


# 第8章 定时计数控制接口



## 8.0 定时与计数技术概述

✧ 计算机系统中经常要用到定时信号

如定时检测、定时扫描和时钟定时等；

✧ 许多场合还需要对脉冲信号进行计数

例如，产品包装流水线上对产品计数，将若干个小包装组合为大的出厂包装。

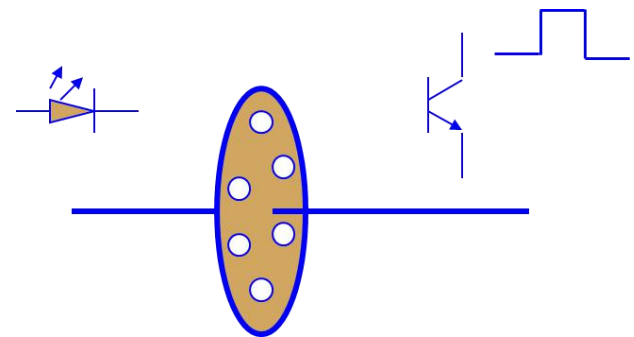
✧ 计数器与定时器的工作方式有不少相似之处，可以通过设计，用同一个电路实现定时器和计数器的功能。

## 8.0 定时与计数技术概述

### 1. 计数

在用计算机处理计数问题时，首先要将被计数的量转化为电脉冲的形式，于是计数任务就转化为对电脉冲的出现次数进行计数。引发计数的电脉冲称为计数脉冲。

在进行计数时，计数脉冲的出现与否受外界控制，不具有有一定的规律性，所以要求计数系统具有良好的实时性，即无论计数脉冲出现的快慢，计数系统都必须能准确记录计数脉冲的出现。



## 8.0 定时与计数技术概述

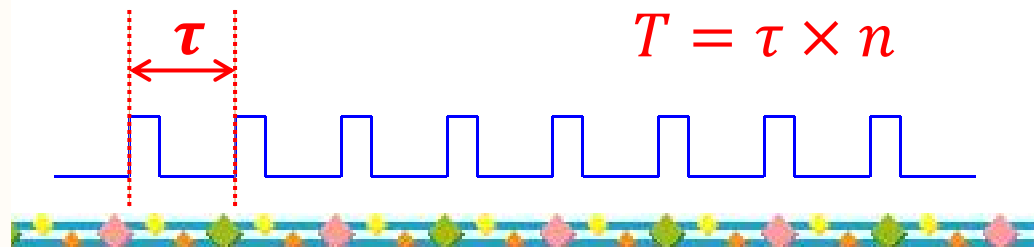
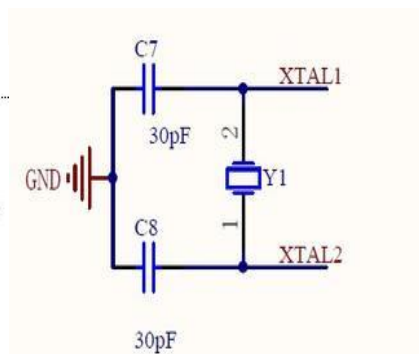
### 2. 定时

在计算机系统中，更常用的是定时工作。

计时：是指对两个事件发生的时间间隔进行计量，如比赛计时；

延时：是指在某事件发生后开始进行时间计量，到某个时间间隔后触发另一事件，如进程切换，DRAM定时刷新等。

时间计量是在统一的时间标准上进行的，这个时间标准通常由振荡器一类的电路提供的恒定频率的电脉冲来担当，这样定时的工作就可转化为对具有恒定频率的计数脉冲进行计数的工作了。

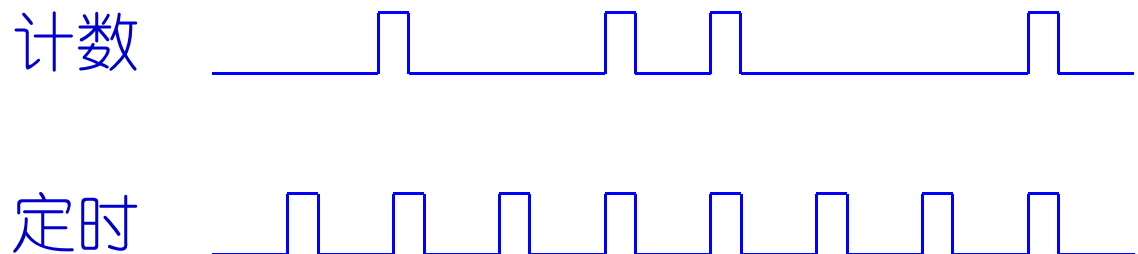


## 8.0 定时与计数技术概述

### 3. 定时与计数的联系

✧定时与计数本质上是一致的。

✧计数的信号随机，定时的信号具有周期性



## 8.0 定时与计数技术概述



### 4. 定时方法

定时方法通常有以下三种：

- (1) 软件定时
- (2) 不可编程的硬件定时
- (3) 可编程的硬件定时



## 8.0 定时与计数技术概述

### 4. 定时方法

(1) 软件定时 *for(int i=0;i<2000;i++);*

所谓“软件定时”是执行一个循环程序，通过“延时”来实现定时，时间的长短通过循环次数和循环嵌套层数来调节。这种方法不需要专用的硬件，简单、灵活。但是，软件延时要占用**CPU**时间，降低了**CPU**的效率，定时精度不高。软件延时通常用在延时时间不长，精度要求不高的场合。

```
delay proc
```

```
    push cx
```

```
    mov cx,400H
```

```
Lp:   loop lp
```

```
    pop cx
```

```
    ret
```

```
delay endp
```

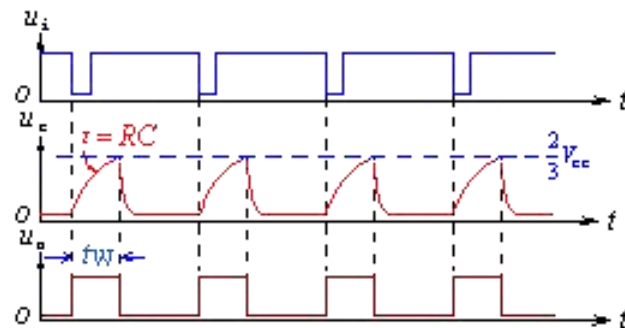
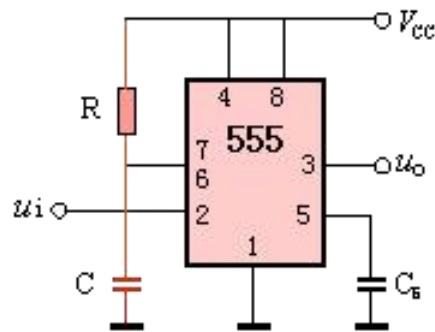
## 8.0 定时与计数技术概述

### 4. 定时方法

#### (2) 不可编程的硬件定时

采用计数器等元件组成专用的计时电路实现定时。

这种方法不占用**CPU**时间，电路也不复杂。缺点是灵活性差，在电路连接好后，定时时间和范围就不能改变。





## 8.0 定时与计数技术概述



### 4. 定时方法

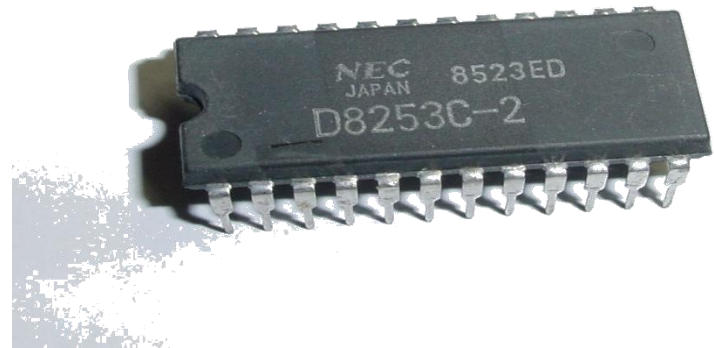
#### (3) 可编程的硬件定时

用大规模集成电路构成的可编程定时器/计数器，定时时间可以通过软件来设置。对芯片设置初值后，计数器开始工作，**CPU**就可以去做其他工作，定时时间到，电路会产生一个中断请求信号通知**CPU**。这种方法定时精确，使用方便，灵活性大，因而得到广泛应用。



## 8.1 可编程定时器/计数器8254

本节介绍 **Intel 8254** 定时/计数器芯片，它是一种能够完成上述功能的可编程器件。早期的**PC**机中使用 **Intel 8253** (2MHz) 作为系统的定时 / 计数器，现代微机中采用的 **Intel 8254** (10MHz)是它的增强型。

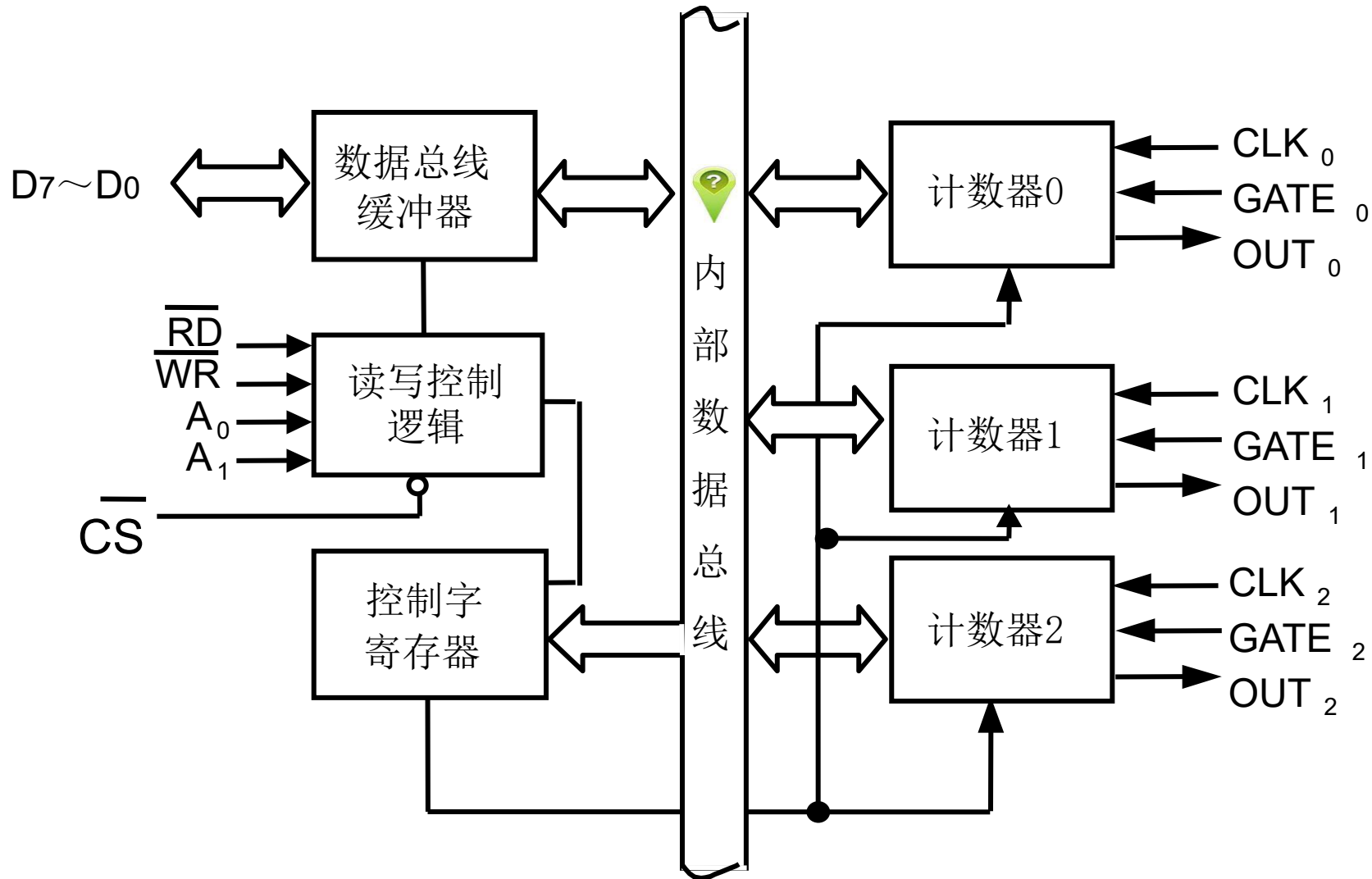


## 8.1 可编程定时器/计数器8254

### 8254的主要特性

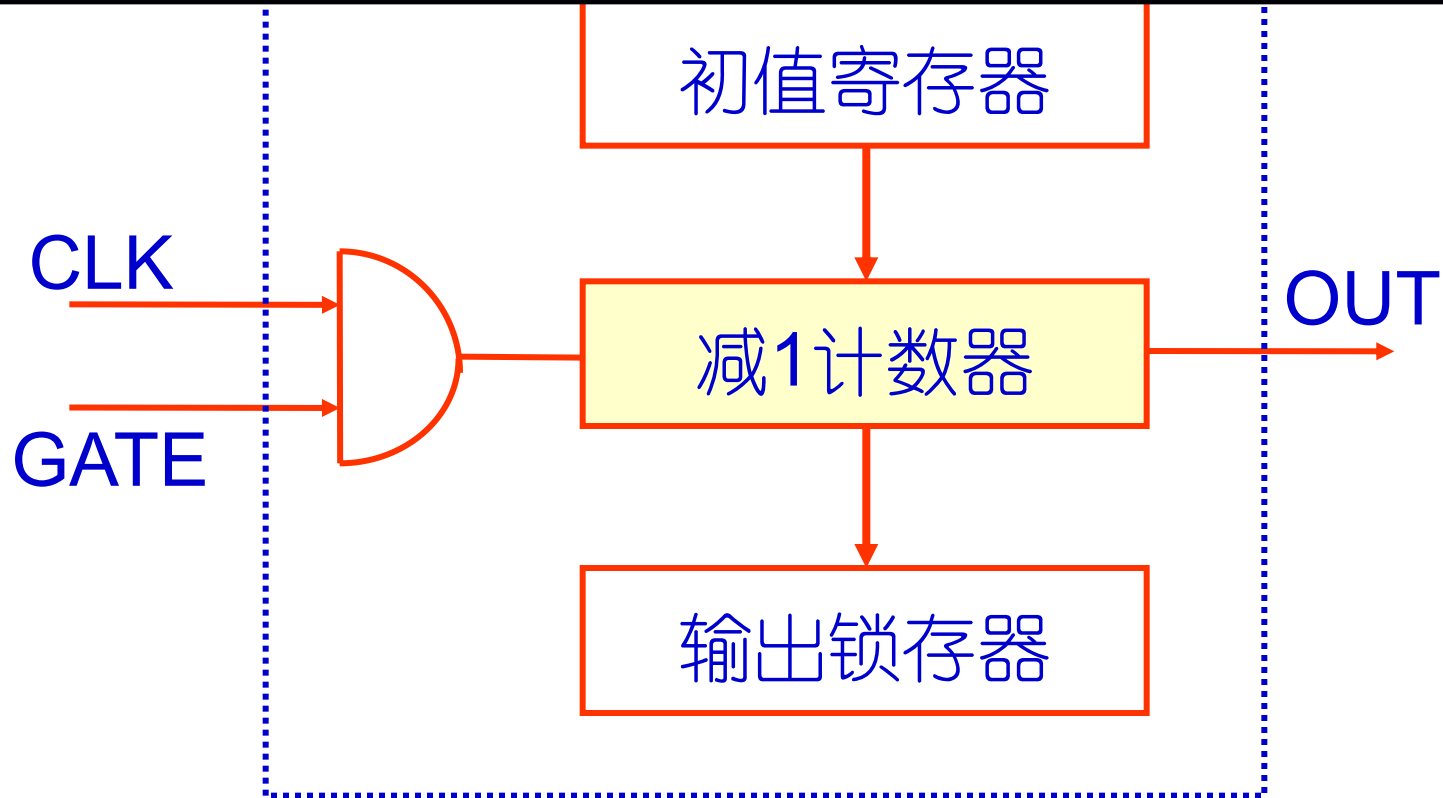
- (1) 每片8254内部有3个独立的16位定时/计数通道。
- (2) 每个通道有6种工作方式，既能够实现精确定时，又可以对外部脉冲进行计数。
- (3) 8254的最高计数频率为10MHz（8253为2MHz）。
- (4) 可以按二进制或BCD码两种方式计数。
- (5) 定时的时间长短用软件设置，可由软件或硬件控制开始计数或停止计数。
- (6) 可以同时锁存1~3个计数器的计数值和状态值，供CPU读取（8253每次只能锁存和读取一个通道的计数值）。

