第六章 典型接口芯片原理和应用

7.1 简单I/O接口电路及其应用

7.2 可编程计数器/定时器8253及其应用

7.3 可编程外围接口芯片8255A及其应用

7.4 串口通信和可编程接口芯片8251A及其应用

7.1 简单I/O接口电路及其应用

一．I/O接口的功能

1.采用I/O接口的必要性

计算机和外设之间的信息交换带来一些问题：

- 速度不匹配
- 信号电平不匹配
- 信号格式不匹配
- 时序不匹配

因此I/O设备不能直接与CPU的系统总线相连，必须在CPU与外设之间设置专门的接口电路来解决这些问题。

2. 接口的功能：

① 设置数据缓冲器以解决两者速度差异所带来的不协调问题

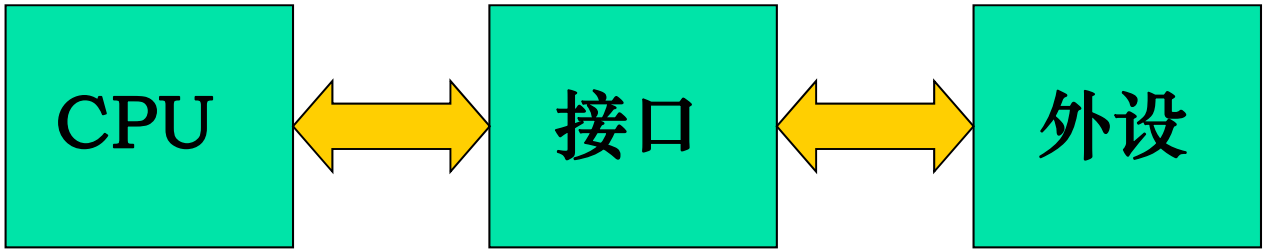
② 设置信号电平转换电路

③ 设置信息转换逻辑以满足对各自格式的要求

④ 设置时序控制电路来同步CPU和外设的工作

⑤ 提供地址译码电路，使CPU在同一时刻只能选中某一个I/O端口。

I/O接口电路是外设和计算机之间传送信息的交换部件，也称为界面，它使两者之间能很好地协调工作，每一个外设都要通过接口电路才能和主机相连。



可编程输入输出接口芯片

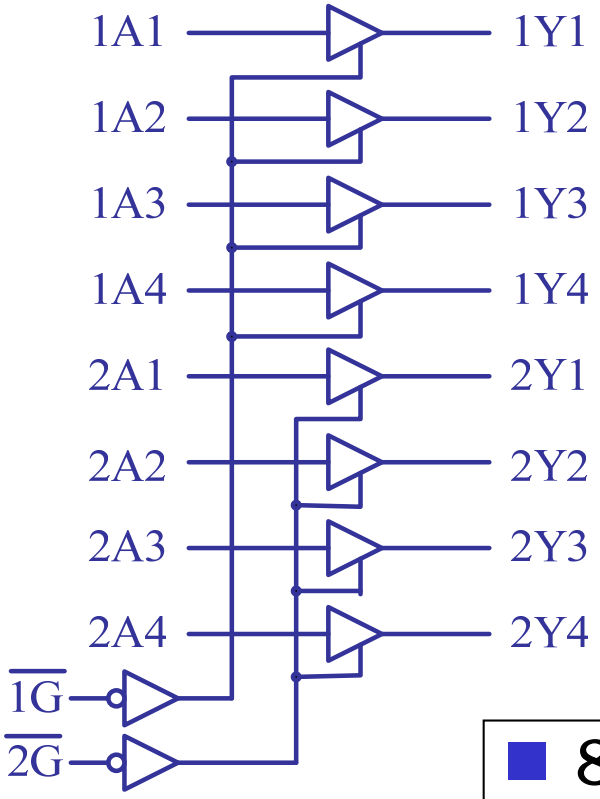
- 随着大规模集成电路技术的发展，出现了许多通用的可编程接口芯片，可用它们来方便地构成接口电路。后面几章将介绍常见的可编程I/O接口芯片的原理、编程方法及与CPU的连接方法。
  - 可编程中断控制器8259A
  - 可编程计数器/定时器8253
  - 可编程外围接口芯片8255A
  - 串行通信和可编程接口芯片8251A
  - A/D和D/A转换芯片。
- 本章介绍最常用的简单I/O接口芯片,主要有缓冲器(Buffer)和锁存器(Latch)。

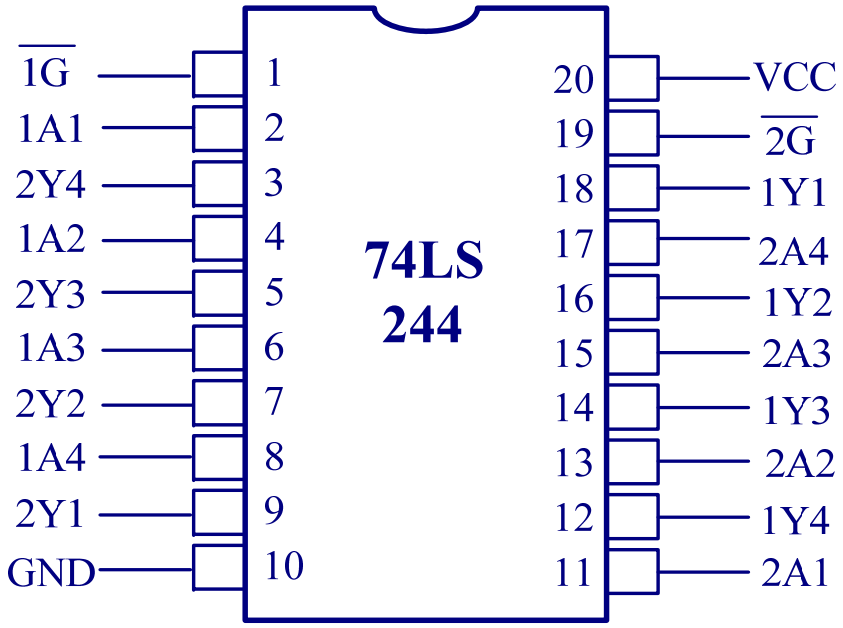
二、简单的输入输出接口芯片

1. 缓冲器74LS244和74LS245

- 连接在总线上的缓冲器都具有三态输出能力。
  - 在CPU或I/O接口电路需要输入输出数据时，在它的使能控制端EN（或G）作用一个低电平脉冲，使它的内部的各缓冲单元接通，即处在输出0或1的透明状态。数据被送上总线。
  - 当使能脉冲撤除后，它处于高阻态。这时，各缓冲单元像一个断开的开关，等于将它所连接的电路从总线脱开。
- 74LS244和74LS245就是最常用的数据缓冲器。除缓冲作用外，它们还能提高总线的驱动能力。

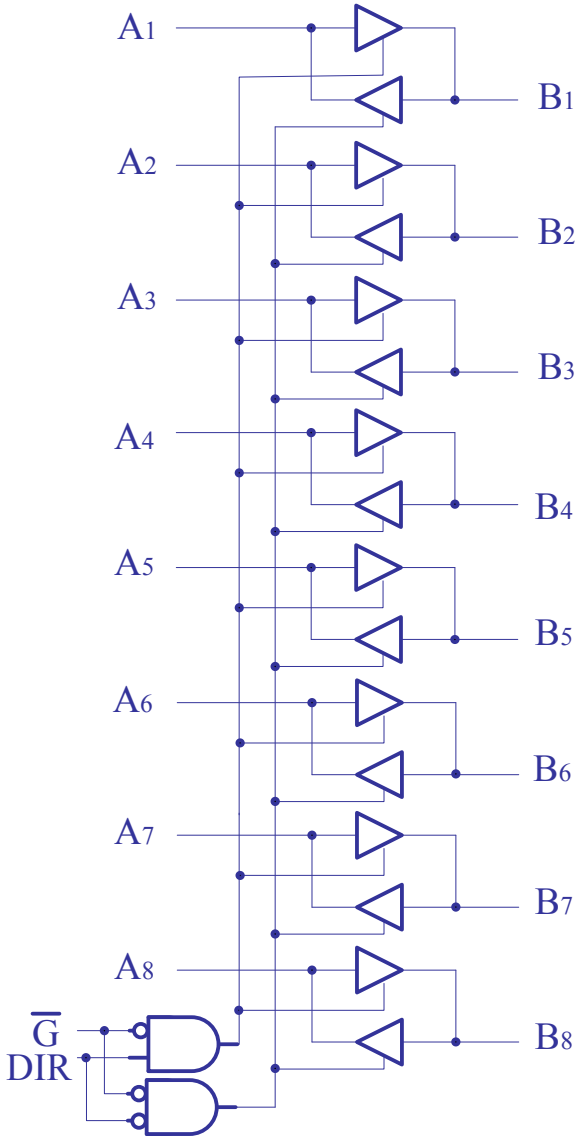
（1）74LS244 —单向数据缓冲器

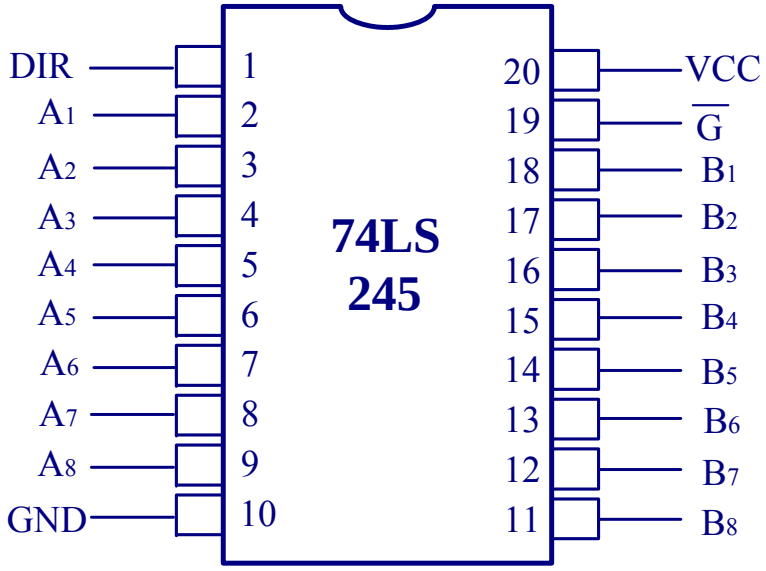




- 8个三态缓冲单元，分成两组，分别由门控信号$\overline{1G}$和$\overline{2G}$控制。$\overline{1G}$和$\overline{2G}$为低电平时，数据传送；高电平时，输出高阻态。
- 单向缓冲器，只能从A端到Y端。

(2) 74LS245 —双向数据缓冲器





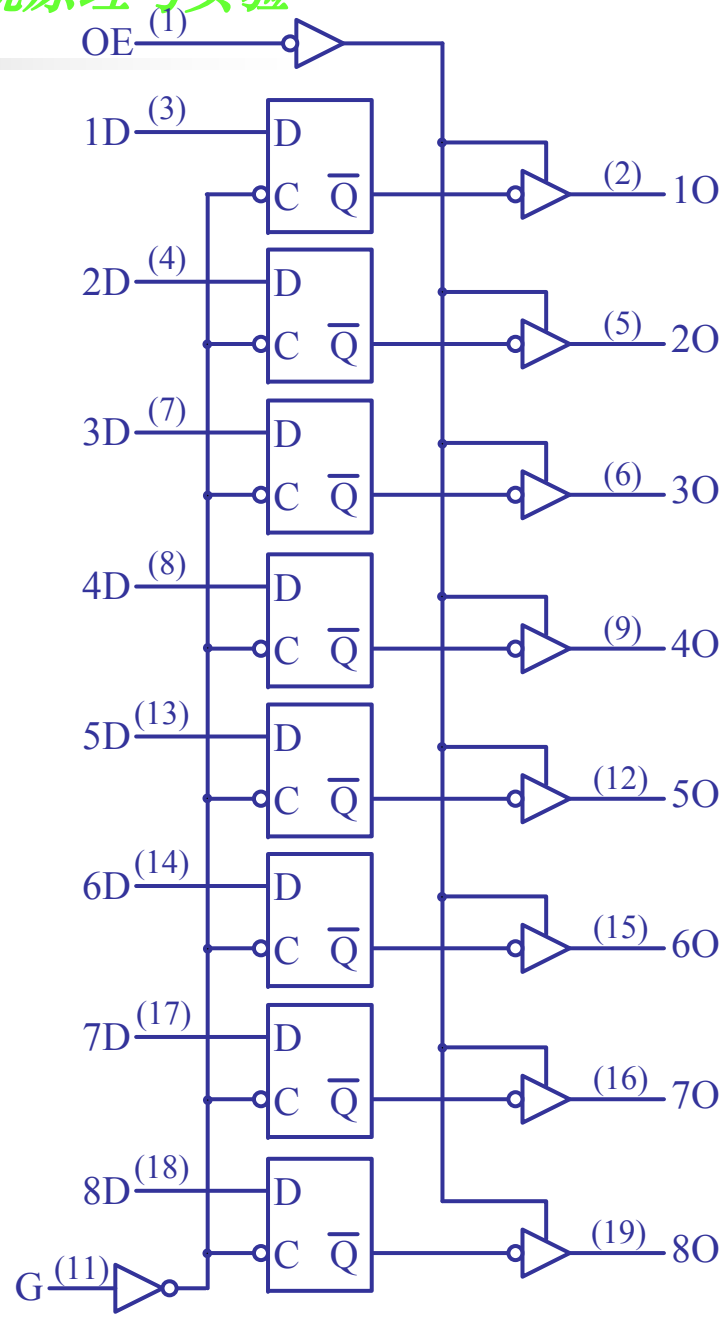
- 8个双向、三态缓冲器。
- 门控信号输入端$\overline{G}$。
- 方向控制端DIR，高电平时，数据从A端传向B端；低电平时，从B端传向A端。

2. 锁存器74LS373

- 锁存器具有暂存数据的能力，能将数据锁住，在输出控制信号的作用下将数据传送出去。74LS373是一种常用的8D锁存器，它可以直接挂在总线上，并具有三态总线驱动能力。
- 两个控制输入端：输入使能端G和允许输出端$\overline{OE}$。
- $\overline{OE}$为低时：G为高时，D端数据到O端；G为低时，O端将是前面锁存的数据，不受D端的变化影响。
- $\overline{OE}$为高时：输出将呈高阻态。

$\overline{OE}$	G	D	O
低	高	高	高
低	高	低	低
低	低	X	锁存
高	X	X	高阻态

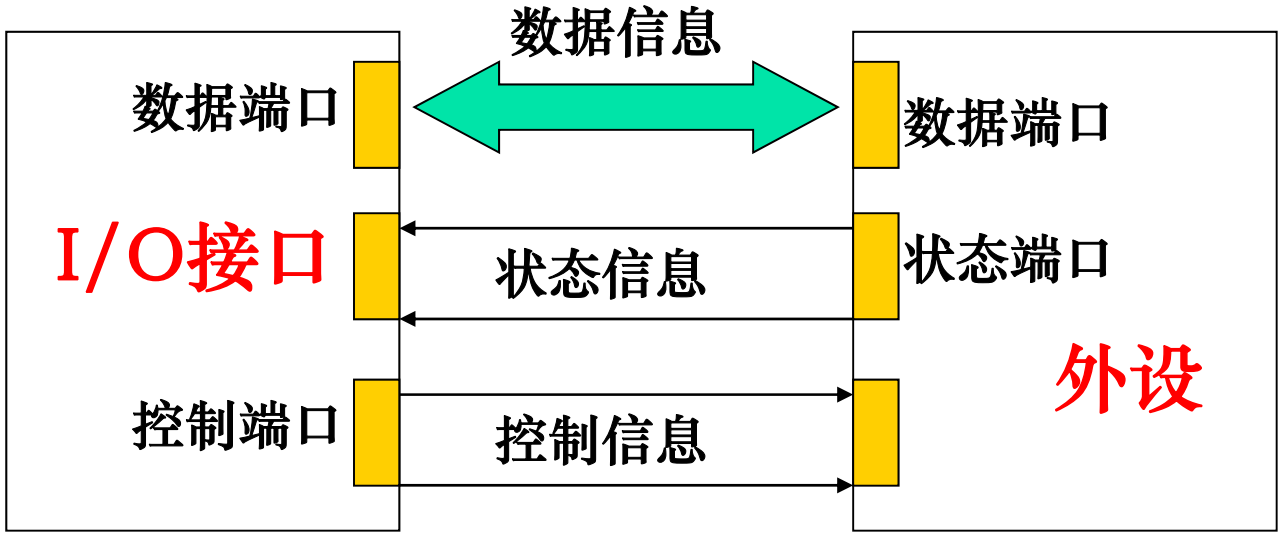
真值表



三、I/O端口及其寻址方式

1. I/O端口

- CPU与外设通信时，传送的信息主要包括数据信息、状态信息和控制信息。
- 这些信息分别进入不同的寄存器，通常将这些寄存器和它们的控制逻辑统称为I/O端口（Port），CPU可对端口中的信息直接进行读写。



a) 数据端口（Data Port）

用来存放CPU与外设之间交换的数据，长度一般为1-2个字节，主要起缓冲作用。

b) 状态端口（Status Port）

用来指示外设的当前状态。每种状态用1位表示，由CPU读取。几种最常用的状态位有：

① 准备就绪位（Ready）

- 针对CPU输入端口：1：数据寄存器已准备好数据，等待CPU读取，取走后，位清0。
- 针对CPU输出端口：1：输出数据寄存器已空，可以接收下一个数据；新数据到达后，位清0。

② 忙碌位（Busy）

- 表明外设是否能接收数据。1：外设忙，暂时不允许CPU送信的数据过来。0：外设已空闲，允许CPU发送下一个数据。

③ 错误位（Error）

- 1：指示在数据传送过程中出现错误。CPU进行相应的处理，如重新传送或中止操作等。

c) 命令端口（Command Port）

- 也称为控制端口（Control Port），用来存放CPU向接口发出的各种命令和控制字，以便控制接口和设备的动作。
- 常见的命令信息有启动位、停止位、允许中断位等。
- 接口芯片不同，控制字的格式和内容是各不相同的，常见的控制字有方式控制字、操作命令字等。

★ 在微型计算机系统中，CPU通过接口和外设交换数据时，只有输入（IN）和输出（OUT）两种指令，所以只能把状态信息和命令信息当作数据来传送，并且将状态信息作为输入数据，控制信息作为输出数据，于是三种信息都可以通过数据总线来传送了。

★ 这三种信息被送入三种不同端口的寄存器，因而能实施不同的功能。

2. I/O端口的寻址方法

CPU对外设的访问实质上是对I/O接口电路中相应端口的访问，因此和存储器一样，也需要由译码电路来形成I/O端口地址。

a) 存储器映像寻址方式（Memory Mapped I/O）

b) I/O单独编址方式

a) 存储器映像寻址方式（Memory Mapped I/O）

- 把系统中的每一个I/O端口看作一个存储单元，并与存储单元一样统一编址。访问存储器的所有指令均可用来访问I/O端口，不用设置专门的I/O指令。
- 实际上是把I/O地址映射到存储空间，作为整个存储空间的一小部分。
- 适用于不包括专门I/O操作指令的CPU
  - 优点：简化了指令系统的设计，不必包含I/O操作指令；能用功能强的存储器指令，操作方便灵活；I/O地址空间可调。
  - 缺点：I/O端口占用存储器的地址空间；译码电路复杂；指令较长，延长了输入输出的操作时间。

b) I/O单独编址方式

- 对I/O端口单独编址来构成一个I/O空间，不占用存储空间，用专门的IN和OUT指令来访问端口。
- Intel 8086和8088等采用这种方式。
- 在8086中，用地址总线的低16位来寻址I/O口。输入和输出端口可用相同的地址。CPU中的M/$\overline{IO}$控制信号用来区分是I/O寻址和存储器寻址。
  - 优点：将I/O指令和访存指令区分开，使程序清晰，可读性好；I/O指令较短，执行速度快，也不占用内存空间；I/O译码电路较简单。
  - 缺点：CPU指令系统必须有专门的IN和OUT指令，没有访存指令的功能强。CPU必须提供区分存储器和I/O读写的控制信号（如8086的M/$\overline{IO}$信号等）。

四、CPU与外设间的数据传送方式

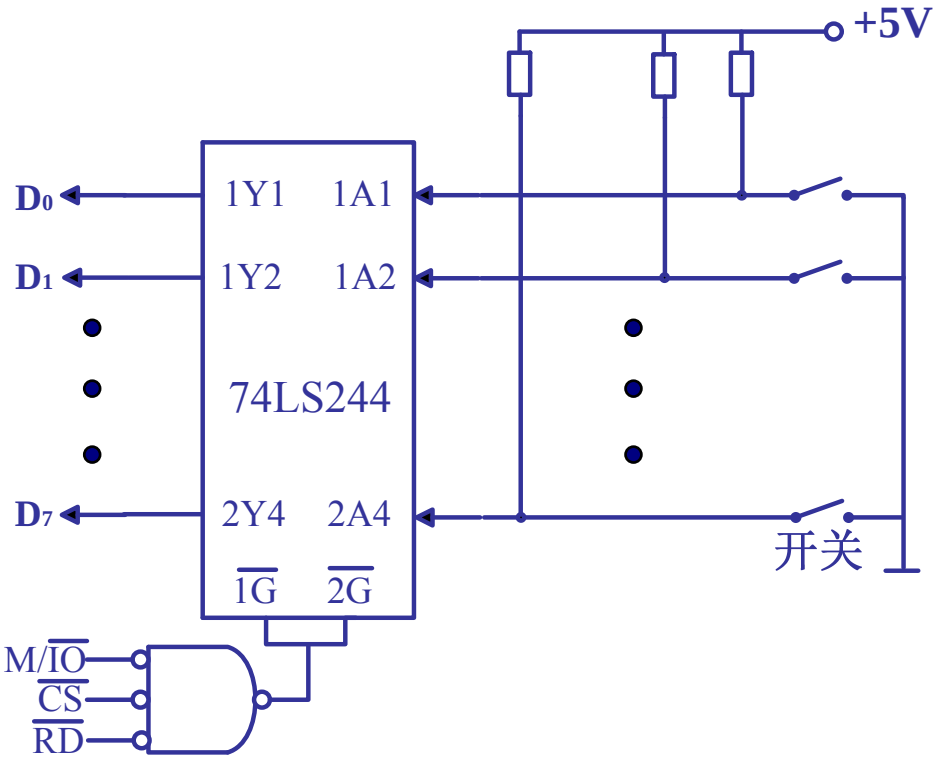
- 软件实现：程序控制方式、中断方式。
- 硬件实现：DMA方式。

1．程序控制方式

① 无条件方式（同步传送方式）

- 最简单的传送方式，主要用于对简单的外设进行操作，或者外设的定时是固定的或已知的场合。
- 程序可以不必检查外设的状态，而在需要进行I/O操作时，直接执行I/O指令。

无条件传送方式

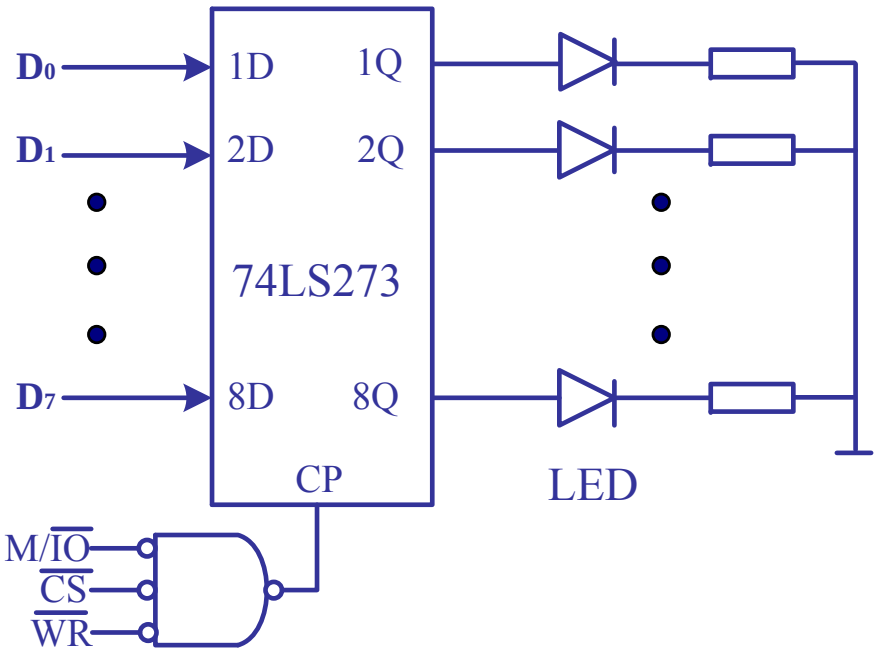


简单输入端口

CPU查询按键开关的状态

- 开关连接到三态缓冲器，缓冲器输出端接到CPU的数据总线。
  - 开关断开：高电平输入；
  - 开关闭合：低电平输入。
- 执行输入指令，使M/$\overline{IO}$、$\overline{RD}$和片选信号CS同时变为低电平，经过反向与非门变成有效的低电平开启缓冲器的三态门，使开关的当前状态以二进制的形式被读入CPU。
- 检查字节各位的内容，就能了解各开关当前状态。

无条件传送方式



简单输出端口

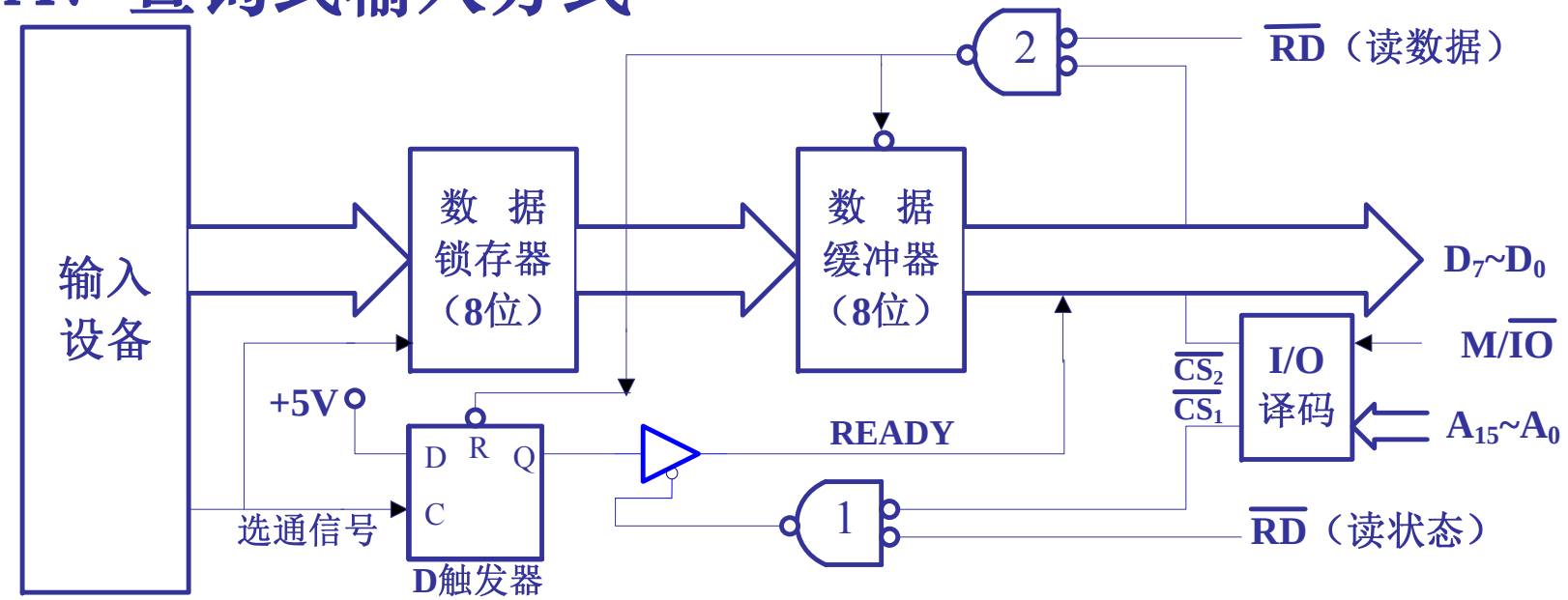
控制LED显示器的点灭

- 用一个由锁存器构成的输出端口来把LED接到计算机的数据总线上，并串接一个限流电阻，共阴连接。
- 点燃LED的位是1，灭的是0。
- 输出指令使$M/\overline{IO}$、$\overline{WR}$和片选信号$\overline{CS}$同时变低，相与后的低电平信号经反相后出发锁存器。
- 由于锁存器的作用，输出值能一直保持到下一个输出指令到达为止，这段时间内，LED的状态也将保持不变。

② 条件传送（查询式传送）

- 一般情况下，CPU在执行I/O时，很难保证输入设备总是准备好数据，或是输出设备已经处在接收数据状态。CPU必须先确认外设已处于准备传送数据状态，才能进行传送。
- CPU必须先执行一条输入指令，从外设的状态口读取当前的状态。如果外设未准备好或处于忙碌状态，则程序要转回去反复执行读状态指令，不断检测外设的状态；直到外设准备就绪为止，然后CPU才可以进行正常的I/O操作。

A、查询式输入方式

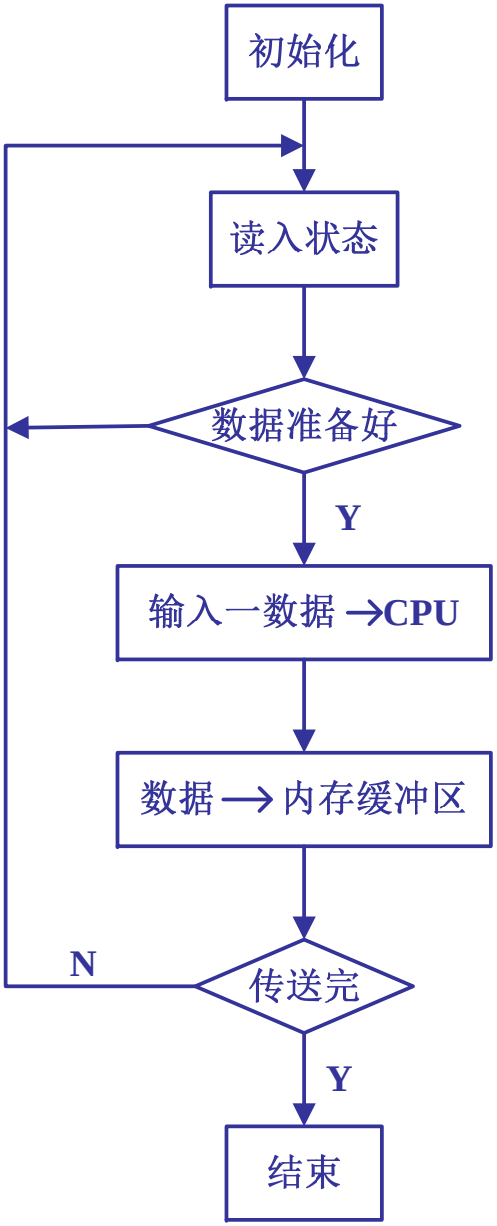


- 当输入设备准备好数据后，就向I/O接口电路发一个选通信号。
  - 将外设的数据打入数据锁存器中。
  - 使D触发器的Q端置1，表明数据准备好。
- CPU先执行IN指令读状态口的信息，三态门开启，Q端的1送到D0位，并被读入累加器。
- 程序检测到D0为1后，便执行IN指令读数据口。
  - 开启数据缓冲器，将外设送到锁存器中的数据经缓冲器送到数据总线后进累加器。
  - 将D触发器清0，一次数据传送完毕。

