

第四章 组合逻辑电路

梁华国

电子科学与技术系

<http://dwxy.hfut.edu.cn/>

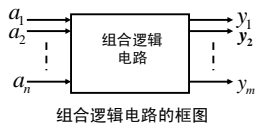
组合逻辑电路

- 概述
- 组合逻辑电路分析
- 组合逻辑电路设计
- 考虑特殊问题的逻辑设计
- 若干常用的组合逻辑电路
- 组合逻辑电路中的竞争-冒险

概述

组合逻辑电路的定义:

- 是指电路在任何时刻产生的稳定输出信号，仅取决于该时刻电路的输入信号。



$$y_1 = f_1(a_1, a_2, \dots, a_n)$$

$$y_2 = f_2(a_1, a_2, \dots, a_n)$$

$$\vdots$$

$$y_m = f_m(a_1, a_2, \dots, a_n)$$

组合逻辑电路

- 概述
- 组合逻辑电路分析
- 组合逻辑电路设计
- 若干常用的组合逻辑电路
- 考虑特殊问题的逻辑设计
- 组合逻辑中竞争冒险
- 常用的中规模组合逻辑标准构件

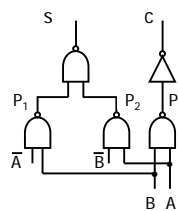
组合逻辑电路分析

一般分析步骤:

- 阅读组合逻辑电路图
- 列写逻辑表达式
- 列出真值表
- 指出电路的逻辑功能
- 做出对逻辑电路图的评价和改进

组合逻辑电路分析

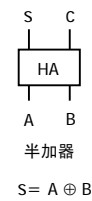
分析下图逻辑功能



$$S = P_1 \cdot P_2 = \overline{P_1} + \overline{P_2} = \overline{AB} + A\overline{B}$$

$$C = \overline{P_3} = AB$$

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路分析

$$S_i = A_i \oplus B_i \oplus C_{i-1} = \bar{S}_i C_{i-1} + S_i \bar{C}_{i-1}$$

$$= (\bar{A}_i \bar{B}_i + A_i B_i) C_{i-1} + (\bar{A}_i B_i + A_i \bar{B}_i) \bar{C}_{i-1}$$

$$= \bar{A}_i \bar{B}_i C_{i-1} + A_i B_i C_{i-1} + \bar{A}_i B_i \bar{C}_{i-1} + A_i \bar{B}_i \bar{C}_{i-1}$$

$$C_i = S_i C_{i-1} + C_i = (\bar{A}_i B_i + A_i \bar{B}_i) C_{i-1} + A_i B_i$$

$$= \bar{A}_i B_i C_{i-1} + A_i \bar{B}_i C_{i-1} + A_i B_i \bar{C}_{i-1} + A_i B_i C_{i-1}$$

A_i	B_i	C_{i-1}	S_i	C_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

全加器

电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路分析

逐级电平推导法

$$F = 1$$

全加器

电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路分析

列写逻辑表达式分析下图逻辑功能

$$P_1 = \bar{a}b, P_2 = \bar{a} + c$$

$$P_3 = b \oplus \bar{c}, P_4 = P_1 \cdot P_2$$

$$f(a,b,c) = \overline{P_3 + P_4} = \overline{(b \oplus \bar{c}) + \bar{a}b(\bar{a} + c)}$$

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

$$= bc + \bar{b}\bar{c} = \bar{b} \odot c = b \oplus c$$

电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路分析

译码器的分析

译码器

译码为“1”输出

译码为“0”输出

n-to-2ⁿ的译码器，对于每一种输入可能，只有一个输出信号被译中
n个输入2ⁿ个输出，功能相当于最小项产生器

电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路分析

数据选择器的分析

A	B	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

$$Y = \bar{A}\bar{B} \cdot D_0 + \bar{A}B \cdot D_1 + A\bar{B} \cdot D_2 + AB \cdot D_3$$

电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路分析

多路分配器的分析

多路分配器

P209-210: 题4.1、4.2、4.3

第四章 组合逻辑电路

梁华国

电子科学与技术系

<http://dwxy.hfut.edu.cn/>



组合逻辑电路

- 概述
- 组合逻辑电路分析
- 组合逻辑电路设计
- 考虑特殊问题的逻辑设计
- 若干常用的组合逻辑电路
- 组合逻辑电路中的竞争-冒险



组合逻辑电路设计

一般设计步骤:

- 根据功能描述列出真值表
- 根据真值表化简逻辑函数为最简的“与-或”表达式
- 根据选用的门电路的类型及其实际问题的要求, 将函数转化成所需要的表达式
- 画出逻辑图



组合逻辑电路设计

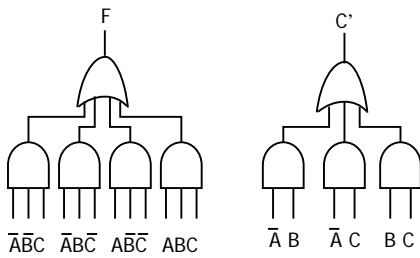
例: 设计一位全减器, 它有三个输入端: 被减数A, 减数B, 低位借位C; 输出: 差F, 向高位的借位C'.

A	B	C	F	C'
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	0	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$
 $C' = \bar{A}B + \bar{A}C + BC$



组合逻辑电路设计



组合逻辑电路设计

例: 某学期考试四门课程:

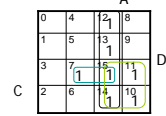
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

数学: 7 学分;
 英语: 5 学分;
 政治: 4 学分;
 体育: 2 学分

每个学生总计要获得10个以上学分才能通过本学期的考试。要求写出反映学生是否通过本学期的逻辑函数。

设 A、B、C、D 分别为四门课, “1”表示通过此门课程通过, “0”表示不通过; F 为“1”时表示本学期的考试通过, “0”为没通过。

$$F = AB + AC + BCD$$



组合逻辑电路设计

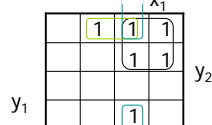
例：已知 $X=x_1x_2$ 和 $Y=y_1y_2$ 是两个正整数，写出判断 $X>Y$ 的逻辑表达式。

$X>Y$ 的简化真值表

x_1	x_2	y_1	y_2	F
1	x	0	x	1
0	1	0	0	1
1	1	1	0	1

$$x_1\bar{y}_1 \quad \bar{x}_1x_2\bar{y}_1\bar{y}_2 \quad x_1x_2y_1\bar{y}_2$$

$$F = x_1y_1 + \bar{x}_1x_2y_1y_2 + x_1x_2y_1\bar{y}_2$$



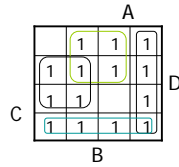
$$F = x_1\bar{y}_1 + x_2\bar{y}_1\bar{y}_2 + x_1x_2\bar{y}_2$$

组合逻辑设计

逻辑函数的“与非”门实现

1. 原函数二次反演，一次展开：

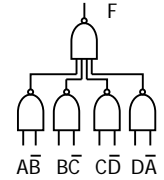
例如： $F = A\bar{B} + B\bar{C} + C\bar{D} + D\bar{A}$



$$F = A\bar{B} + B\bar{C} + C\bar{D} + D\bar{A}$$

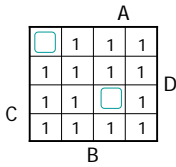
$$= \overline{\overline{A\bar{B} + B\bar{C} + C\bar{D} + D\bar{A}}}$$

$$= \overline{\overline{A\bar{B}} \cdot \overline{B\bar{C}} \cdot \overline{C\bar{D}} \cdot \overline{D\bar{A}}}$$



组合逻辑电路设计

2. 反函数三次反演，一次展开：

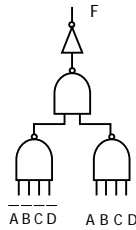


$$\bar{F} = \overline{\overline{A\bar{B}\bar{C}\bar{D} + ABCD}}$$

$$= \overline{\overline{A\bar{B}\bar{C}\bar{D}} \cdot \overline{ABCD}}$$

$$= \overline{\overline{A\bar{B}\bar{C}\bar{D}} \cdot \overline{ABCD}}$$

$$= \overline{\overline{A\bar{B}\bar{C}\bar{D}} \cdot \overline{ABCD}}$$

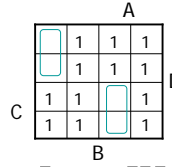


组合逻辑电路设计

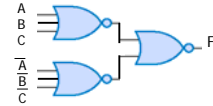
逻辑函数的“或非”门实现

▪采用“与非”门实现的对偶方法 ▪直接采用反演定理

例如：用或非门实现函数 $F = A\bar{B} + B\bar{C} + C\bar{A}$



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



$$F = \overline{\overline{A\bar{B} + B\bar{C} + C\bar{A}}}$$

组合逻辑电路设计

逻辑函数的“与或非”门实现

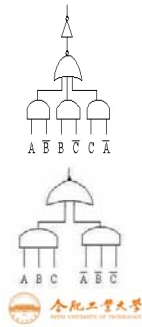
例如：用与或非门实现函数 $F = A\bar{B} + B\bar{C} + C\bar{A}$

$$\text{对 } F \text{ 两次求反 } \bar{\bar{F}} = \overline{\overline{A\bar{B} + B\bar{C} + C\bar{A}}}$$

$$\text{对 } \bar{F} \text{ 一次求反 } \bar{F} = \overline{A\bar{B} + B\bar{C} + C\bar{A}}$$

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

$$F = \bar{\bar{F}} = \overline{\overline{A\bar{B}} \cdot \overline{B\bar{C}} \cdot \overline{C\bar{A}}}$$



