

第7章 可编程接口芯片

- 随着集成电路技术的发展，接口电路早已集成化，并出现了许多可编程接口芯片。
- 为了具有通用性，这些芯片通常被设计成具有多项功能或多种工作方式，用户在使用时通过编程选择自己所需的功能或工作方式。
- 尽管早期的可编程接口芯片在现代微机中已不再独立出现，但是本章在介绍时，还是以它们为讲解对象，因为这些是理解现代微机所用芯片的基础。
- 此外，在单片机等微机应用系统或输入/输出设备中还常常用到它们。

7.1 可编程并行输入/输出接口芯片 8255A

- Intel 8255A是一种通用的可编程序并行I/O接口芯片，是为Intel 8080/8085系列微处理器设计的，也可用于其他系列的微机系统。

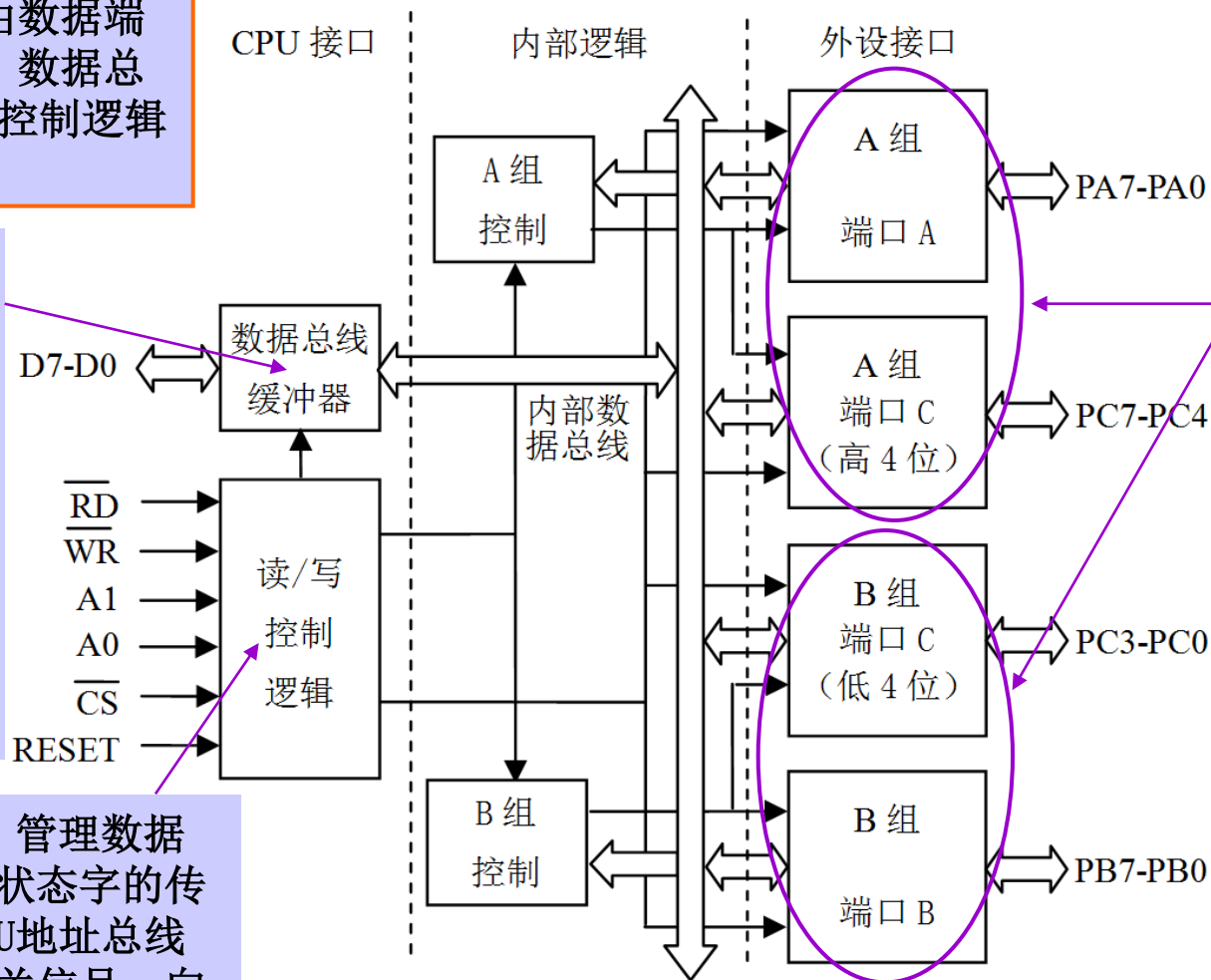
7.1.1 8255A 的内部结构及引脚功能

8255A为40引脚、双列直插封装。内部结构由数据端口、组控制电路、数据总线缓冲器、读/写控制逻辑四部分组成。

数据总线缓冲器:

三态8位双向缓冲器，D7-D0同系统数据总线相连。CPU通过执行输入/输出指令来实现对缓冲器发送或接收数据，传送控制字和状态字。

读/写控制逻辑: 管理数据信息、控制字和状态字的传送，接收来自CPU地址总线和控制总线的有关信号，向8255A的A、B两组控制部件发送命令。



A组和B组的控制电路: 3个端口分成两组来管理。A口及C口高4位为A组，B口及C口低4位为B组。两组分别设有控制电路，每个控制组都接收来自读/写控制逻辑的“命令”，接收来自内部数据总线的“控制字”，并向与其相连的端口发出适当的控制信号。

数据端口: 3个8位数据端口：端口A、端口B和端口C。

图 7.1.1 8255A 的引脚及内部结构示意图

8255A端口功能选择

表7.1.1 8255A端口功能选择

操作类型	\overline{CS}	A1	A0	\overline{RD}	\overline{WR}	功能
读操作	0	0	0	0	1	数据总线←端口A
	0	0	1	0	1	数据总线←端口B
	0	1	0	0	1	数据总线←端口C
写操作	0	0	0	1	0	端口A←数据总线
	0	0	1	1	0	端口B←数据总线
	0	1	0	1	0	端口C←数据总线
	0	1	1	1	0	控制寄存器←数据总线
无操作	0	1	1	0	1	无意义
与总线 “脱开”	1	×	×	×	×	数据总线三态
	0	×	×	1	1	数据总线三态

A1	A0	选择端口
0	0	A口（数据口）
0	1	B口（数据口）
1	0	C口（数据口）
1	1	控制口

7.1.2 8255A的控制字-方式选择控制字

8255A有两种控制字。一个是方式选择控制字，另一个是对C口进行置位或复位的控制字。

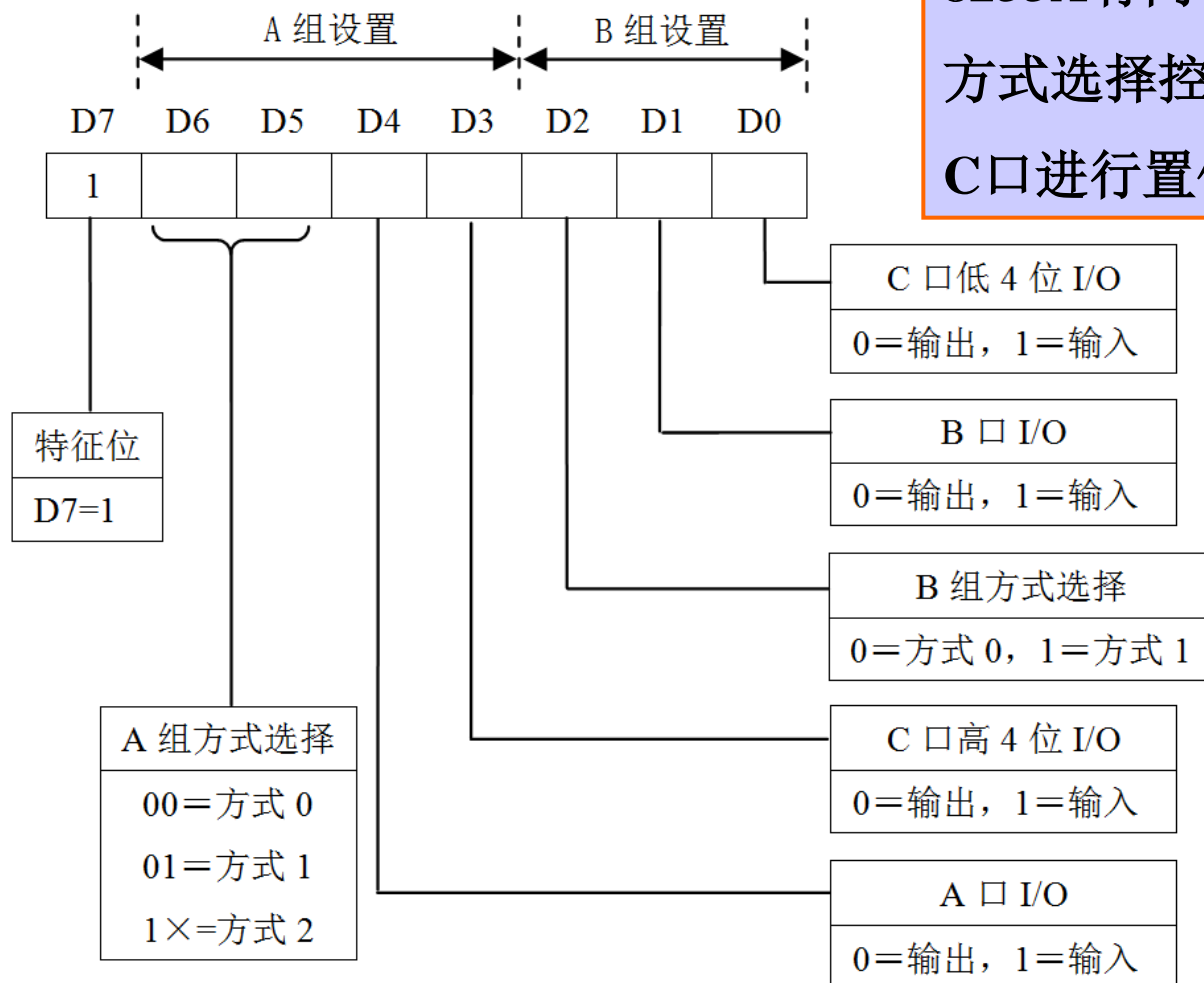


图 7.1.2 8255A 的方式选择控制字

7.1.2 8255A的控制字

对C口进行置位或复位的控制字

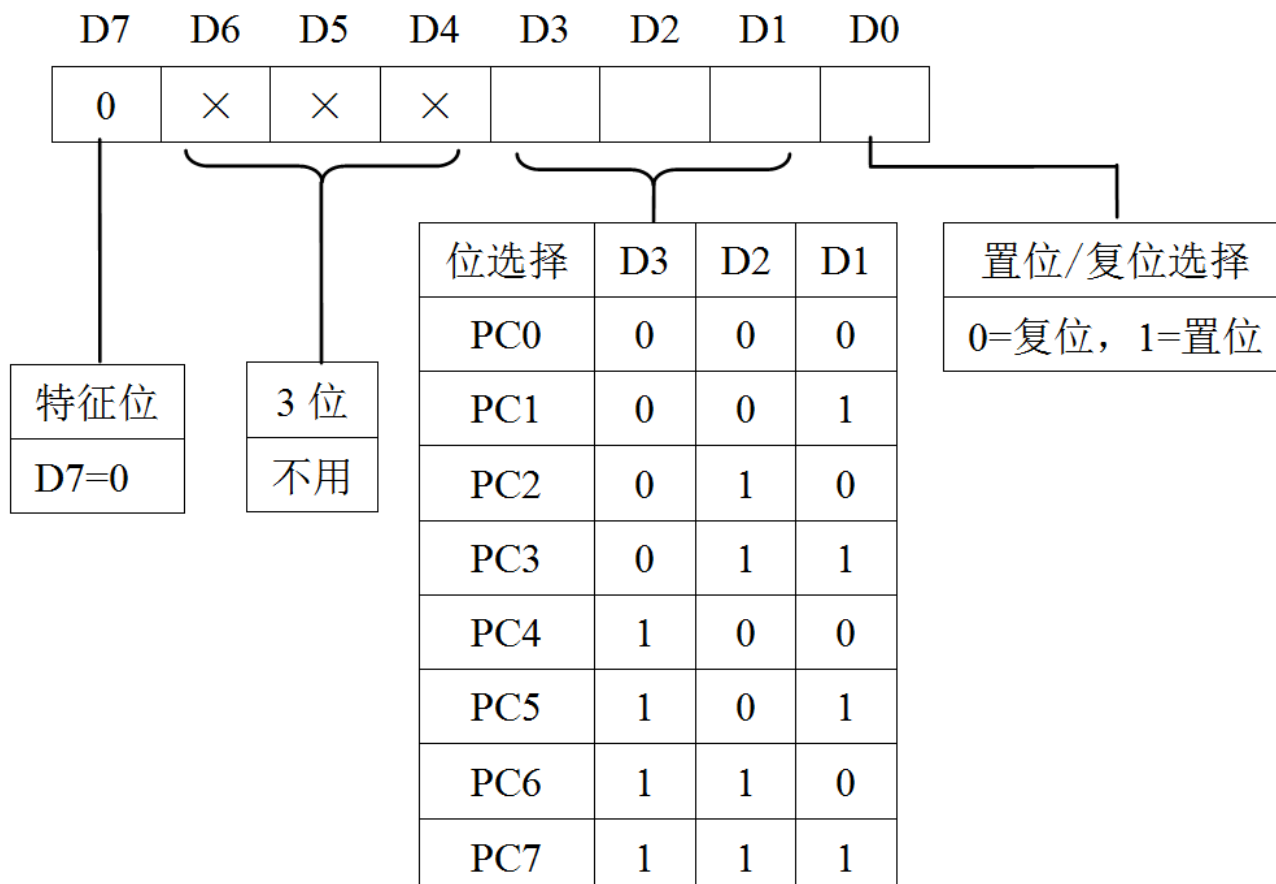
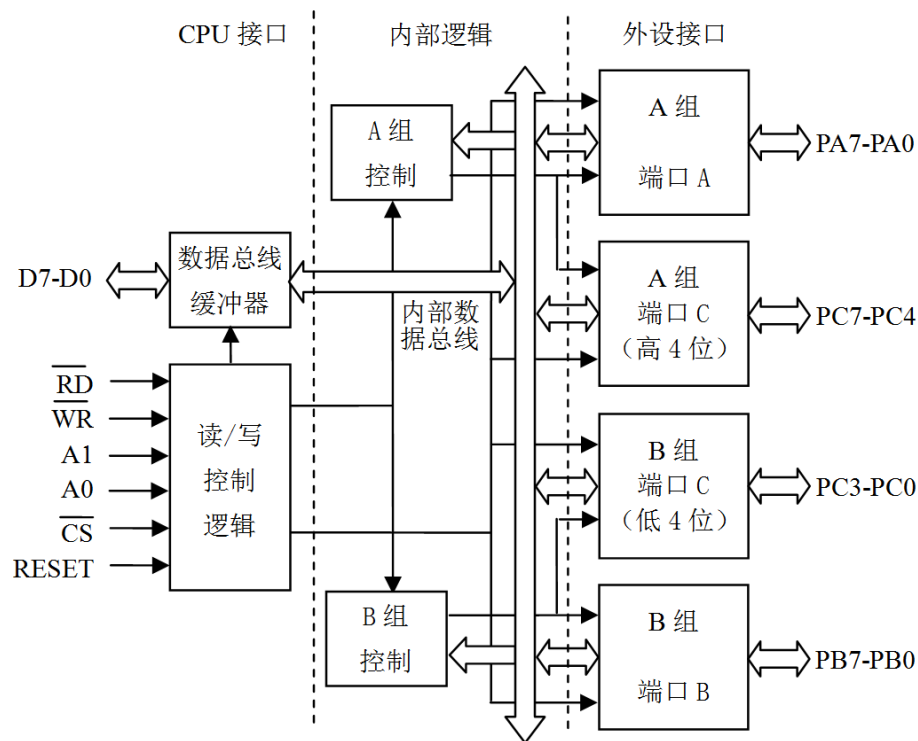


