

# 数字逻辑设计

Digital Logic Design

高翠芸

School of Computer Science

gaocuiyun@hit.edu.cn

# 数字逻辑的知识脉络

后续课程：  
如计算机  
组成原理  
等

本课程的内容

数字系统或计算机应用系统

Logisim、Verilog、VHDL等工具

组合逻辑电路

时序逻辑电路

编码器  
译码器  
全加器  
比较器  
...  
数据选择器

寄存器  
计数器  
.....  
节拍发生器

FPGA

CPLD

...

PROM

ROM

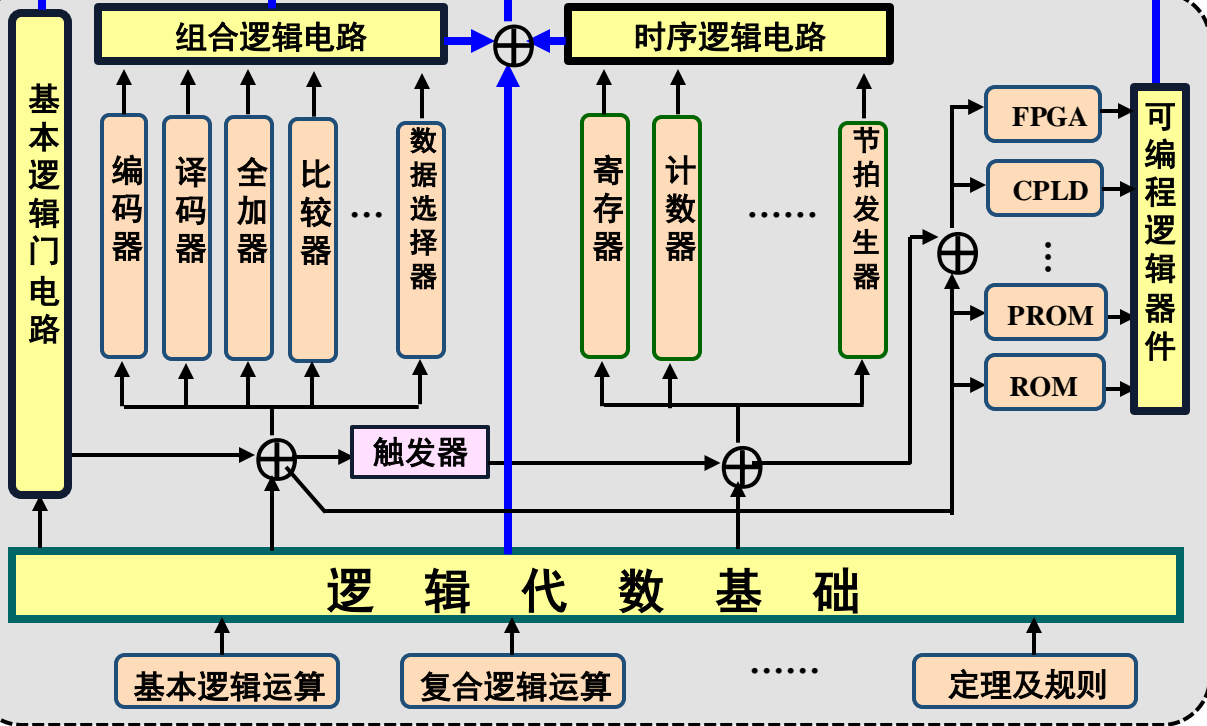
可编程逻辑器件

逻辑代数基础

基本逻辑运算

复合逻辑运算

定理及规则



# 时序逻辑元件

---

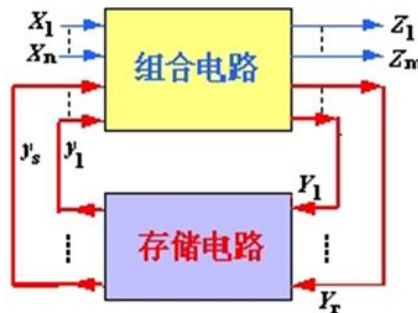
- 锁存器 (Latch)
- 触发器 (Flip-Flop)
- 带附加输入端的边沿触发器
- 触发器类型转换

# 时序逻辑电路的特点

## 组合电路的特点

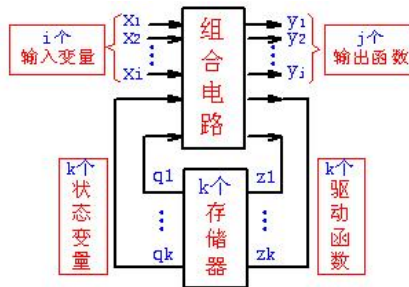
- 它是一种**无记忆电路**——输入信号消失，则输出信号也会立即消失
- 电路输出端的状态完全由输入端的状态决定

- 有时需要将参与运算的数据和运算结果保存起来——在组合逻辑电路的输出端添加具有**记忆功能**的部件



# 组合逻辑电路 vs 时序逻辑电路

- **锁存器和触发器**是构成存储电路的基本元件
- 两个重要概念：现态（原态）和 次态（新态）



构成		定义	结构	电路框图	逻辑函数表达式
数字逻辑电路	组合逻辑电路	<p>任意时刻的输出——</p> <ul style="list-style-type: none"> <li>■ 仅与当前时刻的输入有关</li> </ul> $Z_m = f_m(x_1, \dots, x_n)$	不包含存储元件		<p>只有一组：</p> $Z_m = f_m(x_1, \dots, x_n)$
	时序逻辑电路	<p>任意时刻的输出与以下均有关：</p> <ul style="list-style-type: none"> <li>■ 当前时刻的输入</li> <li>■ 电路过去（上一个时刻）的工作状态</li> </ul> $Z_m = f_m(x_1, \dots, x_n, y_1, \dots, y_s)$	包含存储元件		<p>有三组：</p> <p><b>输出方程, 驱动方程, 状态方程：</b></p> $Z_m = f_m(x_1, \dots, x_n, y_1, \dots, y_r)$ $Y_r = g_r(x_1, \dots, x_n, y_1, \dots, y_s)$ $Y_s^{n+1} = q_s(x_1, \dots, x_n, Y_1^n, \dots, Y_s^n)$

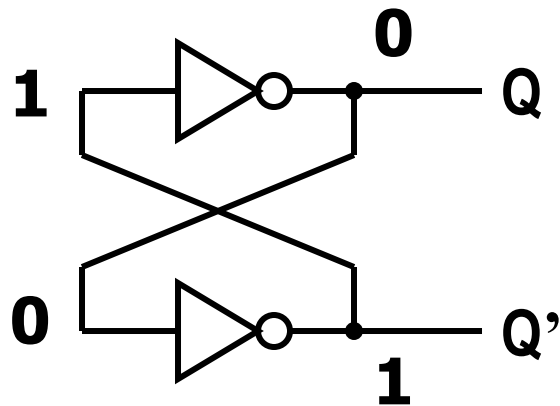
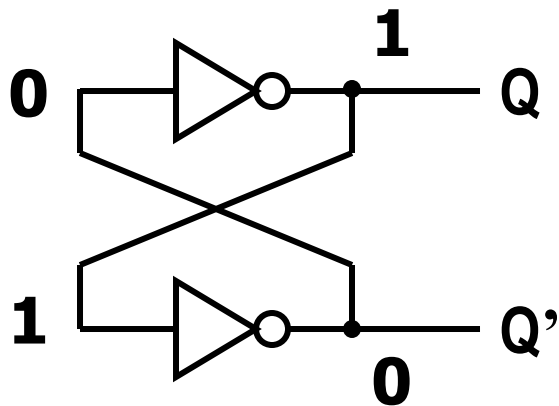
# 锁存器和触发器

- 锁存器：没有时钟输入端
- 触发器：有时钟输入端，并且只在时钟信号到来时，才发生状态转换

## ◆ 锁存器与触发器的特性（双稳态）

1. 有两个互补的输出端  $Q$  和  $Q'$
2. 有两个稳定的状态：0态，1态
3. 在外界信号的刺激下，可以从一个稳定状态转变到另一个稳定状态。
4. 没有（或无效的）外界信号刺激，维持当前状态不变。

# 双稳态



是否满足双稳态，有什么不足？

# 时序逻辑元件

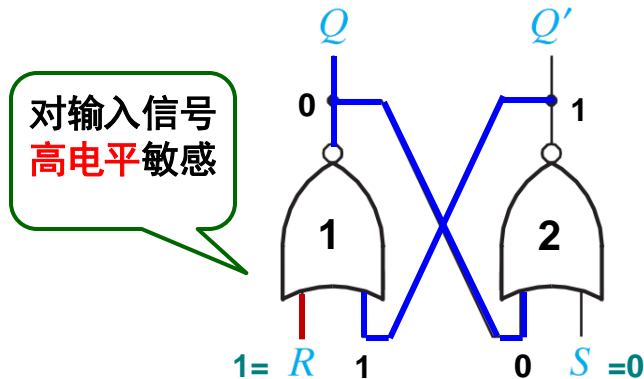
---

- 锁存器 (Latch)
  - 或非门构成的基本RS锁存器
  - 与非门构成的基本RS锁存器
  - 门控D锁存器
- 触发器 (Flip-Flop)
- 带附加输入端的边沿触发器
- 触发器类型转换



# 基本RS锁存器（触发器的鼻祖）

(1) 电路构成（或非门）



$Q (Q_n)$  —— 现态

$Q^+ (Q_{n+1})$  —— 次态

$Q = 0 (\bar{Q} = 1)$  : state 0

$Q = 1 (\bar{Q} = 0)$  : state 1

$R$  : 置0端(Reset the output to  $Q=0$ )

$S$  : 置1端(Set the output to  $Q=1$ )

(2) 功能表

置0端 $R$	置1端 $S$	现态 $Q_n$	次态 $Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	—
1	1	1	—

RS 对 同时  
取1互斥

保持

置 1

置 0

× 不允许

置0端 $R$	置1端 $S$	次态 $Q_{n+1}$
0	0	$Q_n$
0	1	1
1	0	0
1	1	—

输入高电平  
有效

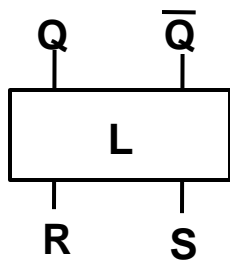
# 基本RS锁存器次态方程、逻辑符号等

## (3) 次态方程

$$Q_{n+1} = S + \bar{R}Q_n$$

(SR = 0) 约束条件

## (4) 逻辑符号



		S	
		0	1
RQ	00	0	1
	01	1	1
	11	0	X
	10	0	X

功能表

置0端 R	置1端 S	现态 $Q_n$	次态 $Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	—
1	1	1	—

