

数 字 逻 辑

Digital Logic Circuit

丁 贤 庆

ahhfdxq@163.com

Home work (P218)

1、下次课会有一次10分钟的随堂测验。

2、今天的作业：

4.4.14

4.4.20

4.4.26

4.4.36

4.4.37

第4章 组合逻辑电路

Combinational Logic Circuit

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

4.4.2 译码器/数据分配器

Decoders

1 译码器的定义与分类

译码：译码是编码的逆过程，它可将二进制码翻译成代表某一特定含义的信号。(即电路的某种状态)

译码器：具有译码功能的逻辑电路称为译码器。

译码器的分类：

地址译码器 将一系列代码转换成与之一一对应的有效信号。

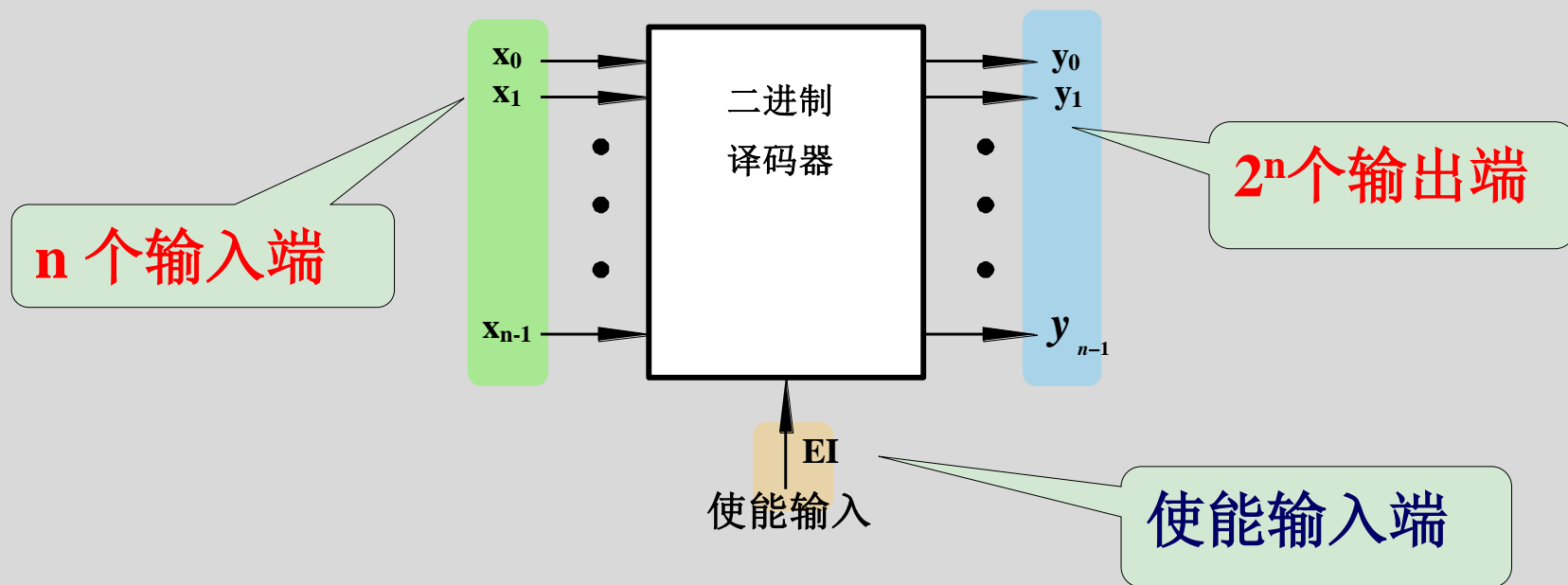
代码变换器 将一种代码转换成另一种代码。

常见的地址译码器：

- 二进制译码器
- 二—十进制译码器
- 显示译码器

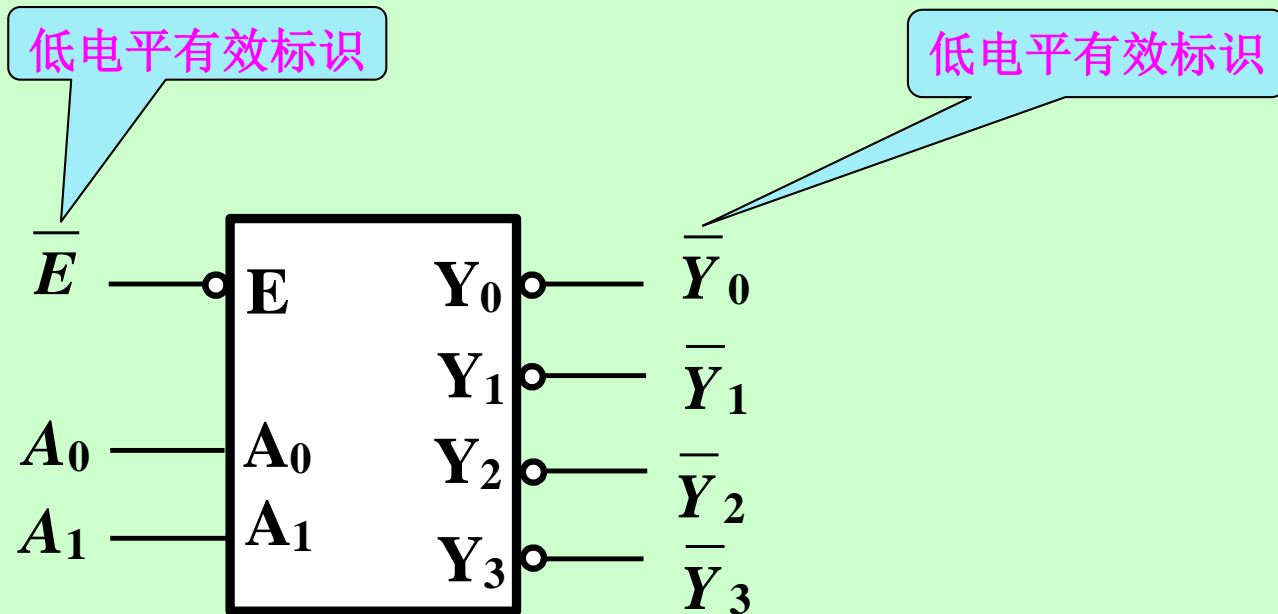
2. 典型译码器电路及应用

(1) 二进制译码器



设输入端的个数为 n ，输出端的个数为 M
则有 $M=2^n$

(a) 2线-4线译码器



在使能信号 $E=1$ 的情况下, 四个输出全部为1。(静态电流比较小, 功耗比较小)

在使能信号有效 ($E=0$) 的情况下, 对应的输出中, 只有1个为0, 其它输出全部为1。

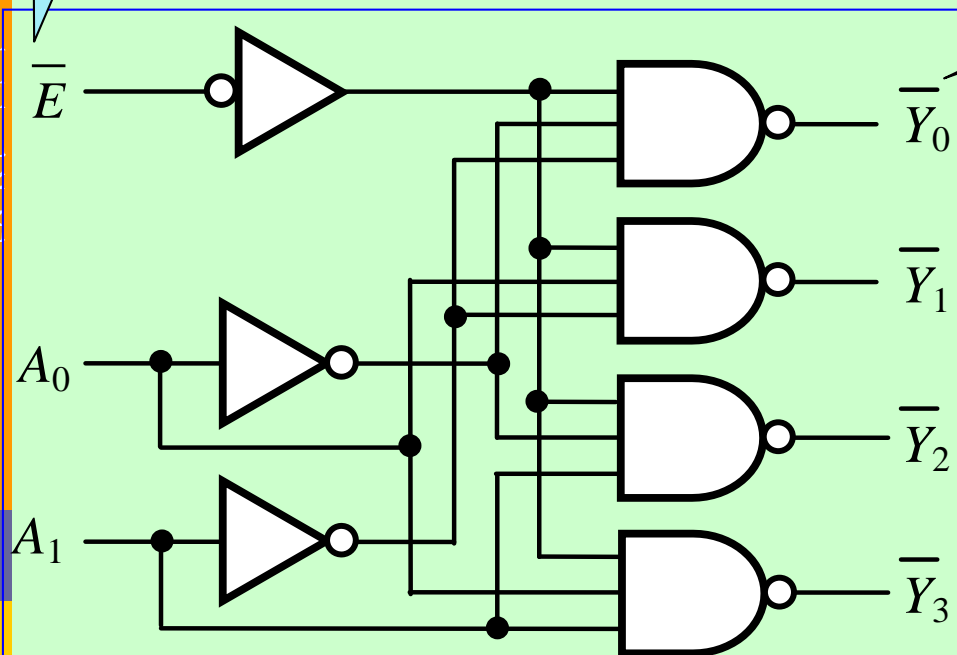
说明: 当电路的输出状态变化时, 静态电流比较大, 漏电流比较大, 功耗比较大。

低电平有效标识

在使能信号有效的情况下，每种输入对应的输出，只有1个为0。

2线 - 4线译码器的逻辑电路(分析)

低电平有效标识



功能表

输入			输出			
\overline{E}	A_1	A_0	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3
1	×	×	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

