

# 数字电路分析与设计

基本组合逻辑电路

(4.1.1 ~ 4.1.2、4.1.4)

## n 逻辑电路

ü 在数字系统中，常用的各种数字逻辑电路按其功能可分为：  
组合逻辑电路（combinational logic circuit）；  
时序逻辑电路（sequential logic circuit）。

ü 组合逻辑电路：

任意时刻的输出仅取决于该时刻的输入，与电路的初始状态无关。

（只要输入改变，输出随之改变）

（电路的输出与输入之间无反馈，电路不需要记忆元件）

ü 时序逻辑电路：

输出由输入和电路的初始状态共同决定。

（电路中一定包含具有记忆功能的触发器）

## n 组合逻辑电路

✓ 组合逻辑电路的分析与设计（4.1.1 ~ 4.1.2）

✓ 常见基本组合逻辑电路（4.1.4）：

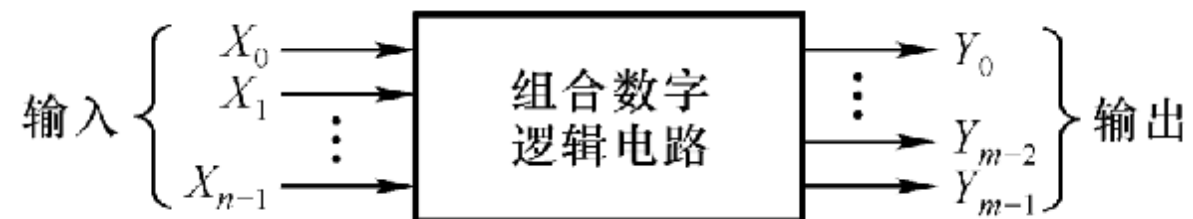
编码器、译码器；

数据选择器、数据分配器；

加法器、数值比较器。

## ✓ 组合逻辑电路

ü 下图所示组合逻辑电路的结构框图。



输入：  $X_0 \sim X_{n-1}$  ； 输出：  $Y_0 \sim Y_{m-1}$  ；

逻辑关系：  $Y_i = f_i(X_0, X_1, \mathbf{L}, X_{n-1}), \quad i = 0, 1, \mathbf{L}, m-1$

ü 组合逻辑电路的表达方式：

真值表、逻辑函数表达式、逻辑电路图、波形图和卡诺图。

## Ø 组合逻辑电路（分析）

ü 分析：已知逻辑电路图，说明电路的功能。

ü 分析步骤：

根据电路图，逐级写出各输出的逻辑函数式；

根据函数式，用真值表或计算法，得出所有输入情况下的所有结果；

根据结果，说明电路功能。

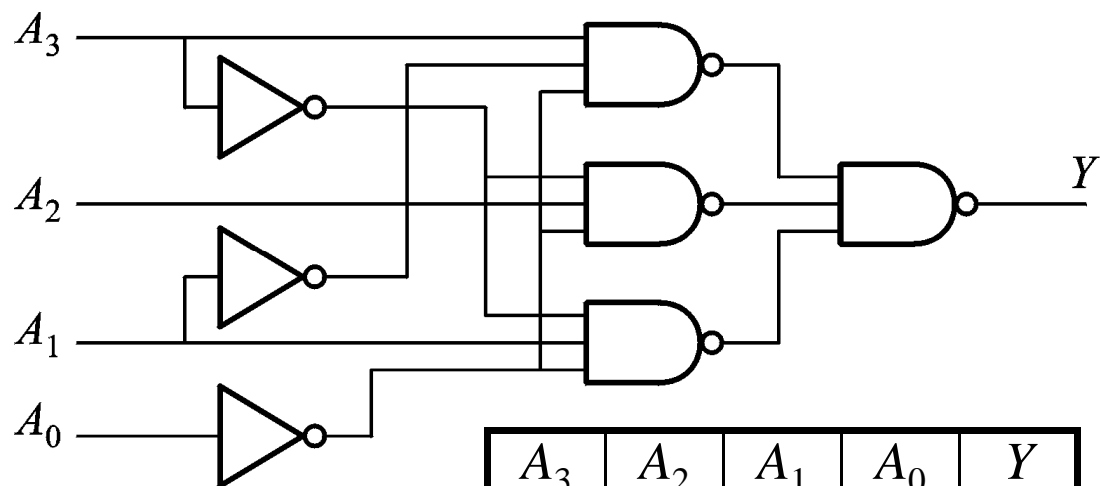
ü 怎么说功能？

（已知功能电路 ... ）

<i>A</i>	<i>B</i>	<i>L</i>
0	0	0
0	1	0
1	0	0
1	1	1

### 【例1.1】

分析右图所示电路功能。



解：（1）根据电路图，写逻辑函数式：

$$\begin{aligned} Y &= \overline{A_3} \overline{A_1} \overline{A_0} + \overline{A_3} A_2 \overline{A_0} + \overline{A_3} A_1 \overline{A_0} \\ &= A_3 \overline{A_1} \overline{A_0} + \overline{A_3} A_2 \overline{A_0} + \overline{A_3} A_1 \overline{A_0} \end{aligned}$$

（2）列真值表

（3）说明功能

除法电路。

输入二进制码小于 8 时，能被 2 整除；

输入二进制码大于 8 时，能被 4 整除。

$A_3$	$A_2$	$A_1$	$A_0$	$Y$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

## Ø 组合逻辑电路（设计）

ü 设计：已知电路的功能，要求画出对应的逻辑电路图。

ü 设计步骤：

根据功能，确定输入输出逻辑变量，以及相关逻辑定义；

根据功能，列出真值表；

（有时，需要根据实际情况确定算法，列出简化真值表）

应用代数法、卡诺图法等，求得输入输出的简化逻辑函数式；

应用规定的门电路实现逻辑函数。

ü 怎么确定算法？

（根据实际情况 ...）

怎么简化逻辑函数式？

（卡诺图，最简与或、或与表达式，根据实际情况 ...）

### 【例1.2】

设计电路，要求能检测 4 位二进制代码的输入是否为 8421BCD 伪码。

解：（1）定义输入为  $A_3 \sim A_0$ ，输出为  $Y$ ；

定义输入为 8421BCD 伪码时，输出为 1。

（2）列出真值表

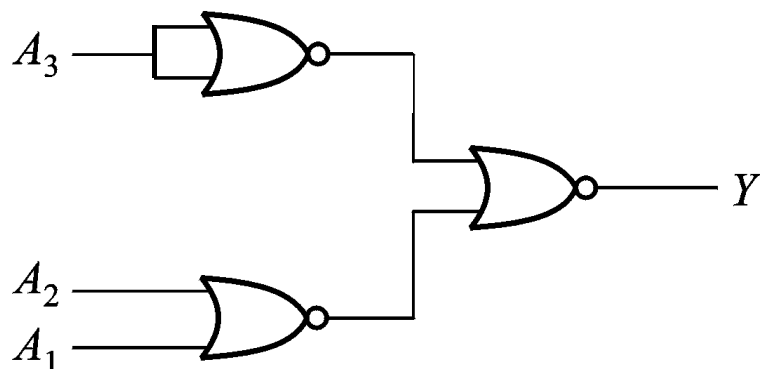
（略）

（3）用卡诺图法化简：

$$Y = A_3A_2 + A_3A_1$$

（4）采用或非门实现：

$$Y = \overline{\overline{A_3} + \overline{A_2} + \overline{A_1}}$$



$Y \backslash A_1 A_0$		$A_3 A_2$			
		00	01	11	10
00	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	1	1



## Ø 组合逻辑电路（竞争与冒险）

ü 前述组合逻辑电路分析和设计，基于稳态；

稳态：输入输出的状态变化时，未考虑门电路的传输延迟时间。

ü 实际电路中，延迟时间是必然存在的；

延迟时间的存在，会使逻辑电路产生异动作，或误动作。

（*JK* 触发器？）

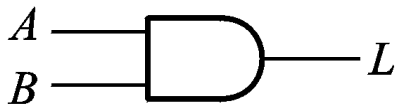
ü *JK* 触发器，利用了延迟时间；

更多情况下，要考虑的是延迟时间的存在对系统工作可靠性的影响。

ü 观察当输入信号的逻辑电平发生变化的瞬间，电路的工作情况。

## Ø 组合逻辑电路（竞争与冒险现象）

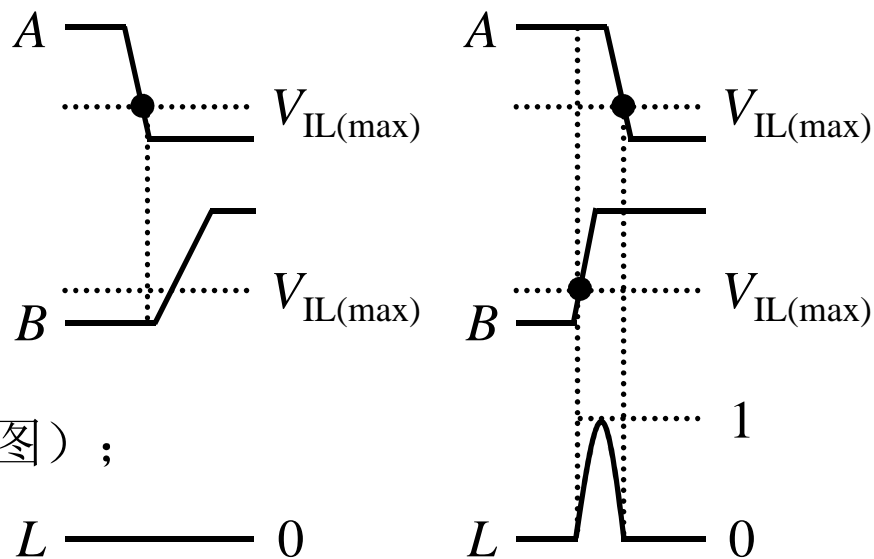
ü 右图所示与门电路。



稳态时，若  $A = 1$  且  $B = 0$ ，或  $A = 0$  且  $B = 1$ ，输出  $L = 0$ 。

ü 设想：在某一时刻， $A$  由 1 跳变为 0，且  $B$  同时由 0 跳变为 1；  
由于传输延迟时间的存在， $A$ 、 $B$  的跳变不可能完全同步。

