

信息与电子工程导论

Introduction to Information Science and Electronic Engineering

4.2 组合逻辑和时序逻辑

章献民 主编

浙江大学出版社

2023年9月

知识图谱

- 2.1 时域和频域
- 2.2 模拟和数字
- 2.3 编码和调制
- 2.4 电磁场与波

2 信号与数据

场与波

信号

数据

信息

1 信息与信息技术概述

- 1.1 信息
- 1.2 信息科学技术概述
- 1.3 知识图谱

3 电子器件与电路

- 3.1 电路模型和基本定律
- 3.2 晶体管和集成电路
- 3.3 集成运算放大器

器件

电路

处理器

计算机

网络

4 逻辑与数字系统

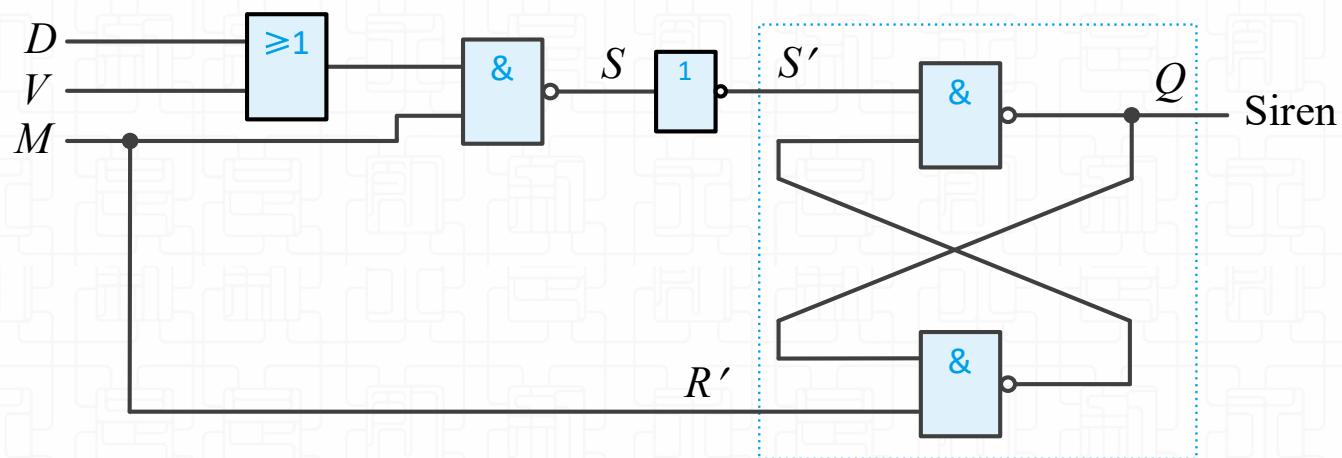
- 4.1 数字逻辑和电路
- 4.2 组合逻辑和时序逻辑
- 4.3 微处理器和计算机系统
- 4.4 嵌入式系统
- 4.5 EDA技术

5 互联与计算

- 5.1 通信与网络
- 5.2 物联与数联
- 5.3 计算与智能

内容提要

- ❖ 组合逻辑电路
- ❖ 时序逻辑电路
- ❖ 有限状态机



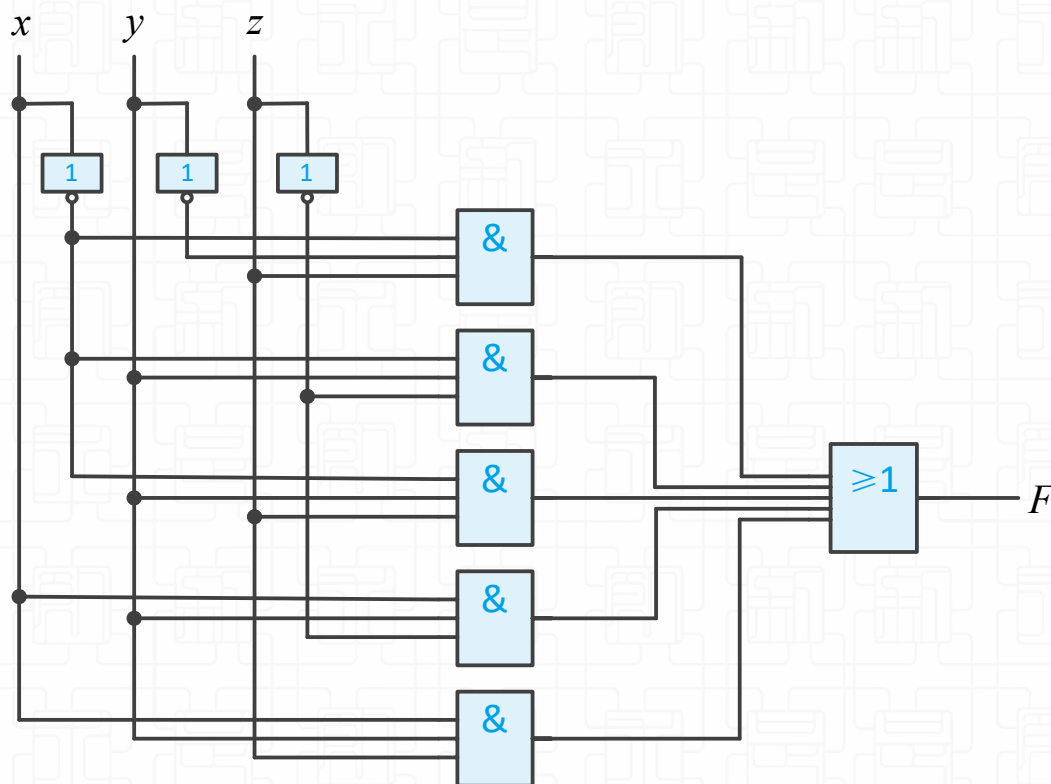
从晶体管到处理器

- ❖ 晶体管是数字系统最基本的组成单元
- ❖ 晶体管的通断可以表达数字逻辑的0或者1
- ❖ 数字逻辑门可以由晶体管连接而成
- ❖ **组合逻辑**：由多个逻辑门连接而成
 - 输出只和当前的输入有关，即没有记忆功能。
- ❖ **时序逻辑**：由多个逻辑门连接而成
 - 与组合逻辑的连接方式不同，输出不仅和当前的输入有关，还和以前的输入有关，即具有记忆功能。
- ❖ **微处理器**：由组合逻辑电路和时序逻辑电路组成
- ❖ 无论是组合逻辑电路，还是时序逻辑电路，都可以由最基本的与门，或门和非门组成。

组合逻辑电路的“综合”和“分析”

- ❖ 组合逻辑电路的“综合”和“分析”是互逆的过程。
- ❖ 在组合电路的“综合”中，首先对电路的功能进行描述，然后得到精确描述电路的真值表或布尔方程，并进而得到电路图。
- ❖ 组合逻辑电路的“分析”是“综合”的反向过程，其步骤是：写出逻辑表达式，画出真值表，进而分析其功能。

组合逻辑电路的功能描述方法：真值表



x	y	z	F
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	1	1

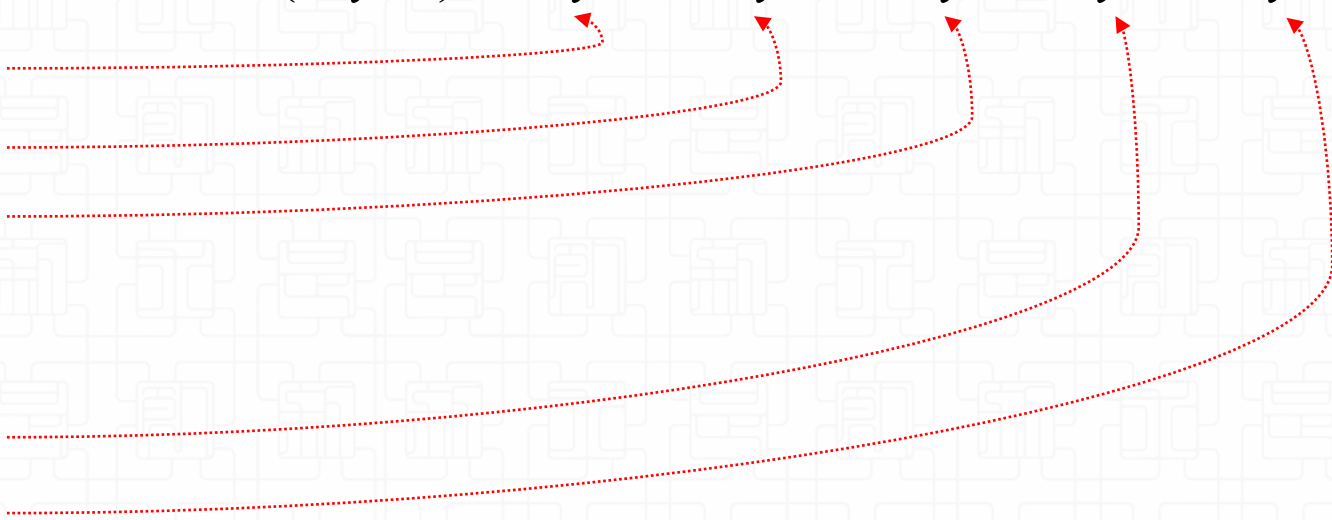
- ❖ 真值表列出输入所有的组合情况下对应的输出值，可用于描述组合逻辑电路的功能。
- ❖ 真值表的行数等于2的 n 次方，其中 n 为输入变量数目。
- ❖ 真值表描述和组合电路功能描述是等效的，是电路功能描述的严格形式。

从真值表可以得到逻辑表达式

真值表

x	y	z	F
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	1	1

布尔方程、逻辑表达式

$$F(x, y, z) = x'y'z + x'yz' + x'yz + xyz' + xyz$$


- ❖ 根据真值表，可写出逻辑表达式的“与-或 (Sum of Product, SOP)”形式。
- ❖ 对于真值表每一行，如果输出为1，则在逻辑表达式中出现为一“与”项，否则不出现。
- ❖ 对每一个“与”项，根据每个输入变量为0或者为1，对应输入变量取“非”或不取“非”。

数字电路

❖ 逻辑函数的三种表示方法可以相互转换。

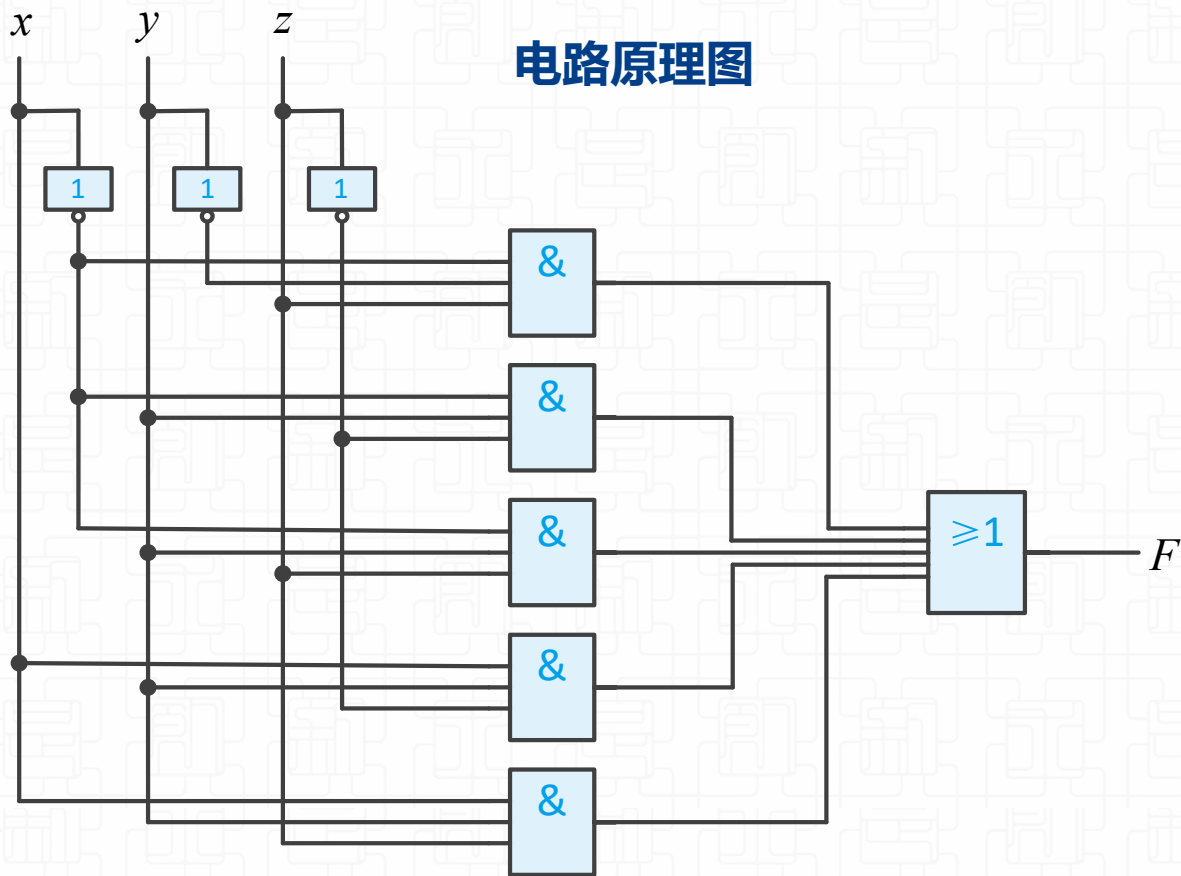
真值表

x	y	z	F
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	1	1

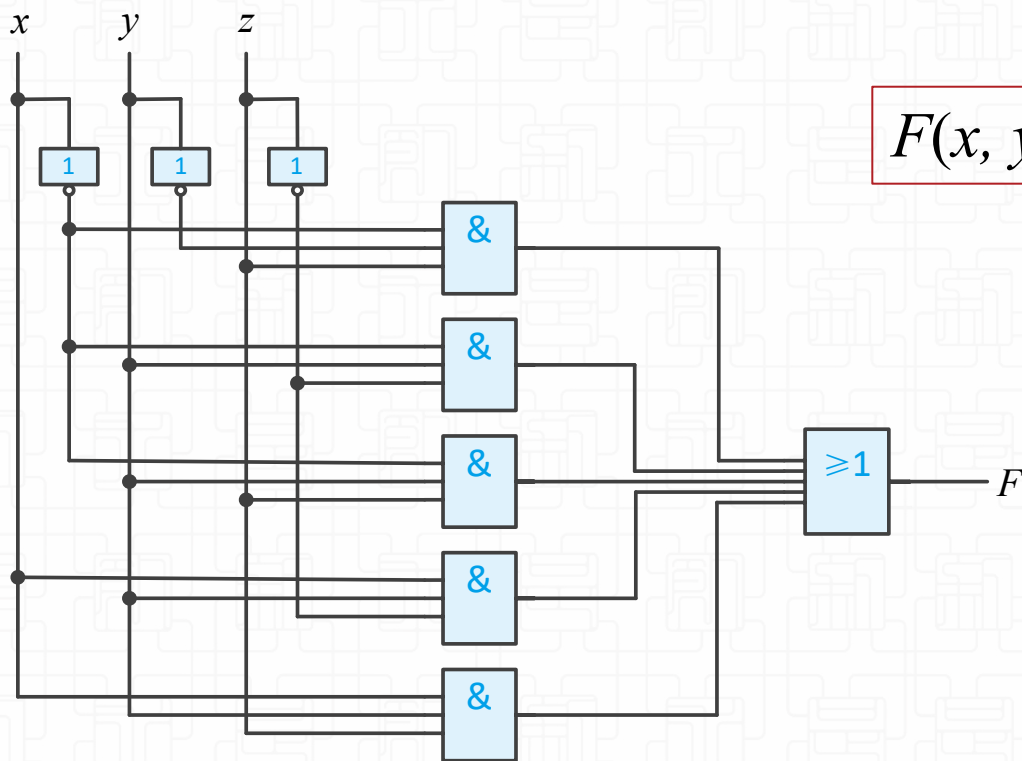
布尔方程、逻辑表达式

$$F(x, y, z) = x'y'z + x'yz' + x'yz + xyz' + xyz$$

电路原理图



从逻辑表达式即可综合 (Synthesize) 出电路



$$F(x, y, z) = x'y'z + x'yz' + x'yz + xyz' + xyz$$

“综合”是指由电路功能得到电路（与或非门）连接图；
“分析”是指给出与或非门的连接图，分析出电路的功能。

- ❖ 任意的组合逻辑，都可以用 SOP 形式表示。SOP 的表达式，只需要用与门，或门和非门即可综合出来。
- ❖ 由上图所示，最后一级表示 5 个“与”项的或，用一个“或”门；中间一级是表示每一个“与”项，一共有 5 个与项，所以用 5 个与门，对于每一个与门，其输入根据是否取非，来确定是否需要加入一个非门。

例：汽车报警器

❖ 输入：三个变量 M, D, V

- M: 报警器开关
- D: 门上的传感器检测到门被打开了
- V: 震动传感器检测到车辆的震动

❖ 输出：报警信号S

❖ 罗列出输出和输入的关系

- 报警器开关关闭，报警信号S一直为0
- 报警器开关打开，当检测到震动，或者检测到门打开，或者同时检测到震动和门打开，报警信号S为1

M	D	V	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

从真值表可以得到逻辑表达式

主开关	车门	震动	警报信号
M	D	V	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$S = MD'V + MDV' + MDV$$


