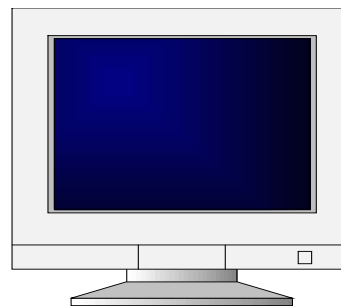


第八章

常用数字接口电路





主要内容：



- 可编程定时计数器8253
- 可编程并行接口8255
- 可编程串行通信接口8250



§ 7.1 可编程定时器8253



7.1.1 概述



在微机系统中,要求对外部信号进行计数或对时间进行精确定时----计数/定时器.计数由外部脉冲信号提供,当达到设定的计数值时,输出一个电平信号,告知外部设备已经计满。定时时间基准由8086内部时钟源提供,经定时时钟分频后得到所需的时间信号,当定时时间到后也输出1个电平信号,告知外部设备定时时间到。

7.1.2 定时与计数的三种方式

(1) 软件定时

利用CPU每执行一条指令都需要几个固定的指令周期的原理，因此执行一个程序段就需要一定的时间，运用软件编程的方式，通过改变指令执行的循环次数就可以控制定时时间，由于它占用了CPU，因而降低了CPU的利用率。

```
例：          MOV    CX, 1000H
              DEALY:  MOV    BX, 1000H
                   LOOP  DEALY
```



(2) 硬件定时

**利用专门的定时电路实现精确定时。不受控制,无法修改。
如555电路。**

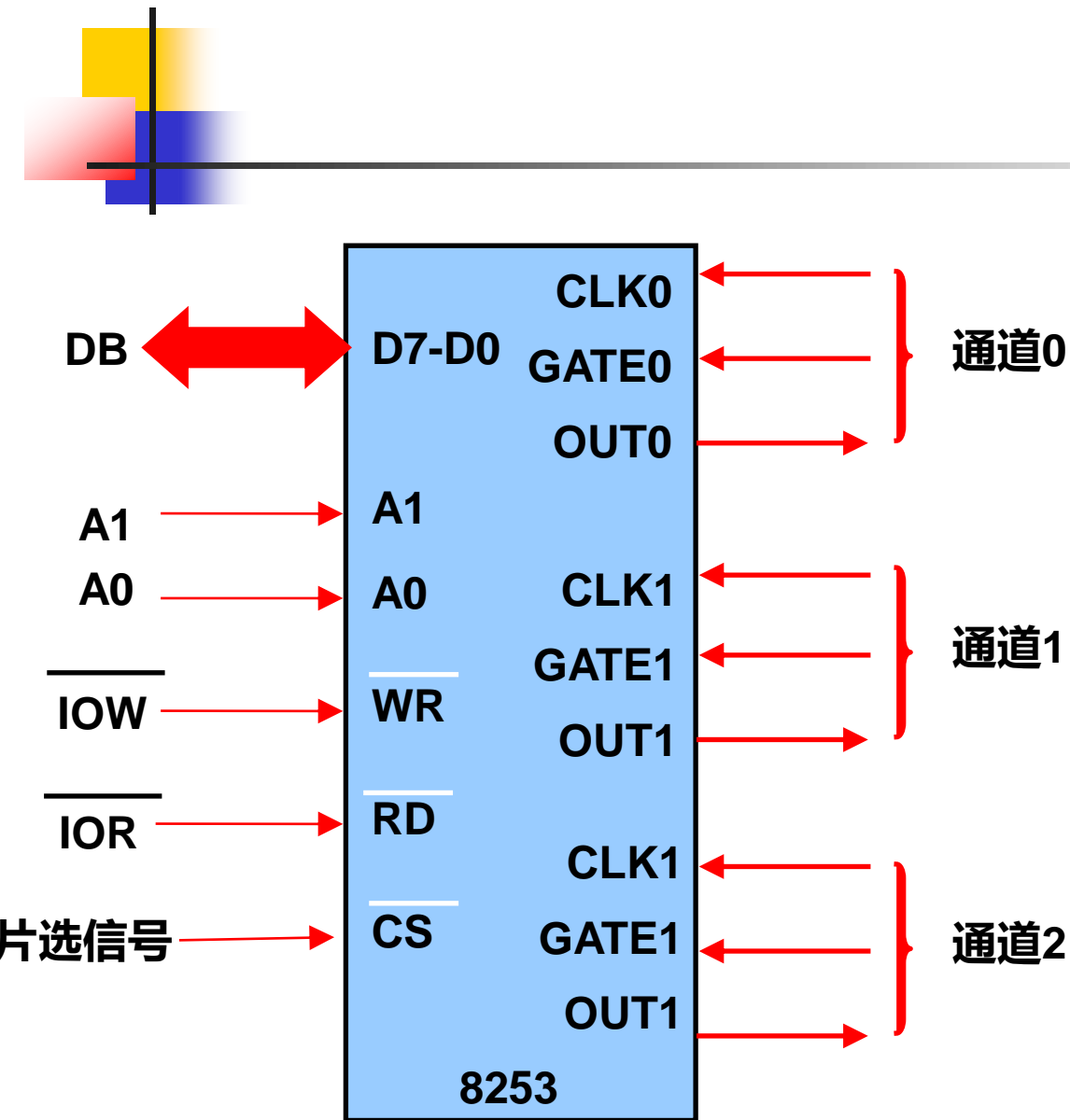
(3) 通用可编程接口芯片定时

**由硬件和软件相结合，可以有多种定时和计数工作方式供
选用，可应用与各中不同型号的计算机系统。**

7.1.3 8253的结构及引脚

8253是Intel公司生产的可编程计数/定时器芯片。8253的操作对所在系统没有特殊要求，其通用性强，适用于各种微处理器组成的系统。它有3个独立的16位减1计数器，每个计数器有6种工作方式，能进行2进制或10进制（BCD码）计数或定时操作，计数速率可达2MHz，所有的输入/输出都与TTL电平兼容。

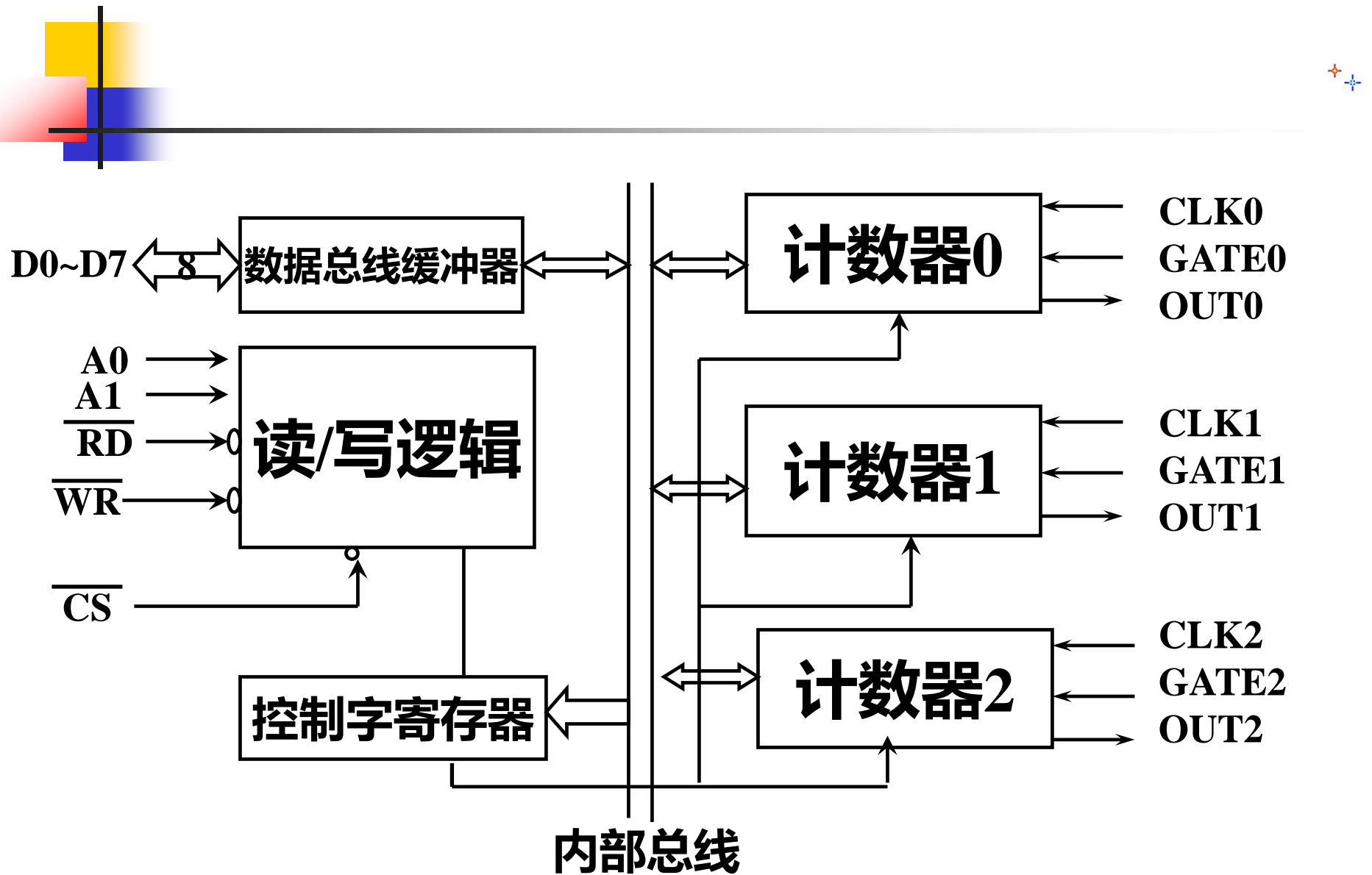
根据计数脉冲的频率及需要定时的时间长度确定计数初值



CLK_n 时钟脉冲输入,
计数器的计时基准。
GATE_n 门控信号输入, 控制
计数器的启停。

OUT_n 计数器输出信号。

A1	A0	选 择
0	0	计数通道0
0	1	计数通道1
1	0	计数通道2
1	1	控制寄存器





(1) 数据总线缓冲器



八位双向三态，用于暂时存放数据，存放的数据有：

- 初时化编程时向8253写入的控制字。
- 向某一通道写入的计数值。
- 从某一通道读计数值。

注意：CPU用输入/输出指令对8253进行读/写操作的所有信息都是通过这8条总线传送的。

(2) 读/写控制逻辑

- 接受CPU的RD、WR、CS、A0、A1 信号，经组合产生对某一端口的操作。
- 当片选信号有效，即CS = 0时，读/写逻辑才能工作。该控制逻辑根据读/写命令及送来的地址信息，决定三个计数器和控制字寄存器中哪一个工作，并控制内部总线上数据传送的方向。

(3) 控制字寄存器

接受CPU的控制字，只能写，不能读，根据控制字决定每个通道的工作方式。



三个计数/定时器的特点



A.计数/定时器0、1、2是3个16位减1计数器，它们互相独立，内部结构和功能相同。

B.每个计数器有3根信号线，它们是时钟输入CLK、门控输入GATE和输出OUT。最高计数频率2MHZ。

C.计数/定时器从CLK端接收时钟脉冲或事件计数脉冲，在脉冲下降沿按照2进制或10进制从预置的初值开始进行减1计数。当计数值减到零时，从OUT端送出一个信号。

D.计数器在开始计数和计数过程中，都要受到门控信号GATE的控制， $GATE = 1$ 可以计数， $GATE = 0$ ，停止计数。

7.1.4 计数启动方式

软件启动过程

- GATE端保持为高电平
- 写入计数初值后的下一个CLK脉冲的下降沿开始计数

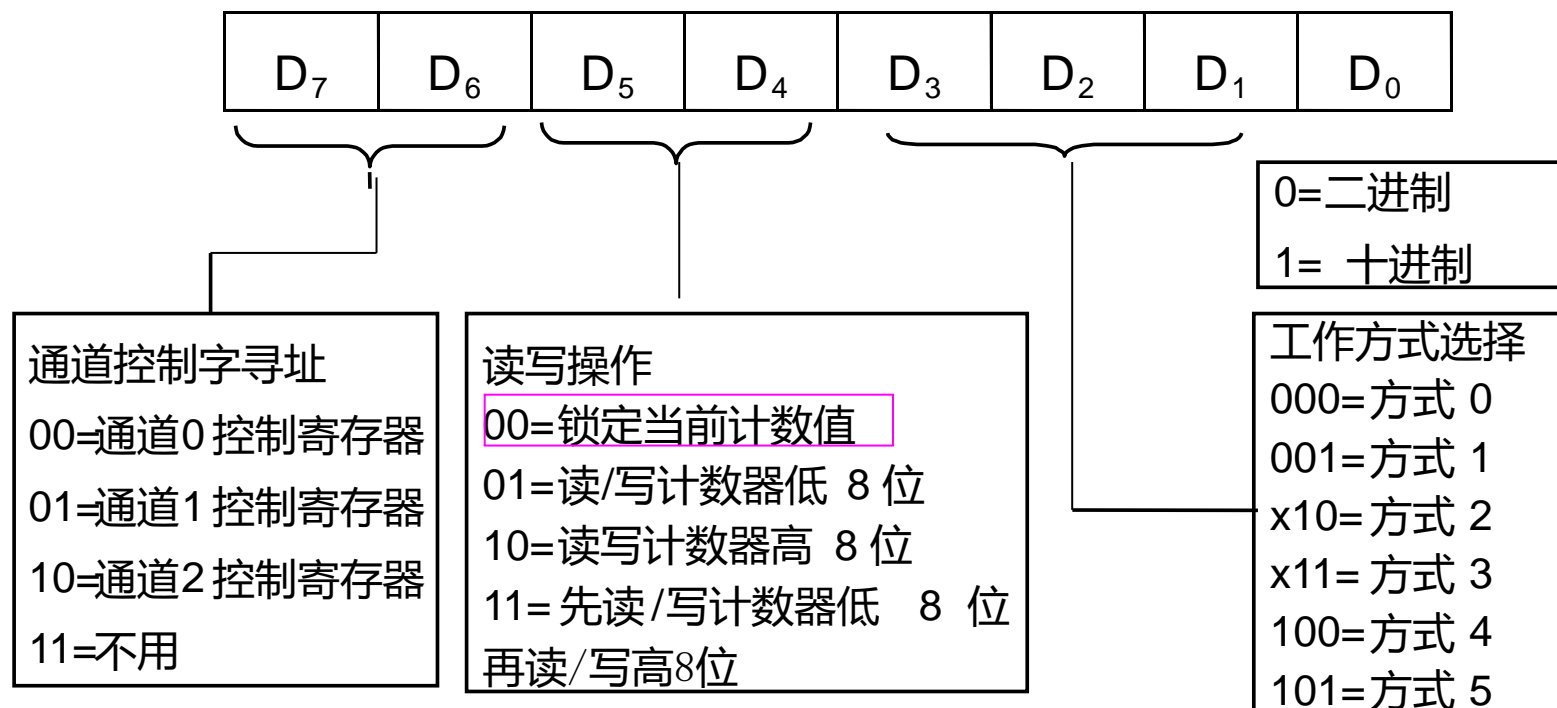
硬件启动过程

- GATE端有一个上升沿
- 对应CLK脉冲的下降沿开始计数

8253内部寄存器的读/写操作

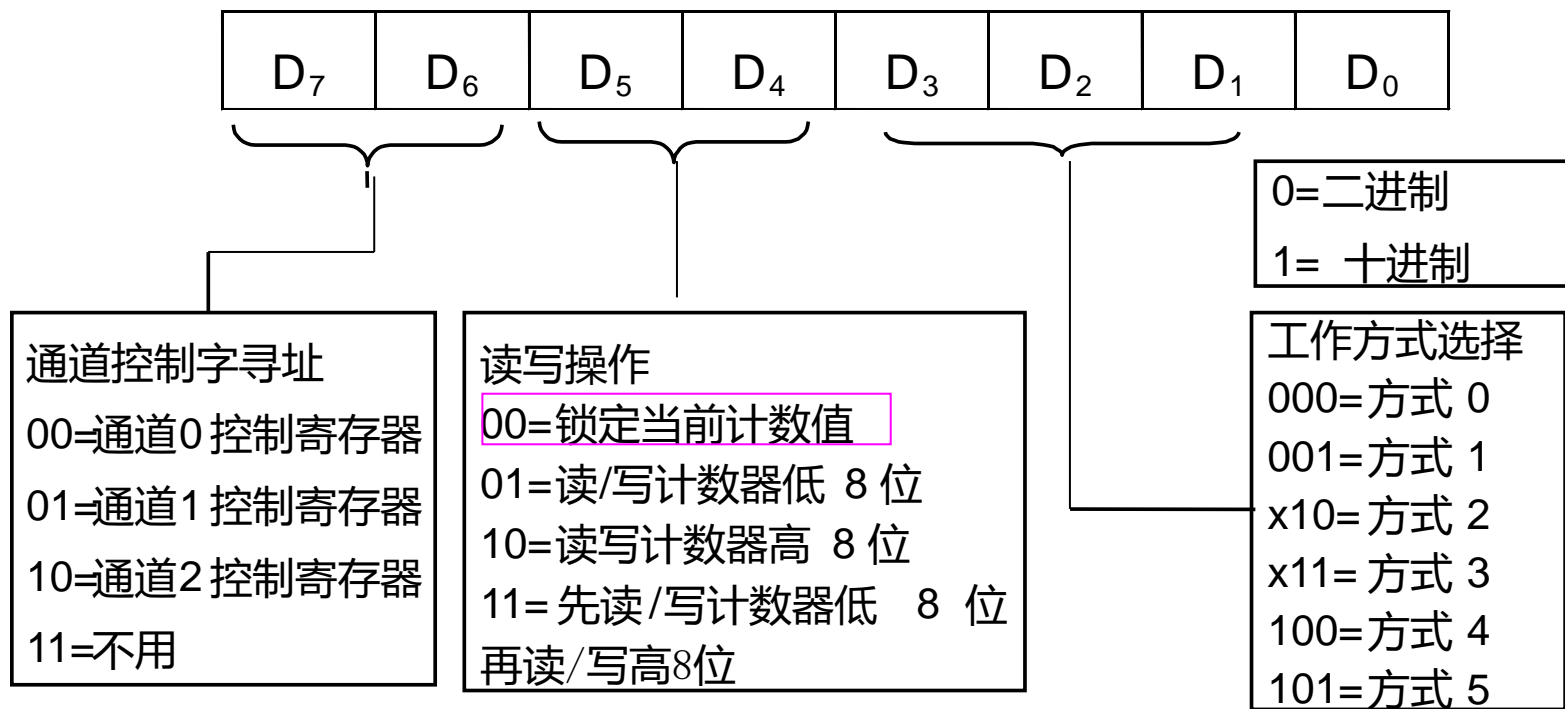
\overline{CS}	A1	A0	选中的端口	\overline{WR}	\overline{RD}	操作
0	0	0	计数器0	0	1	计数初值 写入计数器 0
0	0	1	计数器1	0	1	计数初值 写入计数器 1
0	1	0	计数器2	0	1	计数初值 写入计数器 2
0	1	1	控制寄存器	0	1	向控制字寄存器写控制字
0	0	0	计数器0	1	0	读计数器 0 当前计数值
0	0	1	计数器1	1	0	读计数器 1 当前计数值
0	1	0	计数器2	1	0	读计数器 2 当前计数值

8253控制字格式



(1) D₇D₆: 计数器选择位。这两位表示这个控制字是对哪一个计数器设置的。

00: 计数器0; 01: 计数器1; 10: 计数器2; 11: 非法选择。



(2) D₅D₄: 数据读/写格式选择位。CPU在对计数器写入初值和读取它们的当前值时, 有几种不同的格式, 由这两位来决定。

- 00:** 将计数器当前值锁存于缓冲器中, 以便读出;
- 01:** 只读/写计数器的低8位, 写入时高8位自动设置为0;
- 10:** 只读/写计数器的高8位, 写入时低8位自动设置为0;
- 11:** 对16位计数器进行两次读/写操作, 低字节在前, 高字节在后, 两次操作的地址相同。

(3) D3D2D1: 计数器工作方式选择位

8253的每个计数通道有6种不同的工作方式，由这3位决定。

000——方式0； 001——方式1； x10——方式2
x11——方式3； 100——方式4； 101——方式5

(4) D0: 数制选择

8253的每个计数器有两种数制2进制和10进制，由这一位决定选择哪一种。D0=0表示采用2进制计数，写入的初值范围为0000H~FFFFH，其中0000H是最大值，代表65536；D0=1表示采用10进制计数，写入的初值范围为0000~9999，其中0000是最大值，代表10000。



设置初值命令

初值为8位

初值为16位

先写入低8位

后写入高8位

计数初值的范围

⑩采用二进制计数

若是8位二进制计数（计数值 ≤ 255 ）

若是16位二进制计数（计数值 ≤ 65536 ）

可以先把该十进制计数初值n直接传送给AX，然后分两次写入8253指定端口，即：

```
MOV AX, n
```

```
OUT PORT, AL ; 先写低8位（PORT为端口号）
```

```
MOV AL, AH
```

```
OUT PORT, AL ; 后写高8位
```



⑩采用十进制 (BCD码) 计数

须在8253初始化编程中把计算得到的十进制计数初值n加上**后缀**

H，以便在相应的传送指令执行后能够在AL(或AX)中得到十进制制

数n的BCD码表示形式。

例如 $n=50$ ，则应按如下方式写入：

MOV AL, 50H



如果 $n=1250$ ，则需分两次写入，即：

```
MOV AL, 50H ;  
OUT PORT, AL ; 先写低8位  
MOV AL, 12H ;  
OUT PORT, AL ; 后写高8位
```

也可按如下方法两次写入：

```
MOV AX, 1250H ;  
OUT PORT, AL ; 先写低8位  
MOV AL, AH ;  
OUT PORT, AL ; 后写高8位
```

8253初始化步骤

① 向控制字寄存器写入控制字，对使用的计数器规定工作方式。

② 向使用的计数器写入计数初值。

例题：设计计数器1 工作在方式1，按BCD码计数计数值为4000。设8253的端口地址为E0H~E3H。试写出初始化程序段。

控制字=01 11 001 1B = 73H

```
MOV AL, 73H;
```

```
OUT 0E3H, AL;
```

```
MOV AX, 4000H;
```

```
OUT 0E1H, AL; 先写入低字节 到计数器 1。
```

```
MOV AL, AH;
```

```
OUT 0E1H, AL; 后写入高字节 到计数器 1。
```



7.1.5 工作方式

- 方式0：计数结束产生中断输出（软件启动）**
- 方式1：重复触发的单稳输出（硬件启动）**
- 方式2：分频器（软件启动或硬件启动）**
- 方式3：方波发生器（软件控制或硬件控制）**
- 方式4：选通信号发生器（软件触发）**
- 方式5：选通信号发生器（硬件触发）**



六种工作方式均遵循的以下几条原则：



(1) 写入控制字时所有的控制逻辑电路复位，输出端OUT进入初始化状态。工作方式不同，输出电平不同。

(2) 初值写入后，需经过一个时钟周期减1计数器才开始工作，时钟脉冲的下降沿使计数器减1计数（GATE必须为高时）。如果初值为0，则为最大值（65536:二进制或为10000:十进制）。

(3) 对门控信号GATE的采样是在计数脉冲的上升沿进行，所以有电平触发和边沿触发两种方式

(4) 8253在上电后必须先对其初始化设置



以下各例均设8253各端口地址40H~43H。

A端口=40H， B端口=41H， C端口=42H， 控制端

口=43H

(1) 方式0 —计数结束中断

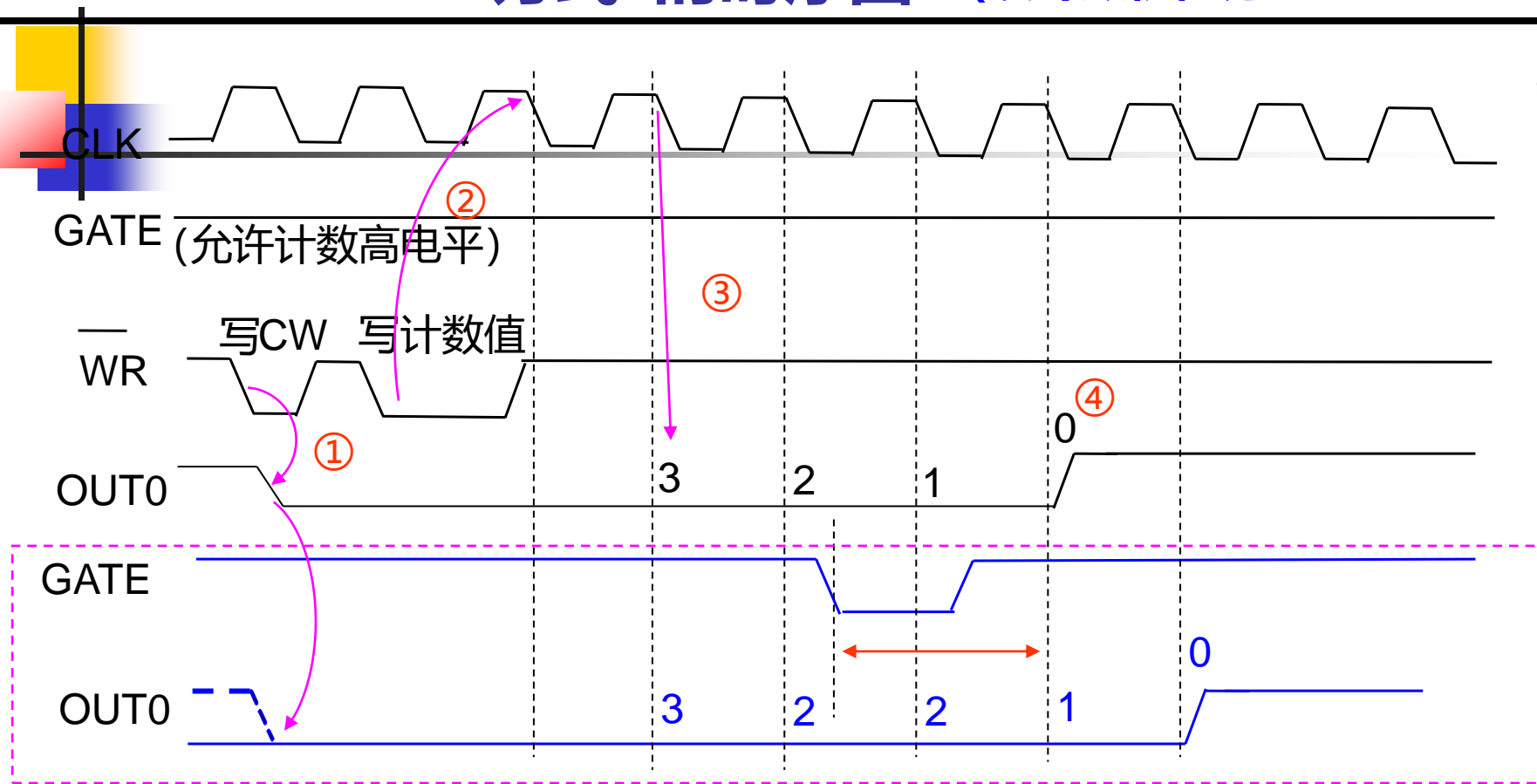
A. 写入方式0控制字后，OUT立即变成低电平。当写入计数初值N后，若GATE为高电平，计数器开始计数。

B. 在计数过程中，OUT端一直维持为低，直到计数为0（结束）时，OUT端变为高，（可以设置为向CPU发出中断请求）。

C. 在计数过程中，若GATE变为低电平，暂时停止计数，OUT不变，直到GATE变为高电平后继续接着计数，到计数为0（结束）时，OUT端变为高。（可以设置为向CPU发出中断请求）。

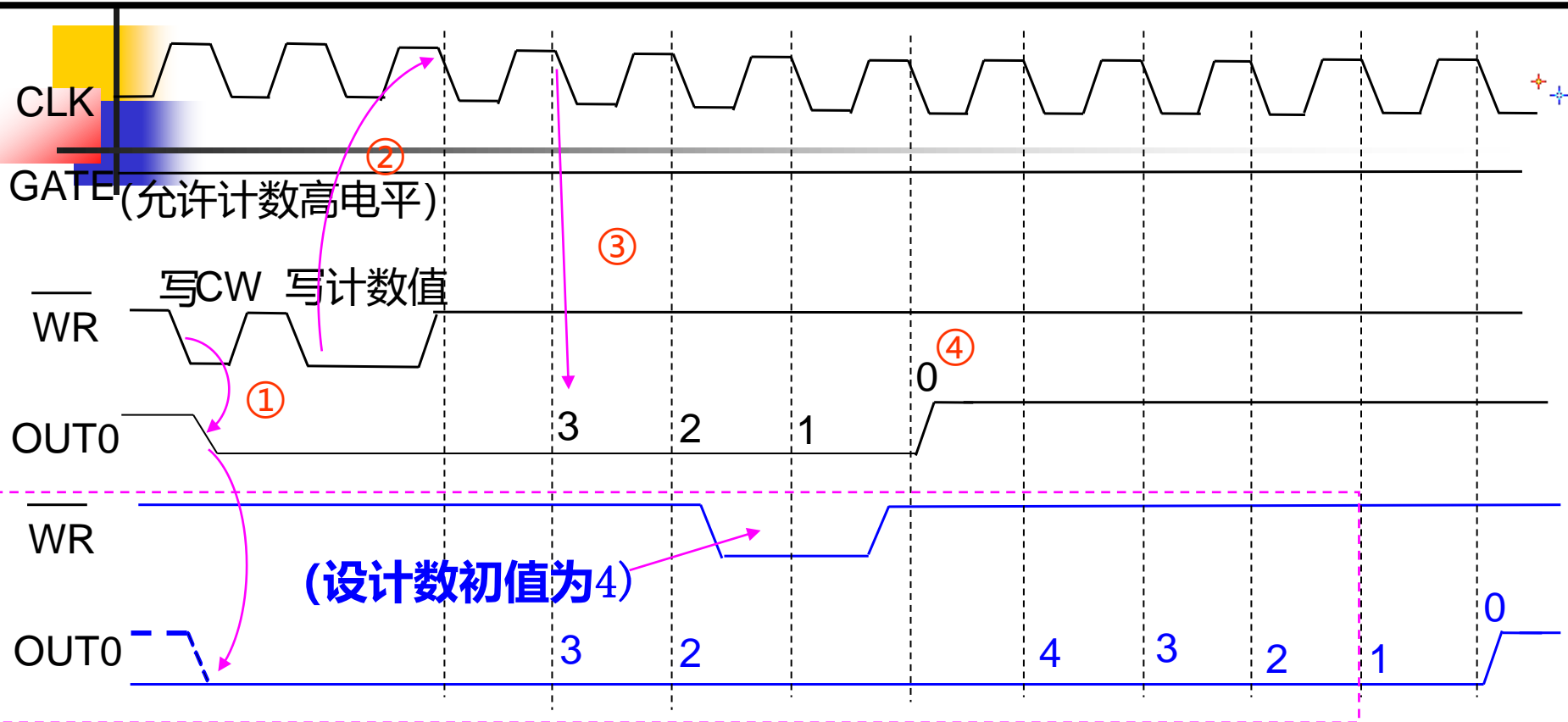
8253用作计数时，一般都工作在方式0。

方式0的时序图 (设计数初值为3)



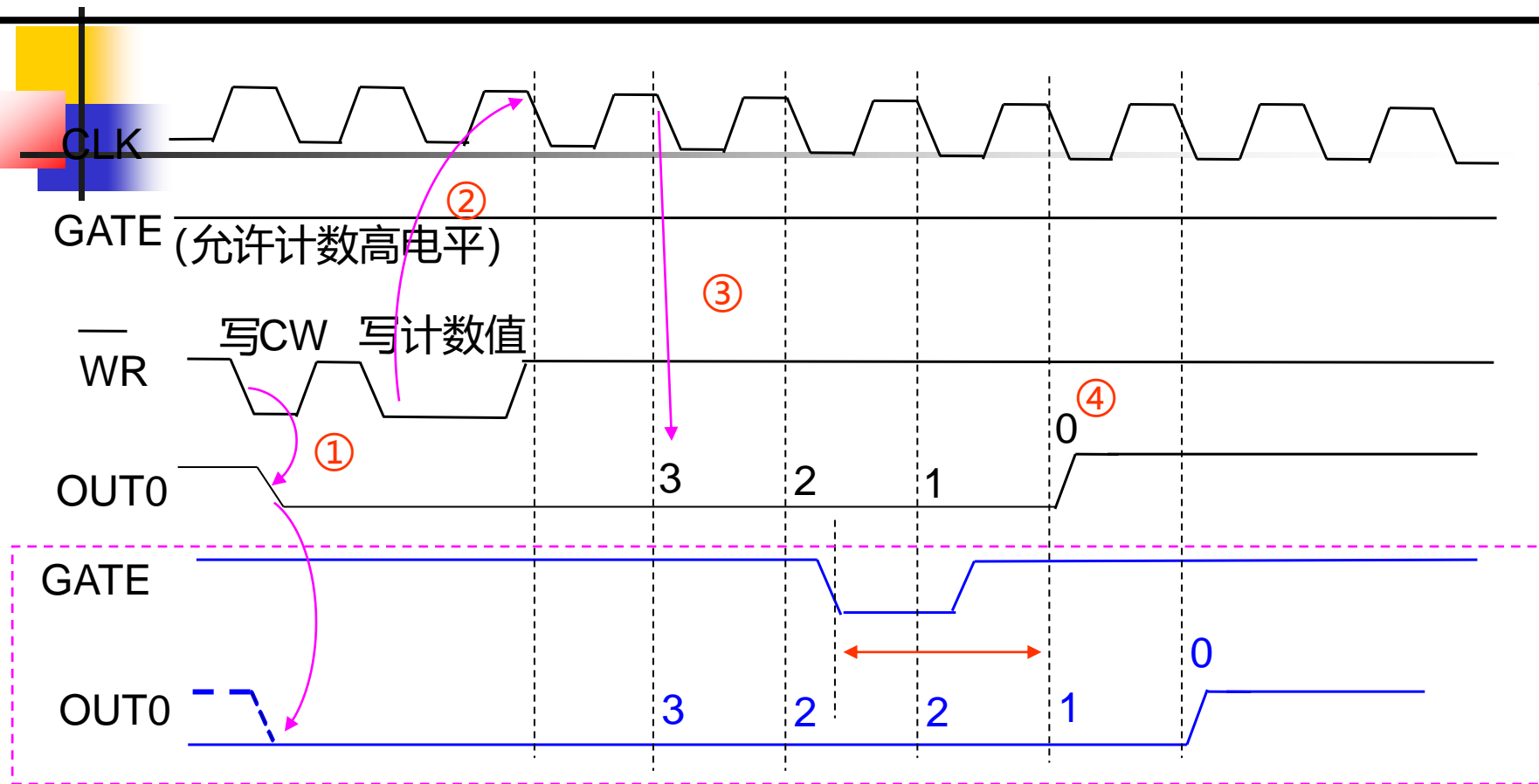
特点①: 计数器只计一遍数。当计数到0时，并不恢复计数初值，不开始重新计数，输出端OUT由低变高且一直保持为高。只有当写入一个新的计数初值后，OUT才变低，开始新的计数。

