2.晶振
 - 3.复位电路
- 对单片机任意IO口的随意操作
 - 1.输出控制电平高低
 - 2.输出检测电平高低。
- 定时器：重点掌握最常用的方式2
- 中断：外部中断、定时器中断、串口中断
- 串口通信：单片机之间、单片机与计算机间

KEIL的使用

- 掌握**KEIL**工程的建立和软件的开发；
- 掌握**KEIL**中软件仿真的基本应用；
- 掌握用**KEIL**直接硬件仿真**TX-1C**单片机学习板的方法

Main 函数

格式: `void main()`

特点: 无返回值, 无参。

任何一个C程序有且仅有一个main函数, 它是整个程序开始执行的入口。

例: `void main()`

{

总程序从这里开始执行;

其他语句;

}

专用寄存器名称	符号	地址	位地址与位名称							
			D7	D6	D5	D4	D3	D2	D1	D0
P0 口	P0	80H	87	86	85	84	83	82	81	80
堆栈指针	SP	81H								
数据指针低字节	DPL DPTR DPH	82H								
数据指针高字节		83H								
定时器/计数器控制	TCON	88H	TF1 8F	TR1 8E	TF0 8D	TR0 8C	IE1 8B	IT1 8A	IE0 89	IT0 88
定时器/计数器方式控制	TMOD	89H	GATE	C/ \bar{T}	M ₁	M ₀	GATE	C/ \bar{T}	M ₁	M ₀
定时器/计数器 0 低字节	TL0	8AH								
定时器/计数器 1 低字节	TL1	8BH								
定时器/计数器 0 高字节	TH0	8CH								
定时器/计数器 1 高字节	TH1	8DH								
P1 口	P1	90H	97	96	95	94	93	92	91	90
电源控制	PCON	97H	SMOD	—	—	—	GF1	GF0	PD	IDL
串行控制	SCON	98H	SMO 9F	SM1 9E	SM2 9D	REN 9C	TB8 9B	RB8 9A	TI 99	RI 98

专用寄存器名称	符号	地址	位地址与位名称							
			D7	D6	D5	D4	D3	D2	D1	D0
串行数据缓冲器	SBUF	99H								
P2 口	P2	A0H	A7	A6	A5	A4	A3	A2	A1	A0
中断允许控制	IE	A8H	EA AF	— —	ET2 AD	ES AC	ET1 AB	EX1 AA	ET0 A9	EX0 A8
P3 口	P3	B0H	B7	B6	B5	B4	B3	B2	B1	B0
中断优先级控制	IP	B8H	— —	— —	PT2 BD	PS BC	PT1 BB	PX1 BA	PT0 B9	PX0 B8
定时器/计数器 2 控制	T2CON *	C8H	TE2 CF	EXF2 CE	RCLK CD	TCLK CC	EXEN2 CB	TR2 CA	C/T $\bar{2}$ C9	CP/ PL2 C8
定时器/计数器 2 自动重装载低字节	RLDL *	CAH								
定时器/计数器 2 自动重装载高字节	RLDH *	CBH								
定时器/计数器 2 低字节	TL2 *	CCH								
定时器/计数器 2 高字节	TH2 *	CDH								
程序状态字	PSW	D0H	Cr D7	AC D6	F0 D5	RS1 D4	RS0 D3	OV D2	— D1	P D0
累加器	A	E0H	E7	E6	E5	E4	E3	E2	E1	E0
B 寄存器	B	F0H	F7	F6	F5	F4	F3	F2	F1	F0

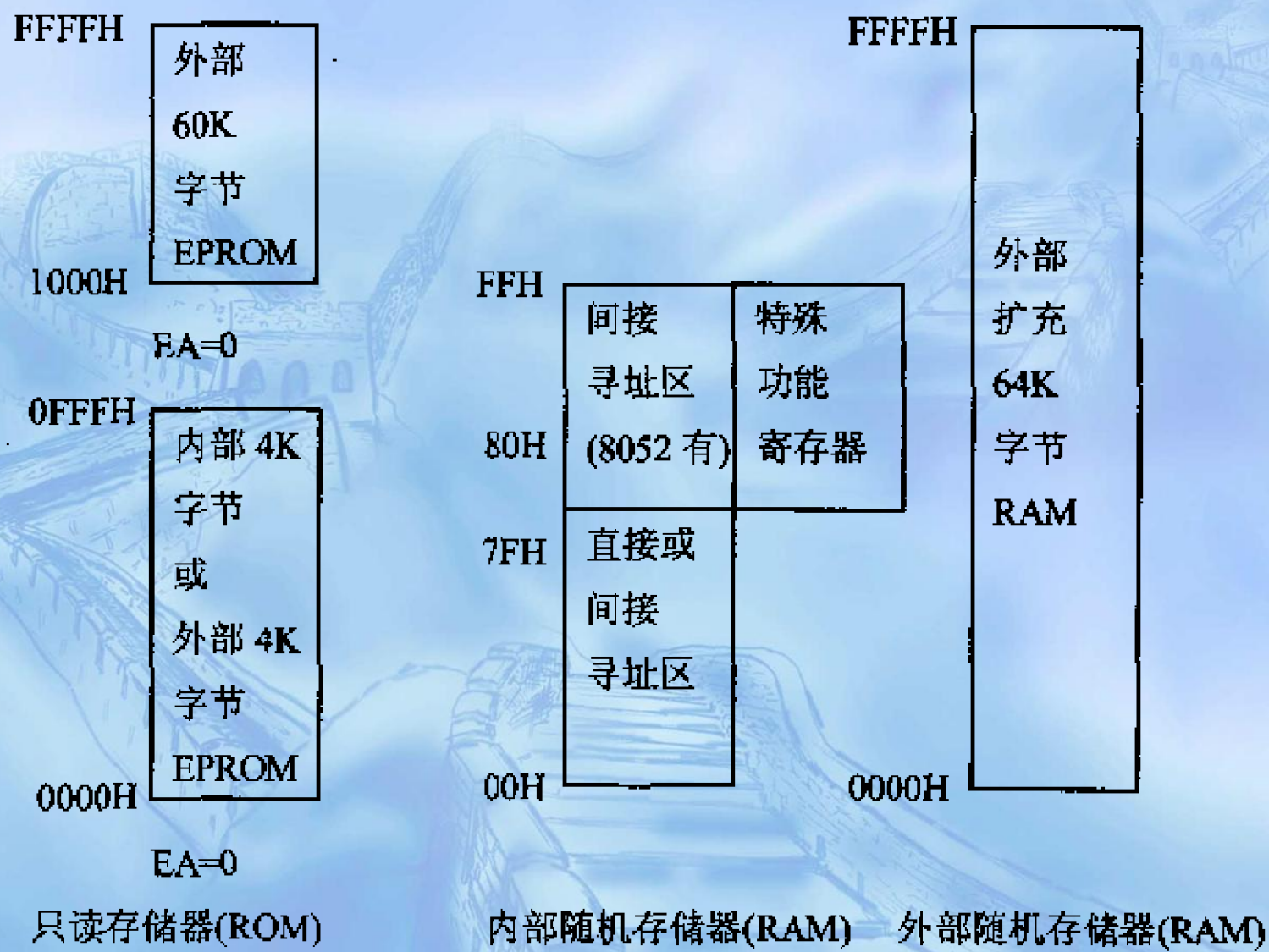


图 1-4 8051 内存分配图

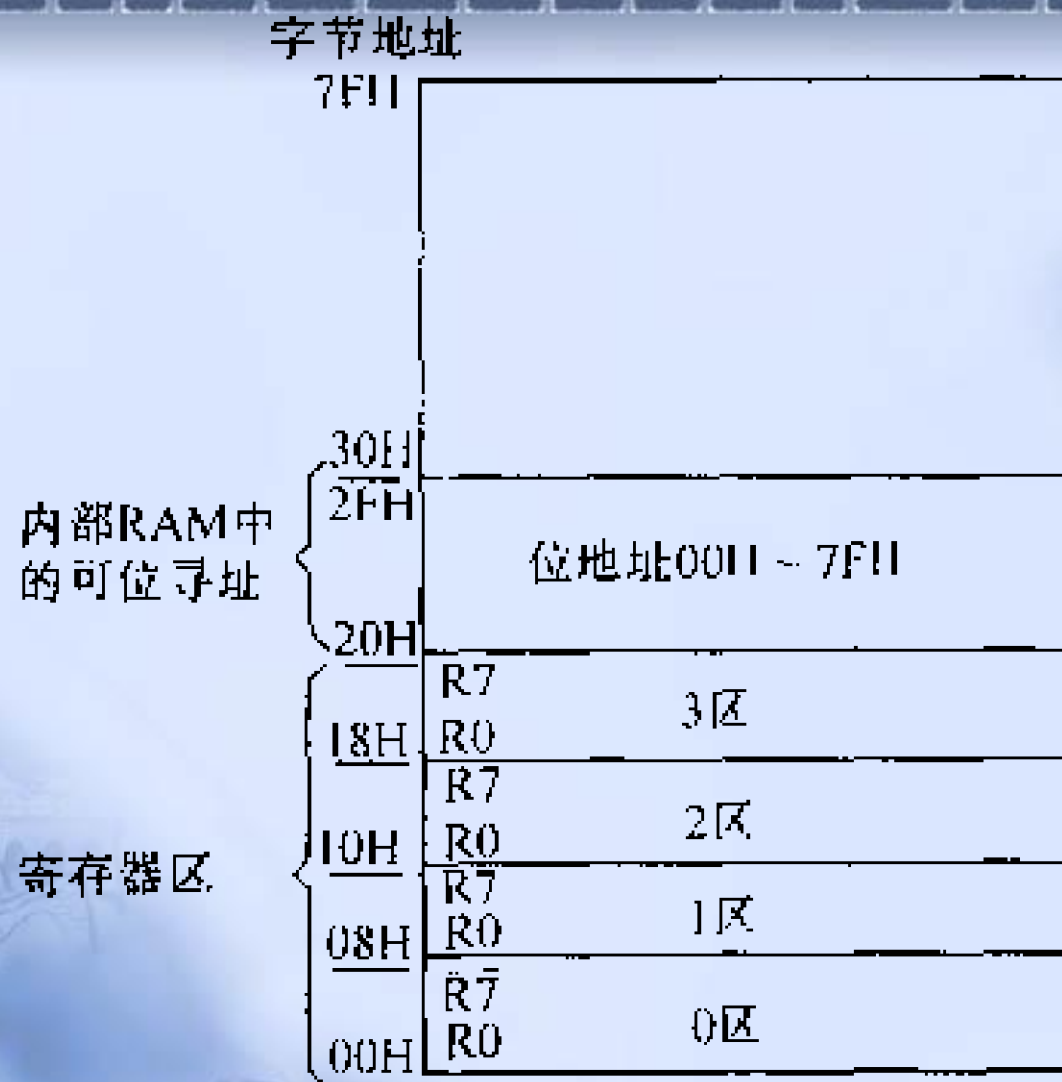


图 2-5 MCS-51 内部 RAM 的结构

F8								
F0	B							
E8								
E0	A							
D8								
D0	PSW							
C8								
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	ICON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH				PCON 87

可位寻址

E0H	E7	E6	E5	E4	E3	E2	E1	E0	A
D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
B8H	—	—	—	BC	BB	BA	B9	B8	IP
B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8H	AF	—	—	AC	AB	AA	A9	A8	IE
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
88H	8F	9E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

课后练习

- 对照TX-1C单片机学习板原理图写程序
 - 用位操作和总线操作两种方法完成以下题目
- 1.熟练建立**KEIL**工程
 - 2.点亮第一个发光管.
 - 3.点亮最后一个发光管
 - 4.点亮1、3、5、7
 - 5.点亮二、四、五、六
 - 6.尝试让第一个发光管闪烁
 - 7.尝试设计出流水灯程序

自己动手实践需要硬件材料

- DIP40脚座一个。
- 杜邦线，单头带冒，40根。
- 单排针 两排。
- 小电路板一块。
- DS12C887 一片（旧10元，新22元）。
- 焊锡、

单片机工作的基本时序

- 机器周期和指令周期
- (1) 振荡周期: 也称时钟周期, 是指为单片机提供时钟脉冲信号的振荡源的周期, **TX实验板上为11.0592MHZ。**
- (2) 状态周期: 每个状态周期为时钟周期的 **2** 倍, 是振荡周期经二分频后得到的。
- (3) 机器周期: **一个机器周期**包含 **6** 个状态周期**S1~S6, 也就是 12 个时钟周期。** 在一个机器周期内, **CPU**可以完成一个独立的操作。
- (4) 指令周期: 它是指**CPU**完成一条操作所需的全部时间。 每条指令执行时间都是有一个或几个机器周期组成。**MCS - 51** 系统中, 有单周期指令、双周期指令和四周期指令。

第二讲

- 如何申请免费样片
- 简单延时程序
- 子程序调用
- 带参数子程序设计
- **C51**库函数的方便调用
- 如何驱动蜂鸣器
- 如何驱动继电器
- 集电极开路的概念及应用。

While （表达式）

{语句（内部也可为空）}

特点：先判断表达式，后执行语句。

原则：若表达式不是0，即为真，那么执行语句。否则跳出while语句。

For (表达式1;表达式2;表达式3)

{ 语句（内部可为空） }

执行过程：

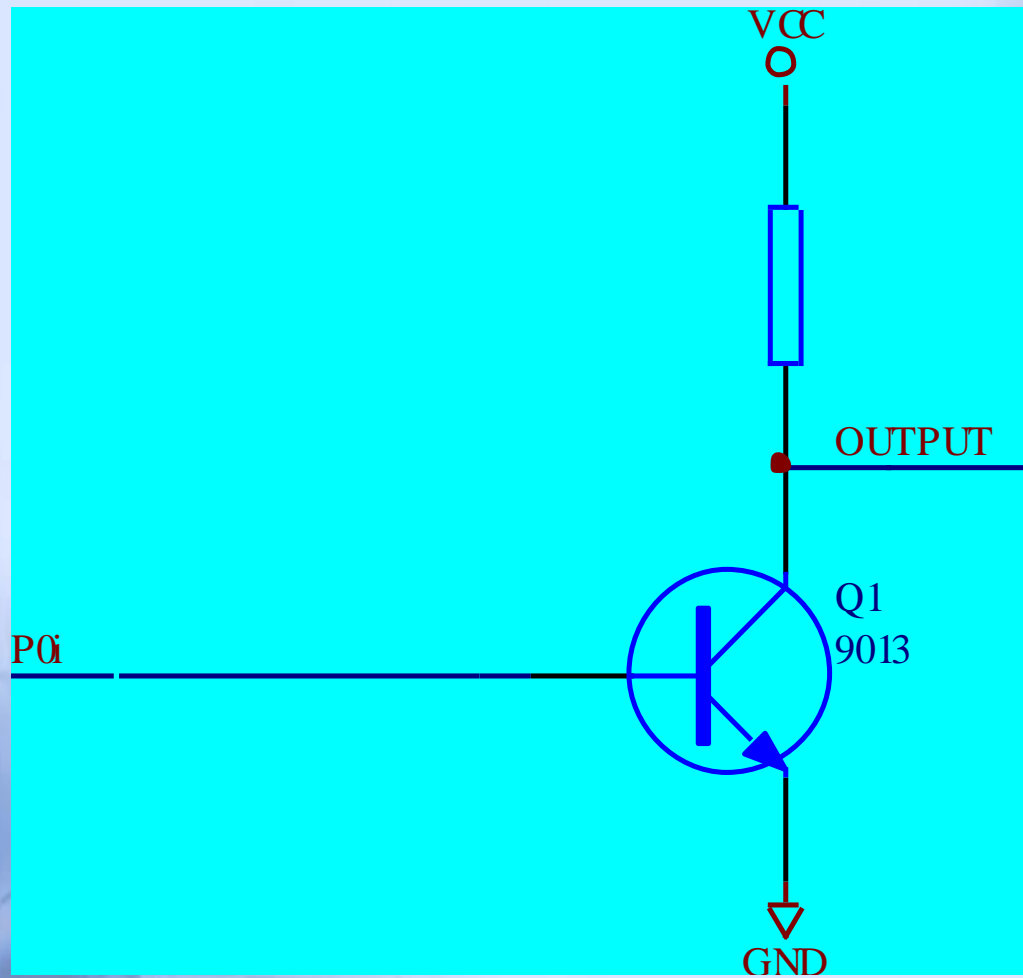
1.求解一次表达式1.

2.求解表达式2，若其值为真（非0即为真），则执行for中语句。然后执行第3步。否则结束for 语句，直接跳出，不再执行第3步。

3.求解表达式3.

4.跳到第2步重复执行。

集电极开路典型图



课后练习

1. 第一个发光管以间隔**200ms**闪烁
2. 8个发光管由上至下间隔**1s**流动，其中每个管亮**500ms**,灭**500ms**,亮时蜂鸣器响，灭时关闭蜂鸣器，一直重复下去。
3. 8个发光管来回流动，第个管亮**100ms**,流动时让蜂鸣器发出“滴滴”声。
4. 用8个发光管演示出**8位二进制数累加**过程。
5. 8个发光管间隔**200ms**由上至下，再由下至上，再重复一次，然后全部熄灭再以**300ms**间隔全部闪烁**5次**。重复此过程。
6. 间隔**300ms**第一次一个管亮流动一次，第二次两个管亮流动，依次到8个管亮，然后重复整个过程。
7. 间隔**300ms**先奇数亮再偶数亮，循环三次；一个灯上下循环三次；两个分别从两边往中间流动三次；再从中间往两边流动三次；8个全部闪烁**3次**；关闭发光管，程序停止。

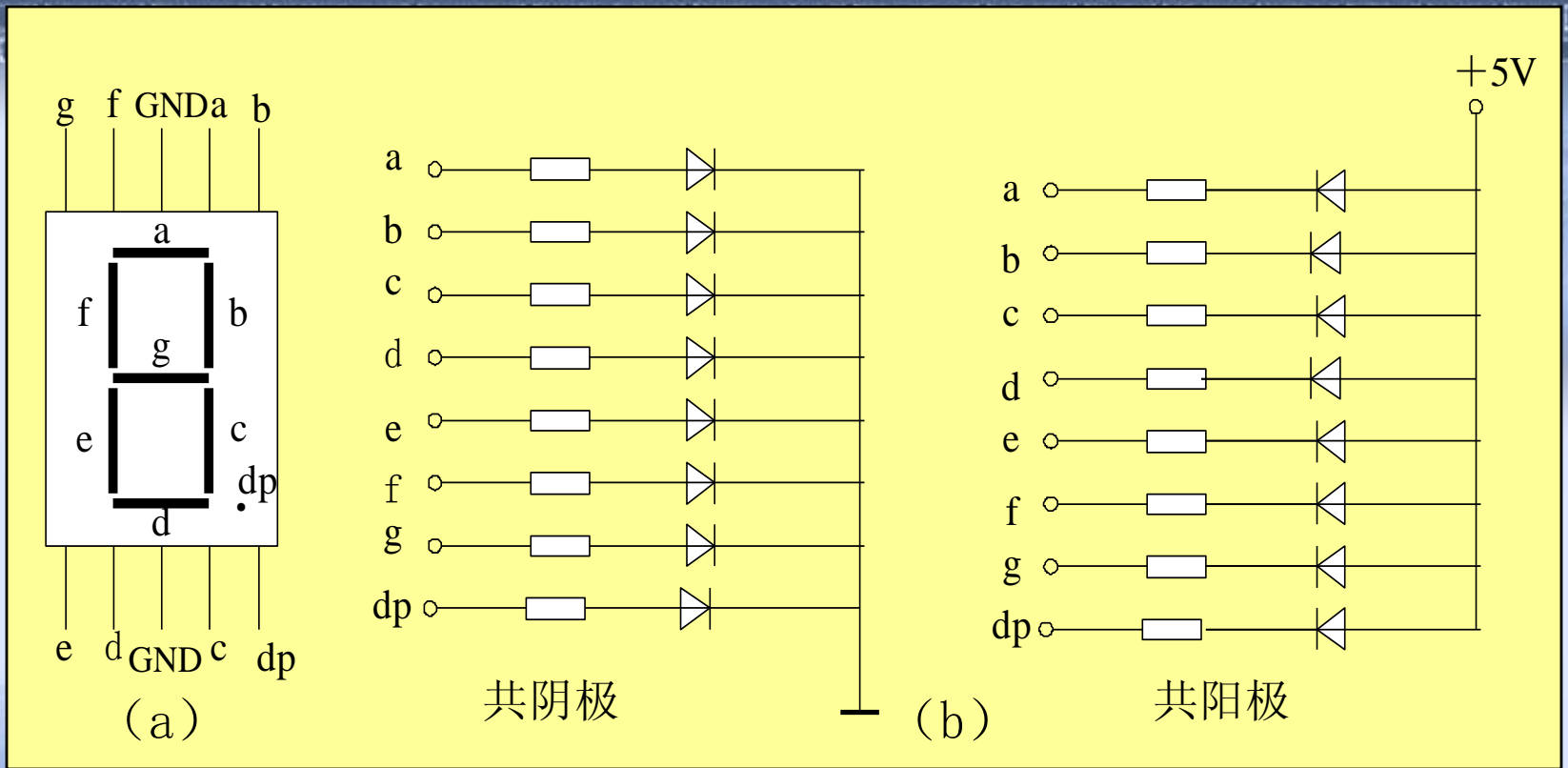
第三讲

- 数码管是如何显示出字符的
- 数码管静态显示与动态显示原理
- 中断概念
- 单片机的定时器应用

显示器及其接口

单片机系统中常用的显示器有：

发光二极管LED(Light Emitting Diode)显示器、液晶LCD(Liquid Crystal Display)显示器、CRT显示器等。LED、LCD显示器有两种显示结构：段显示（7段、米字型等）和点阵显示（ 5×8 、 8×8 点阵等）。



使用LED显示器时，要注意区分这两种不同的接法。为了显示数字或字符，必须对数字或字符进行编码。七段数码管加上一个小数点，共计8段。因此为LED显示器提供的编码正好是一个字节。TX实验板用共阴LED显示器，根据电路连接图显示16进制数的编码已列在下表。

共阴数码管码表

- 0x3f , 0x06 , 0x5b , 0x4f , 0x66 , 0x6d ,
- 0 1 2 3 4 5
- 0x7d , 0x07 , 0x7f , 0x6f , 0x77 , 0x7c ,
- 6 7 8 9 A B
- 0x39 , 0x5e , 0x79 , 0x71 , 0x00
- C D E F 无显示

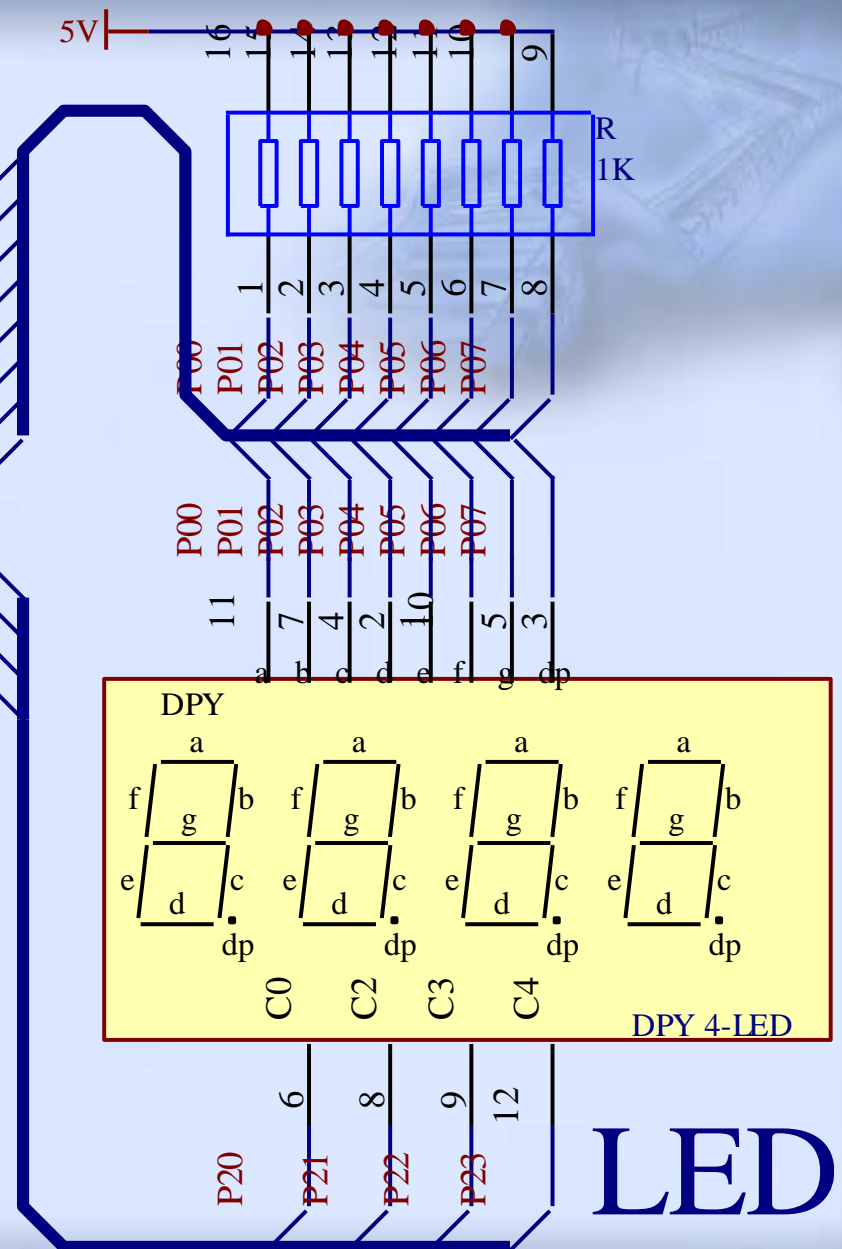
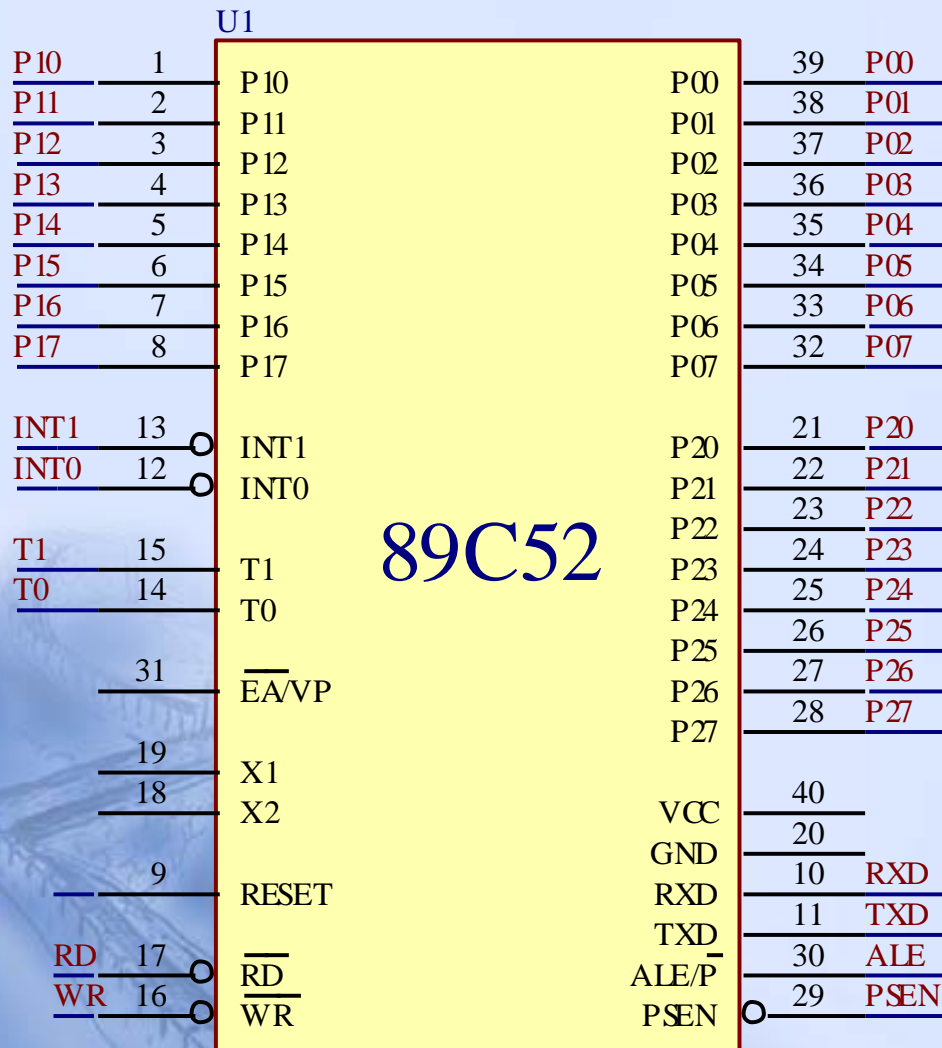
LED数码显示方式及电路

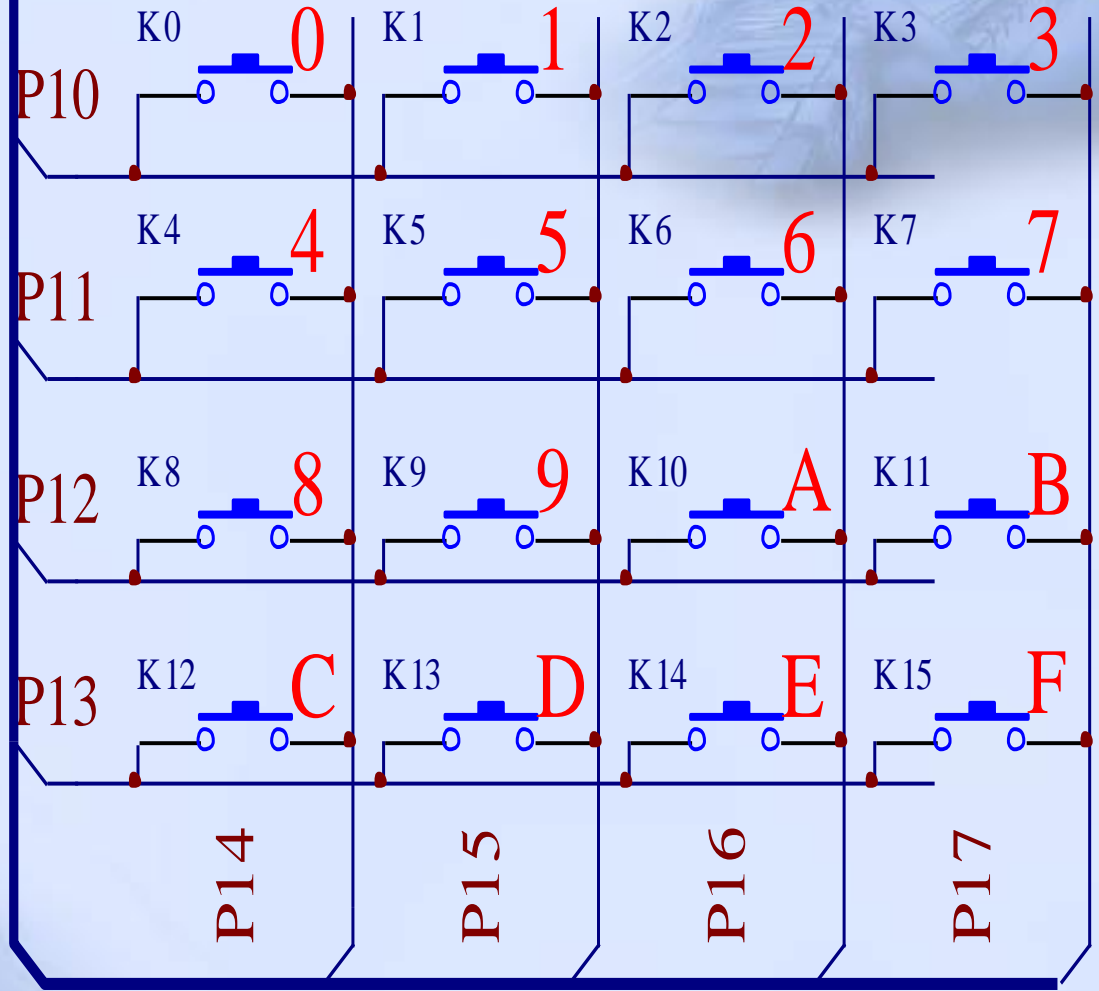
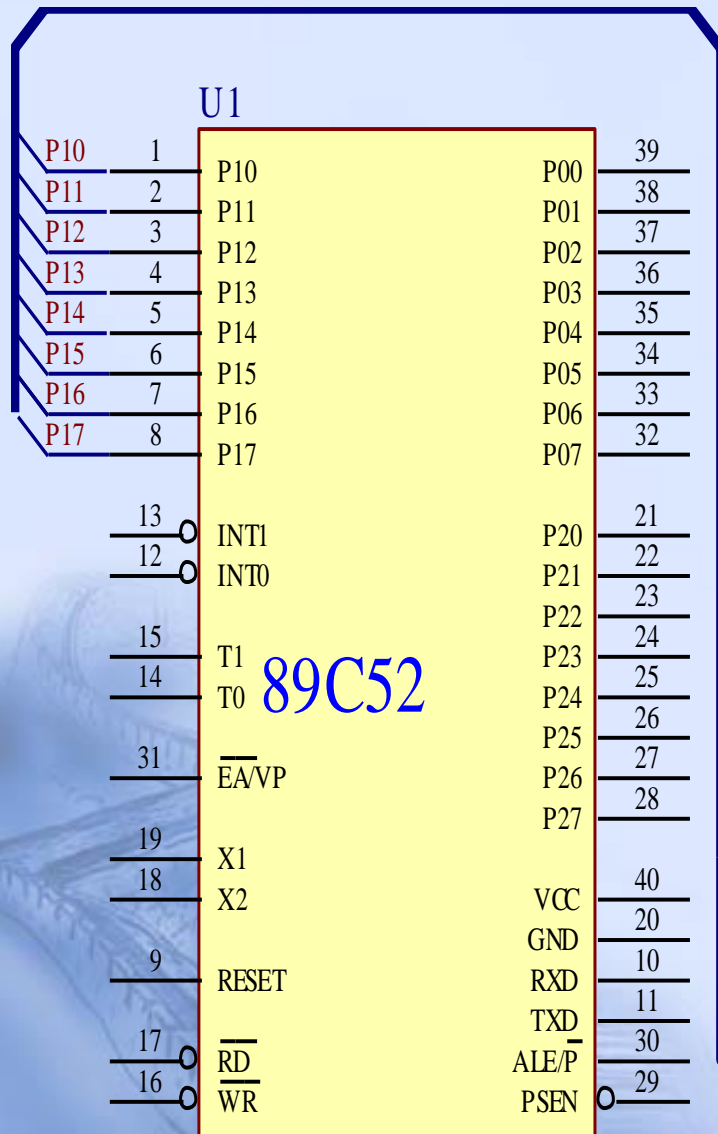
静态显示方式

