

数字电路与逻辑设计B

期末复习

南京邮电大学

电子与光学工程学院

臧裕斌

主题一 数制与码制

1.1 数制基本概念：计数规则

1.2 数制三要素：数码、位权、基数

1.3 常用数制

计数制	数码	基数	位权	举例
十进制	0~9	10	10^i	$(123)_{10}$ $(456.321)_D$
二进制	0、1	2	2^i	$(1010)_2$ $(1001.101)_B$
八进制	0~7	8	8^i	$(567)_8$ $(745.217)_O$
十六进制	0~9、A~F	16	16^i	$(2A2B)_{16}$ $(1B3.EC)_H$

主题一 数制与码制

1.4 数制转换

(1) 二转十——按权展开

$$(10.01)_B = 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (2.25)_D$$

(2) 十转二——基数乘除

整数部分：除2取余，倒序整理

小数部分：乘2取整，顺序整理

	余数	有效位		整数	有效位
2 57				0.6875	
2 28	1	k_0 (最低位)	$\times 2$	1.3750	1 k_{-1} (最高位)
2 14	0	k_1	$\times 2$	0.7500	0 k_{-2}
2 7	0	k_2	$\times 2$	1.5000	1 k_{-3}
2 3	1	k_3	$\times 2$	1.0000	1 k_{-4} (最低位)
2 1	1	k_4			
0	1	k_5 (最高位)			

主题一 数制与码制


1.4 数制转换

(3) 二转八——三个一组，整体缩合

(4) 八转二——三个一组，整体展开

(5) 二转十六——四个一组，整体缩合

(6) 十六转二——四个一组，整体展开

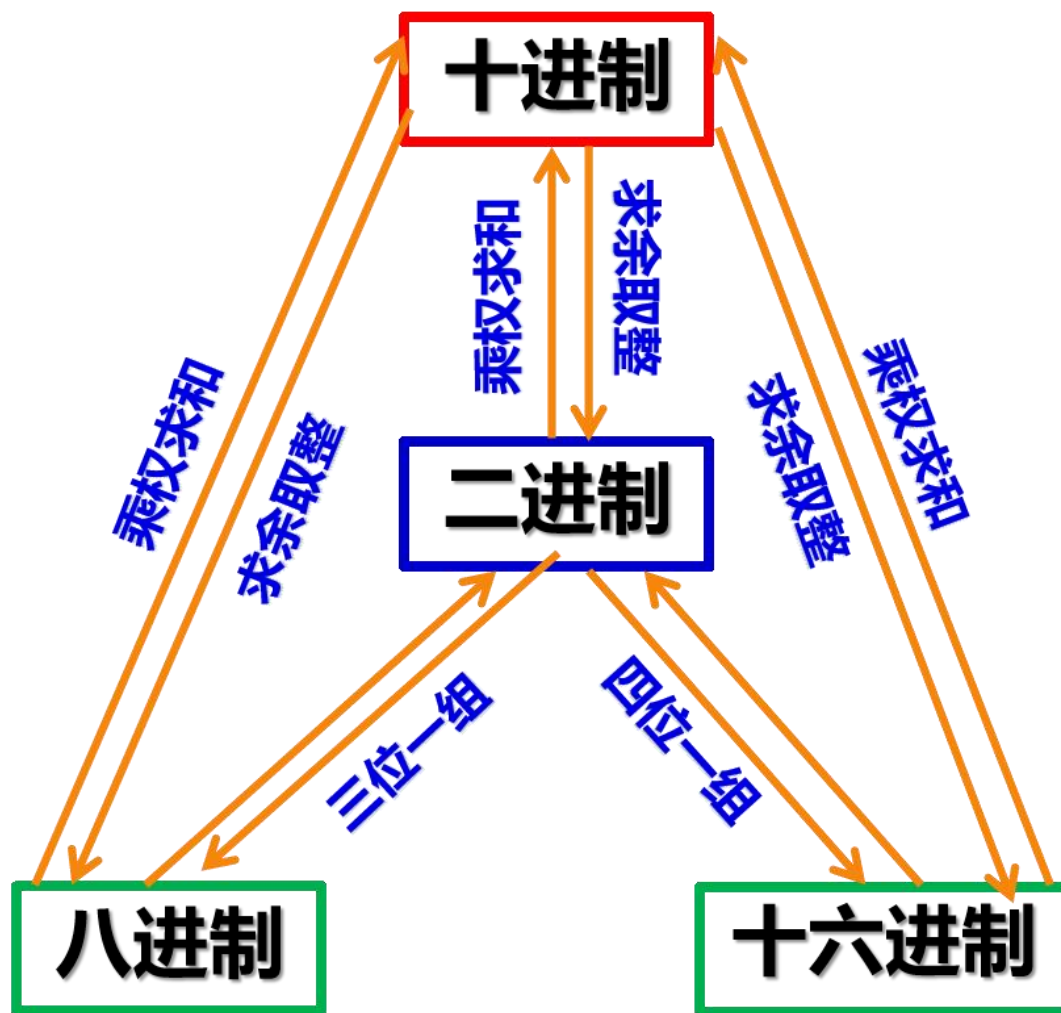
$$(101011.1)_2 = (101011.100)_2 = (53.4)_8$$


$$(101011.1)_2 = (00101011.1000)_2 = (2B.8)_{16}$$

(7) 2^n 转 2^m ——借助二进制中转

(8) 其他——借助十进制中转

1.4 数制转换



主题一 数制与码制

1.5 常见二进制码

- (1) 自然二进制码
- (2) 循环二进制码（格雷码）
- (3) 8421BCD码
- (4) 余三码
- (5) 5421BCD码（格雷码）

主题一 数制与码制

1.5 常见二进制码

十进制数	8421码	余3码	循环码	余3循环码
0		0011		0010
1		0100		0110
2		0101		0111
3	0011	0110	0010	0101
4	0100	0111	0110	0100
5	0101	1000	0111	1100
6	0110	1001	0101	1101
7	0111	1010	0100	1111
8	1000	1011	1100	1110
9	1001	1100	1101	1010
	1010		1111	
	1011		1110	
	1100		1010	

主题一 数制与码制

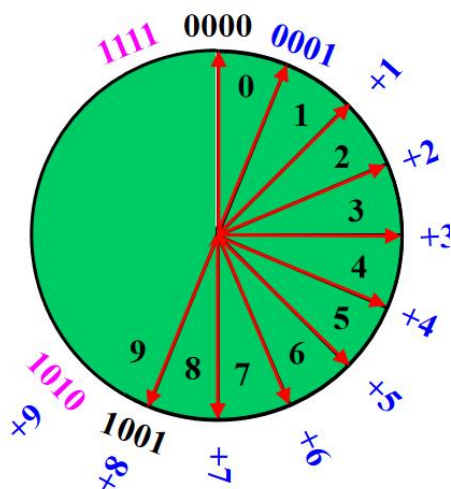
1.6 8421BCD码

(1) 合法码范围：0000-1001

(2) BCD数加法——加0110修正的时机

$$\begin{array}{r} 0001 \\ + 1001 \\ \hline 1010 \quad \text{非法码} \\ + 0110 \quad \text{加6修正} \\ \hline 0001 \quad 0000 \end{array}$$

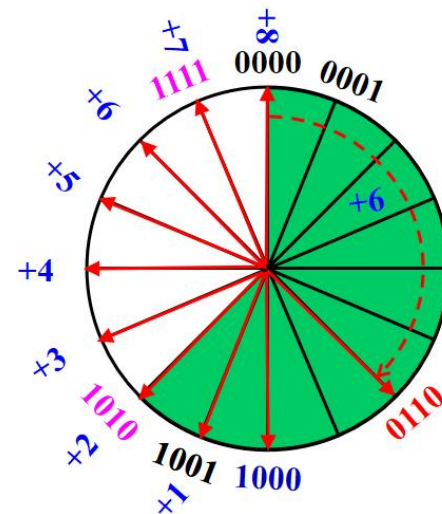
解：



$$(0001)_{8421BCD} + (1001)_{8421BCD} = (0001 \quad 0000)_{8421BCD}$$

$$\begin{array}{r} 1000 \\ + 1000 \\ \hline 1 \quad 0000 \quad \text{向高位进位} \\ + 0110 \quad \text{加6修正} \\ \hline 0001 \quad 0110 \end{array}$$

解：

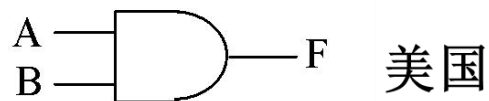
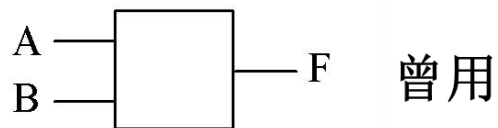
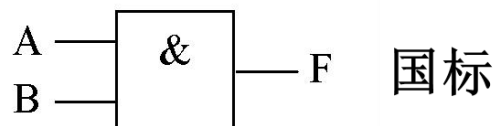


$$(1000)_{8421BCD} + (1000)_{8421BCD} = (0001 \quad 0110)_{8421BCD}$$

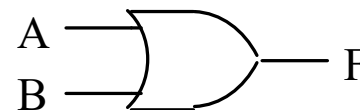
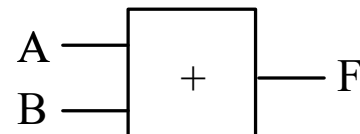
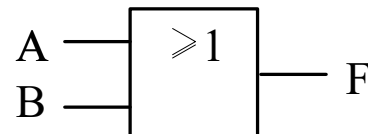
主题二 逻辑代数基础

2.1 逻辑代数基本概念、运算与符号

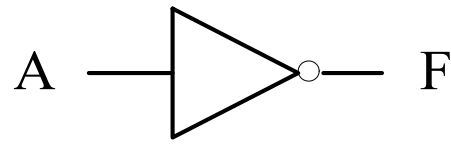
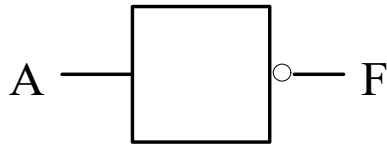
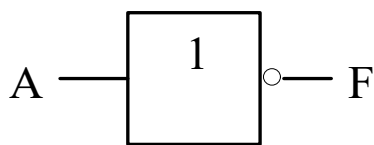
(1) 与 $F = A \cdot B$



(2) 或 $F = A + B$



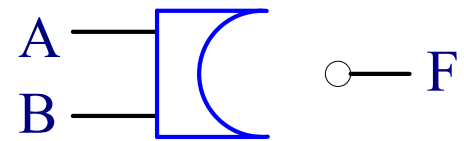
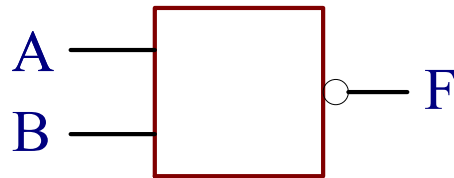
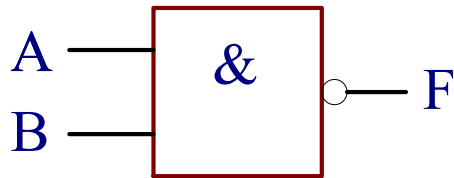
(3) 非 $F = \overline{A}$



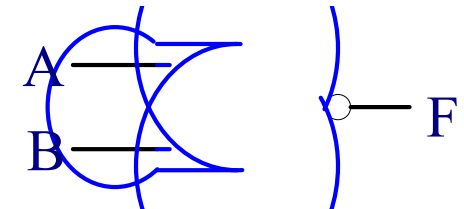
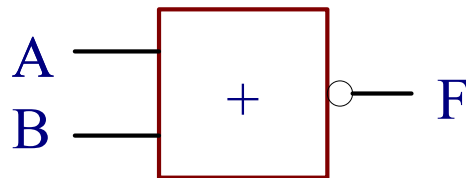
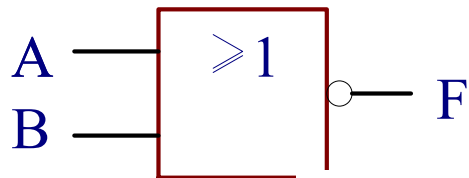
主题二 逻辑代数基础

2.1 逻辑代数基本概念、运算与符号

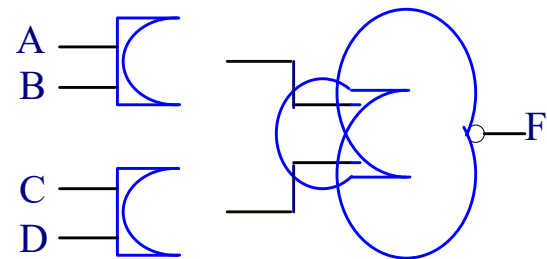
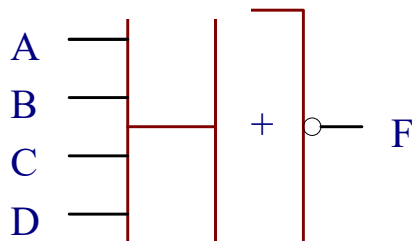
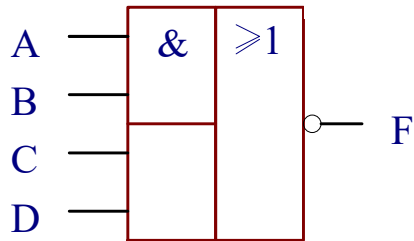
(4) 与非 $F = \overline{AB}$



(5) 或非 $F = \overline{A+B}$



(6) 与或非 $F = \overline{AB+CD}$



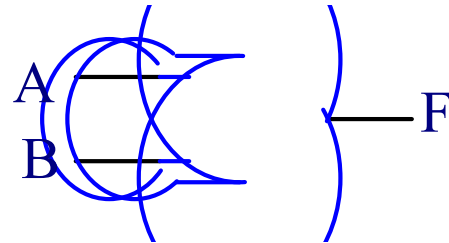
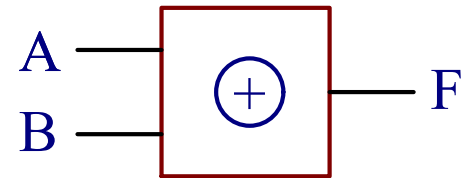
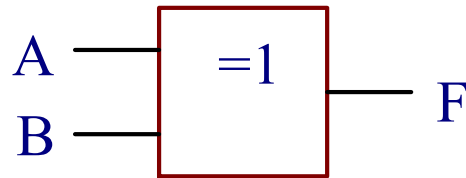
主题二 逻辑代数基础

2.1 逻辑代数基本概念、运算与符号

(7) 异或

$$F = A \oplus B = A\bar{B} + \bar{A}B$$

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



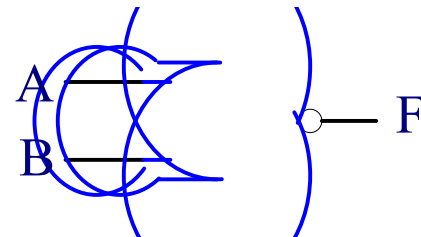
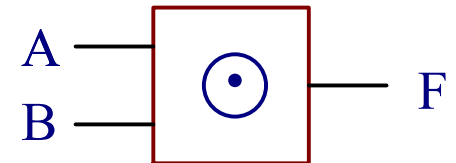
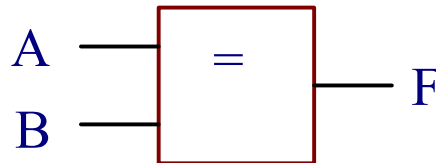
主题二 逻辑代数基础

2.1 逻辑代数基本概念、运算与符号

(8) 同或

$$F = A \odot B = AB + \bar{A}\bar{B}$$

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



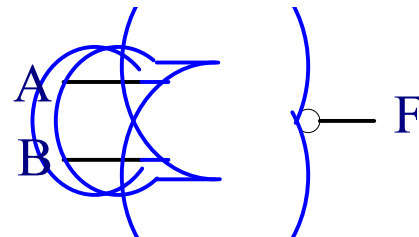
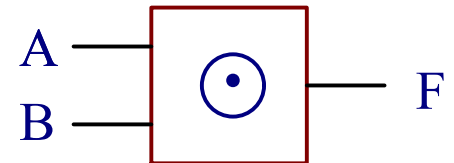
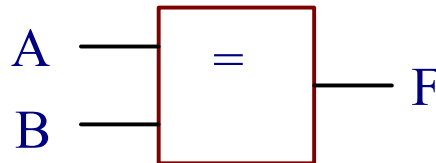
主题二 逻辑代数基础

2.1 逻辑代数基本概念、运算与符号

(8) 同或

$$F = A \odot B = AB + \bar{A}\bar{B}$$

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



主题二 逻辑代数基础

2.2 逻辑代数基本运算公式

1.自等律

$$A + 0 = A$$

$$A \cdot 1 = A$$

2.吸收律

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

3.重叠律

$$A + A = A$$

$$A \cdot A = A$$

4.互补律

$$A + \overline{A} = 1$$

$$A \cdot \overline{A} = 0$$

5.还原律

$$\overline{\overline{A}} = A$$

6.交换律

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

主题二 逻辑代数基础

2.2 逻辑代数基本运算公式

1.自等律

$$A + 0 = A$$

$$A \cdot 1 = A$$

2.吸收律

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

3.重叠律

$$A + A = A$$

$$A \cdot A = A$$

4.互补律

$$A + \overline{A} = 1$$

$$A \cdot \overline{A} = 0$$

5.还原律

$$\overline{\overline{A}} = A$$

6.交换律

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

主题二 逻辑代数基础

2.2 逻辑代数基本运算公式

7.结合律

$$A + B + C$$

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

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

$$A \cdot B \cdot C$$

$$= (A \cdot B) \cdot C$$

$$= A \cdot (B \cdot C)$$

8.分配律

$$A \cdot (B + C)$$

$$= AB + AC$$

$$A + BC$$

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

9.反演律

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\overline{AB} = \bar{A} + \bar{B}$$

主题二 逻辑代数基础

2.3 逻辑代数常用公式

1.合并相邻项公式 $AB + A\bar{B} = A$

2. 消项公式 $A + AB = A$

3. 消去互补因子公式 $A + \bar{A}B = A + B$

4. 多余项（生成项）公式

$$AB + \bar{A}C + BC = AB + \bar{A}C$$

证明： $AB + \bar{A}C + BC = AB + \bar{A}C + (A + \bar{A})BC$

$$= \underline{AB} + \underline{\bar{A}C} + \underline{ABC} + \underline{\bar{A}BC} = AB + \bar{A}C$$

