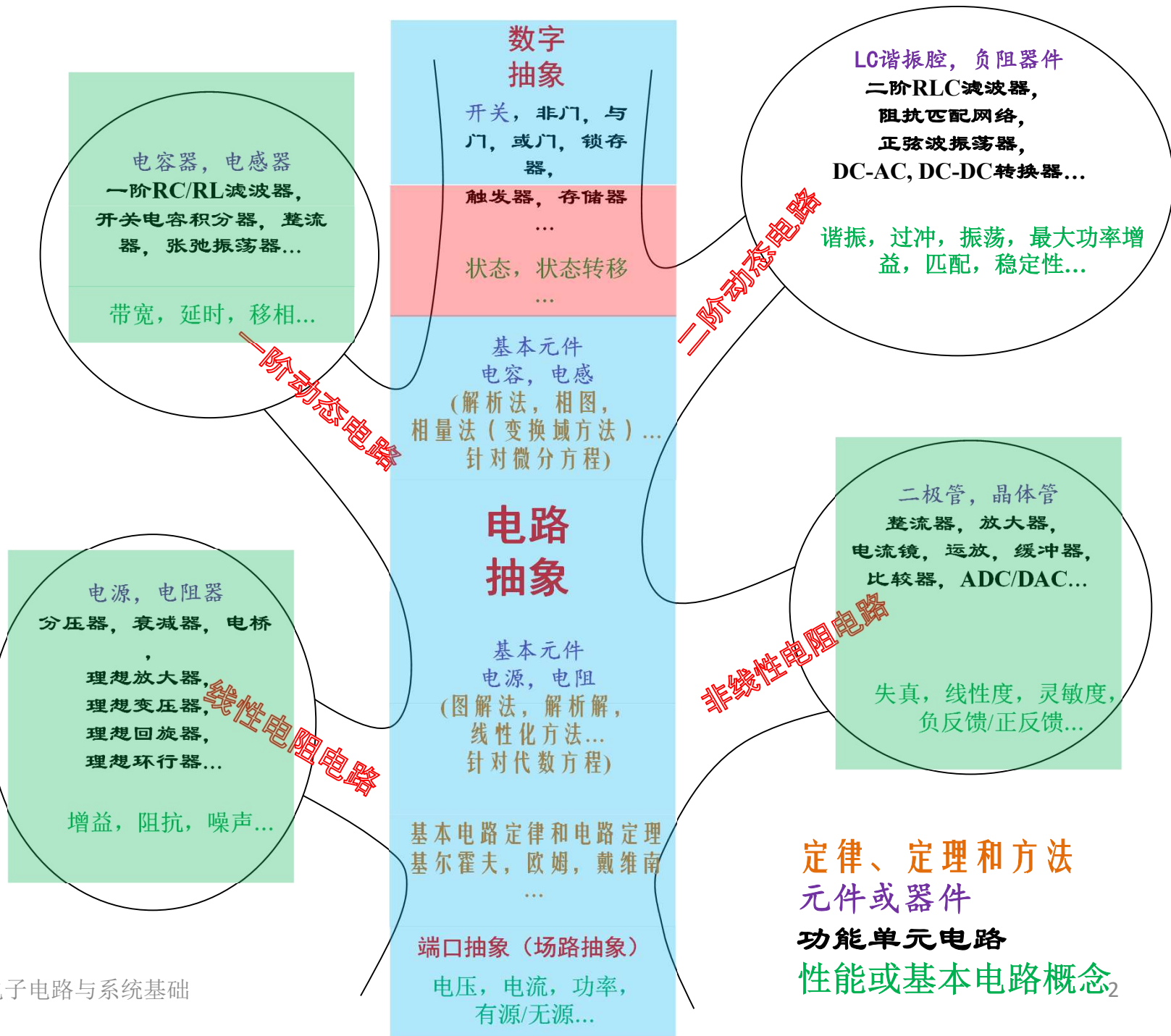


电子电路与系统基础II

理论课7讲 状态记忆单元
(时序逻辑电路中的锁存器、触发器、存储器)

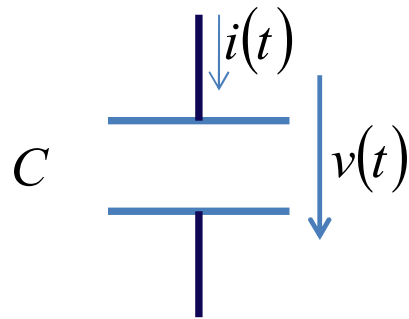
李国林
清华大学电子工程系

一条主干 四个分支



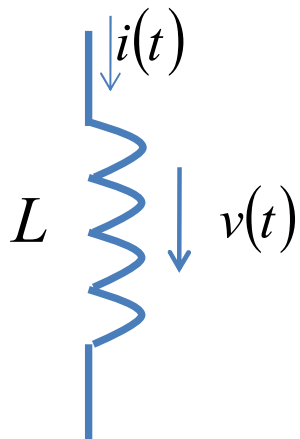
电容、电感具有记忆功能

- 电容的当前电压（状态变量）不仅和当前输入电流有关，还和之前输入电流、以及初始电压有关，它是有记忆的元件



$$v(t) = \frac{1}{C} \int_{-\infty}^t i(\tau) \cdot d\tau = V_0 + \frac{1}{C} \int_0^t i(\tau) \cdot d\tau$$

$$Q(t) = \int_{-\infty}^t i(\tau) \cdot d\tau = Q_0 + \int_0^t i(\tau) \cdot d\tau$$



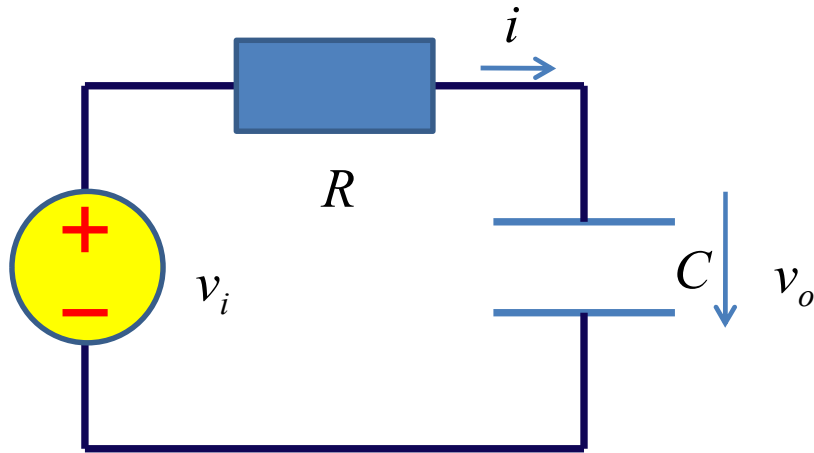
$$i(t) = \frac{1}{L} \int_{-\infty}^t v(\tau) \cdot d\tau = I_0 + \frac{1}{L} \int_0^t v(\tau) \cdot d\tau$$

$$\Phi(t) = \int_{-\infty}^t v(\tau) \cdot d\tau = \Phi_0 + \int_0^t v(\tau) \cdot d\tau$$

存储电荷、电能
状态变量：电荷、电压

存储磁通、磁能
状态变量：磁通、电流

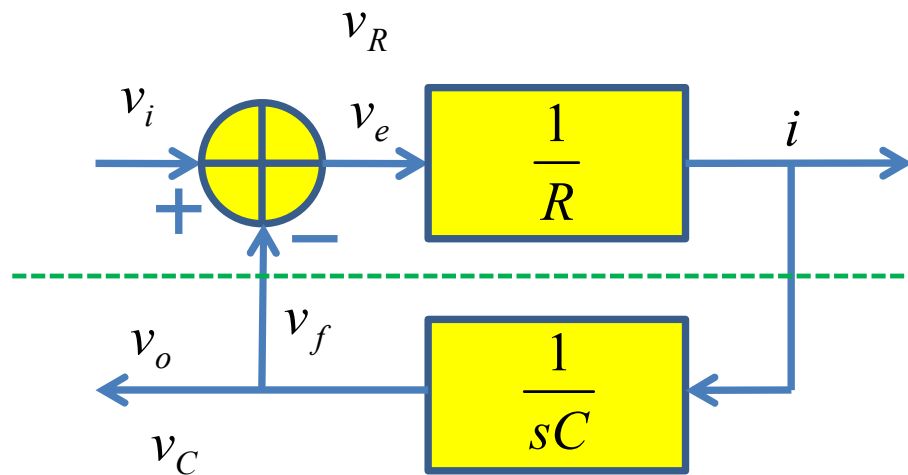
用反馈观点重新考察RC积分电路



$$\frac{V_o(s)}{V_i(s)} = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} = \frac{1}{sRC + 1} = \frac{\omega_0}{s + \omega_0}$$

$$s = j\omega$$

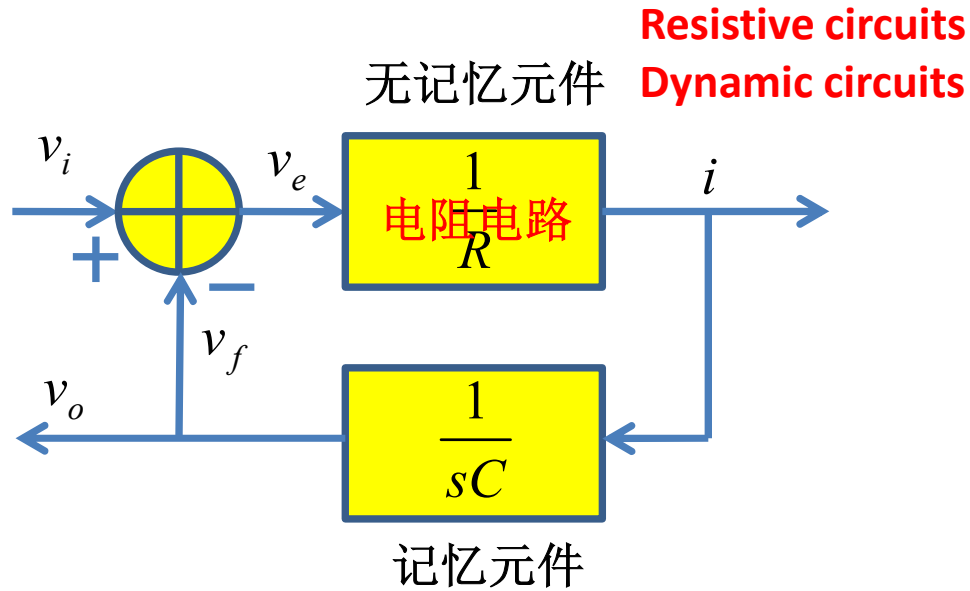
电阻电路处理的信号被电容存储下来，作为内在的激励源用于下一时刻的即时处理，从而形成动态行为：由于存储（记忆），而有状态转移，有时间效应，有频率效应



$$G_{m0} = \frac{1}{R} \quad R_F = \frac{1}{sC} \quad T = G_{m0}R_F = \frac{1}{sRC}$$

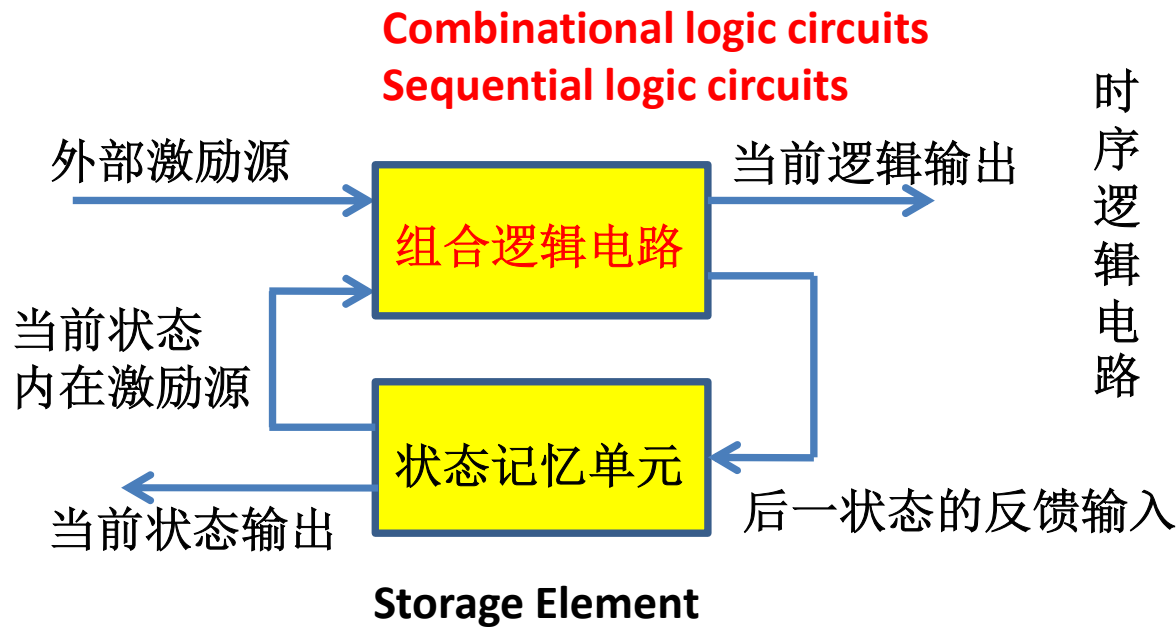
$$G_{mF} = \frac{G_{m0}}{1 + G_{m0}R_F} = \frac{\frac{1}{R}}{1 + \frac{1}{sRC}} = \frac{sC}{1 + sRC}$$

数字电路中的记忆单元



动态电路

电阻电路处理的信号被电容、电感等记忆元件存储下来，用于下一时刻的电阻即时处理，从而形成动态电路：由于存储（记忆），而有状态转移，有延时效应，有频率效应



时序逻辑电路

组合逻辑电路处理的逻辑信号被状态记忆单元存储下来，用于下一时刻的逻辑处理，从而形成时序逻辑：由于存储记忆，而有状态转移，有时序效应

有记忆才能形成状态及状态转移

- 电阻电路以记忆元件为反馈网络则构成动态电路
 - 电阻电路输出仅和当前输入有关
 - 动态电路输出不仅和当前输入有关，还和以前输入或当前状态有关
 - 记忆元件包括电容、电感、传输线、...
- 组合逻辑电路以状态记忆单元为反馈网络则构成时序逻辑电路
 - 组合逻辑输出仅和当前输入逻辑有关
 - 时序逻辑电路输出不仅和当前输入逻辑有关，还和以前输入或当前状态有关
 - 状态记忆单元包括触发器、寄存器、存储器、...

