

数制和码制

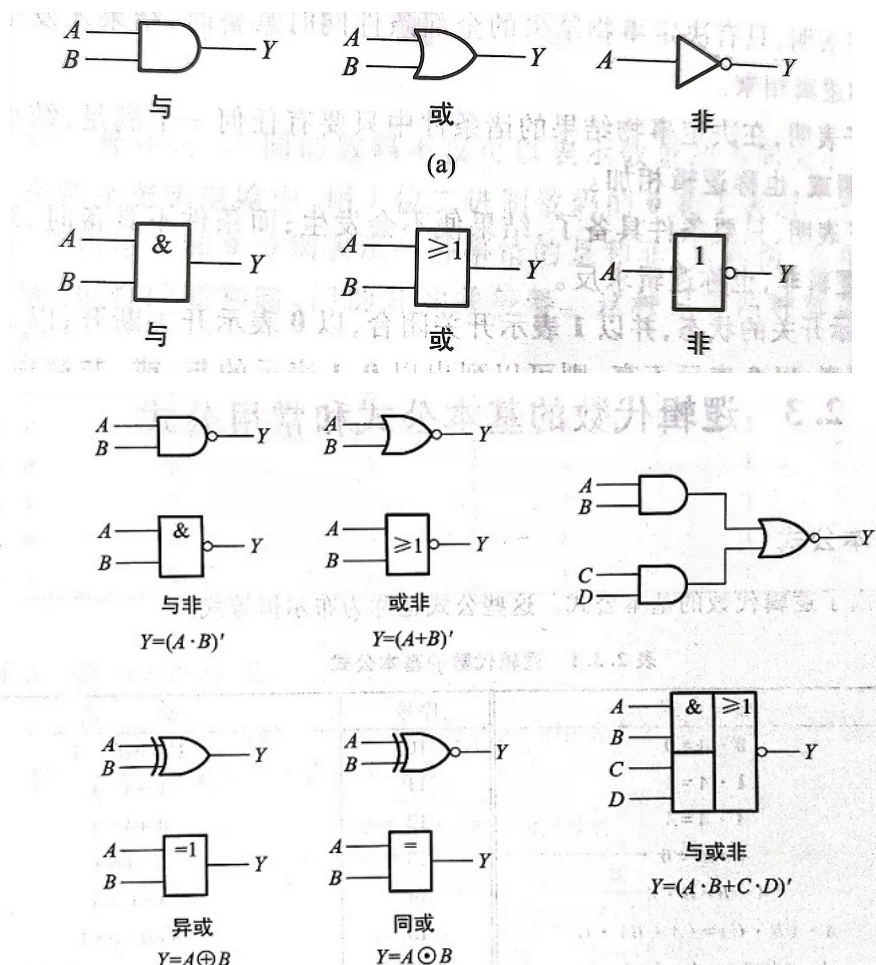
1. 负数补码=反码+1, 负数反码: 符号位不变, 其余取反
2. 1000是-8的原码/补码而不是-0的原码/补码
3. 同符号的数相加, **绝对值之和**不能超过有效数字位能表示的最大值之和
4. 格雷码: 不产生噪声
5. 余三码: 原码+3

N.B. 1. 正数原码、反码补码一致, 但需要**在前加0**

2. 补码计算出来还是补码, 如果是负数需要转化成原码

逻辑代数基础

1. 异或 $Y = A \oplus B = A'B + AB'$ (相异即为真)
同或 $Y = A \odot B = AB + A'B'$ (图形符号加圆圈, 相等即为真)
2. 图形符号



3. 常用公式

- $A + AB = A$
- $A + A'B = A + B$
- $AB + A'C + BC = AB + A'C$

- $AB+A'C+BCD=AB+A'C$

4. 最大项和最小项

表 2.5.5 三变量最小项的编号表

最小项	使最小项为 1 的变量取值			对应的十进制数	编号
	A	B	C		
$A' B' C'$	0	0	0	0	m_0
$A' B' C$	0	0	1	1	m_1
$A' B C'$	0	1	0	2	m_2
$A' B C$	0	1	1	3	m_3
$A B' C'$	1	0	0	4	m_4
$A B' C$	1	0	1	5	m_5
$A B C'$	1	1	0	6	m_6
$A B C$	1	1	1	7	m_7

