



# 第5章 时序逻辑电路

## 5.1 时序逻辑电路概述

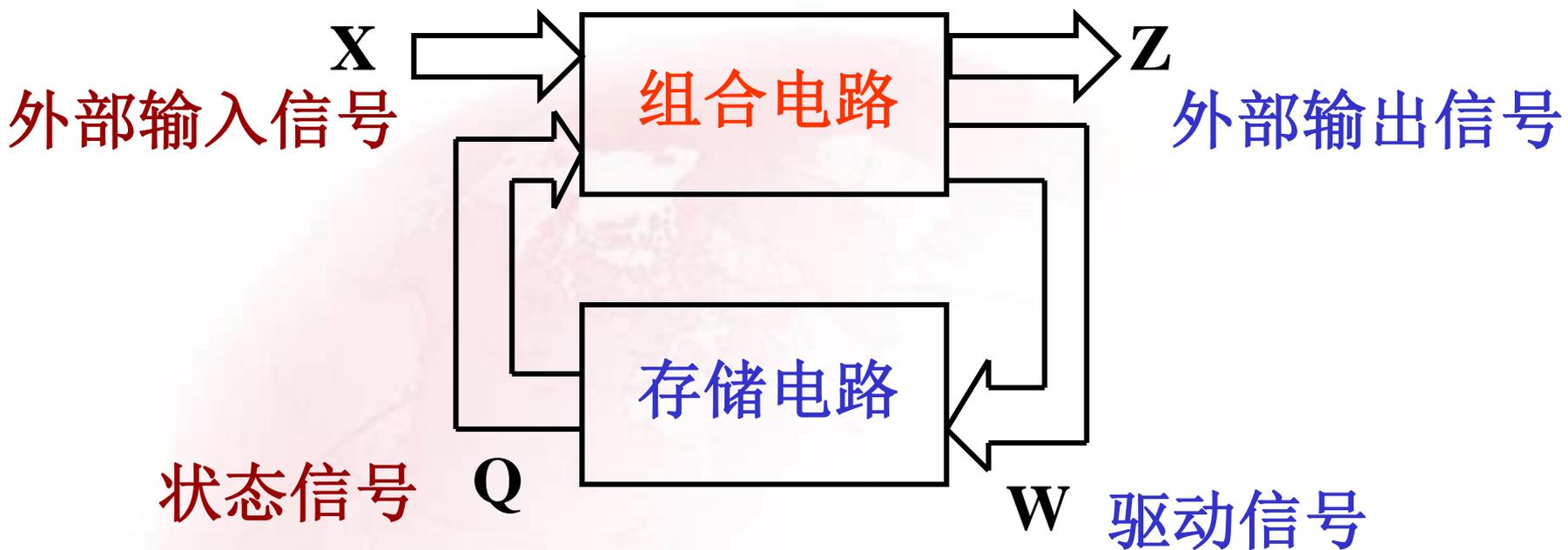
### 1. 时序逻辑电路的基本概念

时序逻辑电路的特点：

电路在任何时候的**输出稳定值**，不仅与该时刻的**输入信号**有关，而且与该时刻**以前的电路状态**有关；电路结构具有**反馈回路**。



## 2. 时序逻辑电路的结构模型





## 3. 时序逻辑电路的描述方法

### (1) 逻辑方程

输出方程:

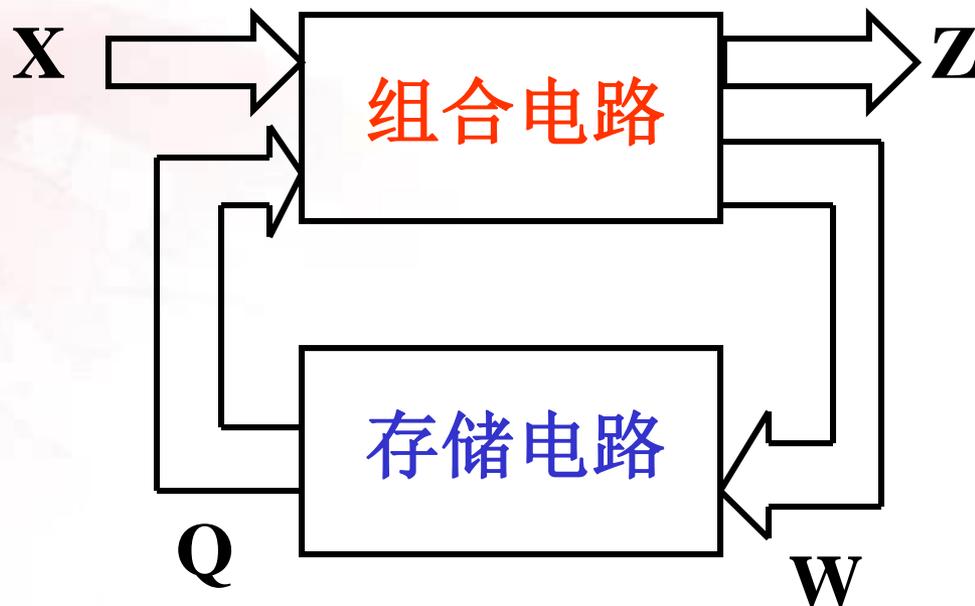
$$Z(t_n) = F[X(t_n), Q(t_n)]$$

驱动方程:

$$W(t_n) = G[X(t_n), Q(t_n)]$$

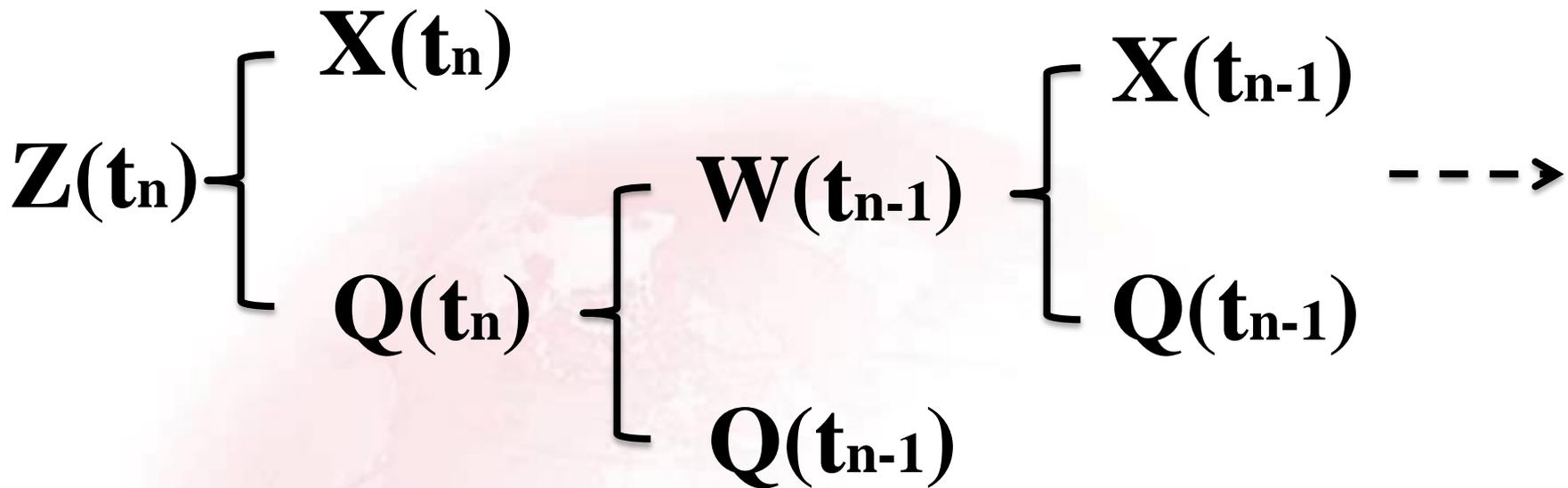
状态方程:

$$Q(t_{n+1}) = H[W(t_n), Q(t_n)]$$





## 信号流程:

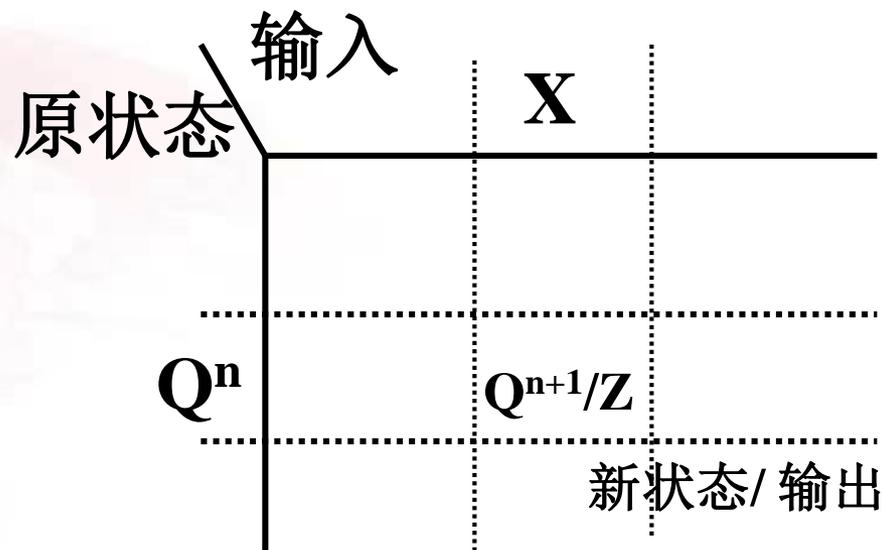


说明任何时刻的输出不仅和该时刻的外部输入信号有关，而且和该时刻的电路状态及以前的输入信号有关。



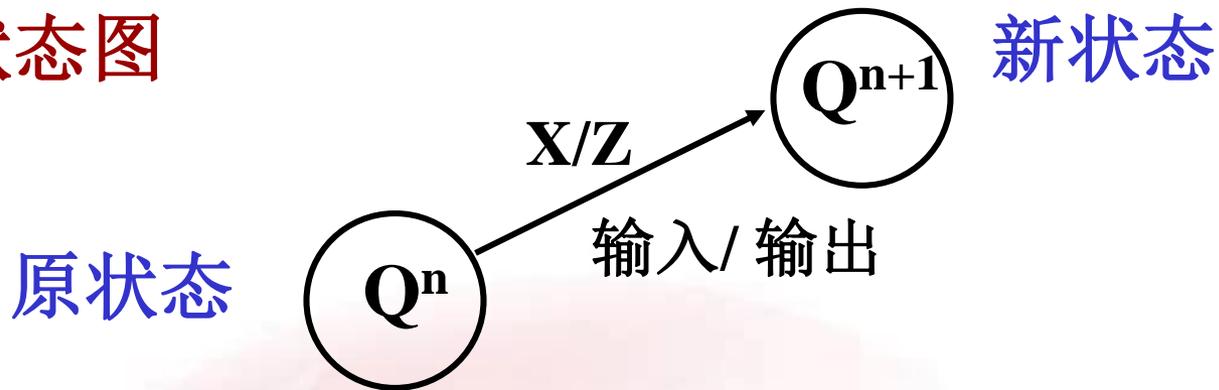
## (2) 状态表

输入	原状态	新状态	输出
$X$	$Q^n$	$Q^{n+1}$	$Z$

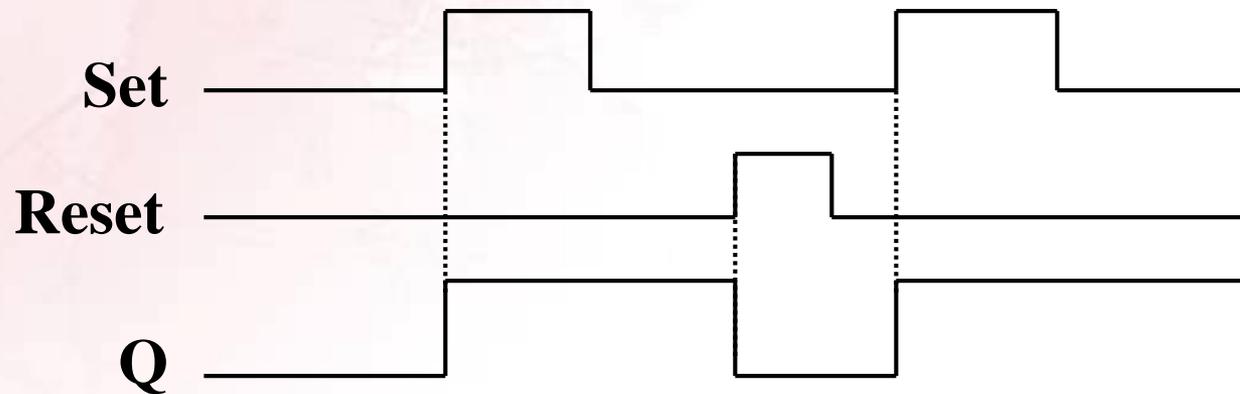




## (3) 状态图



## (4) 时序图(定时波形图)





## 4. 时序逻辑电路的分类

### (1) 按存储电路中存储单元状态改变的特点分类

同步时序电路  
异步时序电路

### (2) 按输出信号的特点分类

米里 (Mealy) 型  
摩尔 (Moore) 型

### (3) 按时序电路的逻辑功能分类

计数器  
寄存器  
移位寄存器



## 5.2 锁存器

存储电路由**存储器件**组成,能存储一位二值信号的器件称为**存储单元电路**.存储单元电路大多是**双稳态**电路.

### 双稳态电路特点:

- ①具有两个稳定状态,用0和1表示,在无外信号作用时,电路长期处于某个稳定状态,这两个稳定状态可用来表示一位二进制代码。
- ②它有一个或多个输入端,在外加信号激励下,可使电路从一个状态转换成另一个状态。



两类存储单元电路：

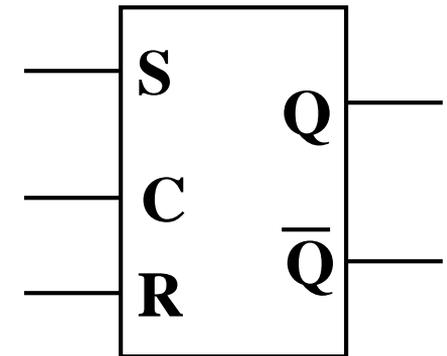
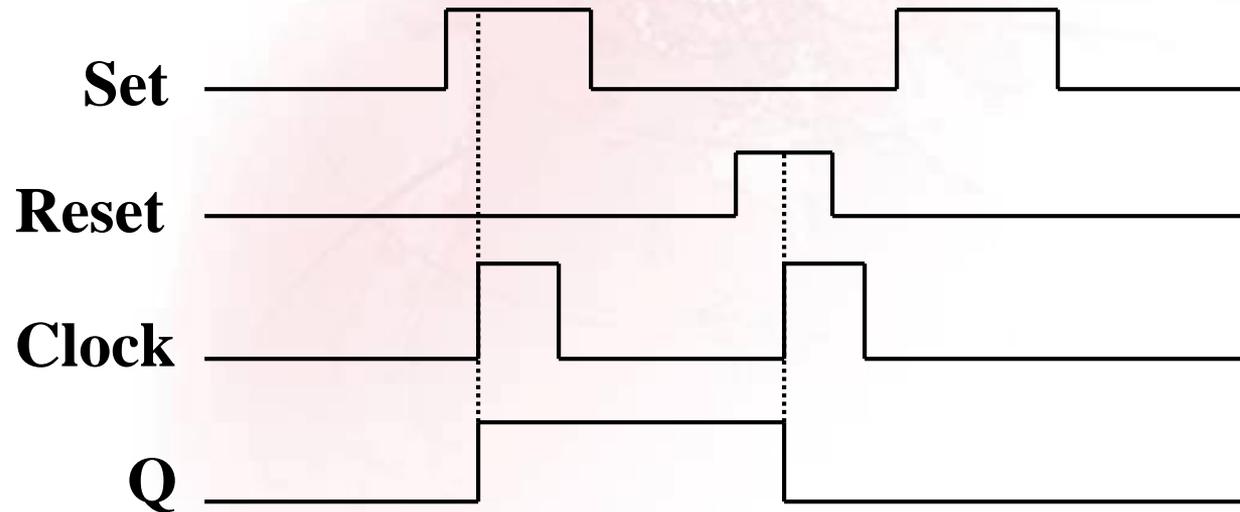
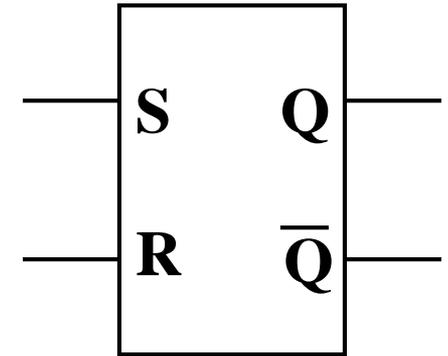
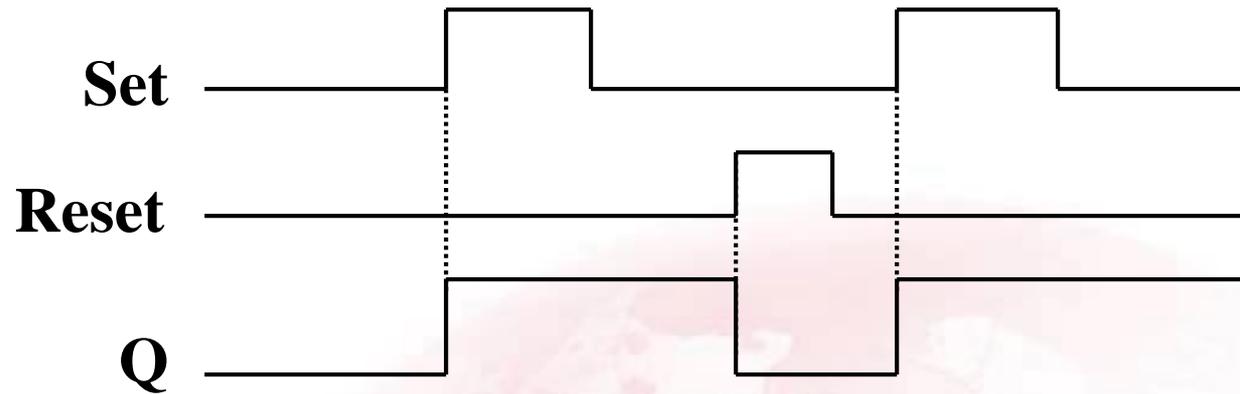
- (1) 锁存器
- (2) 触发器

锁存器：直接由激励信号控制电路状态的存储单元。

触发器：除激励信号外，还包含一个称为时钟的控制信号输入端。激励信号和时钟一起控制电路的状态。



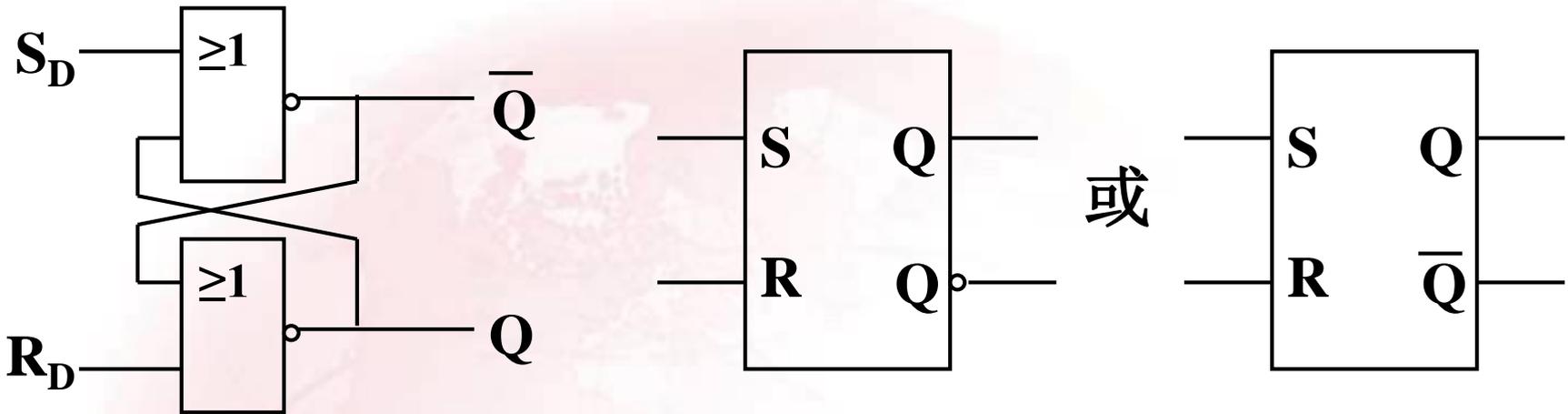
## 锁存器和触发器工作波形示意图:





## 5.2.1 普通锁存器

### 1. RS 锁存器的电路结构及逻辑符号



两个输入端(激励端):

$S_D$  : 置位端(置1端);

$R_D$  : 复位端(置0端);

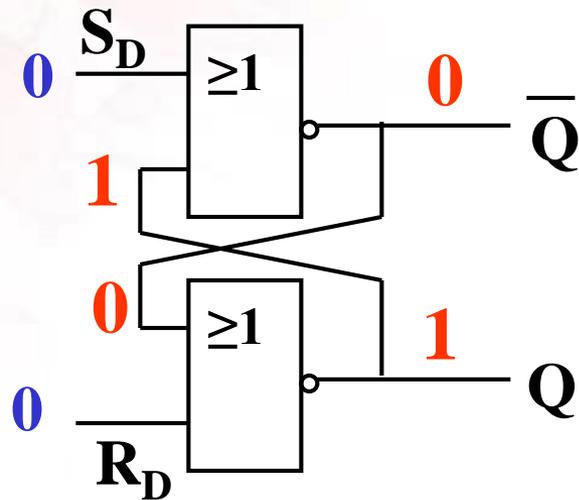
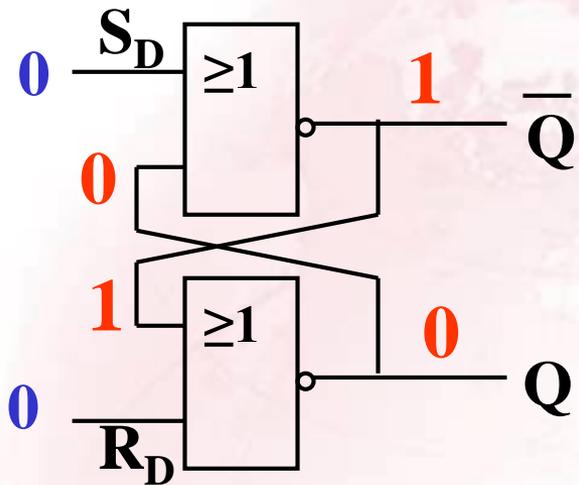
定义:  $Q=0, \bar{Q}=1$  为0状态;  $Q=1, \bar{Q}=0$  为1状态.



