

第10章 定时/计数技术



本章教学内容

- 定时与计数器的基本概念；
- 可编程定时/计数器82C54，包括内部结构、方式命令、初值的计算和装入等；
- 6种工作方式：方式0-方式5的特点，GATE对它们的影响等。
- 初始化编程举例；

重点：82C54的工作原理、端口地址、方式字的含义、时间常数的计算、6种工作方式的特点，如何对定时器编程初始化



定时与计数



微机系统中的定时类型



外部定时方法及硬件定时器



可编程定时/计数器82C54A



定时/计数器的应用

10.1 定时与计数

1、定时

- ① 一天24小时的计时，称为日时钟
- ② 在监测系统中，对被测点的定时取样
- ③ 打印机忙信号等待10s进行超时处理
- ④ 在读键盘时，为去抖，一般延迟一段时间，再读
- ⑤ 在步进电机速度控制程序中，利用在前一次和后一次发送相序代码之间延时的时间间隔来控制步进电机的转速

2、计数

- ① 对零件和产品的计数
- ② 对大桥和高速公路上车流量的统计等等

3、定时与计数的关系

定时与计数本质上都是计数，但它们所计的‘数’来源是不同的。

10.2 微机系统中的定时类型

1、内部定时

内部定时产生运算器、控制器等CPU内部的控制时序，如取指周期、读/写周期、中断周期等，**主要用于CPU内部指令执行过程的定时**，计时单位一般是ns级。

2、外部定时

外部定时是**外设在实现某种功能时所需要的一种时序关系**。例如，打印机接口标准Centronics，就规定了打印机与CPU之间传输信息应遵守的工作时序。

外部定时和内部定时是相互独立的两个定时系统。内部定时固定不变，外部定时随外设而变。以内部定时为依据设计外部定时结构，即“时序配合”。

10.3 外部定时方法及硬件定时器

1、定时方法

(1) 软件定时：是利用CPU内部定时机构，运用软件编程去循环执行一段程序而产生的等待延时。

例： MOV BX,0FFH

DELAY: DEC BX ;BX称为“延时常数”

 JNZ DELAY

优点：不需增加硬件设备，只需编制相应的延时程序以备调用

缺点：CPU执行延时等待时间，增加了CPU的时间开销。延时时间越长，CPU的效率也越低，浪费了CPU的资源，另外还受主频的影响。

(2) 硬件定时：采用可编程的定时/计数器或单稳延时电路产生定时或延时。

优点：

- ① **不占用CPU的时间，定时时间长，使用灵活；**
- ② **定时准确，定时时间不受主机频率影响，定时程序具有通用性，故得到广泛应用。**

缺点：设置定时电路，增加额外硬件开销。

2、外部硬件定时器

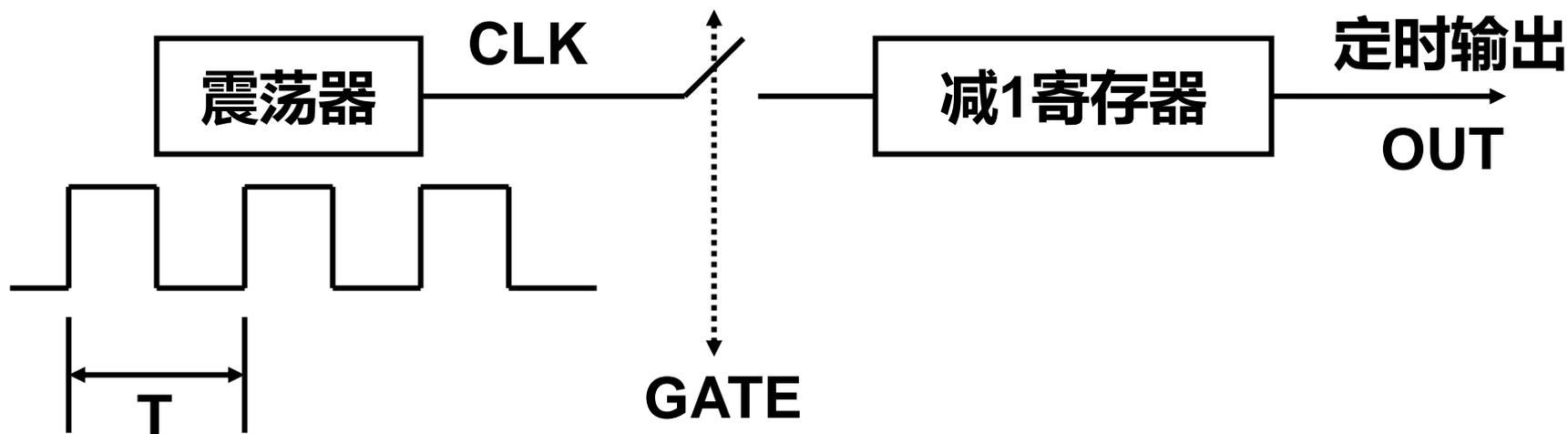
- **不可编程定时器**

不可编程定时器的**定时间隔**和**定时范围**固定，采用中小规模集成电路器件构成的定时电路。常见的定时器件有**单稳触发器**和**555、556定时器**等，利用其外接电阻、电容的组合，可实现一定范围的定时。

- **可编程定时器**

可编程定时器的**定时间隔**和**定时范围**可由**程序进行设定和改变**，使用方便灵活。可编程定时电路一般都是采用**可编程定时/计数器**外围支持**芯片**，如Intel 82C54A（8254是8253的改进型）。

• 计数器/定时器的的工作原理



定时长度 = $T * (\text{减1寄存器的初始值})$

定时器三要素:

- 稳定、准确的振荡频率。
- GATE控制定时开始。
- 定时时间到后输出的波型。

10.4 可编程定时/计数器82C54A

10.4.1 82C54的外部连接特性与内部结构

1、外部特性

(1)面向CPU的信号线

数据线：D0-D7

地址线：片选 \overline{CS} ，片内地址 A_0 、 A_1

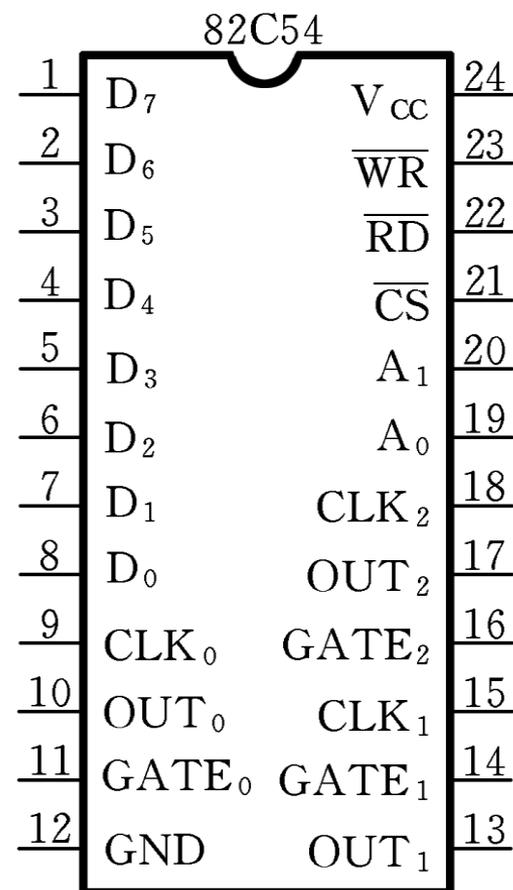
控制线： \overline{WR} 、 \overline{RD}

(2)面向I/O设备的信号线

时钟信号：CLK0-CLK2

门控信号：GATE0-GATE2

输出信号：OUT0-OUT2



面向CPU的信号线

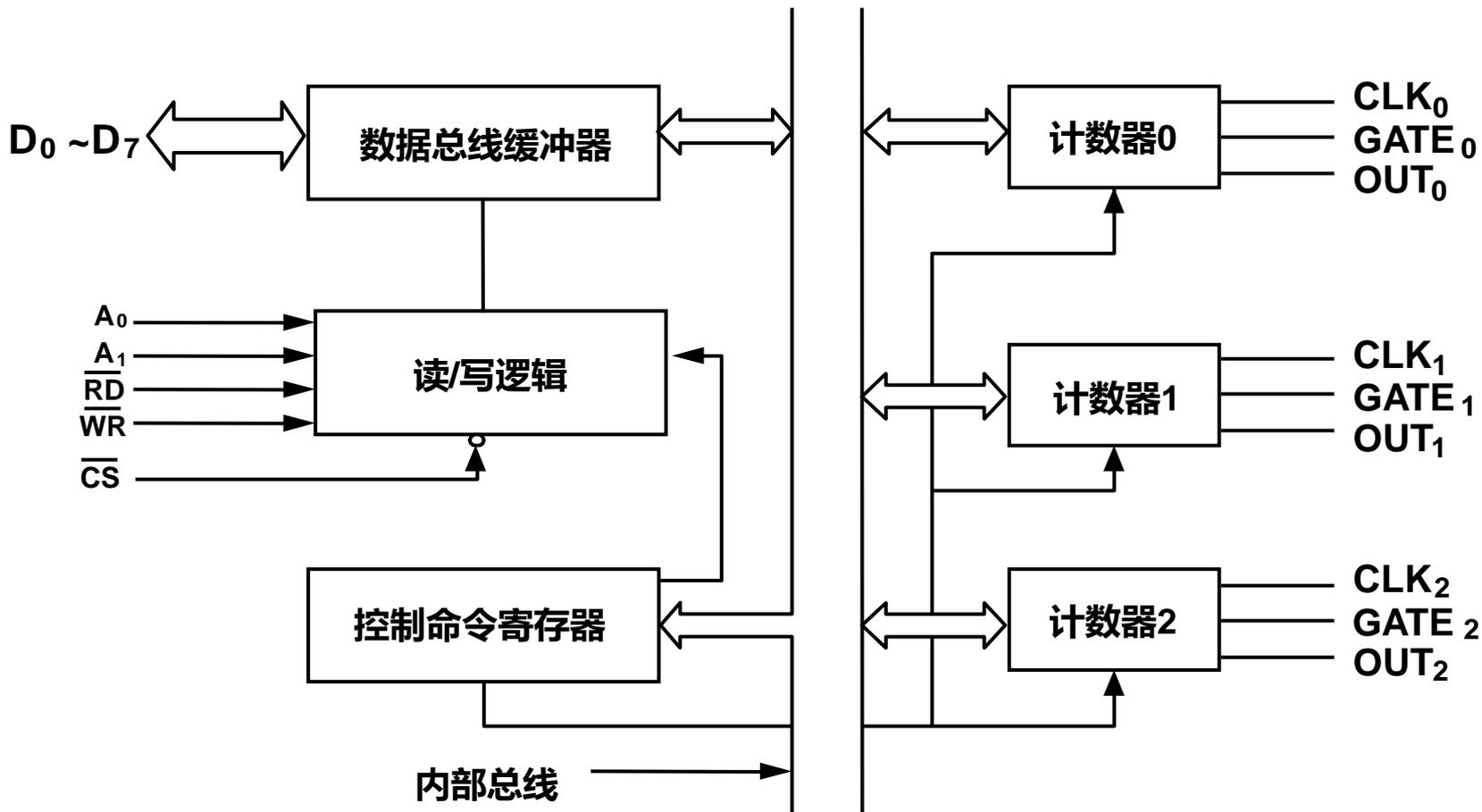
- ① **数据总线 D_0-D_7 ：三态输出输入线。** 用于将8254与系统数据总线相连
- ② **片选线 \overline{CS} ：输入信号，低电平有效。** 当为低电平时，CPU选中82C54，并可以对82C54进行读/写操作；当为高电平时，CPU没有选中82C54。 \overline{CS} 由CPU输出的地址码经译码产生。
- ③ **读信号 \overline{RD} ：输入信号，低电平有效。**
- ④ **写信号 \overline{WR} ：输入信号，低电平有效。**
- ⑤ **地址线 A_1, A_0 ：这两根线接到系统地址总线的 A_1, A_0 上，用于片内寻址** (8254有4个I/O端口地址，按从低到高地址的字节顺序为：通道0、1、2、方式字（命令寄存器），各占一个字节宽度的I/O地址。如果要向其所在I/O地址传送两个字节，那么需要连续两次向该IO地址传输内容)。

面向I/O设备的信号线

- **计数器时钟信号CLK：输入的计时脉冲信号**
- **计数器门控选通信号GATE：输入的计时启动信号**
- **计数器输出信号OUT：定时结束时的输出信号**

2、内部结构

82C54A内部6个模块

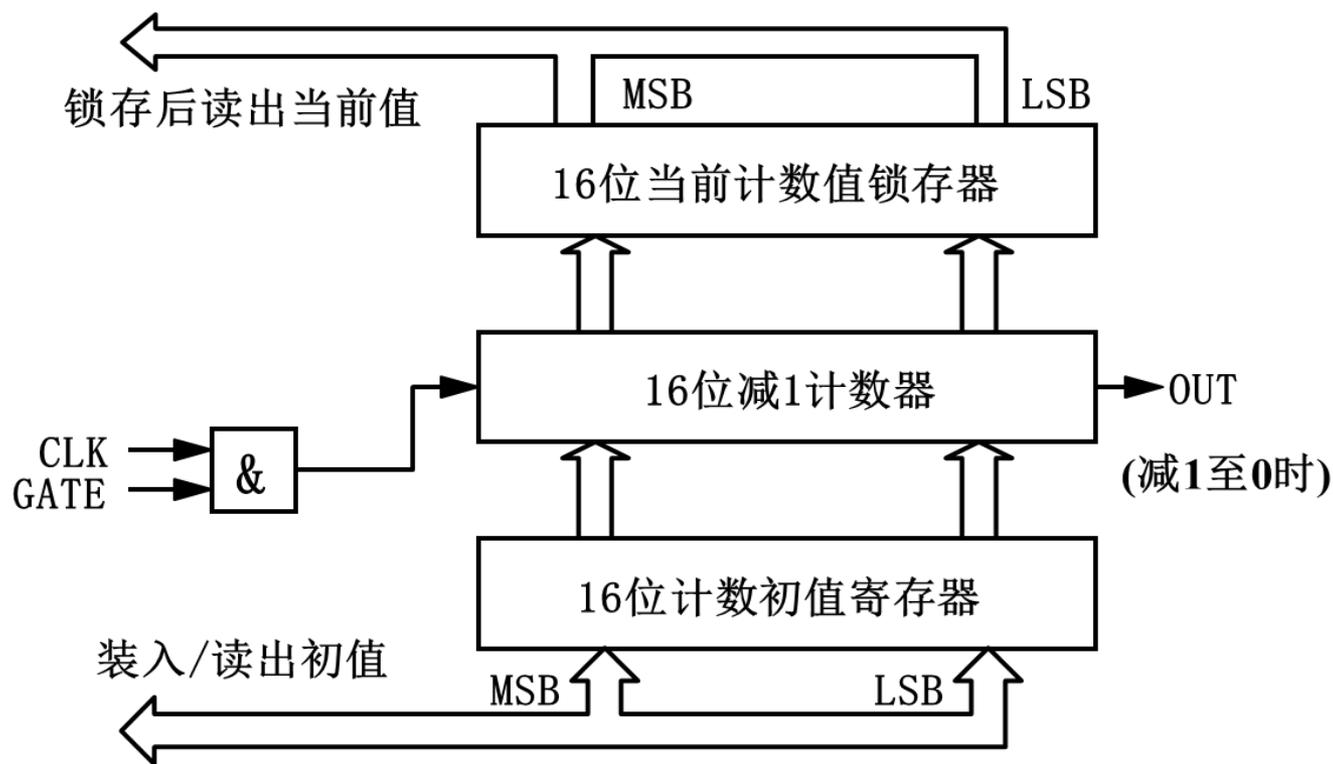


各组成部分的含义：

完成CPU和8254之间命令、数据、地址和数据的交换

- ① **数据总线缓冲器**：它是一个三态、双向8位寄存器，用于将8254与系统数据总线 D_0-D_7 相连。
- ② **读/写逻辑**：用于读/写控制和片选。
- ③ **控制命令寄存器**：它接收CPU送来的控制字。
- ④ **计数器**：8254有3个独立的计数器，其内部结构完全相同，每个计数器由16位的**计数初值寄存器**、**减法计数器**和**当前计数值锁存器**组成。

• 计数器通道的组成



计数通道内部逻辑图

- **计数初值寄存器 (16位)**

用于存放计数初值 (定时常数、分频系数)，其长度为16位，故最大计数值为65536 (64KB)。在初始化时同减1计数器的初值一起装入。计数初值寄存器的计数初值，在计数过程中保持不变。[数据线是8位，需要装2次]

- **减1计数器 (16位)**

用于进行减1计数操作，每来一个时钟脉冲，它就作减1运算，直至将计数初值减为零。

- **当前计数值锁存器 (16位)**

用于锁存减1计数器的内容，以供读出和查询。

8254读写操作及端口地址

/CS	/RD	/WR	A₁	A₀	操 作
0	1	0	0	0	加载T/C ₀ (向计数器0写入“计数初值”)
0	1	0	0	1	加载T/C ₁ (向计数器1写入“计数初值”)
0	1	0	1	0	加载T/C ₂ (向计数器2写入“计数初值”)
0	1	0	1	1	向控制器寄存器写“方式控制字”
0	0	1	0	0	读T/C ₀ (从计数器0读出“当前计数值”)
0	0	1	0	1	读T/C ₁ (从计数器1读出“当前计数值”)
0	0	1	1	0	读T/C ₂ (从计数器2读出“当前计数值”)
0	0	1	1	1	无操作三态
1	×	×	×	×	禁止三态
0	1	1	×	×	无操作三态

10.4.2 82C54A的命令字

- 82C54有3个命令字和1个状态字
- 3个命令字：方式命令、锁存命令和读回命令。
其中**方式命令是必须的**，其它两个命令根据需要
使用。

※ 注意：

这3个命令字使用同一个端口，按方式命令在先，
其它命令在后的顺序写入端口。

1、方式命令

初始化82C54A，包括选定计数通道、设定工作方式、确定字节读写顺序以及计数值码制

方式命令的格式：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
计数器选择		读写字节数		工作方式			码制

D₇D₆ = 00, 选择0号计数器;
D₇D₆ = 01, 选择1号计数器;
D₇D₆ = 10, 选择2号计数器;
D₇D₆ = 11, 无效。

D₃D₂D₁ = 000, 选择0方式;
D₃D₂D₁ = 001, 选择1方式;
D₃D₂D₁ = X10, 选择2方式;
D₃D₂D₁ = X11, 选择3方式;
D₃D₂D₁ = 100, 选择4方式;
D₃D₂D₁ = 101, 选择5方式。

1、方式命令

初始化82C54A，包括选定计数通道、设定工作方式、确定字节读写顺序以及计数值码制

$D_0 = 0$ ，选择二进制计数；
 $D_0 = 1$ ，选择十进制BCD计数。

方式命令的格式：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
计数器选择		读写字节数		工作方式			码制

$D_5D_4 = 00$ ，锁存命令，计数器值复制到锁存器， $D_{3\sim 0}$ 无效；
 $D_5D_4 = 01$ ，仅读/写一个低字节，高字节为0；
 $D_5D_4 = 10$ ，仅读/写一个高字节，低字节为0；
 $D_5D_4 = 11$ ，读/写2个字节，先是低字节，后是高字节。
影响对锁存器的读和对初值寄存器写操作

有关计数码制:

■ 若按二进制计数

- 计数范围: 16位二进制数
- 最大计数值: $2^{16} = 65536$, 对应计数初值0000H
- 注: 需要将计算得到的十进制计数初值n转换成二进制, 然后写入指定端口