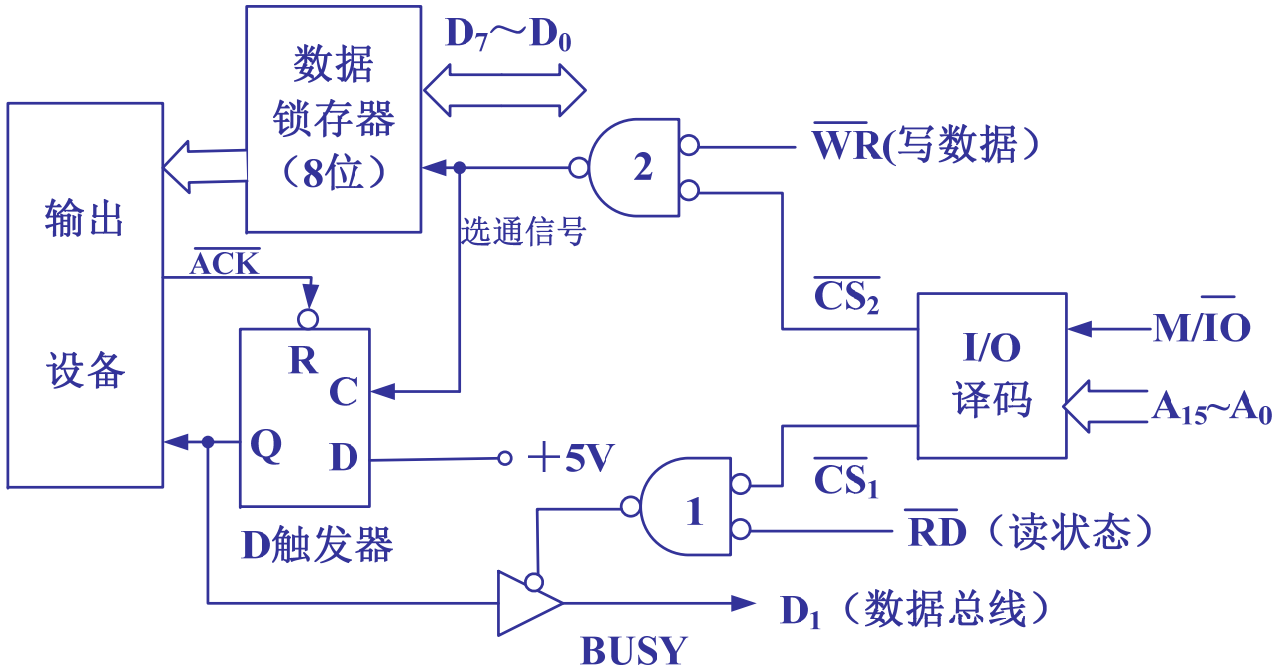
A、查询式输入方式

- 设状态口的地址是PORT_S1,输入数据口的地址是PORT_IN,传送数据的总字节数为COUNT_1:

```
        MOV BX,0          ；初始化地址指针BX
        MOV CX,COUNT_1    ；字节数
READ_S1:IN AL,PORT_S1     ；读入状态位
        TEST AL,01H  ；数据准备好否?
        JZ READ_S1   ；否，循环检测
        IN AL,PORT_IN ；已准备好，读数据
        MOV [BX],AL  ；存到内存缓冲区
        INC BX       ；修改地址指针
        LOOP READ_S1 ；未传送完，继续传送
            …
```



B. 查询式输出方式

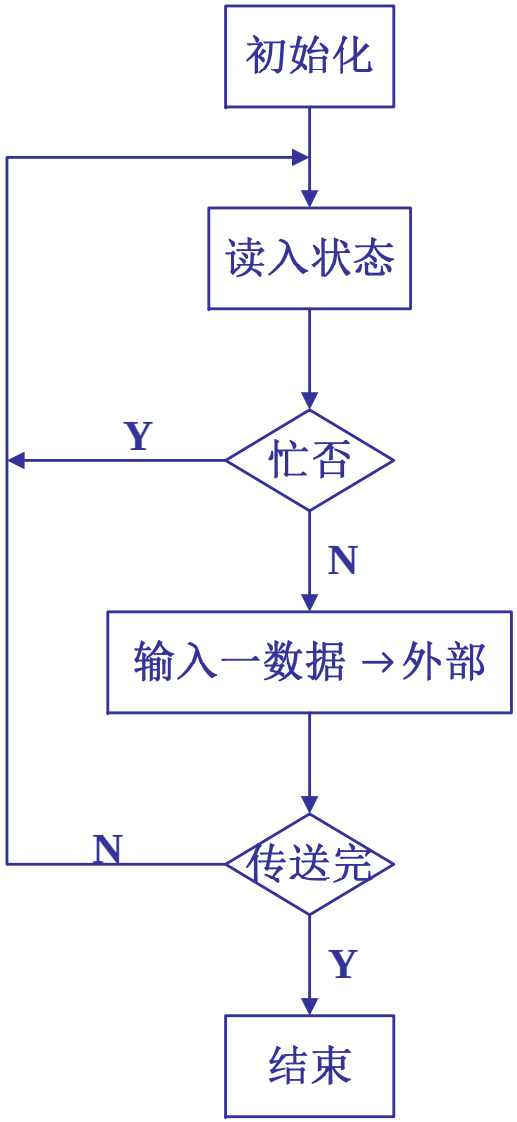


- 当CPU准备向外设输出数据时，先执行IN指令读状态口的信息。三态门开启，从数据总线D1位读入BUSY状态。
  - 若BUSY=1，表示外设在接收上一个数据的忙碌状态。
  - 只有在BUSY=0时，CPU才能向外设输出新的数据。
- 程序检测到D1(BUSY)为0后，便执行OUT指令输出数据。
  - 选通数据锁存器，将数据送向外设。
  - 选通信号的后沿使D触发器翻转，置Q为高电平，将状态口的BUSY置1。
- 输出设备从接口中取出数据后，就送回一个应答信号ACK，将D触发器清0，即置BUSY为0，允许CPU送出下一个数据。

B、查询式输出方式

- 设状态口的地址是PORT_S2,输出数据口的地址是PORT_OUT,传送数据的总字节数为COUNT_2:

```
   MOV CX，COUNT_2          ；传送的字节数
READ_S2: IN AL,PORT_S2      ；读入状态位
   TEST AL,02H              ；忙否?
   JNZ READ_S2              ；忙，循环检测
   MOV AL,输出数据    ；不忙
   OUT PORT_OUT,AL；存到内存缓冲区
   LOOP READ_S2     ；未传送完，循环
         …                  ；已送完
```



2、中断方式

- 用查询方式须反复查询外设的状态。由于许多外设的速度很低，查询等待过程会占去CPU的绝大部分时间，而真正用于数据交换的数据却很少，使CPU的利用率变得很低。为了提高CPU的利用率和进行实时数据处理，CPU常采用中断方式与外设交换数据。
- 采用中断方式后，CPU平时可以执行主程序，只有当输入设备将数据准备好，或输出设备的数据缓冲器已空时，才向CPU发出中断请求。
- CPU响应中断后，暂停执行当前的程序，转去执行管理外设的中断服务程序。其中，用输入或输出指令在CPU和外设之间进行一次数据交换。
- 等输入输出操作完成之后，CPU又回去执行原来的程序。

3、DMA方式

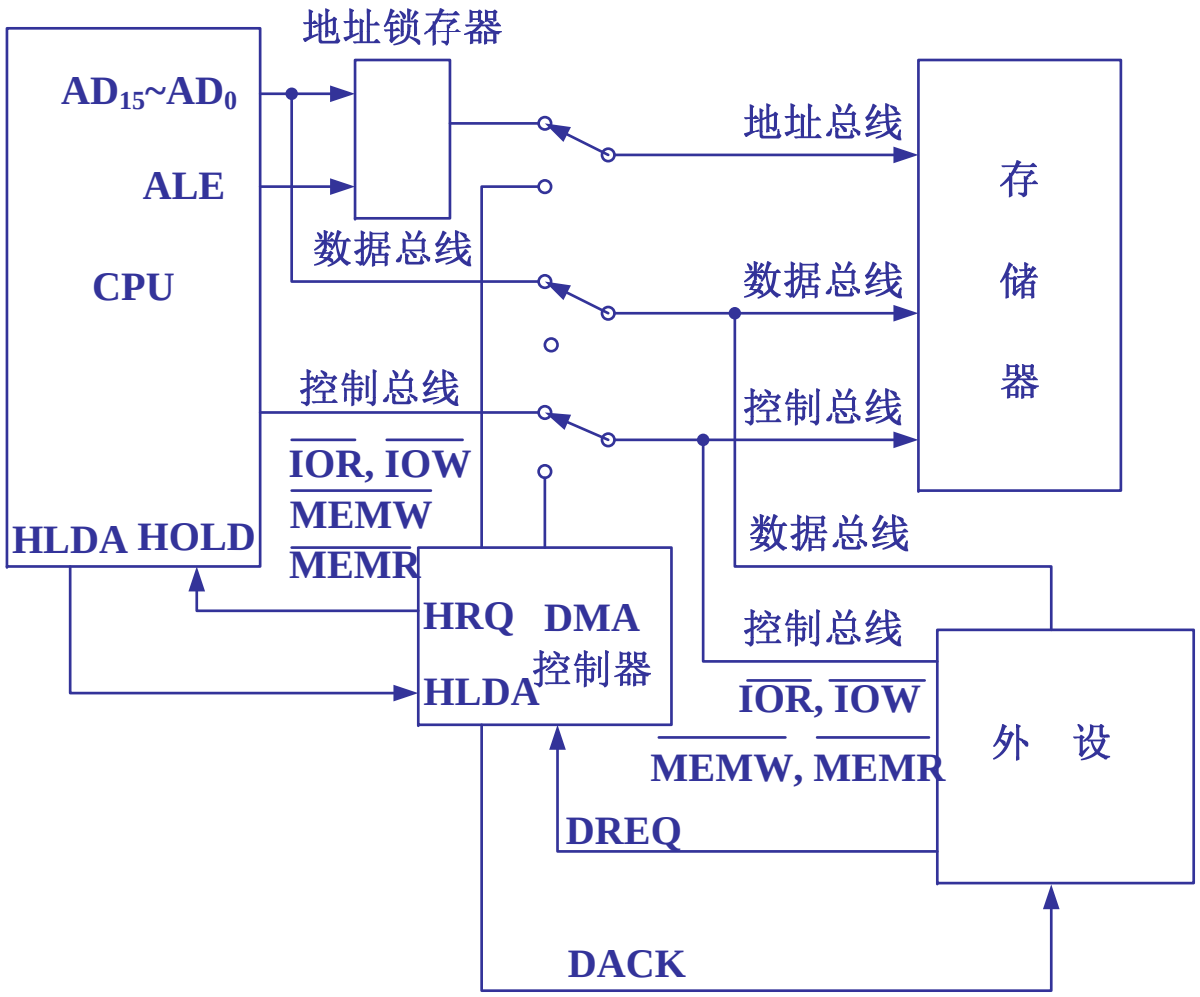
- 中断方式的缺点：无法实现大容量的快速数据交换。
  - 每进行一次传送，CPU都要执行一次中断服务程序，且都要保护和恢复断点，及保护现场等。这些操作与数据传送并无直接联系。
  - 保护和恢复断点使执行部件与总线接口部件不能并行工作，这也会造成数据传送效率的降低。
  - 当CPU与高速I/O设备交换数据，或与外设进行成组数据交换时，中断方式仍显得太慢。
- 为了解决这些问题，提出DMA传送方式。

(1) DMA — Direct Memory Access

- DMA方式也要利用系统的总线来传送数据。
  - 总线由CPU管理，但当外设需要利用DMA方式传送数据时，接口电路可以向CPU要求其出让对总线的控制权，用DMA控制器来取代CPU临时接管总线，控制外设与存储器之间直接进行高速的数据传送，而不要CPU来干预。
  - DMA控制器能给出访问内存所需要的地址信息，自动修改地址指针，设定和修改传送字节数，发出相应的读写信号。DMA传送结束后，它能释放总线，把总线的控制权交换给CPU。
  - DMA方式不需要进行保护和恢复断点及现场之类的额外操作，一旦进入DMA操作，就可直接在硬件的控制下快速完成一批数据的交换任务，数据传送速度取决于外设和存储器的存取速度。

(2) DMA方式控制器的连线和操作

- DMA控制器经常与磁盘驱动器、磁带机和数据采集卡等配合使用，来实现这些外设与内存之间的高速数据交换。
- Intel 8237A是一种较常用的DMA控制器，被用在PC机中，以提高系统存取软驱的速度。



7.2 可编程计数器/定时器8253及其应用

0. 概述
1. 8253的外部引脚及内部结构
2. 8253的工作方式
3. 8253的控制字
4. 8253的应用举例

0. 概述

定时信号的产生

- 软件编程的方法
- 硬件的方法

- 软件定时：设计一个延时子程序，子程序中全部指令执行时间的总和就是该子程序的延时时间。

特点：简单，易实现，需要了解延时子程序中每条指令的执行时间；定时时间不太精确；仅适用于延时时间较短、重复次数有限的场合；占用CPU大量的时间，使CPU的利用率降低。

硬件的方法：利用专用的硬件定时/计数器，在简单软件控制下产生准确的延时时间。

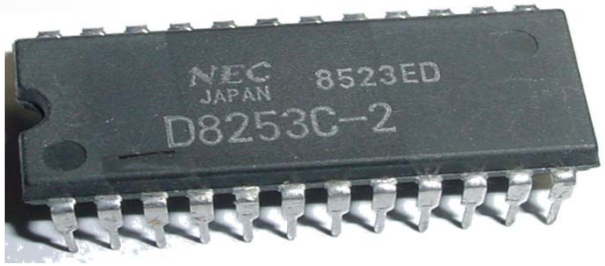
基本原理：

- 通过软件确定定时/计数器的工作方式、设置计数初值并启动计数器工作
- 当计数到给定值时，便自动产生定时信号
- 成本不高，程序简单，几乎不占用CPU资源
- 适合长时间、多次重复的定时，也可用于延时时间较短的场合

定时/计数器的计数方式：

- 加法计数器和减法计数器
- 加法计数器是每有一个计数脉冲就加1，当加到预先设定的计数值时，产生一个定时信号
- 减法计数器是在送入计数初值后，每来一个计数脉冲就减1，减到0时产生一个定时信号输出
- 可编程定时/计数器8253就是一个减法计数器，它是Intel公司专为80×86系列配套开发的16位可编程定时/计数器芯片。

1. 8253的外部引脚及内部结构



三通道的16位定时/计数器;

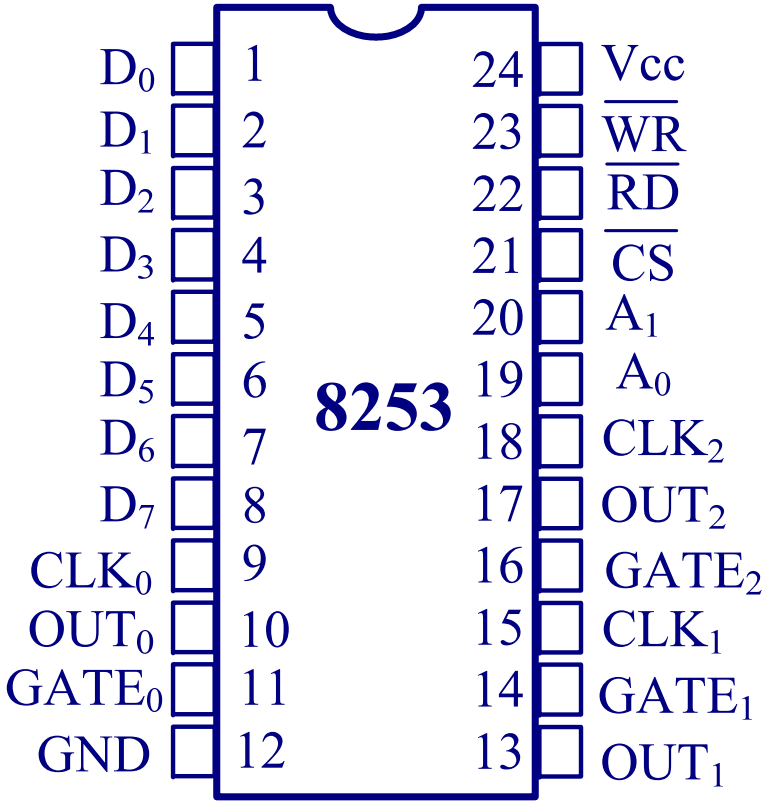
计数器的工作方式及计数常数分别由软件编程选择;

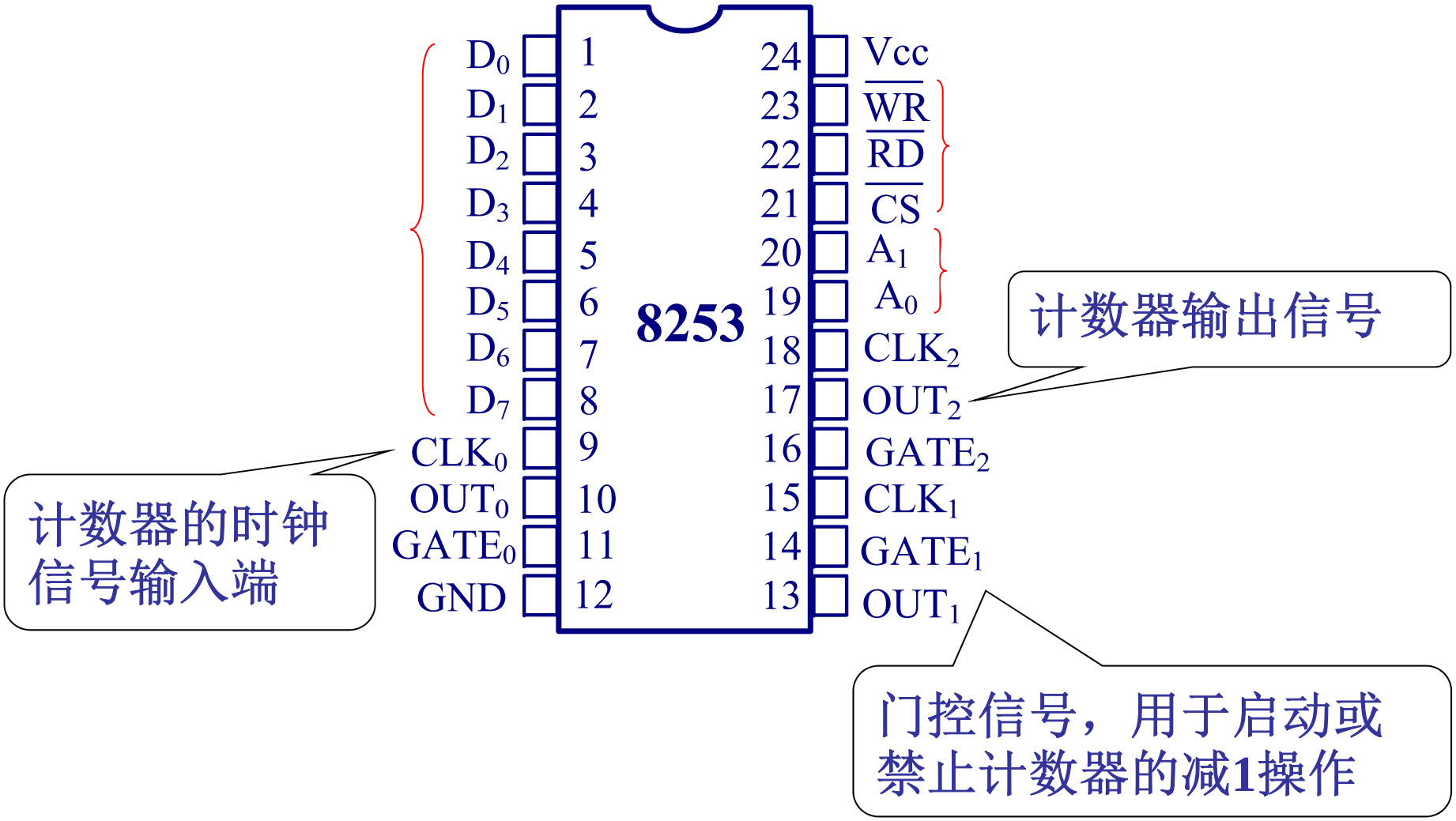
可进行二进制或十进制计数或定时操作;

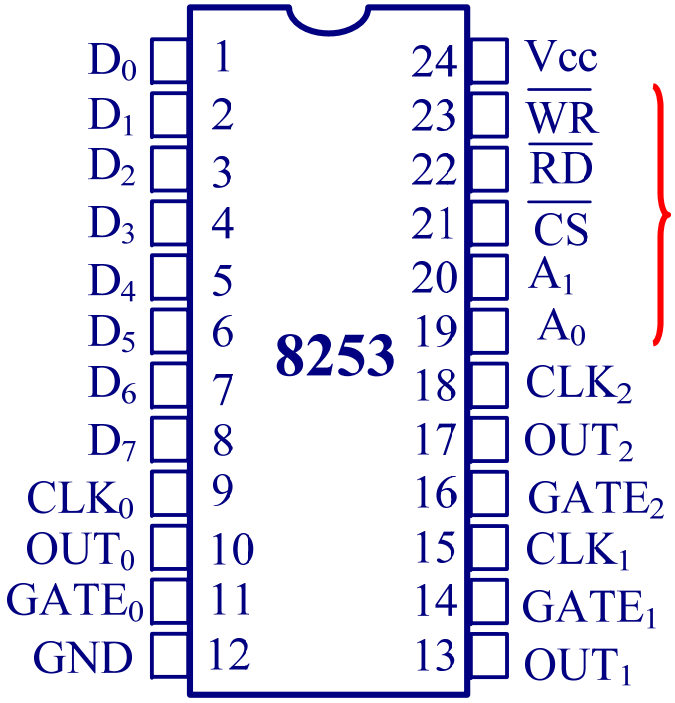
最高计数频率为2 MHz;

使用单电源+5 V供电;

输入/输出均与TTL电平兼容。

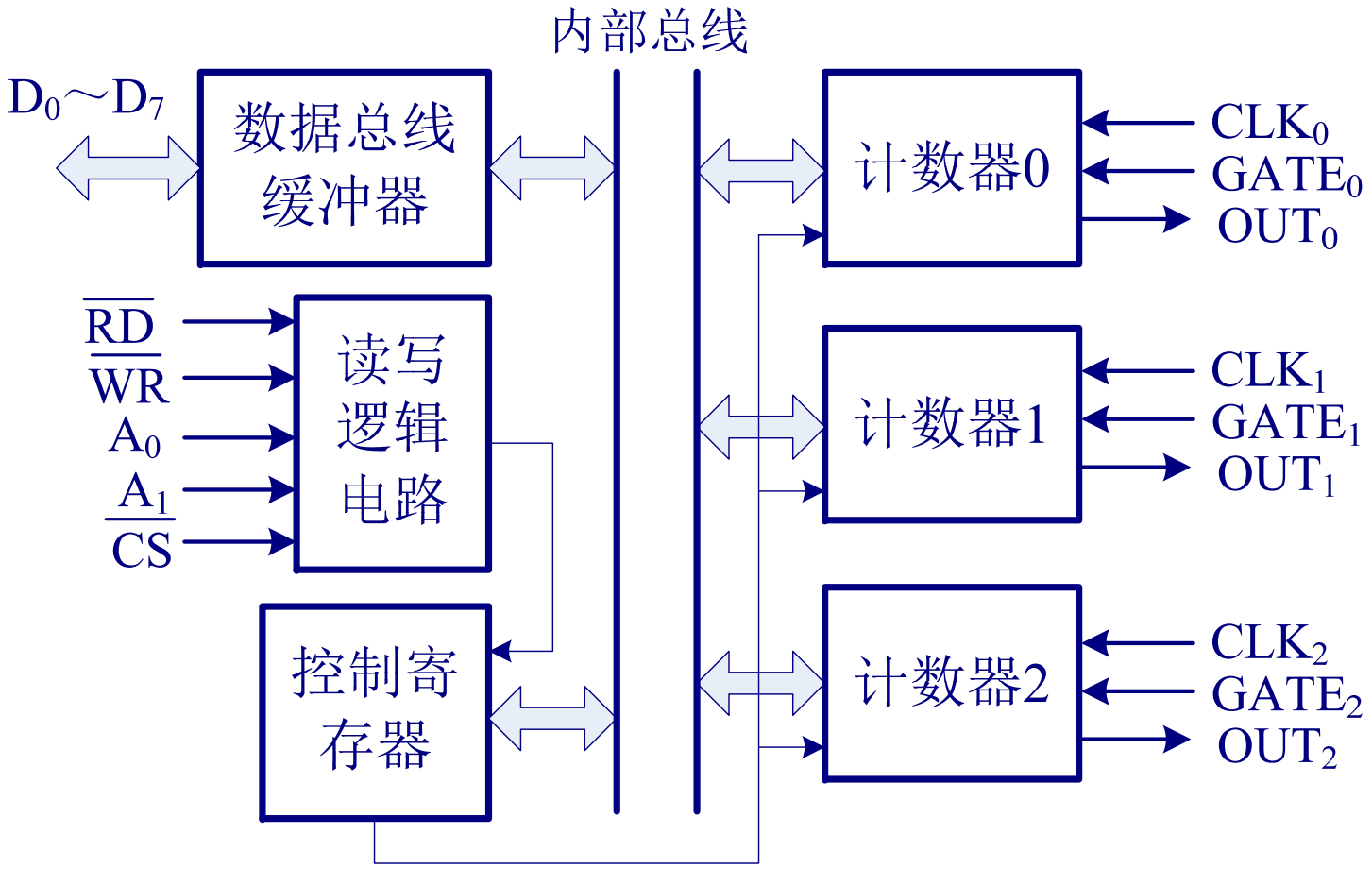



8253
D0 1
D1 2
D2 3
D3 4
D4 5
D5 6
D6 7
D7 8
CLK0 9
OUT0 10
GATE0 11
GND 12
24 Vcc
23 WR
22 RD
21 CS
20 A1
19 A0
18 CLK2
17 OUT2
16 GATE2
15 CLK1
14 GATE1
13 OUT1
计数器输出信号
计数器的时钟信号输入端
门控信号，用于启动或禁止计数器的减1操作



$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	$A_0 A_1$	定义
0	1	0	00	写入计数器0
0	1	0	01	写入计数器1
0	1	0	10	写入计数器2
0	1	0	11	写入控制寄存器
0	0	1	00	读计数器0
0	0	1	01	读计数器1
0	0	1	10	读计数器2
0	0	1	11	无操作
1	×	×	××	禁止使用
0	1	1	××	无操作

内部结构


内部总线
D0～D7
数据总线缓冲器
RD
WR
A0
A1
CS
读写逻辑电路
控制寄存器
计数器0
计数器1
计数器2
CLK0
GATE0
OUT0
CLK1
GATE1
OUT1
CLK2
GATE2
OUT2

(1) 计数器

- 计数器0、1和2：3个相同的16位减1计数器
- 互相独立按各自的方式进行工作
- 每个计数器都包括一个16位的初值寄存器、一个计数执行单元和一个输出锁存器

计数器工作过程

- 当置入初值后，计数执行单元开始对输入脉冲CLK进行减1计数，在减到0时，从OUT端输出一个信号
- 整个过程可以重复进行
- 计数器既可按二进制计数，也可按十进制计数
- 在计数过程中，计数器还受到门控信号GATE的控制
- 在不同的工作方式下，计数器的输入CLK、输出OUT和门控信号GATE之间的关系将会不同

(2) 控制寄存器

- 存放操作方式控制字
- 通过向控制寄存器中写入所需的控制字，决定计数器的工作方式
- 控制字是在8253初始化时用输出指令写入控制寄存器的
- 该寄存器只能写入，不能读出。

(3) 数据总线缓冲器

- 8位，双向，三态
- 用于8253和CPU数据总线之间的接口
- CPU通过该数据缓冲器对8253进行读/写。

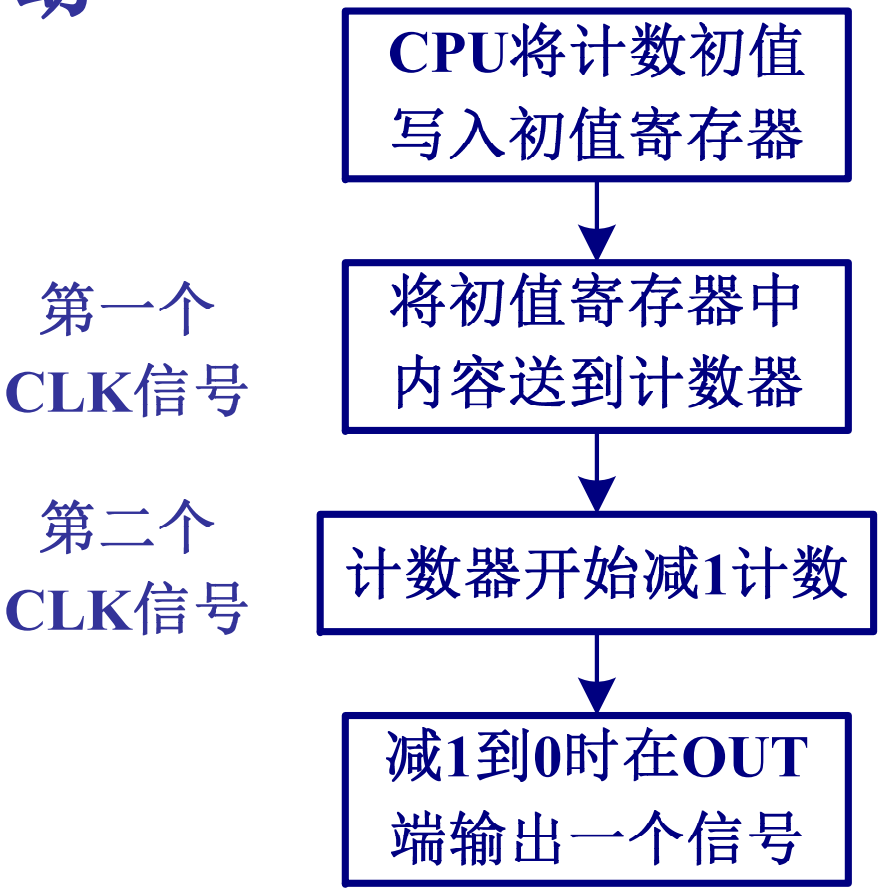
(4) 读/写控制逻辑

- 选片信号有效时，读/写控制逻辑从系统总线接收输入信号，经过逻辑组合，产生对各部分的控制信号
- 当选片信号无效(高电平)时，数据总线缓冲器处于高阻态，8253与总线断开，CPU不能对其进行读/写操作。

计数启动方法

- 8253计数器的计数过程，可以由程序指令启动，称为软件启动
- 也可由外部电路信号启动，称为硬件启动

(1) 软件启动



从CPU写入计数初值到计数结束，实际的CLK脉冲个数比编程写入的计数初值*N*要多一个，即*N*+1（不可避免）。

(2) 硬件启动

- 写入计数初值后并不启动计数，而是在门控信号GATE由低电平变高后，再经CLK信号的上升沿采样，之后在该CLK的下降沿才开始计数。
- 由于GATE信号与CLK信号不一定同步，故在极端情况下，从GATE变高到CLK采样之间的延时可能会经历一个CLK脉冲宽度，因此在计数初值与实际的CLK脉冲个数之间也会有一个时钟脉冲的误差。

