

# 第六讲

---

# 前三讲简要回顾

## ■ 布尔代数：分析与设计数字系统的重要理论工具

- 逻辑代数基本概念：逻辑常量/变量，典型逻辑运算
- 逻辑代数的运算法则：公理、定律、定理、基本公式及其推论
- 逻辑函数的表达式：真值表  $\rightarrow$  最小项表达式、最大项表达式
- 逻辑函数的简化法：合并乘积项法、吸收项法、配项法

## ■ 硬件描述语言：Verilog HDL（自学）

- Verilog HDL的模块、词法、常用语句
- 不同抽象级别的Verilog HDL模型：行为描述和结构描述

## ■ 逻辑门电路：数字电路中的基本逻辑单元电路

- 由晶体管和MOS管（晶体二极管、晶体三极管、NMOS、PMOS）构建门电路（与、或、非、与非、或非等）

## ■ 基本组合逻辑部件设计

- 运算单元电路：加法、减法、乘法、比较器、ALU
- 编码器/译码器：三种编码器/译码器
- 多路选择器：数据选择、多功能运算

## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. **RS**锁存器/**D**锁存器
2. **D**触发器
3. **JK**触发器

### 二. 有限状态机

1. **Moore**型有限状态机
2. **Mealy**型有限状态机

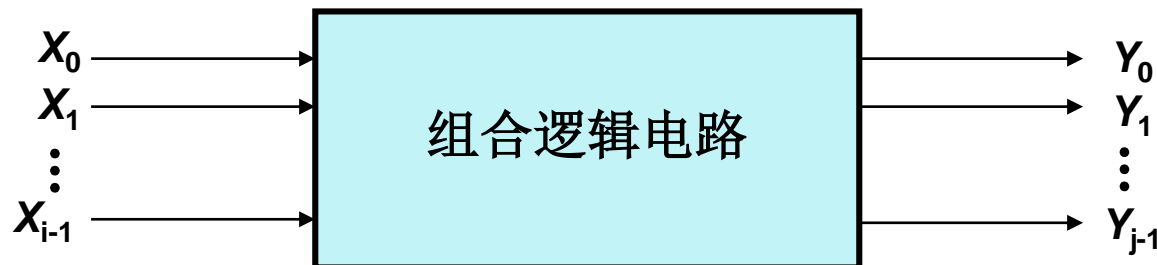
### 三. 时序逻辑电路设计分析

1. 数据寄存器
2. 移位寄存器
3. 计数器

# 组合逻辑电路的不足

## ❖ 组合逻辑电路的特点

- 电路输出端的状态完全由输入端的状态决定，不受系统中时钟脉冲的控制
- 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

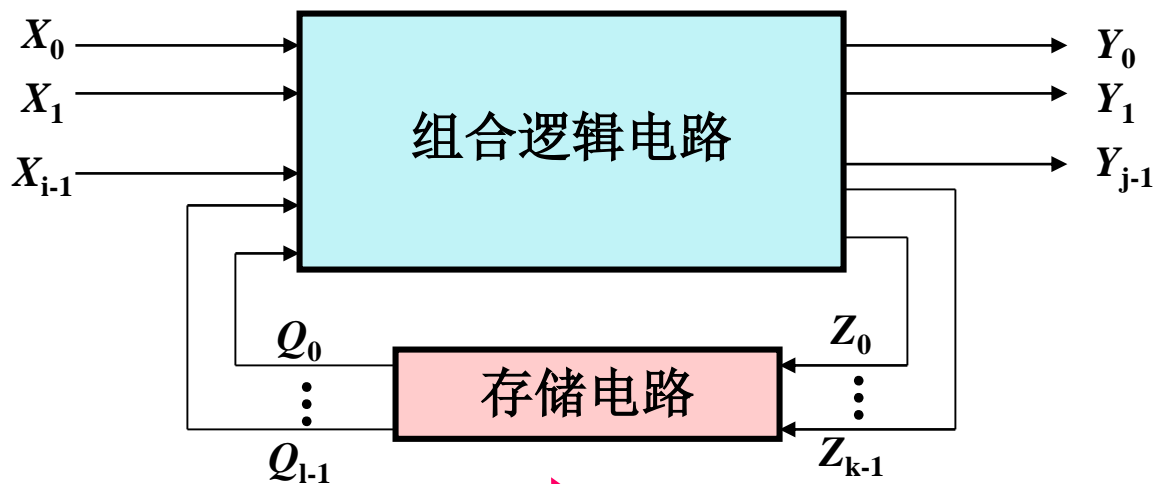


❖ 在数字系统中，有时需要将参与（算术或逻辑）运算的数据和运算结果保存起来 —— 在组合逻辑电路的输出端需要具有记忆功能的部件

❖ **触发器（Flip-Flop, FF）** 就是构成记忆功能部件的基本单元，或者说是**实现存储（记忆）功能的基本单元电路**。

## 时序逻辑电路的特点

- ❖ 时序逻辑电路：当时的输出由当时的输入和电路的**原来状态**共同决定 —— 具有“**记忆**”功能
- ❖ 结构特点：由组合逻辑电路和存储电路构成



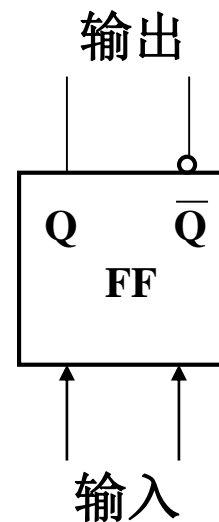
触发器 (Flip-Flop, FF) 或寄存器

# 触发器的概念和特点

❖ **触发器**：一种有记忆功能的逻辑单元电路，是构成时序逻辑电路的基本器件

❖ **触发器的特点**：

- 1、有两个互非的输出 $Q$ 和 $\bar{Q}$ ， $Q$ 称为**状态变量**  
当 $Q = 0 (\bar{Q} = 1)$ 时称为0态，  
当 $Q = 1 (\bar{Q} = 0)$ 时称为1态。
- 2、无外加信号时，触发器保持原有状态（原态）不变，  
 $n$ 级触发器可以记忆 $n$ 位二进制信息的 $2^n$ 种状态。
- 3、在外加信号作用（触发）时，可以改变原态（具有置0或置1功能）。  
 $Q^n(\text{原态}) \rightarrow Q^{n+1}(\text{次态})$

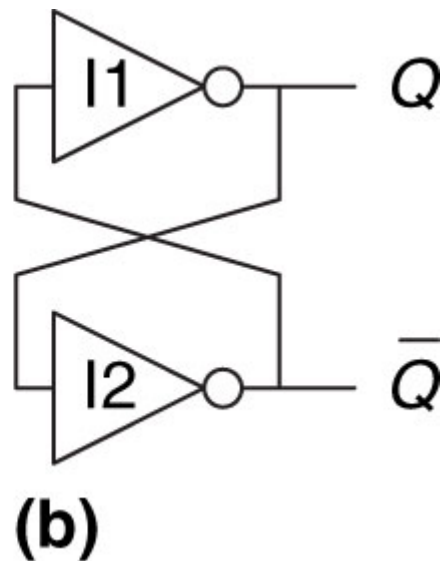
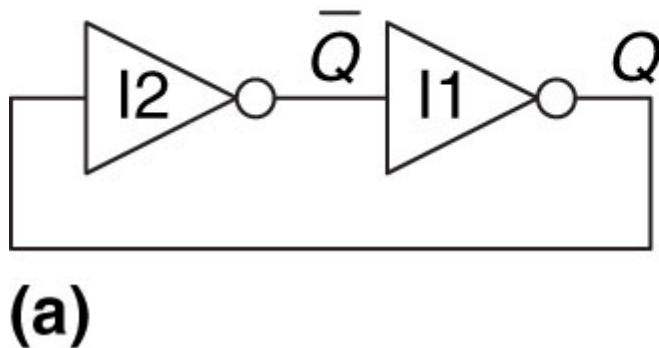


电路符号

# 双稳态电路

## ❖ 双稳态电路

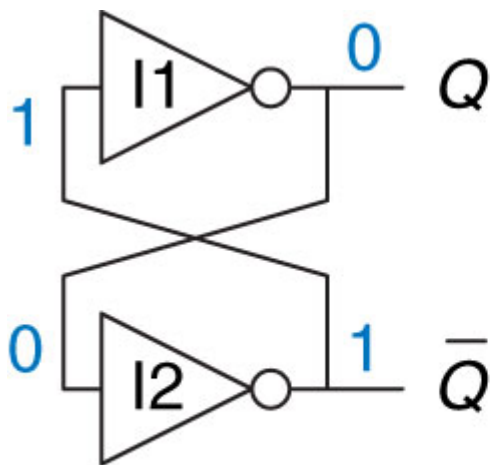
- 可以有两种稳定状态的电路
- 可以用2个反相器构成
- 图(b)电路是图(a)电路的变形



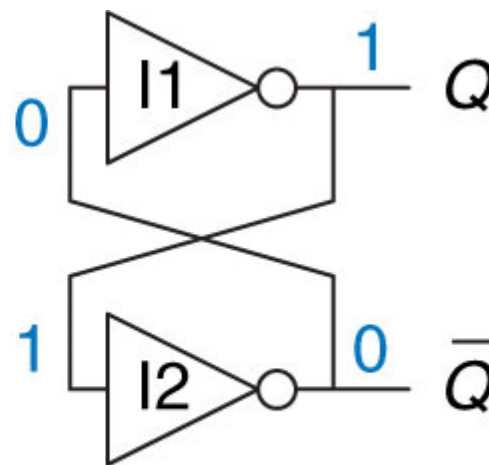
# 双稳态电路

## ❖ 双稳态电路

- 可输出两种稳定的状态，储存1位二进制信息
- 但是有两个问题：
  - ✓ 首次加电，初值未知，Q未知
  - ✓ 缺少输入，状态无法变换，没有实际价值



(a)



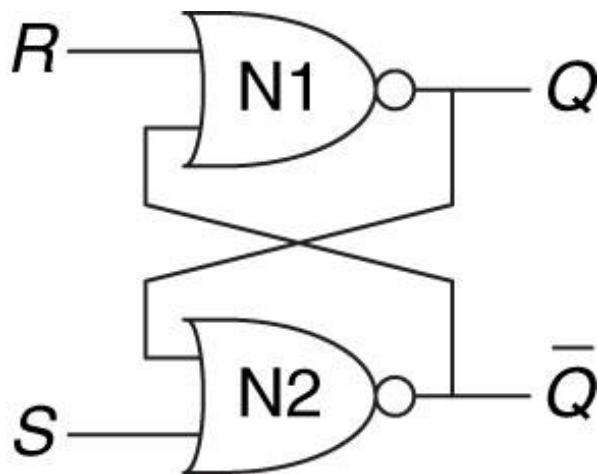
(b)



# 双稳态电路

## ❖ 双稳态电路

- 有两种稳定的状态的电路，可储存1位二进制信息
- 在外加信号触发下，电路可从一种稳定的工作状态转换到另一种稳定的工作状态



两个或非门交叉连接

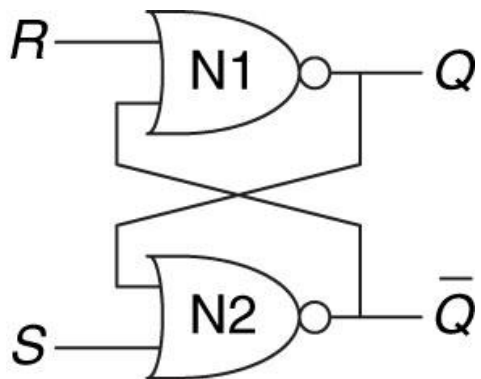
R	S	$Q^n$	$/Q^n$	$Q^{n+1}$	$/Q^{n+1}$
1	0	0/1	1/0	0	1
0	1	0/1	1/0	1	0
0	0	0	1	0	1
0	0	1	0	1	0
<del>1</del>	<del>1</del>	0/1	1/0	0	0

不允许Q和/Q同值

# 双稳态电路

## ❖ 符合触发器的特点:

- 1、有两个互非的输出 $Q$ 和 $\bar{Q}$ ，  
当 $Q = 0$  ( $\bar{Q} = 1$ ) 时称为 0 态，  
当 $Q = 1$  ( $\bar{Q} = 0$ ) 时称为 1 态。
- 2、无外加信号时，触发器保持原有状态（原态）不变，  
 $n$ 级触发器可以记忆 $n$ 位二进制信息的  $2^n$  种状态。
- 3、在外加信号作用（触发）时，可以改变原态  
（具有置 0 或置 1 功能）。  
 $Q^n(\text{原态}) \rightarrow Q^{n+1}(\text{次态})$



R	S	$Q^n$	$\bar{Q}^n$	$Q^{n+1}$	功能
1	0	0/1	1/0	0	置0
0	1	0/1	1/0	1	置1
0	0	0	1	0	保持
0	0	1	0	1	保持
1	1	0/1	1/0	0	非法

# 基本RS锁存器

- ❖ 基本RS锁存器可以自行保持输出状态，是各种触发器的基本构成部分
- ❖ 基本RS锁存器可用与非门或者或非门构成
- ❖ RS: Reset(置0) / Set (置1)
- ❖ 功能

(1) 保持功能  $\overline{R}_D = 1, \overline{S}_D = 1$

触发器保持原来的状态不变

(2) 置0功能  $\overline{R}_D = 0, \overline{S}_D = 1$

触发器的次态变为0

(3) 置1功能  $\overline{R}_D = 1, \overline{S}_D = 0$

触发器的次态变为1

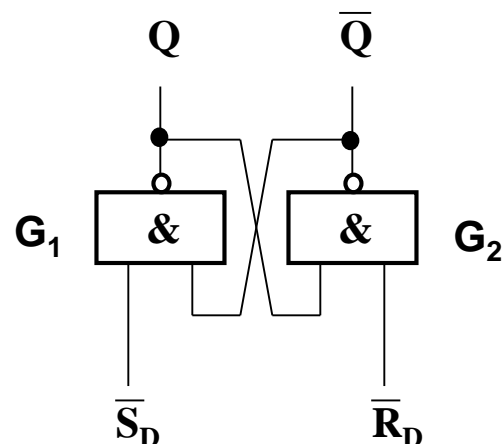
❖ 约束条件  $\overline{R}_D + \overline{S}_D = 1$

$\overline{R}_D, \overline{S}_D$  不能同时为0

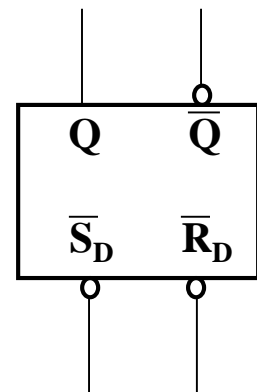
非号，低有效

**D: Direct**

输入信号直接控制锁存器的输出



电路结构



逻辑符号

## 锁存器逻辑功能的表示方法 —— 功能表和特性表

- ❖ 锁存器的逻辑功能可以用功能表、特性表（真值表）、特性方程（函数表达式）、状态转换图、时序图等表示

(1) 功能表

$\overline{S}_D \overline{R}_D Q^n$	$Q^{n+1}$	功能
1 1 0	0	保持
1 1 1	1	
1 $\overline{1}$ 0	0	置0
1 $\overline{1}$ 1	0	
$\overline{1}$ 1 0	1	置1
$\overline{1}$ 1 1	1	
0 0 0	X	不确定
0 0 1	X	

(2) 真值表(特性表)

$\overline{S}_D$	$\overline{R}_D$	$Q^n$	$Q^{n+1}$
1	1	0	0
1	1	1	1
1	0	0	0
1	0	1	0
0	1	0	1
0	1	1	1
0	0	0	x
0	0	1	x

- ❖ 将原态也作为一个变量列入了真值表，将这种含有状态变量的真值表称为特性表
- ❖ 特性表：电路输出次态与原态以及输入之间功能关系的表格

# 锁存器逻辑功能的表示方法 —— 特性方程

**(3) 特性方程：**反映锁存器次态与原态以及输入之间功能关系的函数表达式。

➤由特性表，利用最小项推导法推导得出：

$$\begin{aligned} Q^{n+1} &= \overline{S}_D \overline{R}_D \overline{Q}^n + \overline{S}_D \overline{R}_D Q^n + \overline{S}_D \overline{R}_D Q^n \\ &= \overline{S}_D \overline{R}_D + \overline{S}_D \overline{R}_D Q^n = S_D \overline{R}_D + \overline{S}_D \overline{R}_D Q^n \end{aligned}$$

➤利用约束条件化简，得到特性方程：

$$\begin{aligned} Q^{n+1} &= S_D \overline{R}_D + \overline{S}_D \overline{R}_D Q^n \\ &= (S_D \overline{R}_D + \overline{S}_D \overline{R}_D) (S_D \overline{R}_D + Q^n) \\ &= \overline{R}_D (S_D \overline{R}_D + Q^n) \\ &= S_D \overline{R}_D + \overline{R}_D Q^n \\ &= S_D \overline{R}_D + S_D R_D + \overline{R}_D Q^n \quad (\text{约束条件 } S_D R_D = 0) \\ &= S_D + \overline{R}_D Q^n \end{aligned}$$

真值表(特性表)

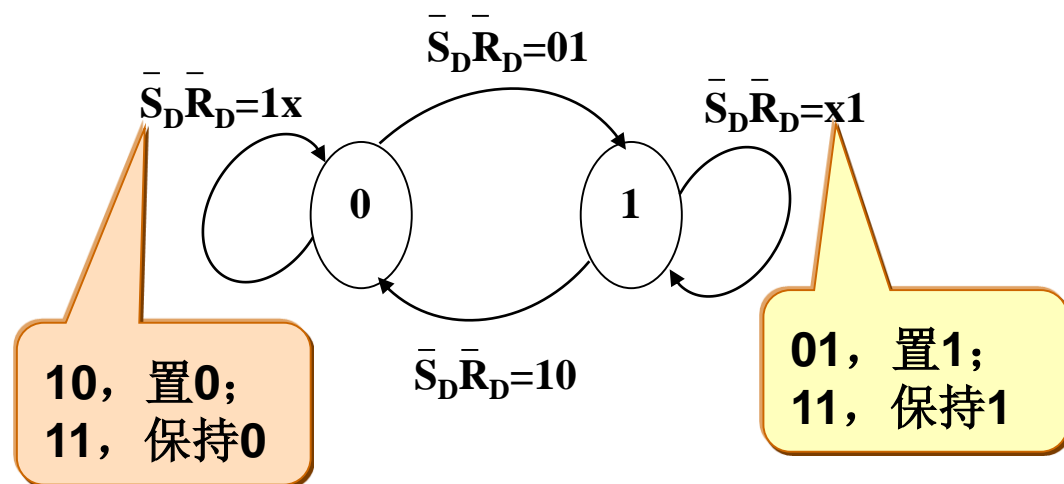
$\overline{S}_D$	$\overline{R}_D$	$Q^n$	$Q^{n+1}$
0	0	0	x
0	0	1	x
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

约束条件  $\overline{R}_D + \overline{S}_D = 1$

# 锁存器逻辑功能的表示方法 —— 状态转换图

(4) 状态转换图：简称**状态图**，是用来表示触发器状态变化（转移）的图形

- 用圆圈表示**0**或**1**状态，用带箭头的线表示状态变化的方向，线上的数据表示状态变化需要的输入条件
- 状态图是触发器、时序逻辑电路特有的描述方法

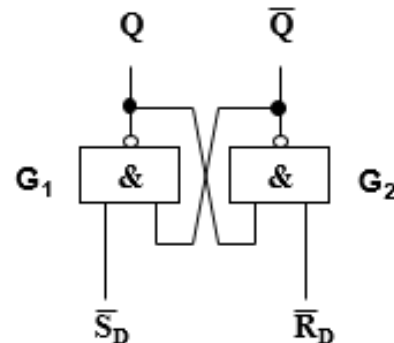
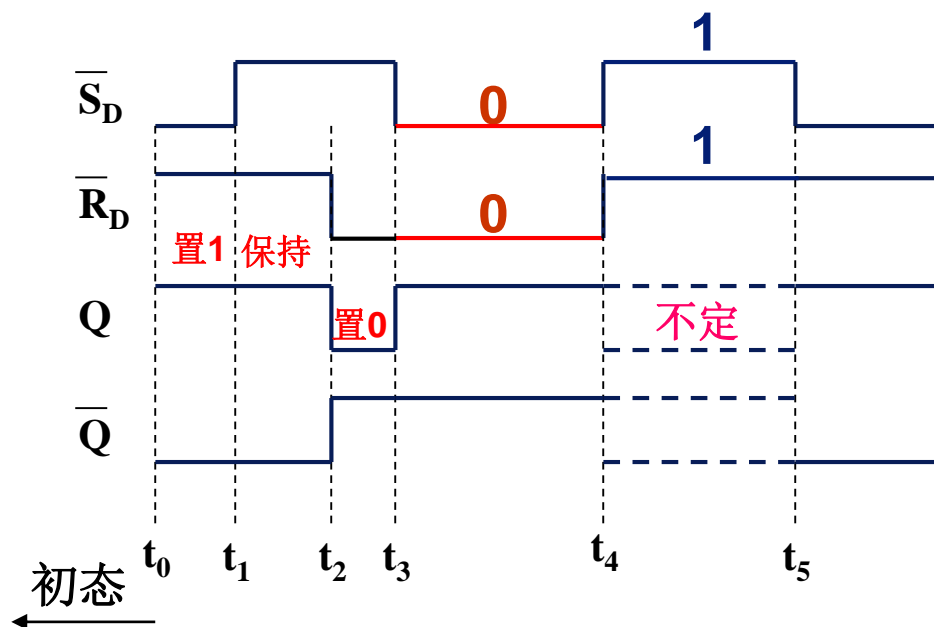


真值表(特性表)

$\bar{S}_D$	$\bar{R}_D$	$Q^n$	$Q^{n+1}$
0	0	0	x
0	0	1	x
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

## 锁存器逻辑功能的表示方法 —— 时序图

(5) 时序图：输出随输入变化的波形（初态=0）



真值表(特性表)

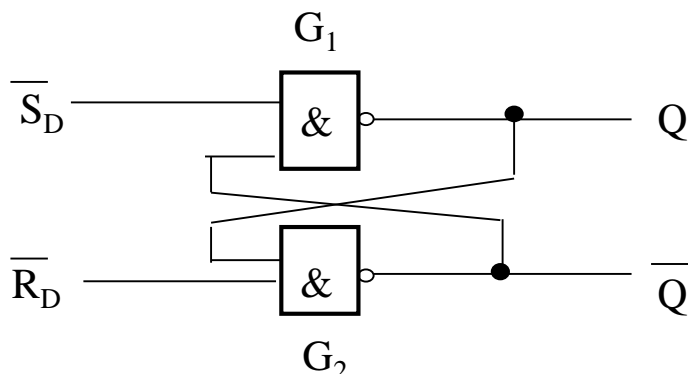
$\overline{S}_D$	$\overline{R}_D$	$Q^n$	$Q^{n+1}$
0	0	0	x
0	0	1	x
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

- ❖ 当两个输入为“0”后，同时变为“1”时，因门传输延迟的不同而产生竞争，使输出状态不确定
- ❖ 故两个输入端不允许同时为“0”!

# 基本RS锁存器的HDL设计

## ❖ 结构描述方式

- 根据电路结构，写出输出信号的逻辑表达式；
- 采用**assign**语句描述



$$Q = \overline{\overline{S_D} \cdot \overline{Q}}$$

$$\overline{Q} = \overline{\overline{R_D} \cdot Q}$$

由与非门构成的基本RS锁存器

```
module RS_FF(Q, QN, SDN, RDN);  
    input  SDN, RDN;  
    output Q, QN;  
    assign Q = !(SDN && QN);  
    assign QN = !(RDN && Q);  
endmodule
```

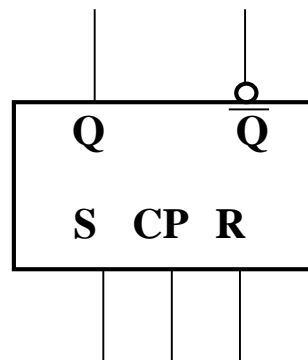
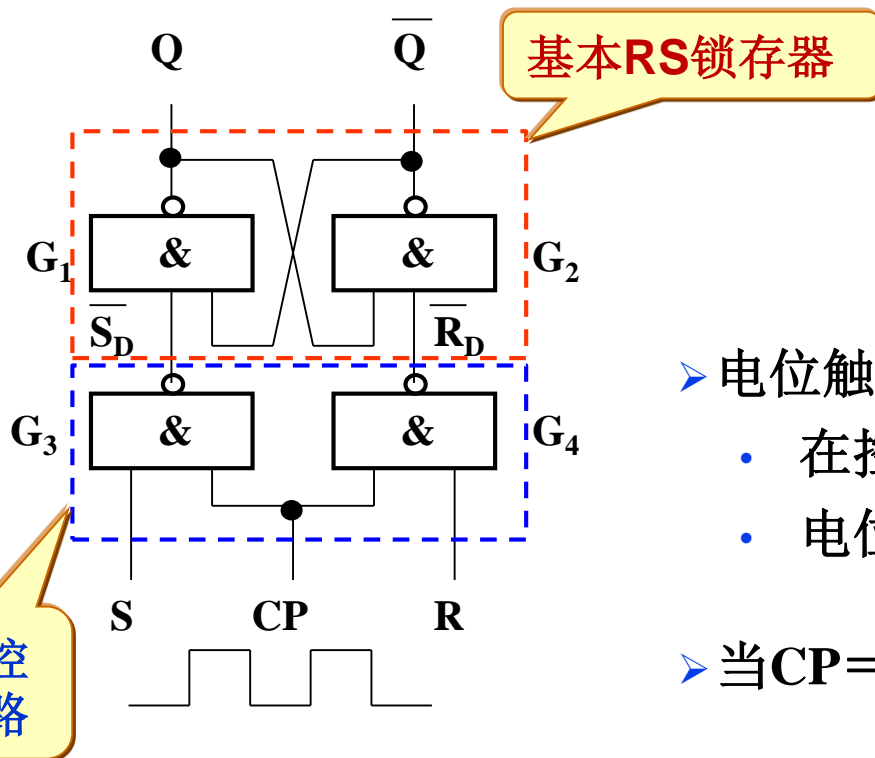


## 钟控RS锁存器

- ❖ 基本RS 锁存器中，是由输入信号直接控制锁存器的输出
- ❖ 在数字系统中，为了协调各部分电路的运行，常常要求某些锁存器在时钟信号的控制下同时动作，这就需要增加一个控制端（时钟），只有在控制端作用脉冲时，锁存器才能动作，这种有时钟控制端的锁存器叫做**钟控锁存器**
- ❖ 由于这里时钟信号为高电位（或低电位）时锁存器的状态随输入变化，因此，钟控锁存器是**电位触发方式**的锁存器
- ❖ 钟控锁存器在时钟控制下**同步**工作，因此也称**同步锁存器**

# 钟控RS锁存器的工作原理

## ❖ 电路结构和逻辑符号



### ➤ 电位触发:

- 在控制电位CP的控制下接收数据。
- 电位触发有正电位触发和负电位触发。

➤ 当 $CP=0$ 时,  $\overline{S_D}=\overline{R_D}=1$ , 保持原有状态。

➤ 当 $CP=1$ 时, 其输出状态由R、S端的输入信号决定——具有基本RS锁存器的功能。

# 钟控RS锁存器的逻辑功能表示

- **CP=0**时，锁存器处于保持状态；
- **CP=1** 时，具有基本**RS**锁存器的功能——称为钟控**RS**锁存器

特性表(**CP=1**)

S	R	$Q^n$	$Q^{n+1}$	功能
0	0	0	0	保持
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	置1
1	0	1	1	
1	1	0	X	不确定
1	1	1	X	

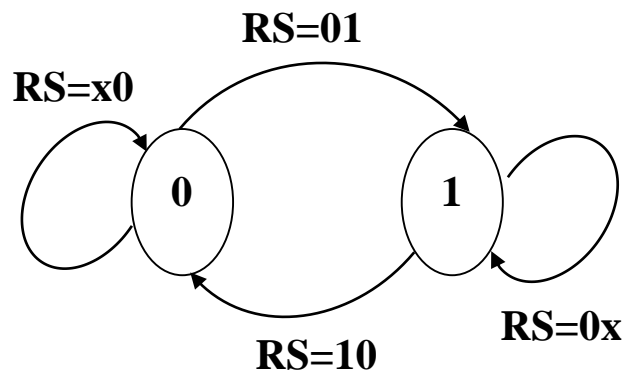
**R、S不能同时为1**

## 特性方程

$$Q^{n+1} = S + \bar{R} \cdot Q^n$$

$$S \cdot R = 0 \text{ (约束条件)}$$

$$Q^{n+1} = \bar{S}\bar{R}Q^n + (S\bar{R}\bar{Q}^n + S\bar{R}Q^n) = \bar{S}\bar{R}Q^n + S\bar{R}$$