ü 若  $A$  的跳变相对较快，  
则输出  $L$  始终保持为 0（右图）；  
（无竞争冒险现象）



若  $B$  的跳变相对较快，  
则输出  $L$  会出现尖峰脉冲（毛刺，右图）；  
（竞争冒险现象）

## Ø 组合逻辑电路（竞争与冒险）

🕒 竞争（race）：组合逻辑电路中，当门电路的输入同时向相反方向变化时，由于传输延迟时间的不同，其到达输出门的时间有先有后，导致电路在瞬间可能出现异常情况；

竞争与冒险（race & hazard）：由于竞争，导致组合逻辑电路的输出端可能产生尖峰脉冲；

组合逻辑电路的设计完成后，务必要检查是否存在竞争冒险现象。

🕒 检查原则：

如果输出中同时包含输入  $A$  和  $\bar{A}$ ，且两者经过不同的传输途径而来，那么当输入  $A$  的状态发生突变时，输出端即有可能产生尖峰脉冲。

即，只要输出端的逻辑函数能简化成： $Y = A + \bar{A}$  或  $Y = A \cdot \bar{A}$  则可以判定该电路存在竞争冒险现象。

【例1.3-1】

判断函数  $Y = AB + \overline{A}BC$  是否存在竞争冒险现象。

解：此逻辑函数中存在一对互补变量  $A$  和  $\overline{A}$ ；

且，当  $B = C = 1$  时，函数可变换为： $Y = A + \overline{A}$

所以，电路存在竞争冒险现象。

【例1.3-2】

判断函数  $Y = (A + \overline{B})(B + C)$  是否存在竞争冒险现象。

解：此逻辑函数中存在一对互补变量  $B$  和  $\overline{B}$  ；

且，当  $A = C = 0$  时，函数可变换为： $Y = B \cdot \overline{B}$

所以，电路存在竞争冒险现象。

## Ø 组合逻辑电路（消除竞争与冒险）

ü 方案 1：引入选通脉冲。

ü 原理：

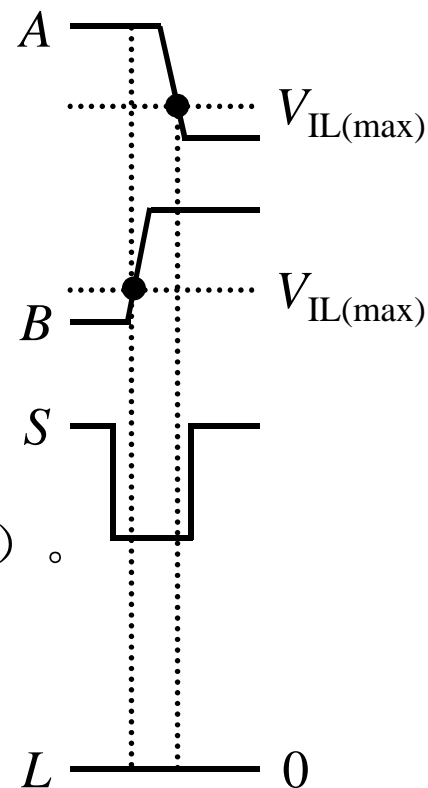
尖峰脉冲是在瞬间产生的，所以只要在这段时间内封锁门电路即可。

（待信号稳定后，选通脉冲有效，输出有效）

（思路简单，且不增加器件数目，  
但必须要增加选通脉冲，  
且对脉冲的宽度和时间有严格要求）

ü 例，右图所示与门电路。  $\overline{S}$   $\begin{matrix} A \\ B \end{matrix}$   $\rightarrow$   $L$

在门电路的输入端增加选通控制信号  $S$ （上、右图）。



## Ø 组合逻辑电路（消除竞争与冒险）

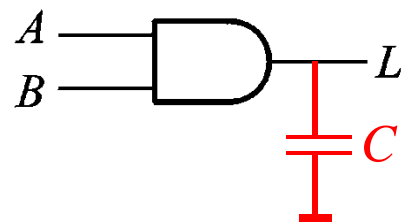
ü 方案 2：接滤波电容。

ü 原理：

尖峰脉冲很窄（高频信号），所以只要在输出端接一小电容即可。

（方法简单，但是增加了输出波形的上升时间和下降时间，使波形边沿变坏；一般适用于对输出波形边沿无严格要求的场合）

ü 例，下图所示与门电路。



## Ø 组合逻辑电路（消除竞争与冒险）

ü 方案 3：修改逻辑设计，增加冗余项。

ü 原理：

使逻辑表达式中不出现  $A + \bar{A}$  或  $A \cdot \bar{A}$ 。

（有时可以获得很好的效果，但一般需增加器件）

ü 例，将  $Y = AB + \bar{A}BC$  调整为： $Y = AB + \bar{A}BC + BC$

将  $Y = (A + \bar{B})(B + C)$  调整为： $Y = (A + \bar{B})(B + C)(A + C)$



## ✓ 编码器

ü 编码器：将一个特定对象变换成一组二进制码的组合逻辑电路。

ü 例：单位、家庭、部门、运动员等用一组  $n$  位的十进制代码表示；实现代码表示的具体电路就是编码器。

## Ø 基本编码器

☺ 例：将 4 个开关量编制成 1 组 2 位二进制代码。

（4 线 – 2 线编码器）

☺ 真值表

输入的编码对象，与输出的代码，  
具有一一对应关系。

$$Y_1 = \overline{W_0}\overline{W_1}W_2\overline{W_3} + \overline{W_0}\overline{W_1}\overline{W_2}W_3$$

$$Y_0 = \overline{W_0}W_1\overline{W_2}\overline{W_3} + \overline{W_0}\overline{W_1}\overline{W_2}W_3$$

编码器输入				2 位码输出	
$W_0$	$W_1$	$W_2$	$W_3$	$Y_1$	$Y_0$
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

利用约束条件（不能同时输入两个以上编码对象），化简：

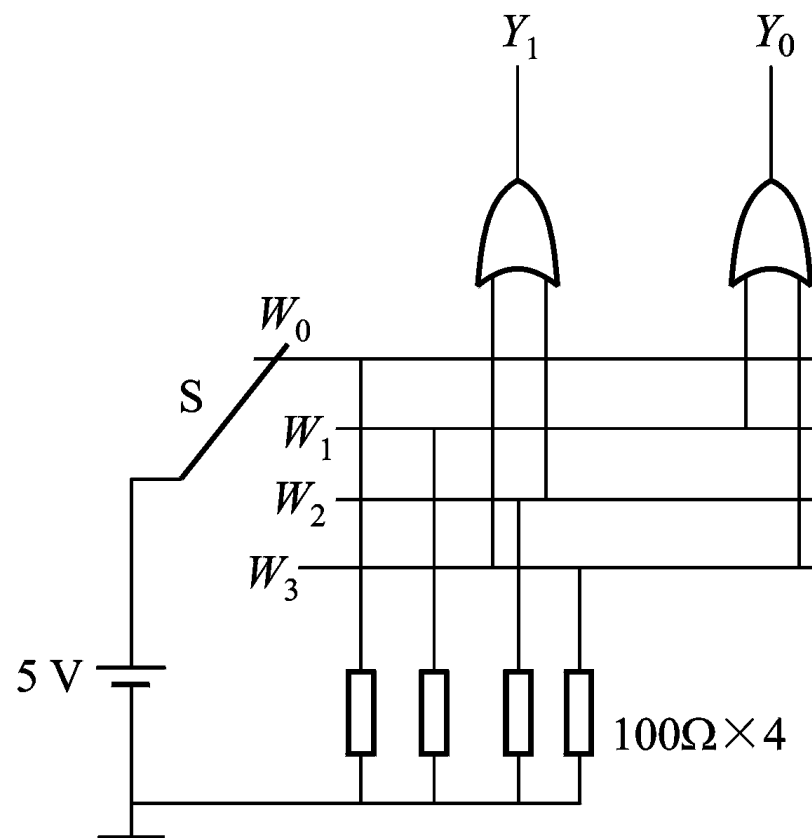
$$Y_1 = W_2 + W_3$$

$$Y_0 = W_1 + W_3$$

## Ø 基本编码器

ü 右图所示逻辑电路。

ü 组合逻辑电路设计：  
逻辑变量、逻辑定义；  
列真值表；  
逻辑简化；  
应用门电路实现逻辑函数。



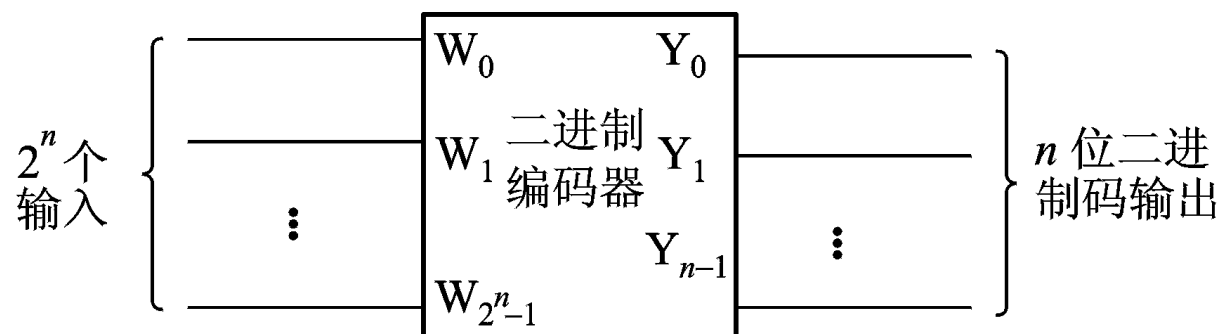
$$Y_1 = W_2 + W_3$$

$$Y_0 = W_1 + W_3$$

## Ø 二进制编码器

ü 功能：将  $2^n$  个输入对象编制成  $n$  位二进制代码输出。

ü 电路结构图

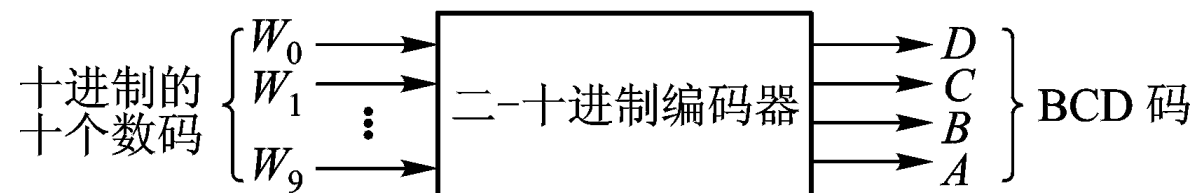


ü 常见形式：4-2 编码器、8-3 编码器、16-4 编码器 ...