根据真值表中输出F=1的情况
能够写出F的最小项表达式。

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

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

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

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

根据真值表中输出F=0的情况
能够写出F的最小项表达式。

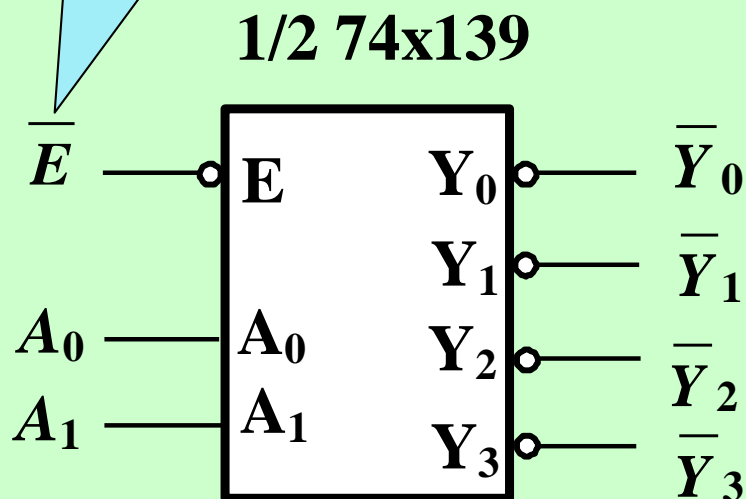
不考虑使能信号E的情况下，译码器的每个输出是一个最小项的非。

(a) 2线-4线译码器 (74HC139芯片)

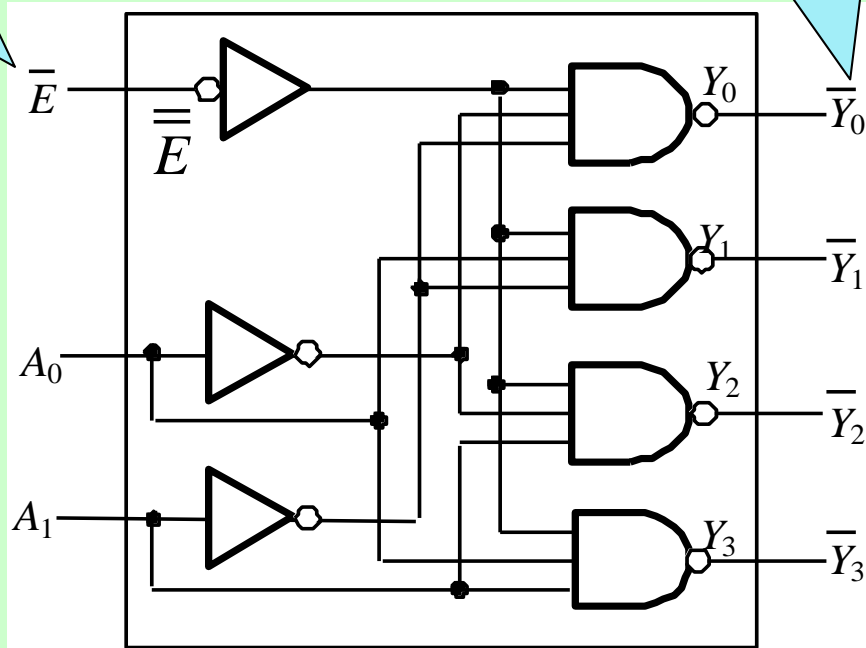
逻辑符号框外部的符号，表示外部输入或输出信号名称，字母上面的“—”号说明该输入或输出是低电平有效。在推导表达式的过程中，表示低有效的输入或输出变量(如)上面的“—”号只是一个提示符，不是“非”，不能参与运算。

低电平有效标识

低电平有效标识



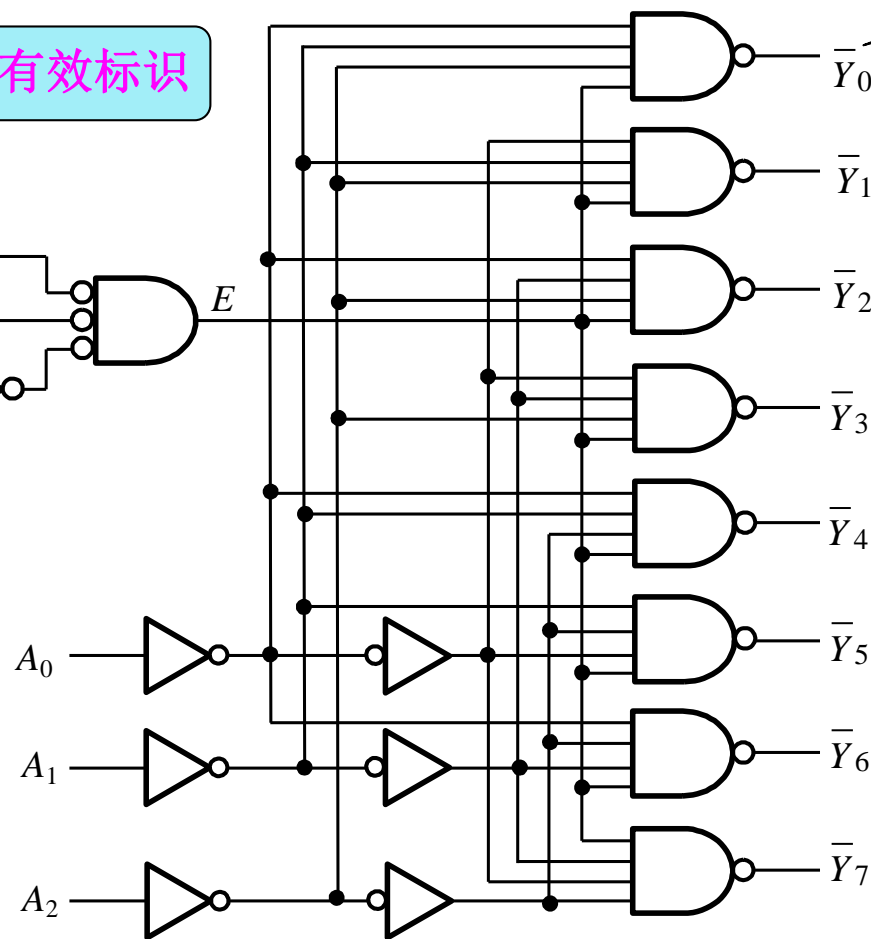
低电平有效标识



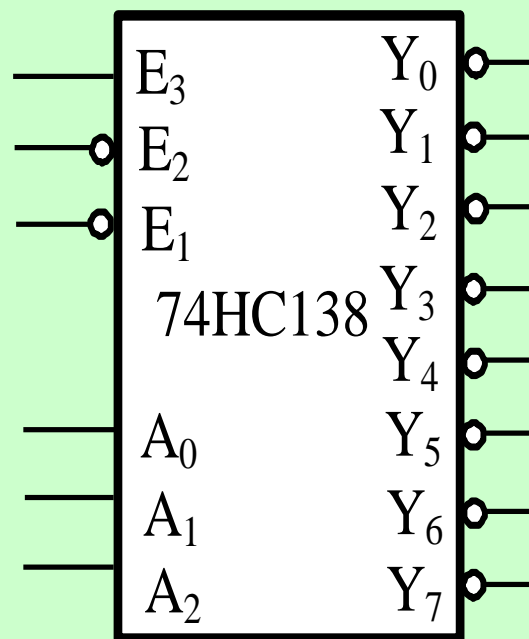
(b) 3线-8线译码器 (74HC138芯片)

低电平有效标识

低电平有效标识



逻辑图



逻辑符号

3线-8线译码器（74HC138芯片）功能表

低电平有效标识

低电平有效标识

输 入			输 出										
E_3	\overline{E}_2	\overline{E}_1	A_2	A_1	A_0	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3	\overline{Y}_4	\overline{Y}_5	\overline{Y}_6	\overline{Y}_7
×	1	×	×	×	×	1	1	1	1	1	1	1	1
×	X	1	×	×	×	1	1	1	1	1	1	1	1
0	×	×	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

根据真值表中输出 $F=1$ 的情况能够写出 F 的最小项表达式。

$$\overline{Y}_0 = \overline{A}_2 \cdot \overline{A}_1 \cdot \overline{A}_0;$$

根据真值表中输出 $F=0$ 的情况能够写出 \overline{F} 的最小项表达式。

$$\overline{Y_0} = \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}; \quad \overline{Y_1} = \overline{A_2} \cdot \overline{A_1} \cdot A_0; \quad \overline{Y_2} = \overline{A_2} \cdot A_1 \cdot \overline{A_0}; \quad \overline{Y_3} = \overline{A_2} \cdot A_1 \cdot A_0;$$