主题二 逻辑代数基础

2.4 逻辑函数运算三大规则

(1) 代入规则——适用于等式

(2) 反演规则

(3) 对偶规则

F		\overline{F}
.	→	+
+	→	.
1	→	0
0	→	1
A	→	\overline{A}
\overline{A}	→	A

F		F'
.	→	+
+	→	.
1	→	0
0	→	1

主题二 逻辑代数基础

2.5 逻辑函数的表达形式

(1) 与非——与非

(2) 或非——或非

(3) 标准与或式（由最小项构成）

【最小项】是一种特殊的乘积项（与项），在该乘积项中逻辑函数的所有变量都要以原变量或反变量的形式出现一次，而且只能出现一次。

(4) 最简与或式（如何化简？）

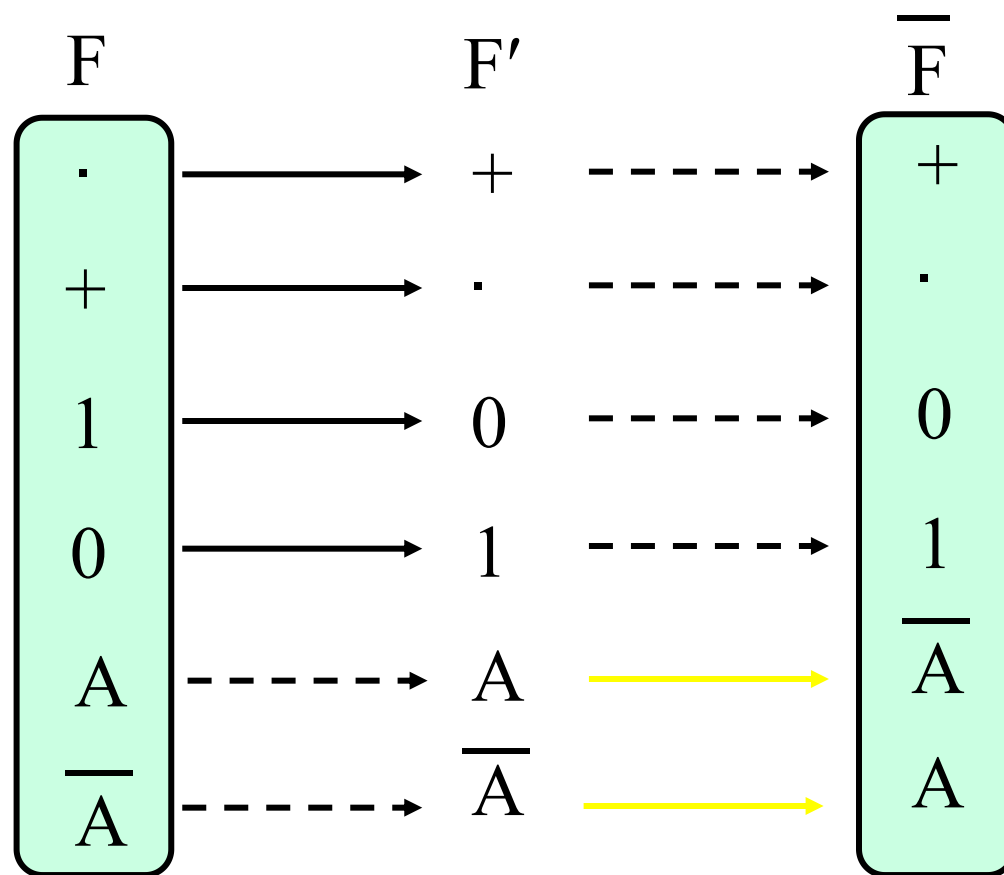
【公式法】

【卡诺图】

主题二 逻辑代数基础

2.6 原函数、反函数与对偶函数

(1) 公式法之间的关联



主题二 逻辑代数基础

2.6 原函数、反函数与对偶函数

(2) 标准与或式之间的关联

$$\text{若 } F = \sum m_j, \text{ 则 } \overline{F} = \sum m_k$$

$$\text{若 } \overline{F} = \sum m_j, \text{ 则 } F' = \sum m_k \quad (k = (2^n - 1) - j)$$

主题二 逻辑代数基础

2.7 逻辑函数的化简

(1) 公式法

(2) 卡诺图法

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

$AB \backslash CD$		CD			
		00	01	11	10
00	0	0	0	0	
01	1	1	0	1	
11	0	1	1	1	
10	\emptyset	\emptyset	\emptyset	\emptyset	

主题三 组合逻辑电路

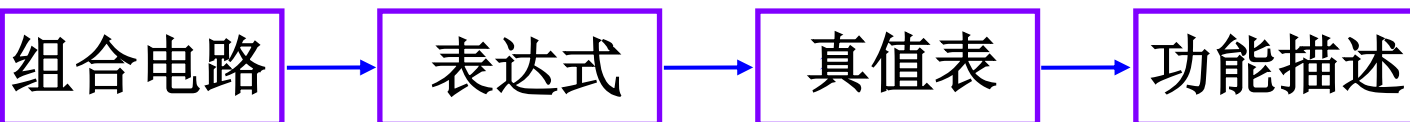
3.1 SSI组合逻辑电路的分析

(1) 分析步骤

(a) 从输入端开始，逐级推导函数表达式，再变换、化简

(b) 列真值表

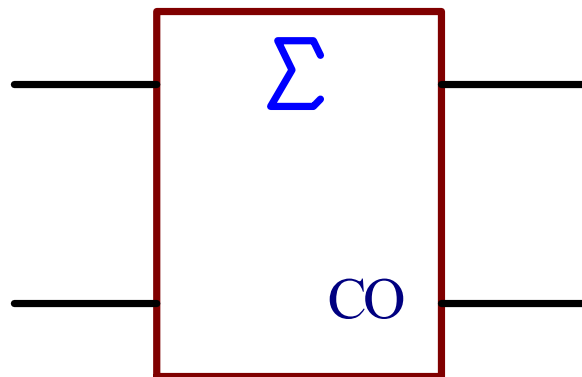
(c) 确定逻辑功能



主题三 组合逻辑电路

3.1 常用的SSI组合逻辑电路

(1) 半加器

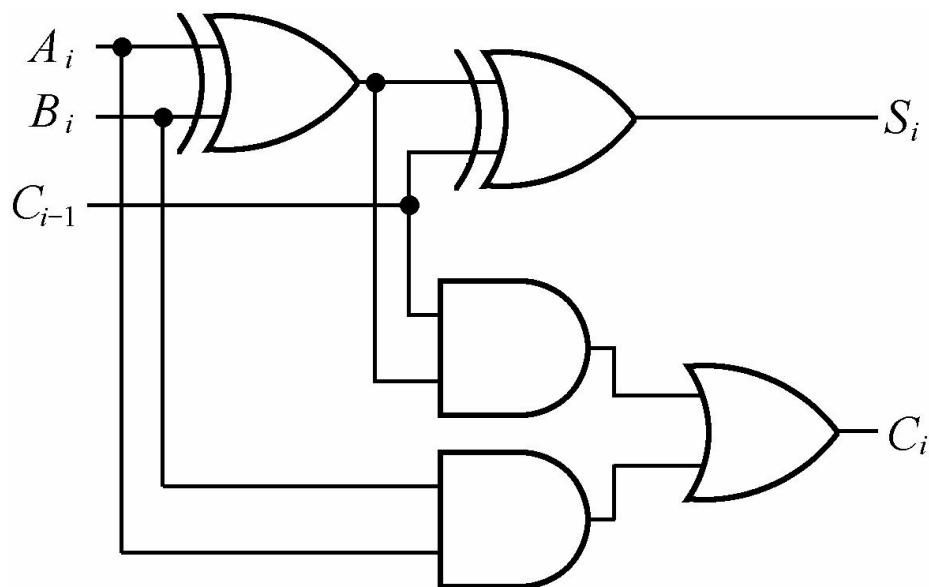


输入		输出	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

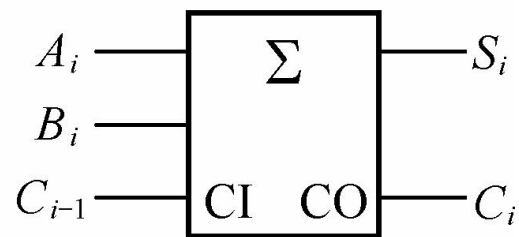
主题三 组合逻辑电路

3.1 常用的SSI组合逻辑电路

(2) 全加器



(a) 全加器电路

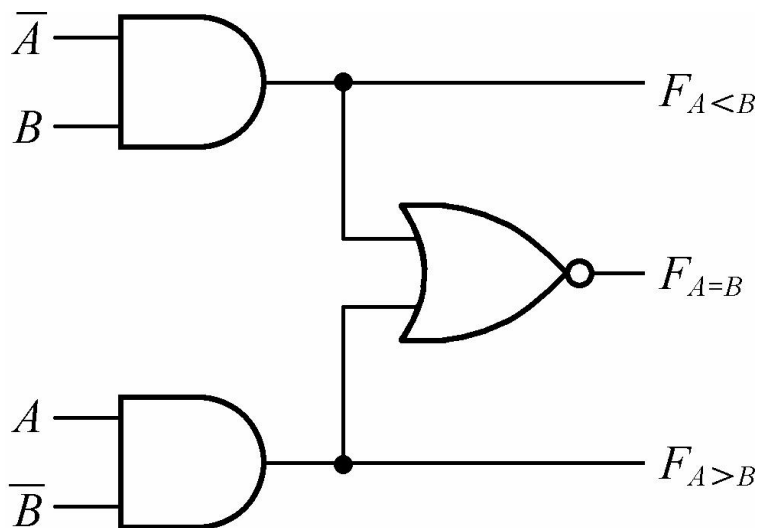


(b) 逻辑符号

主题三 组合逻辑电路

3.1 常用的SSI组合逻辑电路

(3) 1位比较器



输 入		输 出		
A	B	$F_{A < B}$	$F_{A = B}$	$F_{A > B}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

主题三 组合逻辑电路

3.2 SSI组合逻辑电路的设计

(1) 设计步骤

(a)列真值表;

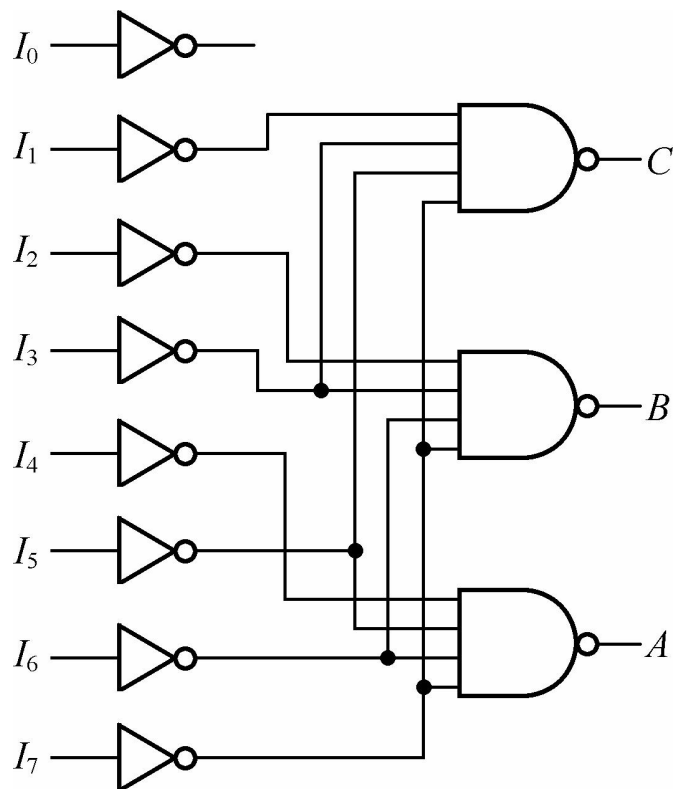
(b)写最简表达式;

(c)画逻辑电路

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(1) 8—3普通编码器

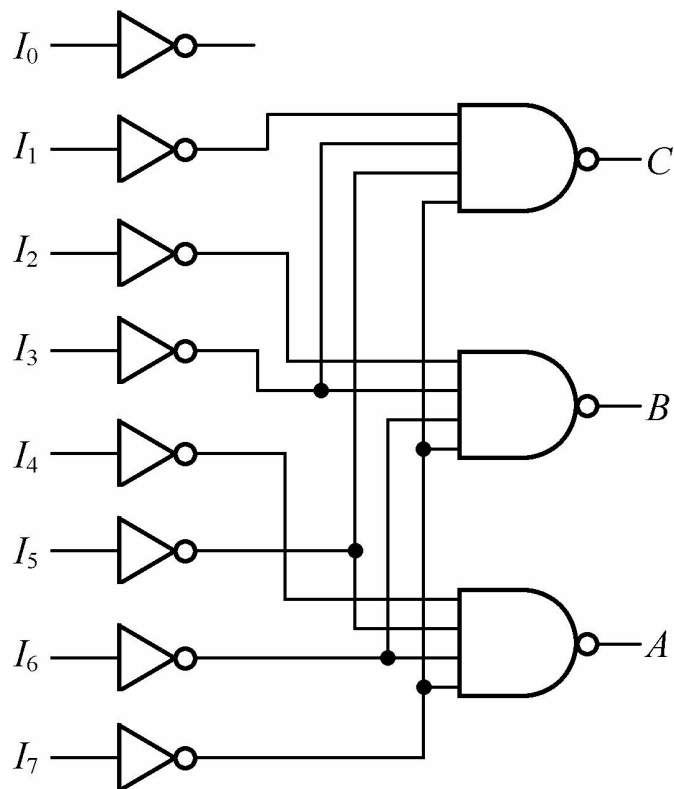


输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(1) 8—3普通编码器

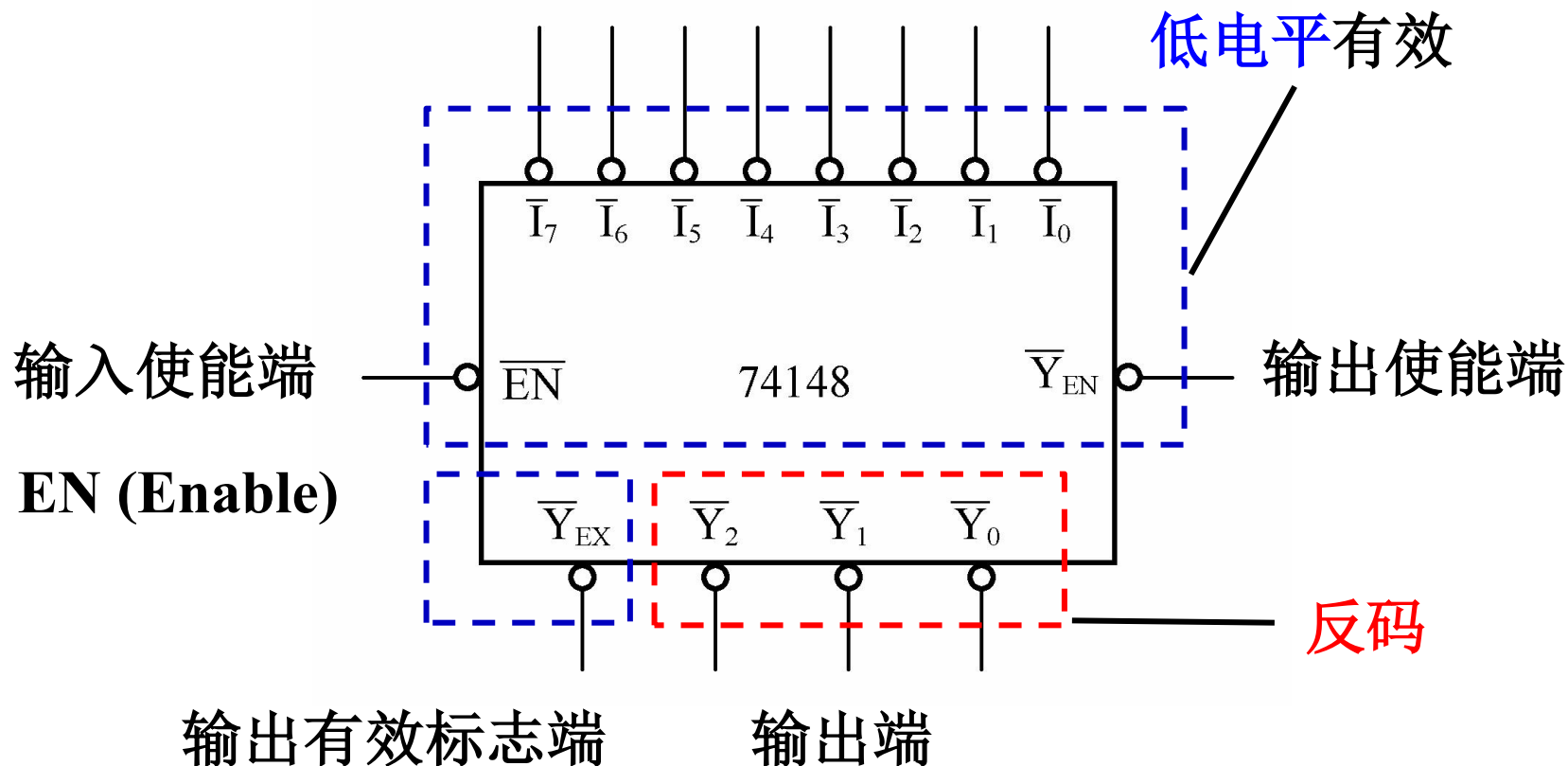


输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(2) 8—3优先编码器74148



主题三 组合逻辑电路

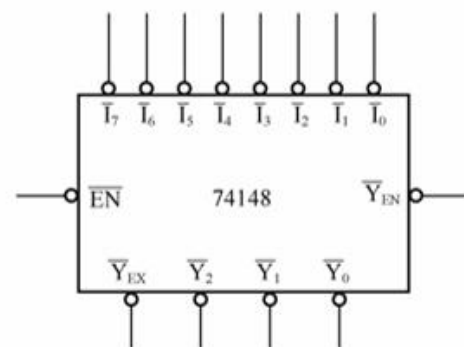
3.3 常用的MSI组合逻辑电路模块

(2) 8—3优先编码器74148

使能输入	输 入	输 出	输出标志	使能输出
\overline{EN}	$\overline{I}_7 \ \overline{I}_6 \ \overline{I}_5 \ \overline{I}_4 \ \overline{I}_3 \ \overline{I}_2 \ \overline{I}_1 \ \overline{I}_0$	$\overline{Y}_2 \ \overline{Y}_1 \ \overline{Y}_0$	\overline{Y}_{EX}	\overline{Y}_{EN}
1	0 0 0 0 0 0 0 0	1 1 1	1	1
0	1 1 1 1 1 1 1 1	1 1 1	1	0
0	0 0 0 0 0 0 0 0	0 0 0	0	1
0	1 0 0 0 0 0 0 0	0 0 1	0	1
0	1 1 0 0 0 0 0 0	0 1 0	0	1
0	1 1 1 0 0 0 0 0	0 1 1	0	1
0	1 1 1 1 0 0 0 0	1 0 0	0	1
0	1 1 1 1 1 0 0 0	1 0 1	0	1
0	1 1 1 1 1 1 0 0	1 1 0	0	1
0	1 1 1 1 1 1 1 0	1 1 1	0	1

● 输入端低电平
(即逻辑“0”)有效

● 输入端十进制下
标的自然二进制码
的反码



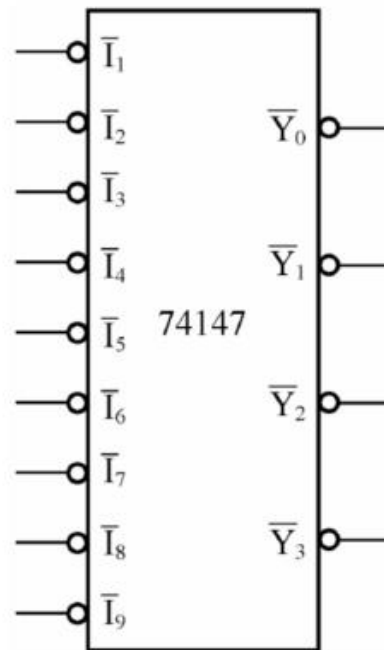
主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(3) 二—十进制优先编码器74147