方式0的时序图 (设计数初值为3)



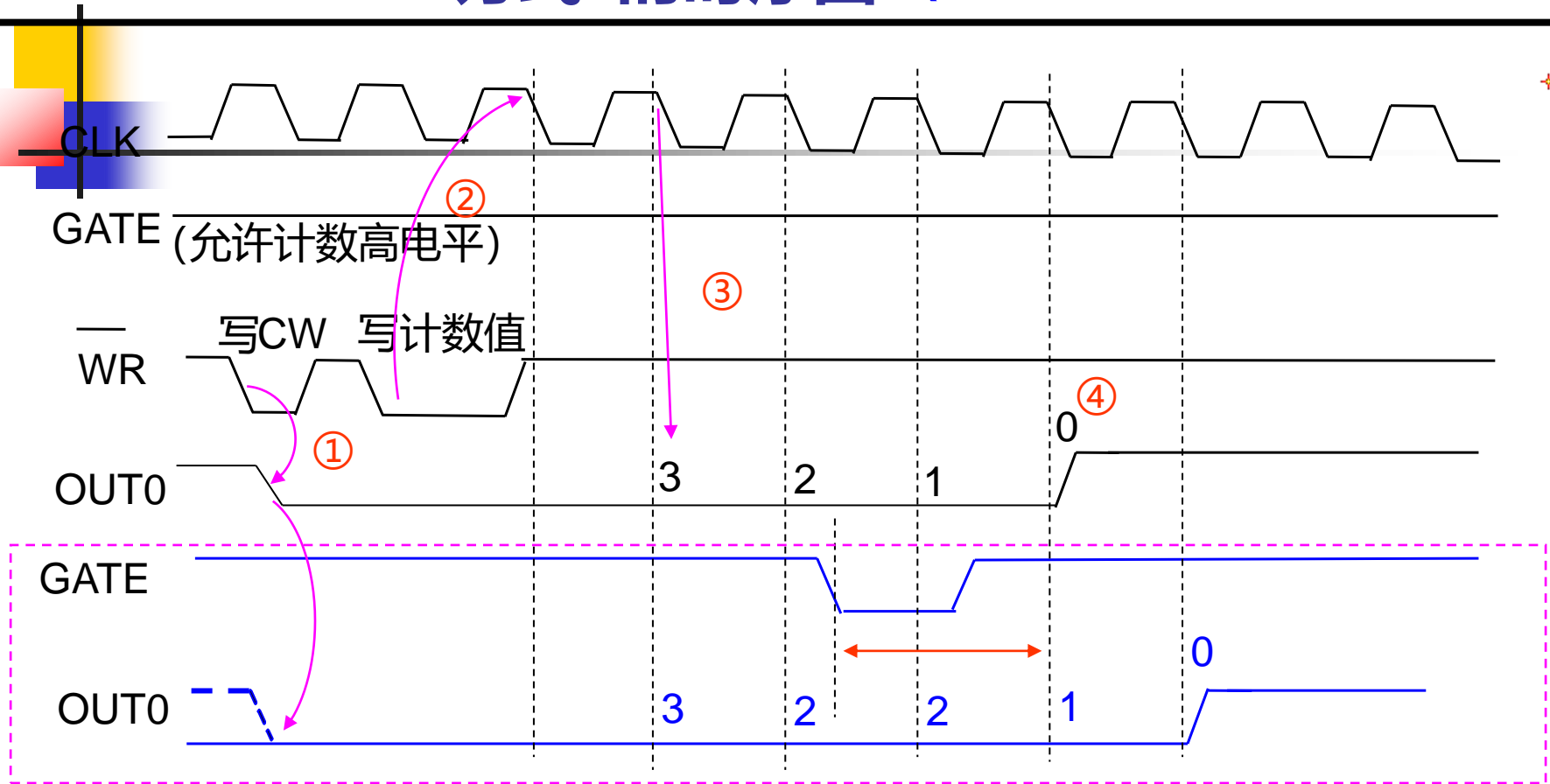
特点②：在计数过程中可改变计数值。如果计数值为8位，在写入新的8位计数值后，计数器将按新的计数值重新开始计数。如果计数值为16位，在写入第一个字节后，计数器停止计数，在写入第二个字节后，计数器按照新的数值开始计数。

方式0的时序图 (设计数初值为3)



特点③： 在计数过程中，可由门控信号GATE控制暂停。当GATE=0时，计数器暂停计数；当GATE变为1后，就接着计数。

方式0的时序图 (设计数初值为3)



特点④: 方式0的OUT信号在计数到0时由低变高，可作为中断请求信号。但由于在8253内部没有中断控制电路，通常接8259中断控制器。

例：设8253计数器通道0工作于方式0。分别采用用8位2进制和BCD码计数，初始化程序如下：

MOV AL, 00010000B ; 设置控制字

OUT 43H, AL ; 写入控制字寄存器

MOV AL, 50 ; 设置计数初值 (计数初值为50)

OUT 40H, AL ; 写入计数初值寄存器

其计数值为5000，十进制，初始化程序如下：

MOV AL, 00110001B ; 设置控制字

OUT 43H, AL ; 写入控制字寄存器

MOV AX, 5000H ; 写入计数初值寄存器 (计数初值为5000H)

OUT 40H, AL ; 设置计数初值低8位

MOV AL, AH ;

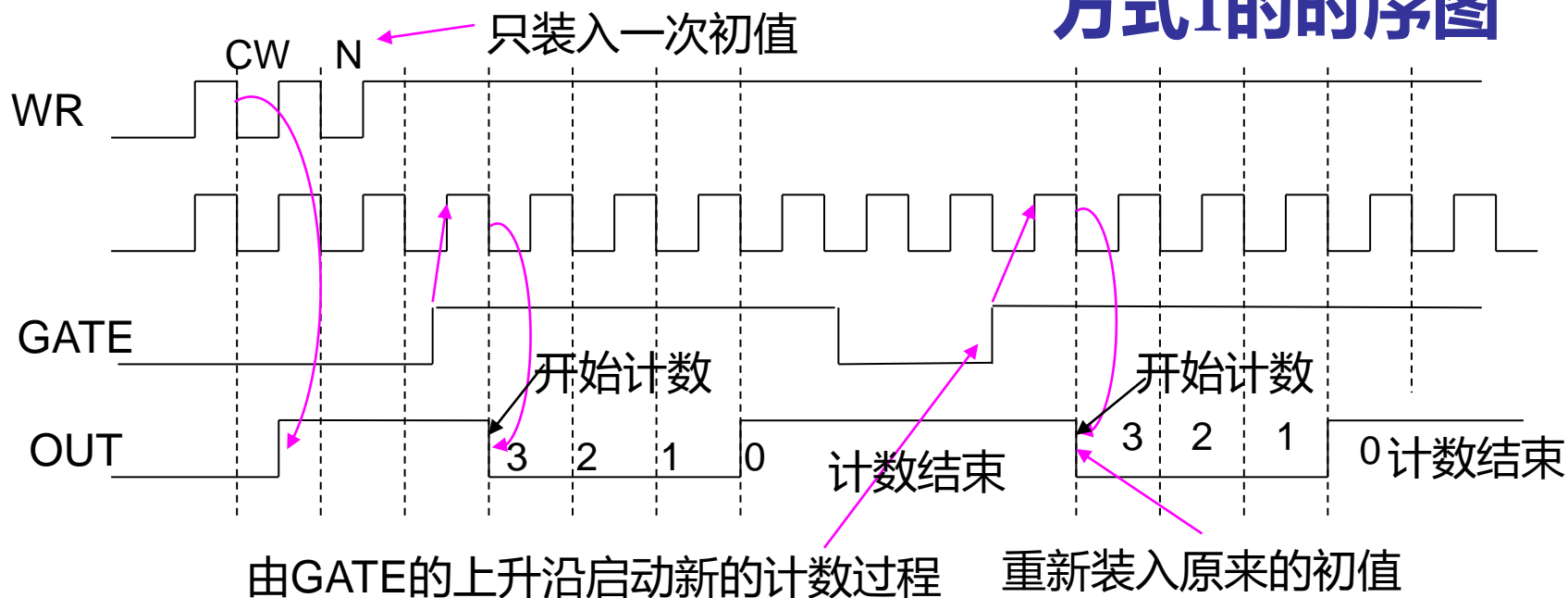
OUT 40H, AL ; 设置计数初值高8位

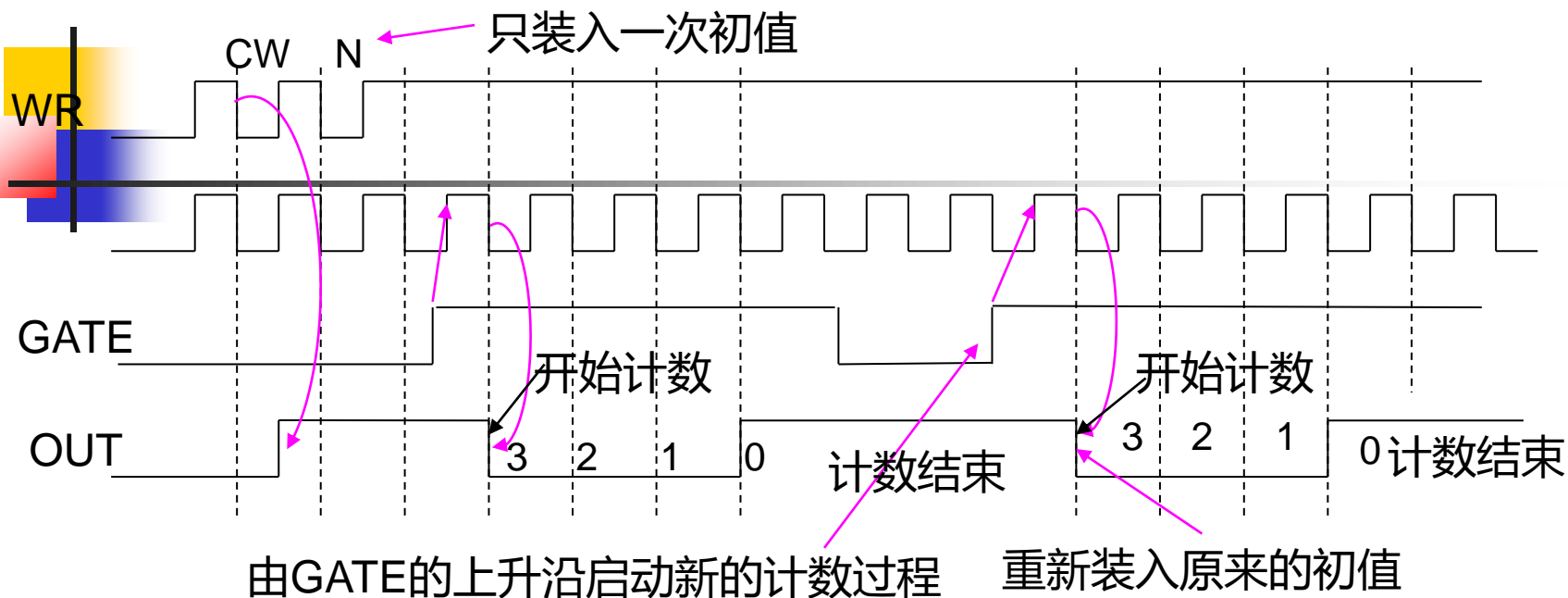
注意：计数方式为10进制，但送数据时，计算机要把10进制数转换成16进制数！

(2) 方式1—可编程的单脉冲

工作过程：写入控制字后，OUT输出高电平，此时并不开始计数。写入计数初值后当门控信号GATE变为高电平时（边沿触发），启动计数，OUT输出变低。在整个计数过程中，OUT都维持为低，直到计数到0时，输出变为高。因此，输出为一单脉冲，其低电平维持时间由装入的计数初值来决定。

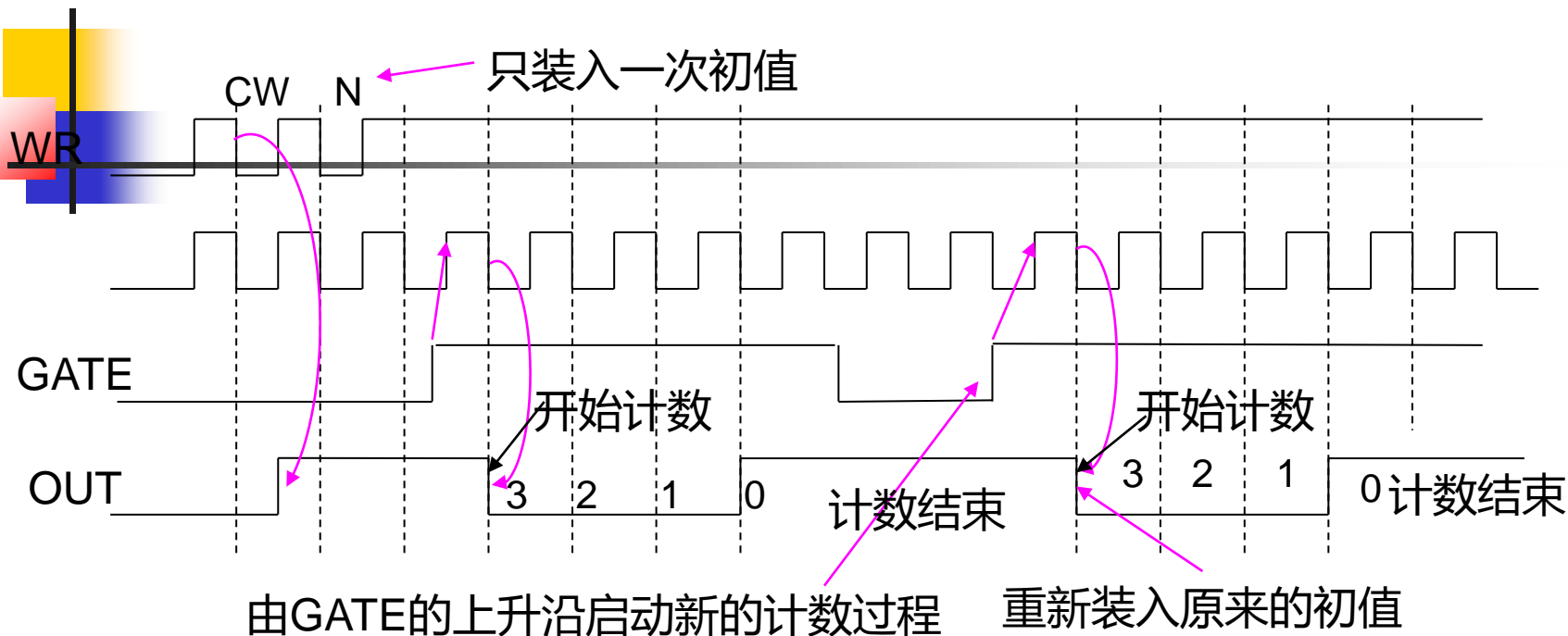
方式1的时序图





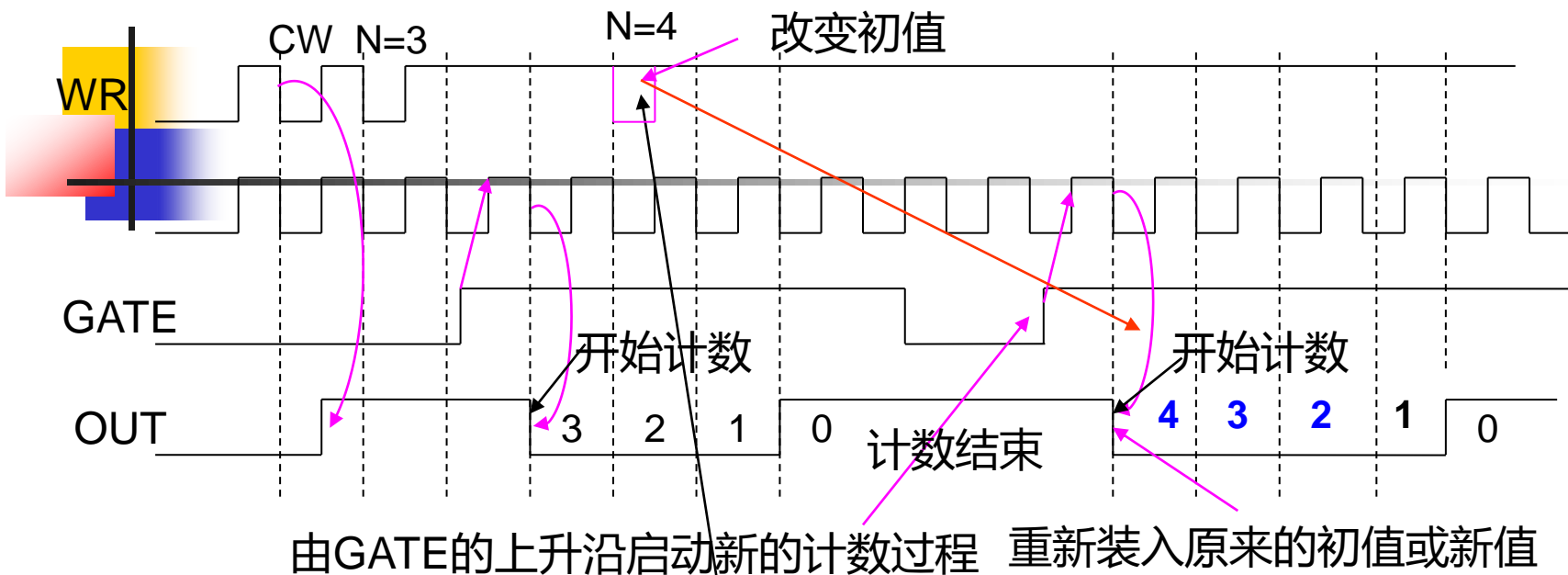
方式1的特点①：

当计数到0后，计数器可再次由外部启动(GATE信号上升沿)，按原计数初值重新开始计数，输出单脉冲，而不需要再次送一个计数初值。



方式1的特点②:

计数过程中门控信号再次被触发，计数器按计数初值寄存器的值重新开始计数。



方式1的特点③:

计数过程中改变计数初值不影响本次计数过程。若门控信号再次被触发，则计数器才按新的计数值计数。



比较方式0和方式1， 有以下几点不同：

A. 方式0设置计数初值后立即计数；方式1设置计数初值后不立即计数，直到有外部触发信号后才开始计数。

B. 方式0在计数过程中能用门控信号暂停计数；方式1在计数过程中若有门控脉冲时不停止计数，而是使计数过程重新开始。

例：设计数器通道1工作于方式1，按二进制计数，计数初值为40H，它的初始化程序为：

~~MOV AL, 01010010B ; 工作方式控制字~~

OUT 43H, AL

MOV AL, 40H ; 送计数初值

OUT 41H, AL ; 注意计数初值送到该通道

计数初值为1234H，它的初始化程序为：

MOV AL, 01110010B ; 工作方式控制字

OUT 43H, AL

MOV AL, 34H ; 送计数初值

OUT 41H, AL ; 注意顺序

MOV AL, 12H ; 送计数初值

OUT 41H, AL

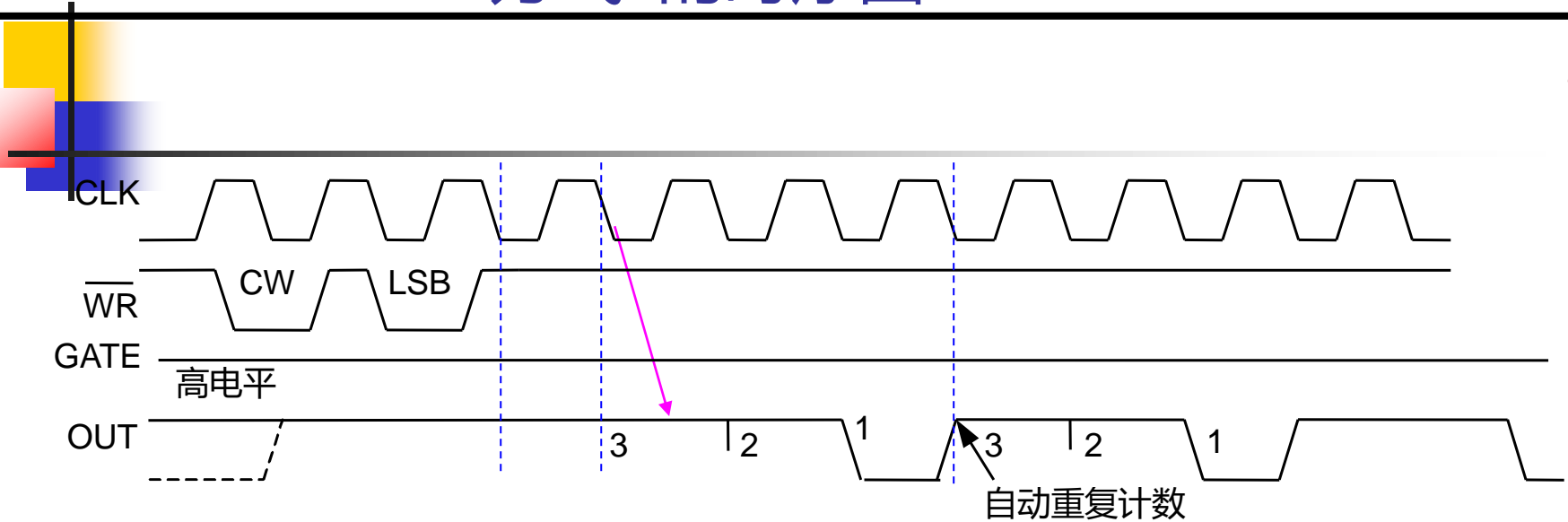
(3) 方式2 —频率发生器 (分频器)

计数器既可以软件启动，也可以硬件启动。

写入控制字后，计数器输出端为高电平。写入计数初值后，只要GATE为高电平(电平触发)计数器立即对CLK端的输入脉冲计数。

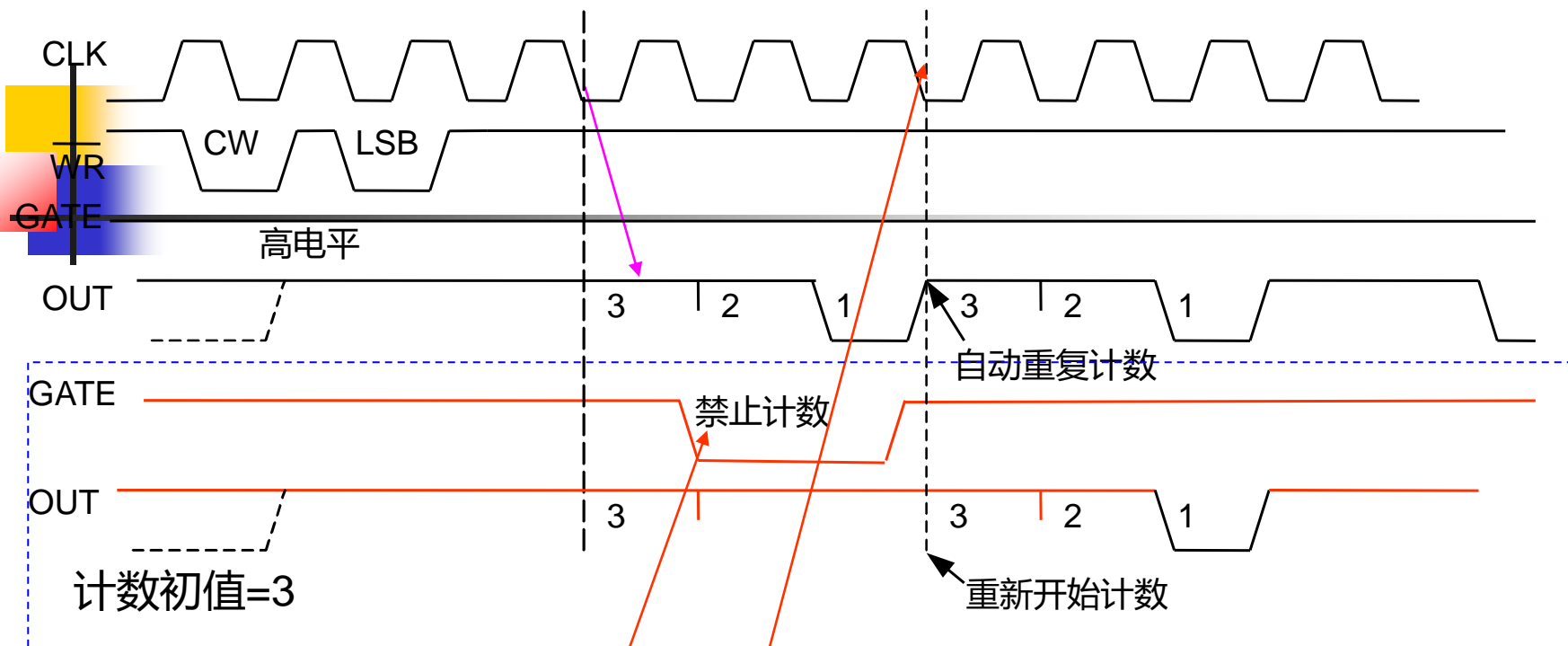
在计数过程中输出端始终保持为高，直到计数器减为1时，输出变低。经过一个CLK周期，输出恢复为高，同时按照原计数初值重新开始计数。如果计数值为N，则在CLK端每输入N-1个脉冲后，OUT就输出一个脉冲。因此，这种方式可以作为分频器或用于产生实时时钟中断。

方式2的时序图



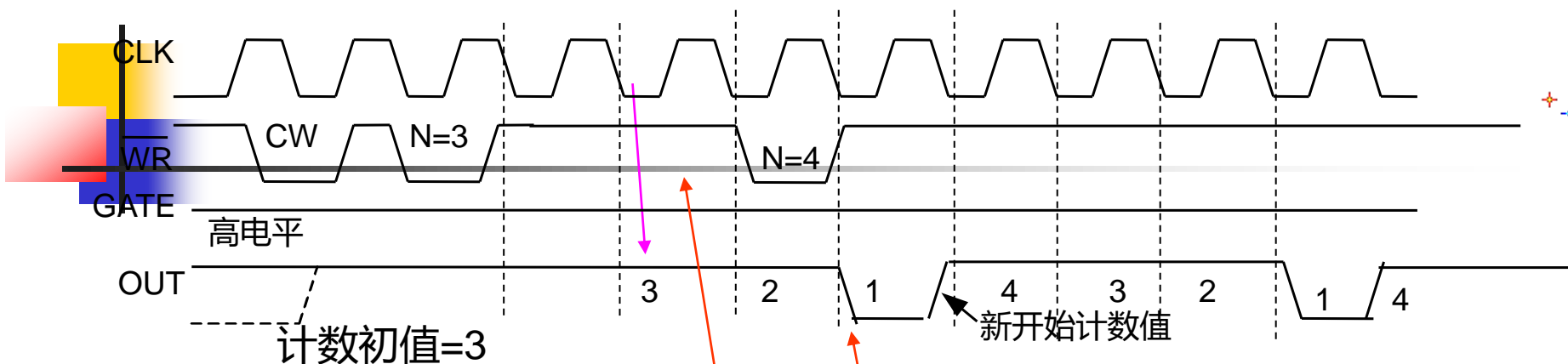
方式2的特点①:

不用重新设置计数初值，计数到0后自动装入初值,计数器能够连续工作，输出固定频率的脉冲。



方式2的特点②:

计数过程可由门控信号GATE控制。当GATE为0时，暂停计数。当GATE变为1后，下一个CLK脉冲使计数器恢复初值，重新开始计数。



方式2的特点③:

在计数过程中可以改变计数初值，这对正在进行的计数过程没有影响。但当计数到1时输出变低，过一个CLK周期输出又变高，计数器将按新的计数值计数。所以对方式2改变计数初值时，在下一次计数有效。

例：设8253计数器0工作于方式2，按二进制计数，计数初值为0304H。

MOV AL, 00110100B ; 先读/写低8位；再读写高8位，方式2，二进制。

OUT 43H, AL

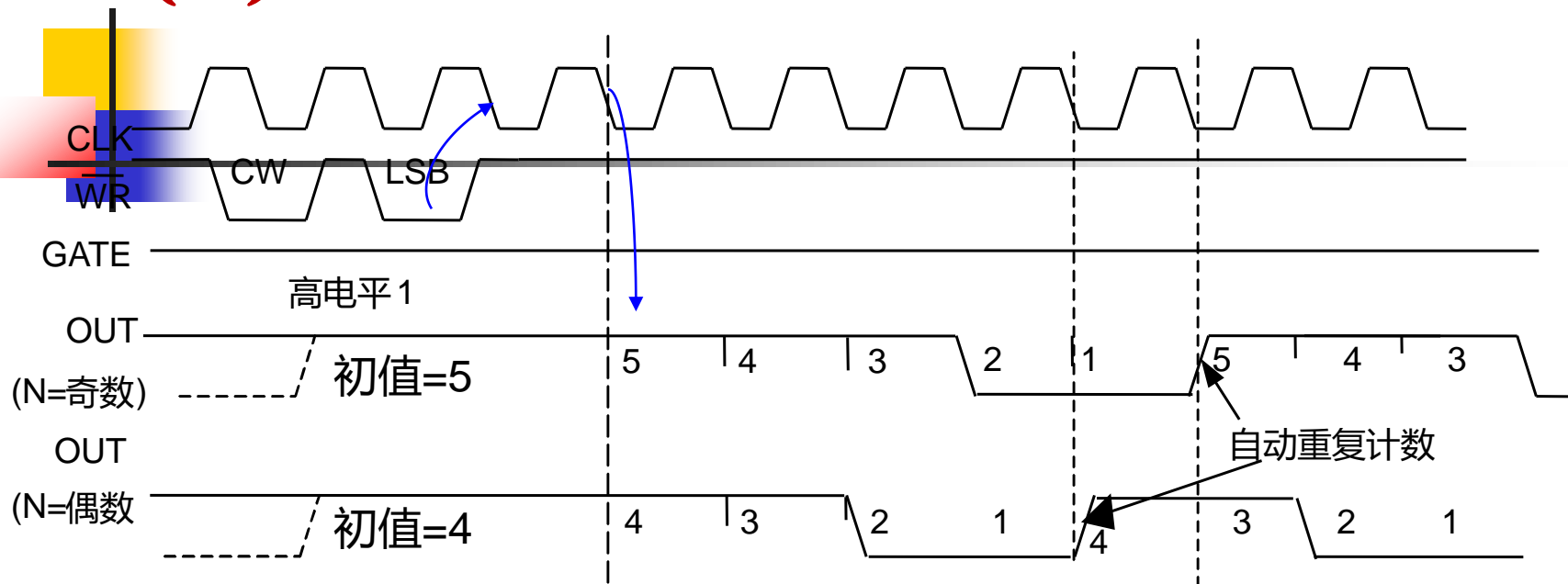
MOV AL, 04H ; 送计数值低字节

OUT 40H, AL

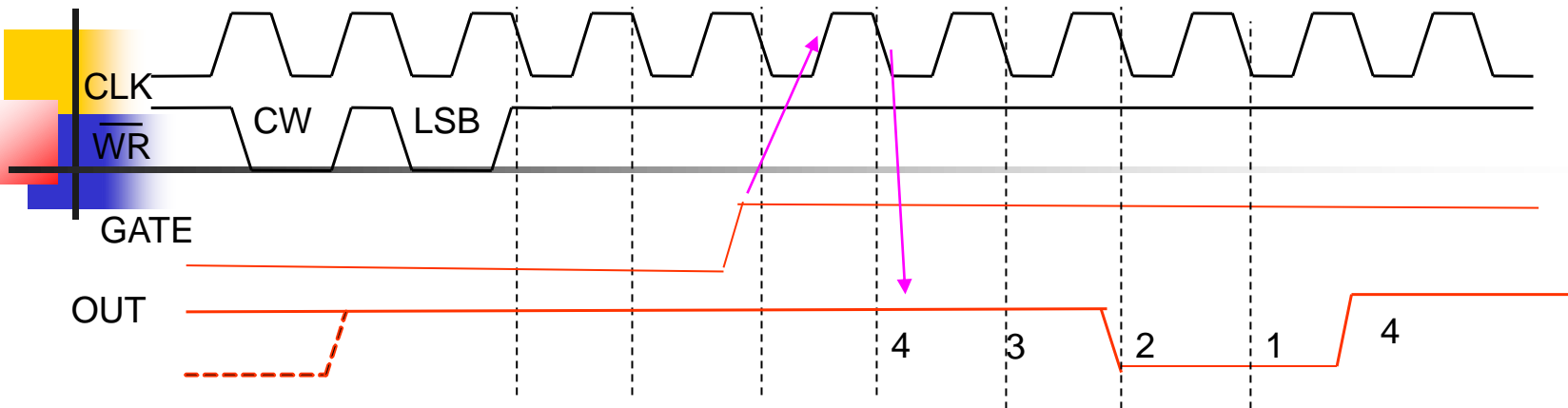
MOV AL, 03H ; 如只有低8位，不要这两条指令

OUT 40H, AL ; 送计数值高字节

(4) 方式3 一方波发生器

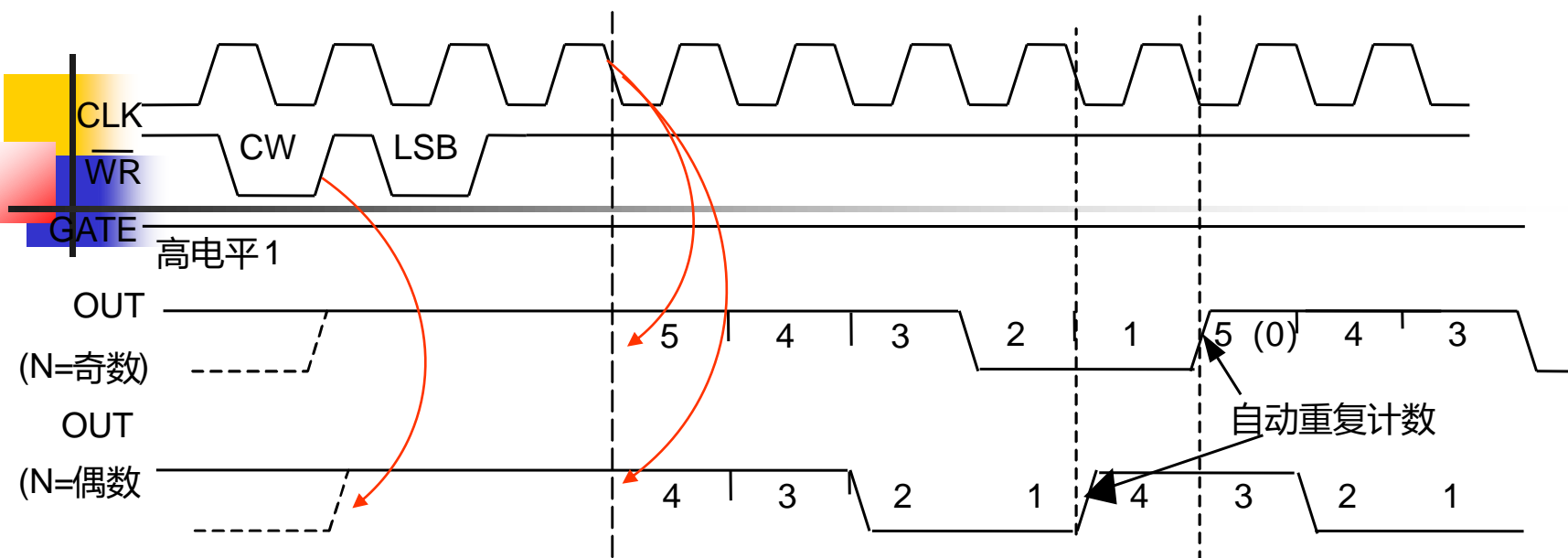


方式3的特点①： 可以软件启动或者硬件启动。写入控制字后，输出为高电平。GATE为高电平时(电平触发),在写入计数初值后就开始计数，输出保持为高。当计数到一半计数初值时，输出变为低，直至计数到0，输出又变为高，重新开始计数。