## 2. RS 锁存器的逻辑功能分析

设：电路的原状态表示为 $Q^n$ ，新状态表示为 $Q^{n+1}$ 。

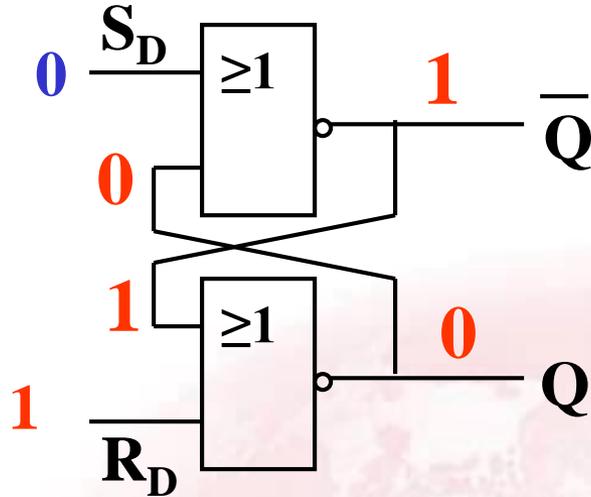
①  $S_D=0; R_D=0$  (无激励信号), 有下列两种情况:



结论:  $Q^{n+1}=Q^n$

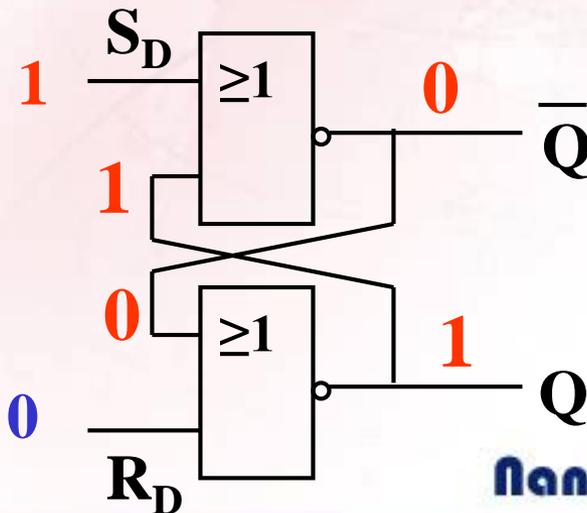


②  $S_D=0$ ;  $R_D=1$  (置0信号有效):



结论:  $Q^{n+1}=0$

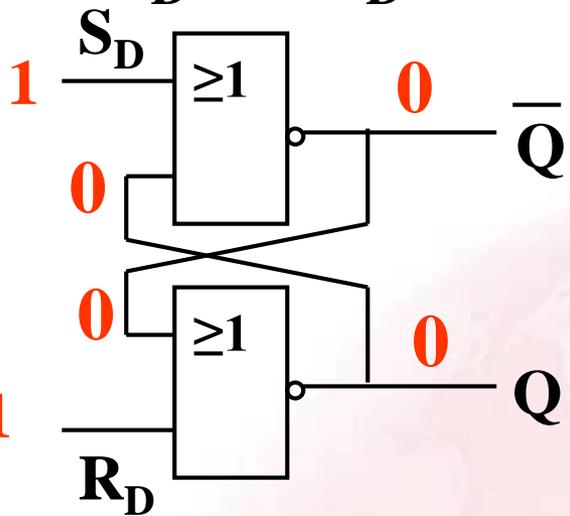
③  $S_D=1$ ;  $R_D=0$  (置1信号有效):



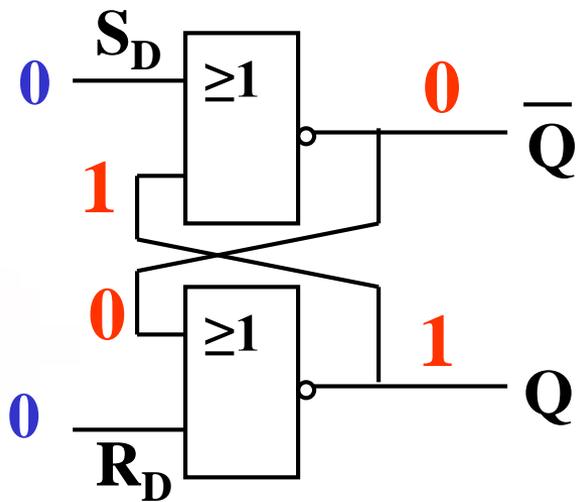
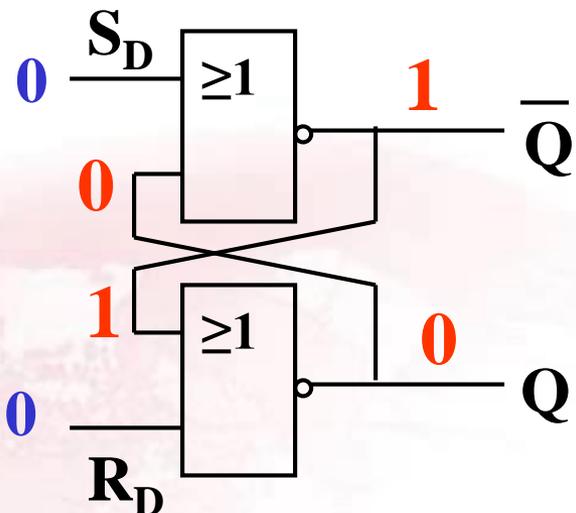
结论:  $Q^{n+1}=1$



④  $S_D=1; R_D=1$  (置0、置1同时信号有效):



作用时



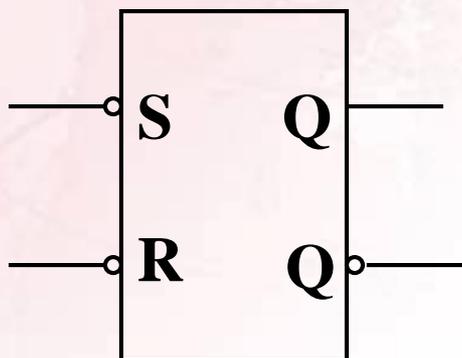
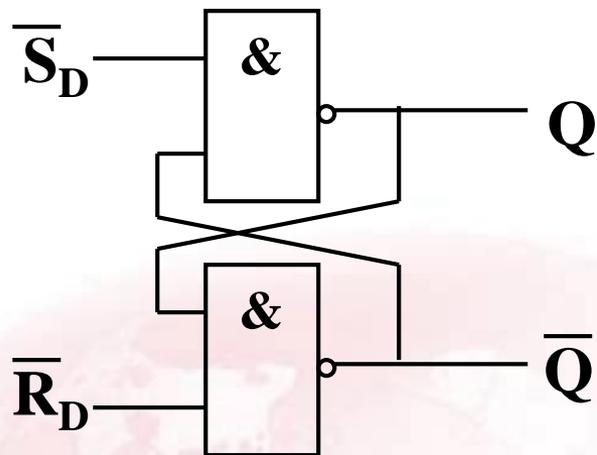
激励信号同时消失后

一般情况下， $S_D=R_D=1$ 应禁止使用。

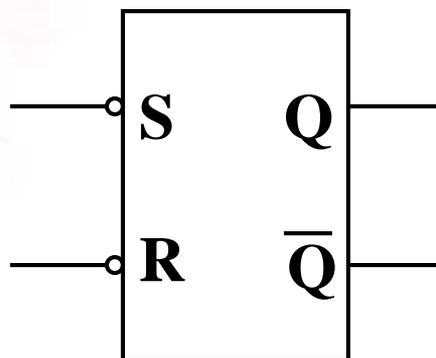
RS锁存器的约束条件： $S_D R_D=0$ 。



## 由与非门构成的RS锁存器:



或





### 3. RS锁存器的功能描述

#### (1) 特性表

$S_D$	$R_D$	$Q^n$	$Q^{n+1}$	
0	0	0	0	} 保持
0	0	1	1	
0	1	0	0	} 置0
0	1	1	0	
1	0	0	1	} 置1
1	0	1	1	
1	1	0	×	} 禁止
1	1	1	×	

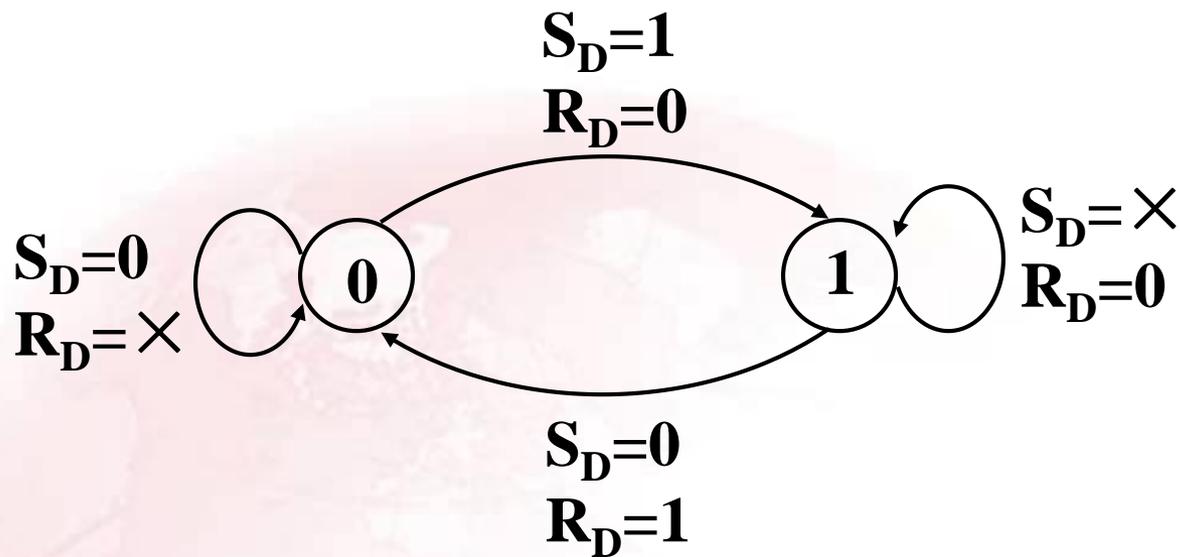
#### (2) 特性方程

$$Q^{n+1} = S_D + \bar{R}_D Q^n$$

$$S_D R_D = 0$$

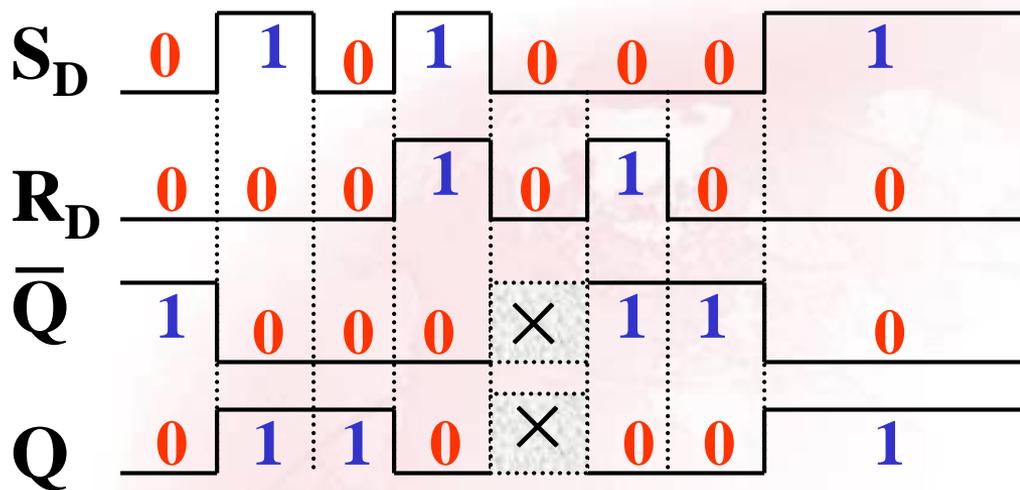


### (3) 状态图





## (4) RS锁存器工作波形图 (初态假设为0)



$S_D$	$R_D$	$Q^n$	$Q^{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	×
1	1	1	×



## 4. RS 锁存器的 VHDL 描述

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY rslatch IS
    PORT(
        nr,ns : IN std_logic;
        q,qb : BUFFER std_logic);
END rslatch;

ARCHITECTURE rtl OF rslatch IS
BEGIN
    q<=NOT(ns AND qb);
    qb<=NOT(nr AND q);
END rtl;
```

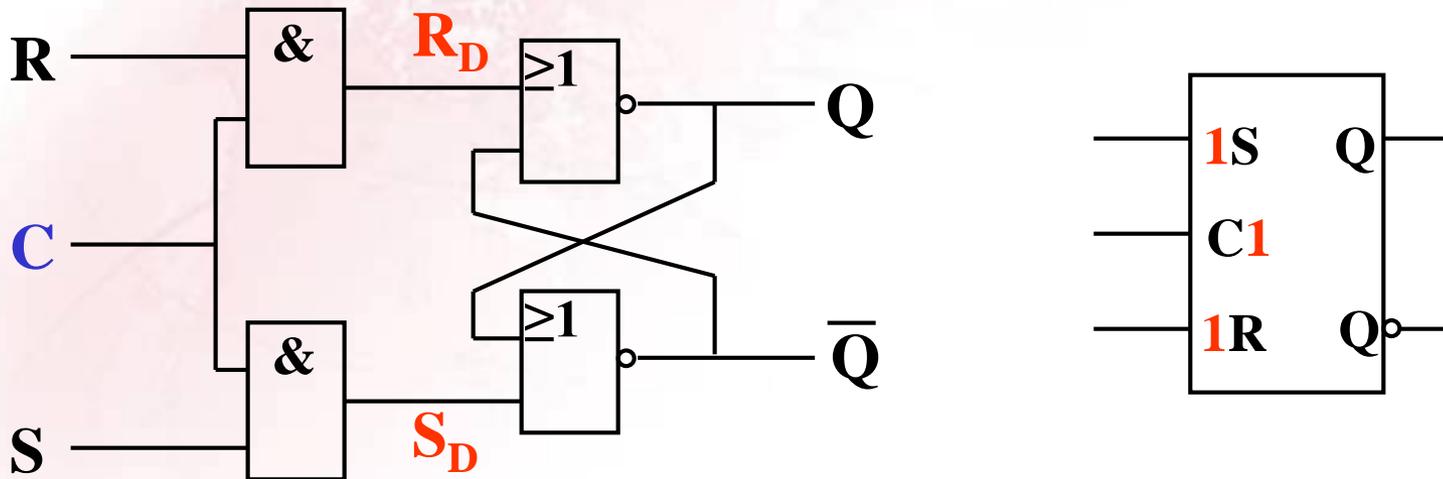


## 5.5.2 门控锁存器

### 1. 门控RS锁存器

在RS锁存器的基础上,加控制信号,使锁存器状态转换的时间,受控制信号的控制。

#### (1) 门控RS锁存器的电路结构及逻辑符号





## (2) 门控 RS 锁存器的逻辑功能分析

$$R_D = R \cdot C$$

$$S_D = S \cdot C$$

当  $C=1$  时: 门控 RS 锁存器功能和 RS 锁存器完全相同;  
当  $C=0$  时:  $R_D = S_D = 0$ , 锁存器状态保持不变.



### (3) 门控 RS 锁存器的逻辑功能描述

#### 1) 门控RS锁存器特性方程:

$$\begin{cases} Q^{n+1} = S + \bar{R}Q^n \\ SR = 0 \end{cases} \quad C=1 \text{时成立}$$

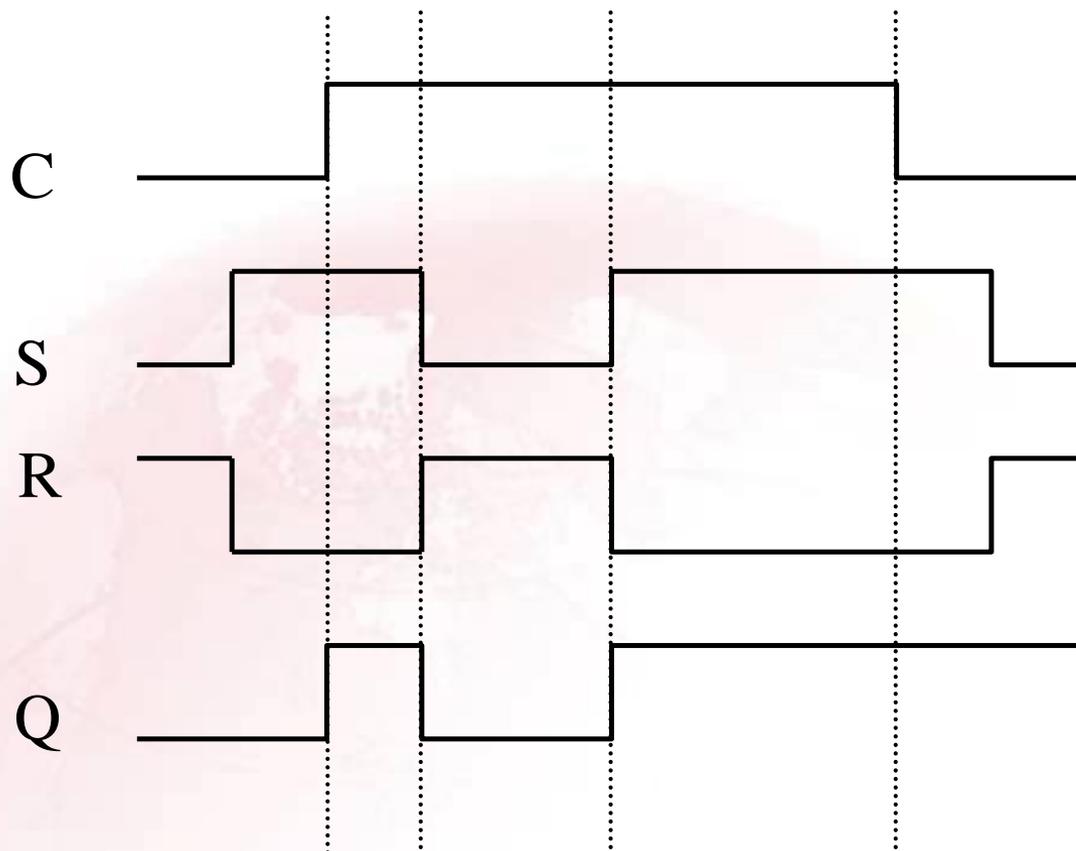


## 2) 门控RS锁存器特性表:

<b>C</b>	<b>S</b>	<b>R</b>	<b>Q<sup>n</sup></b>	<b>Q<sup>n+1</sup></b>
<b>0</b>	<b>×</b>	<b>×</b>	<b>×</b>	<b>Q<sup>n</sup></b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>×</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>×</b>



## 3) 门控RS锁存器工作波形图:

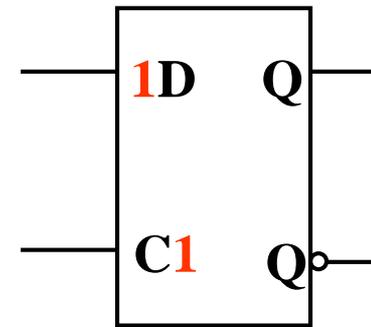
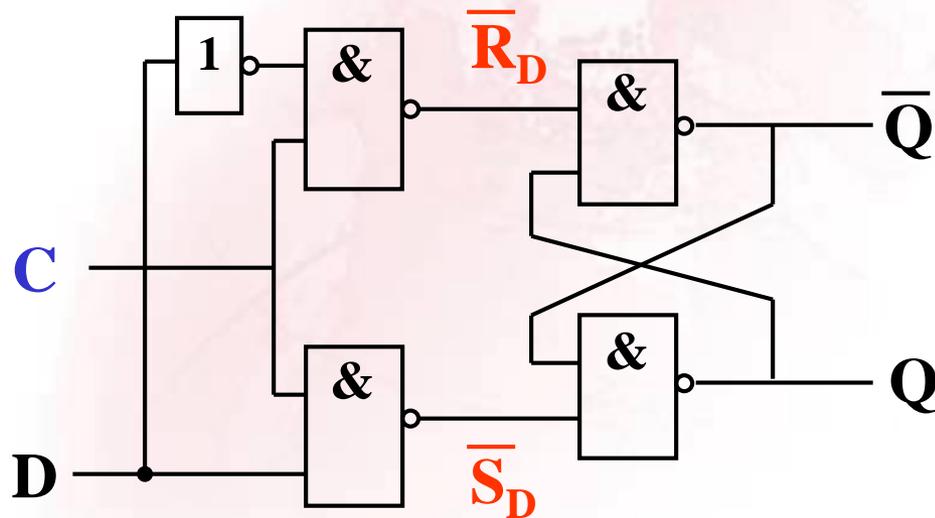




## 2. 门控 D 锁存器

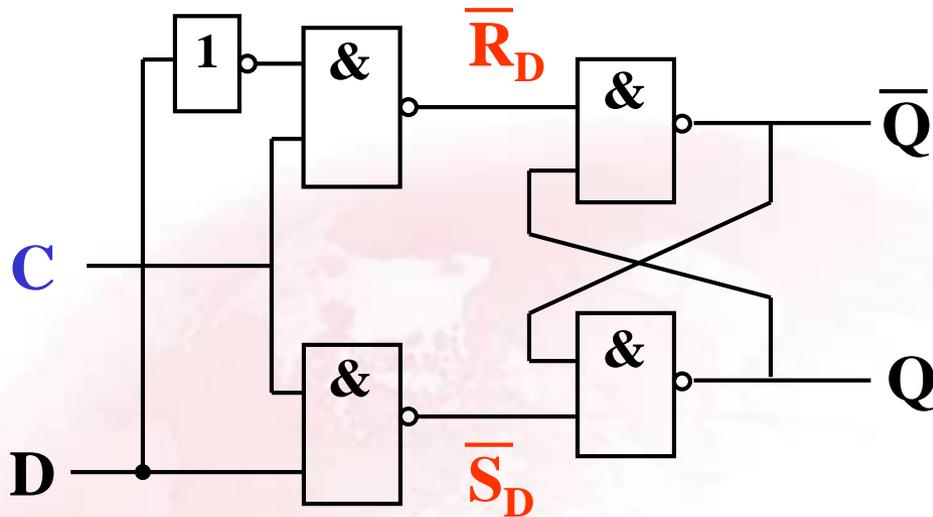
能将呈现在激励输入端的单路数据D存入交叉耦合结构的锁存器单元中。

### (1) 门控 D 锁存器的电路结构及逻辑符号





## (2) 门控 D 锁存器的逻辑功能分析



- 1) 当  $C=0$  时,  $\bar{R}_D = \bar{S}_D = 1$ , 电路处于保持状态;
- 2) 当  $C=1$  时,  $\bar{R}_D = D$ ,  $\bar{S}_D = \bar{D}$ , 电路的新状态为  $D$ 。



### (3) 门控 D 锁存器的逻辑功能描述

#### 1) D锁存器特性方程:

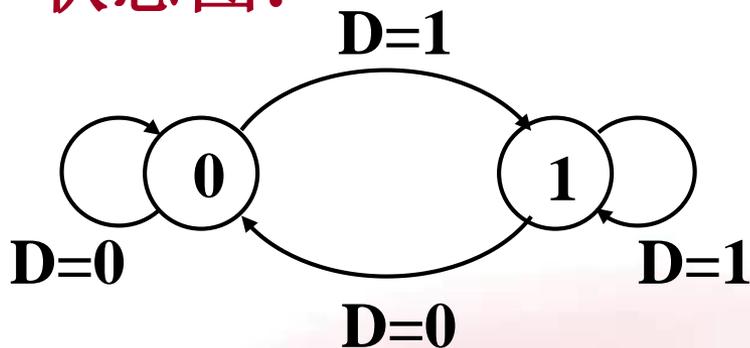
$$Q^{n+1}=D$$

#### 2) D锁存器特性表:

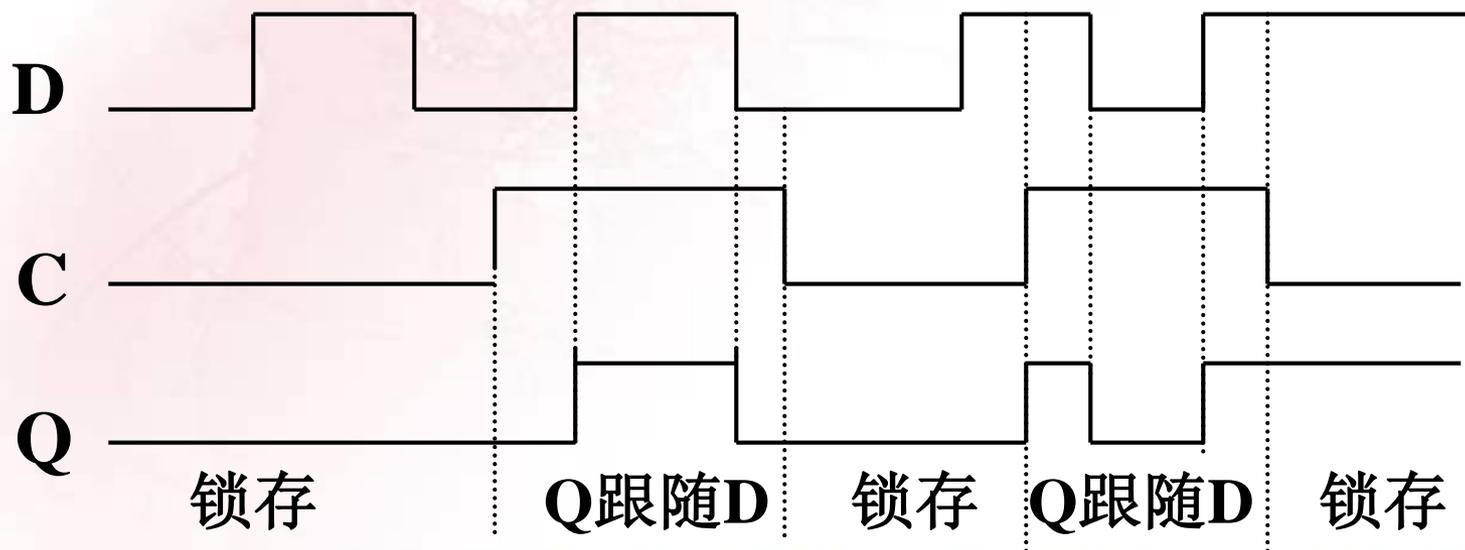
D	$Q^n$	$Q^{n+1}$
0	0	0
0	1	0
1	0	1
1	1	1



## 3) 状态图:



## 4) D锁存器工作波形图: (假设初态为0)





## (4) 门控 D 锁存器的 VHDL 描述

```
LIBRARY ieee;  
USE ieee.std_logic_1164.ALL;  
  
ENTITY dlatch IS  
  PORT( c,d : IN std_logic;  
         q,qb : OUT std_logic);  
END dlatch;  
  