## 8.1.1 8254的内部结构

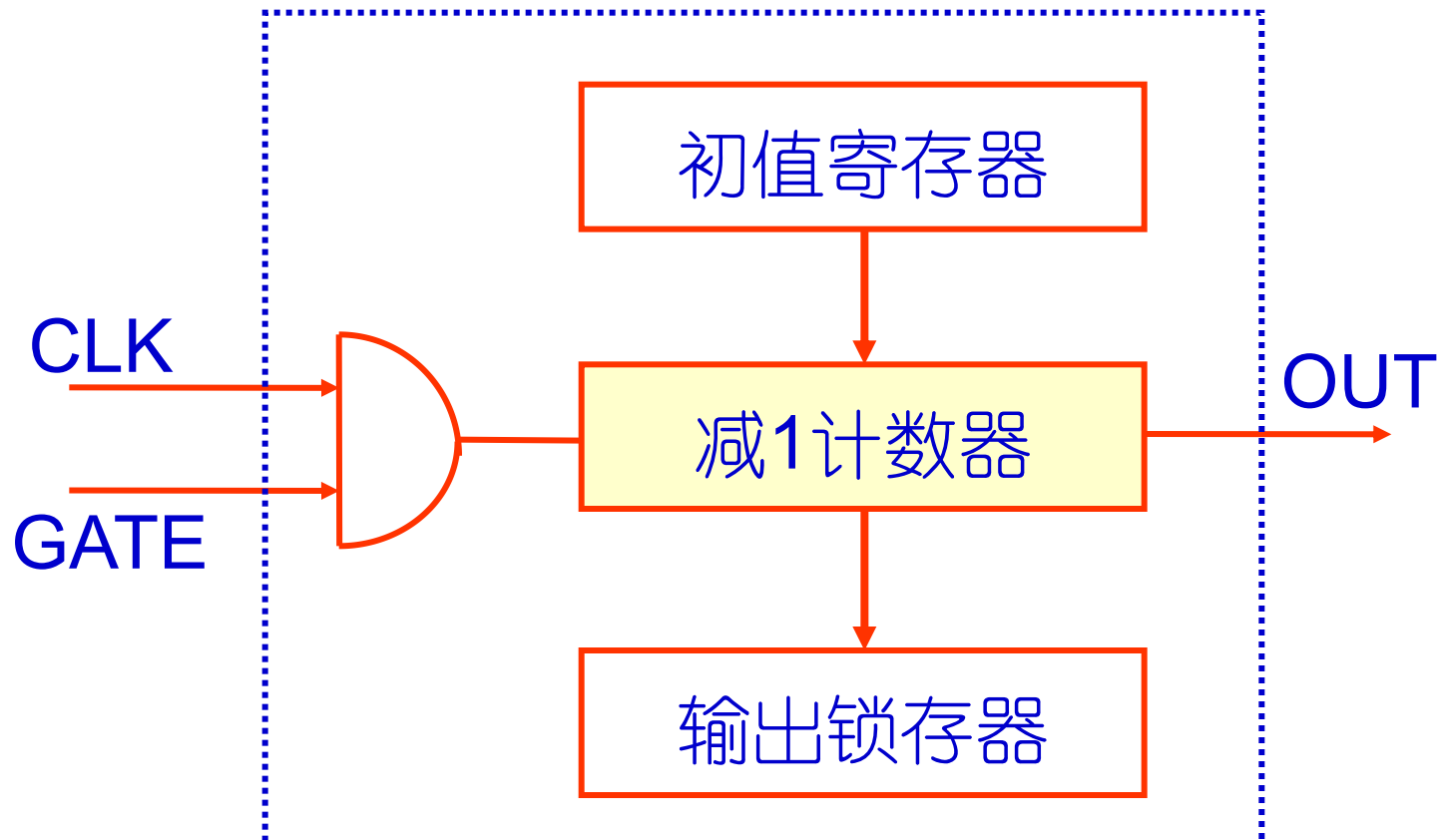


# 1. 计数器结构示意图

**CLK:** 时钟输入信号，在计数过程中，此引脚上每输入一个时钟信号（下降沿），计数器的计数值减1。



# 1. 计数器结构示意图

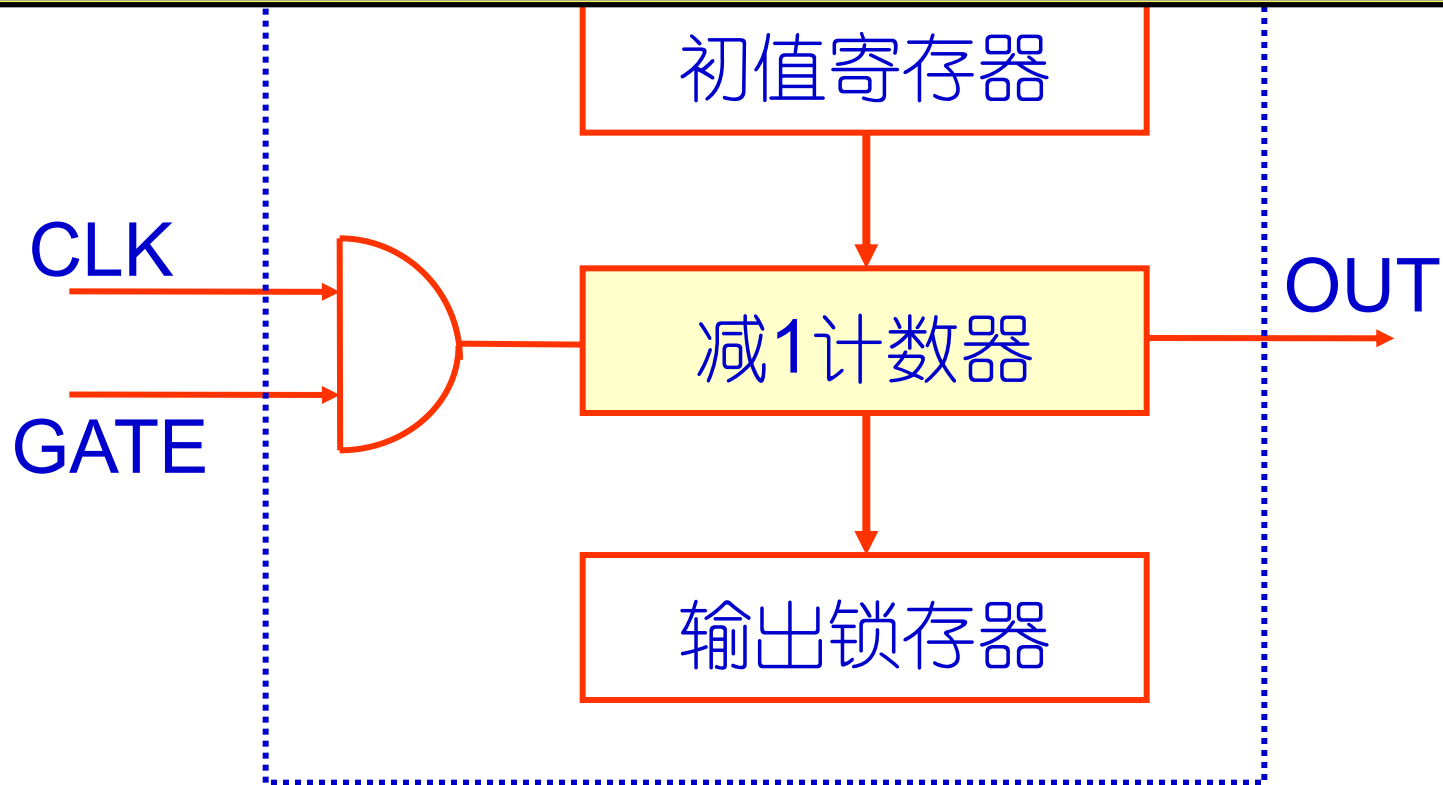


**GATE:** 门控输入信号，控制计数器工作，可分成电平控制和上升沿控制两种类型。



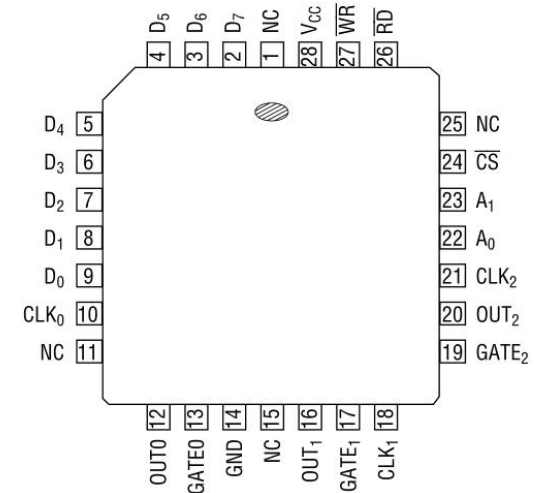
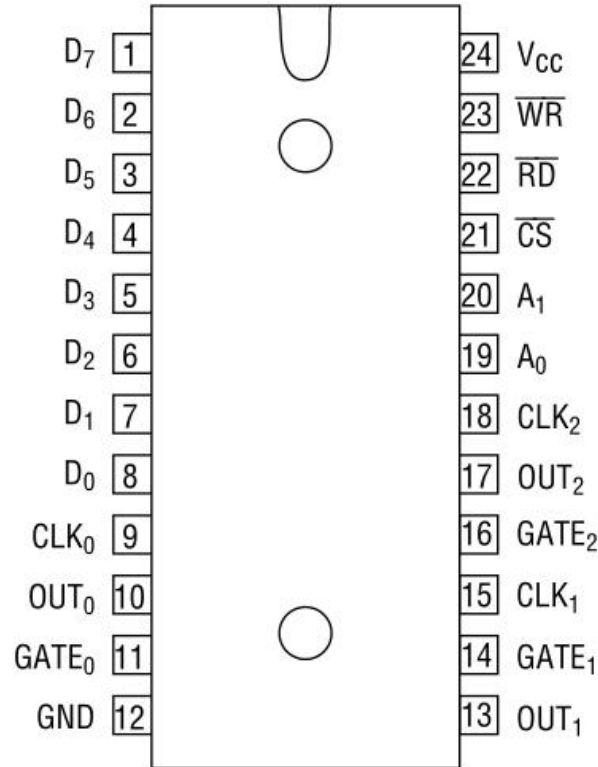
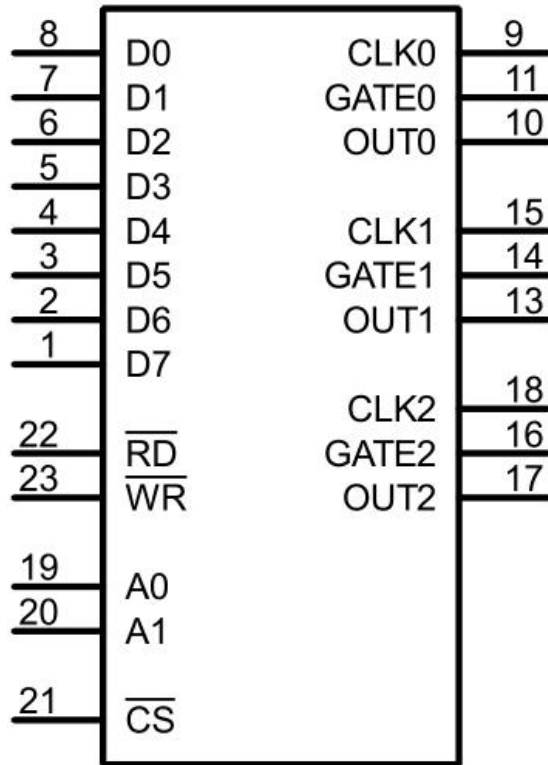
# 1. 计数器结构示意图

**OUT:** 计数器输出信号，当一次计数过程结束（计数值减为0），OUT引脚上将产生一个输出信号。



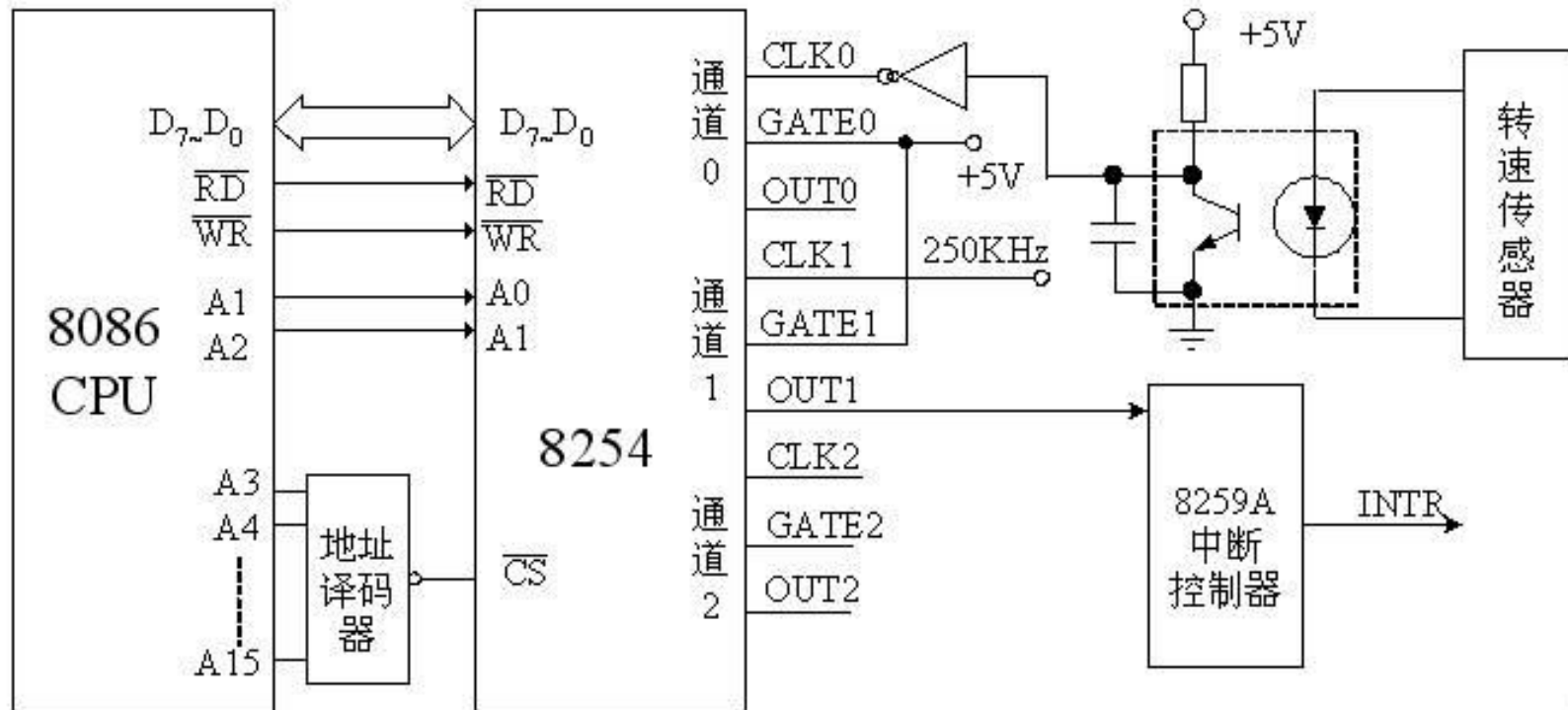
## 2. 8254 的引脚

Intel 8254共有24个引脚, 使用单一的+5V电源。





### 3. 8254 与处理器的接口



### 3. 8254 与处理器的接口

D<sub>0</sub> ~ D<sub>7</sub>数据线

A<sub>0</sub> ~ A<sub>1</sub>地址线

RD\*读信号

WR\*写信号

CS\*片选信号

CS* A1 A0	I/O地址	读操作RD*	写操作WR*
0 0 0	0H	读计数器0	写计数器0
0 0 1	1H	读计数器1	写计数器1
0 1 0	2H	读计数器2	写计数器2
0 1 1	3H	无操作	写控制字

## 8.1.2 8254的工作方式



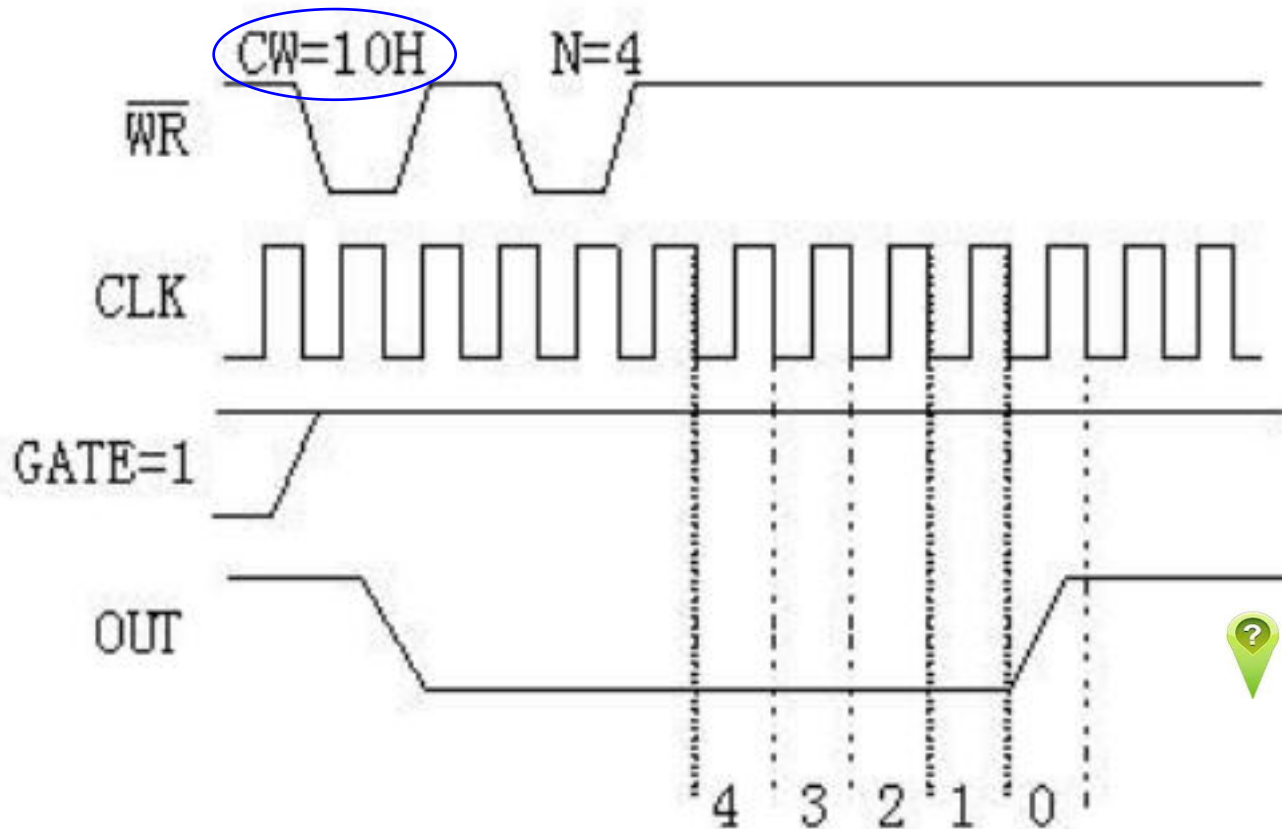
8254内部的每个定时器/计数器通道都有6种可编程选择的工作方式，分别用作定时器、计数器、单稳态触发器。