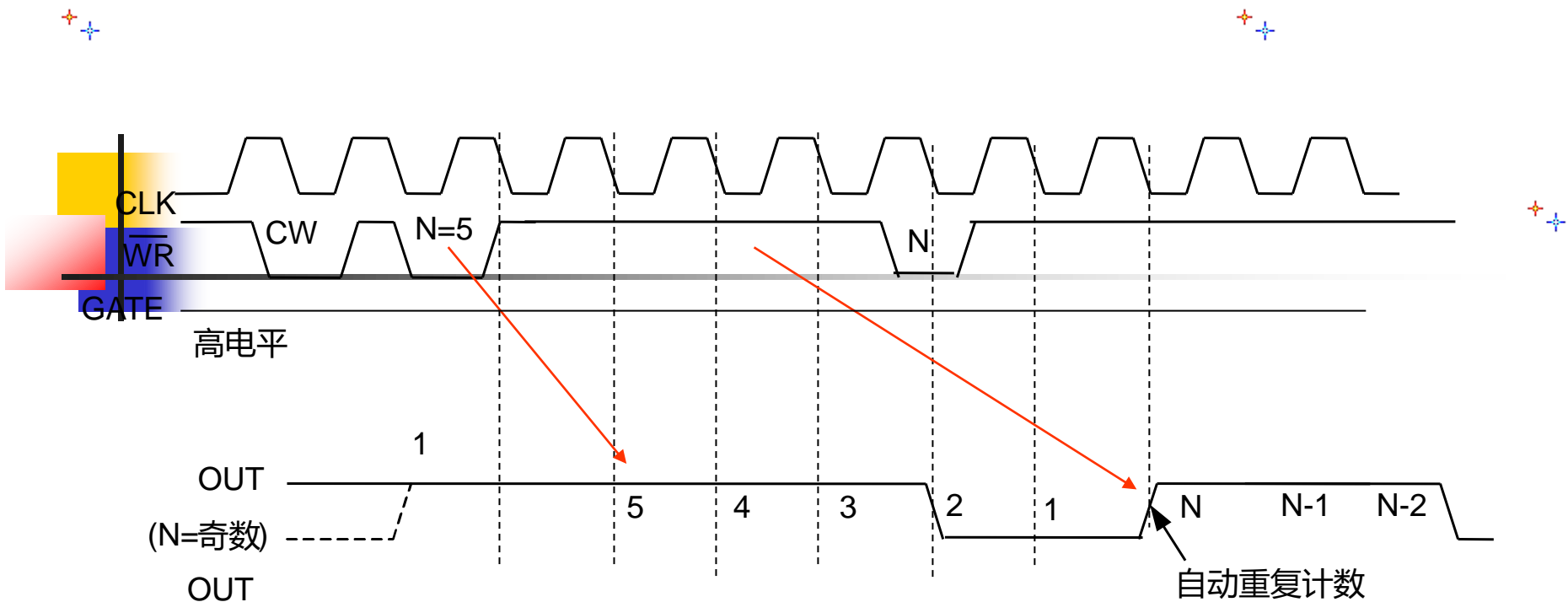
方式3的特点②:

如果写入初值时GATE为0，则写完初值后并不开始计数，而是等到GATE由低变高，才启动计数。



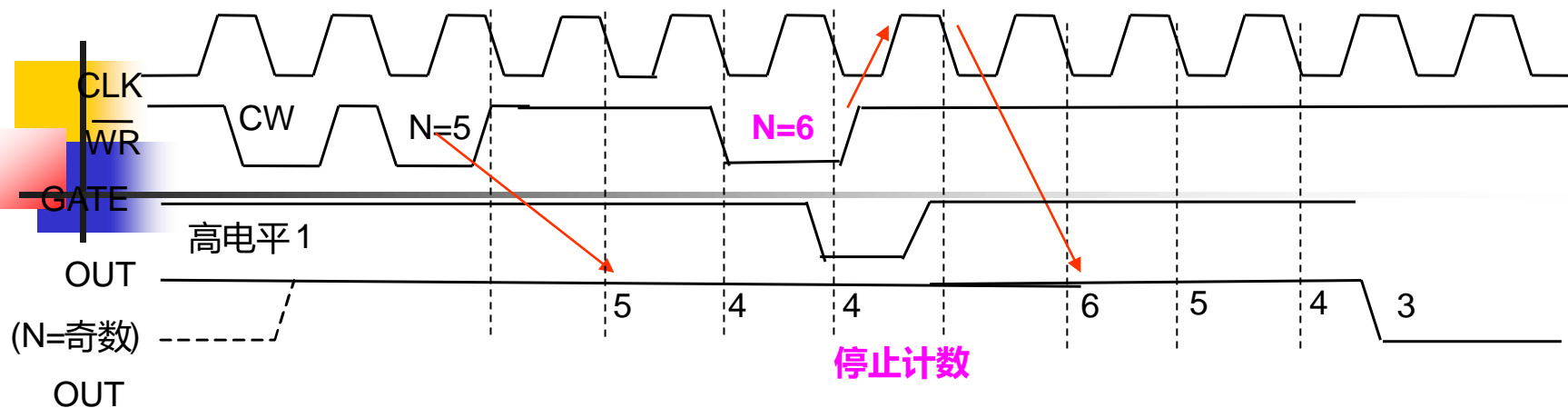
方式3的特点③：

如果GATE信号一直为高电平，在**写入控制字和计数值后**，将在下一个CLK脉冲来到时装入计数初值并开始计数。



方式3的特点⑤：

在计数期间写入一个新的计数初值，则要到现行计数周期结束后才按新的计数初值开始计数。（当前计数过程不受影响）



方式3的特点⑥：

GATE=1，允许计数；GATE=0，停止计数。当GATE变高以后，在下一个CLK脉冲来到时，计数器将重新装入初始值，开始计数。

例：设8253计数器2工作在方式3，按十进制计数，计数初值为4，则它的初始化程序如下：

MOV AL, 10010111B; 计数器2, 只读/写低8位, 工作方式3, 十进制

OUT 43H, AL ; 控制字送控制字寄存器

MOV AL, 4 ; 送计数初值

OUT 42H, AL

计数初值为1234，则它的初始化程序如下：

MOV AL, 10110111B ; 计数器2, 只读/写低8位, 工作方式3, 十进制

OUT 43H, AL ; 控制字送控制字寄存器

MOV AL, 34 ; 送计数初值低8位

MOV AX, 1234H ;

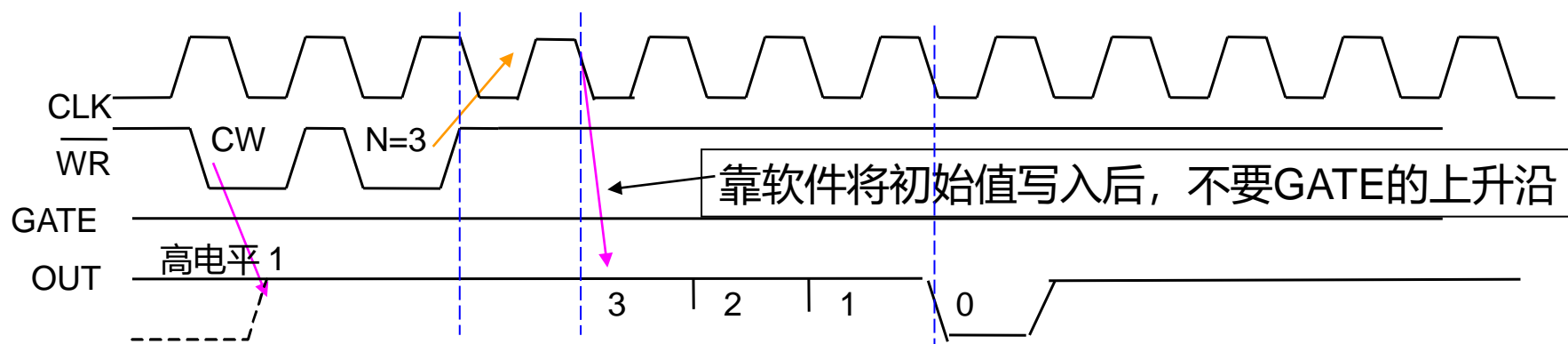
OUT 42H, AL

MOV AL, AH ; 送计数初值高8位

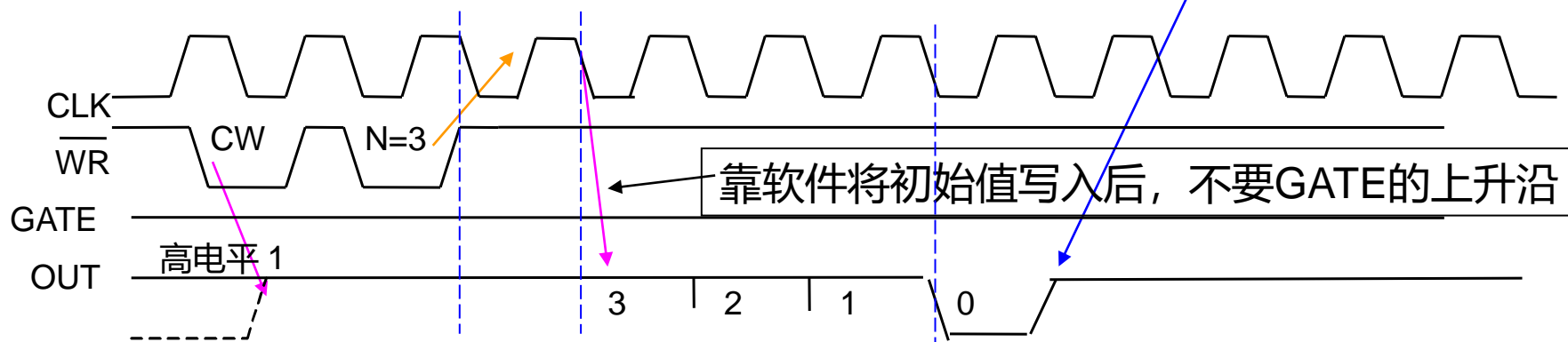
OUT 42H, AL

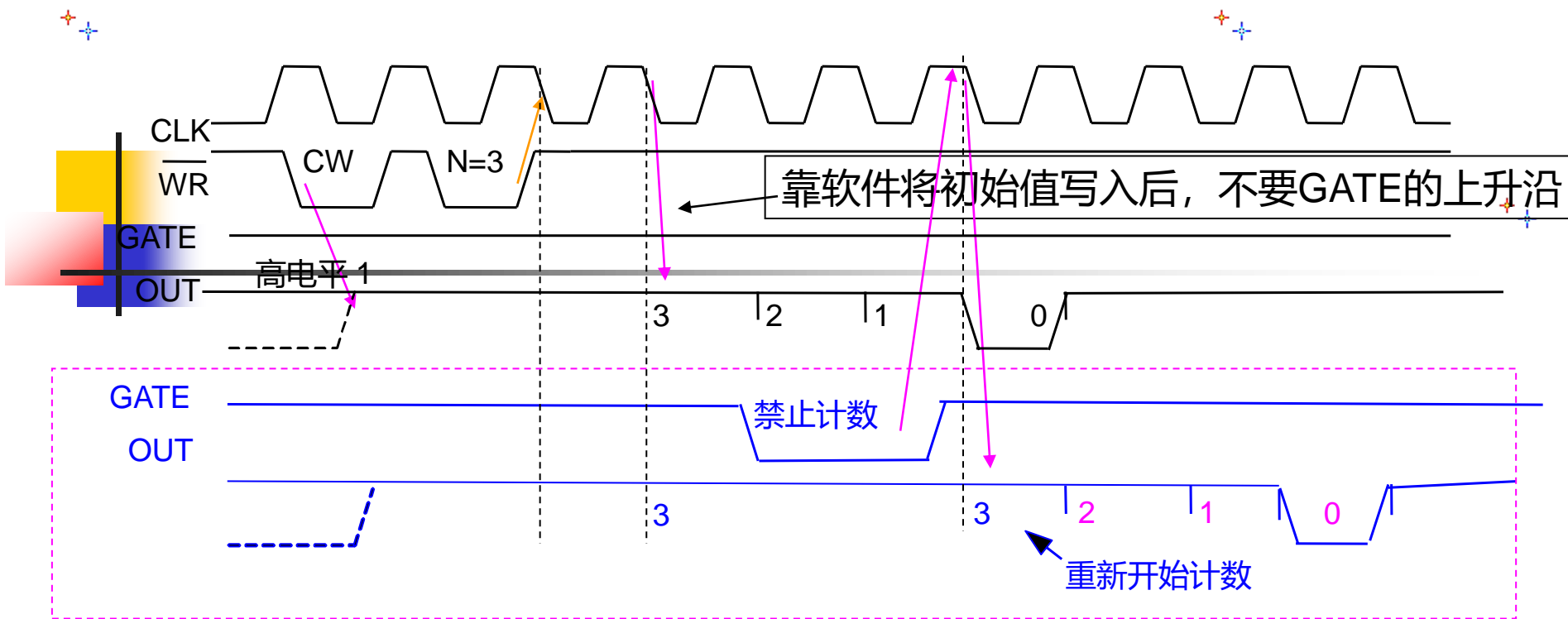
(5) 方式4 — 软件触发选通

工作工程：软件启动，写入控制字后，输出端OUT变为高电平，以此作为初始电平。当写入计数初值后开始计数，称为软件触发(前提条件是GATE为高电平)。



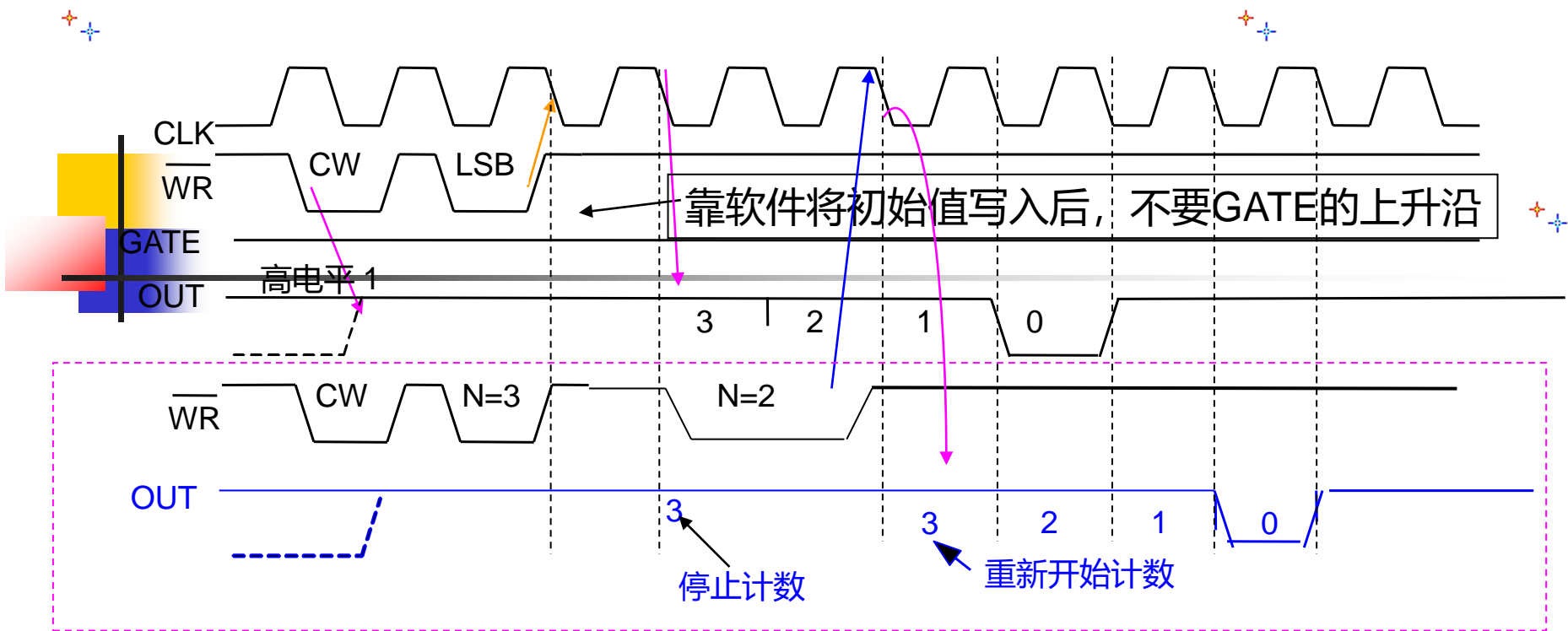
当计数到0后，输出变低，经过一个CLK周期，输出又变高，计数器停止计数。这种方式计数也是一次性的，只有在输入新的计数值后，才开始新的计数过程。若设置的计数初值为N，则在写入了计数初值后经过N个CLK脉冲，输出一个负脉冲。一般将此负脉冲作为选通信号。





方式4的特点①:

当GATE=1时, 允许计数; 当GATE=0时, 禁止计数。所以, 要做到软件触发, GATE应保持为1。在计数过程中如果GATE变为低, 则停止计数, 当GATE再变为高时, 重新开始计数。



方式4的特点②:

在计数过程中, 若改变计数初值, 则停止计数, 在下一个CLK的下降沿按新的计数初值开始计数。这称为软件再触发。16位数, 在设置第一个字节后停止计数, 在设置第二个字节后开始计数

例：设8253计数器1工作于方式4，按二进制计数，计数初值为3，则初始化程序为：

MOV AL, 01011000B ; 设置控制字寄存器

OUT 43H, AL ; 送控制字

MOV AL, 3; 置计数初值

OUT 41H, AL ; 送计数初值

计数初值为1234H，则初始化程序为：

MOV AL, 01111000B ; 设置控制字寄存器

OUT 43H, AL ; 送控制字

MOV AL, 34H ; 置计数初值低8位

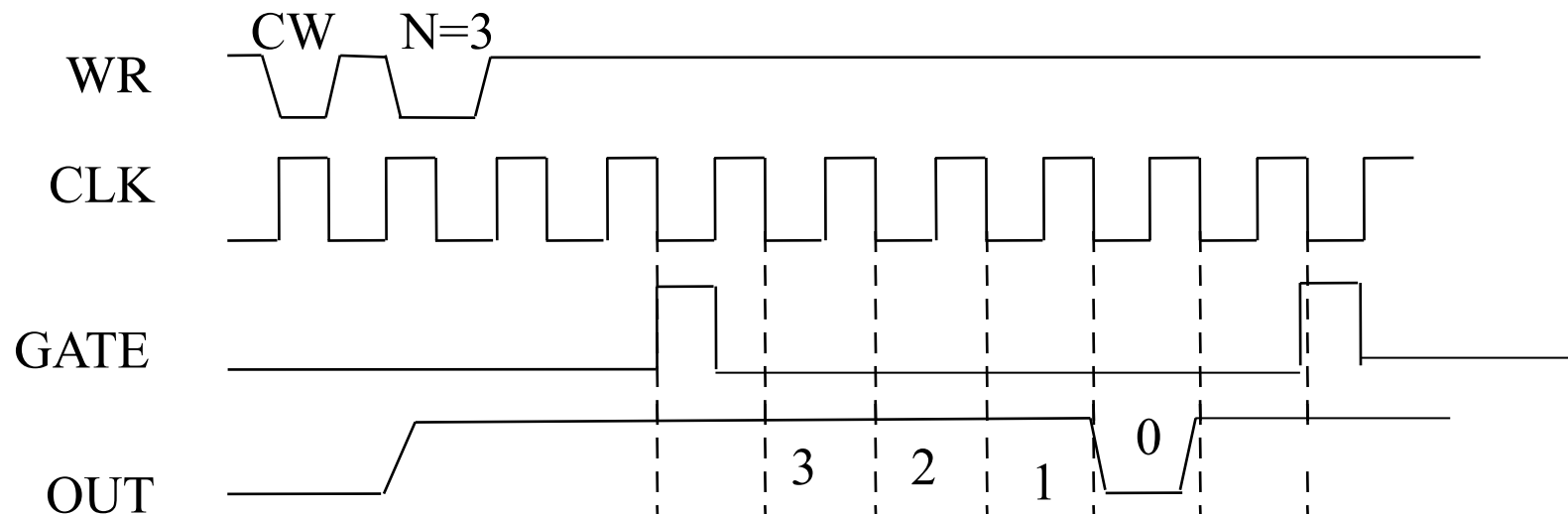
OUT 41H, AL ; 送计数初值

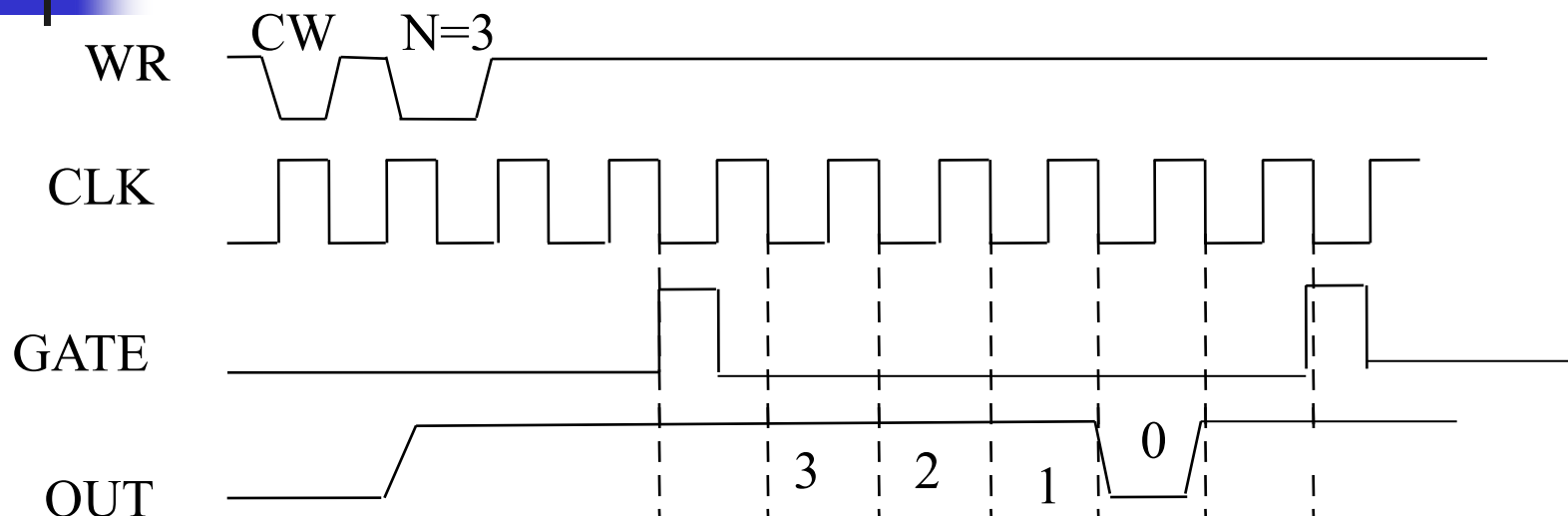
MOV AL, 12H ; 置计数初值高8位

OUT 41H, AL ; 送计数初值

(6) 方式5—硬件触发选通

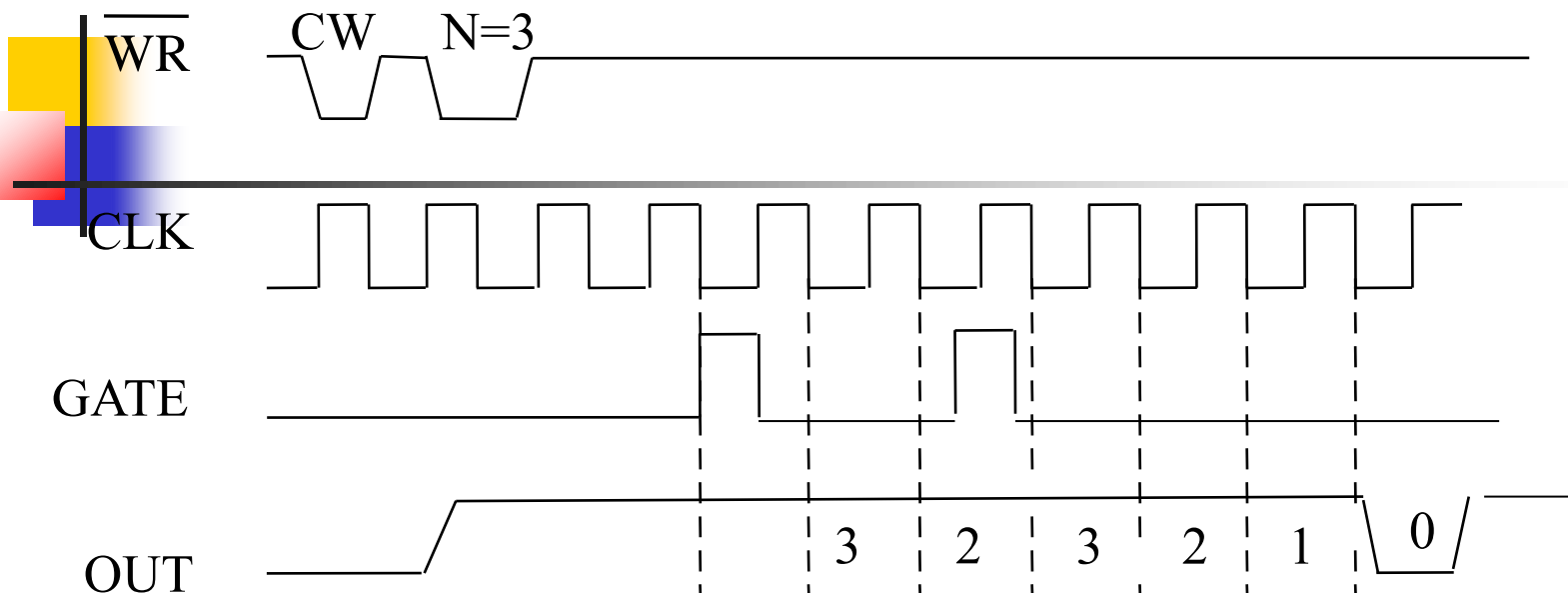
工作过程：硬件启动，写入控制字后，输出变为高电平，写入计数初值后，计数器并不立即开始计数，而是要由门控脉冲的上升沿来启动计数，称为硬件触发。当计数到0时，输出变低，又经过一个CLK脉冲，输出恢复为高。这样在输出端得到一个负脉冲选通信号。计数器停止计数后要等到下次门控脉冲触发，才能再进行计数。





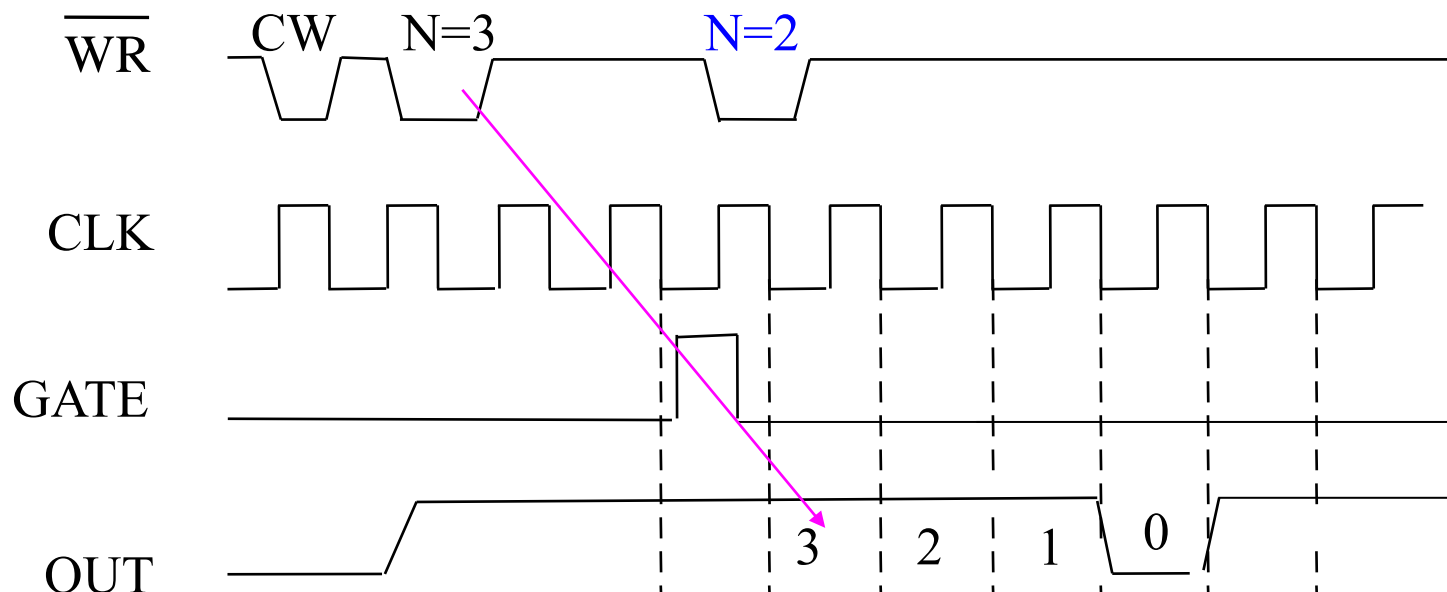
方式5的特点①：

若设置计数初值为N，则在门控脉冲触发后，经过N个CLK脉冲，才输出一个CLK周期的负脉冲。



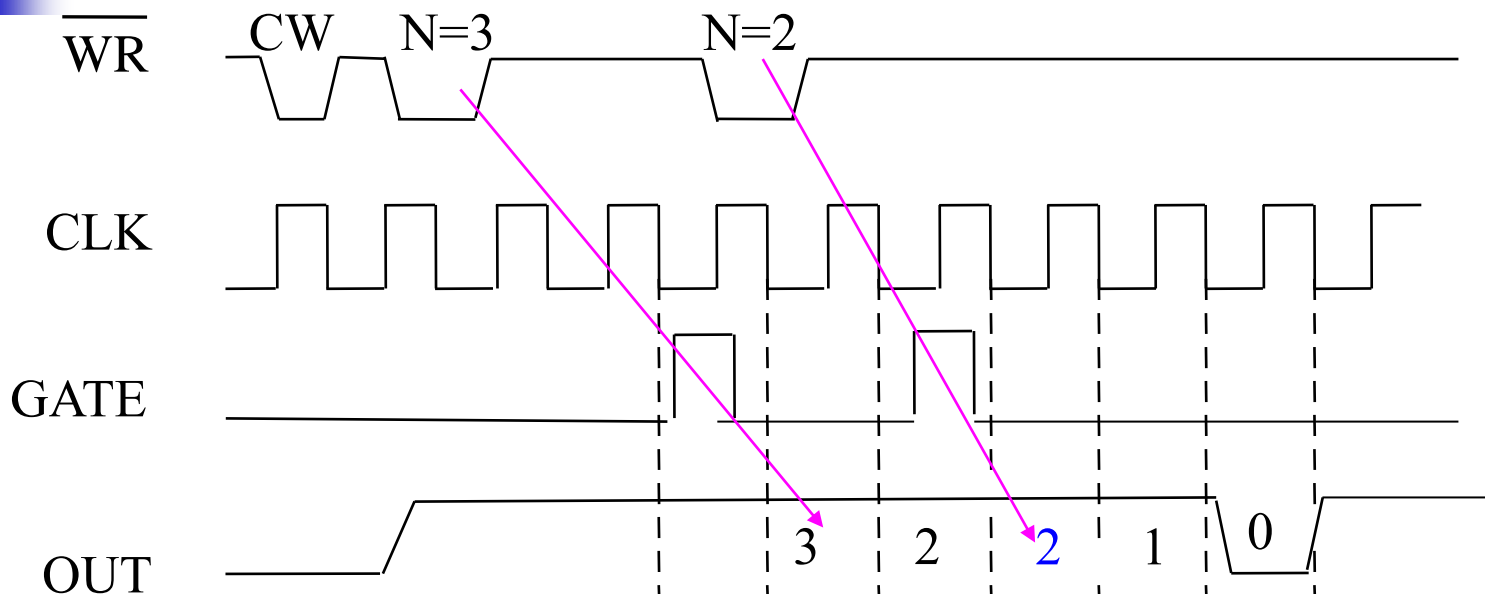
方式5的特点②:

在计数过程中，若GATE端又出现一个脉冲进行触发，则使计数器重新开始计数，但对输出状态没有影响。



方式5的特点③：

若在计数过程中改变计数值，只要没有门控信号的触发，就不影响本次计数过程。



方式5的特点③：

若在计数过程中改变计数值，若有新的门控信号的触发，则按新的计数初值计数。



方式5和方式4都产生选通脉冲

这两种方式的区别在于：

- 1) 方式4每次要靠软件设置计数初值后才能计数（软件触发）
- 2) 方式5的计数初值只需设置一次，但是每次计数要靠门控信号的触发（硬件触发）。
- 3) 方式4软件更改计数初值后立即起作用，方式5软件更改计数初值后要有新的门控信号的触发才能起作用。

例：设8253的通道1工作于方式5，按二进制计数，计数初值为4000H，则它的初始化程序段为：

MOV AL, 01101010B ; 通道1，只读写高字节，
OUT 43H, AL ; 方式5，二进制计数。

MOV AL, 40H ;

OUT 41H, AL ; 送计数初值

*MOV AL, 01111011B ; 通道1，先读写低字节，
OUT 43H, AL ; 后读写高字节，方式5，十进制数。*

MOV AL, 00H ; 低8位为0

OUT 41H, AL ;

MOV AL, 40H ; 高8位40，

OUT 41H, AL ; 送计数初值



7.1.6 六种工作方式的比较



(1) 方式0、4是由软件触发（写入计数值）启动计数，无自动装入计数初值的功能，除非再写入初值。GATE高时，减1计数器减1，GATE低时，减1计数器停止计数。

区别：方式0输出为低电平，计数结束输出高电平并一直保持。方式4输入输出为高电平，计数结束时输出一个时钟周期的负脉冲，然后又保持高电平。



(2) 方式1、5是硬件触发（GATE上升沿）启动计数，写入初值后并不马上计数，必须在GATE的上升沿触发后才开始计数。

区别：方式1输出 $N \cdot T_{CLK}$ 的负脉冲。

方式5结束后输出一个 T_{CLK} 的负脉冲。

(3)方式2、3都具有自动装入计数值的能力，都是分频器，但方式2输出 $N-1$ 占空比的波形，方式3则输出方波。

4、8253的读/写操作

8253的写操作包括写控制字和写计数初值两项内容。具体要求是：

① 各计数器的控制字都写到同一地址单元，而各计数初值写到各自的地址单元中。

② 对于每个计数器，必须先写控制字，后写计数初值。因为后者的格式是由前者决定的。

③ 写入的计数初值必须符合控制字决定的格式。16位数据应先写低8位，再写高8位。

当给多于一个的计数器写入控制字和计数初值时，其顺序无要求，只要遵循上述要求即可。



当对8253的计数器进行读操作时,可以读出计数值,具体实现方法有如下两种:

①使计数器停止计数时,先写入控制字,规定好RL1和RL0(控制字的D5D4位)的状态——也就是规定读一个字节还是读两个字节。

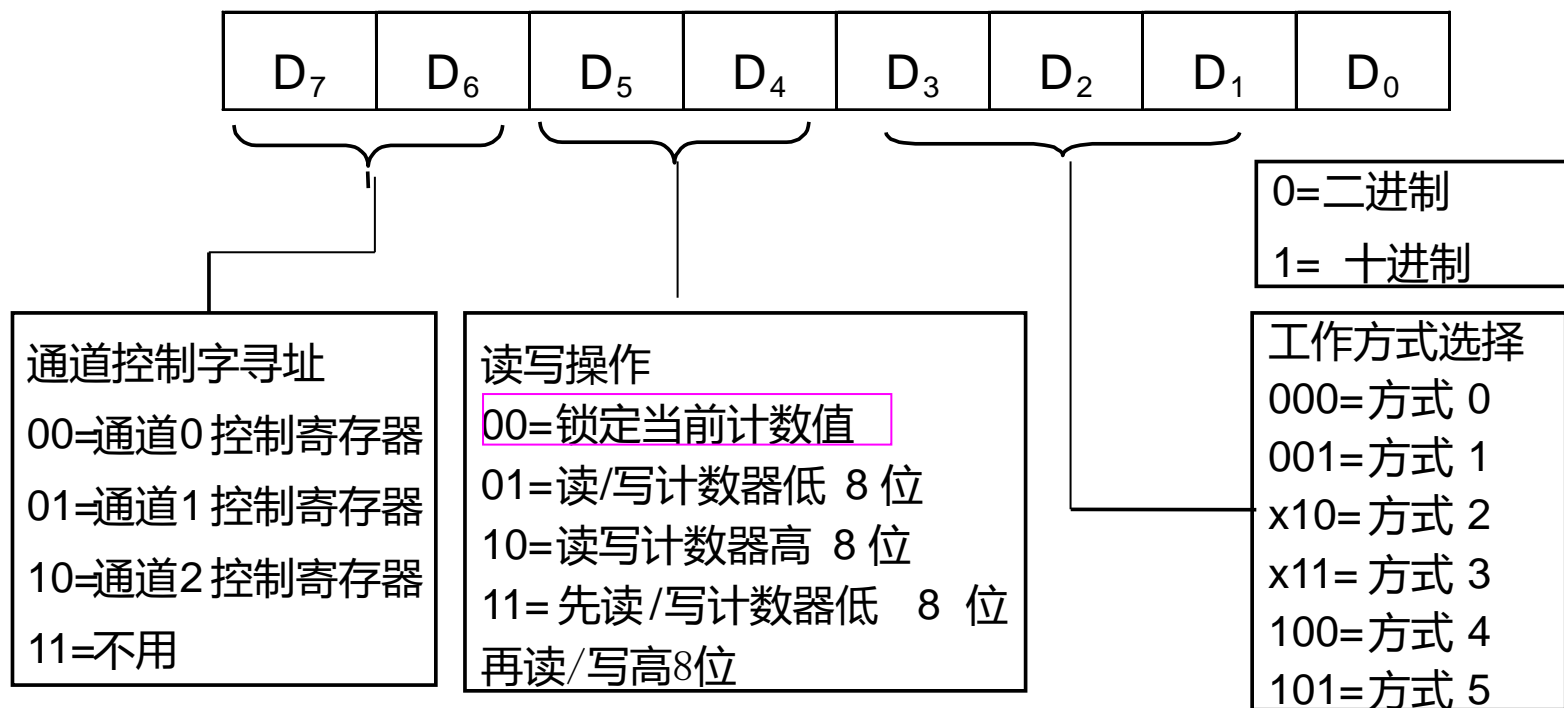
②在计数过程中读计数值。这时读出当前的计数值并不影响计数器的工作。为做到这一点,首先写入8253一个特定的控制字:XX00XXXX。这是控制字的一种形式。



③ 当计数器为16位时，为了避免在两次读出过程中计数值的变化，要求先将计数值锁存。

锁存计数值的常用方法是使用计数器锁存命令：

控制字的D7D6两位为所要锁存的计数器号，D5D4两位置为00。8253的每个计数器都有一个输出锁存器（16位），平时它的值跟随计数值而变化。当向计数器写入锁存命令后，现行计数值被锁存（如不锁存计数器值如正在跳变，读出的值不确定）。这样CPU读取的就是锁存器中的值。当CPU读取了计数值或对计数器重新编程以后，锁存状态被解除，输出锁存器的值又随计数值变化。



读计数器的值时，控制字的D₇D₆两位为所要锁存的计数器号，D₅D₄两位置为00，其它位不用考虑。



读计数值的程序设计：

MOV AL, 40 ; 通道1锁存, 16位数据

OUT 43H, AL ;

IN AL, 41H ; 读通道1, 低8位

MOV CL, AL ; 存在CL中

IN AL, 41H ; 读通道1, 高8位

MOV CH, AL ; 存在CH中

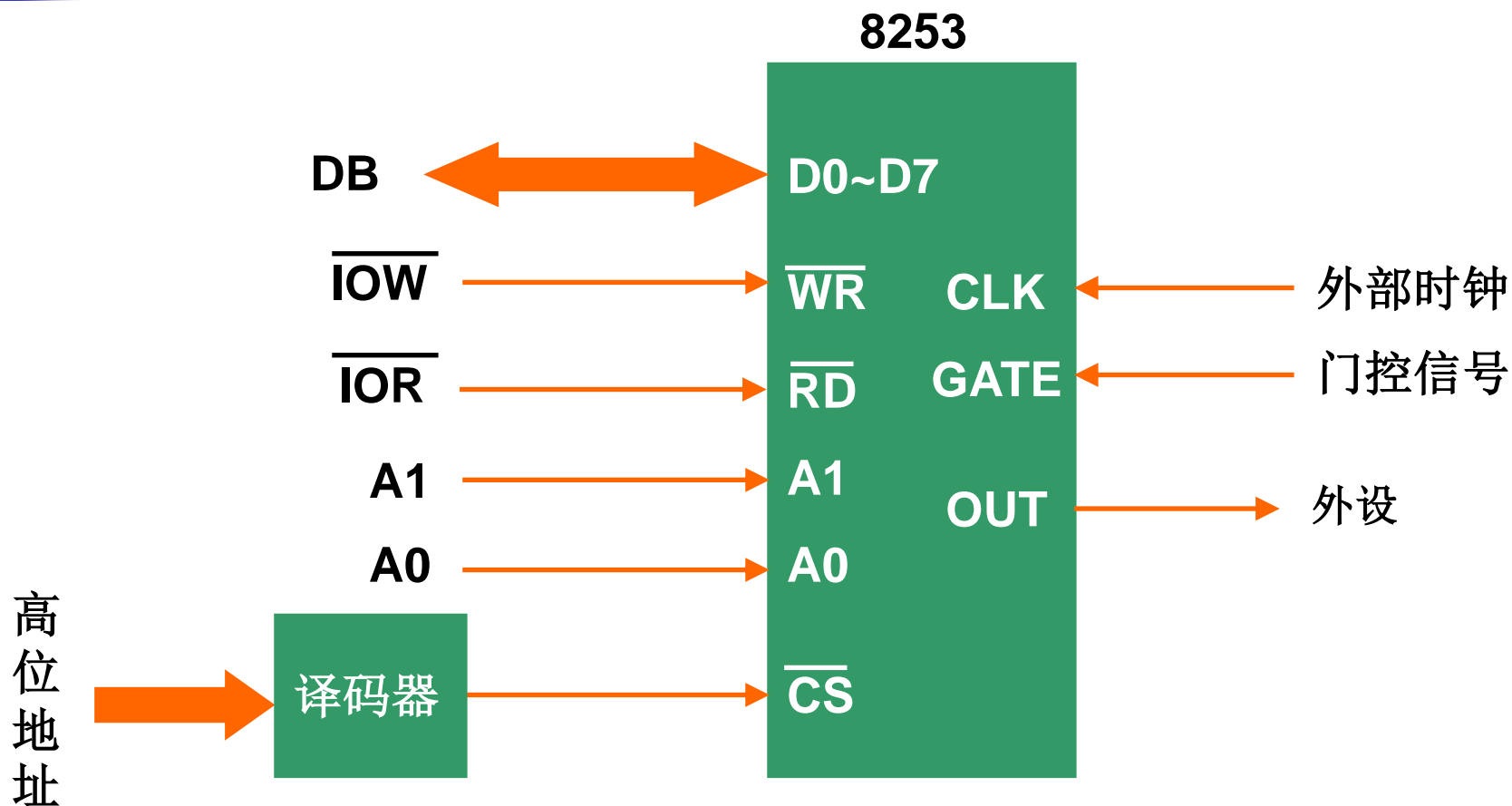
注意：读入数据的程序设计过程，先要锁存，然后才开始读入。



7.1.7 8253的应用

- 与系统的连接
- 初始化程序设计
 - 写入控制字
 - 置计数初值

与系统的连接示意





应用中的注意点

- 每一次启动计数，需有两次写操作：
 - 写控制字
 - 写计数器初值
 - 如果初值为8位字长，则一次写入；若初值为16位字长，则需两次写入
- 每个计数器的控制命令字均送入控制寄存器
- 各计数器的计数初值送到该计数器的计数寄存器及初值寄存器；

初始化程序流程

写控制字

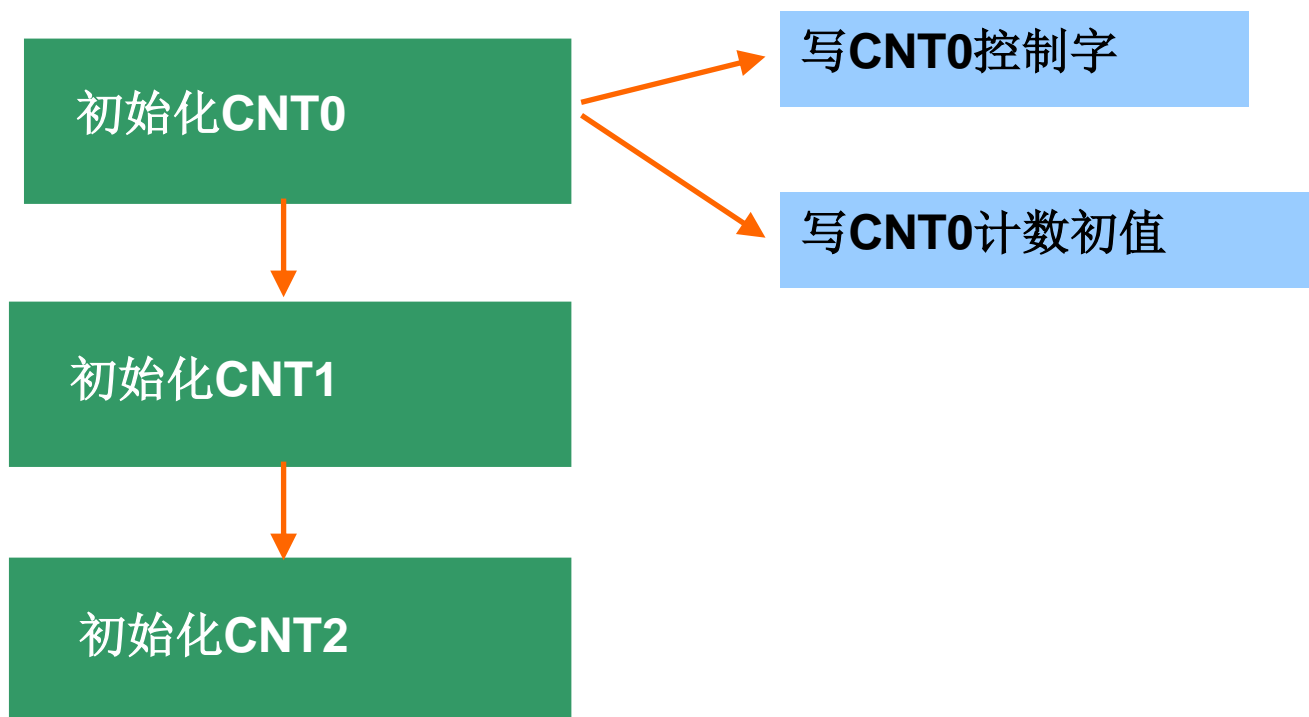
写计数值低8位

* 写计数值高8位

非必须

两个以上计数器初始化程序流程

■ 第一种初始化方式



■ 第二种初始化方式

写入全部计数器控制字



写CNT0计数初值



写CNT1计数初值



写CNT2计数初值

原则:

- ◆ 先写入控制字
- ◆ 后写入计数初值

8253应用例1

- 采用8253作定时/计数器，其接口地址为0120H~0123H。
- 输入8253的时钟频率为2Mhz。要求：
 - CNT0每10ms输出一个CLK周期宽的连续负脉冲
 - CNT1输出10KHz的连续方波信号
 - CNT2在定时5ms后产生输出高电平
- 画线路连接图，并编写初始化程序。

启动方式

工作方式

计数脉冲频率

计数初值

例1解

■ 计算计数初值:

CNT0: $10\text{ms}/0.5\mu\text{s}=20000$

CNT1: $2\text{ MHz}/10\text{KHz}=200$

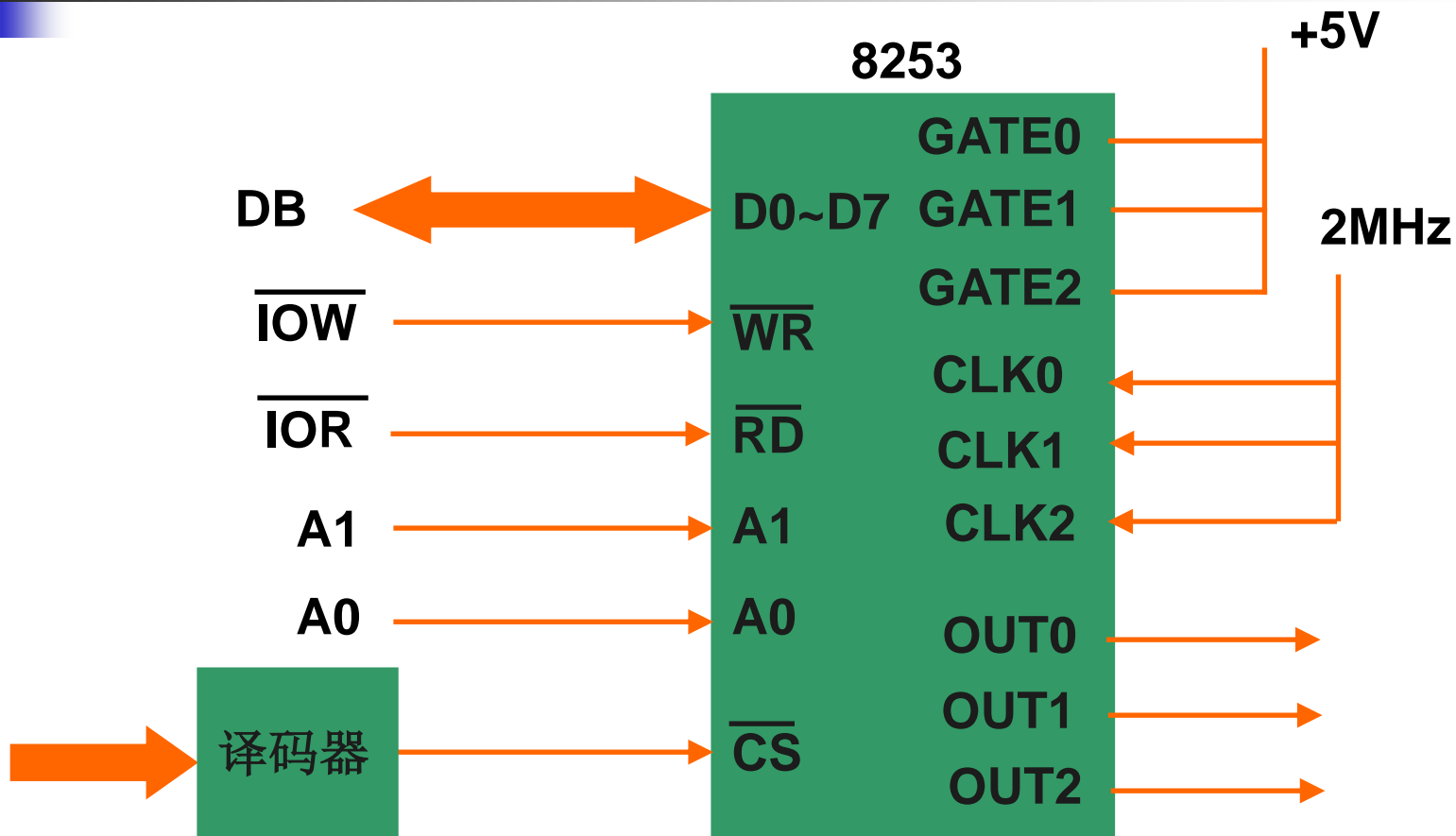
CNT2: $5\text{ms}/0.5\mu\text{s}=10000$

■ 确定控制字:

CNT0: 方式2, 16位计数值 → 00110100

CNT1: 方式3, 低8位计数值 → 01010110

CNT2: 方式0, 16位计数值 → 10110000





初始化程序

CNT0:

MOV DX, 0123H

MOV AL, 34H

OUT DX, AL

MOV DX, 0120H

MOV AX, 20000

OUT DX, AL

MOV AL, AH

OUT DX, AL

CNT1:

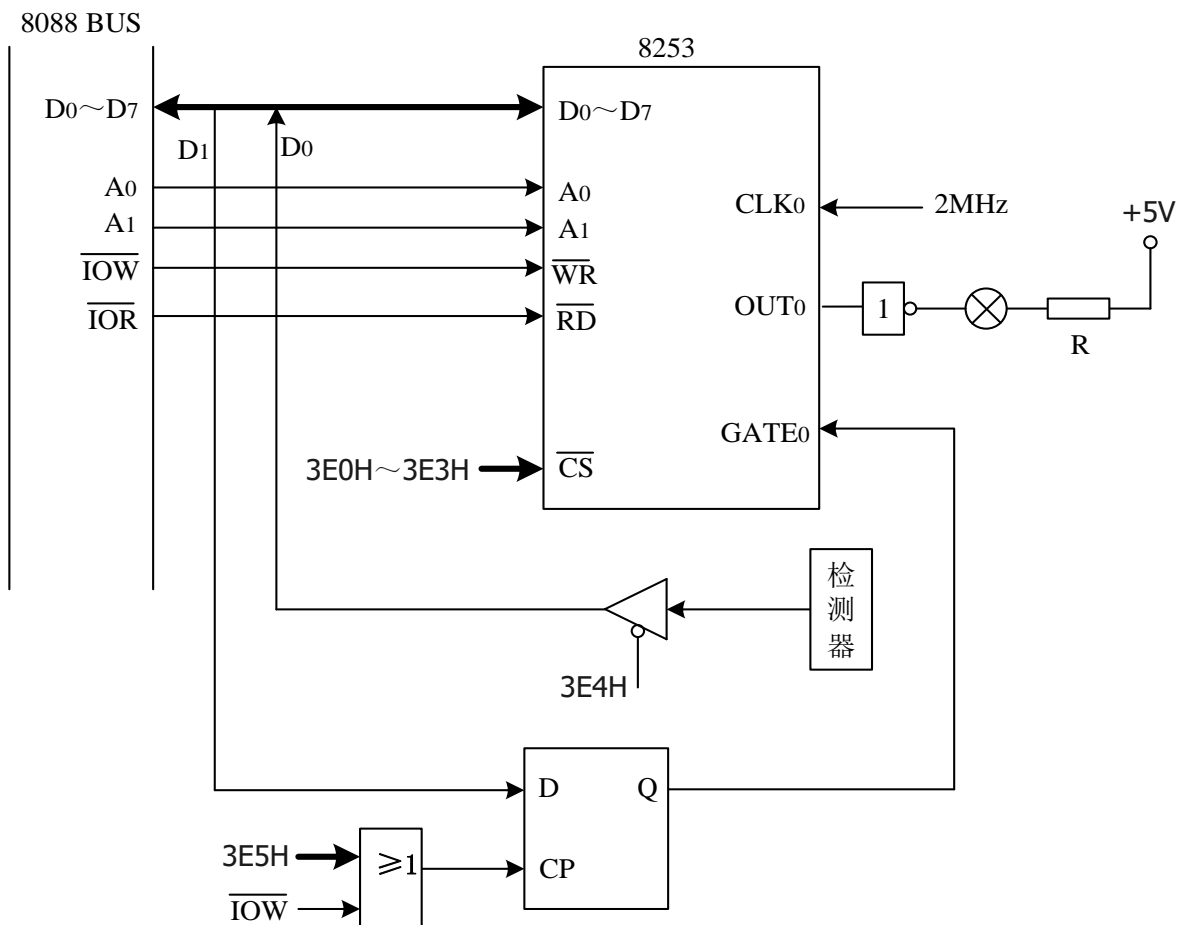
.....

CNT2:

.....

8253应用例2

- 安全检测 and 报警控制系统。





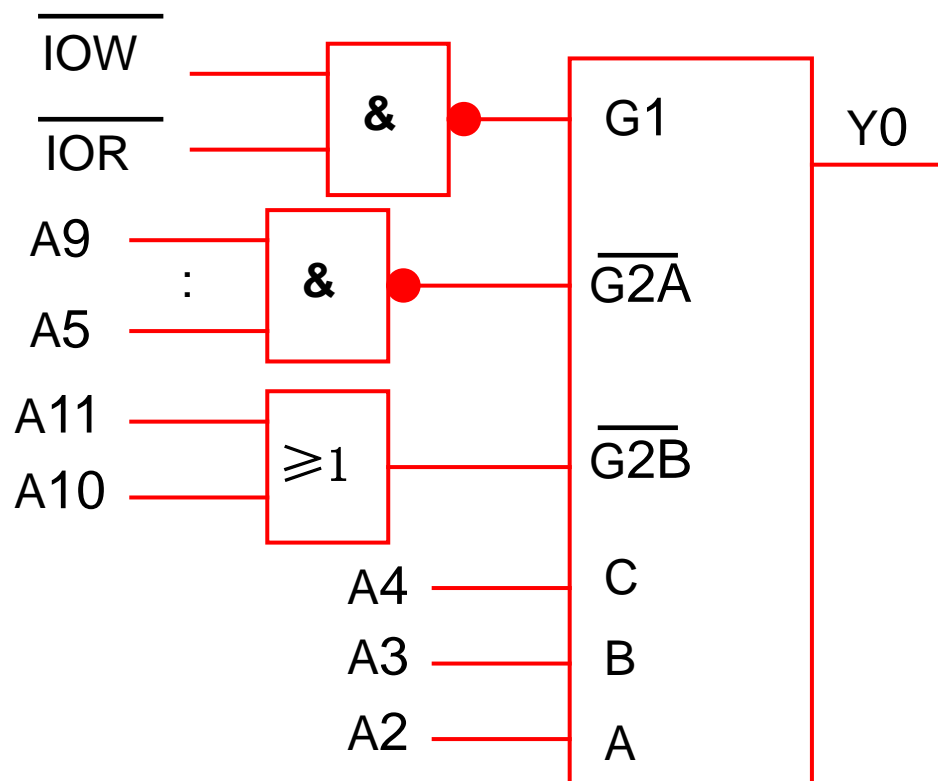
- 初始状态下，D触发器的Q端输出低电平；
- 系统通过三态门循环读取检测器状态，有异常出现时，检测器输出高电平。此时在D触发器的Q端输出高电平，启动8253计数器的通道0输出100Hz的连续方波信号，使报警灯闪烁，直到有任意键按下时停止；
- 使计数器停止输出方波的方法是在Q端输出低电平。
CLK0的输入脉冲为2MHz。要求：
 - 设计8253的译码电路；
 - 编写8253计数器的初始化程序及实现上述功能的控制程序。

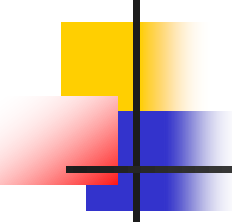
例2解

■ 8253地址范围:

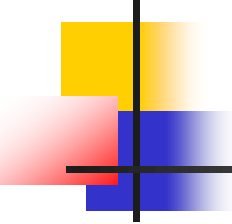
■ 0011 1110 0000 ~ 0011 1110 0011

■ 译码电路:



- 
- **8253计数初值:**
 - **2MHz/100Hz=20000**
 - **8253工作方式:**
 - **方式3**

- 
- **8253初始化程序:**
 - **MOV DX, 3E3H**
 - **MOV AL,**
00110110
 - **OUT DX, AL**
 - **MOV DX, 3E0H**
 - **MOV AX, 20000**
 - **OUT DX, AL**
 - **MOV AL, AH**
 - **OUT DX, AL**

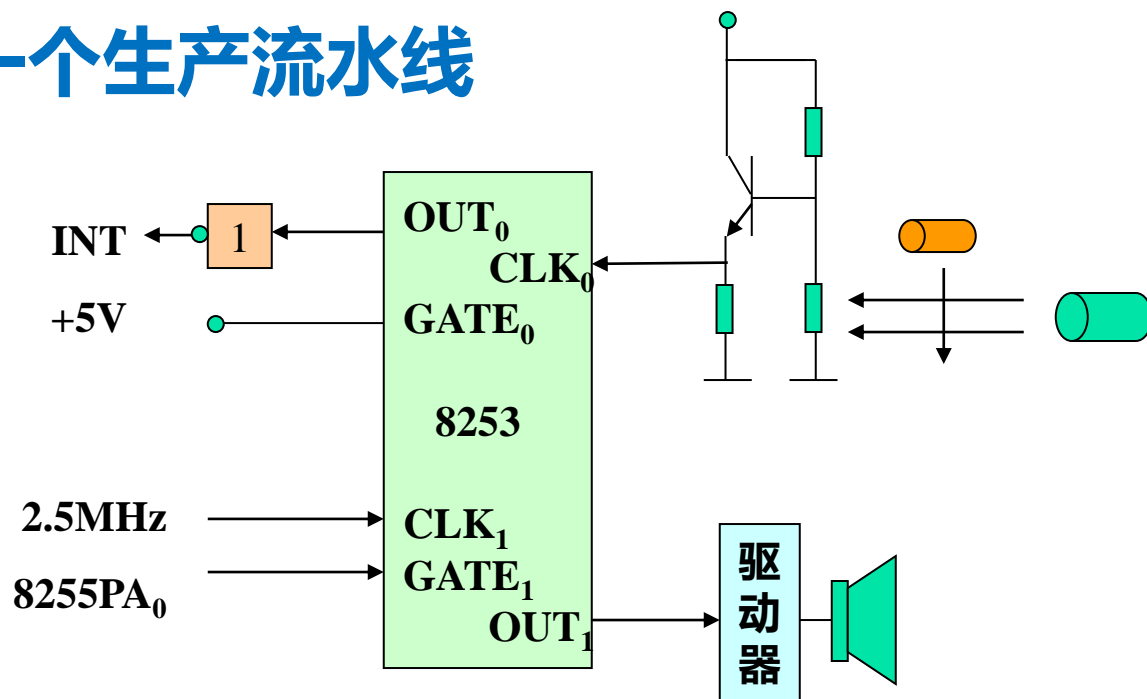


```
XOR AL, AL
MOV DX, 3E5H
OUT DX, AL
MOV DX, 3E4H
NEXT: IN AL, DX
AND AL, 01H
JZ NEXT
MOV DX, 3E5H
```

```
MOV AL, 2; D1 (D触发器)
OUT DX, AL
GOON: MOV AH, 1; 键盘
INT 16H
JZ GOON
XOR AL, AL
OUT DX, AL
MOV AH, 4C
INT 21H
```

8253应用例3

■ 监视一个生产流水线



8253的应用—计数和定时



设计要求及设计方案确定



- 使用8253监视一个生产流水线，每通过50个工件，扬声器响5秒钟，频率2000Hz。

工作原理：

- (1) 用8253的通道0设定为计数方式计数，计数满50后，由OUT₀输出一个负脉冲，经反向后作为8259A的中断请求信号，在中断服务程序中，启动8253通道1工作，有通道1连续输出频率为2000Hz的方波信号，持续5秒后结束。
- (2) 通道0工作于方式2，通道1工作于方式3，通道1的门控信号GATE₁由8255A的PA0控制。

控制字设置

- 通道0计数器工作于方式2，采用BCD计数，因计数初值为50，采用 $RL_1RL_0=01$ （读/写计数器的低8位），则工作方式字为 $00010101=15H$ 。
- 通道1计数器工作于方式3， CLK_1 接2.5MHz时钟，要求产生2000Hz的方波，则计数初值应为 $2.5 \times 10^6 / 2000 = 1250$ ，采用 $RL_1RL_0=11$ （先读计数器的低8位，再读计数器的高8位）。则工作方式字为 $01110111=77H$ 。
- 设通道0的地址为40H，通道1的地址为41H，控制口地址为43H，8255A的口地址为80-83H。



主程序：

MOV AL, 15H; 通道0初始化, 方式2

OUT 43H, AL

MOV AL, 50H; 置计数初值

OUT 40H, AL

STI ; 开中断

LPO: HLT ; 等待中断

JMP LPO

中断服务程序为:

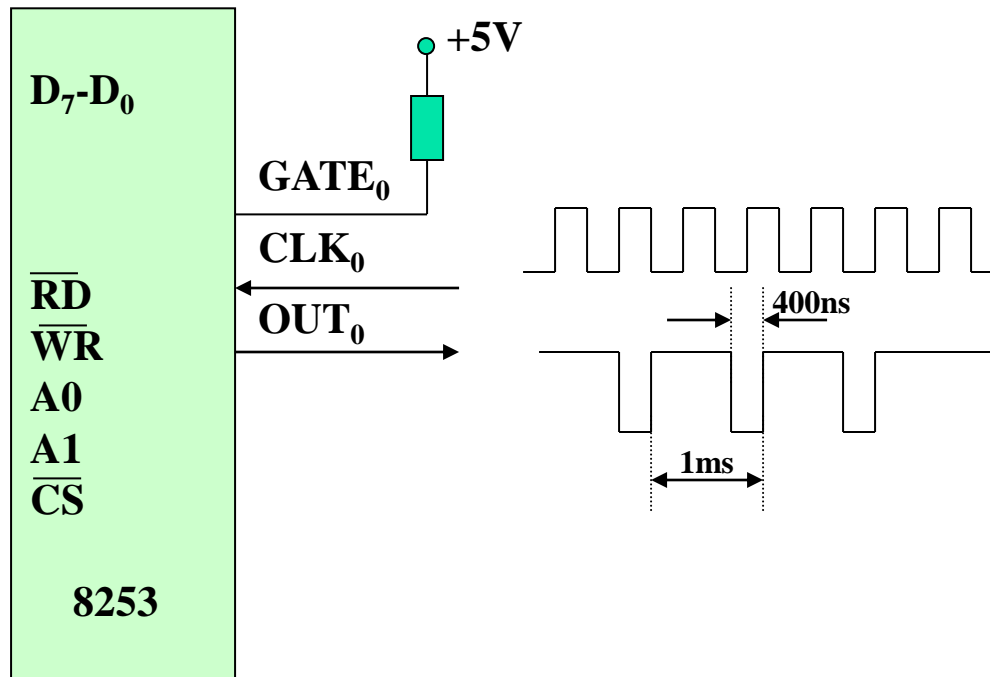
```
MOV AL, 01H ; 通道1的GATE1置1, 启动计数
OUT 80H, AL ; 认为8255已经初始化, 向8255
              ; 端口送数, 使PA0为 "1"
MOV AL, 77H ; 通道1初始化, 方式3
OUT 43H, AL
MOV AL, 50H ; 置计数初值低8位
OUT 41H, AL
MOV AL, 12H ; 置计数初值高8位
OUT 41H, AL
CALL DL5S   ; 延时5秒, DL5S为5秒延时程序
MOV AL, 00H ; 通道1的GATE1置0, 停止计数
OUT 80H, AL ; 复位8255端口, 使PA0为 "0"
IRET
```