LED显示器工作方式有两种：静态显示方式和动态显示方式。静态显示的特点是每个数码管的段选必须接一个8位数据线来保持显示的字形码。当送入一次字形码后，显示字形可一直保持，直到送入新字形码为止。这种方法的优点是占用CPU时间少，显示便于监测和控制。缺点是硬件电路比较复杂，成本较高。

动态显示

动态显示的特点是将所有位数数码管的段选线并联在一起，由位选线控制是哪一位数码管有效。选亮数码管采用动态扫描显示。所谓动态扫描显示即轮流向各位数码管送出字形码和相应的位选，利用发光管的余辉和人眼视觉暂留作用，使人的感觉好像各位数码管同时都在显示。动态显示的亮度比静态显示要差一些，所以在选择限流电阻时应略小于静态显示电路中的。





中断系统

3.1 80C51的中断系统

3.1.1 80C51的中断系统结构

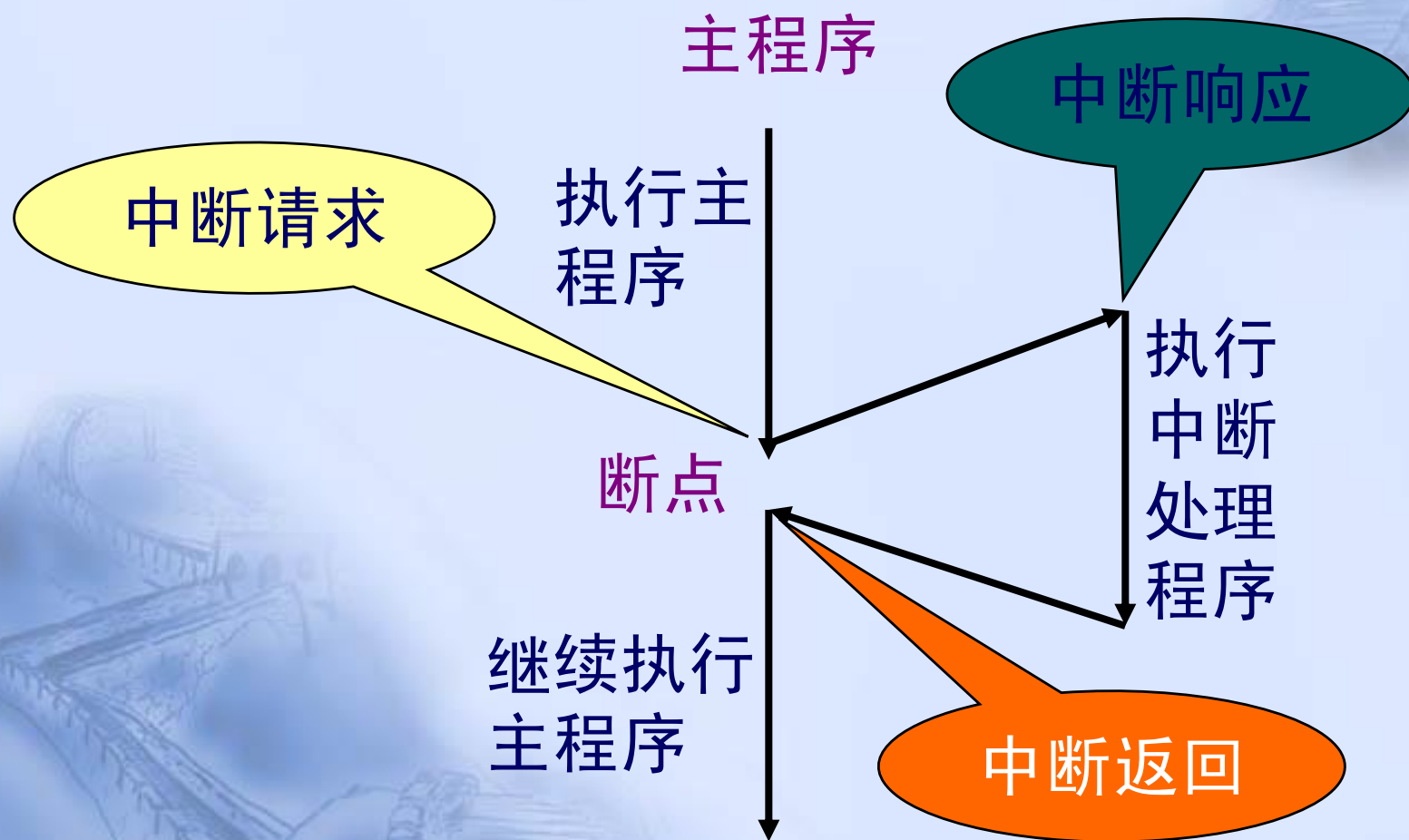
一、中断的概念

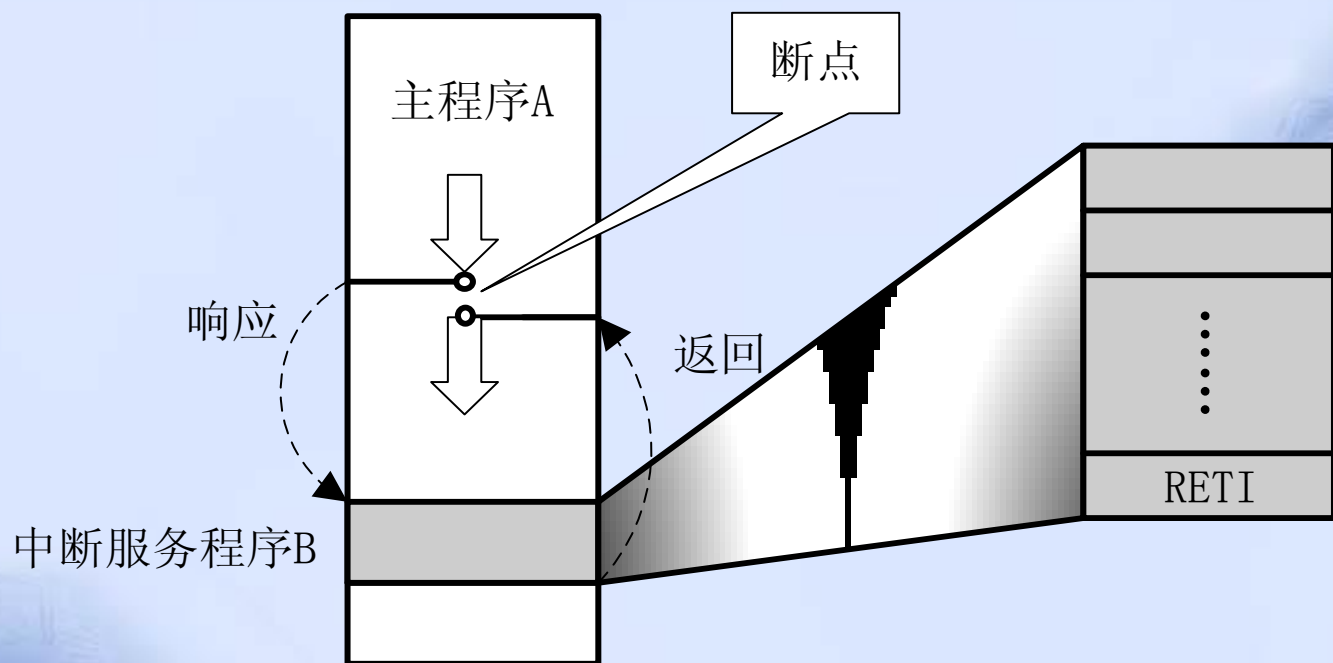
CPU在处理某一事件A时，发生了另一事件B请求CPU迅速去处理（**中断发生**）；

CPU暂时中断当前的工作，转去处理事件B（**中断响应和中断服务**）；

待CPU将事件B处理完毕后，再回到原来事件A被中断的地方继续处理事件A（**中断返回**），这一过程称为**中断**。

MCS-51单片机的中断系统结构





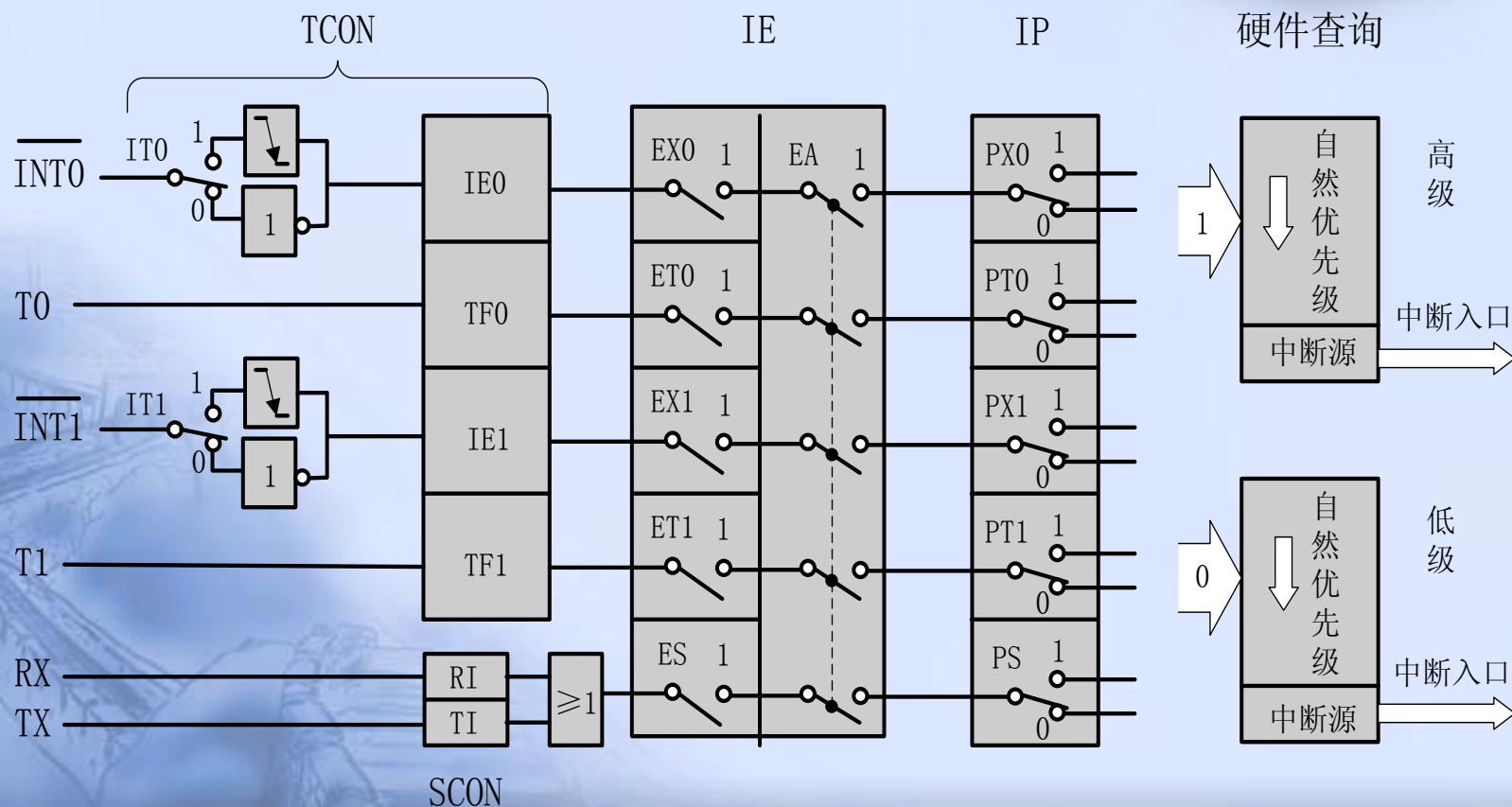
引起**CPU**中断的根源，称为**中断源**。中断源向**CPU**提出的中断请求。**CPU**暂时中断原来的事务**A**，转去处理事件**B**。对事件**B**处理完毕后，再回到原来被中断的地方（即**断点**），称为中断返回。实现上述中断功能的部件称为**中断系统**（中断机构）。

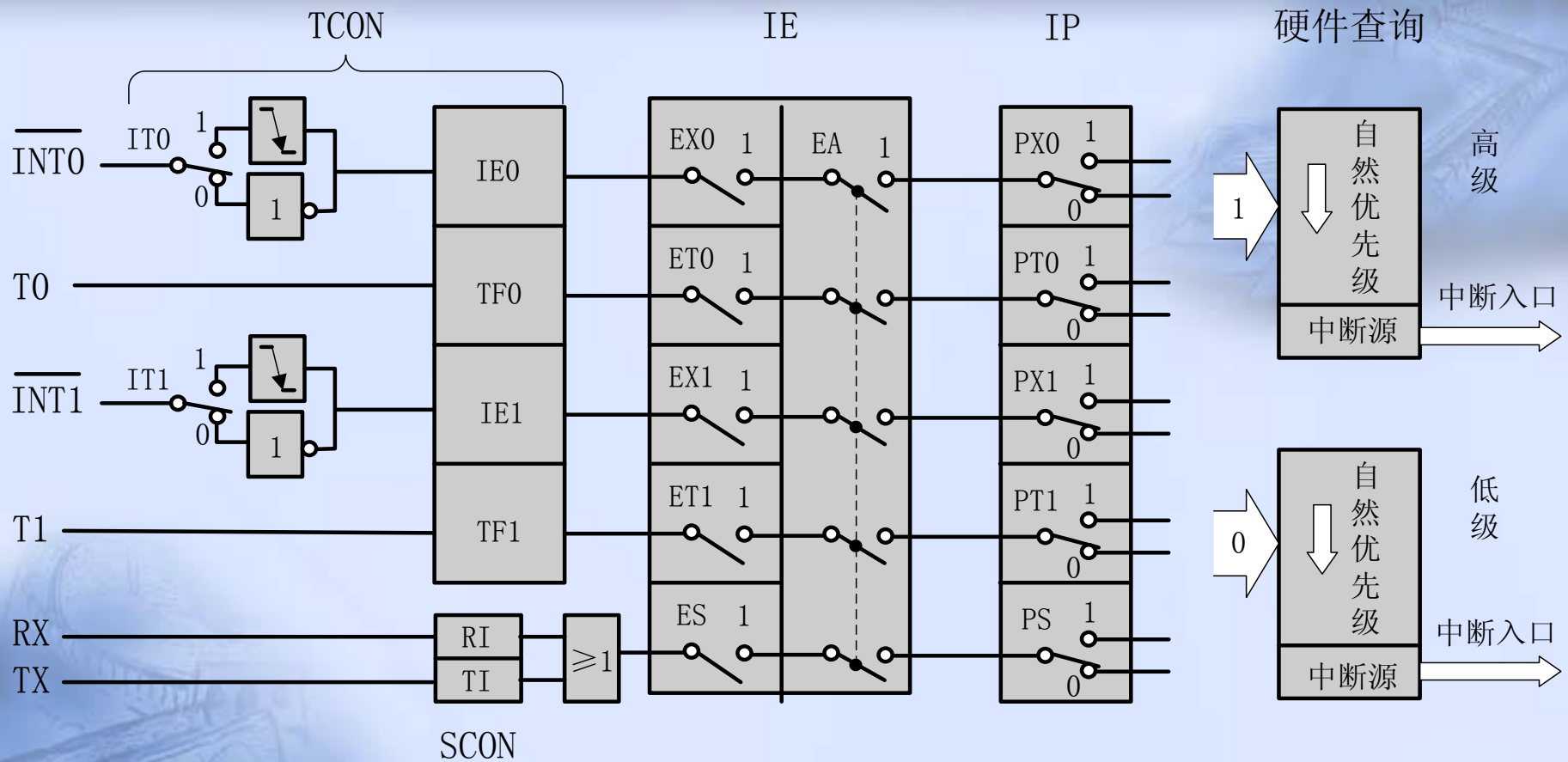
随着计算机技术的应用，人们发现中断技术不仅解决了快速主机与慢速I/O设备的数据传送问题，而且还具有如下优点：

- **分时操作**。**CPU**可以分时为多个I/O设备服务，提高了计算机的利用率；
- **实时响应**。**CPU**能够及时处理应用系统的随机事件，系统的实时性大大增强；
- **可靠性高**。**CPU**具有处理设备故障及掉电等突发性事件能力，从而使系统可靠性提高。

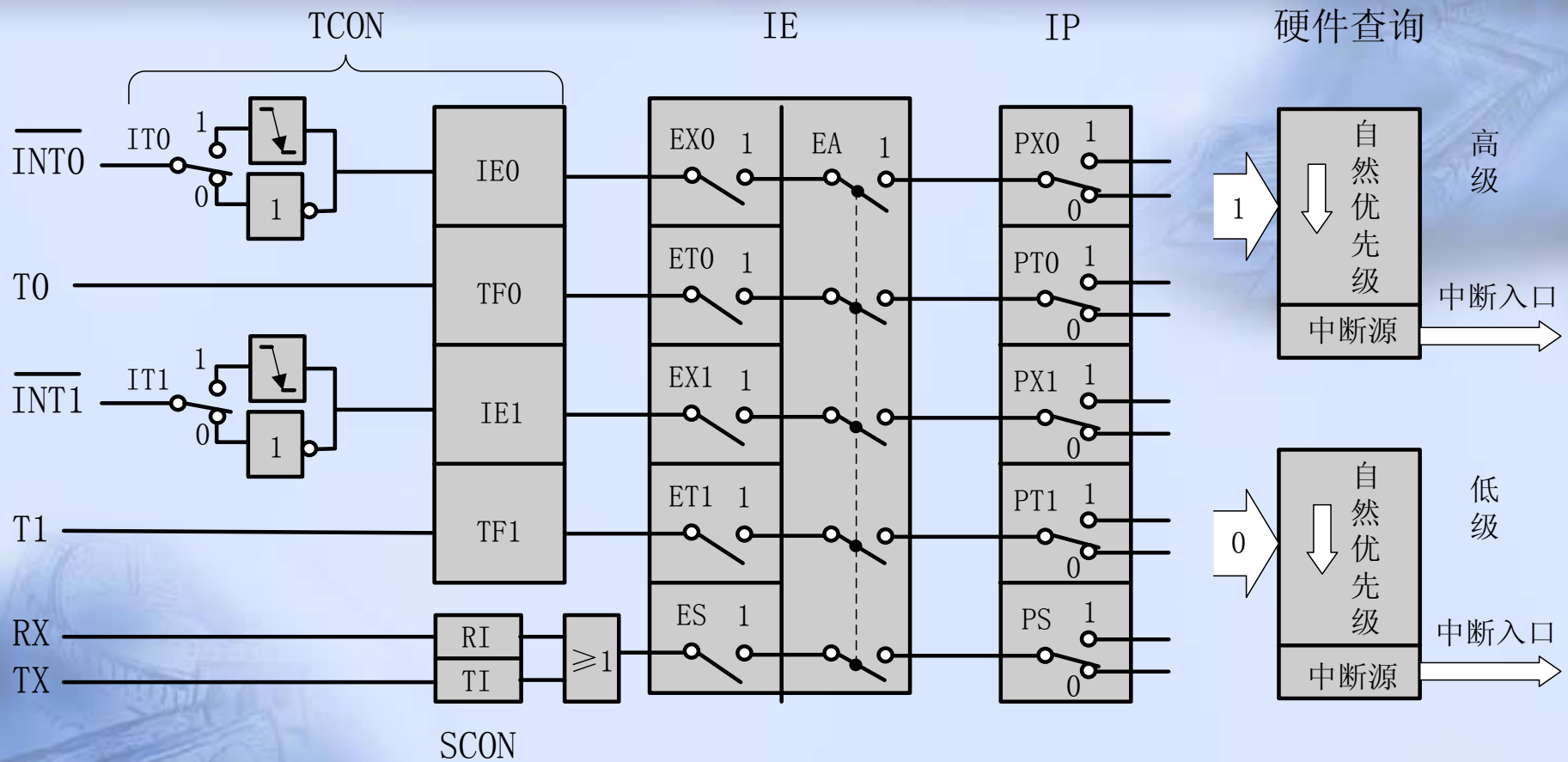
80C51中断系统的结构

80C51的中断系统有**5**个中断源（8052有 6 个），**2**个优先级，可实现二级中断嵌套。

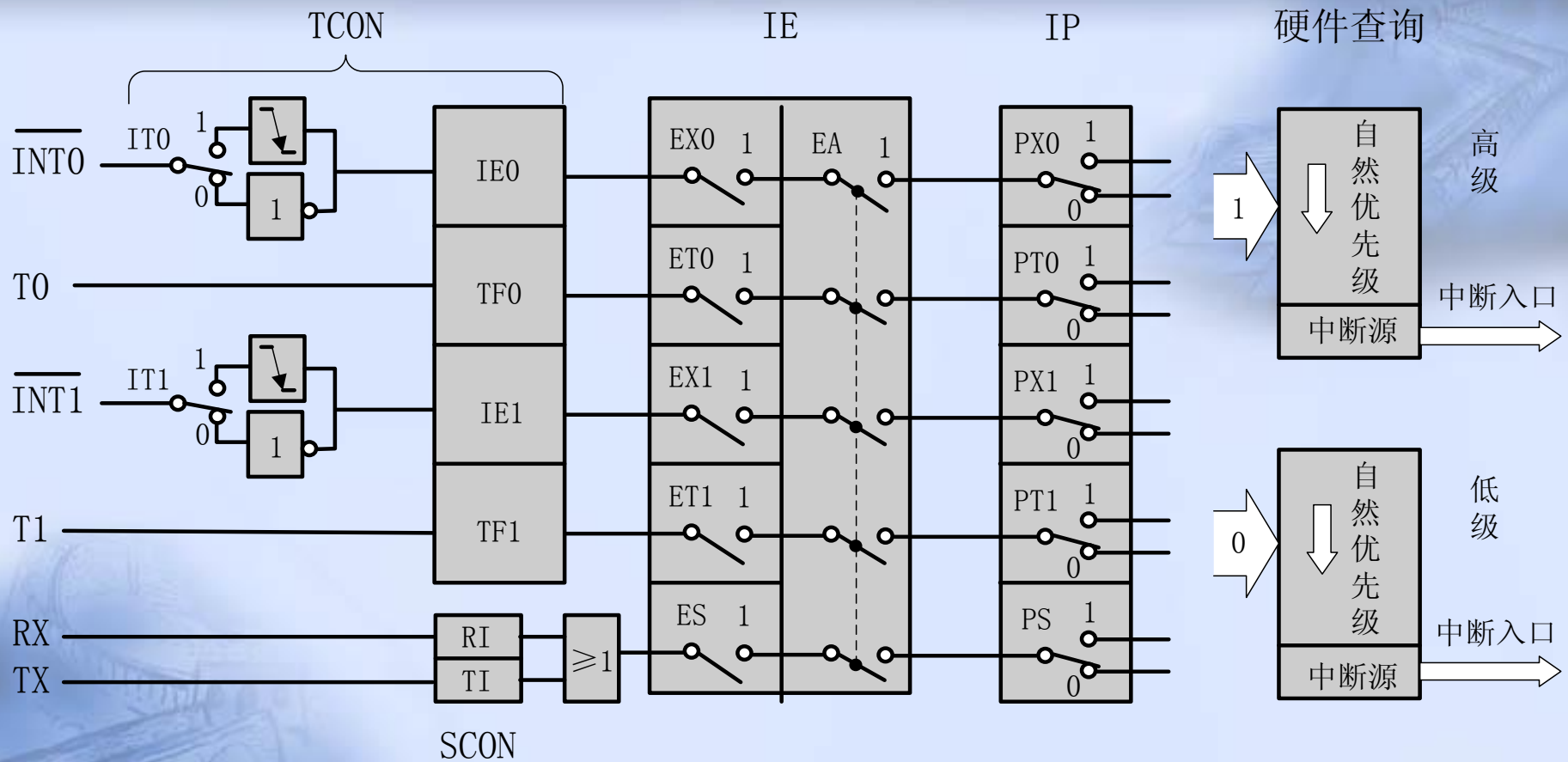




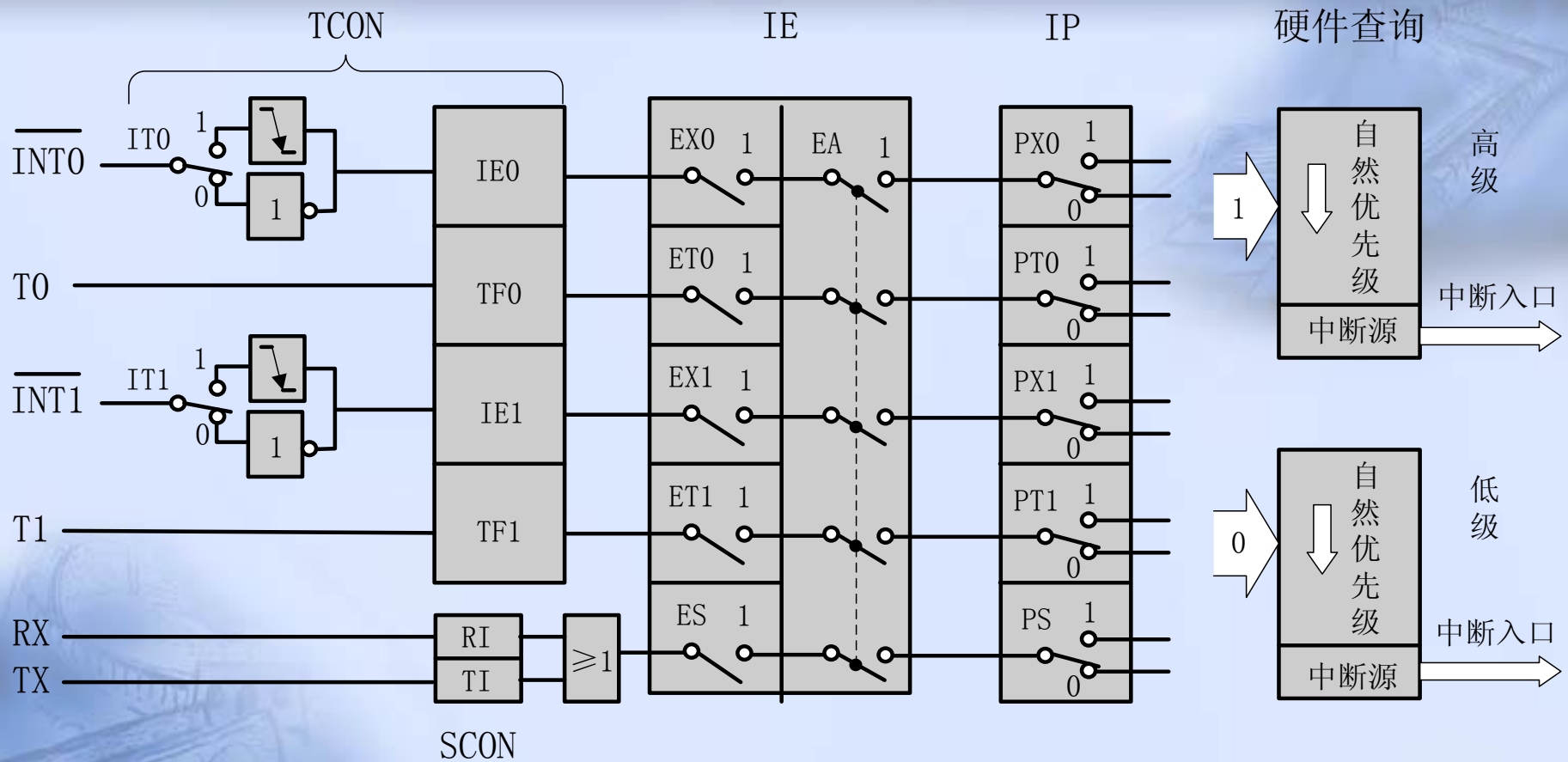
1、（P3.2）可由IT0 (TCON. 0) 选择其为低电平有效还是下降沿有效。当CPU检测到P3. 2引脚上出现有效的中断信号时，中断标志IE0 (TCON. 1) 置1，向CPU申请中断。



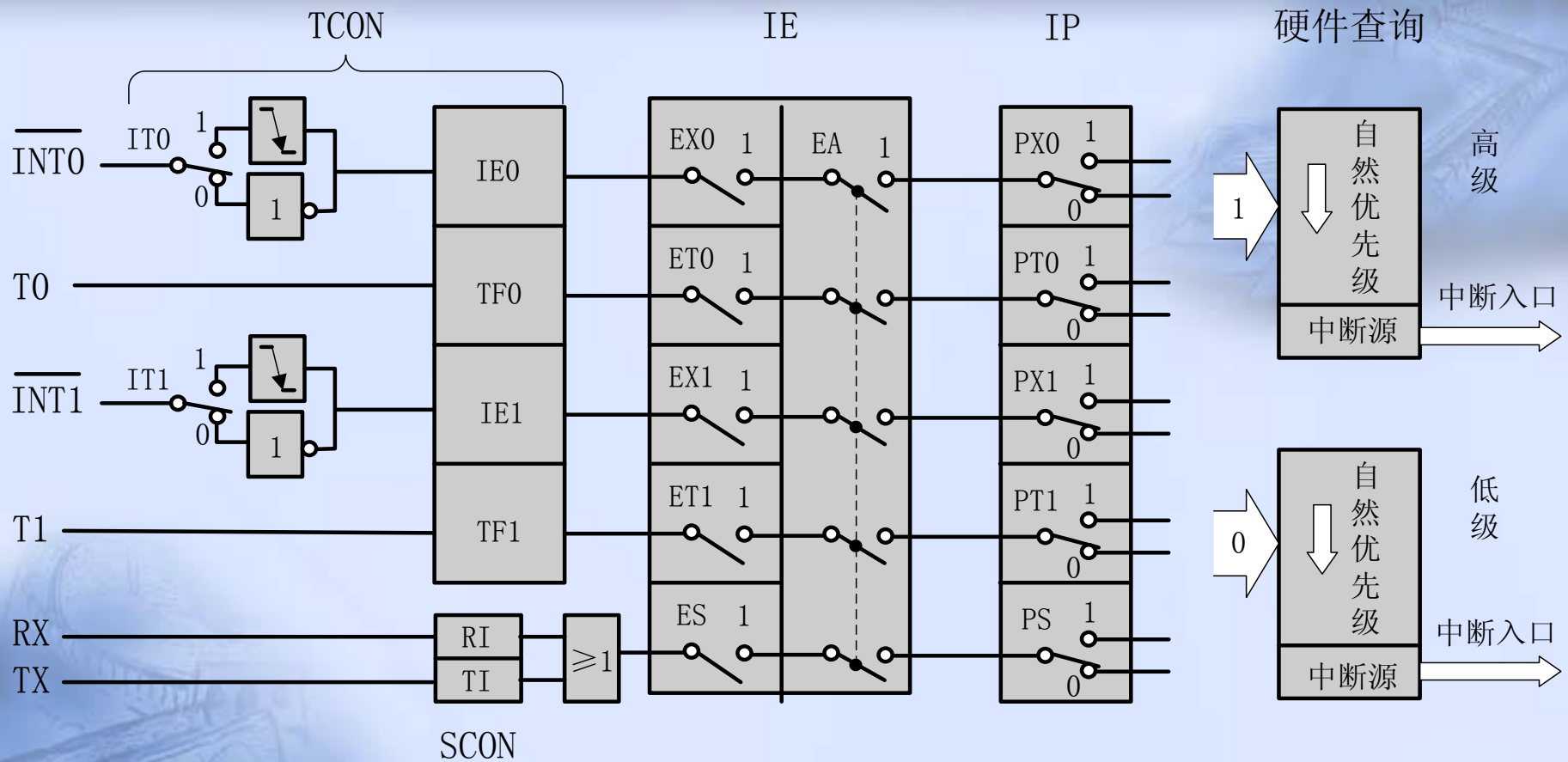
2、(P3.3) 可由IT1 (TCON. 2) 选择其为低电平有效还是下降沿有效。当CPU检测到P3.3引脚上出现有效的中断信号时，中断标志IE1 (TCON. 3) 置1, 向CPU申请中断。



