

数字逻辑与数字系统

复习随记

计算机 74 任隽阳

【目录】

第一章	数字逻辑基础	1
第二章	组合逻辑电路	6
第三章	时序逻辑电路	13
第四章	可编程逻辑器件	20
第五章	数字系统	23

第一章 数字逻辑基础

1. 物理量的表示:

模拟量: 连续变化

数字量: 离散变化

2. 进制转换

可以十进制作为转换桥梁。

(1) 多项式替代法

在 β 进制下完成 $N_\alpha \rightarrow N_\beta$ 的转换

多项式替代法是在 β 进制下完成 N_α 到 N_β 的转换的, 因此, 要求熟悉 β 进制的算术运算规则。

例如: 将 $(121.2)_3$ 转换为二进制

$$\begin{aligned}(121.2)_3 &= (1 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 + 2 \times 10^{-1})_3 \\&= (1 \times 11^2 + 10 \times 11^1 + 1 \times 11^0 + 10 \times 11^{-1})_2 \\&= (1001 + 110 + 1 + 0.101010...)_2 \\&= (10000.101010...)_2\end{aligned}$$

(2) 基数乘法

要点: 在 α 进制下完成 $N_\alpha \rightarrow N_\beta$ 的转换

① 整数部分转换用基数除法 (逆序取余)

② 小数部分转换用基数乘法 (正序取证)

(3) 直接转换法

由小数点开始向左右划分, 3 位二进制数对应一位八进制数, 4 位二进制数对应一位十六进制数

(4) 确定转换精度

α 进制的小数有 i 位, 转换成 β 进制后, 至少具有相同精度的小数是 j 位, 则

$$(0.1)_\alpha^i \geq (0.1)_\beta^j$$

在十进制中可表示为

$$\left(\frac{1}{\alpha}\right)^i \geq \left(\frac{1}{\beta}\right)^j, \text{ 即 } \beta^j \geq \alpha^i$$

因此需要

$$j \geq i \cdot \frac{\lg \alpha}{\lg \beta}$$

3. 数的表示及其运算

(1) 定点表示法

选取合适比例因子，乘以处理数，使其为定点小数表示（小数点在最高位之前，符号位之后）或定点整数表示（小数点在最低位之后）。

(2) 浮点表示法

将字长划分为阶符+阶码+尾数符+尾数表示

(3) 带符号数的代码表示及其运算

正数：[X]_原=[X]_反=[X]_补

负数：[X]_反=[X]_原除符号位外按位取反，[X]_补=[X]_反+1

反码的本质：

$$[X]_{\text{反}} = \begin{cases} X, & 0 \leq X \leq 2^{n-1} - 1 \\ (2^{n-1} - 1) + X, & -(2^{n-1} - 1) \leq X \leq 0 \end{cases}$$

运算性质：

- $[z]_{\text{反}} = [x+y]_{\text{反}} = [x]_{\text{反}} + [y]_{\text{反}}$
- $[x-y]_{\text{反}} = [x]_{\text{反}} + [-y]_{\text{反}}$
- 当符号位产生进位时，将产生的进位要加到数值位的最低位

补码的本质：

$$[X]_{\text{补}} = \begin{cases} X, & 0 \leq X \leq 2^{n-1} - 1 \\ (2^n - 1) + X, & -(2^{n-1} - 1) \leq X \leq 0 \end{cases}$$

运算性质：

- $[z]_{\text{补}} = [x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}}$
- $[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}}$
- 当符号位产生进位时，将产生的进位丢掉。

4. 十进制数的代码表示及其运算

十进制整数	8421 码 (BCD 码)	2421 码	余 3 码
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

代码	对应的十进制数值	代码直接按位转换
8421 码	有权码 (<i>Weighted code</i>) $8A_3 + 4A_2 + 2A_1 + 1A_0$	$(13)_{10} = (00010011)_{\text{BCD}}$ $(1011101010000)_{\text{BCD}} = (1750)_{10}$
2421 码	有权码、对 9 自补码 $2A_3 + 4A_2 + 2A_1 + 1A_0$	$(13)_{10} = (00010011)_{2421}$ $(1110110110000)_{2421} = (1750)_{10}$
余 3 码	无权码、对 9 自补码 $8A_3 + 4A_2 + 2A_1 + 1A_0 - 0011$	$(13)_{10} = (01000110)_{\text{余}3}$ $(100101010000011)_{\text{余}3} = (1750)_{10}$

代码运算若直接按二进制数运算，则运算结果要修正。

两个 BCD 码（或 2421 码、余 3 码）相加，其和可能不是一个正确的 BCD 码（或 2421 码、余 3 码）
2421 码、余 3 码是一种对 9 的自补代码。

5. 异或运算

A	$0 \oplus A$	$1 \oplus A$	$A \oplus A$	$A \oplus \bar{A}$
0	0	1	0	1
1	1	0	0	1

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

相同得 0，不同得 1，与 1 运算取反，与 0 运算不变。

6. 可靠性编码

(1) 格雷码

十进制数	二进制数	典型 Gray	十进制 Gray
0	0000	0000	0000
1	0001	0001	0001
2	0010	0011	0011
3	0011	0010	0010
4	0100	0110	0110
5	0101	0111	1110
6	0110	0101	1010
7	0111	0100	1011
8	1000	1100	1001
9	1001	1101	1000
10	1010	1111	
11	1011	1110	
12	1100	1010	
13	1101	1011	
14	1110	1001	
15	1111	1000	

$$G_0 = B_1 \oplus B_0$$

$$G_1 = B_2 \oplus B_1$$

$$G_{n-2} = B_{n-1} \oplus B_{n-2}$$

$$G_{n-1} = 0 \oplus B_{n-1} = B_{n-1}$$

典型格雷码第 i 位计算： $G_i = B_{i+1} \oplus B_i$ $B_i = G_i \oplus B_{i+1}$

(2) 奇偶校验码

① 奇校验：

构成编码中 1 个数为奇数。

$$\text{校验位 } P_{\text{奇}} = B_{n-1} \oplus B_{n-2} \oplus \dots \oplus B_1 \oplus B_0 \oplus 1$$

$$\text{校验关系：} B_{n-1} \oplus B_{n-2} \oplus \dots \oplus B_1 \oplus B_0 \oplus P_{\text{奇}} = 1$$

② 偶校验

构成编码中 1 个数为偶数。

$$\text{校验位：} P_{\text{偶}} = B_{n-1} \oplus B_{n-2} \oplus \dots \oplus B_1 \oplus B_0$$

$$\text{校验关系：} B_{n-1} \oplus B_{n-2} \oplus \dots \oplus B_1 \oplus B_0 \oplus P_{\text{偶}} = 0$$

(3) 海明校验码

以四位信息位 $B_4 B_3 B_2 B_1$ 为例，在传输前生成它的海明校验码：

位序： $B_8 B_4 B_2 P_3 B_1 P_2 P_1$

校验位的生成公式： $P_3 = B_8 \oplus B_4 \oplus B_2$

$$P_2 = B_8 \oplus B_4 \oplus B_1 \quad \text{偶校验}$$

$$P_1 = B_8 \oplus B_2 \oplus B_1$$

$$\text{校验位关系：} S_3 = B_8 \oplus B_4 \oplus B_2 \oplus P_3 = 0$$

$$S_2 = B_8 \oplus B_4 \oplus B_1 \oplus P_2 = 0$$

$$S_1 = B_8 \oplus B_2 \oplus B_1 \oplus P_1 = 0$$

否则以最低位为第 1 位, 第 $(S_3S_2S_1)_2$ 位出现错误。

信息位 n 位, 校验位 k 位的海明校验码: $2^k - 1 \geq k + n$

7. 逻辑代数运算

基本运算: 逻辑与、逻辑或、逻辑非

复合运算: 与非、或非、与或非、异或、同或

同或运算: (真值表见右)

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

- 相同得 1, 不同得 0, 与 1 运算得自身, 与 0 运算取反
- 当变量为偶数时, 同或运算与异或运算之间具有互补关系;
- 当变量为奇数时, 同或运算与异或运算之间具有相等关系。

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

反演规则: 将原函数 F 所有变量取反、0 变 1, 1 变 0, $+$ 变 \cdot , \cdot 变 $+$, 得到原函数的反函数 \bar{F}

对偶规则: 将原函数 F 0 变 1, 1 变 0, $+$ 变 \cdot , \cdot 变 $+$, 得到原函数的对偶函数 F' (公理定理的对偶式都成立)

8. 逻辑函数的表示

(1) 真值表

(2) 表达式

① 基本表达式

② 标准形式

- 最小项 m_i : 每个变量以原变量或反变量形式出现且仅出现一次, 组成与项。其中 i 的确定方法: 用 1 代替原变量, 用 0 代替反变量, 所得二进制数等值十进制数。
- 最小项标准式: $\sum m^n(i_1, i_2, \dots, i_k)$, 反复应用 $A = A(B + \bar{B})$ 变形, n 变量的所有最小项之和为 1。
- 最大项 M_i : 每个变量以原变量或反变量形式出现且仅出现一次, 组成或项。其中 i 的确定方法: 用 1 代替原变量, 用 0 代替反变量, 所得二进制数等值十进制数。
- 最大项标准式: $\prod M^n(i_1, i_2, \dots, i_k)$, 反复应用 $A = A + (B\bar{B}) = (A + B)(A + \bar{B})$ 变形, n 变量所有最大项之积为 0

掌握: 通过原函数的一种标准形式写出另一种标准形式、写出其反函数的两种形式。

(3) 卡诺图

每一小格对应一个最小项。最小项标 1, 最大项标 0, 无关项标 d 。

卡诺图的行和列坐标按照变量组合的二进制格雷码顺序标注, 其变量顺序遵从真值表中变量从左至右的顺序。

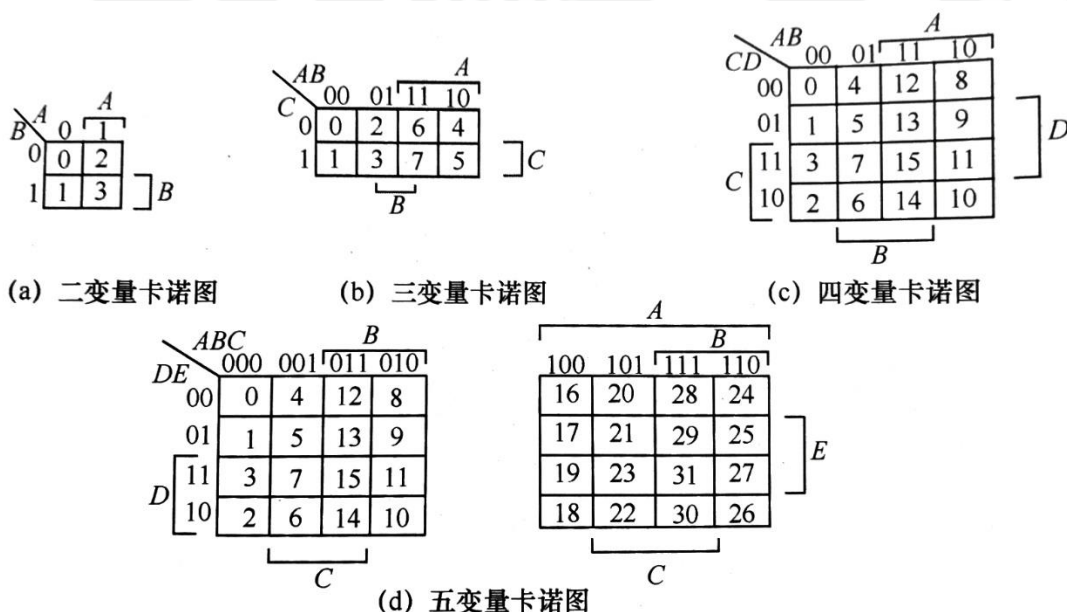


图 1.3 几种变量数的卡诺图

逻辑运算的几何含义:

与——卡诺图公共区域; 或——卡诺图覆盖全部区域; 非——卡诺图函数覆盖以外区域。

9. 逻辑函数化简

(1) 代数法化简

- ① 最简与或式
- ② 最简或与式

二次求反法：先求反函数的最简与或式，再求反

二次求偶法：先求对偶函数的最简与或式，再将对偶函数求对偶

- ③ 化简为与非式

替代尾因子法：

头因子：每个与项中的原变量分子 尾因子：每个与项中的反变量部分

把头因子中的任何变量放入任一个尾因子中，与项逻辑值不变。

- a. 将最简与或式中含有相同头因子的与项合并成一个与项
- b. 列出最简与或式中所有与项的头因子、尾因子及替代尾因子
- c. 选择共享的替代尾因子（共享与项数尽可能多）

例：求函数 $F = BC\bar{A} + \bar{A}CD + BC\bar{D} + \bar{A}\bar{B}\bar{C} + \bar{B}CD + \bar{A}\bar{C}\bar{D}$ 的最简与非式

解： $F = \overline{AC\bar{B}D} + \overline{BC\bar{A}D} + \overline{CD\bar{A}B}$

与项	头因子	尾因子	替代尾因子
$AC\bar{B}D$	A	\bar{C}	\bar{C}, \bar{AC}
		$\bar{B}D$	$\bar{B}D, \bar{ABD}$
$BC\bar{A}D$	BC	$\bar{A}D$	$\bar{A}D, \bar{ABD}, \bar{ACD}, \bar{ABCD}$
$CD\bar{A}B$	CD	$\bar{A}B$	$\bar{A}B, \bar{ABC}, \bar{ABD}, \bar{ABCD}$

替代尾因子 \bar{ABD} 被 3 个与项共享， \bar{C} 与 \bar{AC} 中 \bar{C} 形式简单，故得到最终表达式：

$$F = \overline{AC\bar{B}D} + \overline{BC\bar{A}D} + \overline{CD\bar{A}B} = \overline{\overline{AC} \bar{A} \bar{B} D} \overline{BC \bar{A} B D} \overline{CD \bar{A} B D}$$

(2) 利用卡诺图化简

- ① 将逻辑函数表示在卡诺图上，无关项用 d 表示
 - ② 画出所有极大圈，将 d 包括在内，但不能仅包含 d
 - ③ 确定全部实质最小项，找到所有必要极大圈
 - ④ 挑选最少数量的极大圈覆盖没有被必要极大圈覆盖的小方格，得到最小覆盖
 - ⑤ 写出对应表达式，得到最简与或式。
- 求最简或与式：利用反函数或对偶函数的最简与或式求得。
 - 求与非式：禁止逻辑法（任何函数如用属于它的最小项之非与其相乘，相当于从该函数中去除了这几个最小项，该方法基本思路是先将共享 0 项纳入极大圈，再将其作为禁止项禁止）
 - 求或非式：根据对偶规则，先求出原函数的对偶式（与或式），再用前面的替代尾因子法、禁止逻辑法等寻找尽可能多的共享与非项，化简后再求一次对偶即可。

第二章 组合逻辑电路

电路名称	长方形符号	变形符号	等效符号
跟随器			
非门			
与门			
或门			
与非门			
或非门			
与或非门			
异或门			

图 2.8 常用逻辑门的三种表示形式

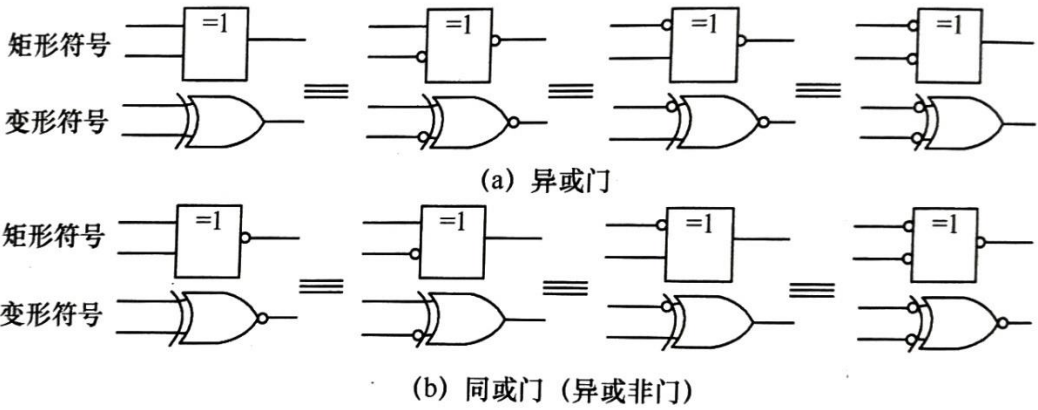


图 2.10 异或门和同或门的变形符号和等效符号

- 1. 信号有效级：
高有效：信号为高电平有效 (+, .H)
低有效：信号为低电平有效 (-, .L, 前缀/, 后缀*)
- 2. 引端有效级变换
选用器件的引端的有效级要尽量与所给信号的有效级相匹配一致
变换规则：
 - ① 引端添加或删除逻辑非，对应信号有效级取反
 - ② 内部连线两端同时添加或删除逻辑非符号
 - ③ 单个逻辑非符号在连线两端移动
 - ④ 若一个门的输入输出端同时加上或删除去逻辑非符号，或输入、输出信号有效级同时取反，且门的符号“与”、“或”互变时，则得到的新的逻辑图的功能不变。
- 3. 组合逻辑电路分析
基本分析过程：
 - ① 阅读组合逻辑电路图
 - ② 列出逻辑表达式，利用代数法、卡诺图法化简

- ③ 列出真值表或简易真值表
- ④ 分析真值表，用文字描述电路逻辑功能
- ⑤ 评价电路性能，必要时提出改进方案

4. 组合逻辑电路设计

基本设计过程：

- ① 通过真值表、卡诺图列出逻辑函数表达式
- ② 通过选择的门类型变换化简表达式
- ③ 画出逻辑电路图
- ④ 对电路分析和综合评价

5. 竞争与险象

竞争：同一信号经过不同路径到达某一点有时差。

险象：由于临界竞争，电路稳定输出前的短暂错误输出。

(1) 静态险象

分类：功能险象（无法用改变设计的方法消除）、逻辑险象

静态 1 险象：稳态 1 输出，出现 1→0→1 负向窄脉冲

静态 0 险象：稳态 0 输出，出现 0→1→0 正向窄脉冲

(2) 动态险象

输入变化前后的稳态输出值不同，且在输出稳定前输出变化 3 次（1→0→1→0 或 0→1→0→1）

(3) 险象的判别与消除

① 逻辑代数法

某个变量同时以原变量和反变量形式出现在逻辑表达式中，则该变量具备竞争条件。

保留具备竞争条件的变量，给其他变量赋值 0 或 1，如果得到下列形式，会出现相应险象

$$F = A + \bar{A} \text{ 静态 1 险象 (若 } A \text{ 从 } 1 \rightarrow 0) \quad F = A \cdot \bar{A} \text{ 静态 0 险象 (若 } A \text{ 从 } 0 \rightarrow 1)$$

$$\text{动态险象: } F = A + A \cdot \bar{A} \quad F = \bar{A} + A \cdot \bar{A} \quad F = A \cdot (\bar{A} + A) \quad F = \bar{A} \cdot (\bar{A} + A)$$

② 卡诺图法

在卡诺图中，与或式中的每个与项对应于圈 1 的一个卡诺圈，如果两个卡诺圈存在着部分相切，而这个相切的部分又没有被另外的卡诺圈所包含，则该电路必然存在静态 1 险象。相切部分是哪个变量的交替面，就是哪个变量由 1→0 时出现的险象。

消除方法：增加多余项使其 1 卡诺圈覆盖相切部分

在卡诺图中，按照圈 0 单元的卡诺圈是否存在部分相切，而这个相切的部分又没有被另外的卡诺圈所包含，则该电路必然存在静态 0 险象。相切部分是哪个变量的交替面，就是哪个变量由 0→1 时出现的险象。

消除方法：增加多余项使其 0 卡诺圈覆盖相切部分

③ 其他险象的消除方法

- 在输出端连接低通环节以减弱干扰
- 利用取样脉冲避开险象
- 时钟同步法

6. 常用 MSI 组合逻辑器件

(1) 译码器

① 2-4 译码器

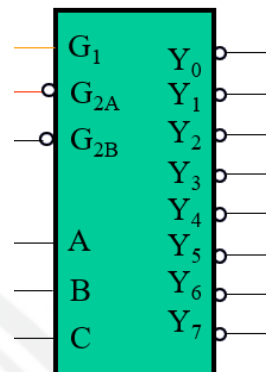
输 入			输 出			
EN	I ₁	I ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	d	d	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



② 74LS138

输 入						输 出							
G_1	$/G_{2A}$	$/G_{2B}$	C	B	A	$/Y_7$	$/Y_6$	$/Y_5$	$/Y_4$	$/Y_3$	$/Y_2$	$/Y_1$	$/Y_0$
0	d	d	d	d	d	1	1	1	1	1	1	1	1
d	1	d	d	d	d	1	1	1	1	1	1	1	1
d	d	1	d	d	d	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

74LS138



• 74LS138 的输出信号为低有效，它有三个使能输入端 (G_1 、 $/G_{2A}$ 、 $/G_{2B}$)，只有在三个使能输入全部有效时，才能有正确的有效输出。

• 74LS138 的内部功能可用逻辑表达式描述如下： $Y_i = G_1 \cdot G_{2A} \cdot G_{2B} \cdot m_i$

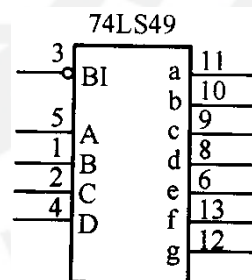
其中， Y_i 为内部输出编码字的第 i 位， m_i 为输入变量 C、B、A 的最小项。

• 74LS138 外部信号之间的关系为：

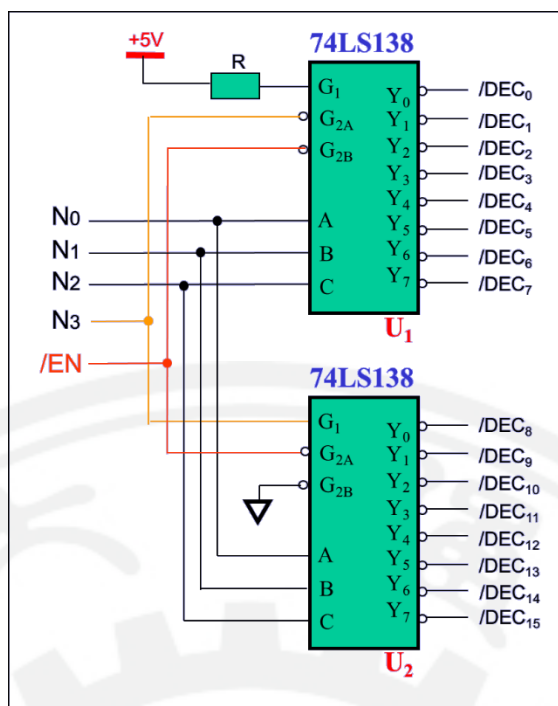
$$/Y_i = \overline{G_1} \cdot \overline{/G_{2A}} \cdot \overline{/G_{2B}} \cdot m_i$$

③ 七段译码器 74LS49

输 入					输 出						
$/BI$	D	C	B	A	a	b	c	d	e	f	g
0	d	d	d	d	0	0	0	0	0	0	0
0	1	0	0	0	1	1	1	1	1	1	0
1	1	0	0	0	0	1	1	0	0	0	0
2	1	0	0	1	1	1	0	1	1	0	1
3	1	0	0	1	1	1	1	0	0	0	1
4	1	0	1	0	0	1	1	0	0	1	1
5	1	0	1	0	1	0	1	1	0	1	1
6	1	0	1	1	0	0	1	1	1	1	1
7	1	0	1	1	1	1	1	0	0	0	0
8	1	1	0	0	0	1	1	1	1	1	1
9	1	1	0	0	1	1	1	0	0	1	1
1	1	0	1	0	0	0	0	1	1	0	1
1	1	0	1	1	0	0	1	1	0	0	1
1	1	1	0	0	0	1	0	0	0	1	1
1	1	1	0	1	1	0	0	1	0	1	1
1	1	1	1	0	0	0	0	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0