输 入	输 出
$\bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 \bar{I}_6 \bar{I}_7 \bar{I}_8 \bar{I}_9$	$\bar{Y}_3 \bar{Y}_2 \bar{Y}_1 \bar{Y}_0$
0 1 1 1 1 1 1 1 1	1 1 1 0
0 0 1 1 1 1 1 1 1	1 1 0 1
0 0 0 1 1 1 1 1 1	1 1 0 0
0 0 0 0 1 1 1 1 1	1 0 1 1
0 0 0 0 0 1 1 1 1	1 0 1 0
0 0 0 0 0 0 1 1 1	1 0 0 1
0 0 0 0 0 0 0 1 1	1 0 0 0
0 0 0 0 0 0 0 0 1	0 1 1 1
0 0 0 0 0 0 0 0 0	0 1 1 0
1 1 1 1 1 1 1 1 1	1 1 1 1

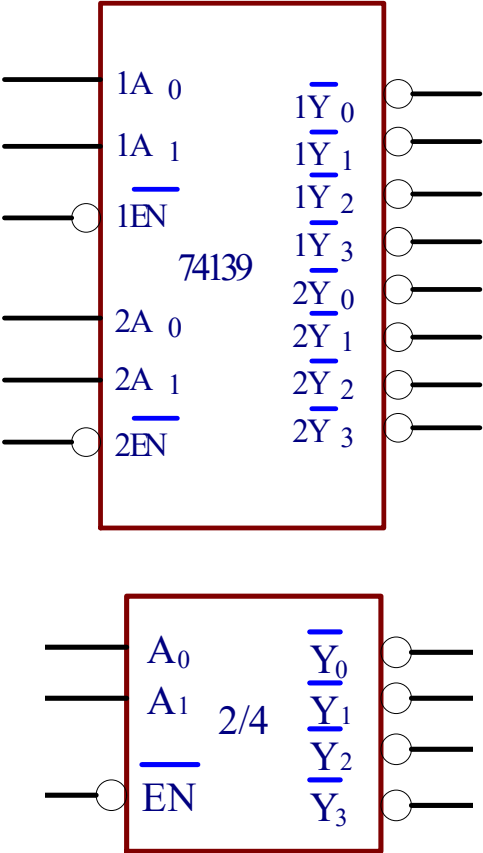
- 输入端低电平（即逻辑“0”）有效
- 输入端十进制下标的8421BCD码的反码



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(4) 双2-4线译码器74139

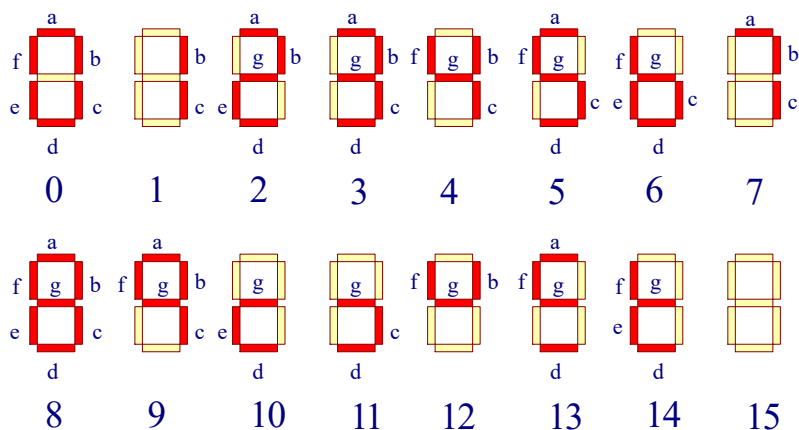
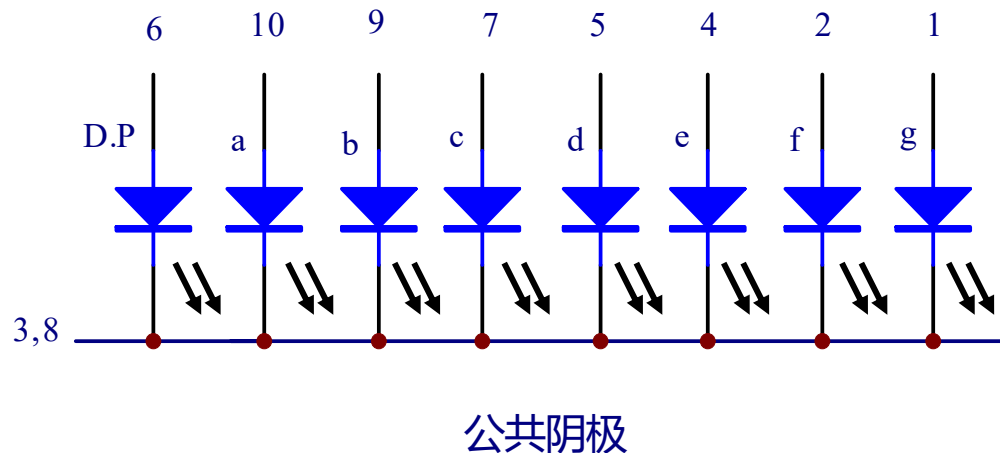
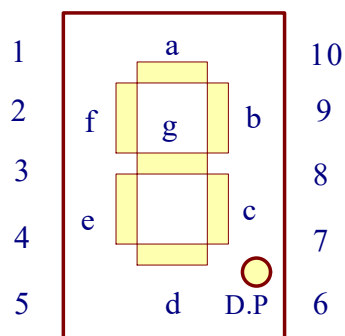
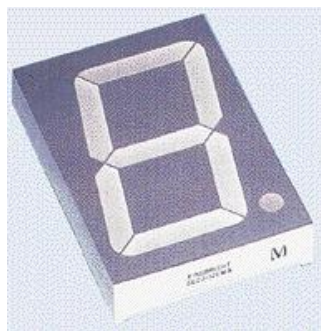


使能 输入	输 入		输 出			
\overline{EN}	A_1	A_0	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3
1	\emptyset	\emptyset	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

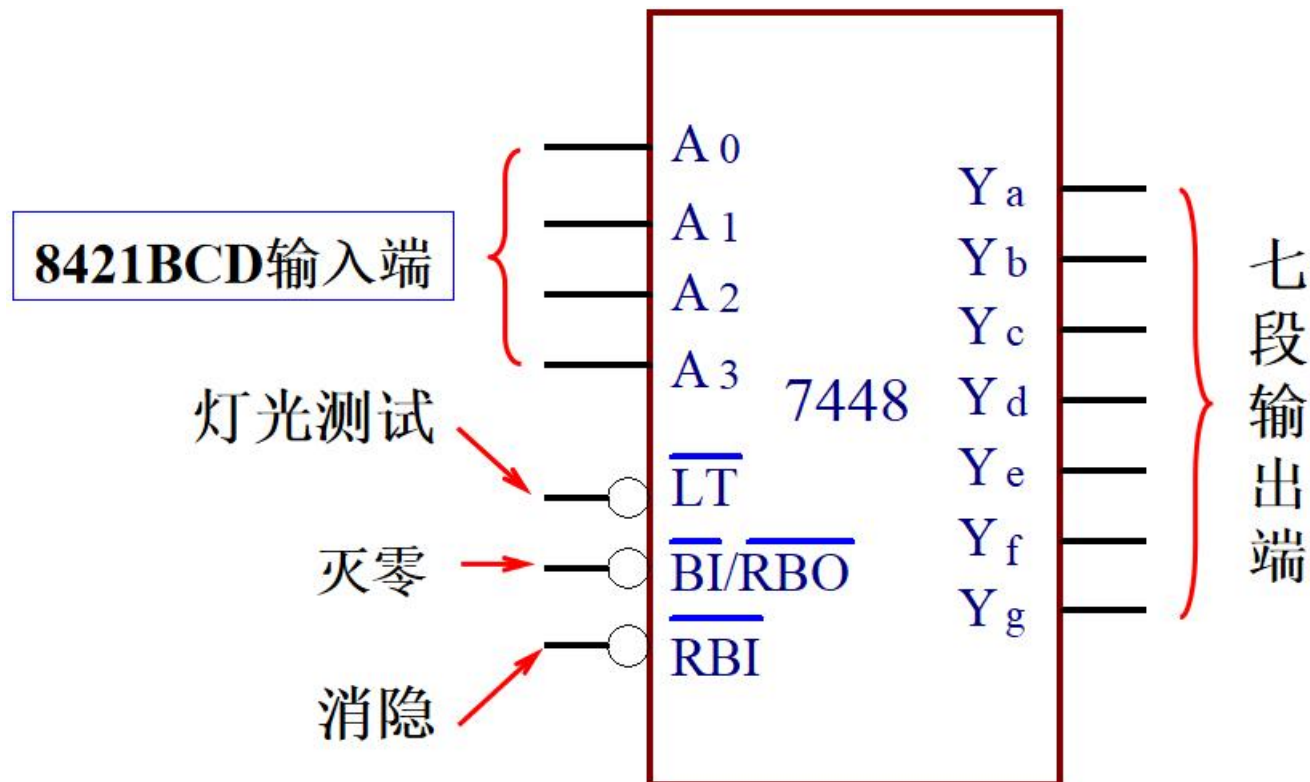
(5) 七段数码管



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(6) 数字显示译码器7448



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(6) 数字显示译码器7448

	十进制数功能或能														输入				\overline{BI} / \overline{RBO}	输出							说明 译 码 显 示
	\overline{LT}	\overline{RBI}	A_3	A_2	A_1	A_0	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g														
	0	1	1	0	0	0	0	1	1	1	1	1	1	0													
	1	1	\emptyset	0	0	0	1	1	0	1	1	0	0	0													
	2	1	\emptyset	0	0	1	0	1	1	0	1	1	0	1													
	3	1	\emptyset	0	0	1	1	1	1	1	0	0	0	1													
	4	1	\emptyset	0	1	0	0	1	0	1	1	0	0	1													
	5	1	\emptyset	0	1	0	1	1	0	1	1	0	1	1													
	6	1	\emptyset	0	1	1	0	1	0	0	1	1	1	1													
	7	1	\emptyset	0	1	1	1	1	1	1	0	0	0	0													
8	1	\emptyset	1	0	0	0	1	1	1	1	1	1	1														
9	1	\emptyset	1	0	0	1	1	1	1	0	0	1	1														

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

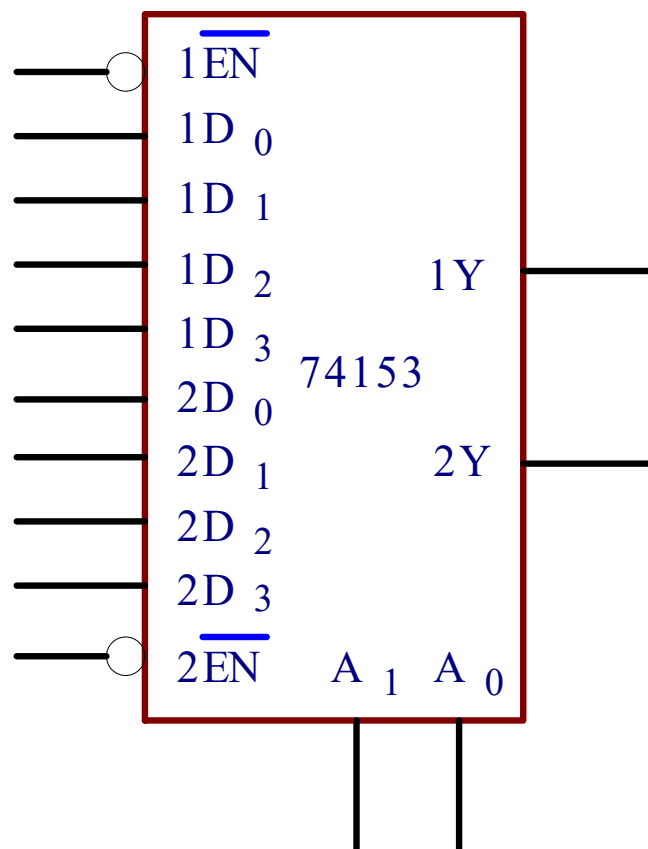
(6) 数字显示译码器7448

十进制 数 或 功 能	输 入			$\overline{BI} / \overline{RBO}$	输 出							说明
	\overline{LT}	\overline{RBI}	$A_3 A_2 A_1 A_0$		$Y_a Y_b Y_c Y_d Y_e Y_f Y_g$							
10	1	\emptyset	1 0 1 0	1	0 0 0 1 1 0 1	译 码 显 示						
11	1	\emptyset	1 0 1 1	1	0 0 1 1 0 0 1							
12	1	\emptyset	1 1 0 0	1	0 1 0 0 0 1 1							
13	1	\emptyset	1 1 0 1	1	1 0 0 1 0 1 1							
14	1	\emptyset	1 1 1 0	1	0 0 0 1 1 1 1							
15	1	\emptyset	1 1 1 1	1	0 0 0 0 0 0 0							
$\overline{BI}=0$	\emptyset	\emptyset	$\emptyset \ \emptyset \ \emptyset \ \emptyset$	0	0 0 0 0 0 0 0	熄灭						
$\overline{LT}=0$	0	\emptyset	$\emptyset \ \emptyset \ \emptyset \ \emptyset$	1	1 1 1 1 1 1 1	测试						
$\overline{RBI}=0$	1	0	0 0 0 0	0	0 0 0 0 0 0 0	灭零						

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

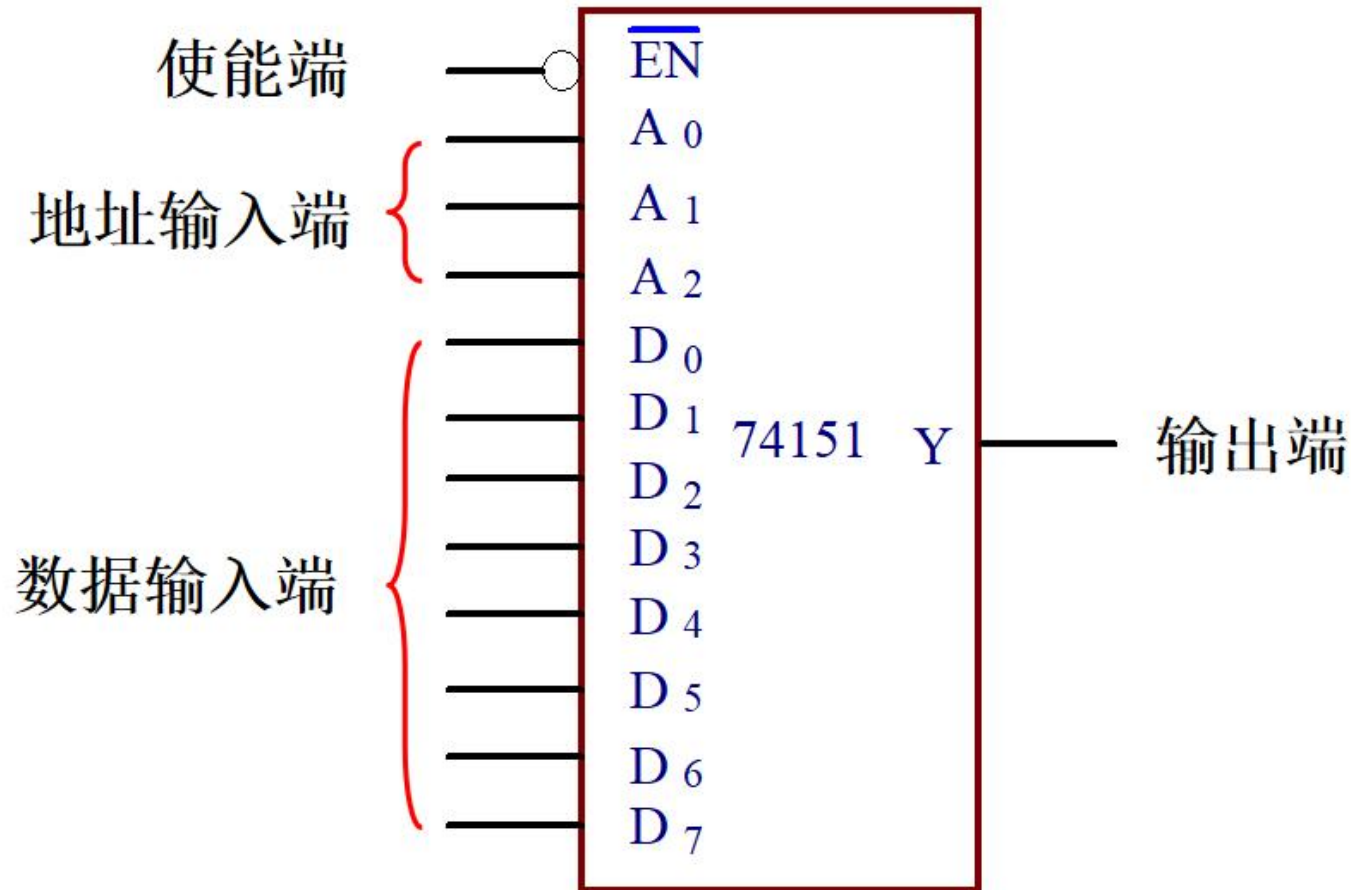
(7) 双四选一数据选择器74153



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(8) 八选一数据选择器74151



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(8) 八选一数据选择器74151

使能输入	输 入	输 出
\overline{EN}	$A_2 A_1 A_0$	Y
1	$\emptyset \ \emptyset \ \emptyset$	0
0	0 0 0	D_0
0	0 0 1	D_1
0	0 1 0	D_2
0	0 1 1	D_3
0	1 0 0	D_4
0	1 0 1	D_5
0	1 1 0	D_6
0	1 1 1	D_7

$A_1 A_0$					
A_2		00	01	11	10
0		D_0	D_1	D_3	D_2
1		D_4	D_5	D_7	D_6

$$\overline{EN} = 1 \quad Y = 0$$

$$\overline{EN} = 0 \quad Y = \sum_{i=0}^7 m_i D_i$$

m_i 为 A_2, A_1, A_0 构成的最小项

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(9) 利用数据选择器实现任意逻辑函数（卡诺图降维）

例 分别用一片74151和½ 74153

$$\text{实现 } F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{D} + A\bar{B}D + AC\bar{D}$$

解：这是一个四变量函数，对其一次降维后可用74151实现，两次降维后可用½ 74153实现。

(1) 一次降维，用74151实现

AB	CD			
	00	01	11	10
00	1	1		
01			1	1
11	1			1
10		1	1	1

降1维

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

A ₂	A ₁ A ₀			
	00	01	11	10
0	D ₀	D ₁	D ₃	D ₂
1	D ₄	D ₅	D ₇	D ₆

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(9) 利用数据选择器实现任意逻辑函数（卡诺图降维）

例 分别用一片74151和½ 74153

$$\text{实现 } F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{D} + A\bar{B}D + ACD\bar{D}$$