- ❖ 对于真值表中输出为1的行，在逻辑表达式中出现为一“与”项。
- ❖ 对每一个“与”项，根据每个输入变量为0或者为1，对应输入变量取“非”或不取“非”。

从逻辑表达式即可综合出逻辑电路图

- ❖ 由逻辑符号及相应连线构成的电路图。
- ❖ 根据逻辑式画逻辑图的方法：将各级逻辑运算用相应逻辑门去实现。

$$S = MD'V + MDV' + MDV$$

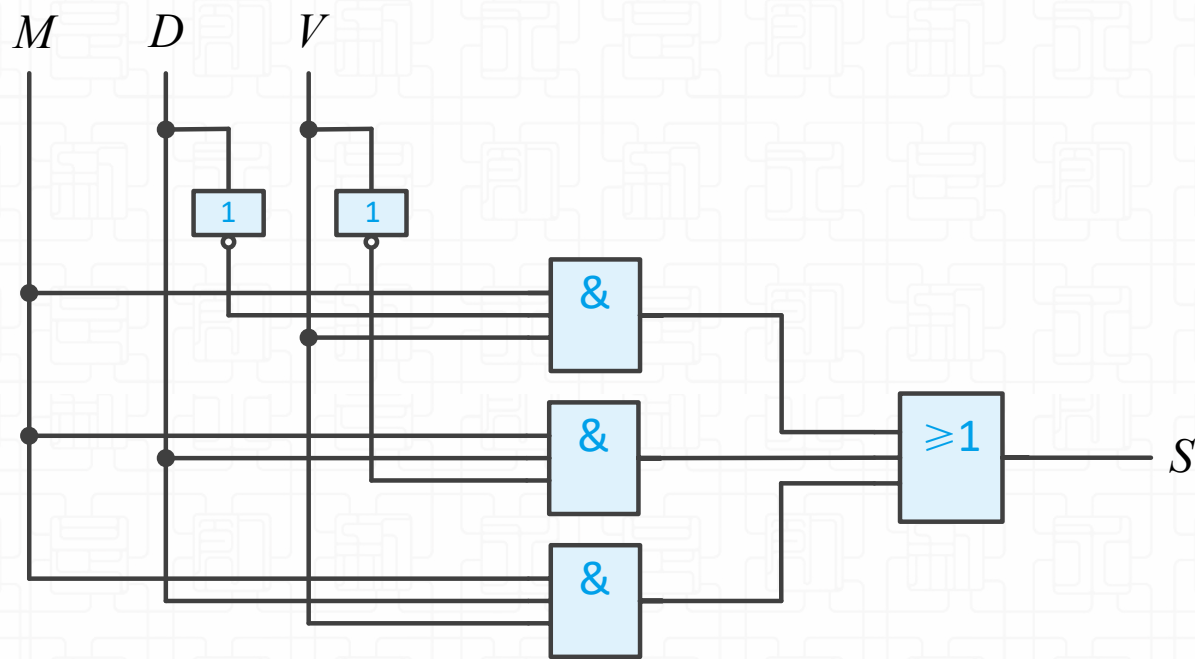
反变量用非门实现

相加项用或门实现

与项用与门实现

- ❖ 运算次序为**先非 后与 再或**。

- ❖ 因此用三级电路实现。



电路简化

$$\begin{aligned} S &= MD'V + MDV' + MDV \\ &= MD'V + MD(V' + V) \end{aligned}$$

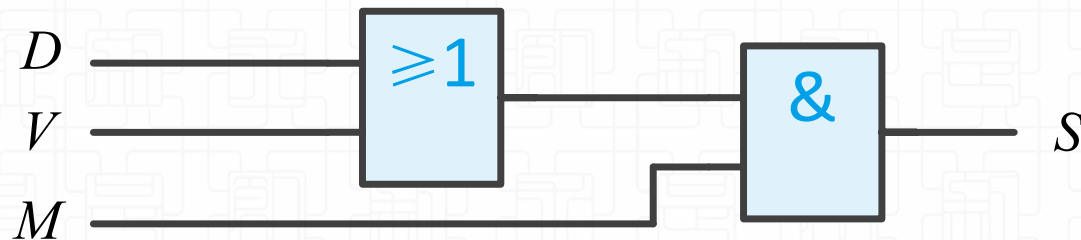
应用 $x' + x = 1$

$$\begin{aligned} S &= MD'V + MD \\ &= M(D'V + D) \end{aligned}$$

应用 $x + yz = (x + y)(x + z)$

$$\begin{aligned} S &= M(D + D')(D + V) \\ &= M(D + V) \end{aligned}$$

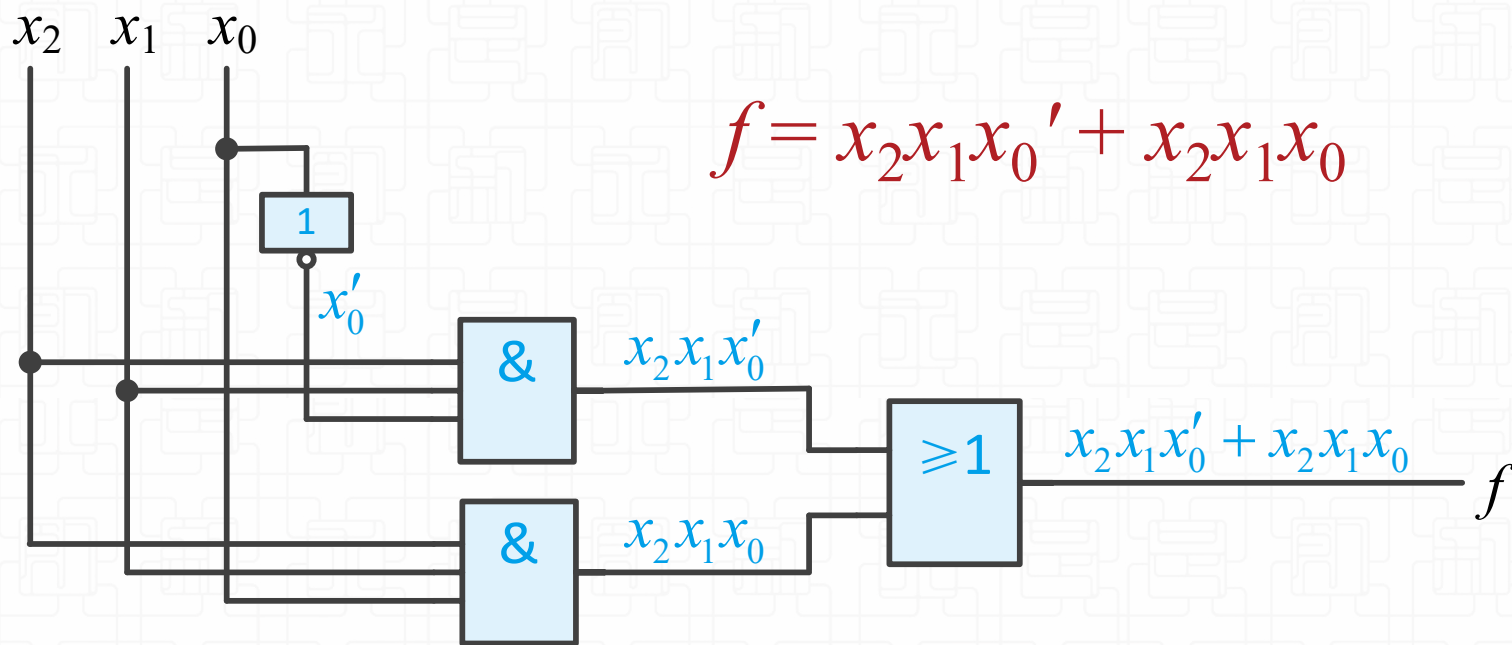
$$S = M(D + V)$$



例：三比特无符号比较器

❖ 功能要求：判断一个三比特的二进制数是否大于等于6，
若是则输出1，不是则输出0

❖ 第一步：真值表，第二步：表达式；第三步，电路图

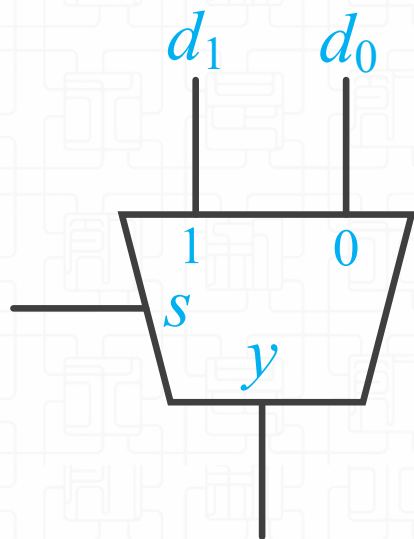


十进制数	二进制数			输出 f
	x_2	x_1	x_0	
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

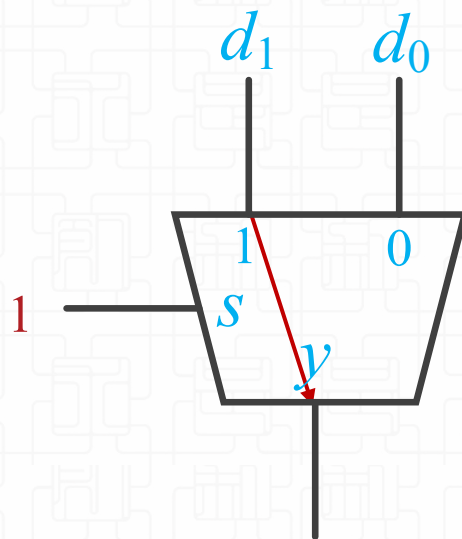
例3：二选一多路选择器

❖ 功能要求：

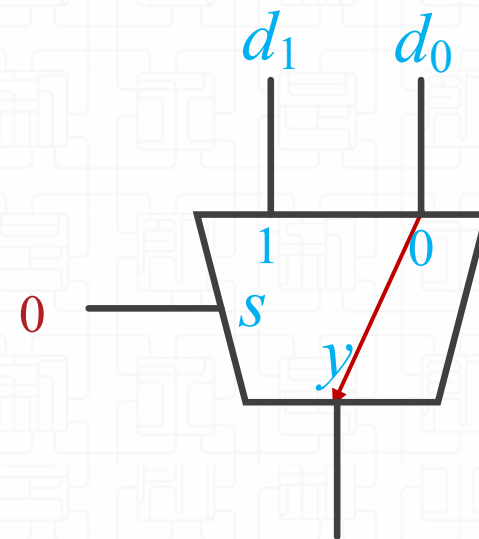
- 设计一个选择器，其中2个数据输入为 d_0 和 d_1 ，一个选择输入为 s ，当 $s=0$ 是输出等于 d_0 ，当 $s=1$ 时，输出等于 d_1



2-1选择器



$s=1, y=d_1$

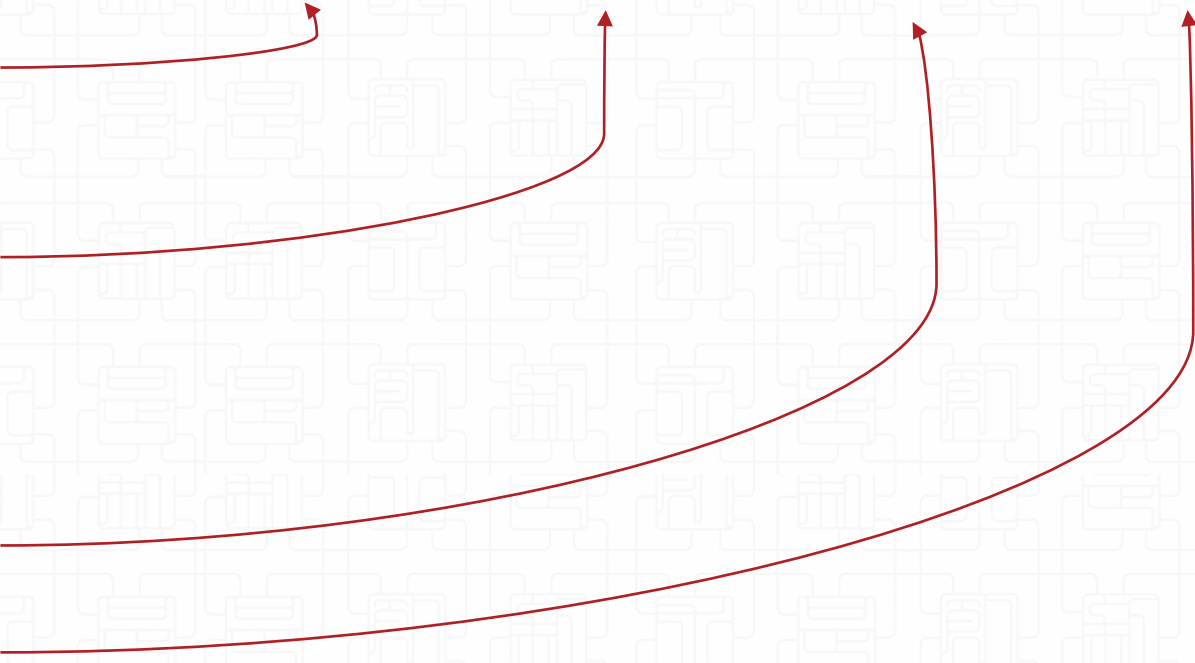


$s=0, y=d_0$

选择器设计

❖ 第一步：真值表，第二步：表达式；第三步，电路图

s	d ₁	d ₀	y
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

$$y = s'd_1'd_0 + s'd_1d_0 + sd_1d'_0 + sd_1d_0$$


选择器设计

s	d ₁	d ₀	y
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

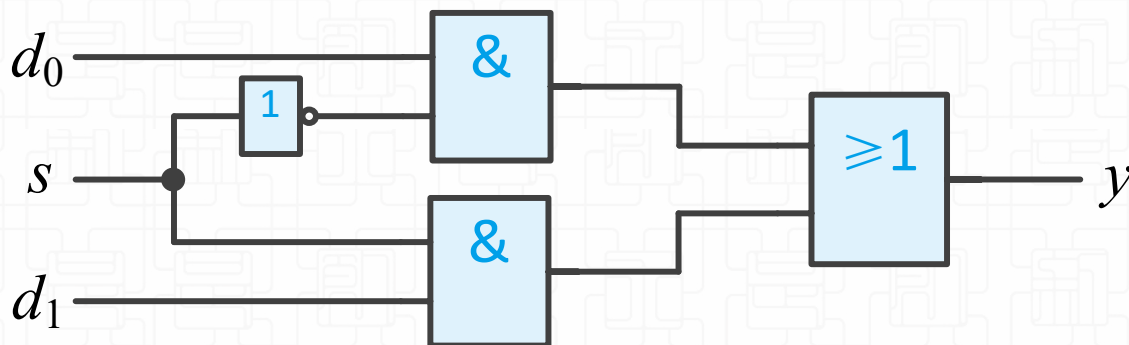
$$y = s'd_1'd_0 + s'd_1d_0 + sd_1d'_0 + sd_1d_0$$

$$\begin{aligned} & s'd_1'd_0 + s'd_1d_0 \\ &= s'd_0(d'_1 + d_1) \\ &= s'd_0 \end{aligned}$$

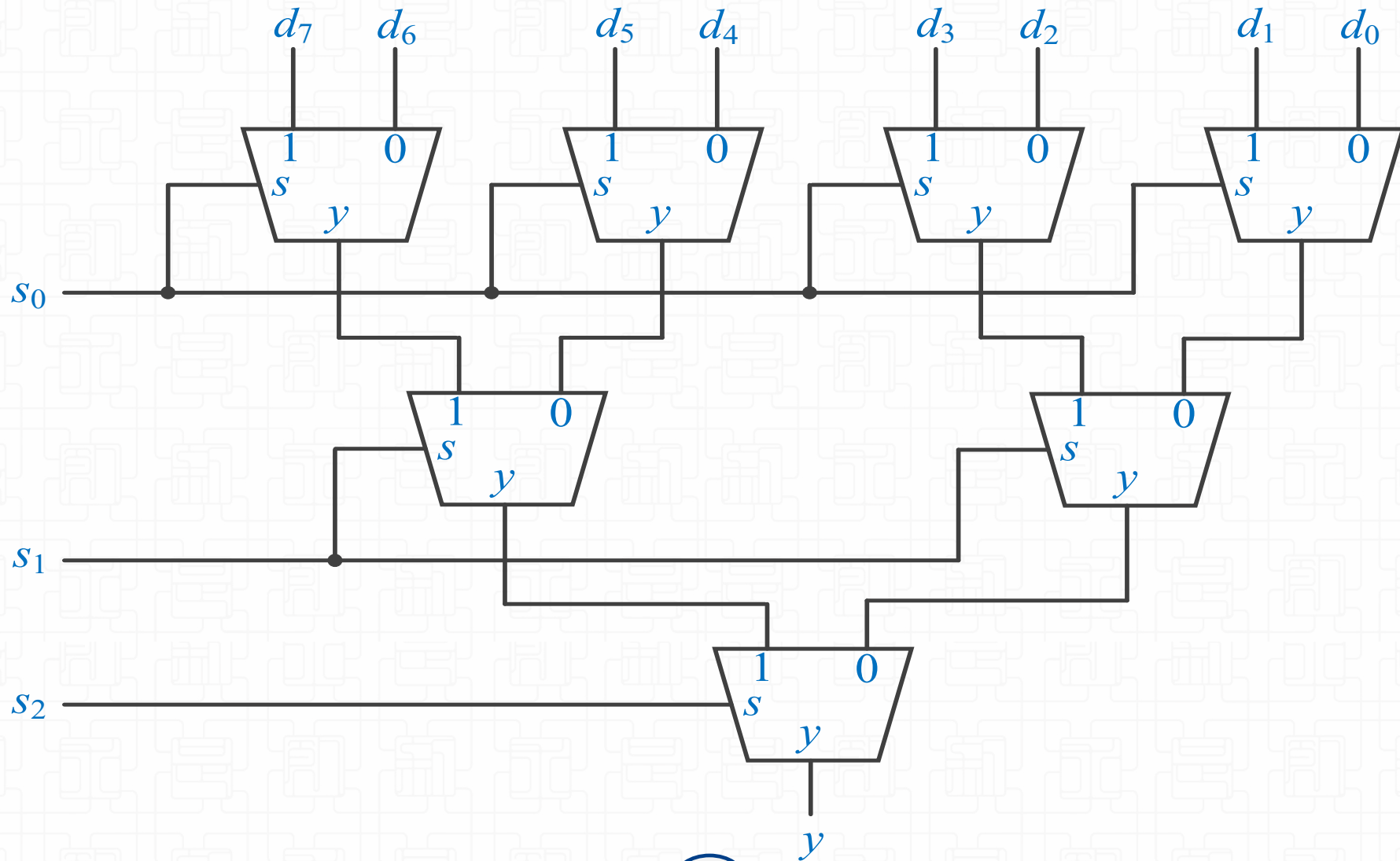
逻辑化简

$$\begin{aligned} & sd_1d'_0 + sd_1d_0 \\ &= sd_1(d'_0 + d_0) \\ &= sd_1 \end{aligned}$$

$$y = s'd_0 + sd_1$$

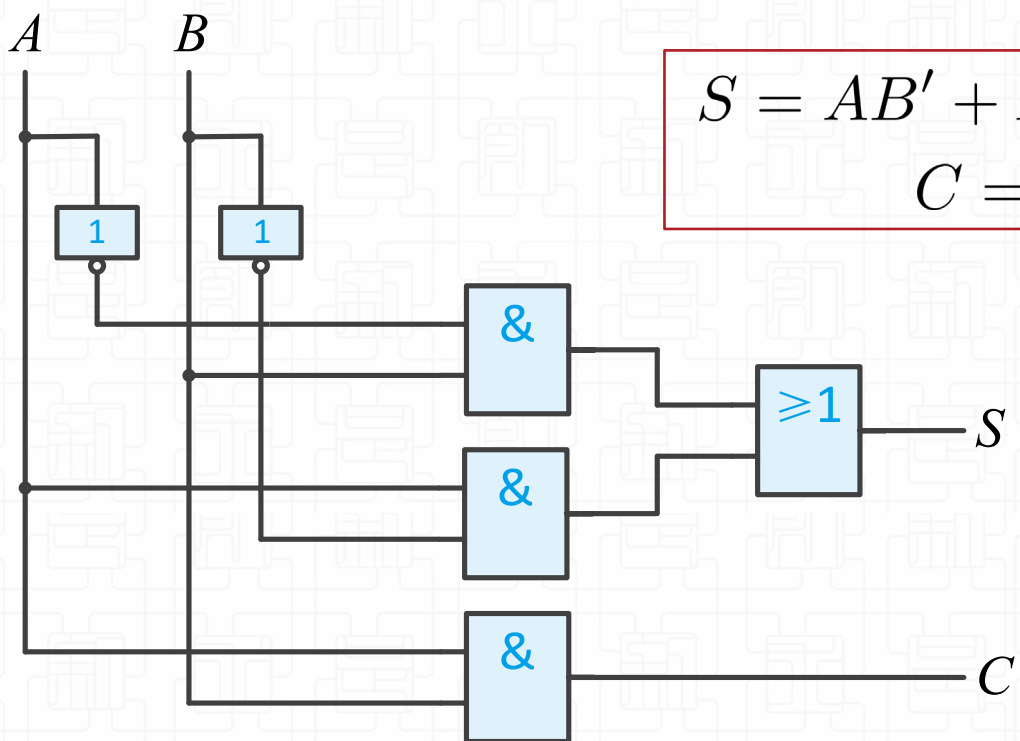


使用多个2-1选择器扩展8-1选择器



半加器

- ❖ 加法器是完成 1 bit 二进制相加的一种组合逻辑电路，是算术运算电路中的基本单元。
- ❖ 半加器的功能是将两个 1 bit 二进制数相加，输出“和 S”和“进位 C”。



