



《微机原理与接口》

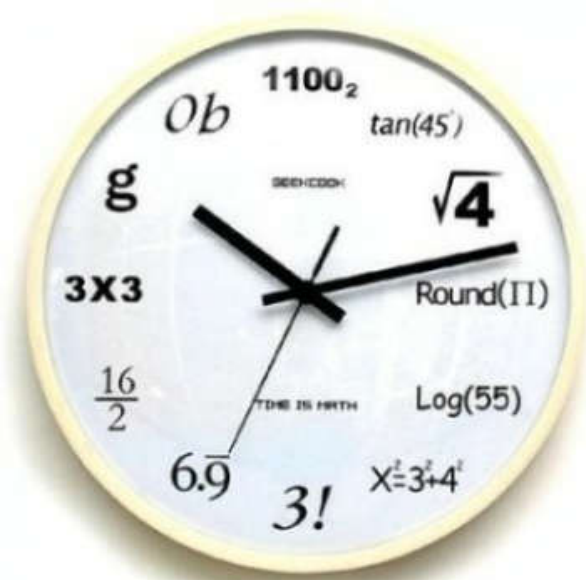
第5章 定时器/计数器和8253A

教师：苏曙光

华中科技大学软件学院

● 教学内容

- 第1节 定时器和计数器的概念
- 第2节 8253A的结构和工作原理
- 第3节 8253A的初始化和基本操作
- 第4节 8253A的工作方式和应用
- 第5节 8253A在IBM PC/XT中的应用





第1节 定时器和计数器的概念

- 定时（计时）

- 为**CPU**和外设提供时间标记或一段时间。

- 时序，各种周期，时间片、系统时间、时间间隔、运行时间

定时的方法

- 软件方法

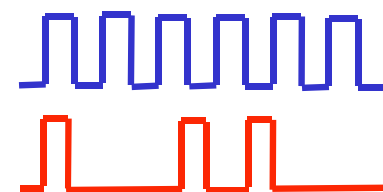
- 运用**循环执行一段指令**产生的一定时长的延时。
- 缺点：增加CPU开销；延时依赖CPU频率；
- 优点：不需增加硬件设备。
- 适用：用于短时且精度要求不高的延时

- 硬件方法

- 采用**专用电路**（例如：**定时/计数器**）产生定时。
- 优点：不占用CPU时间；定时准确不受主机频率影响
- 适用：定时时间长，精度要求较高
- 典型的定时专用电路/芯片：INTEL **8253/8254**系列芯片

- 计数/ Counter

- 统计某对象的数量 （典型的计数对象是脉冲）



- 定时/ Timer

- 定时的本质是计数 

- 当计数对象出现的“**时间间隔**”固定且已知，则
定时时间 = 计数数量 × 时间间隔



● 典型的计数/定时器 (Counter/Timer) 芯片

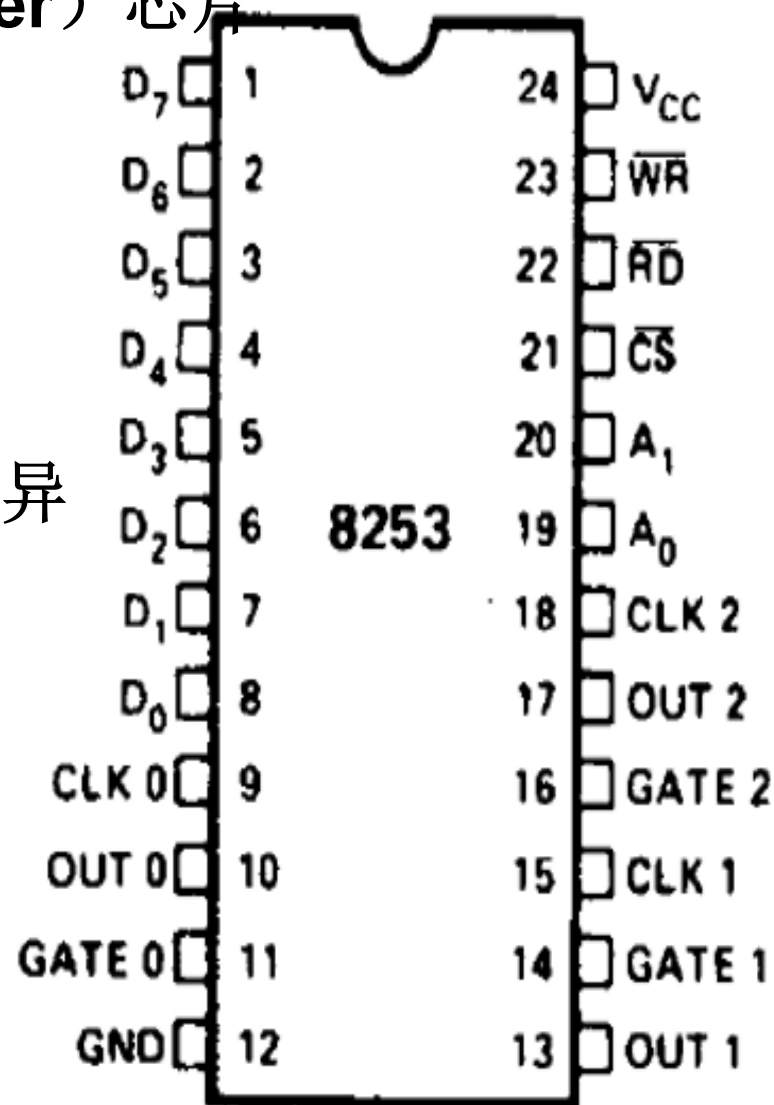
■ **8253A** (2MHz)

■ **8254A** (8MHz)

■ **8254A-2** (10MHz)

■ **8254A-5** (5MHz)

■ 特点：相互兼容，工作频率有差异



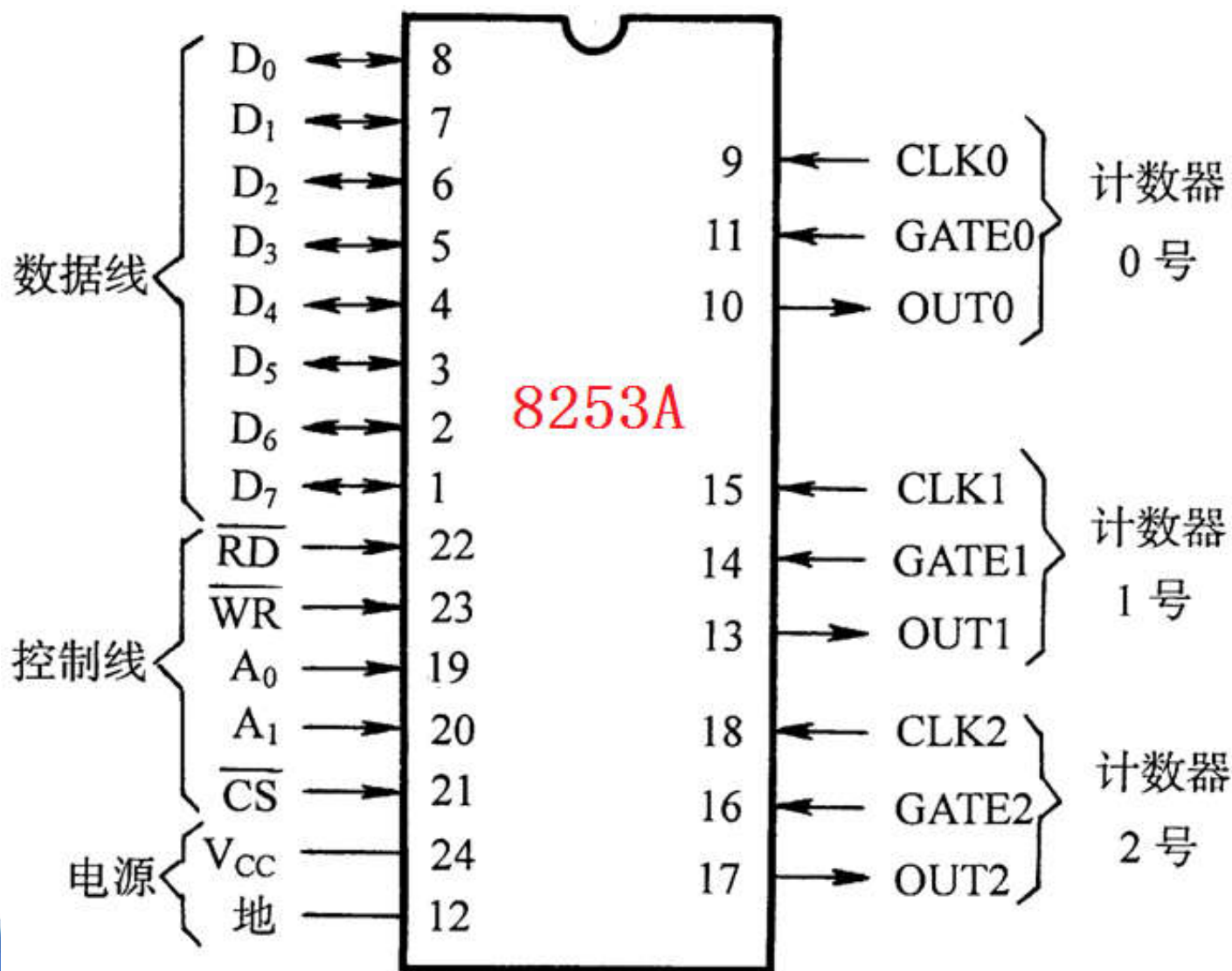


第2节 8253A的结构和工作原理

● 8253A的结构和基本特点

- 有**3个16位**相互独立的计数器：T0, T1, T2
- 每个计数器都可以按照**二进制**或**二—十进制**计数
- 每个计数器可设置**6种**不同的工作方式
- 每个计数器可以预置**计数初值**（**时间常数**）
- 计数器的**当前计数值**可被**CPU**读出

