

可以用三个方程组来描述:

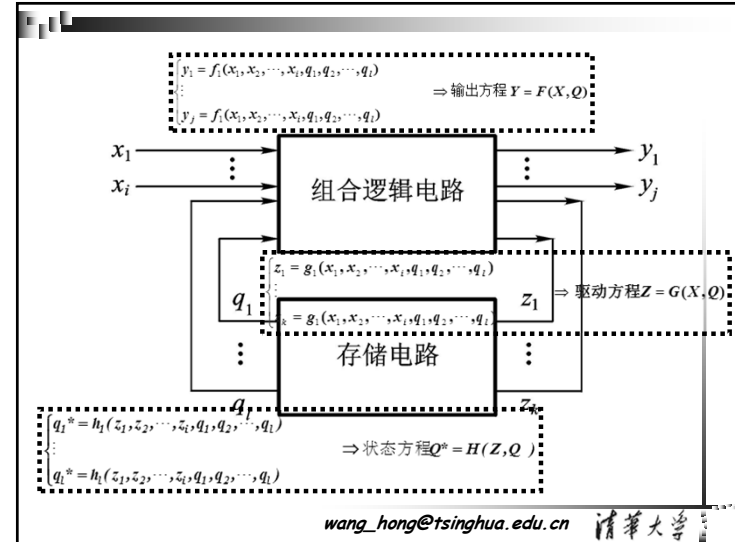
$$\begin{cases} y_1 = f_1(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \\ \vdots \\ y_j = f_j(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \end{cases} \Rightarrow \text{输出方程 } Y = F(X, Q)$$

$$\begin{cases} z_1 = g_1(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \\ \vdots \\ z_k = g_k(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \end{cases} \Rightarrow \text{驱动方程 } Z = G(X, Q)$$

$$\begin{cases} q_1^* = h_1(z_1, z_2, \dots, z_k, q_1, q_2, \dots, q_l) \\ \vdots \\ q_l^* = h_l(z_1, z_2, \dots, z_k, q_1, q_2, \dots, q_l) \end{cases} \Rightarrow \text{状态方程 } Q^* = H(Z, Q)$$

wang_hong@tsinghua.edu.cn

清华大学



wang_hong@tsinghua.edu.cn

清华大学

三、时序电路的分类

1. 同步时序电路与异步时序电路

同步: 存储电路中所有触发器的时钟使用统一的clk, 状态变化发生在同一时刻

异步: 没有统一的clk, 触发器状态的变化有先有后

2. Mealy型和Moore型

Mealy型: $Y = F(X, Q)$ 与 X 、 Q 有关

Moore型: $Y = F(Q)$ 仅取决于电路状态

wang_hong@tsinghua.edu.cn

清华大学

6.2 时序电路的分析方法

6.2.1 同步时序电路的分析方法

分析: 找出给定时序电路的逻辑功能
即找出在输入和CLK作用下, 电路的次态和输出。

一般步骤:

①从给定电路写出存储电路中每个触发器的驱动方程 (输入的逻辑式), 得到整个电路的驱动方程。

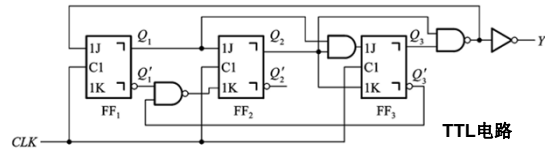
②将驱动方程代入触发器的特性方程, 得到状态方程。

③从给定电路写出输出方程。

wang_hong@tsinghua.edu.cn

清华大学

例:



1. 写驱动方程:

$$\begin{cases} J_1 = (Q_2 Q_3)', & K_1 = 1 \\ J_2 = Q_1, & K_2 = (Q_1' Q_3')' \\ J_3 = Q_1 Q_2, & K_3 = Q_2 \end{cases}$$

3. 输出方程

$$Y = Q_2 Q_3$$

2. 代入JK触发器的特性方程 $Q^* = JQ' + K'Q$, 得状态方程:

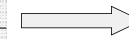
$$\begin{cases} Q_1^* = (Q_2 Q_3)' \cdot Q_1' \\ Q_2^* = Q_1 Q_2' + Q_1' Q_3' Q_2 \\ Q_3^* = Q_1 Q_2 Q_3' + Q_2' Q_3 \end{cases}$$

wang_hong@tsinghua.edu.cn

清华大学

一、状态转换表

Q_3	Q_2	Q_1	Q_3'	Q_2'	Q_1'	Y
0	0	0	0	0	1	0
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	1	0	0	0	1