## Ø 二-十进制编码器

ü 功能：将 10 个输入的十进制数编制成 4 位 BCD 码输出。

ü 电路结构图



ü 由于 BCD 码的编码方案非常多（8421、5421、2421、余三码等），因此二-十进制编码器的电路方案也多种多样。

## Ø 优先编码器

- ü 允许同时输入两个（或以上）的编码对象；  
但只对其中优先权最高的输入对象实现编码。
- ü 优先权的高低在设计时事先约定。

### 【例1.4】

设计：将十进制 10 个数字按 8421BCD 方案编码的优先编码器。

定义：输入为低电平有效，输出为 4 位 BCD 反码。

解：（1）定义输入的十进制 10 个数字分别为： $\overline{W}_9 \sim \overline{W}_0$

输出的 8421 BCD 编码分别为： $\overline{Y}_3 \sim \overline{Y}_0$

（2）根据题意，  
列真值表。

（3）根据真值表，  
求逻辑函数：

$$\begin{aligned}\overline{\overline{Y}}_3 &= \overline{\overline{W}}_9 \overline{\overline{W}}_8 + \overline{\overline{W}}_9 \\ &= W_9 + W_8\end{aligned}$$

$$\overline{Y}_3 = \overline{W_8 + W_9}$$

（4）根据函数式，  
用门电路实现。

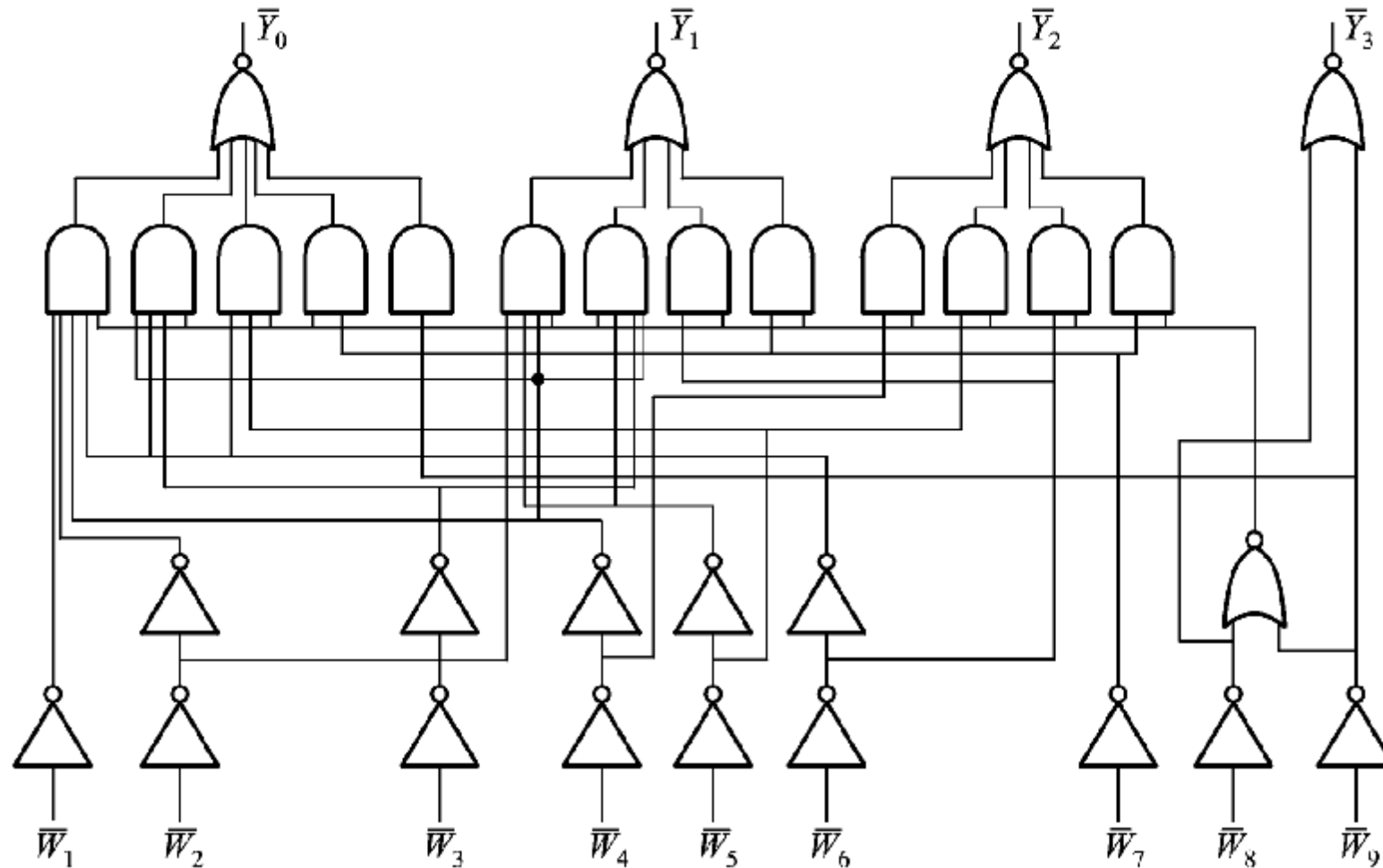
编 码 器 输 入										BCD 码输出			
$\overline{W}_9$	$\overline{W}_8$	$\overline{W}_7$	$\overline{W}_6$	$\overline{W}_5$	$\overline{W}_4$	$\overline{W}_3$	$\overline{W}_2$	$\overline{W}_1$	$\overline{W}_0$	$\overline{Y}_3$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$
1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	0	×	1	1	1	0
1	1	1	1	1	1	1	0	×	×	1	1	0	1
1	1	1	1	1	1	0	×	×	×	1	1	0	0
1	1	1	1	1	0	×	×	×	×	1	0	1	1
1	1	1	1	0	×	×	×	×	×	1	0	1	0
1	1	1	0	×	×	×	×	×	×	1	0	0	1
1	1	0	×	×	×	×	×	×	×	1	0	0	0
1	0	×	×	×	×	×	×	×	×	0	1	1	1
0	×	×	×	×	×	×	×	×	×	0	1	1	0

$$\bar{Y}_3 = \overline{\bar{W}_8 + W_9}$$

$$\bar{Y}_2 = \overline{(\bar{W}_9 \bar{W}_8)(W_4 + W_5 + W_6 + W_7)}$$

$$\bar{Y}_1 = \overline{(\bar{W}_9 \bar{W}_8)(\bar{W}_5 \bar{W}_4 W_3 + \bar{W}_5 \bar{W}_4 W_2 + W_7 + W_6)}$$

$$\bar{Y}_0 = \overline{\bar{W}_9 \bar{W}_8 (W_7 + \bar{W}_6 W_5 + \bar{W}_6 \bar{W}_4 W_3 + \bar{W}_6 \bar{W}_4 \bar{W}_2 W_1) + W_9}$$





## ✓ 译码器

ü 译码器：与编码器的功能相反，将输入的二进制代码所表示的对象还原。

ü 基本分类：

二进制译码器（变量译码器）；

二—十进制译码器（显示译码器、码制变换译码器）。

## 基本译码器

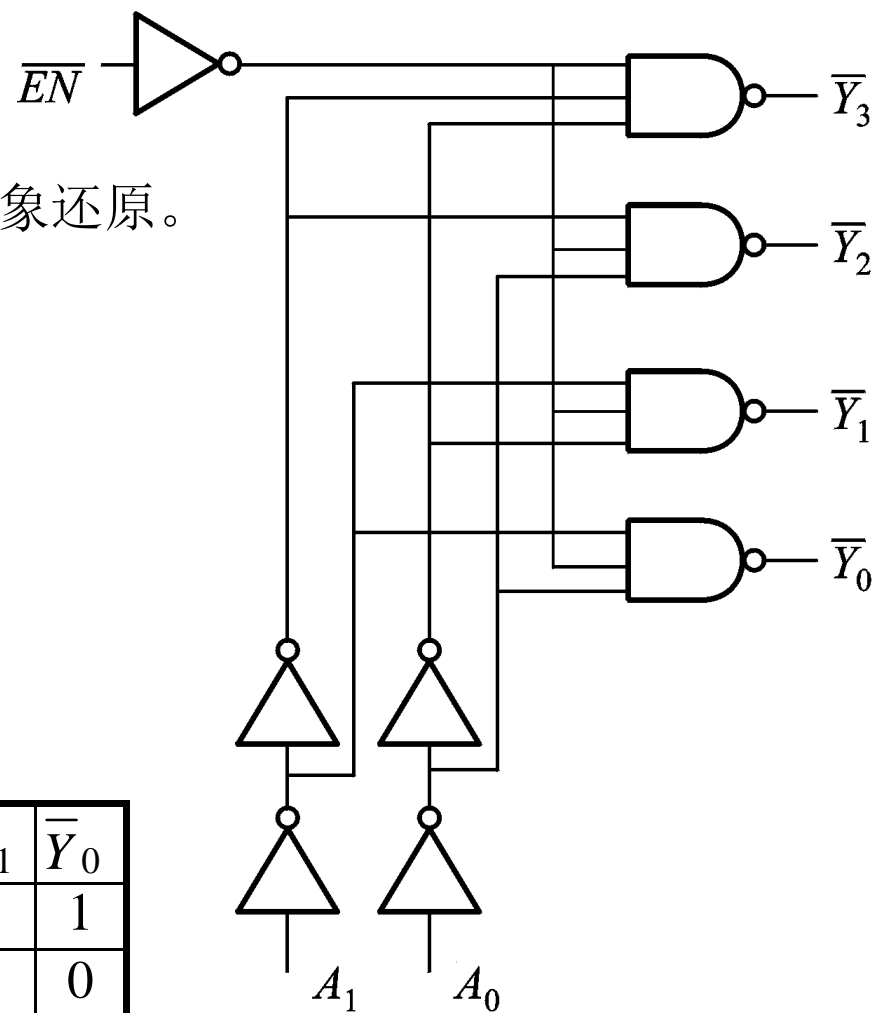
例：将 2 位代码所表示的 4 种对象还原。

右图所示逻辑电路。

$A_1$ 、 $A_0$ ：输入；  
 $\overline{EN}$ ：使能控制端；  
 $\overline{Y}_3 \sim \overline{Y}_0$ ：输出。

真值表

$\overline{EN}$	$A_1$	$A_0$	$\overline{Y}_3$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$
1	×	×	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1