ARCHITECTURE rtl OF dlatch IS  
  SIGNAL q_temp,qb_temp:std_logic;  
BEGIN
```



```
PROCESS(c,d)  
BEGIN  
  IF(c='1') THEN  
    q_temp<=d;  
    qb_temp<=NOT(d);  
  END IF;  
END PROCESS;  
q<=q_temp;  
qb<=qb_temp;  
END rtl;
```



表5.3 部分常用集成锁存器

型 号	集成器件数	功能说明
<b>7475</b>	<b>4</b>	门控 <b>D</b> 锁存器
<b>74116</b>	<b>2</b>	<b>4</b> 位门控锁存器 有 $\overline{RD}$ 和双使能控制端
<b>74279</b>	<b>4</b>	低电平有效 <b>RS</b> 锁存器
<b>74LS373</b>	<b>1</b>	<b>8</b> 位 <b>D</b> 锁存器，三态输出
<b>74LS375</b>	<b>4</b>	门控 <b>D</b> 锁存器



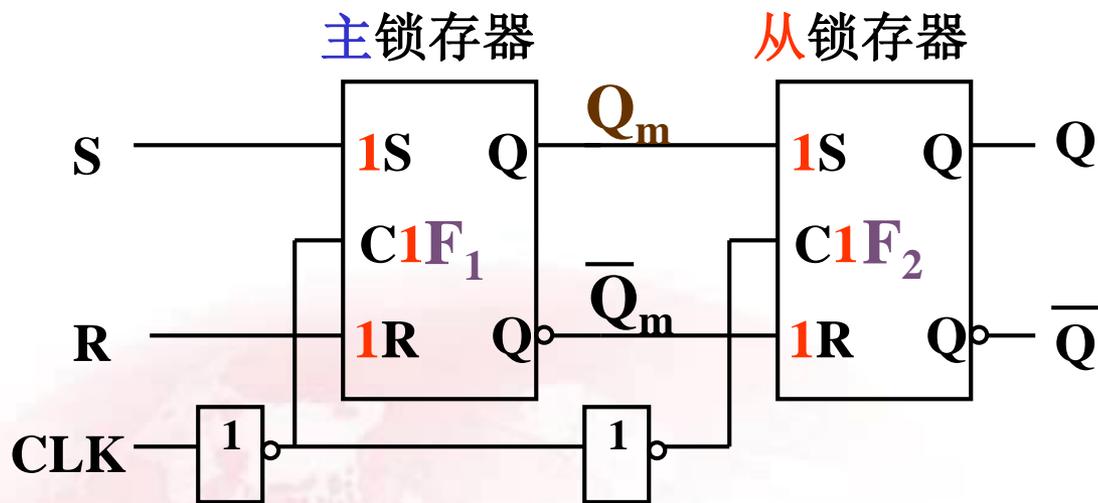
## 5.3 触发器

利用一个称为“**时钟**”的特殊**定时控制信号**去限制存储单元状态的**改变时间**, 具有这种特点的存储单元电路称为**触发器**.





## (2) 主从 RS 触发器的工作原理



- 1) 在 $CLK=0$ 时,主锁存器 $F_1$ 的控制门打开,处于**工作状态**,主锁存器按 $S$ 、 $R$ 的值改变**中间状态** $Q_m$ ;从锁存器 $F_2$ 的控制门关闭,处于**保持状态**;
- 2) 在 $CLK=1$ 时,主锁存器 $F_1$ 的控制门关闭,进入**保持状态**;从锁存器 $F_2$ 的控制门打开,处于**工作状态**,电路根据 $Q_m$ 的状态改变**输出状态**;

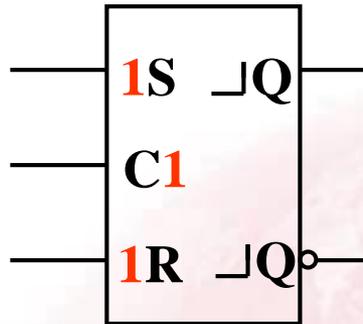


## 主从 RS 触发器的电路特点:

- 1) CLK脉冲不论在低电平或高电平期间,电路的输出状态最多只**改变一次**;(常把控制信号有效期间,输出状态发生**多次变化**的现象称为**空翻**)
- 2) 将主从RS触发器用于时序电路中,不会因**不稳定**而产生**振荡**.



主从 RS 触发器的电路符号：



“┐”称为延迟符号,表示该触发器在 $CLK=0$ 时接收R、S的数据,而在CLK的上升沿时,输出改变状态



### (3) 主从 RS 触发器的逻辑功能描述

#### 1) 主从RS触发器的特性表

主从RS触发器的特性表和特性方程和RS锁存器基本相同,只是在列特性表时,要加上CLK脉冲标志.

CLK	$S_D$	$R_D$	$Q^n$	$Q^{n+1}$
×	×	×	×	$Q^n$
⌋	0	0	0	0
⌋	0	0	1	1
⌋	0	1	0	0
⌋	0	1	1	0
⌋	1	0	0	1
⌋	1	0	1	1
⌋	1	1	0	×
⌋	1	1	1	×

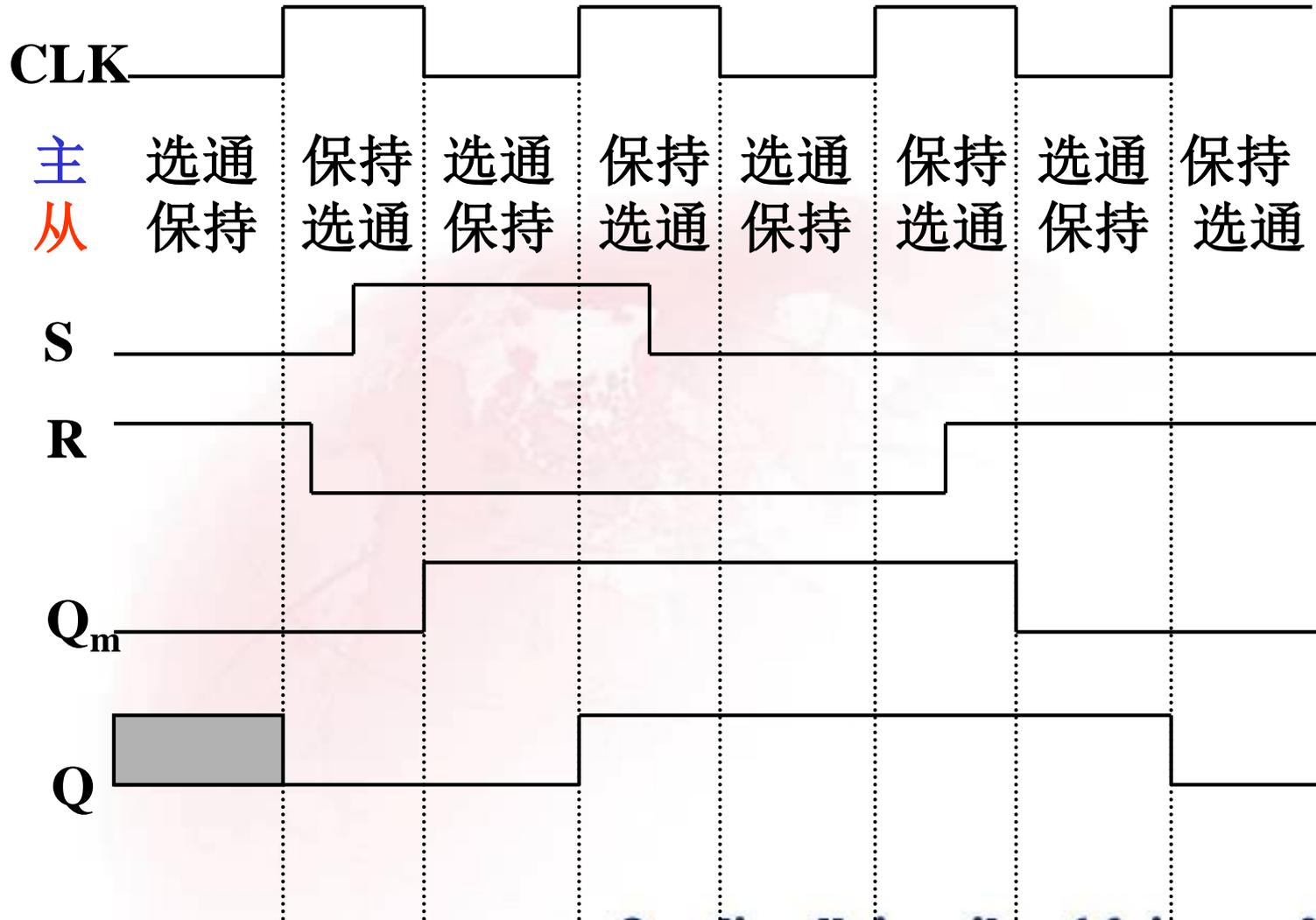


## 2) 主从RS触发器的特性方程

$$\begin{cases} Q^{n+1} = S + \bar{R}Q^n \\ SR = 0 \end{cases}$$

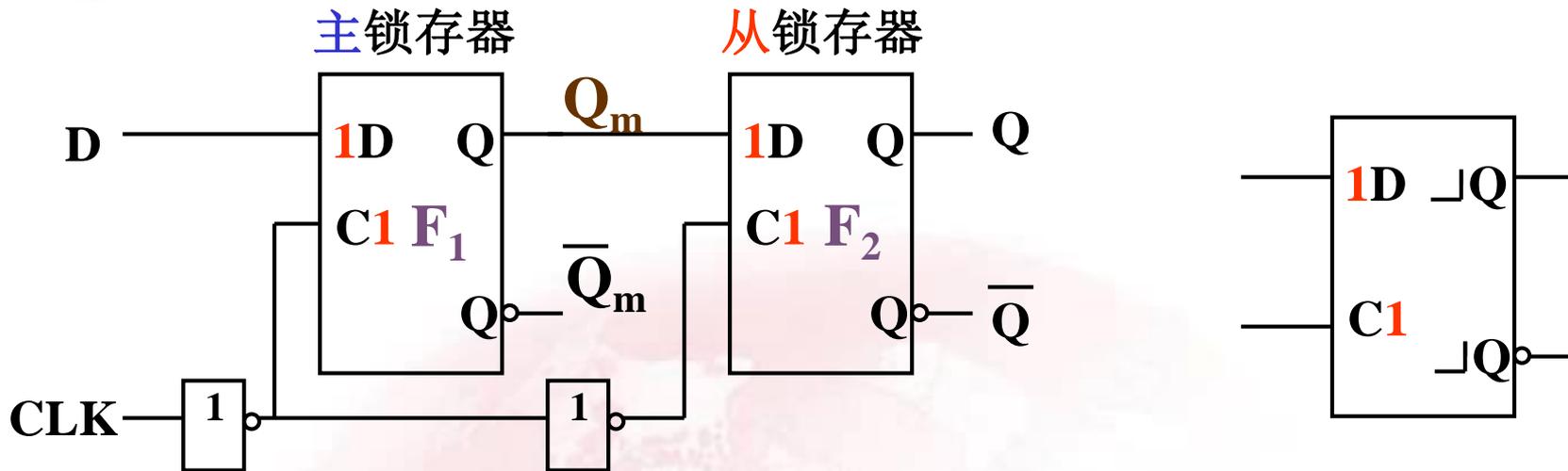


## 3) 主从RS 触发器的定时波形





## 2. 主从D触发器



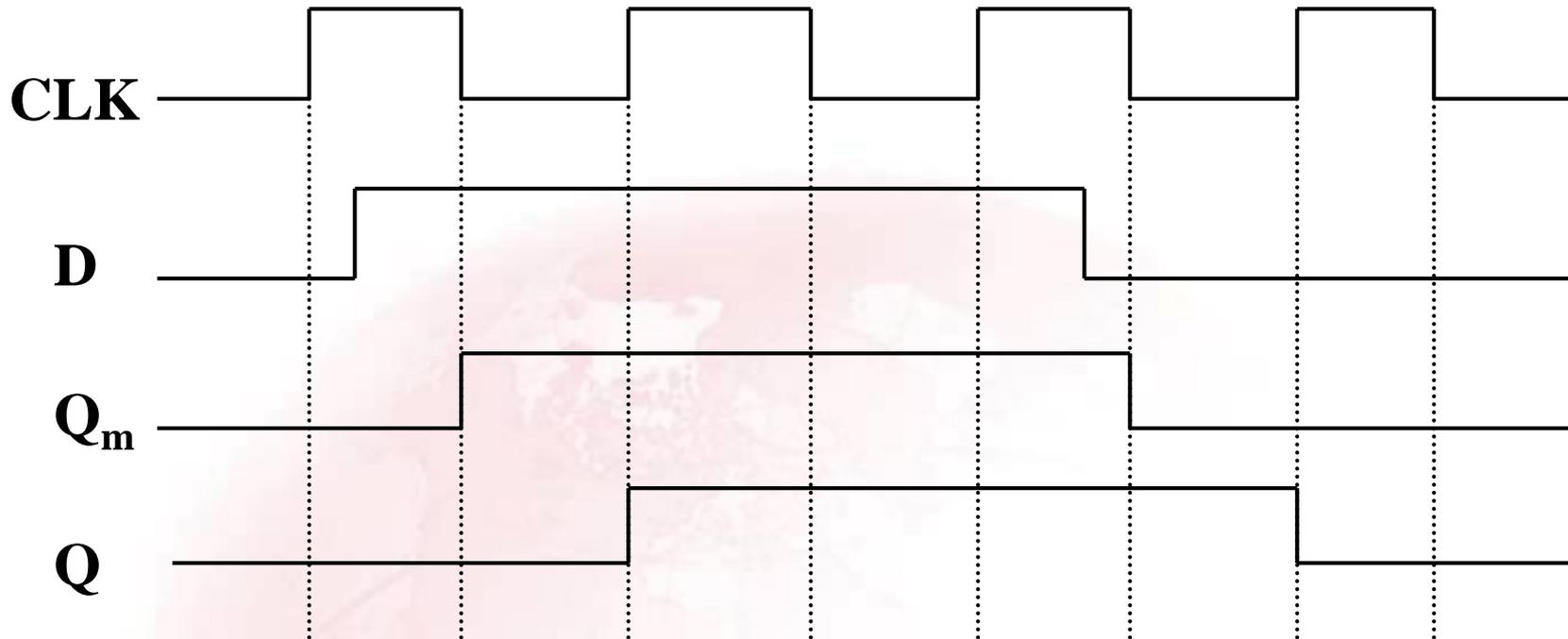
工作原理:

- (1) 当 $CLK=0$ 时,主锁存器被选通, $Q_m=D$ ,从锁存器保持原态;
- (2) 当 $CLK=1$ 时,主锁存器保持原态,从锁存器被选通, $Q=Q_m$ ;

特性方程:  $Q^{n+1}=D$



## 定时波形图

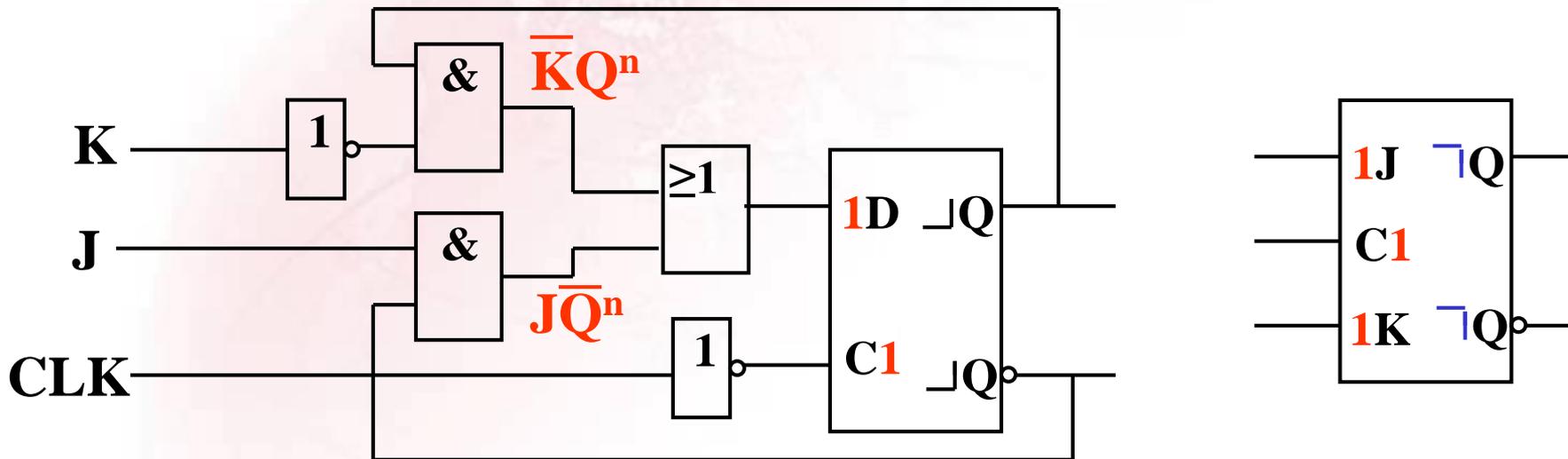




## 3. 主从JK 触发器

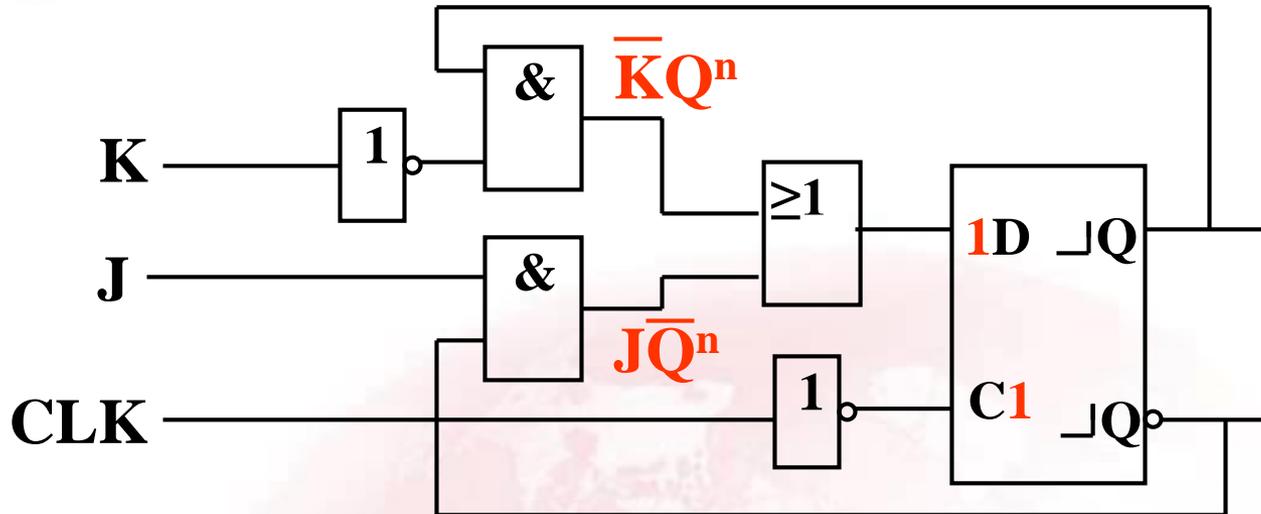
为去除主从RS触发器的约束条件: $RS=0$ ,设计出主从JK触发器.

### (1) 主从JK触发器的一种结构和逻辑符号





## (2) 主从JK触发器的特点



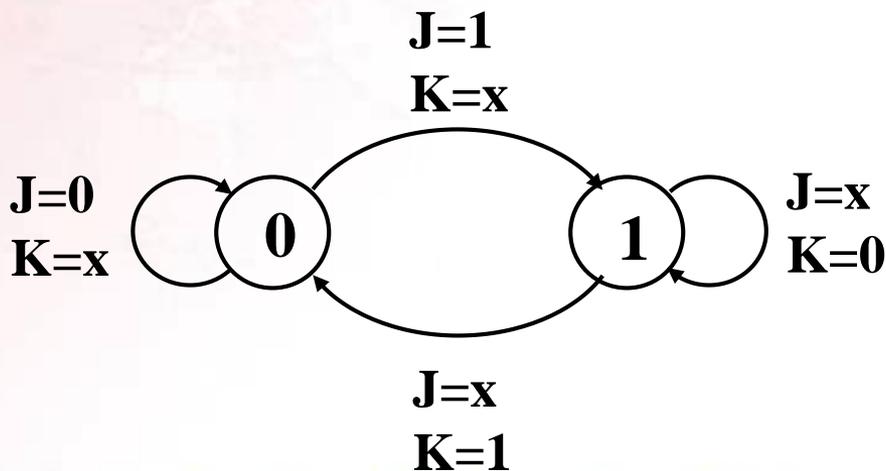
- 1) 电路以D触发器为核心,故不存在约束条件;
- 2)  $D=J\bar{Q}^n+\bar{K}Q^n$ , 所以,  $Q^{n+1}=D=J\bar{Q}^n+\bar{K}Q^n$
- 3) 由电路可见,  $CLK$ 是经一个非门送入D触发器,所以这种结构的JK触发器为  $CLK$  下降沿到达时改变状态.