CLK	Q_3	Q_2	Q_1	Y
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	0	0	0	0
0	1	1	1	1
1	0	0	0	0

$$Y = Q_2 Q_3$$

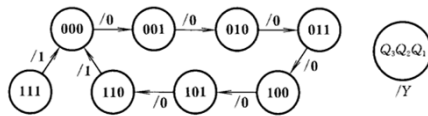
$$\begin{cases} Q_1^* = (Q_2 Q_3)' \cdot Q_1' \\ Q_2^* = Q_1 Q_2' + Q_1' Q_3' Q_2 \\ Q_3^* = Q_1 Q_2 Q_3' + Q_2' Q_3 \end{cases}$$

wang_hong@tsinghua.edu.cn

清华大学

二、状态转换图

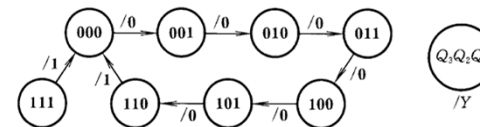
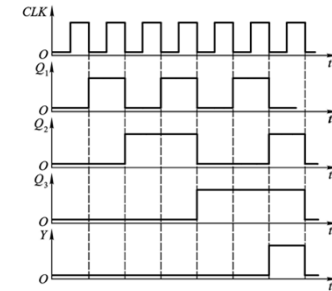
CLK	Q_3	Q_2	Q_1	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	0	0	0	0
0	1	1	1	1
1	0	0	0	0



wang_hong@tsinghua.edu.cn

清华大学

三、时序图



wang_hong@tsinghua.edu.cn

清华大学

例:

(1) 驱动方程: $\begin{cases} D_1 = Q_1' \\ D_2 = A \oplus Q_1 \oplus Q_2 \end{cases}$

(2) 状态方程: $\begin{cases} Q_1^* = Q_1' \\ Q_2^* = A \oplus Q_1 \oplus Q_2 \end{cases}$

(3) 输出方程: $Y = [(A'Q_1Q_2)' \cdot (AQ_1'Q_2)']' = A'Q_1Q_2 + AQ_1'Q_2'$

wang_hong@tsinghua.edu.cn 清华大学

(4) 列状态转换表:

$Q_2^* \backslash Q_1^* \backslash A$	00	01	10	11
0	01/0	10/0	11/0	00/1
1	11/1	00/0	01/0	10/0

状态方程: $\begin{cases} Q_1^* = Q_1' \\ Q_2^* = A \oplus Q_1 \oplus Q_2 \end{cases}$

输出方程: $Y = A'Q_1Q_2 + AQ_1'Q_2'$

(5) 状态转换图

wang_hong@tsinghua.edu.cn 清华大学

***6.2.3 异步时序逻辑电路的分析方法**

各触发器的时钟不同

例:

TTL 电路

$Q_2^* = Q_2' \cdot clk_2$

$Q_1^* = Q_3'Q_1' \cdot clk_1$

wang_hong@tsinghua.edu.cn 清华大学

6.3 若干常用的时序逻辑电路

6.3.1 寄存器和移位寄存器

一、寄存器

① 用于寄存一组二值代码，N位寄存器由N个触发器组成，可存放一组N位二值代码。

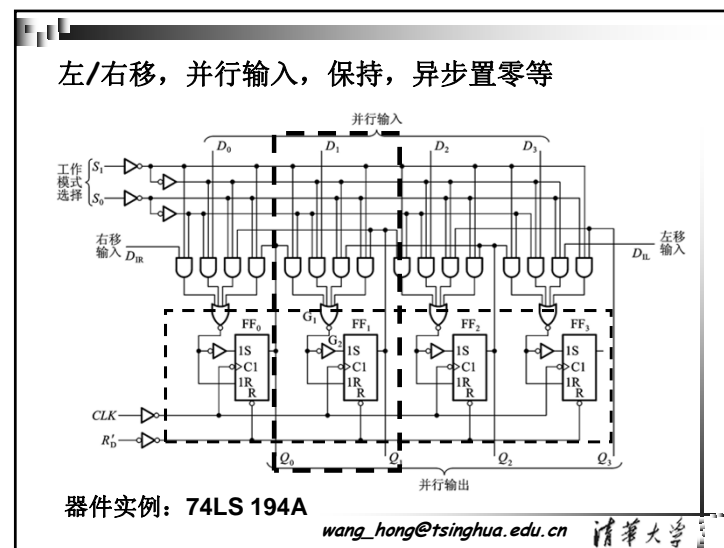
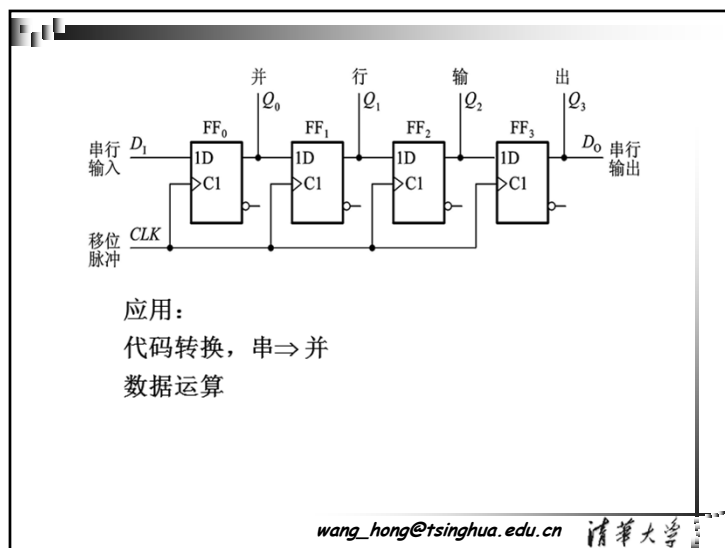
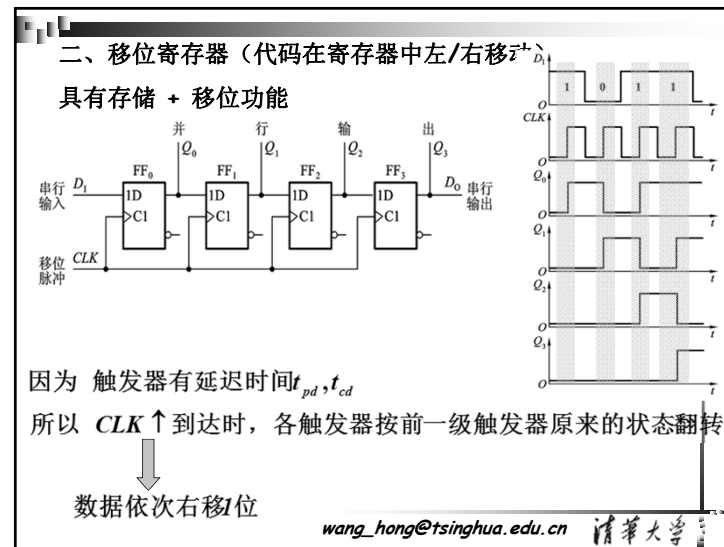
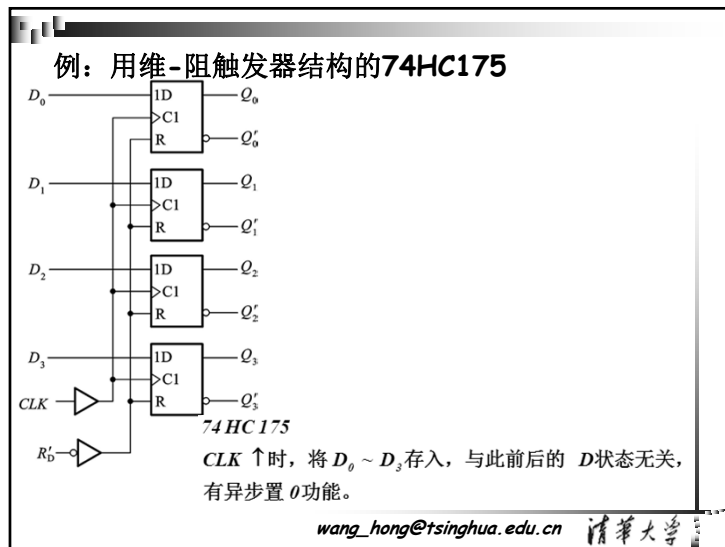
② 只要求其中每个触发器可置1，置0。

例1:

74 HC 75

clk 高电平期间 Q 随 D 改变

wang_hong@tsinghua.edu.cn 清华大学



$S_{Q1} = S'_1 S'_0 \cdot Q_1 + S'_1 S_0 \cdot Q_0 + S_1 S'_0 Q_2 + S_1 S_0 Q_1$
 $R_{Q1} = S'_{Q1}$
 $Q_1^* = S_{Q1}$

通过控制 S_1, S_0 就可以选择 194 的工作状态

R_D	S_1	S_0	工作状态
0	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

wang_hong@tsinghua.edu.cn 清华大学

扩展应用 (4位 → 8位)

并行数据输入

右移 串行输入

左移 串行输入

并行数据输出

wang_hong@tsinghua.edu.cn 清华大学

6.3.2 计数器

- 用于计数、分频、定时、产生节拍脉冲等
- 分类：按时钟分，同步、异步

按计数过程中数字增减分，加、减和可逆

按计数器中的数字编码分，二进制、二-十进制和

按计数容量分，十进制，六十进制...

wang_hong@tsinghua.edu.cn 清华大学

一、同步计数器

1. 同步二进制计数器

① 同步二进制加法计数器

原理：根据二进制加法运算规则可知：在多位二进制数末位加1，若第 i 位以下皆为1时，则第 i 位应翻转。

由此得出规律，若用T触发器构成计数器，则第 i 位触发器输入端 T_i 的逻辑式应为：

$$T_i = Q_{i-1} Q_{i-2} \dots Q_0$$

$$T_0 \equiv 1$$

计数脉冲 CLK

$T_0 = 1$

T_1

T_2

T_3

C

wang_hong@tsinghua.edu.cn 清华大学

器件实例：74161

CLK	R'_D	LD'	EP	ET	工作状态
X	0	X	X	X	置 0 (异步)
	1	0	X	X	预置数 (同步)
X	1	1	0	1	保持 (包括C)
X	1	1	X	0	保持 (C=0)
	1	1	1	1	计数

wang_hong@tsinghua.edu.cn

清华大学

CLK	R _D '	LD'	EP	ET	工作状态
	0	X	X	X	置 0 (异步)
	1	0	X	X	预置数 (同步)
	1	1	0	1	保持 (包括C)
	1	1	X	0	保持 (C=0)
	1	1	1	1	计数

