上海大学
Shanghai University

数字逻辑电路
分析与设计

一、 数制及转换

二、 二值编码

三、 常用二-十进制码

四、 逻辑关系

一、数制及转换：

$$(N)_r = (K_{n-1}K_{n-2}\cdots K_1K_0.K_{-1}K_{-2}\cdots K_{-m})_r$$

$$= K_{n-1}\times r^{n-1} + \cdots + K_1\times r^1 + K_0\times r^0$$

$$+ K_{-1}\times r^{-1} + \cdots + K_{-m}\times r^{-m}$$

$$= \sum_{i=-m}^{n-1} K_i \times r^i$$

1、十进制与r进制间转换：

(1) r进制转换成十进制数

方法：按权展开，求算术和.

(2) 十进制数转换成r进制数

方法：整数部分与小数部分分别加以转换.

a. 整数部分采用 “除r取余” 法。

b. 小数部分采用 “乘r取整” 法。

十进制转换为二进制：

a. 整数部分采用 “除2取余” 法

b. 小数部分采用 “乘2取整” 法

2、非十进制间转换：

(1) 以十进制为桥梁

(2) 二、八、十六进制间转换

a. 3位二进制数相当于1位八进制数

b. 4位二进制数相当于1位十六进制数

二、二值编码

- 原码：正数的原码与真值相同；负数的原码等于真值绝对值最高位上加1
- 反码：正数的反码就是其原码；负数的反码是原码中的符号位不变，数值部分按位取反
- 补码：正数的补码就是其原码；负数的补码是原码中的符号位不变，数值部分按位取反再加1

- 说明：补码等于反码＋1；减去一个正数可以当作加一个对应的补码

三、常用二-十进制码

BCD码（用4位二进制码表示一位十进制）分为如下三种：

- 有权码（各码位有固定的权值）

  8421，2421，5211，631-1

- 偏权码（在有权码的基础上加一个偏值）

  余3码

- 无权码（各码位没有固定的权值）

  格雷码(循环性，反射性)

- BCD码与二进制码间转换

表 1-5 常用的 BCD 码

十进制数	8421	2421	631-1	余 3 码	格雷码
0	0000	0000	0011	0011	0010
1	0001	0001	0010	0100	0110
2	0010	0010	0101	0101	0111
3	0011	0011	0111	0110	0101
4	0100	0100	0110	0111	0100
5	0101	1011	1001	1000	1100
6	0110	1100	1000	1001	1101
7	0111	1101	1010	1010	1111
8	1000	1110	1101	1011	1110
9	1001	1111	1100	1100	1010

四、逻辑关系

- “与” 的运算关系

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 0 \quad 1 \cdot 0 = 0 \quad 1 \cdot 1 = 1$$

$$0 \cdot A = 0 \quad 1 \cdot A = A \quad A \cdot A = A$$

- “或” 的运算关系

$$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 1$$

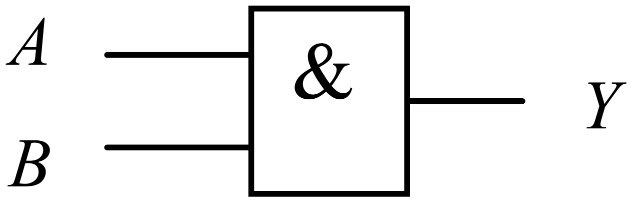
$$0 + A = A \quad 1 + A = 1 \quad A + A = A$$

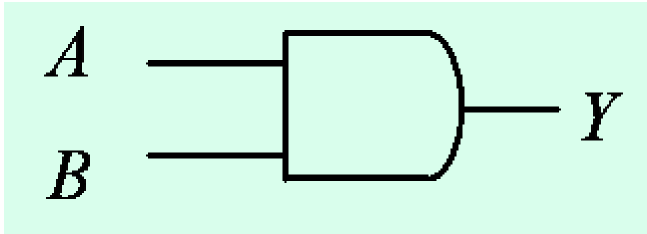
- “非” 的运算关系

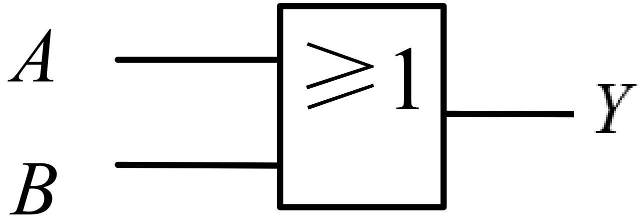
$$\overline{0} = 1 \qquad \overline{1} = 0$$

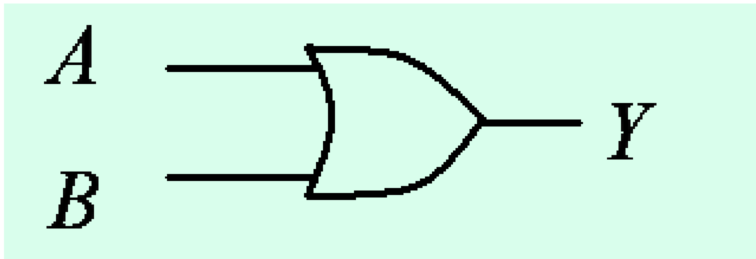
$$\overline{\overline{A}} = A \quad A + \overline{A} = 1 \quad A \cdot \overline{A} = 0$$

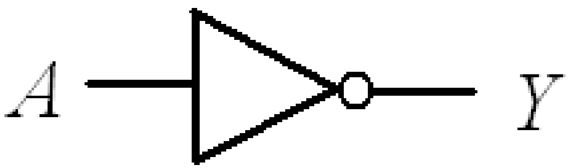
基本逻辑关系与逻辑符号

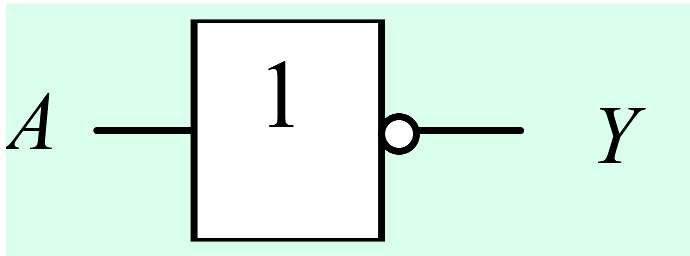




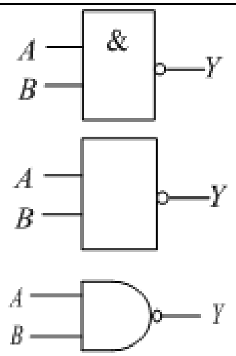
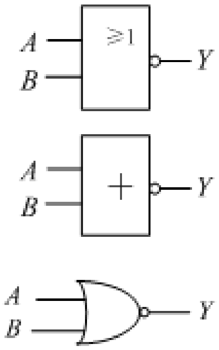
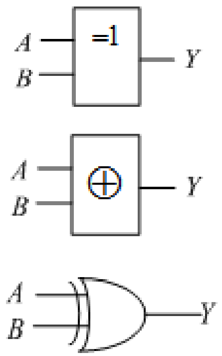
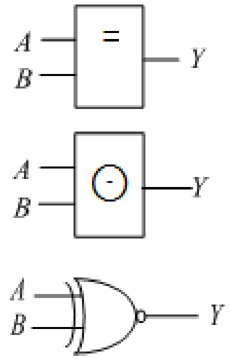
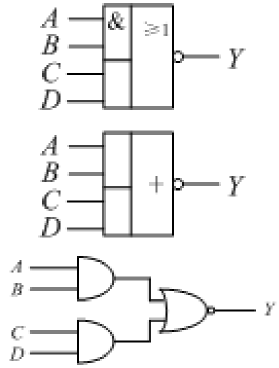








复合逻辑关系与逻辑符号

<table>
<tr><td colspan="2">逻辑关系</td><td>与非</td><td>或非</td><td>异或</td><td>同或</td><td>与或非</td></tr>
<tr><td colspan="2">表达式</td><td>$Y=\overline{A\cdot B}$</td><td>$Y=\overline{A+B}$</td><td>$Y=\bar{A}B+A\bar{B}$
$=A\oplus B$</td><td>$Y=\bar{A}\bar{B}+AB$
$=A\odot B$</td><td>$Y=\overline{AB+CD}$</td></tr>
<tr><td colspan="2">逻辑符号</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">功能特征</td><td>输入全 1 时，输出为 0</td><td>输入全 0 时，输出为 1</td><td>输入相异时，输出为 1</td><td>输入相同时，输出为 1</td><td>与项全为 0 时，输出为 1</td></tr>
<tr><td rowspan="5">真值表</td><td>A B</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td rowspan="5"></td></tr>
<tr><td>0 0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0 1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1 0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1 1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>

一、逻辑代数

二、最小项与最大项

三、卡诺图化简

一、逻辑代数

表 2-1 逻辑代数的常用公式

	或	与
交换律	$a+b=b+a$	$ab = ba$
结合律	$a+(b+c)=(a+b)+c$	$a\cdot(b\cdot c)=(a\cdot b)\cdot c$
分配律	$a+b\cdot c=(a+b)(a+c)$	$a(b+c)=ab+ac$
0-1 律	$0+a=a$	$1\cdot a=a$
	$1+a=1$	$0\cdot a=0$
互补律	$a+\overline{a}=1$	$a\cdot\overline{a}=0$
吸收律	$a+ab=a$	$a(a+b)=a$
	$a+\overline{a}b=a+b$	$a(\overline{a}+b)=ab$
重叠律	$a+a=a$	$a\cdot a=a$
反演律	$\overline{a+b}=\overline{a}\cdot\overline{b}$	$\overline{a\cdot b}=\overline{a}+\overline{b}$
对合律	$\overline{\overline{a}}=a$	
包含律	$ab+\overline{a}c+bc=ab+\overline{a}c$	$(a+b)(\overline{a}+c)(b+c)=(a+b)(\overline{a}+c)$