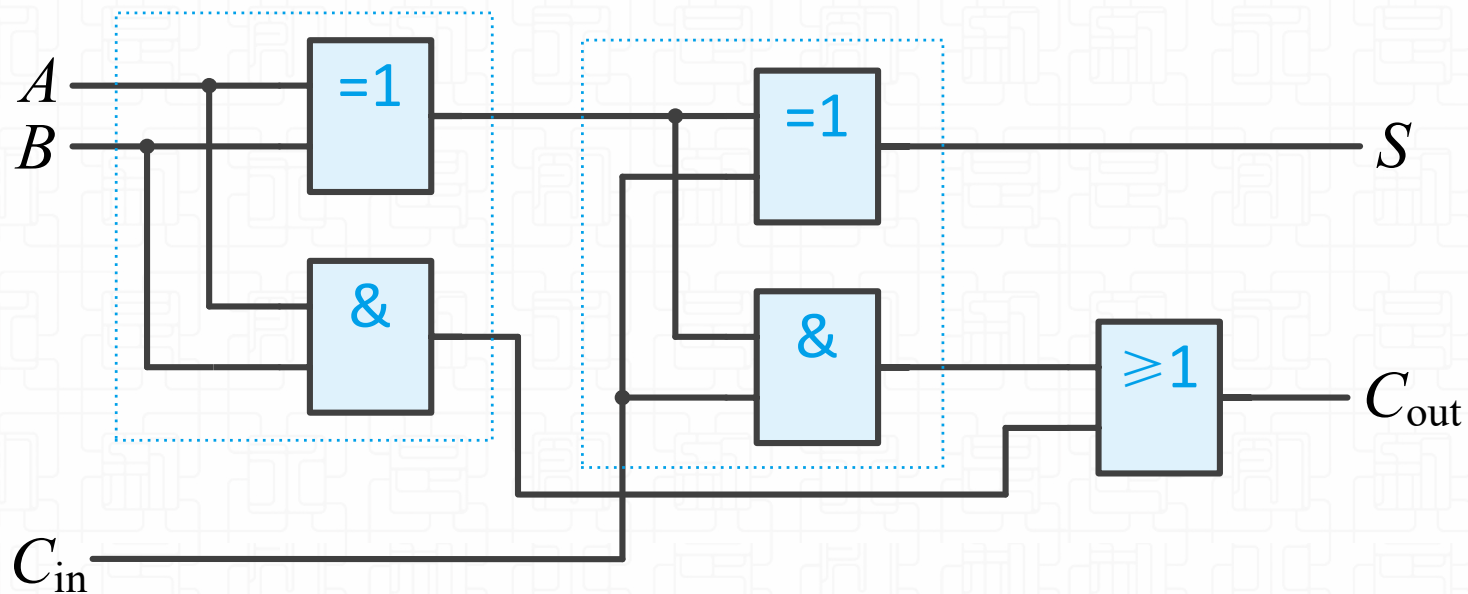
$$S = AB' + A'B = A \oplus B$$

$$C = AB$$

输入		输出	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

全加器

❖ 与半加器不同之处在于它还能接收一个低位进位输入信号 C_{in} 。



输入			输出	
C_{in}	A	B	C_{out}	S
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

汽车报警器

❖ 输入：三个变量D, V, M

- M: 报警器开关
- D: 门上的传感器检测到门被打开了
- V: 震动传感器检测到车辆的震动

❖ 输出：报警信号S

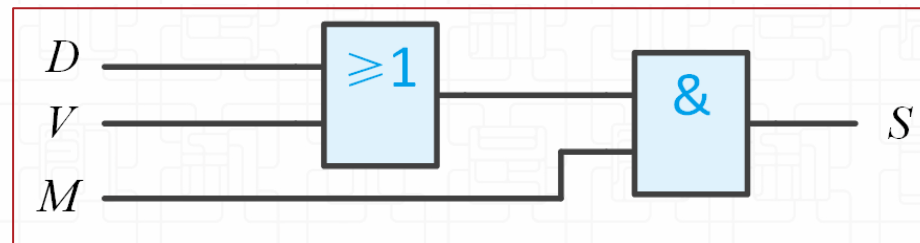
❖ 问题:

- 当一个小偷打开门后，报警信号被触发
- 小偷迅速关上门，报警信号又被关闭了
- 于是小偷可以毫无压力的在车内偷东西

❖ 为什么会这样？

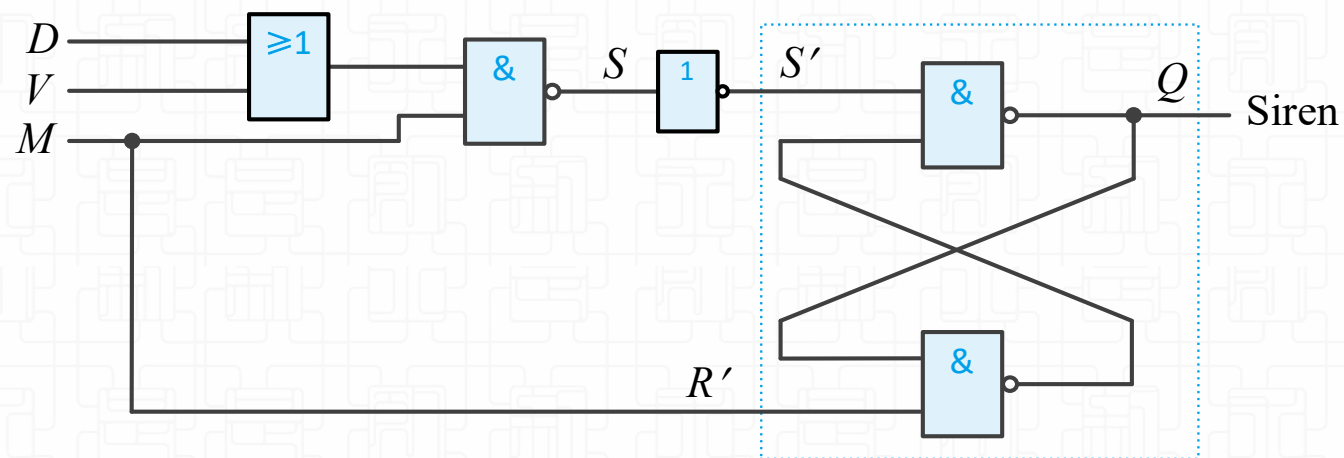
- 因为这个报警器电路没有记忆功能

M	D	V	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



内容提要

- ❖ 组合逻辑电路
- ❖ 时序逻辑电路
- ❖ 有限状态机



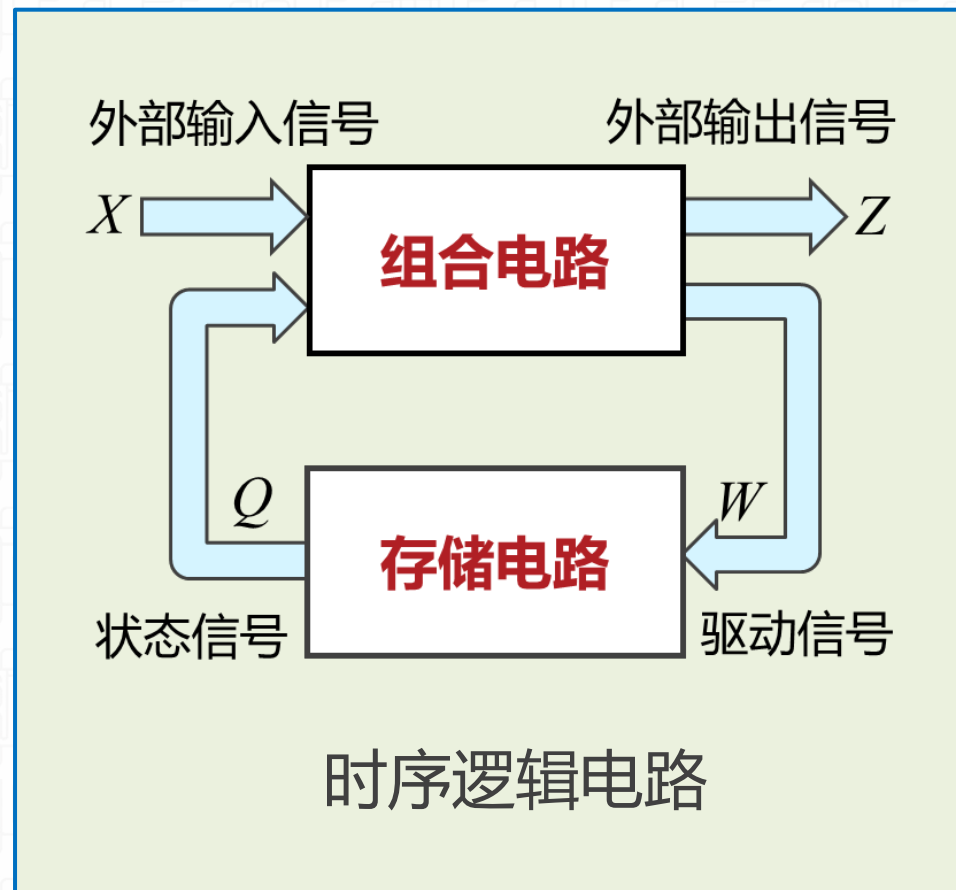
数字逻辑电路

❖ 所有数字逻辑电路都可以被分成两类：

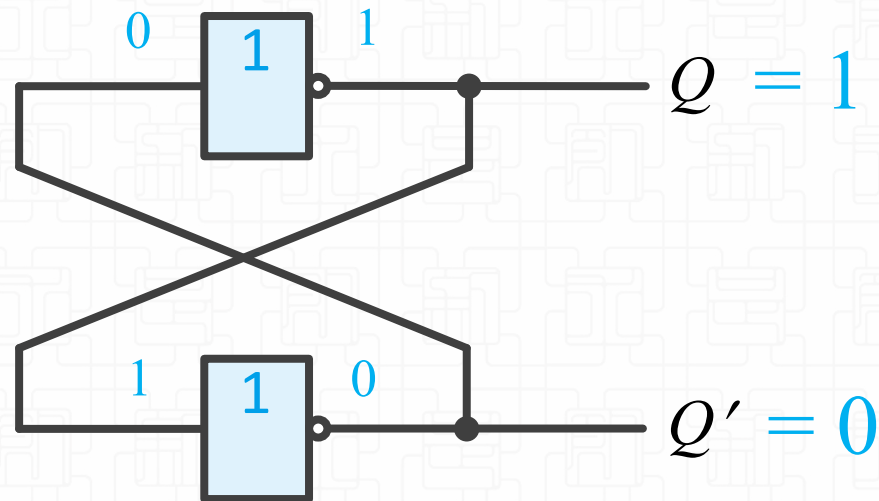
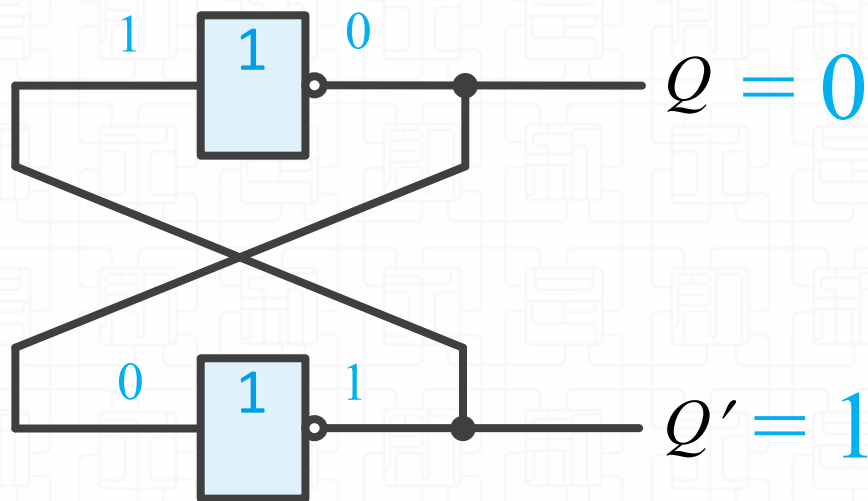
- **组合逻辑电路**：输出只和当前的输入有关，即没有记忆功能。
- **时序逻辑电路**：输出不仅和当前的输入有关，还和以前的输入有关，即具有记忆功能。

❖ 无论是组合逻辑电路，还是时序逻辑电路，都可以由最基本的与门，或门和非门组成。

❖ 所不同的是连接方式不一样，时序逻辑电路的基本门电路连接中，带有**反馈回路**。



如何使电路具有记忆功能：引入反馈



❖ Q and Q' 是对称的两个信号

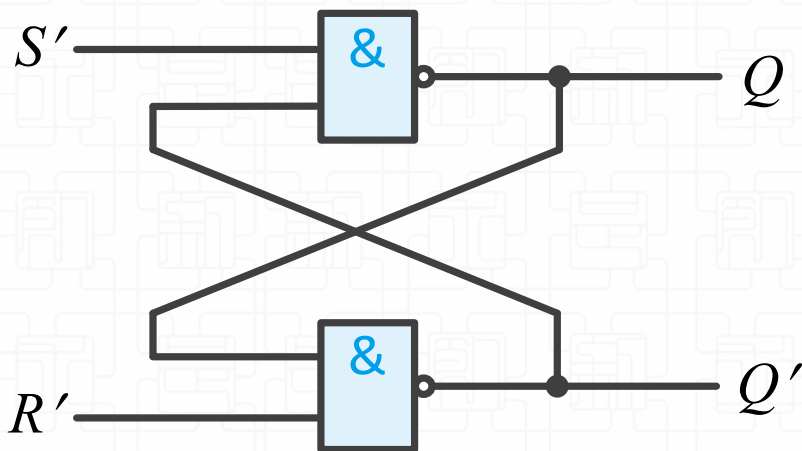
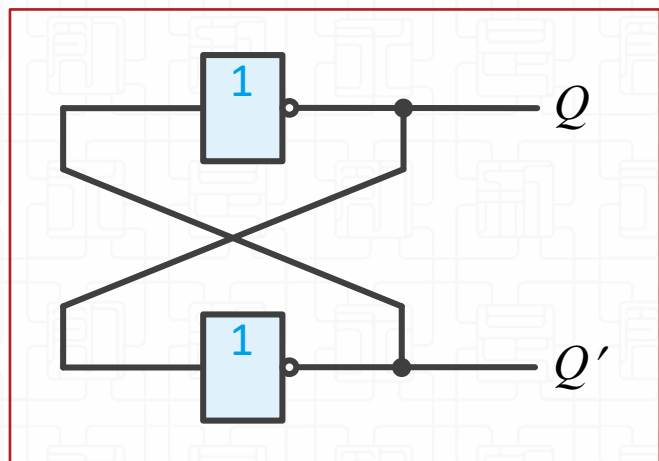
- 假设 $Q=0$, 则下面的反相器输出 $Q'=1$, 作为上面反相器输入后 $Q=0$; 是一个稳定状态
- 假设 $Q=1$, 则下面的反相器输出 $Q'=0$, 作为上面反相器输入后 $Q=1$; 也是一个稳定状态

双稳态电路

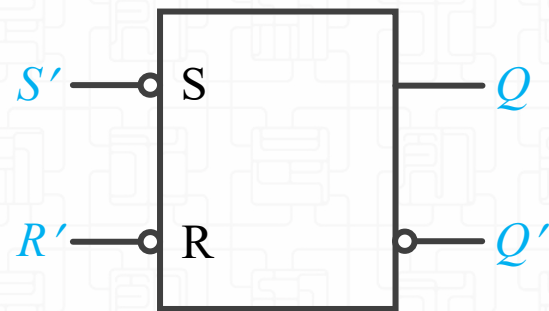
❖ 这一电路就具有了记忆功能, 能够记住电路的状态

❖ 双稳态电路的缺点: 虽然能够记住电路状态, 但是不能改变电路状态

SR锁存器



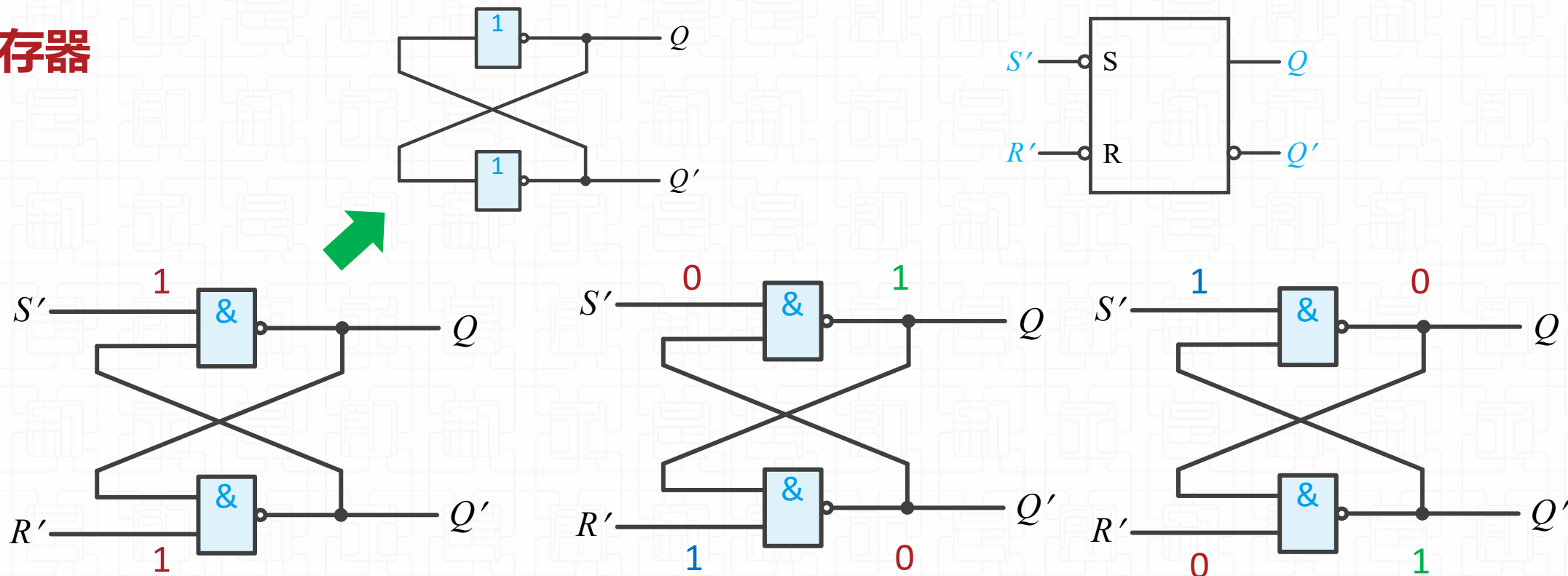
电路图



符号

❖ SR锁存器把双稳态电路中的**非门**换成了**与非门**，使得与非门的输入可以作为外部输入，用于改变电路状态。

SR锁存器

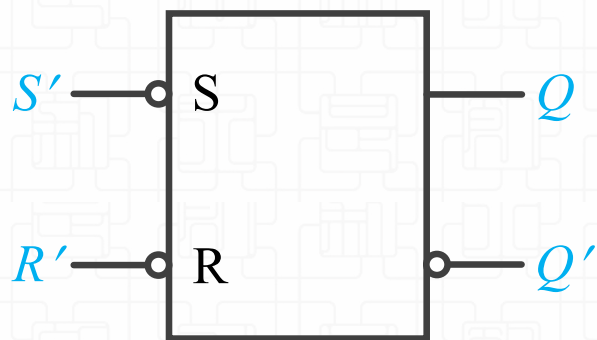
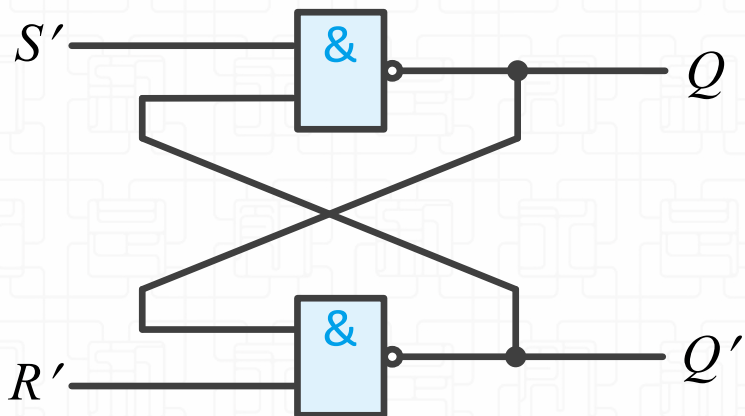


❖ 若 $S'=1$ 且 $R'=1$, 则退化为双稳态电路。

❖ 若两者中间有一个为0, 如 $S'=0$, 则 Q 被置为1; 如 $R'=0$, 则 $Q'=1$

❖ 若两者均为0, 则系统状态可能会进入不稳定或不确定, 应予以避免。

SR锁存器状态表

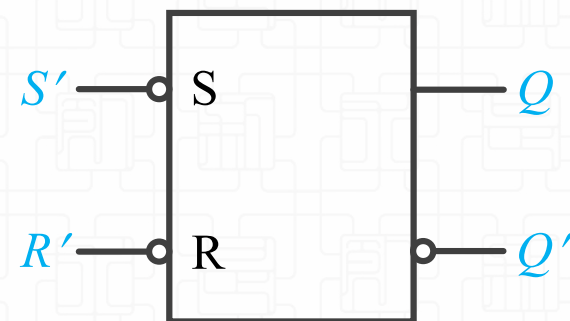


S'	R'	Q	Q_{next}	功能
0	0	0	×	不定
0	0	1	×	
0	1	0	1	$Q_{next} = 1$ 置 1
0	1	1	1	
1	0	0	0	$Q_{next} = 0$ 置 0
1	0	1	0	
1	1	0	0	$Q_{next} = Q$ 保持
1	1	1	1	