■ 若按BCD码计数

- 计数范围: 4位十进制数
- 最大计数值: $10^4 = 10000$, 对应计数初值0000
- 注: BCD常用于用户输入, 计数值应由16进制转换为BCD码才行

例：使用**计数器T₁**工作在**4方式**，且时间常数为3F8H；
设307H是整个8254的命令寄存器的地址，305H是8254
中定时器1的数据口地址，则**初始化程序段**为：

```
MOV DX, 307H      ; 命令口
MOV AL, 01111000B ; 方式字
OUT DX, AL        ; 写入控制命令字寄存器
MOV DX, 305H      ; T1数据口
MOV AL, 0F8H      ; 低8位计数值
OUT DX, AL        ; 送低字节到计数器1初值寄存器
MOV AL, 03H       ; 高8位计数值
OUT DX, AL        ; 送高字节到计数器1初值寄存器
```

二进制和BCD使用过程中的注意:

例: 设8254的口地址为40H ~ 43H, 要求2#计数器工作在方式3, 计数初值=4000, 写出初始化程序段

■ 若按二进制计数

MOV AL, 10110110B (设初值为2进制, 初值写入顺序为先低8位后高8位)

OUT 43H, AL

MOV AX, 4000

OUT 42H, AL

MOV AL, AH

OUT 42H, AL

设初值为2进制, 写入顺序为仅写高8位

```
1 | MOV AL, 10100110B
2 | OUT 43H, AL
3 | MOV AL, 40
4 | OUT 42H, AL
```

注意: 解2是错误的, 因为4000=0FA0H, 其低8位不等于0。

■ 若按BCD计数

MOV AL, 10110111B (设初值为BCD码, 写入顺序为先低后高)

OUT 43H, AL

MOV AL, 0

OUT 42H, AL

MOV AL, 40H

OUT 42H, AL

设初值为BCD码, 仅写高8位

```
1 | MOV AL, 10100111B
2 | OUT 43H, AL
3 | MOV AL, 40H
4 | OUT 42H, AL
```

注意: 4000的BCD码数应写成4000H, 解3、解4是正确的。

程序中如何表示BCD码数？

设初值 = 4000且采用BCD码计数

则 `MOV AX, 4000H` 是正确的

写成 `MOV AX, 4000` 是错误的

BCD码表示的就是没有A~F的16进制数，而十进制数转换成BCD是按每一位来表示的。所以十进制==BCD码。

如果要存一个4000的数 存二进制，存的就是0FA0H

存BCD码，存的是4000H。

一个普通的十进制数，存到寄存器中，就自动转换为2进制了，低8位和高8位就会出错。

而把十进制数用BCD码来存，低8位和高8位就不会出问题。

小节:

设初值 = 4000且采用BCD码计数

- 若按二进制计数

MOV AX, 4000 或 MOV AX, 0FA0H

然后先低后高out

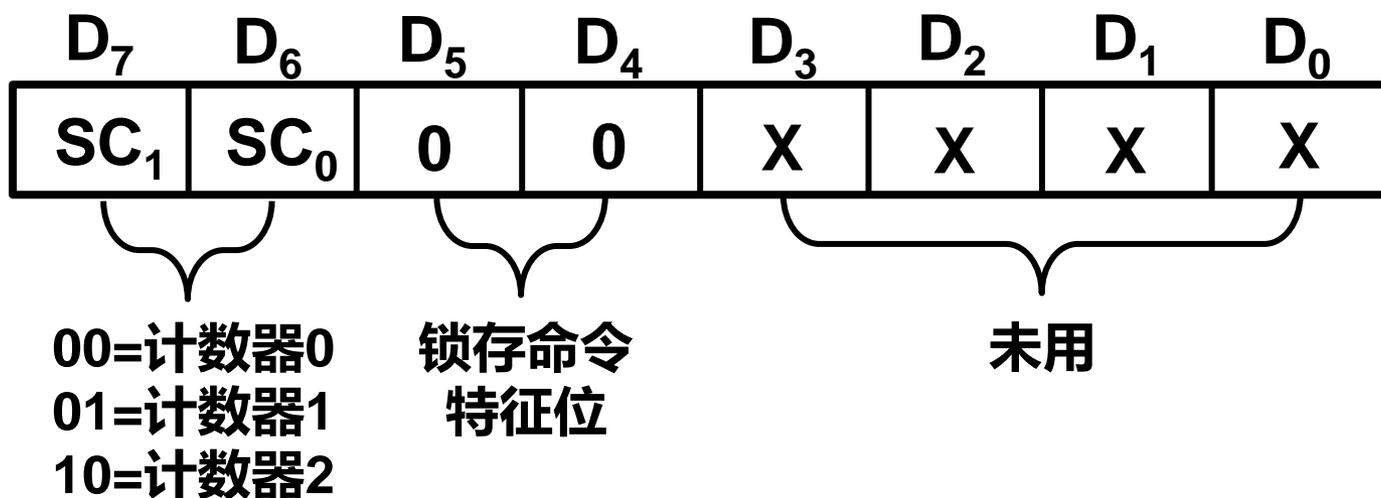
- 若按BCD计数

MOV AX, 4000H

然后先低后高out

2、锁存命令

将减1寄存器的内容锁存到输出寄存器中以供CPU读取。执行锁存命令指示把计数器的当前值锁存到暂存寄存器。锁存命令的格式：



● 读当前计数值为什么要先锁存计数值？

减1计数器是16位的，而定时器的数据线是8位的，必须读两次才能读出16位数据，在两次读操作时可能使减1计数器的内容发生变化

第1次读减1计数器中的值

第2次读减1计数器中的值

0 0 0 0 0 0 0 1 | 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1



若先读高字节 { 00000001
11111111

若先读低字节 { 00000000
00000000

例：要求在计数器通道1的计数过程中读取当前的计数值，并把读取的计数值装入到AX寄存器中。设4个端口的地址为：304H（通道0），305H（通道1），306H（通道2），307H（命令寄存器）

命令口

数据口

```
MOV DX, 307H
```

```
MOV AL, 0100XXXXB ; 锁存计数器1
```

```
OUT DX, AL
```

可以取任意数，一般取0

```
MOV DX, 305H
```

```
IN AL, DX ; 读低字节
```

```
MOV BL, AL
```

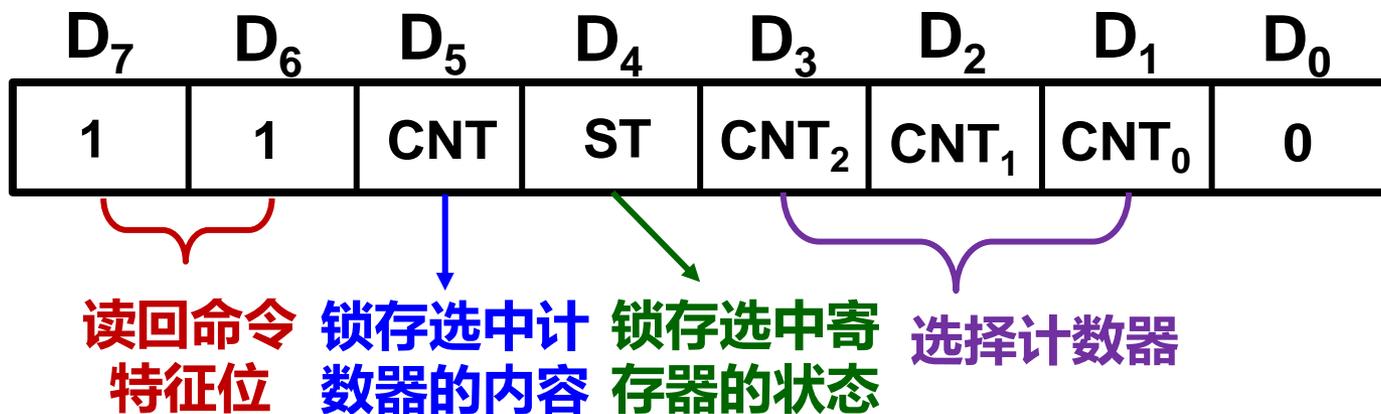
```
IN AL, DX ; 读高字节
```

```
MOV AH, AL
```

```
MOV AL, BL
```

3、读回命令(8253不具有这种功能)

读回命令与前面的锁存命令不同，它即能锁存计数值又能锁存状态信息，而且一条读回命令可以锁存3个计数通道的当前计数值与状态。读回命令的格式：



注意：读回命令只是一个锁存的功能，要读回计数值和状态还要发一条读命令。

含义:

D_1, D_2, D_3 用于选择3个计数器

1: 选中, 0: 未选中;

D_4, D_5 用于选择读当前状态还是当前计数值

0: 要读取, 1: 不读取

例题:

读取计数器2的当前计数值和状态

读回命令=11001000B

读回命令=11001110B

读取全部计数器的当前计数值和状态

4、状态字

- ✓ 状态字中的低6位D0-D5与方式字中的低6位是相同的;
- ✓ D6表示计数器的输出状态:
 - 1: 输出引脚为逻辑1
 - 0: 输出引脚为逻辑0
- ✓ D7表示是否有空计数值:
 - 1: 空计数值, 0: 计数值有效

注意: 8253没有此状态字

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
计数状态	输出状态	读写字节数		工作方式			码制

- 8254编程：
 - 先写控制字，再写计数初值；
 - 计数初值必须遵循控制字设定的格式；
 - 如果初值是两个字节，那么在载入初值期间则不允许发送控制字命令，否则会载入错误的计数器。
- 当减1计数器为0时，OUT端上输出一个脉冲。
- GATE=1时允许开始计数；GATE=0时计数值保持不变。
- 只有当CPU向8254控制字寄存器中写入控制字后，8254才开始定时/计数过程；
- 写入控制字后，所有的控制逻辑电路将立即复位，输出端OUT立即进入初始状态(不同的工作模式下OUT的初始状态是不同的)。
- GATE=1时，在写入计数初值后，需要经过一个时钟脉冲之后，计数器才开始在脉冲下降沿进行减1计数。

- 计数通道作为**计数器**用时：**对其CLK端上的脉冲[边沿]计数**，不关注CLK脉冲本身的频率、占空比等特征；
- 计数通道做**定时器**用时：所定的时间=CLK的周期***CLK的计数数目**；此时CLK端上的输入脉冲应当是标准的、精确的；

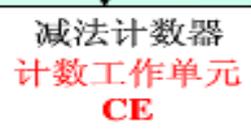
饮料生产线

基本计数工作原理：



CLK

计数初值N



OUT

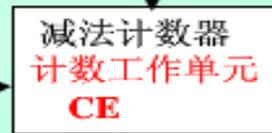
根据不同的工作方式，**OUT**输出不同的信号

基本定时工作原理：



CLK

计数初值N
(定时系数)



OUT

定时系数 = 要求定时的时间 / 时钟脉冲周期

注意！！

定时的本质还是**计数**，计数的对象必须是频率精确的脉冲串。如果**时钟脉冲**的周期是**1μs**，要求定时**1ms**，则计数**1000**次即可，即：**定时系数**为**1000**；
(对秒计数**60**次就是**1**分钟)

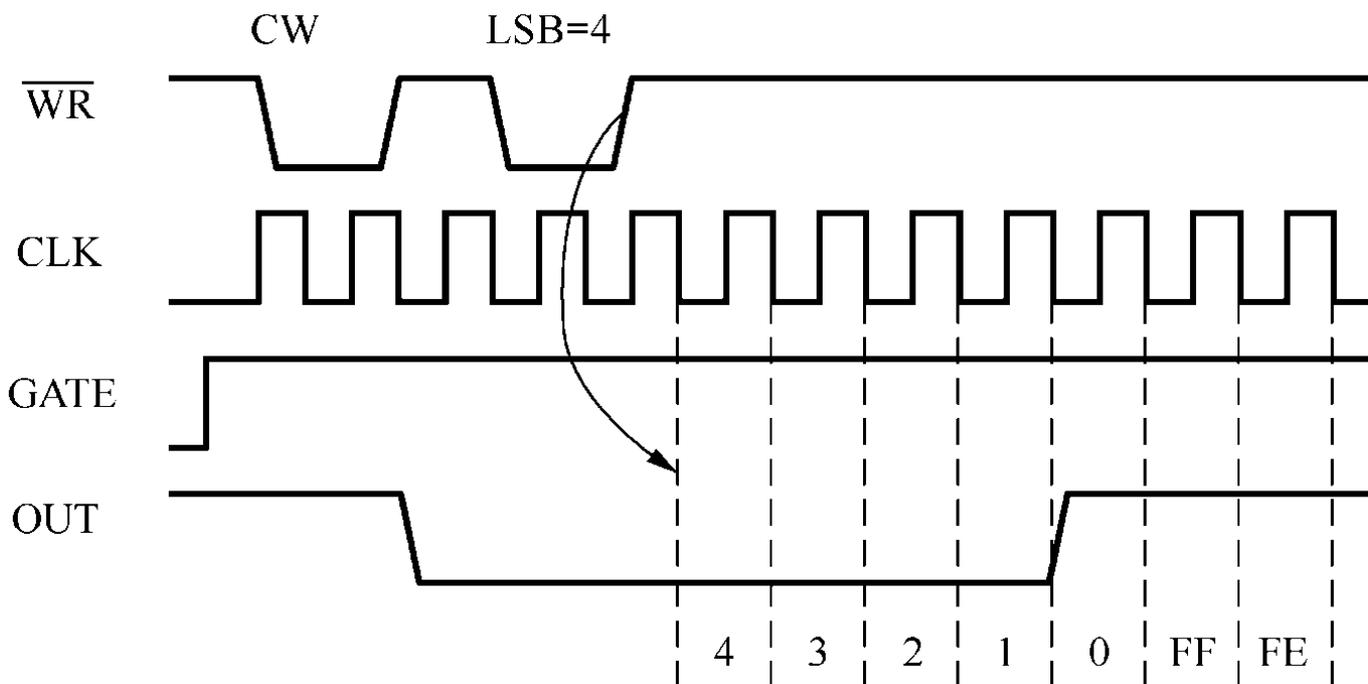
闹钟、分频器

10.4.3 82C54的工作方式

1、82C54在不同工作方式下的功能和特点

- 82C54有6种工作方式，不同的工作方式主要体现在输出波形、启动方式、初值装入，以及GATE信号的控制作用的不同，由**方式控制字**确定。
- 每种工作方式的过程类似：
 - 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

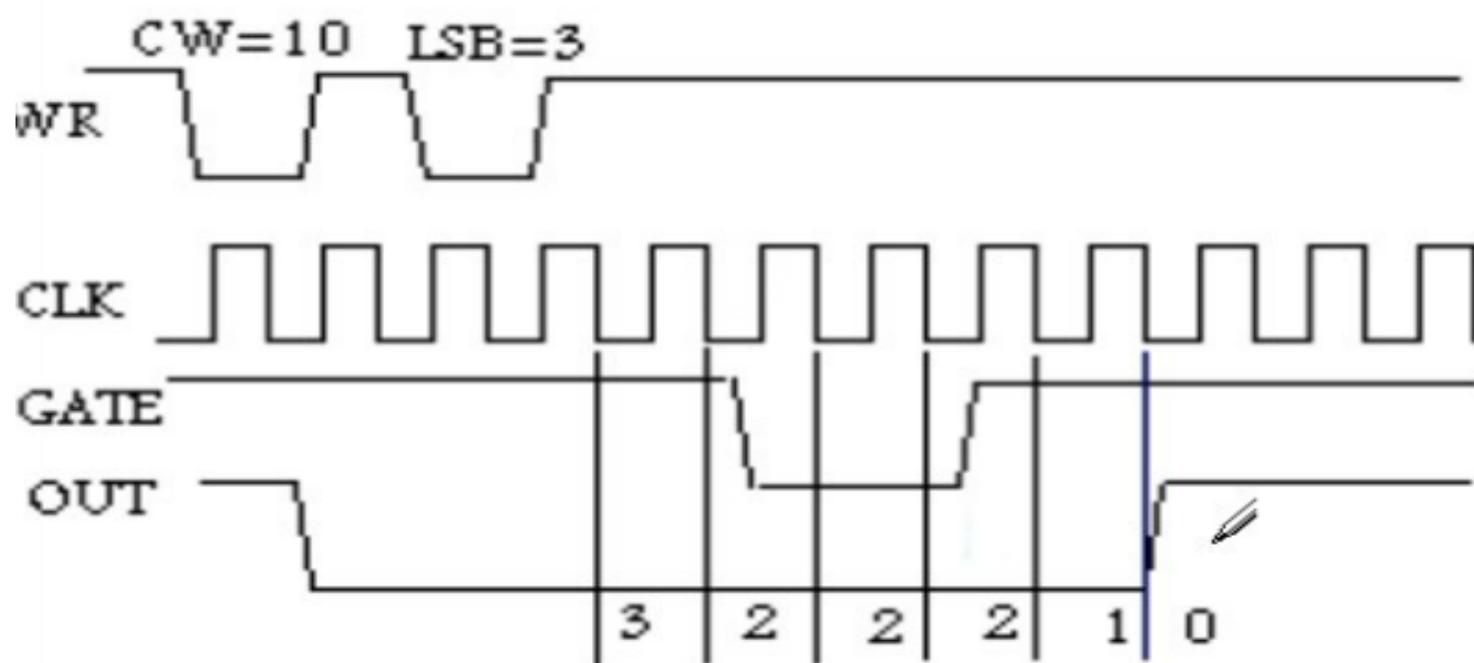
具体过程：首先CPU将控制字写入控制寄存器，并在写控制信号WR的上升沿，OUT输出变成低电平，然后CPU向计数初值寄存器CR写入计数初值，并且在WR上升沿之后的第一个CLK脉冲的下降沿，将CR的内容送入计数执行部件CE。当GATE为高电平时，在CR内容送入CE之后的每一个CLK脉冲下降沿，都使CE减1计数，并且在技术过程中OUT一直维持低电平，直到计数到零时，OUT输出才由低电平跳到高电平。可以用此OUT输出信号作为中断请求信号。



方式0过程特点:

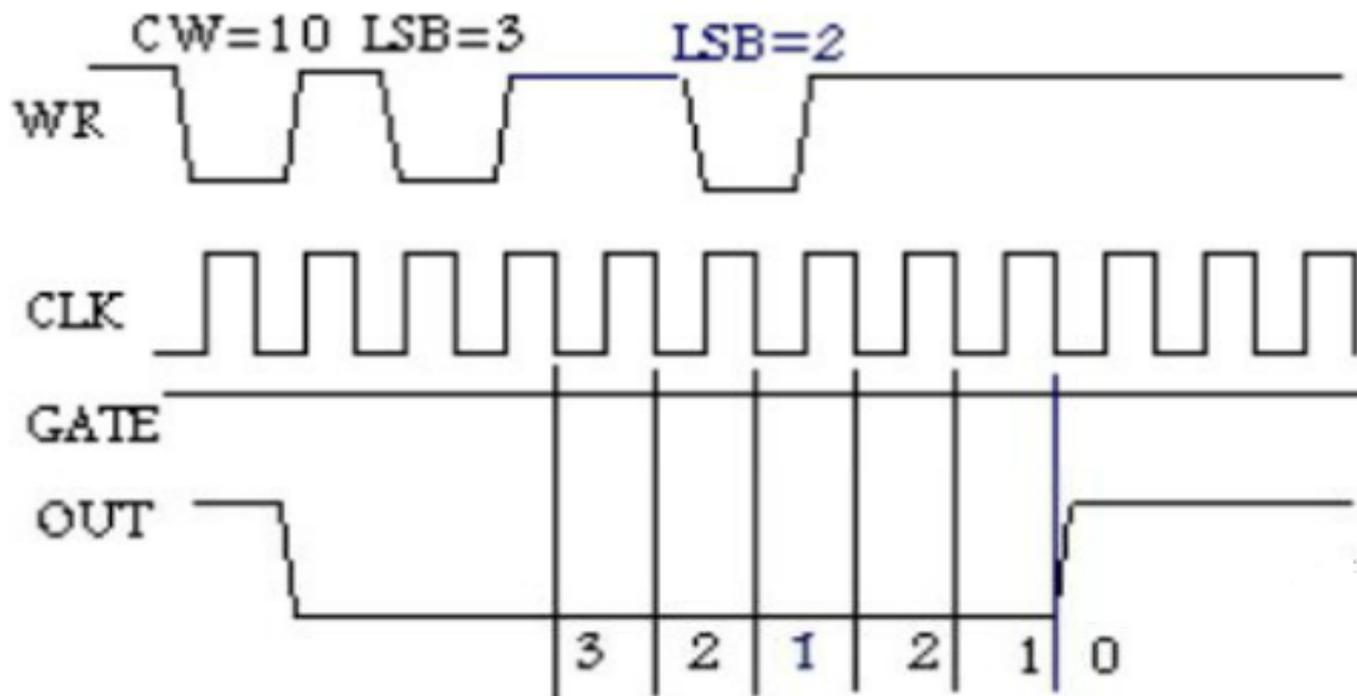
控制特征	OUT和计数值的变化
1 控制字写入	当控制字写入后, OUT变为低电平, 并在计数值为0前一直保持低电平
2 计数值写入	OUT=0
3 启动方式	写入非0计数值即可启动。 在写入计数值N后的下一个时钟脉冲时开始减1计数 (所以是N+1个CLK宽度)。
4 计数期间	OUT=0
5 计数为0时	OUT=1, 一直保持到有新计数值或控制字为止
6 计数期间写入新的计数值	如果计数值为1个字节, 在其写入后的下一个时钟脉冲时开始减1计数; 如果为2个字节, 则写入第一个字节时中止计数, 写入第2个字节后的下一个时钟脉冲时, 按新的计数值开始减1计数;
7 GATE的作用	GATE=0时, 计数器暂停减1; GATE=1时, 计数器继续减1;
8 计数值有效期	一次有效

GATE信号作用



计数期间写入新初值

10指10H
，后同



1. 定时器方式0：计数结束中断

0方式特点：

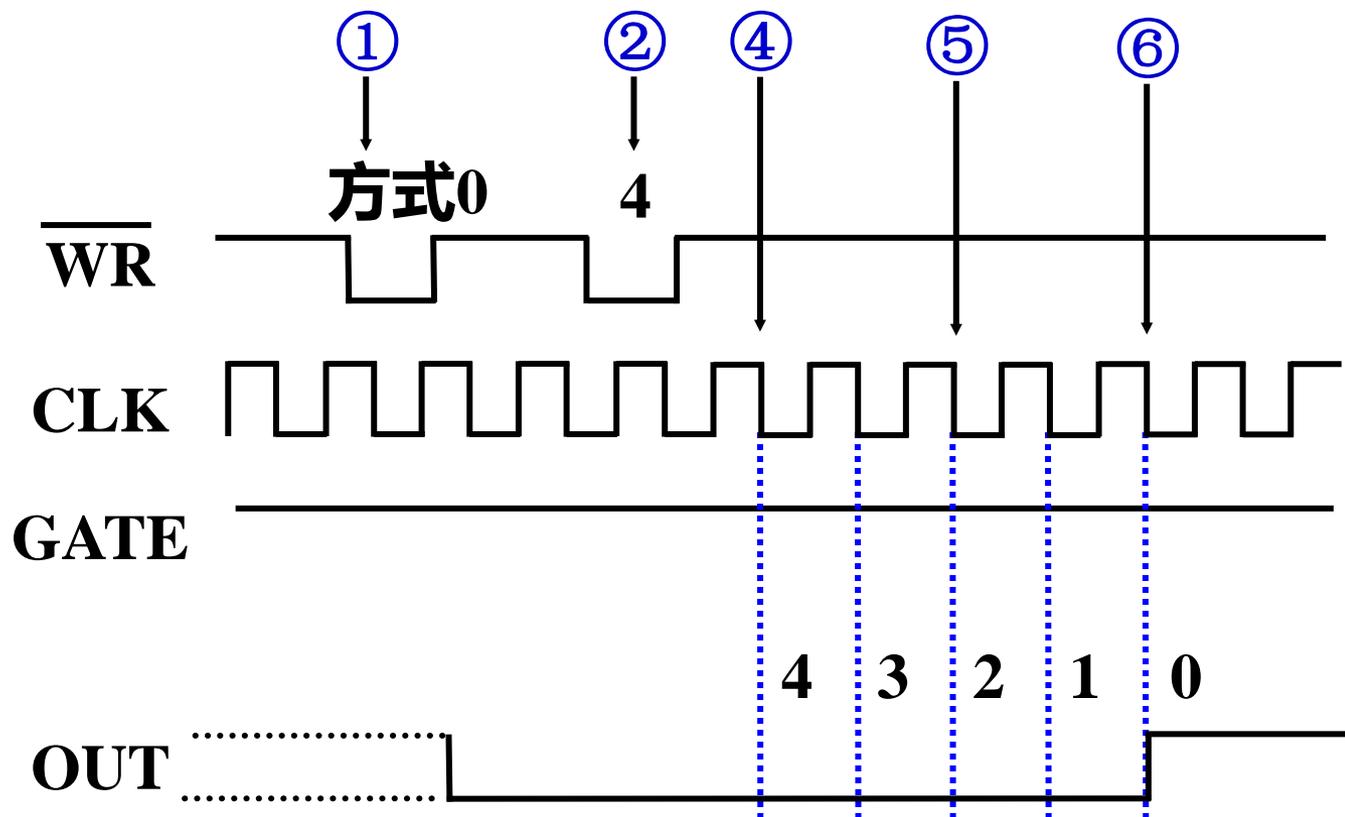
- 0方式的典型应用是做事件计数器，计数器的大小就是计数初值（初值+1），改变计数初值就可以改变计数器的大小。
- 0方式当写入计数初值后，启动计数器开始计数，OUT信号变为低电平，并维持低电平至减法计数器的内容减到0时，停止工作，OUT信号变为高电平，并维持高电平到再次写入新的计数值，可见，0方式输出单次波。
- 0方式的计数过程由写信号软启动，写入计数初值后开始计数，不需外加启动信号；计数结束，自动停止，不需外加停止信号。

1. 定时器方式0：计数结束中断

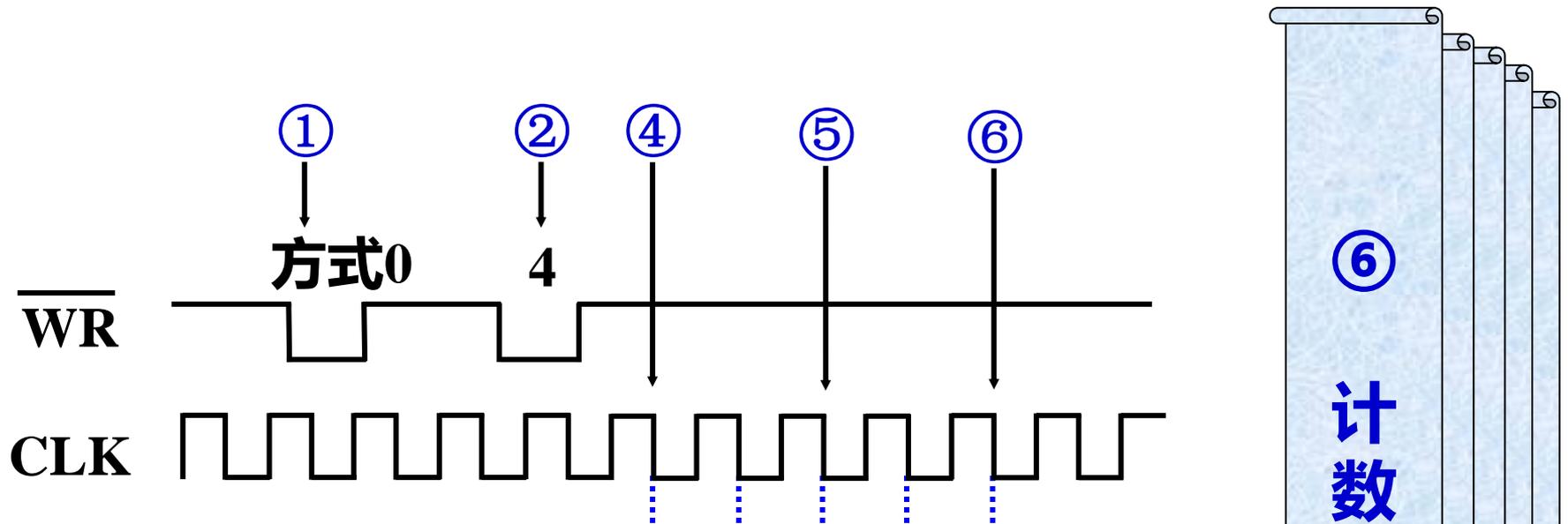
注意：

- 当向计数器写完计数值后，开始计数，计数一旦开始，输出端OUT就变成低电平，并在计数过程中一直保持低电平，当计数器减到零时，OUT立即变成高电平。
- 门控信号GATE为高电平时，计数器工作；当GATE为低电平时，计数器停止了工作，其计数值保持不变。如果门控信号GATE再次变高时，计数器从中止处继续计数。
- 在计数器工作期间，如果重新写入新的计数值，计数器将按新写入的计数初值重新工作。

1. 定时器方式0：计数结束中断



1. 定时器方式0：计数结束中断

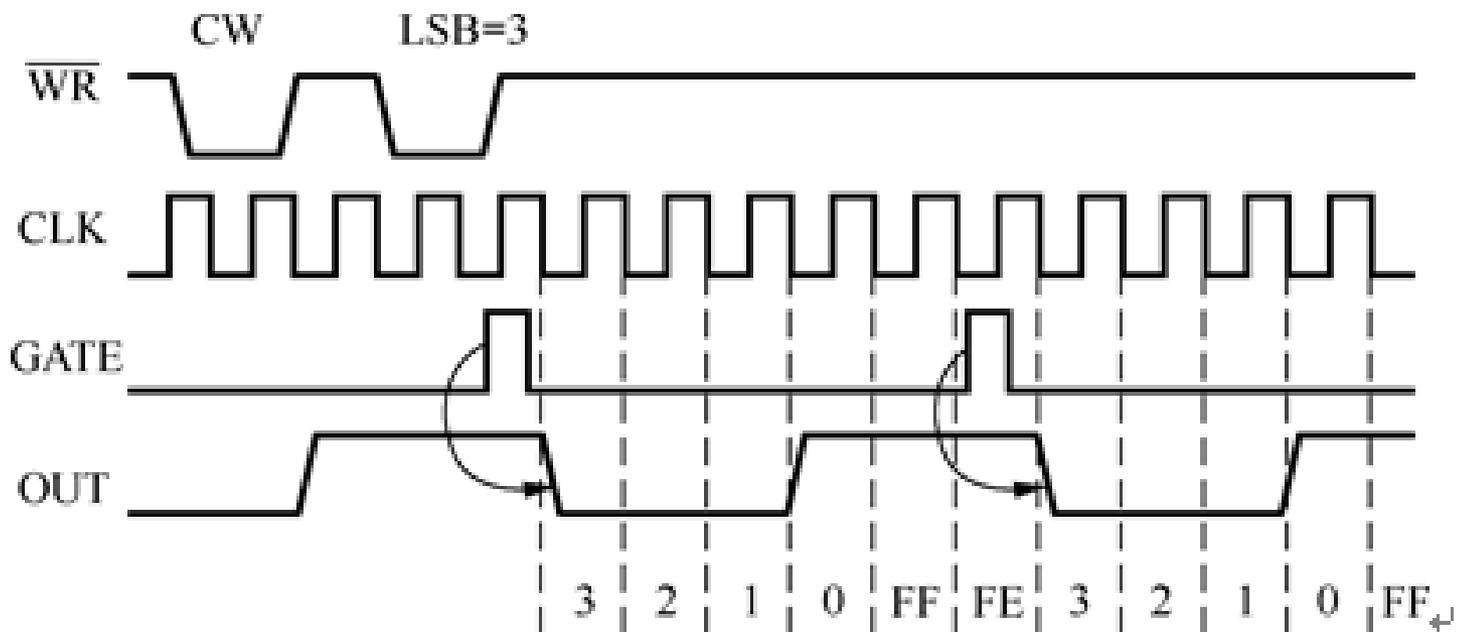


有一个细节请注意：

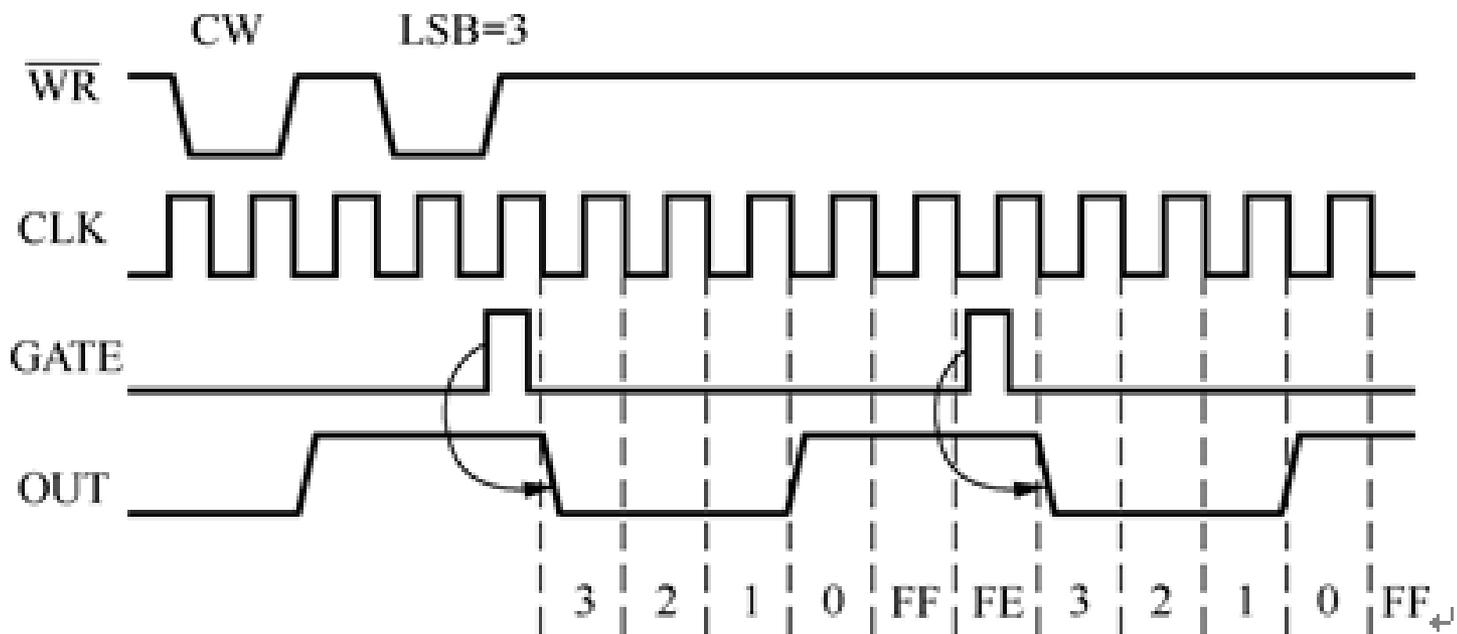
程序写完计数初值只是写入初值寄存器，之后到来的第一个CLK输入脉冲（先由低电平变为高电平，再由高电平变为低电平）才将初值送入减1计数器。从第二个CLK的下降沿才开始减1计数。

2. 定时器方式1：可编程单稳脉冲

1方式是硬件可重触发单稳态方式（每次触发产生单次输出波形）。计数过程由硬件启动，GATE出现0→1的跃变后开始计数；门控信号GATE用于计数过程的启动。



具体过程：当CPU输出控制字后随着WR的上升沿，OUT输出变为高电平；在CPU写入计数初值后，计数器并不开始计数，直至门控信号GATE上升沿（即门控触发信号）出现，并且在下一个CLK脉冲的下降沿，CR的内容送入CE，同时使OUT输出变为低电平，然后对随后的CLK脉冲进行减1计数。技术过程中，OUT一直维持低电平，直至减1计数器减至0时，OUT输出变成高电平。可见，由于GATE上升沿的触发，使得OUT输出端产生一个宽度为N个CLK周期的负脉冲。此后若再次由GATE上升沿触发，则输出再次产生一个同样宽度的负脉冲。



方式1过程特点:

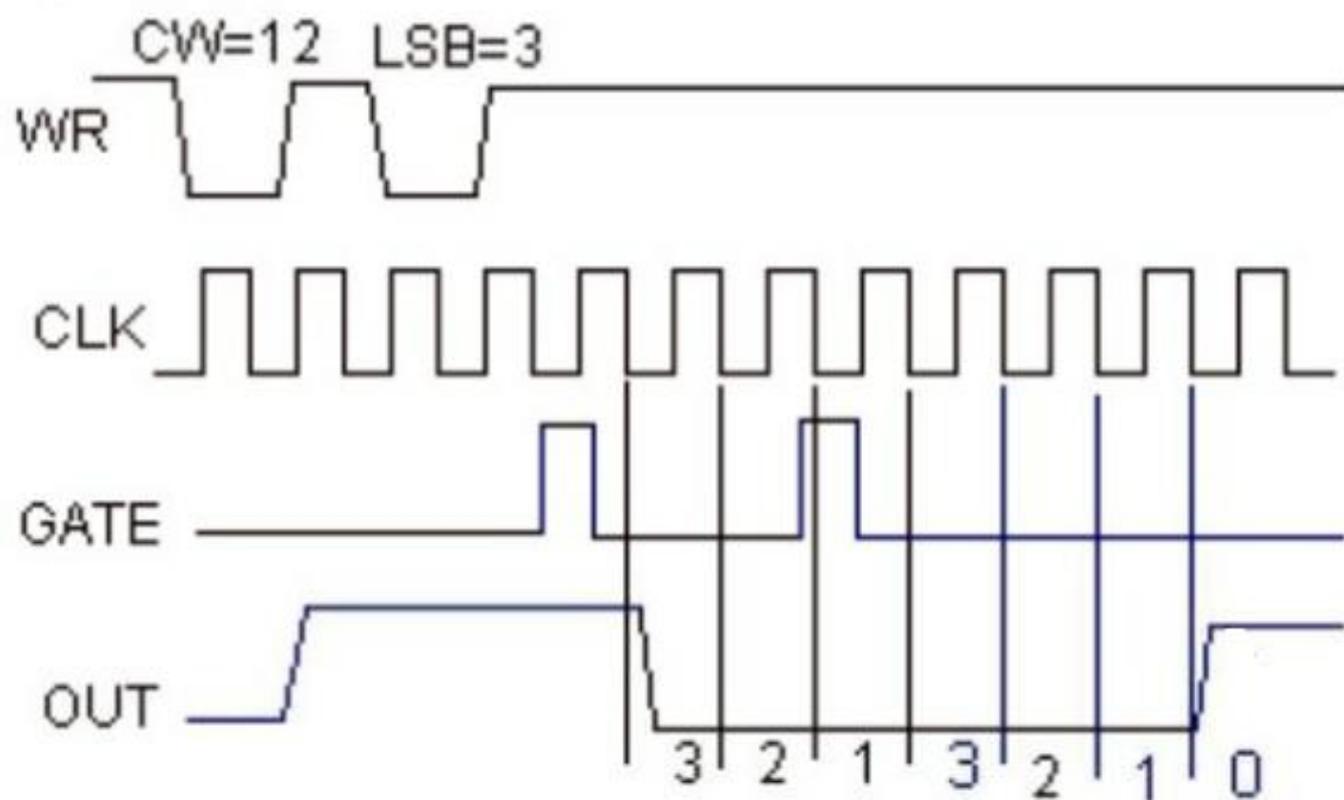
控制特征	OUT和计数值的变化
1 控制字写入	当控制字写入后, $OUT=1$
2 计数值写入	$OUT=1$
3 启动方式	GATE上升沿可启动 。在GATE上升沿后的下一个时钟脉冲时 $OUT=0$
4 计数期间	$OUT=0$
5 计数为0时	$OUT=1$, 一直保持到有GATE上升沿为止
6 计数期间写入新的计数值	不影响原计数过程, 只有当下一个GATE上升沿时才使用新计数值
7 GATE的作用	$GATE=0$ 或 1 期间, 不影响计数器减1动作; 但若出现GATE上升沿, 则按新计数值重启计数器, 无论原计数过程如何。
8 计数值有效期	多次有效; 未写入新计数值时, 每次GATE上升沿都自动重载原计数值