表 2.5.6 三变量最大项的编号表

最大项	使最大项为 0 的变量取值			对应的十进制数	编号
	A	B	C		
$A+B+C$	0	0	0	0	M_0
$A+B+C'$	0	0	1	1	M_1
$A+B'+C$	0	1	0	2	M_2
$A+B'+C'$	0	1	1	3	M_3
$A'+B+C$	1	0	0	4	M_4
$A'+B+C'$	1	0	1	5	M_5
$A'+B'+C$	1	1	0	6	M_6
$A'+B'+C'$	1	1	1	7	M_7

最小项的编号之集合取全体编号的补得最大项的编号之集合

5. 对偶定理：对偶式相等，原式也相等
6. 多输出函数的化简：寻找共用项
7. 逻辑函数变换

$$Y = AB + BC + AC$$

$$= ((AB)'(BC)'(AC)')'$$

$$= ((A' + B')(B' + C')(A' + C'))'$$

$$= (A' + B')' + (B' + C')' + (A' + C')'$$

$$= (A'B' + B'C' + A'C')'$$

$$= (A'B')'(B'C')'(A'C')'$$

$$= (A + B)(B + C)(A + C)$$

$$= ((A + B)' + (B + C)' + (A + C)')$$

与或式（积之和）

与非—与非式

或与非式

或非—或式

与或非式

与非—与式

或与式（和之积）

或非—或非式

以上的最简式的得出均需掌握

8. 无关项分为约束项（为0）或任意项（为0位为1均可）

N.B. 1. $m_0 = A'B'C'$, $M_0 = A + B + C$

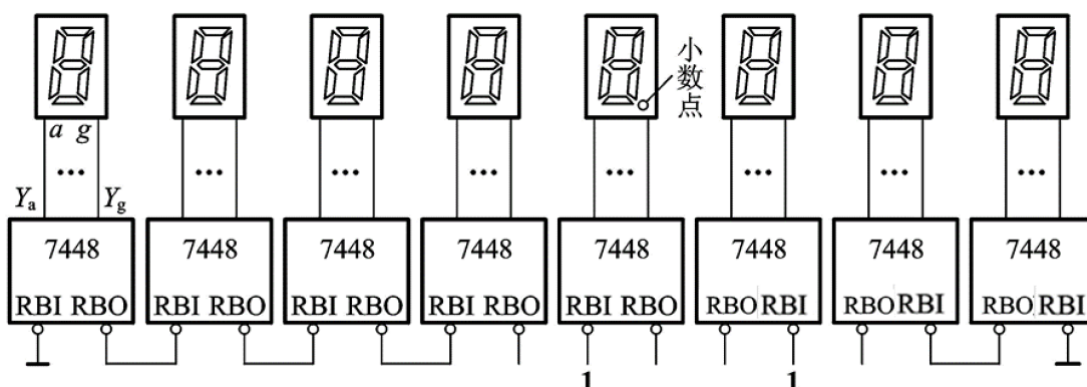
2. 最大项最小项看清一共几个变量，将它们按照顺序排好
3. 两个逻辑函数之间的与、或、异或运算可以通过将它们的卡诺图中对应的最小项做与、或、异或运算来实现
4. 有无无关项时，先画有关项，不要一开始就画无关项

组合逻辑电路

1. 功能表中A'的取反体现在了与芯片相邻的引脚处画的圆圈上，一般表示低电平有效；外接的输入线即表示对A'的输入
2. 7448多位灭零

例：利用 RBI' 和 RBO' 的配合，实现多位显示系统的灭零控制

- ❖ 整数部分：最高位是0，而且灭掉以后，输出 RBO' 作为次高位的 RBI' 输入信号
- ❖ 小数部分：最低位是0，而且灭掉以后，输出 RBO' 作为次低位的 RBI' 输入信号



$RBI'=0$ 灭零（前提是该位为0）

$RBO'=0$ 表示此位已灭零

3. 半加：不考虑低位进位
全加：考虑低位进位
4. 竞争：两个输入“同时向相反的逻辑电平变化”，称存在“竞争”

冒险：在输出产生尖峰脉冲的现象，称为“竞争-冒险”

检查竞争-冒险现象的方法：其余变量任意赋值，只要输出端的逻辑函数在一定条件下能简化成 $Y = AA'/Y = A + A'$ 则说明有竞争冒险

修改电路逻辑，如 $Y=AB+A'C$ ，当 $B=C=1$ 时 $Y=A+A'$ ；可以修改原电路为 $Y=AB+A'C+BC$

N.B.