习题

P61 题2.12、2.13

P210-211题4.5、4.6、4.9

组合逻辑电路

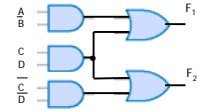
- 概述
- 组合逻辑电路分析
- 组合逻辑电路设计
- 考虑特殊问题的逻辑设计
- 若干常用的组合逻辑电路
- 组合逻辑电路中的竞争-冒险

考虑特殊问题的逻辑设计

多输出函数的逻辑设计

例如: $F_1 = A\bar{B} + CD$

$F_2 = CD + \bar{C}\bar{D}$

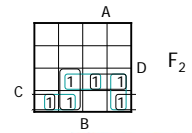


公共的项并不总是很明显:

$F_1 = \bar{A}BD + \bar{A}\bar{C}\bar{D}$

$F_2 = \bar{A}BC + ACD + \bar{B}\bar{C}\bar{D}$

等效的 $F_2 = \bar{A}\bar{C}\bar{D} + BCD + \bar{A}\bar{B}C$



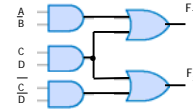
组合逻辑电路

- 概述
- 组合逻辑电路分析
- 组合逻辑电路设计
- 考虑特殊问题的逻辑设计
- 若干常用的组合逻辑电路
- 组合逻辑电路中的竞争-冒险

考虑特殊问题的逻辑设计

多输出函数的逻辑设计

例如： $F_1 = A\bar{B} + CD$
 $F_2 = CD + \bar{C}\bar{D}$

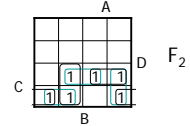


公共的项并不总是很明显：

$$F_1 = \bar{A}BD + \bar{A}\bar{C}\bar{D}$$

$$F_2 = \bar{A}BC + ACD + \bar{B}\bar{C}\bar{D}$$

等效的 $F_2 = \bar{A}\bar{C}\bar{D} + BCD + \bar{A}BC$



考虑特殊问题的逻辑设计

多输出的逻辑函数的化简步骤：

1. 构成各自函数以及各个函数之间所有组合相交的卡诺图。
2. 找出每个卡诺图的所有极大块，如果在高一级以上相交卡诺图已圈过的极大块，在后继卡诺图中，不再圈出。
3. 构成函数的最小覆盖。对函数有关的所有卡诺图，从高到低优先选择极大块，如果发现某级的极大块被其它级极大块包含，此块应删去。

特点：点面相结合

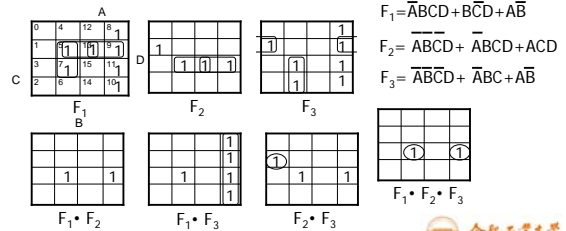
考虑特殊问题的逻辑设计

例如：已知有四个输入变量的三个逻辑函数：

$$F_1 = \sum m(5, 7, 8, 9, 10, 11, 13) \quad F_2 = \sum m(1, 7, 11, 15)$$

$$F_3 = \sum m(1, 6, 7, 8, 9, 10, 11)$$

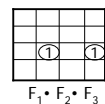
求其一组最简的 F_1, F_2, F_3 与-或-表达式。



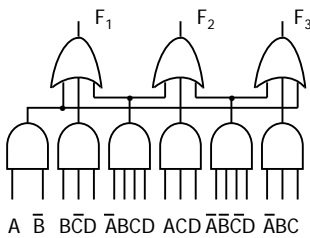
$$F_1 = \bar{A}BCD + B\bar{C}D + A\bar{B}$$

$$F_2 = \bar{A}BCD + \bar{A}BCD + ACD$$

$$F_3 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}BC + A\bar{B}$$



考虑特殊问题的逻辑设计



考虑特殊问题的逻辑设计

包含无关最小项的逻辑设计

在 2^n 个最小项中，一部分最小项并不能决定函数的值，我们把这些最小项称为无关最小项

无关最小项发生在两种情况：

- 输入某些组合不可能出现
- 所有输入都可能出现，但其中部分输入对其输出并不关心

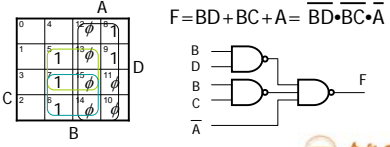
化简依据：逻辑函数加上或者去掉无关最小项，对原函数逻辑功能无影响

考虑特殊问题的逻辑设计

例如：用与非门设计一个判别电路，以判别8421码所表示的十进制数之值是否大于等于5

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	φ
1	1	0	1	φ
1	1	1	0	φ
1	1	1	1	φ

设：8421码对应输入变量：A,B,C,D, 输出函数为F,
 $ABCD \geq 0101$ 时, $F=1$; 当 $ABCD < 0101$ 时, $F=0$
 $F = \sum m(5,6,7,8,9) + \sum \phi(10,11,12,13,14,15)$
 $\sum \phi(10,11,12,13,14,15) = 0$

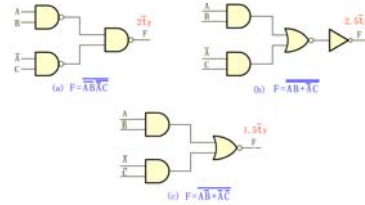


考虑特殊问题的逻辑设计

考虑级数的逻辑设计

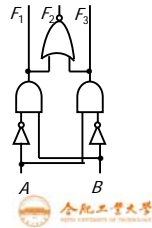
例：用与非门、与或非门分别实现以下函数：

$$F = AB + \overline{AC}$$



课堂练习

- 用代数法化简函数
 $F = \overline{A}B + AC + \overline{C}D + \overline{B}C\overline{D} + B\overline{C}E + \overline{B}C\overline{G} + \overline{B}CF$
- 已知逻辑函数： $F = \overline{A}BE + BCDE + \overline{B}C\overline{D}E + \overline{B}CDE$
 的最简表达式为： $F = BE + BDC$
 求其可确认的无关最小项。
- 分析右图，并说明其逻辑功能
- 设计一个二位二进制乘法器的逻辑电路





电子科学与应用物理学院
School of Electronic Science & Applied Physics

第四章 组合逻辑电路

梁华国 黄正峰
电子科学与技术系
<http://dwxy.hfut.edu.cn/>



1



电子科学与应用物理学院
School of Electronic Science & Applied Physics

组合逻辑电路

- 概述
- 组合逻辑电路分析
- 组合逻辑电路设计
- 考虑特殊问题的逻辑设计
- 若干常用的组合逻辑电路
- 组合逻辑电路中的竞争-冒险



2



4.3 若干常用的组合逻辑电路

4.3.1 编码器(Encoder)

编码：为了区分一系列不同的事物，将其中的每个事物用二值代码表示。

编码器：由于在二值逻辑电路中，信号是以高低电平给出的，故编码器就是把输入的每一个高低电平信号变成一个对应的二进制代码。

编码器分为**普通编码器**和**优先权编码器**。根据进制可分为**二进制编码器**和**二十进制编码器**。

3



4.3.1 编码器

一、普通编码器

3位二进制普通编码器，也称为8线-3线编码器，其框图如图4.3.1所示

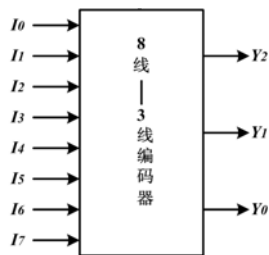


图4.3.1 位二进制编码器的框图

$I_0 \sim I_7$ 为信号输入端，高电平有效； $Y_2 Y_1 Y_0$ 为三位二进制代码输出端，由于输入端为8个，输出端为3个，故也叫做8线-3线编码器

4





4.3.1 编码器

其输入、输出的真值表为

输入								输出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
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

利用无关项化简得到
其输出端逻辑式为

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

特点：任何时刻只允许输入一个编码信号

5



4.3.1 编码器

其逻辑电路如图4.3.2所示

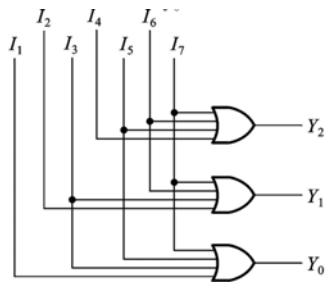


图4.3.2 3位二进制编码器（8线-3线编码器）

6





二、优先编码器

普通编码器每次只能输入一个信号。而优先编码器可以同时输入几个信号，但在设计时已经将各输入信号的优先顺序排好。当几个信号同时输入时，优先权最高的信号优先编码。

下面以8线-3线优先编码器 **74HC148** 为例，其逻辑符号如图4.3.3所示，内部电路如书P₁₇₀图4.3.3所示。

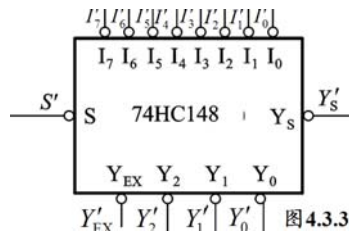


图4.3.3



二、优先编码器

由P₁₇₀图4.3.3可知，如果不考虑输出扩展端，8线-3线优先编码器（设 I_7 优先权最高，...， I_0 优先权最低）其输出端的逻辑式为

$$Y_2' = [(I_7 + I_6 + I_5 + I_4)S']$$

$$Y_1' = [(I_7 + I_6 + I_5 I_4' I_3' + I_2 I_4' I_5')S']$$

$$Y_0' = [(I_7 + I_6' I_5 + I_3 I_4' I_6' + I_1 I_2 I_4' I_6')S']$$

其中 S 为选通输入端，当 $S=0$ ($S'=1$) 时所有输出端均被锁定在高电平，即 $I_7' \sim I_0' = 1 \dots 1$ 。当 $S=1$ ($S'=0$) 时，编码器正常工作。