3、**TF0** (TCON.5)，片内定时/计数器T0溢出中断请求标志。当定时/计数器T0发生溢出时，置位TF0，并向CPU申请中断。



4、**TF1** (TCON. 7)，片内定时/计数器T1溢出中断请求标志。当定时/计数器T1发生溢出时，置位TF1，并向CPU申请中断。



5、**RI** (SCON.0) 或 **TI** (SCON.1)，串行口中断请求标志。当串行口接收完一帧串行数据时置位RI或当串行口发送完一帧串行数据时置位TI，向CPU申请中断。

二、中断请求标志

1、TCON的中断标志

位	7	6	5	4	3	2	1	0	
字节地址: 88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

IT0 (TCON.0)，外部中断0触发方式控制位。

当**IT0=0**时，为电平触发方式。

当**IT0=1**时，为边沿触发方式（下降沿有效）。

IE0 (TCON.1)，外部中断0中断请求标志位。

IT1 (TCON.2)，外部中断1触发方式控制位。

IE1 (TCON.3)，外部中断1中断请求标志位。

TF0 (TCON.5)，定时/计数器T0溢出中断请求标志位。

TF1 (TCON.7)，定时/计数器T1溢出中断请求标志位。

2、SCON的中断标志

位	7	6	5	4	3	2	1	0	
字节地址: 98H							TI	RI	SCON

- **RI (SCON.0)**，串行口接收中断标志位。当允许串行口接收数据时，每接收完一个串行帧，由硬件置位RI。注意，RI必须由软件清除。
- **TI (SCON.1)**，串行口发送中断标志位。当CPU将一个发送数据写入串行口发送缓冲器时，就启动了发送过程。每发送完一个串行帧，由硬件置位TI。CPU响应中断时，不能自动清除TI，TI必须由软件清除。

3.1.3 80C51中断的控制

一、中断允许控制

CPU对中断系统所有中断以及某个中断源的开放和屏蔽是由中断允许寄存器**IE**控制的。

位	7	6	5	4	3	2	1	0	
字节地址: A8H	EA			ES	ET1	EX1	ET0	EX0	IE

- **EX0(IE.0)**, 外部中断**0**允许位;
- **ET0(IE.1)**, 定时/计数器**T0**中断允许位;
- **EX1(IE.2)**, 外部中断**0**允许位;
- **ET1(IE.3)**, 定时/计数器**T1**中断允许位;
- **ES (IE.4)**, 串行口中断允许位;
- **EA (IE.7)**, **CPU**中断允许（总允许）位。

二、中断优先级控制

80C51单片机有两个中断优先级，即可实现二级中断服务嵌套。每个中断源的中断优先级都是由中断优先级寄存器**IP**中的相应位的状态来规定的。

位	7	6	5	4	3	2	1	0	
字节地址: B8H			PT2	PS	PT1	PX1	PT0	PX0	IP

- **PX0** (**IP.0**)，外部中断**0**优先级设定位；
- **PT0** (**IP.1**)，定时/计数器**T0**优先级设定位；
- **PX1** (**IP.2**)，外部中断**0**优先级设定位；
- **PT1** (**IP.3**)，定时/计数器**T1**优先级设定位；
- **PS** (**IP.4**)，串行口优先级设定位；
- **PT2** (**IP.5**)，定时/计数器**T2**优先级设定位。

而**80C52**单片机有四个中断优先级，即可实现四级中断服务嵌套。每个中断源的中断优先级由中断优先级寄存器**IP**和**IPH**中的相应位的状态来规定的。

位	7	6	5	4	3	2	1	0	
字节地址: B7H			PT2	PS	PT1	PX1	PT0	PX0	IPH

- **PX0** (**IPH.0**)，外部中断**0**优先级设定位；
- **PT0** (**IPH.1**)，定时/计数器**T0**优先级设定位；
- **PX1** (**IPH.2**)，外部中断**0**优先级设定位；
- **PT1** (**IPH.3**)，定时/计数器**T1**优先级设定位；
- **PS** (**IPH.4**)，串行口优先级设定位；
- **PT2** (**IPH.5**)，定时/计数器**T2**优先级设定位。

同一优先级中的中断申请不止一个时，则有中断优先权排队问题。同一优先级的中断优先权排队，由中断系统硬件确定的自然优先级形成，其排列如图所示：

各中断源响应优先级及中断服务程序入口表

中断源	中断标志	中断服务程序入口	优先级顺序
外部中断 0 ($\overline{\text{INT0}}$)	IE0	0003H	高
定时/计数器 0 (T0)	TF0	000BH	↓
外部中断 1 ($\overline{\text{INT1}}$)	IE1	0013H	↓
定时/计数器 1 (T1)	TF1	001BH	↓
串行口	RI 或 TI	0023H	低

- 设置52单片机的4个中断源,使他们的优顺序为T1,INT1,INT0,T0.

IPH = 0X08; PT1 = 1;

IP = 0X40; PX1 = 1;

80C51单片机的中断优先级有三条原则：

- CPU同时接收到几个中断时，首先响应优先级别最高的中断请求。
- 正在进行的中断过程不能被新的同级或低优先级的中断请求所中断。
- 正在进行的低优先级中断服务，能被高优先级中断请求所中断。

为了实现上述后两条原则，中断系统内部设有两个用户不能寻址的优先级状态触发器。其中一个置1，表示正在响应高优先级的中断，它将阻断后来所有的中断请求；另一个置1，表示正在响应低优先级中断，它将阻断后来所有的低优先级中断请求。

3.2 80C51单片机中断处理过程

3.2.1 中断响应条件和时间 中断响应条件

- 中断源有中断请求；
- 此中断源的中断允许位为1；
- **CPU**开中断（即**EA=1**）。

以上三条同时满足时，**CPU**才有可能响应中断。



3.3 80C51的定时/计数器

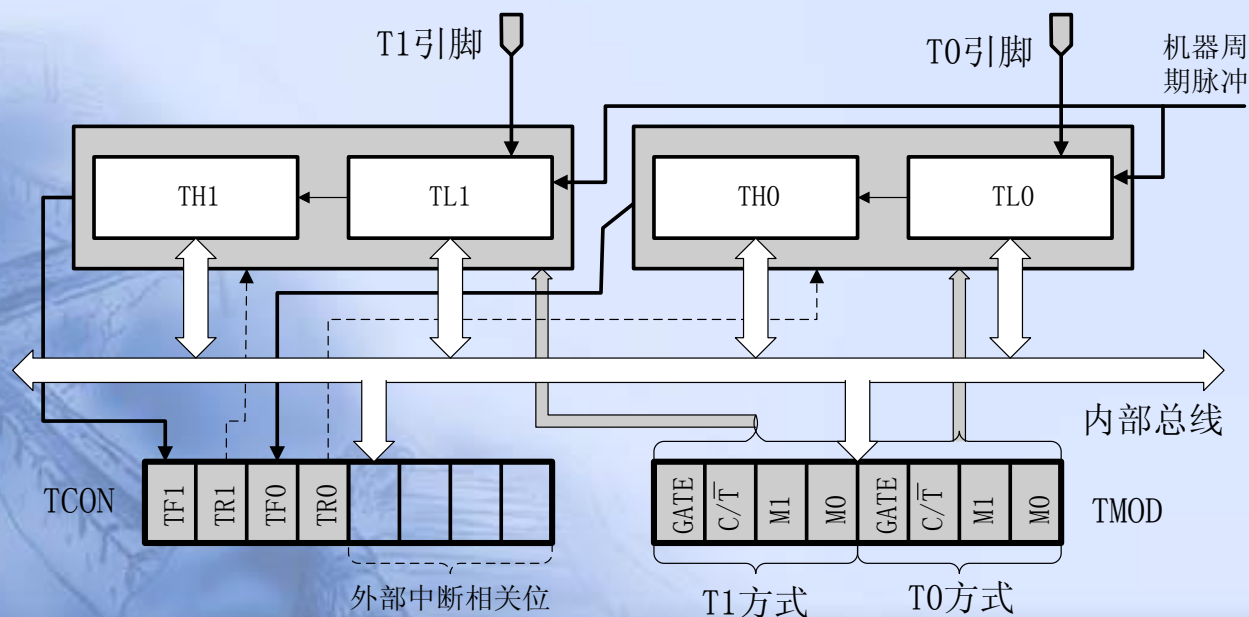
实现定时功能，**比较方便的办法是利用单片机内部的定时/计数器**。也可以采用下面三种方法：

- **软件定时**：软件定时不占用硬件资源，但占用了**CPU**时间，降低了**CPU**的利用率。
- **采用时基电路定时**：例如采用**555**电路，外接必要的元器件（电阻和电容），即可构成硬件定时电路。但在硬件连接好以后，定时值与定时范围不能由软件进行控制和修改，即不可编程。
- **采用可编程芯片定时**：这种定时芯片的定时值及定时范围很容易用软件来确定和修改，此种芯片定时功能强，使用灵活。在单片机的定时/计数器不够用时，可以考虑进行扩展。

3.3.1 定时/计数器的结构和工作原理

一、定时/计数器的结构

定时/计数器的实质是加1计数器（**16位**），由高**8位**和低**8位**两个寄存器组成。**TMOD**是定时/计数器的工作方式寄存器，确定工作方式和功能；**TCON**是控制寄存器，控制**T0**、**T1**的启动和停止及设置溢出标志。



二、定时/计数器的工作原理

加1计数器输入的计数脉冲有两个来源,一个是由系统的时钟振荡器输出脉冲经**12**分频后送来;一个是**T0**或**T1**引脚输入的外部脉冲源。每来一个脉冲计数器加**1**,当加到计数器为全**1**时,再输入一个脉冲就使计数器回零,且计数器的溢出使**TCON**中**TF0**或**TF1**置**1**,向**CPU**发出中断请求(定时/计数器中断允许时)。如果定时/计数器工作于定时模式,则表示定时时间已到;如果工作于计数模式,则表示计数值已满。

可见,由溢出时计数器的值减去计数初值才是加**1**计数器的计数值。

■ **设置为定时器模式时**，加1计数器是对内部机器周期计数（1个机器周期等于12个振荡周期，即计数频率为晶振频率的1/12）。**计数值N乘以机器周期Tcy就是定时时间t。**

■ **设置为计数器模式时**，外部事件计数脉冲由T0或T1引脚输入到计数器。在每个机器周期的S5P2期间采样T0、T1引脚电平。当某周期采样到一高电平输入，而下一周期又采样到一低电平时，则计数器加1，更新的计数值在下一个机器周期的S3P1期间装入计数器。由于检测一个从1到0的下降沿需要2个机器周期，因此要求被采样的电平至少要维持一个机器周期。当晶振频率为**12MHz**时，最高计数频率不超过1/2MHz，即计数脉冲的周期要大于**2 μs**。

3.3.2 定时/计数器的控制

80C51单片机定时/计数器的工作由两个特殊功能寄存器控制。**TMOD**用于设置其工作方式；**TCON**用于控制其启动和中断申请。

一、工作方式寄存器**TMOD**

工作方式寄存器**TMOD**用于设置定时/计数器的工作方式，低四位用于**T0**，高四位用于**T1**。其格式如下：

位	7	6	5	4	3	2	1	0	
字节地址：89H	GATE	C/ \bar{T}	M1	M0	GATE	C/ \bar{T}	M1	M0	TMOD

GATE: 门控位。**GATE=0**时，只要用软件使**TCON**中的**TR0**或**TR1**为**1**，就可以启动定时/计数器工作；**GATE=1**时，要用软件使**TR0**或**TR1**为**1**，同时外部中断引脚或也为高电平时，才能启动定时/计数器工作。即此时定时器的启动多了一条条件。

C/T :定时/计数模式选择位。**C/T=0**为定时模式；**C/T=1**为计数模式。

M1M0: 工作方式设置位。定时/计数器有四种工作方式，由**M1M0**进行设置。

定时/计数器工作方式设置表		
M1M0	工作方式	说 明
00	方式 0	13 位定时/计数器
01	方式 1	16 位定时/计数器
10	方式 2	8 位自动重装定时/计数器
11	方式 3	T0 分成两个独立的 8 位定时/计数器；T1 此方式停止计数

二、控制寄存器TCON

TCON的低4位用于控制外部中断,已在前面介绍。**TCON**的高4位用于控制定时/计数器的启动和中断申请。其格式如下:

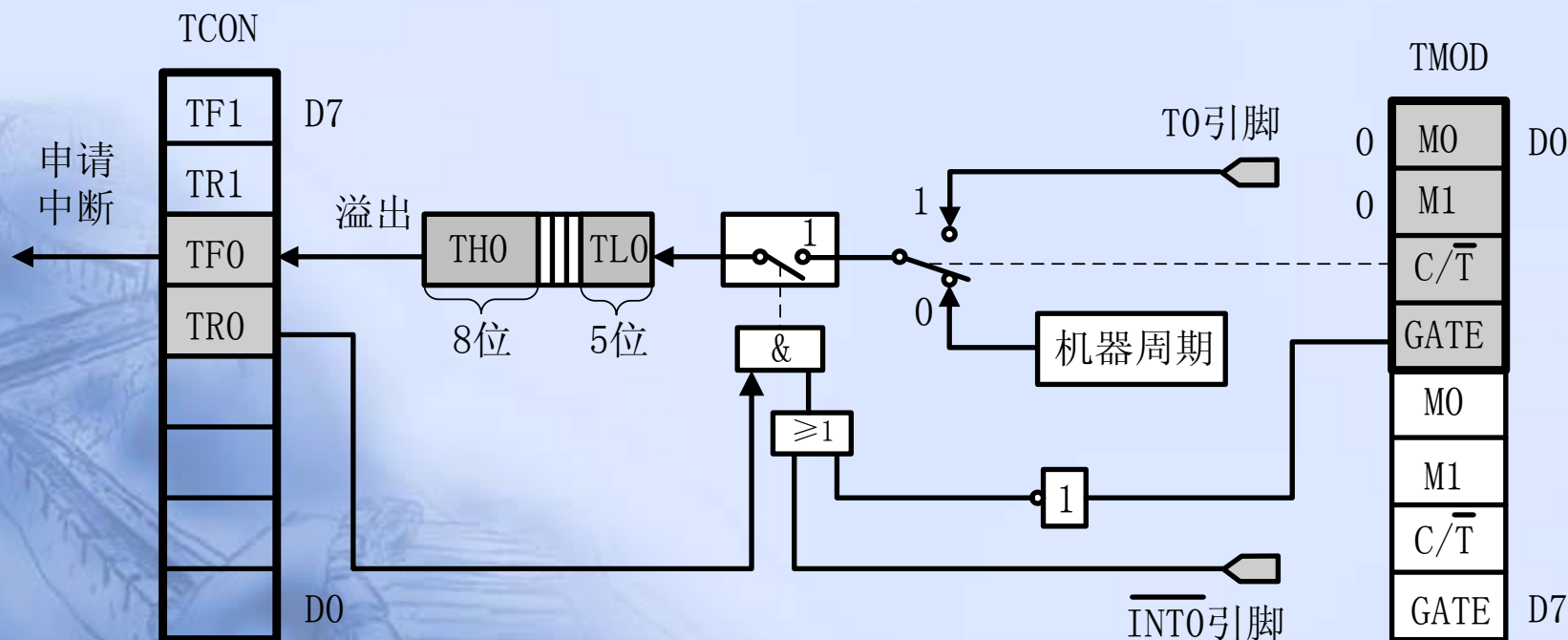
位	7	6	5	4	3	2	1	0	
字节地址: 88H	TF1	TR1	TF0	TR0					TCON

- **TF1 (TCON.7)**: T1溢出中断请求标志位。T1计数溢出时由硬件自动置TF1为1。CPU响应中断后TF1由硬件自动清0。T1工作时, CPU可随时查询TF1的状态。所以, TF1可用作查询测试的标志。TF1也可以用软件置1或清0, 同硬件置1或清0的效果一样。
- **TR1 (TCON.6)**: T1运行控制位。TR1置1时, T1开始工作; TR1置0时, T1停止工作。TR1由软件置1或清0。所以, 用软件可控制定时/计数器的启动与停止。
- **TF0 (TCON.5)**: T0溢出中断请求标志位, 其功能与TF1类同。
- **TR0 (TCON.4)**: T0运行控制位, 其功能与TR1类同。

3.3.3 定时/计数器的工作方式

一、方式0

方式0为13位计数，由**TL0**的低5位（高3位未用）和**TH0**的8位组成。**TL0**的低5位溢出时向**TH0**进位，**TH0**溢出时，置位**TCON**中的**TF0**标志，向**CPU**发出中断请求。



定时器模式时有： $N = t / T_{cy}$

计数初值计算的公式为： $X = 2^{13} - N$

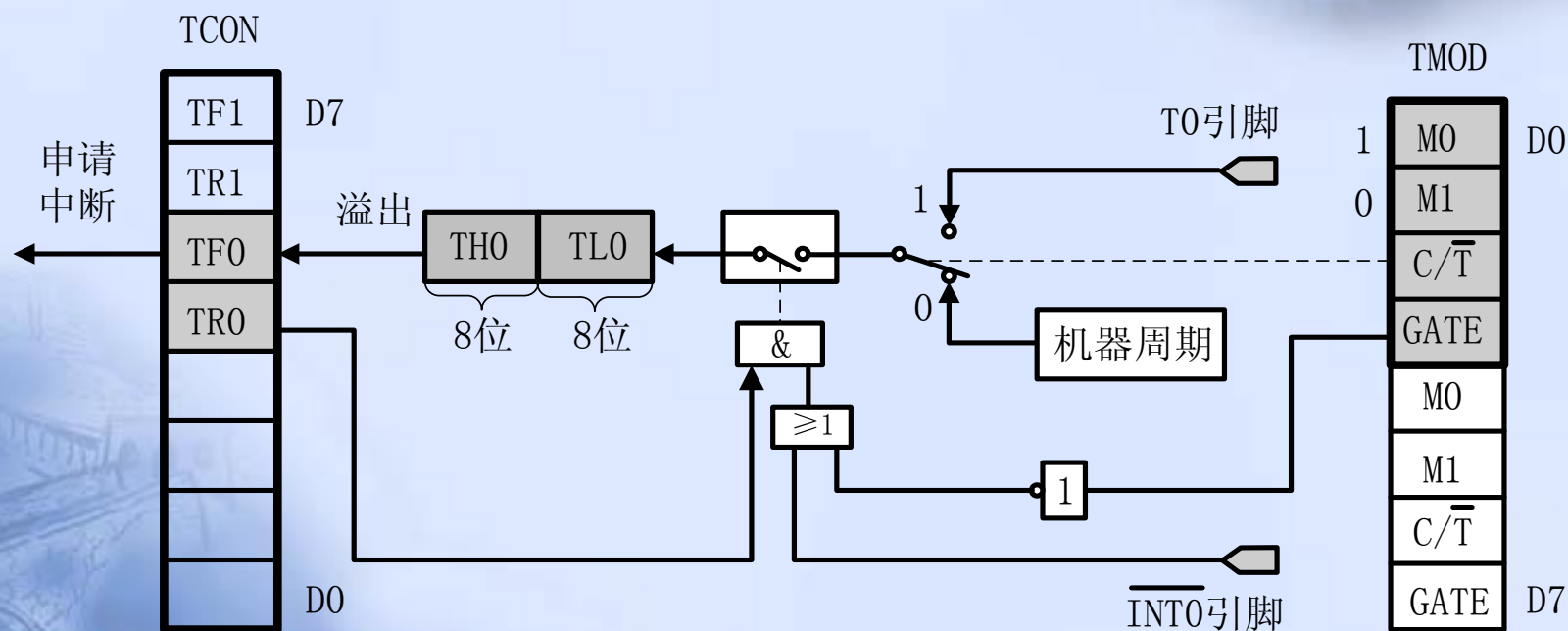
定时器的初值还可以采用计数个数直接取补法获得。

计数模式时，计数脉冲是**T0**引脚上的外部脉冲。

门控位**GATE**具有特殊的作用。当**GATE=0**时，经反相后使或门输出为**1**，此时仅由**TR0**控制与门的开启，与门输出**1**时，控制开关接通，计数开始；**当GATE=1时**，由外中断引脚信号控制或门的输出，此时控制**与门的开启由外中断引脚信号和TR0共同控制**。当**TR0=1**时，外中断引脚信号引脚的**高电平启动计数**，外中断引脚信号引脚的**低电平停止计数**。这种方式常用来测量外中断引脚上正脉冲的宽度。

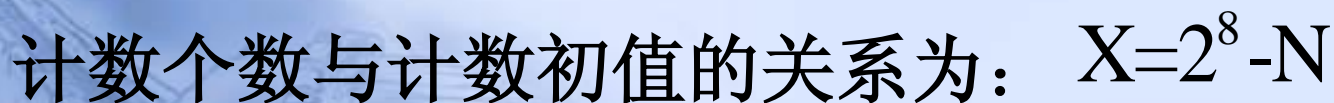
二、方式1

方式1的计数位数是**16位**，由**TL0**作为低8位、**TH0**作为高8位，组成了**16位加1计数器**。



计数个数与计数初值的关系为： $X=2^{16}-N$

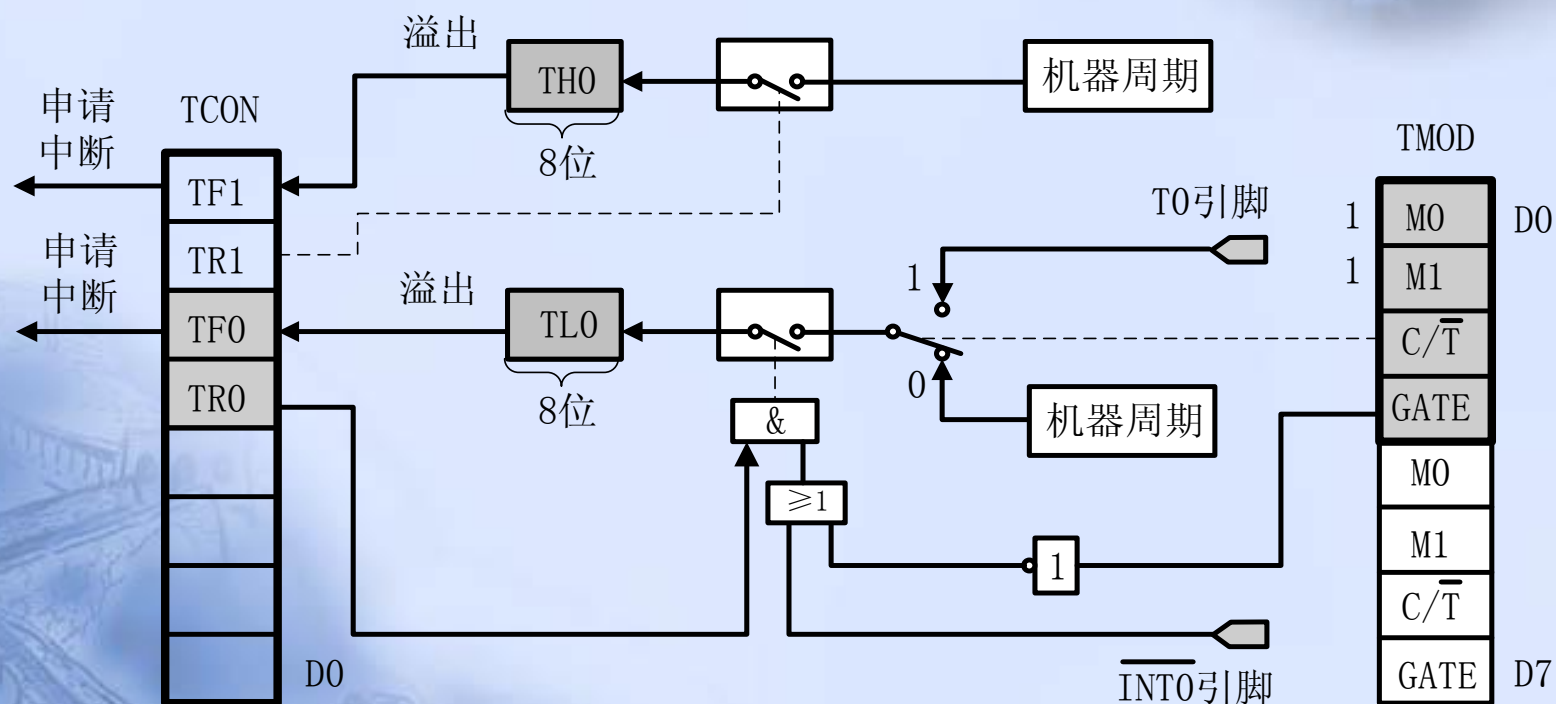
方式2为自动重装初值的8位计数方式。



工作方式2特别适合于用作较精确的脉冲信号发生器。

四、方式3

方式3只适用于定时/计数器T0，定时器T1处于方式3时相当于TR1=0，停止计数。



工作方式3将T0分成为两个独立的8位计数器TL0和TH0。

3.3.4 定时/计数器应用举例

初始化程序应完成如下工作：

- 对**TMOD**赋值，以确定**T0**和**T1**的工作方式。
- 计算初值,并将其写入**TH0**、**TL0**或**TH1**、**TL1**。
- 中断方式时，则对**IE**赋值，开放中断。
- 使**TR0**或**TR1**置位，启动定时/计数器定时或计数。

例 利用定时/计数器T1的方式1，产生10ms的定时，并使P1.0引脚上输出周期为20ms的方波，采用中断方式，设系统时钟频率为12 MHz。

解：1、计算计数初值X：

由于晶振为12 MHz，所以机器周期Tcy为1 μ s。

所以：

$$N = t / T_{cy} = 10000 / 1 = 10000$$

$$X = 65536 - 10000 = 55536 = D8F0H$$

即应将D8H送入TH0中，F0H送入TL0中

2、求T1的方式控制字TMOD：

M1M0=01，GATE=0，C/T=0，可取方式控制字为01H；



1、利用定时/计数器**T0**从**P1.0**输出周期为**1s**的方波，让发光二极管以**1HZ**闪烁，设晶振频率为**12MHz**。。

2、利用定时/计数器**T1**产生定时时钟,由**P1**口控制**8**个发光二极管,使**8**个指示灯依次一个一个闪动，闪动频率为**10次/秒**(**8**个灯依次亮一遍为一个周期)，循环。

3、同时用两个定时器控制蜂鸣器发声，定时器**0**控制频率，定时器**1**控制同个频率持续的时间，间隔**300ms**依次输出**1, 10, 50,100, 200,400,800, 1k (hz)**的方波。

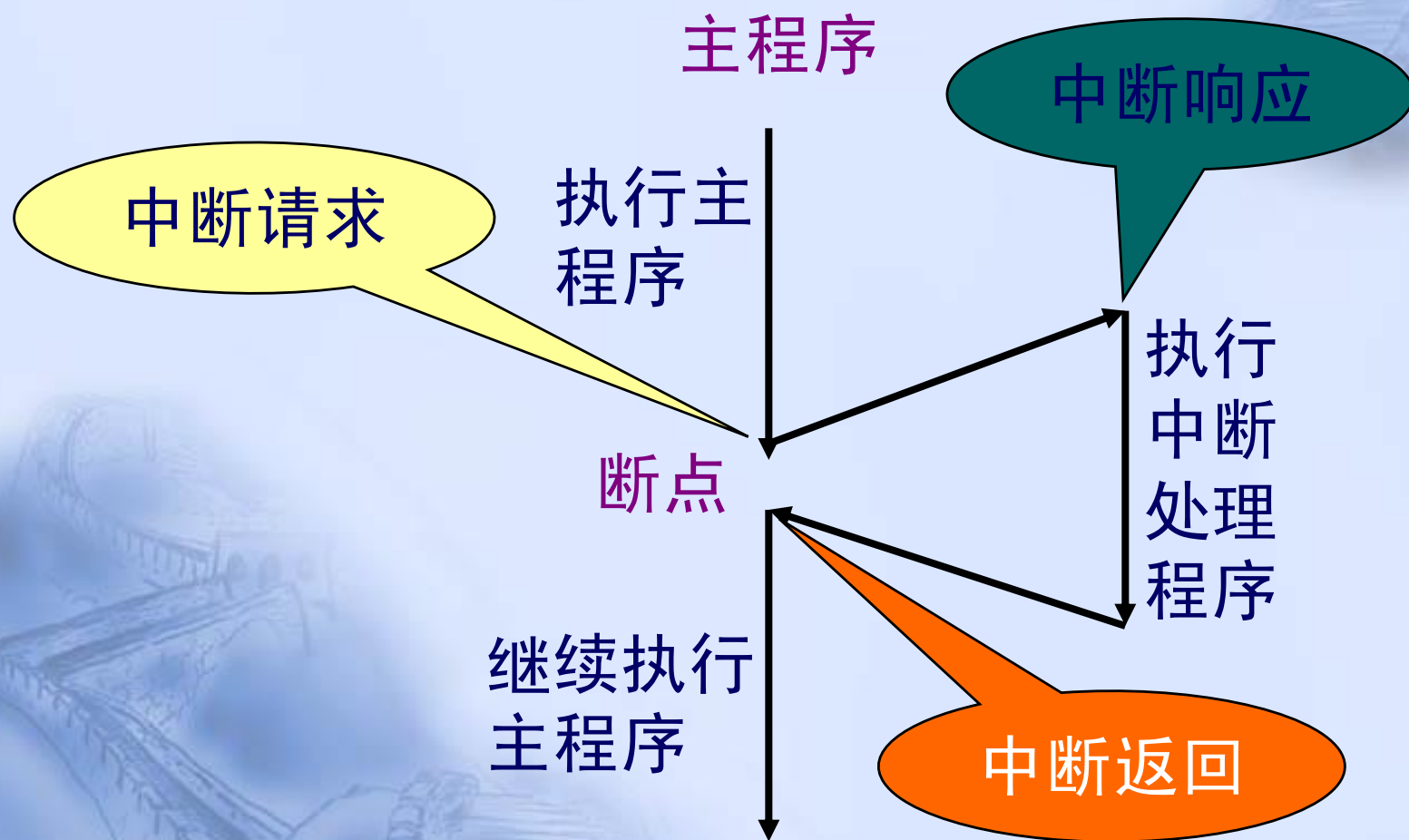
4、用定时器以间隔**500MS**在**6**位数码管上依次显示**0、1、2、3....C、D、E、F**，重复。



动态显示

动态显示的特点是将所有位数数码管的段选线并联在一起，由位选线控制是哪一位数码管有效。这样一来，就没有必要每一位数码管配一个锁存器，从而大大地简化了硬件电路。选亮数码管采用动态扫描显示。所谓动态扫描显示即轮流向各位数码管送出字形码和相应的位选，利用发光管的余辉和人眼视觉暂留作用，使人的感觉好像各位数码管同时都在显示。动态显示的亮度比静态显示要差一些，所以在选择限流电阻时应略小于静态显示电路中的。

MCS-51单片机的中断系统结构



4.1 80C51中断的控制

一、中断允许控制

CPU对中断系统所有中断以及某个中断源的开放和屏蔽是由中断允许寄存器**IE**控制的。

位	7	6	5	4	3	2	1	0	
字节地址: A8H	EA			ES	ET1	EX1	ET0	EX0	IE

- **EX0(IE.0)**, 外部中断**0**允许位;
- **ET0(IE.1)**, 定时/计数器**T0**中断允许位;
- **EX1(IE.2)**, 外部中断**0**允许位;
- **ET1(IE.3)**, 定时/计数器**T1**中断允许位;
- **ES (IE.4)**, 串行口中断允许位;
- **EA (IE.7)**, **CPU**中断允许 (总允许) 位。

同一优先级中的中断申请不止一个时，则有中断优先权排队问题。同一优先级的中断优先权排队，由中断系统硬件确定的自然优先级形成，其排列如图所示：

各中断源响应优先级及中断服务程序入口表

中断源	中断标志	中断服务程序入口	优先级顺序
外部中断 0 ($\overline{\text{INT0}}$)	IE0	0003H	高
定时/计数器 0 (T0)	TF0	000BH	↓
外部中断 1 ($\overline{\text{INT1}}$)	IE1	0013H	↓
定时/计数器 1 (T1)	TF1	001BH	↓
串行口	RI 或 TI	0023H	低