● 8253A的外部引脚【24脚，+5V】



● 功能引脚（面向CPU的信号线）

■ ① 数据总线D0~D7

◆ 三态输出/输入线：数据、命令和状态。

■ ② 片选线（ \overline{CS} ）

◆ 低电平有效。由地址译码的结果控制。

■ ③ 读信号（ \overline{RD} ）

◆ 低电平有效。对8253A寄存器进行读操作

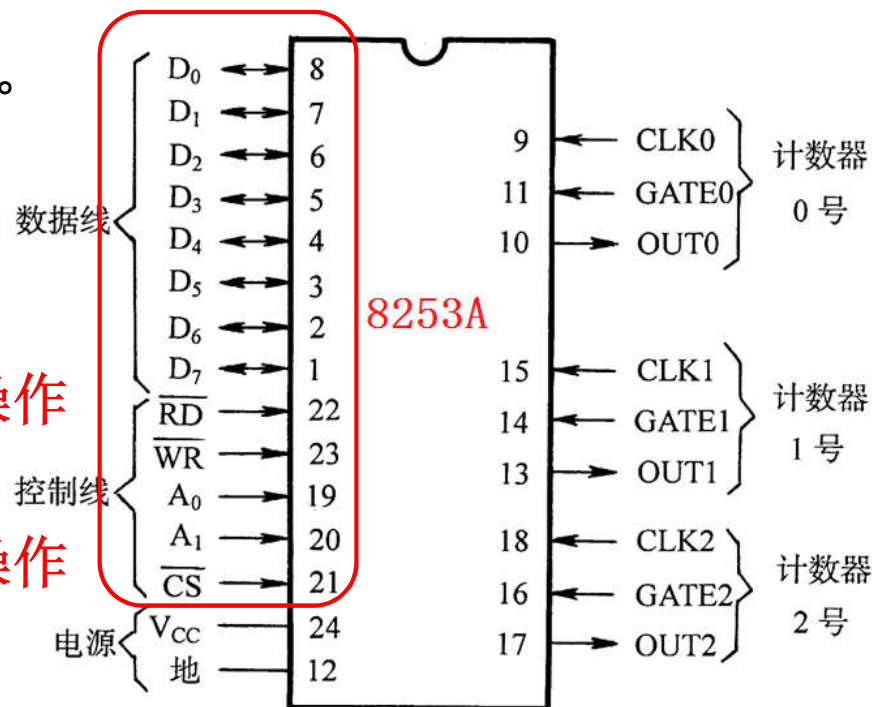
■ ④ 写信号（ \overline{WR} ）

◆ 低电平有效。对8253A寄存器进行写操作

■ ⑤ 地址线A1A0

◆ 接到系统地址总线的A1A0

◆ A1A0用于选择8253A内部寄存器。



● 功能引脚【面向I/O的信号线】

■ 三个计数/定时器，分三组

■ ⑥时钟信号**CLK**

◆计数的对象：每输入1个脉冲便计数1次。

◆**CLK 0、CLK 1、CLK 2**

■ ⑦门控信号**GATE**

◆控制计数的启动、暂停、禁止

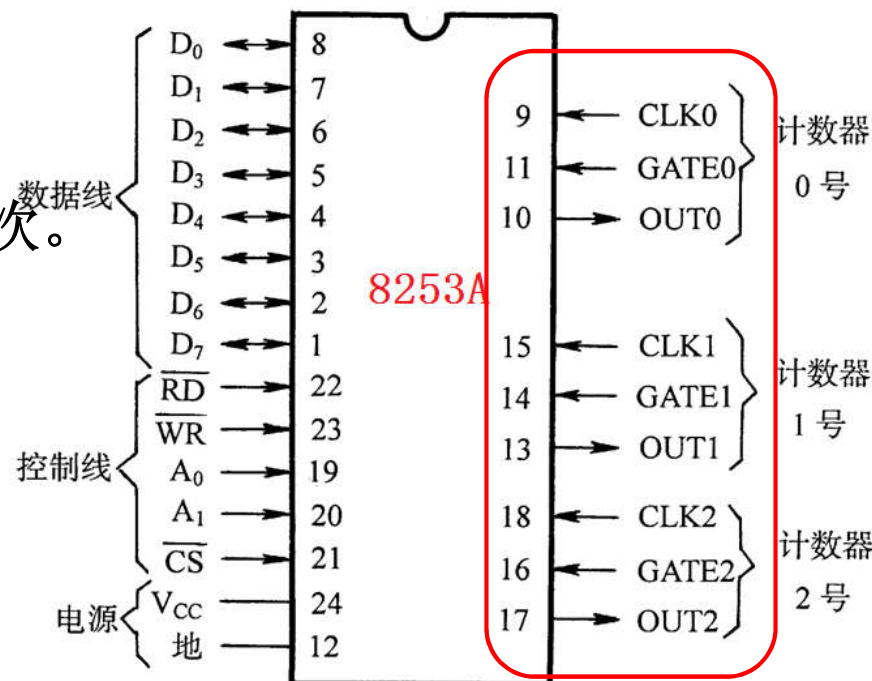
□具体功能和工作方式有关

◆**GATE0、GATE1、GATE2**

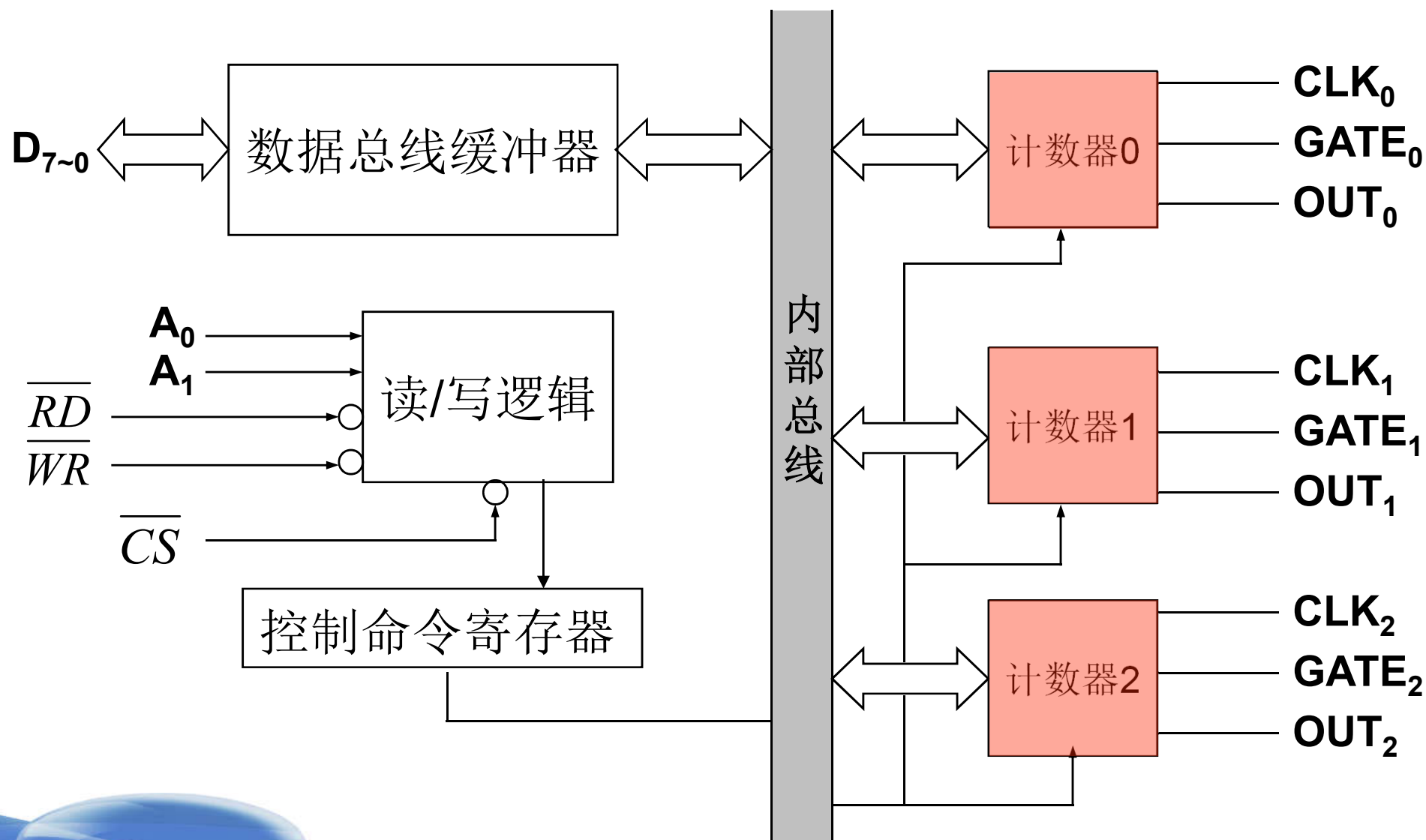
■ ⑧计数器输出**OUT**

◆输出特定波形标识定时或计数完毕或计数过程

◆**OUT0、OUT1、OUT2**



● 8253A的内部结构



- 8253A内部有6个模块

- ① 3个独立的计数器

- ◆ 结构完全相同，完成计数过程

- ② 数据总线缓冲器。

- ◆ 与CPU数据总线D0~D7相连

- 写入命令字；

- 写入计数初值；

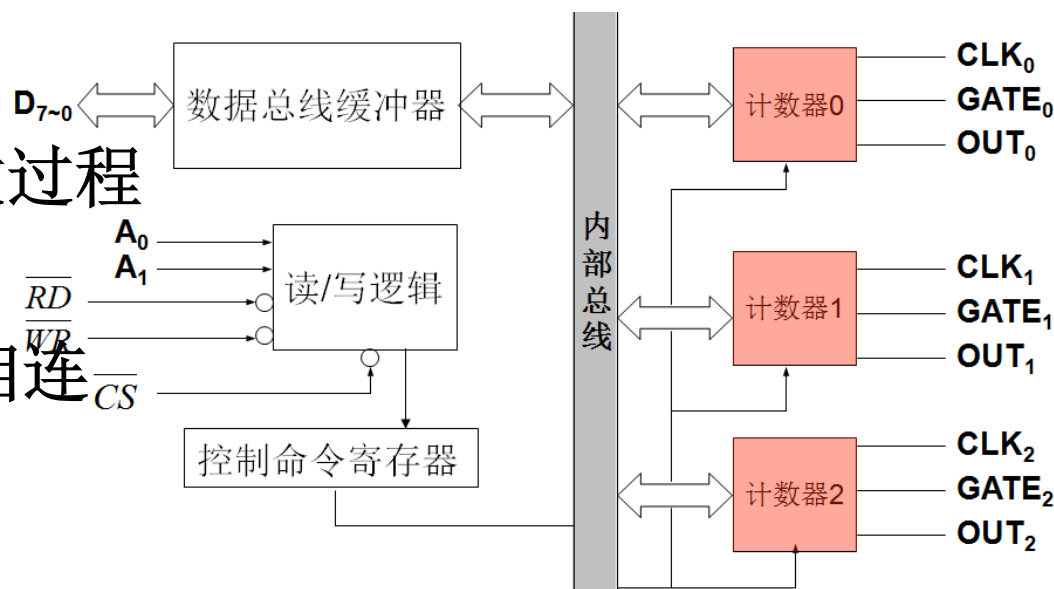
- 读出计数初值或当前值

- ③ 控制命令寄存器。

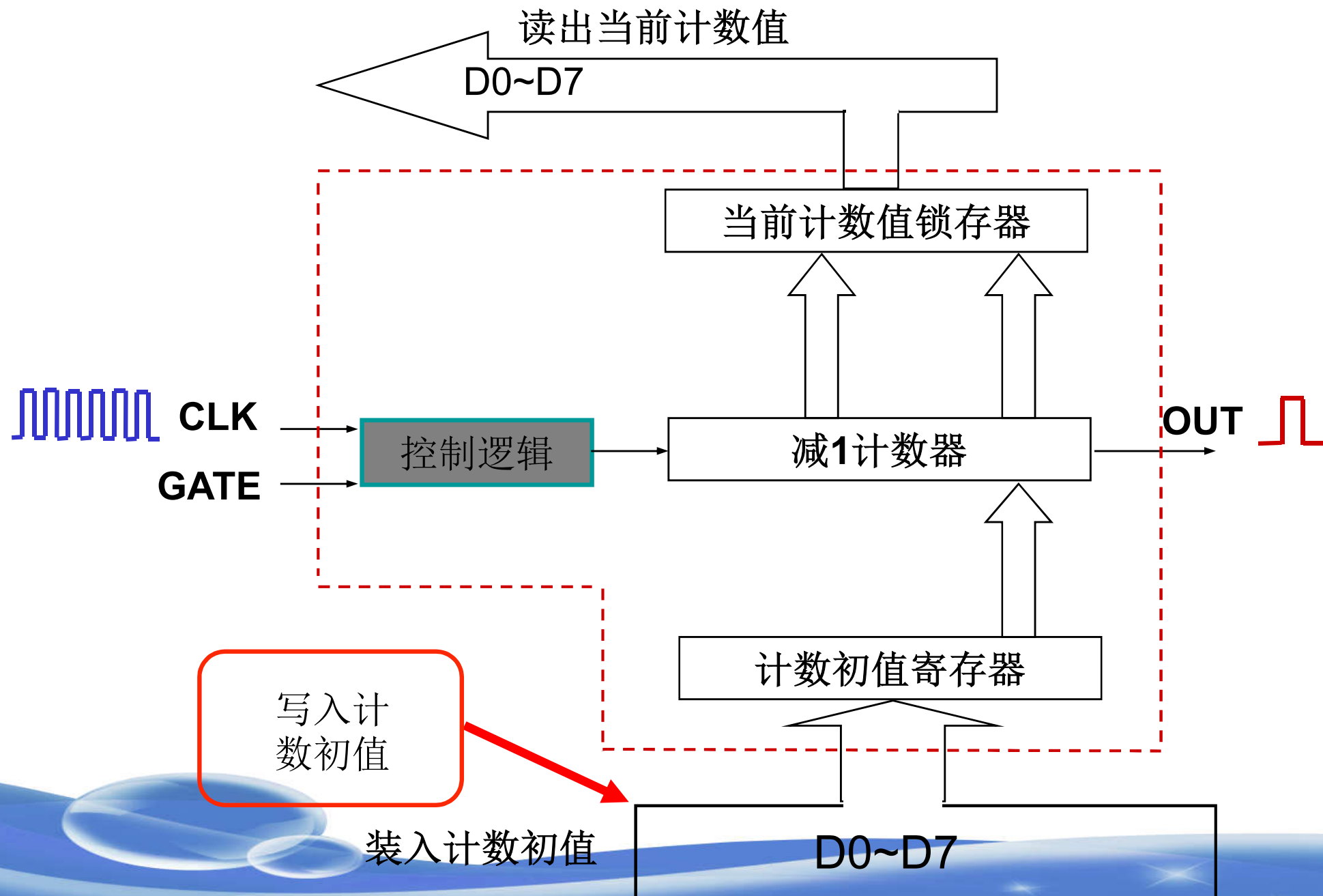
- ◆ 接收控制命令，选择计数器及设定工作方式

- ④ 读/写逻辑。