解：这是一个四变量函数，对其一次降维后可用7415实现，两次降维后可用½ 74153实现。

(2) 两次降维，用74153实现

CD		AB			
		00	01	11	10
AB	00	1	1		
	01			1	1
	11	1			1
	10		1	1	1



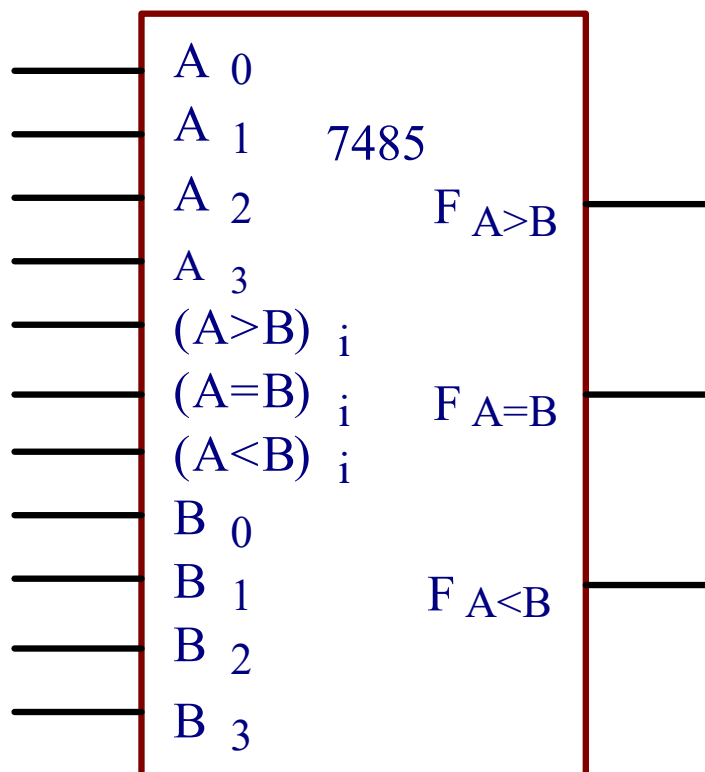
B		A	
		0	1
A	0	\bar{C}	C
	1	$C+D$	\bar{D}

A ₁		A ₀	
		0	1
A ₁	0	D_0	D_1
	1	D_2	D_3

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(10) 数据比较器 7485



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(10) 数据比较器 7485

输 入				输 出		
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$(A>B)_i$ $(A<B)_i$ $(A=B)_i$	$F_{A>B}$	$F_{A<B}$ $F_{A=B}$
$A_3>B_3$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \quad \emptyset \quad \emptyset$	1	0 0
$A_3<B_3$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \quad \emptyset \quad \emptyset$	0	1 0
$A_3=B_3$	$A_2>B_2$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \quad \emptyset \quad \emptyset$	1	0 0
$A_3=B_3$	$A_2<B_2$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \quad \emptyset \quad \emptyset$	0	1 0
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	$\emptyset \emptyset$	$\emptyset \quad \emptyset \quad \emptyset$	1	0 0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	$\emptyset \emptyset$	$\emptyset \quad \emptyset \quad \emptyset$	0	1 0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	$\emptyset \quad \emptyset \quad \emptyset$	1	0 0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	$\emptyset \quad \emptyset \quad \emptyset$	0	1 0

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

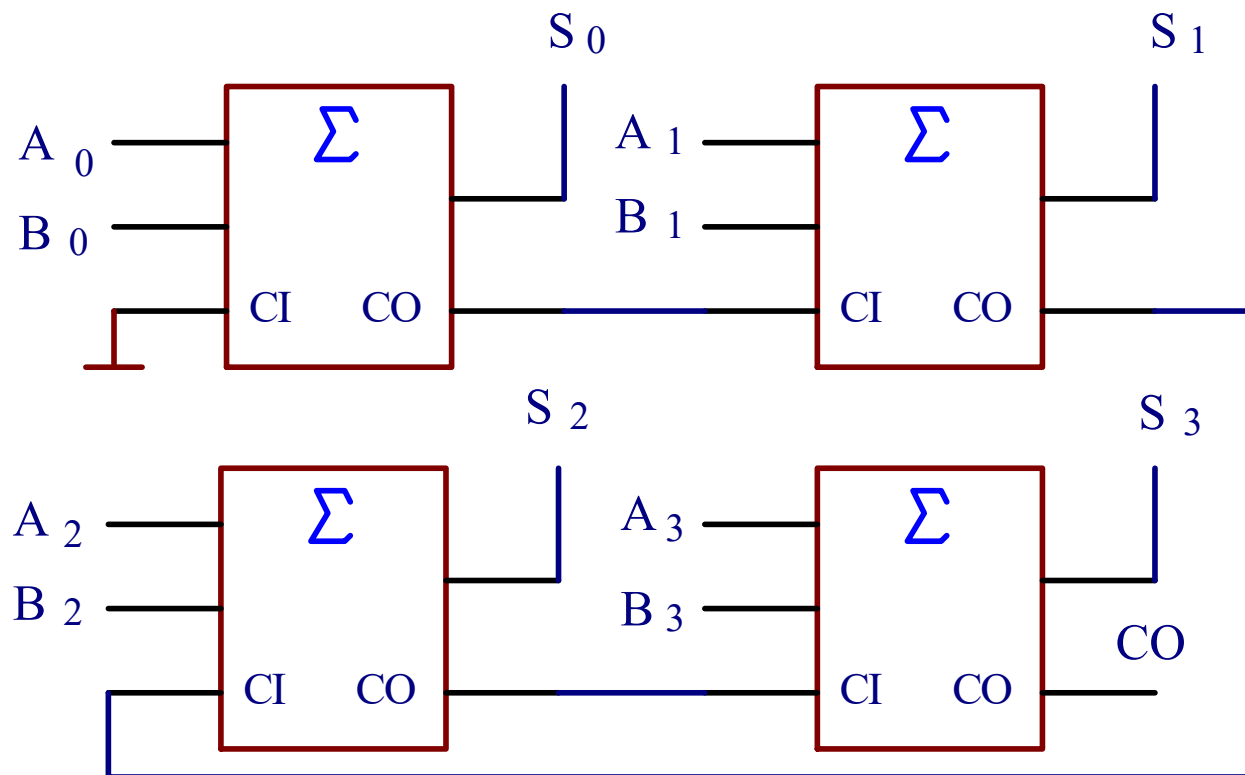
(10) 数据比较器 7485

输 入				输 出		
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$(A>B)_i$	$(A<B)_i$	$(A=B)_i$
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	1	1

主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

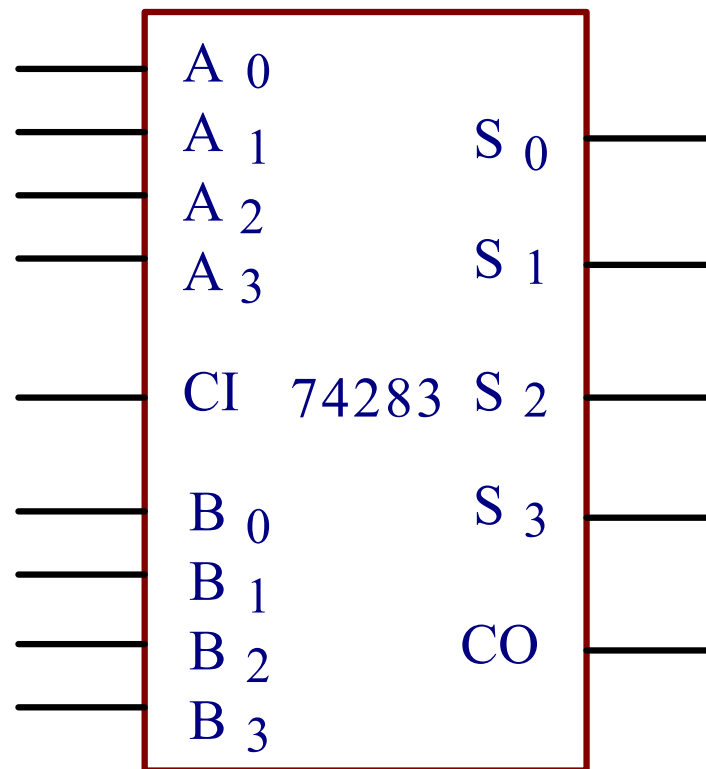
(11) 四位串行进位全加器



主题三 组合逻辑电路

3.3 常用的MSI组合逻辑电路模块

(12) 四位超前进位全加器



3.4 MSI组合逻辑电路的分析与设计

主题三 组合逻辑电路

3.5 组合逻辑电路的竞争与冒险

(1) 竞争的概念

(2) 冒险的概念——0型、1型

(3) 判别

$$F = A + \bar{A} \quad (\text{对应0型逻辑冒险的判别式})$$

$$F = A \cdot \bar{A} \quad (\text{对应1型逻辑冒险的判别式})$$

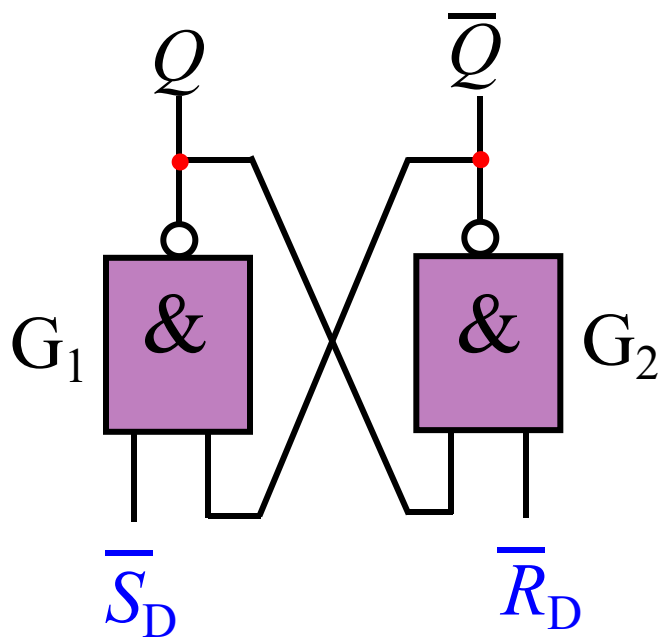
卡诺图相切

(4) 消除

(a) 增加冗余项 (b) 加滤波电容 (c) 加取样脉冲

主题四 触发器

4.1 基本RS触发器



(a) 逻辑电路

\bar{S}_D 、 \bar{R}_D : 输入端

直接置1(或0)端

直接置位(或复位)端

控制输入端

数据输入端

激励输入端

触发输入端

主题四 触发器

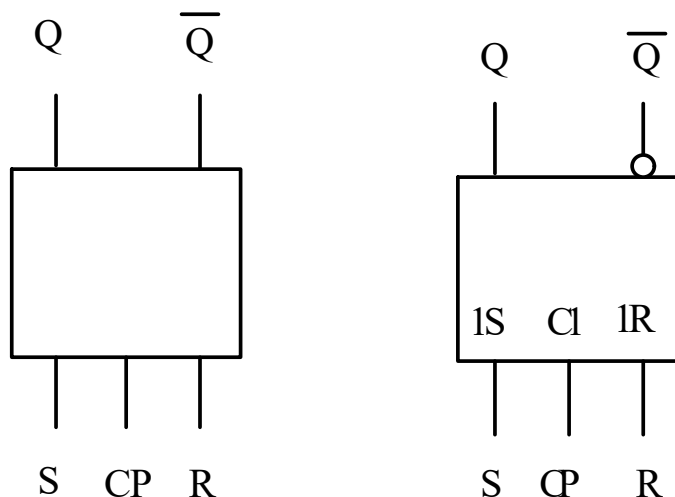
4.1 基本RS触发器

S_D	R_D	Q^n	Q^{n+1}	功能
0	0	0	1*	禁止
0	0	1	1*	禁止
0	1	0	1	置1
0	1	1	1	置1
1	0	0	0	置0
1	0	1	0	置0
1	1	0	0	保持
1	1	1	1	保持

$$\begin{cases} Q^{n+1} = \overline{\overline{S_D}} + \overline{R_D} Q^n \\ \overline{S_D} + \overline{R_D} = 1 \quad (\text{约束条件}) \end{cases}$$

主题四 触发器

4.2 钟控电平SR触发器

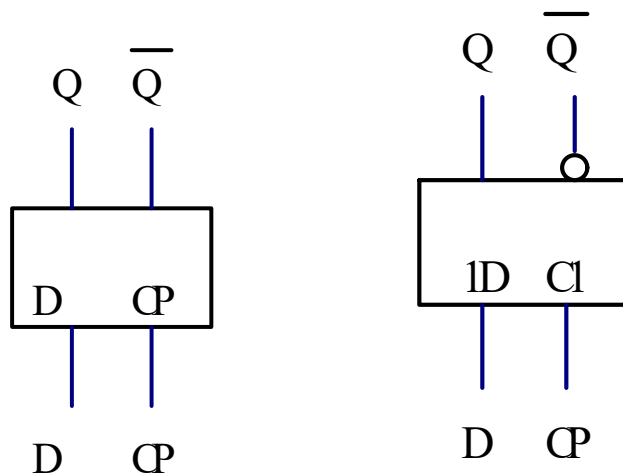


$$CP=0 \text{ 时, } Q^{n+1}=Q^n$$

$$CP=1 \text{ 时, } \begin{cases} Q^{n+1}=S+\overline{R}Q^n \\ SR=0 \end{cases}$$

主题四 触发器

4.3 钟控电平D触发器

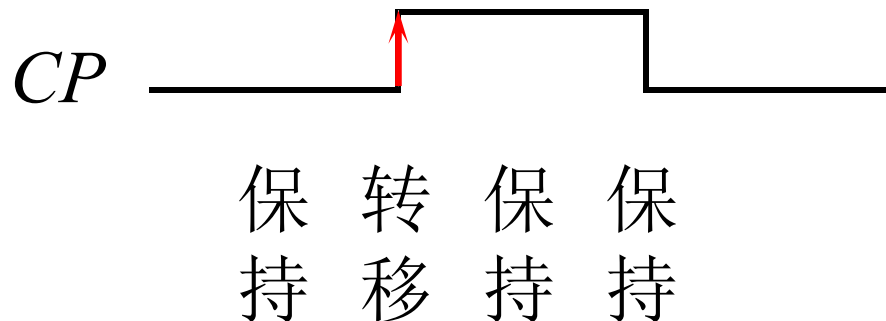
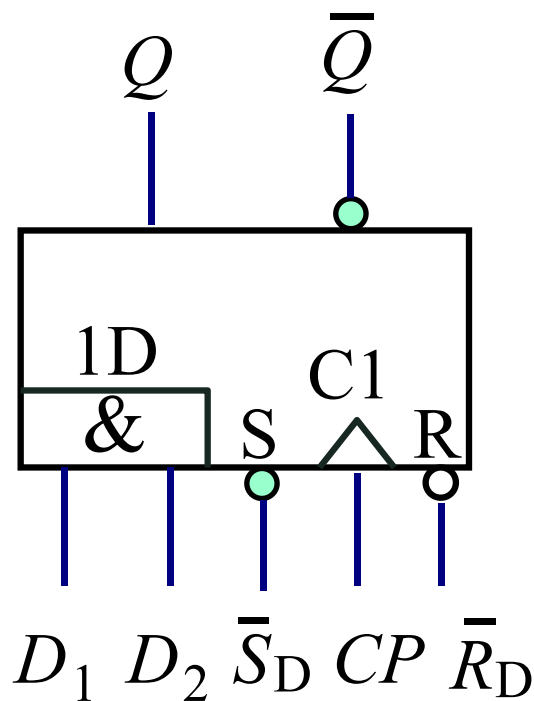


$$CP=0\text{时}, Q^{n+1}=Q^n$$

$$CP=1\text{时}, Q^{n+1}=D$$

主题四 触发器

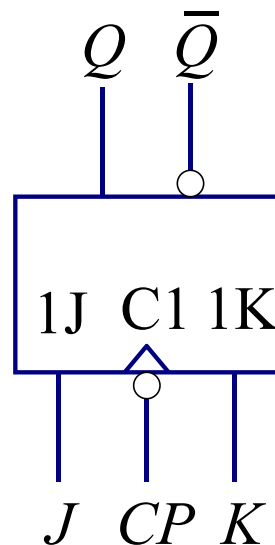
4.4 钟控边沿D触发器



次态方程: $Q^{n+1} = [D] \cdot CP \uparrow$

主题四 触发器

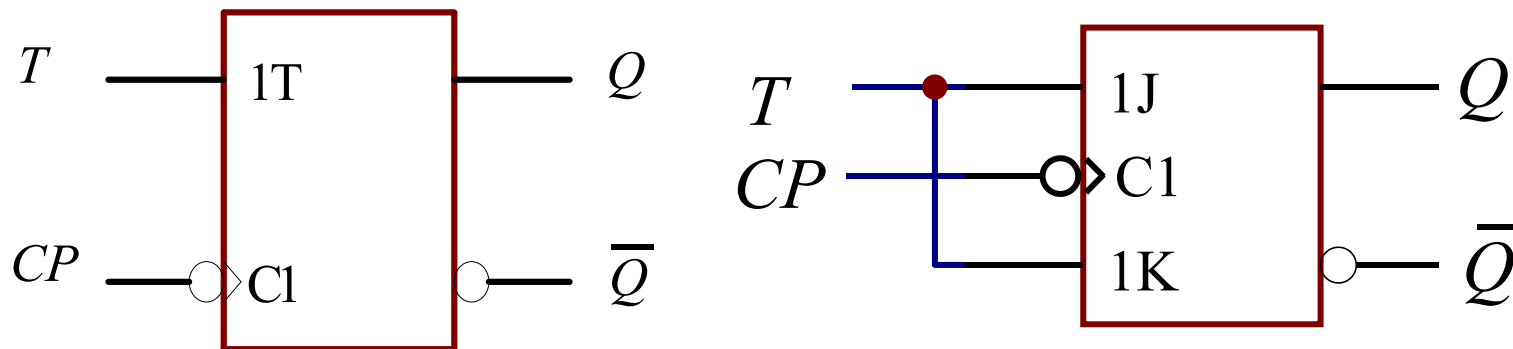
4.5 钟控边沿JK触发器



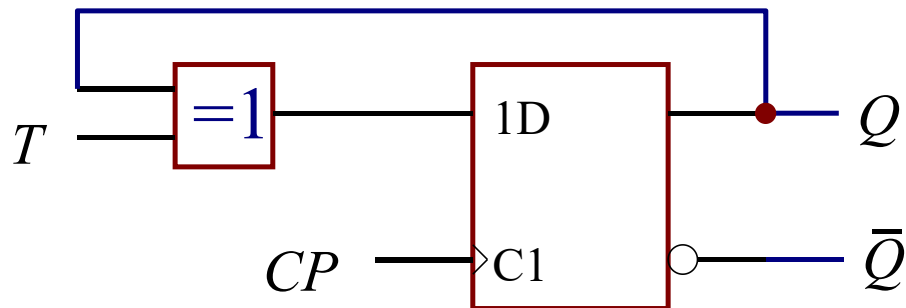
$$Q^{n+1} = [J\bar{Q}^n + \bar{K}Q^n] \cdot CP \downarrow$$