状态记忆单元 大纲

- 状态记忆单元
 - 双稳态
 - 计数器设计例
 - 单稳态
 - 无稳态
- 数字系统综合的层次

一、分类

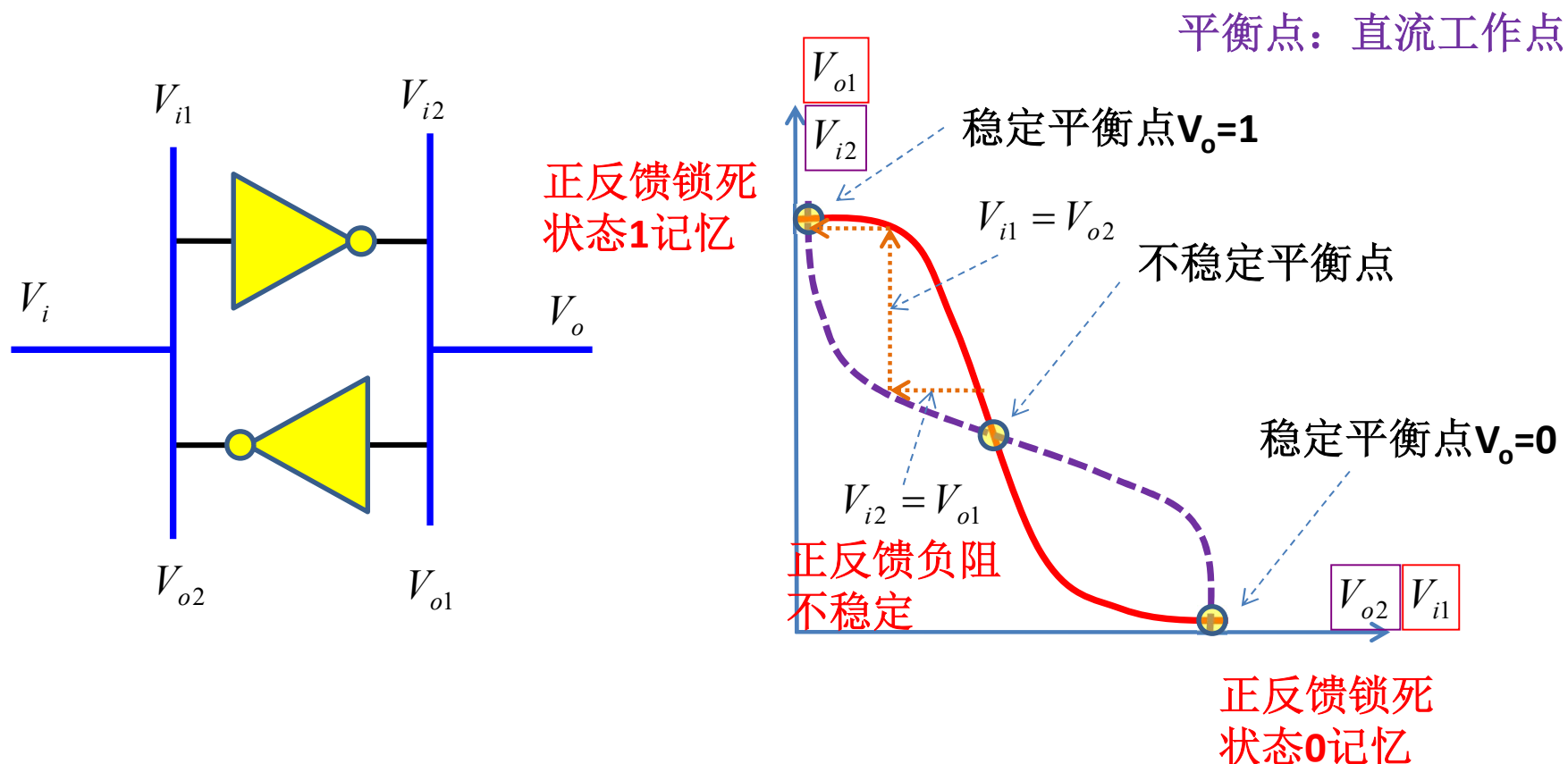
- 时序逻辑电路中的状态记忆单元，可分三类
 - 双稳态：**bistable**
 - 这类记忆单元具有两个稳定状态，用于记忆状态**0**和状态**1**
 - 双稳态记忆单元是计算机系统中最常见的状态记忆单元
 - 单稳态：**monostable**
 - 这类记忆单元只有一个稳定状态，如果在强干扰下偏离了稳定状态，经过一定时间后，该类记忆单元会返回到这个稳定状态
 - 单稳态记忆单元可用来实现定时、振荡等功能
 - 无稳态：**astable**
 - 这类记忆单元没有可以保持的稳定状态，因而只能在状态间来回转换，从而形成振荡
 - 无稳态记忆单元可用来实现振荡功能

二、双稳态记忆单元

- 基本双稳态单元
 - SR锁存器
 - D锁存器
 - 基于SR锁存器
 - 基于多路选择器
 - D触发器
 - 存储器

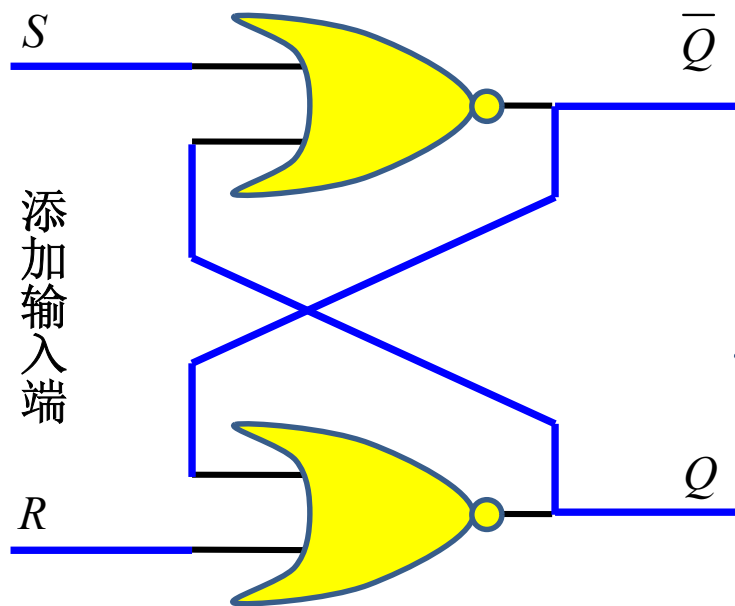
2.1 基本双稳态单元

- 我们采用正反馈建立双稳态电路，两个稳定状态分别代表**0**和**1**

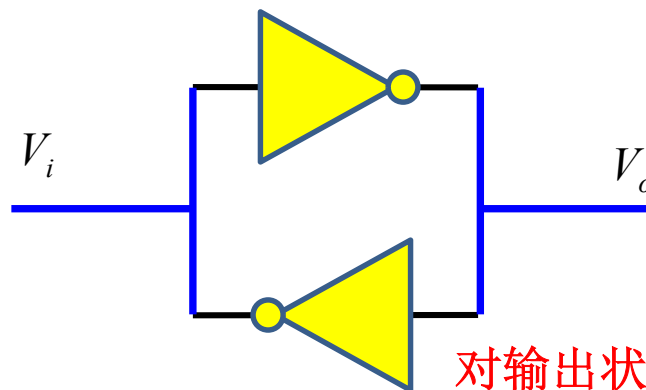


2.2 SR锁存器

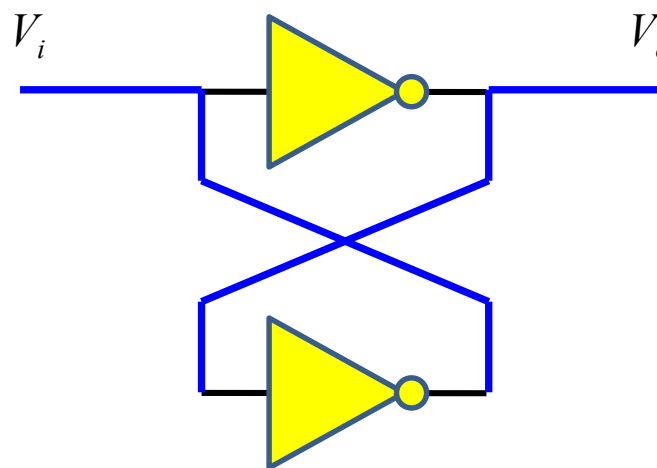
- SR Latch
 - S: Set, 输出Q置1: 置位
 - R: Reset, 输出Q清零: 复位



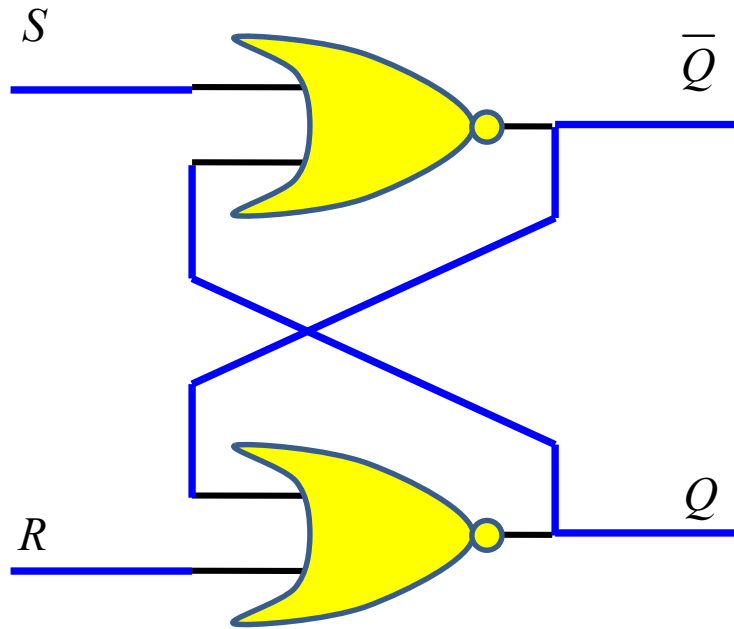
NOR based SR Latch



对输出状态直接进行操作，对N型负阻进行操作，激励源本身需要较大的驱动能力，改变记忆状态不方便



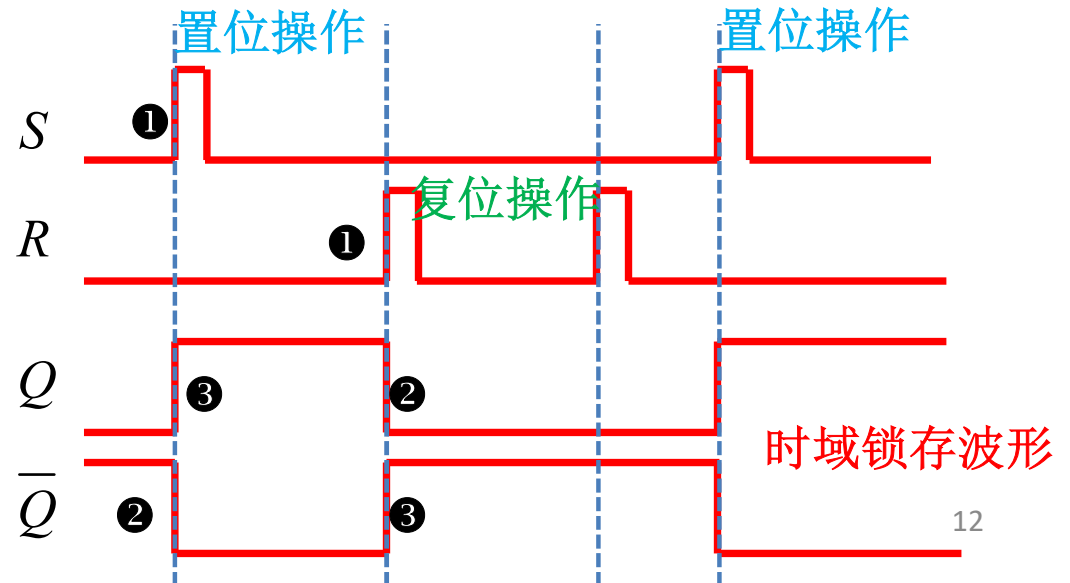
S置1, R清0

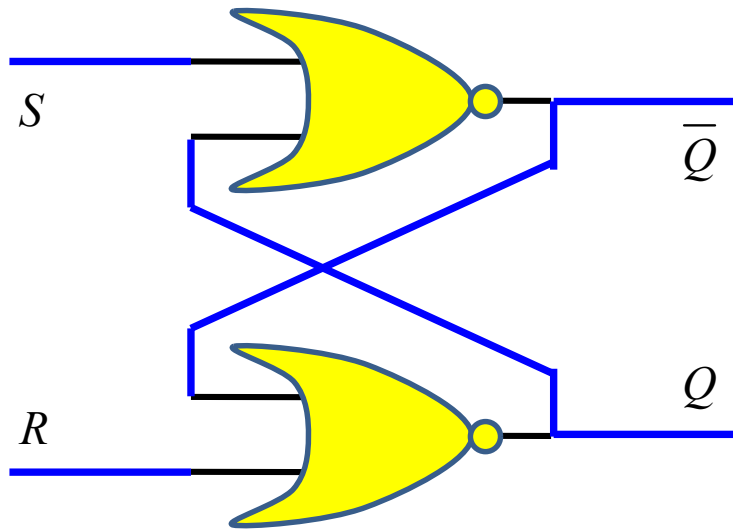


S	R	Q	\bar{Q}	说明
0	0	Q	\bar{Q}	状态保持不变
1	0	1	0	S置1, 之后, S可安全返回0状态
0	1	0	1	R清0, 之后, R可安全返回0状态
1	1	0	0	禁止这种情况出现



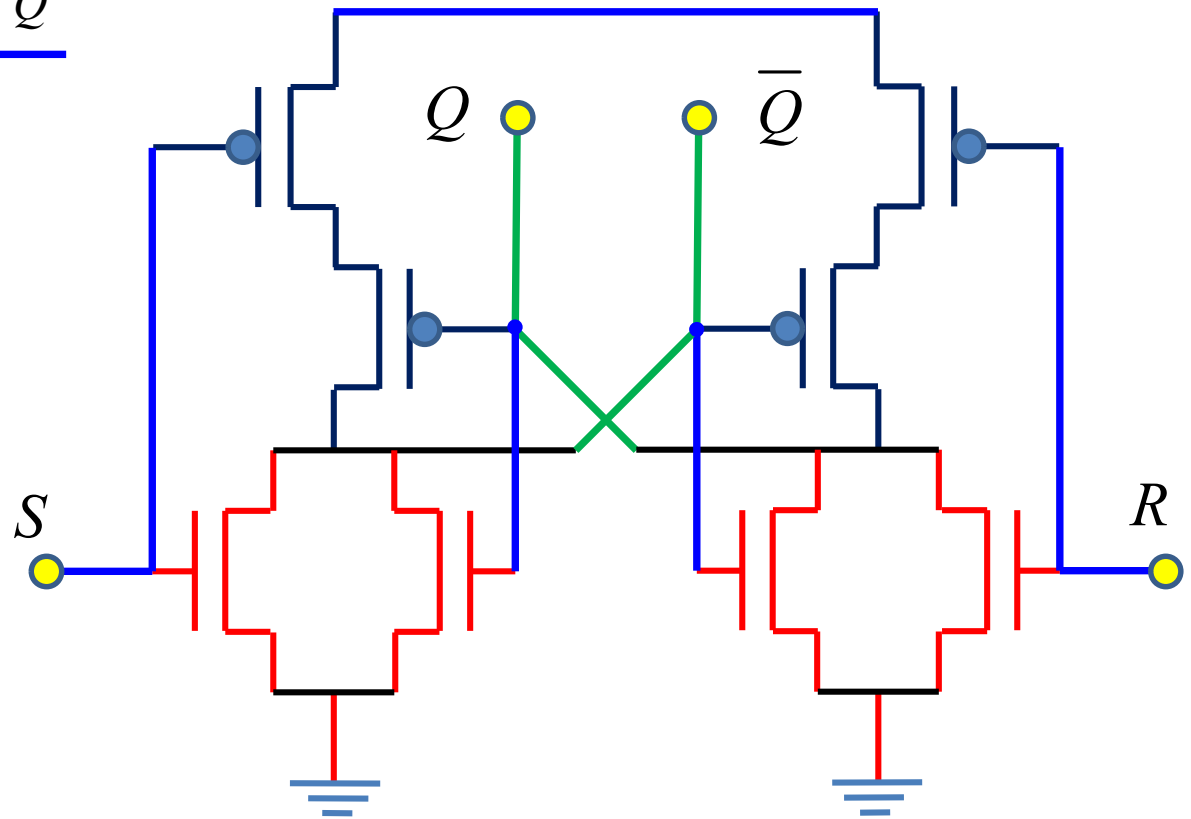
不考虑寄生电容效应, 门延时为零





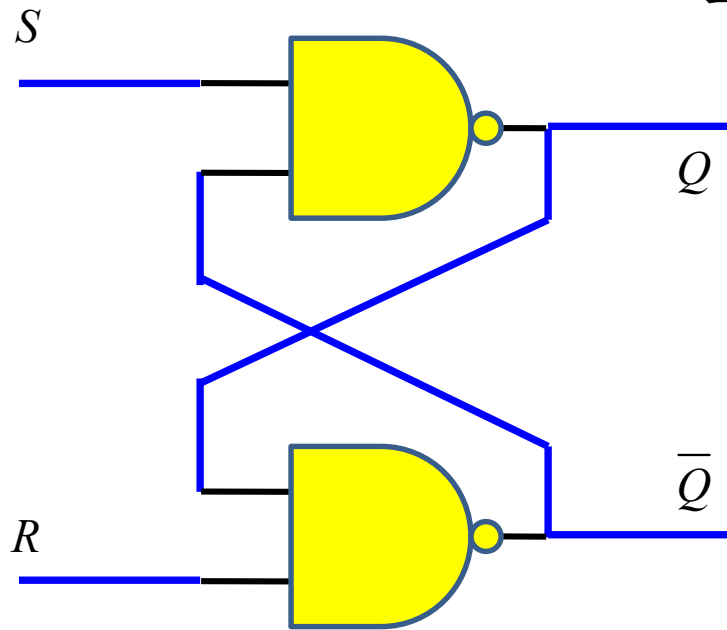
对应的门电路用CMOS门实现即可

CMOS 电路实现



基于与非门的SR锁存器

S置位，R复位



NAND based SR Latch

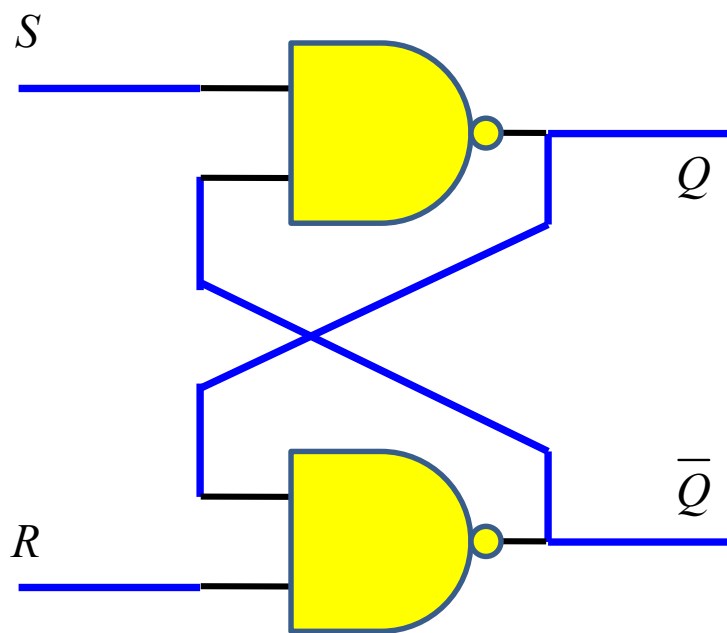


S	R	Q	\bar{Q}	说明
1	1	Q	\bar{Q}	状态保持不变
0	1	1	0	S置1, 之后, S可安全返回0状态
1	0	0	1	R清0, 之后, R可安全返回0状态
0	0	1	1	禁止这种情况出现

