第4章 数字逻辑电路

数字逻辑电路一般分为组合逻辑电路和时序逻辑电路，在对组合逻辑电路和时序逻辑电路分析和设计基础上，重点介绍中规模数字集成电路的功能和应用。

组合逻辑电路，任意时刻的输出状态仅取决于该时刻的输入信号，而与电路原来的状态无关。电路的输出与输入之间无反馈，组合逻辑电路不需要记忆元件。

时序逻辑电路，输出状态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。

4.1.2 基本组合逻辑电路的分析与设计

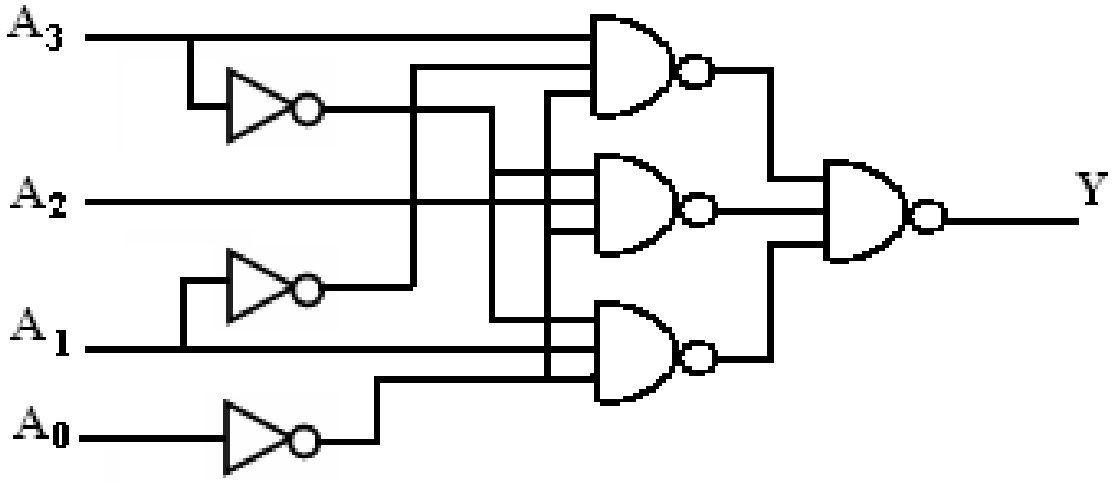
基本组合逻辑电路是根据简化的逻辑函数式，用各种逻辑符号画出来的电路。电路的输出状态（结果），只由当时电路的各输入取值决定。一旦输入取值确定后，输出结果就可以确定。

常见的组合逻辑电路很多：二进制数的四则运算电路、编码电路、译码电路、奇偶校验电路、数据分配器和数据选择器等。

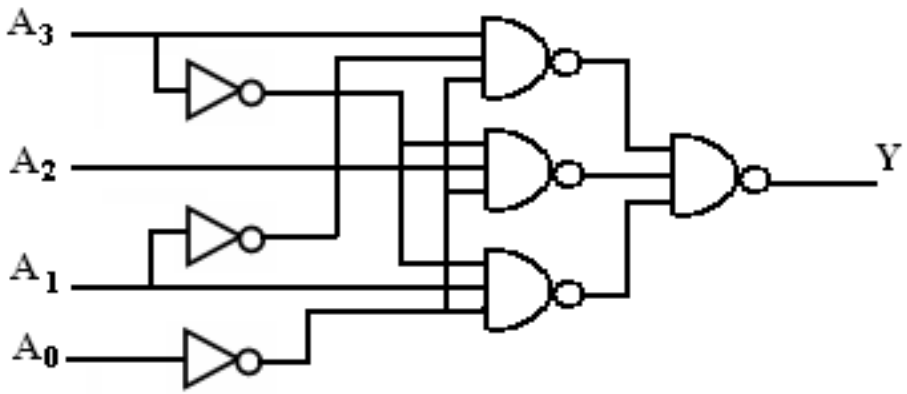
一、基本组合逻辑电路的分析和设计方法

组合电路的分析方法：（1）应先写出每一位输出的逻辑表达式，（2）在给定各个变量的取值后，列出真值表，（3）最后确定电路的逻辑功能。

例 请分析给出电路的逻辑功能。



解：



由于电路是单输出，所以输出函数为：

$$Y = f(A_3, A_2, A_1, A_0) = \overline{\overline{A_3}\,\overline{A_1}\,\overline{A_0} \cdot \overline{A_3}A_2\,\overline{A_0} \cdot \overline{A_3}A_1\,\overline{A_0}}$$

$$= A_3\,\overline{A_1}\,\overline{A_0} + \overline{A_3}A_2\,\overline{A_0} + \overline{A_3}A_1\,\overline{A_0}$$

4位二进制码输入				输出
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

从真值表可知：

当输入4位二进制码小于8时，能被2整除；而输入二进制码大于8时，能被4整除的一个除法电路。

2.组合电路设计

设计是分析的反过程,通常要实现的功能要求是给定的,选定门电路后,能设计出完成该功能的具体电路。

一般设计过程为:

（1）根据题意或给定功能要求找出输入和输出逻辑变量;

（2）列出真值表;

（3）求出各个输出的最简与—或表达式（建议用卡诺图法）;

（4）用规定的逻辑门画出整个逻辑电路图。

例 设计一个4位二进制代码输入时，检测8421BCD伪码的组合逻辑电路。

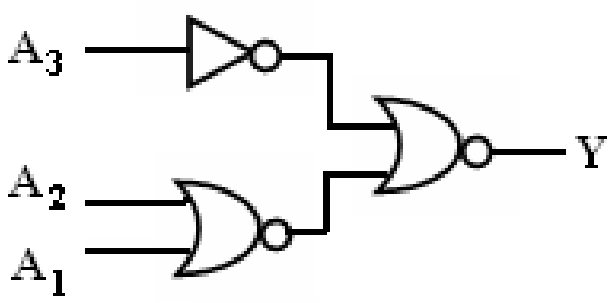
解：

分析设计要求，输入是4变量，设为$A_3A_2A_1A_0$，一个检测结果设为Y。

由于逻辑关系比较简单，所以直接填卡诺图，得出结果。

假定输入4位码是8421码时，输出为“0”，反之输出为“1”，其卡诺图如图所示。

Y A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1



$$Y = (A_3, A_2, A_1, A_0) = A_3A_2 + A_3A_1$$

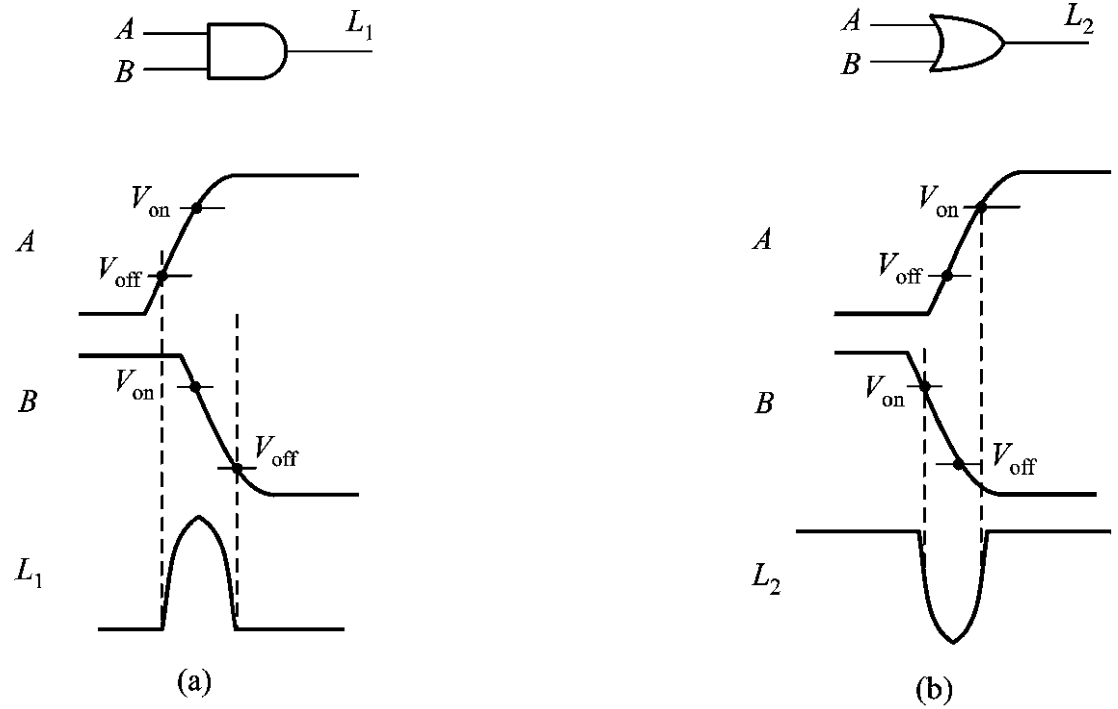
$$= A_3(A_2 + A_1) = \overline{\overline{A_3} + \overline{A_2 + A_1}}$$

例：设计一个组合逻辑电路，其输入$X=X_3X_2X_1X_0$为8421BCD码，输出$Y=Y_3Y_2Y_1Y_0$为二进制数。要求：当$0\leqslant X\leqslant 3$时，$Y=X^2$,当$X>3$时，$Y=X+4$。试求：（1）列出真值表；（2）写出每一个输出的最简“与—或—非”表达式。

三、组合逻辑电路中的竞争与冒险

1、 竞争—冒险现象的产生

在组合逻辑电路中，当电路从一种稳定状态转换到另一种稳定状态的瞬间，某个门电路的两个输入信号同时向相反方向变化（一个从“1”变为“0”，另一个从“0”变为“1”），由于传输延迟时间的不同，所以到达输出门的时间有先有后，这种现象称为竞争。由于竞争而在逻辑电路的输出端有可能产生尖峰脉冲，把这种现象叫做竞争—冒险



2、竞争—冒险现象的判别方法

输出端的逻辑函数在一定条件下能简化成如下表达式：

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

则可以判定该电路存在竞争—冒险

【例】 判断下列逻辑函数表达式是否存在竞争—冒险现象。

(1) $Y = AB + \overline{A}BC$

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

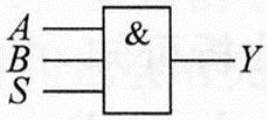
解：(1) 由于逻辑函数中存在一对互补变量A和 $\overline{A}$，当B=C=“1”时，函数将成为 $Y = A + \overline{A}$，故电路存在竞争—冒险现象。

(2) 由于逻辑函数中存在一对互补变量B和 $\overline{B}$，当A=C=“0”时，函数将成为 $Y = B \cdot \overline{B}$，故电路存在竞争—冒险现象。

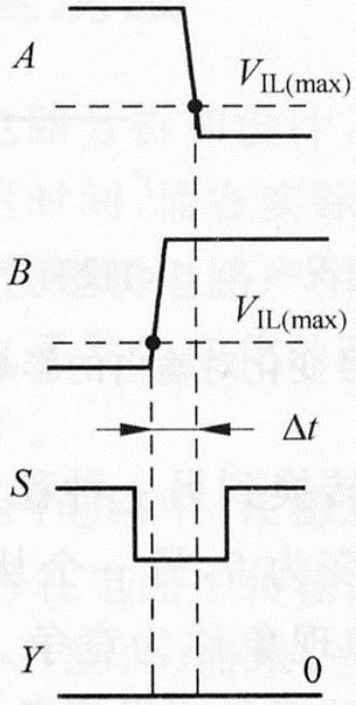
3、 消除竞争—冒险现象的方法

（1）引入选通脉冲

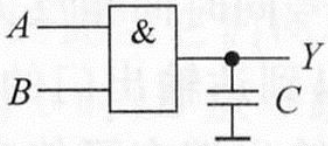
（2）接入滤波电容



(a)与门



(b)引入选通脉冲



(c)接入滤波电容

（3）修改逻辑设计，增加冗余项

增加冗余项或乘上冗余因子，使之不出现 $A+\overline{A}$ 或 $A\cdot\overline{A}$ 的形式，即可消除冒险现象。

【例】$Y=AB+\overline{A}BC$

B=C=1时存在竞争—冒险现象 → $Y=AB+\overline{A}BC+BC$

【例】$Y=(A+\overline{B})(B+C)$

在A=C=0时产生竞争—冒险现象 → $Y=(A+\overline{B})(B+C)(A+C)$

4.1.4 常见逻辑电路功能分析与设计

一、编码器与译码器

编码器

编码器将一个特定对象变换成一组二进制码的电路。

如一个单位、一户家庭、一个部门、一个运动员等都可用一组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。

1. 基本编码器

如将4个开关量编制成4组二位二进制代码。

真值表为：

编码器输入				二位码输出	
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

真值表说明：

（1）同一时间只允许1个编码对象输入，其余不能输入；

（2）一个对象和一组代码相对应。00代表W_0、01代表W_1、10代表W_2、11代表W_3；

二位代码输出中每位的函数为：

编码器输入				二位码输出	
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

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

$$Y_0 = \overline{W}_0 W_1 \overline{W}_2 \overline{W}_3 + \overline{W}_0 \overline{W}_1 \overline{W}_2 W_3$$

利用同时不能出现二个以上编码对象的约束条件，化简后得：

编码器输入				二位码输出	
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

$$Y_1 = \overline{W}_0\overline{W}_1W_2\overline{W}_3 + \overline{W}_0\overline{W}_1\overline{W}_2W_3$$

$$= \overline{W}_0\overline{W}_1W_2\overline{W}_3 + W_0W_2 + W_1W_2 + W_3W_2$$

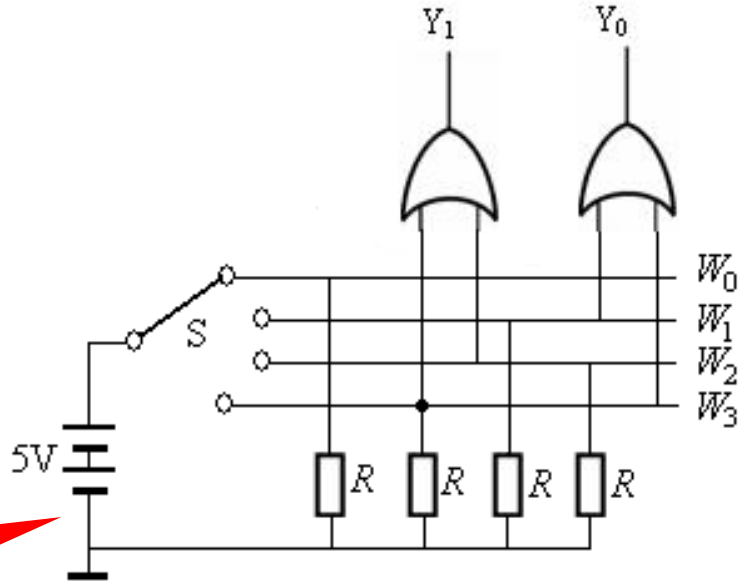
$$+ \overline{W}_0\overline{W}_1\overline{W}_2W_3 + W_0W_3 + W_1W_3 + W_2W_3$$

$$= \overline{W}_0\overline{W}_1\overline{W}_3W_2 + (W_0 + W_1 + W_3)W_2$$

$$+ \overline{W}_0\overline{W}_1\overline{W}_2W_3 + (W_0 + W_1 + W_2)W_3$$

$$= W_2 + W_3$$

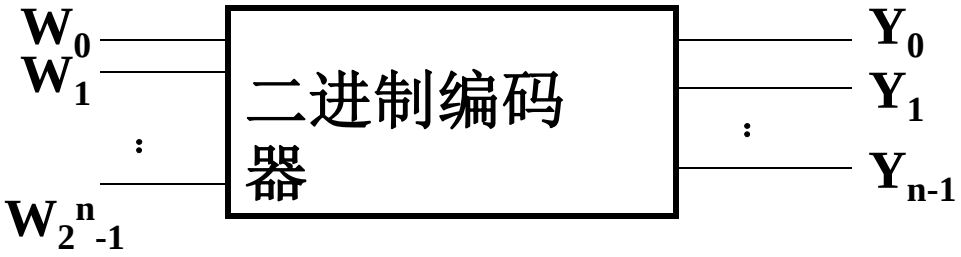
$$Y_0 = W_1 + W_3$$



4线—2线编码器

2. 二进制编码器

将2^n个特定对象编制成n位二进制代码的一种组合逻辑电路。它在数字系统中应用的非常普遍，例如：4线-2线（4/2）编码器，8线-3线（8/3）编码器，16线-4线（16/4）编码器等。



3. 二-十进制编码器

它是将十进制的0～9十个数字，用一组4位的二-十进制代码（BCD码）表示。

4. 优先编码器

这种编码器允许同时输入二个或二个以上的输入信号，但编码器只对其中优先权最高的待编码对象实施编码。编码对象的优先权高低可以在设计时预先规定。

4.2 中规模集成逻辑电路及应用

把这些基本组合逻辑电路集成化，加上电源和某些控制端后，就成为一片中规模集成电路。

中规模集成电路的功能完善，连接和功能扩展方便。

集成门电路系列型号

1、TTL逻辑电路系列

74××　　标准系列

74L××　　低功耗系列

74H××　　高速系列

74S××　　肖特基系列

74LS××　　低功耗肖特基系列

74AS××　　先进的肖特基系列

74ALS××　　先进的低功耗肖特基系列

2、CMOS逻辑器件系列

4000系列	标准系列
74C××系列	普通系列
74HC/HCU/HCT ××系列	高速系列
74AC/ACT ×× 系列	先进CMOS系列

74HCT ××和74ACT××系列可直接与TTL相兼容；

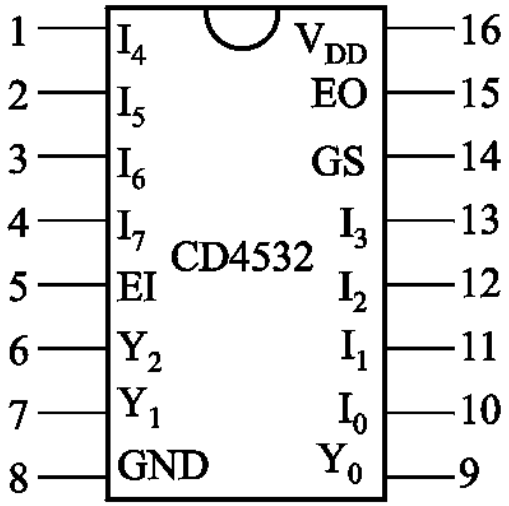
74HC能够直接驱动TTL电路，而TTL电路却不能直接驱动74HC

在中规模集成电路中，通常给出某电路的功能表和芯片引脚图，然后依据基本电路的工作原理，将该中规模集成电路应用起来。

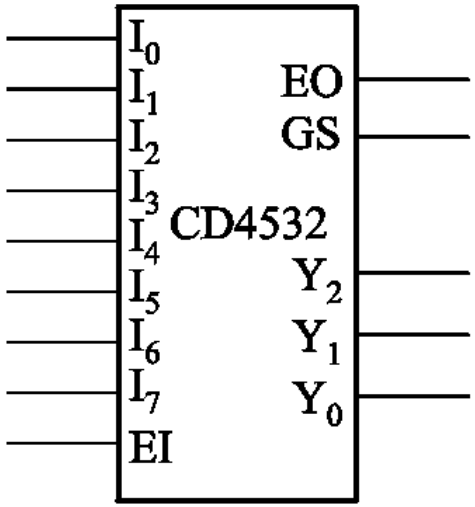
因此，学习方法是首先读懂该中规模集成电路的功能表，相应引脚功能。

1. 中规模集成编码器及应用

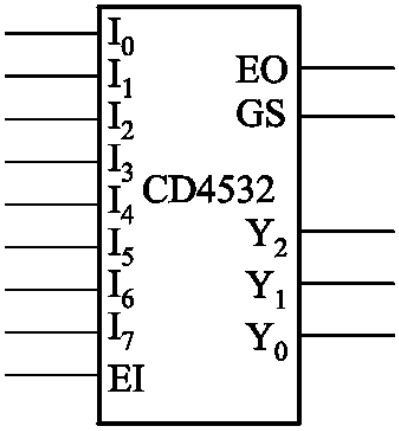
CD4532是一片应用广泛的8线—3线中规模集成优先编码器。



引脚排列图



简化逻辑图



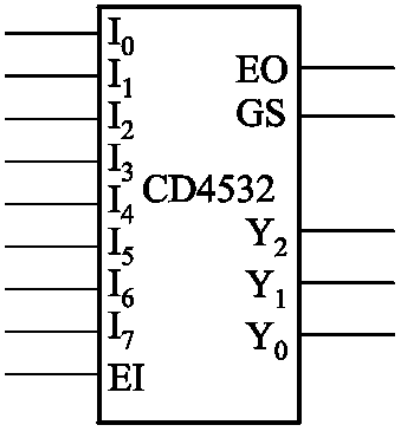
CD4532功能表

编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	GS	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0
1	0	0	0	0	0	1	×	×	0	1	0	1	0
1	0	0	0	0	0	0	1	×	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0

（1）明了输入/输出，8个输入高电平有效，大数优先，三位原码输出，是一个8线/3线优先编码器。

CD4532功能表

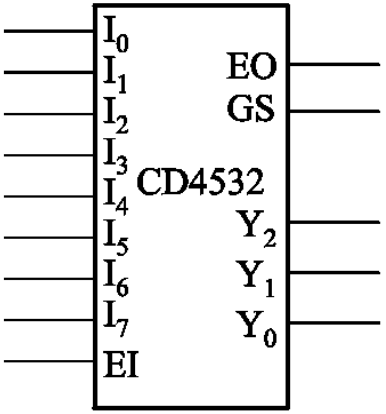
编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	GS	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0
1	0	0	0	0	0	1	×	×	0	1	0	1	0
1	0	0	0	0	0	0	1	×	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0



（2）*EI*使能端（高电平使能）。*EO*、*GS*的作用：*EO*只有在*EI*=1使能，而无编码输入时为“1”，其余情况为“0”，它可以控制相同编码器的*EI*使能端。