主题四 触发器

4.6 钟控边沿T触发器



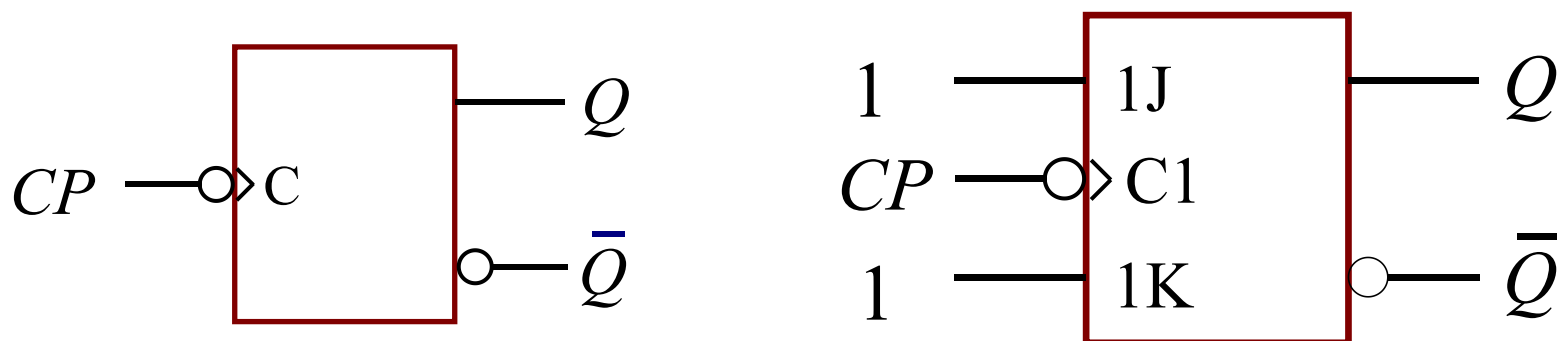
$$Q^{n+1} = [T\bar{Q}^n + \bar{T}Q^n] \cdot CP \downarrow$$



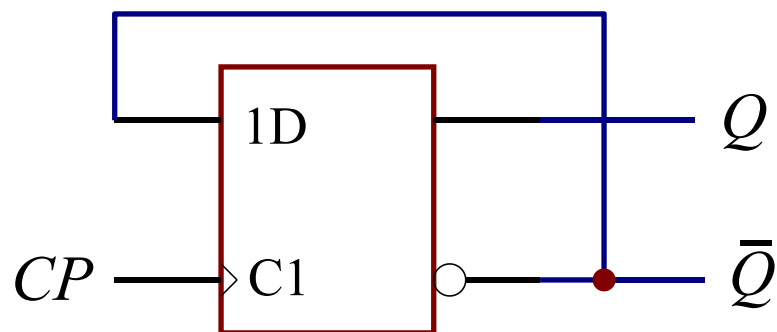
$$Q^{n+1} = [T\bar{Q}^n + \bar{T}Q^n] \cdot CP \uparrow$$

主题四 触发器

4.7 钟控边沿T' 触发器



$$Q^{n+1} = [\bar{Q}^n] \cdot CP \downarrow$$



$$Q^{n+1} = [\bar{Q}^n] \cdot CP \uparrow$$

主题五 时序逻辑电路

5.1 时序逻辑电路分析



(1) 激励方程 $W = F[X, Q]$

(2) 状态方程 $Q^{n+1} = G[W, Q^n]$

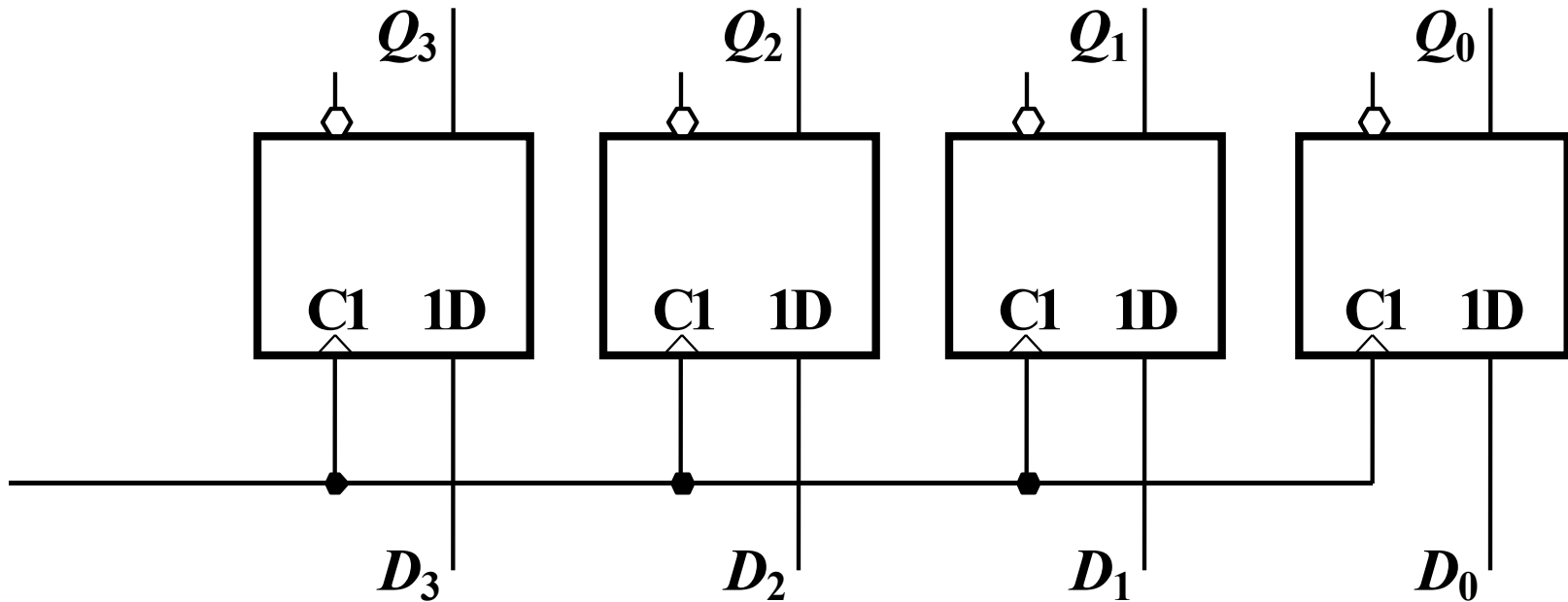
(3) 输出方程 $Z = H[X, Q]$

(4) 时钟方程

主题五 时序逻辑电路

5.2 寄存器与移存器

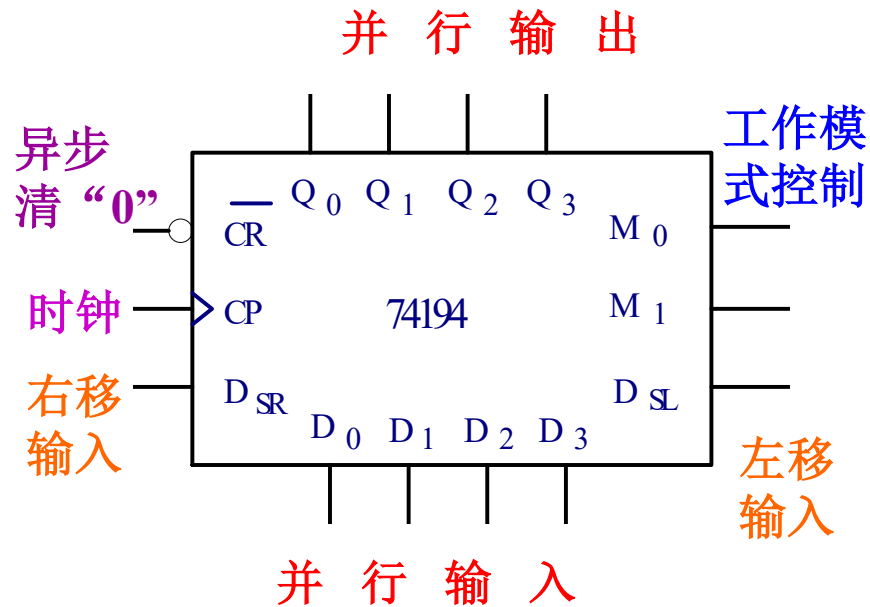
(1) 级联型寄存器



主题五 时序逻辑电路

5.2 寄存器与移存器

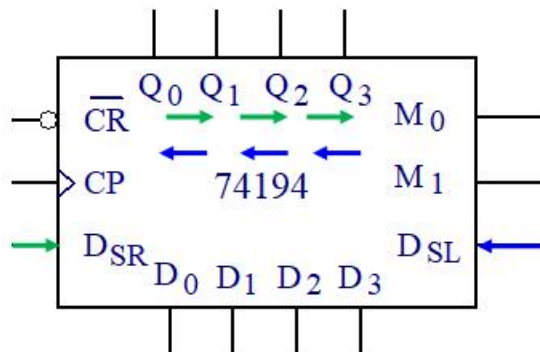
(2) MSI移存器74194



主题五 时序逻辑电路

5.2 寄存器与移存器

(2) MSI 移存器74194



$$\begin{array}{cccccc}
 & & Q_0 & Q_1 & Q_2 & Q_3 \\
 t_n & D_{SR} & Q_0^n & Q_1^n & Q_2^n & Q_3^n \\
 t_{n+1} & D_{SR} & Q_0^n & Q_1^n & Q_2^n & Q_3^n
 \end{array}$$

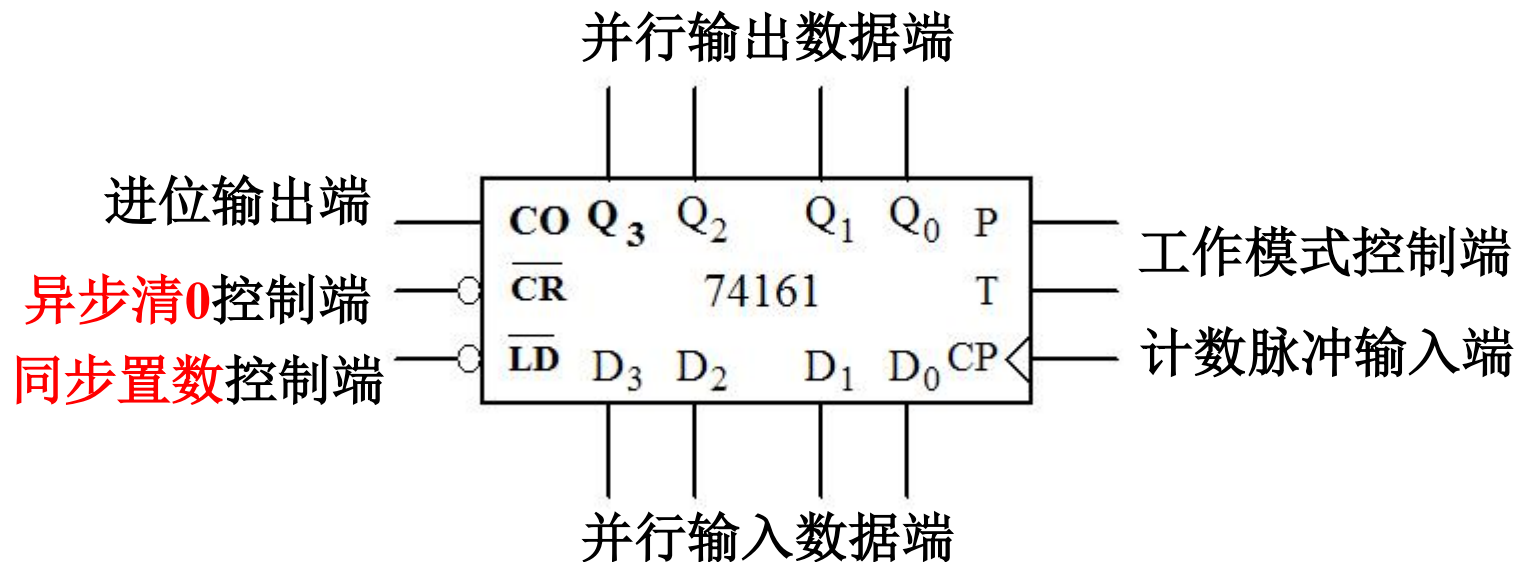
功能	\overline{CR}	M_0	M_1	CP	D_{SR}	D_{SL}	D_0	D_1	D_2	D_3	Q_0^{n+1}	Q_1^{n+1}	Q_2^{n+1}	Q_3^{n+1}
清除	0	×	×	×	×	×	×	×	×	×	0	0	0	0
并入	1	1	1	↑	×	×	d_0	d_1	d_2	d_3	d_0	d_1	d_2	d_3
保持	1	×	×	0	×	×	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n
	1	0	0	×	×	×	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n
右移	1	1	0	↑	1	×	×	×	×	×	1	Q_0^n	Q_1^n	Q_2^n
	1	1	0	↑	0	×	×	×	×	×	0	Q_0^n	Q_1^n	Q_2^n
左移	1	0	1	↑	×	1	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	1
	1	0	1	↑	×	0	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	0



主题五 时序逻辑电路

5.3 计数器

(2) MSI 同步计数器74161



主题五 时序逻辑电路

5.3 计数器

(2) MSI 同步模16加法计数器74161

\overline{CR}	\overline{LD}	P(S ₁)	T(S ₂)	CP	D ₃ D ₂ D ₁ D ₀	Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}	功能
0	0	0	0	0	0 0 0 0	0 0 0 0	异步清除
1	0	0	0	↑	d ₃ d ₂ d ₁ d ₀	d ₃ d ₂ d ₁ d ₀	同步并入
1	1	1	1	↑	0 0 0 0	0000 ~ 1111	计数
1	1	0	1	0	0 0 0 0	Q_3^n Q_2^n Q_1^n Q_0^n CO ⁿ	保持
1	1	0	0	0	0 0 0 0	Q_3^n Q_2^n Q_1^n Q_0^n CO = 0	

$$CO = Q_3 Q_2 Q_1 Q_0 T$$

主题五 时序逻辑电路

5.3 计数器

(3) MSI 同步模16加法计数器74163

\overline{CR}	\overline{LD}	P(S ₁)	T(S ₂)	CP	D ₃ D ₂ D ₁ D ₀	Q_3^{n+1} Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}	功能
0	0	0	0	↑	0 0 0 0	0 0 0 0	同步清除
1	0	0	0	↑	d ₃ d ₂ d ₁ d ₀	d ₃ d ₂ d ₁ d ₀	同步并入
1	1	1	1	↑	0 0 0 0	0000 ~ 1111	计数
1	1	0	1	0	0 0 0 0	Q_3^n Q_2^n Q_1^n Q_0^n CO ⁿ	保持
1	1	0	0	0	0 0 0 0	Q_3^n Q_2^n Q_1^n Q_0^n CO = 0	

$$CO = Q_3 Q_2 Q_1 Q_0 T$$

主题五 时序逻辑电路

5.3 计数器

(4) MSI同步模10加法计数器74160

\overline{CR}	\overline{LD}	P	T	CP	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	功 能
0	1	Ø	Ø	Ø	0	0	0	0	异步清0
1	0	Ø	Ø	↑	d_3	d_2	d_1	d_0	同步并入
1	1	1	1	↑	0 0 0 0 ~ 1 0 0 1				8421BCD计数
1	1	0	1	Ø	Q_3^n	Q_2^n	Q_1^n	Q_0^n	保持 CO^n
1	1	Ø	0	Ø	Q_3^n	Q_2^n	Q_1^n	Q_0^n	保持 $CO = 0$

74LS160 模10, $CO = Q_3 Q_0 T$

主题五 时序逻辑电路

5.3 计数器

(5) 三类MSI同步加法计数器比较

- 74161 模16 (同步计数、异步清0)

- 74163 模16 (同步计数、同步清0)

进位 $CO = Q_3Q_2Q_1Q_0 \cdot T$ (1111时产生进位)

- 74160 模10 (同步计数、异步清0)

进位 $CO = Q_3Q_0 \cdot T$ (1001时产生进位)