1. 双四选一选择器：当未知量有三种时，可以一种当D，另外两种当 A_2, A_1 ；当未知量有四种时，一种当S（一边是M一边是M'），一种当D，另外两种当 A_2, A_1
2. 二进制数乘2直接在尾端加0，计算一个数的3倍数可以借助加法器将2倍数和1倍数相加
3. 用芯片设计电路时，设计好后一定要验证所有情景下的可能

半导体储存电路

1. 或非门锁存器

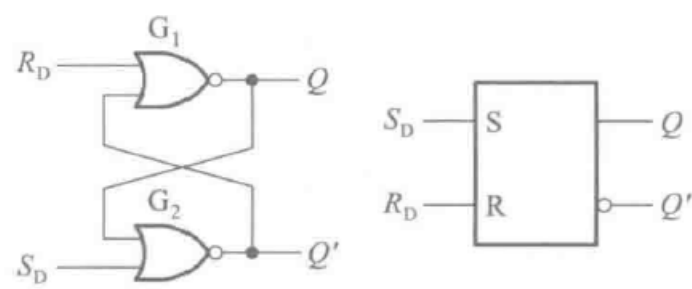


表 5.2.1 用或非门组成的
SR 锁存器的特性表

S_D	R_D	Q	Q'
0	0	0	0
0	0	1	1
1	0	0	1
1	0	1	1
0	1	0	0
0	1	1	0
1	1	0	0 ^①
1	1	1	0 ^①

① S_D 、 R_D 的 1 状态同时消失后状态不定。

注：S和R同为1时 $Q=Q'=0$ （二者相等）

与非门锁存器

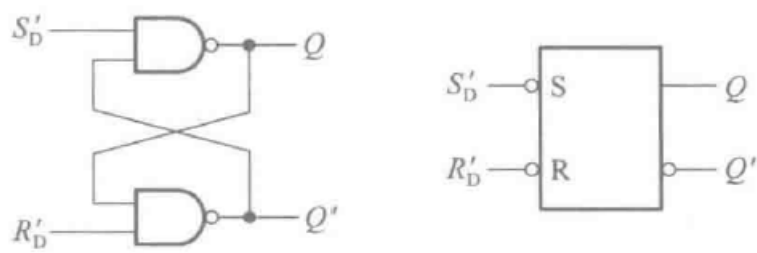


表 5.2.2 用与非门组成的
SR 锁存器的特性表

S'_D	R'_D	Q	Q^*
1	1	0	0
1	1	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
0	0	0	1 ^①
0	0	1	1 ^①