现态和次态的概念

❖ 现态 Q 锁存器接收输入信号之前的状态。

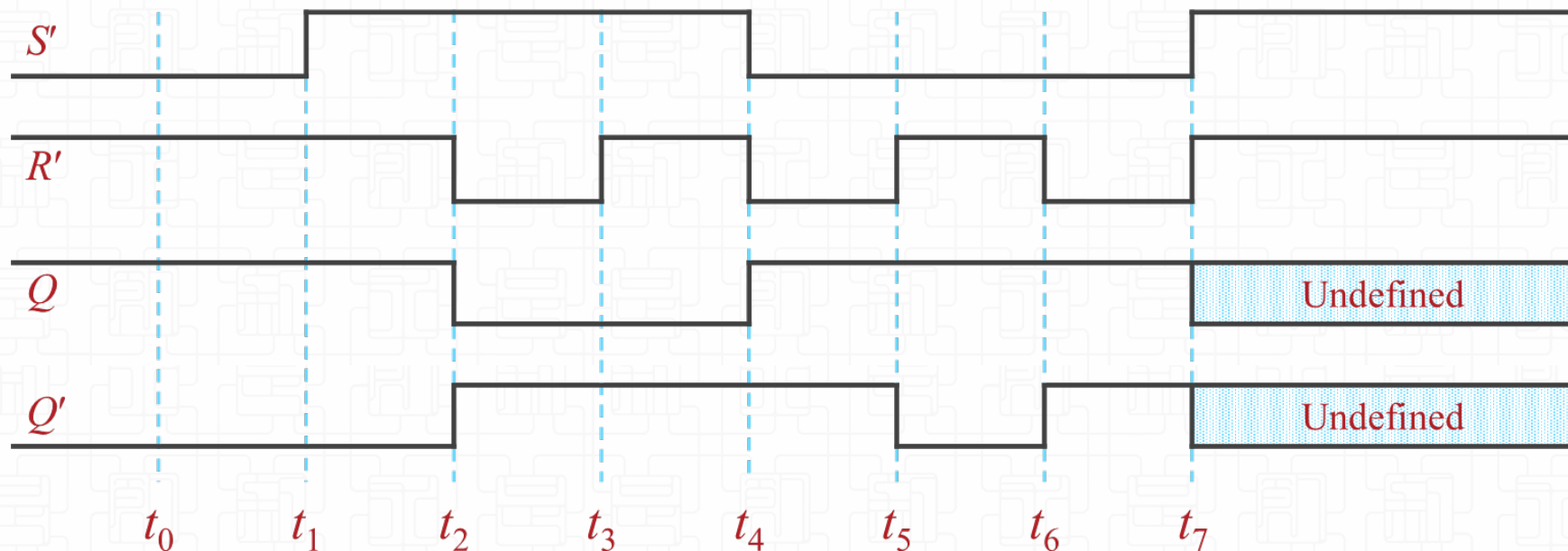
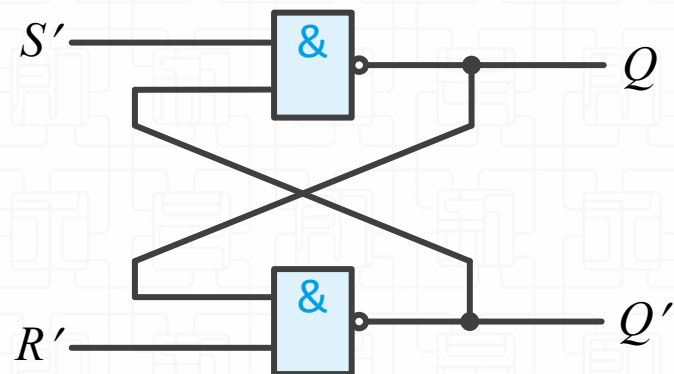
❖ 次态 Q_{next} 锁存器接收输入信号之后的状态。



S'	R'	Q	Q_{next}	功能
0	0	0	×	不定
0	0	1	×	
0	1	0	1	$Q_{next} = 1$ 置1
0	1	1	1	
1	0	0	0	$Q_{next} = 0$ 置0
1	0	1	0	
1	1	0	0	$Q_{next} = Q$ 保持
1	1	1	1	

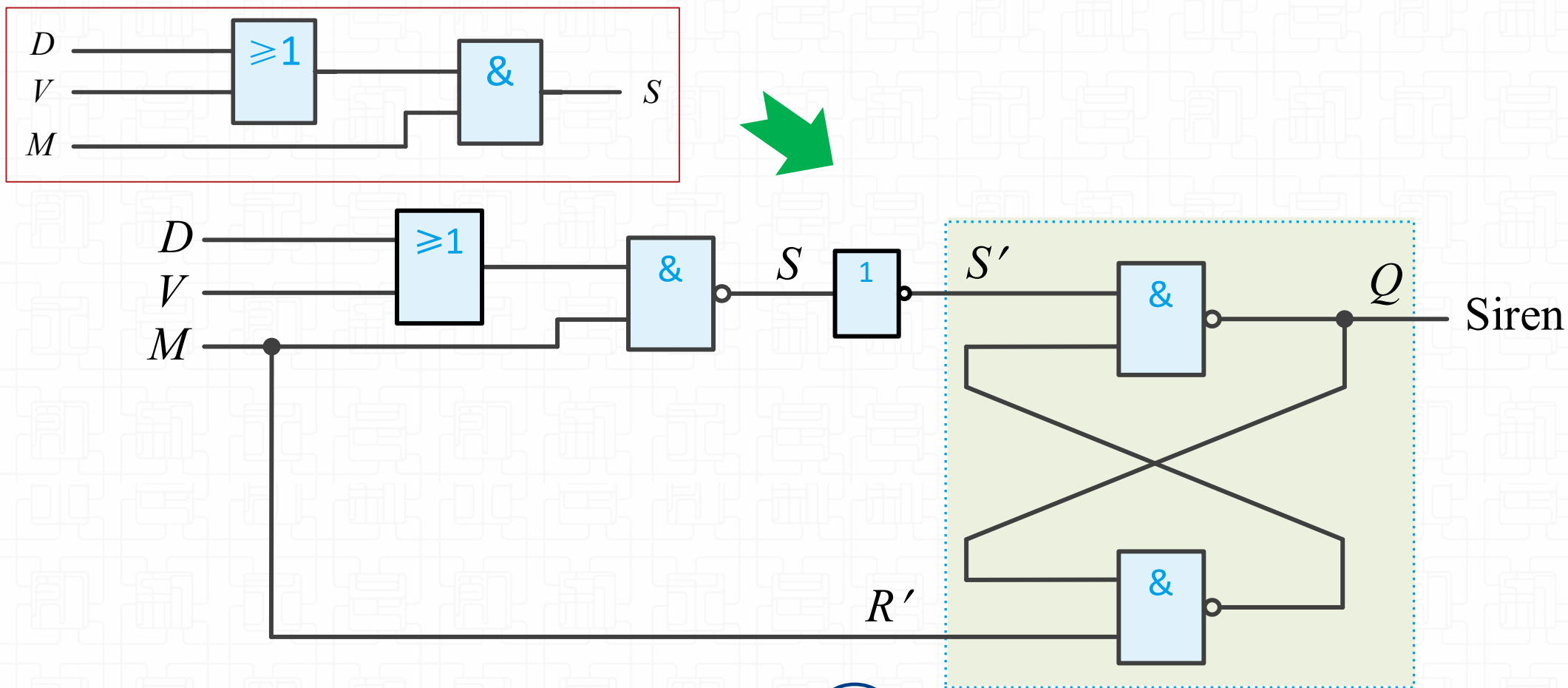
时序

❖ 时序，简单说就是时间顺序。数字系统中每种数字设备具有特定的逻辑功能，要求各个部分按照预先规定的逻辑程序进行工作，因此，数字系统中的信号是**有序的信息流**，各信号之间有**严格的时序关系**。



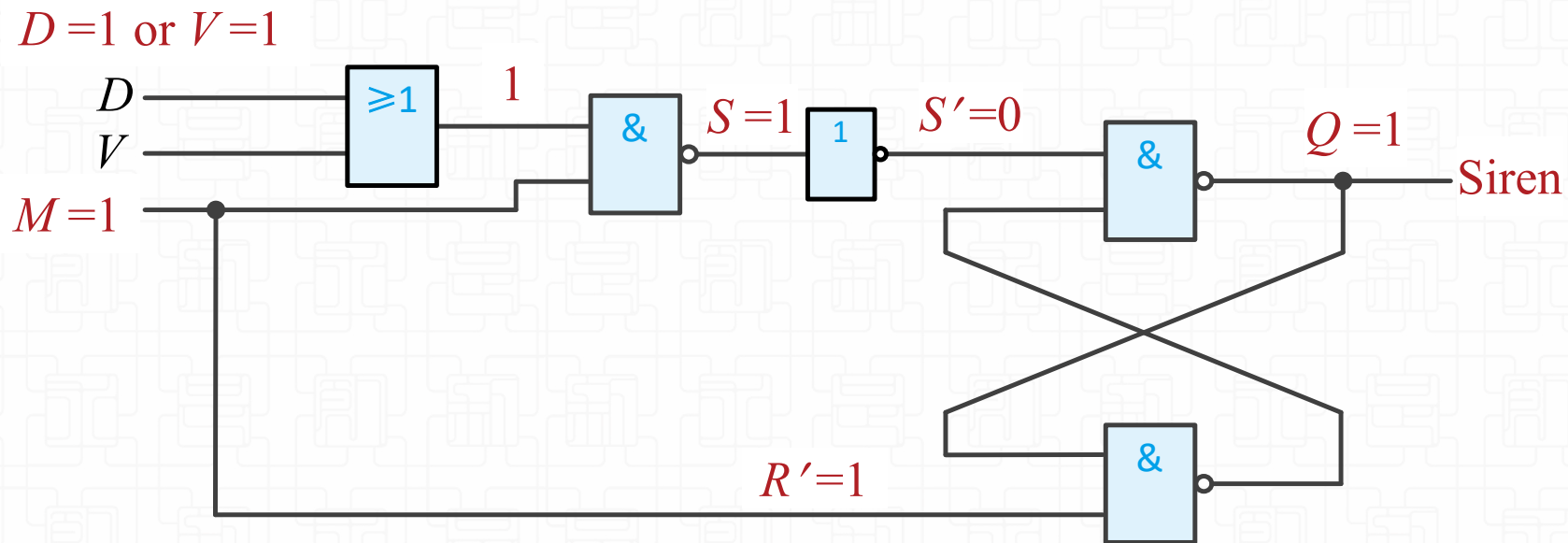
具有记忆功能的汽车报警器

❖ 在原先用组合逻辑电路实现的汽车报警器的基础上，加入SR锁存器，可以设计出新的汽车报警器。

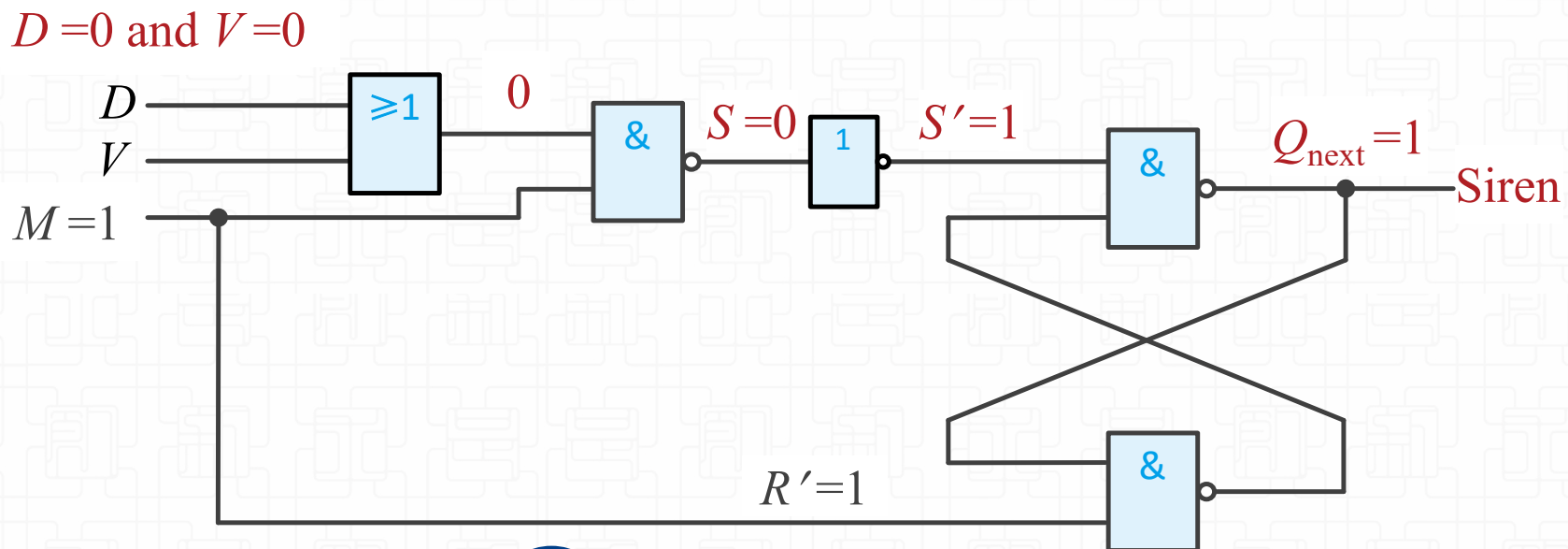


汽车报警器v2.0

车门开关传感器或
震动传感器被触发

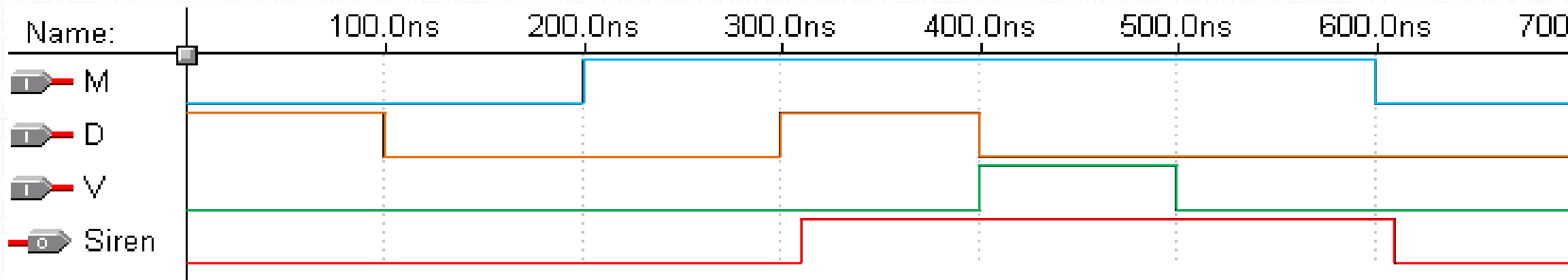
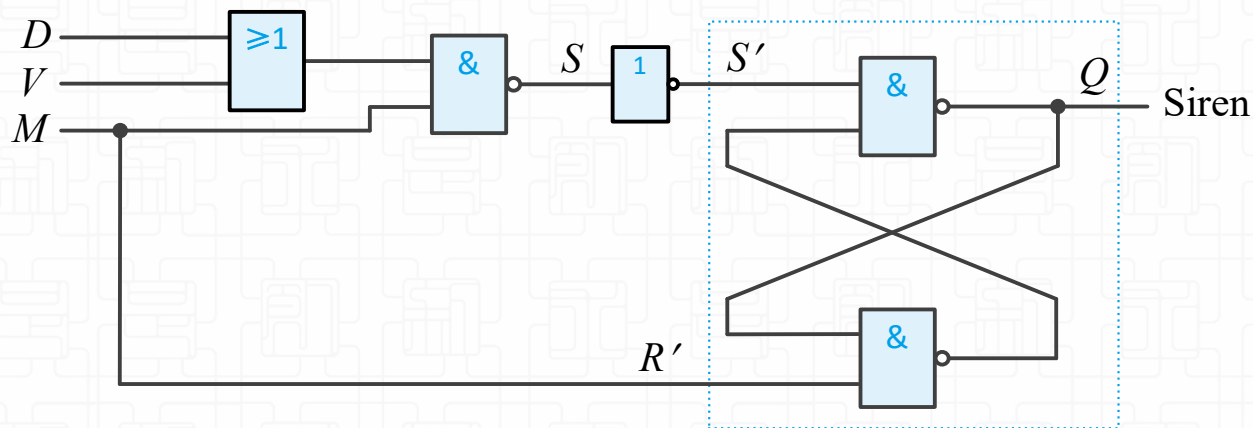


此后，车门开关传
感器和震动传感器
被关闭



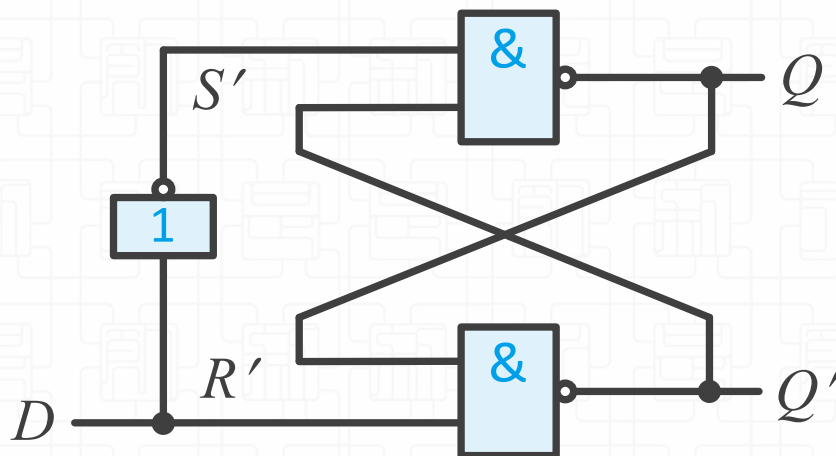
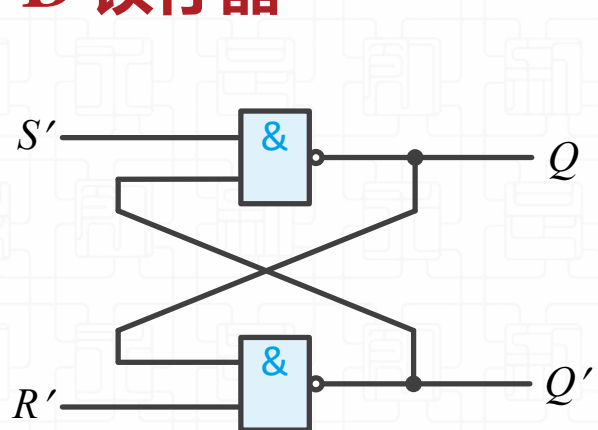
汽车报警器v2.0

❖ 在原先用组合逻辑电路实现的汽车报警器的基础上，加入SR锁存器。

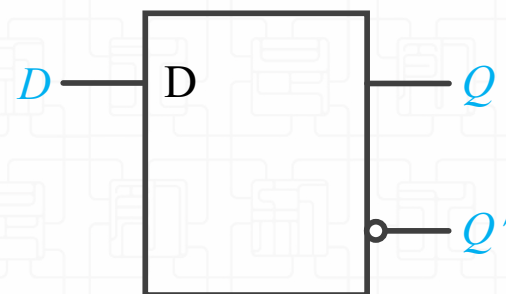


❖ **200ns** 之前，M为0（开关未开时），报警器输出为0；门被打开（**300ns** 时），或检测到汽车震动（**400ns** 时），报警器输出为1；**500ns** 时，虽然 D 和 V 都回复到 0，但报警器仍为 1，即具有记忆功能；直到 **600ns** 时，M被关闭，报警器输出为0。

D 锁存器



D锁存器电路图



电路符号

❖ SR锁存器存在的问题:

- 输入均为0, 则系统状态可能会进入不稳定或不确定, 应予以避免。

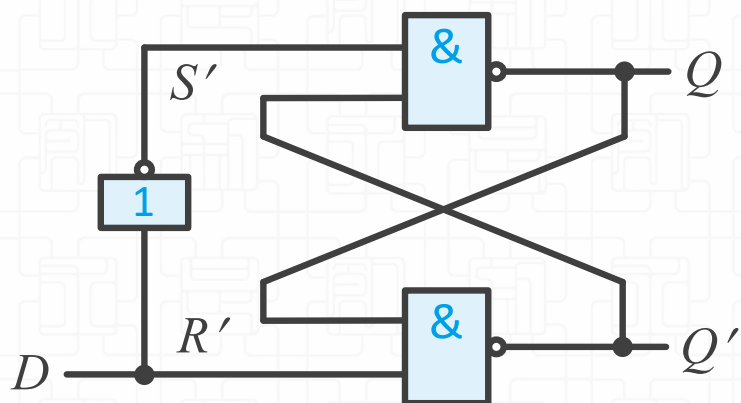
❖ 如何避免?