CD4532功能表

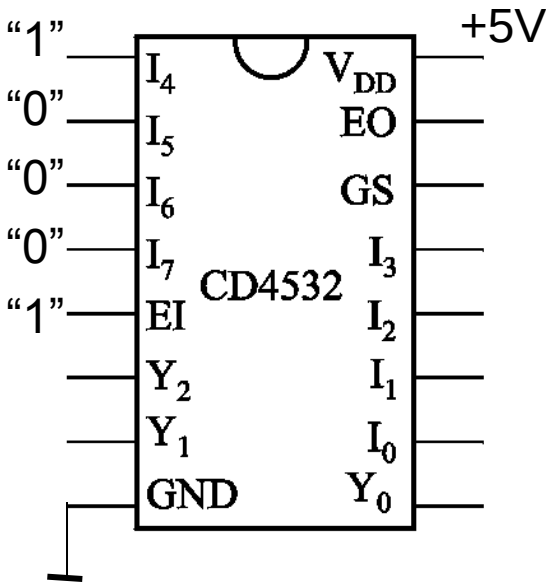
编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	GS	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0
1	0	0	0	0	0	1	×	×	0	1	0	1	0
1	0	0	0	0	0	0	1	×	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0



（3）*GS*端只有在编码器使能情况下，而且编码器有输入时才为“1”，表示编码器处于工作状态。并区别无输入和仅为I_0输入时的三位码000。

CD4532功能表

编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	GS	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0
1	0	0	0	0	0	1	×	×	0	1	0	1	0
1	0	0	0	0	0	0	1	×	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0

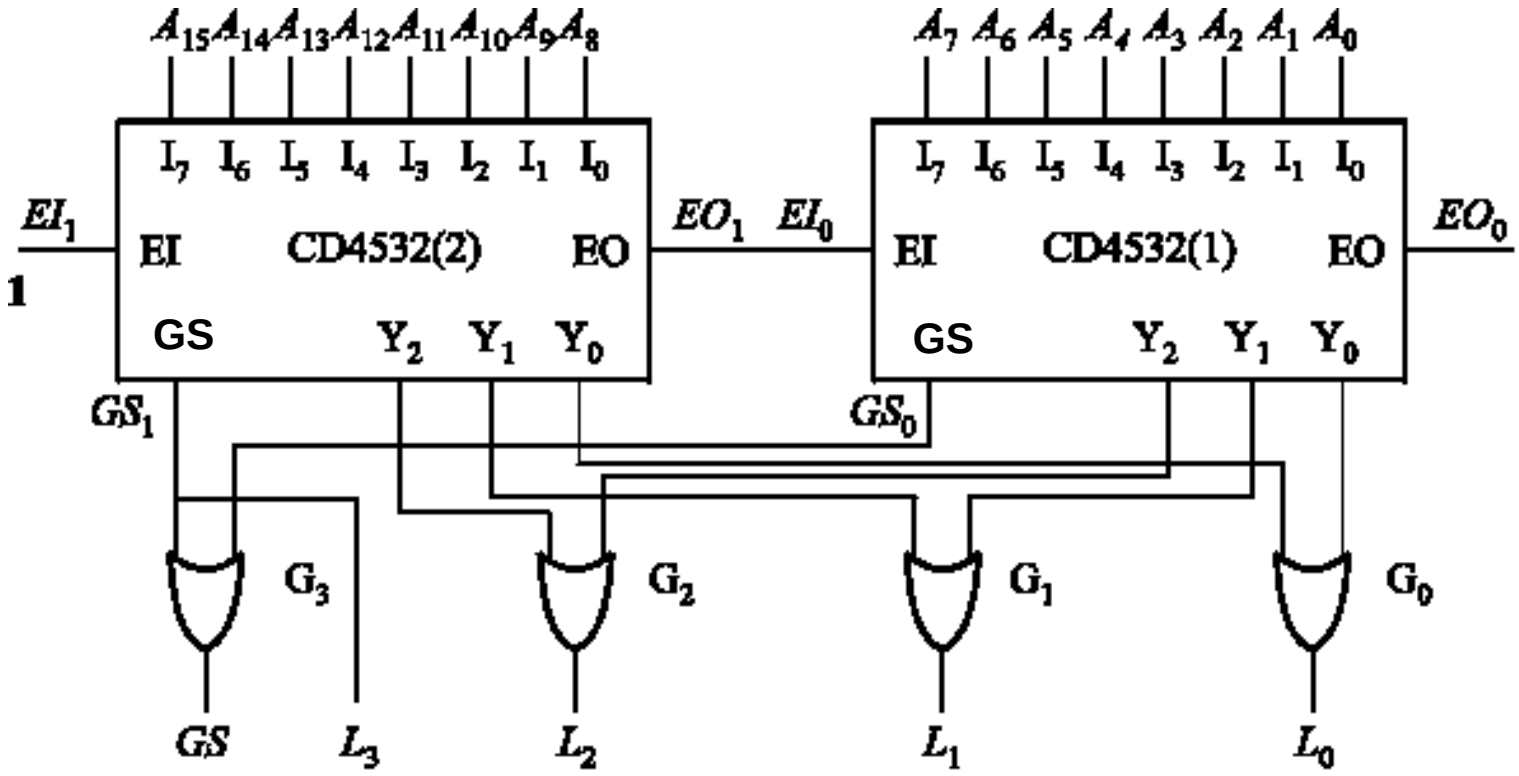


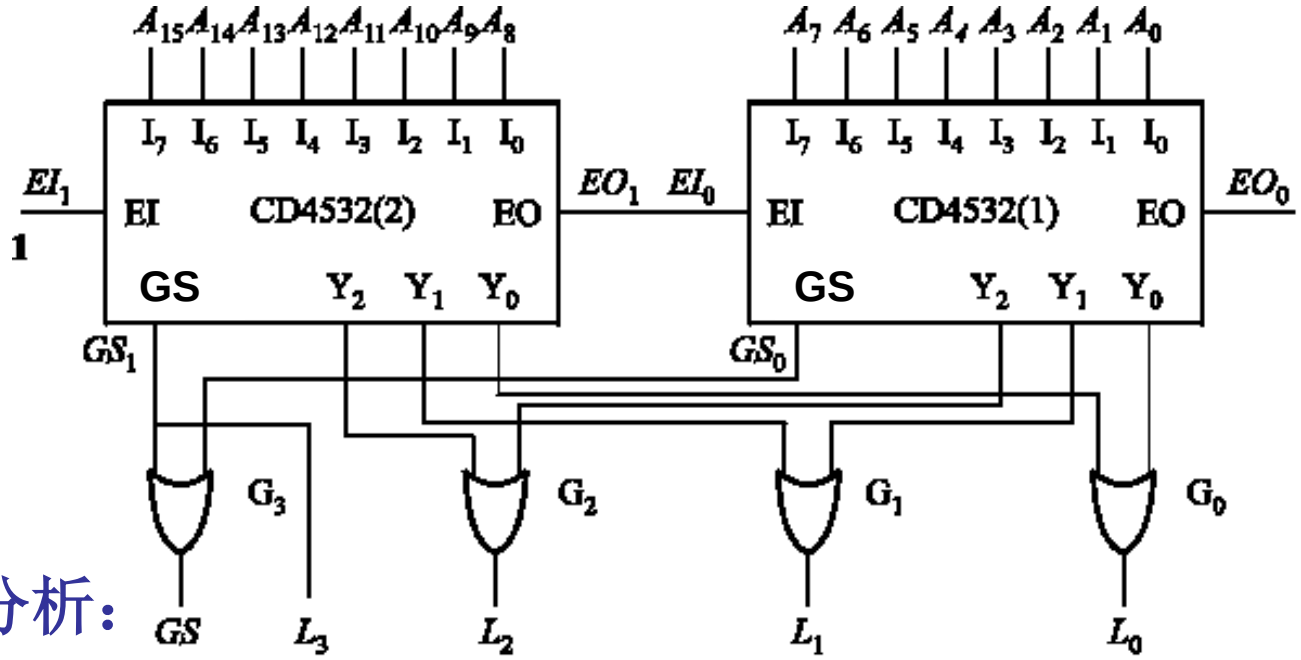
例： 请列出CD4532如图所示连接时，输出三位代码和*EO*、*GS*状态。

解： 此时$Y_2Y_1Y_0$=100，*EO*=“0”，*GS*=“1”。

例2：用两片CD4532扩展成16线/4线的优先编码器。

解：利用使能端*EI*和*EO*、*GS*端将两片连接成分时工作制，输出4位码用或门扩展。

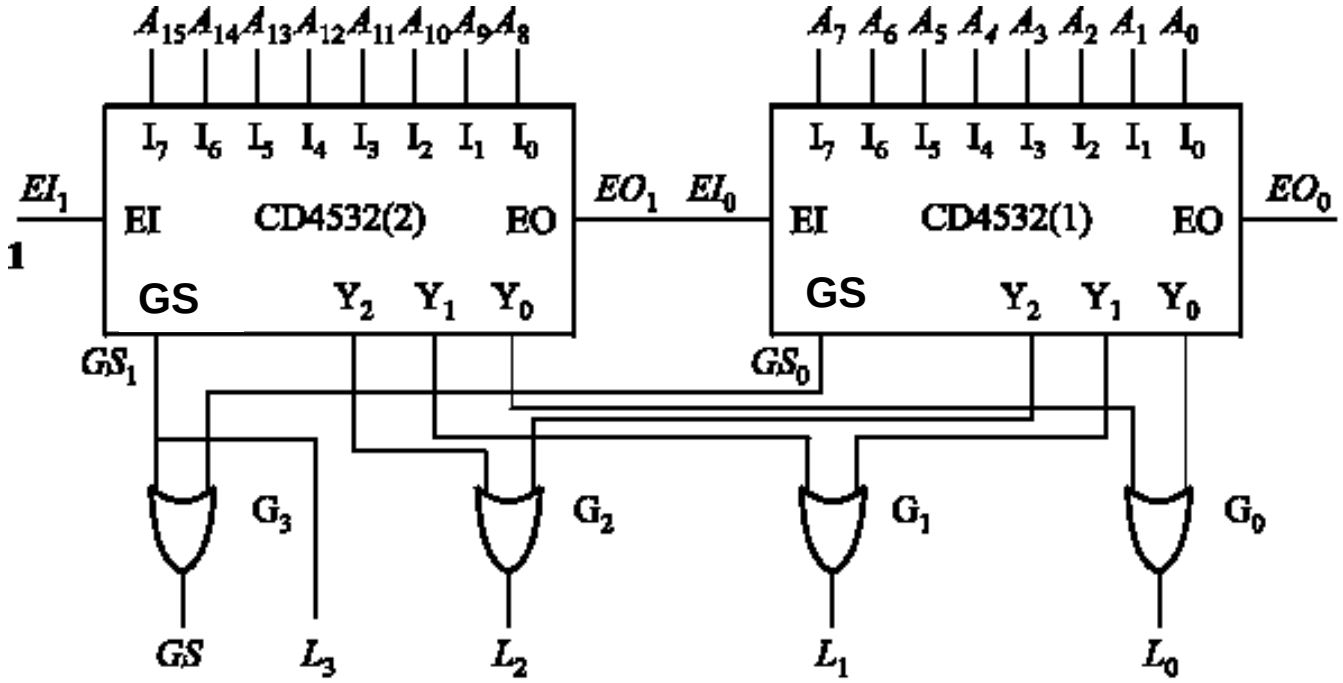




工作原理分析：

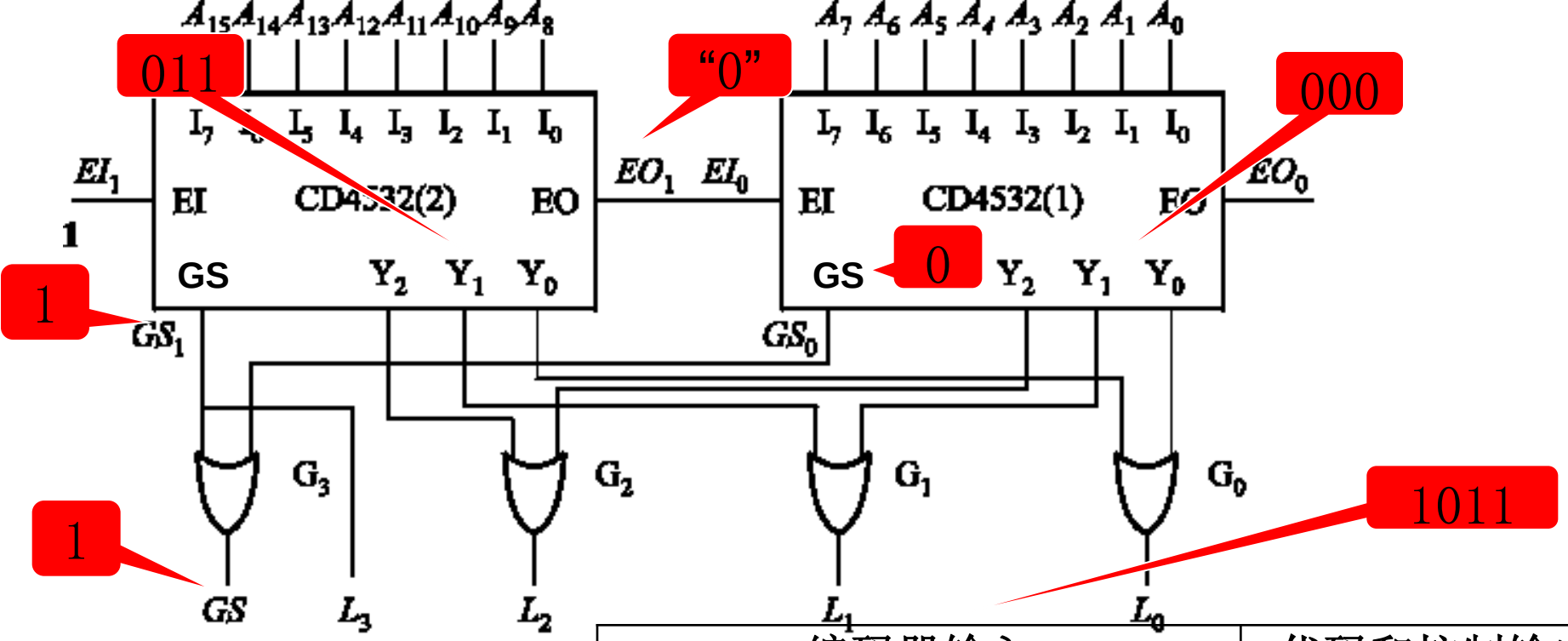
（1）因CD4532（2）的*EI*接“1”，当A_{15}～A_8无输入时，该片的$Y_2Y_1Y_0$=000，*GS*=0，*EO*为“1”， CD4532（1）处于编码状态，输出3位代码由片（1）决定。

编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	*GS*	*EO*
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0



（2）因CD4532（2）的*EI*接“1”，当A_{15}～A_8有输入时，该片的*EO*为“0”， CD4532（1）处于禁止编码状态，输出3位代码为000。由片（2）的位输出代码决定输出。

编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	*GS*	*EO*
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0



若电路输入为 $A_{15}A_{14}\ldots A_8$=00001××, $A_7\ldots A_0$=×××…. × 时, 各处的输出如标记所示。

编码器输入									代码和控制输出				
EI	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0	GS	EO
0	×	×	×	×	×	×	×	×	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	×	×	×	×	×	×	×	1	1	1	1	0
1	0	1	×	×	×	×	×	×	1	1	0	1	0
1	0	0	1	×	×	×	×	×	1	0	1	1	0
1	0	0	0	1	×	×	×	×	1	0	0	1	0
1	0	0	0	0	1	×	×	×	0	1	1	1	0
1	0	0	0	0	0	1	×	×	0	1	0	1	0
1	0	0	0	0	0	0	1	×	0	0	1	1	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0

译码器

译码是编码的反过程。即：将二进制代码所代表的特定对象还原出来的电路。根据还原（翻译）对象的不同，分为二进制译码器和二-十进制译码器（显示译码器）。

一、二进制基本译码器

电路的输入是*n*位二进制代码，输出为2^n 种特定对象。如2/4、3/8、4/16等译码器。

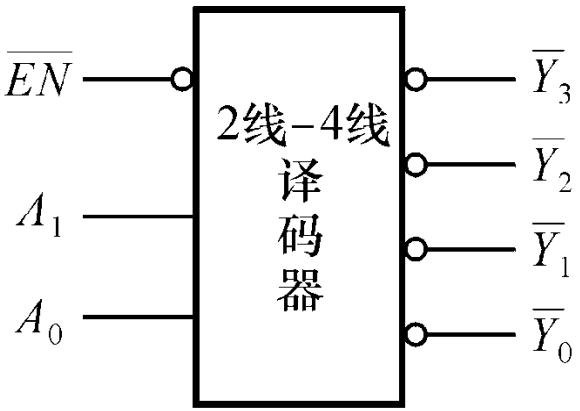
主要是能看懂电路符号和真值表

以2线-4线译码电路为例：

真值表

使能控制	输 入		输 出			
$\overline{EN}$	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

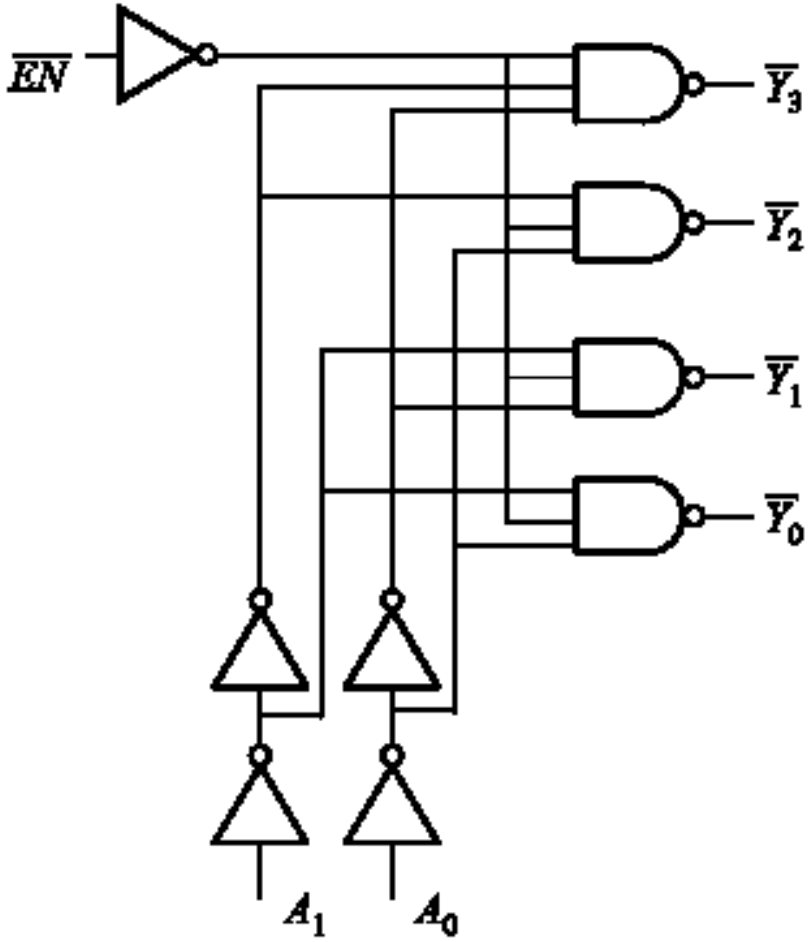
译码器电路符号



由上可知：一组代码和输出对象是一一对应的关系；每一个输出的逻辑函数是一个最小项（这一结论后面有用）。

$$\overline{Y_3} = \overline{A_1 A_0} \qquad \overline{Y_2} = \overline{A_1 \overline{A_0}}$$

$$\overline{Y_1} = \overline{\overline{A_1} A_0} \qquad \overline{Y_0} = \overline{\overline{A_1}\,\overline{A_0}}$$



真值表

使能控制	输 入		输 出			
$\overline{EN}$	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

$\overline{EN}$ 是译码还是不译码的使能控制端。当 $\overline{EN}$ =0时，各输出逻辑函数式为：

$$\overline{Y_3} = \overline{A_1 A_0} \qquad \overline{Y_2} = \overline{A_1}\,\overline{A_0}$$

$$\overline{Y_1} = \overline{A_1 A_0} \qquad \overline{Y_0} = \overline{A_1}\,\overline{A_0}$$

给予使能控制端以及输入变量的各种取值后，得到各输出的结果

2. 中规模集成译码器及应用

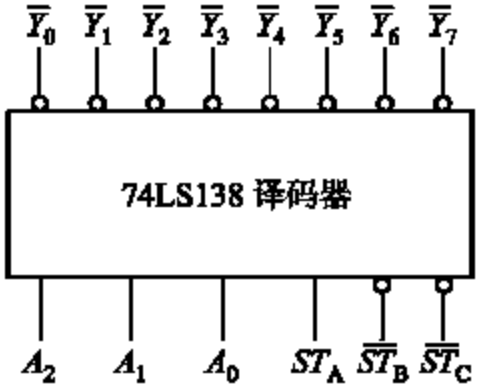
中规模集成译码器种类很多，有二进制译码器，二—十进制BCD码译码器等等，这些译码器连接方便，应用广泛。

（1）中规模集成译码器74HC138

74HC138是一片中规模集成的3线—8线译码器，3位码输入，8个输出，加上电源和译码控制端引线，做成16脚的集成封装。

74HC138译码器真值表

控制与代码输入					译码器输出							
ST_A	$\overline{ST}_B+\overline{ST}_C$	A_2	A_1	A_0	$\overline{Y}_7$	$\overline{Y}_6$	$\overline{Y}_5$	$\overline{Y}_4$	$\overline{Y}_3$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	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