一、逻辑代数

基本规则

- 代入规则：在一个逻辑等式中，若将等式两边出现的某变量A都用同一个逻辑式替代，则替代后等式仍然成立，这个规则称为“代入定理”
- 反演规则：对任何一个逻辑表达式F，如果将式中所有的“·”换成“+”，“+”换成“·”，“0”换成“1”，“1”换成“0”，原变量换成反变量，反变量换成原变量，则可以得到原式F的反函数$\overline{F}$
- 对偶规则：对任何一个逻辑表达式F，如果将式中所有的“.”换成“+”，“+”换成“.”，“0”换成“1”，“1”换成“0”，这样得到一个新的逻辑表达式F_D。F和F_D互为对偶式

逻辑函数的化简

$$F = ab + \bar{b}c + a\bar{b}c$$

③最简与非-与非式

在最简与-或式的基础上，利用摩根定理，得

$$F = \overline{\overline{ab + \bar{b}c}} = \overline{\overline{ab} \cdot \overline{\bar{b}c}}$$

④最简或非-或非式

在最简或-与式的基础上，利用摩根定理，得

$$F = \overline{\overline{(a+\bar{b})(b+c)}} = \overline{\overline{a+\bar{b}} + \overline{b+c}}$$

⑤最简与-或-非式

先利用反演规则得到最简反函数 $\overline{F}$ 的与-或式，再求得 $\overline{F}$ 的反函数 F。

$$\because \overline{F} = (\bar{a}+\bar{b})(b+\bar{c}) = \bar{a}b + \bar{a}\bar{c} + \bar{b}b + \bar{b}\bar{c} = \bar{a}b + \bar{b}\bar{c} + \bar{a}\bar{c} = \bar{a}b + \bar{b}\bar{c}$$

$$\therefore F = \overline{\overline{F}} = \overline{\bar{a}b + \bar{b}\bar{c}}$$

二、最小项与最大项

➢ 最小项：

- 在n个变量的逻辑函数中，包含全部n个变量的“与”项称为最小项
- 在每一个最小项中，每个变量都以原变量或反变量的形式出现一次，且仅出现一次

➢ 最小项表达式：

由给定函数的最小项之和所组成的逻辑表达式称为最小项表达式，又叫标准“与-或”式。为书写和使用方便，可以用“$\sum$”表示累计，用圆括号内的十进制数表示参与“或”运算的各最小项的项号。

二、最小项与最大项

- 最大项：
  - 在n个变量的逻辑函数中，包含全部n个变量的“或”项称为最大项
  - 每一个最大项中，每个变量都以原变量或反变量的形式出现一次，且仅出现一次
- 最大项表达式：

由给定函数的最大项之积所组成的逻辑表达式称为最大项表达式，又叫标准“或-与”式。为书写和使用方便，可以用“$\prod$”表示累计，用圆括号内的十进制数表示参与“与”运算的各最大项的项号。

表 2-2 三变量的最小项及编号

最小项	变量取值			编号
	A	B	C	
$\overline{A}\,\overline{B}\,\overline{C}$	0	0	0	m_0
$\overline{A}\,\overline{B}C$	0	0	1	m_1
$\overline{A}B\overline{C}$	0	1	0	m_2
$\overline{A}BC$	0	1	1	m_3
$A\overline{B}\,\overline{C}$	1	0	0	m_4
$A\overline{B}C$	1	0	1	m_5
$AB\overline{C}$	1	1	0	m_6
ABC	1	1	1	m_7

表 2-4 三变量最大项及编号

最大项	变量取值			编号
	A	B	C	
$A+B+C$	0	0	0	M_0
$A+B+\overline{C}$	0	0	1	M_1
$A+\overline{B}+C$	0	1	0	M_2
$A+\overline{B}+\overline{C}$	0	1	1	M_3
$\overline{A}+B+C$	1	0	0	M_4
$\overline{A}+B+\overline{C}$	1	0	1	M_5
$\overline{A}+\overline{B}+C$	1	1	0	M_6
$\overline{A}+\overline{B}+\overline{C}$	1	1	1	M_7

二、最小项与最大项

➢ 最小项和最大项的关系：

（1）最小项和最大项之间具有对偶性。

（2）下标i相同的最小项与最大项互补，即$m_i = \overline{M}_i$。

（3）对于同一逻辑函数F，有$F = \sum_i m_i = \prod_{j \neq i} M_j$。

三、卡诺图化简

- 卡诺图
- 卡诺图表示逻辑函数

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	1	0	0
11	1	0	1	1
10	0	0	0	0

F

$$F=\sum m(1,3,4,5,12,14,15)=\prod M(0,2,6,7,8,9,10,11,13)$$

三、卡诺图化简

➢ 卡诺图化简逻辑函数

（1）两个相邻最小项合并成一项，可消去一个变量。

（2）四个相邻最小项合并成一项，可消去两变量。

（3）八个相邻最小项合并成一项，可消去三个个变量。

（4）十六个相邻最小项合并成一项，可消去四个变量。

注意：

① 卡诺圈内的方格数必须是 2^n 个（n=0、1、2……）。

② 相邻方格包括上下底相邻、左右边相邻和四角相邻。

③ 同一方格可以被不同的卡诺圈重复包围，但新增卡诺圈中一定要有新的方格。

④ 卡诺圈内的方格数要尽可能多，卡诺圈的数目要尽可能少。

圈最小项：圈卡诺图相邻位置上的“1”，得到最简与-或式、最简与非-与非式。

圈最大项：圈卡诺图相邻位置上的“0”，得到最简或-与式、最简或非-或非式。

三、卡诺图化简

➢ 含无关项的逻辑函数卡诺图化简

无关项：**指在一逻辑函数中，输入变量的某些取值组合不会出现，或一旦出现逻辑输出值可以是任意的。称为该取值组合对应的最小（大）项为无关项、任意项或约束项**

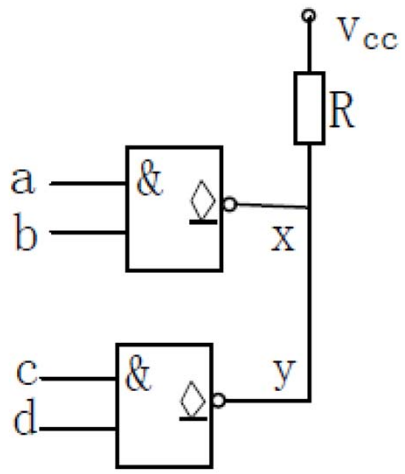
最小项表达式：$F=\sum m(\ )+\sum d(\ )$或者$\begin{cases}F=\sum m(\ )\\ \sum d(\ )=0\end{cases}$

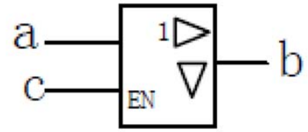
最大项表达式：$F=\prod \mathrm{M}(\ )\cdot\prod \mathrm{D}(\ )$或者$\begin{cases}F=\prod \mathrm{M}(\ )\\ \prod \mathrm{D}(\ )=1\end{cases}$

充分利用无关项所对应的逻辑函数值既可看做1也可看做0的特性，尽量扩大卡诺圈，减少卡诺圈个数，这样才能尽可能多地消除乘积项或变量，使得逻辑函数更简单。

第三章
组合逻辑
电路设计

- OC门、三态门
- 常用组合逻辑模块
  - 编码器
  - 并行加法器
  - 比较器
  - 译码器
  - 数据选择器
- 应用实例





➢ 编码器

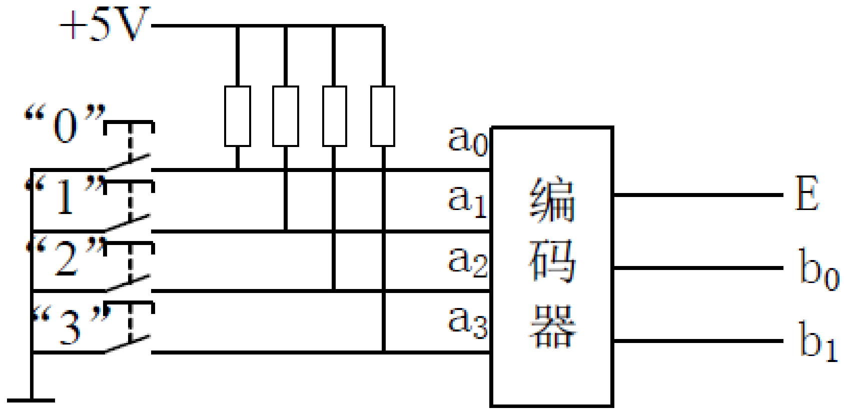


图 2-15 4 线-2 线编码器的原理图

$$E = a_3 a_2 a_1 a_0$$

$$b_1 = \overline{a_3} + \overline{a_2}$$

$$b_0 = \overline{a_3} + a_2 \overline{a_1}$$

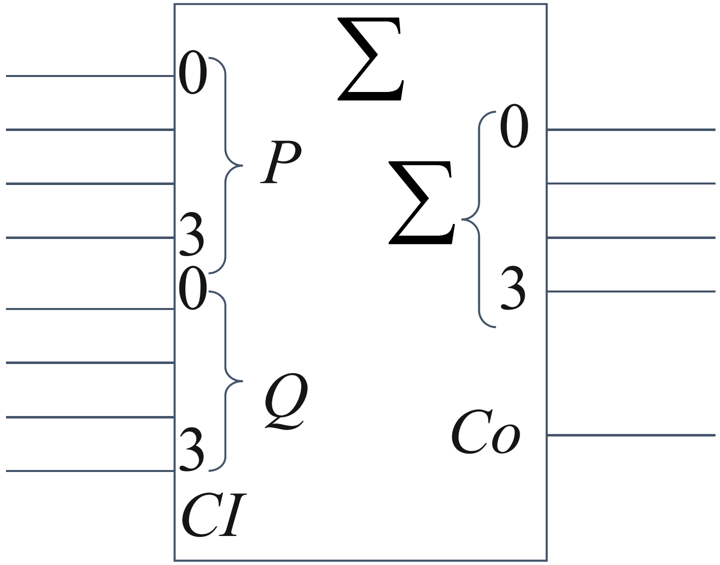
➢ 优先编码器

4-2线优先编码器功能表

a_3	a_2	a_1	a_0	E	b_1	b_0
1	1	1	1	1	x	x
0	x	x	x	0	1	1
1	0	x	x	0	1	0
1	1	0	x	0	0	1
1	1	1	0	0	0	0

