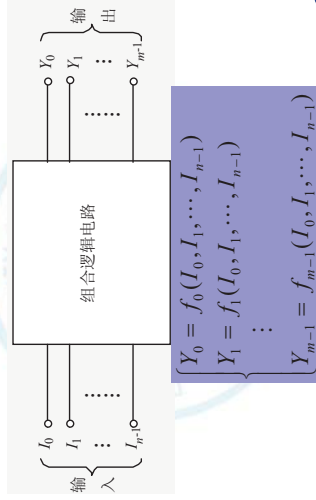


第四章 组合逻辑电路



1

组合电路：输出仅由输入决定，与电路当前状态无关；电路结构中**无**反馈环路（无记忆）



4

引言

- ◆ 组合逻辑电路：指该电路在任一时刻输出的稳定状态，仅取决于该时刻的输入信号，而与输入信号作用前电路所处的状态无关。
- ◆ 特点：从电路结构上看，组合逻辑电路仅由门电路组成，电路中无记忆元件，输入与输出之间无反馈



5

第四章 组合逻辑电路

- 第1节 组合逻辑电路分析
- 第2节 组合逻辑电路设计
- 第3节 编码器
- 第4节 译码器
- 第5节 数据选择器
- 第6节 数值比较器
- 第7节 加法电路
- 第8节 组合逻辑电路的竞争冒险



2

引言

- ◆ 组合逻辑电路：指该电路在任一时刻输出的稳定状态，仅取决于该时刻的输入信号，而与输入信号作用前电路所处的状态无关。
- ◆ 特点：从电路结构上看，组合逻辑电路仅由门电路组成，电路中无记忆元件，输入与输出之间无反馈



5

概念的引入

组合逻辑电路
现时的输出仅取决于现时的输入

逻辑电路

时序逻辑电路
除与现时输入有关外还与原状态有关



3

4.1 组合逻辑电路的分析

- ◆ 分析组合逻辑电路，一般是根据已知的逻辑电路，找出其逻辑函数表达式，或写出其真值表，从而了解其电路的逻辑功能。有时分析的目的在于检验所设计的逻辑电路是否能实现预定的逻辑功能。
- ◆ 分析组合逻辑电路的一般步骤：
 - ① 用文字或符号标出各个门的输入或输出。
 - ② 从输入端到输出端逐级写出输出函数对输入变量的逻辑函数表达式，也可由输出端向输入端逐级推导，最后得到以输入变量表示的输出逻辑函数表达式。
 - ③ 用逻辑代数或卡诺图化简或变换各逻辑函数表达式，或列出真值表。
 - ④ 根据真值表或逻辑函数表达式确定电路的逻辑功能。



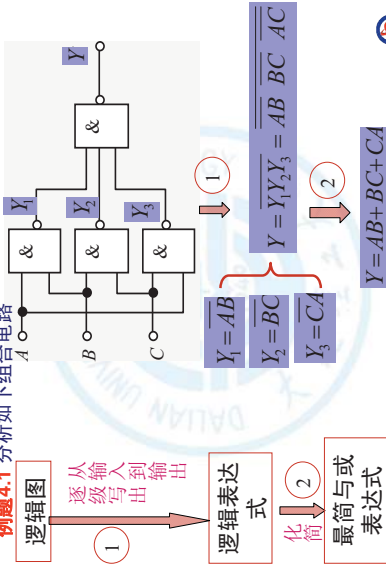
6

分析过程的四个步骤



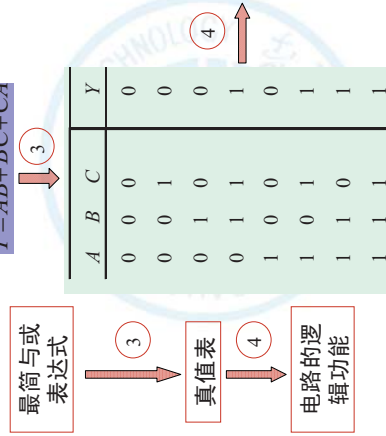
7

例4.1 分析如下组合电路



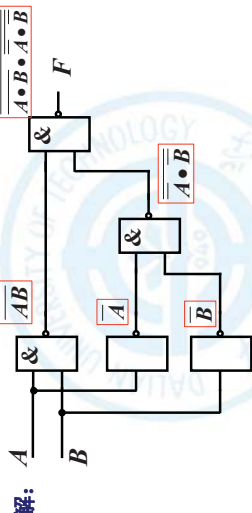
8

例4.1 分析如下组合电路



9

例4.3 分析电路



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

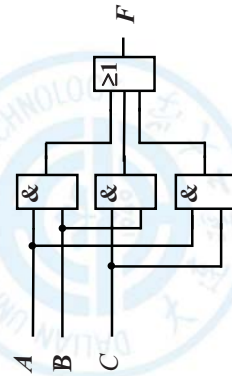
例4.5 设计三人表决电路

设计三人表决电路 (A、B、C)。每人一个按键，如果同意则按下，不同意则不按。结果用指示灯表示，多数同意时指示灯亮，否则不亮。

- 1) 首先指明逻辑符号取“0”、“1”的含义。
三个按键A、B、C按下时为“1”，不按时为“0”。
输出量为 F，多数赞成时是“1”，否则是“0”。
- 2) 根据题意列出逻辑状态表。

4) 根据逻辑表达式画出逻辑图。

$$F = AB + BC + CA$$

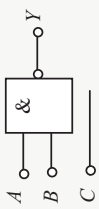


电路的逻辑功能

电路的输出Y只与输入A、B有关，而与输入C无关。Y和A、B的逻辑关系为：A、B中只要一个为0，Y=1；A、B全为1时，Y=0。所以Y和A、B的逻辑关系为与非运算的关系。

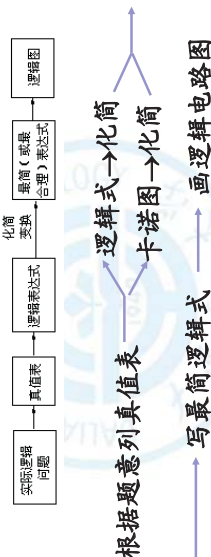
用与非门实现

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

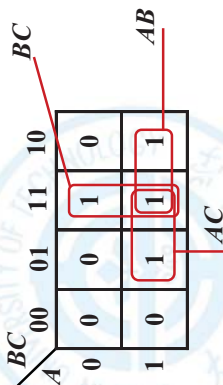


4.2 组合逻辑电路的设计

设计过程的基本步骤:



3) 画出卡诺图:



$$F = AB + BC + CA$$

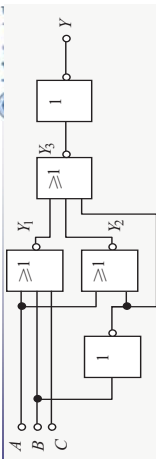
例4.2 分析电路

逻辑图

解:

逻辑表达式

最简与或表达式

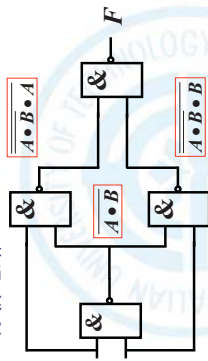


$$\begin{aligned} Y_1 &= A+B+C \\ Y_2 &= A+B \\ Y_3 &= X+Y+B \end{aligned}$$

$$Y = \overline{ABC} + \overline{AB} + \overline{B} = \overline{A} + \overline{B}$$

例4.4 分析电路

解:



$$\begin{aligned} F &= A \cdot B \cdot A \cdot B \cdot B \\ &= A \cdot B \cdot A \cdot B \cdot B \\ &= (\overline{A} + \overline{B}) \cdot A + (\overline{A} + \overline{B}) \cdot B = A \cdot \overline{B} + \overline{A} \cdot B \end{aligned}$$

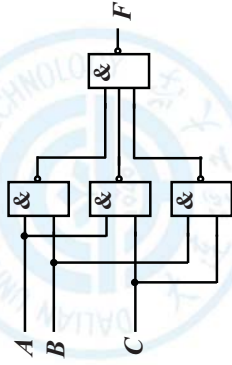
逻辑状态表

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

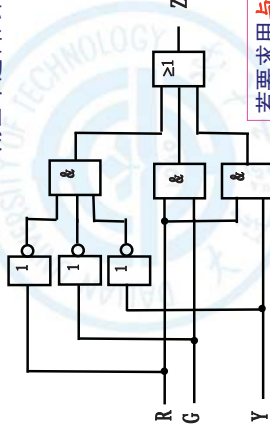
若用与非门实现

$$F = AB + BC + CA$$

$$= \overline{AB + BC + CA} = \overline{AB} \cdot \overline{BC} \cdot \overline{CA}$$



用基本逻辑门构成逻辑电路



若要求用与非门构成逻辑电路呢?

(2) 列真值表: 如表3.2所示。

A	B	C	Y	G
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

例4.6 交通灯故障监测逻辑电路的设计

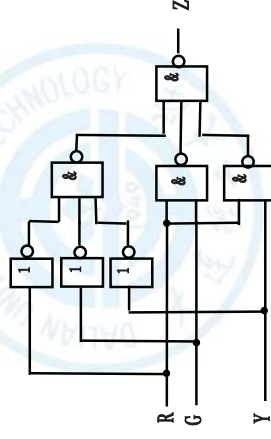
红灯R
黄灯Y
绿灯G

单灯亮→正常
黄、绿同时亮→正常
其它情况→不正常

用与非门构成逻辑电路

$$Z = \overline{RY} + \overline{RG} + \overline{RY} = \overline{RY} + \overline{RG} + \overline{RY} \cdot \overline{RY}$$

(利用反演定理 $A+B=\overline{\overline{A}\cdot\overline{B}}$, $A+B+C=\overline{\overline{A}\cdot\overline{B}\cdot\overline{C}}$)



(3) 化简: 利用卡诺图化简, 如图3.4所示可得:

$$Y = BC + AC + AB$$

$$G = \overline{ABC} + \overline{ABC} + \overline{ABC} + ABC$$

$$= \overline{A}(B \oplus C) + A(B \odot C)$$

$$= A \oplus B \oplus C$$

(4) 画逻辑图: 逻辑电路图如下图所示。若要求用 TTL 与非门, 实现该设计电路的设计步骤如下: 首先, 将化简后的与或逻辑表达式转换为与非形式; 然后再画出如下图所示的逻辑图; 最后, 画出用与非门实现的组合逻辑电路。

设: 灯亮为“1”, 不亮为“0”, 单灯亮→正常
黄、绿同时亮→正常
其它情况→不正常

1、列真值表 2、卡诺图→化简

R	Y	G	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

例4.7 设计如下要求的电路

■ 有三个班学生上自习, 大教室能容纳两个班学生, 小教室能容纳一班学生。设计两个教室是否开灯的逻辑控制电路, 要求如下:

(1) 一个班学生上自习, 开小教室的灯。

(2) 两个班上自习, 开大教室的灯。

(3) 三个班上自习, 两教室均开灯。

解: (1) 确定输入、输出变量的个数: 根据电路要求, 设输入变量 A、B、C 分别表示三个班学生是否上自习, 1 表示上自习, 0 表示不上自习; 输出变量 Y、G 分别表示大教室、小教室的灯是否亮, 1 表示亮, 0 表示灭。

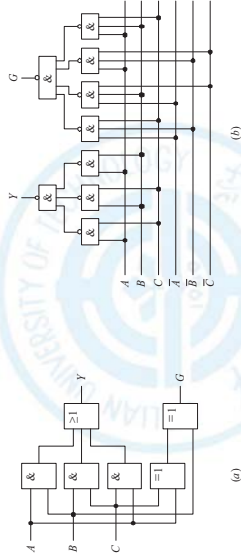
$$Y = AC + BC + AB$$

$$= \overline{AC} \cdot \overline{BC} \cdot \overline{AB}$$

$$G = \overline{ABC} + \overline{ABC} + \overline{ABC} + ABC$$

$$= \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC}$$

Y	BC	00	01	11	10
0	0	0	0	1	0
1	0	0	1	1	1



(a) 直接实现;