图 7.1.3 8255A 对端口 C 置位/复位的控制字

7.1.3 8255A的工作方式

- **3种基本工作方式：**
- **方式0：**基本的输入/输出方式。
适用**A、B、C口**。
- **方式1：**选通的输入/输出方式。
适用**A、B口**。
- **方式2：**双向选通输入/输出方式。适用**A口**。

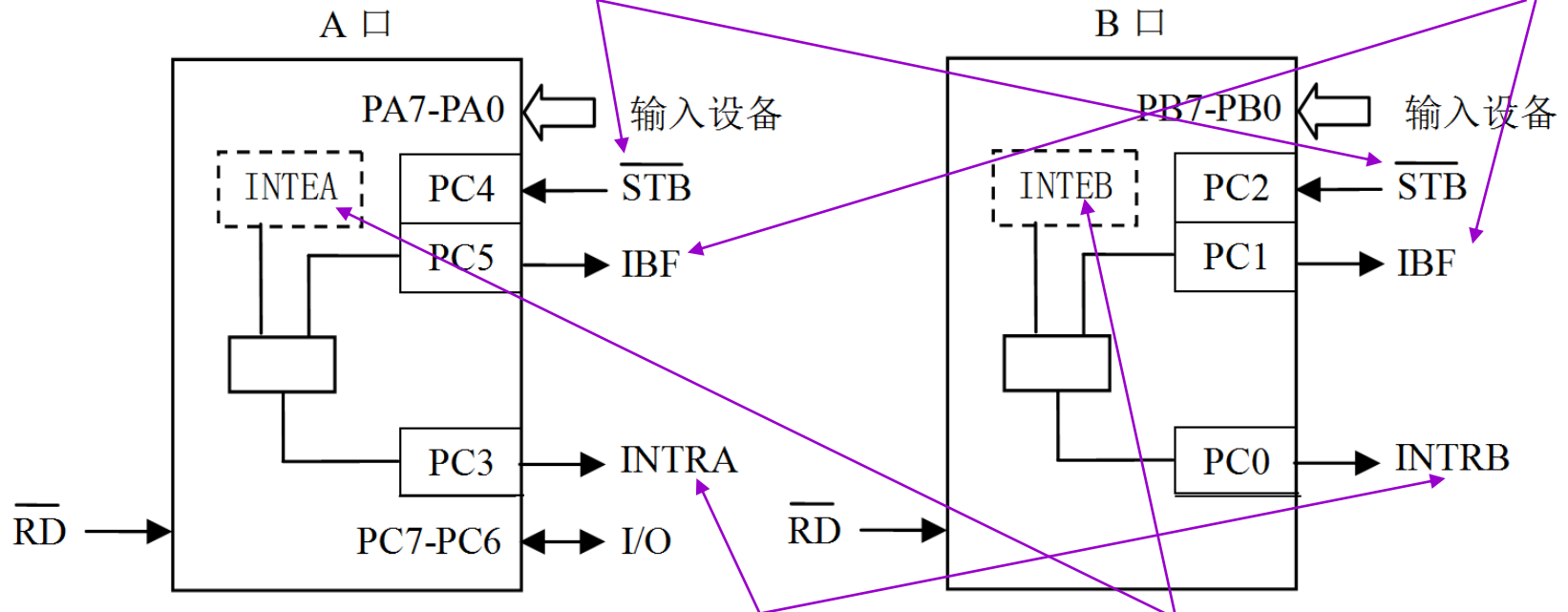


- 在方式0下，C口的高4位和低4位以及A口、B口都可以独立地设置为基本的输入口或输出口。4个口的输入/输出可以有16种组合方式。

方式1输入的逻辑功能结构

STB: 输入的选通信号，低电平有效。由外设提供，为低电平时，把输入的数据送入A端口或B端口的数据锁存器。

IBF: 输入缓冲器满信号，高电平有效。由8255A输出，有效时，用以通知外部设备输入的数据已写入缓冲器。



INTR: 中断请求信号，高电平有效。当 $\overline{\text{STB}}$ 、IBF和INTE都为高电平时，表明数据锁存器内已写入了数据，使INTR成为高电平输出。

CPU响应中断执行IN指令后，在控制下从8255A中读取数据时，RD的下降沿使INTR复位，RD的上升沿又使IBF复位，使外设知道可以进行下一字节输入了。

INTE: **中断允许信号**。
A端口用PC4位的置位/复位控制，
B端口用PC2位的置位/复位控制。
只有当PC4或PC2置“1”时，才允许对应的端口送出中断请求。

方式1输入时序

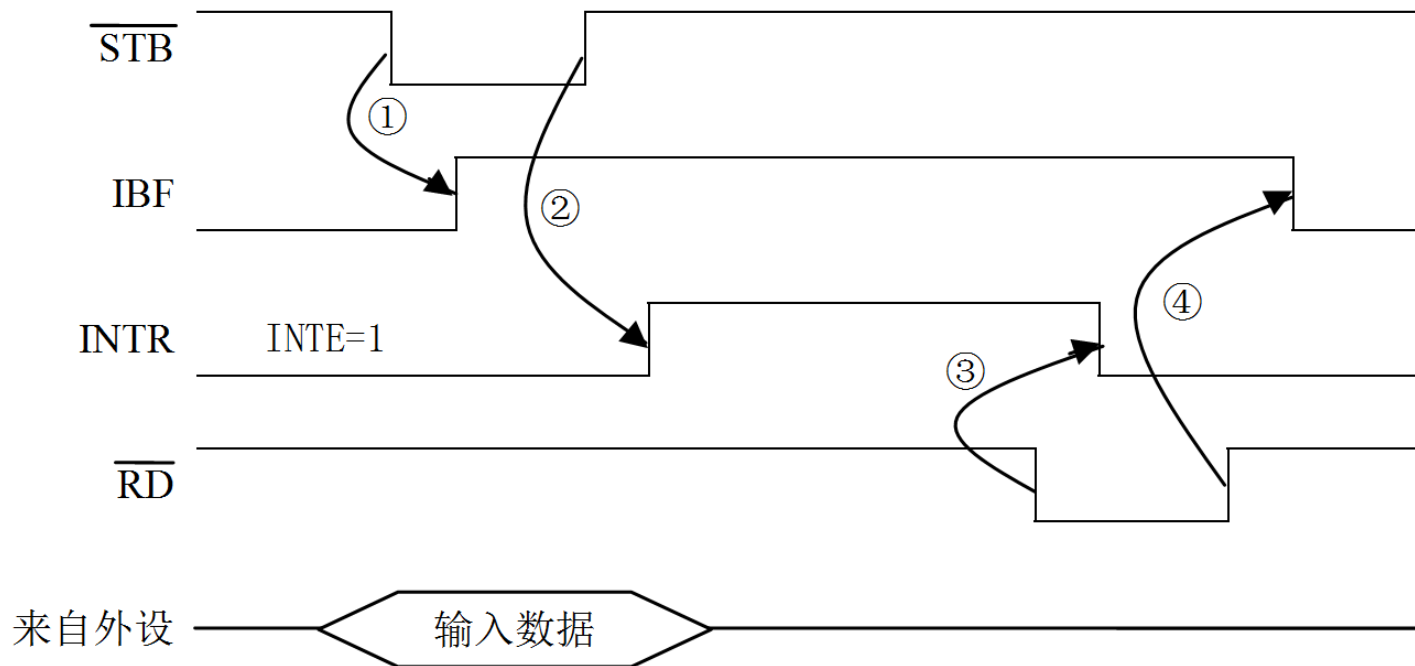
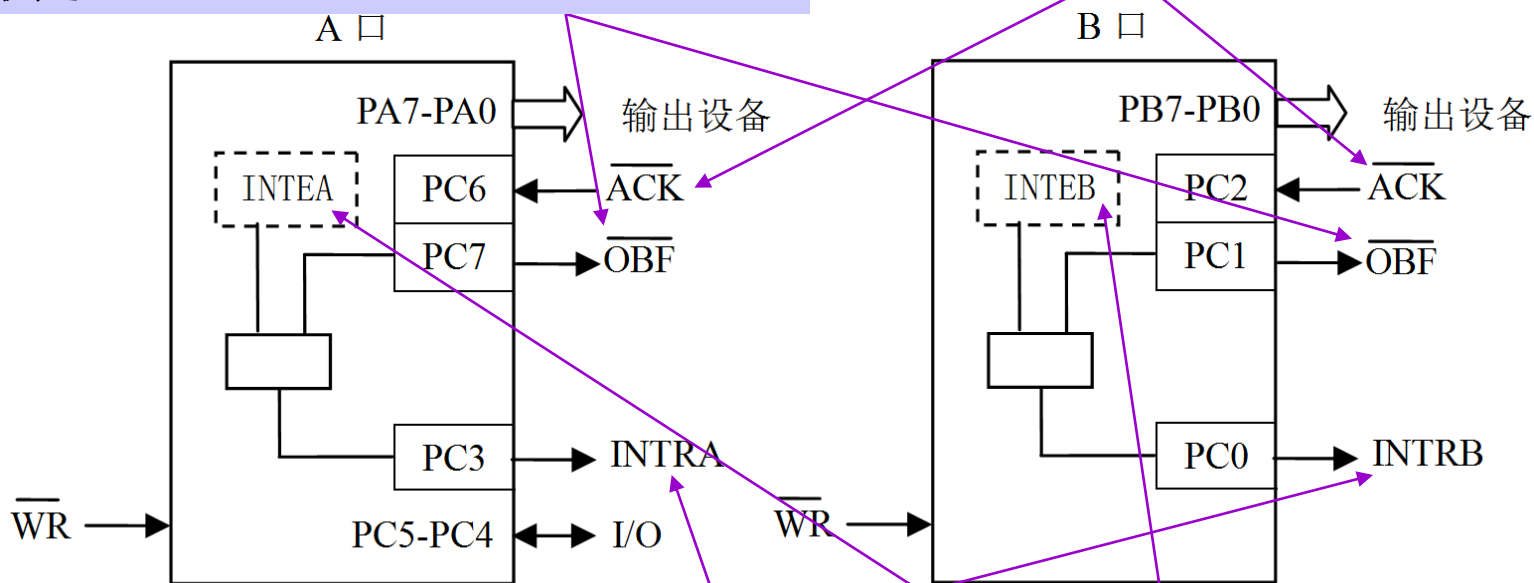


图 7.1.5 方式 1 输入时序

方式1输出的逻辑功能结构

$\overline{\text{OBF}}$: **输出缓冲器满信号**，低电平有效。由8255A输出，当其有效时，表示CPU已经将数据输出到指定的端口，通知外设可以将数据取走。

$\overline{\text{ACK}}$: **响应信号**，低电平有效。由外设送来，有效时表示8255A的数据已经被外设所接收。



INTR: 中断请求信号，高电平有效。当外设接收了由CPU送给8255A的数据后，8255A就用INTR端向CPU发出中断请求，请求CPU再输出后面的数据。INTR是当 $\overline{\text{OBF}}$ 、 $\overline{\text{ACK}}$ 和INTE都为高电平时，才能被置成高电平。由的 $\overline{\text{WR}}$ 下降沿清除。