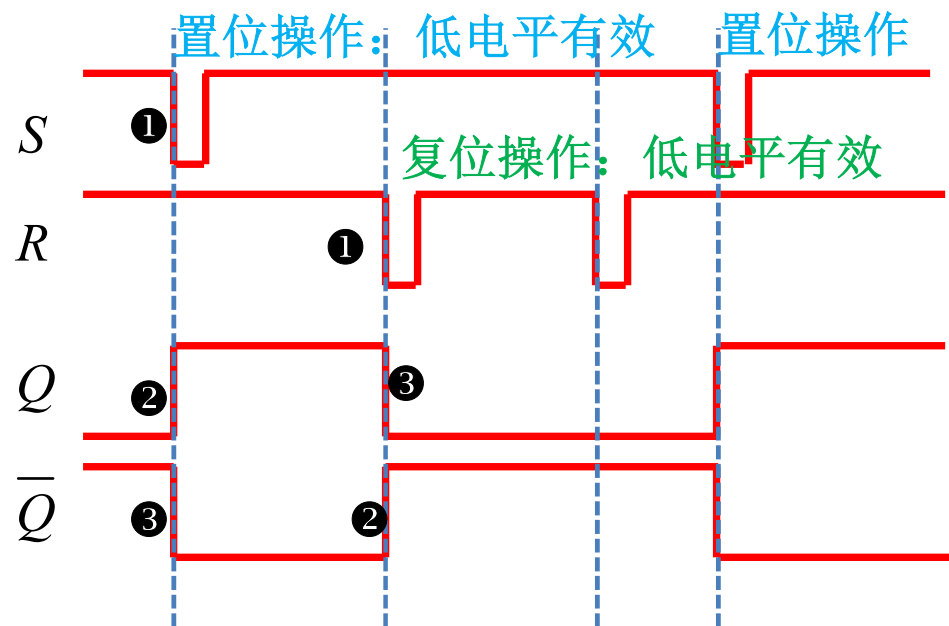
基于与非门的SR锁存器对有效的低电平作出反应，常态是高电平，低电平触发：图中两个圆圈表示低电平触发有效

同样存在禁止态

时域锁存波形



S	R	Q	\bar{Q}
1	1	Q	\bar{Q}
0	1	1	0
1	0	0	1
0	0	1	1



不考虑寄生电容效应，门延时为零

2.3 D锁存器 D Latch

- D触发器是数字电路中最常见的触发器记忆单元，我们首先从D锁存器入手

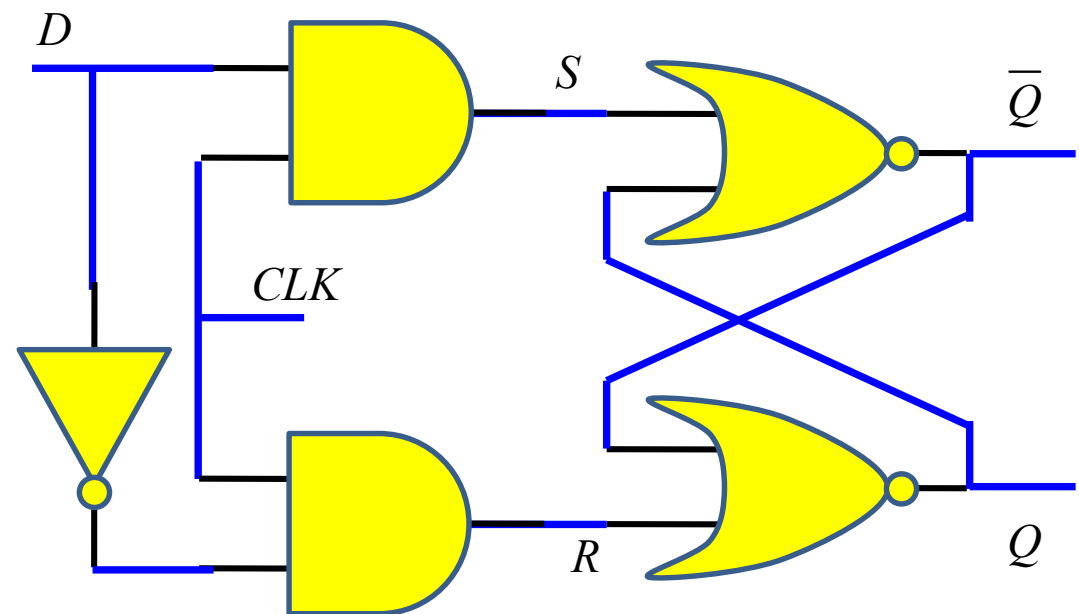


透明传输的D Latch

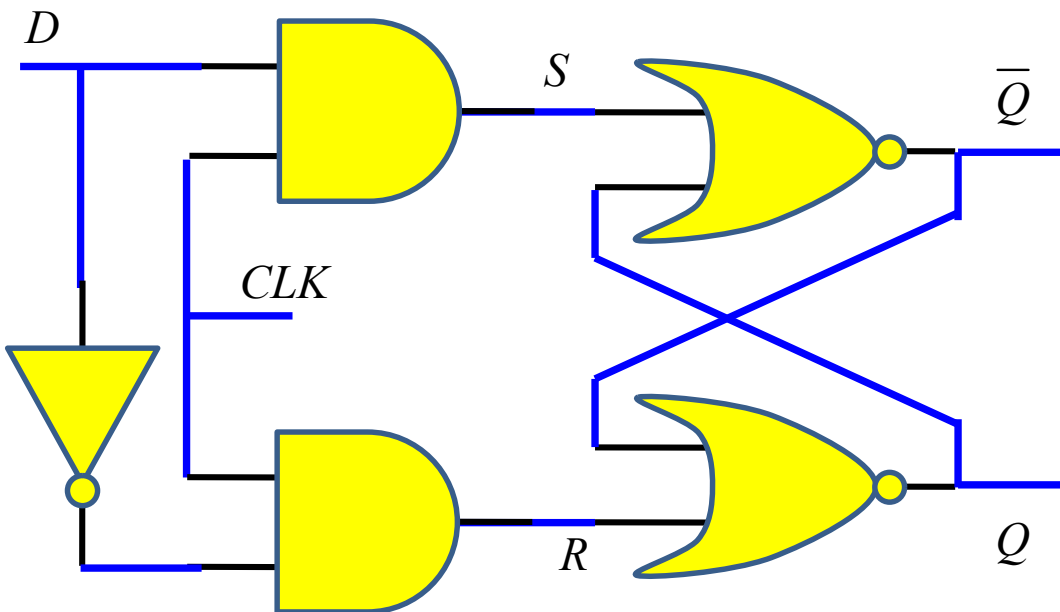
当CLK逻辑1时:

D	Q
0	0
1	1

D:Delay延时一个时钟周期传输过去
Data, 用于数据记忆与传输



时域锁存波形



$CLK = 1$

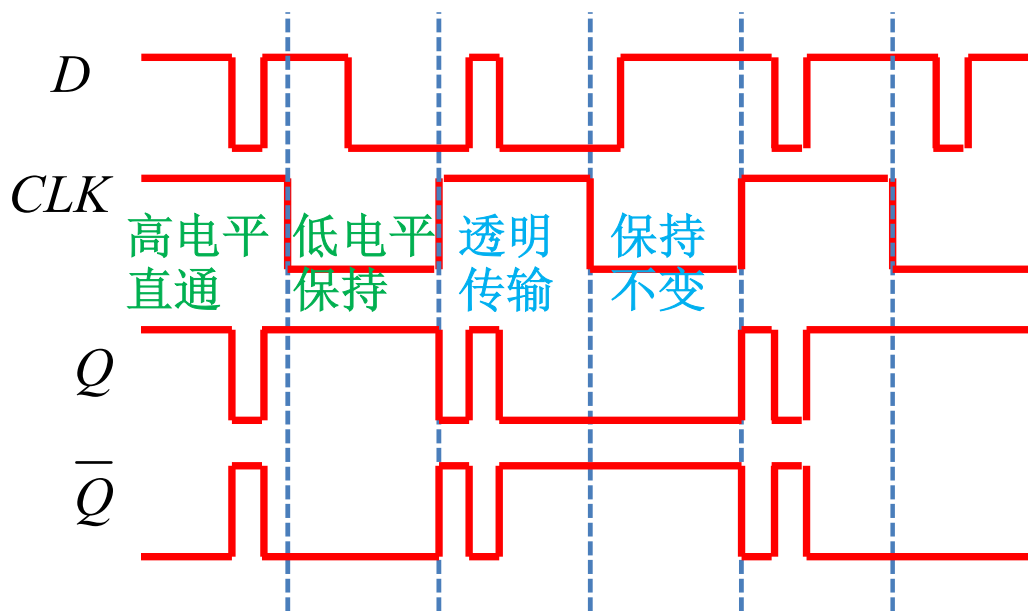
D	Q	\bar{Q}
0	0	1
1	1	0

CLK=1: 透明传输

$CLK = 0$

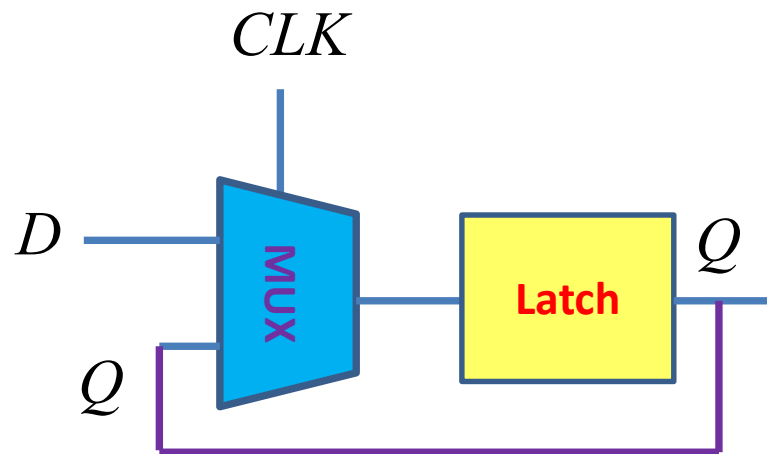
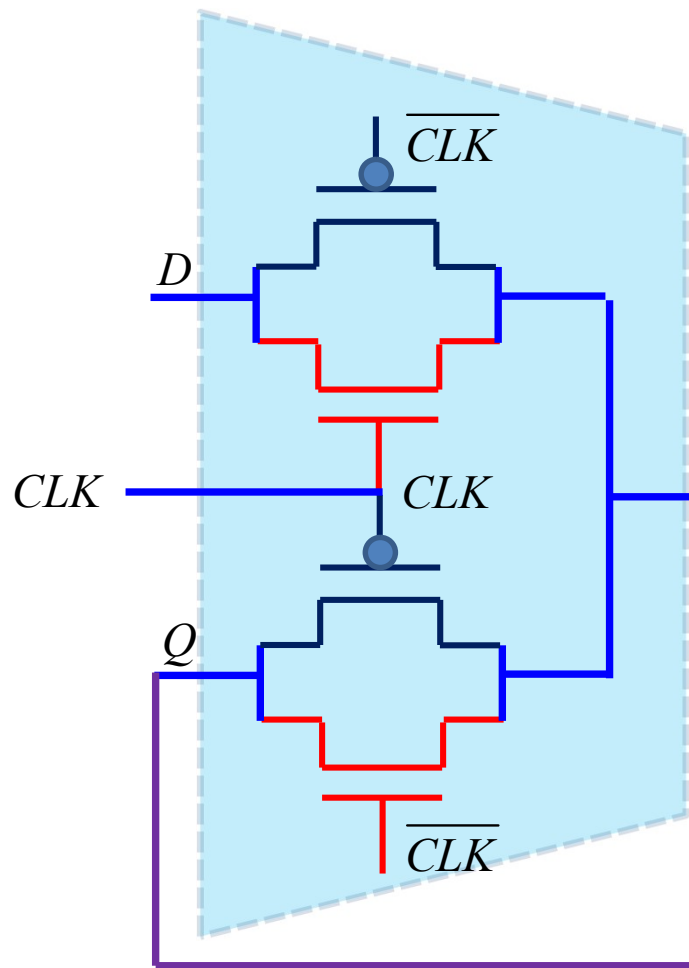
D	Q	\bar{Q}
0	Q	\bar{Q}
1	Q	\bar{Q}