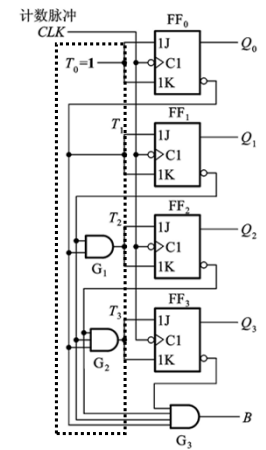
wang_hong@tsinghua.edu.cn

清华大学

原理：根据二进制减法运算规则可知：在多位二进制数末位减1，若第*i*位以下皆为0时，则第*i*位应翻转。

$$T_i = Q'_{i-1}Q'_{i-2}...Q'_0$$

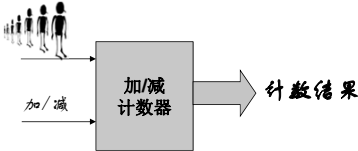
$$T_0 \equiv 1$$



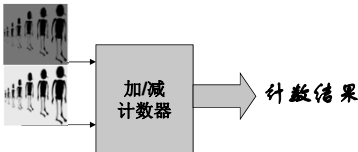
wang_hong@tsinghua.edu.cn

清华大学

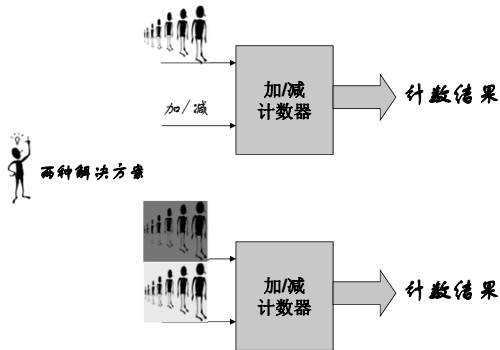
③同步加减计数器



两种解决方案



wang_hong@tsinghua.edu.cn 清华大学



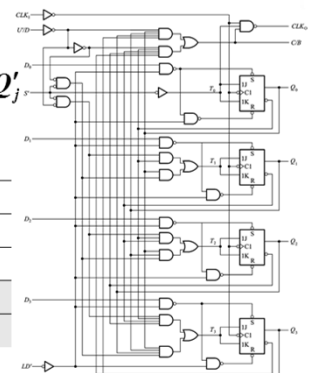
wang_hong@tsinghua.edu.cn

清华大学

加/减脉冲用同一输入端，
由加/减控制线的高低电平决定加/减
器件实例：74LS191（用T触发器）

$$\begin{cases} T_i = (U'/D)' \prod_{j=0}^{i-1} Q_j + (U'/D) \prod_{j=0}^{i-1} \\ T_0 = I \end{cases}$$

CLK_I	S'	LD'	U'/D	工作状态
X	1	1	X	保持
X	X	0	X	预置数(异步)
	0	1	0	加计数
	0	1	1	减计数



wang_hong@tsinghua.edu.cn

清华大学

b. 双时钟方式
 器件实例: 74LS193 (采用T'触发器, 即T=1)

$$CLK_i = CLK_U \prod_{j=0}^{i-1} Q_j + CLK_D \prod_{j=0}^{i-1} Q'_j$$

$$CLK_0 = CLK_U + CLK_D$$

$$CLK_2 = CLK_U Q_1 Q_0 + CLK_D Q'_1 Q'_0$$

wang_hong@tsinghua.edu.cn 清华大学

2. 同步十进制计数器
 ①加法计数器
 基本原理: 在四位二进制计数器基础上修改,
 当计到1001时, 则下一个CLK电路状态回到0000。

wang_hong@tsinghua.edu.cn 清华大学

2. 同步十进制计数器
 ①加法计数器
 基本原理: 在四位二进制计数器基础上修改,
 当计到1001时, 则下一个CLK电路状态回到0000。

wang_hong@tsinghua.edu.cn 清华大学

2. 同步十进制计数器
 ①加法计数器
 基本原理: 在四位二进制计数器基础上修改, 当计到1001时, 则下一个CLK电路状态回到0000。

$$T_0 = 1$$

$$T_1 = Q_0 \Rightarrow Q_0 Q_1$$

$$T_2 = Q_1 Q_2$$

$$T_3 = Q_2 Q_1 Q_0 \Rightarrow Q_2 Q_1 Q_0 + Q_3 Q_0$$

清华大学

器件实例：74 160

74160 Pinout and Truth Table:

CLK	R ₀	LD'	EP	ET	工作状态
X	0	X	X	X	置 0 (异步)
1	0	X	X	X	预置数 (同步)
X	1	1	0	1	保持 (包括C)
X	1	1	X	0	保持 (C=0)
1	1	1	1	1	计数