④ 二进制译码器级联

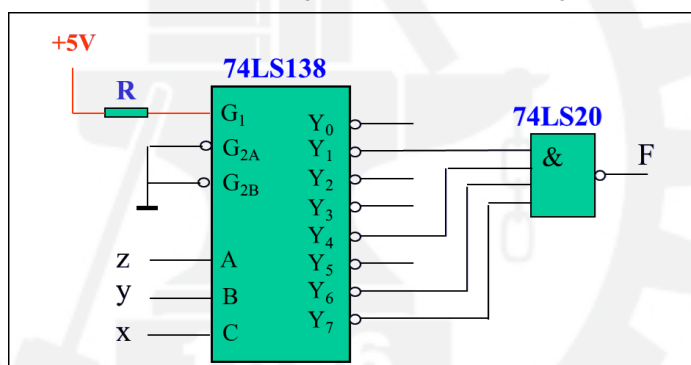


⑤ 用二进制译码器实现组合逻辑函数

二进制译码器的输出，对应于 n 变量函数的 2^n 个最小项。

例：用译码器 74LS138 实现 $F(x, y, z) = \sum m^3(1, 4, 6, 7)$

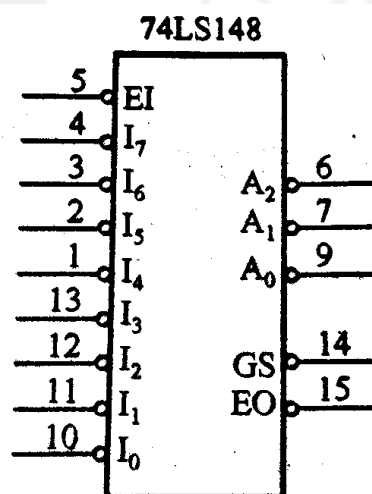
$$F = m_1 + m_4 + m_6 + m_7 = \overline{m_1} \cdot \overline{m_4} \cdot \overline{m_6} \cdot \overline{m_7}$$



(2) 编码器

① 优先权编码器 74LS148

输 入									输 出				
/EI	/I ₀	/I ₁	/I ₂	/I ₃	/I ₄	/I ₅	/I ₆	/I ₇	/A ₂	/A ₁	/A ₀	/GS	/EO
1	d	d	d	d	d	d	d	d	1	1	1	1	1
0	d	d	d	d	d	d	d	0	0	0	0	0	1
0	d	d	d	d	d	d	0	1	0	0	1	0	1
0	d	d	d	d	d	0	1	1	0	1	0	0	1
0	d	d	d	0	1	1	1	1	1	0	0	0	1
0	d	d	0	1	1	1	1	1	1	0	1	0	1
0	d	0	1	1	1	1	1	1	1	1	0	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



(3) 数据分配器

① 1~4 多路分配器

图中 I: 传送数据输入端;

A_1, A_0 : 地址码输入端;

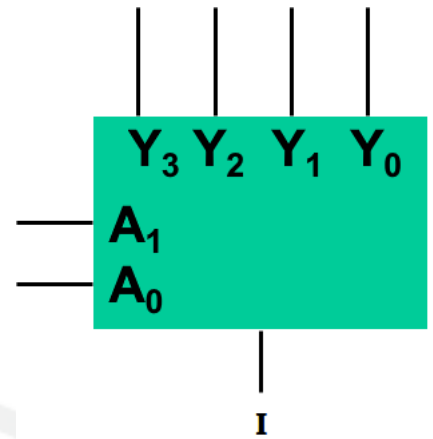
y_3, y_2, y_1, y_0 : 输出的数据通道。

这种分配器被称为“1~4 多路分配器”

一般表达式为: $y_i = I$ 其中 i 为地址码 $A_{n-1} \dots A_0$ 的十进制值

② 用二进制译码器作为数据分配器

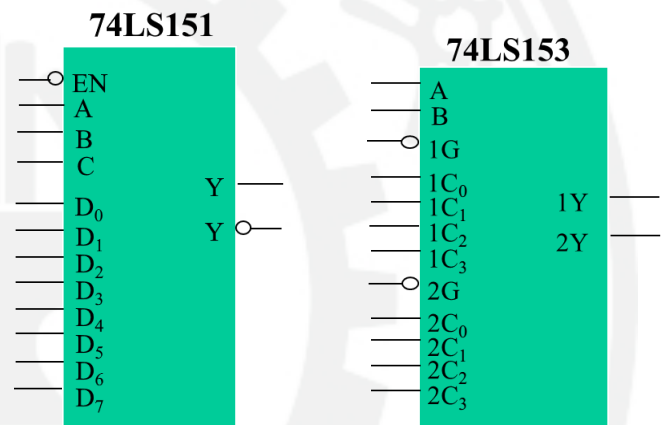
以某一使能端作为 I, 只能输出使能端有效时的数值



(4) 多路选择器

① 74LS151

输 入				输 出	
/EN	C	B	A	Y	/Y
1	d	d	d	0	1
0	0	0	0	D_0	D_0
0	0	0	1	D_1	D_1
0	0	1	0	D_2	D_2
0	0	1	1	D_3	D_3
0	1	0	0	D_4	D_4
0	1	0	1	D_5	D_5
0	1	1	0	D_6	D_6
0	1	1	1	D_7	D_7



② 74LS153

输 入				输 出	
/1G	/2G	B	A	1Y	2Y
1	d	d	d	0	0
0	0	0	0	$1C_0$	$2C_0$
0	0	0	1	$1C_1$	$2C_1$
0	1	0	0	$1C_2$	$2C_2$
0	1	0	1	$1C_3$	$2C_3$

③ 多路选择器的扩展 (见右图)

④ 用多路选择器实现任意组合逻辑函数

由于多路选择器的输出表达式为

$$KY = \sum_{i=0}^{n-1} EN \cdot m_i \cdot KD_i$$

若使 EN 有效且 $K = 1$, 则

$$Y = \sum_{i=0}^{n-1} m_i \cdot D_i$$

因此只要根据需要设计组合逻辑函数最小项表达式中的最小项来给定 D_i , 则任意 S 个变量的逻辑函数均可以用 2^S 输入的多路选择器实现。

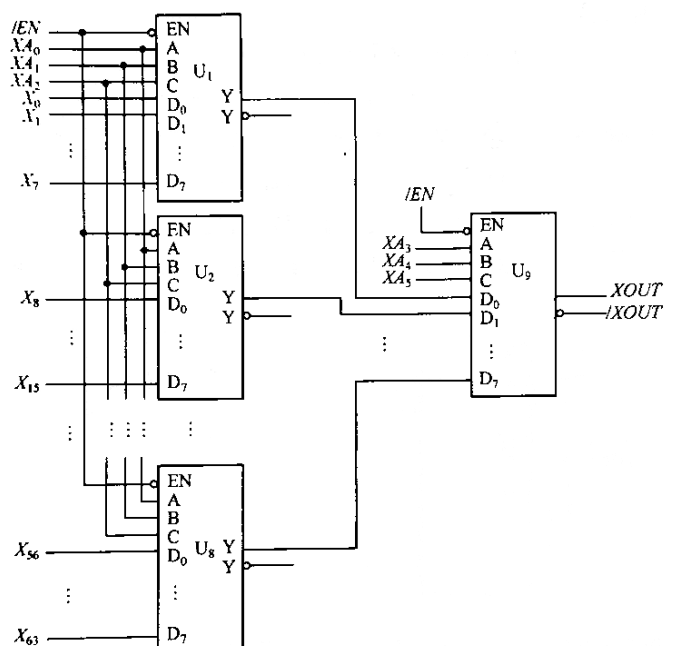


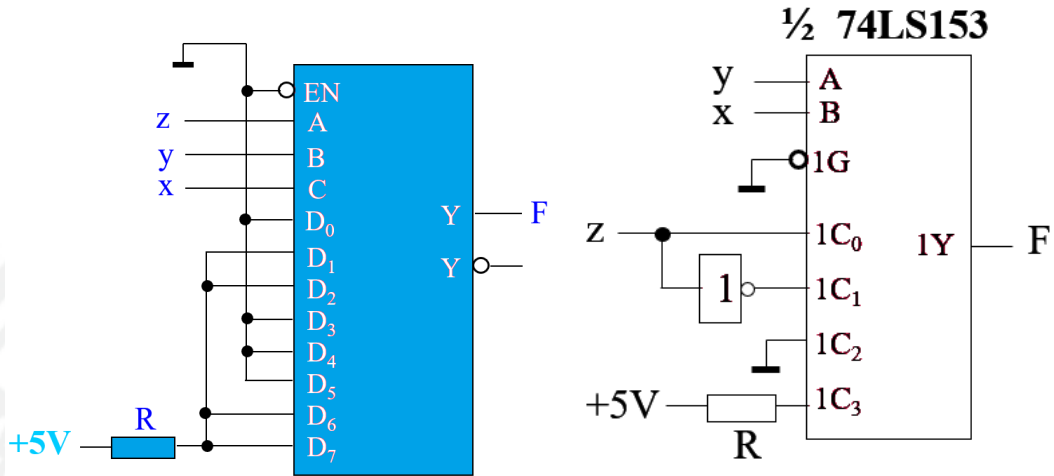
图 2.53 六十四选一数据选择器逻辑图

例 1: $F(x, y, z) = \sum m^3(1, 2, 6, 7)$

用 74LS151 实现(下左图): 令 $D_1 = D_2 = D_6 = D_7 = 1$

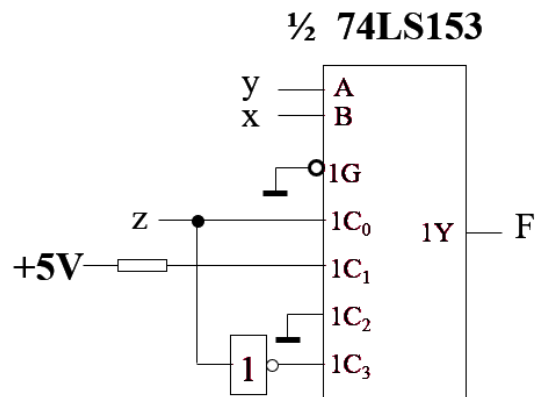
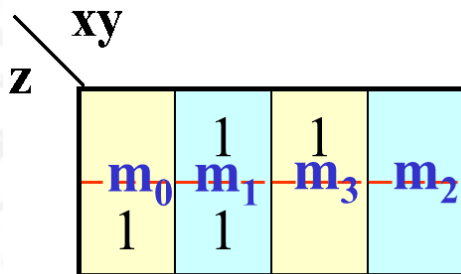
用 74LS153 实现(下右图): 改写函数

$$\begin{aligned} F(x, y, z) &= \sum m^3(1, 2, 6, 7) = \bar{x}\bar{y}z + \bar{x}y\bar{z} + xy\bar{z} + xyz = \bar{x}\bar{y}z + \bar{x}y\bar{z} + xy \\ &= (\bar{x}\bar{y}) \cdot z + (\bar{x}y) \cdot \bar{z} + (xy) \cdot 1 \end{aligned}$$