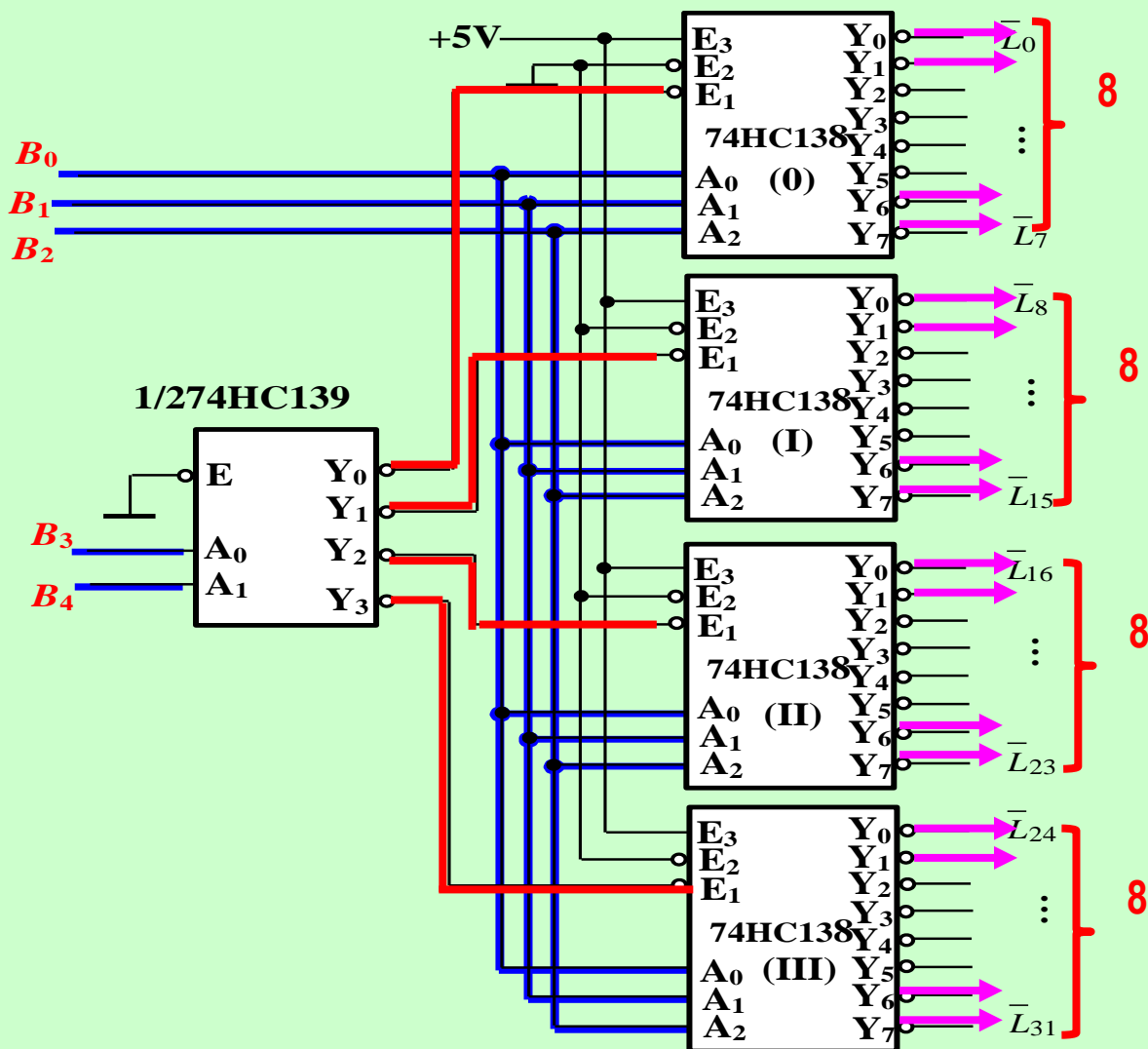
$$\overline{Y_4} = A_2 \cdot \overline{A_1} \cdot \overline{A_0}; \quad \overline{Y_5} = A_2 \cdot \overline{A_1} \cdot A_0; \quad \overline{Y_6} = A_2 \cdot A_1 \cdot \overline{A_0}; \quad \overline{Y_7} = A_2 \cdot A_1 \cdot A_0;$$

输 入						输 出							
E_3	$\overline{E_2}$	$\overline{E_1}$	A_2	A_1	A_0	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$
×	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
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

不考虑使能信号E的情况下，译码器的每个输出是一个最小项的非。

1、译码器的扩展

用74HC139和74HC138构成5线-32线译码器



74HC139是2线-4线译码器

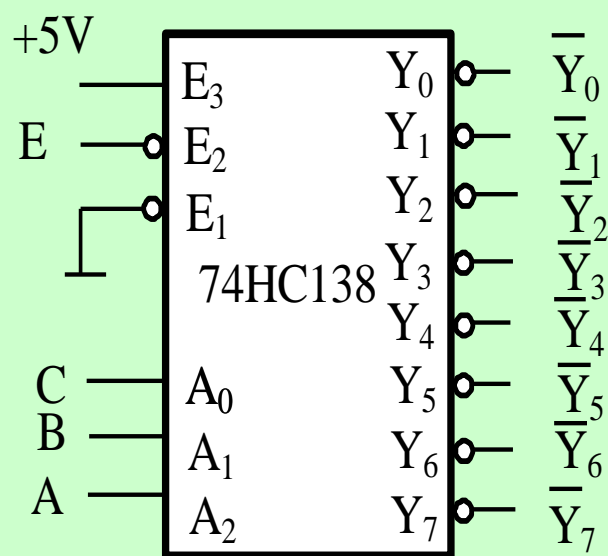
74HC138是3线-8线译码器

5线中的3线分给138芯片，
另外2线分给139芯片。

32线的输出可以看成
是由4个8线的输出组成。

2、用译码器实现逻辑函数。

当 $E_3=1$ ， $E_2=E_1=0$ 时，如果 A_2 、 A_1 、 A_0 分别连接 A 、 B 、 C 三路信号。则：



$$\overline{Y}_0 = \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0} = \overline{A} \cdot \overline{B} \cdot \overline{C} = \overline{m_0}$$

$$\overline{Y}_1 = \overline{A_2} \cdot \overline{A_1} \cdot A_0 = \overline{A} \cdot \overline{B} \cdot C = \overline{m_1}$$

$$\overline{Y}_2 = \overline{A_2} \cdot A_1 \cdot \overline{A_0} = \overline{A} \cdot B \cdot \overline{C} = \overline{m_2}$$

⋮

⋮

⋮

$$\overline{Y}_7 = A_2 \cdot A_1 \cdot A_0 = A \cdot B \cdot C = \overline{m_7}$$

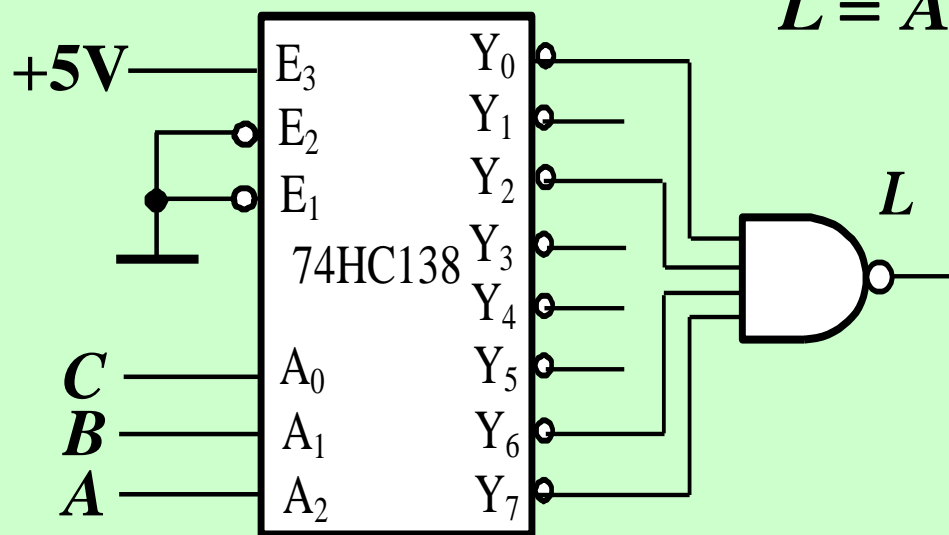
3线-8线译码器的 $\overline{Y}_0 \sim \overline{Y}_7$ 是三变量函数的最小项的非。

基于这一点用该器件能够方便地实现三变量逻辑函数。

例：用一片74HC138和与非门实现函数

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

解答：首先将函数式变换为最小项之和的形式



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

$$= m_0 + m_2 + m_6 + m_7$$

$$= \overline{m_0} \cdot \overline{m_2} \cdot \overline{m_6} \cdot \overline{m_7}$$

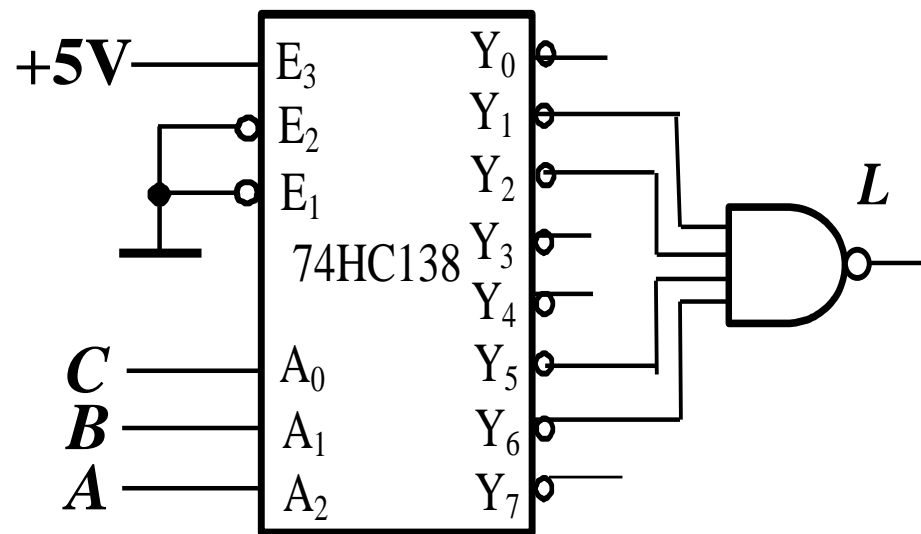
$$= \overline{Y_0} \cdot \overline{Y_2} \cdot \overline{Y_6} \cdot \overline{Y_7}$$