wang_hong@tsinghua.edu.cn 清华大学

②减法计数器
基本原理：对二进制减法计数器进行修改，在0000时减“1”后跳变为1001，然后按二进制减法计数就行了。

计数脉冲 CLK

$T_0 = 1$

$T_1 = Q_0 \Rightarrow Q_0'(Q_3Q_2Q_1)'$

$T_2 = Q_1Q_0 \Rightarrow Q_1Q_0'(Q_3Q_2Q_1)'$

$T_3 = Q_2Q_1Q_0$

0000 → 1111
0000 → 1001

wang_hong@tsinghua.edu.cn 清华大学

$T_0 = 1$
 $T_1 = Q_0'(Q_3Q_2Q_1)'$
 $T_2 = Q_1Q_0'(Q_3Q_2Q_1)'$
 $T_3 = Q_2Q_1Q_0$

能自启动

wang_hong@tsinghua.edu.cn 清华大学

③十进制可逆计数器
基本原理一致，电路只用到0000~1001的十个状态

实例模块（器件）
 单时钟：74190, 168
 双时钟：74192

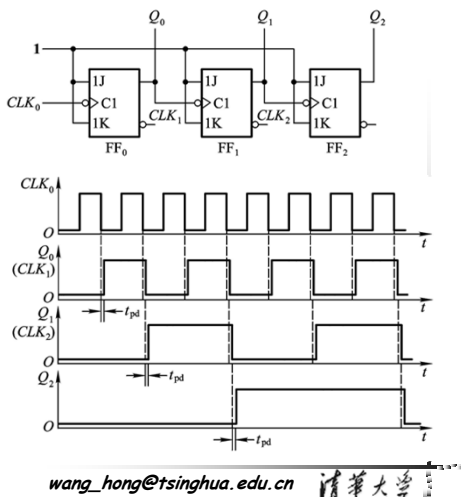
wang_hong@tsinghua.edu.cn 清华大学

二. 异步计数器

1. 二进制计数器

①异步二进制加法计数器
在末位+1时, 从低位到高位逐位进位方式工作。

原则: 每1位从“1”变“0”时, 向高位发出进位, 使高位翻转

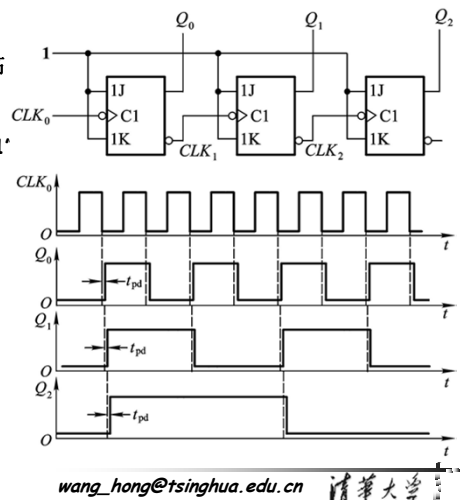


wang_hong@tsinghua.edu.cn

清华大学

②异步二进制减法计数器
在末位-1时, 从低位到高位逐位借位方式工作。

原则: 每1位从“0”变“1”时, 向高位发出进位, 使高位翻转

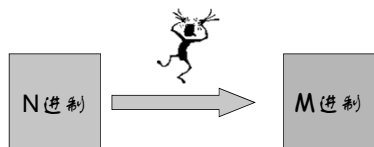


wang_hong@tsinghua.edu.cn

清华大学

三、任意进制计数器的构成方法

用已有的N进制芯片, 组成M进制计数器, 是常用的方法。



$$\begin{cases} N > M \\ N < M \end{cases}$$

wang_hong@tsinghua.edu.cn

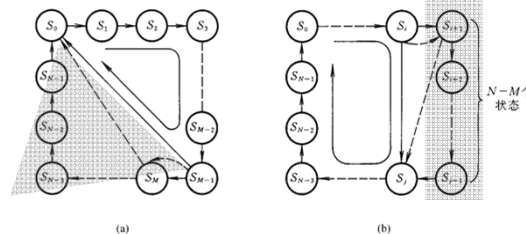
清华大学

1. $N > M$

原理: 计数循环过程中设法跳过 $N-M$ 个状态。

具体方法: 置零法

置数法



异步置零法
同步置零法

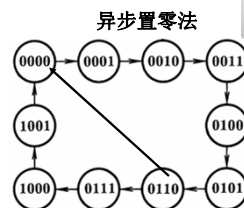
异步预置数法
同步预置数法

wang_hong@tsinghua.edu.cn

清华大学

例：将十进制的74160接成六进制计数器

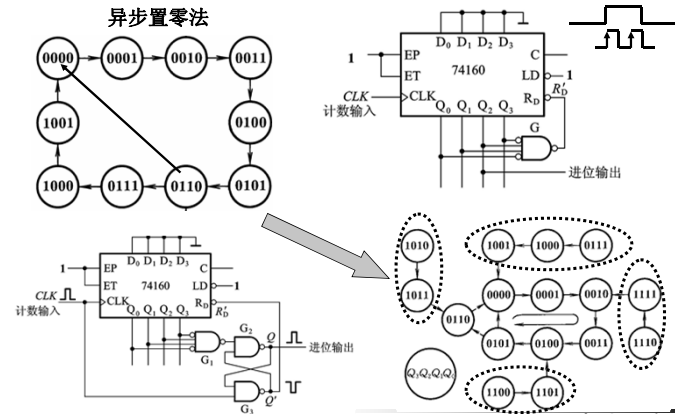
CLK	R_p	LD'	EP	ET	工作状态
X	0	X	X	X	置0 (异步)
	1	0	X	X	预置数 (同步)
X	1	1	0	1	保持 (包括C)
X	1	1	X	0	保持 (C=0)
	1	1	1	1	计数