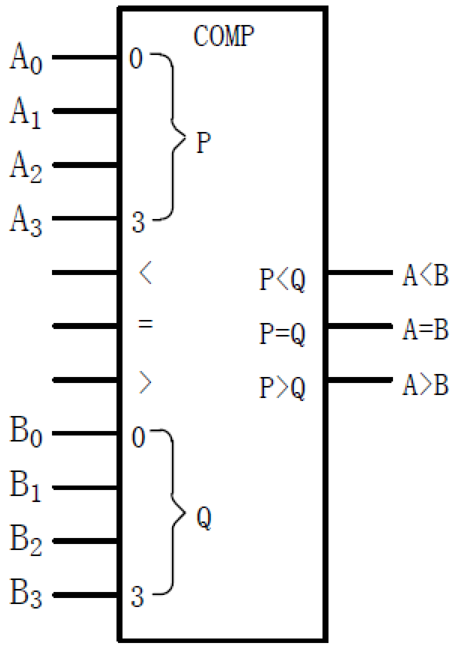
➤ 四位并行加法器

例3-1



➤ 四位并行比较器

例3-2，级联方法

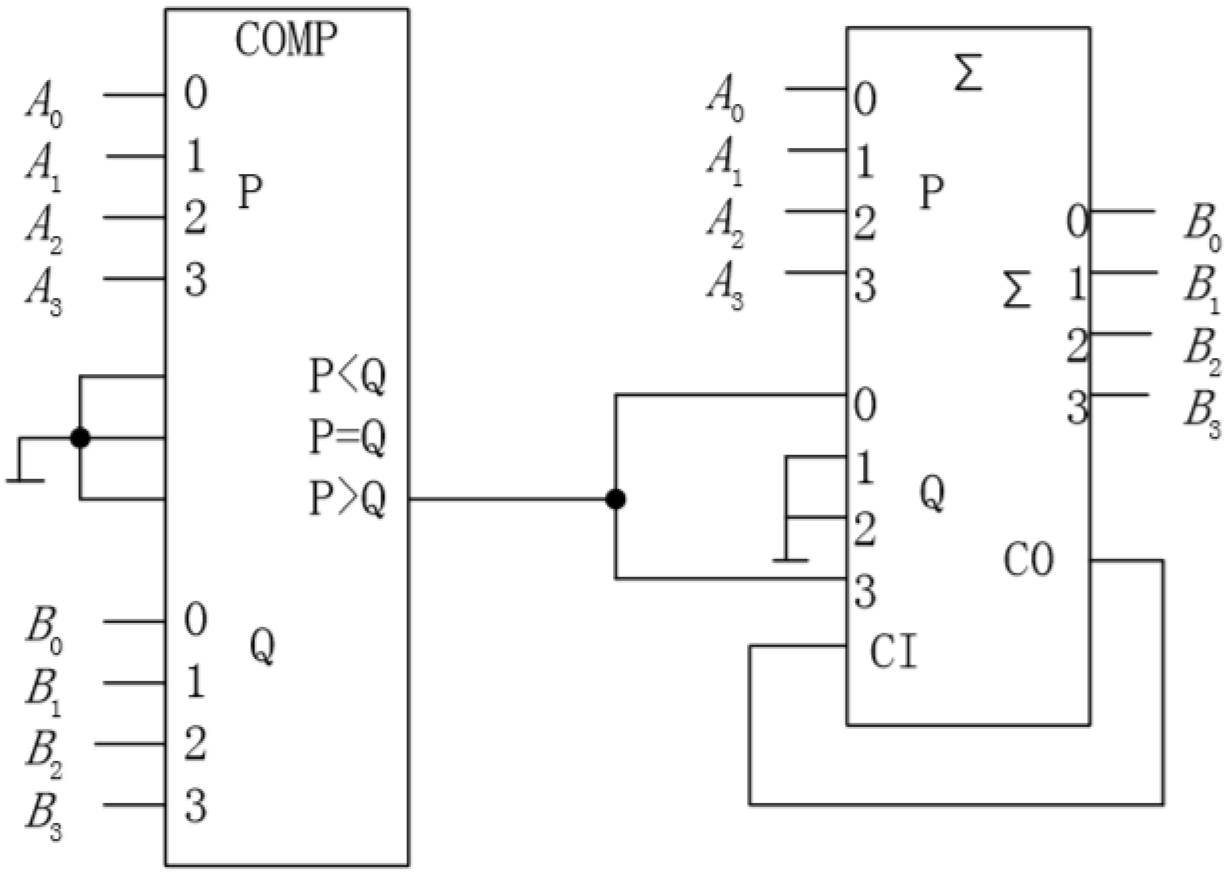


4 位并行比较器的逻辑符号

利用4位比较器和4位并行加法器实现2421BCD码转8421BCD码

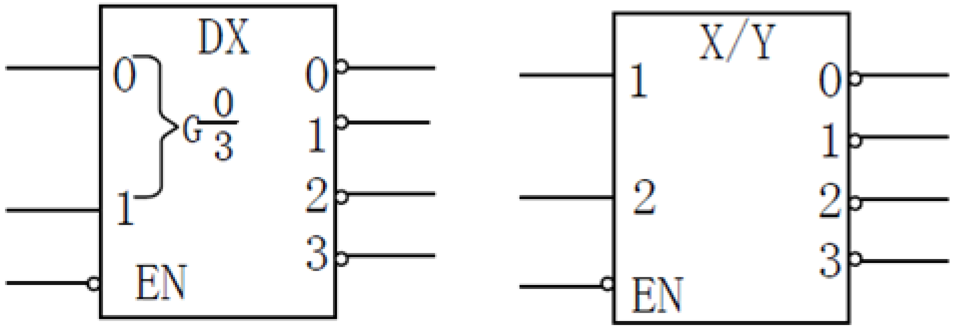
2421BCD				8421BCD			
A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

- **当输入数据小于等于0100时，输出等于输入**
- **当输入数据大于0100时，输出等于输入−6**
- **−6：$(1110)_{原码}$ $(1001)_{反码}$**

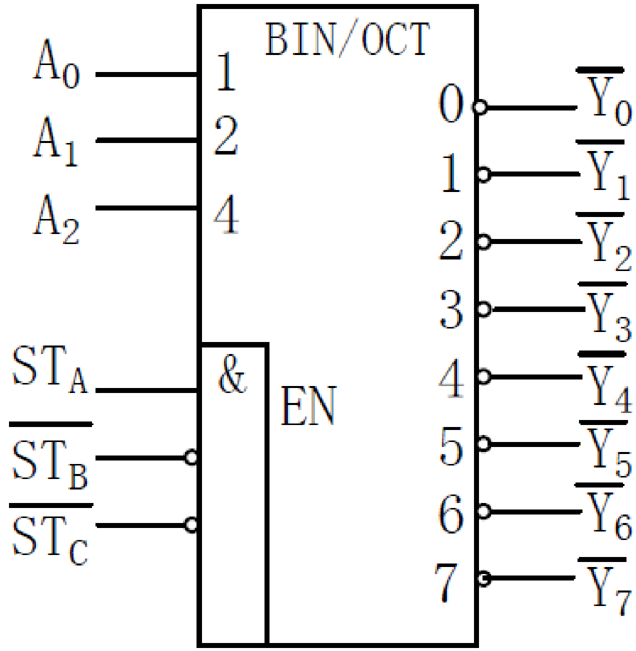


COMP
A0
A1
A2
A3
0
1
2
3
P
P<Q
P=Q
P>Q
B0
B1
B2
B3
0
1
2
3
Q
Σ
A0
A1
A2
A3
0
1
2
3
P
Σ
0
1
2
3
B0
B1
B2
B3
0
1
2
3
Q
CO
CI