2. 定时器方式1：可编程单稳脉冲

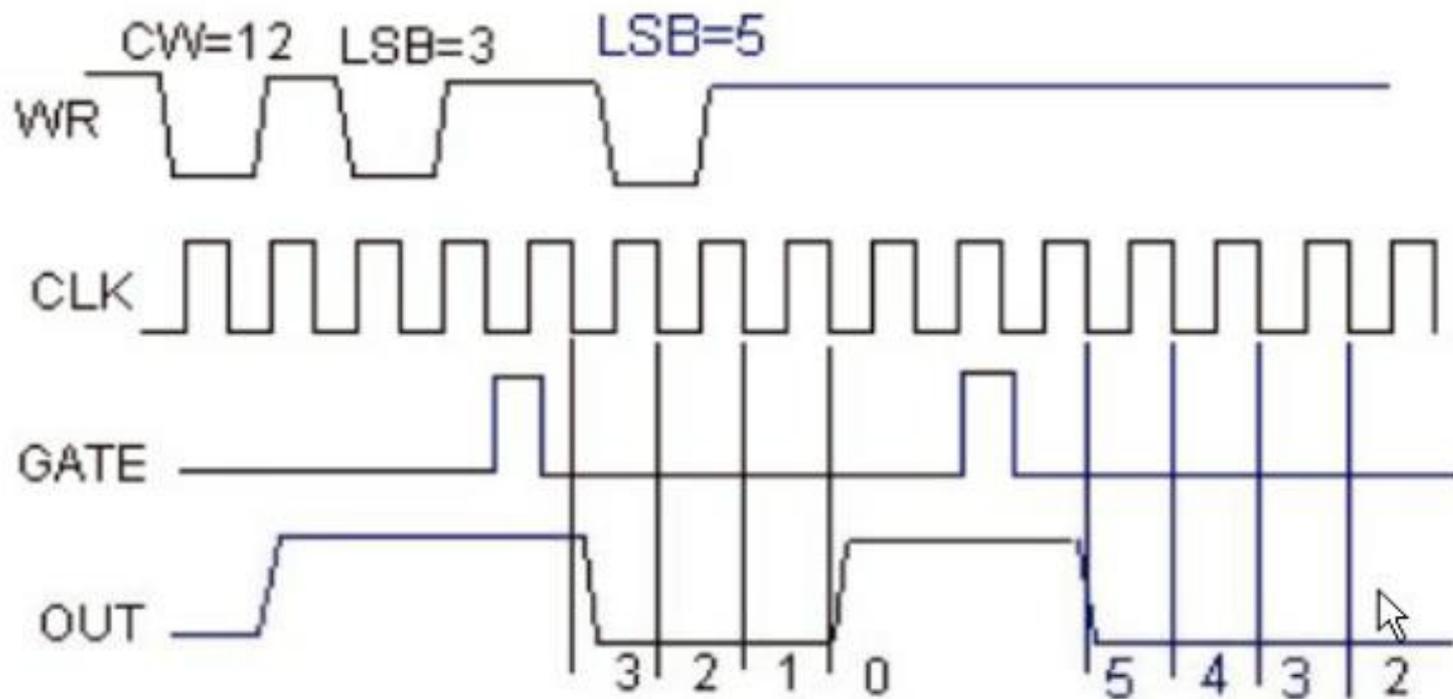
1方式特点：

- 1方式的典型应用是作可编程单稳态触发器，单稳延迟时间=计数初值×时钟脉宽，延时期间输出的是低电平，低电平的宽度可以由程序控制，即改变计数初值就可以改变延时时间。
- 1方式在写入计数初值后，再由GATE门信号硬启动计数，OUT变为低电平，每来一个CLK，计数器减1直到计数值减到0时，停止工作，OUT输出高电平，并维持高电平到GATE门信号再次启动。
- 1方式由GATE门信号每触发1次仅输出一个波形，故1方式也是输出单次波。

GATE作用



计数期间写入新初值



2. 定时器方式1：可编程单稳脉冲

情况一：

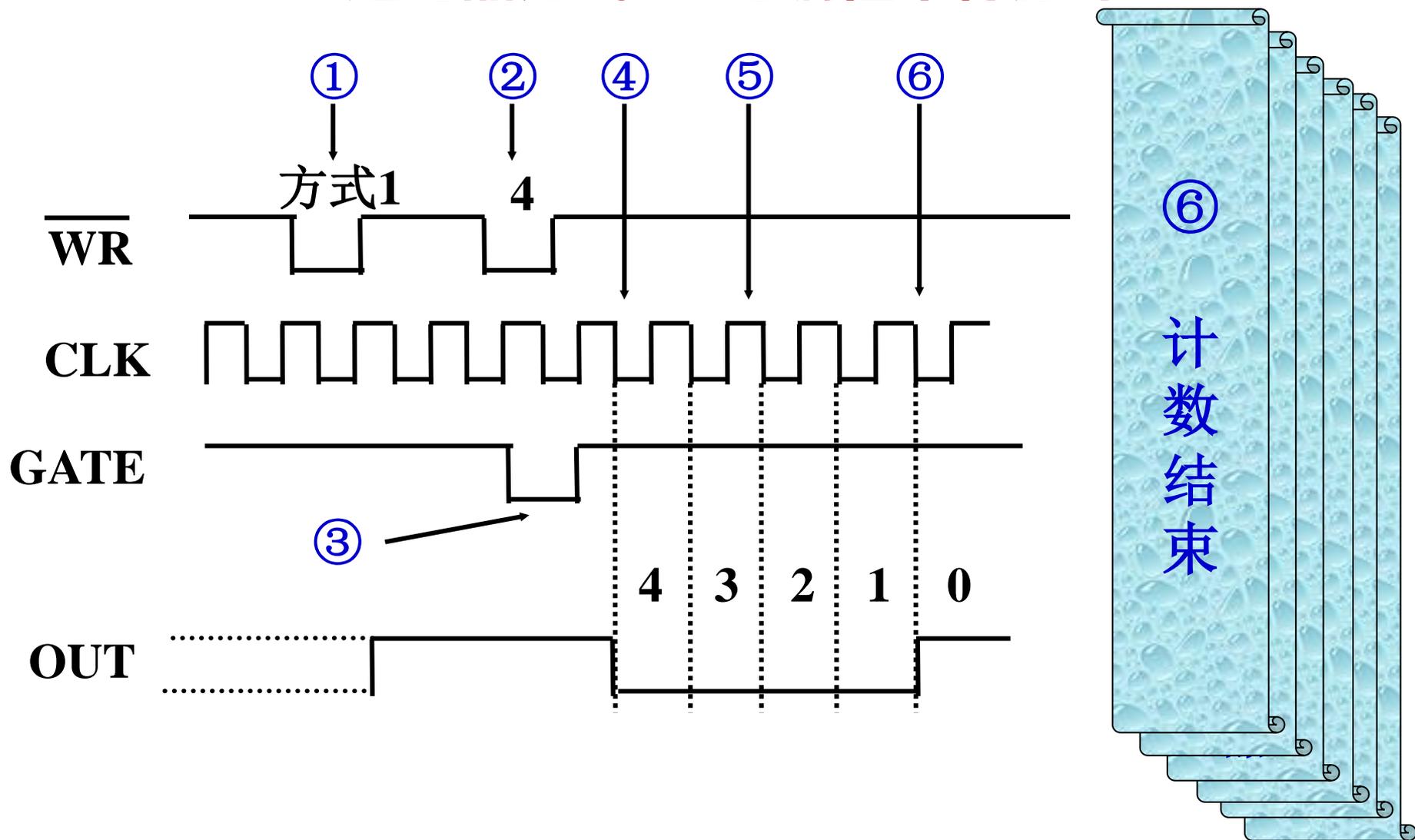
1. 写入计数初值后，计数器并不立即开始工作；
2. 门控信号GATE有效，才开始工作，使输出OUT变成低电平；
3. 直到计数器值减到零后，输出才变高电平。

情况二：

1. 在计数器工作期间，当GATE又出现一个上升沿时，计数器重新装入原计数初值并重新开始计数。
2. 如果工作期间对计数器写入新的计数初值，则要等到门控信号再次出现上升沿后，才按新写入的计数初值开始工作。

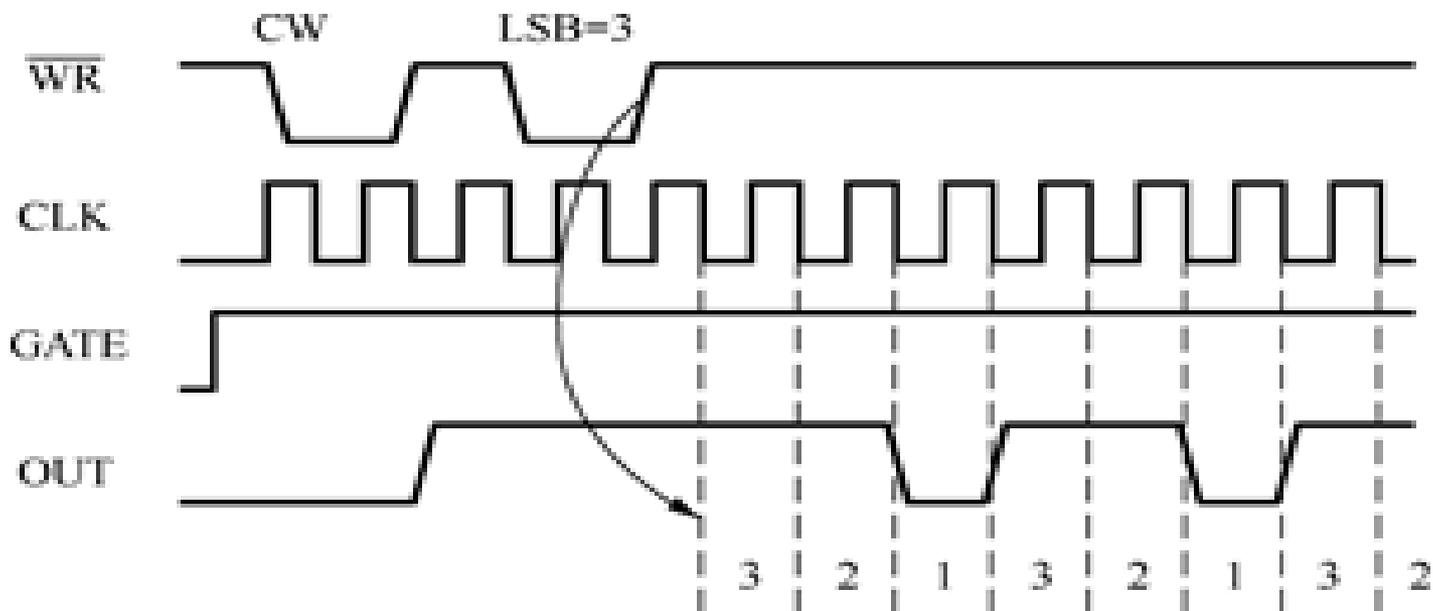
- 负脉冲宽度为**计数初值N和GATE出现的时机共同**决定。
- 不必重写同样的计数初值，只要GATE重新触发即可按上一次的计数初值重计数。
- 与方式0相比，方式1下的GATE不会暂停计数减1过程，只会重新启动计数过程【**即硬件启动计数方式**】，新启动的计数过程将按最新输入的计数初始值进行计数！
- 方式0和方式1的作用都是产生**单个负脉冲**，**负脉冲宽度可编程**，但一个是软件启动计数且脉冲宽度为 $N+1$ ，另一个是硬件启动计数且脉冲宽度为 N 。两种方式下计数值都有为0的时候。

2. 定时器方式1：可编程单稳脉冲



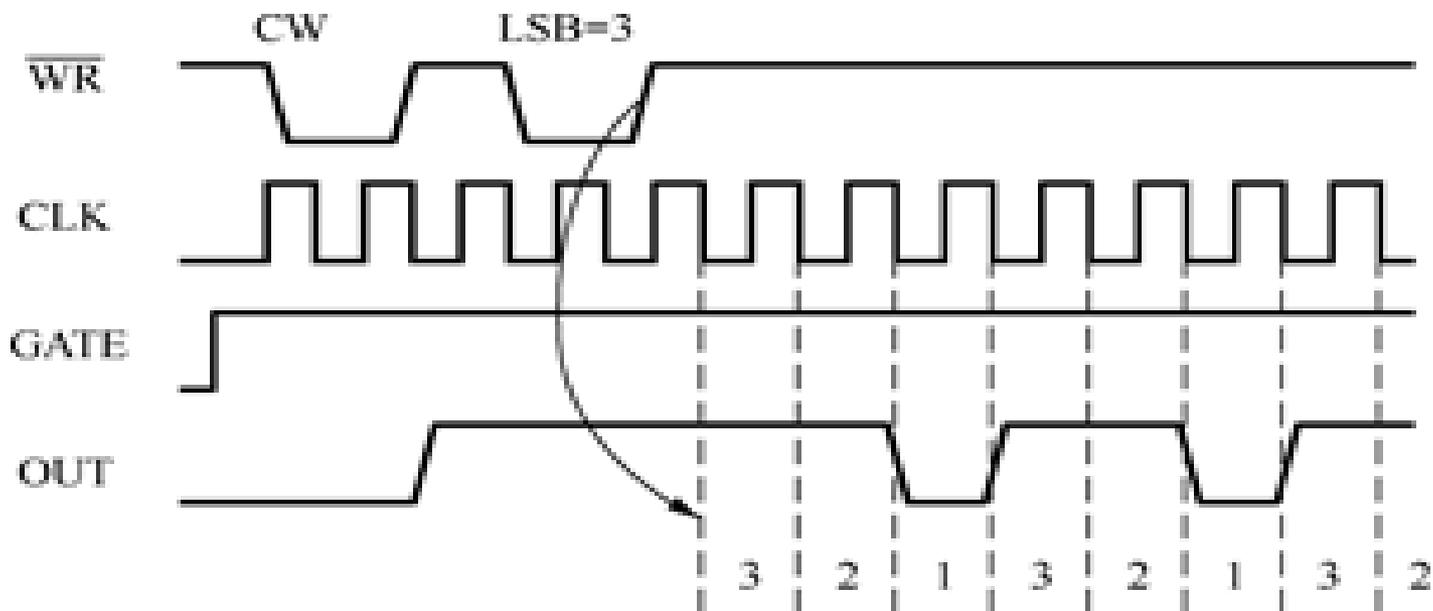
3. 定时器方式2：频率发生器（分频器）

该方式是N分频器方式或速率波发生器方式。分频系数就是计数初值，可产生连续的负脉冲，且负脉冲的宽度固定（周期输出波形）。



3. 定时器方式2：频率发生器（分频器）

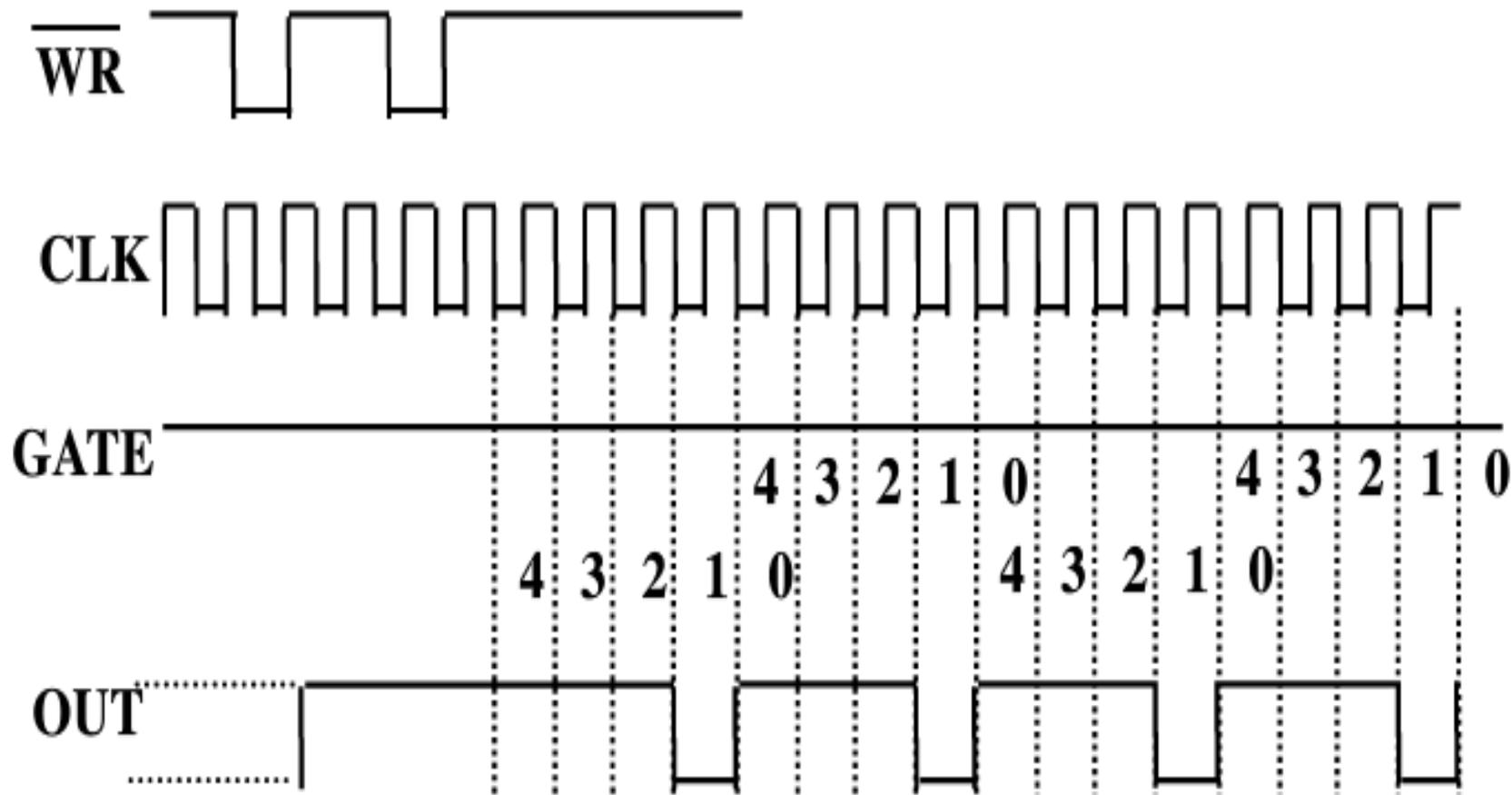
具体过程：当写入计数初值后，启动计数器开始减1计数，直到减到1时，OUT 输出一个宽度为时钟CLK周期的低电平，接着又变回高电平，且计数初值自动重装，开始新一轮计数，如此往复，不停地输出连续的负脉冲。