计数方式

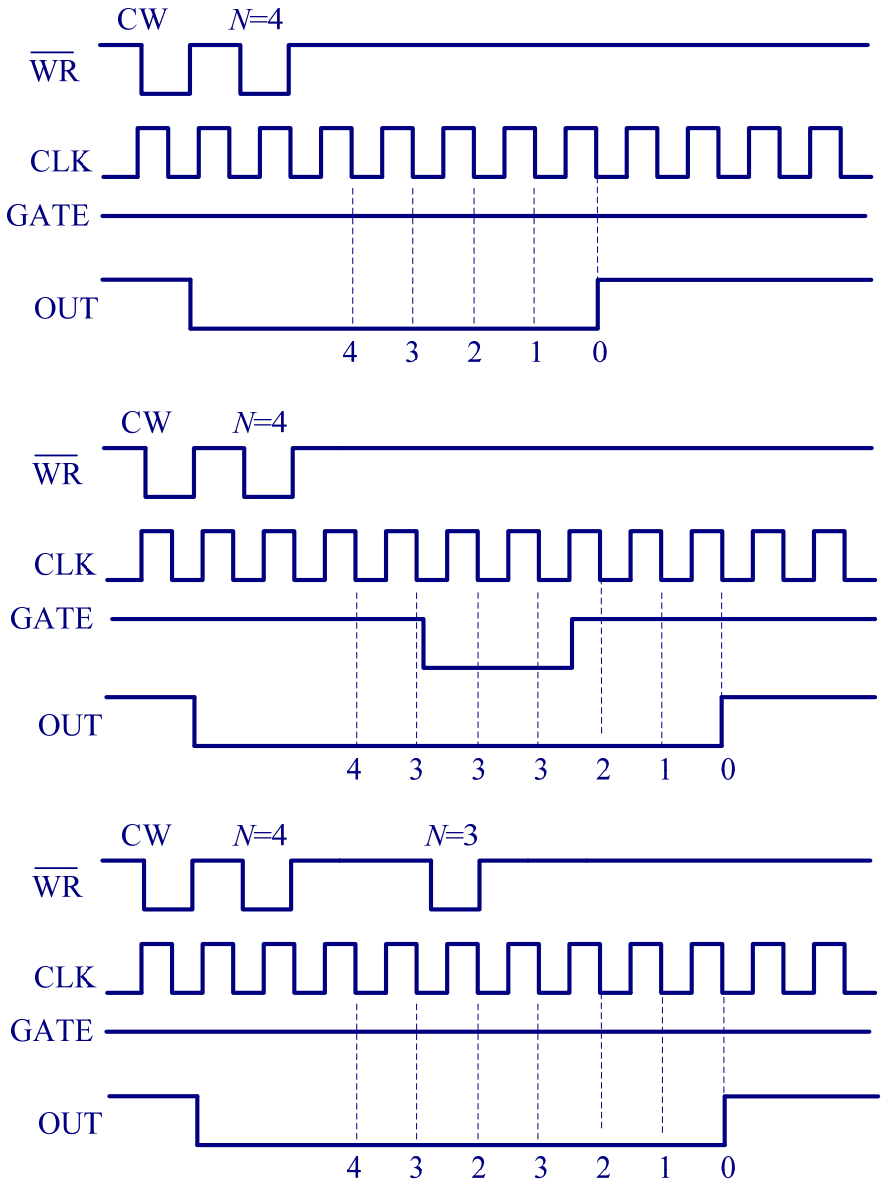
- 不自动重复的计数方式：计数器每启动一次只工作一个周期(即从初值减到0)
- 自动重复的计数方式：一旦计数启动，只要门控信号GATE保持高电平，计数过程就会自动周而复始地重复下去，这时OUT端可以产生连续的波形输出。
- 在自动重复计数方式下，达到稳定状态后，因启动造成的实际计数值和计数初值之间的误差就不再存在。

2．8253的工作方式

方式	启动计数	中止计数	自动重复	更新初值	输出波形
0	软件(写入初值)	GATE=0	否	立即有效	延时时间可变的上跳沿
1	硬件(GATE正跳变)		否	下一轮有效	宽度为$N\times T_{CLK}$的单一负脉冲
2	软件(写入初值) 硬件(GATE正跳变)	GATE=0	是	下一轮有效	周期为$N\times T_{CLK}$、宽度为T_{CLK}的连续负脉冲
3	软件(写入初值) 硬件(GATE正跳变)	GATE=0	是	下半轮有效	周期为$N\times T_{CLK}$的连续方波
4	软件(写入初值)	GATE=0	否	下一轮有效	宽度为T_{CLK}的单一负脉冲
5	硬件(GATE正跳变)		否	下一轮有效	宽度为T_{CLK}的单一负脉冲

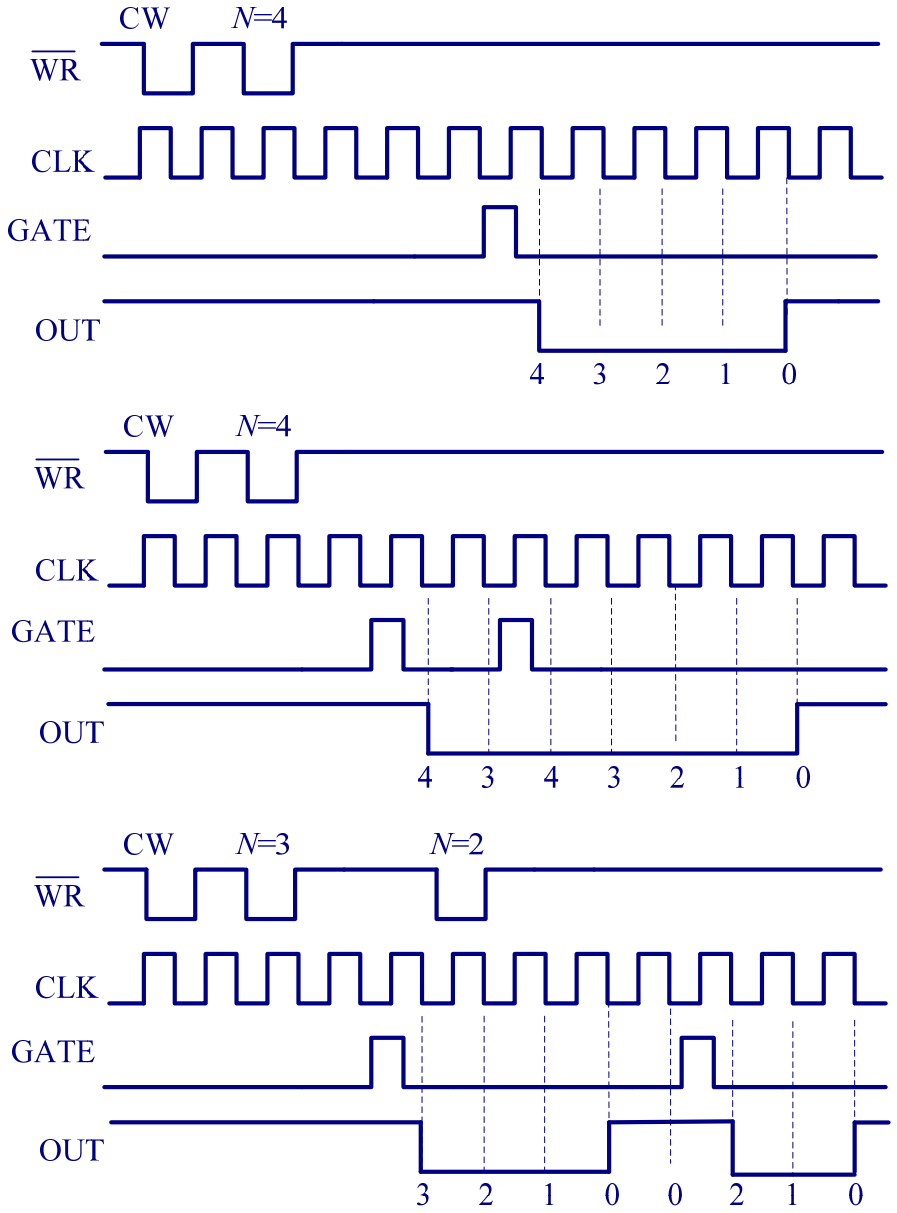
方式0工作波形

(计数结束中断方式)



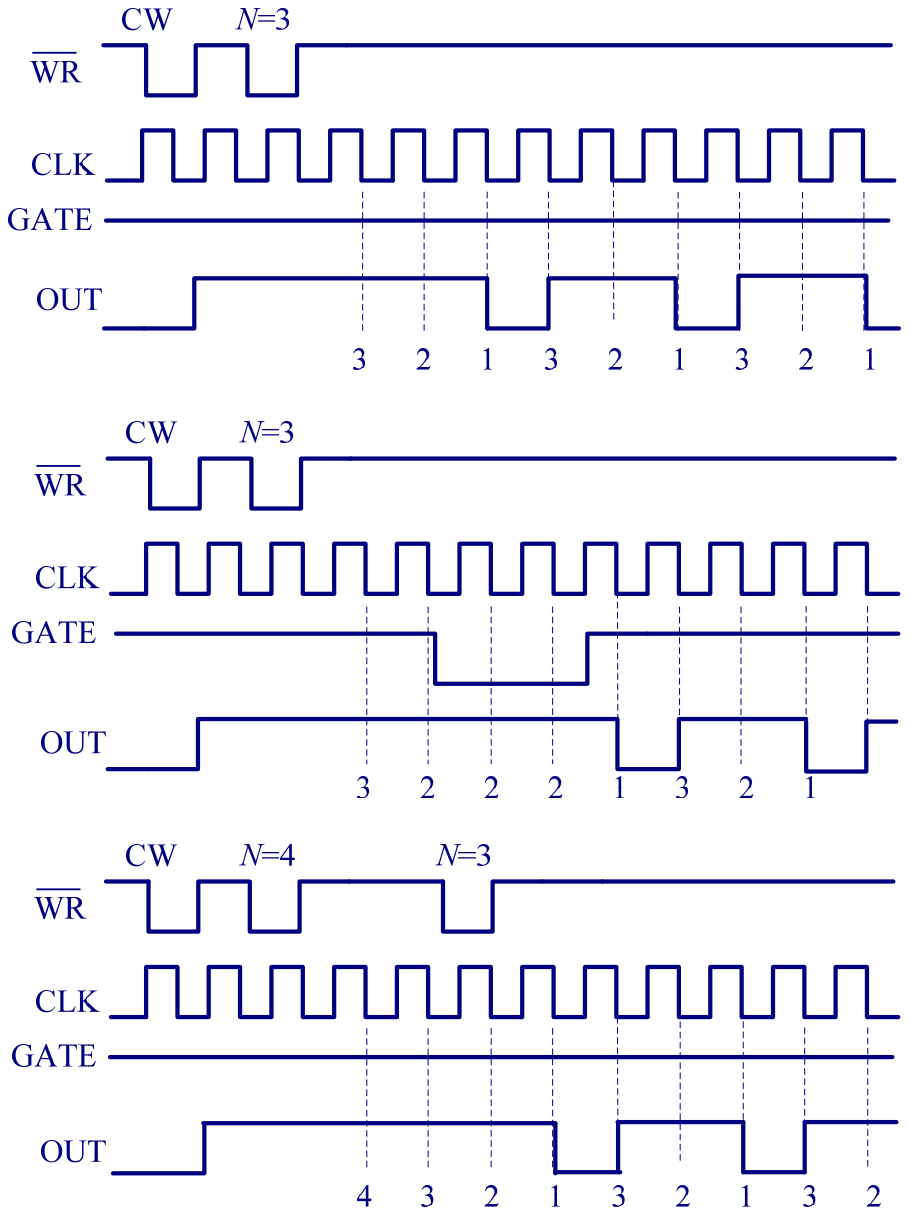
方式1工作波形

(可重复触发的单稳态触发器)



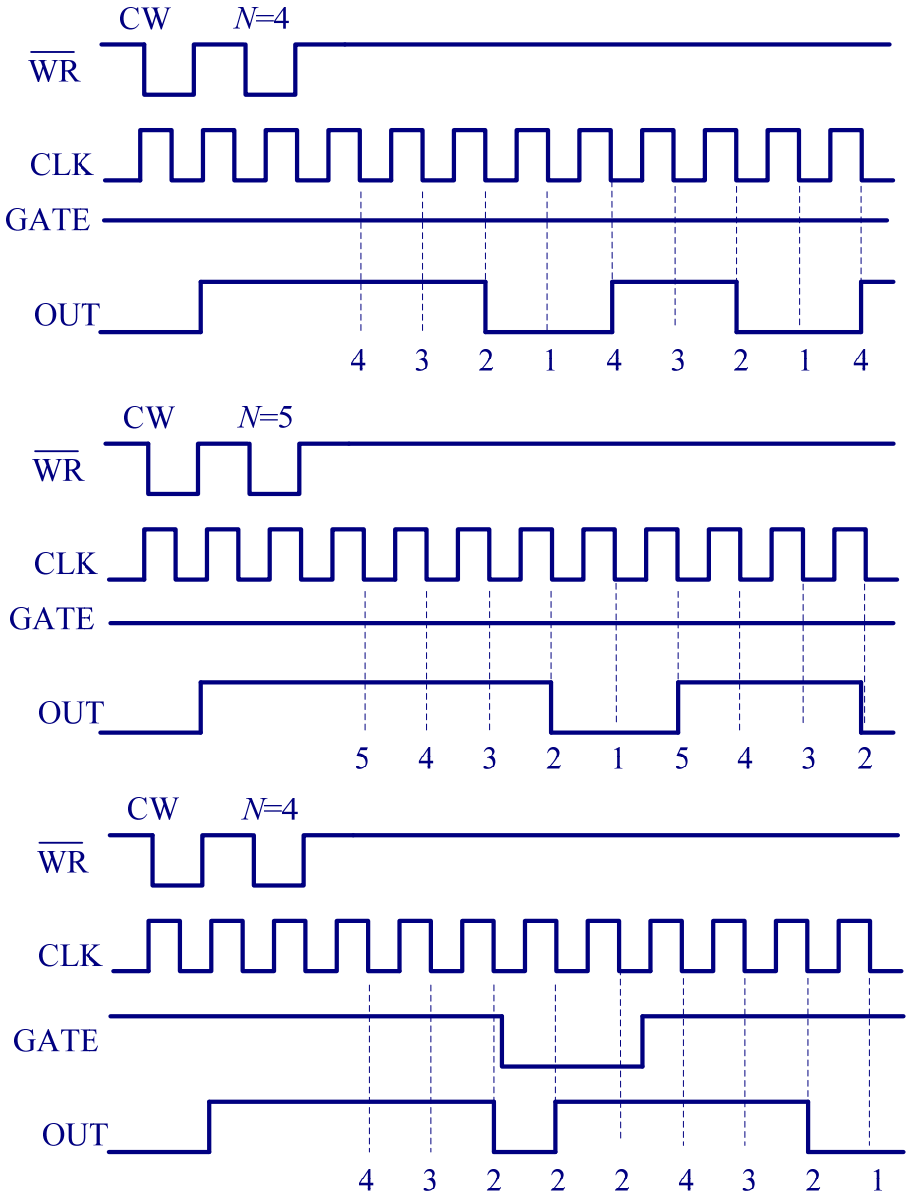
方式2工作波形

(频率发生器)



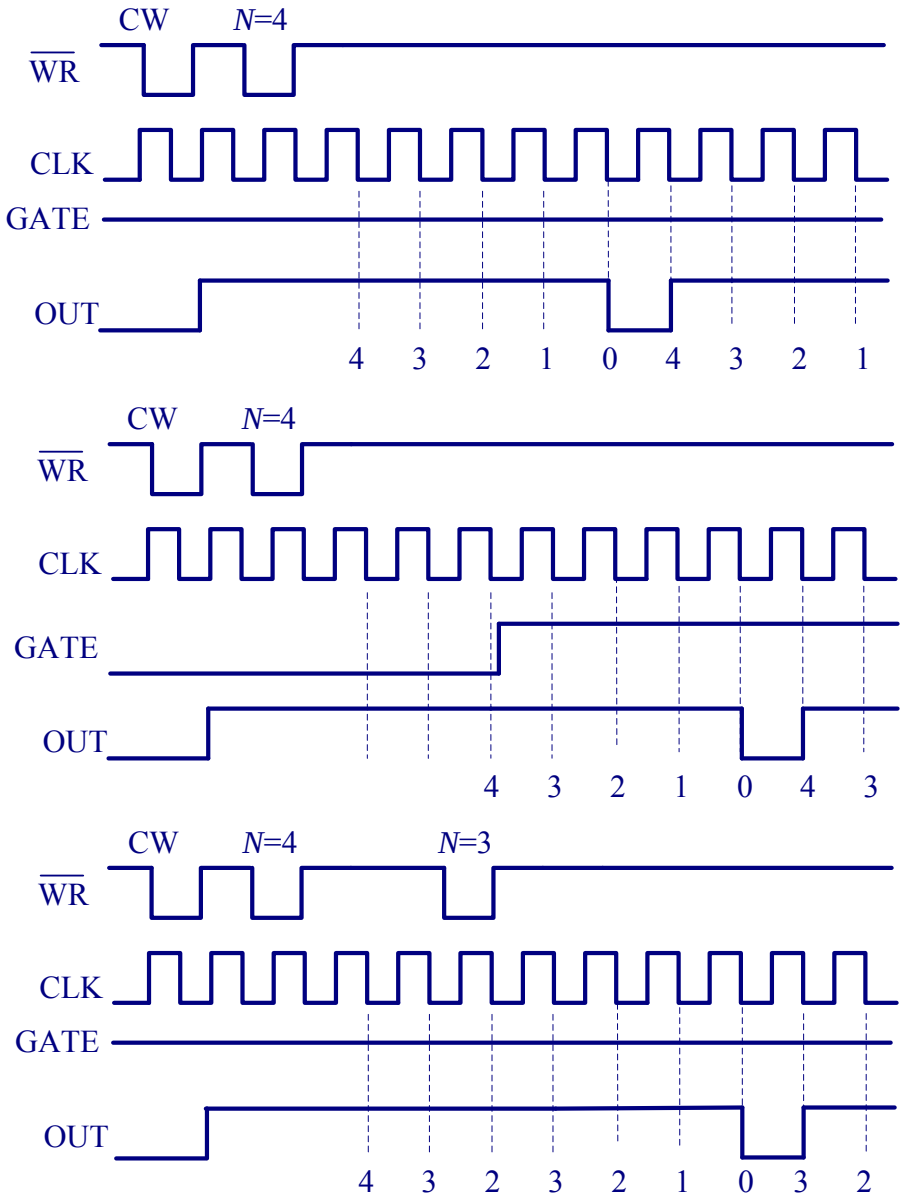
方式3工作波形

(方波发生器)



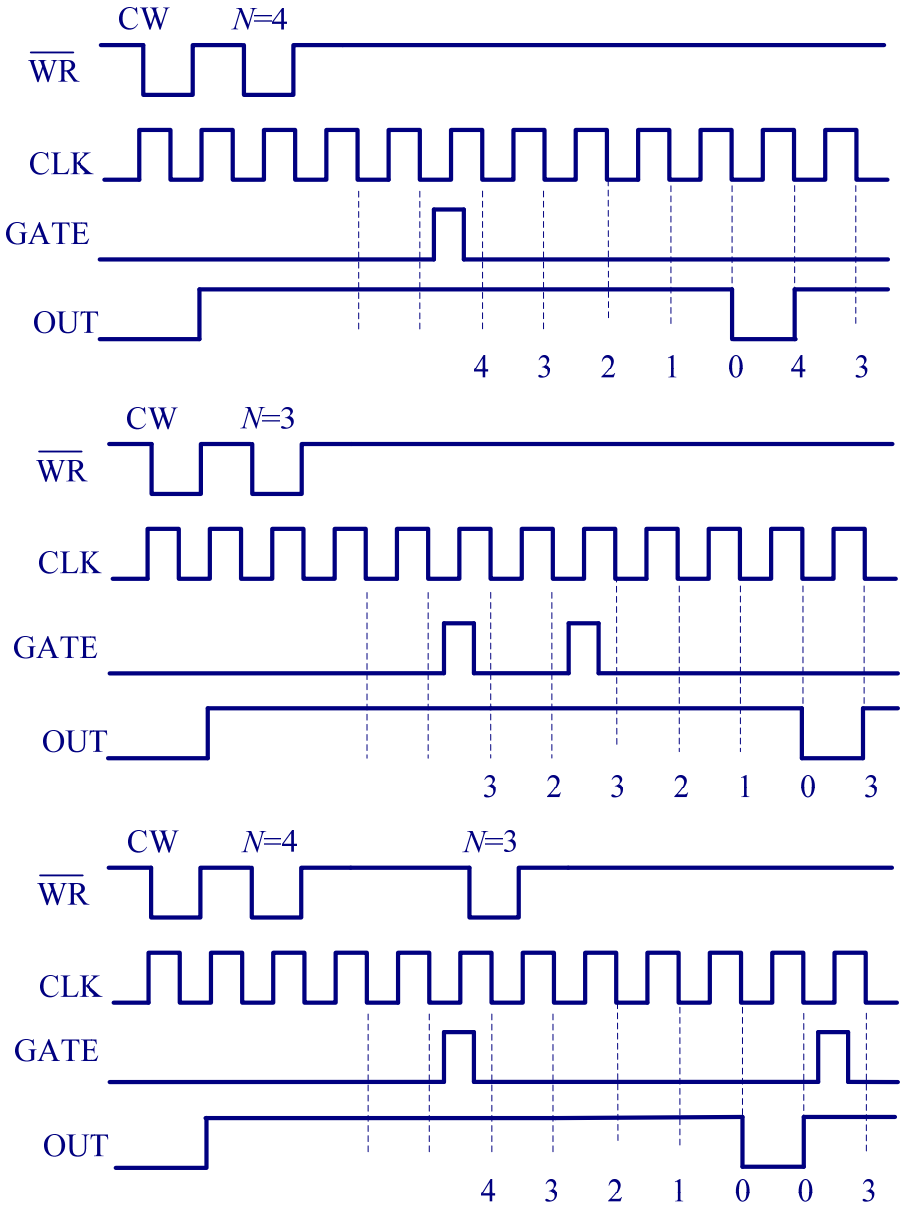
方式4工作波形

(软件触发选通)

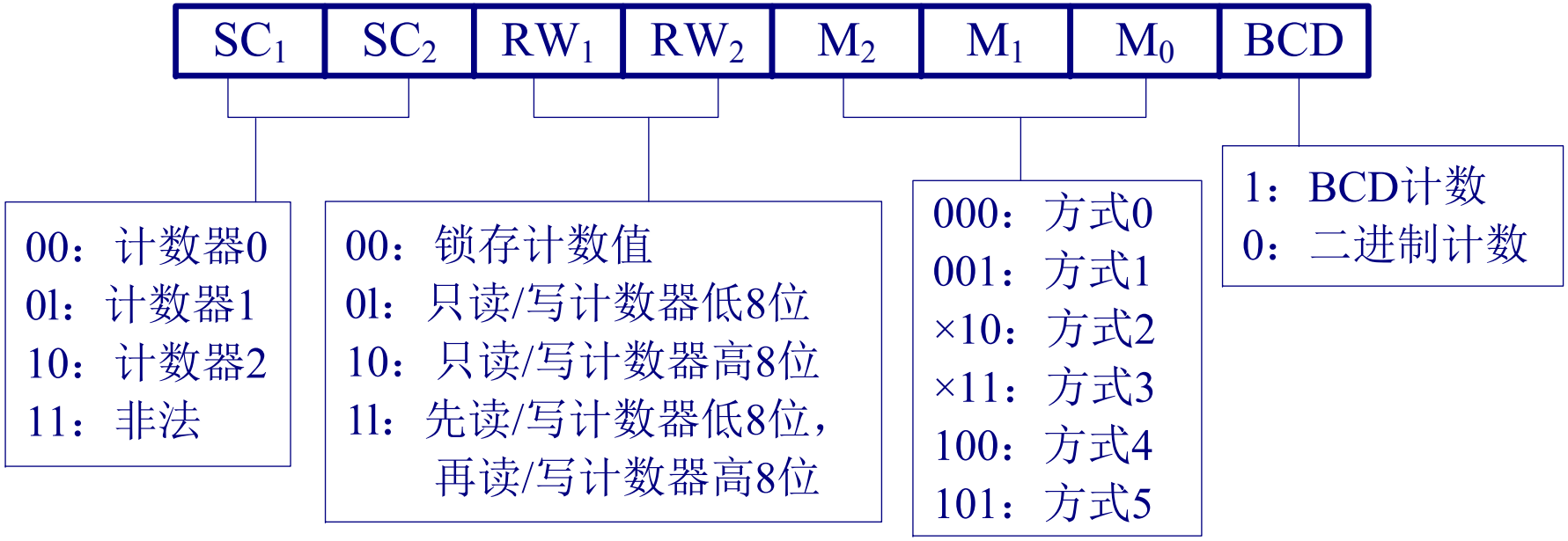


方式5工作波形

(硬件触发选通)



3. 8253的控制字


SC1 SC2 RW1 RW2 M2 M1 M0 BCD
00：计数器0
01：计数器1
10：计数器2
11：非法
00：锁存计数值
01：只读/写计数器低8位
10：只读/写计数器高8位
11：先读/写计数器低8位，
再读/写计数器高8位
000：方式0
001：方式1
×10：方式2
×11：方式3
100：方式4
101：方式5
1：BCD计数
0：二进制计数

8253的初始化

- 写入方式控制字
  - 三个通道用的控制字端口地址是相同的
  - 三个控制字写入后存入通道对应的寄存器中
- 写入计数初始值
- 读计数值（计时）
  - 以普通对计数器端口读的方法取得当前计数值
  - 锁存计数器的当前计数值

例：某8086微机系统中，8253的三个计数器端口地址分别为　3F0H，3F1H，3F2H，控制字寄存器端口地址为3F3H，要求通道0工作于方式3，且计数初值n = 1234 。则初始化程序为：

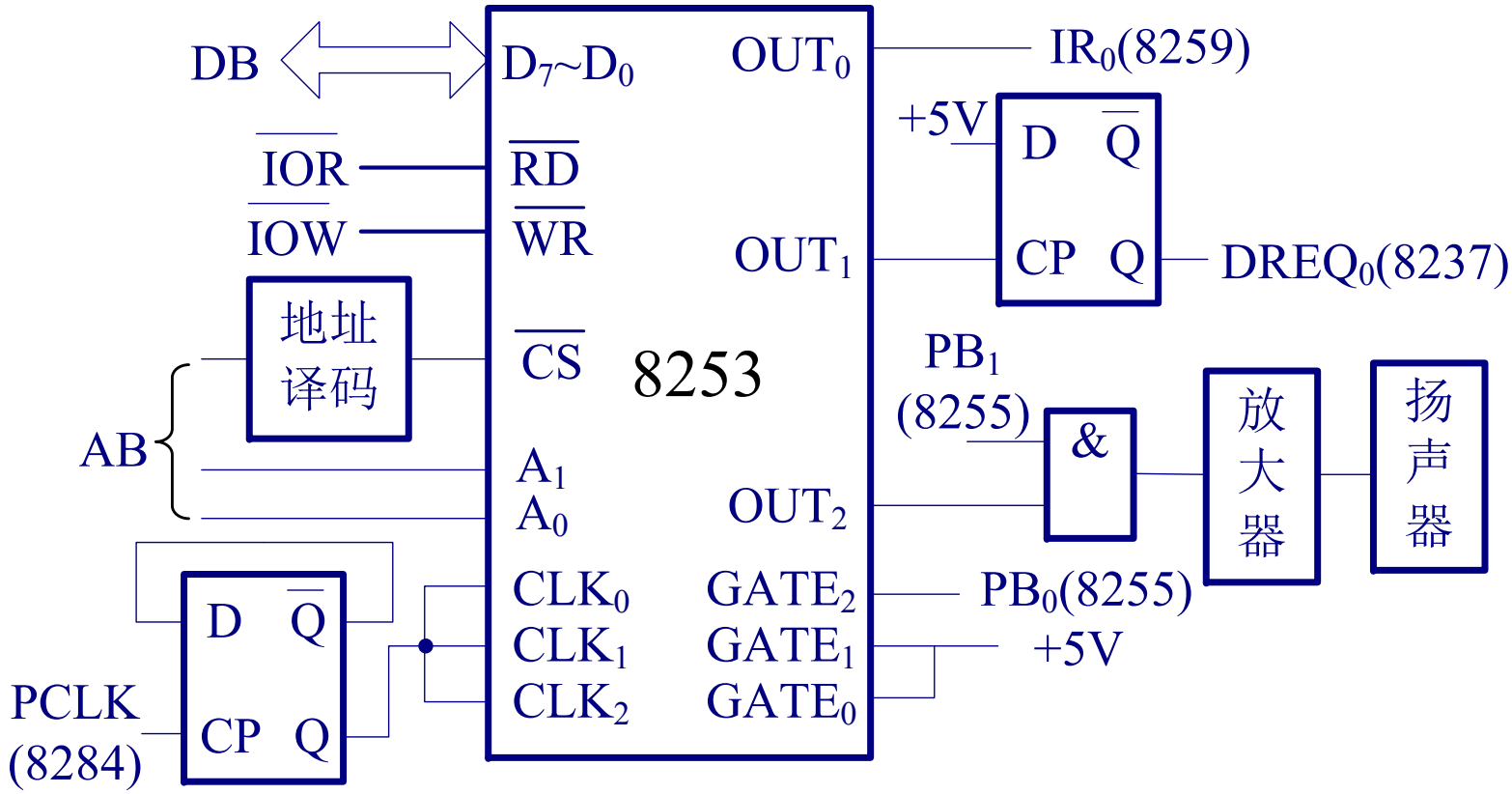
```
MOV AL，00110111B；控制字
MOV DX，3F3H　　；控制端口
OUT DX，AL　　；送控制字
MOV DX，3F0H　；通道0口的地址
MOV AL，34H　　；计数值低字节
OUT DX，AL　；写低字节
MOV AL，12H　；计数值高字节
OUT DX，AL　；写高字节
```

例：接上例，8253的端口地址分别为3F0H，3F1H，3F2H，3F3H，要求通道0工作于方式3，且计数初值n = 1234 。读当前计数值的程序为：

```
MOV AL，00000111B；控制字
MOV DX，3F3H ；控制端口
OUT DX，AL     ；送控制字
MOV DX，3F0H ；通道0口的地址
IN AL，DX      ；读低字节
MOV AH, AL     ；保存
IN AL，DX      ；读高字节
XCHG AH, AL    ；存入AX
MOV CX, 1234+1 ；求计数值
SUB CX, AX     ；得到计数值
```

4. 8253的应用举例

(1) IBM PC机中8253的应用



- PC机系统板上使用了一片8253定时/计数器
- 计数器1(CNT1)用于DRAM的定时刷新
- 计数器0(CNT0)用来为系统的电子钟提供时间基准，它的输出端作为系统的时钟中断源，接到8259的IR0端，用以产生计时脉冲
- 计数器2(CNT2)主要用来作为机内扬声器的音频信号源，可输出不同频率的方波信号
- 接口地址采用了部分译码方式，占用外设地址40H~5FH。在编程时，只使用这个地址范围中的40H~43H。
- 3个计数器的输入时钟频率均为1.19 MHz。