## Ø 基本译码器

ü 每组输入代码仅对应某一特定输出对象；  
每个输出对象为输入代码的一个最小项。

ü 逻辑表达式

$$\overline{Y}_3 = \overline{\overline{\overline{EN}}} \overline{A_1} \overline{A_0}$$

$$\overline{Y}_2 = \overline{\overline{\overline{EN}}} \overline{A_1} A_0$$

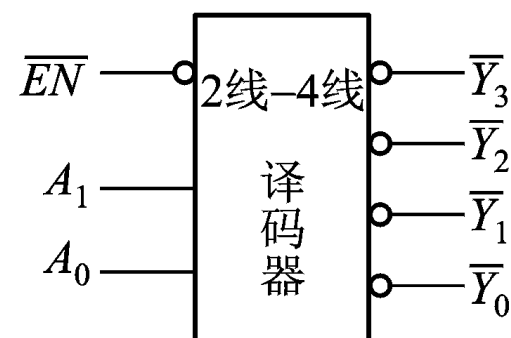
$$\overline{Y}_1 = \overline{\overline{\overline{EN}}} A_1 \overline{A_0}$$

$$\overline{Y}_0 = \overline{\overline{\overline{EN}}} A_1 A_0$$

ü 右下图所示简化电路符号。

ü 真值表

$\overline{EN}$	$A_1$	$A_0$	$\overline{Y}_3$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$
1	×	×	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1



## Ø 二进制译码器

ü 功能：将  $n$  位二进制代码代表的  $2^n$  个对象还原出来。

ü 电路结构图



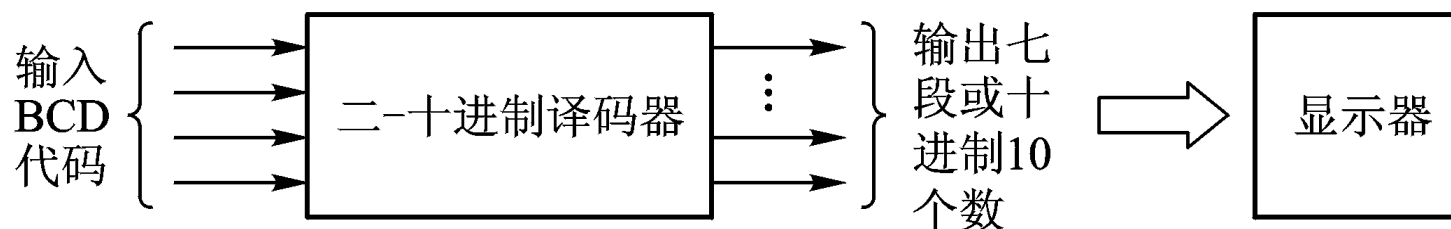
ü 常见形式：2-4 译码器、3-8 译码器、4-16 译码器 ...

## Ø 二-十进制译码器

ü 功能：将 4 位 BCD 码代表的 10 个输入的十进制代码还原出来。

ü 译码后的结果经常用于显示器显示；  
又称显示译码器、码制变换译码器。

ü 电路结构图



ü 输出函数由译码器所要驱动的显示器类型决定；  
常用显示器类型：点阵式、分段式（包括半导体数码管、荧光数码管和液晶）等。

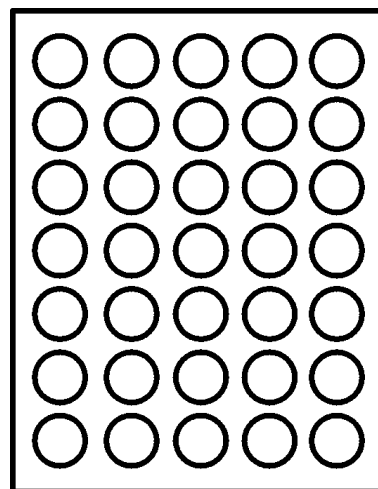
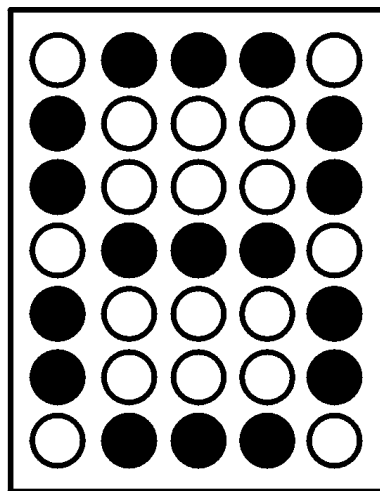
## Ø 点阵式显示器

ü 组成形式：发光二极管、灯泡等。

ü 用途：广告、字符发生器等。

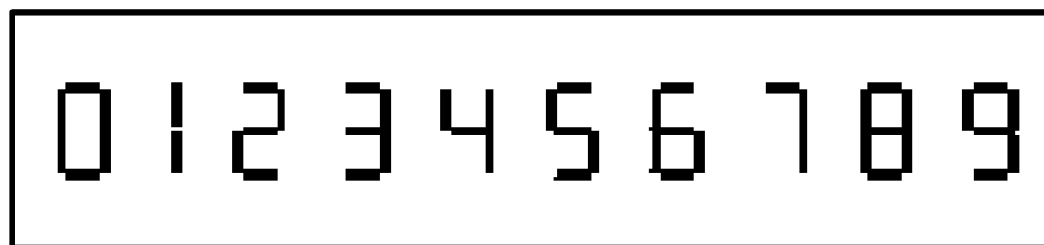
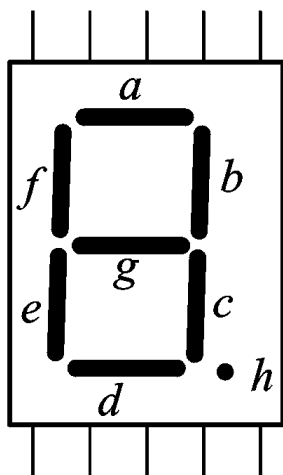
ü 右图所示  $5 \times 7$  点阵。

下图显示数字 8：

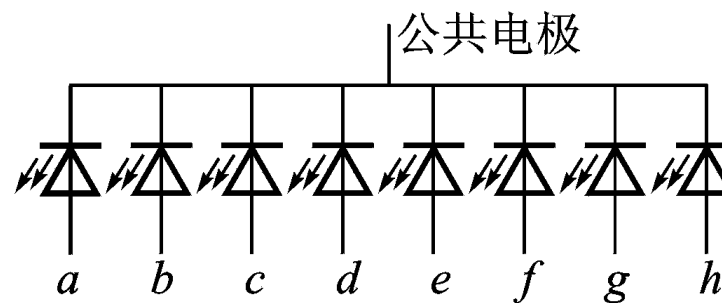
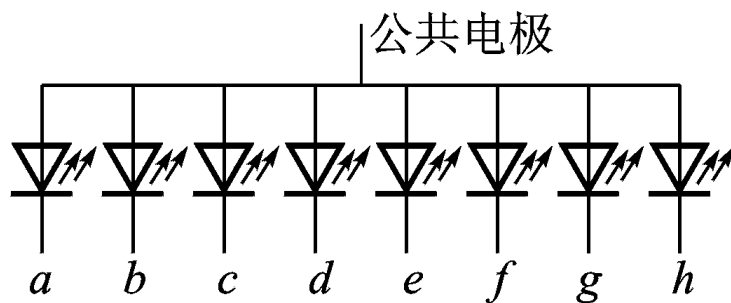


## Ø 半导体数码管（LED）

ü 类型：七段或八段。



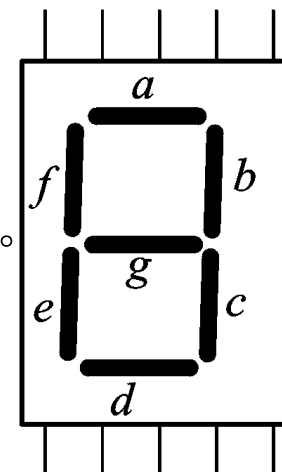
ü 结构：共阳极和共阴极。



### 【例1.5】

用非门、或非门设计一个二 - 十进制译码器。

要求：输入是 8421BCD 码，输出能驱动共阴七段半导体数码管。



解：（1）定义输入的 8421 BCD 编码分别为：  $D C B A$  ；

输出分别为：  $a \sim g$  。

（2）根据题意，  
列真值表。

（3）根据真值表，  
求逻辑函数。

（卡诺图）

（4）根据函数式，  
用门电路实现。

具体参书 P163

输入				输出							显示
$D$	$C$	$B$	$A$	$a$	$b$	$c$	$d$	$e$	$f$	$g$	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

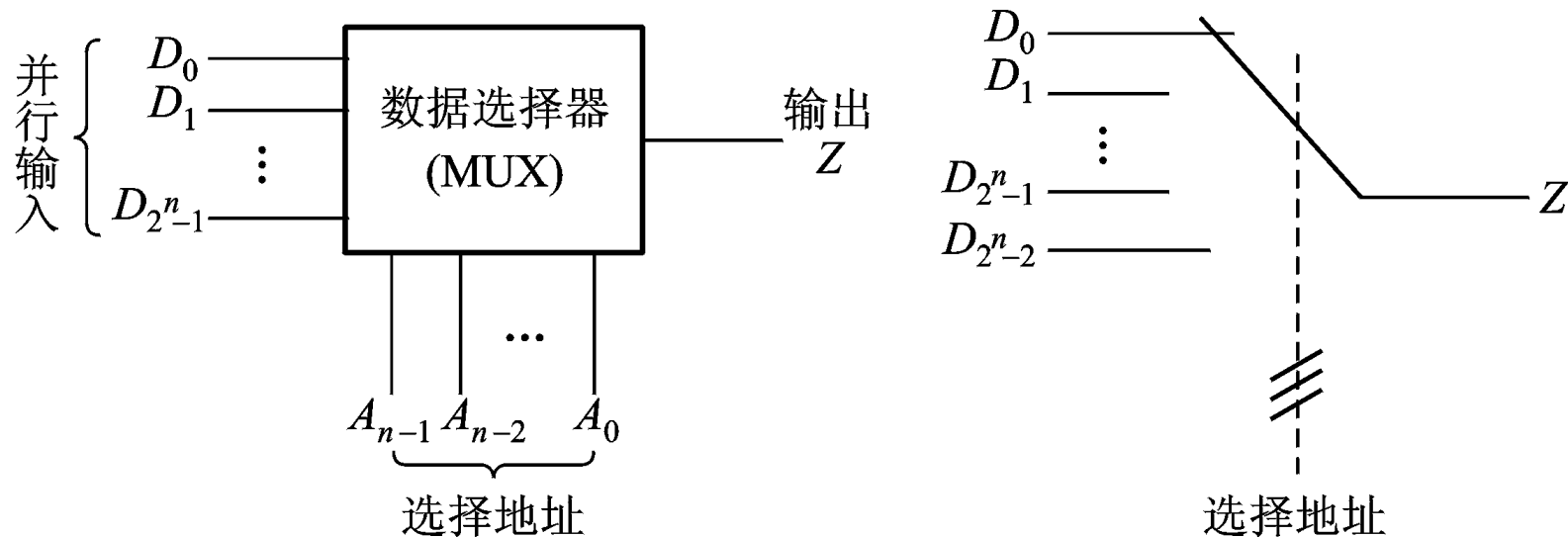


## ✓ 数据选择器

ü 在数字信号传输系统中，有时需要从一组信号中选出某一个；  
在数据采集系统中，有时需要从多路数据中选出某一路；  
能实现上述功能的电路就是数据选择器。

ü 数据选择器在数据采集、数字信号处理与通信系统中，被大量应用。

ü 基本结构及其等效电路（MUX）。



## Ø 数据选择器

ü 例：将 4 路并行输入数据，  
在 2 位地址码控制下，分别选通输出。  
(4 选 1 数据选择器)