4.2 80C51单片机中断处理过程

4.2.1 中断响应条件和时间 中断响应条件

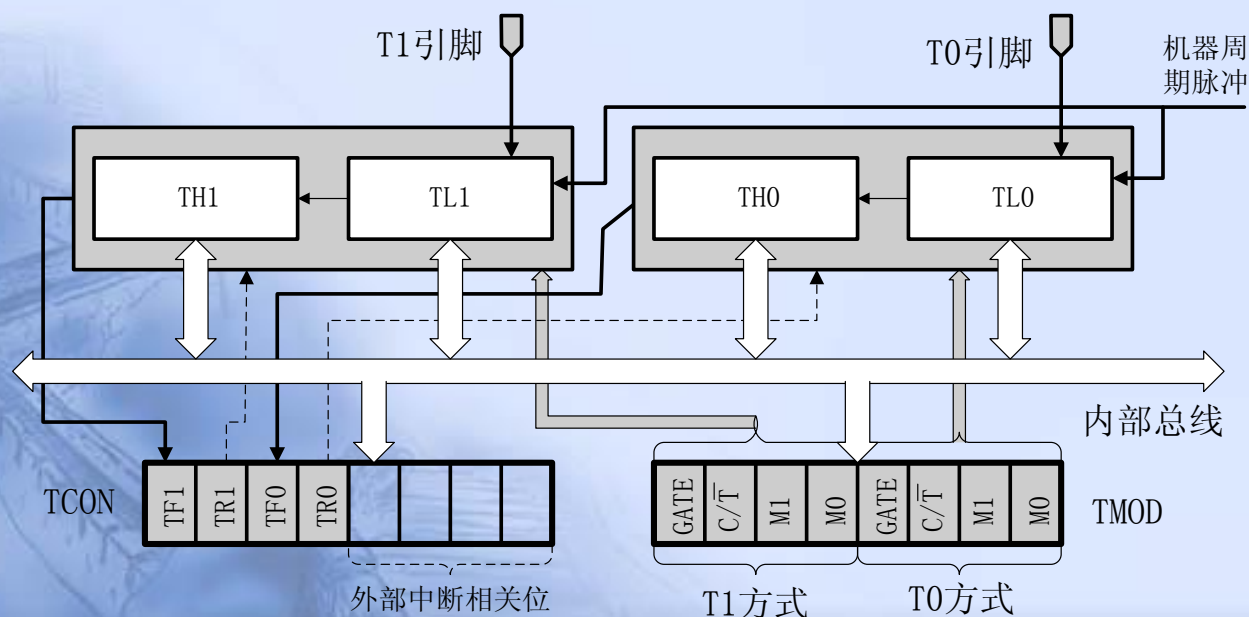
- 中断源有中断请求；
- 此中断源的中断允许位为1；
- **CPU**开中断（即**EA=1**）。

以上三条同时满足时，**CPU**才有可能响应中断。

定时/计数器的结构和工作原理

一、定时/计数器的结构

定时/计数器的实质是加1计数器（**16位**），由高**8位**和低**8位**两个寄存器组成。**TMOD**是定时/计数器的工作方式寄存器，确定工作方式和功能；**TCON**是控制寄存器，控制**T0**、**T1**的启动和停止及设置溢出标志。



4.2.2 定时/计数器的控制

80C51单片机定时/计数器的工作由两个特殊功能寄存器控制。**TMOD**用于设置其工作方式；**TCON**用于控制其启动和中断申请。

一、工作方式寄存器**TMOD**

工作方式寄存器**TMOD**用于设置定时/计数器的工作方式，低四位用于**T0**，高四位用于**T1**。其格式如下：

位	7	6	5	4	3	2	1	0	
字节地址：89H	GATE	C/ \bar{T}	M1	M0	GATE	C/ \bar{T}	M1	M0	TMOD

GATE: 门控位。**GATE=0**时，只要用软件使**TCON**中的**TR0**或**TR1**为**1**，就可以启动定时/计数器工作；**GATE=1**时，要用软件使**TR0**或**TR1**为**1**，同时外部中断引脚**INT0/1**也为高电平时，才能启动定时/计数器工作。即此时定时器的启动条件，加上了**INT0/1**引脚为高电平这一条件。

C/T :定时/计数模式选择位。**C/T=0**为定时模式；**C/T=1**为计数模式。

M1M0: 工作方式设置位。定时/计数器有四种工作方式，由**M1M0**进行设置。

定时/计数器工作方式设置表		
M1M0	工作方式	说明
00	方式 0	13 位定时/计数器
01	方式 1	16 位定时/计数器
10	方式 2	8 位自动重装定时/计数器
11	方式 3	T0 分成两个独立的 8 位定时/计数器；T1 此方式停止计数

二、控制寄存器TCON

TCON的低4位用于控制外部中断,已在前面介绍。**TCON**的高4位用于控制定时/计数器的启动和中断申请。其格式如下:

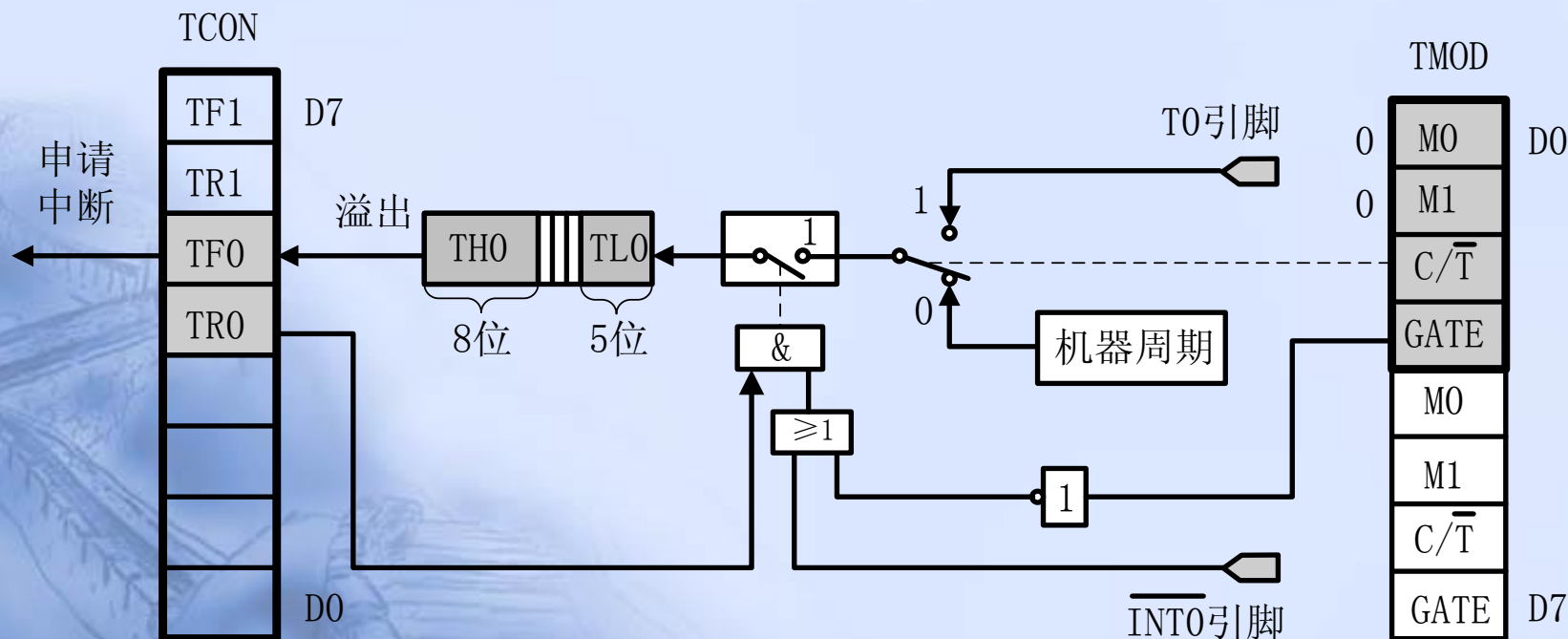
位	7	6	5	4	3	2	1	0	
字节地址: 88H	TF1	TR1	TF0	TR0					TCON

- **TF1 (TCON.7)**: T1溢出中断请求标志位。T1计数溢出时由硬件自动置TF1为1。CPU响应中断后TF1由硬件自动清0。T1工作时, CPU可随时查询TF1的状态。所以, TF1可用作查询测试的标志。TF1也可以用软件置1或清0, 同硬件置1或清0的效果一样。
- **TR1 (TCON.6)**: T1运行控制位。TR1置1时, T1开始工作; TR1置0时, T1停止工作。TR1由软件置1或清0。所以, 用软件可控制定时/计数器的启动与停止。
- **TF0 (TCON.5)**: T0溢出中断请求标志位, 其功能与TF1类同。
- **TR0 (TCON.4)**: T0运行控制位, 其功能与TR1类同。

定时/计数器的工作方式

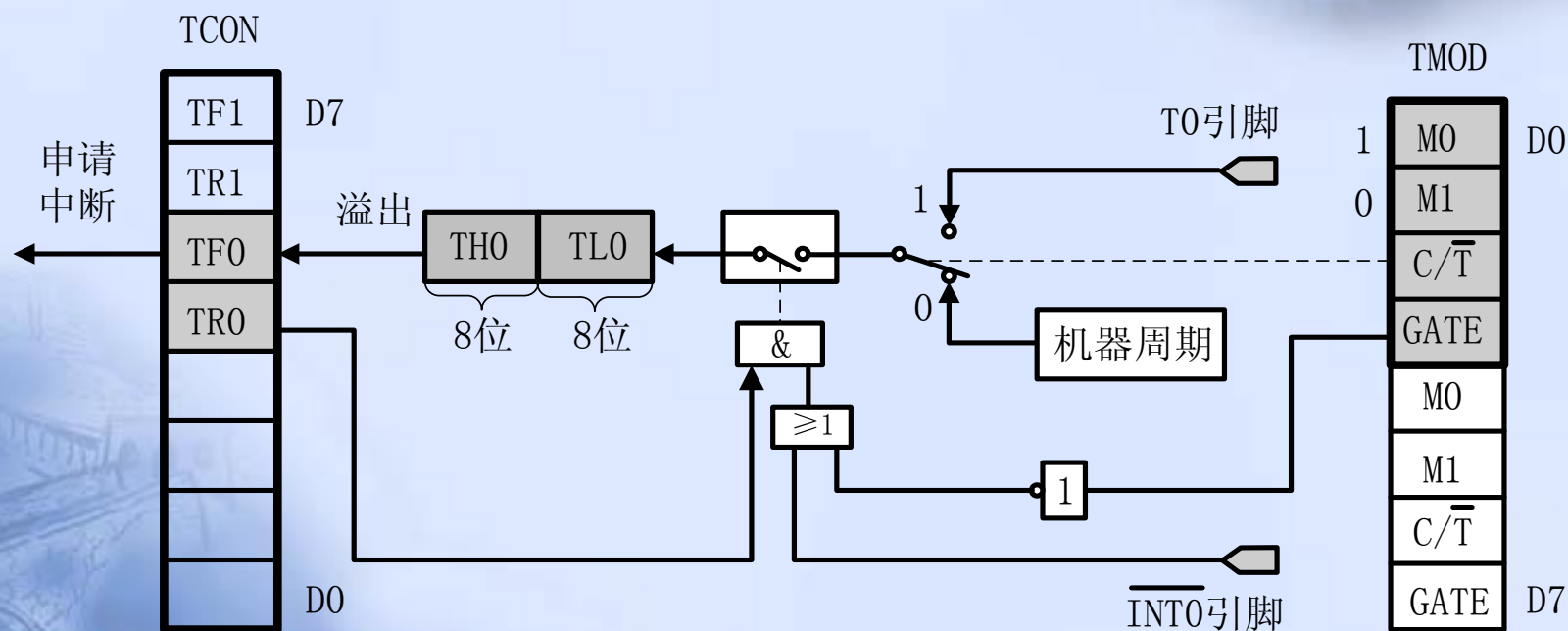
一、方式0

方式0为13位计数，由**TL0**的低5位（高3位未用）和**TH0**的8位组成。**TL0**的低5位溢出时向**TH0**进位，**TH0**溢出时，置位**TCON**中的**TF0**标志，向**CPU**发出中断请求。



二、方式1

方式1的计数位数是**16位**，由**TL0**作为低8位、**TH0**作为高8位，组成了**16位加1计数器**。



计数个数与计数初值的关系为： $X=2^{16}-N$

第四课练习

- 1.利用动态扫描方法在六位数码管上显示出稳定的**654321**.
- 2.用动态扫描方法和定时器1在数码管的前三位显示出秒表，精确到**1%**秒，即后两位显示**1%**秒，一直循环下去。
- 3.利用动态扫描和定时器1在数码管上显示出从**765432**开始以**1/10**秒的速度往下递减直至**765398**并保持显示此数，与此同时利用定时器0以**500MS**速度进行流水灯从上至下移动，当数码管上数减到停止时，实验板上流水灯也停止然后全部开始闪烁，**3**秒后（用**T0**定时）流水灯全部关闭、数码管上显示出“**HELLO**”。到此保持住。

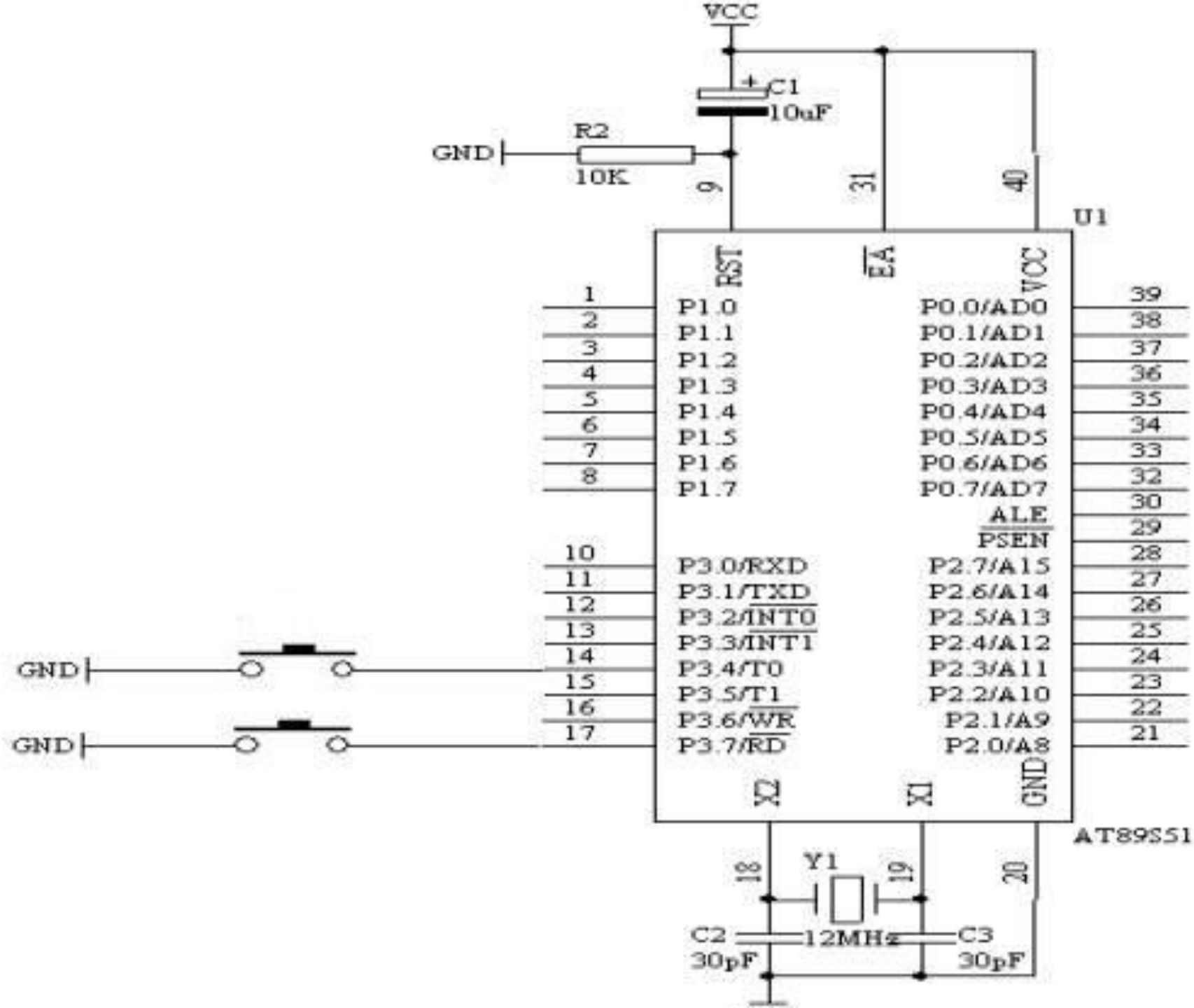


键盘的应用

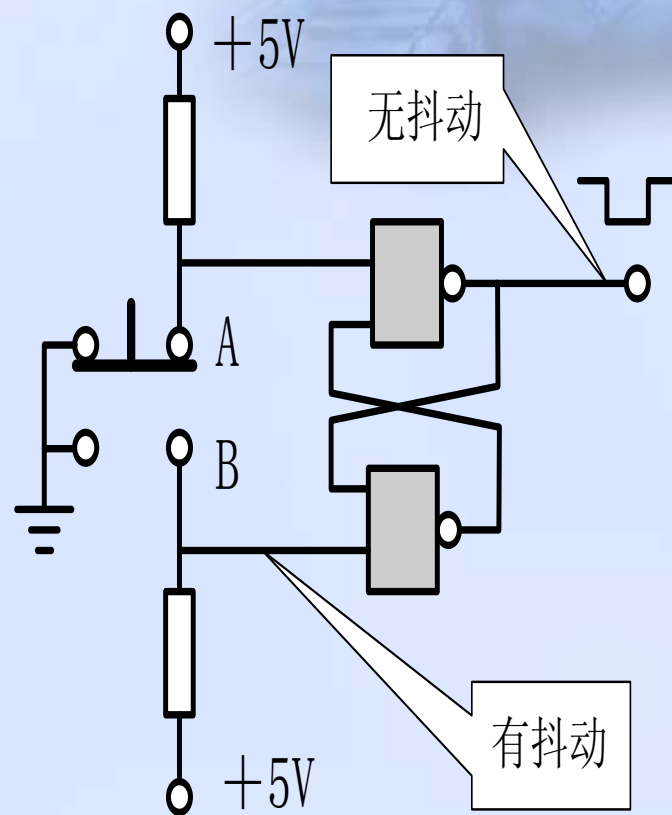
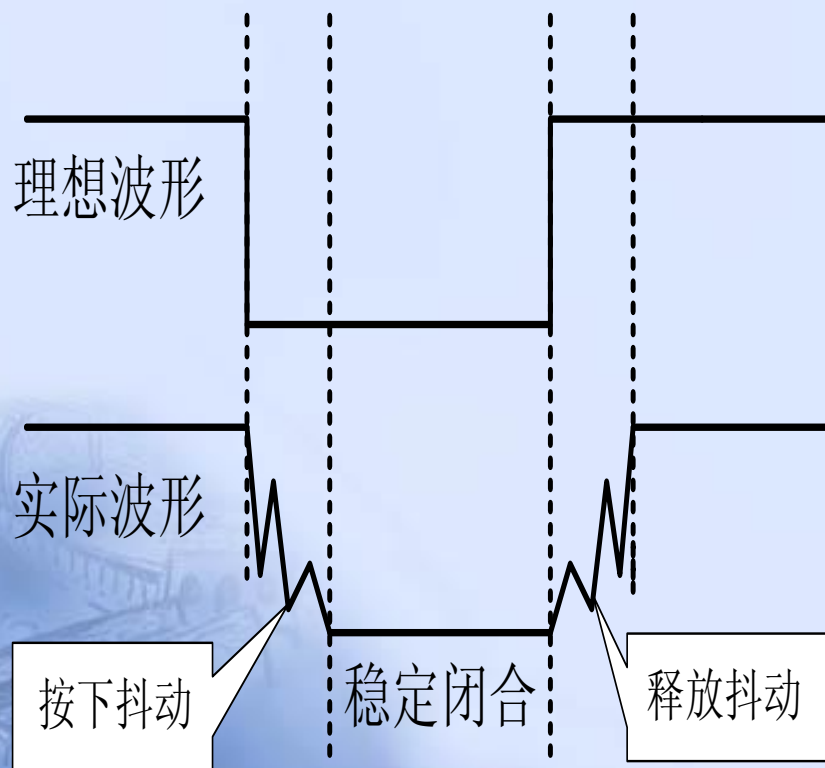
■ 1、键盘的分类

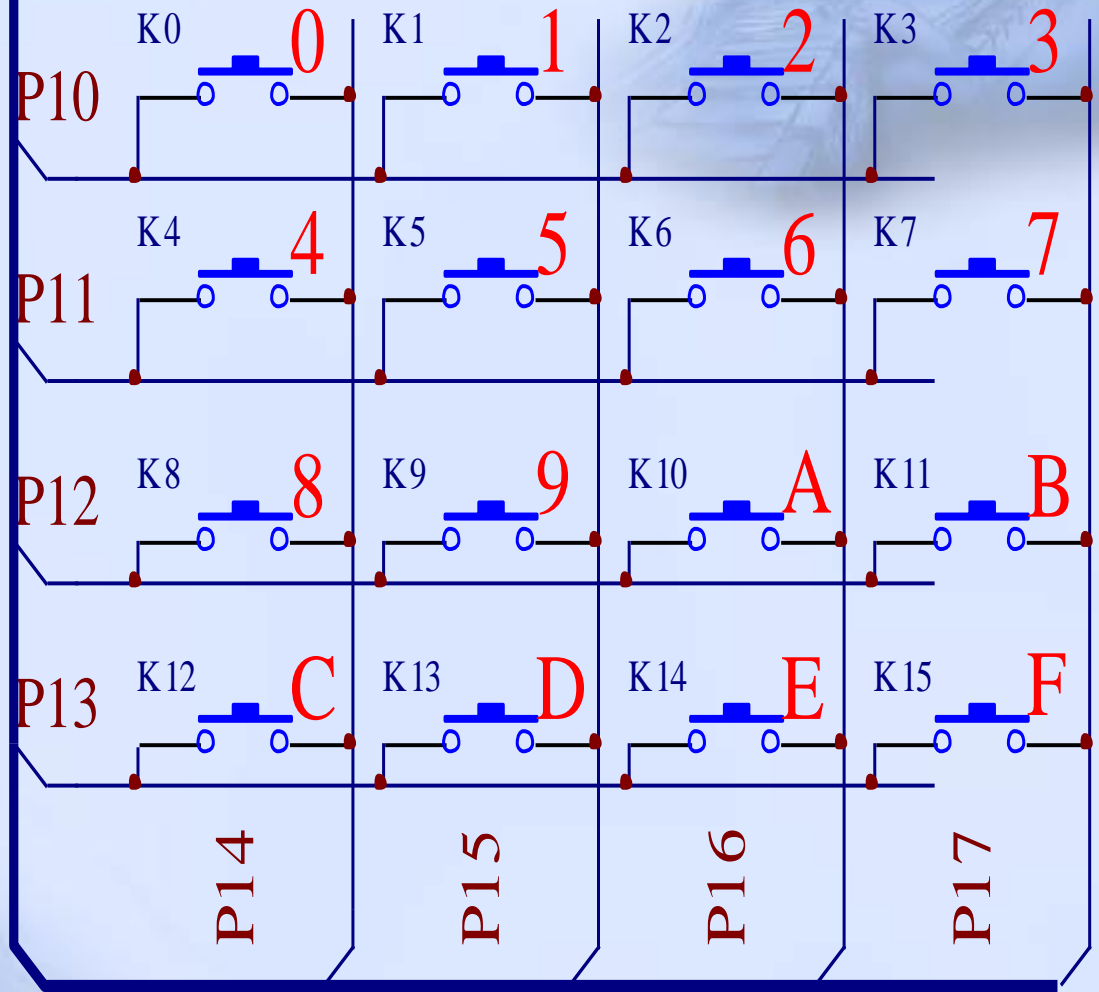
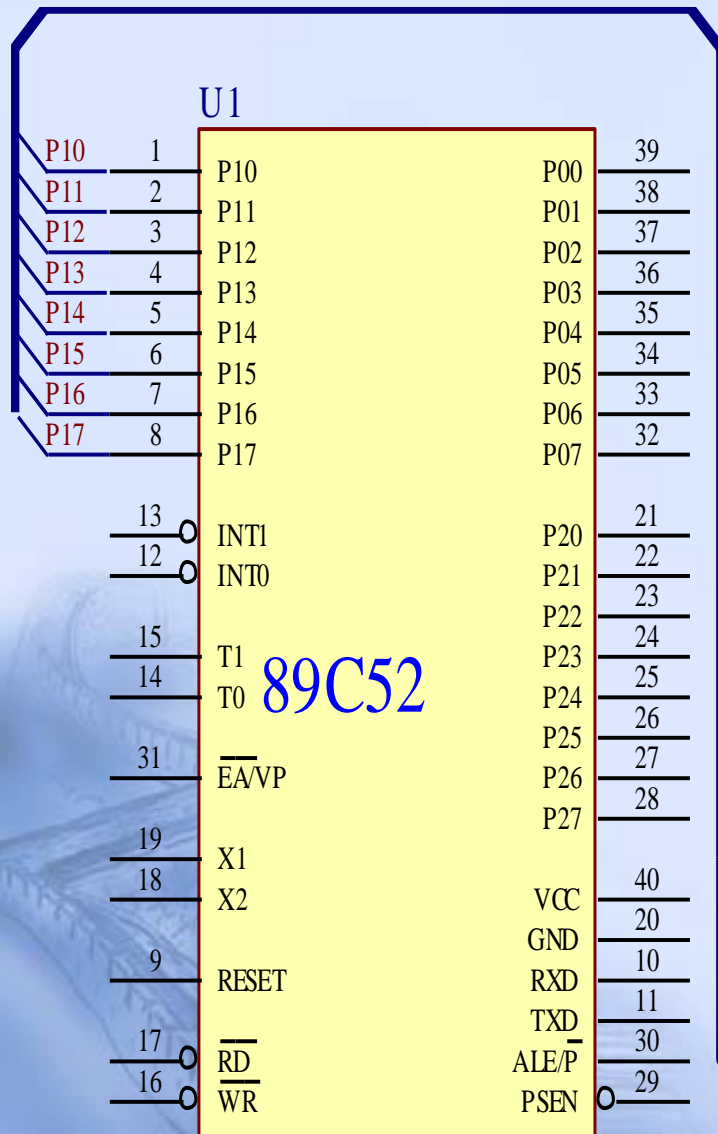
- 键盘分编码键盘和非编码键盘。键盘上闭合键的识别由专用的硬件编码器实现，并产生键编码号或键值的称为编码键盘，如计算机键盘。
- 而靠软件编程来识别的称为非编码键盘；
- 在单片机组成的各种系统中，用的最多的是非编码键盘。也有用到编码键盘的。

- 非编码键盘有分为：独立键盘和行列式（又称为矩阵式）键盘。



按键在闭合和断开时，触点会存在抖动现象：





P3.0

P3.1

P3.2

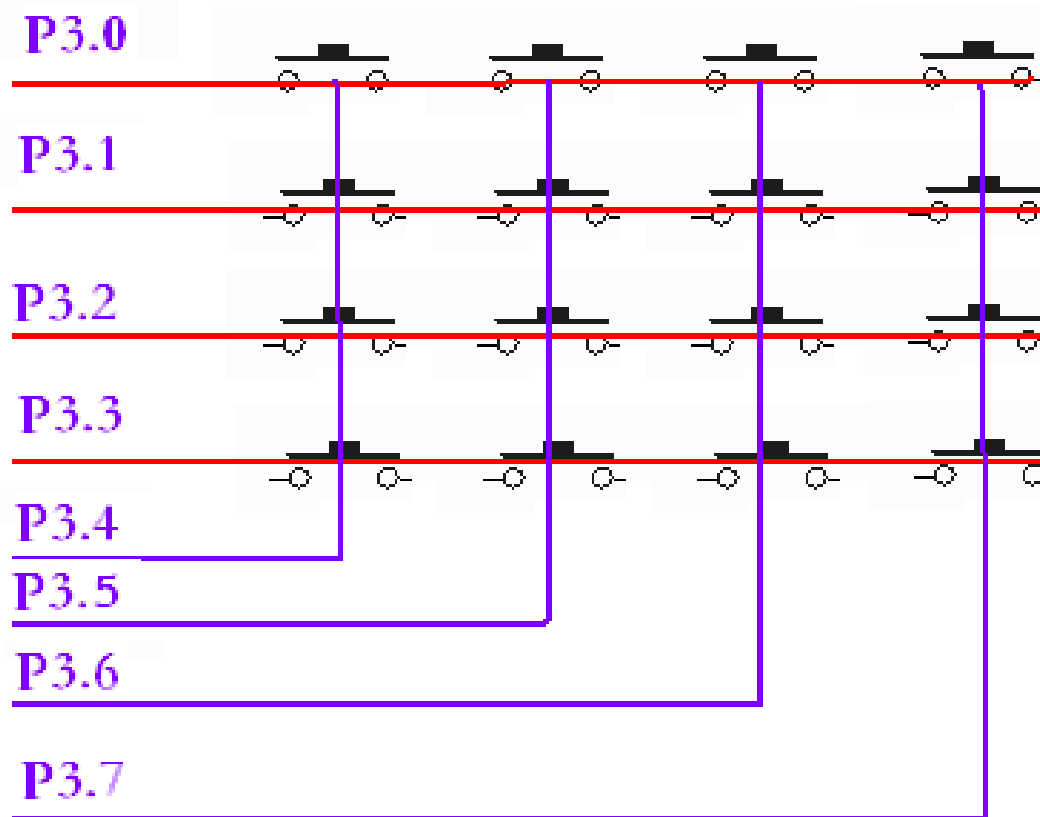
P3.3

P3.4

P3.5

P3.6

P3.7



- 数码管前三位显示一个跑表，从000到999之间以1%秒速度运行，当按下一个独立键盘时跑表停止，松开手后跑表继续运行。(用定时器设计表)。
- 在上题的基础上，用另外三个独立键盘实现按下第一个时计时停止，按下第二个时计时开始，按下第三个是计数值清零从头开始。
- 按下16个矩阵键盘依次在数码管上显示1-16的平方。如按下第一个显示1，第二个显示4...