- 计数器0初始化为方式3，产生周期的方波信号，计数初值为0000H。OUT_0输出方波信号的频率为1.19 MHz/65536=18.2 Hz。由于OUT_0在系统主板上与8259的中断请求输入线IR_0相连接，所以每秒将会产生18.2次中断请求，该中断请求用于维护系统的日历钟

- 计数器1初始化为方式2，计数初值取18，18/1.19 MHz=15μs，即每15μs对动态存储器刷新一次

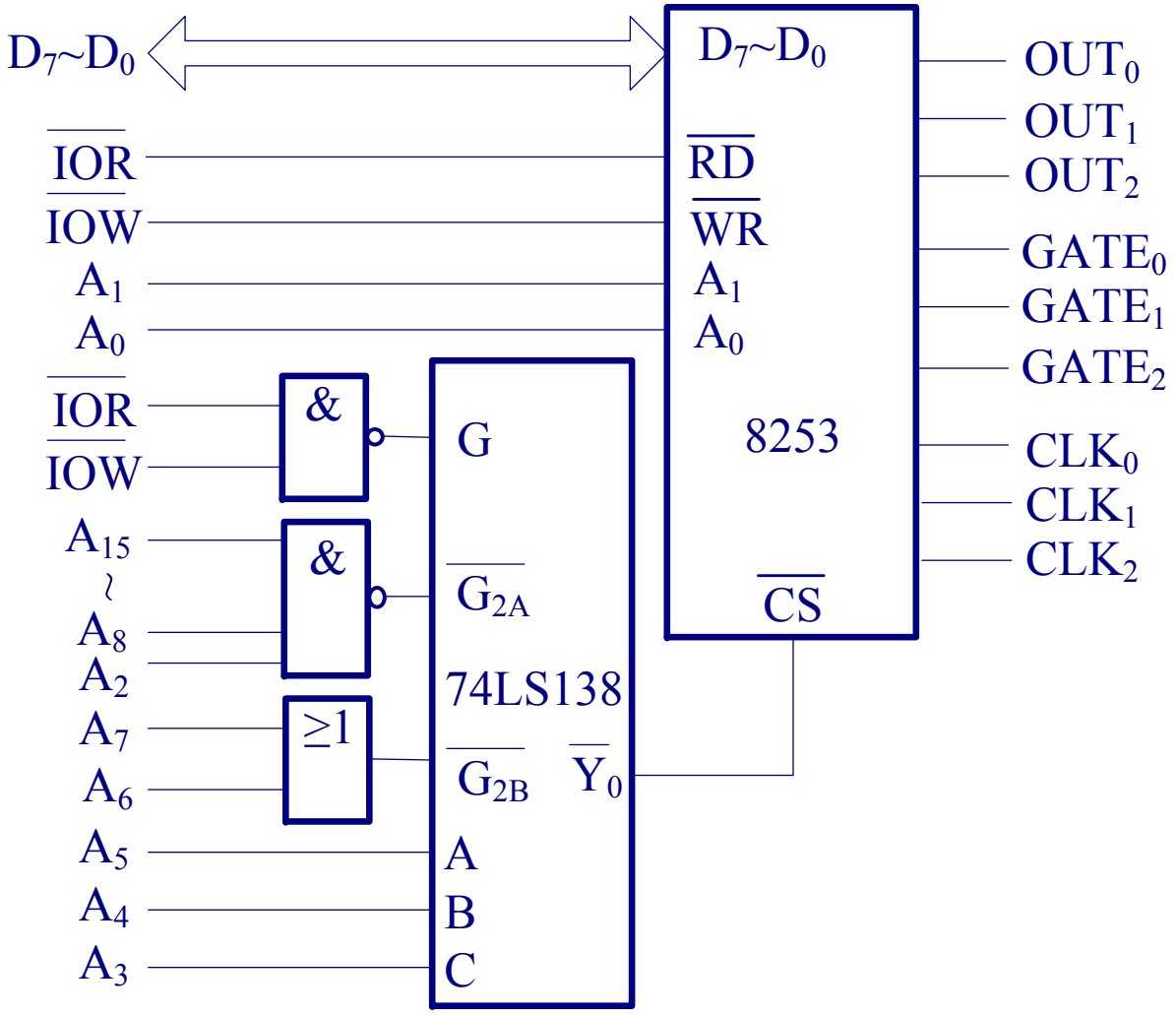
- 计数器2初始化为方式3，控制扬声器发出频率为1 kHz的声音，故取计数初值为1190。在IBM PC中，要使扬声器发声，还必须使8255的PB_1和PB_0输出高电平。8255的B口地址为61H

```
; CNT0初始化
  MOV AL, 36H   ; 选择计数器0，写双字节计数值
                ; 方式3，二进制计数
  OUT 43H, AL   ; 控制字写入控制寄存器
  MOV AL, 0     ; 选最大计数值(65 536)
  OUT 40H, AL   ; 写低8位计数值
  OUT 40H, AL   ; 写高8位计数值
; CNT1初始化
  MOV AL, 54H   ; 选择计数器1，低8位单字节计数值
                ; 方式2，二进制计数
   OUT 43H, AL
   MOV AL, 18
   OUT 41H, AL    ; 计数值写入计数器1
```

```
; CNT2初始化
  MOV AL, 0B6H    ; 选择计数器2，双字节计数值
                  ; 方式3，二进制计数
  OUT 43H, AL
  MOV AX, 1190
  OUT 42H, AL     ; 送低字节到计数器2
  MOV AL, AH      ; (AH)←高字节计数值
  OUT 42H, AL     ; 高8位计数值写入计数器2
  IN AL, 61H      ; 读8255的B口
  MOV AH, AL      ; 将B口内容保存
  OR AL, 03       ; 使PB0=PBl=1
  OUT 61H, AL     ; 使扬声器发声
  …
  MOV AL, AH      ; 恢复8255B口状态
  OUT 61H, AL
```

(2) 脉冲发生器

要求：3个计数器CLK频率均为2MHz。要求计数器0在定时100μs后产生中断请求；计数器1用于产生周期为10μs的对称方波；计数器2每1ms产生一个负脉冲。编写8253的初始化程序。



图中地址总线信号A15~A2经译码电路产生片选信号选中8253。根据图中译码器的连接方式可知，该8253的接口地址范围为FF04H~FF07H。FF04H对应于计数器1，FF05H对应于计数器2，FF06H对应于计数器3，FF07H对应于控制寄存器。

根据要求可知，计数器0应工作于方式0，计数初值=100μs/0.5μs=200(CLK的周期=0.5μs)；计数器1应工作于方式3，计数初值=10μs/0.5μs=20；计数器2应工作于方式2，计数初值=1 ms/0.5μs=2000。以下是8253的初始化程序。

```
INIT8253: MOV DX, 0FF07H
   MOV AL, 10H    ；计数器0，只写计数值低8位，方式0，二进制计数
   OUT DX, AL
   MOV AL, 56H    ；计数器1，只写计数值低8位，方式3，二进制计数
   OUT DX, AL
   MOV AL, 0B4H  ;计数器2,先写高8位,再写低8位,方式2,二进制计数
   OUT DX, AL
   MOV DX, 0FF04H
   MOV AL, 200    ；计数器0的计数初值
   OUT DX, AL
   MOV DX, 0FF05H
   MOV AL, 20     ；计数器1的计数初值
   OUT DX, AL
   MOV DX, 0FF06H
   MOV AX, 2000    ；计数器2的计数初值
   OUT DX, AL
   MOV AL, AH
   OUT DX, AL
```

7.3 可编程外围接口芯片8255A及其应用

0. 概述
1. 8255的外部引脚及内部结构
2. 8255的工作方式
3. 8255的控制字及状态字
4. 8255的应用举例

0. 概述

- 并行接口：一次可以同时传送一个数据的所有位
- 并行接口的数据传送方向：一是单向传送(只作为输入口或输出口)，另一种是双向传送(既可作为输入口，也可作为输出口)
- 并行接口的可编程性：

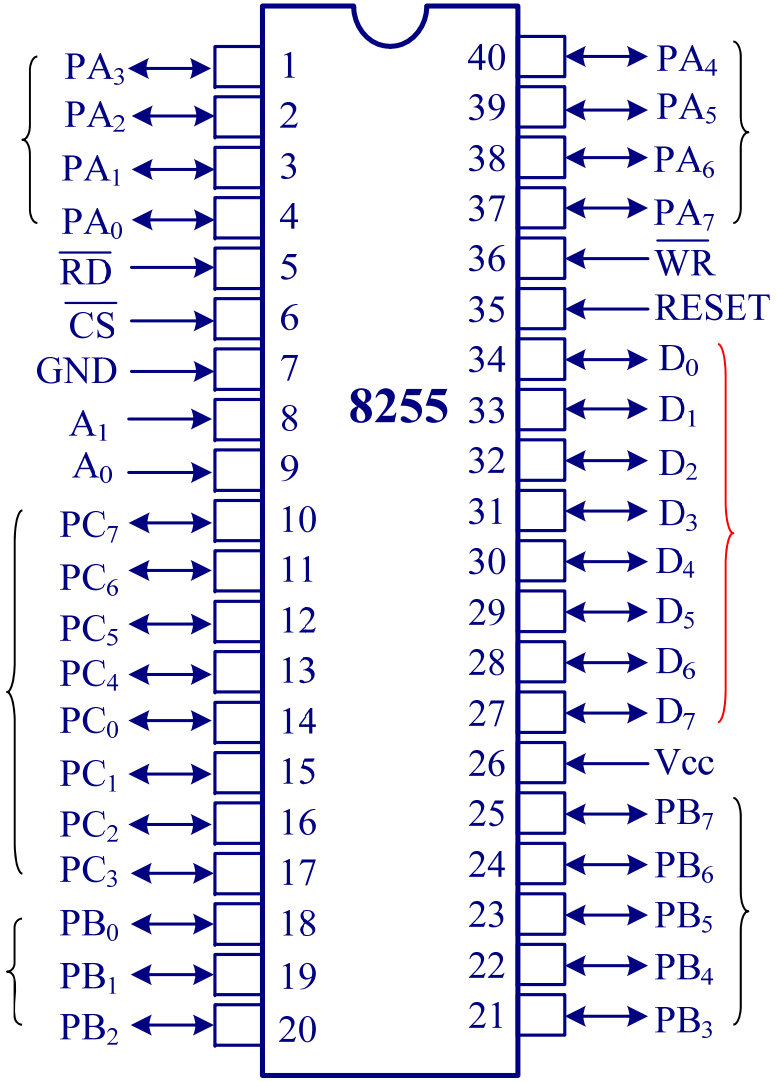
  不可编程：简单(如锁存器或三态门)

  可编程：复杂，功能完善的并行接口中一般都包括输入/输出数据寄存器、控制寄存器(存放控制命令)、状态寄存器(保存当前工作状态)和总线缓冲器等部件

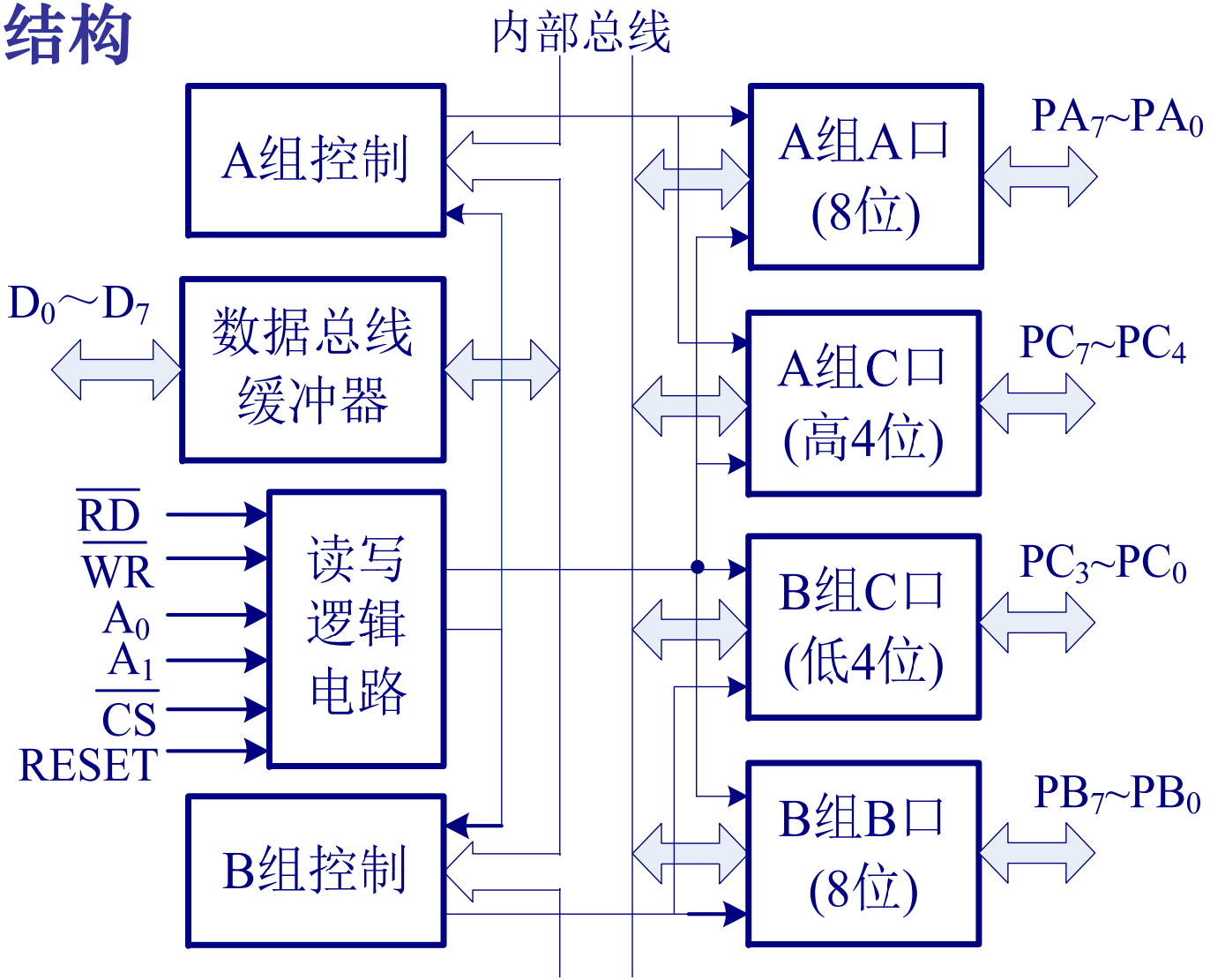
8255是Intel公司为80×86系列CPU配套的可编程并行接口芯片。通用性较强；使用灵活；典型的可编程并行接口。

1. 8255的外部引脚及内部结构

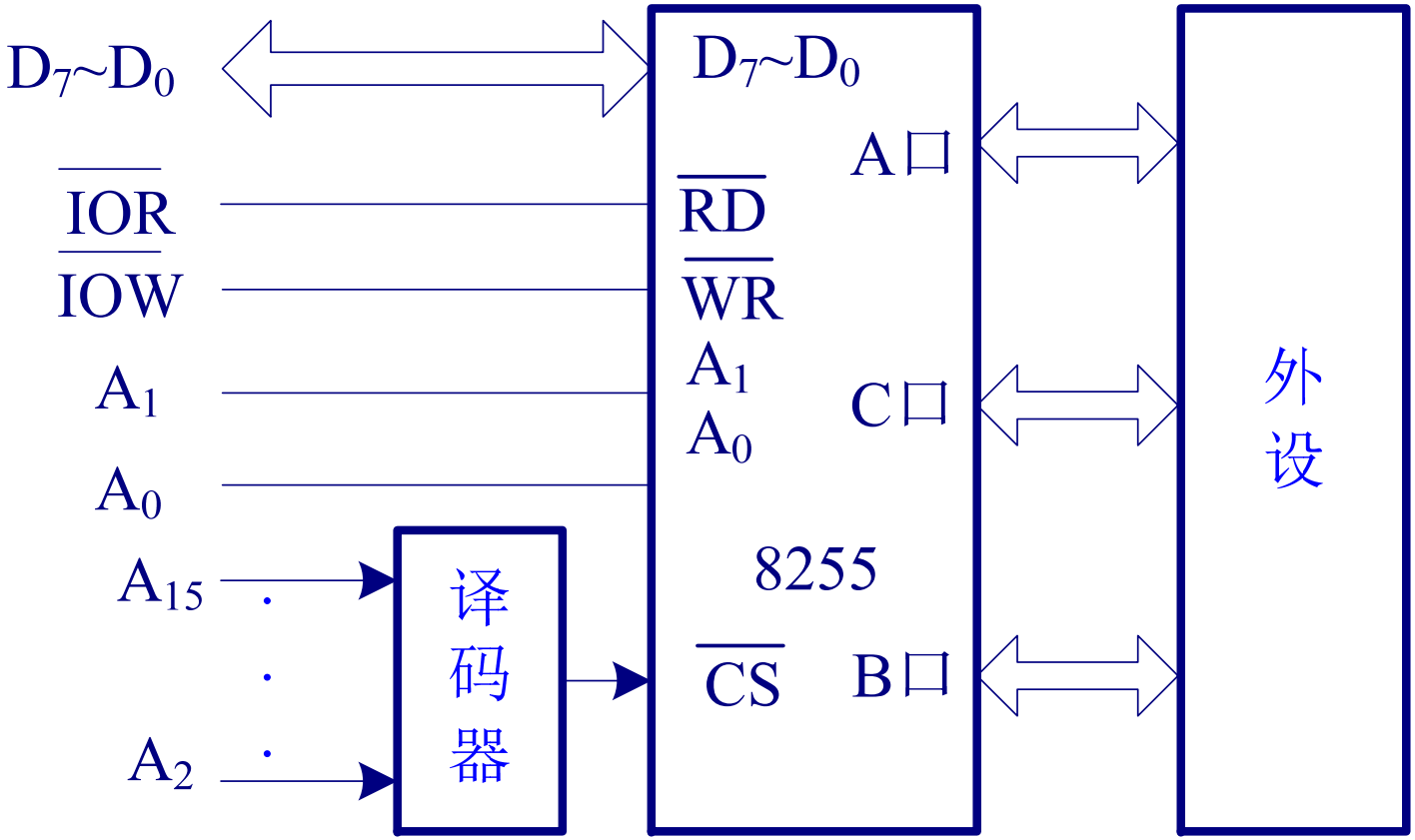
A_1 A_0	定义
0 0	选择A口
0 1	选择B口
1 0	选择C口
1 1	选择控制寄存器



内部结构


内部总线
A组控制
数据总线
缓冲器
读写
逻辑
电路
B组控制
A组A口
(8位)
A组C口
(高4位)
B组C口
(低4位)
B组B口
(8位)
D0~D7
RD
WR
A0
A1
CS
RESET
PA7~PA0
PC7~PC4
PC3~PC0
PB7~PB0

与系统总线的连接示意图



2．8255的工作方式

- 三种基本的工作方式：

  方式0：基本输入/输出方式

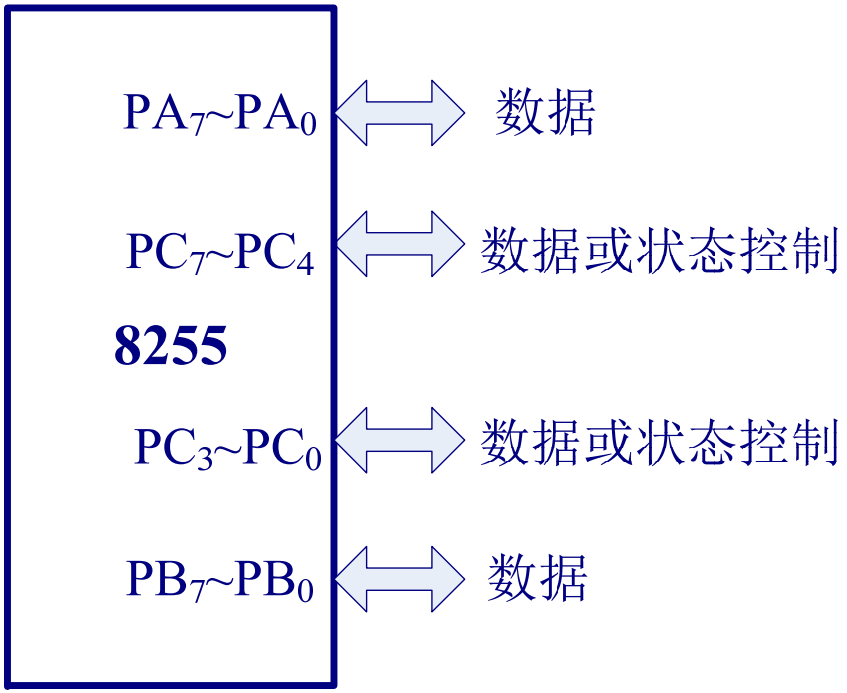
  方式1：选通的输入/输出方式

  方式2：双向传输方式

- A口可以工作在方式0、方式1或方式2，B口和C口只能工作于方式0或方式1。
- 3个端口的工作方式可通过软件编程来设定。

方式0

- A口、C口的高4位，B口、C口的低4位可分别定义为输入或输出，它们互相独立，故共有16种不同的组合。例如，可定义A口和C口高4位为输入口，B口和C口低4位为输出口



- 定义为输出的口均有锁存数据的能力，而定义为输入的口则无锁存能力
- 在方式0下，C口有按位进行置位和复位的能力

方式0的应用方式之一

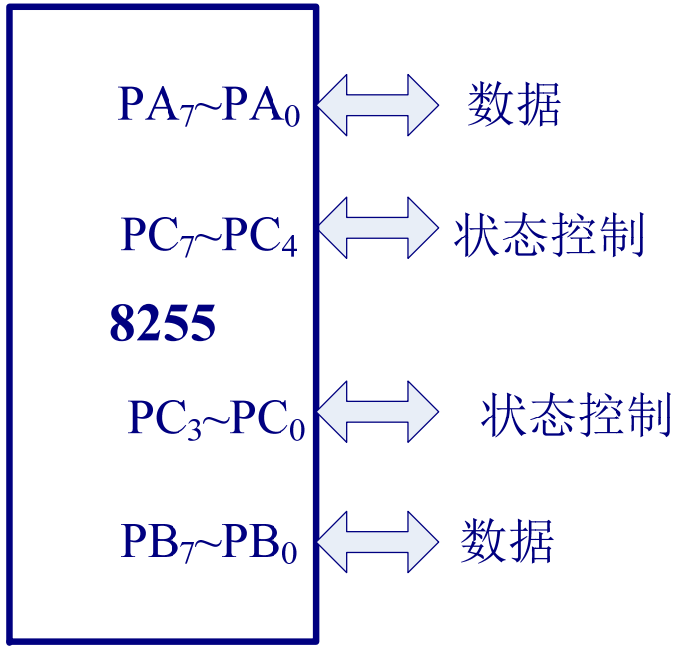
- 无条件传送方式：传送数据的双方互相了解，不需要发控制信号给对方，也不需要查询对方状态，CPU只需直接执行输入/输出指令便可将数据读入或写出。
- 在无条件传送方式下，A、B、C 3个口的全部24位都可以用做数据线

方式0的应用方式之二

- 查询工作方式：需要通信双方互相了解对方的状态，但方式0由于没有规定固定的应答信号，这时常将C口的高4位(或低4位)定义为输入，用来接收外设的状态信号；而将C口的另外4位定义为输出，用来产生控制信号。
- 此时的A、B口可用来传送数据。

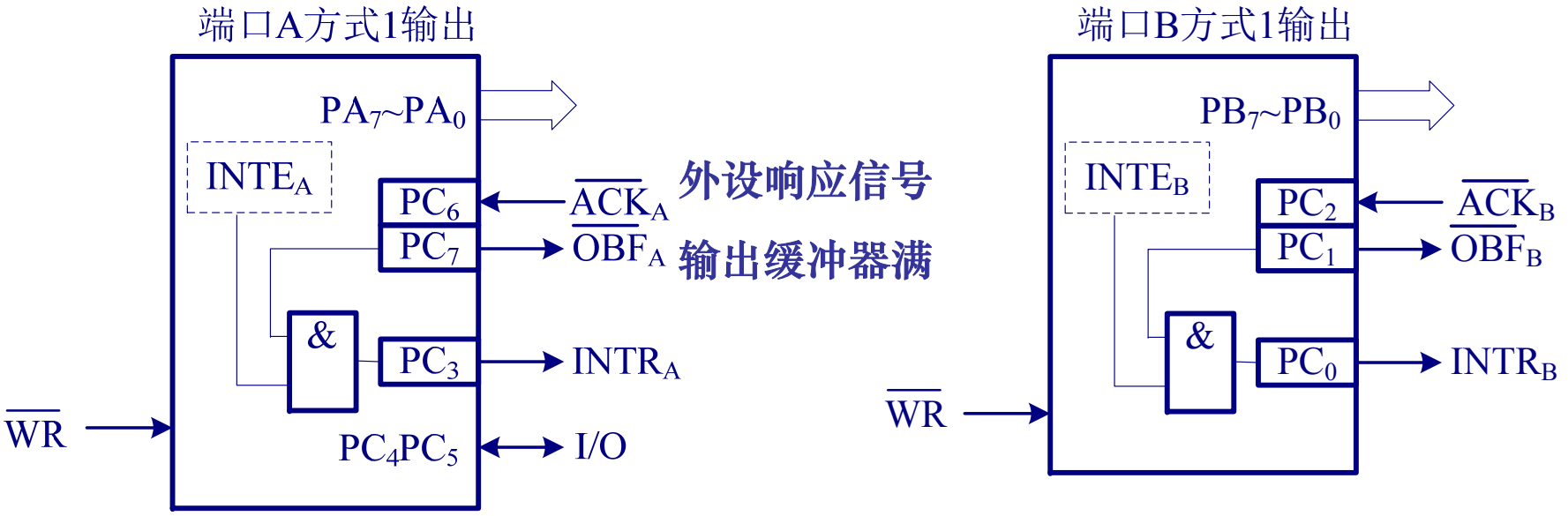
方式1

- 也称为选通的输入/输出方式
- A、B、C 3个口被分为两组
- A组：A口和C口的高4位，A口可由编程任意设定为输入口或输出口，C口的高4位则用作A口输入/输出操作的控制和同步信号



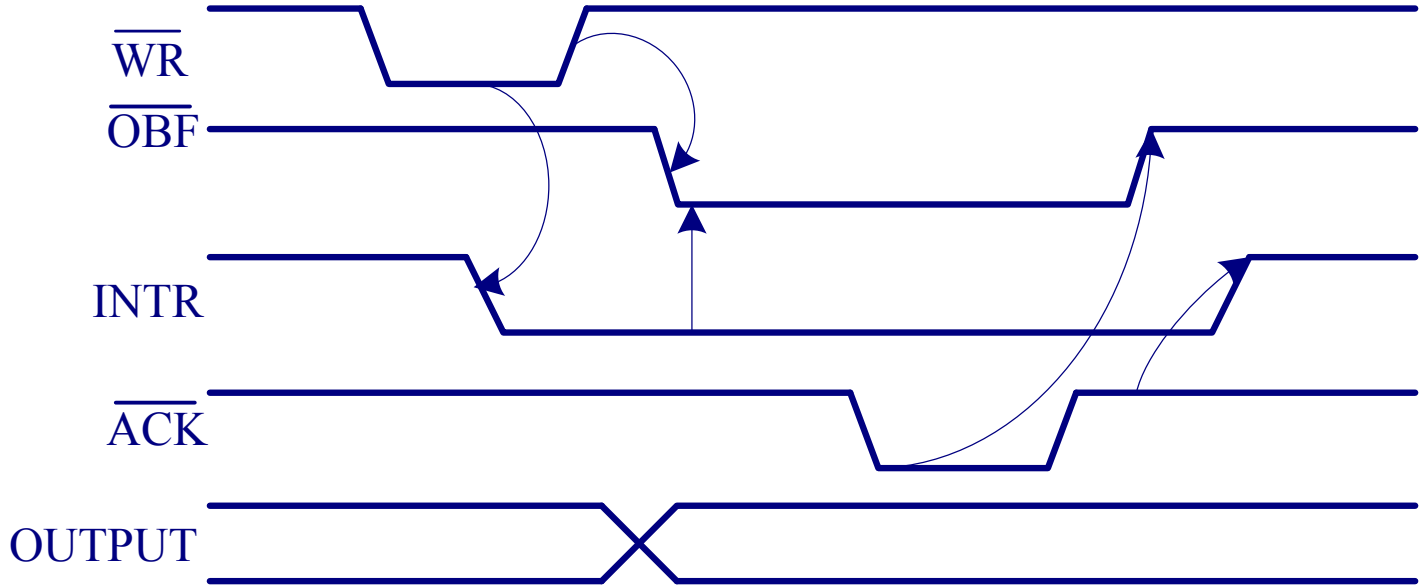
- B组包括B口和C口的低4位，B口可由编程任意设定为输入口或输出口，C口的低4位则用作B口输入/输出操作的控制和同步信号
- A口和B口的输入数据和输出数据都被锁存