状态转换图

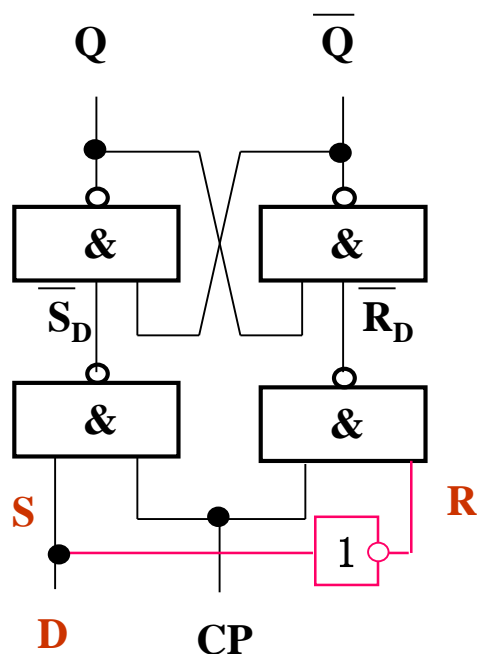
## 钟控D锁存器

❖ 问题 —— 如何消除钟控RS锁存器的不确定状态？

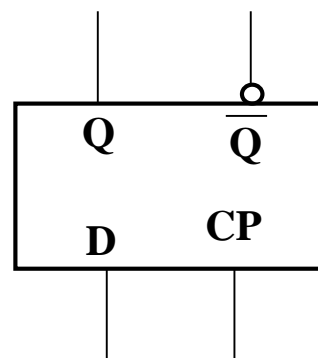
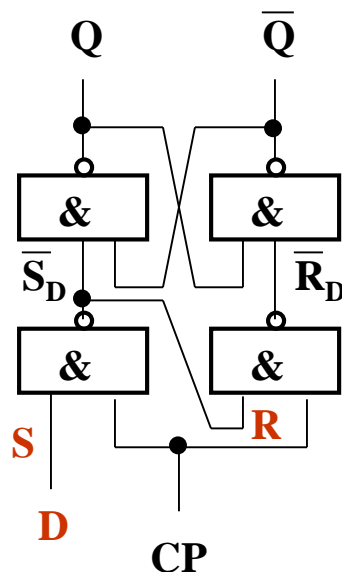
➤ 将钟控RS锁存器的输入由R、S双端输入，改为单端输入（D），即：将其S输入端改为D输入端，然后经过非门接R端。

➤ 这样，S、R总是互非，钟控D锁存器不会出现不定状态！

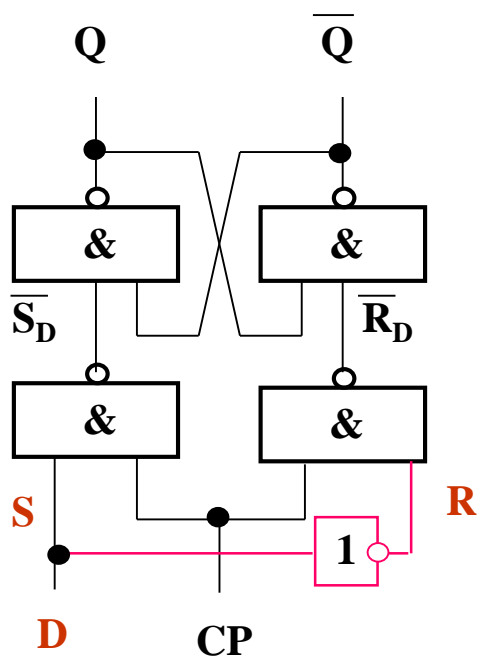
❖ 钟控D锁存器电路结构和逻辑符号



或者



# 钟控D锁存器工作原理

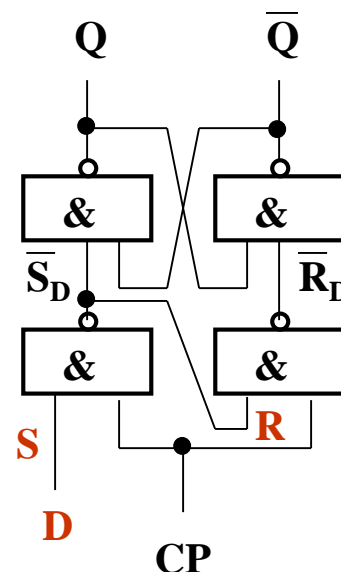


➤ 当 $CP=0$ 时， $\bar{S}_D=\bar{R}_D=1$ ，保持原态。

➤ 当 $CP=1$ 时：

- 若 $D=0$ ，相当于 $S=0$ ， $R=1$ ，置“0”
- 若 $D=1$ ，相当于 $S=1$ ， $R=0$ ，置“1”

➤ 借用输入控制电路的一个与非门，对D反相后，反馈到另一个与非门的输入端，作为R信号



# 钟控D锁存器的逻辑功能表示

## ❖ 电路功能

➤ CP=0时保持

CP=1时的特性方程

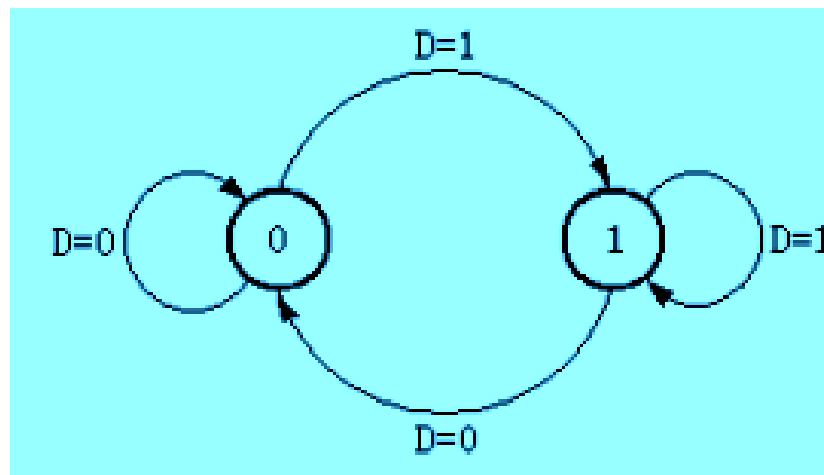
$$Q^{n+1} = D$$

特性表(CP=1)

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

$$Q^{n+1} = D\overline{Q}^n + DQ^n = D$$

状态图



# D锁存器的HDL设计

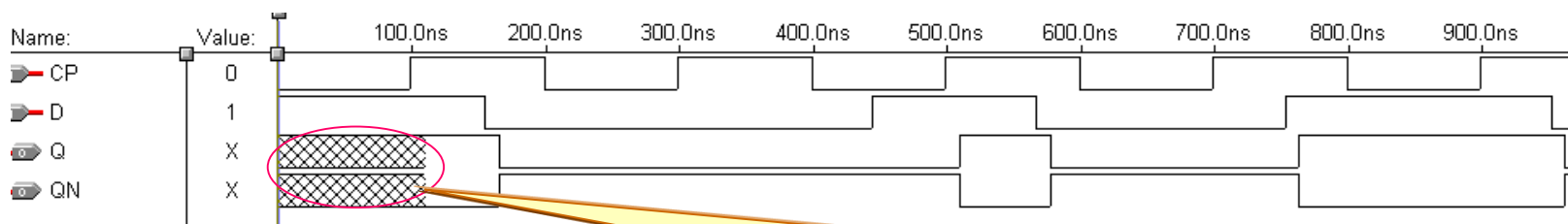
❖ 根据特性表，采用行为描述方式

➤ D锁存器为电位触发器，假定为高电平触发

$$Q^{n+1} = D \quad (CP = 1)$$

$$Q^{n+1} = Q^n \quad (CP = 0)$$

```
module D_FF_1(CP,D,Q,QN);  
    input  CP,D;  
    output Q,QN;  
    reg    Q,QN;  
  
    always  
    begin  
        if (CP == 1)begin Q = D;QN = ~Q; end  
        else          begin Q = Q;QN = QN; end  
    end  
endmodule
```



加电之初，锁存器状态随机。CP=0时保持原态

## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. RS锁存器/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. Moore型有限状态机
2. Mealy型有限状态机

### 三. 时序逻辑电路设计分析

1. 数据寄存器
2. 移位寄存器
3. 计数器



# D触发器

❖ 一个D触发器可由两个反相的D锁存器构成 (图(a)) :

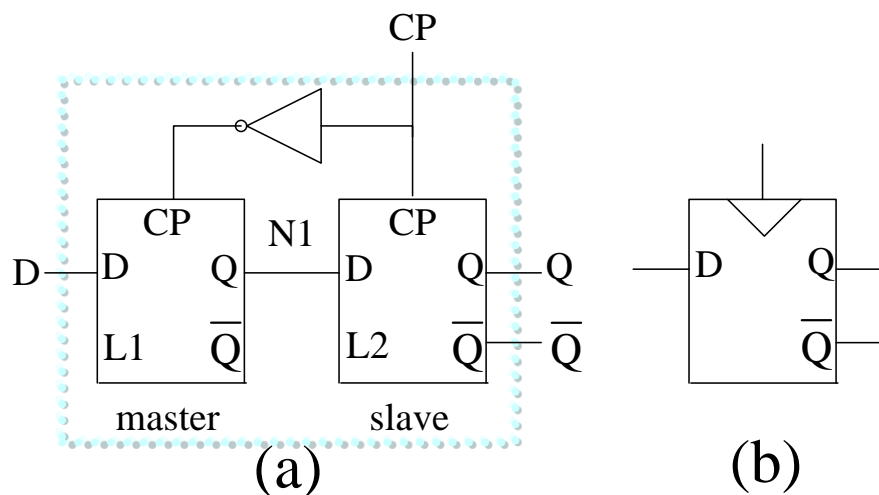
- 锁存器L1为主锁存器, L2为从锁存器
- L1和L2的时钟输入是反相的

❖ 工作原理

- $CP=0$ : L1是通路,  $N1 \leftarrow D$ , 而L2是断路, Q值不变
- $CP$ 从0上升到1:  $Q \leftarrow N1$ , 触发时刻
- $CP=1$ : L1是断路, N1值不变, 而L2是通路,  $Q \leftarrow N1$

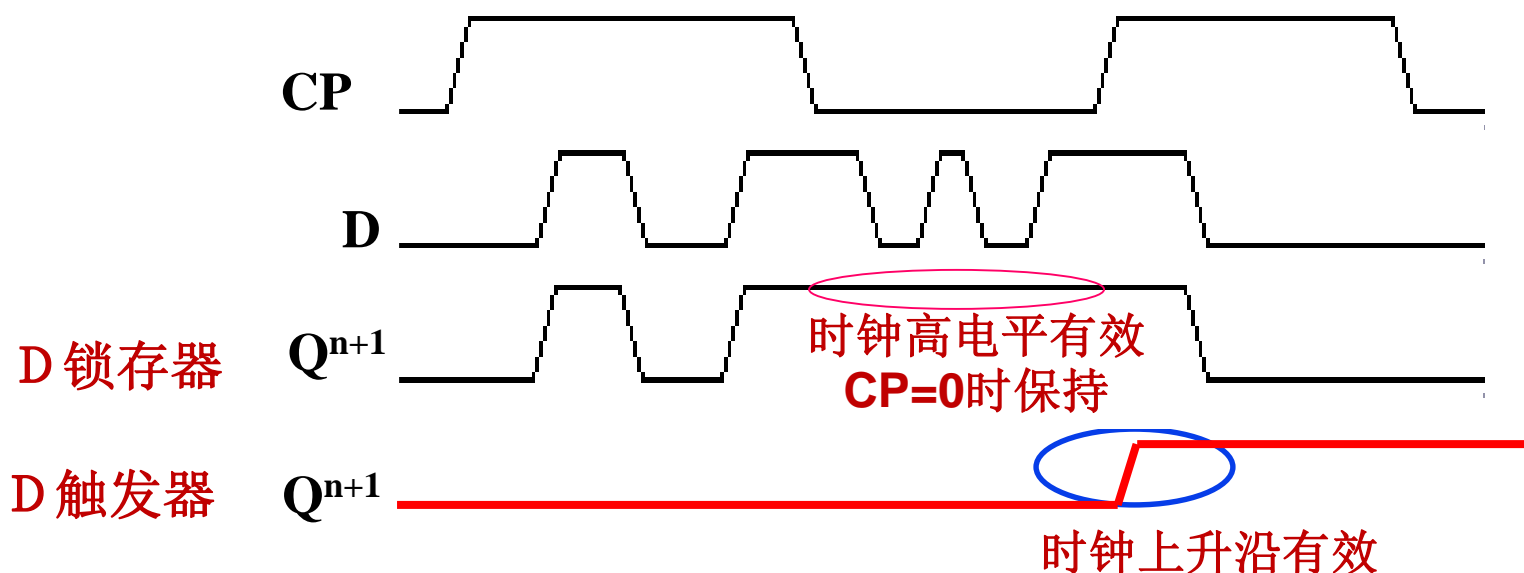
❖ D触发器分类

- 主从触发器
- 边沿触发器
- 维持一阻塞触发器



## 锁存器与触发器的区别

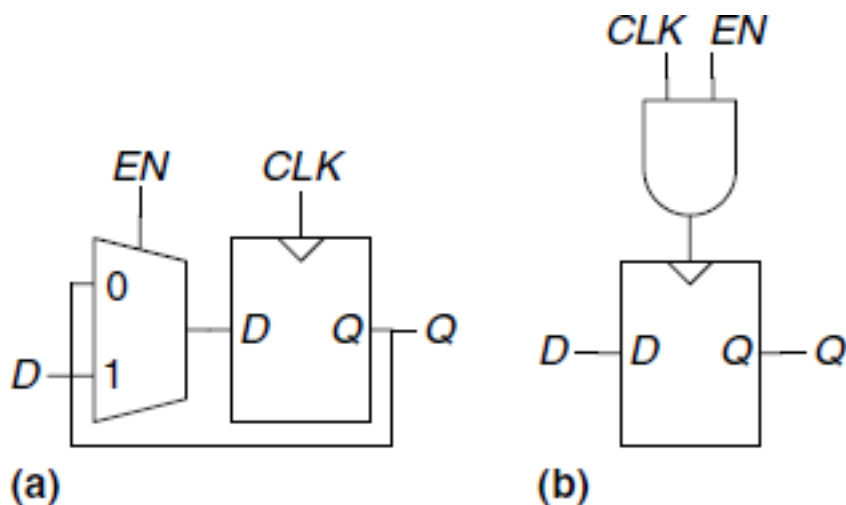
- ❖ 锁存器是电位（电平）触发的，只有在时钟CP有效电平（高电平CP=1或低电平CP=0）期间，锁存器的状态才有可能发生变化。
- ❖ 触发器的状态变化只发生在时钟CP的有效沿（上升沿或者下降沿）期间，CP=1、CP=0时触发器的状态不会发生变化。



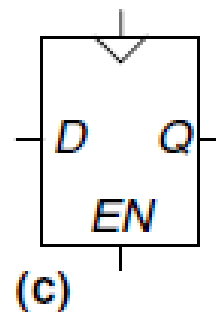
## 带使能端的D触发器

❖ 增加输入使能信号**EN** (**ENable**)，用于确定在有效时钟沿到来时，触发器是否能够载入数据，如图(a)和(b)所示

- **EN=1** 时， **D**触发器正常工作
- **EN=0** 时， **D**触发器状态不变
- 在时钟信号上一般不要设置逻辑，否则可能因延迟导致时序错误



(a,b) EN使能信号原理图



(c) D触发器电路符号

## 带复位功能的D触发器

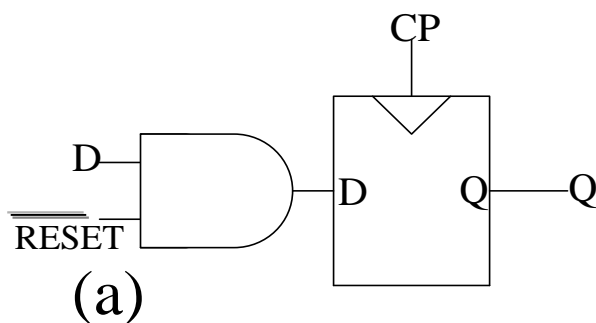
### ❖ 增加输入复位信号/**RESET**，如下图所示

- 当系统加电时，将触发器设置为已知状态 ( $Q=0$ )
- **/RESET** 有效时 ( $=0$ )，**D**触发器复位 ( $Q=0$ )
- **/RESET** 无效时 ( $=1$ )，**D**触发器正常工作

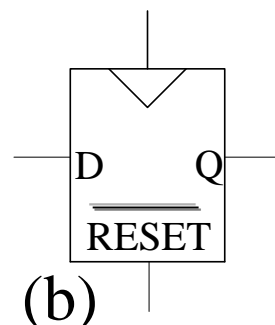
### ❖ 复位方式

- 同步复位：复位信号有效和时钟有效沿同时有效才能复位（置0）
- 异步复位：只要复位信号有效就能复位

### ❖ 有的触发器还带有置位 (**SET**) 功能 ( $Q=1$ )



(a)原理图



(b)电路符号

## 由D触发器构成寄存器

### ❖ 由同一时钟控制的 $N$ 个D触发器可以构成 $N$ 位寄存器

➤ 图(a)为4位寄存器，图(b)为电路符号

D<sub>3</sub>D<sub>2</sub>D<sub>1</sub>D<sub>0</sub>: 并行数据输入

Q<sub>3</sub>Q<sub>2</sub>Q<sub>1</sub>Q<sub>0</sub>: 并行数据输出

### ❖ 工作原理

(1) 清除（复位）

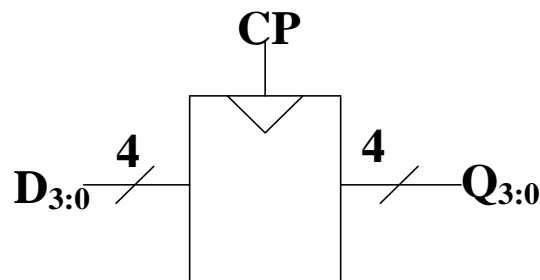
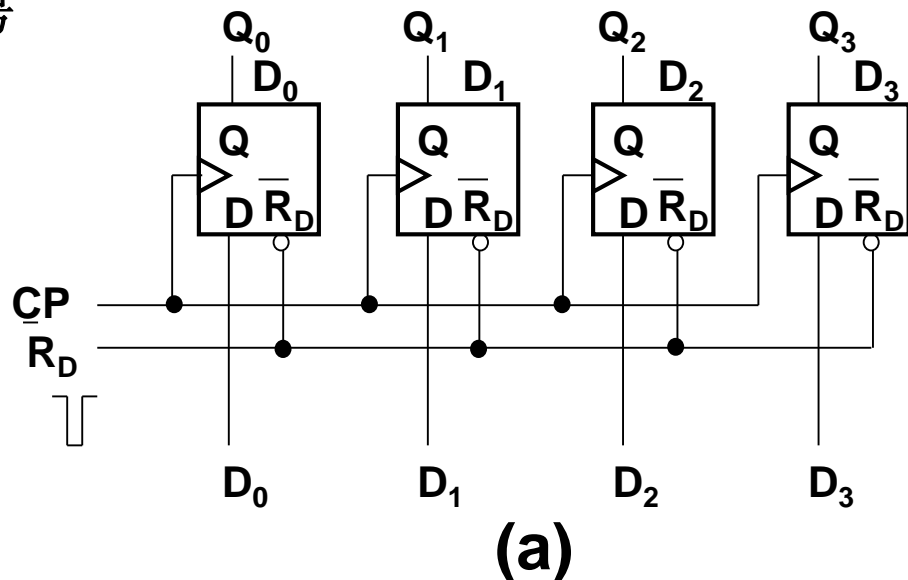
当  $\overline{R_D} = 0$ ,  $Q_0Q_1Q_2Q_3 = 0000$

(2) 置数（复位端无效时）

当CP上升沿到来是,  $Q_0Q_1Q_2Q_3 = D_0D_1D_2D_3$

### ❖ 工作方式（数据输入输出方式）

➤ 并入并出



(b) 寄存器电路符号

## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

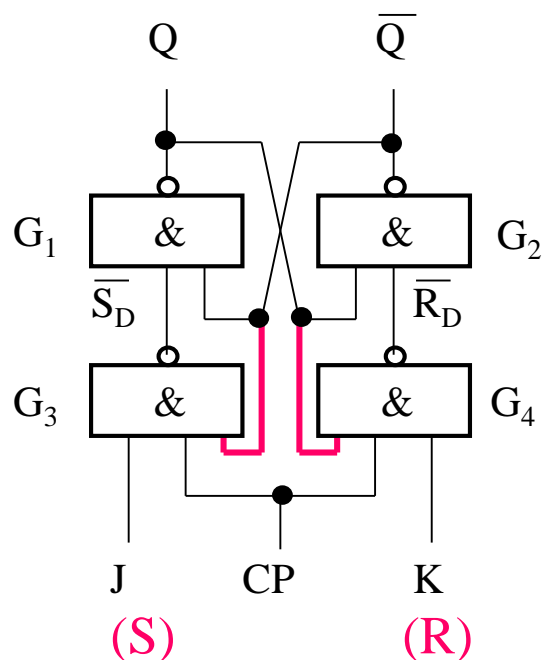
1. Moore型有限状态机
2. Mealy型有限状态机

### 三. 时序逻辑电路设计分析

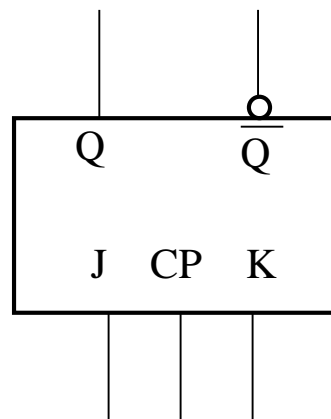
1. 数据寄存器
2. 移位寄存器
3. 计数器

## 钟控JK触发器

- ❖ **D**锁存器虽然没有约束条件，但功能较少（只有置**0**、置**1**功能）。
- ❖ **JK**触发器是一种功能最全面，而且没有约束条件的**FF**。
- ❖ **JK**触发器是在钟控**RS FF**的基础上，增加两条反馈线：
  - **Q** 反馈到 **R** 钟控门的输入端，并把 **R** 改名为 **K**
  - **/Q** 反馈到 **S** 钟控门的输入端，并把 **S** 改名为 **J**



钟控**JK**触发器的电路结构



- ❖ 把**RS=11**的无效状态变为**JK**触发器的**翻转（计数）**功能

# 钟控JK触发器的逻辑功能表示

## ❖ 电路功能

**CP=0**时为保持功能  
特性表(**CP=1**)

J	K	$Q^n$	$Q^{n+1}$	功能
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	1	翻转
1	1	1	0	(计数)

$$\begin{aligned}
 Q^{n+1} &= \bar{J}\bar{K}Q^n + J\bar{K}\bar{Q}^n + J\bar{K}Q^n + JK\bar{Q}^n \\
 &= (J\bar{K}\bar{Q}^n + JK\bar{Q}^n) + (\bar{J}\bar{K}Q^n + J\bar{K}Q^n) \\
 &= J\bar{Q}^n + \bar{K}Q^n
 \end{aligned}$$

简化特性表(**CP=1**)

J K	$Q^{n+1}$	功能
0 0	$Q^n$	保持
0 1	0	置0
1 0	1	置1
1 1	$\bar{Q}^n$	翻转

特性方程

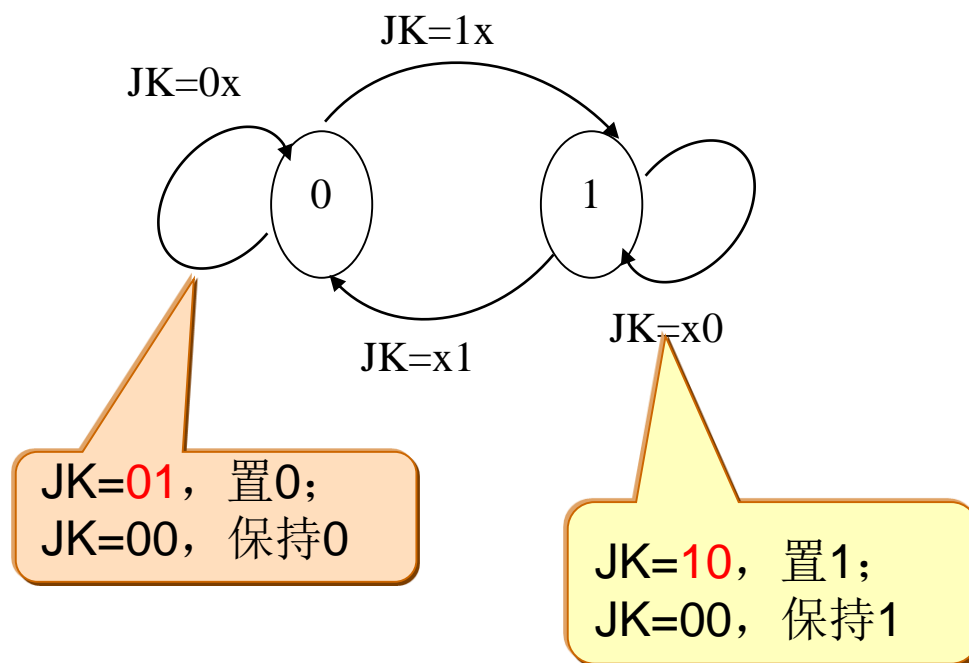
$$Q^{n+1} = J\bar{Q}^n + \bar{K} \cdot Q^n$$

**J0K0**, 输出不变; **J0K1**, 输出为0;  
**J1K0**, 输出为1; **J1K1**, 分频计数



# 钟控JK触发器的状态图

状态转换图



# 钟控JK触发器的HDL设计

## ❖ 设计分析

- ◆ 根据**CP=0**和**1**，分**2**种情况，适合用**if-else**语句来描述
  - **CP=0**时保持
  - **CP=1**时完成**JK FF**的功能。根据简化特性表有**4**种功能，适于用**case**语句来描述

简化特性表(**CP=1**)

J K	$Q^{n+1}$	功能
0 0	$Q^n$	保持
0 1	0	置0
1 0	1	置1
<b>1 1</b>	$\overline{Q^n}$	<b>翻转</b>

## 钟控JK触发器的源程序

```
module JK_FF(CP,J, K,Q,QN);
```

```
input    CP, J, K;
```

```
output   Q,QN;
```

```
reg      Q,QN;
```

```
always @(CP or J or K)
```

```
begin
```

```
    if (CP==0)
```

//保持

```
        begin Q = Q; QN = QN; end
```

```
    else if (CP==1)
```

```
        case ({J,K})
```

```
            2'b00: begin Q = Q;    QN = QN; end //保持
```

```
            2'b01: begin Q = 1'b0; QN = 1'b1; end //置0
```

```
            2'b10: begin Q = 1'b1; QN = 1'b0; end //置1
```

```
            2'b11: begin Q = !Q;   QN = !QN; end //翻转
```

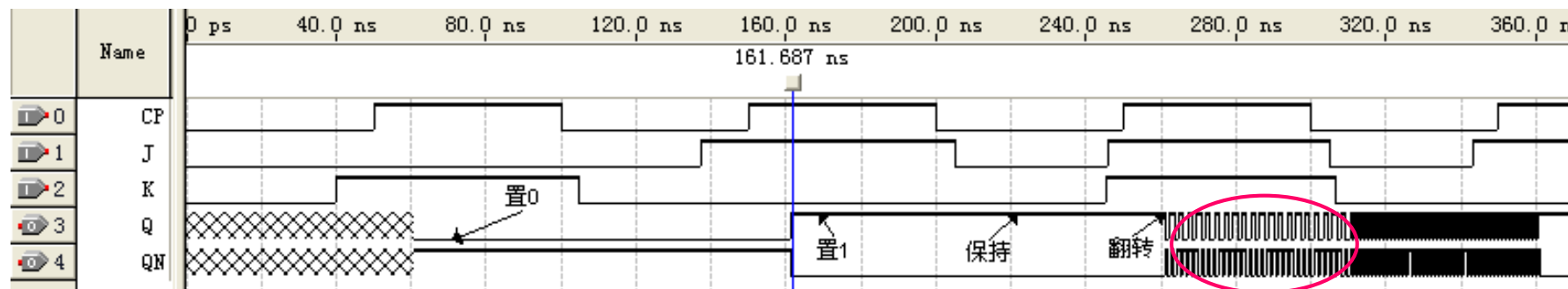
```
        endcase
```

```
    end
```

```
endmodule
```

