

Lect02

码制

8421 BCD 有权 权为8421 后六个数不允许出现表示十进制一个一个转

5421 有权 $5 = 1000$ 一半以后首位是1

2421 有权 只能表示0-9 一半以后首位是1 $4=0100$ $5=1011$

余3 无权 $= 8421+0011$ 相加时注意和减3 进位则加3

为什么？4位才是1个10进制位 直接相加并没有道理 其他有权码甚至不能正确判断相加的进位。余3因为两个余3就是一个6, $10+6=16$ 可以进位并且抵消低位的余3

四种十进制数的编码表

十进制数	8421码	5421码	2421码	余三码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1000	1011	1000
6	0110	1001	1100	1001
7	0111	1010	1101	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100

格雷码 无权码 多形式：任何相邻的十进制数格雷码仅有1位不同

减少代码变化中电路书瞬间产生的错误，可靠性高

十进制数	8421码	格雷码1	格雷码2	典型格雷码	修改格雷码
0	0000	0000	0000	0000	0010
1	0001	0001	0001	0001	0110
2	0010	0011	0011	0011	0111
3	0011	0010	0010	0010	0101
4	0100	0110	0110	0110	0100
5	0101	1110	0111	0111	1100
6	0110	1010	0101	0101	1101
7	0111	1011	0100	0100	1111
8	1000	1001	1100	1100	1110
9	1001	1000	1000	1101	1010

典型格雷不能9回1，修改格雷不从0000开始

- 格雷码1：除最高位之外中线对称 45 36 27 18 90
- 典型格雷码：对任意长的二进制数编码（除10进制是对BCD）
 $G=B_{i+1} \text{ xor } B_i$ 每个位都是自己异或高位 最高位直接拿来用（或者假设虚高位是0）
- 解码？
- 修改格雷码：从余3开始编典型格雷码，为了前后循环 3和12衔接，也可满足中线对称。

字符编码

7位ASCII 教材P13 高3位区分控制字符、数字符号、大写英文、小写

Chap 2 逻辑代数

2.1 基本运算公式

- n 输入的不同逻辑共有多少种？

相当于对于定义域在所有输入组合上，有多少种不同的函数关系？ $2^n \times 2^n$

同或：not (xor) 相同取1，不同为0 $AB + A!B$

符号名称	国标符号	常用符号	国际流行符号	IEEE逻辑符号
与门				
或门				
非门				
与非门				
或非门				
异或门				
同或门				

互补——A与!A 1律 0律 重叠律A A 双非 (对合)

交换 结合 分配 不加括号也是&先算，即使在后面也可以&对|分配

分配律 (Distributivity)
$$\begin{cases} A \bullet (B + C) = A \bullet B + A \bullet C \\ A + B \bullet C = (A + B) \bullet (A + C) \end{cases}$$

自己+自己求反&条件B = 自己+条件B

自己+自己&条件B = 自己

吸收律 (Absorption)

$$\begin{cases} A + \overline{A} B = A + B \\ A \bullet (\overline{A} + B) = A \bullet B \\ A + A \bullet B = A \\ A(A + B) = A \end{cases}$$

反演律(德·摩根定律)
$$\begin{cases} \overline{A \bullet B} = \overline{A} + \overline{B} \\ \overline{A + B} = \overline{A} \bullet \overline{B} \end{cases}$$

(De Morgan's)

包含律

(Consensus)

$$\left\{ \begin{array}{l} AB + \overline{A} C + BC = AB + \overline{A} C \\ (A + B)(\overline{A} + C)(B + C) = (A + B)(\overline{A} + C) \end{array} \right.$$

包含律推论 $AB + \overline{A} C + BCD = AB + \overline{A} C$

反演规则反函数 与或互换；变量求反；01互换即可 得到反函数

对偶规则 与或互换；01互换；得到对偶式 ($F' = F; F = G \Rightarrow F' = G'$)

2.2 公式法化简——吸收！包含！

► 最简与或表达式：

1. 乘积的个数最少(用门电路实现，用的与门数最少)
2. 在满足1的条件下，乘和项中的变量最少(与门的输入端最少)

—

或与表达式：

$$F = (A + B)(\overline{A} + C)(C + DE)$$

结果要求：化简结果仍为或与表达式

最简条件：

- (1)、或项个数最少(或门用的最少)
- (2)、在满足1的条件下，或项中变量数最少

貌似还是用与或去化简然后狄摩根成或与。。

化简不动的时候试试拆分一个假1变成 $A + !A$ 的形式：

尤其是在很对称工整的形式里面化简，化成不对称不公整但是可以凑一起的形式

$$\begin{aligned} F &= \overline{\overline{A} \oplus B} + \overline{B \oplus C} \\ &= AB + \overline{AB} + BC + \overline{BC} \\ &= AB(C + \overline{C}) + \overline{AB} + BC + \overline{BC}(A + \overline{A}) \\ &= ABC + A\overline{BC} + \overline{AB} + BC + \overline{ABC} + \overline{ABC} \\ &= (ABC + BC) + (\overline{AB} + \overline{ABC}) + (A\overline{BC} + \overline{ABC}) \\ &= BC + \overline{AB} + \overline{AC} \end{aligned}$$

作业

■ 2.6 (1) ~ (6)

2.6 用公式法将下列函数化简为最简“与或”式。

$$(1) F = \overline{A}\overline{B} + (AB + A\overline{B} + \overline{A}B)C$$

$$(2) F = (X+Y)Z + \overline{X}\overline{Y}W + ZW$$

$$(3) F = AB + \overline{A}C + \overline{B}\overline{C}$$

$$(4) F = AB + \overline{A}\overline{B}C + BC$$

$$(5) F = \overline{A}B + \overline{A}C + \overline{B}\overline{C} + AD$$

$$(6) F = \overline{A}\overline{B} + \overline{A}\overline{C}D + AC + B\overline{C}$$

$$(7) F = AC + \overline{A}\overline{B} + \overline{B}\overline{C}\overline{D} + BE\overline{C} + DE\overline{C}$$

$$(8) F = A(B + \overline{C}) + \overline{A}(\overline{B} + C) + BCD + \overline{B}\overline{C}D$$

$$(9) F = \overline{X}\overline{Y} + (X+Y)Z$$

$$(10) F = (X+Y+Z+\overline{W})(V+X)(\overline{V}+Y+Z+\overline{W})$$

Lect 3

最小项 n个变量与 从000开始到111排序 三变量0-7 C是最高位

只有一组取值为1剩下都是0 任何逻辑函数表示为唯一一组最小项的或

某个最小项不是在F中就是在F的反函数中，用 m_i 来表示 下图0-7

$$\begin{array}{cccc} \overline{\overline{A}}\overline{\overline{B}}\overline{C} & \overline{\overline{A}}\overline{B}\overline{C} & \overline{A}\overline{\overline{B}}\overline{C} & A\overline{B}\overline{C} \\ \overline{\overline{A}}\overline{B}\overline{C} & \overline{A}\overline{\overline{B}}\overline{C} & \overline{A}\overline{B}\overline{\overline{C}} & A\overline{B}\overline{C} \end{array}$$

最大项 n个变量或一起，唯一一组为0 从0到7 C最高位

任何F都是最大项之积（与），标准或与大家在一起必为0

$$\begin{array}{ll} M_0 = A + B + C & M_1 = \overline{A} + B + C \\ M_2 = A + \overline{B} + C & M_3 = \overline{A} + \overline{B} + C \\ M_4 = A + B + \overline{C} & M_5 = \overline{A} + B + \overline{C} \\ M_6 = A + \overline{B} + \overline{C} & M_7 = \overline{A} + \overline{B} + \overline{C} \end{array}$$

m 和 M 之间是求反的关系 $M_i = !m_i$

而 $\sum m_i$ 和 $\prod M_i$ 为对偶式

卡诺图：逻辑函数的图示表示，把最小项填入卡诺图，利用相邻最小项的互补性，消去一个变量，实现化简。

卡诺图的构成

- (1) 由矩形或正方形组成的图形
- (2) 将矩形分成若干小方块，每个小方块对应一个最小项。

按循环码顺序，首尾满足格雷性质，00 01 11 10 110 111 101 100

1. 画图
2. 填数
3. 合并
4. 先大后小，能大就大——多检查上下左右边界

5. 不重不漏

多输出化简：在与或中尽量找公共项让多个函数共享，容易最简，并不是单个最简

无关项：打X，可以1

Lect4 与非门与外特征

3.1 引言

输出取决于输入逻辑组合，和过去状态无关，只有关于当前

实际情况：前后沿存在延迟且不相等

典型的组合逻辑电路

- 门电路 (Gates)
- 译码器 (Decoders) 和 编码器 (Encoders)
- 数据选择器 (Multiplexer)
- 加法器 (Adders)
- 奇偶校验器
- 算术逻辑单元 (ALU: Arithmetic Logic Units)

集成电路？

■ 集成电路的分类

- 按功能分：
 - 模拟电路：也叫线性电路，如通信电路
 - 数字电路：从门电路到微处理器、存储器等多种
- 按半导体制造工艺：
 - TTL：（也叫双极型，TTL,LTTL,STTL,LSTTL,ECL…）
 - MOS：（PMOS,NMOS,CMOS,BiCMOS…）
- 按封装（外形）分：
 - 双列直插
 - 表面封装
 - BGA(Ball Grid Array)
 - PLCC，等等
- 按集成规模分：
 - 小、中、大、超大、甚大规模集成电路

	速度	功耗	集成度
TTL(晶体管晶体管逻辑)	快	大	低
MOS (金属氧化物半导体)	慢	小	高

| 目前常用器件的工艺：

□ CMOS (互补金属氧化物半导体)

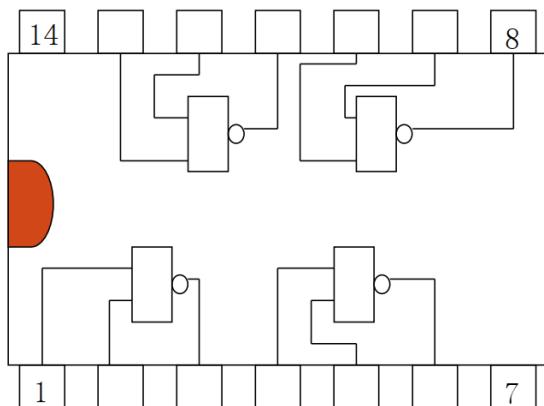
集成度18月翻一番 (Moore Law) XXSI 05\12 M\100 L\10,000 VL \1,000,000 UL

三维集成延续摩尔定律——功耗太高 (电力、热问题、延迟、可靠性)

系统功耗是关键因素，低功耗设计环节，优化、封装

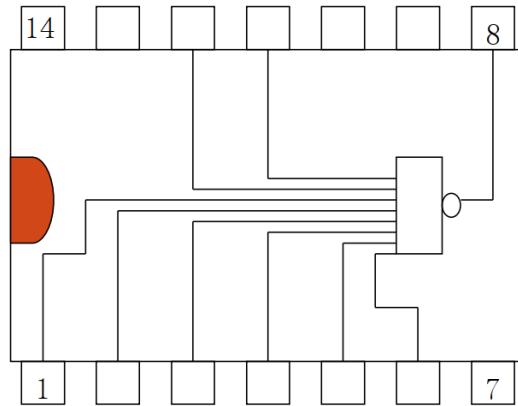
3.2 门电路

74LSXX元件速查 00 30



74LS00

2输入4与非门

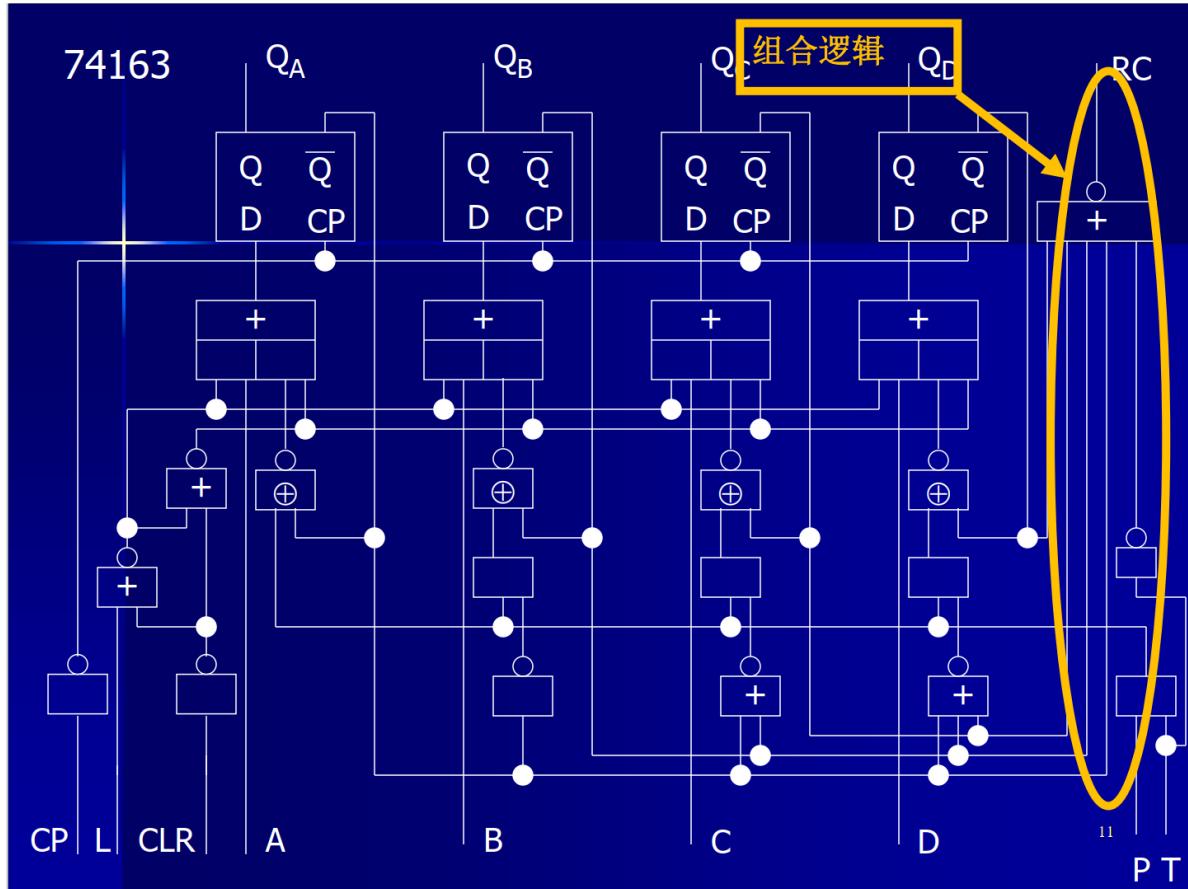


74LS30

8输入与非门

74LS139 138 154 LECT5 P37

74LS85 四位比较器 2*4bitAB输入，前面大小等于输入以及后面大小等于输出



与非门的封锁 0即封锁为1

与或非门的封锁 1即封锁为0

与或非的数据选择，在两路数据分别与上C和!C即可控制，1为选择端

正逻辑和负逻辑，用1表示高电平为正逻辑

简单二值逻辑

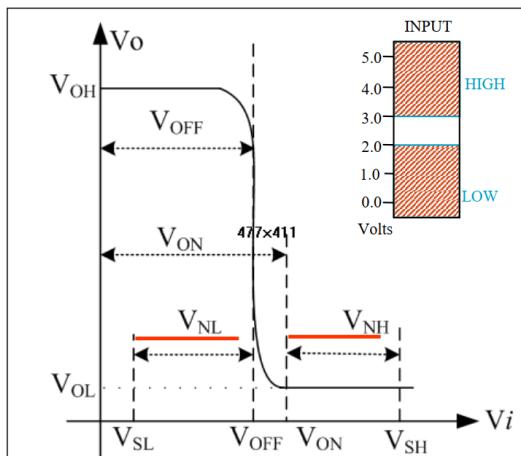
- 开关：闭合为L 打开H 输出在Vcc一侧
- 晶体管：截止、放大、饱和、倒置（不用）**Lect4 P50**

与非门的外部特性

- 开关特性：扇入（可用输入）、扇出（驱动数）、传输延迟（输入传输到输出的时间，速度和最大传输延迟反比）、功耗
 - 传输延迟：一般参考HL的中间电平来参考时间点， t_{PHL} t_{PLH} 不同， $t_{pd} = \max(t_{PHL}, t_{PLH})$
 - 测量延迟时间？一个输入即可， $tpd = (tpd1 + tpd2) / 2n$
 - 自激励震荡电路：奇数个非门串联，周期为 $n * \text{单个总延迟}$
- 转移特性：门电路中输出电压随输入电压的变化特性 $V_{in}-V_{out}$
 - V_{out} 急剧改变的门槛电压、阈值电压 V_T
- 直流参数：0输入电流 $\leq 1.6\text{mA}$ 1输出 $|IOH| \leq 400\mu\text{A}$ $V_{oh} \geq 3\text{V}$...
 - 一般高电压H的时候要求的电流比较小，在几十几百 μA 左右 输入电流比输入小上十倍差不多（因为正常负载能力在10）
- *技术参数：噪音容限
 - 叠加到正常输入的外部噪声电压，不会造成不可预料变化

□ 叠加到正常输入值的最大的外部噪音电压，它不会在电路的输出产生不可预料的变化

与非门的电压传输特性曲线



低电平电平噪声容限：
 $V_{NL} = V_{OFF} - V_{SL} = V_{OFF} - 0.4$

高电平电平噪声容限：
 $V_{NH} = V_{SH} - V_{ON} = 2.4 - V_{ON}$

93

与非门电路级联

- 负载计算
 - 计算负载能力？ $N = I_{OH}/I_{IH} = 400\mu A/40\mu A = 10$ 输出1的情况
输入0 则 $I_{OL}/I_{IL} = 16/1.6mA = 10$ 输出0的情况，电流方向反
 - 负载太大？ 非正常 低电平变高 高电平变低整个逻辑工作异常

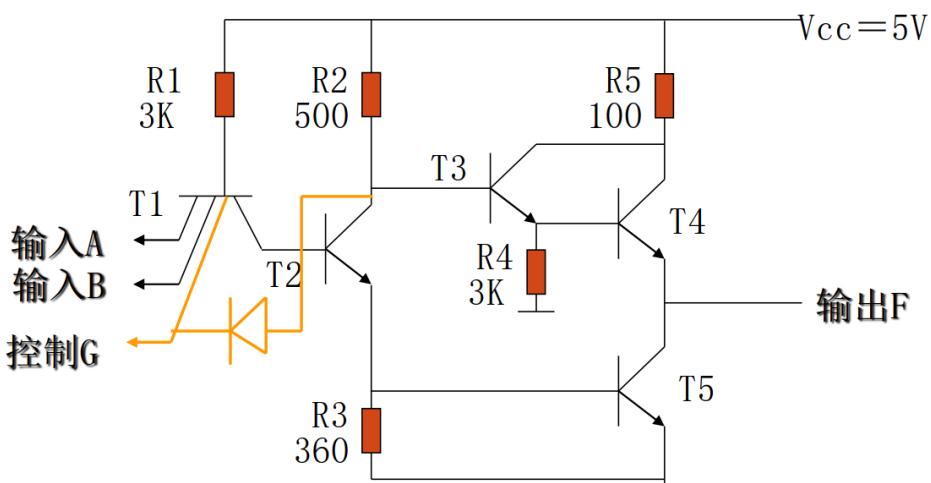
电路设计的“线与”问题 需要多个外设输出连接一个输入总线，要求一个发送数据D时候其他都为1，线与在一起为 $1*1*D$ 即可