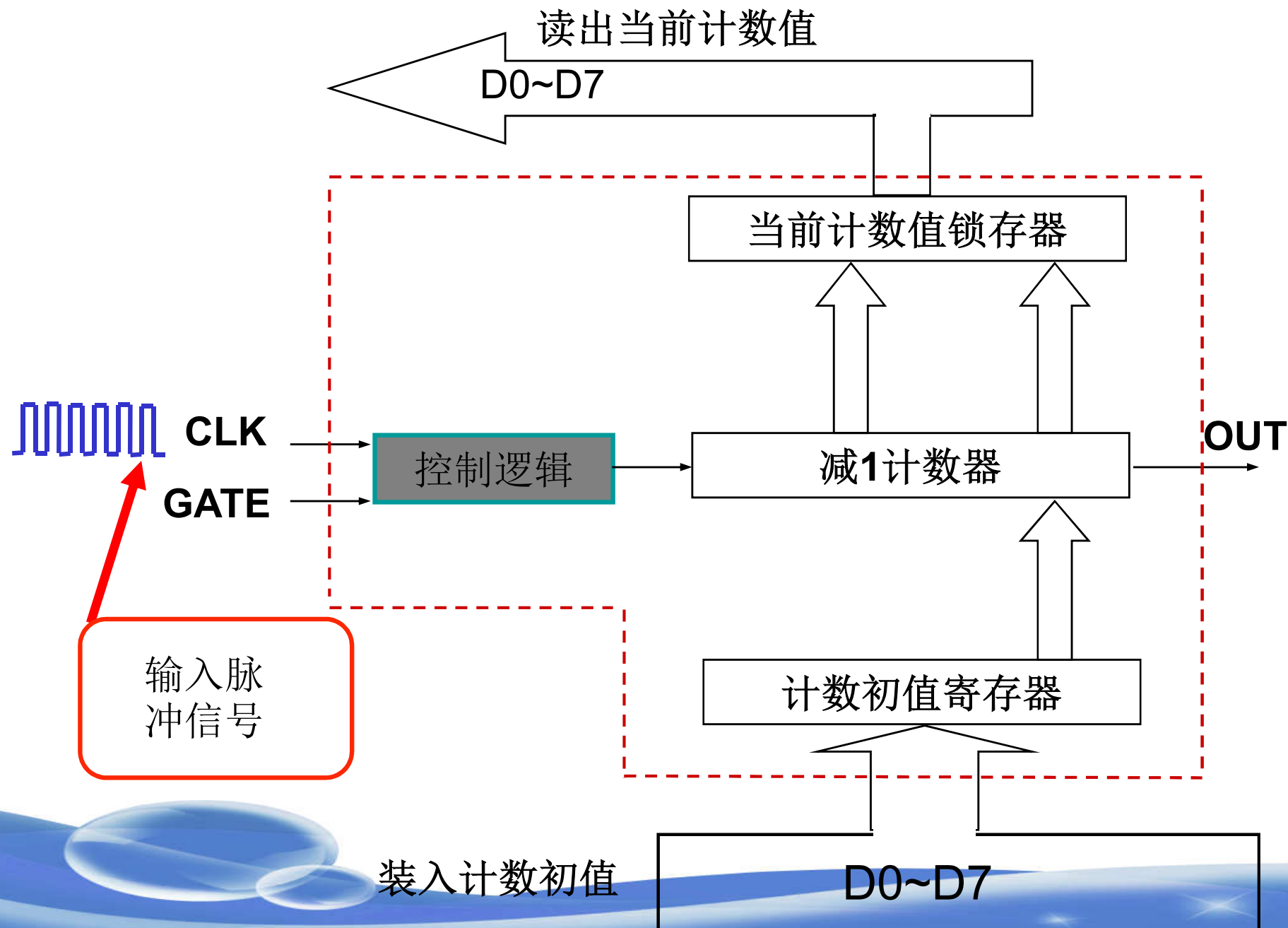
- ◆ 根据CPU读/写信号和地址信号选择数据传输方向和端口



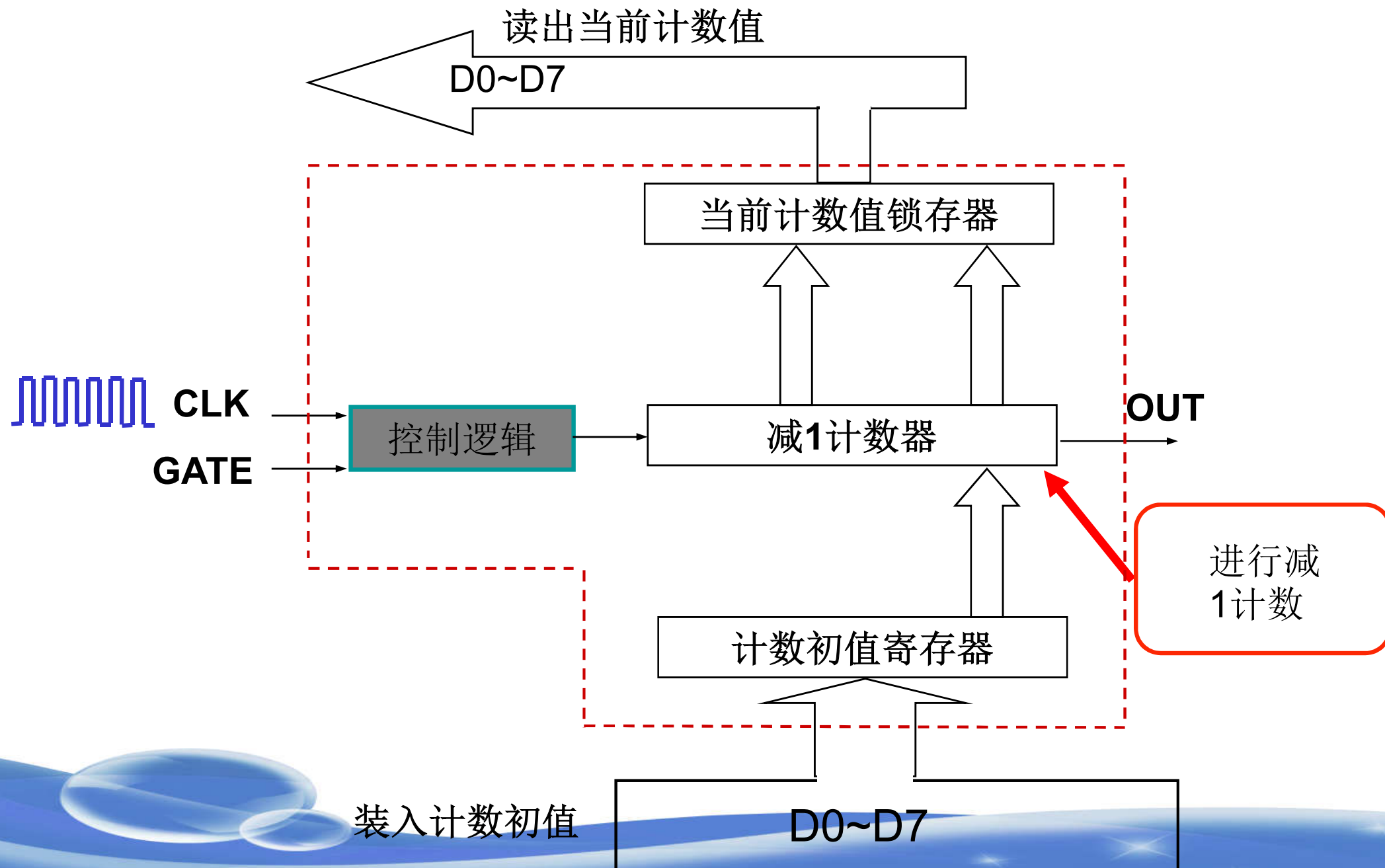
计数器工作原理



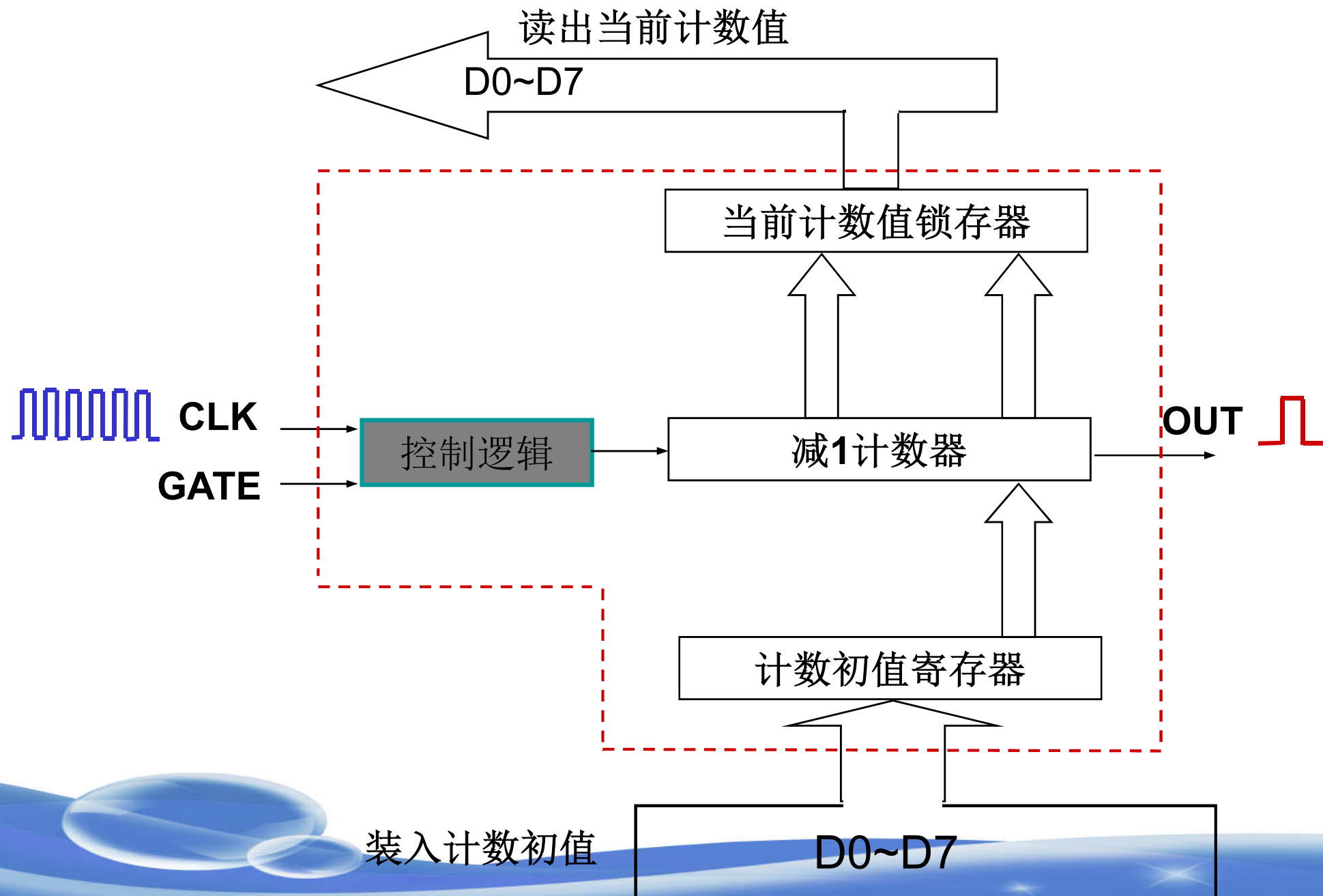
计数器工作原理



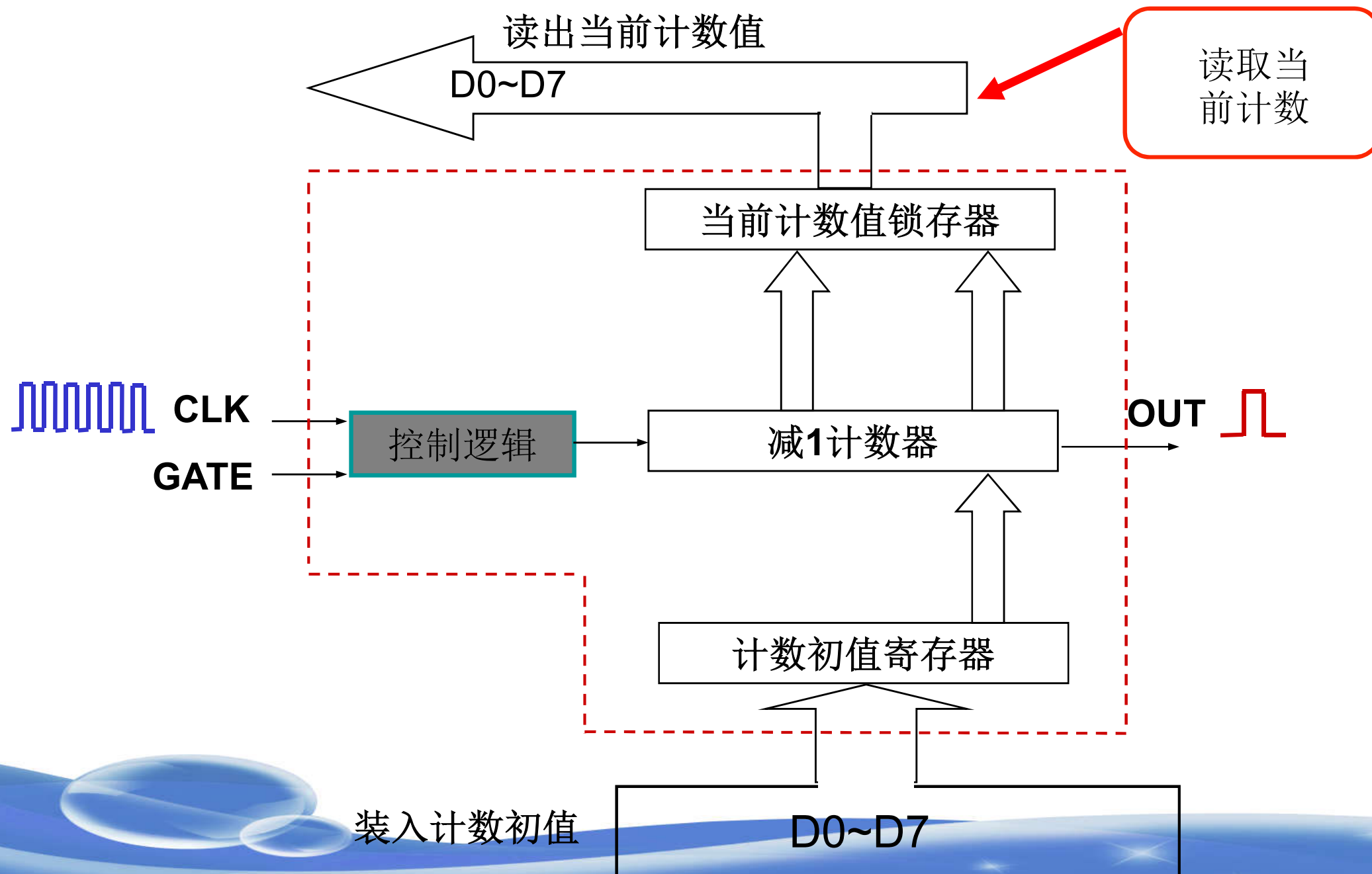
计数器工作原理



计数器工作原理



计数器工作原理



计数器工作原理

当前计数值锁存器（16位）：锁存**减1计数器**的内容供CPU读取。

读出当前计数值

D0~D7

减1计数器：16位，用于进行减1计数操作。初值和**初值寄存器**相同。每来1个CLK自动减1，直到0。



CLK

GATE

控制逻辑

当前计数值锁存器

OUT：计数过程中输出特定波形且计数结束时指示计数完成。

OUT

减1计数器

计数初值寄存器

计数初值寄存器：16位，存放计数初值，最大65535。计数中保持不变。**16位初值分两次**装入。

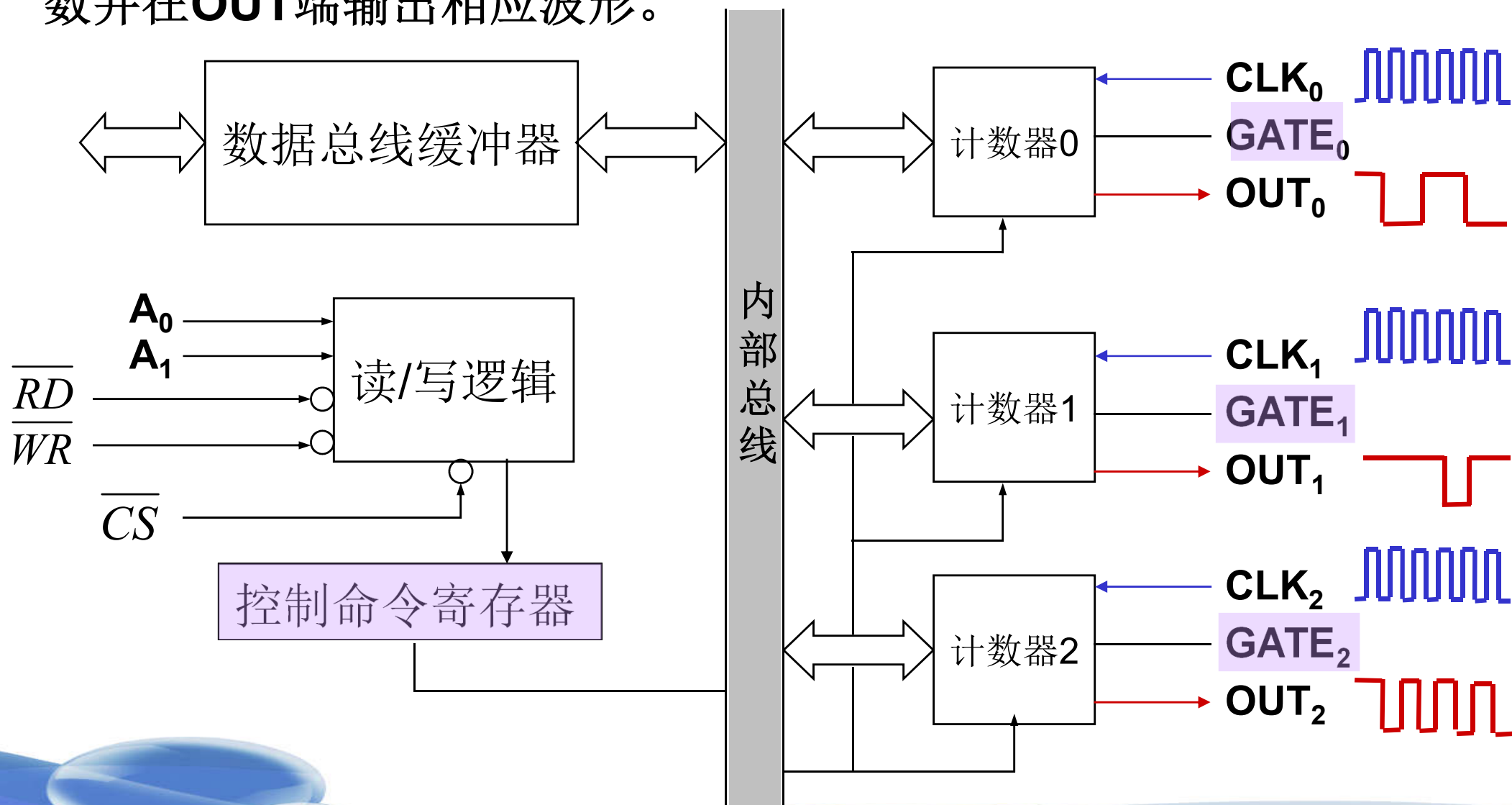
重新计数：把计数初值寄存器数据重新装入**减1计数器**。

装入计数初值

D0~D7

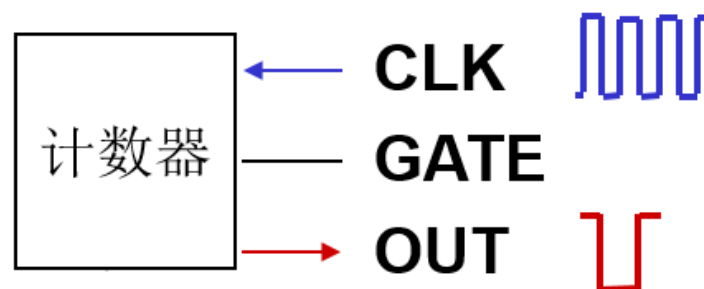
计数器工作原理

通过**控制命令**结合**GATE**，以某种特定方式对**CLK**脉冲进行计数并在**OUT**端输出相应波形。



计数初值C的确定

- 计数初值C决定了计数的次数或定时的长度



- 1.单纯的计数：直接设定
- 2.作为定时用，把时间L转成相应的计数C

◆ 假设时钟周期为 T_{CLK} （或频率 f_{CLK} ）

$$C = L / T_{CLK} = L * f_{CLK}$$

例：定时 5ms，时钟1.19318MHz，

$$\text{则 } C = L * f_{CLK} = 5 * 10^{-3} * 1.19318 * 10^6 = 5965$$

