

# 数 字 逻 辑

# Digital Logic Circuit

丁 贤 庆

ahhfdxq@163.com

# Home work (P74)

---

☞ 2.4.3(5)(6) (7)

☞ 2.4.4(3)(4)

☞ 4.1.1

## Home work (P218)

☞ 4.1.3

☞ 4.2.2

☞ 4.2.3

本周四课快结束时，有10分钟的随堂测验。

---

### 2.4.3、具有无关项的化简

#### **“Don't-Care” Input Combinations** (“无关”输入组合)

(1) 什么叫无关项:

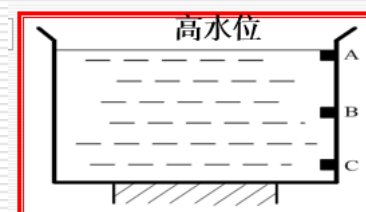
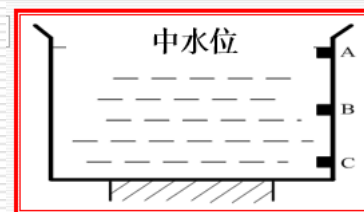
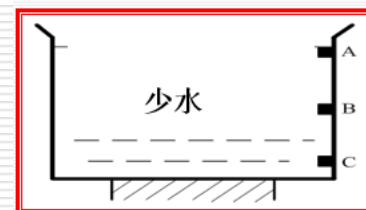
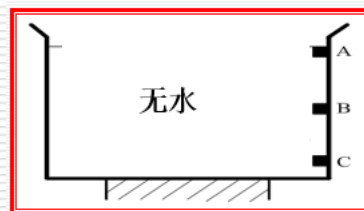
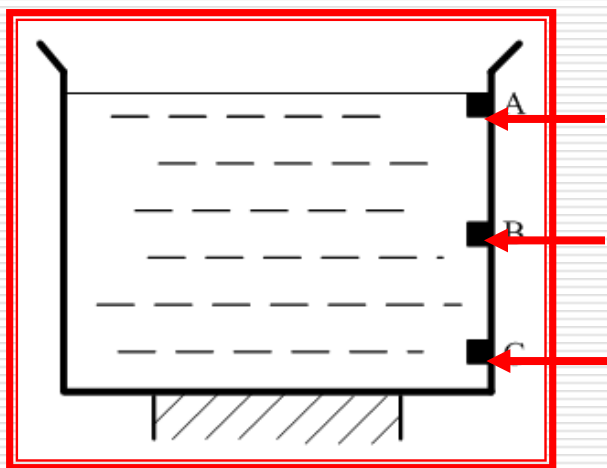
在真值表内对应于变量的某些取值下，函数的值可以是任意的，或者这些变量的取值根本不会出现，这些变量取值所对应的最小项称为无关项或任意项。

在含有无关项逻辑函数的卡诺图化简中，它的值可以取0或取1，具体取什么值，可以根据使函数尽量得到简化而定。

---

例：下图所示的水箱中设置了3个水位检测元件A、B、C，当水位高于检测元件时，检测元件输出为0，当水位低于检测元件时，检测元件输出为1。根据常识可知，检测元件A、B、C共有000、100、110和111四种取值组合，其余4种取值001、010、

011、101没有实际意义，因此不能取。在这种情况下，称变量A、B、C为一组具有约束的变量，不能取的这4种取值组合所对应的最小项称为该逻辑问题的约束项。





对于要设计的电路，约束项的输入值并不会出现，所以将约束项写入函数表达式或者不写入，对逻辑函数并没有影响。也就是说，在卡诺图中约束项对应的格子中填入1或者0都可以，一般填入“×”。表示既可以取1也可以取0。

例如：设计此电路时假设水位介于A和C之间是安全的，红灯不亮。水位介于A上或者C下都是危险的，红灯亮。红灯用Y表示，对应的真值表如图所示。

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>
0	0	0	1
0	0	1	x
0	1	0	x
0	1	1	x
1	0	0	0
1	0	1	x
1	1	0	0
1	1	1	1

例: 要求设计一个逻辑电路, 能够判断一位十进制数是奇数还是偶数, 当十进制数为奇数时, 电路输出为1, 当十进制数为偶数时, 电路输出为0。

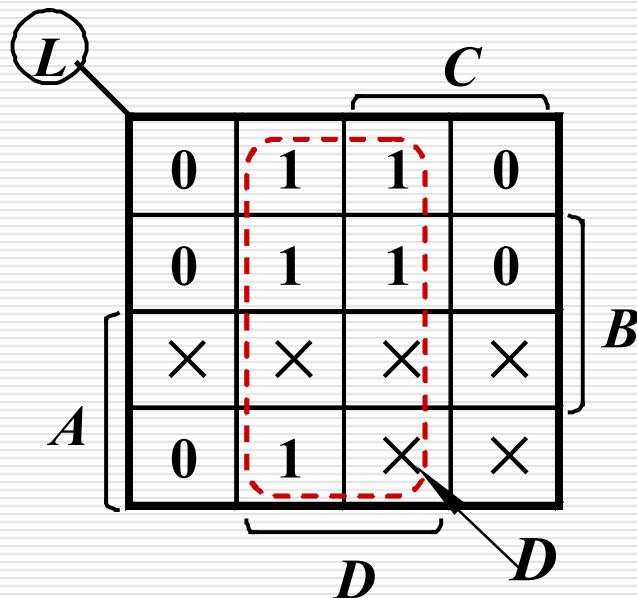
解:

(1) 列出真值表

(2) 画出卡诺图

(3) 卡诺图化简

$$L = D$$



ABCD	L
0000	0
0001	1
0010	0
0011	1
0100	0
0101	1
0110	0
0111	1
1000	0
1001	1
1010	×
1011	×
1100	×
1101	×
1110	×
1111	×

## ● 具有无关项逻辑函数的化简

### 无关项

$$F(A,B,C,D) = \underbrace{\sum (1,2,7,8,11)}_{F=1\text{项}} + \underbrace{\sum_{\phi} (0,6,9,15)}_{\text{无关项}}$$

例 化简函数  $F(A,B,C,D) = \sum (3,5,7)$ ，且无关项为  $\sum \phi(10,11,12,13,14,15)$

解：

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

$$F(A,B,C,D) = CD + BD$$

例

化简函数  $F(A,B,C,D) = \sum(0,7,13,14,15)$ ，且无关项为

$$\sum \phi(1,2,3,9,10,11)$$

解：

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

$$F(A,B,C,D) = \overline{A}\overline{B} + CD + AD + AC$$



---

## 2.5 硬件描述语言Verilog HDL基础(推后讲)

**2.5.1 Verilog语言的基本语法规则**

**2.5.2 变量的数据类型**

**2.5.3 运算符及其优先级**

**2.5.4 Verilog内部的基本门级元件**

**2.5.5 Verilog程序的基本结构**

**2.5.6 逻辑功能的仿真与测试**

---

# 3 逻辑门电路(推后讲)

---

3.1 逻辑门电路简介

3.2 基本CMOS逻辑门电路

3.3 CMOS逻辑门电路的不同输出结构及参数

3.4 类NMOS和BiCMOS逻辑门电路

3.5 TTL逻辑门电路

**\*3.6 ECL逻辑门电路**

3.7 逻辑描述中的几个问题

3.8 逻辑门电路使用中的几个实际问题

3.9 用VerilogHDL描述CMOS逻辑门电路

# 第4章 组合逻辑电路

## Combinational Logic Circuit

(为了安排实验，本周先讲第四章内容)

# 4 组合逻辑电路

---

4.1 组合逻辑电路的分析

4.2 组合逻辑电路的设计

4.3 组合逻辑电路中的竞争和冒险

4.4 若干典型的组合逻辑电路

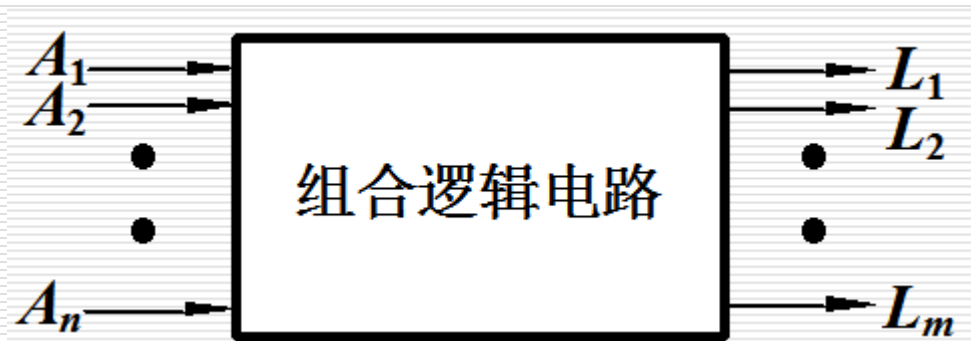
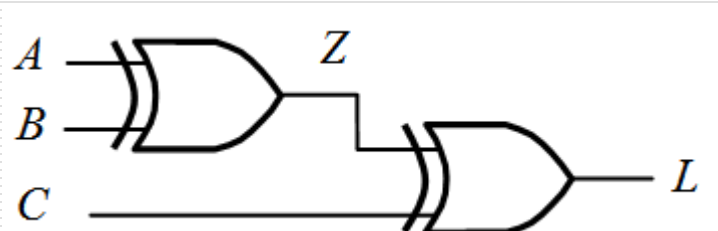
4.5 组合可编程逻辑器件

