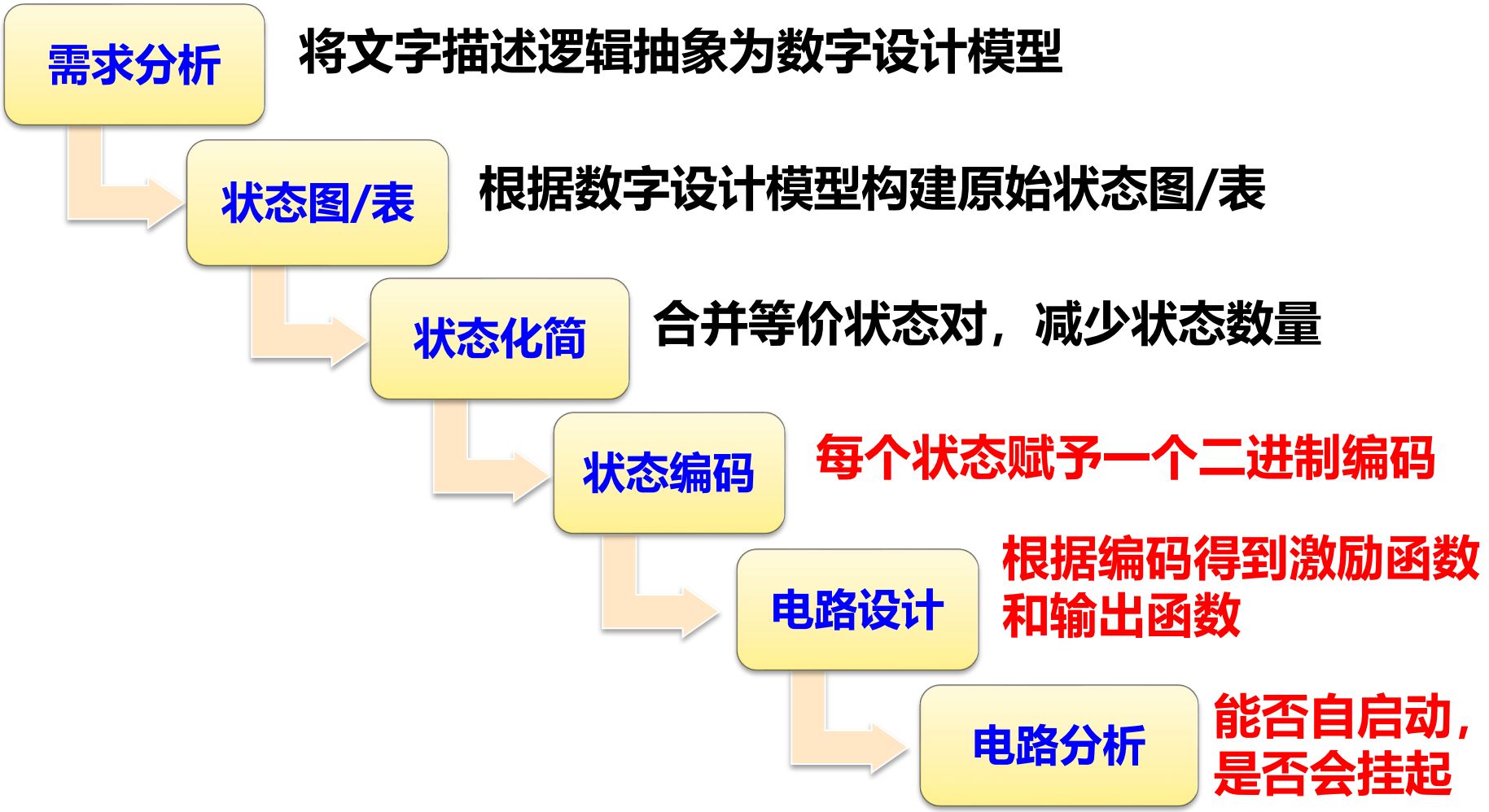
第三讲　同步时序逻辑设计

- 同步时序逻辑设计步骤
- 状态图/状态表设计
- 状态化简和状态编码
- 电路设计和分析

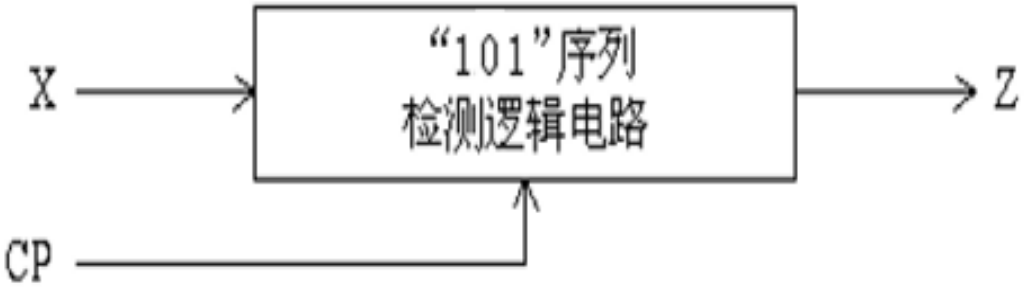
3.1 同步时序逻辑设计步骤



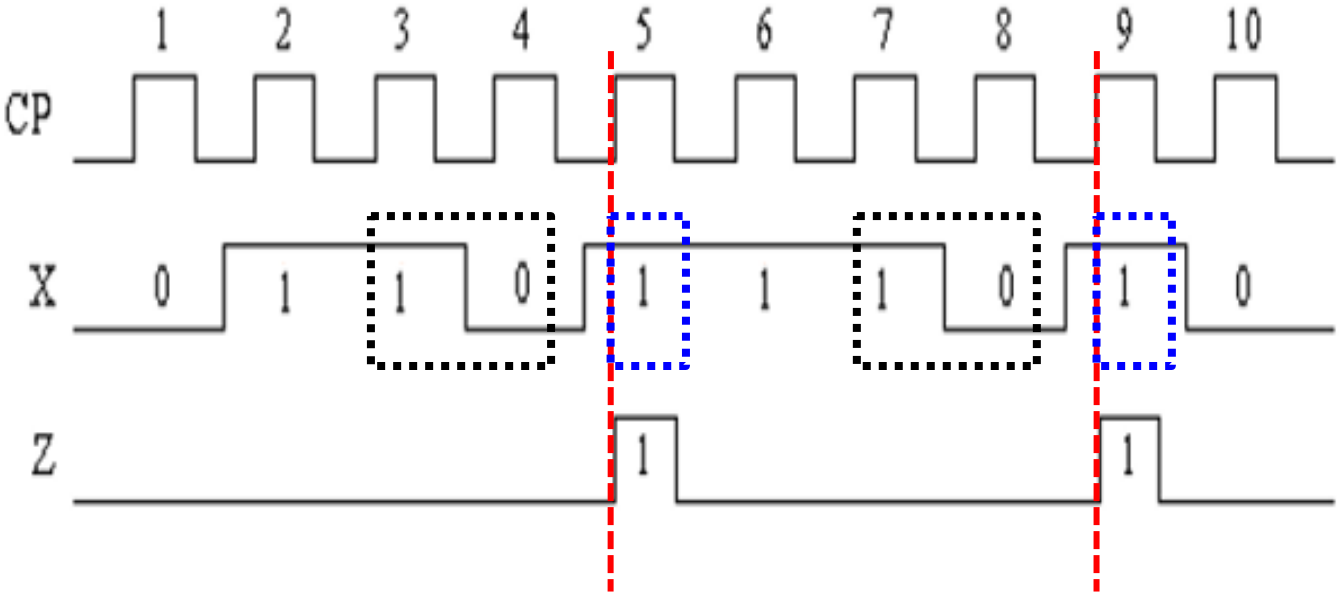
3.2 需求分析

例：检测一连串0/1输入序列中是否出现“101”

1位输入端X；
1位输出端Z。



CP脉冲到来时，根据当前X的值，确定输入序列中是否出现“101”