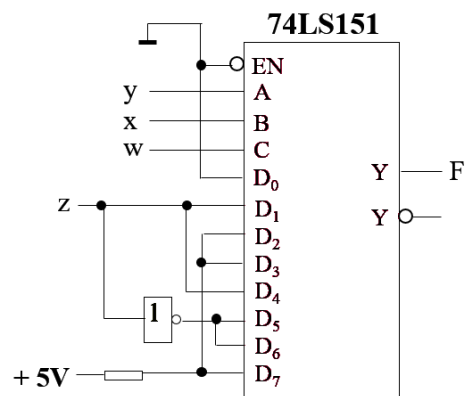
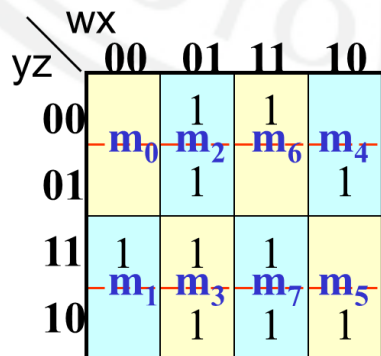
例 2: $F(x, y, z) = \sum m^3(1, 2, 3, 6)$

用 74LS153 实现。画卡诺图:



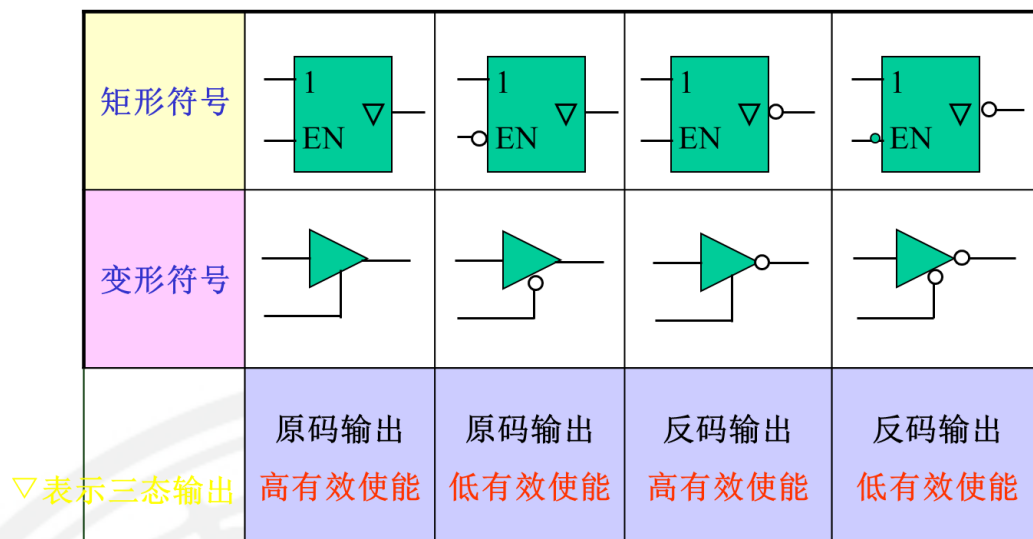
例 3: $F(w, x, y, z) = \sum m^4(3, 4, 5, 6, 7, 9, 10, 12, 14, 15)$

用 74LS151 实现:



(5) 三态缓冲器

① 三态缓冲器



② 74LS541

③ 74LS245

(6) 比较器

比较单元可用同或门/异或门实现

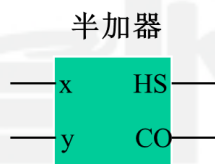
四位比较器 74LS85(见右图, 详细功能见书 P83)

(7) 加法器

① 半加器

$$HS = x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$$

$$CO = x \cdot y \text{ (输出进位)}$$

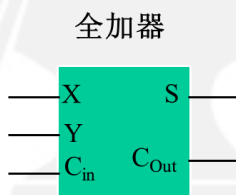


② 全加器

$$S = x \oplus y \oplus C_{in}$$

$$C_{out} = x \cdot y + x \cdot C_{in} + y \cdot C_{in}$$

如 74LS183 的一半



③ 74LS283

例子: 用 74LS283 实现 8421BCD

码到 2421 码的转换。

输入为 8421 码 $B_8B_4B_2B_1$, 输出为 2421 码

$Y_4Y_3Y_2Y_1$, 从真值表可以看出: $Y_4Y_3Y_2Y_1 = B_8B_4B_2B_1$

$+A_4A_3A_2A_1$

十进制数	$B_8B_4B_2B_1$	$Y_4Y_3Y_2Y_1$	$A_4A_3A_2A_1$
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 0
2	0 0 1 0	0 0 1 0	0 0 0 0
3	0 0 1 1	0 0 1 1	0 0 0 0
4	0 1 0 0	0 1 0 0	0 0 0 0
5	0 1 0 1	1 0 1 1	0 1 1 0
6	0 1 1 0	1 1 0 0	0 1 1 0
7	0 1 1 1	1 1 0 1	0 1 1 0
8	1 0 0 0	1 1 1 0	0 1 1 0
9	1 0 0 1	1 1 1 1	0 1 1 0

B_8B_4	B_2B_1	A_4	A_3
00	00	d	1
01	00	1	d
10	00	1	d
11	00	1	d

由表可知 $A_4=A_1=0$ $A_3=A_2$, 又画 A_2 的卡诺图如右上, 可得:

$$A_2 = B_8 + B_4B_2 + B_4B_1$$

故最终可画出电路如右。

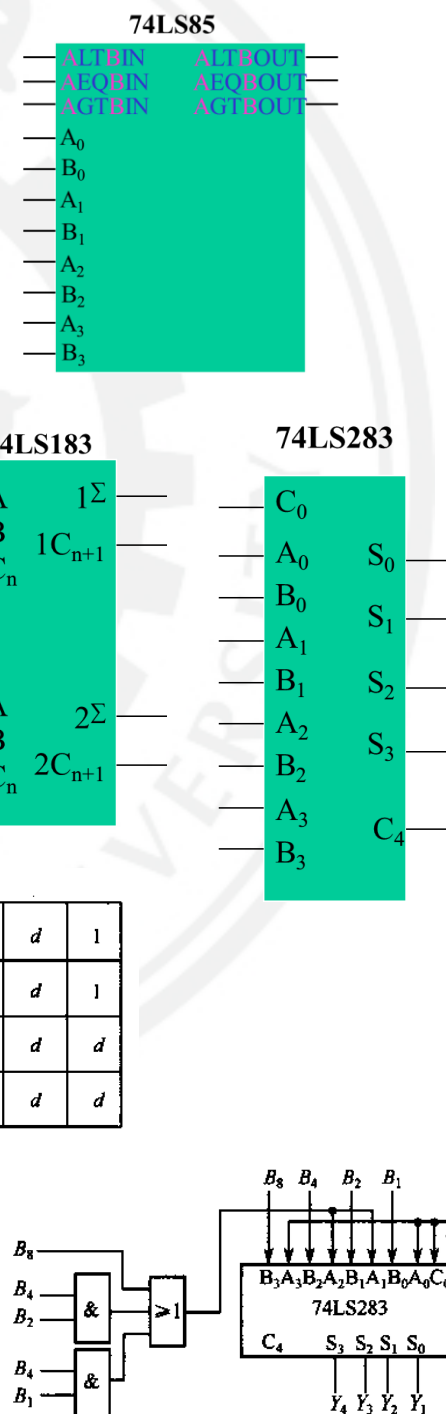


图 2.80 8421 码转换为 2421 码的电路

第三章 时序逻辑电路

1. 相关概念

- 输出函数 $z_i = f_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_l)$ $i = 1, 2, \dots, m$;
- 激励（控制）函数 $Y_i = g_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_l)$ $i = 1, 2, \dots, r$;
- 次态 $y^{n+1} = Q$ (输入 x , 现态 y)
- 同步时序电路：其状态的改变受同一个时钟脉冲的控制，且与时钟脉冲同步。即电路在统一时钟控制 CLK（或 CP）下，同步改变状态。
- 异步时序电路：无统一的时钟脉冲使整个系统的工作同步，输入直接引起状态改变。
- Mealy 型时序电路输出与现态、与输入直接相关；Moore 型电路与现态、输入相关，但与输入没有直接相关。

2. 时序电路的描述

(1) 次态方程

次态 = Q(输入, 现态)

可根据画出次态卡诺图

(2) 功能表

(3) 次态真值表

(4) 状态表

(5) 状态图

3. 时序电路的双稳态元件

(1) 锁存器

① S-R 锁存器

次态真值表：

S	R	Q	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	d
1	1	1	d

次态方程 $Q^{n+1} = S + \bar{R} \cdot Q$

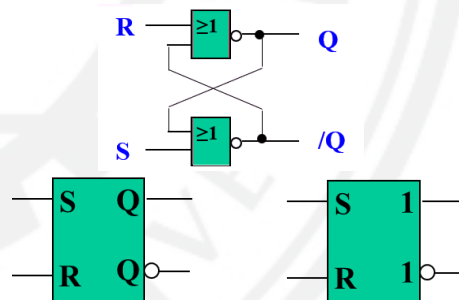
约束条件： $S \cdot R = 0$

简化次态真值表：

S	R	Q^{n+1}
0	0	Q
0	1	0
1	0	1
1	1	d

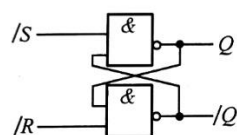
卡诺图：

SR	00	01	11	10
Q				
0	0	0	d	1
1	1	0	d	1



② /S-/R 锁存器

次态方程 $Q^{n+1} = \bar{S} + /R \cdot Q$ 约束条件： $\bar{S} \cdot \bar{R} = 0$



(a) 一对与非门组成/S-/R锁存器

/S	/R	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	保持	不变

(b) 功能表

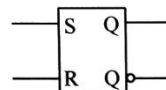
/S	/R	Q^{n+1}
0	0	d
0	1	1
1	0	0
1	1	Q

(c) 简化次态真值表

/S/R	00	01	11	10
Q				
0	d	1	0	0
1	d	1	1	0

$Q^{n+1} = \bar{S} + /R \cdot Q$ 约束条件： $\bar{S} \cdot \bar{R} = 0$

(d) 次态卡诺图

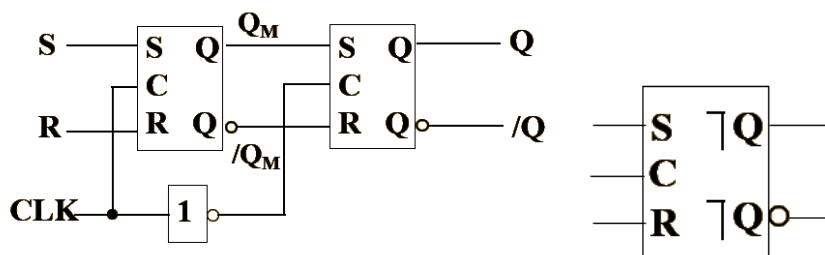


(e) 逻辑符号

图 3.6 /S-/R 锁存器工作过程

(2) 触发器

① 主从 S-R 触发器和主从 J-K 触发器



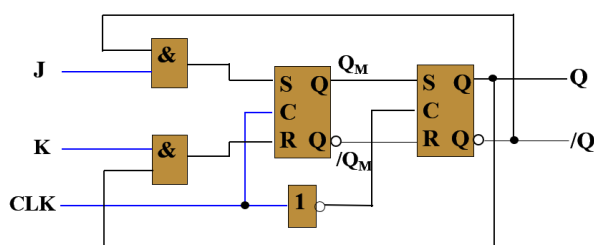
CLK=1: 主锁存器状态方程为 $Q_M^{n+1} = S + \bar{R}Q_M$