4.6 用Verilog HDL描述组合逻辑电路

# 4.1 组合逻辑电路分析

## 4.1.1 组合逻辑电路的定义

组合逻辑电路的一般框图



$$L_i = f(A_1, A_2, \dots, A_n) \quad (i=1, 2, \dots, m)$$

结构特征:

- 1、输出、输入之间没有反馈延迟通路,
- 2、不含记忆单元

工作特征:

**组合逻辑电路工作特点:**在任何时刻, 电路的**输出状态**只取决于同一时刻的**输入状态**而与电路**原来的状态**无关。

## 4.1.2 组合逻辑电路的分析方法

---

### 一. 组合逻辑电路分析

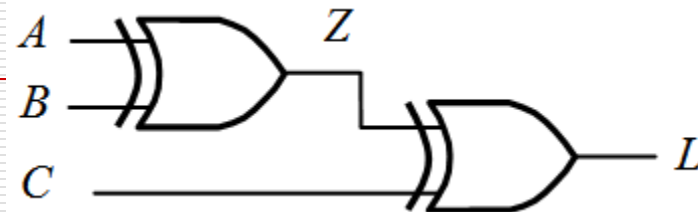
根据已知逻辑电路，经分析确定电路的逻辑功能。

### 二. 组合逻辑电路的分析步骤：

- 1、由逻辑图写出各输出端的逻辑表达式；
  - 2、化简和变换逻辑表达式；
  - 3、列出真值表；
  - 4、根据真值表或逻辑表达式，经分析最后确定其功能。
-

### 三、组合逻辑电路的分析举例

例1 分析如图所示逻辑电路的功能。



解：1.根据逻辑图写出输出函数的**逻辑表达式**

$$L = Z \oplus C$$

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

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

2. 列写**真值表**。

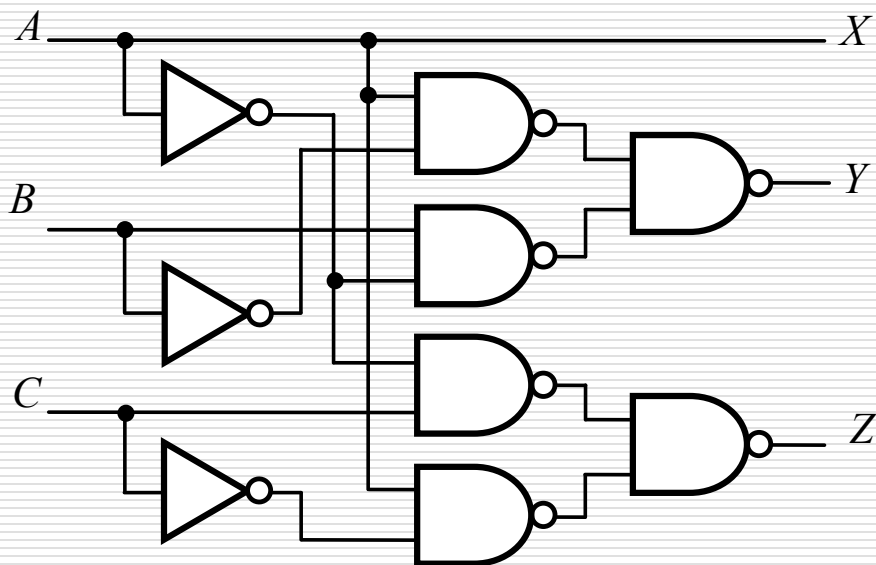
3. 确定逻辑功能：

输入变量的取值中**有奇数个1时，L为1**，否则L为0，  
电路具有为**奇校验功能**。

A	B	C	$Z = A \oplus B$	$L = (A \oplus B \oplus C)$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

## 例2 试分析下图所示组合逻辑电路的逻辑功能。

解：1、根据逻辑电路写出各输出端的逻辑表达式，并进行化简和变换。



$$X = A$$

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

$$Z = \overline{\overline{A} \overline{C}} \cdot \overline{\overline{A} \overline{C}}$$



## 2、列写真值表

$$X = A$$

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

$$Z = \overline{\overline{A} \overline{C}} \cdot \overline{\overline{A} \overline{C}} = A \overline{C} + \overline{A} C$$

真值表

<i>A</i>	<i>B</i>	<i>C</i>	<i>X</i>	<i>Y</i>	<i>Z</i>
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

### 3、确定电路逻辑功能

这个电路逻辑功能是对输入的二进制码求反码。最高位为符号位，0表示正数，1表示负数，正数的反码与原码相同；负数反码的数值部分是在原码的基础上逐位求反。

真值表

A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	0	0

## 4.2 组合逻辑电路的设计

---

### 4.2.1 组合逻辑电路的设计过程

一、组合逻辑电路的设计：根据实际逻辑问题，求出所要求逻辑功能的最简单逻辑电路。

### 二、组合逻辑电路的设计步骤

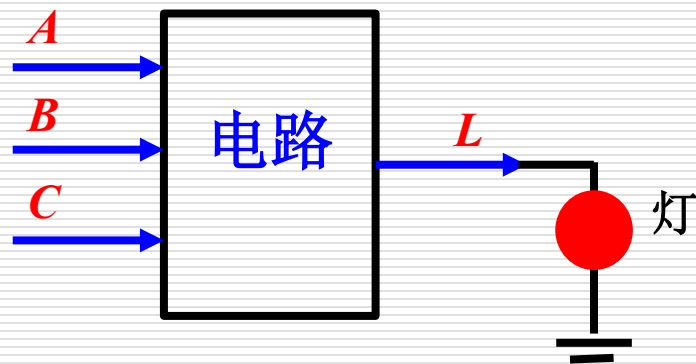
- 1、逻辑抽象：根据实际逻辑问题的因果关系确定输入、输出变量，并定义逻辑状态的含义；
  - 2、根据逻辑描述列出真值表；
  - 3、由真值表写出逻辑表达式；
  - 4、简化和变换逻辑表达式，画出逻辑图。
-

例1 某火车站有特快、直快和慢车三种类型的客运列车进出，试设计一个指示列车等待进站的逻辑电路，**当有两种或以上的列车等待进站时，要求发出信号**，提示工作人员安排进站事宜。

解：（1）逻辑抽象。

输入信号： **$A$ 、 $B$ 、 $C$** 分别表示**特快、直快和慢车**，且有进站请求时为1，没有请求时为0。

输出信号： **$L$** 表示**进站状况**，有两种以上的车进站为1，否则为0。



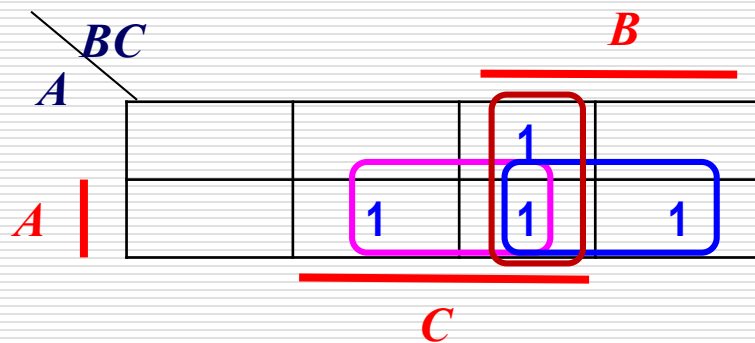
## (2) 根据题意列出真值表

输 入			输 出
<i>A</i>	<i>B</i>	<i>C</i>	<i>L</i>
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

(3) 写出输出逻辑表达式,并化简。

$$L = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$L = AB + AC + BC$$

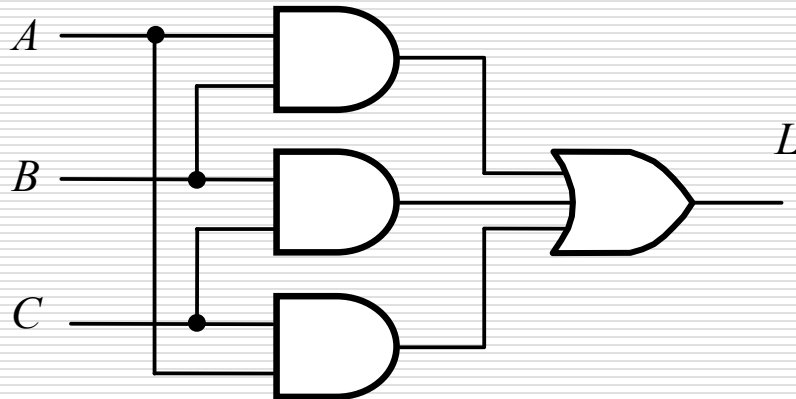


(4) 根据输出逻辑表达式画出逻辑图。

---

$$L = AB + AC + BC$$

表达式为最简与或式，用与门和或门实现两级“与-或”结构的最简电路如图。



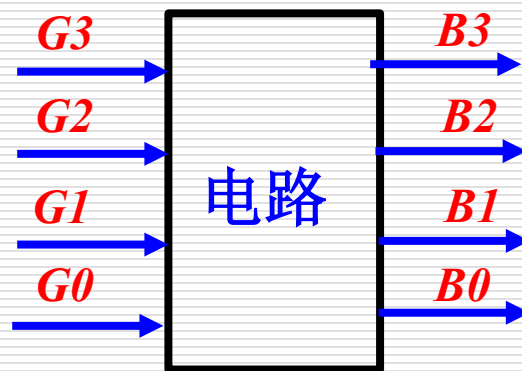
例2 试设计一个码转换电路，将4位格雷码转换为4位自然二进制码。可以采用任何逻辑门电路来实现。