●使能控制端的作用？

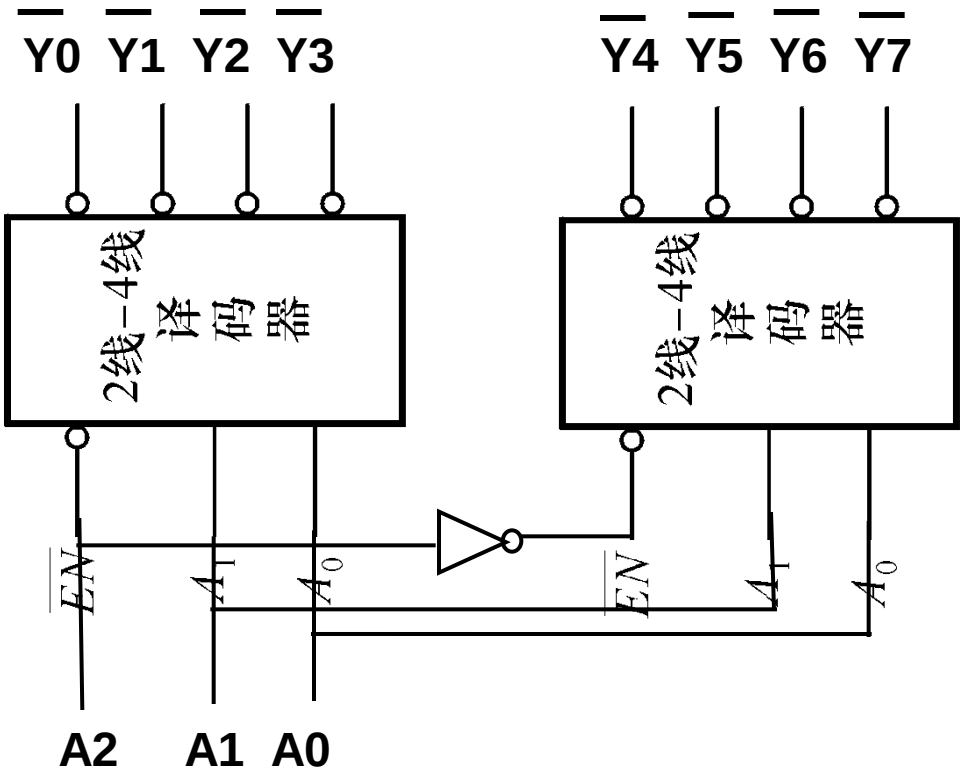
$$ST_A = "1"$$

$$\overline{ST}_B + \overline{ST}_C = "0"$$

译码器使能工作。

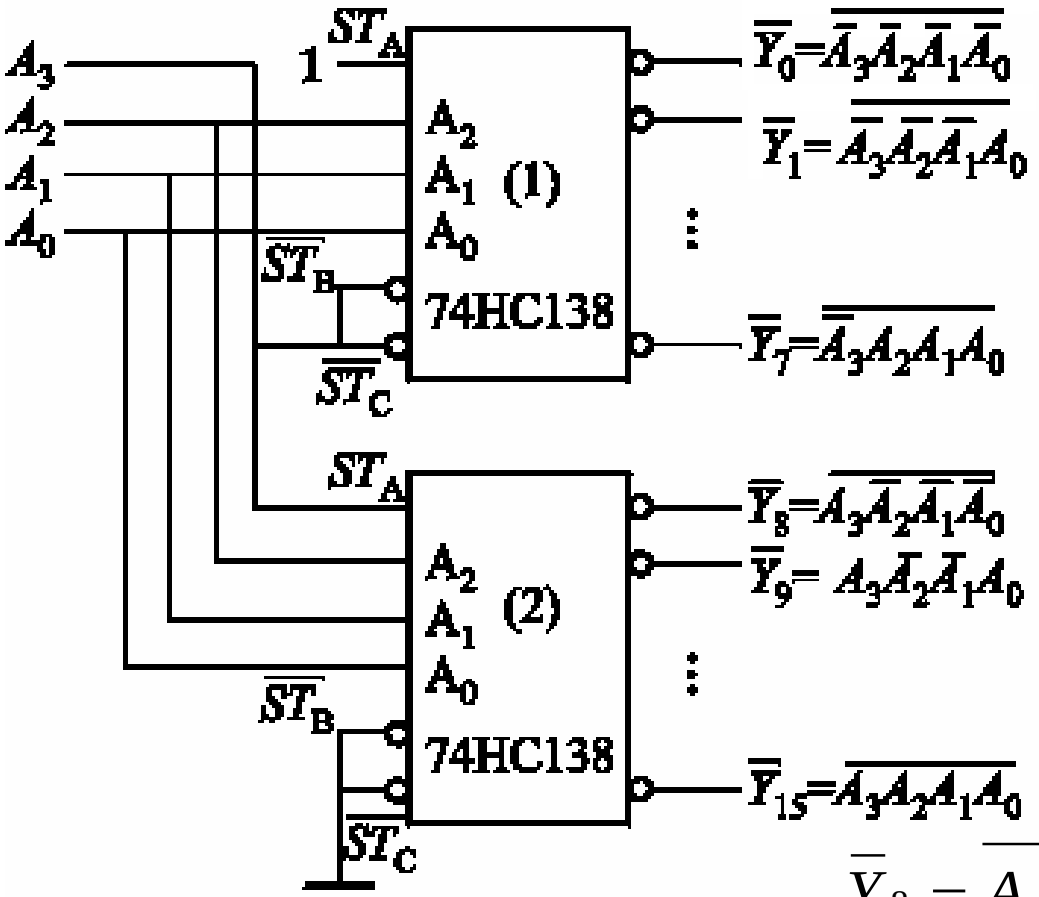
●● 低电平表示有输出（低电平有效），所以输出变量上加了一个非号。

(1) 2/4译码器扩展成3/8译码器



（2）用2片74HC138扩展成4线—16线译码器

利用控制端，采用分时制的工作方式，能方便地实现4线—16线的译码功能。



当$A_3A_2A_1A_0$为0000～0111时，片1使能，片2禁止，输出为Y_0~Y_7；

$$\overline{Y}_0 = \overline{\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0}$$

……

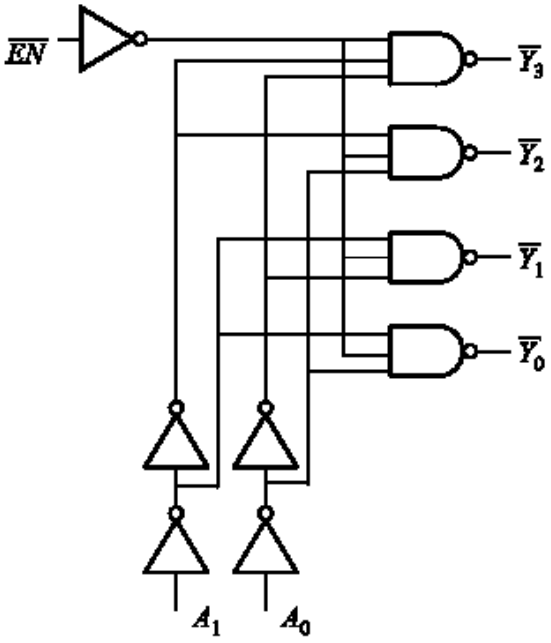
$$\overline{Y}_7 = \overline{\overline{A}_3A_2A_1A_0}$$

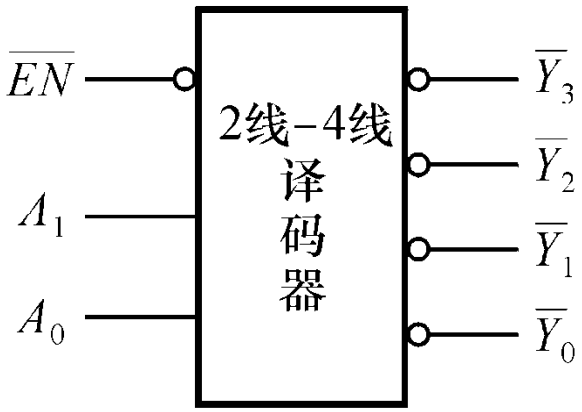
当$A_3A_2A_1A_0$为1000～1111时，片1禁止，片2使能，输出为Y_8~Y_{15}。

$$\overline{Y}_8 = \overline{A_3\overline{A}_2\overline{A}_1\overline{A}_0} \quad \cdots\cdots \quad \overline{Y}_{15} = \overline{A_3A_2A_1A_0}$$

（3）用二进制译码器加适量的门电路，可以组成各种组合电路

由于译码器的每一个输出就是一个最小项，而任何一个输出函数都可以表示为最小项之和表达式。所以，译码器配上适当的逻辑门电路就可实现各种组合电路。





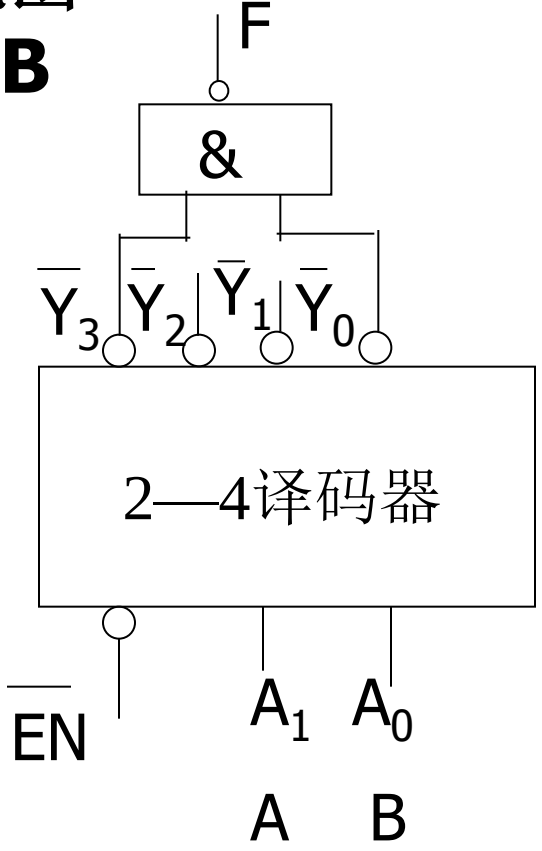
$$\overline{Y}_0 = \overline{\overline{A}_1 \overline{A}_0} = \overline{m}_0$$

$$\overline{Y}_1 = \overline{\overline{A}_1 A_0} = \overline{m}_1$$

$$\overline{Y}_2 = \overline{A_1 \overline{A}_0} = \overline{m}_2$$

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

如果将一逻辑函数的变量有序地加到译码器的输入端，则译码器每一个输出函数对应输入代码的一个最小项，因此，译码器可用以实现组合逻辑电路的设计。如用**2—4**译码器实现函数 $F(A,B)=AB+\overline{A}\overline{B}$ ，则将**A**、**B**分别接到地址$\mathbf{A_1}$、$\mathbf{A_0}$，如图连接。



$$F(A,B)=AB+\overline{A}\overline{B}=m_3+m_0$$

$$=\overline{\overline{Y}_3}+\overline{\overline{Y}_0}=\overline{\overline{Y}_3\cdot\overline{Y}_0}$$

【例】试用3/8译码器设计一个能判别四位二进制码中1的位数是奇数还是偶数的奇偶识别电路。可用“与非”和“与或非”两种门电路。

解：令四位二进制码为$A_3A_2A_1A_0$，输出Y_{OD}表示1的位数为奇数，Y_E为偶数。则卡诺图为：

A_3A_2 \ A_1A_0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

图中1方格表示奇数，0格为偶数。所以结合1格得到Y_{OD}函数，结合0格得Y_E函数。

A_3A_2 \ A_1A_0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

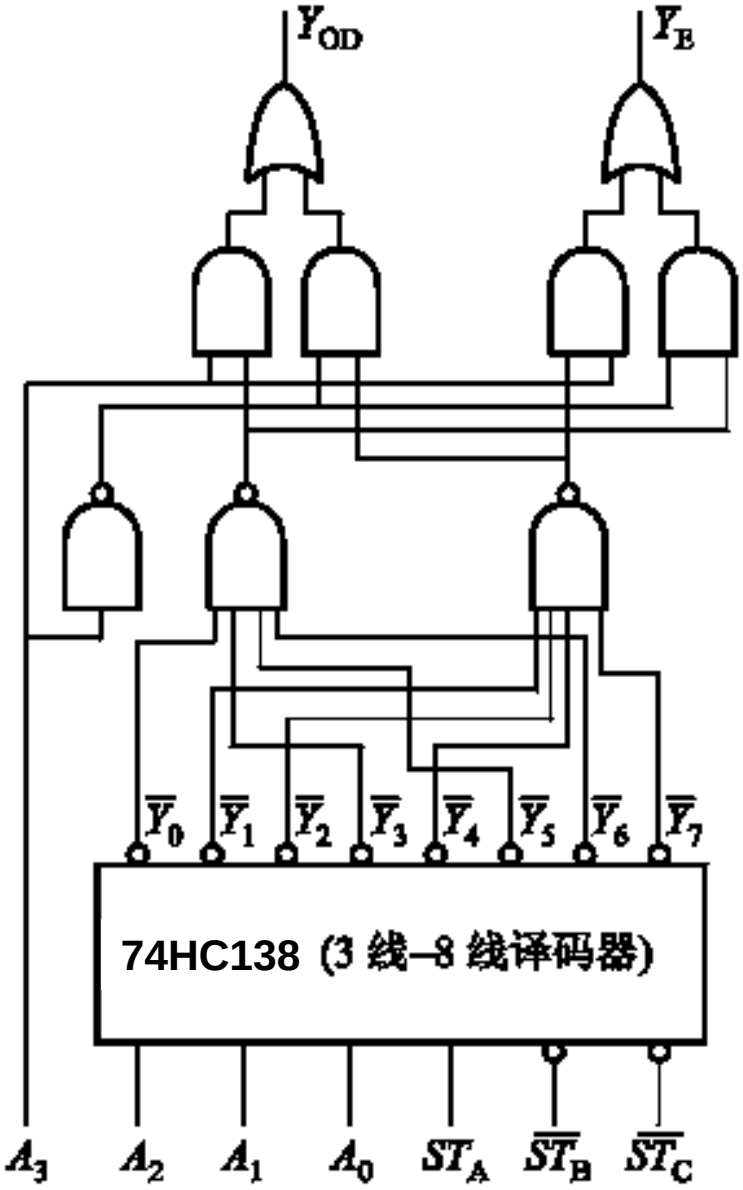
$$Y_{OD} = \overline{A_3}(\overline{A_2}\,\overline{A_1}\,\overline{A_0} + \overline{A_2}A_1\overline{A_0} + A_2\overline{A_1}\,\overline{A_0} + A_2A_1A_0)$$
$$+ A_3(\overline{A_2}\,\overline{A_1}\,\overline{A_0} + \overline{A_2}A_1A_0 + A_2\overline{A_1}A_0 + A_2A_1\overline{A_0})$$
$$= \overline{A_3}(\overline{Y_1} + \overline{Y_2} + \overline{Y_4} + \overline{Y_7}) + A_3(\overline{Y_0} + \overline{Y_3} + \overline{Y_5} + \overline{Y_6})$$
$$= \overline{A_3}(\overline{Y_1}\,\overline{Y_2}\,\overline{Y_4}\,\overline{Y_7}) + A_3(\overline{Y_0}\,\overline{Y_3}\,\overline{Y_5}\,\overline{Y_6})$$

$$Y_E = \overline{A_3}(\overline{A_2}\,\overline{A_1}\,\overline{A_0} + \overline{A_2}A_1A_0 + A_2\overline{A_1}A_0 + A_2A_1\overline{A_0})$$
$$+ A_3(\overline{A_2}\,\overline{A_1}A_0 + \overline{A_2}A_1\overline{A_0} + A_2\overline{A_1}\,\overline{A_0} + A_2A_1A_0)$$
$$= A_3(\overline{Y_1}\,\overline{Y_2}\,\overline{Y_4}\,\overline{Y_7}) + \overline{A_3}(\overline{Y_0}\,\overline{Y_3}\,\overline{Y_5}\,\overline{Y_6})$$

连接后的电路

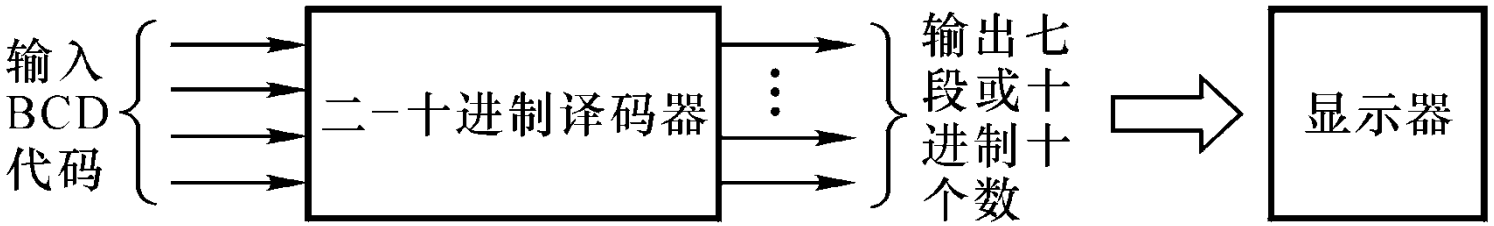
$$Y_{OD} = \overline{A_3}(\overline{Y_1 Y_2 Y_4 Y_7}) + A_3(\overline{Y_0 Y_3 Y_5 Y_6})$$

$$Y_E = A_3(\overline{Y_1 Y_2 Y_4 Y_7}) + \overline{A_3}(\overline{Y_0 Y_3 Y_5 Y_6})$$



3. 二-十进制译码器

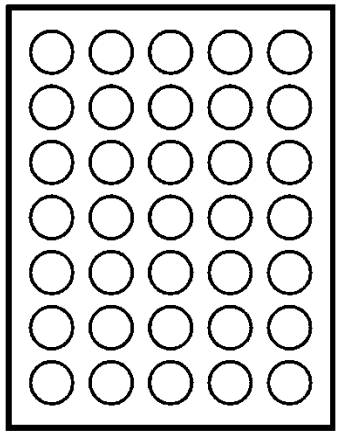
将输入BCD码翻译成十进制数码的组合逻辑电路，所以，又称显示译码器（译码后的结果能用显示器显示出来）、码制变换译码器多种。



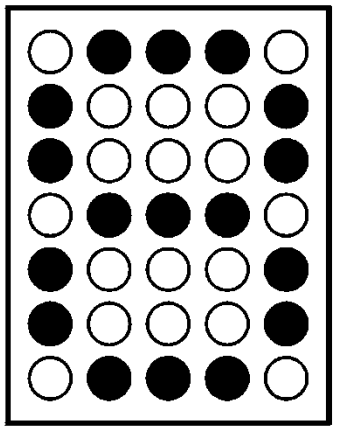
➢显示器简介

显示器分为点阵式和分段式两种（也可按器件分为半导体和液晶两类）。

点阵式显示器



一般由发光二极管等矩阵组成，常用于广告、车站等场合。



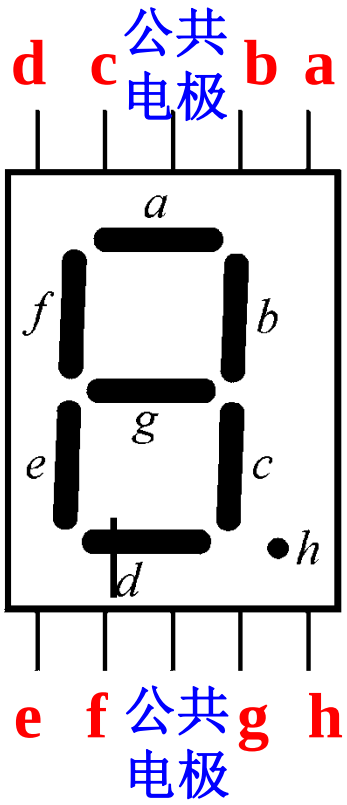
要点亮某一个字形时，只要点亮这些字形的点就行。

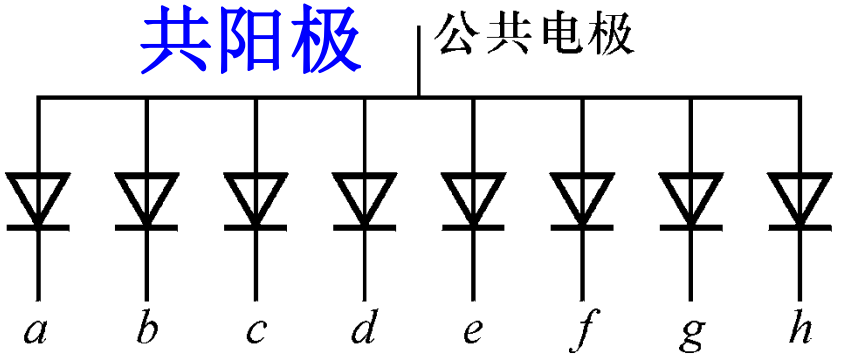
分段显示器

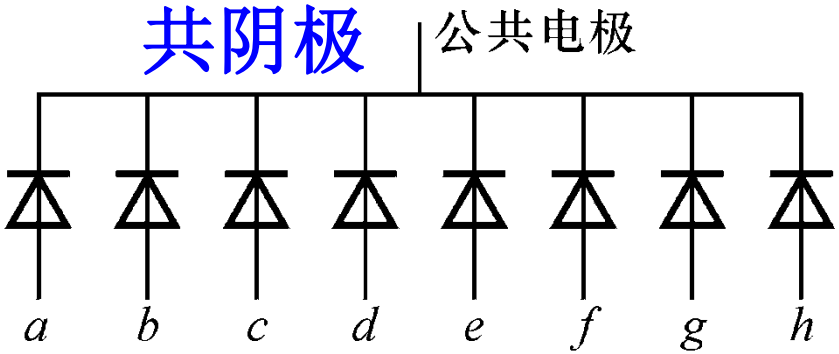
它可以是半导体分段式（LED数码管）和液晶分段式两种。

LED显示器

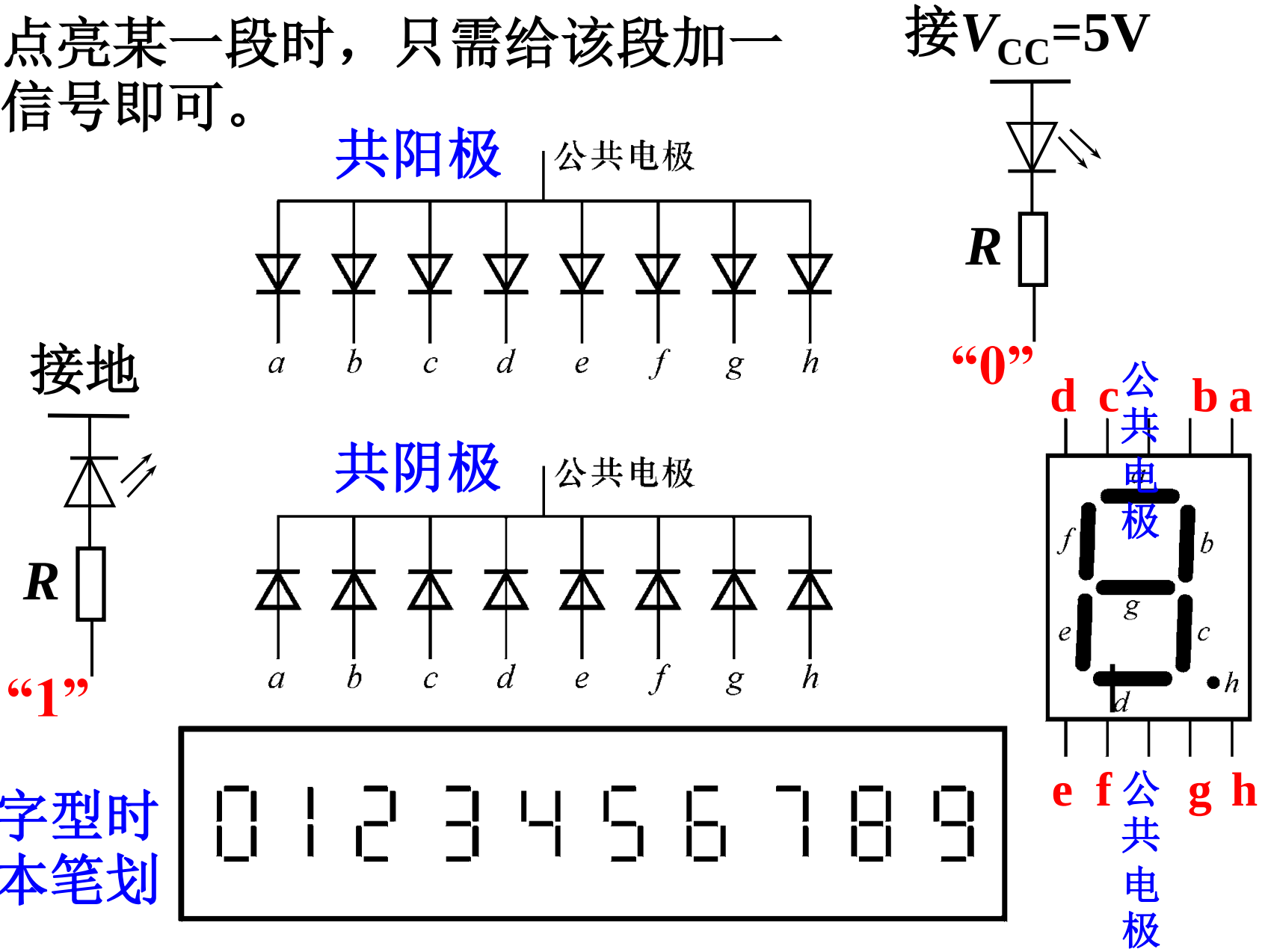
它由7段（不含小数点）、8段（含小数点）两种，每一段就是一只发光二极管。又分为共阳极和共阴极结构。