CLK=0: 保持不变



假设门延时极小，
可以忽略不计

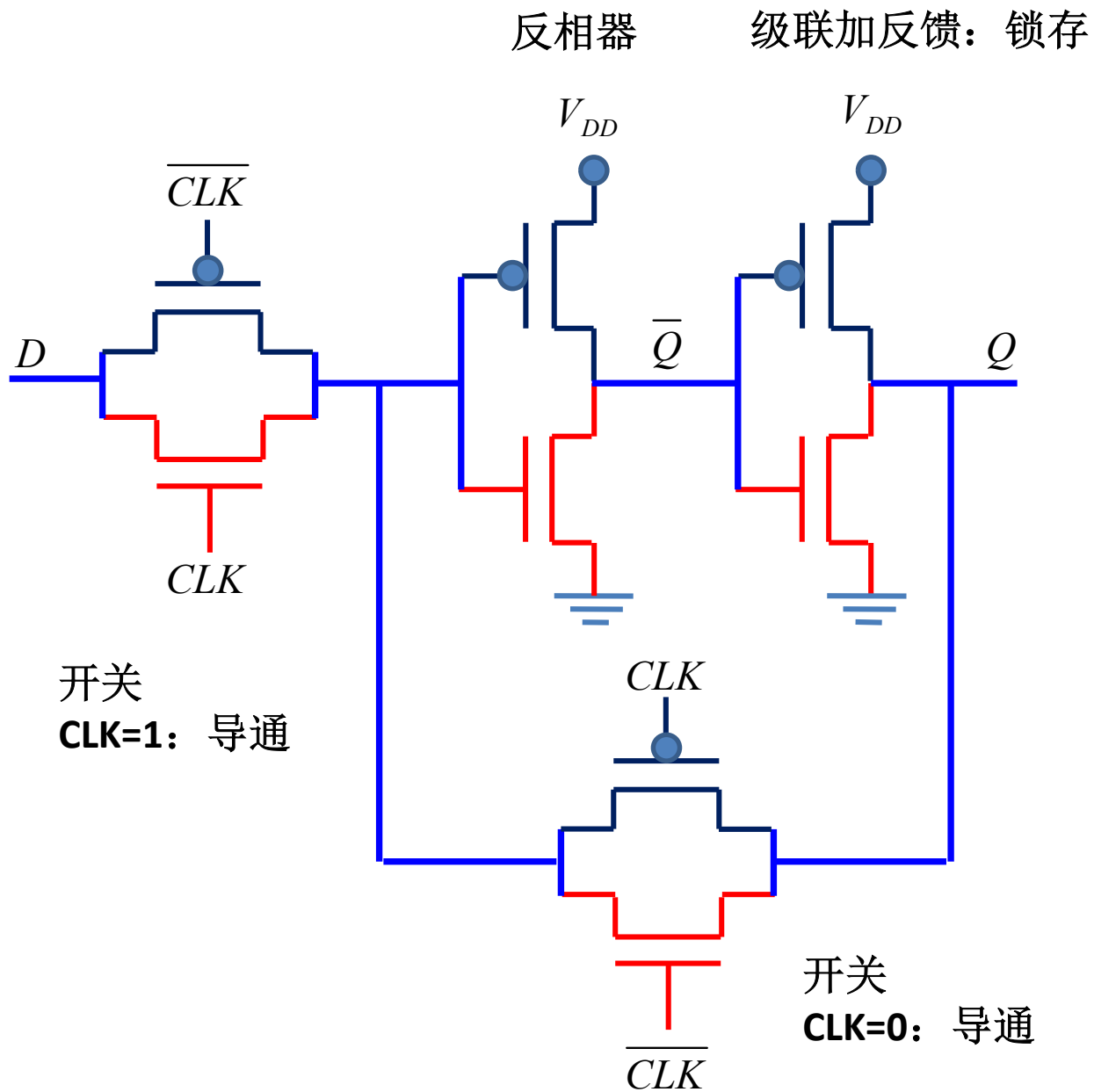
基于多路选择的实现方案



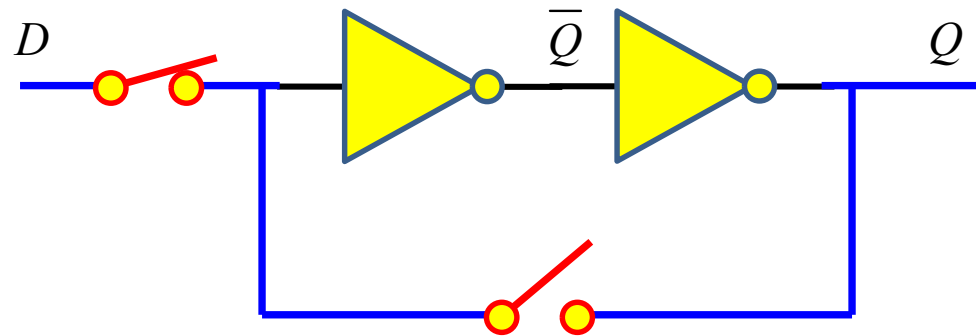
CLK为1，则选通D通过RS透明锁存器
CLK为0，则选通Q通过：保持原状态

晶体管个数：
 $2+2+2+2=8!$

八管方案



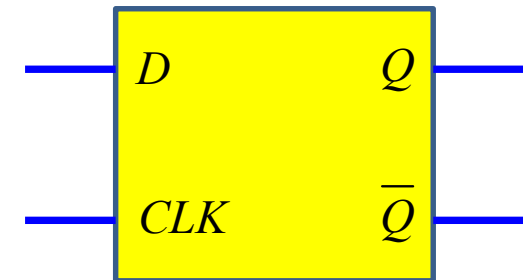
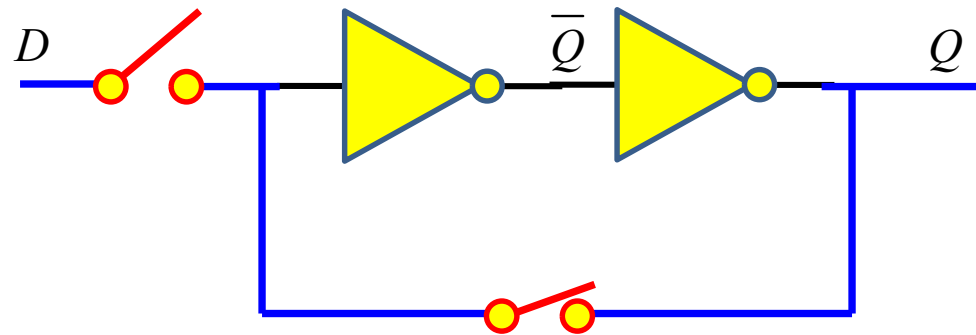
工作原理



CLK=1

$Q = D$

透明传输



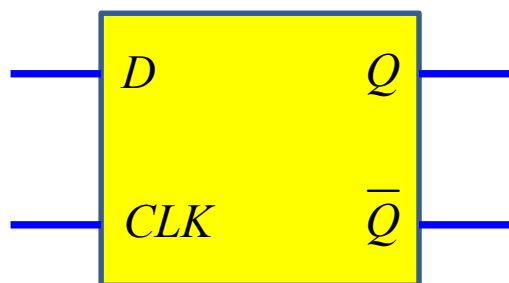
D Latch

CLK=0

Q hold

保持

2.4 D触发器 D flip-flop

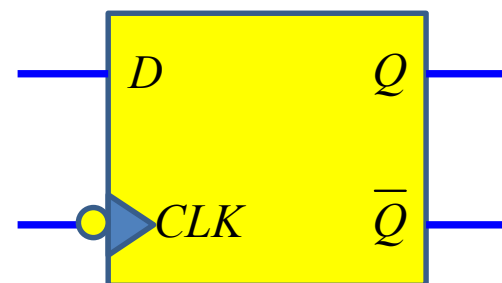


透明传输的D锁存器

输入到输出是透明的
transparent

输入变化，最多两个门延时后，则可反应到输出端：如果不考虑门延时，输出即时对输入的变化作出反应

锁存器latch和触发器flip-flop的区别就在于输入到输出是否透明



边沿触发的D触发器

CLK线上的三角：表示边沿触发

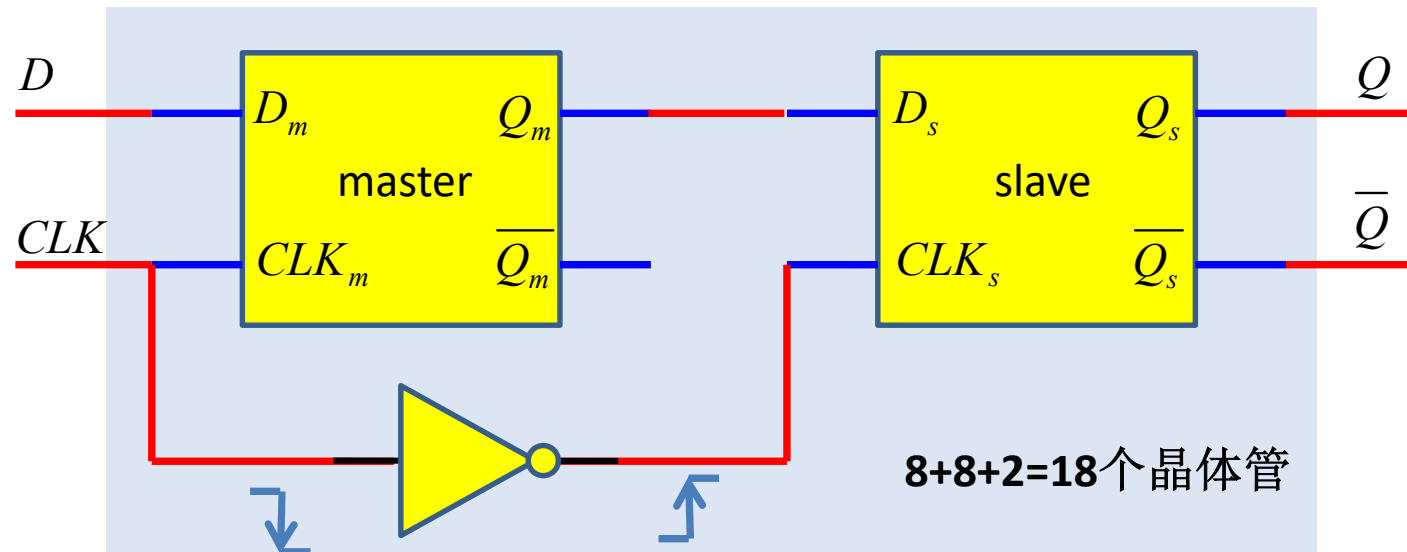
CLK线上的圆圈：表示下降沿触发

相当于在特定时间点（时钟下降沿）对数据采样

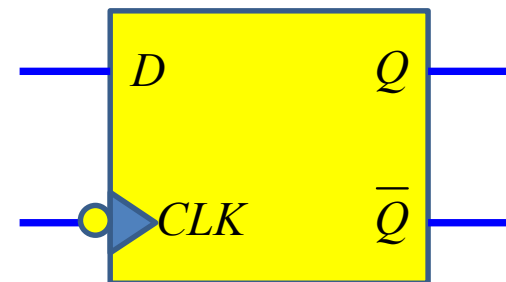
Q_n	D	CLK	Q_{n+1}
×	0		0
×	1		1
×	×		Q_n

仅在时钟下降沿对数据采样并传输，其他时间都保持

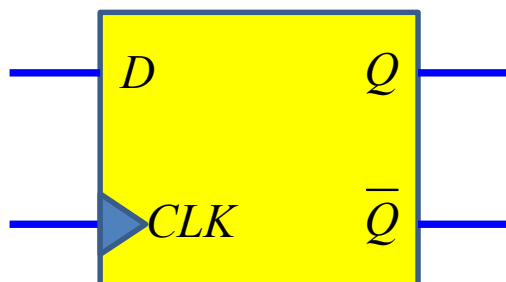
主从结构的D触发器



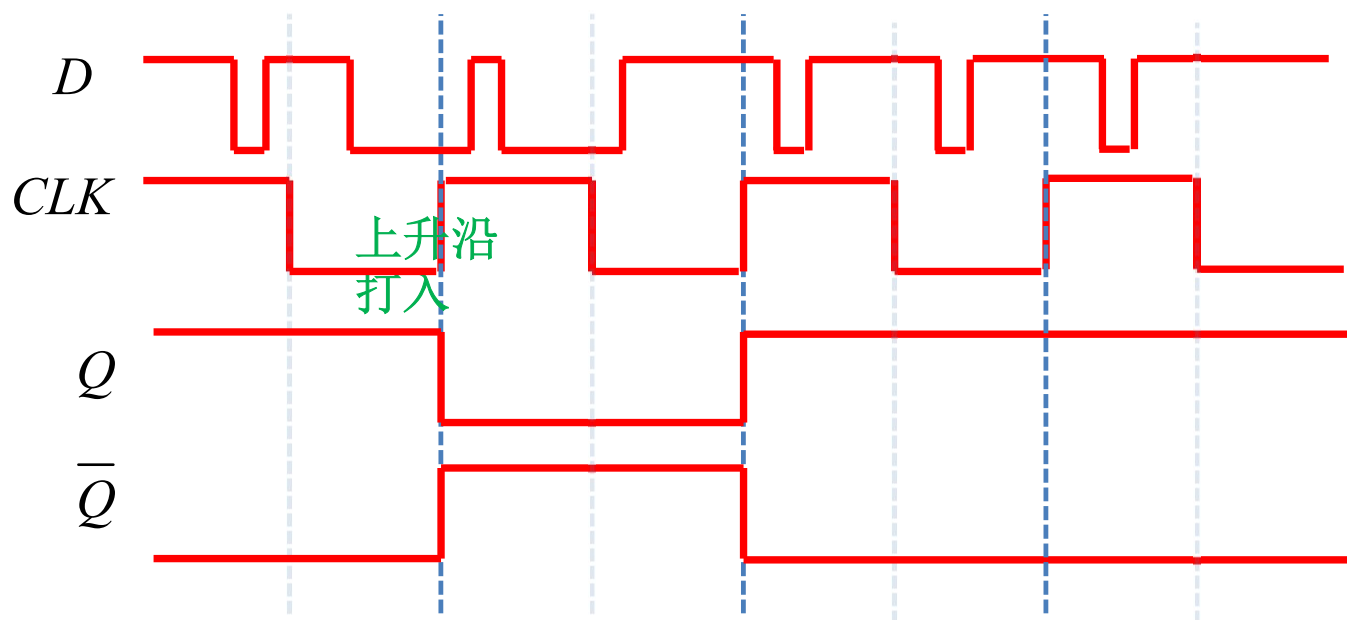
主从结构：两个相同的锁存器级联，但它们的时钟反相：时钟为1时，主锁存器透明传输，从锁存器保持不变；时钟变化为0后，主锁存器保持不变，从锁存器透明传输；显然从锁存器的输出是在时钟下降沿的时候将数据打入到从锁存器并输出



时域触发波形



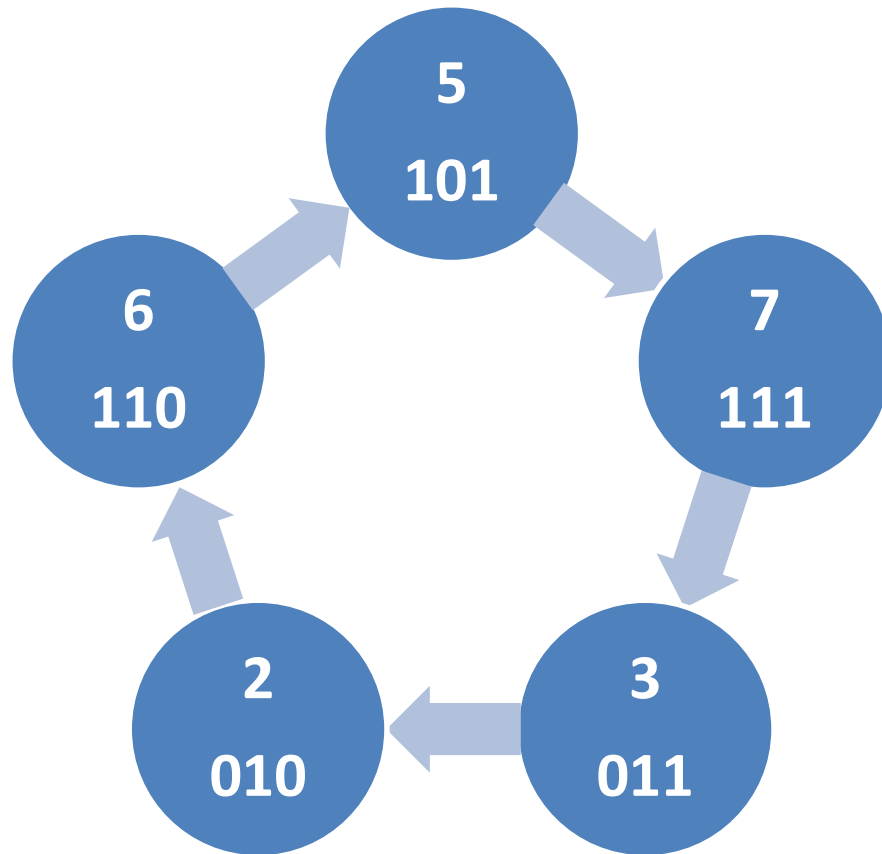
Q_n	D	CLK	Q_{n+1}
×	×		D
×	×		Q_n



例：用D触发器设计计数器

- D触发器是数字电路系统应用最广泛的触发器
- 计数器是时序电路的典型例子：我们以**3bit**计数器的D触发器实现为例，说明时序电路设计的简单流程
- 例：请用D触发器设计一个**3bit**计数器，该计数器在时钟驱动下，可以依次循环输出**5, 7, 3, 2, 6**

状态转移图



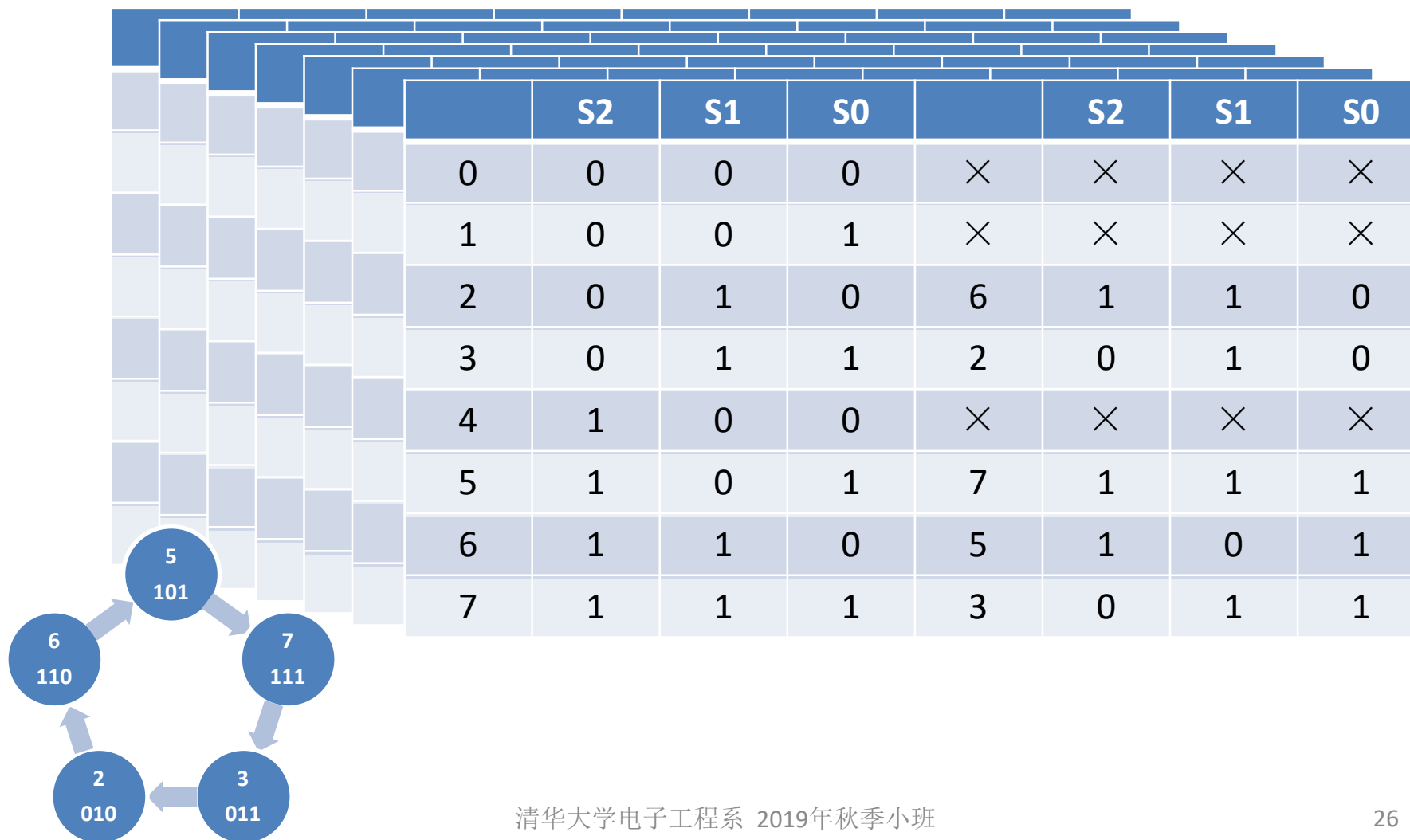
时序逻辑电路
离散时间离散（有限）状态转移图
Finite-State Machine: 有限状态机