CLK=0: 从锁存器状态方程为 $Q^{n+1} = Q_M^{n+1} + \overline{Q_M^{n+1}}Q = Q_M^{n+1}$

则有 $Q^{n+1} = Q_M^{n+1} = S + \bar{R}Q_M$

又由于 $Q = Q_M$

则 $Q^{n+1} = S + \bar{R}Q$



(a) 用S-R 锁存器构成的JK触发器

J	K	S	R	Q^{n+1}
0	0	0	0	Q
0	1	0	Q	0
1	0	/Q	0	1
1	1	/Q	Q	/Q

J	K	C	Q	/Q
d	d	0	保持不变	保持不变
0	0	1	保持不变	保持不变
0	1	1	0	1
1	0	1	1	0
1	1	1	变反	变反

J	K	Q^{n+1}
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

次态方程 $Q^{n+1} = J \cdot \bar{Q} + \bar{K} \cdot Q$

② 边沿触发的 D 触发器

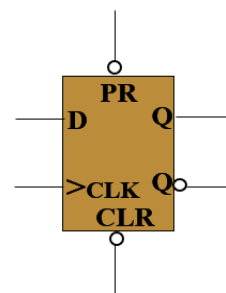
前沿触发:

功能表:

D	CLK	Q	/Q
0	↑	0	1
1	↑	1	0
d	0	保持不变	保持不变
d	1	保持不变	保持不变

次态真值表: $Q^{n+1} = D$

D	Q^{n+1}
0	0
1	1



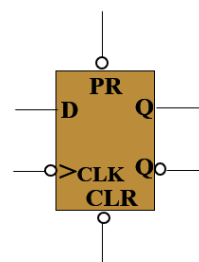
后沿触发:

功能表:

D	CLK	Q	/Q
0	↓	0	1
1	↓	1	0
d	0	保持不变	保持不变
d	1	保持不变	保持不变

次态真值表: $Q^{n+1} = D$

D	Q^{n+1}
0	0
1	1



D 触发器激励表：

Q	Q ⁿ⁺¹	D
0	0	0
0	1	1
1	0	0
1	1	1

③ 边沿触发 J-K 触发器

功能表：

J	K	CLK	Q /Q
d	d	0	保持不变
d	d	1	保持不变
0	0	┐	保持不变
0	1	┐	0 1
1	0	┐	1 0
1	1	┐	变反

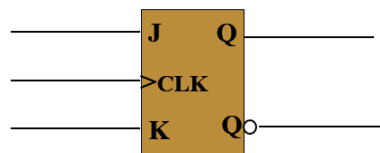
次态真值表：

J	K	Q ⁿ⁺¹
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

次态方程： $Q^{n+1} = J \cdot \bar{Q} + \bar{K} \cdot Q$

激励表：

Q	Q ⁿ⁺¹	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0



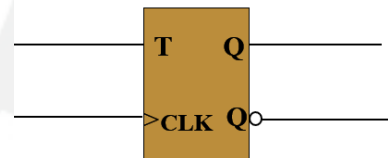
④ T 触发器

功能表：

T	CLK	Q /Q
d	0	保持不变
d	1	保持不变
0	d	保持不变
1	┐	变反

次态真值表：

T	Q	Q ⁿ⁺¹
0	0	0
0	1	1
1	0	1
1	1	0



简化次态真值表：(次态方程 $Q^{n+1} = T \cdot \bar{Q} + \bar{T} \cdot Q$)

T	Q ⁿ⁺¹
0	Q
1	\bar{Q}

激励表：

Q	Q ⁿ⁺¹	T
0	0	0
0	1	1
1	0	1
1	1	0

4. 同步时序电路分析

- ① 标注现态和激励信号，列出激励函数和输出函数表达式
- ② 根据触发器的次态方程得到各个状态变量的次态方程

- ③ 根据状态变量的次态方程填写二进制状态表。
- ④ 根据输出表达式填写输出值，得到二进制状态输出表。
- ⑤ 每一个状态分配一个字母状态名，从而得到状态输出表。
- ⑥ 根据状态输出表，画出状态图。
- ⑦ 电路特性描述，确定电路的逻辑功能。(必要时可画时序图)

5. 同步时序电路设计

- ① 建立**原始状态图**和**状态表**
- ② 状态化简求得**最简状态表**

状态等效：两个状态加任意输入序列，产生的输出序列完全一致

等效类：包含所有可以互相构成等效对的状态集合

判断条件：

对每一组可能的输入，满足：

- 输出完全相同
- 次态满足下列情况之一：
 - 状态相同；状态交错；状态维持；后继状态等效；次态循环

常用方法：隐含表法

- 完全给定同步时序电路状态表化简
 - a) 画隐含表格
 - b) 顺序比较
 - ✧ 相同输入，输出不同，不等效，打×
 - ✧ 相同输入，输出相同，次态相同、交错、维持，等效，打√
 - ✧ 相同输入，输出相同，次态不为相同、交错、维持，需进一步确定后继状态对等效关系，填入次态的状态对名
 - c) 关联比较

检查 b) 步骤中的遗留问题，主要检查后继状态相等和次态循环的情况，如(A D)→(B E)→(A D)
 - d) 列出最大等效类

确定全部等效对，由等效传递性得到等效类，再获得最大等效类
 - e) 重命名状态，列出最小化状态表
- 不完全给定同步时序电路状态表化简

利用隐含表寻找相容对、利用状态合并图确定最大相容类、采用覆盖闭合表进行相容类的选择，从而建立最小化状态表。

✧ **状态相容：**

S_1 和 S_2 是不完全给定时序电路 M_1 和 M_2 (M_1 和 M_2 可以是同一个电路) 的两个状态，作为初态同时加入预定的允许输入序列(加入该序列后，电路除最后一个次态外，其他次态都是确定的)，所产生的输出序列一致(认为确定的输出与对应的不确定输出相同)，则状态 S_1 和 S_2 是相容的，称 S_1 和 S_2 是相容对。记为：(S_1, S_2)

✧ **状态相容反映在状态表上：**

两个状态的确定部分相同，两个状态中的无关项按确定的部分取同一值，则这两个状态就变为等效状态。因此相容状态有可能变为等效，从而导致合并。

✧ **状态相容无传递性。**

✧ **相容对的判别标准**

条件一：它们的输出相同；

条件二：它们的次态必须满足下列情况之一：

次态相同；次态交错；次态维持；后继状态相容；次态循环

注意：一方给定，一方不给定的次态均当作相同。

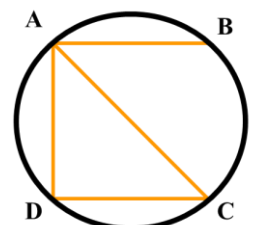
✧ **合并状态：**

右表中的相容对为：

(A,B), (A,C), (A,D), (C,D)

将所有相容对填入合并图，顶点之间都有连线的多边形就构成了“相容类”。

	0	1
A	A/0	D/d
B	A/0	D/0
C	A/0	D/1
D	A/0	C/1



可以得到两个最大相容类为:(A,B), (A,C,D)

✧ 作出最小化状态表

选择的相容类集需满足:

- 覆盖: 包含全部原始状态
- 闭合: 次态应仅属于一个相容类
- 最小化: 数目最少

一个例子:

	0	1							
			覆盖					闭合	
			A	B	C	D	E	X=0	X=1
A	A/d	d/d	ABD	A	B		D	AC	B
B	C/1	B/0		A		C	D	AD	B
C	D/0	d/1		A		C		AD	C
D	d/d	B/d	ACD						
E	A/0	C/1	ACE						

选择 ABD 及 ACE

③ 状态分配求得二进制状态表

如果触发器个数为 K , 有 2^K 种二进制组合, 用来对 n 个状态进行分配时就有 N 种独立的分配方案:

$$N = \frac{(2^K - 1)!}{(2^K - 1)! K!}$$

相邻状态分配法:

基本思想: 尽可能使次态和输出函数在卡诺图上“1”单元的分布为相邻, 以便形成较大的卡诺圈, 从而得到最简的次态和输出函数表达式。

主要规则:

规则 I: 在相同输入条件下, 次态相同, 现态相邻。(列相邻)

在有 K 个变量(触发器)的情况下, 如果满足规则 I 一次, 则可保证 K 个次态函数卡诺图中各有一对 1 单元(或 0 单元)列向相邻。若满足 R 次意味着可保证次态函数卡诺图上有 $K \times R$ 对“1”或“0”相邻, 记为: 改善效果 $I = K \times R$

规则 II: 在相邻的输入条件下, 同一现态, 次态相邻。(行相邻)

在有 K 个变量(触发器)的情况下, 如果满足规则 II 一次, 则可保证 $(K - 1)$ 个次态函数卡诺图中各有一对 1 单元(或 0 单元)行向相邻。若满足 m 次意味着可保证次态函数卡诺图上有 $(K - 1) \times m$ 对“1”或“0”相邻, 记为: 改善效果 $II = (K - 1) \times m$

规则 III: 输出完全相同, 现态相邻。

在有 p 个输入组合、 q 个输出的情况下, 如果满足规则 III 一次, 则可保证 q 个输出函数卡诺图中各有 p 对 1 单元(或 0 单元)列向相邻。若满足 l 次意味着可保证输出函数卡诺图上有 $(p \times q) \times l$ 对“1”或“0”相邻, 记为: 改善效果 $III = (p \times q) \times l$

满足状态 S_1 、 S_2 相邻要求的总改善效果为: (即两个“1”的相邻数与两个“0”的相邻数之和)

$$E_{S_1 S_2} = \text{改善效果 I} + \text{改善效果 II} + \text{改善效果 III} = K \times R + (K - 1) \times m + (p \times q) \times l$$

- 一个例子:

状态数为 n , 则触发器个数 $K = \lceil \log_2 n \rceil$, 输入组合数 p , 输出函数位数 q

例1 完成如图所示状态表的状态分配。

$$K = 2; p = 2; q = 1.$$

总改善效果为:

$$E_{SIS2} = K \times R + (K-1) \times m + (p \times q) \times l \\ = 2R + m + 2l$$

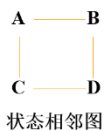
分析状态表, 求得:

- ①根据规则I: $R_{AB} = 1, R_{AC} = 1$ $E_{AB} = 2R_{AB} + m_{AB} + 2l_{AB} = 5$
- ②根据规则II: $m_{CD} = 1, m_{AC} = 1$ $E_{AC} = 2R_{AC} + m_{AC} + 2l_{AC} = 5$
- ③根据规则III: $l_{AB} = 1, l_{AC} = 1$ $E_{CD} = m_{CD} = 1$
 $l_{BC} = 1$ $E_{BD} = m_{BD} = 1$
 $E_{BC} = 2l_{BC} = 2$

$y \backslash x$	0	1
A	C/0	D/0
B	C/0	A/0
C	B/0	D/0
D	A/0	B/1

y^{n+1}/Z

例1 完成如图所示状态表的状态分配。



$y_1 \backslash y_0$	0	1
0	A	C
1	B	D

状态分配

$y \backslash x$	0	1
A	C/0	D/0
B	C/0	A/0
C	B/0	D/0
D	A/0	B/1

y^{n+1}/Z

$y_1 y_0 \backslash x$	0	1
A 00	10/0	11/0
B 01	10/0	00/0
D 11	00/0	01/1
C 10	01/0	11/0

二进制状态表

$$E_{AB} = 2R_{AB} + m_{AB} + 2l_{AB} = 5$$

$$E_{AC} = 2R_{AC} + m_{AC} + 2l_{AC} = 5$$

$$E_{CD} = m_{CD} = 1$$

$$E_{BD} = m_{BD} = 1$$

$$E_{BC} = 2l_{BC} = 2$$

- ④ 选择触发器, 确定激励函数和输出函数
- ⑤ 画出逻辑电路图
- ⑥ 分析解决挂起问题
- 任意打破无效循环, 强行将无关项赋值, 再求逻辑函数

6. 脉冲异步时序电路的分析

异步时序电路: 电路状态改变由输入信号变化直接引起

特点: 没有统一的同步时钟脉冲, 电路状态的改变是由输入信号的变化直接引起的。

按输入信号的特征分为: 脉冲型与电平型。

◇ 异步时序电路的特点:

- 电路状态的改变直接依赖于输入脉冲。即输入脉冲和原状态生成的激励函数 (包括时钟脉冲) 使电路翻转到预定的新状态。
- 新状态建立后, 输入脉冲消失, 电路仍保持在该状态, 直至下一个输入脉冲到达时, 电路状态才发生变化。
- 所有触发器的时钟信号也是由组合电路产生, 且不一定同时到来, 也呈无规律性