普通与非门：不能线与，Vcc和GND形成通路有50mA的电流损坏T4,5

图腾结构不能线与的理解：有H有L会有大电流流过烧坏

1. 使用OC门（集电极开路输出的门电路）？？？速度慢
2. 三态门TSC 保留totem输出结构但是可以输出线与的特点，速度快

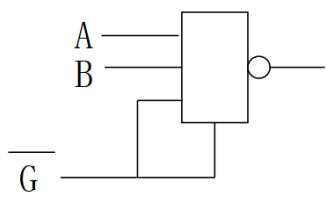
■ 三态门电路的基本原理



当控制G=1时， 电路是一个图腾结构的与非门

当G=0， T3、T4、T5均截止， 与非门输出F=Z（高阻态）

功能表



A	B	\bar{G}	F
X	X	0	Z
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	0

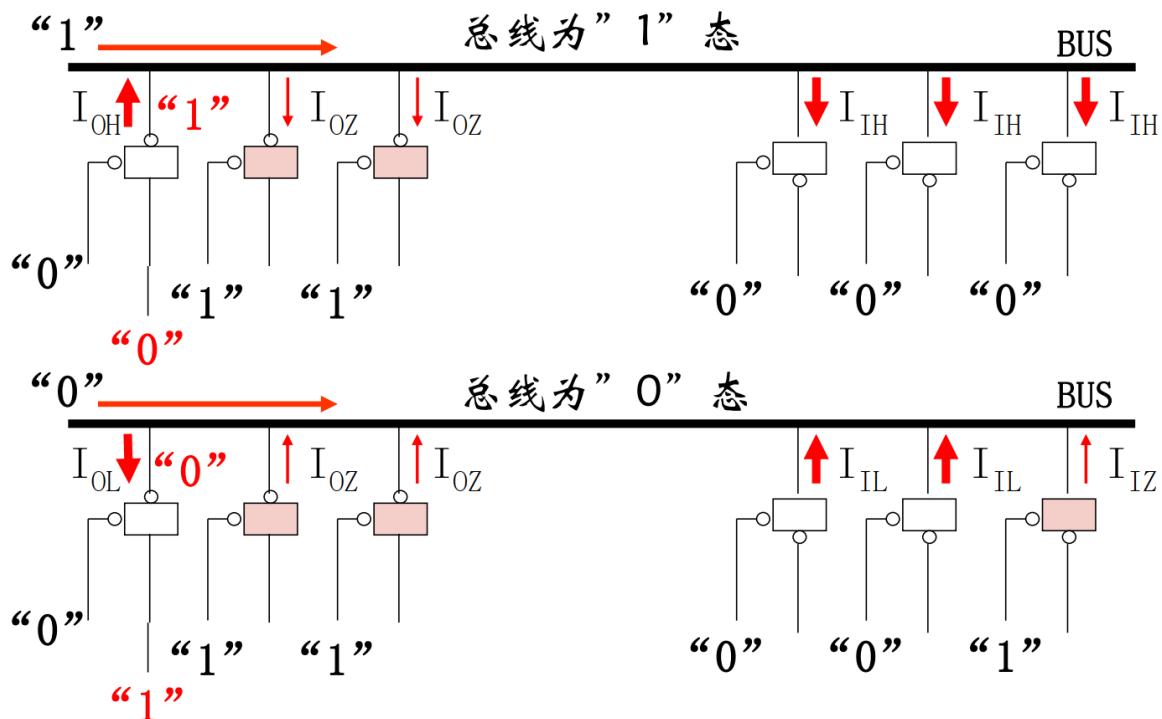
——高阻态
} 正常态

叫G非的控制端，为0的时候切换为高阻态，平时1的时候正常与非门

有的时候G非为0阻断，有时候是1阻断切换高阻态，看有无圈圈。总的来说是在与门接受0的时候阻断
多个三态门线与的时候：要先都到高阻态再有个别到正常态否则出现“浪涌电流”影响工作

		I_{IL}	I_{IH}	I_{OL}	I_{OH}	V_H	V_L
普通门		1.6mA	40μA	16mA	0.4mA	3.6V	0.3V
三态门	正常态	1.6mA	40μA	64mA	6.5mA	3.6V	0.3V
	Z态	40μA	40μA	40μA	40μA	5V 1.5V 0V	

I_{IZ} I_{OZ}



电流方向！=数据流向

Lect5 译码器 数据选择器

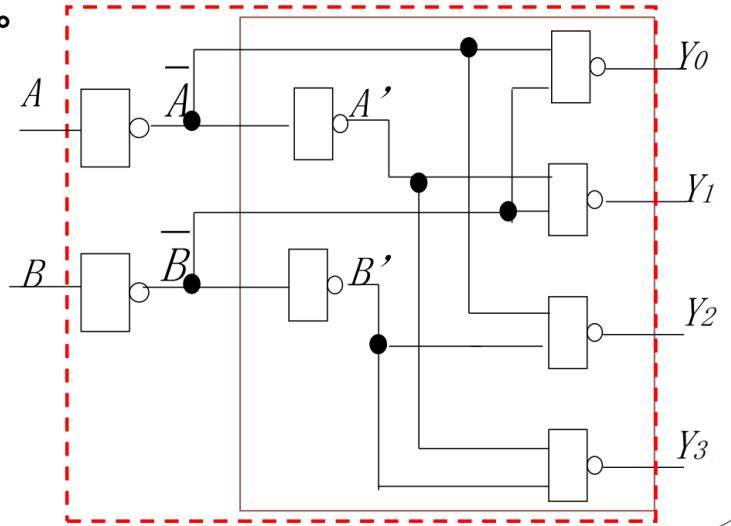
3.3 常用的中规模组合逻辑电路

3.3.1 译码器 变量译码 (所有组合2-4 3-8 4-16)、码制译码、显示译码

变量译码器 只有一个输入为0

□ 集成电路的设计原则：一个输入只能按照一个输入负载计算。

解决办法：增加一级输入缓冲



11

输入只能接1个负载，所以加1级缓冲之后再译码。用的都是与非门所以化简之后的表达式要加大非。

应用 CPU 2bit 控制信号到译码器，连接 4bit 控制信号到 4 个设备上

有使能端的 2-4 译码 $E \neq 1$ 时 所有 Y 为 1 $E \neq$ 的时候为 disable 而不是 enable 则输入为 1 时候连到与门里面是 0 封锁——扩展、选通

多片 2-4 扩展 3-8 高位输入 C 选片 AB 译码

5 片 2-4 扩展 4-16 用高位 CD 来接入 2-4 选片，剩下 AB 译码

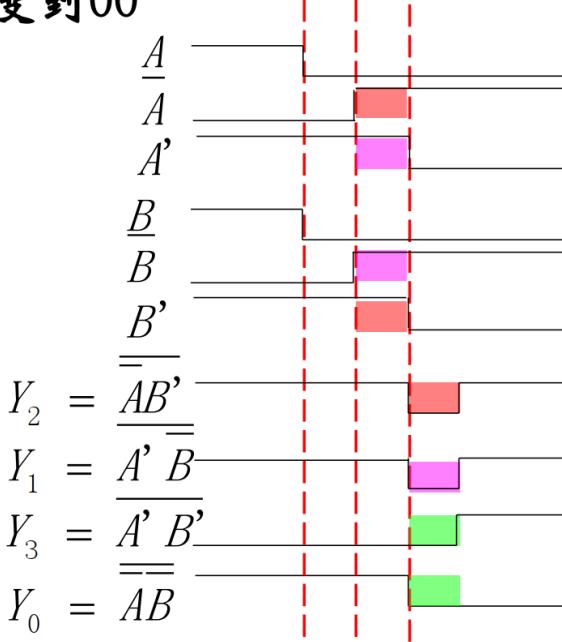
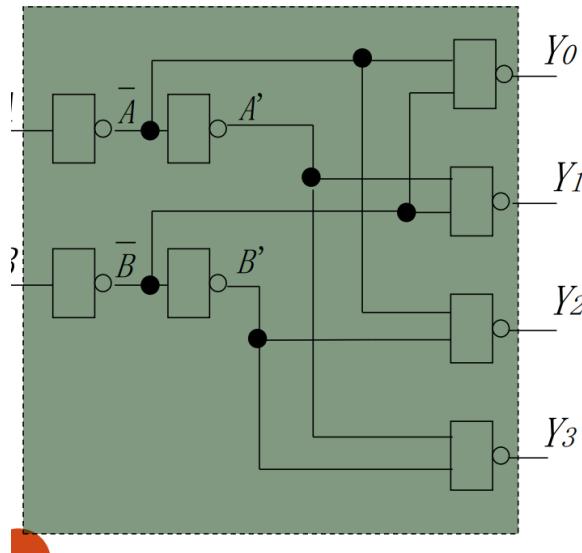
选通——解决竞争与冒险 A 和 $A' \neq$ 输入，与非产生负向尖峰 Spike 或非会出现正向尖峰：用 EN 解决

有时还会有“0”重叠 Overlap 现象，就同一时刻选择器中产生两个 0，因为 $A'B'$ 这种会有两极延迟相比于 AB ，导致 AB 和 $A'B'$ 产生 0 重叠的危险

注意画尖峰的时候不光前面的延迟，产生尖峰本身也要有延迟

■ 若无使能端，延迟产生尖峰和零重叠问题

□ 若A B同时到来：从11变到00

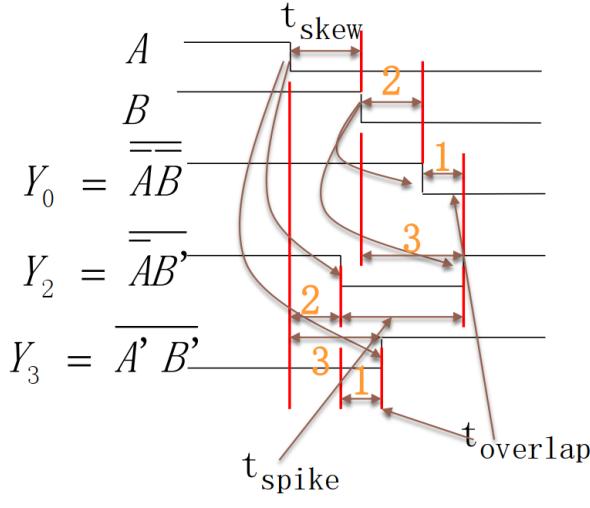
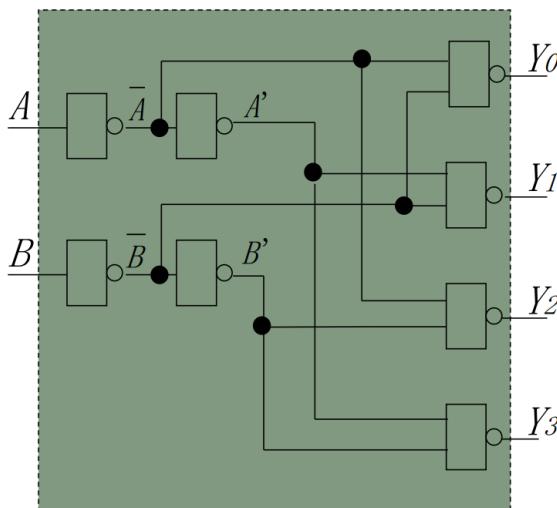


如果AB的改变没有同时到来存在skew的偏移，尖峰更宽 overlap两段

$$t_{\text{spike}} = t_{\text{skew}} + \text{delay} * 1 \quad t_{\text{overlap}} = \text{delay} * 1$$

■ 当AB从“11”变到“00”时，输出应从 $Y_3=0$ 变成 $Y_0=0$ 。

假设AB不能同时到来，存在偏移(Skew)，导致尖峰信号更宽。



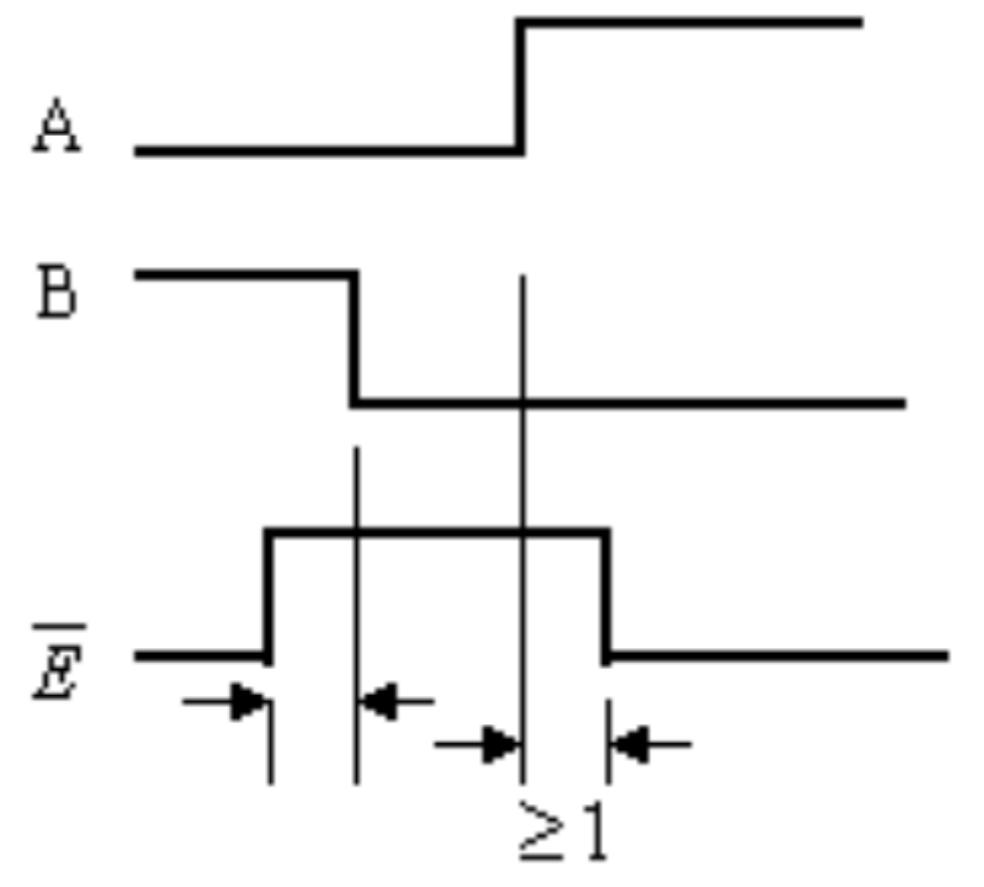
30

t_{spike} 加宽、两处出现零重叠

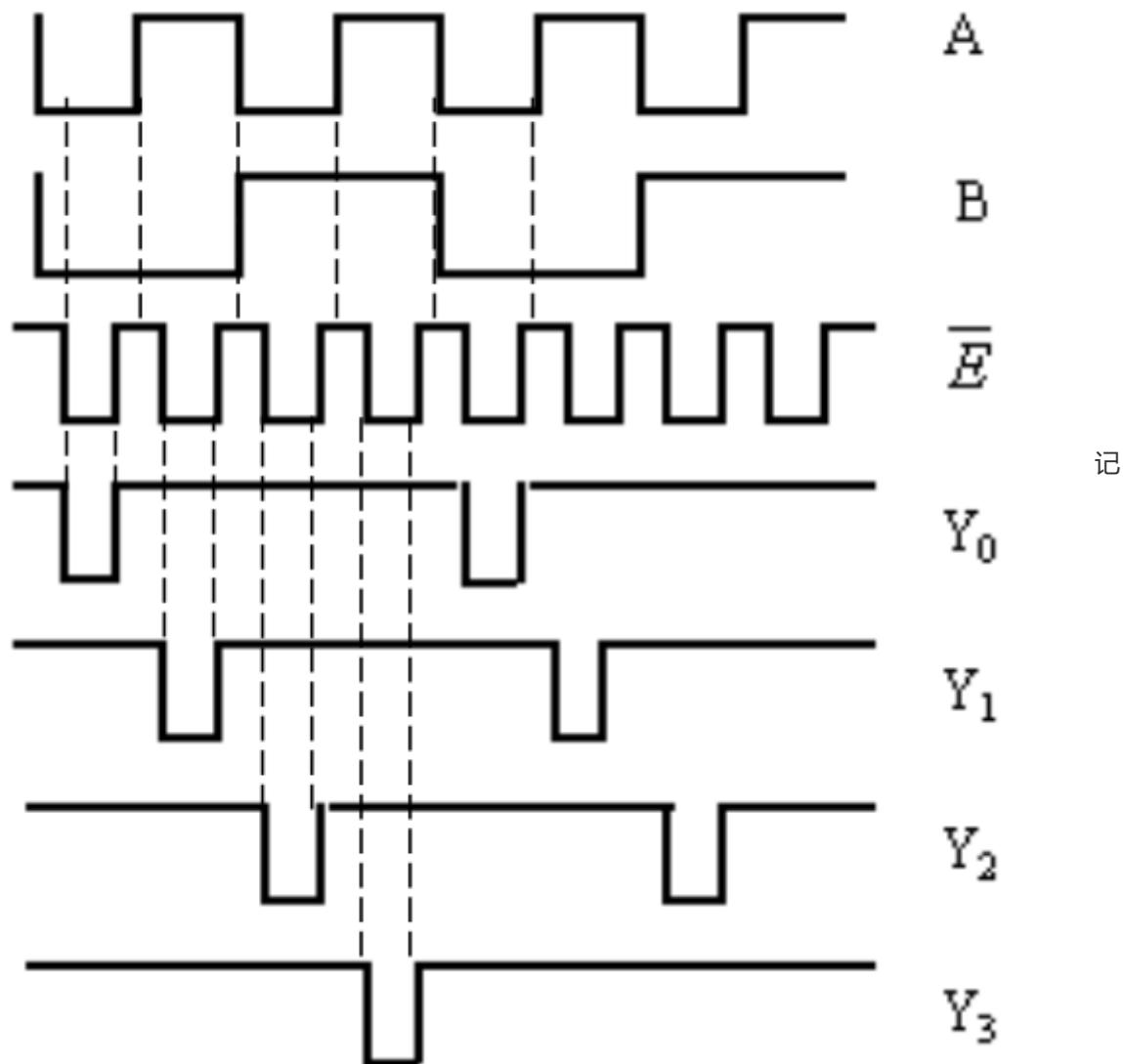
$$\begin{aligned} t_{\text{overlap}} &= 1 \text{ 级延迟} \\ t_{\text{spike}} &= t_{\text{skew}} + 1 \text{ 级延迟} \end{aligned}$$

用使能端消除AB变化中覆盖变化的EN非 = 1 强制 $Y_0 = Y_3 = 1$ 消干扰

这个使能信号要早于等于变量变化到来，对晚变化的信号至少滞后1级门延迟



使用 \bar{E}



得覆盖掉变化，但是会使得输出有效波形变窄（有效时间短）

3-8译码器一样的写ABC连接三输入与非门即可

扩展4-16? 用高位D选片，低三位译码。。

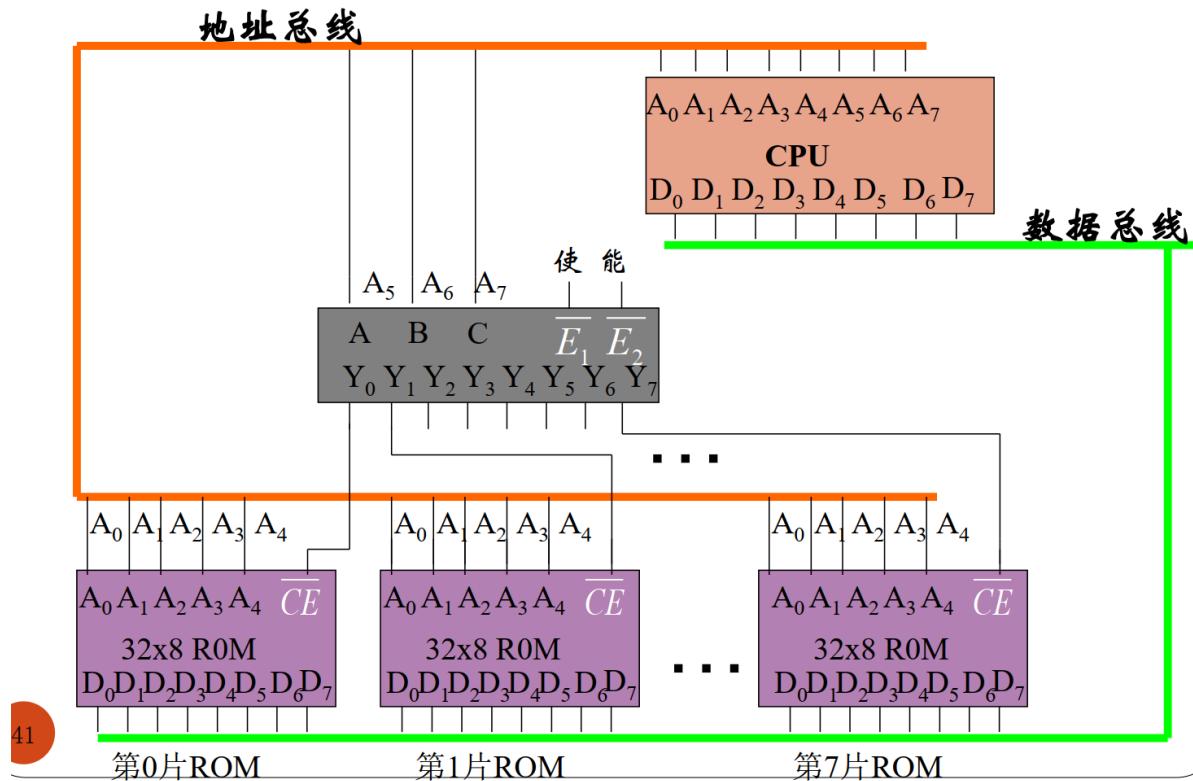
多使能端的译码器? 3EN的3-8: $E2A+E2B$ 为使不能1封锁// $E1$ 是使能0封锁

