

# 数电大题总结

## 电路设计

### 组合电路

- 小规模，逻辑门，只用与非/或非门，两个输入连起来就做成一个非门，用德摩根定理
- 中规模，译码器和多路复用器，全加器和半加器，比较器，奇偶校验电路

### 时序电路

- 小规模，触发器
- 中规模，计数器，移位寄存器
- 大规模，PLA

## 常见术语

### 逻辑符号

输出方程，激励方程（也叫驱动方程）：激励信号由当前状态与输入构成的函数，状态方程，触发器的特征方程

激励表：写出状态转变对应的功能

功能表：根据输入端画真值表，写出状态转变和功能

激励/转换表、状态/输出表

X Y					X Y						
Q1	Q2	Q3	00	01	10	11	S	00	01	10	11
000	001	001	101	101	A	B	B	F	F		
001	001	011	101	111	B	B	D	F	H		
010	001	001	101	101	C	B	B	F	F		
011	001	011	101	111	D	B	D	F	H		
100	010	001	100	101	E	C	B	E	F		
101	000	011	110	111	F	A	D	G	H		
110	010	000	100	100	G	C	A	E	E		
111	001	010	110	110	H	B	C	G	G		
Q1* Q2* Q3*					S*						

一般Moore型的把输出画在最右边一栏，Mealy由于和输入有关，一般次态/输出

状态转换图，画图的时候一定要有图例，一般Moore输出画在现态的圈圈内，mealy输出画在箭头上，输入/输出

## 解题步骤

### 74x138或MUX实现函数

级联74x138，下面是高位！两片的输入连起来，两个相反的使能端连起来作为选片输入，两个相同的使能端连起来作为整体的使能端。

译码器用真值表

mux用卡诺图，做地址的放一起，一行行写D的表达式，注意11是D3

### 奇偶校验电路

检验1的个数用异或（奇数个1为1），检验0的个数用同或（偶数个0为1）

异或有因果交换律，奇数个变量的同或等于异或，偶数个异或和同或相反

两种结构：每次加一位，先两两一组比较

### 比较器

等值比较，串行/并行

数值比较

从高到低比较，高位相等时才能比较低位

### 加法器

半加器

全加器

8421码加法器，如果(X+Y)产生进位信号C 或在 1010~1111 之间需要进行修正 —— 结果加6（相当于是取了个位）

余三码加法器，进位+3，不进位-3（或+7）

## 设计模m的计数器

看清题目是二进制计数器，还是BCD码计数器

### 小规模

1.同步时序 模m， $2^{n-1} < m \leq 2^n$ ，n个计数器

D触发器设计小规模电路的基本流程：

画状态转移图，状态编码确定用几个触发器，画状态转移表，借助卡诺图（圈最大最小）写出触发器的输入表达式和输出表达式

输出要有进位位、当前计数

Moore型（输出只取决于当前状态），Mealy型（输出取决于当前状态和输入）

2.异步时序 行波计数器

每个计数器都在翻转，所以用T触发器或者D触发器 $D_n = Q_n'$ 或者JK触发器 $J=K=1$ ，而且是在前一个计数器的边沿触发翻转

上升计数就是第n+1个触发器,在 $Q_n$ 的下降沿触发状态翻转

所以n+1的时钟信号接入 $Q_n'$ ，

下降计数是在 $Q_n$ 的上升沿触发n+1个触发器的状态翻转

画图的时候注意不要漏掉第一个触发器接CLK时钟信号

### 中规模

1.74x163同步二进制计数器

设计原理是仅当第i位以下的各位都为1，+1时，第i位的状态才会改变，最低位的状态每次加1都要改变。

$EN_i = Q_{i-1} \cdot Q_{i-2} \cdot \dots \cdot Q_1 \cdot Q_0$ , 可以用T触发器接入同一个时钟信号, 并行使能 (每次都Q和CNTEN重新算) 或者串行使能 (利用上一次的使能端)