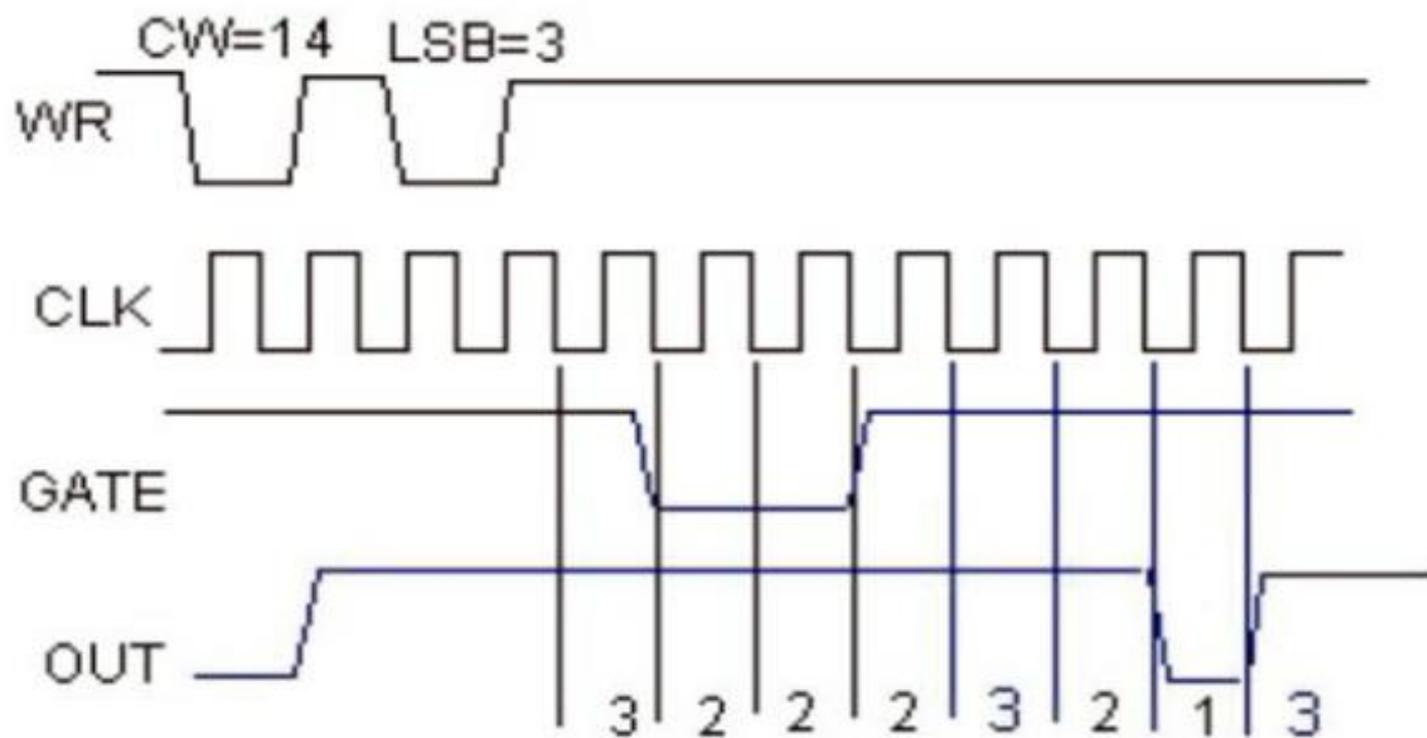
方式2过程特点:

控制特征	OUT和计数值的变化
1 控制字写入	当控制字写入后, $OUT=1$
2 计数值写入	$OUT=1$
3 启动方式	GATE上升沿可启动, 或者GATE=1时写入计数值也可以启动。
4 计数期间	$OUT=1$, 但计数值为1时, OUT输出一个时钟周期的负脉冲
5 计数为0时	$OUT=1$, 开始下一个周期的计数
6 计数期间写入新的计数值	影响随后的脉冲周期
7 GATE的作用	$GATE=0$ 时, $OUT=1$, 停止计数; GATE上升沿时, 按新计数值重启计数器, 无论原计数过程如何; $GATE=1$时不影响计数过程;
8 计数值有效期	计数值重复有效; 当值减到0时, 计数初值自动重载N(N可以在值变为0之前被更新), 此时0就被看做是N, 实现循环计数。 也可以认为计数值没有0!

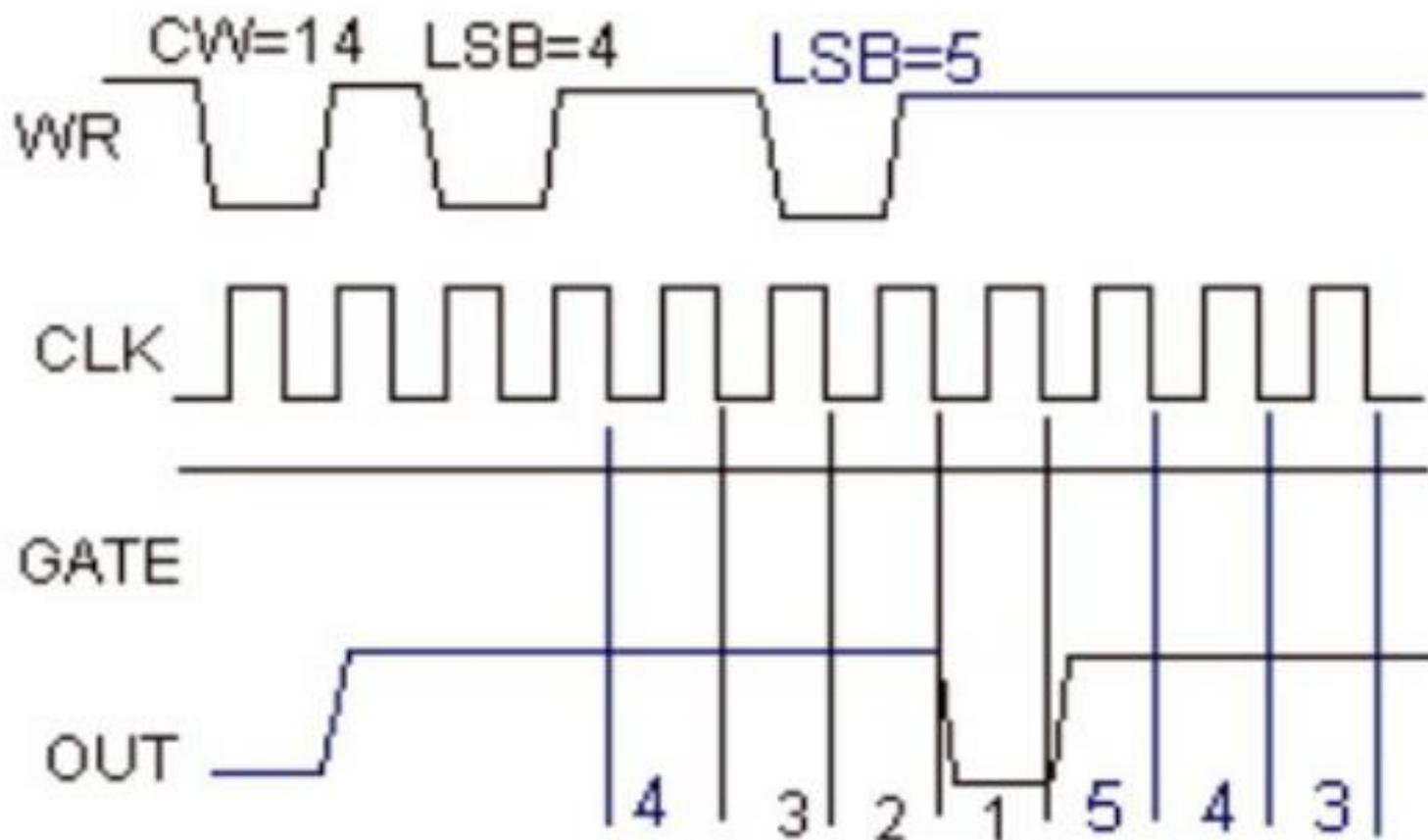
方式2 4



GATE作用



计数期间写入新初值



3. 定时器方式2：频率发生器（分频器）

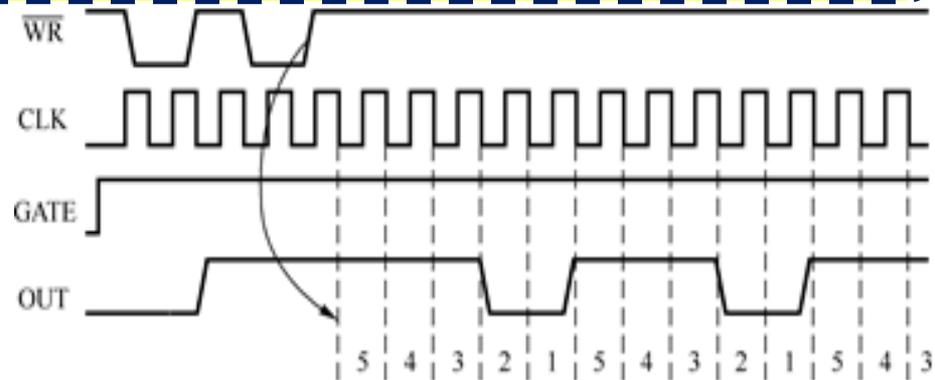
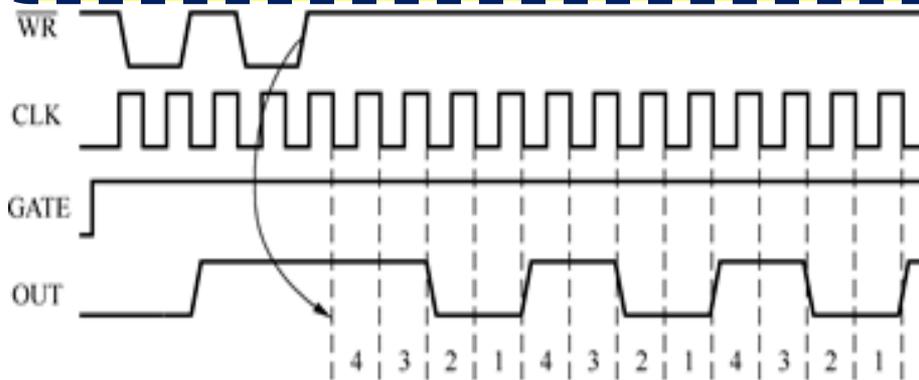
2方式特点：

- 不用重新设置计数值，能够连续工作，输出固定频率的负脉冲；
- 计数过程可以由GATE信号控制，GATE变低时立即暂停现行计数，变高后从计数初值开始重新计数；
- 如果在计数过程中，CPU重新写入计数初值，则对现在正在进行的计数过程无影响，而是从下一个操作周期开始按新的计数值改变输出脉冲的频率。

4. 定时器方式3：方波发生器

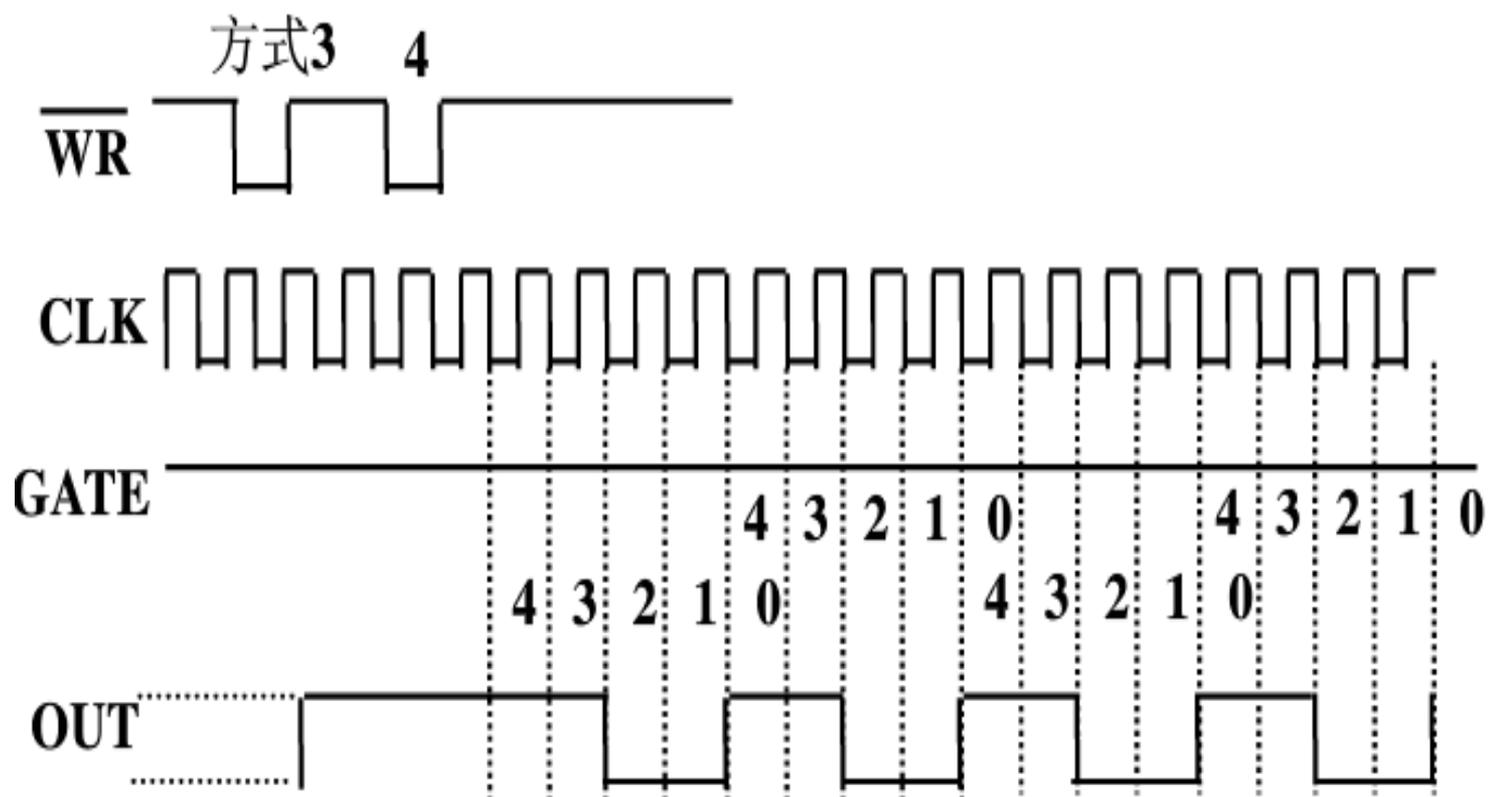
该方式产生占空比为1:1（N为偶数）或接近1:1（N为奇数）的连续方波。方波的周期=计数初值×时钟脉冲的周期。改变计数初值就可以改变输出方波的频率。

具体过程：当写入计数初值后，计数器自动开始计数，保持OUT为高，当计数到 $N/2$ 或者 $(N+1)/2$ 时，输出变低，直至计数到0。计数结束后，计数初值自动重装，开始新一轮计数，如此往复，不停的工作，因此，3方式连续输出方波。

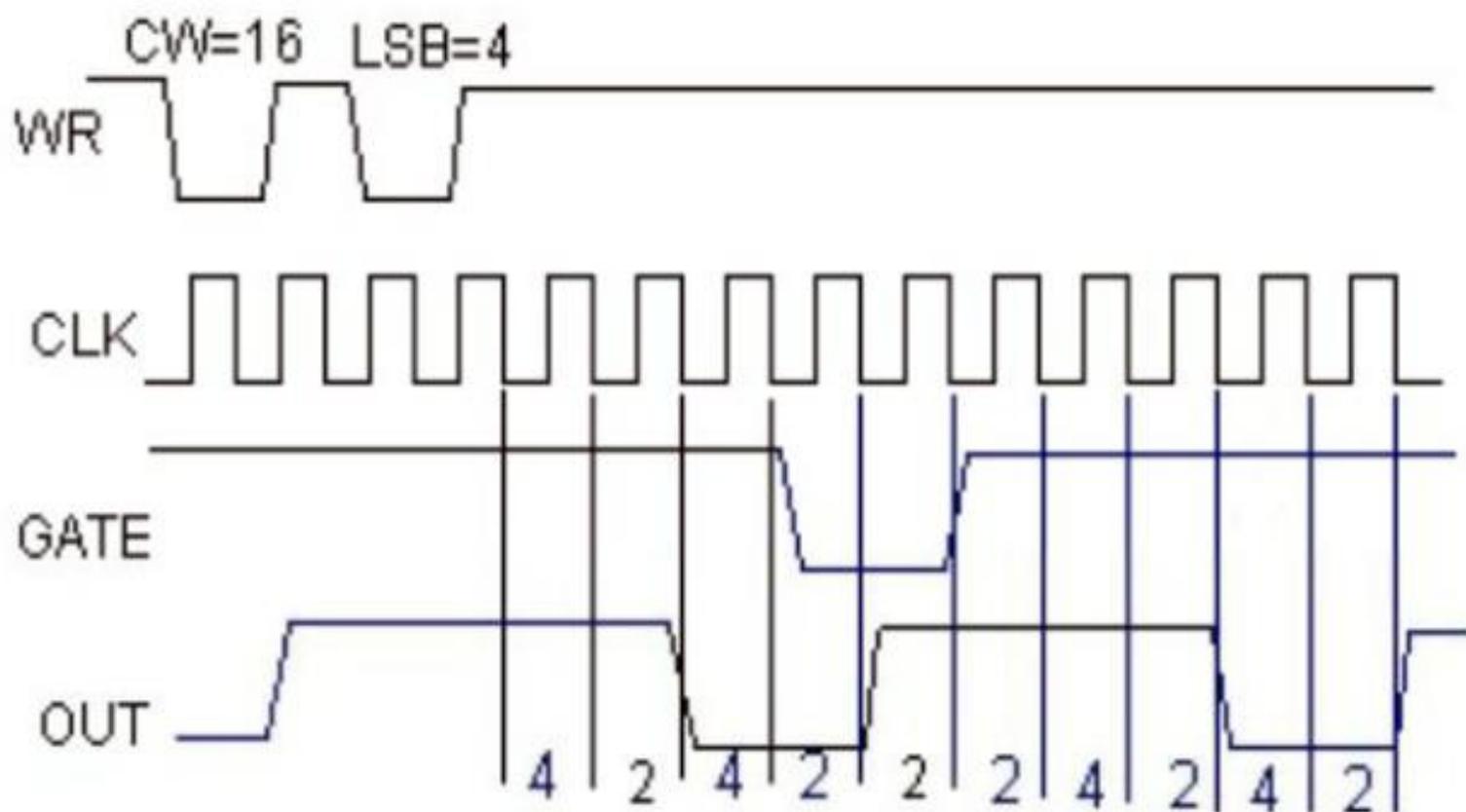


方式3过程特点:

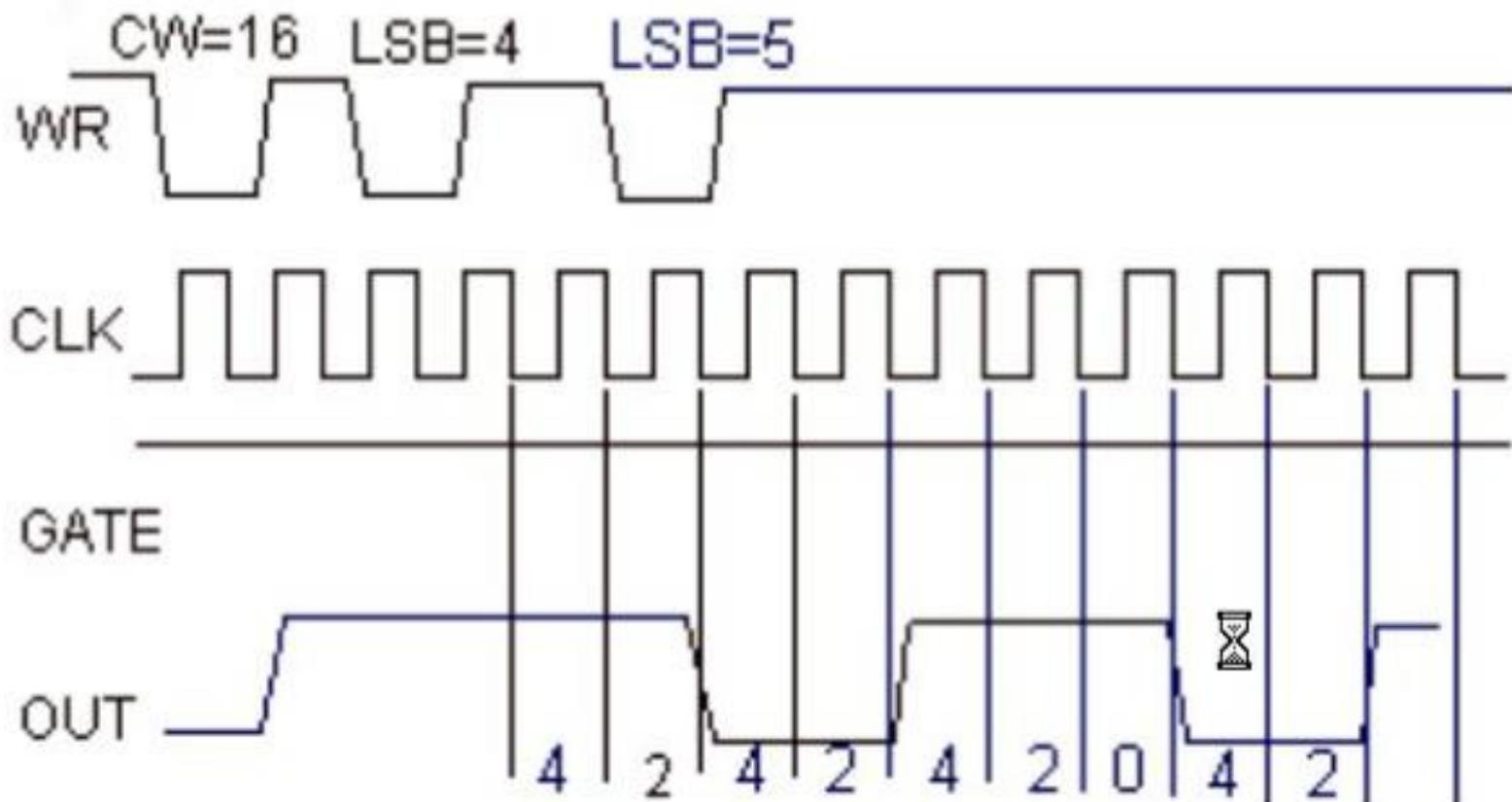
控制特征	OUT和计数值的变化
1 控制字写入	当控制字写入后, $OUT=1$
2 计数值写入	$OUT=1$
3 启动方式	GATE上升沿可启动, 或者GATE=1时写入计数值也可以启动。 在启动后的下一个时钟脉冲时 $OUT=1$
4 计数期间	若计数值N为偶数, 则在前 $N/2$ 计数期间, $OUT=1$;后 $N/2$ 计数期间 $OUT=0$; 如为奇数, 则前 $(N+1)/2$ 期间, $OUT=1$; 后 $(N-1)/2$ 期间, $OUT=0$
5 计数为0时	$OUT=1$, 完成一个周期; 然后计数值自动装入减1计数器, 开始下一周期
6 计数期间写入新的计数值	不影响当前输出周期; 当计数值为0或者GATE有上升沿后, 在当前输出周期结束以后时钟下降沿以新计数周期输出方波
7 GATE的作用	$GATE=0$ 时, $OUT=1$, 停止计数; GATE上升沿时, 下一周期按新计数值重启计数器; $GATE=1$ 时不影响计数过程;
8 计数值有效期	计数值重复有效; 计数值可以看做没有0



GATE作用



对本次没有影响，下次开始



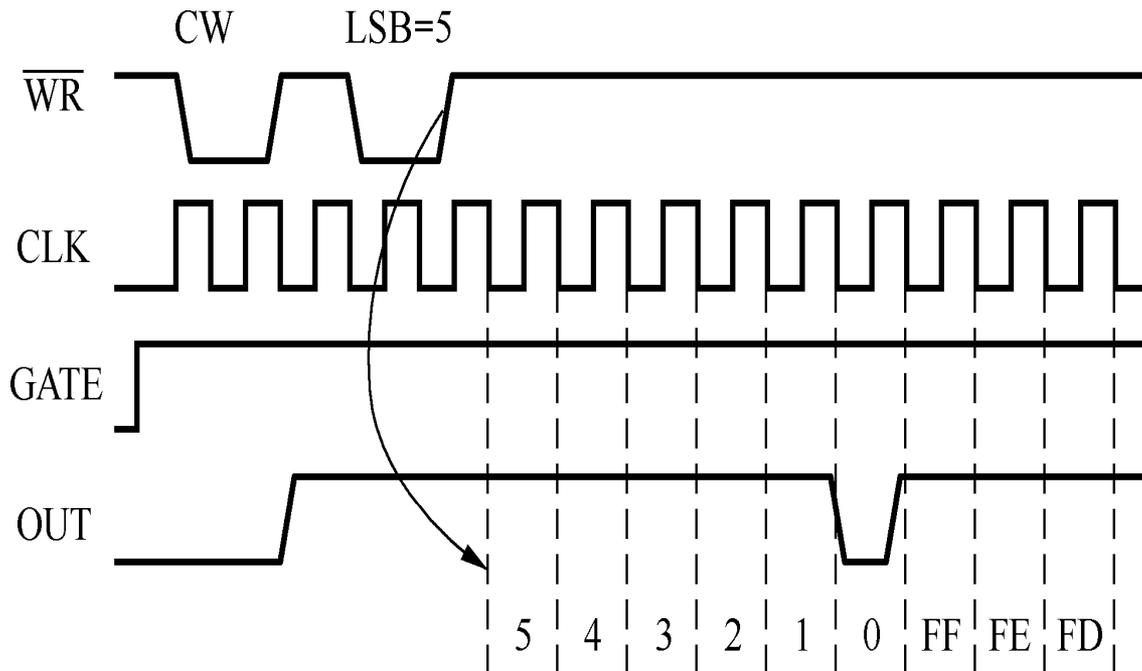
4. 定时器方式3：方波发生器

3方式特点：

- 若计数初值 N 为偶数，则输出波形是周期为 N 个CLK周期的对称方波；若计数初值 N 为奇数，则输出波形是周期为 N 个CLK周期的基本对称波形，其高电平持续时间比低电平持续时间多一个CLK周期；
- 如果在计数过程中，GATE信号变低，则暂停现行计数过程，直到GATE再次有效，将从计数初值开始重新计数；
- 如果要求改变输出方波的频率，则CPU可在任何时候重新写入新的计数初值，并从下一个计数操作周期开始改变输出方波的频率。

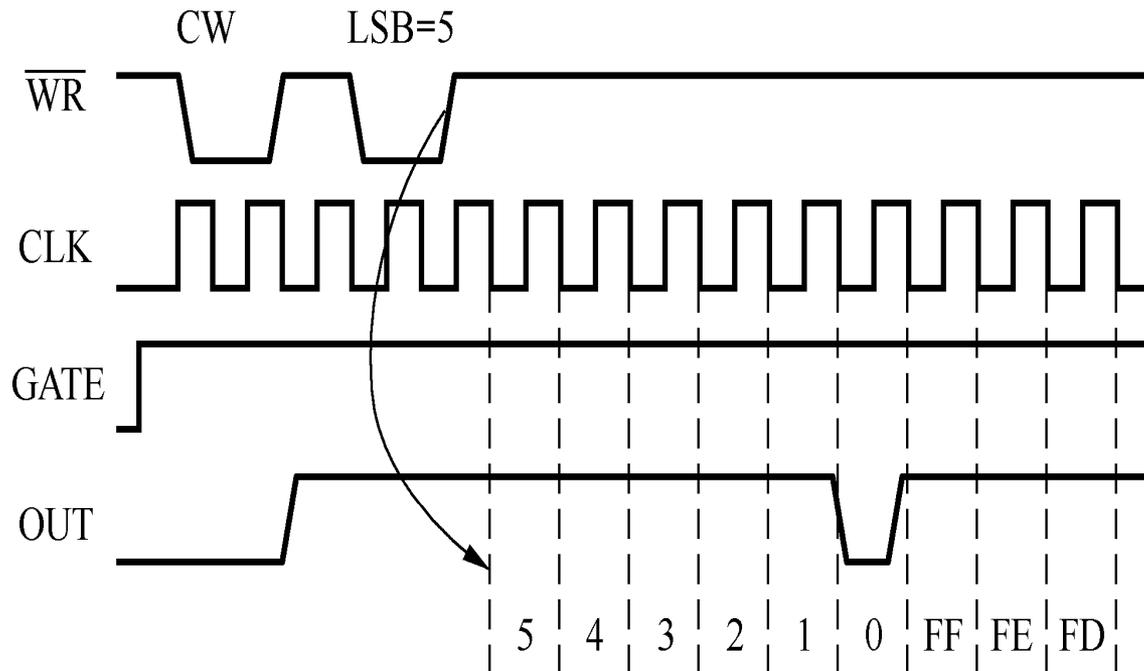
5. 定时器方式4：软件触发选通信号

该方式是一种由软件启动的闸门式计数方式，当写入计数初值后计数器即开始计数（即软件触发），当计数到0时，输出变低，经过一个CLK周期，输出又变高。门控信号GATE为高电平时，允许计数器工作。



5. 定时器方式4：软件触发选通信号

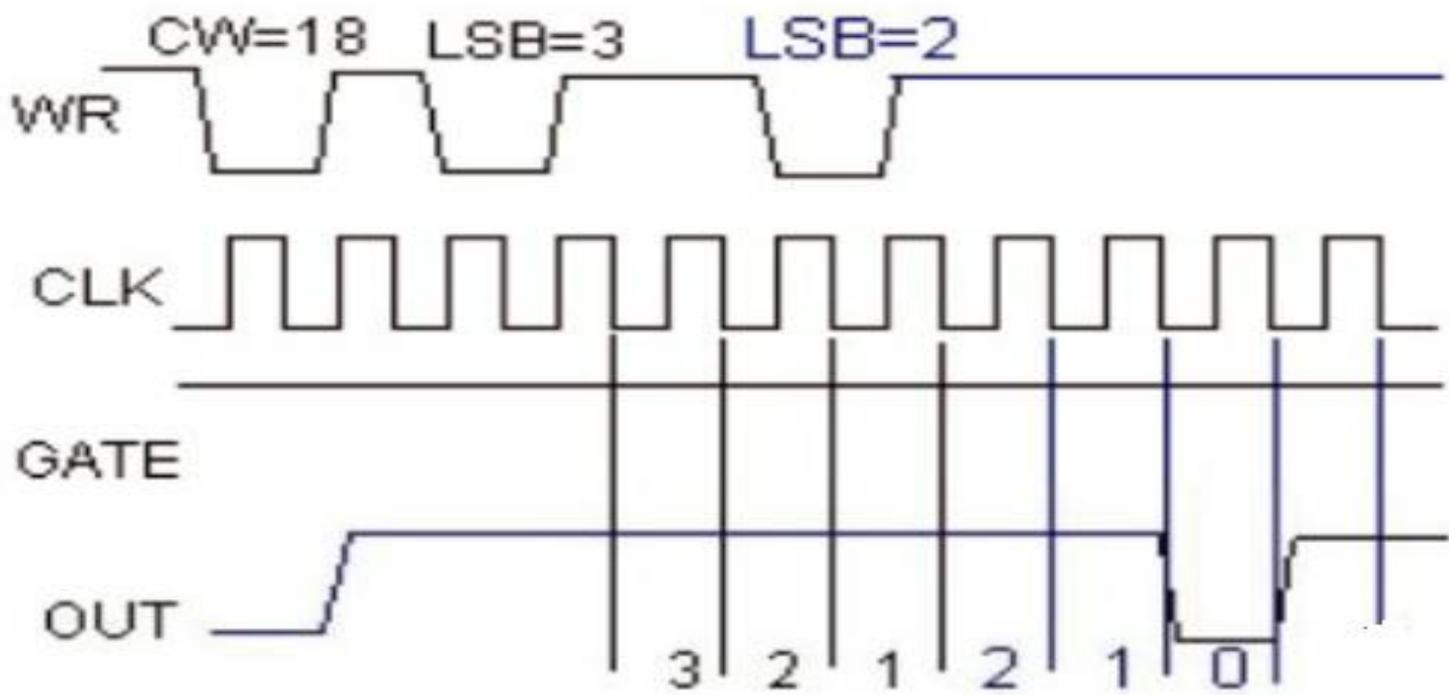
具体过程：当写入计数初值后，计数器自动开始计数，保持OUT为高，当计数值减到0时，在OUT端输出一个宽度等于时钟CLK脉冲周期的负脉冲，并停止工作。然后OUT信号变为高电平，并维持到再次写入新的计数值。



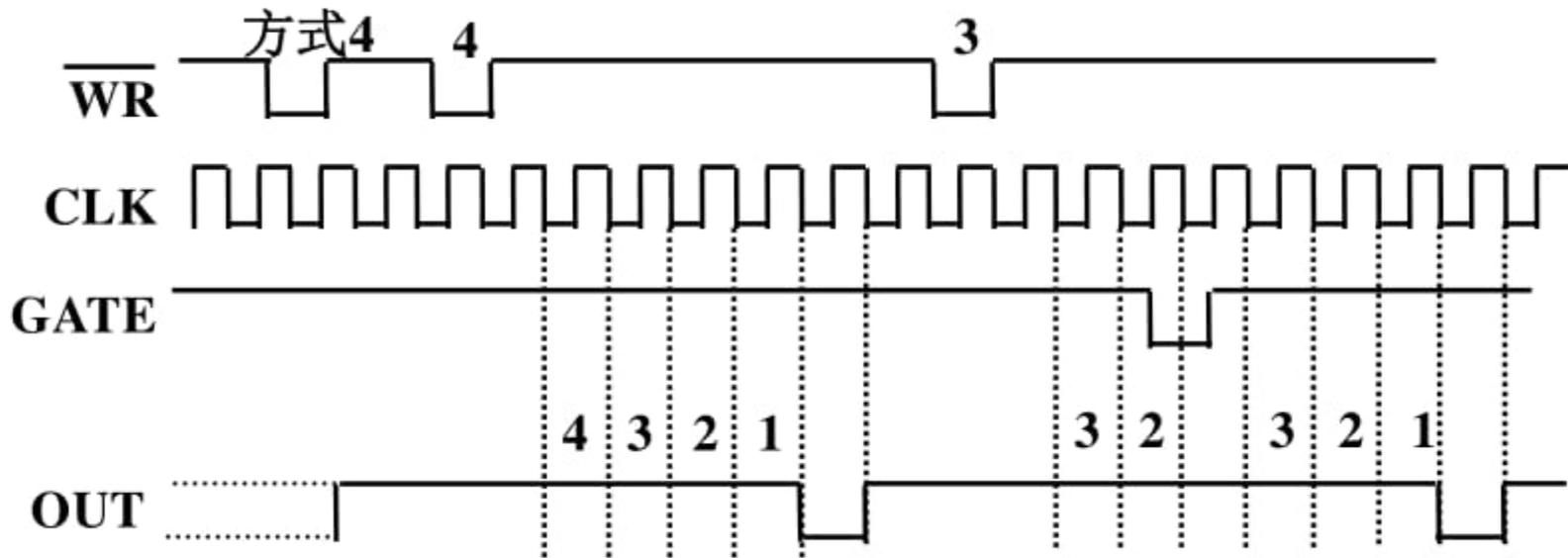
方式4过程特点:

控制特征	OUT和计数值的变化
1 控制字写入	当控制字写入后, $OUT=1$
2 计数值写入	$OUT=1$
3 启动方式	写入计数值后再过一个时钟周期开始根据计数初值减1计数
4 计数期间	$OUT=1$
5 计数为0时	输出一个负脉冲, 其宽度为1个时钟周期, 然后自动变为高电平并保持下去。
6 计数期间写入新的计数值	立即有效。在写入新计数值之后, 在下一个时钟周期时以新计数值计数减1。如果写入的计数值是2个字节, 在写入第一个字节时, 计数不受影响; 在写入第二个字节时, 下一个时钟周期以新计数值计数减1。
7 GATE的作用	$GATE=0$ 时, $OUT=1$, 停止计数; $GATE=1$ 时开始计数;
8 计数值有效期	计数值一次有效;

计数期间写入新初值



Gate 作用



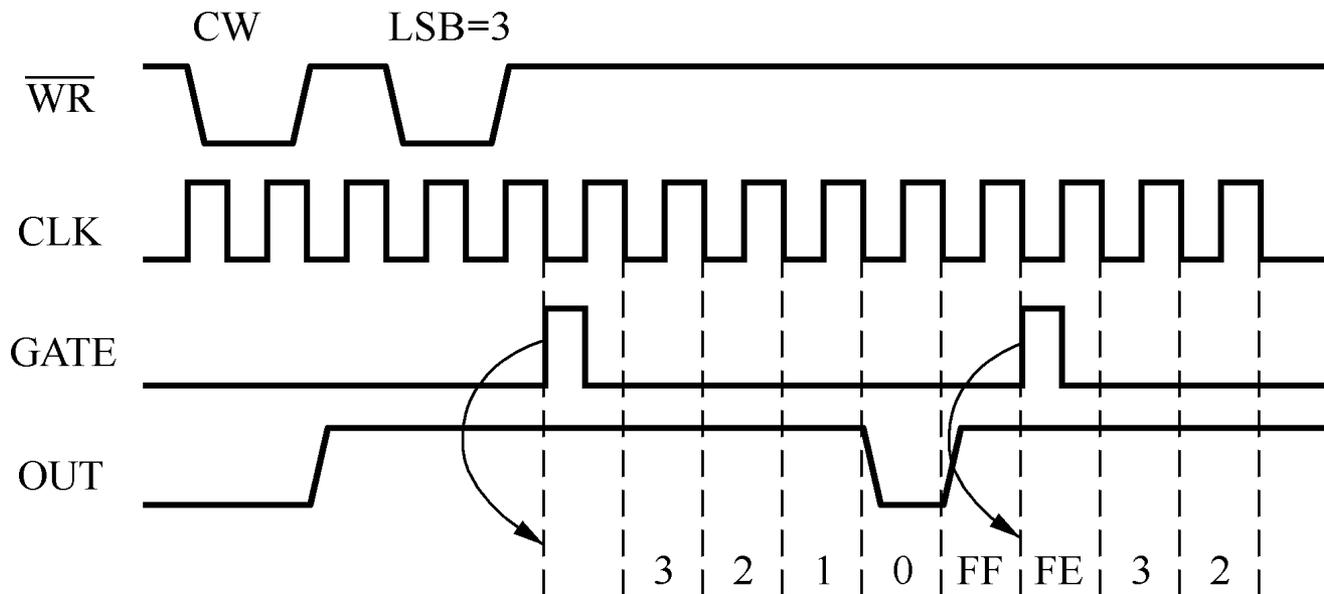
5. 定时器方式4：软件触发选通信号

4方式特点：

- 若设置计数初值为N，则在写入计数初值后的N个CLK脉冲，才输出一个负脉冲，负脉冲宽度为1个CLK周期；
- GATE为高时，允许计数，GATE为低时，禁止计数；
- 若在计数过程中改变计数值，则按新的计数值重新开始计数，即改变计数值是立即有效的。
- 4方式不具备计数初值重装能力，故输出单次波形。