(b) 用与非门实现

例 4.8 设计复合如下要求的电路

设计一个楼上、楼下开关的控制逻辑电路来控制楼梯上的路灯，使之在上楼前，用楼下开关打开电灯，上楼后，用楼上开关关闭电灯；或者在下楼前，用楼上开关打开电灯，下楼后，用楼下开关关闭电灯。

4.3 编码器

所谓编码就是将特定含义的输入信号（文字、数字、符号）转换成二进制代码的过程。实现编码操作的数字电路称为编码器。按照编码方式不同，编码器可分为普通编码器和优先编码器；按照输出代码种类的不同，可分为二进制编码器和非二进制编码器。

一般而言，N个不同的信号，至少需要n位二进制数编码。N和n之间满足下列关系：

$$2^n \geq N$$

解:

电路功能描述

设楼上开关为A，楼下开关为B，灯泡为Y。并设A、B闭合时为1，断开时为0；灯亮时Y为1，灯灭时Y为0。根据逻辑要求列出真值表。

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

真值表

4.3.1 普通编码器

例 4.9 设计一个4线-2线编码器（用或门）

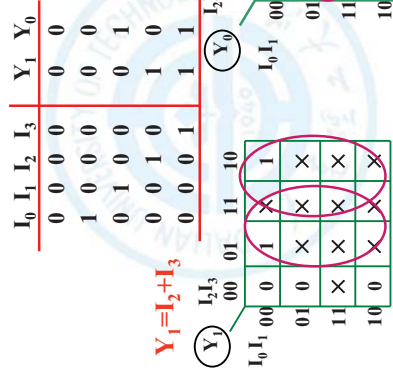
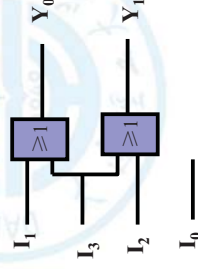
解:

(1) 确定输入、输出变量个数：由题意知输入为I₀、I₁、I₂、I₃四个信息，输出为Y₀、Y₁，当I_i编码时为1，不编码为0，并依此按I_i下角标的值与Y₀、Y₁二进制代码的值相对应进行编码。

简化的真值表

Y ₁	Y ₀
I ₀	0 0
I ₁	0 1
I ₂	1 0
I ₃	1 1

逻辑图:



逻辑表达式
或卡诺图
化简

已为最简与或表达式
 $Y = \overline{A}B + A\overline{B}$

用与非门实现
 $Y = \overline{\overline{\overline{A}B} \cdot \overline{A\overline{B}}}$

用异或门实现
 $Y = A \oplus B$

逻辑电路图

完整的真值表

I ₀	I ₁	I ₂	I ₃	Y ₁	Y ₀
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	1	0
0	0	1	0	0	1
0	0	0	1	1	1
0	0	1	1	1	1

(2) 列编码表: 如表所示。

I_i	Y_1	Y_0
I_0	0	0
I_1	0	1
I_2	1	0
I_3	1	1

(3) 化简

$$Y_0 = I_1 + I_3$$

$$Y_1 = I_2 + I_3$$

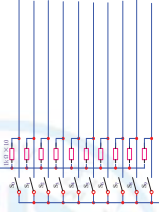
例题4.10 设计8线—3线编码器(用与非门)

设八个输入端为 $I_1 \sim I_8$, 八种状态, 与之对应的输出设为 F_1 、 F_2 、 F_3 , 共三位二进制数。

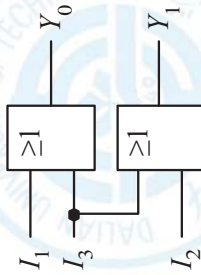
设计编码器的过程与设计一般的组合逻辑电路相同, 首先要列出状态表, 然后写出逻辑表达式并进行化简, 最后画出逻辑图。

例题4.11 设计一个键盘输入8421BCD码编码器

入					出				
入	出	入	出	入	出	入	出	入	出
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1	0
1	1	1							



(4) 画编码器电路如图所示。



真值表

I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	F_1	F_2	F_3
0	1	1	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1

$$F_1 = I_2 + \overline{I_4} + \overline{I_6} + \overline{I_8} = \overline{I_2} \cdot \overline{I_4} \cdot \overline{I_6} \cdot \overline{I_8}$$

$$F_2 = \overline{I_3} \cdot \overline{I_7} \cdot \overline{I_8} \quad F_3 = \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7}$$

解: (1) 列出真值表:

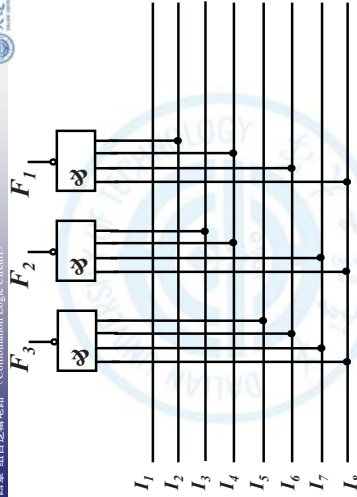
入					出				
S_1^1	S_2^1	S_3^1	S_4^1	S_5^1	S_1^2	S_2^2	S_3^2	S_4^2	S_5^2
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1			

(2) 由真值表写出各输出的逻辑表达式为:

$$A = \overline{S_8} \cdot \overline{S_9} = \overline{S_8} \cdot \overline{S_9} \quad B = \overline{S_4} + \overline{S_5} + \overline{S_6} + \overline{S_7} = \overline{S_4} \cdot \overline{S_5} \cdot \overline{S_6} \cdot \overline{S_7}$$

二线编码器功能

按以黄色字形的按钮可看到编码器的输出



8-3译码器逻辑图

$$C = \overline{S_2} + \overline{S_3} + \overline{S_6} + \overline{S_7} = \overline{S_2} \cdot \overline{S_3} \cdot \overline{S_6} \cdot \overline{S_7}$$

$$D = \overline{S_1} + \overline{S_3} + \overline{S_5} + \overline{S_7} + \overline{S_9} = \overline{S_1} \cdot \overline{S_3} \cdot \overline{S_5} \cdot \overline{S_7} \cdot \overline{S_9}$$

重新整理得:

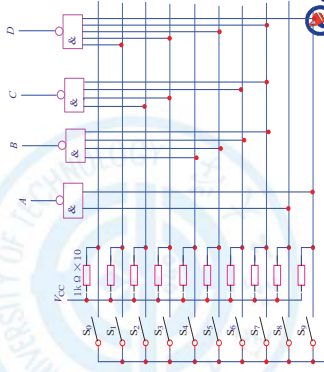
$$A = \overline{S_8} \cdot \overline{S_9}$$

$$B = \overline{S_4} \cdot \overline{S_5} \cdot \overline{S_6} \cdot \overline{S_7}$$

$$C = \overline{S_2} \cdot \overline{S_3} \cdot \overline{S_6} \cdot \overline{S_7}$$

$$D = \overline{S_1} \cdot \overline{S_3} \cdot \overline{S_5} \cdot \overline{S_7} \cdot \overline{S_9}$$

(3) 由表达式画出逻辑图:



(4) 增加控制使能标志GS :

当按下 $S_0 \sim S_9$

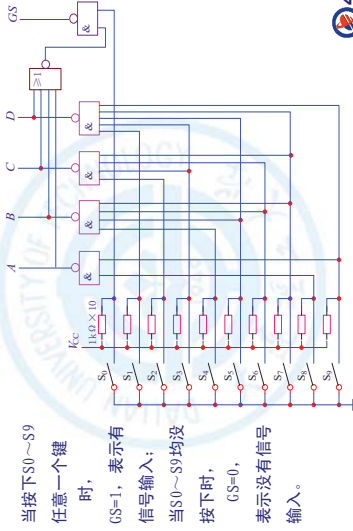
任意一个键时,

$GS=1$, 表示有信号输入;

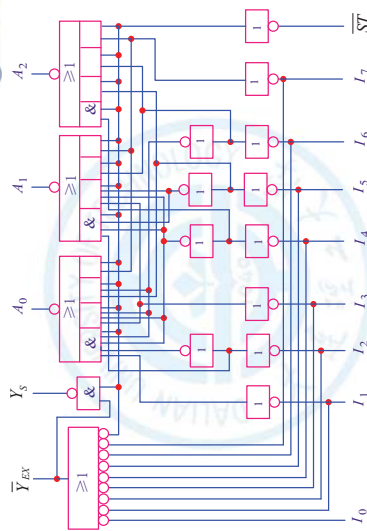
当 $S_0 \sim S_9$ 均没按下时,

$GS=0$,

表示没有信号输入。



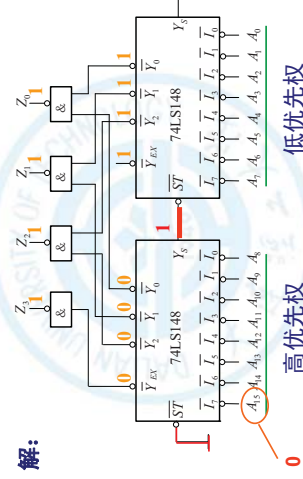
46



49

例题4.12 请用 TTL74148 实现 16-4 线编码器

解:

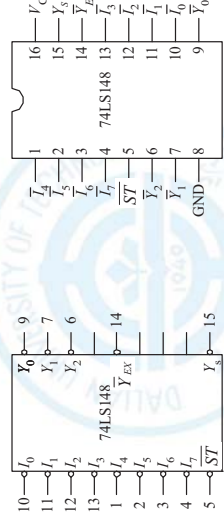


52

4.3.2 优先编码器

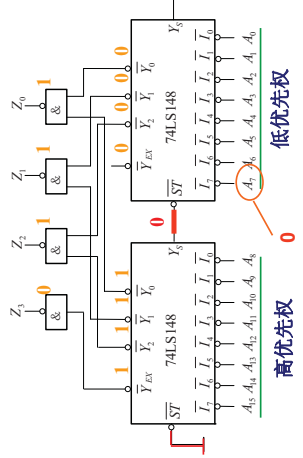
- ◆ 集成优先编码器举例——74148 (8线-3线)
- ◆ 注意: 该电路为反码输出。 \overline{ST} 为使能输入端(低电平有效), Y_5 为使能输出端(高电平有效), $\overline{Y_{EX}}$ 为优先编码工作标志(低电平有效)。

47



(a) 符号图
(b) 管脚图

50



53

增加优先级

设 I_7 的优先级最高, I_0 最低。

增加标志位 Y_5

有编码请求时 $Y_5=1$, 以区分无编码请求和 I_0 有效。

输入	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	$\overline{Y_{EX}}$	Y_5
1	x	x	x	x	x	x	x	x	1	1
0	1	1	1	1	1	1	1	1	0	0
0	x	x	x	x	x	x	x	0	0	0
0	x	x	x	x	x	x	0	1	0	1
0	x	x	x	x	0	1	1	0	1	0
0	x	x	x	0	1	1	1	0	1	0
0	x	x	0	1	1	1	1	0	1	0
0	x	x	0	1	1	1	1	1	0	0
0	x	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	0

48

一个 74LS148 的工作情况



51

4.4 译码器

译码器的基本概念及工作原理

译码器——将输入代码转换成特定的输出信号

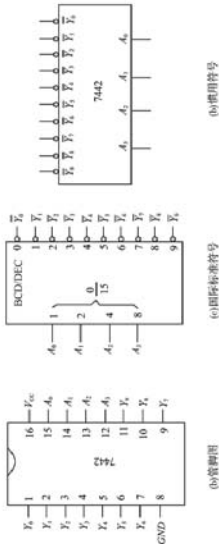
下面是一个2线-4线译码器

表 4.2.1 2线-4线译码器功能表

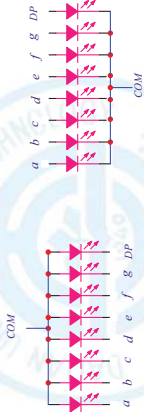
输入	A	B	Y_0	Y_1	Y_2	Y_3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