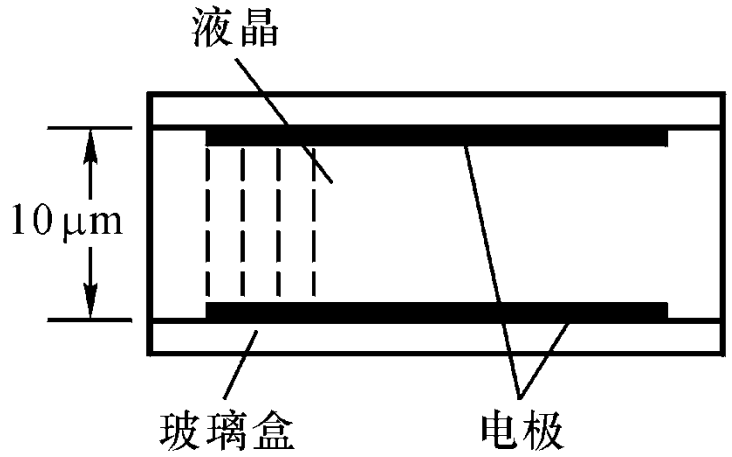
要点亮某一段时，只需给该段加一个信号即可。



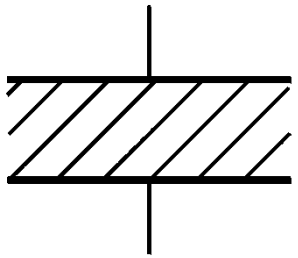
分段式液晶显示器（LCD）

一种具有流动性的有机化合物的奇特光学特性而发光

- 目前使用日益普遍
- 功耗极微
- 工作电压低（2~5V）
- 显示清晰、体积小、寿命长
- 缺点是响应速度慢（10~200ms）



结构

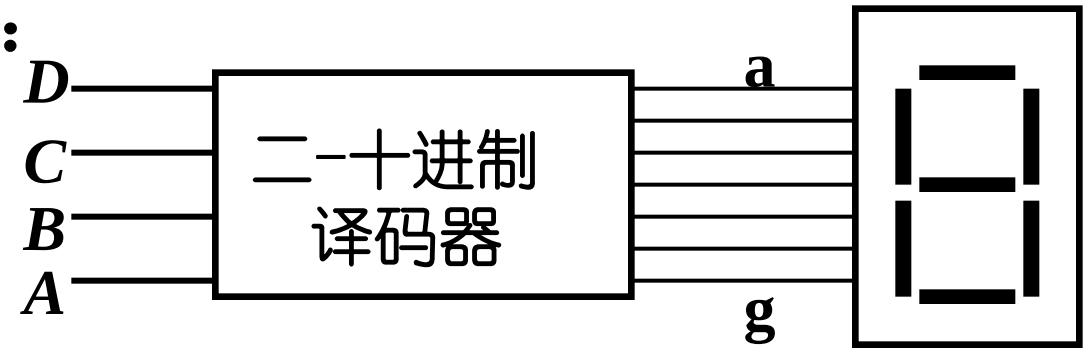


电路符号

二-十进制译码器设计举例

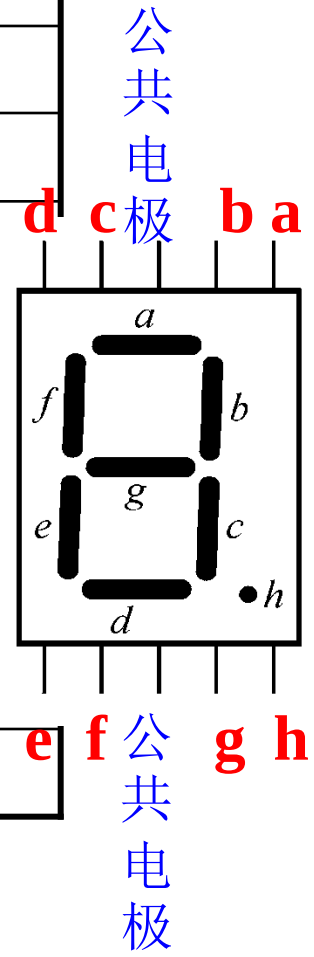
【例2.3.1】试用非门和或非门设计一个8421BCD码输入的驱动七段半导体数码管（共阴极）的二-十进制译码器。

解：由于显示器为七段半导体数码管，所以译码器的输出为七个输出，四位BCD 码输入，设计电路如下框图所示：



由于LED是共阴极，所以译码器输出应为高电平才能点亮某段数码管。列出真值表：

输入8421代码				输出对应段亮暗							字形
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



用卡诺图化简，得出七段的逻辑函数式，由于用或非门且驱动共阴极，所以用包围“0”格，求或与式的最简原函数。以a段为例：

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

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

a	BA 00	01	11	10
DC 00	1	0	1	1
01	0	1	1	1
11	×	×	×	×
10	1	1	×	×

同理可得其它各段的函数式。

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

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

$$c = \overline{\overline{C + \overline{B} + A}}$$

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

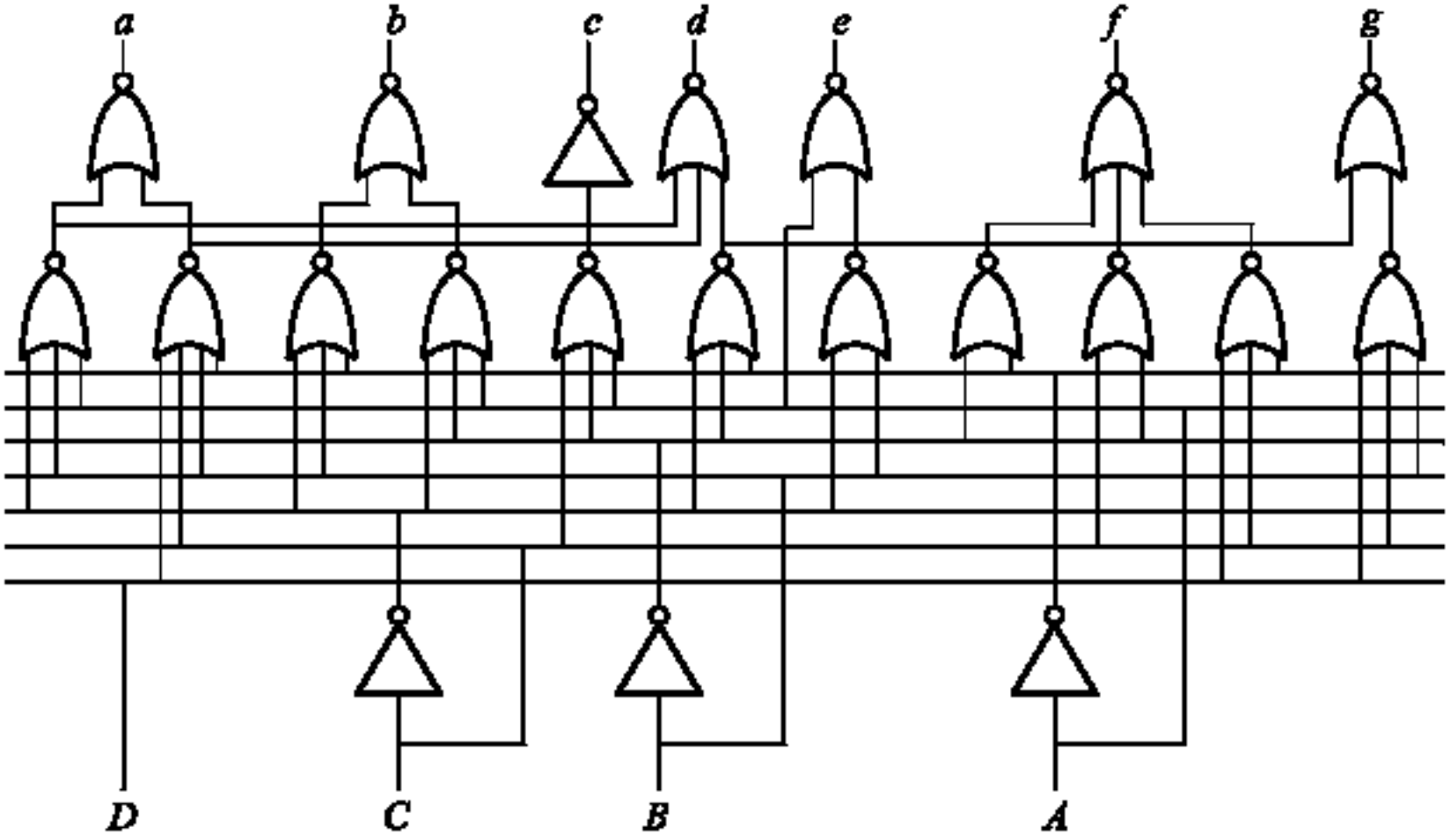
$$= \overline{\overline{\overline{C} + B + A} + \overline{\overline{C} + \overline{B} + \overline{A}} + \overline{D + C + B + \overline{A}}}$$

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

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

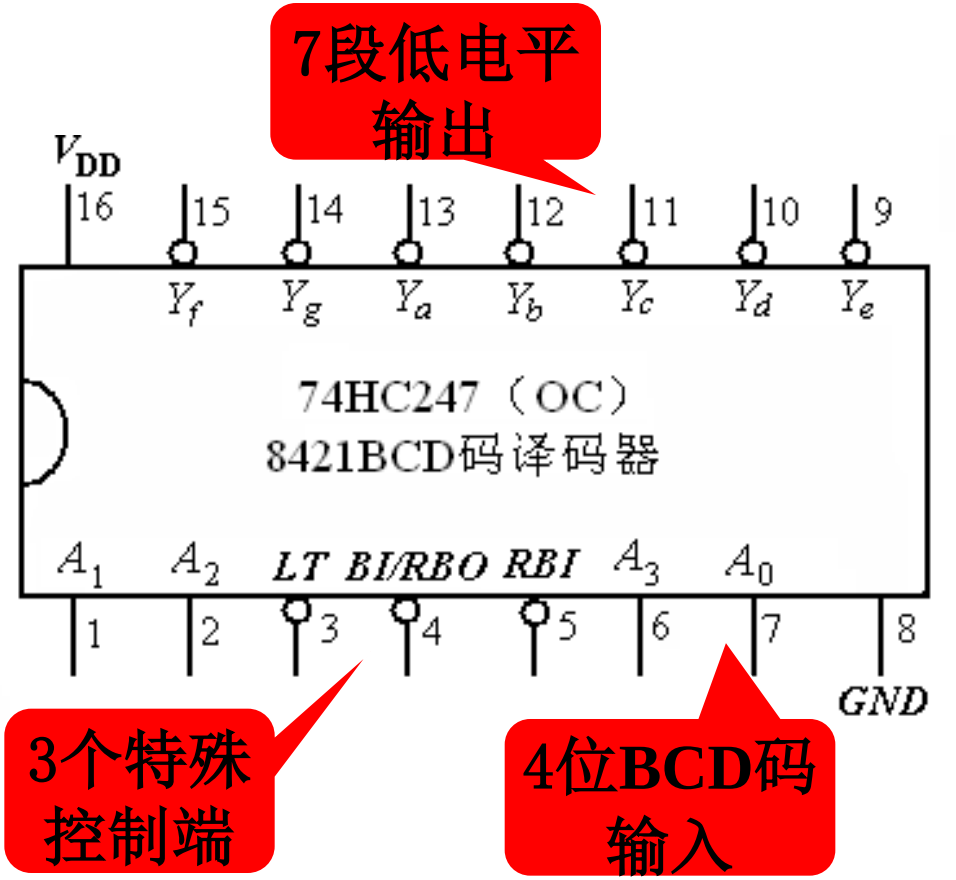
$$g = (D + C + B)(\overline{C} + \overline{B} + \overline{A}) = \overline{\overline{D + C + B} + \overline{\overline{C} + \overline{B} + \overline{A}}}$$

从逻辑式可以计算出需要或非门的个数，画出该设计电路如图所示：

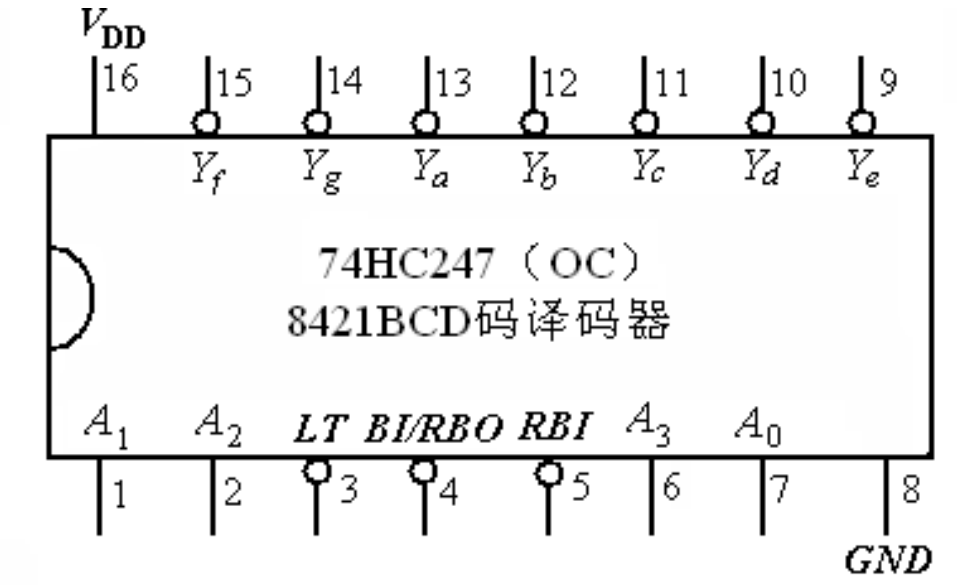


（4）二—十进制译码器和数码管的连接

74HC247（OC）二—十进制译码器

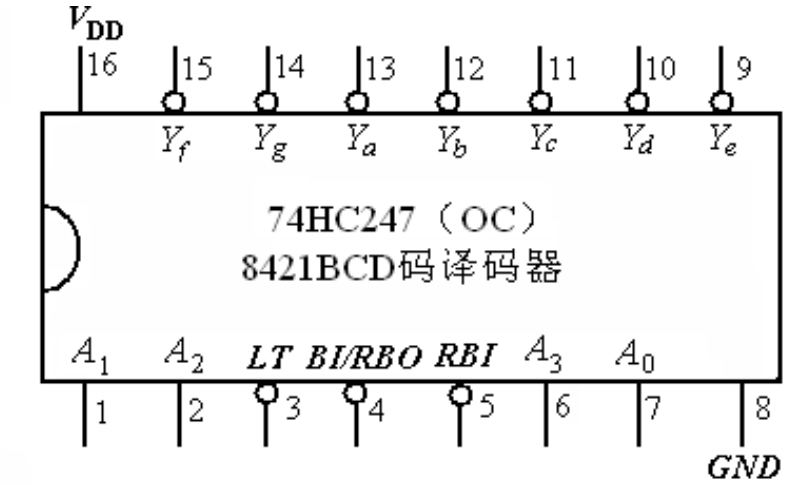


A_3～A_0是四位8421BCD码输入端，Y_a～Y_g驱动七段数码管的七个输出，低电平输出，集电极开路门结构，适用于七段共阳极数码管。



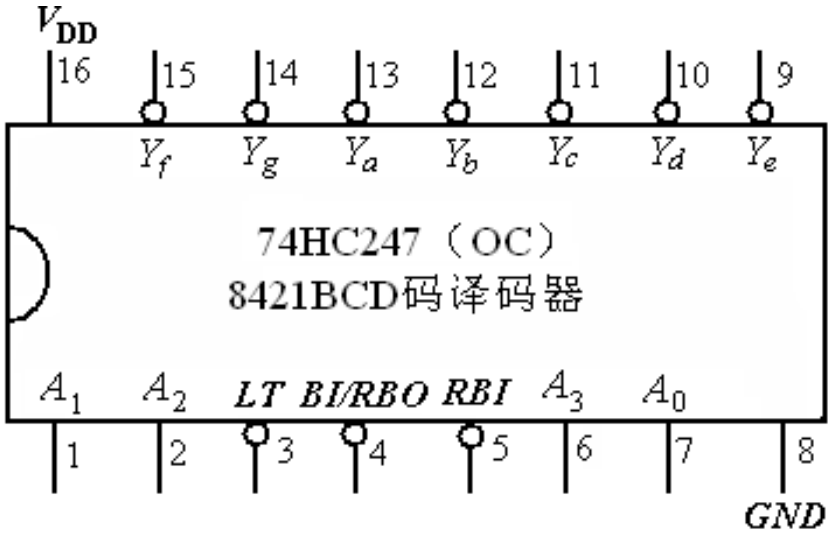
① *LT*灯测试控制：

低电平有效，*LT*=0时，不管A_3～A_0状态如何，Y_a～Y_g输出都为低电平，共阳极七段数码都能点亮，不测试时，*LT*置“1”。



② *BI/RBO*灭灯输入/灭零输出控制端

***BI*=0为灭灯输入，不管A_3～A_0状态如何，Ya～Yg输出都为高电平，把共阳的七段数码管熄灭。作为灭零输出端用时，可以作为下一位的灭零输入。正常时应置高电平。**

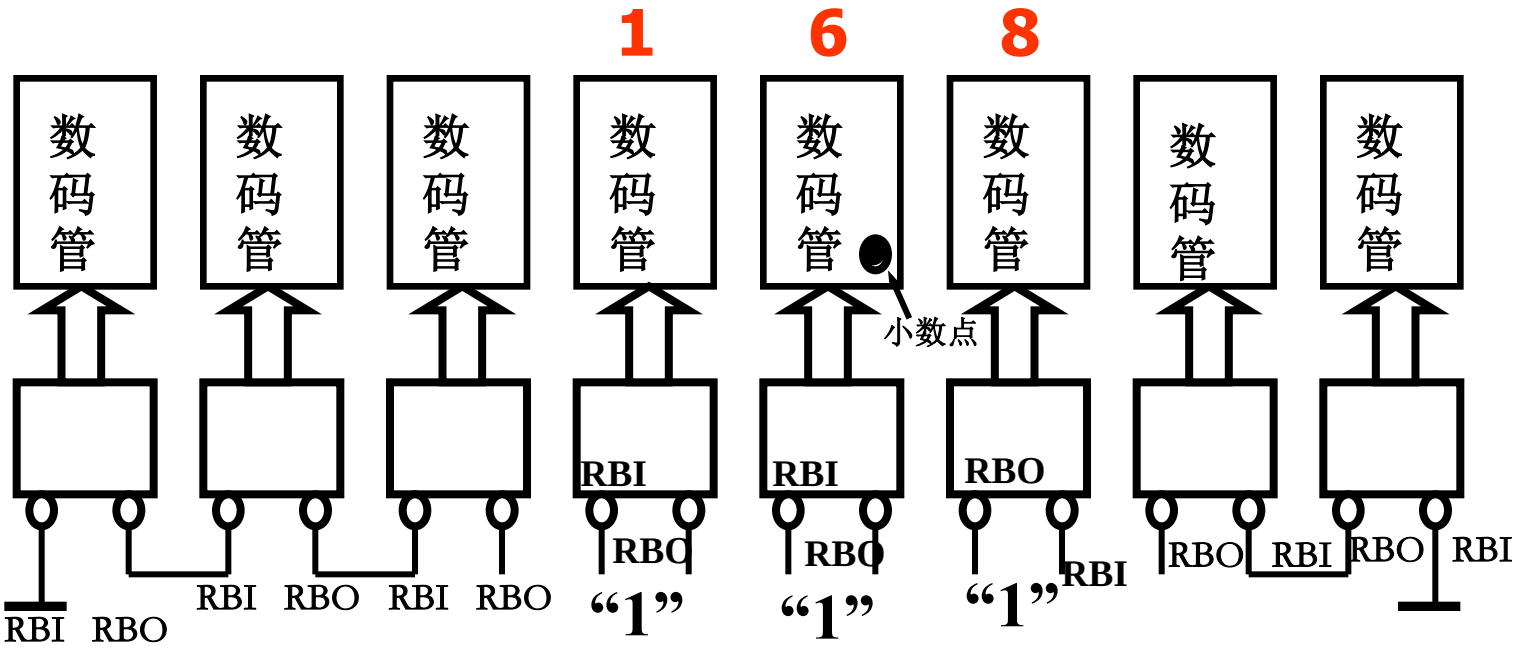


② *BI/RBO*灭灯输入/灭零输出控制端

③ *RBI*灭零输入端

它与***RBO***配合使用，熄灭不必显示的零。如有一个8位数码显示电路，整数部分5位，小数部分3位，在显示16.8这个数字时将呈现00016.800字样，如将前后多余零灭掉，则更加明了醒目。