8253A的端口选择和操作

- 端口选择：4个端口：T0,T1,T2,控制端口
- 地址线：A1A0
- 读写操作：RD,WR

\overline{CS}	\overline{RD}	\overline{WR}	A1	A0	选中的对象和操作
0	1	0	0	0	T0, 写入“计数初值”
0	1	0	0	1	T1, 写入“计数初值”
0	1	0	1	0	T2, 写入“计数初值”
0	1	0	1	1	控制寄存器, 写“工作方式控制字”
0	0	1	0	0	T0, 读“当前计数值”
0	0	1	0	1	T1, 读“当前计数值”
0	0	1	1	0	T2, 读“当前计数值”
0	0	1	1	1	三态
1	×	×	×	×	三态
0	1	1	×	×	三态



第3节 8253A的初始化和基本操作

● 8253A的初始化

\overline{CS}	\overline{RD}	\overline{WR}	A1	A0	选中的对象和操作
0	1	0	0	0	T0, 写入“计数初值”
0	1	0	0	1	T1, 写入“计数初值”
0	1	0	1	0	T2, 写入“计数初值”
0	1	0	1	1	控制寄存器, 写“工作方式控制字”

■ 控制端口：工作方式控制字（选择计数器并设定工作方式）

- ◆（1）选择计数器：**T0, T1或 T2**
- ◆（2）确定读写数据方式（**8位或16位**，字节位置和顺序）
- ◆（3）确定计数器的工作方式（方式**0~方式5**）
- ◆（4）确定计数的数制（二进制码或**BCD码**）

■ **T0或T1或T2**端口：设定计数初值

- ◆写入计数初值：向**选定的计数器**写入计数初值

● 工作方式控制字：Control Word, CW

- (1) 选择计数器 (**T0, T1, T2**)
- (2) 确定读写数据的方式 (**8位或16位**, 字节位置和顺序)
- (3) 确定计数器的工作方式 (方式**0~方式5**)
- (4) 确定计数的数制 (二进制码或**BCD**码)