采用行为描述方式

# 钟控JK触发器的仿真波形



**J=1、K=1时  
发生空翻**

❖ 钟控JK触发器比D锁存器新增的功能:

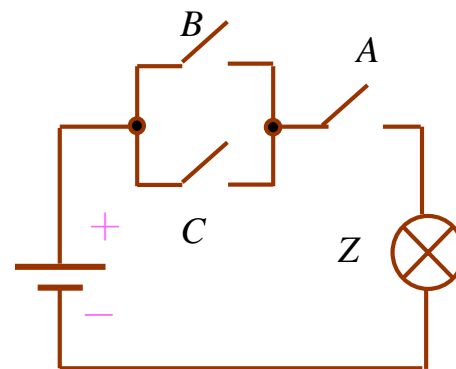
当JK=11时, 输出的波形翻转 (可用于计数);  
当JK=00时, 触发器保持原来的状态。

## 触发器应用举例

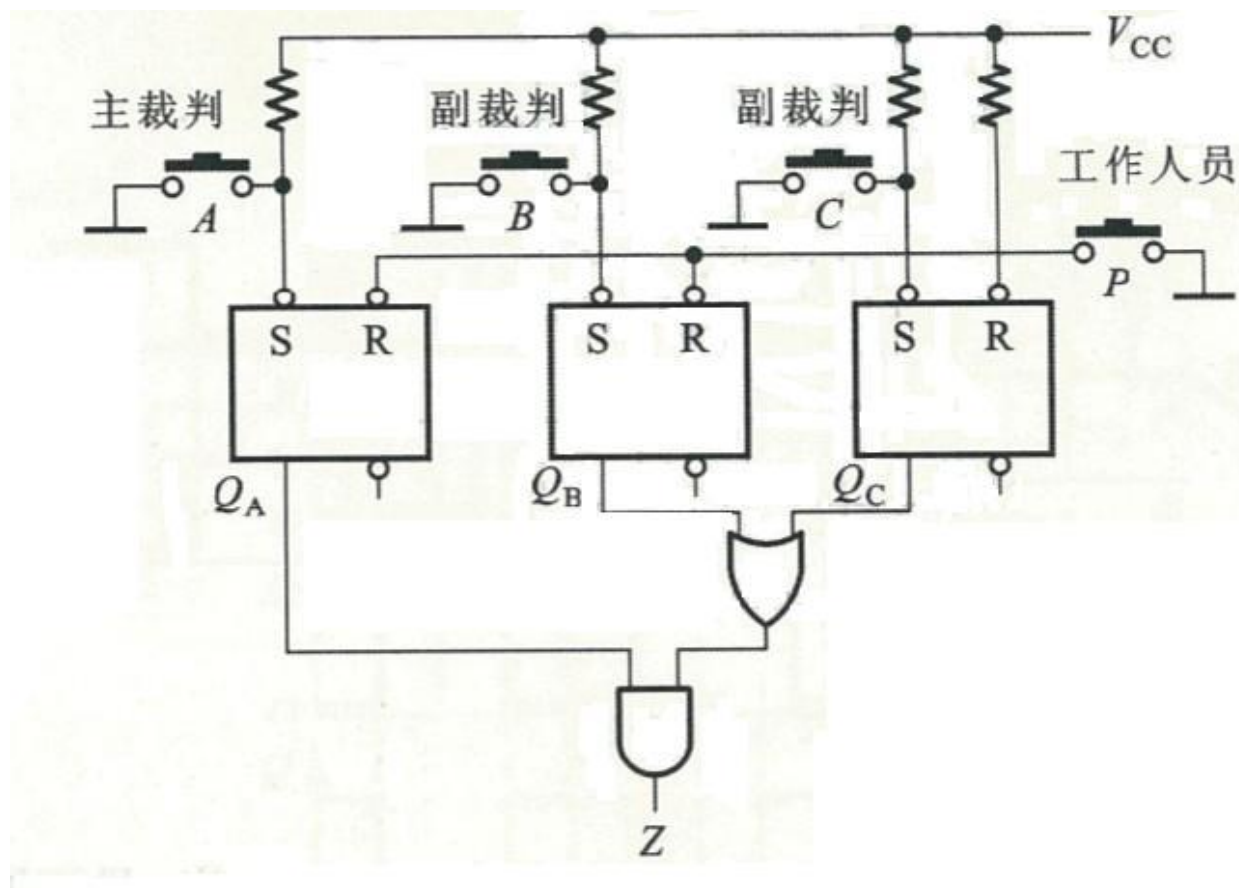
【例1】设计一个举重裁判逻辑电路。在一个主裁判员和两个副裁判员当中，只有包含主裁判员在内的两人以上认定试举动作合格，并按下自己的按钮时，则表示试举成功的输出信号 $Z=1$ 。而且，要求这个 $Z=1$ 的信号一直保持下去，直到工作人员按下清除按钮为止。

解：

- (1) 由于三个裁判员按下按钮发出的信号不能自行保持，而且按动的动作可能有先后、长短之别，因此需要用三个触发器分别保存三个按钮给出的信号。
- (2) 选用基本RS锁存器，则可用裁判员按下按钮A、B、C输出的低电平信号接触发器的S输入端，作为置1信号；用工作人员按动按钮P给出的低电平信号接R输入端，作为置0信号。
- (3) 试举成功的信号Z由三个触发器的输出状态判别——或与逻辑。



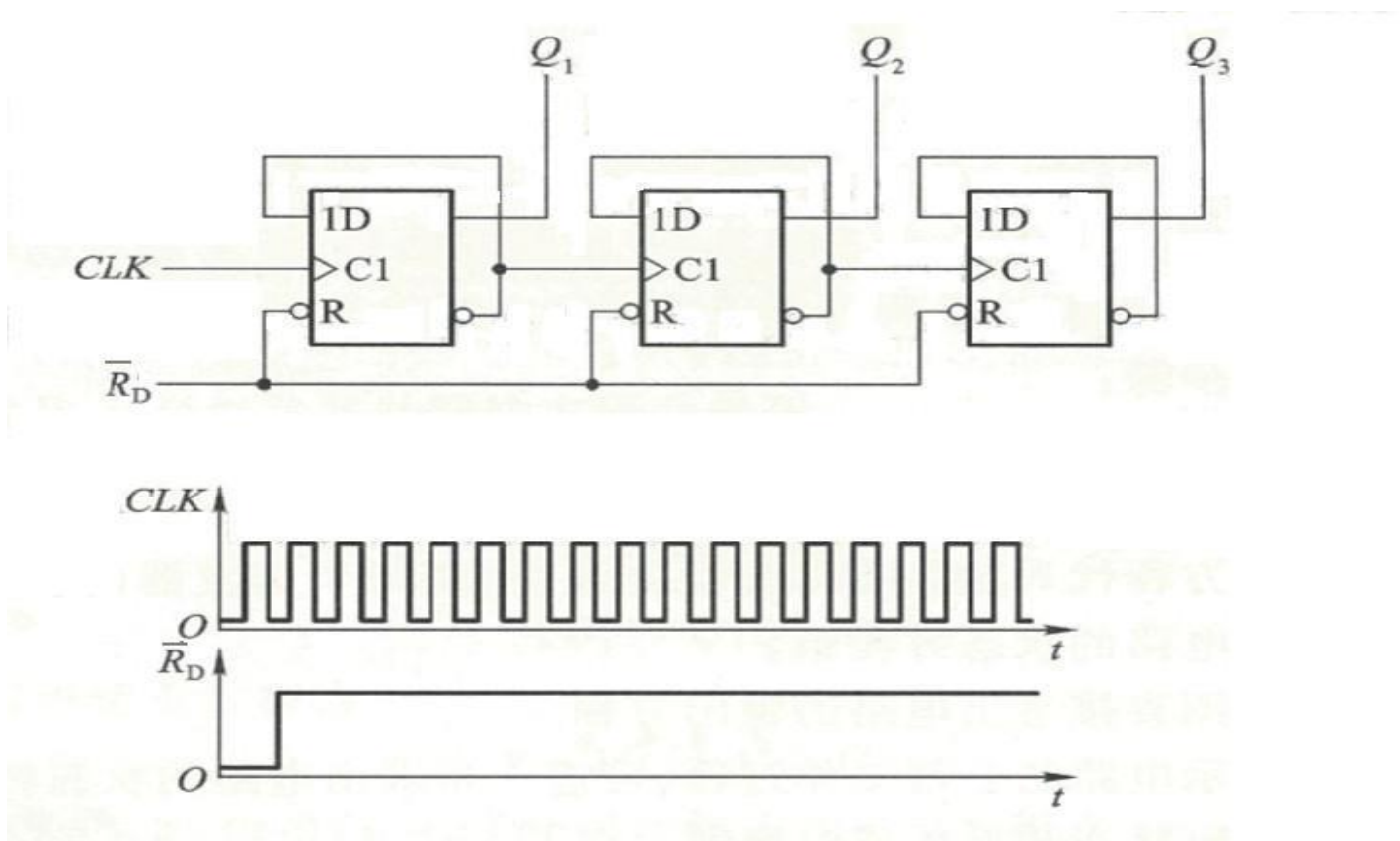
## 触发器应用举例



如果改用D触发器，如何设计？

## 触发器应用举例

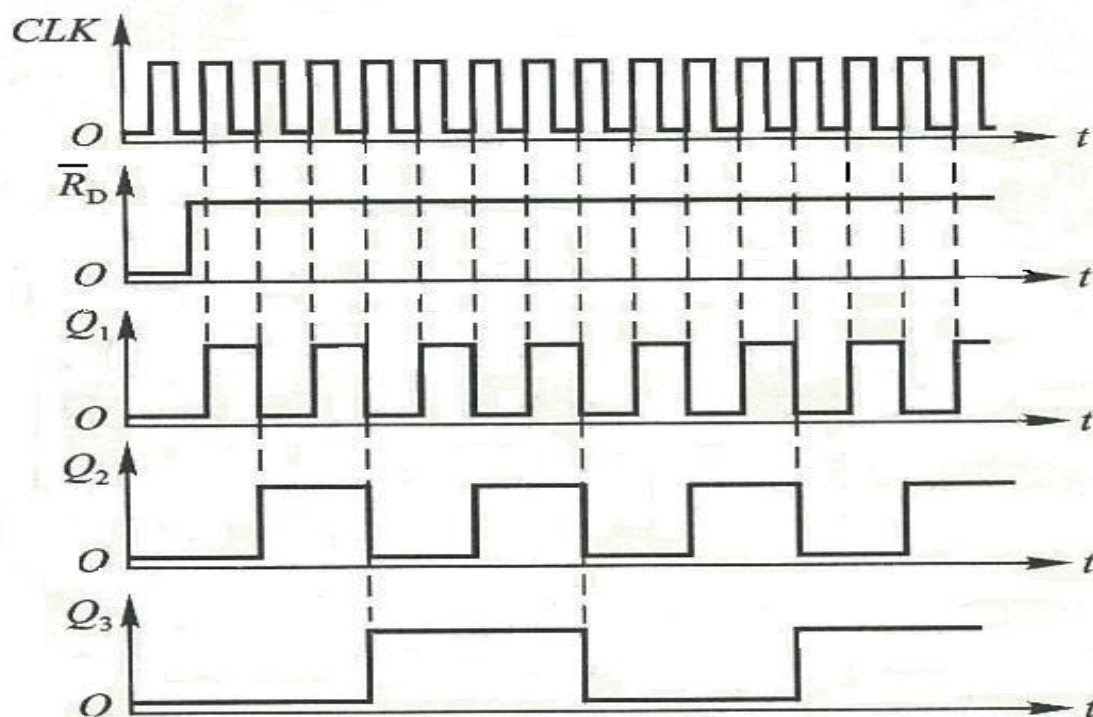
**【例2】** 分析下图中CLK、 $\overline{R_D}$  信号作用下输出信号Q1、Q2、Q3的频率与CLK信号频率之间的关系。



## 触发器应用举例

**解：**（1）由于每个D触发器都接成了  $D=\neg Q$ ，所以  $Q^{n+1}=\neg Q^n$ ，即每次时钟上升沿到来后D触发器都要翻转。

（2）C1的触发时钟是CLK，Q2的触发时钟是C1的输出 $\neg Q_1$ ，Q3的触发时钟是C2的输出 $\neg Q_2$ 。若CLK的频率为 $f_0$ ，则 $Q_1$ 、 $Q_2$ 、 $Q_3$ 输出脉冲频率依次为 $f_0/2$ 、 $f_0/4$ 、 $f_0/8$ 。



$Q_1$ 、 $Q_2$ 、 $Q_3$ 输出波形



## 小结

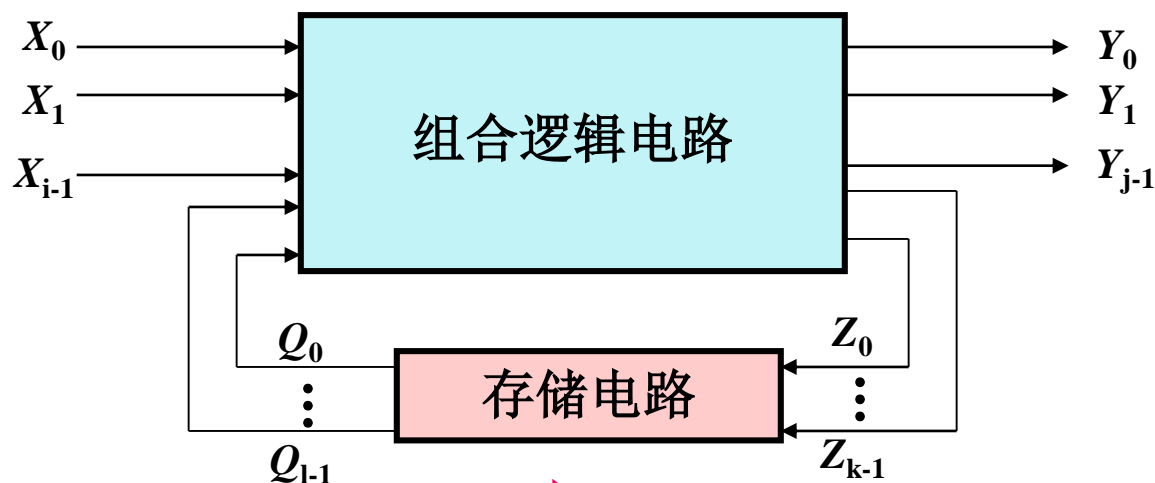
- ❖ **基本RS锁存器**：具有保持、置0、置1功能，其输入信号可以直接控制锁存器的输出（有约束条件）
- ❖ **钟控RS锁存器**：时钟信号CP有效时，锁存器的状态随输入变化（有约束条件：R、S不能同时为1）
- ❖ **钟控D锁存器**：为消除钟控RS锁存器的不定状态，将钟控RS锁存器双端输入改为单端输入（D），即D锁存器
- ❖ **D触发器**：两个反向钟控D锁存器构成D触发器。时钟信号CP的边沿（上升沿或下降沿）触发
- ❖ D锁存器是电平敏感的，D触发器是边沿触发的
- ❖ 寄存器由共享时钟信号CLK的多个D触发器构成
- ❖ JK触发器是一种功能全面，而且没有约束条件的FF

# 第七讲

---

## 上一讲简要回顾

- ❖ 组合逻辑电路：输出完全由输入决定，不受系统中统一时钟信号的控制；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
- ❖ 时序逻辑电路：当时的输出由当时的输入和电路的原来状态共同决定，并且具有“记忆”功能
- ❖ 时序逻辑电路结构特点：由组合逻辑电路和存储电路构成



触发器 (Flip-Flop, FF) 或寄存器

## 上一讲简要回顾

- ❖ **基本RS锁存器**：具有保持、置0、置1功能，其输入信号可以直接控制锁存器的输出（有约束条件）
- ❖ **钟控RS锁存器**：**CP**为有效电平（高电平或低电平）时，锁存器的状态随输入变化（有约束条件）
- ❖ **钟控D锁存器**：为消除钟控**RS**锁存器的不定状态，将钟控**RS**锁存器双端输入改为单端输入（**D**），即**D**锁存器
- ❖ **D触发器**：两个反相钟控**D**锁存器构成**D**触发器。时钟信号**CP**的边沿（上升沿或下降沿）触发
- ❖ **寄存器**：由共享时钟信号**CLK**的多个**D**触发器构成
- ❖ **JK触发器**：是一种功能全面，而且没有约束条件的**FF**

## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. **Moore**型有限状态机
2. **Mealy**型有限状态机

### 三. 时序逻辑电路设计分析

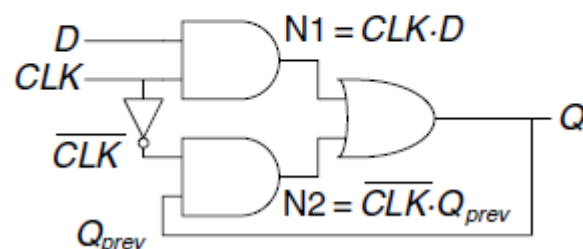
1. 数据寄存器
2. 移位寄存器
3. 计数器

# 时序电路特点

## 【例】电路实例分析

右图是一个新型D锁存器及其期望的真值表：  
即CLK有效时Q=D，CLK无效时Q状态保持，  
该D锁存器能正确工作吗？

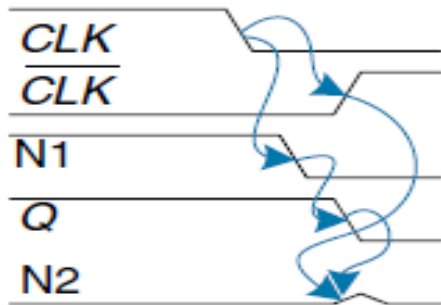
$$Q = CLK \cdot D + \overline{CLK} \cdot Q_{prev}$$



解：

- (1) 假设CLK=D=1时，N1=1，Q=1，Q<sub>prev</sub>=1。
- (2) 如不考虑门电路延迟，CLK由1变为0时，N1变0，N2变1，Q状态保持1，能实现D锁存器的功能。
- (3) 但如假设反相器的延迟比“与”和“或”门的长，现在CLK变为0，在/CLK变为1之前，N1和Q先后变为0，N2会因此不能变为1。Q值被钳为0。不能实现D锁存器的功能。

CLK	D	Q <sub>prev</sub>	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

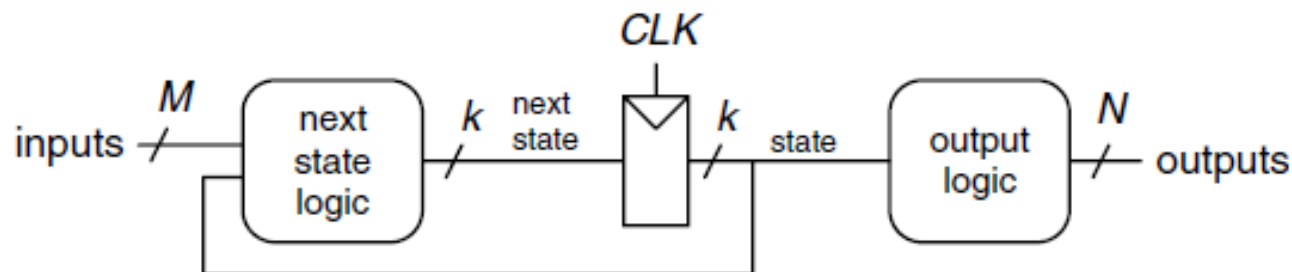


如时序电路存在直通环路，输出直接反馈到输入。  
这类时序电路存在不良竞争，门电路延时、温度、电压等都可能影响功能，且难发现。

# 同步时序电路

## ❖ 改进措施 —— 同步时序电路

- 在环路中插入寄存器以断开环路，电路变成组合逻辑电路和寄存器的组合
- 加入时钟，寄存器只在时钟有效沿到达时发生改变，时钟周期保证在下一个时钟沿达到之前，输入到寄存器的信号维持稳定



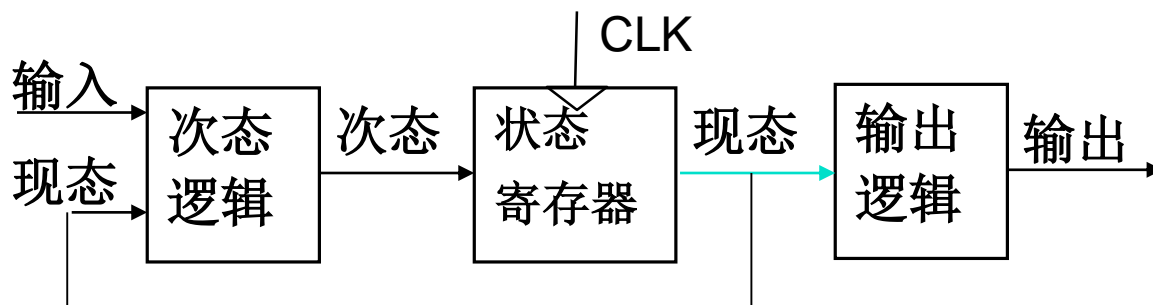
## ❖ 同步时序电路特点

- 每个电路元件都是组合逻辑或寄存器，且至少有一个寄存器；
- 每一个环路至少有一个寄存器；
- 所有寄存器接收同一个时钟信号。
- 同步时序电路可以描绘成有限状态机。

## 有限状态机概述

❖ **有限状态机**（Finite State Machine, FSM）：可描述有限个**状态**以及这些状态之间的**转移**及引起转移的**动作**等的离散数学模型。

- **次态逻辑**：组合逻辑，根据当前状态和输入计算下一状态
- **状态寄存器**：在时钟沿到来之前，保持现态，并为输出逻辑和次态逻辑提供稳定输入；在时钟沿到来时，锁存次态逻辑输出的状态值
- **输出逻辑**：组合逻辑，根据现态形成输出信号



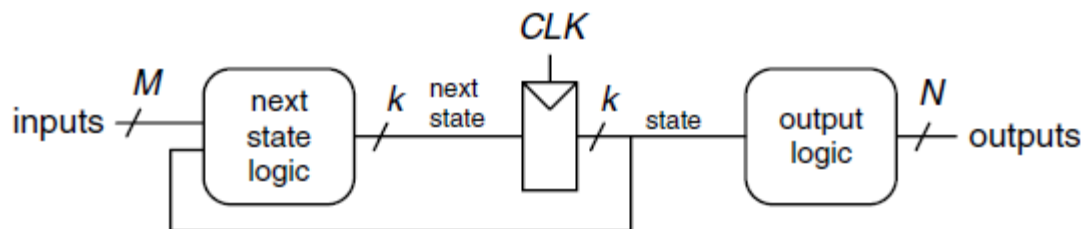
有限状态机典型结构



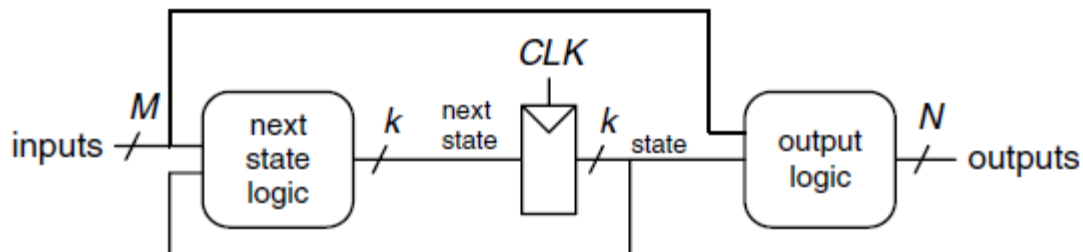
# 有限状态机概述

❖ 根据输出信号产生机理的不同，有限状态机可分成两类：

- 摩尔 (Moore) 型状态机： 输出信号仅与当前状态有关
- 米里 (Mealy) 型状态机： 输出信号与当前状态及输入信号有关



Moore型有限状态机



Mealy型有限状态机

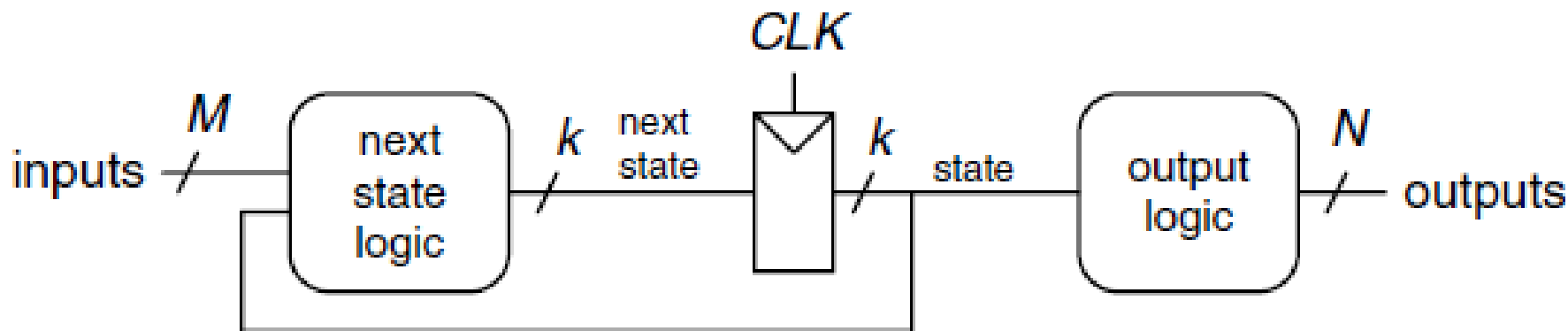
## 2.1 Moore型FSM设计

### ❖ Moore型有限状态机

- 输出信号仅与当前状态相关
- 输入 $M$ 位，输出 $N$ 位，状态 $K$ 位（可以表示 $2^k$ 个不同状态）
- **次态逻辑**：组合逻辑电路，输入 $M+K$ 位，输出 $K$ 位次态
- **状态寄存器**： $K$ 位寄存器
- **输出逻辑**：组合逻辑电路，输入 $K$ 位状态，输出 $N$ 位信号

### ❖ Moore型FSM的设计

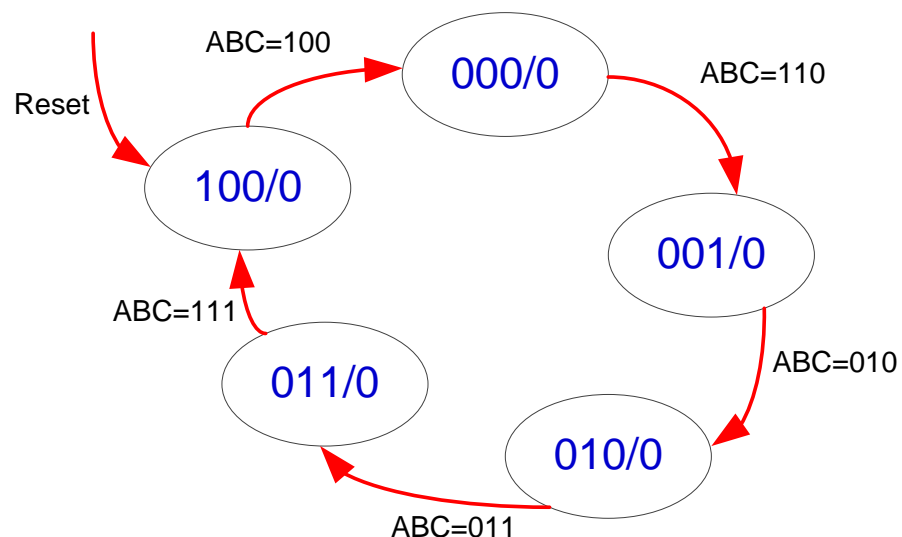
- 确定3个参数： $M$ ， $K$ ， $N$
- 设计2个逻辑：输出逻辑（真值表）、次态逻辑（状态转换图）