注: 本例中, 通道0工作于计数状态, 通道1工作于计时状态。

8253应用例4

计数通道0连接如图所示, 试回答:

- (1) 计数通道0工作于何种方式, 并写出工作方式名称;
- (2) 写出计数通道0的计数初值 (列出计算式)
- (3) 写出初始化程序



解:

(1) 由于出现一个CLK周期宽度的负脉冲的方式有方式2\5两种,但是GATE0是一直保持高电平,因此方式5是无法触发的,因此可以判断计数通道0工作于方式2

(2) 因为CLK周期宽度为400ns, CLK正脉冲宽度为200ns, 因此计数通道0的计数初值= $1\text{ms}/400\text{ns}=2500$

(3) 初始化程序段(设8253的口地址为:80H-83H)

```
MOV AL, 0001 0101B
```

```
MOV 83H, AL
```

```
MOV AL, 00H
```

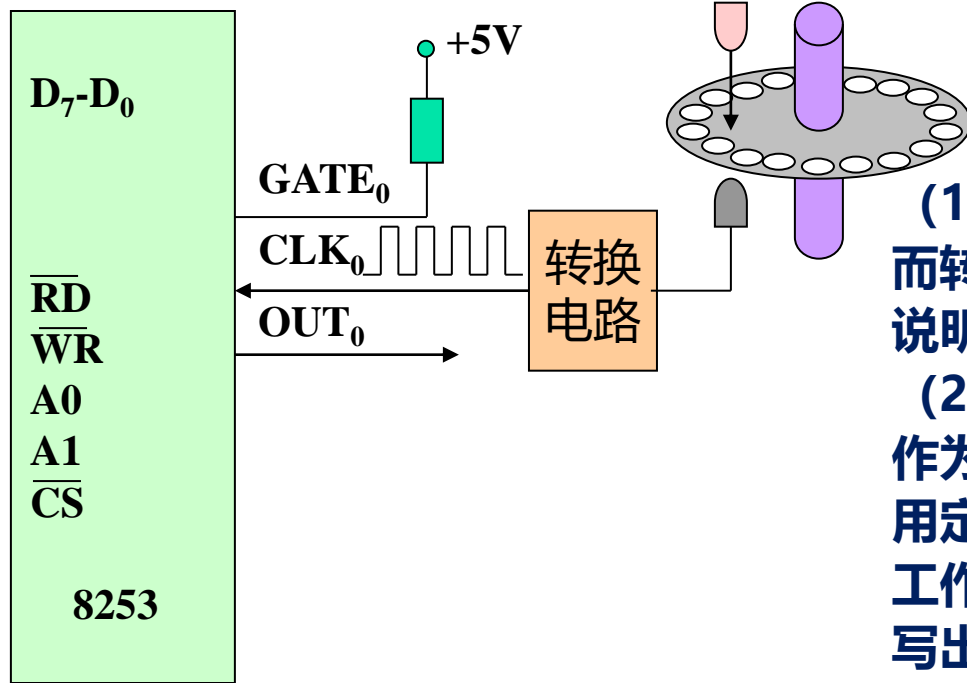
```
OUT 80H, AL
```

```
MOV AL, 25H
```

```
OUT 80H, AL
```

8253应用例5

若8253计数器对设备的转轴的旋转速度进行测试，接口电路如图所示。从图可知，若与转轴相连的转盘上均匀地钻有50个圆孔，当旋转时，通过光电转换，每通过一个小孔，产生一个正脉冲，当转轴转一圈，就会产生50个脉冲通过CLK输入8253计数器进行减法计数，若假设此转轴的速度范围在50-1000转/秒，并设8253的端口地址为84H—87H。



(1) 若采用定时测试已转换过的脉冲个数而转换为测试转轴的速度，单位为转/秒。说明它的计算过程。

(2) 若用计数器0对脉冲计数，用计数器1作为定时，设它的CLK1频率为200KHz，用定时100ms来计数。写出计数器0、1的工作方式控制字和计数初值，并注释，并且写出8253的初始化程序。

解:

因为最高转速为1000转/秒, 因此, 转一圈的脉冲数最高为
 $1000 \times 50 = 50000$ 个 = C350H, 因此计数器必须用16位计数, 并且用二进制计数。

(1) 采用定时测试已转速, 计算方法为:

是定时时间为: $\text{TIME} = 100\text{ms} = 0.1\text{s}$

则转速 = (定时到的计数值/50)/0.1s转/秒

(2) 初始化程序段:

■ 计数器0初始化程序段:

计数初值 = 0000H (或者50000)

MOV AL, 0011 0000B ;通道0,方式0,16位计数,二进制计数

OUT 87H, AL

MOV AL, 00

OUT 84H, AL

OUT 84H, AL



■ 计数器1初始化程序段:

计数初值 = $100\text{ms} / (1/200\text{kHz}) = 20000\text{D} = 4\text{E}20\text{H}$

MOV AL, 0111 0100B; 通道1, 方式2, 16计数, 二进制计数

OUT 87H, AL

MOV AL, 20H

OUT 85H, AL

MOV AL, 4EH

OUT 85H, AL



§ 7.2 可编程并行接口 8255



掌握：

- **主要引线功能及结构**
- **3种工作方式及其特点**
- **应用：**
 - **芯片与系统的连接**
 - **芯片的初始化编程**
 - **数据输入/输出控制程序设计**



7.2.1 并行通信和串行通信

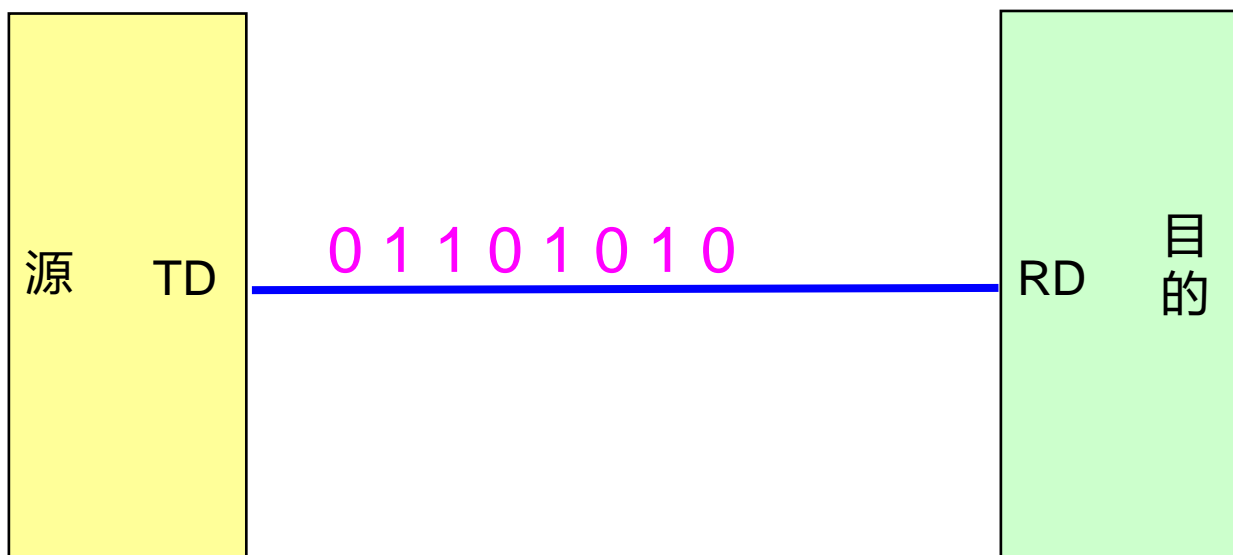


通信指计算机与外设、计算机与计算机间的信息交换

通信的基本方法: 并行通信和串行通信

串行通信

将数据的各位按时间顺序依次在一根传输线上传输。



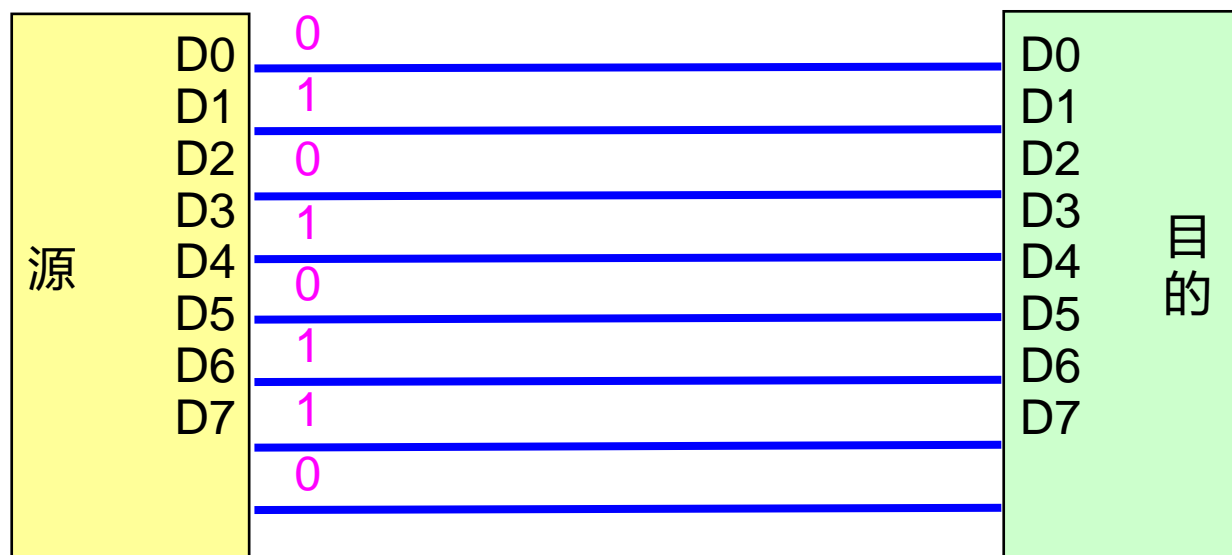
数据的各位依次由源到达目的地 → 慢

数据线少 → 远程, 费用低

串行通信适于长距离、中低速通信

并行通信

将数据的各位**同时**在**多根并行传输线**上进行传输。



数据的各位同时由源到达目的地 → 快

多根数据线 → 距离短、远程费用高

并行通信适于**短距离**、**高速**通信



并行接口芯片

不同的产品系列研制出与之相匹配的接口芯片。

例

在80X86系列中使用8255A芯片

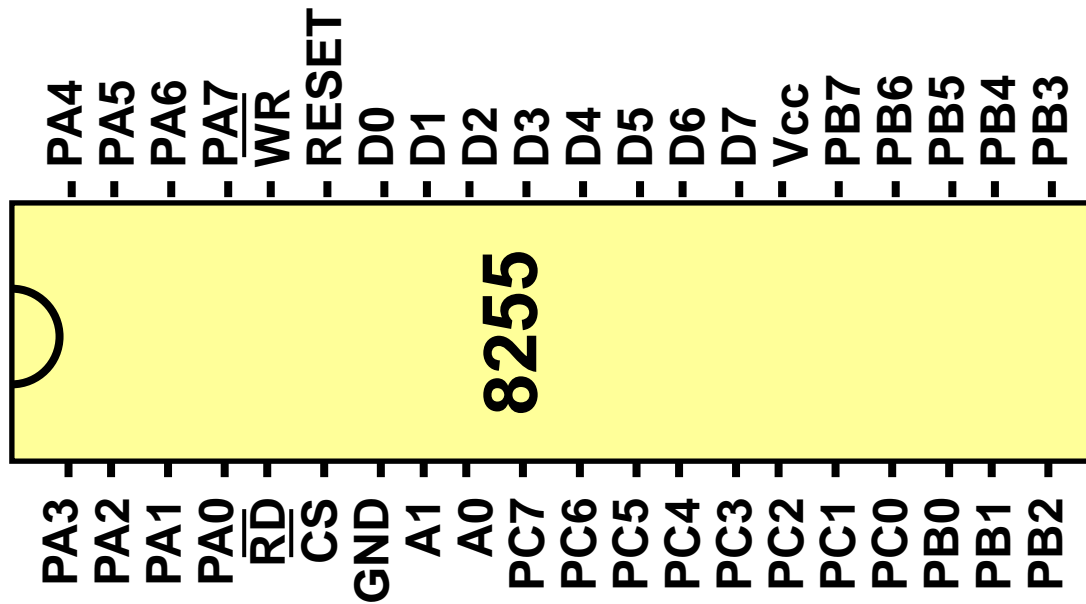
在Z80系列中使用Z80-PIO芯片

在6800系列中使用MC6820PIA芯片

一般情况下不交叉使用，因为：

只有本系列的芯片才能更好地保证时序上的配合，
和各种功能的发挥，使CPU可靠与外设交换信息。

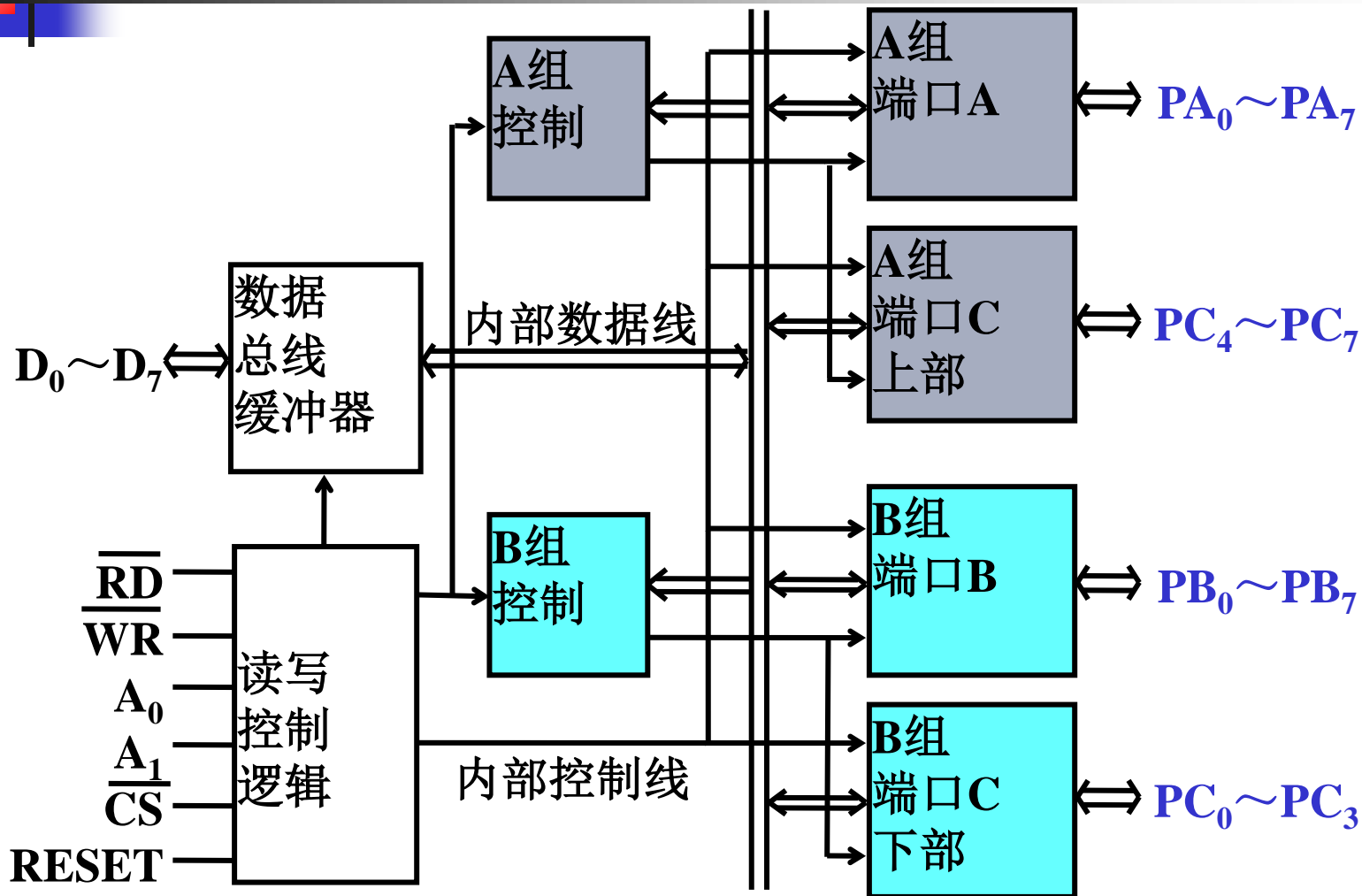
7.2.2 并行接口 8255



引脚图

8255是Intel系列的8位并行接口芯片通用性强，使用灵活，可用程序设置和改变芯片的工作方式，是一种常用的可编程并行接口芯片。

功能结构





端口说明

- **端口A: PA0 ~ PA7**
 - 支持工作方式0、1、2
 - 常作数据端口，输入输出都有锁存能力，功能最强大
- **端口B: PB0 ~ PB7**
 - 支持工作方式0、1
 - 常作数据端口，输入输出都有锁存能力。
- **端口C: PC0 ~ PC7**
 - 仅支持工作方式0
 - A组控制高4位PC4 ~ PC7；B组控制低4位PC0 ~ PC3
 - 可作数据、状态和控制端口。输出有锁存能力。
 - 分两个4位，每位可独立操作



■ 在8255A内部硬件结构中:

A口和B口之间没有硬件关系，即可分别作为独立的输入或输出端口；

C口和A口，C口和B口之间有硬件联系，通过向控制口设置控制字可以改变这种联系。

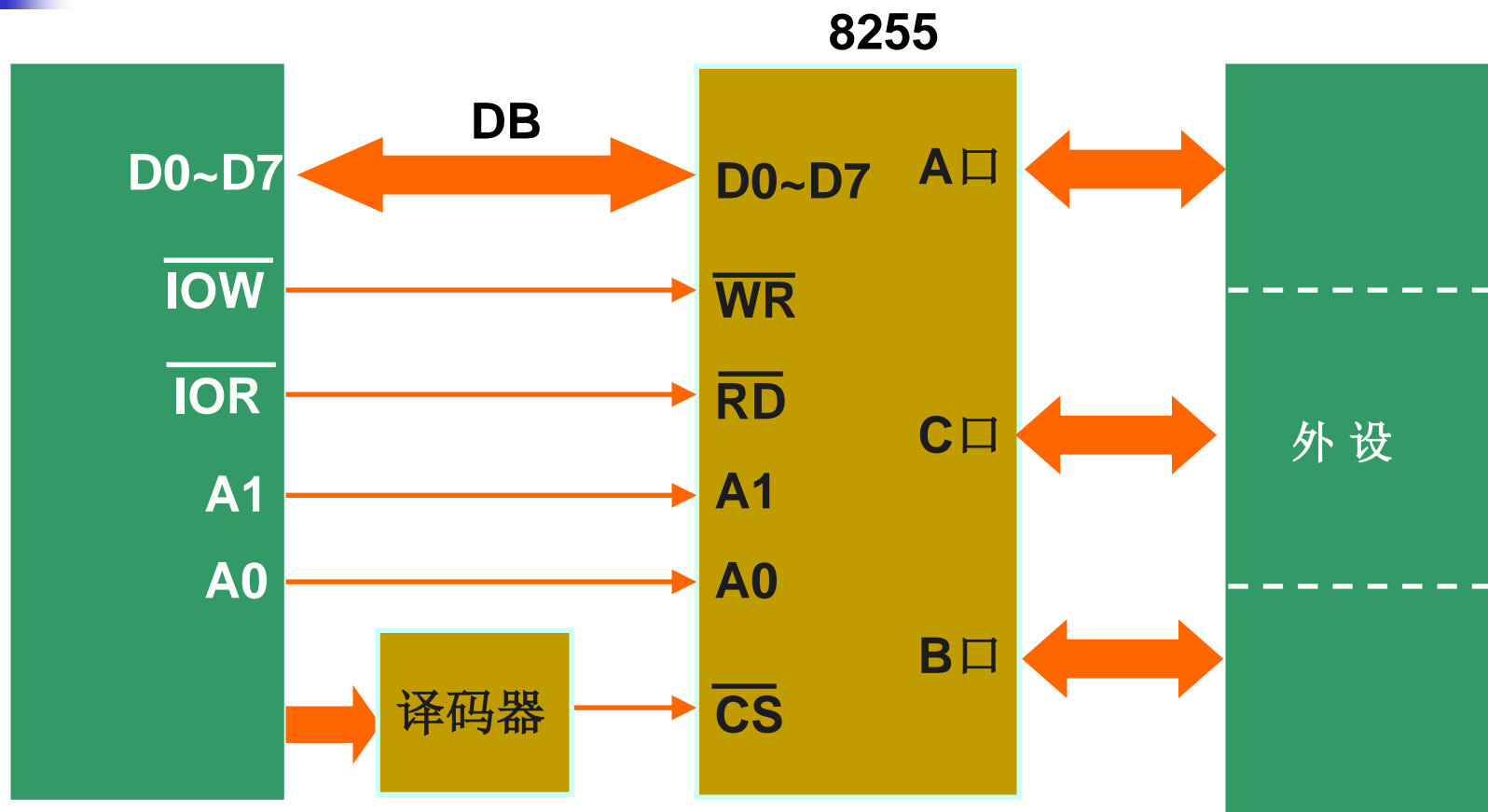
端口识别

片外寻址

片内寻址

\overline{CS}	A1	A0	选中端口
0	0	0	端口A
0	0	1	端口B
0	1	0	端口C
0	1	1	方式控制端口
1	X	X	未选中

8255与系统的连接示意图





7.2.3 方式控制字与状态字



初始化过程中：写入方式控制字

利用控制端口地址， $A_1A_0=11$

工作过程中：

通过数据端口来读写外设数据：

利用端口A、B和C的地址， A_1A_0 依次等于00、01、10

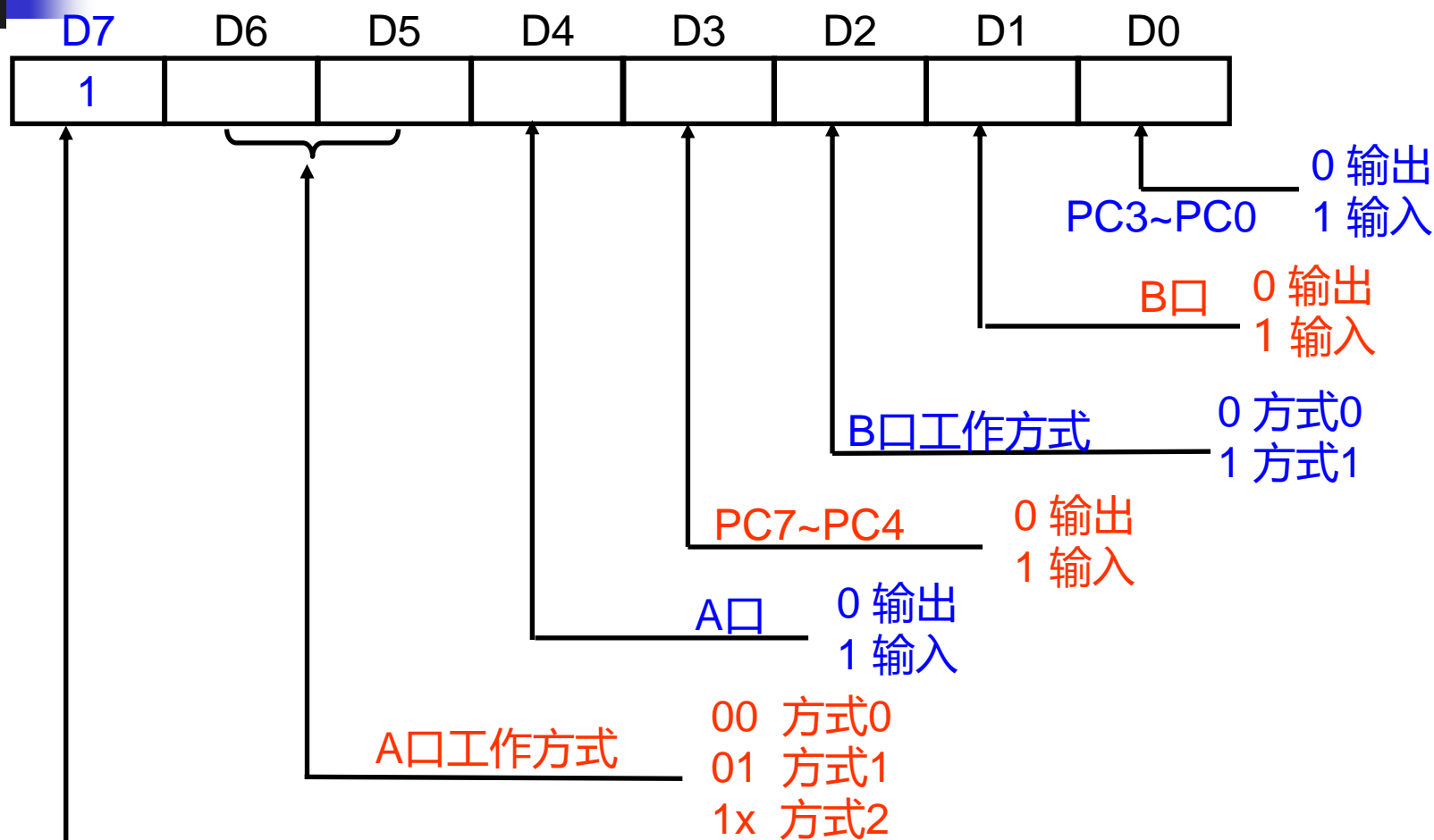
通过端口C来读取状态信息：

利用端口C的地址， $A_1A_0=10$

通过控制端口来写入端口C的位控制字：

利用控制端口地址， $A_1A_0=11$

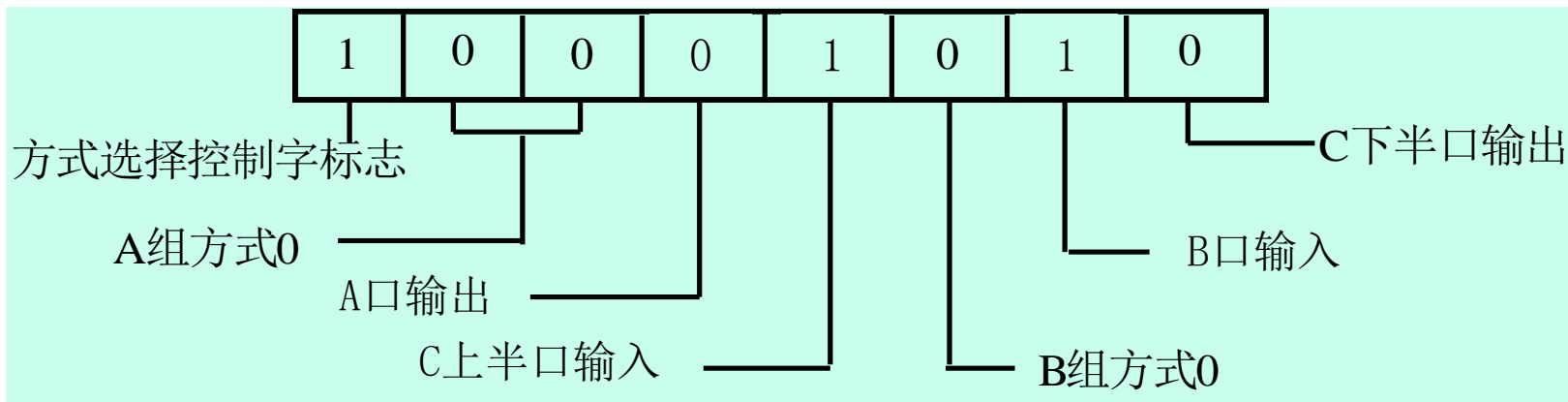
(1) 控制字



特征位，D7=1表示是方式控制字，D7=0表示位操作

方式控制字例1

- 编写对8255的初始化程序片段，要求：使A口为方式0输出，B口为方式0输入，C口上半口作为4位的输入端口使用。C口下半口作为4位的输出端口使用。8255控制端口的地址为43BH。则控制字为：

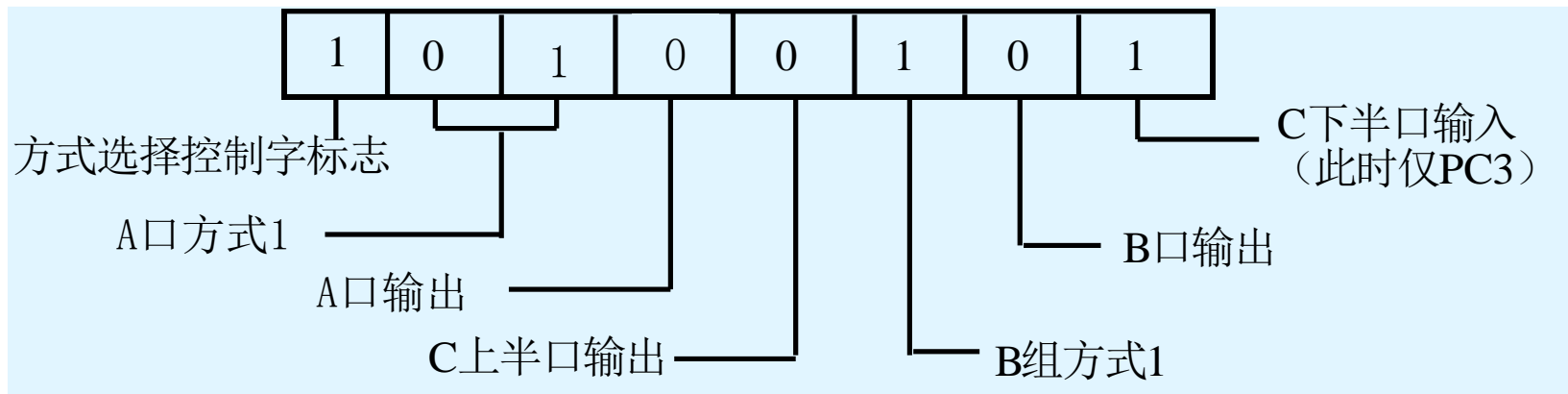


初始化程序片段：

```
MOV AL, 8AH      ; 方式选择控制字
MOV DX, 43BH     ; 由于端口号>256，所以用DX间接寻址
OUT DX, AL       ; 写入控制寄存器
```

方式控制字例2

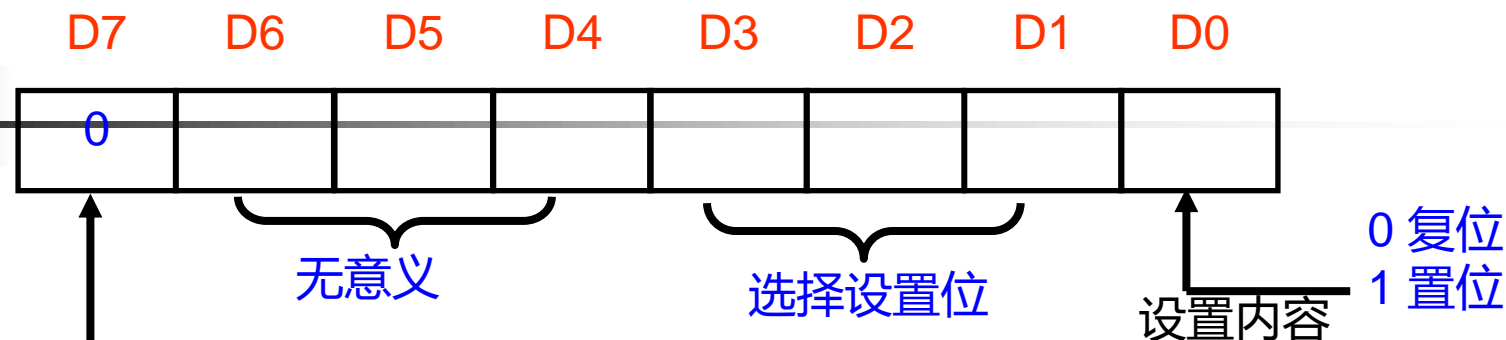
- 如果把A口设置为方式1输出，B口设成方式1输入，C口PC4~PC5输出。则（假设8255的口地址为60H~63H）



方式选择控制字为10100111B=A7H，A、B、C和控制端口的口地址分别为60H、61H、62H和63H。初始化程序只需要两条指令如下

```
MOV    AL,    0A7H    ; 方式选择控制字
OUT    63H,    AL      ; 写入控制寄存器
```

C口按位置位/复位控制字各位含义:



特征位, D7=0
表示是C口按位
置位/复位控制字

D3	D2	D1	设置位
0	0	0	PC0
0	0	1	PC1
0	1	0	PC2
0	1	1	PC3
1	0	0	PC4
1	0	1	PC5
1	1	0	PC6
1	1	1	PC7

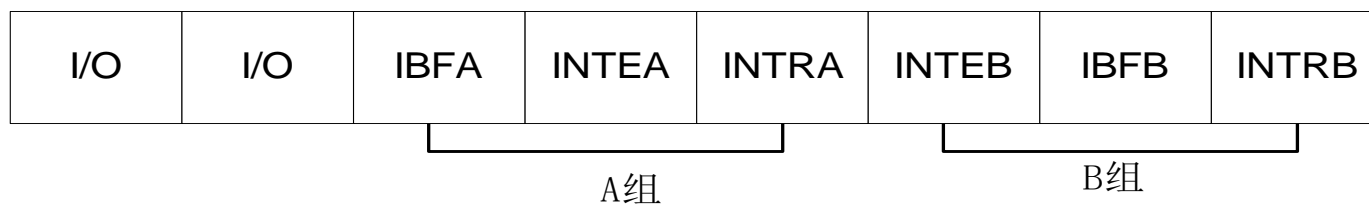
按位操作控制字例1

使A□为方式1输入，且采用中断传送，B□采用方式0输出，其余C□的位作为输出。假设控制端口地址为63H。则：

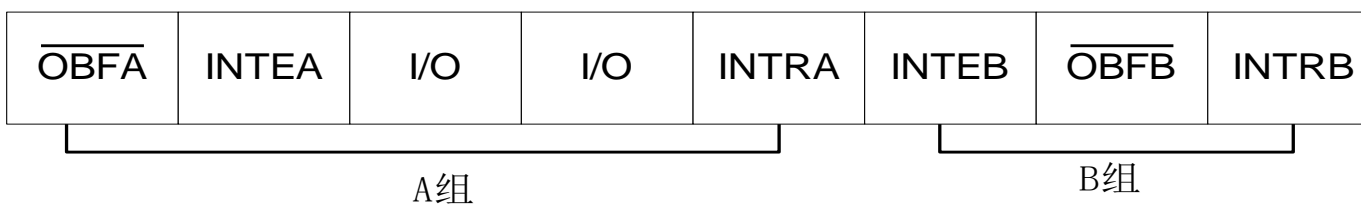
```
MOV AL, 10110000B ; A口方式1输入， B口方式0输出， 其余C口位输出
OUT 63H, AL ; 方式选择控制字写入控制寄存器
MOV AL, 00001001B ; 置位/复位控制字， 使PC4=1， 即使INTE=1允许内部中断
OUT 63H, AL ; 置位/复位控制字写入控制寄存器
```

(2) 状态字

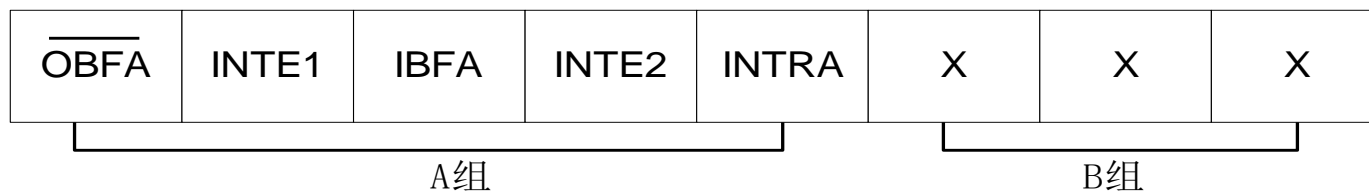
方式1: 输入



方式1: 输出



方式2





使用状态字时要注意的几个问题



- 通过读C端口取得，与C的外部引脚无关
- 查询输入：IBF 和 INTR
- 查询输出：OBF 和 INTR
- 一般查INTR位
- INTE位是通过按位置位/复位来设置的



7.2.4 工作方式

- **方式0：基本输入输出方式（PA、PB和PC）**
 - 适用于**无条件传送和查询方式**的接口电路
- **方式1：选通输入输出方式（PA和PB）**
 - 适用于**查询和中断方式**的接口电路
- **方式2：双向选通传送方式（PA）**
 - 适用于与双向传送数据的外设
 - 适用于**查询和中断方式**的接口电路

A:方式0 — 基本输入输出方式

(1) 端口A、B和C (PC7~PC4及PC3~PC0) 都可工作于输入或输出。三个端口无必然联系, 各端口输入输出组合共有16种, 可适用于多种用途。

(2) 对C口, 其高低4位两部分可相同 (同为输入或输出) 也可不同。但对端口8位的读写是在一条指令中作为整体来操作, 可采用以下措施:

CPU	高4	低4	
IN	入	出	必须屏蔽低四位
IN	出	入	必须屏蔽高四位
IN	入	入	读入的8位均有用
OUT	入	出	输出数据只设在低4位
OUT	出	入	输出数据只设在高4位
OUT	出	出	输出8位数据

(3) CPU与外设采用无条件输入/出, 不用联络线, 也不需要查询。

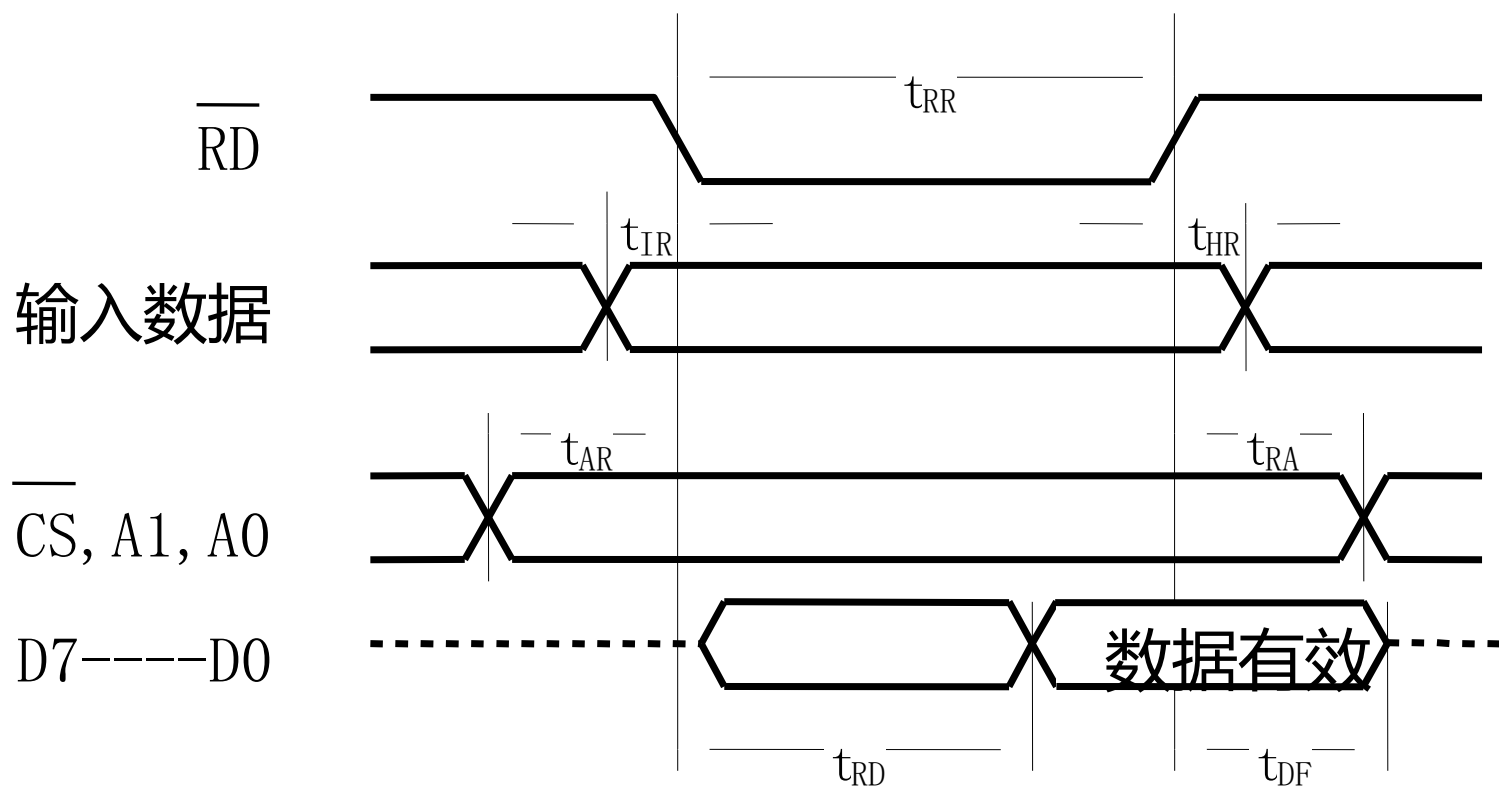
(4) 方式0也可用于查询传送的场合, 这时可令A、B口为数据口, C口做与外设的联络信号, A、B、C配合即可工作在查询场合。

(5) 方式0下输入不缓冲, 输出锁存。CPU对数据口的输入指令读取端口当前信息, 对数据口的输出指令把数据输出到端口输出锁存器。

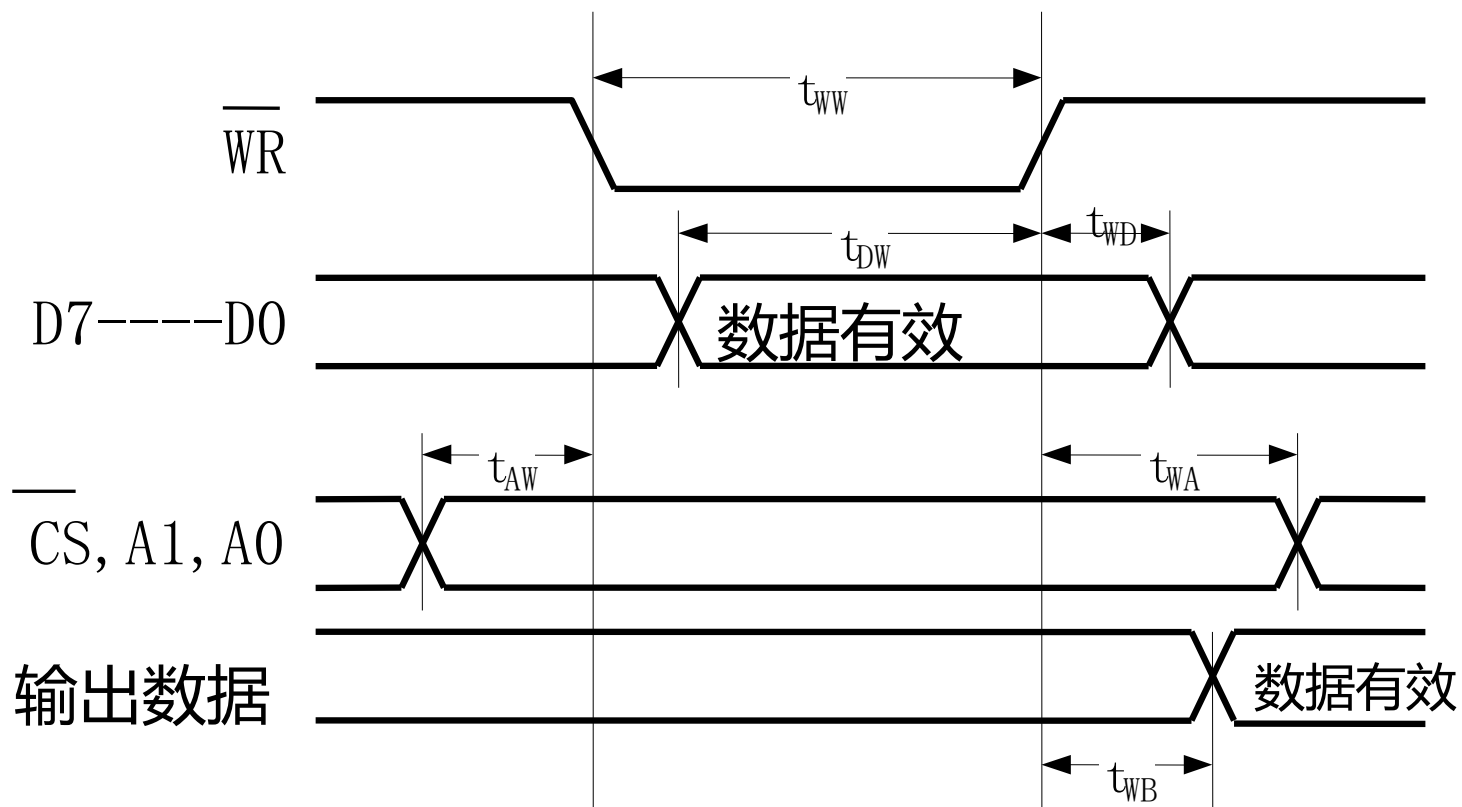


A		B		Group A			Group B	
D ₄	D ₃	D ₁	D ₀	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

方式0作为输入口时序



方式0作为输出口时序



方式0编程示例

➤ 端口地址30H~33H。

MOV AL, 89H	; A口、B口输出, C口输入
OUT 33H, AL	; 写方式控制字
IN AL, 32H	; 读C口
NOT AL	; C口数据取反
OUT 30H, AL	; 写A口
MOV AL, 0F0H	; 取B口数据
OUT 31H, AL	; 写B口

➤ 端口地址60H~63H。

MOV AL, 88H	; C口高4位输入, 低4位输出
OUT 63H, AL	; 写方式控制字
IN AL, 62H	; 读C口 (高4位)
MOV CL, 4	
SHR AL, CL	; C口的高4位移到低4位
OUT 62H, AL	; 写C口 (低4位)

B：方式1 — 选通输入输出方式

方式1端口A和B输入/输出数据时，必须利用端口C提供联络信号（选通、应答信号）。

一般情况下，用中断传送数据方式（也可用查询方式）

这时：

通道A：8位数据端口（A口），3位控制位（C口）。

通道B：8位数据端口（B口），3位控制位（C口）。

这种配合固定，不可以用程序修改。

◆ 特点：

- ① 端口A、B既可做输入又可做输出，数据输入输出都被锁存。
- ② 端口C引脚分别作为控制、状态和中断申请信号等。
- ③ 若只有一个端口工作在方式1，则余下13位可工作在方式0。

若两个端口都工作在方式1，则余下2位可由程序指定为输入或输出。

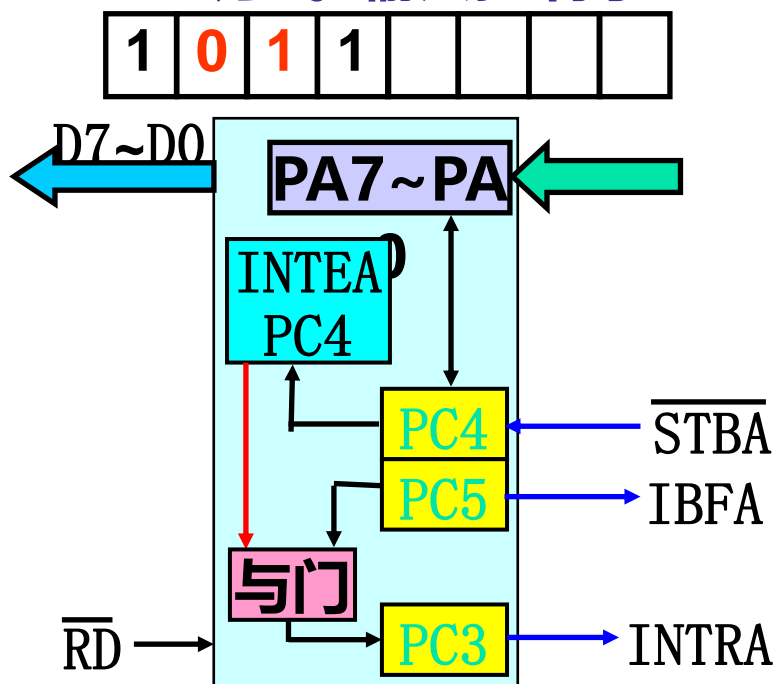
端口A、B工作于方式1的输入或输出时，端口C联络信号的引脚和意义都不同，故以下分开讨论。

(1) 选通输入方式

◆信号说明:

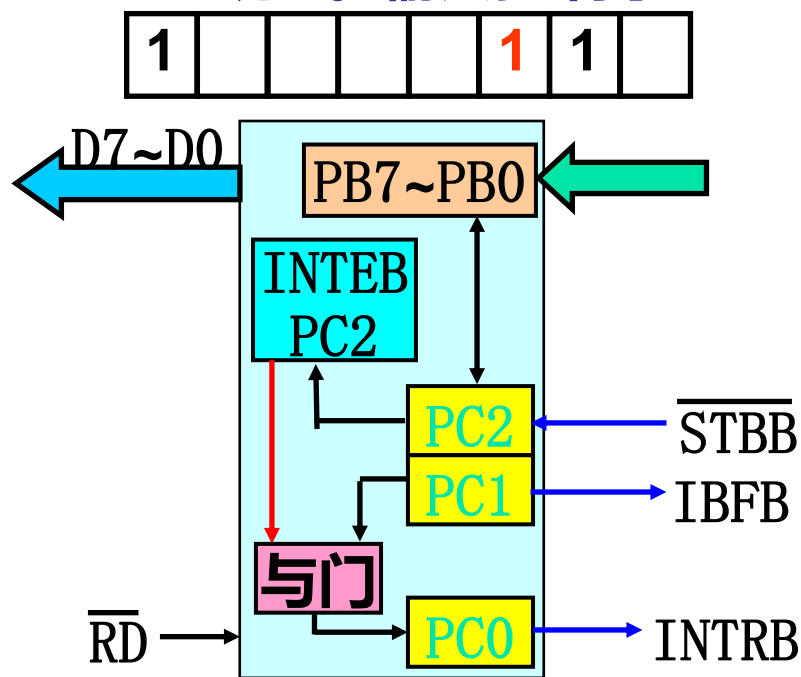
- a. STB 选通输入: 低电平有效, 表示外设送出新的数据。
- b. IBF 输入缓冲区满: 输出, 高电平有效。通知外设输入缓冲区满, 不要送新数据, 由CPU读取数据时 (RD的上升沿) 清除。
- c. INTR 中断请求信号: 输出, 高电平有效, 可向CPU申请中断。

A口方式1输入控制字



A口方式1输入时相应的联络信号

B口方式1输入控制字



B口方式1输入时相应的联络信号

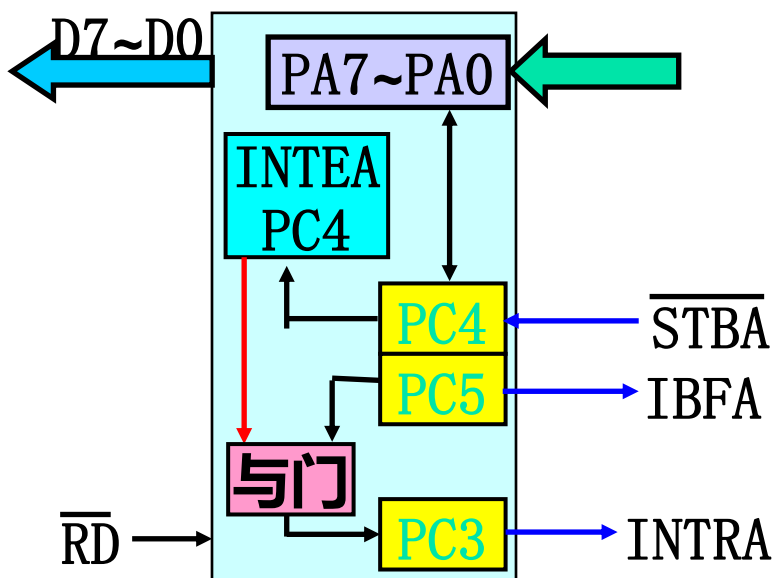
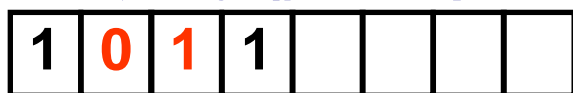
STB、IBF、INTE均为1时，8255A自动发出INTR。

d. INTE 中断允许：INTEA是C口输出锁存器的PC4，INTEB是C口输出锁存器的PC2，由它们分别控制A口和B口的中断申请。

INTE=1 中断允许，

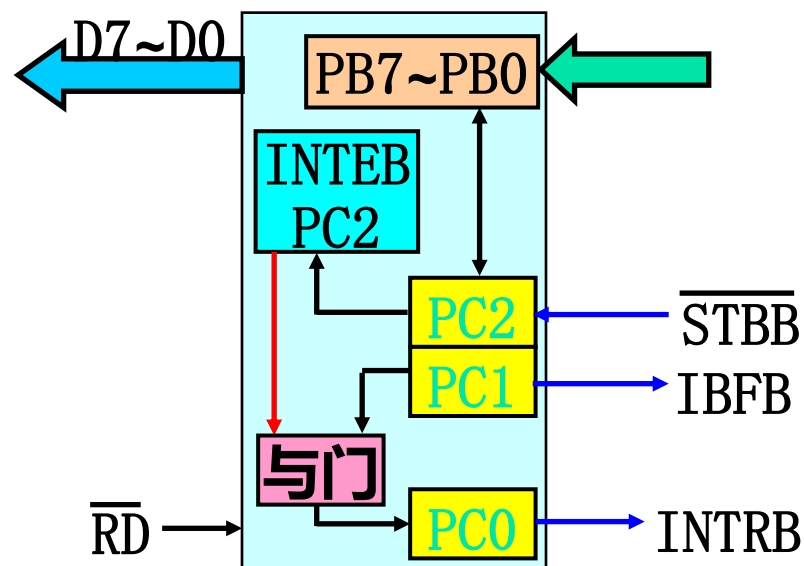
INTE=0 禁止中断。（可由C口置 /复位控制字将INTE置1 或复位为0）

A口方式1输入控制字

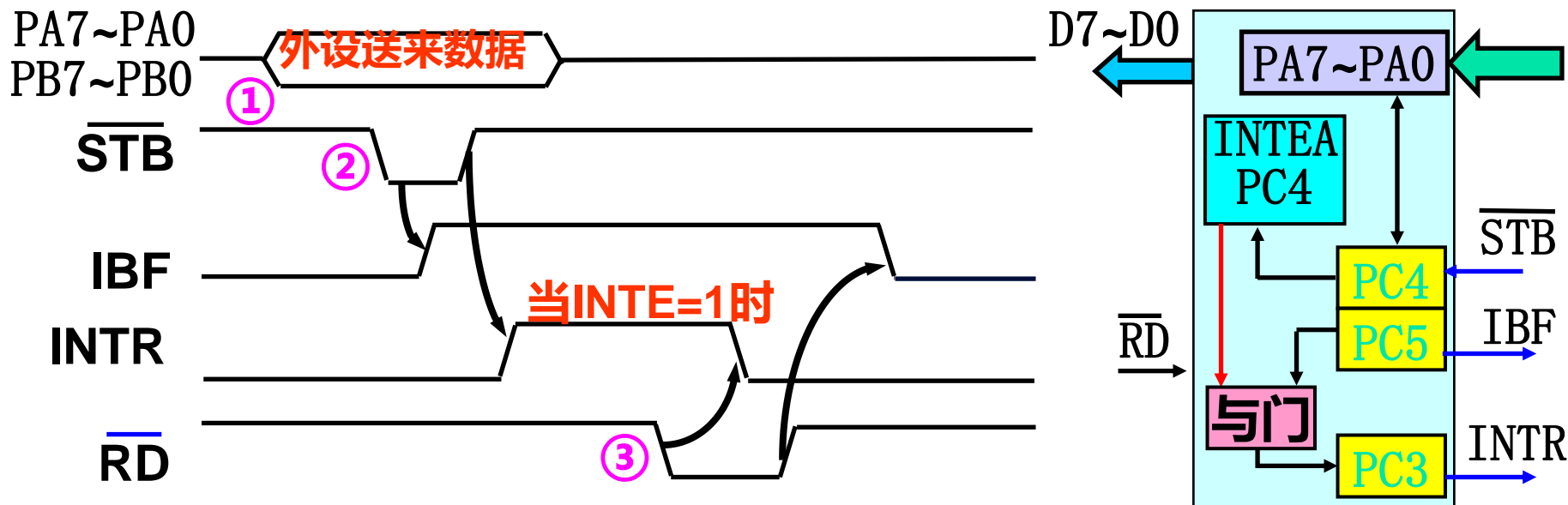


A口方式1输入时相应的联络信号

B口方式1输入控制字



B口方式1输入时相应的联络信号



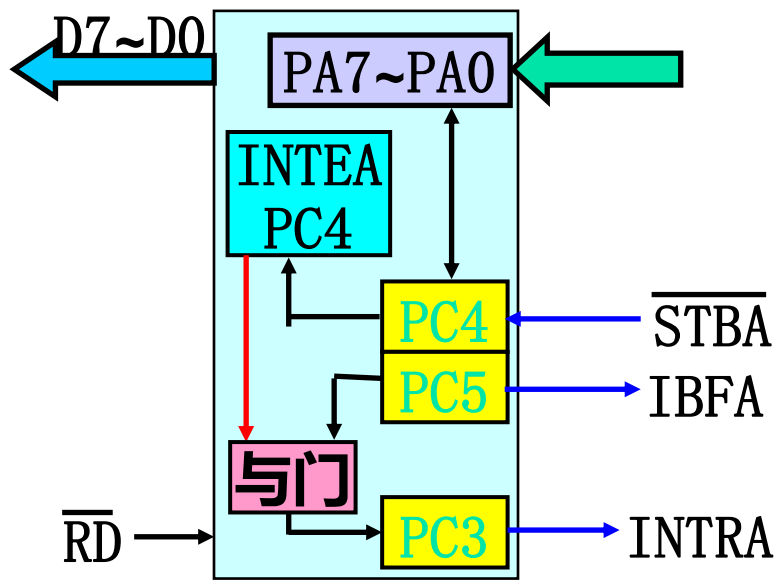
以中断方式输入数据:

- ① 当外设数据准备好，将数据送至端口线PA7~PA0或PB7~PB0
- ② 外设向8255发出选通信号STB:
 - (1) 将数据锁存在输入端口内。
 - (2) 使IBF变高，表示输入端口满，可用于阻止外设输入新数据
 - (3) 如果INTE=1, STB的上升沿使INTR变高，发出中断请求。
- ③ 中断处理程序中CPU 读取数据，发出RD信号:
 - (1) RD 的下降沿清除INTR
 - (2) RD 的上升沿清除IBF
 - (3) 端口内的数据进入CPU

注意:

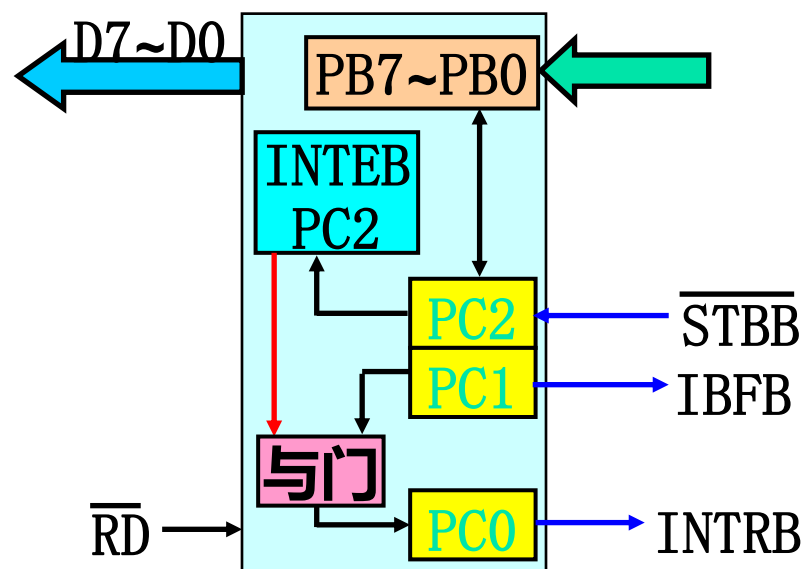
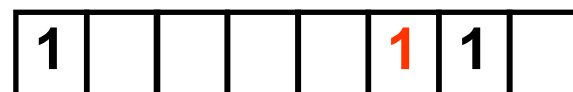
在方式1下, 作为联络信号的外部引脚PC4、PC2, 不受C口按位置位/复位控制字控制, 即对这些位的置位/复位不影响这些引脚信号的输入/输出, 而只在8255A内部对INTE信号起作用。

A口方式1输入控制字

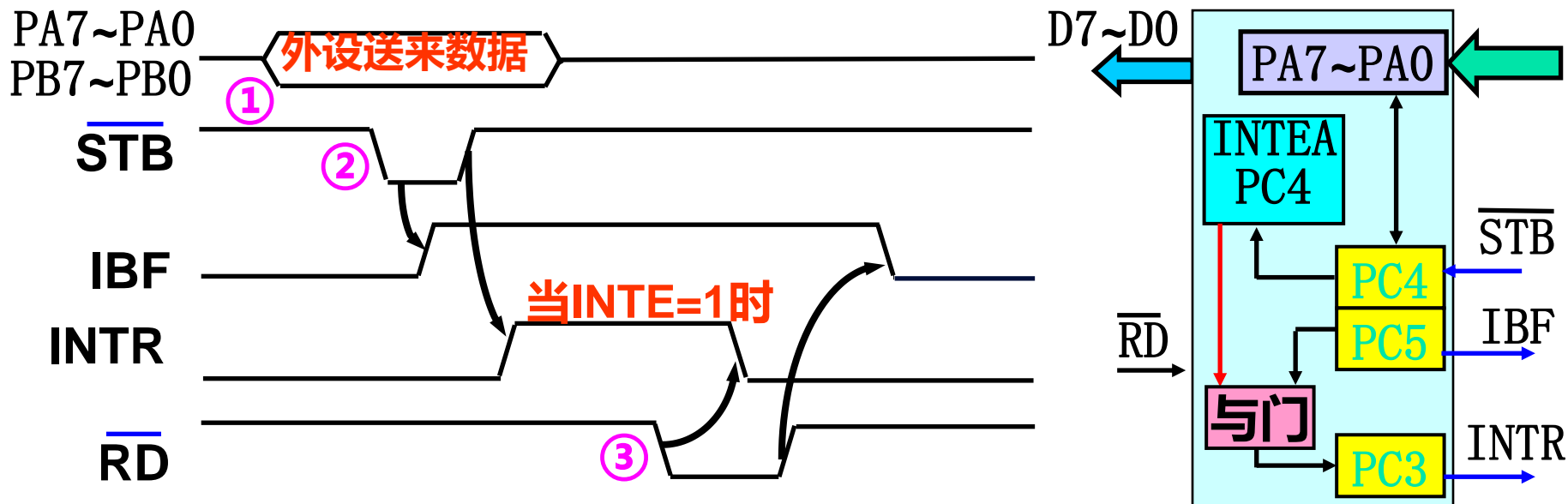


A口方式1输入时相应的联络信号

B口方式1输入控制字

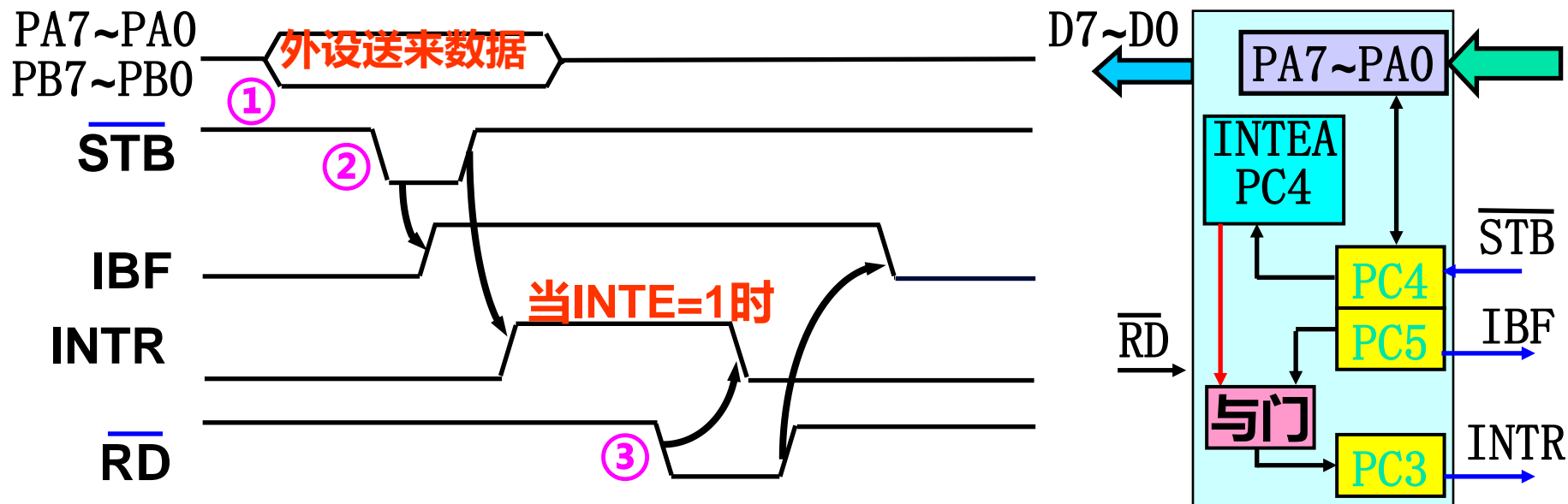


B口方式1输入时相应的联络信号



以查询方式输入数据:

- ① 当外设数据准备好，将数据送至端口线PA7~PA0或PB7~PB0
- ② 外设向8255发出选通信号STB:
 - (1) 将数据锁存在输入端口内。
 - (2) 使IBF变高，表示输入端口满，可用于阻止外设输入新数据
- ③ 在主程序中CPU 查询IBF，若其为高电平，读取数据，发出RD信号:
 - (1) RD 的上升沿清除IBF
 - (2) 端口内的数据进入CPU



查询方法从8255的A口向CPU输入数据:

方式1的状态信号可通过读取端口C得到

```

INPUT:    IN      AL, PortC      ; 读方式1状态信号
          TEST    AL, 20H        ; 查A口的IBF
          JE      INPUT          ; IBF=0, 数据未准备好。
          IN      AL, PortA      ; IBF≠0, CPU读入数据。
  
```

注意:

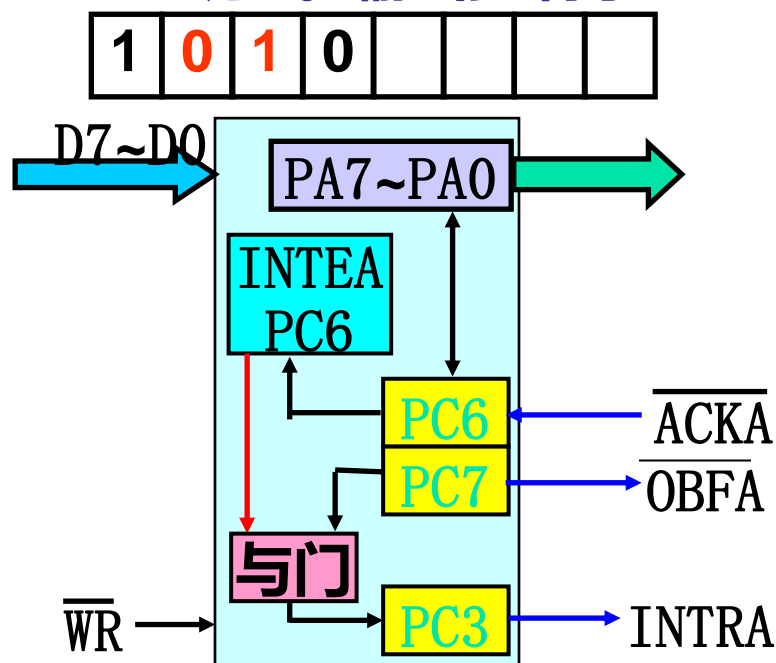
方式1输入时共用到C口6个引脚, 剩余2个引脚PC6、PC7可做I/O位, 或由C口置 /复位控制字决定其输出。

(2) 选通输出方式

◆信号说明:

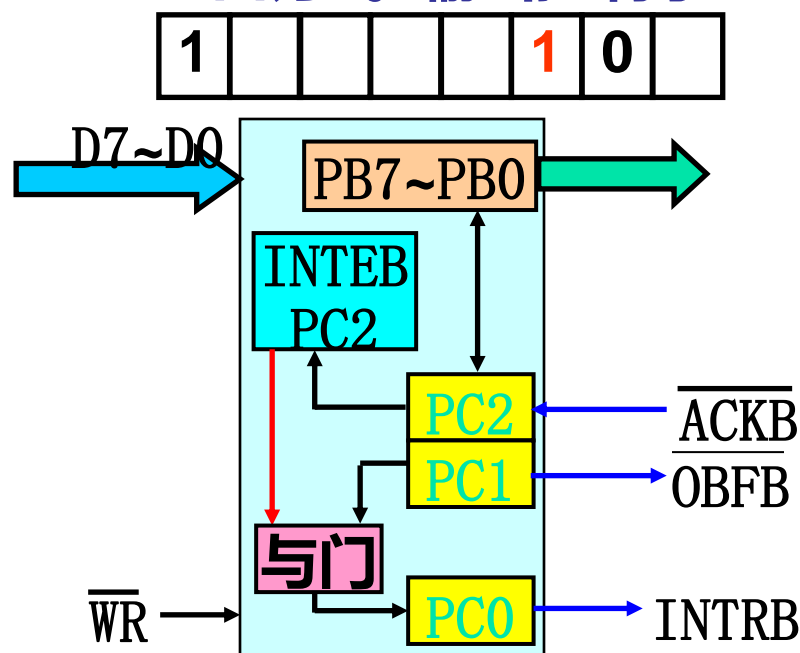
- a. $\overline{\text{OBF}}$ 输出缓冲器满, 输出, 低电平有效, 通知外设端口已有可用数据。
- b. $\overline{\text{ACK}}$ 收到信号, 输入, 低电平有效。外设通知8255已取走数据, 由其清除 $\overline{\text{OBF}}$, 并置“1”INTR。
- c. INTR 中断请求信号, 输出, 高电平有效, 可向CPU申请中断。

A口方式1输出控制字



A口方式1输出时相应的联络信号

B口方式1输出控制字



B口方式1输出时相应的联络信号

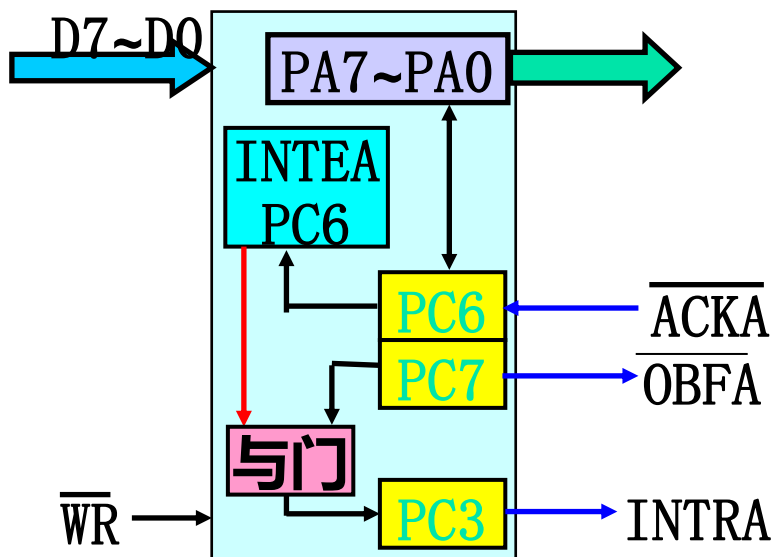
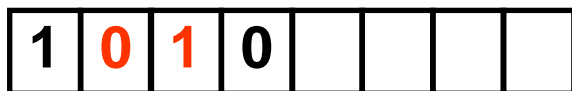
OBF、ACK、INTE均为 1 时，8255A自动发出INTR。

d. INTE 中断允许信号：INTEA为C口输出锁存器的PC6，INTEB为C口输出锁存器的PC2。由它们分别控制A口和B口的中断申请。

INTE=1 中断允许，

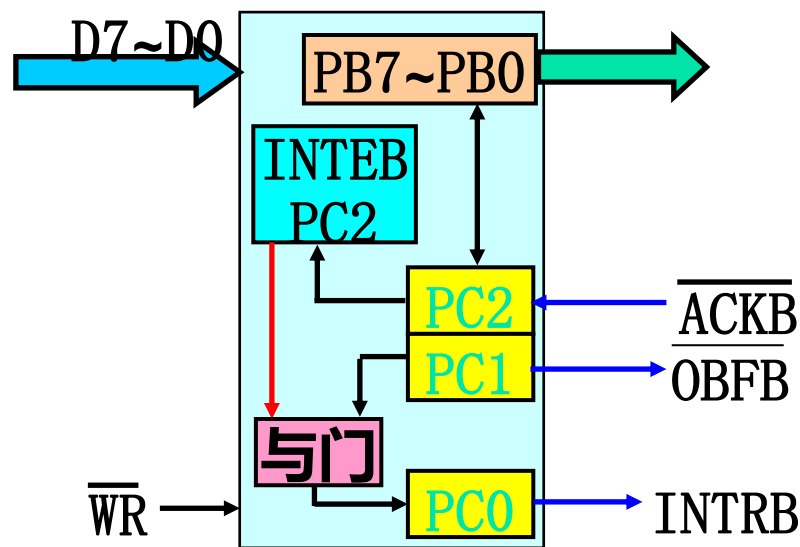
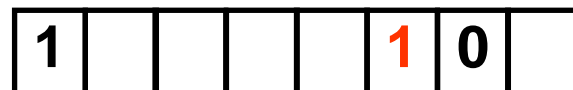
INTE=0 禁止中断。（可由C口置/复位控制字将INTE置1或复位为0）

A口方式1输出控制字

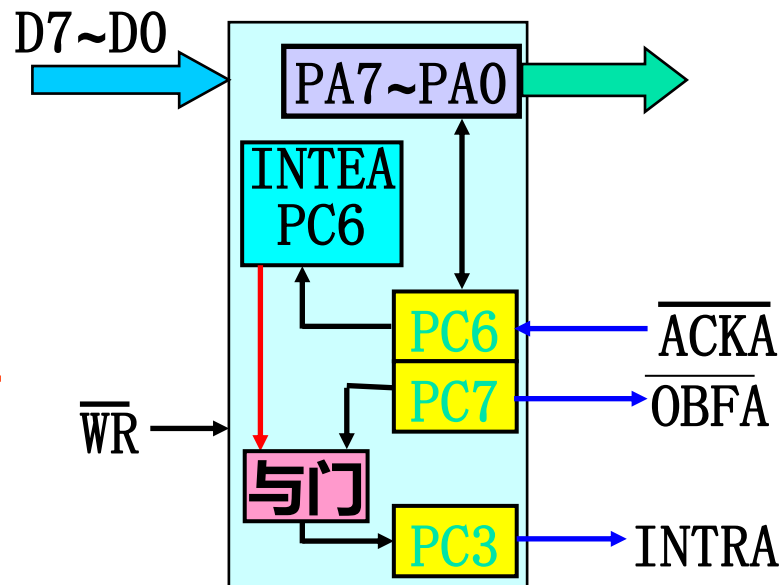
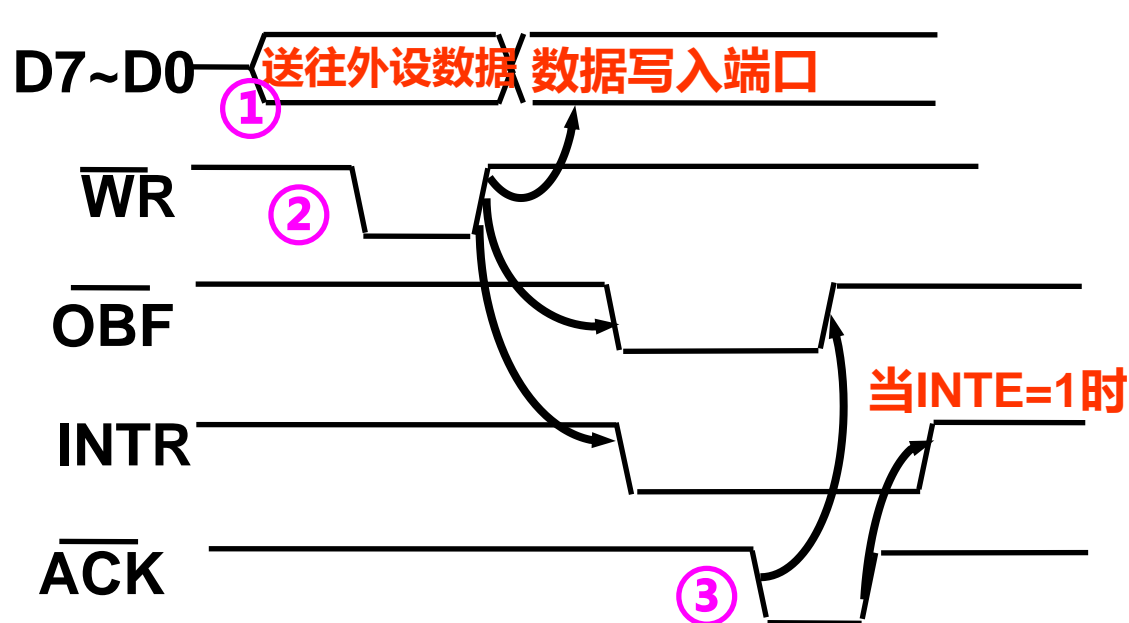


A口方式1输出时相应的联络信号

B口方式1输出控制字



B口方式1输出时相应的联络信号



以中断方式输出数据:

① CPU输出数据

② CPU发出WR:

(1) 数据写到8255A的端口

(2) 使OBF有效, 表示输出端口满, 可作为外设的选通信号, 通知外设取数据。

(3) 清除中断请求信号INTR。

③ 外设接收到数据后, 向8255发出ACK信号:

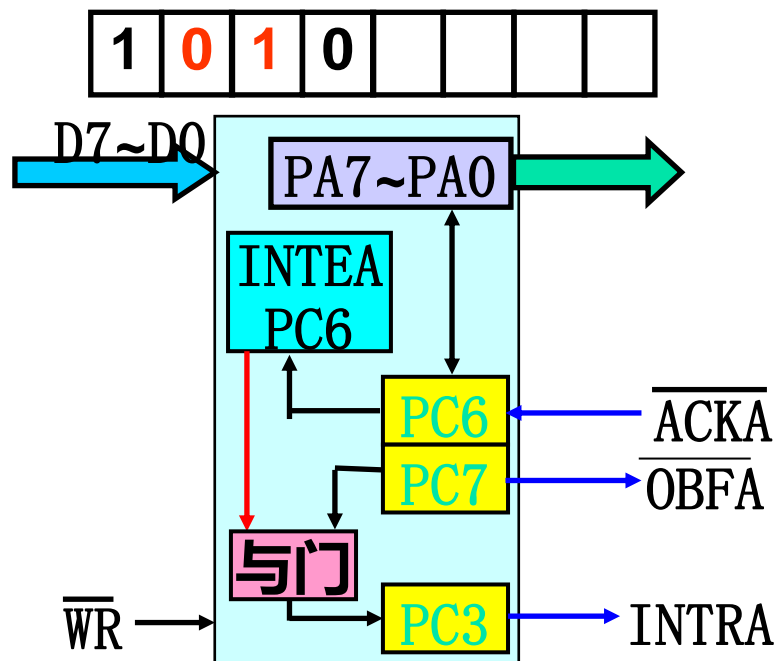
(1) ACK的下降沿使OBF变高,

(2) 当INTE=1, ACK的上升沿使INTR变高, 发出中断请求, 请求CPU输出新的数据。

注意:

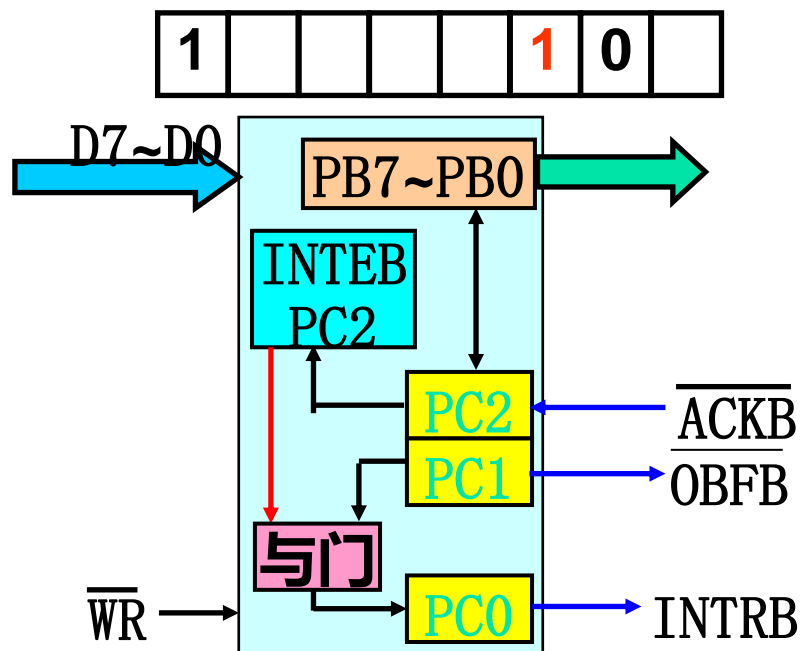
在方式1下，作为联络信号的外部引脚PC6、PC2，不受C口按位置位/复位控制字控制，即对这些位的置位/复位不影响这些引脚信号的输入/输出，而只在8255A内部对INTE信号起作用。

A口方式1输出控制字



A口方式1输出时相应的联络信号

B口方式1输出控制字



B口方式1输出时相应的联络信号

注意：方式1输出时共用到C口6个引脚，剩余2个引脚PC4、PC5可做I/O位，或由C口置/复位控制字决定其输出。

方式1联络信号表

端口方式	联络线	输入信号	输出信号
端口A 方式1	PC_7	I/O	\overline{OBF}_A
	PC_6	I/O	ACK_A/\overline{INTE}_A
	PC_5	IBF_A	I/O
	PC_4	STB_A/\overline{INTE}_A	I/O
	PC_3	$INTR_A$	$INTR_A$
端口B 方式1	PC_2	STB_B/\overline{INTE}_B	ACK_B/\overline{INTE}_B
	PC_1	IBF_B	\overline{OBF}_B
	PC_0	$INTR_B$	$INTR_B$

8255A方式1的工作特点

- 端口A或端口B可以选择使用方式1，并可选择各自为输入或输出。在选定的同时，自动规定了端口C有关的联络、控制和中断请求信号。
- 若采用方式1中断方式，必须将相关的INTE置为1，端口A或端口B才可以使用各自的INTR信号申请中断。
- 若采用方式1查询方式，可以通过查询端口A或端口B相关IBF或OBF信号的当前状态，决定是否能进行数据传输。
- 若端口A和端口B均选定为方式1，则需端口C提供6位做联络信号，剩下的2位还可工作在方式0的输入/出方式。
- 若端口A和端口B中只有一个工作在方式1，而另一个工作在方式0，则端口C中有3位作为方式1的联络信号，端口C其余5位均可工作在方式0的输入/出方式。

8255A方式1编程示例

- 8255端口A采用方式1（中断方式）的数据输入；端口B为方式0的数据输出。
- 设8255的端口地址：60H~63H。

.....

CLI

A1: MOV AL, 0B0H ; 方式控制字 (0B0H)

OUT 63H, AL

MOV AL, 09H ; C口置/复位控制字 ($PC_4=1$)

OUT 63H, AL ; 端口A中断允许 ($INTE_A=1$)

A2: (其他相关中断的设置)

STI

.....

端口A的中断请求

C: 方式2 — 双向方式 (仅A口)

(1) 与方式0、方式1的单向传送不同，工作在方式2下的端口，具有双向传送功能。

(2) 端口工作在方式2时，C口的某5根引脚作为端口的联络信号；
是方式1下A口输入、输出联络信号的组合。

(3) C口未做联络信号的3条引脚，可作为B口在方式1下的联络线，也可和B口一样工作在方式0。由方式控制字决定其输入/输出。

(4) **方式2下的时序为方式1输入/输出两者的组合。**

关于8255中断允许/禁止的设定：

例： A口 INTE=1 → PC4=1 (方式1, 2) 输入

0	0	0	0	1	0	0	1
---	---	---	---	---	---	---	---

or B口 INTE=1 → PC2=1 (方式1) 输入

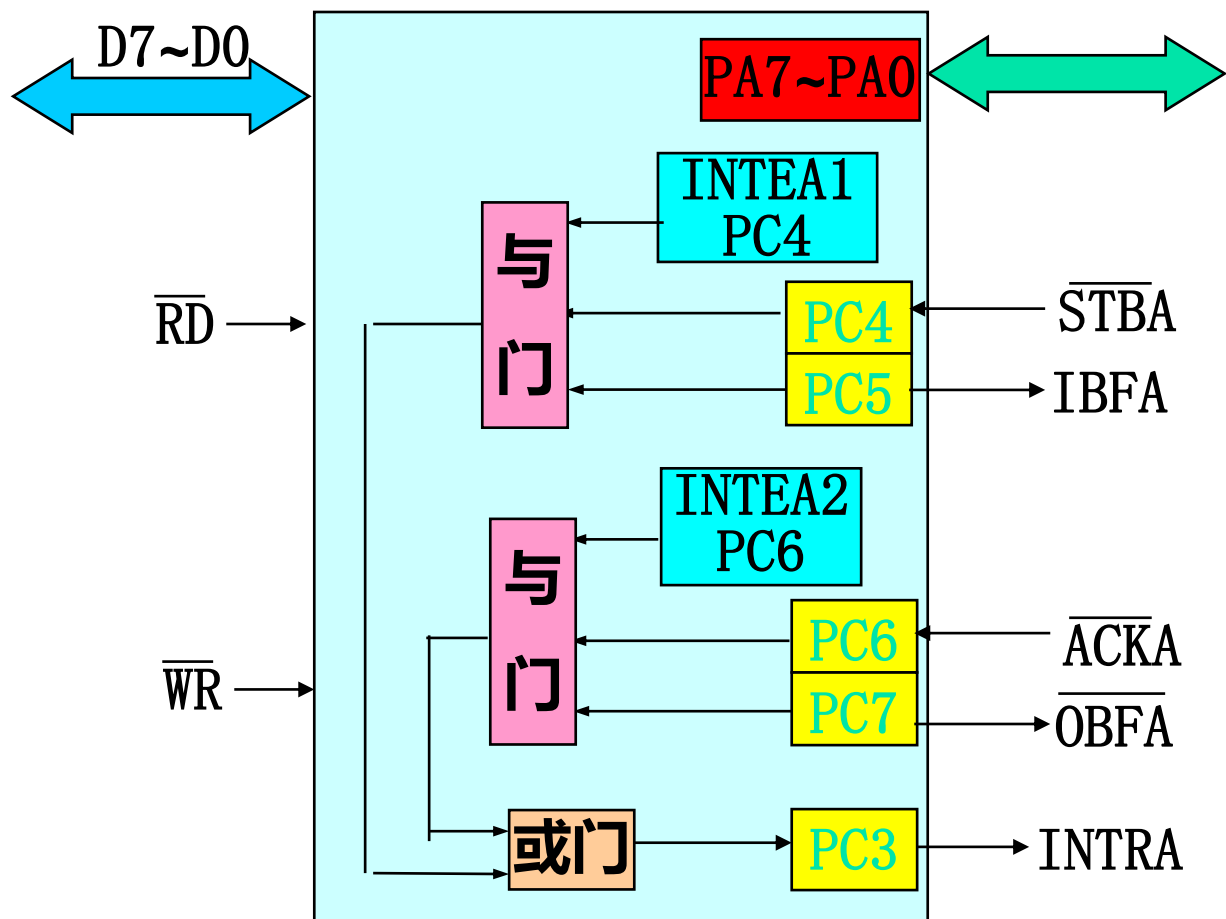
0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

MOV AL, 09H

写入控制口 OUT 63H, AL

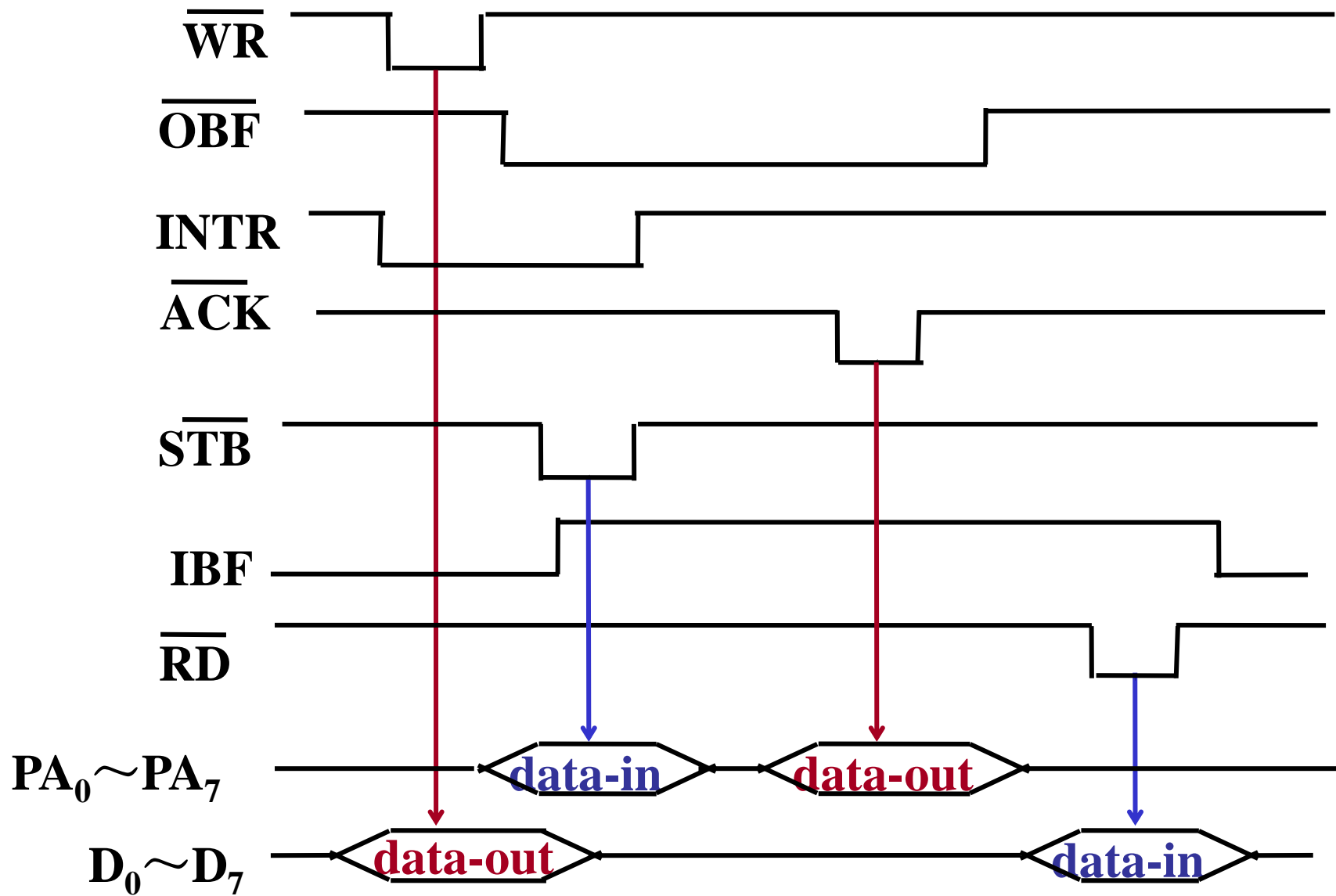
这时的位操作只影响INTE的状态，而不影响PC4引脚的状态。

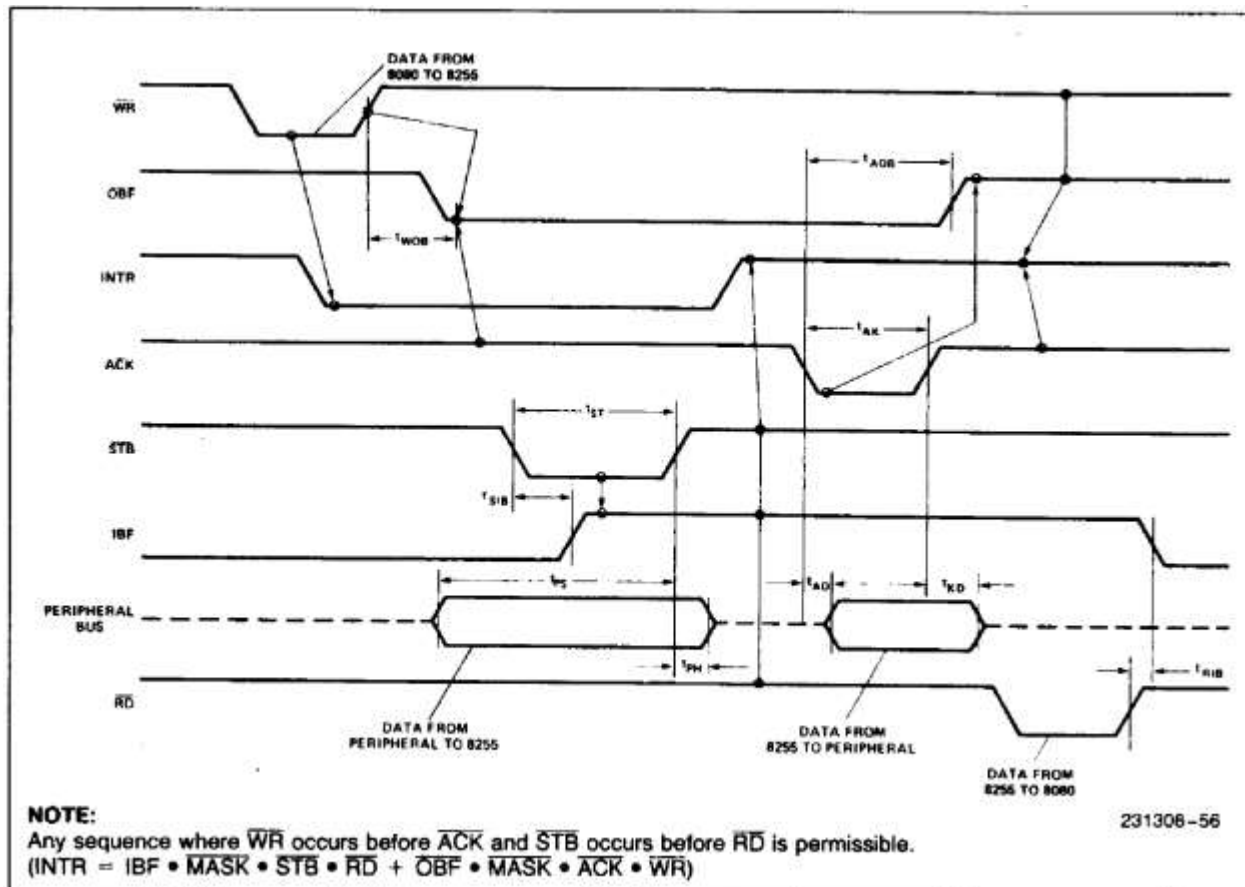
工作在方式2时，C口有5根引脚作为A口的联络信号，是方式1下A口输入、输出联络信号的组合。



方式2下的时序为方式1输入/输出两者的组合(略)

方式2双向时序





8255A方式2的工作特点

- ◆ **方式2** 是双向传输，相当于方式1的输入和输出的组合，只适用于**端口A**。外设在端口A的8位数据线上分时向CPU发送数据或从CPU接收数据。
- ◆ **方式2** 需要**端口C**提供固定的5位，即端口C的 $PC_7 \sim PC_3$ ，做指定的控制/联络信号。
- ◆ **方式2** 一般采用中断方式进行数据传输。

联络线	联络信号
PC_7	\overline{OBF}_A
PC_6	$\overline{ACK}_A / INTE_1$
PC_5	IBF_A
PC_4	$\overline{STB}_A / INTE_2$
PC_3	$INTR_A$



7.2.5 8255芯片的应用

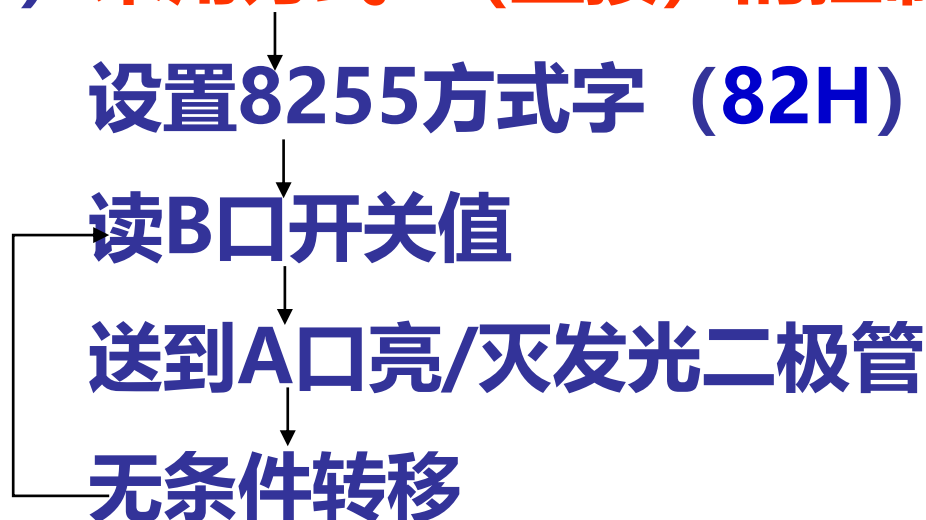
- 芯片与系统的连接
- 芯片的初始化
- 相应的控制程序

8255A应用例1

开关控制发光二极管显示电路。

- ◆ 8255的端口A接发光二极管组，端口B接开关组。
- ◆ 8255的端口地址： 60H~63H。

(1) 采用方式0（直接）的控制：

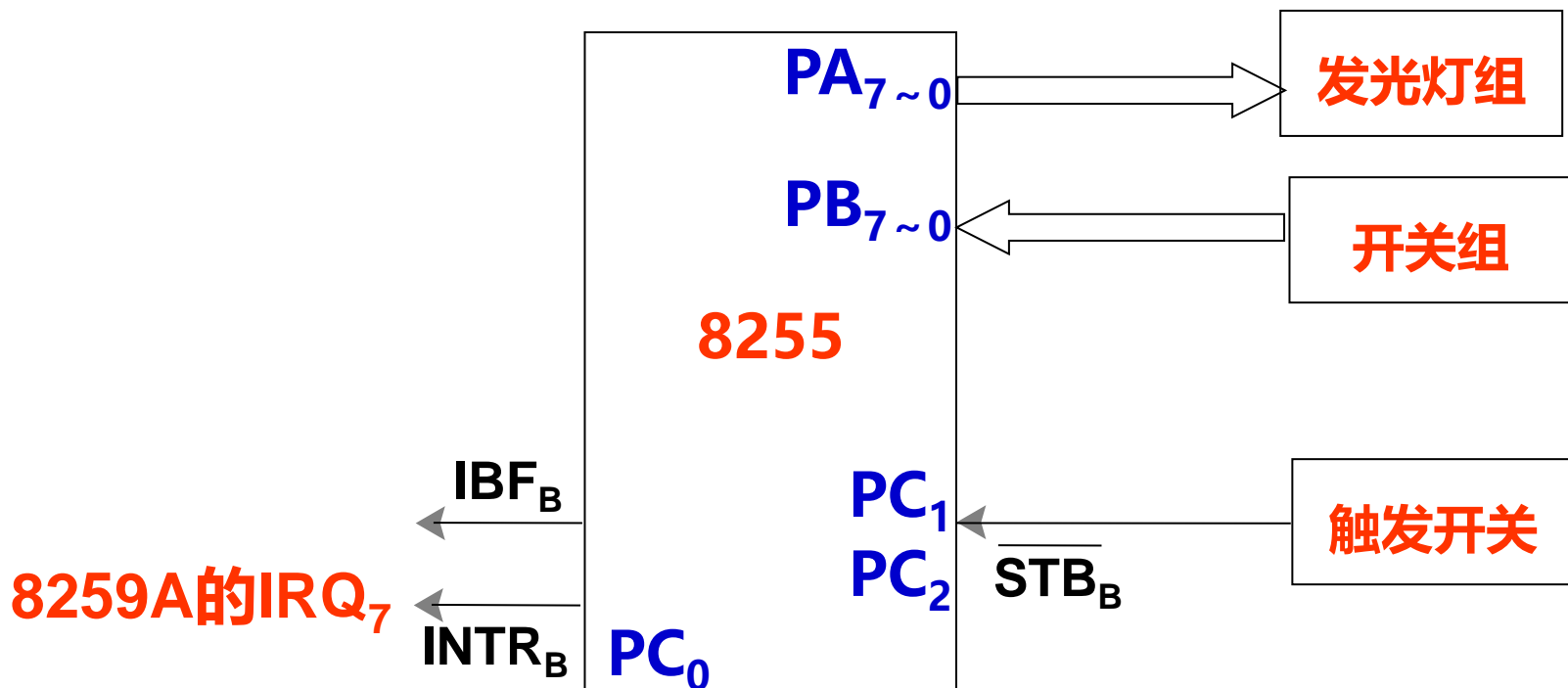


```
MOV AL, 82H
OUT 63H, AL
A1: IN AL, 61H
OUT 60H, AL
JMP A1
```