这样的选片更方便，直接使能和使不能连接高位D，总的使不能连接到分片的使不能上。

3-8译码分配地址0~7 256 CPU地址换算到每个ROM中的0~4共32地址空间

cpu高三位用来选片 (CE非) , 低5位用来寻址

用3-8译码器分配地址区(1)



译码器作为数据分配器（一分多）使能端为Data，C1C2控制多输出

Data变，C1C2选择把当前位分配给哪一个通道，如果通道恒1则被封锁

4-16译码器 真值表/功能表 组合逻辑中差不多，列所有输入和对应的输出

有两个使不能端，有一个1就锁死全1

问题：负载大，A非要负载9 ($2^{n/2+1}$)个门，A要负载8个门 $2^{n/2}$ ，使能端负载16门 2^n

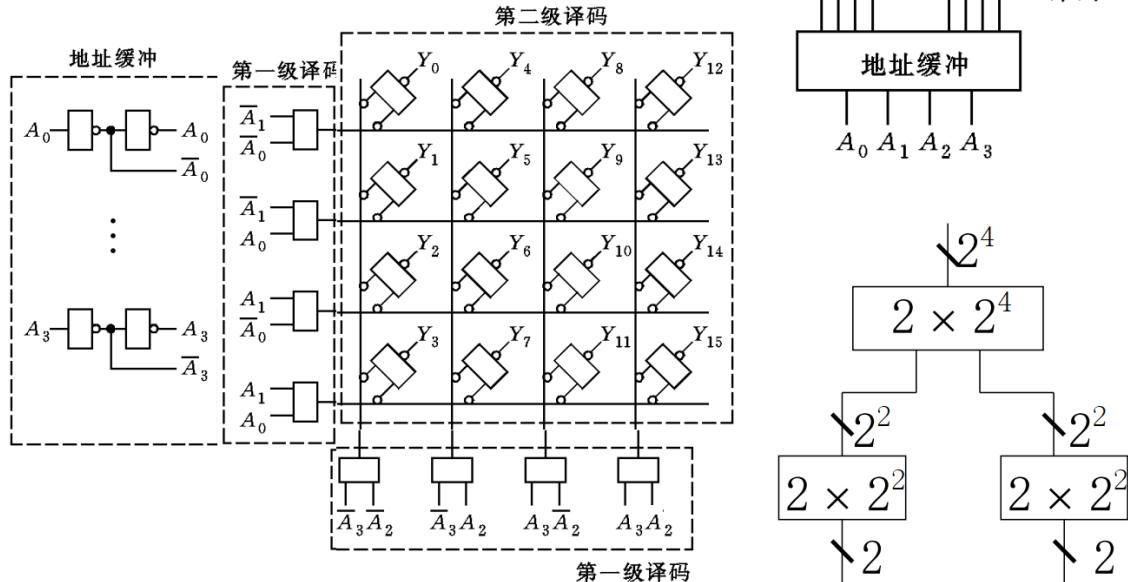
当输入变量增多，单级译码器不好实现，负载太大了。一般采用**多级译码**

4-16的多级译码，先前后翻译两个

先把四位输入翻译成EFGH WXYZ的两组合共8种有效值，再解码2-4

二级译码

用两级译码电路实现4-16译码器



(2×2^2 表示2输入与门4个, 2×2^4 表示2输入与非门16个)

每个A只有4个负载降低了一半, 每个与门也是4负载。相当于先粗分类成4+4类打包, 然后讨论这4+4的两两组合每个对应一种结果

8-256的分级? 划分为 (4-4) 对四输入粗分类成 (16-16) 然后讨论16*16的组合结果 每个A带8个负载 A非带9个 每个与门16个

码制译码器 编码换编码, 如二换十进制译码 (8421BCD to 1 bit Dec)

不完全译码比较简单, 有不少X项用来化简但是完全的必须每个连4位

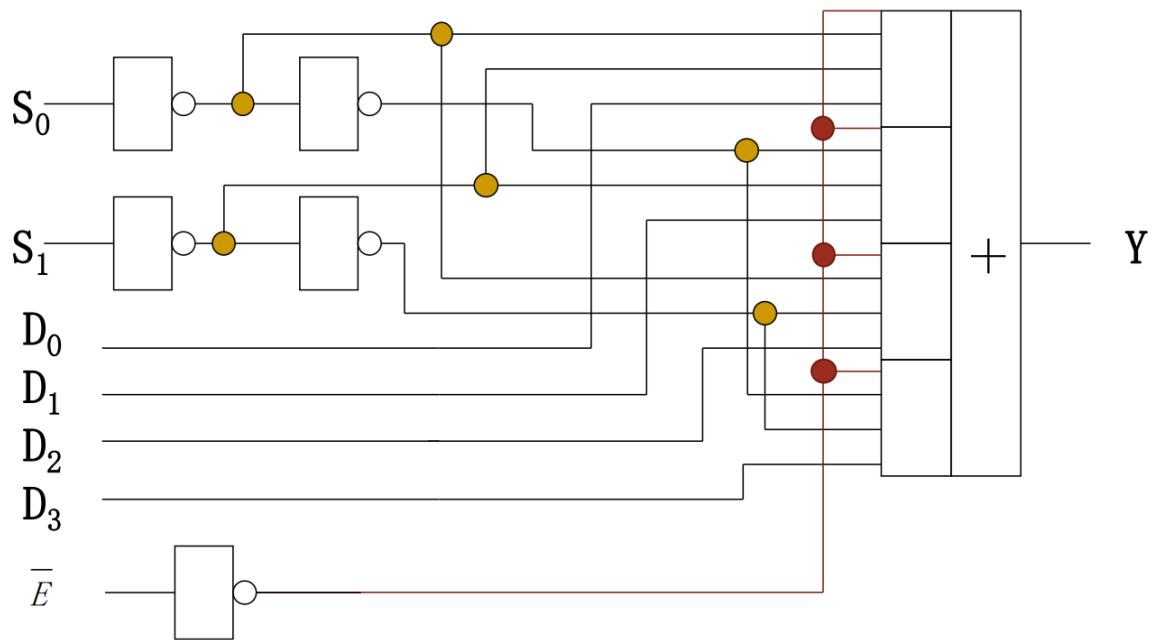
显示译码器 以共阳极, Low亮为例 4bit BCD转换为7bit亮灯 (码制) 对每一个输出都化简表达式即得。。

3.3.2 数据选择器 (多选一\多选多)

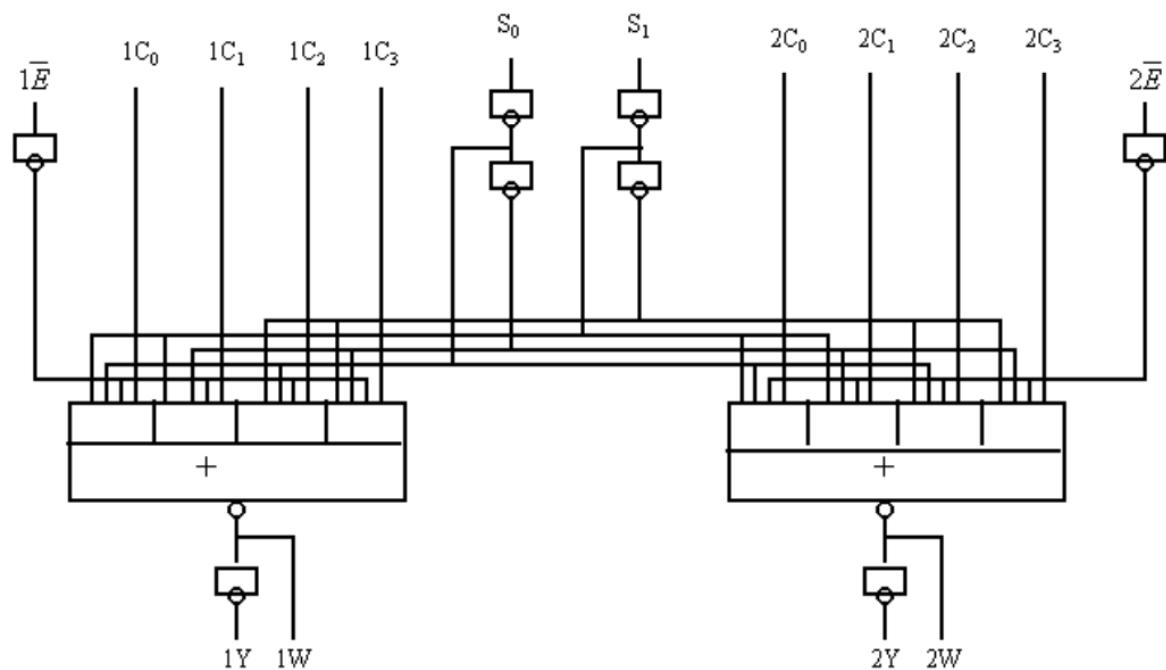
4选1 四个D 两个控制S 一个输出Y 写出 (化简) Y 连接即得

EN为1的时候使 $\overline{S_0} \overline{S_1}$ 为0, 输出恒为0

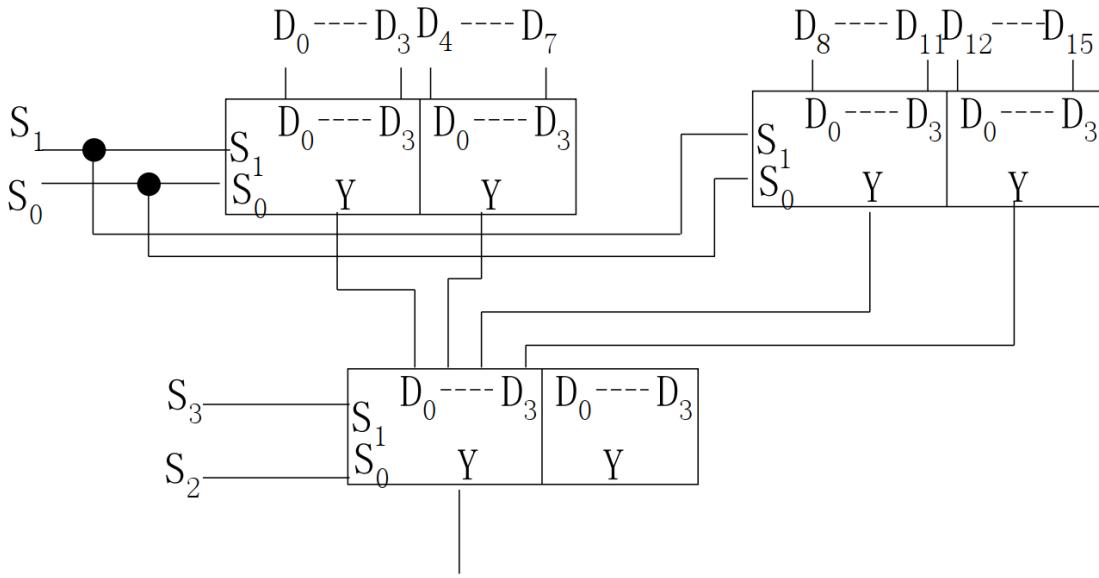
$$Y = S_0 S_1 D_0 + S_0 \overline{S_1} D_1 + \overline{S_0} S_1 D_2 + S_0 \overline{S_1} D_3$$



有两个EN的双4选1，提供正反两个输出，用3个双4选1得到一个16选4



16路输入，4个控制，这个选择的过程是从输出到入的。从1分成4然后每个都是一个1分4（无EN）



逻辑结构： $S_1 S_0$ 控制第一层选择， $S_3 S_2$ 控制第二层选择。

有使能端时候，S23译码成四个选择器的EN，然后同时3个使能不能，这样四个四选一其中三个都是0，如果三态门可以直接“线与”

总线发送控制

译码器是CPU往多个外设分配信息，这个是多个外设向CPU发送数据选择

译码器：看成N个输入组成的 2^N 个最小项。加一层与非即可实现与或逻辑

可以实现与或表达式的逻辑 一个译码器选择几个最小项接入与非即可

数据选择器 逻辑是与或表达 N 个控制端的 2^N 最小项 和 2^N 个输入组成的与或，可以用数据来调节01。只需把含有的最小项对应的D进1就好。

甚至8选1可以实现四变量函数：三个控制+一个数据输入

把对应的ABC代入函数得到1 0 或者D D非，然后把这个结果放到对应Di上

Lect6 编码器 比较器 奇偶校验

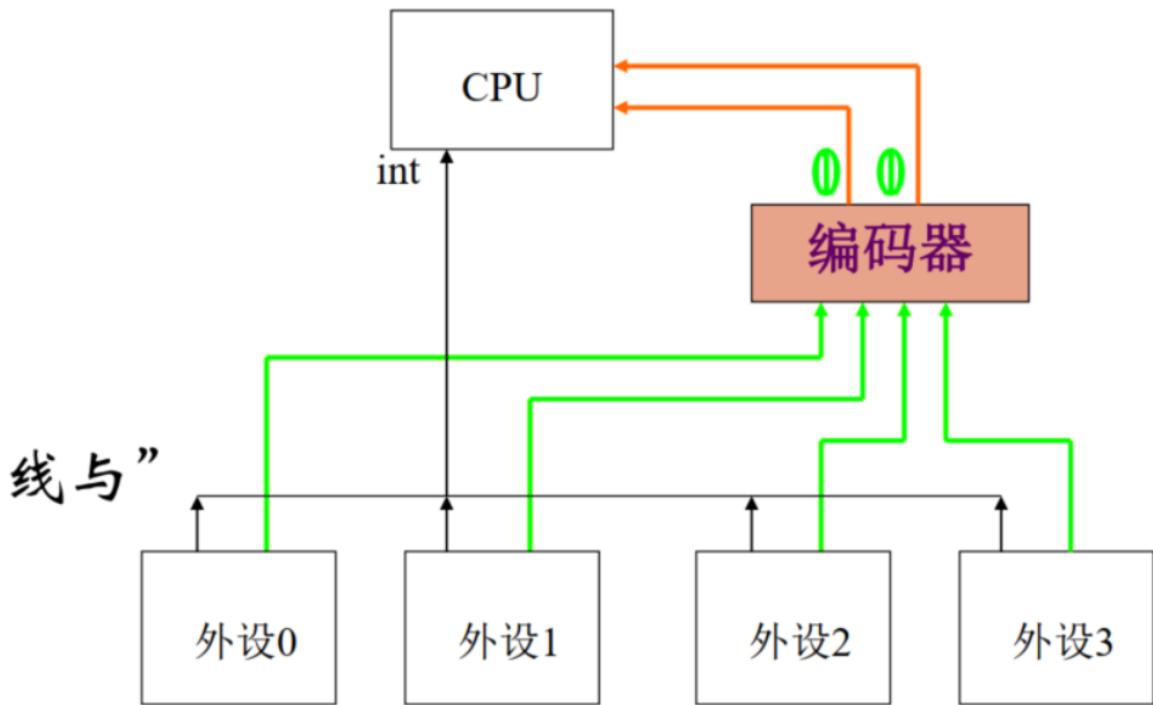
3.3.3 编码器

- 编码器原理

反向译码，对应编码如4-2 8-3 10-BCD等等

应用在CPU控制端告诉那个外设在上传数据（只一个0 其他都1）

键盘的输入检测，哪个键被按下...



2-4编码的时候，如果取大非可以直接对里面的取补 8-3类似？？？

8421编码器 0-9的十个信号转成4 bit 8421码 L6 P12

$A+B+C+D = AB \text{或非 } CD \text{或非再与非到一起}$

- 优先编码器：两条及以上为0的时候，优先输出高位编码

\bar{E}_i	0	1	2	3	4	5	6	7	A_0	A_1	A_2	G_S	E_O
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	1	0	0	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	1	1	0	0	1
0	X	X	X	0	1	1	1	1	0	0	1	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
1	X	X	X	X	X	X	X	X	1	1	1	1	1

模拟键盘，如果同时被按下输出高位编码，以3-8编码为例

除了8个输入，还需要使不能 E_i 非，

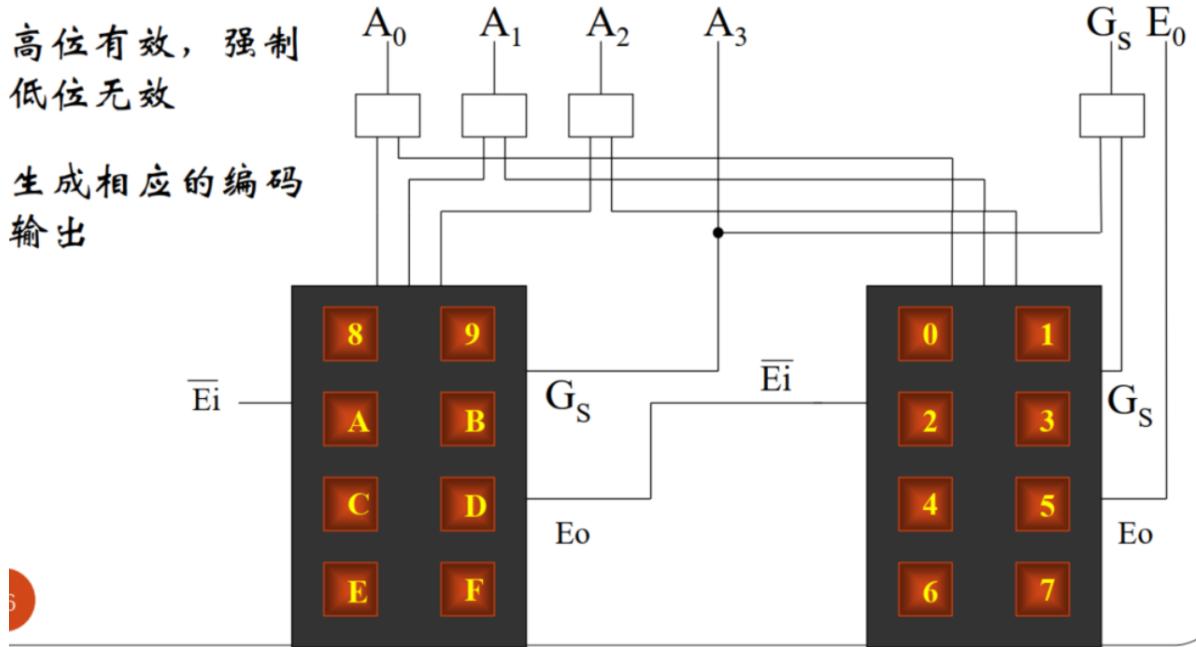
输出有效 G_S （当本八位有有效输出的时候为0，可以区分是0被按下还是没有按键被按下的状态），**输出级联 E_O** （如果是0则允许级联，意思是按键被按下且没有使不能）