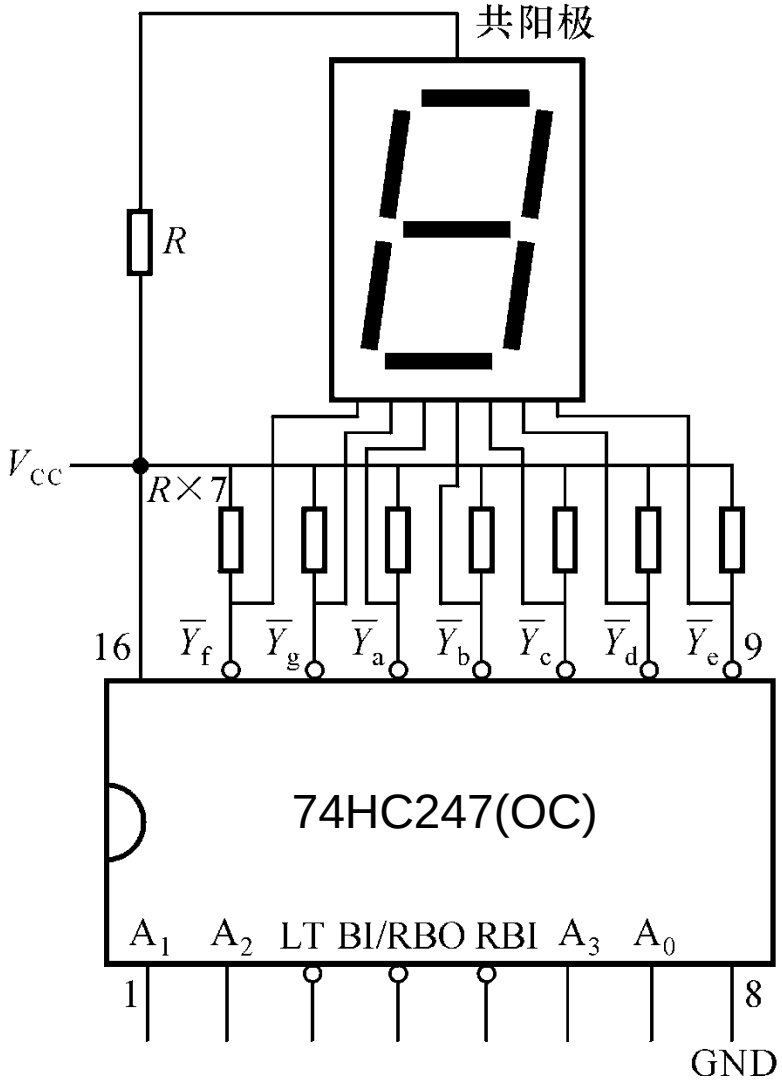
灭零输入RBI和灭零输出RBO连接图



③ *RBI*灭零输入端

② *BI/RBO*灭灯输入/灭零输出控制端

74HC247（OC）与共阳极数码管的具体连接



“1”

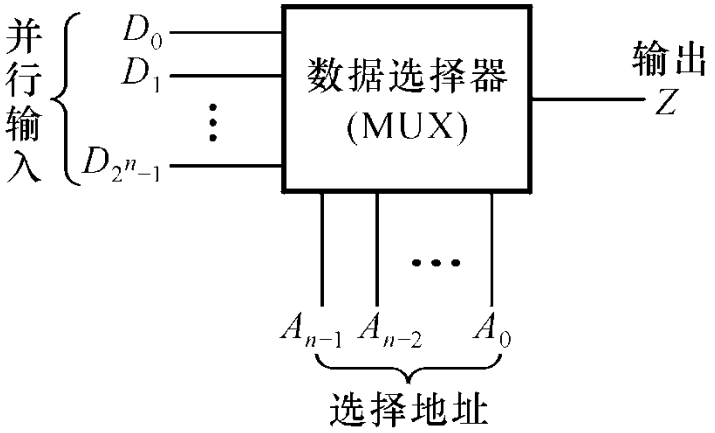
二、数据选择器和数据分配器

数据选择器

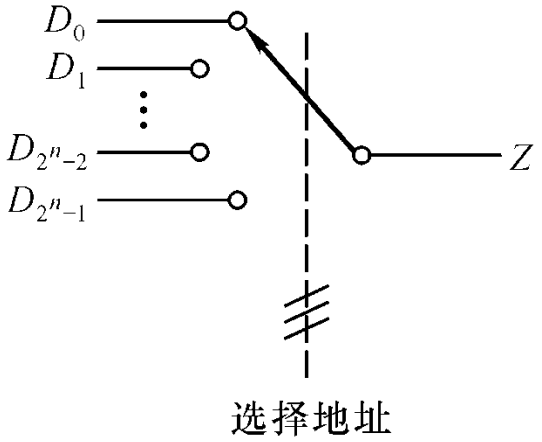
数据分配器和数据选择器大量应用在数据采集和数字信号处理与通信系统中。

1、数据选择器

在数字信号的传输过程中，有时需要从一组输入数据中选出某一个来，或在多路数据采集系统中，选出某一路来。能实现这一功能的电路就是多路数据选择器。



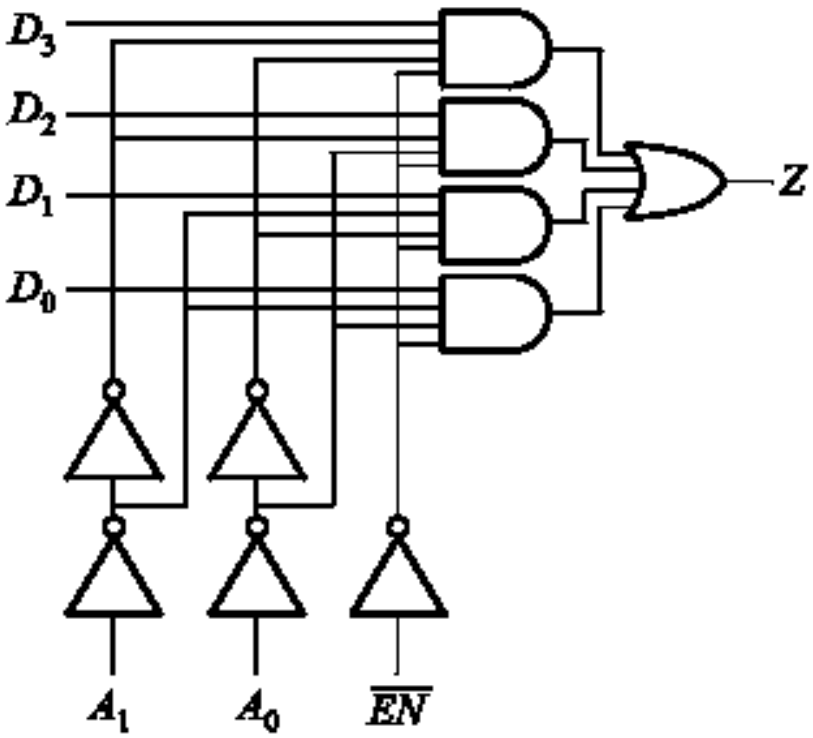
原理图



模拟图

从数据的传输方式讲，它是一个并行/串行的传输转换电路。

一个四选一(4/1)数据选择器如图所示：



真值表

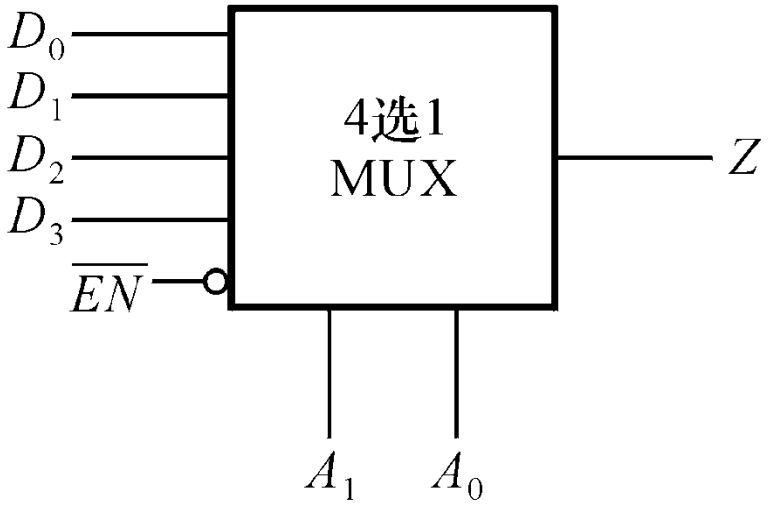
使能	输 入		输出
$\overline{EN}$	A_1	A_0	Z
1	×	×	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

在使能控制端 $\overline{EN}=0$ 时，其输出函数为：

$$Z = D_3 A_1 A_0 + D_2 A_1 \overline{A_0} + D_1 \overline{A_1} A_0 + D_0 \overline{A_1}\,\overline{A_0}$$

$$= \sum_{i=0}^{2^n-1} D_i m_i$$

4选1数据选择器电路符号

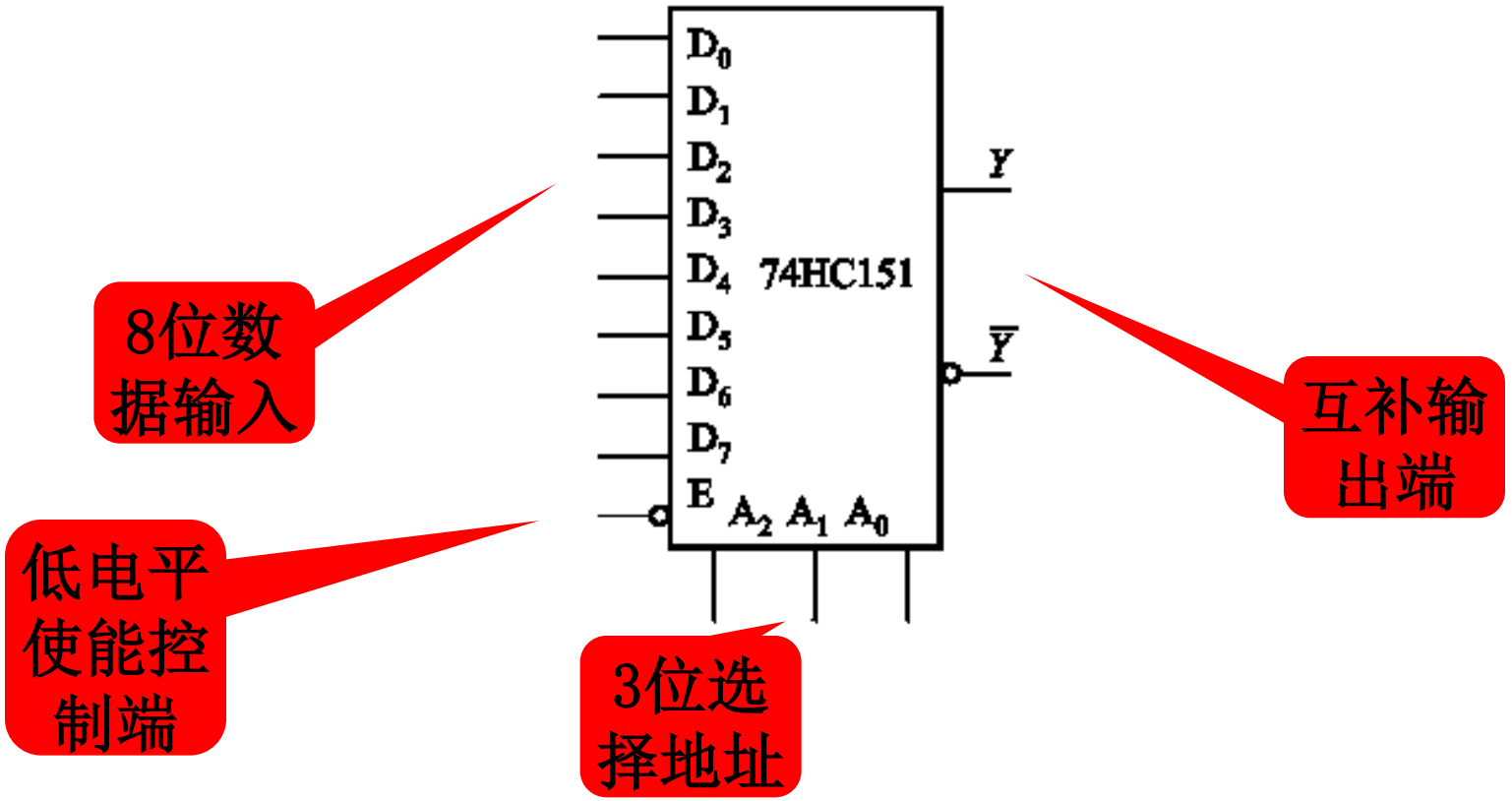


中规模集成数据选择器和数据分配器

1. 数据选择器

74HC151数据选择器

8选1的数据选择器（多路选择器MUX）



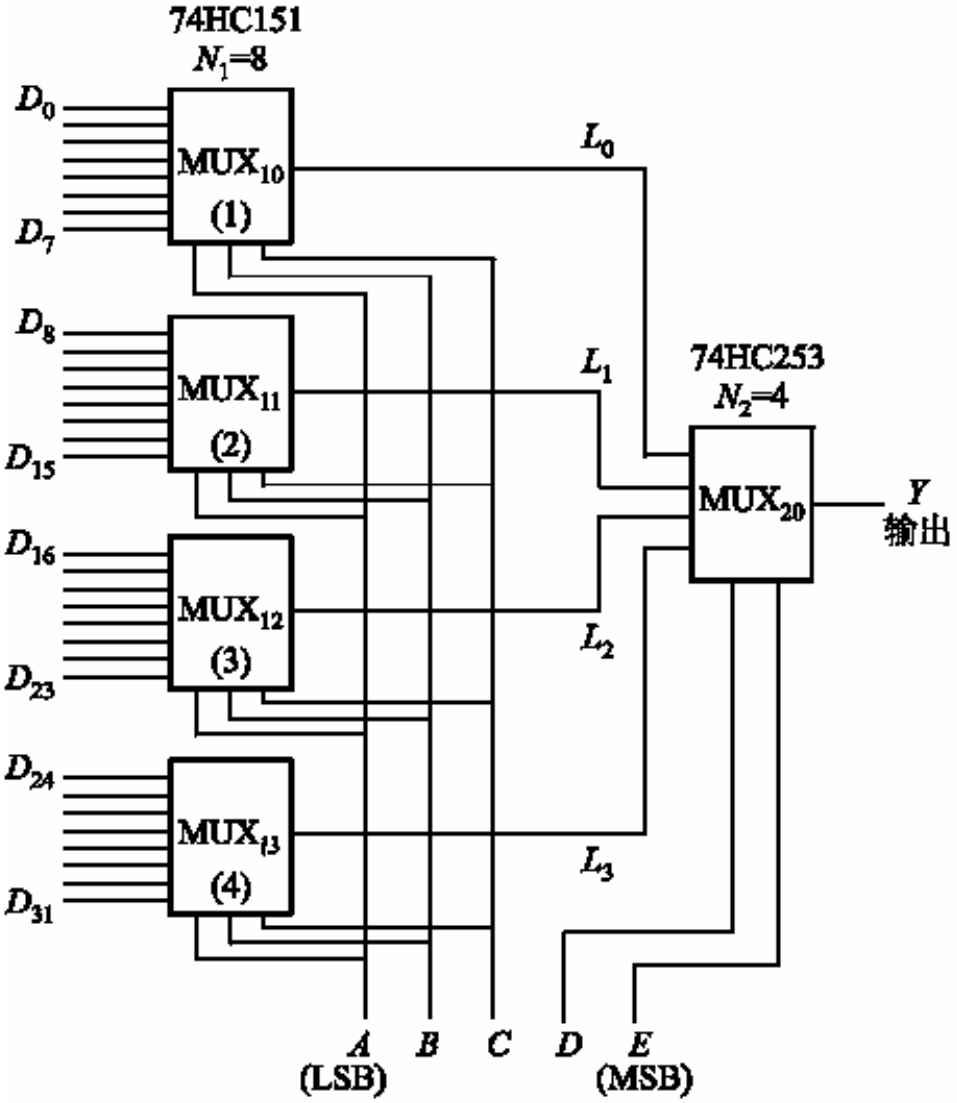
（2）扩大数据选择范围

EDCBA
=00000
～00111

EDCBA
=01000
～01111

EDCBA
=10000
～10111

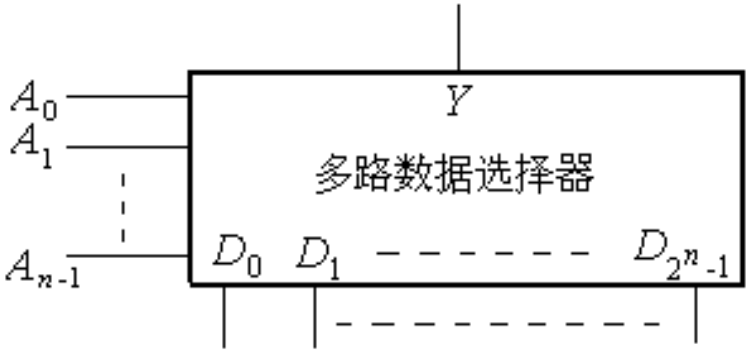
EDCBA
=11000
～11111



四片8选1数据选择器(74HC251)和一片4选1数据选择器(74HC253)构成一个32选1的数据选择器。

（3）实现各种组合型逻辑函数

在选择器使能条件下，选择器的输出函数为：



$$Y = D_{n-1}m_{n-1} + D_{n-2}m_{n-2} + \cdots + D_1m_1 + D_0m_0$$

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

从电路的输出函数可知，数据选择器是一个与-或表达式，而电路的结构又是一个与或逻辑结构。因此，用数据选择器可以产生各种各样的组合逻辑电路。

用数据选择器实现逻辑函数的方法：

①把函数的输入变量分为两组，一组加到数据选择器的地址端，余下的一组变量送到数据选择器的数据输入端。

②求出加到每个数据输入端的值。

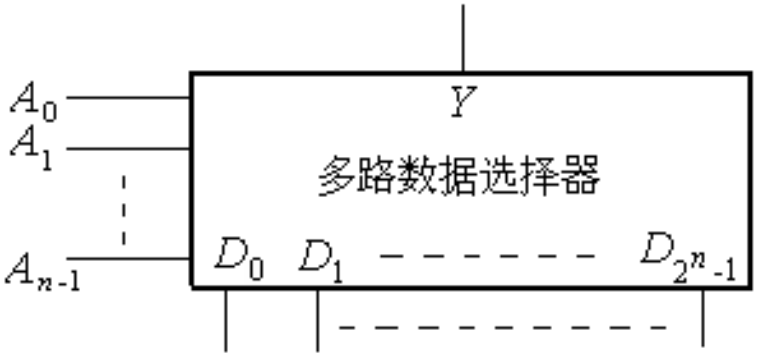
③画出要实现的逻辑函数的逻辑图。

具体设计方法分三种情况说明：

①采用具有*n*个地址端的数据选择器实现*n*变量的函数时，应将函数的输入变量加到地址端（*A*），将函数卡诺图各方格内的值接到相应的数据输入端（*D*）。

②当函数输入变量数小于数据选择器的地址端时，应将不用的地址端及不用的数据输入端都接0（或接1）。

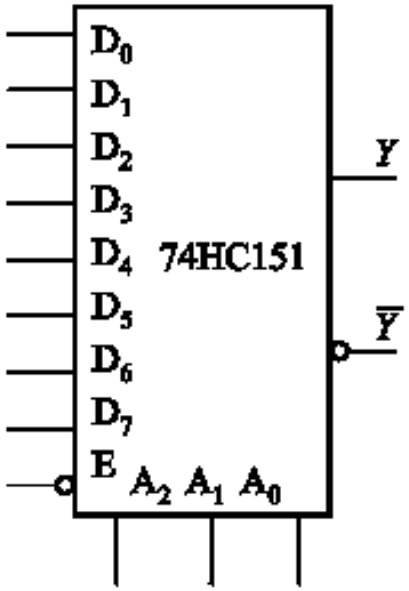
③当函数输入变量大于数据选择器地址端时，可任选几个变量接到地址端，剩下的变量以一定的方式接到数据端。



【例1】用8选项数据选择器74HC151实现以下三变量函数。

$$Z = f(A,B,C) = \overline{A}\,\overline{B} + \overline{B}C + A\overline{B}\overline{C}$$

解：令函数的3个变量都作选择器地址输入，然后将函数配成最小项之和形式。



$$Z = f(A,B,C) = \overline{A}\overline{B} + \overline{B}C + AB\overline{C}$$

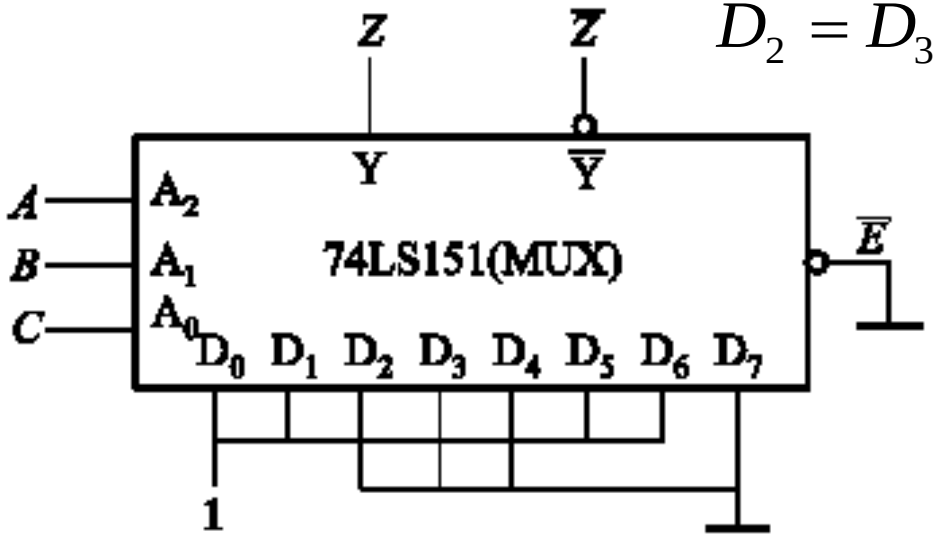
$$= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}C + A\overline{B}C + AB\overline{C}$$

$$= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}C + AB\overline{C} = m_0D_0 + m_1D_1 + m_5D_5 + m_6D_6$$