若是，则输出Z为1；否则Z为0。

3.2 状态图/状态表设计

2. 构建状态图/表：分析系统内部的状态转换关系

I. 设定任一状态为电路初始状态；

II. 从初始状态开始，分析每一个状态在不同输入作用下的状态转移情况和输出取值；

III. 如果某状态下出现的输出响应（次态、输出）不能用已有状态表示，则产生新的状态；

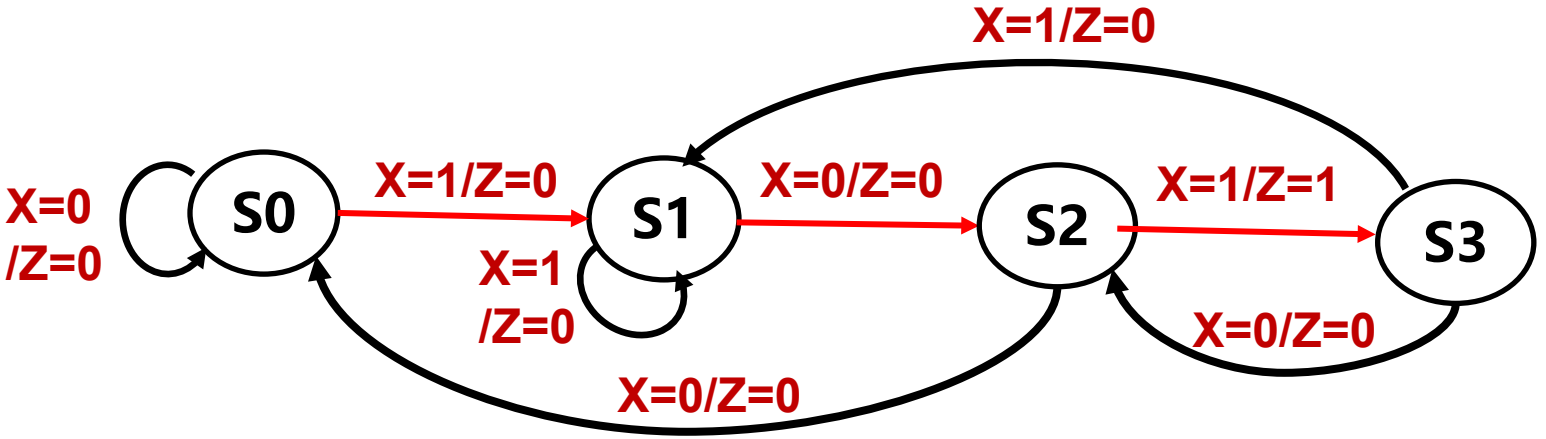
IV. 重复第II、III两步，直到不产生新状态为止。

S0：初始状态，等待接收输入

S1：接收到“101”序列中第一个1

S2：接收到该序列中的10

S3：接收到一个“101”序列



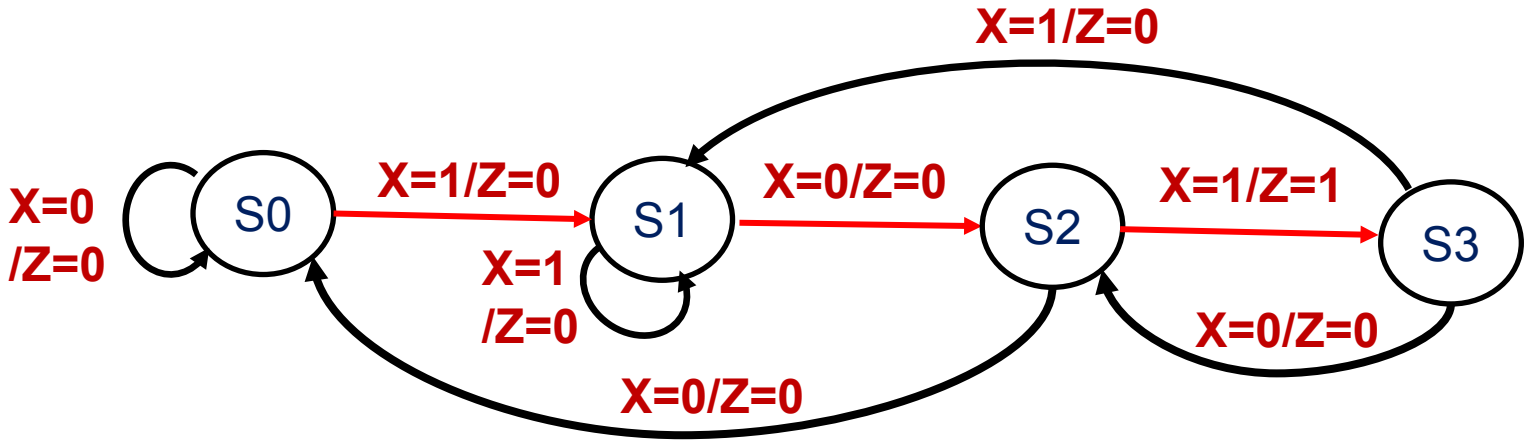
3.2 状态图/状态表设计

S0：初始状态，等待接收输入

S1：接收到“101”序列中第一个1

S2：接收到该序列中的10

S3：接收到一个“101”序列



2. 构建状态图/表

- 根据状态图构建状态表

  X：输入数据；Z：检测结果

  S：当前状态；S^*：次态

状态表

现态S	S*/Z	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S3/1
S3	S2/0	S1/0

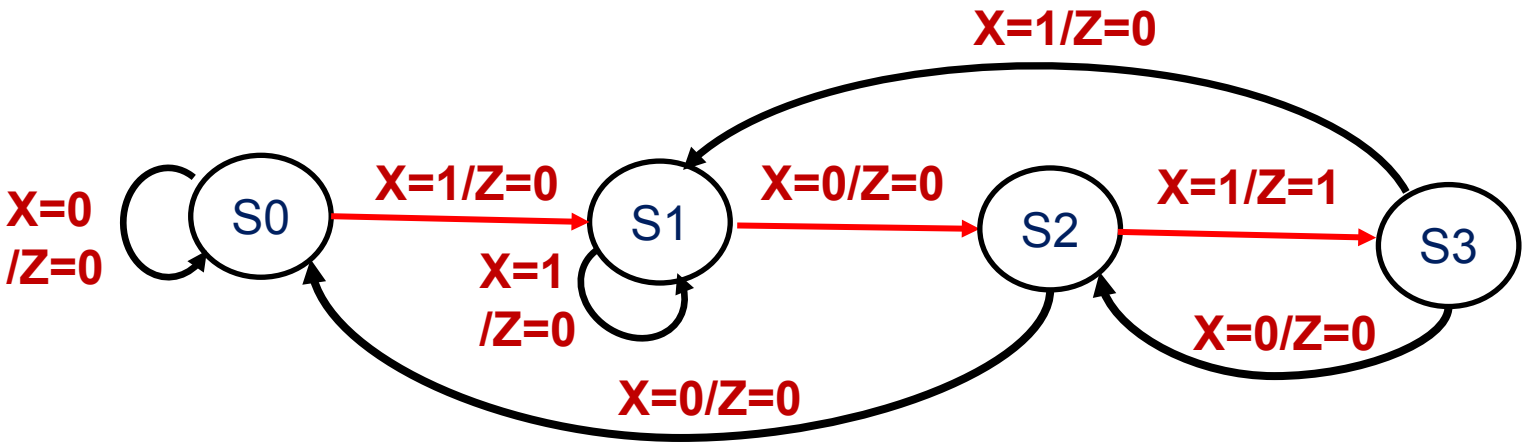
3.2 状态图/状态表设计

- 构建状态图/表时，状态转移需满足下列两个条件。

  互斥性：从每个状态出发的所有状态转换路径上的转换条件都是互斥的，如本例中，转移条件分别是X=0和X=1，互斥。

  完备性：从每个状态出发的所有状态转换路径上的转移表达式的**逻辑或等于1**（逻辑真）。如X=0和X=1，0+1=1。

- 在状态图中，也可以使用逻辑表达式来表示转移条件。本例中，可以使用X和$\overline{X}$分别表示输入X=1和X=0。



3.3a 状态化简

- 合并等价状态，以得到更加精简的状态表
- 两个状态等价指在所有输入组合下，它们的输出相同且次态相同或次态等价

状态表

现态S	S*/Z	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S3/1
S3	S2/0	S1/0

S1和S3构成等价类，可合并化简后，有3个状态

现态S	S*/Z	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S1/1

- 等价关系具有传递性
  - 例如，若状态A和B等价，同时B和C等价，则A和C也等价。状态A、B和C属于一个等价类，可以合并为一个状态。

3.3b 状态编码

- 对状态表中每个状态赋予唯一的二进制编码，也称状态赋值
- 寻找最优编码方案是一个非常复杂的问题
- 通常在具体设计时采用相邻法寻求较优编码方案：

  准则1：若两个状态的次态相同，则其对应编码应尽量相邻

  准则2：同一个现态的各个次态其编码应尽量相邻

  准则3：若两个现态的输出相同，则它们的编码应尽量相邻

现态S	S*/Z	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S1/1

根据准则1：S0和S2可相邻

根据准则2： S0和S1、S1和S2可相邻

根据准则3： S0和S1可相邻

根据不同的取舍可得到不同编码方案

若S0和S1、S1和S2相邻，则编码方案为：S0:00，S1:01，S2:11

若S0和S2、S0和S1相邻，则编码方案为：S0:00，S1:01，S2:10

3.4 电路设计（次态函数，输出函数）

- 在选定的状态编码方案基础上进行电路设计

若编码方案S0:00，S1:01，S2:11

现态S	S*/Z	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S1/1

生成状态转移表（下表是简化的）

Y1Y0	Y1*Y0*/Z	
	X=0	X=1
00	00/0	01/0
01	11/0	01/0
11	00/0	01/1

- 根据状态转移表，推导次态逻辑函数和输出逻辑函数
- 次态函数/次态方程为：

$Y1^* = \overline{Y1} \cdot Y0 \cdot \overline{X}$

$Y0^* = \overline{Y1} \cdot Y0 \cdot \overline{X} + X \cdot (\overline{Y1} \cdot \overline{Y0} + \overline{Y1} \cdot Y0 + Y1 \cdot Y0)$

$= \overline{Y1} \cdot Y0 + X \cdot \overline{Y1} + X \cdot Y0$

$Z = Y1 \cdot Y0 \cdot X + Y1 \cdot$

3.4 电路设计（函数化简）

Y1Y0	Y1*Y0*/Z	
	X=0	X=1
00	00/0	01/0
01	11/0	01/0
11	00/0	01/1
10	00/0	01/1

采用代数法化简

$Y1^*=\overline{Y1}\cdot Y0\cdot\overline{X}$

$Y0^*=\overline{Y1}\cdot Y0\cdot\overline{X}+X\cdot(\overline{Y1}\cdot\overline{Y0}+\overline{Y1}\cdot Y0+Y1\cdot Y0)$

$=\overline{Y1}\cdot Y0\cdot\overline{X}+X\cdot\overline{Y1}\cdot\overline{Y0}+X\cdot\overline{Y1}\cdot Y0+X\cdot Y1\cdot Y0$

$=\overline{Y1}\cdot Y0\cdot\overline{X}+\overline{Y1}\cdot Y0\cdot X+X\cdot\overline{Y1}\cdot\overline{Y0}+X\cdot\overline{Y1}\cdot Y0+X\cdot Y1\cdot Y0+X\cdot\overline{Y1}\cdot Y0$

$=\overline{Y1}\cdot Y0+X\cdot\overline{Y1}+X\cdot Y0$

将无关项编码Y1Y0=10引入化简，则

$Y0^*=\overline{Y1}\cdot Y0\cdot\overline{X}+X\cdot(\overline{Y1}\cdot\overline{Y0}+\overline{Y1}\cdot Y0+Y1\cdot Y0+Y1\cdot\overline{Y0})$

$=\overline{Y1}\cdot Y0\cdot\overline{X}+X\cdot\overline{Y1}\cdot\overline{Y0}+X\cdot\overline{Y1}\cdot Y0+X\cdot Y1\cdot Y0+X\cdot Y1\cdot\overline{Y0}$

$=\overline{Y1}\cdot Y0\cdot\overline{X}+\overline{Y1}\cdot Y0\cdot X+X\cdot\overline{Y1}\cdot\overline{Y0}+X\cdot\overline{Y1}\cdot Y0+X\cdot Y1\cdot Y0+X\cdot\overline{Y1}\cdot Y0$

$+X\cdot Y1\cdot Y0+X\cdot Y1\cdot\overline{Y0}+X\cdot\overline{Y1}\cdot\overline{Y0}+X\cdot Y1\cdot\overline{Y0}$

$=\overline{Y1}\cdot Y0+X\cdot\overline{Y1}+X\cdot Y0+X\cdot Y1+X\cdot\overline{Y0}$

$=\overline{Y1}\cdot Y0+X$

$Z=Y1\cdot Y0\cdot X+Y1\cdot\overline{Y0}\cdot X=Y1\cdot X$

3.4 电路设计（函数化简）

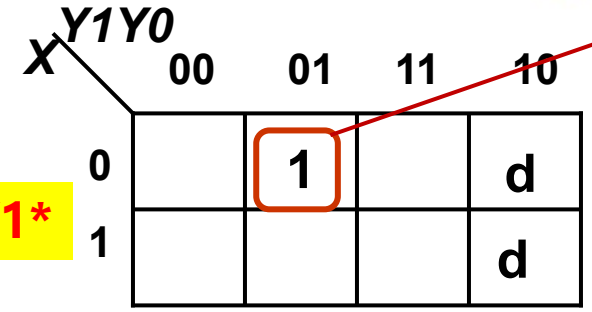
利用卡诺图化简

状态转移表

Y1	Y0	X	Y1*Y0*Z		
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	d	d	d
1	0	1	d	d	d
1	1	0	0	0	0
1	1	1	0	1	1

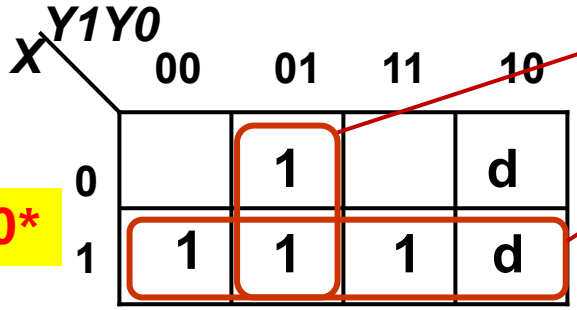
$Y1^* = \overline{Y1} \cdot Y0 \cdot \overline{X}$

Y1*



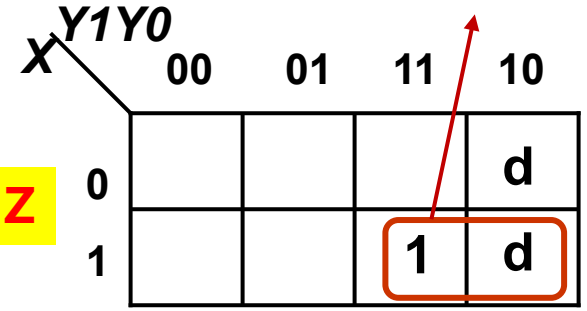
$Y0^* = \overline{Y1} \cdot Y0 + X$

Y0*



$Z = Y1 \cdot X$

Z



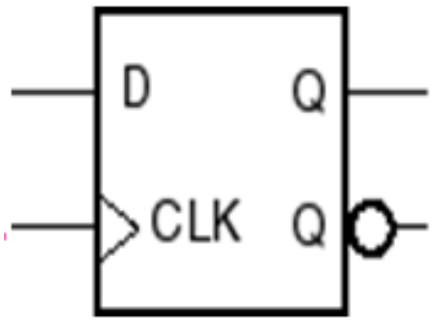
3.4 电路设计（确定元件，画电路图）

- 根据次态函数和选择的状态记忆单元（触发器），推导出激励函数
  - 假设采用D触发器，其特征方程Q* = D，则：

    $D1=Y1^*=\overline{Y1}\cdot Y0\cdot\overline{X}$

    $D0=Y0^*=\overline{Y1}\cdot Y0+X$
- 输出函数为：$Z=Y1\cdot X$

要基于Q和激励函数，计算出新状态，传给D



Q一直稳定输出，代表的是当前状态

要基于Q和输入X，用输出函数计算出输出Z

需要几个D触发器呢?