wang_hong@tsinghua.edu.cn

清华大学

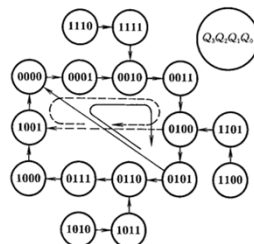
例：将十进制的74160接成六进制计数器



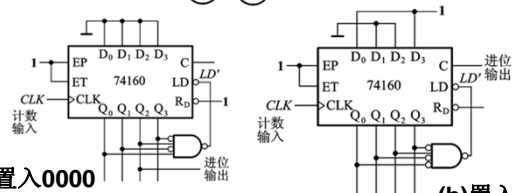
wang_hong@tsinghua.edu.cn

清华大学

置数法



(a)置入0000



(b)置入1001

wang_hong@tsinghua.edu.cn

清华大学

2. $N < M$

① $M = N1 \times N2$

先用前面的方法分别接成N1和N2两个计数器。

N1和N2间的连接有两种方式：

a. 并行进位方式：用同一个CLK，低位片的进位输出作为高位片的计数控制信号（如74160的EP和ET）

b. 串行进位方式：低位片的进位输出作为高位片的CLK，两片始终同时处于计数状态

wang_hong@tsinghua.edu.cn

清华大学

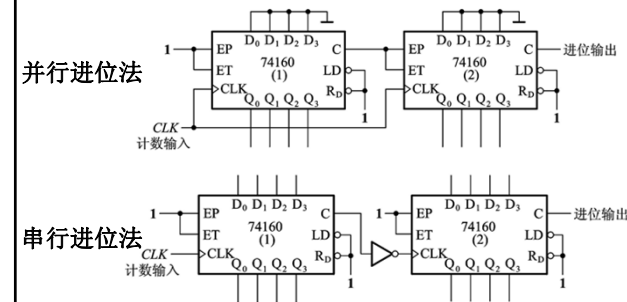
例：用74160接成100进制

CLK	R_0	LD'	EP	ET	工作状态
X	0	X	X	X	置 0 (异步)
	1	0	X	X	预置数 (同步)
X	1	1	0	1	保持 (包括C)
X	1	1	X	0	保持 (C=0)
	1	1	1	1	计数

wang_hong@tsinghua.edu.cn

清华大学

例：用两片74160接成100进制计数器



wang_hong@tsinghua.edu.cn

清华大学

②M不可分解

采用整体置零和整体置数法：

先用两片接成 $M' > M$ 的计数器

然后再采用置零或置数的方法

wang_hong@tsinghua.edu.cn

清华大学

例：用74160接成29进制

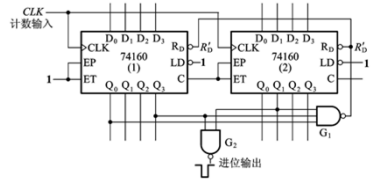
CLK	R_0	LD'	EP	ET	工作状态
X	0	X	X	X	置 0 (异步)
	1	0	X	X	预置数 (同步)
X	1	1	0	1	保持 (包括C)
X	1	1	X	0	保持 (C=0)
	1	1	1	1	计数

wang_hong@tsinghua.edu.cn

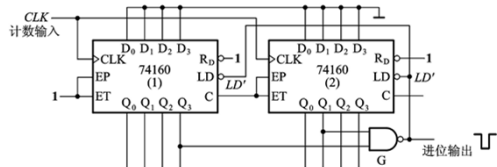
清华大学

例：用74160接成29进制

整体置零
(异步)



整体置数
(同步)