➢ 变量译码器



2线-4线译码器



3线-8线译码器

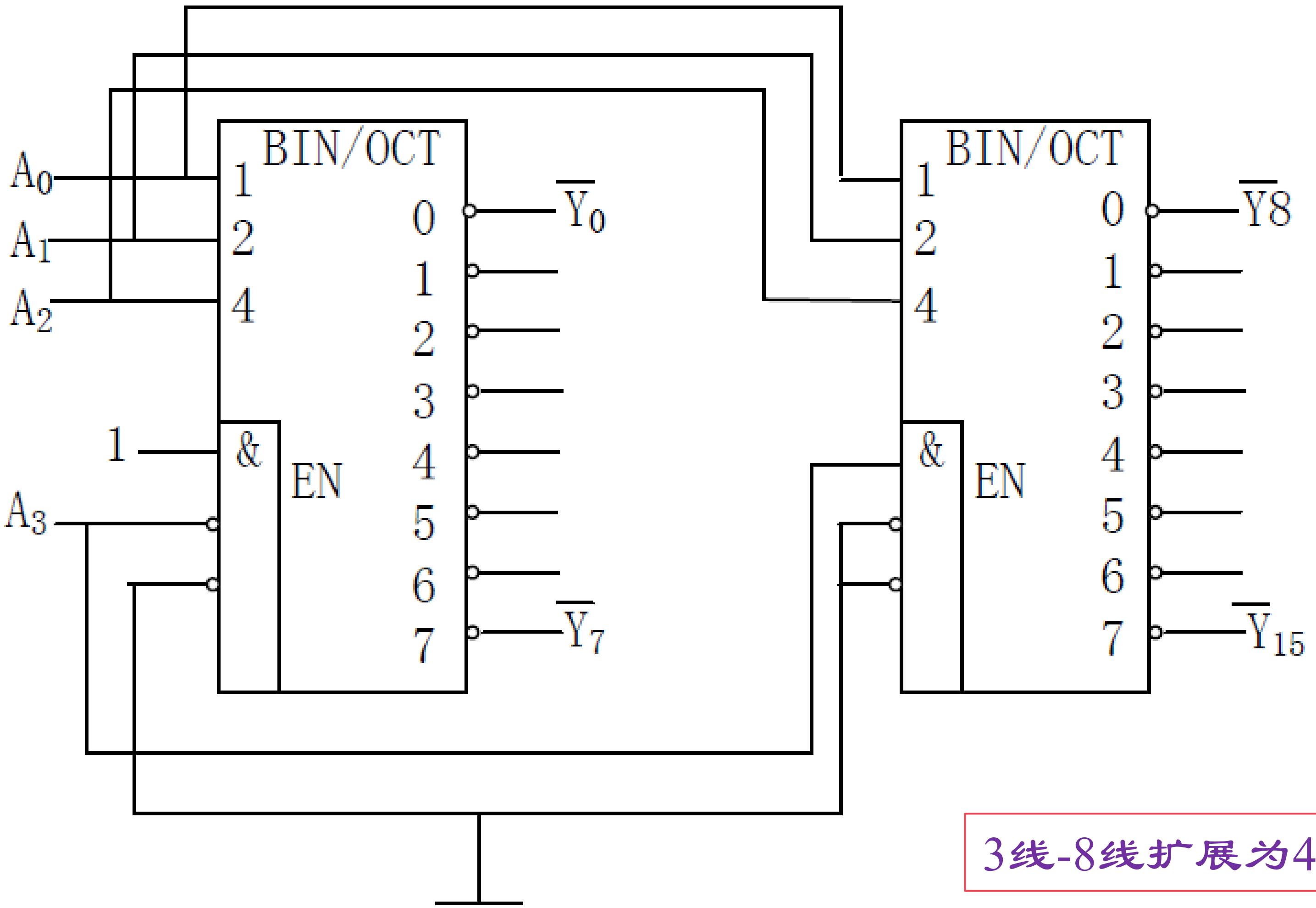
- BIN/OCT：二进制入，八进制出
- 控制输入端：ST_A、$\overline{ST_B}$、$\overline{ST_C}$
- $EN = ST_A \cdot \overline{\overline{ST_B}} \cdot \overline{\overline{ST_C}}$
- 输出端低电平有效：

$$\overline{Y_i} = \overline{ST_A \cdot \overline{\overline{ST_B}} \cdot \overline{\overline{ST_C}} \cdot m_i}$$

当$ST_A \cdot \overline{\overline{ST_B}} \cdot \overline{\overline{ST_C}} = 1$时，

$$\overline{Y_i} = \overline{m_i}$$

➢ 3线-8线译码器——使能端的运用



3线-8线扩展为4线-16线

➢ 变量译码器实现任意组合逻辑电路

利用$\overline{Y_i}=\overline{m_i}$，辅以适当门电路，可实现任何组合逻辑函数的标准“与或”、“或与”式

例：3线-8线译码器组成1位全加器

X_i	Y_i	C_{Ii}	$\sum_i$	Co_i
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

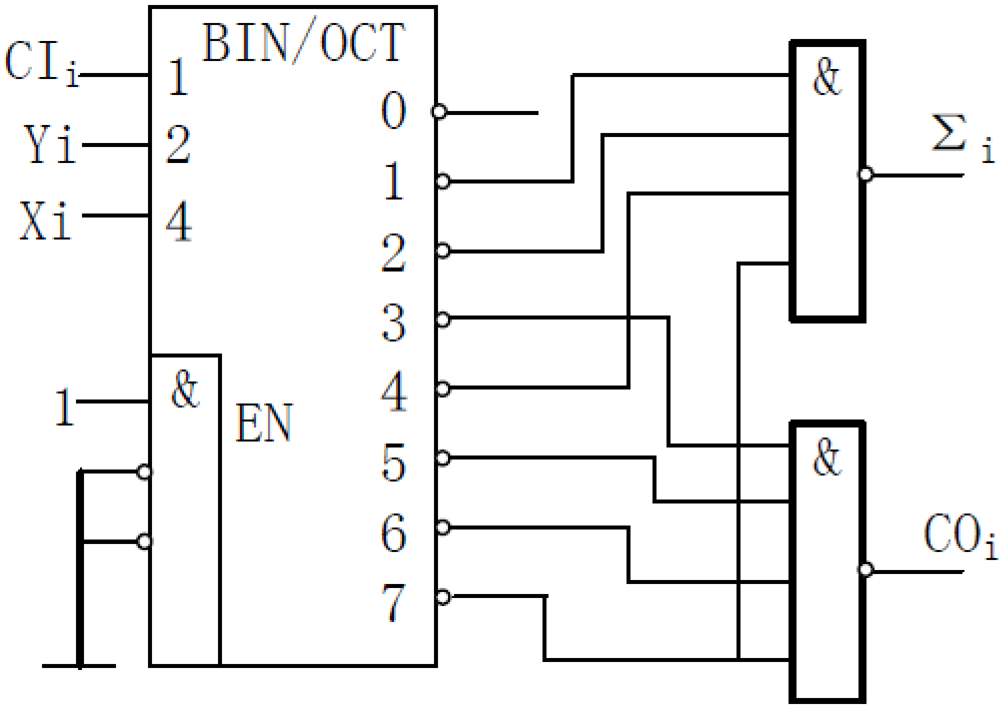
$$\Sigma_i(x_i,y_i,CI_i)=\Sigma_m(1,2,4,7)$$

$$CO_i(x_i,y_i,CI_i)=\Sigma_m(3,5,6,7)$$

$$\begin{aligned}\sum\nolimits_i(x_i,y_i,CI_i)&=m_1+m_2+m_4+m_7\\&=\overline{\overline{m_1+m_2+m_4+m_7}}\\&=\overline{\overline{m_1}\cdot\overline{m_2}\cdot\overline{m_4}\cdot\overline{m_7}}=\overline{\overline{Y_1}\cdot\overline{Y_2}\cdot\overline{Y_4}\cdot\overline{Y_7}}\end{aligned}$$

$$CO_i(x_i,y_i,CI_i)=\overline{\overline{Y_3}\cdot\overline{Y_5}\cdot\overline{Y_6}\cdot\overline{Y_7}}$$

例：3线-8线译码器组成1位全加器



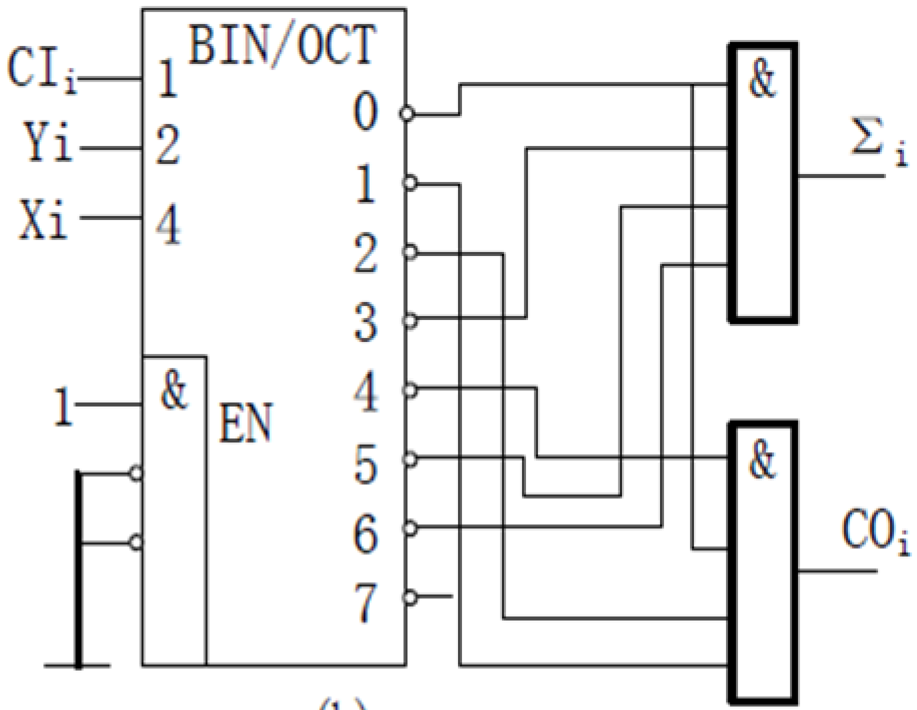
$$\Sigma_i(x_i, y_i, CI_i) = \Sigma_m(1,2,4,7)$$

$$CO_i(x_i, y_i, CI_i) = \Sigma_m(3,5,6,7)$$

$$\sum_i(x_i, y_i, CI_i) = m_1 + m_2 + m_4 + m_7$$
$$= \overline{\overline{m_1 + m_2 + m_4 + m_7}}$$
$$= \overline{\overline{m_1} \cdot \overline{m_2} \cdot \overline{m_4} \cdot \overline{m_7}} = \overline{\overline{Y_1} \cdot \overline{Y_2} \cdot \overline{Y_4} \cdot \overline{Y_7}}$$

$$CO_i(x_i, y_i, CI_i) = \overline{\overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7}}$$