方式1（A口、B口都设定为输出口）

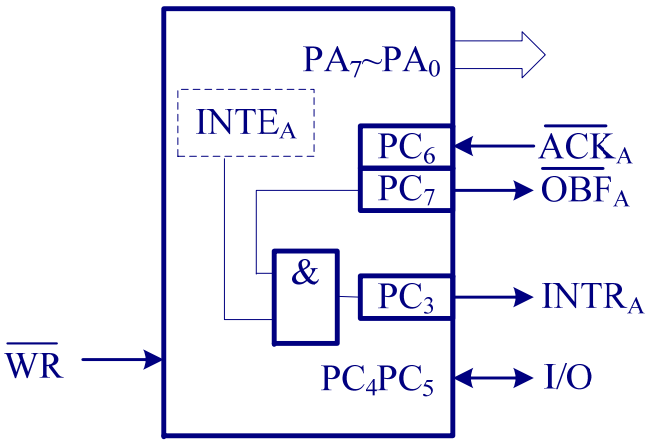


- 利用C口的6根线作为A口和B口的控制信号。控制信号线的安排是固定的，不允许改变：A口使用PC_3、PC_6和PC_7，而B口用PC_0、PC_1和PC_2。

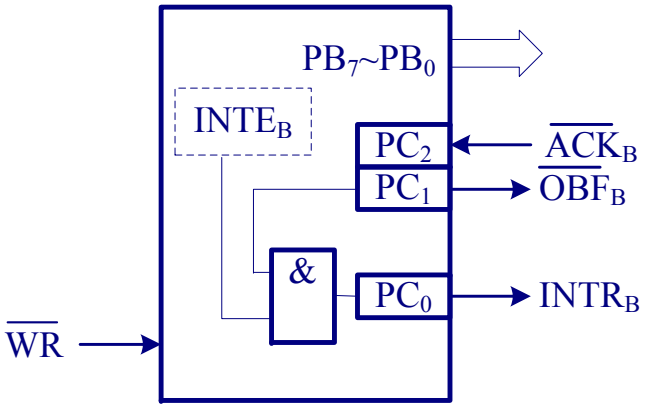
工作方式1下的数据输出时序

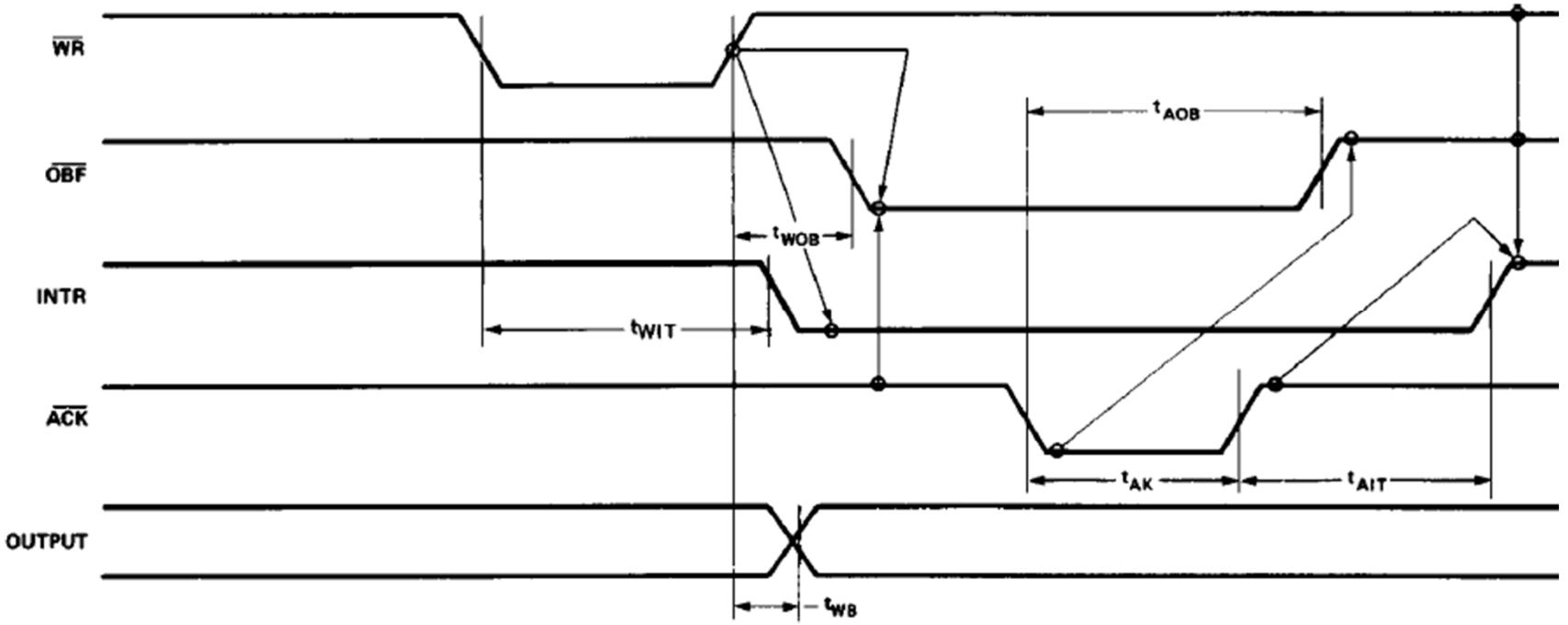


端口A方式1输出

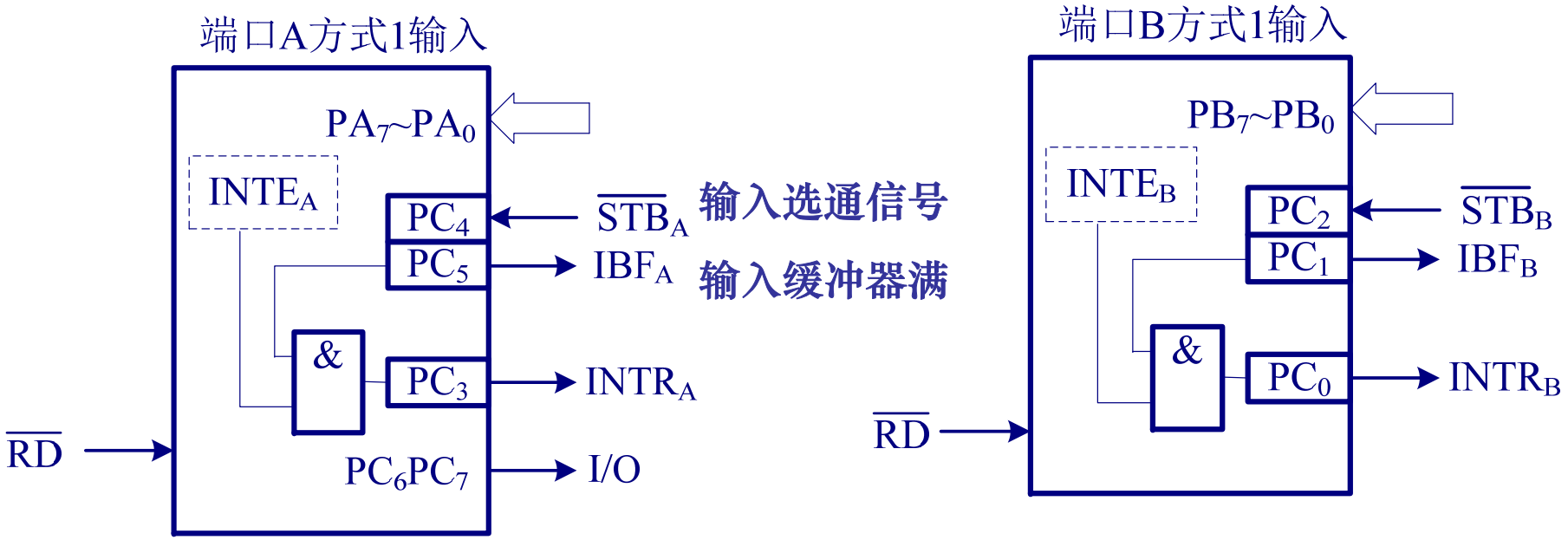


端口B方式1输出



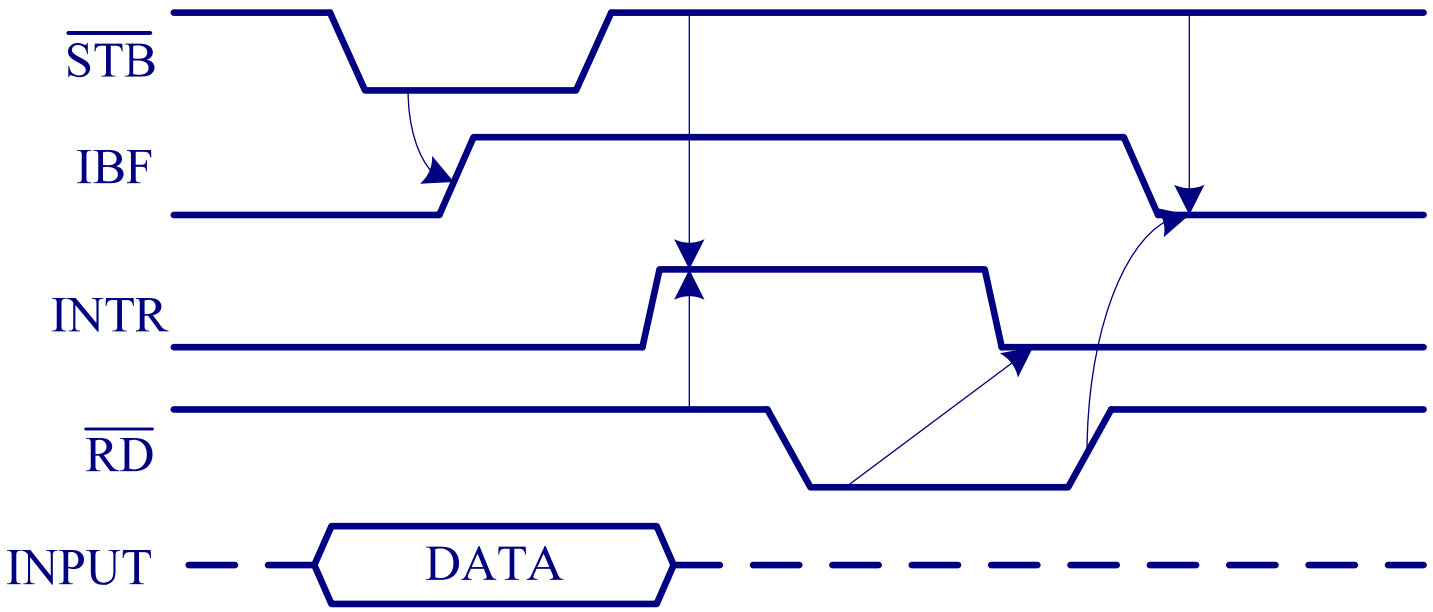


方式1（A口、B口都设定为输入口）

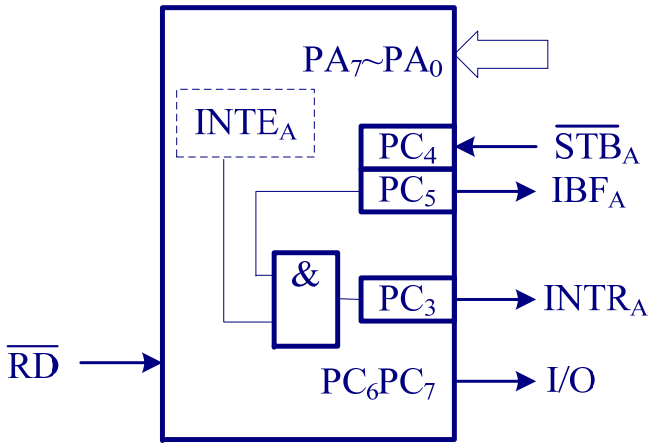


- 利用C口的6根线作为A口和B口的控制信号。控制信号线的安排是固定的，不允许改变：A口使用PC_3、PC_4和PC_5，而B口用PC_0、PC_1和PC_2。

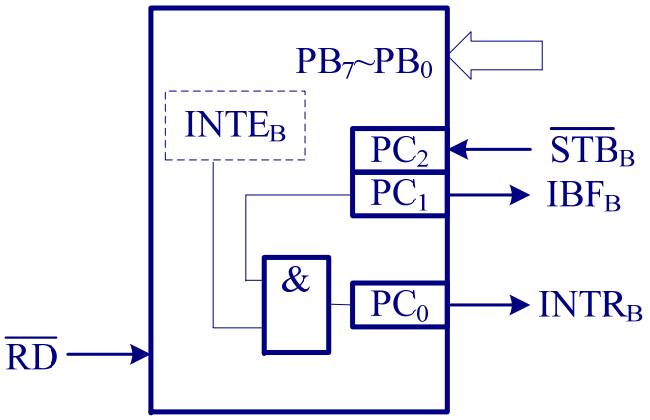
工作方式1下的数据输入时序

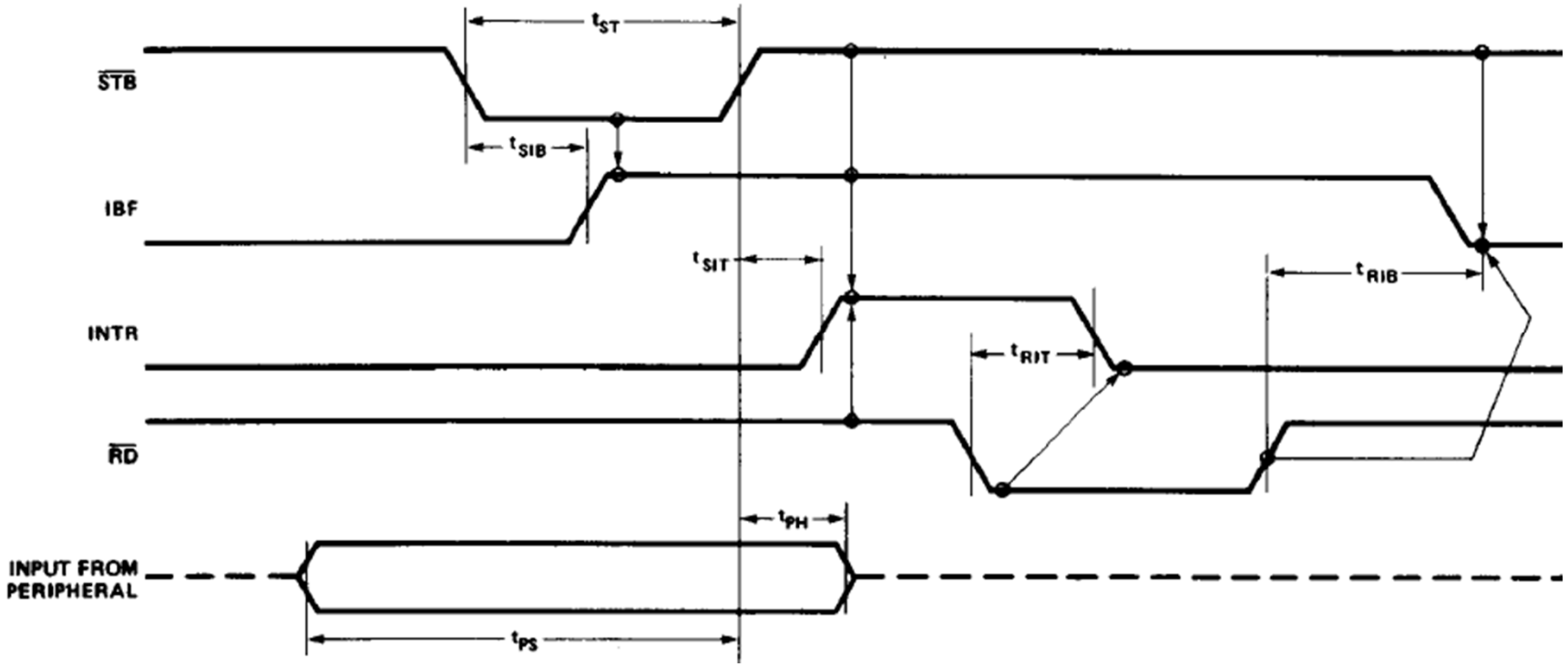


端口A方式1输入



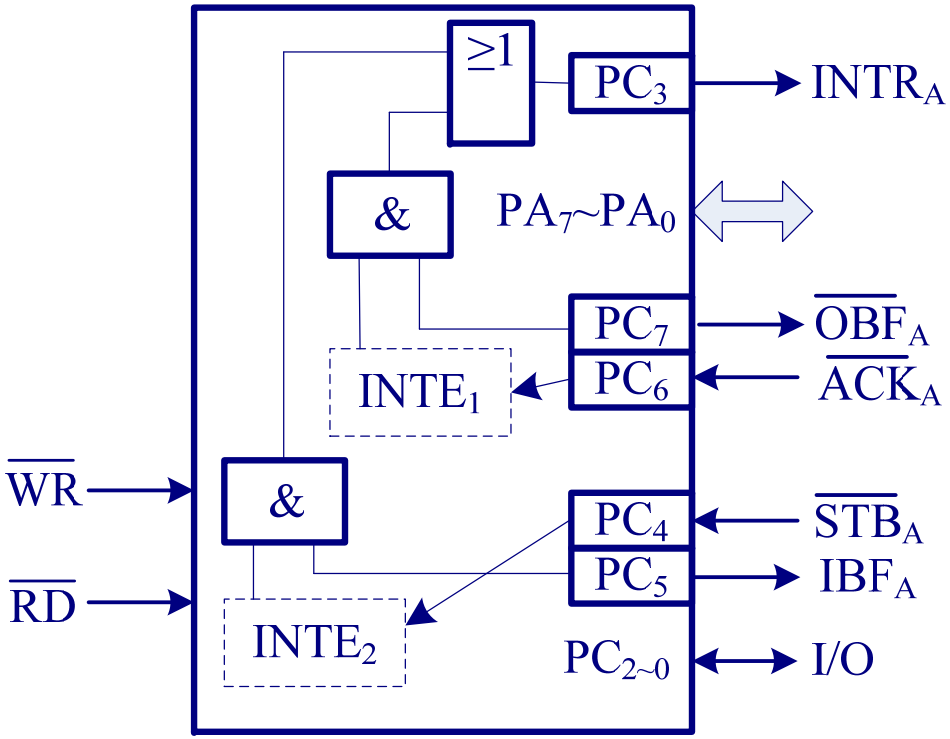
端口B方式1输入





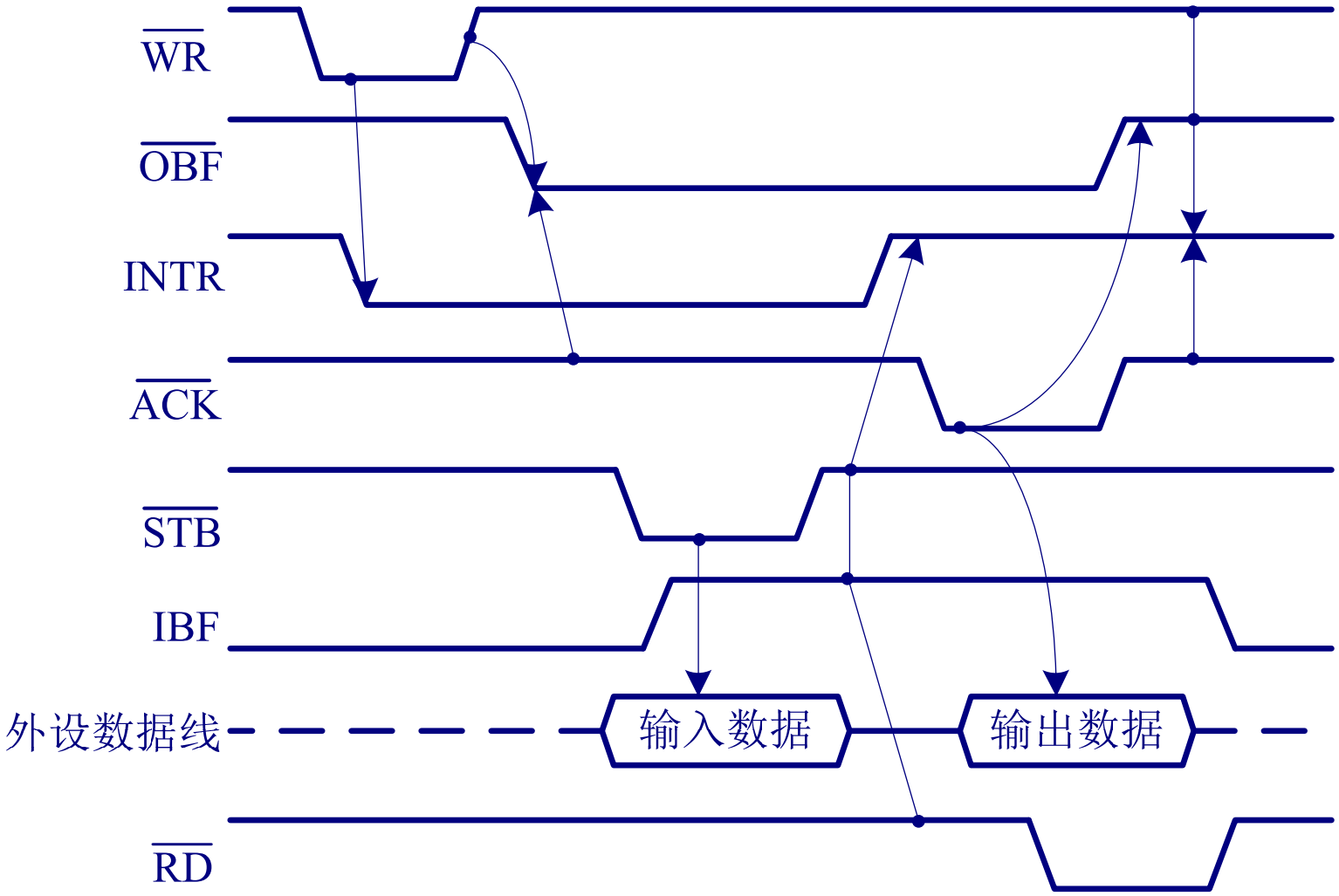
方式2

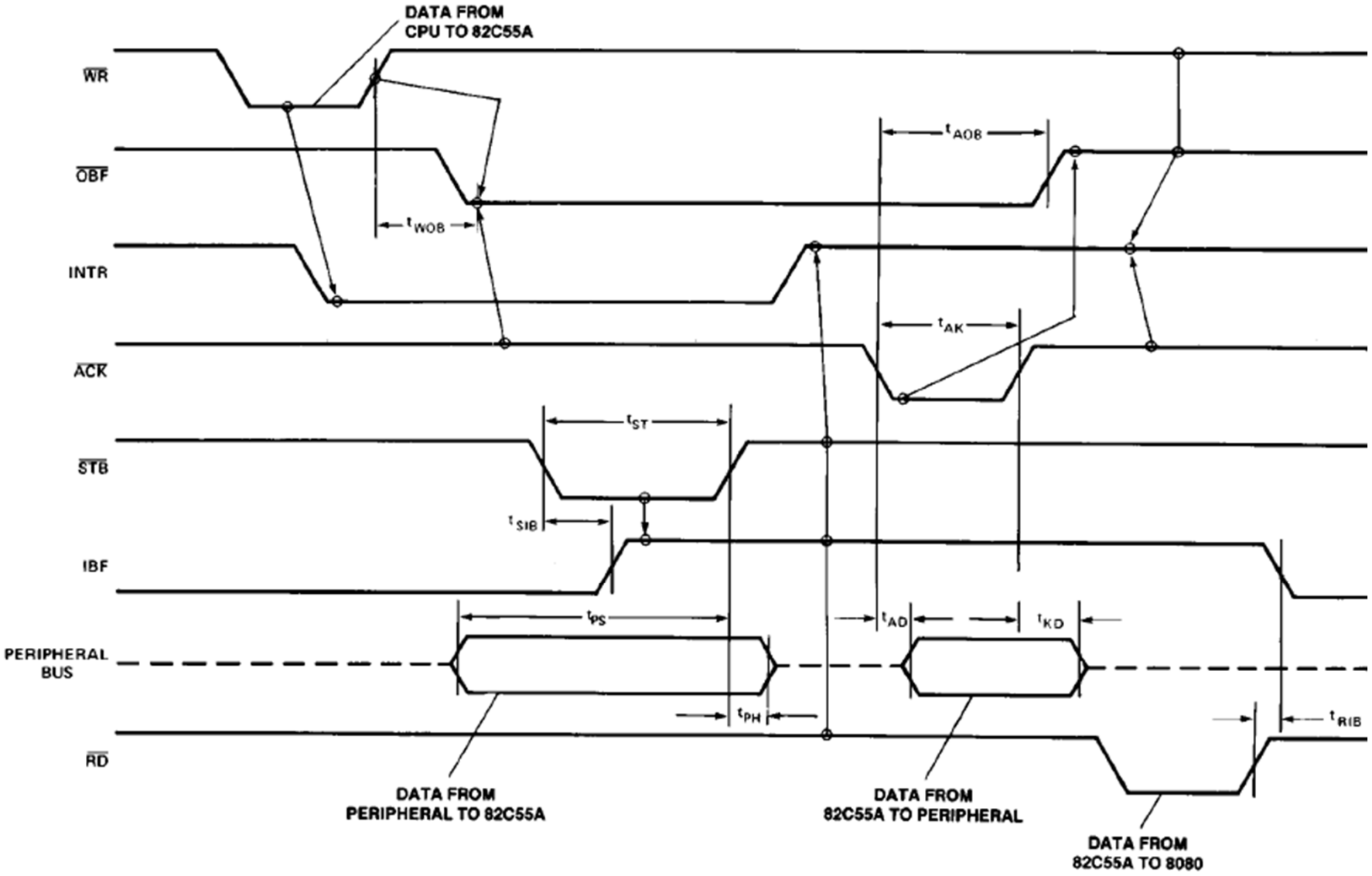
- 双向传输方式。只有A口可以工作在这种方式
- 外设能利用8位数据线与CPU进行双向通信，此时A口既作为输入口又作为输出口



- 利用C口的5根线来提供双向传输所需的控制信号
- 当A口工作于方式2时，B口可以工作在方式0或方式1，而C口剩下的3根线可作为零散的输入/输出线使用或用做B口方式1之下的控制信号线。

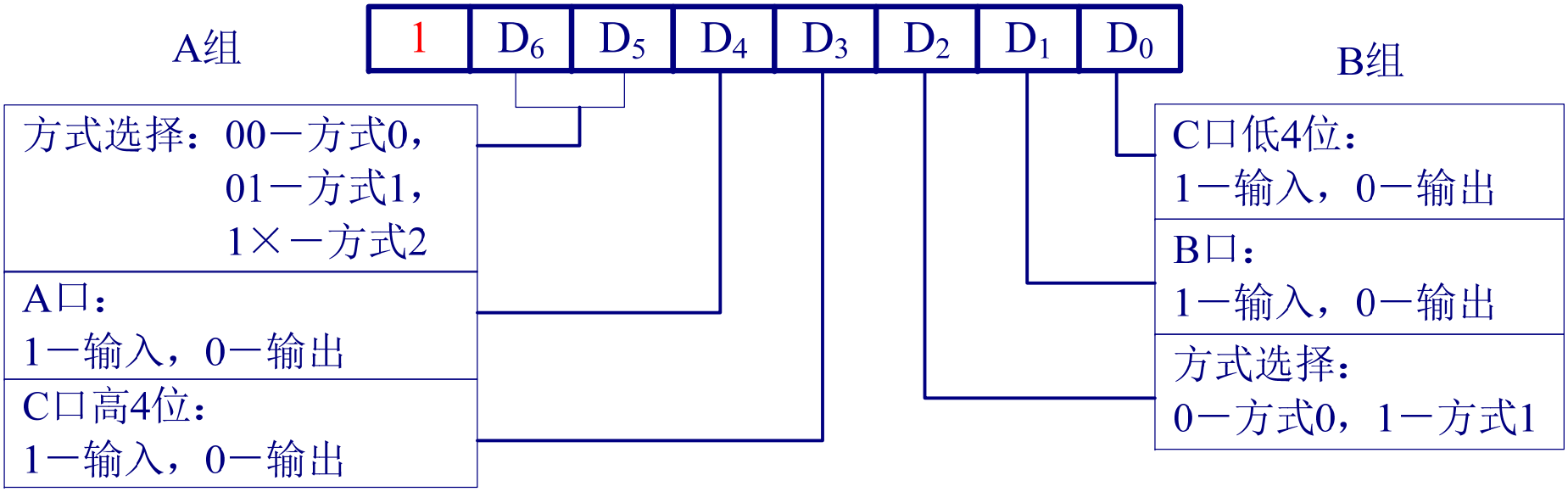
工作方式2下的工作时序



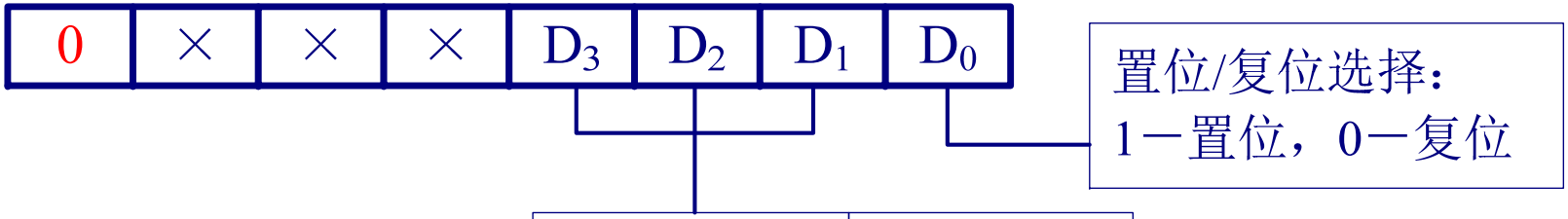

DATA FROM
CPU TO 82C55A
WR
OBF
INTR
ACK
STB
IBF
PERIPHERAL
BUS
RD
t_{WOB}
t_{AOB}
t_{AK}
t_{ST}
t_{SIB}
t_{PS}
t_{AD}
t_{KD}
t_{PH}
t_{RIB}
DATA FROM
PERIPHERAL TO 82C55A
DATA FROM
82C55A TO PERIPHERAL
DATA FROM
82C55A TO 8080