二、优先编码器

由P170图4.3.3可知，不考虑扩展端，8线-3线优先编码器（设 I_7 优先权最高，...， I_0 优先权最低）其真值表如表所示

输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
X	X	X	X	X	X	X	1	1	1	1
X	X	X	X	X	X	1	0	1	1	0
X	X	X	X	X	1	0	0	1	0	1
X	X	X	X	1	0	0	0	1	0	0
X	X	X	1	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0
X	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0

9



二、优先编码器

为了扩展电路的功能和使用的灵活性，在8线-3线优先编码器74HC148中附加了选通输出端 Y_S 和扩展端 Y_{EX} ，且由P170图4.3.3可知

为0时，电路工作无编码输入

$$Y'_S = (I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0 S)'$$

$$Y'_{EX} = [(I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0 S) S]'$$

$$[(I_7 + I_6 + I_5 + I_4 + I_3 + I_2 + I_1 + I_0) S]'$$

为0时，电路工作有编码输入

10





二、优先编码器

74HC148的真值表如下表

输入								输出					
S'	I'_0	I'_1	I'_2	I'_3	I'_4	I'_5	I'_6	I'_7	Y'_2	Y'_1	Y'_0	Y'_S	Y'_{EX}
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	x	x	x	x	x	x	x	0	0	0	0	1	0
0	x	x	x	x	x	x	0	1	0	0	1	1	0
0	x	x	x	x	x	0	1	1	0	1	0	1	0
0	x	x	x	x	0	1	1	1	0	1	1	1	0
0	x	x	x	0	1	1	1	1	1	0	0	1	0
0	x	x	0	1	1	1	1	1	1	0	1	1	0
0	x	0	1	1	1	1	1	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Y'_S	Y'_{EX}	状态
1	1	不工作
0	1	工作, 但无输入
1	0	工作, 且有输入
0	0	不可能出现



优先编码器的特点 (1)

输入								输出					
S'	I'_0	I'_1	I'_2	I'_3	I'_4	I'_5	I'_6	I'_7	Y'_2	Y'_1	Y'_0	Y'_S	Y'_{EX}
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	x	x	x	x	x	x	x	0	0	0	0	1	0
0	x	x	x	x	x	x	0	1	0	0	1	1	0
0	x	x	x	x	x	0	1	1	0	1	0	1	0
0	x	x	x	x	0	1	1	1	0	1	1	1	0
0	x	x	x	0	1	1	1	1	1	0	0	1	0
0	x	x	0	1	1	1	1	1	1	1	0	1	0
0	x	0	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

1.74 LS148 输入端的优先级为 I'_7 最高, 其次为 I'_6 , 依次下去, I'_0 优先级最低;



优先编码器的特点 (2)

输入		输出											
S'	I'_0	I'_1	I'_2	I'_3	I'_4	I'_5	I'_6	I'_7	Y'_2	Y'_1	Y'_0	Y'_S	Y'_{EX}
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	x	x	x	x	x	x	x	0	0	0	0	1	0
0	x	x	x	x	x	x	0	1	0	0	1	1	0
0	x	x	x	x	x	0	1	1	0	1	0	1	0
0	x	x	x	x	0	1	1	1	0	1	1	1	0
0	x	x	x	0	1	1	1	1	1	0	0	1	0
0	x	x	0	1	1	1	1	1	1	0	1	1	0
0	x	0	1	1	1	1	1	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0

2.当 $S' = 1$ 时, 编码器不能工作, 此时输出端

$$Y'_2 Y'_1 Y'_0 = 111, Y'_S = 1, Y'_{EX} = 1;$$



优先编码器的特点 (3)

输入		输出											
S'	I'_0	I'_1	I'_2	I'_3	I'_4	I'_5	I'_6	I'_7	Y'_2	Y'_1	Y'_0	Y'_S	Y'_{EX}
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	x	x	x	x	x	x	x	0	0	0	0	1	0
0	x	x	x	x	x	x	0	1	0	0	1	1	0
0	x	x	x	x	x	0	1	1	0	1	0	1	0
0	x	x	x	x	0	1	1	1	0	1	1	1	0
0	x	x	x	0	1	1	1	1	1	0	0	1	0
0	x	x	0	1	1	1	1	1	1	0	1	1	0
0	x	0	1	1	1	1	1	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0

3.当 $S' = 0$ 时, 编码器可以工作。此时如果没有输入信号输入, 即 $I'_7 \sim I'_0 = 1 \cdots 1$ 时, 则输出端为 $Y'_2 Y'_1 Y'_0 = 111$, 但 $Y'_S = 0, Y'_{EX} = 1$;



优先编码器的特点 (4)

	输入								输出				
	S'	I'_6	I'_5	I'_4	I'_3	I'_2	I'_1	I'_0	Y'_2	Y'_1	Y'_0	Y'_S	Y'_{EX}
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	x	x	x	x	x	x	x	0	0	0	0	1	0
0	x	x	x	x	x	x	0	1	0	0	1	1	0
0	x	x	x	x	x	0	1	1	0	1	0	1	0
0	x	x	x	x	0	1	1	1	1	0	0	1	0
0	x	x	0	1	1	1	1	1	1	0	1	1	0
0	x	0	1	1	1	1	1	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0

4当 $\bar{S} = 0$,编码器有输入时,若几个信号同时输入,则优先级高得信号优先编码,但在比这个信号优先权高的输入信号必须为高电平,比这个信号优先权低的信号,可以是高电平也可以是低电平.

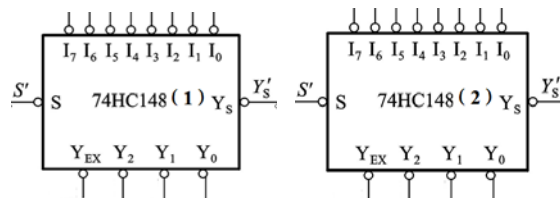
如 $\bar{I}_7\bar{I}_6\bar{I}_5 = 111, \bar{I}_4 = 0, \bar{I}_3 \sim \bar{I}_0$ 任意时,则输出为 $\bar{Y}_2\bar{Y}_1\bar{Y}_0 = 011$.输出是以反码的形式输出。



例4.3.1 (1)

例4.3.1 试用两片74HC148接成16线-4线优先编码器,将 $A'_0 \sim A'_{15}$ 16个低电平输入信号编为0000~1111 16个4位二进制代码,其中 A'_{15} 的优先权最高, A'_0 的优先权最低

解: a.要求16个输入端,正好每个74LS148有8个输入端,两片正好16个输入端,满足输入端的要求;

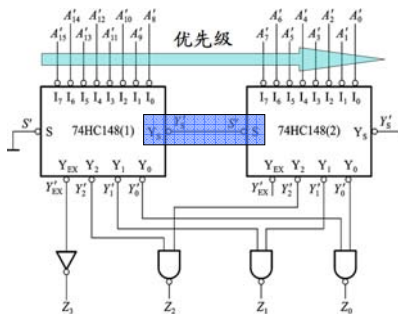




例4.3.1 (2)

b. 根据优先权的要求，若第一片的优先级比第二片高，则第一片的输入为 $A'_{15} \sim A'_8$ ，第二片的输入为 $A'_7 \sim A'_0$ 。当第一片工作，即有输入信号时，第二片禁止工作，也就是使得第二片的 $S'=1$ 。

由表中可知可将第一片的 Y'_S 接到第二片的 S' 上



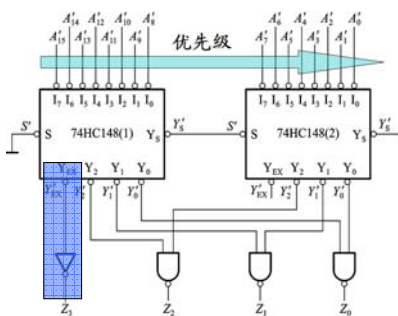
Y'_S	Y'_{EX}	状态
1	1	不工作
0	1	工作，但无输入
1	0	工作，且有输入
0	0	不可能出现

17



例4.3.1 (3)

c. 由于74HC148输出端只有3个，要想根据要求输出为4线，必须借用第一片的扩展端 Y'_{EX} 。由于有输入时， $Y'_{EX}=0$ ，无输入时 $Y'_{EX}=1$ ，故加反相器可作输出四位二进制数码的最高位。



Y'_S	Y'_{EX}	状态
1	1	不工作
0	1	工作，但无输入
1	0	工作，且有输入
0	0	不可能出现

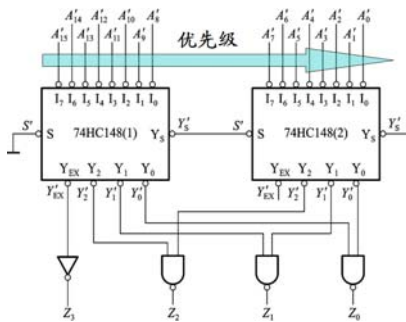
18





例4.3.1 (4)

其逻辑接线图如图4.3.4所示。



- 第一片为高优先级
- 只有(1)无编码输入时，(2)才允许工作
- 第(1)片 $Y'_{EX} = 0$ 时表示对 $A'_{15} \sim A'_8$ 的编码
- 低3位输出应是两片的输出的“与非”