ü 右图所示逻辑电路。

$D_3 \sim D_0$ ：并行输入数据；

$A_1$ 、 $A_0$ ：地址码输入；

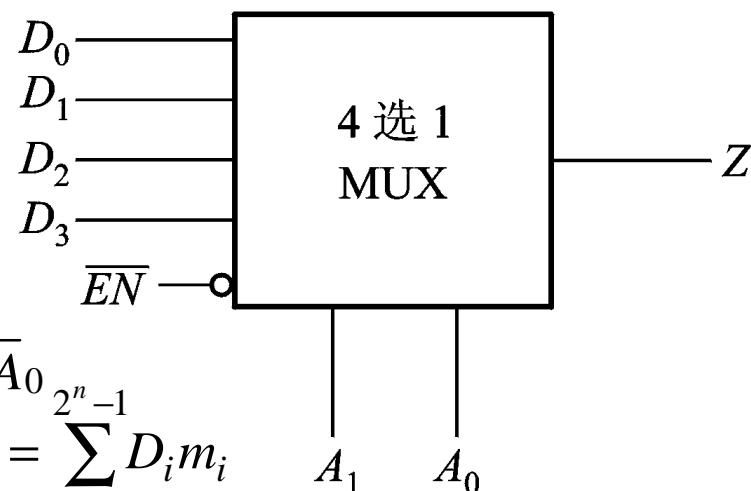
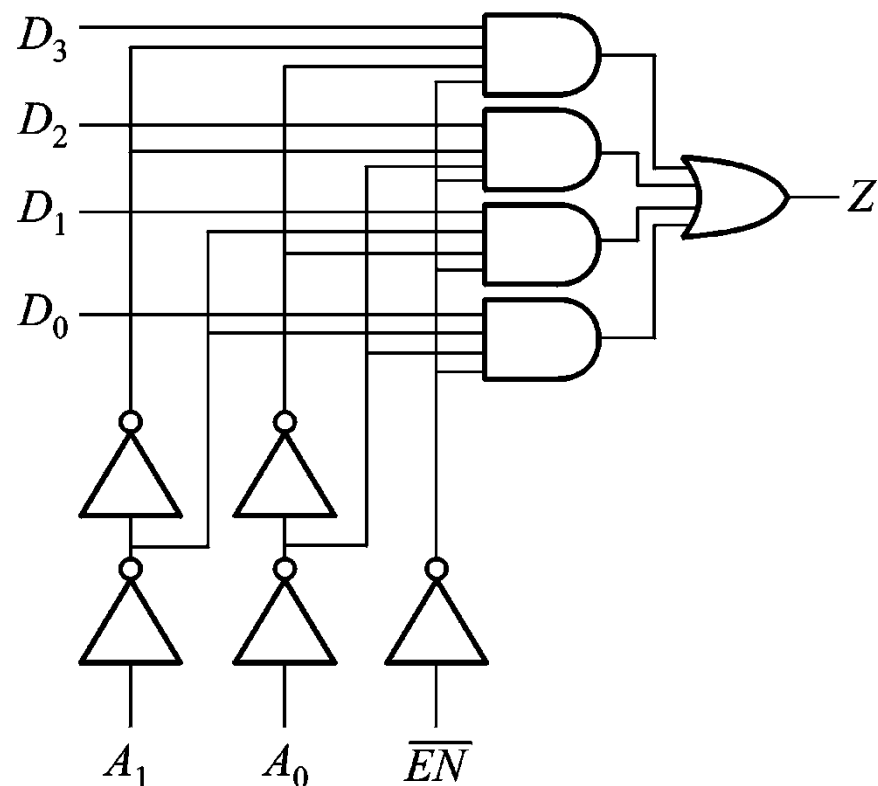
$\overline{EN}$ ：使能控制端；

$Z$ ：输出。

ü 右图所示逻辑符号。

ü 使能端有效时：

$$\begin{aligned} Z &= D_3 A_1 A_0 + D_2 A_1 \overline{A_0} + D_1 \overline{A_1} A_0 + D_0 \overline{A_1} \overline{A_0} \\ &= D_3(m_3) + D_2(m_2) + D_1(m_1) + D_0(m_0) = \sum_{i=0}^{2^n-1} D_i m_i \end{aligned}$$



## Ø 数据选择器（特征）

ü 从电路的结构看：

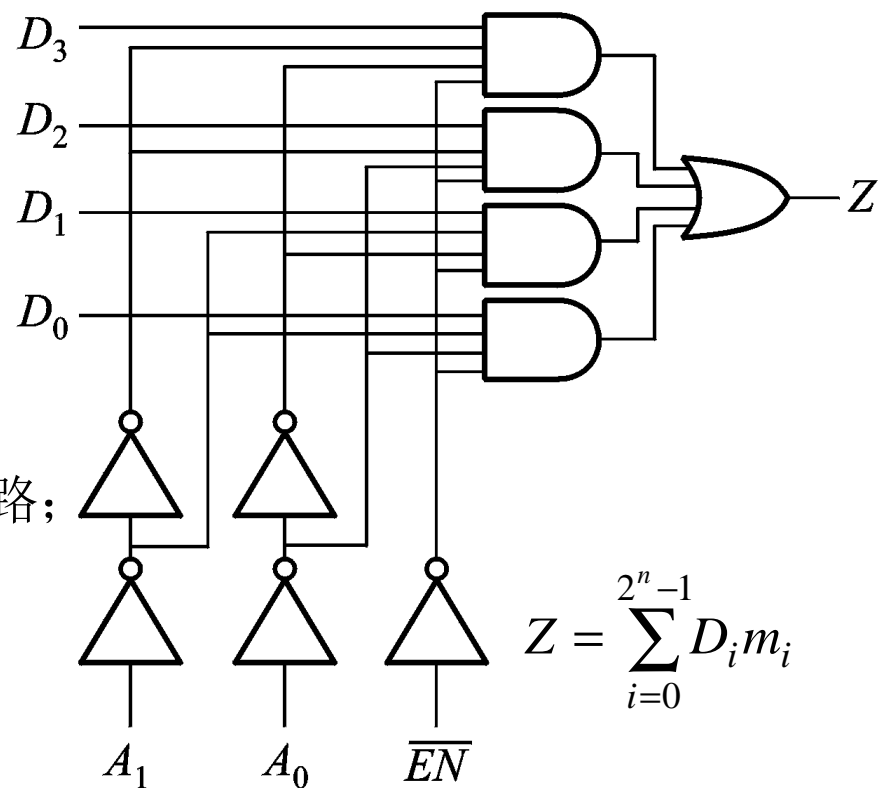
这是一个与或逻辑结构；

从数据的传输方式看：

这是一个并行/串行数据的传输转换电路；

从实现的逻辑函数看：

这是一个与或（最小项之和）表达式。



ü 因此，用该电路可以实现：

任意的组合逻辑函数（组合逻辑电路）。

（具体的例题参下一章节内容）

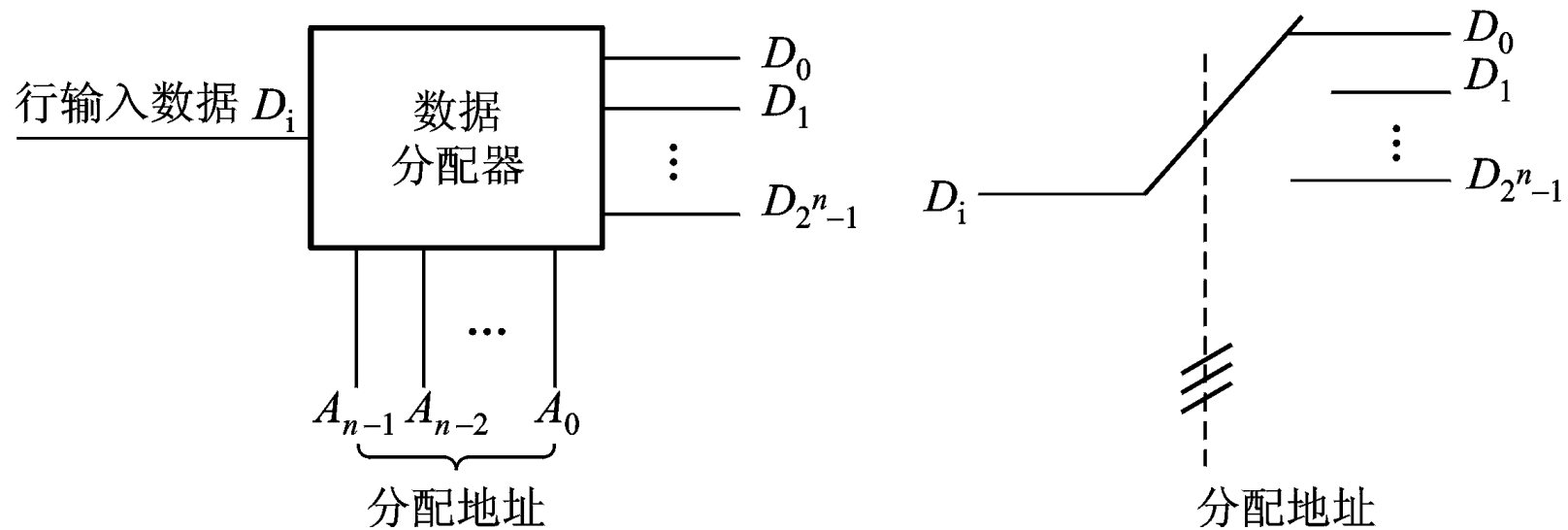
## ✓ 数据分配器

ü 数据分配器将串行输入数据，在分配地址的控制下，分别送至相应的输出通道；

功能和数据选择器相反。

ü 数据分配器在数据采集、数字信号处理与通信系统中，被大量应用。

ü 基本结构及其等效电路。



## Ø 数据分配器

ü 例：将 1 路串行输入数据，  
在 2 位地址码控制下，分别选通输出。  
(1 分 4 数据分配器)