例：3线-8线译码器组成1位全加器



(b)

$$\Sigma_i(x_i, y_i, CI_i) = \Sigma_m(1,2,4,7)$$

$$CO_i(x_i, y_i, CI_i) = \Sigma_m(3,5,6,7)$$

$$F = \sum_i m_i = \prod_{j \neq i} M_j$$

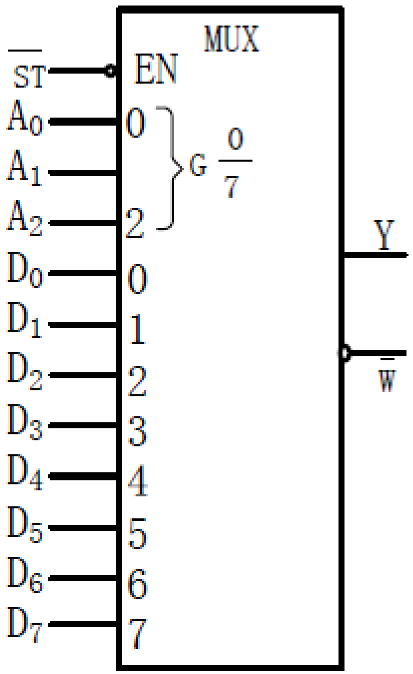
$$m_i = \bar{M}_i$$

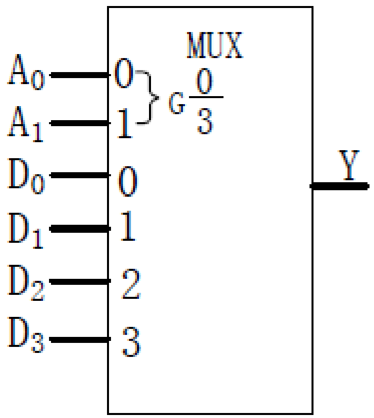
$$\sum_i (x_i, y_i, CI_i) = M_0 M_3 M_5 M_6 = \overline{\overline{M_0 M_3 M_5 M_6}} = \overline{\overline{M_0} + \overline{M_3} + \overline{M_5} + \overline{M_6}}$$

$$= \overline{m_0 + m_3 + m_5 + m_6} = \overline{m_0} \cdot \overline{m_3} \cdot \overline{m_5} \cdot \overline{m_6} = \bar{Y}_0 \cdot \bar{Y}_3 \cdot \bar{Y}_5 \cdot \bar{Y}_6$$

$$CO_i(x_i, y_i, CI_i) = M_0 M_1 M_2 M_4 = \bar{Y}_0 \cdot \bar{Y}_1 \cdot \bar{Y}_2 \cdot \bar{Y}_4$$

➢ 选择器





4选1数据选择器

$$Y=\overline{A_2}\,\overline{A_1}\,\overline{A_0}D_0+\overline{A_2}\,\overline{A_1}A_0D_1+\overline{A_2}A_1\overline{A_0}D_2+\overline{A_2}A_1A_0D_3+$$

$$A_2\overline{A_1}\,\overline{A_0}D_4+A_2\overline{A_1}A_0D_5+A_2A_1\overline{A_0}D_6+A_2A_1A_0D_7$$

$$=\sum_{i=0}^{7}m_iD_i$$

- $\overline{ST}$：使能端
  $\overline{ST}=0$，选中，选择器工作
  $\overline{ST}=1$，不选中，$Y=0$，$\overline{W}=1$
- $A_2A_1A_0$：地址输入
- $D_7\sim D_0$：数据输入

- $A_2A_1A_0=011$，选中D_3：
  $Y=D_3$（原码） $\overline{W}=\overline{D_3}$（反码）

用选择器实现组合逻辑函数

有 n 个地址变量的 2^n 选 1 MUX

$$Y(A_{n-1}, A_{n-2}, \ldots, A_0) = \sum_{i=0}^{2^n-1} m_i D_i$$

n 个输入变量的组合函数的最小项表达式

$$F(x_{n-1}, x_{n-2}, \ldots, x_0) = \sum_{i=0}^{2^n-1} m_i a_i$$

$a_i = 0$ 或 $a_i = 1$

- 令组合函数的自变量为 MUX 的地址变量
- 由组合函数的最小项表达式决定 a_i,
- 令 $D_i = a_i$

- 用数据选择器实现组合逻辑函数
  - $n=m \quad D_i=a_i$ 其中：n—地址数 m—变量数
  - $n>m$：地址高位接地
  - $n<m$：用卡诺图降维
    - 卡诺图的变量数称为该图的维数
    - 将某些变量作为卡诺图方格内的值，可以降低维数

- 用数据选择器实现组合逻辑函数
  - $n<m$：用卡诺图降维
    - 卡诺图的变量数称为该图的维数
    - 将某些变量作为卡诺图方格内的值，可以降低维数

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	1	0	0	0
10	1	0	0	0

A \ BC	00	01	11	10
0	0	$\overline{D}$	$\overline{D}$	0
1	$\overline{D}$	0	0	$\overline{D}$

记图变量为X，对于原卡诺图，当$X=0$时，原图单元值为F，$X=1$时，原图单元值为G，则在新的降维图中，对应的降维图单元中填入子函数$\bar{X}F+XG$

➢ 用数据选择器实现组合逻辑函数

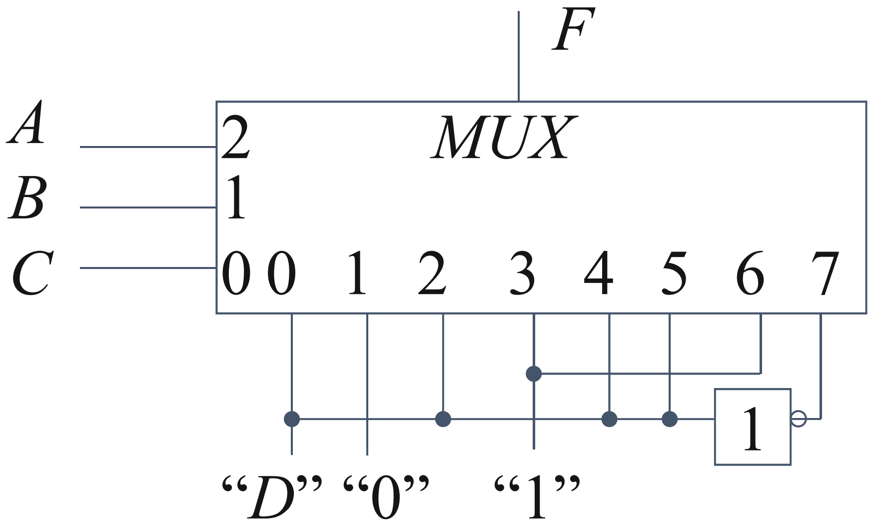
➢ $n < m$：用卡诺图降维

例 （$n < m$）用8选1数据选择器实现函数$F(A,B,C,D) = \sum m(1, 5, 6, 7, 9, 11, 12, 13, 14)$

解

AB \ CD	00	01	11	10
00	0	1	0	0
01	0	1	1	1
11	1	1	0	1
10	0	1	1	0

A \ BC	00	01	11	10
0	D	0	1	D
1	D	D	$\overline{D}$	1

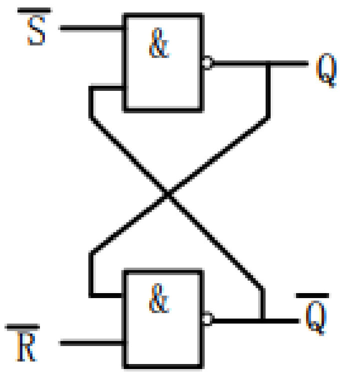


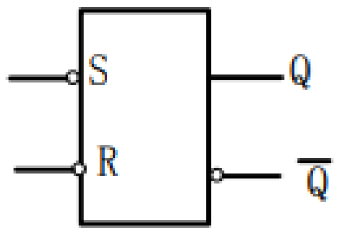
第四章 时序电路基础

主要内容

- 集成触发器
  - R-S触发器、D触发器、J-K触发器
- 同步时序电路
  - 逻辑系统的描述、米里型和莫尔型状态表、状态图、功能描述和典型电路
- 集成计数器及其应用
  - 异步计数器、同步集成计数器和类型、任意进制计数器设计、计数器扩展、计数器应用范例
- 集成移位寄存器及其应用
  - 集成移位寄存器功能和应用，串—并变换和并—串变换，线性移位寄存器

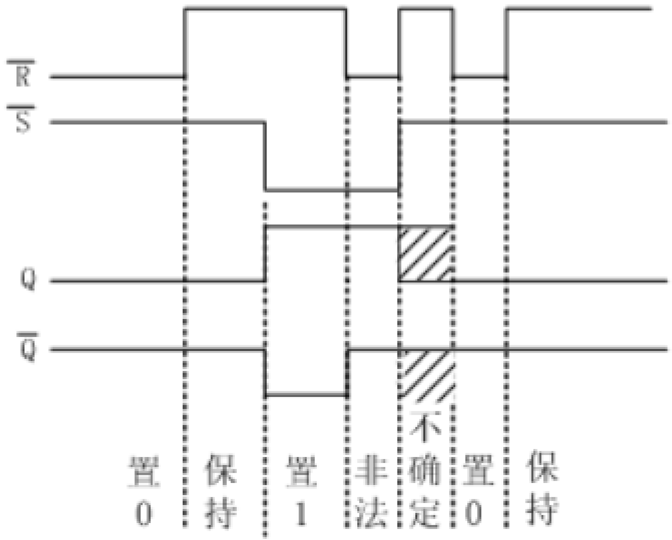
➢ 基本R-S触发器





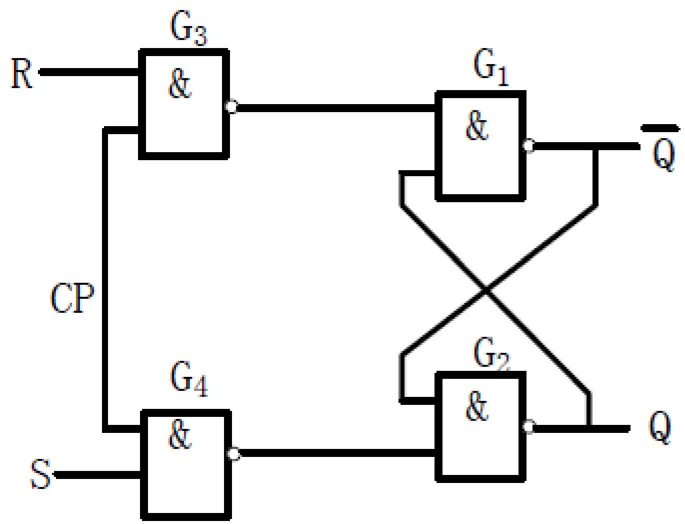
状态方程： $Q^{n+1} = \bar{\bar{S}} + Q^n\bar{R}$