解：(1) 明确逻辑功能，列出真值表。

设输入变量为 $G_3$ 、 $G_2$ 、 $G_1$ 、 $G_0$ 为格雷码，  
输出变量 $B_3$ 、 $B_2$ 、 $B_1$ 和 $B_0$ 为自然二进制码。

当输入格雷码按照从0到15递增排序时，

可列出逻辑电路真值表



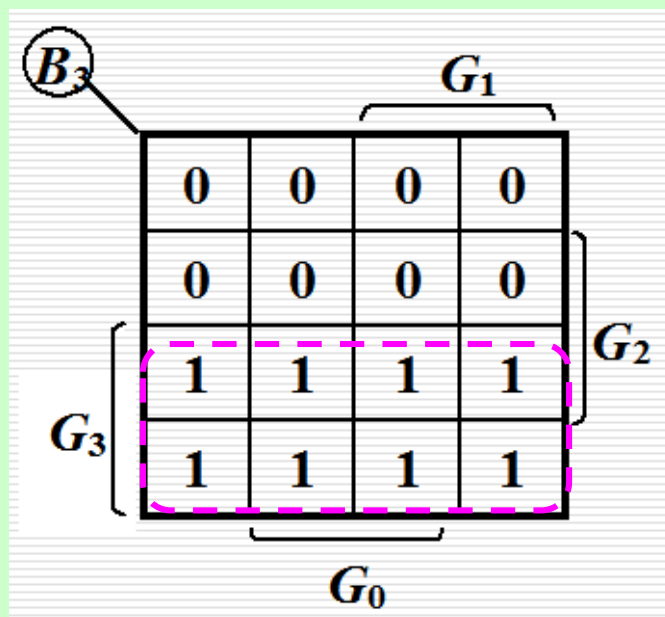
# 逻辑电路真值表

输 入				输 出				输 入				输 出			
$G_3$	$G_2$	$G_1$	$G_0$	$B_3$	$B_2$	$B_1$	$B_0$	$G_3$	$G_2$	$G_1$	$G_0$	$B_3$	$B_2$	$B_1$	$B_0$
0	0	0	0					1	1	0	0				
0	0	0	1					1	1	0	1				
0	0	1	1					1	1	1	1				
0	0	1	0					1	1	1	0				
0	1	1	0					1	0	1	0				
0	1	1	1					1	0	1	1				
0	1	0	1					1	0	0	1				
0	1	0	0					1	0	0	0				



输 入				输 出
$G_3$	$G_2$	$G_1$	$G_0$	$B_3$
0	0	0	0	0
0	0	0	1	0
0	0	1	1	0
0	0	1	0	0
0	1	1	0	0
0	1	1	1	0
0	1	0	1	0
0	1	0	0	0
1	1	0	0	1
1	1	0	1	1
1	1	1	1	1
1	1	1	0	1
1	0	1	0	1
1	0	1	1	1
1	0	0	1	1
1	0	0	0	1

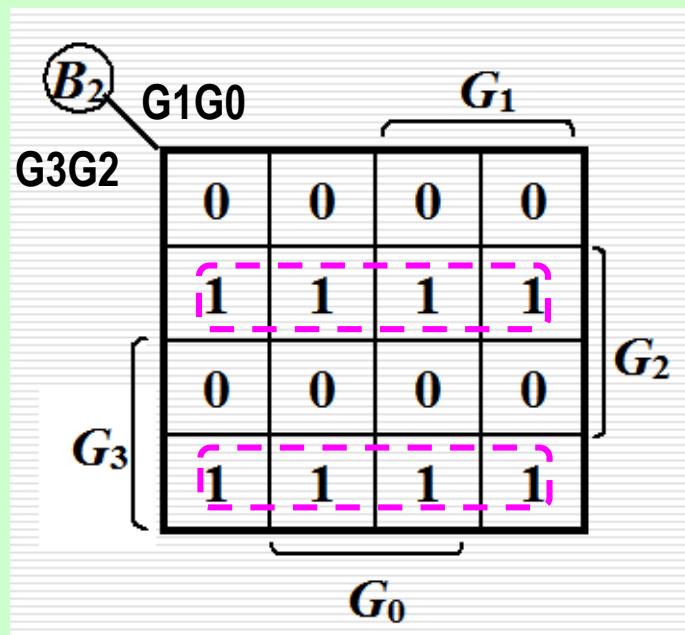
(2) 画出各输出函数的卡诺图，并化简和变换。



$$B_3 = G_3$$

输 入					输 出
$G_3$	$G_2$	$G_1$	$G_0$	$B_2$	
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	1	0	0
0	0	1	0	0	0
0	1	1	0	1	1
0	1	1	1	1	1
0	1	0	1	1	1
0	1	0	0	1	1
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	1	0	0
1	1	1	0	0	0
1	0	1	0	1	1
1	0	1	1	1	1
1	0	0	1	1	1
1	0	0	0	1	1

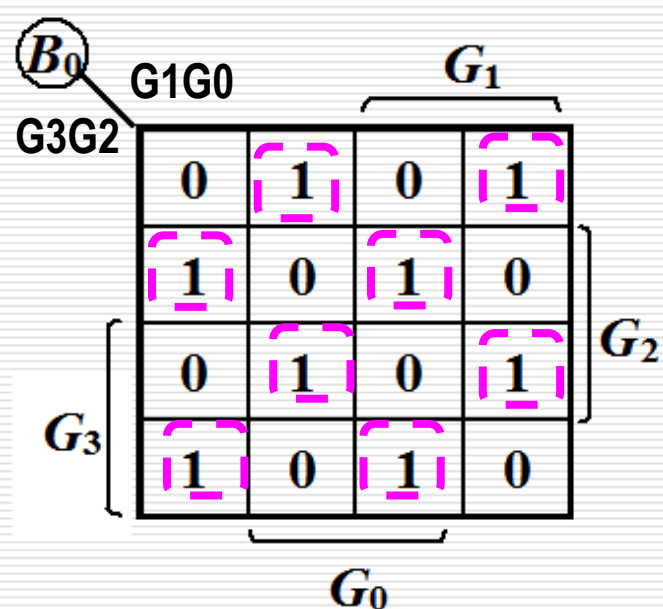
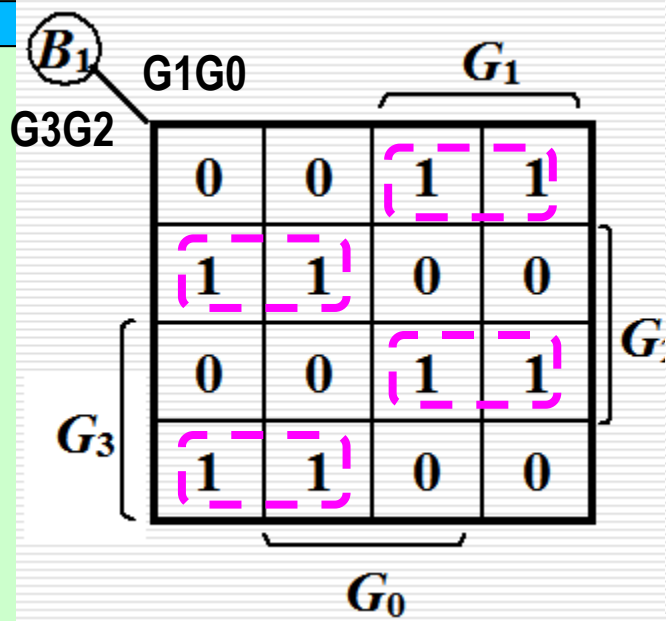
(2) 画出各输出函数的卡诺图，并化简和变换。



$$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2$$

$$= G_3 \oplus G_2$$

当卡诺图中1的分布位于45度角方向，写表达式时，可以向异或门靠拢。



$$B_1 = G_3 \bar{G}_2 \bar{G}_1 + \bar{G}_3 G_2 \bar{G}_1 + G_3 G_2 G_1 + \bar{G}_3 \bar{G}_2 G_1$$

$$= (G_3 \bar{G}_2 + \bar{G}_3 G_2) \bar{G}_1 + \overline{G_3 \bar{G}_2 + \bar{G}_3 G_2} G_1$$

$$= G_3 \oplus G_2 \oplus G_1$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