Y'_S Y'_{EX}	状态
1 1	不工作
0 1	工作，但无输入
1 0	工作，且有输入
0 0	不可能出现



二-十进制优先编码器74LS147

将十个信号编成10个BCD代码。其内部逻辑图见书P₁₇₃图4.3.5所示。其逻辑符号如图4.3.5所示

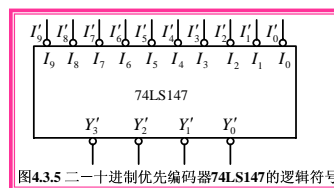


图4.3.5 二-十进制优先编码器74LS147的逻辑符号

$I'_9 \sim I'_0$ 为10个输入信号， I'_9 的优先权最高， I'_0 的优先权最低； $Y'_3 \sim Y'_0$ 为四位二进制BCD码的输出端



二-十进制优先编码器

其功能表为：



输入									输出			
I'_1	I'_2	I'_3	I'_4	I'_5	I'_6	I'_7	I'_8	I'_9	Y'_3	Y'_2	Y'_1	Y'_0
1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	0	0	1	1	0
x	x	x	x	x	x	x	0	1	0	1	1	1
x	x	x	x	x	0	1	1	1	1	0	0	0
x	x	x	x	0	1	1	1	1	1	0	0	1
x	x	x	0	1	1	1	1	1	1	0	1	0
x	x	0	1	1	1	1	1	1	1	0	1	1
x	0	1	1	1	1	1	1	1	1	1	0	0
x	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

1. 当 I'_0 有输入信号，其他输入为高电平，输出 $Y'_3 Y'_2 Y'_1 Y'_0 = 1111$ ；

2. 输出代码为对应二进制BCD码的**反码**，如 $I'_6 = 0$ 时，输出为 $Y'_3 Y'_2 Y'_1 Y'_0 = 1001$ ，为0110的反码



4.3.2 译码器

译码器就是将每个输入的二进制代码译成对应的输出高、低电平信号，和编码器逆过程。常用的译码器分为二进制译码器、二—十进制译码器和显示译码器。

一、二进制译码器

即将N位二进制代码译成 2^N 个高低电平信号，称为N线- 2^N 线译码器。如N=3,则可译 $2^3=8$ 个高低电平信号，称为3线-8线译码器。

图4.3.6为3线-8线译码器的框图。其中： $A_2 \sim A_0$ —二进制代码输入端； $Y_7 \sim Y_0$ —信号输出端

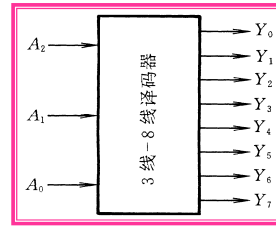


图4.3.6 3线-8线译码器的框图



3线-8线译码器

其真值表如下表

输入			输出							
A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

各输出端逻辑式为

$$Y_0 = m_0 = A_2' A_1' A_0'$$

$$Y_1 = m_1 = A_2' A_1' A_0$$

$$Y_2 = m_2 = A_2' A_1 A_0'$$

$$Y_3 = m_3 = A_2' A_1 A_0$$

$$Y_4 = m_4 = A_2 A_1' A_0'$$

$$Y_5 = m_5 = A_2 A_1' A_0$$

$$Y_6 = m_6 = A_2 A_1 A_0'$$

$$Y_7 = m_7 = A_2 A_1 A_0$$

称为最小项译码器





二极管与门阵列构成3位二进制译码器（1）

上述最小项3线-8线译码器由二极管与门阵列构成的电路如图4.3.7所示：

设 $V_{CC}=5V$ ，输入信号的高低电平为3V和0V，二极管导通压降为0.7V。

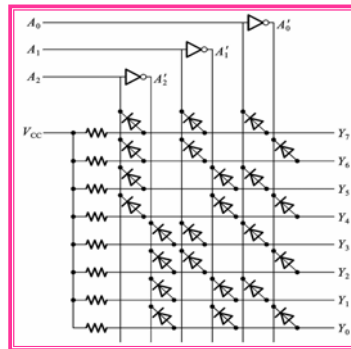
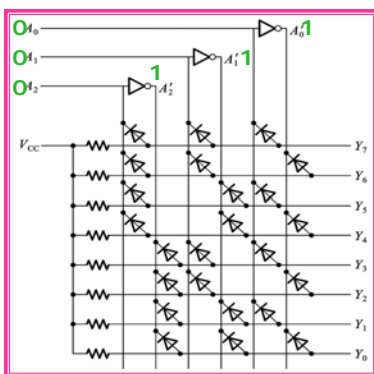


图4.3.7 二极管与门阵列构成的3线-8线译码器

3



二极管与门阵列构成3位二进制译码器（2）



则当 $A_2A_1A_0=000$ 时，则只有 $Y_0=1$

注：二极管构成的译码器优点是电路比较简单。缺点是电路的输入电阻低、输出电阻高。另外存在输出电平移动问题。通常用在中大规模的集成电路中。

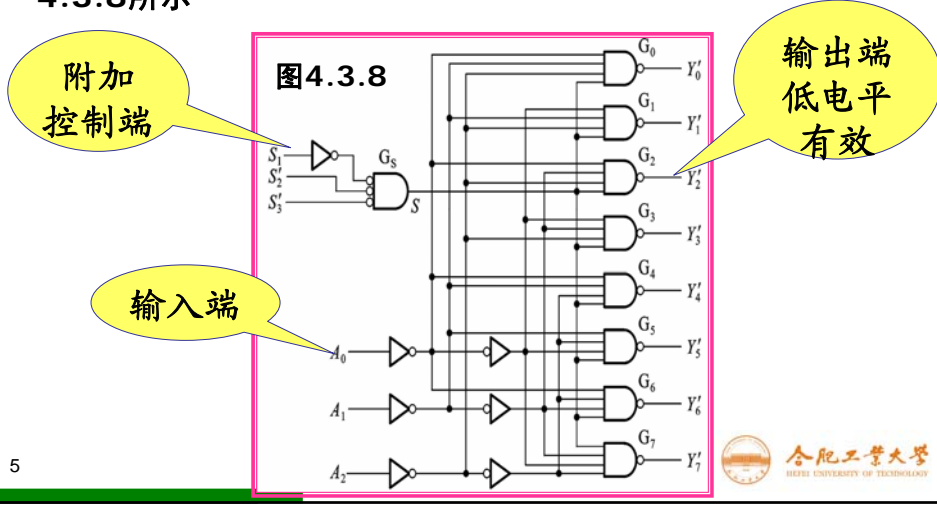
4





中规模集成译码器74HC138 (1)

74HC138是由CMOS门构成的3线-8线译码器，其逻辑图如图4.3.8所示



5



译码器74HC138 (2)

其逻辑功能表为:

输入			输出									
S_1	$S_2' + S_3'$	A_2	A_1	A_0	Y_7'	Y_6'	Y_5'	Y_4'	Y_3'	Y_2'	Y_1'	Y_0'
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

图4.3.9为74HC138的逻辑符号:

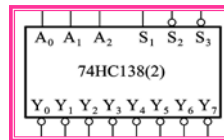


图4.3.9 74HC138的逻辑符号

6





译码器74HC138 (3)

其逻辑功能表为:

a. 当附加控制端 $S_1=0$ 或 $S_2+S_3=1$ 时, 译码器被禁止工作, 输出端状态全部为高电平。

b. 当 $S_1=1, S_2+S_3=0$ 时, 译码器处于工作状态

c. 当译码器工作时, 输出端的逻辑式为

输 入				输 出								
S_1	$S_2 + S_3$	A_2	A_1	A_0	Y_7'	Y_6'	Y_5'	Y_4'	Y_3'	Y_2'	Y_1'	Y_0'
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

$$Y_0' = (A_2 A_1 A_0)' = m_0'; Y_1' = (A_2 A_1 A_0)' = m_1'; Y_2' = (A_2 A_1 A_0)' = m_2'; Y_3' = (A_2 A_1 A_0)' = m_3';$$

$$Y_4' = (A_2 A_1 A_0)' = m_4'; Y_5' = (A_2 A_1 A_0)' = m_5'; Y_6' = (A_2 A_1 A_0)' = m_6'; Y_7' = (A_2 A_1 A_0)' = m_7'.$$



译码器74HC138 (4)

由上面分析可知, 输出端的逻辑式是以输入的三个变量最小项取反的形式, 故这种译码器也叫**最小项译码器**。

输 入				输 出								
S_1	$S_2 + S_3$	A_2	A_1	A_0	Y_7'	Y_6'	Y_5'	Y_4'	Y_3'	Y_2'	Y_1'	Y_0'
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	0	1
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

d. 此译码器也是一个数据分配器(多路输出器), 当 $S_2=S_3=0$ 时, 数据由 S_1 端输入, 具体从哪端输出要由地址 $A_2A_1A_0$ 来定。故 S_1 称为数据输入端, $A_2A_1A_0$ 称为地址输入端。如当 $A_2A_1A_0=101$ 时, 其它门输出端全是高电平, 只有 $Y_5'=(Sm_5)'=S'$, 以反码的形式输出。



例3.3.2 (1)

例3.3.2 试用两片3线-8线译码器74HC138组成4线-16线译码器，将输出的4位二进制代码 $D_3 D_2 D_1 D_0$ 译成16个独立的低电平信号 $Z'_0 \sim Z'_{15}$

解：由于74HC138为3线-8线译码器，要构成4线-16线译码器，需要4个输入地址线，故要除了74HC138的3个输入端外，还要利用附加控制端，根据74HC138功能表，利用 S_1 和 S'_2 及 S'_3

S_1	$S'_2 + S'_3$	Y'_i
0	X	1
X	1	1
1	0	m'_i

(注: m'_i 为输入端组合的最小项)

74HC138的功能表

S_1	$S'_2 + S'_3$	Y'_i
0	X	1
X	1	1
1	0	m'_i

(注: m'_i 为输入端组合的最小项)