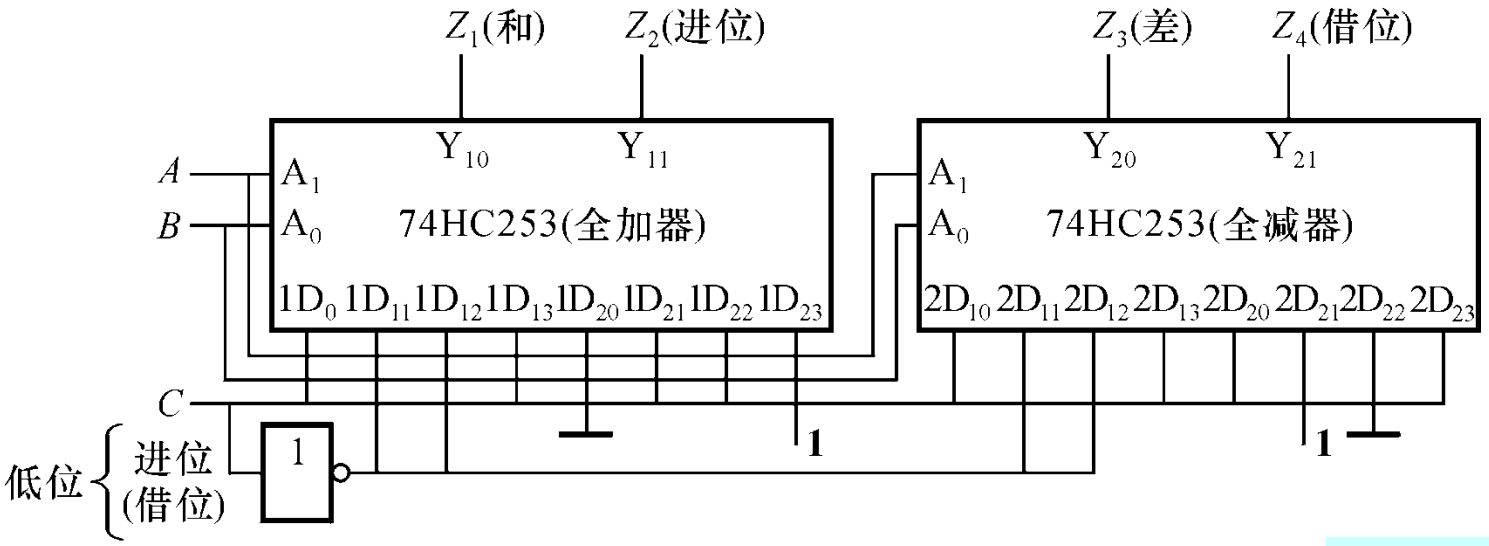
$$= m_0.1 + m_1.1 + m_2.0 + m_3.0 + m_4.0 + m_5.1 + m_6.1 + m_7.0$$

所以有 $D_0 = D_1 = D_5 = D_6 = 1$

$D_2 = D_3 = D_4 = D_7 = 0$



【例2】试写出图示电路输出函数式，并说明电路的逻辑功能是实现全加器和全减器功能。



解：

$$Z_1 = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC = A \oplus B \oplus C$$

全加器和

$$Z_2 = \overline{A}\,\overline{B} \cdot 0 + \overline{A}BC + A\overline{B}C + AB = (A \oplus B)C + AB$$

全加器进位

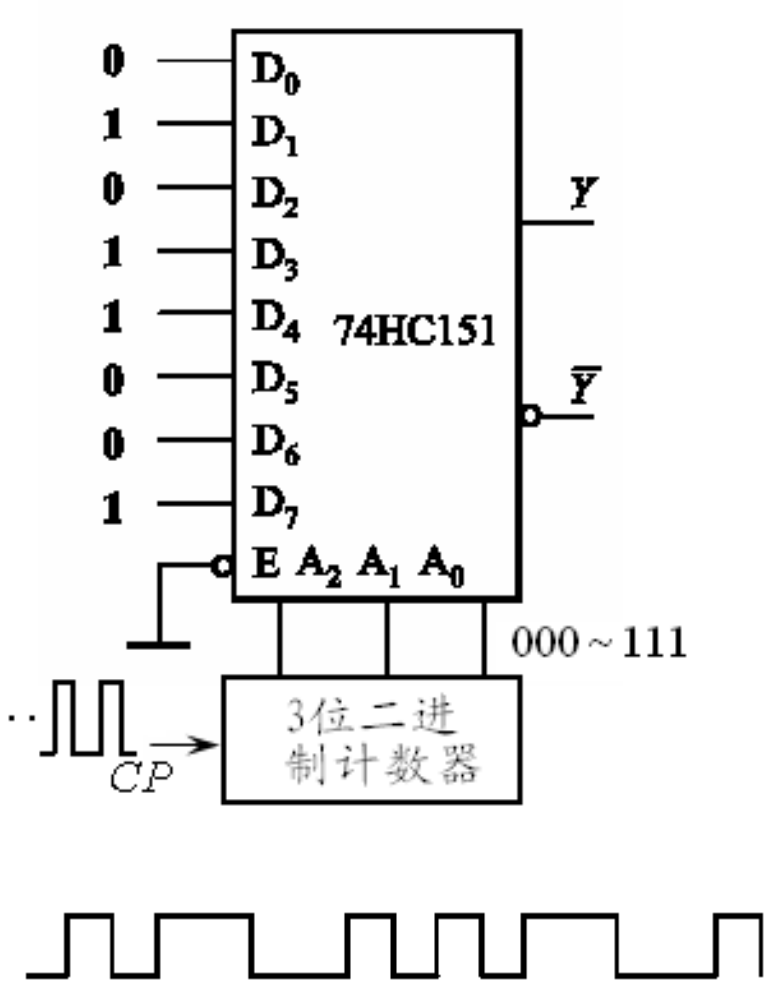
$$Z_3 = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC = A \oplus B \oplus C$$

全减器差

$$Z_4 = \overline{A}\,\overline{B}C + \overline{A}B + ABC = (A \odot B)C + \overline{A}B$$

全减器借位

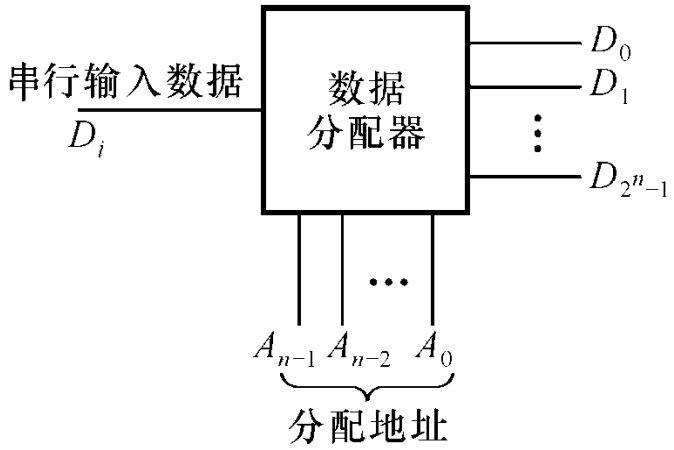
【例3】用8选1数据选择器74HC151实现序列脉冲输出。



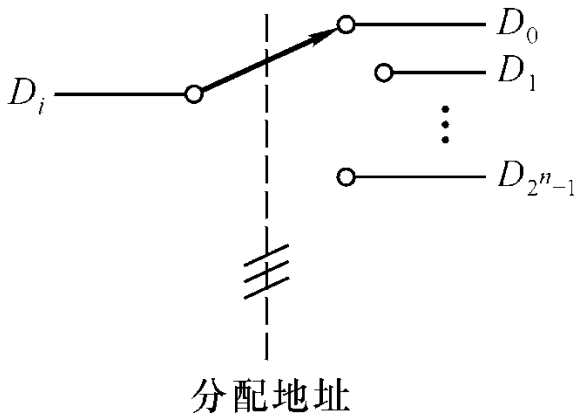
把多路数据选择器的数据输入端接上预先设计好的序列数据，而在地址控制端依次加上地址，则在选择器的输出端Y将可以输出一个序列脉冲。图示电路是产生01011001序列脉冲的逻辑电路。

二、数据分配器

数据分配器是将一串行输入数据，在*n*位地址的控制下，依次分配到2^n个通道上去。

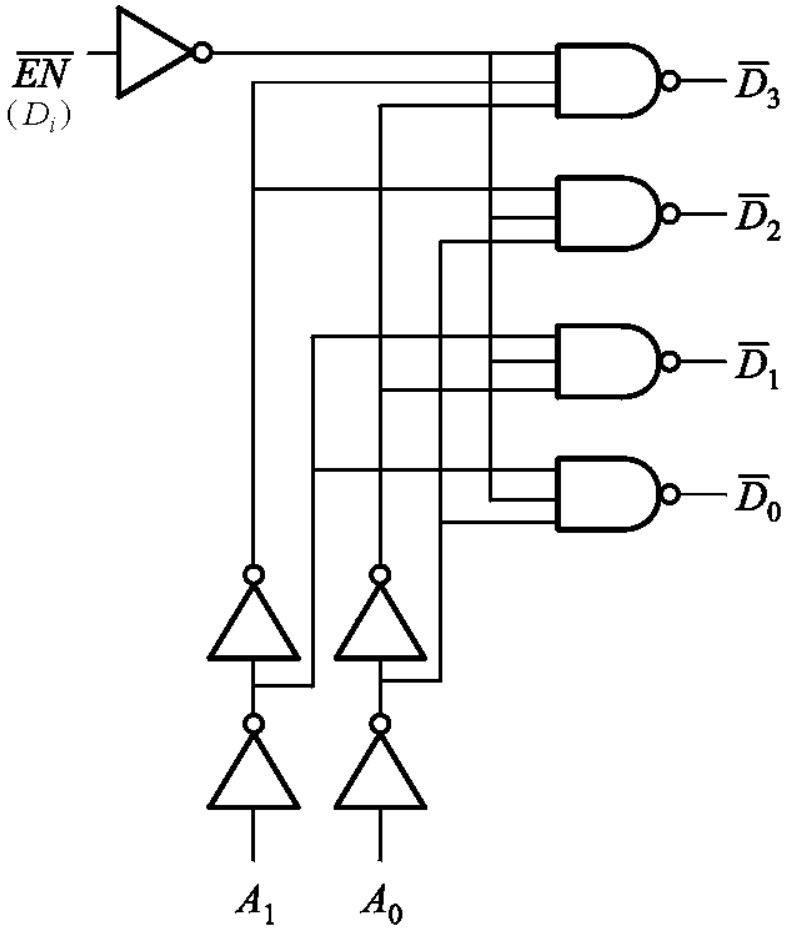


原理图



模拟图

数据分配器电路图



图中D_i是串行数据，

A_1A_0是分配地址，

$\overline{D}_3, \overline{D}_2, \overline{D}_1, \overline{D}_0$ 是四个输出通道。

称为**1/4分配器**。

在某种意义上，数据分配器是将串行输入信号转换成并行输出。

数据分配器真值表：

地址		数据	输　出			
D_i	A_1	A_0	$\overline{D_3}$	$\overline{D_2}$	$\overline{D_1}$	$\overline{D_0}$
D_i	0	0	1	1	1	D_i
D_i	0	1	1	1	D_i	1
D_i	1	0	1	D_i	1	1
D_i	1	1	D_i	1	1	1

2线-4线译码电路真值表

使能控制	输 入		输　出			
$\overline{EN}$	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

数据分配器实际上是一个译码器，A_1A_0当作译码器的代码输入，D_i作译码器的使能控制。因此，一个具有使能控制端的译码器又可作数据分配器。

三、二进制加法器

- 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、除四种最基本的算术运算。
- 而在数字电路中，又只需具有加法运算和移位操作就能实现乘除法的运算。所以，加法电路是最基本的。
- 在加法电路中半加电路和全加电路又是最低层的。

一、一位加法器

➢ 半加器

仅由两数据相应位相加，不计进位。若相应位为 A_i 、B_i，相加后产生半加和为 S_i，向高位进位为 C_i 。

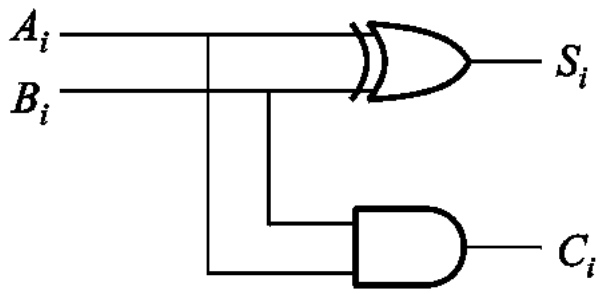
由此得到真值表：

输入		输出	
被加数 A_i	加数 B_i	半加和 S_i	进位 C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

由真值表得到两个输出函数式：

$$S_i = \overline{A_i} B_i + A_i \overline{B_i}$$

$$C_i = A_i B_i$$



- 由异或门、与门实现的电路：

- 全部用或非门实现

必须把函数式变换成或非-或非表达式。卡诺图中包围“0”格得或与表达式后，由二次求反得到：

$$\overline{S_i} = \overline{A_i}\,\overline{B_i} + A_i B_i$$

$$S_i = \overline{\overline{A_i}\,\overline{B_i} + A_i B_i} = \overline{\overline{\overline{A_i}\,\overline{B_i}} + \overline{A_i B_i}} = \overline{\overline{A_i} + \overline{B_i} + \overline{A_i + B_i}}$$

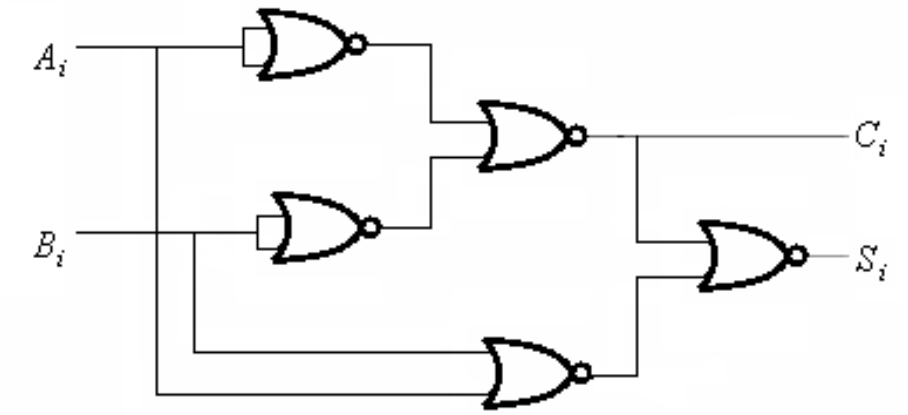
$$C_i = A_i B_i$$

$$S_i = (A_i + B_i)(\overline{A_i} + \overline{B_i})$$

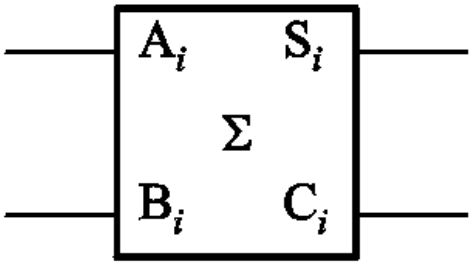
$$C_i = \overline{\overline{A_i} + \overline{B_i}}$$

输入		输出	
被加数 A_i	加数 B_i	半加和 S_i	进位 C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

电路图为：

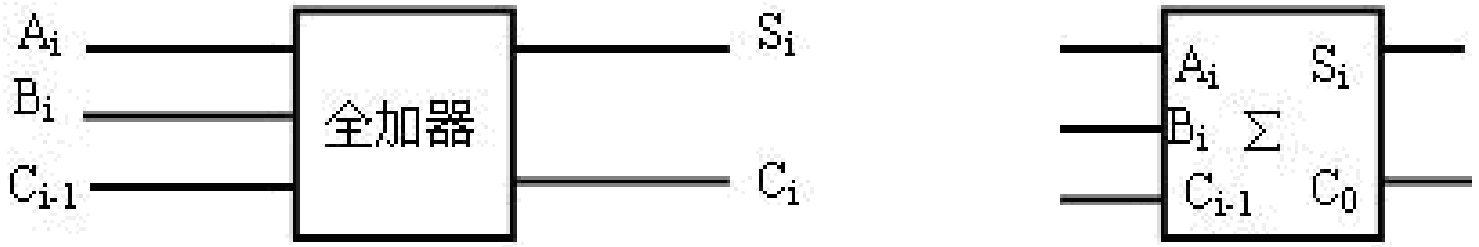


半加器内部的电路不管采用何种逻辑实现，都用逻辑符号表示：



一位全加器

能实现二个加数的对应位和相邻低位的进位一起相加的加法电路。



列出真值表：

全加器输入			结果输出	
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

从表可得到二个输出函数如下：

$$S_i = \overline{A_i}\,\overline{B_i}C_{i-1} + \overline{A_i}B_i\overline{C_{i-1}} + A_i\overline{B_i}\,\overline{C_{i-1}} + A_iB_iC_{i-1}$$
$$= \Sigma m(1,2,4,7)$$

$$C_i = \overline{A_i}B_iC_{i-1} + A_i\overline{B_i}C_{i-1} + A_iB_i\overline{C_{i-1}} + A_iB_iC_{i-1}$$
$$= \Sigma m(3,5,6,7)$$

S_i A_i \ B_iC_{i-1}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

C_i A_i \ B_iC_{i-1}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

（1）由上式可用与非门实现，图略。

（2）当用半加器实现时，需对上述式子作变换。

S_i / A_i	B_iC_{i-1} 00	01	11	10
0	0	1	0	1
1	1	0	1	0

棋盘格→异或逻辑关系

C_i / A_i	B_iC_{i-1} 00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$S_i = \overline{A_i}\,\overline{B_i}C_{i-1} + \overline{A_i}B_i\overline{C_{i-1}} + A_i\overline{B_i}\,\overline{C_{i-1}} + A_iB_iC_{i-1}$$

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

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

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

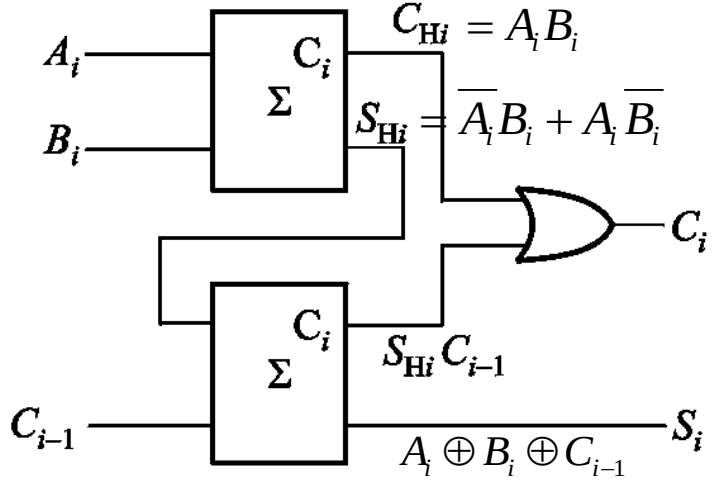
$$C_i = A_iB_i + A_i\overline{B_i}C_{i-1} + \overline{A_i}B_iC_{i-1}$$

$$= A_iB_i + S_{Hi}C_{i-1}$$

$$= C_{Hi} + S_{Hi}C_{i-1}$$

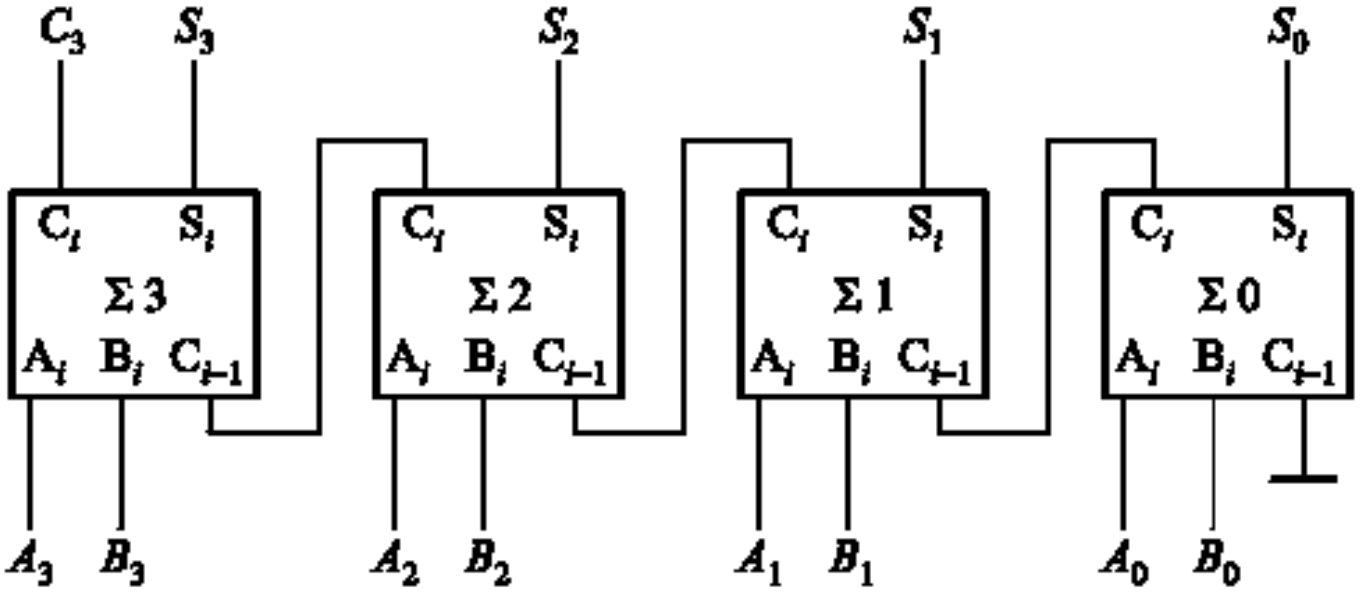
$$S_{Hi} = \overline{A_i}B_i + A_i\overline{B_i}$$

$$C_{Hi} = A_iB_i$$



二、多位二进制加法器

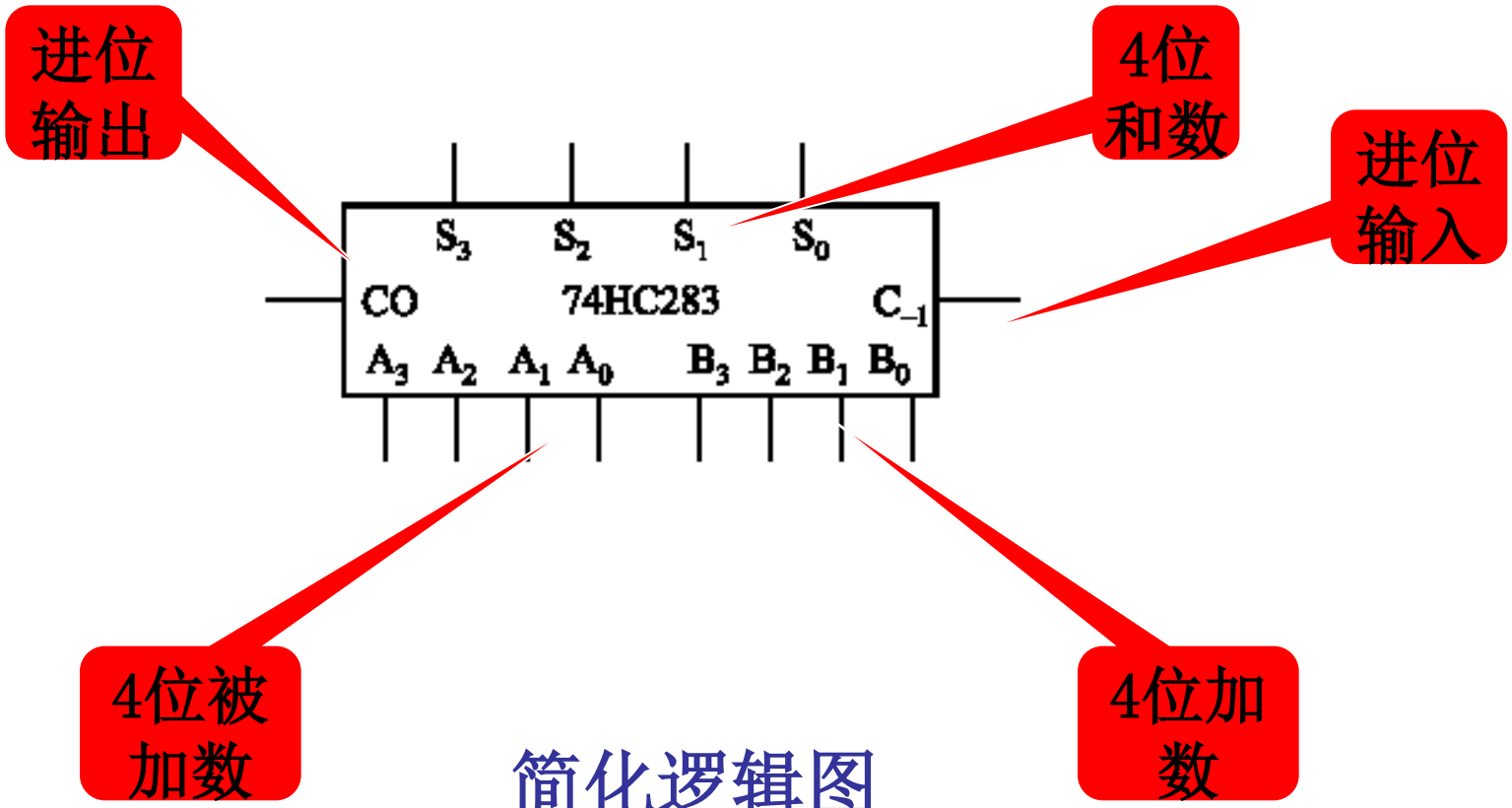
多位二进制加法电路种类很多，如4位并行输入串行进位加法电路，可由四个1位全加器组成，如图所示：