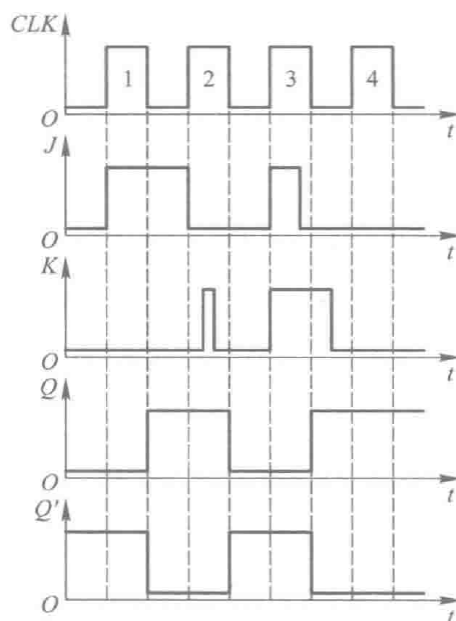
① S'_D 、 R'_D 的 0 状态同时消失以后状态不定。

注：S和R同为1时 $Q=Q'=1$ （二者相等），由此产生的脉冲触发SR触发器也按照此规则变

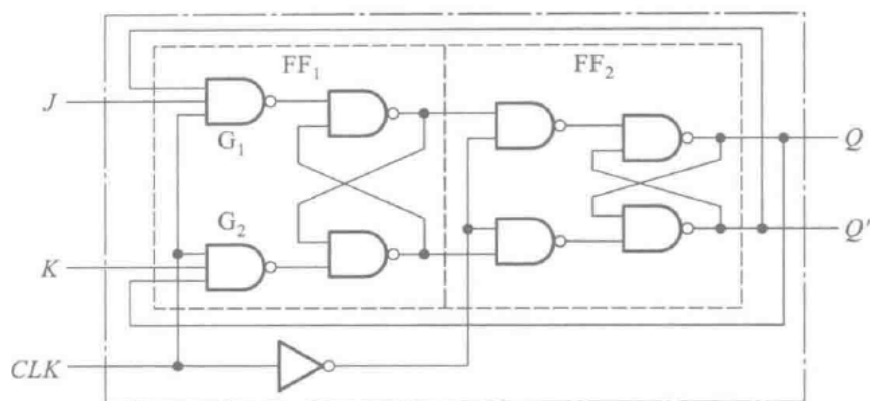
2. 脉冲触发的JK锁存器

- 对于和CLK同一时刻变的输入，边沿触发看的是前一刻，脉冲看的是已经变化完的时刻
- $J=K=1$ 的主触发器跳变，抑或是 $J=Q'$ ， $K=Q$ 引起的主触发器变化，在一个CLK上升周期中只能有一个，而且改变后就无法改变回来，也可以总结为：

一个周期内主触发器只改变一次

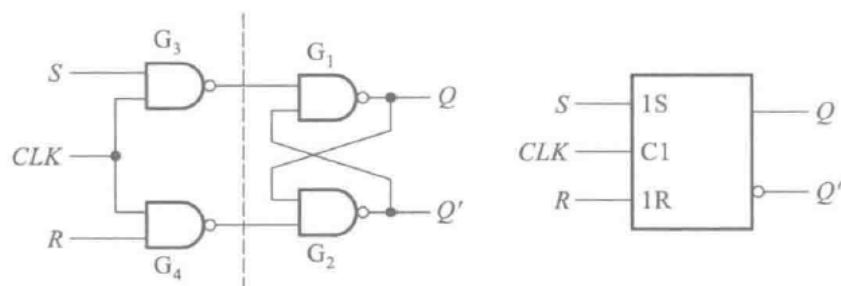


- 一定要搞清楚内部电路



3. 电平触发的SR锁存器

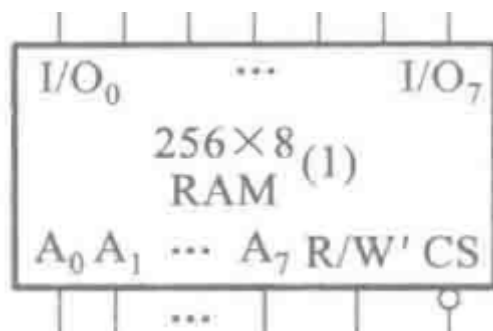
输入是S、R而非S'、R'，原于负负得正



4. T触发器: $Q^* = TQ' + T'Q$; 若 $T \equiv 1$ 则有 $Q^* \equiv Q'$, 为JK触发器两个输入口接在一起得到的, 它的频率为时钟频率的1/2

5. D触发器, J=输入, K=输入反得到 (也可以是SR触发器)

6. ROM、RAM结构



其中 $256 = 2^8$, 即有8位地址位, 而 $\times 8$ 指的是RAM为8位, 共8个输入/输出端

7. 数据表

N.B.

1. 注意 与或型锁存器 输入的是取反后的值
2. 注意SR锁存器输入均为1时 $Q=Q'$, 且与非型和或非型不同
3. 看清楚触发器类型
4. 画波形图时, 一定要看清楚输入的是S'还是S, 如果是S'请务必不嫌麻烦地画取反波形图后再做
5. 下面两个问法不同:

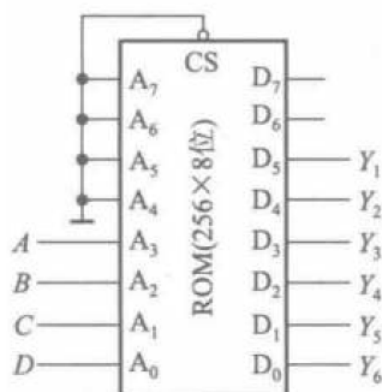
用 ROM 设计一个组合逻辑电路,用来产生下列一组逻辑函数

$$\begin{cases} Y_1 = A'B'C'D' + A'BC'D + AB'CD' + ABCD \\ Y_2 = A'B'CD' + A'BCD + AB'C'D' + ABC'D \\ Y_3 = A'BD + B'CD' \\ Y_4 = BD + B'D' \end{cases}$$