(2) 采用方式1（中断）的控制：

用手动触发开关做 $\overline{\text{STB}}_{\text{B}}$ 信号，引发B端口中断。 INTR_{B} (PC_0) 接8259A的 IRQ_7 端（中断类型为0FH）。

8259端口地址：20H~21H。



应用例1中断方式程序

MOV AX,OFFSET IRQ7

MOV [003CH],AX

MOV AX, 0

MOV [003EH],AX ; 设置IRQ7中断向量表

CLI

IN AL, 21H

AND AL,7FH

; 设置IRQ7中断开放

OUT 21H,AL

IRQ7中断子程序:

MOV AL, 86H

IRQ7: IN AL, 61H

OUT 63H,AL ; 设置

OUT 60H, AL

MOV AL,05H

MOV AL, 20H

OUT 63H,AL ; 设置

OUT 20H, AL

A1: STI

IRET

JMP A1

(3) 采用方式1 (查询) 的控制:

通过查询8255的PC₁ (IBF_B) 了解触发开关是否按动。每按动一次触发开关, 读开关组数据, 亮/灭LED发光管组。

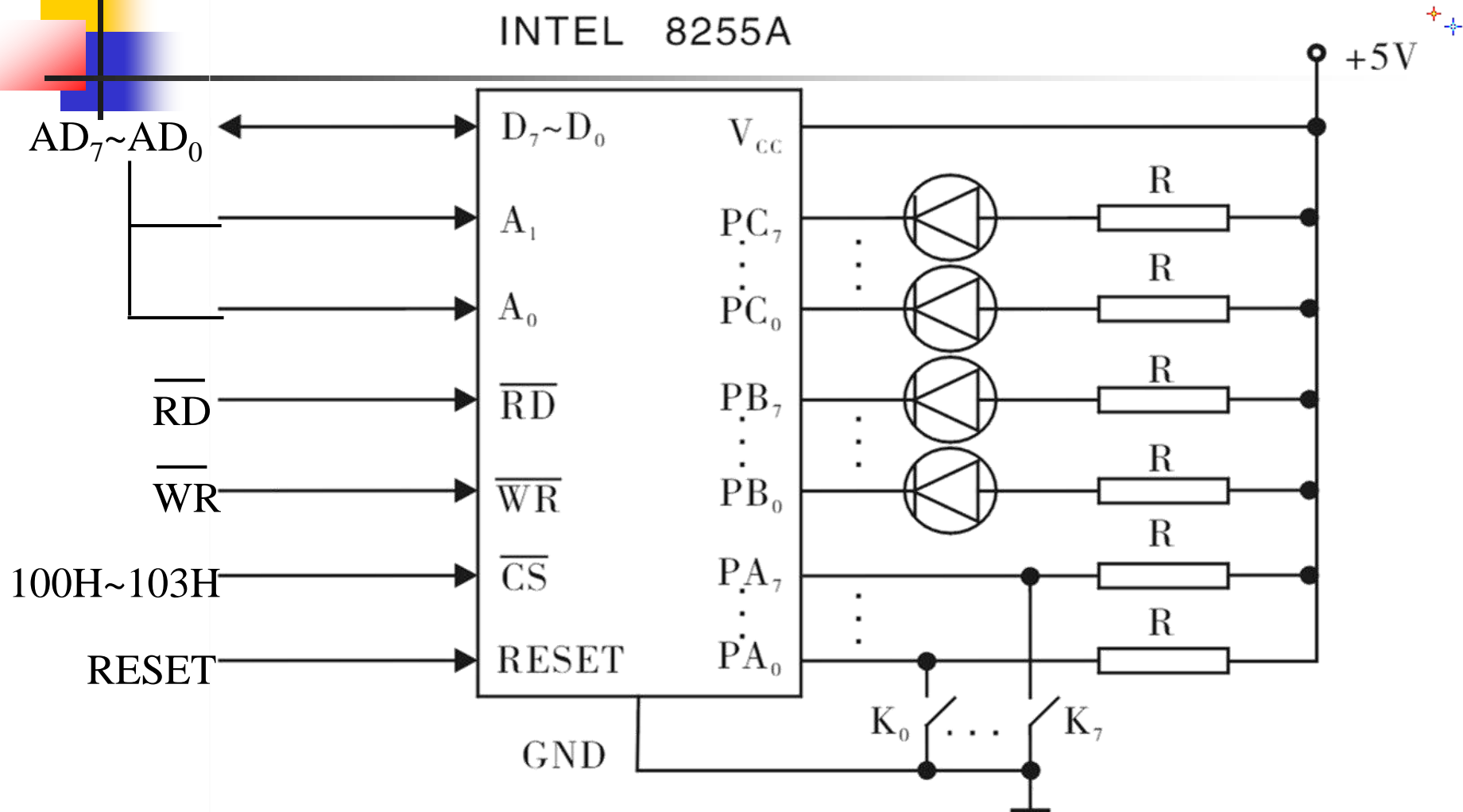
```
MOV AL, 86H      ; 设置8255方式字 (86H)
OUT 63H, AL
MOV AL, 04H      ; 设置PC2=0, B口中断屏蔽
OUT 63H, AL
A1: IN AL, 62H
TEST AL, 02H ; 测试PC1 (IBFB) = 1 ?
JZ A1
IN AL, 61H ; 读B口开关值, 并清除IBFB
OUT 60H, AL ; 亮/灭对应的发光二极管
JMP A1 ; 循环
```



例2：

- 在工业控制过程中，经常需要检测某些开关的状态。例如，在某一系统中，有8个开关K7 ~ K0，要求不断地检测它们的通断状态，并随时在发光二极管上显示。通过8255A的端口A读入开关状态信息，使端口B、端口C连接的发光二极管的状态与端口A开关状态相呼应，并重复执行。假设8255A在系统中端口A、B、C及控制口的地址分别为100H，101H，102H，103H，试设计出硬件电路图，并编写初始化程序

解：(1) 硬件电路

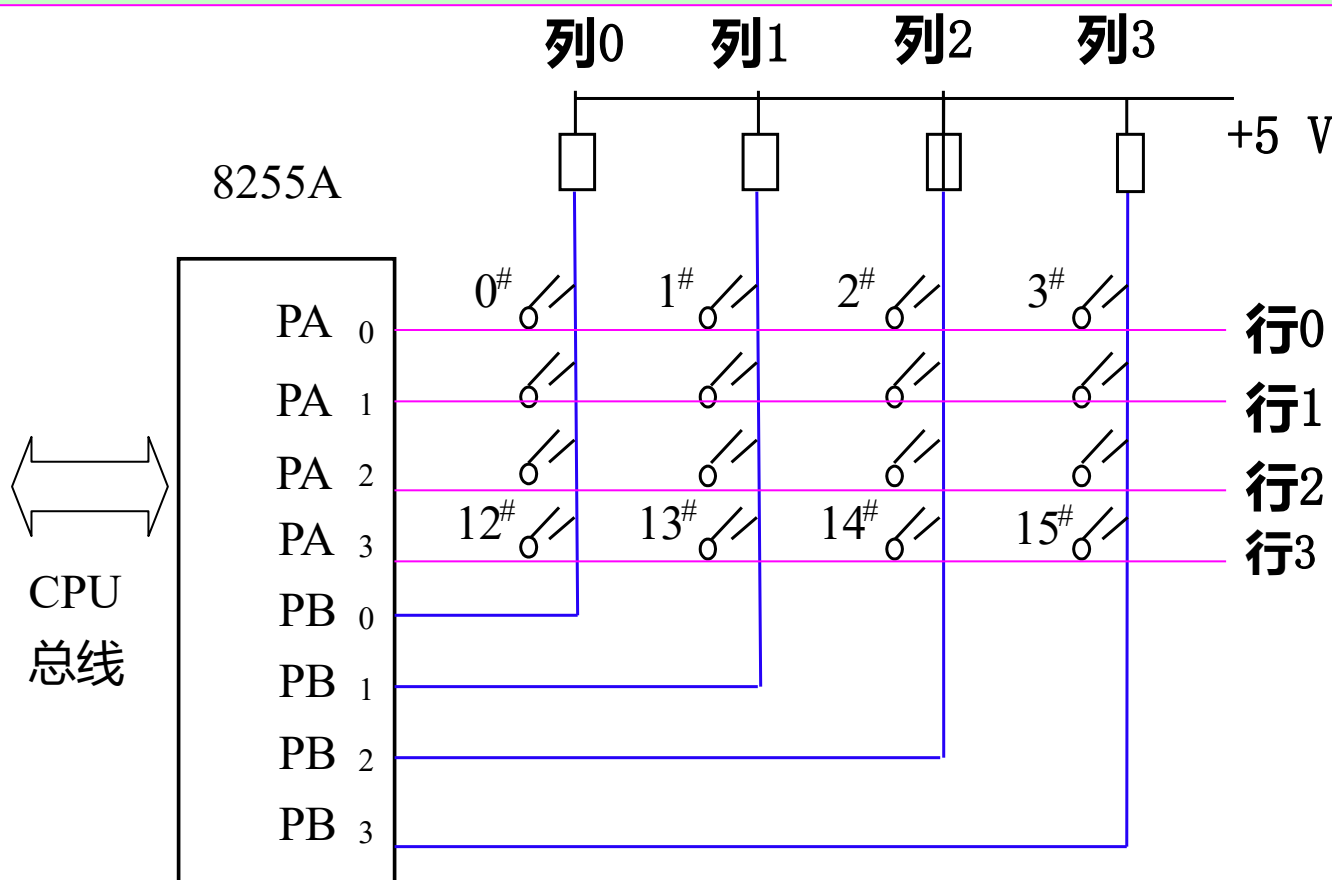


(2) 相应的8255A程序为:

```
MOV DX, 103H          ; 控制端口地址送给DX
MOV AL, 10010000B     ; 控制字
OUT DX, AL            ; 写入控制字
L1: MOV DX, 100H       ; 端口A地址送给DX
    IN AL, DX          ; 从端口A读入开关状态
    MOV DX, 101H       ; 端口B地址送给DX
    OUT DX, AL          ; 从端口B输出, 控制LED
    XOR AL, 0FFH       ; AL ← AL取反
    MOV DX, 102H       ; 端口C地址送给DX
    OUT DX, AL          ; 从端口C输出
    JMP L1             ; 循环
```

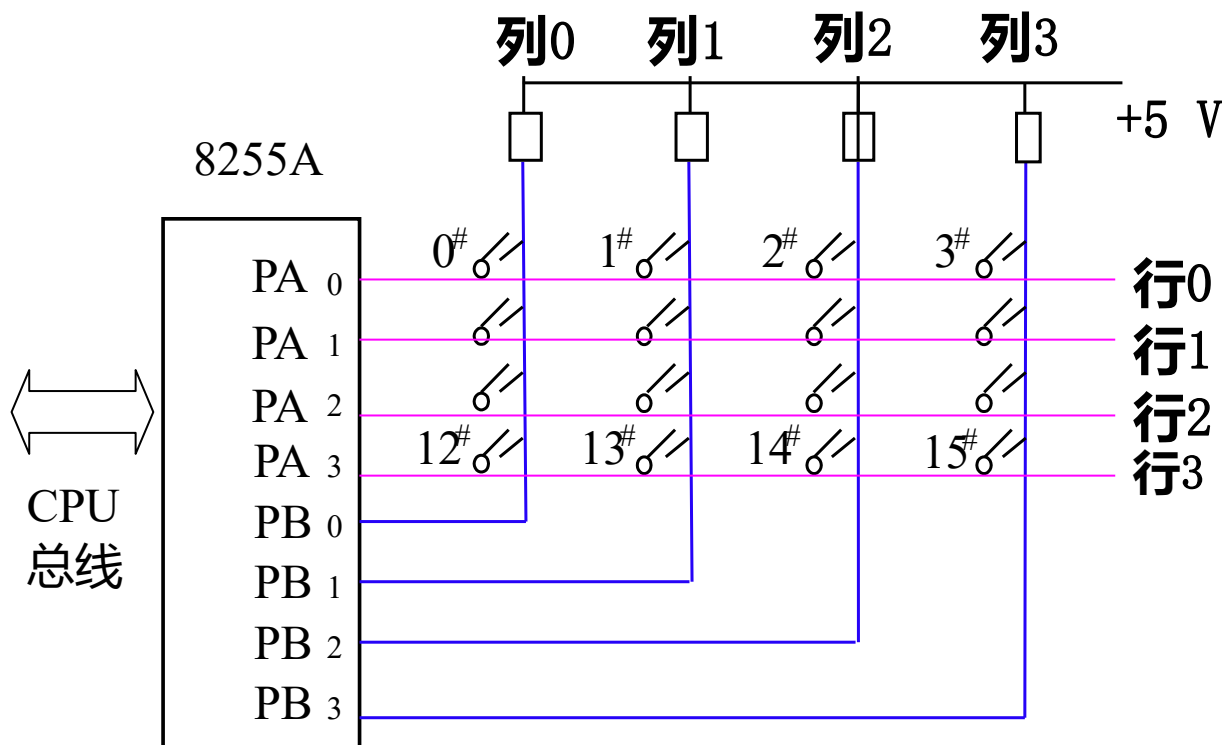
例3 用8255作为小键盘接口的设计

图中8255A的A口工作于方式0输出，B口工作于方式0输入。



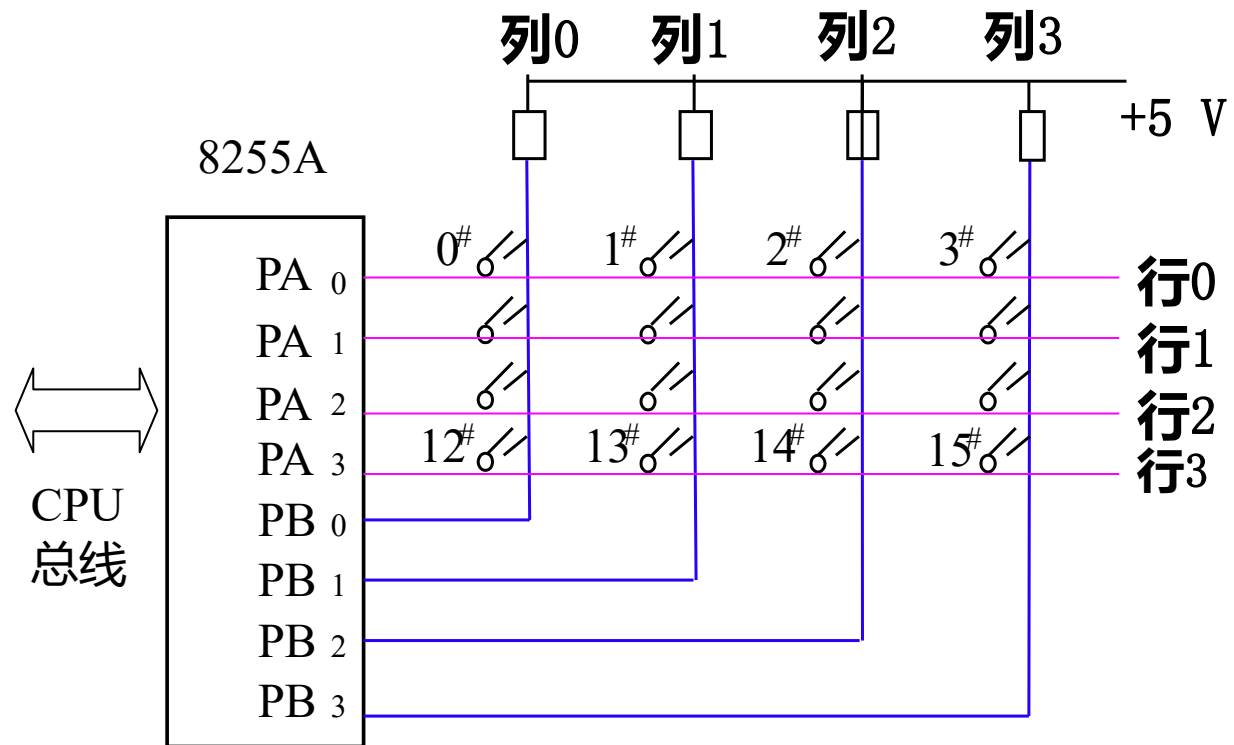
工作原理：使全部输出线为0，读回输入线的数据，如有键被按下，则肯定有一条线的数据（有一位）为0。

键盘工作过程：首先进行第1次键盘扫描（判断是否有键按下）。



①使A口PA3~PA0输出均为0，然后读入B口的值，查看PB3~PB0是否有低电平，若没有低电平，则说明没有键按下，继续进行扫描（重复过程①）。

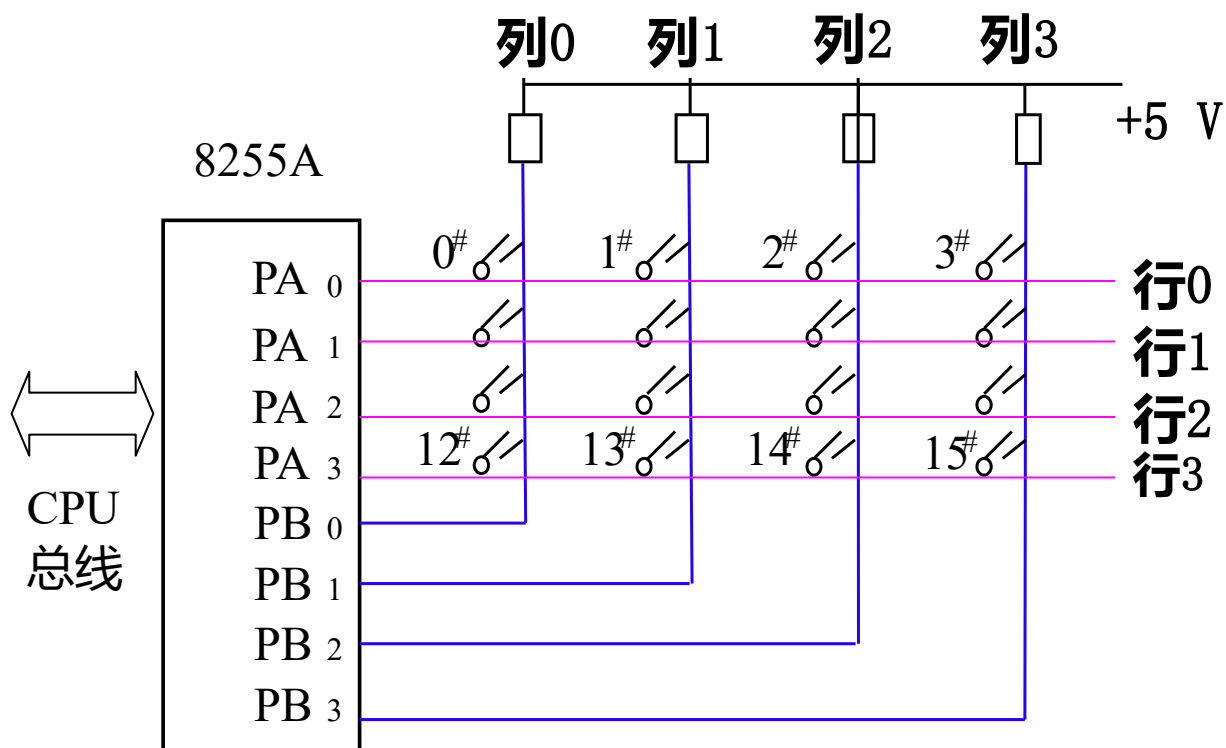
若PB3~PB0中有一位为低电平，使用软件延时10~20 ms以消除抖动，若低电平消失，则说明低电平是由干扰或按键的抖动引起的，必须再次扫描，否则，则确认有键按下，接着进行第2次扫描（行扫描，判断所按键的位置）。



②通过A口输出使 $PA_0=0$, $PA_1=1$, $PA_2=1$, $PA_3=1$ (逐位为0) 对第0行进行扫描, 此时, 读入B口的值, 判断 $PB_3\sim PB_0$ 中是否有某一位为低电平, 若有低电平, 则说明第0行某一系列上有键按下。

如果没有低电平, 接着使A口输出 $PA_0=1$, $PA_1=0$, $PA_2=1$, $PA_3=1$ 对第1行进行扫描。

按上述方法判断, 直到找到被按下的键, 并识别出其在矩阵中的位置, 从而可根据键号去执行该键对应的处理程序。



程序设计:

设8255A的A口、B口和控制寄存器的地址分别为80H、81H和83H。

```
MOV    AL, 82H    ; A口方式0输出, B口方式0输入
OUT    83H, AL    ; 将工作方式控制字送控制寄存器
MOV    AL, 00H
OUT    80H, AL    ; 使PA3=PA2=PA1=PA0=0
LOOA:  IN    AL, 81H    ; 读B口, 判断PB3~0是否有一位为低电平
AND    AL, 0FH    ; 只考虑低四位的值
CMP    AL, 0FH    ; 有键闭合吗?
JZ     L00A        ; 无键被按下, 转L00A等待
CALL   LD20ms      ; 有键被按下, PB3~PB0有一位为低, 调延
                        ; 时20 ms子程序, 消除抖动引起的误动作
IN     AL, 81H     ; 再次读入B口值。
```

AND AL, 0FH ; 有键按下, 在检查是否真的按下。
CMP AL, 0FH ; 如果延时后PB₃~PB₀中低电平不再存在,
JZ LOOA ; 说明是由干扰或抖动引起的, 则转
LOOA继续扫描, 延时后仍有某位为0, 说明有键被按下, 查键
值。

下段程序为判断哪一个键按下

START: MOV BL, 4 ; 行数送BL (共四行)
MOV BH, 4 ; 列数送BH (共四列)
MOV AL, 0FEH ; 准备扫描0行, 使D0位=0。
MOV CL, 0FH ; 键盘屏蔽码送CL (只要低四位)
MOV CH, 00H ; CH中存放起始键号
LOP1: OUT 80H, AL ; A口输出, 扫描一行
(D0=0)
ROL AL, 1 ; 行值左移一位, 修改扫描码, 准
; 备扫描下一行。
MOV AH, AL ; 暂时保存行线值
IN AL, 81H ; 读B口, 以便确定所按键的列值

AND AL, CL ; 比较是低四位, 只保留低四位
CMP AL, CL ; 检查该列线是否为0,
JNZ LOP2 ; 有列线为0, 转LOP2, 找列值
ADD CH, BH ; 无键按下, 修改键号, 检查下一行
MOV AL, AH ; 恢复扫描码
DEC BL ; 行数减1
JNZ LOP1 ; 行未扫描完转LOP1, 扫描下一行
MOV CX, 16 ; 16个键, 要找最多16次
LOP2: MOV SI, TABLE ; 取键码表值
CMP AX, [SI] ; 查找该键值
JZ LOP3 ; 找到该值, 到键处理子程序
INC SI
INC SI
LOOP LOP2

键值表:

ORG 1000H

TABLE: DW FE0E ; 0

键

DW FD0E ; 1键

DW FB0E ; 2键

DW F70E ; 3键

DW FE0D ; 4键

DW FE0D ; 5键

DW FD0D ; 6键

DW FB0D ; 7键

DW F70D ; 8键

DW FE0D ; 9键

;

;

键处理子程序:

LOP3: MOV AX, SI; AL内容=该键值

:

:

例4：

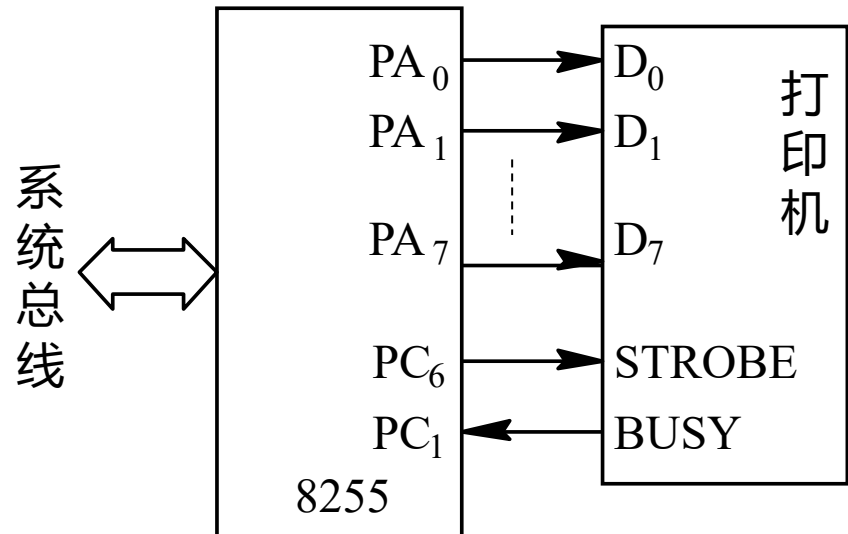
8255与打印机连接的硬软件设计

查询式，用C口的一些位做工作状态（灵活应用！）

在使用中打印机的很多控制线中只使用少数的几个控制信号就可以满足正确打印的要求，其它信号线可以不要。

STROBE：告诉打印机是否能发数据。

**BUSY：=1，数据被打
打印机取走，=0，数据
被打打印机取走。**



8255与打印机的连接

打印程序如下:

```
INIT55: MOV DX, 0383H; 控制口地址
        MOV AL, 13H ; A方式0、输出、B方式0, 输入
        OUT DX, AL   ; C7-4输出、C3-0输入
        MOV AL, 0DH  ; C6置1
        OUT DX, AL   ; 给打印机一个接受数据信号
PRINT:  MOV AL, BLAK ; BLAK为要打印的数据长度
        MOV CL, AL; 数据长度送到CL保存
        MOV SI, OFFSETDATA1 : 取打印数据存放地址
GOON:   MOV DX, 0382H ; 设C口地址
PWAIT:  IN  AL, DX    ; 取回C口的内容
        AND AL, 02H; 判断C1位, 打印是否准备好
        JNZ PWAIT    ; C1=0 ZF=0 则转去继续查询等待
        MOV AL, [SI] ; 准备好, 取一个字符
        MOV DX, 0380H; 送打印字给A口
        OUT DX, AL   ; 送数据
```

```
MOV DX, 0383H
MOV AL, 0CH:  C6置0, 使该位有跳变
OUT DX, AL ; 按位置位控制字
MOV AL, 0DH ; C6置1, 给打印机一个接受数据信号
OUT DX, AL ; 送脉冲
INC SI ; 打印数据区地址移到下一个数据位置
DEC CL ; 数据长度减1
JNZ GOON ; 全部数据打印完吗? 没完返回继续打印
RET ; 全部数据打印完, 返回主程序
```

在此例中，为了保证打印机正确的接收到数据，先给出一个查询信号STROBE，在打印机接到这个信号后并且处在“闲”状态时，就回答给出BUSY=1，CPU就可以将要打印的数据送到A口，同时发出STROBE信号，告诉打印机，要发送下一个数据。



例5：采用8255A进行双机并行通信

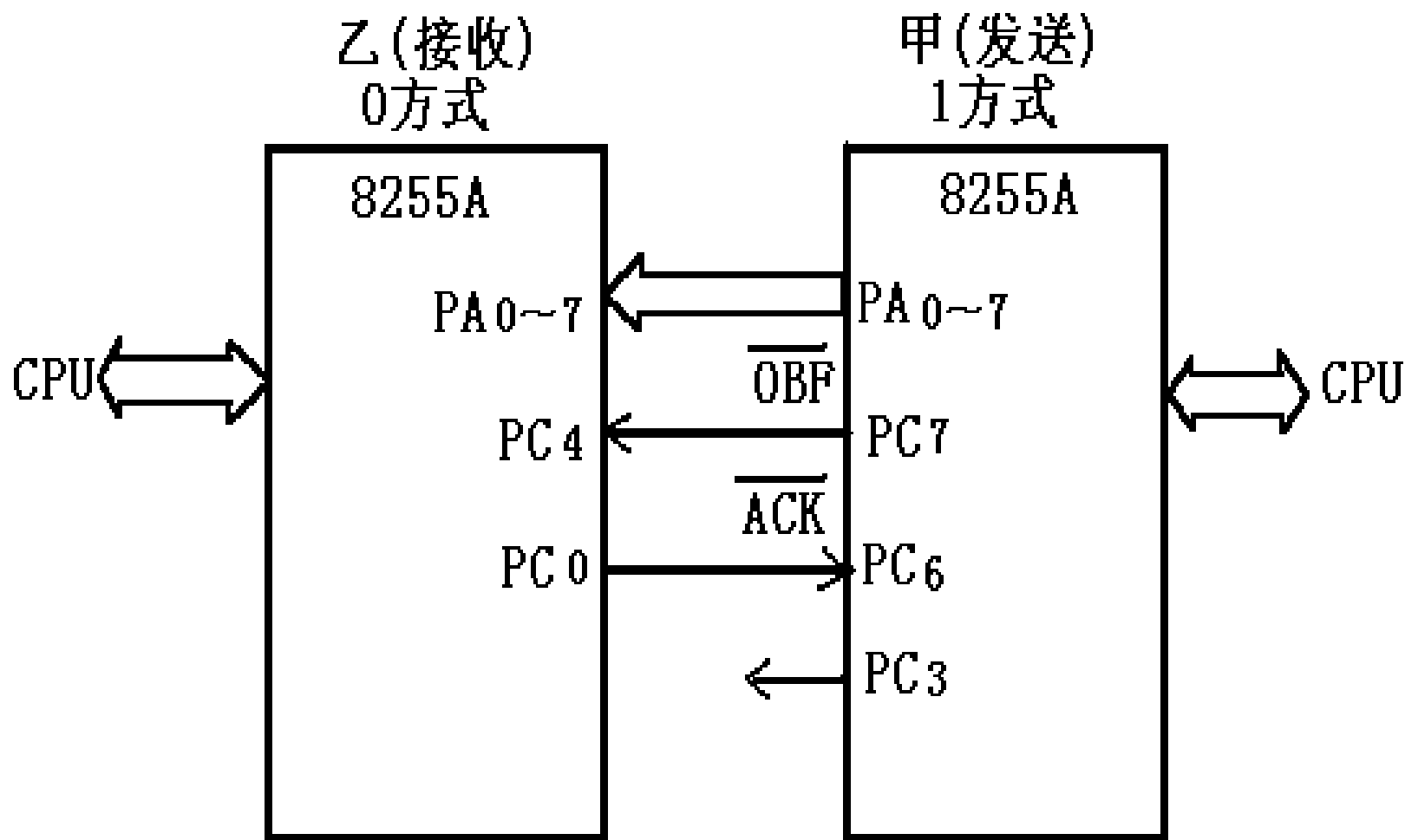
要求在甲乙两台微机之间并行传送1K字节数据。两机的CPU与接口之间都采用查询方式交换数据。

1. 甲机发送, 1方式, PA口为输出, PC7和 PC6引脚分别固定作联络线OBF和ACK

2. 乙机接收。0方式, PA口为输入, PC4和 PC0作联络线。

虽然, 两侧的8255A都设置了联络线, 但有本质的差别: 甲机8255A是1方式, 其联络线是固定的不可替代; 乙机的8255A是0方式, 其联络线是不固定的可选择, 比如可选择PC4和PC1或PC3、PC2等任意组合。

双机并行传送接口电路框图



分别对两个系统进行程序设计

甲机发送程序:

MOV DX,0303H ; 8255A命令口

MOV AL,10100000B ; 初始化工作方式字

OUT DX,AL

MOV AL,0DH ; 置发送中断允许INTEA=1

OUT DX,AL ; PC6=1

MOV AX,0030H ; 发送数据内存首址

MOV ES,AX

MOV BX,00H ;

MOV CX,0400H ; 发送字节数

MOV DX,0300H ; 向A口写第一个数,产生第一个OBF信号

MOV AL,ES:[BX] ; 送给对方, 以便获取对方的ACK信号

OUT DX,AL

INC BX ; 内存加1

DEC CX ; 字节数减1

L: MOV DX,0302H ; 8255A状态口

IN AL,DX ; 查发送中断请求 INTRA=1?

AND AL, 08H ; PC3= 1?

JZ L ; 若无中断请求, 则等待; 有中断请求, 则向A口写数。

MOV DX,0300H ; 8255A PA口地址

MOV AL, ES:[BX] ; 从内存取数

OUT DX, AL ; 通过A口向乙机发送第一个数据

```
INC  BX      ; 内存地址加 1
DEC  CX      ; 字节数减1
JNZ  L       ; 字节未完, 继续
MOV  AX, 4C00H ; 已完, 退出
RET
```

在上述发送程序中，是查状态字的中断请求 INTR位（PC3），实际上，也可以查发送缓冲器满OBF（PC7）的状态, 只有当发送缓冲器空时CPU才能送下一个数据。

乙机接收程序:

```
MOV DX,0303H      ; 8255A命令口
MOV AL,10011000B   ; 初始化工作方式字
OUT DX,AL
MOV AL,00000001B    ; 置ACK=1 (PC0=1)
OUT DX, AL
MOV AX,0040H        ; 接收数据内存首址
MOV ES, AX
MOV BX, 0000H
MOV CX, 0400H        ; 接收字节数
L1: MOV DX, 302H      ; 255A PC口
IN AL,DX             ; 查甲机的OBF=0? (PC4=0)
AND AL,10H           ; 即查甲机是否有数据发来
```

JNZ L1 ; 若无数据发来, 则等待; 若有数据, 则从A口读数
MOV DX, 300H ; 8255A PA口地址
IN AL,DX ; 从A口读入数据
MOV ES:[BX],AL ; 存入内存
MOV DX,303H ; 产生ACK信号. 并发回给甲机
MOV AL,00000000B ; PC0置 "0"
OUT DX, AL
NOP
MOV AL, 00000001B ; PC0置 "1"
OUT DX, AL ;
INC BX ; 内存地址加1
DEC CX ; 字节数减1
JNZ L1 ; 字节未完, 则继续
MOV AX, 4C00H ; 已完, 退出



谢谢大家！