◇ 脉冲异步时序电路与同步时序电路的异同

- 相同点:
 - (1) 状态的改变都依赖于外加脉冲。
 - (2) 存储元件都是触发器 (或锁存器)。
- 差异:
 - (1) 脉冲异步时序电路无外加的统一的时钟脉冲。
 - (2) 输入变量为脉冲信号, 由输入脉冲直接引起电路的状态改变。
 - (3) 由次态逻辑产生各触发器控制输入信号(Y_1, Y_2, \dots, Y_r), 而且还产生时间有先后的各触发器的时钟控制信号($CLK_1, CLK_2, \dots, CLK_r$)。

◇ 脉冲异步时序电路输入的限制:

- (1) 每个外部输入脉冲加入时, 电路中所有的触发器均发生现态到次态的转换。如果其中触发器的时钟端无时钟脉冲, 则认为该触发器的次态等于现态。
为了使电路可靠工作, 电路状态变化可预知, 对脉冲异步时序电路的输入作如下规定。
- (2) 不允许两根或两根以上输入线上同时有输入脉冲。
- (3) 在上一个输入脉冲引起的电路状态变化未稳定以前, 不允许加入新的输入脉冲。

◇ 关于 CLK:

- 在同步时序电路中, 每次电路状态改变时, 时钟脉冲总是同时送到各个触发器, 故在触发器的次态方程中, 可以认为时钟脉冲 $CLK = 1$ (有脉冲), 而不必写出。
- 而在异步时序电路中, 每次电路状态改变时, 时钟脉冲并不一定送到各个触发器。或者说各个触发器只有在其相应的 CLK_i 到达时, 其状态才发生变化。因此, 这种异步时序电路的次态方程中必须反映出是否有 CLK_i 信号, 或者说, 必须将 CLK_i 信号也作为逻辑变量写入次态方程。

◇ 脉冲异步时序电路的分析步骤基本上与同步电路一样, 仅作以下修改:

- (1) 输入变量取值为 1 表示有脉冲信号, 取值为 0 表示无脉冲信号。触发器的时钟输入端也按上述规定。
- (2) 控制函数包括触发器的控制输入(Y_1, Y_2, \dots, Y_r)及触发器的时钟输入($CLK_1, CLK_2, \dots, CLK_r$)。
- (3) 两个或两个以上的输入变量不能同时为 1; 输入变量全为 0 时, 电路状态不变。

✧ 分析步骤:

- ① 列出函数和控制函数 CLK 的表达式
- ② 列出状态真值表和次态真值表
- ③ 画出状态表和状态图
- ④ 电路功能说明

7. 常用时序逻辑器件

(1) 二进制同步计数器

由 D 触发器构成的 n 位二进制同步加 1 计数器的连接规律:

$$D_i = (Q_{i-1} \cdot Q_{i-2} \cdot \dots \cdot Q_1) \oplus Q_i \quad (i \neq 1)$$

$$D_1 = \overline{Q_1}$$

其中: $Q_{i-1} \cdot Q_{i-2} \cdot \dots \cdot Q_1$ ——进位传递函数

由 D 触发器构成的 n 位二进制同步减 1 计数器的第 i 位触发器的次态方程:

$$Q_i^{n+1} = (\overline{Q_{i-1}} \overline{Q_{i-2}} \dots \overline{Q_1}) \oplus Q_i \quad (i \neq 1)$$

$$Q_1^{n+1} = \overline{Q_1}$$

(2) 二进制串行计数器

二进制串行计数器连接规律

加 1 计数器:

前沿触发的触发器: $CLK_i = \overline{Q_{i-1}}$

后沿触发的触发器: $CLK_i = Q_{i-1}$

最低位的触发器: $CLK_1 = CLK$ (外加计数脉冲)

减 1 计数器:

前沿触发的触发器: $CLK_i = Q_{i-1}$

后沿触发的触发器: $CLK_i = \overline{Q_{i-1}}$

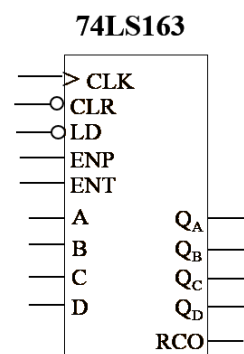
最低位的触发器: $CLK_1 = CLK$ (外加计数脉冲)

(3) 74LS163

具有加载(LD)和清除(CLR)输入端:

- 当 /CLR 输入有效, 输出为 0;
- 当 /LD 有效而 /CLR 无效, 输入 A,B,C,D 被送到输出端;
- 如果 /CLR 和 /LD 都无效, 而使能输入端 ENP、ENT 都有效, 74LS163 进行加 1 计数。

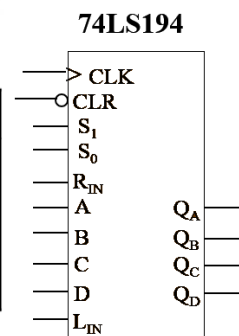
74LS169 还有 UP/DN 端, 决定加一计数或减一计数。



(4) 74LS194(移位寄存器)

- RIN 是右移串行输入端;
- LIN 是左移串行输入端;
- A、B、C、D 是并行输入端;
- QD 是右移串行输出端;
- QA 是左移串行输出端;
- QA、QB、QC、QD 是并行输出端;
- /CLR 是异步的寄存器清“0”信号;
- S1S0 是工作方式控制。

功能	$S_1 S_0$	Q_A^{n+1}	Q_B^{n+1}	Q_C^{n+1}	Q_D^{n+1}
保持	0 0	Q_A	Q_B	Q_C	Q_D
右移	0 1	R_{IN}	Q_A	Q_B	Q_C
左移	1 0	Q_B	Q_C	Q_D	L_{IN}
置数	1 1	A	B	C	D



(5) 施密特触发器

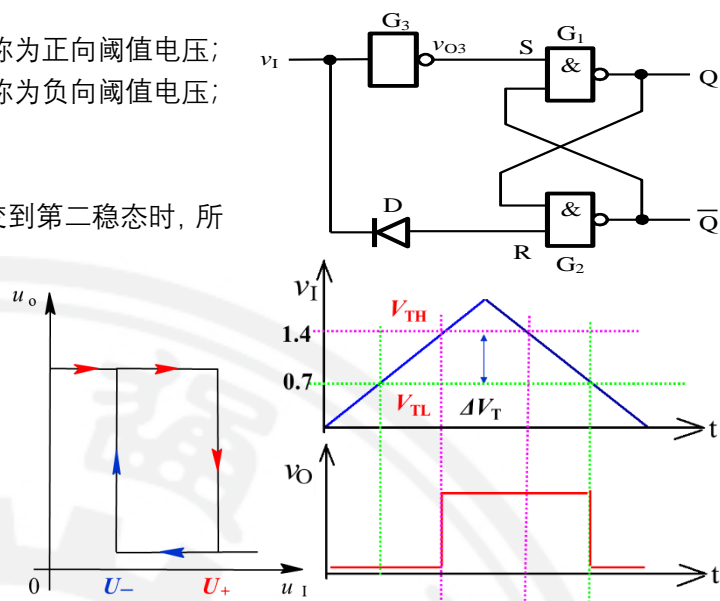
工作特性:

- ① 具有两个稳态;
- ② 属于电平触发, 缓慢变化的信号也可以作为输入信号, 当输入信号达到某一特定值时, 输出电平就发生突变;

③ 输入信号从低电平上升时，电路状态转换时对应的输入电平，与输入信号从高电平下降时对应的输入转换电平不同。

电路参数：

- 输入信号上升时对应的转换电平 U_+ ，称为正向阈值电压；
- 输入信号下降时对应的转换电平 U_- ，称为负向阈值电压；
- 差值 $\Delta U = U_+ - U_-$ ，称为回差电压
- 上限阈值电压 V_{TH}
 v_I 上升过程中，输出电压由第一稳态跳变到第二稳态时，所对应的输入电压值。（ $V_{TH}=1.4V$ ）
- 下限阈值电压 V_{TL}
 v_I 下降过程中，输出电压由第二稳态跳变到第一稳态时，所对应的输入电压值。（ $V_{TL}=0.7V$ ）
- 回差电压 ΔV_T
 $\Delta V_T = V_{TH} - V_{TL} = 0.7V$



第四章 可编程逻辑器件

PLD 是由工厂制好，不需要定制任何掩模，用户可以利用开发工具，对芯片功能进行编程的大规模集成电路器件。

- 分类：

SPLD(PROM,PLA,PAL,GAL)、CPLD(ISP,FPGA)

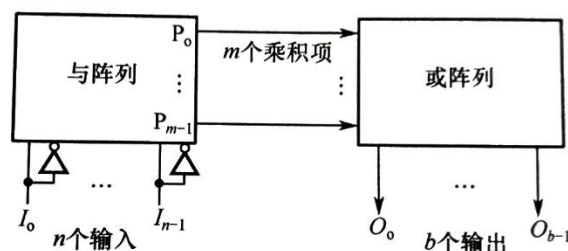
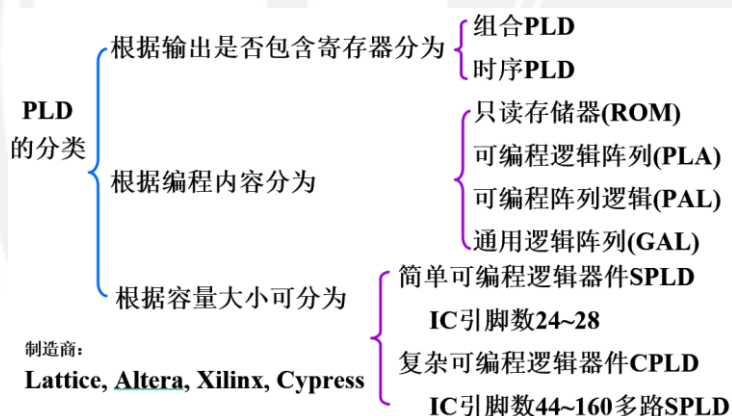


图 4.1 PLD 的基本结构

只读存储器 (ROM): 与阵列固定 或阵列可编程

可编程逻辑阵列 (PLA): 与阵列可编程 或阵列可编程

可编程阵列逻辑 (PAL): 与阵列可编程 或阵列固定

通用逻辑阵列(GAL): 兼容 PAL, 增加可擦除、可重新编程及可组态结构等特点。使用最广泛的 PLD 产品之一。

- 设计过程：设计输入、设计验证、设计实现

- 优点：

- 简化系统设计（满足各种使用要求，最有效地利用芯片，减少芯片数量；能快速进行设计。）
- 功能集成度高（高于中小规模集成电路，且具有更高的利用率。）
- 可靠性高

1. 可编程只读存储器 (PROM)

与阵列**固定**、或阵列**可编程**

分类：一次可编程 PROM、多次可编程 PROM（电可擦除 E²PROM 光可擦除 EPROM）

2. 可编程逻辑阵列 (PLA)

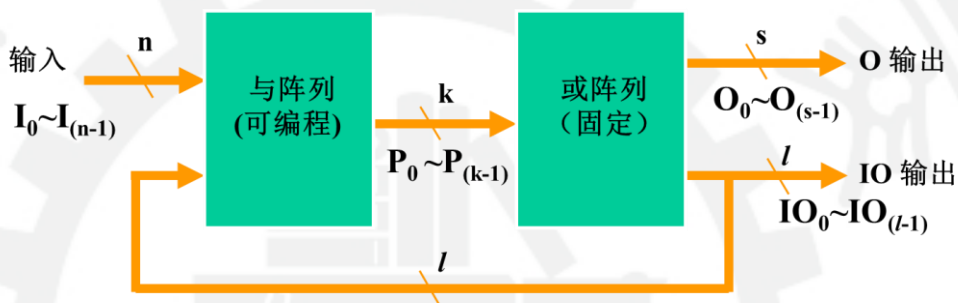
特点：与、或阵列都可编程

- 针对逻辑函数的最简与或式——
 - PLA 中的与阵列被编程产生所需的全部与项
 - PLA 中的或阵列被编程完成相应与项间的或运算，并产生输出。逻辑功能越复杂，其优点越明显。这样，就大大提高了芯片面积的有效利用率。
- PLA 分组合 PLA 和时序 PLA(包含有触发器)。