54

4线10线译码器7442管脚图和符号图

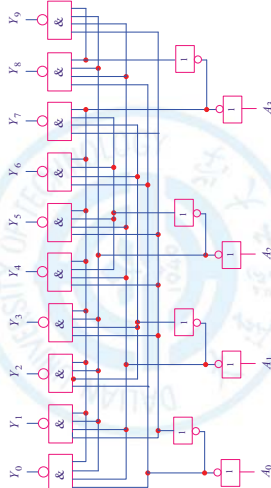


内部连接方式不同，七段数字显示器分为共阴极和共阳极。

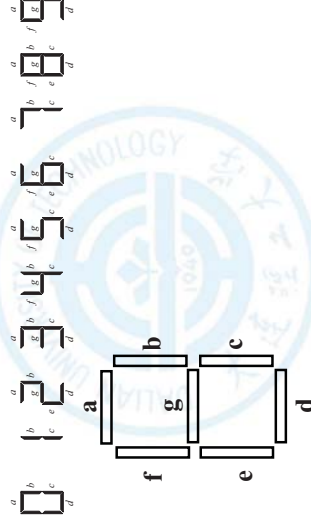


7448的逻辑功能：

- (1) **正常译码显示**。当 $LT=1$, $B/RBO=1$ 时，对输入为十进制数 ~ 15 的二进制码 (0001~1111) 进行译码，产生对应的七段显示码。
- (2) **灭零**。当 $LT=1$ ，而输入为0的二进制码0000时，只有当 $RBI=1$ 时，才产生0的七段显示码。如果此时输入 $RBI=0$ ，则译码器的 $a\sim g$ 输出全0，使显示器全灭；所以 RBI 称为**灭零输入端**。
- (3) **试灯**。当 $LT=0$ 时，无论输入怎样， $a\sim g$ 输出全1，数码管七段全亮，由此可以检测显示器七个发光管的好坏， LT 称为**试灯输入端**。
- (4) **特殊控制端 B/RBO** 。 B/RBO 可以作输入端，也可以作输出端。
作输入端使用时，如果 $B/RBO=0$ 时，不管其他输入端为何值， $a\sim g$ 均输出0，显示器全灭。因此 B/RBO 称为**灭灯输入端**。
作输出端使用时，受控于 RBI 。当 $RBI=0$ ，输入为0的二进制码0000时， $RBO=0$ ，用以指示该片正处于灭零状态。所以， RBO 又称为**灭零输出端**。



显示器件：常用的是七段显示器



七段显示译码器7448功能表

十进制 功能	输入				B/RBO	输出						
	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	1	0
1	1	1	X _c	0	0	0	0	1	1	1	0	0
2	1	1	X _c	0	0	1	0	1	1	1	0	0
3	1	1	X _c	0	1	1	0	1	1	1	0	0
4	1	1	X _c	0	1	0	0	1	1	1	0	1
5	1	1	X _c	0	1	0	1	1	1	0	1	1
6	1	1	X _c	0	1	1	0	1	1	0	1	1
7	1	1	X _c	0	1	1	1	0	1	0	0	0
8	1	1	X _c	1	0	0	0	1	1	1	1	1
9	1	1	X _c	1	0	0	1	1	1	0	1	1
10	1	1	X _c	1	0	1	1	1	0	0	1	1
11	1	1	X _c	1	1	0	1	1	0	0	1	1
12	1	1	X _c	1	1	0	0	1	1	0	0	1
13	1	1	X _c	1	1	1	0	1	1	0	0	1
14	1	1	X _c	1	1	1	1	0	0	0	1	1
15	1	1	X _c	1	1	1	1	1	0	0	0	0
熄灭/灯测试	X	X	X	X	0 (输入)	0	0	0	0	0	0	0
灯测试	0	X	X	X	0	1	1	1	1	1	1	1

4线10线译码器7442功能表

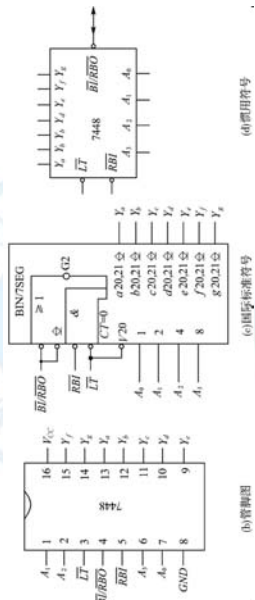
十进制	输入	输出
	$A_3 A_2 A_1 A_0$	$Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8 Y_9$
0	0 0 0 0	1 1 1 1 1 1 1 1 1 1
1	0 0 0 1	1 1 1 1 1 1 1 1 1 1
2	0 0 1 0	1 1 1 1 1 1 1 1 1 1
3	0 0 1 1	1 1 1 1 1 1 1 1 1 1
4	0 1 0 0	1 1 1 1 1 1 1 1 1 1
5	0 1 0 1	1 1 1 1 1 1 1 1 1 1
6	0 1 1 0	1 1 1 1 1 1 1 1 1 1
7	0 1 1 1	1 1 1 1 1 1 1 1 1 1
8	1 0 0 0	1 1 1 1 1 1 1 1 1 1
9	1 0 0 1	1 1 1 1 1 1 1 1 1 1

4.4.3 显示译码器

在数字系统中，常常需要将运算结果用人们习惯的十进制显示出来，这就要用到**显示译码器**



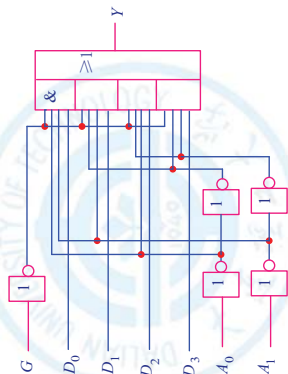
- 七段显示译码器7448是一种与共阴极数字显示器配合使用的集成译码器。



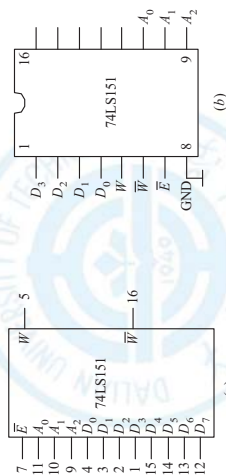
数字显示译码器简介

下按
一停
再按
上本
页按
三角
键放
：按

由逻辑表达式画出逻辑图



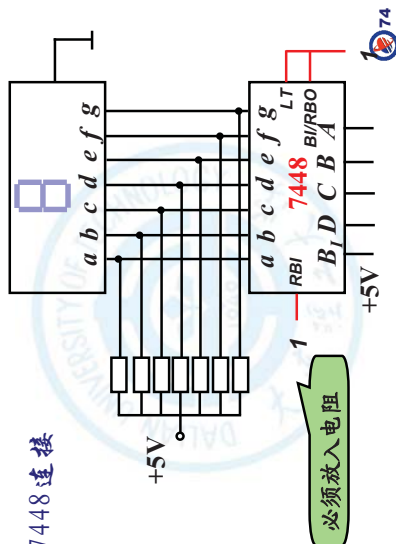
74LS151 数据选择器



(a) 符号图;