能结构

INTE: 中断允许信号。
A口的INTE由PC6置/复位，
B口的INTE由PC2置/复位。

方式1输出时序

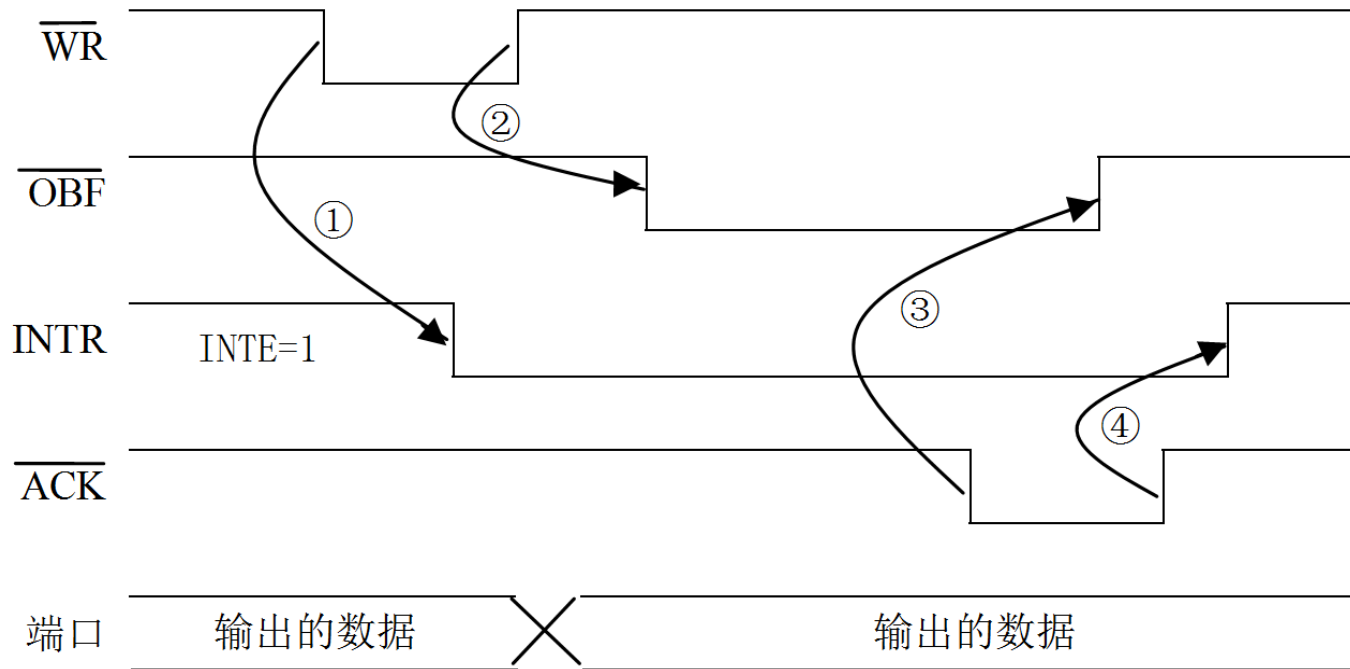
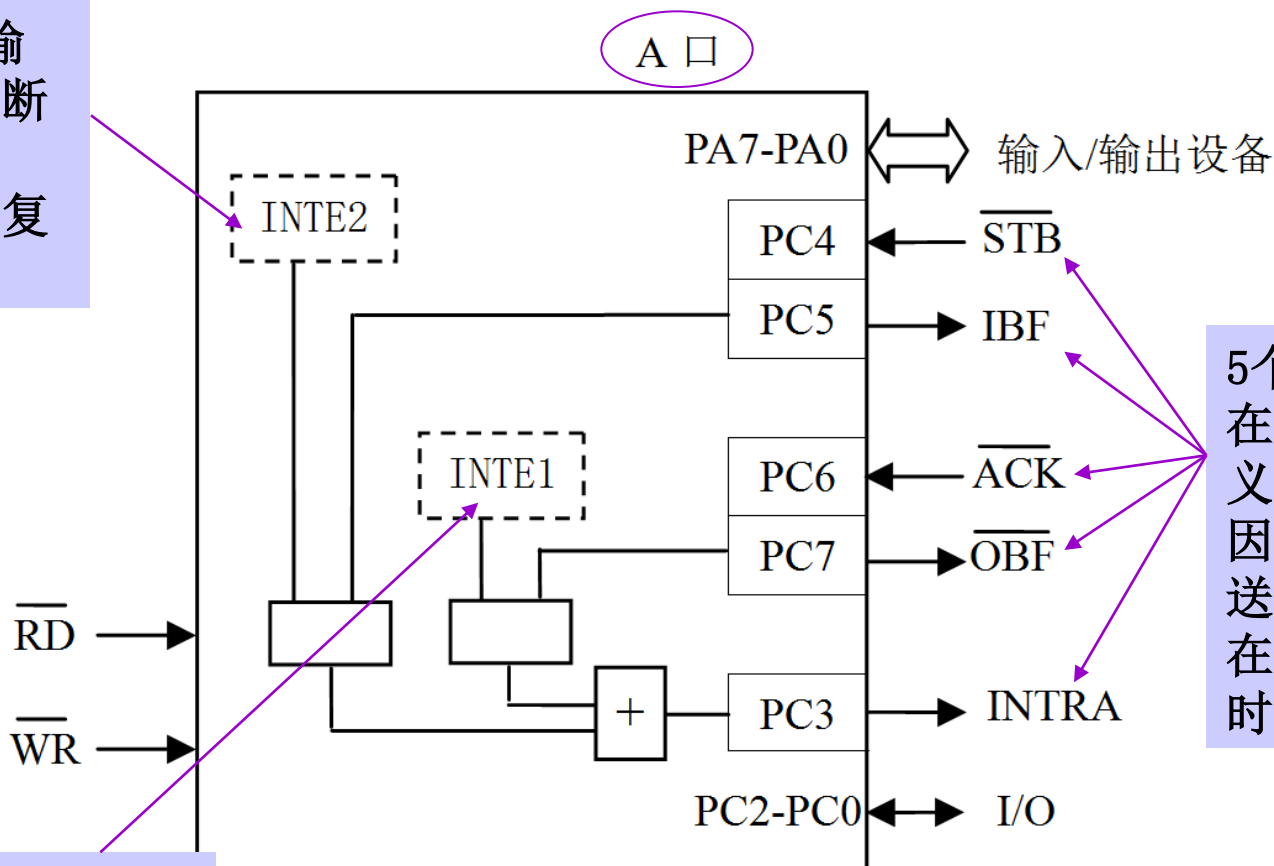


图 7.1.7 方式 1 输出时序

方式2的逻辑功能结构

INTE2是与输入相关的中断允许，由对PC4的置位/复位控制。



5个联络信号与在方式1中的含义基本相同，因为是双向传送，所以INTRA在输入或输出时都可以产生。

INTE1是与输出相关的中断允许，由对PC6的置位/复位控制。

图 7.1.8 方式 2 的逻辑功能结构

7.1.4 8255A 的应用举例

- 例 7.1.1 8255A 方式 0 应用：8255A 产生波形接口电路。
- 利用 8255A 在方式 0 下工作，使其在 PC 0、PC 3 引脚产生如图所示的波形，试编写相应程序段。设 8255A 各端口地址分别设为 60H、61H、62H 和 63H，波形延时时间可调用延时 1ms 子程序（D1ms）实现。

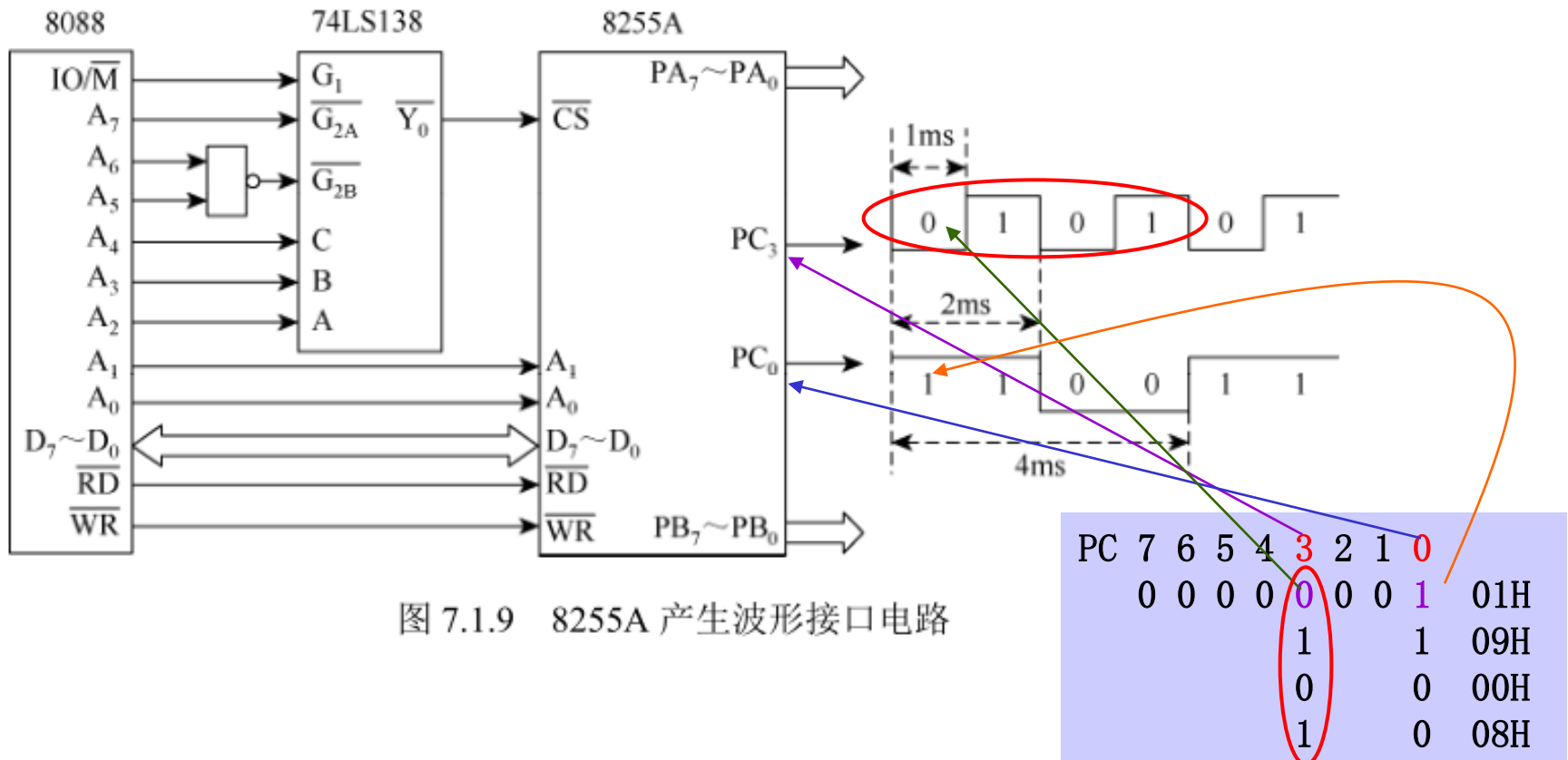


图 7.1.9 8255A 产生波形接口电路

程序段

```
START: MOV    AL, 80H          ; 送各口方式0输出控制字
        OUT    63H, AL


---


A1:     MOV    AL, 01H
        OUT    62H, AL
        CALL   D1ms


---


        MOV    AL, 09H
        OUT    62H, AL
        CALL   D1ms


---


        MOV    AL, 00H
        OUT    62H, AL
        CALL   D1ms


---


        MOV    AL, 08H
        OUT    62H, AL
        CALL   D1ms


---


        JMP    A1
```

7.2 可编程计数器/定时器 8253

• 7.2.1 8253的基本功能

1. 3个独立的16位计数器，能够进行3个16位的独立计数。
2. 每一个计数器具有六种工作方式。
3. 能进行二进制/十进制计数（减法计数）。所谓十进制计数，是指BCD码计数，每个计数器可表示4位十进制数的BCD码，每来一个计数脉冲，按照十进制数减1规律进行计数。
4. 计数频率为0~2MHz。
5. 可作计数器或定时器。

7.2.2 8253 的内部结构及引脚功能

初值计算:

计数方式: 计数值 = 要求的计数次数

定时方式: 计数值 = OUT时间 ÷ CLK周期

A1 A0 选择端口

0 0 计时器0 (数据口)

0 1 计时器1 (数据口)

1 0 计时器2 (数据口)

1 1 控制口

计数值读写格式:

1. 只读写低8位, 高8位清0

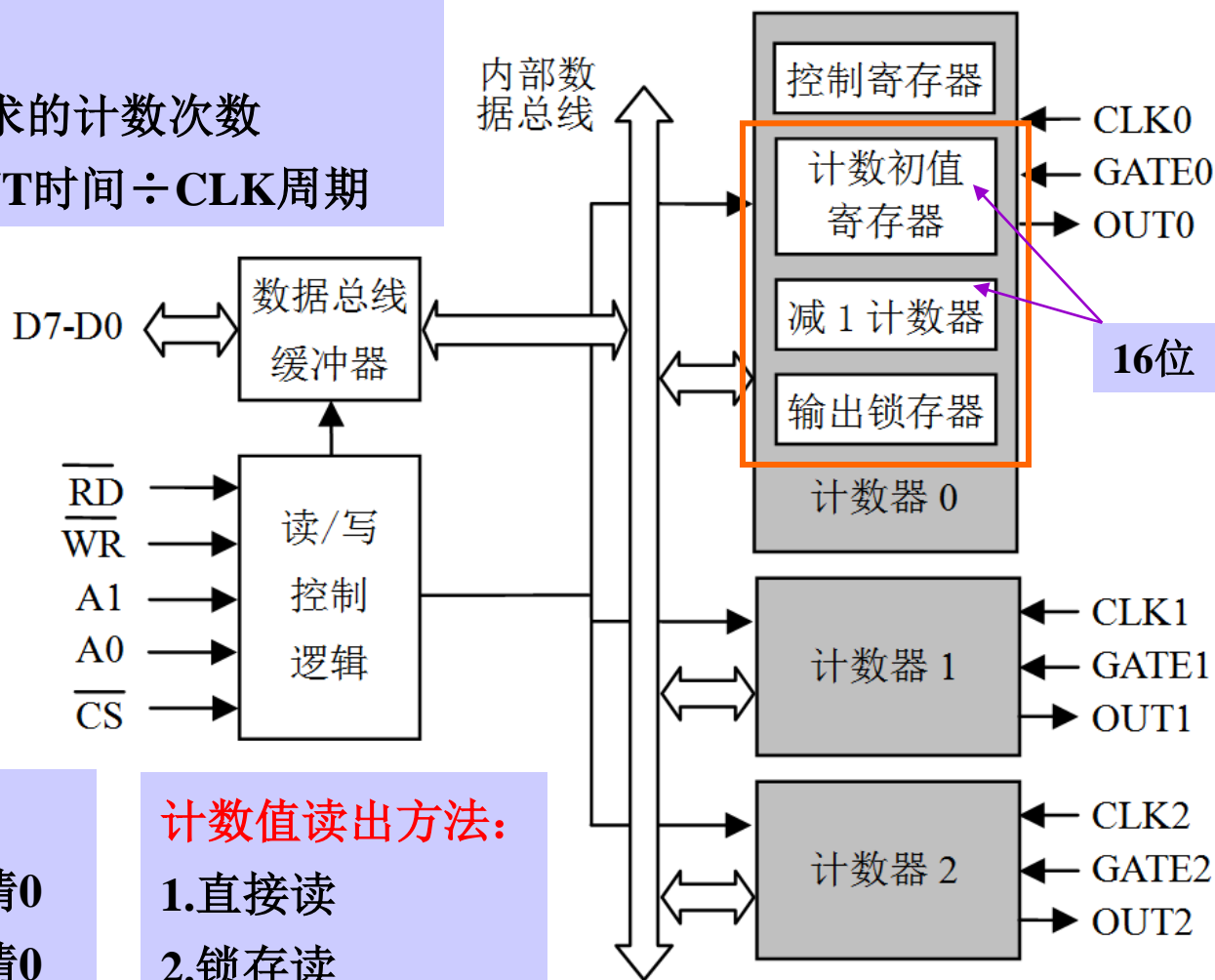
2. 只读写高8位, 低8位清0

3. 先低8位, 后高8位

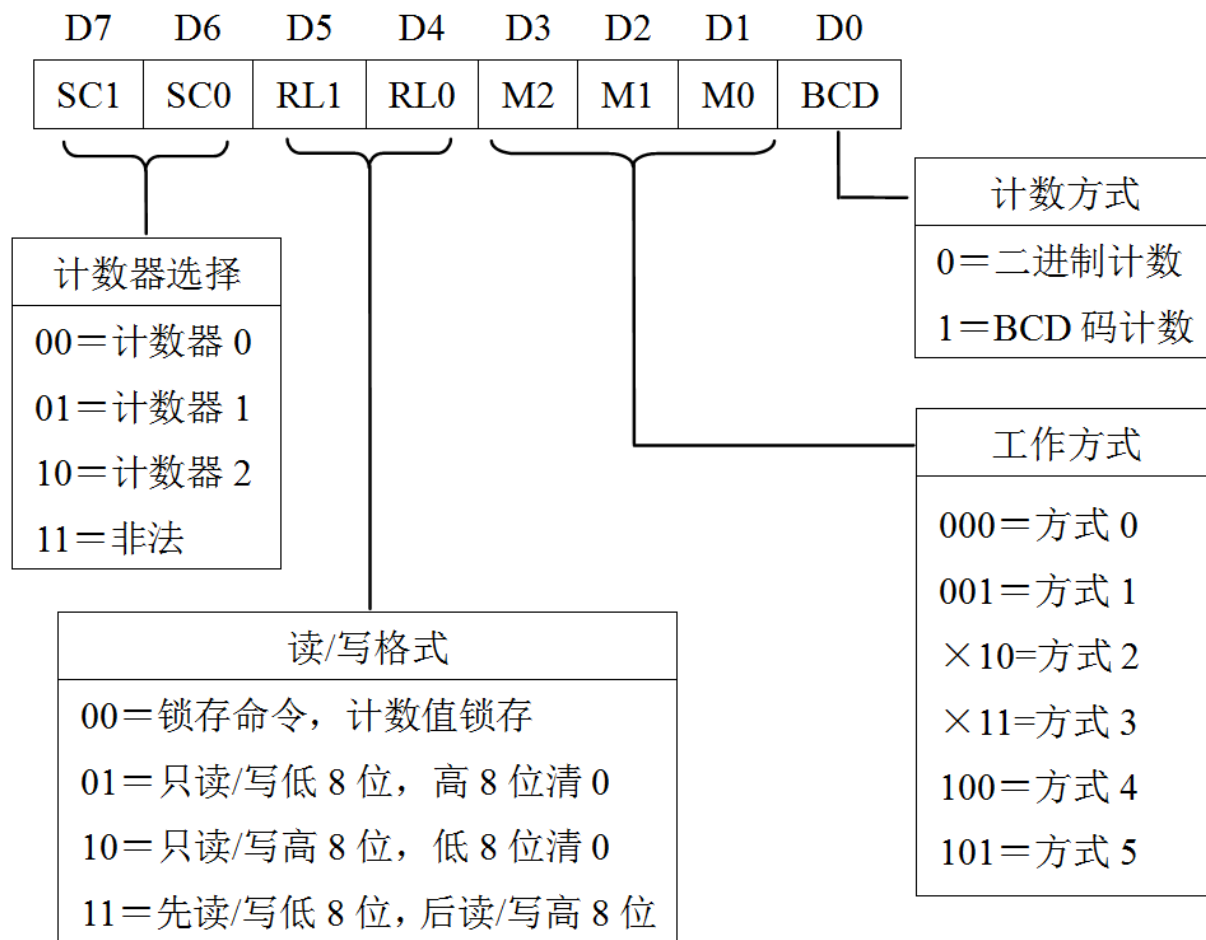
计数值读出方法:

1. 直接读


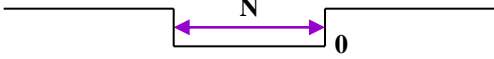
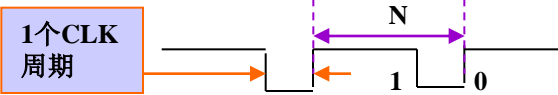
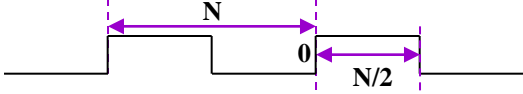
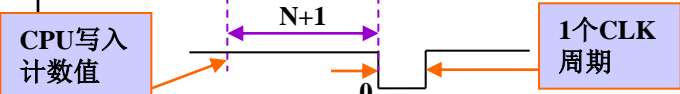
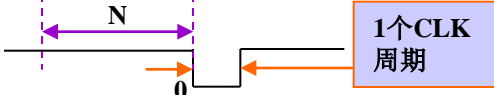
2. 锁存读



7.2.3 8253的控制字



7.2.4 8253的工作方式

方式	名称	OUT输出波形	启动方式	计数值使用
0	计数结束中断方式		软	一次有效
1	可编程单稳方式		硬	多次有效
2	脉冲频率发生器方式		软、硬	重复有效
3	方波发生器方式		软、硬	重复有效
4	软件触发选通方式		软	一次有效
5	硬件触发选通方式		硬	多次有效

启动方式:

软件启动: CPU写入计数值

硬件启动: GATE上跳沿

计数值极限:

方式	最大值	最小值
0,1,4,5	0	1
2,3	0	2

十进制: 10000
二进制: 65536

7.2.5 8253 的应用举例

- 例 7.2.1 8253 的初始化设计。
- 设某8253通道1工作于方式0，按BCD方式计数，计数初值为400。计数器0、计数器1、计数器2和控制寄存器的端口地址依次为80H-83H，试编写8253的初始化程序。
- 1. 控制字：控制字为01110001B，写入控制寄存器，端口地址为83H。
- 2. 计数值：计数初值为400，由于采用BCD计数，故应按BCD码方式组成，即0400H，送入计数器1的数据端口，地址是81H。16位数送两次，先送低8位00H，后送高8位04H。
- 3. 初始化程序：

MOV	AL, 71H	； 控制字
OUT	83H, AL	
<hr/>		
MOV	AL, 00H	； 低8位计数值
OUT	81H, AL	
<hr/>		
MOV	AL, 04H	； 高8位计数值
OUT	81H, AL	

例 7.2.2 8253 在 PC 中的应用

- 在PC/XT微机中使用了一片8253，端口地址设置为40H-43H，接入1.193MHz的时钟信号。
- **计数器0**：提供系统电子时钟的**时间基准**。**方式3**，产生方波。计数器0的输出端OUT0连接到8259A的IR0，作为系统的中断源。
- **计数器1**：用于DRAM的**刷新定时**。**方式2**，产生连续的脉冲信号。
DRAM要求在2ms内对全部128行存储单元完成一次刷新操作。OUT1输出的脉冲信号连接到一个触发器，由触发器向8237A提出DRAM刷新的DMA请求，DRAM的刷新是在DMA周期中完成的，8253的计数器1只是提供一个产生DRAM刷新DMA请求的定时触发信号。
- **计数器2**：提供驱动器内扬声器的**音频信号**，控制扬声器发出频率为1KHz的声音。**方式3**，产生方波。

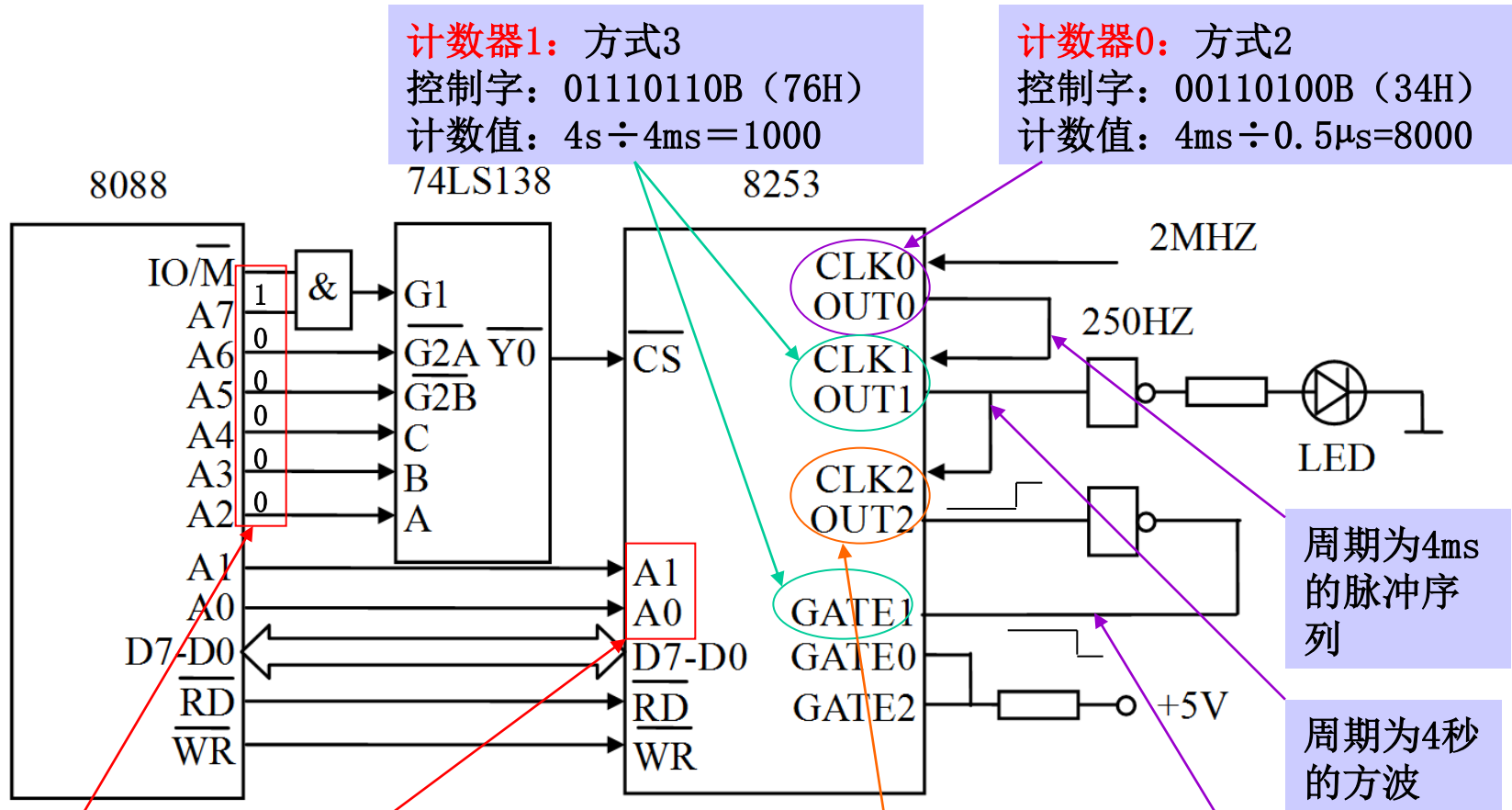
例 7.2.3 8253 应用系统设计

- 某8088系统采用8253精确控制一个发光二极管闪亮，系统要求启动8253后使发光二极管**点亮2秒**，**熄灭2秒**，**亮灭50次**后停止闪动，系统工作结束。现有一个时钟脉冲源，频率为**2MHz**，其他器件任选。试完成下列工作：
 - （1）分析该系统接口电路。
 - （2）编写完成上述功能的程序。

系统分析

- 8253与CPU的接线及端口地址译码电路的设计按常规方法进行，按接口电路确定地址。
- 系统主要是控制发光二极管的亮灭，亮2秒灭2秒，恰好是一个方波周期，周期为4秒，因此可用8253的方式3。
- 2MHz的周期为 $0.5\mu\text{s}$ ，一个通道的最大输出周期为
$$65536 \times 0.5\mu\text{s} = 32.768\text{ms}$$
- 达不到4秒要求。而如果用该通道级联下一个通道，则下一个通道的最大输出周期为
$$65536 \times 32.768\text{ms} = 2147\text{s}$$
- 因此，由两个通道级联来产生最后的方波。
- 前一个通道的CLK接2MHz，工作于脉冲频率发生器方式，产生一个脉冲波，假设脉冲波周期为4ms（250HZ），于是它的计数值为 $4\text{ms} \div 0.5\mu\text{s} = 8000$ ，它的OUT输出接后一个计数器通道的CLK输入。
- 后一个计数器通道工作于方波发生器方式，产生周期为4秒的方波，于是它的计数值为 $4\text{秒} \div 4\text{ms} = 1000$ 。

图 7.2.9 8253 应用系统设计接口电路



A7	A6	A5	A4	A3	A2	A1	A0	端口地址	通道
1	0	0	0	0	0	0	0	80H	#0
1	0	0	0	0	0	0	1	81H	#1
1	0	0	0	0	0	1	0	82H	#2
1	0	0	0	0	0	1	1	83H	控制口

计数器2: 方式0,
控制字: 10010000B
计数值: 计满次数50

输出OUT2通过一个非门连GATE1，用于控制计数器1通道的启停。

工作程序设计

MOV	AL, 90H	; #2控制字
OUT	83H, AL	
<hr/>		
MOV	AL, 50	; #2计数值
OUT	82H, AL	
<hr/>		
MOV	AL, 34H	; #0控制字
OUT	83H, AL	
<hr/>		
MOV	AX, 8000	; #0计数值
OUT	80H, AL	
MOV	AL, AH	
OUT	80H, AL	
<hr/>		
MOV	AL, 76H	; #1控制字
OUT	83H, AL	
<hr/>		
MOV	AX, 1000	; #1计数值
OUT	81H, AL	
MOV	AL, AH	
OUT	81H, AL	
<hr/>		
.....		

7.3 可编程串行通信接口芯片 8251A

- 基本的通信方式有两种：并行通信、串行通信。
- 7.3.1 串行通信概述
- 在数字通信中，首先要解决的问题是数字中的1和0如何表示与传送。
- 有两种常用方法：基带传输、频带传输。

1. 数字信号的基带传输和频带传输

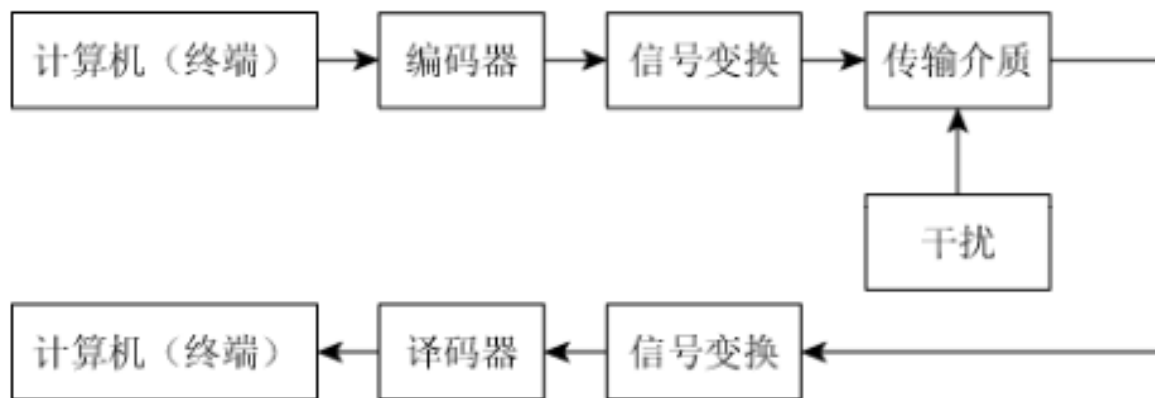


图 7.3.1 数字信号的基带传输系统框图

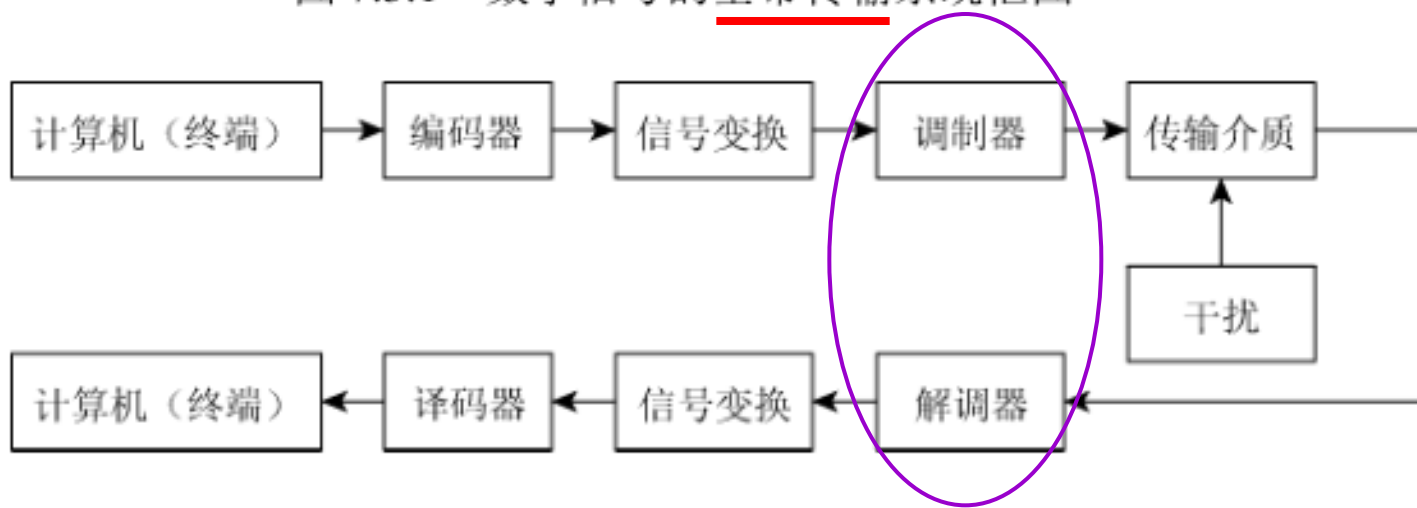


图 7.3.2 数字信号的频带传输系统框图

调制与解调

- **调制**：用数字信号控制载波的一个参数的变化，就可以实现数字信号转换成频带信号，这种变换就是调制。
- **解调**：已调信号经信道传输到接收端，在接收端通过反变换，将已调信号恢复成数字信号，这一变换过程称为解调。
- **调制解调器**：具有调制和解调两种功能的装置称为调制解调器（Modem）。
- 按调制方式，Modem可分三类：**调幅、调频和调相**。
- 调频方式是常用的一种调制方式，这种形式的调制称为频移键控FSK。