与大规模结合, 多一个表示成最小项 (与阵列) 之和 (或阵列) 的过程

考点: 设置置数或者清零, 同步在m-1触发, 异步--m

清零法 (使用CLR或者LD置数为0): 只用接1, BCD还是二进制, 级联,  $m_1 * m_2$ , RCO接入下一片的使能端, 低位什么时候清零? 高位清零要低位和高位同时满足条件才能清零, 比如60进制, 低位是9, 高位是5, 先与非, 两个与非同时是低才能使LD低有效, 所以要用或门

采用整体预置数法,  $63 \sim 255$ ,  $256 - 193 = 63$ , 两个都进位 (与非两个RCO) 的时候置数成63

不用的注意使无效, 写0或1, 使能端置1

TC/RCO进位输出, PE/LD

同步计数器: <b>BCD</b> 码是对十进制的每一位分别编码		
复位/置位	<b>BCD (模10)</b>	<b>二进制 (模16)</b>
同步/同步	<b>74LS162</b>	<b>74LS163</b>
异步/同步	<b>74LS160</b>	<b>74LS161</b>
异步/异步	<b>74LS190 (加/减)</b>	<b>74LS191 (加/减)</b>

!! 同时为低才有效, 用与非门连接后再接或门

题型变式: 给电路图, 识别模 写出清零或者置数的表达式, 然后画出状态转移表和图

2. 既不是升序, 也不是降序->移位寄存器+反馈逻辑

n位移位寄存器能够构成模n的环形计数器

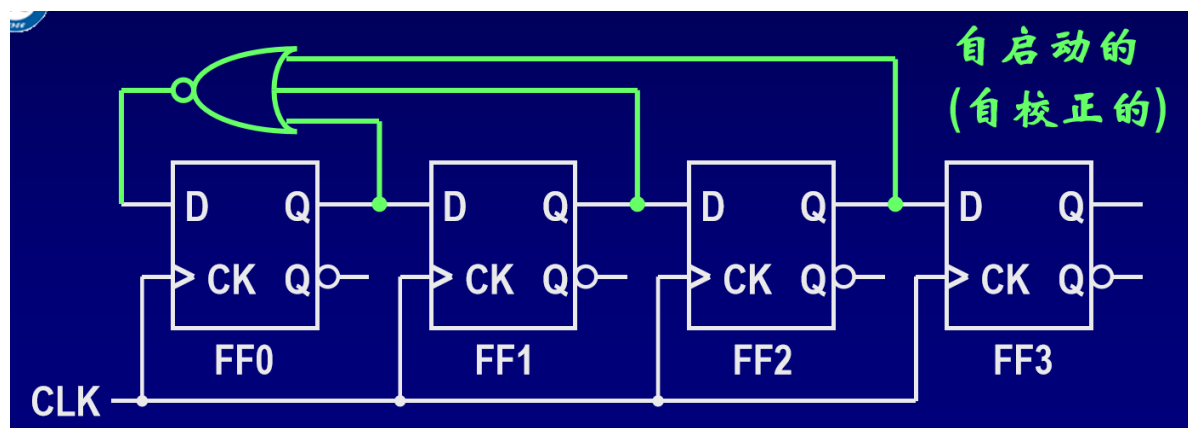
n位移位寄存器能够构成模2n的扭环形计数器

3. 环形计数器  $D_{n+1} = Q_n$ ,  $D_0$  = 最后一个Q

就是每次右移一位, 最后一位往左边补 非自启动

自校正的:  $D_0 = (Q_{n-2} (\text{倒数第二个}) + \dots + Q_1 + Q_0)'$

扭环计数器  $D_{n+1} = Q_n$ ,  $D_0$  = 最后一个Q非



自校正的方法

1. 打破无效环, 使进入有效环

借助卡诺图，先把无效的也填进去，然后再修正

2.利用置数功能

3.利用清零

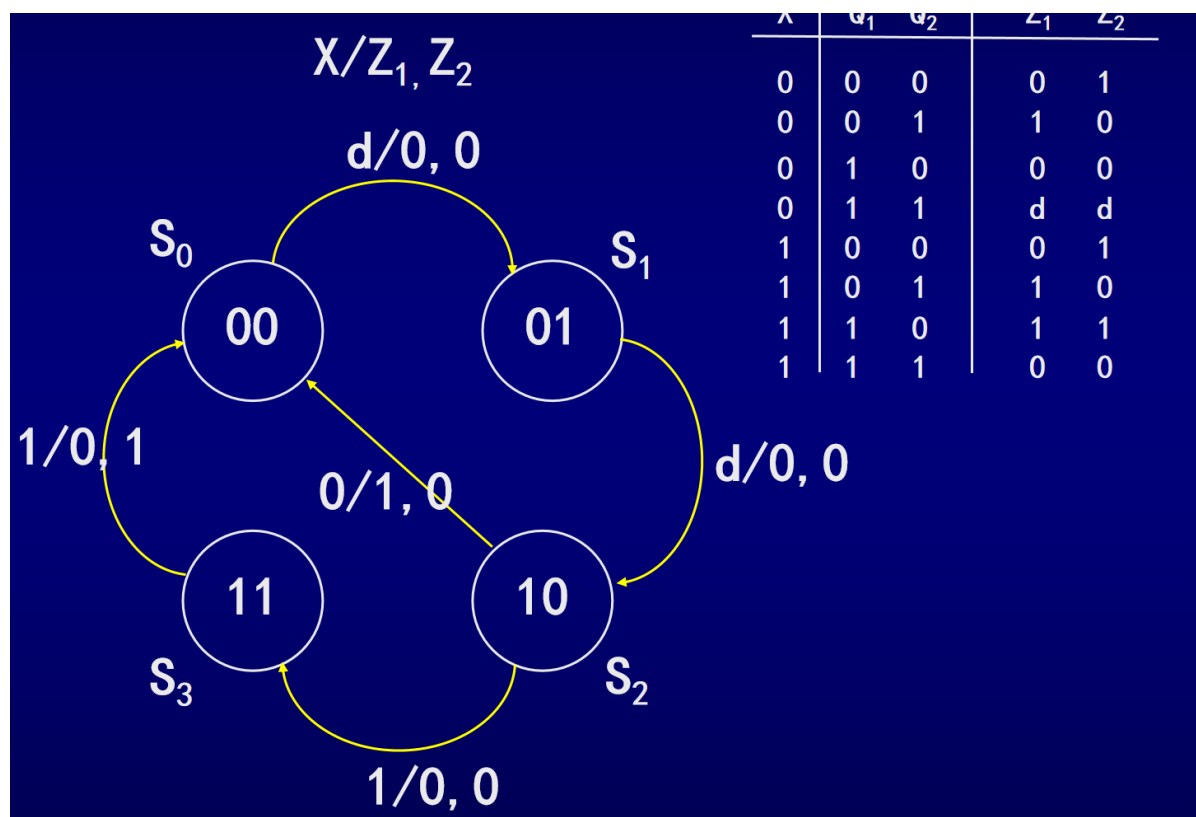
D触发器的结构要和74x194对应