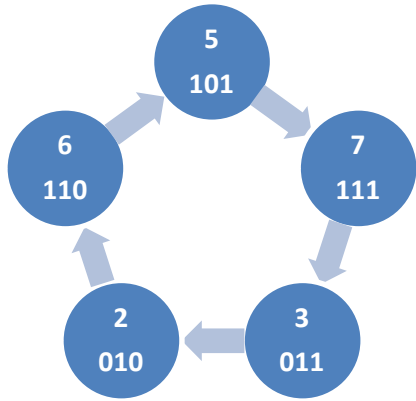
动态模拟电路
连续时间连续状态转移图
phase portrait: 相图

当前状态与下一状态转移表

当前状态

下一状态

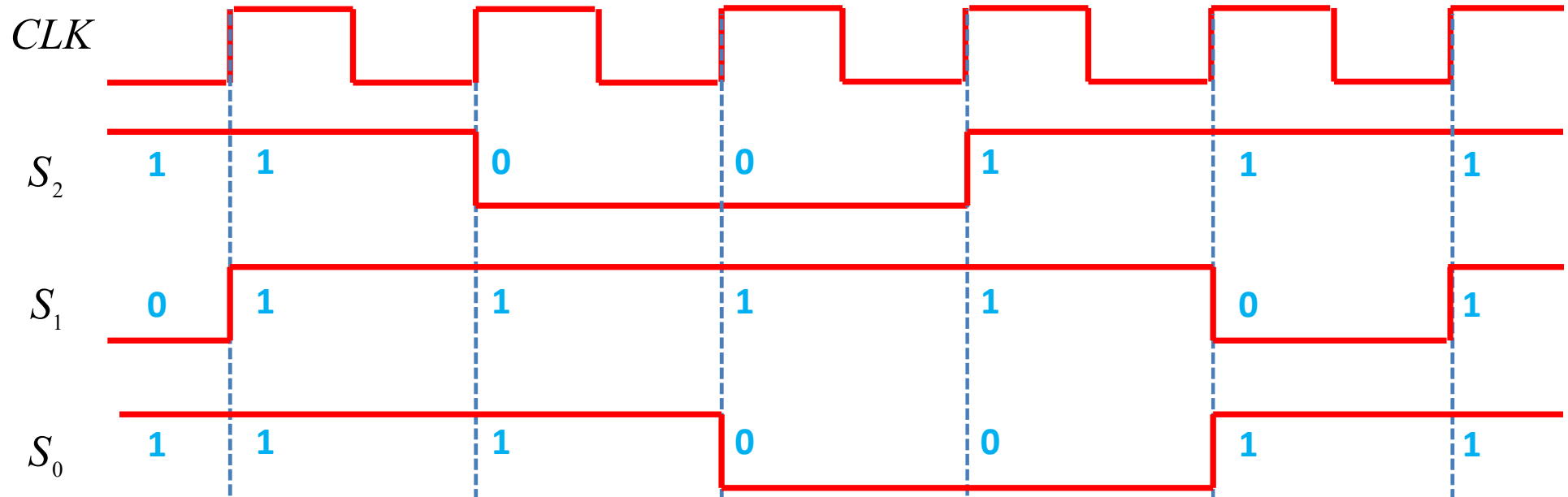




状态转换时域波形

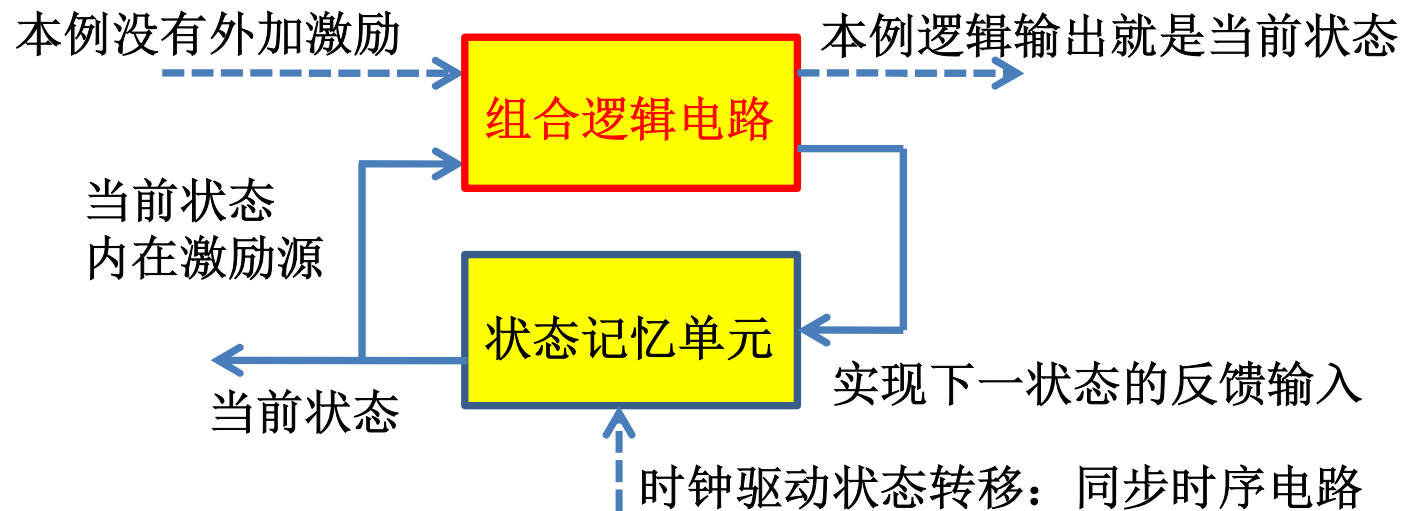
时钟上升沿来临，
进入下一状态

时钟上升沿来临，
进入下一状态



如何从现有状态转移到下一状态？

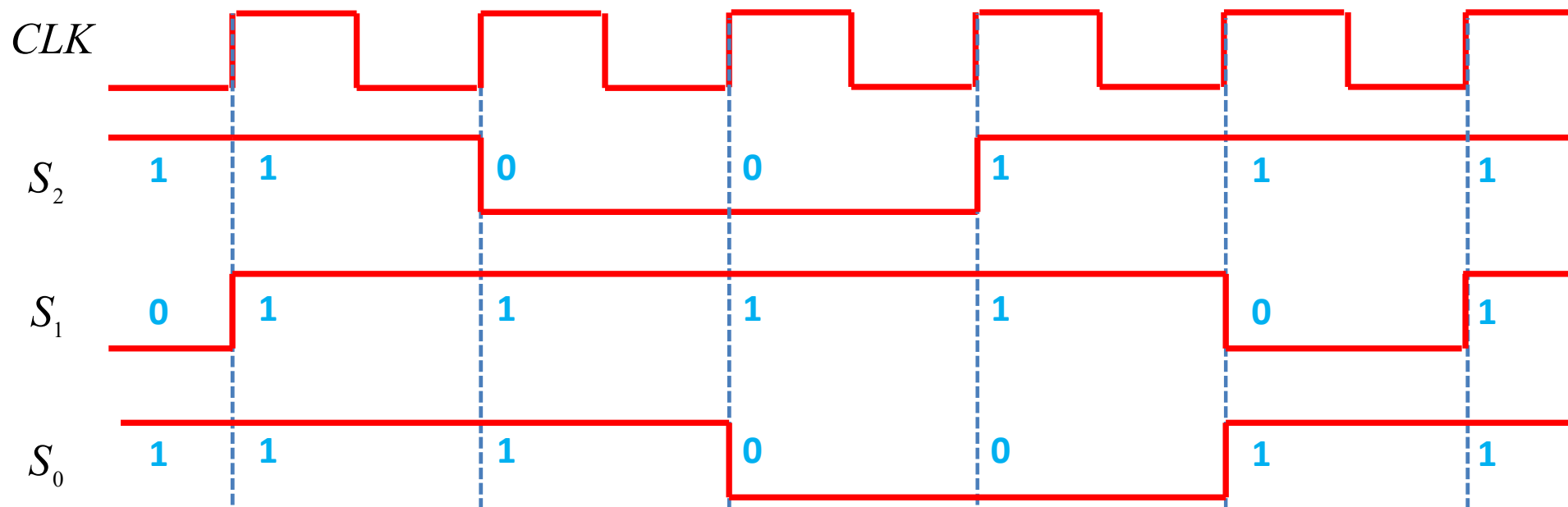
- 在时钟上升沿来临前，用组合逻辑电路根据当前状态计算出下一个状态，作为D触发器的D输入端
- 在时钟上升沿来临时，将触发器D输入数据打入触发器，记忆，传输，获得下一个状态



时钟上升沿来临时的触发器数据输入

	当前状态			下一状态				数据输入			
	S_2	S_1	S_0		S_2	S_1	S_0		D_2	D_1	D_0
0	0	0	0	×	×	×	×	×	×	×	×
1	0	0	1	×	×	×	×	×	×	×	×
2	0	1	0	6	1	1	0	6	1	1	0
3	0	1	1	2	0	1	0	2	0	1	0
4	1	0	0	×	×	×	×	×	×	×	×
5	1	0	1	7	1	1	1	7	1	1	1
6	1	1	0	5	1	0	1	5	1	0	1
7	1	1	1	3	0	1	1	3	0	1	1

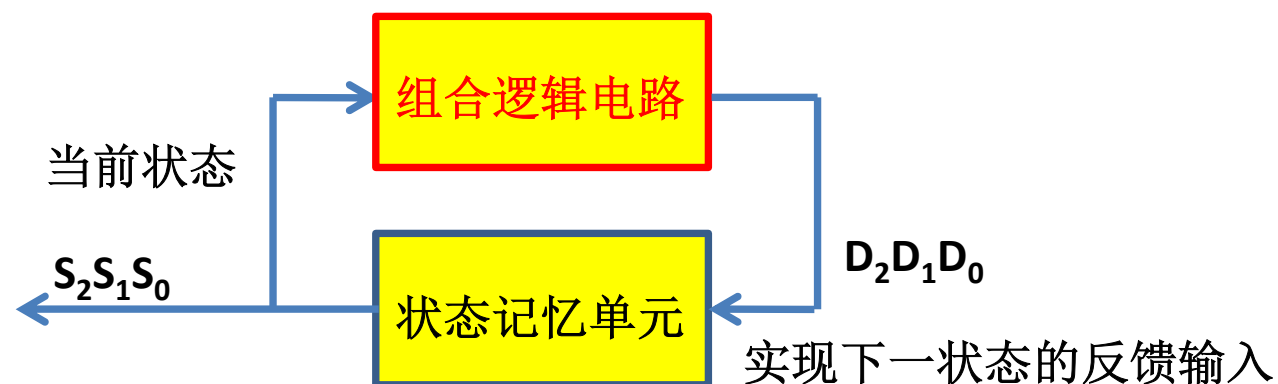
由于D触发器是简单的延时传输（简单存储器），因而在时钟上升沿来临前，触发器输入端准备好的数据输入正好就是下一个状态值
 如果采用其他类型的触发器，其数据输入并非下一状态值，则需特别设计



状态记忆单元：采用3个D触发器来实现5个状态的循环，这里没有其他输入

组合 逻辑 电路 设计

组合逻辑电路：其设计已经掌握



	S_2	S_1	S_0		D_2	D_1	D_0
0	0	0	0	×	×	×	×
1	0	0	1	×	×	×	×
2	0	1	0	6	1	1	0
3	0	1	1	2	0	1	0
4	1	0	0	×	×	×	×
5	1	0	1	7	1	1	1
6	1	1	0	5	1	0	1
7	1	1	1	3	0	1	1

D_2

S_2S_1	S_0	0	1
00		×	×
01		1	0
11		1	0
10		×	1

$$D_2 = \overline{S_0} + \overline{S_1}$$

D_1

S_2S_1	S_0	0	1
00		×	×
01		1	1
11		0	1
10		×	1

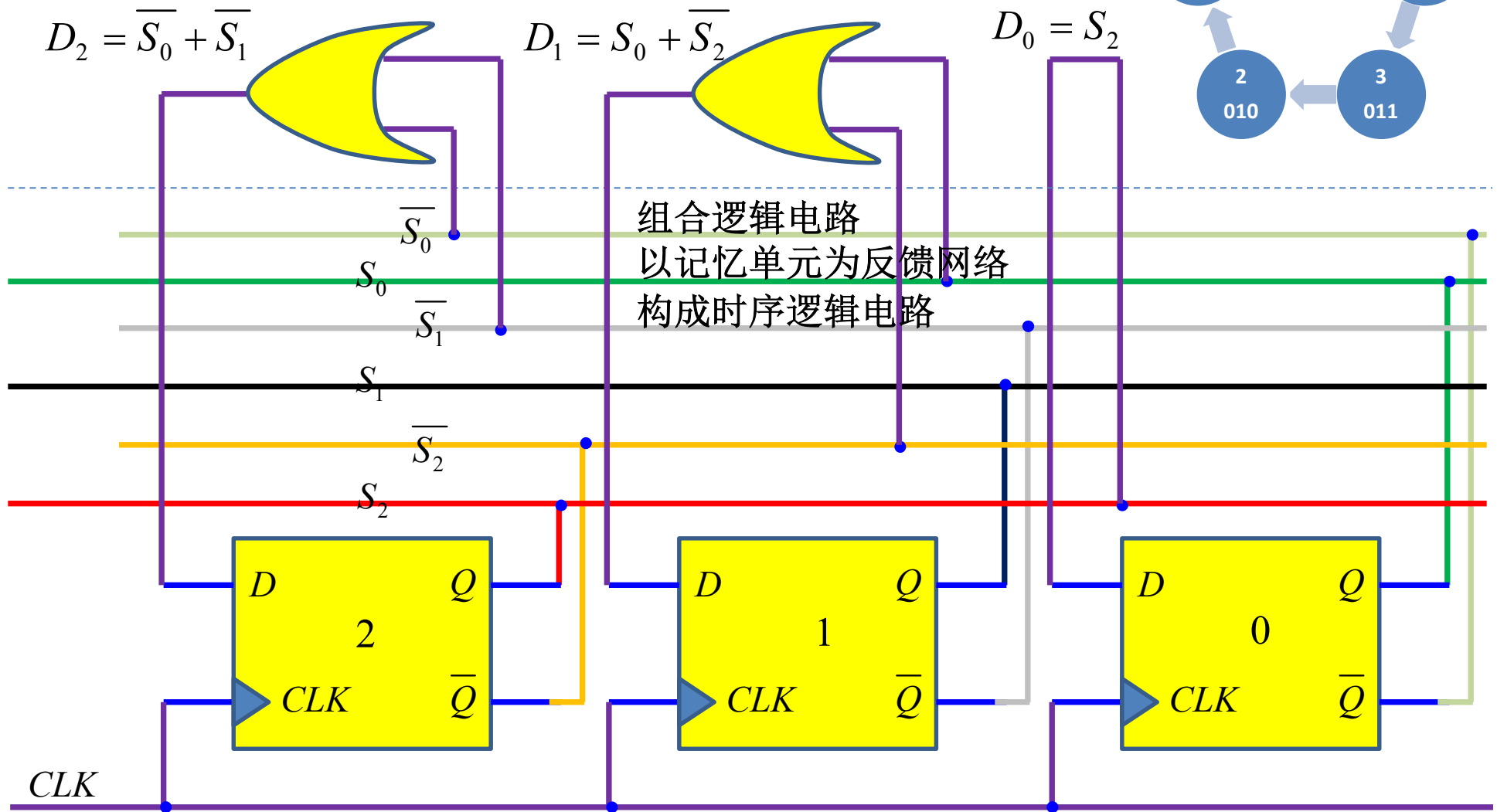
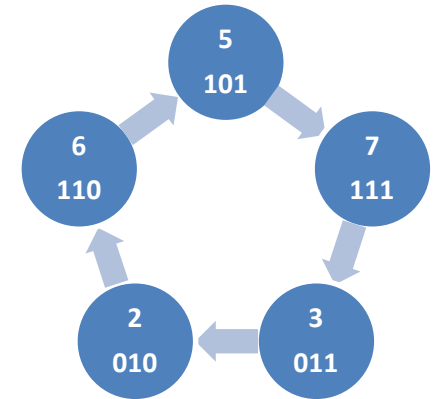
$$D_1 = S_0 + \overline{S_2}$$