(3) 根据特性方程  $Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$ , 容易求得特性表:

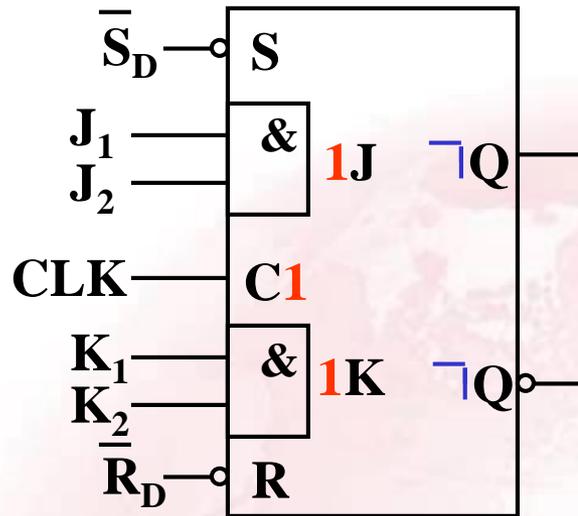
CLK	J	K	$Q^n$	$Q^{n+1}$	CLK	J	K	$Q^n$	$Q^{n+1}$
×	×	×	×	$Q^n$	⌋	1	0	0	1 } 置
⌋	0	0	0	0 } 保持	⌋	1	0	1	1 } “1”
⌋	0	0	1	1 } 保持	⌋	1	1	0	1 } 翻
⌋	0	1	0	0 } 置	⌋	1	1	1	0 } 转
⌋	0	1	1	0 } “0”					

## (4) 状态图





## (5) 带异步清零、置1端并具有多驱动输入的JK触发器。



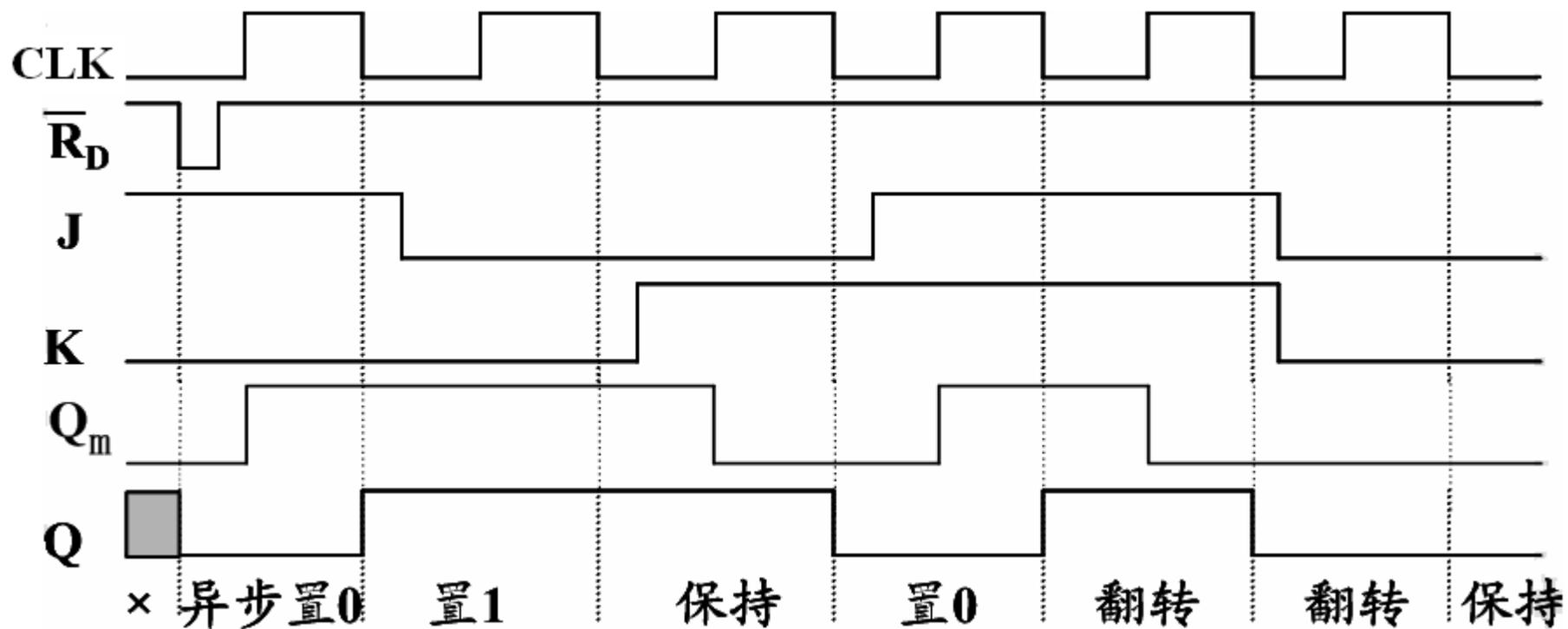
$\bar{S}_D$  : 异步置1端;  
 $\bar{R}_D$  : 异步清零端。

$$J = J_1 \cdot J_2$$

$$K = K_1 \cdot K_2$$



## (6) 主从JK触发器定时波形

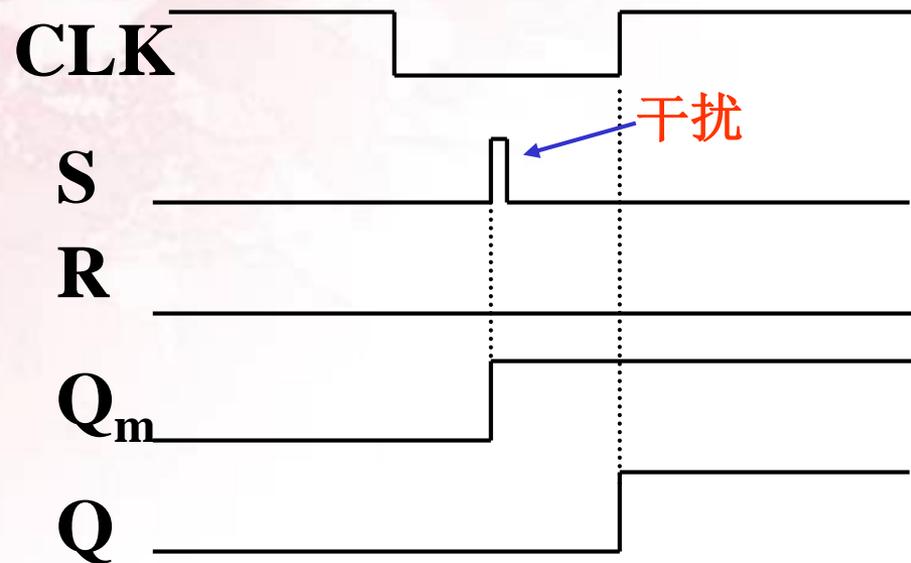
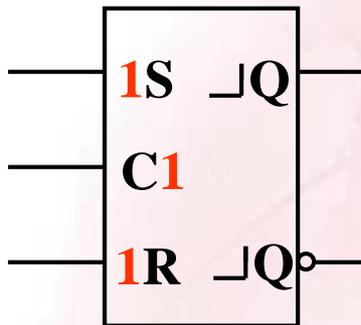




## 4. 主从触发器的缺陷

主从触发器抗干扰能力不强

上升沿翻转的主从RS触发器





## 5.3.2 边沿触发器

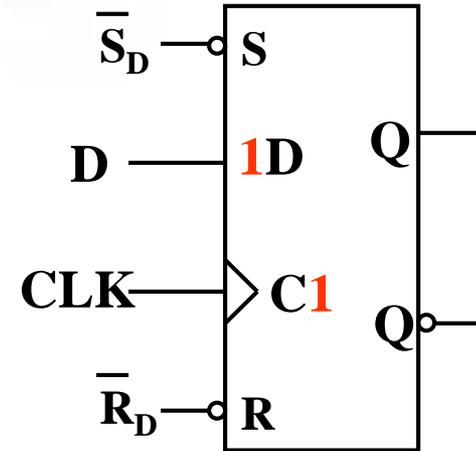
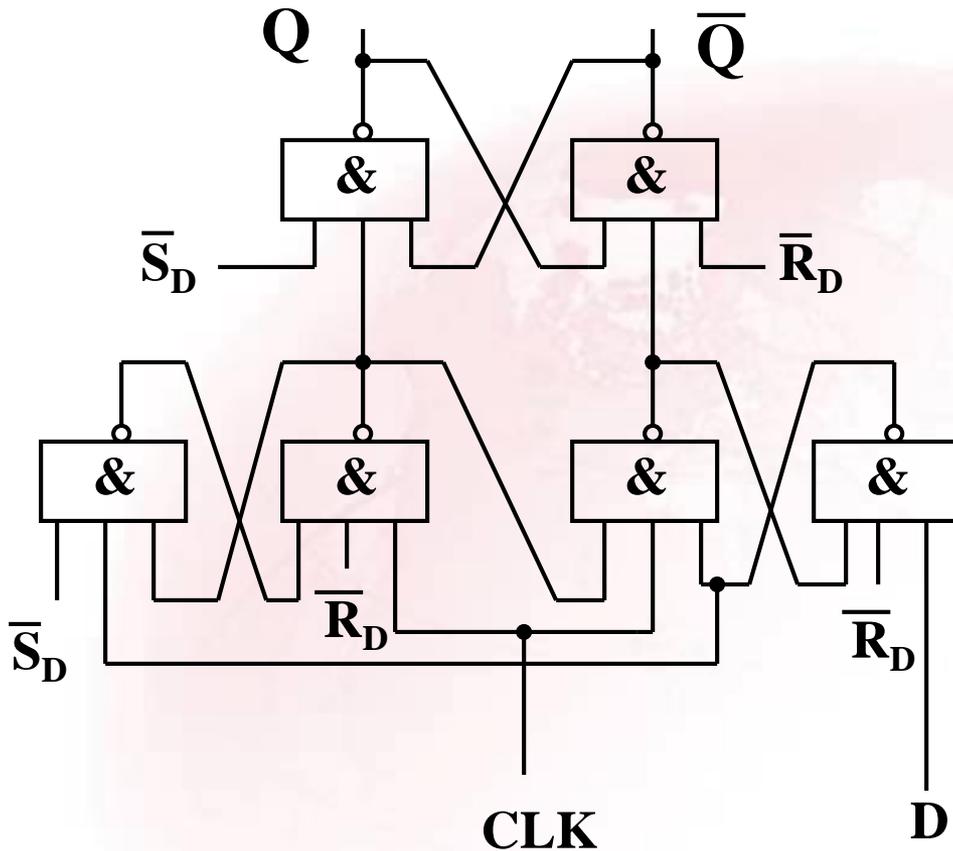
边沿触发器的特点：

在时钟为稳定的0或1期间，输入信号都不能进入触发器，触发器的新状态**仅**决定于时钟脉冲有效边沿**到达前一瞬间以及到达后极短一段时间内**的输入信号。边沿触发器具有较好的**抗干扰性能**。



## 1. 维持阻塞D触发器

### (1) 电路结构与逻辑符号





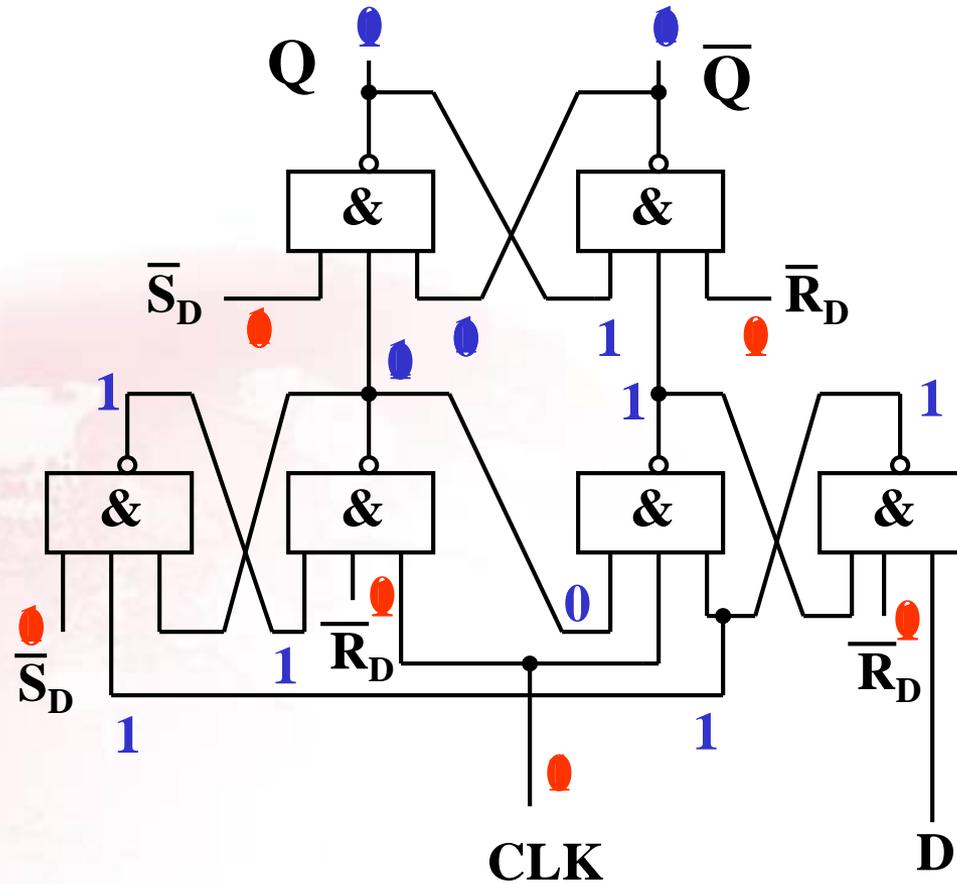
## (2) 工作原理

① 异步清零

② 异步置1

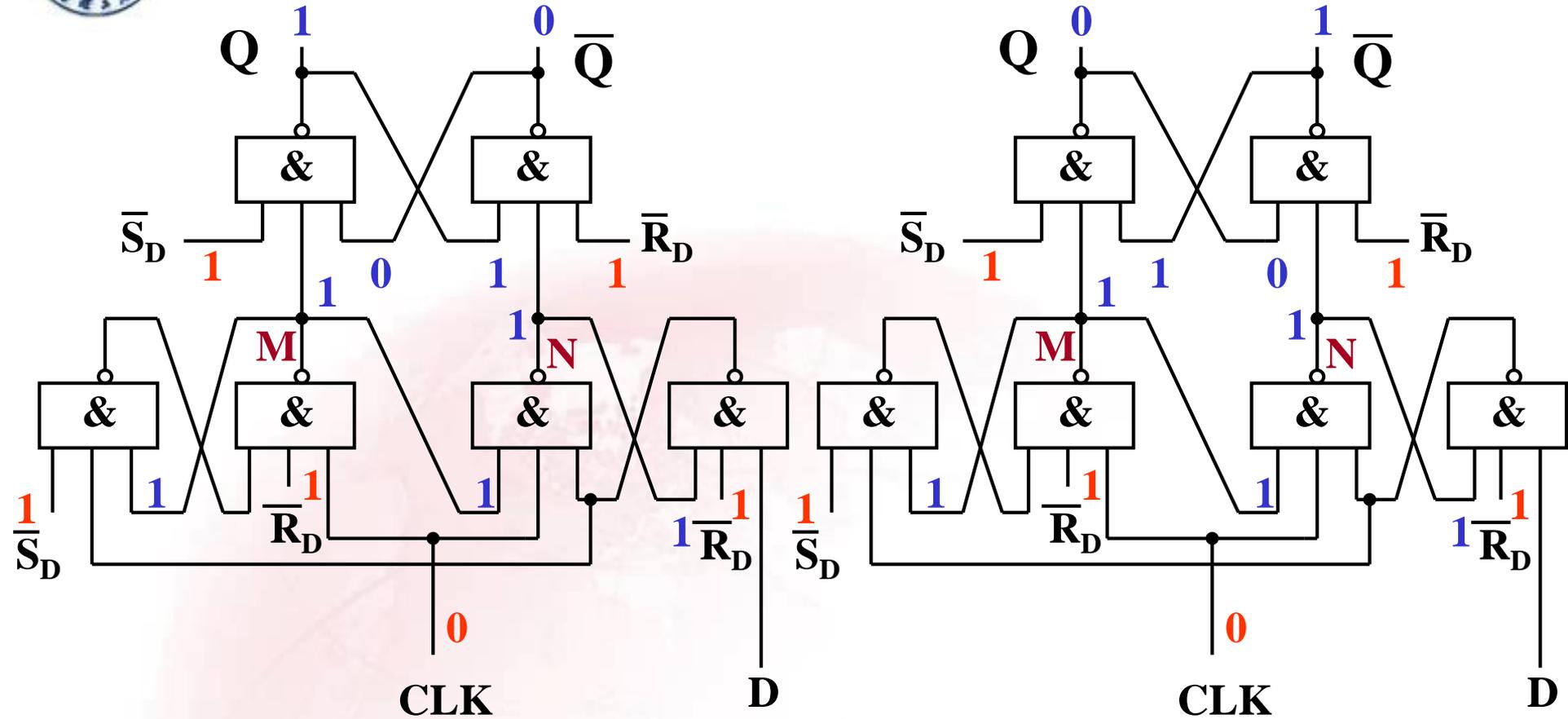
a. CLK=0

b. CLK=1





③ 如  $\bar{S}_D = \bar{R}_D = 1$ , 并  $CLK = 0$ : 由于  $M = N = 1$ , 所以输出  $Q$  保持不变



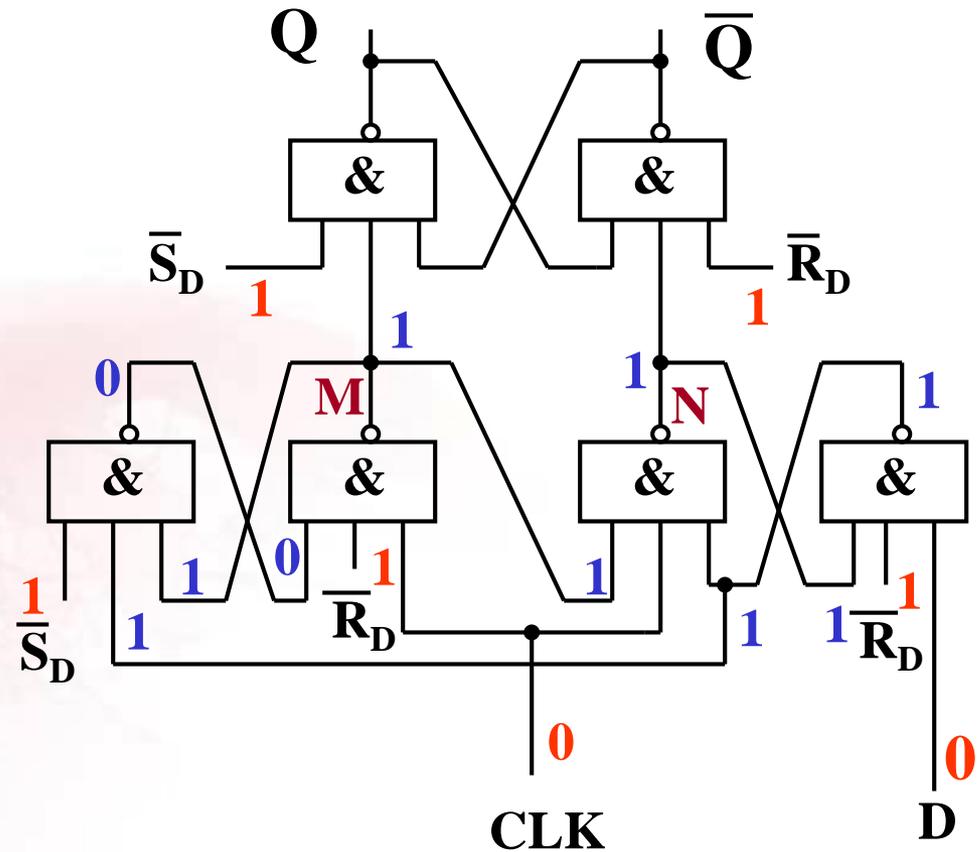
$Q^n=1$ 的情况

$Q^n=0$ 的情况

注意：在该时刻， $D$ 的改变不会使输出状态变化。



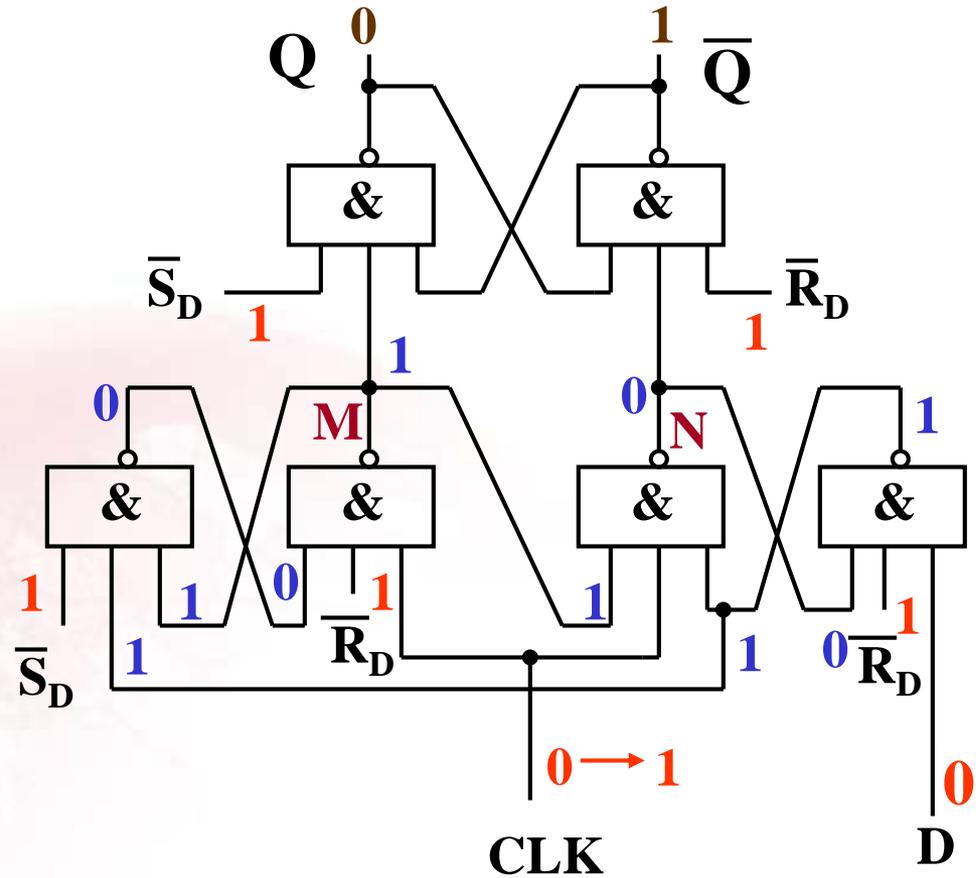
④ 在 $CLK=1$ 前一瞬间  
( $CLK=0$ )，如加入信  
号 $D=0$ ，则有：



由于 $M=N=1$ ，所以输出 $Q$ 保持不变。



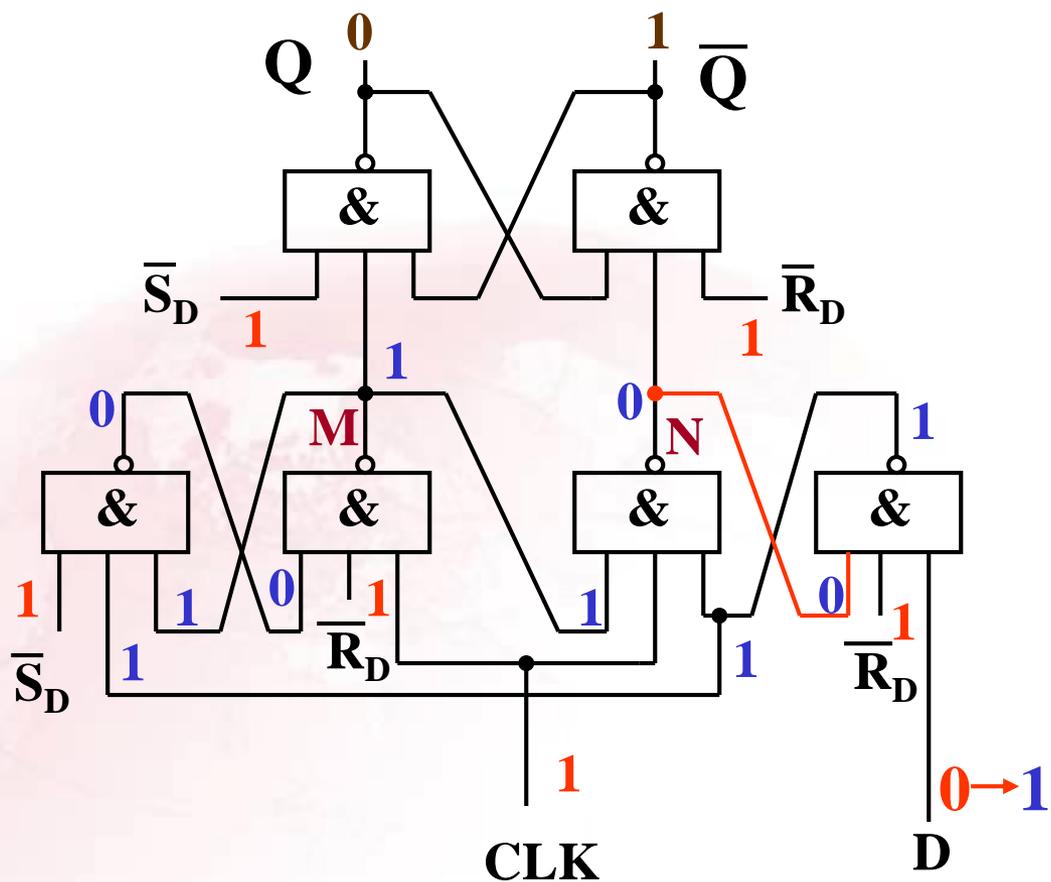
⑤ CLK由0变为1，在这一短时间内， $D=0$ 保持不变，则有：



$$Q^{n+1} = D = 0$$



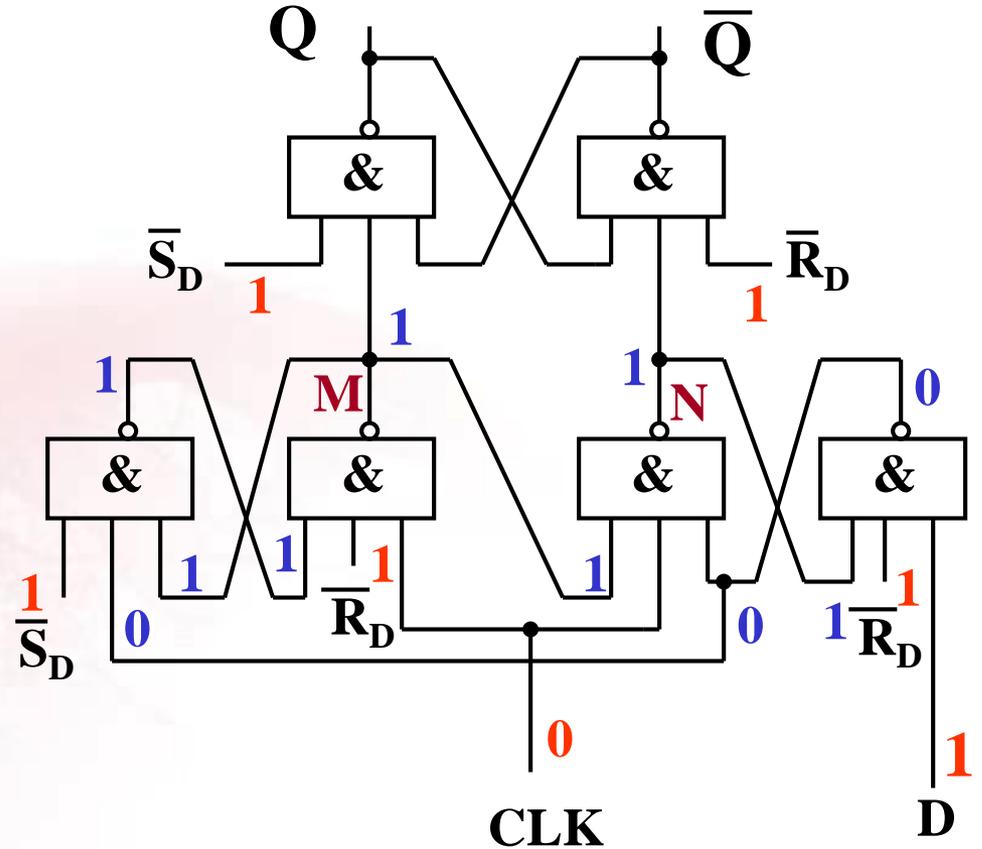
⑥ 在上面的情况下，如D发生变化，即由0变为1



由于图中红线的的作用，电路输出状态保持不变。



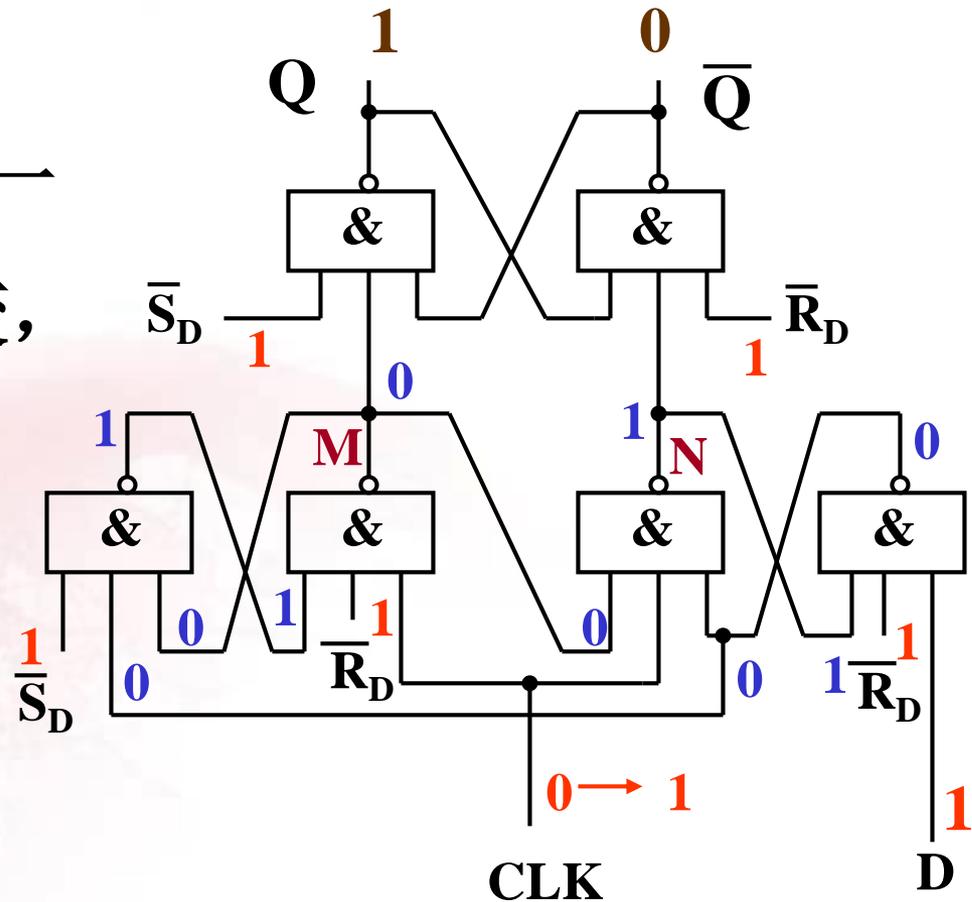
⑦ 在 $CLK=1$ 前一瞬间 ( $CLK=0$ )，如加入信号 $D=1$ ，则有：



由于 $M=N=1$ ，所以输出 $Q$ 保持不变。



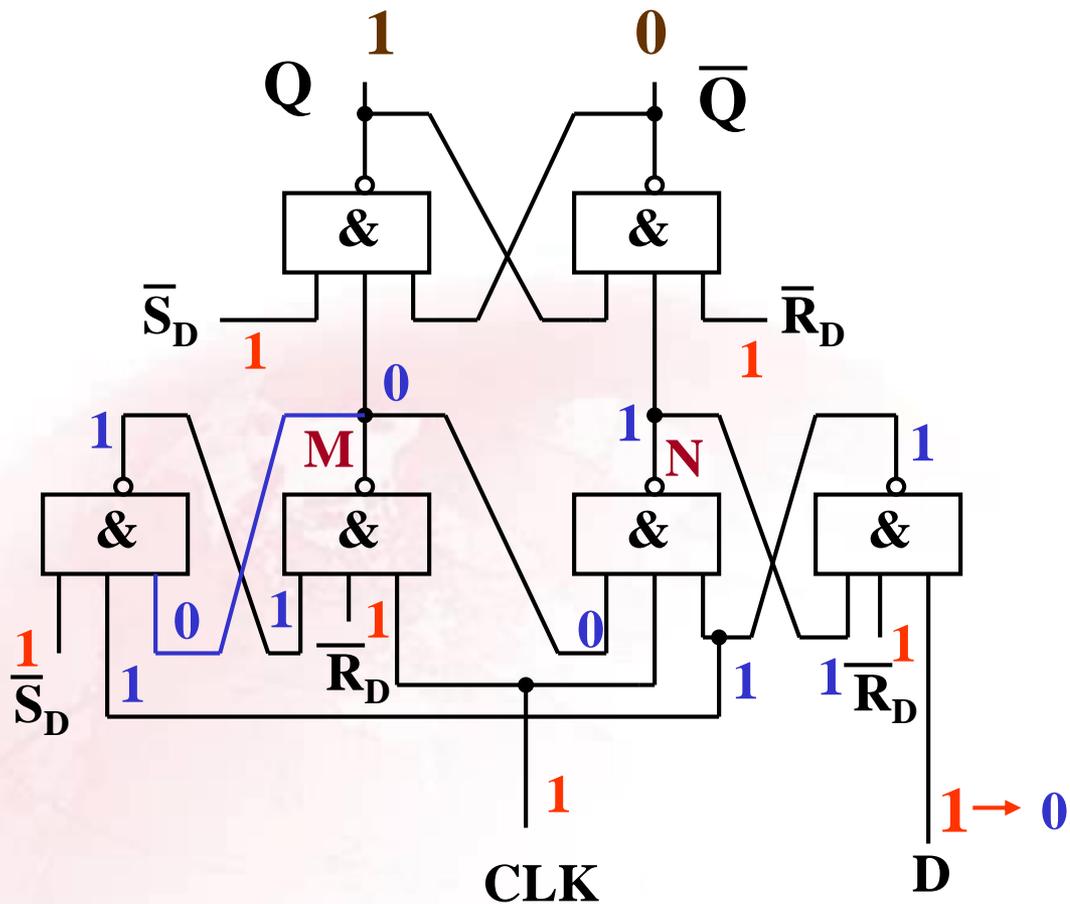
⑧ CLK由0变为1，在这一短时间内， $D=1$ 保持不变，则有：



$$Q^{n+1} = D = 1$$



⑨ 在上面的情况下，如D发生变化，即由1变为0



由于和M端连接的蓝线的作用，使输出保持不变

$$Q^{n+1} = D = 1$$

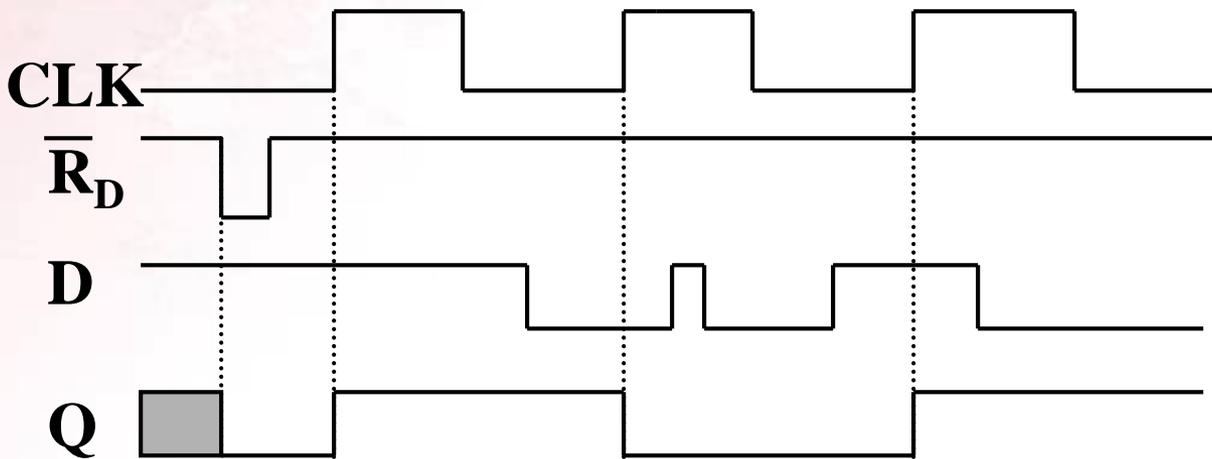


## (3) 维持阻塞D触发器特性表和工作波形图

CLK	$\overline{S}_D$	$\overline{R}_D$	D	$Q^n$	$Q^{n+1}$
×	0	1	×	×	1
×	1	0	×	×	0
↑	1	1	0	0	0
↑	1	1	0	1	0
↑	1	1	1	0	1
↑	1	1	1	1	1

↑表示上升沿触发.

当 $\overline{S}_D=1$ 时波形图:





## (4) 正边沿 D 触发器的 VHDL 描述