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	RW1	RW0	M2	M1	M0	BCD
计数器选择		读写数据方式		工作方式			码制

0 0 : T0
0 1 : T1
1 0 : T2
1 1 : 不用

00: 不用 (锁存)
0 1 : 低8位
1 0 : 高8位
1 1 : 先低后高

0 0 0 : 方式0
0 0 1 : 方式1
X 1 0 : 方式2
X 1 1 : 方式3
1 0 0 : 方式4
1 0 1 : 方式5

0 : 二进制
1 : BCD

课堂作业1

- 工作方式控制字 芯片地址304（即304~307且控制端口307）

■ 例： T1 ， 方式0， 二进制， 计数初值BYTEH: BYTEL。

```
MOV _____ ; 命令口
MOV AL, _____ ; 工作方式控制字
OUT DX, AL ; 写入命令寄存器
MOV _____ ; T1数据口
MOV AL, BYTEL ; 计数初值低字节
OUT DX, AL
MOV _____ ; 计数初值高字节
OUT _____
```

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	RW1	RW0	M2	M1	M0	BCD
计数器选择		读写数据方式		工作方式			码制

0 1 1 1 0 0 0 0

● 8253A的基本操作

- 获取当前计数值：直接读取或锁存命令
- 获得工作状态：获得状态字
- 通过向控制端口写特定的字完成。

获取当前计数值：直接读取

- 方法一

- 使用IN指令读取（两次）

- 端口选择：**A1A0**

- 端口操作： **\overline{RD} , \overline{WR}**

\overline{CS}	\overline{RD}	\overline{WR}	A1	A0	选中的对象和操作
0	0	1	0	0	T0, 读“当前计数值”
0	0	1	0	1	T1, 读“当前计数值”
0	0	1	1	0	T2, 读“当前计数值”

- 读计数器**T0**的当前计数值（假定地址：**40- 43**）

```
IN    AL,    40H    ; 读入计数器0的低8位
MOV   BL,    AL
IN    AL,    40H    ; 读入计数器0的高8位
MOV   BH,    AL
```

- 缺陷
 - 读数不准确
- 改进措施
 - 硬件配合：先禁止计数，再读数

获取当前计数值：锁存命令

● 锁存命令

■ 将减1计数器的内容锁存到当前计数值锁存器，供CPU读取

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	0	0	x	x	x	x
计数器选择		锁存特征位		不用			

0 0 : T0
0 1 : T1
1 0 : T2
1 1 : 不用

0 0 : 锁存
0 1 : 不用
1 0 : 不用
1 1 : 不用

} 读写数据方式。

● 锁存命令

芯片地址304（即304~307且控制端口307）

■ 例：读出T1当前计数值是否全“1”（假定只用低8位）。

```
MOV DX, 307H      ;命令口
MOV AL, 01000000B ; T1的锁存命令
OUT DX, AL        ; 写入命令寄存器
MOV DX, 305H      ; T1数据口
IN AL, DX         ; 读T1的当前计数值
CMP AL, 0FFH      ; 比较
```

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	0	0	x	x	x	x
计数器选择		锁存特征位		不用			
0	1	0	0	0	0	0	0

课堂作业2

● 锁存命令

芯片地址304（即304~307且控制端口307）

■ 例：读出T2当前计数值（16位），并装入AX寄存器。

```
MOV _____ ;命令口
MOV AL, _____ ; T2的锁存命令
OUT DX, AL ; 写入命令寄存器
MOV _____ ; T2数据口
IN AL, DX ; 读T2的当前计数值的低8位
MOV BL, AL
IN _____ ; 读T2的当前计数值的高8位
MOV _____
MOV AL, BL
```

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	0	0	x	x	x	x
1	0	0	0	0	0	0	0



第4节 8253A的工作方式和应用

- 工作方式

 - 方式0~方式5

- 工作方式的主要差异

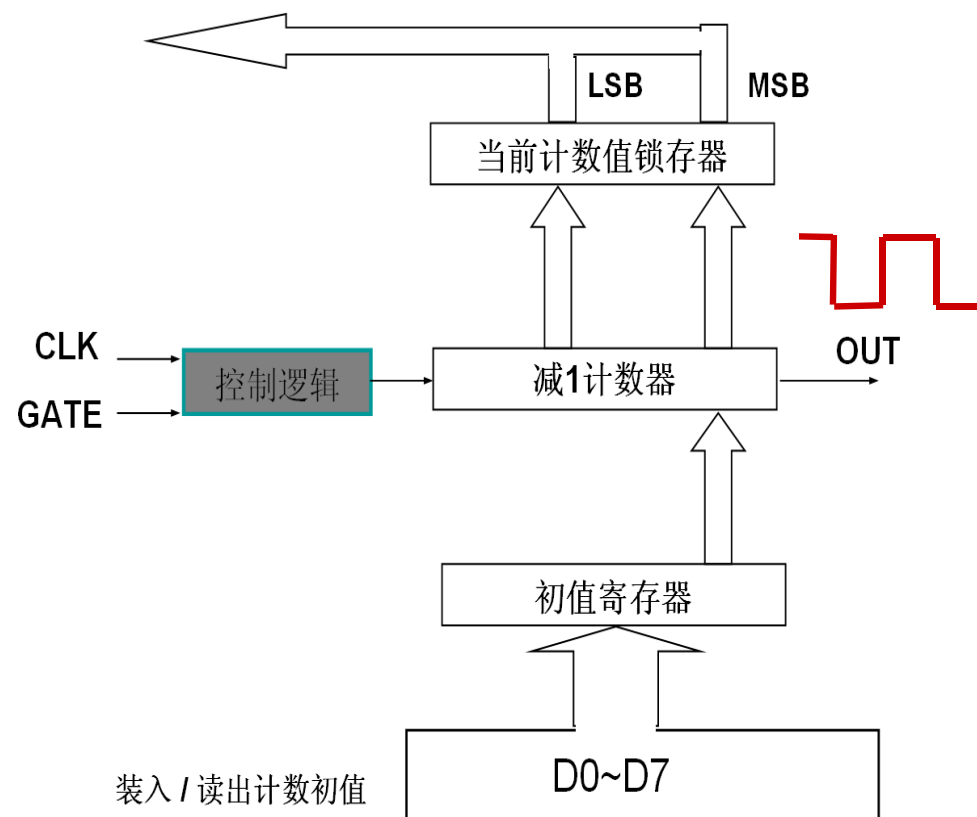
 - OUT信号或波形

 - 计数的启动方式

 - ◆ 软件启动

 - ◆ 硬件启动 (**GATE**)

 - 初值 (/重新) 装入的生效方式

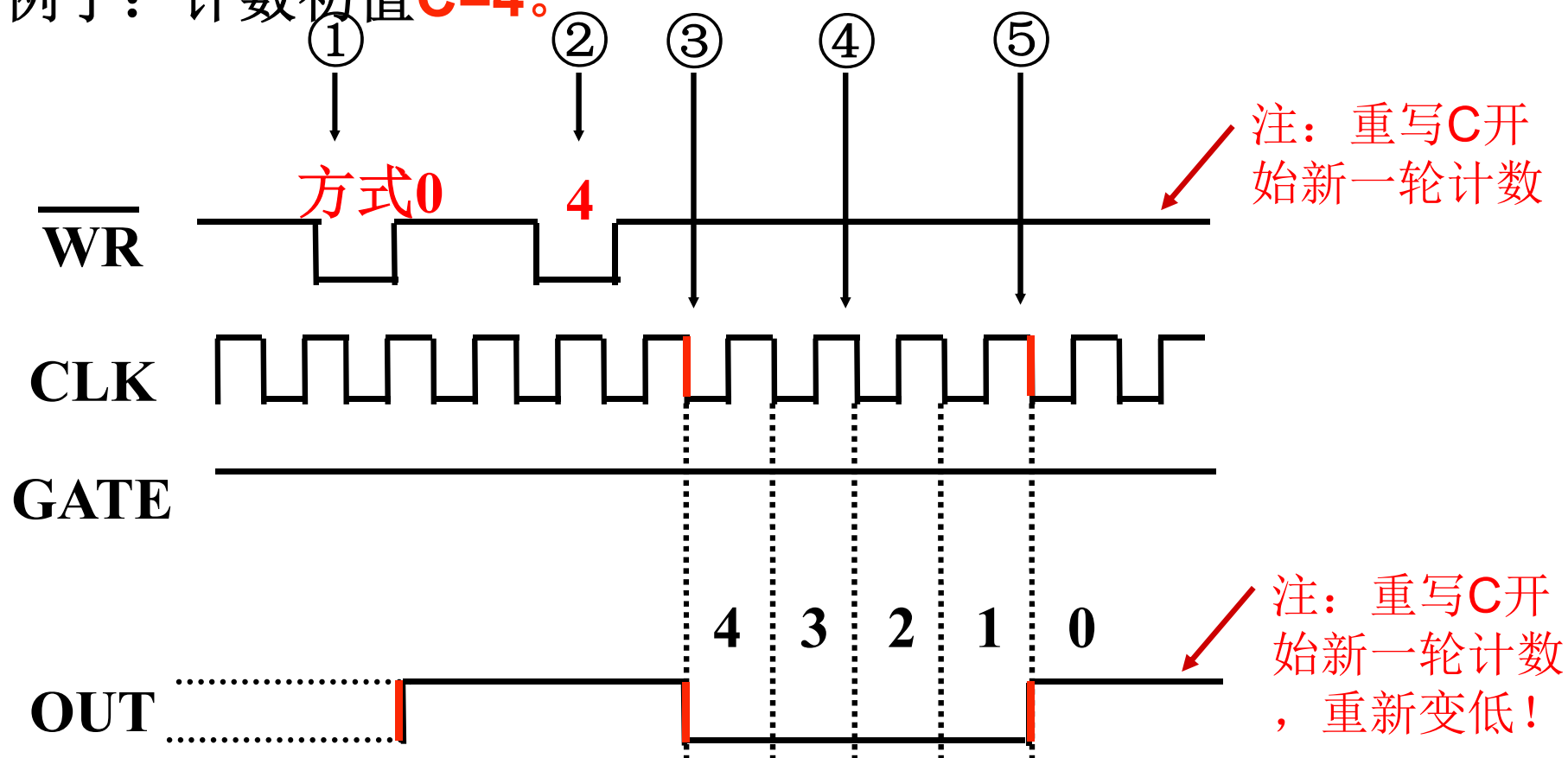


方式0：计数结束中断方式

- 基本特点

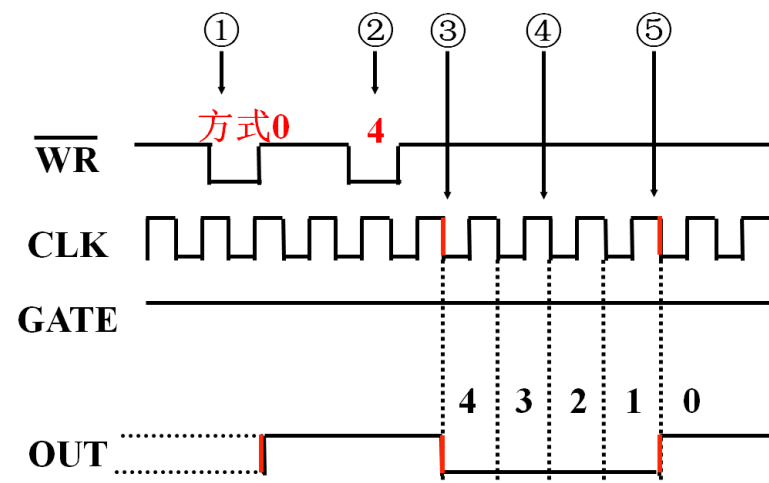
- 典型的事件计数用法

- 例子：计数初值 **C=4**。



方式0的计数过程和特点

- 1、写入**工作方式控制字**（**CW, Control Word**）
 - **OUT**开始变成**高电平**
- 2、写**计数初值C**到**初值寄存器**
 - 下一个**CLK**周期把**C**装入**减1计数器**。
 - **GATE****高电平**时立即开始计数。
- 3、计数期间**OUT**维持**低电平**。
- 4、当**减1计数器**减到**0**时**OUT**变为**高**电平。
- 5、当重新写入**C**后，立即开始新一轮计数
 - **OUT**再次变成低电平
 - 计数期间，如果重写计数值，立即**重新开始**计数。
- 6、**GATE**：高电平：允许/继续计数；低电平：禁止计数
- 7、**OUT**可作中断请求信号【特点：被响应后才变低】