3．8255的控制字及状态字

方式控制字

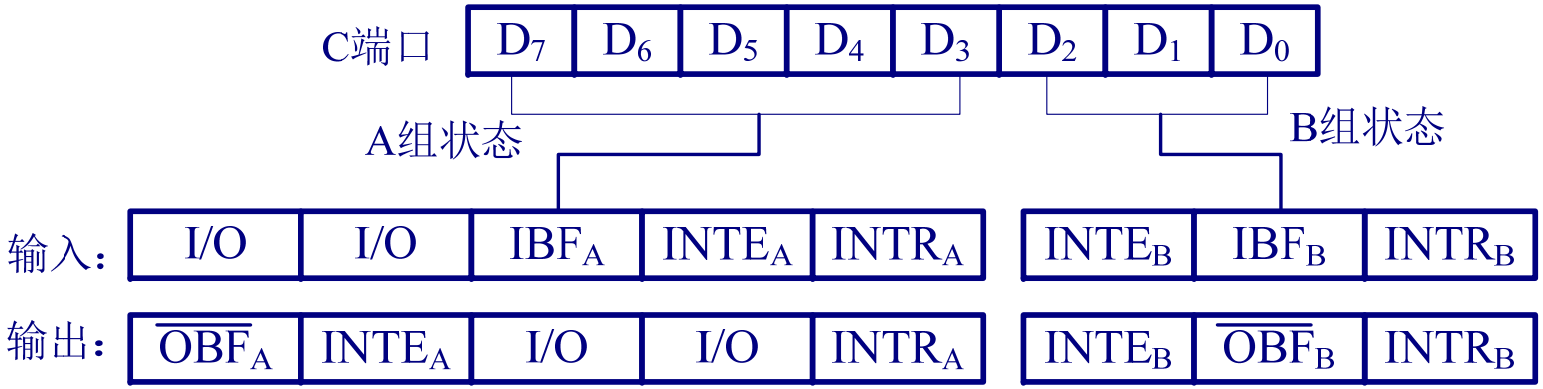


C口的位控制字

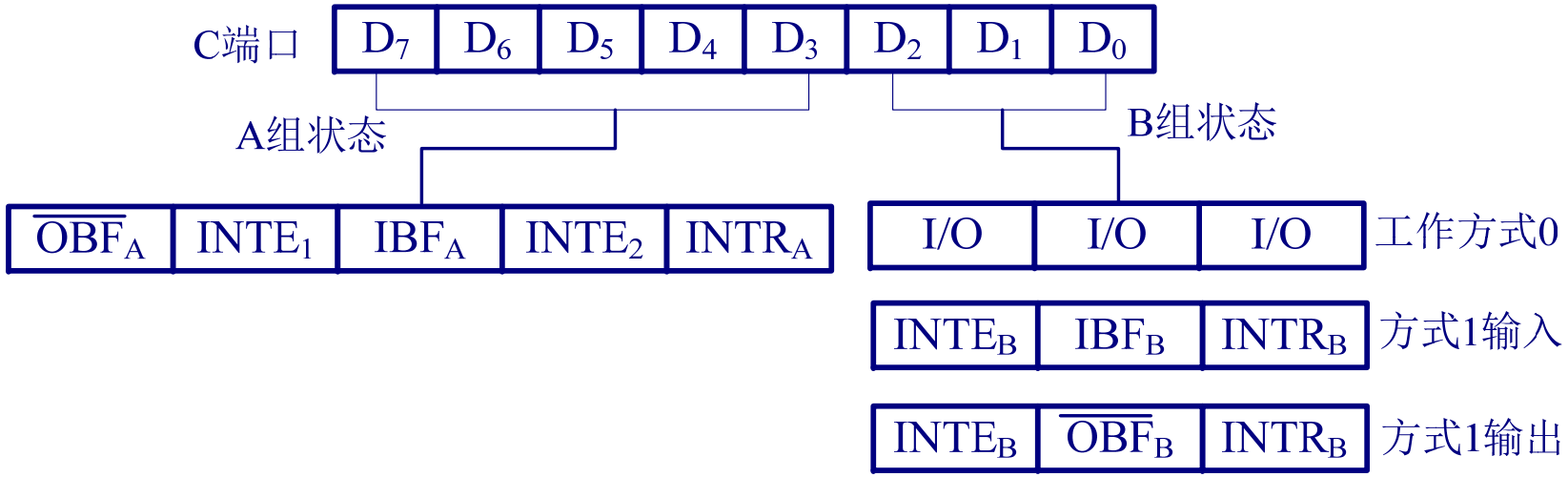


D_3	D_2	D_1	位选择
0	0	0	PC_0
0	0	1	PC_1
0	1	0	PC_2
0	1	1	PC_3
1	0	0	PC_4
1	0	1	PC_5
1	1	0	PC_6
1	1	1	PC_7

方式1时的状态字



方式2时的状态字

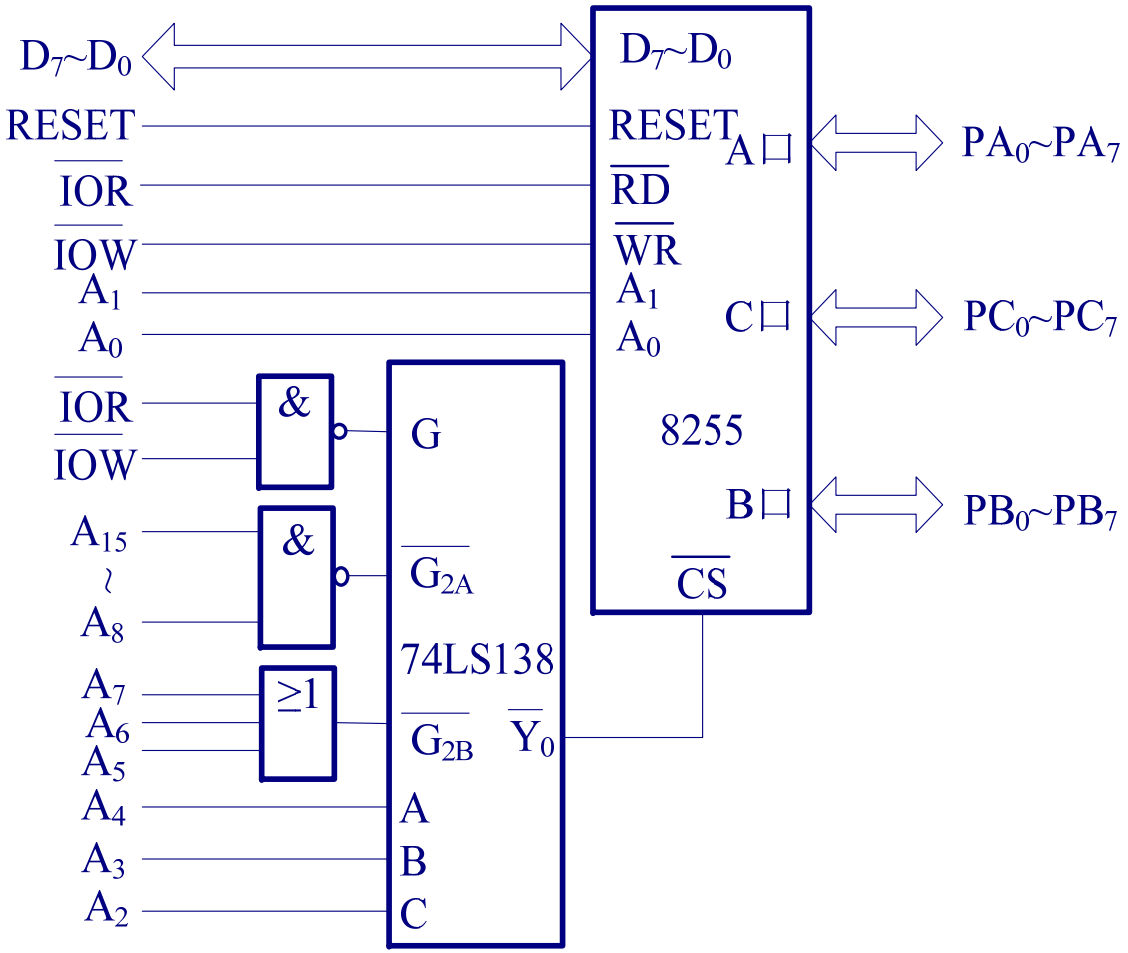


控制信号状态定义

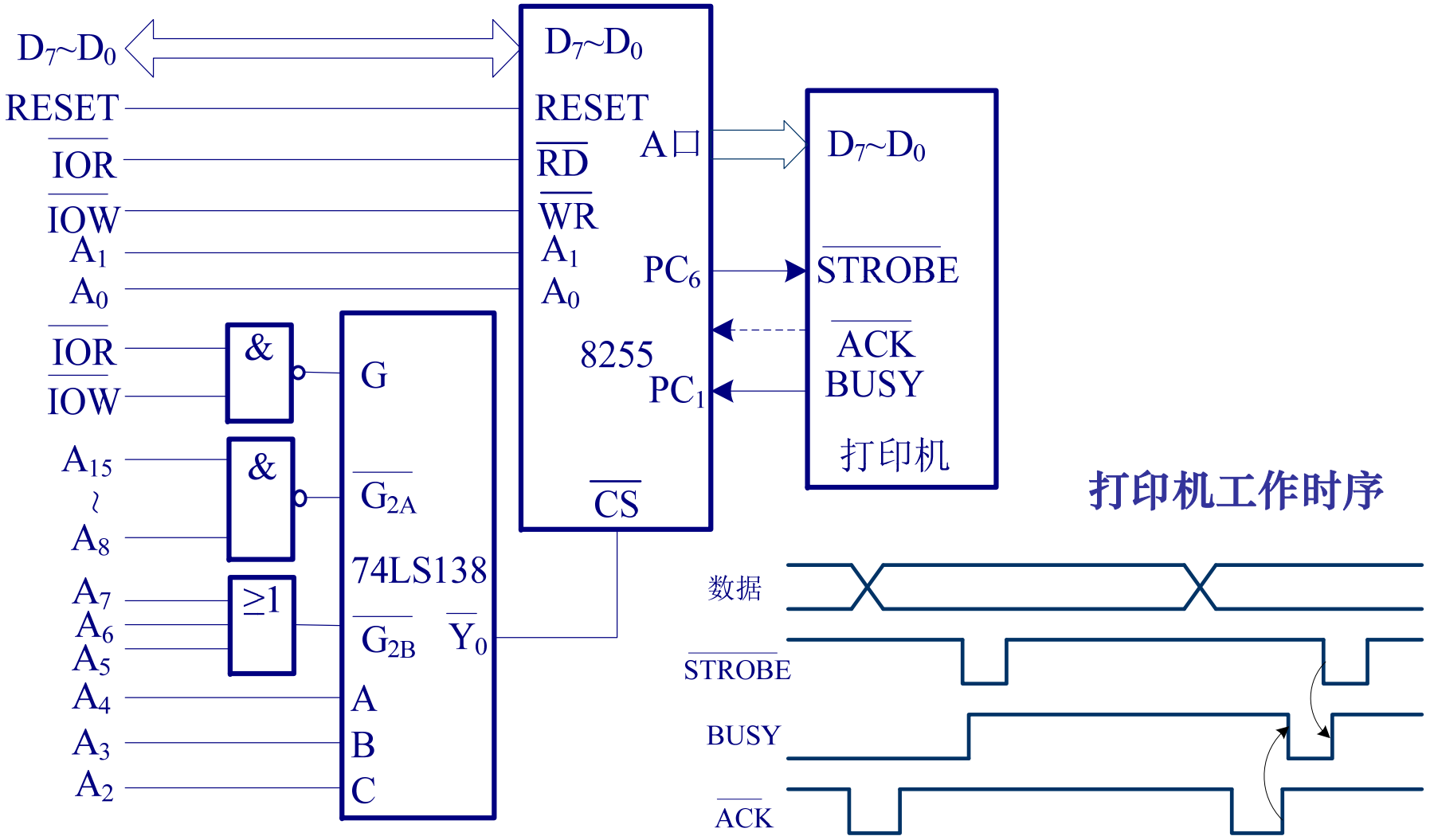
$\overline{CS}$	A_1	A_0	$\overline{RD}$	$\overline{WR}$	功能
0	0	0	0	1	读A口
0	0	1	0	1	读B口
0	1	0	0	1	读C口
0	0	0	1	0	写A口
0	0	1	1	0	写B口
0	1	0	1	0	写C口
0	1	1	1	0	写控制寄存器
1	×	×	1	1	D_0~D_7三态

4. 8255的应用举例

（1）与系统总线的连接（全译码）

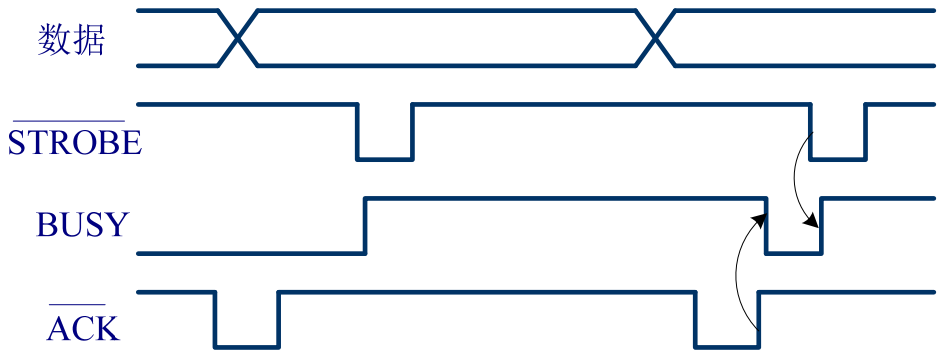


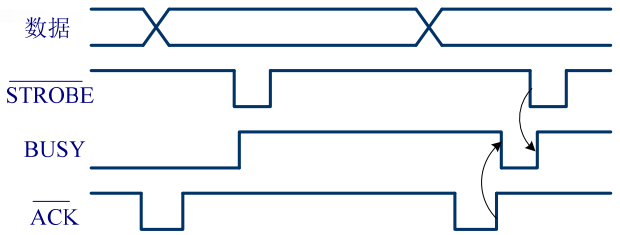
（2）初始化及应用举例：查询方式的打印机接口



8255的初始化程序如下：(8255的地址范围为0FF00H～0FF03H)

```
INIT:  MOV DX, FF03H     ;8255的控制寄存器端口地址送DX
       MOV AL, 10000001B  ;A组方式0：A口、C口高4位输出
                          ;B组方式0：B 口、C 口低4位输入
       OUT DX, AL         ;方式控制字送控制寄存器
       MOV AL, 00001101B  ;C 口的按位操作控制字
                          ;使PC6初始状态置为1
       OUT DX, AL         ;C口位操作控制字送控制寄存器
```





打印一批字符的程序段：

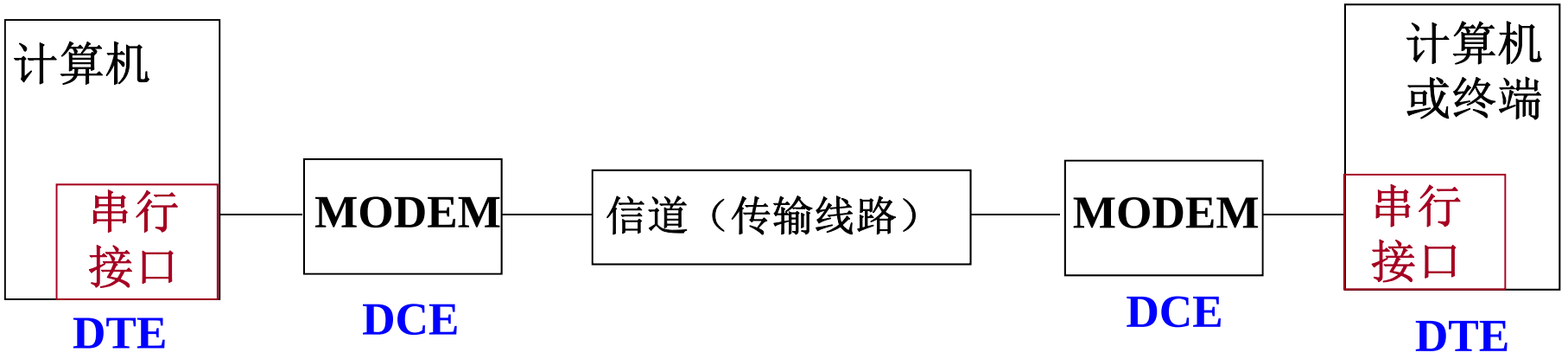
```
        MOV CX, COUNT    ;将字符串长度作为循环次数
        MOV SI, OFFSET DATA   ;取字符串首地址
GOON:  MOV DX，0FF02H   ;0FF02H为C口的地址
        IN AL, DX  ;从C口读入打印机的BUSY信号状态
        AND AL, 02H    ;测试打印机状态(位1)
        JNZ GOON   ;若BUSY为高电子，则循环等待
        MOV AL, [SI]    ;否则取一个字符
        MOV DX, 0FF00H    ;0FF00H为A口的地址
        OUT DX, AL    ;输出一个字符到A口
        MOV DX, 0FF02H    ;准备在PC6上生成一个负脉冲
        MOV AL, 0
        OUT DX, AL   ； 因仅PC6接打印机，故由C口输出00H将使PC6变低
        MOV AL, 40H
        OUT DX, AL    ;再使PC6变高，在PC6上生成一个STROBE负脉冲
        INC SI    ;指向下一个字符
        LOOP  GOON    ;若未结束，则继续
        HLT
```

```
 MOV DX, 0FF03H
 MOV AL, 00001100B
;PC6复位(=0)
 OUT DX, AL
 MOV AL, 00001101B
;PC6置位(=1)
OUT DX, AL
```

7.4 串口通信和可编程接口芯片8251A

0. 串行通信的基本概念
1. 可编程串行通信接口芯片8251A
2. RS-232C串行接口和8251A应用

0. 串行通信的基本概念



串行数据通信系统模型

DTE- Data Terminal Equipment,常常是计算机.

DCE- Data Communication Equipment,常常是MODEM,也可以是计算机.

串行接口 – 主要是**8251A,16550,8250等IC, 连接DTE和DCE.**

基本概念

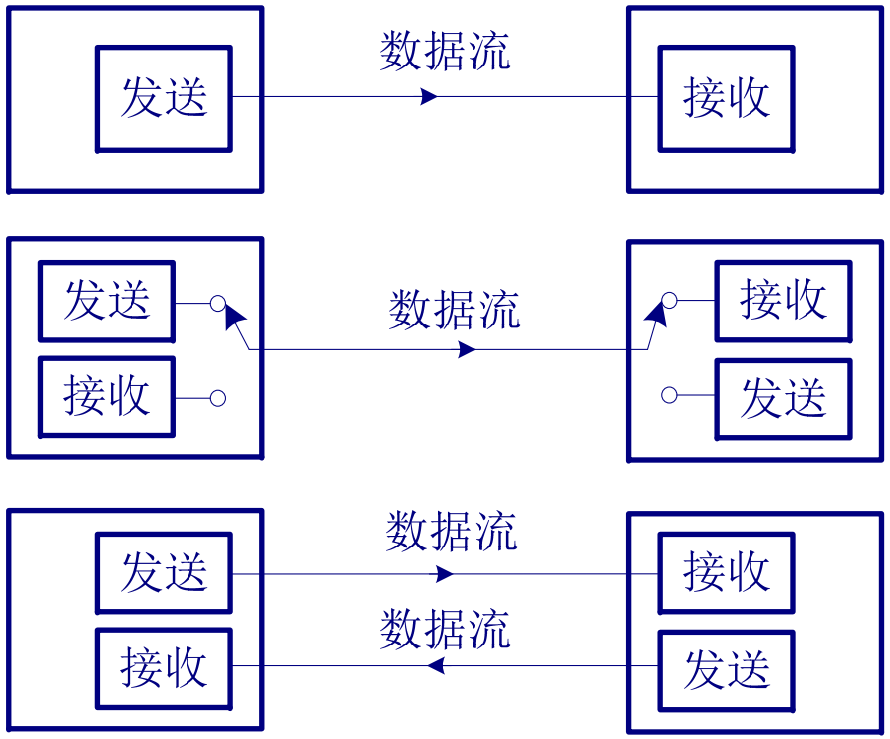
- 数据传送的方向
- 数据的传输速率
- 信号的调制与解调
- 串行通信数据格式
- 串行通信的数据校验

(1) 数据传送的方向

1.单工(Simplex)： 通信是单向的。

2.半双工(Half duplex)： 半双工指A能发信号给B，B也能发信号给A，但这两个过程不能同时进行,A或B发送完后要切换到接受状态。典型的例子就是对讲机。

3.全双工(Duplex)： 在A给B发信号的同时，B也可以给A发信号。典型的例子就是打电话。



(2) 传输速率

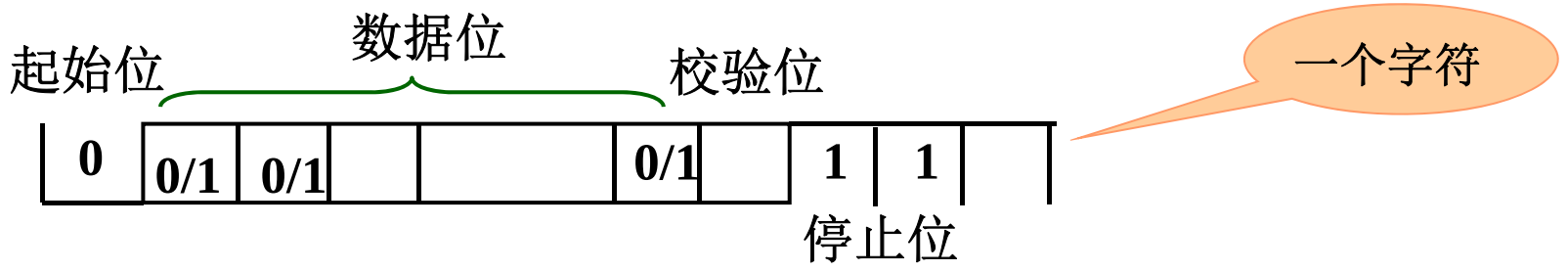
波特率： 单位时间传送的位数，单位**bps(bit/s)**。

波特率因子**K**：每**BIT**占用的时钟周期数。

K=接接收或发送时钟频率/比特率，可取**1，16，32，64**

例1：一个异步串行发送器，发送具有8位数据位的字符，在系统中使用一个奇偶校验位和两个停止位。若每秒发送100个字符，则其波特率为多少？

格式



100*（1+8+1+2）=1200 bps

例2：一个异步串行发送器，发送具有7位数据位的字符, 传送波特率为1800，字符格式为：1个奇偶校验位，1个停止位，问，十秒钟内传送了多少个字符?

10 * 1800/(1+7+1+1) = 1800

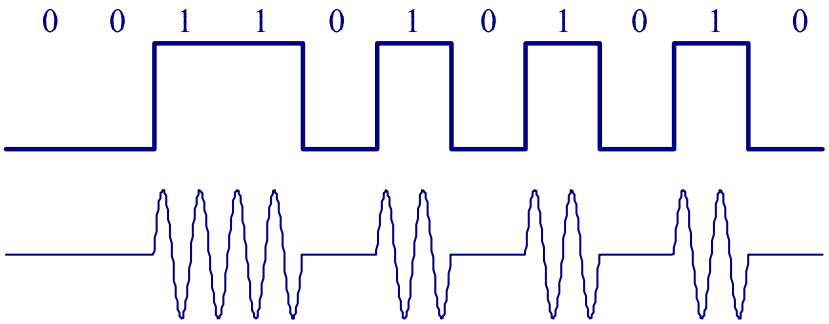
(3) 信号的调制与解调

数据通信传输的是数字信号，要求传送线的频带很宽，若传输带宽很窄，直接传输数字信号，信号就要发生畸变。 因此，需用调制器将数字信号转换成模拟信号，经传输后再用解调器将其转换成数字信号。

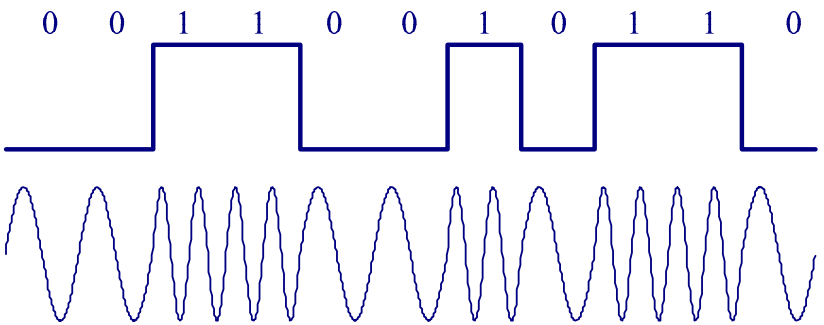
根据载波 $Asin(\omega t + \varphi)$ 的三个参数：幅度、频率、相位，常用的调制技术：

–幅度调制 **Amplitude-Modulating (AM)**

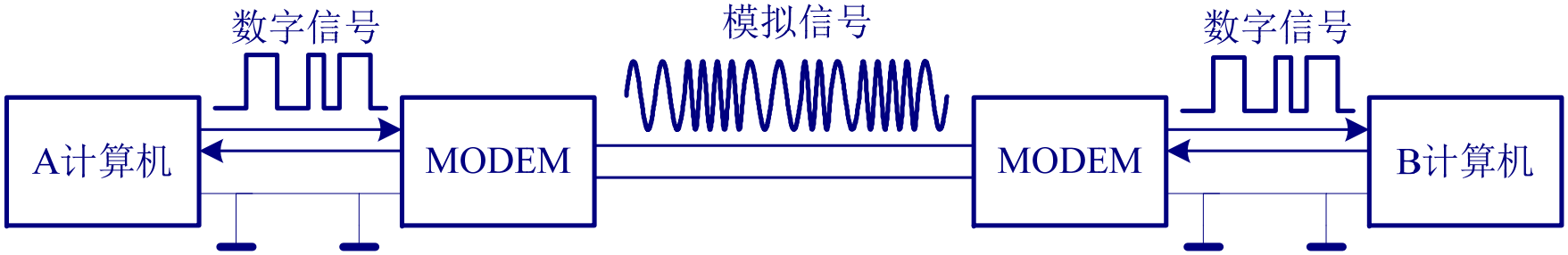
–频移键控法 **Frequency-Shift Keying (FSK)**



用调幅正弦波表示数字1和0



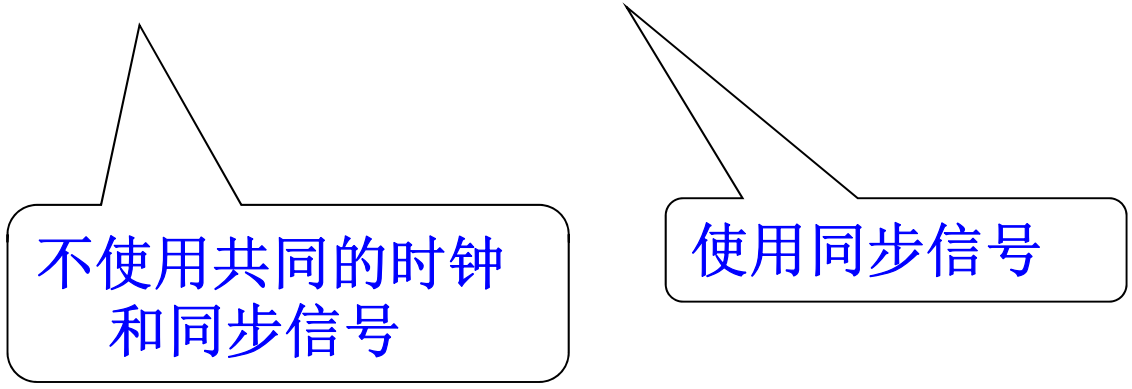
用两种不同频率正弦波表示数字1和0

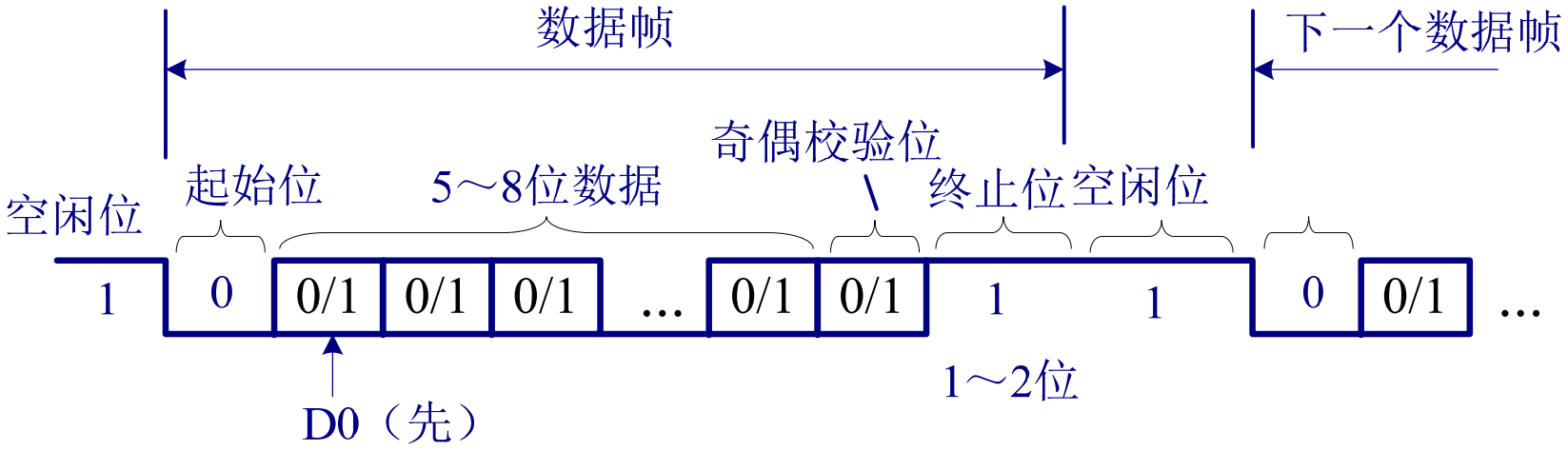


(4) 串行通信数据格式

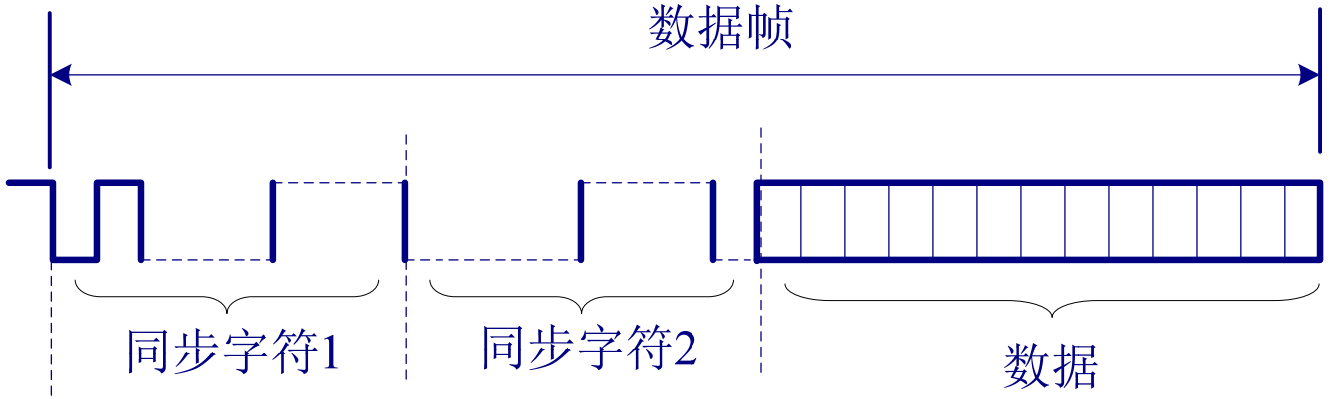
在数据通信中，传输的对象是一系列的 0 和1，这些 0、1 在不同的位置有不同的含义，这些含义都要事先约定好。

在通信中，两种最基本的串行通信方式：异步串行通信、同步串行通信





异步通讯的数据帧格式



同步通讯的数据帧格式

异步串行通信与同步串行通信的对比

异步串行通信	同步串行通信
双方使用各自的时钟	双方使用同一时钟
一帧以字符为单位(一个字符帧的长度取决于帧格式)	以数据块为单位(数据块长度可变)
传输效率低	效率高，速度快
应用于传输速率不高时，简单，应用较广	应用于大批量，高速率数据通信场合

(5) 串行通信的校验方法

串行通信主要适用于远距离通信，因而噪声和干扰较大，为了保证高效而无差错地传送数据，对传送的数据进行校验就成了串行通信中必不可少的重要环节。

常用的校验方法有：

奇偶校验

循环冗余校验(CRC)