在译码器的输出端加一个与非门，即可实现给定的组合逻辑函数。

下图所示的电路，输出L对应的表达式，正确的是（ ）

- A $L = \overline{m_1} + \overline{m_2} + \overline{m_5} + \overline{m_6}$
- B $L = m_1 + m_2 + m_5 + m_6$**
- C $L = \overline{m_1 \cdot m_2 \cdot m_5 \cdot m_6}$
- D $L = \overline{m_1 + m_2 + m_5 + m_6}$

提交



(2) 二-十进制译码器的真值表

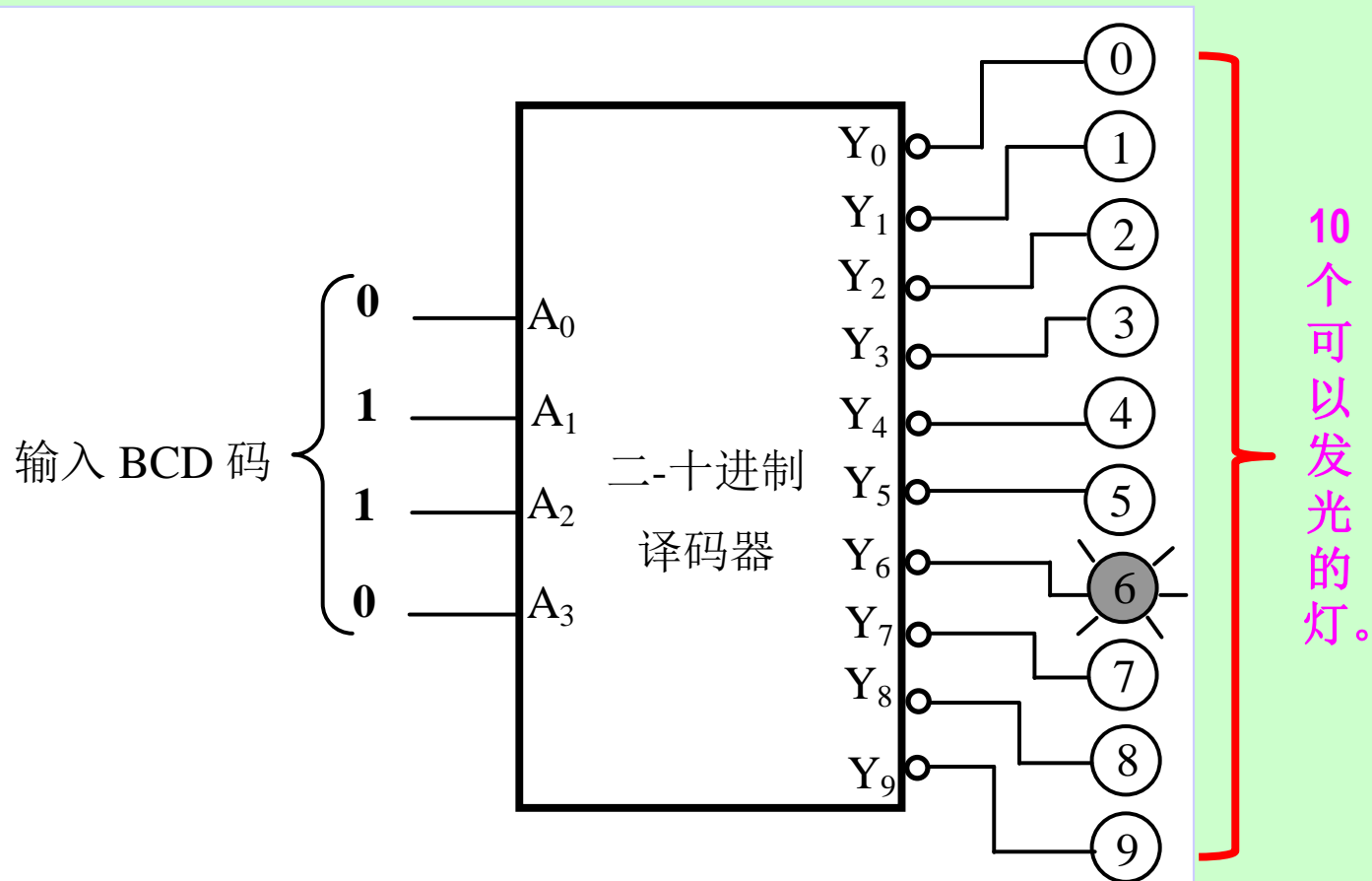
对于BCD代码以外的伪码（1010~1111这6个代码） $Y_0 \sim Y_9$ 均为高电平。

0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0

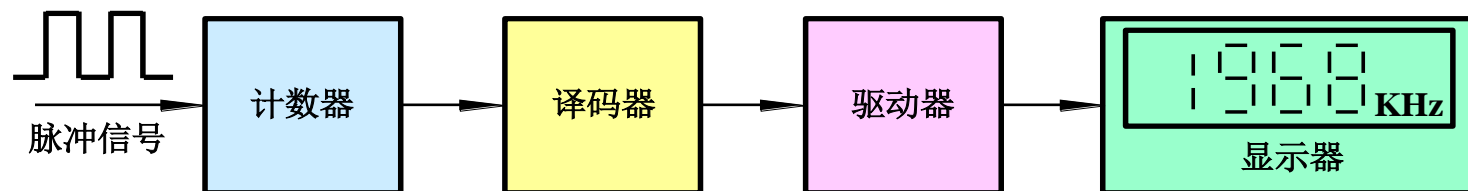
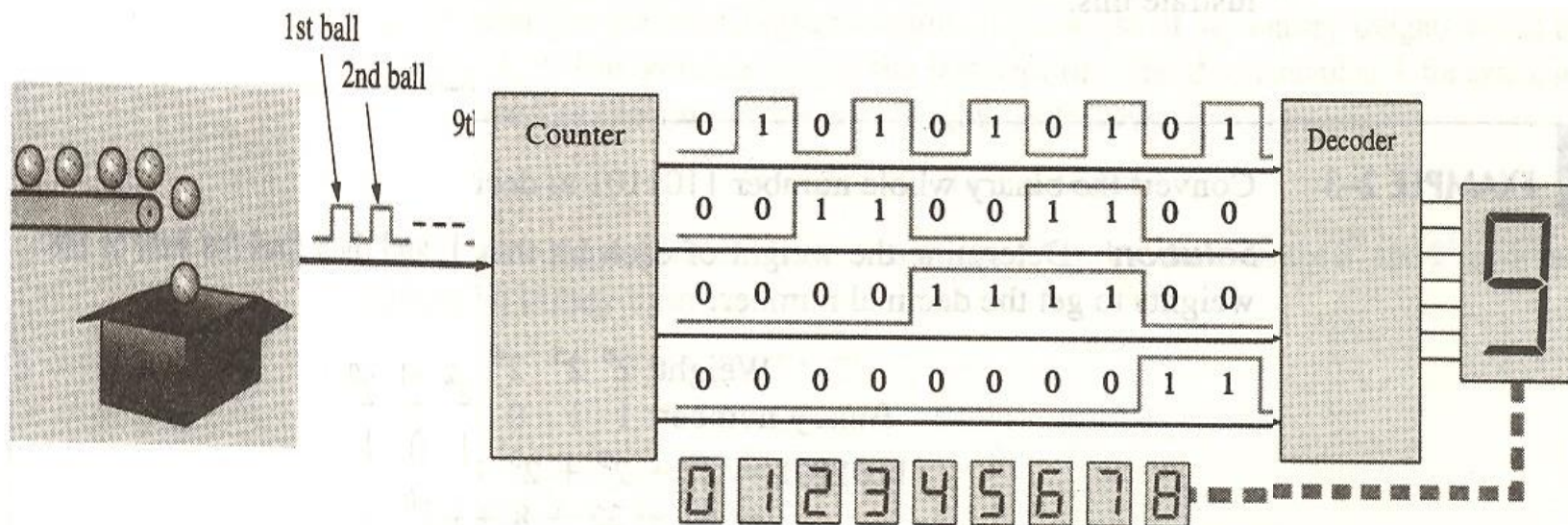
每种输入，对应的输出中，只有1位输出为0，其它位输出全部为1。