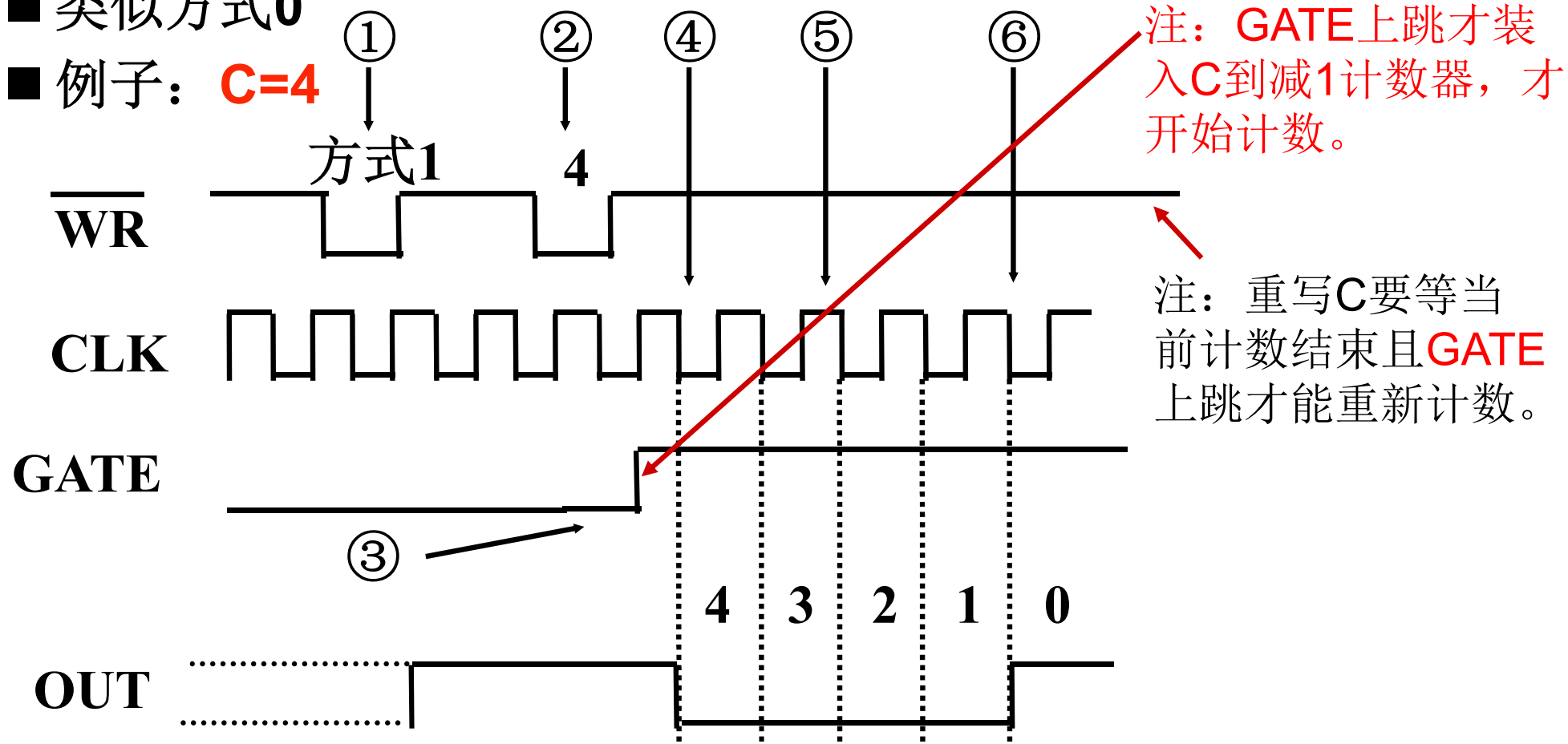


方式1

- 基本特点

- 类似方式0

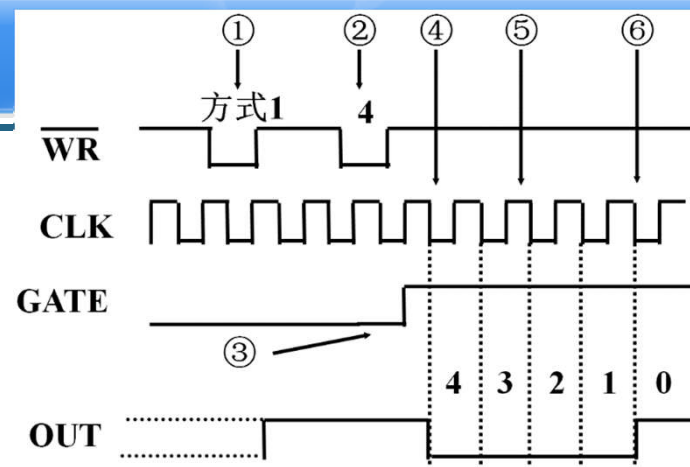
- 例子: **C=4**



方式1的计数过程和特点

同方式0很类似

- 1、写入控制字**CW**后，**OUT**变**高**电平。
- 2、写入计数初值**C**
- 3、**GATE**上跳时，**C**装入**减1计数器**，开始计数，**OUT**变**低**。
- 4、整个计数过程中**OUT**维持低电平，直到计数结束才变高。
 - **OUT**负脉冲宽度 = 计数初值 * **CLK**周期。
- 5、如果计数过程中**GATE**出现上跳，则重新计数。
- 6、计数过程中如果重写初值，则要等当前计数结束且**GATE**再次出现上升沿后，才能开始新的计数。



方式1的应用场合

- 改变计数初值**C**可产生**不同宽度**的低电平，获取**可变宽脉冲信号**



■ 实现脉宽调制（ **PWM** ， **Pulse Width Modulate** ）

◆ 用**数字信号**去控制/产生**模拟信号**的方法。

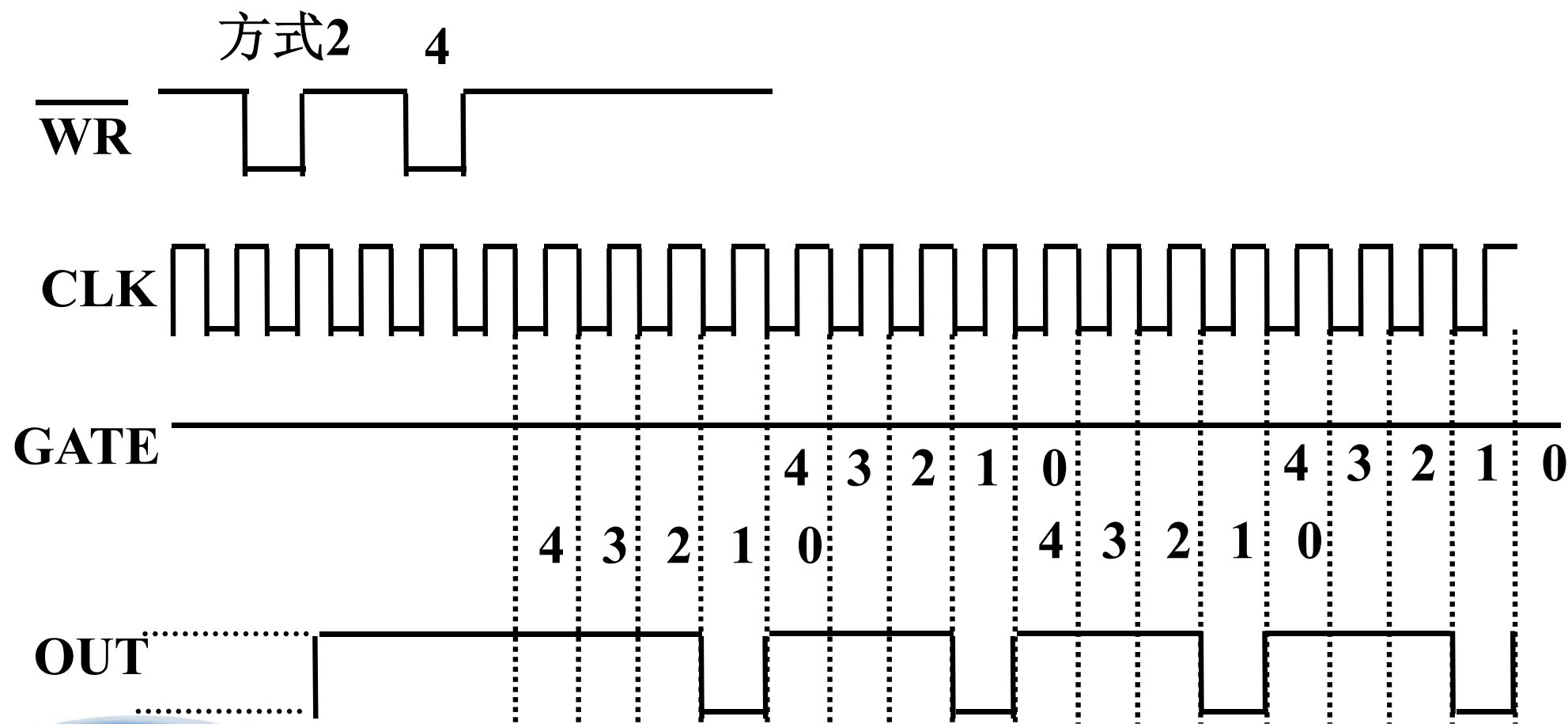
□ 利用微机的**数字输出**来对**模拟电路**进行控制的技术，
广泛应用于测量、通信、功率控制与变换的领域中。

◆ 对模拟信号电平进行数字编码的方法

□ 对**占空比**的编码

方式2：周期性负脉冲输出，分频器

- 若计数初值为**N**，则**OUT**频率为**CLK**的**1/N**。又称**N**分频器。
- 例子：**C=4**



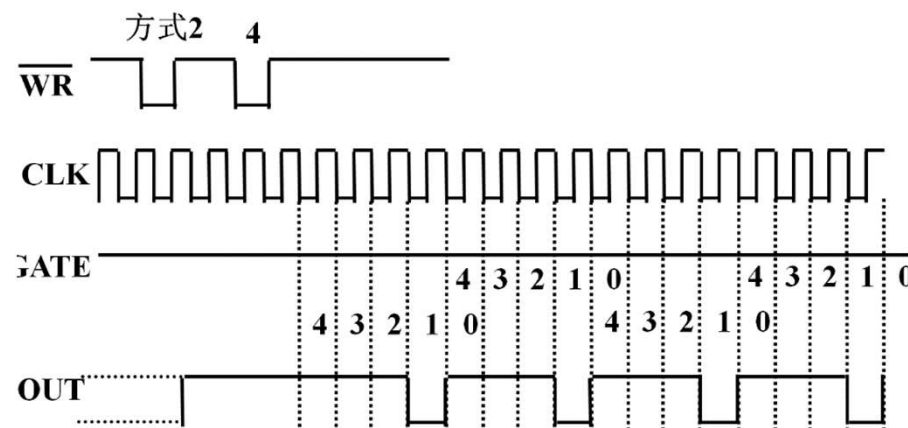
方式2的计数过程和特点

- 1、写入控制字**CW**，**OUT**变成高电平。
- 2、写入**计数初值C**。
- 3、若**GATE**为高电平时，下一个**CLK**周期把**C**写入**减1计数器**，并开始计数。
- 4、计数过程中**OUT**保持**高**电平，直到倒数第**2**个**CLK**。
- 5、减一计数到**1**时，**OUT**输出**1个负脉冲**（**宽度为1个CLK周期**），
- 6、**自动装入C**，**OUT**再次变成高电平，开始新一轮计数。
- 注意：