奇偶校验

- 这种校验方法主要用于对一个字符的传送过程进行校验。

- 奇偶校验可以检查出一个字节中发生的单个错误。

- 奇偶校验不能自动纠错，发现错误后需“重传”。

2. 可编程USART通信接口芯片8251A

(1) 基本功能：

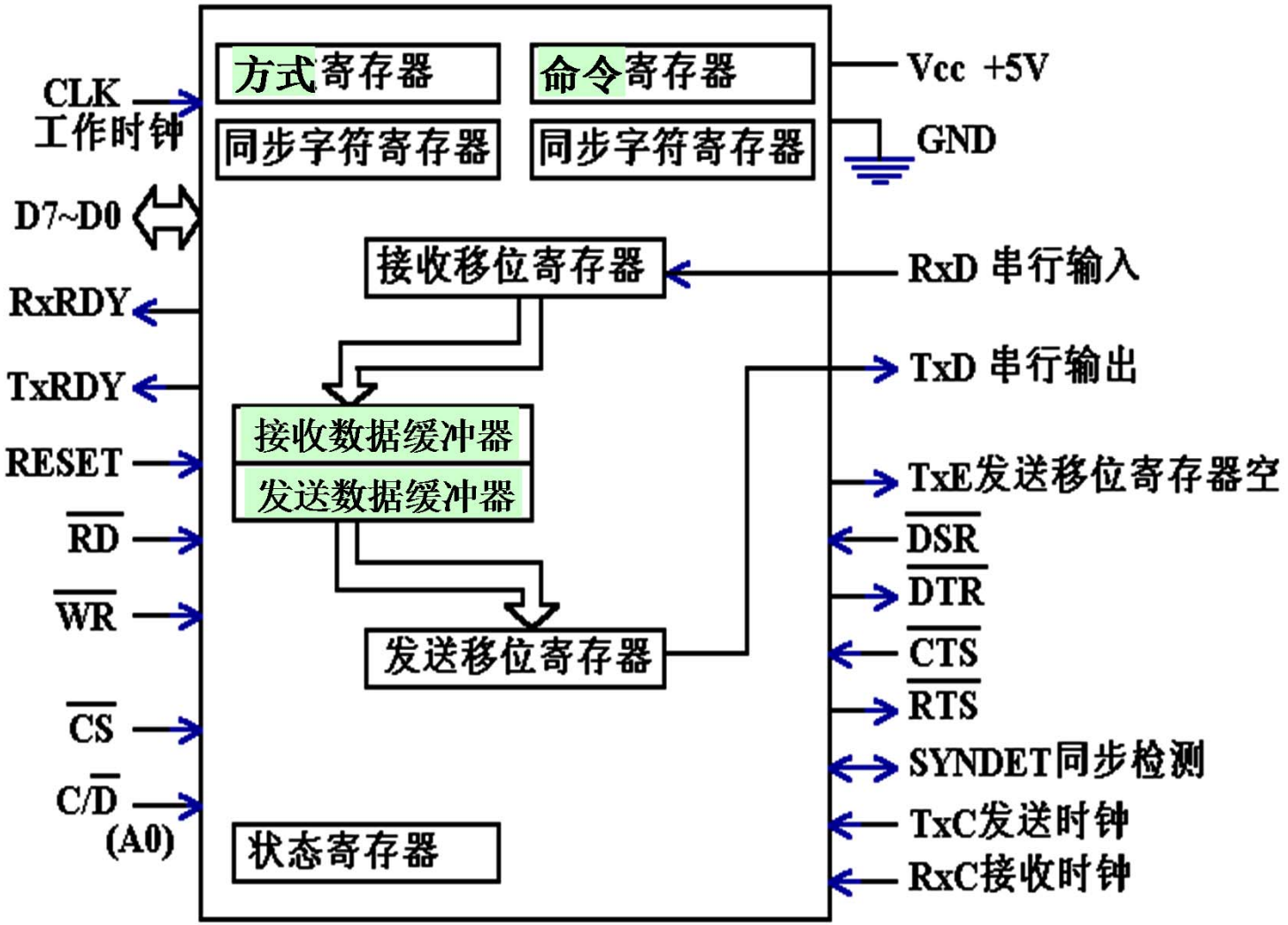
1) 可用于同步和异步通信方式(通信方式通过对方式字编程规定)

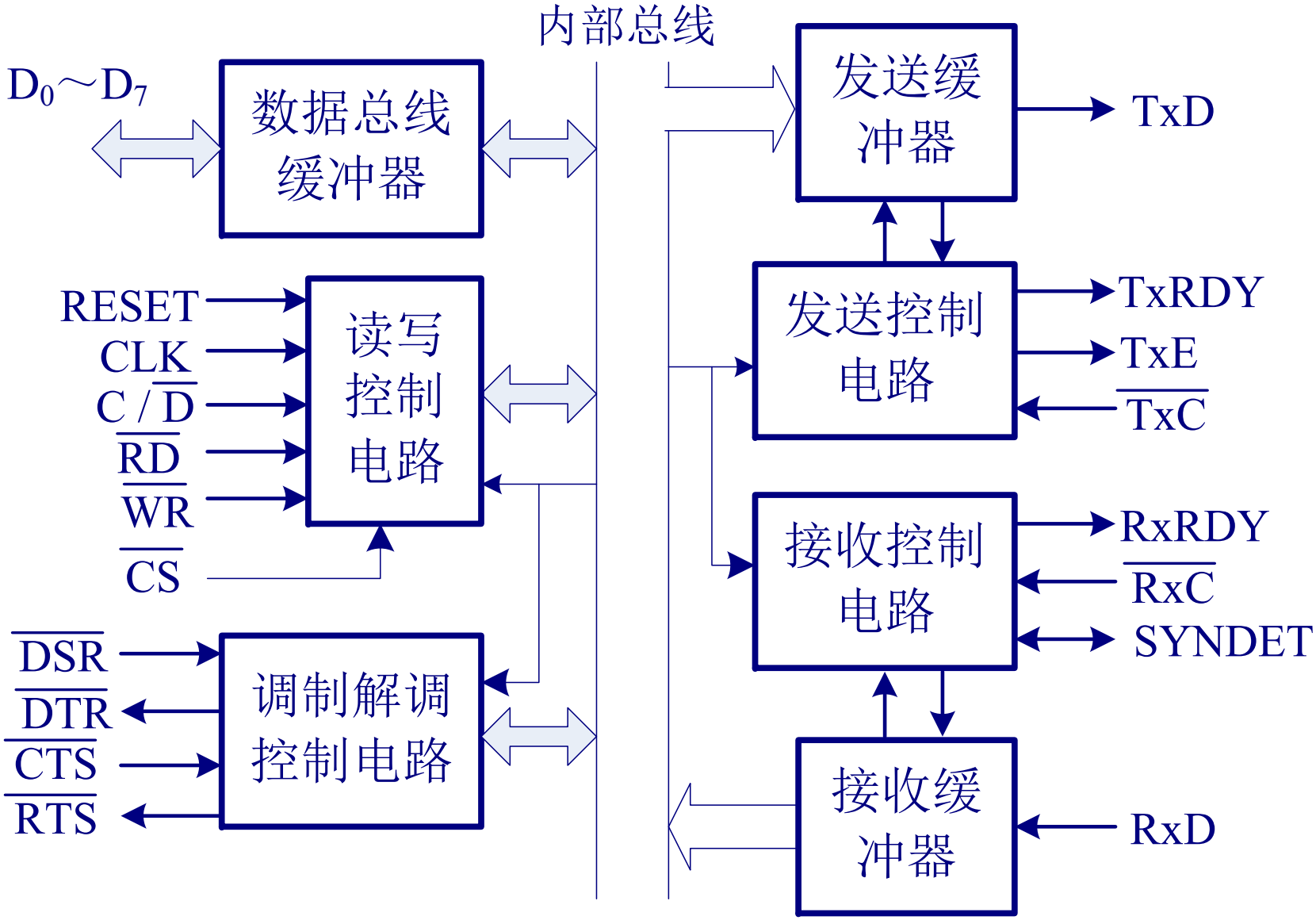
同步方式： 波特率**0-64Kbps**，每字符为**5**，**6**，**7**，**8**位，可使用内部同步检测和外部同步检测，能自动插入同步字符。

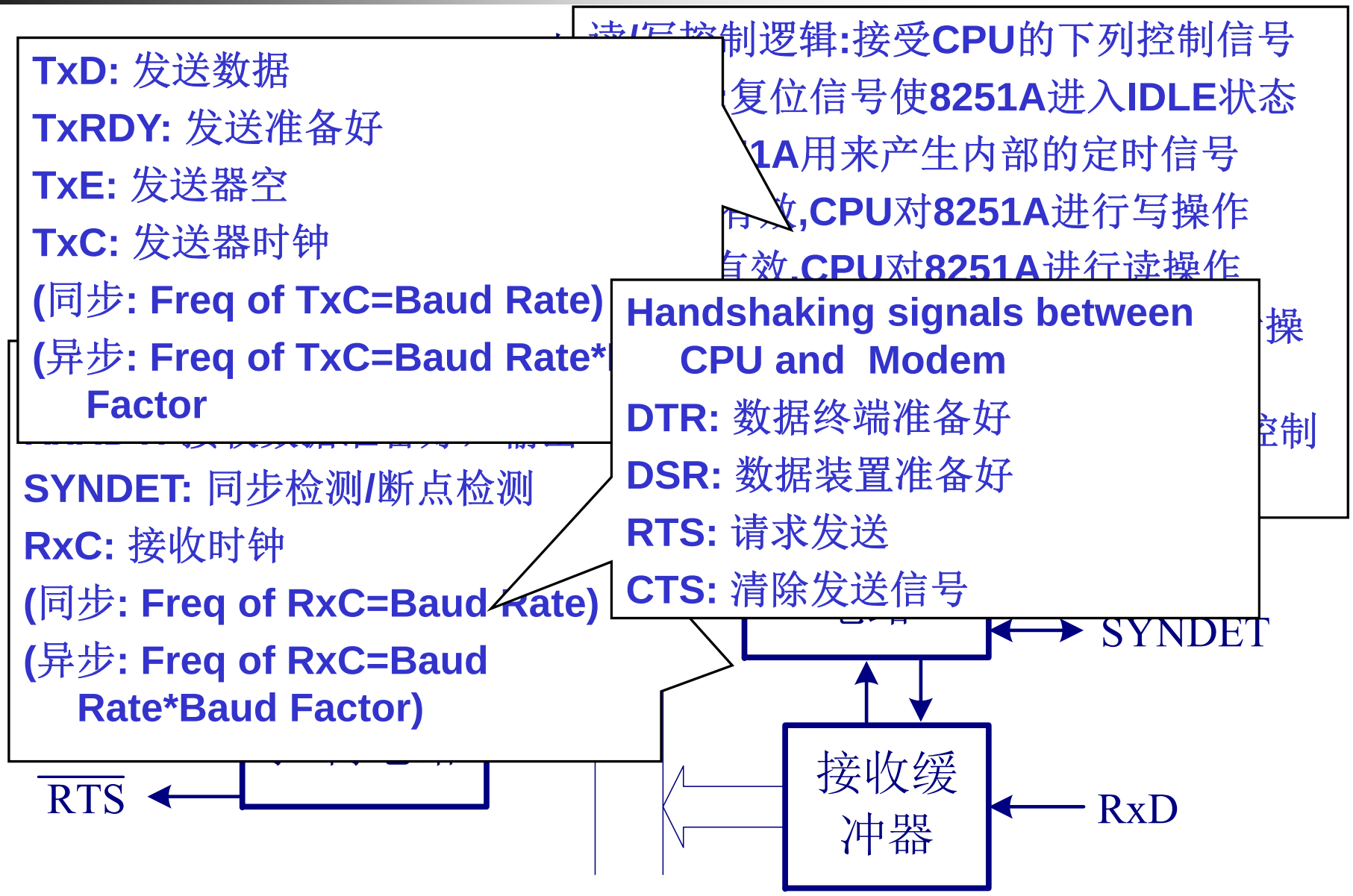
异步方式：波特率 **0-19.2Kbps**，每字符可为**5**，**6**，**7**，**8**位，自动增加起始位、停止位和校验位。时钟**TxC, RxC**速率为波特率的**1**，**16**和**64**倍。

2) 完全双工，双缓冲器接收器和发送器；

3) 出错检测：具有奇偶、溢出和帧错等检测电路。

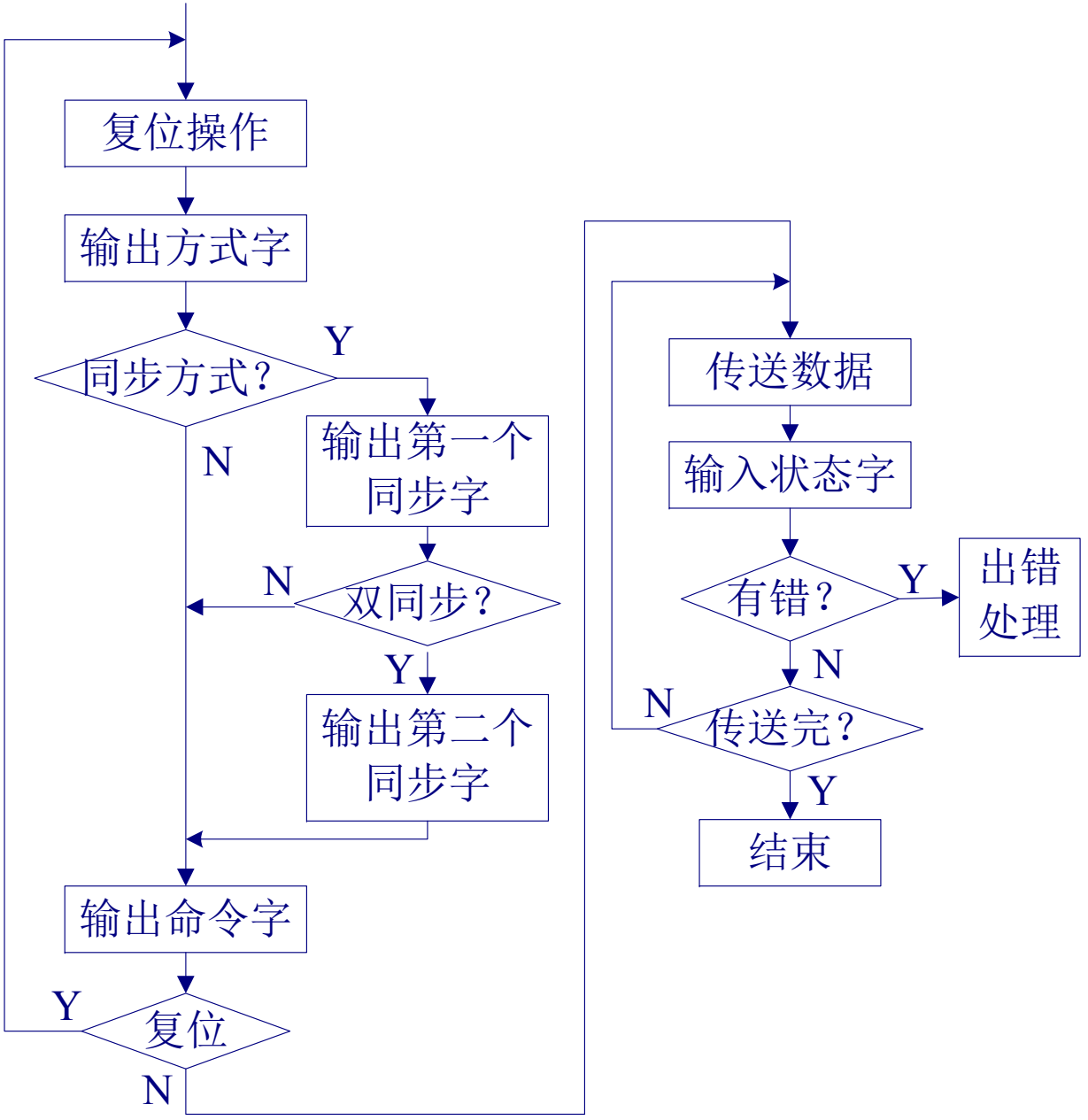

方式寄存器
命令寄存器
同步字符寄存器
同步字符寄存器
接收移位寄存器
接收数据缓冲器
发送数据缓冲器
发送移位寄存器
状态寄存器
CLK
工作时钟
D7~D0
RxRDY
TxRDY
RESET
RD
WR
CS
C/D
(A0)
Vcc +5V
GND
RxD 串行输入
TxD 串行输出
TxE发送移位寄存器空
DSR
DTR
CTS
RTS
SYNDET同步检测
TxC发送时钟
RxC接收时钟


内部总线
D0~D7
数据总线缓冲器
RESET
CLK
C/D
RD
WR
CS
读写控制电路
DSR
DTR
CTS
RTS
调制解调控制电路
发送缓冲器
TxD
发送控制电路
TxRDY
TxE
TxC
接收控制电路
RxRDY
RxC
SYNDET
接收缓冲器
RxD


读/写控制逻辑:接受CPU的下列控制信号
复位信号使8251A进入IDLE状态
1A用来产生内部的定时信号
效,CPU对8251A进行写操作
有效,CPU对8251A进行读操作
TxD: 发送数据
TxRDY: 发送准备好
TxE: 发送器空
TxC: 发送器时钟
(同步: Freq of TxC=Baud Rate)
(异步: Freq of TxC=Baud Rate*
Factor
Handshaking signals between
CPU and Modem
DTR: 数据终端准备好
DSR: 数据装置准备好
RTS: 请求发送
CTS: 清除发送信号
SYNDET: 同步检测/断点检测
RxC: 接收时钟
(同步: Freq of RxC=Baud Rate)
(异步: Freq of RxC=Baud
Rate*Baud Factor)
SYNDET
RTS
接收缓
冲器
RxD

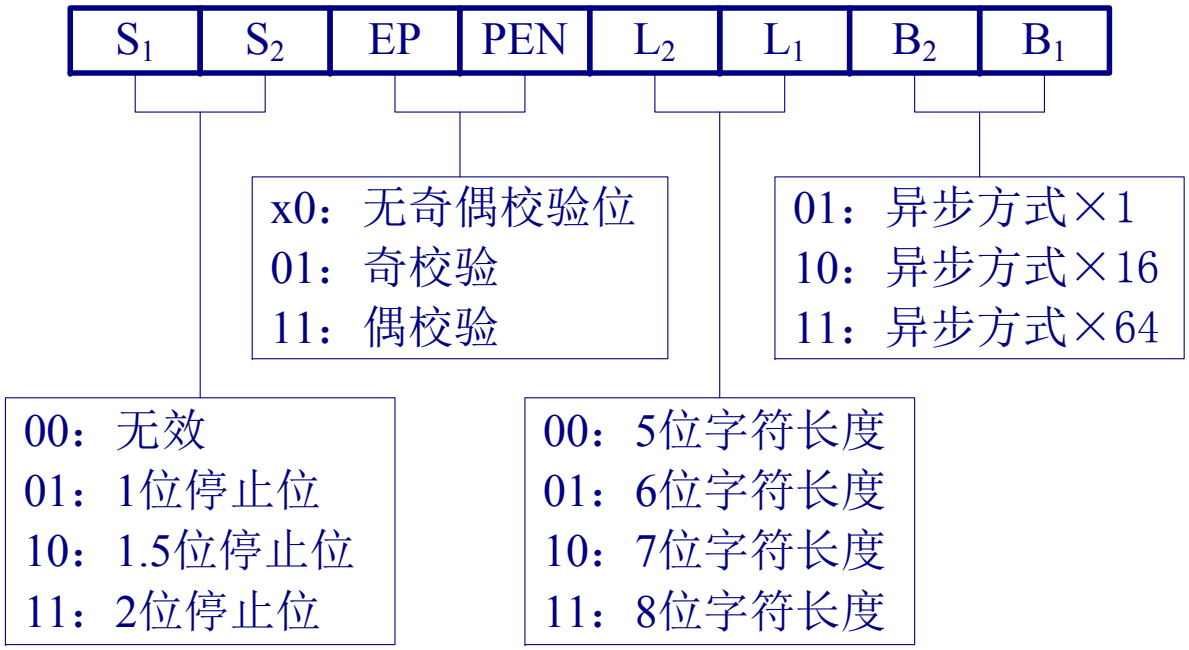
(2) 8251A编程

编程流程

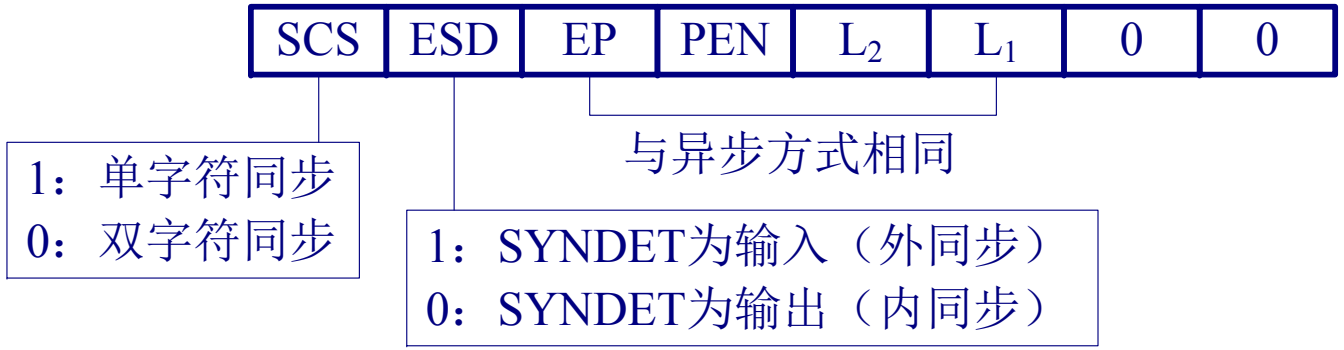


方式字

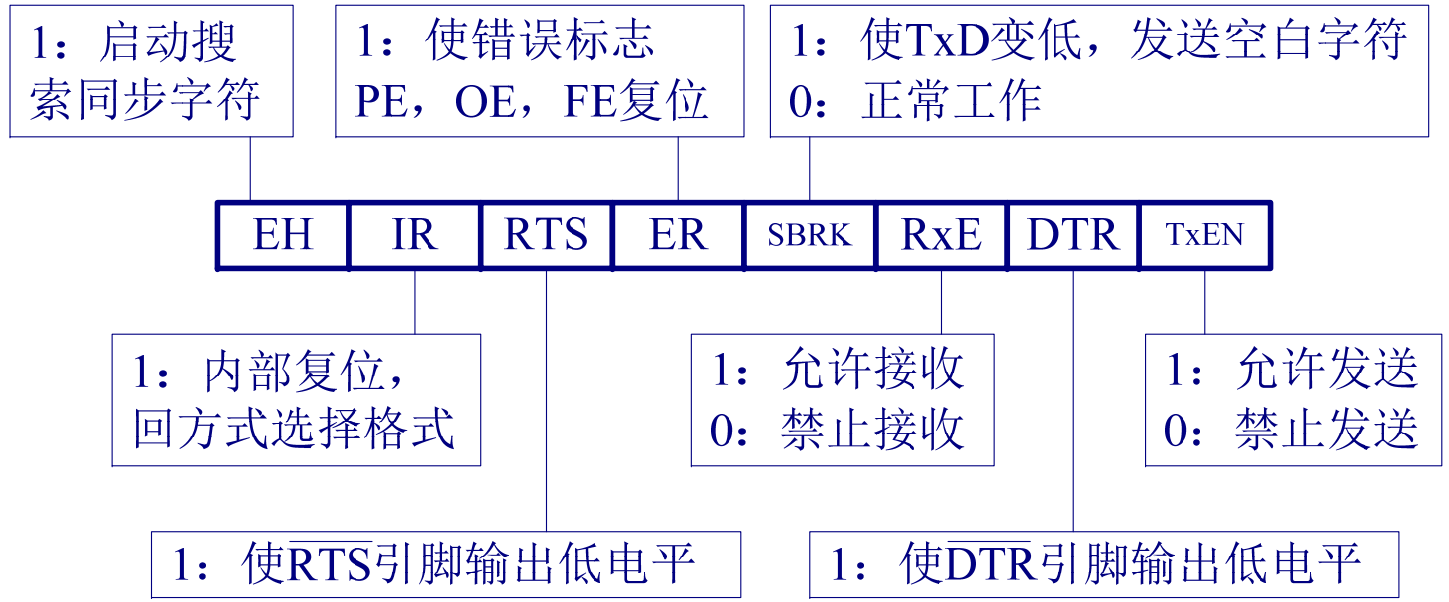
异步方式



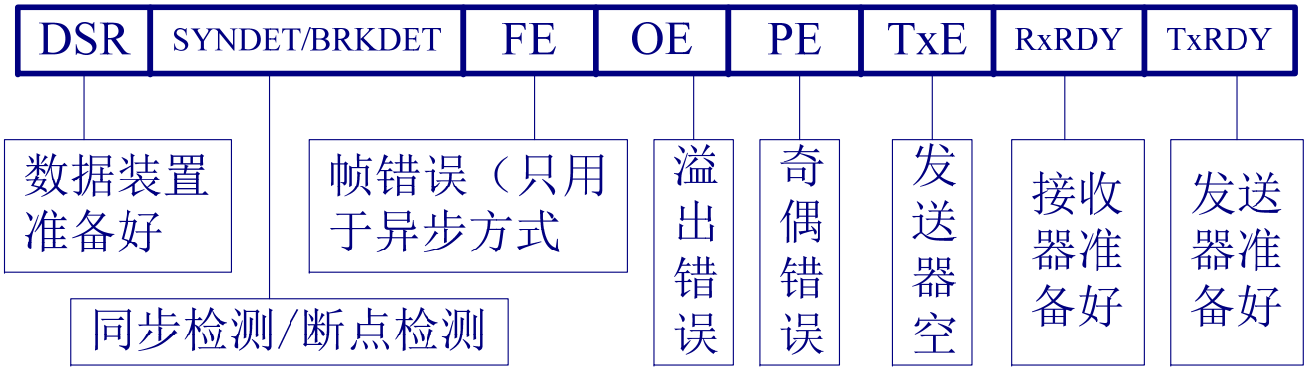
同步方式



命令字

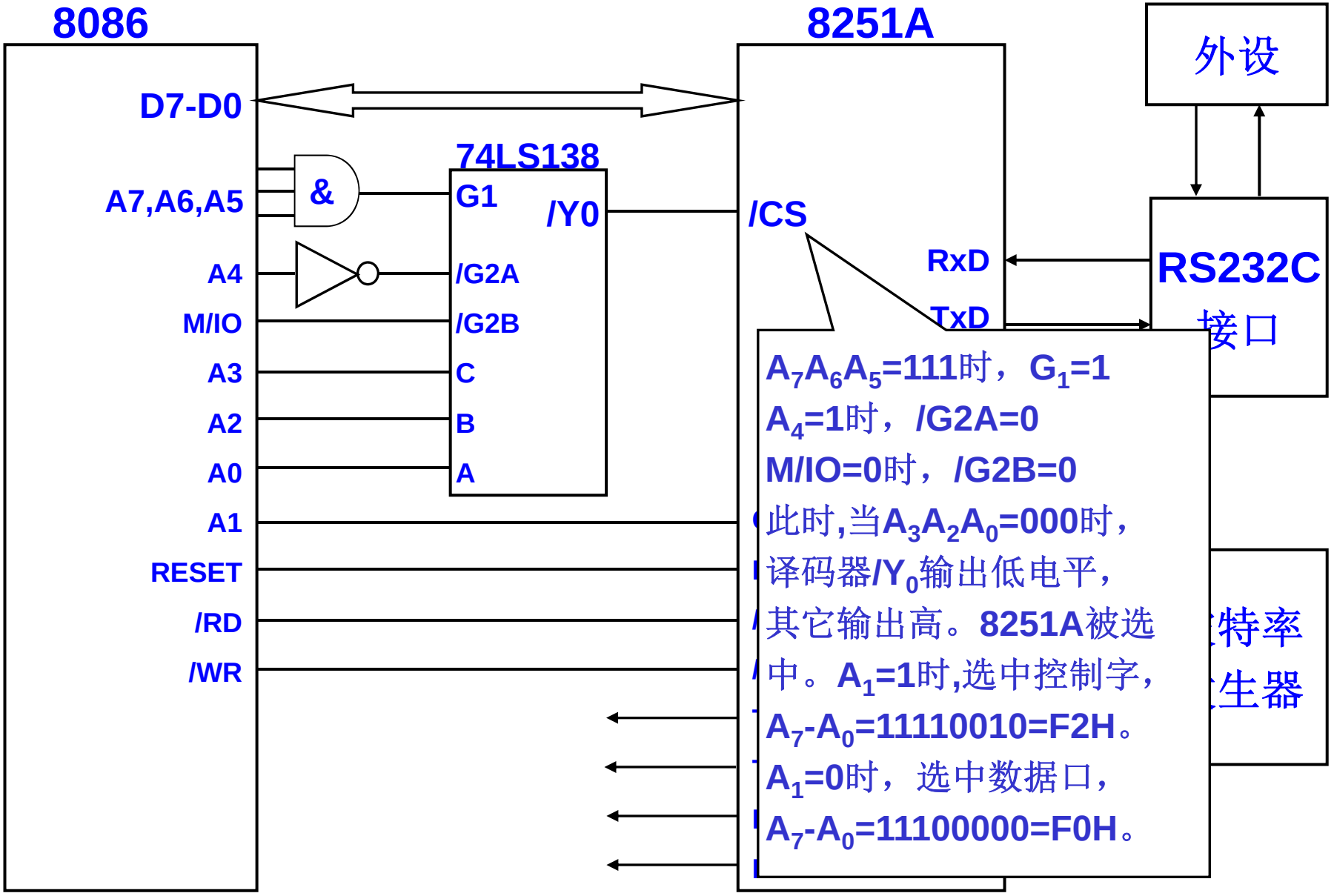

1：启动搜索同步字符
1：使错误标志PE，OE，FE复位
1：使TxD变低，发送空白字符
0：正常工作
EH
IR
RTS
ER
SBRK
RxE
DTR
TxEN
1：内部复位，回方式选择格式
1：允许接收
0：禁止接收
1：允许发送
0：禁止发送
1：使RTS引脚输出低电平
1：使DTR引脚输出低电平

状态字


DSR
SYNDET/BRKDET
FE
OE
PE
TxE
RxRDY
TxRDY
数据装置准备好
同步检测/断点检测
帧错误（只用于异步方式
溢出错误
奇偶错误
发送器空
接收器准备好
发送器准备好

8251A读写操作端口选择表

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	端口选择和操作
0	0	1	0	CPU从8251A接收数据寄存器读数据
0	1	0	0	CPU向8251A发送数据缓冲器写数据
1	0	1	0	CPU从8251A状态寄存器读状态
1	1	0	0	CPU向8251A写控制字（先方式字寄存器、后命令字寄存器）
X	X	X	1	数据总线悬空


8086
D7-D0
A7,A6,A5
A4
M/IO
A3
A2
A0
A1
RESET
/RD
/WR
&
74LS138
G1
/Y0
/G2A
/G2B
C
B
A
8251A
/CS
RxD
TxD
外设
RS232C
接口
$A_7A_6A_5$=111时，G_1=1
A_4=1时，/G2A=0
M/IO=0时，/G2B=0
此时,当$A_3A_2A_0$=000时，
译码器/Y_0输出低电平，
其它输出高。8251A被选
中。A_1=1时,选中控制字，
A_7-A_0=11110010=F2H。
A_1=0时，选中数据口，
A_7-A_0=11100000=F0H。

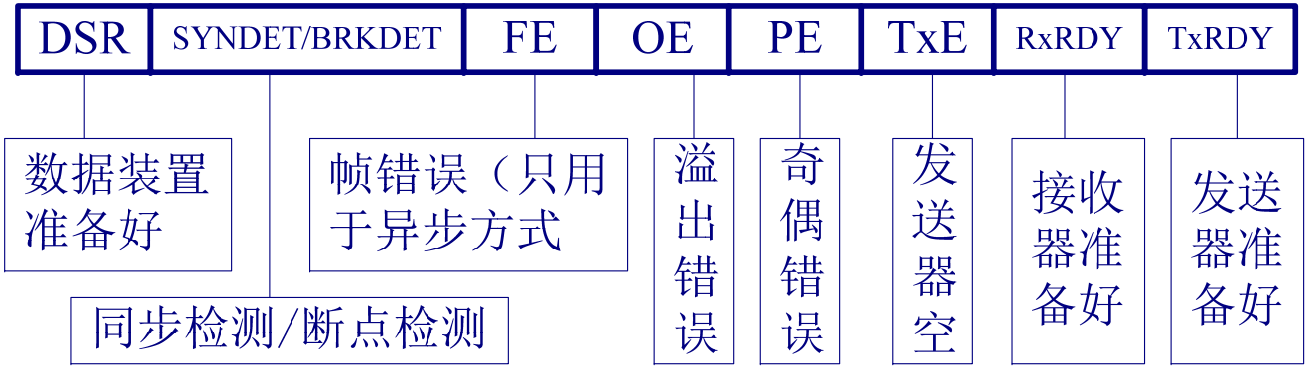
(3) 8251与CPU的数据交换

1) 查询方式

采用查询方式，在数据交换前应读取状态寄存器。

状态寄存器D_0=1，CPU可以向8251数据端口写入数据，完成串行数据的发送；

状态寄存器D_1=1，CPU可以从8251数据端口读出数据，完成一帧数据的接收。



2) 中断方式

8251没有单独的中断请求引脚：

- **TxRDY**引脚可以作为发送中断请求
- **RxRDY**引脚可以作为接收中断请求
- 收发均采用中断方式时， **TxRDY**、 **RxRDY**可以通过或门与系统总线的中断请求线连接。在**CPU**响应中断转到**ISP**中时，再对状态寄存器进行查询，以区分是发送中断还是接收中断。