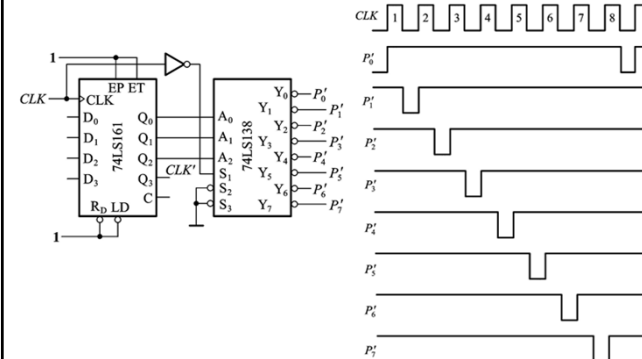


wang_hong@tsinghua.edu.cn

清华大学

计数器应用实例

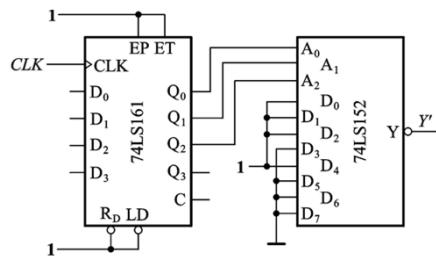
例1，计数器+译码器→顺序节拍脉冲发生器



wang_hong@tsinghua.edu.cn

清华大学

例2，计数器+数据选择器→序列脉冲发生器



发生的序列：00010111

wang_hong@tsinghua.edu.cn

清华大学

6.4 时序逻辑电路的设计方法

6.4.1 同步时序逻辑电路的设计方法

设计的一般步骤

一、逻辑抽象，求出状态转换图或状态转换表

1. 确定输入/输出变量、电路状态数。
2. 定义输入/输出逻辑状态以及每个电路状态的含意，并对电路状态进行编号。
3. 按设计要求列出状态转换表，或画出状态转换图。

二、状态化简

若两个状态在相同的输入下有相同的输出，并转换到同一个次态，则称为等价状态；等价状态可以合并。

wang_hong@tsinghua.edu.cn

清华大学

三、状态分配（编码）

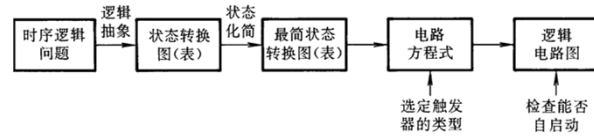
1. 确定触发器数目。
2. 给每个状态规定一个代码。
(通常编码的取法、排列顺序都依照一定的规律)

四、选定触发器类型

求出状态方程，驱动方程，输出方程。

五、画出逻辑图

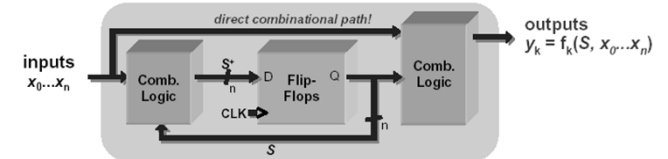
六、检查自启动



wang_hong@tsinghua.edu.cn

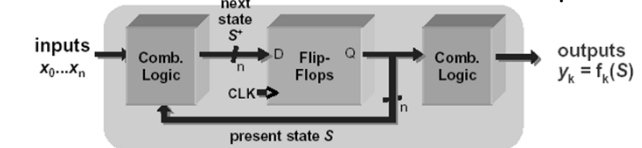
清华大学

Mealy FSM:



Outputs on Transitions

Moore FSM:



Outputs on States

wang_hong@tsinghua.edu.cn

清华大学

例：设计一个数字密码锁

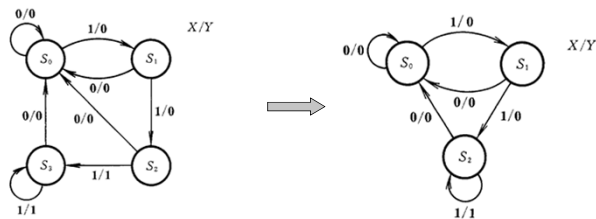
顺序输入3个或3个以上“1”时，锁打开。

一、抽象、画出状态转换图

二、状态化简

用X（1位）表示输入数据

用Y（1位）表示输出（密码是否相符）

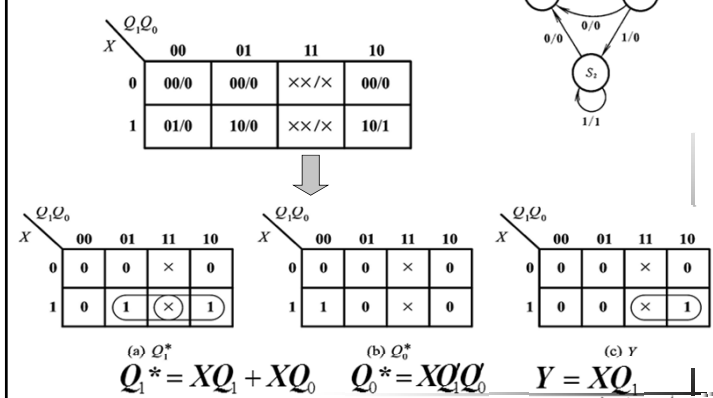


wang_hong@tsinghua.edu.cn

清华大学

三、状态分配

取 $n=2$ ，令 Q_1Q_0 的00、01、10为 S_0 、 S_1 、 S_2 ，则，



wang_hong@tsinghua.edu.cn

清华大学

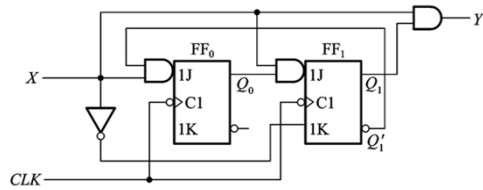
四、选用JK触发器，求方程组

$$Q_1^* = XQ_1 + XQ_0 \quad Q_0^* = XQ_1'Q_0' \quad Y = XQ_1$$

$$\Downarrow$$

$$Q_1^* = XQ_1 + XQ_0(Q_1 + Q_1') \quad Q_0^* = XQ_1'Q_0' \\ = (XQ_0)Q_1' + (X')Q_1 \quad = (XQ_1')Q_0' + 1'Q_0$$