(b) 管脚图

7448连接



例4.17 设计一个四选—数据选择器

解:

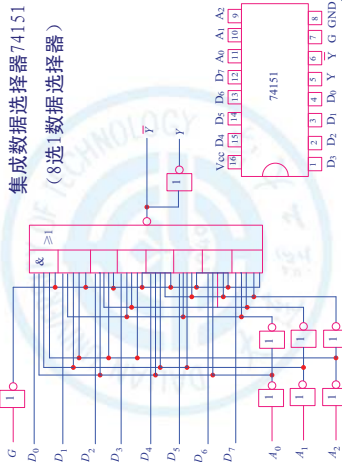
输入		输出	
A_1	A_0	D_3	Y
1	x	x	x
0	0	x	0
0	1	x	1
1	0	x	0
1	1	x	1

根据功能表, 可写出输出逻辑表达式:

$$Y = (A_1 A_0 D_0 + A_1 A_0 D_1 + A_1 A_0 D_2 + A_1 A_0 D_3) \cdot \bar{G}$$

4.5.1 集成数据选择器

集成数据选择器 74151
(8选1数据选择器)



- 将BI/RBO和RBI配合使用, 可以实现多位数显示时的“无效0消隐”功能。

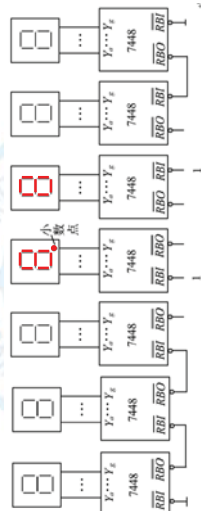


图 4-29 有受控的数码显示系统示意图。

4.5 数据选择器

数据选择器的基本概念及工作原理

数据选择器——根据地址选择码从多路输入数据中选择一个, 送到输出。



数据选择器功能总结

下按
一停
再按
上本
页按
三角
键放
：按

8选1数据选择器74151功能表

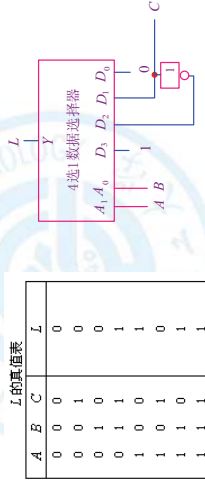
使能	输入			输出	
	A_2	A_1	A_0	Y	\bar{Y}
1	×	×	×	0	1
0	0	0	0	D_0	\bar{D}_0
0	0	0	1	D_1	\bar{D}_1
0	0	1	0	D_2	\bar{D}_2
0	0	1	1	D_3	\bar{D}_3
0	1	0	0	D_4	\bar{D}_4
0	1	0	1	D_5	\bar{D}_5
0	1	1	0	D_6	\bar{D}_6
0	1	1	1	D_7	\bar{D}_7

82

注意二：当逻辑函数的变量个数大于数据选择器的地址输入变量个数时。

$$L = AB + BC + AC$$

解：将A、B接到地址输入端，C加到适当的数据输入端。作出逻辑函数L的真值表，根据真值表画出连线图。



85

4.6 数值比较器

- 在数字系统中，特别是在计算机中，经常需要比较两个数A和B的大小，数字比较器就是对两个位数相同的二进制数A、B进行比较，其结果有A>B、A<B和A=B三种可能性。
- 用来完成两个二进制数的大小比较的逻辑电路称为数值比较器，简称比较器。

88

例题4.18 用两片74LS151连接成一个十六选一的数

据选择器

解：

十六选一的数据选择器的地址输入端有四位，最高位A3的输入可以由两片八选一数据选择器的使能端接非门来实现，低三位地址输入端由两片74LS151的地址输入端相连而成。

当A3=0时，低位片74LS151工作，根据地址控制信号A3A2A1A0选择数据D0~D7输出；A3=1时，高位片工作，选择D8~D15进行输出。

83

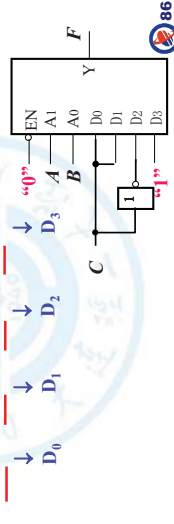
例题4.21 用4-1数据选择器实现下式

$$F(A, B, C) = \sum m(1, 3, 4, 6, 7)$$

解：4-1数据选择器

$$Y = \bar{A}_1\bar{A}_0D_0 + \bar{A}_1A_0D_1 + A_1\bar{A}_0D_2 + A_1A_0D_3$$

$$F(A, B, C) = \bar{A}BC + \bar{A}BC + \bar{A}BC + \bar{A}BC + ABC = \bar{A}BC + \bar{A}BC + \bar{A}BC + \bar{A}BC + ABC$$



86

4.6.1 一位数值比较器

设计比较两个一位二进制数A和B大小的数字电路，输入变量是两个比较数A和B，输出变量Y A>B、Y A<B、Y A=B分别表示A>B、A<B和A=B三种比较结果，其真值表如下所示。