第六讲

D/A & A/D转换器及其与单片机接口



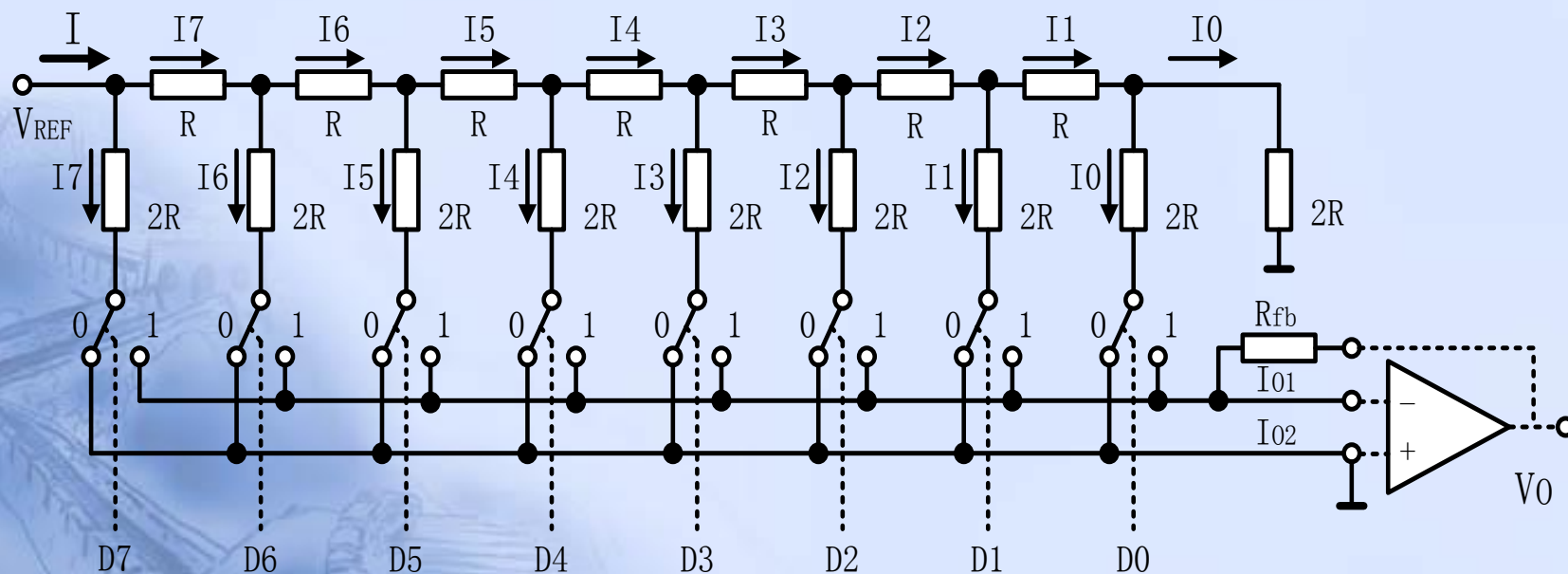
D/A

6.1 D/A转换器及其与单片机接口

6.1.1 D/A转换器的原理及主要技术指标

一、D/A转换器的基本原理及分类

T型电阻网络D/A转换器：



$$I = V_{REF}/R$$

$$I_7 = I/2^1, I_6 = I/2^2, I_5 = I/2^3, I_4 = I/2^4, I_3 = I/2^5, I_2 = I/2^6, I_1 = I/2^7, I_0 = I/2^8$$

当输入数据 D7~D0 为 1111 1111B 时，有：

$$I_{O1} = I_7 + I_6 + I_5 + I_4 + I_3 + I_2 + I_1 + I_0 = (I/2^8) \times (2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0)$$

$$I_{O2} = 0$$

若 $R_{fb} = R$ ，则：

$$V_O = -I_{O1} \times R_{fb}$$

$$= -I_{O1} \times R$$

$$= - \left((V_{REF}/R) / 2^8 \right) \times (2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0) \times R$$

$$= - (V_{REF} / 2^8) \times (2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0)$$

输出电压 的大小与数字量具有对应的关系。

二、D/A转换器的主要性能指标

1、分辨率

分辨率是指输入数字量的最低有效位（LSB）发生变化时，所对应的输出模拟量（电压或电流）的变化量。它反映了输出模拟量的最小变化值。

分辨率与输入数字量的位数有确定的关系，可以表示成 $FS / 2^n$ 。FS表示满量程输入值，n为二进制位数。对于5V的满量程，采用8位的DAC时，分辨率为 $5V/256=19.5mV$ ；当采用12位的DAC时，分辨率则为 $5V/4096=1.22mV$ 。显然，位数越多分辨率就越高。

2、线性度

线性度（也称非线性误差）是实际转换特性曲线与理想直线特性之间的最大偏差。常以相对于满量程的百分数表示。如 $\pm 1\%$ 是指实际输出值与理论值之差在满刻度的 $\pm 1\%$ 以内。

3、绝对精度和相对精度

- **绝对精度（简称精度）** 是指在整个刻度范围内，任一输入数码所对应的模拟量实际输出值与理论值之间的最大误差。绝对精度是由**DAC**的增益误差（当输入数码为全1时，实际输出值与理想输出值之差）、零点误差（数码输入为全0时，**DAC**的非零输出值）、非线性误差和噪声等引起的。绝对精度（即最大误差）应小于**1个LSB**。
- **相对精度**与绝对精度表示同一含义，用最大误差相对于满刻度的百分比表示。

4、建立时间

■ **建立时间**是指输入的数字量发生满刻度变化时，输出模拟信号达到满刻度值的 $\pm 1/2\text{LSB}$ 所需的时间。是描述**D/A**转换速率的一个动态指标。

电流输出型DAC的建立时间短。电压输出型**DAC**的建立时间主要决定于运算放大器的响应时间。根据建立时间的长短，可以将**DAC**分成超高速（ $< 1\mu\text{S}$ ）、高速（ $10\sim 1\mu\text{S}$ ）、中速（ $100\sim 10\mu\text{S}$ ）、低速（ $\geq 100\mu\text{S}$ ）几档。

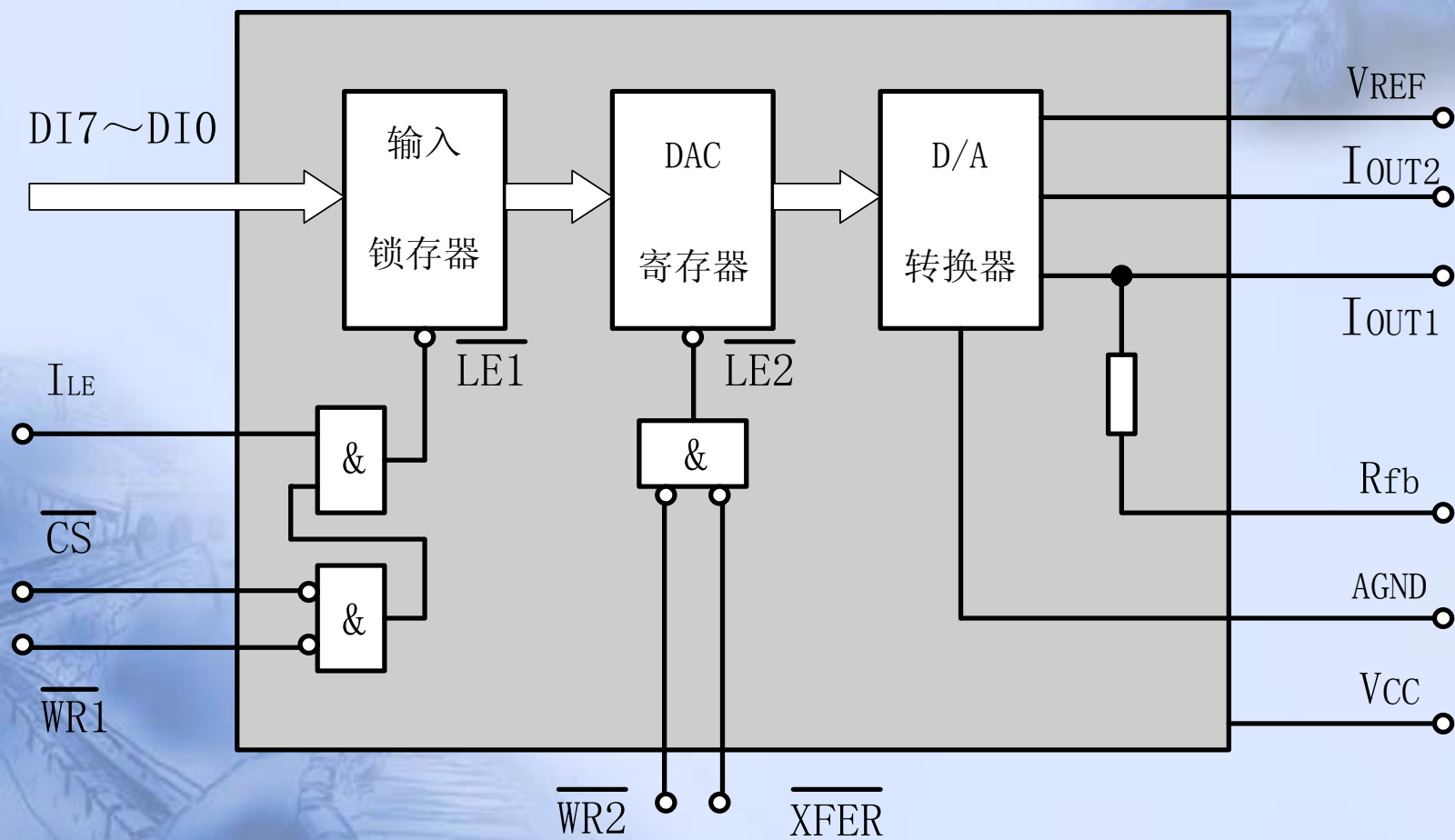
应当注意，精度和分辨率具有一定的联系，但概念不同。**DAC**的位数多时，分辨率会提高，对应于影响精度的量化误差会减小。但其它误差（如温度漂移、线性不良等）的影响仍会使**DAC**的精度变差。

6.1.2 DAC0832芯片及其与单片机接口

DAC0832是使用非常普遍的 8 位D/A转换器，由于其片内有输入数据寄存器，故可以直接与单片机接口。DAC0832以电流形式输出，当需要转换为电压输出时，可外接运算放大器。属于该系列的芯片还有DAC0830、DAC0831，它们可以相互代换。**DAC0832主要特性：**

- 分辨率 8 位；
- 电流建立时间 $1\ \mu\text{S}$ ；
- 数据输入可采用双缓冲、单缓冲或直通方式；
- 输出电流线性度可在满量程下调节；
- 逻辑电平输入与TTL电平兼容；
- 单一电源供电（ $+5\text{V}\sim+15\text{V}$ ）；
- 低功耗， 20mW 。

一、DAC0832内部结构及引脚



(1) \overline{CS} : 片选信号, 输入低电平有效。与 I_{LE} 相配合, 可对写信号 $\overline{WR1}$ 是否有效起到控制作用。

(2) I_{LE} : 允许锁存信号, 输入高电平有效。输入锁存器的信号 $\overline{LE1}$ 由 I_{LE} 、 \overline{CS} 、 $\overline{WR1}$ 的逻辑组合产生。当 I_{LE} 为高电平, \overline{CS} 为低电平, $\overline{WR1}$ 输入负脉冲时, $\overline{LE1}$ 信号为正脉冲。 $\overline{LE1}$ 为高电平时, 输入锁存器的状态随着数据输入线的状态变化, $\overline{LE1}$ 的负跳变将数据线上的信息锁入输入锁存器。

(3) $\overline{WR1}$: 写信号 1, 输入低电平有效。当 $\overline{WR1}$ 、 \overline{CS} 、 I_{LE} 均为有效时, 可将数据写入输入锁存器。

(4) $\overline{WR2}$: 写信号 2, 输入低电平有效。当其有效时, 在传送控制信号 \overline{XFER} 的作用下, 可将锁存在输入锁存器的 8 位数据送到 DAC 寄存器。

(5) \overline{XFER} : 数据传送控制信号, 输入低电平有效。当 \overline{XFER} 为低电平, $\overline{WR2}$ 输入负脉冲时, 则在 $\overline{LE2}$ 产生正脉冲。 $\overline{LE2}$ 为高电平时, DAC 寄存器的输出和输入锁存器状态一致, $\overline{LE2}$ 的负跳变将输入锁存器的内容锁入 DAC 寄存器。

(6) V_{REF} : 基准电压输入端, 可在 $-10V \sim +10V$ 范围内调节。

(7) $DI7 \sim DI0$: 数字量数据输入端。

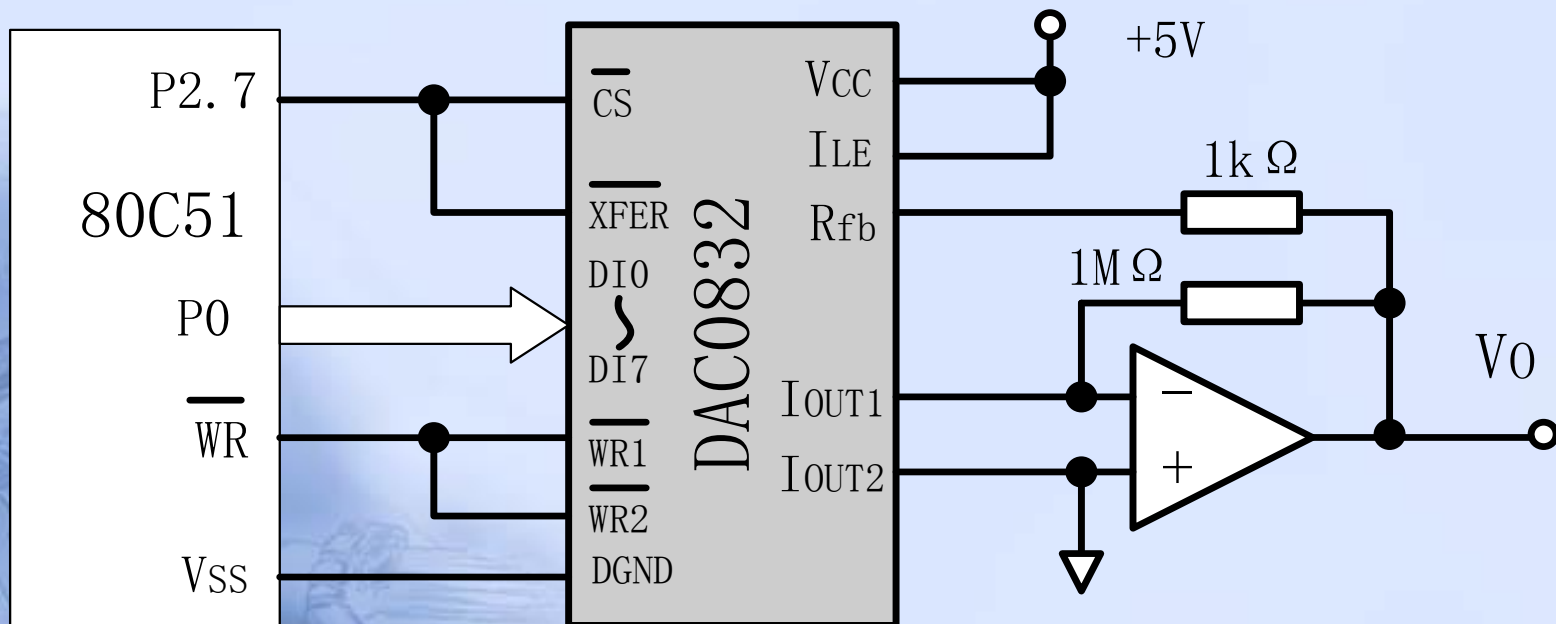
(8) I_{OUT1} 、 I_{OUT2} : 电流输出引脚。电流 I_{OUT1} 与 I_{OUT2} 的和为常数, I_{OUT1} 、 I_{OUT2} 随寄存器的内容线性变化。

(9) R_{fb} : DAC0832 芯片内部反馈电阻引脚。

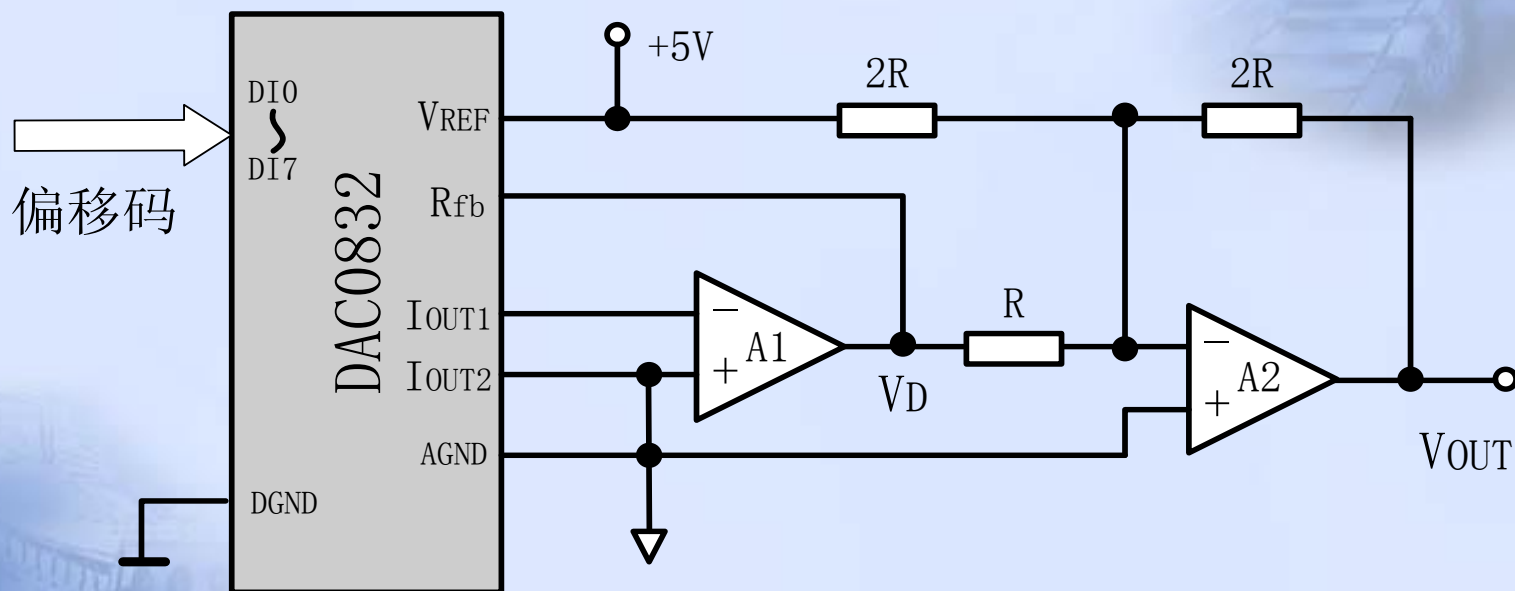
二、DAC0832与80C51单片机的接口

1、单缓冲工作方式

此方式适用于只有一路模拟量输出，或有几路模拟量输出但并不要求同步的系统。



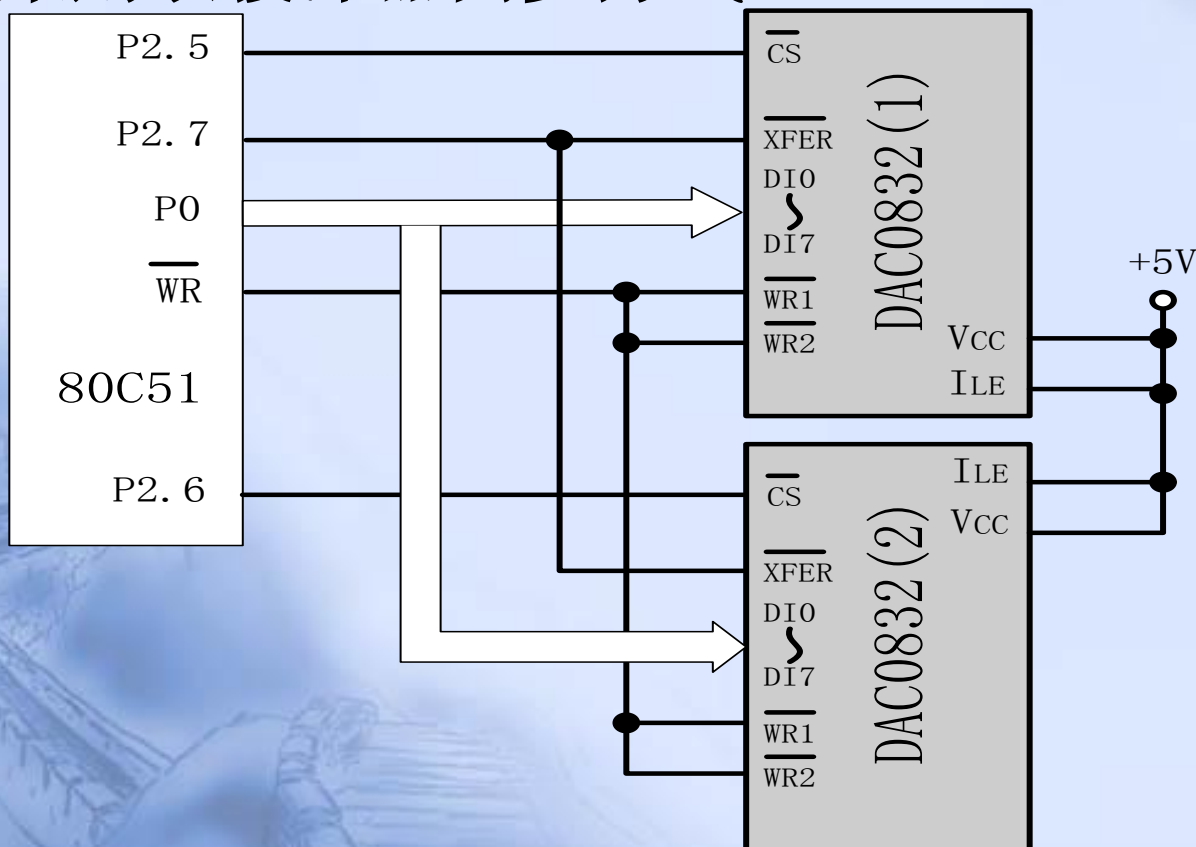
双极性模拟输出电压：



双极性输出时的分辨率比单极性输出时降低 **1/2**，这是由于对双极性输出而言，最高位作为符号位，只有**7**位数值位。

2、双缓冲工作方式

多路D/A转换输出，如果要求同步进行，就应该采用双缓冲器同步方式。



3、直通工作方式

- 当**DAC0832**芯片的片选信号、写信号、及传送控制信号的引脚全部接地，允许输入锁存信号**ILE**引脚接+5V时，**DAC0832**芯片就处于直通工作方式，数字量一旦输入，就直接进入**DAC**寄存器，进行**D/A**转换。

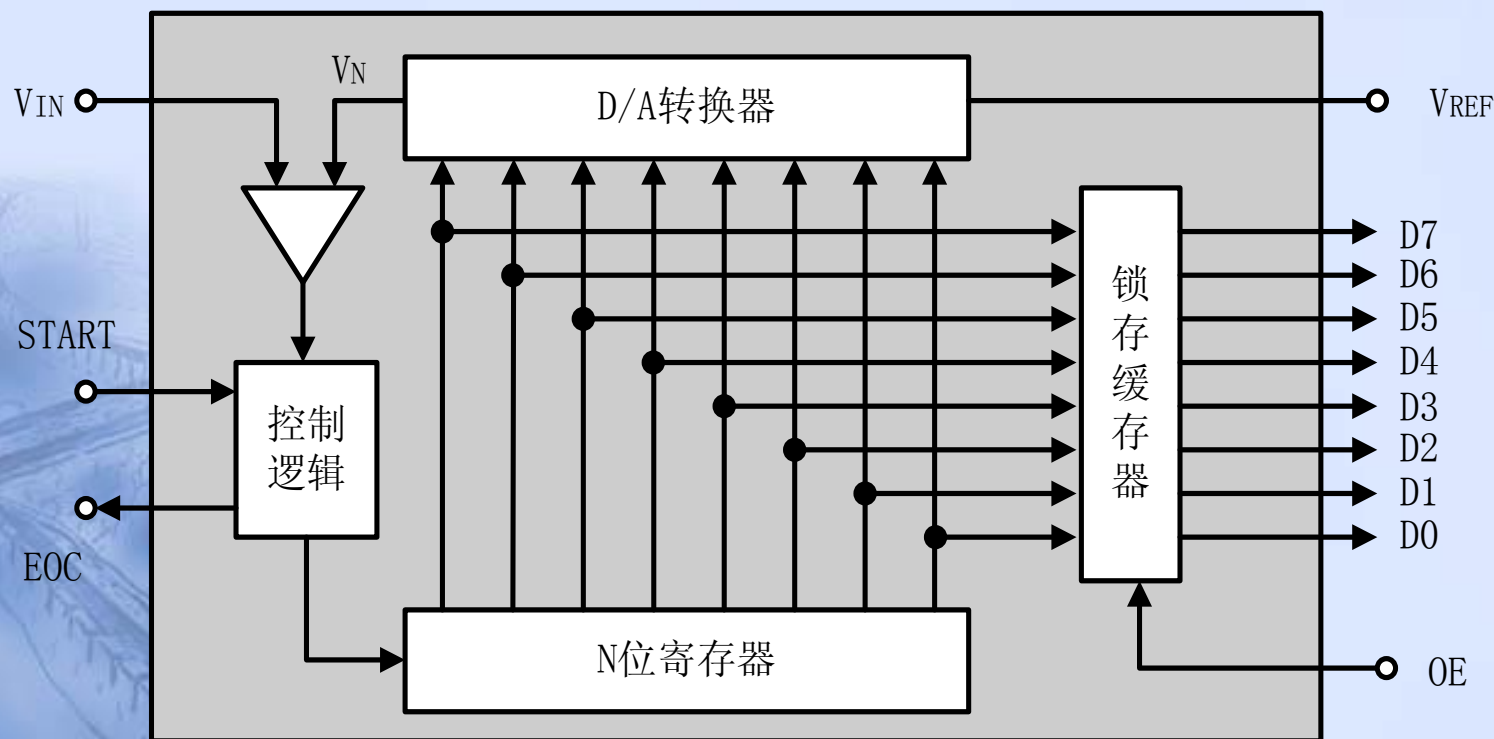
The background is a light blue gradient with a faint, artistic illustration of a city. In the upper right, a person is shown sitting and reading a large book. In the lower left, a bridge with a railing spans a body of water, with a cityscape visible in the distance. The entire scene is framed by a decorative border of small, dark rectangular blocks at the top and bottom.

[http://www.](http://www.Alldatasheet.com/)
Alldatasheet
.com/

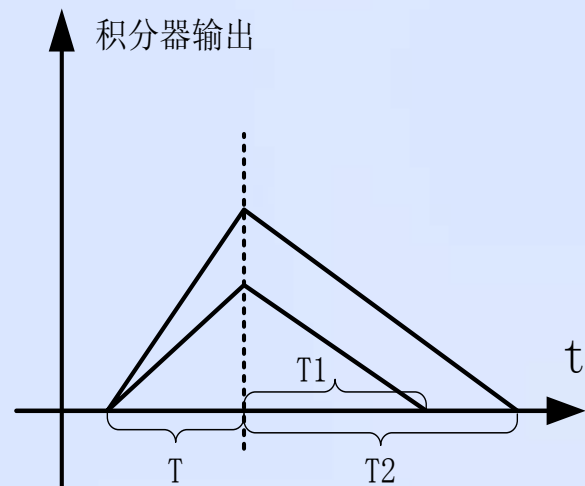
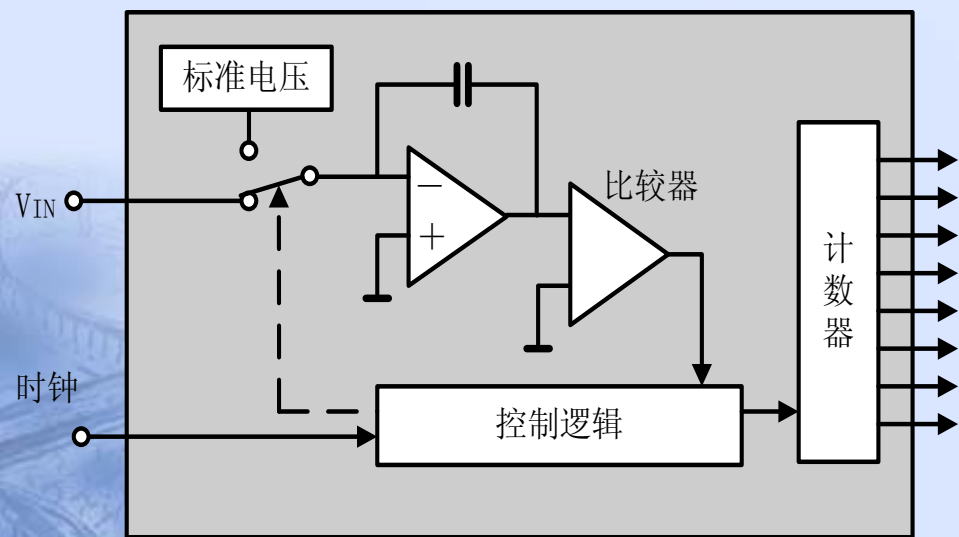
6.2 A/D 转换器及其与单片机接口

6.2.1 A / D 转换器的原理及主要技术指标

一、逐次逼近式ADC的转换原理



二、双积分式ADC的转换原理



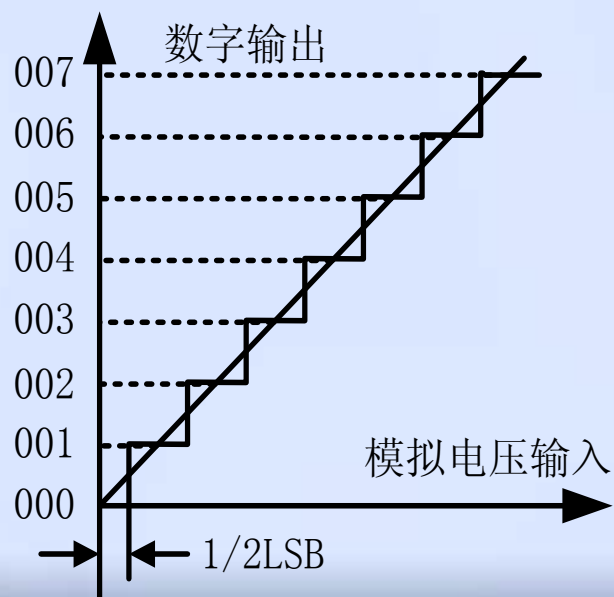
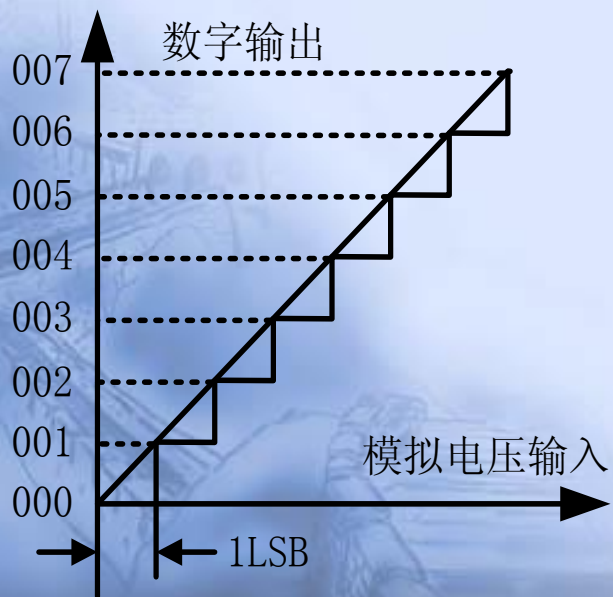
三、A/D转换器的主要技术指标

1、分辨率

■ **ADC的分辨率**是指使输出数字量变化一个相邻数码所需输入模拟电压的变化量。常用二进制的位数表示。例如**12位ADC**的分辨率就是**12位**，或者说分辨率为满刻度**FS**的 **$1/2^{12}$** 。一个**10V**满刻度的**12位ADC**能分辨输入电压变化最小值是 **$10V \times 1/2^{12} = 2.44mV$** 。

2、量化误差

■ **ADC**把模拟量变为数字量，用数字量近似表示模拟量，这个过程称为量化。**量化误差是ADC的有限位数对模拟量进行量化而引起的误差**。实际上，要准确表示模拟量，**ADC**的位数需很大甚至无穷大。一个分辨率有限的**ADC**的阶梯状转换特性曲线与具有无限分辨率的**ADC**转换特性曲线（直线）之间的最大偏差即是量化误差。



3、偏移误差

■ **偏移误差**是指输入信号为零时，输出信号不为零的值，所以有时又称为零值误差。假定**ADC**没有非线性误差，则其转换特性曲线各阶梯中点的连线必定是直线，这条直线与横轴相交点所对应的输入电压值就是偏移误差。

4、满刻度误差

■ **满刻度误差**又称为增益误差。**ADC**的满刻度误差是指满刻度输出数码所对应的实际输入电压与理想输入电压之差。

5、线性度

■ **线性度**有时又称为非线性度，它是指转换器实际的转换特性与理想直线的最大偏差。

6、绝对精度

■ 在一个转换器中，任何数码所对应的实际模拟量输入与理论模拟输入之差的最大值，称为**绝对精度**。对于**ADC**而言，可以在每一个阶梯的水平中点进行测量，它包括了所有的误差。

7、转换速率

■ **ADC**的**转换速率**是能够重复进行数据转换的速度，即每秒转换的次数。而完成一次**A/D**转换所需的时间（包括稳定时间），则是转换速率的倒数。

第七讲

7.1 计算机串行通信基础

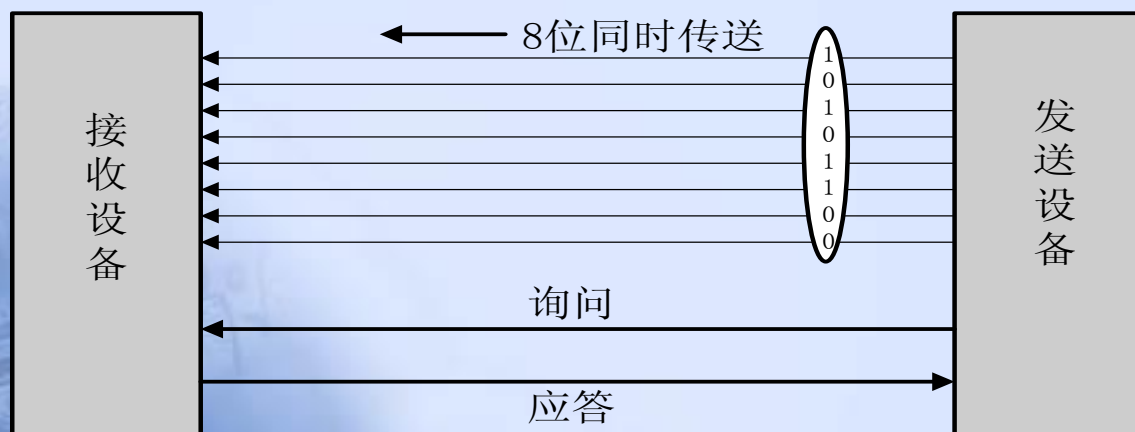
7.2 80C51的串行口

7.3 单片机串行口编程应用 举例

7.1 计算机串行通信基础

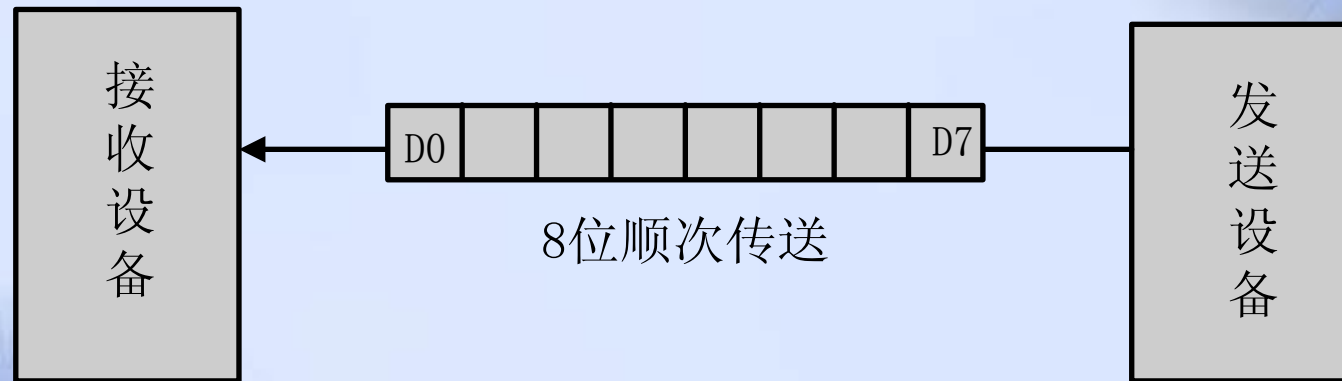
- 随着多微机系统的广泛应用和计算机网络技术的普及，计算机的通信功能愈来愈显得重要。**计算机通信**是指计算机与外部设备或计算机与计算机之间的信息交换。
- 通信有**并行通信**和**串行通信**两种方式。在多微机系统以及现代测控系统中信息的交换多采用串行通信方式。