# 基本RS锁存器驱动表

(5) 驱动表：完成状态转换需要满足的输入条件

用于时序  
电路设计

$Q_n \rightarrow Q_{n+1}$	R	S
0 → 0	X	0
0 → 1	0	1
1 → 0	1	0
1 → 1	0	X

置0端 R	置1端 S	次态 $Q_{n+1}$
0	0	$Q_n$
0	1	1
1	0	0
1	1	—

保持

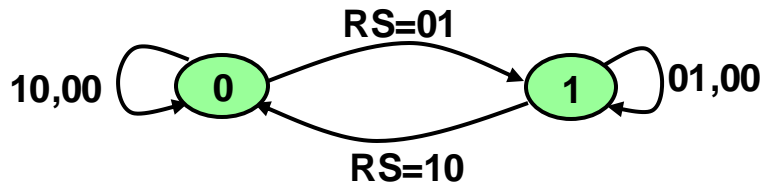
置 0

(6) 状态图

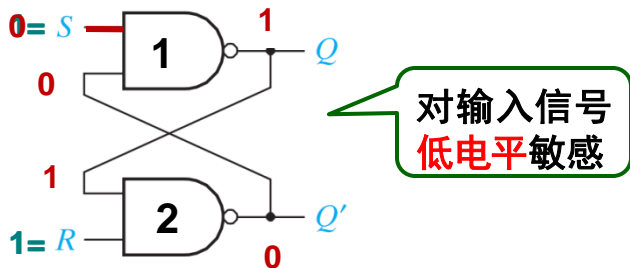
反映时序电路状态转移规律及相应输入、输出取值关系的有向图

图中元素的含义

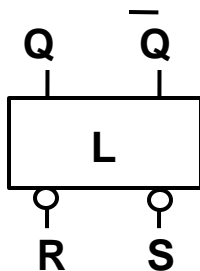
- 圆圈：表示电路的状态
- 有向线段：表示状态的转换关系
- 有向线段旁的文字：表示转换条件，即输入信号取值



# 另一种形式的基本RS锁存器（与非门）



逻辑符号:



(2) 功能表

置0端 R	置1端 S	现态 $Q_n$	次态 $Q_{n+1}$
1	1	0	0
1	1	1	1
1	0	0	1
1	0	1	1
0	1	0	0
0	1	1	0
0	0	0	—
0	0	1	—

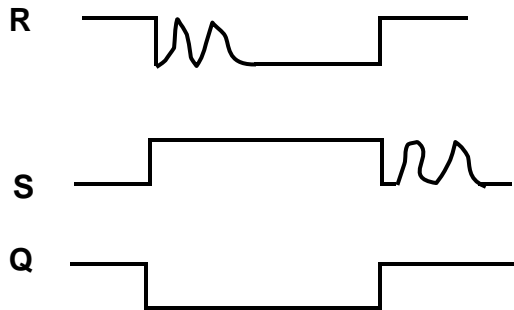
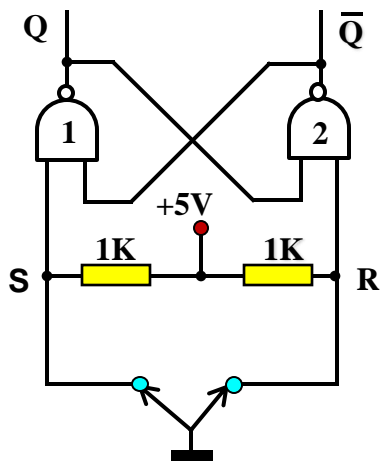
保持

置 1

置 0

× 不允许

# 锁存器的应用——开关去抖



- ❖ 由于机械弹性作用, 机械式开关在使用中, 通常伴随有一定时间的触点机械抖动。
- ❖ 触点抖动可能导致判断出错(一次按下或释放被错误地认为是多次操作)

## (7) 典型芯片

74LS279: 4 R-S latches

# RS锁存器小结

---

- 优点：结构简单

- 缺点：

  - ① 输入存在约束，使用不便；

  - ② 状态改变由输入直接控制。给使用带来局限性。

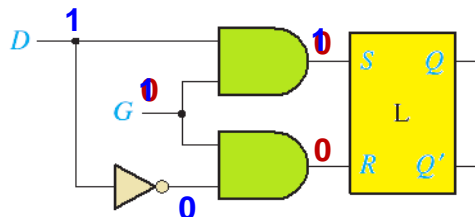
- 用途：记忆输入状态

- 基本RS锁存器是众多触发器的鼻祖

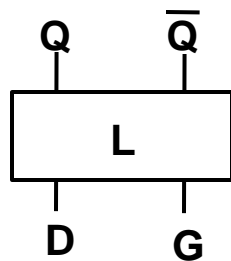
  - 其余的触发器都是在其基础上逐步改进和完善后形成的

# 门控D锁存器

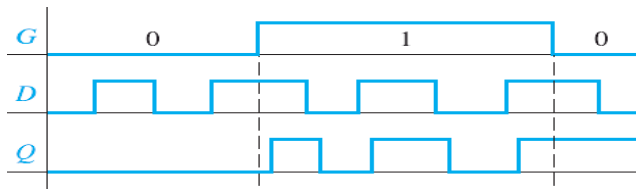
## (1) 电路构成



## (4) 逻辑符号



## (5) 时序分析



## (2) 功能表

使能端 G	输入端 D	现态 $Q_n$	次态 $Q_{n+1}$
0	X	0	0
0	X	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

## (3) 次态方程

$Q \backslash GD$	00	01	11	10
0	0	0	1	0
1	1	1	1	0

$$Q_{n+1} = GD + \bar{G}Q_n$$

在G为高电平期间，Q端的输出直接拷贝D端波形

## (6) 典型芯片

74LS373: 8D锁存器

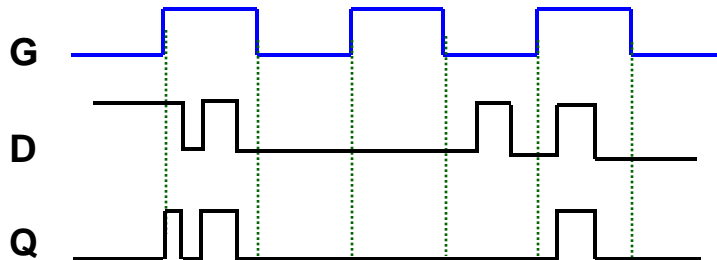
# 门控D锁存器的优缺点

- ◆ 特点：结构简单，仅一个输入端，不存在输入约束问题。
- ◆ 缺点：使能电位G作用期间，只要输入信号D改变（有时是干扰信号），Q也跟着改变，存在“空翻”现象

违背了构造时钟触发器的初衷：一个时钟内，最多允许触发器状态翻转一次

锁存器的使能端送时钟信号，电平触发方式的触发器

一个时钟内，触发器状态发生多次变化



“空翻”现象是锁存器（或电平方式触发器）共有的问题

“空翻”使以上器件不能正确实现计数功能！

- ☆ 关键问题：电平（电位）触发
- ☆ 解决方案：改电平触发为边沿触发

特征：时钟信号的上升沿或下降沿，触发器改变状态



# 什么样的Verilog描述会生成锁存器

只发生在组合逻辑电路中

## 1、if...else...语句没有else

```
always @ (*) begin
    if (d_en) q = d;
end
```

## 2、case语句没有default

锁存器的危害:

- 使静态时序分析变得非常复杂
- 对毛刺敏感，不能异步复位，上电后处于不确定的状态

```
reg q;
always @ (*) begin
    if (d_en) q2 = d;
    else      q2 = 1'h0;
end

always @ (*) begin
    case (cnt[1:0])
        2'b00 : q = d1;
        2'b01 : q = d2;
        2'b10 : q = d3;
    endcase
end
```

# 生成锁存器（latch）的示例

```
7   output reg [3:0] led
8 );
9   always @(*) begin
10    if(enable==1'b1)begin
11      if(select==1'b1)led=input_a-input_b;
12      else if(select==1'b0)led=input_a+input_b;
13      else led=0;
14    end
15    else if(enable==1'b0) led=4'b0000;
```

Open Block Design

Generate Block Design

## SIMULATION

Run Simulation

## RTL ANALYSIS

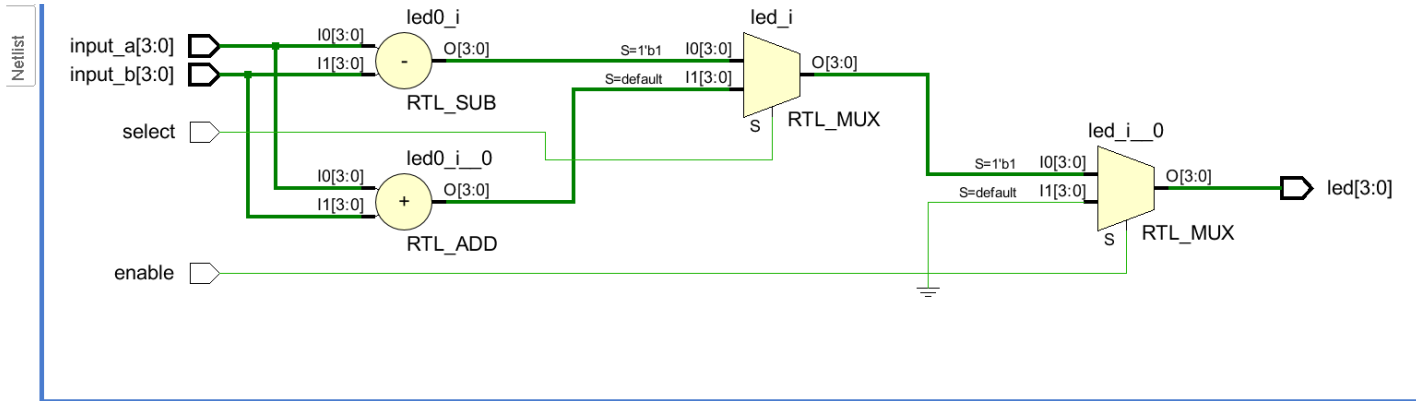
### Open Elaborated Design

Report Methodology

Report DRC

Report Noise

Schematic



Tcl Console

Messages

Log

Reports

Design Runs

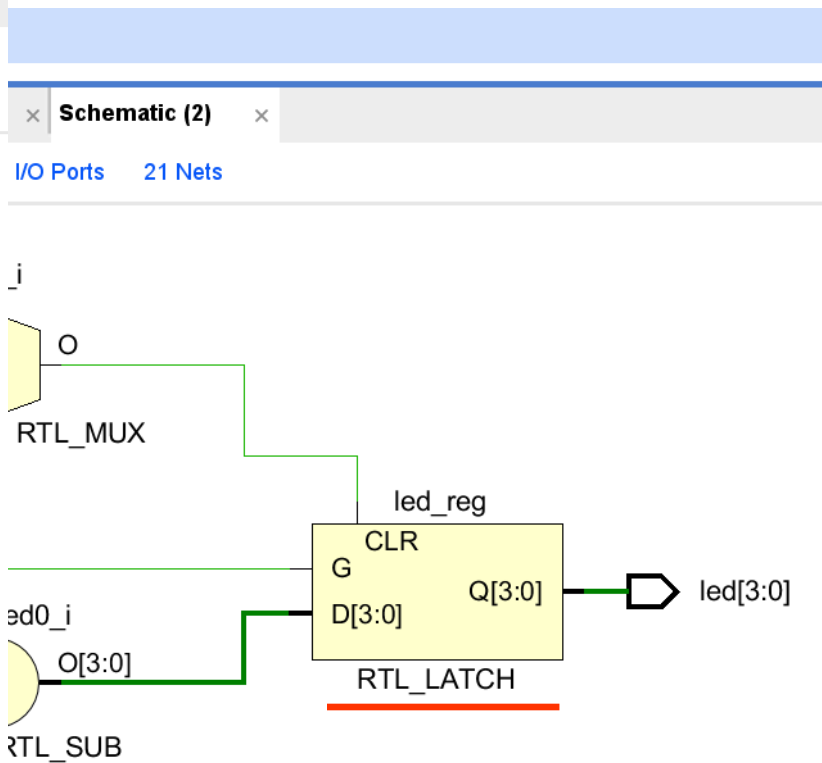
# 生成锁存器（latch）的示例——续

Project Summary x Schematic x multiplexers.v x testbench.v x Schematic (2) x

C:/VerilogCode/multiplexer/multiplexer.srcs/sources\_1/imports/multipler/multiplexers.v