实现的电路如图4.3.10所示

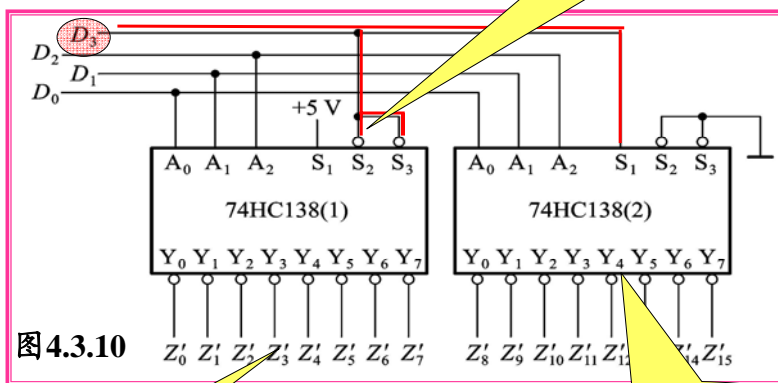


图4.3.10

$D_3=0$ (1) 片工作,
(2) 片不工作

$Z'_i = m'_i$

$D_3=1$ (1) 片不工
作, (2) 片工作



二-十进制译码器(1)

二-十进制译码器就是将10个BCD代码译成10个高低电平的输出信号，BCD码以外的伪码(1010~1111)，输出均无低电平信号产生。

74HC42即为二-十进制的译码器，其内部逻辑图如图4.3.11所示，

$$Y_i' = m_i' \quad (i = 0 \sim 9)$$

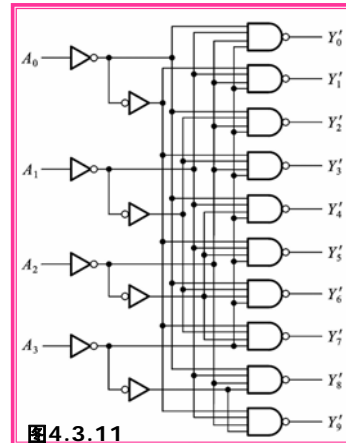


图4.3.11



用译码器设计组合逻辑电路

1. 基本原理

由于译码器的输出为最小项取反，而逻辑函数可以写成最小项之和的形式，故可以利用附加的门电路和译码器实现逻辑函数。

2. 举例

例4.3.1 利用74HC138设计一个多输出的组合逻辑电路，输出逻辑函数式为：

$$\begin{aligned} Z_1 &= AC' + A'BC + AB'C \\ Z_2 &= BC + A'B'C \\ Z_3 &= A'B + AB'C \\ Z_4 &= A'BC' + B'C' + ABC \end{aligned}$$





例4.3.1 (1)

解：先将要输出的逻辑函数化成最小项之和的形式，即

$$\begin{aligned} Z_1 &= AC' + A'BC + AB'C = m_3 + m_4 + m_5 + m_6 \\ Z_2 &= BC + A'B'C = m_1 + m_3 + m_7 \\ Z_3 &= A'B + AB'C = m_2 + m_3 + m_5 \\ Z_4 &= A'BC' + B'C' + ABC = m_0 + m_2 + m_4 + m_7 \end{aligned}$$

将要实现的输出逻辑函数的最小项之和的形式两次取反，即

$$\begin{aligned} Z_1 &= ((m_3 + m_4 + m_5 + m_6)')' = (m_3' \cdot m_4' \cdot m_5' \cdot m_6')' \\ Z_2 &= ((m_1 + m_3 + m_7)')' = (m_1' \cdot m_3' \cdot m_7')' \\ Z_3 &= ((m_2 + m_3 + m_5)')' = (m_2' \cdot m_3' \cdot m_5')' \\ Z_4 &= ((m_0 + m_2 + m_4 + m_7)')' = (m_0' \cdot m_2' \cdot m_4' \cdot m_7')' \end{aligned}$$

13



例4.3.1 (2)

由于74HC138的输出为 $Y_i'(A_2, A_1, A_0) = m_i'(i = 0 \sim 7)$

则用74HC138实现的电路如图4.3.12所示

$$\begin{aligned} Z_1 &= ((m_3 + m_4 + m_5 + m_6)')' = (m_3' \cdot m_4' \cdot m_5' \cdot m_6')' \\ Z_2 &= ((m_1 + m_3 + m_7)')' = (m_1' \cdot m_3' \cdot m_7')' \\ Z_3 &= ((m_2 + m_3 + m_5)')' = (m_2' \cdot m_3' \cdot m_5')' \\ Z_4 &= ((m_0 + m_2 + m_4 + m_7)')' = (m_0' \cdot m_2' \cdot m_4' \cdot m_7')' \end{aligned}$$

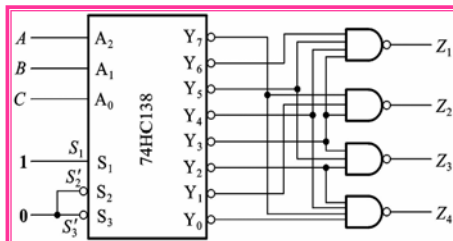


图4.3.12

14





显示译码器

1. 七段字符显示器

即用七段字符显示0~9个十进制数码，常用的七段字符显示器有半导体数码管和液晶显示器两种。

图4.3.12为半导体数码管BS201A（共阴极）的外形示意图及内部等效电路

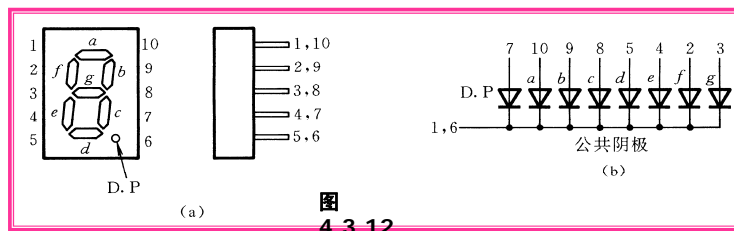


图 4.3.12



显示译码器

注：(1) 半导体数码管每段都是一个发光二极管（LED），材料不同，LED发出光线的波长不同，其发光的颜色也不一样。

(2) 半导体数码管分共阴极和共阳极两类，BS201A属于共阴极类型，因为从内部电路上看，其各发光二极管的阴极是接在一起的。当外加高电平时，发光二极管亮，故高电平有效。而共阳极内部电路如图4.3.15所示，故低电平有效。

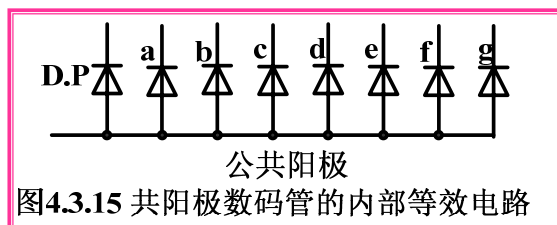


图4.3.15 共阳极数码管的内部等效电路



显示译码器

(3) 半导体数码管的优点是工作电压低，体积小、寿命长、可靠性高、响应时间短、亮度高等。缺点为工作电流大（10mA）。



显示译码器

b.液晶显示器（LCD显示器）：

液晶是一种既有液体的流动性又具有光学特性的有机化合物。它的透明度和呈现的颜色是受外加电场的影响，利用这一点做成七段字符显示器。

七段液晶电极也排列成8字形，当没有外加电场时，由于液晶分子整齐地排列，呈透明状态，射入的光线大部分被返回，显示器呈白色；





显示译码器

当有外加电场，并且选择不同的电极组合并加以电压，由于液晶分子的整齐排列被破坏，呈浑浊状态，射入的光线大部分被吸收，故呈暗灰色，可以显示出各种字符来。

液晶显示器的最大优点是功耗极低，工作电压也低，但亮度很差，另外它的响应速度较低。一般应用在小仪器仪表中。

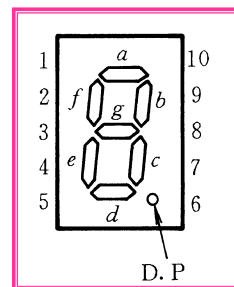
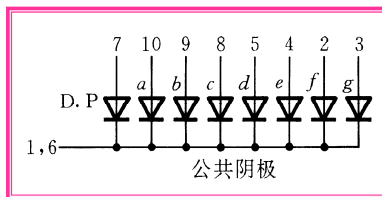
2. BCD- 七段显示译码器

七段数码管需要驱动电路，使其点亮。驱动电路可以是TTL电路或者CMOS电路，其作用是将BCD代码转换成数码管所需要的驱动信号，**共阳极数码管需要低电平驱动；共阴极数码管需要高电平驱动。**

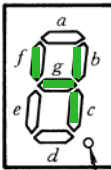


显示译码器

如共阴极数码管BS201A



当某段加高电平时，则点亮，加低电平时，熄灭。那么如果显示某一数字如“3”，则abcdg=11111，fe=00。

数字	输入				输出							字形
	A_3	A_2	A_1	A_0	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0	0	1	1	0	0	1	1	
5	0	1	0	1	1	0	1	1	0	1	1	
6	0	1	1	0	0	0	1	1	1	1	1	
7	0	1	1	1	1	1	1	0	0	0	0	
8	1	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	0	0	1	1	
10	1	0	1	0	0	0	0	1	1	0	1	
11	1	0	1	1	0	0	1	1	0	0	1	
12	1	1	0	0	0	1	0	0	0	1	1	
13	1	1	0	1	1	0	0	1	0	1	1	
14	1	1	1	0	0	0	0	1	1	1	1	
15	1	1	1	1	0	0	0	0	0	0	0	