二-十进制译码器

功能：将8421BCD码译成为10个状态输出。



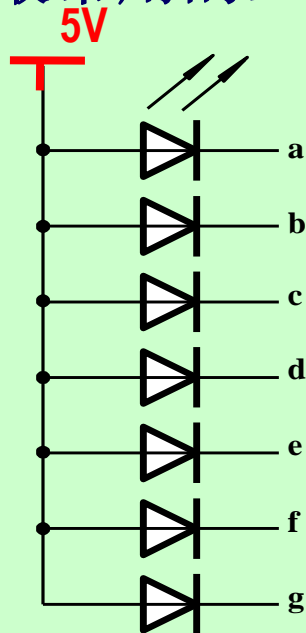
(3) 显示译码器



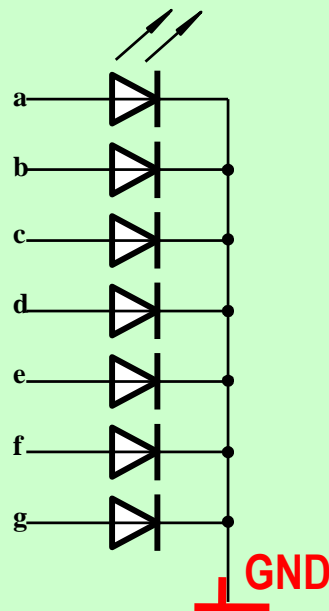
1. 七段显示译码器

The BCD-to-7-Segment Decoder

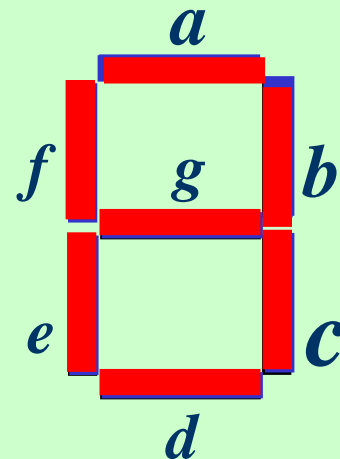
(1) 最常用的显示器有：半导体发光二极管和液晶显示器。



共阳极显示器



共阴极显示器



显示器分段布局图

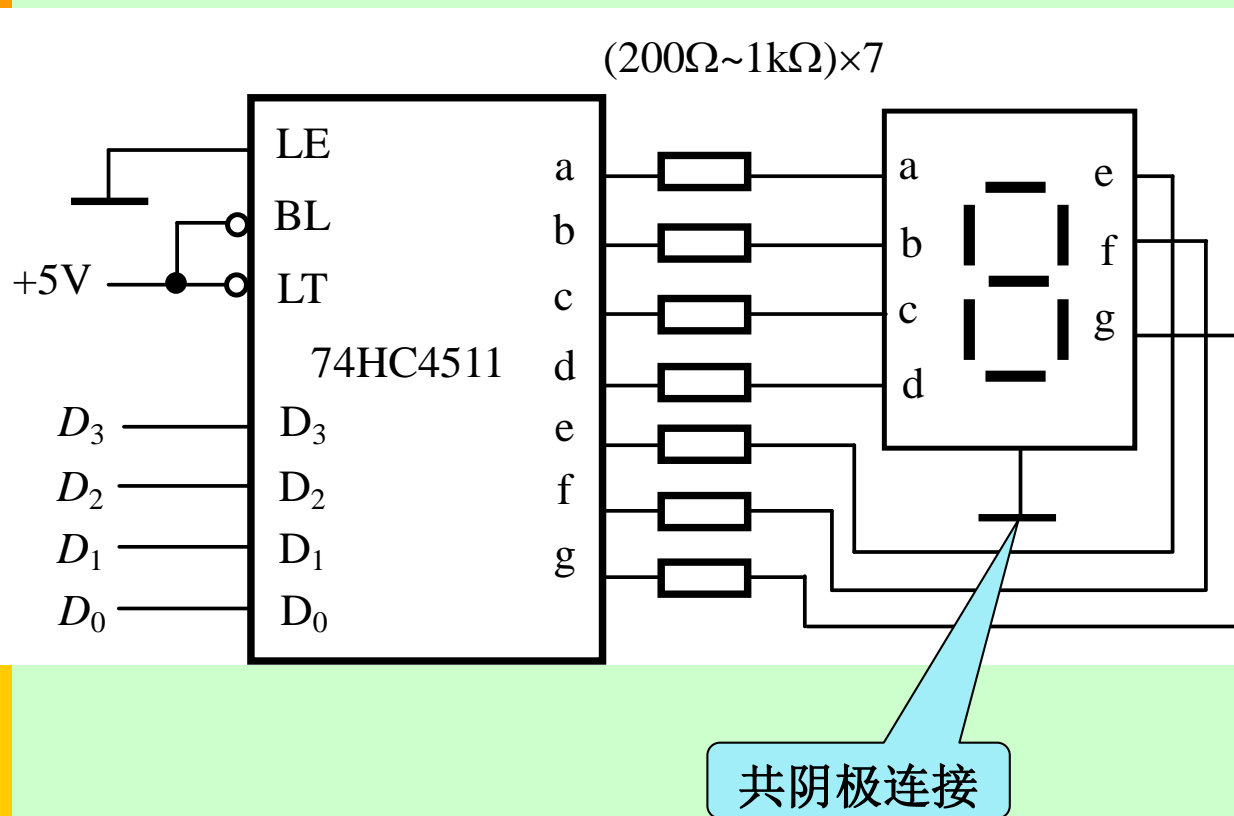
当共阳极接0V时，共阳极显示器不亮。

当共阴极接5V时，共阴极显示器不亮。

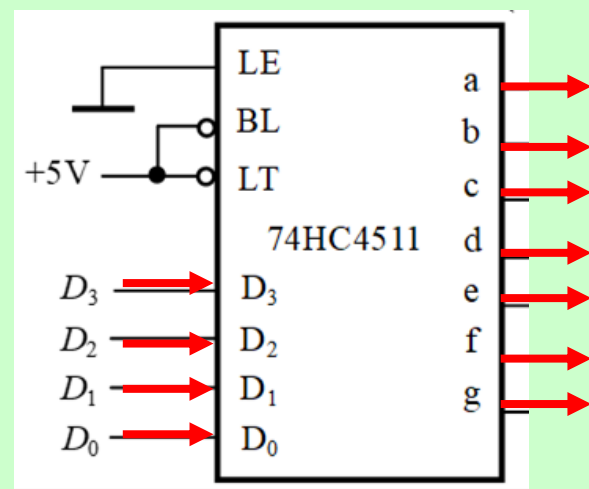
常用的集成七段显示译码器

-----CMOS七段显示译码器**74HC4511**

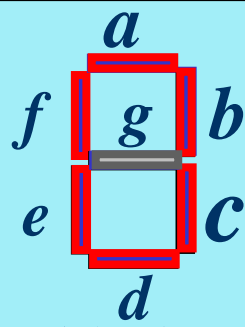
显示译码器与显示器的连接方式



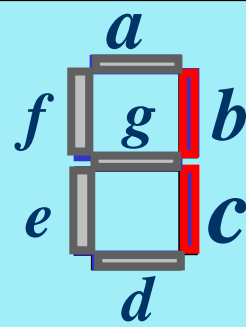
74HC4511输入与输出



CMOS七段显示译码器74HC4511功能表



显示“0”字形



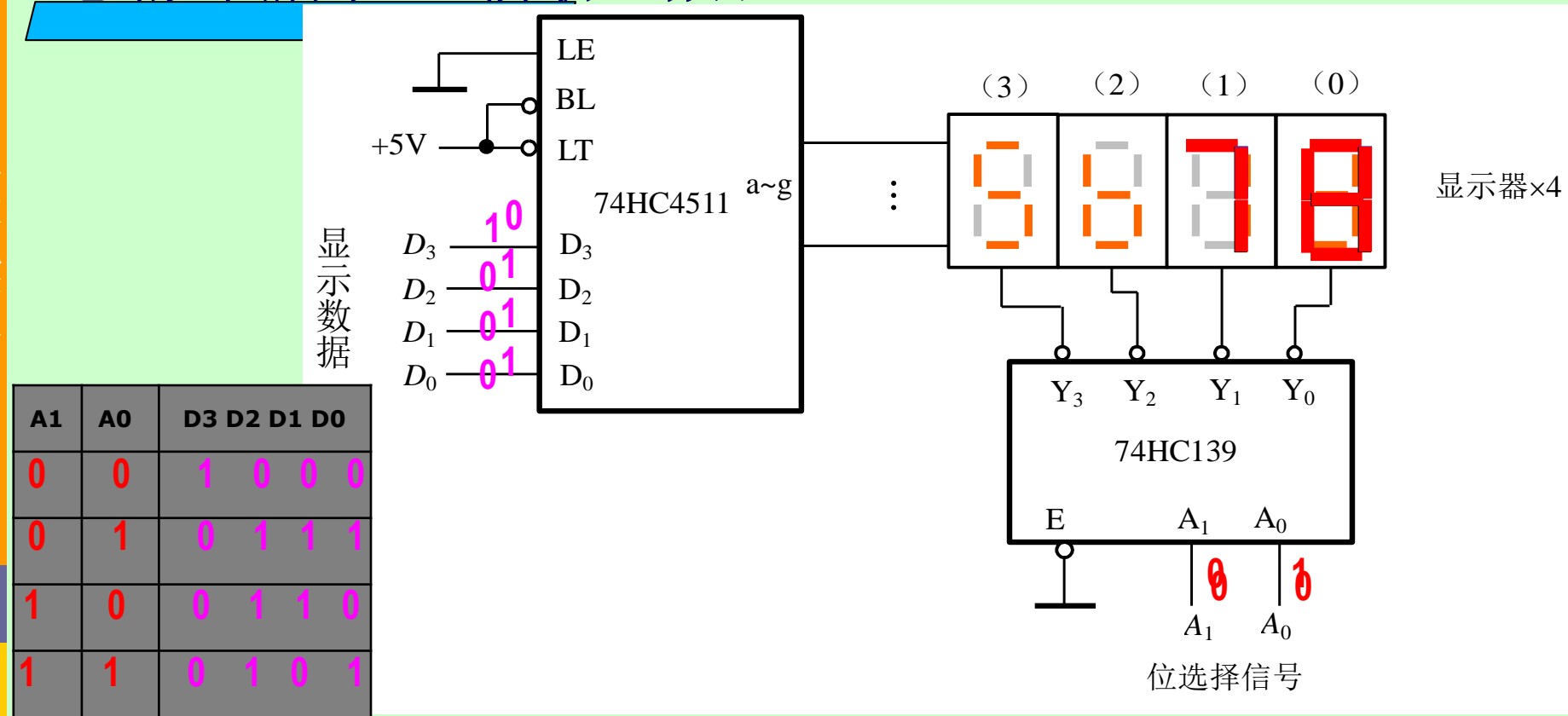
显示“1”字形

十进制或功能	输入							输出							字形
	LE	\overline{BL}	\overline{LT}	D_3	D_2	D_1	D_0	a	b	c	d	e	f	g	
0	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
3	0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
7	0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
8	0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
9	0	1	1	1	0	0	1	1	1	1	1	0	1	1	9