```
LIBRARY ieee;  
USE ieee.std_logic_1164.ALL;  
ENTITY dff1 IS  
  PORT(  
    d,clk,rd,sd : IN std_logic;  
    q,qb : OUT std_logic);  
END dff1;  
ARCHITECTURE rtl OF dff1 IS  
  SIGNAL q_temp,qb_temp:std_logic;  
BEGIN  
  PROCESS(clk,rd,sd)  
  BEGIN
```

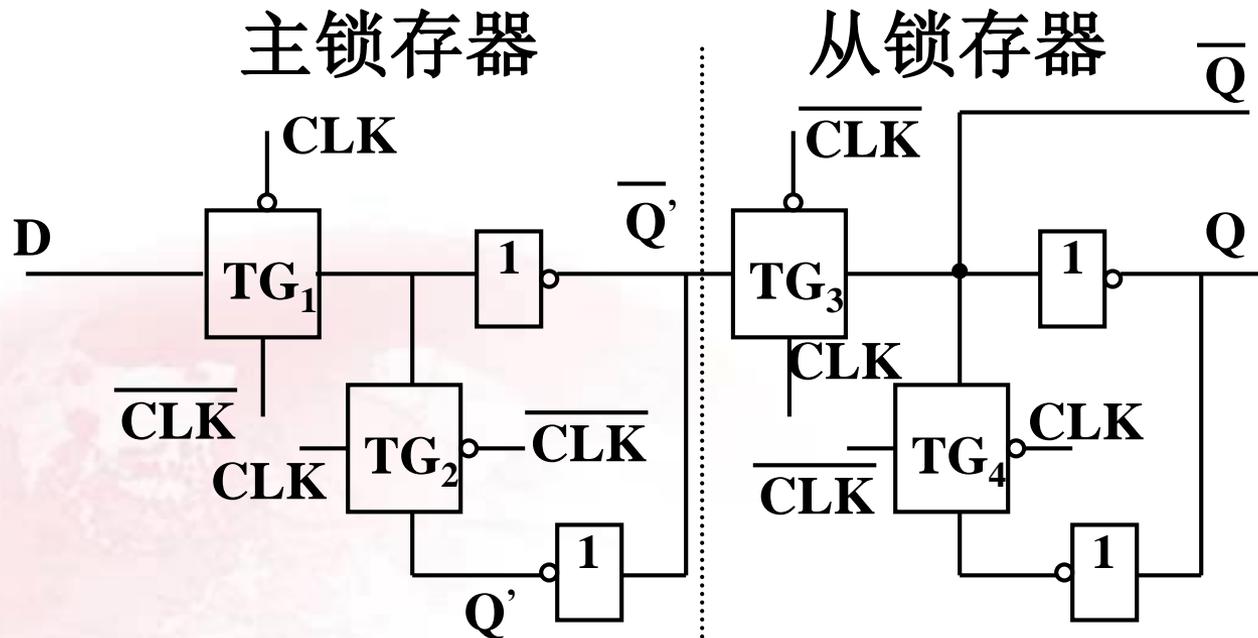


```
IF(rd='0' AND sd='1') THEN  
    q_temp<='0';  
    qb_temp<='1';  
ELSIF (rd='1' AND sd='0') THEN  
    q_temp<='1';  
    qb_temp<='0';  
  