D_0

S_2S_1	S_0	0	1
00		×	×
01		0	0
11		1	1
10		×	1

$$D_0 = S_2$$

组合逻辑以记忆单元为反馈 构成时序逻辑电路：计数器例



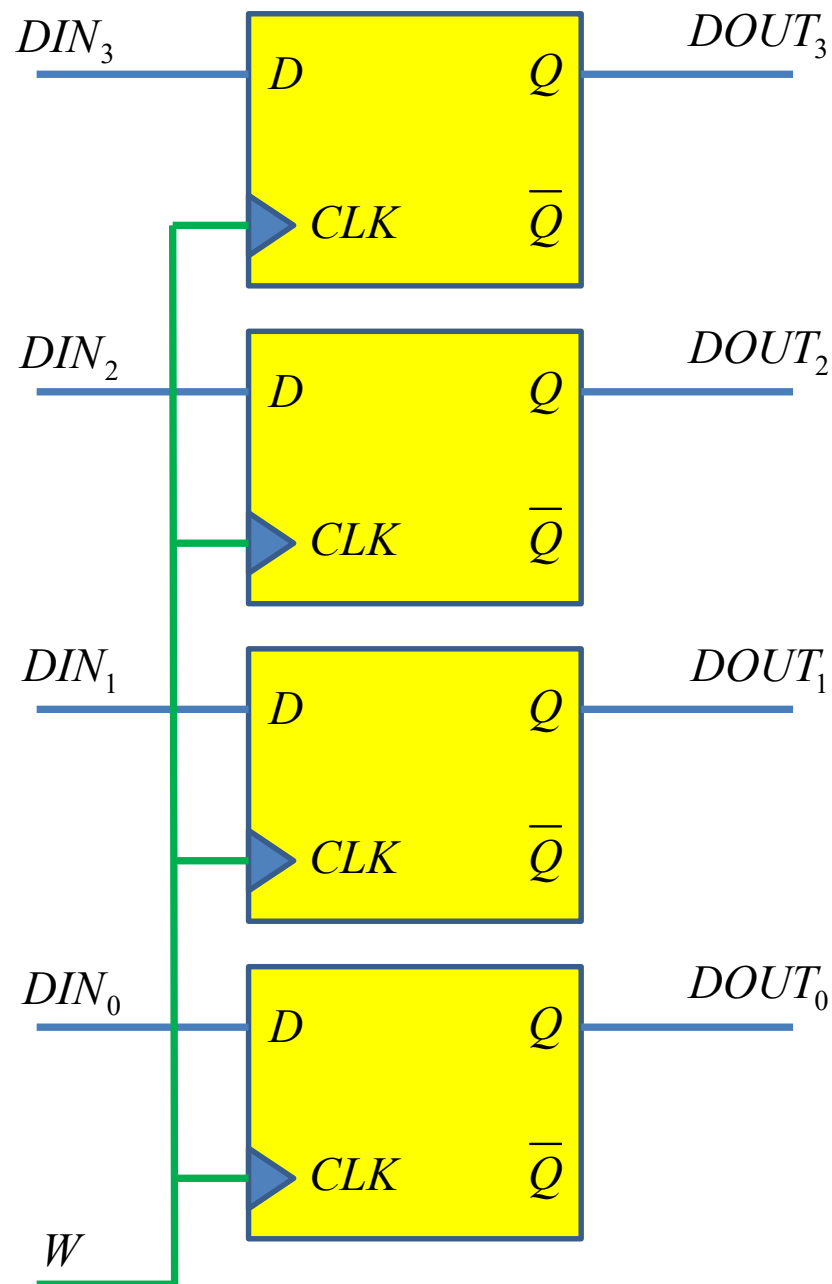
2.7 存储器 Memory

- 这里的存储器指的是被整合后的可用于计算机运算过程中进行大量数据、地址或其他中间变量存储的记忆单元
 - 寄存器
 - **SRAM**
 - **DRAM**
 - **ROM**
 - **FLASH**

2.7.1 寄存器

- 由时钟边沿触发的记忆单元还被称为寄存器register
 - 寄存：暂时存储，便于调用
 - 寄存器往往直接被嵌入在CPU逻辑电路中，便于CPU快速调用，可构成单个的寄存器或寄存器组
 - 显然，边沿触发的D触发器就是寄存器
 - 由简单数个晶体管构成的寄存器，这里不再讨论

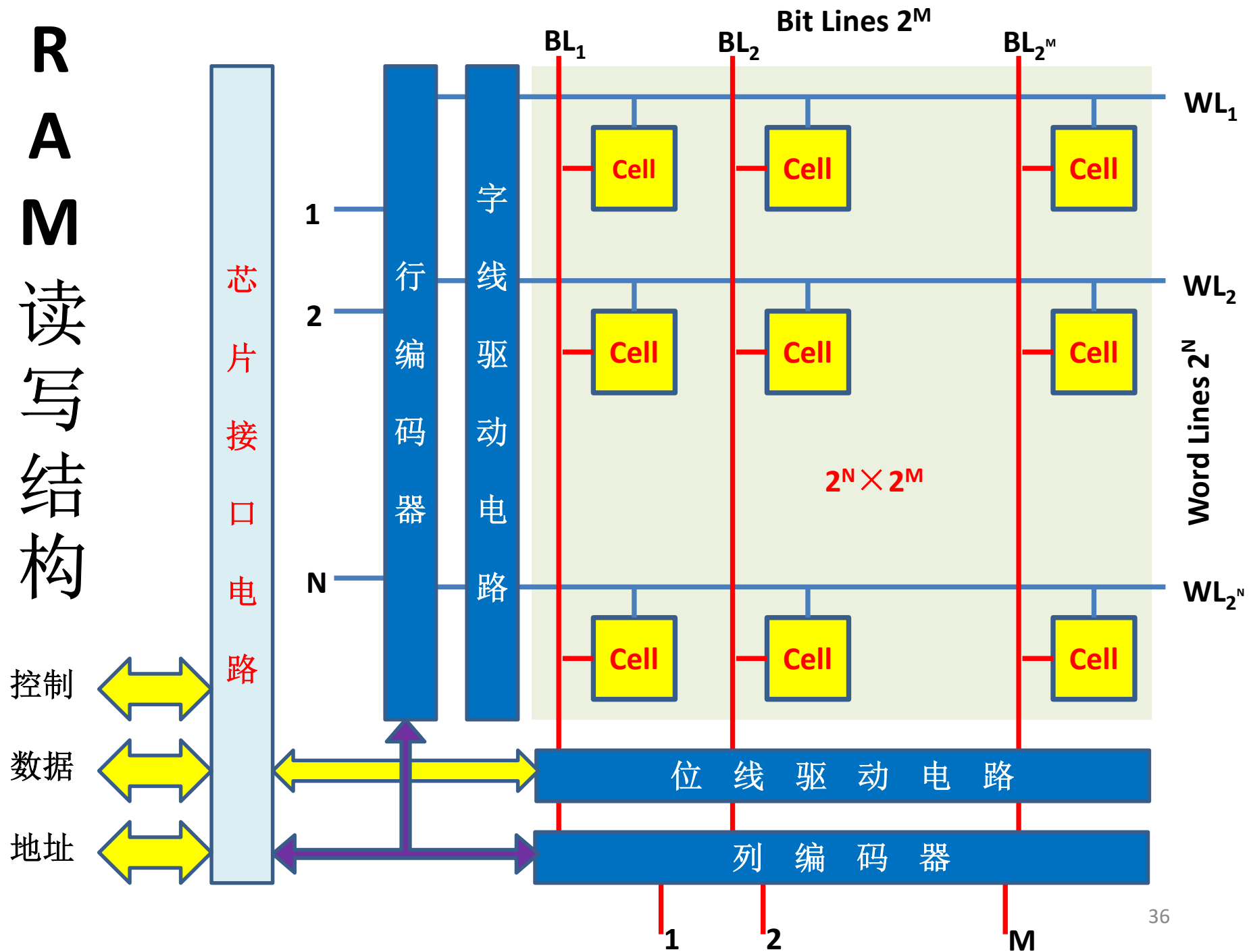
W上升沿触发后，输入数据被暂时保留在输出端，之后的时间段内，无论输入数据是否改变，只要没有新的触发信号，输出就不会改变，以前的数据被记忆（存储）下来了，可用于后期调用



Memory

- 大批量记忆单元以阵列结构形式存在，**CPU**对其调用需要阵列寻址，这种记忆单元被称为**存储器Memory**
 - 寄存器也可被称为前台存储器
 - 相应地，后者被称为后台存储器
- 描述存储器的主要指标
 - 容量：**Kbyte, Mbyte, Gbyte, Tbyte**
 - 时序：是读出还是写入，需要多长时间才能完成读出和写入
 - 存取方式：读写（**RWM**），只读（**ROM**）
 - 随机存取：**SRAM, DRAM**
 - 非随机存取：**FIFO, LIFO, ...**
 - 只读：**ROM, PROM, EEPROM, FLASH, FeRAM, ...**
 - 输入输出结构
 - 单端口：写入和读出共用
 - 多端口：写入和读出分离

RAM 读写结构

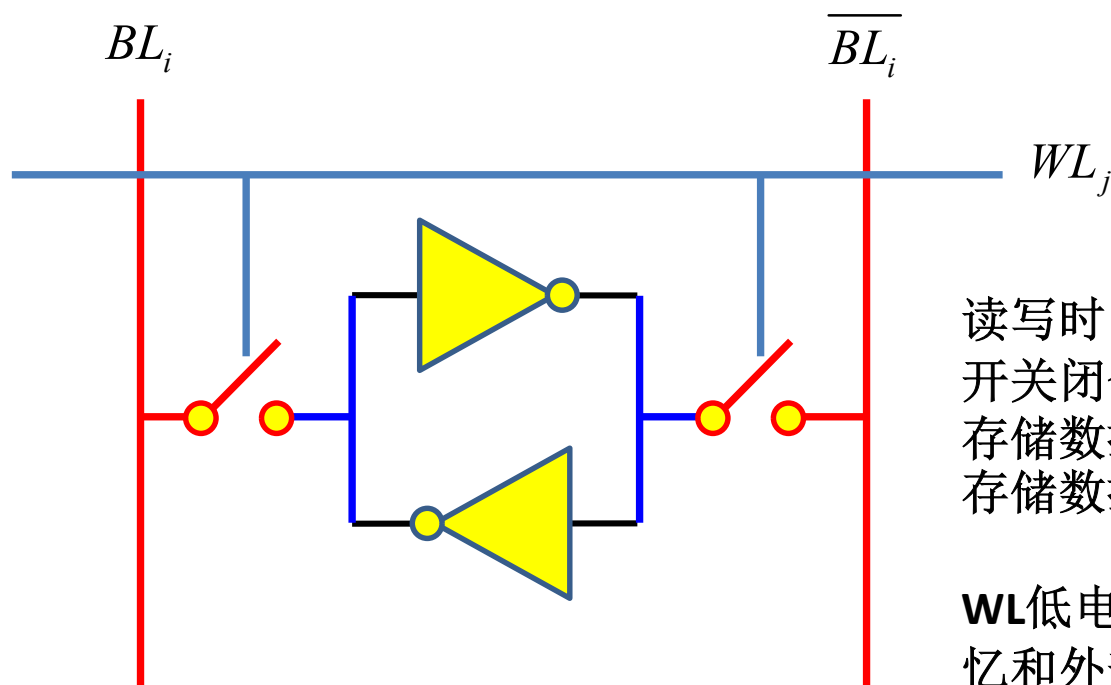
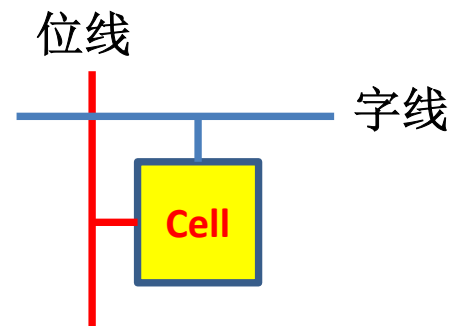


为什么采用阵列结构

- 现在集成电路对存储器的需求很大，很多集成电路内部绝大部分面积都被存储器占据
- 面积就是成本，因此为了降低成本，则需降低单个记忆单元的面积
- 采用阵列结构，可以通过降低记忆单元的某些数字特性来提高记忆单元密度
 - 噪声容限、隔离度、扇出能力、...降低
 - 形成一个记忆单元的晶体管个数降低
 - D触发器: **18**个晶体管, ...
 - SRAM Cell: **6**个晶体管
 - DRAM Cell: **4、3、2、1**个晶体管
- 阵列结构和外界的接口电路将内部记忆单元的变差了的数字特性再纠正到正常状态，便于和其他数字电路连接