熟悉每种工作方式的特点，才能根据实际应用需求选择正确的工作方式。



## 8.1.2 8254的工作方式

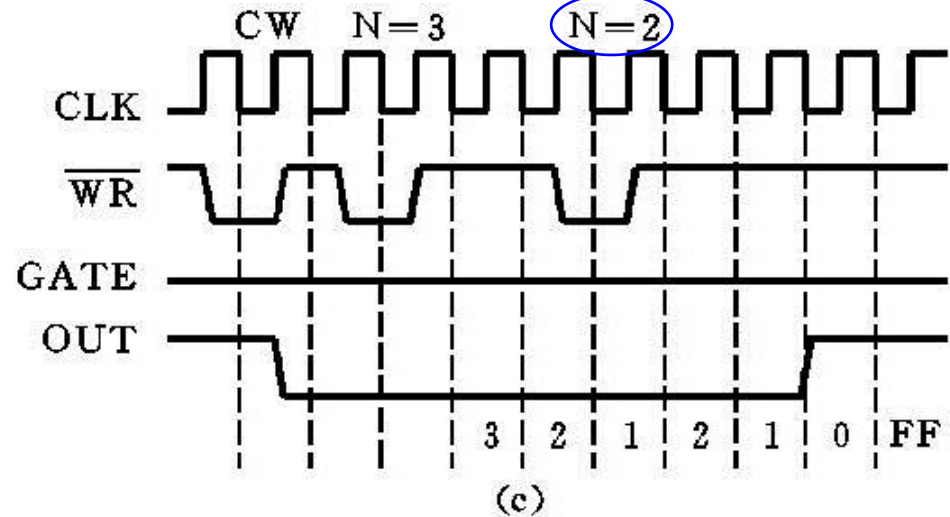
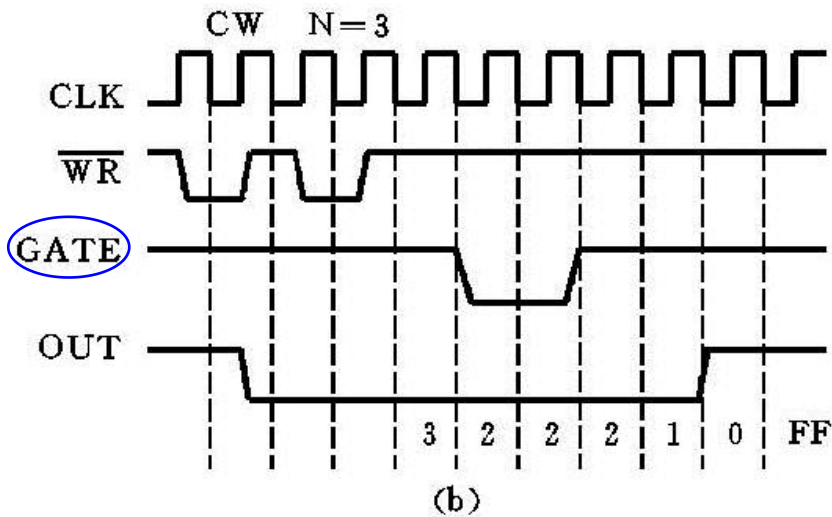
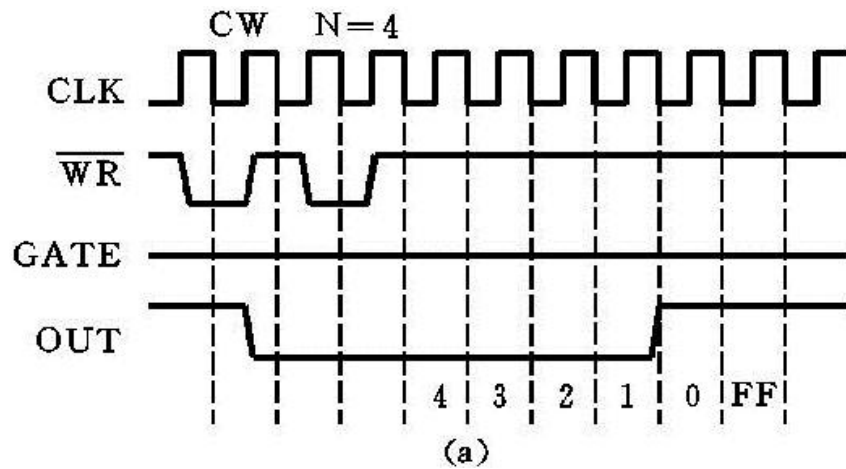
### 1. 方式0 – 事件计数器



计数结束  
OUT  
输出高  
电平

✧ 软件启动，减1计数；没有初值自动重装功能

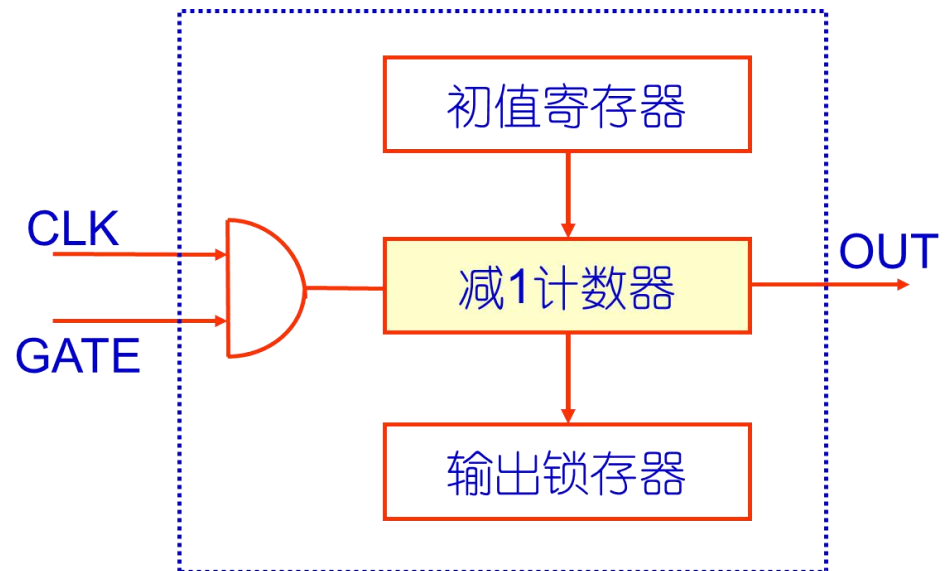
# 1. 方式0 – 事件计数器



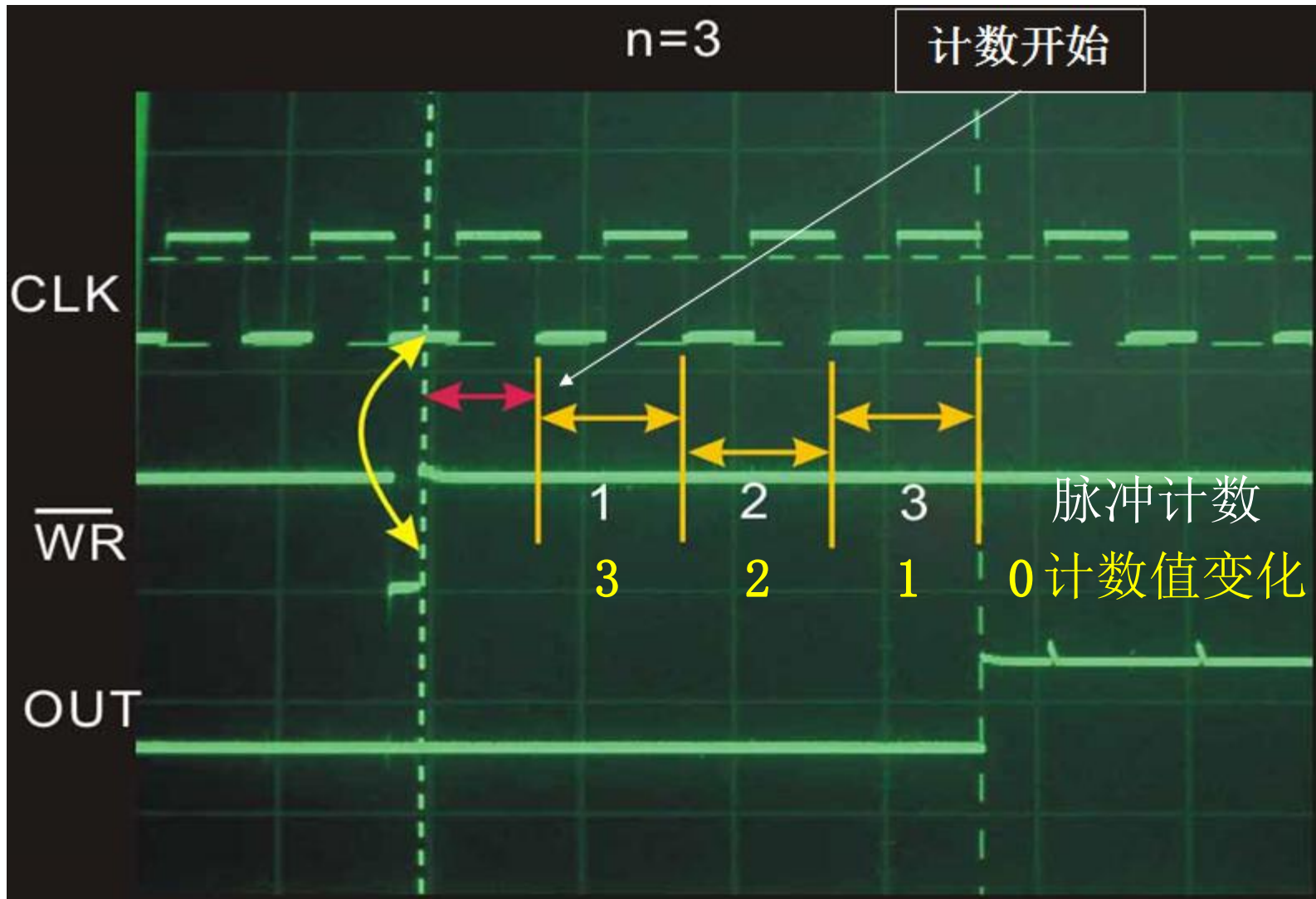
## 实验探究 计数开始的时刻

处理器写入**8254**的计数初值只是写入了初值寄存器，之后到来的第一个**CLK**输入脉冲才将计数初值送到减1计数器(该脉冲需先由低电平变高，再由高变低)。

从第二个**CLK**信号的下降沿，计数器才真正开始减1计数。

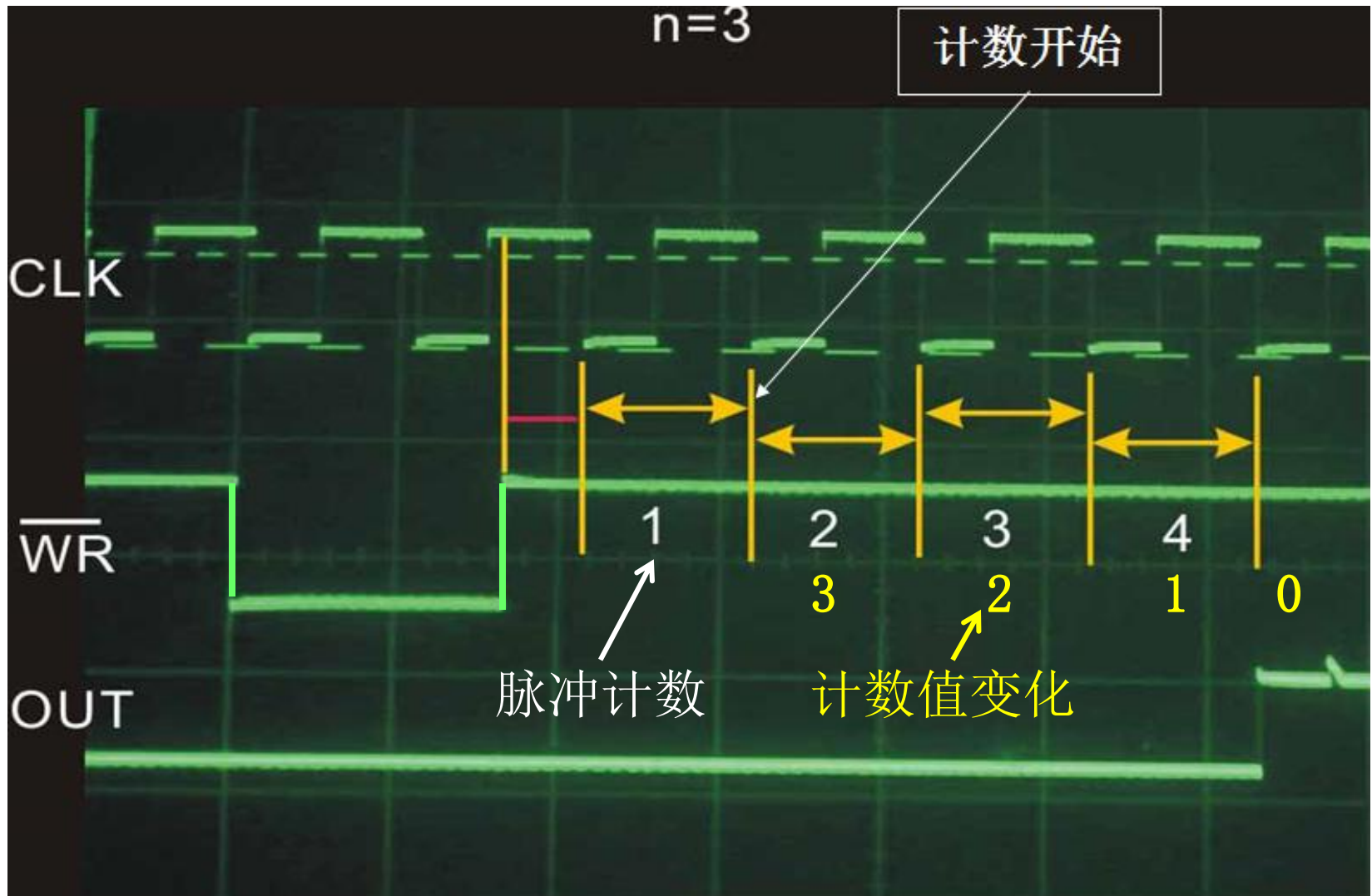


## 方式0的波形 1





## 方式0的波形 2

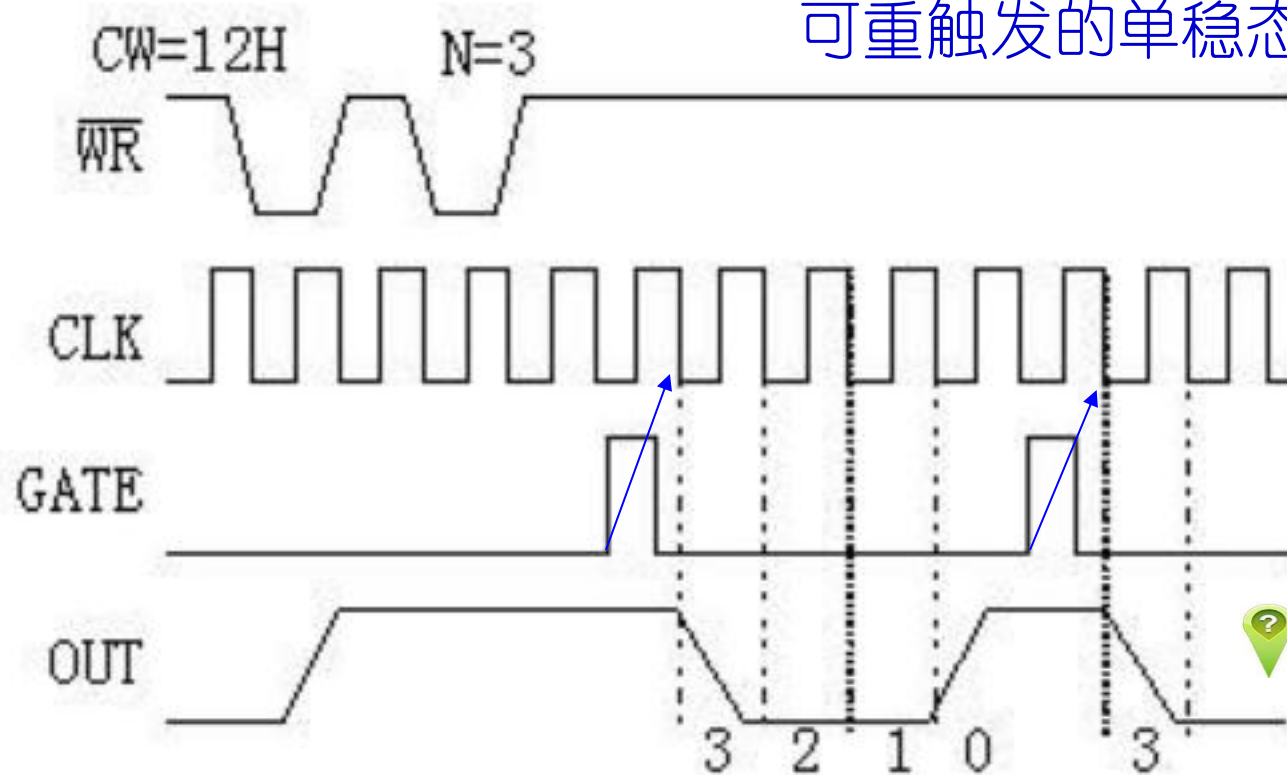




## 8.1.2 8254的工作方式

### 2. 方式1 – 硬件启动的单脉冲发生器

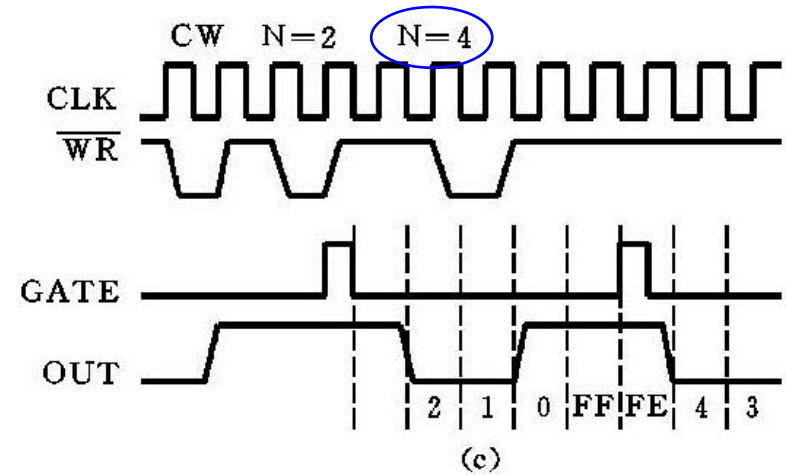
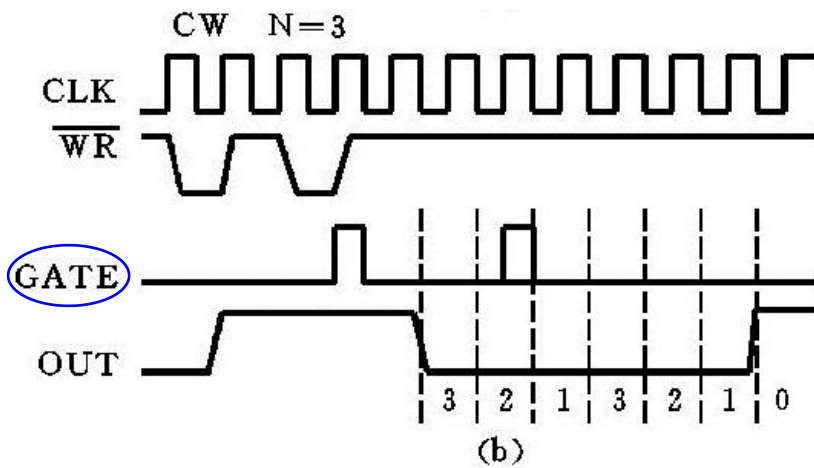
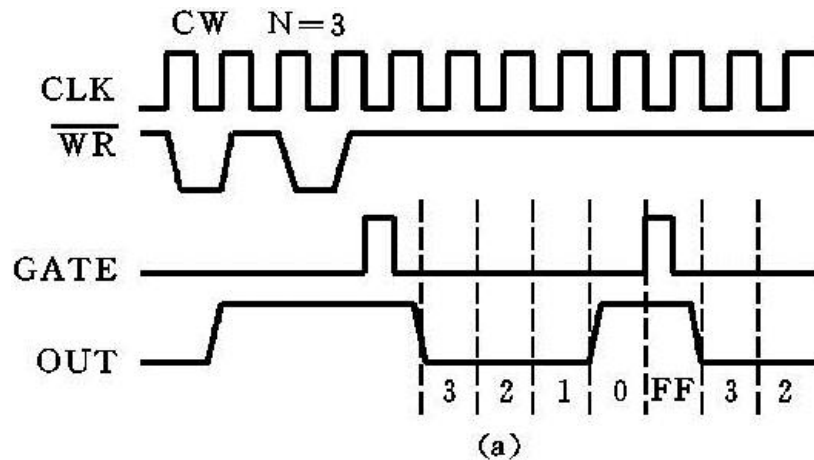
可重触发的单稳态触发器