■ 方式2中启动计数器的方法有两种

◆ 软件启动过程：装入计数初值

◆ 硬件启动过程：**GATE**上升沿



方式2的应用场合

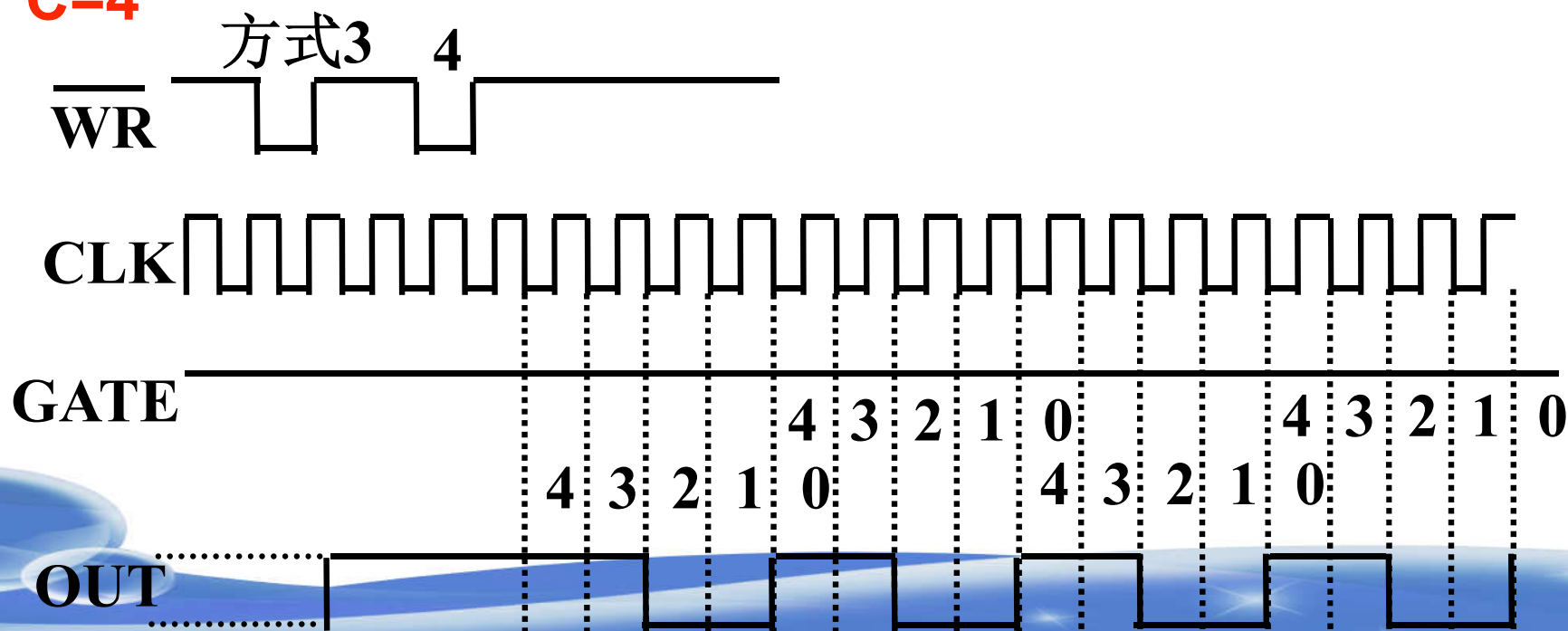
- **OUT**正脉冲宽度 = (计数初值 - 1) x **CLK**周期
- **OUT**负脉冲宽度 = 1个**CLK**周期。
- 典型应用：**N分频器**（即速率发生器）：
 - 对已知频率的信号（**CLK**），通过改变计数初值**C**，即可获得不同频率的信号（**OUT**）

$$f_{\text{OUT}} = f_{\text{CLK}} / N$$

方式3：周期性方波输出

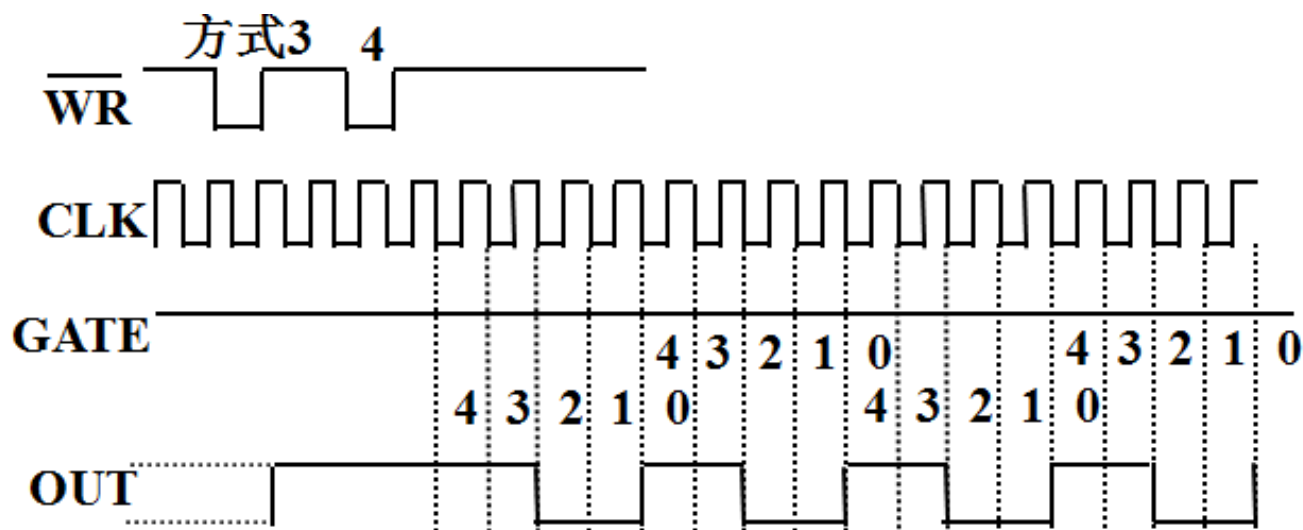
- 与方式2基本相同：自动装入计数初值C，循环计数
- 不同之处：OUT输出占空比为1:1或近似1:1的方波
 - C为偶数时，OUT的高、低电平前后各一半。
 - C为奇数时：
 - ◆ 前 $(N+1)/2$ 个CLK：OUT高电平
 - ◆ 后 $(N-1)/2$ 个CLK：OUT低电平。

- 例子：C=4



方式3的特点和应用场合

- 方式3的特点
 - 1、减1计数器每次计数减2。
 - 2、OUT输出方波信号
 - 3、计数器有软件启动和有硬件启动两种方式。
- 方式3的应用场合
 - 波特率发生器（或速率发生器）



方式4：单次负脉冲输出

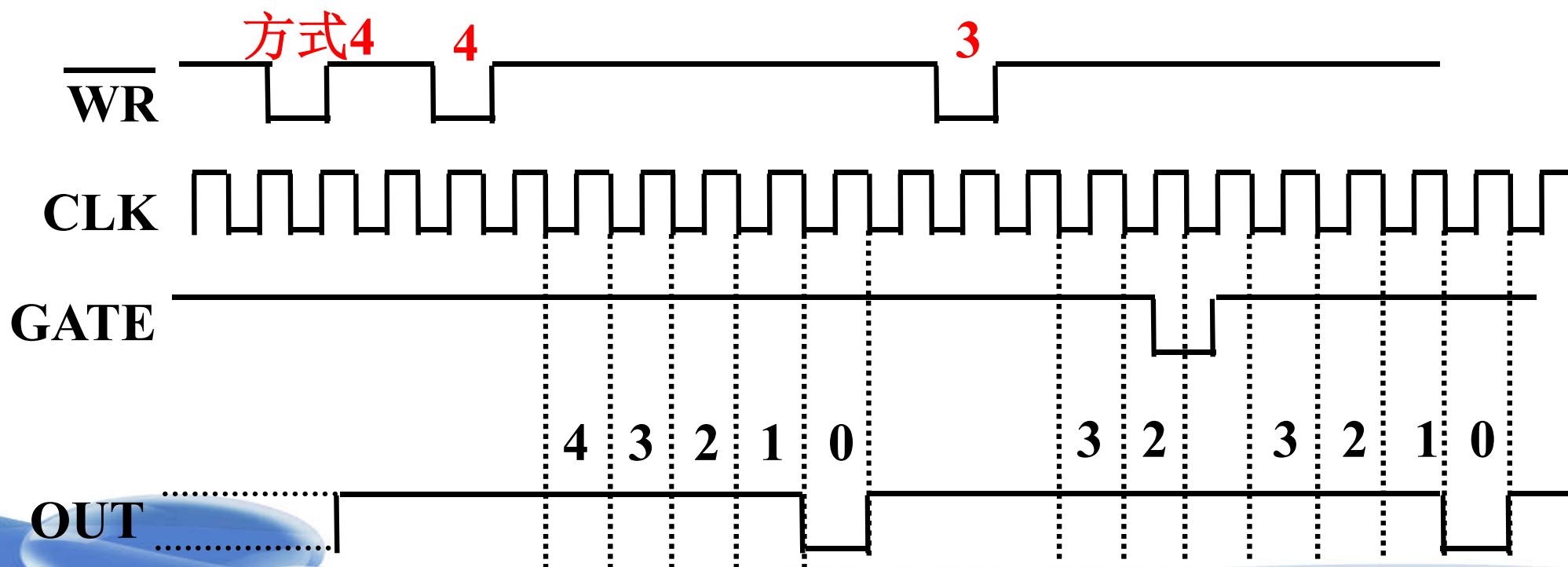
- 方式4：单次负脉冲输出，软件触发

- 例子：C=4（后面改为3），注意观察：

- ◆ 负脉冲位置；

- ◆ 重写新的C时，计数过程的变化；

- ◆ GATE变低然后恢复高时，计数过程的变化



方式4的计数过程和特点

- 由**软件**启动计数：写入计数初值**C**触发计数器开始工作；
- 1、**设定方式**后，**OUT**为**高**电平；
- 2、写入**C**后，在下一个**CLK**开始计数
- 3、计数期间**OUT**保持**高**电平
- 4、计数结束输出1个**CLK**的**负**脉冲，然后**OUT**恢复**高**电平。