2.7.2 SRAM

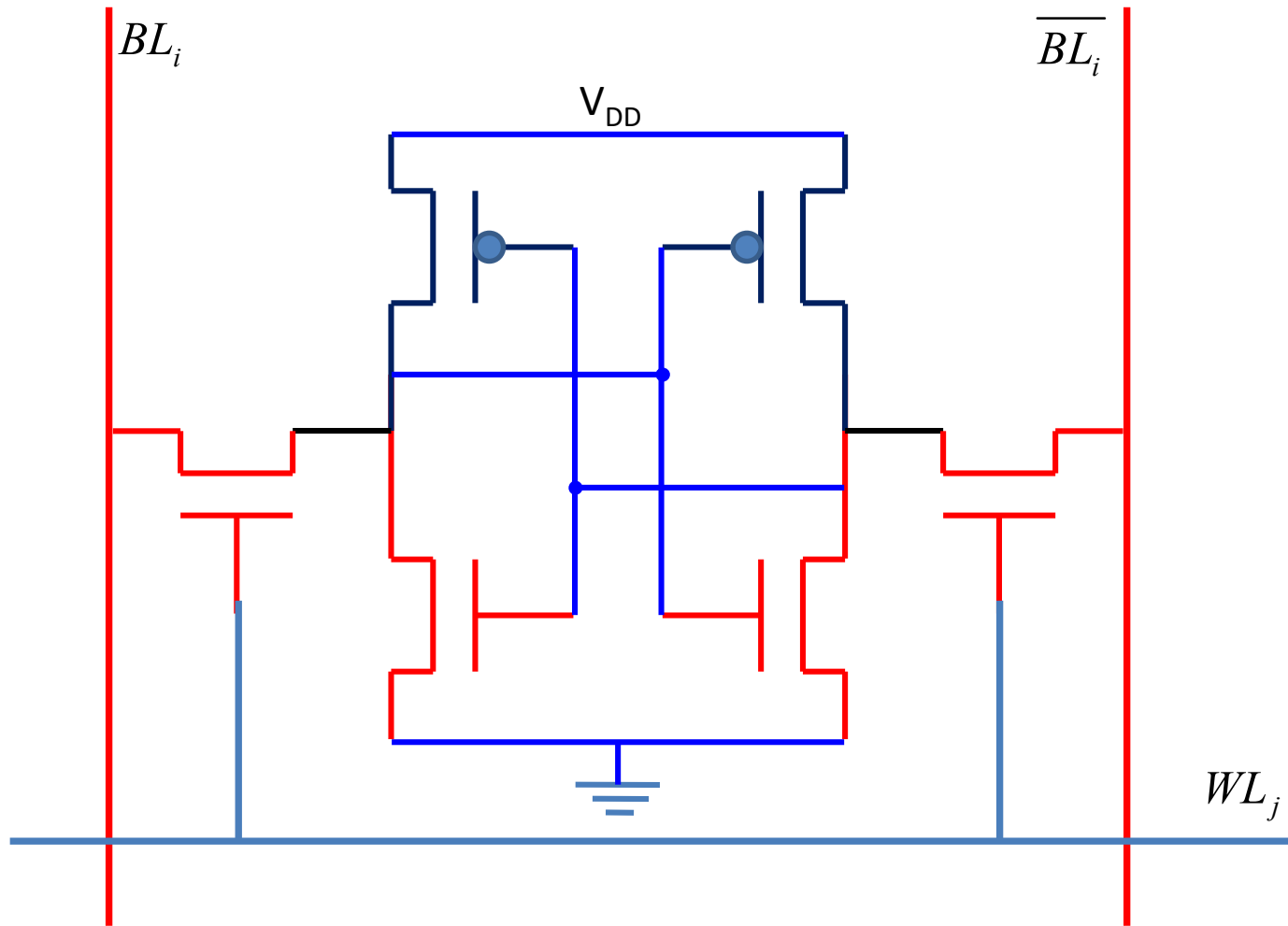
- 记忆单元基本结构



读写时，**WL**高电平，开关闭合：可读出存储数据，可写入存储数据

WL低电平，内部记忆和外部隔离，记忆保持状态

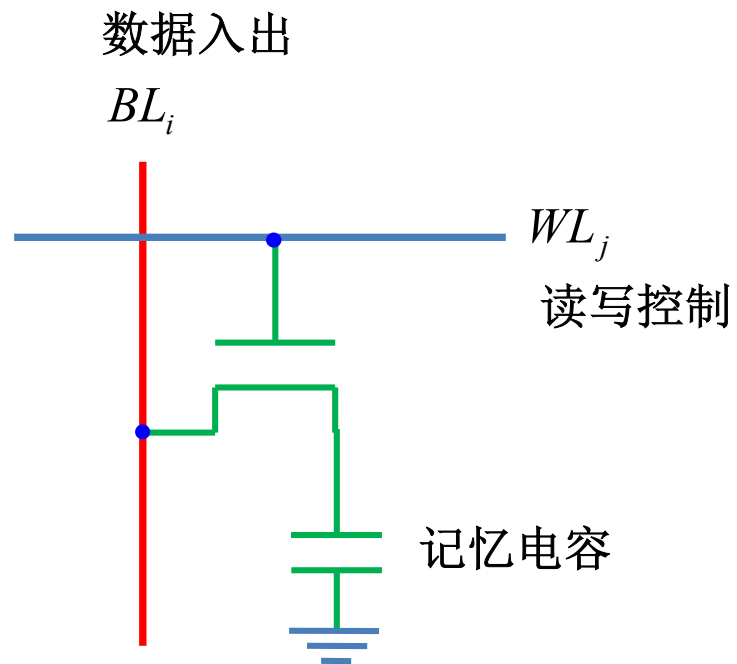
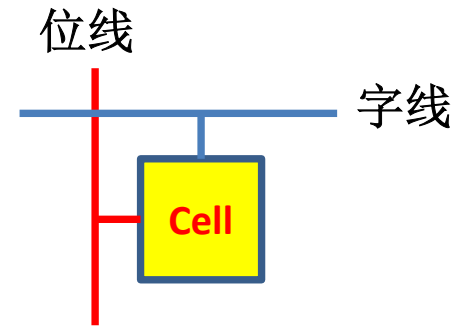
6管SRAM基本记忆单元



2.7.3 DRAM

- **SRAM: Static Random Access Memory**
 - **SRAM**通过正反馈形成双稳态，形成记忆单元
- **DRAM: Dynamic Random Access Memory**
 - **DRAM**则将状态存储于电容之上
 - 通过电容存储电荷形成记忆
 - 电荷有可能泄露，有可能在读出状态时遭到破坏，因此需要动态刷新(**refresh**)
 - **DRAM**有**4管**、**3管**、**2管**和单管结构，下面以当前最常用的单管结构为例

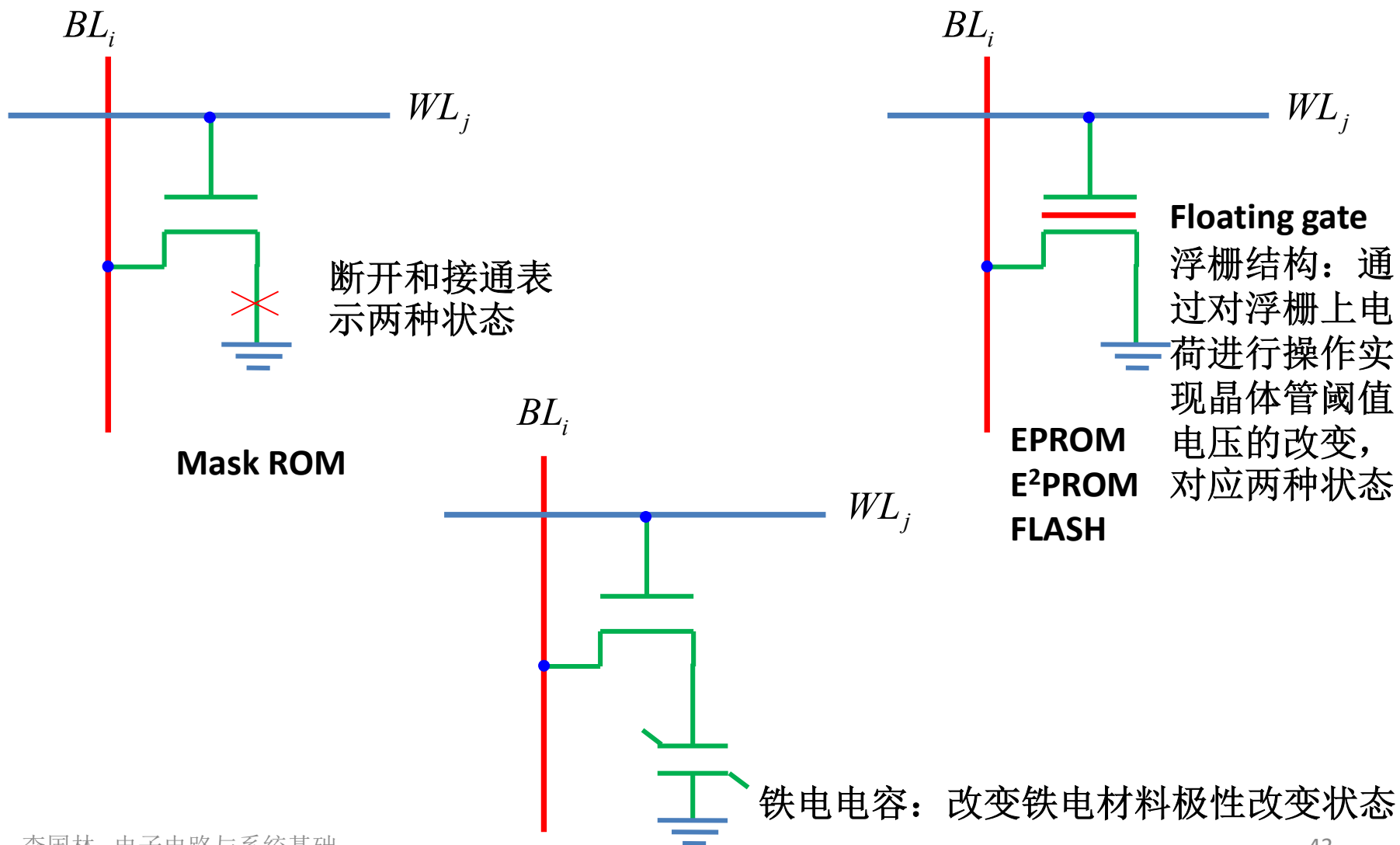
DRAM单管记忆单元



2.7.4 ROM

- 易失性存储器 **volatile memory**
 - **RAM: Random Access Memory**
- 非易失性存储器 **non-volatile memory**
 - 断电重启后，**NVM**中的记忆保持
 - 计算机的启动程序需装入**ROM**中
- **NVM**种类
 - **ROM: Mask ROM**，在芯片制作过程中写好，不可更改
 - **PROM: Programmable ROM**，可编程**ROM**，在芯片制作完成后，可编程写入
 - **Erasable PROM**: 可擦除**PROM**，紫外线擦除，可多次写
 - **Electrically Erasable PROM**: 电可擦除**PROM**，1-10万次，慢ms
 - **FLASH**: 和**E²PROM**相类似，电可擦除
 - **FeRAM: Ferroelectric RAM**，铁电存储器，同**DRAM**结构，利用铁电电容的滞回特性形成记忆

ROM基本记忆单元晶体管结构



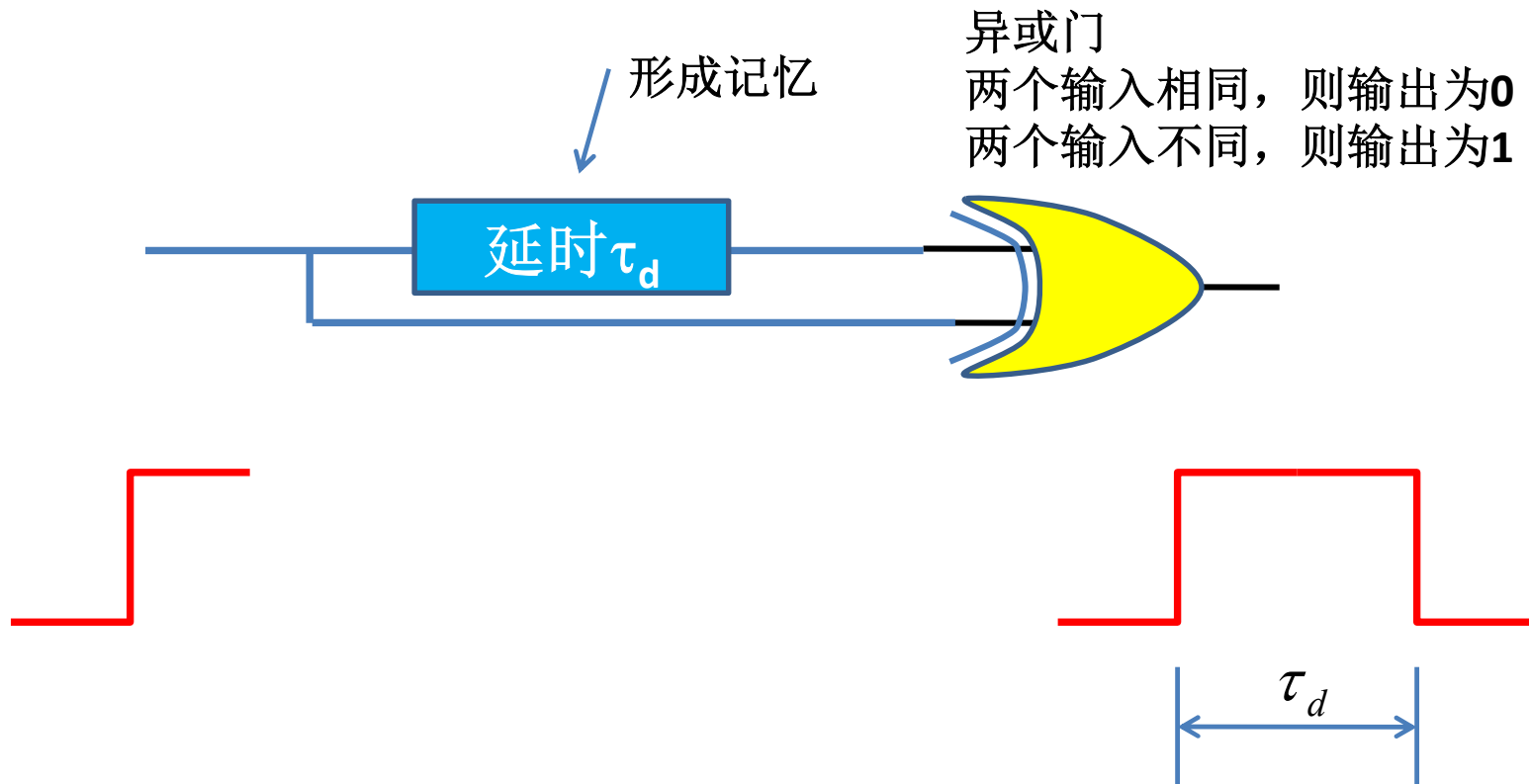
MEMEROY总结

	DRAM	SRAM	EPROM	EEPROM	Flash	FeRAM
数据易失性	是	是	否	否	否	否
刷新操作	需要	无需	无需	无需	无需	无需
单元结构	1T+1C	6T	1T	2T	1T	1T+1C
单元密度	高	低	高	低	高	高
功耗	高	高/低	低	低	低	高
读取速度	50ns	10-70ns	50ns	50ns	50ns	100ns
写入速度	40ns	5-40ns	10us	5ms	10us-1ms	100ns
成本	低	高	低	高	低	低
系统内写入	可	可	否	可	可	可
供电电源	单	单	单	多	单	单
应用例	主存储器	快速缓存	游戏机	ID卡	存储卡, U盘	照相机

二、单稳态记忆单元

- 单稳态**monostable**记忆单元，是通过延时形成记忆
 - 只有一个稳定状态：静态时的状态
 - 另一个是准稳状态，在准稳状态不能停留很长时间
 - 一个触发事件导致一个短时进入准稳状态，形成一个确定脉宽的脉冲，故而又称单脉冲电路**one-shot**
 - 上升沿、下降沿检测，地址变化检测
 - 通过检测变化，产生一个脉冲，启动一个操作

简单实现原理



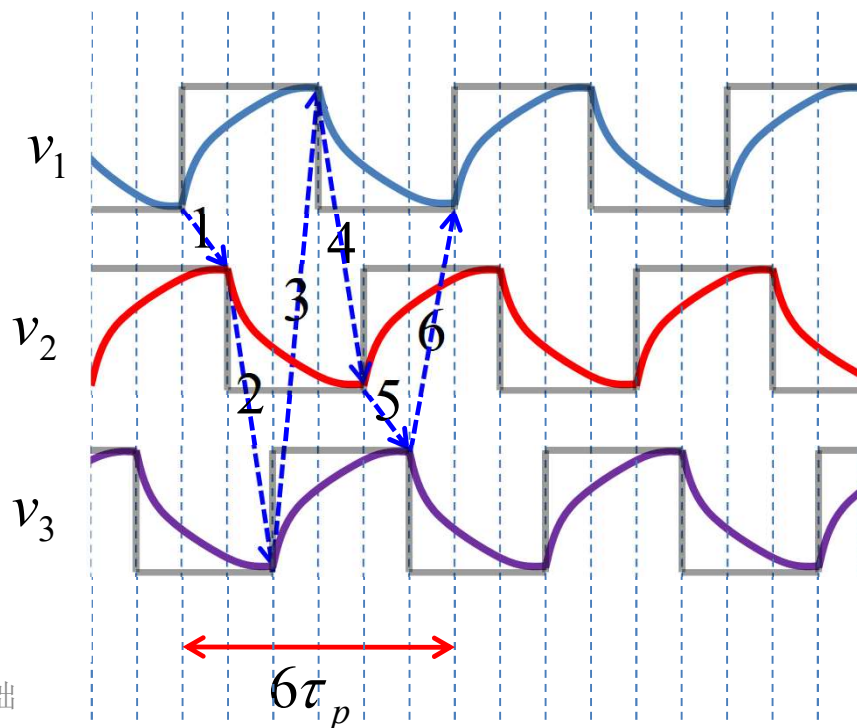
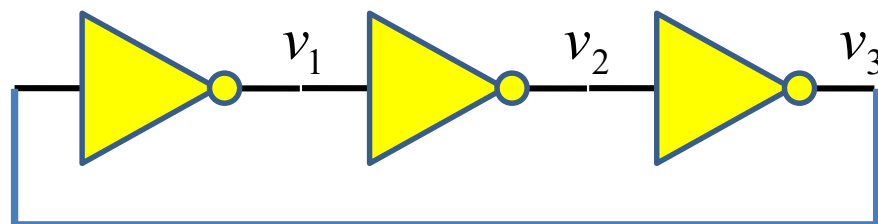
上次布置作业为单稳态电路

三、无稳态记忆单元

- 无稳态**astable**电路不具可停留的稳定状态
 - 它有两个准稳态，电路在两个准稳态之间来回转换，形成振荡
 - 在两个准稳态停留的时间由电路中存在的寄生延时或人为设计的延时决定
 - 延时就是所谓的记忆
- 无稳态记忆单元可形成振荡器电路，用于产生周期性方波信号
 - 方波频率由延时决定