设A>B时L1=1；A<B时L2=1；A=B时L3=1。得1位数值比较器的真值表。

89

4.5.2 数据选择器实现逻辑函数

注意一：输入变量个数相同时，可直接用数据选择器来实现逻辑函数。

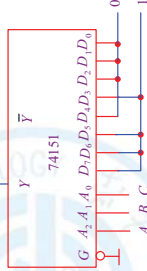
例题 试用8选1数据选择器74151实现逻辑函数：

$$L = AB + BC + AC$$

解：将逻辑函数转换成最小项表达式：

$$L = \bar{A}BC + \bar{A}BC + \bar{A}BC + \bar{A}BC + ABC = m_3 + m_5 + m_6 + m_7$$

画出连线图。



84

总结



用n位输入的数据选择器，可以产生任何一种输入变量数不大于n+1的组合逻辑函数。



设计时可采用函数式比较法。控制端作为输入端，数据输入端可以综合为一个输入端。

87

一位数值比较器真值表

A	B	$L_1(A>B)$	$L_2(A<B)$	$L_3(A=B)$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

90

一位数字比较器表达式

输入		输出		
A	B	$Y_{A \oplus B}$	$Y_{A \cdot B}$	$Y_{A \oplus B}$
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

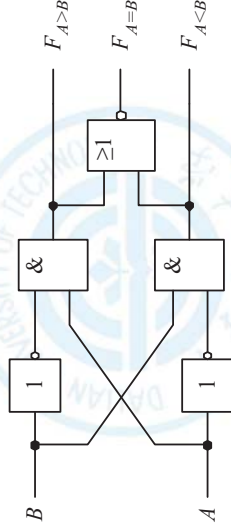
根据真值表写出逻辑表达式:

$$\begin{aligned} Y_{A>B} &= A \overline{B} \\ Y_{A<B} &= \overline{A} B \\ Y_{A=B} &= AB + \overline{A\overline{B}} = A \oplus B \end{aligned}$$

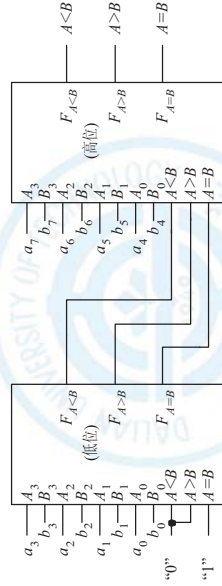
4.6.3 数值比较器的位数扩展

74LS85 数字比较器的串级输入端 $I_A > B$ 、 $I_A < B$ 、 $I_A = B$ 是为了扩大比较器功能设置的, 当不需要扩大比较位数时, $I_A > B$ 、 $I_A < B$ 接低电平, $I_A = B$ 接高电平; 若需要扩大比较器的位数时, 只要将低位的 $F_A > B$ 、 $F_A < B$ 和 $F_A = B$ 分别接高位相应的串级输入端 $I_A > B$ 、 $I_A < B$ 、 $I_A = B$ 即可。

■ 由逻辑表达式画出逻辑图



兩片74LS85擴展連接圖



4.7.1 半加器

只能进行本位加数、被加数的加法运算而
不考虑低位进位。

	A	B	C	S
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

$$S = \overline{AB} + \overline{AB} = A \oplus B$$

4.6.2 四位数值比较器7485

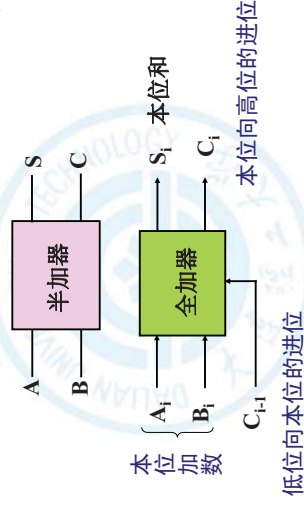
比较输入			级联输入		输出	
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$A > B^*$	$A < B^*$	$A = B^*$
$A_3 > B_3$	\times	\times	\times	\times	\times	1
$A_3 < B_3$	\times	\times	\times	\times	\times	0
$A_3 = B_3$	$A_2 > B_2$	\times	\times	\times	\times	0
$A_3 = B_3$	$A_2 < B_2$	\times	\times	\times	\times	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	\times	\times	\times	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	\times	\times	\times	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	\times	\times	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	\times	\times	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	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

4.7 加法电路

加法器——实现两个二进制数的加法运算

4.7.2 全加器

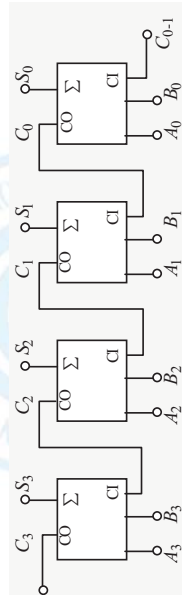
能同时进行本位数和相邻低位的进位信号的加法运算



4.7.3 超前进位加法器 74283

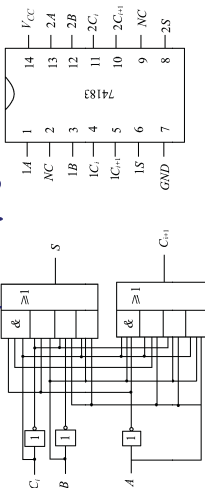
串行进位加法器

构成: 把n位全加器串联起来, 低位全加器的进位输出连接到相邻的高位全加器的进位输入。

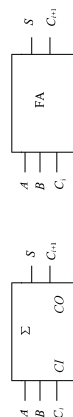


特点: 进位信号是由低位向高位逐级传递的, 速度不高。

全加器逻辑图



(a) 逻辑图



(b) 管脚图

(c) 常用符号

能对两个1位二进制数进行相加并考虑低位来的进位, 即相当于3个1位二进制数相加, 求得和及进位的逻辑电路称全加器。

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

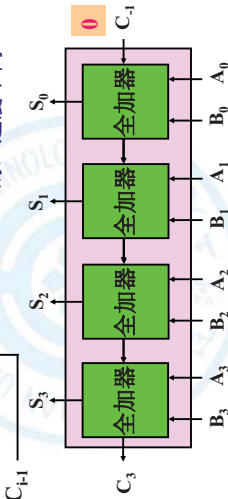
A_i, B_i : 加数, C_{i-1} : 低位来的进位, S_i : 本位的和, C_i : 向高位的进位。

$$S_i = m_0 + m_1 + m_2 + m_3 = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = m_3 + m_2 + m_1 + m_0 = (A_i \oplus B_i)C_{i-1} + A_i B_i$$