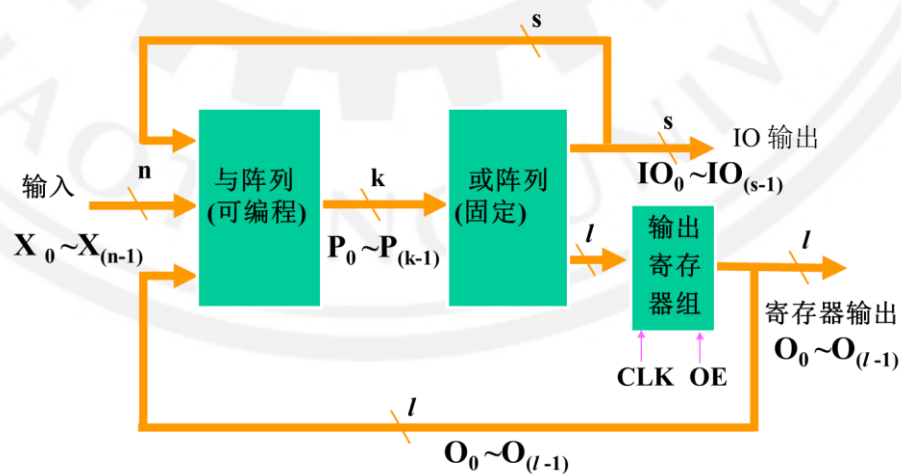
3. 可编程阵列逻辑 (PAL)

特点:

- 可编程的与阵列和固定的或阵列
 - 与同样位数的 PLA 相比，PAL 减少了编程点数(或阵列固定)，简化了编程工作(仅对与阵列编程)。
 - PAL 器件内所提供的与项数目较少，通常为 7~8 个，但是典型的逻辑设计中一般只需要 4~5 个与项。
 - 这种结构也提供了较高的性能和速度，更有利于辅助设计系统的开发，所以在较长时间里 PAL 成为了 PLD 发展史上的主流。
- 这样，就更有利于辅助设计系统的开发。
- PAL 分组合 PAL 和时序 PAL (包含有触发器)。
 - 组合 PAL 结构:



- ◇ 组合 PAL 器件的基本结构如上图所示,输出不带有寄存器,输出引脚分为两类:纯组合 O 输出和 IO 输出,每一个 O 输出或 IO 输出都对应一个独立的可编程与门阵列和固定的或门阵列。图中应有 $s+l$ 个独立的与或阵列。
- ◇ 每个可编程与阵列可根据 $I_{[0 \sim (n-1)]}$ 和 $IO_{[0 \sim (l-1)]}$ 生成一组任意的输出 $P_{[0 \sim (k-1)]}$,与项通常为 7~8 个。每组与阵列的输出被固定连接到一个或阵列的输入端。因此,或阵列之间不能共享与门的输出。如果用 PAL 器件实现某一逻辑功能时,若有两个输出或门有相同的与项输入,则必须由相应的两个与阵列产生两个相同的与项。
- ◇ 时序 PAL 结构:



4. 通用逻辑阵列 (GAL)

(1) GAL 器件的主要特点

与阵列可编程，或阵列固定

① 工艺上的改进

高速电可擦除 CMOS(Electrically Erasable Complementary Metal-Oxide Semiconductor (E²CMOS))

特点:

- (1) 可测试性; (最大优势之一, 直接测试各种特性, 最适合于样机的研制)
- (2) 低功耗, 使集成度更高; (CMOS)
- (3) 速度不低于其他 TTL 可编程器件
- (4) 可重复编程 100 次以上

② 结构上的改进 通用性→灵活性

(1) 每个输出端增加了一个逻辑输出宏单元(OLMC—Output Logic Macro Cell): 允许设计者以编程的方式确定每一个 OLMC 的组态和功能, 设计者可以在每一个输出端上“随意”实现所要求的功能和结构。

(2) 加密: 器件内增加了可被编程的保密位, 以防对逻辑的复制。

制作者可以大量生产少数几个型号的 GAL 器件, 从而进一步降低成本; 也使得设计者简化了选择器件的过程, 减少了数量与体积, 降低了成本, 提高了可靠性和稳定性。

(2) 基本结构

(1) 有 8 个输入缓冲器 (第 2~9 管脚) 和 8 个反馈缓冲器, 它们的输出作为与阵列的输入 (与阵列的 32 条列线)。

(2) 与阵列有 64 个乘积项输出, PT0~PT63 (标有数字的行线), 64 行×32 列=2048 个可编程单元构成与阵列。

(3) 有 8 个输出逻辑宏单元 (第 12~19 管脚)

(4) 1 个时钟输入端 (第 1 脚) 和 1 个三态使能输入端 OE (第 11 脚), 它们也可作为数据输入端。

(5) 5 V 电源端 (第 20 脚) 和接地端 (第 10 脚), 图中未画出

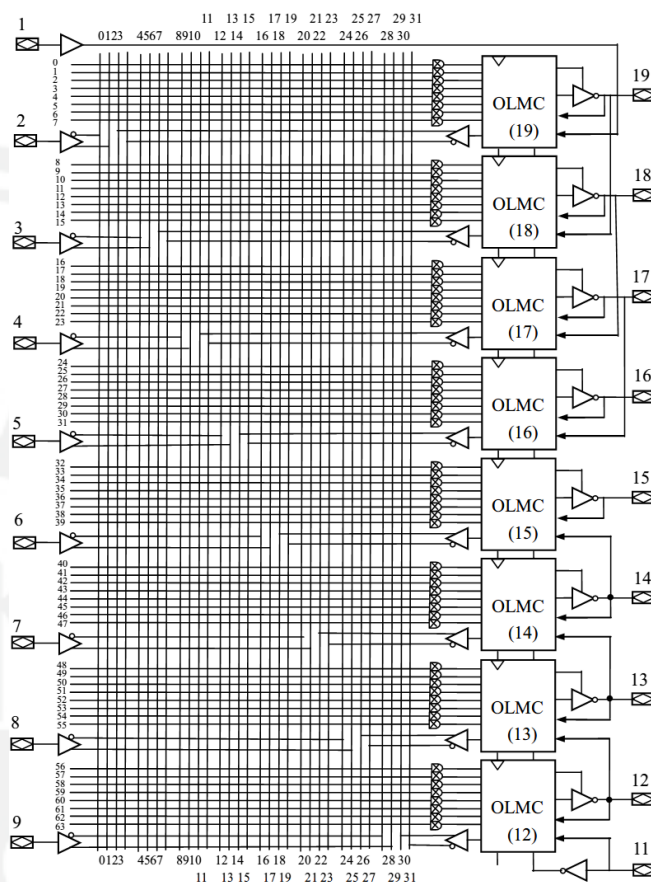
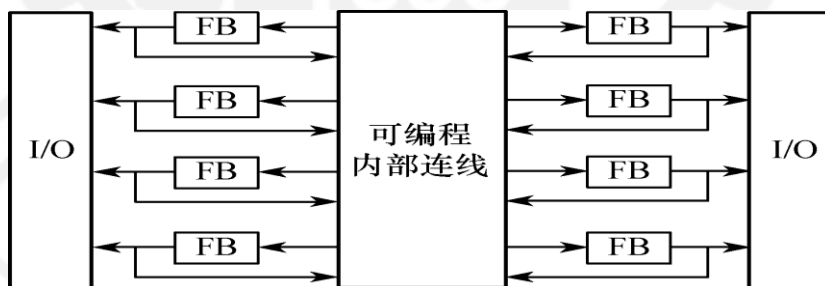


图4.17 GAL16V8

5. CPLD

CPLD 器件中至少包含 3 种结构: 可编程逻辑功能块 FB、可编程 I/O 单元、可编程内部连线 PIA。可编程逻辑功能块 FB 中包含有乘积项、宏单元等。



6. 在系统可编程器件 ISP

ispLSI 公司发明的 ISP, 称为在系统可编程逻辑器件。

ISP 是用户具有在自己设计的目标系统中或线路板上, 为重构逻辑而对逻辑器件进行编程或反复改写的能力。(先装配后编程)

ISP 器件的出现, 从实践上全面实现了硬件设计与修改的软件化, 使得数字系统的设计变得像软件一样易于修改。

◇ 在系统编程原理

ISP 技术的特点:

不用编程器, 用户直接在自己设计的目标系统中或线路板上对 ISP 器件进行编程。

可以先装配后编程, 成为产品后还可以反复编程。

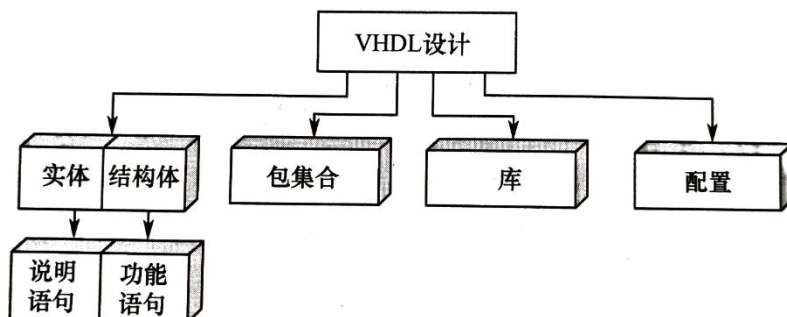
7. VHDL 基本知识点

(1) 信号的含义和信号的两种最常用数据类型: std_logic 和 std_logic_vector。

- (2) 5 种常用语句:说明语句、赋值语句、if 语句、case 语句和 process 语句。
- (3) 实体(Entity)、结构体(Atchitecture)、一个实体和结构体对构成了设计实体。
- (4) 库(Library) 和程序包(Package) 的基本使用方法。
- (5) 层次结构的设计要掌握元件 (Component)语句和端口映射(Port map)语句。

8. VHDL 程序基本结构

一个 VHDL 程序基本结构如图 4.39 所示, 它包含实体(Entity)、结构体(Architecture)、包集合(Package)、库(Library)、配置(Configuration) 5 个部分。



实体用于描述所设计的系统的外部接口信号。结构体用于描述系统内部的结构和行为；

包集合用于存放各种设计模块能共享的数据类型、常数、程序等。库用于存放已编译的实体、结构体、包集合、配置。有两种：一种是由用户自行生成的库；另一种是 PLD 制造商提供的库，用户可直接引用，不必从头编写。配置用来从库中选取所需单元来组成新系统。

任何一个 VHDL 程序中两个必需的元素是实体和结构体，且它们必须同时使用。实体(Entity)通过被称为端口(PORT)的外部输入和输出来描述一个给定的逻辑功能，而结构体(Architecture) 用来描述系统内部的结构和行为。

结构体功能描述的方法有 3 种:①数据流描述;②结构描述;③行为描述。不同的描述方式只体现在描述语句上，而结构体的结构是完全一样的。

第五章 数字系统

数字系统是由一组或几种基本的标准逻辑门构成的复杂的、使用数字量来传递、加工和处理信息的实际工程系统。数字系统能够进行多种信息处理，如采集、传输、加工处理和使用等。常见的数字系统包括数字计算机、数字手机、数字音像设备及通信网络系统等。

1. 数字系统的基本模型

数字系统一般包括 3 个功能部件：输入与输出接口、信息处理器和控制器，数字系统的各个功能部件是一个有机的整体，如图 5.1 所示。输入与输出接口是完成物理量与数字量转化的功能部件，而信息处理器是对数字量进行处理和加工的功能部件，控制器是对整个数字系统进行协调和控制的功能部件，又可称为数字逻辑子系统。