- **计算机通信**是将计算机技术和通信技术的相结合，完成计算机与外部设备或计算机与计算机之间的信息交换。可以分为两大类：并行通信与串行通信。
- **并行通信**通常是将数据字节的各位用多条数据线同时进行传送。



并行通信控制简单、传输速度快；由于传输线较多，长距离传送时成本高且接收方的各位同时接收存在困难。

串行通信是将数据字节分成一位一位的形式在一条传输线上逐个地传送。



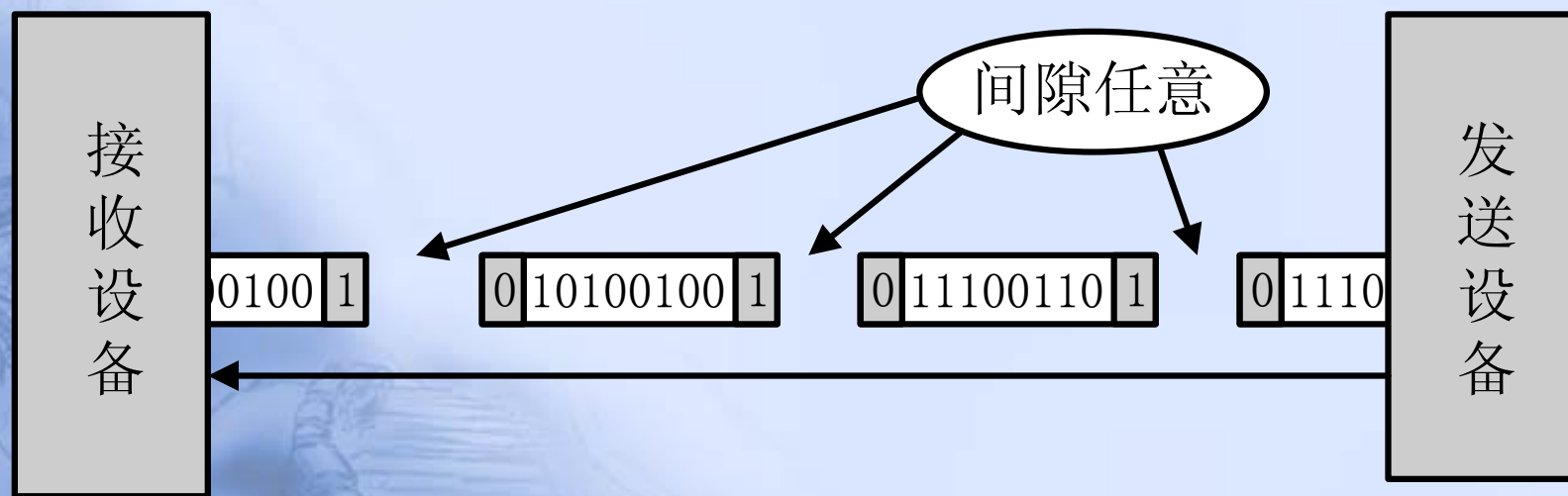
串行通信的特点：传输线少，长距离传送时成本低，且可以利用电话网等现成的设备，但数据的传送控制比并行通信复杂。

7.1.1 串行通信的基本概念

一、异步通信与同步通信

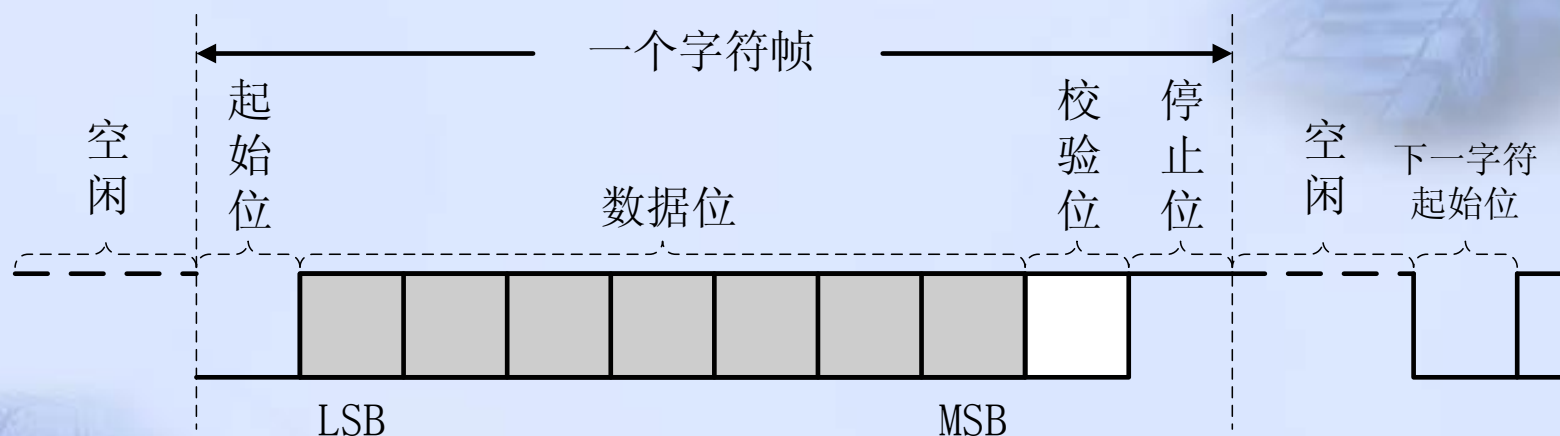
1、异步通信

异步通信是指通信的发送与接收设备使用各自的时钟控制数据的发送和接收过程。为使双方的收发协调，要求发送和接收设备的时钟尽可能一致。



异步通信是以字符（构成的帧）为单位进行传输，字符与字符之间的间隙（时间间隔）是任意的，但每个字符中的各位是以固定的时间传送的，即字符之间不一定有“位间隔”的整数倍的关系，但同一字符内的各位之间的距离均为“位间隔”的整数倍。

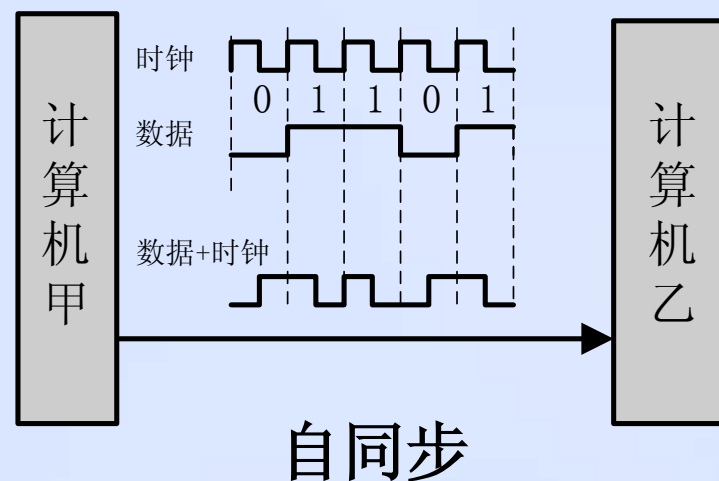
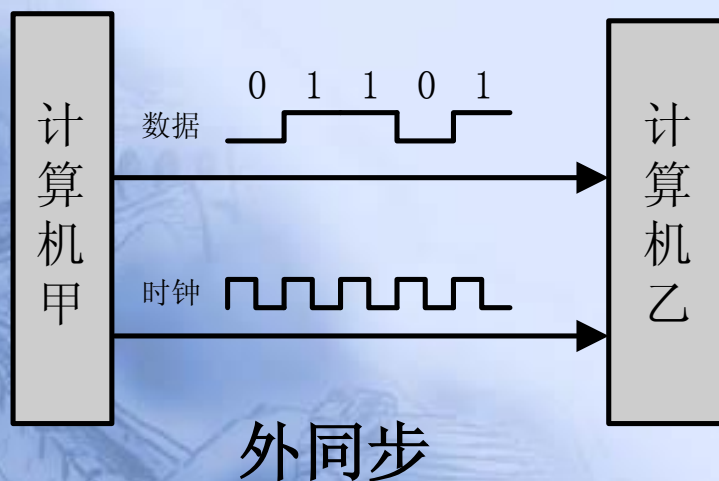
异步通信的数据格式：



异步通信的特点：不要求收发双方时钟的严格一致，实现容易，设备开销较小，但每个字符要附加**2~3**位用于起止位，各帧之间还有间隔，因此传输效率不高。

2、同步通信

同步通信时要建立发送方时钟对接收方时钟的直接控制，使双方达到完全同步。此时，传输数据的位之间的距离均为“位间隔”的整数倍，同时传送的字符间不留间隙，即**保持位同步**关系，**也保持字符同步**关系。发送方对接收方的同步可以通过两种方法实现。



面向字符的同步格式：

SYN	SYN	SOH	标题	STX	数据块	ETB/ETX	块校验
-----	-----	-----	----	-----	-----	---------	-----

此时，传送的数据和控制信息都必须由规定的字符集（如**ASCII**码）中的字符所组成。图中帧头为1个或2个同步字符**SYN**（**ASCII**码为16H）。**SOH**为序始字符（**ASCII**码为01H），表示标题的开始，标题中包含源地址、目标地址和路由指示等信息。**STX**为文始字符（**ASCII**码为02H），表示传送的数据块开始。数据块是传送的正文内容，由多个字符组成。数据块后面是组终字符**ETB**（**ASCII**码为17H）或文终字符**ETX**（**ASCII**码为03H）。然后是校验码。典型的面向字符的同步规程如**IBM**的二进制同步规程**BSC**。

面向位的同步格式：

8位	8位	8位	≥ 0 位	16位	8位
01111110	地址场	控制场	信息场	校验场	01111110

此时，将数据块看作数据流，并用序列**01111110**作为开始和结束标志。为了避免在数据流中出现序列**01111110**时引起的混乱，发送方总是在其发送的数据流中每出现**5**个连续的**1**就插入一个附加的**0**；接收方则每检测到**5**个连续的**1**并且其后有一个**0**时，就删除该**0**。

典型的面向位的同步协议如**ISO**的高级数据链路控制规程**HDLC**和**IBM**的同步数据链路控制规程**SDLC**。

同步通信的特点是以特定的位组合“**01111110**”作为帧的开始和结束标志，所传输的一帧数据可以是任意位。所以传输的效率较高，但实现的硬件设备比异步通信复杂。

二、串行通信的传输方向

1、单工

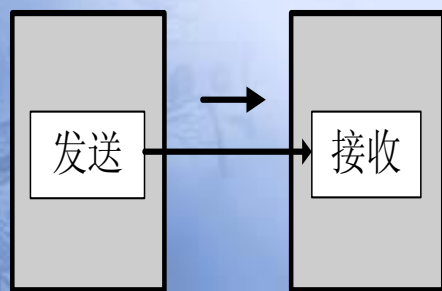
单工是指数据传输仅能沿一个方向，不能实现反向传输。

2、半双工

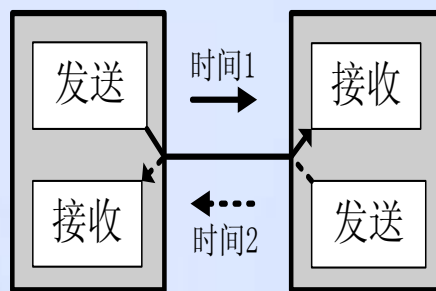
半双工是指数据传输可以沿两个方向，但需要分时进行。

3、全双工

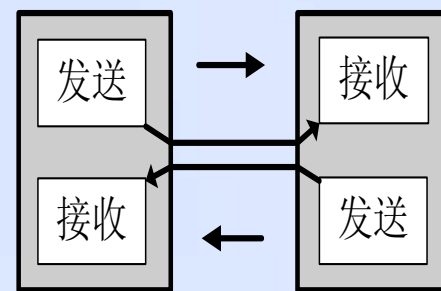
全双工是指数据可以同时进行双向传输。



单工



半双工



全双工

三、信号的调制与解调

利用调制器（**Modulator**）把**数字信号转换成模拟信号**，然后送到通信线路上去，再由解调器（**Demodulator**）把从通信线路上收到的**模拟信号转换成数字信号**。由于通信是双向的，调制器和解调器合并在一个装置中，这就是调制解调器**MODEM**。



四、串行通信的错误校验

1、奇偶校验

在发送数据时，数据位尾随的1位为奇偶校验位（1或0）。奇校验时，数据中“1”的个数与校验位“1”的个数之和应为奇数；偶校验时，数据中“1”的个数与校验位“1”的个数之和应为偶数。接收字符时，对“1”的个数进行校验，若发现不一致，则说明传输数据过程中出现了差错。

2、代码和校验

代码和校验是发送方将所发数据块求和（或各字节异或），产生一个字节的校验字符（校验和）附加到数据块末尾。接收方接收数据同时对数据块（除校验字节外）求和（或各字节异或），将所得的结果与发送方的“校验和”进行比较，相符则无差错，否则即认为传送过程中出现了差错。

3、循环冗余校验

这种校验是通过某种数学运算实现有效信息与校验位之间的循环校验，常用于对磁盘信息的传输、存储区的完整性校验等。这种校验方法纠错能力强，广泛应用于同步通信中。

五、传输速率与传输距离

1、传输速率

比特率是每秒钟传输二进制代码的位数，单位是：位 / 秒 (**bps**)。如每秒钟传送**240**个字符，而每个字符格式包含**10**位(1个起始位、1个停止位、**8**个数据位)，这时的比特率为：

$$10\text{位} \times 240\text{个/秒} = 2400\text{ bps}$$

2、传输距离与传输速率的关系

串行接口或终端直接传送串行信息位流的最大距离与传输速率及传输线的电气特性有关。当传输线使用每**0.3m**（约1英尺）有**50PF**电容的非平衡屏蔽双绞线时，传输距离随传输速率的增加而减小。当比特率超过**1000 bps** 时，最大传输距离迅速下降，如**9600 bps** 时最大距离下降到只有**76m**（约250英尺）。

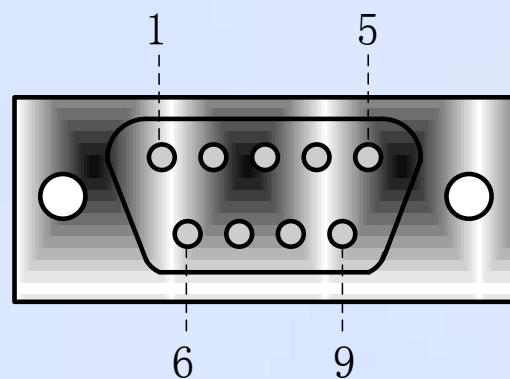
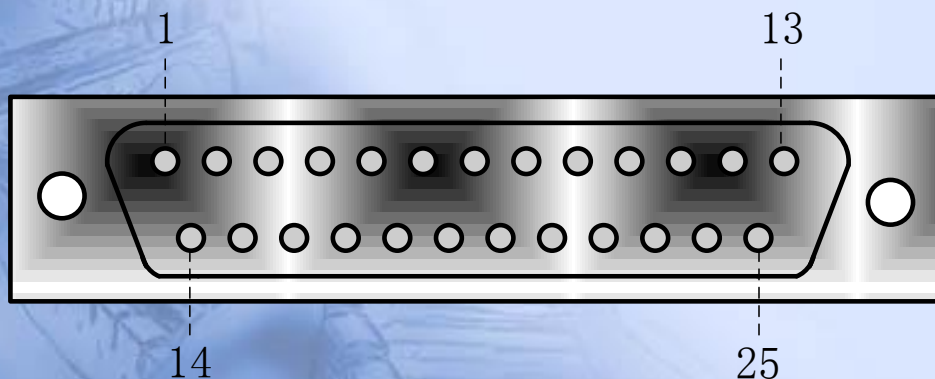
7.1.2 串行通信接口标准

一、RS-232C接口

RS-232C是**EIA**（美国电子工业协会）**1969**年修订**RS-232C**标准。**RS-232C**定义了数据终端设备（**DTE**）与数据通信设备（**DCE**）之间的物理接口标准。

1、机械特性

RS-232C接口规定使用**25**针连接器，连接器的尺寸及每个插针的排列位置都有明确的定义。（阳头）



2、功能特性

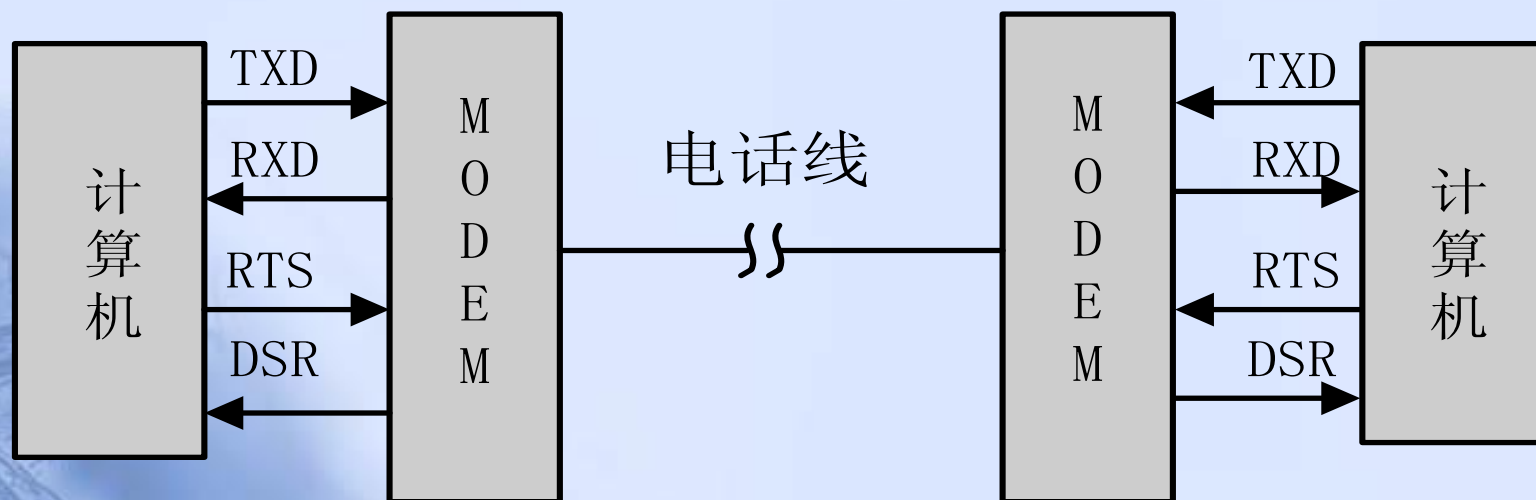
RS-232C 标准接口主要引脚定义

插针序号	信号名称	功能	信号方向
1	PGND	保护接地	
2 (3)	TXD	发送数据 (串行输出)	DTE→DCE
3 (2)	RXD	接收数据 (串行输入)	DTE←DCE
4 (7)	RTS	请求发送	DTE→DCE
5 (8)	CTS	允许发送	DTE←DCE
6 (6)	DSR	DCE 就绪 (数据建立就绪)	DTE←DCE
7 (5)	SGND	信号接地	
8 (1)	DCD	载波检测	DTE←DCE
20 (4)	DTR	DTE 就绪 (数据终端准备就绪)	DTE→DCE
22 (9)	RI	振铃指示	DTE←DCE

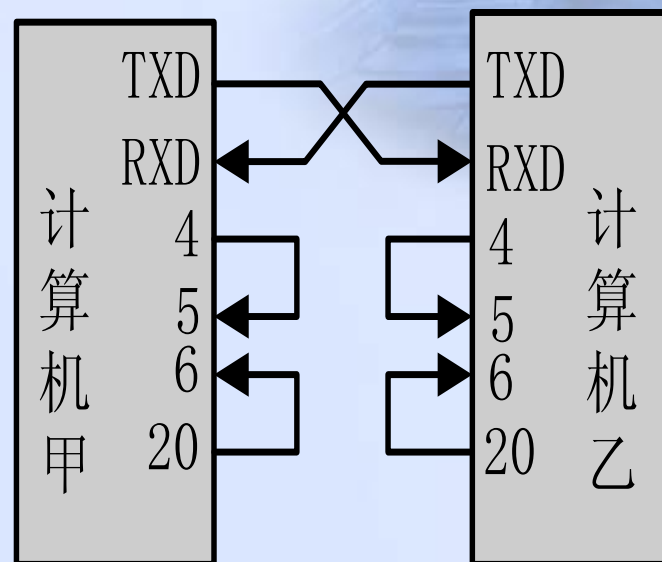
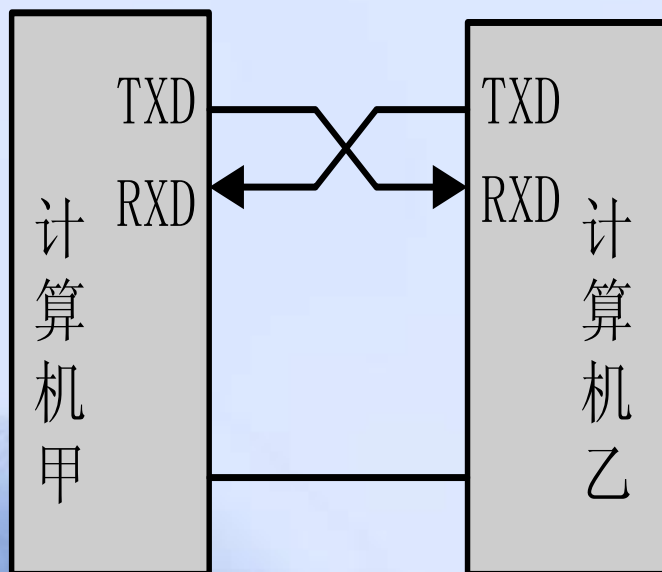
注：插针序号 () 内为 9 针非标准连接器的引脚号

4、过程特性

过程特性规定了信号之间的时序关系，以便正确地接收和发送数据。

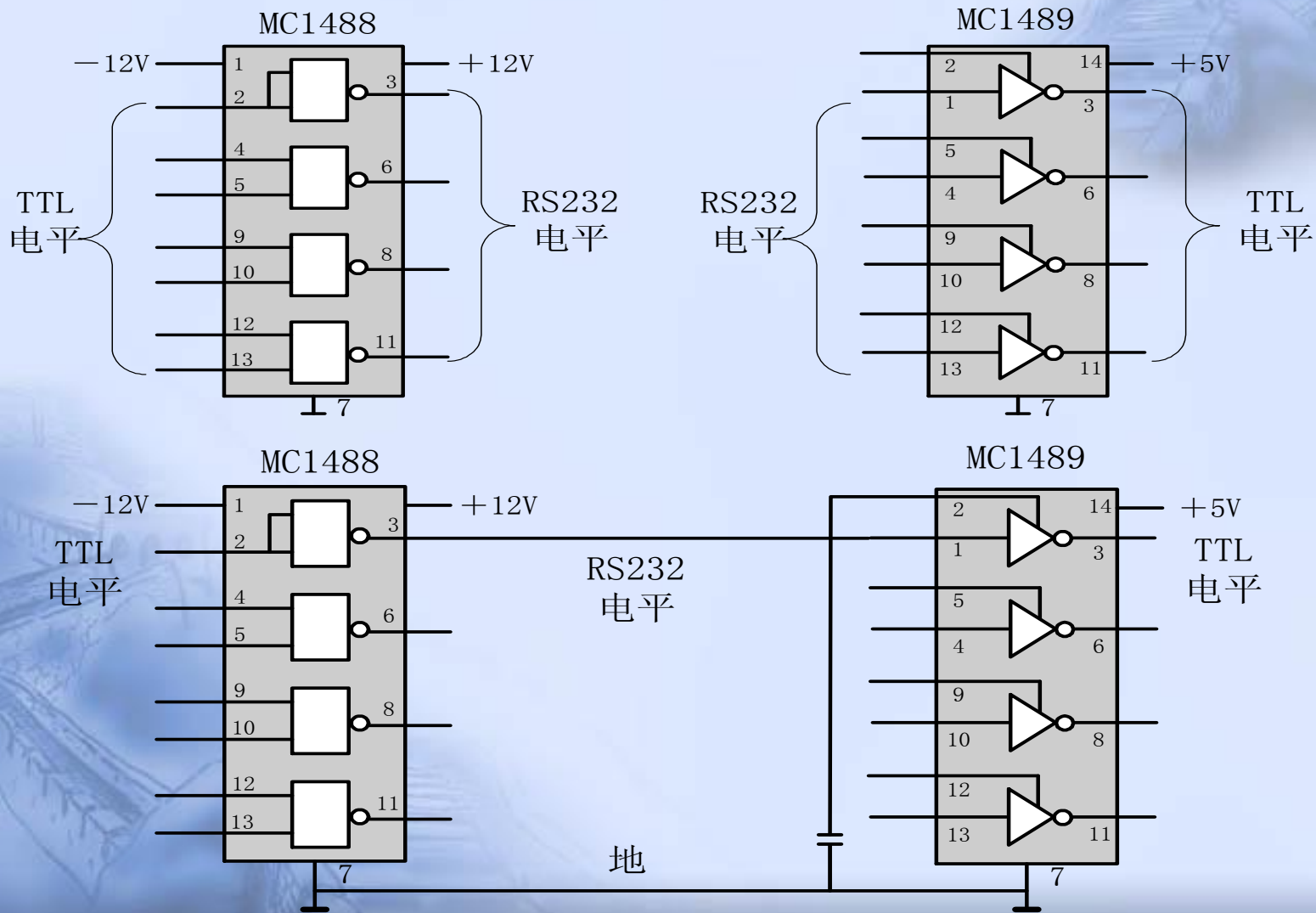


远程通信连接



近程通信连接

5、RS-232C电平与TTL电平转换驱动电路



6、采用RS-232C接口存在的问题

1、传输距离短，传输速率低

RS-232C总线标准受电容允许值的约束，使用时传输距离一般不要超过**15米**（线路条件好时也不超过几十米）。最高传送速率为**20Kbps**。

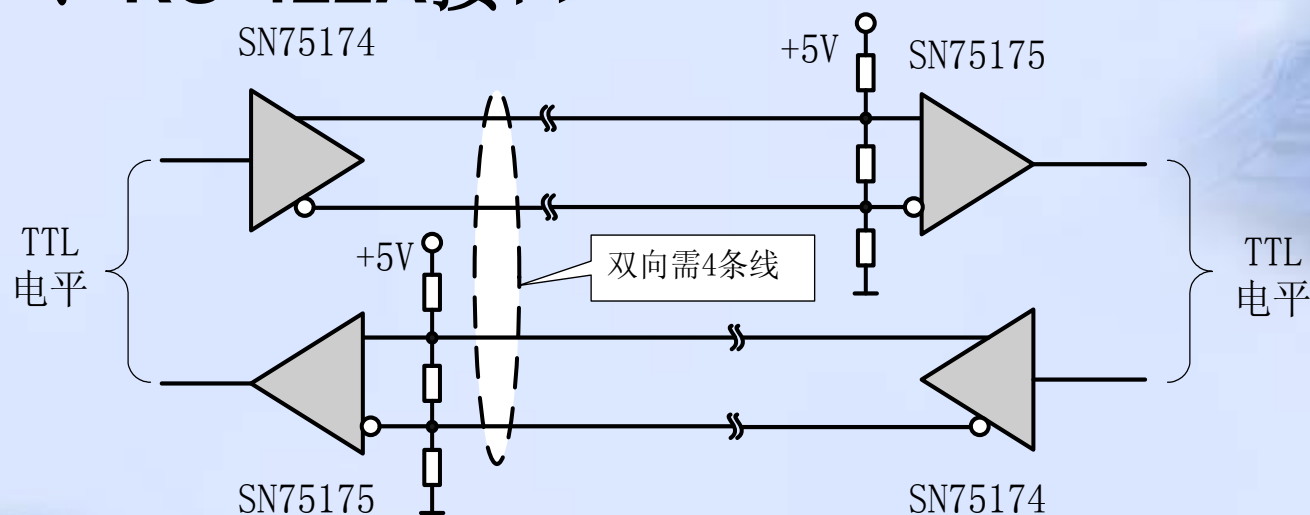
2、有电平偏移

RS-232C总线标准要求收发双方共地。通信距离较大时，收发双方的地电位差别较大，在信号地上将有比较大的地电流并产生压降。

3、抗干扰能力差

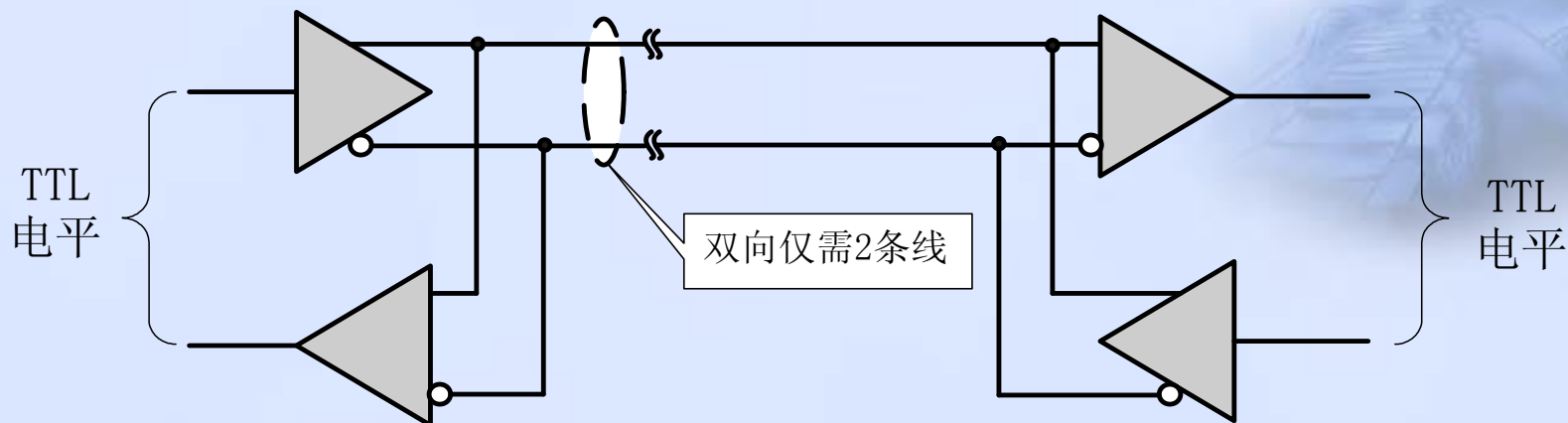
RS-232C在电平转换时采用单端输入输出，在传输过程中当干扰和噪声混在正常的信号中。为了提高信噪比，**RS-232C**总线标准不得不采用比较大的电压摆幅。

二、RS-422A接口



RS-422A输出驱动器为双端平衡驱动器。如果其中一条线为逻辑“1”状态，另一条线就为逻辑“0”，比采用单端不平衡驱动对电压的放大倍数大一倍。**差分电路**能从地线干扰中拾取有效信号，差分接收器可以分辨**200mV**以上电位差。若传输过程中混入了干扰和噪声，由于差分放大器的作用，可使干扰和噪声相互抵消。因此可以避免或大大减弱地线干扰和电磁干扰的影响。**RS-422A传输速率（90Kbps）时，传输距离可达1200米。**