ü 右图所示逻辑电路。

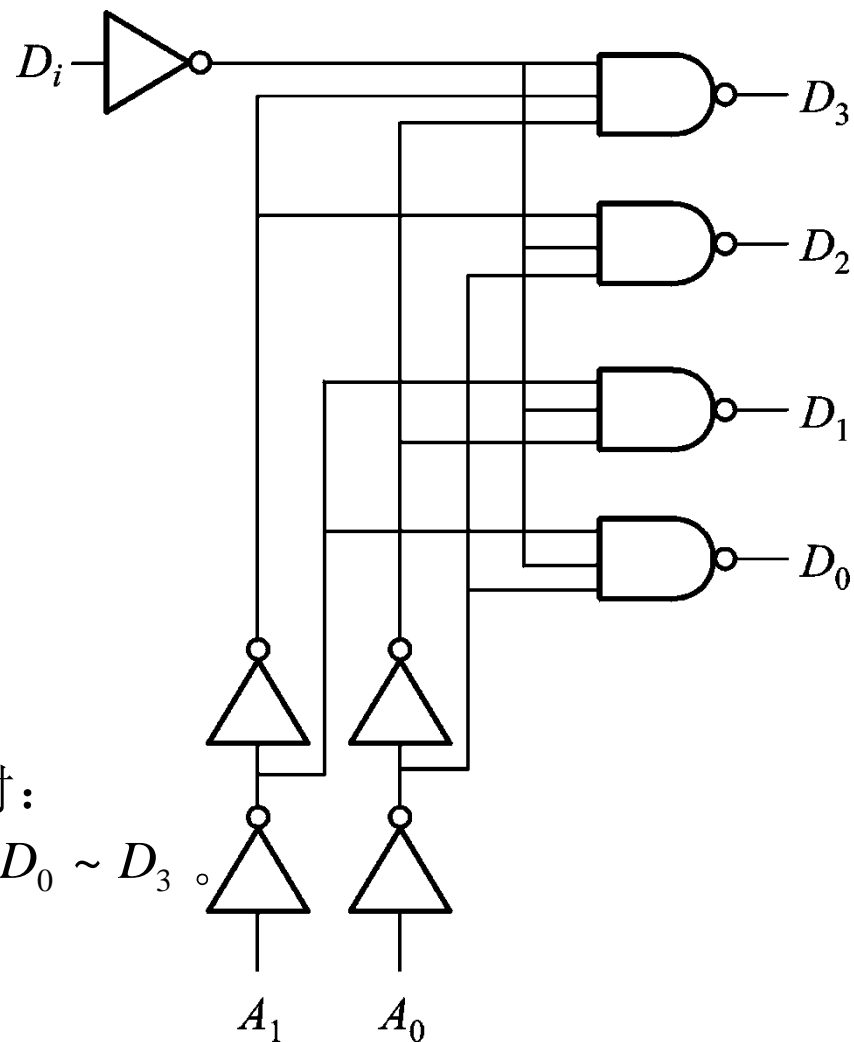
$D_i$ ：串行输入数据；

$A_1$ 、 $A_0$ ：地址码输入；

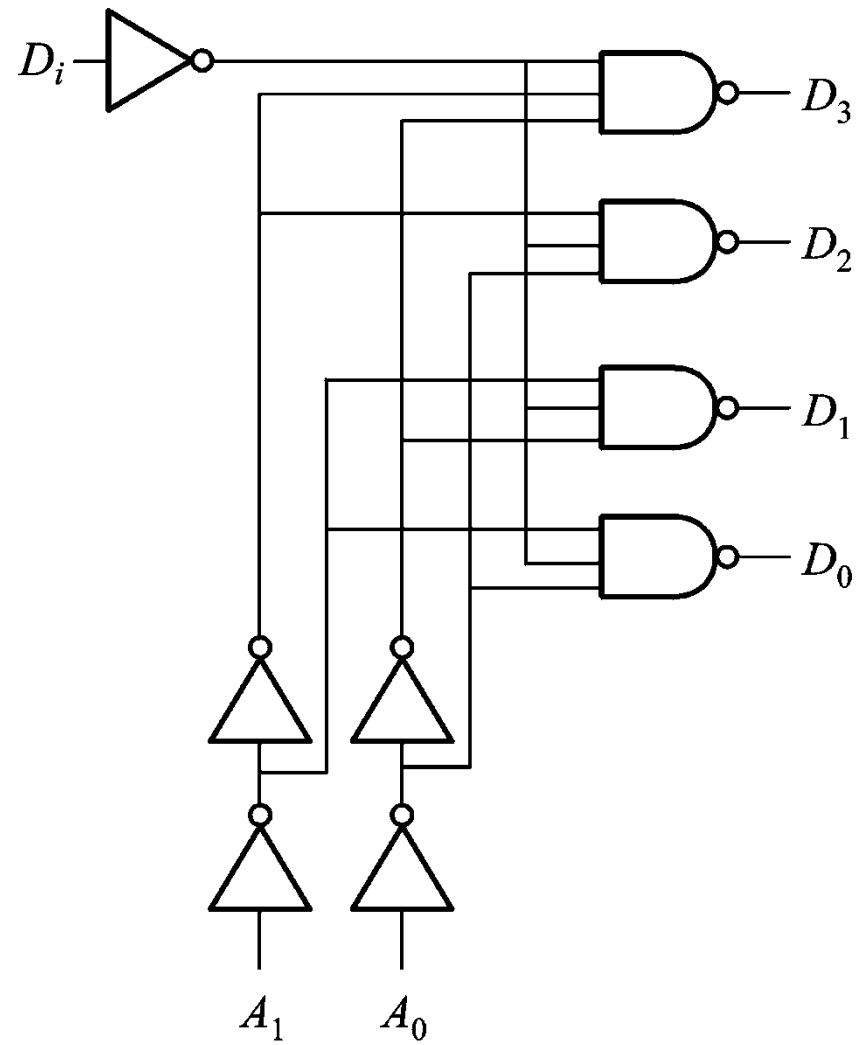
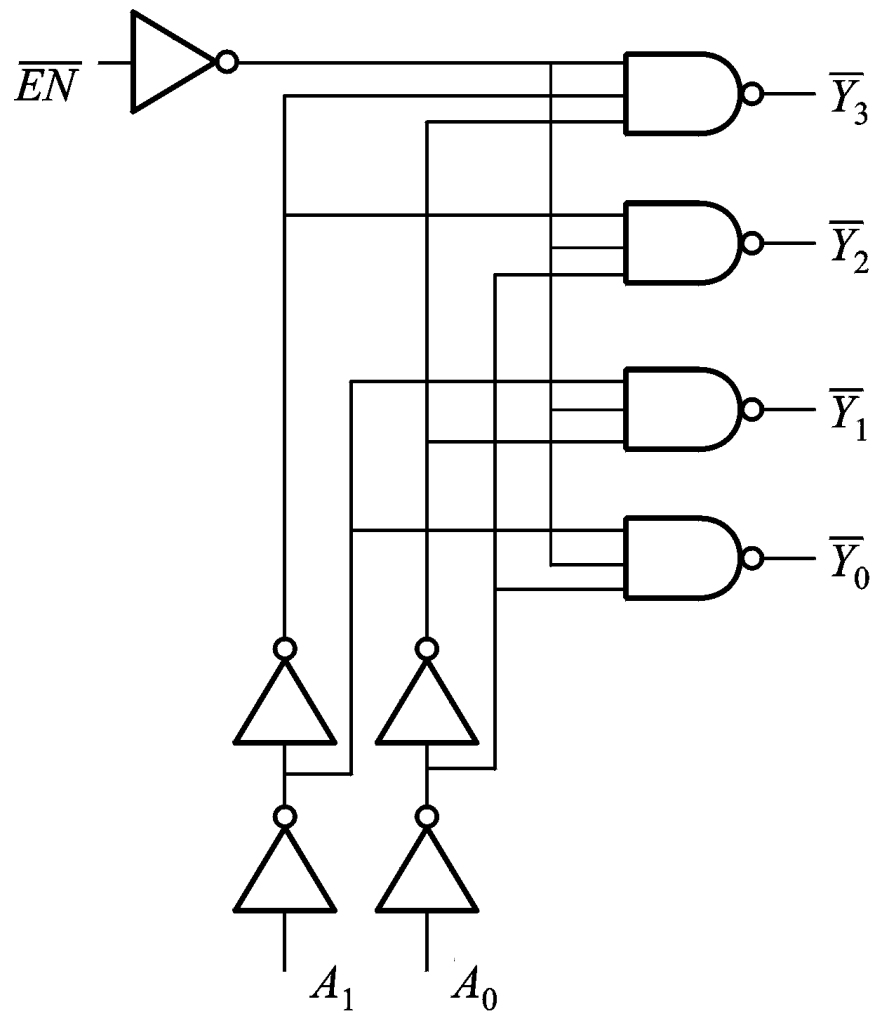
$\overline{D}_3 \sim \overline{D}_0$ ：输出。

ü 当地址码分别为 00、01、10 和 11 时：

串行输入数据  $D_i$  分别被选通输出至  $D_0 \sim D_3$ 。



## 数据分配器



数据分配器 = 具有使能端的译码器

## ✓ 二进制加法器

ü 数字系统要完成各种复杂运算和操作，首先必须具备加、减、乘、除这四种最基本的算术运算。

ü 减法可以由补码加法实现；

加法配合移位操作，能实现乘除法的运算。

ü 加法电路是最基本的数字运算电路；

实现二进制加法运算的基本电路有：半加器、全加器。

## 半加器

半加器：实现被加数和加数相加，不考虑相邻低位的进位。

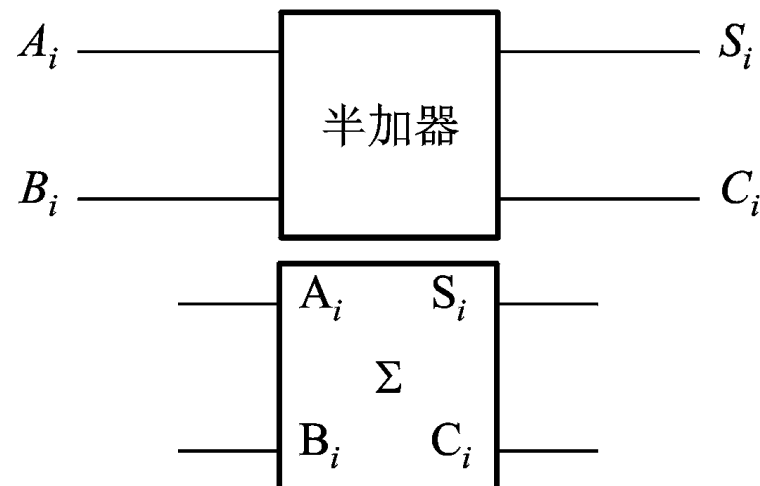
右图所示电路结构。

$A_i$ 、 $B_i$ ：被加数、加数；

$S_i$ ：和；

$C_i$ ：进位。

右图所示逻辑符号。



真值表

$A_i$	$B_i$	$S_i$	$C_i$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

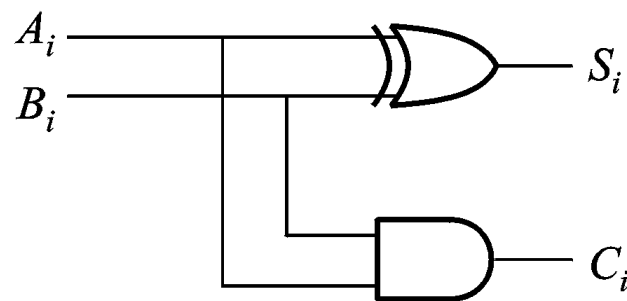


## Ø 半加器（实现电路）

ü 根据真值表，有： $S_i = \bar{A}_i B_i + A_i \bar{B}_i = A_i \oplus B_i$ ， $C_i = A_i B_i$