计算机学院

数字逻辑电路

例 由译码器、显示译码及4个七段显示器构成的4位动态显示电路如图所示，试分析工作原理。

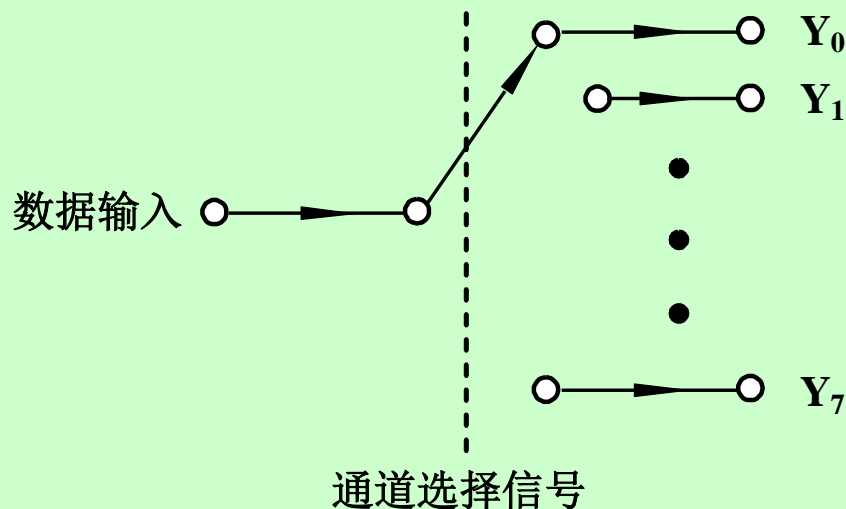


数字逻辑电路

位选择信号A1、A0控制 $\overline{Y_3} \sim \overline{Y_0}$ 依次产生低电平，使4个显示器轮流显示。要显示的数据组依次送到 $D_3D_2D_1D_0$ 分别在4个显示器上显示。利用人的视觉暂留时间，可以看到稳定的数字。

$$25\text{Hz} < f_c < 100\text{Hz}$$

数据分配器示意图

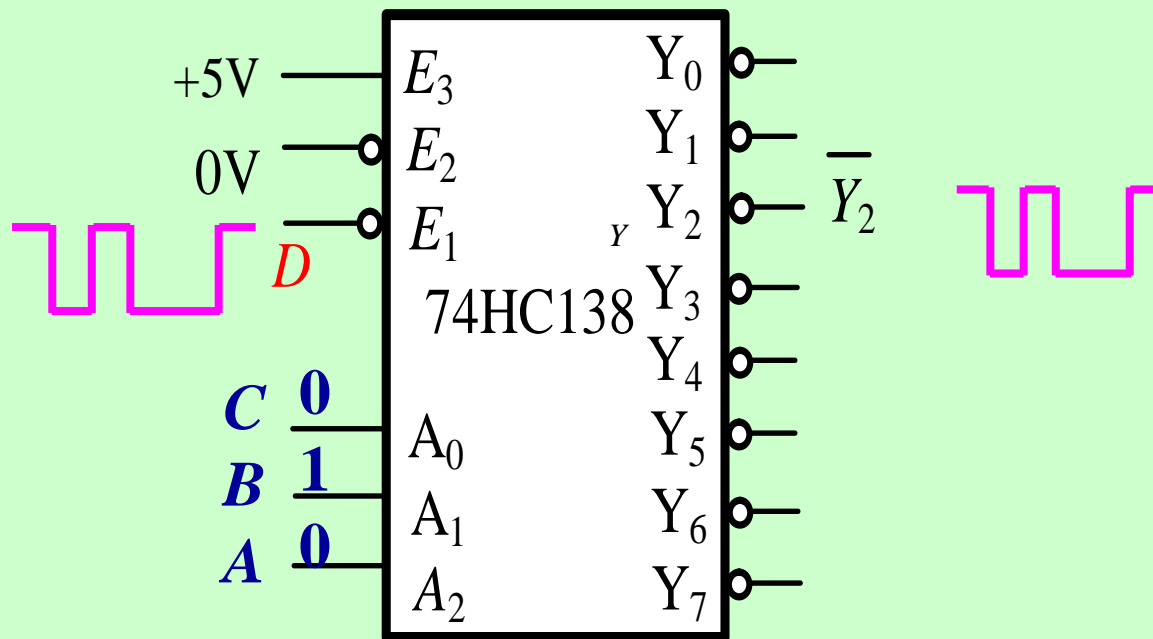


数据分配器：相当于多输出的单刀多掷开关，是将公共数据线上的数据按需要送到不同的通道上去的逻辑电路。

用译码器实现数据分配器

$$\overline{Y_2} = \overline{E_3 E_2 D A B C}$$

当 $ABC = 010$ 时, $\overline{Y_2} = D$



74HC138译码器作为数据分配器时的功能表

输 入						输 出							
E_3	$\overline{E_2}$	$\overline{E_1}$	A_2	A_1	A_0	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$
0	0	X	X	X	X	1	1	1	1	1	1	1	1
1	0	D	0	0	0	D	1	1	1	1	1	1	1
1	0	D	0	0	1	1	D	1	1	1	1	1	1
1	0	D	0	1	0	1	1	D	1	1	1	1	1
1	0	D	0	1	1	1	1	1	D	1	1	1	1
1	0	D	1	0	0	1	1	1	1	D	1	1	1
1	0	D	1	0	1	1	1	1	1	1	D	1	1
1	0	D	1	1	0	1	1	1	1	1	1	D	1
1	0	D	1	1	1	1	1	1	1	H	1	1	D

例：试用门电路设计一个具有低电平使能控制的1线—4线数据分配器，使能信号无效时，电路所有的输出为高阻态。当通道选择信号将1路输入信号连接到其中1路输出端时，其他输出端为高阻状态。

1. 列真值表
输出端有3种状态（0、1、z），输出级是4个三态门组成。其控制信号由 \overline{E} 、S1、S0共同作用产生。

输 入			输 出			
\overline{E}	S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	z	z	z	In
0	0	1	z	z	In	z
0	1	0	z	In	z	z
0	1	1	In	z	z	z
1	x	x	z	z	z	z

2. 写出4个三态门控制端的逻辑表达式(见下页图)

$$C_0 = \overline{\overline{E}} \cdot \overline{S_1} \cdot \overline{S_0} ; C_1 = \overline{\overline{E}} \cdot \overline{S_1} \cdot S_0 ; C_2 = \overline{\overline{E}} \cdot S_1 \cdot \overline{S_0} ; C_3 = \overline{\overline{E}} \cdot S_1 \cdot S_0 ;$$