**例：**设计一个模可变的同步递增计数器。当控制信号 $X=0$ 时为三进制计数，当 $X=1$ 时为四进制计数。要求：**D**触发器实现。

$X$ ——控制输入端

$Z_1$ ——三进制进位输出端

$Z_2$ ——四进制进位输出端



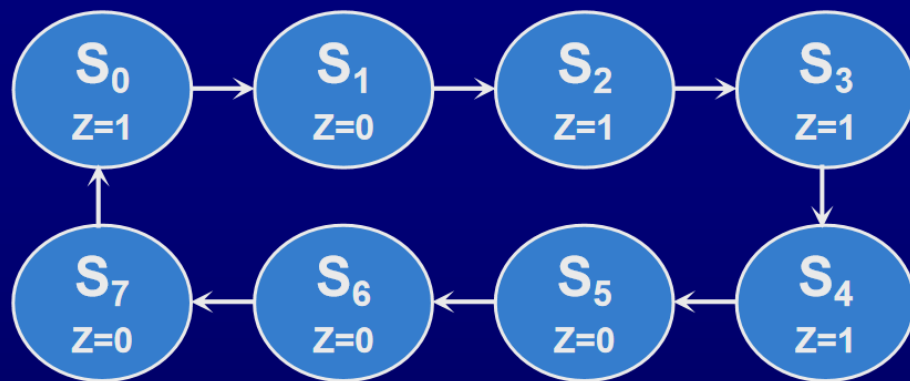
## 序列发生器

1.同步的触发器，没有输入，相当于是moore

输出的是序列，长度为 $n$ 则有 $n$ 个状态，然后给状态编码，确定用几个触发器

## (1) 构造状态表或状态图

10111000



2.移位寄存器构成——注意自校正（环形计数器）

左移，输出现态的最高位，D0是右边补进来的数，所以应该是次态的最低位

注意确定状态变量的个数，不能出现状态重叠，就是你划下划线的不能出现两组相同的，现态相同，但移入信号又不同就无法运行

至少需要5个状态变量： $Q_4Q_3Q_2Q_1Q_0$

1 1 1 1 1 0 0 0

$S_0$  : 1111

$S_1$  : 1110

$S_2$  : 1100