用4个全加器构成一个4位二进制加法器

特点: 进位信号是由低位向高位逐级传递的, 速度不高。

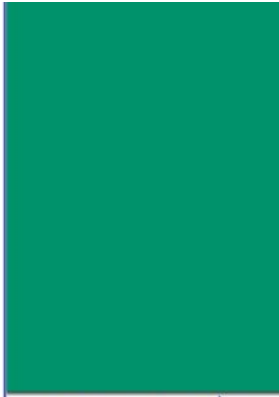


74LS83

串行加法器实验演示

串行加法器

按左面的字, 右面的操作按照提



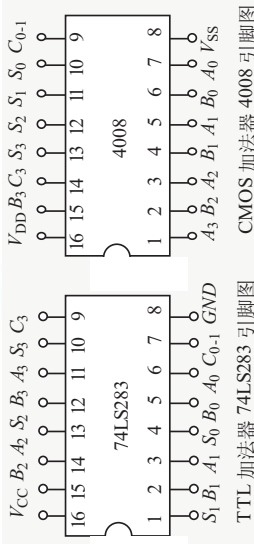
实验目的
实验组件
实验内容
方案设计
方案验证
结论思考
返回

4.8 组合逻辑电路的竞争冒险

A. 产生竞争和冒险的原因

- 组合电路中, 若某个变量通过两条以上途径到达输入端。由于每条路径上的延迟时间不同, 到达逻辑门的时间就有先后, 这种现象称为竞争。由于竞争, 就有可能使真值表描述的逻辑关系受到短暂的破坏, 在输出端产生错误结果, 这种现象称为冒险。

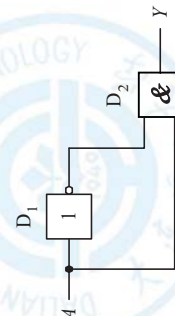
集成2进制4位超前进位加法器



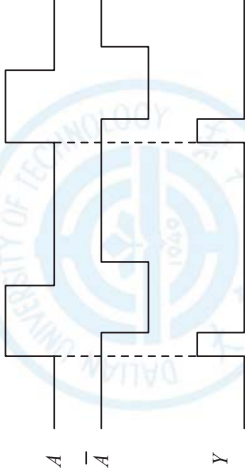
TTL 加法器 74LS283 引脚图

CMOS 加法器 4008 引脚图

其逻辑表达式 $Y = A \cdot \bar{A}$, 由于D1的延迟, 的输入要滞后于A的输入, 致使D2的输出出现一个高电平窄脉冲。



延迟的出现造成错误输出



09

消除竞争冒险的方法

(A) 增加乘积项。
例如: 与或表达式 $Y = AB + \bar{B}C$ 中, 当 $A=1$, $C=1$ 时, $Y = B + \bar{B}$, 此时若直接连成逻辑电路, 可能存在“0”型冒险。可以在该式中增加多余项, 变换为 $Y = AB + \bar{B}C + AC$, 当 $A=C=1$ 时, $Y=1$, 克服了“0”型冒险。

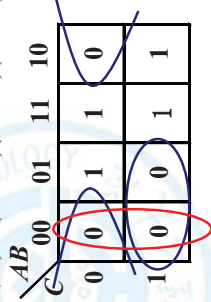
12

例题4.26 判断下图的竞争冒险类型

$$N_2(A, B, C) = (A + \bar{C})(B + C)$$

解: 其卡诺图如上, 有相切“0”圈, 存在1态冒险, 加入红圈覆盖。从而,

$$N_2(A, B, C) = (A + \bar{C})(B + C)(A + B)$$



15

冒险的分类

所示出现高电平窄脉冲, 这种冒险也称为“1”型冒险。使输出出现低电平窄脉冲, 这种冒险称为“0”型冒险。

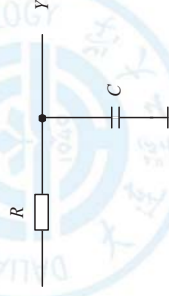
C. 判断冒险的方法

(A) 代数法。

可以用公式法判断是否有冒险, 例如 $Y = AC + B\bar{C}$, 其中 C 有原变量和反变量, 改变 A 、 B 的取值判断是否出现冒险。 $A=1, B=1$ 时, $Y = C + \bar{C}$ 有“0”型冒险。因此, $Y = AC + \bar{B}C$ 会出现“0”型冒险。

10

(B) 输出端并联电容器。如果逻辑电路在较慢速度下工作, 为了消除竞争冒险, 可以在输出端并联一个电容器, 如图3.37所示。由于加电容会影响电路的工作速度, 故电容量的选取要合适, 通常靠试验来调试确定。



13

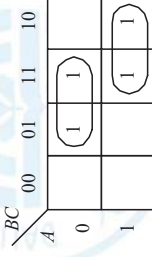
本章总结

- ◆ 掌握组合逻辑电路的分析;
- ◆ 掌握组合逻辑电路的设计;
- ◆ 会灵活使用编码器;
- ◆ 会灵活使用译码器;
- ◆ 会灵活使用数据选择器;
- ◆ 会灵活使用数值比较器;
- ◆ 理解加法电路原理;
- ◆ 会判断组合电路的竞争和冒险。

16

(B) 卡诺图法。

如图3.36所示, 图中的卡诺图相切则有竞争冒险, 如圈“1”则为“0”型冒险, 而圈“0”则为“1”型冒险, 当卡诺圈相交或相离时均无竞争冒险产生。



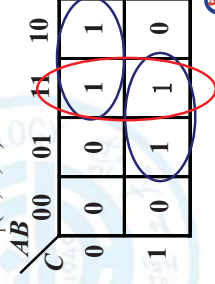
11

例题4.25 判断下图的竞争冒险类型

$$N_1(A, B, C) = \bar{A}\bar{C} + BC$$

解: 其卡诺图如下, 有相切“1”圈, 存在0态冒险, 加入红圈覆盖。从而,

$$N_1(A, B, C) = \bar{A}\bar{C} + BC + AB$$



14

第四章 组合逻辑电路 结束

17