ü 可由异或门 + 与门实现半加器。

还可以由与非门、或非门... 实现



ü 真值表

$A_i$	$B_i$	$S_i$	$C_i$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

## Ø 全加器

ü 全加器：实现被加数和加数，以及相邻低位进位的加法。

ü 右图所示电路结构。

$A_i$ 、 $B_i$ 、 $C_{i-1}$ ：被加数、加数、低位进位；

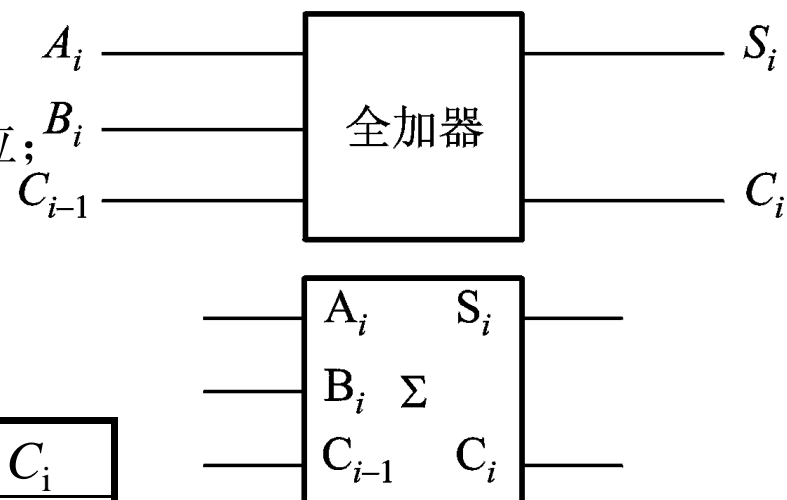
$S_i$ ：和；

$C_i$ ：进位。

ü 右图所示逻辑符号。

ü 真值表

$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
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



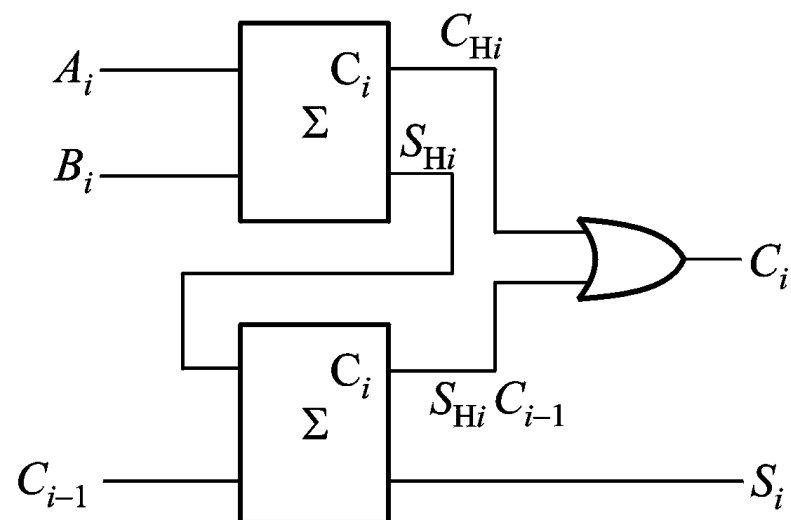
## Ø 全加器（实现电路）

ü 根据真值表（并整理后），有：

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

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

ü 可由半加器 + 或门实现。

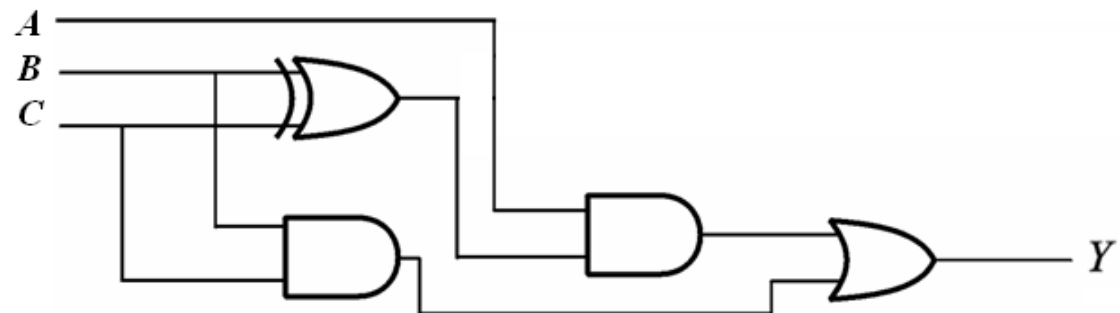
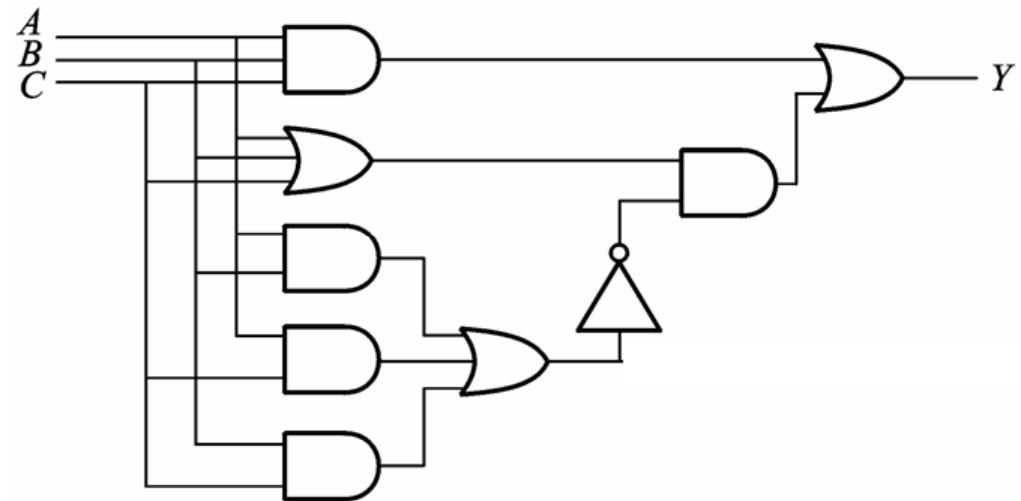
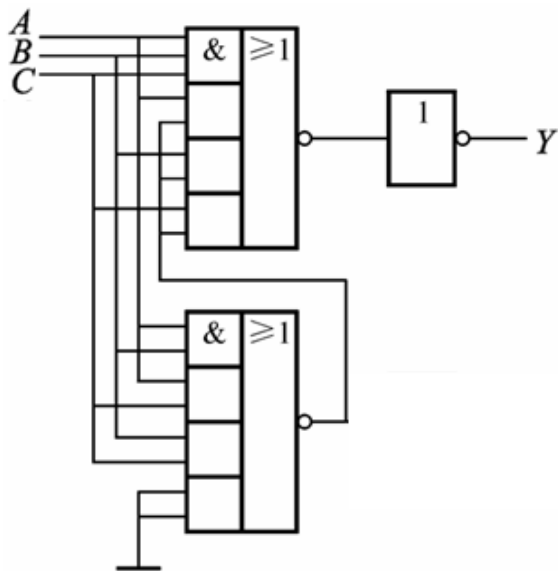
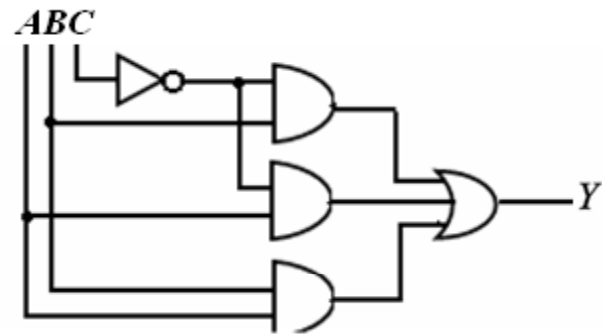
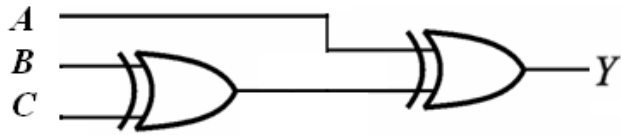