### (3) 根据逻辑表达式，画出逻辑图

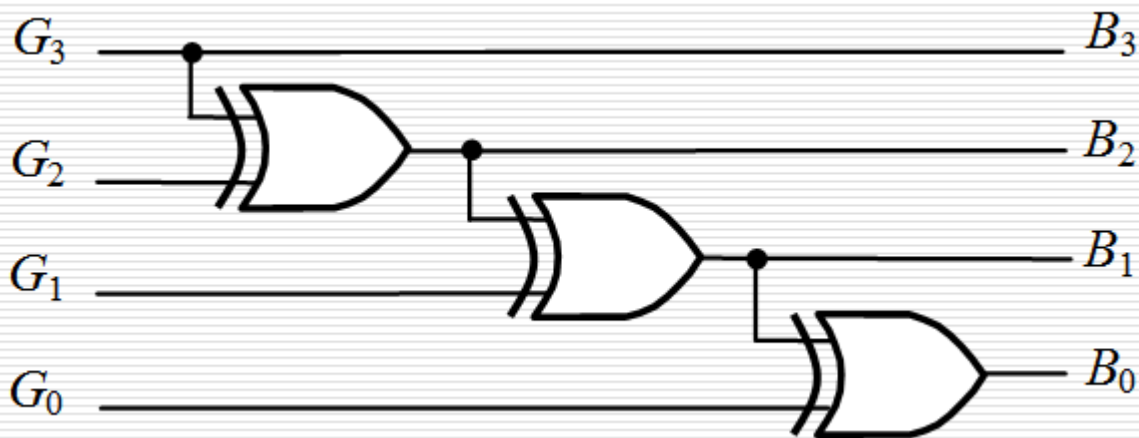
$$B_3 = G_3$$

$$B_2 = \bar{G}_3 G_2 + \bar{G}_3 G_2$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

用异或门代替与门和或门能使逻辑电路比较简单。考虑相同乘积项可以减少门电路数目，降低实现电路的成本。

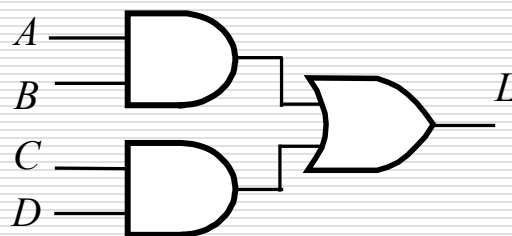


## 4.2.2 组合逻辑电路的优化实现

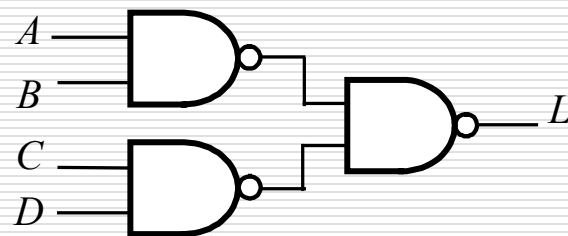
用指定芯片中特定资源实现逻辑函数，使电路的成本低并且工作速度快。因此需要对逻辑表达式进行变换，以减少芯片资源的数目和连线。

### 1、单输出电路

$$\begin{aligned} L &= AB + CD \\ &= \overline{\overline{AB} \cdot \overline{CD}} \end{aligned}$$



(a) 与-或结构



(b) 与非门结构

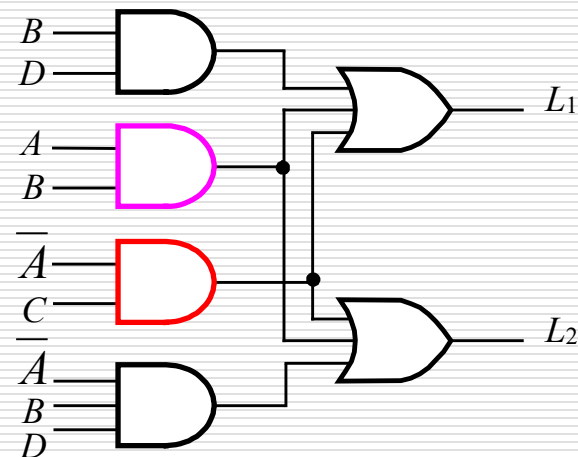
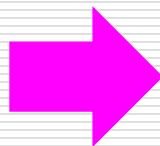
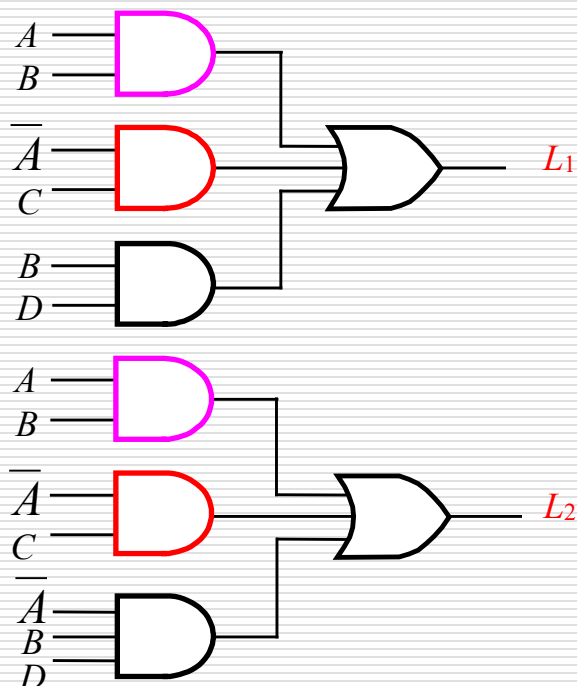
相同输入端的与非门比与门或者或门所用晶体管少，速度快。图(b)电路最优

## 2、多输出电路

输出多个逻辑函数时需要考虑共享相同乘积项，减少逻辑门数目。

$$L_1 = AB + \bar{A}C + BD$$

$$L_2 = AB + \bar{A}C + \bar{A}BD$$



- (a) 如果分别实现两个逻辑函数，需要6个与门和2个或门。  
(b) 如果考虑相同乘积项，需要4个与门和2个或门，如图。

### 3、多级逻辑电路

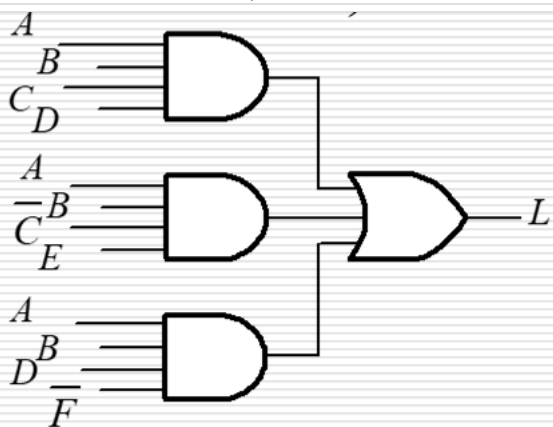
当限定逻辑门输入端数目，则需要进行逻辑变换。

#### (1) 提取公因子

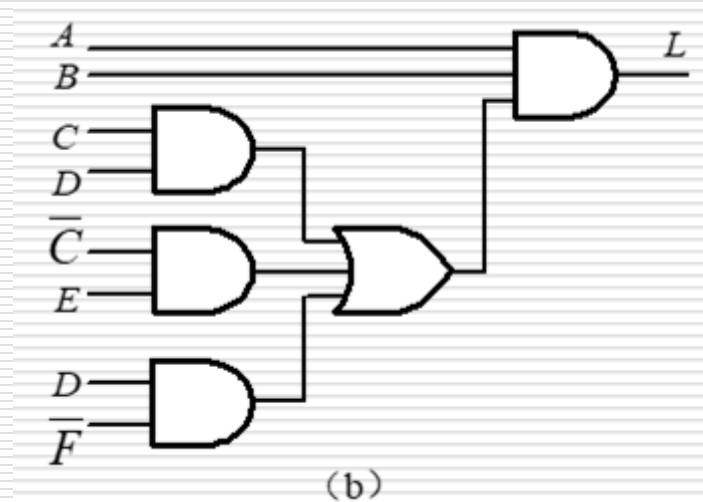
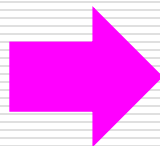
$$L = ABCD + AB\bar{C}E + ABD\bar{F} \quad (\text{每项中有4个输入变量})$$

用与门、或门实现时，限定逻辑门的扇入数为3，需要变换成：

$$L = AB(CD + \bar{C}E + D\bar{F})$$



(a)



(b)

图(a)电路为2级，图(b)为3级，但电路连线减少了。图(a)16根连线，图(b)13根。

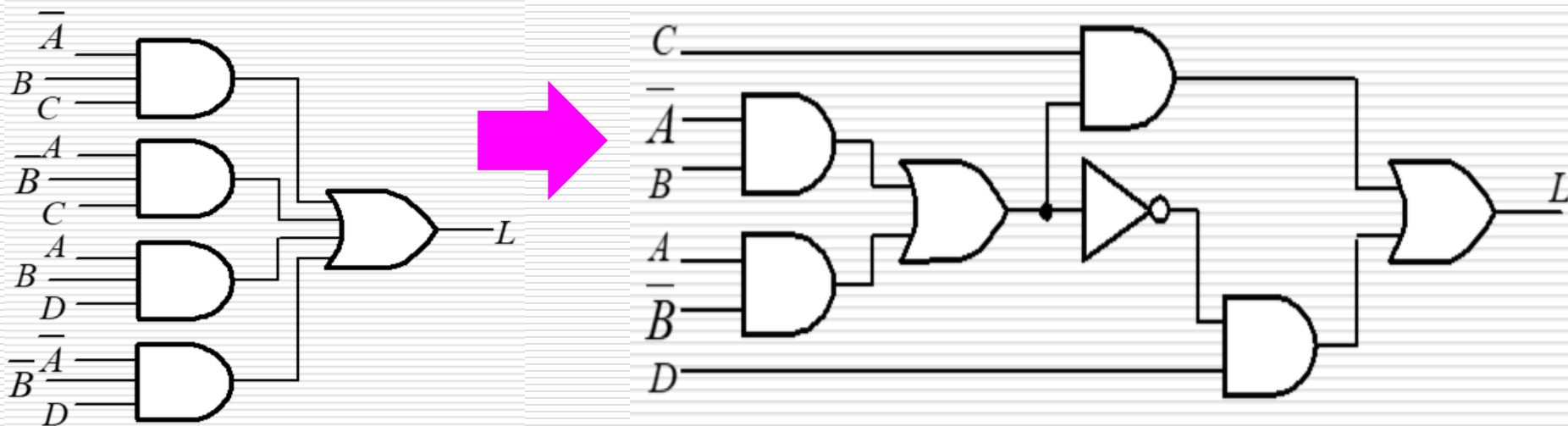
## (2) 函数分解

$$L = \overline{A}BC + A\overline{B}C + ABD + \overline{A}\overline{B}D$$

其中有4个与项要进行或运算。

用与门、或门实现时，限定逻辑门的扇入数为3，需要变换成：

$$L = (\overline{A}B + A\overline{B})C + (AB + \overline{A}\overline{B})D = (\overline{A}B + A\overline{B})C + \overline{(\overline{A}B + A\overline{B})}D$$