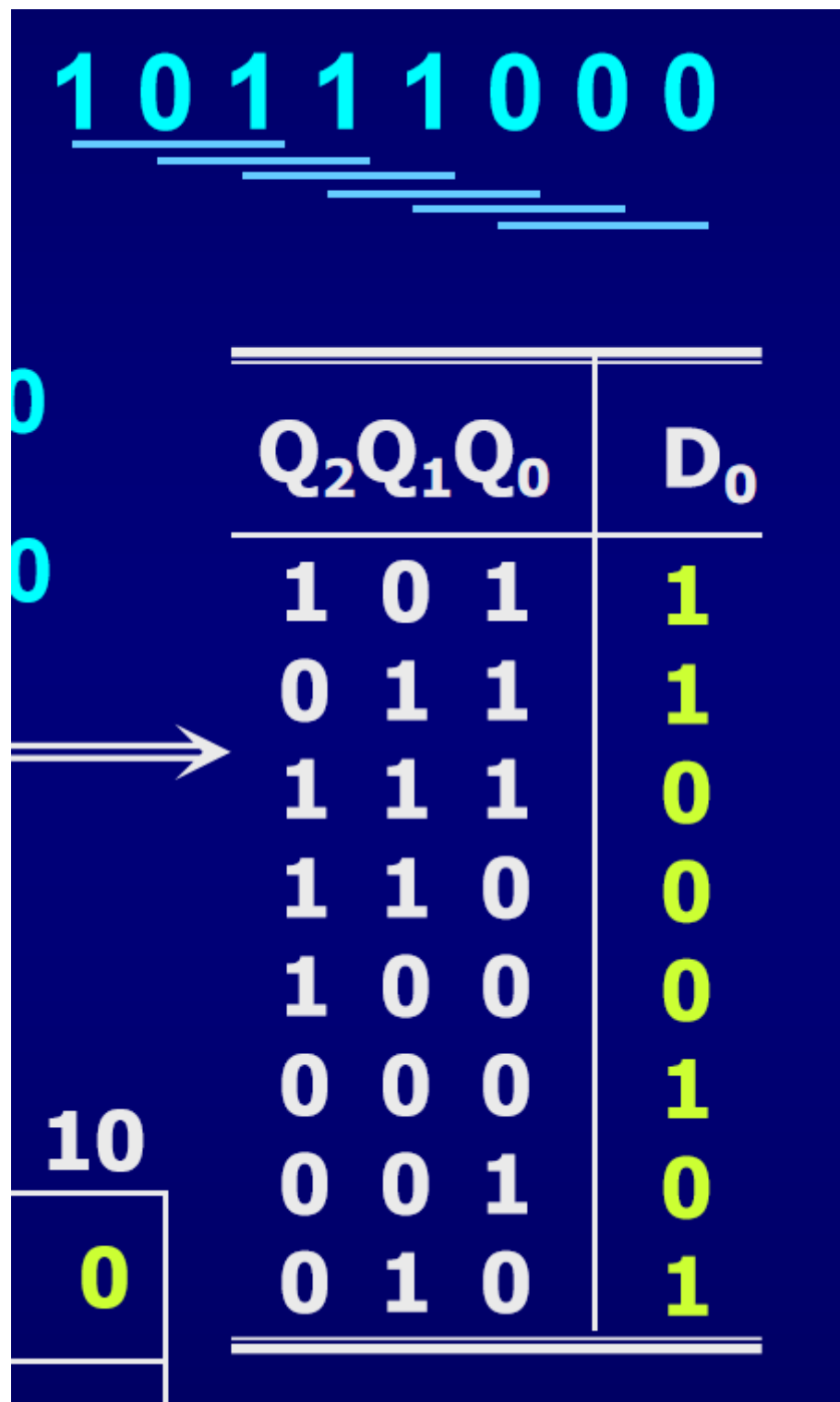
$S_3$  : 11000

$S_4$  : 10001

$S_5$  : 00011

$S_6$  : 00111

$S_7$  : 01111



注意74x194左移的时候接LIN，右移的时候接RIN，不是看进来的数是在左边还是右边

3.计数器+多路复用器

8选1的MUX，输入序列，然后计数器产生选址信号，依次选出来就行

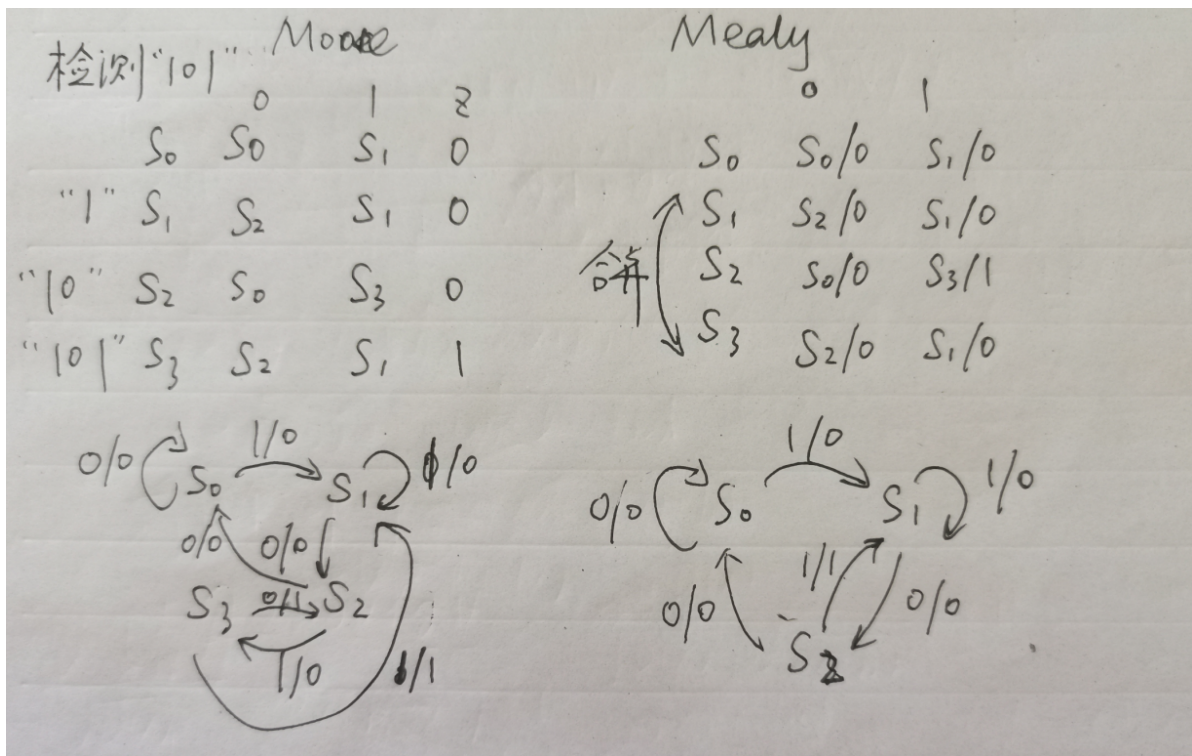
## 序列检测器

注意是否可重叠（上次检测到完整序列的最后几位能否作为下次检测的开头）

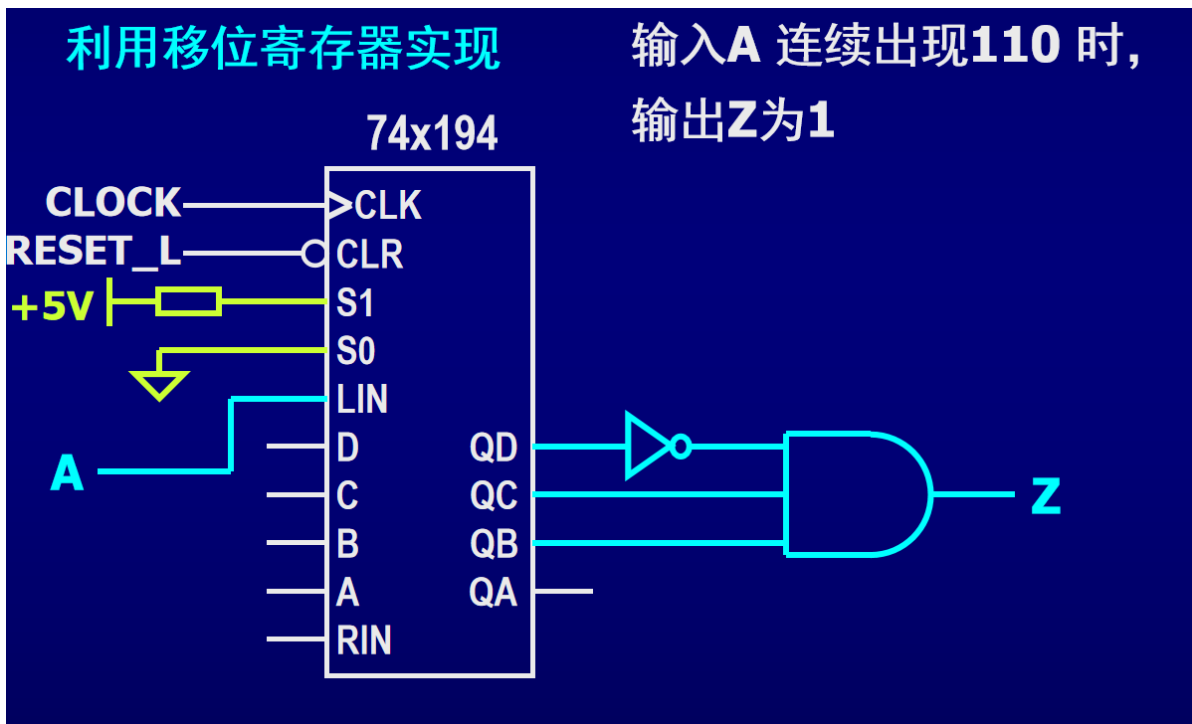
1.用触发器，一般mealy状态会比Moore少

等待S0，捕获第一位S1，.....，捕获最后一位->状态合并

如果能自启动但有错误输出，一般是由于圈卡诺图的时候圈上了d，修改圈卡诺图的方式就能更正



2.用74x194移位寄存器，直接LIN输入A然后左移，检测输出是不是该序列