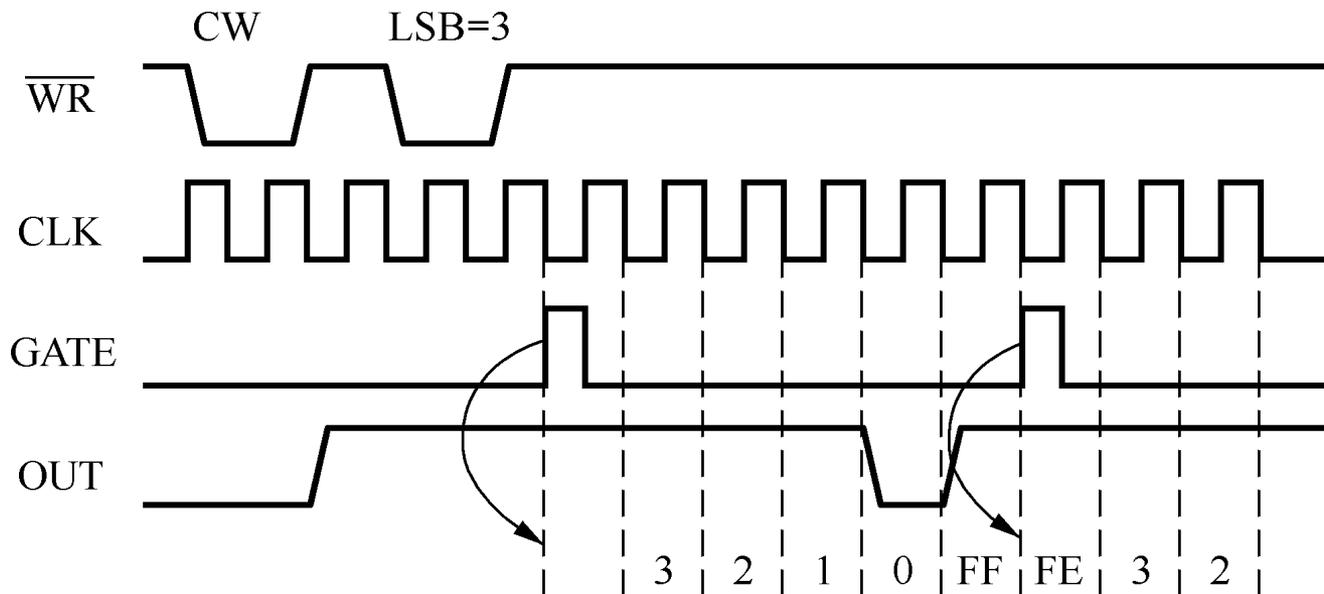
6. 定时器方式5：硬件触发选通信号

5方式在设置了控制字后，输出为高。在设置了计数初值之后，计数器并不立即开始计数，而是由门控信号GATE的上升沿触发启动（即硬件触发）。当计数到0时，经过一个CLK周期后，输出恢复为高，停止计数。



6. 定时器方式5：硬件触发选通信号

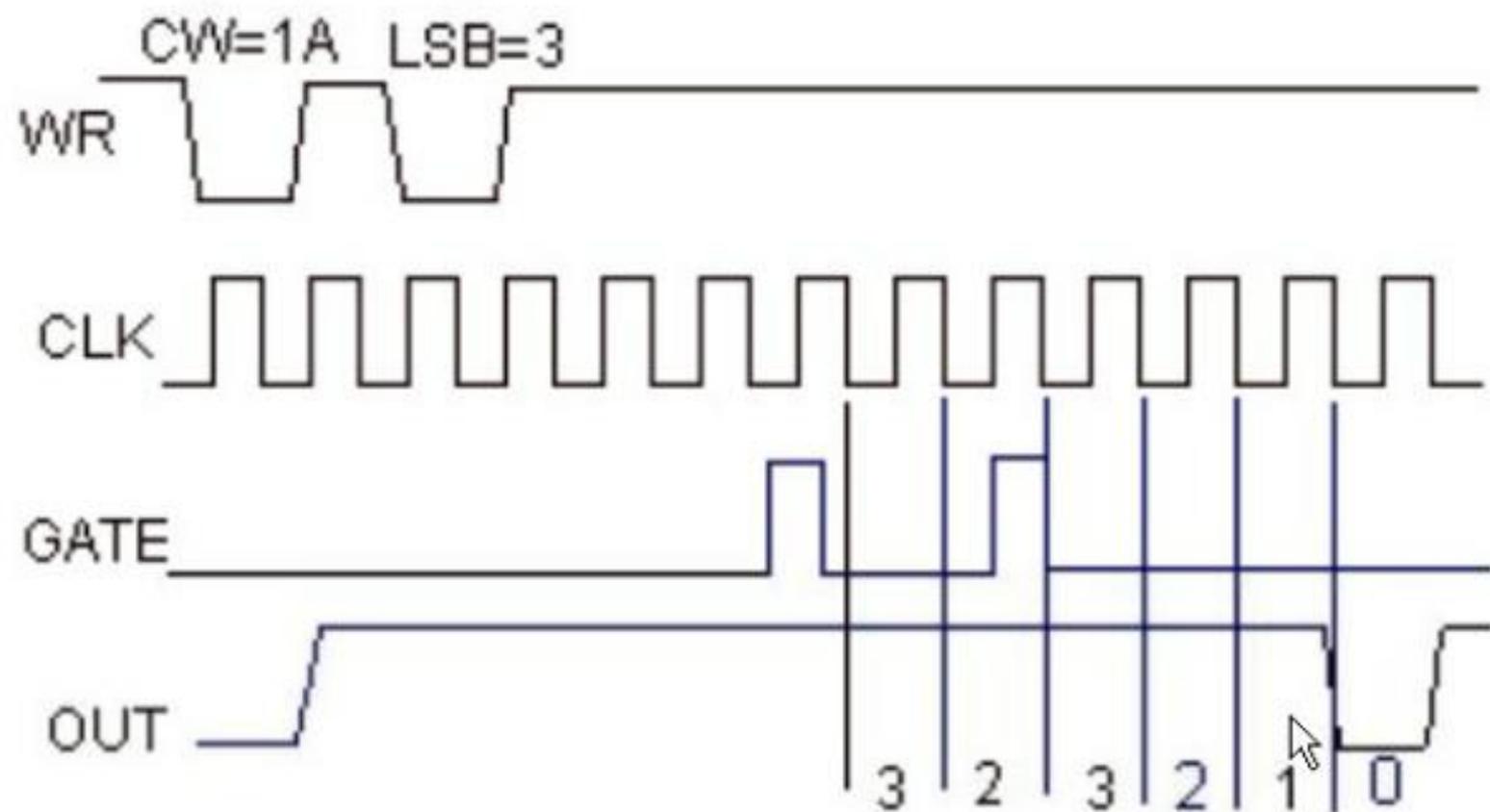
具体过程：当写入计数初值后，由GATE门信号启动计数，OUT输出高电平，当计数值减到0时，在OUT端输出一个宽度等于时钟CLK脉冲周期的负脉冲，并停止工作。然后OUT信号变为高电平，并维持到再次写入新的计数值。



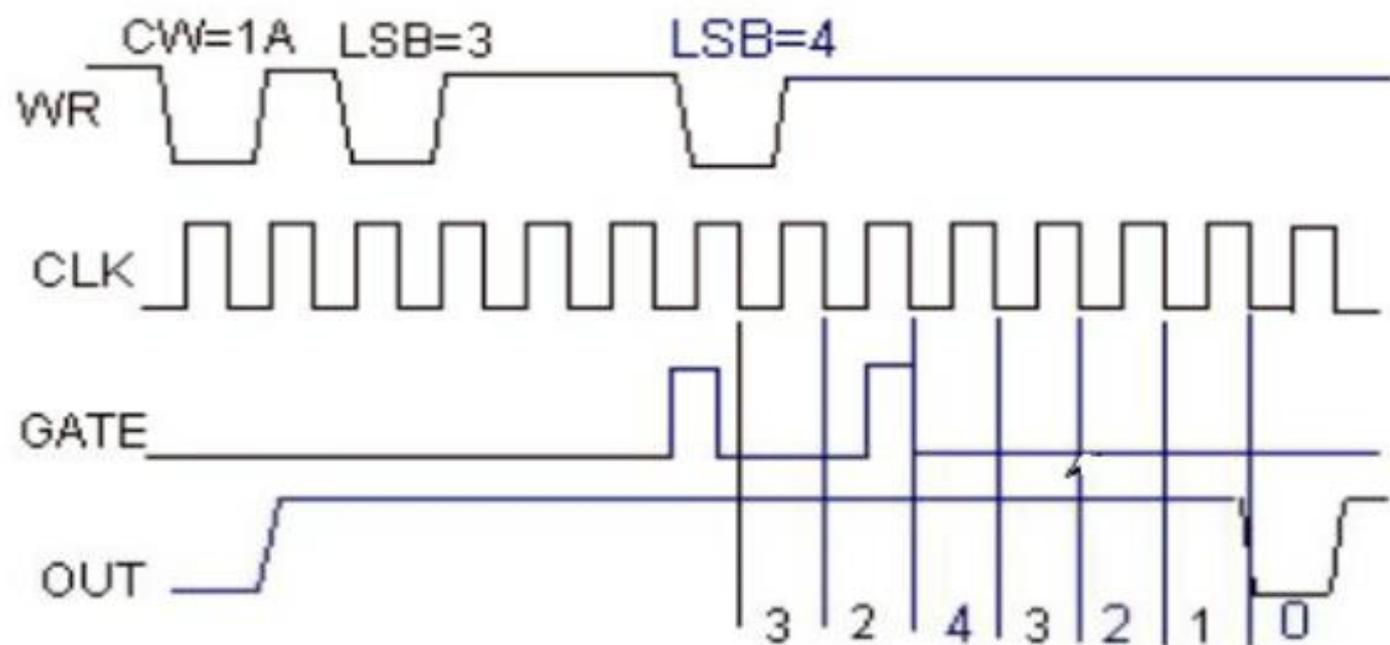
方式5过程特点:

控制特征	OUT和计数值的变化
1 控制字写入	当控制字写入后, $OUT=1$
2 计数值写入	$OUT=1$
3 启动方式	GATE上升沿。GATE上升沿之后的下一个时钟周期开始对计数初值减1计数。
4 计数期间	$OUT=1$
5 计数为0时	输出一个负脉冲, 其宽度为1个时钟周期, 然后OUT自动变为高电平并保持下去。
6 计数期间写入新的计数值	不影响本次计数。当GATE重新启动后, 将按新的计数值计数。
7 GATE的作用	GATE=0或1期间, 不影响计数器减1动作; 但若出现GATE上升沿, 则按新计数值重启计数器, 无论原计数过程如何。
8 计数值有效期	计数值多次有效; 当计数值减到0后, 将重新装入计数值, 但不计数, 直到出现GATE上升沿。

GATE作用



计数期间写入新初值



6. 定时器方式5：硬件触发选通信号

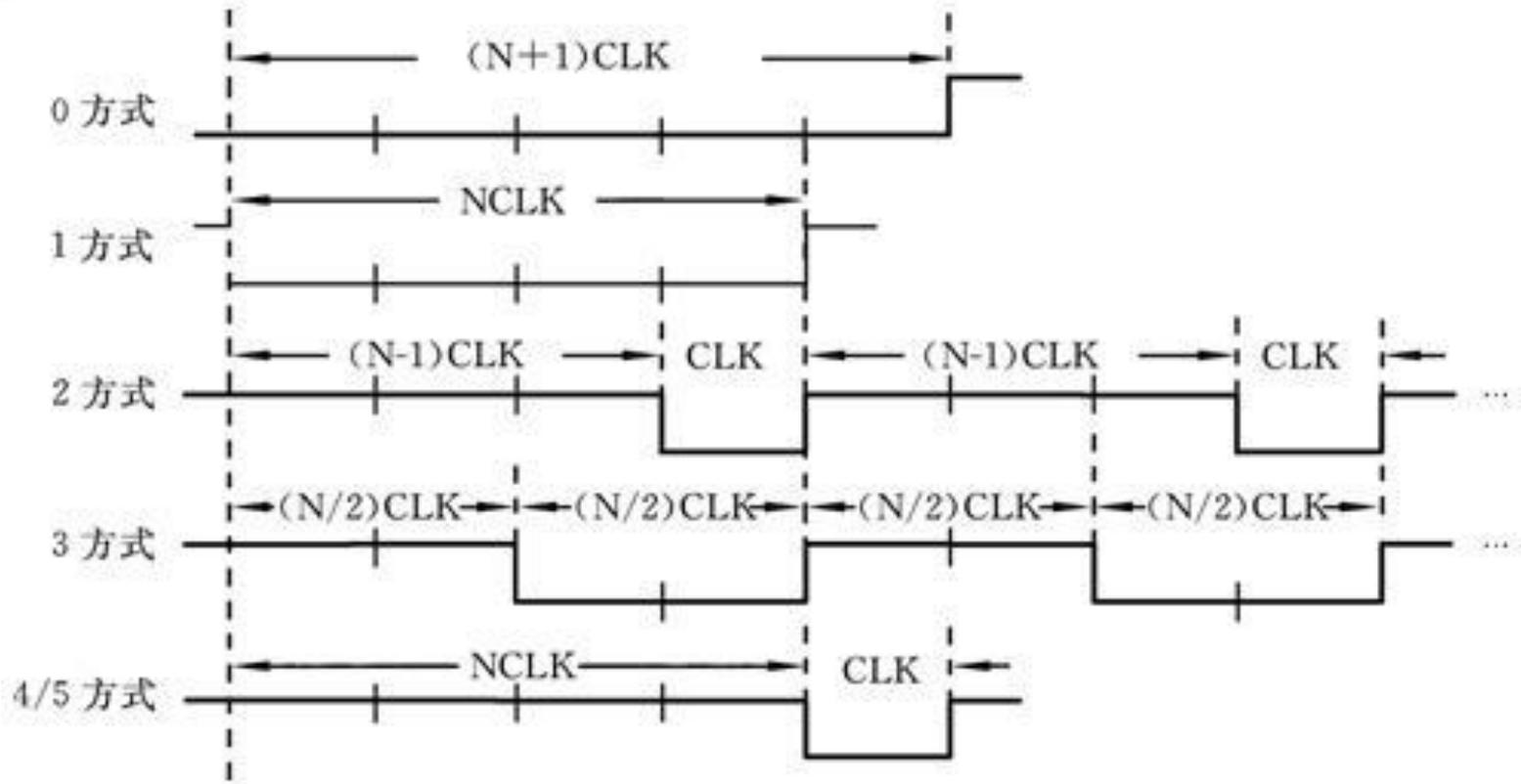
5方式特点：

- 若设置计数初值为N，则在门控信号GATE上升沿触发后，经过N个CLK脉冲，才输出一个负脉冲，负脉冲宽度为1个CLK周期。选通脉冲宽度等于时钟脉冲的周期。改变计数初值就可以改变选通脉冲产生的时间。
- 若在计数过程中再次出现门控GATE触发信号，则将使计数器从计数初值开始重新计数，但OUT输出的高电平不受影响；
- 若在计数过程中改变计数值，只要在计数到0之前不出现新的门控触发信号，则原计数过程不受影响，等计数到0并出现新的门控触发信号后，再按照新的计数值计数；若在写入了新的计数值后，在未计到0之前有门控触发信号出现，则立即按照新的计数值重新开始计数。
- 5方式不具备计数初值重装能力，也是输出单次波形。

小结：6种工作方式的比较

- 0方式和1方式，这两种方式的输出波形类似，均为单次波。区别在于计数启动方式不同，前者为软启动，后者是硬启动。
- 2方式和3方式，这两种方式共同的特点是具有自动再加载（装入）功能能力。二者的区别在于：2方式连续输出宽度为1个CLK的负脉冲（或占空比为N-1:1的方波），3方式连续输出占空比为1:1的方波或近似方波。
- 4方式和5方式。这两种方式的OUT输出波形相同，两者的区别是计数启动的触发信号不同，前者由写信号软启动计数，后者由GATE信号硬启动计数。

82C54的输出基本波形



CLK表示时钟频率， NCLK表示N个时钟频率

		方式0	方式1	方式2	方式3	方式4	方式5
OUT输出状态		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CLK变1, 重装初值继续计数	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CLK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
初值自动重装		无	无	计数到0重装	根据初值奇偶分别重装	无	无
计数过程中改变初值		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
GATE	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

10.4.4 82C54的计数初值计算及装入

1、计数初值的计算

计数初值分两种情况：若作**计数器**用时，则无须计算，将计数的次数直接装入；若作**定时器**用时，则计数初值需要经过换算才能得到。其换算方法如下。

① 产生时间间隔 τ 的时间常数 T_c

$$T_c = \frac{\text{要求定时的时间}}{\text{时钟脉冲周期}} = \frac{\tau}{1/\text{CLK}} = \tau \times \text{CLK}$$

② 产生频率为 f 的信号波形的时间常数 T_c

$$T_c = \frac{\text{时钟脉冲的频率}}{\text{要求的波形频率}} = \frac{\text{CLK}}{f}$$

2、计数初值的范围

计数初值寄存器和减法计数器是16位，故计数初值的范围

以二进制数表示为**0000H ~ FFFFH**；

以十进制数（BCD）表示为**0000 ~ 9999**。

3、计数初值的装入

- 由于82C54A内部的减法计数器和计数初值寄存器是16位，而82C54A外部引脚数据信号线只有8位，故**16位计数初值要分两次装入，并且按先装低8位、后装高8位的顺序写入计数器的数据口。**
- **只有2方式和3方式具有自动重装计数初值的功能，其他方式都需要用户通过程序人工重装计数初值。**

82C54的启动方式与中止方式

1、启动方式

(1) 软件启动

在GATE=1时（固定为1），计数初值一旦写入减1计数器，就开始计数。

(2) 硬件启动

计数初值已经写到减1计数器中，GATE信号由0→1（外部可控）的上升沿开始计数

2、计数过程的中止

对于周期定时，通过GATE=0来中止计数；

对于脉冲定时，要等到单次定时结束为止。

10.4.5 82C54A的初始化

1、向命令寄存器写入方式命令字，包括选择计数器（3个计数器之一）、确定工作方式（6种方式之一）、指定计数器计数初值的长度、装入顺序、计数值的码制（BCD或二进制码）。