回顾第10次课

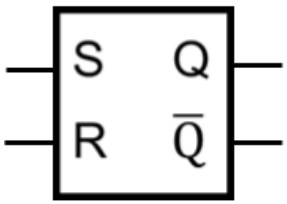
- 锁存器：用输入信号电平 驱动双稳态元件变为 输入信号指定的状态（0或1）
- 触发器：由时钟信号边沿 驱动双稳态元件变为 输入信号指定的状态（0或1）
- 状态改变所需的时间：锁存（触发）延迟

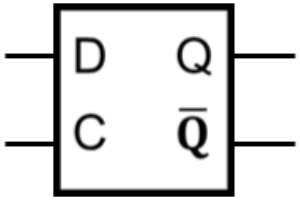
- SR锁存器（输入S和R决定Q），D锁存器（输入D决定Q，C电平控制）

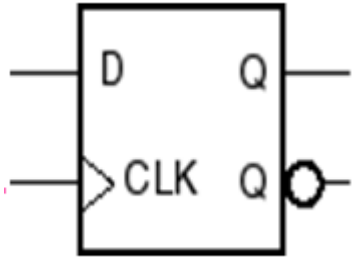
- D触发器（输入D决定Q，时钟边沿控制）：
  Clk-to-Q，setup，hold
  带使能，带预置
- T触发器（每个时钟边沿到来时，输出反转）

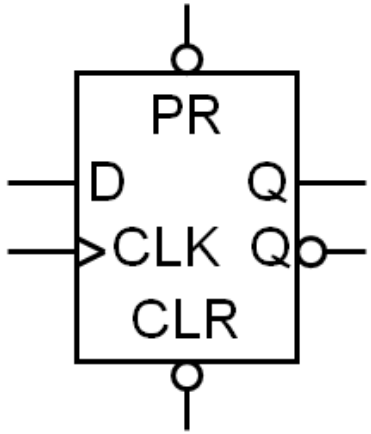
- 时序逻辑电路的设计：
  状态图，状态化简，状态编码（若干规则）
  状态转移表（若干真值表），逻辑表达式（函数）化简

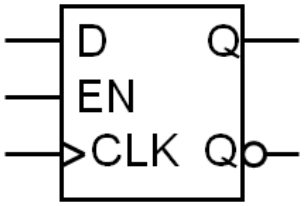
回顾第10次课（续）

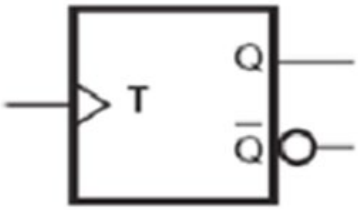












次态方程

激励方程

输出方程

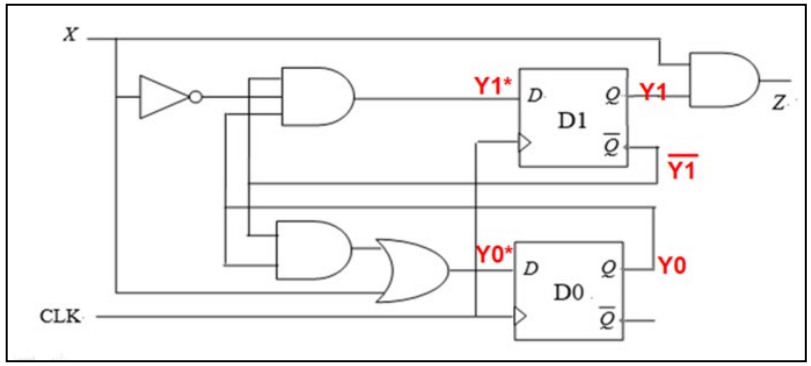
$Z=Y1 \cdot X$

$D1=\overline{Y1} \cdot Y0 \cdot \overline{X}$

$D0=\overline{Y1} \cdot Y0 + X$

$Y1^*=\overline{Y1} \cdot Y0 \cdot \overline{X}$

$Y0^*=\overline{Y1} \cdot Y0 + X$



3.4 电路设计（确定元件，画电路图）

- 根据次态函数和选择的状态记忆单元（触发器），推导出激励函数
  - 假设采用D触发器，其特征方程Q* = D，则：

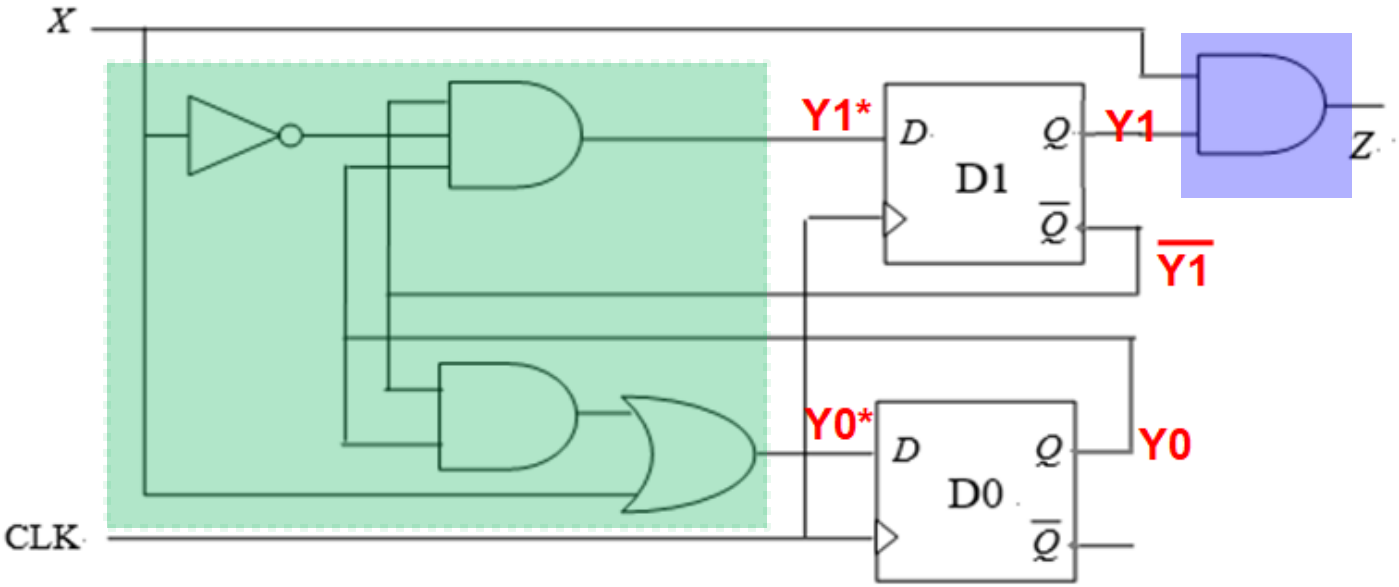
$D1=Y1^*=\overline{Y1}\cdot Y0\cdot\overline{X}$

$D0=Y0^*=\overline{Y1}\cdot Y0+X$

- 输出函数为：$Z=Y1\cdot X$

(1) 触发器存储着旧状态Y0，Y1
(2) 结合输入的X，激励模块和输出模块计算出新状态和输出结果
(3) 新状态存入触发器，回到 (1)

- 根据激励函数和输出函数，画出逻辑电路图



3.4 电路设计后的分析（未用状态分析）

- **电路分析：包括未用状态分析和电路定时分析等**
  - **通常编码空间比状态**
  - **如前述例子中，编码(**
  - **若电路加电后进入未**
    而无法进入工作状态
  - **若时序逻辑电路中的触发器具有预置功能，则可以通过预置处理，使电路进入正常的初始工作状态，从而避免“挂起”**
  - **可利用未用状态的无关项进行化简。但需对未用状态进行分析，以判定电路进入未用状态时能否在有限个时钟周期后进入工作状态。若能，且没有错误输出，则称电路为具有“自启动”能力；若不能或有错误输出，则需调整电路设计**

如果初始是未用状态，经过几个时钟能够进入工作状态，但进入的状态是错的（比如才输入一个1，就进入了连续输入两个1才应该进入的状态），那么就会导致后续输出错误。一般也会归入不能自启动的情况。

分析要点：使用未用状态化简后，能否“自启动”而不会发生“挂起”

3.4 电路分析（未用状态分析）

Y1Y0	Y1*Y0*/Z	
	X=0	X=1
00	00/0	01/0
01	11/0	01/0
11	00/0	01/1
10	00/0	01/1

◆未用状态分析举例

- 对于前述例子，利用未用状态10作为无关项化简后，得到：

$Y1^*=\overline{Y1}\cdot Y0\cdot\overline{X}$

$Y0^*=\overline{Y1}\cdot Y0+X$

$Z=Y1\cdot X$

$Y1^*=\overline{Y1}\cdot Y0\cdot\overline{X}$
$Y0^*=\overline{Y1}\cdot Y0+X$
$Z=Y1\cdot Y0\cdot X$

最终得到的函数

- 当处于未用状态Y1Y0=10时，根据上述逻辑表达式，可知：

若输入X=0，则次态=00，输出Z=0

若输入X=1，则次态=01，输出Z=1

- 分析是否具有“自启动”能力

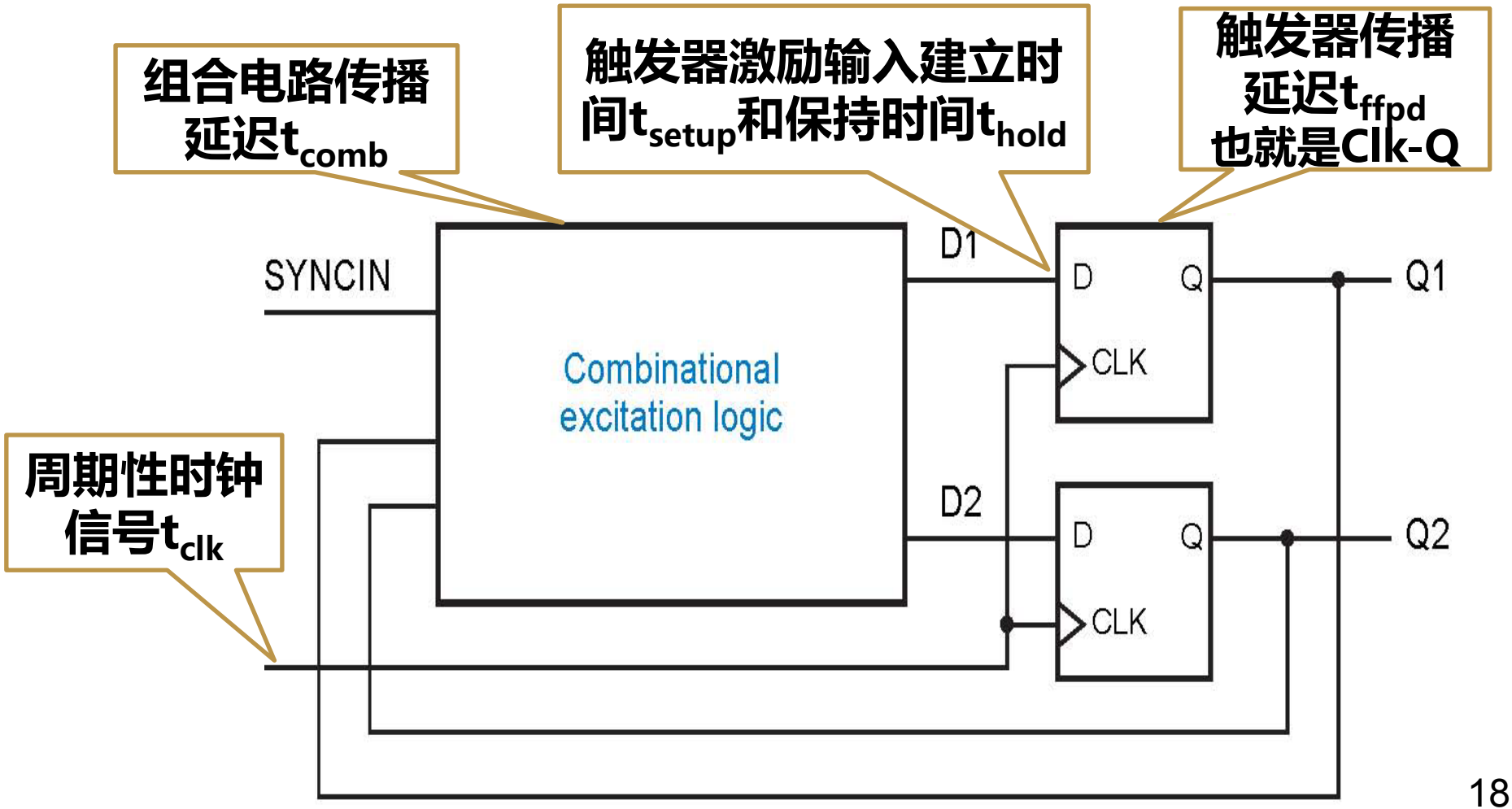
经过1个时钟周期就能进入正常工作状态，但是，当输入X=1时，输出Z=1，是错误输出，需调整输出模块的设计