变式：检测BCD码是否合法，是检测连续的四个输入而且每一位都有两种不同情况，画成一棵二叉树，左0，右1，在第三层之后的输入判断是否合法，然后回到根

## 触发器的转换

用A触发器实现B触发器，其实输入信号是以B的形式表示的，实际接入的是A，所以是用B来表示A，不仅可以用到输入信号和Q，Q'，还可以用到时钟信号、使能端做改装，比如D改成T'（T的上升沿翻转）， $D=Q'$ （这样就能不断翻转）， $CLK=T$ ，有时候也要靠分析

代数法，联立方程

联立激励表，画出A与Q，B的真值表，求出 $A=f(Q,B)$



可加可减计数器，进位111->000，退位000->111

三/四进制计数器

看波形图求功能：计数器的特点--顺次下降/上升

根据波形图逆推电路图：上升/下降沿触发，把波形图标上0,1，转换成真值表，写出状态转移方程

## 电路分析

---

功能一般就那么几个，先看有没有循环和进/借位输出--计数器，再看是不是序列检测器，没有输入有可能是计数器或者序列发生器

## Notice

---

1.约束条件 $SR=0$

2.分析题先看时钟信号，同一个时钟信号就是同步，上一个的输出作为下一个的时钟信号就是异步，主从-时钟信号相反