3) 8251编程示例

例：编写8251异步模式下的接收和发送程序，完成256个字符的发送和接收，设端口地址：208H，209H，波特率因子16，1起始位，1停止位，无奇偶校验，每字符8位。

发送程序

```
LEA DI, Buf1
MOV DX,209H
MOV AL,00H        ;复位
OUT DX，AL
CALL DELAY
MOV AL,00H        ;复位
OUT DX，AL
CALL DELAY
MOV AL,00H        ;复位
OUT DX，AL
CALL DELAY
MOV AL,40H        ;复位命令
OUT DX，AL
MOV AL，01001110B    ;方式选择字
MOV DX，AL
MOV AL，00110111B    ;工作命令字
OUT DX，AL
MOV CX，256     ;发送256字节
NEXT：  MOV DX, 209H ; 状态字寄存器209H
IN AL，DX        ;状态字
AND AL，01H   ;TxRDY?
JZ NEXT
MOV AL，[DI]
MOV DX，208H ;数据寄存器208H
OUT DX，AL     ;发送
INC DI
LOOP NEXT
```

接收程序：接收256字节，放在buf2中

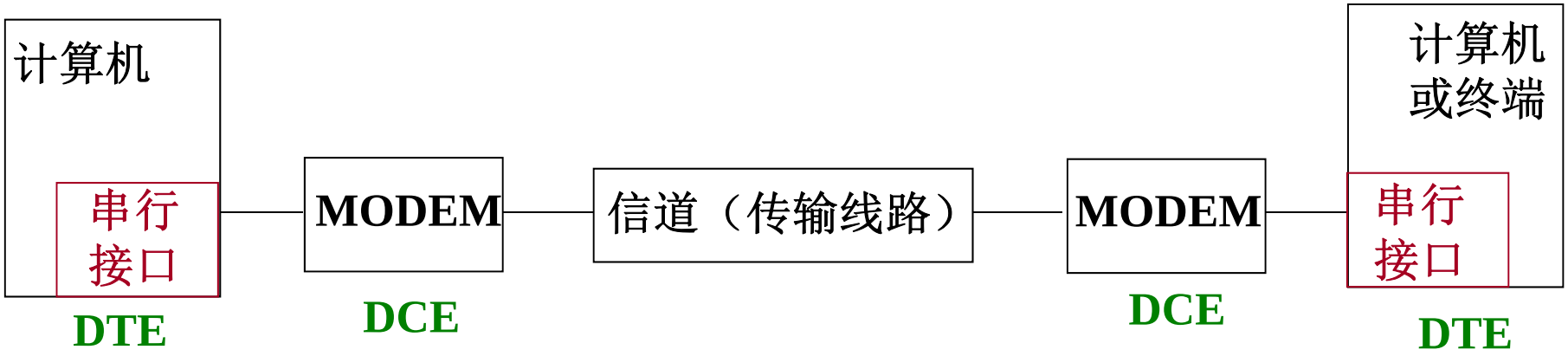
```
Data segment
buf2 DB 256 dup(?)
Data ends
      ┆

      MOV DX,209H
      MOV AL,00H        ;复位
      OUT DX，AL
      CALL DELAY
       MOV AL,00H       ;复位
       OUT DX，AL
            CALL DELAY
       MOV AL,00H       ;复位
       OUT DX，AL
            CALL DELAY
        MOV AL,40H      ;复位
       OUT DX，AL
       MOV AL，01001110B   ;方式字
       OUT DX，AL
       MOV AL，00110111B   ;命令字
       OUT DX，AL
        MOV CX，256      ;接收256字节
        MOV SI，0
NEXT:  MOV DX，209H
        IN AL，DX            ;状态字
       AND AL，02H      ;RXRDY?
       JZ NEXT
       MOV DX，208H
       IN AL，DX          ;接收1字符
       MOV buf2[SI]，AL
       INC SI
       LOOP NEXT
```

3. 串行接口标准

RS-232C串行接口标准

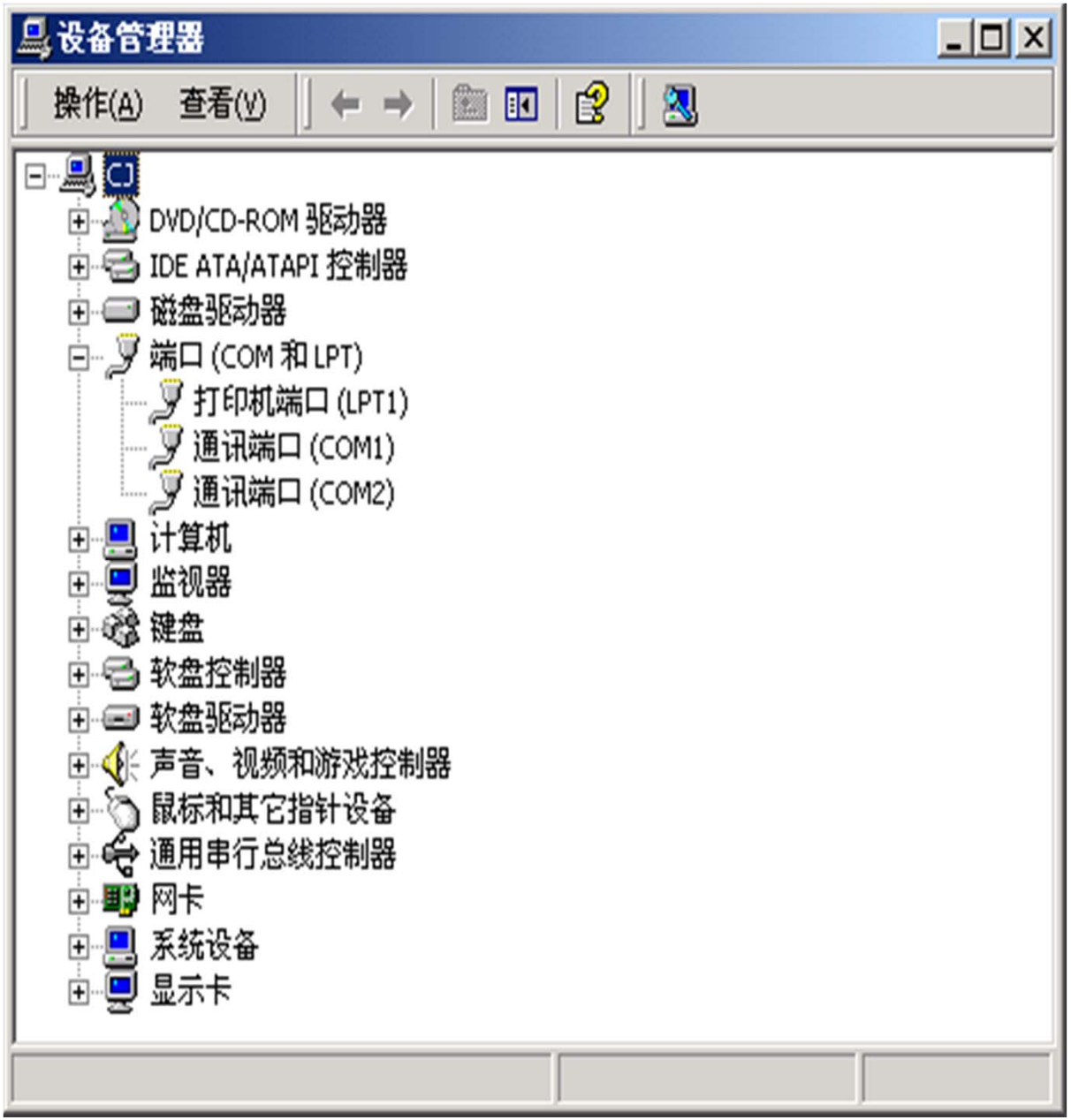
串行通讯系统



为了使来自不同厂家的计算机、外部设备以及数据通信设备都能正确连接，这个接口的机械特性、电气特性、功能特性都要遵循一定的规范，也就是要有一个标准。

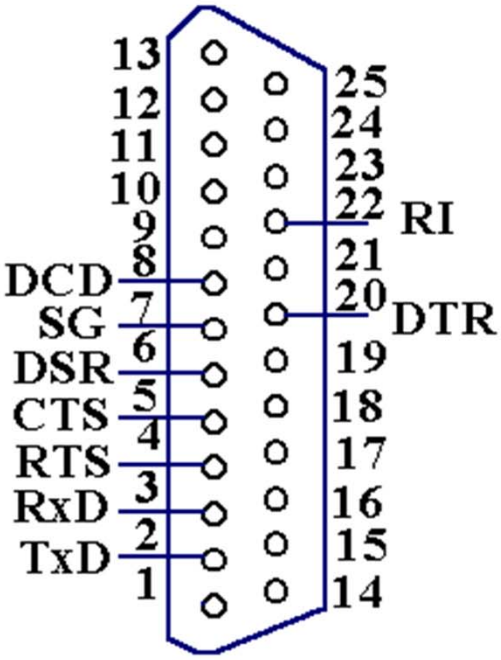
RS-232C

（Recommended Standard推荐标准）是美国电子工业协会（EIA）推荐的标准串行接口，是应用于数据通信设备和数据终端设备之间的标准接口。通常有两个信道COM1、COM2。

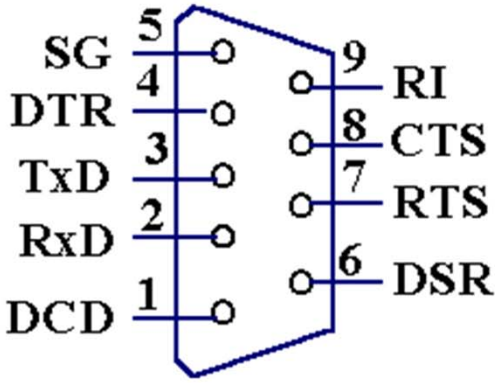


接口机械特性

目前**COM1**、**COM2** 均使用**9**针连接器。



DB-25



DB-9

接口功能特性

RS-232C的引脚

9针引脚	25针引脚	名称	25针引脚	名称
	1	保护地	12	次信号载波检测
3	2	发送数据TxD	13	次信号清除发送
2	3	接收数据RxD	14	次信号发送数据
7	4	请求发送RTS	16	次信号接收数据
8	5	清除发送CTS	19	次信号请求发送
6	6	数据装置准备好DSR	21	信号质量检测
5	7	信号地GND	23	数据速率检测
1	8	载波检测CD	24	终端发生器时钟
4	20	数据终端准备好DTR	9、10	保留
9	22	振铃提示RI	11	未定义
	15	发送时钟TxC	18	未定义
	17	接收时钟RxC	25	未定义

DCD 数据载波检出（DCE→DTE）当本地DCE收到对方的DCE设备送来的载波信号时，使DCD有效，通知DTE准备接收，并且由DCE将接收到的载波信号解调为数字信号，经RxD线送给DTE。

RI 振铃信号（DCE→DTE）当DCE收到交换机送来的振铃呼叫信号时，使该信号有效，通知DTE已被呼叫。

接口电气特性

标准规定：逻辑“1”信号，电平在 –3V ~ -15V 之间；

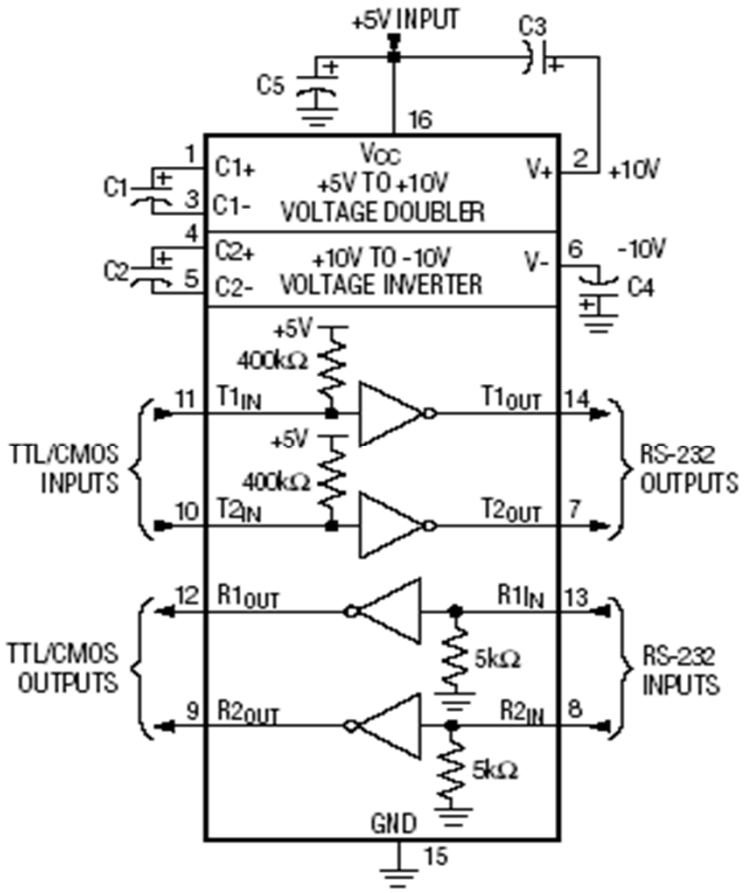
逻辑“0”信号，电平在 +3V ~ +15V 之间；

因此，使用RS-232C与微机接口时，需要将TTL电平（0 ~ 5V）与RS-232C电平进行转换。

MAXIM公司

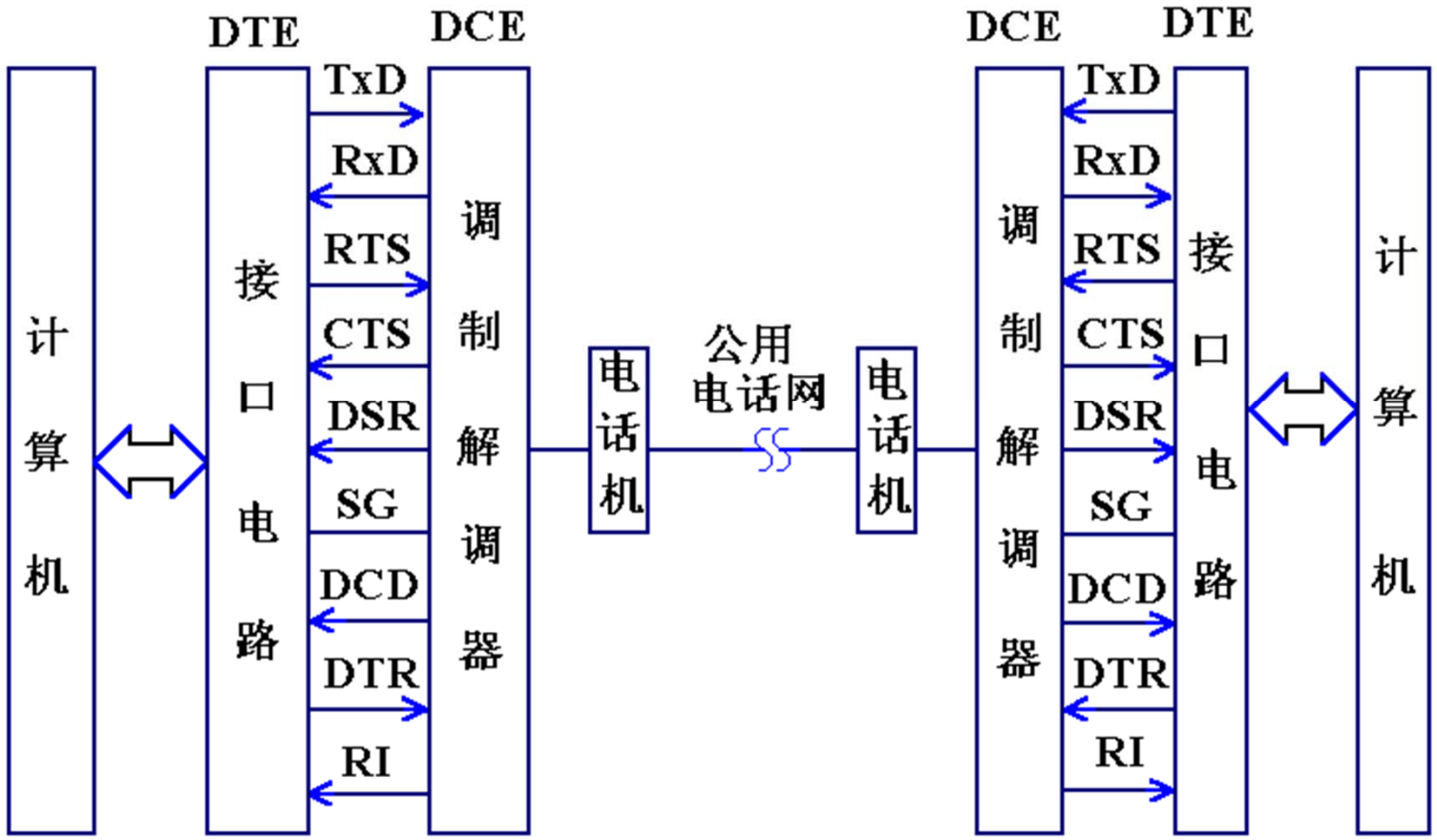
MAX232/MAX233

是现在常用的TTL↔RS232C电平转换器，其最大的优点是只需单+5V电源。

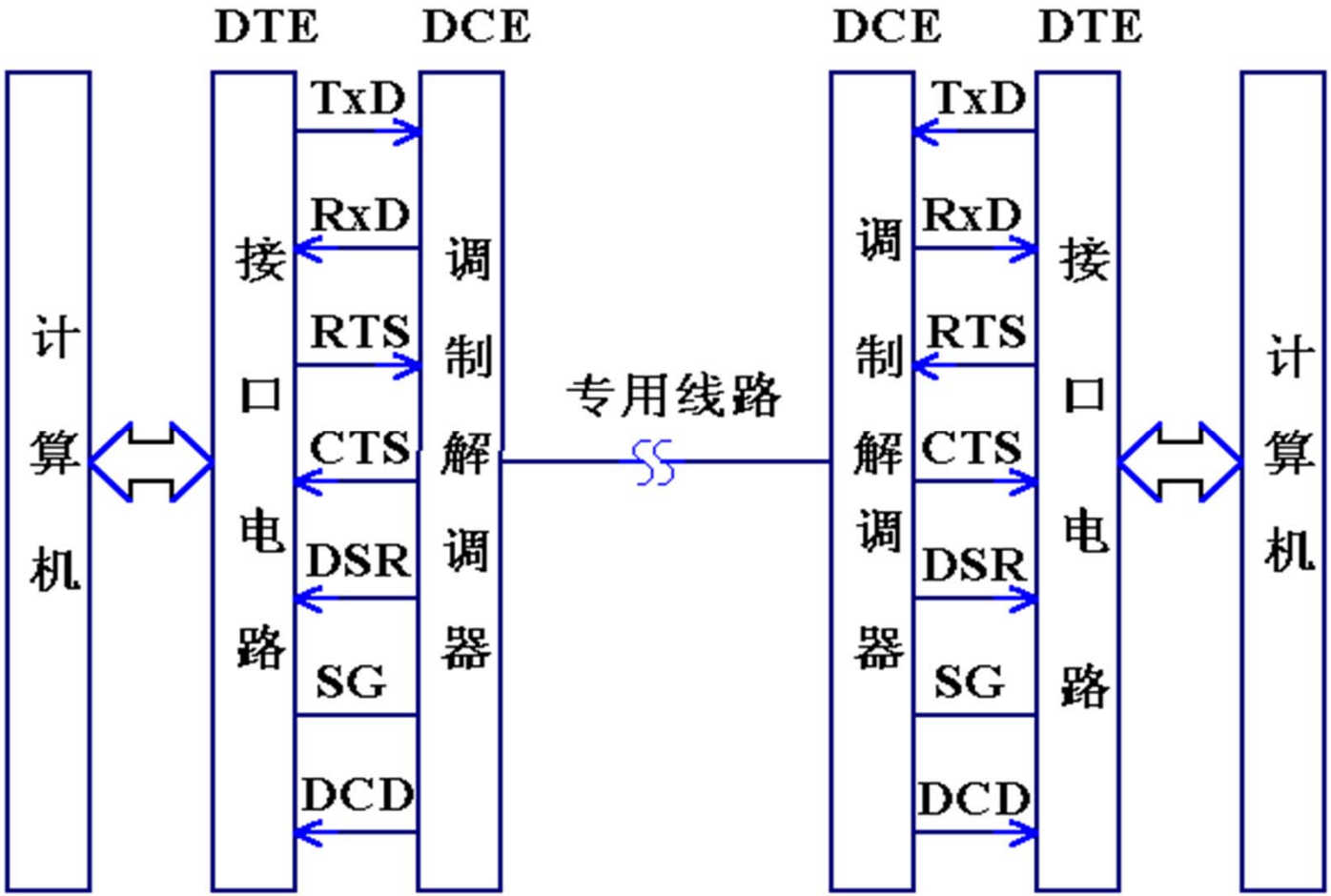


系统连接

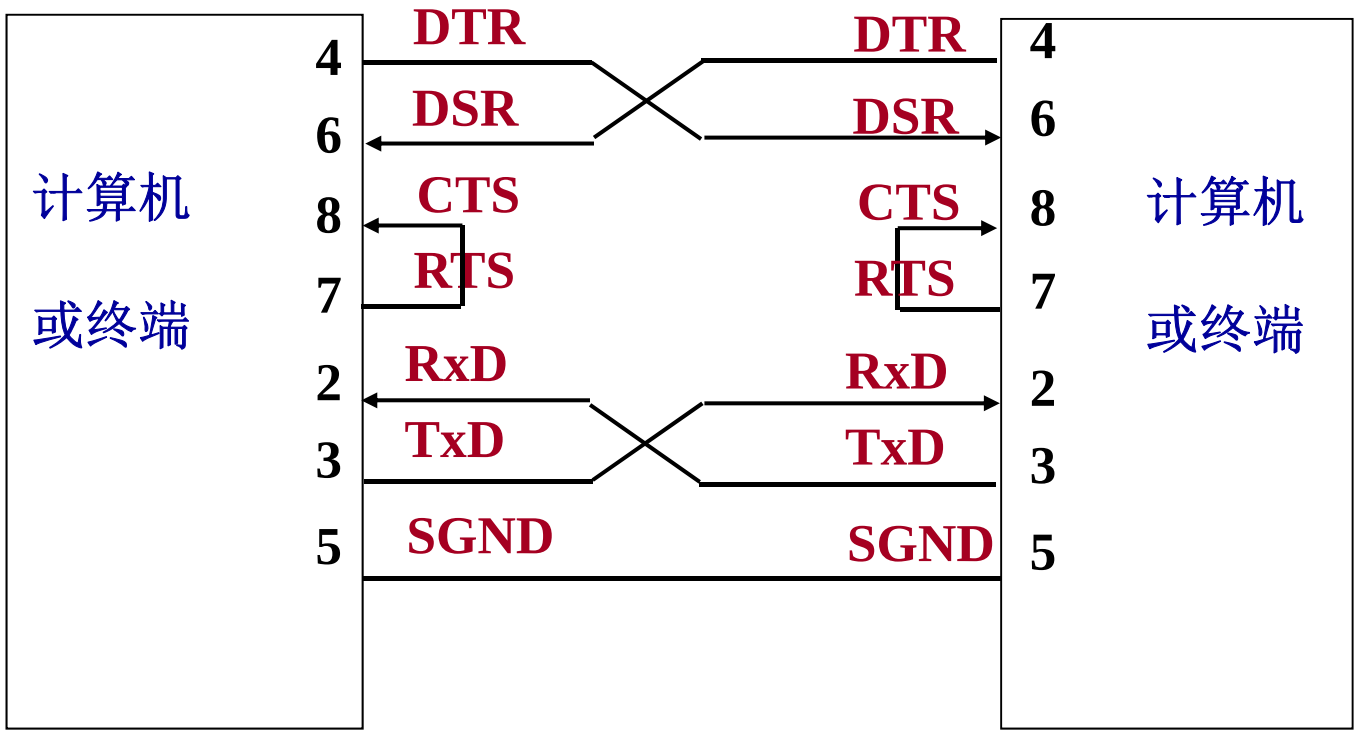
(1) 采用Modem(DCE)和电话网通信时的信号连接



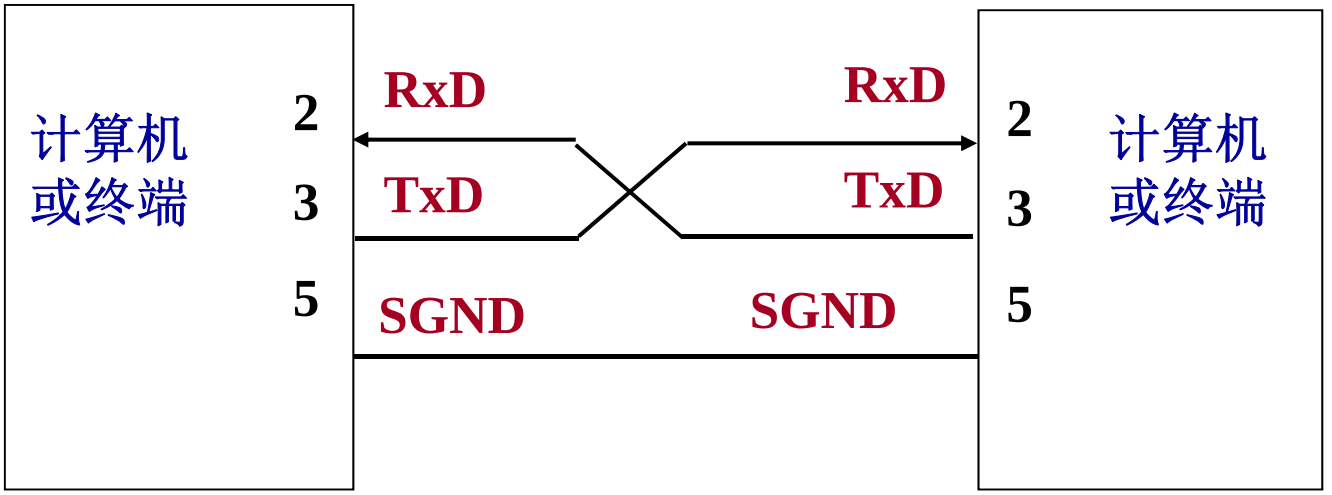
(2) 采用专用线通讯时的信号连接



(3) 无Modem的标准连接

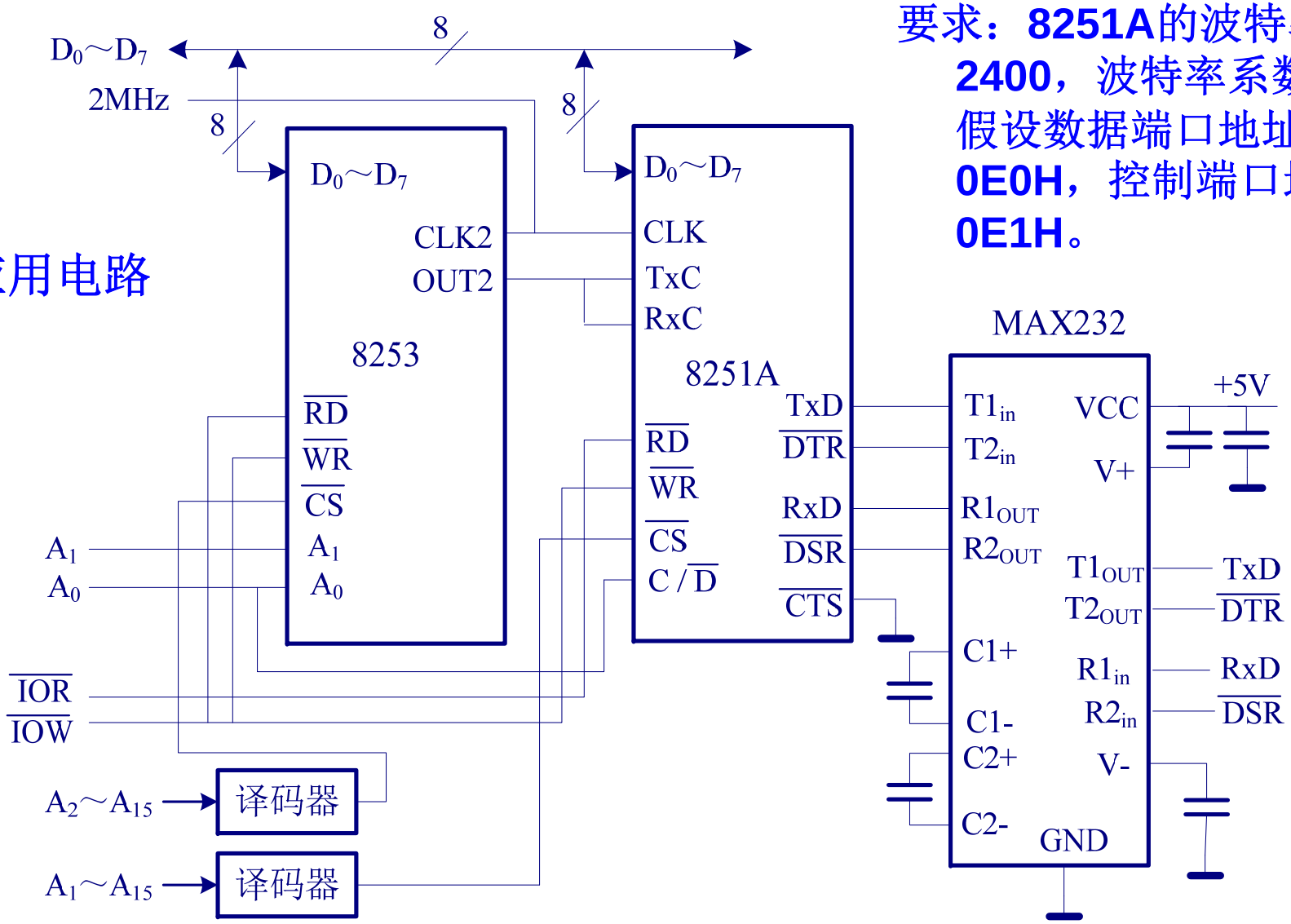


(4) 无Modem的最简单的方式：



应用电路

要求：**8251A**的波特率为**2400**，波特率系数为**16**，假设数据端口地址为**0E0H**，控制端口地址为**0E1H**。



初始化程序：

```
MOV AL,00H        ;复位
OUT 0E1H，AL
CALL DELAY
 OUT 0E1H，AL   ;复位
 CALL DELAY
 OUT 0E1H，AL   ;复位
 CALL DELAY
  MOV AL,40H      ;复位
 OUT 0E1H，AL   ;复位
 CALL DELAY
 MOV AL, 01001110B ;方式字
 OUT 0E1H,AL
 MOV AL, 00100111B
; 命令字，启动发送器和接收器
 OUT 0E1H,AL
```

设发送数据已放入AH中，数据输出程序如下：

```
WAIT：  IN AL，0E1H        ;状态字
      TEST AL，01H      ;RXRDY？
      JZ WAIT
      MOV AL，AH
      OUT 0E0H, AL
```