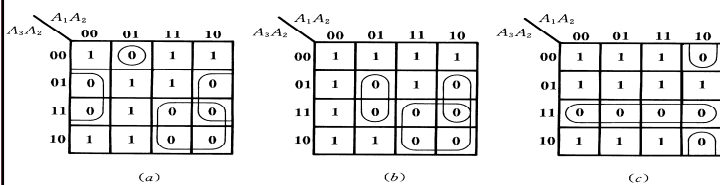
21

D. P

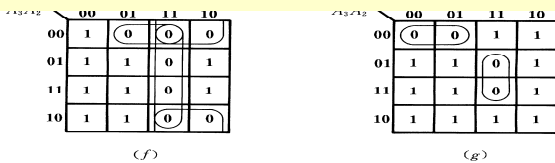
4.3.2 译码器

电子科学与应用物理学院
School of Electronic Science & Applied Physics

显示译码器



从真值表画出 $Y_a \sim Y_g$ 的卡诺图，圈“0”然后求反可得各输出端的逻辑式





显示译码器

各输出端的逻辑式为：

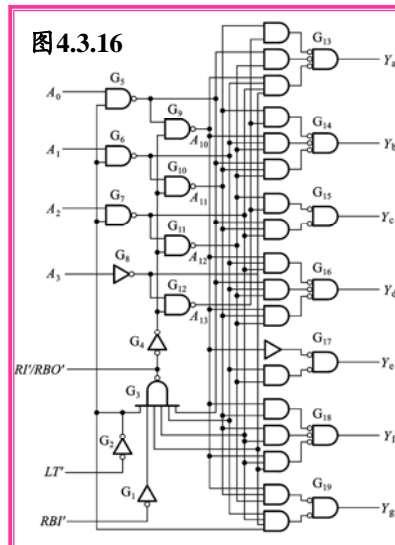
$$\begin{aligned}
 Y_a &= (A_3' A_2' A_1' A_0 + A_3 A_1 + A_2 A_0)' \\
 Y_b &= (A_3 A_1 + A_2 A_1 A_0' + A_2 A_1' A_0)' \\
 Y_c &= (A_3 A_2 + A_2' A_1' A_0)' \\
 Y_d &= (A_2 A_1 A_0 + A_2 A_1' A_0' + A_2' A_1' A_0)' \\
 Y_e &= (A_2 A_1' + A_0)' \\
 Y_f &= (A_3' A_2' A_0 + A_2' A_1 + A_1 A_0)' \\
 Y_g &= (A_3' A_2' A_1' + A_2 A_1 A_0)'
 \end{aligned}$$

注：BCD-七段显示译码器，不是最小项译码器，它是将4位BCD码译成7个代码，广义上也是译码器。



显示译码器

7448就是按照上面的逻辑式设计，并添加一些附加控制端和输出端，集成的BCD—七段显示译码器，可以驱动共阴极数码管。其逻辑图如图4.3.16所示





显示译码器



其逻辑符号如图4.3.17所示

其中：

$A_3 \sim A_0$ ：四位BCD码的输入端

$Y_a \sim Y_g$ ：驱动数码管七段字符的7个输出端



4.3.3 数据选择器(Mux)

一、数据选择器的工作原理

数据选择就是在数字信号的传输过程中，从一组数据中选出某一个来送到输出端，也叫多路开关。

现以双4选1数据选择器
74HC153为例说明数据选择器的工作原理

其内部电路如右图所示

multiplexer: 数据选择器
multiplier: 乘法器

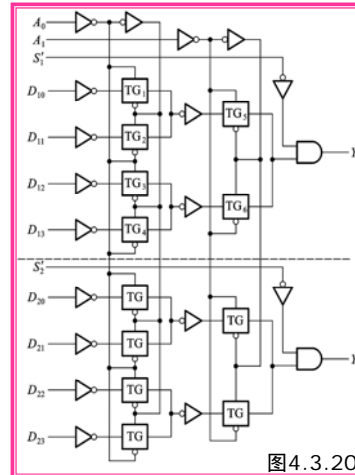
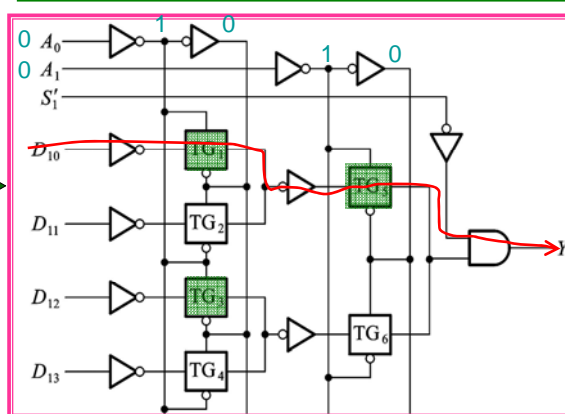
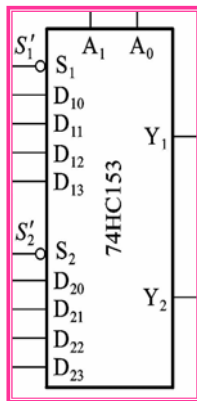


图4.3.20



4.3.3 数据选择器



$$Y_1 = [D_{10}A_1'A_0' + D_{11}A_1'A_0 + D_{12}A_1A_0' + D_{13}A_1A_0]S_1$$





4.3.3 数据选择器

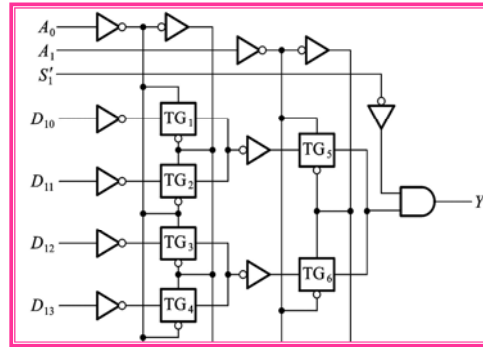
对于一个数据选择器：

$D_{10} \sim D_{13}$: 数据输入端

A_1, A_0 : 选通地址输入端

Y_1 : 输出端

S_1 : 附加控制端



当 $S'_1 = 0, S_1 = 1$ 时，

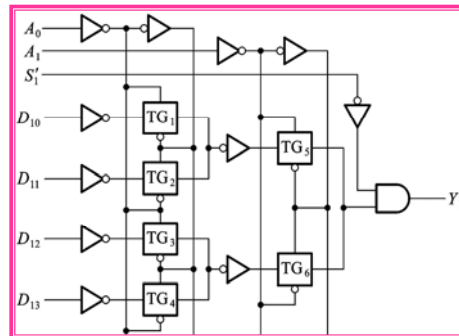
$$Y_1 = D_{10} \bar{A}_1 \bar{A}_0 + D_{11} \bar{A}_1 A_0 + D_{12} A_1 \bar{A}_0 + D_{13} A_1 A_0$$



4.3.3 数据选择器

真值表如下表所示

S'_1	A_1	A_0	Y_1
1	X	X	0
0	0	0	D_{10}
0	0	1	D_{11}
0	1	0	D_{12}
0	1	1	D_{13}





4.3.3 数据选择器

例4.3.4 试用双4选1数据选择器74HC153组成8选1数据选择器。

解：“四选一”只有2位地址输入，从四个输入中选一个；“八选一”的八个数据需要3位地址代码指定其中任何一个，故利用S'做为第3位地址输入端，其实现电路如图4.3.22所示

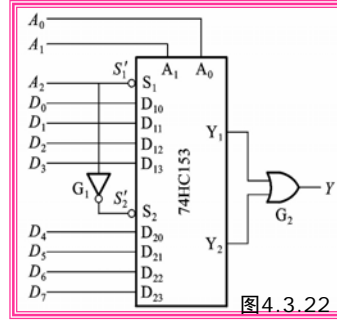


图4.3.22

输出端的逻辑式为

$$Y = (A_2' A_1' A_0') D_0 + (A_2' A_1' A_0) D_1 + (A_2' A_1 A_0') D_2 + (A_2' A_1 A_0) D_3 + (A_2 A_1' A_0') D_4 + (A_2 A_1' A_0) D_5 + (A_2 A_1 A_0') D_6 + (A_2 A_1 A_0) D_7$$

S'	A ₁	A ₀	Y ₁
1	0	0	0
1	X	X	0
0	0	0	D ₁₀
0	0	1	D ₁₁
0	1	0	D ₁₂
0	1	1	D ₁₃

5

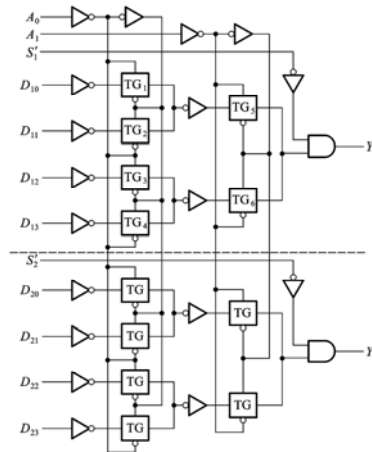


用数据选择器设计组合逻辑电路

对于4选1数据选择器，在S₁=1时，输出与输入的逻辑式为：

$$\text{当 } S_1' = 0, S_1 = 1 \text{ 时,} \\ Y_1 = D_{10} A_1' A_0' + D_{11} A_1' A_0 + D_{12} A_1 A_0' + D_{13} A_1 A_0$$