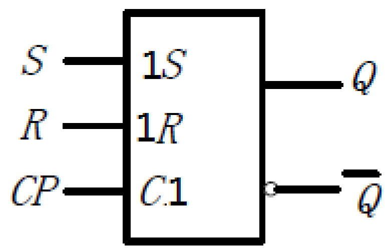
约束方程： $\bar{\bar{R}}\bar{\bar{S}} = 0 \Rightarrow \bar{R} + \bar{S} = 1$

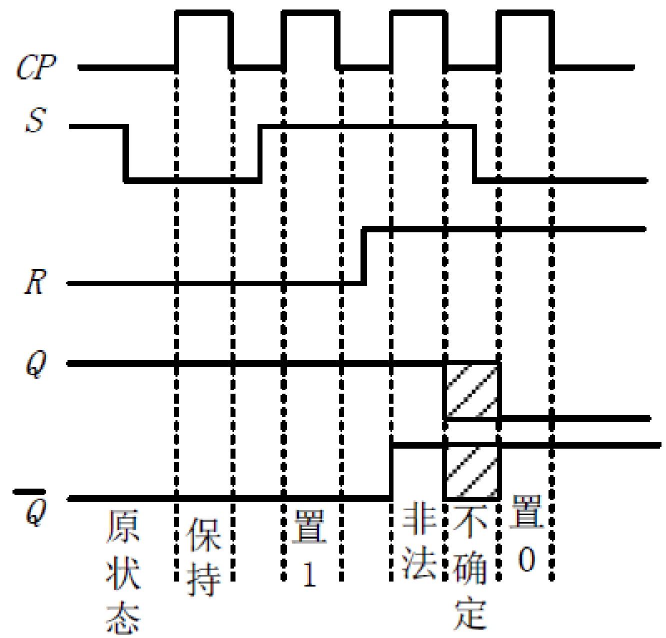


基本 RS 触发器的工作波形

➢ 钟控R-S触发器



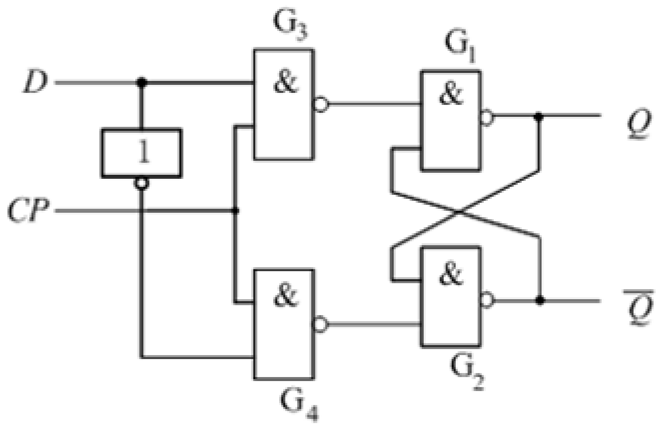


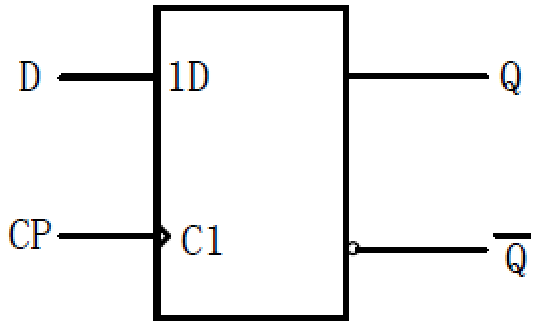


状态方程： $Q^{n+1} = S + \bar{R}Q^n$

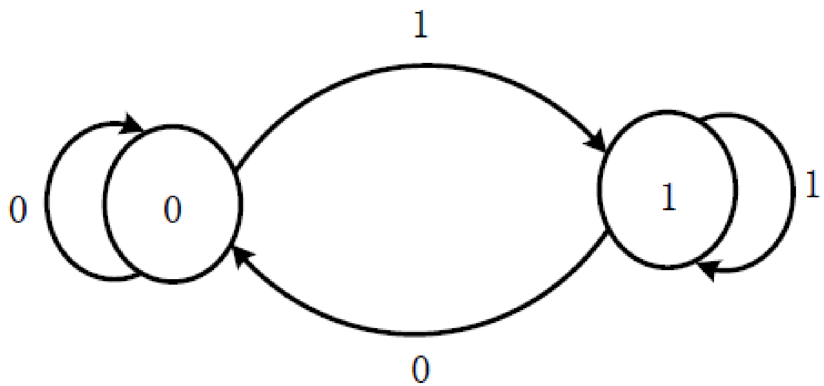
约束方程： $RS = 0$

➢ D触发器

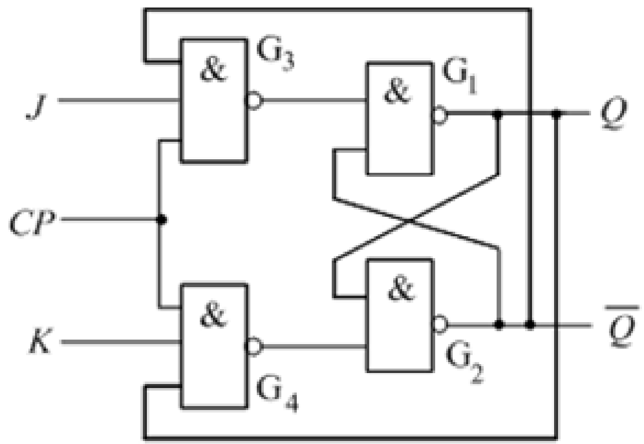


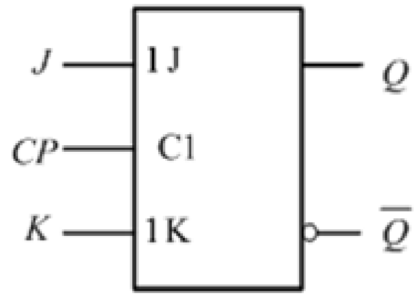


状态方程： $Q^{n+1} = D$

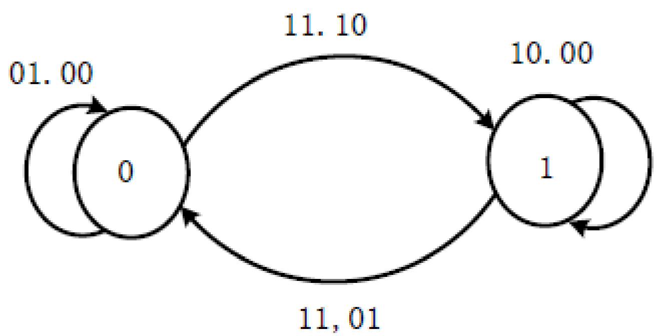


➢ JK触发器

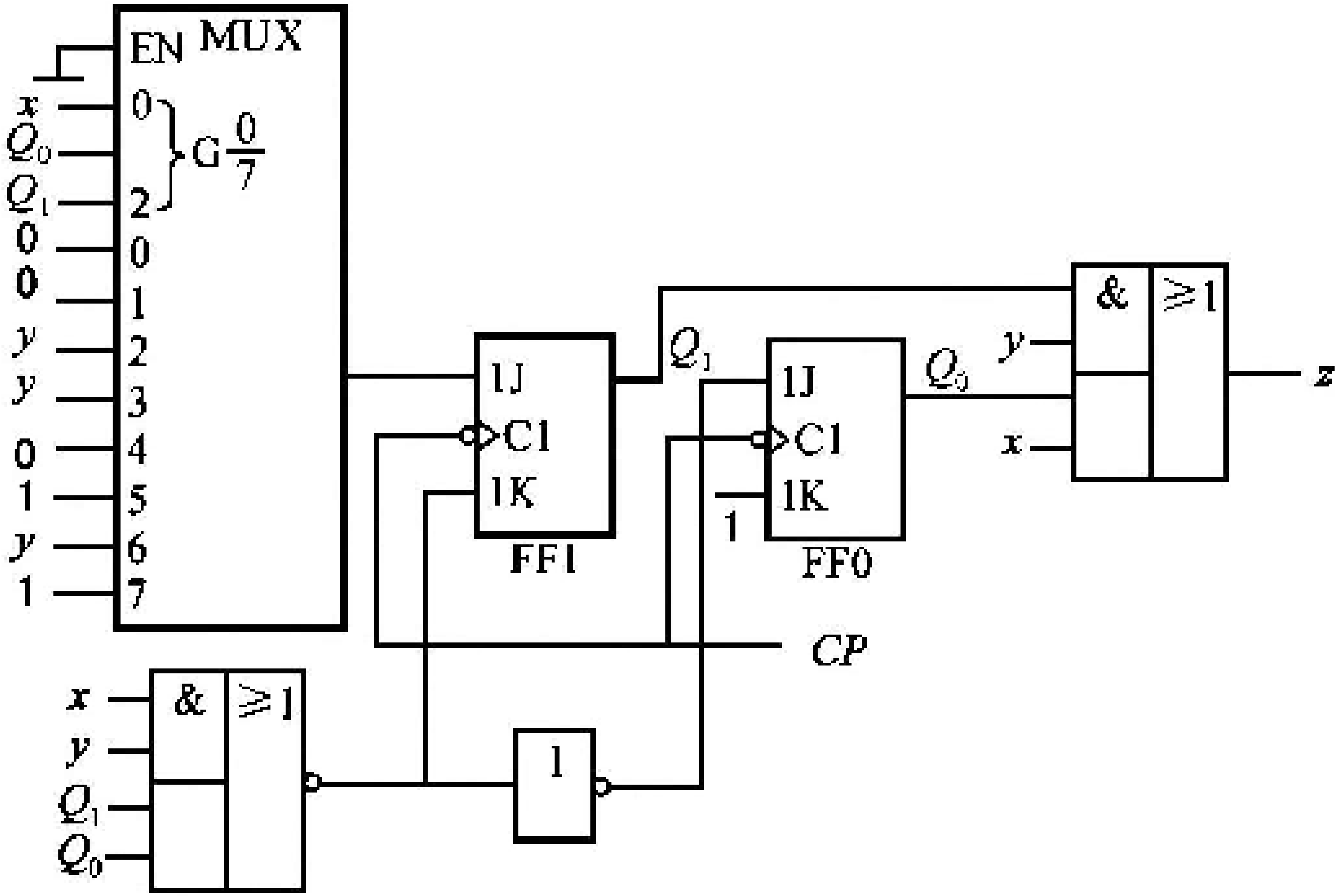




状态方程： $Q^{n+1} = J\overline{Q^n} + \overline{K}Q^n$

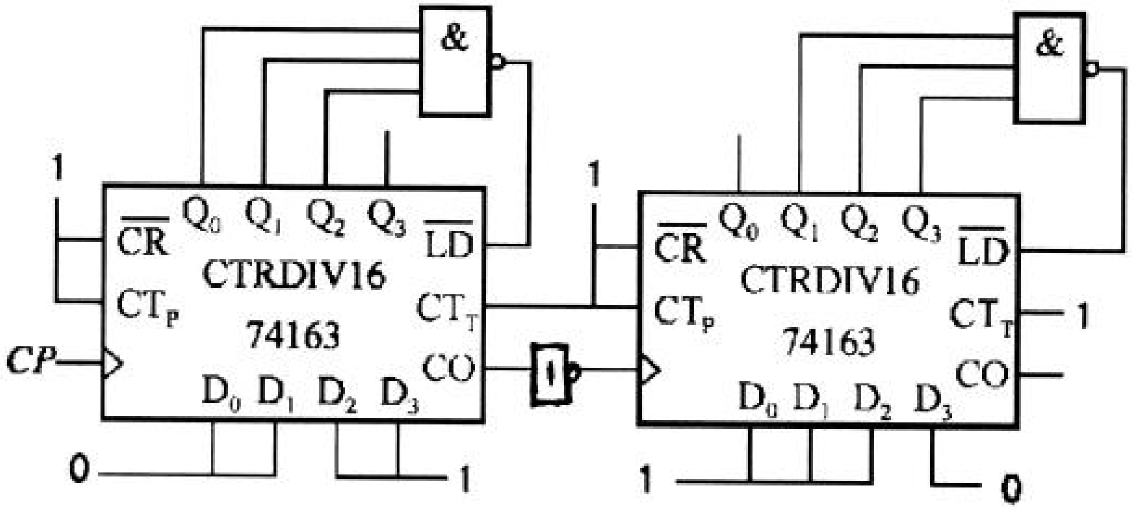


➢ 时序电路分析



EN MUX
G $\frac{0}{7}$
0
2
0
1
2
3
4
5
6
7
x
Q_0
Q_1
0
0
y
y
0
1
y
1
1J
C1
1K
FF1
Q_1
1J
C1
1K
FF0
1
Q_0
y
x
&
≥1
z
CP
x
y
Q_1
Q_0
&
≥1
1