    ELSIF (clk'event AND clk='1') THEN  
    q_temp<=d;  
    qb_temp<=NOT(d);  
END IF;  
END PROCESS;  
q<=q_temp;  
qb<=qb_temp;  
END rtl;
```

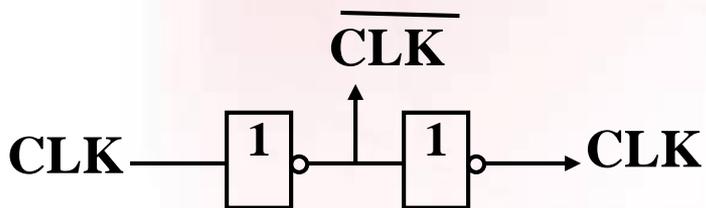


## 2.CMOS边沿D触发器

CMOS边沿D触发器由CMOS传输门构成，属主从结构，但具有边沿触发器的特点。



### (1)电路结构



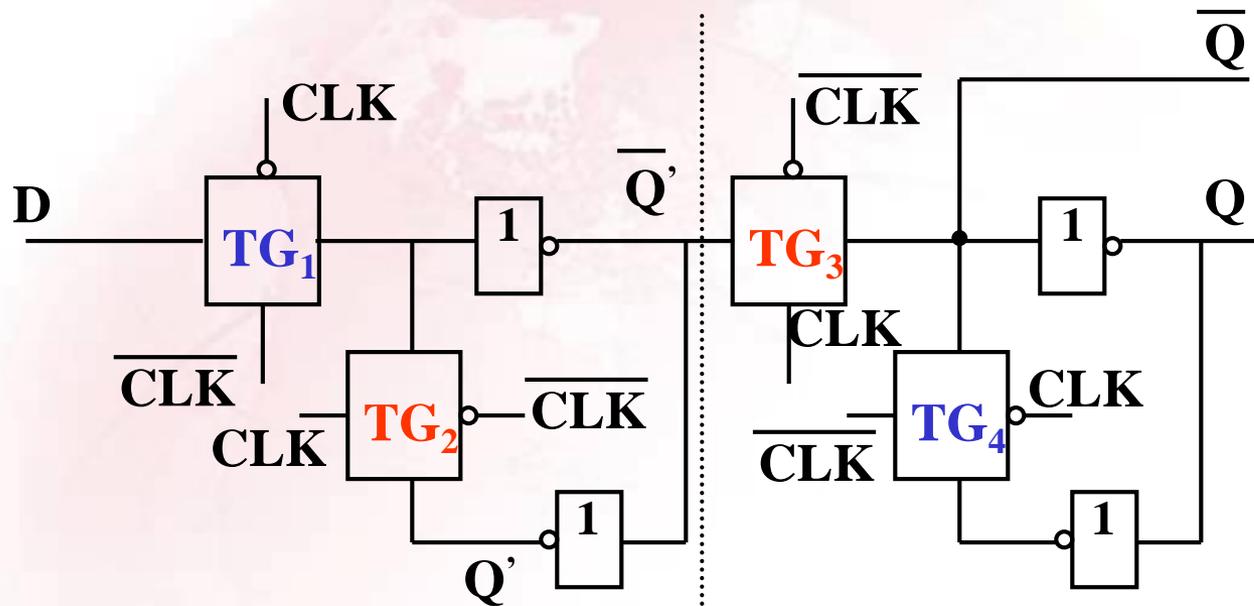
由图可知，当CLK=0时，TG<sub>1</sub>和TG<sub>4</sub>导通，TG<sub>2</sub>和TG<sub>3</sub>截止；当CLK=1时，TG<sub>1</sub>和TG<sub>4</sub>截止，TG<sub>2</sub>和TG<sub>3</sub>导通。



## (2) 工作原理

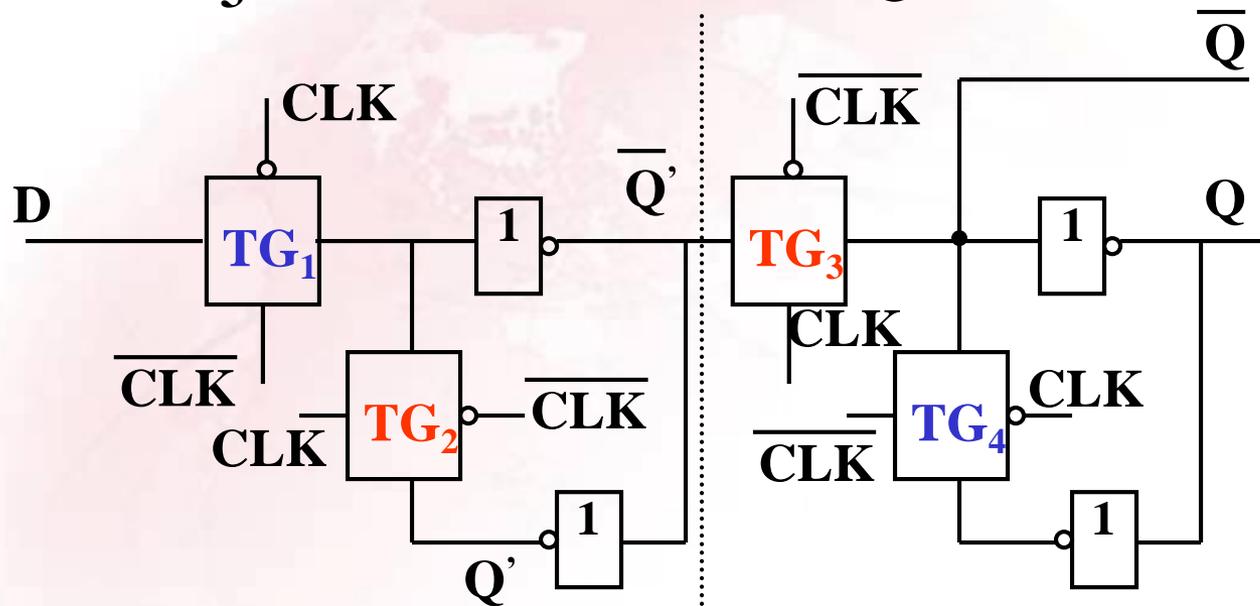
① 当 $\text{CLK}=0$ 时,  $\text{TG}_1$ 和 $\text{TG}_4$ 导通,  $\text{TG}_2$ 和 $\text{TG}_3$ 截止:

$\bar{Q}' = \bar{D}$ ;  $Q$ 经 $\text{TG}_4$ 回路保持原态。





- ② 当CLK由0变成1时,  $TG_2$ 和 $TG_3$ 导通, $TG_1$ 和 $TG_4$ 截止:  
 $\bar{Q}'$ 经 $TG_2$ 保持CLK上升沿到来前一瞬间时的D信号;  
 $\bar{D}$ 经 $TG_3$ 再经非门送到输出Q端.





## 表5.6 部分常用集成触发器

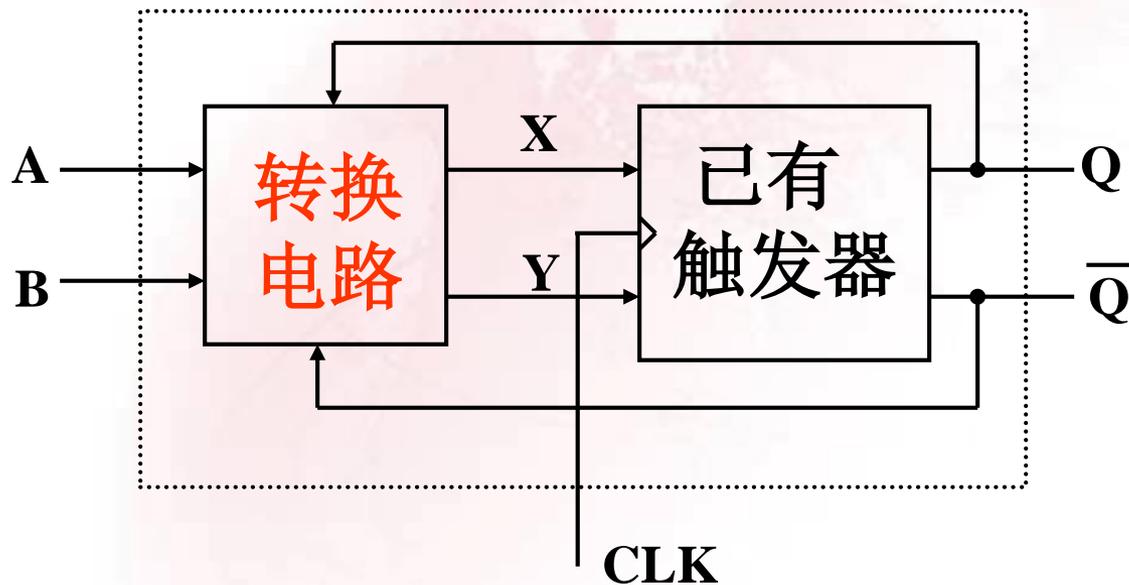
型 号	集成 器件数	功 能 说 明
7473A	2	负边沿 JK 触发器，有 $\overline{R}_D$ 端
7474	2	正边沿 D 触发器，有 $\overline{R}_D$ 和 $\overline{S}_D$ 端
74109	2	正边沿 JK 触发器，有 $\overline{R}_D$ 和 $\overline{S}_D$ 端
74LS112	2	负边沿 JK 触发器，有 $\overline{R}_D$ 和 $\overline{S}_D$ 端
74S113	2	负边沿 JK 触发器，有 $\overline{S}_D$ 端
74LS114	2	负边沿 JK 触发器，有 $\overline{R}_D$ 和 $\overline{S}_D$ 端
74174	6	正边沿 D 触发器，有 $\overline{R}_D$ 端
74175	4	正边沿 D 触发器，有 $\overline{R}_D$ 端
74273	8	正边沿 D 触发器，有 $\overline{R}_D$ 端
74276	4	负边沿 JK 触发器，有 $\overline{R}_D$ 和 $\overline{S}_D$ 端
74LS374	8	正边沿 D 触发器，含输出使能，三态输出



## 5.4 触发器使用中的几个问题

### 5.4.1 触发器逻辑功能的转换

触发器逻辑功能转换示意图：



将已有触发器转换为所需触发器的功能,实际上是求**转换电路**,即求转换电路的**函数表达式**:

$$X=f_1(A,B,Q^n)$$

$$Y=f_2(A,B,Q^n)$$



## 1. 代数法

通过比较已有触发器和待求触发器的特性方程, 求转换电路的函数表达式.



例: 把JK触发器转换为D触发器.

解: 已有JK触发器的特性方程为:  $Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$

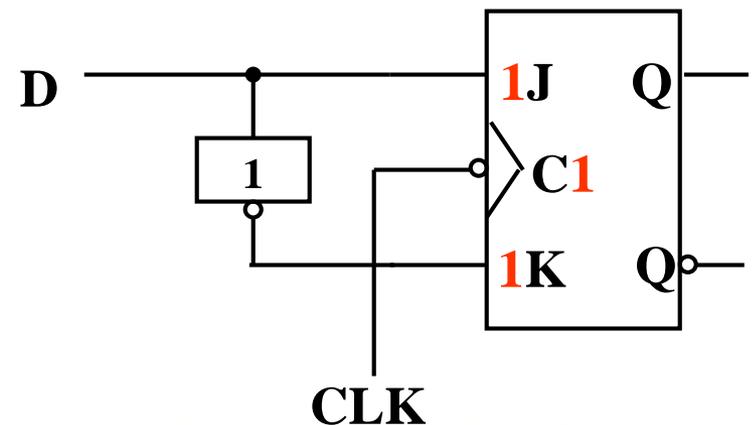
待求D触发器的特性方程为:  $Q^{n+1} = D$

为求出转换电路的函数表达式, 可将D触发器的特性方程转换为:

$$Q^{n+1} = D = D(\bar{Q}^n + Q^n) = D\bar{Q}^n + DQ^n$$

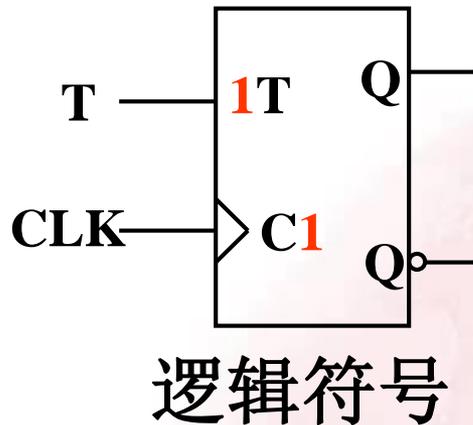
比较JK触发器的特性方程, 可得:

$$J = D \quad K = \bar{D}$$





例：将JK触发器转换为T触发器。



T触发器特性表

T	$Q^n$	$Q^{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

T触发器的特性归纳为：

T=0 保持

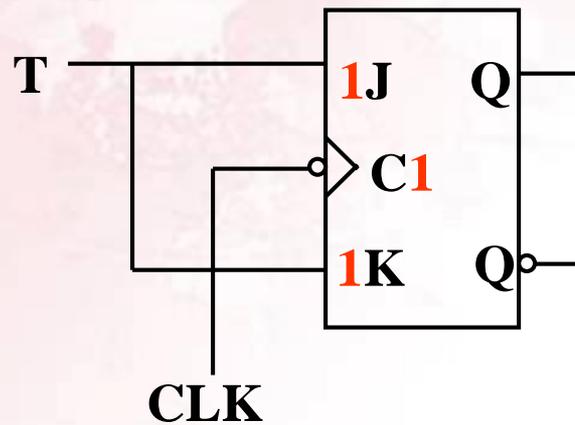
T=1 翻转



**T**触发器的特性方程:  $Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n$

将上式和JK触发器特性方程  $Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$  比较,可得

$$J=K=T$$



**注意:** 在这个电路中, 由于采用的是下降边沿JK触发器, 所以得到的T触发器也是下降边沿的。



## 2. 图表法

例：把RS触发器转换为JK触发器。

① 首先列出JK触发器的特性表；

② 根据RS触发器的特性,列出当满足JK触发器特性时S、R端应加的信号；

J	K	$Q^n$	$Q^{n+1}$	S	R
0	0	0	0	0	×
0	0	1	1	×	0
0	1	0	0	0	×
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	×	0
1	1	0	1	1	0
1	1	1	0	0	1



③ 写出下列两个表达式:

$$S=f_1(J, K, Q^n)$$

$$R=f_2(J, K, Q^n)$$

J \ KQ <sup>n</sup>	00	01	11	10
0	0	×	0	0
1	1	×	0	1

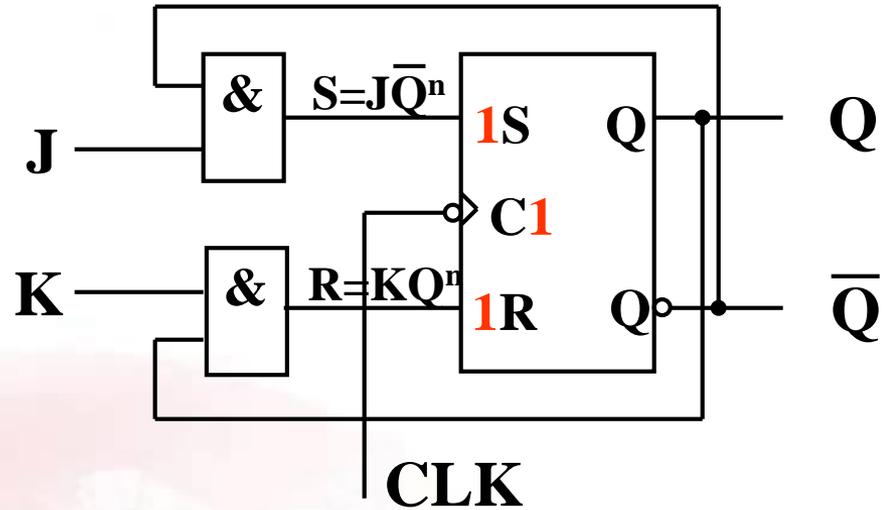
$$S=J\bar{Q}^n$$

J \ KQ <sup>n</sup>	00	01	11	10
0	×	0	1	×
1	0	0	1	0

$$R=KQ^n$$



## 转换电路图



例：试用D触发器和四选一MUX构成一个多功能触发器，其功能如下表所示。表中L、T为控制变量，N为数据输入变量。

L	T	N	$Q^{n+1}$
0	0	×	$Q^n$
0	1	×	$\overline{Q^n}$
1	0	N	N
1	1	N	$\overline{N}$

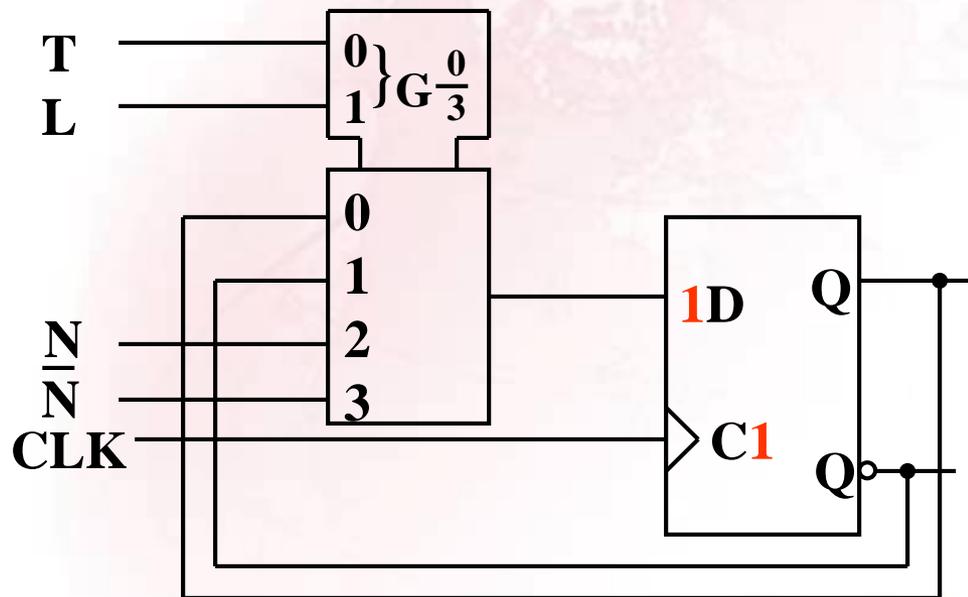


解：① 列表；

② 设L、T为MUX的地址变量，求MUX的数据端输入信号；

③ 画逻辑图。

L	T	N	$Q^{n+1}$	D
0	0	×	$Q^n$	$Q^n$
0	1	×	$\overline{Q^n}$	$\overline{Q^n}$
1	0	N	N	N
1	1	N	$\overline{N}$	$\overline{N}$



LT	N		
	0	1	
00	$Q^n$	$Q^n$	$D_0=Q^n$
01	$\overline{Q^n}$	$\overline{Q^n}$	$D_1=\overline{Q^n}$
11	1	0	$D_3=\overline{N}$
10	0	1	$D_2=N$



## 5.4.2 触发器的脉冲工作特性

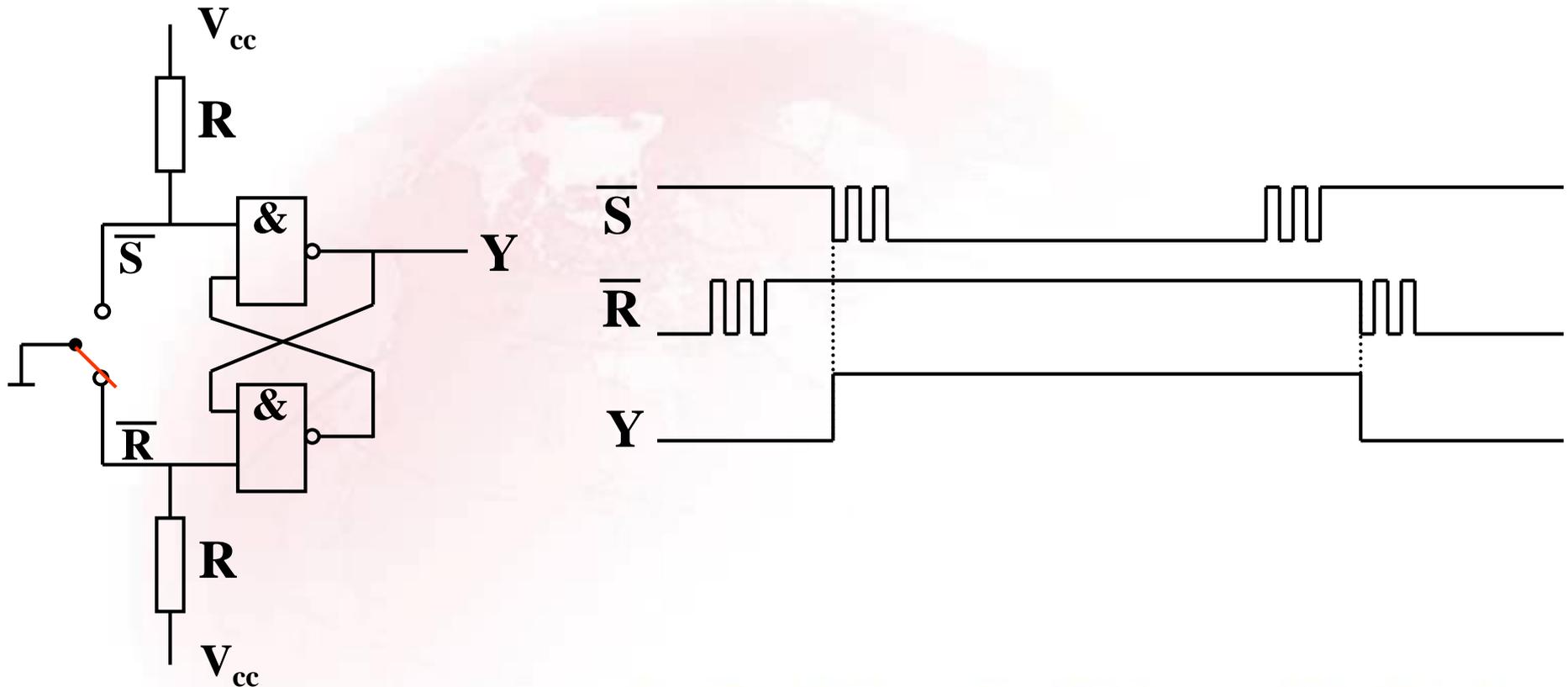
### 触发器的脉冲工作特性:

指为了保证触发器可靠的动作，而对时钟脉冲、输入信号以及它们之间的时间关系所提出的要求。



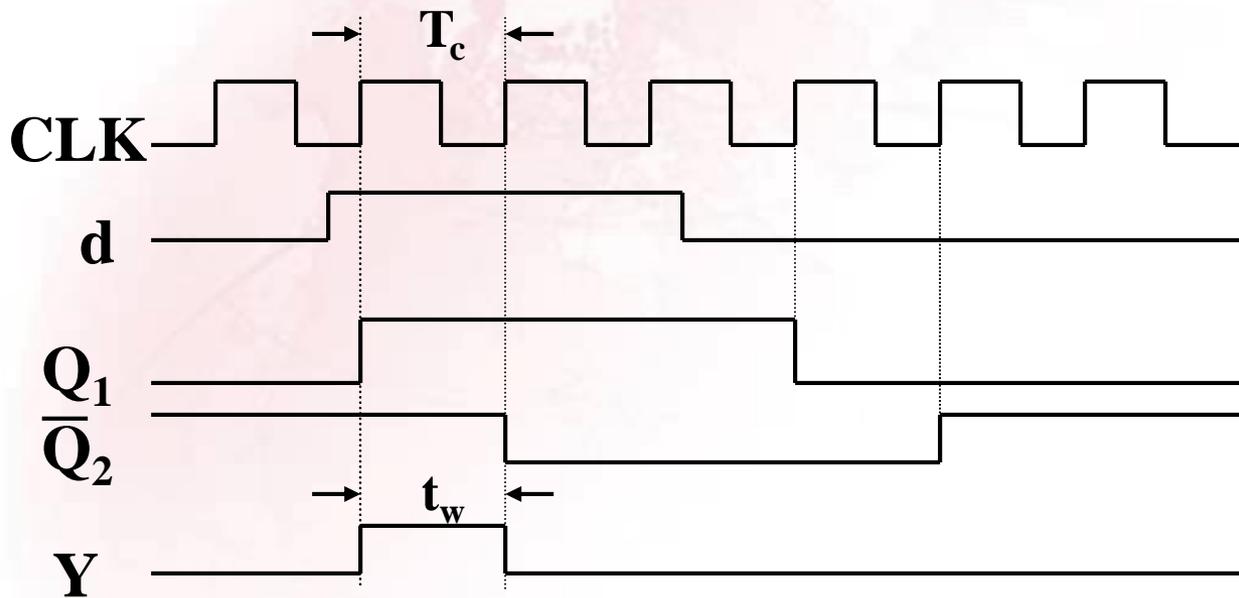
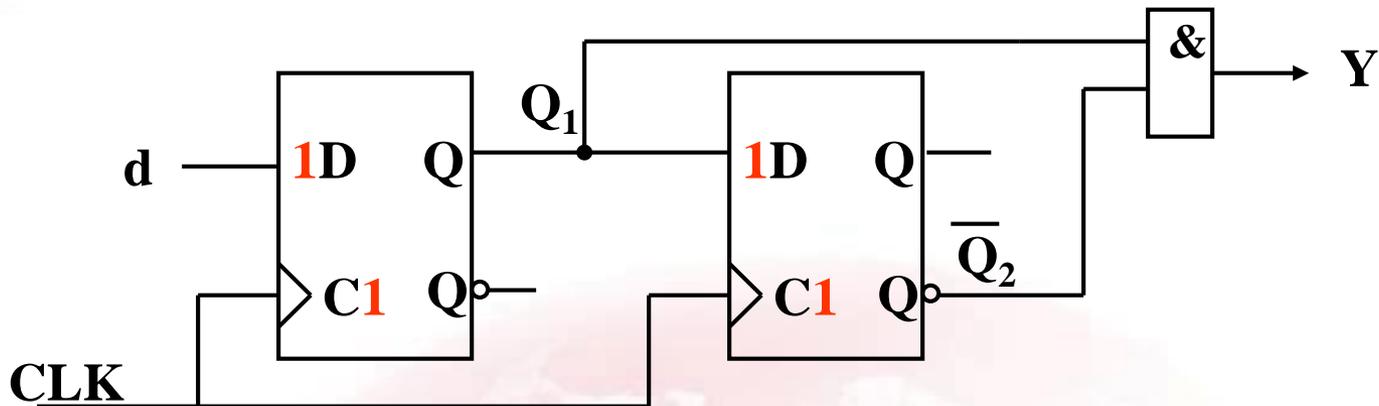
## 5.5 触发器应用举例

### 1. 消颤开关





## 2. 单脉冲发生器





## 5.6 时序逻辑电路的分析与设计

### 时序逻辑电路的分析方法

**分析目的:** 所谓分析, 就是由给定电路, 来找出电路的**功能**。对时序逻辑电路而言, 本质上是求**电路在不同的外部输入和当前状态条件下的输出情况和状态转换规律**。

**同步**时序逻辑电路和**异步**时序逻辑电路有不同的分析方法。



## 5.6.1 同步时序逻辑电路的分析

由于在同步时序电路中，各触发器的动作变化是在**CLK**脉冲作用下**同时**发生的，因此，在同步电路的分析中，只要知道了在**当前状态下各触发器的输入**（即**驱动信号**），就能根据触发器的**特性方程**，求得电路的**下一个状态**，最终找到电路的**状态转换规律**。



## 分析步骤:

- (1) 列出时序电路的**输出方程**和**驱动方程**(即该时序电路中**组合电路部分**的逻辑函数表达式);
- (2) 将上一步所得的**驱动方程**代入触发器的**特性方程**,导出电路的**状态方程**;
- (3) 根据**状态方程**和**输出方程**,列出**状态表**;
- (4) 根据**状态表**画出**状态图**或**时序图**;
- (5) 由**状态表**或**状态图**(或**时序图**)说明电路的**逻辑功能**.



例：分析下列时序电路。

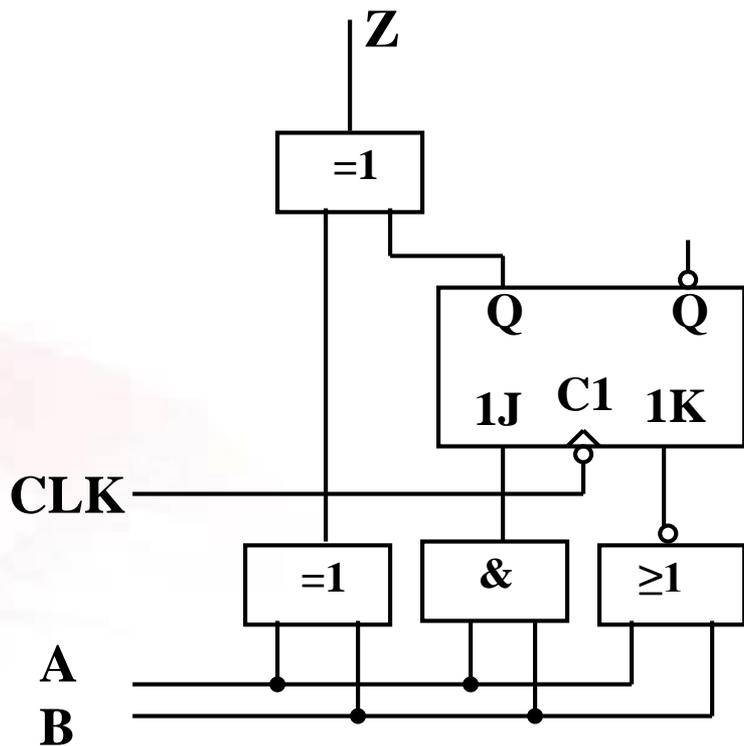
(1) 写出输出方程和驱动方程。

$$Z = A \oplus B \oplus Q^n$$

$$J = AB, \quad K = \overline{A+B}$$

(2) 写出状态方程。

$$\begin{aligned} Q^{n+1} &= J\overline{Q}^n + \overline{K}Q^n \\ &= AB\overline{Q}^n + (A+B)Q^n \\ &= AB\overline{Q}^n + AQ^n + BQ^n \end{aligned}$$

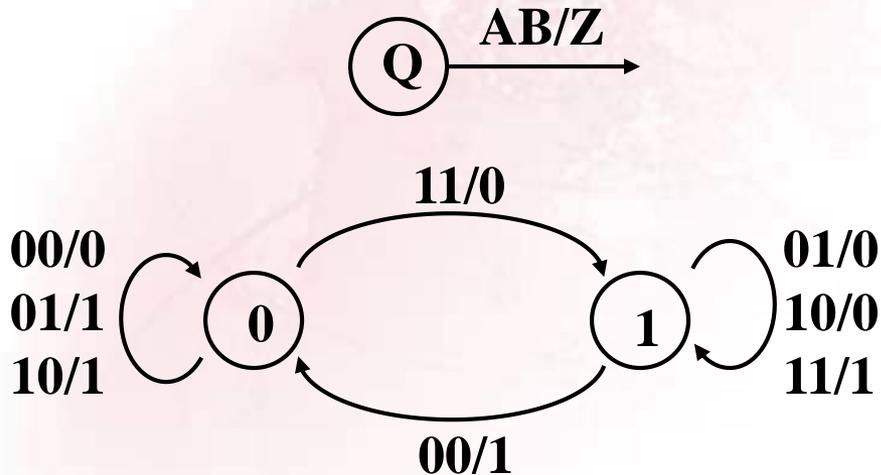




(3) 列出状态表.

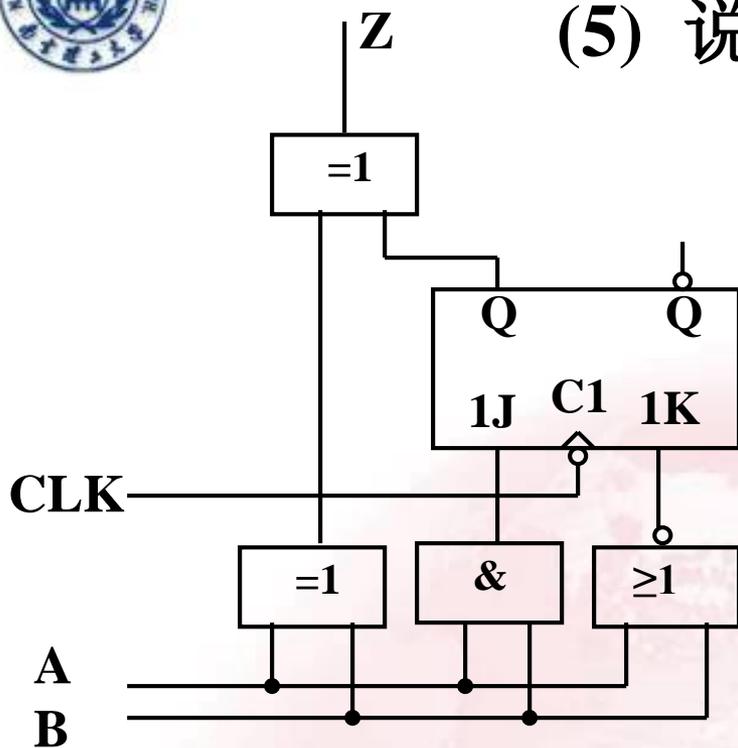
A	B	$Q^n$	$Q^{n+1}$	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(4) 列状态图.





## (5) 说明逻辑功能.



A	B	$Q^n$	$Q^{n+1}$	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

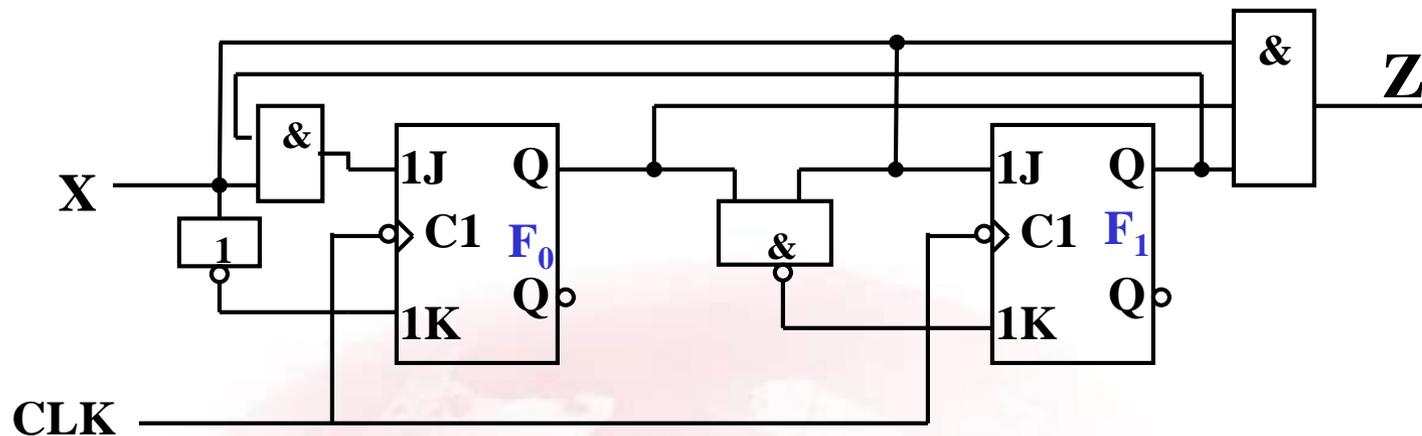
串行输入串行输出的时序全加器。A和B为两个二进制加数， $Q^n$ 为低位来的进位，Z表示相加的结果， $Q^{n+1}$ 表示向高位的进位。

问题：全加器如何工作？

一位一位串行加



例：分析下列时序电路的逻辑功能。



输出方程： $Z = XQ_0^n Q_1^n$

驱动方程：

$$J_0 = XQ_1^n, \quad K_0 = \bar{X}$$

$$J_1 = X, \quad K_1 = \bar{X} + \bar{Q}_0^n$$

JK触发器的特性方程：

$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$$

状态方程：

$$Q_0^{n+1} = XQ_1^n \bar{Q}_0^n + XQ_0^n$$

$$= X(Q_0^n + Q_1^n)$$

$$Q_1^{n+1} = X\bar{Q}_1^n + \bar{X} + \bar{Q}_0^n Q_1^n$$

$$= X(Q_0^n + \bar{Q}_1^n)$$



输出方程:  $Z = XQ_0^n Q_1^n$

状态方程:

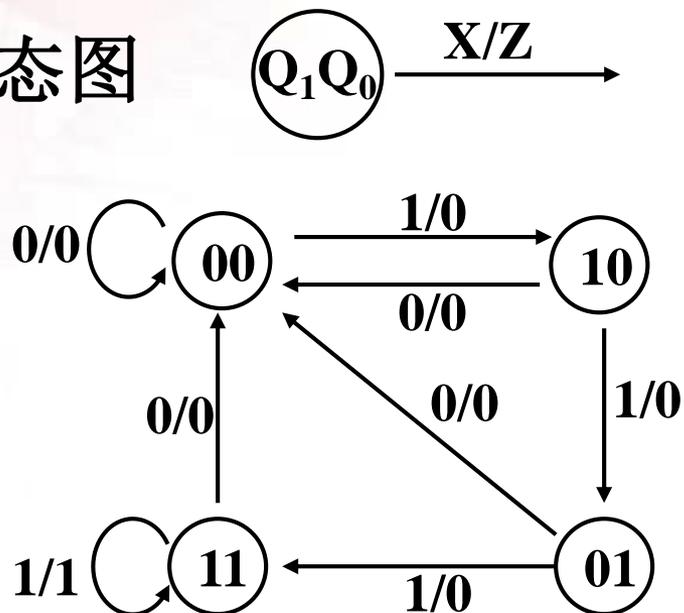
$$Q_0^{n+1} = XQ_1^n \bar{Q}_0^n + XQ_0^n = X(Q_0^n + Q_1^n)$$

$$Q_1^{n+1} = X\bar{Q}_1^n + \bar{X} + \bar{Q}_0^n Q_1^n = X(Q_0^n + \bar{Q}_1^n)$$

状态表

X	$Q_1^n$	$Q_0^n$	$Q_1^{n+1}$	$Q_0^{n+1}$	Z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	1	1	1	1

状态图



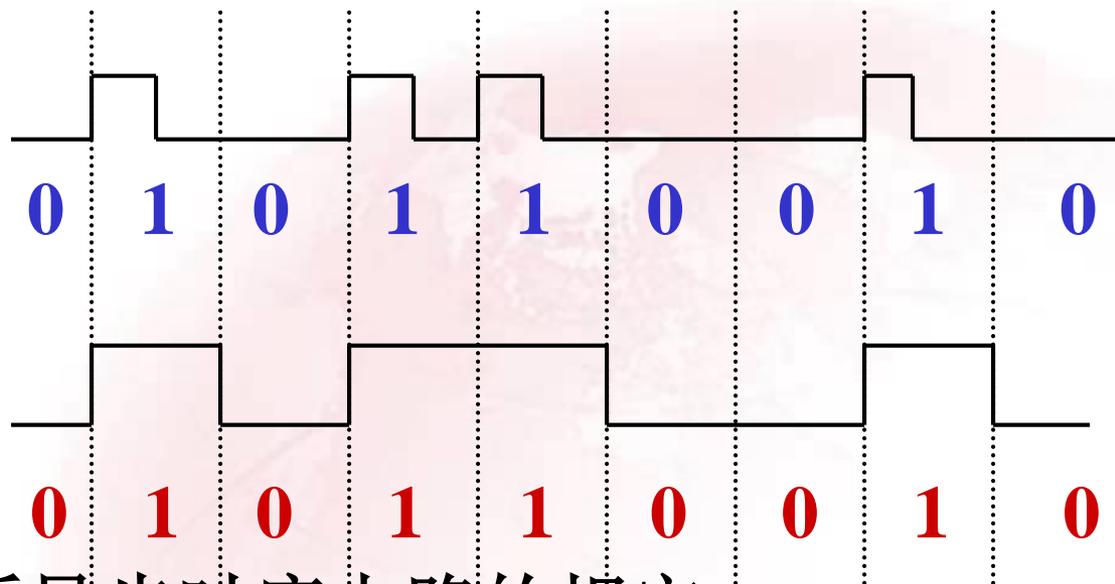
功能: **1111序列检测器**



## 5.6.2 异步时序逻辑电路的分析方法

异步时序逻辑电路分类:

- 1) **脉冲型**: 用脉冲的有无表示信号;
- 2) **电位型**: 用电位的高低表示信号;



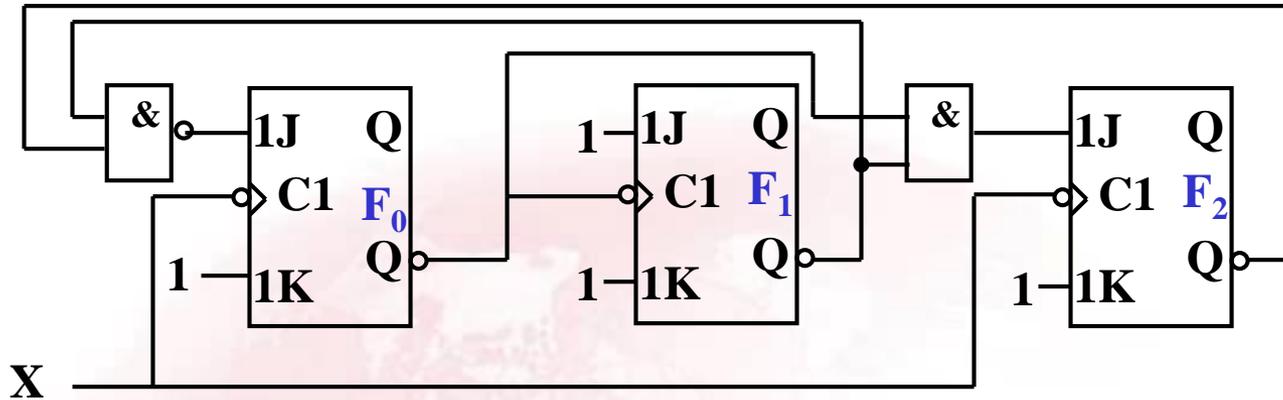
分析异步时序电路的规定:

- 1) 输入信号只有在电路**稳定状态**时才发生变化;
- 2) 每一个时刻仅允许一个**输入变量**发生变化.



## 1、脉冲型异步时序电路的分析方法(波形分析法)

例：试分析下列异步时序电路的逻辑功能



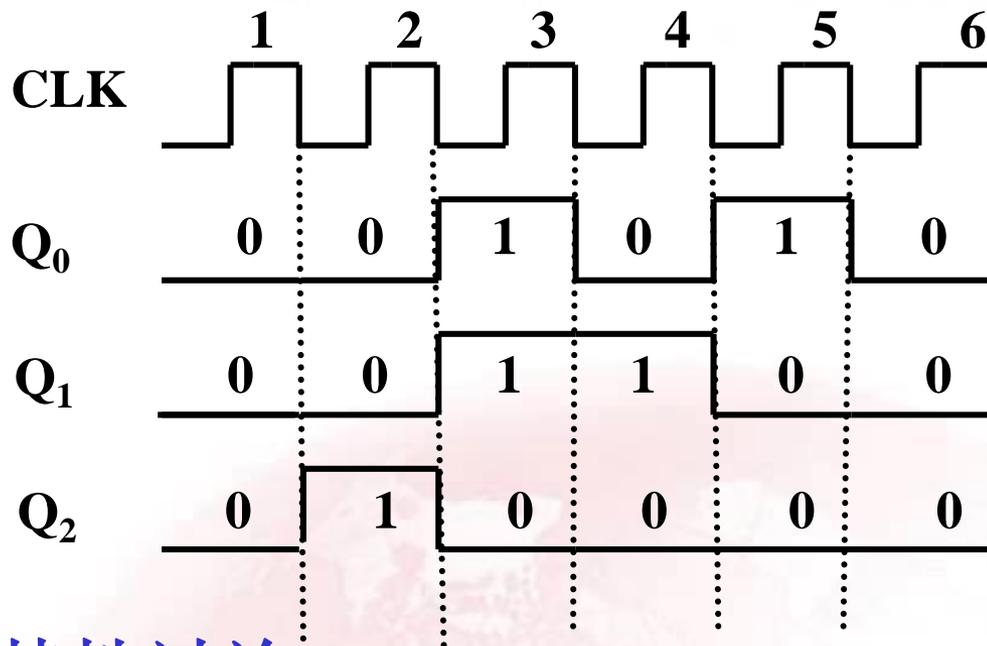
解：时钟信号  
驱动方程

$$CLK_1 = \bar{Q}_0 \quad CLK_0 = CLK_2 = X$$

$$J_0 = \bar{Q}_2^n \bar{Q}_1^n = Q_2^n + Q_1^n, \quad K_0 = 1$$

$$J_1 = K_1 = 1$$

$$J_2 = \bar{Q}_1^n \bar{Q}_0^n, \quad K_2 = 1$$



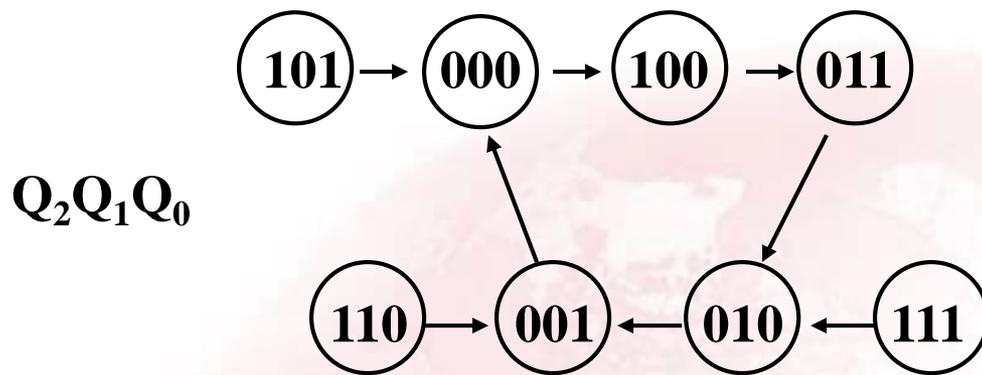
## 自启动特性讨论:

- (1) 当 $Q_2Q_1Q_0=101$ 时, 下一个状态为 $000$ ;
- (2) 当 $Q_2Q_1Q_0=110$ 时, 下一个状态为 $001$ ;
- (3) 当 $Q_2Q_1Q_0=111$ 时, 下一个状态为 $010$ ;

电路能自启动



## 状态图

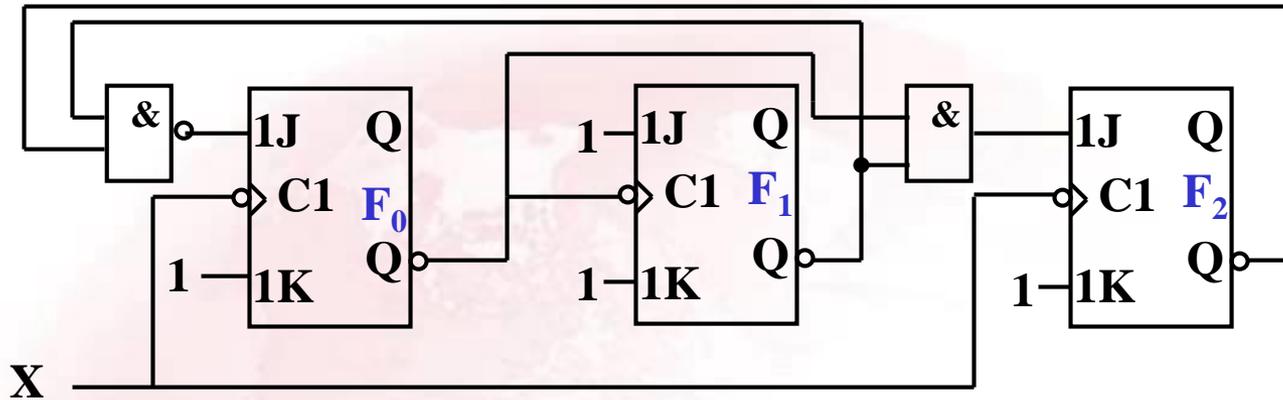


功能:

能自启动的五进制减法计数器

## (分析法二)

例：试分析下列异步时序电路的逻辑功能



解：1) 写驱动方程

$$J_0 = \overline{Q_2^n} \overline{Q_1^n} = Q_2^n + Q_1^n, \quad K_0 = 1$$

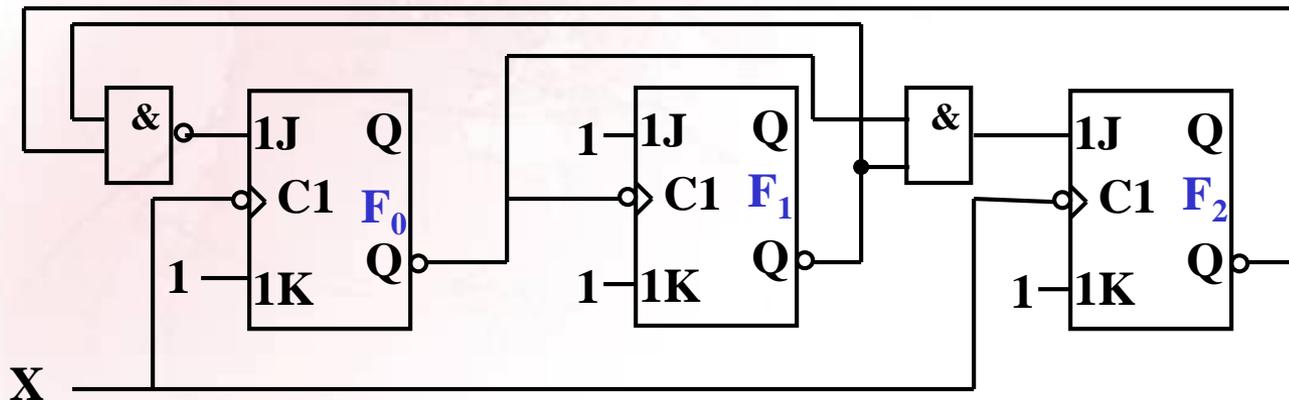
$$J_1 = K_1 = 1$$

$$J_2 = \overline{Q_1^n} \overline{Q_0^n}, \quad K_2 = 1$$