若将A₁、A₀作为两个输入变量，D₁₀~D₁₃为第三个变量的输入或其他形式，则可由4选1数据选择器实现3变量以下的组合逻辑函数。



6





用数据选择器设计组合逻辑电路

例4.3.5 分别用4选1和8选1数据选择器实现逻辑函数

$$Y = AB' + AC' + A'B'C' + ABC$$

解：（1）用四路数据选择器实现

若将B、C作为地址输入线，A或其他形式作为各数据的输入端，将所给的逻辑函数表示成最小项之和地形式，即

$$\begin{aligned} Y &= AB'(C+C') + AC'(B+B') + A'B'C' + ABC \\ &= AB'C + AB'C' + ABC' + AB'C' + A'B'C' + ABC \\ &= (A+A') \cdot B'C' + A \cdot B'C + A \cdot BC' + A \cdot BC \\ &= 1 \cdot B'C' + A \cdot B'C + A \cdot BC' + A \cdot BC \end{aligned}$$



用数据选择器设计组合逻辑电路

双4选1数据选择器74HC153的一个4选1数据选择器的输出端逻辑函数为

$$\begin{aligned} \text{当 } S'_1 = 0, S_1 = 1 \text{ 时,} \\ Y_1 = D_{10}A_1A'_0 + D_{11}A_1A_0 \\ + D_{12}A_1A'_0 + D_{13}A_1A_0 \end{aligned}$$

则和所给函数相比较得：

$$Y = 1 \cdot B'C' + A \cdot B'C + A \cdot BC' + A \cdot BC$$

令 $A_1 = B$, $A_0 = C$, $D_{10} = 1$, $D_{11} = D_{12} = D_{13} = A$

其电路连线如图4.3.23所示

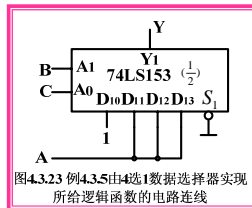
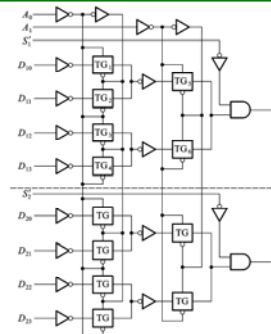


图4.3.23 例4.3.5由4选1数据选择器实现所给逻辑函数的电路连线





用数据选择器设计组合逻辑电路

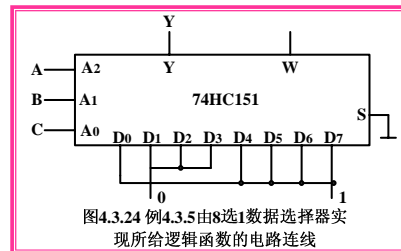
(2) 由8选1数据选择器实现

先将所给逻辑函数写成最小项之和形式，即

$$\begin{aligned}
 Y &= AB' + AC' + A'B'C' + ABC \\
 &= AB'(C + C') + AC'(B + B') + A'B'C' + ABC \\
 &= AB'C + AB'C' + ABC' + AB'C' + A'B'C' + ABC \\
 &= 1 \cdot \overline{A}\overline{B}\overline{C} + 0 \cdot \overline{A}\overline{B}C + 0 \cdot \overline{A}B\overline{C} + 0 \cdot \overline{A}BC \\
 &\quad + 1 \cdot A\overline{B}\overline{C} + 1 \cdot A\overline{B}C + 1 \cdot AB\overline{C} + 1 \cdot ABC
 \end{aligned}$$

令： $A_2 = A$, $A_1 = B$, $A_0 = C$,
 $D_1 = D_2 = D_3 = 0$,
 $D_0 = D_4 = D_5 = D_6 = D_7 = 1$

外部接线图如下图所示





4.3.4 加法器(Adder)

一、1位加法器

1. 半加器(Half Adder)

半加器只考虑两个1位二进制数相加，不考虑低位的进位。

输出端的逻辑式为

$$S = A'B + AB' = A \oplus B$$
$$CO = AB$$

其真值表为

输 入		输 出	
A	B	S	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S: SUM 本位和
CO: Carry Out 向高位的进位。

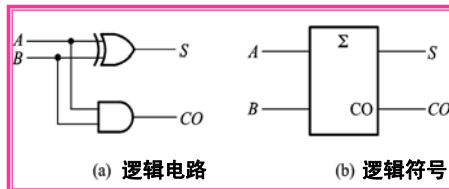
1



4.3.4 加法器

其逻辑电路及逻辑符号如图4.3.26所示

$$S = A'B + AB' = A \oplus B$$
$$CO = AB$$



半加器的逻辑电路及逻辑符号

2





4.3.4 加法器

2. 全加器 (Full Adder)

全加器除了加数和被加数外，还要考虑低位的进位。其真值表如右表

其输出端的逻辑式为

$$\begin{aligned} S &= (A'B'CI' + A'B \cdot CI + AB'CI + ABCI)' \\ &= A \oplus B \oplus CI \\ CO &= (A'B' + B'CI' + A'CI)' = AB + CI \cdot (A \oplus B) \end{aligned}$$

输 入			输 出	
A	B	CI	S	CO
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

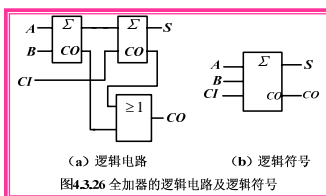
3



4.3.4 加法器

$$\begin{aligned} S &= A \oplus B \oplus CI \\ CO &= AB + CI \cdot (A \oplus B) \end{aligned}$$

由半加器组成的全加器的逻辑电路和逻辑符号如图4.3.27所示



4





4.3.4 加法器

双全加器74LS183的内部电路是按式构建的，如下图所示

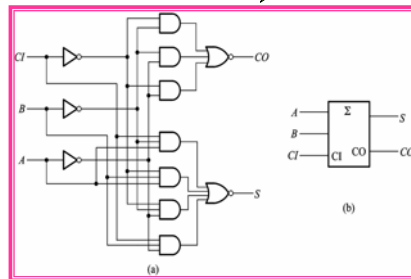
$$S = A \oplus B \oplus CI$$

$$CO = AB + CI \cdot (A \oplus B)$$

合并0再求反

$$S = (A'B'CI' + A'B \cdot CI + AB'CI + ABCI)'$$

$$CO = (A'B' + B'CI' + A'CI)'$$



5



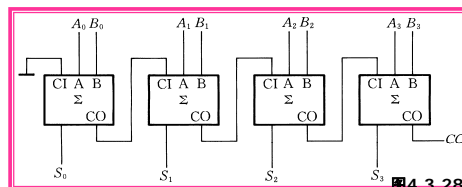
4.3.4 加法器

二、多位加法器

1. 串行进位加法器（行波进位加法器）

两个多位二进制数相加，必须利用全加器，1位二进制数相加用1个全加器，n位二进制数相加用n个全加器。只要将低位的进位输出接到高位的进位输入。

图4.3.28所示电路为4位全加器，由于低位的进位输出接到高位的进位输入，故为串行进位加法器。



6

图4.3.28





4.3.4 加法器

2. 超前进位加法器

为了提高速度，若使进位信号不逐级传递，而是运算开始时，即可得到各位的进位信号，采用这个原理构成的加法器，就是超前进位 (Carry Look-ahead) 加法器，也叫快速进位 (Fast carry) 加法器。

由全加器真值表可知，高位的进位信号的产生是在两种情况下：①在 $A \cdot B = 1$ ；②在 $A + B = 1$ 且 $CI = 1$ 。故向高位的进位信号为

$$(CO)_i = A_i B_i + (A_i + B_i)(CI)_i$$

输入			输出	
A	B	CI	S	CO
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

7



4.3.4 加法器

$$(CO)_i = A_i B_i + (A_i + B_i)(CI)_i$$

设 $G_i = A_i B_i$ 为进位生成函数 (Generate)， $P_i = A_i + B_i$ 为进位传递函数 (Propagate)，则上式可写成

$$\begin{aligned} (CO)_i &= G_i + P_i(CI)_i = G_i + P_i[G_{i-1} + P_{i-1}(CI)_{i-1}] \\ &= G_i + P_i G_{i-1} + P_i P_{i-1} [G_{i-2} + P_{i-2}(CI)_{i-2}] \\ &\dots \\ &= G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \dots + P_i P_{i-1} P_{i-2} \dots P_1 G_0 \\ &\quad + P_i P_{i-1} P_{i-2} \dots P_0 (CI)_0 \end{aligned}$$

和为：

$$S_i = A_i \oplus B_i \oplus (CI)_i$$

8





4.3.4 加法器

74LS283就是采用这种超前进位的原理构成的**4 位超前进位加法器**，其内部电路如图4.3.29所示

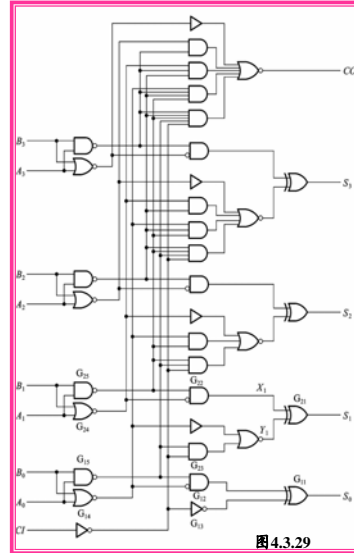


图4.3.29



4.3.4 加法器

逻辑图形符号如图4.3.30所示。

其中： $A_3 \sim A_0$ 为一个四位二进制数的输入； $B_3 \sim B_0$ 为另一个二进制数的输入；CI为最低位的进位；CO是最高位的进位； $S_3 \sim S_0$ 为各位相加后的和。