图(a)电路为2级，图(b)为5级。

上述变换方法只适合手工化简，当变量数很多时，优化策略写入程序由计算机完成。



## 4.3 组合逻辑电路中的竞争冒险

---

### 4.3.1 产生的竞争冒险的原因

### 4.3.2 消去竞争冒险的方法

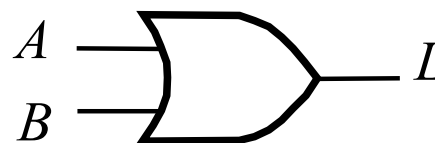
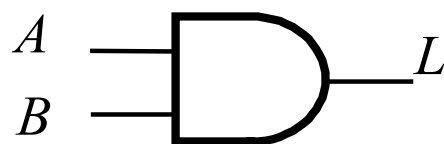
## 4.3 组合逻辑电路中的竞争冒险

### 4.3.1 产生的竞争冒险的原因

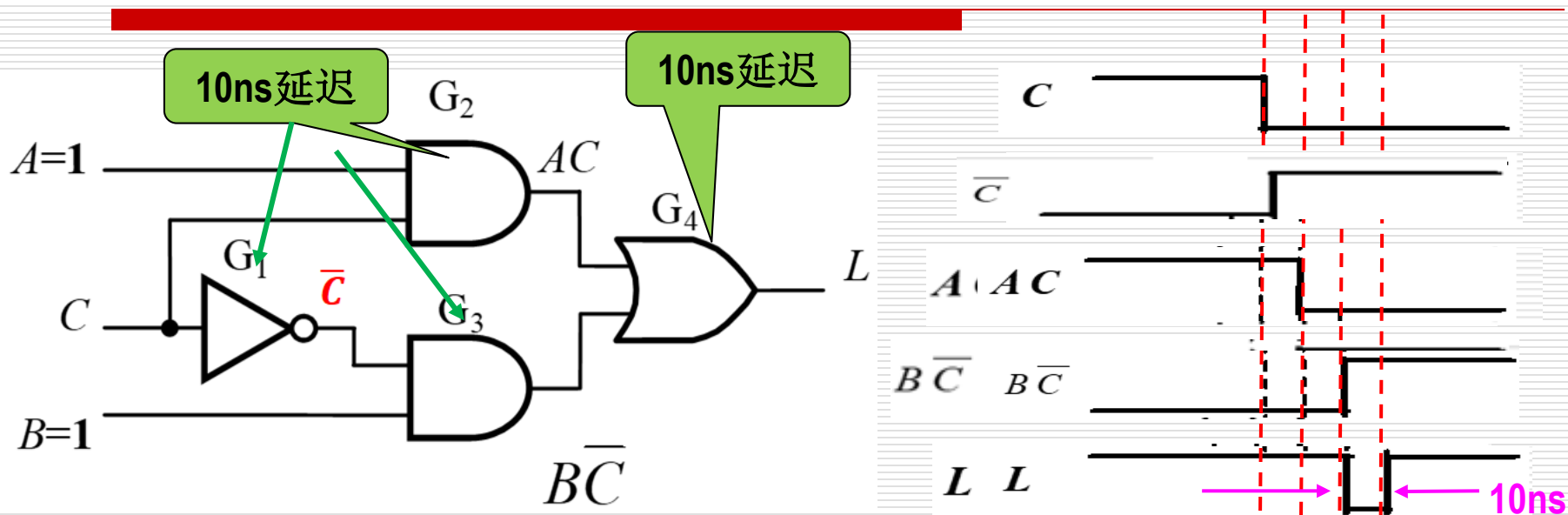
不考虑门的延时时间，且 $B=\bar{A}$

$$L = AB = 0$$

$$L = A + B = 1$$



考虑门的延时时间，且用非门实现 $B=\bar{A}$ 时



若不考虑10ns的门延时： $L = AC + B\bar{C} = 1 \cdot C + 1 \cdot \bar{C} = 1$

**竞争:** 当一个逻辑门的两个输入端的信号同时向相反方向变化，而变化的时间有差异的现象。

**冒险:** 两个输入端的信号取值的变化方向是相反时，如门电路输出端的逻辑表达式简化成两个互补信号相乘或者相加，由竞争而可能产生输出干扰脉冲的现象。

## 竞争-冒险的检查方法

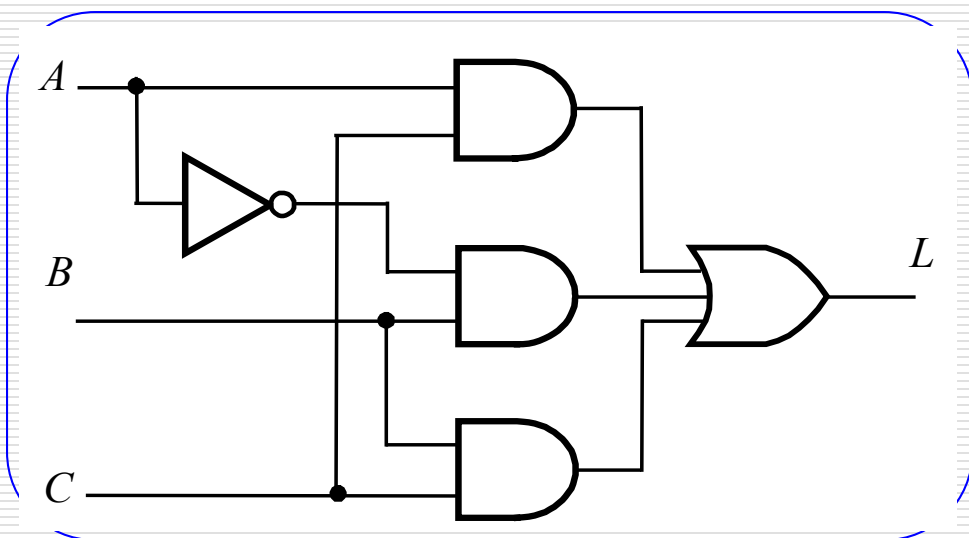
在输入变量每次只有一个改变状态的简单情况下，如果函数表达式中同时存在有 $A$ 和 $\bar{A}$ ，那么我们称 $A$ 为具有竞争能力的变量。对于具有竞争能力的变量，若将其余变量任意取值，函数表达式能够转化成 $Y=A\bar{A}$ 或者 $Y=A+\bar{A}$ 形式之一的，会发生竞争-冒险。

由于竞争-冒险产生的尖峰脉冲持续的时间很短，包含的能量很小，所以大多数竞争-冒险并不会对电路造成危害。但是，如果负载是对尖峰脉冲敏感的存储电路时，竞争-冒险就有可能使存储电路发生误动作而产生错误，因此设计数字系统时应尽量避免竞争-冒险现象的发生。