如果7是0则剩下无所谓，一定翻译000出去（反码编码，平时111）

由于反码编码，化简表达式也是化简 A_i 非=找0行

级联 高位板子允许低位级联把 E_O 连到低位 E_i 使不能 允许0的时候低位使能正常工作。**高位级联接低位使能 两个有效与为总的有效**

注意还是个3-8编码，编码结果只能接低三位，高位要高位板子的是否按下的有效Gs判断。切换的是提供低三位的片



应用 按照优先级编码，中断相应，键盘输入读取等

3.3.4 数据比较器

$$W_i = "A_i < B_i" = B_i \cdot \overline{A_i \cdot B_i}$$

$$Y_i = "A_i = B_i" = \overline{(A_i \cdot \overline{A_i \cdot B_i}) + (\overline{A_i \cdot B_i} \cdot B_i)}$$

$$Z_i = "A_i > B_i" = A_i \cdot \overline{A_i \cdot B_i}$$

可以级联，以4位为例，高位比不出来才比低位

单bit比较: >10 <01 not xor 同或为1

设计的级联是低位连接高位，即最高的四位作为最终输出。

如果当前片比不出来才会去读取输入结果，否则以当前片结果为准

3.3.5 奇偶校验

8个输入xor到4个xor到2个xor到一个即为结果，1是奇数 如果多位则在对应层级补进去

组合逻辑电路的竞争冒险

竞争 输入信号由于延迟，走不同路影响到输出的时间不同。

- 非临界 不产生错误输出 临界 产生错误输出

冒险 由竞争导致了错误信号

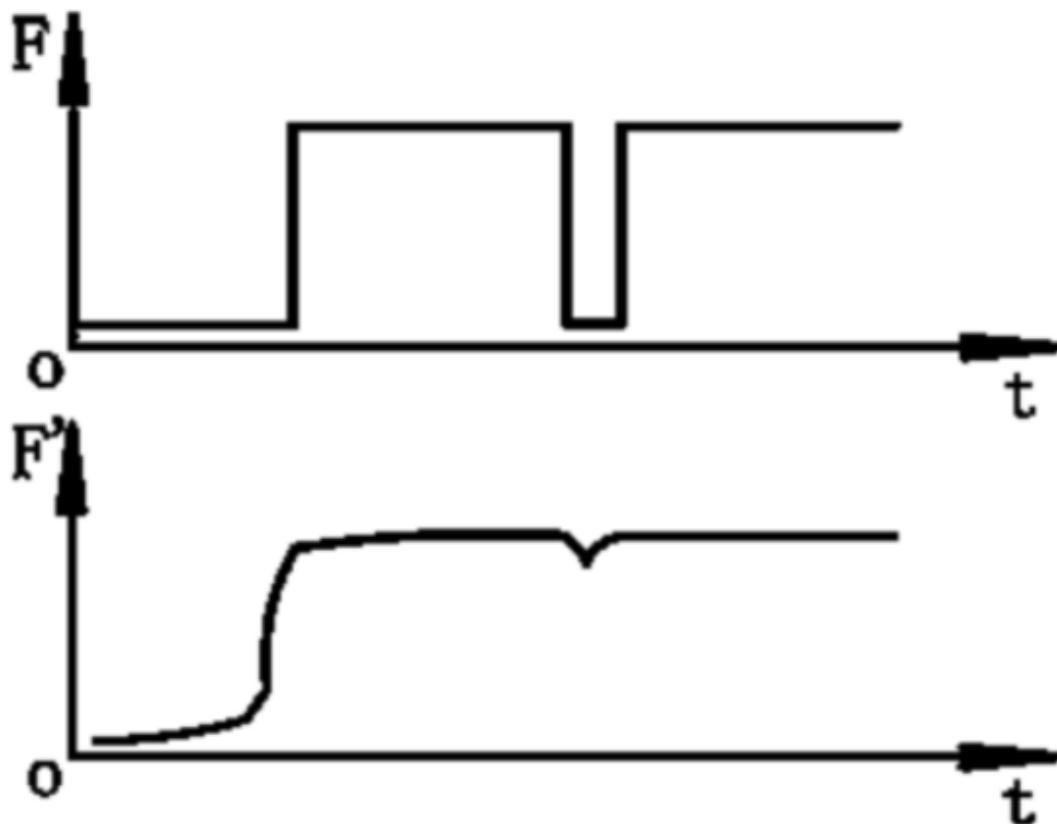
！组合电路中的险象是瞬态、暂时的尖脉冲，稳定后可恢复正常逻辑关系

判断存在险象 代数法：函数表达式可能存在 X 和 X^* 非尝试消除其他变量得到 $X+X^*$ 或者 X^*X^* 的形式。

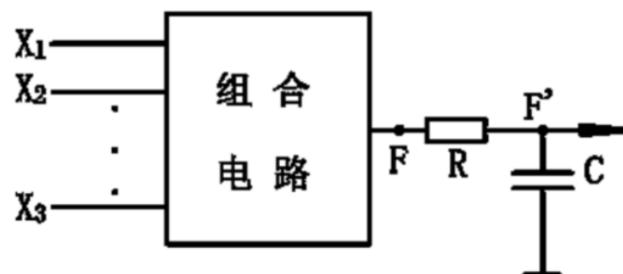
卡诺图把函数表达式化成卡诺圈进去，如果有圈相切（相邻但是不被同圈包含）那这两个相邻项就是危险的

避免险象

- **增加冗余项** 卡诺圈相切的时候手动把他们圈一起，冗余项可以保证当只有一个输入量变化的时候不再产生险象。在代数上理解是：把化简到 $A+A\bar{A}$ 时候限制的其他条件加这么一项
- **电容滤波** 过滤高频信号 要求惯性环节时间常数 T 大于尖脉冲 不要太大否则畸变严重



消除险象的另一种方法是在组合电路输出端连接一个惯性延时环节。通常采用RC电路作惯性延时环节，如图所示。



图中的RC电路实际上是一个低通滤波器。由于竞争引起的险象都是一些频率很高的尖脉冲信号，因此，险象在通过RC电路后能基本被滤掉，保留下来的仅仅是一些幅度极小的毛刺，它们不再对电路的可靠性产生影响

- **选通** 在最终结果的与门上再增加一路选通脉冲的输入，我只在电路稳定之后出现选通高电平脉冲，避开险象脉冲

3.3.6 可编程逻辑器件PLD

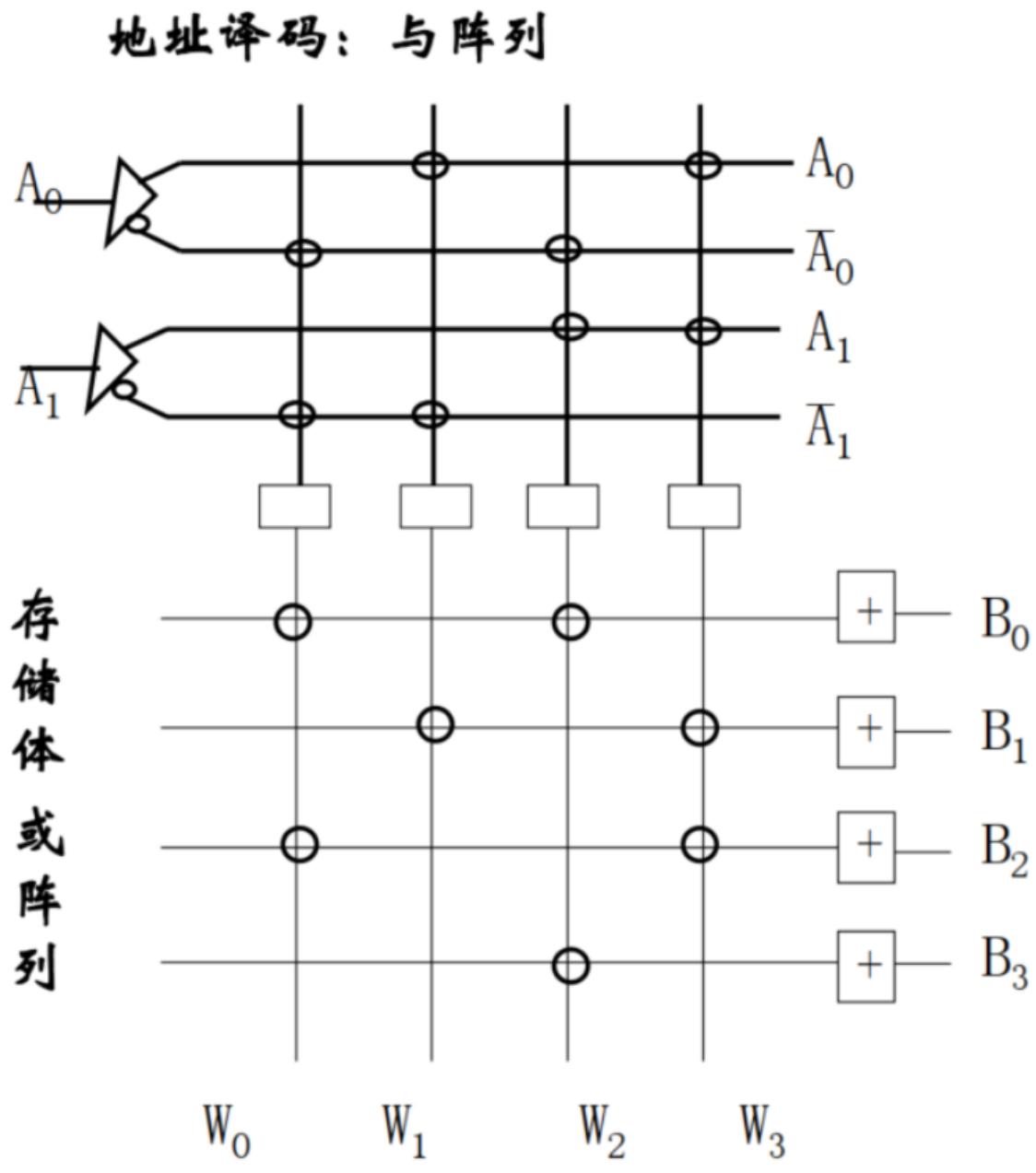
包括ROM PLA逻辑阵列 PAL 阵列逻辑 GAL通用阵列逻辑) L6 P60

- ROM 写入后只读（简单、规律、大容量） 区别可读可写RAM

与阵列+或阵列，输入后与阵列附近只有一条是1的其他都0

然后每个输出都是或阵列，可以直接出

其实本质上是对不同输入的分情况讨论，上面按照从小到大排列输入，下面每一列都是一种输出结果。讨论好标上对应点出去就好了



应用 比如翻译8421到格雷码之类

3输入4输出的rom为 8×4 的rom 可以实现0-7的平方

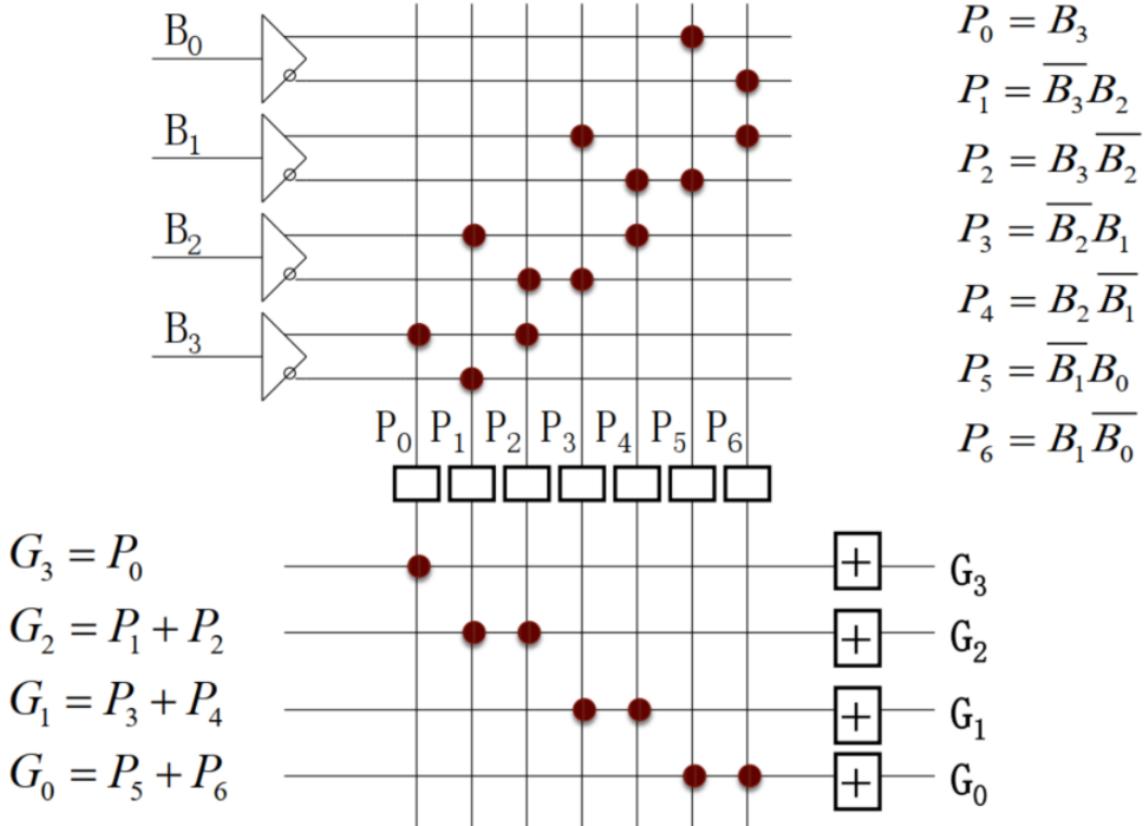
储存计算机程序，每个地址（一列）为一个字节8bit

- PLA可编程逻辑阵列 逻辑压缩 PROM或阵列可编程
- PAL是与阵列可编程或阵列固定，分组合的和时序的
- GAL是可以同时实现组合+时序的芯片

rom是完全讨论的，输入相当于是地址需要完全寻址 2^N

PLA是压缩的，逻辑函数需要什么我们才保留什么，我们需要最简与或表达式。

每一列代表输出中的一项，上面是项的与表示，下面输出的或



91

与阵列 $8 \times 7 = 56$, 或阵列 $4 \times 7 = 28$, 总点数 84

储存讨论的信息的时候也可以压缩最简单的输出项表达式，然后用每一列代表一个乘积项最后或在一起。一组地址多个P ok

PLA的容量表示：

(输入数) \times (P项数) \times (输出数)

Lect7 运算器

3.3.6 加法器

半加器：不考虑进位，异或就好了

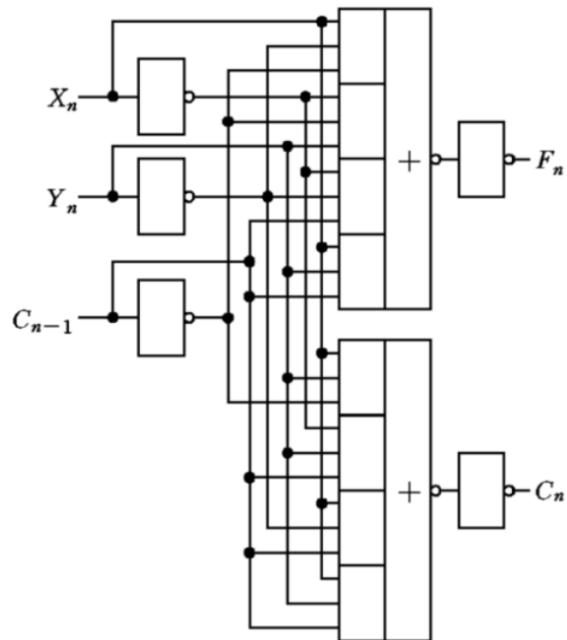
一位全加器 写出表达式不化简直接得

● 一位全加器实现方案1

$$F_n = X_n \overline{Y_n} \overline{C}_{n-1} + \overline{X_n} Y_n \overline{C}_{n-1} \\ + \overline{X_n} \overline{Y_n} C_{n-1} + X_n Y_n C_{n-1}$$

$$C_n = X_n Y_n \overline{C}_{n-1} + X_n \overline{Y_n} C_{n-1} \\ + \overline{X_n} Y_n C_{n-1} + X_n Y_n C_{n-1}$$

不化简，用全部最小项实现，需要3级门。

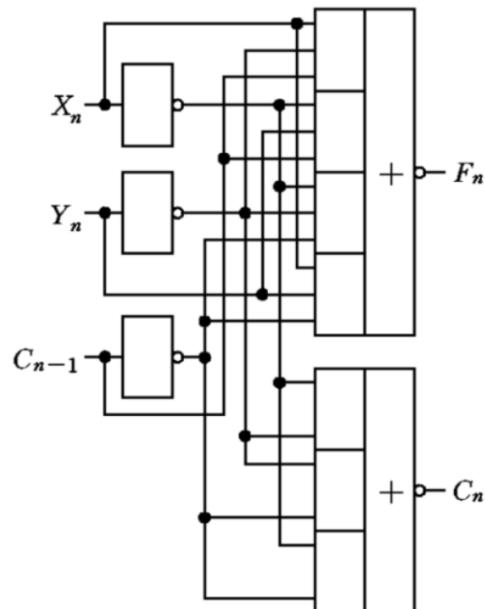


卡诺图化简一下都表示成与非、与或非等形式。直接找的是F非和C非（找0）这样都是2级门延迟

● 一位全加器实现方案2(续)

$$F = \overline{\overline{F}} = \overline{\overline{X_n Y_n C_{n-1}}} + \overline{\overline{X_n} Y_n C_{n-1}} + \overline{X_n \overline{Y_n} C_{n-1}} + \overline{X_n Y_n \overline{C_{n-1}}}$$

$$\overline{C}_n = \overline{\overline{C}_n} = \overline{\overline{X_n Y_n}} + \overline{\overline{X_n} C_{n-1}} + \overline{\overline{Y_n} C_{n-1}}$$



11

尝试用Cn表示Fn，即先把Cn算出来之后用其结果和ABC组合得Fn

注意这里一种思路是写Cn和Fn，另一种是写Cn非和Fn非

四种形式的1位全加器(2)

真值表

X _n	Y _n	C _{n-1}	F _n	C _n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

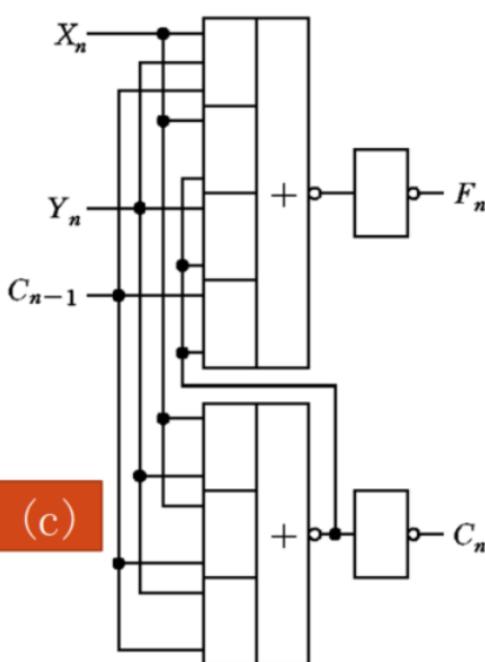
分析全加器真值表中F_n和C_n的对应关系，F_n为“1”的条件有两个：
 1、X_n Y_n C_{n-1}均为“1”
 2、X_n Y_n C_{n-1}只有一个为“1”（有至少1个“1”且C_n为“0”）
 用C_n表示F_n，先组合出C_n，再有F_n

$$F_n = X_n Y_n C_{n-1} + X_n \bar{C}_n + Y_n \bar{C}_n + C_{n-1} \bar{C}_n$$

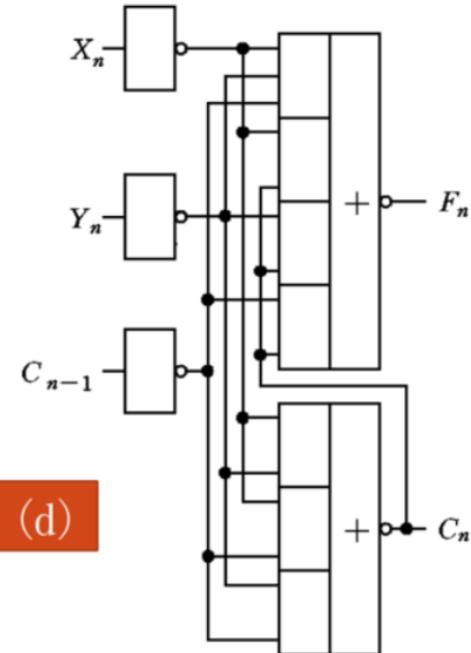
$$C_n = X_n Y_n + (X_n + Y_n) C_{n-1}$$

$$F_n = \bar{X}_n \bar{Y}_n \bar{C}_{n-1} + \bar{X}_n C_n + \bar{Y}_n C_n + \bar{C}_{n-1} C_n$$

$$C_n = \bar{X}_n \bar{Y}_n + (\bar{X}_n + \bar{Y}_n) \bar{C}_{n-1}$$



C_n的形成需要二级门延迟
 F_n的形成需要三级门延迟



C_n的形成需要二级门延迟
 F_n的形成需要三级门延迟

四位串行加法器

如果都用方案2 最终F4 C4都是8级门延迟

都用方案3，最终为F4 9级 C4 8级

方案34交叉串联呢？C1两级但是C1非其实1级就好直接扔D得到C2也是2级。C3非一级就好C3 C4都是4级那么C5就是5级

这种方式主要是交替利用C和C非作为传递使得C的计算飞快。F总是当前的C+1级即可

超前进位加法器 提前算出C1234 在表达式中看出：

搞一个产生进位函数 $G = XY$ 传递进位函数 $P = X + Y$ 记得写表达式的时候直接找 $C = \text{大非}(\text{XXX})$ XXX 为 C 的表达式