任何数字系统都可用最基本的两大部件——信息处理器和控制器来描述其基本结构。信息处理器按控制器提供的不同的操作序列来执行相应的操作并完成最基本的信息传送，对信息进行不同的处理。控制器可接收外部控制信息和信息处理器的状态信息来改变正在执行的或选择下一个需要执行的操作序列。在系统工作的整个过程中，控制器不断地生成和发送控制信号操作序列使信息处理器不断地执行指定的操作，同时也产生影响其他控制器操作的控制信息。以这种方式，通过控制器和信息处理器之间的密切配合、协调工作，构成了一个自动的、有机统一的数字系统。

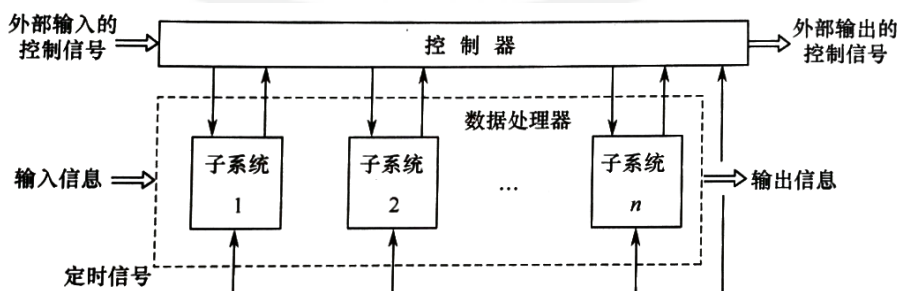


图 5.1 数字系统的一般模型

综上所述，按预定要求能够产生或加工处理数字信息的装置一般都称为一个独立的数字系统。功能比较单一的部件通常称为逻辑功能部件。数字系统通常由几个逻辑功能部件构成，并由一个控制部件统一指挥。一个数字系统通常有几个子系统构成，一个子系统又可以由更小的子系统构成，当子系统比较简单、功能十分明确时，这个子系统一般就是一个逻辑功能部件。当这些子系统合并为大系统时，就必须有一个控制部件来统一协调和管理各子系统，按一定的程序统一指挥整个系统工作。**因此是否有控制器是区别数字系统和逻辑功能部件的重要标志。**

2. 控制器

控制单元的核心是时序电路，本质上是一个状态寄存器。状态寄存器主要有两个功能——寄存控制单元的现态，生成次态。采用触发器作为状态寄存器的元件。

存在着两种不同的控制单元实现方法：

- ✧ 硬件逻辑方法——用逻辑电路生成每一个微操作的控制信号；特点：速度快、牵一发而动全身。
- ✧ 微程序方法——计算机的每一条指令的功能通过执行一个微指令序列(微程序)来实现的。设计好的微程序被固化在只读存储器中，这个存储器称为控制存储器。特点：速度低、但设计、修改及扩充容易。

显然，采用不同的实现方法，将影响控制单元的组成和结构。在现代计算机控制器中，常常两者混用。

3. 信息处理器

典型的信息处理器包括三个部分：组合逻辑网络（根据输入值和存储单元的源数据生成结果单元的结果数据、输出信号和状态信号，由控制信号 C 控制）、存储单元（暂存操作处理中的源数据和中间结果）、专用控制网络（根据来自控制器的控制信号和传送脉冲，生成存储单元传送信号，并由此决定信息处理器执行的传送操作）

4. 数字系统的设计方法

- 第一步：确定系统的逻辑功能。
设计者应对系统仔细分析、消化和理解，逐步明确其逻辑功能，输入、输出信号等内容。
- 第二步：确定系统方案。
这是设计工作中最困难、最有创造性的工作，设计者根据设计要求分析推演出信息处理的基本原理和可供选择的结构形式，因为可以采用不同的原理和方法实现某一逻辑功能。为此设计者要进行认真的比较和权衡，从中选取较为满意的方案。
- 第三步：对系统进行逻辑划分。
将系统按信息处理单元和控制单元划分为两大部分，列出信息处理单元的说明，并用流程图等方法描述信息处理的算法（即控制单元的逻辑要求）。每个部分应具备基本独立的逻辑功能。逻辑划分和确定系统方案的过程要同时进行、相辅相成。
- 第四步：设计信息处理单元和控制单元。
 - ✧ 定义要求信息处理单元必须执行的处理和操作，列出操作表；
 - ✧ 提出实现的算法，确定控制单元必须保存的、或产生的状态及状态之间的转换关系；
 - ✧ 由系统的控制状态、信息处理单元产生的状态信息和要求生成的控制信号建立控制单元的状态转换表；
 - ✧ 根据信息处理单元的操作表和状态信息、控制单元的状态转换表，进行逻辑设计。
 - ✧ 选择逻辑功能部件，如 SSI、MSI、LSI 等，进行合理连接，构成严格能协调工作的系统。

5. 描述工具

- 逻辑流程图（描述对象是控制器）

也称流程图，它用约定的几何图形(矩形、菱形、椭圆形等)、指向线和简练的文字说明，描述系统的基本工作过程。

逻辑流程图的描述对象是控制单元，并且以系统时钟来驱动整个流程。这一点与由事件驱动的软件流程图不同。

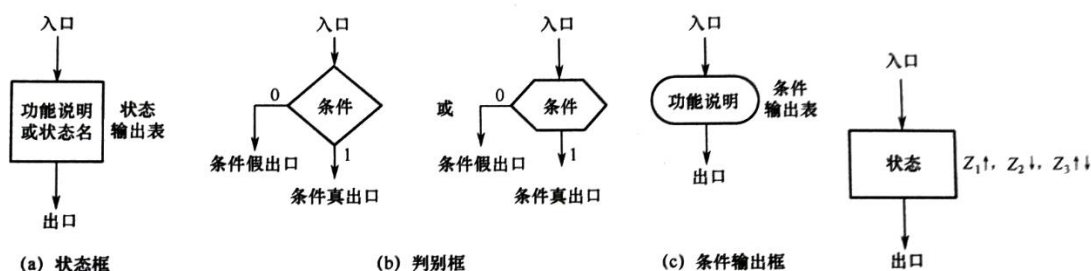


图 5.6 逻辑流程图的基本符号

图 5.7 状态输出表

z1 ↑表示进入状态 state，输出 z1 有效。

z2 ↓表示进入状态 state，输出 z2 无效。

z3 ↑↓表示进入状态 state，输出 z3 有效，并在退出状态 state，输出 z3 无效

● ASM 图

逻辑流程图是数字系统中使用得最广泛的一种非形式化的描述工具，但它的规范性不够。经过不断改进，将流程图改造成描述数字系统硬件的形式化工具

ASM 图是形式化描述数字系统的流程图，是按照一组简单和明确的规则绘制的。在时序电路的 ASM 图中，每一个状态由一个 ASM 块来表示。一个 ASM 块是一种小型的符号化结构图形，由它指出了当前状态、它的输出和次态的条件。一个时序数字系统的所有状态的 ASM 块，按照一定的顺序联结成一个网状结构，由此产生的 ASM 图可用来精确地描述系统所具有的功能。

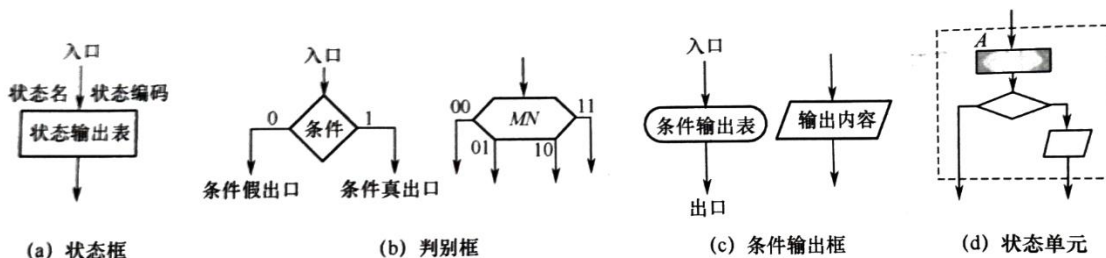


图 5.9 ASM 图的基本符号

由逻辑流程图转换成 ASM 图

ASM 图可以直接送入计算机辅助逻辑设计系统，由系统自动完成控制单元的设计。

逻辑流程图可以很容易地转换成 ASM 图。

在 ASM 图中，所有的当前状态下的 Moore 型有效输出都应列在状态框内；

所有的当前状态下的满足输入条件的 Mealy 型有效输出都应列在条件输出框内。

ASM 图的特点

ASM 图为时序电路系统提供了形式化描述方法。

在 ASM 图中，每个状态框仅有一个出口（分支由判断框提供），保证了无二义性，且自动满足闭合性和完整性。

逻辑流程图于 ASM 图的主要区别？

逻辑流程图的描述对象是控制器，并且以系统时钟来驱动整个流程。逻辑流程图能够描述整个数字系统对信息的处理过程，也能描述控制器所提供的控制步骤，它既便于设计者发现和改良信息处理过程中的错误和缺陷，又可以成为电路后续设计的依据。

ASM 图即算法流程图，是在逻辑流程图的基础上发展起来的，它能和实现它的硬件很好地对应起来。二者的判别框及条件输出框基本相同，能够很好转换。但其状态框的状态输出表填在框内，状态名在左上角，而逻辑流程图的状态输出表标在框外，里面是状态名。在 ASM 图中判别框忽略了在特定条件下的无效输入信号。

【写在最后】

本复习随记适用教材为朱正东、伍卫国等老师编写的《数字逻辑与数字系统》，由于电信学部内各专业的讲授内容和重点可能有所不同，且每年的讲授内容会有所变更和调整，本复习随记仅整理了一些基本概念、基本元件和基本方法，若在复习时梳理使用，请对照任课老师讲授的实际情况进行增删。大家如果发现本资料有任何问题，请以书本和老师讲授为主，也可以联系我进行修改。

本复习能够成稿，离不开田琛老师一学期认真、负责、细致的讲授，也要感谢朱正东老师在数电实验中的耐心指导能让我在实践中加深对理论知识的掌握，同时在复习整理的过程中我的舍友高天宇、张晨、孙鹏程，以及“双一流重点建设编辑部（群号 689822823，若感兴趣，欢迎加入）”的同学也提出了很多宝贵的修改意见。

祝大家学习进步！

西安交通大学

电子与信息工程学部·计算机学院

计算机 74 班任隽阳

2019 年 6 月 26 日

获取其他资料，请扫描右侧小程序码：（以下资料非自编部分若有因版权问题需要删除，请及时联系）

1. 优化方法基础复习随记
2. 《凸优化》英文原版电子书及课后习题答案
3. 生命科学基础复习梳理
4. 模拟电子技术复习随记
5. 模拟电子技术基础各章节学习指导（摘自徐正红老师的 MOOC）
6. 概率统计与随机过程复习随记
7. 《概率统计与随机过程习题解集》机械工业出版社
8. 学术英语听说课程考察词汇及表达
9. (即将发行)欧洲文化渊源复习随记
10. 《军事理论教程（第 4 版）》书本知识要点及笔记整理（附 MOOC 测试题参考答案及军理复习小贴士等）
11. 《思想道德修养与法律基础（2015 年修订版）》考点整理（附 PPT 法律部分文字及十九大报告）
12.



双一流重点建设编辑部（群号：689822823） 出品

整理不易 打赏鼓励

(打印时可在打印选项中选择不打印此页)



支付就用支付宝



打开支付宝【扫一扫】

免费寄送收钱码：拨打95188-6

扫码领红包

天天可领 想花就花



打开支付宝【扫一扫】

活动规则：
活动期间每人每天限领1次红包，在门店付款时自动立减红包金额（活动非营业时间无法领取）

更多活动详情上支付宝

搜索“领钱红包”



推荐使用微信支付



服务通知(**阳)



微信支付