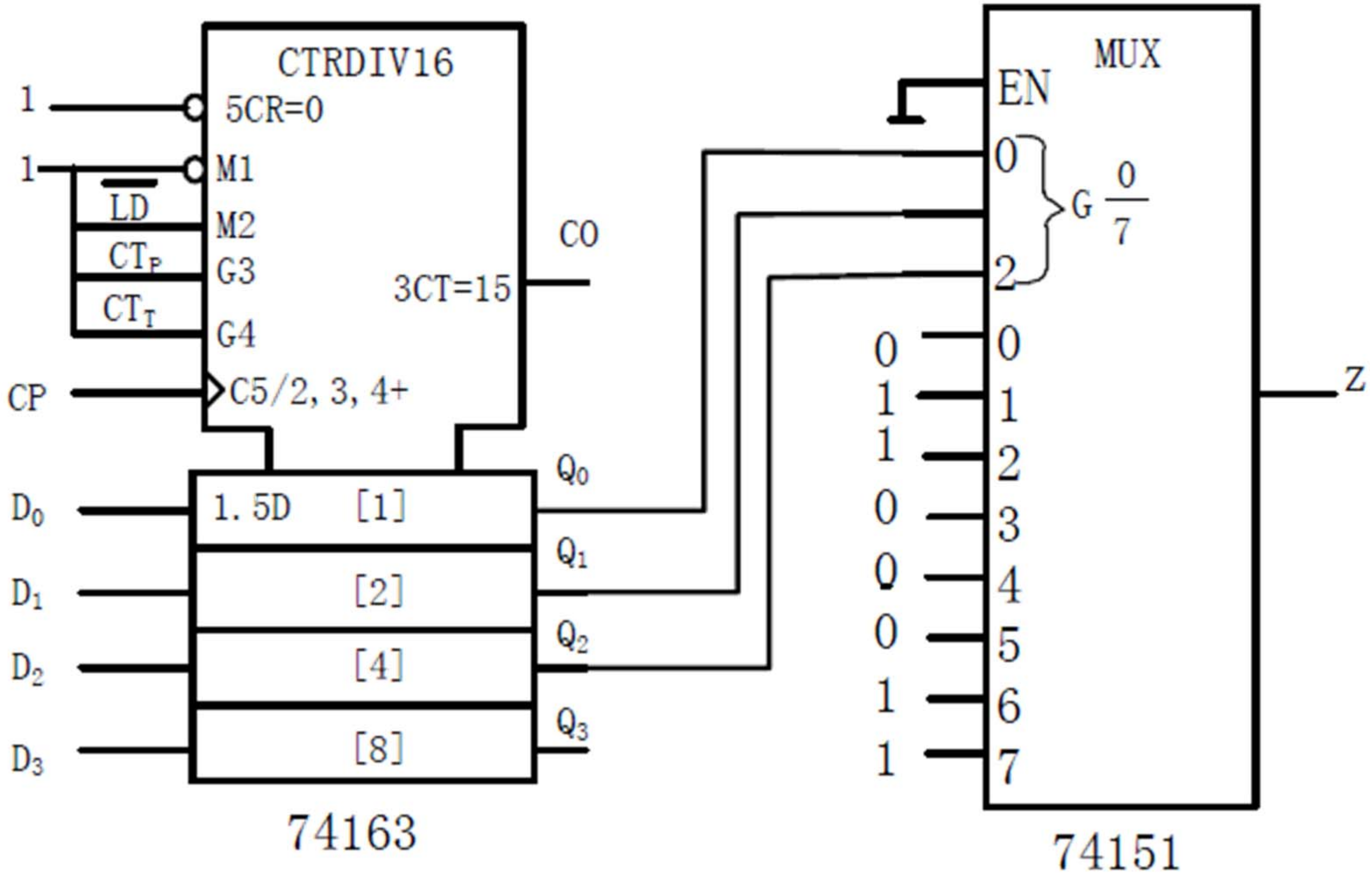
> 集成计数器电路分析



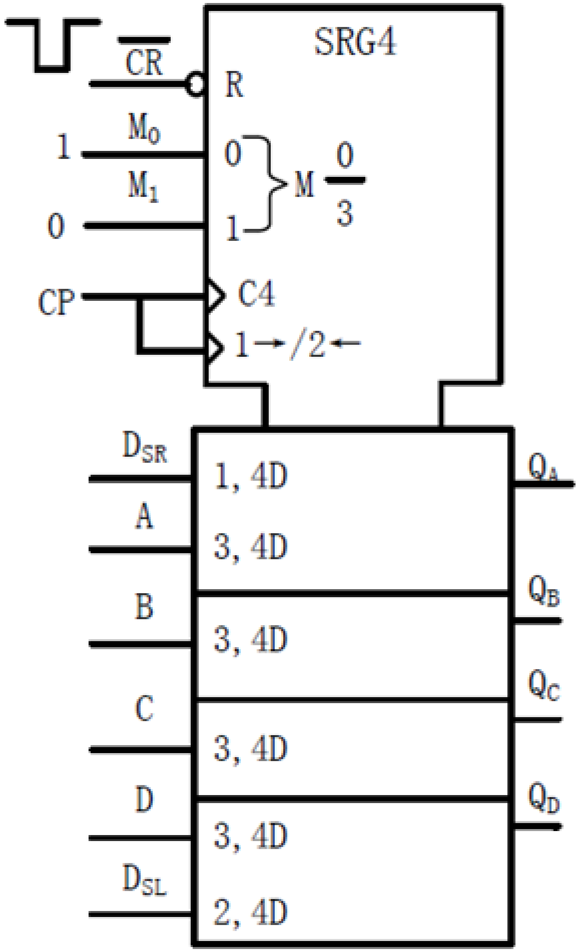
同步/异步清零
同步/异步置位

➢ 集成计数器电路应用

例：用74163设计一个01100011序列发生器



➢ 典型移位寄存器——74194



- SRG4：4位移位寄存器
- $\overline{CR}$异步复位，低电平有效
- M_0、M_1：控制输入端。

当$\overline{CR}=1$时，$M_1M_0=01$，右移

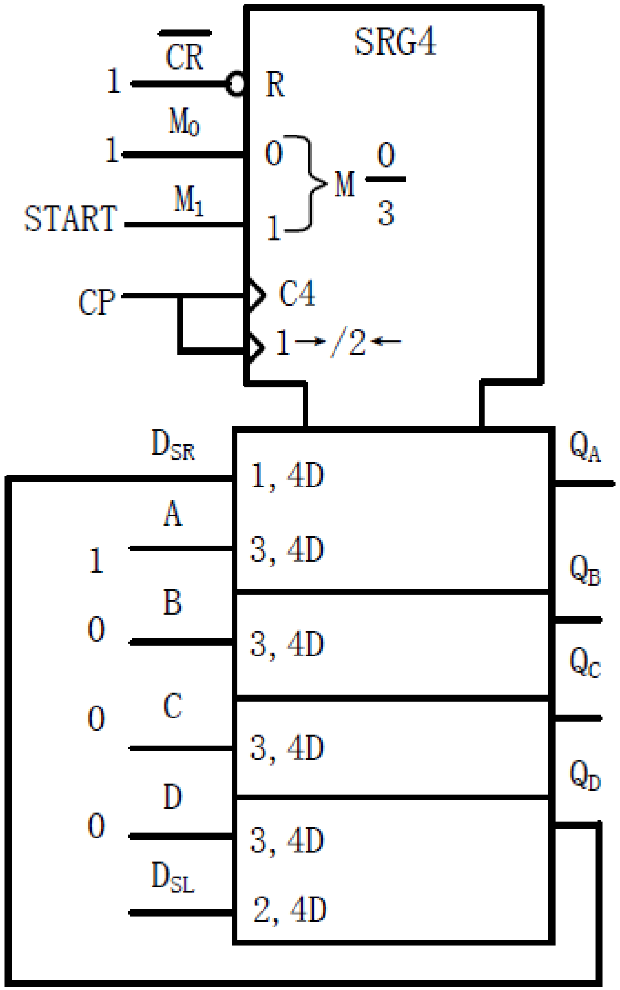
$M_1M_0=10$，左移

$M_1M_0=11$，置数

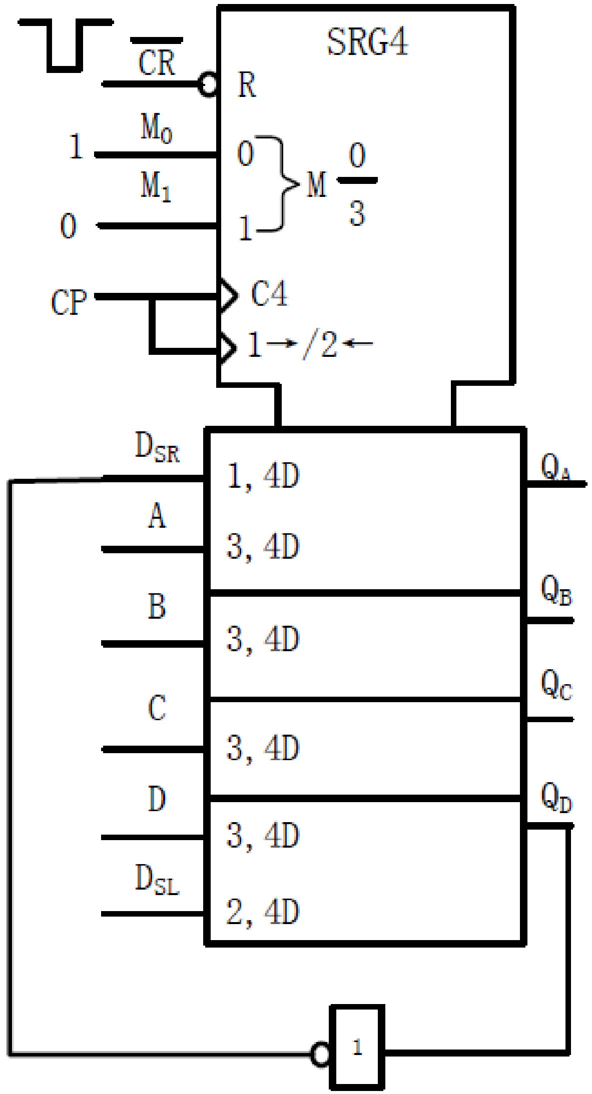
$M_1M_0=00$，保持

- D_{SR}：右移串行输入
- Q_D：右移串行输出
- D_{SL}：左移串行输入
- Q_A：左移串行输出
- A、B、C、D：并行输入端
- $Q_AQ_BQ_CQ_D$：并行输出端

环形计数器



扭环形计数器



上海大学 Shanghai University

第五章

同步时序电路和数字系统设计

主要内容

- 状态图、状态表和同步时序电路基本设计方法
- 初始状态图、状态表的建立、状态表化简、状态编码
- 用JK、D触发器实现同步时序组合状态法、直接状态法电路

➢ 基本设计步骤

a. 根据要求，作出原始状态图或状态表

b. 进行状态化简

c. 状态分配(赋码)，决定触发器的个数

d. 触发器选型，写出各触发器输入端方程、输出方程和次态方程

e. 检查所设计的电路能否自启动，如不能，应修改成自启动电路

f. 作出逻辑图，并画出完整状态图

➢ 原始状态图和状态表的建立

1. 分析电路的逻辑功能，明确输入信号的逻辑组合，确定逻辑功能的时间顺序和特点。
2. 列出电路不同的输入/输出序列的特征，确定电路应包含的状态，称为“现态”
3. 考察各种可能输入组合作用下电路由现态转入次态的相应逻辑输出
4. 按功能描述和时序分析现态和次态关系，构成完整状态图和状态表

第六章

集成ADC和DAC的基本原理与结构

- DAC的原理和主要技术参数
  - 二进制全电阻网络DAC、二进制T型电阻电压网络DAC、DAC技术参数
- ADC的原理和主要技术参数
  - 并行ADC、二进制逐次比较ADC、ADC技术参数

➢ DAC

$$v_O = -\frac{V_{REF}}{2^n}\sum_{i=0}^{n-1} D_i \times 2^i$$

- **最小输出电压：** V_{LSB}指输入数字量$D = D_{n-1} \dots D_0$中仅当最低位（D_0）的数码为1时，对应的输出模拟电压值。

- **满量程输出电压：** V_{FSR}是输入数字量各位为全1时，对应的输出模拟电压值。