2) 写触发器 $F_0$ 和 $F_2$ 的**状态方程**(由于触发器 $F_0$ 和 $F_2$ 是在**X** 脉冲作用下同步工作的,列方程时将**X**隐含)

$$Q_2^{n+1} = J_2 \bar{Q}_2^n + \bar{K}_2 Q_2^n = \bar{Q}_1^n \bar{Q}_0^n \bar{Q}_2^n$$
$$Q_0^{n+1} = J_0 \bar{Q}_0^n + \bar{K}_0 Q_0^n = (Q_2^n + Q_1^n) \bar{Q}_0^n$$







## 4) 求CLK<sub>1</sub>

由Q<sub>2</sub>、Q<sub>0</sub>的状态方程

$$Q_2^{n+1} = J_2 \bar{Q}_2^n + \bar{K}_2 Q_2^n = \bar{Q}_1^n \bar{Q}_0^n \bar{Q}_2^n$$

$$Q_0^{n+1} = J_0 \bar{Q}_0^n + \bar{K}_0 Q_0^n = (Q_2^n + Q_1^n) \bar{Q}_0^n$$

Q <sub>2</sub> <sup>n</sup>	Q <sub>1</sub> <sup>n</sup>	Q <sub>0</sub> <sup>n</sup>	Q <sub>2</sub> <sup>n+1</sup>	Q <sub>1</sub> <sup>n+1</sup>	Q <sub>0</sub> <sup>n+1</sup>	CLK <sub>1</sub>
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	0	1	1	1
0	1	1	0	0	0	0
1	0	0	0	1	1	1
1	0	1	0	0	0	0
1	1	0	0	1	1	1
1	1	1	0	0	0	0

		Q <sub>1</sub> Q <sub>0</sub>			
		00	01	11	10
Q <sub>2</sub>	0				1
	1	1			1

$$CLK_1 = Q_2^n \bar{Q}_0^n + Q_1^n \bar{Q}_0^n$$



5) 将驱动方程 $J_1$ 、 $K_1$ 和时钟方程 $CLK_1$ 代入触发器 $F_1$ 的特性方程,求得 $F_1$ 的状态方程.

$$Q_1^{n+1} = (J_1 \bar{Q}_1^n + \bar{K}_1 Q_1^n) CLK_1 + Q_1^n \overline{CLK_1} \quad (J_1 = K_1 = 1)$$

$$= \bar{Q}_1^n (Q_2^n \bar{Q}_0^n + Q_1^n \bar{Q}_0^n) + Q_1^n (Q_2^n \bar{Q}_0^n + Q_1^n \bar{Q}_0^n)$$

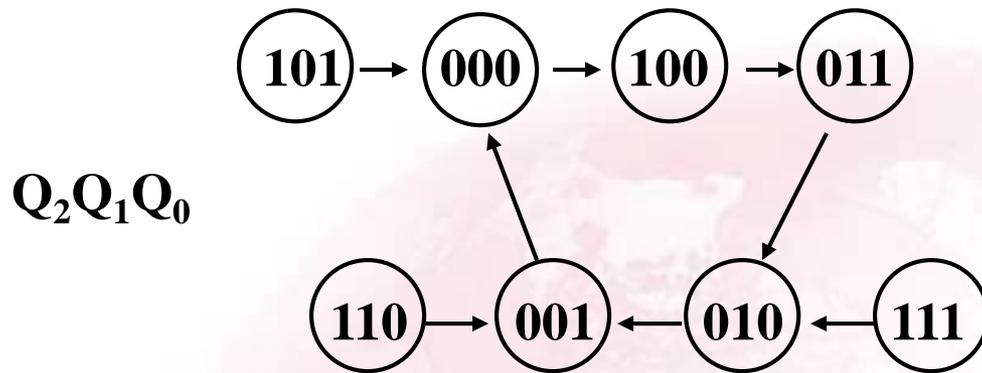
$$= Q_2^n \bar{Q}_1^n \bar{Q}_0^n + Q_1^n Q_0^n$$

根据状态方程,补齐上述状态表.

$Q_2^n$	$Q_1^n$	$Q_0^n$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	0	1
1	1	1	0	1	0



6) 画出状态图,分析电路功能.



功能:

能自启动的五进制减法计数器



## 5.6.3 同步时序逻辑电路的设计

### 1. 同步时序逻辑电路的一般步骤

- (1) 根据逻辑要求,建立原始状态表或原始状态图;
- (2) 利用状态化简技术,简化原始状态表,消去多余状态;
- (3) 状态分配或状态编码,即将简化后的状态用二进制代码表示;
- (4) 选择触发器类型,并根据编码后的状态表求出驱动方程和输出方程;
- (5) 检查自启动性,若在所设计电路中存在无效状态,则必须检查电路能否自启动,如果不能自启动,则需修改设计;
- (6) 画出逻辑图.



例：试设计一个“111”序列检测器.要求：当连续输入三个或三个以上“1”时,输出为“1”,否则输出为“0”.

**X: 0 1 1 0 1 1 1 0 1 1 1 1 0**

**Z: 0 0 0 0 0 0 1 0 0 0 1 1 0**



解: (1) 建立原始状态表

$S_0$ : 输入0以后的状态;(即未收到一个“1”以前的状态);

$S_1$ : 输入一个“1”以后的状态;

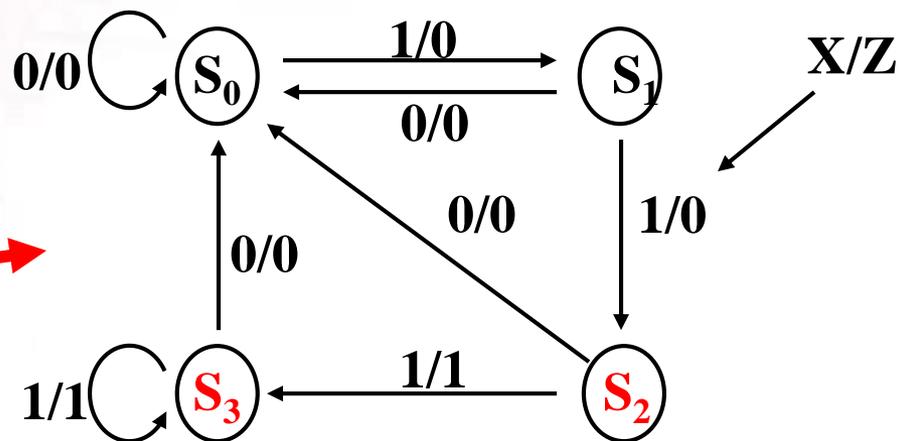


$S_2$ : 连续输入二个“1”以后的状态;

$S_3$ : 连续输入三个或三个以上“1”以后的状态。

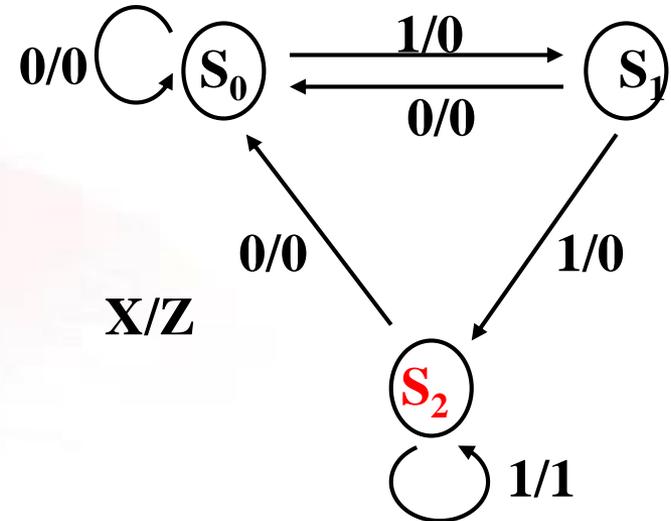
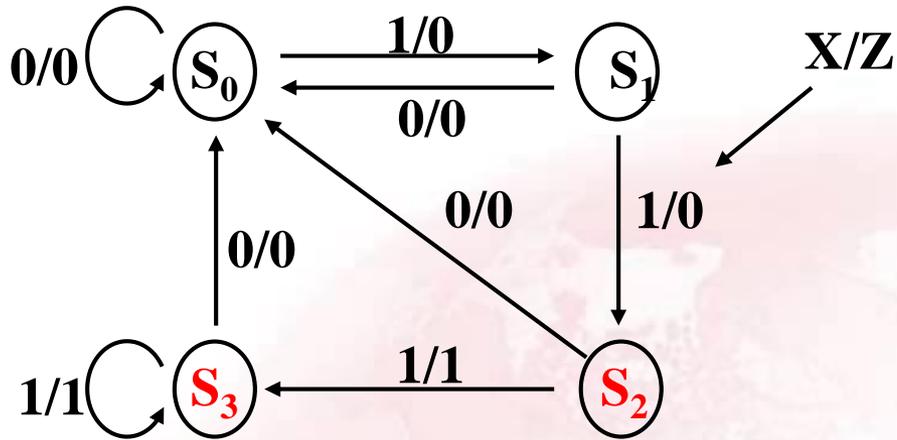
$S \backslash X$	0	1
$S_0$	$S_0/0$	$S_1/0$
$S_1$	$S_0/0$	$S_2/0$
$S_2$	$S_0/0$	$S_3/1$
$S_3$	$S_0/0$	$S_3/1$

原始状态图





## (2) 状态化简

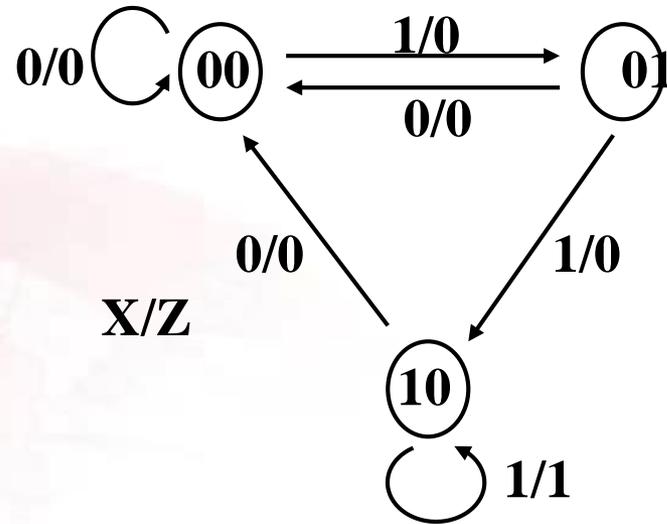
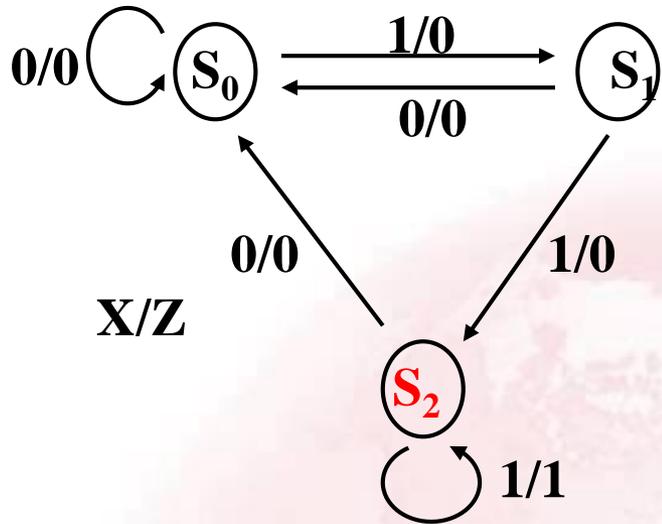


状态 $S_2$ 和 $S_3$ 在相同的输入下有相同的输出,而次态也相同,称 $S_2$ 和 $S_3$ 两个状态等价.等价状态仅需保留一个.这里,去除 $S_3$ ,保留 $S_2$ ,可得简化状态图.



## (3) 状态编码

3个状态,需要2个触发器,每个状态用2位二进制编码.



S \ X	0	1
S <sub>0</sub>	S <sub>0</sub> /0	S <sub>1</sub> /0
S <sub>1</sub>	S <sub>0</sub> /0	S <sub>2</sub> /0
S <sub>2</sub>	S <sub>0</sub> /0	S <sub>3</sub> /1
S <sub>3</sub>	S <sub>0</sub> /0	S <sub>3</sub> /1

Q <sub>1</sub> <sup>n</sup> Q <sub>0</sub> <sup>n</sup> \ X	0	1
00	00/0	01/0
01	00/0	10/0
11	××/×	××/×
10	00/0	10/1



#### (4) 选择触发器类型,求驱动方程和输出方程;

当触发器选定以后, 可根据状态表, 对照触发器的**输入表**, 求出驱动方程.

#### 四种常用触发器的输入表

$Q^n$	$Q^{n+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

$Q^n$	$Q^{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

$Q^n$	$Q^{n+1}$	S	R
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

$Q^n$	$Q^{n+1}$	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0



本例如选用JK触发器,对照状态表和JK触发器的输入表,可列出驱动卡诺图和输出卡诺图.

$Q_1^n Q_0^n \backslash X$	0	1
00	00/0	01/0
01	00/0	10/0
11	××/×	××/×
10	00/0	10/1

$$Q_1^{n+1} Q_0^{n+1} / Z$$

$Q^n$	$Q^{n+1}$	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

$Q_1^n Q_0^n \backslash X$	0	1
00	0	0
01	0	0
11	×	×
10	0	1

$$Z = XQ_1^n$$

$Q_1^n Q_0^n \backslash X$	0	1
00	0	0
01	0	1
11	×	×
10	×	×

$$J_1 = XQ_0^n$$

$Q_1^n Q_0^n \backslash X$	0	1
00	×	×
01	×	×
11	×	×
10	1	0

$$K_1 = \bar{X}$$

$Q_1^n Q_0^n \backslash X$	0	1
00	0	1
01	×	×
11	×	×
10	0	0

$$J_0 = X\bar{Q}_1^n$$

$Q_1^n Q_0^n \backslash X$	0	1
00	×	×
01	1	1
11	×	×
10	×	×

$$K_0 = 1$$



本例如选用D触发器,对照状态表和D触发器的输入表,可列出驱动卡诺图和输出卡诺图.

$Q_1^n Q_0^n \backslash X$	0	1
00	00/0	01/0
01	00/0	10/0
11	××/×	××/×
10	00/0	10/1

$Q^n$	$Q^{n+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

$Q_1^{n+1} Q_0^{n+1} / Z$

$Q_1^n Q_0^n \backslash X$	0	1
00	0	0
01	0	1
11	×	×
10	0	1

$$D_1 = X(Q_0^n + Q_1^n)$$

$Q_1^n Q_0^n \backslash X$	0	1
00	0	1
01	0	0
11	×	×
10	0	0

$$D_0 = X\bar{Q}_0^n \bar{Q}_1^n$$

$Q_1^n Q_0^n \backslash X$	0	1
00	0	0
01	0	0
11	×	×
10	0	1

$$Z = XQ_1^n$$



## (5) 检查自启动特性

本例存在无效状态 $Q_1Q_0=11$ ,由上面卡诺图可见:

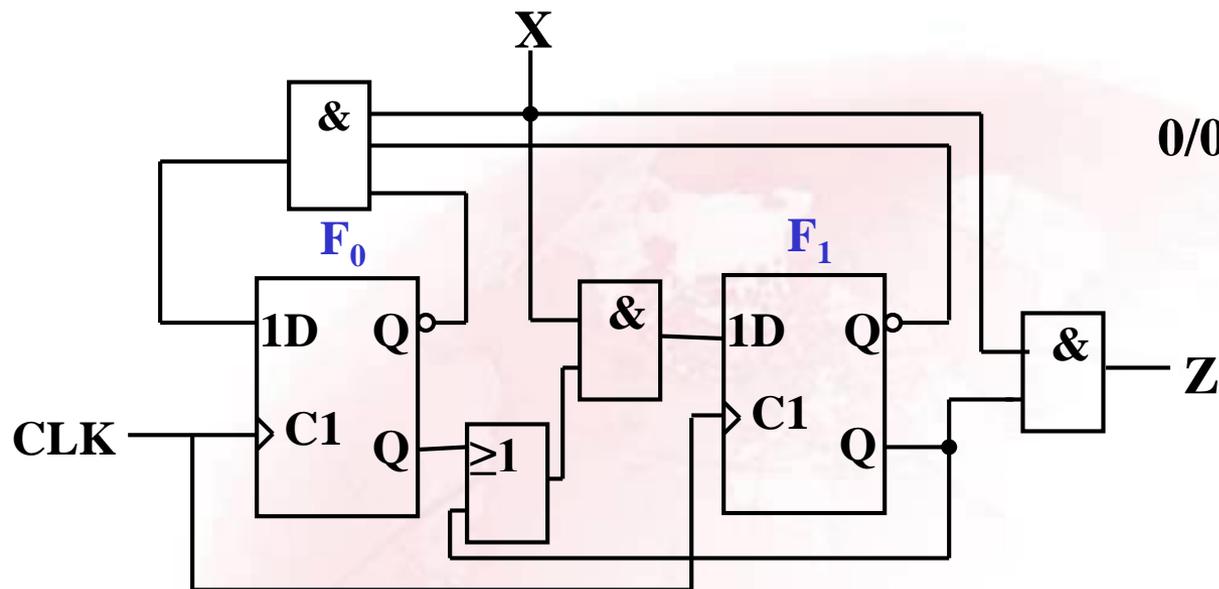
当 $Q_1Q_0=11$ 时,若 $X=0$ ,则 $D_1D_0=00$ ,次态为**00**;

当 $Q_1Q_0=11$ 时,若 $X=1$ ,则 $D_1D_0=10$ ,次态为**10**;

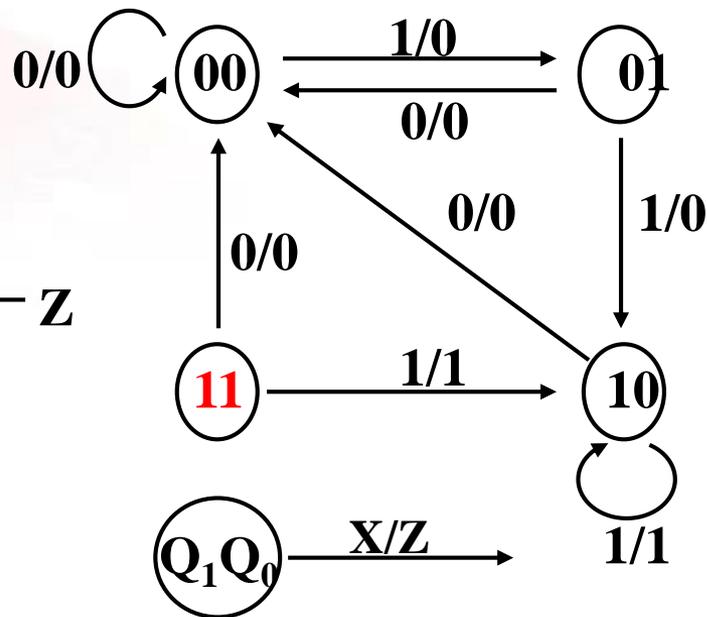
结论: **能自启动**.



(6) 画逻辑图和完整的状态图.



完整状态图





说明:

由于D触发器的特性方程 $Q^{n+1}=D$ ，所以:

$$\begin{aligned} Q_1^{n+1} &= D_1 = XQ_0^n + XQ_1^n \\ &= XQ_0^n (Q_1^n + \overline{Q_1^n}) + XQ_1^n \\ &= XQ_0^n \overline{Q_1^n} + XQ_1^n \end{aligned}$$

得到:  $J_1 = XQ_0^n$        $K_1 = \overline{X}$



$$\begin{aligned} Q_0^{n+1} &= D_0 = X\overline{Q_0}^n\overline{Q_1}^n \\ &= X\overline{Q_1}^n\overline{Q_0}^n + \overline{1}Q_0^n \end{aligned}$$

可得： $J_0 = X\overline{Q_0}^n$        $K_0 = 1$

用方程比较法较图标法简单，但同样存在结果出错的可能。



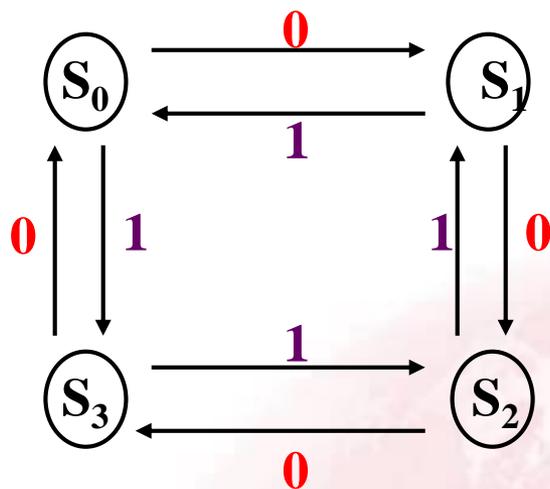
例:

试用**JK触发器**设计一个可控电路： $X$ 为控制信号，当 **$X=0$** 时，电路按照**0, 1, 2, 3, 0, 1, 2, 3, ……**的规律做**加法计数**；当 **$X=1$** 时，电路按照**3, 2, 1, 0, 3, 2, 1, 0, ……**的规律做**减法计数**。

（该电路称为模4可逆计数器，有关计数器的概念将在第6章中详细介绍）



解：电路有4个状态： $S_0$ 、 $S_1$ 、 $S_2$ 和 $S_3$ ,状态图和状态表为



$S \backslash X$	0	1
$S_0$	$S_1$	$S_3$
$S_1$	$S_2$	$S_0$
$S_2$	$S_3$	$S_1$
$S_3$	$S_0$	$S_2$

由于有四个状态，需用两个触发器，

定义： $S_0 \longrightarrow 00$

$S_1 \longrightarrow 01$

$S_2 \longrightarrow 10$

$S_3 \longrightarrow 11$

编码后的真值表

$Q_1^n Q_0^n \backslash X$	0	1
0 0	01	11
0 1	10	00
1 0	11	01
1 1	00	10



分离状态表，求出状态方程(求驱动方程的**第二种方法**)

编码后的真值表

$Q_1^n Q_0^n \backslash X$	0	1
00	01	11
01	10	00
10	11	01
11	00	10

$Q_1^n Q_0^n \backslash X$	0	1
00	0	1
01	1	0
11	0	1
10	1	0

$Q_1^{n+1}$

$Q_1^n Q_0^n \backslash X$	0	1
00	1	1
01	0	0
11	0	0
10	1	1

$Q_0^{n+1}$

$$\begin{aligned} Q_1^{n+1} &= \bar{Q}_1^n \bar{Q}_0^n X + \bar{Q}_1^n Q_0^n \bar{X} + Q_1^n Q_0^n X + Q_1^n \bar{Q}_0^n \bar{X} \\ &= (Q_0^n \oplus X) \bar{Q}_1^n + \overline{(Q_0^n \oplus X)} Q_1^n \\ &= (Q_0^n \oplus X) \oplus Q_1^n \end{aligned}$$



$$Q_1^{n+1} = (Q_0^n \oplus X) \oplus Q_1^n$$

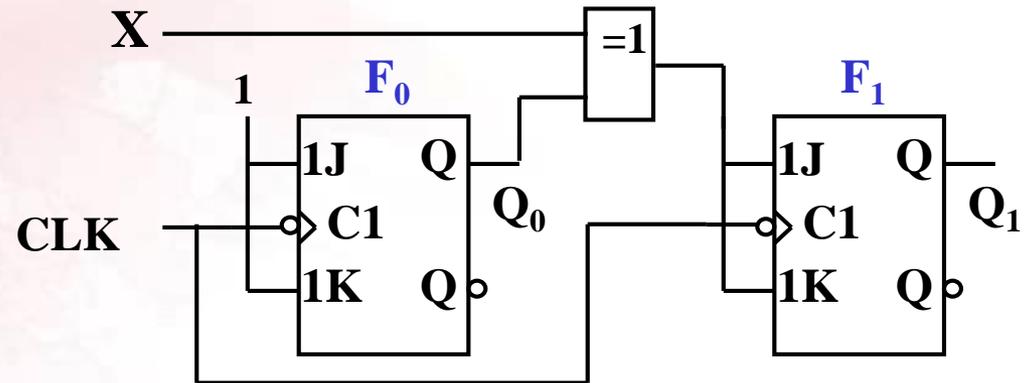
参考T触发器特性方程:

$$Q^{n+1} = T \oplus Q^n$$

可得:  $J_1 = K_1 = Q_0 \oplus X$

$$Q_0^{n+1} = \bar{Q}_0^n = 1 \oplus Q_0^n$$

可得:  $J_0 = K_0 = 1$



逻辑图



## 5.6.4 有限状态机的VHDL描述

**有限状态机**（Finite State Machine，简称FSM）是指输出取决于过去输入部分和当前输入部分的时序逻辑电路。

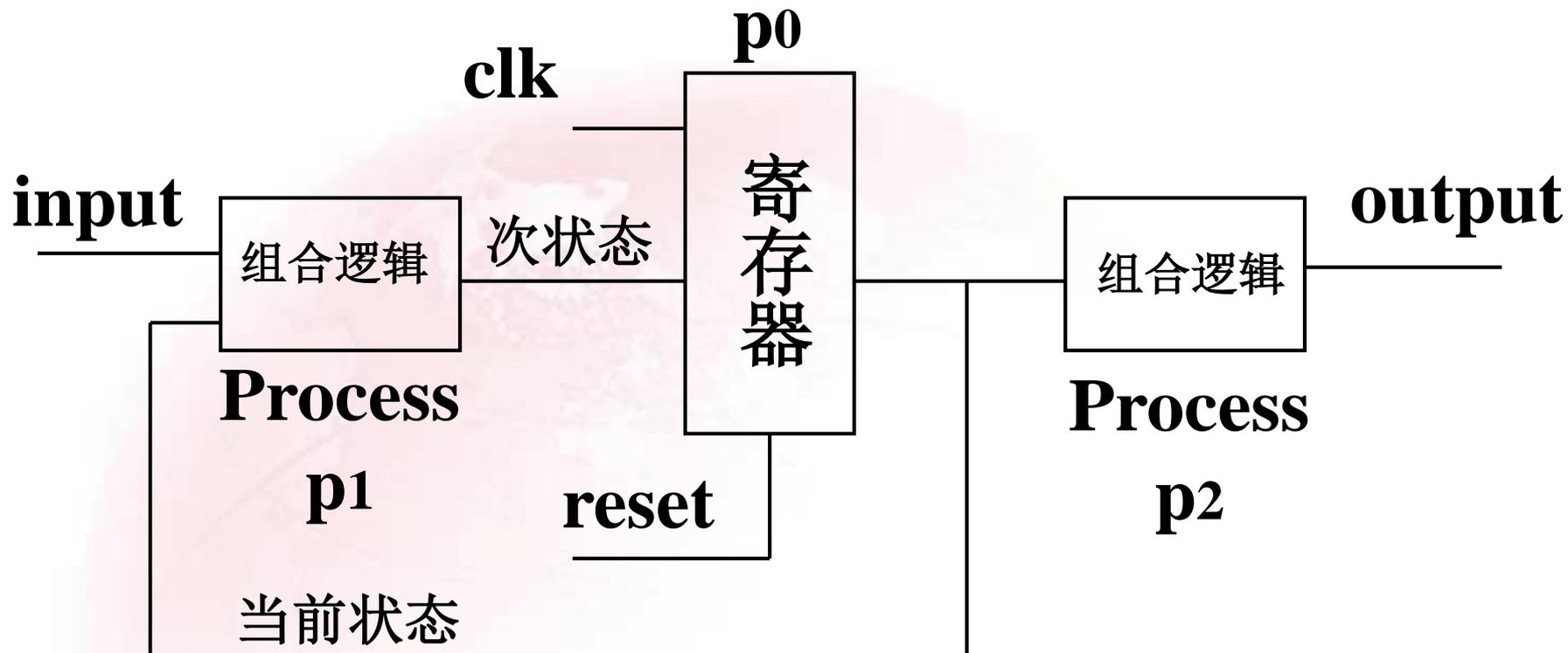
**有限状态机分类：**

**Moore型有限状态机和Mealy型有限状态机**



# 1. Moore型有限状态机的VHDL描述

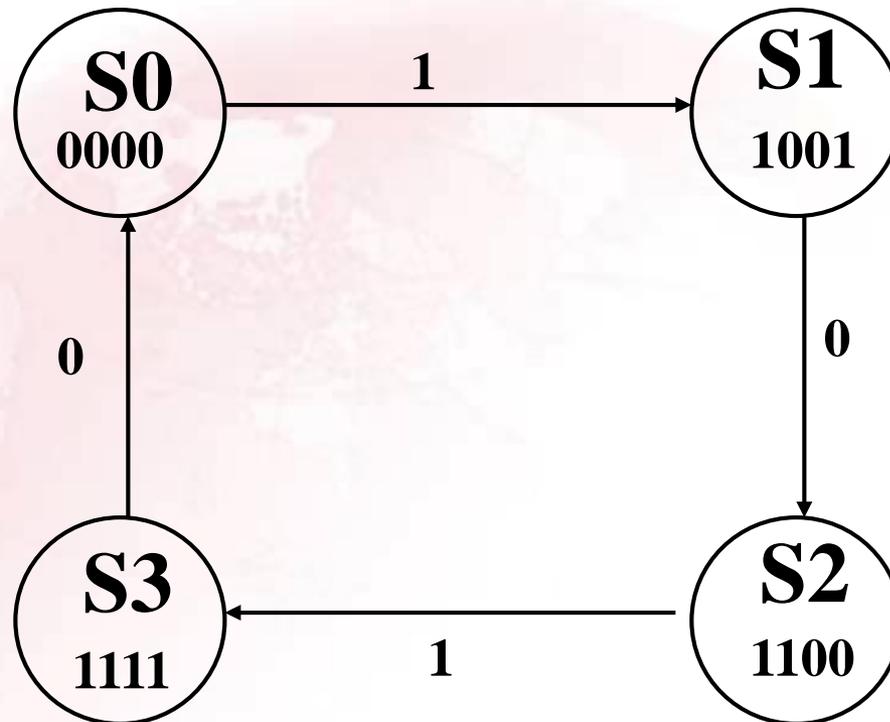
## Process



Moore型状态机结构框图



【例5.16】 用VHDL语言设计一个如下面的状态图所示的Moore型有限状态机。