- 重新设计逻辑电路中的输出模块（调整为化简前的表达式）

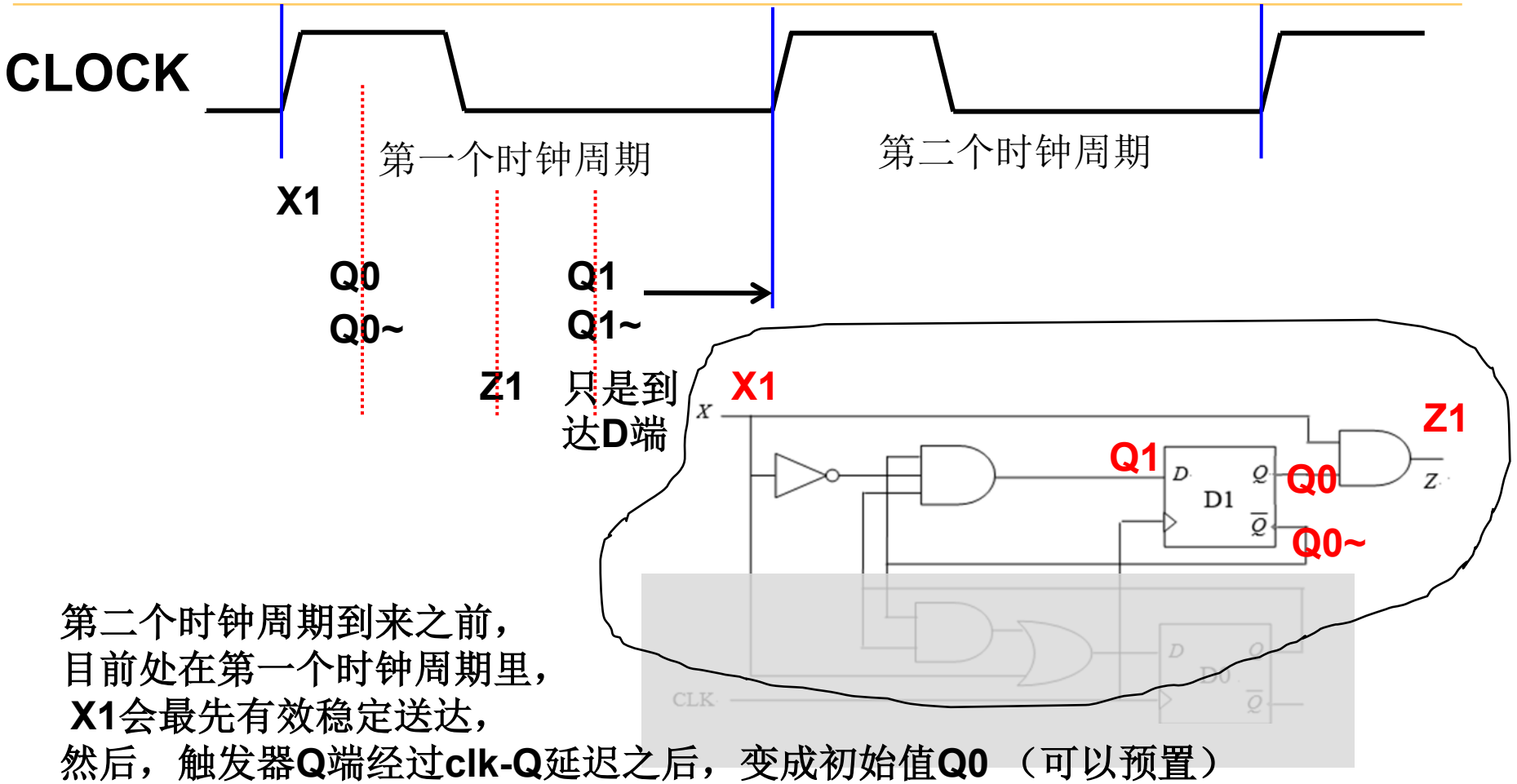
$Z=Y1\cdot Y0\cdot X$ 在未用状态10时，若输入X=1时，则输出Z=0。此时，不会发生误输出

3.4 电路分析（定时分析）

- 时序逻辑电路定时分析
  - 电路的工作频率与组合逻辑电路传输延迟、触发器建立和保持时间、触发器传输延迟等密切相关。

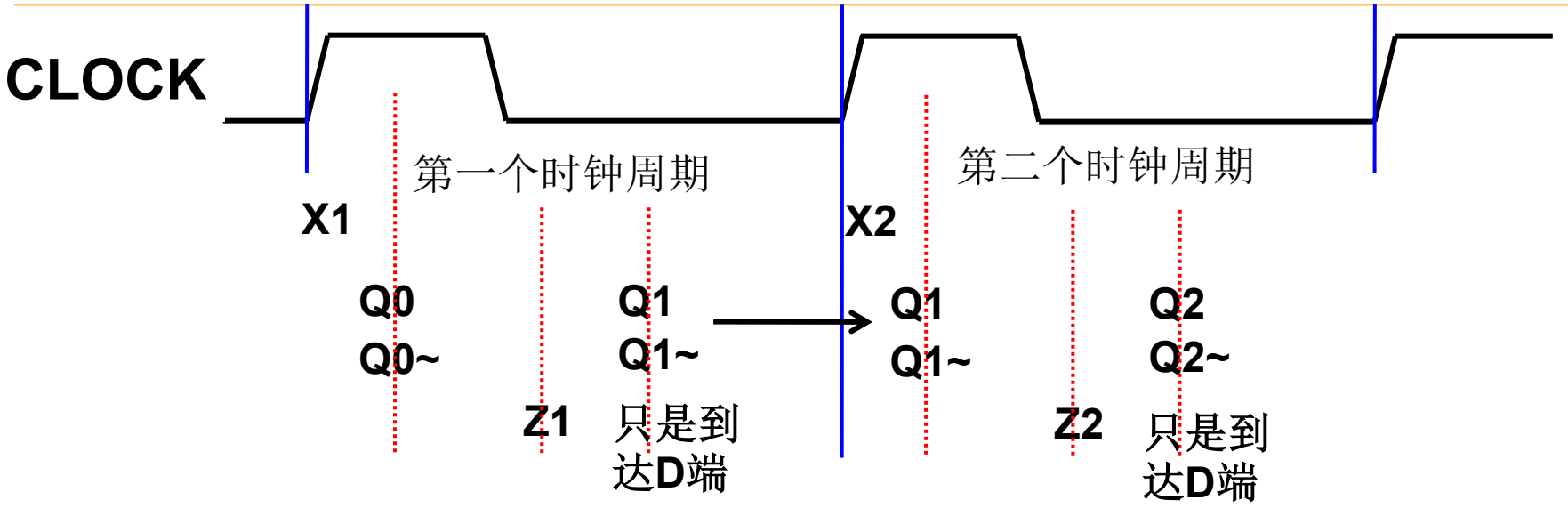


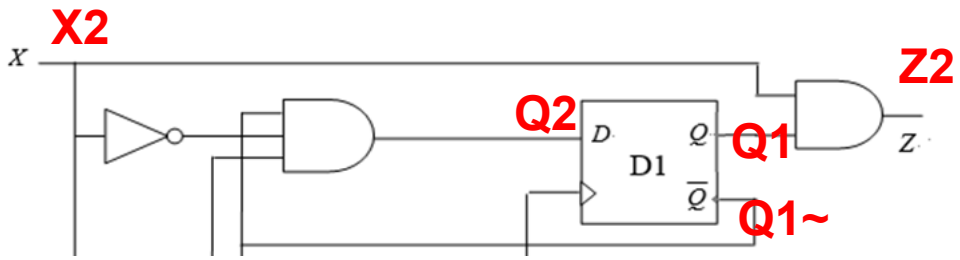
3.4 电路分析（定时分析）



第二个时钟周期到来之前，
目前处在第一个时钟周期里，
X1会最先有效稳定送达，
然后，触发器Q端经过clk-Q延迟之后，变成初始值Q0 （可以预置）
——过了次态运算逻辑延迟之后，基于Q0和X1，算出了Q1，并到达触发器D端
（第二个时钟周期到来之前，这个D端必须保持稳定超过setup时间）
——过了输出运算延迟之后，基于Q0和X1完成了Z1的计算，且该结果不影响其它过程，所以这个延迟只要小于时钟周期就行了，与次态计算无关

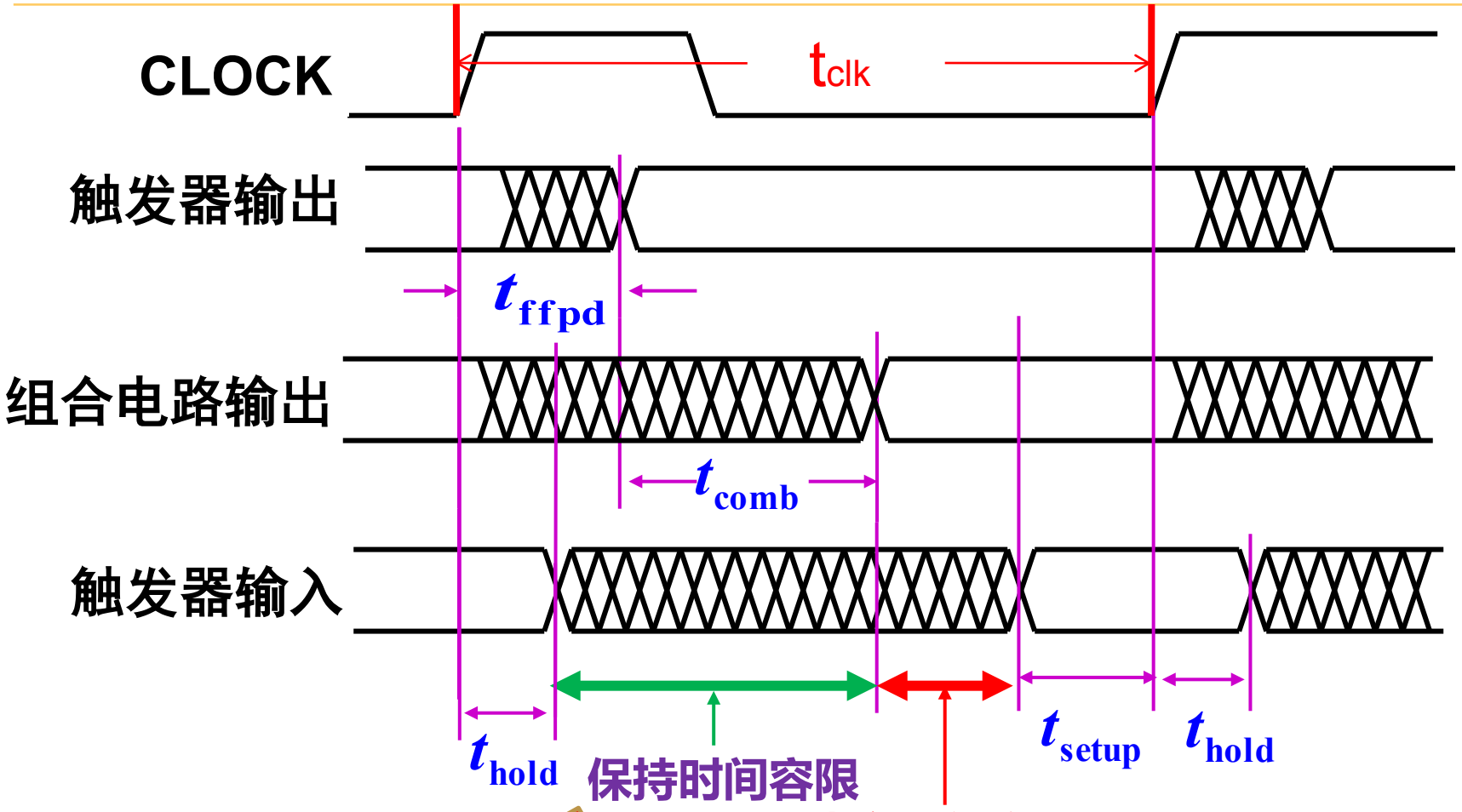
3.4 电路分析（定时分析）





第二个时钟周期到来之后，
X2会最先有效稳定送达，
算出的Q1保持不变，稳定在触发器的D端（hold时间）
然后，触发器Q端经过clk-Q延迟之后，变成Q1
——过了次态运算逻辑延迟之后，基于Q1和X2，算出了Q2，并到达触发器D端
（第3个时钟周期到来之前，这个D端必须保持稳定超过setup时间）
——过了输出运算延迟之后，基于Q1和X2完成了Z2的计算