2、向已选定的计数器按方式命令的要求写入计数初值

$RW_1RW_0=01$ ：只写入低8位，高位自动置0

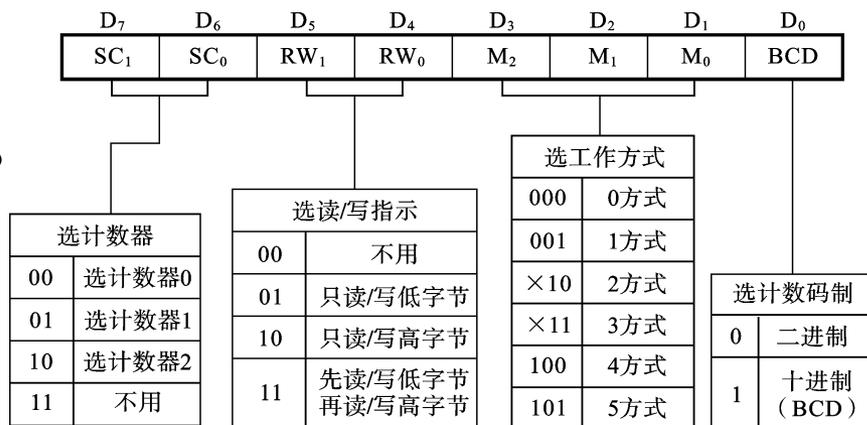
$RW_1RW_0=10$ ：只写入高8位，低位自动置0

$RW_1RW_0=11$ ：先写低8位后写高8位

例如，选择2号计数器，工作在3方式，计数初值为533H（2个字节），采用二进制计数。

汇编语言初始化程序段如下。

```
MOV DX,307H
MOV AL,10110110B
OUT DX,AL
MOV DX,306H
MOV AX,533H
OUT DX,AL
MOV AL,AH
OUT DX,AL
```



;2号计数器数据口

;计数初值

;先送低字节到2号计数器

;取高字节送AL

;后送高字节到2号计数器

10.5 定时/计数器的应用

10.5.1 82C54A微机系统中的应用设置

OUT₀用作系统时钟中断

OUT₁用要用做动态存储器定时刷新

OUT₂用用作扬声器发声控制

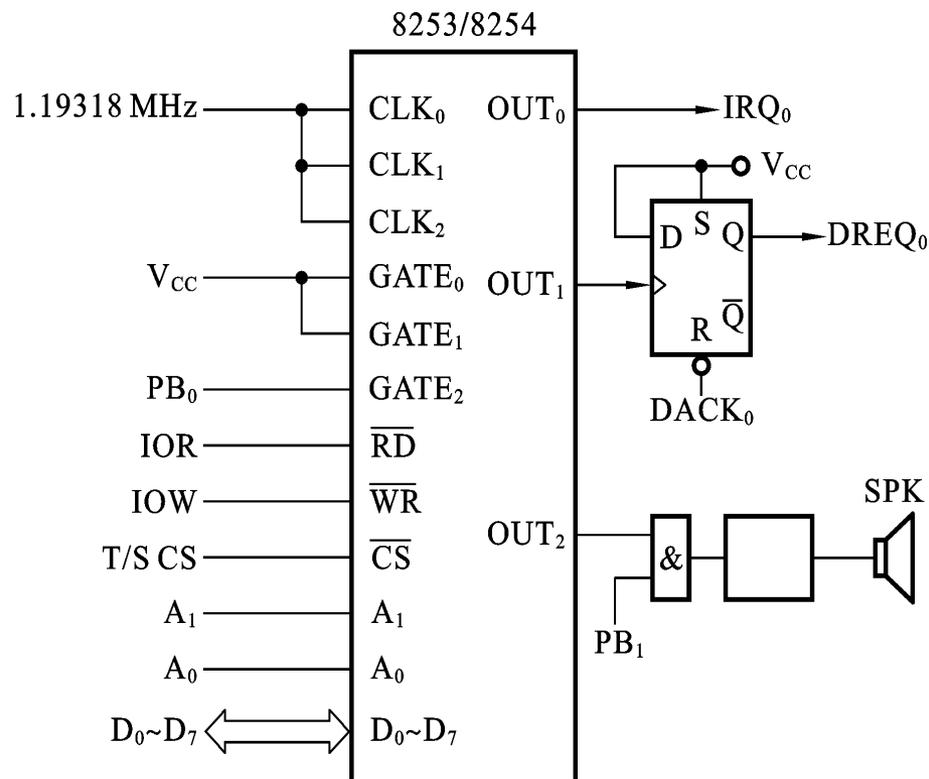
0号计数器=40H (端口地址)

1号计数器=41H (端口地址)

2号计数器=42H (端口地址)

方式命令寄存器=43H (端口地址)

以上用途设置及地址的分配用户不能更改。系统提供的时钟脉冲频率为1.19318MHz。



82C54A在微机系统中：通道0用于日时钟的中断请求，通道1用于DRAM的刷新请求，通道2用于扬声器的发声设置（扬声器的启动/关闭信号GATE由并口电路控制）

系统的应用配置如下：

82C54A在系统中的应用设置

计数通道	读/写方式	工作方式	计数码制	计数初值	CLK/MHz	GATE	T _{out}	F _{out}	OUT	用途
0	高/低字节	3	二进制	0000H	1.19318	+5V	55ms	18.2Hz	IRQ ₀	日时钟中断请求
1	只写低字节	2	二进制	12H	1.19318	+5V	15 s	66.3kHz	DREQ ₀	DRAM刷新请求
2	高/低字节	3	二进制	533H	1.19318	PB ₀ 控制	1.1155ms	896Hz	SPK	扬声器发声

例1：计数通道0每55ms产生一次中断请求

分析：T0作定时中断，选用方式3，控制字为36H
(00110110B)，初值为0000H（即65536）。

初值： $T=1.19318\text{MHz} / (1/55\text{ms}) \approx 65536$

```
MOV AL,00110110B      ;初始化方式命令
OUT 43H,AL
MOV AX,00H             ;初值为0000H（最大值）
OUT 40H,AL             ;先写低字节
MOV AL,AH
OUT 40H,AL             ;再写高字节
```

例2：计数通道1每15μs请求一次DMA传输

分析：T1向DMA控制器定时发出动态存储器刷新请求，选用方式2，控制字为54H（01010100B），计数初值为18（12H）。

初值： $T=1.19318\text{MHz} / (1/15\mu\text{s}) \approx 12\text{H}$

```
MOV AL,01010100B ;初始化方式命令  
OUT 43H,AL  
MOV AL,12H ;初值为12H  
OUT 41H,AL ;只写低字节
```

例3：利用计数通道2产生896HZ的方波使扬声器发声

分析：计数器2控制喇叭发声，选用方式3，控制字为B6H
(10110110B)，OUT1输出方波的频率为1.19MHz/计数初值。
初值： $T=1.19318\text{MHz}/896\text{Hz}=1331$ (533H)

```
MOV AL,10110110B    ;初始化方式命令
OUT 43H,AL
MOV AX,533H          ;初值为533H
OUT 42H,AL           ;先写低字节
MOV AL,AH
OUT 42H,AL           ;再写高字节
```

10.5.2 微机系统配置的82C54A初始化程序段

① 计数器0：用于定时中断（约55ms申请1次中断）

```
MOV AL,00110110B      ;初始化方式命令
OUT 43H,AL
MOV AX,00H             ;初值为0000H（最大值）
OUT 40H,AL             ;先写低字节
MOV AL,AH
OUT 40H,AL             ;再写高字节
```

② 计数器1：用于DRAM定时刷新（每隔15 μ s请求1次DMA传输）

```
MOV AL,01010100B      ;初始化方式命令
OUT 43H,AL
MOV AL,12H             ;初值为12H
OUT 41H,AL             ;只写低字节
```

③ 计数器2：用于产生约900Hz的方波使扬声器发声

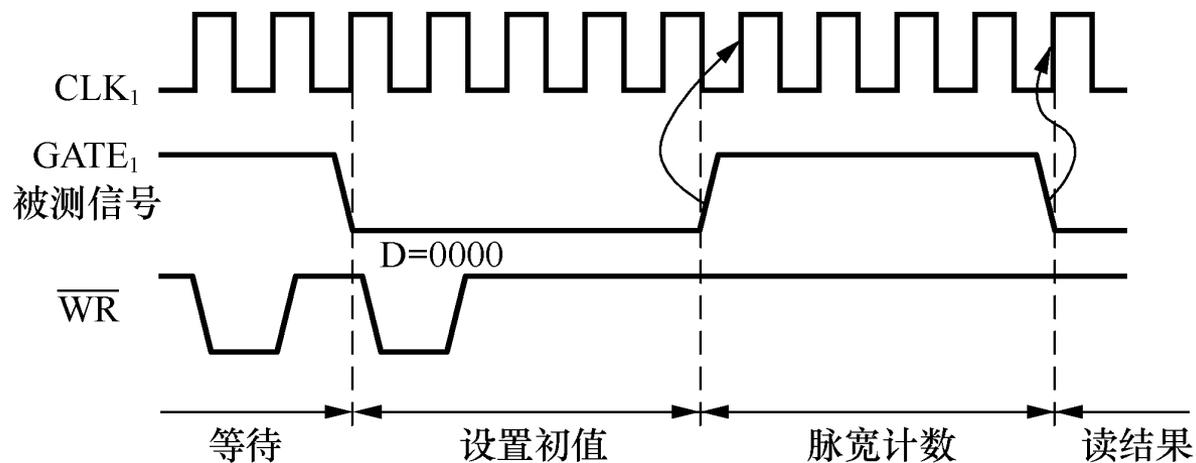
```
MOV AL,10110110B      ;初始化方式命令
OUT 43H,AL
MOV AX,533H            ;初值为533H
OUT 42H,AL             ;先写低字节
MOV AL,AH
OUT 42H,AL             ;再写高字节
```

10.5.2 定时/计数器82C54A的应用举例

例1：某应用系统测量脉冲的宽度。系统提供的输入时钟CLK=1MHz，采用二进制计数。

分析：82C54A在软启动时，门控信号GATE的作用是允许或禁止计数。例如，0方式就有这种特点，因此可以**利用GATE门进行脉冲宽度测量，把被测的脉冲作为GATE门信号来控制计数器的计数过程**，即控制计数的启/停。

在被测脉冲信号为低电平时，装计数初值，当被测脉冲信号变高电平时，开始计数，直至被测脉冲信号再次变低电平，停止计数，并锁存。然后读出计数器1的当前值n，最后得到**被测脉冲宽度是 (计数初值65536-n) μ s。**



设计：选择计数器通道1，工作方式为0方式。

从波形图中可以看出，**当**GATE1门信号也就是**被测信号为低电平时**，即使写入计数初值也**不开始计数**，只有**当**GATE1门信号也就是**被测信号变为高电平时才开始计数操作**，并且**直到**GATE1门信号也就是**被测信号再次变为低电平时才结束**。于是**把这一段计数过程中所累积的时钟计数脉冲的个数乘以脉冲的周期就得到被测信号的宽度**。

小结：地址范围为304H~307H；T1通道工作于方式0，计数初值为0000H、二进制，方式字：01110000B=70H

```
MOV DX,307H           ;82C54A的命令口
MOV AL,70H            ;方式命令
OUT DX,AL
MOV DX,305H           ;1号计数器的数据口
MOV AX,0000H          ;定时常数
OUT DX,AL             ;定时常数低字节
MOV AL,AH             ;定时常数高字节
OUT DX,AL
MOV DX,307H           ;82C54A的命令口
MOV AL,40H            ;1号计数器的锁存命令 (0100XXXXH)
OUT DX,AL
MOV DX,305H
IN AL,DX              ;从1号计数器读当前计数值，保存
MOV BL,AL
IN AL,DX
MOV BH,AL
MOV AX,0000H
SUB AX,BX             ;65536-BX,可得被测脉冲的宽度
```

[用82C54A定时]

例2：某应用系统要求每隔5ms发出一个扫描负脉冲，系统提供的时钟CLK为20KHZ，采用十进制计数。

分析： (1) 选择工作方式

为了产生每隔5ms一次的连续的定时脉冲，**选择82C54A的2方式**是合适的。为此，**利用82C54A的计数器2**，将它的OUT2作为定时脉冲输出。

(2) 计算定时常数

因为是做定时器，故需要进行定时常数计算。将系统提供的CLK作为通道2的输入时钟CLK2，其周期 $T=1/20\text{kHz}=0.05\text{ms}$ ，按照要求定时时间为5ms，**可得定时常数**为 $TC=5\text{ms}/0.05\text{ms}=100$

初始化程序：

```
MOV DX,307H           ;82C54A的命令口
MOV AL,95H            ;方式命令 (10010101B)
OUT DX,AL
MOV DX,306H           ;计数器2的数据口
MOV AL,100H           ;定时常数
OUT DX,AL
```

[用82C54A分频]

例3：某应用系统要求产生频率为1KHz的方波，系统输入时钟频率为1.19318MHz，采用二进制计数

分析： (1) 选择工作方式

为了产生**方波**，选择**82C54A的3方式**是合适的。为此，利用82C54A的计数器0，将它的OUT0作为方波输出。

(2) 计算计数常数

将系统提供的CLK作为计数器0的输入时钟CLK0，按照要求输出OUT0=1000Hz的方波，可得**定时常数**为：

$$TC=CLK0/OUT0=1.19318MHz/1000Hz=1193=4A9H$$

初始化程序：

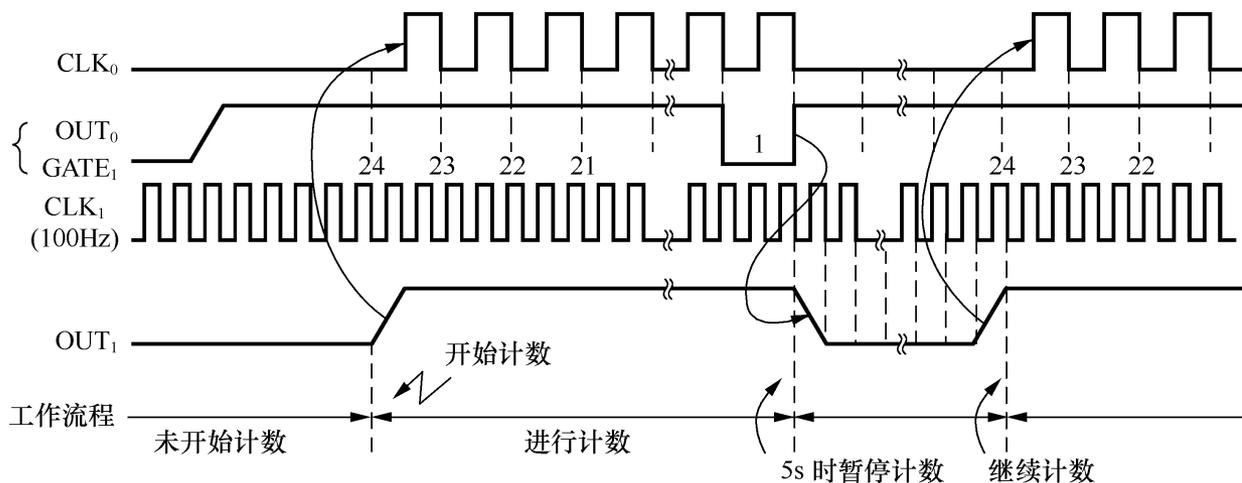
MOV DX,307H	;82C54A的命令口
MOV AL,36H	;方式命令 (00110110B)
OUT DX,AL	
MOV DX,304H	;计数器0的数据口
MOV AX,4A9H	;定时常数
OUT DX,AL	;装入定时常数低字节
MOV AL,AH	
OUT DX,AL	;装入定时常数高字节

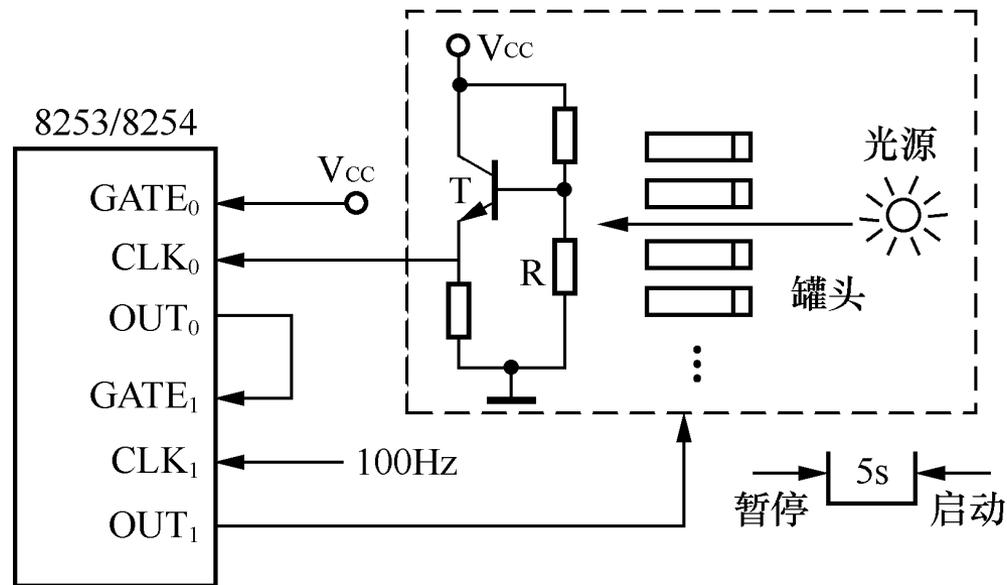
例4：某罐头包装流水线能装24罐，要求每通过24罐，流水线暂停5s，等待封箱打包完毕，重启流水线，继续装箱。按Esc键则停止生产。

[用82C54A进行计数与定时]

分析：为了实现上述要求，有**两个工作要做**：**一是对24罐计数**；**二是对5s停顿定时**。并且，两者之间又是相互关联的。

为此，**选用82C54A的计数器0作计数器、计数器1作定时器**，并且**把计数器0的计数已到（24）的输出信号OUT0，连到计数器1的GATE1线上**，作为外部硬件启动信号触发**计数器1的5s定时**，去控制流水线的暂停与重启。





硬件设计：检测原理是罐头从光源和光敏电阻（R）之间通过时，在晶体管（T）发射极上会产生罐头的脉冲信号，此脉冲信号作为计数脉冲，接到计数器0的CLK0，对罐头进行计数。

计数器0作为计数器，工作在2方式，计数初值24（18H），方式字00010101B（15H），端口地址304H；

计数器1作为定时器，工作在1方式，计数初值 $5 \div (1/100) = 500$ （1F4H），方式字01110011B（73H），端口地址305H；

软件设计：包装流水线汇编语言程序段（只写出代码段）

code segment

```
START: MOV DX,307H           ;计数器0初始化
        MOV AL,15H           ;命令字
        OUT DX,AL
        MOV DX,304H         ;写计数器0计数初值
        MOV AL,24H          ;计数器0计数初值
        OUT DX,AL
        MOV DX,307H         ;计数器1初始化
        MOV AL,73H          ;命令字
        OUT DX,AL
        MOV AX, 500H        ;计数器1定时常数
        MOV DX,305H
        OUT DX,AL           ;先写低字节
        MOV AL,AH
        OUT DX,AL           ;再写高字节
CHECK: MOV AH,0BH          ;有键按下?
        INT 21H
        CMP AL,00H
        JE CHECK            ;无键, 则等待
        MOV AH,08H         ;是ESC?
        INT 21H
        CMP AL,1BH
        JEN CHECK
        MOV AX,4C00H        ;是ESC, 则返回DOS
        INT 21H
```

code ends

end start