✱ 硬件启动；可重触发；减1计数；无初值自动重装功能。

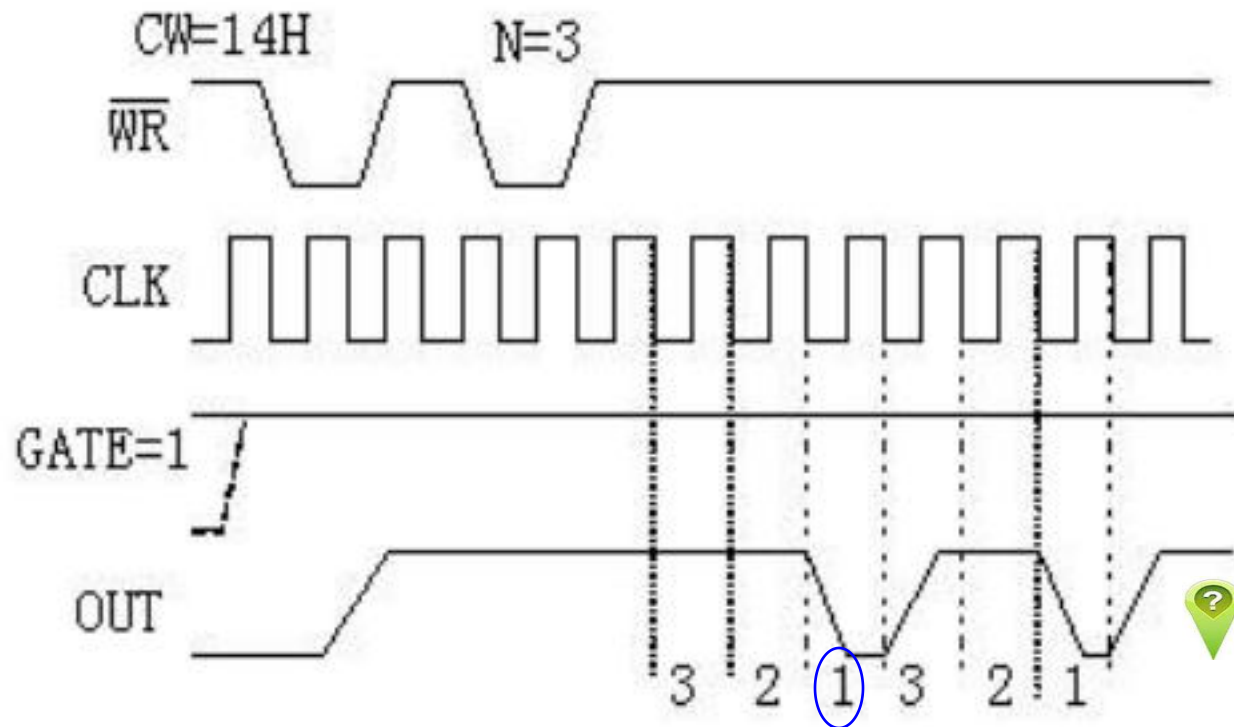
PWM

## 2. 方式1－硬件启动的单脉冲发生器



## 8.1.2 8254的工作方式

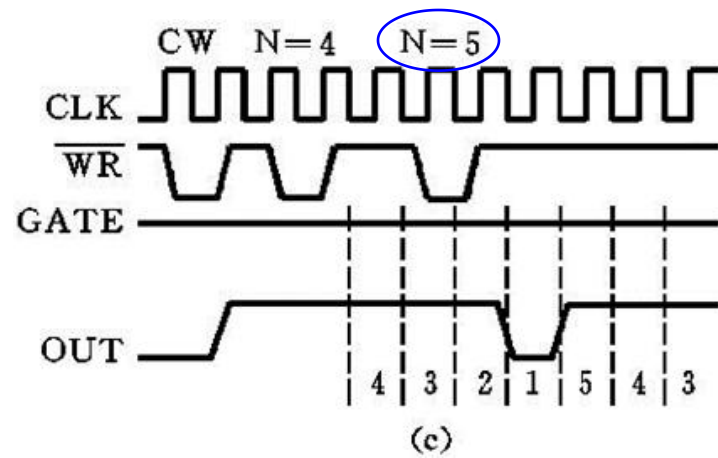
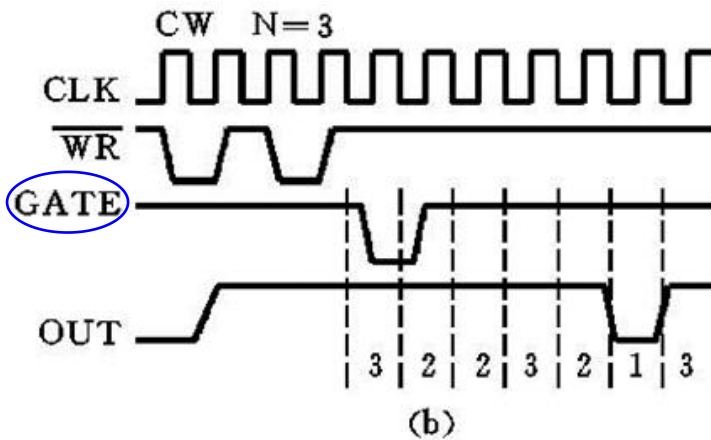
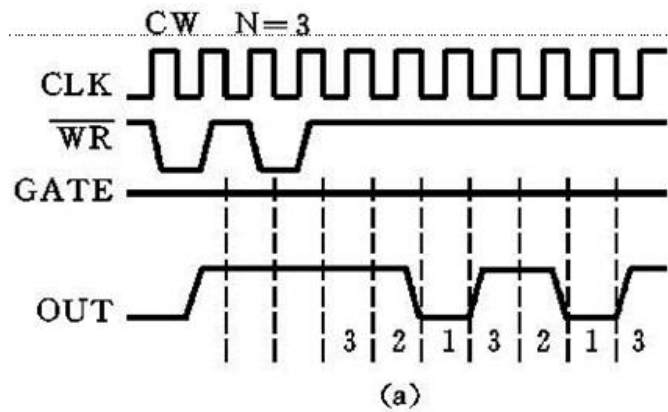
### 3. 方式2—脉冲发生器（频率发生器、分频器）



✱ 软件启动，减1计数，初值自动重装

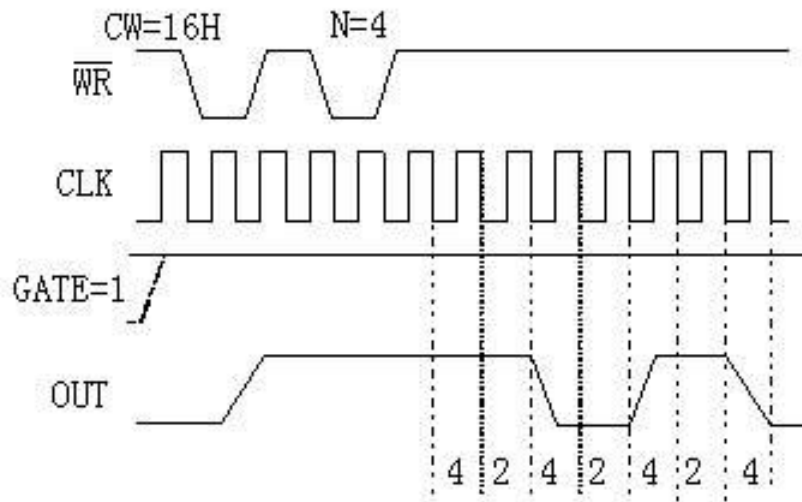
PWM

### 3. 方式2—脉冲发生器

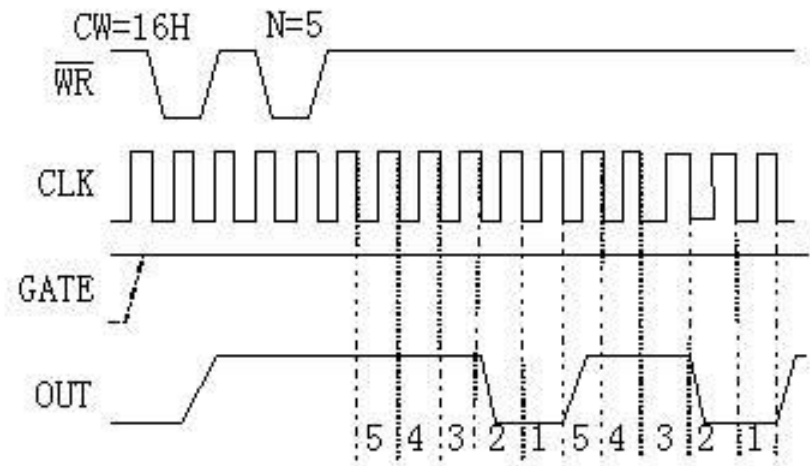


## 8.1.2 8254的工作方式

### 4. 方式3 – 方波发生器



(a) 计数初值为偶数

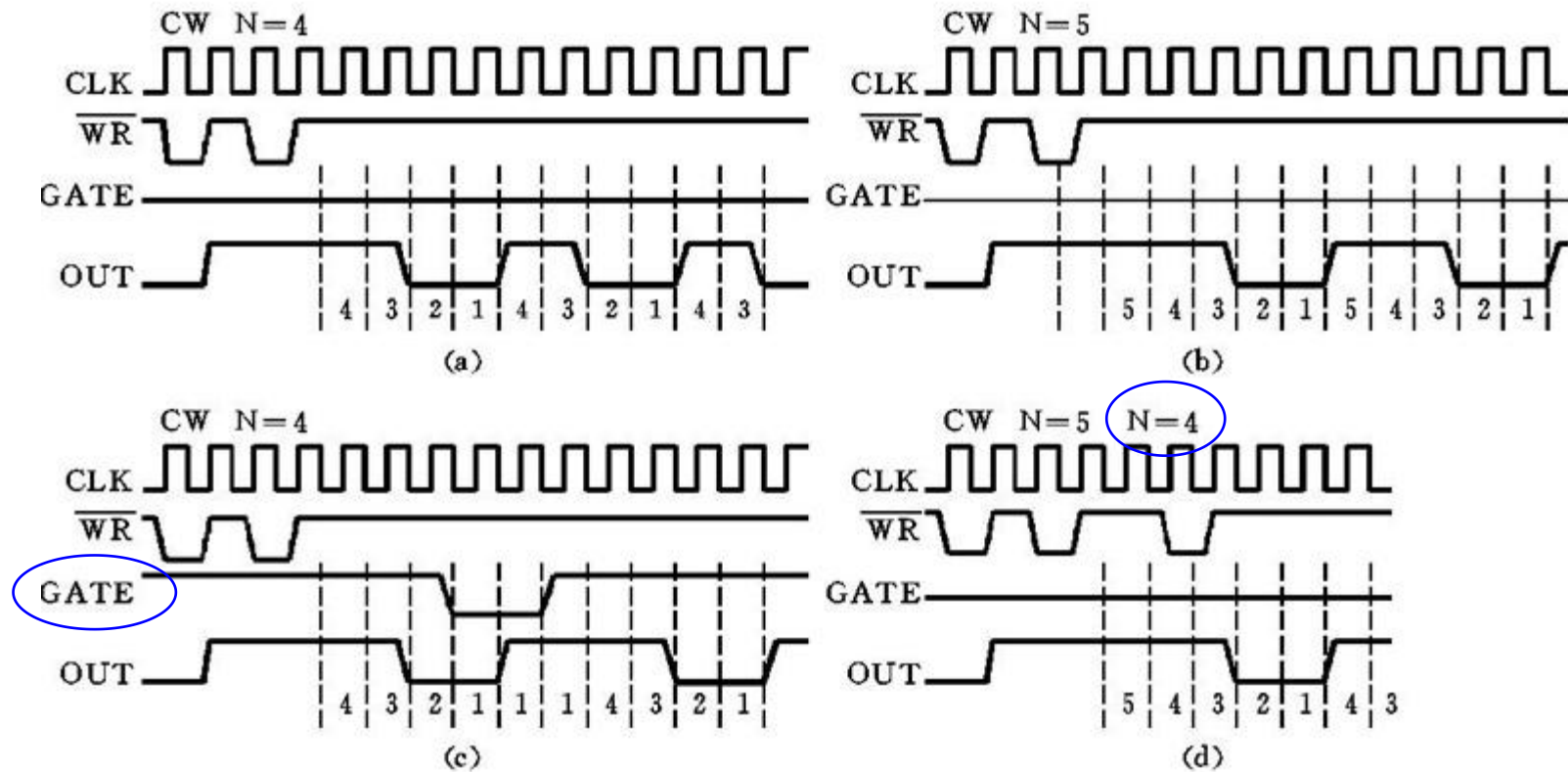


(b) 计数初值为奇数

✱ 软件启动；减2计数；有初值自动重装功能

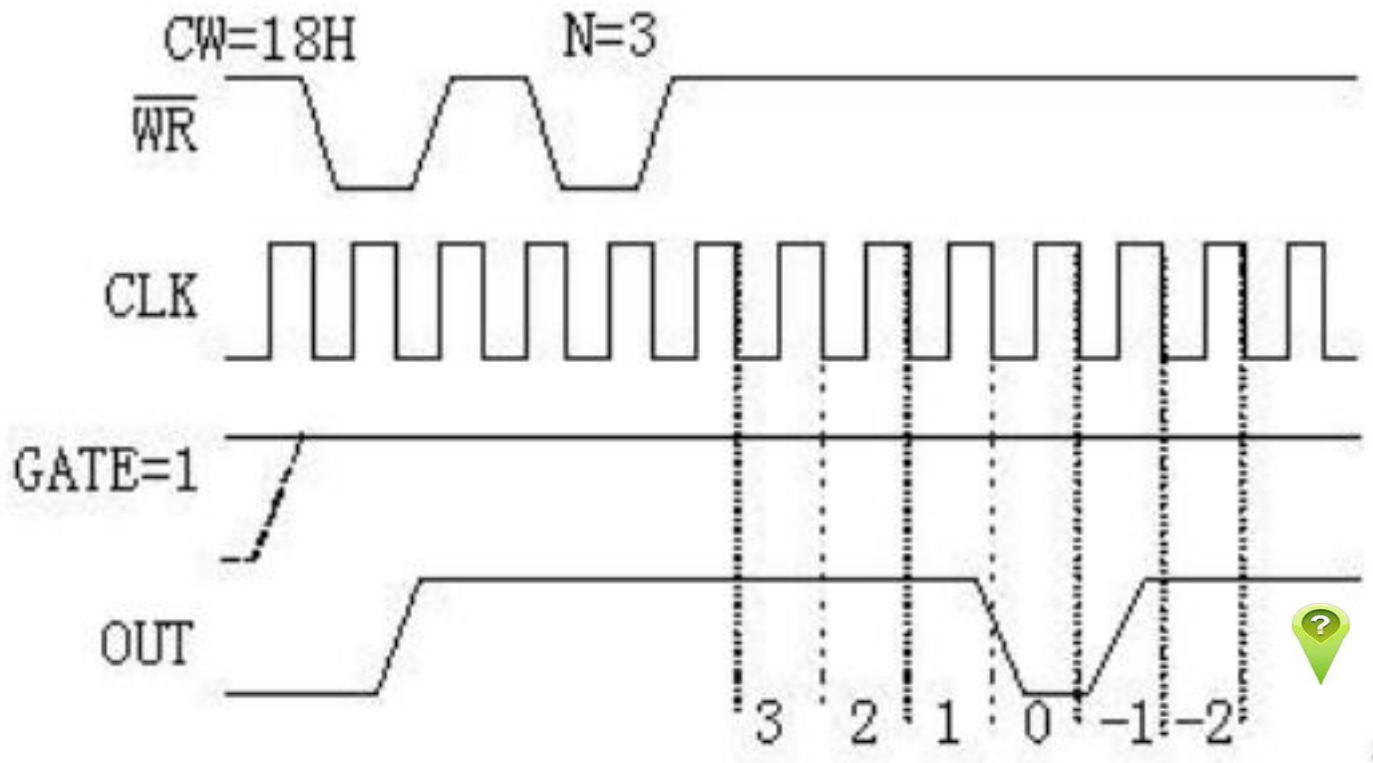
## 8.1.2 8254的工作方式

### 4. 方式3－方波发生器



## 8.1.2 8254的工作方式

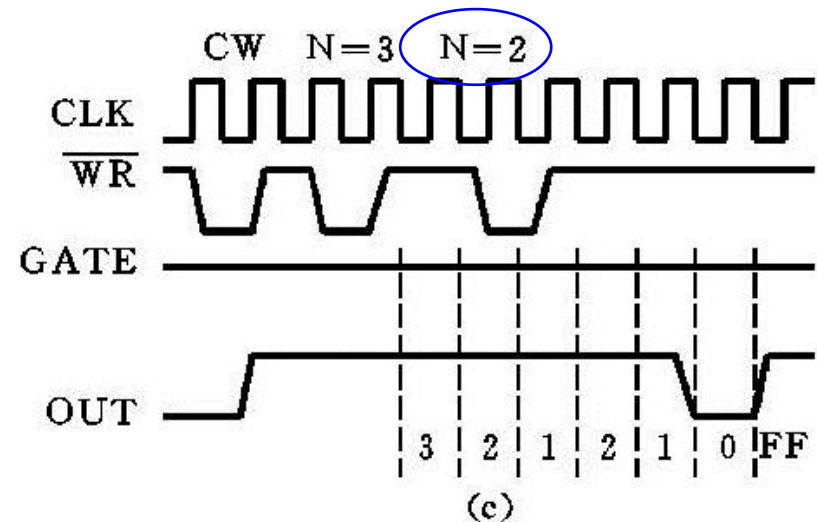
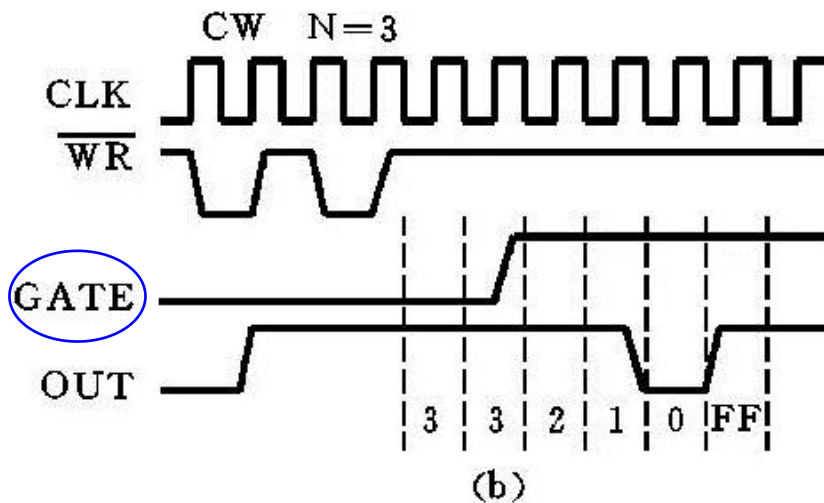
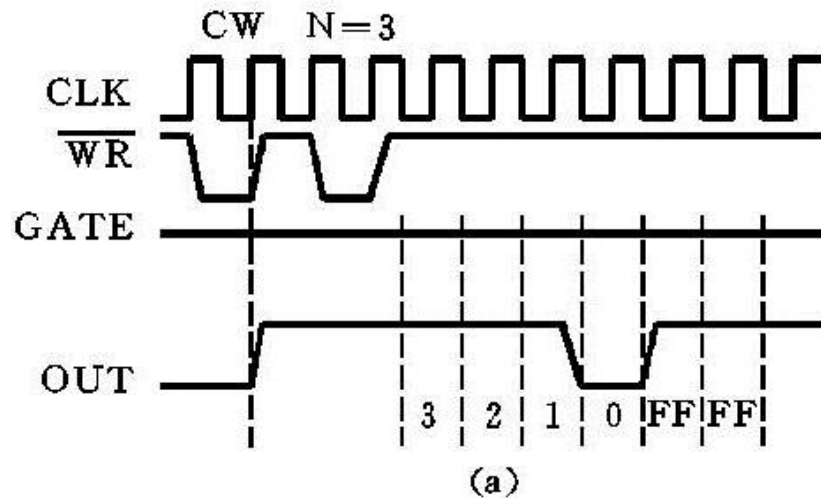
### 5. 方式4 – 软件启动的单脉冲发生器（软件触发选通）