五、画逻辑图



wang_hong@tsinghua.edu.cn

清华大学

六、检查电路能否自启动

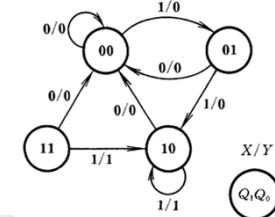
将状态“11”代入状态方程和输出方程，分别求X=0/1下的次态和现态下的输出，得到：

$$Q_1^* = XQ_1 + XQ_0(Q_1 + Q_1') \quad Q_0^* = XQ_1'Q_0' \\ = (XQ_0)Q_1' + (X')Q_1 \quad = (XQ_1')Q_0' + 1'Q_0$$

X=0时, $Q_1^*Q_0^* = 00$, Y=0

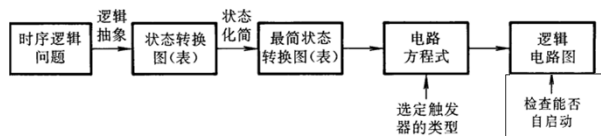
X=1时, $Q_1^*Q_0^* = 10$, Y=1

能自启动



wang_hong@tsinghua.edu.cn

清华大学

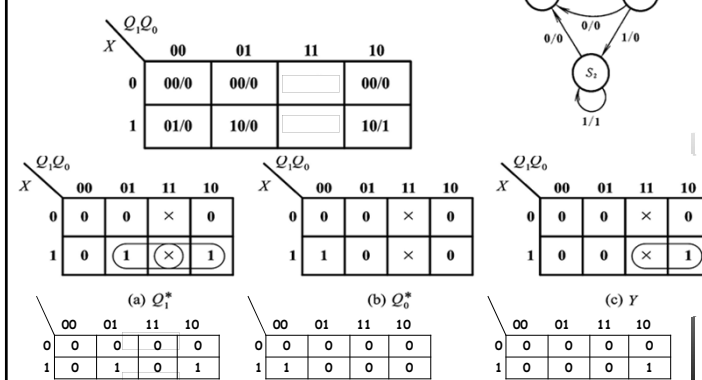


wang_hong@tsinghua.edu.cn

清华大学

三、状态分配

取n=2, 令 Q_1Q_0 的00、01、10为 S_0 、 S_1 、 S_2 则,

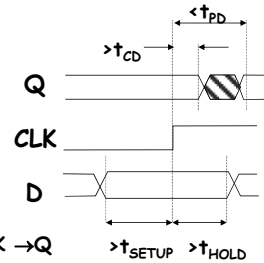


wang_hong@tsinghua.edu.cn

清华大学

Flip Flop Timing - I

$D \rightarrow D \quad Q \rightarrow Q$
 $CLK \rightarrow >$



t_{PD} : maximum propagation delay, $CLK \rightarrow Q$

t_{CD} : minimum contamination delay, $CLK \rightarrow Q$

t_{SETUP} : setup time
guarantee that D has propagated through feedback path before master closes

t_{HOLD} : hold time
guarantee master is closed and data is stable before allowing D to change

wang_hong@tsinghua.edu.cn

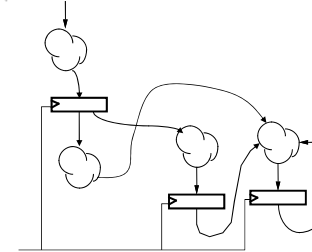
清华大学

Single-clock Synchronous Circuits

We'll use Flip Flops and *Registers* - groups of FFs sharing a clock input - in a highly constrained way to build digital systems:



Single-clock Synchronous Discipline

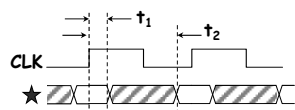
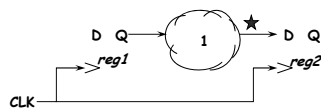


- No combinational cycles
- Single clock signal shared among all clocked devices
- Only care about value of combinational circuits just before rising edge of clock
- Period greater than every combinational delay
- Change saved state after noise-inducing logic transitions have stopped!

wang_hong@tsinghua.edu.cn

清华大学

Flip Flop Timing - II



$$t_1 = t_{CD,reg1} + t_{CD,1} > t_{HOLD,reg2}$$

$$t_2 = t_{PD,reg1} + t_{PD,1} < t_{CLK} - t_{SETUP,reg2}$$

wang_hong@tsinghua.edu.cn

清华大学

- Questions for register-based designs:
 - how much time for useful work (i.e. for combinational logic delay)?
 - does it help to guarantee a minimum t_{CD} ? How 'bout designing registers so that $t_{CD,reg} > t_{HOLD,reg}$?
 - what happens if CLK signal doesn't arrive at the two registers at exactly the same time (a phenomenon known as "clock skew")?