## 4.3.2 消去竞争冒险的方法

### 1. 发现并消除互补变量

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



$B = C = 0$ 时

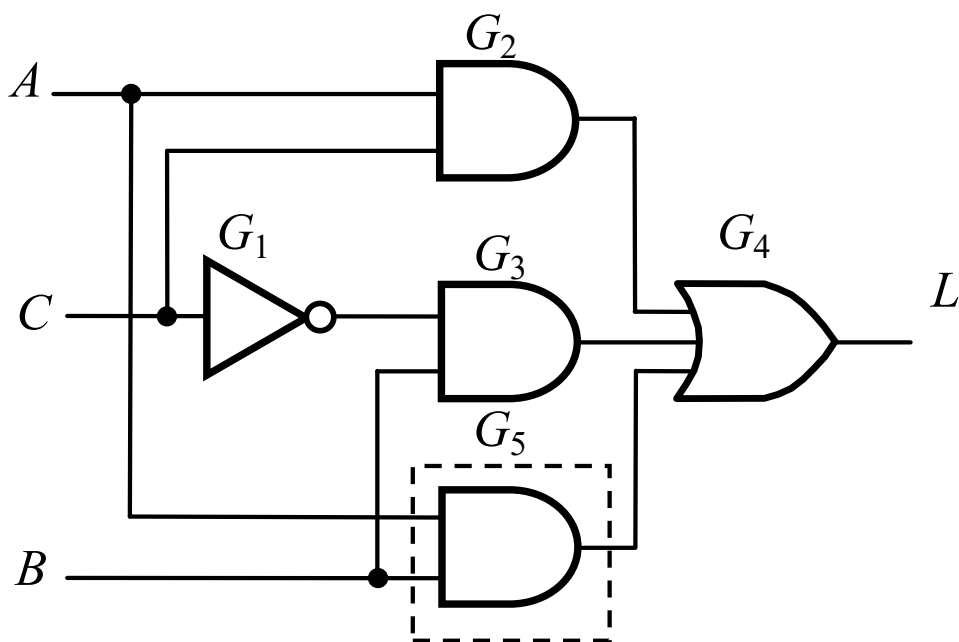
$$L = A\overline{A}$$

可能出现竞争冒险。

为消掉 $A\overline{A}$ ，变换逻辑函数式为

$$L = AC + \overline{A}B + BC$$

## 2. 增加乘积项, 避免互补项相加



$$L = AC + B\bar{C}$$

当  $A=B=1$  时

$$L = C + \bar{C}$$

$$L = AC + B\bar{C}$$

$$L = AC + B\bar{C} + AB$$

		$C$			
		00	01	11	10
$A$	0	0	0	0	1
	1	0	1	1	1

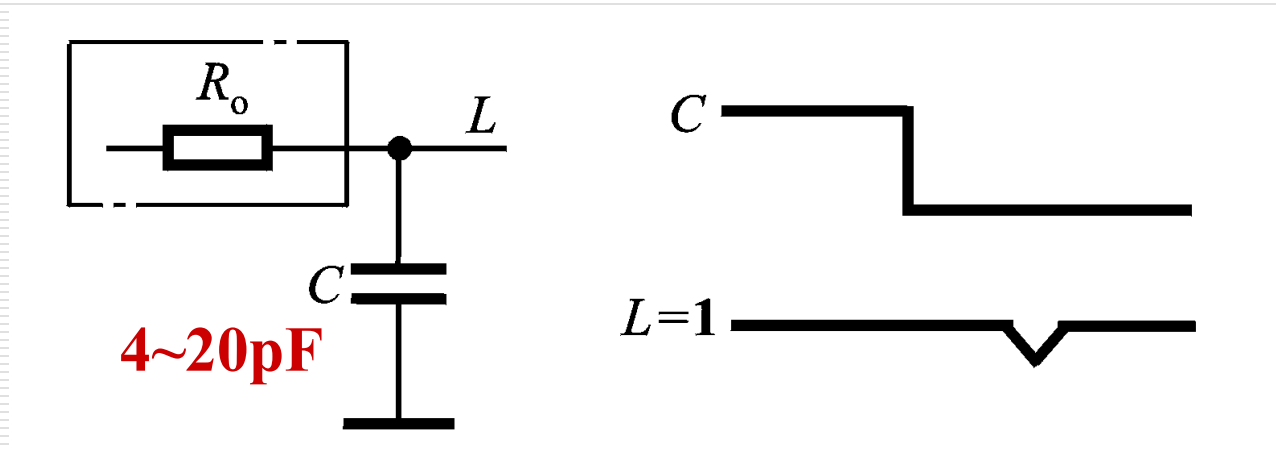
当  $A=B=1$  时, 根据逻辑表达式有

$$L = C + \bar{C} + 1$$

$AB$

### 3. 输出端并联电容器

如果逻辑电路在较慢速度下工作，为了消去竞争冒险，可以在输出端并联一电容器，致使输出波形上升沿和下降沿变化比较缓慢，可对于很窄的负跳变脉冲起到平波的作用。



## 4.4 若干典型的组合逻辑电路

---

### 4.4.1 编码器

### 4.4.2 译码器/数据分配器

### 4.4.3 数据选择器

### 4.4.4 数值比较器

### 4.4.5 算术运算电路



## 4.4 若干典型的组合逻辑集成电路

---

### 4.4.1 编码器 Encoders

#### 1、编码器 (Encoder)的定义与分类

**编码：**赋予二进制代码特定含义的过程称为**编码**。

如：8421BCD码中，用1000表示数字8

如：ASCII码中，用1000001表示字母A等

**编码器：**具有**编码功能的逻辑电路**。

**编码器**能将每一个编码输入信号变换为不同的二进制的代码输出。

---

## 1、编码器 (Encoder)的分类

按照编码器的编码进制来分, 有:

**BCD编码器**: 将10个编码输入信号分别编成10个4位码输出。

**8线-3线编码器**: 将8个输入的信号分别编成 8个3位二进制数码输出。

按照优先级来分: **普通编码器**和**优先编码器**。

---

---

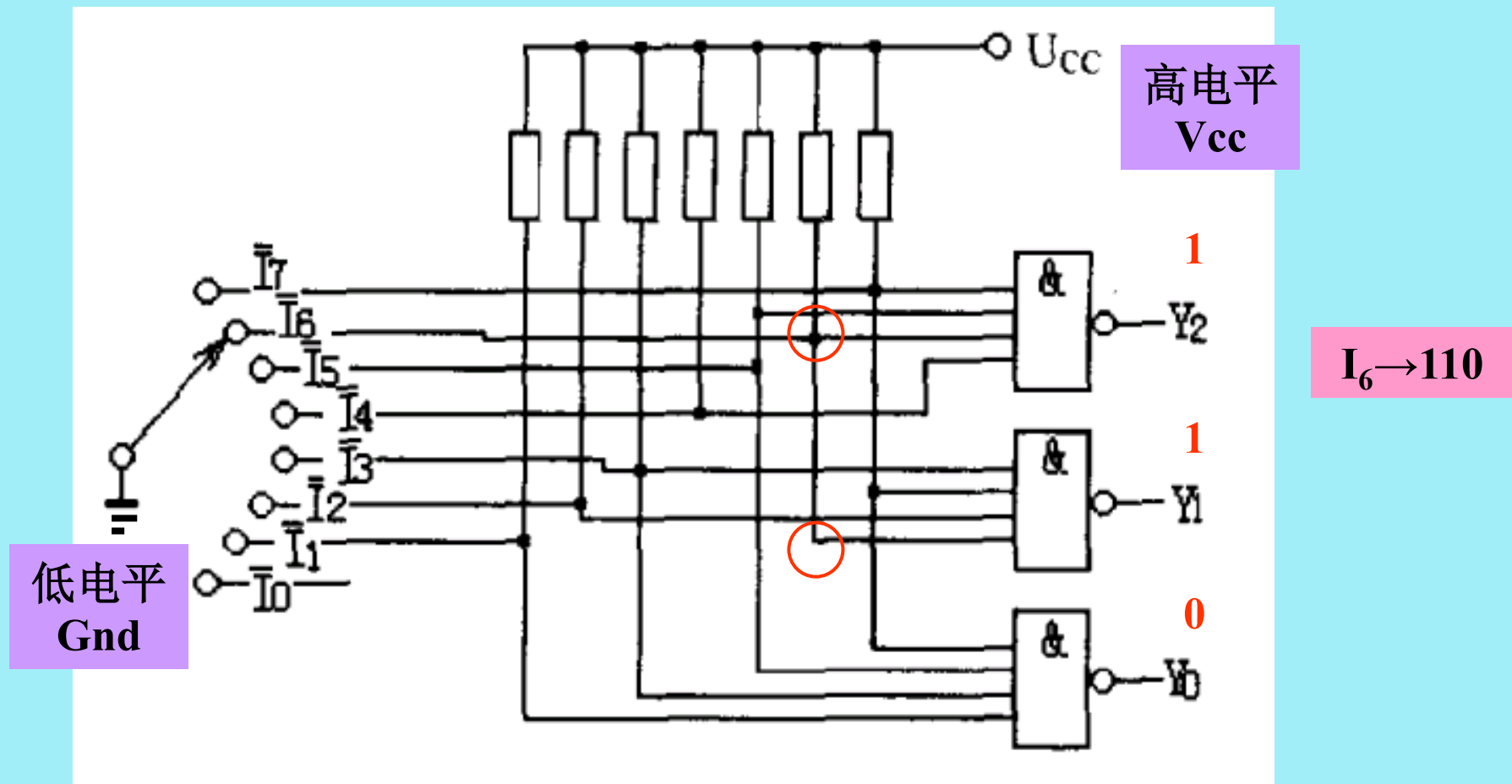
## 普通编码器和优先编码器的区别：

**普通编码器：**任何时候只允许输入一个有效编码信号，否则输出就会发生混乱。

**优先编码器：**允许同时输入两个以上的有效编码信号。当同时输入几个有效编码信号时，优先编码器能按预先设定的优先级别，只对其中优先权最高的一个进行编码。

---

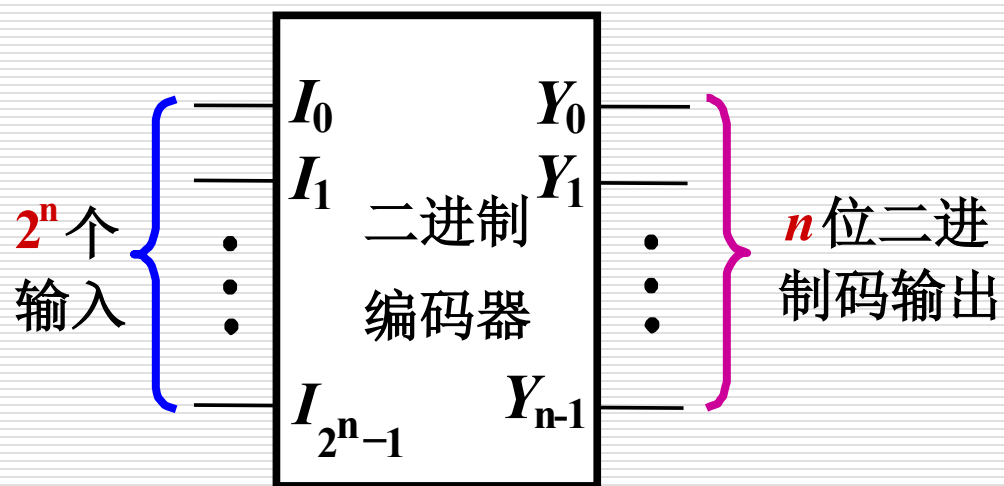
## 普通编码器电路



## 2、编码器的工作原理

### 普通二进制编码器

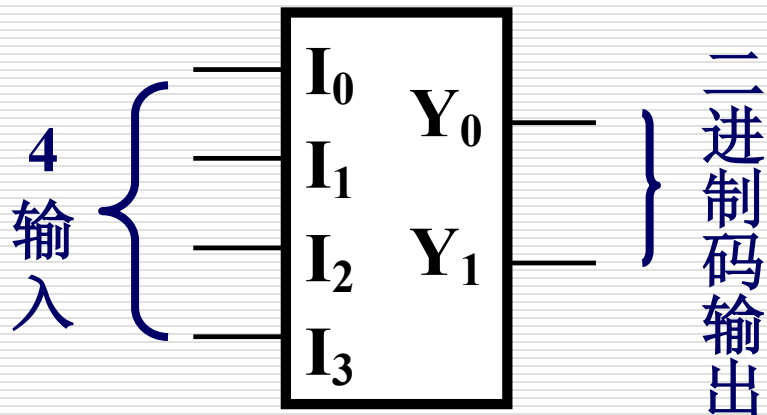
#### 二进制编码器的结构框图



## 2、编码器的工作原理

### (1) 4线—2线普通二进制编码器 (设计) (2) 逻辑功能表

(a) 逻辑框图



$$Y_1 = \bar{I}_0 \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

$$Y_0 = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

其它12种组合

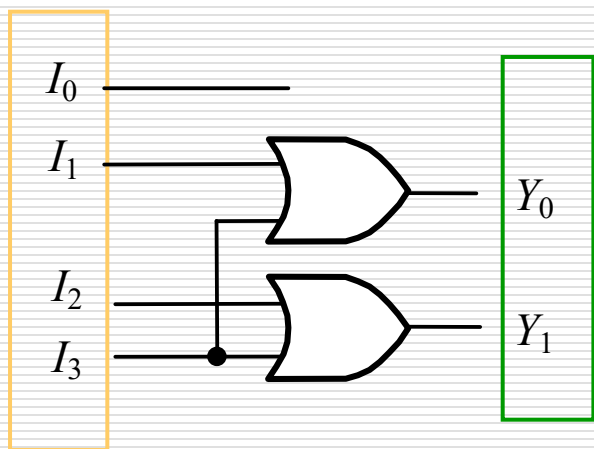
$I_0$	$I_1$	$I_2$	$I_3$	$Y_1$	$Y_0$
1	0	0	0		
0	1	0	0		
0	0	1	0		
0	0	0	1		
0	0	0	0	0	0
...	...			0	0
1	1	1	1	0	0

编码器的输入为高电平有效。

上述是将输入的其它12种组合对应的输出看做0。如果看做无关项，则表达式为

$$Y_1 = I_2 + I_3$$

$$Y_0 = I_1 + I_3$$

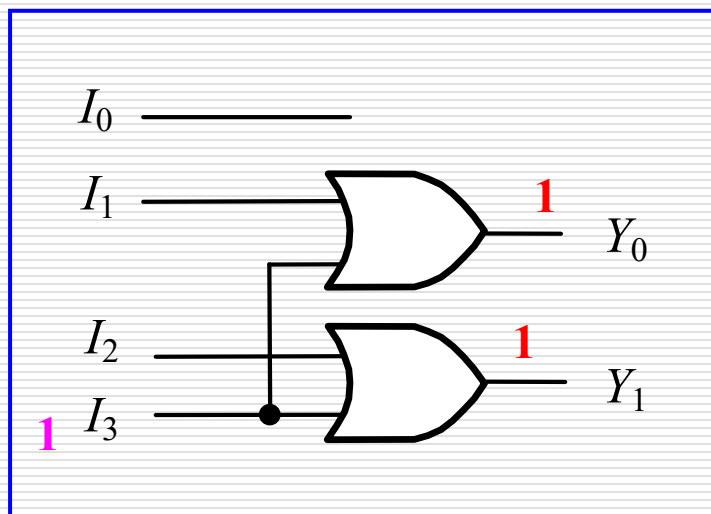


(2) 逻辑功能表

$I_0$	$I_1$	$I_2$	$I_3$	$Y_1$	$Y_0$
1	0	0	0		
0	1	0	0		
0	0	1	0		
0	0	0	1		
0	0	0	0		
...	...	...	...		
1	1	1	1		

其它12种组合

若有2个以上的输入为有效信号？



当只有 $I_3$ 为1时，

$$Y_1 Y_0 = ? \quad Y_1 Y_0 = 11$$

若 $I_1 = I_2 = 1$ ， $I_0 = I_3 = 0$ 时，

$$Y_1 Y_0 = ? \quad Y_1 Y_0 = 11$$

编码相同

无法输出有效编码。

结论：普通编码器不能同时输入两个以上的有效编码信号



### 3. 优先编码器

实际应用中，经常有两个或更多输入编码信号同时有效。



必须根据轻重缓急，规定好这些外设允许操作的先后次序，即优先级。

识别多个编码请求信号的优先级，并进行相应编码的逻辑部件称为优先编码器。

## (2) 优先编码器线(4—2 线优先编码器) (设计)

输入为编码信号  $I_3 \sim I_0$  输出为  $Y_1 Y_0$

输入编码信号高电平有效，输出为二进制代码

输入编码信号优先级从高到低为  $I_3 \sim I_0$

(1) 列出功能表

输 入				输 出	
$I_0$	$I_1$	$I_2$	$I_3$	$Y_1$	$Y_0$
1	0	0	0	0	0
×	1	0	0	0	1
×	×	1	0	1	0
×	×	×	1	1	1

低 ← 高

(2) 写出逻辑表达式

$$Y_1 = I_2 \bar{I}_3 + I_3$$

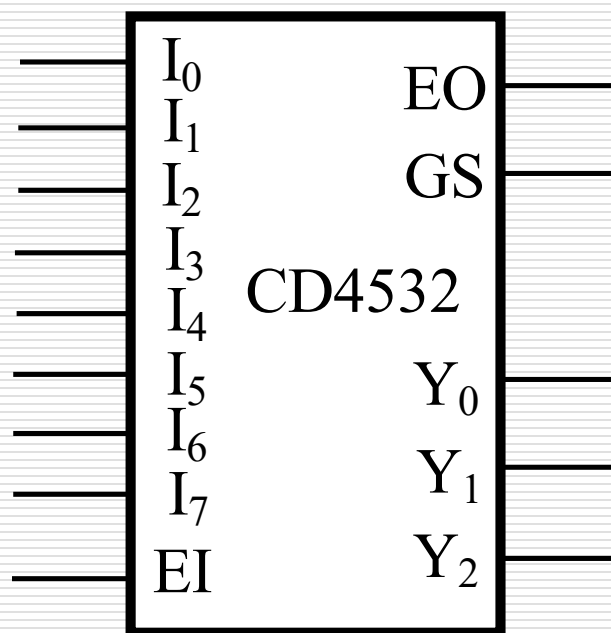
$$Y_0 = \bar{I}_1 \bar{I}_2 I_3 + I_3$$

(3) 画出逻辑电路 (略)

## 2 典型编码器电路

---

### 优先编码器CD4532的示意框图



8—3 线优先编码器,  
真值表如下页所示。

当EI为0时，不论I7~I0怎样变化，GS=0,EO=0

当EI为1时，如I7~I0中有一个1，则GS=1,EO=0

优先编码器CD4532功能表

输 入									输 出			
<i>EI</i>	<i>I</i> <sub>7</sub>	<i>I</i> <sub>6</sub>	<i>I</i> <sub>5</sub>	<i>I</i> <sub>4</sub>	<i>I</i> <sub>3</sub>	<i>I</i> <sub>2</sub>	<i>I</i> <sub>1</sub>	<i>I</i> <sub>0</sub>	<i>Y</i> <sub>2</sub>	<i>Y</i> <sub>1</sub>	<i>Y</i> <sub>0</sub>	<i>EO</i>
0	×	×	×	×	×	×	×	×	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1
1	0	1	×	×	×	×	×	×	1	1	0	1
1	0	0	1	×	×	×	×	×	1	0	1	1
1	0	0	0	1	×	×	×	×	1	0	0	1
1	0	0	0	0	1	×	×	×	0	1	1	1
1	0	0	0	0	0	1	×	×	0	1	0	1
1	0	0	0	0	0	0	1	×	0	0	1	1
1	0	0	0	0	0	0	0	1	0	0	0	1