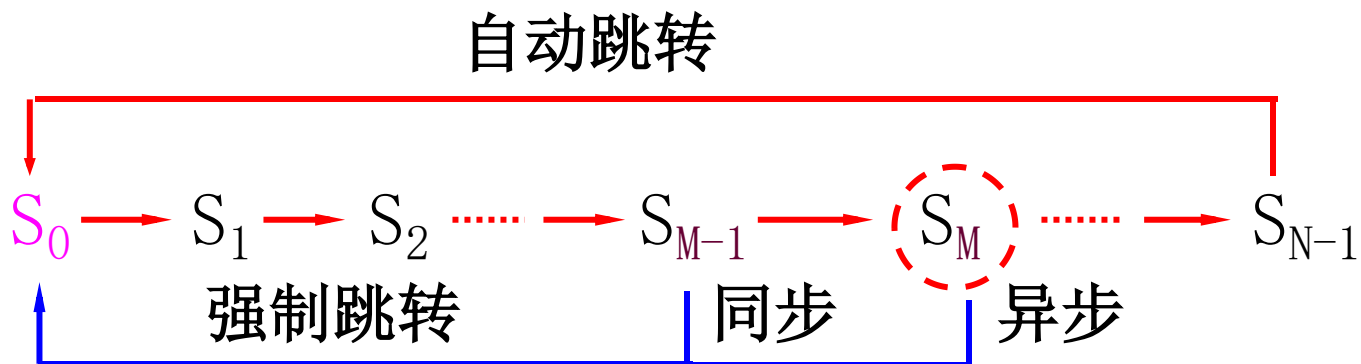
主题五 时序逻辑电路

5.3 计数器

(6) 同步级联（控制C0）、异步级联（控制CLK）

(7) 任意 $M < 16$ 计数器设计

a. 清零法（利用复位端）



涉及清零、计数工作模式

主题五 时序逻辑电路

5.3 计数器

(6) 同步级联（控制C0）、异步级联（控制CLK）

(7) 任意 $M < 16$ 计数器设计

b. 置数法（利用置数控制端，并行输入端）



关键：控制端同步/异步，起跳状态选择，
反馈引导门的设计

主题五 时序逻辑电路

5.3 计数器

(6) 同步级联（控制C0）、异步级联（控制CLK）

(7) 任意 $M < 16$ 计数器设计

b. 置数法（利用置数控制端，并行输入端）



关键：控制端同步/异步，起跳状态选择，
反馈引导门的设计
⋮

(8) 移存型计数器——环形、扭环形

主题五 时序逻辑电路

5.4 序列码发生器

(1) 级联触发器+反馈门电路

STEP1. 求触发器的级数

STEP2. 划线法确定状态（移动方向与划线方向相反）

STEP3. 检查状态表循环模态内有无重复状态

STEP3a. 有，则触发器级数+1，重做STEP2

STEP3b. 无，做STEP4

STEP4. 根据状态表生成反馈门电路的真值表

STEP5. 利用卡诺图化简并求取反馈函数

STEP6. 得到最终设计电路

主题五 时序逻辑电路

5.4 序列码发生器

(2) 级联触发器+数据选择器

STEP1. 求触发器的级数

STEP2. 划线法确定状态（移动方向与划线方向相反）

STEP3. 检查状态表循环模态内有无重复状态

STEP3a. 有，则触发器级数+1，重做STEP2

STEP3b. 无，做STEP4

STEP4. 根据状态表生成反馈逻辑的真值表

STEP5. 检查MUX地址线是否大于等于触发器状态数

STEP5a. 大于等于，无需降维

STEP5b. 小于，卡诺图降维，多余状态踢至D端

主题五 时序逻辑电路

5.4 序列码发生器

(3) 计数器+数据选择器

STEP1. 求生成序列模长M

STEP2. 设计模M计数器

STEP3. 根据计数状态表生成反馈逻辑的真值表

STEP4. 检查MUX地址线是否大于等于触发器状态数

STEP4a. 大于等于，无需降维

STEP4b. 小于，卡诺图降维，多余状态踢至D端

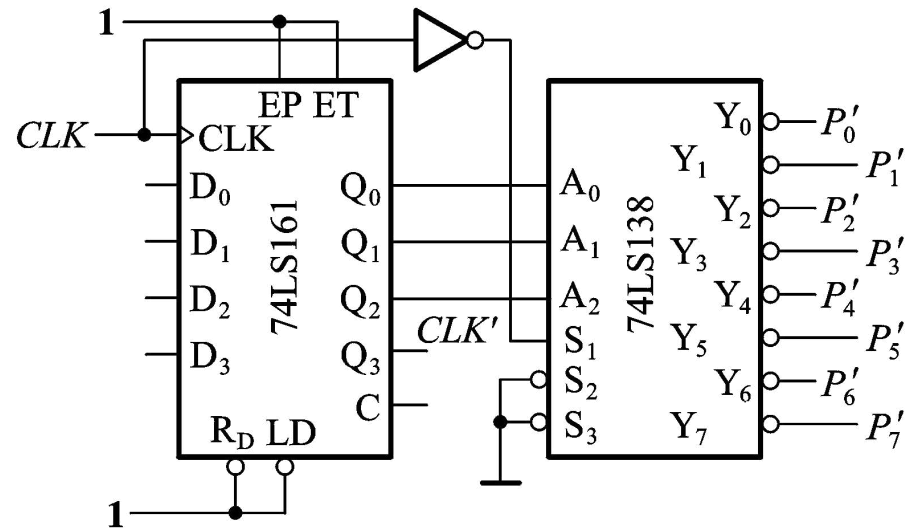
(4) 其他方案

MSI移存器+门电路/数据选择器。。。

主题五 时序逻辑电路

5.5 顺序脉冲发生器

- (1) 环形计数器模式
- (2) 扭环形计数器模式
- (3) 计数器+译码器模式



5.6 分频器、节拍器。。。。

主题五 时序逻辑电路

5.7 一般时序电路分析

1. 分析电路结构

- (1) 组合电路、存储电路
- (2) 输入信号 X 、输出信号 Z
- (3) 电路类型（同步、异步；Mealy、Moore）

2. 写出四组方程

- (1) 时钟方程
- (2) 各触发器的激励方程
- (3) 各触发器的次态方程
- (4) 电路的输出方程

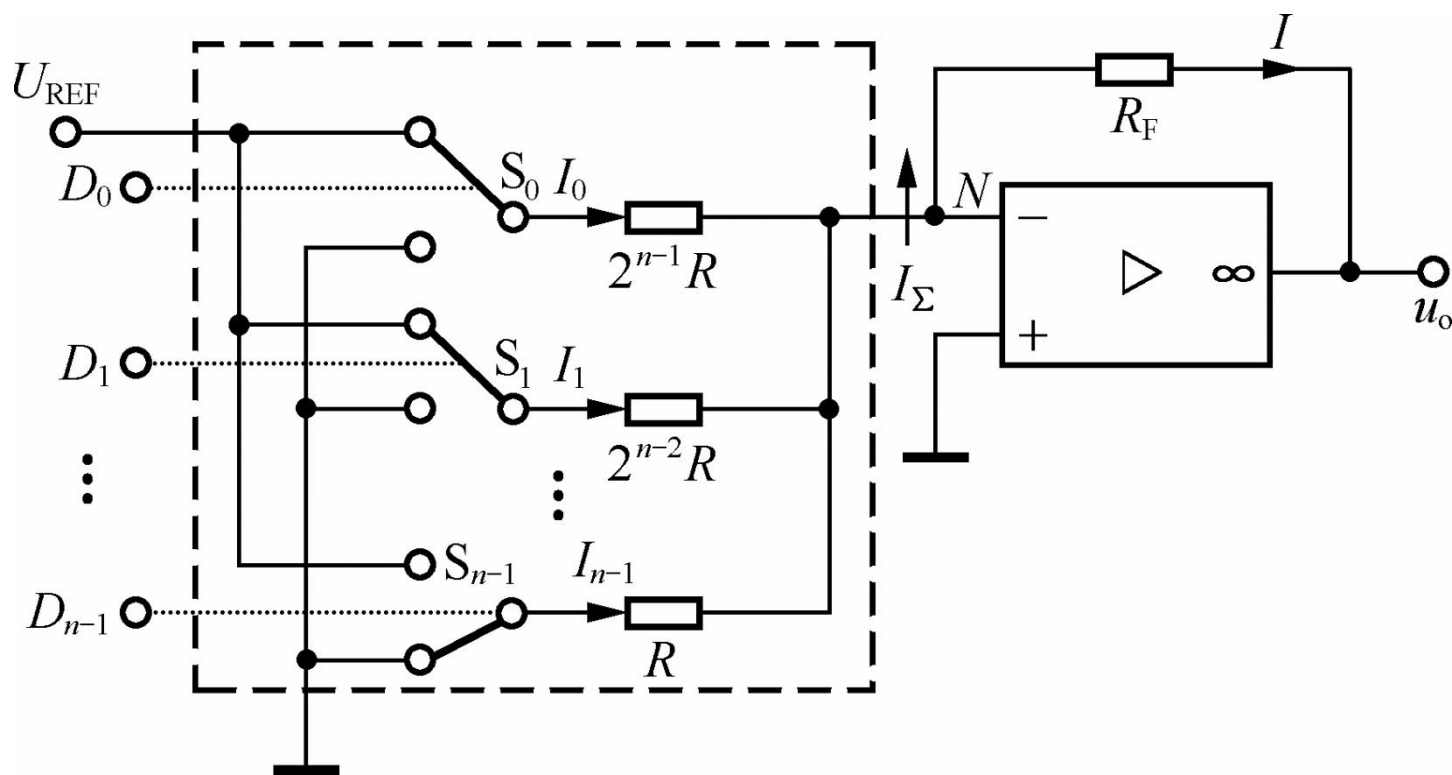
3. 作状态转移表或状态转移图或波形图

如无特别说明，所有时序电路分析设计后均需考虑自启动性问题！

主题六 A/D与D/A转换

6.1 常用的D/A转换电路

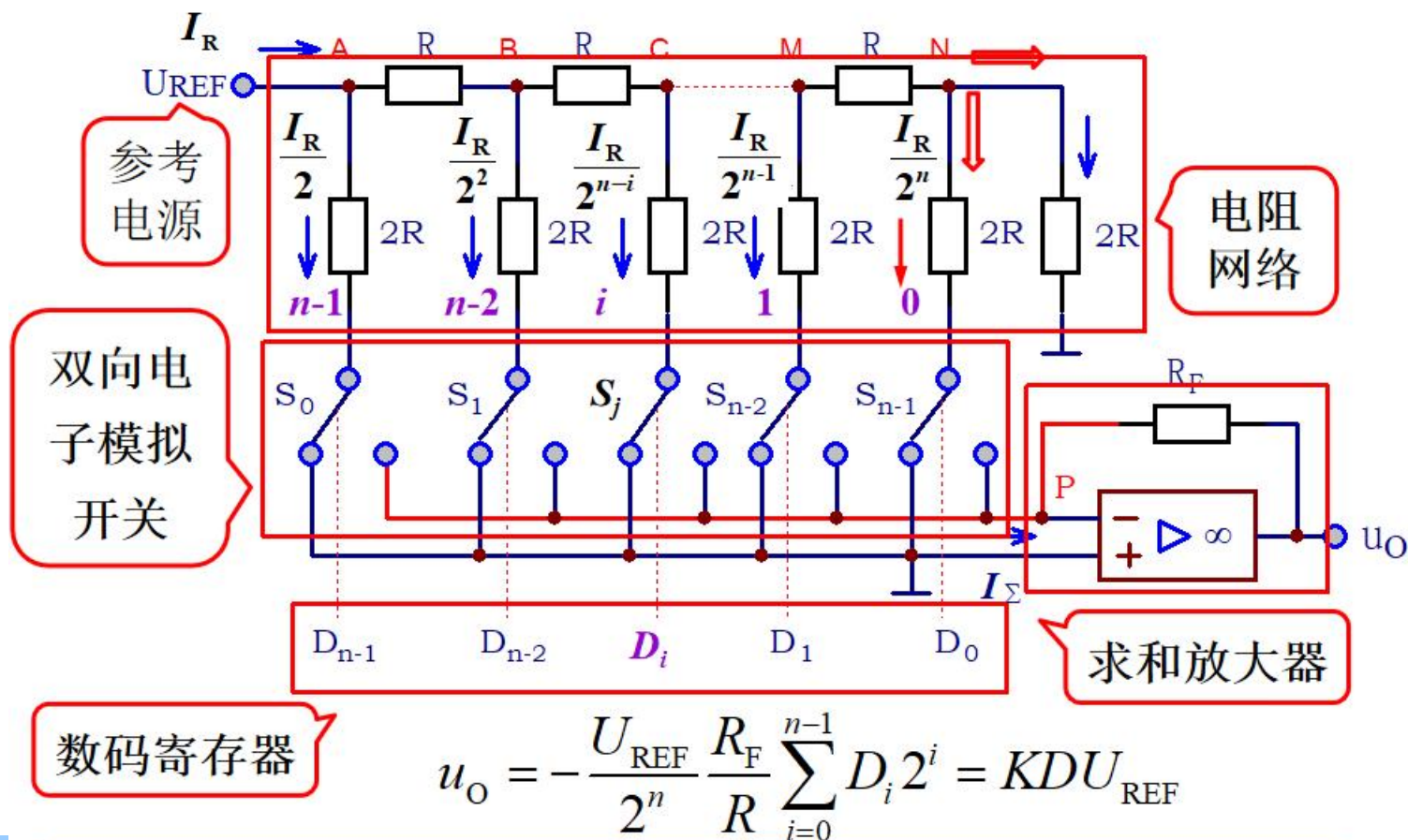
(1) 权电阻网络



主题六 A/D与D/A转换

6.1 常用的D/A转换电路

(2) 倒T型网络



主题六 A/D与D/A转换

6.2 DAC常用指标

(1) 转换电压

$$u_O = -\frac{U_{\text{REF}}}{2^n} D$$

(2) 满量程电压

$$U_{\text{Om}} = -\frac{2^n - 1}{2^n} U_{\text{REF}}$$

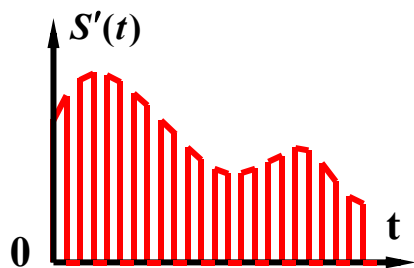
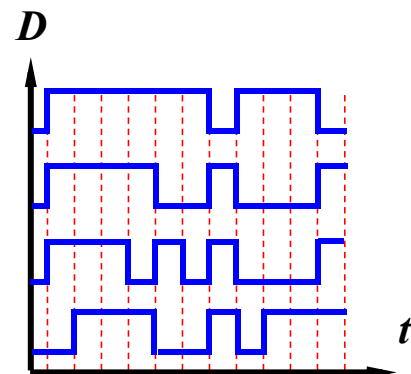
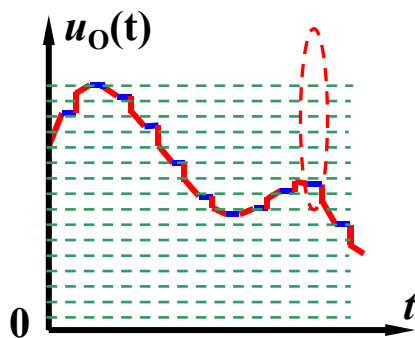
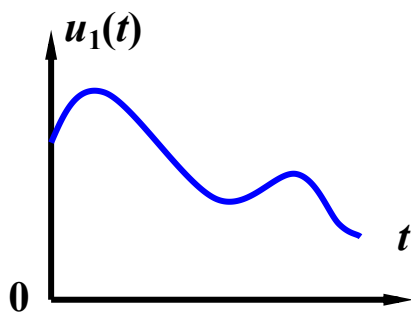
(3) 分辨率

$$u_O = -\frac{U_{\text{REF}}}{2^n} D$$
$$R' = \Delta u_O \big|_{\Delta D=1} = -\frac{U_{\text{REF}}}{2^n} = \frac{U_{\text{Om}}}{2^n - 1}$$
$$R' = \frac{\Delta u_O \big|_{\Delta D=1}}{U_{\text{Om}}} = \frac{1}{2^n - 1}$$

主题六 A/D与D/A转换

6.3 A/D的一般流程

采样 → 保持 → 量化 → 编码



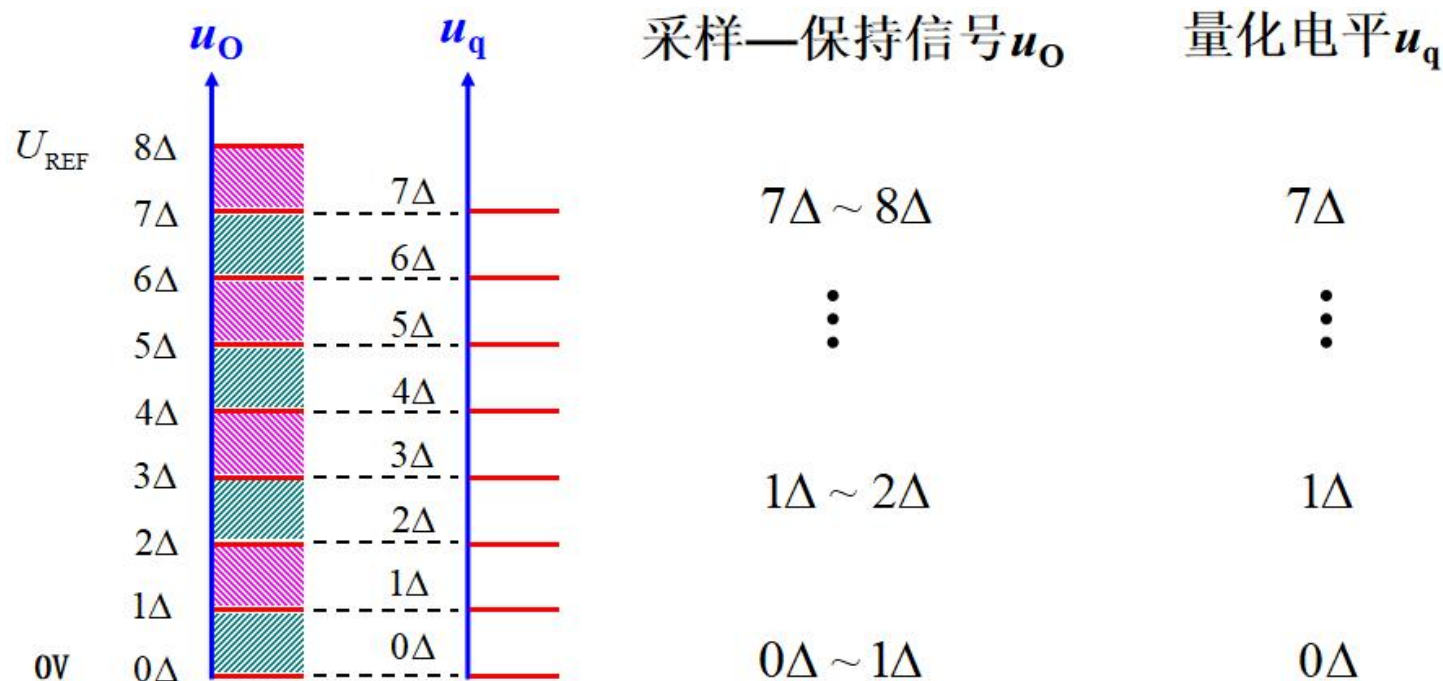
量化电平

量化单位 Δ 的整数倍

主题六 A/D与D/A转换

6.4 A/D的量化方案

(1) 舍尾方案



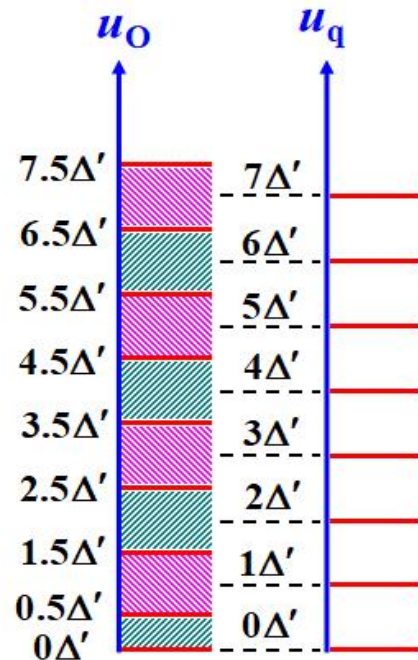
$$\Delta = \frac{U_{REF}}{2^n} = \frac{1}{8} U_{REF}$$

$$\text{最大量化误差 } \varepsilon_{\max} = 1\Delta$$

主题六 A/D与D/A转换

6.4 A/D的量化方案

(2) 四舍五入方案



采样—保持信号 u_O 量化电平 u_q

$6.5\Delta' \sim 7.5\Delta'$

$7\Delta'$

\vdots

\vdots

$0.5\Delta' \sim 1.5\Delta'$

$1\Delta'$

$0\Delta' \sim 0.5\Delta'$

$0\Delta'$

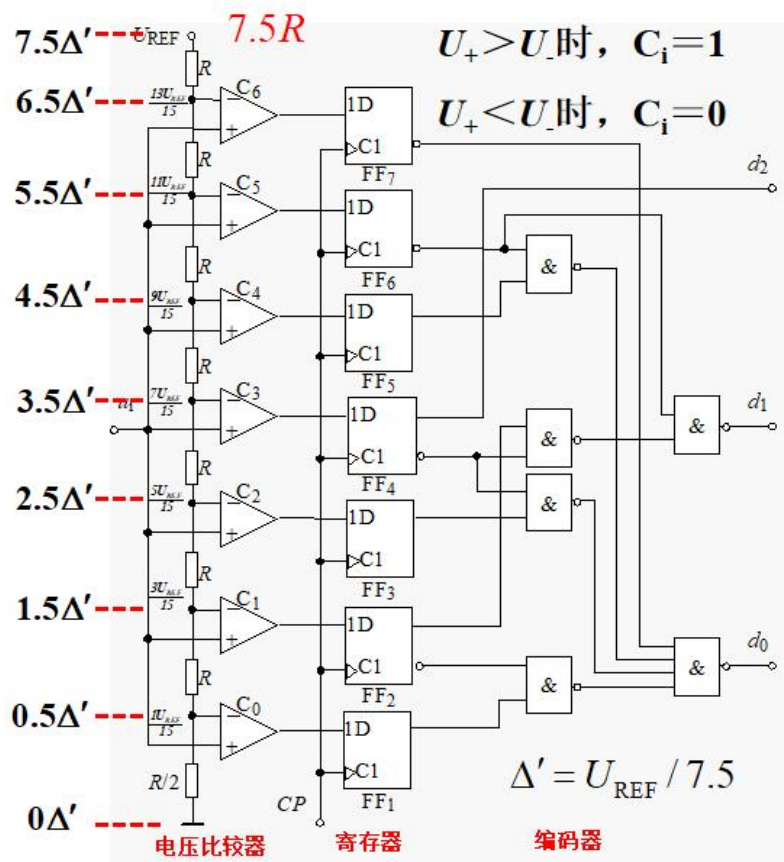
$$\Delta' = \frac{2}{2^{n+1} - 1} U_{\text{REF}} = \frac{2}{15} U_{\text{REF}}$$

$$\varepsilon_{\text{max}} = \frac{\Delta'}{2}$$

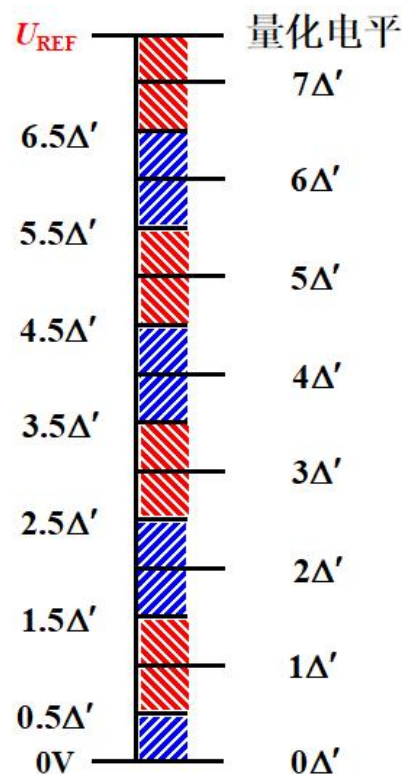
主题六 A/D与D/A转换

6.5 常用的A/D电路

(1) 并行比较型ADC



四舍五入方法



$$\begin{aligned} Q^{n+1} &= J\bar{Q}^n + \bar{K}Q^n \\ &= F\bar{Q}^n + FQ^n = F \end{aligned}$$