用一片 256×8 位的 ROM 产生如下一组组合逻辑函数

$$\begin{cases} Y_1 = AB + BC + CD + DA \\ Y_2 = A'B' + B'C' + C'D' + D'A' \\ Y_3 = ABC + BCD + ABD + ACD \\ Y_4 = A'B'C' + B'C'D' + A'B'D' + A'C'D' \\ Y_5 = ABCD \\ Y_6 = A'B'C'D' \end{cases}$$

前者需要画出储存矩阵的点阵图, 后者只需画出:



时序逻辑电路

1.

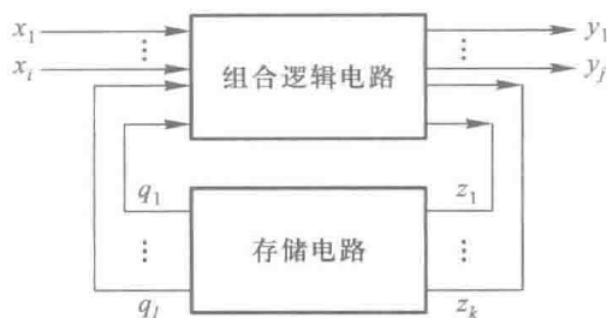
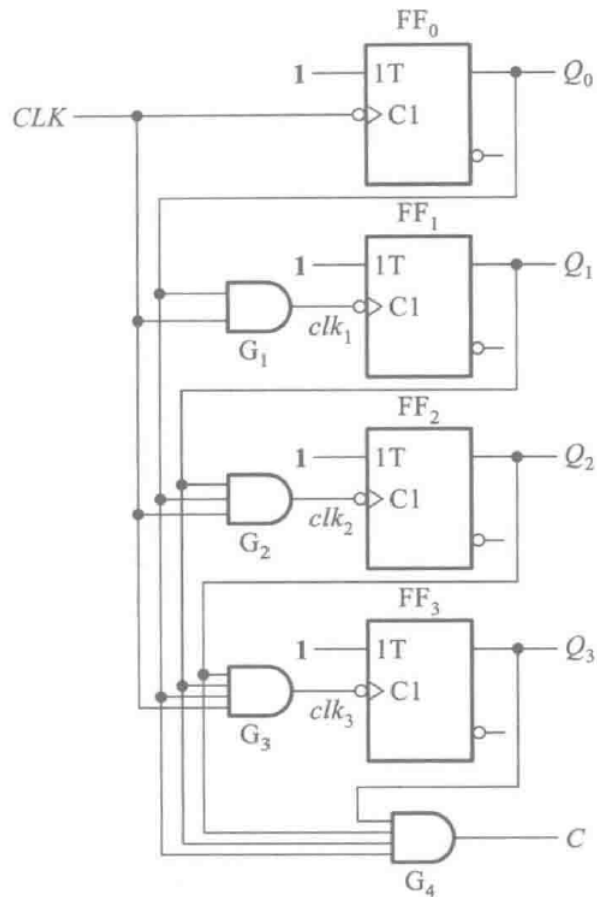
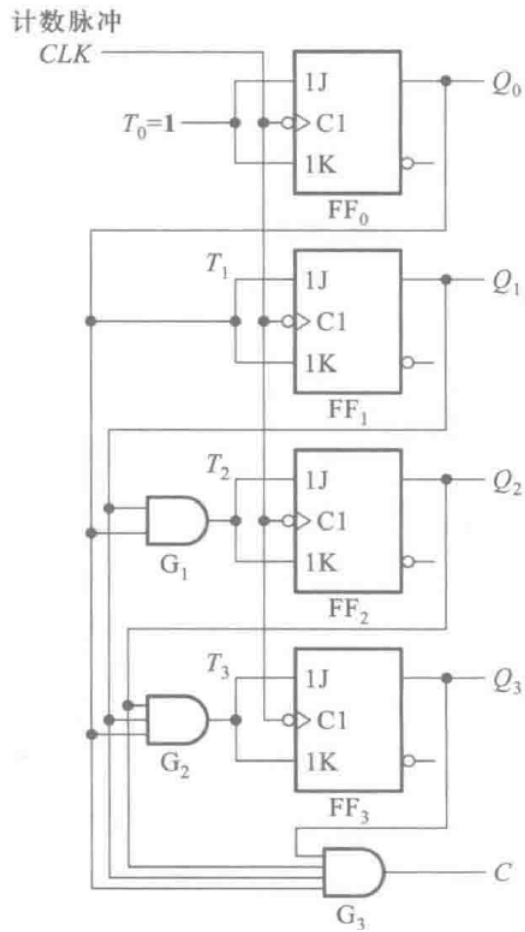