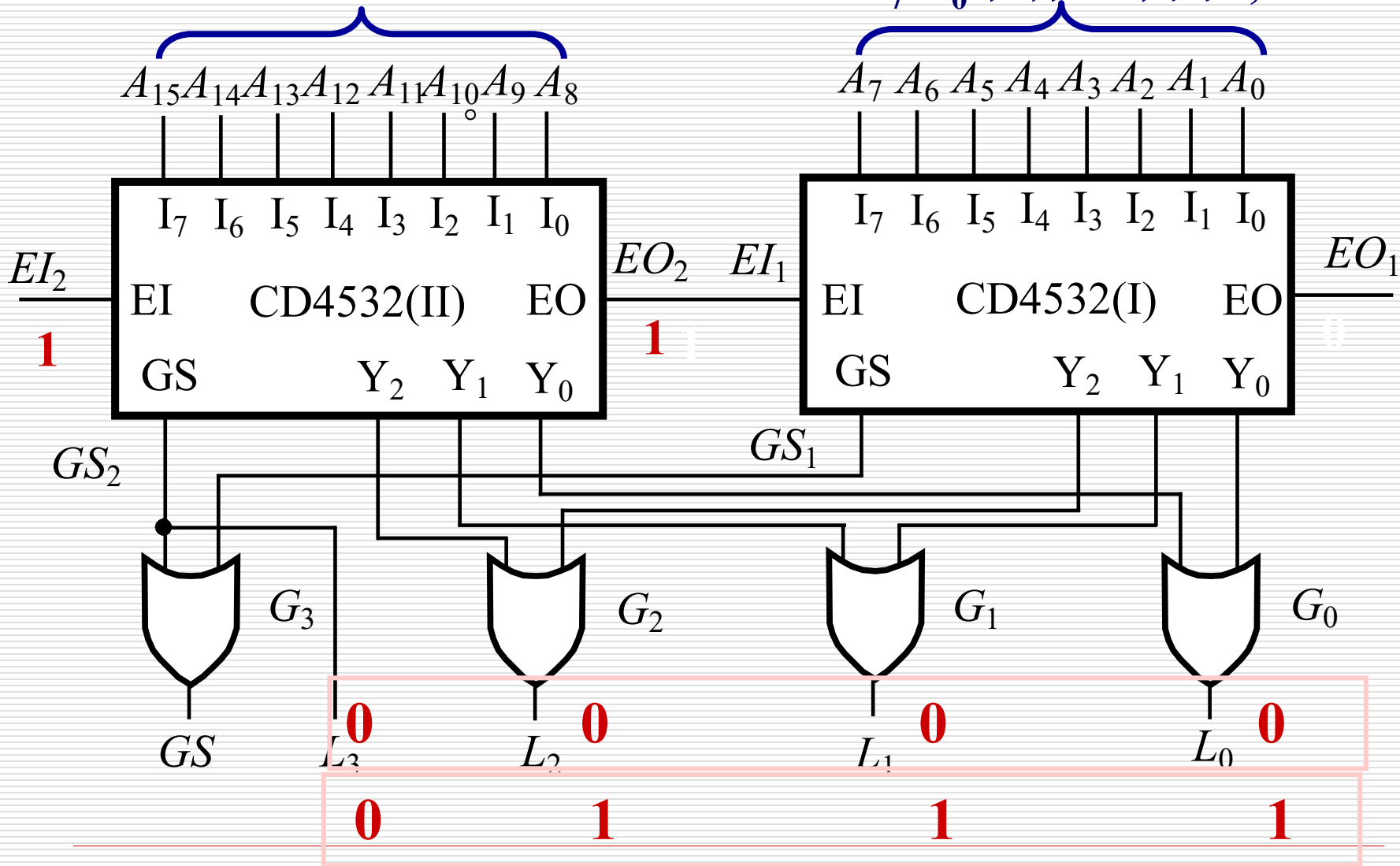
为什么要设计GS、EO输出信号？

GS、EO输出信号常用多个芯片的级联

(了解)

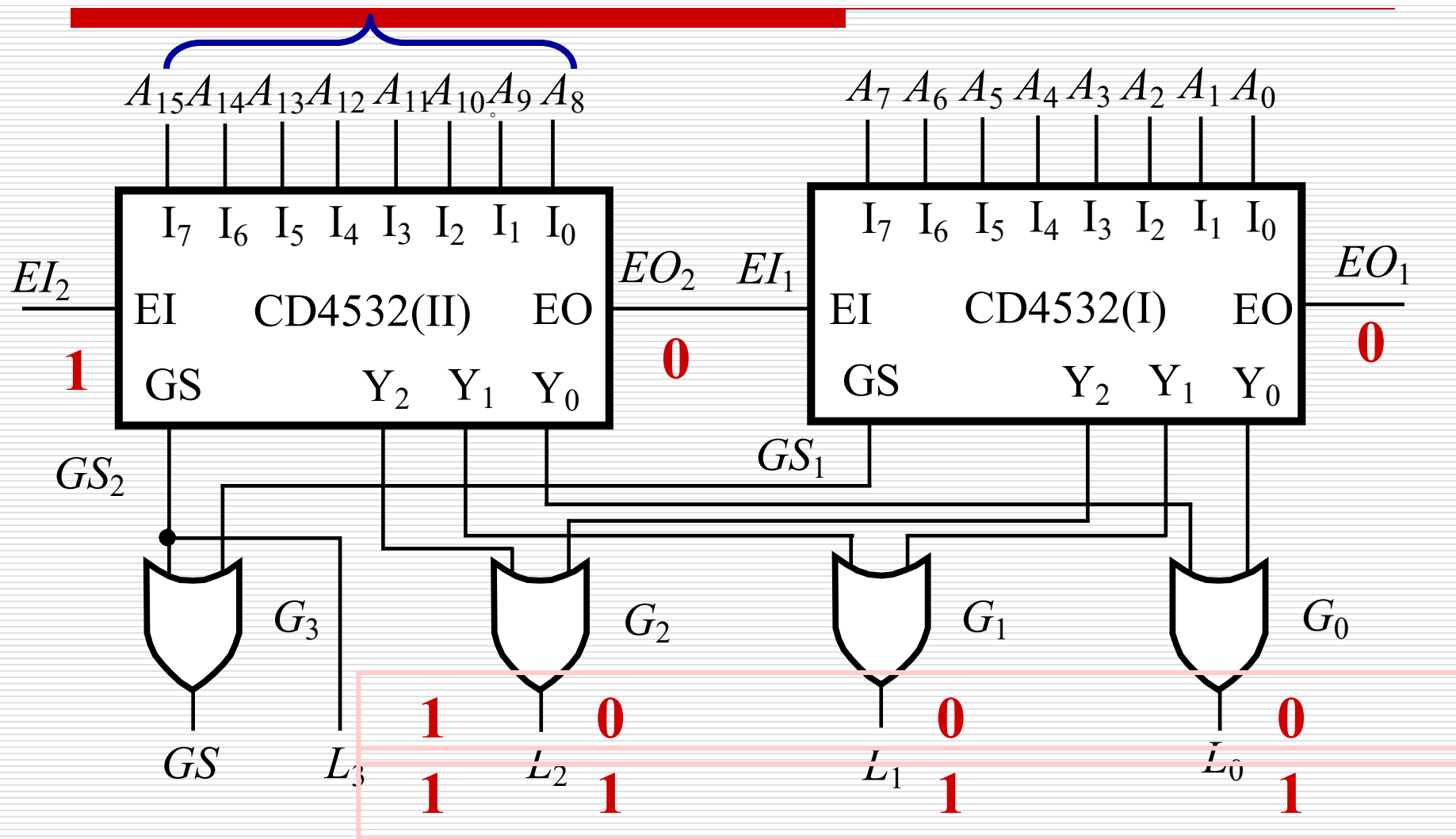
若无有效电平输入,即  
 $I_{15} \sim I_8 = 0 \sim 0$ ;

若有效电平输入,即  
 $I_7 \sim I_0$ 中有一个为1;



(了解) 若有效电平输入即 $I_{15} \sim I_8$ 中有一个为1;

哪块芯片的优先级高?



## 2. 8421BCD码编码器功能表

输 入										输 出					
$\overline{S_0}$	$\overline{S_1}$	$\overline{S_2}$	$\overline{S_3}$	$\overline{S_4}$	$\overline{S_5}$	$\overline{S_6}$	$\overline{S_7}$	$\overline{S_8}$	$\overline{S_9}$	A	B	C	D	GS	
1	1	1	1	1	1	1	1	1	1						
1	1	1	1	1	1	1	1	1	0						
1	1	1	1	1	1	1	1	0	1						
1	1	1	1	1	1	1	0	1	1						
1	1	1	1	1	1	0	1	1	1						
1	1	1	1	1	0	1	1	1	1						
1	1	1	1	0	1	1	1	1	1						
1	1	1	0	1	1	1	1	1	1						
1	1	0	1	1	1	1	1	1	1						
1	0	1	1	1	1	1	1	1	1						
0	1	1	1	1	1	1	1	1	1						

该编码器为输入低电平有效，输出高电平有效，**GS**为标志位。