Q [Icons]

```
3 module multiplexer (  
4     input  wire enable,  
5     input  wire select,  
6     input  wire [3:0] input_a, input_b,  
7     output reg [3:0] led  
8 );  
9 always @(*) begin  
10     if(enable==1'b1)begin  
11         if(select==1'b1)led=input_a-input_b;  
12         //     else if(select==1'b0)led=input_a+input_b;  
13         //     else led=0;  
14     end  
15     else if(enable==1'b0) led=4'b0000;  
16 end
```



# 时序逻辑元件

---

- 锁存器 (Latch)
  - 或非门构成的基本RS锁存器
  - 与非门构成的基本RS锁存器
  - 门控D锁存器
- 触发器 (Flip-Flop)
- 带附加输入端的边沿触发器
- 触发器类型转换

# 时序逻辑元件

---

- 锁存器 (Latch)
- 触发器 (Flip-Flop)
- 带附加输入端的边沿触发器
- 触发器类型转换

# 边沿触发器

---

- D触发器
- RS触发器
- JK锁存器
- T触发器
- T'触发器
- 带附加输入端的触发器

# 时钟触发器

- 受时钟脉冲控制的触发器称作**时钟触发器**。
- 时钟也称**同步信号**。将多个触发器的时钟端相连，可以控制它们同一时刻动作。

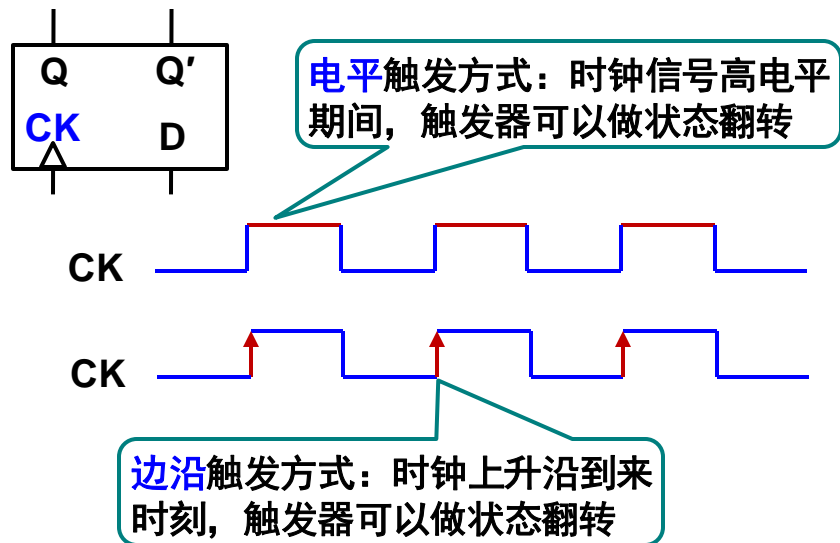
## 时钟触发器分类

### 按逻辑功能

SR触发器  
D触发器  
JK触发器  
T触发器  
T'触发器

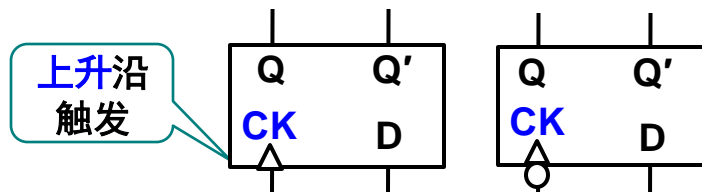
### 按触发方式

电平触发  
**边沿触发**



# 边沿触发器——D触发器

## (1) 逻辑符号



## (2) 功能表（上升沿为例）

时钟端 CK	输入端 D	现态 $Q_n$	次态 $Q_{n+1}$
↑	0	0	0
↑	0	1	0
↑	1	0	1
↑	1	1	1

## (3) 次态方程

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

## 时钟触发器的特点

- ◆由时钟脉冲确定状态转换的时刻（即何时转换？）
- ◆由输入信号确定触发器状态转换的方向（即如何转换？）

上升沿时刻

CK

D

采样D端的数据

Q

必须保证上升沿时刻能采样到正确数据

避免了空翻

## (4) 驱动表

$Q_n$	→	$Q_{n+1}$	D
0	→	0	0
0	→	1	1
1	→	0	0
1	→	1	1

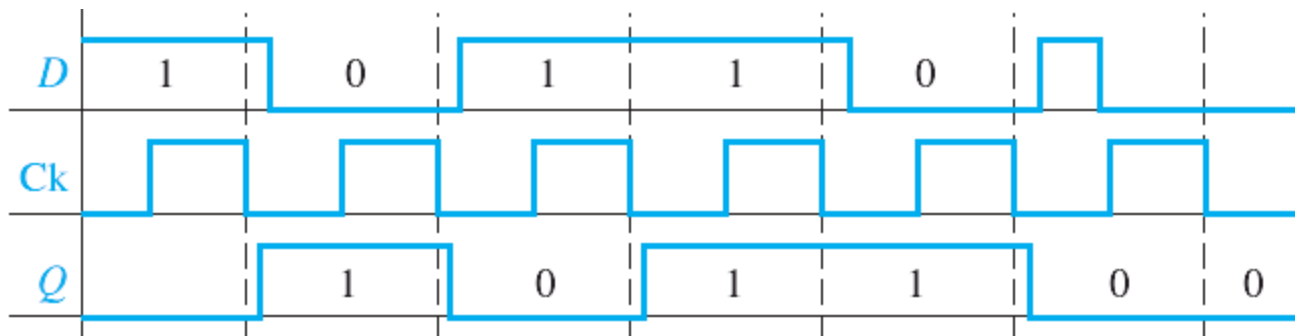
## D触发器的特点：

最简单，应用最广



# 边沿触发器——D触发器

- 为了保证正常工作，输入信号D在**触发时钟边沿前后**保持恒定的值。
- 如果D在**触发边沿的同时**发生改变，最终的行为将变得难以预测。



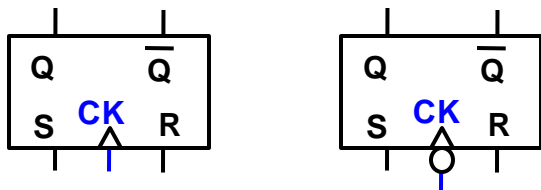
# 用Verilog实现D触发器

---

```
module VrDff(input CLK, input D, output reg Q);  
    always @ (posedge CLK)  
        Q <= D;  
endmodule
```

# 边沿触发器——RS触发器

## (1) 逻辑符号



## (3) 次态方程

$$Q_{n+1} = S + \bar{R}Q_n$$

$$SR = 0 \quad (\text{约束条件})$$

## (2) 功能表（上升沿）

时钟端 CK	输入端 R	输入端 S	现态 $Q_n$	次态 $Q_{n+1}$
↑	0	0	0	0
↑	0	0	1	1
↑	0	1	0	1
↑	0	1	1	1
↑	1	0	0	0
↑	1	0	1	0
↑	1	1	0	—
↑	1	1	1	—