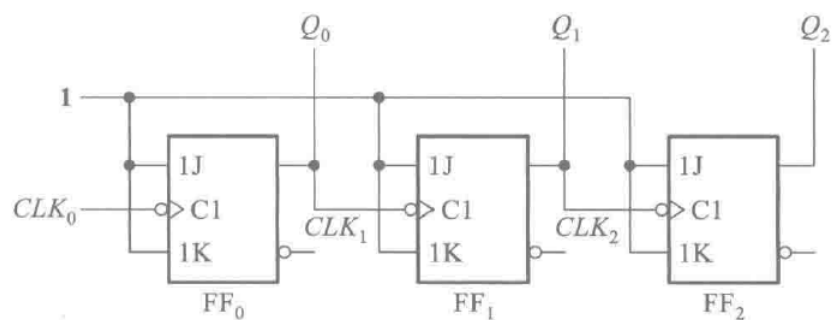
图 6.1.2 时序逻辑电路的结构框图

- 输出方程: $y=?$
- 驱动方程: $z=?$
- 状态方程: $q^*=?$
- Moore: 无x输入
- Mealy: 有x输入
- **状态转换表的两种形式**

- 状态转换图 (圆圈)
 - 状态机流程图
2. 移位寄存器：移位功能是指寄存器里的代码能在移位脉冲下依次左移或右移
 3. 计数器：计数器能计到的最大值为容量
 - 两种同步加法计数器



- 单时钟加/减法计数器74LS191、双时钟加/减法计数器74LS193
- 异步



加法

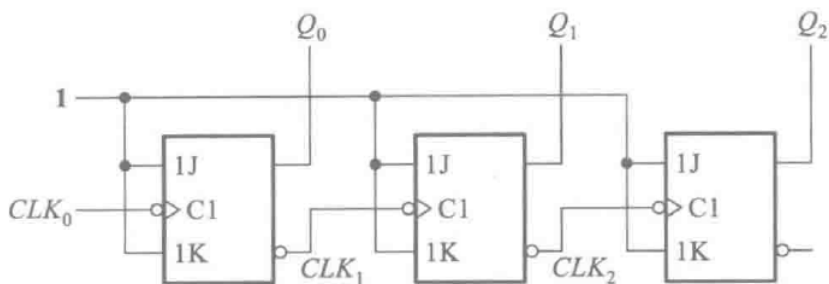
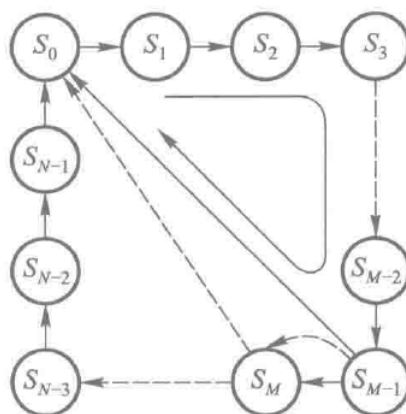