主题六 A/D与D/A转换

6.6 ADC的常用指标

(1) 分辨率

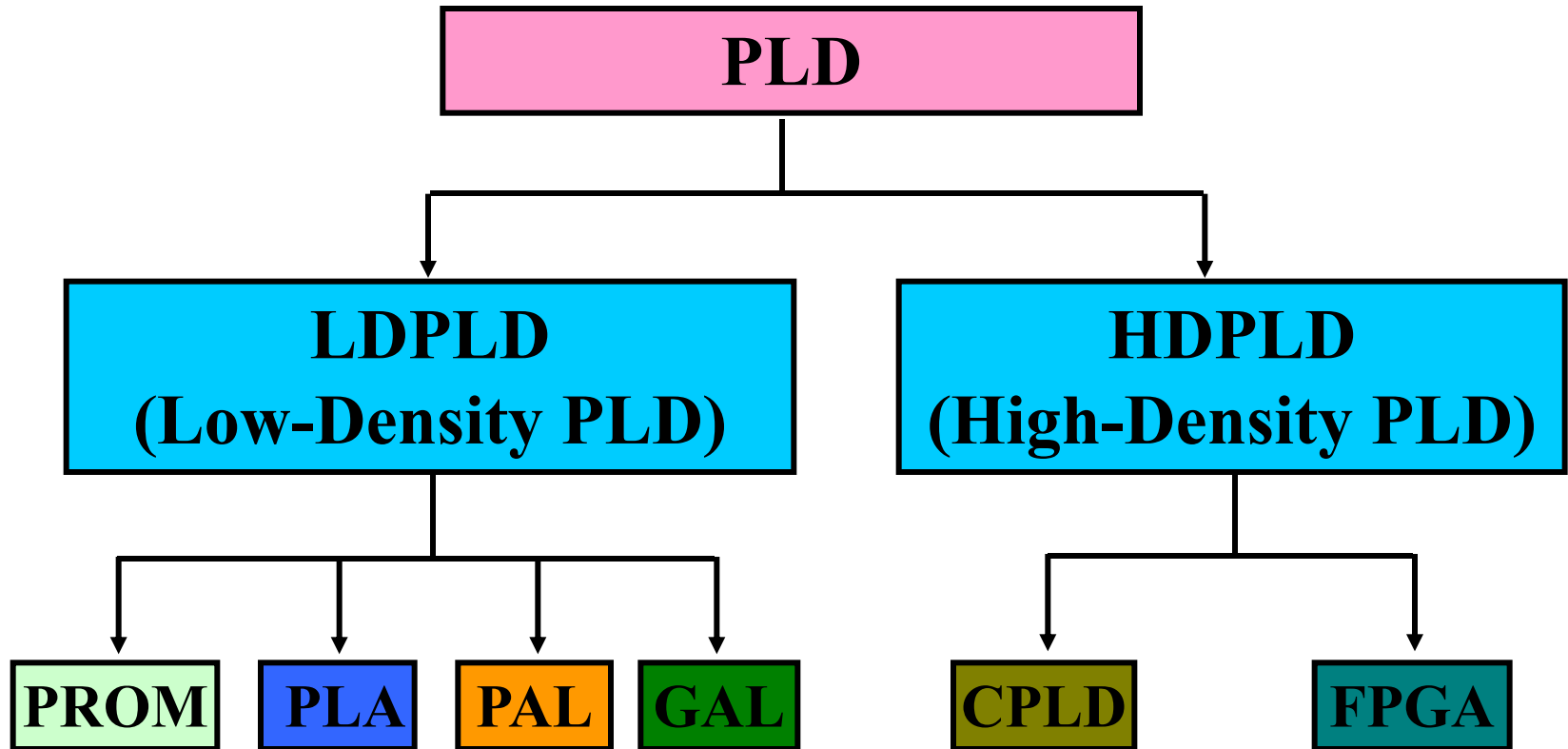
$$R' \approx \frac{U_{\text{REF}}}{2^n} \approx \frac{U_{\text{Im}}}{2^n}$$

$$R' = \frac{\Delta u}{U_{\text{REF}}} \approx \frac{\frac{U_{\text{REF}}}{2^n}}{U_{\text{REF}}} \approx \frac{1}{2^n}$$