还可以由与非门、或非门、与或非门... 实现

ü 真值表

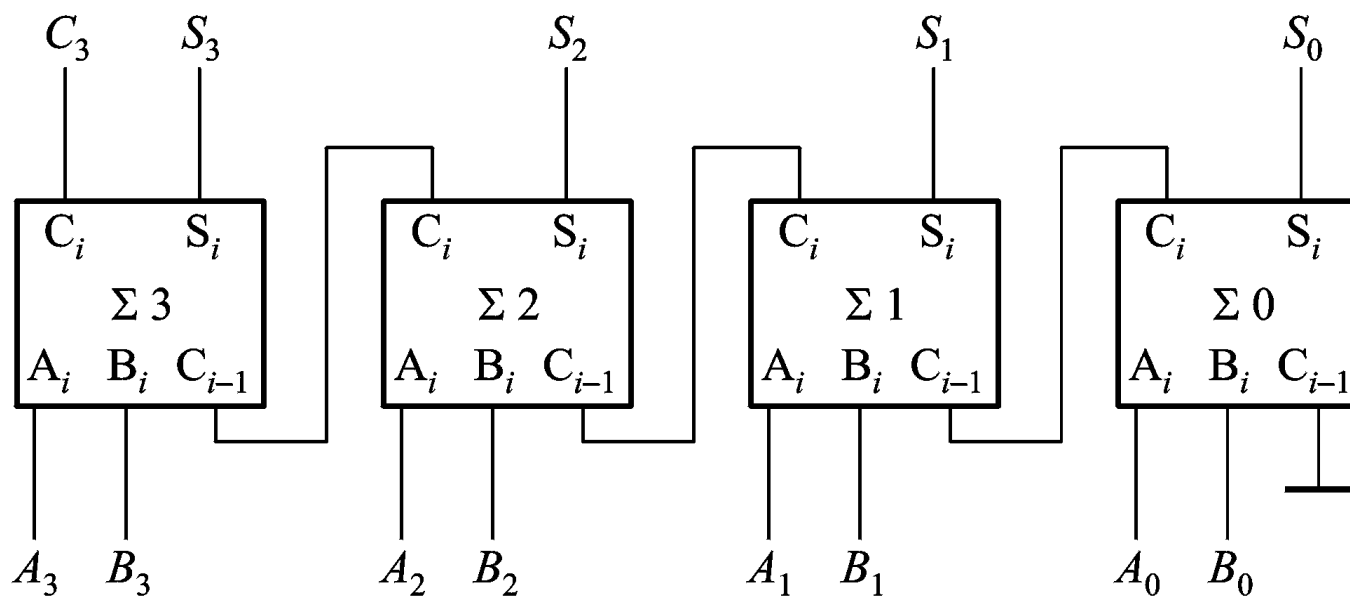
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

## Ø 全加器（学习兴趣）



## Ø 全加器（多位二进制）

ü 下图所示 4 位二进制串行进位加法器。



ü  $A_3 \sim A_0$ 、 $B_3 \sim B_0$ 、 $C_{i-1}$ ：被加数、加数、低位进位；  
 $S_3 \sim S_0$ 、 $C_3$ ：和、进位。

ü 缺陷：进位速度慢（可能需要经过 4 个全加器的传输延迟时间）。

## ✓ 数值比较器

- ü 数值比较器可用来比较两个数据的大、小或是否相等；  
可用于逻辑判断，执行程序的跳转路径或执行何种操作等。

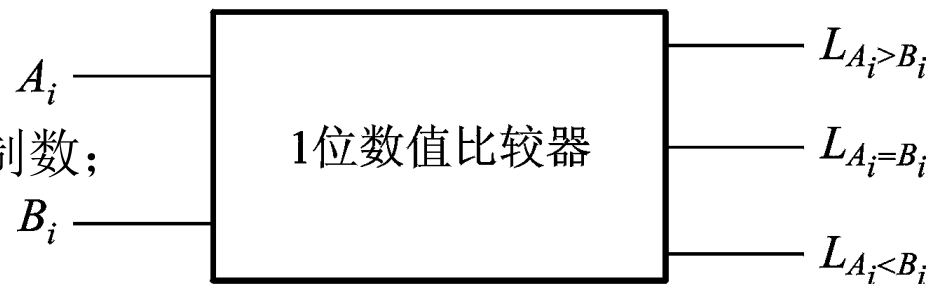
## Ø 数值比较器

ü 例：1 位二进制数值比较器。

ü 右图所示电路结构。

$A_i$ 、 $B_i$ ：待比较的两个 1 位二进制数；

$L_{A>B}$ 、 $L_{A=B}$ 、 $L_{A<B}$ ：比较结果。



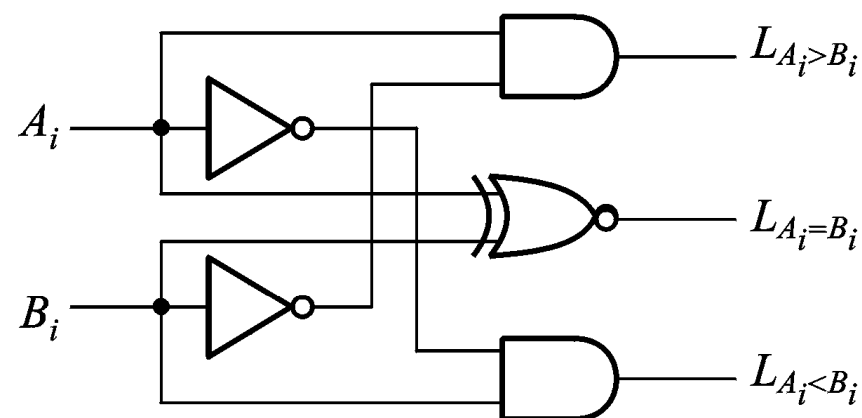
ü 真值表

$A_i$	$B_i$	$L_{A>B}$	$L_{A<B}$	$L_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

## Ø 数值比较器（实现电路）

ü 根据真值表，有： $L_{A_i > B_i} = A_i \bar{B}_i$ ， $L_{A_i < B_i} = \bar{A}_i B_i$ ， $L_{A_i = B_i} = \bar{A}_i \bar{B}_i + A_i B_i$

ü 可由异或门 + 与门 + 非门实现。



ü 真值表

$A_i$	$B_i$	$L_{A > B}$	$L_{A < B}$	$L_{A = B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1



## Ø 数值比较器（多位二进制）

☺ 以 1 位数值比较器为基础，按高位优先原则，可实现多位数值比较。

☺ 例：4 位二进制数值比较器真值表

4 位二进制数输入				低位比较结果输入			比较结果输出		
$A_3, B_3$	$A_2, B_2$	$A_1, B_1$	$A_0, B_0$	$l_{a>b}$	$l_{a=b}$	$l_{a<b}$	$L_{a>b}$	$L_{a=b}$	$L_{a<b}$
$G_3$	×	×	×	×	×	×	1	0	0
$L_3$	×	×	×	×	×	×	0	0	1
$E_3$	$G_2$	×	×	×	×	×	1	0	0
$E_3$	$L_2$	×	×	×	×	×	0	0	1
$E_3$	$E_2$	$G_1$	×	×	×	×	1	0	0
$E_3$	$E_2$	$L_1$	×	×	×	×	0	0	1
$E_3$	$E_2$	$E_1$	$G_0$	×	×	×	1	0	0
$E_3$	$E_2$	$E_1$	$L_0$	×	×	×	0	0	1
$E_3$	$E_2$	$E_1$	$E_0$	1	0	0	1	0	0
$E_3$	$E_2$	$E_1$	$E_0$	0	0	1	0	0	1
$E_3$	$E_2$	$E_1$	$E_0$	0	1	0	0	1	0

## ✓ 本节作业

### ü 习题 4（P231）

4、补充题 1；

补充题 2、补充题 3（下周）。

### ü 说明：

4 题，“与或”门，改成“或”门。

所有的题目，需要有解题过程（不是给一个答案即可）。

## Ø 补充题 1

ü 设计：4 输入的优先编码器；

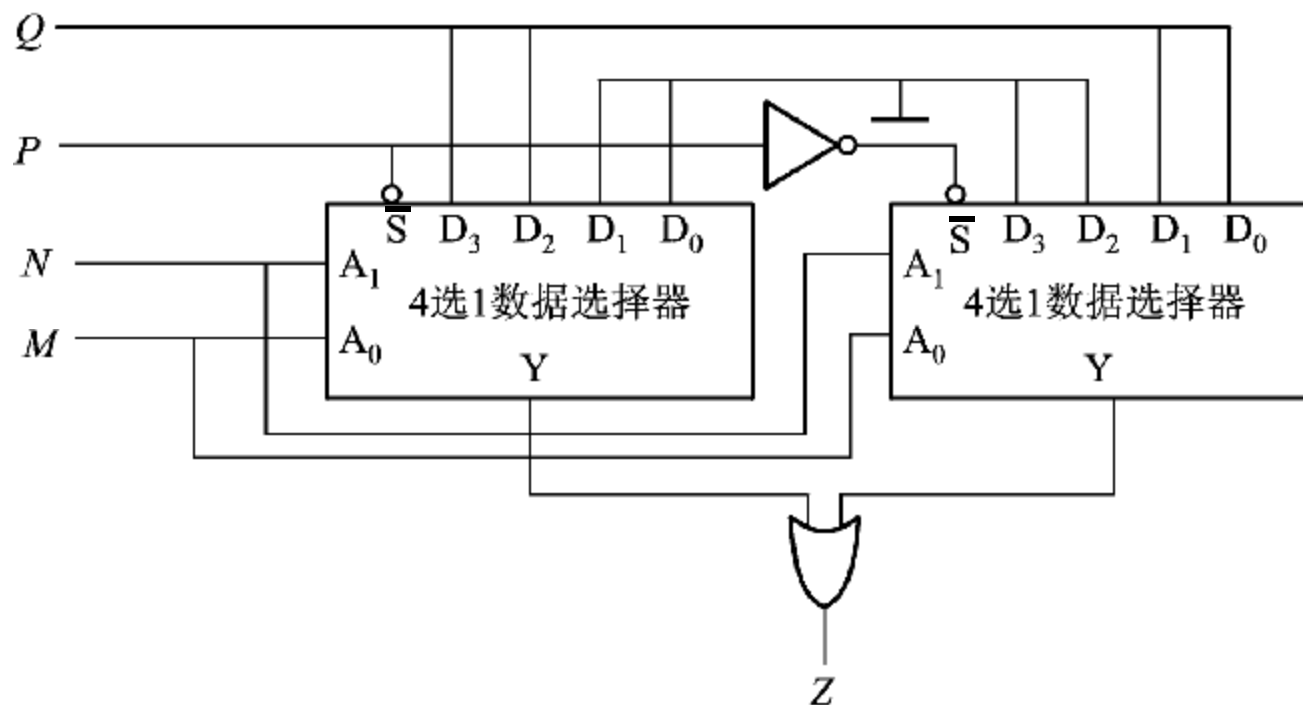
要求：大数优先原则；

所有端口均为高电平有效；

用“与非”门（逻辑）实现该电路。

## Ø 补充题 2

ü 下图所示由两个 4 选 1 数据选择器组成的逻辑电路。



已知：数据选择器满足  $Y = (D_0 \bar{A}_1 \bar{A}_0 + D_1 \bar{A}_1 A_0 + D_2 A_1 \bar{A}_0 + D_3 A_1 A_0) \cdot \bar{S}$

要求：写出输出  $Z$  与输入  $M$ 、 $N$ 、 $P$ 、 $Q$  之间的逻辑函数式。

### Ø 补充题 3

ü 设计：1 位二进制数的全减器；

定义：被减数  $A$ ，减数  $B$ ，从低位来的借位信号  $J_0$ ；

差数  $D$ ，向高位的借位信号  $J_1$ ；

用“与非”门（逻辑）实现该电路。