## (4) 驱动表

$Q_n$	→	$Q_{n+1}$	R	S
0	→	0	X	0
0	→	1	0	1
1	→	0	1	0
1	→	1	0	X

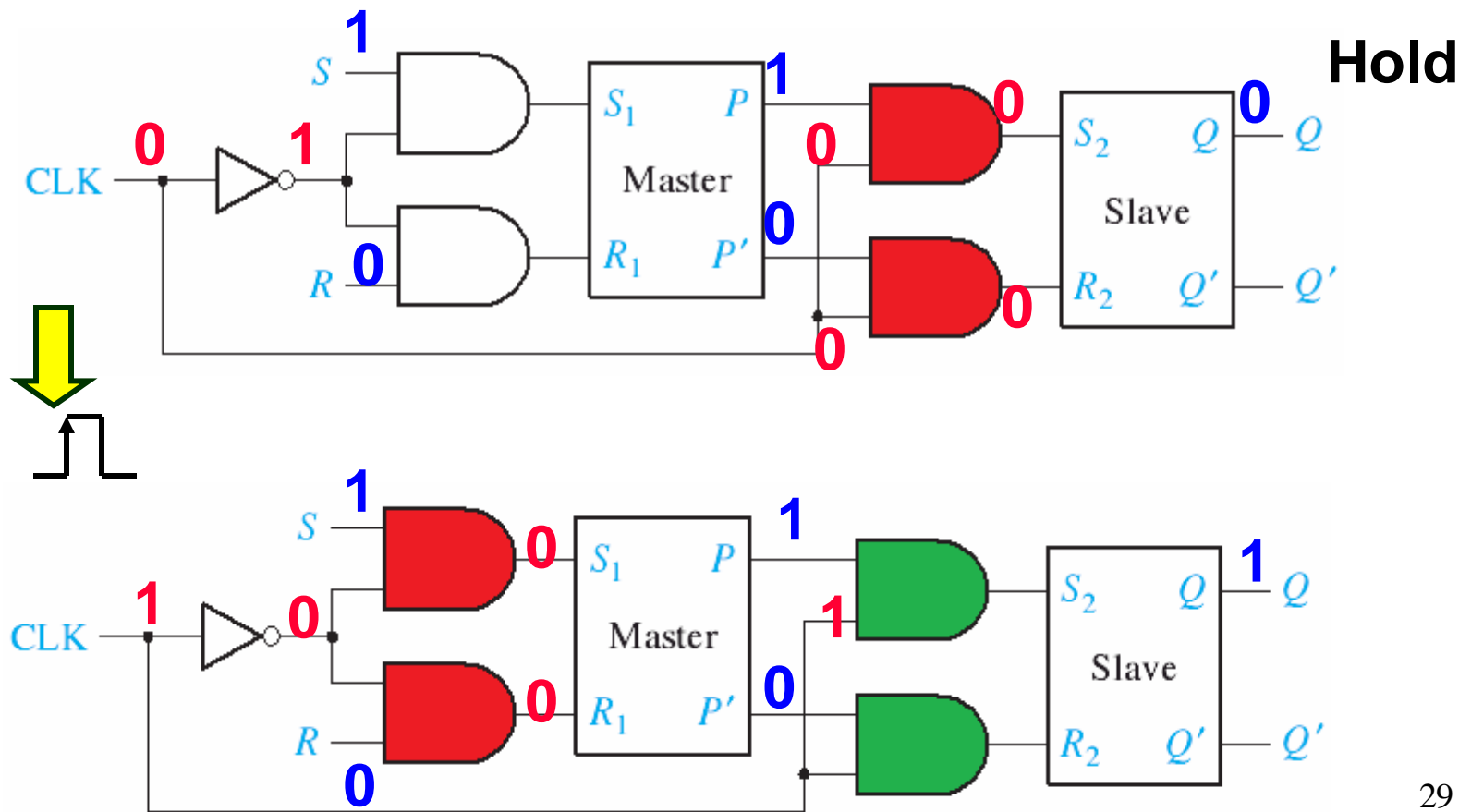
驱动表可以从触发器  
功能推导出来

输入存在约束

RS触发器：输入存在约束

D触发器：没有约束，但是只有一个输入端

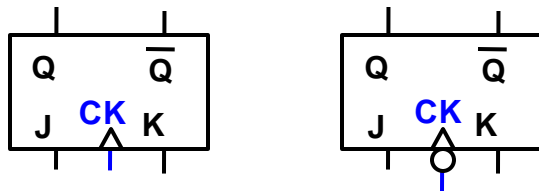
# RS触发器——主从触发器



# 边沿触发器——JK触发器

## (3) 次态方程

### (1) 逻辑符号



### (2) 功能表（下降沿）

时钟端 CK	输入端 J	输入端 K	现态 $Q_n$	次态 $Q_{n+1}$
↓	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	1	0

功能最全，输入没有约束

保持

置0

置1

翻转

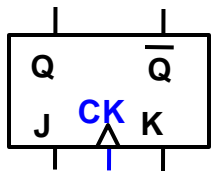
$$Q_{n+1} = J \bar{Q}_n + \bar{K} Q_n$$

JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

### (4) 驱动表

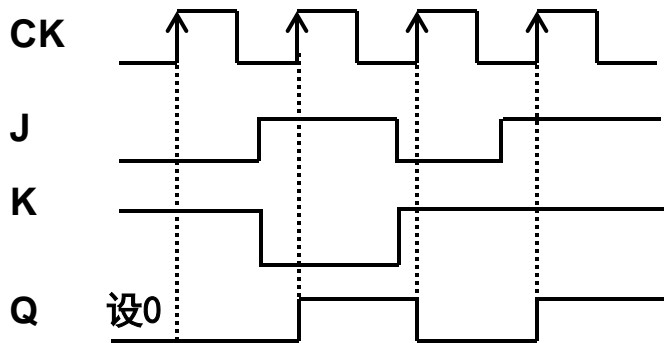
$Q_n \rightarrow Q_{n+1}$	J	K
0 → 0	0	X
0 → 1	1	X
1 → 0	X	1
1 → 1	X	0

# 边沿触发器—— JK触发器



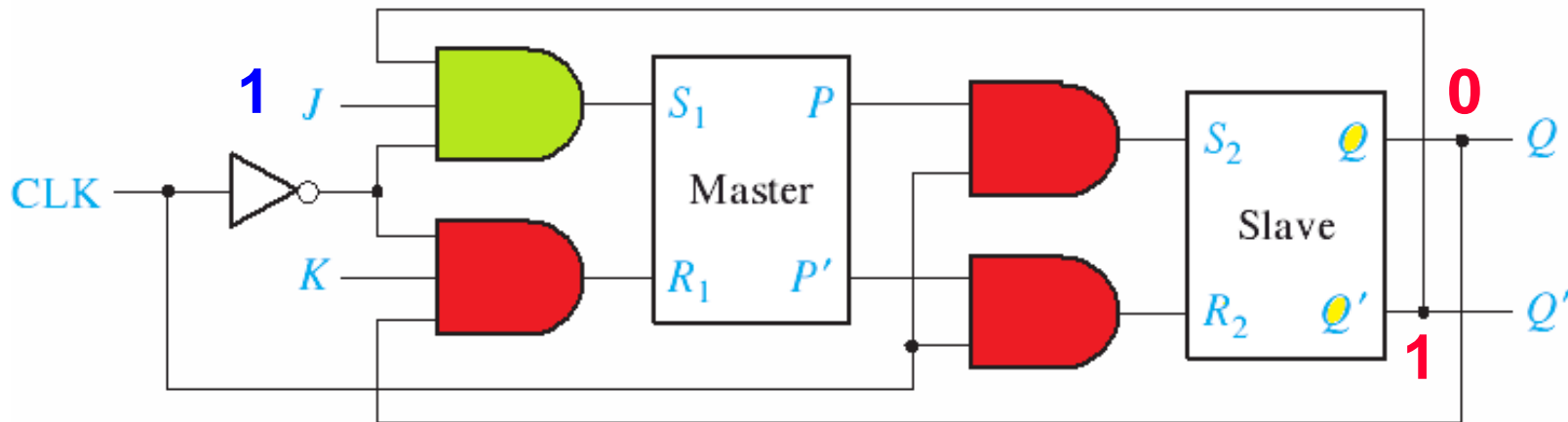
## 时钟边沿触发器

- ◆何时转换？——时钟脉冲有效边沿到来时刻
- ◆如何转换？——输入信号取值确定



输入端		次态
J	K	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\overline{Q}_n$

# JK触发器——主从触发器



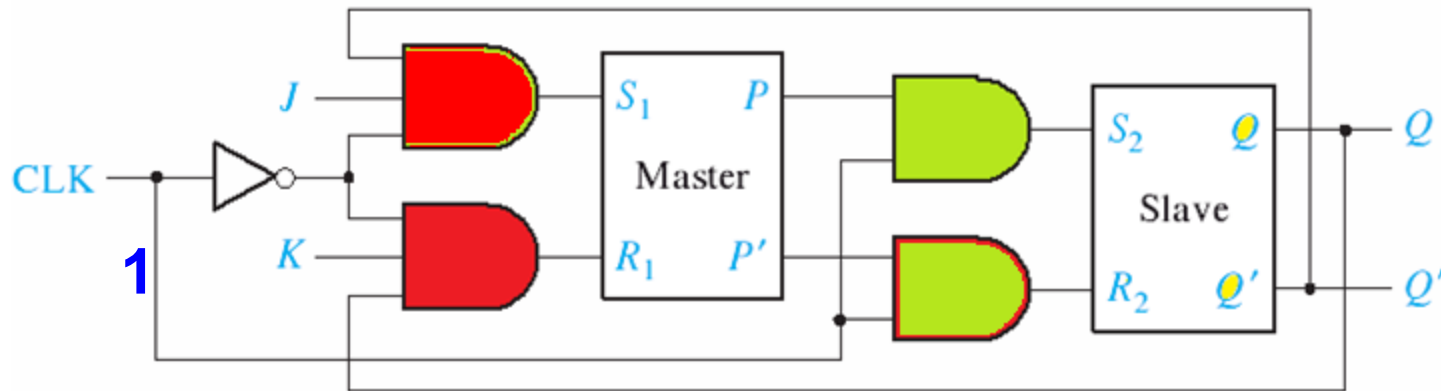
clk=0: If  $Q=0$  and  $J=1$ , then  $S_1=1, R_1=0$  ( $P=1, P'=0$ ), 无论  $K$  取何值.  
If  $Q=1$  and  $K=1$ , then  $S_1=0, R_1=1$  ( $P=0, P'=1$ ), 无论  $J$  取何值.

主触发器的输出取决于从触发器的状态:

从触发器: **0态**; 主触发器: **1态**

从触发器: **1态**; 主触发器: **0态**

# JK触发器——主从触发器



Clk  $\uparrow$  :

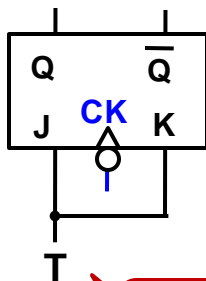
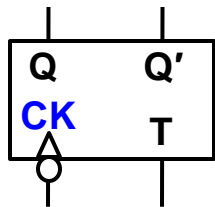
If  $Q=0$  and  $J=1$ :  $P=1, P'=0$ , then  $S_2=1$  and  $R_2=0$ ,  $Q^+=1$

If  $Q=1$  and  $K=1$ :  $P=0, P'=1$ , then  $S_2=0$  and  $R_2=1$ ,  $Q^+=0$



# 边沿触发器——T触发器

## (1) 逻辑符号



是JK触发器的特例

## (2) 功能表（下降沿）

时钟端 CK	输入端 T	现态 $Q_n$	次态 $Q_{n+1}$
↓	0	0	0
↓	0	1	1
↓	1	0	1
↓	1	1	0

保持

翻转

输入端 T	次态 $Q_{n+1}$
0	$Q_n$
1	$\overline{Q_n}$

## (3) 次态方程

$$Q_{n+1} = J \overline{Q_n} + \overline{K} Q_n$$

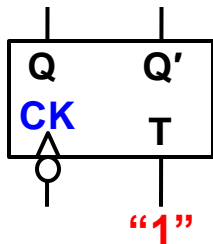
IF  $J=K=T$



$$\begin{aligned} Q_{n+1} &= T \overline{Q_n} + T Q_n \\ &= T \oplus Q_n \end{aligned}$$