(2) 时间复杂度、空间复杂度

主题七 可编程逻辑器件

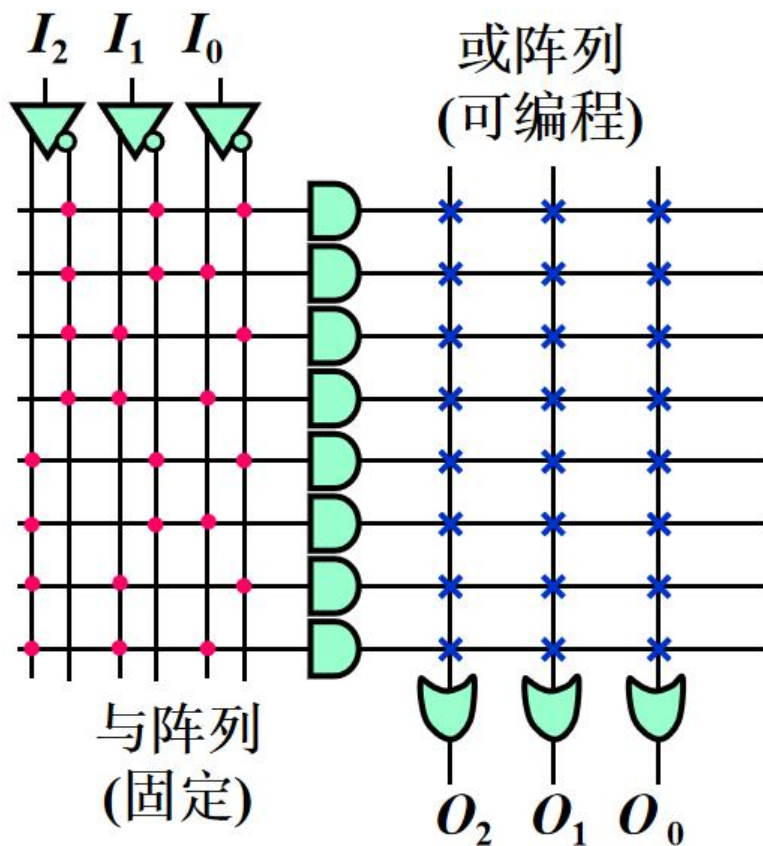
7.1 PROM、PLA、PAL、GAL



主题七 可编程逻辑器件

7.1 PROM、PLA、PAL、GAL

(1) PROM



20世纪70年代初。

缺点：

- 只能实现标准与或式
- 芯片面积大
- 利用率低，不经济

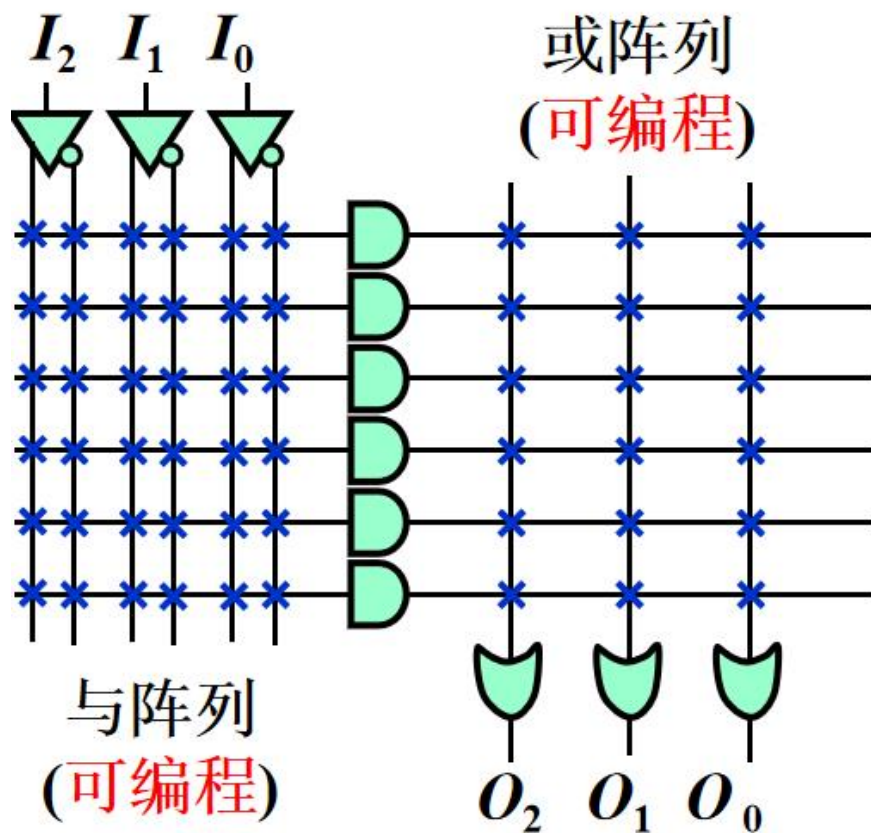
用途：

- 存储器
- 函数表
- 显示译码电路

主题七 可编程逻辑器件

7.1 PROM、PLA、PAL、GAL

(2) PLA



20世纪 70年代初。

优点：

- 与阵列或阵列都可编程
- 能实现最简与或式

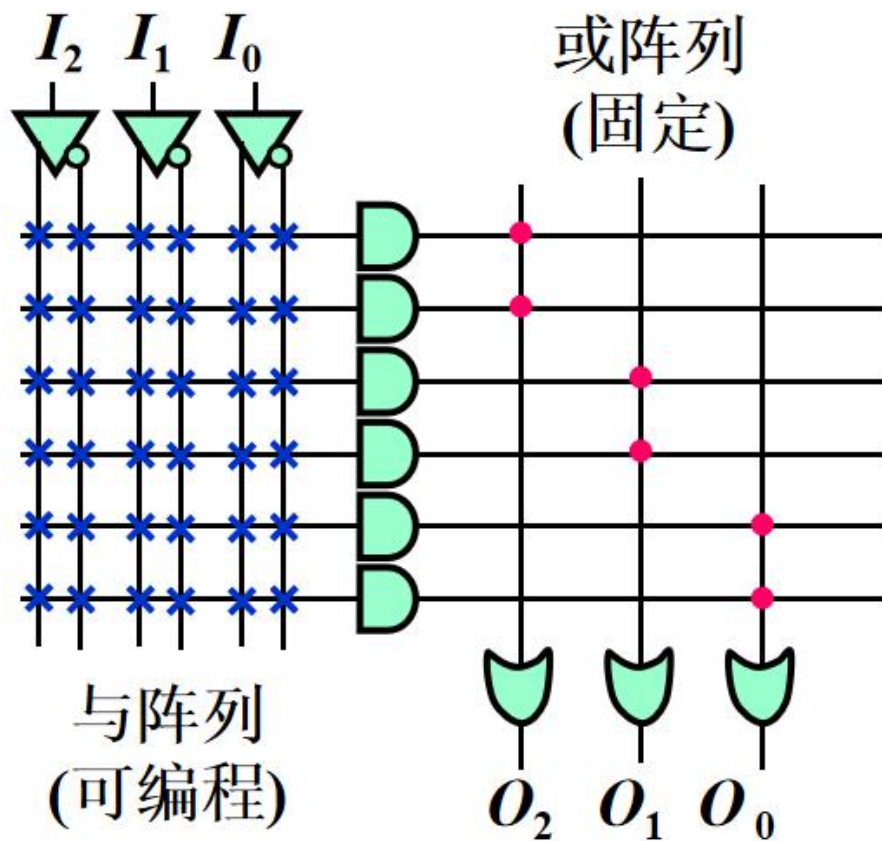
缺点：

- 价格较高
- 资源利用率不高

主题七 可编程逻辑器件

7.1 PROM、PLA、PAL、GAL

(3) PAL



20世纪70年代末。

优点：

- 速度高
- 价格低
- 采用编程器现场编程

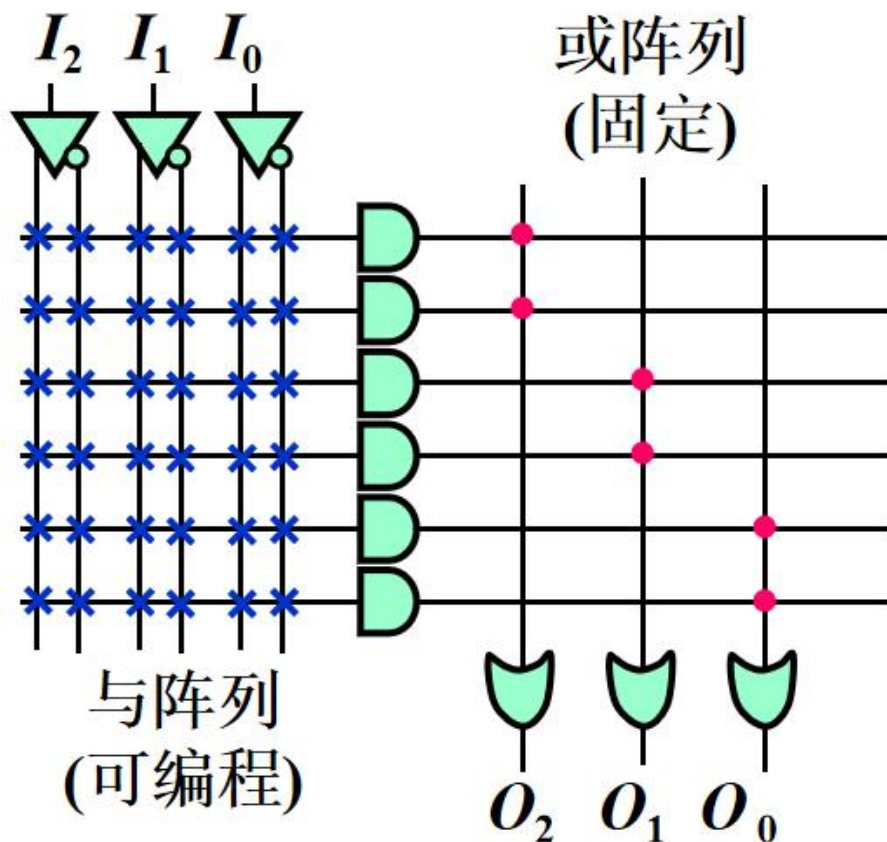
缺点：

- 输出方式固定
- 一次编程

主题七 可编程逻辑器件

7.1 PROM、PLA、PAL、GAL

(4) GAL



20世纪80年代初。

优点：

- 具有 PAL 的功能
- 采用逻辑宏单元使输出自行组态
- 功能更强，使用灵活，应用广泛

主题七 可编程逻辑器件

7.1 PROM、PLA、PAL、GAL

(5) 比较

名称	与阵列	或阵列	输出部分
PROM	固定	可编程	固定
PLA	可编程	可编程	固定
PAL	可编程	固定	固定
GAL	可编程	固定	可配置

主题七 可编程逻辑器件

7.1 PROM、PLA、PAL、GAL

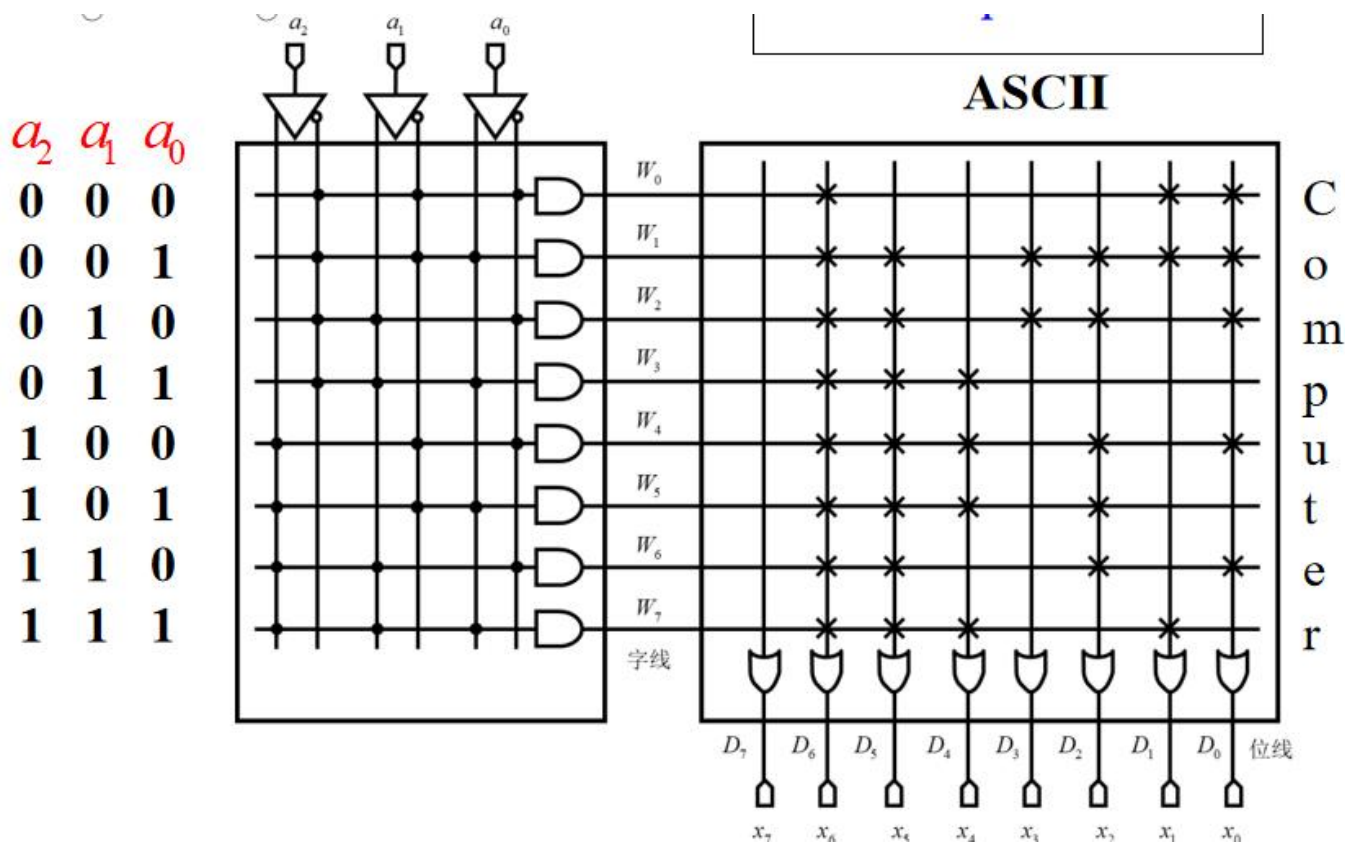
(5) 比较

名称	与阵列	或阵列	输出部分
PROM	固定	可编程	固定
PLA	可编程	可编程	固定
PAL	可编程	固定	固定
GAL	可编程	固定	可配置

主题七 可编程逻辑器件

7.2 ROM的功能

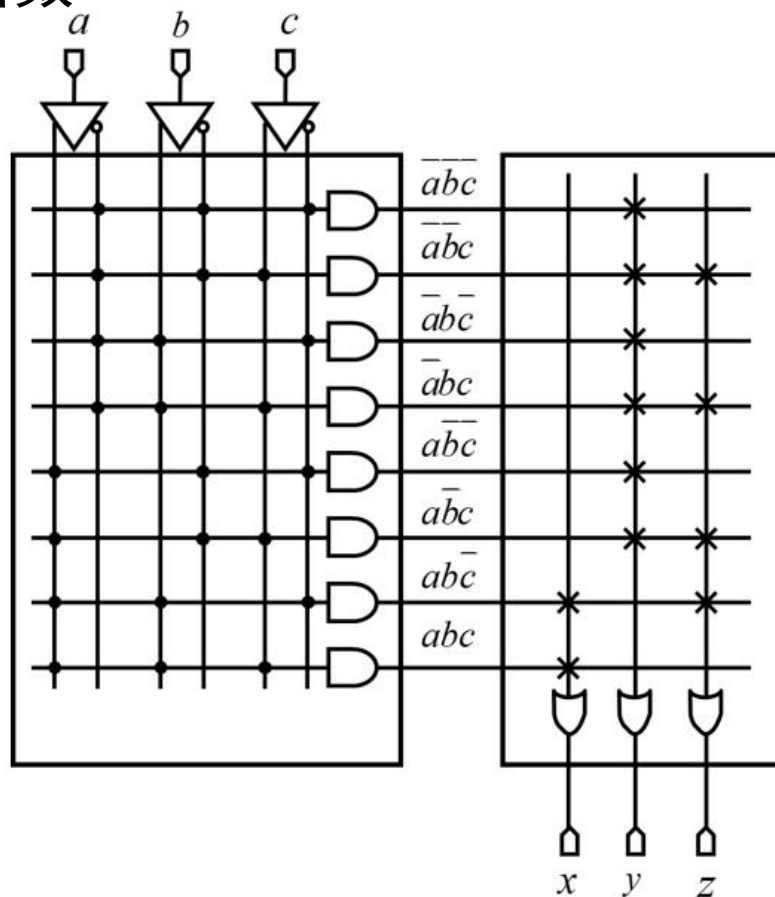
(1) 存储功能



主题七 可编程逻辑器件

7.2 ROM的作用

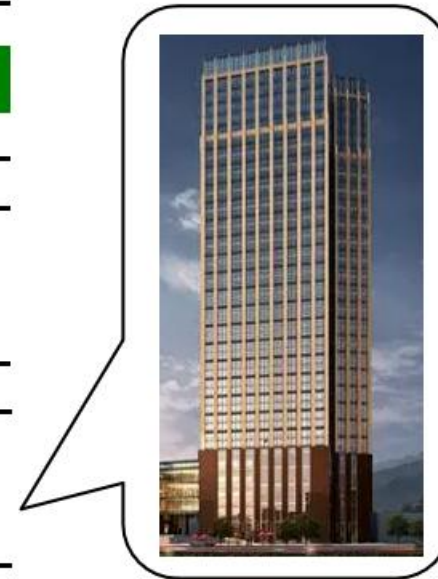
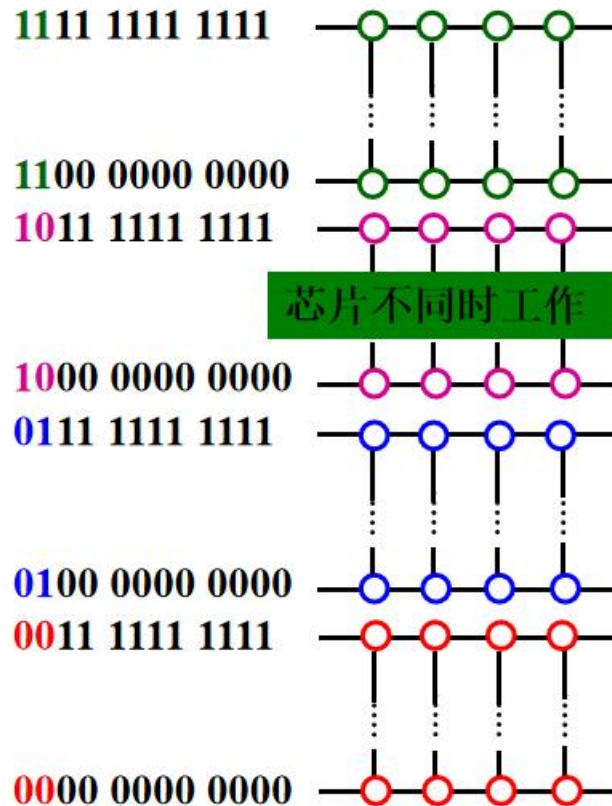
(2) 实现逻辑函数



主题七 可编程逻辑器件

7.3 ROM的扩展

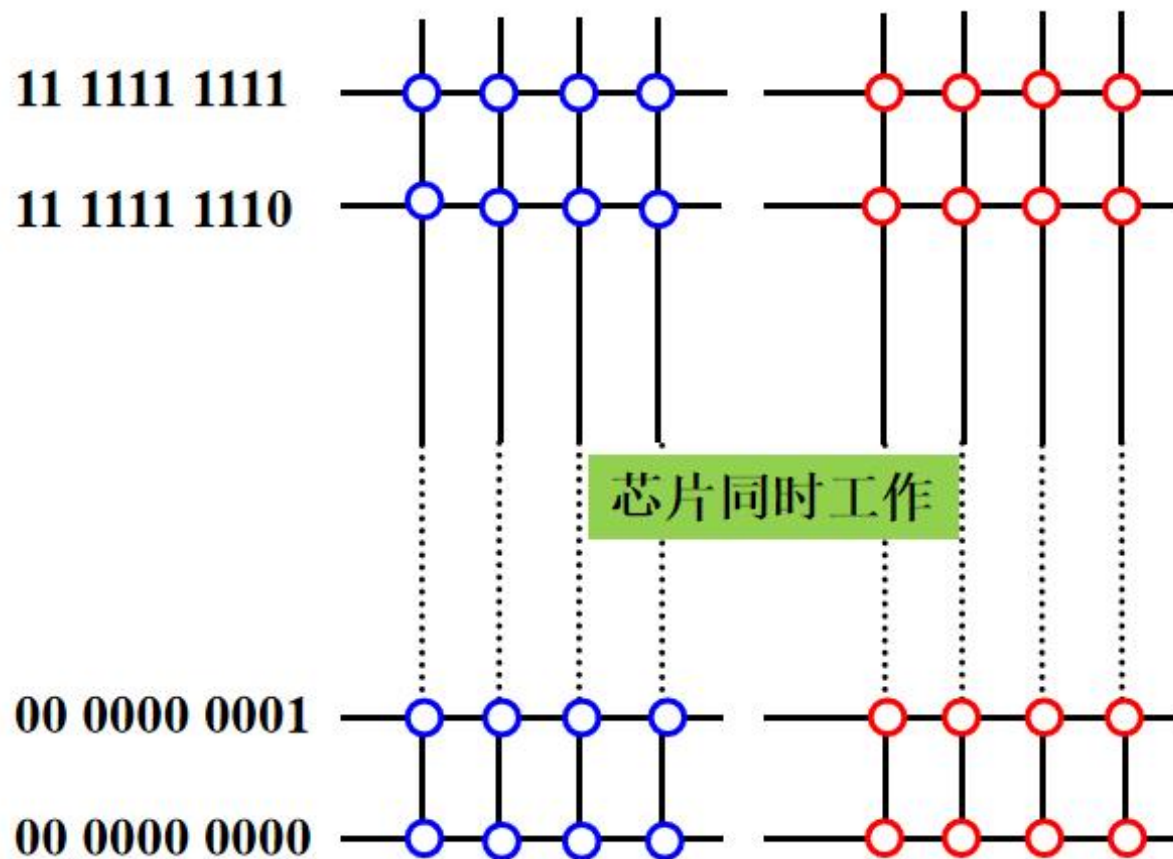
(1) 字扩展



主题七 可编程逻辑器件

7.3 ROM的扩展

(2) 位扩展



主题八 Verilog语言初步

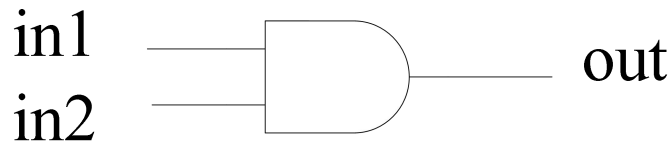
8.1 门原语定义器件语法

门原语名 实例名 (端口连接)

注：实例名可省略，端口连接采用输出在前，输入在后。

and (与)	or (或)	xor (异或)
nand (与非)	nor (或非)	xnor (同或)

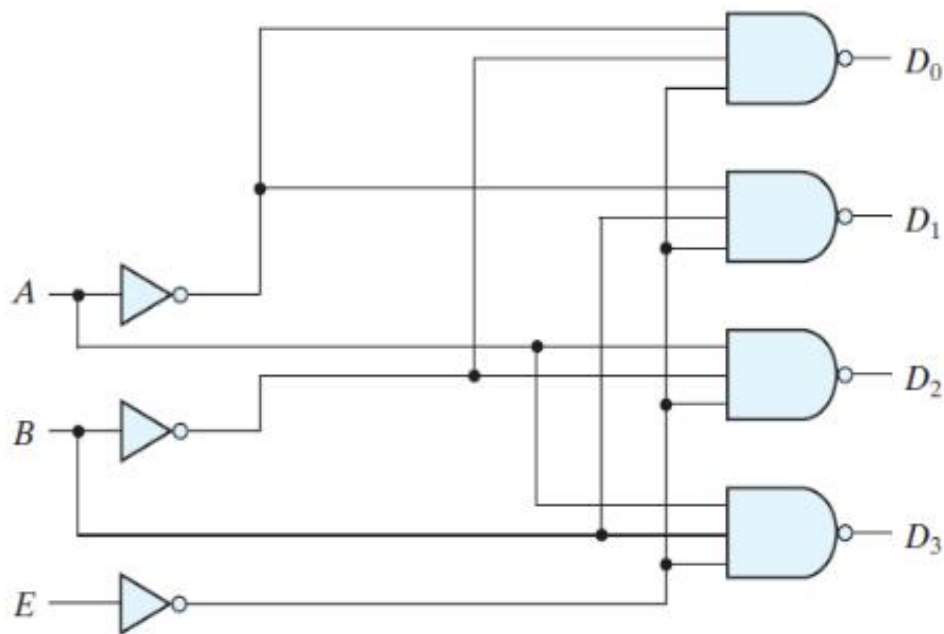
and (out, in1, in2);



主题八 Verilog语言初步

8.2 实现组合逻辑电路

(1) 门级建模



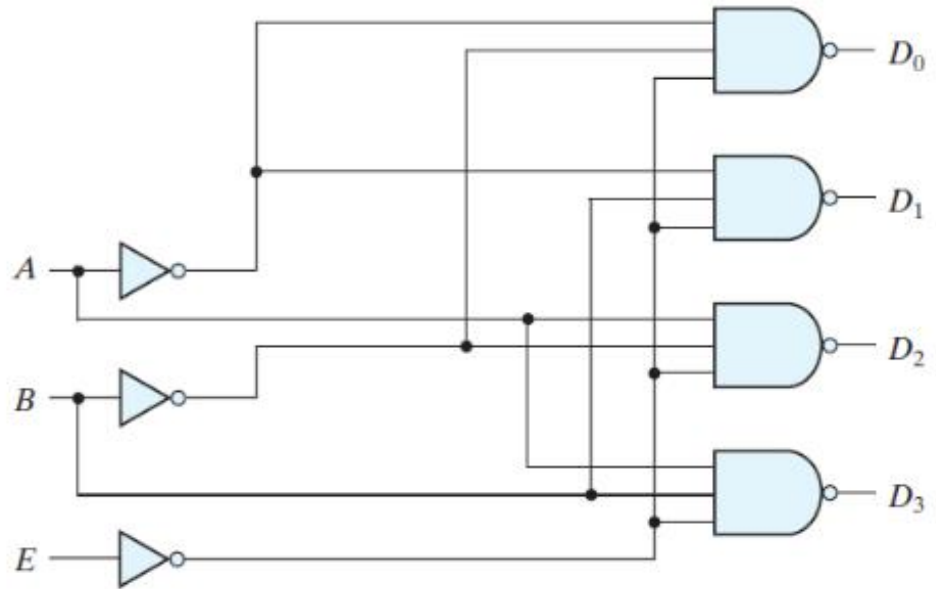
<i>E</i>	<i>A</i>	<i>B</i>	<i>D</i> ₀	<i>D</i> ₁	<i>D</i> ₂	<i>D</i> ₃
1	<i>X</i>	<i>X</i>	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	1

主题八 Verilog语言初步

8.2 实现组合逻辑电路

(1) 门级建模

```
module decoder_2x4_gates (D, A, B, E);  
  output [3:0] D;  
  input A,B;  
  input E;  
  wire A_not, B_not, E_not;  
  not  
    G1 (A_not, A),  
    G2 (B_not, B),  
    G3 (E_not, E);  
  nand  
    G4 (D[0], A_not, B_not, E_not),  
    G5 (D[1], A_not, B, E_not),  
    G6 (D[2], A, B_not, E_not),  
    G7 (D[3], A, B, E_not);  
endmodule
```

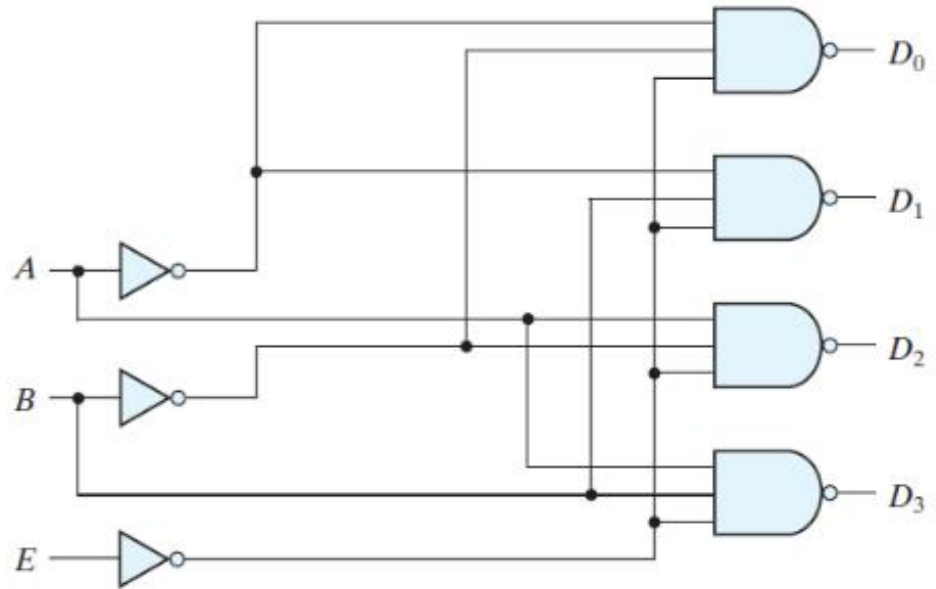


主题八 Verilog语言初步

8.2 实现组合逻辑电路

(1) 门级建模

```
module decoder_2x4_gates (D, A, B, E);  
output [3:0] D;  
input A,B;  
input E;  
wire A_not, B_not, E_not;  
not  
    G1 (A_not, A),  
    G2 (B_not, B),  
    G3 (E_not, E);  
nand  
    G4 (D[0], A_not, B_not, E_not),  
    G5 (D[1], A_not, B, E_not),  
    G6 (D[2], A, B_not, E_not),  
    G7 (D[3], A, B, E_not);  
endmodule
```

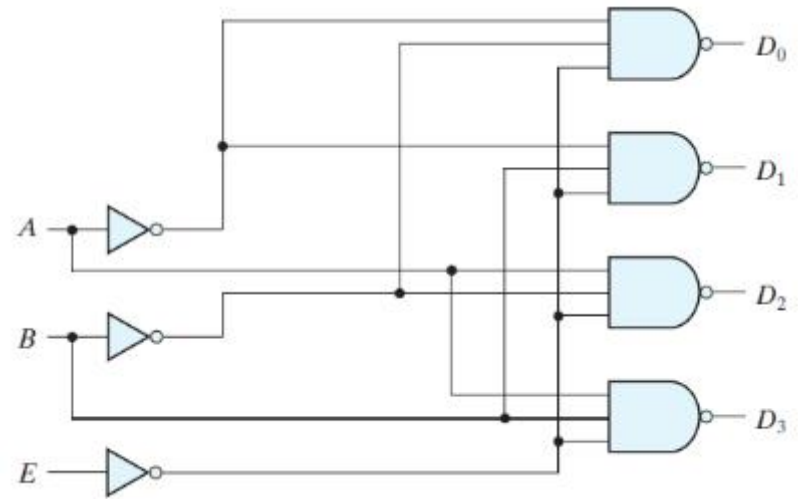


主题八 Verilog语言初步

8.2 实现组合逻辑电路

(2) 数据流建模

```
module decoder_2x4_df (  
  output [3:0] D,  
  input A, B, enable  
);  
assign D[0] = (!((A) && (!B) && (!enable)),  
  D[1] = (!((A) && B && (!enable)),  
  D[2] = !(A && (!B) && (!enable)),  
  D[3] = !(A && B && (!enable));  
endmodule
```



主题八 Verilog语言初步

8.3 实现触发器

```
module dff_rs_async(clk,r,s,d,q);  
input clk,r,s,d;  
output q;  
reg q;  
always@(posedge clk or posedge r or posedge s)  
begin  
  if(r) q<=1'b0;  
  else if(s) q<=1'b1;  
  else q<=d;  
end  
endmodule
```