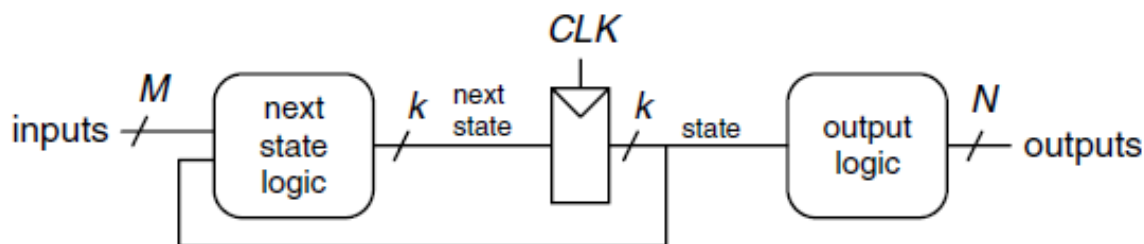
## 2.1 Moore型FSM设计

### ❖ Moore型FSM的表示方法

- **状态图 (State Diagram)**：圆圈表示状态，圈内“ $Q_2Q_1Q_0 / D$ ”分别表示状态组合 $Q_2Q_1Q_0$ （或状态编码）及输出信号 $D$ ；带箭头的线段表示状态转移，线段上的文字表示转移发生时的信号输入
- **状态表 (State Table)**：反映下一状态与当前状态和输入的关系



当前状态 ( $Q_2Q_1Q_0$ )	输入 (ABC)	下一状态 ( $Q_2'Q_1'Q_0'$ )
S4 (100)	100	S0 (000)
S0 (000)	110	S1 (001)
S1 (001)	010	S2 (010)
S2 (010)	011	S3 (011)
S3 (011)	111	S4 (100)



## 2.1 Moore型FSM设计

**【例】** 交通信号灯控制器：A大道，信号灯 $L_A$ ，连传感器 $T_A$ ；  
B大道，信号灯 $L_B$ ，连传感器 $T_B$ 。灯分红/绿/黄三色

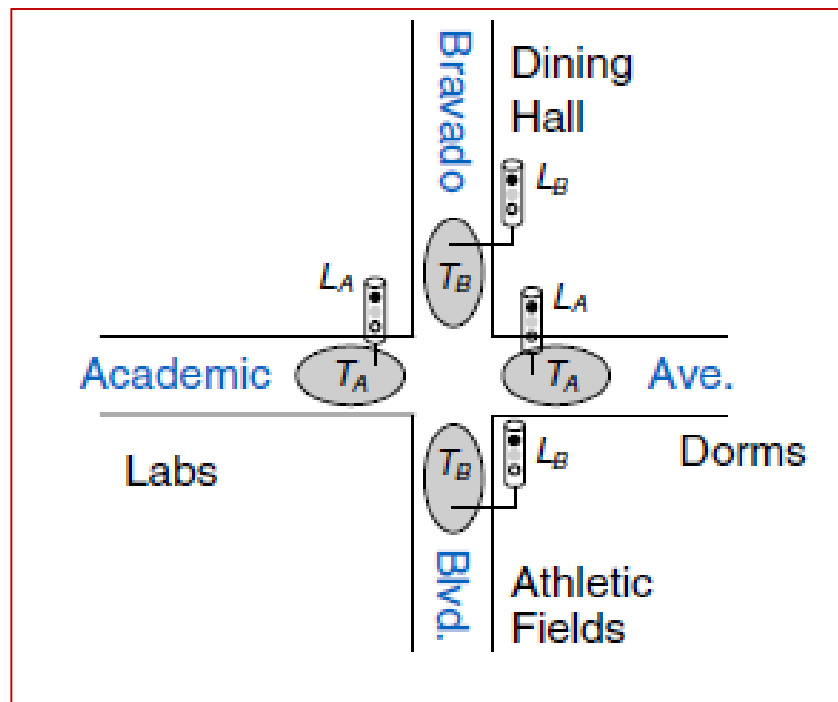
**解：** (1) 建立FSM模型

(1.1) 确定输入：东西A大道、南北B大道 分别安装 传感器  $T_A$  和  $T_B$ ：  
为1表示该路上有人等待； 为0表示无人等待

(1.2) 确定输出： 信号灯 $L_A$ 、 $L_B$ ，编码：**00表示绿，01表示黄，10表示红**

(1.3) 信号灯控制器FSM模型

- 输入：  $T_A$  和  $T_B$ ， 共 2 位
- 输出：  $L_A$  和  $L_B$ ， 共 4 位
- 状态： 共4个不同状态
  - S0:  $L_A$ 绿 且  $L_B$ 红
  - S1:  $L_A$ 黄 且  $L_B$ 红
  - S2:  $L_A$ 红 且  $L_B$ 绿
  - S3:  $L_A$ 红 且  $L_B$ 黄
- 状态寄存器： 2位



## 2.1 Moore型FSM设计

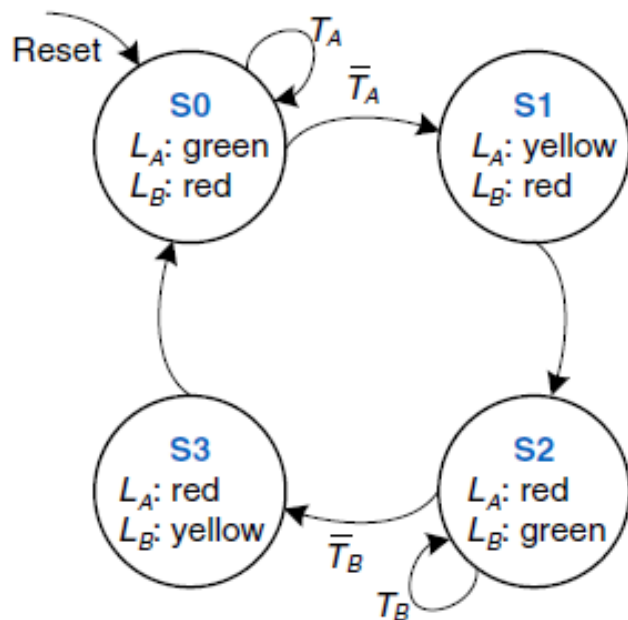
### (2) 画出状态转换图

控制器复位时，最初状态为： $L_A$ 绿、 $L_B$ 红。

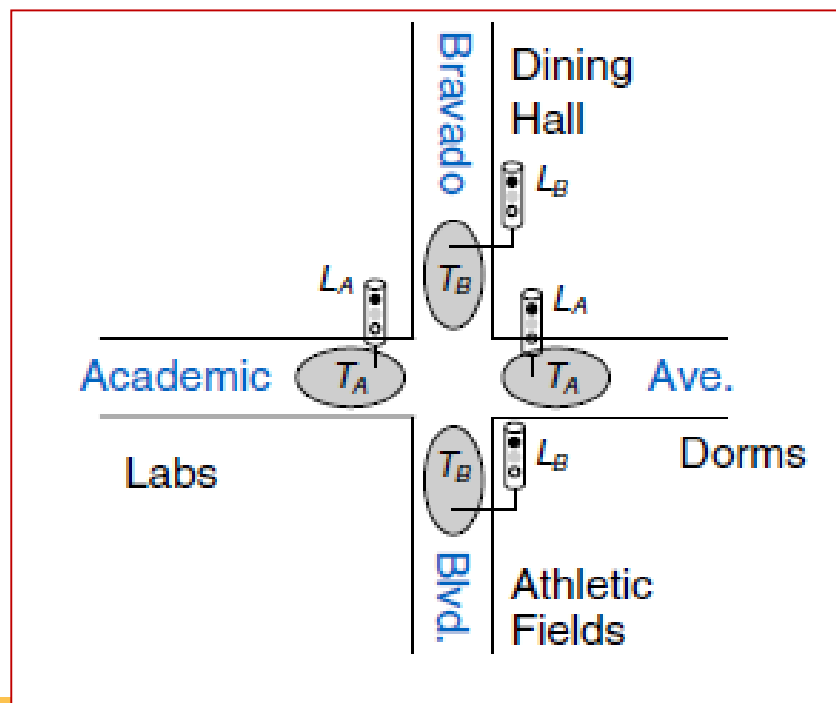
每隔5秒，检测A大道路口人员等待情况，并决定如何改变信号灯：

若 $T_A=1$ （表示A大道有人等待或到达），信号灯维持不变；

若 $T_A=0$ ，则 $L_A$ 先变黄并保持5秒，然后， $L_A$ 变红 且  $L_B$ 同时变绿  
B大道按同样方式处理。得到状态转换图如下。



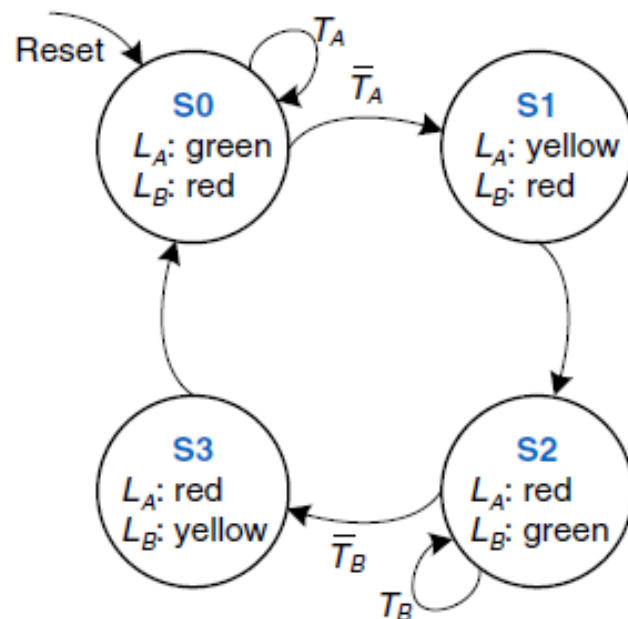
状态转换图



## 2.1 Moore型FSM设计

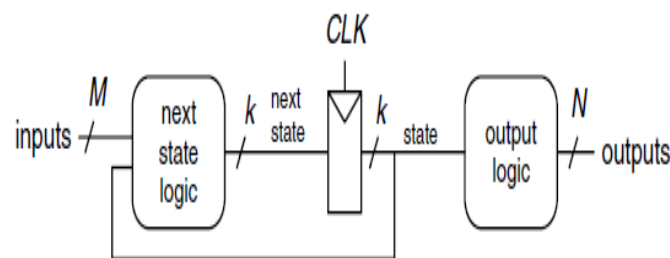
(3) 根据状态转换图，得到 状态表 和 输出逻辑表

当前状态 ( $S_1S_0$ )	输入	下一状态 ( $S'_1S'_0$ )
S0 (00)	$T_A=1, T_B=X$	S0 (00)
S0 (00)	$T_A=0, T_B=X$	S1 (01)
S1 (01)	$T_A=X, T_B=X$	S2 (10)
S2 (10)	$T_A=X, T_B=1$	S2 (10)
S2 (10)	$T_A=X, T_B=0$	S3 (11)
S3 (11)	$T_A=X, T_B=X$	S0 (00)



状态转换图

当前状态 ( $S_1S_0$ )	输出 $L_A$ ( $L_{A1}L_{A0}$ )	输出 $L_B$
S0 (00)	绿 (00)	红 (10)
S1 (01)	黄 (01)	红 (10)
S2 (10)	红 (10)	绿 (00)
S3 (11)	红 (10)	黄 (01)



## 2.1 Moore型FSM设计

(4) 根据状态表和输出逻辑表，写出 次态逻辑表达式 和 输出逻辑表达式

次态逻辑表达式：

$$\begin{aligned} S_1' &= \overline{S_1}S_0 + S_1\overline{S_0}T_B + S_1\overline{S_0}\overline{T_B} \\ &= \overline{S_1}S_0 + S_1\overline{S_0} \\ &= S_0 \oplus S_1 \end{aligned}$$

$$S_0' = \overline{S_1}\overline{S_0}T_A + S_1\overline{S_0}\overline{T_B}$$

输出逻辑表达式：

$$\begin{aligned} L_{A1} &= S_1 & L_{A0} &= \overline{S_1}S_0 \\ L_{B1} &= \overline{S_1} & L_{B0} &= S_1S_0 \end{aligned}$$

当前状态 ( <b>S<sub>1</sub></b> <b>S<sub>0</sub></b> )	输入	下一状态 ( <b>S'<sub>1</sub></b> <b>S'<sub>0</sub></b> )
S0 ( <b>00</b> )	T <sub>A</sub> =1, T <sub>B</sub> =X	S0 ( <b>00</b> )
S0 ( <b>00</b> )	T <sub>A</sub> =0, T <sub>B</sub> =X	S1 ( <b>01</b> )
S1 ( <b>01</b> )	T <sub>A</sub> =X, T <sub>B</sub> =X	S2 ( <b>10</b> )
S2 ( <b>10</b> )	T <sub>A</sub> =X, T <sub>B</sub> =1	S2 ( <b>10</b> )
S2 ( <b>10</b> )	T <sub>A</sub> =X, T <sub>B</sub> =0	S3 ( <b>11</b> )
S3 ( <b>11</b> )	T <sub>A</sub> =X, T <sub>B</sub> =X	S0 ( <b>00</b> )

当前状态 ( <b>S<sub>1</sub></b> <b>S<sub>0</sub></b> )	输出L <sub>A</sub> ( <b>L<sub>A1</sub></b> <b>L<sub>A0</sub></b> )	输出L <sub>B</sub>
S0 ( <b>00</b> )	绿 ( <b>00</b> )	红 ( <b>10</b> )
S1 ( <b>01</b> )	黄 ( <b>01</b> )	红 ( <b>10</b> )
S2 ( <b>10</b> )	红 ( <b>10</b> )	绿 ( <b>00</b> )
S3 ( <b>11</b> )	红 ( <b>10</b> )	黄 ( <b>01</b> )

## 2.1 Moore型FSM设计

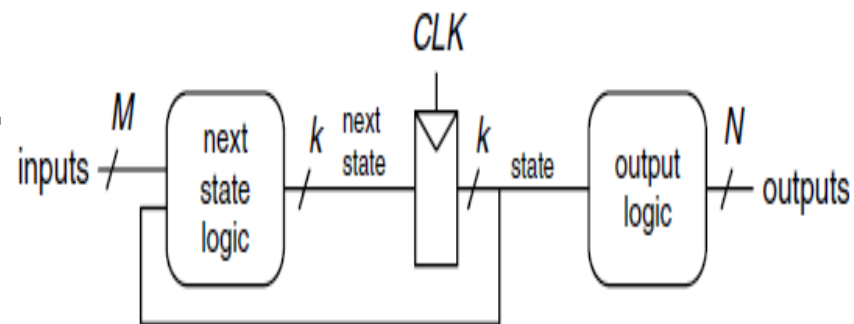
(5) 根据逻辑表达式画出逻辑电路图

$$S_1' = \overline{S_1}S_0 + S_1\overline{S_0}T_B + S_1\overline{S_0}\overline{T_B}$$

$$= \overline{S_1}S_0 + S_1\overline{S_0}$$

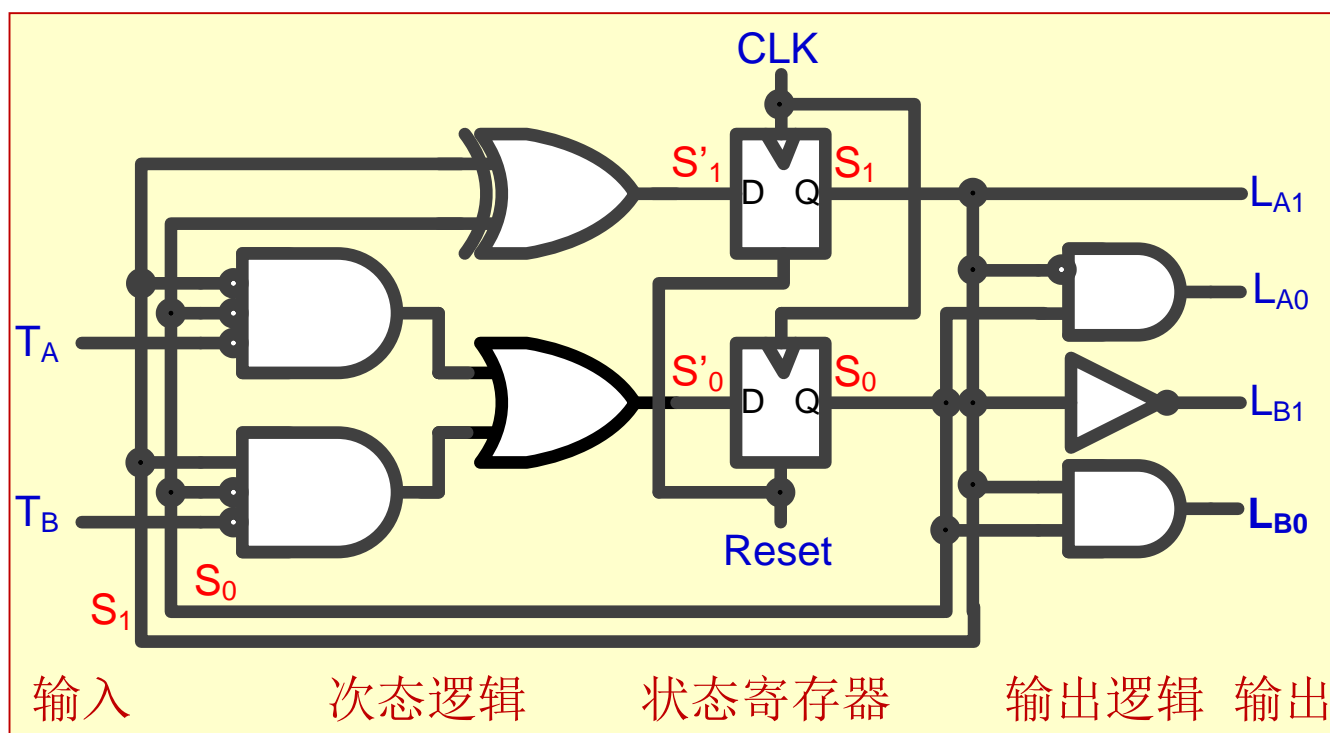
$$= S_0 \oplus S_1$$

$$S_0' = \overline{S_1}\overline{S_0}T_A + S_1\overline{S_0}\overline{T_B}$$



$$L_{A1} = S_1 \quad L_{A0} = \overline{S_1}S_0$$

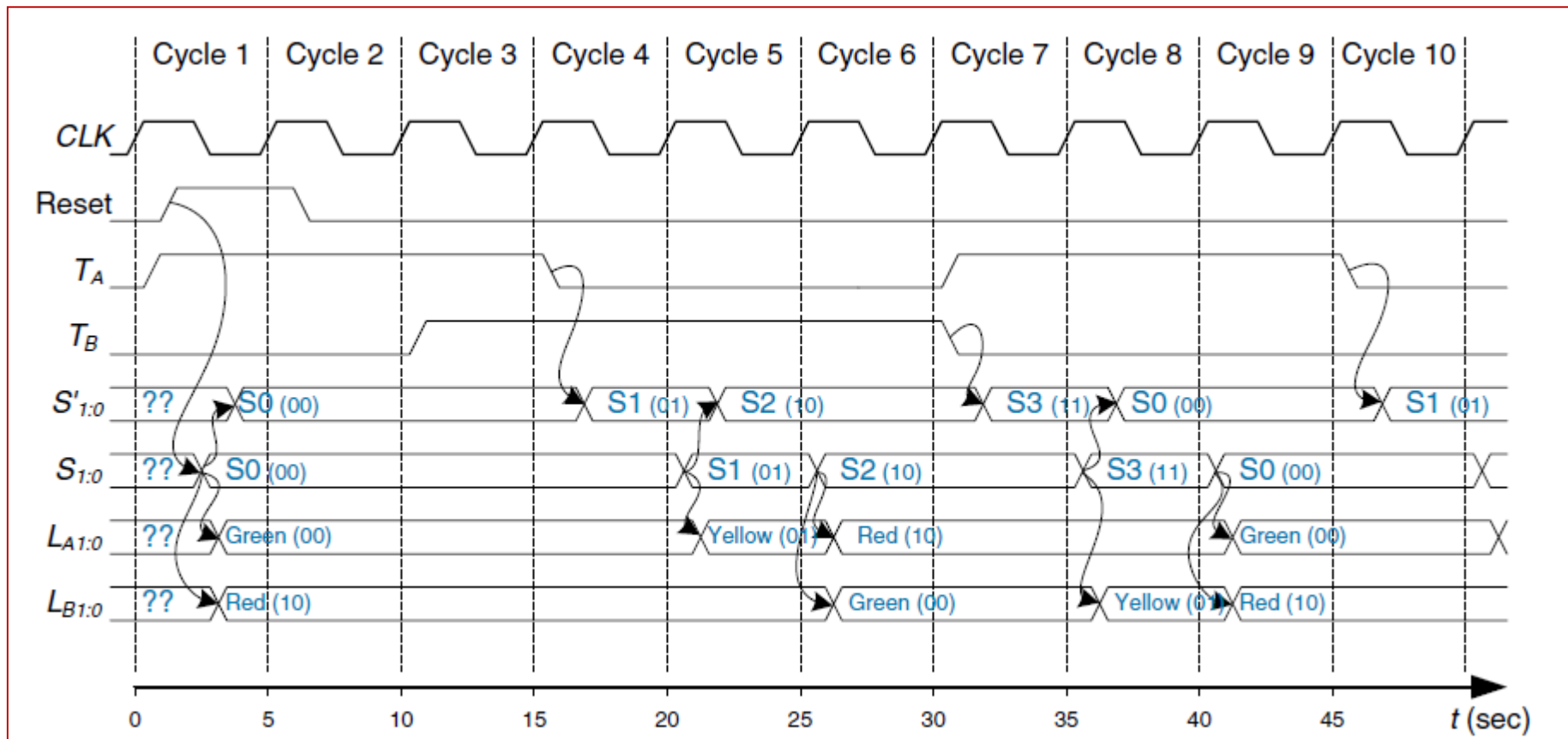
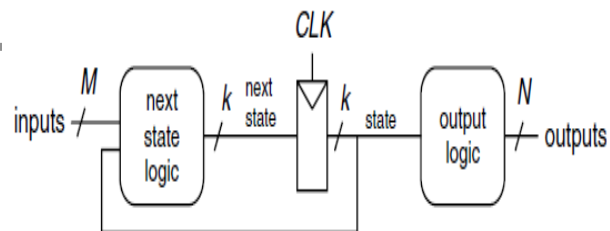
$$L_{B1} = \overline{S_1} \quad L_{B0} = S_1S_0$$





## 2.1 Moore型FSM设计

### (6) 信号灯控制器状态机时序图 (CLK周期 = 5秒)



信号灯控制器状态机时序图

## 第三讲：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. Moore型有限状态机
2. Mealy型有限状态机

### 三. 时序逻辑电路设计与分析

1. 数据寄存器
2. 移位寄存器
3. 计数器

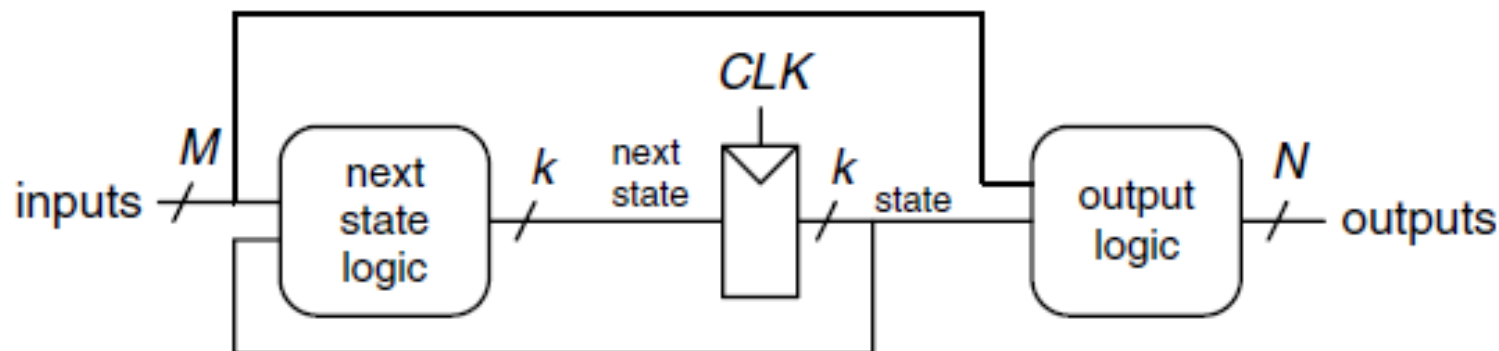
## 2.2 Mealy型FSM设计

### ❖ Mealy型有限状态机

- 输出信号与当前状态及**输入信号**有关
- 输入**M**位，输出**N**位，状态**K**位（可以表示 $2^k$ 个不同状态）
- **次态逻辑**：组合逻辑，输入**M+K**位，输出**K**位次态
- **状态寄存器**：**K**位寄存器
- **输出逻辑**：组合逻辑，输入**M+K**位，输出**N**位信号

### ❖ Mealy型FSM的设计

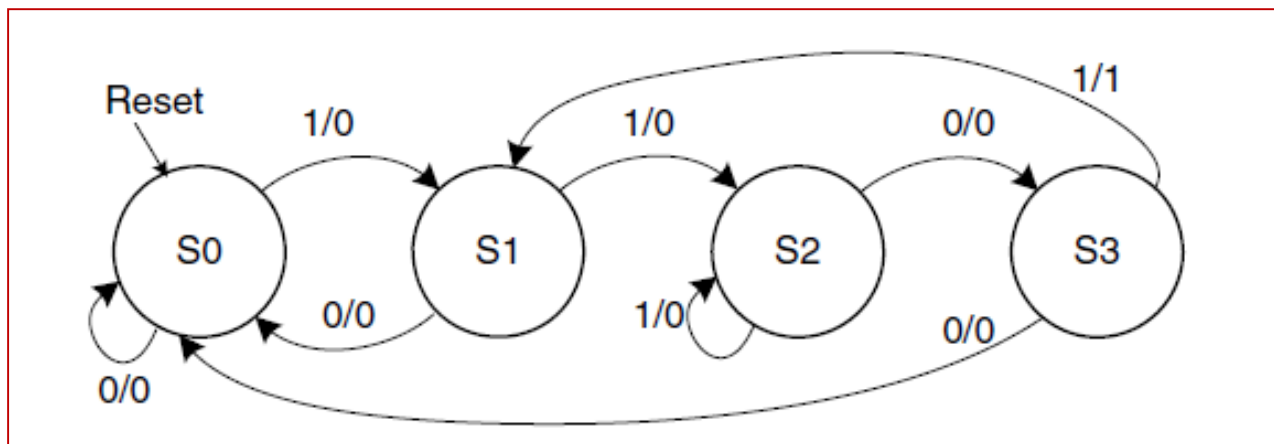
- 确定3个参数：**M**，**K**，**N**
- 设计2个逻辑：次态逻辑（状态转换图）、输出逻辑（真值表）



## 2.2 Mealy型FSM设计

### ❖ Mealy型FSM的表示方法 —— 状态图

- 圆圈表示状态，带箭头的线段表示状态转移
- 状态圈内“S0”、“S1”等代表状态名（对应状态编码）
- 与Moore型FSM不一样的是，输出信号不再标注在圈内，而是以“输入/输出”的形式标注在状态转移线上：“输入”表示引起状态转换的输入信号；“输出”表示状态转换同时产生的输出信号



Mealy型FSM状态转换图

## 2.2 Mealy型FSM设计

【例】二进制序列检测器：当接收到二进制序列“**1101**”时，输出检测标志为**1**，否则输出检测标志为**0**。（不重复检测，即收到**1101**输出**1**后，下一次从下一个输入信号重新开始检测）