# 边沿触发器——T'触发器

## (1) 逻辑符号



是T触发器的特例

## (2) 功能表（下降沿）

时钟端 CK	输入端 T	现态 $Q_n$	次态 $Q_{n+1}$
↓	1	0	1
↓	1	1	0

## (3) 次态方程

$$Q_{n+1} = J \bar{Q}_n + \bar{K} Q_n$$

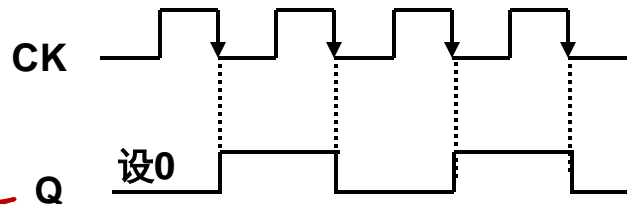


IF  $J=K=T=1$

$$Q_{n+1} = \bar{Q}_n$$

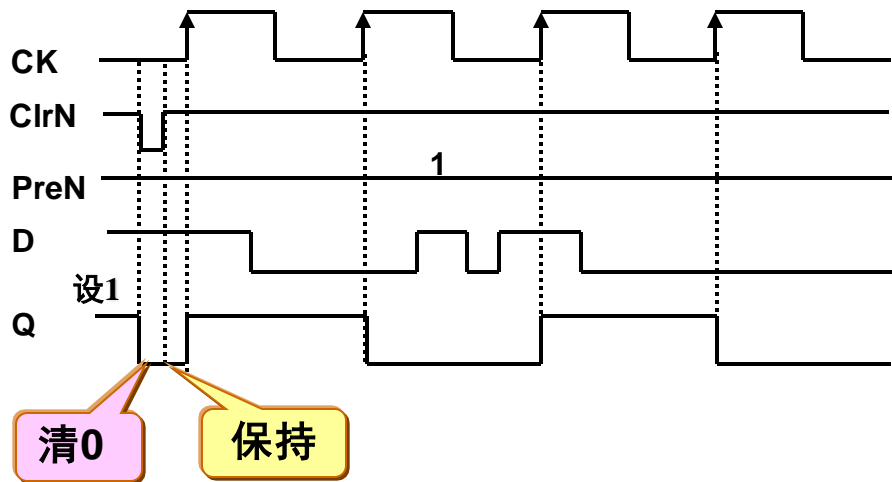
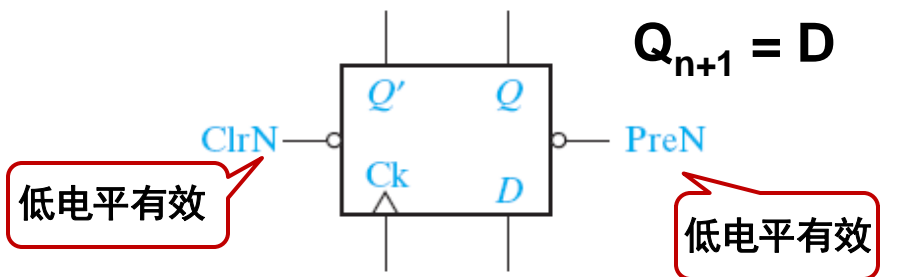
二分频

## (4) 波形分析



# 带附加输入端的边沿触发器

## ■ 带异步清零端和异步置1端



异步：独立于时钟信号

用途：为触发器设置指定状态

时钟端 CK	输入端 D	异步置1端 PreN	异步清零端 ClrN	次态 $Q_{n+1}$
X	X	0	0	不允许
X	X	0	1	1
X	X	1	0	0
↑	0	1	1	0
↑	1	1	1	1
0,1, ↓	X	1	1	$Q_n$

- 被用于将触发器设置成**初始态**，而**不依赖于时钟信号**。

# 用Verilog实现带异步清零端的D触发器

---

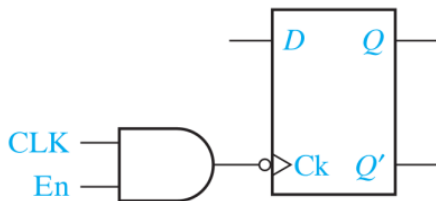
```
module VrDffC(input CLK, input CLRN, input D,  
              output reg Q);  
    always @ (posedge CLK or negedge CLRN)  
        if (CLRN==1'b0) Q <= 0;  
        else Q <= D;  
endmodule
```

# 带附加输入端的边沿触发器—续

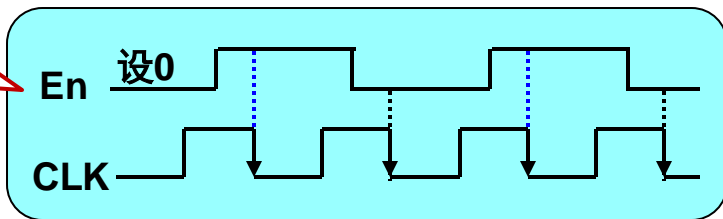
## ■ 带时钟使能端

解决方案:

使能端CE不与时钟端CLK捆绑使用

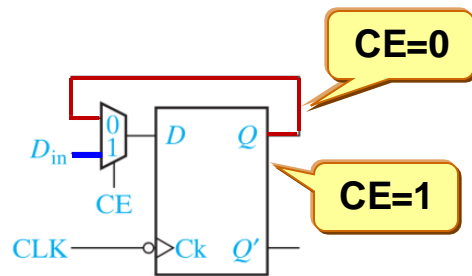
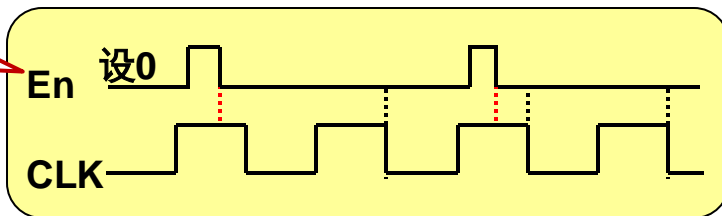


信号给定正确



后果: 失去同步性!

信号给定错误



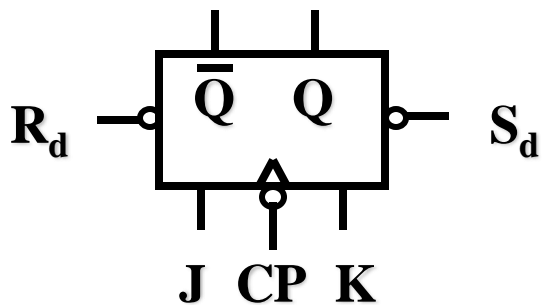
$$Q^+ = Q \cdot CE' + D \cdot CE$$

# 带时钟使能端的D触发器

---

```
module VrDffSE(input CLK, input CE,  
               input D, output reg Q);  
  
  always @ (posedge CLK)  
    if (CE==1) Q <= D;      //时钟使能有效  
    else Q <= Q;  
  
endmodule
```

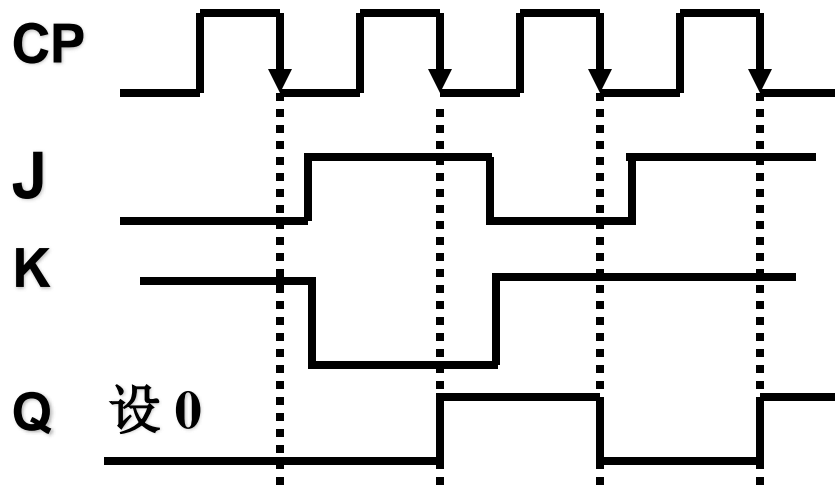
# 带附加输入端的边沿触发器—续



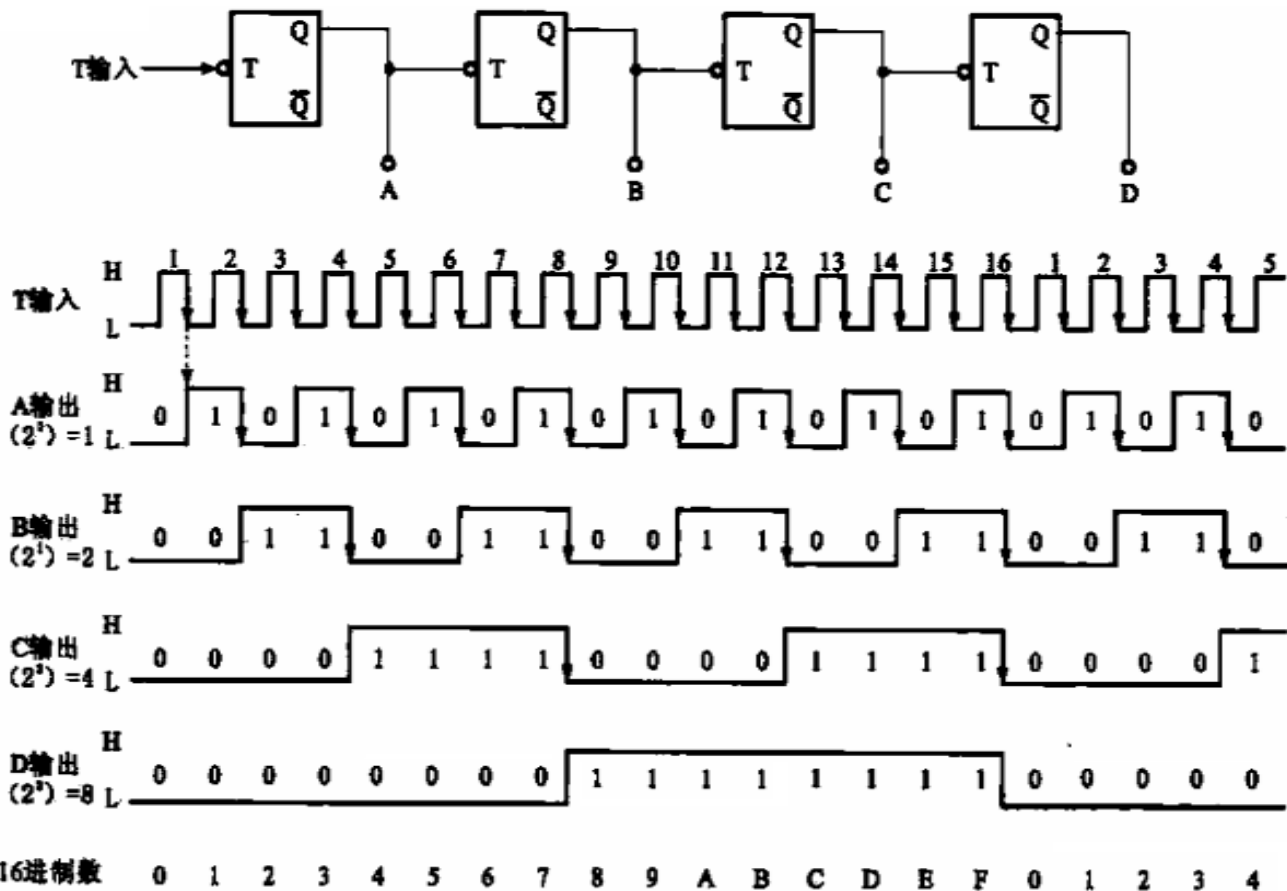
Sd	Rd	CP	J	K	$Q^n$	$Q^{n+1}$
1	0	x	x	x	x	0
0	1	x	x	x	x	1
1	1	↓	0	0	0	0
			0	0	1	1
			0	1	0	0
			0	1	1	0
			1	0	0	1
			1	0	1	1
			1	1	0	1
			1	1	1	0