- 仅使用一个输入, 分别将该信号和它的取反信号接入到SR的输入端
- 虽然避免了同时取0的情况, 但也无法同时取1了, 即失去了保持状态的功能。

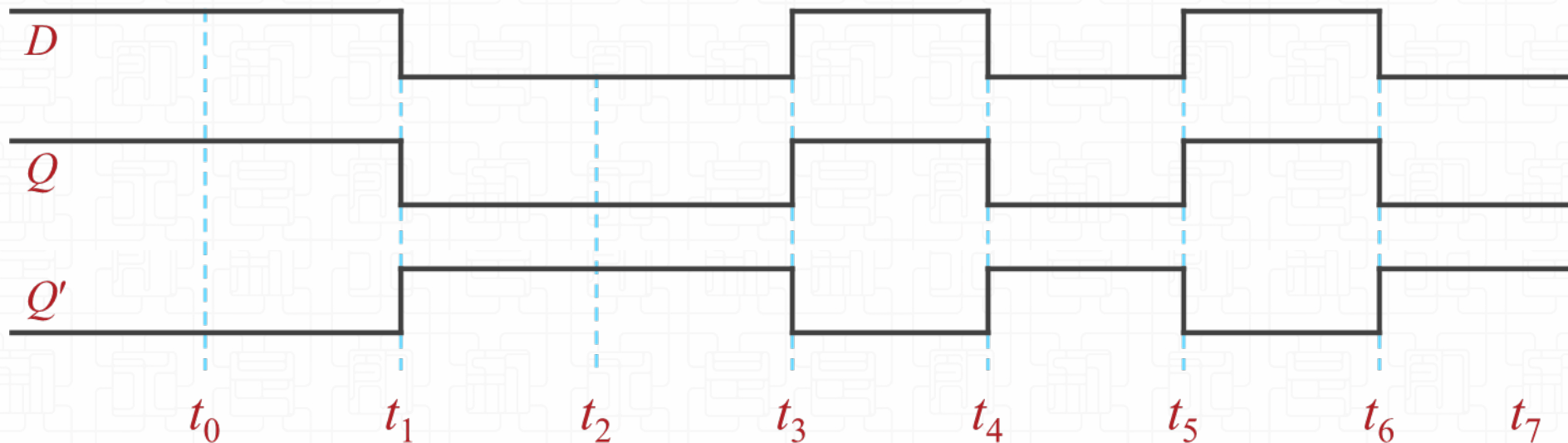
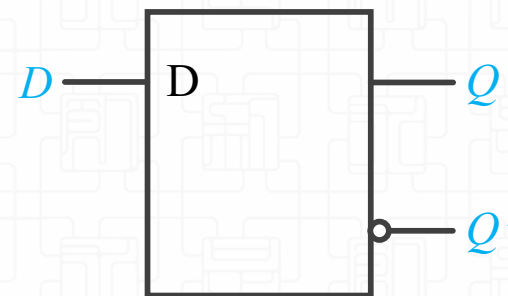
D锁存器真值表

D	Q	Q_{next}	Q'_{next}
0	×	0	1
1	×	1	0

D 锁存器的时序图

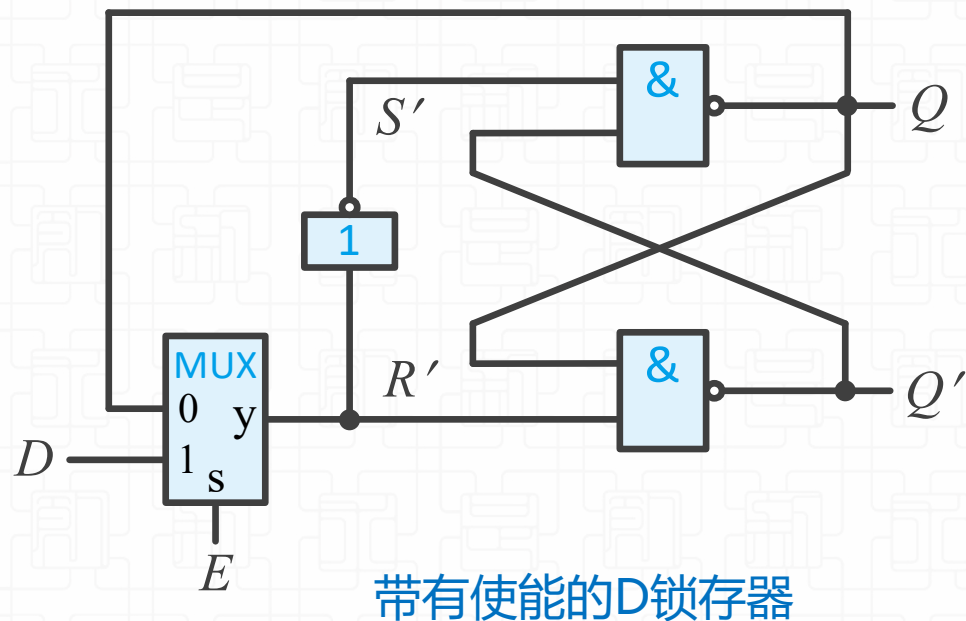
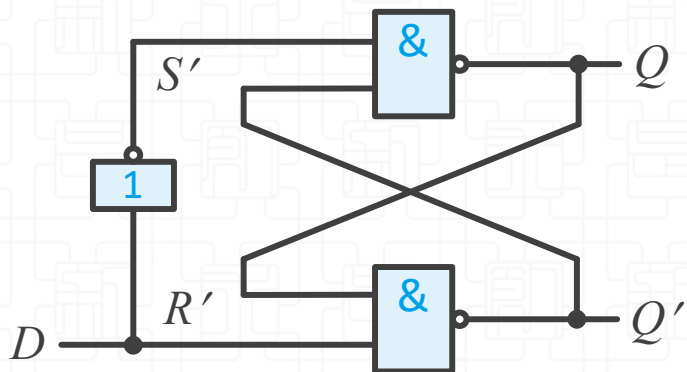


D	Q	Q_{next}	Q'_{next}
0	\times	0	1
1	\times	1	0

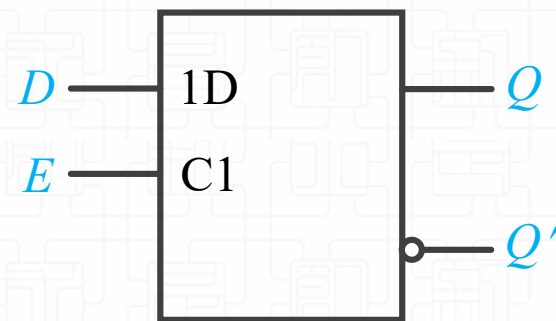


带有使能的 D 锁存器

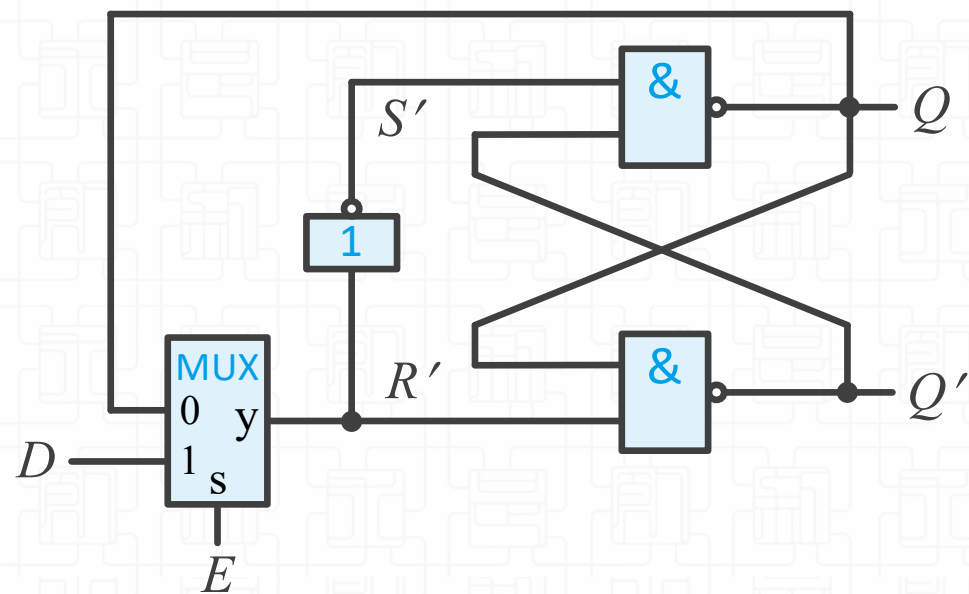
❖ 加入输入使能信号，即带有使能的D锁存器。



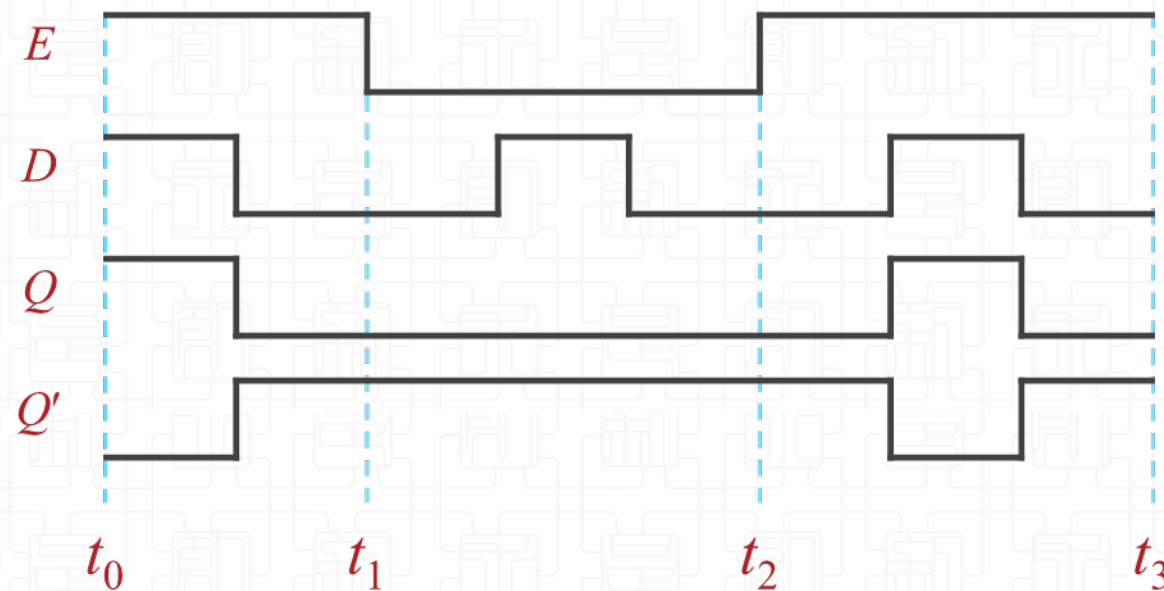
E	D	Q	Q'	Q_{next}	Q'_{next}
0	×	0	1	0	1
0	×	1	0	1	0
1	0	×	×	0	1
1	1	×	×	1	0



带有使能的 D 锁存器时序图

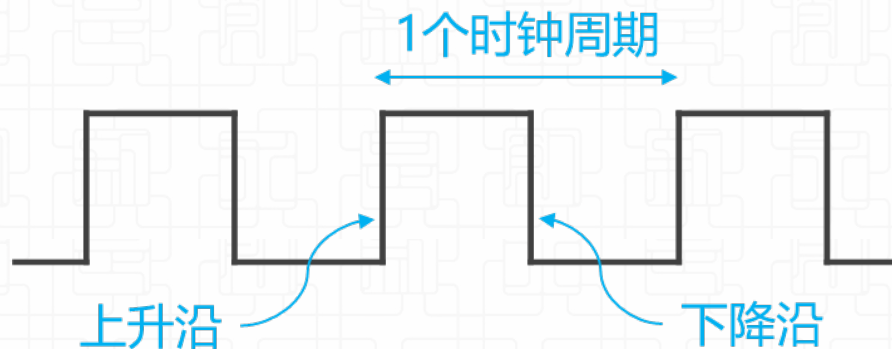


E	D	Q	Q'	Q_{next}	Q'_{next}
0	\times	0	1	0	1
0	\times	1	0	1	0
1	0	\times	\times	0	1
1	1	\times	\times	1	0

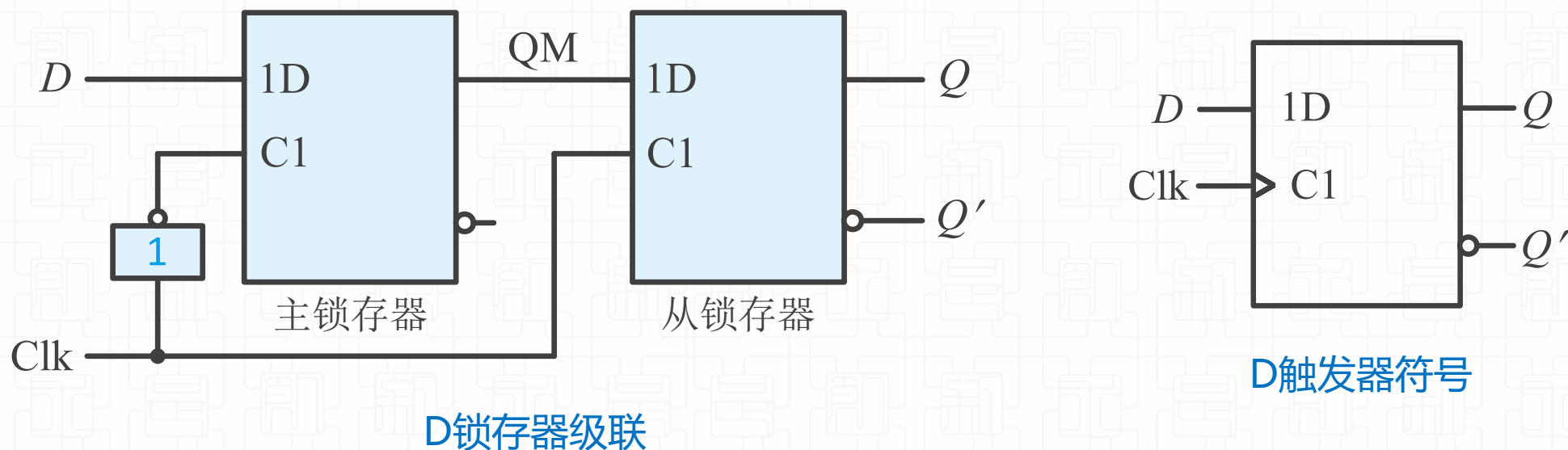


时钟与时序电路的同步工作

- ❖ 类似于很多人进行起步走一样，需要有一个1-2-1的同步指令，一个复杂的数字系统要正确工作，也需要有一个用于同步的时钟信号。
- ❖ 时钟信号具有上升沿和下降沿，如果可以使用**上升沿**（或**下降沿**）进行触发，则可以实现电路**同步**工作。
- ❖ 如何在锁存器中使用时钟的上升沿或下降沿进行触发？无法做到。
- ❖ 需要一种可以用**时钟边沿**进行触发的时序电路。



D 触发器



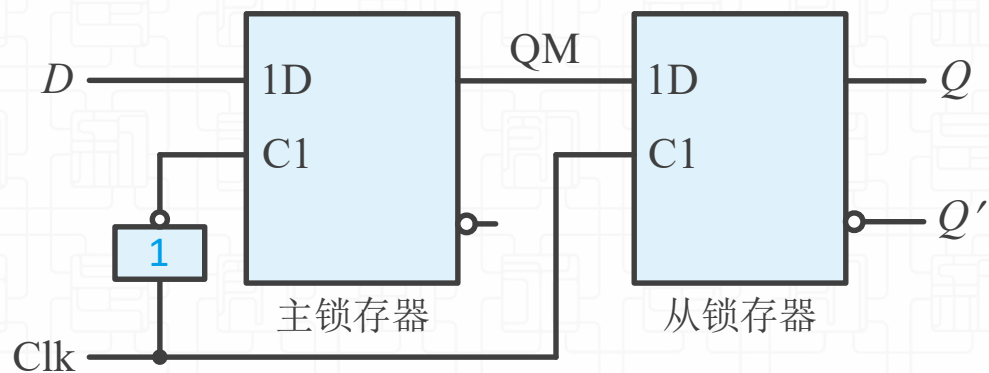
❖ 将两个带使能的 D 锁存器进行级联

❖ 锁存器是电平触发

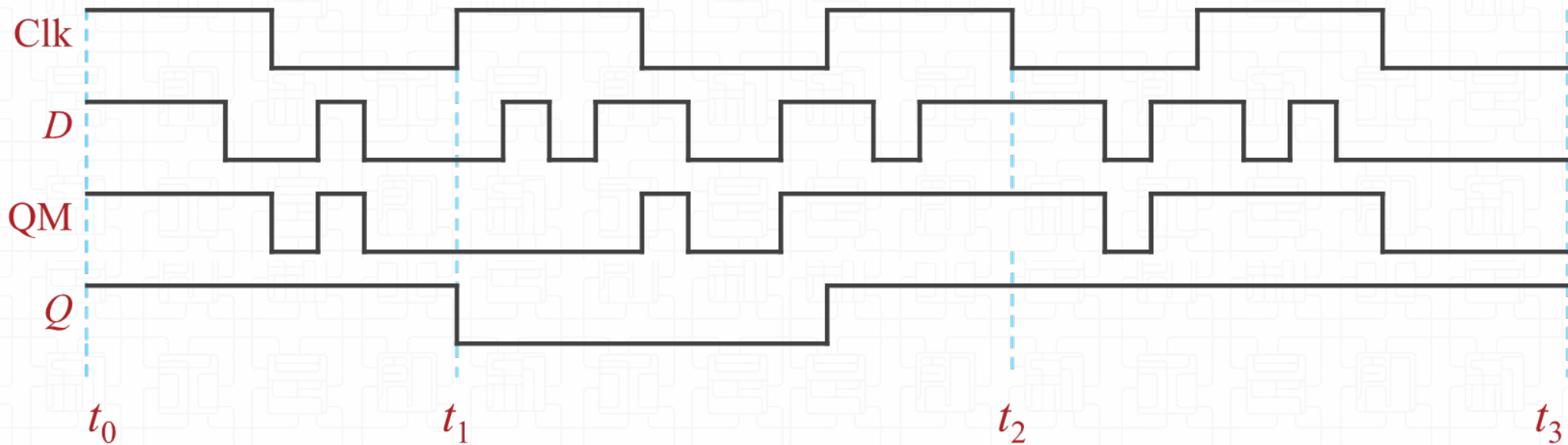
- 对于前一锁存器，只有 Clk 低电平时的信号能够通过
- 对于后一锁存器， Clk 为高时候的输出，是 Clk 为低时候最后时刻的输入