3.4 电路分析（定时分析）



$$t_{ffpd(min)}+t_{comb(min)}-t_{hold}$$

$$t_{clk}-t_{ffpd(max)}-t_{comb(max)}-t_{setup}$$

时间容限指为保证电路正常工作某信号定时所允许的时间范围，都大于0

3.4 电路分析（定时分析）

- 建立时间容限=$t_{clk}-t_{ffpd(max)}-t_{comb(max)}-t_{setup}$，>0
- 保持时间容限=$t_{ffpd(min)}+t_{comb(min)}-t_{hold}$，>0

因此，得到**时序约束关系**：

(1) $t_{clk} > t_{ffpd(max)} + t_{comb(max)} + t_{setup}$

时钟周期不能小于这个值，但也不需要大过很多

为使触发器正常工作，必须保证时钟周期t_{clk}不能小于触发器 锁存延迟t_{ffpd} 加 次态信号经过激励逻辑延迟t_{comb} 加 触发器的建立时间t_{setup}。

(2) $t_{hold} < t_{ffpd(min)} + t_{comb(min)}$

为使触发器正常工作，必须保证外部激励信号在时钟有效边沿到来后的保持时间t_{hold}内能保持稳定不变。这就要求次态信号不能反馈太快。

第四讲 典型时序逻辑部件设计

- 计数器
- 寄存器和寄存器堆
- 移位寄存器

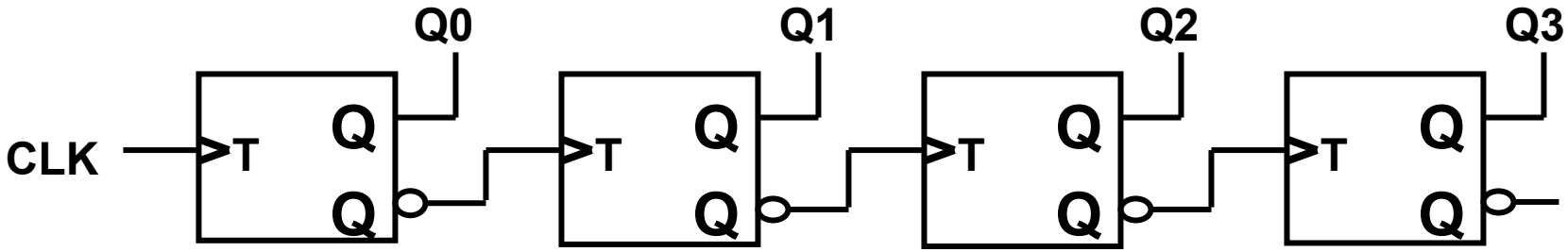
4.1 计数器

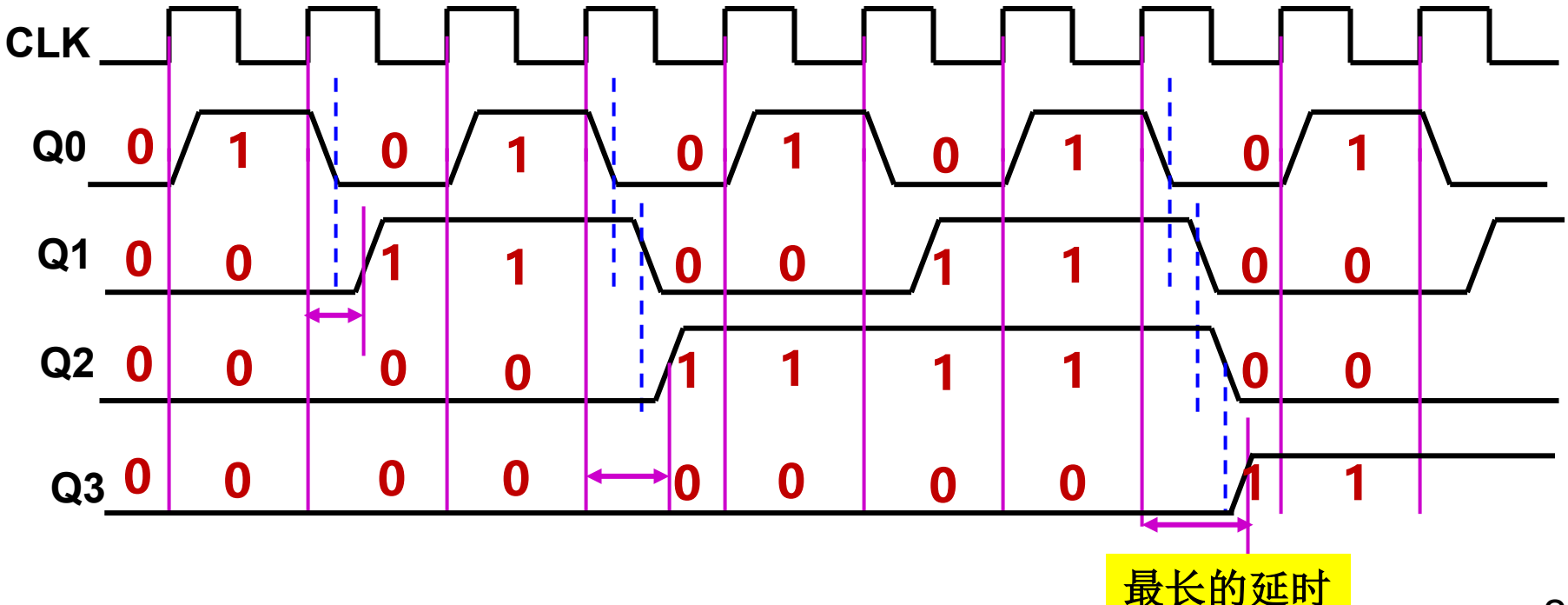
- 计数器是一种对外部信号进行总数统计的时序逻辑元件
- 一般从0开始计数，在达到最大计数值时输出一次计数完成信号，并重新开始计数
- 最大计数值为计数器的模
- 计数器的分类
  - 按时钟定时的使用方式：同步、异步
  - 按计数方式：加法、减法、可逆
  - 按编码方式：二进制、十进制BCD码、循环码
  - 按进位方式：行波（串行）进位、并行进位