状态真值表

时序分析:



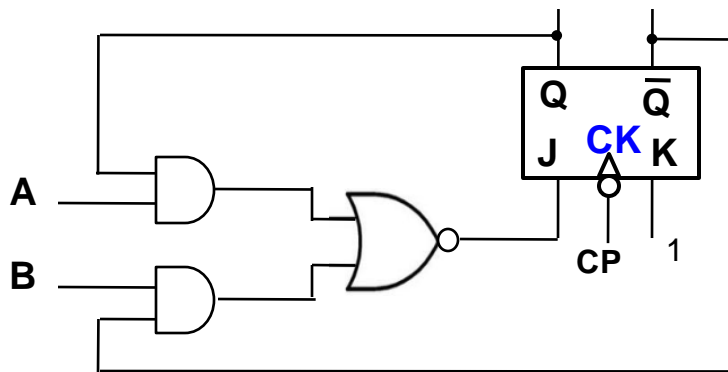
# T触发器的应用—二进制加法计数器





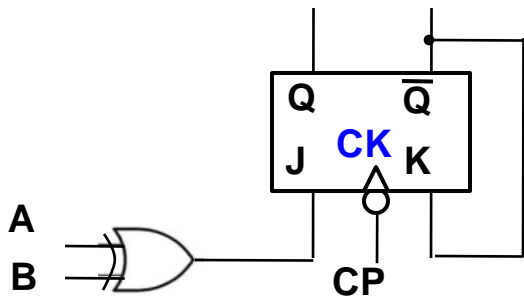
# JK触发器的应用实例

例1：写出JK触发器的次态方程



# JK触发器的应用实例2

例2：画出Q端波形图



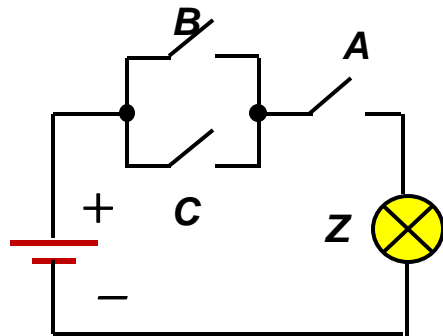
# 触发器的应用——1

## 1. 存储功能的应用——保存瞬态信号，直到清除为止

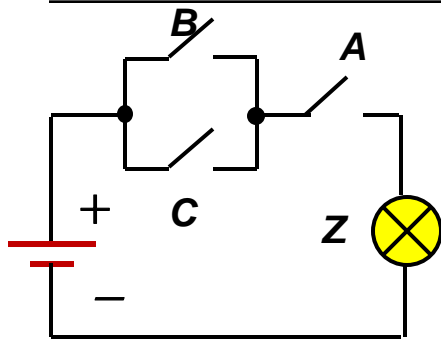
【例】举重裁判逻辑电路V2.0：一个主裁判A和两个副裁判B和C，只有两人以上(必须包含主裁判在内)认定试举动作合格，并按下自己的按钮时，输出信号 $Z=1$ ，该信号一直保持下去，直到工作人员按下清除按钮 P为止。

分析：

- 三个人的按钮动作有先后、长短之别，所以需要3个存储元件分别保存三个按钮信号；
- 存储元件有置1和置0功能即可（锁存器和SR、JK、D触发器均可）



# 触发器的应用——续

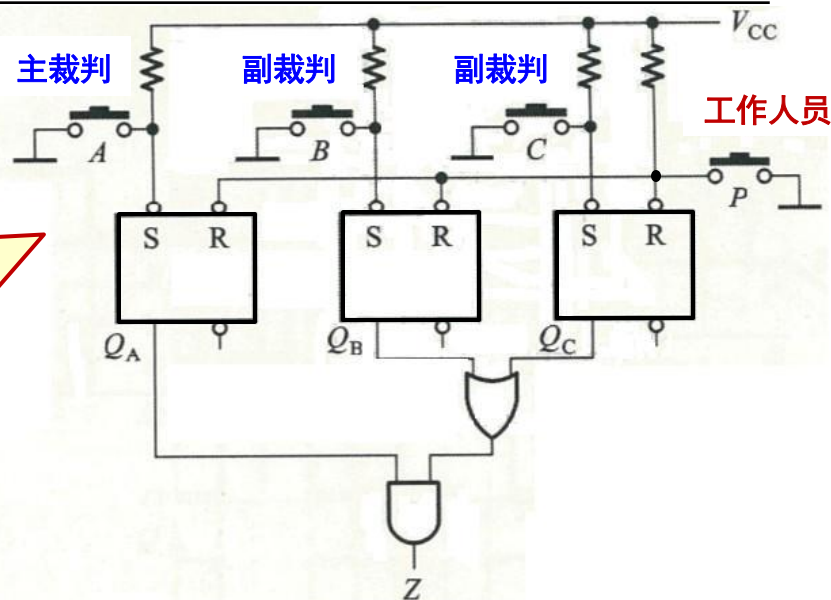


对于每个锁存器：

- 裁判按钮按下，执行**置1**功能；按钮弹起，执行**保持**功能；
- 按钮P按下，全体执行**置0**功能

## 基本RS锁存器

- 锁存器的置1端S：连接主裁按钮A、和副裁B、C输出的低电平
- 锁存器的置0端R：连接工作人员按钮P给出的低电平
- 输出信号Z：三个锁存器输出状态的或与逻辑。



采用JK触发器或D触发器，  
如何实现？哪种方法更简单？

# 三人表决器设计

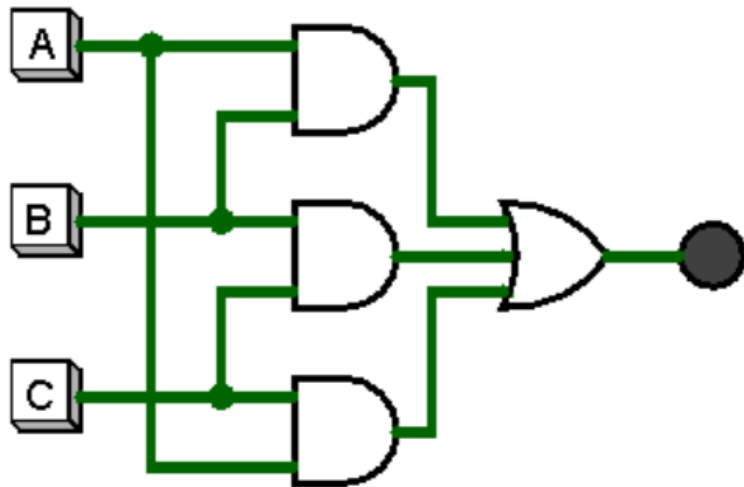
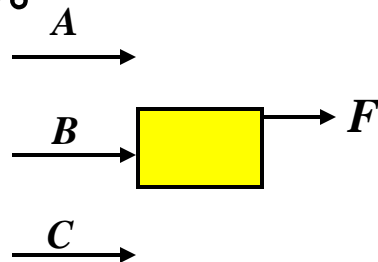
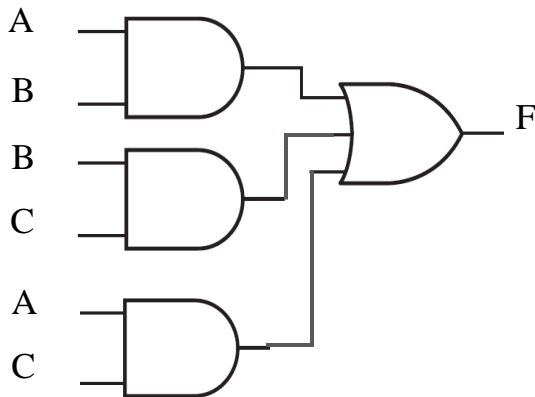
■少数服从多数，结果为多数人的选择。