❖ D 触发器的输出，为 Clk 上升沿时候的信号

D 触发器时序图

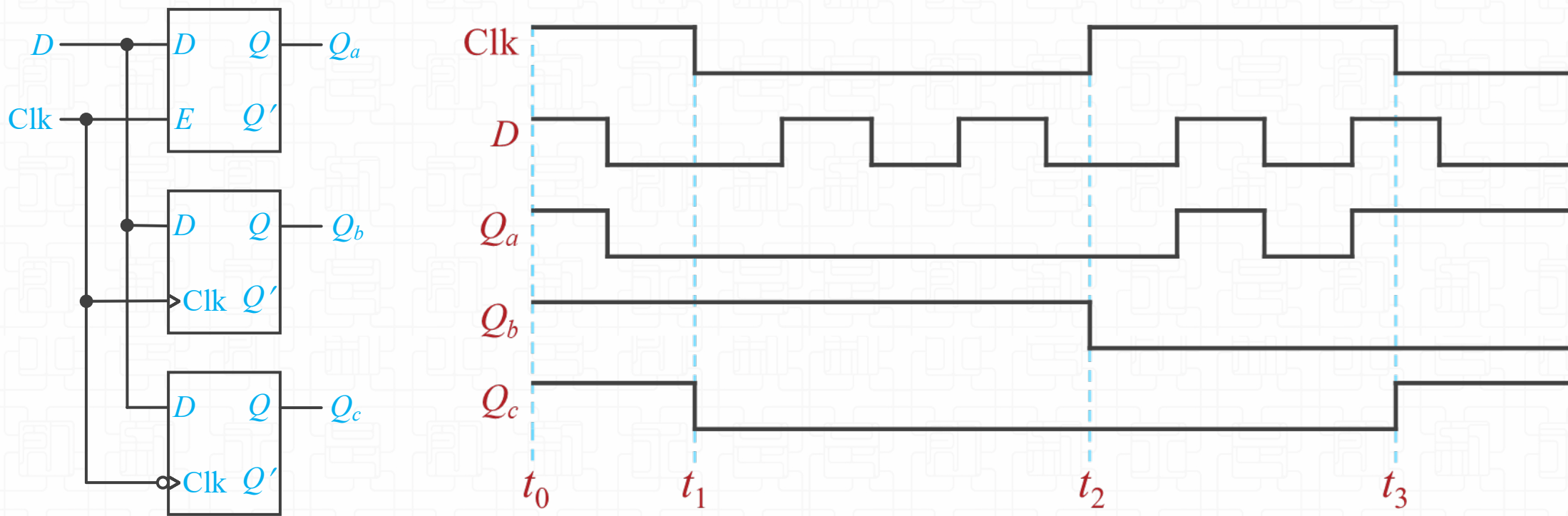


Clk	D	Q	Q'	Q_{next}	Q'_{next}
0	\times	0	1	0	1
0	\times	1	0	1	0
1	\times	0	1	0	1
1	\times	1	0	1	0
\uparrow	0	\times	\times	0	1
\uparrow	1	\times	\times	1	0



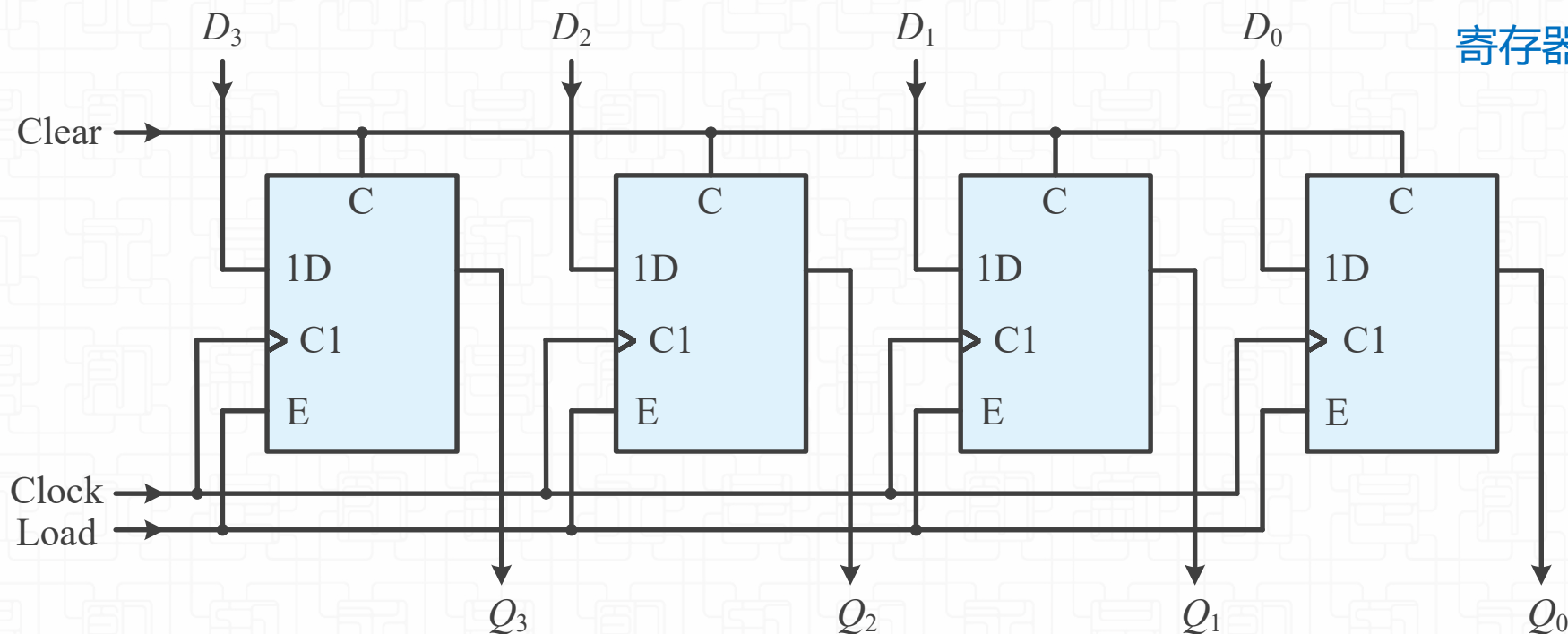
D锁存器与D触发器功能比较

- ❖ 第一个是D锁存器，只要时钟为**高电平**，输出即等于输入。
- ❖ 第二个是上升沿触发，只有在时钟**上升沿**出现的信号才会被采样输出。
- ❖ 第三个是下降沿触发，只有在时钟**下降沿**出现的信号才会被采样输出。



寄存器

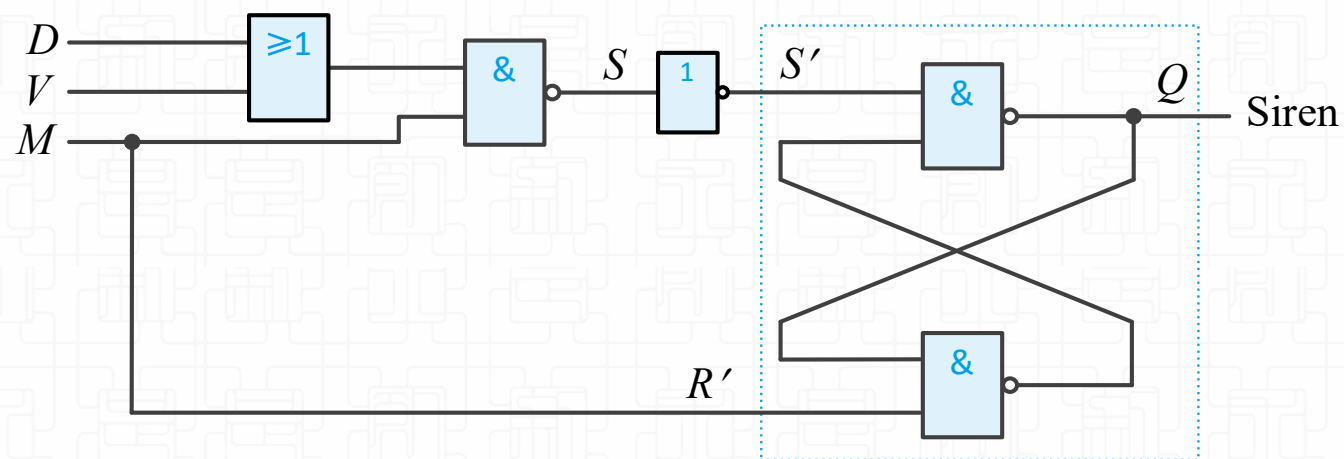
寄存器电路结构图



- ❖ 将 n 个D触发器并行连接，即可构成一个 n 位的寄存器
- ❖ 在同一个时刻（时钟上升沿），输出信号的每一位被更新为对应的输入信号，输出是同步的。
- ❖ 两个控制信号Clear和Load
 - Clear=1 (有效) 时，寄存器所有输出位均为0
 - Load=1 (有效) 时，寄存器才会在 Clk 上升沿进行更新，否则不变。

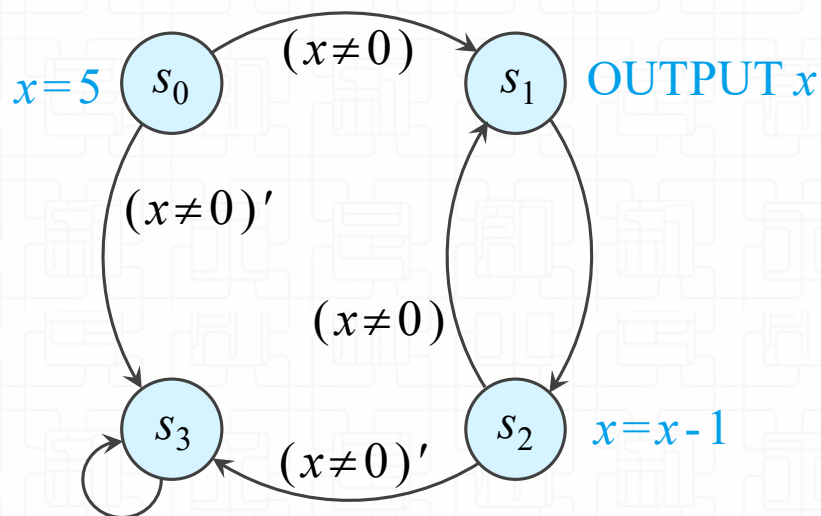
内容提要

- ❖ 组合逻辑电路
- ❖ 时序逻辑电路
- ❖ 有限状态机



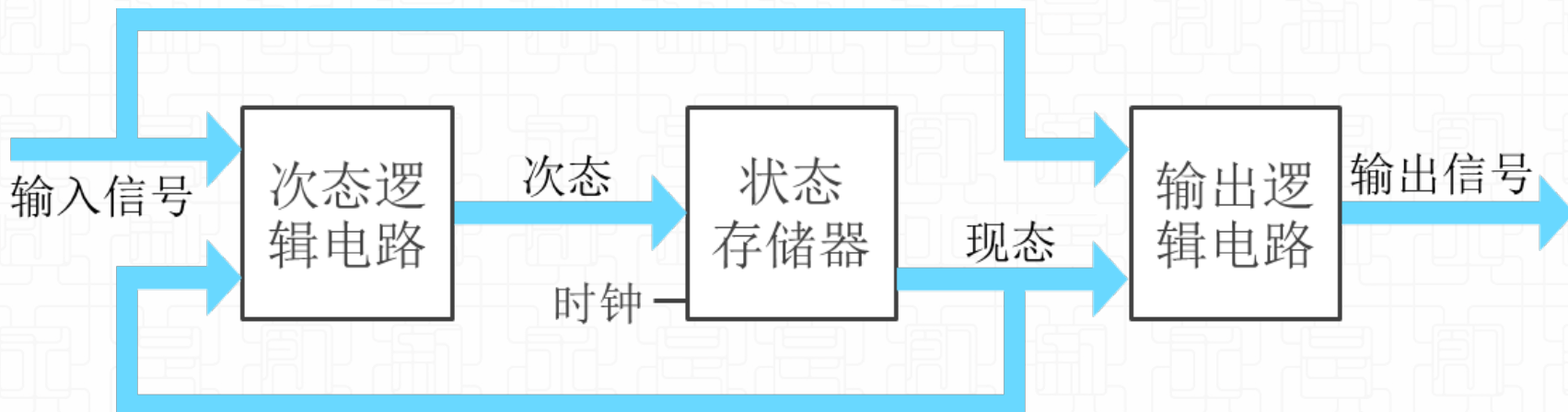
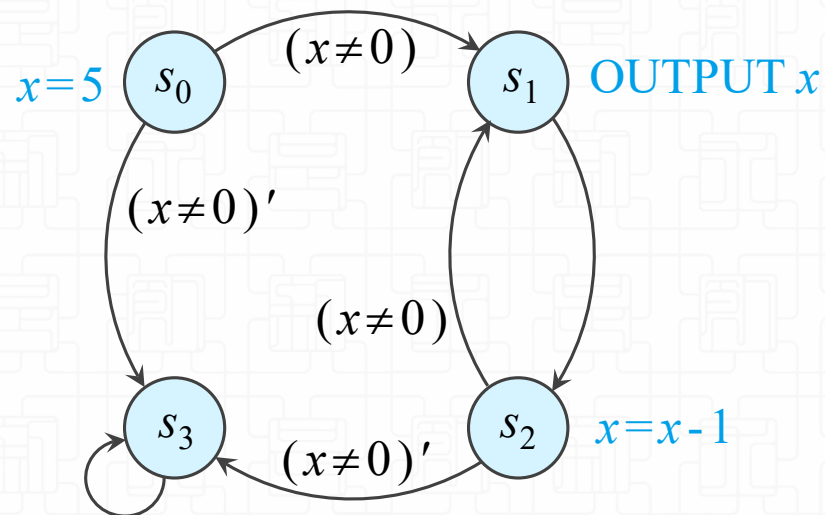
有限状态机 (Finite-State Machine, FSM) 概念

- ❖ 有限状态机又称有限状态自动机，简称状态机，是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型。
- ❖ 是一种用来进行对象行为建模的工具，其作用主要是描述对象在它的生命周期内所经历的状态序列，以及如何响应来自外界的各种事件。
- ❖ 特征
 - 状态总数 (state) 是有限的。
 - 任一时刻，只处在一种状态之中。
 - 某种条件下，会从一种状态转变 (transition) 到另一种状态。



有限状态机电路

- ❖ 有限状态机是一种实现多个状态任务的时序电路的有效抽象。
- ❖ 要完成一个任务，需要分很多步骤，每个步骤看成一个状态。
- ❖ 有限状态机由组合逻辑和时序逻辑（寄存器）两部分组成。
- ❖ 有限状态机的“有限”是指状态机的状态数目是有限的。

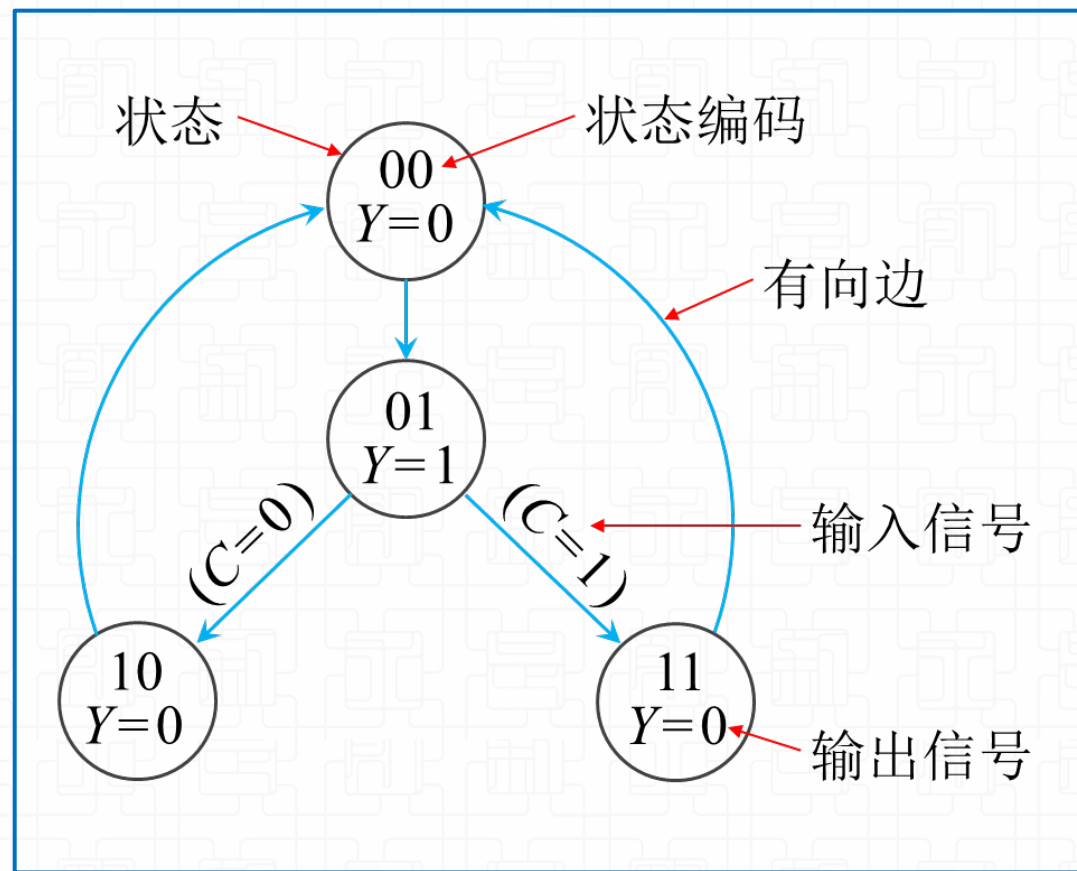


有限状态机

- ❖ 每个**时钟周期**可以看成**一个状态**。
- ❖ 在每一个**时钟边沿**，组合电路根据**当前输入和当前状态**计算输出**下一状态**。
- ❖ 虽然**状态的数量有限**，但是，有限状态机可以不止一次地进入这些状态，所以该有限状态机经过的**状态序列可以无限长**。
- ❖ 有限状态机的状态是**决定性的**，不是随机的。在当前状态下，只要输入信号确定了，组合电路输出的下一状态就是确定的。
- ❖ 有限状态机是所有微处理器的控制电路。

状态机的描述：状态图

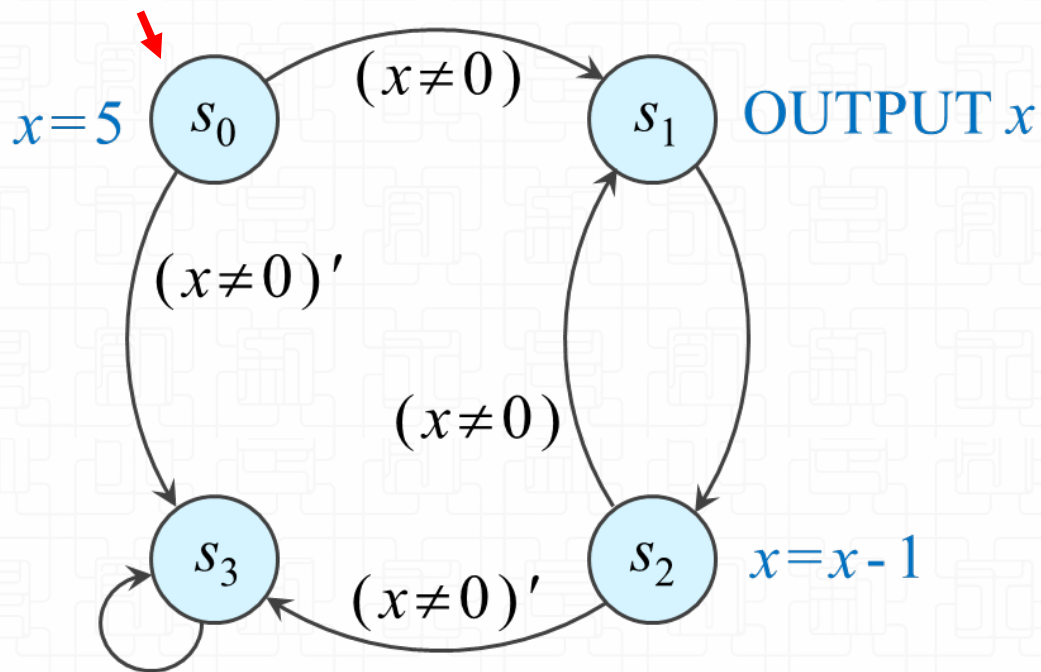
- ❖ 状态图是用于精准描述有限状态机的工作的。
- ❖ 状态图是由多个节点以及节点间连接的有向边构成的确定性图。
- ❖ 有向边表示的是状态之间的跳转，可以是条件性跳转，也可以是无条件跳转。
- ❖ 图中，圆圈表示状态节点，或简称状态，圆圈中的二进制表示状态编码，圆圈中的表达式表示该状态下的输出。
- ❖ 图中，有向边上的表达式表示跳转条件。若无，则是无条件跳转。