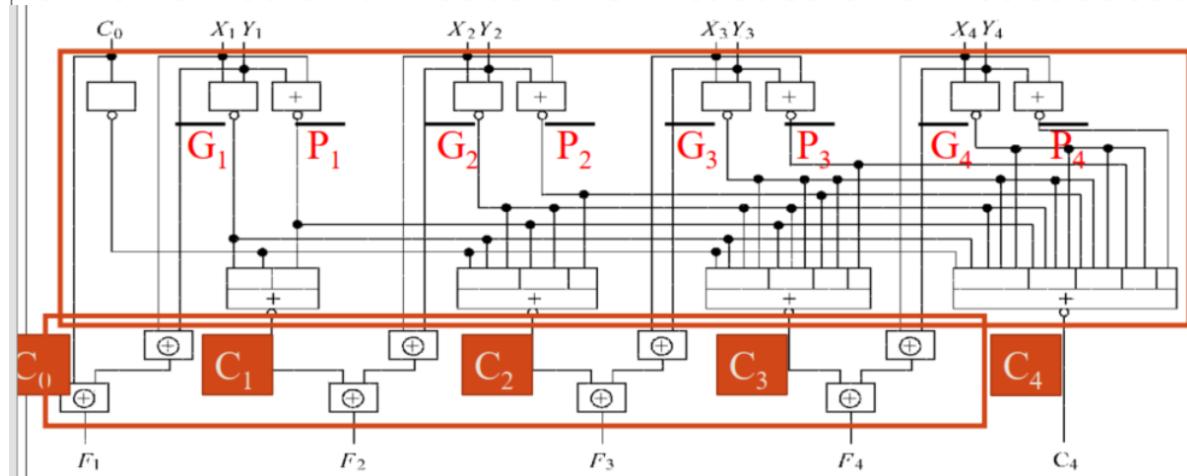
$$C_1 = X_1 Y_1 + (X_1 + Y_1) C_0$$

$$C_2 = X_2 Y_2 + (X_2 + Y_2) X_1 Y_1 + (X_2 + Y_2) (X_1 + Y_1) C_0$$

$$C_3 = X_3 Y_3 + (X_3 + Y_3) X_2 Y_2 + (X_3 + Y_3) (X_2 + Y_2) X_1 Y_1 \\ + (X_3 + Y_3) (X_2 + Y_2) (X_1 + Y_1) C_0$$

$$C_4 = X_4 Y_4 + (X_4 + Y_4) X_3 Y_3 + (X_4 + Y_4) (X_3 + Y_3) X_2 Y_2 \\ + (X_4 + Y_4) (X_3 + Y_3) (X_2 + Y_2) (X_1 + Y_1) C_0$$

$$\left. \begin{aligned} C_1 &= \overline{\overline{X_1 + Y_1}} + \overline{\overline{X_1 Y_1}} C_0 = \overline{\overline{P_1}} + \overline{\overline{G_1}} C_0 \\ C_2 &= \overline{\overline{X_2 + Y_2}} + \overline{\overline{X_2 Y_2}} (\overline{\overline{X_1 + Y_1}}) + \overline{\overline{X_2 Y_2}} \overline{\overline{X_1 Y_1}} C_0 = \overline{\overline{P_2}} + \overline{\overline{G_2}} \overline{\overline{P_1}} + \overline{\overline{G_2}} \overline{\overline{G_1}} C_0 \\ C_3 &= \overline{\overline{X_3 + Y_3}} + \overline{\overline{X_3 Y_3}} (\overline{\overline{X_2 + Y_2}}) + \overline{\overline{X_3 Y_3}} \overline{\overline{X_2 Y_2}} (\overline{\overline{X_1 + Y_1}}) + \overline{\overline{X_3 Y_3}} \overline{\overline{X_2 Y_2}} \overline{\overline{X_1 Y_1}} C_0 \\ C_4 &= \overline{\overline{X_4 + Y_4}} + \overline{\overline{X_4 Y_4}} (\overline{\overline{X_3 + Y_3}}) + \overline{\overline{X_4 Y_4}} \overline{\overline{X_3 Y_3}} (\overline{\overline{X_2 + Y_2}}) + \overline{\overline{X_4 Y_4}} \overline{\overline{X_3 Y_3}} \overline{\overline{X_2 Y_2}} (\overline{\overline{X_1 + Y_1}}) + \overline{\overline{X_4 Y_4}} \overline{\overline{X_3 Y_3}} \overline{\overline{X_2 Y_2}} \overline{\overline{X_1 Y_1}} C_0 \end{aligned} \right\}$$

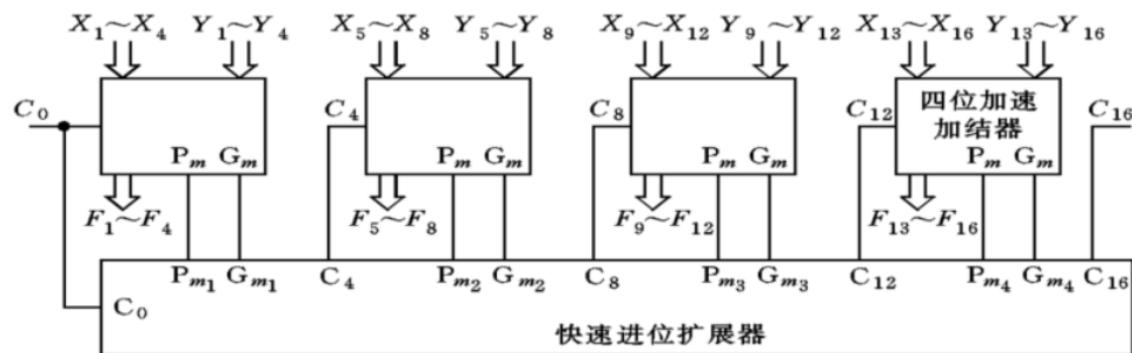


C_i 都是2级 F_i 都是3级

16位加法器 记得算延迟都要先算各级的 C 再算 F

4个超前进位4位加法器也可以再加一个超前进位器 P_{34} 为构成方法

这时 $C_4 8 12 16$ 的延迟都是3 F_1-4 是3 后面都是6级



3.3.7 算术运算逻辑单元 4位

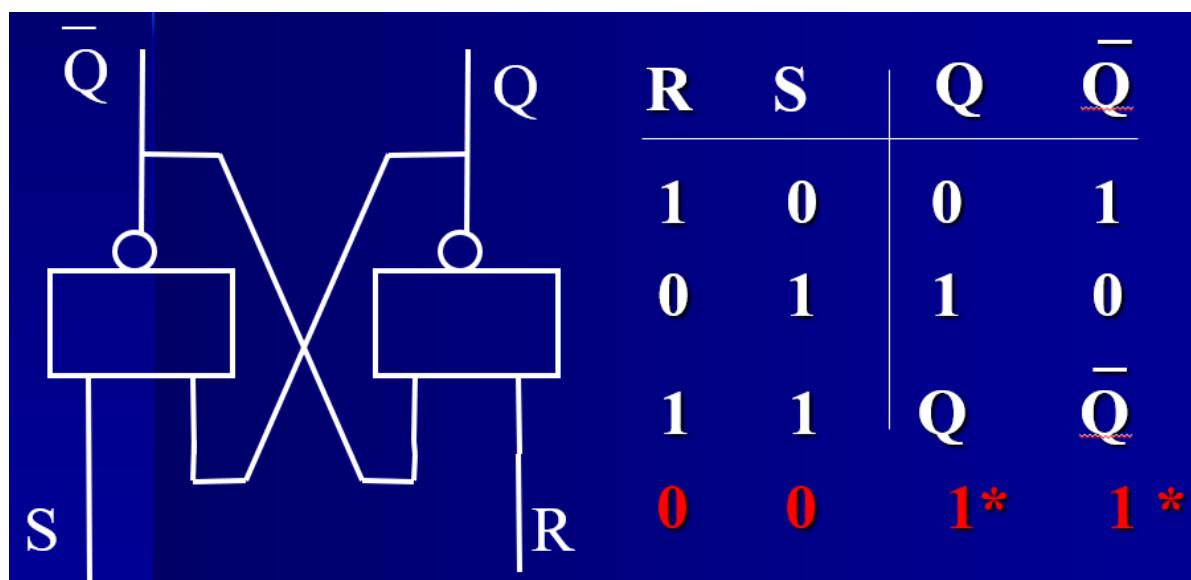
功能控制：加减 比较 与或非 移位 核心是四位并行加法器

- 控制进位逻辑（简单但少
 - 10得同或结果时： $X+Y$ (非00即可) 和 $!(XY)$ (非11即可) 与 然后和1进行异或。
 - 11得与结果时： $!(XY)$ 和1取异或
- 改变 G_i 和 P_i 获得多种运算能力（多但复杂
 - 四个S是控制如何产生 G_i 和 P_i 的
 - M状态控制， 是否进行逻辑运算

L7 P51 有各种16种的运算 书P141 的元件74181

Lect9

RS基本触发器 问题：0 0时非正常状态（本时刻一定是11，下一时刻如果11读取 就为不定状态了） 11
读取 两个与非交叉



时钟控制方式：电位触发、边沿触发、主从触发器

功能分类：D、R-S、J-K、T (Toggle)