3. 画逻辑电路

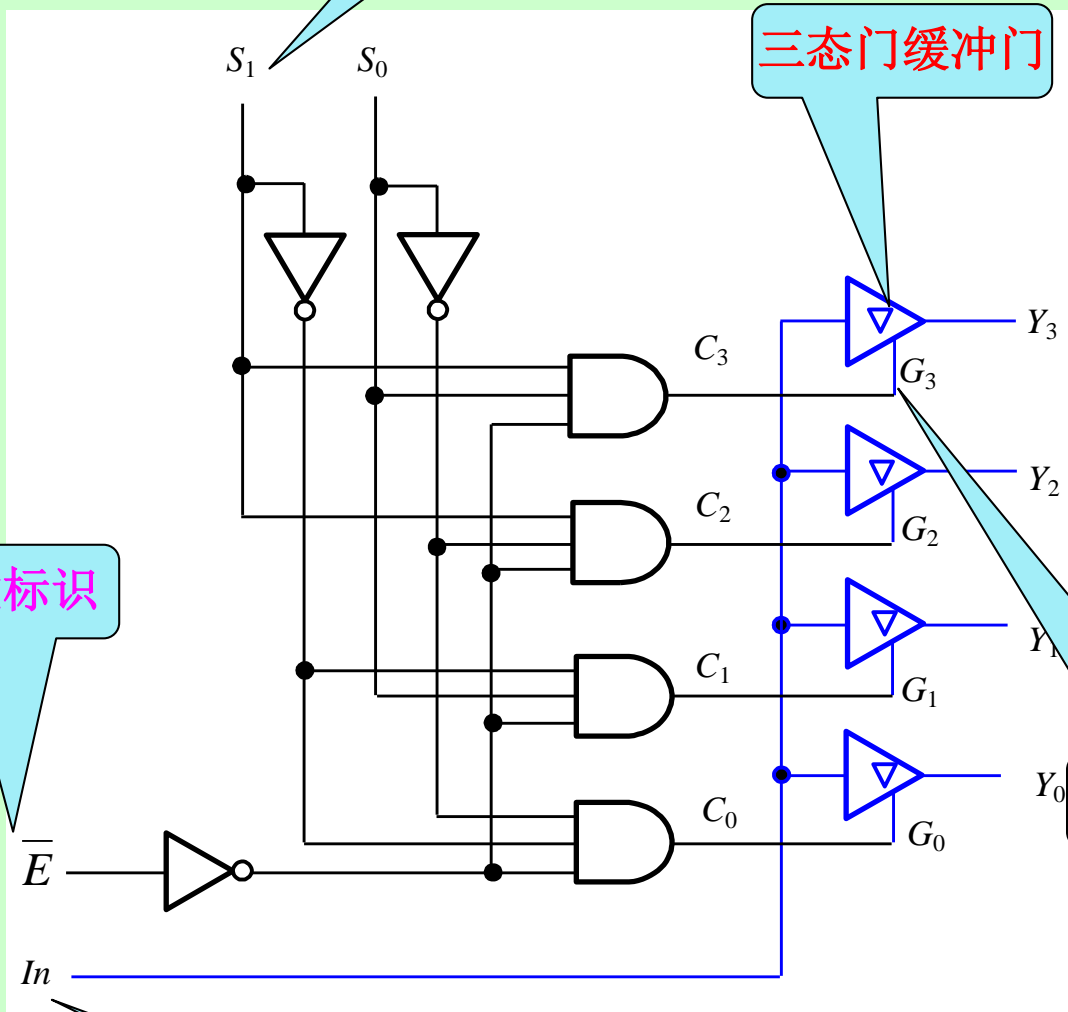
地址输入端

三态门缓冲门

低电平有效标识

三态门控制端

数据输入端

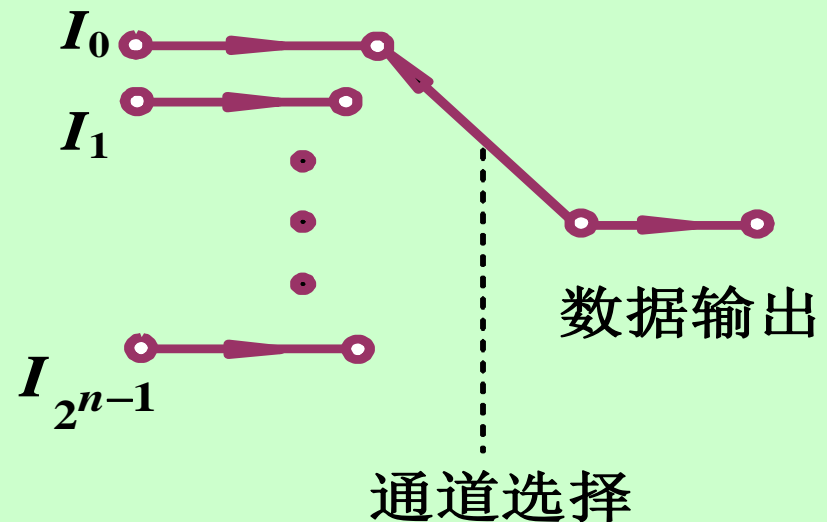


4.4.3 数据选择器 Multiplexers (Data Selectors)

1、数据选择器的定义与功能

数据选择器： 能实现数据选择功能的逻辑电路。它的作用相当于多个输入的单刀多掷开关，又称“多路开关”。

数据选择的功能： 在通道选择信号的作用下，将多个通道的数据分时传送到公共的数据通道上去的。



2选1数据选择器

简化真值表

完整的真值表

1位地址码
输入端

数据
输入
端

1路数据输出端

Y=1的情况共有两种情况：

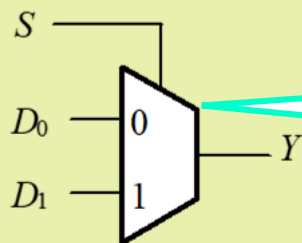
(1) $SD_0=01$

(2) $SD_1=11$

$$Y = \overline{S}D_0 + SD_1$$

选择 输入	输 出
S	Y
0	D_0
1	D_1

输入			输出
S	D_0	D_1	Y
0	1	x	1
0	0	x	0
1	x	1	1
1	x	0	0
...			



逻辑符号

4选1数据选择器

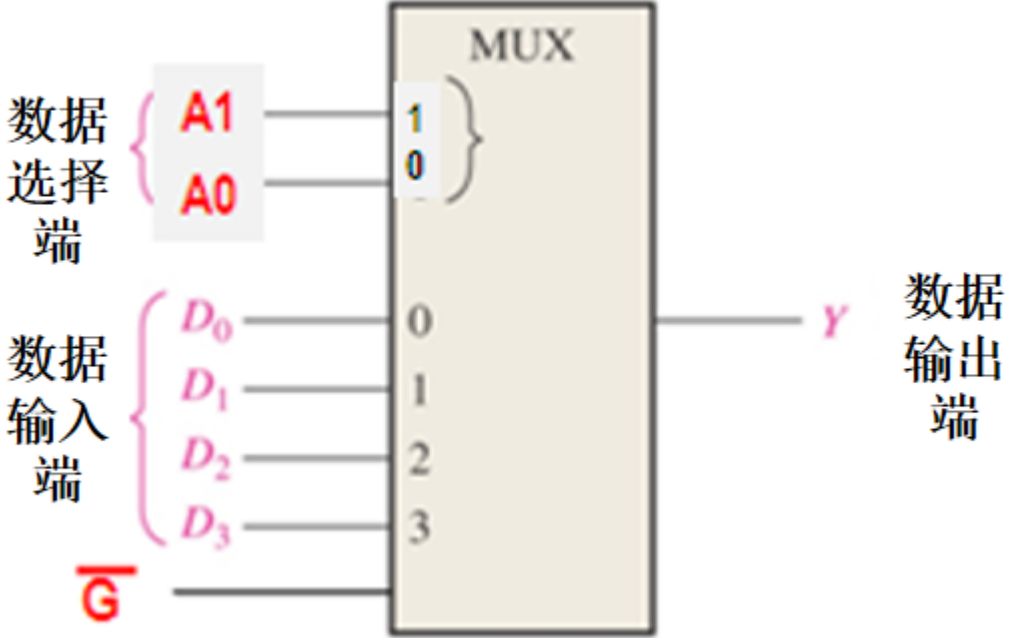
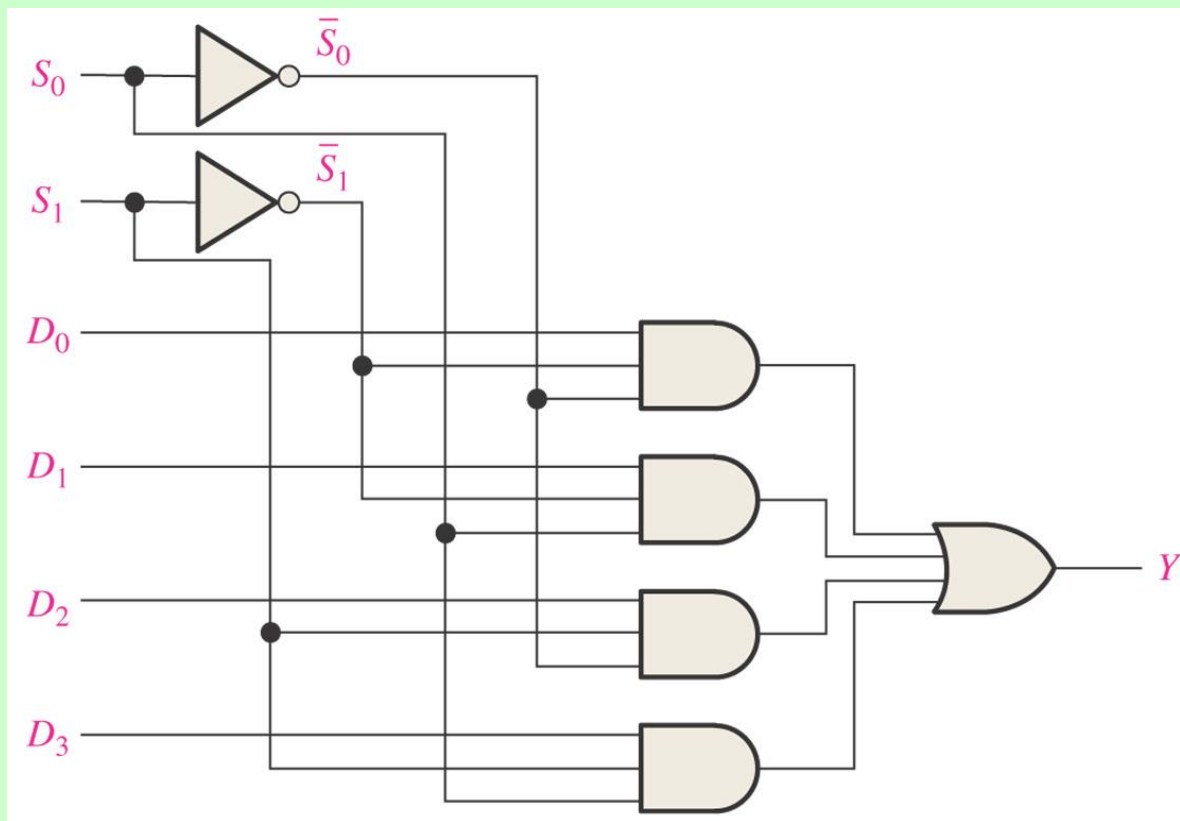