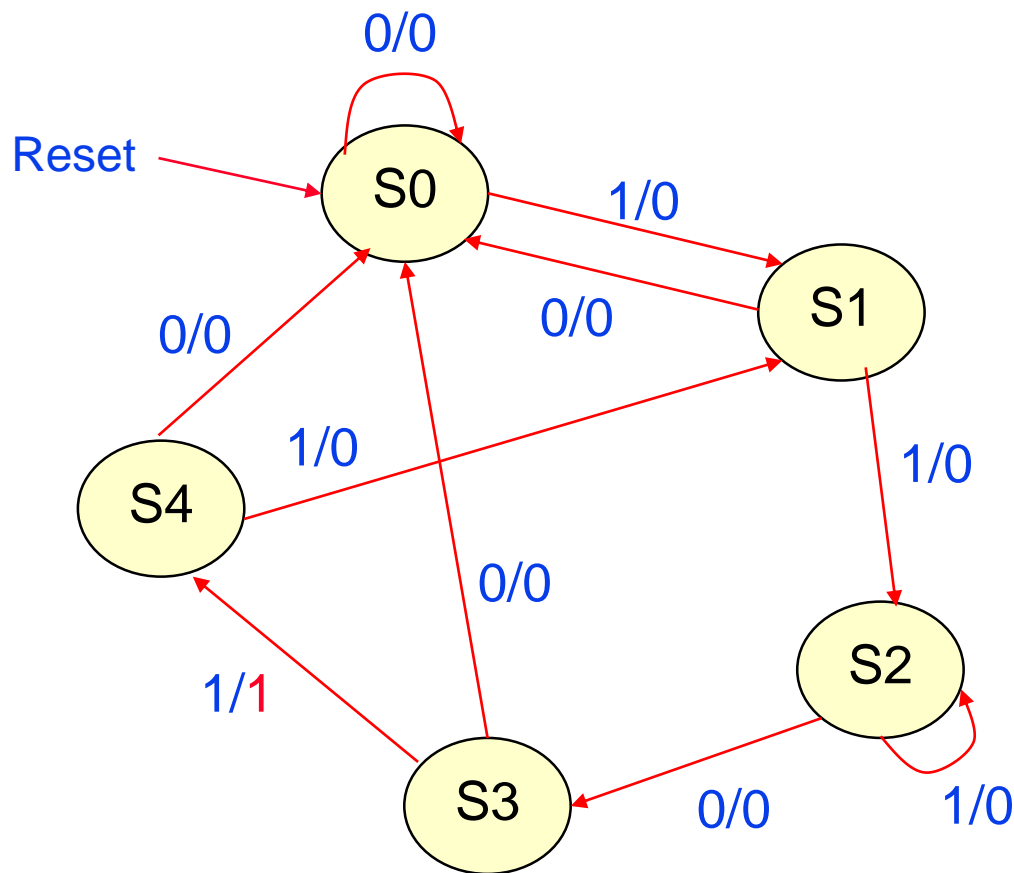
解：

### （1）建立FSM模型

- 输入：二进制序列输入信号 **A**，1位
- 输出：检测标志信号 **Y**，1位
- 状态：共5个不同状态
  - **S0**：未收到第一个有效位（输入为**0**，输出**0**）
  - **S1**：收到第一个有效位（输入为**1**，输出**0**）
  - **S2**：收到第二个有效位（即**S1**后输入为**1**，输出**0**）
  - **S3**：收到第三个有效位（即**S2**后输入为**0**，输出**0**）
  - **S4**：连续收到四个有效位（即**S3**后输入为**1**，输出**1**）
- 状态寄存器：3位

## 2.2 Mealy型FSM设计

### (2) 画出状态转换图



“**1101**” 检测器FSM模型:

**S0**: 未收到第一个有效位

(输入为**0**, 输出**0**)

**S1**: 收到第一个有效位

(输入为**1**, 输出**0**)

**S2**: 收到第二个有效位

(即**S1**后输入为**1**, 输出**0**)

**S3**: 收到第三个有效位

(即**S2**后输入为**0**, 输出**0**)

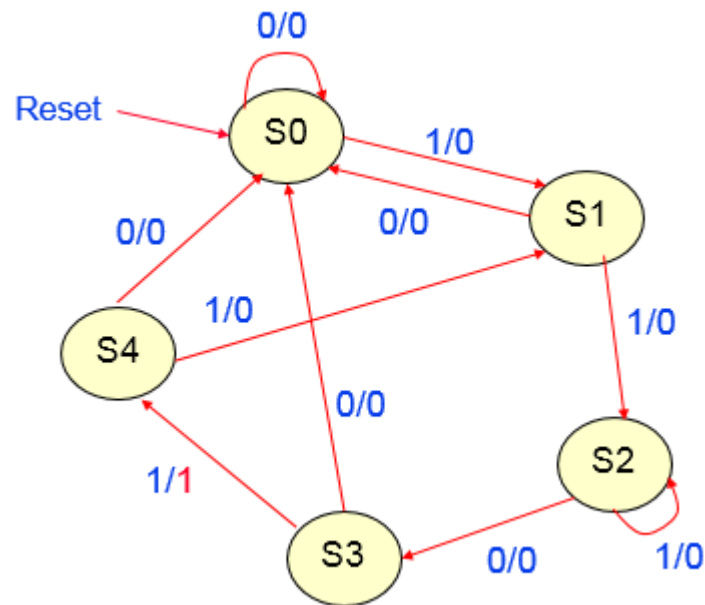
**S4**: 连续收到四个有效位

(即**S3**后输入为**1**, 输出**1**)

## 2.2 Mealy型FSM设计

(3) 根据状态转换图得到状态转换表

当前状态 ( $S_2S_1S_0$ )	输入 (A)	下一状态 ( $S'_2S'_1S'_0$ )	输出 (Y)
S0 (000)	0	S0 (000)	0
S0 (000)	1	S1 (001)	0
S1 (001)	0	S0 (000)	0
S1 (001)	1	S2 (010)	0
S2 (010)	0	S3 (011)	0
S2 (010)	1	S2 (010)	0
S3 (011)	0	S0 (000)	0
S3 (011)	1	S4 (100)	1
S4 (100)	0	S0 (000)	0
S4 (100)	1	S1 (001)	0



## 2.2 Mealy型FSM设计

(4) 根据状态转换表写出 次态逻辑表达式 和 输出逻辑表达式

$$S_2' = \overline{S_2} S_1 S_0 A$$

$$\begin{aligned} S_1' &= \overline{S_2} \overline{S_1} S_0 A + \overline{S_2} S_1 \overline{S_0} A + \overline{S_2} S_1 \overline{S_0} A \\ &= \overline{S_2} \overline{S_1} S_0 A + \overline{S_2} S_1 \overline{S_0} \end{aligned}$$

$$\begin{aligned} S_0' &= \overline{S_2} \overline{S_1} S_0 A + \overline{S_2} S_1 \overline{S_0} A + S_2 \overline{S_1} \overline{S_0} A \\ &= \overline{S_1} \overline{S_0} A + \overline{S_2} S_1 \overline{S_0} A \end{aligned}$$

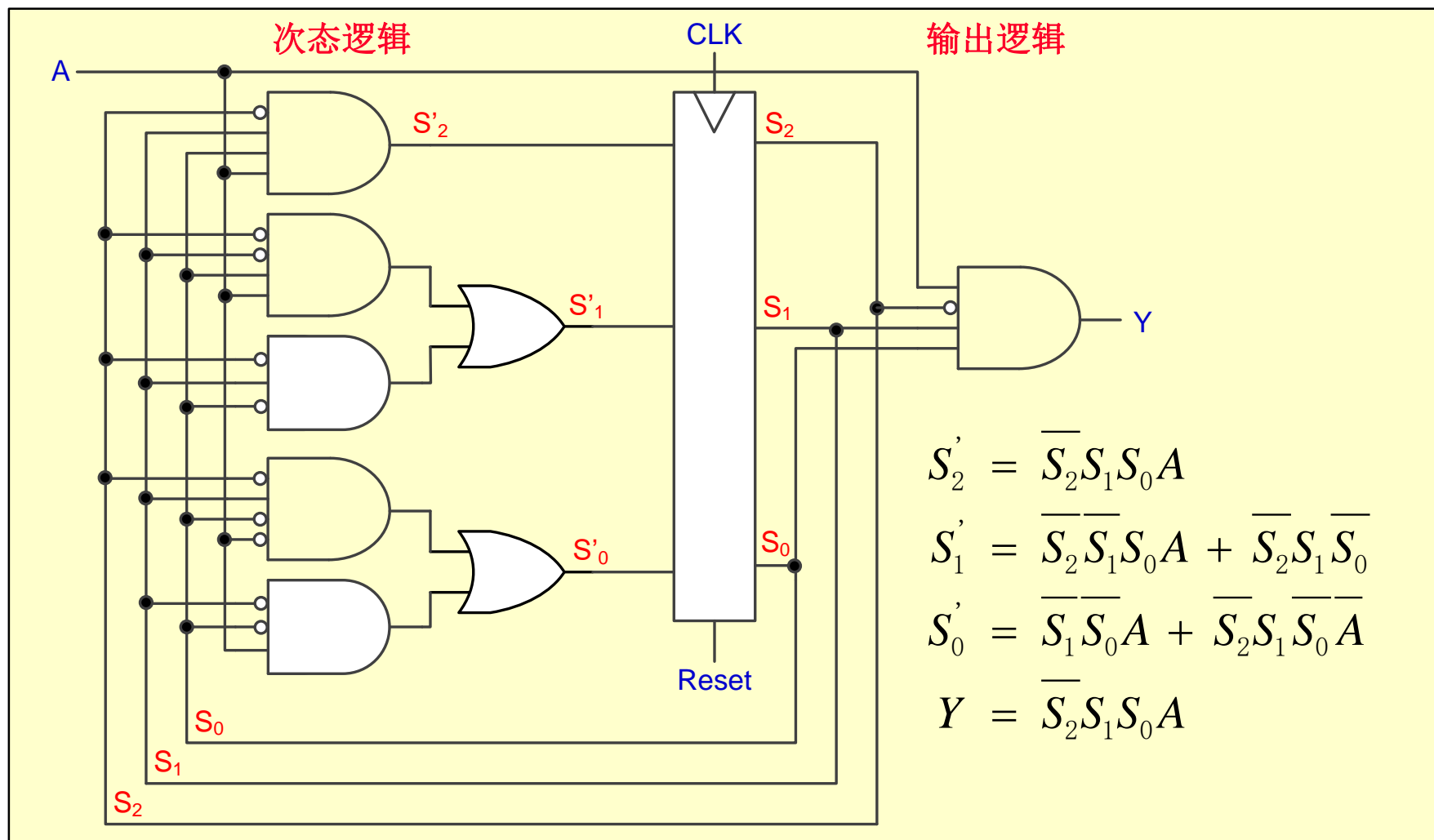
$$Y = \overline{S_2} S_1 S_0 A$$

当前状态 ( $S_2 S_1 S_0$ )	输入 ( $A$ )	下一状态 ( $S_2' S_1' S_0'$ )	输出 ( $Y$ )
S0 (000)	0	S0 (000)	0
S0 (000)	1	S1 (001)	0
S1 (001)	0	S0 (000)	0
S1 (001)	1	S2 (010)	0
S2 (010)	0	S3 (011)	0
S2 (010)	1	S2 (010)	0
S3 (011)	0	S0 (000)	0
S3 (011)	1	S4 (100)	1
S4 (100)	0	S0 (000)	0
S4 (100)	1	S1 (001)	0



## 2.2 Mealy型FSM设计

(5) 根据逻辑表达式画出 逻辑图



## 2.2 Mealy型FSM设计

### (6) 序列检测器的Verilog HDL设计

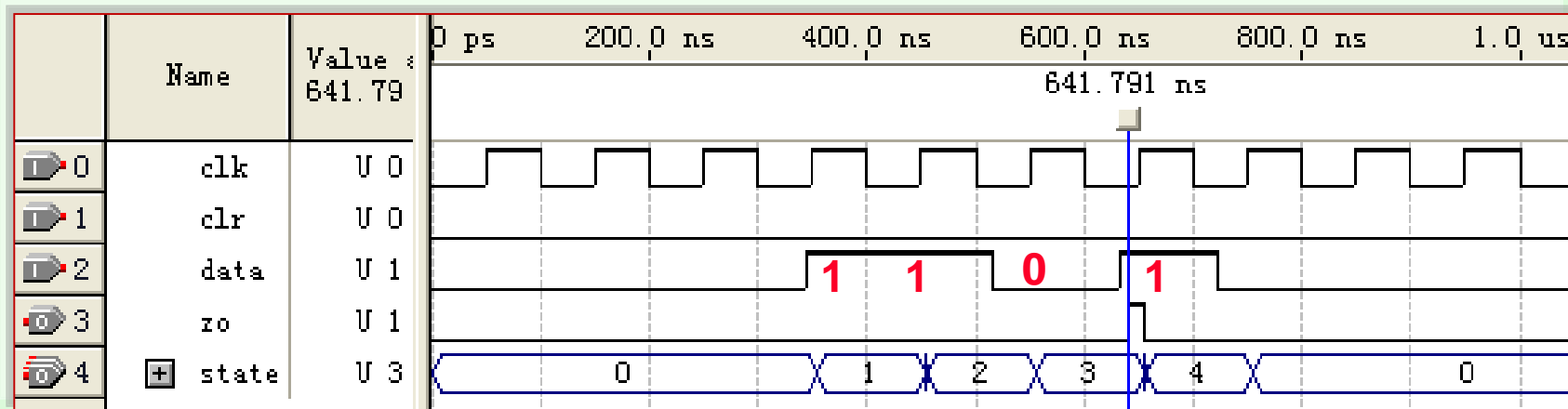
```
module monitor2_good(clk,clr,data,zo,state);
    parameter S0=3'b000, S1=3'b001,
               S2=3'b010, S3=3'b011, S4=3'b100;
    input clk,clr,data;
    output zo;
    output[2:0] state;
    reg [2:0] state;
    reg zo;
    always @(posedge clk or posedge clr)
        begin
            if (clr) state=S0; // 复位时回到初始状态
            else
                begin
                    case (state) // 状态的转移
                        S0: if (data==1'b1) state=S1; else state=S0;
                        S1: if (data==1'b1) state=S2; else state=S0;
                        S2: if (data==1'b0) state=S3; else state=S2;
                        S3: if (data==1'b1) state=S4; else state=S0;
                        S4: if (data==1'b1) state=S1; else state=S0;
                        default: state=S0;
                    endcase
                end
        end
end
```

```
always @(state) // 状态机的输出
begin
    case (state)
        S0: zo=1'b0;
        S1: zo=1'b0;
        S2: zo=1'b0;
        S3: if (data==1'b1) zo=1'b1;
            else zo=1'b0;
        S4: zo=1'b0;
        default: zo=1'b0;
    endcase
end
endmodule
```

在**S3**时，若输入  
**data=1**，则**zo**置1：  
输出是内部状态和外部输入的函数

## 2.2 Mealy型FSM设计

### (7) 序列检测器的仿真波形



## 2.3 有限状态机的状态编码问题

### ❖ 起始状态的选择:

- 起始状态指电路复位后所处的初态，选择合适的起始状态将使设计简捷高效。FSM必须有**时钟**信号和**复位**信号。

### ❖ 状态编码方式的选择（假定FSM有N个状态）

- **二进制编码**：采用 $\log_2 N$ 个**触发器**来表示这N个状态，按二进制顺序编码，节省逻辑资源，但可能产生输出毛刺。
- **格雷编码**：采用 $\log_2 N$ 个**触发器**来表示这N个状态，但**相邻状态只有一个比特位不同**。节省逻辑资源，降低了输出毛刺的可能，状态转换中，相邻状态只有一个比特位产生变化。
- **一位热码状态机编码（One-Hot State Machine Encoding）**：采用**N个触发器**来表示这N个状态。逻辑资源消耗最大，但可以避免状态机产生错误的输出，并且有时可简化输出逻辑。

## 2.3 有限状态机的状态编码问题

对8个状态三种编码方式的对比

状态	二进制编码	格雷编码	一位热码编码
state0	000	000	0000000 <b>1</b>
state1	001	001	000000 <b>10</b>
state2	010	011	00000 <b>100</b>
state3	011	010	0000 <b>1000</b>
state4	100	110	000 <b>10000</b>
state5	101	111	00 <b>100000</b>
state6	110	101	0 <b>1000000</b>
state7	111	100	<b>10000000</b>

- ❖ 采用一位热码编码，虽使用触发器较多，但可简化组合逻辑电路。
- ❖ **FPGA**有丰富的寄存器资源，门逻辑相对缺乏，采用一位热码编码可有效提高电路的速度和可靠性，也有利于提高器件资源的利用率。

## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. Moore型有限状态机
2. Mealy型有限状态机

### 三. 时序逻辑电路设计分析

1. 数据寄存器
2. 移位寄存器
3. 计数器
4. 时序电路的时序

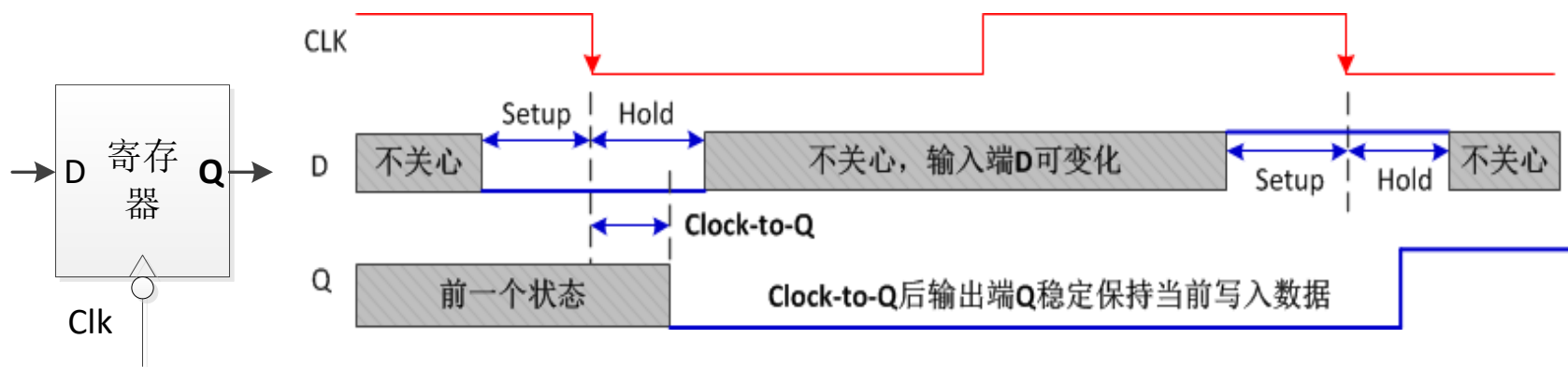
### 3 时序电路的时序



照相机拍照的孔径时间（**aperture time**）内，被照对象必须保持稳定，否则不能拍到清晰的照片。

### 3 时序电路的时序

#### ❖ 寄存器的时序

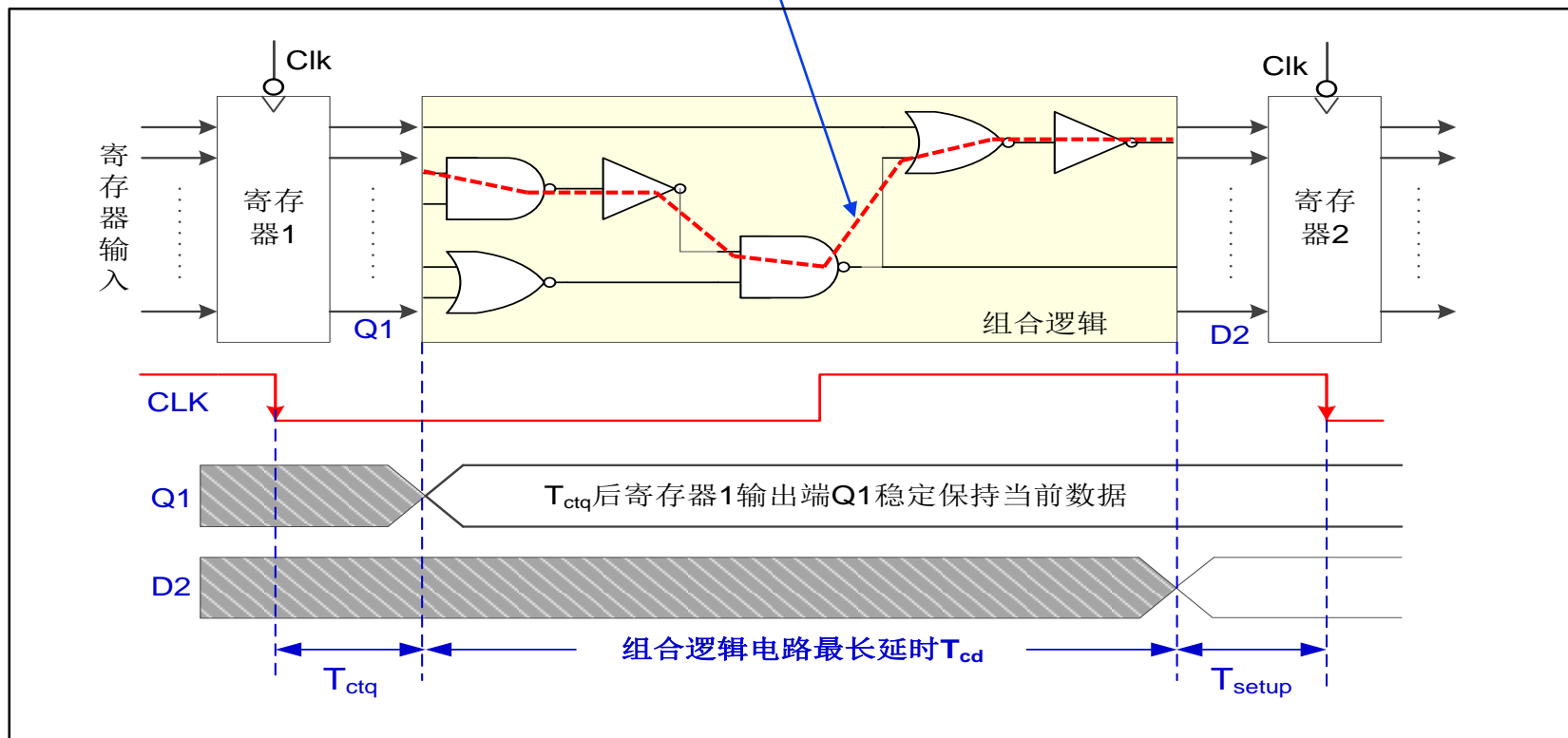


- **建立时间  $T_{\text{setup}}$  (Setup Time):** 触发时钟沿之前, 输入必须稳定的时间
- **保持时间  $T_{\text{hold}}$  (Hold Time):** 触发时钟沿之后, 输入仍需稳定的时间
- **孔径时间 =  $T_{\text{setup}} + T_{\text{hold}}$ ,** 输入信号在孔径时间内必须稳定不变
- **Clock-to-Q时间  $T_{\text{ctq}}$ :** 从触发时钟边沿到输出稳定的时间



### 3 时序电路的时序

#### ❖ 同步时序电路时钟周期

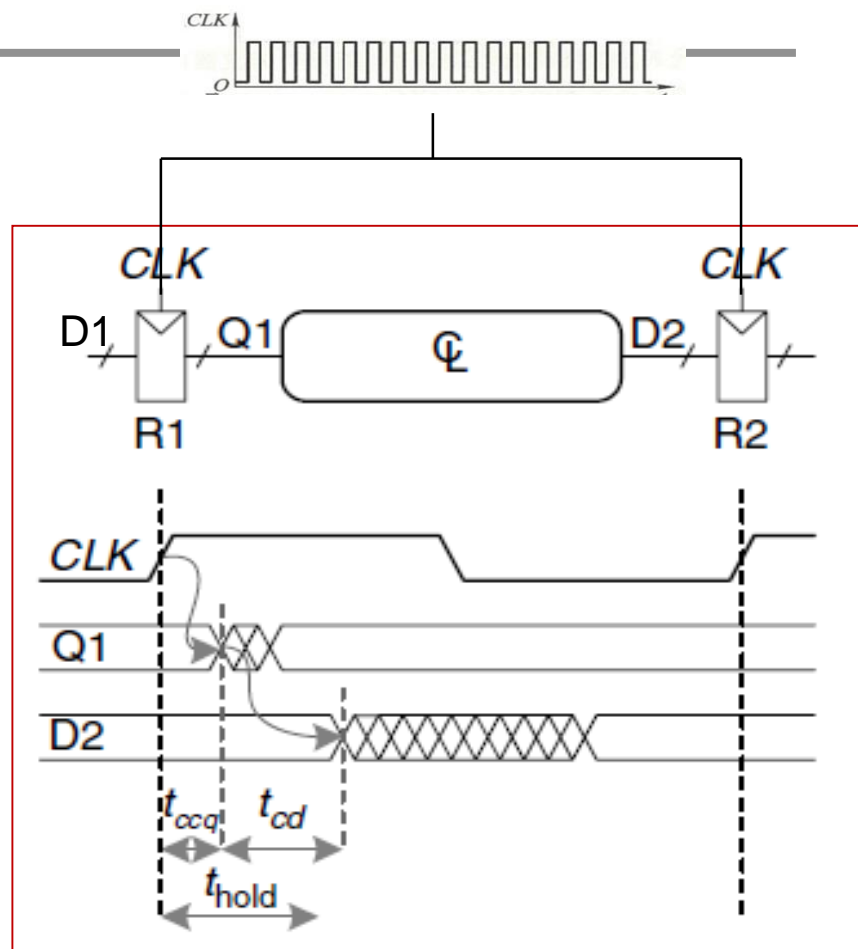


- 时序电路由“寄存器元件+ 操作元件(组合电路)+ 寄存器元件”组成
- 时钟周期  $T_C \geq T_{ctq} + T_{cd}(\text{最长路径延时}) + T_{setup} + \text{时钟偏移}$
- 时钟偏移：由于时钟信号源到各个寄存器部件的连线长度不同等原因所引起的各个寄存器时钟信号达到的细微时间差异，一般忽略

### 3 时序电路的时序

#### ❖ 保持时间约束

- 右图中寄存器R2有保持时间约束，在时钟有效沿之后的  $t_{\text{hold}}$  之内，D2必须保持稳定不变
- 时钟有效沿之后  $t_{\text{ccq}}$  时，R1的输出Q1将发生变化（但不一定稳定），这种变化经过组合逻辑电路延迟  $t_{\text{cd}}$  (最短路径延时) 后传播到D2，使得D2发生变化
- 保持时间约束为  $t_{\text{ccq}} + t_{\text{cd}} \geq t_{\text{hold}}$ ，否则R2不能正确锁存上一个时钟周期内所形成D2的输入，因为在上一个D2的保持时间内，D2就发生了变化



### 3 时序电路的时序

【例】假定下面的电路触发器时钟到Q的最小延迟和稳定时间分别为**30ps**和**80ps**，建立时间和保持时间分别为**50ps**和**60ps**，每个门电路的最小延迟和最大延迟分别是**25ps**和**40ps**。该时序电路的最小时钟周期是多少？并对时序电路进行时序分析。

解：

(1)  $T_{ctq}=80, T_{setup}=50$

组合逻辑最大延迟  $T_{cd}=3*40=120$

所以最小时钟周期为：

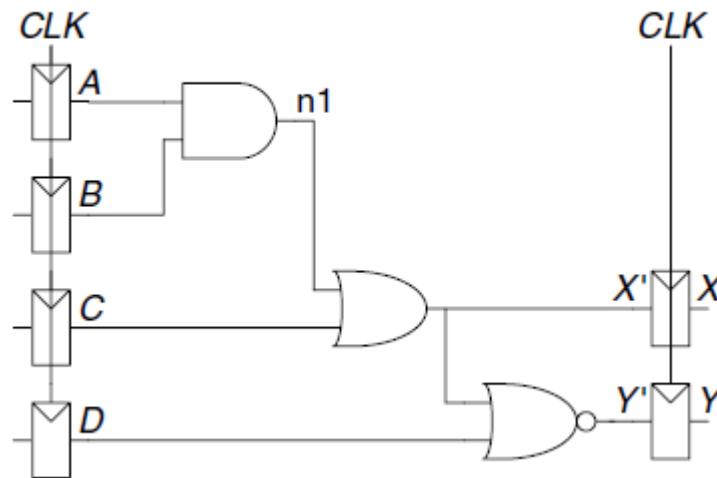
$$\begin{aligned} T_C &= T_{ctq} + T_{cd} + T_{setup} \\ &= 80 + 120 + 50 = 250ps \end{aligned}$$