✳软件启动；减1计数；无初值自动重装功能



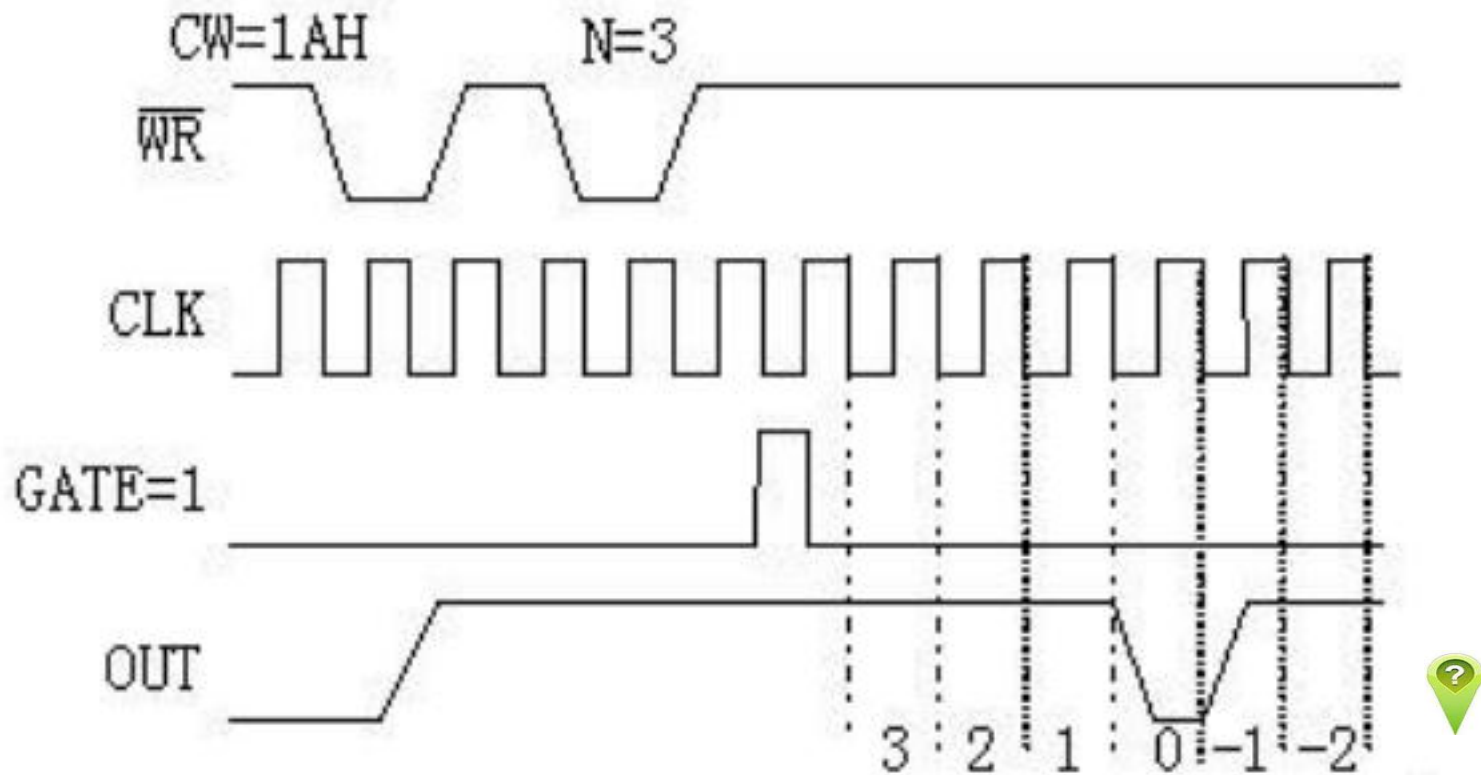
## 5. 方式4 – 软件启动的单脉冲发生器





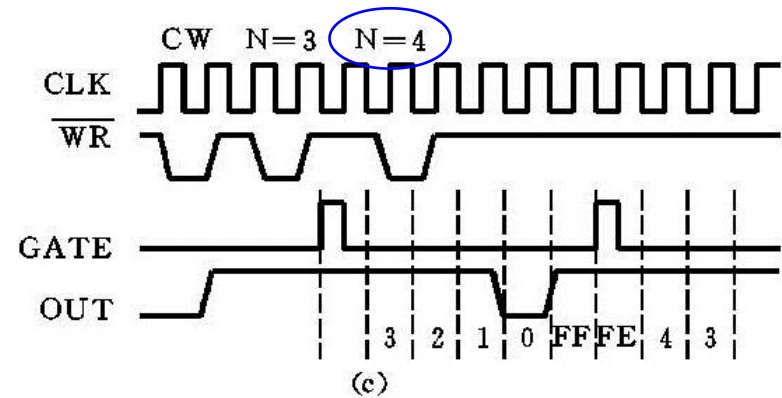
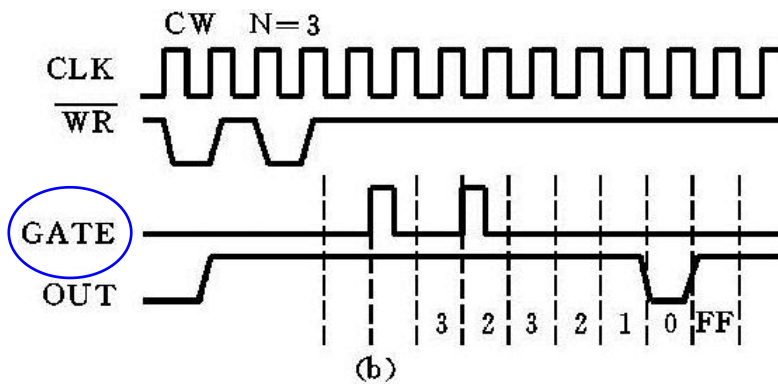
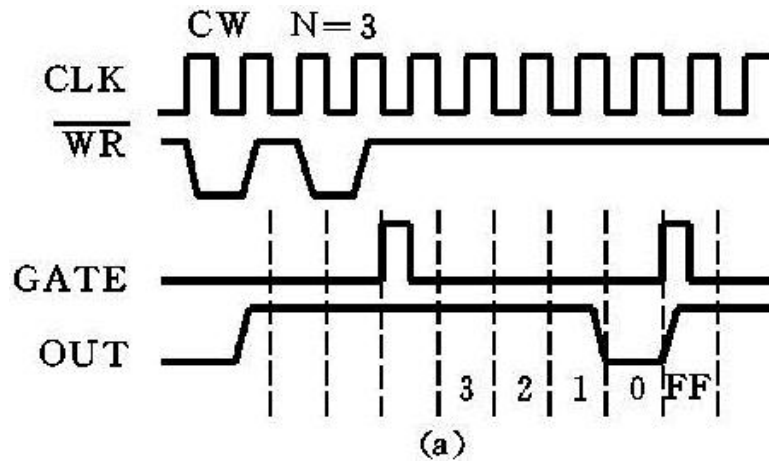
## 8.1.2 8254的工作方式

### 6. 方式5 – 硬件触发的单脉冲发生器



✱硬件启动；减1计数；无初值自动重装功能

## 6. 方式5－硬件触发的单脉冲发生器



## 7. 8254六种工作方式的比较

		方式0	方式1	方式2	方式3	方式4	方式5
<b>OUT输出状态</b>		写入控制字后变0, 计数结束变1, 并维持至重写控制字或计数初值	写入控制字后变1, GATE上升沿触发变0, 开始计数, 计数结束变1	写入控制字后变1, 计数到1变0, 维持一个Tclk变1	写入控制字后变1, 装入初值且GATE=1则OUT变1, 计数到变0, 重装初值继续计数, 计数到则反向	写入控制字后变1, 计数结束变0, 维持一个Tclk变1	写入控制字后变1, GATE上升沿触发开始计数, 计数结束输出一个CLK的负脉冲
<b>初值自动重装</b>		无	无	计数到0重装	根据初值奇偶分别重装;	无	无
<b>计数过程中改变初值</b>		立即有效	GATE触发后有效	计数到1或GATE触发后有效	计数结束或GATE触发后有效	立即有效	GATE触发后有效
<b>GATE</b>	0	禁止计数	无影响	禁止计数	禁止计数	禁止计数	无影响
	下降沿	暂停计数	无影响	停止计数	停止计数	停止计数	无影响
	上升沿	继续计数	从初值开始重新计数	从初值开始重新计数	从初值开始重新计数	从初值开始重新计数	从初值开始重新计数
	1	允许计数	无影响	允许计数	允许计数	允许计数	无影响

## 8.1.3 8254的编程

### 1. 8254的控制字与初始化

8254是可编程接口芯片，使用前必须先对它进行初始化。

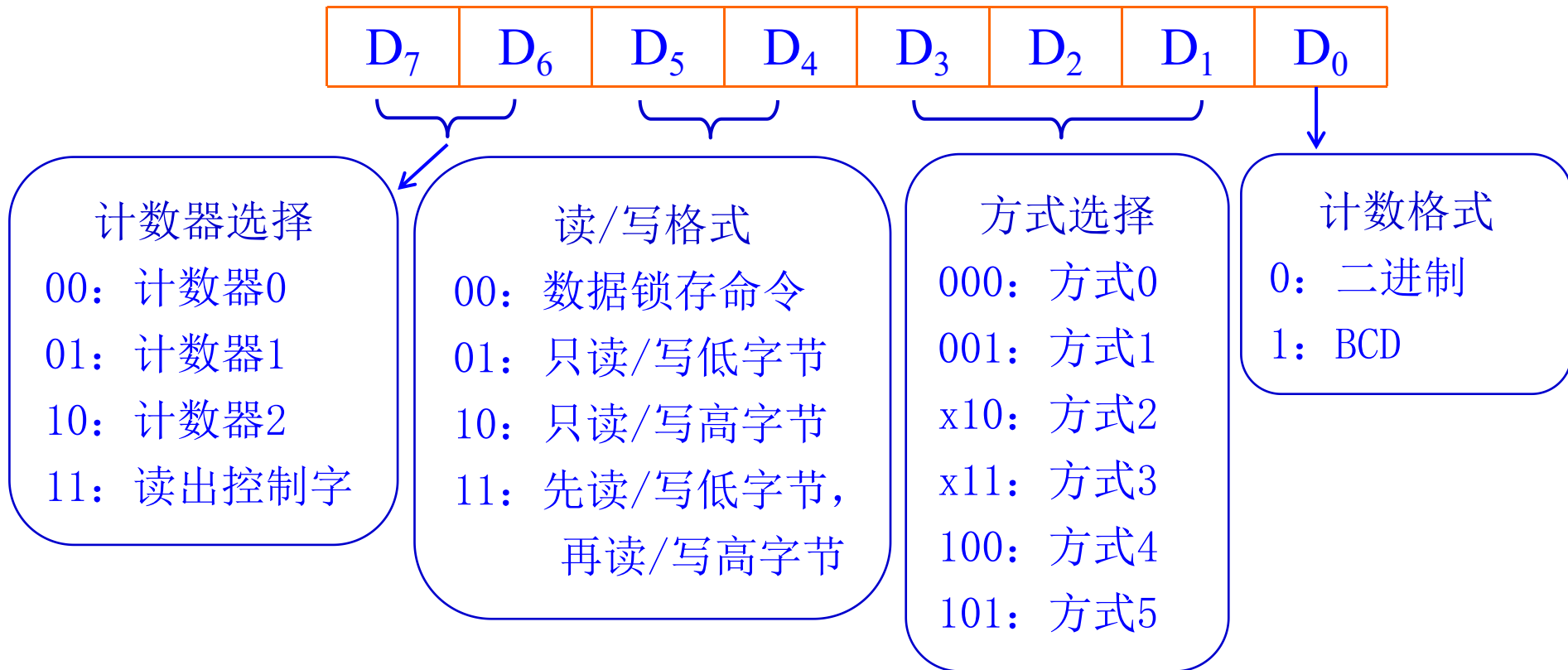
8254的初始化编程包括以下两个步骤：

(1)向8254写入控制字，用于确定所选通道的工作方式和计数格式，写入控制字同时起到复位作用。

(2)向选定的计数通道写入计数初值，每个通道都有一个独立的端口地址。每个通道在写入控制字和计数初值之后开始工作。

## 8.1.3 8254的编程

### 1. 8254的方式控制字





### 8.1.3 8254的编程

例1 设某片8254三个计数器的CR/OL（计数/输出锁存寄存器）端口地址为70H、71H、72H，控制寄存器端口地址73H。编程使计数器0工作在方式2，CR/OL仅使用低8位，初值为100，计数值使用二进制。

MOV AL, 14H	;构造控制字
OUT 73H, AL	;向控制端口输出
MOV AL, 100	;计算计数初值
OUT 70H, AL	;向选定计数通道输出

### 8.1.3 8254的编程

例2 设三个计数器的CR/OL端口地址为70H、71H、72H，控制寄存器端口地址73H。计数器1，工作模式1，CR/OL使用16位，初值为1234，计数值使用BCD格式。

```
MOV  AL, 73H    ;构造控制字
OUT  73H, AL    ;向控制口输出

MOV  AX, 1234H  ;取得计数初值
OUT  71H, AL    ;先输出低8位
MOV  AL, AH     ;再输出高8位
OUT  71H, AL
```

更多例子 >>

## 8.1.3 8254的编程

### 2. 8254的锁存命令

D7	D6	0	0	X	X	X	X
----	----	---	---	---	---	---	---

00:锁存0号计数器当前计数值

01:锁存1号计数器当前计数值

10:锁存2号计数器当前计数值

11:非法

通道结构





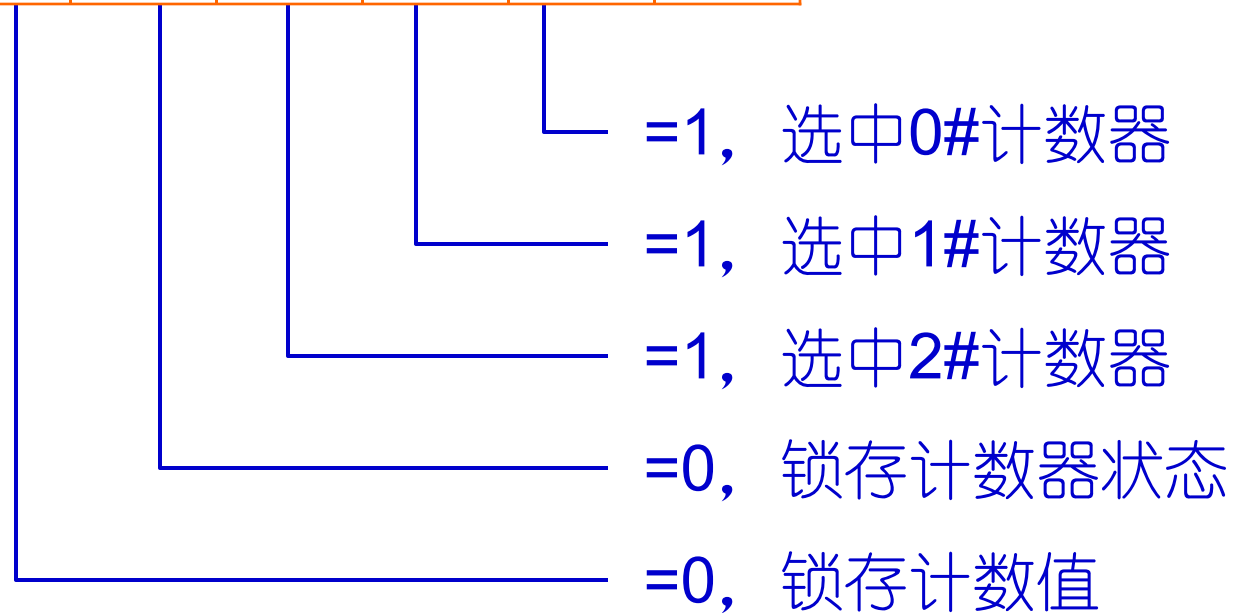
# 锁存命令编程举例

例3 设三个计数器的CR/OL端口地址为70H、71H、72H，控制寄存器端口地址73H。读出计数器0的当前计数值，放在BX中。

```
MOV  AL, 0H    ;构造锁存命令字
OUT  73H,AL    ;向控制口输出
IN   AL,70H    ;读低8位计数值
MOV  BL,AL
IN   AL,70H    ;读高8位计数值
MOV  BH,AL
```