4.1 计数器——异步行波加法计数器

- 用T 触发器实现（上升沿触发），激励输入串行传递，每个时钟周期传递一次。

Q_{i+1}总是在Q_i由1变0时开始改变





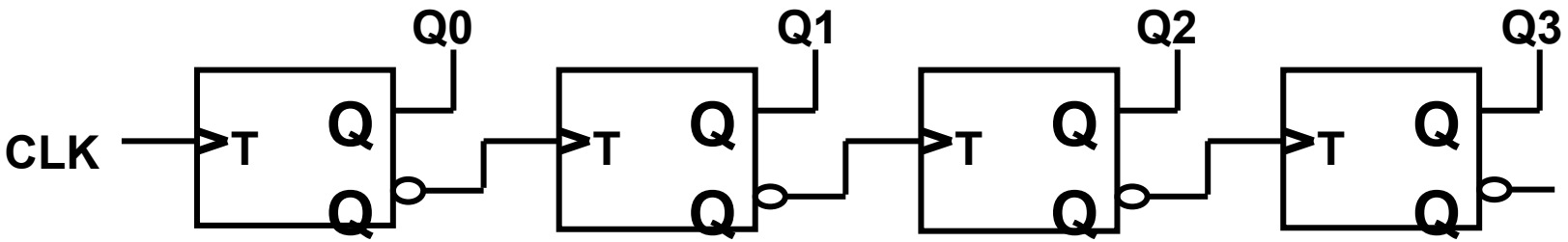
4.1 计数器——异步行波加法计数器

Q0每个时钟转1次，Q1每2个转1次，
Q2每4个转1次，Q3每8个转1次

- $Q_3Q_2Q_1Q_0$转换过程为（注意：最高位为Q3）

  0000→0001→0010→0011→0100→

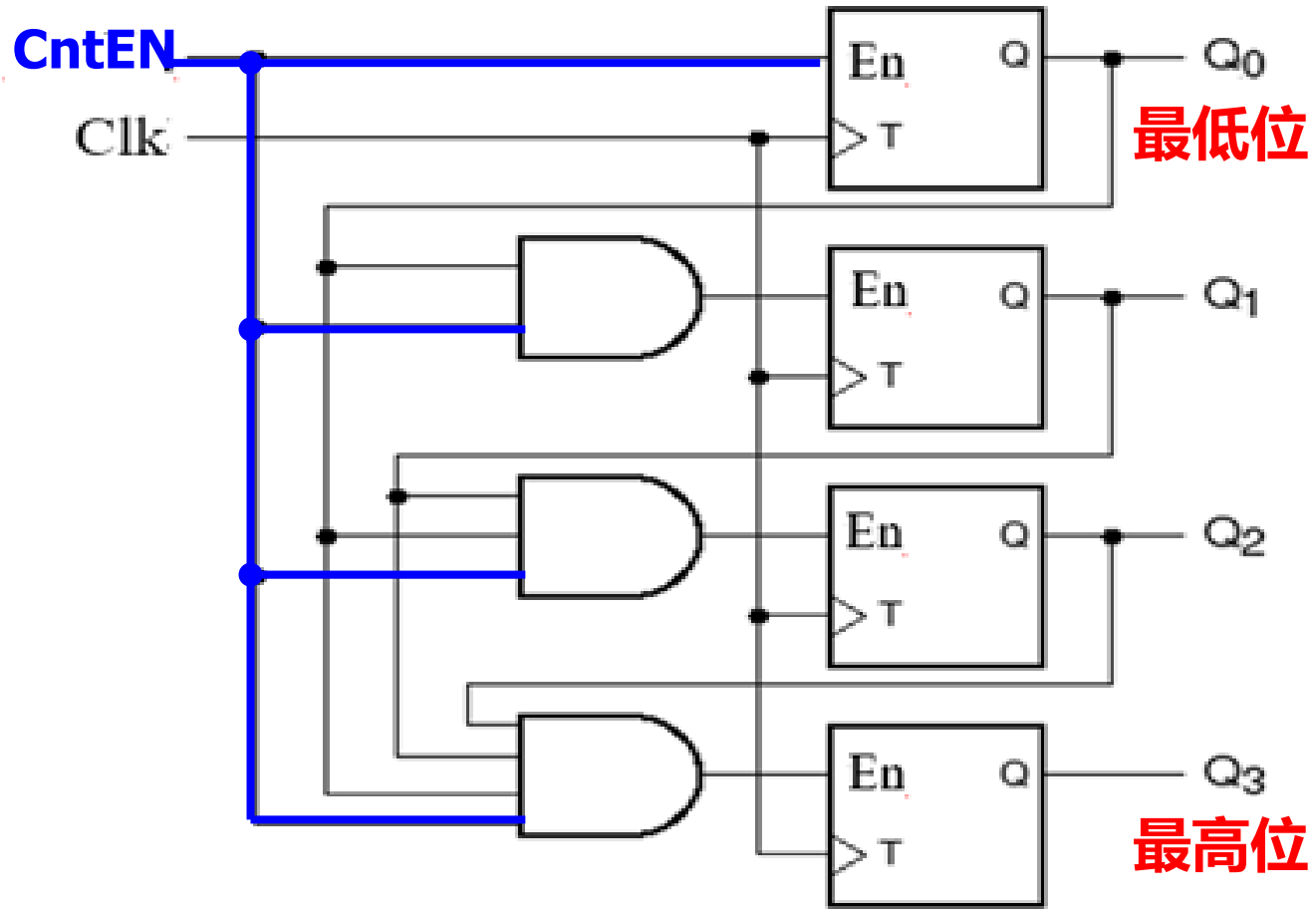
  0101→0110→ … … →1111



当编码为1111时，下个时钟到达后，经过最长的延时（4个锁存延迟），又回到编码0000

4.1 计数器——同步并行加法计数器

- 所有触发器共用同一个时钟信号，
- 在时钟信号边沿到达后，所有触发器的输出同时发生变化。
- 带使能端EN的T触发器，上升沿触发



每个Qi一定会变化吗？

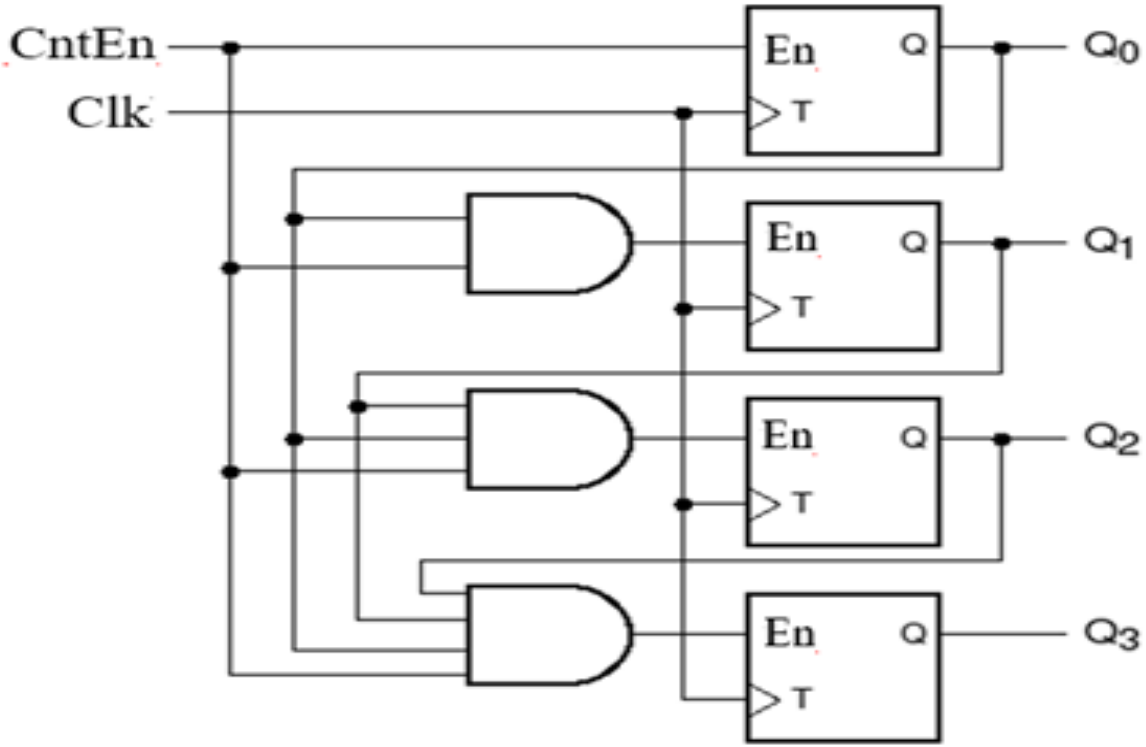
——不一定。也不应该每次都变化：用EN信号（低位Q和CntEN相与的结果）来控制

4.1 计数器——同步并行加法计数器

- 计数器的状态编码$Q_3Q_2Q_1Q_0$从0000开始，转换过程为
  0000→0001→0010→……→1101→1110→1111

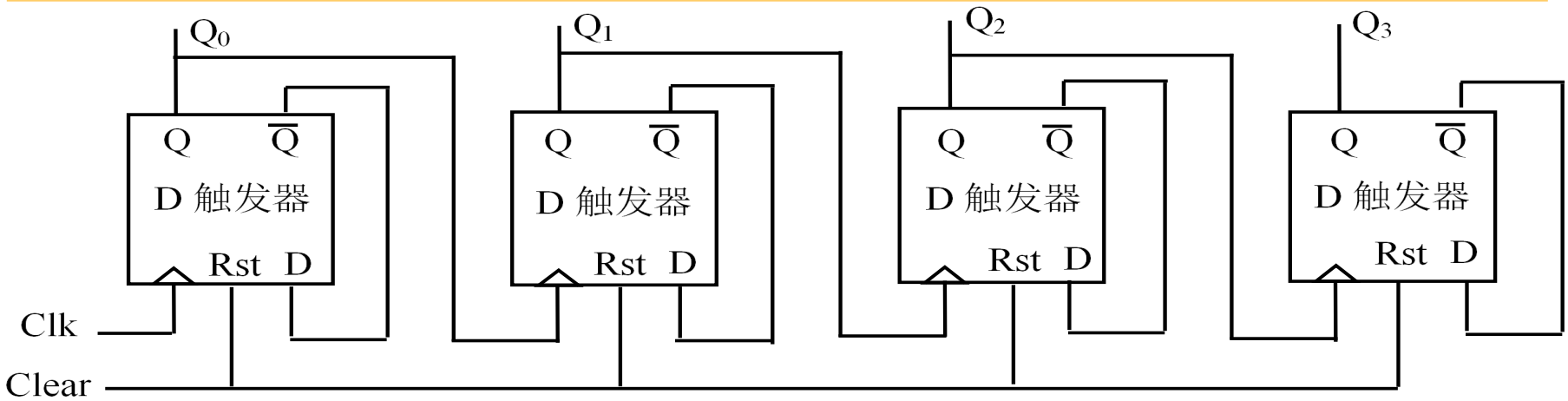
CntEN有效时，每个时钟Q_0都会发生状态改变；

对于Q_1、Q_2和Q_3，只有在其**所有低位状态都是1**的情况下，**下个时钟边沿到来**后才会发生状态反转。



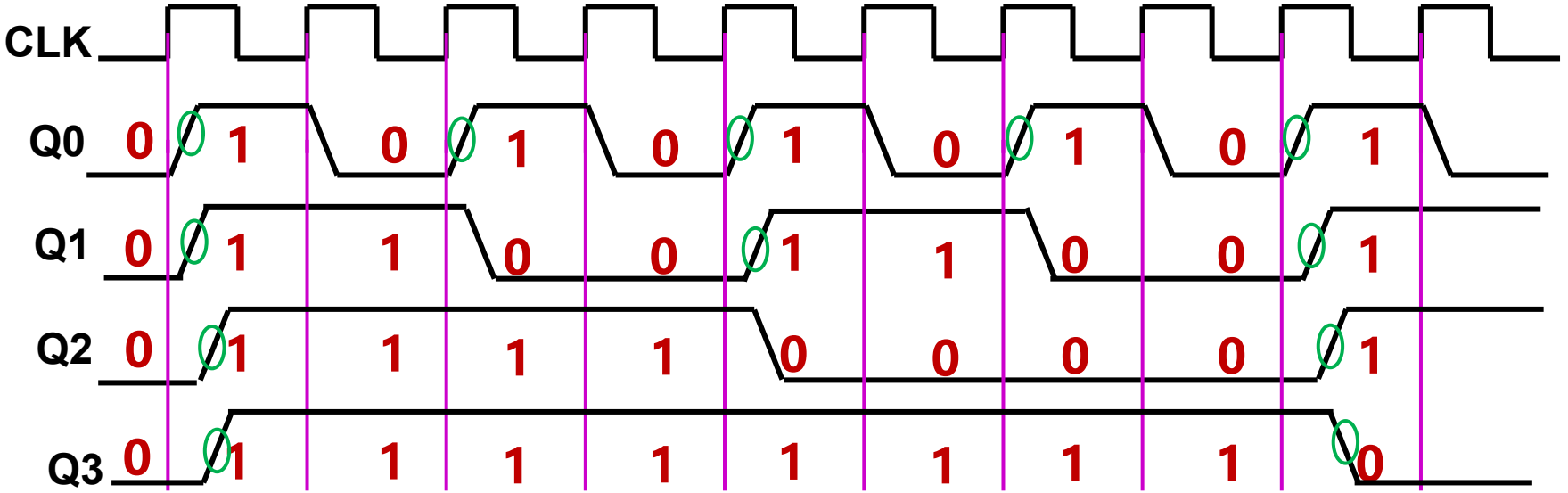
当编码为1111时，只要经过一个与门+锁存延时，就可回到编码0000，比行波（串行）加法计数器快