真值表

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

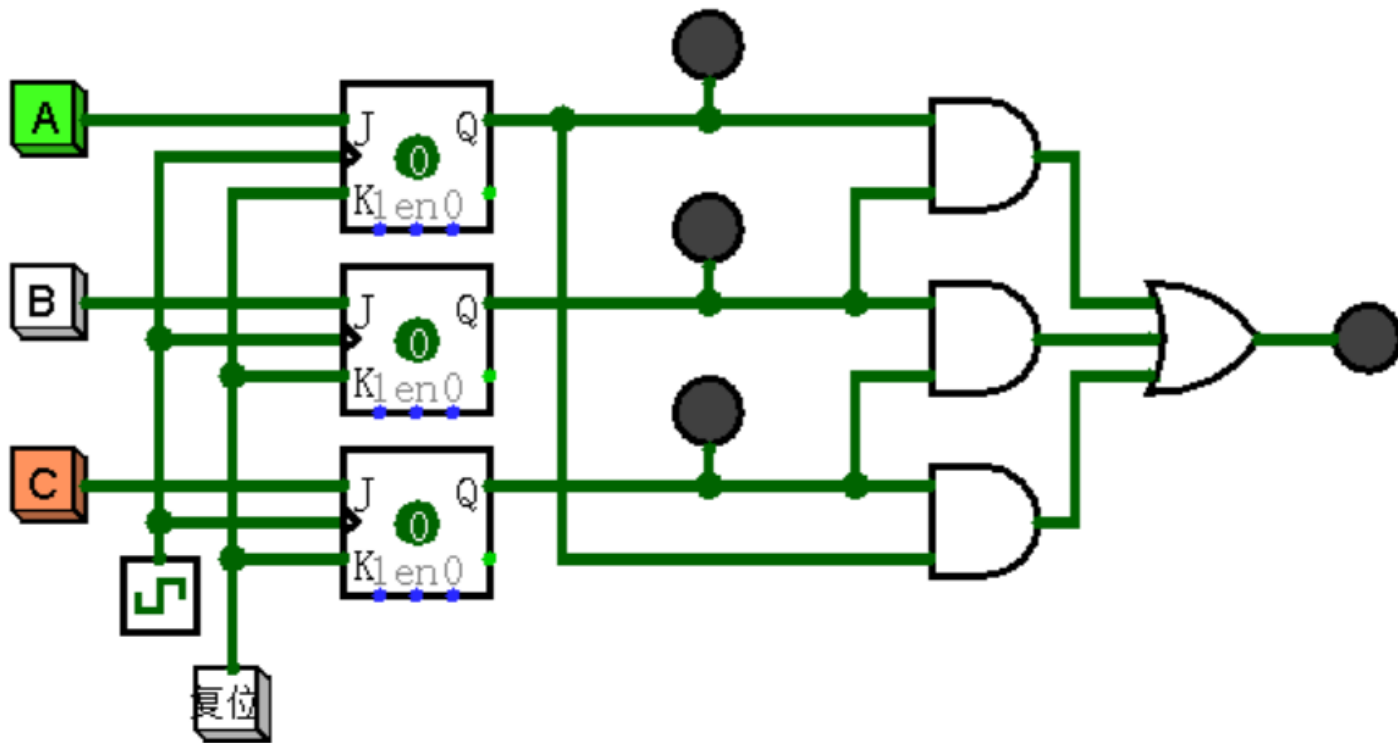
BC	00	01	11	10
A	0	0	1	0
1	0	1	1	1

$$F=AB+AC+BC$$



# 触发器的应用——存储功能

保存瞬态信号，直到清除为止

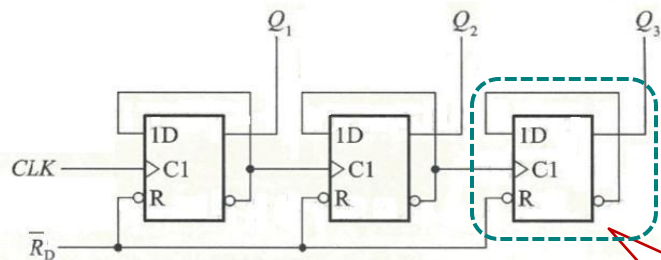


# 触发器的应用——2

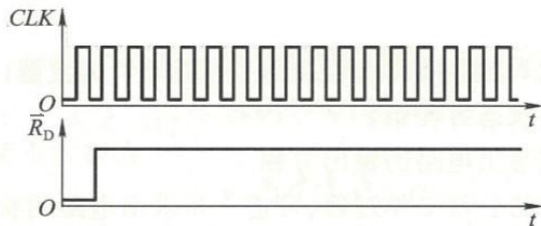
## 2. 分频/计数功能的应用——

利用触发器的置0、置1功能，由多个触发器组成分频电路，对输入的时钟信号进行分频。

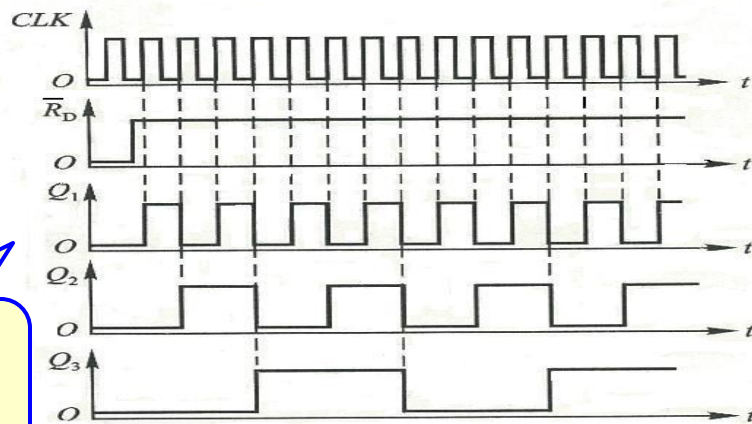
【例】分析输出信号 $Q_1$ 、 $Q_2$ 、 $Q_3$ 与时钟信号CLK之间的频率关系，R为清零端



T'触发器



$Q_1$ 对CLK二分频  
 $Q_2$ 对CLK四分频  
 $Q_3$ 对CLK八分频



$Q_1$ 、 $Q_2$ 、 $Q_3$ 输出波形

# 边沿触发器——总结

## 时钟边沿触发器的特点

- ◆ 由时钟脉冲边沿确定状态转换的时刻(即何时转换?)  
其余时刻都是保持功能
- ◆ 由输入信号确定触发器状态转换的方向(即如何转换?)

思考：对于一个下降沿触发的JK触发器，如果让它实现保持功能，有几种方法可以做到？

### ◆ 方法1：

最简单的方法：不给有效的时钟边沿（此时不用考虑J端和K端的信号）

## 方法

### ◆ 方法2：

给时钟下降沿，此时触发器的保持功能就必须依靠J端和K端的信号配合才能完成



# 边沿触发器

---

- D触发器
- SR触发器
- JK触发器
- T触发器
- T'触发器
- 带附加输入端的触发器

# 时序逻辑元件

---

- 锁存器 (Latch)
- 触发器 (Flip-Flop)
- 带附加输入端的边沿触发器
- 触发器类型转换

# 触发器类型转换——代数法

- 触发器类型主要有5种，用到最多的是D触发器
- 触发器类型可以相互转换（例如，设计中手头没有需要的触发器类型）

转换方法 {

- 代数法
- 卡诺图法

□ 代数法

从次态方程入手

## 1. JK → D、T (T')、RS

### (1) JK → D

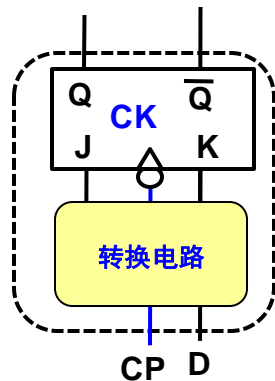
$$\left. \begin{array}{l} \text{JK: } Q_{n+1} = J \bar{Q}_n + \bar{K} Q_n \\ \text{D: } Q_{n+1} = D \end{array} \right\}$$

$$D = J \bar{Q}_n + \bar{K} Q_n$$

$$D(Q_n + \bar{Q}_n) = J \bar{Q}_n + \bar{K} Q_n$$

$$J = f(D, Q)$$

$$K = f(D, Q)$$



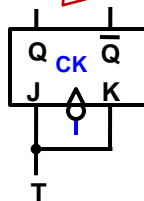
$$\left\{ \begin{array}{l} J=D \\ \bar{K}=D \quad (K=\bar{D}) \end{array} \right.$$

# 触发器类型转换——JK转其他

## (2) JK $\rightarrow$ T

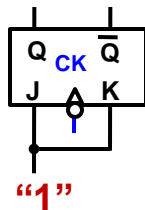
$$\left. \begin{array}{l} \text{JK: } Q_{n+1} = J \bar{Q}_n + \bar{K} Q_n \\ \text{T: } Q_{n+1} = T \bar{Q}_n + \bar{T} Q_n \end{array} \right\} \begin{array}{l} J=T \\ K=T \end{array}$$

JK触发器的特例



$T'$

T触发器的特例



## (3) JK $\rightarrow$ RS

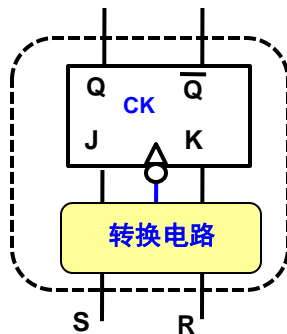
$$\text{JK: } Q_{n+1} = J \bar{Q}_n + \bar{K} Q_n$$

$$\begin{aligned} \text{SR: } Q_{n+1} &= S + \bar{R} Q_n = S(Q_n + \bar{Q}_n) + \bar{R} Q_n \\ &= S Q_n + S \bar{Q}_n + \bar{R} Q_n \\ &= S Q_n (R + \bar{R}) + S \bar{Q}_n + \bar{R} Q_n \\ &= R S Q_n + \bar{R} S Q_n + S \bar{Q}_n + \bar{R} Q_n \\ &= R S Q_n + S \bar{Q}_n + \bar{R} Q_n \end{aligned}$$

RS=0

$$J = f(R, S, Q)$$

$$K = f(R, S, Q)$$



$$J=S$$

$$K=R$$

# 触发器类型转换——D转其他

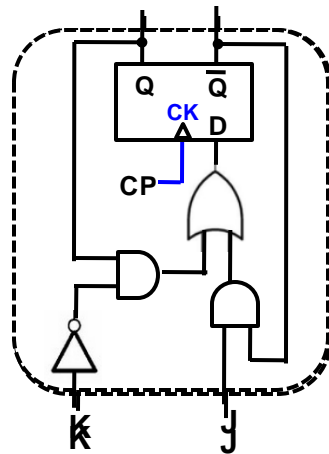
## 2. $D \rightarrow JK$ 、 $T$ ( $T'$ )、 $RS$

### (1) $D \rightarrow JK$

$$\left. \begin{array}{l} JK: \quad Q_{n+1} = J \bar{Q}_n + \bar{K} Q_n \\ D: \quad Q_{n+1} = D \end{array} \right\}$$

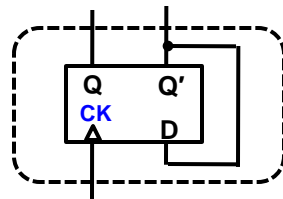
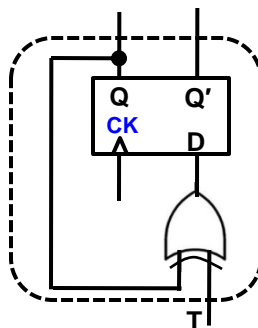
$$D = f(J, K, Q)$$

$$D = J \bar{Q}_n + \bar{K} Q_n$$



### (2) $D \rightarrow T$ ( $T'$ )

$$\left. \begin{array}{l} T: \quad Q_{n+1} = T \oplus Q_n \\ D: \quad Q_{n+1} = D \\ T': \quad Q_{n+1} = \bar{Q}_n \end{array} \right\} \quad D = T \oplus Q_n$$



### (3) $D \rightarrow RS$ ?

# 触发器类型转换——D转其他

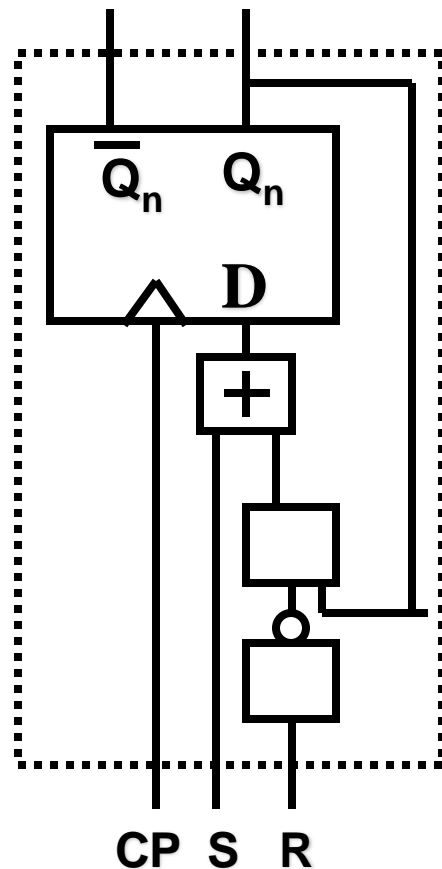
## 2. $D \rightarrow JK$ 、 $T$ ( $T'$ )、 $RS$