环形振荡器

- 环形振荡器由奇数个反相器头尾相连构成




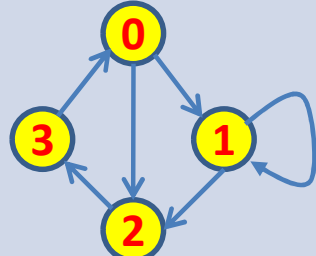

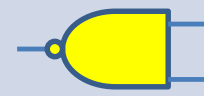

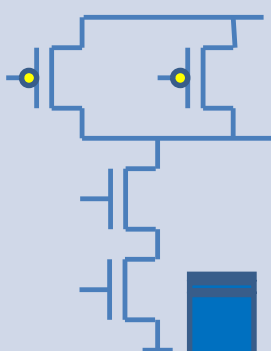

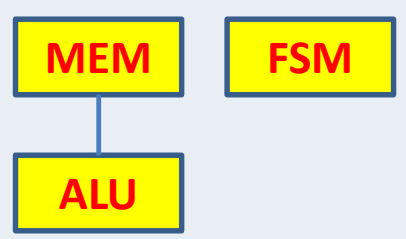
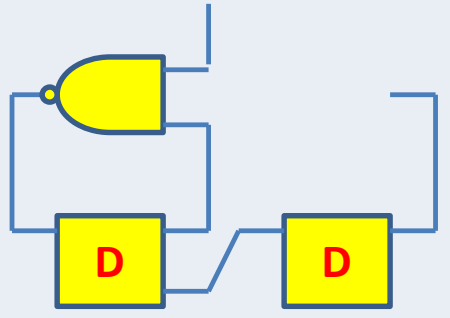
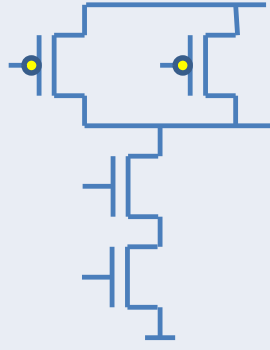
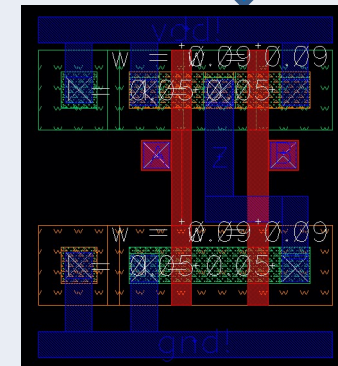
经过Schmitt触发器，
可形成方波输出

$$f_{osc} = \frac{1}{6\tau_p}$$

四、数字系统综合的层次

- 数字系统综合
 - 将某种信号处理的要求转换为数字电路的实现
- 数字系统综合分为四个层次
 - 体系结构级综合
 - 逻辑级综合
 - 电路级综合
 - 版图级综合
- 无论那个层次的综合，都需要经过相同的过程
 - 将行为描述转换为结构描述

数字系统综合

	体系结构级	逻辑级	电路级	版图级
行为描述	<pre>for i=1:100 sum=sum+a(i)*b(i) end</pre> 	 	$z = \overline{A}B$  	 
结构描述				

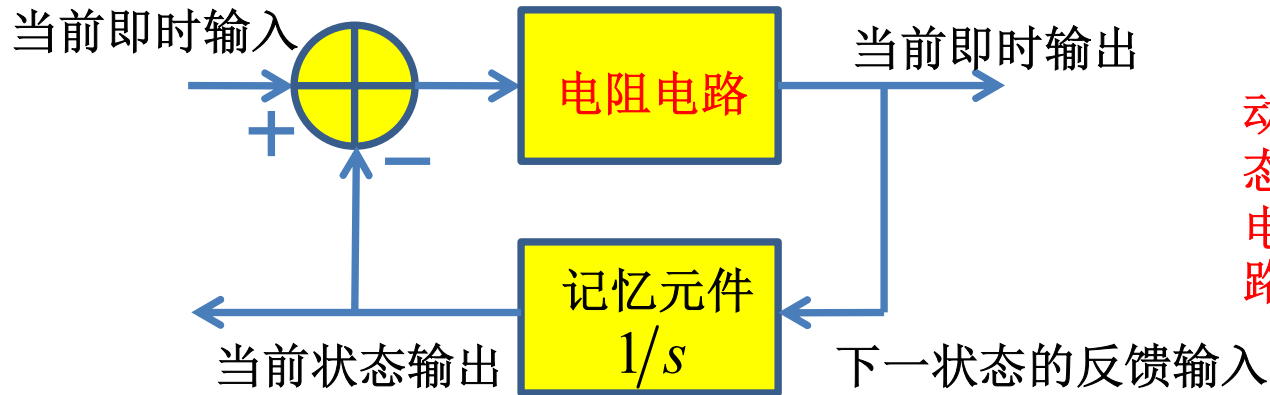
MEM: memory

ALU: Arithmetic Logic Unit

FSM: Finite-State Machine

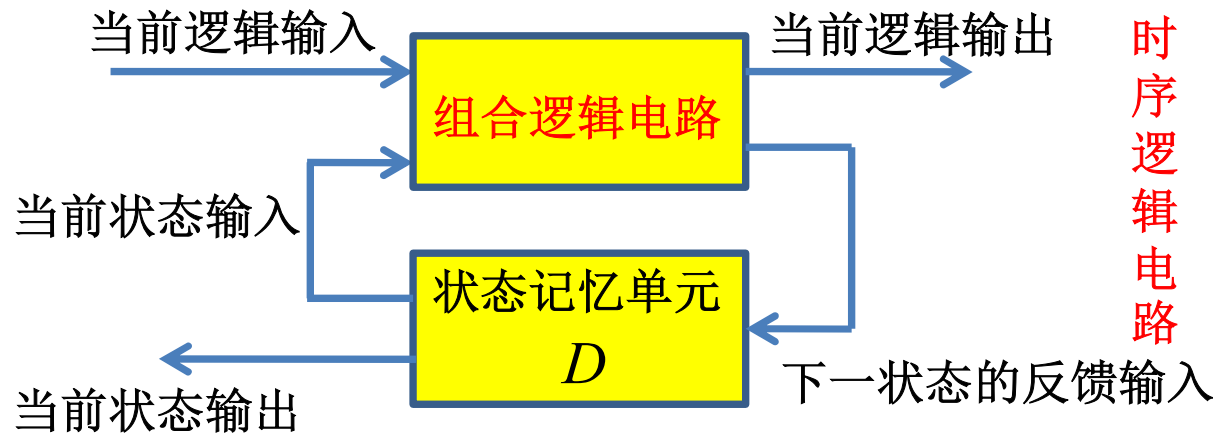
小结

当前输入和当前状态共同决定当前输出



动态电路

电阻电路处理的模拟信号被电容、电感、传输线等记忆元件存储下来，用于下一时刻的即时处理，从而形成动态电路：由于存储记忆而有状态转移，有时间效应，有频率效应



时序逻辑电路

组合逻辑电路处理的逻辑信号被触发器、寄存器、存储器等状态记忆单元存储下来，用于下一时刻的逻辑处理，从而形成时序逻辑：由于存储记忆而有状态转移，有时序效应

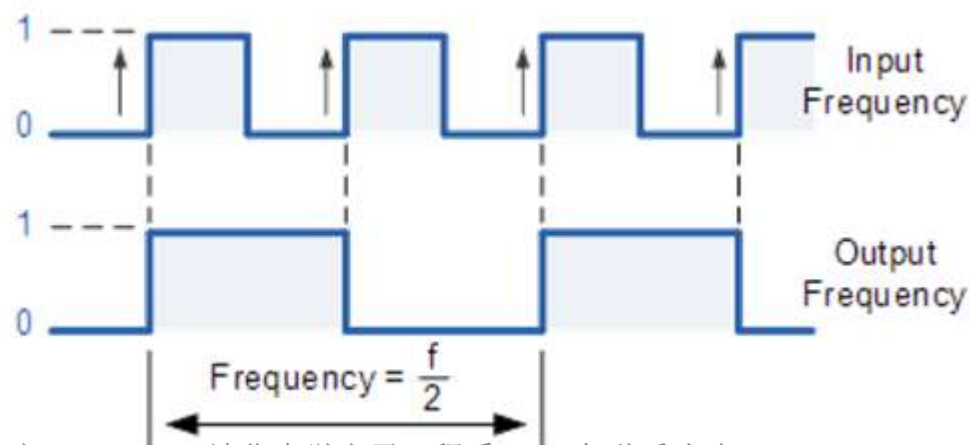
模拟电路中的积分和数字电路中的延时是等价的记忆单元

作业01 计数器设计的后验算

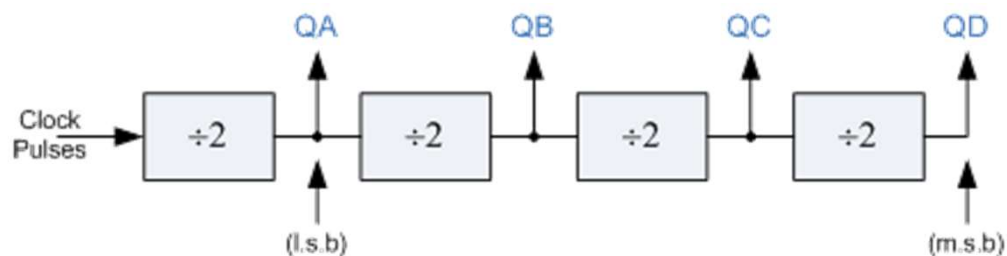
- 课件设计的**5**状态计数器，采用**3**个**D**触发器作为记忆单元，**3**个**D**触发器共具**8**个状态，其中有**3**个状态是不用的，确认剩下的**3**个状态可并入到状态转移图中
 - 如果这**3**个状态形成了自闭合的状态转移，形成了自闭合的状态空间，则设计是有问题的，因为加电后初始状态可能是这**3**个状态之一
 - 如果出现这种情况，计数器设计需要有某种机制使得它自动进入设计的状态空间中

作业

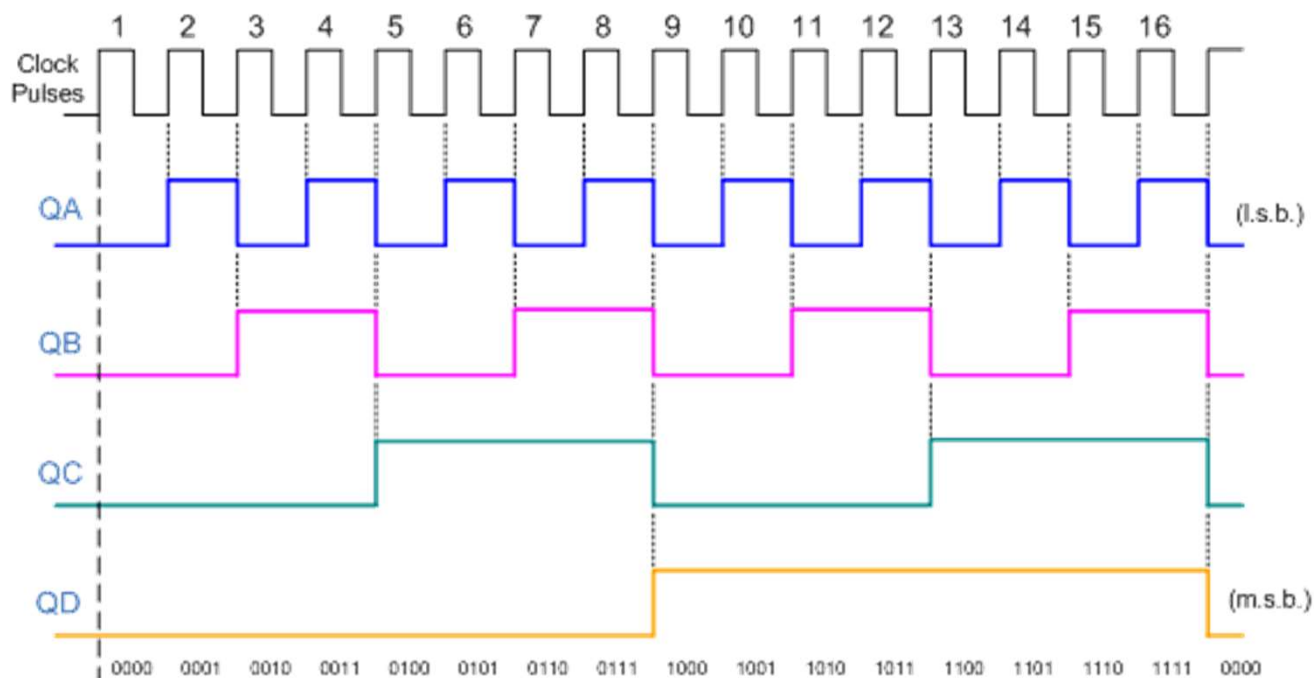
- **02** 采用和课件完全相同的处理手法，请用**D**触发器设计一个**4bit**的十计数器，该计数器在时钟驱动下，可以依次循环输出**0,1,2,3,4,5,6,7,8,9**
 - 画状态转移图
 - 设计组合逻辑电路
 - 检查剩余状态是否可自动进入设计的状态空间，否则重新设计
- **03** 请用**D**触发器实现**2分频器** **先给出状态转移图，再设计组合逻辑电路**



作业04 顺序计数器：画逻辑电路图



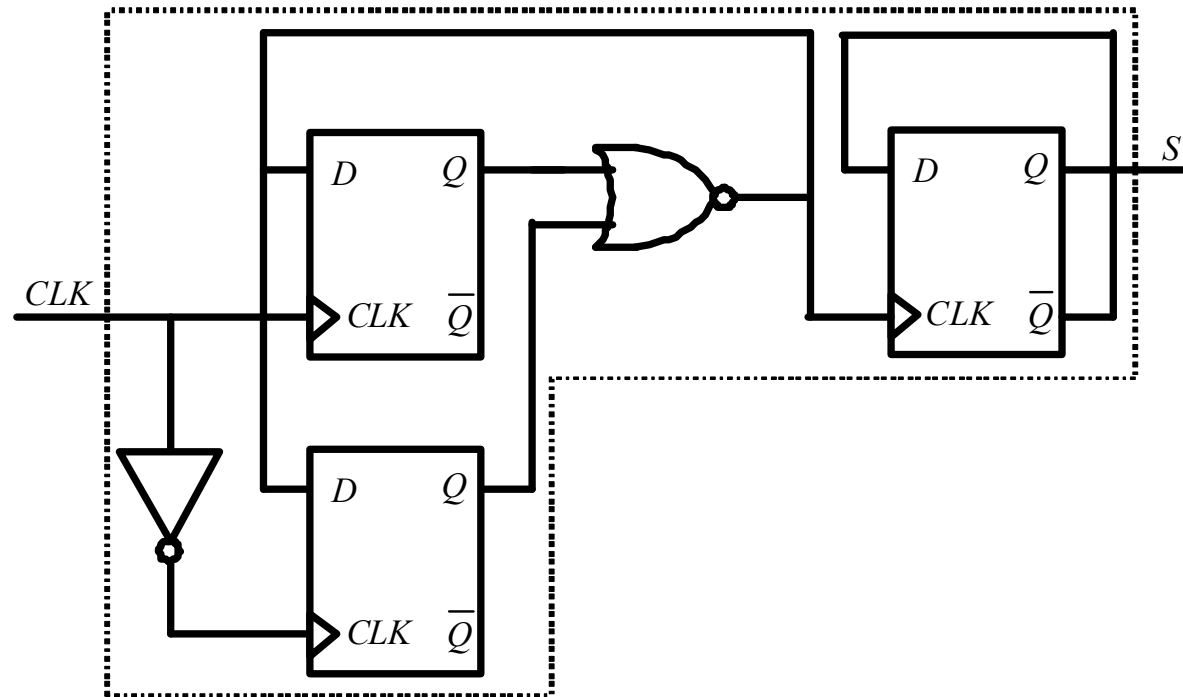
行为描述



逻辑电路级实现?

作业05 分析电路功能

- 试分析下面电路的功能，已知**CLK**为输入方波时钟信号，**S**为输出信号
 - **D**触发器初始状态任意，可以从**000**出发进行分析



CAD仿真

- 构造两个CMOS数字非门，通过调整晶体管尺寸，使得其翻转电压为 $0.5V_{DD}$
 - 建模
 - 理论计算和仿真符合
- 加压求流，仿真确认图示单端口为N型负阻
- 在该单端口加什么样的器件，可形成振荡器？仿真确认确实振起来了，且波形符合设计
 - 张弛振荡
 - （选作：正弦振荡）
- 选作：用两个数字非门和其他无源器件，构成振荡电路，仿真波形和分析波形一致。