4.1 计数器 ◆二进制异步行波减法计数器



Q_{i+1}总是在Q_i由0变1时 (上升沿) 发生改变

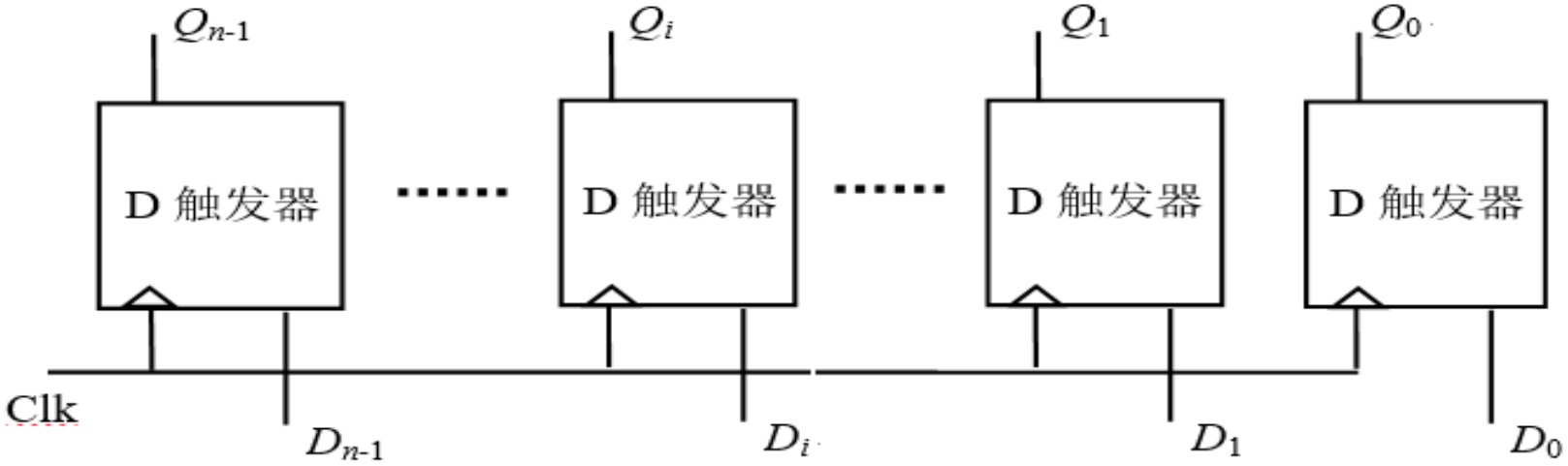
Clear清0，初态为0000

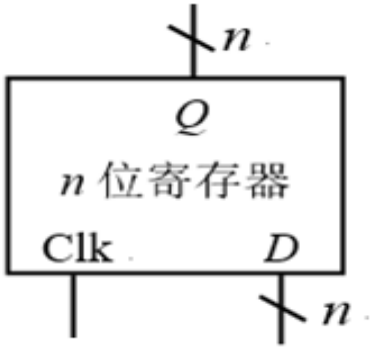


Q0每个时钟转1次，Q1每2个转1次，Q2每4个转1次，Q3每8个转1次

4.2 寄存器

- 寄存器是用来暂存信息的逻辑部件
- 寄存器可直接由若干个触发器组成



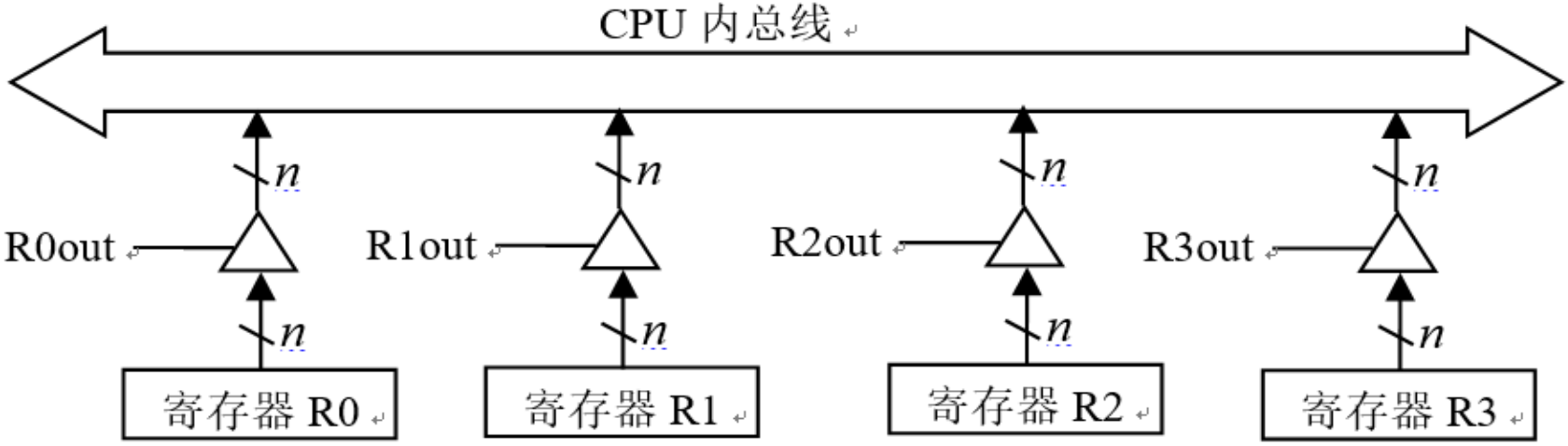


每个时钟到来，存放内容都会改变吗？
——不一定。
——可以加使能端，或者让D输入不变

4.2 寄存器

- **寄存器通过三态门和总线互连**

任何时刻至多只能一个Rout有效



寄存器是一种时序逻辑电路，但只包含存储电路

在没有新的**CLK**脉冲来之前，寄存器一定能保存原有内容不变

附加一些控制电路就能够实现：置零、保持（ **CLK**信号到达时触发器不随**D**端的输入信号而改变状态，保持原来的状态不变）等功能。

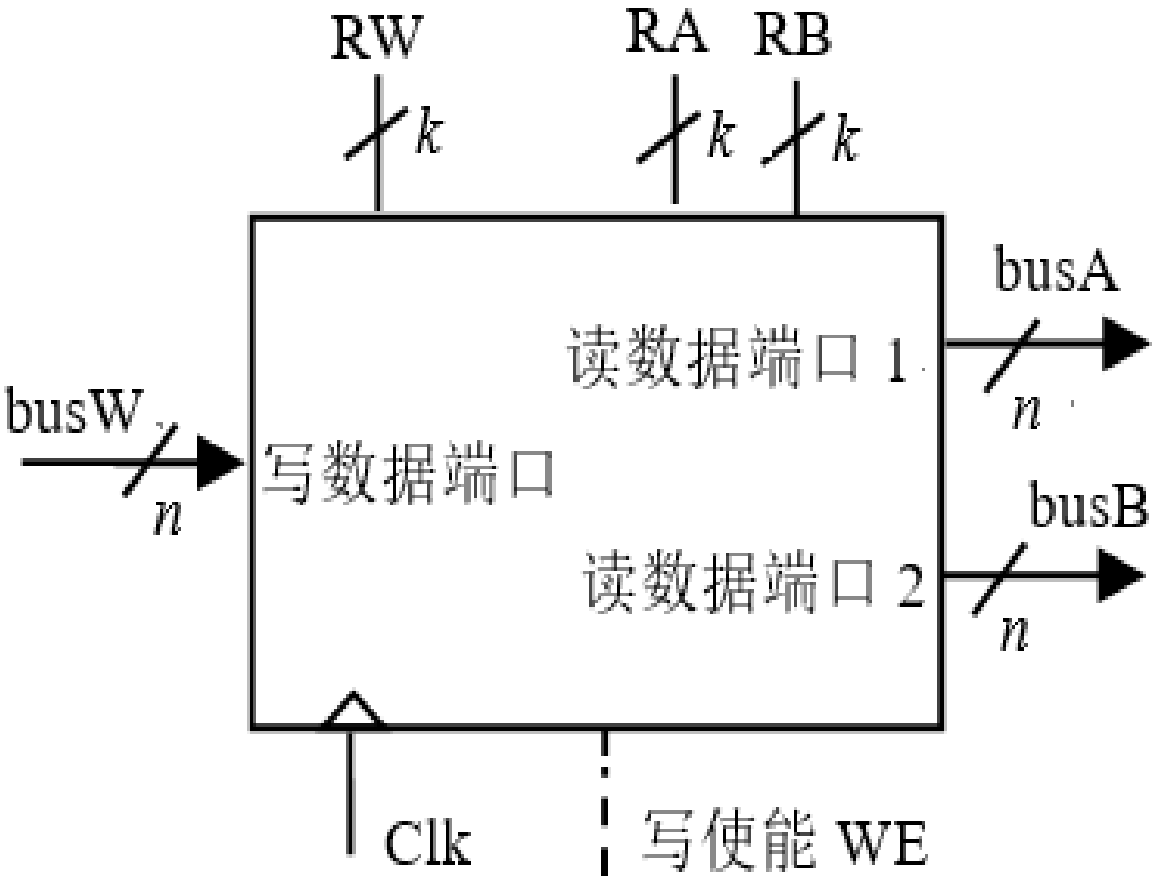
接收数据时所有各位信息都是同时输入的，而且触发器中的数据是并行地出现在输出端的。

4.2 寄存器堆

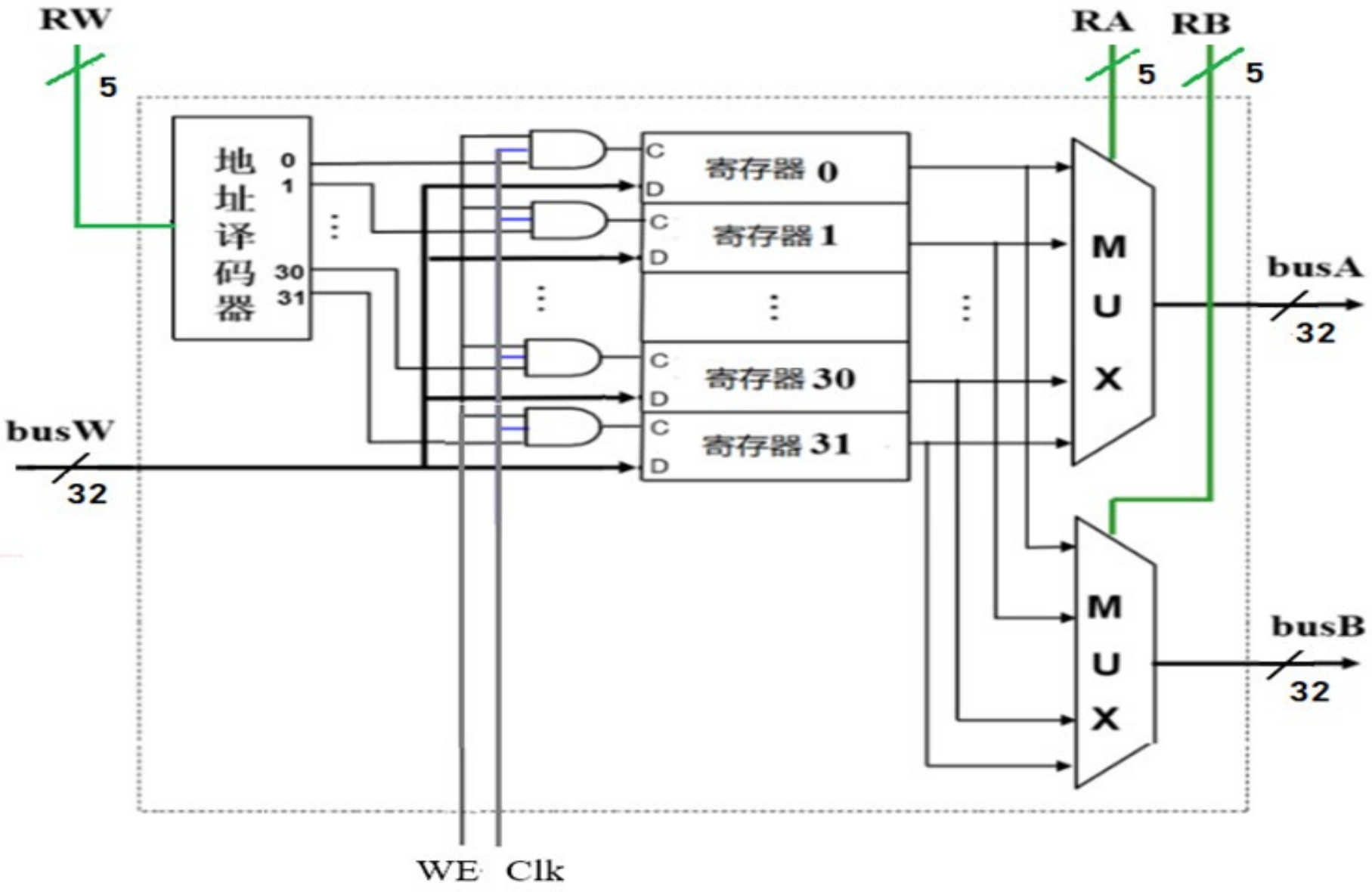
- 寄存器堆(Register File)：CPU内部用于暂存指令执行过程中的中间数据，也称通用寄存器组(General Purpose Register set，GPRs)
- 由许多寄存器组成，每个寄存器有一个编号，CPU可对指定编号的寄存器进行读写

寄存器堆中共有2^k个寄存器，每个寄存器位数为n，RA和RB分别是读口1和读口2的寄存器编号，RW是写口的寄存器编号

读操作属于组合逻辑操作;
写操作属于时序逻辑操作，需要时钟信号Clk和写使能信号WE的控制



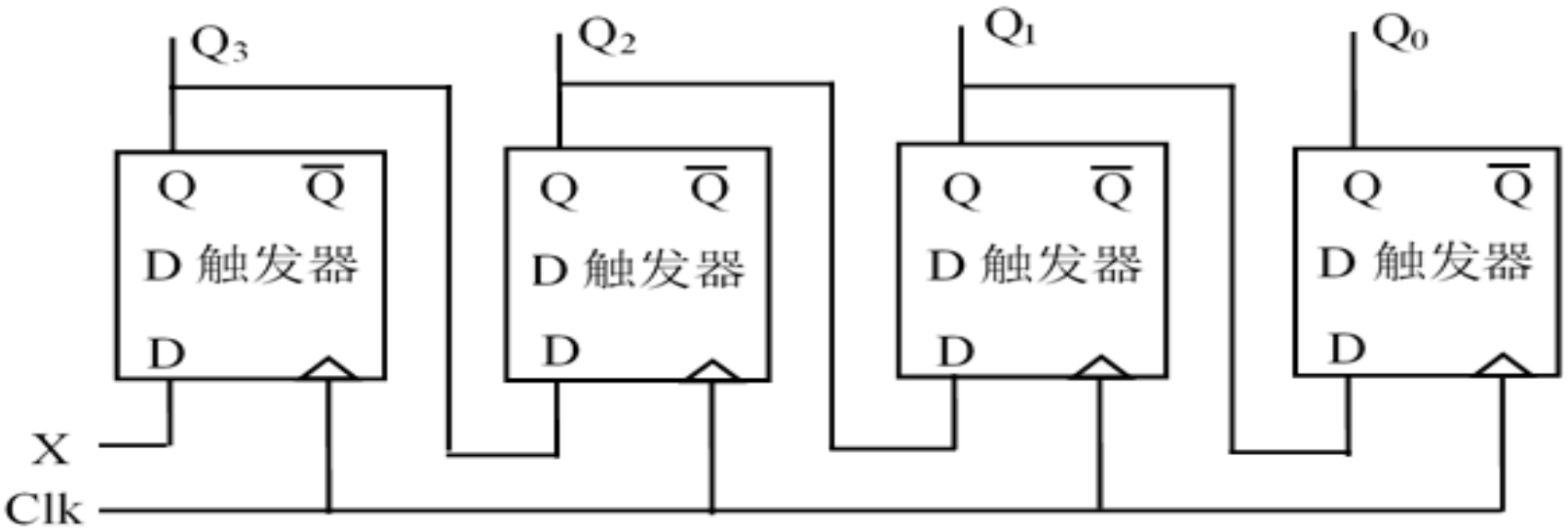
4.2 寄存器堆内部结构


RW
5
RA
RB
5
5
地址译码器
0
1
30
31
C
D
寄存器 0
寄存器 1
寄存器 30
寄存器 31
M
U
X
busA
32
busW
32
busB
32
WE
Clk