这种加法方式称为串行进位，其运算速度是比较低的。每做一次加法运算，需要经过4个全加器的传输延迟时间，才能得到稳定可靠的运算结果。

中规模集成二进制加法器

1. 74HC283型4位二进制加法器



简化逻辑图

2. 用74HC283实现减法运算

二进制的减法运算可以通过补码的加法来实现，首先将被减数和减数都变成补码，然后做加法运算。

两个N位的二进制数相减，可以通过被减数和减数的补码相加后再减去2^n实现。

$$D=A-B=A-(2^n-[B]_{补码})=A+[B]_{补码}-2^n$$

1000-0011

1000

+1100

+ 1

10101

如**co=1**,表明被减数>减数

0011-1000

0011

+0111

+ 1

01011

如**co=0**,表明被减数<减数

要再求补码

$$A \oplus A = 0 \qquad A \oplus 1 = \overline{A}$$

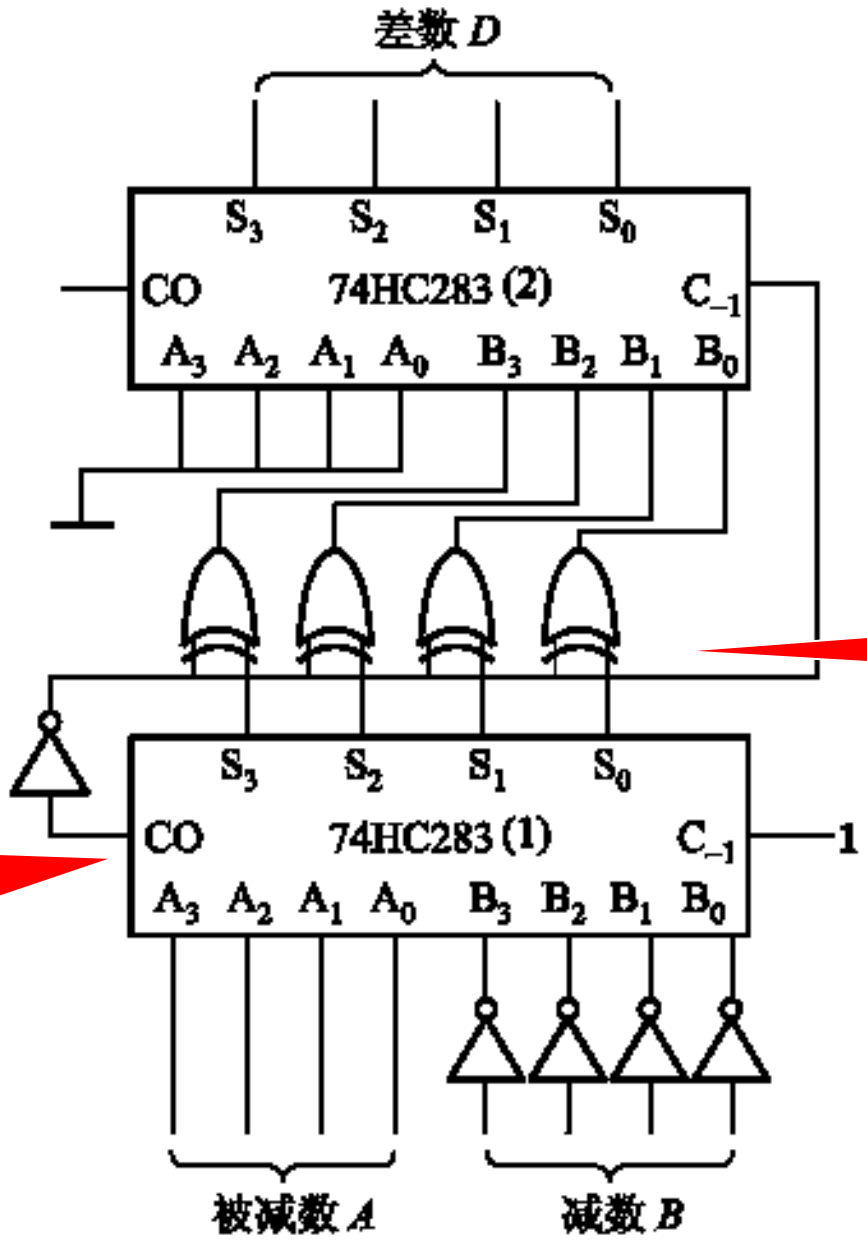
$$A \oplus \overline{A} = 1 \qquad A \oplus 0 = A$$

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

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

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

两个4位二进制数相减的逻辑电路



得到和数补码

该片做被减数和减数补码相加运算

1000-0011

1000

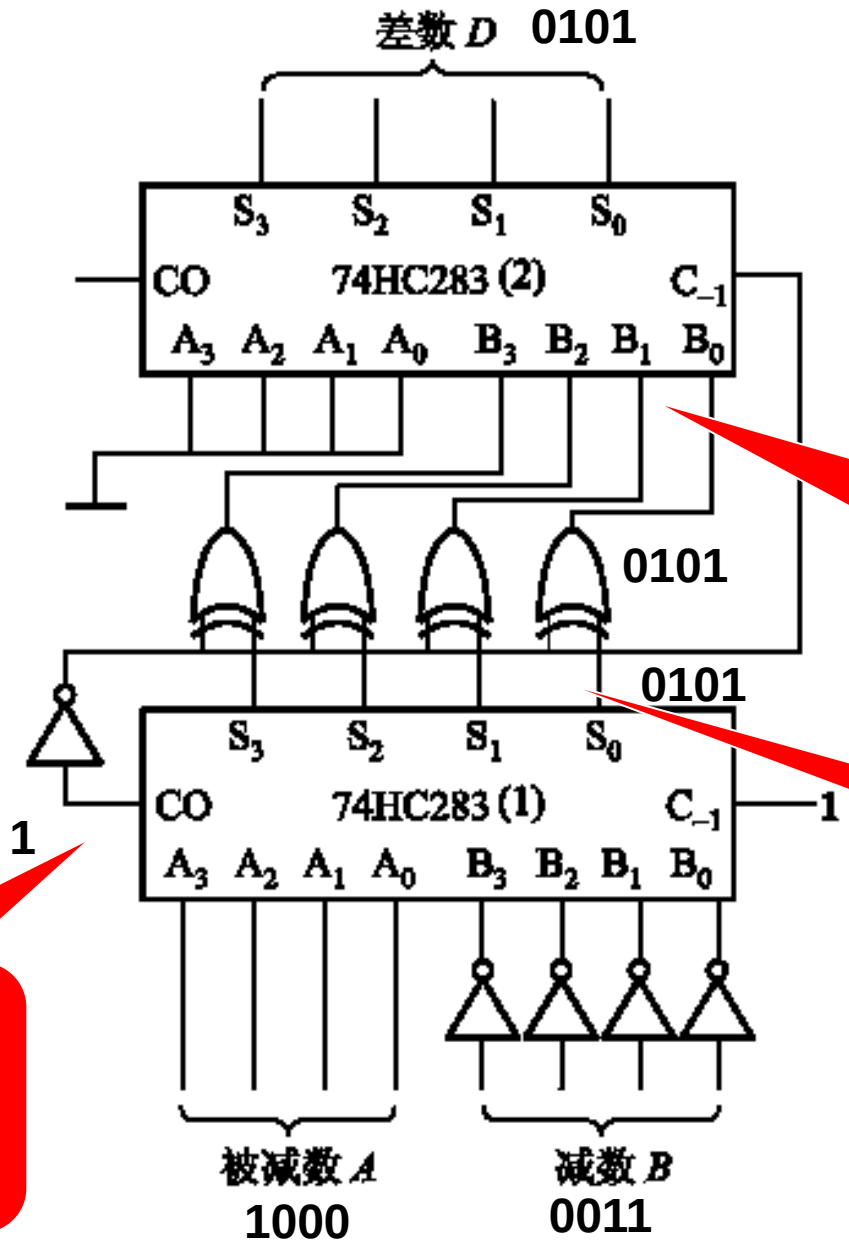
+1100

+ 1

10101

$A \oplus 0 = A$

$A \oplus 1 = \overline{A}$



该片将片I的正数加零，结果差也为正数

得到和数为正

如co=1，表明被减数>减数

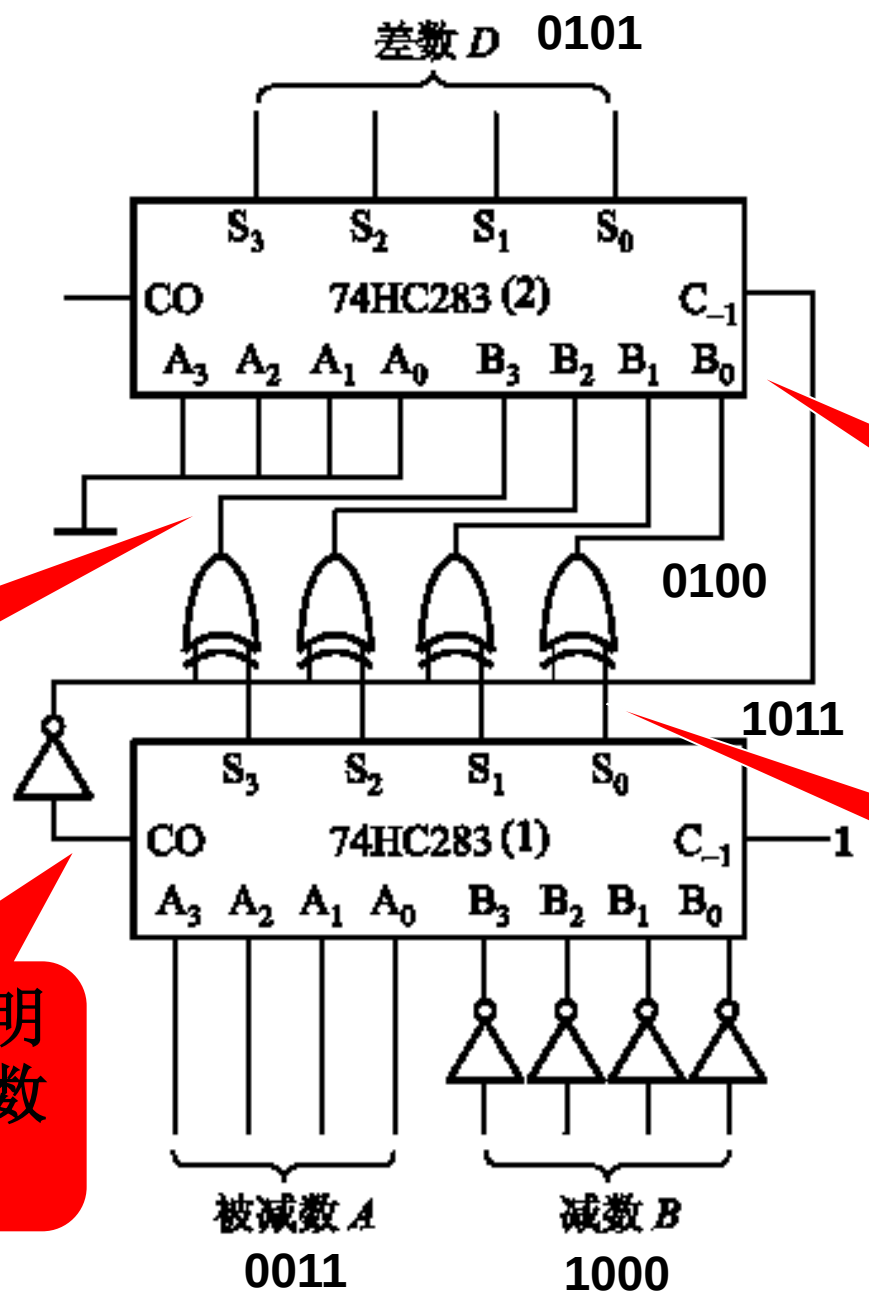
0011-1000

0011

+0111

+ 1

01011



将补码求反加1

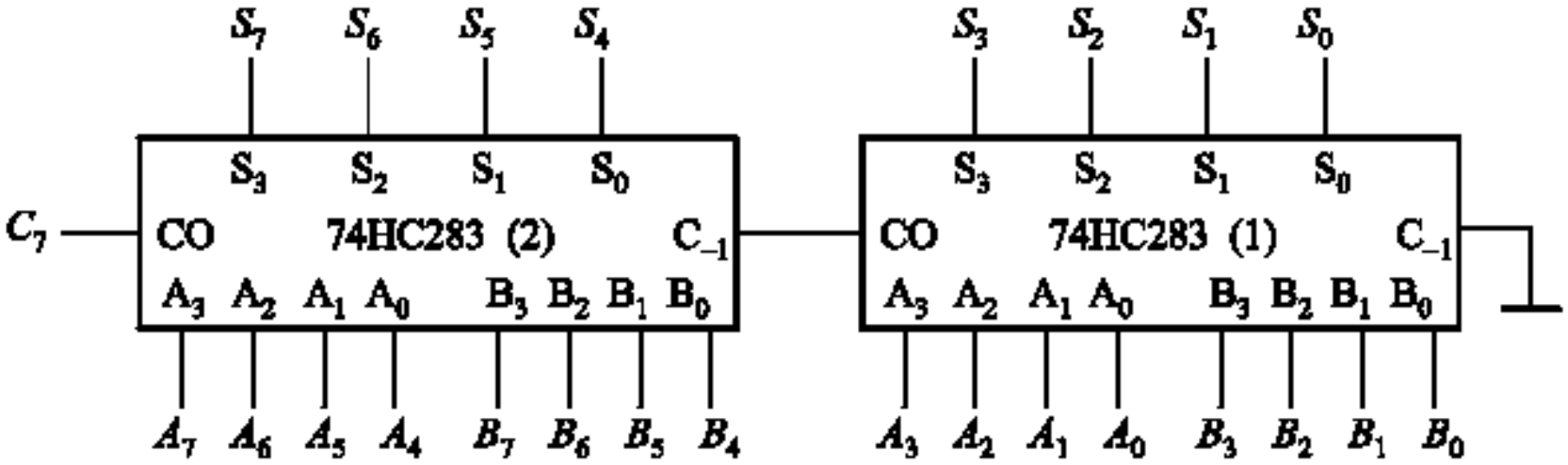
该片实现反码加1，实现原码的差

得到和数为补码

如co=0, 表明被减数<减数

3. 实现多数二进制数相加

用两片74HC283实现两个8位的二进制数加法运算。

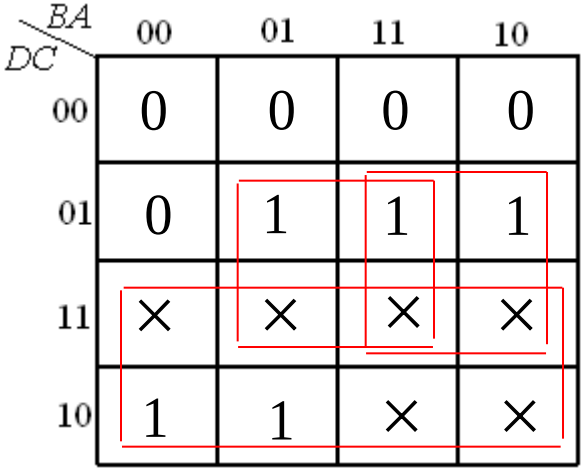


4. 实现代码间的转换

74HC283只能做加法，其功能不能改变。要实现代码转换其基本思路是：待转换的代码加上某个数即成目的代码了。

将8421BCD码转换成2421BCD码时，其真值表如表所示。

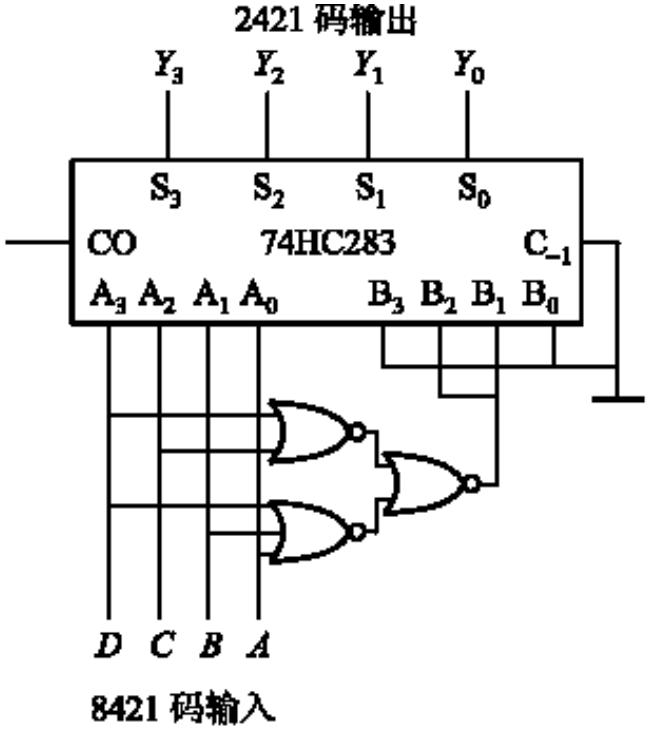
被加数				加数				结果代码			
8	4	2	1	B_3	B_2	B_1	B_0	2	4	2	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	1	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	0	1	0	0
0	1	0	1	0	1	1	0	1	0	1	1
0	1	1	0	0	1	1	0	1	1	0	0
0	1	1	1	0	1	1	0	1	1	0	1
1	0	0	0	0	1	1	0	1	1	1	0
1	0	0	1	0	1	1	0	1	1	1	1



B2、B1卡诺图

被加数				加数				结果代码			
8	4	2	1	B_3	B_2	B_1	B_0	2	4	2	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	1	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	0	1	0	0
0	1	0	1	0	1	1	0	1	0	1	1
0	1	1	0	0	1	1	0	1	1	0	0
0	1	1	1	0	1	1	0	1	1	0	1
1	0	0	0	0	1	1	0	1	1	1	0
1	0	0	1	0	1	1	0	1	1	1	1

逻辑图



$$B_2 = B_1 = D + CA + CB$$

圈1结果

圈0结果

$$= \overline{\overline{DC} + \overline{DBA}}$$

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

四、 数值比较器

数值比较器用来比较二个数据的大、小、是否相等，它经常用在逻辑判断，执行程序的跳转路径或执行何种操作等场合。分为串行比较器和并行比较器。

一、1位并行数值比较器

一位比较器框图

比较输入		结果输出		
A_i	B_i	$L_{A_i>B_i}$	$L_{A_i<B_i}$	$L_{A_i=B_i}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