## 8.1.3 8254的编程

### 3. 8254的读出控制字



## 读出控制字编程举例

例4 用读出命令字读出计数器2的当前计数值，计数值为16位。假设：端口地址为208H~20BH。

```
MOV  DX, 20BH           ;控口地址
MOV  AL, 1101 1000B;
OUT  DX,AL              ;输出读命令字
MOV  DX,20AH            ;2号口地址
IN   AL,DX              ;读出低8位
MOV  BL,AL
IN   AL,DX              ;读出高8位
MOV  BH,AL
```

## 8.1.3 8254的编程

### 4. 8254的状态字

D7	D6	D5	D4	D3	D2	D1	D0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

**OUTPUT**即**D<sub>7</sub>**位，表示**OUT**引脚的输出状态，**D<sub>7</sub>=1**表示**OUT**端当前输出高电平，**D<sub>7</sub>=0**表示**OUT**端当前输出低电平；

**NULL COUNT**，**D<sub>6</sub>**位，表示是否已经装入计数初值，**D<sub>6</sub>=0**，代表已装入初值，读取的计数值有效；

**D<sub>5</sub> ~ D<sub>0</sub>**各位是由方式控制字确定的，与方式控制字的对应位相同。



## 8.2 8254/8253在 PC机中的应用

计数器0:

向系统日历时钟提供定时中断

方式3, 控制字36H, 计数器初始值0

计数器1:

动态RAM刷新

方式2, 控制字54H, 计数器初始值18 (12H)

计数器2:

控制扬声器发声

方式3, 控制字B6H, 计数器初始值1331 (533H)

PC机中, 8254的端口地址为40H~43H

# 计数器0 定时中断

- ✧ 计数器0工作在方式3，计数初值为0，输出频率为 $1.19318\text{MHz} \div 65536 = 18.206\text{Hz}$ 的方波
- ✧ 门控为常启状态，这个方波信号不断产生
- ✧ OUT0端接8259A的IRQ0，用作中断请求信号，每秒产生18.206次中断请求，或说每隔55ms（54.925493ms）申请一次中断。
- ✧ DOS系统利用计数器0的这个特点，通过08号中断服务程序实现了日时钟计时功能

## 8.2.1 BIOS对8253初始化

计数器0的初始化程序如下：

MOV AL,36H	;控制字 <u>00,11,011,0</u>
OUT 43H,AL	;通道0,16位,方式3,二进制
MOV AL,0	;初值0,计数65536次(最大)
OUT 40H,AL	;写入CR0的低计数值
OUT 40H,AL	;写入CR0的高计数值



# 计数器1 DRAM定时刷新

- ✧ 需要重复不断提出刷新请求，门控总为高，选择方式2或3；
- ✧ 2ms内刷新128次，即15.6μs刷新一次，计数初值为18。

$$18=15.6/(1/ 1.19318)$$

原理图

# BIOS对8253初始化

计数器1的初始化程序如下：

```
MOV AL,54H      ; 控制字01,01,010,0  
OUT 43H,AL      ; 通道1,只装低8位,方式2,二进制  
MOV AL,12H      ; 初值18  
OUT 41H,AL      ; 写入计数器低8位, 高8位自动为0
```

# BIOS对8253初始化

计数器2的初始化程序如下：

```
MOV AL,0B6H    ; 控制字10,11,011,0  
OUT 43H,AL     ; 通道2,16位,方式3,二进制  
MOV AX,533H    ; 初值 533H = 1331  
OUT 42H,AL     ; 写入计数器低8位  
MOV AL,AH      ;  
OUT 42H,AL     ; 写入计数器高8位
```

## 8.2.2 其它应用举例

例5 扬声器控制。设计一个程序，使扬声器发出600Hz频率的声音，按下任意键声音停止。

PC机的发声系统以计数器2为核心。CLK2的输入频率1.19MHz,改变计数器初值可以由OUT2得到不同频率的方波输出。对于600Hz，计数初值 $1.19\text{MHz}/600\text{Hz}=1938$ 。

发声系统受8255芯片B口的两个输出端线PB0、PB1的控制  
PB0为1，使GATE2为1，计数器2能正常计数；PB1为1，  
打开输出控制门



```
CODE    SEGMENT
        ASSUME CS:CODE
START:
        IN      AL, 61H
        OR      AL, 03H
        OUT     61H, AL
        MOV     AX, 1938
        OUT     42H, AL
        MOV     AL, AH
        OUT     42H, AL
        MOV     AH, 01H
        INT     21H
        IN      AL, 61H
        AND     AL, 0FCH
        OUT     61H, AL
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```

按任意键结束

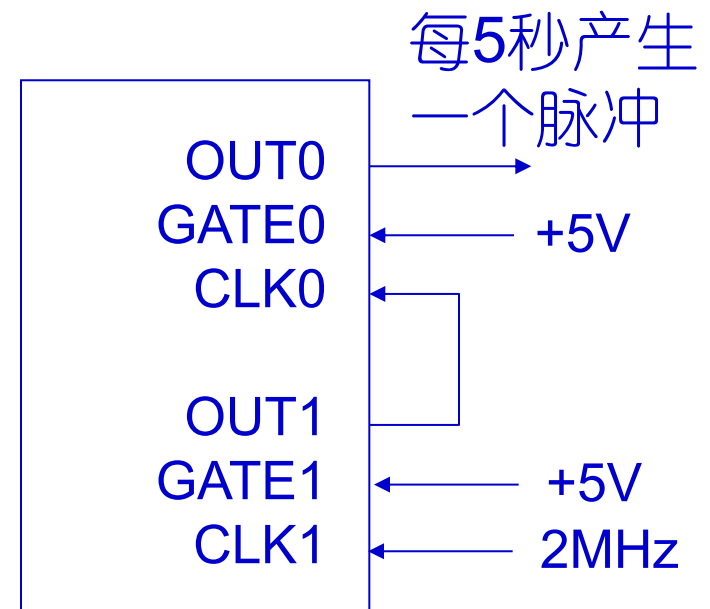
## 8.2.2 其它应用举例

例6 以2MHz输入8253，实现每5秒定时中断。设8253端口地址40H~43H。

分析：能够让8253实现最长计时的初值是0，在CLK=2MHz时可计时的最大时间间隔为：

$$65536/(2 \times 10^6) = 32.769\text{ms}$$

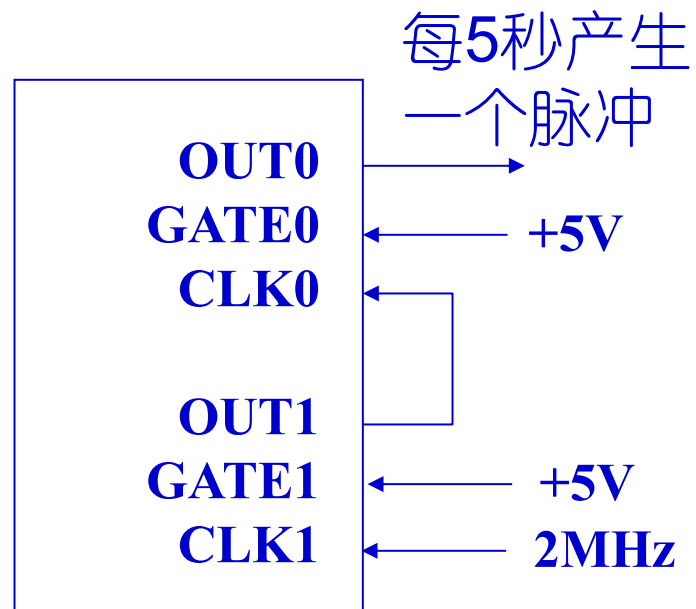
所以需要两个计数器串联，一个计数器的输出作为另一个计数器的输入。



# 例6

计数器1：模式2，OUT1每5ms输出一个脉冲  
初值 $0.005 / (1 / (2 \times 10^6)) = 10000$

计数器0：模式2，OUT0每5s输出一个脉冲  
初值 $5 / 0.005 = 1000$



## 例6程序



```
MOV  AL, 74H           ; 通道1初始化, 01 11 010 0
OUT   43H, AL
MOV   AX, 10000         ; 初值10000, 输出信号周期5毫秒
OUT   41H, AL
MOV   AL, AH
OUT   41H, AL

MOV   AL, 34H           ; 通道0初始化, 00 11 010 0
OUT   43H, AL
MOV   AX, 1000          ; 初值1000, 输出信号周期5秒
OUT   40H, AL
MOV   AL, AH
OUT   40H, AL
```

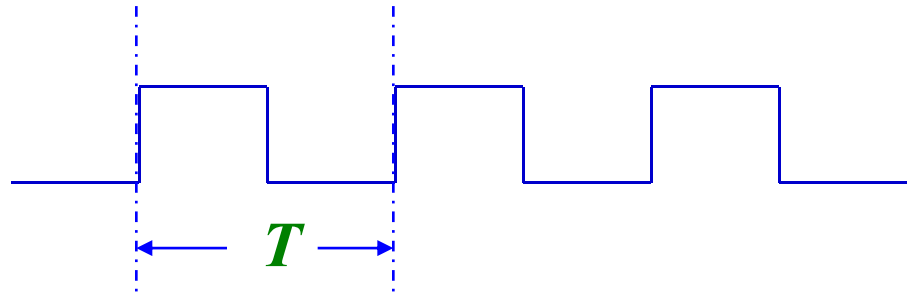




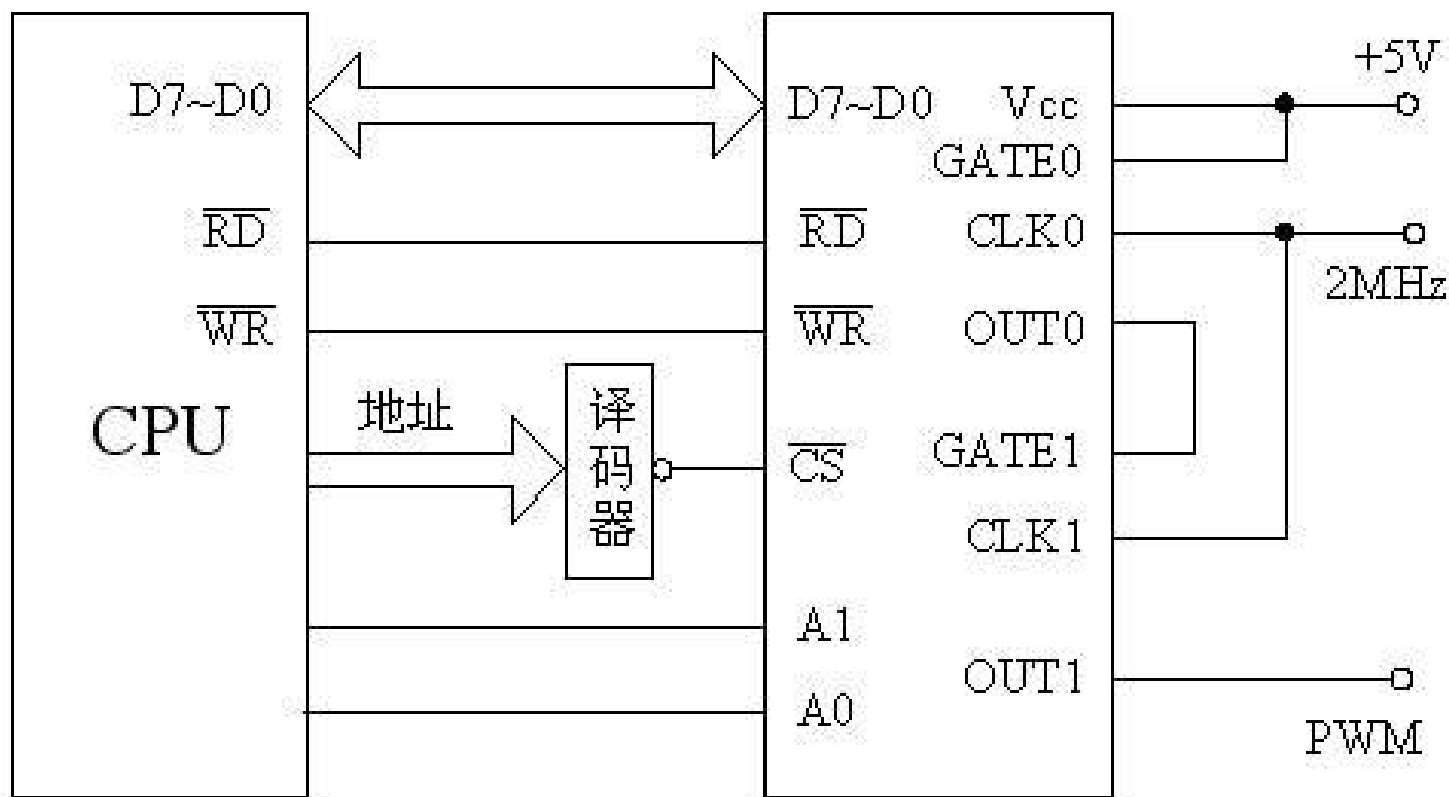
## 8.2.2 其它应用举例

### 例7 PWM 脉宽调制

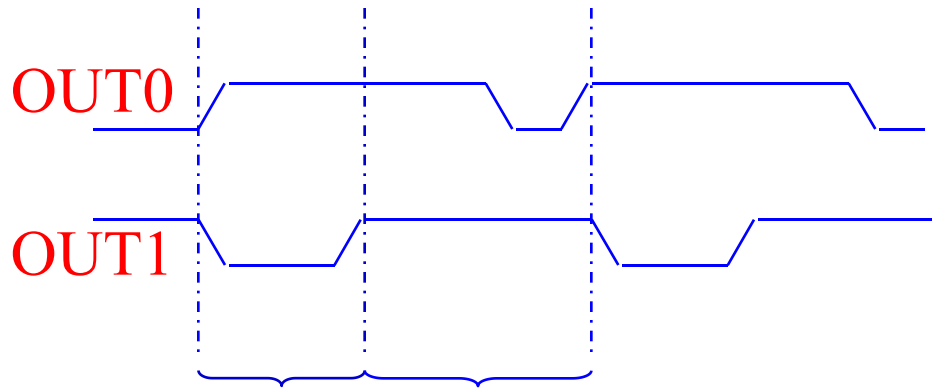
在工业生产和仪器、仪表中，经常需要对交、直流电机进行转速的调节。有多种可以使用的调速方法，其中**PWM**（脉宽调制）因实现容易，调速准确，得到广泛的使用。这种方法用一个开关电源对电机供电，控制电源开、关的时间比例，就可以控制输出的有效电压，从而控制电机的转速。可以用**8254**来定时，输出周期固定、占空比可变的脉冲信号。



如图，设8254端口地址为240H~243H。让计数器0工作在方式2（分频器），产生周期和宽度固定的脉冲信号。计数器1工作在方式1，把OUT0连接到GATE1。由方式1的特点可以知道，OUT1信号与OUT0具有相同的周期。计数器1输出OUT1用作PWM脉冲。可以看出，PWM脉冲周期由计数器0决定，宽度由计数器1决定。



设系统时钟频率为2MHz（时钟周期 $0.5\mu\text{s}$ ）。设PWM周期 $T=5\text{ms}$ ，该周期信号由计数器0控制输出：方式2，计数初值为 $5\text{ms}/0.5\mu\text{s} = 10000$ 。PWM脉冲宽度由计数器1控制产生：方式1，计数值为N时（ $0\sim 10000$ ），低电平时间为 $0.5\mu\text{s} \times N$ ，输出有效电压为最大值的 $(10000-N)/10000$ 。



程序如下：





PWM:

MOV DX, 0243H ; 8254控制口地址送DX

MOV AL, 34H ; 计数器0, 方式2, 写16位, 二进制

OUT DX, AL ; 控制字写入计数器0控制寄存器

MOV DX, 0240H

MOV AX, 10000

OUT DX, AL ; 写入初值低8位

MOV AL, AH ; 计数器0的计数初值高8位

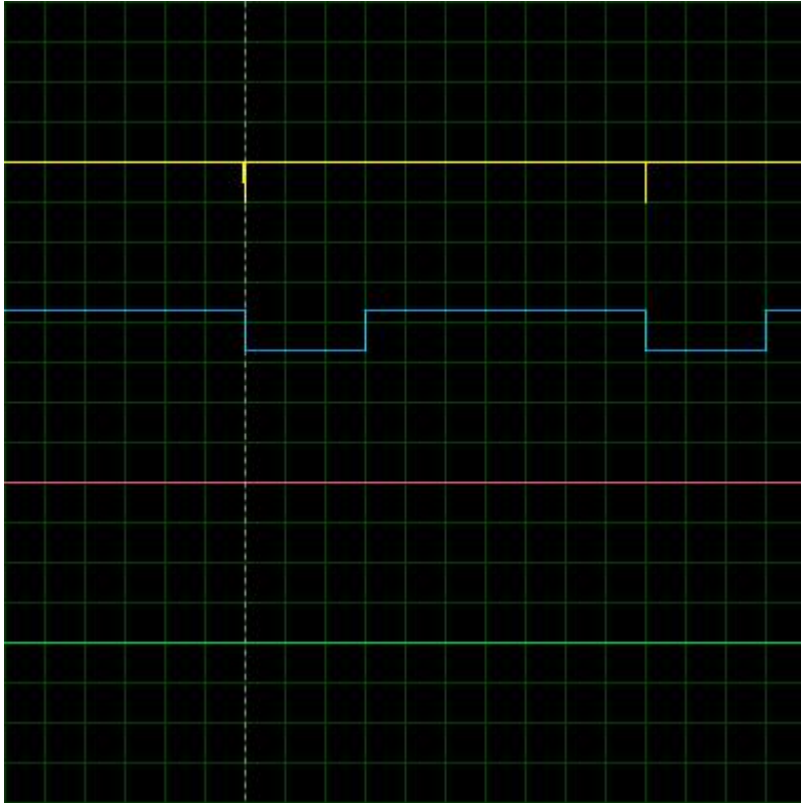
MOV DX, AL ; 写入初值高8位



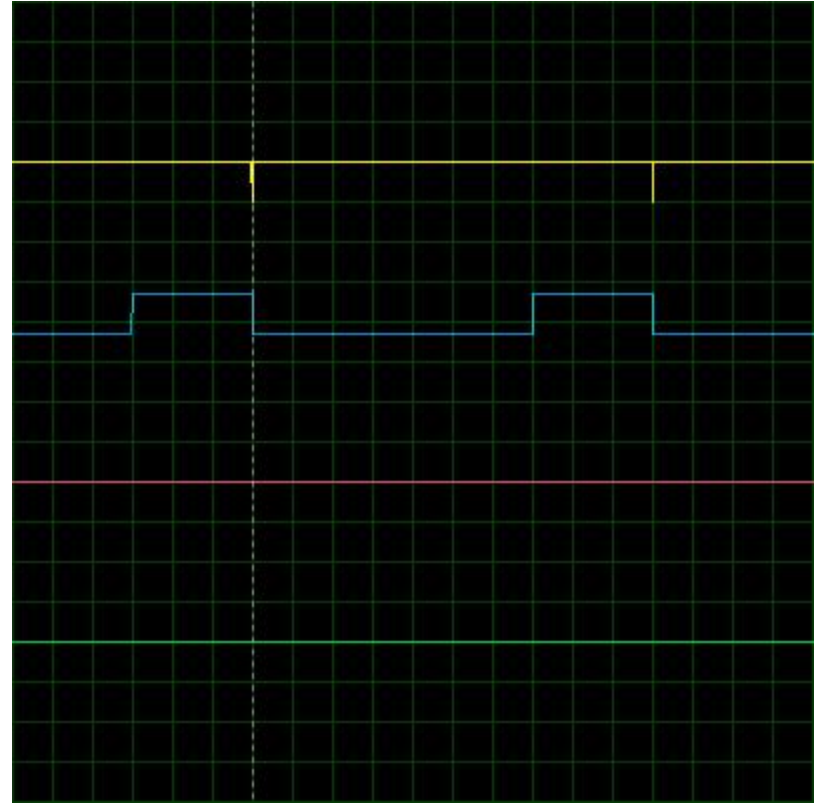
MOV DX, 0243H ; 8254控制口地址送DX  
MOV AL, 72H ; 计数器1, 方式1, 写16位, 二进制  
OUT DX, AL ; 控制字写入计数器0控制寄存器  
MOV DX, 0241H ; 8254计数器1口地址送DX  
MOV AX, N ; 计数器1的计数初值  
MOV DX, AL ; 写入初值低8位  
MOV AL, AH ; 计数器1的计数初值高8位  
MOV DX, AL ; 写入初值高8位



# Proteus仿真结果



**N=3000**



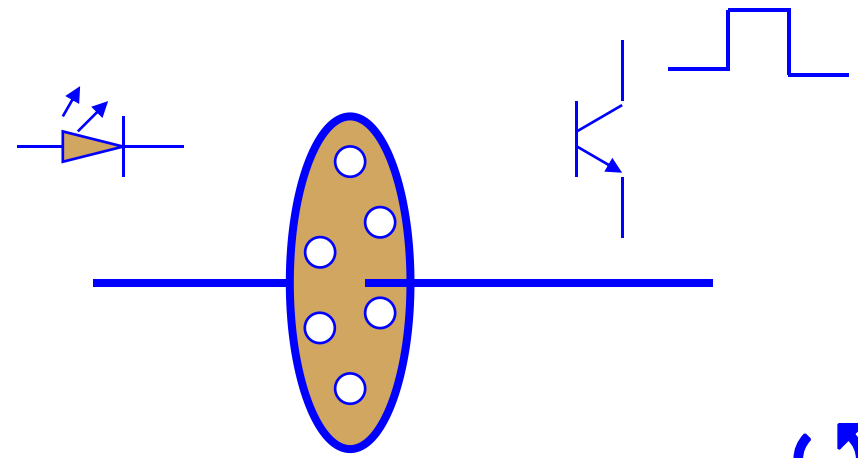
**N=7000**



## 8.2.2 其它应用举例

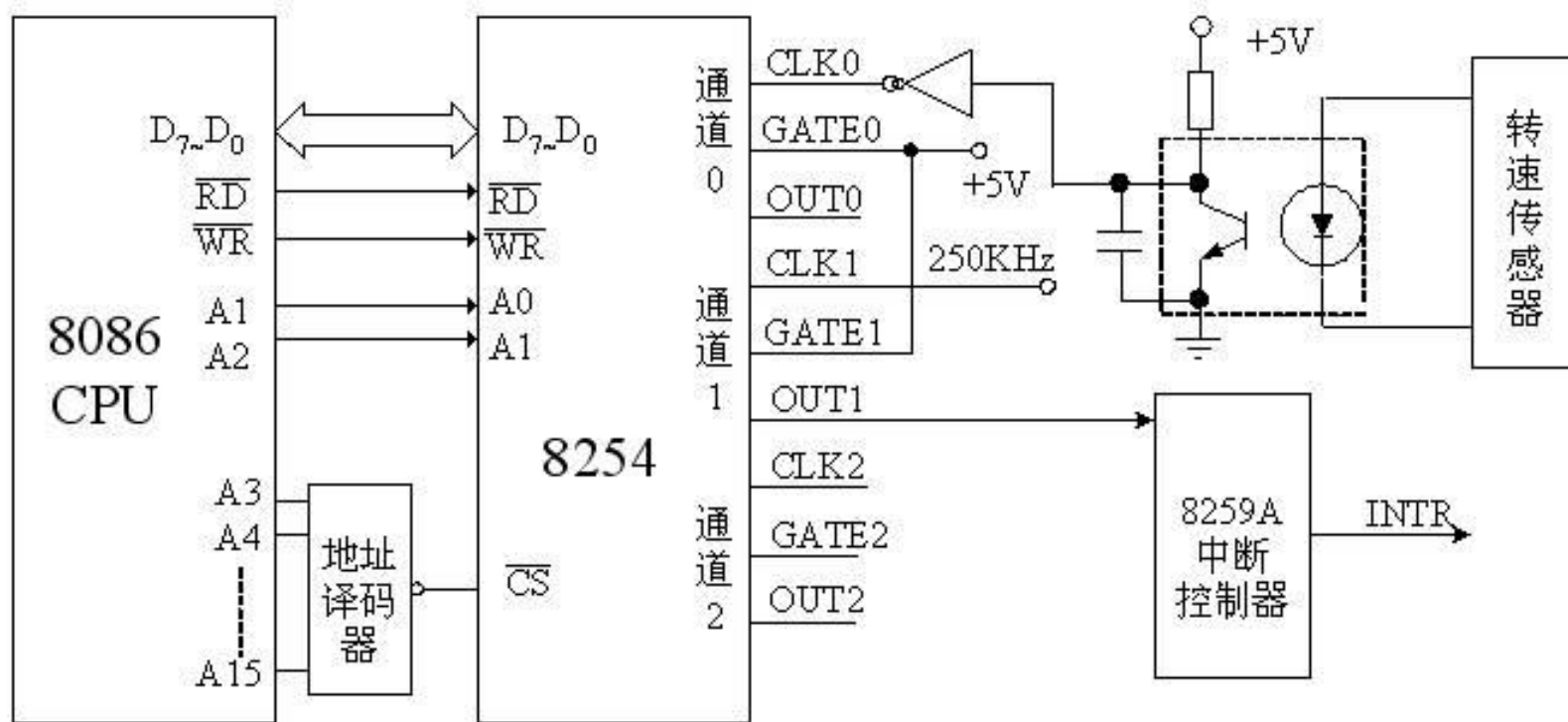
### 例8 电机转速测量

为了测量电机的转速，可以在电机轴安装一个转盘，上面有8个均匀分布的小孔。转盘一侧是发光源，另一侧是光电转换电路。转盘上的小孔转到发光源位置时，光透过小孔使光电二极管导通，产生一个正脉冲。纪录单位时间内脉冲的个数，可以得到电机的转速。



下图是使用8254进行电机转速测量的电路。通道0用来对脉冲进行计数，以方式0工作。通道1用来产生定时信号，工作在方式3。如果基准时钟频率为250KHz，每定时0.1秒钟产生一次中断，则计数初值为25000。设0.1秒钟内计数脉冲数为COUNT，则转速N为：

$$N = 10 \times \text{COUNT} \div 8 \times 60 = \text{COUNT} \times 75 \text{ (转/分)}$$





程序如下：

.data

COUNT DW ?

SPEED DW ?

.code

START: MOV AX, @data

MOV DS, AX


； 设置计数器0、计数器1的工作方式

MOV DX, 0243H ; 8254控制口地址送DX

MOV AL, 30H ; 计数器0, 方式0, 写16位, 二进制


OUT DX, AL ; 控制字写入计数器0控制寄存器







MOV AL,74H ; 计数器1, 方式2, 读写16位, 二进制  
OUT DX,AL ; 控制字写入计数器1控制寄存器  
; 设置计数器0、计数器1的初值  
MOV DX, 0240H ; 8254计数器0端口地址送DX  
MOV AL, 00H ; 计数器0的计数初值  
OUT DX, AL ; 写入初值低8位  
OUT DX, AL ; 写入初值高8位  
MOV DX, 0241H ; 8254计数器1端口地址送DX  
MOV AX, 25000 ; 定时0.1秒, 计数初值为25000