(2)  $T_{ccq}=30, T_{hold}=60$

组合逻辑最小延迟为**25**（D的改变最快经过**25ps**可能引起Y' 改变）。

$T_{ccq}+25=55 < T_{hold}$  违背了保持约束，Y '值（X' 值也同样有可能）不能保持足够长的稳定时间，所以Y值实际上不可预测。

因此，该电路在任何时钟周期下其功能都有可能不正确。



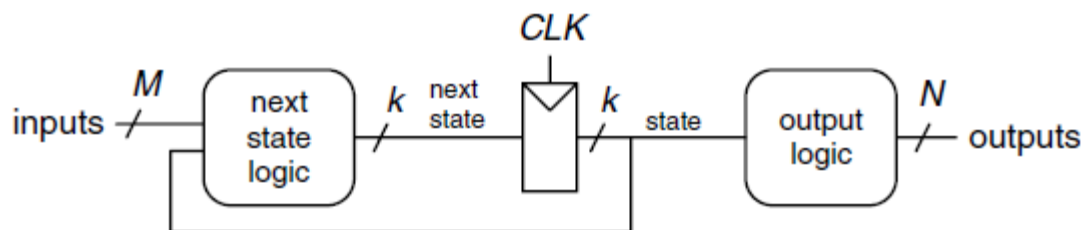
# 第八讲

---

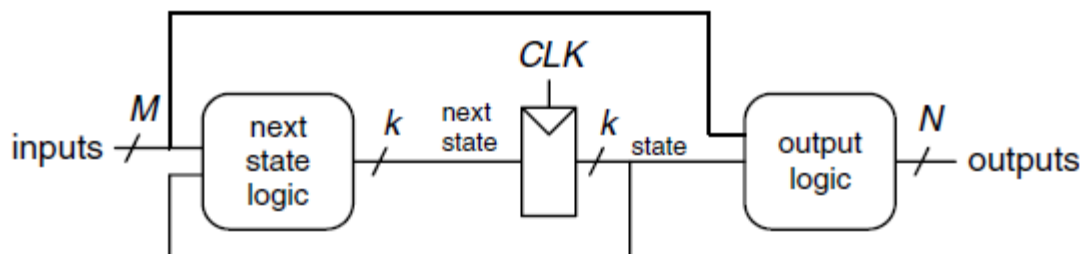
## 上一讲简要回顾 —— 有限状态机及其设计

❖ 根据输出信号产生的机理不同，有限状态机可分成两类：

- 摩尔 (Moore) 型状态机： 输出信号仅与当前状态有关
- 米里 (Mealy) 型状态机： 输出信号与当前状态及输入信号有关



Moore型有限状态机



Mealy型有限状态机

## 上一讲简要回顾 —— 有限状态机及其设计

1. 建模：确定输入、输出，以及状态总数

2. 画出状态转换图

3. 对于**Moore型FSM**

① 写出状态转换表

② 写出输出真值表

对于**Mealy型FSM**

① 写出组合的状态转换表和输出真值表

4. 选择状态编码 —— 这个选择将影响硬件设计

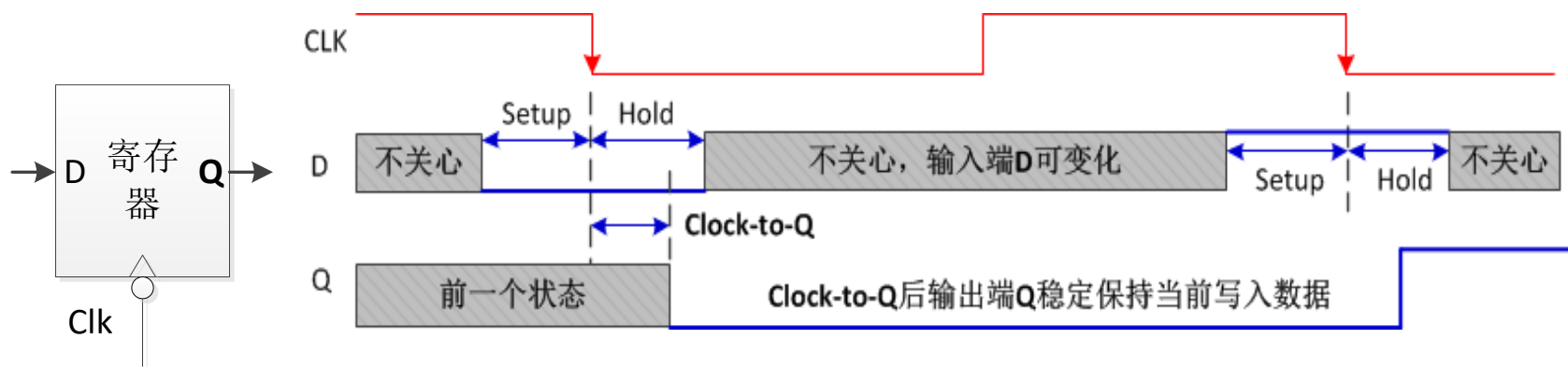
5. 写出次态逻辑和输出逻辑的逻辑表达式

6. 画出电路图

7. 进行**Verilog HDL**设计并仿真测试

## 上一讲简要回顾 —— 时序电路的时序

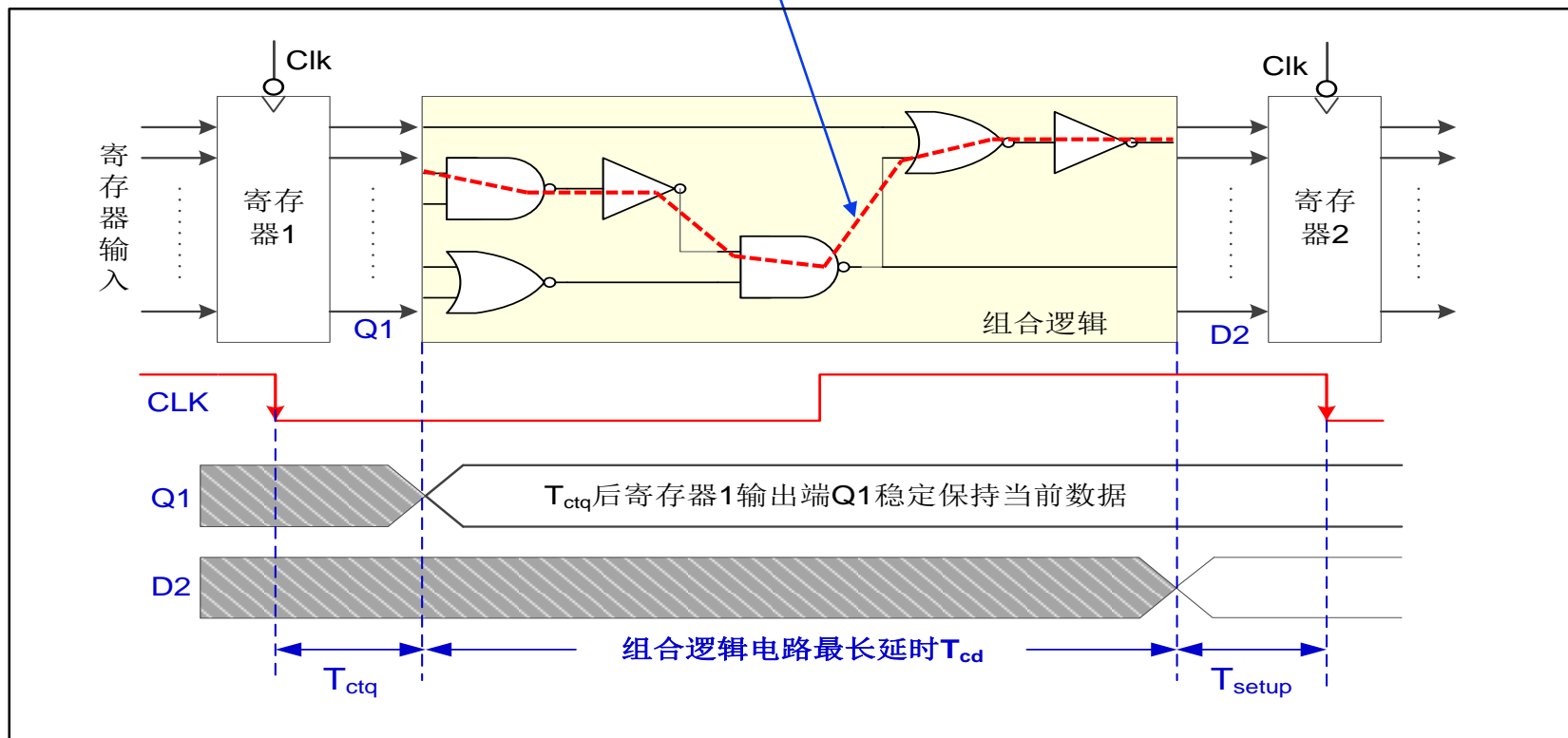
### ❖ 寄存器的时序



- **建立时间  $T_{\text{setup}}$  (Setup Time):** 触发时钟沿之前，输入必须稳定的时间
- **保持时间  $T_{\text{hold}}$  (Hold Time):** 触发时钟沿之后，输入仍需稳定的时间
- **孔径时间 =  $T_{\text{setup}} + T_{\text{hold}}$ ,** 输入信号在孔径时间内必须稳定不变
- **Clock-to-Q时间  $T_{\text{ctq}}$ :** 从触发时钟边沿到输出稳定的时间
- **$T_{\text{ccq}}$ :** 时钟有效沿之后  $T_{\text{ccq}}$  时间，输出Q将发生变化（但不一定稳定）

## 上一讲简要回顾 —— 时序电路的时序

### ❖ 同步时序电路时钟周期



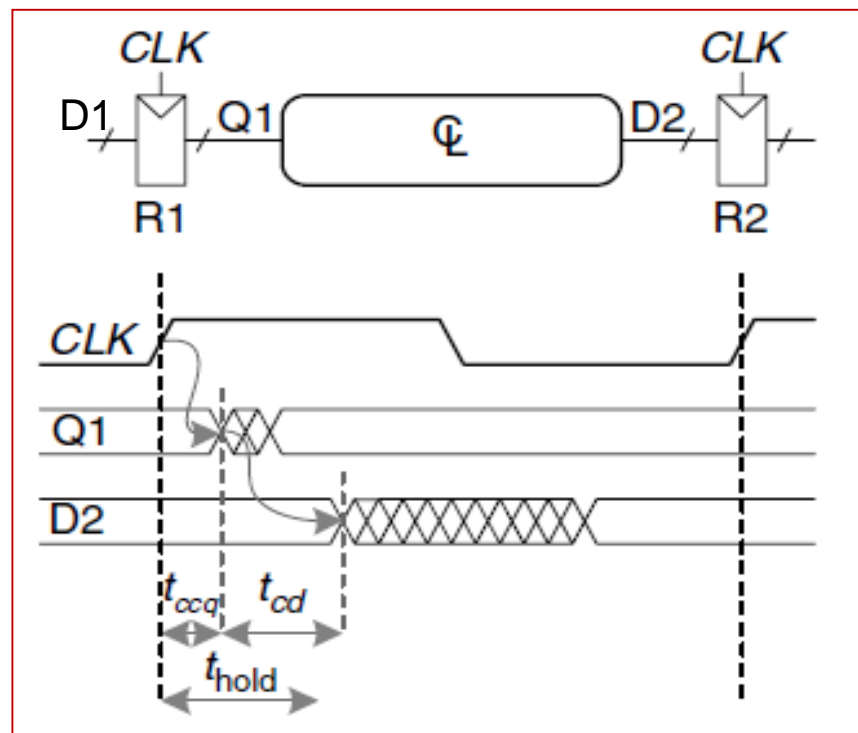
- 时序电路由“寄存器元件+ 操作元件(组合电路)+ 寄存器元件”组成
- 时钟周期  $T_C \geq T_{ctq} + T_{cd} + T_{setup} + \text{时钟偏移}$
- 时钟偏移：由于时钟信号源到各个寄存器部件的连线长度不同等原因所引起的各个寄存器时钟信号达到的细微时间差异，一般忽略



# 时序电路的时序

## ❖ 保持时间约束

- 右图中寄存器R2有保持时间约束，在时钟有效沿之后的  $t_{\text{hold}}$  之内，D2必须保持稳定不变
- 时钟有效沿之后  $t_{\text{ccq}}$  时，R1的输出Q1将发生变化（但不一定稳定），这种变化经过组合逻辑电路延迟  $t_{\text{cd}}$  后传播到D2，使得D2发生变化
- 保持时间约束为  $t_{\text{ccq}} + t_{\text{cd}} \geq t_{\text{hold}}$ ，否则R2不能正确锁存上一个时钟周期内所形成D2的输入，因为在上一个D2的保持时间内，D2就发生了变化



## 时序电路的时序【例】

【例】假定下面的电路触发器时钟到Q的最小延迟和稳定时间分别为**30ps**和**80ps**，建立时间和保持时间分别为**50ps**和**60ps**，每个门电路的最小延迟和最大延迟分别是**25ps**和**40ps**。该时序电路的最小时钟周期是多少？并对时序电路进行时序分析。

解：

(1)  $T_{ctq}=80, T_{setup}=50$

组合逻辑最大延迟  $T_{cd}=3*40=120$ 。

所以最小时钟周期为：

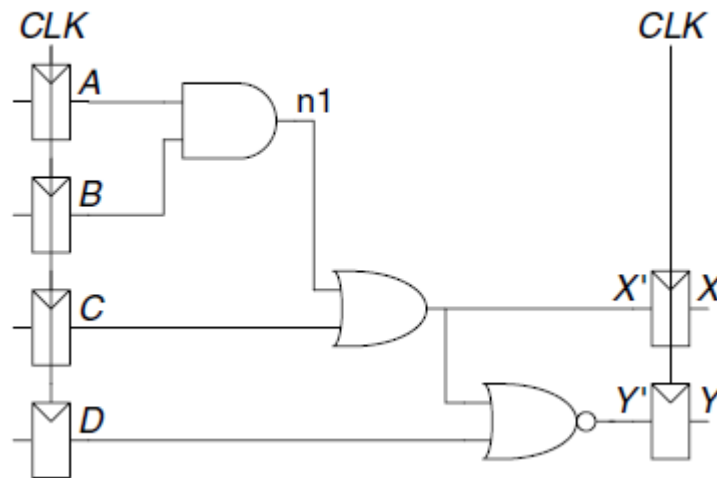
$$\begin{aligned} T_C &= T_{ctq} + T_{cd} + T_{setup} \\ &= 80 + 120 + 50 = 250ps \end{aligned}$$

(2)  $T_{ccq}=30, T_{hold}=60$

组合逻辑最小延迟为**25**（D的改变最快经过**25ps**可能引起Y' 改变）。

$T_{ccq}+25=55 < T_{hold}$  违背了保持约束，Y '值（X' 值也同样有可能）不能保持足够长的稳定时间，所以Y值实际上不可预测。

因此，该电路在任何时钟周期下其功能都有可能不正确。



## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. Moore型有限状态机
2. Mealy型有限状态机

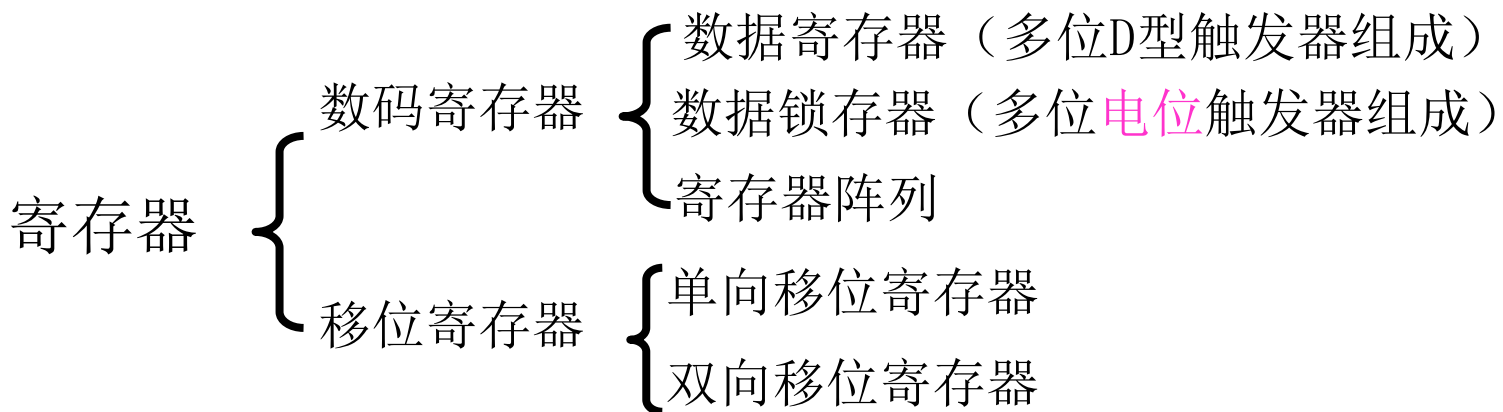
### 三. 时序逻辑电路设计与分析

1. 数据寄存器 / 数据锁存器
2. 移位寄存器
3. 计数器
4. 时序电路的时序

# 寄存器的分类

## ❖ 寄存器

- 计算机中重要部件，可用于存放一组二进制代码，如：指令、参加运算的操作数、运算结果等，广泛用于各类数字系统中
- 基本组成：由触发器和控制门电路组成，其中：一个触发器能储存1位二进制代码；接收数据的控制门电路，控制在同一个接收命令作用下，使各触发器同时接收数据
- 基本操作：写 / 读 / 复位（清零）
- 触发方式：由组成寄存器的触发器的触发方式决定：边沿或电位触发



## 3.1.1 数据寄存器

### ❖ 数据寄存器:

- 由多个边沿触发器组成的用于保存一组二进制代码的寄存单元。
- 当时钟信号的有效沿（上升沿或下降沿）到来时，将输入端数据打入寄存器，此时输出信号等于输入信号；
- 在时钟信号的其它时刻，输出端保持刚才输入的数据，即为寄存状态，而不管此时输入信号是否变化。

### ❖ 4位D型寄存器结构

$D_3D_2D_1D_0$ : 并行数据输入

$Q_3Q_2Q_1Q_0$ : 并行数据输出

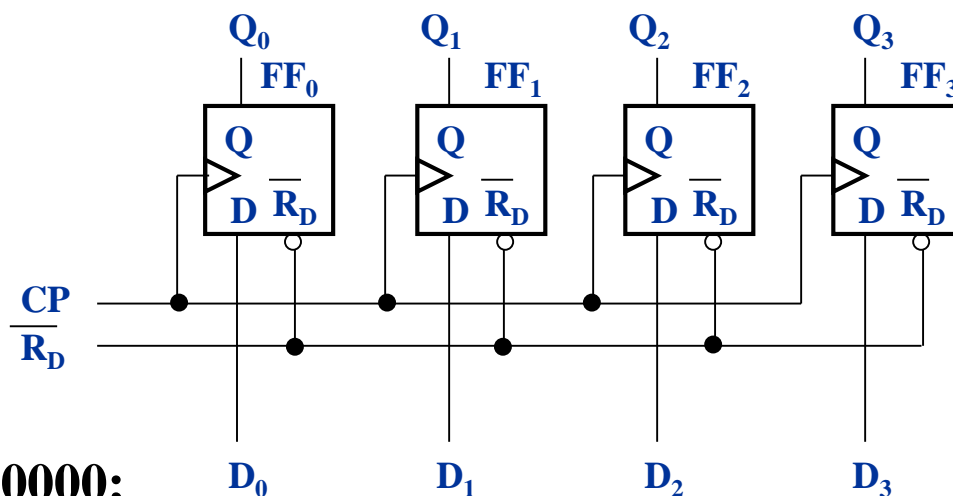
工作原理:

(1) 清除（复位）

当  $\overline{R_D}=0$  ,  $Q_0Q_1Q_2Q_3=0000$ ;

(2) 置数（复位端无效时）

当CP上跳沿到达时,  $Q_0Q_1Q_2Q_3= D_0D_1D_2D_3$



### 3.1.1 数据寄存器 —— HDL设计

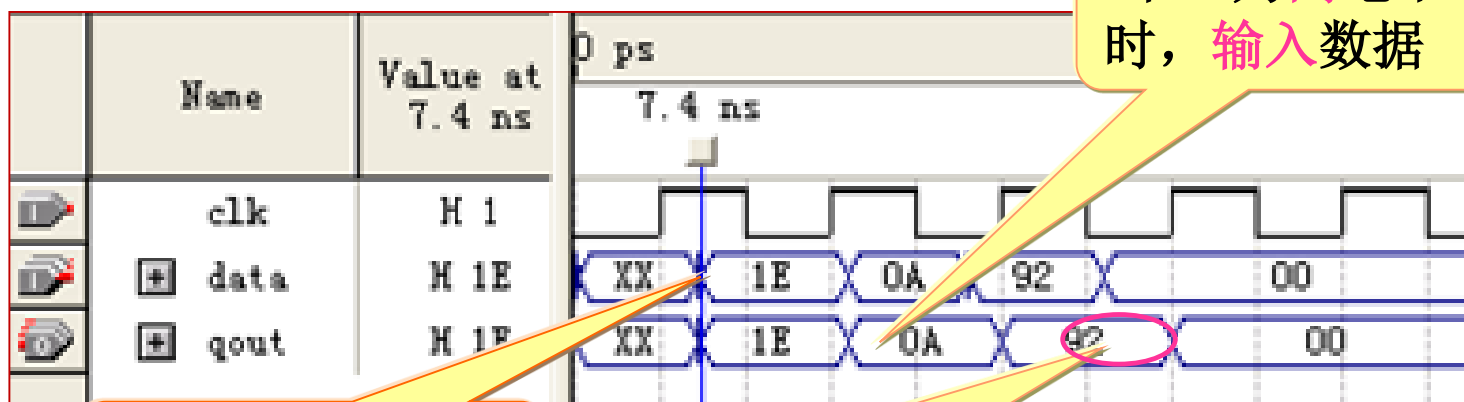
【例】用always块语句描述的8位数据寄存器

```
module reg_8bit(qout,data,clk,clr);  
    output[7:0] qout;  
    input [7:0] data;  
    input clk,clr;  
    reg [7:0] qout;  
    always @(posedge clk or posedge clr) //沿触发  
    begin  
        if (clr) qout = 0; //异步清零  
        else    qout = data;  
    end  
endmodule
```

### 3.1.2 数据锁存器

❖ **数据锁存器**：由多位**电位**触发器组成的用于保存一组二进制代码的寄存单元。

- **功能**：（1）当输入控制信号（如时钟）为**高**电平时，门是**打开**的，输出信号等于输入信号；（2）当输入控制信号为**低**电平时，门是**关闭**的，输出端保持刚才输入的数据，即为**锁存**状态，而不管此时输入信号是否变化。
- 通常由**电平**信号来控制，属于**电平**敏感型，适于数据有效**滞后**于控制信号有效的场合。



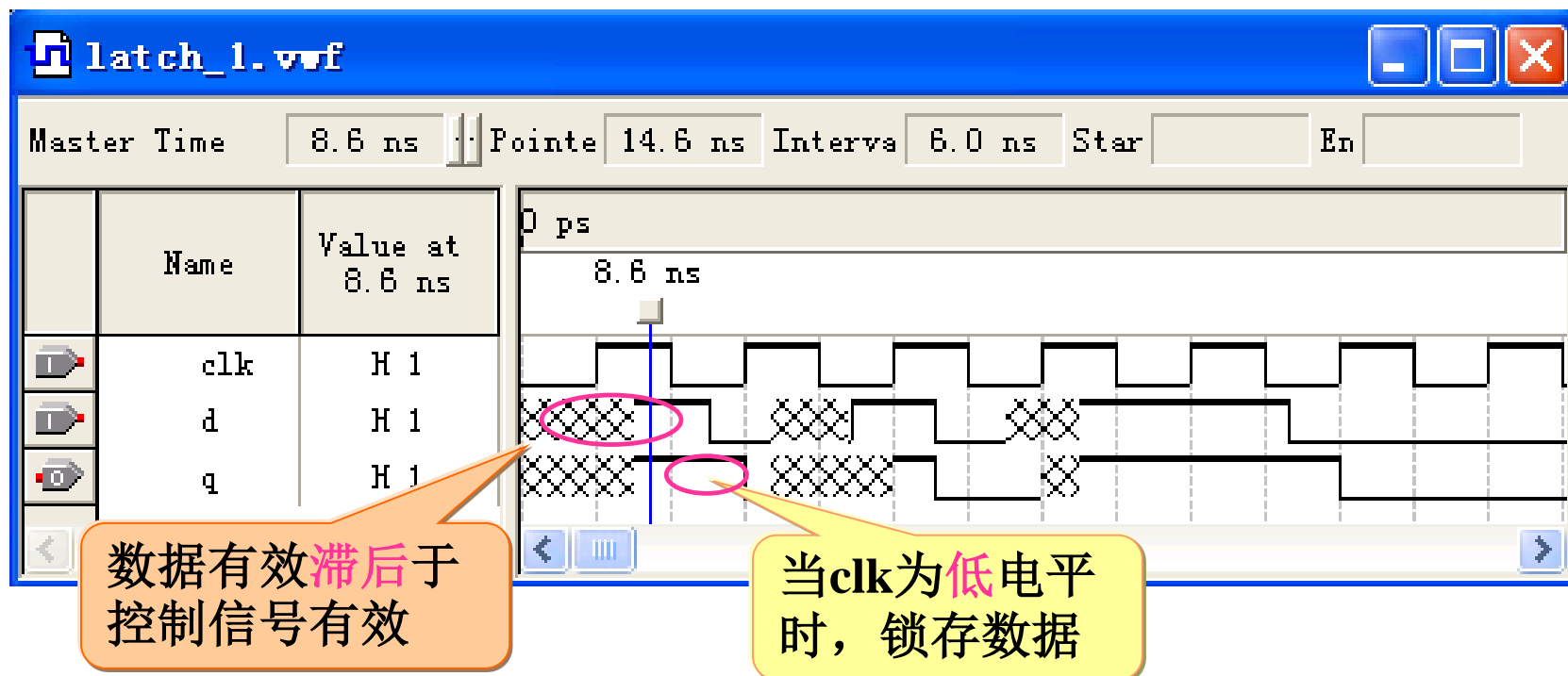
数据有效 滞后于  
控制信号有效

当clk为低电平时，  
锁存数据

## 3.1.2 数据锁存器 —— HDL设计

### 【例】1位数据锁存器的Verilog HDL设计

```
module latch_1(q,d,clk);  
    output q;  
    input d, clk ;  
    assign q = clk ? d : q; /* 时钟信号为高电平，打入数据，否则锁存原数 */  
endmodule
```