### (3) $D \rightarrow RS$

D:  $Q_{n+1} = D$

RS:  $Q_{n+1} = S + \bar{R} Q_n$

$\rightarrow D = S + \bar{R} Q_n$



# 触发器类型转换——卡诺图法

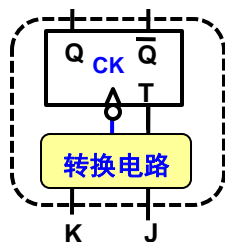
## □ 卡诺图法

### 1. $T \rightarrow JK$ 、 $D$ 、 $RS$

#### (1) $T \rightarrow JK$

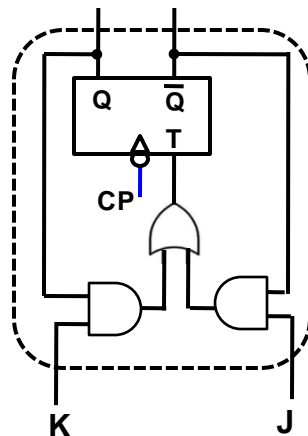
$Q_n \rightarrow Q_{n+1}$	$T$	J	K
0 $\rightarrow$ 0	0	0	X
0 $\rightarrow$ 1	1	1	X
1 $\rightarrow$ 0	1	X	1
1 $\rightarrow$ 1	0	X	0

$$T = f(J, K, Q)$$



$$T = J\bar{Q}_n + KQ_n \rightarrow$$

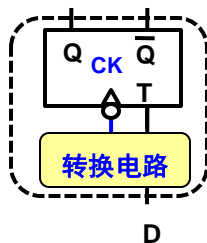
$Q_n \backslash JK$	00	01	11	10
0	0	0	1	1
1	0	1	1	0



#### (2) $T \rightarrow D$

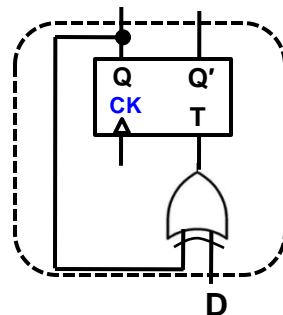
$Q_n \rightarrow Q_{n+1}$	$T$	$D$
0 $\rightarrow$ 0	0	0
0 $\rightarrow$ 1	1	1
1 $\rightarrow$ 0	1	0
1 $\rightarrow$ 1	0	0

$$T = f(D, Q)$$



$$T = D \oplus Q_n \rightarrow$$

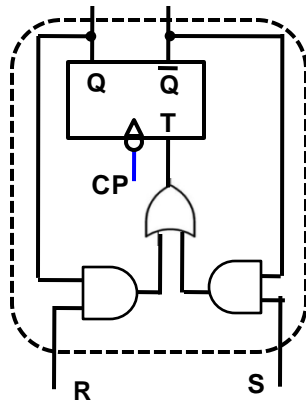
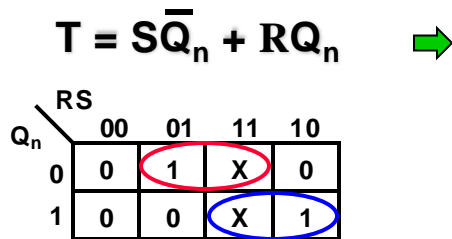
$Q_n \backslash D$	0	1
0	0	1
1	1	0



# 触发器类型转换——卡诺图法

## (3) $T \rightarrow RS$

$Q_n$	$\rightarrow$	$Q_{n+1}$	T	R	S
0	$\rightarrow$	0	0	X	0
0	$\rightarrow$	1	1	0	1
1	$\rightarrow$	0	1	1	0
1	$\rightarrow$	1	0	0	X





# 触发器类型转换——卡诺图法

## 2. RS $\rightarrow$ JK、D、T(T')?

### (1) RS $\longrightarrow$ JK

$Q_n \rightarrow Q_{n+1}$	R S	J	K
0 $\rightarrow$ 0	$\times$ 0	0	$\times$
0 $\rightarrow$ 1	0 1	1	$\times$
1 $\rightarrow$ 0	1 0	$\times$	1
1 $\rightarrow$ 1	0 $\times$	$\times$	0

S

$Q_n \backslash JK$	00	01	11	10
0	0	0	1	1
1	$\times$	0	0	$\times$

R

$Q_n \backslash JK$	00	01	11	10
0	$\times$	$\times$	0	0
1	0	1	1	0

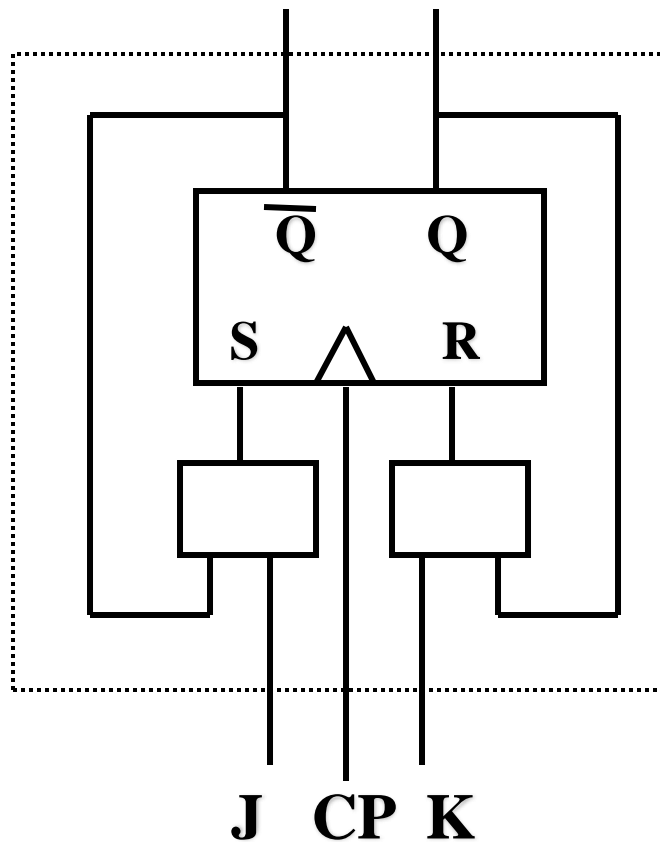
$$\begin{cases} R = K Q_n \\ S = J \bar{Q}_n \end{cases}$$

# 触发器类型转换——卡诺图法

## 2. RS $\rightarrow$ JK、D、T(T')

(1) RS  $\longrightarrow$  JK

$$\begin{cases} R = K Q_n \\ S = J \bar{Q}_n \end{cases}$$



# 触发器类型转换——卡诺图法

(2) RS  $\longrightarrow$  D

$Q_n \longrightarrow Q_{n+1}$	R S	D
0 $\longrightarrow$ 0	$\times$ 0	0
0 $\longrightarrow$ 1	0 1	1
1 $\longrightarrow$ 0	1 0	0
1 $\longrightarrow$ 1	0 $\times$	1

		R	
		$Q_n$	
D	$Q_n$	0	1
	0	$\times$	1
	1	0	0

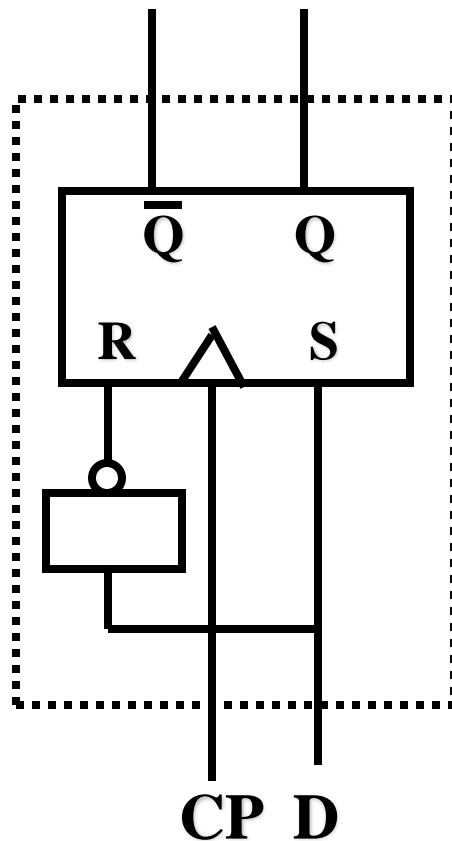
		$Q_n$	
		0	1
S	D	0	0
	1	1	$\times$

$$\begin{cases} S = D \\ R = \overline{D} \end{cases}$$

# 触发器类型转换——卡诺图法

(2) RS  $\longrightarrow$  D

$$\begin{cases} S = D \\ R = \bar{D} \end{cases}$$



# 触发器类型转换——卡诺图法

(3) RS  $\longrightarrow$  T?

$Q_n \longrightarrow Q_{n+1}$	R S	T
0 $\longrightarrow$ 0	$\times$ 0	0
0 $\longrightarrow$ 1	0 1	1
1 $\longrightarrow$ 0	1 0	1
1 $\longrightarrow$ 1	0 $\times$	0

		$Q_n$	
		0	1
S	T	0	$\times$
	1	1	0

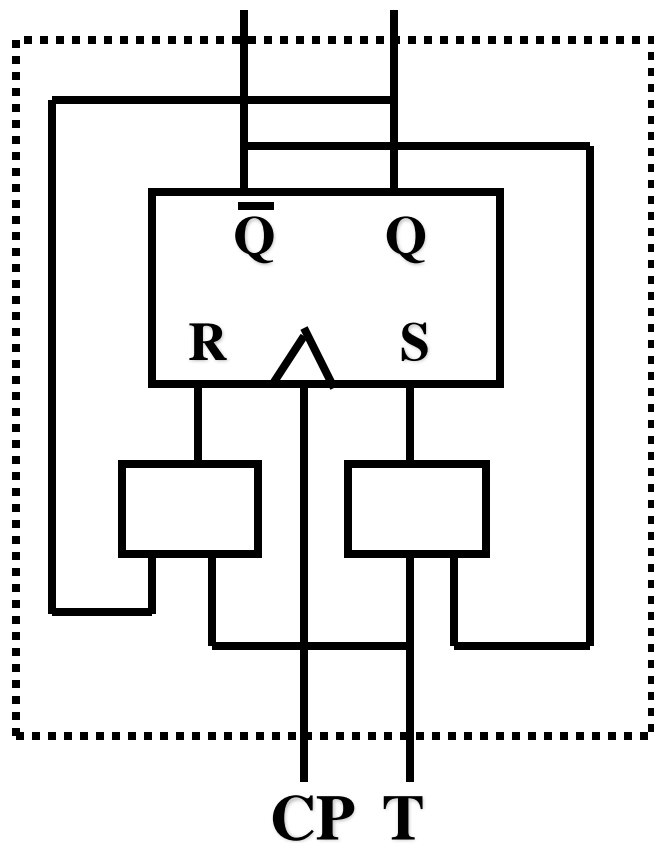
		R	
		0	1
T	$Q_n$	$\times$	0
	1	0	1

$$\left\{ \begin{array}{l} R = T Q_n \\ S = T \bar{Q}_n \end{array} \right.$$

# 触发器类型转换——卡诺图法

(3) RS  $\longrightarrow$  T

$$\begin{cases} R = T Q_n \\ S = T \bar{Q}_n \end{cases}$$



# 时序逻辑元件

---

- 锁存器 (Latch)
- 触发器 (Flip-Flop)
- 带附加输入端的边沿触发器
- 触发器类型转换