三、RS-485接口



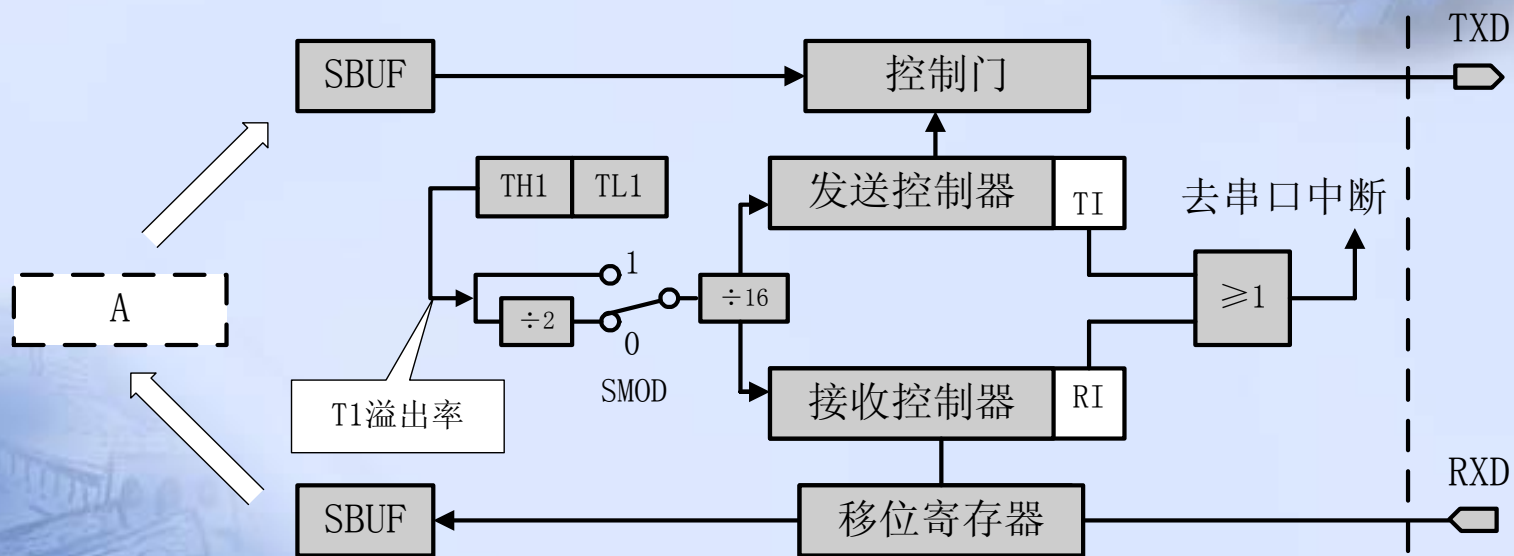
RS-485是RS-422A的变型：RS-422A用于全双工，而RS-485则用于半双工。RS-485是一种多发送器标准，在通信线路上最多可以使用32对差分驱动器/接收器。如果在一个网络中连接的设备超过32个，还可以使用中继器。

RS-485的信号传输采用两线间的电压来表示逻辑1和逻辑0。由于发送方需要两根传输线，接收方也需要两根传输线。传输线采用差动信道，所以它的干扰抑制性极好，又因为它的阻抗低，无接地问题，所以传输距离可达1200米，传输速率可达1Mbps。

RS-485是一点对多点的通信接口，一般采用**双绞线**的结构。普通的**PC**机一般不带**RS485**接口，因此要使用**RS-232C/RS-485**转换器。对于单片机可以通过芯片**MAX485**来完成**TTL/RS-485**的电平转换。在计算机和单片机组成的**RS-485**通信系统中，下位机由单片机系统组成，上位机为普通的**PC**机，负责监视下位机的运行状态，并对其状态信息进行集中处理，以图文方式显示下位机的工作状态以及工业现场被控设备的工作状况。系统中各节点（包括上位机）的识别是通过设置不同的站地址来实现的。

7.2 80C51的串行口

7.2.1 80C51串行口的结构



有两个物理上独立的接收、发送缓冲器**SBUF**，它们占用同一地址**99H**；接收器是双缓冲结构；发送缓冲器，因为发送时**CPU**是主动的，不会产生重叠错误。

7.2.2 80C51串行口的控制寄存器

SCON 是一个特殊功能寄存器，用以设定串行口的工作方式、接收/发送控制以及设置状态标志：

位	7	6	5	4	3	2	1	0	
字节地址：98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

SM0和**SM1**为工作方式选择位，可选择四种工作方式：

串行口的工作方式				
SM0	SM1	方 式	说 明	波特率
0	0	0	移位寄存器	$f_{osc}/12$
0	1	1	10 位异步收发器 (8 位数据)	可变
1	0	2	11 位异步收发器 (9 位数据)	$f_{osc}/64$ 或 $f_{osc}/32$
1	1	3	11 位异步收发器 (9 位数据)	可变

● **SM2**，多机通信控制位，主要用于方式2和方式3。当接收机的**SM2=1**时可以利用收到的**RB8**来控制是否激活**RI**（**RB8=0**时不激活**RI**，收到的信息丢弃；**RB8=1**时收到的数据进入**SBUF**，并激活**RI**，进而在中断服务中将数据从**SBUF**读走）。当**SM2=0**时，不论收到的**RB8**为0和1，均可以使收到的数据进入**SBUF**，并激活**RI**（即此时**RB8**不具有控制**RI**激活的功能）。通过控制**SM2**，可以实现多机通信。

在方式0时，**SM2**必须是0。在方式1时，若**SM2=1**，则只有接收到有效停止位时，**RI**才置1。

● **REN**，允许串行接收位。由软件置**REN=1**，则启动串行口接收数据；若软件置**REN=0**，则禁止接收。

●**TB8**，在方式**2**或方式**3**中，是发送数据的**第九位**，可以用软件规定其作用。可以用作数据的奇偶校验位，或在多机通信中，作为地址帧/数据帧的标志位。

在方式**0**和方式**1**中，该位未用。

●**RB8**，在方式**2**或方式**3**中，是接收到数据的**第九位**，作为奇偶校验位或地址帧/数据帧的标志位。在方式**1**时，若**SM2=0**，则**RB8**是接收到的停止位。

● **TI，发送中断标志位**。在方式0时，当串行发送第8位数据结束时，或在其它方式，串行发送停止位的开始时，由内部硬件使TI置1，向CPU发中断申请。在中断服务程序中，必须用软件将其清0，取消此中断申请。

● **RI，接收中断标志位**。在方式0时，当串行接收第8位数据结束时，或在其它方式，串行接收停止位的中间时，由内部硬件使RI置1，向CPU发中断申请。也必须在中断服务程序中，用软件将其清0，取消此中断申请。

PCON中只有一位SMOD与串行口工作有关：

位	7	6	5	4	3	2	1	0	
字节地址: 97H	SMOD								PCON

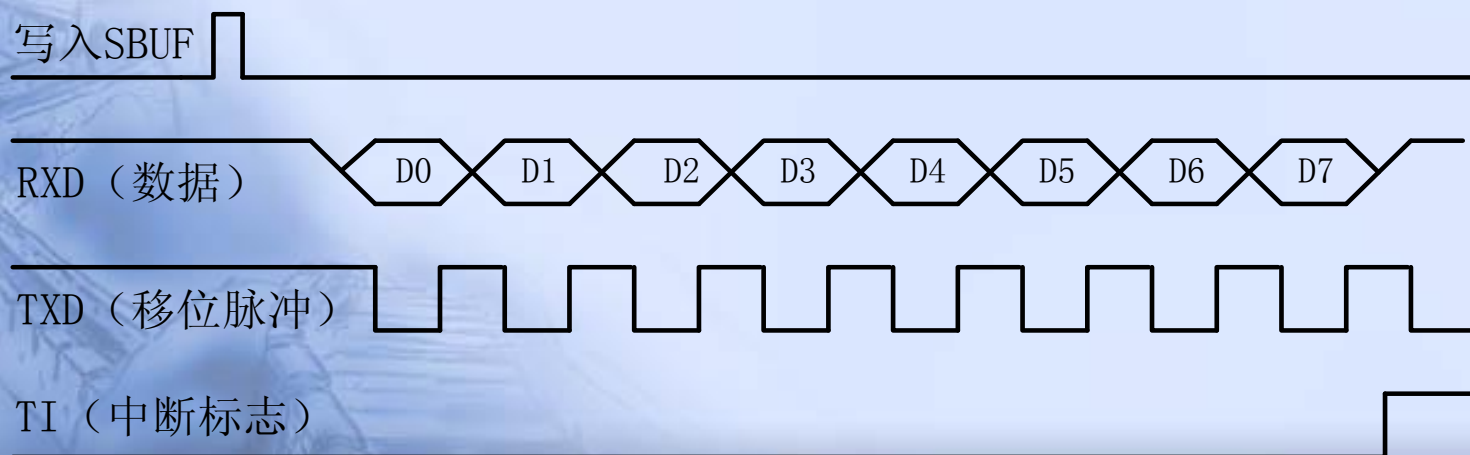
SMOD (PCON.7) 波特率倍增位。在串行口方式1、方式2、方式3时，波特率与**SMOD**有关，当**SMOD=1**时，波特率提高一倍。复位时，**SMOD=0**。

7.2.3 80C51串行口的工作方式

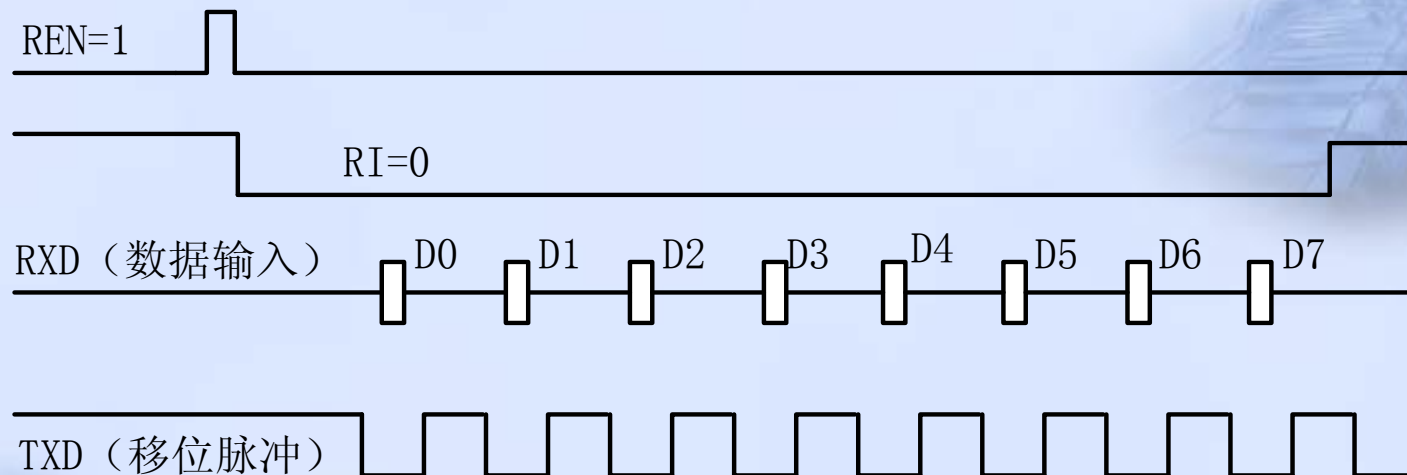
一、方式0

方式0时，串行口为同步移位寄存器的输入输出方式。主要用于扩展并行输入或输出口。数据由**RXD (P3.0)**引脚输入或输出，同步移位脉冲由**TXD (P3.1)**引脚输出。发送和接收均为**8位数据**，低位在先，高位在后。波特率固定为 **$f_{osc}/12$** 。

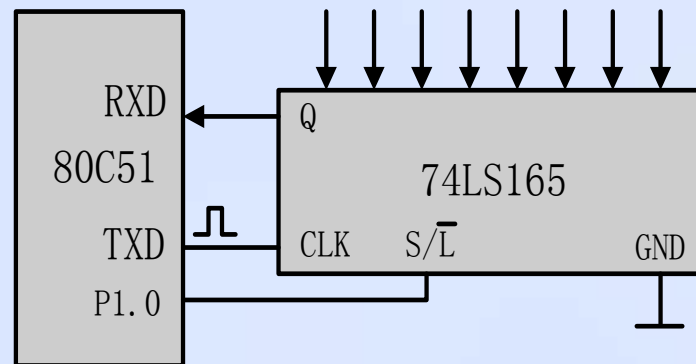
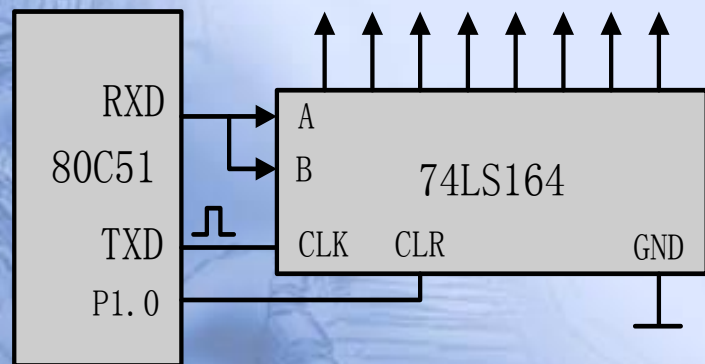
1、方式0输出



2、方式0输入

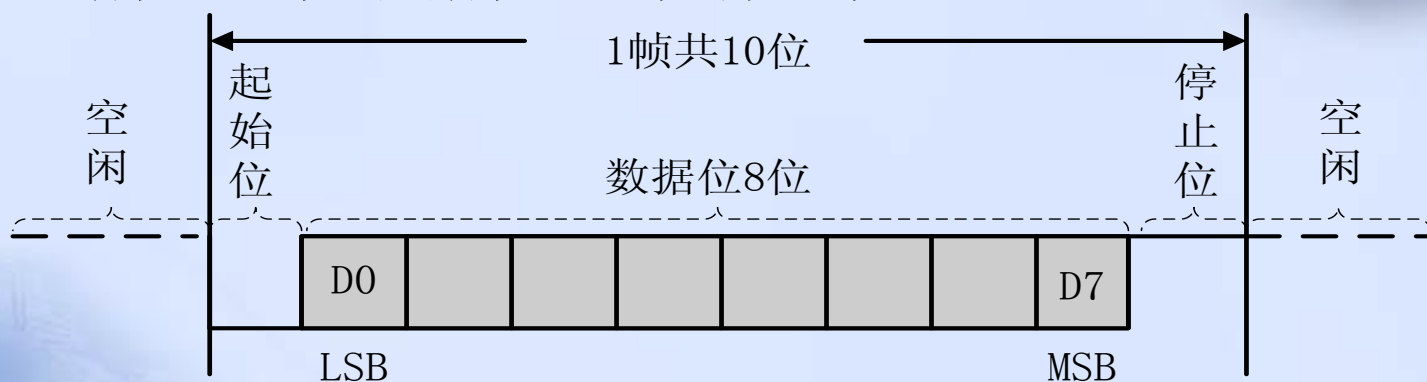


方式0接收和发送电路

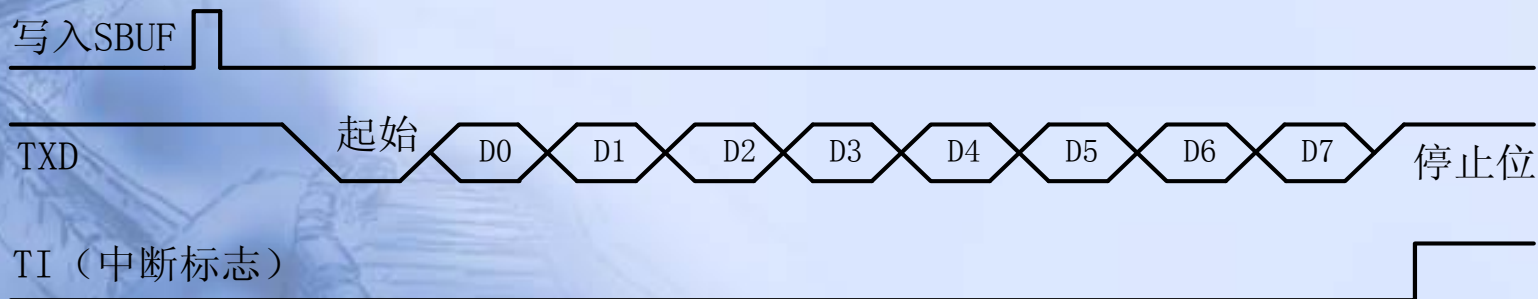


二、方式1

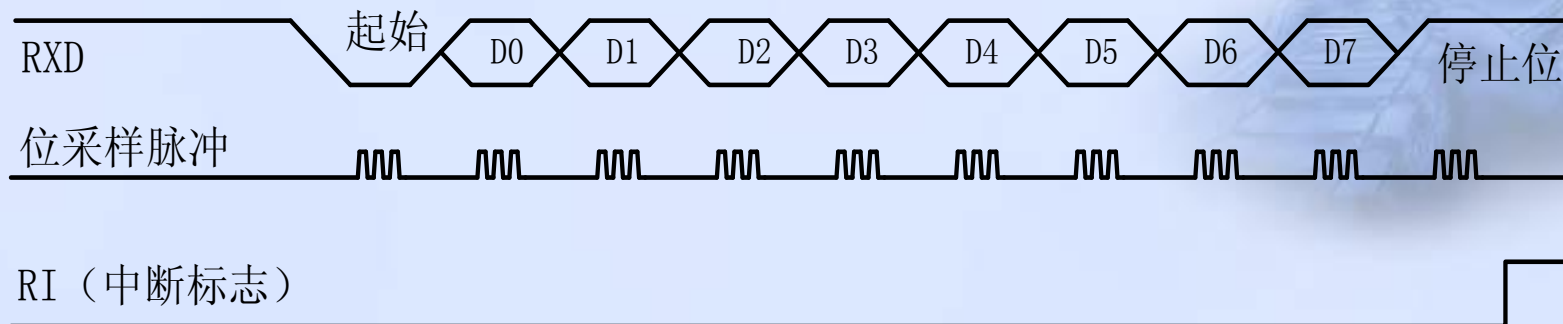
方式1是10位数据的异步通信口。**TXD**为数据发送引脚，**RXD**为数据接收引脚，传送一帧数据的格式如图所示。其中1位起始位，8位数据位，1位停止位。



1、方式1输出



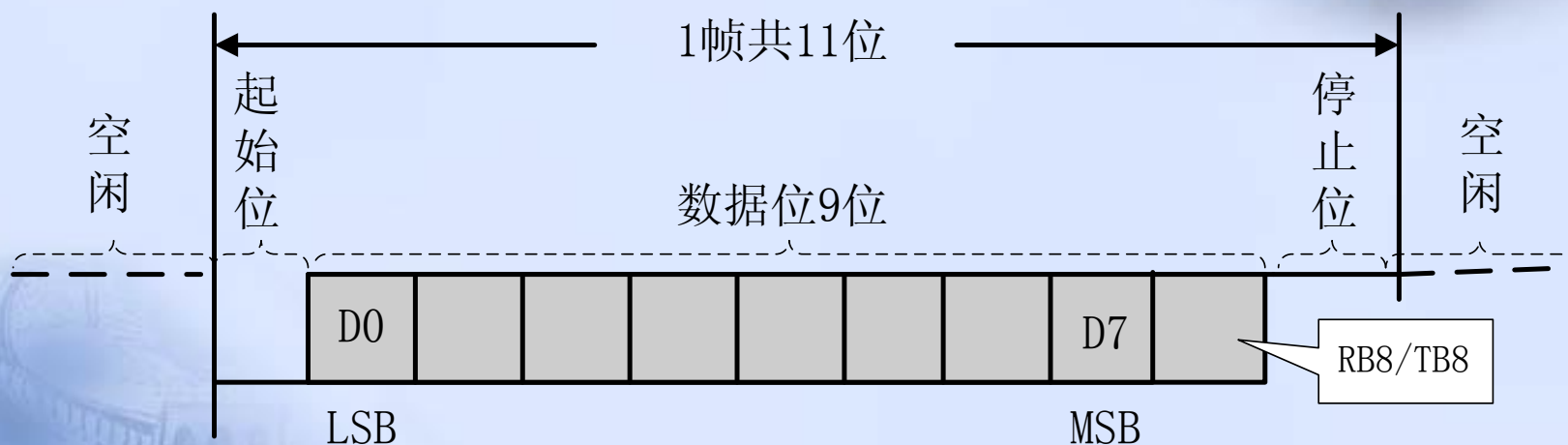
2、方式1输入



用软件置**REN**为1时，接收器以所选择波特率的**16**倍速率采样**RXD**引脚电平，检测到**RXD**引脚输入电平发生负跳变时，则说明起始位有效，将其移入输入移位寄存器，并开始接收这一帧信息的其余位。接收过程中，数据从输入移位寄存器右边移入，起始位移至输入移位寄存器最左边时，控制电路进行最后一次移位。当**RI=0**，且**SM2=0**（或接收到的停止位为1）时，将接收到的**9**位数据的前**8**位数据装入接收**SBUF**，第**9**位（停止位）进入**RB8**，并置**RI=1**，向**CPU**请求中断。

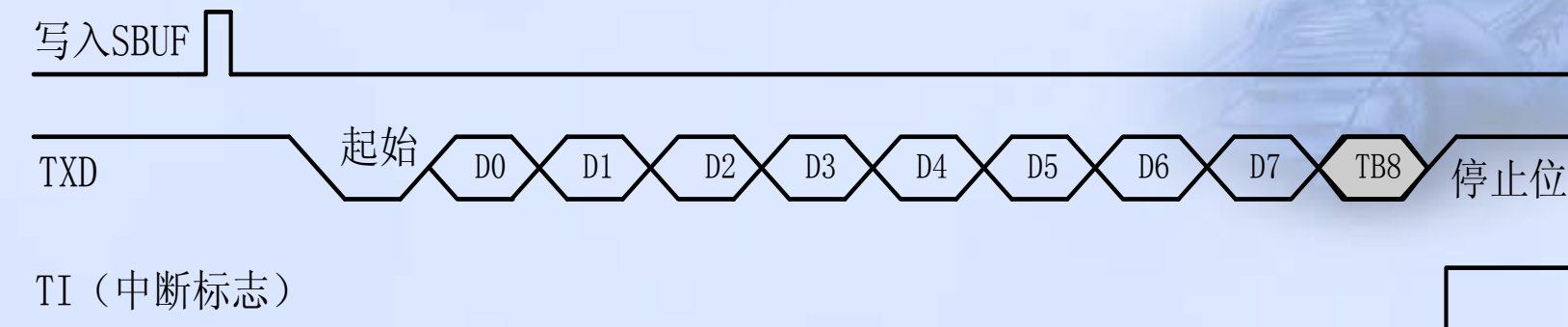
三、方式2和方式3

方式2或方式3时为11位数据的异步通信口。**TXD**为数据发送引脚，**RXD**为数据接收引脚。



方式2和方式3时起始位1位，数据9位（含1位附加的第9位，发送时为**SCON**中的**TB8**，接收时为**RB8**），停止位1位，一帧数据为11位。方式2的波特率固定为晶振频率的1/64或1/32，方式3的波特率由定时器**T1**的溢出率决定。

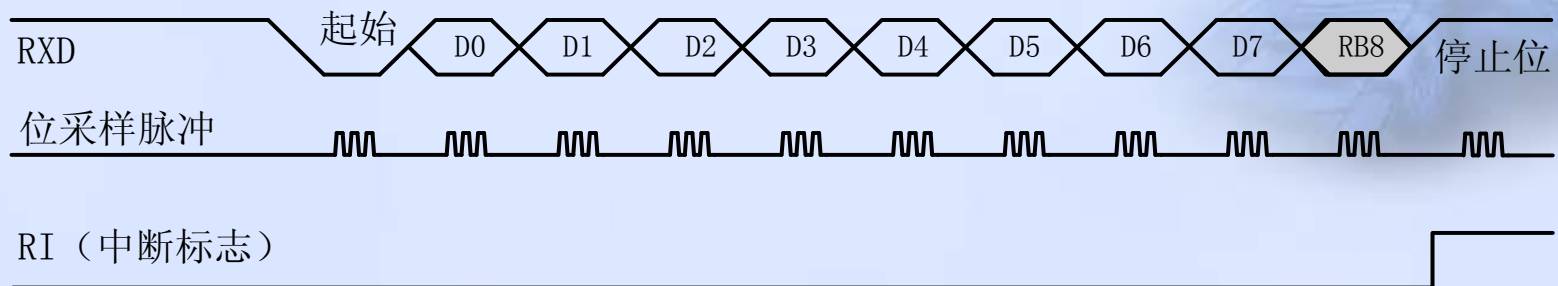
1、方式2和方式3输出



发送开始时，先把起始位**0**输出到**TXD**引脚，然后发送移位寄存器的输出位（**D0**）到**TXD**引脚。每一个移位脉冲都使输出移位寄存器的各位右移一位，并由**TXD**引脚输出。

第一次移位时，停止位“**1**”移入输出移位寄存器的第**9**位上，以后每次移位，左边都移入**0**。当停止位移至输出位时，左边其余位全为**0**，检测电路检测到这一条件时，使控制电路进行最后一次移位，并置**TI=1**，向**CPU**请求中断。

2、方式2和方式3输入



接收时，数据从右边移入输入移位寄存器，在起始位0移到最左边时，控制电路进行最后一次移位。当**RI=0**，且**SM2=0**（或接收到的第9位数据为1）时，接收到的数据装入接收缓冲器**SBUF**和**RB8**（接收数据的第9位），置**RI=1**，向**CPU**请求中断。如果条件不满足，则数据丢失，且不置位**RI**，继续搜索**RXD**引脚的负跳变。

四、波特率的计算

在串行通信中，收发双方对发送或接收数据的速率要有约定。通过软件可对单片机串行口编程为四种工作方式，其中方式0和方式2的波特率是固定的，而方式1和方式3的波特率是可变的，由定时器T1的溢出率来决定。

串行口的四种工作方式对应三种波特率。由于输入的移位时钟的来源不同，所以，各种方式的波特率计算公式也不相同。

方式0的波特率 = $f_{osc}/12$

方式2的波特率 = $(2^{SMOD}/64) \cdot f_{osc}$

方式1的波特率 = $(2^{SMOD}/32) \cdot (T1 \text{溢出率})$

方式3的波特率 = $(2^{SMOD}/32) \cdot (T1 \text{溢出率})$

当T1作为波特率发生器时，最典型的用法是使T1工作在自动再装入的8位定时器方式（即方式2，且TCON的TR1=1，以启动定时器）。这时溢出率取决于TH1中的计数值。

$$T1 \text{ 溢出率} = f_{osc} / \{12 \times [256 - (TH1)]\}$$

在单片机的应用中，常用的晶振频率为：12MHz和11.0592MHz。所以，选用的波特率也相对固定。常用的串行口波特率以及各参数的关系如表所示。

常用波特率与定时器 1 的参数关系

串口工作方式 及波特率/(b/s)	fosc (MHz)	SMOD	定时器 T1		
			C/T	工作方式	初值
方式 1、3	62.5 k	12	1	2	FFH
	19.2 k	11.0592	1	2	FDH
	9600	11.0592	0	2	FDH
	4800	11.0592	0	2	FAH
	2400	11.0592	0	2	F4H
	1200	11.0592	0	2	E8H

串行口工作之前，应对其进行初始化，主要是设置产生波特率的定时器1、串行口控制和中断控制。具体步骤如下：

- 确定**T1**的工作方式（编程**TMOD**寄存器）；
- 计算**T1**的初值，装载**TH1**、**TL1**；
- 启动**T1**（编程**TCON**中的**TR1**位）；
- 确定串行口控制（编程**SCON**寄存器）；

串行口在中断方式工作时，要进行中断设置（编程**IE**、**IP**寄存器）。

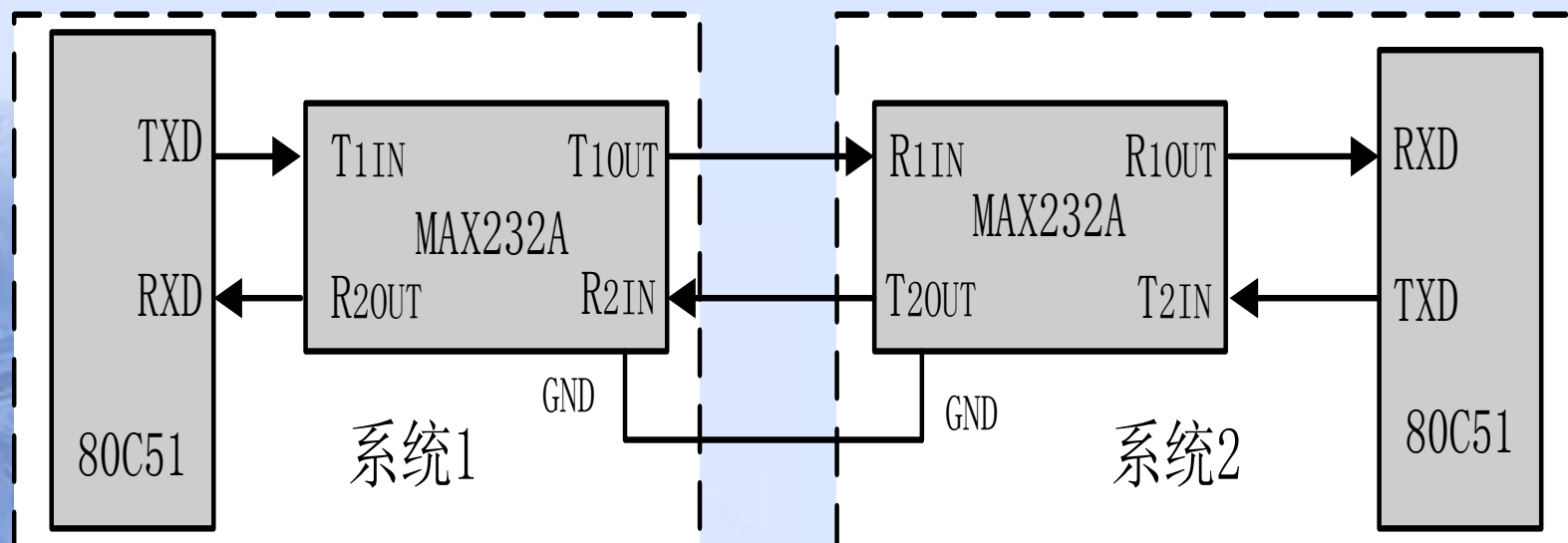
7.3 单片机串行口应用举例

在计算机组成的测控系统中，经常要利用串行通信方式进行数据传输。**80C51**单片机的串行口为计算机间的通信提供了极为便利的条件。利用单片机的串行口还可以方便地扩展键盘和显示器，对于简单的应用非常便利。这里仅介绍单片机串行口在通信方面的应用。

7.3.1 单片机与单片机的通信

一、点对点的通信

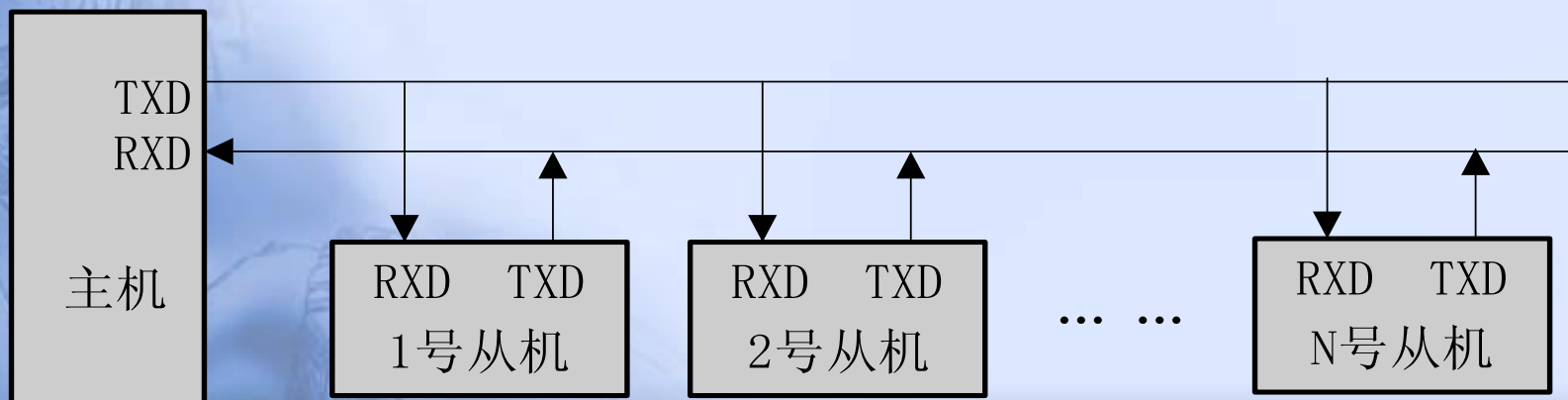
1、硬件连接



二、多机通信

1、硬件连接

单片机构成的多机系统常采用总线型主从式结构。所谓主从式，即在数个单片机中，有一个是主机，其余的是从机，从机要服从主机的调度、支配。**80C51**单片机的串行口方式**2**和方式**3**适于这种主从式的通信结构。当然采用不同的通信标准时，还需进行相应的电平转换，有时还要对信号进行光电隔离。在实际的多机应用系统中，常采用**RS-485**串行标准总线进行数据传输。



2、通信协议

- 所有从机的**SM2**位置**1**，处于接收地址帧状态。
- 主机发送一地址帧，其中**8**位是地址，第**9**位为地址/数据的区分标志，该位置**1**表示该帧为地址帧。
- 所有从机收到地址帧后，都将接收的地址与本机的地址比较。对于地址相符的从机，使自己的**SM2**位置**0**（以接收主机随后发来的数据帧），并把本站地址发回主机作为应答；对于地址不符的从机，仍保持**SM2=1**，对主机随后发来的数据帧不予理睬。
- 从机发送数据结束后，要发送一帧校验和，并置第**9**位（**TB8**）为**1**，作为从机数据传送结束的标志。

- 主机接收数据时先判断数据接收标志（**RB8**），若 **RB8=1**，表示数据传送结束，并比较此帧校验和，若正确则回送正确信号**00H**，此信号命令该从机复位（即重新等待地址帧）；若校验和出错，则发送**0FFH**，命令该从机重发数据。若接收帧的**RB8=0**，则存数据到缓冲区，并准备接收下帧信息。
- 主机收到从机应答地址后，确认地址是否相符，如果地址不符，发复位信号（数据帧中**TB8=1**）；如果地址相符，则清**TB8**，开始发送数据。
- 从机收到复位命令后回到监听地址状态（**SM2=1**）。否则开始接收数据和命令。

3、应用程序

- 主机发送的地址联络信号为：**00H**，**01H**，**02H**，... ..（即从机设备地址），地址**FFH**为命令各从机复位，即恢复**SM2=1**。
- 主机命令编码为：**01H**，主机命令从机接收数据；**02H**，主机命令从机发送数据。其它都按**02H**对待。

从机状态字节格式为：

位	7	6	5	4	3	2	1	0	
	ERR	0	0	0	0	0	TRDY	RRDY	

RRDY=1：表示从机准备好接收。

TRDY=1：表示从机准备好发送。

ERR=1：表示从机接收的命令是非法的。

程序分为主机程序和从机程序。约定一次传递数据为**16**个字节，以**01H**地址的从机为例。

- 1.由上位机发送1给单片机时，蜂鸣器以400ms频率发声，发2时以200ms频率发声，发3时以100ms频率发声，发4时关闭蜂鸣器。
- 1.以2400bps从计算机发送任一字节数据，当单片机收到该数据后，在此数据前加上一序号然后连同此数据一起发送至计算机，当序号超过255时归零。
- 2.以16进制发送一个0-65536之间的任一数，当单片机收到后在数码管上动态显示出来，波特率自定。
- 3. 用AD以1HZ的频率采集模拟信号，然后转换成数字量，再将其以1200bps发送到计算机，在计算机上显示。
- 4.按下矩阵键盘第一行时以1200bps发送，1，2，3，4，第二行时以2400bps发送5，6，7，8，第三行以4800bps发送，9，10，11，12，第四行以9600pbs 发送，13，14，15，16.