- 5、**GATE**作用

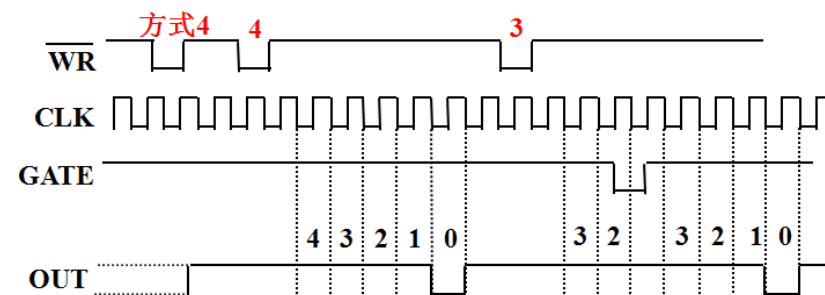
- ◆高电平：允许计数；

- ◆低电平：停止（**不是暂停**）计数

- ◆恢复高电平：重新开始**新一轮**的计数。

- 6、重写新的计数初值

- 不影响当前计数过程，直到**下一轮**计数时才生效。



方式5：单次负脉冲输出

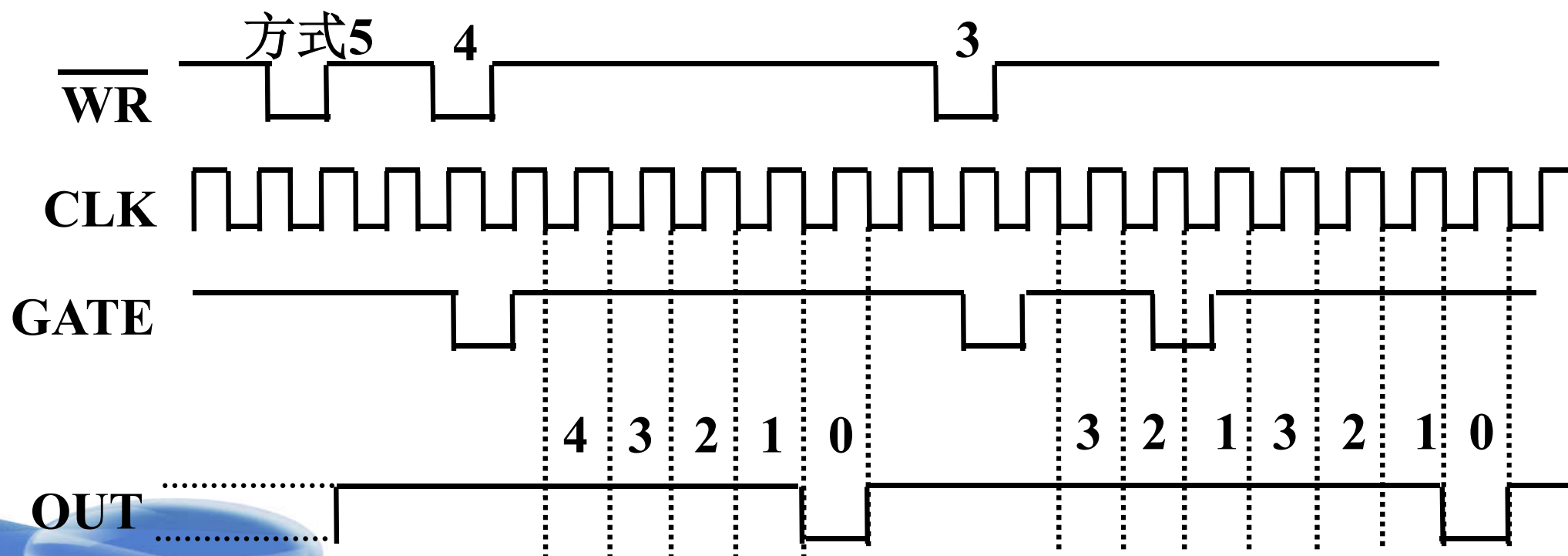
- 方式5：同方式4类似，单次负脉冲输出，但是**硬件**触发

■ 例子：**C=4**，注意观察

◆ **负脉冲**位置；

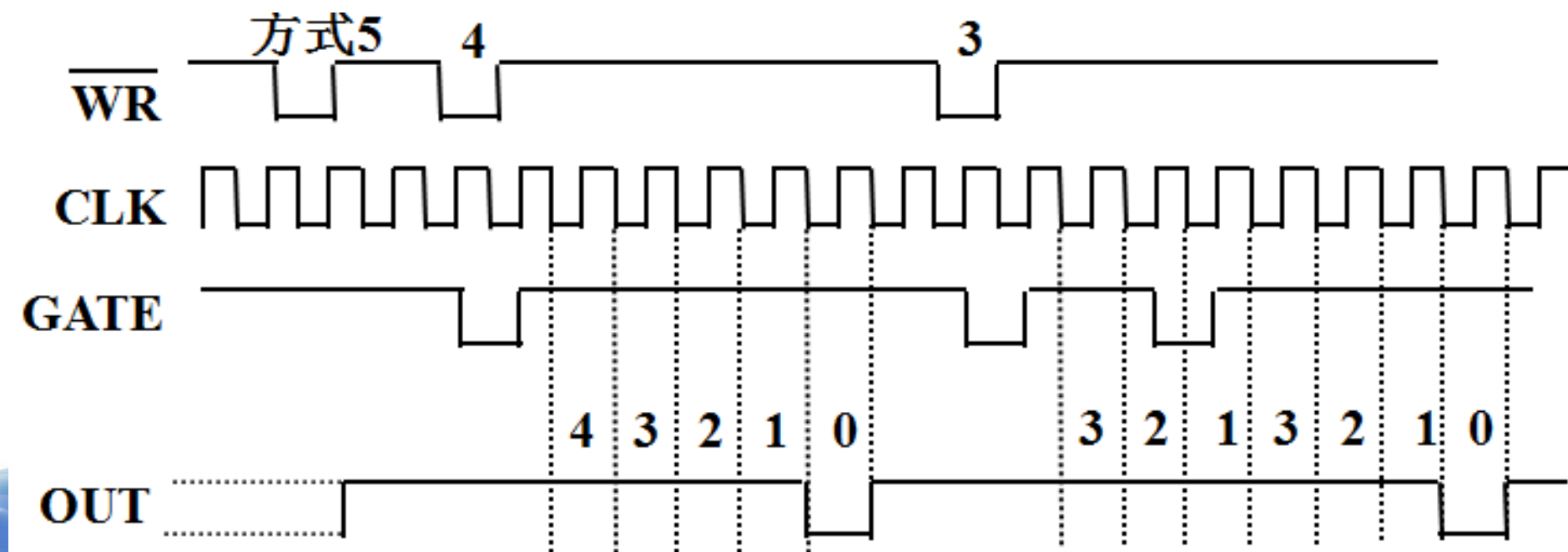
◆ 重写新的**C**时，计数过程的变化；

◆ **GATE**变**低**然后恢复**高**时，计数过程的变化



方式5的计数过程和特点

- 同方式4类似，单次负脉冲输出，**硬件**触发
 - **GATE**上升沿触发计数开始
- 1、写入**C**后，不立即计数，而要由**GATE****上升沿**启动计数。
- 2、计数过程中**OUT**维持高电平
- 3、计数到0后，**OUT**输出1个**CLK**的负脉冲后恢复高电平。
- 4、在计数过程中，如**GATE**出现**上升沿**，立即开始新一轮计数。



6种工作方式的总结

- 差异

- 计数过程和计数结束时OUT电平的高低和持续时间（波形）
- 单次计数或自动循环计数
- 计数的启动方式：软件（写计数初值）或硬件（GATE）

- 方式0和方式 1

- 输出波形类似;
- 无自动重装C的能力;
- 启动计数的触发信号不一样:

- ◆ 方式0: 软件 (写初值)

- ◆ 方式1: 硬件 (**GATE**上沿)

- 方式2 (N分频器) 和方式3 (方波发生器)

- 计数初值自动重装, 循环计数;
- **OUT**频率:**CLK**的N分之一;
- 方式2: 计数时高电平, 结束时1个**CLK**负脉冲;
- 方式3: 前半为高, 后半为低

6种工作方式的比较——

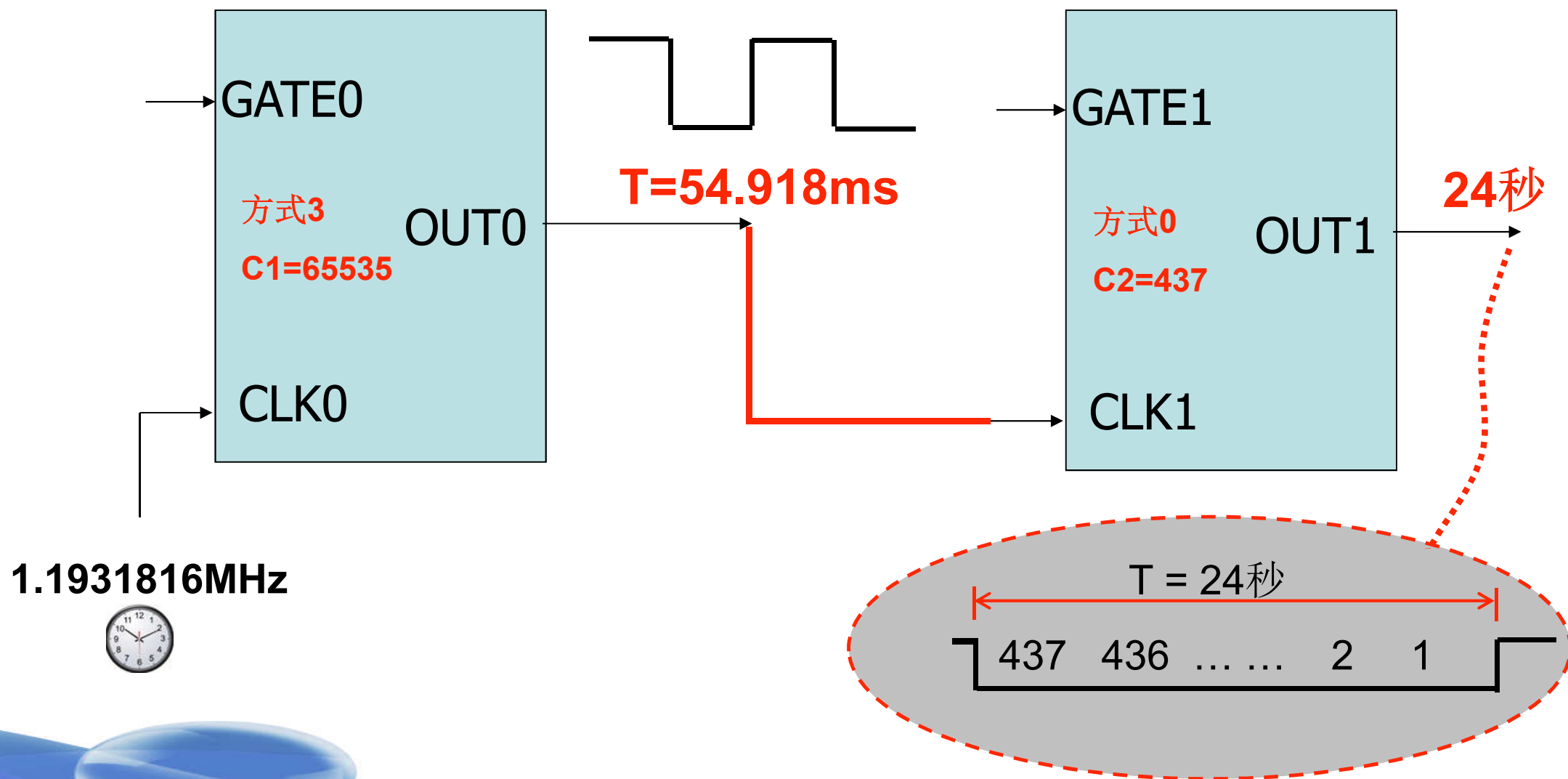
- 方式4（单次负脉冲）和5方式（单次负脉冲）
 - 输出波形相同：单次负脉冲；
 - 无自动重装能力；
 - 启动计数方式不同：方式4：软件，方式5：硬件
- 方式2（N分频）和方式4与5
 - 方式2：周期性负脉冲
 - 方式4与5：单次负脉冲

8253A的应用举例:24秒计时

- 已知计算机系统的计时单位（时钟脉冲）
 - $\text{CLK} = 1.1931816\text{MHz}$
 - $T_{\text{CLK}} = 0.84\text{微秒}$
 - ◆ 若使用最大的计数初值 $C1 = 65535$
则 $T_{\text{OUT}} = 0.84 * 65535 = 54.918\text{ ms}$
- 新的计时单位
 - 新建一个周期性的波形(方式3, 输出周期54.918ms)
 - ◆ 计数初值 $C1 = 65535$
 - ◆ $T_{\text{OUT}} = 0.84\text{微秒} * 65535 = 54.918\text{ms}$
 - 新的时钟信号 $T_{\text{CLK-new}} = 54.918\text{毫秒}$
- 在新的时钟 $T_{\text{CLK-new}}$ 下, 计时24秒
 - 方式0, 计数结束输出高电平。
 - 计数初值 $C2 = 24\text{秒} / 54.918\text{毫秒} = 437$
- 定时器选用: T0,T1
 - T0 :方式3, $C1 = 65535$, CLK0—CLK, OUT0—CLK1
 - T1 :方式0, $C2 = 437$, CLK1 —OUT0, OUT2— 定时到信号

● 8253A的应用举例:24秒计时

假定: 8253A地址: 0x128



8253A的应用举例:流水线产品计数控制

- 每**200**个玩具打包装箱 (加装**红外感应器**, **8253A**)