```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
ENTITY moore IS
```

```
    PORT(
```

```
        clk,in1,reset    : IN    STD_LOGIC;
```

```
        out1: OUT        STD_LOGIC_vector(3 downto 0));
```

```
END ;
```

```
architecture bhv of moore is
```

```
    type state_type is (s0,s1,s2,s3); --状态说明
```

```
    signal current_state,next_state:state_type;
```



```
begin
```

```
p0: process (clk,reset)      --时钟进程
```

```
begin
```

```
  if reset='1' then
```

```
    current_state <= s0;
```

```
  elsif clk'event and clk='1'then
```

```
    current_state<=next_state;
```

```
  end if;
```

```
end process;
```



p1: process(current\_state,in1) --组合进程

begin

CASE current\_state IS

WHEN s0 =>if in1='1'then

next\_state<=s1; end if;

WHEN s1 =>if in1='0'then

next\_state<=s2; end if;

WHEN s2 =>if in1='1'then

next\_state<=s3; end if;

WHEN s3 =>if in1='0'then

next\_state<=s0; end if;

END CASE;

end process;



p2:process(current\_state)   --组合进程

begin

  case current\_state is

    when s0 => out1 <="0000";

    when s1 => out1 <="1001";

    when s2 => out1 <="1100";

    when s3 => out1 <="1111";

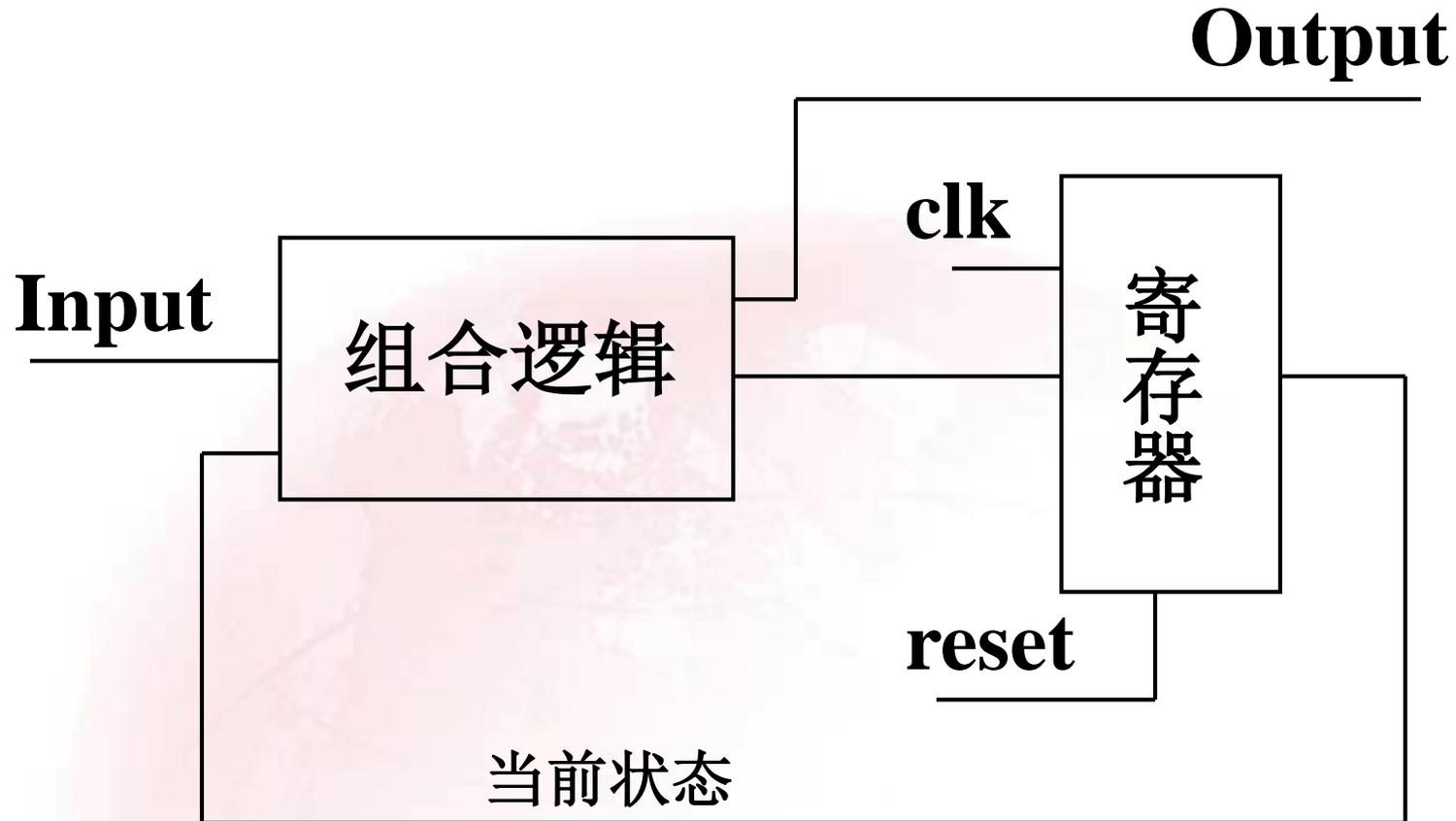
  end case;

end process;

end bhv;



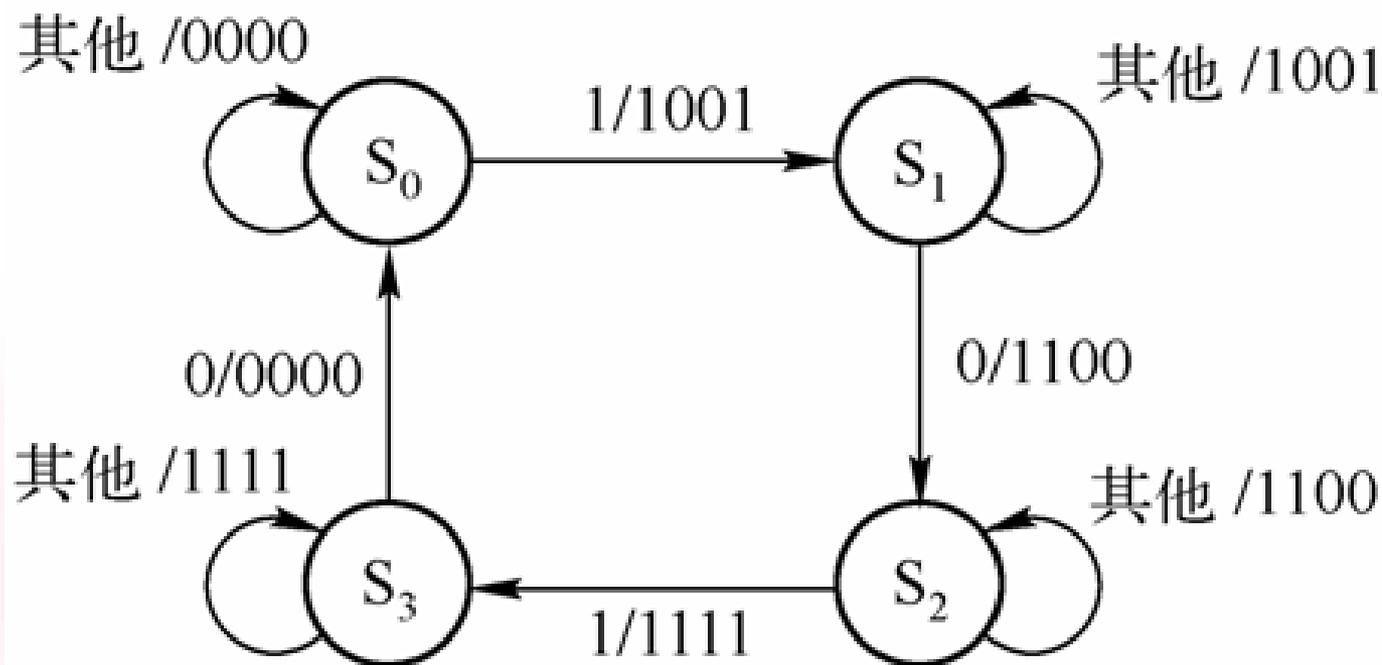
## 2. Mealy型有限状态机的VHDL描述



Mealy型有限状态机的结构框图



**【例5.17】** 用VHDL语言将例5.16设计成一个Mealy型有限状态机，其状态图如下图所示。





```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
ENTITY mealy IS
```

```
PORT(clk,in1,reset: IN   STD_LOGIC;
```

```
      out1: OUT  STD_LOGIC_vector(3 downto 0));
```

```
END ;
```

```
architecture bhv of mealy is
```

```
  type state_type is (s0,s1,s2,s3);
```

```
  signal state:state_type;
```

```
begin
```



```
p0: process (clk,reset)
```

```
begin
```

```
  if reset='1' then
```

```
    state<=s0;
```

```
  elsif clk'event and clk='1'then
```

```
CASE state IS
```

```
  WHEN s0 => if in1='1'then
```

```
    state<=s1;
```

```
  end if;
```

```
  WHEN s1 => if in1='0'then
```

```
    state<=s2;
```

```
  end if;
```

```
  WHEN s2 => if in1='1'then
```

```
    state<=s3;
```

```
  end if;
```

```
  WHEN s3 => if in1='0'then
```

```
    state<=s0;
```

```
  end if;
```

```
END CASE;
```

```
end if;
```

```
end process p0;
```

该进程完成状态转换的描述



```
out_p:process(state,in1)
begin
  case state is
    when s0 => if in1='1' then out1 <="1001";
               else
               out1 <="0000";
               end if;
    when s1 => if in1='0' then out1 <="1100";
               else
               out1 <="1001";
               end if;
```



```
when s2=> if in1='1' then out1 <="1111";
```

```
    else
```

```
        out1 <="1100";
```

```
    end if;
```

```
when s3 => if in1='0' then out1 <="0000";
```

```
    else
```

```
        out1 <="1111";
```

```
    end if;
```

```
end case;
```

```
end process;
```

```
end bhv;
```

该进程完成由状态和输入决定输出