I²C串行总线的组成及工作原理

8.1 I²C串行总线的组成及工作原理

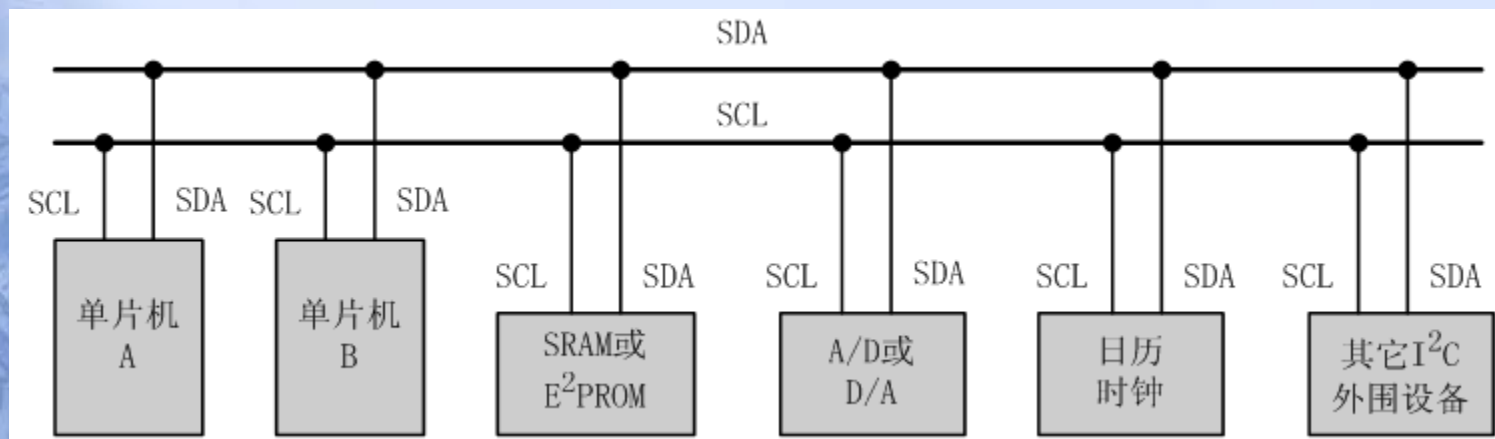
- 采用串行总线技术可以使系统的硬件设计大大简化、系统的体积减小、可靠性提高。同时，系统的更改和扩充极为容易。
- 常用的串行扩展总线有：**I²C**（Inter IC BUS）总线、**单总线**（1—WIRE BUS）、**SPI**（Serial Peripheral Interface）总线及**Microwire/PLUS**等。

本课仅讨论**I²C**串行总线。

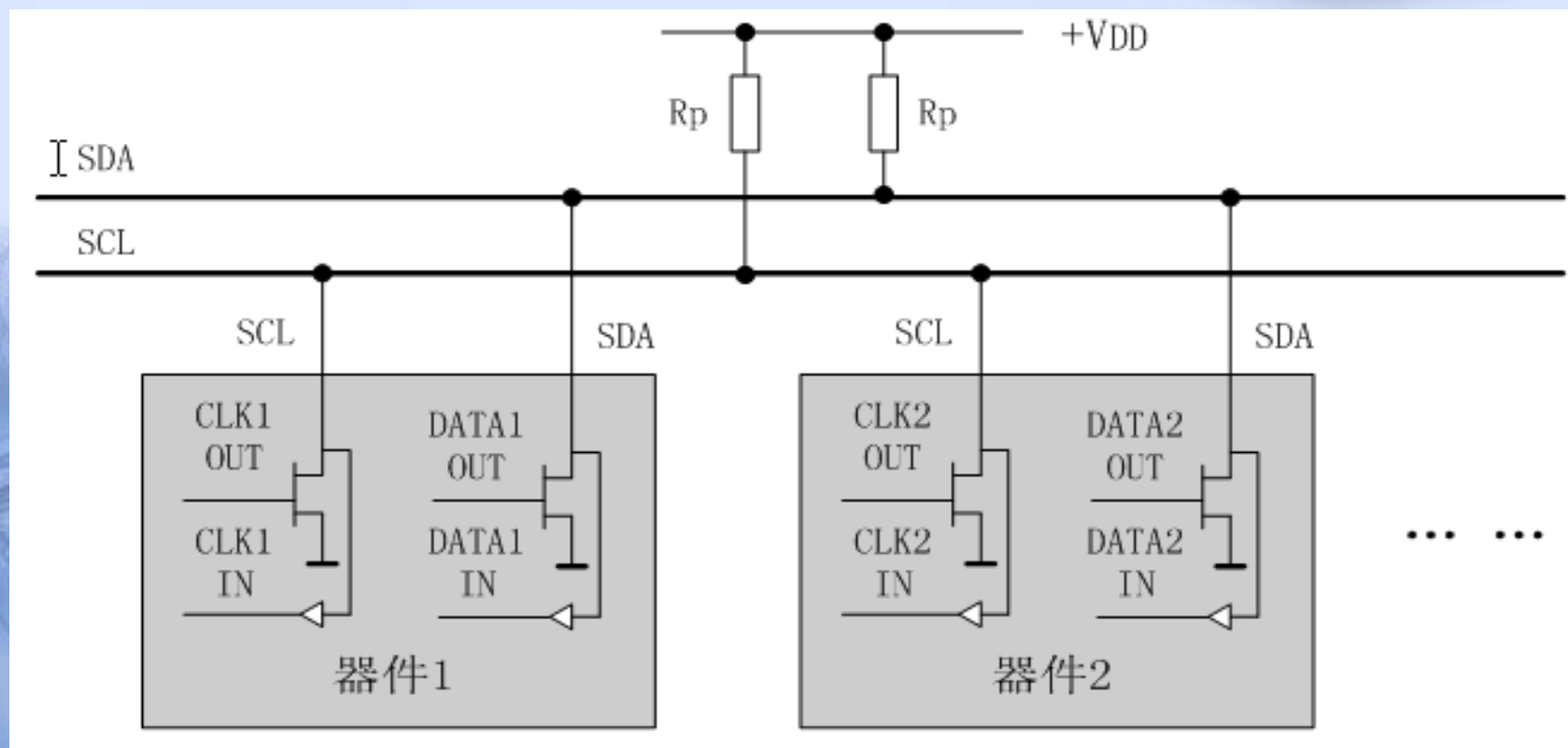
8.1.1 I²C串行总线概述

I²C总线是**PHILIPS**公司推出的一种串行总线，是具备多主机系统所需的包括总线裁决和高低速器件同步功能的高性能串行总线。

I²C总线只有两根双向信号线。一根是数据线**SDA**，另一根是时钟线**SCL**。



I²C总线通过上拉电阻接正电源。当总线空闲时，两根线均为高电平。连到总线上的任一器件输出的低电平，都将使总线的信号变低，即各器件的**SDA**及**SCL**都是线“与”关系。



每个接到I²C总线上的器件都有唯一的地址。主机与其它器件间的数据传送可以由主机发送数据到其它器件，这时主机即为发送器。由总线上接收数据的器件则为接收器。

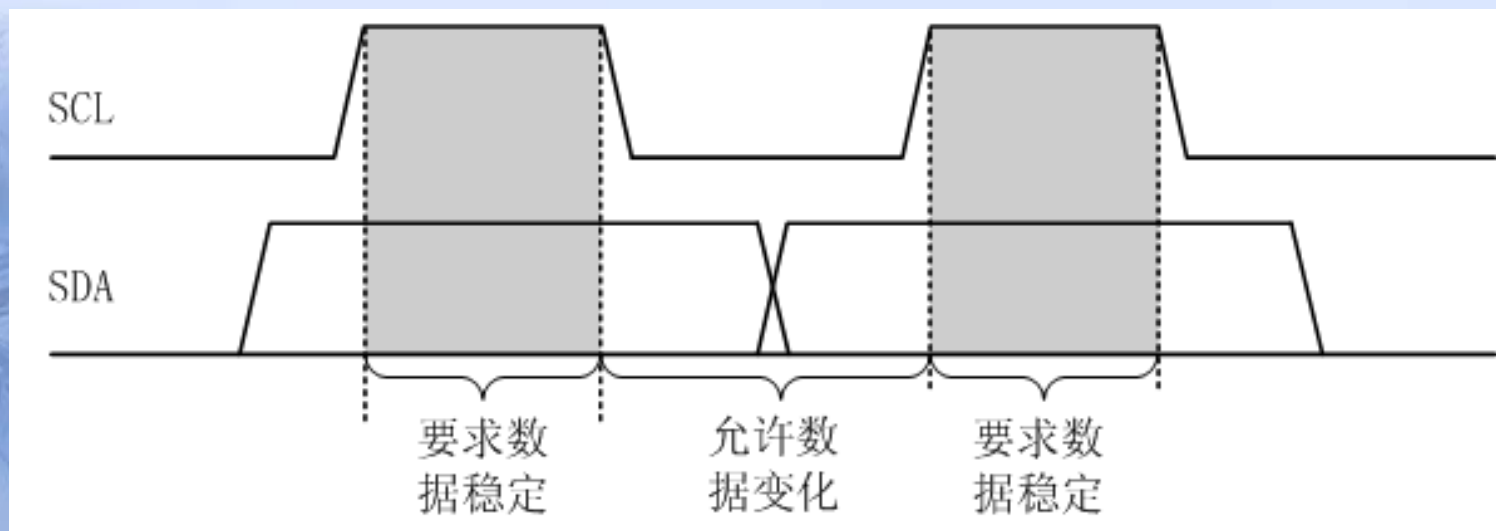
在多主机系统中，可能同时有几个主机企图启动总线传送数据。为了避免混乱，I²C总线要通过总线仲裁，以决定由哪一台主机控制总线。

在80C51单片机应用系统的串行总线扩展中，我们经常遇到的是以80C51单片机为主机，其它接口器件为从机的单主机情况。

8.1.2 I²C总线的数据传送

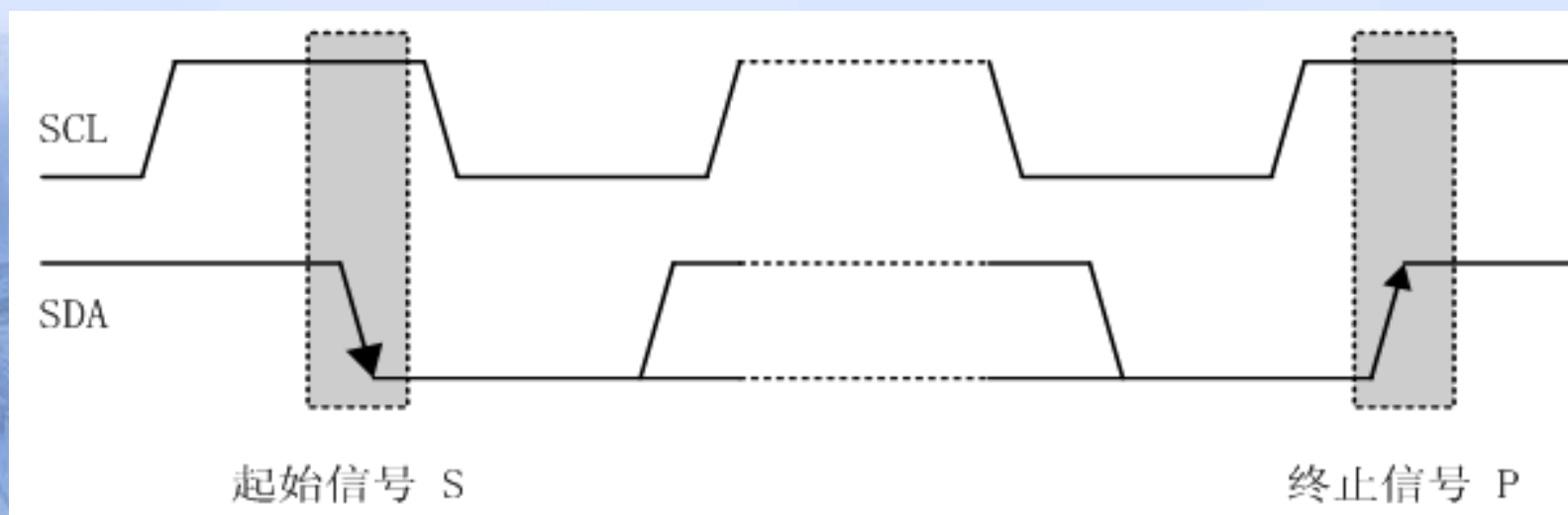
一、数据位的有效性规定

I²C总线进行数据传送时，**时钟信号为高电平期间**，数据线上的数据必须保持稳定，只有在**时钟线上的信号为低电平期间**，数据线上的高电平或低电平状态才允许变化。



二、起始和终止信号

SCL线为高电平期间，**SDA**线由高电平向低电平的变化表示起始信号；**SCL**线为高电平期间，**SDA**线由低电平向高电平的变化表示终止信号。



起始和终止信号都是由主机发出的，在起始信号产生后，总线就处于被占用的状态；在终止信号产生后，总线就处于空闲状态。

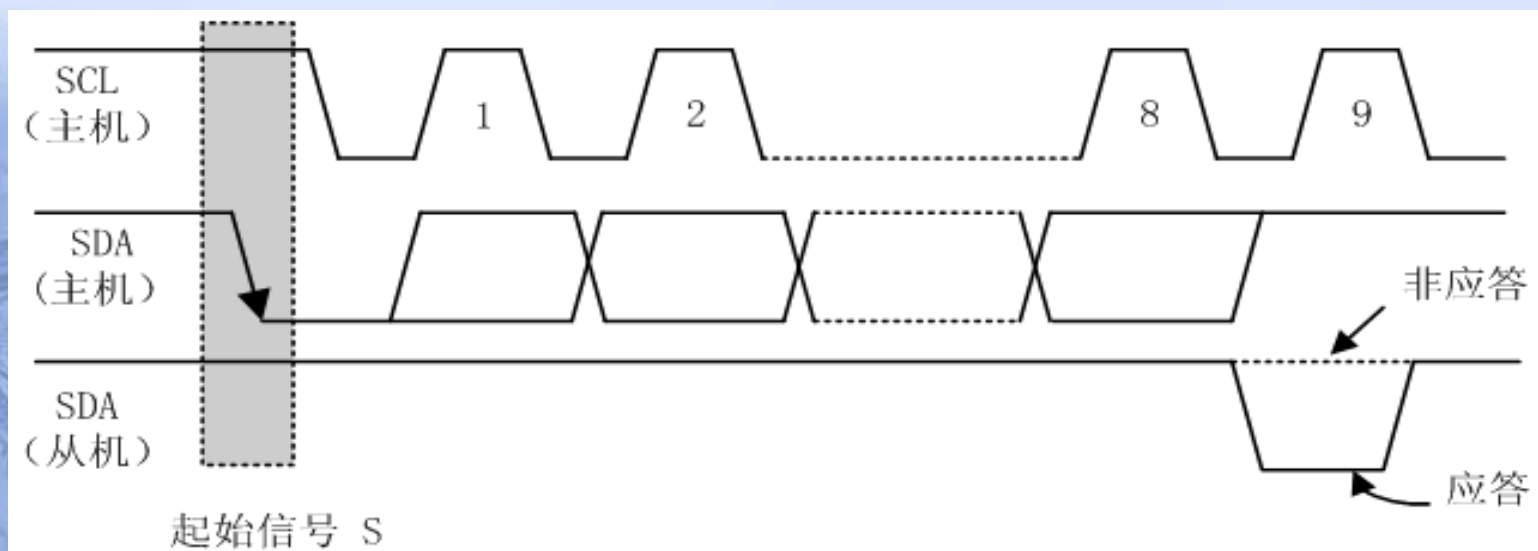
连接到I²C总线上的器件，若具有I²C总线的硬件接口，则很容易检测到起始和终止信号。

接收器件收到一个完整的数据字节后，有可能需要完成一些其它工作，如处理内部中断服务等，可能无法立刻接收下一个字节，这时接收器件可以将**SCL**线拉成低电平，从而使主机处于等待状态。直到接收器件准备好接收下一个字节时，再释放**SCL**线使之成为高电平，从而使数据传送可以继续进行。

三、数据传送格式

(1) 字节传送与应答

每一个字节必须保证是**8**位长度。数据传送时，先传送最高位（**MSB**），每一个被传送的字节后面都必须跟随一位应答位（即一帧共有**9**位）。



由于某种原因从机不对主机寻址信号应答时（如从机正在进行实时性的处理工作而无法接收总线上的数据），它必须将数据线置于高电平，而由主机产生一个终止信号以结束总线的数据传送。

如果从机对主机进行了应答，但在数据传送一段时间后无法继续接收更多的数据时，从机可以通过对无法接收的第一个数据字节的“非应答”通知主机，主机则应发出终止信号以结束数据的继续传送。

当主机接收数据时，它收到最后一个数据字节后，必须向从机发出一个结束传送的信号。这个信号是由对从机的“非应答”来实现的。然后，从机释放**SDA**线，以允许主机产生终止信号。

(2) 数据帧格式

I²C总线上传送的数据信号是广义的，既包括地址信号，又包括真正的数据信号。

在起始信号后必须传送一个从机的地址（7位），第8位是数据的传送方向位（R/T），用“0”表示主机发送数据（T），“1”表示主机接收数据（R）。每次数据传送总是由主机产生的终止信号结束。但是，若主机希望继续占用总线进行新的数据传送，则可以不产生终止信号，马上再次发出起始信号对另一从机进行寻址。

在总线的一次数据传送过程中，可以有以下几种组合方式：

a、主机向从机发送数据，数据传送方向在整个传送过程中不变：



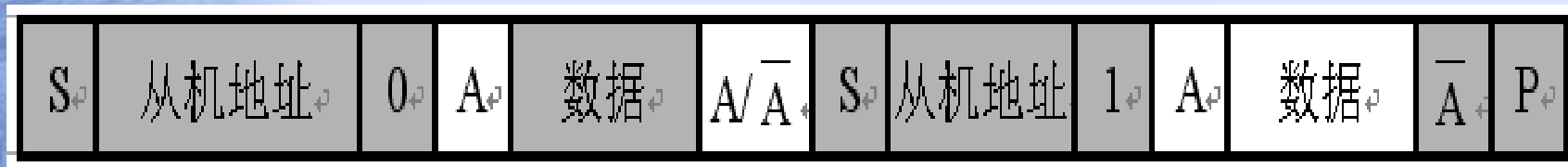
注：有阴影部分表示数据由主机向从机传送，无阴影部分则表示数据由从机向主机传送。

A表示应答，**A**非表示非应答（高电平）。**S**表示起始信号，**P**表示终止信号。。

b、主机在第一个字节后，立即从从机读数据



c、在传送过程中，当需要改变传送方向时，起始信号和从机地址都被重复产生一次，但两次读/写方向位正好反相。



四、总线的寻址

I²C总线协议有明确的规定：采用**7位**的寻址字节（寻址字节是起始信号后的第一个字节）。

（1）寻址字节的位定义

位：	7 ₀	6 ₀	5 ₀	4 ₀	3 ₀	2 ₀	1 ₀	0 ₀	
	从机地址 ₀							R/ \overline{W}	

D7~D1位组成从机的地址。**D0**位是数据传送方向位，为“**0**”时表示主机向从机写数据，为“**1**”时表示主机由从机读数据。

■ **主机发送地址**时，总线上的每个从机都将这7位地址码与自己的地址进行比较，如果相同，则认为自己正被主机寻址，根据**R/T**位将自己确定为发送器或接收器。

■ **从机的地址**由固定部分和可编程部分组成。在一个系统中可能希望接入多个相同的从机，从机地址中可编程部分决定了可接入总线该类器件的最大数目。如一个从机的7位寻址位有4位是固定位，3位是可编程位，这时仅能寻址8个同样的器件，即可以有8个同样的器件接入到该**I2C**总线系统中。

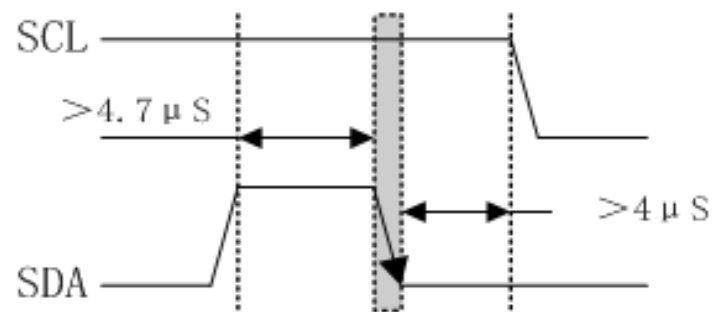
8.2 80C51单片机I²C串行总线器件的接口

8.2.1 总线数据传送的模拟

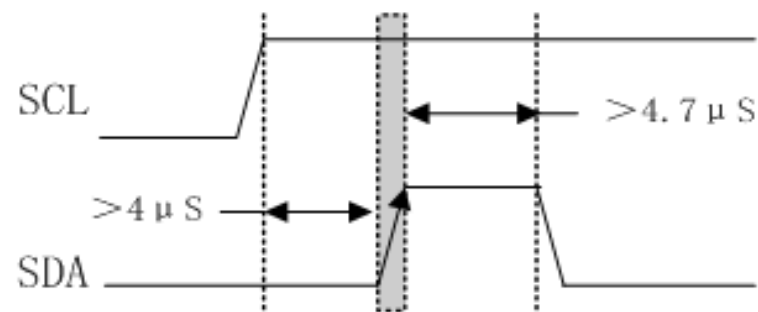
主机可以采用不带I²C总线接口的单片机，如**80C51**、**AT89C2051**等单片机，利用软件实现I²C总线的数据传送，即软件与硬件结合的信号模拟。

一、典型信号模拟

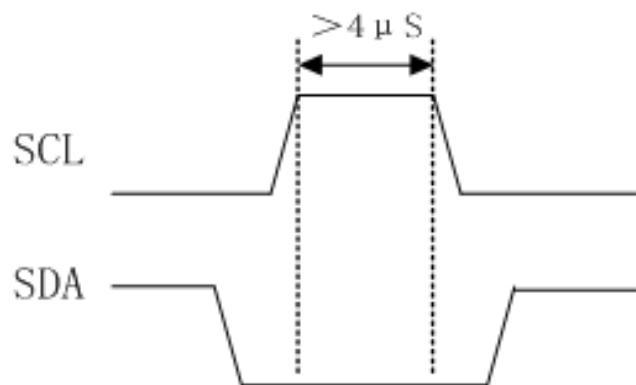
为了保证数据传送的可靠性，标准的I²C总线的数据传送有严格的时序要求。I²C总线的起始信号、终止信号、发送“0”及发送“1”的模拟时序：



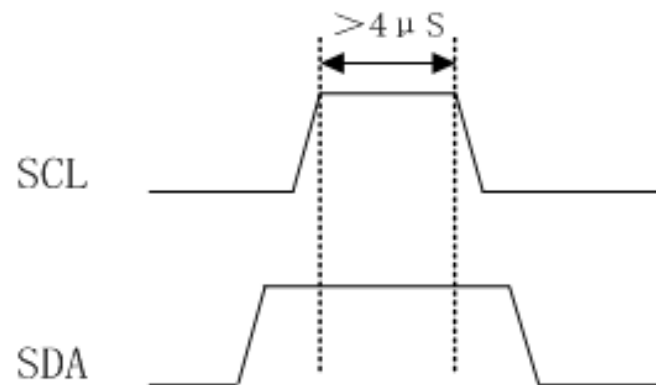
起始信号 S



终止信号 P



应答/“0”



非应答/“1”

二、典型信号模拟子程序

(1) 起始信号

```
Void T2CStart(void)
```

```
{    SDA = 1;  
      SomeNop( );  
      SCL = 1;  
      SomeNop( );  
      SDA = 0;  
      SomeNop( );  
}
```

(2) 终止信号

```
void I2cStop(void)
```

```
{
```

```
    SDA = 0;
```

```
    SomeNop( );
```

```
    SCL = 1;
```

```
    SomeNop( );
```

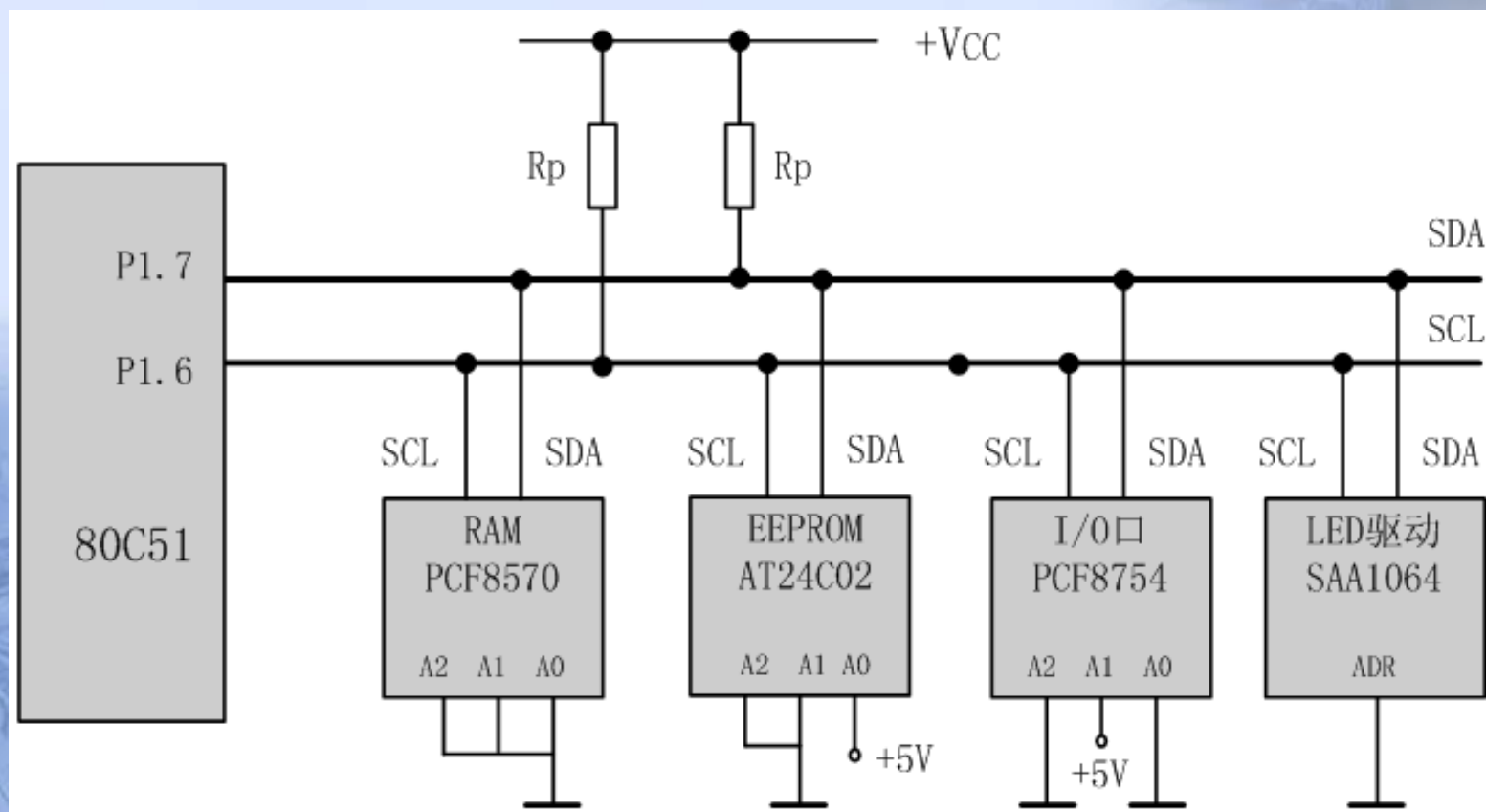
```
    SDA = 1;
```

```
    SomeNop( );
```

```
}
```

8.2.2 I²C总线器件的扩展

一、扩展电路



二、串行E²PROM的扩展

(1) 串行E²PROM典型产品

ATMEL公司的AT24C系列:

AT24C01: 128字节 (128×8位) ;

AT24C02: 256字节 (256×8位) ;

AT24C04: 512字节 (512×8位)

AT24C08: 1K字节 (1K×8位) ;

AT24C16: 2K字节 (2K×8位) ;

(2) 写入过程

AT24C系列**E2PROM**芯片地址的固定部分为**1010**，**A2**、**A1**、**A0**引脚接高、低电平后得到确定的**3**位编码。形成的**7**位编码即为该器件的地址码。

单片机进行写操作时，首先发送该器件的**7**位地址码和写方向位“**0**”（共**8**位，即一个字节），发送完后释放**SDA**线并在**SCL**线上产生第**9**个时钟信号。被选中的存储器器件在确认是自己的地址后，在**SDA**线上产生一个应答信号作为相应，单片机收到应答后就可以传送数据了。

传送数据时，单片机首先发送一个字节的被写入器件的存储区的首地址，收到存储器器件的应答后，单片机就逐个发送各数据字节，但每发送一个字节后都要等待应答。

AT24C系列器件片内地址在接收到每一个数据字节地址后自动加1，在芯片的“一次装载字节数”（不同芯片字节数不同）限度内，只需输入首地址。装载字节数超过芯片的“一次装载字节数”时，数据地址将“上卷”，前面的数据将被覆盖。

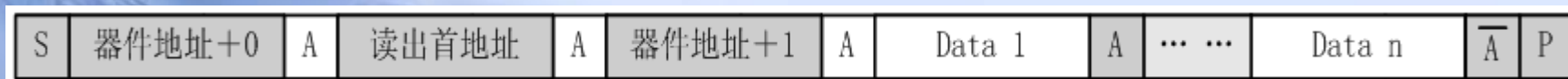
当要写入的数据传送完后，单片机应发出终止信号以结束写入操作。写入n个字节的数据格式：

S	器件地址+0	A	写入首地址	A	Data 1	A	Data n	A	P
---	--------	---	-------	---	--------	---	--------	--------	---	---

(3) 读出过程

单片机先发送该器件的7位地址码和写方向位“0”（“伪写”），发送完后释放SDA线并在SCL线上产生第9个时钟信号。被选中的存储器器件在确认是自己的地址后，在SDA线上产生一个应答信号作为回应。

然后，再发一个字节的要读出器件的存储区的首地址，收到应答后，单片机要重复一次起始信号并发出器件地址和读方向位（“1”），收到器件应答后就可以读出数据字节，每读出一个字节，单片机都要回复应答信号。当最后一个字节数据读完后，单片机应返回以“非应答”（高电平），并发出终止信号以结束读出操作。



移位操作

- 左移时最低位补0，最高位移入PSW的CY位
- 右移时最高位保持原数，最低位移除。