图 6.3.25 下降沿动作的异步二进制减法计数器

减法

4. 任意进制的计数器

- 置零法：注意，逻辑上m进制计数器，
 - 异步置零：需要置零的逻辑是达到m置零，**而需要置零的态是不稳定的**，稳定的循环状态中不包括 S_m



- 同步置零：需要置零的是 S_{m-1}
- 置零信号可能随置零而消失，而此时有可能未完全复位，可靠性不好，锁存器改进：

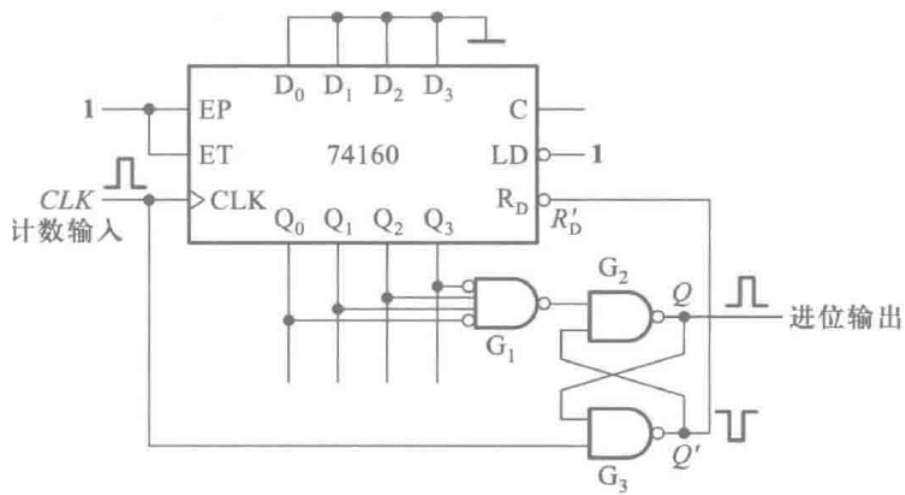


图 6.3.33 图 6.3.31 电路的改进

当CLK上升沿瞬间时 $Q=0$ ，之后置零信号 $Q=0$ 会保持一个CLK高电平时间（ $G_1G_2G_3$ 可以作为一个整体反复利用），此时**还是异步置零**

- 置数法：
 - 可以直接置零，则和置零法一样；
 - 置数法最好包含“容量”这个状态（如十进制计数器最好包含1001），这样可以在进位端C得到进位输出
- 注意改装的时候**看看是74160还是74161**，涉及到高位的计数器何时进位

5. 环形计数器，有效状态n

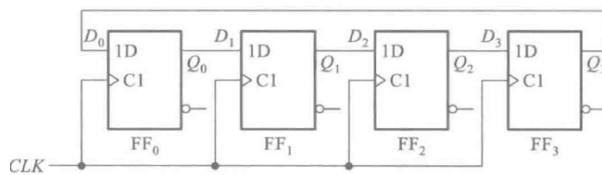


图 6.3.40 环形计数器电路

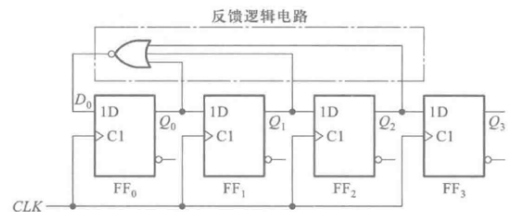


图 6.3.42 能自启动的环形计数器电路

扭环形计数器，有效状态 $2n$

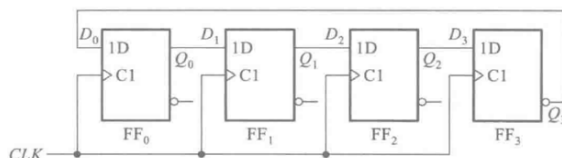


图 6.3.45 扭环形计数器电路

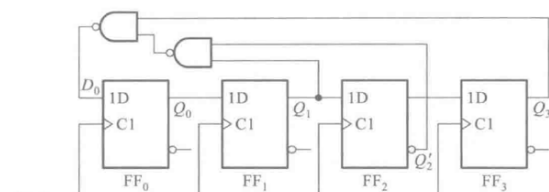


图 6.3.47 能自启动的扭环形计数器

6. 顺序脉冲发生器：计数器+译码器

7. 序列信号发生器：计数器+选择器（注意选择器末端**可能取反**）

N.B.

1. 设计时序逻辑电路时要注意 $Q' + K'Q$ 中的 K' ，**如果 $K'=0/1$ 则 $K=1/0$** ，千万不能忘记取反！！！！！！
2. 设计时序电路画卡诺图时，要注意/前的是状态而不是输入！！！！
3. 实际问题的逻辑抽象输入的时候，比如有两种情况，A和B（投币、探测物品），那么需要充分考虑如果没输入（没投币、没探测到）也需要作为一种状态
4. 注意，不能简单套状态转换图，而要**充分考虑一些置零/置数可能会导致一些其他状态被置零/置数**