```
OUT  DX,AL          ; 写入初值低8位
MOV  AL,AH
OUT  DX,AL          ; 写入初值高8位
MOV  SIGNAL,0       ; 标志单元清0
; 装载中断向量, 清屏蔽位, 开放中断, .....
.....
; 计算转速
MOV  AX, 0
SUB  AX, COUNT      ; 取计数脉冲值
MOV  BX, 75
```





```
MUL    BX                ; 计算转速
MOV    SPEED,AX          ; 保存转速
.....                  ; 输出/显示电机转速
; 定时中断程序
```


```
TIME_INT PROC FAR
```

```
    PUSH    DS
```

```
    PUSH    AX
```

```
    PUSH    DX
```

```
    STI                ; 开放中断
```



MOV AL,00H ; 锁存8254计数器0计数值的控制字  
MOV DX,0243H ; 控制寄存器地址送DX  
OUT DX,AL  
MOV DX,0240H ; 计数器0的端口地址送DX  
IN AL,DX ; 读取低8位数据  
XCHG AL,AH  
IN AL,DX ; 读取高8位数据  
XCHG AL,AH  
MOV COUNT,AX ; 计数脉冲数送COUNT字单元保存



； 重新设置计数器0

MOV DX,0243H ; 8254控制口地址送DX

MOV AL,30H ; 计数器0, 方式0, 写16位, 二进制


OUT DX,AL ; 控制字写入计数器0控制寄存器

MOV DX,0240H ; 8254计数器0端口地址送DX

MOV AX,0 ; 计数器0的计数初值

OUT DX,AL ; 写入初值低8位

OUT DX,AL ; 写入初值高8位