重点：边沿触发的D和JK触发器

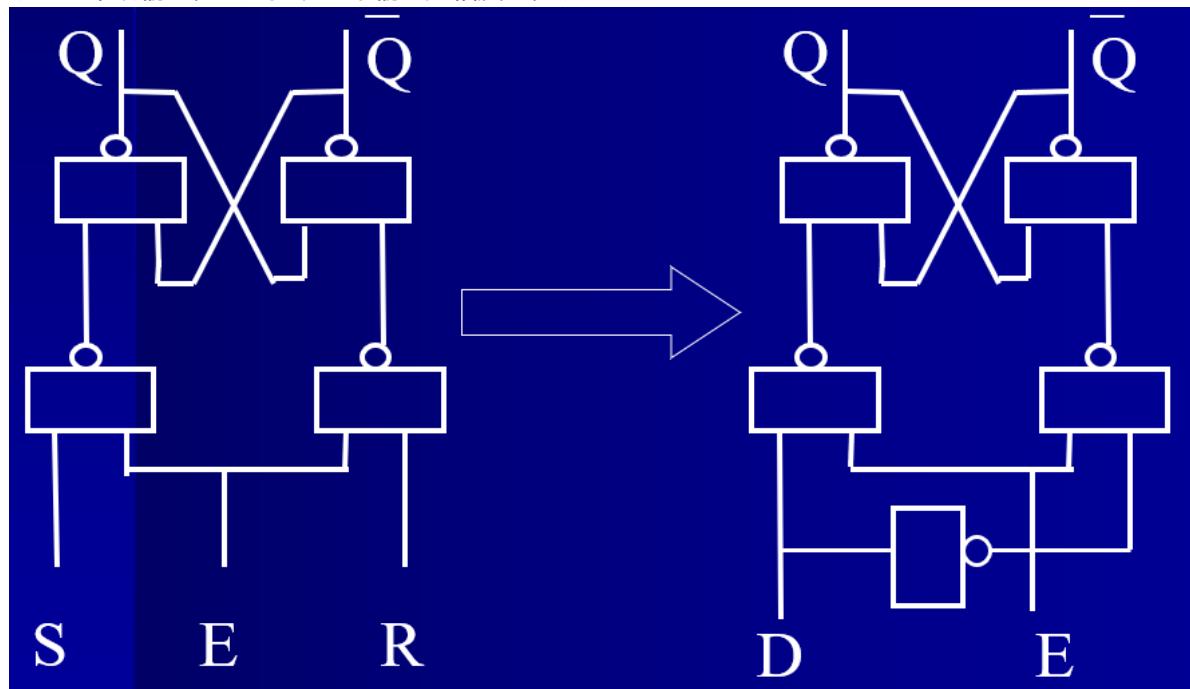
如何解决00时的异常？限制输入or保持稳定不坏

电位方式触发 Clock Pulse

修改R-S的输入，增加使能端，就EN=1的时候一定是保持状态，优化成与非，E成为真正使能，0的时候一定是11保持

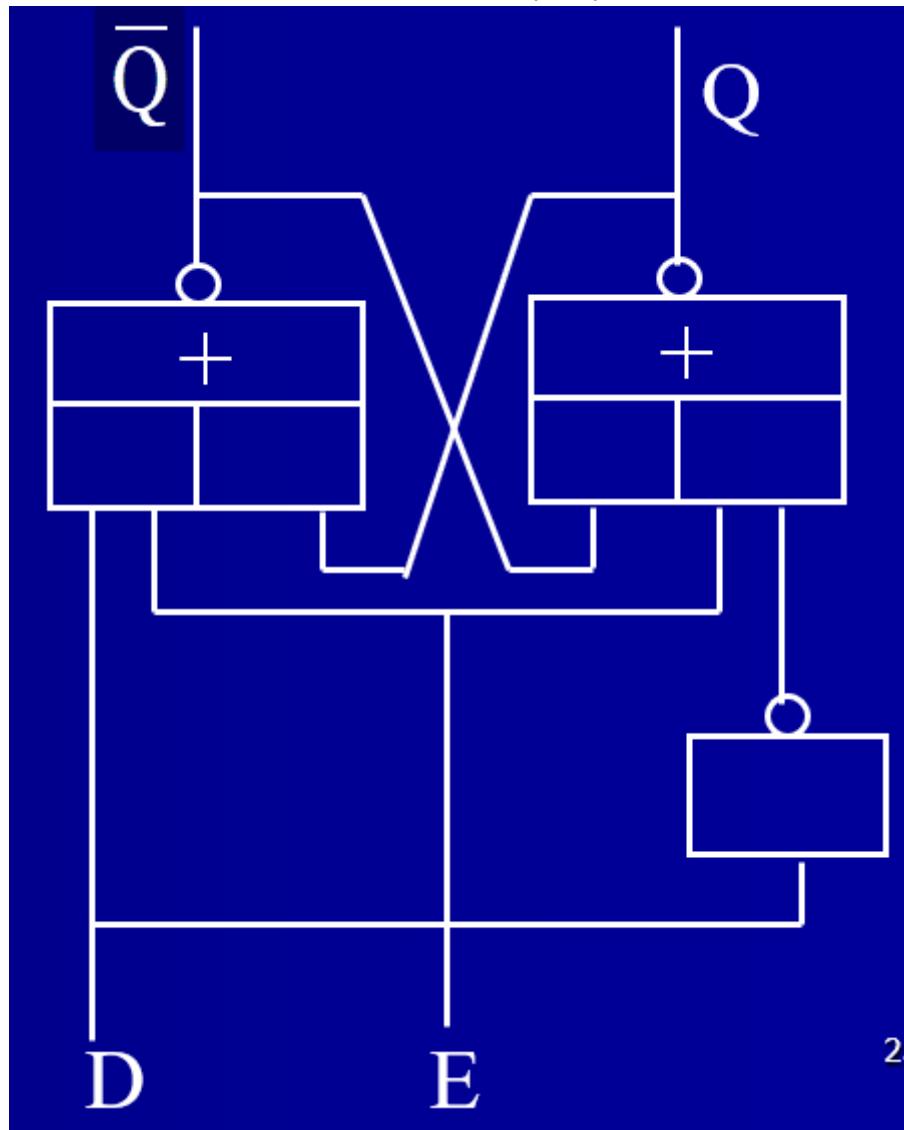
这时候的修改RS电位触发器变成了00保持 11异常

改进？单端输入，一定控制两个输入是相反的，改成D



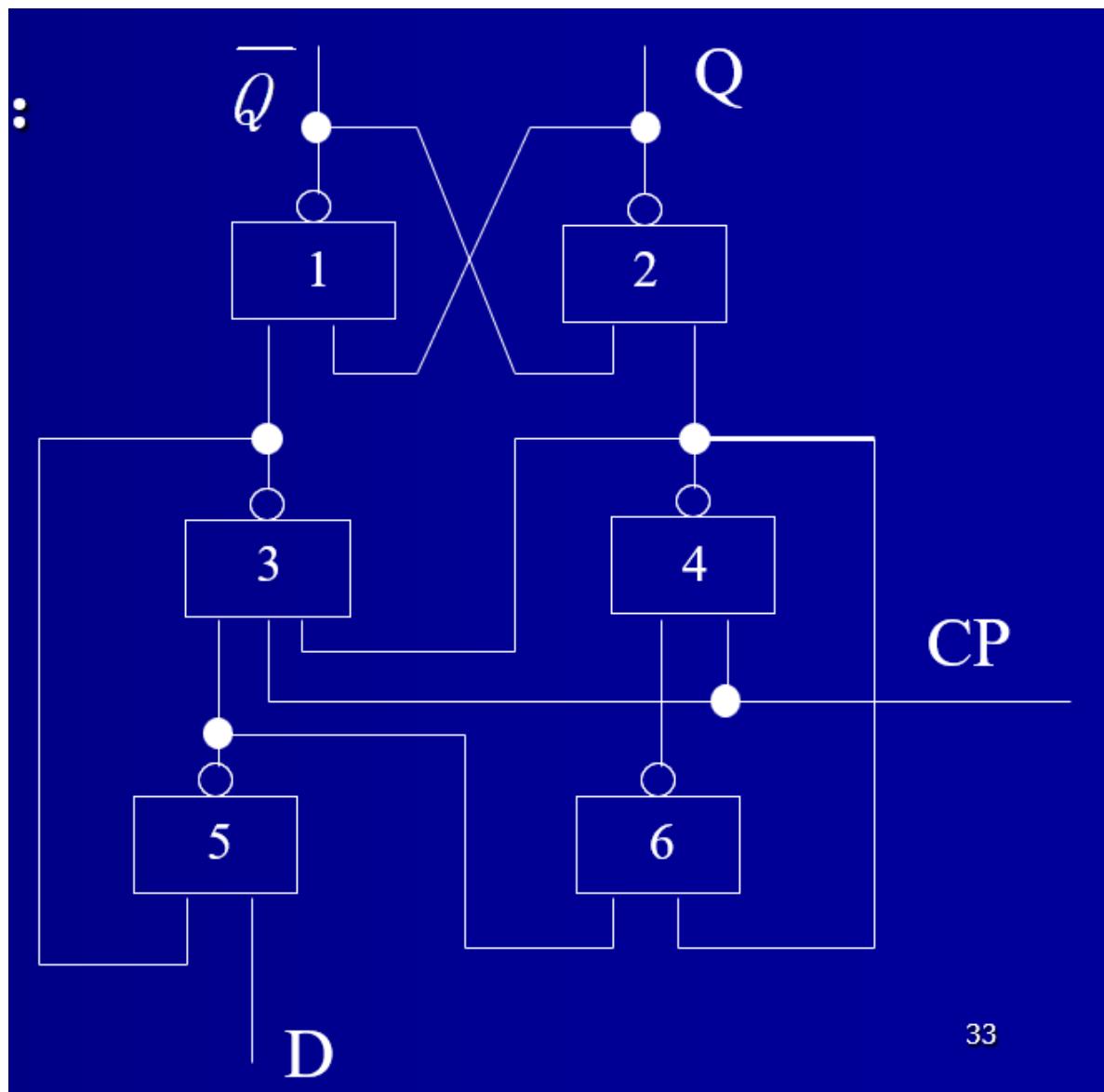
但是延迟太大了吧，D到Q非最大要4级，再改进得到

电位D触发器 E仍然是使能，0的时候是保持（读取）



问题：抗干扰差 使能脉冲某个跳变来时，输出才变化为输入的值

正边沿D触发器 某个约定跳变来才接受数据，没约定跳变就保持



33

CP = 0 时 上面为基本RS的11状态，保持

如果正边沿到来，在信号到来产生一级门作用的时候改变状态

● 正沿D触发器分析

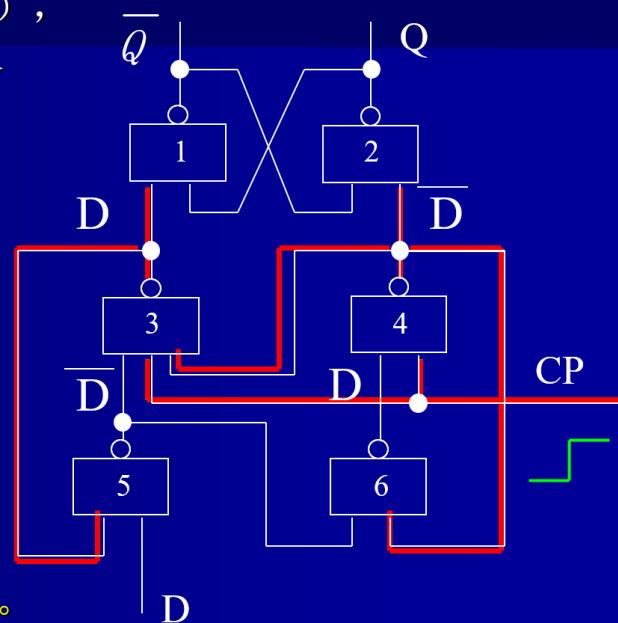
用或门和与门

CP从“0”跳变为“1”后（正沿），门3的两个输入为“1”，另一个输入为 \bar{D} ，门3的输出为D。

同时门6和门4中各有一个的输入为“1”，门6的另一个输入为 \bar{D} ，门4的另一个输入为D，门4的输出为 \bar{D} 。

即顶端的RS触发器的输入为 $S=D$, $R=\bar{D}$ ，则 $Q=D$, $\bar{Q}=\bar{D}$ ；

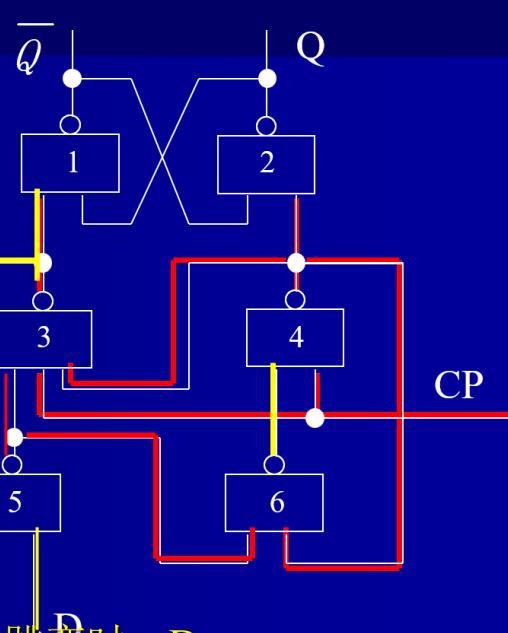
结论：CP从“0”跳变为“1”后，触发器的输出端为输入的数据D。



$CP = "1"$ 时，门3的输出为D，门4的输出为 \bar{D} 。假设在时钟正边沿跳变时， $D = "0"$ ，则门3的输出由“1”变为“0”。不论D如何变化，门5的输出为“1”，门3的输出保持为“0”。门6的输出为“0”，门4的输出为“1”。

顶端RS触发器的输出保持 $Q = "0"$, $\bar{Q} = "1"$ 。

在D触发器结构中，该线称为维持“0”阻塞“1”线。



结论： $CP = "1"$ 时，如果时钟正边沿跳变时， $D = "0"$ ，输出的状态不受D变化的影响。

36

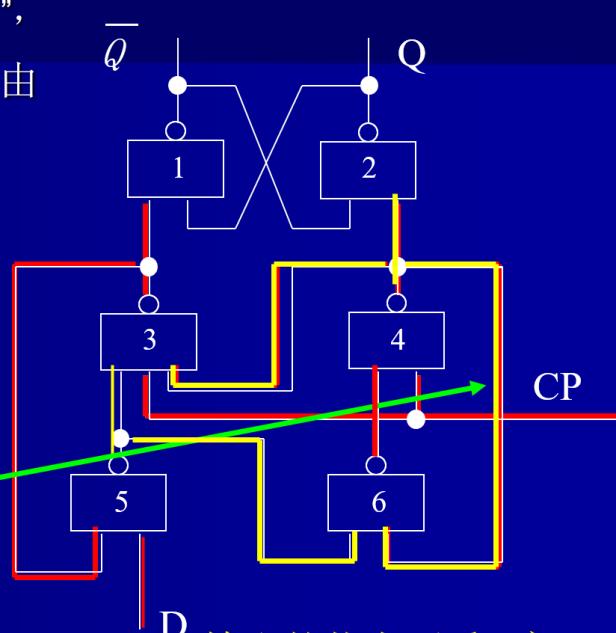
如果在时钟正边沿跳变时 $D = "1"$ ，则门3的输出为“1”。门4的输出由“1”变为“0”，门6的输入为“0”，门4的输出为“0”，不受D变化的影响。

由于门4的输出为“0”，强制门3的输出为“1”，门3的输出也不受D变化的影响。

顶端RS触发器的输出保持 $Q = "1"$, $\bar{Q} = "0"$ 。

在D触发器结构中，该线称为维持“1”阻塞“0”线。

结论：如果时钟正边沿跳变时， $D = "1"$ ，输出的状态不受D变化的影响。



其余情况在 $CP = 1$ 的时候D发生两种改变都不能改变上面的电路状态。分别由两根锁死阻塞线来限制

CP负边沿跳变的时候，34门都输出11为传统RS的保持状态

开关特性

- 数据建立时间 $tsu = \text{set up}$

数据接受的时候要稳定保证门3、4接受好数据(尤其门4要56处理完之后)，有 $tsu \geq tpd5 + tpd6 - 2 * \text{delay}$

- 数据保持时间 $th = \text{hold}$ 要等待阻塞线正常工作之后才能撤除这个有效的Data值

如 $D = 1$ ，必须等门4的阻塞线封锁住3和6才好则

$th \geq \max(tpd3, tpd4) - 1 * \text{delay}$

- 传输延迟参数 $tpd = \text{pass delay}$

分别讨论输出HL还是LH CP到Q还是Q非 共四种

- 主要看两条阻塞线，CP=1时有一条为0则代表一种状态，D只能被阻塞在5或者6门。当低电平两条线都是1时候56都打开才允许数据进入。

故低电平要保证D通过5和6就位在34门口，低电平 $\geq tsu$ 且 tsu 只能在低电平区间

高电平要保证thold形成好封锁 $\geq thold$ 在高电平期间

tpd cpQ 也需要在高电平时间，在高电平让输出稳定下来47 为的是给下一个器件就位比如下一个触发器的输入就位。所以如果两个触发器之间有组合逻辑电路，这里的延迟仍需要算在tpdcp->Q 上。

多个FF最高时钟频率？

$T_{cp-min} = tsu$

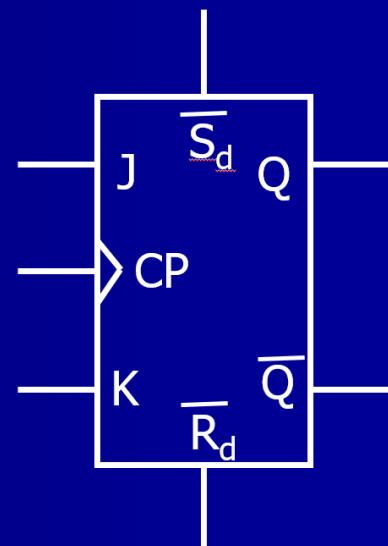
$T_{cp+min} = \max(th, tpdcp->Q + tpdG)$ (组合逻辑电路延迟))

Lect10 负边沿JK，主从触发器

● 负边沿触发的J-K触发器原理

$\overline{R_d}$	$\overline{S_d}$	CP	J	K	Q	\overline{Q}
0	1	X	X	X	0	1
1	0	X	X	X	1	0
1	1	↓	0	1	0	1
1	1	↓	1	0	1	0
1	1	↓	0	0	Q	\overline{Q}
1	1	↓	1	1	\overline{Q}	Q

异步清“0”
异步置“1”
置“0”
置“1”
保持
翻转



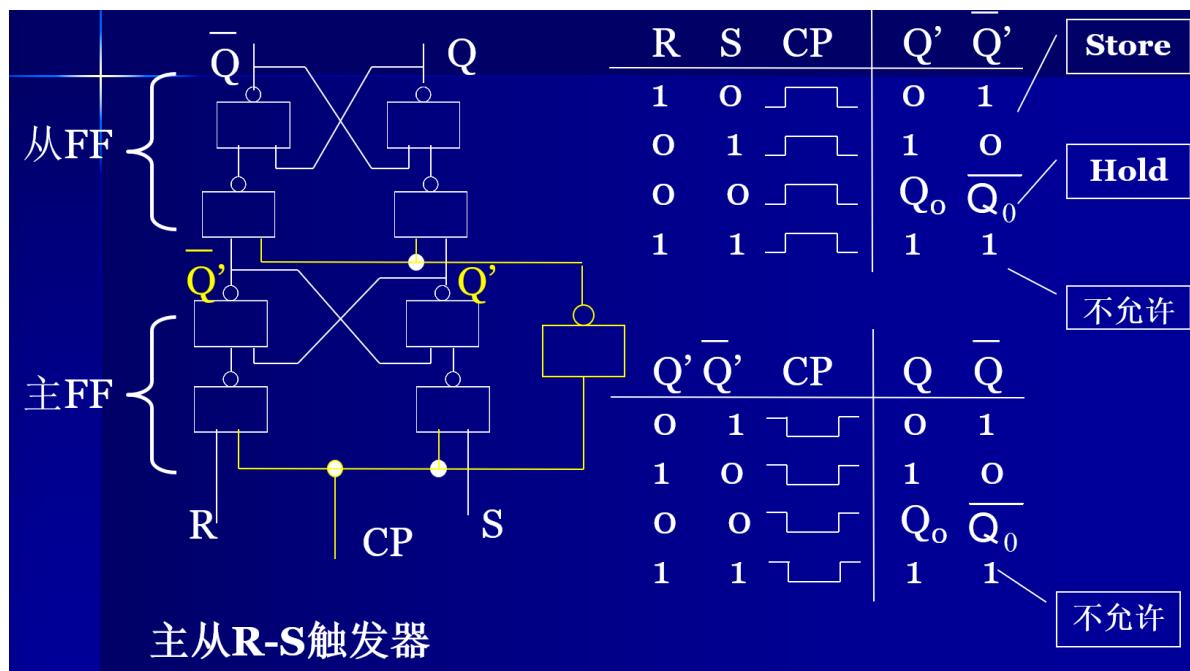
00保持11翻转 JK分别为Set Reset P4-P9 为操作的结构原理

异步是不需要等CP的强制清零，0是选中态，正常为11

主从触发器 两极电位串联 CP=1主触发器接受从封锁(+电位)

负跳变主封锁从触发器接受主触发器的状态 (负跳变)

主从RS触发器 17为加入异步 21有划归到主从RS的方法



主从R-S触发器的功能表

$\overline{R_D}$	$\overline{S_D}$	CP	R	S	Q	\bar{Q}
0	1	\times	\times	\times	0	1
1	0	\times	\times	\times	1	0
1	1	1	0	1	0	1
1	1	1	0	1	1	0
1	1	1	0	0	Q_0	\bar{Q}_0
1	1	1	1	1	1*	1*

负边沿J-K触发器真值表

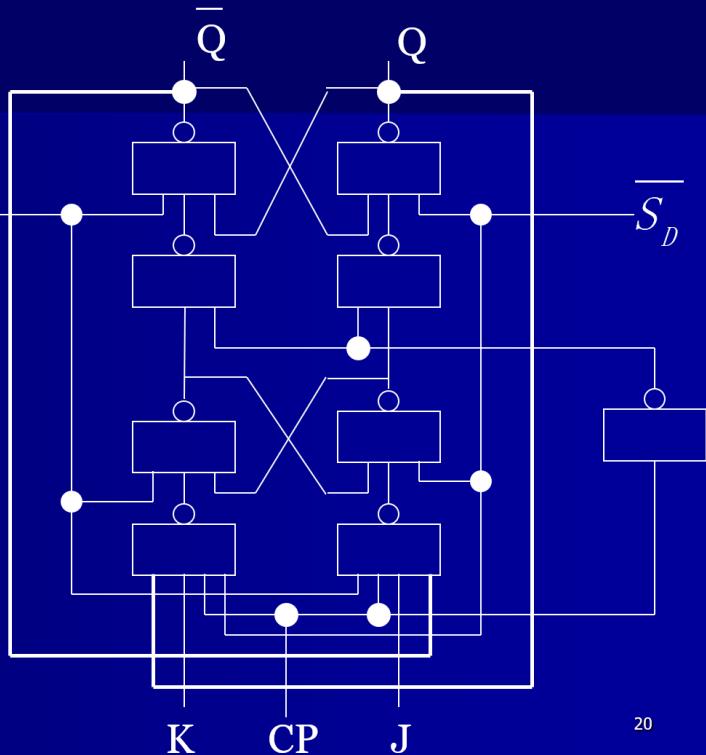
$\overline{R_D}$	$\overline{S_D}$	CP	J	K	Q	\bar{Q}
0	1	\times	\times	\times	0	1
1	0	\times	\times	\times	1	0
1	1	\downarrow	0	1	0	1
1	1	\downarrow	1	0	1	0
1	1	\downarrow	0	0	Q_0	\bar{Q}_0
1	1	\downarrow	1	1	\bar{Q}_0	Q_0

● 主从JK触发器

将Q和 \bar{Q} 非馈接到输入。

在正脉冲期间，主触发器接收是 $J\bar{Q}, KQ$ ；

负脉冲期间，主触发器状态打入从触发器。又称“脉冲触发器”。



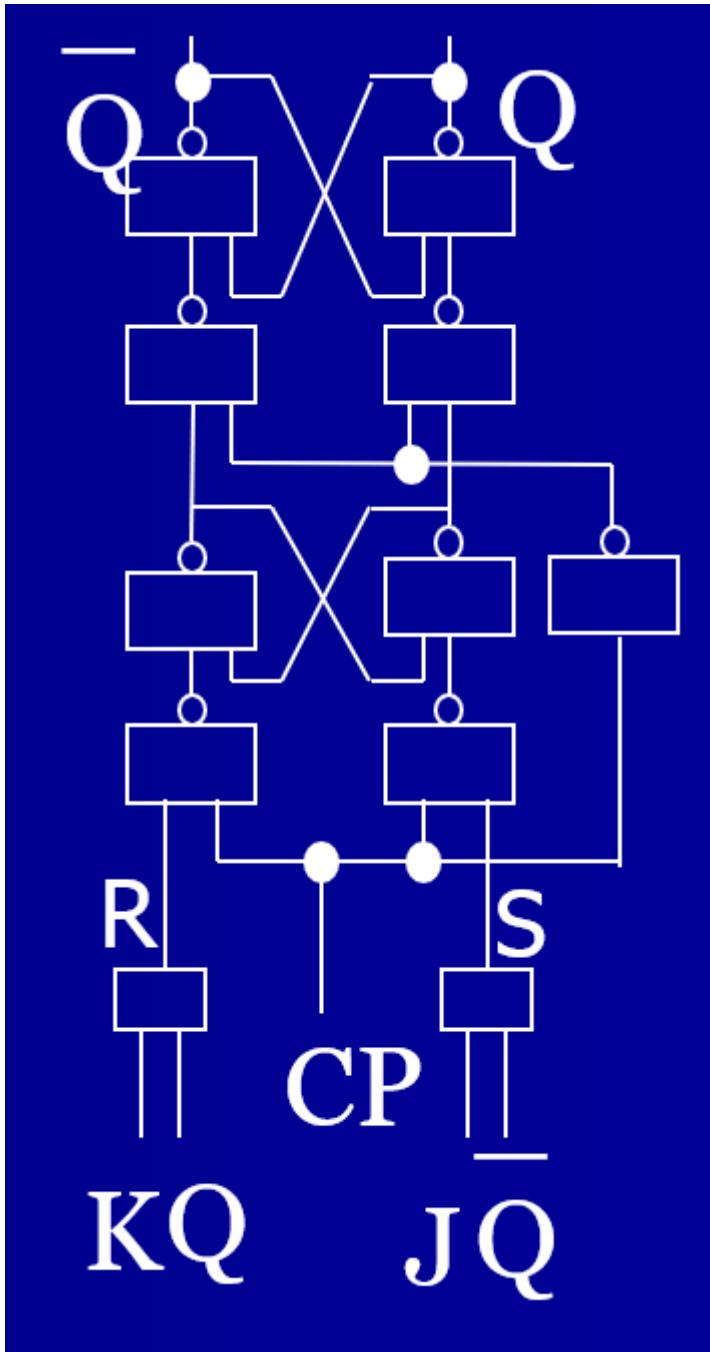
20

主从JK功能表与主从RS相同，只是11支持翻转

只是在正时钟周期中JK不变化的前提下给出，不然能不满足功能表，抗干扰差

主从VS边沿 主从似乎是负边沿因为正边沿接受负跳变才状态变化，但是触发方式不同。主从在负跳变时反应的是从触发器的状态可不一定是JK等输入的状态

经常从触发器在负边沿读主的时候，反应的是正边沿中的第一次变化（第二次有时候由于从触发器Q非没变导致JQ非=0 KQ=0而不发生变化，即最多可以发生和原来Q不同的一次变化比如把Q的1换成0，但是不能变第二次了会hold） 分析具体变化看下面这张图（注意从的之前状态）