- **分辨率：** 能够分辨最小输出电压的能力

$$\text{分辨率} = \frac{V_{LSB}}{V_{FSR}} = \frac{1}{2^n - 1}$$

例： 某 8 位权电阻 DAC，若$V_{\mathrm{REF}}=5.00\mathrm{V}$ ，则：

$$V_{LSB}=\frac{V_{REF}}{2^8}=\frac{5}{256}\approx 20\text{（mV）}$$

$$V_{\mathrm{FSR}}=\frac{V_{REF}}{2^8}\sum_{i=0}^{7}D_i\times 2^i\approx 4.98\text{（V）}$$

或 $V_{\mathrm{FSR}}=V_{REF}-V_{LSB}=5.0\mathrm{V}-20\mathrm{mV}=4.98\mathrm{V}$

习题6.1

- **绝对误差：** 实际值与理想值之间的最大差值。通常以V_{LSB}或V_{FSR}的倍数来表示
- **相对误差：** 绝对误差与满量程的比值，以满量程的百分数表示

例：已知$V_{\mathrm{REF}}=8\mathrm{V}$，12DAC，绝对误差为±LSB。求绝对误差电压值和相对误差值。

解：绝对误差 $\pm\frac{8}{2^{12}}=\pm 1.9\mathrm{mV}$

相对误差 $$\frac{\pm\frac{8}{2^{12}}}{\frac{8\cdot\left(2^{12}-1\right)}{2^{12}}}=\pm\frac{1}{2^{12}-1}=\pm 0.0244\%$$

➢ ADC

- 对逐次比较型A/D转换器，比较周期为

二进制：$n=6$，$T=6T_{CP}$

BCD：$4\frac{1}{2}$BCD: T=17T_{CP}

注意：完成一个完整的AD转换，需要nT_{CP}

$4\dfrac{1}{2}BCD$	• 5位二进制组成，低4位可显示0~9，最高位只有0和1 • 数值范围：00000~19999 • 数码长度17

习题6.12

第七章

可编程逻辑器件及其应用基础

- PLD（可编程逻辑器件）
- ROM
- 可编程阵列

习题7.2

第八章
硬件描述
语言基础

➢ VHDL

- 头文件：对被调用库和程序包的说明
- 电路实体定义和实体说明
- 电路结构和行为的描述

表 8-3　VHDL 的运算操作符

类别	运算符号
算术运算符	+、−、*、/、**、MOD、REM、ABS
关系运算符	NOT、AND、OR、NOR、NAND、XOR
逻辑运算符	=（相等）、/=（不等）、<（小于）、<=（小于等于）、>（大于）、>=（大于等于）
连接运算符	&

➢ 对代码的理解

例8-4：4位8421BCD编码器

例8-6：译码器

例8-8：选择器

例8-9：全加器

例8-19/20：D锁存器

例8-22：非同步复位/置位D锁存器

例8-24：JK触发器

例8-27：六进制计数器