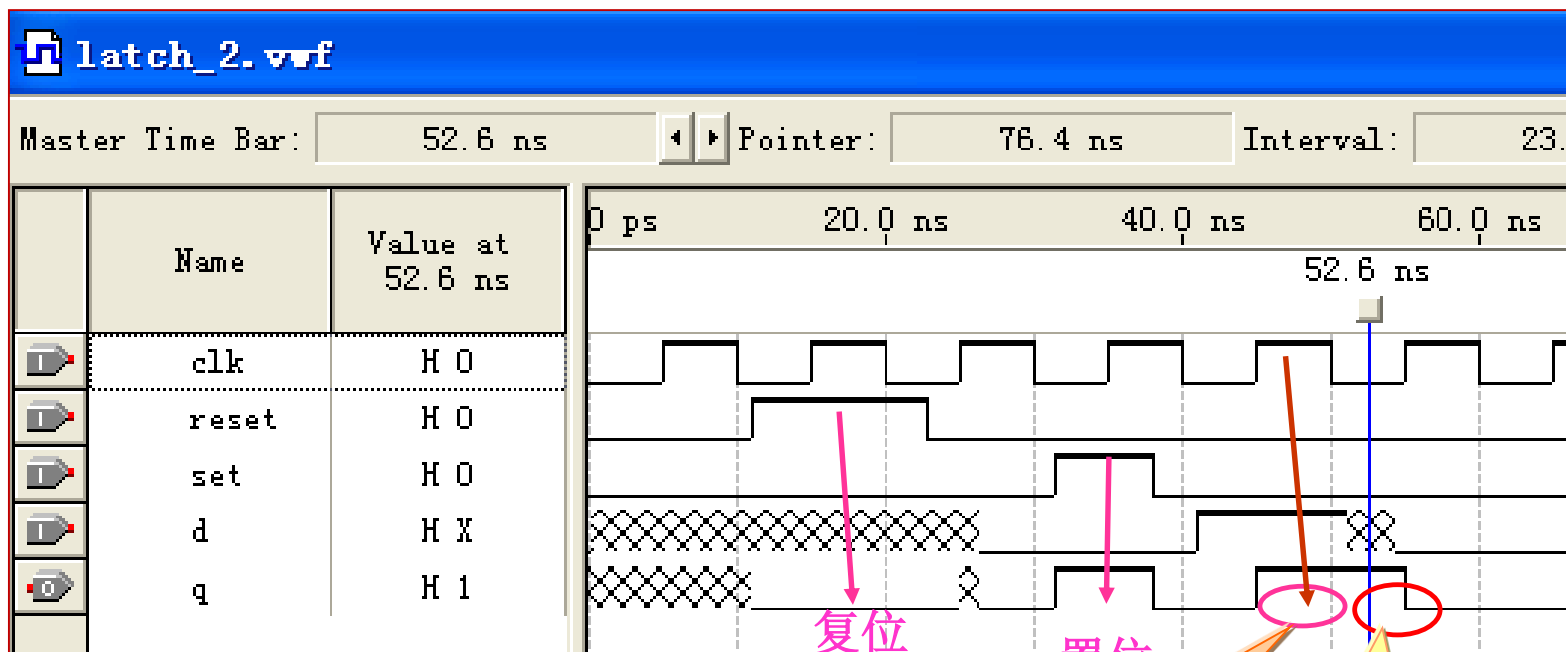




## 3.1.2 数据锁存器 —— HDL设计

### 【例】带置位和复位端的1位数据锁存器的Verilog HDL

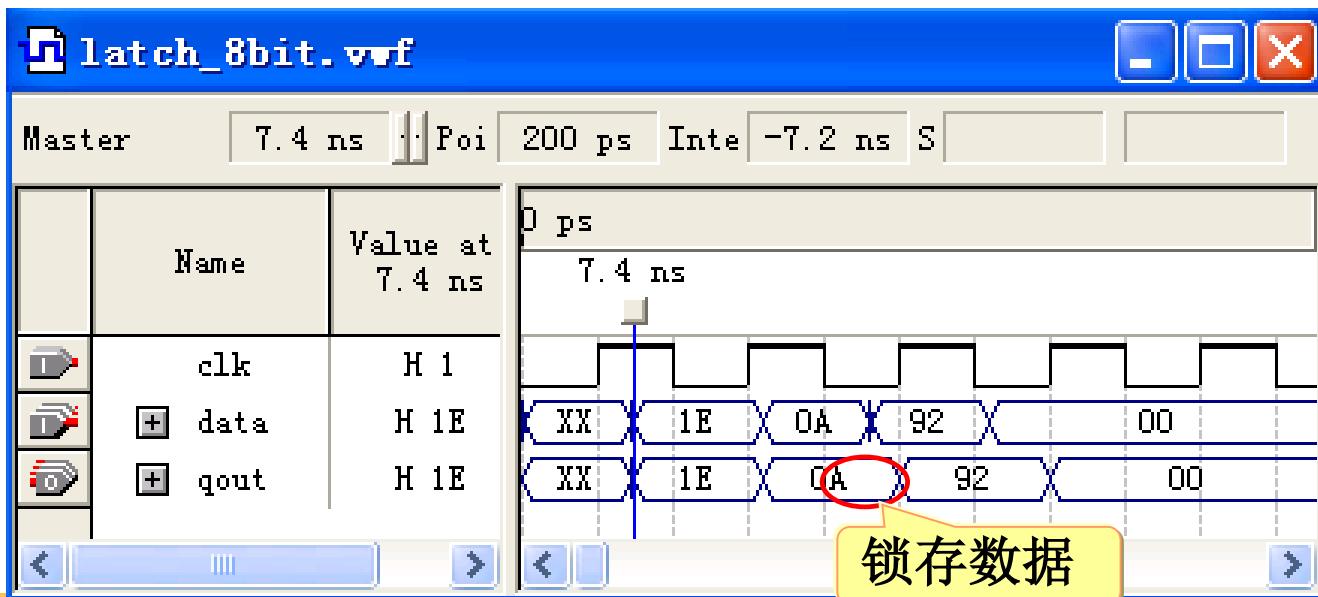
```
module latch_2(q,d,clk,set,reset);  
    output q;  
    input d,clk ,set,reset;  
    assign q= reset ? 0: (set ? 1:(clk ? d:q));  
endmodule
```



## 3.1.2 数据锁存器 —— HDL设计

### 【例】8位数据锁存器的Verilog HDL设计

```
module latch_8bit(qout,data,clk);  
    output [7:0] qout;  
    input [7:0] data;  
    input clk;  
    reg [7:0] qout;  
    always @(clk or data) //电平敏感  
        if (clk) qout = data;  
endmodule
```



## 3.1.2 数据锁存器

### ❖ 数据寄存器和数据锁存器的区别

#### ➤ 数据寄存器：

- 由边沿触发的触发器（D触发器）组成。
- 通常由同步时钟信号来控制，属于脉冲敏感型
- 适于数据有效提前于控制信号（一般为时钟信号）有效、并要求同步操作的场合。

#### ➤ 数据锁存器：

- 由电平触发器（D锁存器）组成。
- 一般由电平信号来控制，属于电平敏感型
- 适于数据有效滞后于控制信号有效的场合

## 第三讲：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. Moore型有限状态机
2. Mealy型有限状态机

### 三. 时序逻辑电路设计分析

1. 数据寄存器/数据锁存器
2. 移位寄存器
3. 计数器
4. 时序电路的时序

## 3.2 移位寄存器

❖ **移位寄存器**：具有移位功能的寄存器。每来一个时钟脉冲，寄存器中数据就依次向左或向右移一位。

❖ **应用**：计算机中经常需要用到移位操作，如：

- 乘法运算中的右移、除法运算中的左移等
- 数据输入输出方式中的串行并行转换等

❖ **分类**：

- 左移移位寄存器
- 右移移位寄存器
- 双向移位寄存器
- 循环移位寄存器

## 3.2 移位寄存器

### 1. 4位右移移位寄存器

- 4个D触发器组成
- 高位输出送低位输入，如  $Q_0 \rightarrow D_1, Q_1 \rightarrow D_2, \dots$

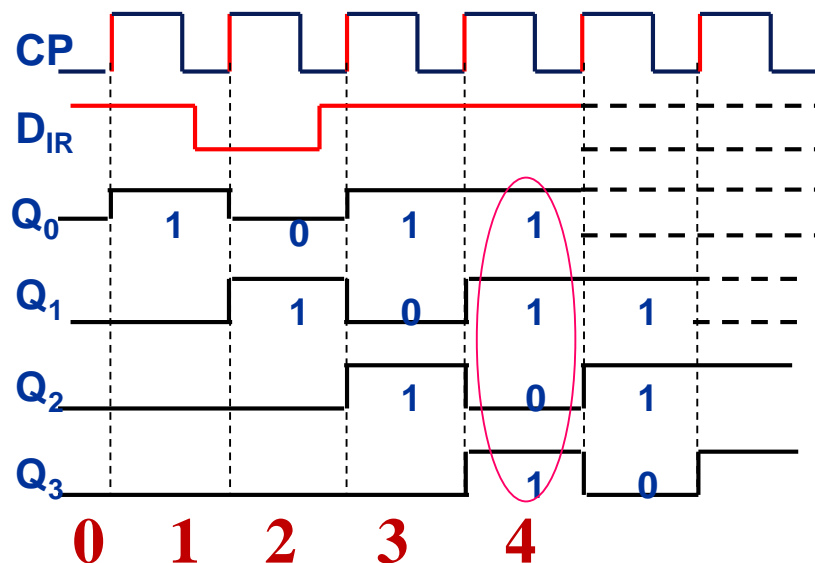
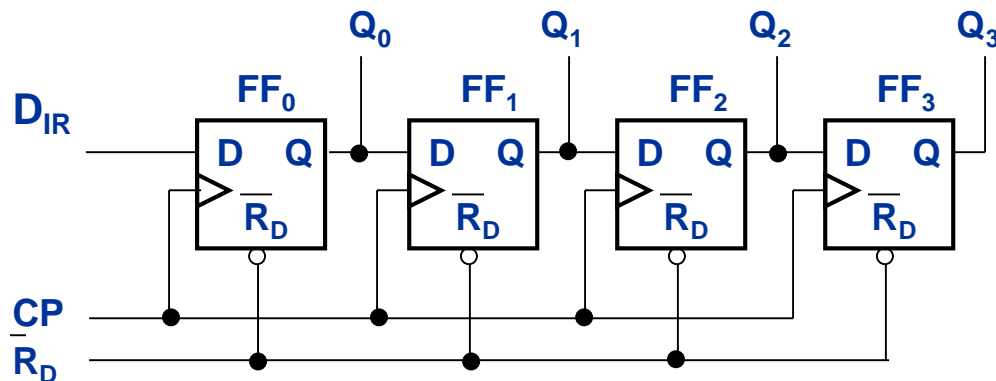
#### 工作原理

- 复位:

$$\overline{R_D} = 0 \rightarrow Q_3 Q_2 Q_1 Q_0 = 0000$$

- 移位:

$$\begin{aligned} Q_0^{n+1} &= D_0 \text{CP} \uparrow = D_{IR} \text{CP} \uparrow \\ Q_1^{n+1} &= D_1 \text{CP} \uparrow = Q_0^n \text{CP} \uparrow \\ Q_2^{n+1} &= D_2 \text{CP} \uparrow = Q_1^n \text{CP} \uparrow \\ Q_3^{n+1} &= D_3 \text{CP} \uparrow = Q_2^n \text{CP} \uparrow \end{aligned}$$



$D_{IR}$ : 右移串行数据输入1011

## 3.2 移位寄存器

### ❖ 4位右移移位寄存器的工作方式

#### ➤ 串入并出

串并转换（需要N个CP周期），经过4个CP，串行输入的4位数据全部移入移位寄存器中，并从 $Q_3Q_2Q_1Q_0$ 并行输出1011

#### ➤ 串入串出

把最右边的触发器的输出作为电路的输出。经过4个CP后， $Q_3$  输出的是最先串行输入的数据。

从每个触发器Q端输出的波形相同，但后级触发器Q端输出波形比前级触发器Q端输出波形滞后一个时钟周期。

把工作于串入串出方式的移位寄存器称为“延迟线”（第N级FF延迟N个CP周期）

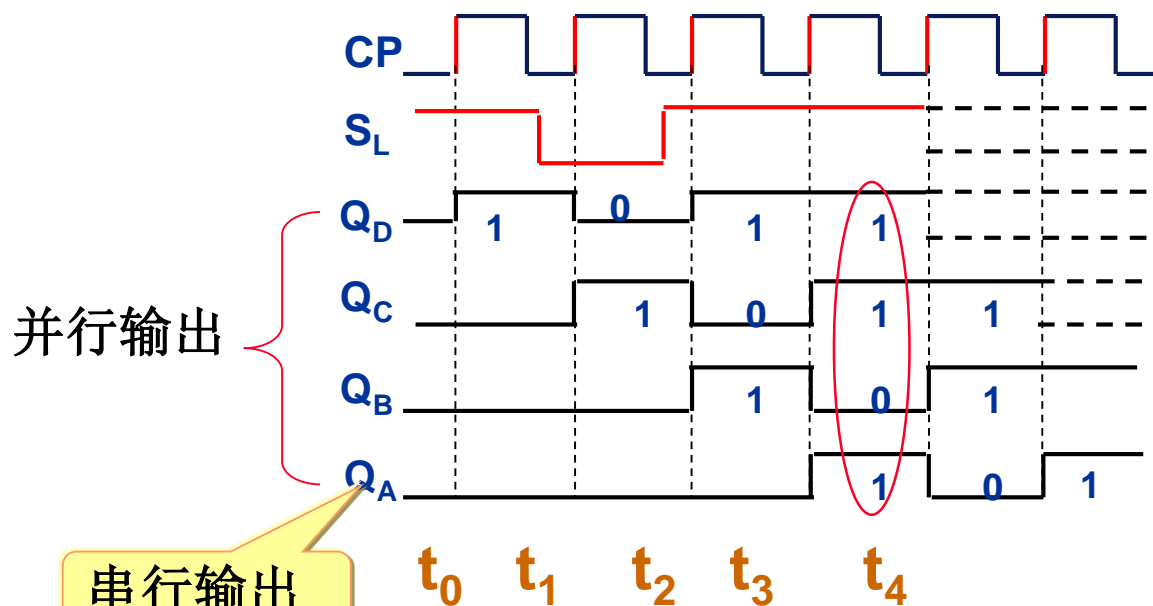




## 3.2 移位寄存器

### ❖ 4位串行输入、串/并行输出双向移位寄存器

输入数据：1011



左移时输出波形 ( $S=0$ )

## 3.2 移位寄存器

### 3. 4位双向移位寄存器（CT74194）

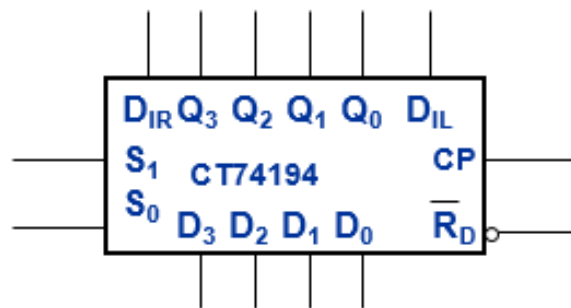
$S_1S_0$ : 功能控制输入端

$D_3D_2D_1D_0$ : 并行数据输入端

$Q_3Q_2Q_1Q_0$ : 数据输出

$D_{IR}$ : 右移串行输入（ $Q_3$ 为串行输出端）

$D_{IL}$ : 左移串行输入（ $Q_0$ 为串行输出端）



功能表

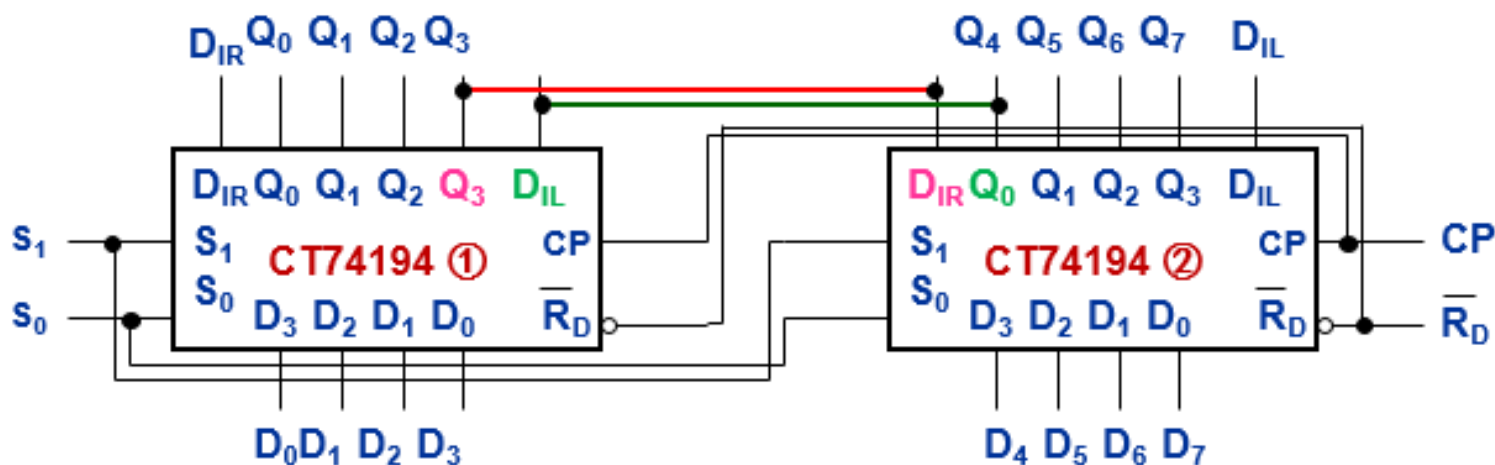
$\overline{R_D}$	CP	$S_1$	$S_0$	功能
0	X	X	X	置零
1	↑	0	0	保持
1	↑	0	1	右移
1	↑	1	0	左移
1	↑	1	1	并行输入

#### ❖ 工作方式

- 串入并出——串并转换
- 串入串出——延迟线
- 并入串出——并串转换
- 并入并出——数据预置

## 3.2 移位寄存器

### ❖ 用2片CT74194扩展为8位移位寄存器



- 把两片的  $S_1$ 、 $S_0$ 、 $CP$  和  $\overline{R_D}$  分别并联，两片同时做同样操作
- 将片①的  $Q_3$  接至片②的  $D_{IR}$ ，当  $S_1S_0=01$  时，两片同时右移，片②的输出  $Q_3$  作为整个电路的右移串行输出端 ( $Q_7$ )
- 将片②的  $Q_0$  接至片①的  $D_{IL}$ ，当  $S_1S_0=10$  时，两片同时左移，片①的输出  $Q_0$  作为整个电路的左移串行输出端  $Q_0$

## 第二部分：时序逻辑电路设计

### 一. 锁存器和触发器

1. SR/D锁存器
2. D触发器
3. JK触发器

### 二. 有限状态机

1. Moore型有限状态机
2. Mealy型有限状态机

### 三. 时序逻辑电路设计与分析

1. 数据寄存器/数据锁存器
2. 移位寄存器
3. 计数器
4. 时序电路的时序

### 3.3.1 同步计数器

❖ **计数器**：可以统计输入脉冲个数的器件

❖ **计数器的用途**

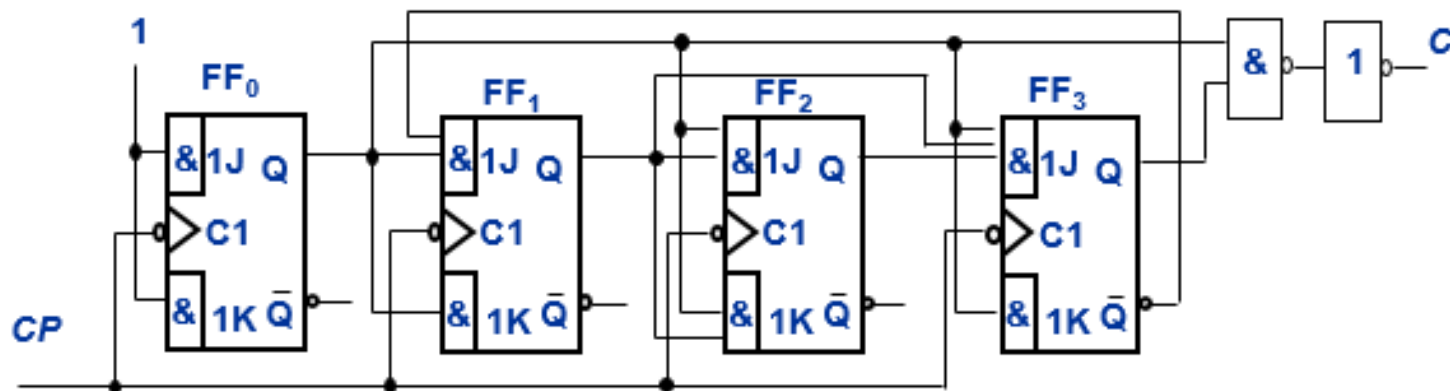
- （脉冲）计数
- 计时
- 定时（定时器）
- 分频
- 产生节拍脉冲（顺序脉冲）和序列脉冲

❖ **计数器的分类**

- 时钟方式：根据计数器中**触发器时钟端的连接方式**，分为同步计数器、异步计数器
- 计数方式：二进制计数器，十进制计数器，**M**进制计数器
- 状态变化：根据计数器中的**状态变化规律**分为加法计数器、减法计数器、加/减法计数器

### 3.3.1 同步计数器

【例1】分析下图电路，说明电路的特点。



解：(1) 写出逻辑表达式（4个负边沿触发的JK触发器组成的电路）

$$J_0 = K_0 = 1;$$

$$J_1 = \overline{Q_3}^n Q_0^n, K_1 = Q_0^n;$$

$$J_2 = K_2 = Q_1^n Q_0^n;$$

$$J_3 = \underline{\underline{Q_2^n Q_1^n Q_0^n}}, K_3 = Q_0^n$$

$$C = Q_3^n Q_0^n = Q_3^n Q_0^n$$

$$Q_0^{n+1} = J_0 \overline{Q_0}^n + \overline{K_0} Q_0^n = \overline{Q_0}^n$$

$$Q_1^{n+1} = J_1 \overline{Q_1}^n + \overline{K_1} Q_1^n = \overline{Q_3}^n Q_0^n \overline{Q_1}^n + \overline{Q_0}^n Q_1^n$$

$$Q_2^{n+1} = J_2 \overline{Q_2}^n + \overline{K_2} Q_2^n = Q_1^n Q_0^n \overline{Q_2}^n + \overline{Q_1^n Q_0^n} Q_2^n$$

$$Q_3^{n+1} = J_3 \overline{Q_3}^n + \overline{K_3} Q_3^n = Q_2^n Q_1^n Q_0^n \overline{Q_3}^n + \overline{Q_0}^n Q_3^n$$

$$CP_0 = CP_1 = CP_2 = CP_3 = CP \downarrow \text{——同步电路}$$

### 3.3.1 同步计数器

(2) 写出状态转换表

状态转换表

$Q_3^n Q_2^n Q_1^n Q_0^n$	$Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$	C
0 0 0 0	0 0 0 1	0
0 0 0 1	0 0 1 0	0
0 0 1 0	0 0 1 1	0
0 0 1 1	0 1 0 0	0
0 1 0 0	0 1 0 1	0
0 1 0 1	0 1 1 0	0
0 1 1 0	0 1 1 1	0
0 1 1 1	1 0 0 0	0
1 0 0 0	1 0 0 1	0
1 0 0 1	0 0 0 0	1
1 0 1 0	1 0 1 1	0
1 0 1 1	0 1 0 0	1
1 1 0 0	1 1 0 1	0
1 1 0 1	0 1 0 0	1
1 1 1 0	1 1 1 1	0
1 1 1 1	0 0 0 0	1

$$Q_0^{n+1} = \overline{Q_0^n}$$

$$Q_1^{n+1} = \overline{Q_3^n} Q_0^n \overline{Q_1^n} + \overline{Q_0^n} Q_1^n$$

$$Q_2^{n+1} = Q_1^n Q_0^n \overline{Q_2^n} + \overline{Q_1^n} Q_0^n Q_2^n$$

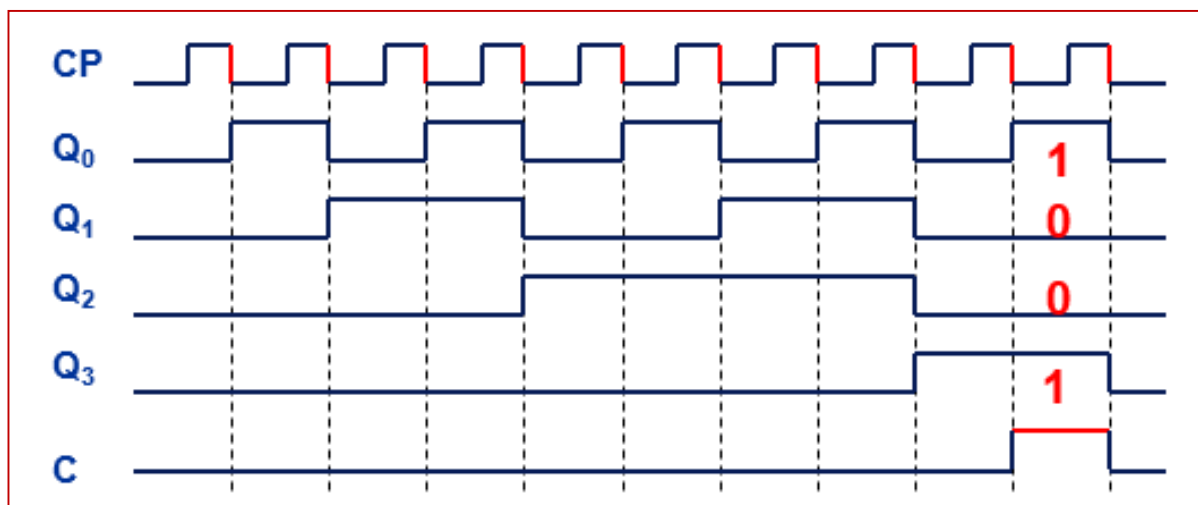
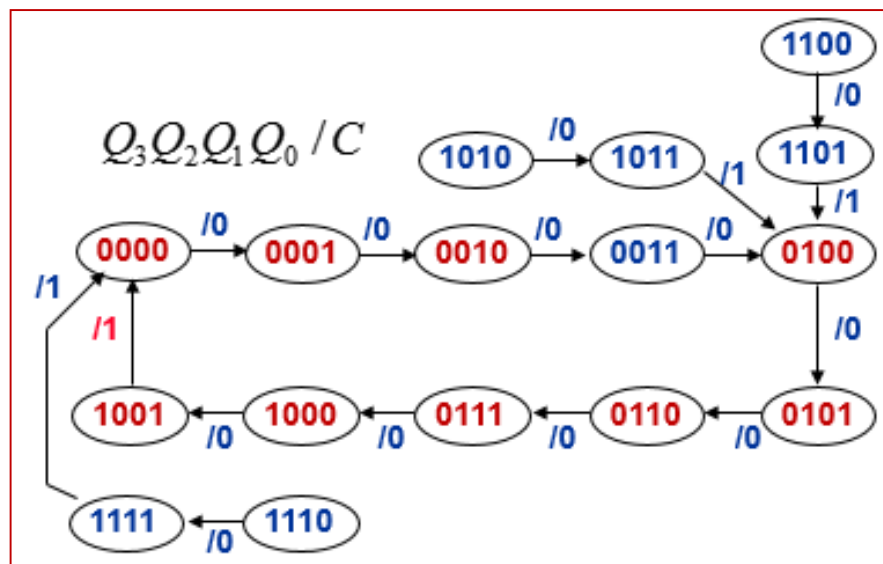
$$Q_3^{n+1} = Q_2^n Q_1^n Q_0^n \overline{Q_3^n} + \overline{Q_0^n} Q_3^n$$

$$C = Q_3^n Q_0^n$$

### 3.3.1 同步计数器

(3) 画出状态转移图（时序图）

- 画状态转换图时，一定要画出全部状态的变化。
- 画时序图时，只画出有效状态构成的计数循环的变化；注意触发器的时钟特性！



**CP下降沿时  
触发器翻转！**

当 $Q_3Q_2Q_1Q_0=1001$ 时， $C=1$ ；下一个CP下降沿到来时， $Q_3Q_2Q_1Q_0$ 变为0000，完成一个计数循环。

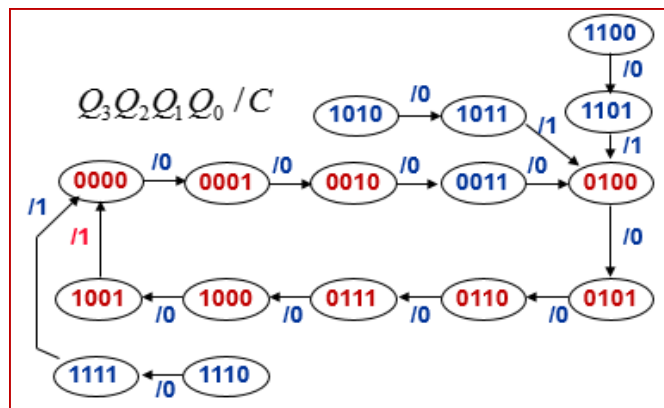
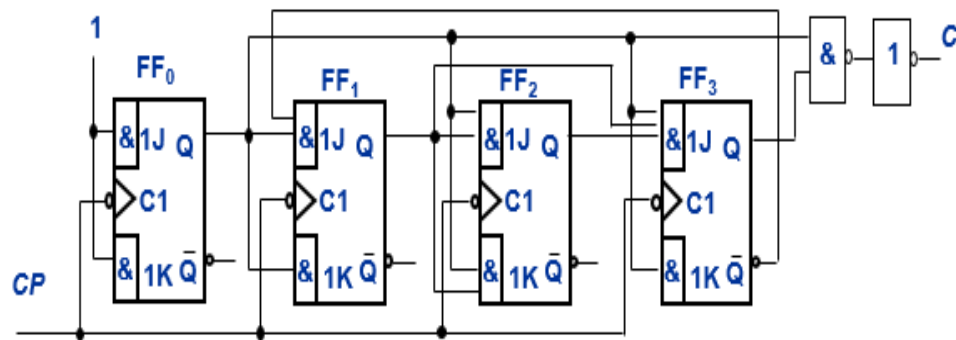


### 3.3.1 同步计数器

#### (4) 电路功能说明

由状态转换图可知，这是一个**同步十进制加法计数器**（也称**同步二进制（模10）加法计数器**），且具备自启动能力

- **计数器**：由若干状态构成一个计数循环
- **同步**：构成电路的全部FF的时钟端连接在一起
- **十进制**：计数循环的状态个数为10（模10计数器）
- **加法**：计数状态按递增方向变化
- **自启动**：不存在死循环，计数循环以外的状态，都能回到计数循环中来