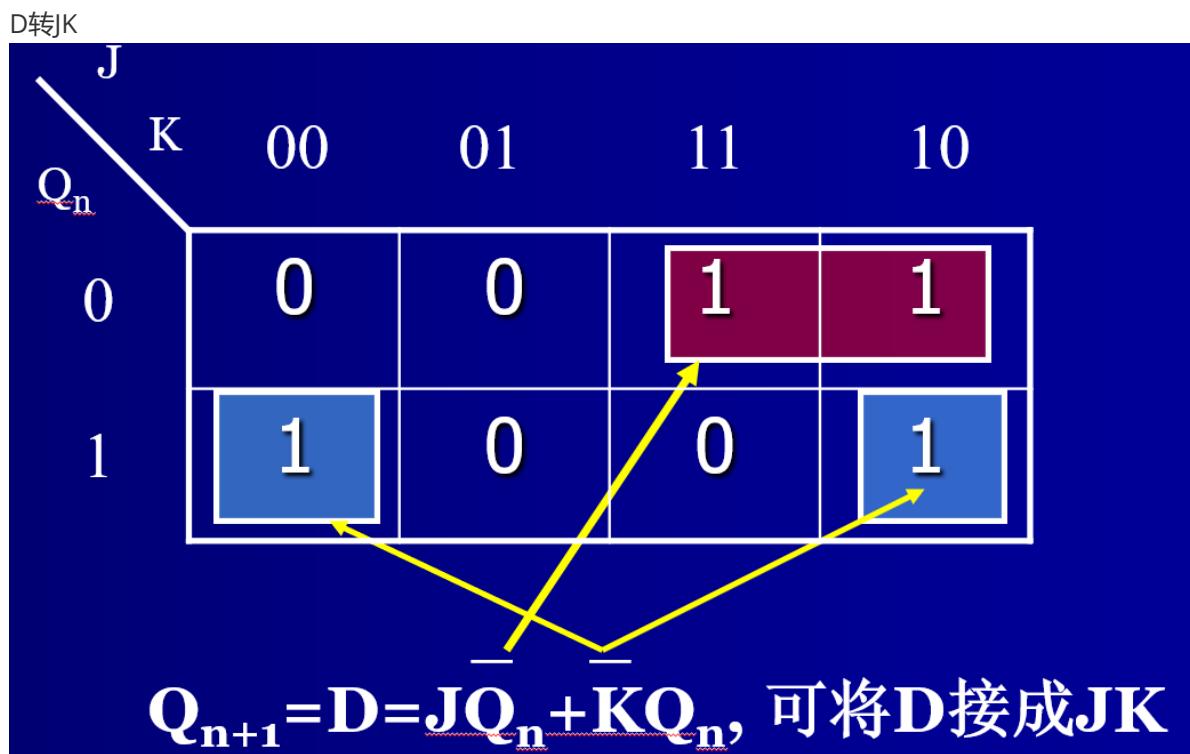
所以主从适合正脉冲窄，短高电平的情况

T触发器 Trigger cp来就翻转，一般都是边沿（由DorJK构成）

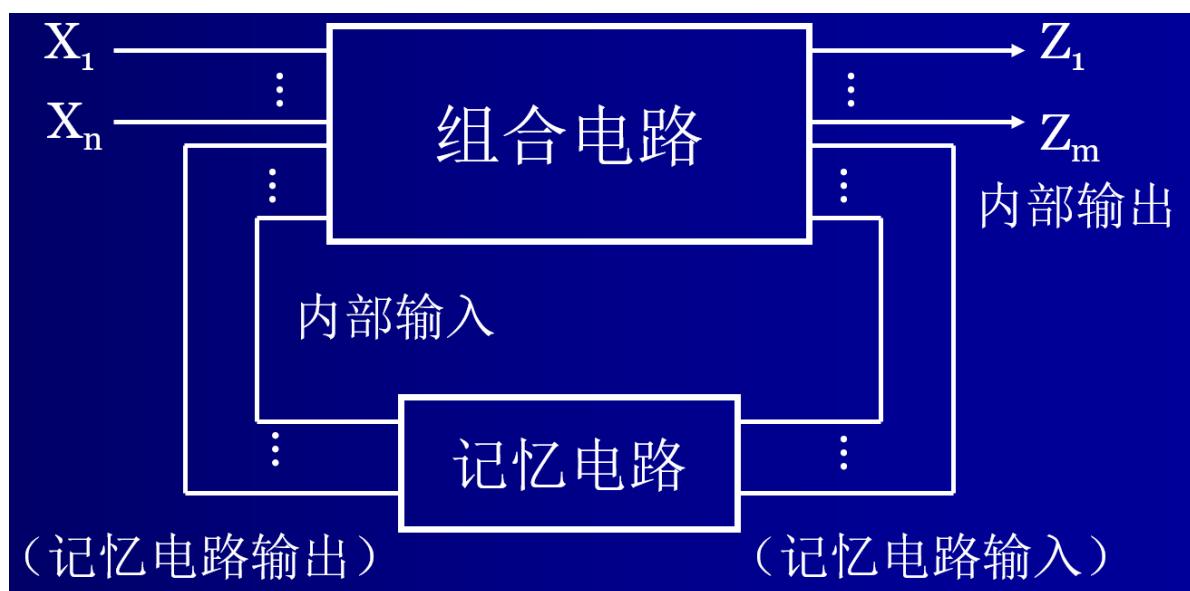
D的翻转需要Q非接D JK=11自带翻转

CP由功能控制T和真实时钟cp与成，T=1时Toggle模式

触发器的功能互换？ 功能需求=》用卡诺图写出Q_{n+1}与Q_n和输入之间的关系，再转换成激励表达式接入电路即可。



Lect11 同步时序电路的分析与设计



统一CP，约定脉冲改变整体的状态且只改变一次

功能表（含Q真值表） 输入（激励）+Q_n 得到 输出+Q_{n+1}

现在的状态+不同的激励能造成什么样的结果（未来状态）

状态表 卡诺图中一维是激励，一维是Q_n 表中内容是Q_{n+1}

现在状态+不同的激励造成什么样的未来状态

状态图 圈里面是状态（Q_n）转移边上有 输入/输出

状态方程 Q_{n+1} = 激励条件 (D)

下一个状态的决定方程

激励表 Q_n Q_{n+1} 得到 激励项（例如D、JK）

状态的不同转移需要什么样的激励条件

逻辑图——状态图——文字描述（设计为逆）

同步时序电路分析

- 根据电路图列出触发器激励函数表达式 $D =$ 和电路输出函数 $F = f(Q)$

输入怎么影响 D Q 怎么影响输出

- 建立 Q_{n+1} 表达式（根据上面的激励）以及 **状态表**（类似真值功能表， Q_n 和输入得到激励得到次态和输出的表格）

常用 $Q_{n+1} = D \quad // \quad Q_{n+1} = J \cdot \bar{Q}_n + \bar{K} \cdot Q_n$

- 状态（转移）图** 记得画完整

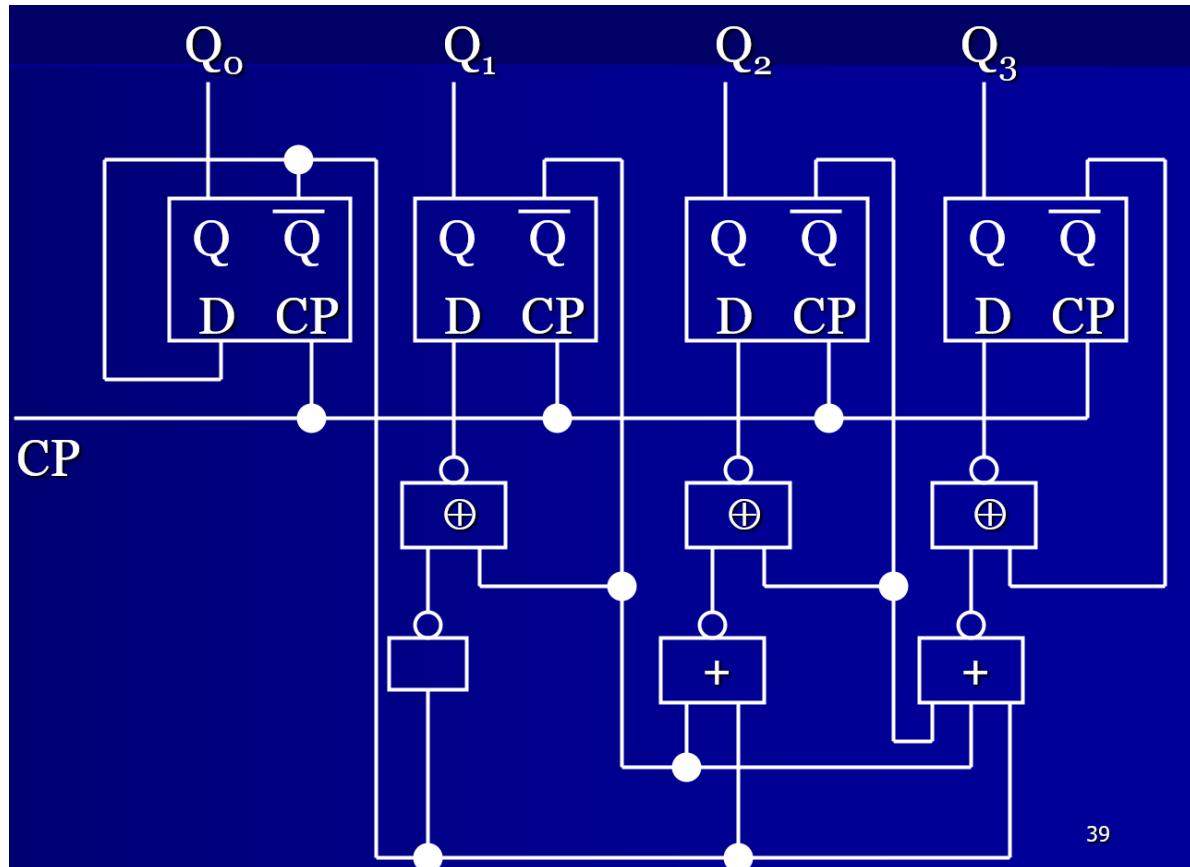
- 由状态图和次态、输出表达式带来的**时序图**（规定 cp 和输入，画 Q_n 和输出的波形）

- 分析输出和输入的关系说明逻辑功能

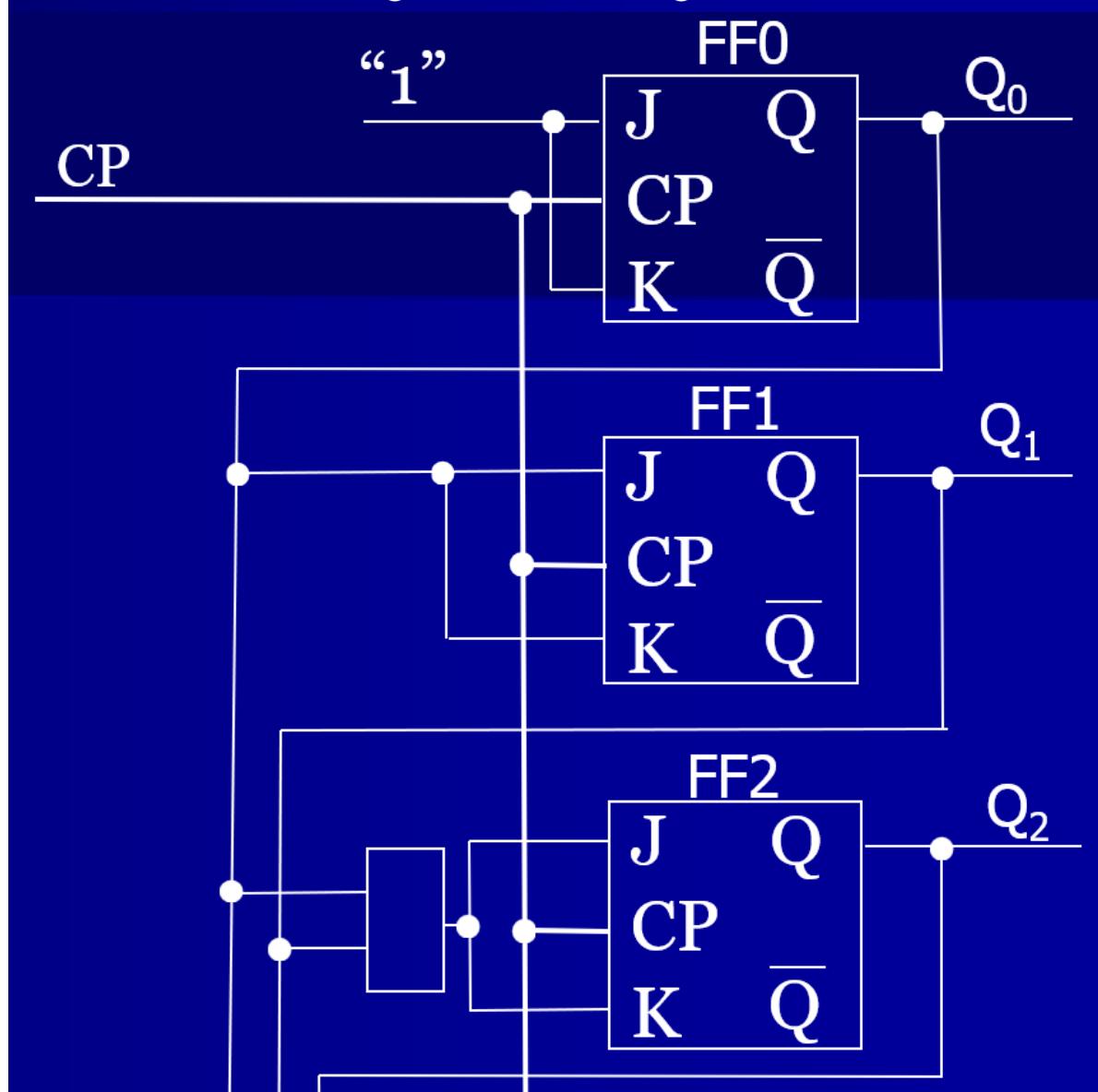
可以先明确里面状态、中间态的含义（如保存上一位的进位结果）

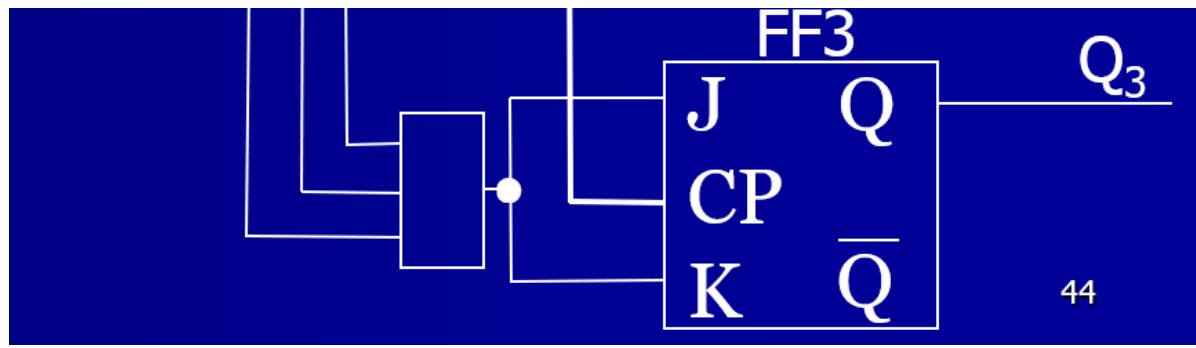
不完全的（6状态集合）也可以叫**3位格雷码计数器** 两个状态互换为 clk 的2分频 这样属于**不能自启动的格雷码计数器（不保证一定能进入主循环）**

这是能够自启动的4位二进制计数器 JK更方便实现“取反”



39





把几个D触发器串行接好，低Q进高D，最高！Q进最低D得计数器 **如何修复自启动？？？**

同步时序电路设计

1. 确定输入输出变量
2. 原始状态图和状态表（可能并不最简）状态表：

$Q_N \backslash X$	0	1
a	b, 0(Q_{N+1}, Z)	c, 0(Q_{N+1}, Z)
b	d, 0	e, 0
c	f, 0	g, 0
d	d, 0	e, 0
e	f, 0	g, 0
f	d, 0	e, 0
g	f, abc	g, 0

3. 状态化简，画出最简的**状态图**和**状态表**

对于完全相同的行，合并其状态，直至不能这么做，

状态表和状态图：一般每个转移边对应表中一行

4. 状态分配：给每个状态用一串**二进制来编码**
5. 求出**激励函数**和**输出函数** 化简前就检查好能否自启动

根据状态图画一个**大号的功能表**（真值）分别为

输入 $Q_n | Q_{n+1} |$ 激励 $D or JK$ (由 Q_{n+1}) | 输出 (由 Q_n 和输入)

根据后两列可以求出激励的表达式和Z的表达式

这时候可以用卡诺图来化简 横纵二维都是输入和各位Q_n, 里面是各个Q_{n+1}。能直接写出来的就直接写

JK触发器激励表			
Q _n	Q _{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

6. 画(电路)逻辑图

计数器 对CP脉冲计数，每次变化一次状态

同步、异步；加法、减法、可逆；二进制、十进制、任意进制；

同步四位二进制计数器 偶尔需要“实用” 这里的Q0Q1都是二、四分频

状态方程	激励函数的原理表达式	激励函数的实用表达式
$Q_{0(n+1)} = \overline{Q}_{0n}$	$D_0 = \overline{Q}_{0n}$	$D_0 = \overline{\overline{Q}_{0n}}$
$Q_{1(n+1)} = Q_{1n} \oplus Q_{0n}$	$D_1 = Q_{1n} \oplus \overline{Q}_{0n}$	$D_1 = \overline{\overline{Q}_{0n}} \oplus \overline{Q}_{1n}$
$Q_{2(n+1)} = Q_{2n} \oplus (Q_{1n} Q_{0n})$	$D_2 = Q_{2n} \oplus (Q_{1n} Q_{0n})$	$D_2 = \overline{Q}_{0n} + Q_{1n} \oplus \overline{Q}_{2n}$
$Q_{3(n+1)} = Q_{3n} \oplus Q_{2n} Q_{1n} Q_{0n}$	$D_3 = Q_{3n} \oplus Q_{2n} Q_{1n} Q_{0n}$	$D_3 = \overline{Q}_{0n} + \overline{Q}_{1n} + \overline{Q}_{2n} \oplus \overline{Q}_{3n}$