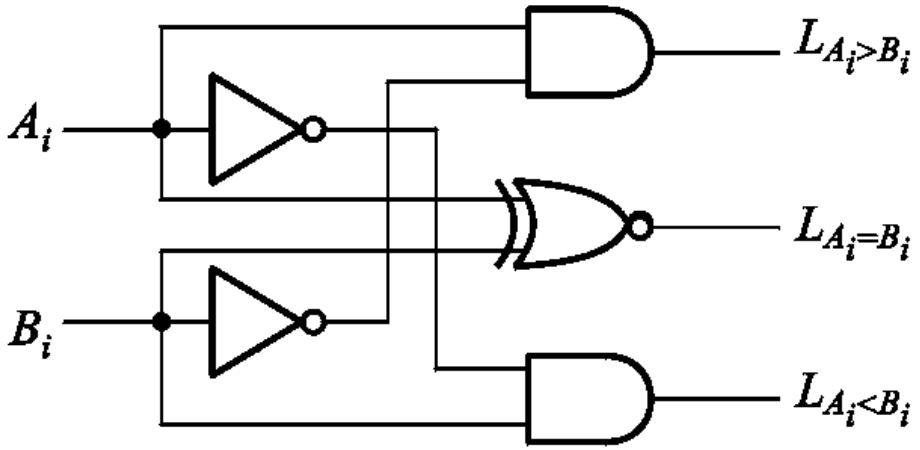
$$L_{A_i>B_i} = A_i \overline{B}_i$$

$$L_{A_i<B_i} = \overline{A}_i B_i$$

$$L_{A_i=B_i} = \overline{A}_i \overline{B}_i + A_i B_i$$

比较输入		结果输出		
A_i	B_i	$L_{A_i>B_i}$	$L_{A_i<B_i}$	$L_{A_i=B_i}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

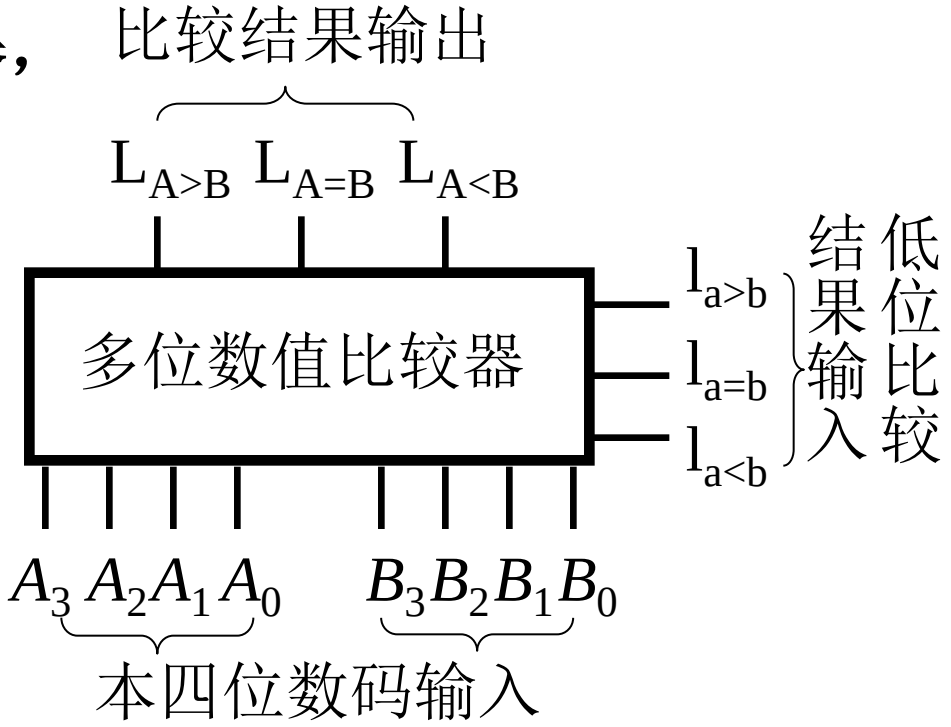
由函数式画出电路图：



二、多位数值比较器

多位数值比较器通常用“高位优先”的比较原则，如两个4位的数值比较器A和B，$A=A_3A_2A_1A_0$、$B=B_3B_2B_1B_0$，若$A_3 > B_3$，则$A > B$；若$A_3 < B_3$，则$A < B$；若高位相等时，按同样的原则比较次高位，如此进行，直到最低位比较完毕。

例：设计一个4位数码比较器，要求除比较本四位以外，在本四位相等时，还能比较低位的比较结果，以便能实现更多位的比较。框图如图所示：



解： 列出真值表如下：

本四位输入				低位结果输入			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	1	0
E_3	G_2	×	×	×	×	×	1	0	0
E_3	L_2	×	×	×	×	×	0	1	0
E_3	E_2	G_1	×	×	×	×	1	0	0
E_3	E_2	L_1	×	×	×	×	0	1	0
E_3	E_2	E_1	G_0	×	×	×	1	0	0
E_3	E_2	E_1	L_0	×	×	×	0	1	0
E_3	E_2	E_1	E_0	1	0	0	1	0	0
E_3	E_2	E_1	E_0	0	1	0	0	1	0
E_3	E_2	E_1	E_0	0	0	1	0	0	1

$$L_{A>B} = G_3 + E_3G_2 + E_3E_2G_1 + E_3E_2E_1G_0 + E_3E_2E_1E_0l_{a>b}$$

$$G_3 = A_3\overline{B_3} \qquad E_3 = \overline{A_3 \oplus B_3} \qquad G_2 = A_2\overline{B_2} \qquad E_2 = \overline{A_2 \oplus B_2}$$

$$G_1 = A_1\overline{B_1} \qquad G_0 = A_0\overline{B_0} \qquad E_1 = \overline{A_1 \oplus B_1} \qquad E_0 = \overline{A_0 \oplus B_0}$$

由真值表得出三个输出函数式如下：

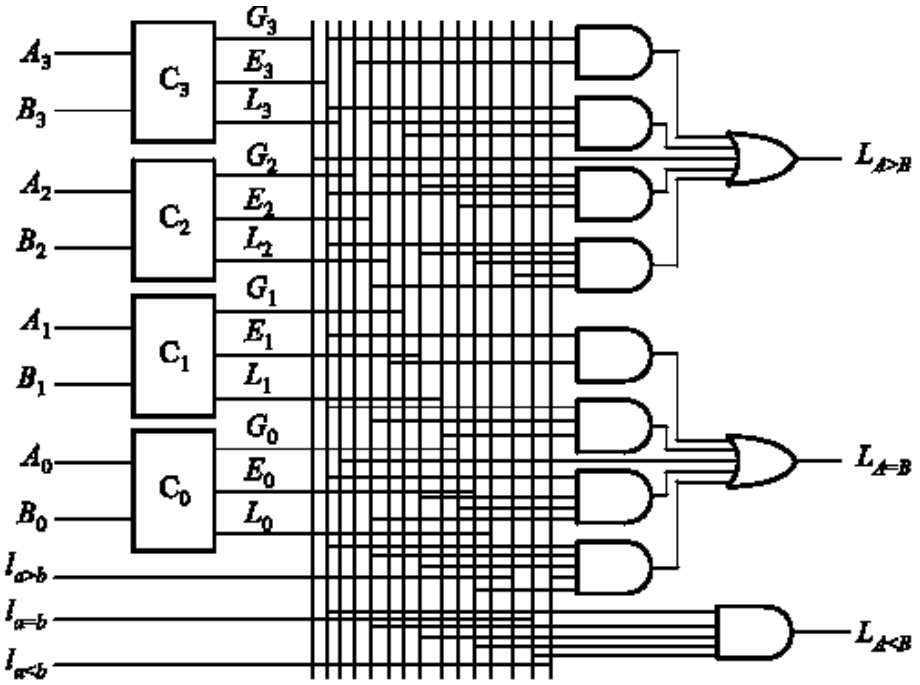
$$L_{A>B} = G_3 + E_3G_2 + E_3E_2G_1 + E_3E_2E_1G_0 + E_3E_2E_1E_0l_{a>b}$$

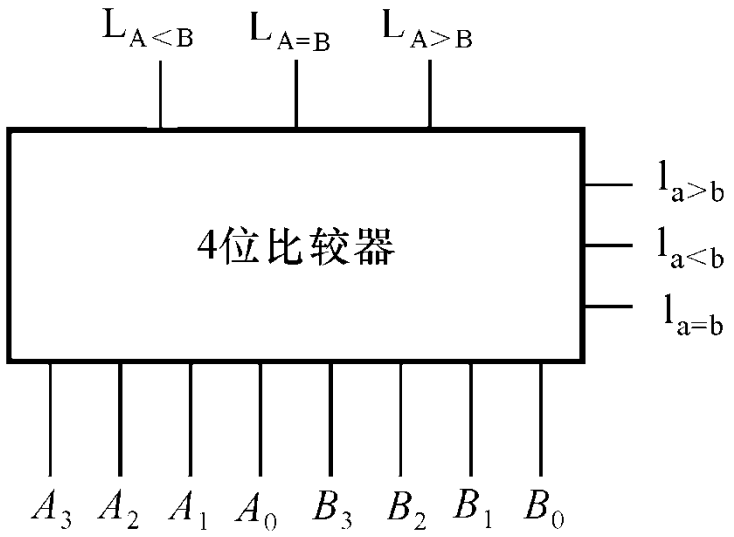
$$L_{A<B} = L_3 + E_3L_2 + E_3E_2L_1 + E_3E_2E_1L_0 + E_3E_2E_1E_0l_{a<b}$$

$$L_3 = \overline{A_3}B_3 \quad L_2 = \overline{A_2}B_2 \quad L_1 = \overline{A_1}B_1 \quad L_0 = \overline{A_0}B_0$$

$$L_{A=B} = E_3E_2E_1E_0l_{a=b}$$

由此画出电路图（基本框图）：



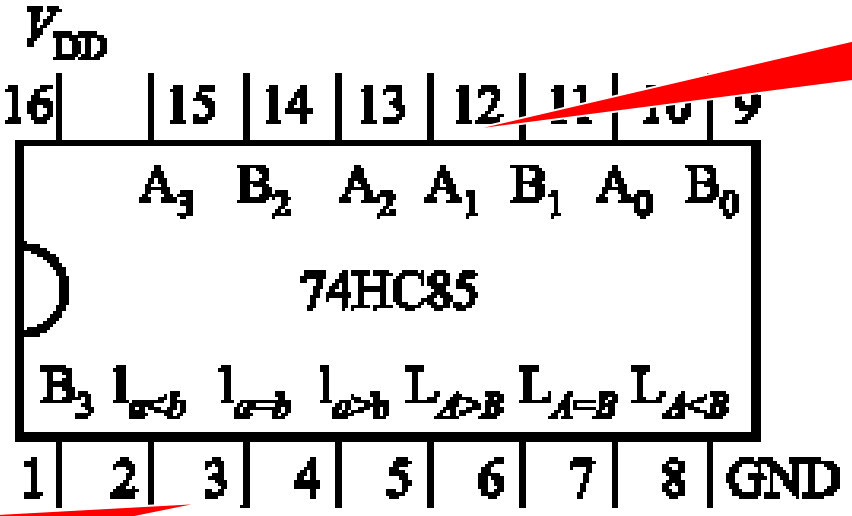


电路符号

任意位数值比较器的实现

1. 四位数值比较器74HC85介绍

74HC85是通用4位数值比较器，用该片可以实现任意位的数值比较。

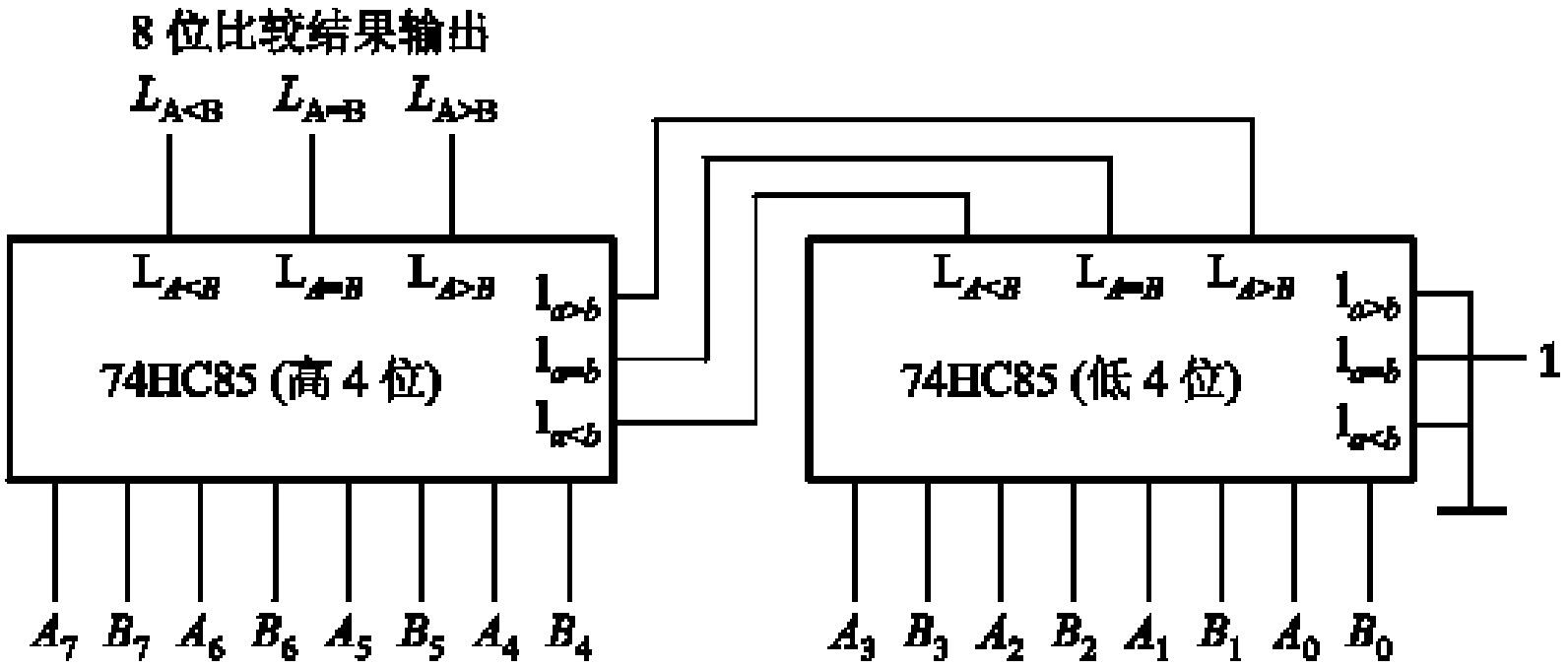


两个4位比较码输入

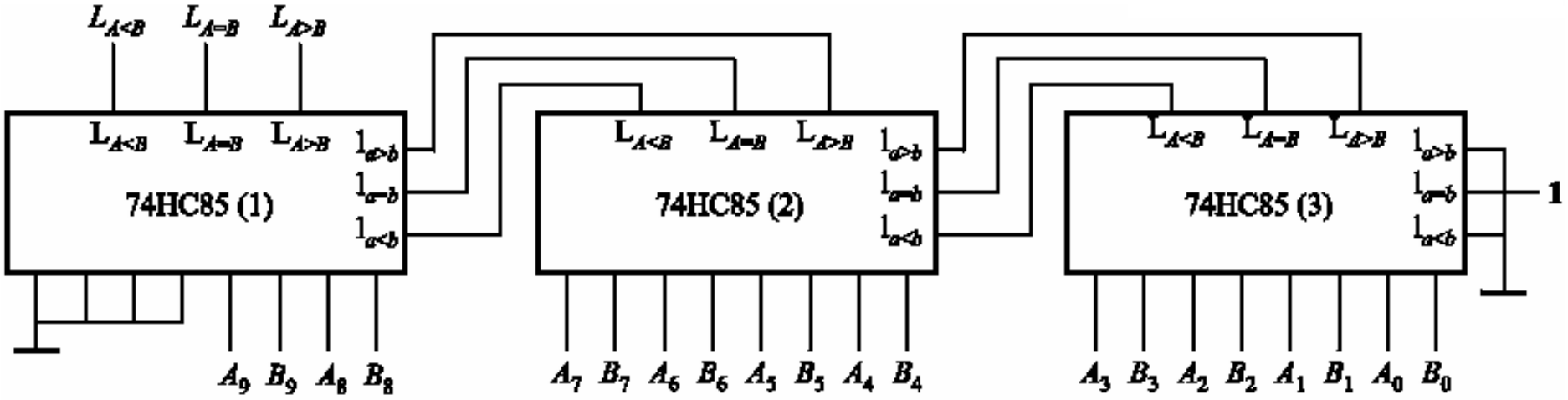
低位比较结果输入

比较结果输出

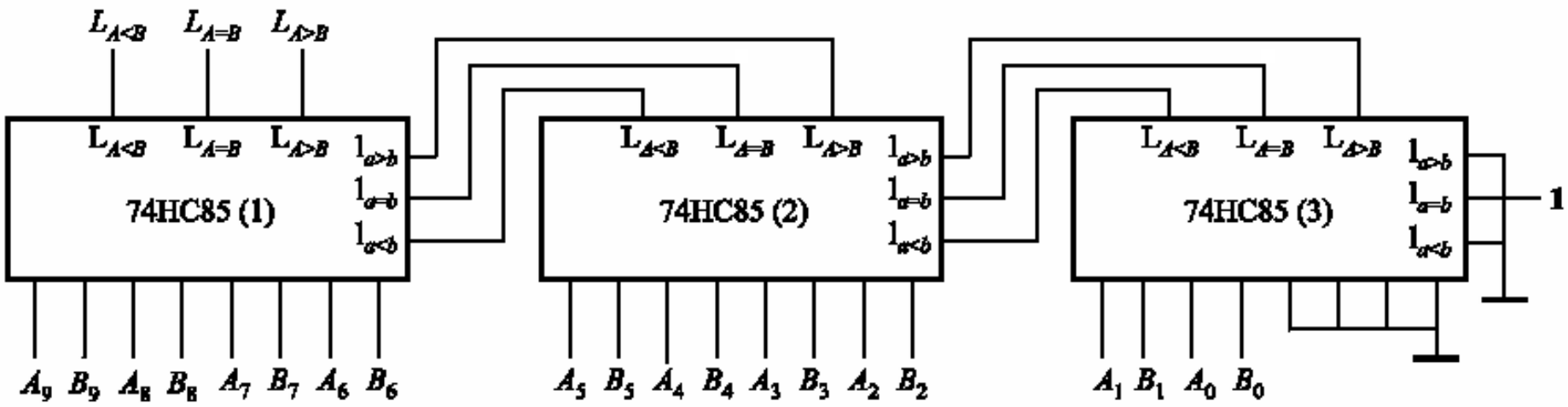
两个4位实现8位数值比较



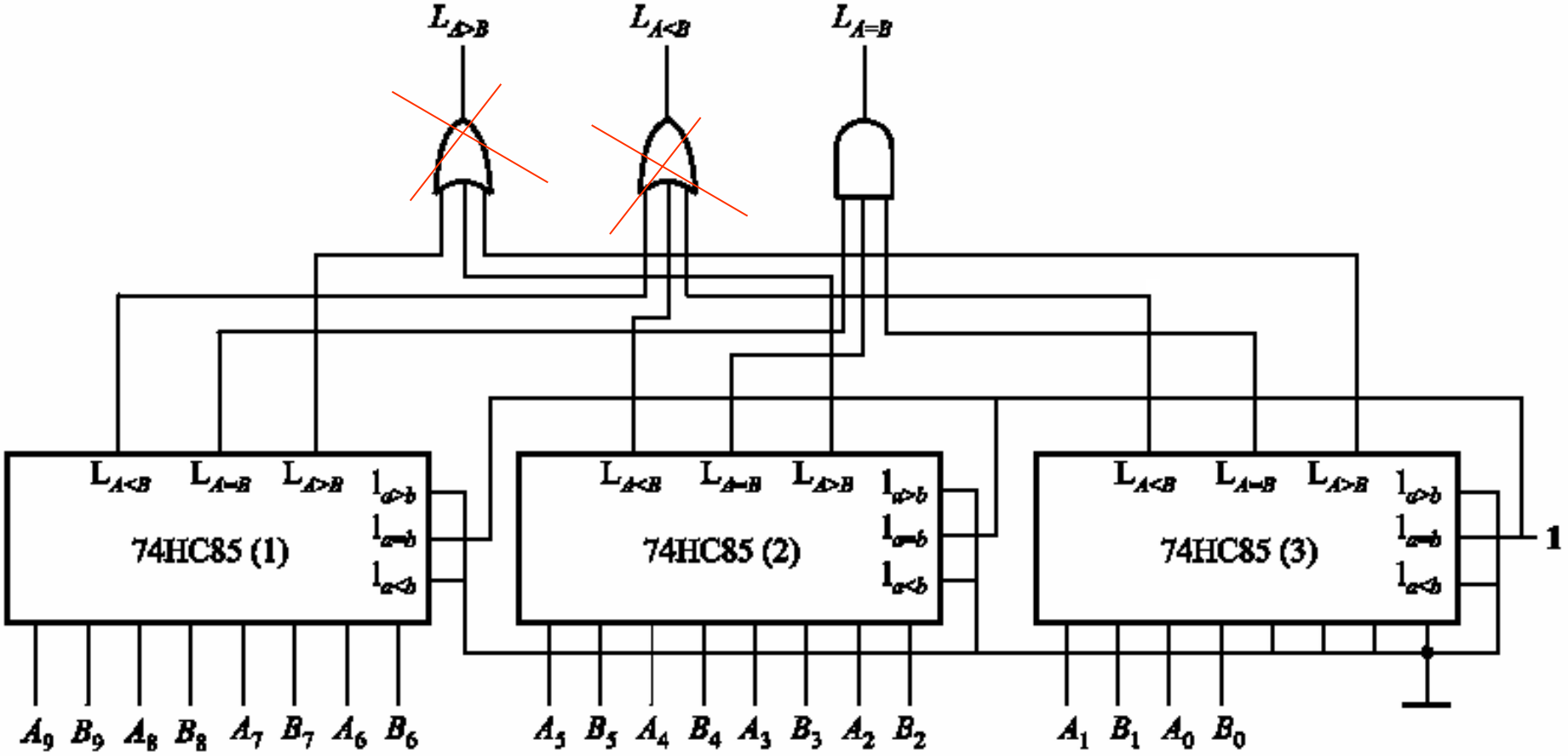
3个4位实现10位数值比较



方案1　　高2位接”0”


$L_{A<B}$ $L_{A=B}$ $L_{A>B}$
74HC85 (1)
74HC85 (2)
74HC85 (3)
$1_{a>b}$ $1_{a=b}$ $1_{a<b}$
1
A_9 B_9 A_8 B_8 A_7 B_7 A_6 B_6
A_5 B_5 A_4 B_4 A_3 B_3 A_2 B_2
A_1 B_1 A_0 B_0

方案2　　低2位接”0”



方案3　并联连接比较

$$L_{A>B} = L_{1A>B} + L_{1A=B}L_{2A>B} + L_{1A=B}L_{2A=B}L_{3A>B}$$

$$L_{A<B} = L_{1A<B} + L_{1A=B}L_{2A<B} + L_{1A=B}L_{2A=B}L_{3A<B}$$