超前进位加法器提高了运算速度，但同时增加了电路的复杂性，而且位数越多，电路就越复杂。

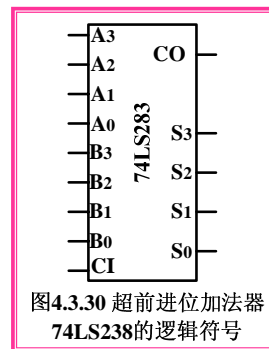


图4.3.30 超前进位加法器
74LS283的逻辑符号



4.3.4 加法器

三、用加法器设计组合逻辑电路

例4.3.8 将BCD的8421码转换为余3码

解：其真值表如右表所示，则

$$Y_3Y_2Y_1Y_0 = DCBA + 0011$$

故实现的电路如图4.3.32所示

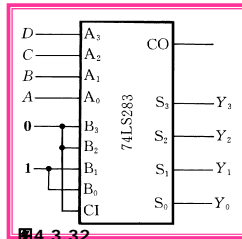


图4.3.32

输 入				输 出			
D	C	B	A	Y3	Y2	Y1	Y0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



4.3.5 数值比较器(Comparator)

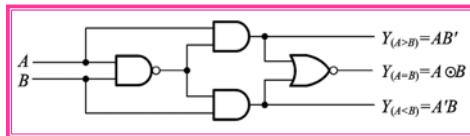
实现比较两个数值大小的逻辑电路即为比较器。

一、1位数值比较器

设有一位二进制数A和B比较有三种可能结果

- * $A > B (A = 1, B = 0)$ 则 $AB' = 1, \therefore Y_{(A>B)} = AB'$
- * $A < B (A = 0, B = 1)$ 则 $A'B = 1, \therefore Y_{(A<B)} = A'B$
- * $A = B (A, B \text{ 同为 } 0 \text{ 或 } 1), \therefore Y_{(A=B)} = (A \oplus B)'$

实现的电路如下图所示



1



4.3.5 数值比较器

1. 多位比较器的原理

在比较两个多位数的大小时，必须自高位向低位逐位比较。

例如：比较两个4为二进制数 $A_3 A_2 A_1 A_0$ 和 $B_3 B_2 B_1 B_0$ ，输出为 $Y_{(A>B)}$ 、 $Y_{(A=B)}$ 和 $Y_{(A<B)}$ 。则

$$Y_{(A<B)} = A'_3 B_3 + (A_3 \oplus B_3)' A'_2 B_2 + (A_3 \oplus B_3)' (A_2 \oplus B_2)' A'_1 B_1 + (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' A'_0 B_0$$

$$Y_{(A=B)} = (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' (A_0 \oplus B_0)'$$

$$Y_{(A>B)} = (Y_{(A<B)} + Y_{(A=B)})'$$

2

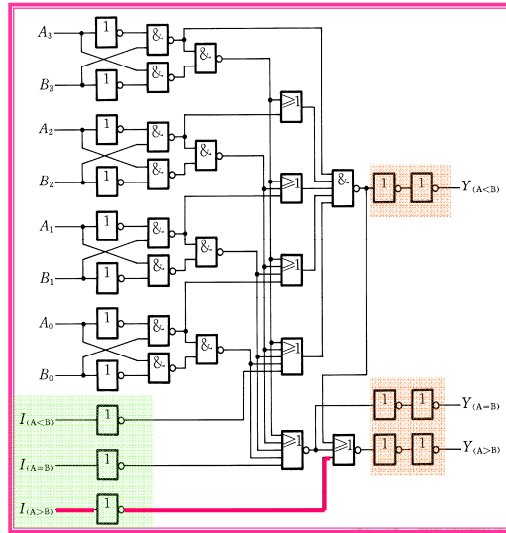




4.3.5 数值比较器

右图为集成4位二进制比较器CC14585的内部逻辑电路

它有三个附加输入端 $I_{(A<B)}$ 、 $I_{(A=B)}$ 和 $I_{(A>B)}$ ，用于扩展



4.3.5 数值比较器

其输出端的逻辑式为

$$\begin{aligned}
 Y_{(A>B)} &= A_3 B_3' + (A_3 \oplus B_3)' A_2 B_2' + (A_3 \oplus B_3)' (A_2 \oplus B_2)' A_1 B_1' \\
 &\quad + (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' A_0 B_0' \\
 &\quad + (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' (A_0 \oplus B_0)' I_{(A>B)} \\
 Y_{(A<B)} &= A_3' B_3 + (A_3 \oplus B_3)' A_2' B_2 + (A_3 \oplus B_3)' (A_2 \oplus B_2)' A_1' B_1 \\
 &\quad + (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' A_0' B_0 \\
 &\quad + (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' (A_0 \oplus B_0)' I_{(A<B)} \\
 Y_{(A=B)} &= (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' (A_0 \oplus B_0)' I_{(A=B)}
 \end{aligned}$$

当比较两个4位数时，没有来自低位的进位结果。应使 $I_{(A<B)} = I_{(A>B)} = 0$ ， $I_{(A=B)} = 1$ 。



4.3.5 数值比较器

例4.3.9 试用两片CC14585组成一个8位数值比较器。

解：其实现的电路如图4.3.35所示。

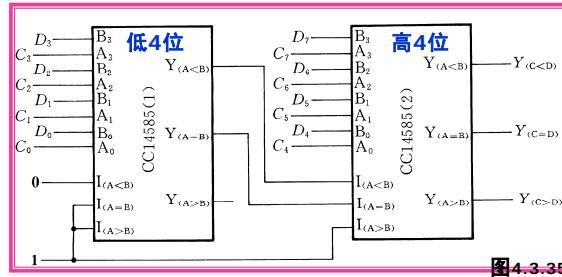


图4.3.35

注意：电路结构不同，扩展输入端的用法也不完全相同。
教材中P₂₀₀是74LS85，接法不一样。



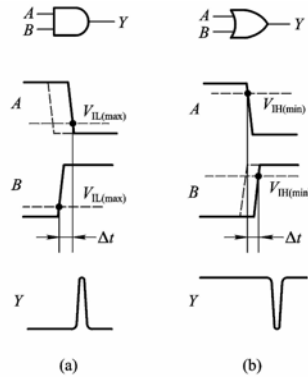
4.4 组合逻辑电路中的竞争-冒险

4.4.1 竞争-冒险现象及成因

一、什么是“竞争”

两个输入“同时向相反的逻辑电平变化”，称存在“竞争”。

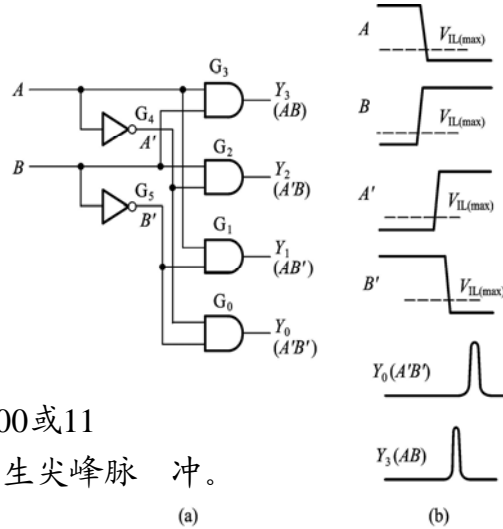
二、因“竞争”而可能在输出产生尖峰脉冲的现象，称为“竞争-冒险”。



1



三、2线-4线译码器中的竞争-冒险



当 AB 从 $10 \rightarrow 01$ 时，

在动态过程中可能出现 00 或 11

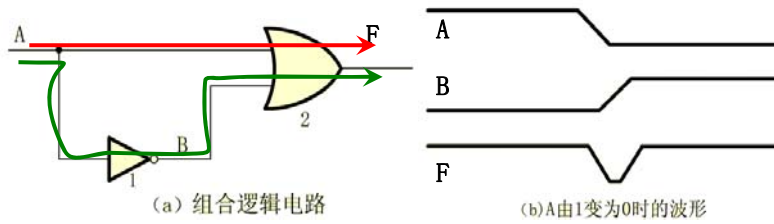
所以 Y_3 和 Y_0 输出端可能产生尖峰脉冲。

2



4.4 组合逻辑电路中的竞争-冒险

竞争冒险的产生原因: 重汇聚 (reconvergence)

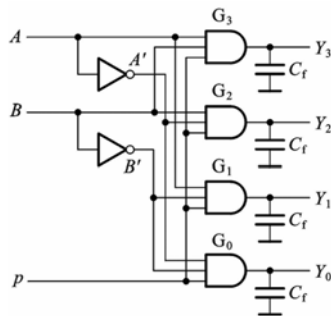


3



消除竞争冒险的方法 (1)

1、加入滤波电容



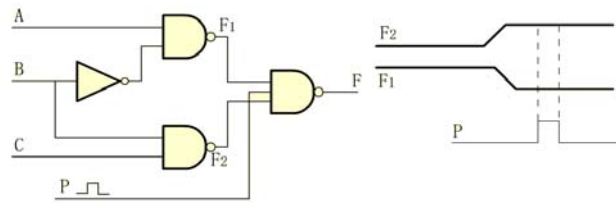
输出电压的波形随之变坏。

4



消除竞争冒险的方法（2）

2、加选通脉冲：



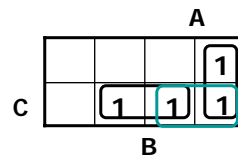
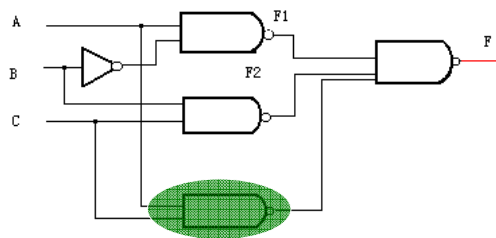
对于选通脉冲的宽度和作用时间都有严格的要求。



消除竞争冒险的方法（3）

3、修改逻辑设计：

$$F = AB' + BC = AB' + BC + AC$$



适用范围有限。



课后作业

P₂₀₉
习题1,3,5,9,12