4.3 移位寄存器

- 移位寄存器
  - 能够实现暂存信息的左移或右移功能，通常由时钟信号控制

例如：4个D触发器可构成一个右移寄存器

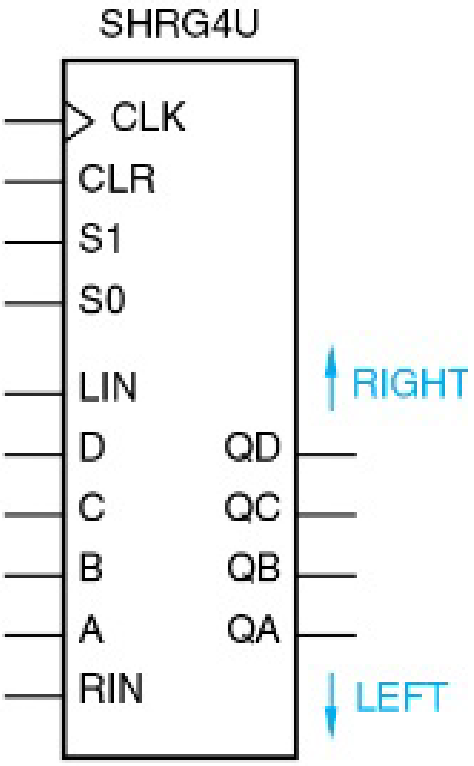


假设初始状态编码$Q_3Q_2Q_1Q_0$=0000，X输入为序列10011011

则$Q_3Q_2Q_1Q_0$的输出编码依次为：0000、1000、0100、0010、1001、1100、0110、1011、1101

4.3 移位寄存器

- 4位通用移位寄存器，如74X194
  - 具有数据左移、数据右移、数据保持和数据载入功能
  - 用CLR，S1，S0这三个信号的排列组合来表示不同功能



Function	Inputs			Next state			
	CLR	S1	S0	QA*	QB*	QC*	QD*
Clear	1	x	x	0	0	0	0
Hold	0	0	0	QA	QB	QC	QD
Shift right	0	0	1	RIN	QA	QB	QC
Shift left	0	1	0	QB	QC	QD	LIN
Load	0	1	1	A	B	C	D

左移从QD移到QA向下移↓　右移从QA移到QD向上移↑

4.3 移位寄存

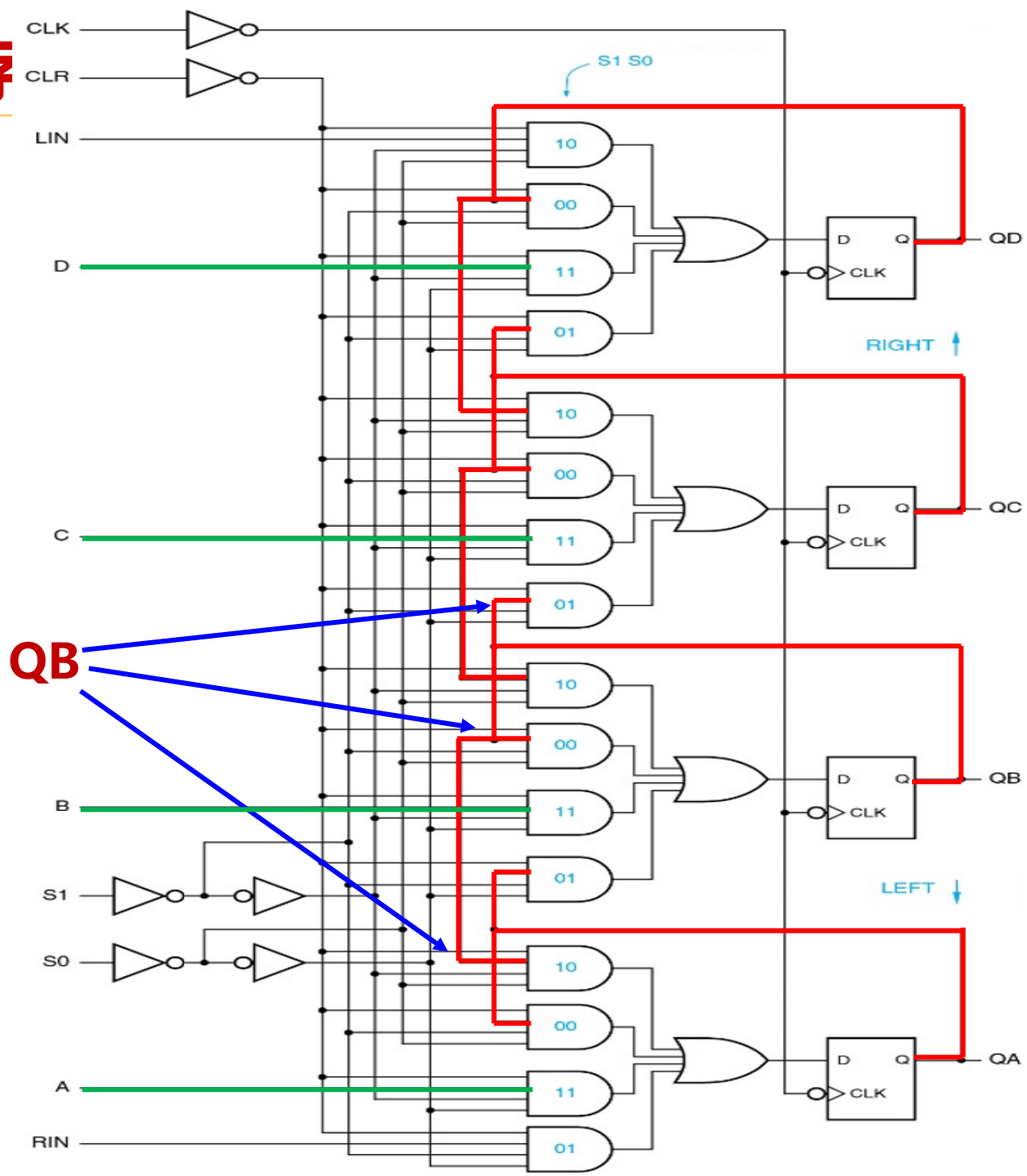
- 4位通用移位寄存器结构图

S1S0=00：保持

S1S0=01：上(右)移

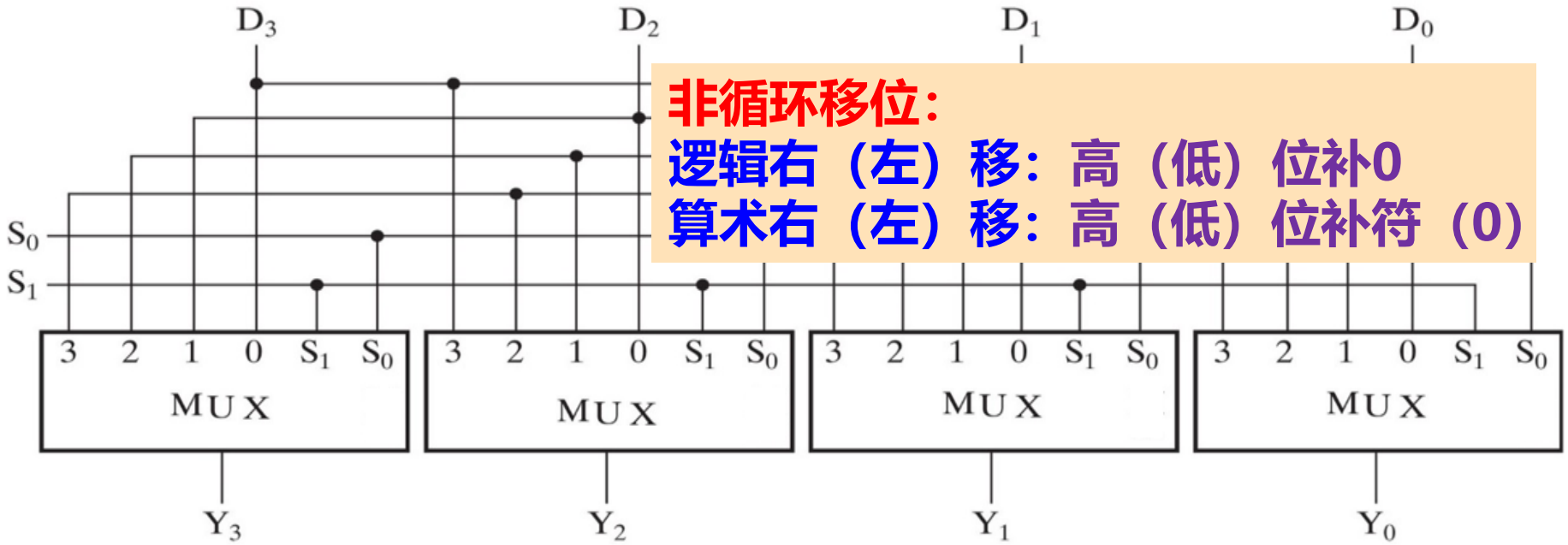
S1S0=10：下(左)移

S1S0=11：加载



4.3 桶形移位器（无寄存功能）

- 一次移动多位。组合逻辑电路，采用大量多路选择器实现



非循环移位：
逻辑右（左）移：高（低）位补0
算术右（左）移：高（低）位补符（0）

Function Table for 4-Bit Barrel Shifter

Select		Output				
S_1	S_0	Y_3	Y_2	Y_1	Y_0	Operation
0	0	D_3	D_2	D_1	D_0	No rotation
0	1	D_2	D_1	D_0	D_3	Rotate one position
1	0	D_1	D_0	D_3	D_2	Rotate two positions
1	1	D_0	D_3	D_2	D_1	Rotate three positions

只有移位功能

没有寄存功能

第4章总结

- 时序逻辑电路不仅依赖当前输入，还依赖电路当前的状态
- 可用时序逻辑电路实现有限状态机。有Mealy型和Mooer型两类
- 可用状态图或状态表描述有限状态机，圈表示状态，有向边表示输入/输出
- 锁存器(电平触发): SR锁存器(设置标志)、D锁存器(锁存数据D)
- 触发器(时钟信号clk边沿触发)：D触发器(寄存器)、T触发器(计数或分频)
- 时序电路设计：功能分析-状态图-状态化简和编码-逻辑表达式-画图-评价
- 电路分析：未用状态分析(挂起/无法自启动)
  定时分析(clk-Q时间、时钟周期、setup时间、hold时间)
- 典型组合逻辑部件：计数器、寄存器/通用寄存器组、移位寄存器
- 计数器：同步/异步、加1/减1、行波（串行）进位/并行进位
- 寄存器：由n个D触发器构成，同时由时钟信号clk定时
- 通用寄存器组：两个读口(组合逻辑)；一个写口(时序逻辑，clk和写使能)
- 移位寄存器（时序逻辑）：每次固定左移或右移1位（或几位）
- 桶型移位器（组合逻辑）：移位位数可变，用大量多路选择器实现

作业：习题4、5、6、9、11、12。提交截止日期：11月1日24:00