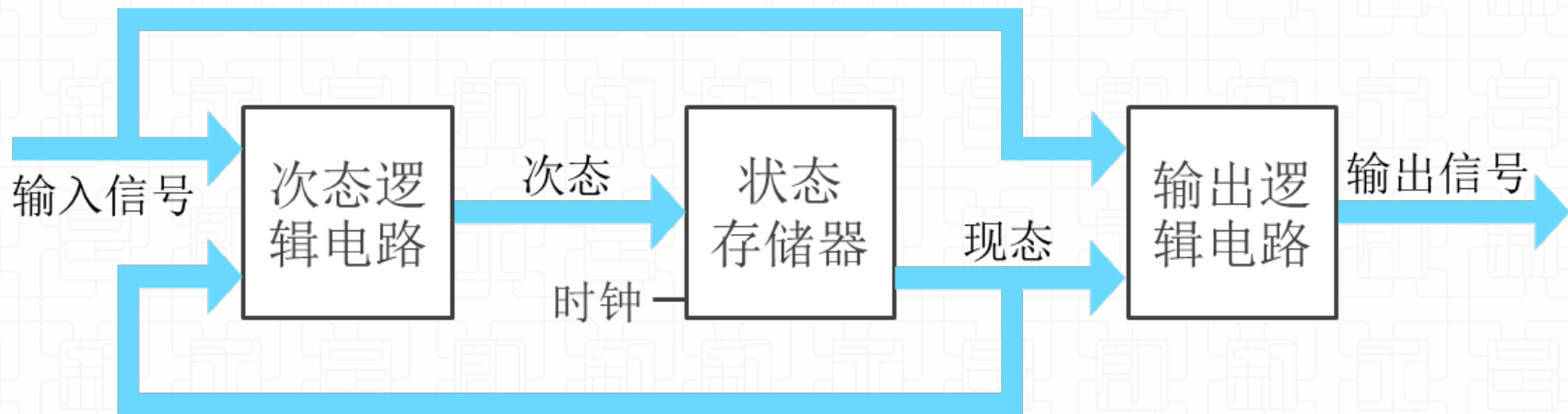
状态图设计

- ❖ 根据跟定的任务写出任务的伪代码
- ❖ 把对于数据的运算和操作对应到状态图的状态节点。
- ❖ 任务所需操作的流程以及条件测试对应到状态图的有向边。
- ❖ 右边例子中，3个操作外加最终的1个停止状态，一共4个状态。
- ❖ 根据伪代码中的执行流程图，可以写出每个状态到下一状态的跳转条件。

```
x = 5  
WHILE (x ≠ 0) {  
    OUTPUT x  
    x = x - 1  
}
```



FSM模型



❖ 次态逻辑 (next-state logic circuit)

- 组合电路；输出依赖于当前状态和当前输入；输出决定了状态机的下一个状态

❖ 状态寄存器 (state memory register)

- 时序电路；由多个D触发器构成；触发器的数据 n 决定了最大状态数 2^n

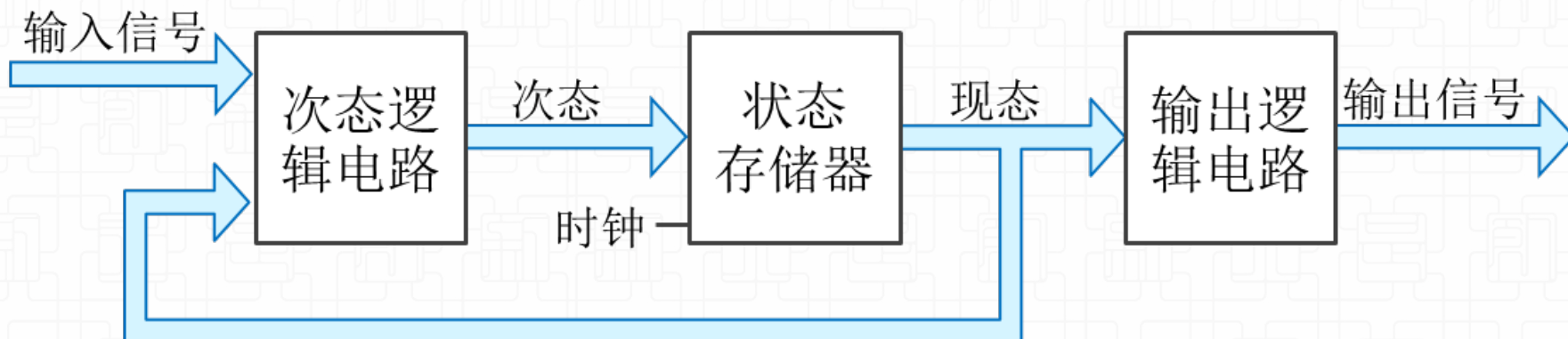
❖ 输出逻辑 (output logic circuit)

- 组合逻辑；输出依赖于当前状态 (Moore) 或同时依赖于当前状态和当前输入 (mealy)；输出控制信号

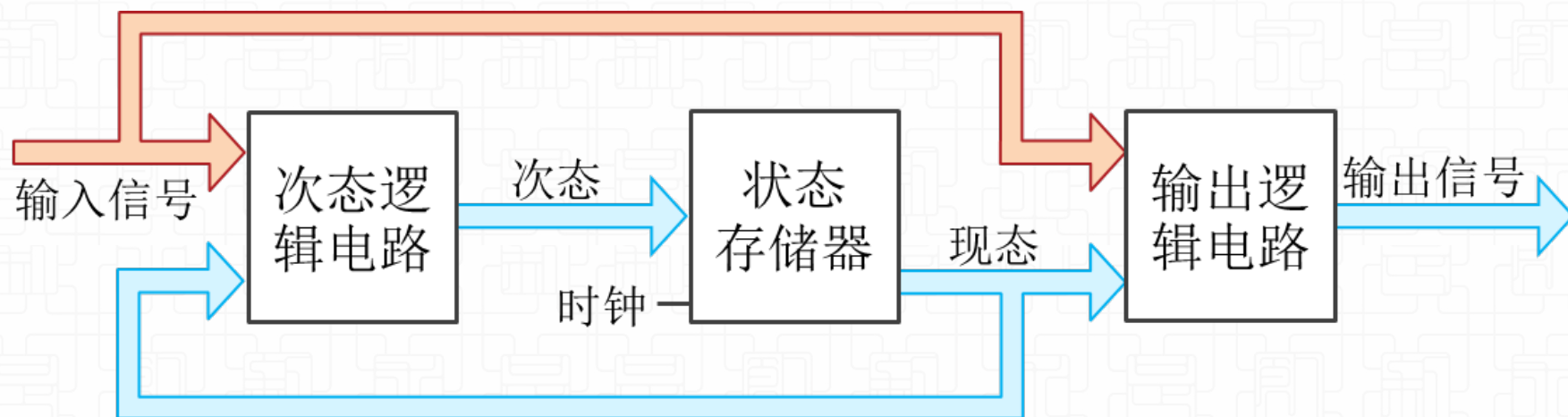
❖ 时钟：每次时钟上升沿状态寄存器内容发生变化，即发生状态跳转。

两种状态机

Moore状态机

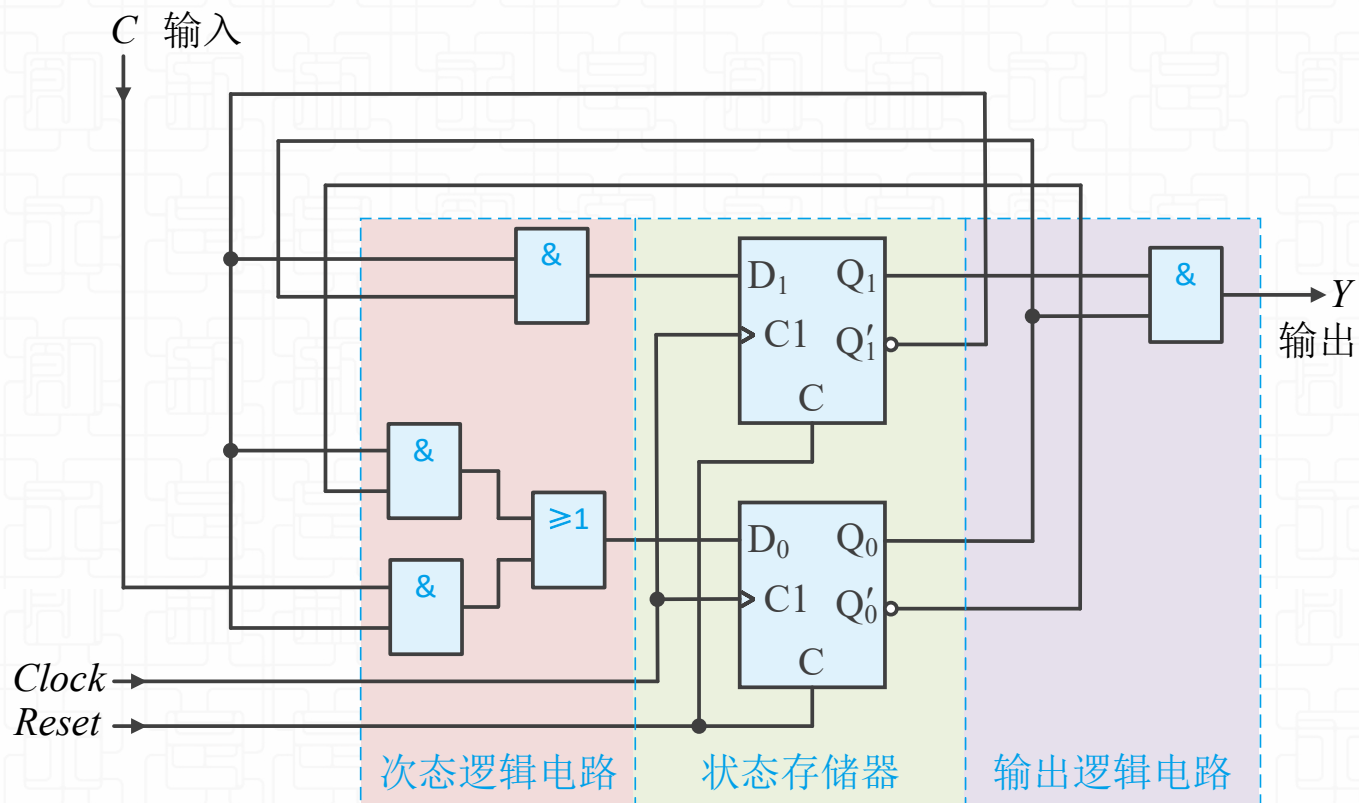
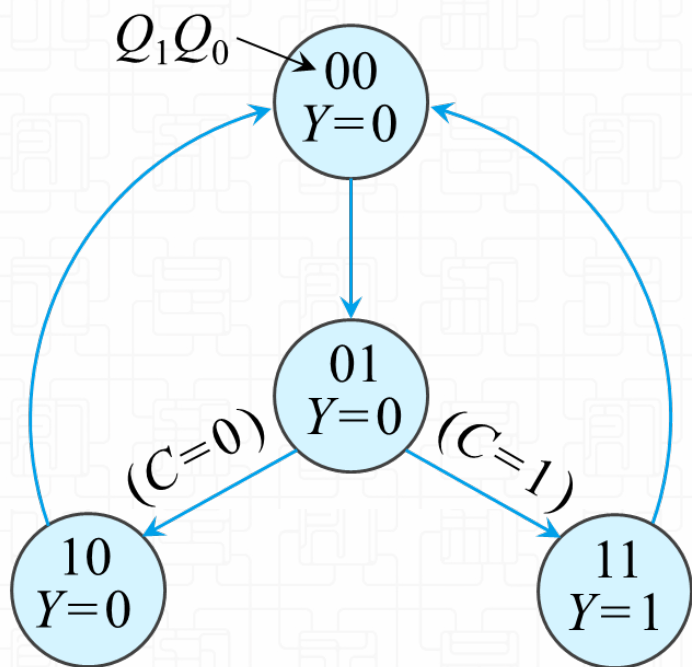


Mealy状态机



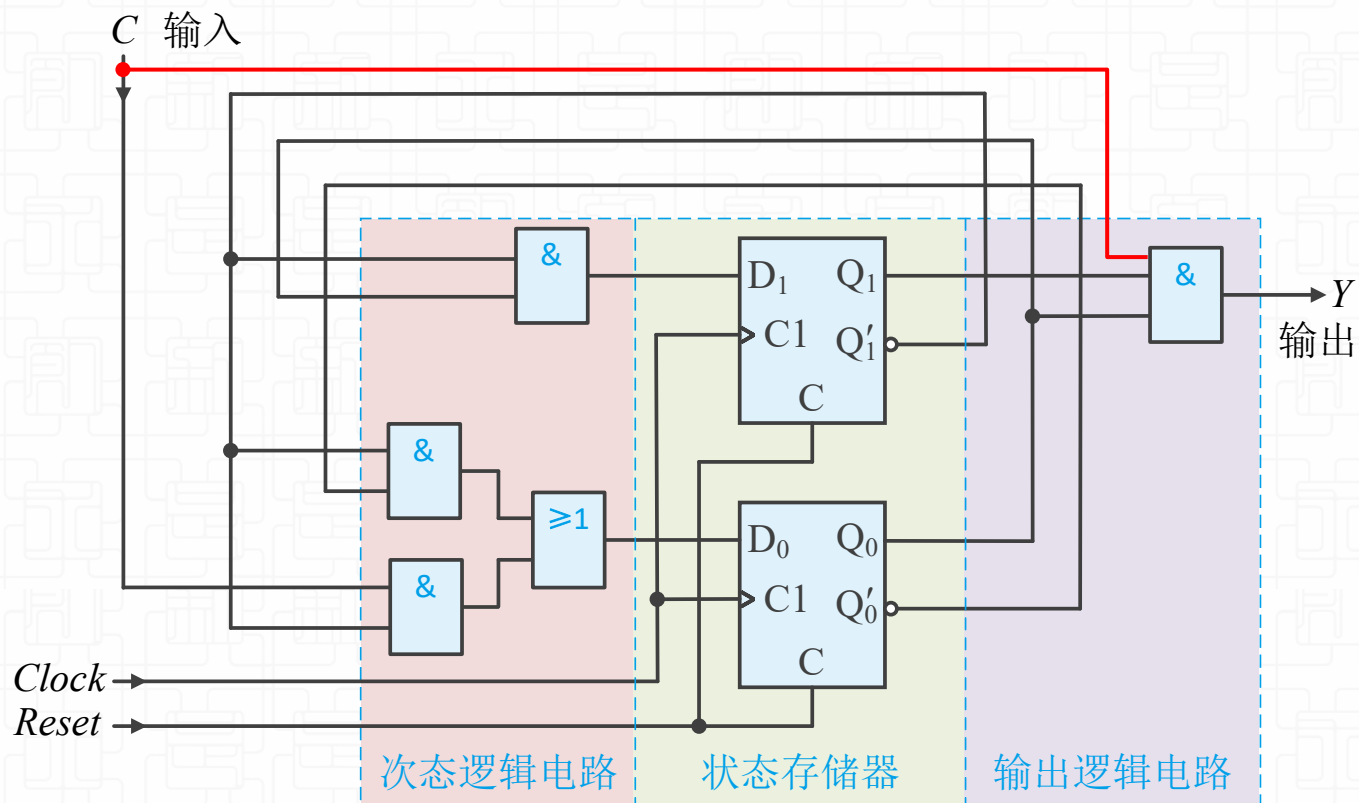
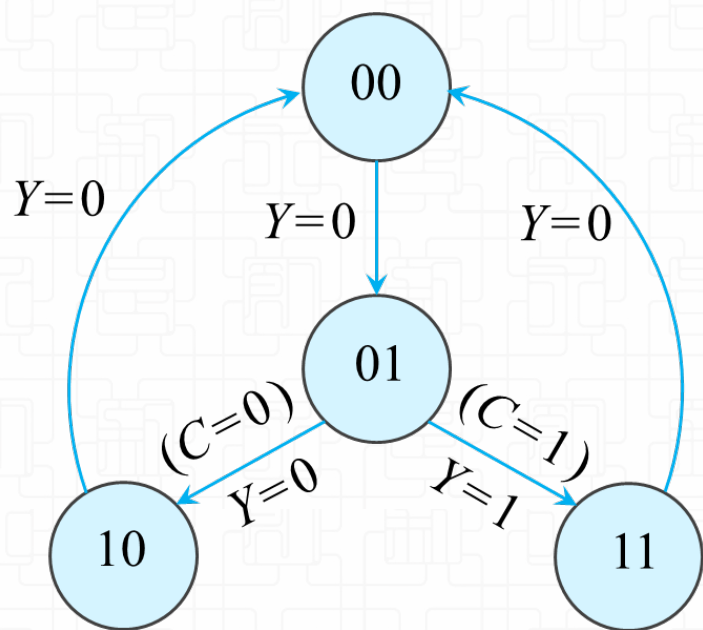
Moore状态机

- ❖ 状态图中，输出信号标于状态中，表示输出仅与当前状态有关
- ❖ 电路图中，输出仅决定于寄存器的输出，与输入信号无关



Mealy状态机

- ❖ 状态图中，输出信号标于有向边上，表示输出与当前状态和当前输入都有关
- ❖ 电路图中，输出决定于寄存器的输出以及输入信号

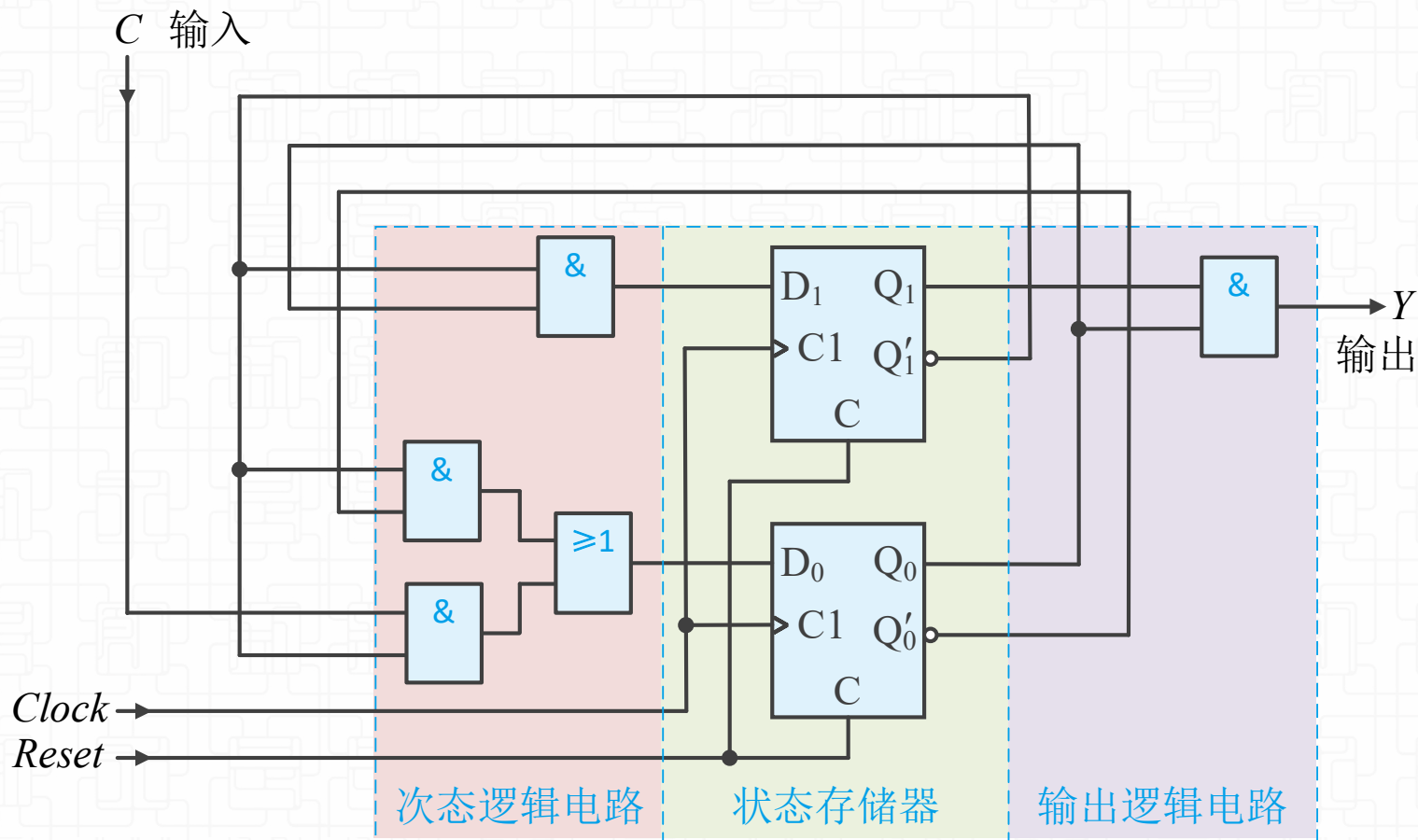


FSM的分析

❖ 有限状态机的分析是指，给定一个状态机电路，分析出该电路的状态图。

❖ 三个步骤：

- 依据次态电路，写出次态电路方程和真值表
- 依据输出电路，写出输出方程和真值表
- 依据第一步和第二步，画出状态图



次态电路方程

$$Q_{1\text{next}} = D_1 = Q_1'Q_0$$

$$Q_{0\text{next}} = D_0 = Q_1'Q_0' + CQ_1'$$

FSM的分析

❖ 有限状态机的分析是指，给定一个状态机电路，分析出该电路的状态图

❖ 三个步骤：

- 依据次态电路，写出次态电路方程和真值表
- 依据输出电路，写出输出方程和真值表
- 依据第一步和第二步，画出状态图

$$\begin{aligned}Q_{1\text{next}} &= D_1 = Q_1'Q_0 \\Q_{0\text{next}} &= D_0 = Q_1'Q_0' + CQ_1'\end{aligned}$$