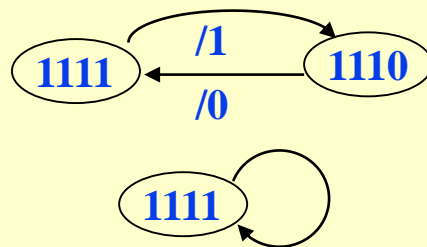


### 3.3.1 同步计数器

#### ❖ 同步计数器的特点

- 所有触发器的时钟端并联在一起，作为计数器的时钟端
- 各触发器同时翻转，不存在时钟到各触发器输出的传输延迟的积累
- 由于其工作频率只与一个触发器的时钟到输出的传输延迟有关，所以它的工作频率比异步计数器高。
- 由于计数器各触发器几乎是同时翻转的，因此，各触发器输出波形的偏移为各触发器时钟到输出的延迟之差，同步计数器输出经译码后所产生的尖峰信号宽度比较小。
- **缺点：**结构比较复杂（各触发器的输入由多个Q输出相与得到），所用元件较多。

设计计数器，不能存在  
死循环（无效循环）



### 3.3.1 同步计数器

【例2】设计一个同步十六进制数加法计数器，按照 $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow F \rightarrow 0$ 的方式循环计数，当计数到 F 后下一个计数时钟到达时产生进位输出1

解：

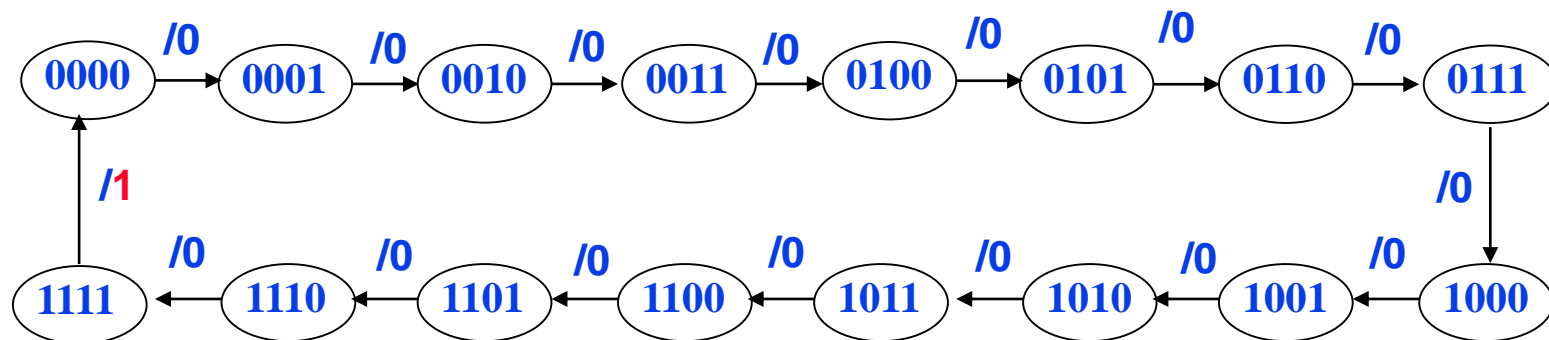
(1) 实际上是一个同步4位二进制加法计数器（模16）

采用4个负沿触发JK触发器

CP为时钟信号，CP下跳沿时计数

C为进位输出

(2) 根据题意，计数器的状态转换图如下



### 3.3.1 同步计数器

(3) 根据状态图写状态表，  
也可直接写出状态表

原态 $Q_3^n Q_2^n Q_1^n Q_0^n$	次态 $Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$	C
0 0 0 0	0 0 0 1	0
0 0 0 1	0 0 1 0	0
0 0 1 0	0 0 1 1	0
0 0 1 1	0 1 0 0	0
0 1 0 0	0 1 0 1	0
0 1 0 1	0 1 1 0	0
0 1 1 0	0 1 1 1	0
0 1 1 1	1 0 0 0	0
1 0 0 0	1 0 0 1	0
1 0 0 1	1 0 1 0	0
1 0 1 0	1 0 1 1	0
1 0 1 1	1 1 0 0	0
1 1 0 0	1 1 0 1	0
1 1 0 1	1 1 1 0	0
1 1 1 0	0 0 0 1	0
1 1 1 1	0 0 0 0	1

### 3.3.1 同步计数器

#### (4) 根据状态表写逻辑表达式 (JK触发器的特性方程)

$$\begin{aligned} Q_0^{n+1} &= \overline{Q_3^n} \overline{Q_2^n} \overline{Q_1^n} \overline{Q_0^n} + \overline{Q_3^n} \overline{Q_2^n} Q_1^n \overline{Q_0^n} + \overline{Q_3^n} Q_2^n \overline{Q_1^n} \overline{Q_0^n} + \overline{Q_3^n} Q_2^n Q_1^n \overline{Q_0^n} \\ &\quad + Q_3^n \overline{Q_2^n} \overline{Q_1^n} \overline{Q_0^n} + Q_3^n \overline{Q_2^n} Q_1^n \overline{Q_0^n} + Q_3^n Q_2^n \overline{Q_1^n} \overline{Q_0^n} + Q_3^n Q_2^n Q_1^n \overline{Q_0^n} \\ &= \overline{Q_0^n} \end{aligned}$$

$$\begin{aligned} Q_1^{n+1} &= \overline{Q_3^n} \overline{Q_2^n} \overline{Q_1^n} Q_0^n + \overline{Q_3^n} \overline{Q_2^n} Q_1^n \overline{Q_0^n} + \overline{Q_3^n} Q_2^n \overline{Q_1^n} \overline{Q_0^n} + \overline{Q_3^n} Q_2^n Q_1^n \overline{Q_0^n} \\ &\quad + Q_3^n \overline{Q_2^n} \overline{Q_1^n} Q_0^n + Q_3^n \overline{Q_2^n} Q_1^n \overline{Q_0^n} + Q_3^n Q_2^n \overline{Q_1^n} \overline{Q_0^n} + Q_3^n Q_2^n Q_1^n \overline{Q_0^n} \\ &= \overline{Q_0^n} \overline{Q_1^n} + \overline{Q_0^n} Q_1^n \end{aligned}$$

$$\begin{aligned} Q_2^{n+1} &= \overline{Q_3^n} \overline{Q_2^n} \overline{Q_1^n} Q_0^n + \overline{Q_3^n} \overline{Q_2^n} Q_1^n \overline{Q_0^n} + \overline{Q_3^n} Q_2^n \overline{Q_1^n} \overline{Q_0^n} + \overline{Q_3^n} Q_2^n Q_1^n \overline{Q_0^n} \\ &\quad + Q_3^n \overline{Q_2^n} \overline{Q_1^n} Q_0^n + Q_3^n \overline{Q_2^n} Q_1^n \overline{Q_0^n} + Q_3^n Q_2^n \overline{Q_1^n} \overline{Q_0^n} + Q_3^n Q_2^n Q_1^n \overline{Q_0^n} \\ &= \overline{Q_0^n} \overline{Q_1^n} \overline{Q_2^n} + \overline{Q_0^n} Q_1^n \overline{Q_2^n} \end{aligned}$$

$$\begin{aligned} Q_3^{n+1} &= \overline{Q_3^n} \overline{Q_2^n} \overline{Q_1^n} Q_0^n + \overline{Q_3^n} \overline{Q_2^n} Q_1^n \overline{Q_0^n} + \overline{Q_3^n} Q_2^n \overline{Q_1^n} \overline{Q_0^n} + \overline{Q_3^n} Q_2^n Q_1^n \overline{Q_0^n} \\ &\quad + Q_3^n \overline{Q_2^n} \overline{Q_1^n} Q_0^n + Q_3^n \overline{Q_2^n} Q_1^n \overline{Q_0^n} + Q_3^n Q_2^n \overline{Q_1^n} \overline{Q_0^n} + Q_3^n Q_2^n Q_1^n \overline{Q_0^n} \\ &= \overline{Q_0^n} \overline{Q_1^n} \overline{Q_2^n} \overline{Q_3^n} + \overline{Q_0^n} \overline{Q_1^n} Q_2^n \overline{Q_3^n} \end{aligned}$$

$$C = Q_0^n Q_1^n Q_2^n Q_3^n$$

### 3.3.1 同步计数器

(5) 根据特性方程反推出4个JK触发器的J、K端驱动方程

因为：JK触发器的通用特性方程是：

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

所以：4个触发器的驱动方程分别是：

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0^n$$

$$J_2 = K_2 = Q_0^n Q_1^n$$

$$J_3 = K_3 = Q_0^n Q_1^n Q_2^n$$

$$Q_0^{n+1} = \overline{Q_0^n}$$

$$Q_1^{n+1} = Q_0^n \overline{Q_1^n} + \overline{Q_0^n} Q_1^n$$

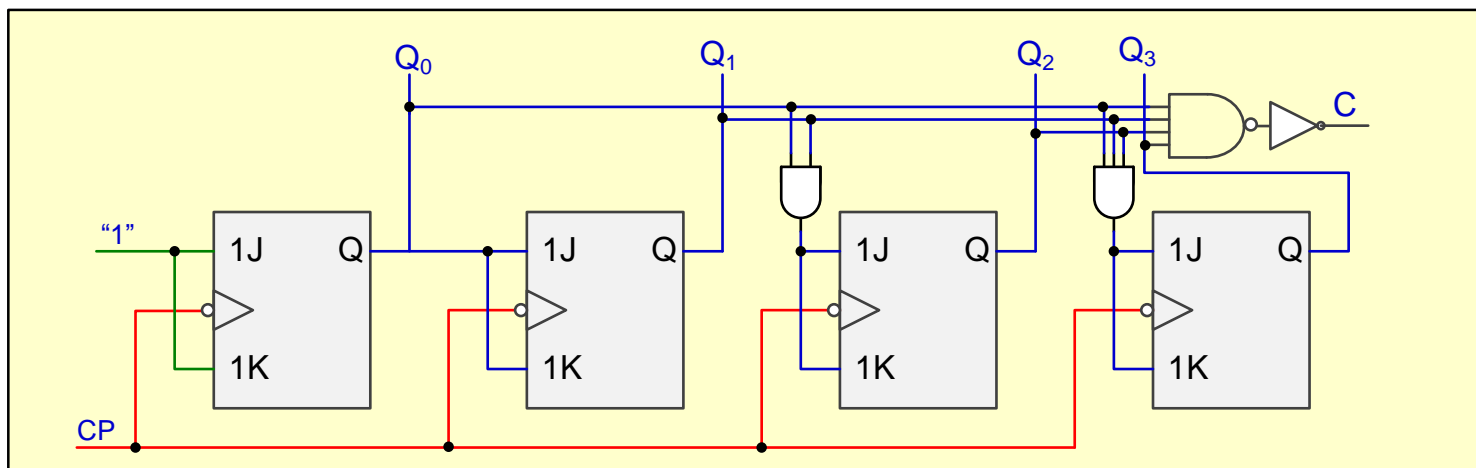
$$Q_2^{n+1} = Q_0^n Q_1^n \overline{Q_2^n} + \overline{Q_0^n} \overline{Q_1^n} Q_2^n$$

$$Q_3^{n+1} = Q_0^n Q_1^n Q_2^n \overline{Q_3^n} + \overline{Q_0^n} \overline{Q_1^n} \overline{Q_2^n} Q_3^n$$

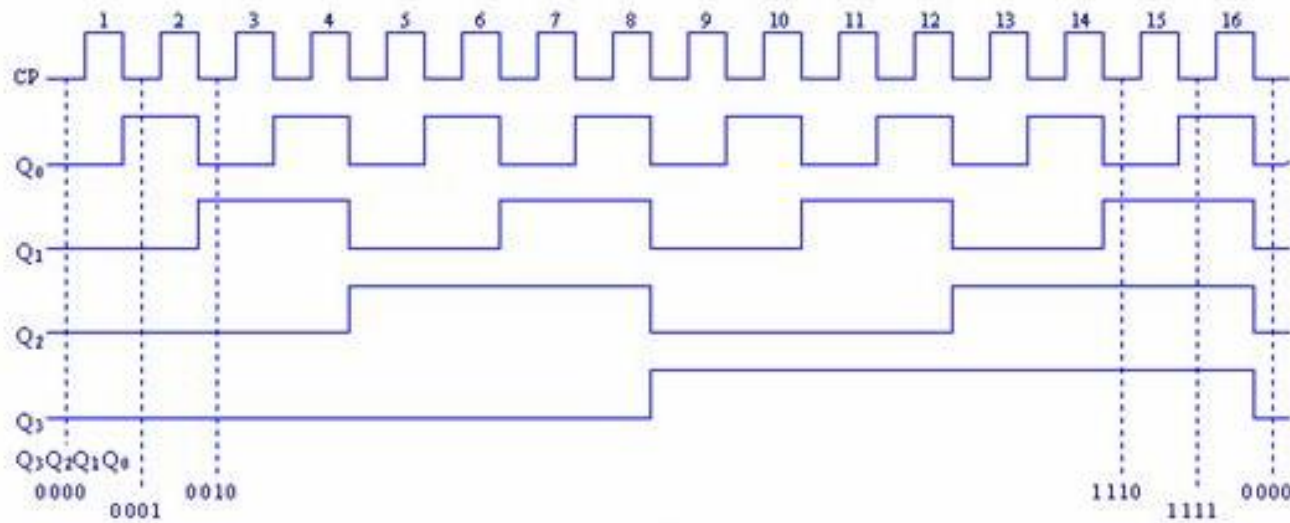
$$C = Q_0^n Q_1^n Q_2^n Q_3^n$$

### 3.3.1 同步计数器

#### (6) 画出逻辑图



$Q_0$ 、 $Q_1$ 、 $Q_2$ 、 $Q_3$  的周期分别是CP周期的2、4、8、16倍，实际上是分别对CP进行了2、4、8、16分频，因此二进制计数器也称为分频器。



### 3.3.2 异步计数器

❖ 异步计数器也有二进制、十进制、任意进制等类型

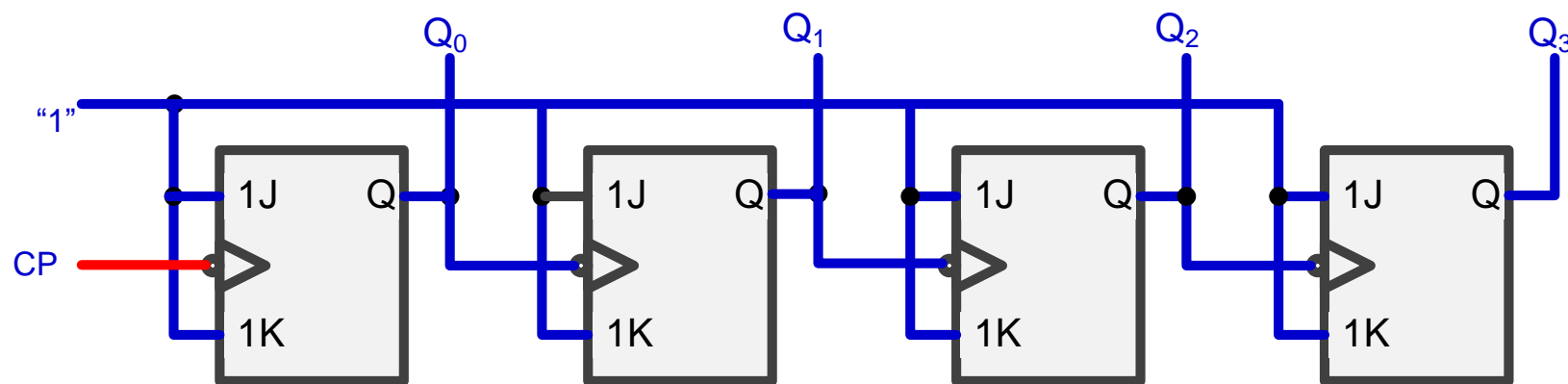
❖ 异步计数器的特点

- 输入系统**时钟脉冲只作用于最低位**触发器，高位触发器的时钟信号往往是由低一位触发器的输出提供的，高位触发器的翻转有待低一位触发器翻转后才能进行。
- 由于每一级触发器都存在传输延迟，因此计数器**工作速度慢**，而且，位数越多计数越慢。在大型数字设备中较少采用。
- 对计数器状态进行译码时，由于触发器不同步，译码器输出**会出现尖峰脉冲**（位数越多，尖峰信号也就越宽），使仪器设备产生误动作。
- **优点**：结构比较简单，所用元件较少。



### 3.3.2 异步计数器

【例3】分析下图异步二进制（M=16）加法计数器电路（N=4）



(1) 状态方程： 利用特性方程  $Q^{n+1} = J\overline{Q}^n + \overline{K}Q^n$ ，加时钟信号

$$\begin{aligned} Q_0^{n+1} &= \overline{Q_0}^n \cdot CP \downarrow; Q_1^{n+1} = \overline{Q_1}^n \cdot Q_0 \downarrow; \\ Q_2^{n+1} &= \overline{Q_2}^n \cdot Q_1 \downarrow; Q_3^{n+1} = \overline{Q_3}^n \cdot Q_2 \downarrow; \end{aligned}$$

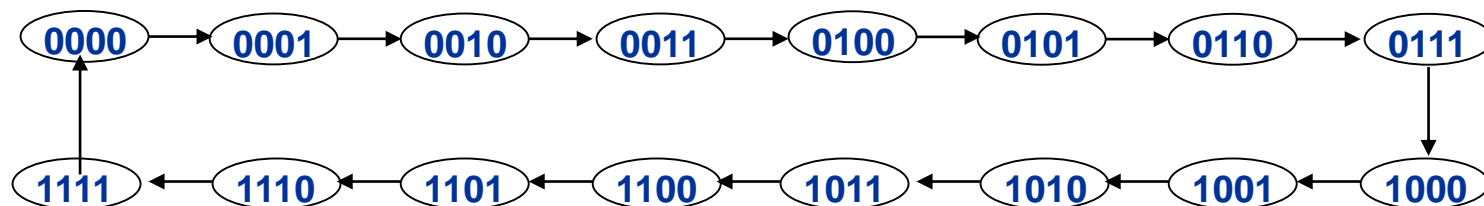
## 3.3.2 异步计数器

(2) 状态转换表

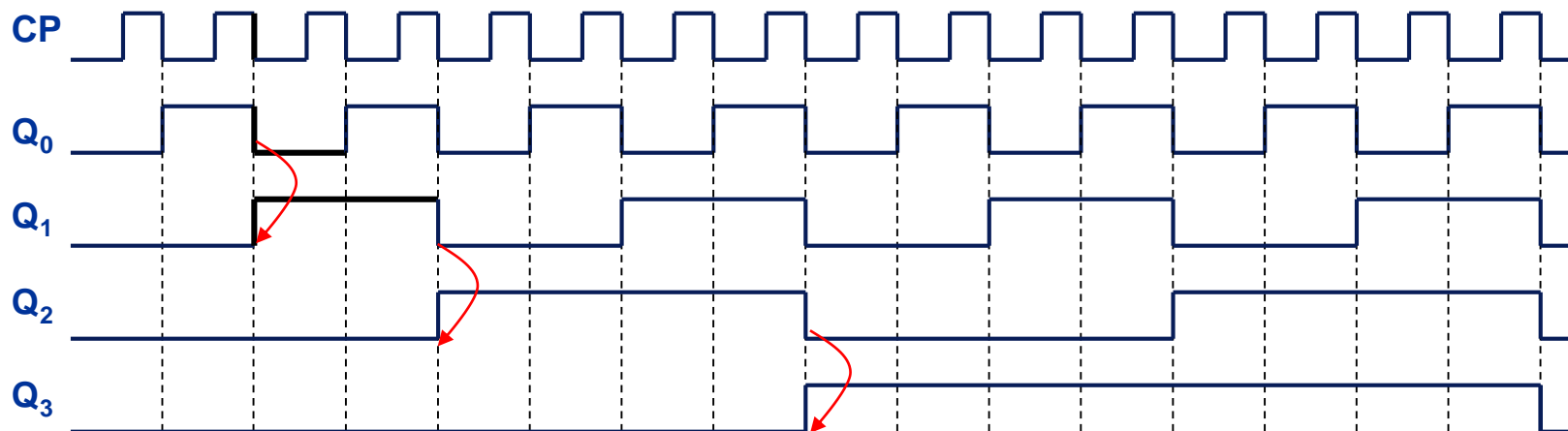
$Q_3^n$ $Q_2^n$ $Q_1^n$ $Q_0^n$	$Q_3^{n+1}$ $Q_2^{n+1}$ $Q_1^{n+1}$ $Q_0^{n+1}$
0 0 0 0	0 0 0 1
0 0 0 1	0 0 1 0
0 0 1 0	0 0 1 1
0 0 1 1	0 1 0 0
0 1 0 0	0 1 0 1
0 1 0 1	0 1 1 0
0 1 1 0	0 1 1 1
0 1 1 1	1 0 0 0
1 0 0 0	1 0 0 1
1 0 0 1	1 0 1 0
1 0 1 0	1 0 1 1
1 0 1 1	1 1 0 0
1 1 0 0	1 1 0 1
1 1 0 1	1 1 1 0
1 1 1 0	1 1 1 1
1 1 1 1	0 0 0 0

## 3.3.2 异步计数器

### (3) 状态转换图

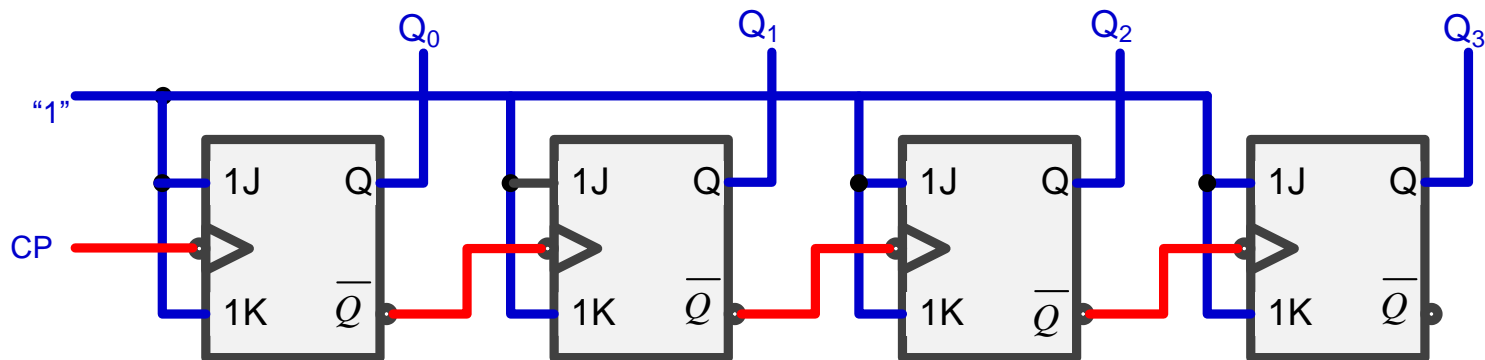


### 时序图



### 3.3.2 异步计数器 —— 其他类型计数器

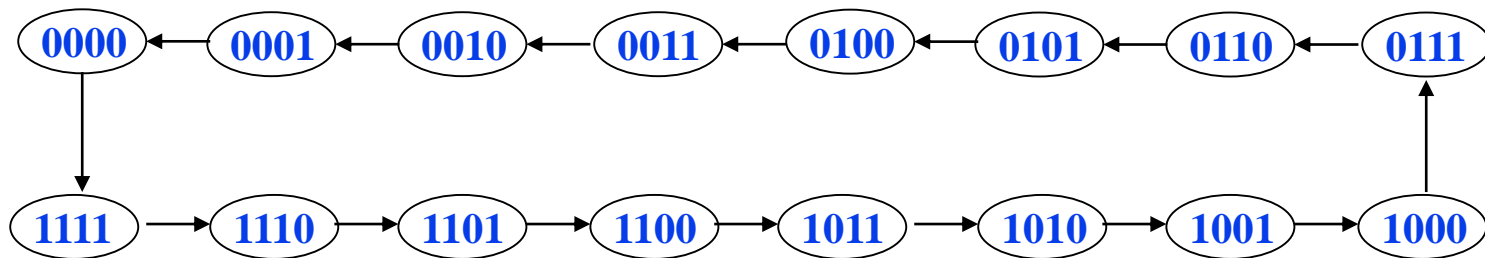
#### 1. 异步二进制（模16）减法计数器



#### ➤ 状态方程

$$\begin{aligned} Q_0^{n+1} &= \overline{Q_0^n} \cdot CP \downarrow; Q_1^{n+1} = \overline{Q_1^n} \cdot \overline{Q_0^n} \downarrow = \overline{Q_1^n} \cdot Q_0^n \uparrow; \\ Q_2^{n+1} &= \overline{Q_2^n} \cdot \overline{Q_1^n} \downarrow = \overline{Q_2^n} \cdot Q_1^n \uparrow; Q_3^{n+1} = \overline{Q_3^n} \cdot \overline{Q_2^n} \downarrow = \overline{Q_3^n} \cdot Q_2^n \uparrow; \end{aligned}$$

#### ➤ 状态转换图





### 3.3.2 异步计数器 —— 其他类型计数器

#### 2. D触发器（上跳沿触发）构成异步二进制（模8）加法计数器（续）

##### （4）根据驱动方程画出逻辑电路图

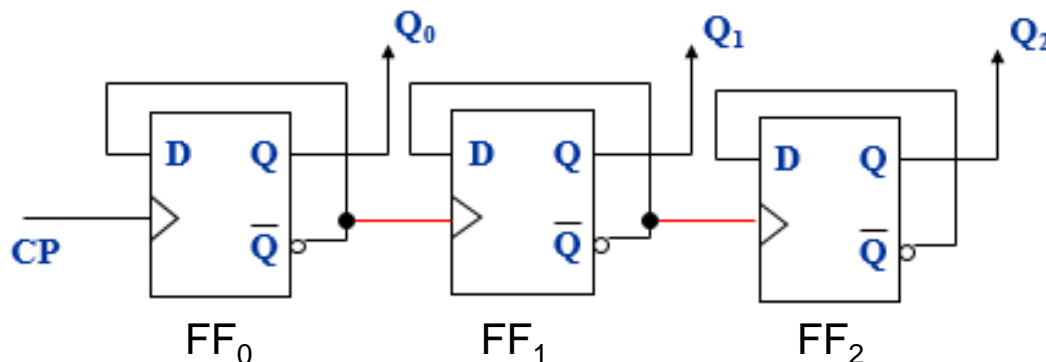
由3个D-FF（D触发器）构成，异步计数器FF<sub>0</sub>用CP时钟触发，FF<sub>1</sub>用FF<sub>0</sub>输出触发，FF<sub>2</sub>用FF<sub>1</sub>输出触发。

假定D触发器CP上升沿触发，而在驱动方程中FF<sub>1</sub>、FF<sub>2</sub>分别是在Q<sub>0</sub>的下降沿和Q<sub>1</sub>的下降沿翻转，也即在/Q<sub>0</sub>的上升沿和/Q<sub>1</sub>的上升沿翻转。故分别将/Q<sub>0</sub>和/Q<sub>1</sub>作为FF<sub>1</sub>、FF<sub>2</sub>的时钟信号，由此画出电路图。

$$D_0 = \overline{Q_0}^n \cdot CP \uparrow$$

$$D_1 = \overline{Q_1}^n \cdot Q_0 \downarrow$$

$$D_2 = \overline{Q_2}^n \cdot Q_1 \downarrow$$



## 异步计数器小结

- ❖  $n$ 位二进制异步计数器由 $n$ 个触发器组成。
  - 用D 触发器构造时，使 $D_i = \neg Q_i$ ；
  - 用JK 触发器构造时，使 $J_i = K_i = 1$ ；
  - 构造加法计数器，若上升沿触发，则应将低位触发器的 $\neg Q$ 端与相邻高位触发器的时钟输入端相连；若下降沿触发，则应将低位触发器的 $Q$ 端与相邻高位触发器的时钟输入端连接；
  - 构造减法计数器，各触发器的连接方式则加法计数器相反。若上升沿触发，则应将低位触发器的 $Q$ 端与相邻高位触发器的时钟输入端相连；若下降沿触发，则应将低位触发器的 $\neg Q$ 非端与相邻高位触发器的时钟输入端连接；
- ❖ 二进制异步计数器中，高位触发器的状态翻转必须在低一位触发器产生进位信号（加计数）或借位信号（减计数）之后才能实现。故又称这种类型的计数器为串行计数器。