```
CLI                ; 关闭中断
MOV  AL,20H
OUT  20H,AL        ; 中断结束命令
POP  DX
POP  AX
POP  DS
IRET
TIME_INT  ENDP
```



# 实验六 8253实验










# 详细说明



# 1、8254的引脚

- 
- (1) **CS\*** 片选信号：低电平有效。接I/O端口译码电路的输出。为低电平时，**CPU**才能对**8254**进行读写操作。
  - (2) **RD\*/WR\***：读 / 写控制信号，低电平有效。接系统总线的外设读写信号**IOR**和**IOW**。
  - (3) **D7~D0**： **8254**的数据线，与系统数据总线相连。
  - (4) **A1A0**：接地址总线低2位，用于片内端口的选择。  
 $A_1A_0=00$ 时，选择通道0； $A_1A_0=01$ ，选择通道1； $A_1A_0=10$ ，选择通道2； $A_1A_0=11$ ，选择控制端口。
- 



如果系统数据总线为8位，可以将 $A_1A_0$ 与地址总线的最低两位 $A_1A_0$ 对应连接；如果系统数据总线为16位，通常将8254的8位数据线接到系统数据总线的低8位。这样，CPU就要求芯片内部的各端口都使用偶地址，于是8254的地址线 $A_1A_0$ 与系统地址总线的 $A_2A_1$ 对应连接（假设 $A_0=0$ ）。

(5) 每个通道有三根对外的信号线：CLK，OUT和GATE。它们的作用在下一小节介绍。

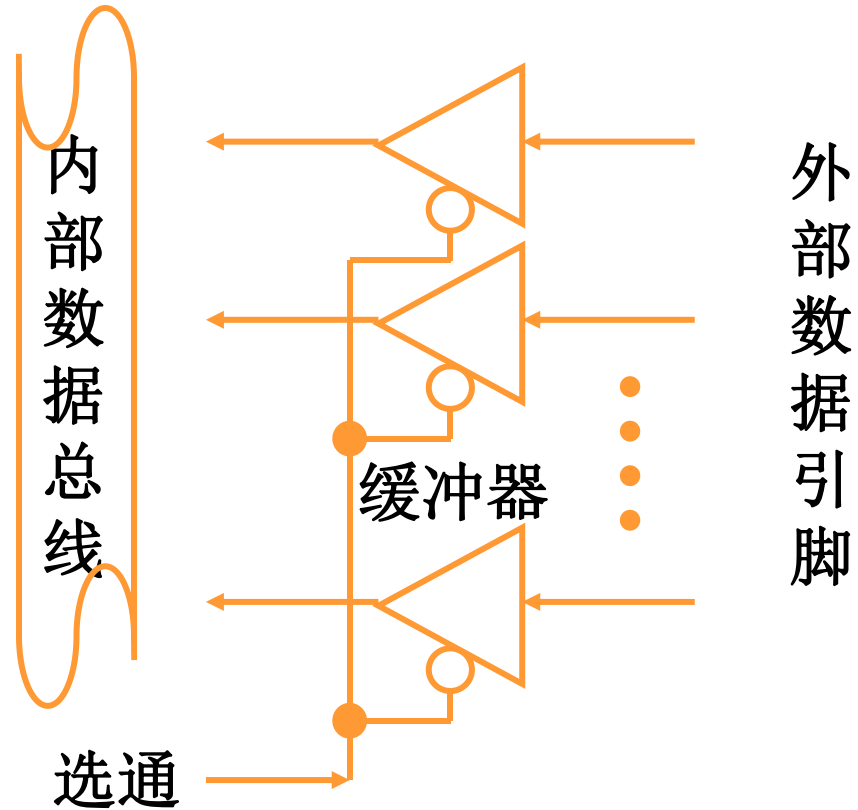
8254的读 / 写操作逻辑见表5-4：



## 1. 数据总线缓冲器

数据总线缓冲器是**8254**与系统数据总线相连的接口电路，该缓冲器为**8位双向三态的缓冲器**，可直接挂在数据总线上。通过数据总线缓冲器，**CPU**用指令对**8254**进行读/写信息的操作，这些信息是：

- (1) **CPU**对**8254**进行初始化编程时，向控制寄存器写入控制字。
- (2) **CPU**向某一计数器写入计数初值。
- (3) **CPU**读出计数器的当前计数值。



内部结构

## 2. 读/写控制逻辑

这是8254内部的控制电路，读/写逻辑的功能是接收来自CPU的控制信号，包括读信号RD、写信号WR、片选信号CS和芯片内部寄存器寻址信号 $A_1A_0$ 。当片选信号CS=0时，由 $A_1A_0$ 信号选择芯片内部寄存器寻址，由读信号和写信号完成对选定寄存器的读、写操作。

内部结构

### 3. 计数器0~2

如图所示，8254有3个结构完全相同的定时器/计数器通道：0、1和2。这3个定时器/计数器的操作是相互独立的。每个通道包含一个8位的控制字寄存器、三个16位的初值寄存器、减1计数器和锁存计数结果输出锁存器。减1计数器的起始值来自初值寄存器，由程序设置；任何时刻计数器的瞬时值都存放在16位输出锁存器中，CPU在需要时可以读出这个值。

每个计数器有3根专用的信号线。其中2根为输入信号：时钟信号CLK和门控信号GATE；1根输出信号OUT。每个通道工作时，都是对CLK端输入的脉冲按二进制或BCD码格式进行计数，也就是对设置的初值做减1操作，计数值减到零时，由输出端OUT输出结束信号，输出信号的波形由工作方式确定。此外，可以用GATE引脚上的门控信号允许或停止计数的过程。

内部结构

# 计数器的3个引脚

## CLK时钟输入信号:

在计数过程中，此引脚上每输入一个时钟信号（下降沿），计数器的计数值减1

## GATE门控输入信号:

控制计数器工作，可分成电平控制和上升沿控制两种类型

## OUT计数器输出信号:

当一次计数过程结束（计数值减为0），OUT引脚上将产生一个输出信号

内部结构



8254作外部事件计数器时，CLK引脚上的计数脉冲从外部输入，这些脉冲的间隔可以不相等。

8254用作定时器时，CLK引脚的输入信号是精确的时钟脉冲。8254的定时时间，取决于时钟脉冲的频率和计数器的初值，即：

定时时间 $T = \text{时钟脉冲周期}t_c \times \text{计数初值}n$

例如，在某微机中，8254的计数脉冲频率是1MHZ，脉冲周期 $t_c = 1\mu s$ ，设置的计数器初值 $n = 1000$ ，则定时时间 $T = 1\mu s \times 1000 = 1ms$ 。

内部结构



## 4. 控制寄存器

控制寄存器是一个只能写入的寄存器，它接收从CPU来的控制字，并由控制字的 $D_7$ 、 $D_6$ 位的编码决定控制字写入哪个计数器的控制寄存器。在对8254进行编程时，CPU用输出指令向它写入控制字，由此确定各计数器通道的工作方式、读写格式和计数的数制。

内部结构

## 4. 8254的工作方式

### 1. 方式0 – 计数结束OUT输出高电平（计数结束中断）

写入方式0控制字后（图中标有CW=10H），输出OUT立即变为低电平，且在计数过程中一直维持低电平。赋初值后（图中标为N=4），在每个CLK时钟下降沿，计数器进行减1计数。计数值减到零时，OUT输出变为高电平，并且一直保持到该通道重新装入计数初值或重新设置工作方式为止。

OUT信号可用于向CPU发出中断请求。

方式0



## 1. 方式0 – 计数结束OUT输出高电平（计数结束中断）

**GATE**用于控制计数过程。

**GATE**为高电平，允许计数；

**GATE**为低电平，暂停计数；当**GATE**重新为高电平时又恢复计数。但**GATE**不影响输出端**OUT**的电平。

方式0每写一次计数值进行一次计数，计数器不会自动恢复初值重新开始计数。若在计数过程中改变计数值，则在写入新值后的下一个时钟下降沿计数器将按新的初值计数。

方式0

## 4. 8254的工作方式

### 2. 方式1 – 可重触发的单稳态触发器 （可编程单稳脉冲）

计数由外部门控脉冲启动。门控信号**GATE**上升沿触发后，可产生一拍负脉冲信号的输出，脉冲宽度由计数初值**N**决定。写入控制字后，**OUT**输出为高电平，写入计数初值后**OUT**继续保持高电平。**GATE**上升沿到达后，**OUT**输出低电平，并在**CLK**脉冲下降沿进行减1计数；计数值减到0时，输出**OUT**变为高电平，从而产生一个宽度为**N**个时钟周期的负脉冲。

方式1

## 2. 方式1 – 可重触发的单稳态触发器 （可编程单稳脉冲）

计数结束后，若再来一个**GATE**信号上升沿，则下一个时钟周期的下降沿又以上次写入的初值开始计数，不需要重新写入初值。也就是说，可以用门控信号重新触发计数。在计数过程中，若再来一个门控信号的上升沿，则下一个时钟下降沿开始从初值起重新计数，即终止原来的计数过程，开始新一轮计数。

在计数过程中可以写入新的初值，它不会影响正在进行的计数过程。在下一个门控信号到来后，按新值开始计数。

方式1

### 3. 方式2 – 分频器（频率发生器）

写入控制字后，输出端**OUT**变成高电平。写入计数初值后，如果**GATE**为高电平，计数器开始减1计数。减到1时（不是0），输出端**OUT**变为低电平，维持一个**CLK**周期，然后输出**OUT**又变成高电平，同时从初值开始新的计数过程。这种方式工作时，计数初值自动重装，计数器能连续工作，输出固定频率的脉冲，因此称为分频器。

方式2中，**GATE**信号为低电平时终止计数。**GATE**的上升沿使计数器恢复初值，并从初值开始计数。

在计数过程中，**GATE**为高电平，此时写入新的计数初值不会影响正在进行的计数过程，只有计数器减到1之后，计数器才装入新的计数初值，并按新的初值开始计数。

方式2


## 4. 方式3－方波发生器

(1) 计数初值为偶数。写入控制字后，输出端OUT变成高电平。写入计数初值后计数器开始进行减2计数，减到0时，输出端OUT改变输出极性，并重新从初值开始新的计数过程。OUT输出连续的、高/低电平各占1/2的方波。

(2) 计数初值为奇数，写入控制字后输出端OUT变成高电平，写入计数初值后开始减1计数。减到  $(N+1)/2$  以后，输出端OUT变为低电平，减到0时，OUT又变成高电平，并重新从初值开始新的计数过程。这时的输出波形高电平时间比低电平多一个CLK信号周期。

**GATE=1**，允许计数，**GATE=0**，禁止计数。**GATE**信号能使计数过程重新开始。如果在输出端OUT为低电平期间，**GATE**变低，则OUT立即变高，并停止计数。**GATE**重新变高以后，计数器重新装入初值并开始新的计数。

方式3



如果在计数过程中写入新的初值，而**GATE**信号一直维持高电平，则新的初值不会立即影响当前的计数过程，只有在此轮计数结束后的下一个计数周期，才按新的初值计数。

方式3



## 4. 8254的工作方式

### 5. 方式4 – 软件触发选通

写入方式控制字后，**OUT**输出高电平。写入初值，经过一个**CLK**脉冲开始减1计数，计到0时（注意：不是减到1时），**OUT**输出为低电平，持续一个**CLK**脉冲周期后再恢复到高电平。这种工作方式不能自动重装初值。要启动下一次计数，必须重新写入计数初值。

**GATE=1**时，允许计数；**GATE=0**，禁止计数，并把输出维持在当时的电平。

如果**GATE=1**时在计数过程中改变计数值，则在写入新值后的下一个时钟下降沿计数器立即按新的初值开始计数。

方式4

## 4. 8254的工作方式

### 6. 方式5－硬件触发选通

#### (1) 基本工作过程

- ①写入**8254**方式字后，**OUT**输出高电平。
- ②写入计数初值后，计数器并不立即开始计数，在**GATE**端输入上升沿触发信号后，计数开始。
- ③计数器减到**0**时，输出一个持续时间为一个时钟周期的负脉冲，然后输出恢复为高电平，并自动装入初值，等待下一个**GATE**触发信号。

输出负脉冲可以用作选通脉冲，它是通过硬件电路产生的门控信号上升沿触发得到的，所以叫硬件触发选通脉冲。这时**8254**相当于一个硬件触发的选通信号发生器。

方式5

## 4. 8254的工作方式

### 6. 方式5－硬件触发选通

(2) 若计数过程中，又有一个门控信号的上升沿出现，则立即终止现行的计数过程，且在下一个时钟下降沿又从初值开始计数。

(3) 如果在计数过程中写入新的初值，则新的初值不会立即影响当前的计数过程，直到下一个门控信号上升沿到来后，才从新的初值开始进行减1计数。

方式5

## 5. 8254的编程：控制字

### (1) D7、D6为“计数器/读出控制字”选择位

8254三个计数通道的方式控制字和读出控制字都共用同一个端口地址，即 $A_1A_0=11$ 。因此，必须用控制字本身来区分写入的是哪一个计数器的命令。控制字的 $D_7D_6$ 两位为00，01，10分别选择三个计数通道，为11选择控制寄存器，用于读出控制寄存器内容。

控制字

## 5. 8254的编程

### (2) $D_5$ 、 $D_4$ 为读/写方式选择位

为了对计数器的计数值进行实时显示、实时检测或对计数值进行数据处理，有时需要读回计数器的当前计数值。 $D_5D_4=00$ ，表示锁存计数器的当前计数值，以便读出。

$D_5D_4=01$ ，表示写入时，只写入计数初值低8位，高8位置0；读出时，只读出低8位的当前计数值。

$D_5D_4=10$ ，表示写入时，只写入计数初值高8位，低8位置0；读出时，只读出高8位的当前计数值。

$D_5D_4=11$ ，计数初值为16位，分两次读/写入计数初值寄存器，先读/写低8位，后读/写高8位。

控制字

## 5. 8254的编程

(3)  $D_3$ 、 $D_2$ 、 $D_1$ 为工作方式选择位。

$D_3D_2D_1$ 取值000到101分别代表方式0到方式5。

(4)  $D_0$ 为计数格式选择位

$D_0=0$ ，按二进制格式计数； $D_0=1$ ，按BCD码格式计数。

控制字

## 2. 8254的读出控制字

读出控制字 $D_7D_6$ 必须为11， $D_0$ 必须为0，这三位合起来构成8254的读出控制字的标志。


$D_5=0$ 锁存计数值，以便CPU读取；

$D_4=0$ 将状态信息锁存进状态寄存器；

$D_3\sim D_1$ 用来选择计数器0~2，无论是锁存计数值还是锁存状态信息，都不影响计数。

读出命令可以同时锁存3个计数器的计数值/状态信息，CPU读取其中一个计数器的计数值/状态信息时，该计数器自动解锁，其他计数器不受影响。

读出控制字

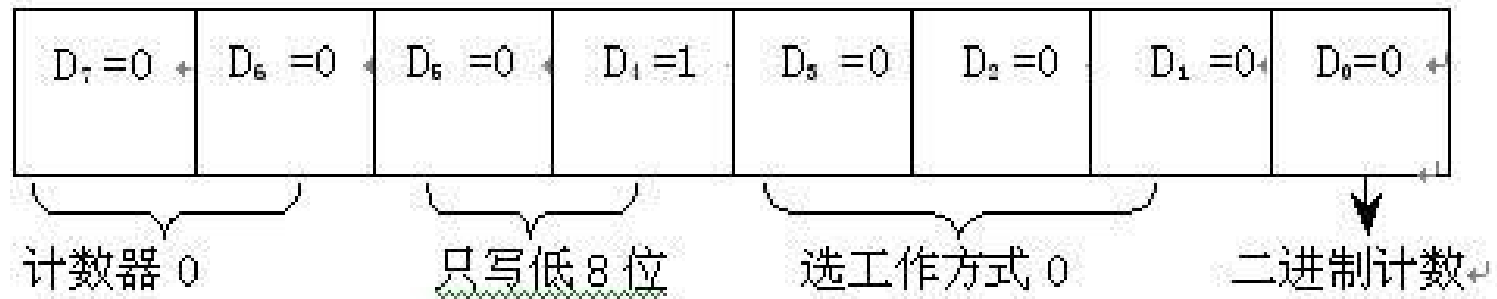


例1: 某微机系统中8254的端口地址为40H—43H, 要求计数器0工作在方式0, 计数初值为0DEH, 按二进制计数; 计数器1工作在方式2, 计数初值为1000D, 按BCD码计数。试写出初始化程序段。





按要求计数器0的控制字为：





计数器1的控制字为：

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



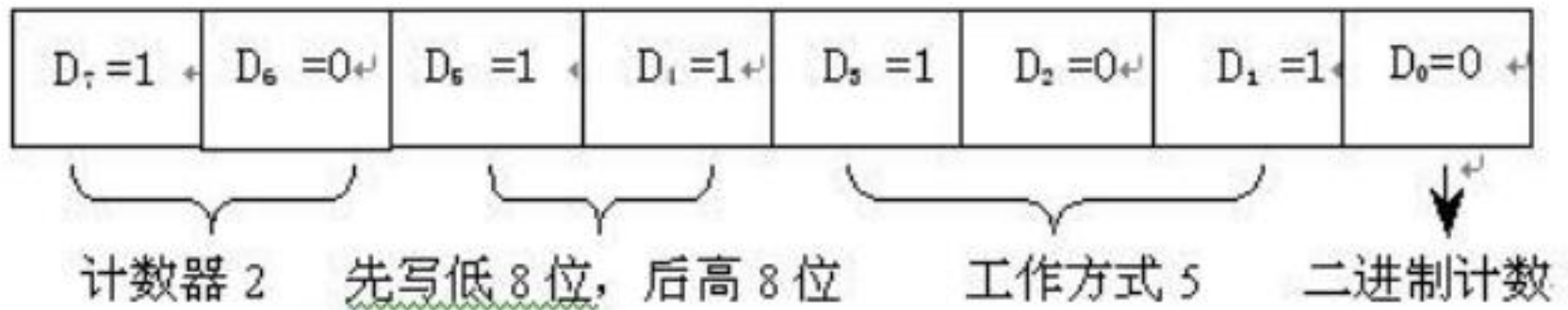


初始化程序如下：

MOV	AL, 10H	； 写通道0控制字
OUT	43H, AL	
MOV	AL, 0DEH	； 写通道0计数初值
OUT	40H, AL	
MOV	AL, 65H	； 写通道1控制字
OUT	43H, AL	
MOV	AL, 10H	； 写通道1计数初值
OUT	41H, AL	

初始化

例2: 设8254端口地址为3FF0H—3FF3H, 要求计数器2工作在方式5, 二进制计数, 初值为2F30H。试按上述要求完成8254的初始化。



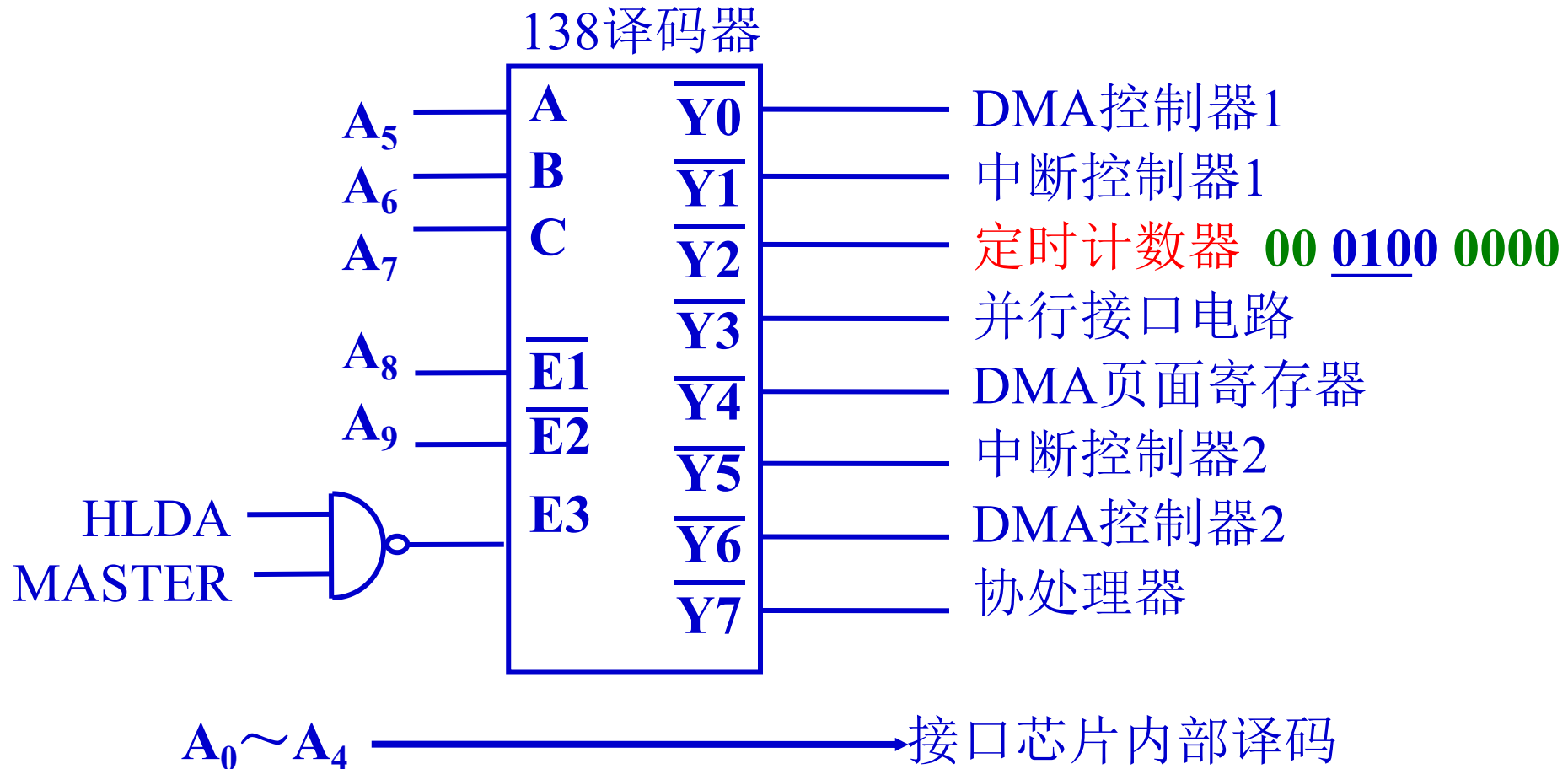
初始化



初始化程序如下：

```
MOV      DX, 3FF3H      ; DX指向控制端口
MOV      AL, 0BAH       ; 控制字
OUT      DX, AL
MOV      DX, 3FF2H      ; DX指向通道2
MOV      AL, 30H        ; 写初值低8位
OUT      DX, AL
MOV      AL, 2FH        ; 写初值高8位
OUT      DX, AL
```

# IBM PC/AT主机板的I/O译码电路



## 5 8254的编程

各种方式的工作过程类似，总结如下：

- (1) 设定工作方式
- (2) 设定计数初值
- 〔 (3) 硬件启动 〕
- (4) 计数初值进入减1计数器
- (5) 每输入一个时钟计数器减1的计数过程
- (6) 计数过程结束

# 产品信息



\*新老包装不定期更新, 请以收到的实物为准

产品名称: 经典原味豆奶粉

产品品牌: 永和豆浆

产品规格: 510g内含17小包

产品配料: 大豆(添加量49%)、白砂糖、麦芽糖、果葡糖浆、全脂奶粉、花生油、碳酸钙、食用盐、维生素A、维生素D、食品用香精  
加工原料大豆为非转基因

保质期: 12个月

生产日期: 见外包装(日期不断更新)

储存方法: 阴凉, 干燥, 常温处。每

后及时将袋口封严, 防止

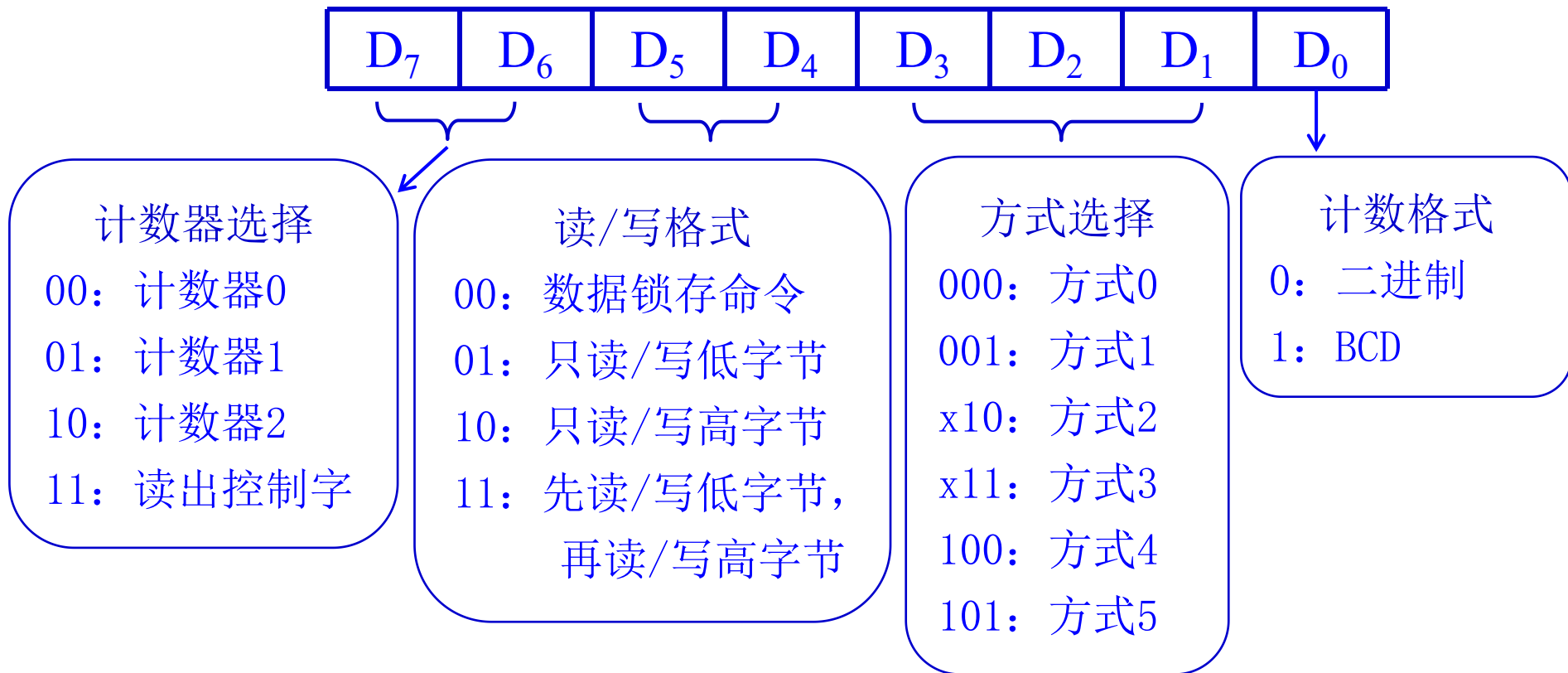


返回



## 8.1.3 8254的编程

### 1. 8254的方式控制字



## 8.1.3 8254的编程

### 1. 8254的方式控制字



#### 计数器选择

00: 计数器0

01: 计数器1

10: 计数器2

11: 读出控制字

## 8.1.3 8254的编程

### 1. 8254的方式控制字



#### 读/写格式

00: 数据锁存命令

01: 只读/写低字节

10: 只读/写高字节

11: 先读/写低字节，再读/写高字节

## 8.1.3 8254的编程

### 1. 8254的方式控制字



#### 方式选择

000: 方式0

001: 方式1

x10: 方式2

x11: 方式3

100: 方式4

101: 方式5

## 8.1.3 8254的编程

### 1. 8254的方式控制字



计数格式

0: 二进制

1: BCD