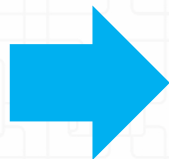
❖ 真值表如右图所示，左边一列表示输入信号的所有可能取值，右边一列表述两个状态寄存器随输入信号变化的输出

CQ_1Q_0	$Q_{1\text{next}}Q_{0\text{next}}$
000	01
001	10
010	00
011	00
100	01
101	11
110	00
111	00

FSM的分析

❖ 真值表形式稍作调整

CQ_1Q_0	$Q_{1next} Q_{0next}$
000	01
001	10
010	00
011	00
100	01
101	11
110	00
111	00



当前状态 Q_1Q_0	次态 $Q_{1next} Q_{0next}$	
	$C = 0$	$C = 1$
00	01	01
01	10	11
10	00	00
11	00	00

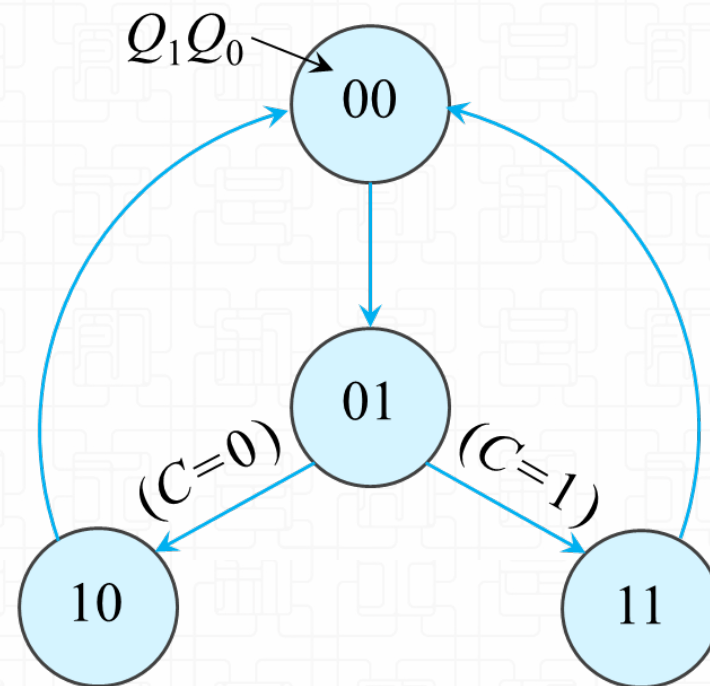
FSM的分析

❖ 有限状态机的分析是指，给定一个状态机电路，分析该电路的状态图

❖ 三个步骤：

- 依据次态电路，写出次态电路方程和真值表
- 依据输出电路，写出输出方程和真值表
- 依据**第一步**和**第二步**，画出**状态图**

当前状态 Q_1Q_0	次态 $Q_{1next} Q_{0next}$	
	$C = 0$	$C = 1$
00	01	01
01	10	11
10	00	00
11	00	00



FSM的分析

❖ 有限状态机的分析是指，给定一个状态机电路，分析出该电路的状态图。

❖ 三个步骤：

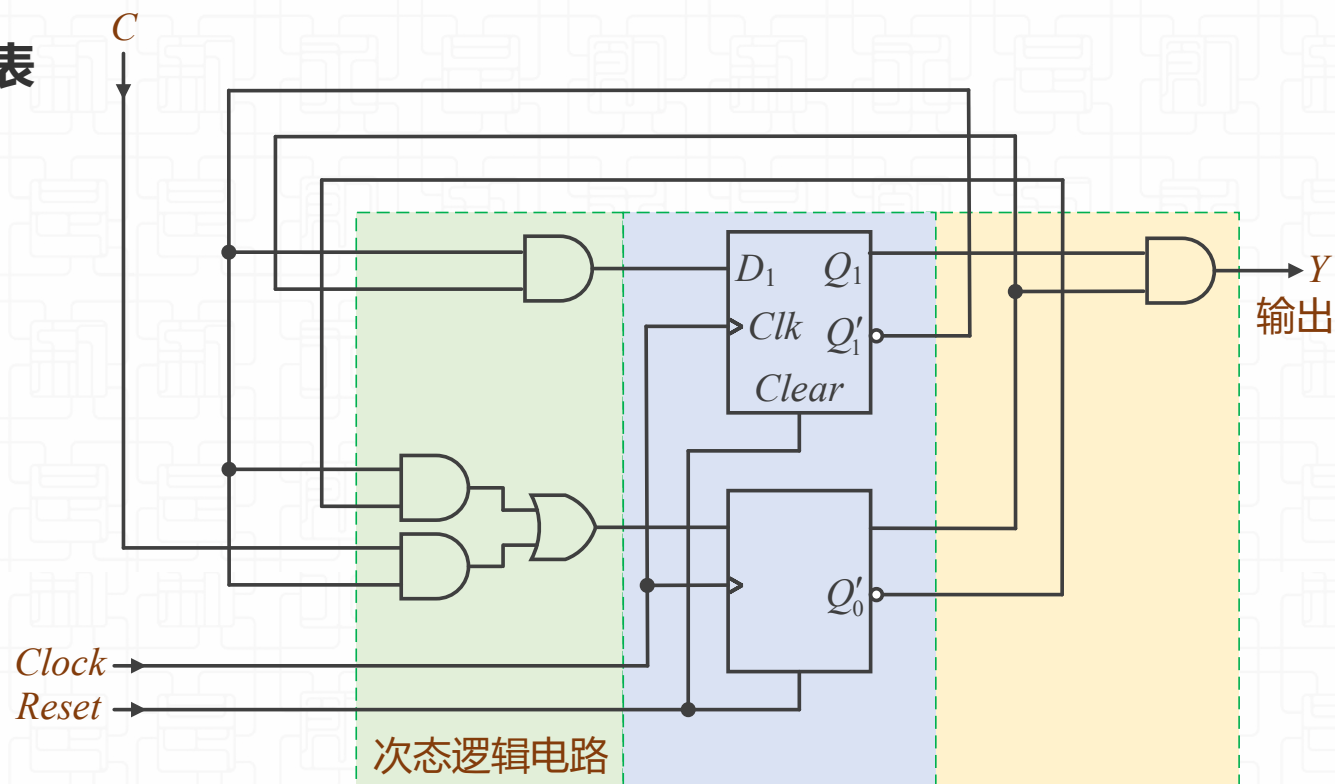
- 依据次态电路，写出次态电路方程和真值表
- 依据**输出电路**，写出**输出方程和真值表**
- 依据第一步和第二步，画出状态图

真值表

Q_1Q_0	Y
00	0
01	0
10	0
11	1

输出方程

$$Y = Q_1Q_0$$



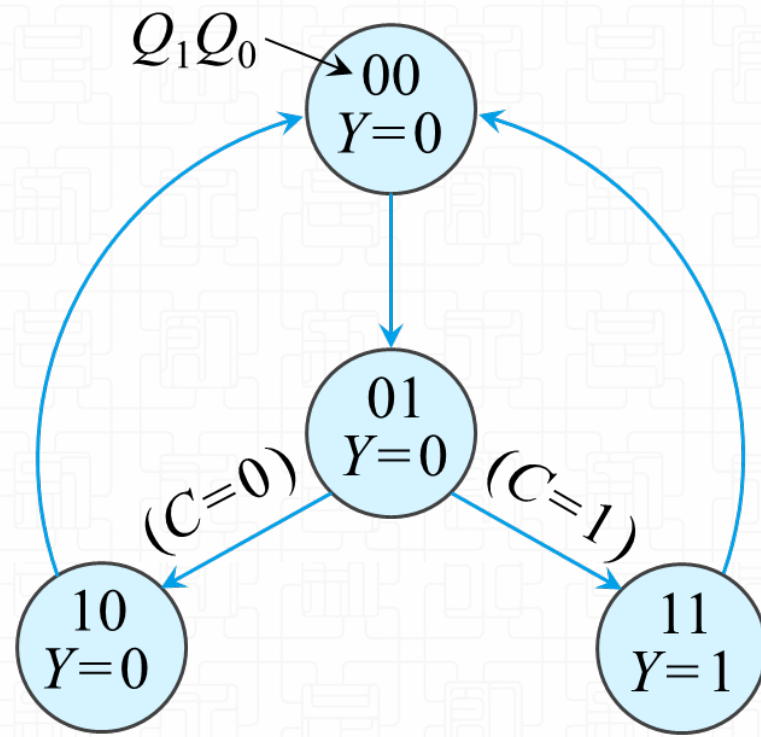
FSM的分析

❖ 有限状态机的分析：给定一个状态机电路，分析出该电路的状态图。

❖ 三个步骤：

- 依据次态电路，写出次态电路方程和真值表
- 依据输出电路，写出输出方程和真值表
- 依据第一步和第二步，画出状态图

Q_1Q_0	Y
00	0
01	0
10	0
11	1



FSM的综合

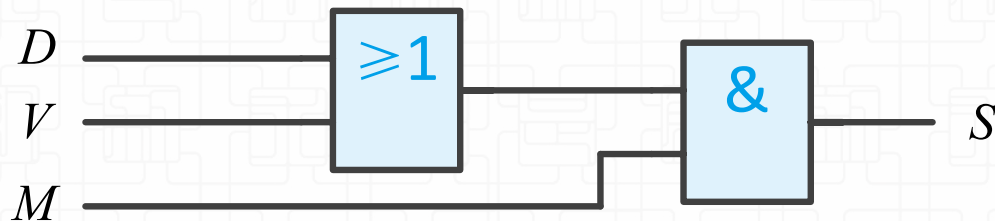
❖ 有限状态机的综合是指，给定一个状态图，综合出该状态图的电路

❖ 三个步骤：

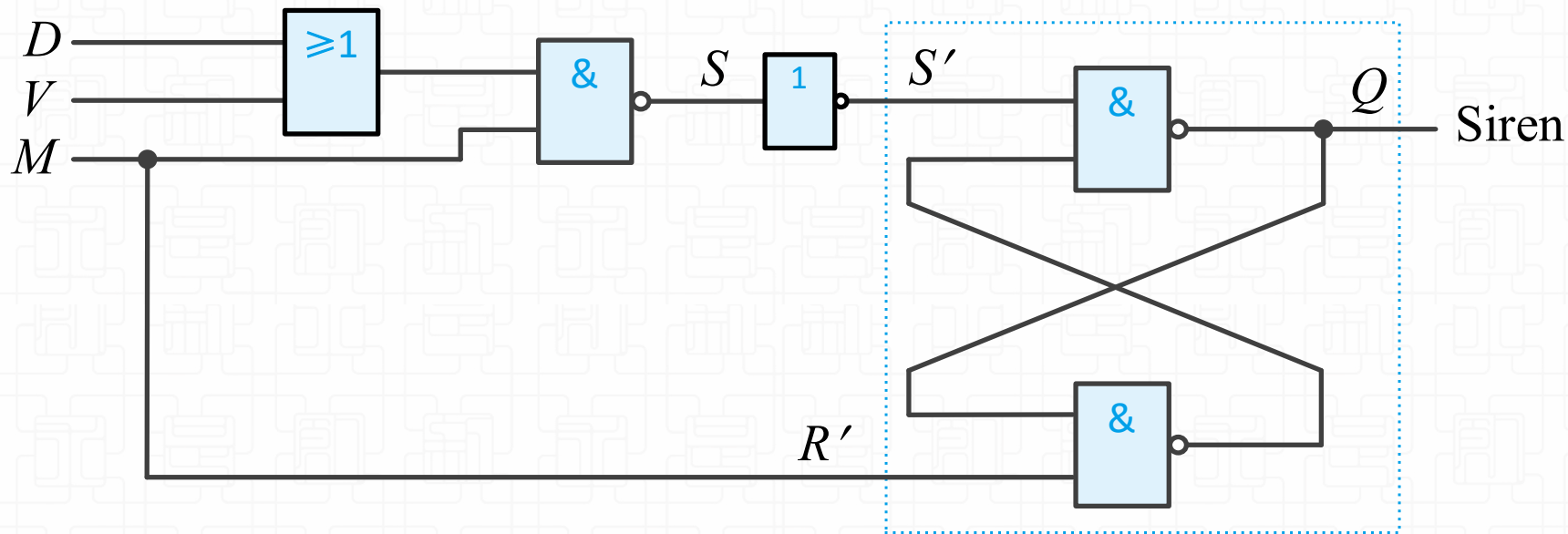
- 依据状态图，做出次态电路真值表，写出次态电路逻辑表达式
- 依据状态图，做出输出电路真值表，写出输出电路逻辑表达式
- 依据第一步和第二步，画出电路图

汽车报警器V1和V2版本回顾

❖ V1版：组合逻辑电路



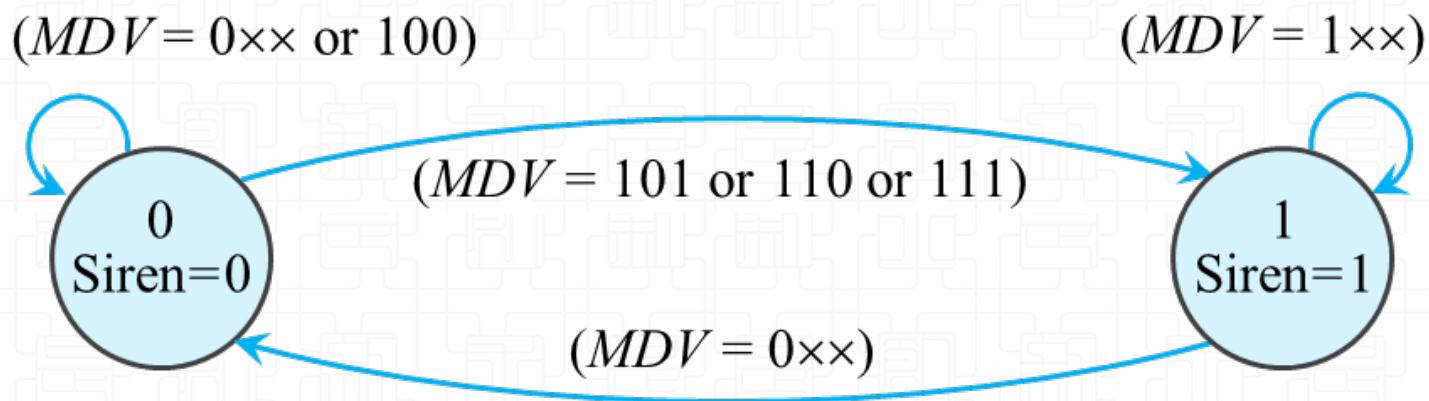
❖ V2版：在v1版的基础上，加入SR锁存器



汽车报警器v3版本：使用状态机

❖ 首先需要画出状态图

- 确定系统状态：汽车报警器的两个状态，一个是警报响，一个是警报不响。只在这两个状态机跳转，不需要额外状态。
- 确定有向边：两个状态之间都是可以互相跳转的，每个状态也是可以继续留在原先状态的，所以有4条有向边。
- 确定跳转条件：根据输入所有可能的组合引起的状态变化，在有向边上标注跳转条件。



汽车报警器v3版本：次态电路真值表和表达式

现态 Q_0	次态 Q_{0next}							
	M, D, V							
	000	001	010	011	100	101	110	111
0	0	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1

$$Q_{0next} = Q'_0 M(D + V) + Q_0 M$$



$$Q_{0next} = M(Q_0 + V + D)$$

$$Q_{0next}$$

$$= Q'_0 M(D + V) + Q_0 M$$

$$= (Q_0 M) + (Q'_0 M)(D + V)$$

应用 $x + yz = (x + y)(x + z)$

$$Q_{0next}$$

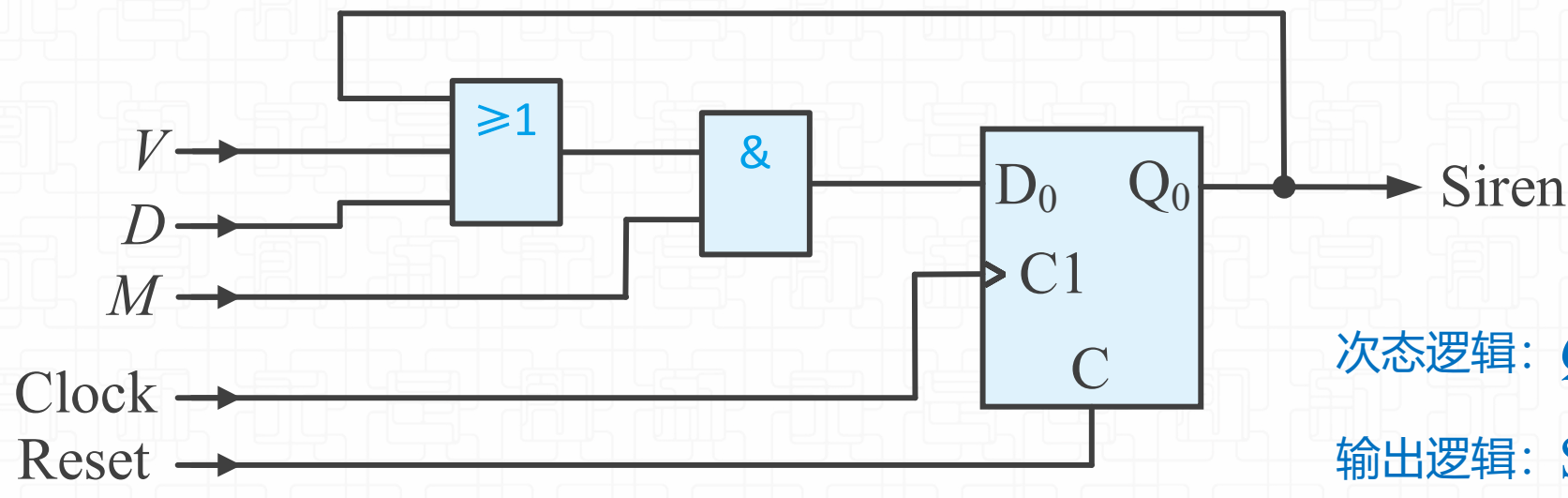
$$= (Q_0 M + Q'_0 M)(Q_0 M + D + V)$$

$$= M(Q_0 M + D + V)$$

应用 $x \cdot x = x$

$$Q_{0next} = M(Q_0 + D + V)$$

汽车报警器v3版本电路图



次态逻辑: $Q_{0next} = M(Q_0 + V + D)$

输出逻辑: $Siren = Q_{0next}$

- ❖ 版本2不需要时钟，版本3需要时钟。版本3提供了外部可控性。
- ❖ 版本2设计需要对系统的较为深入的理解和一些技巧，相比而言，版本3只要按照状态机的设计流程就可以完成，设计方法具有一般性。
- ❖ 考虑设计方法的一般性，以及随着复杂度增加后的设计难度，推荐使用版本3的设计方法。

The End.



中国大学MOOC

章献民

zhangxm@zju.edu.cn