用JK触发器设计时，经常统一JK即可，而

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0 \ Q_1$$

$$J_3 = K_3 = Q_0 \ Q_1 \ Q_2$$

十进制计数器

编码、化简即可

可逆计数器 一个输入X表示是否正向计数 满1111之后进位Z为1

用X表示模式的时候，经常可以两边化简完之后用XF1+!XF2来调节

设计计数器的问题——自启动 修正状态图，改变激励表

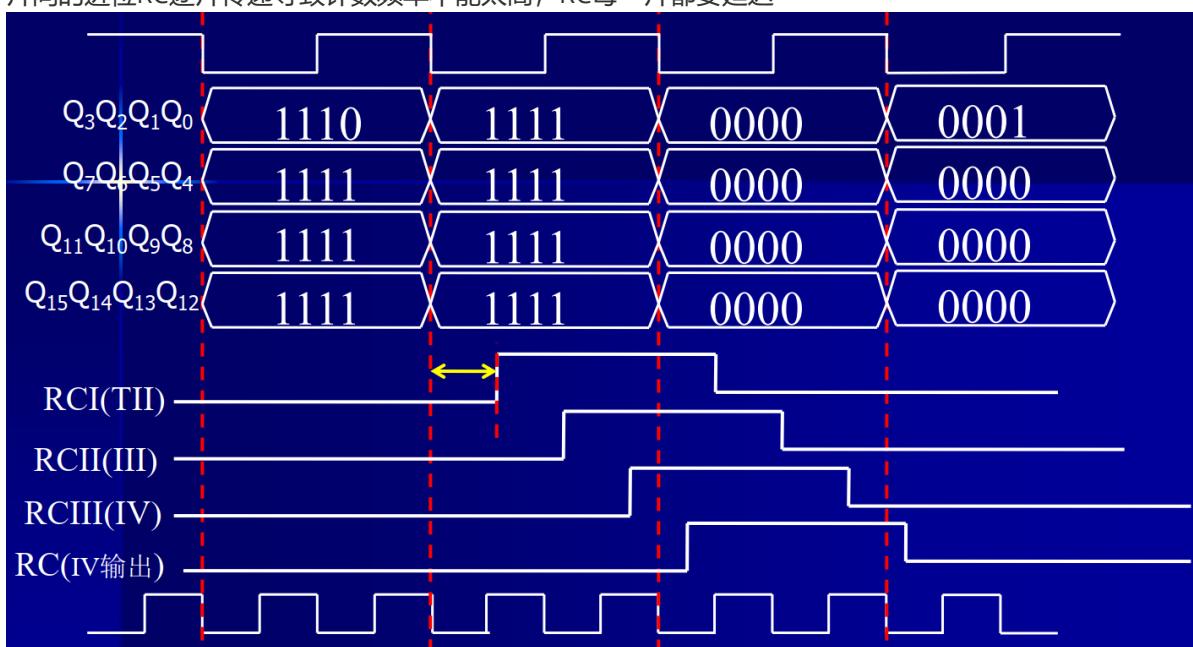
Lect13 集成化的4位二进制计数器

需要：异步、同步清零；同步置数；计数；保持；扩展；进位

仍然是0是确认态，常态为111 RC是进位在1111or1001时，这样下一时刻cp来了可以一起动。

T串行控制：低位RC接入高位的T，常态是0即FFHold状态

片间的进位RC逐片传递导致计数频率不能太高，RC每一片都要延迟



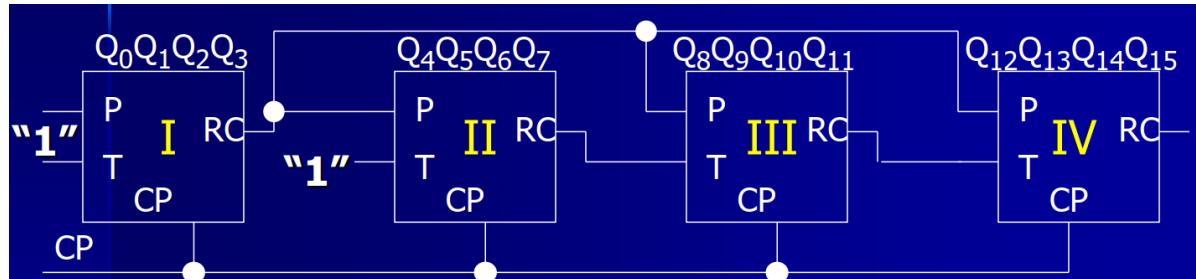
P并行控制：即快速扩展，分析进位条件是所有低位片都是1111

只有当TP都是1的时候本片才进位，最低位RC连各个P，然后T都是RC（次低片可以直接1）当最低片1111 (RC=1) 且上一位RC=1时本片进入进位预备状态。

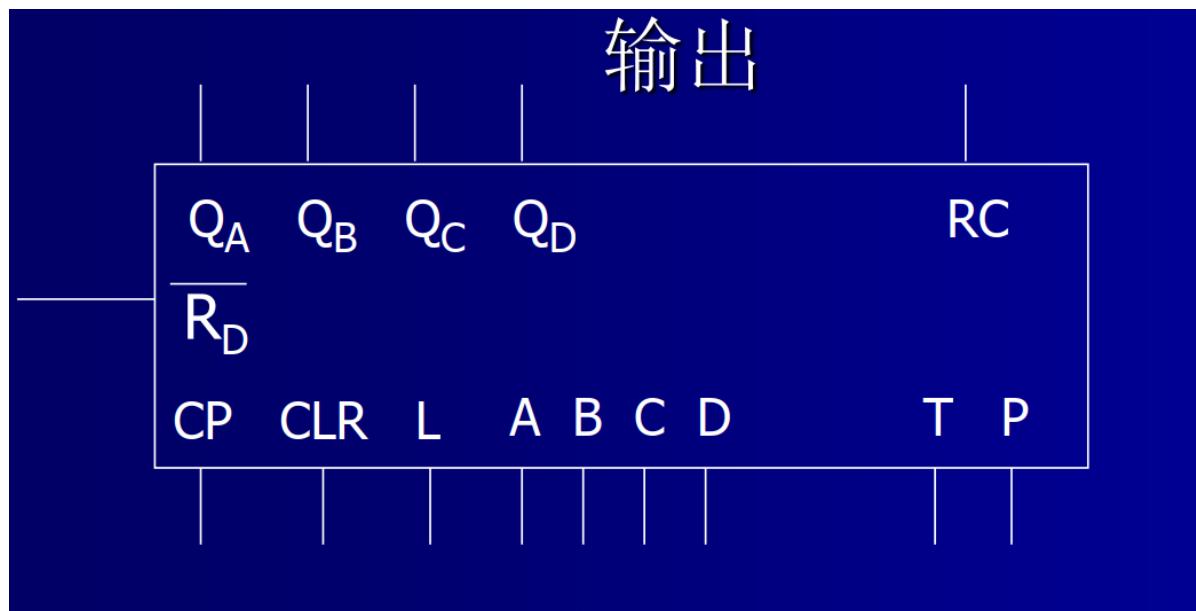
这样就是除了最低位，每片到1111之后就已经在RC—HighT预备好了

随时等待最低位的1111就位得到P=1时候进入预备，只需一个延迟

但是最高位的RC并不代表整体的进位。。



集成化的标准计数器接口与功能表：



➤ 用负边沿D触发器构成的4位二进制计数器

R_D	CLR	L	T	P	CP	功能
0	X	X	X	X	X	清零(异步)
1	0	X	X	X	↓	清零(同步)
1	1	0	X	X	↓	Load(并行数据输入)
1	1	1	1	1	↓	Count(计数)
1	1	1	0	X	X	FF Hold, RC=0
1	1	1	1	0	X	Hold(保持)

$Rd!$: 触发器提供or (谨慎) 组合逻辑直接清零 $Rd * Q$ 这种

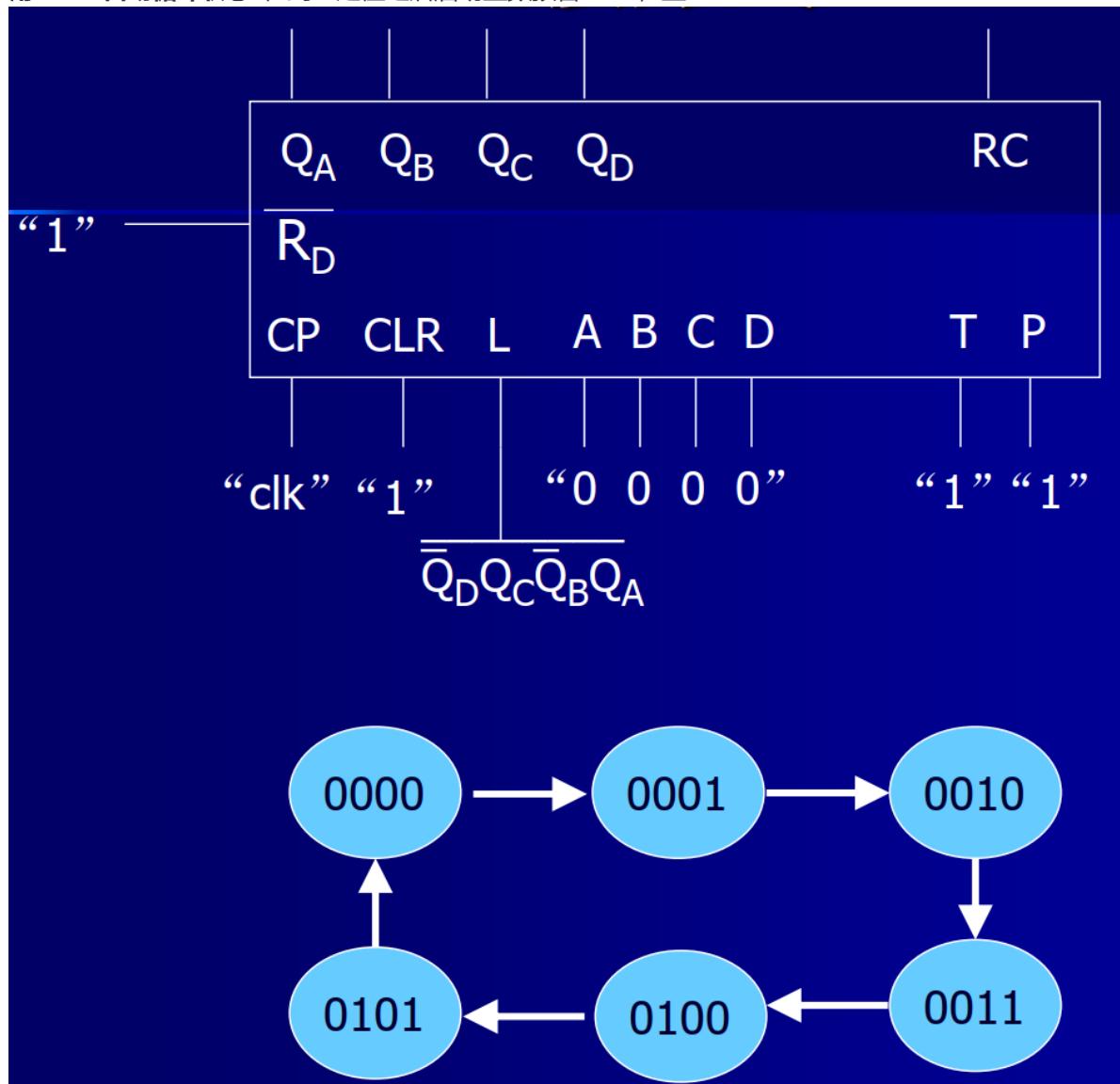
CLR: 从激励方程入手，直接在D上修改，下次检测就可以改了

Load: 接着改激励方程, $L!(\text{输入})+L(Q)$

TP:接着改激励方程记得改RC, 保持的时候D取自己的Q

$$\begin{aligned} D_A &= \text{CLR} \cdot (\overline{L} \cdot A + L \cdot (\overline{T} \cdot P) Q_A + T \cdot P Q_A) \\ D_B &= \text{CLR} \cdot (\overline{L} \cdot B + L \cdot (\overline{T} \cdot P) Q_B + T \cdot P (Q_B \oplus Q_A)) \\ D_C &= \text{CLR} \cdot (\overline{L} \cdot C + L \cdot (\overline{T} \cdot P) Q_C + T \cdot P (Q_C \oplus Q_B Q_A)) \\ D_D &= \text{CLR} \cdot (\overline{L} \cdot D + L \cdot (\overline{T} \cdot P) Q_D + T \cdot P (Q_D \oplus Q_C Q_B Q_A)) \end{aligned}$$

用76163自动循环状态 在到一定值之后启动置数预备Load, 置0000



产生固定序列信号如101001

1. 用上面自动循环状态的计数器
 2. 常规方法, 用D或JK
 3. 循环移位设计, 用QQQ表示长3子串然后循环起来 (记得循环到下一个的前三位), 注意不能一对多 101001101001
- 输出随便接其中一个Q即可

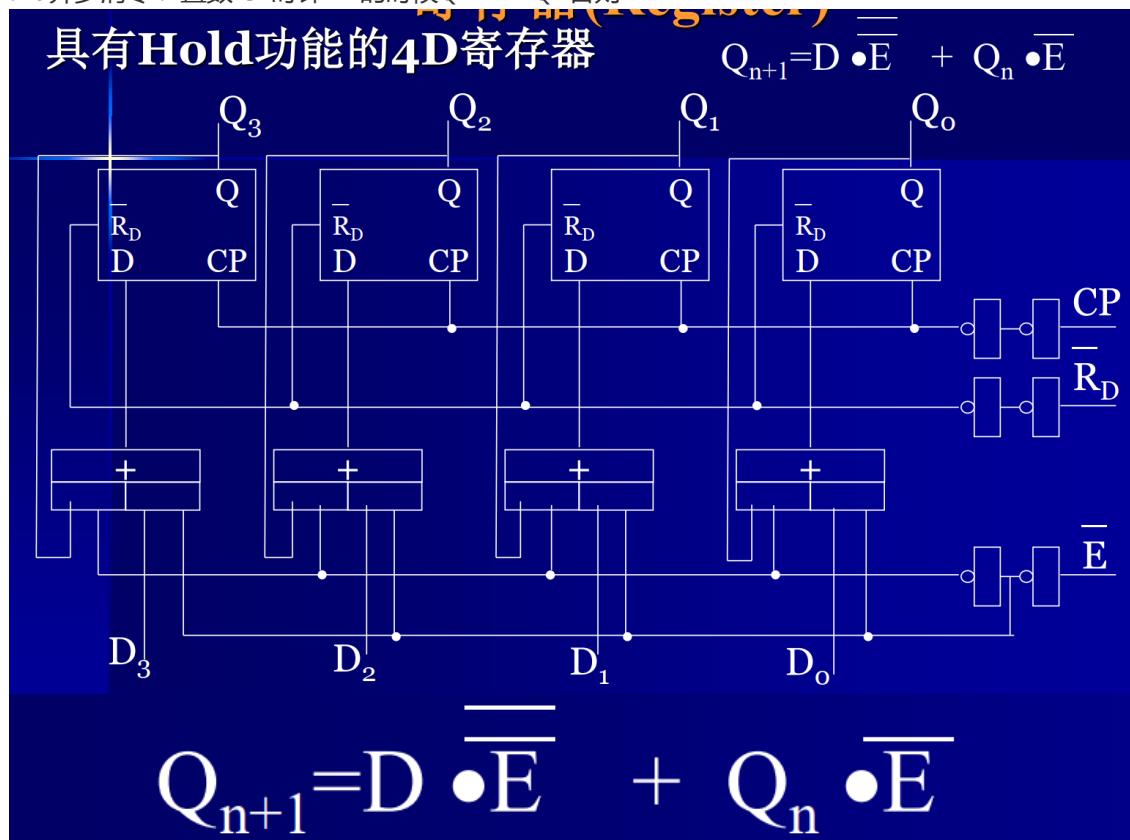
Lect14-1 寄存器——存储

基本寄存器

- 基本4D寄存器，需要异步清零 ($\overline{R_D}=0$) 和置数 ($\overline{R_D}=1$)

记得减少对外负载，但是数据不在的时候不能保留了呜呜呜

- $\overline{R_D}$ 异步清零 \overline{E} 置数 CP时钟 11的时候 $Q_{n+1} = Q_n$ 否则 D_n

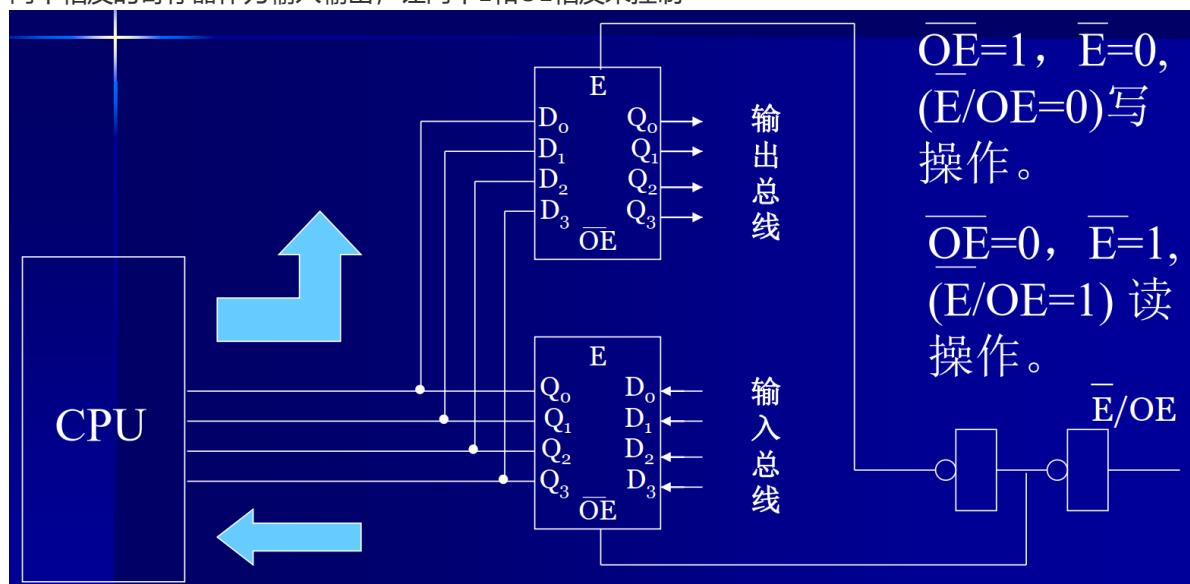


- 置数、保持还能选择？加Select输入可以选择数据源

$$Q_{n+1} = D_A \cdot \overline{S} \cdot \overline{E} + D_B \cdot S \cdot \overline{E} + Q_n \cdot \overline{E}$$

- 再加个使能 $\overline{OE} = 1$ 的时候为高阻态P13

两个相反的寄存器作为输入输出，让两个E和OE相反来控制



移位寄存器

需要寄存器有移位功能，每一个cp脉冲，寄存器数据顺序向左（右）移动一位，要采用边沿触发（或主从触发器）

串入并出的移位，每一个Q接高位D，但是不循环移位（叫右移其实是左移。。。）

并入并出的双向移位Reg 双向用S0S1控制 1就是哪个方向，11置数

R_D	S_0	S_1	CP	功能(Q_{n+1})
0	X	X	X	置“0”
1	0	0	↑	Hold(Q_n)
1	1	0	↑	Shift Right(Q_{Ln})
1	0	1	↑	Shift Left(Q_{Rn})
1	1	1	↑	Load(Store)(D)

$$Q_{n+1} = \overline{S_0} \overline{S_1} Q_n + S_0 \overline{S_1} Q_{Ln} + \overline{S_0} S_1 Q_{Rn} + S_0 S_1 D$$

应用：用移位寄存器设计信号发生器101000 选用串进并出的

取n=3得到 101 010 100 000 001 010有重复010无法区分，增加位数

n=4得到 1010 0100 1000 0001 0010 0101无重复，每次通过Q0123讨论出一个下一位要移进去的数字，做出6/16的卡诺图化简D，下一个Q0讨论即可。

发现并不能自启动。有3个非工作循环。对非工作循环中任取一个状态，让次态进入工作循环比如0000=>0001进入（移入1）卡诺图7/16

记得考虑怎么样比较好通过移位方式进入工作循环

Lect14-2 Ch5可编程逻辑电路

ROM只读存储器 40

PROM与阵列固定（地址译码固定）或可编程（储存信息可以改）

就是全部的输入情况来分情况讨论 计算点数的时候记得与阵列算非

PLA可编程逻辑阵列 47 与或均可编程 组合逻辑表示D

压缩，只表示最小项甚至只表示乘积项即可，不是最小项

先找 F_i 的最简与或表达式然后找乘积项，一条竖线表示一个乘积项上面与上乘积项的表达式。下面每个输出（一行）自取乘积的sum

计算点数（交叉）： $2^{\text{输入}} * P + \text{输出} * P$

计算容量：输入数目 * P项数目 * 输出数 4^*

PAL可编程阵列逻辑 与可编程或固定 72

决定好了每个输出取哪几个P项然后自己编写P项

有反馈的寄存器PAL可以实现时序逻辑但不能组合

演变过程P78

GAL通用可编程阵列逻辑 80 与可编程或固定

输出控制OLMC+反馈结构+多次可编程+集成度++

需要：输入、输出、与阵列、输出逻辑宏单元

GAL16V8@20pin 16入（右侧8个在OLMC下通用出入）

8出 8个OLMC（或阵列in it） P86逻辑图

总的电路经验

xor: 想逐位加法、01个数等等，化简的时候多用xor简化，逐位判同（尤其是计数、加法）

每两位xor: 典型格雷码

分析组合电路：不一定顺着看输入，试试顺着看输出？

模式控制：多在激励方程上之际修改，如置数、清零、数据选择。优先的放外面

组合逻辑电路分析与设计：表达式、功能表、描述

对BCD分级译码：先译AD，因为1xx0和1xx1唯一为8和9

分级选数据和分级译码都一样，一定看好怎么分、分谁

比如S3210选D0~9，可以先用S3区别0123和89XX互换

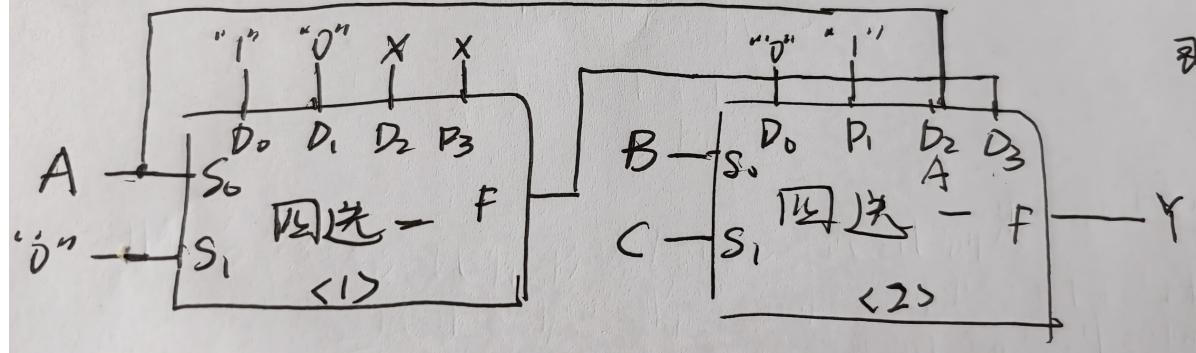
其实BCD译码和数据选择都是对控制端的讨论和翻译

组合电路的开关参数：XX输入到XX输出的延时

2个四选一模拟3输入函数?

A	B	C	F
0	0	0	0
1	0	0	0
0	1	0	1
1	1	0	1
0	0	1	0
1	0	1	1
0	1	1	1
1	1	1	0

③ 四选一



16并4并1：算大PG需2，算C4 8 12 16需3 各片算C需要2 算F1共6