频移键控法FSK调制原理

- **频移键控调制原理：**两个不同频率的模拟信号分别由电子开关控制，在运算放大器的输入端相加，而电子开关由需要传输的数字信号来控制。
- 当信号为1时，控制开关1导通，送出一串频率较高的模拟信号；
- 当信号为0时，控制开关2导通，送出一串频率较低的模拟信号。
- 于是在运算放大器的输出端，就得到了已调制的信号。

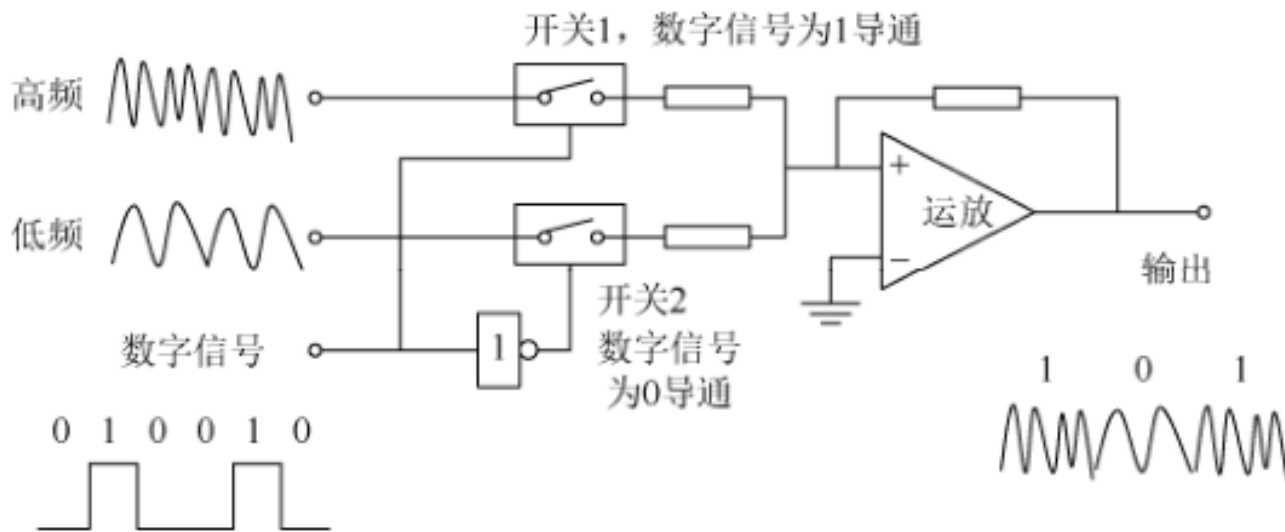
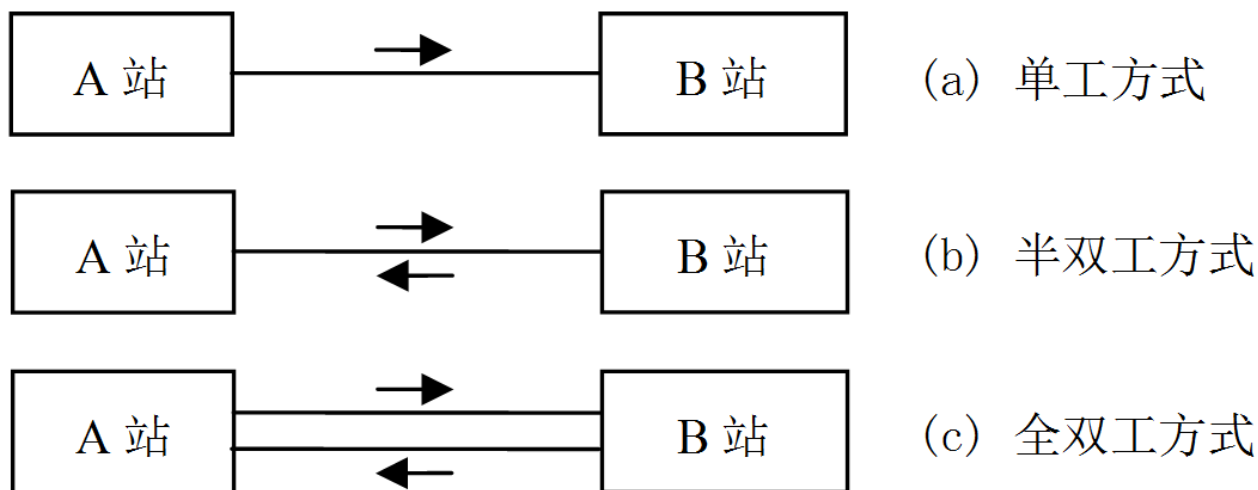


图 7.3.3 频移键控法调制原理图

2. 串行通信的传送方向

- 通常串行通信在两个站（或设备）A与B之间传送数据。
- 按通信线路上数据传递的**方向**和时间的关系，将通信**分成三类**：单工通信、半双工通信、全双工通信。



3. 传输速率

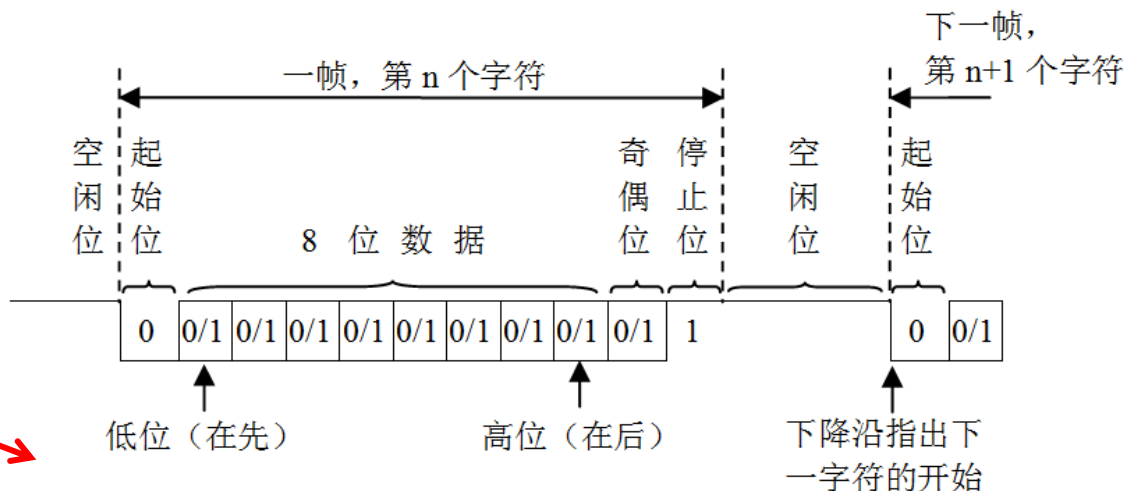
- 串行通信是一位一位传送的。衡量数字通信系统的一项重要指标是它的传输速率。
- **1. 信息传输速率**
- 信息传输速率（传信率或比特率）：单位时间（每秒）内通信系统所传送的信息量，其单位为比特/秒（bit/s或bps）。
- **2. 码元传输速率**
- 码元传输速率（传码率）：单位时间（每秒）内通信系统所传送的码元数目，其单位为波特（Baud）。

4. 异步通信与同步通信

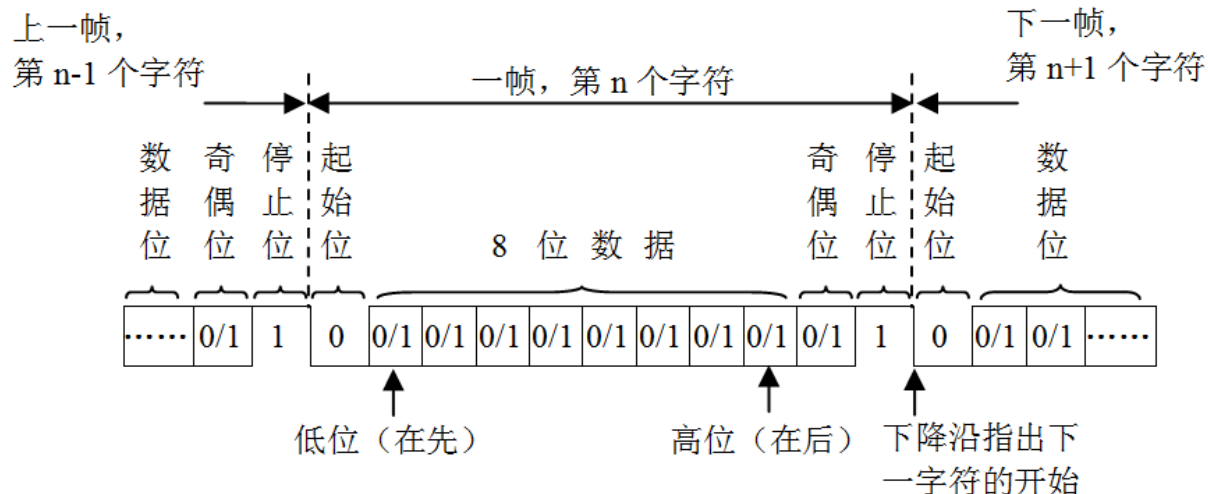
串行通信中有两种基本的通信方式:

异步通信

同步通信



(a) 小于最高数据传送率格式



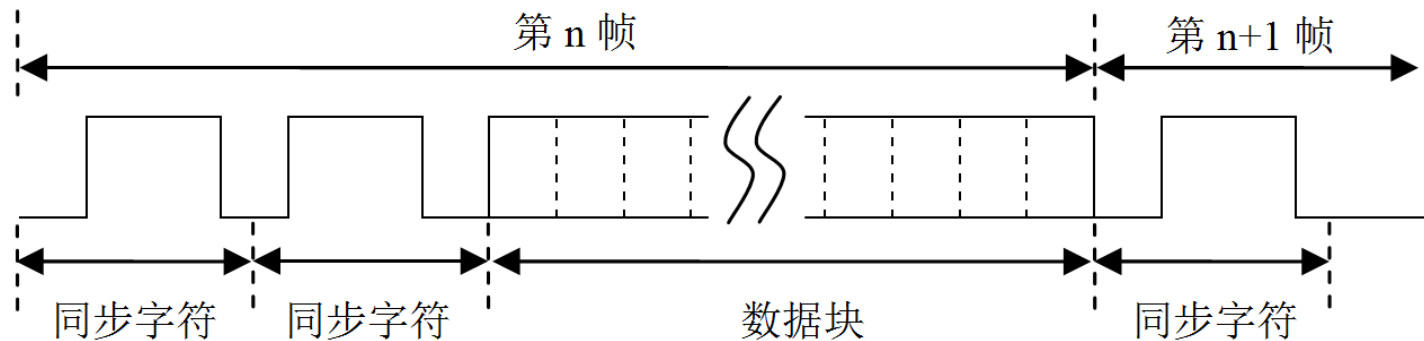
(b) 最高数据传送率格式

同步串行通信

串行通信中有两种
基本的通信方式：

异步通信

同步通信



注：帧与帧之间可以有间隙

5. RS-232-C串行通信接口总线

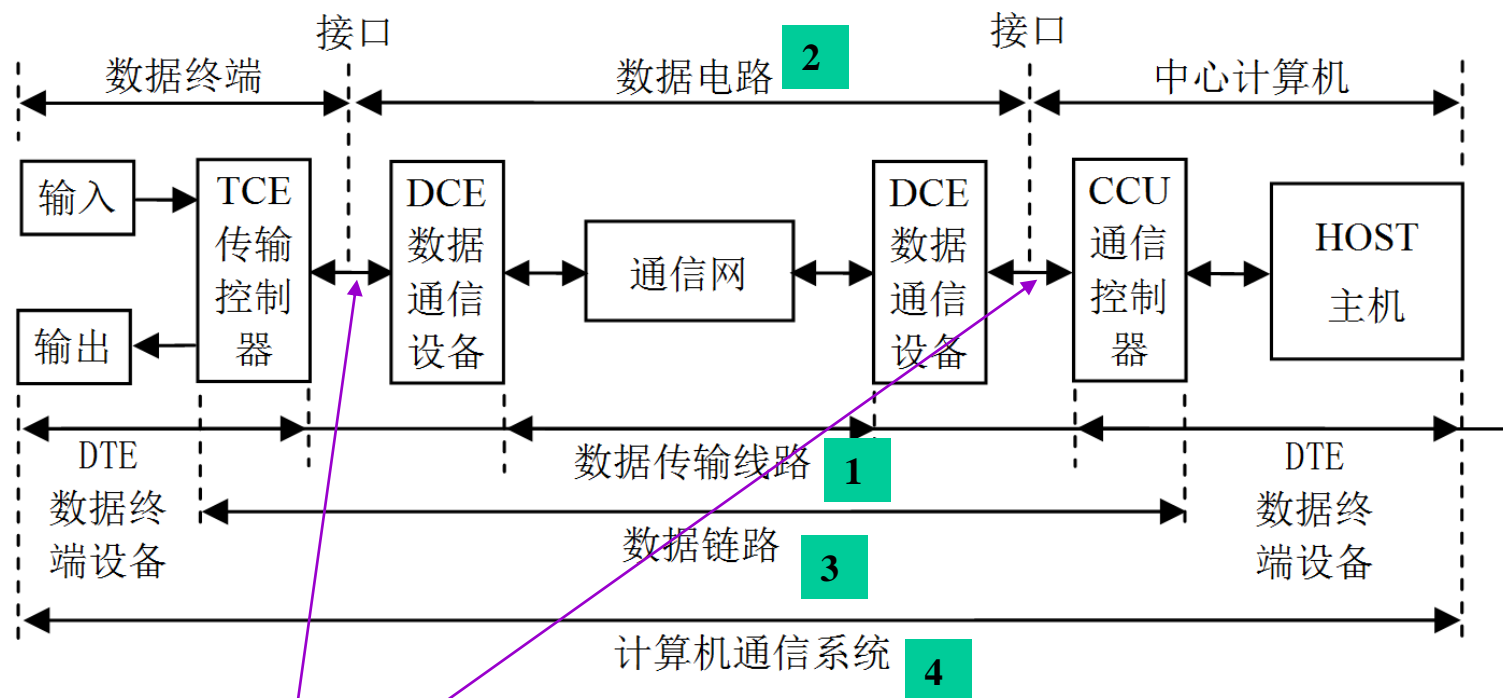


图 7.3.7 串行通信系统的结构框图

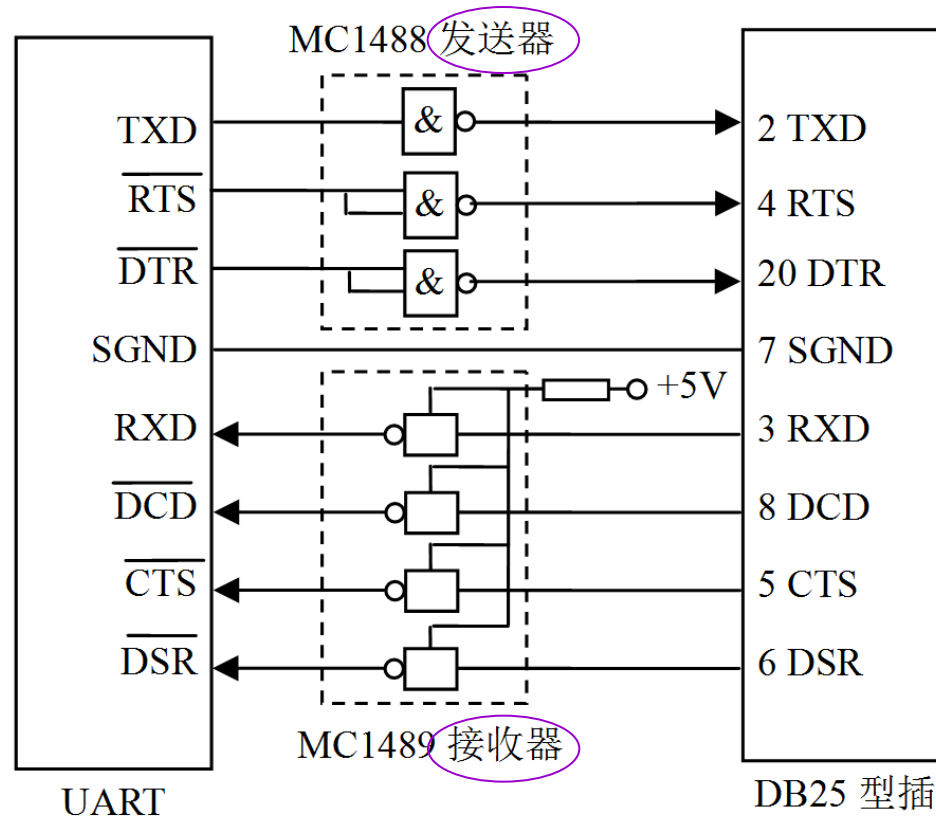
接口处使用RS232C接口标准，对信号功能、电气特性、机械特性做出规定

按分段功能分4个部分：
传输线路
数据电路
数据链路
计算机通信系统

RS-232-C接口标准

- **1. 机械特性**
- RS-232-C并未定义连接器的物理特性，因此，出现了D25、D15和D9各种类型的连接器，其引脚定义也各不相同。常用的是D9型连接器。
- **2. 电气特性**
- 采用负逻辑电平：
- 逻辑0---对信号地线+3V~+15V（通常取+12V）
- 逻辑1---对信号地线-3V~-15V（通常取-12V）
- **3. 功能特性**
- 主要是对接口中的各接口线作出功能定义，并说明相互间的操作关系。

RS-232-C与TTL之间电平转换电路



(1) RS-232-C用正负电压表示的逻辑状态，与TTL以高低电平表示的逻辑状态，其规定不同。

(2) 因此，为了能够使RS-232-C同计算机接口或终端的TTL器件连接，必须在RS-232-C与TTL电路之间进行电平和逻辑关系的变换。

(3) 实现这种变换的方法可用集成电路芯片MC1488和MC1489来完成。

表 7.3.1 RS-232-C 接口功能定义

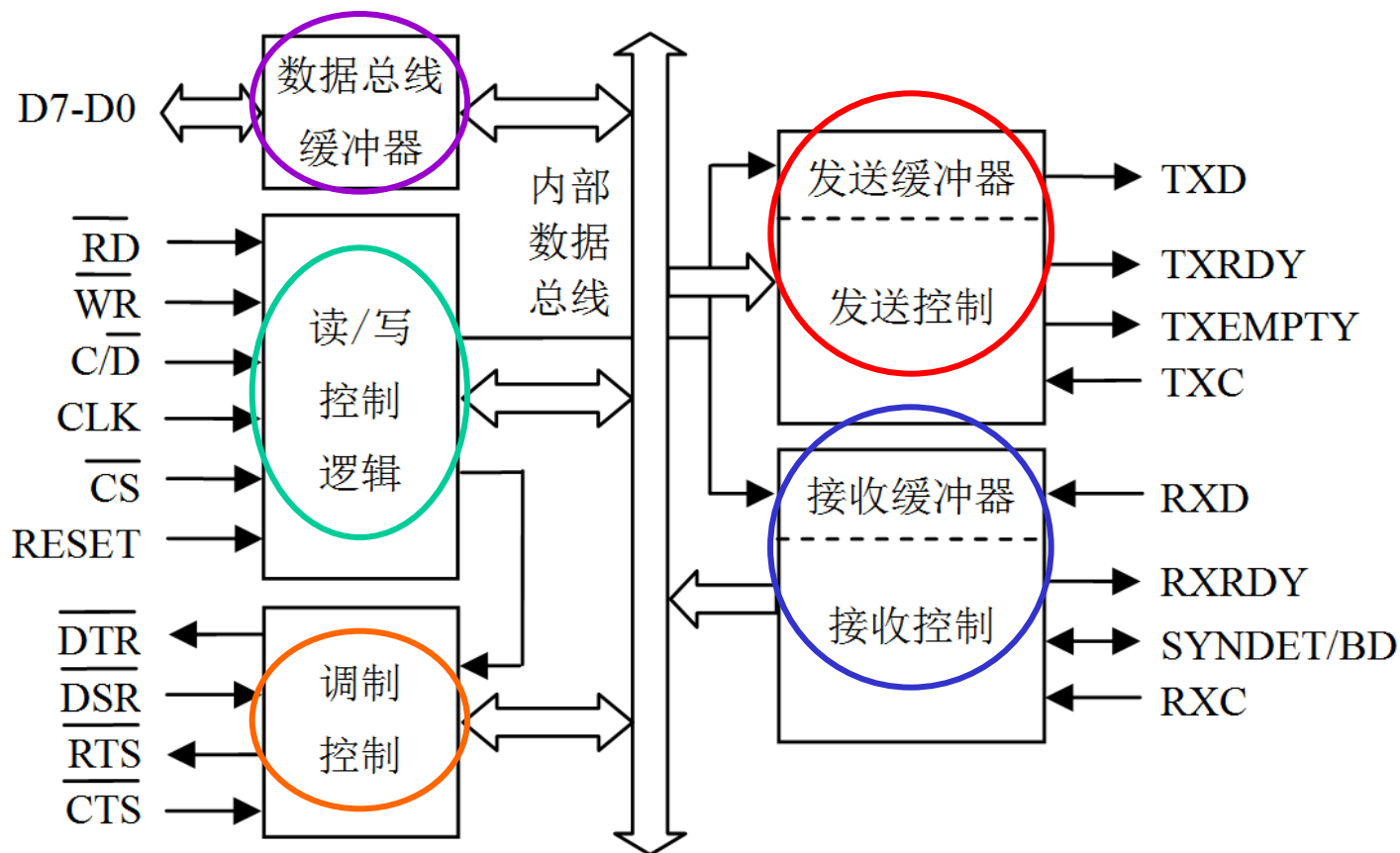
引脚号	符号	信号名称	缩写	描述
1	AA	保护地（屏蔽）	PB	用于设备地
2	BA	发送数据	TXD	输出数据到modem
3	BB	接收数据	RXD	由modem输入数据
4	CA	请求发送	RTS	至modem，打开modem的发送器
5	CB	清除发送（允许发送）	CTS	由modem来，指示modem发送就绪
6	CC	DCE就绪	DSR	由modem来，指示modem电源已接，也不在测试期
7	AB	信号地（公共回路）	SGND	
8	CF	接收线信号检测器（载波检测）	DCD	由modem来，指示modem正接收通信链路的信号
9		测试预留		
10		测试预留		
11		未定义		
12	SCF	第二接收线信号检测器		由modem来，指示modem正接收辅通信链路的信号
13	SCB	第二清除发送		由modem来，指示modem辅信道发送就绪
14	SBA	第二发送数据		至modem，输出低速率数据
15	DB	发送器信号码元定时（DCE源）	TXC	由modem来，给终端或接口提供发送器时序
16	SBB	第二接收数据		由modem来，输入低速率数据
17	DD	接收器信号码元定时	RXC	由modem来，给终端或接口提供接收器时序
18		未定义		
19	SCA	第二请求发送		至modem，打开modem的辅信道发送器
20	CD	DTE就绪	DTR	至modem，准许modem接入通信链路，开始发送数据
21	CG	信号质量检测	SQD	由modem来，接收数据中的差错几率为低时才有效
22	CE	振铃检测	RI	由modem来，指示通信链路测出响铃信号
23	CH/CI	数据速率选择DTE源/DCE源		至modem/由modem来，指示两个同步数据之一的速率或速率范围
24	DA	发送器信号码元定时（DTE源）	TXC	至modem，给modem发送器提供时序
25		未定义		

7.3.2 8251A 内部逻辑与工作原理

- 8251A的基本性能：
 - (1) 可以工作在同步方式，也可以工作在异步方式；在同步方式下，波特率为0~64K波特，在异步方式下，波特率为0~19.2K波特。
 - (2) 在同步方式时，可以用5、6、7或8位来代表字符，并且内部能自动检测同步字符，从而实现同步。此外，8251A还允许在同步方式下增加奇/偶校验位进行校验。
 - (3) 在异步方式时，可以用5、6、7或8位来代表字符，用1位作奇偶校验。此外，能根据编程为每个字符设置1个、1.5个或2个停止位。
 - (4) 所有的输入输出电路都与TTL电平兼容。
 - (5) 全双工双缓冲的接收/发送器。

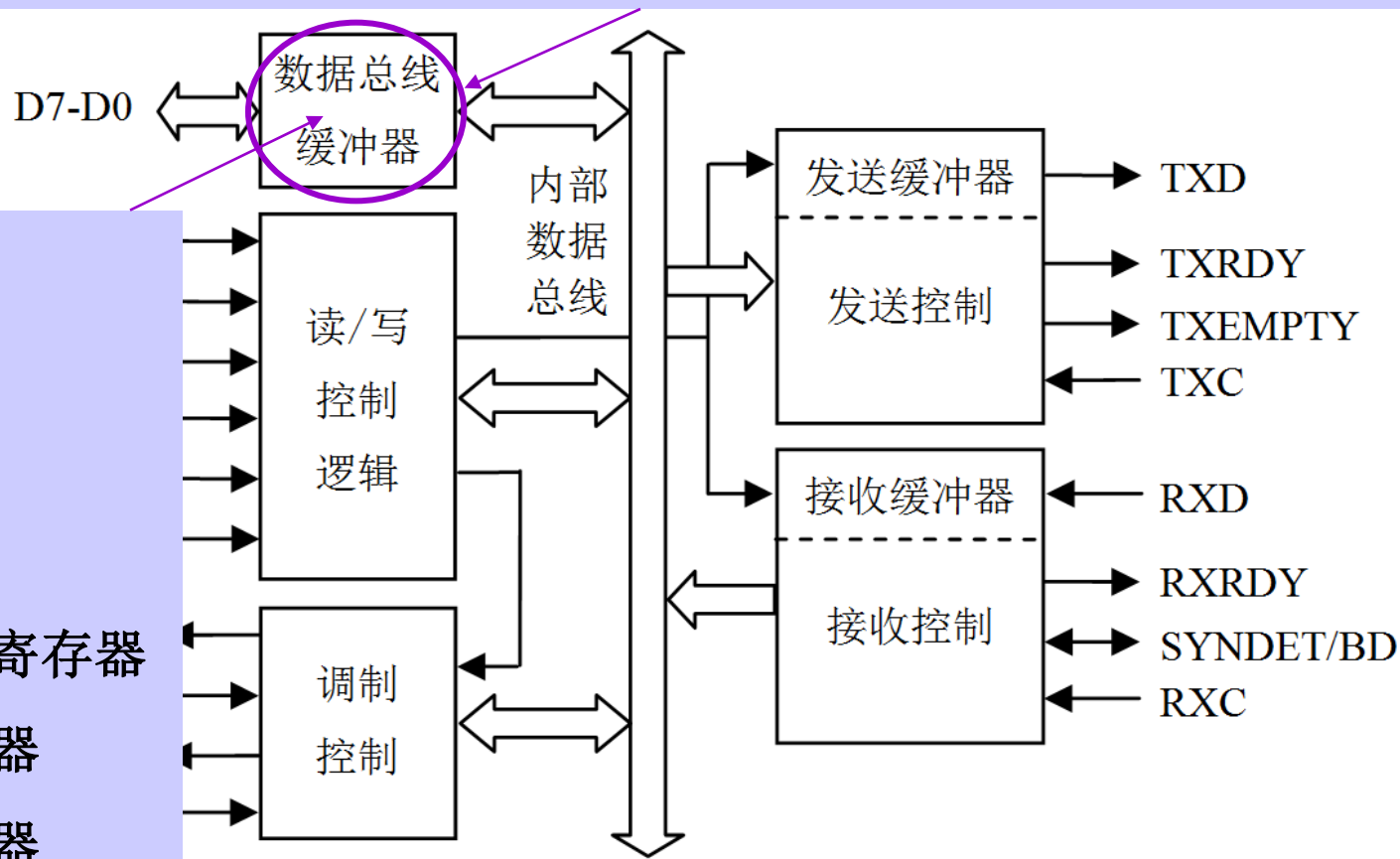
8251A 内部逻辑 (含 7.3.3 8251A 的引脚功能)

5个主要部分： 数据总线缓冲器、接收缓冲器、发送缓冲器、读/写控制逻辑电路、调制解调控制电路



1. 数据总线缓冲器

数据总线缓冲器是三态双向8位缓冲器，它是8251A与微机系统数据总线的接口，数据、控制命令及状态信息均通过此缓冲器传送。



含有:

命令寄存器

状态寄存器

方式寄存器

两个同步字符寄存器

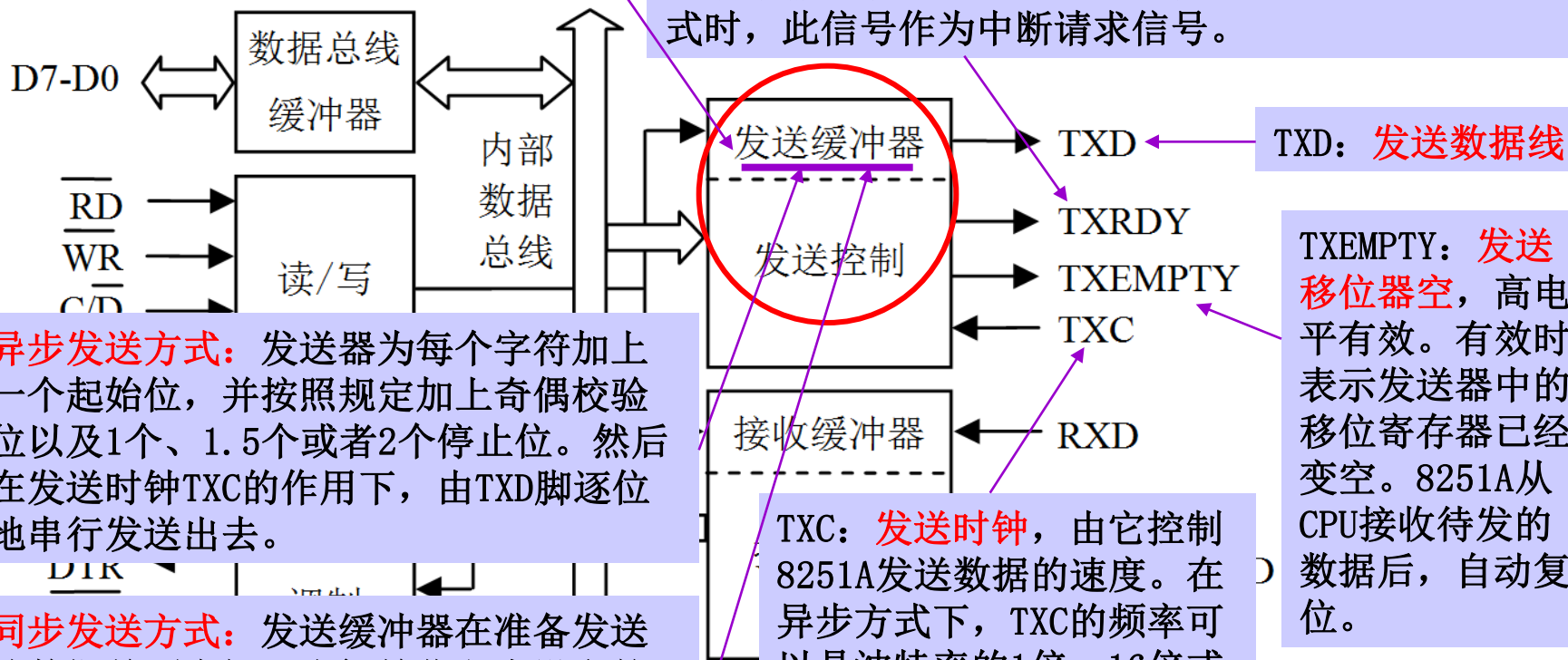
数据输入缓冲器

数据输出缓冲器

2. 发送缓冲器

功能：接收CPU送来的并行数据，按照规定的数据格式变成串行数据流后，由TXD输出线送出。

TXRDY: 发送器准备好，高电平有效。有效时，表示发送器已准备好接收CPU送来的数据。当CPU向8251A写入一个数据后，TXRDY自动复位。当8251A允许发送，且数据输出缓冲器为空时，此信号有效。在用中断方式时，此信号作为中断请求信号。



异步发送方式：发送器为每个字符加上一个起始位，并按照规定加上奇偶校验位以及1个、1.5个或者2个停止位。然后在发送时钟TXC的作用下，由TXD脚逐位地串行发送出去。

同步发送方式：发送缓冲器在准备发送的数据前面先插入由初始化程序设定的一个或两个同步字符，在数据中插入奇偶校验位。然后在发送时钟TXC的作用下，将数据逐位地由TXD引脚发送出去。

TXC：发送时钟，由它控制8251A发送数据的速度。在异步方式下，TXC的频率可以是波特率的1倍、16倍或64倍，可由程序设定。在同步方式下，TXC的频率与发送数据波特率相同。

TXEMPTY：发送移位器空，高电平有效。有效时，表示发送器中的移位寄存器已经变空。8251A从CPU接收待发的数据后，自动复位。

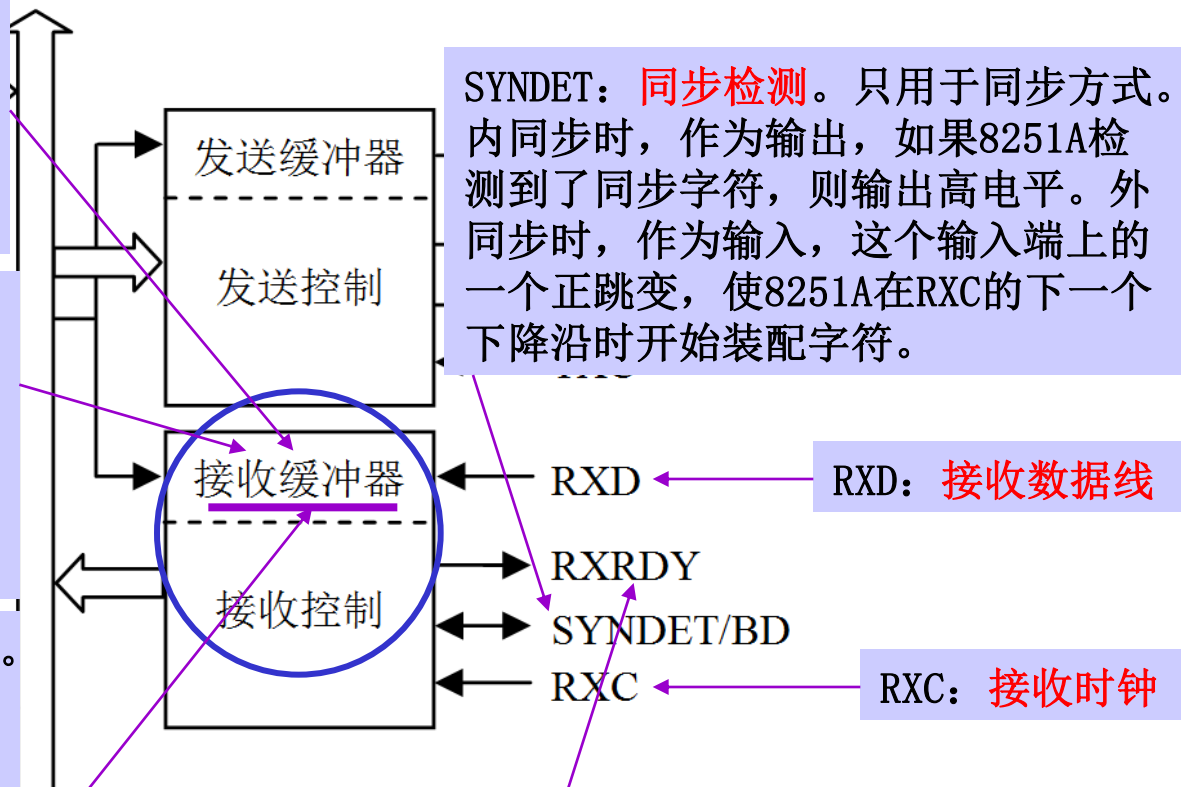
3. 接收缓冲器

功能：接收在RXD脚上输入的串行数据，并按规定的格式把串行数据转换为并行数据，存放在数据总线缓冲器中的数据输入缓冲器中。

异步接收方式：在“允许接收”条件下，接收缓冲器监视RXD线。发现起始位后，开始采样并进行字符装配，装配一个直接后，送到数据输入缓冲器，同时发出RXRDY有效信号。

同步接收方式：先搜索同步字符。匹配同步字符后，SYNDET引脚变为高电平。

外同步方式：SYNDET端出现高电平，8251A便认为已经完成同步。实现同步之后，对RXD线采样，把收到的数据送入移位寄存器中。装配字符，然后送入数据输入缓冲器，并且在RXRDY引脚上发出有效信号。

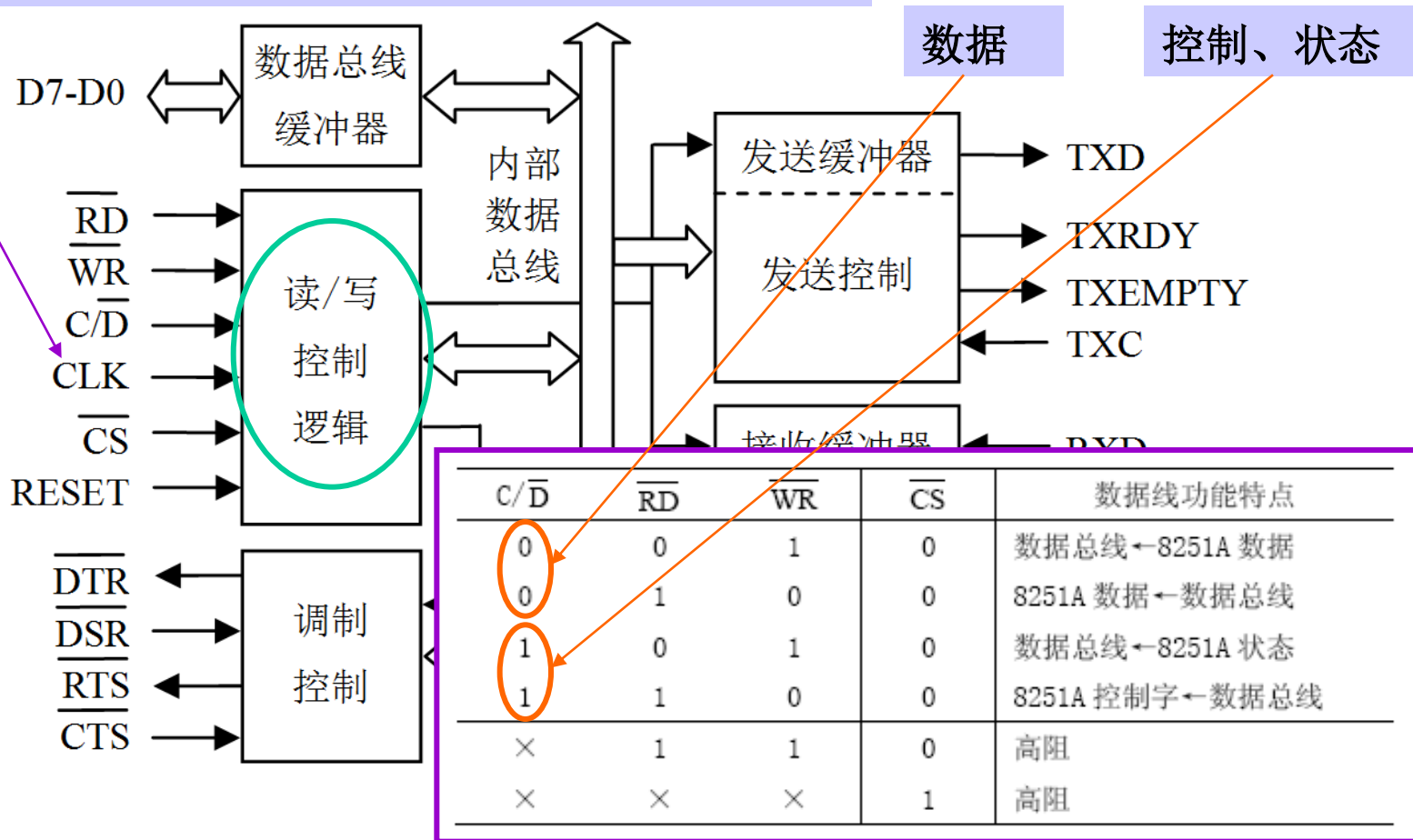


SYNDET：同步检测。只用于同步方式。内同步时，作为输出，如果8251A检测到了同步字符，则输出高电平。外同步时，作为输入，这个输入端上的一个正跳变，使8251A在RXC的下一个下降沿时开始装配字符。

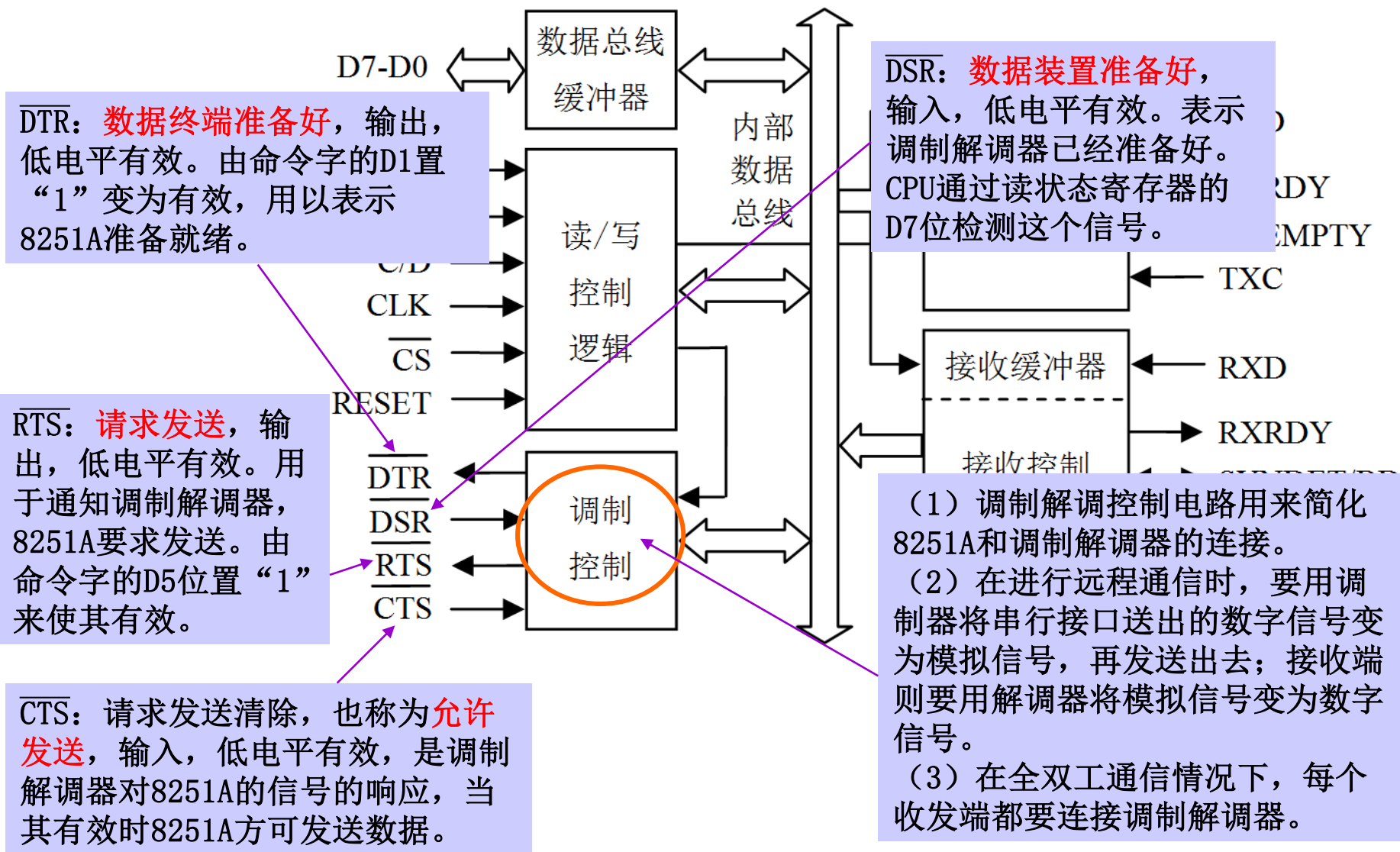
RXRDY：接收数据准备好，高电平有效。在允许接收的条件下，当8251A已经从它的串行输入端接收了一个字符，并完成了格式变换，此信号有效，通知CPU读取数据。当CPU从8251A读取一个字符后，RXRDY信号自动复位。在中断方式时可作为中断请求信号。

4. 读 / 写控制逻辑电路

CLK: 时钟, 产生8251A的内部时序。同步方式时, CLK频率要大于RXC和TXC频率的30倍。异步方式时, 此频率要大于RXC和TXC频率的4.5倍。



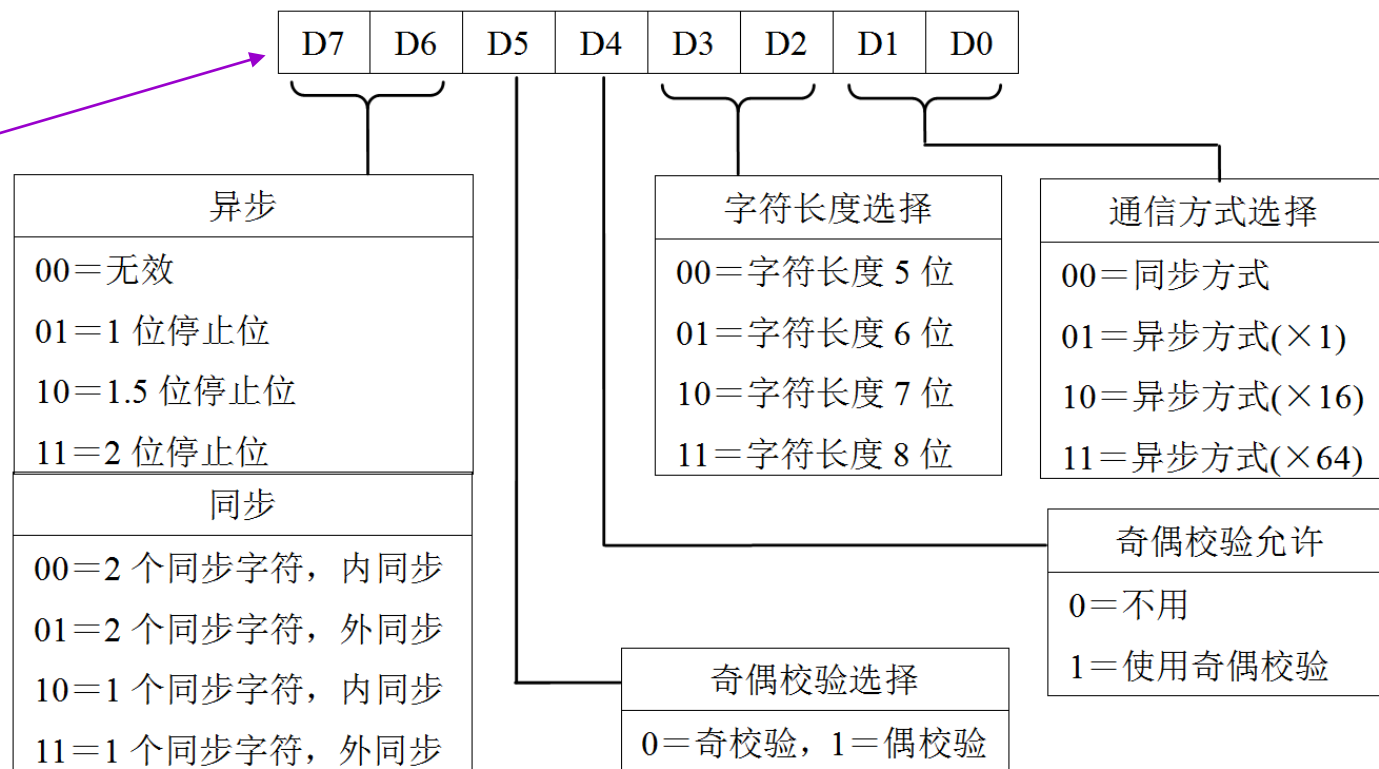
5. 调制解调控制电路



7.3.4 8251A 的控制字-方式选择控制字

8251A有两个控制字：方式选择控制字，操作命令控制字

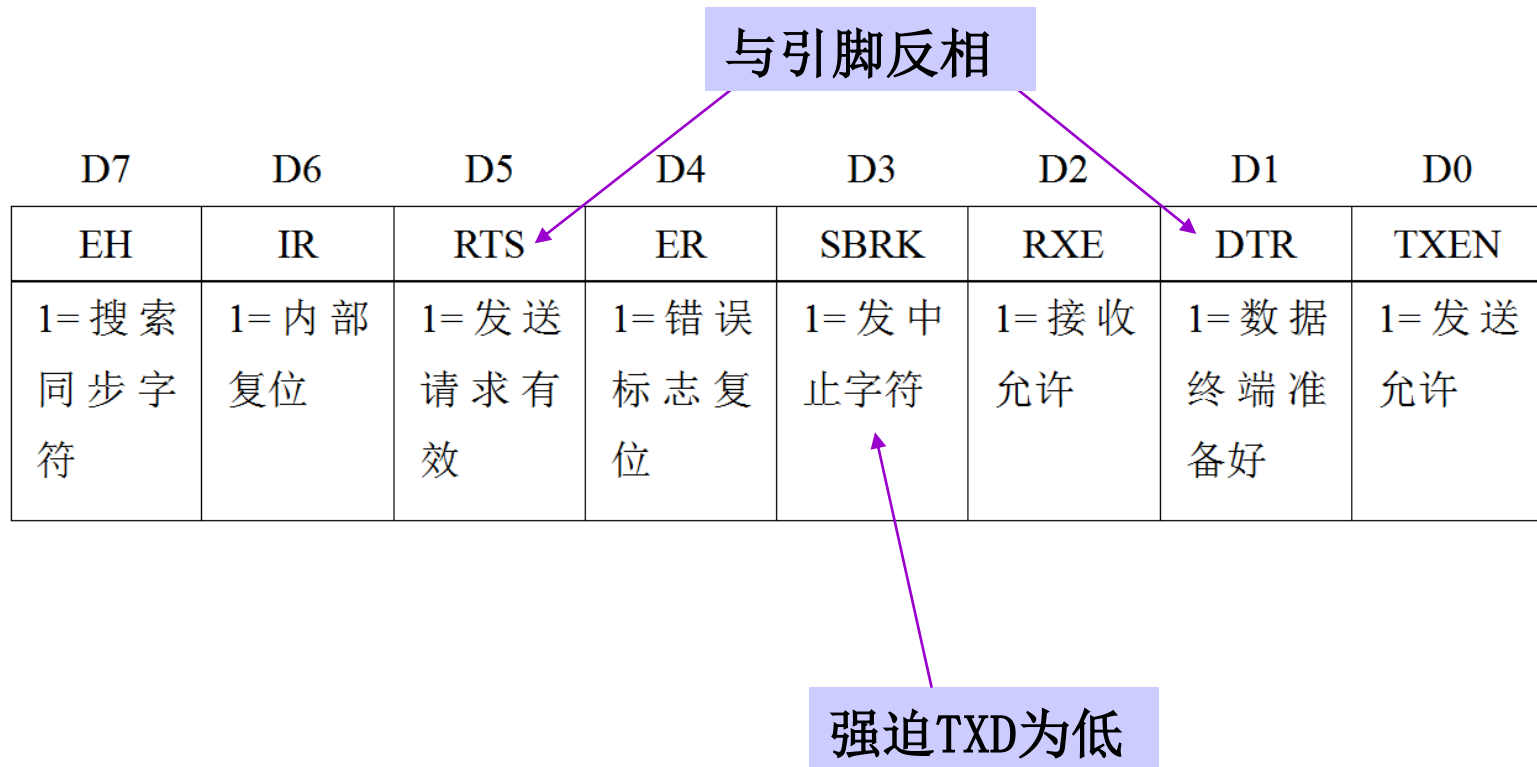
功能：方式选择控制字用来选择工作方式，确定数据位长度、是否要奇偶校验、停止位的位数或同步字符的个数等。



方式选择控制字在8251A复位之后送入，操作命令控制字在方式选择控制字之后的任何时间均可送入。

命令控制字

功能：命令控制字控制8251A的发送、接收、内部复位等的实际操作。



状态字

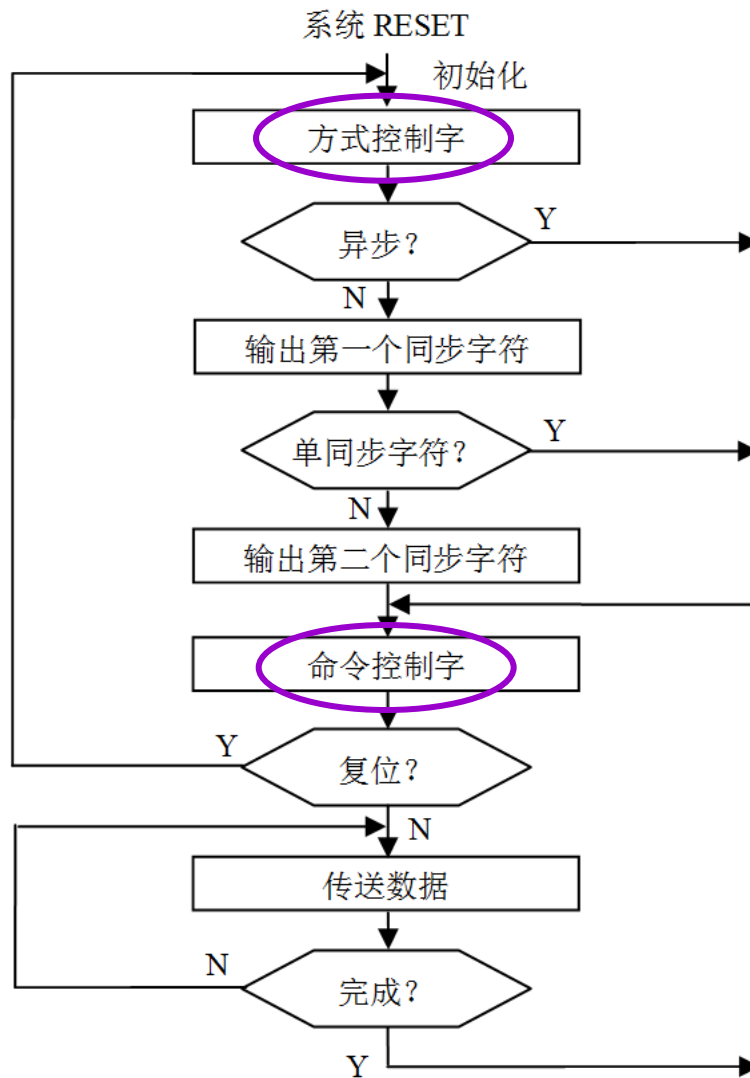
CPU可在任意时刻，通过IN指令将8251A内部状态寄存器的内容（即状态字）读入CPU，以判断8251A当前的工作状态。

与引脚相同							
D7	D6	D5	D4	D3	D2	D1	D0
DSR	SYNDET/BD	FE	OE	PE	TXEMPTY	RXRDY	TXRDY
1= 数据装置就绪	1=同步检出	1= 格式错	1= 溢出错	1=奇偶错	1=发送移位器空	1=接收准备好(输入缓冲器满, 读复位)	1=发送准备好(输出缓冲器空, 写复位)

- (1) PE=1表示当前产生了**奇偶错**。
- (2) OE=1表示当前产生了**溢出错**，CPU还没来得及将上一字符取走，下一个字符又来到了RXD端，8251A继续接收下一字符，结果使上一字符丢失。
- (3) FE=1表示未检测到停止位，产生了**格式错**，只对异步方式有效。

这里TXRDY状态与引脚TXRDY有区别。状态位TXRDY只要数据输出缓冲器空就置位；而**引脚TXRDY**要满足三个条件时才置位（即满足**CTS=0**、**TXRDY=1**，**TXEN=1**时）。当CPU往8251A输出一个字符以后，状态位TXRDY会自动清0。

8251A初始化流程图



7.3.5 8251A 的应用举例

- 例 7.3.1 8251A 异步方式初始化。
- 设8251A工作在异步方式，波特率因子为16，每个字符7个数据位，采用偶校验，2个停止位，允许发送和接收，设8251A数据端口地址为200H，8251A控制端口地址为201H。试编写8251A的初始化程序。
- 方式选择控制字：11111010B（即FAH），写入控制端口201H。
- 命令控制字：00110111B（即37H），允许发送和接收，写入控制端口201H。
- 初始化程序：

```
MOV    AL, 0FAH    ; 方式选择控制字，异步方式，7位/字符，  
                  ; 偶校验，2个停止位  
MOV    DX, 201H    ; 控制端口地址  
OUT    DX, AL      ; 传送  
MOV    AL, 37H     ; 命令控制字，发送和接收允许，清出错标志  
OUT    DX, AL      ; 传送
```


例 7.3.3 8251A 状态字的使用

- 设8251A工作在异步方式，波特率因子为16，每个字符7个数据位，采用偶校验，2个停止位，允许接收，设8251A数据端口地址为70H，8251A控制端口地址为71H。试编写8251A输入100个字符的程序段，字符存入DATA开始的存储区。
- 方式选择控制字：11111010（即FAH），写入控制端口71H。
- 命令控制字：00110111B（即37H），置引脚RTS、DTR有效，出错标志复位，允许发送和接收，写入控制端口71H。
- 状态字：检测状态字D1位的RXRDY，若RXRDY=1，说明已接收一个完整字符，可以读取。读取一个字符后，还要确定接收的字符是否正确，方法是检测状态字的D5D4D3位（帧错、溢出错、奇偶错），相应位为1表明出现对应的错误，需要进行错误处理。

初始化程序

	MOV AL, 0FAH	； 方式选择控制字，异步方式，7位/字符，偶校验，2个停止位
	OUT 71H, AL	； 送控制口
	MOV AL, 37H	； 命令控制字， $\overline{\text{RTS}}$ 、 $\overline{\text{DTR}}$ 有效，发送和接收允许，清出错标志
	OUT 71H, AL	； 送控制口
	MOV DI, OFFSET DATA	； 设置输入字符存储偏移地址
	MOV CX, 100	； 计数器初值，输入100个字符
A1:	IN AL, 71H	； 读状态字
	TEST AL, 02H	； $\text{RXRDY}=1$? 是，接收字符就绪，准备读取并存储
	JZ A1	； RXRDY 不为1，输入字符未就绪，循环等待
	IN AL, 70H	； 读取字符
	MOV [DI], AL	； 存储
	INC DI	； 修改存储指针
	IN AL, 71H	； 读状态字
	TEST AL, 38H	； 测试有无帧错、溢出错、奇偶错（状态字相应位为1表明出错）
	JNZ A2	； 出错，转错误处理程序
	LOOP A1	； 正确，接收下一个字符
	JMP A3	； 全部接收完成，转结束
A2:	； 错误处理程序
A3:	； 结束处理

结 束