表 4.3.1 4 选 1 数据选择器功能表

输 入				输 出			
G	A_1	A_0	D_3	D_2	D_1	D_0	Y
1	×	×	×	×	×	×	0
0	0	0	×	×	×	0	0
			×	×	×	1	1
	0	1	×	×	0	×	0
			×	×	1	×	1
	1	0	×	0	×	×	0
			×	1	×	×	1
	1	1	0	×	×	×	0
			1	×	×	×	1

4选1数据选择器



$$Y = \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_1 S_0 D_1 + S_1 \bar{S}_0 D_2 + S_1 S_0 D_3$$

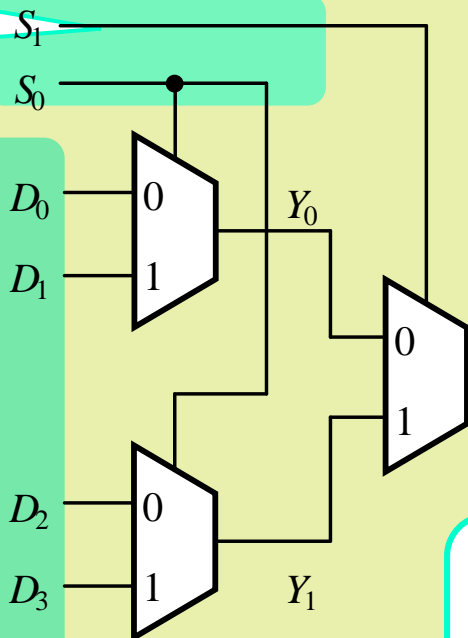
4选1数据选择器

(1) 逻辑电路

由3个2选1数据选择器构成4选1数据选择器。

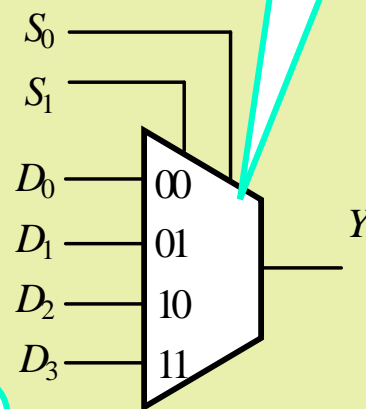
2 位地址
码输入端

数据
输入端

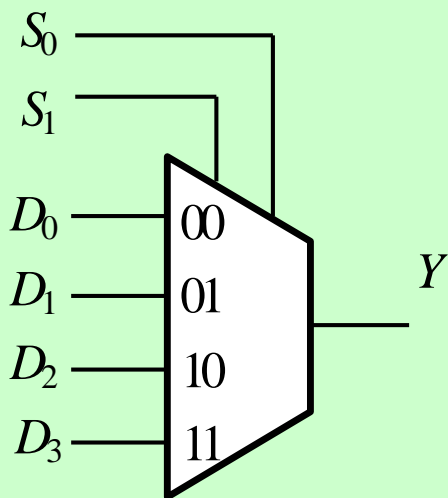


1路数据
输出端

逻辑符号



(2) 工作原理及逻辑功能



简化真值表

选择输入		输出
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = \overline{S_1}\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$

$$Y = D_0m_0 + D_1m_1 + D_2m_2 + D_3m_3$$

这两个公式常用

(3) 数据选择器实现逻辑函数

例4.4.8 试用数据选择器实现下列逻辑函数

① 用4选1数据选择器实现 $L_0 = \overline{A}B + A\overline{B}$

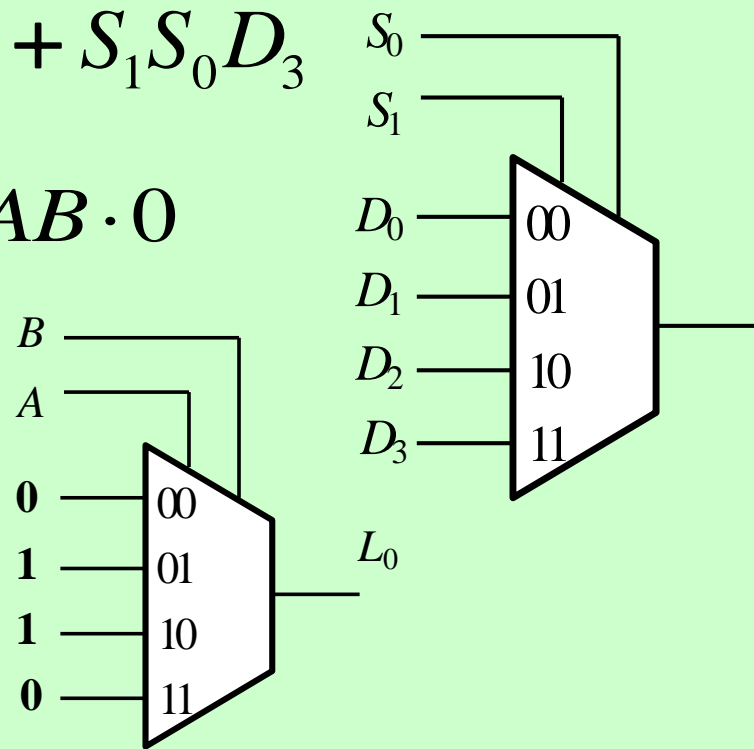
② 用2选1数据选择器和必要的逻辑门实现 $L_1 = AB + A\overline{C} + BC$

$$Y = \overline{S_1}\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$

$$L_0 = \overline{A}\overline{B} \cdot 0 + \overline{A}B \cdot 1 + A\overline{B} \cdot 1 + AB \cdot 0$$

① 当 $S_1 = A, S_0 = B,$

$$D_0 = D_3 = 0, D_1 = D_2 = 1$$



② 用2选1数据选择器和必要的逻辑门实现 $L_1 = AB + A\bar{C} + BC$

2选1数据选择器只有1个选通端接输入 A ，表达式有3个变量。因此数据端需要输入2个变量。考察真值表 B 、 C 与 L_1 的关系。

$$Y = \bar{S}D_0 + SD_1$$

当 $S=A$ 时，

$$\begin{aligned} L_1 &= AB + A\bar{C} + BC \cdot (A + \bar{A}) \\ &= A \cdot (B + \bar{C} + BC) + \bar{A} \cdot BC \\ &= A \cdot (B + \bar{C}) + \bar{A} \cdot BC \end{aligned}$$

令 $S=A$,

$D_0=BC$

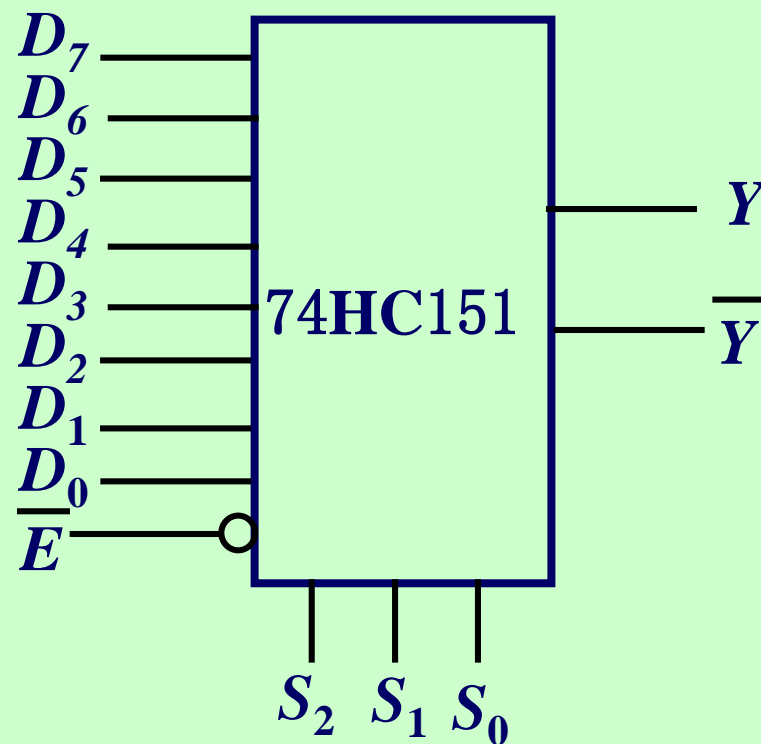
$D_1=B + \bar{C}$ 时，

$Y=L_1$

输入	输出
S	Y
0	$Y=D_0$
1	$Y=D_1$

(6) 集成电路数据选择器

8选1数据选择器74HC151



74HC151逻辑符号

74HC151的功能表

$$Y = D_0m_0 + D_1m_1 + D_2m_2 + D_3m_3 + D_4m_4 + D_5m_5 + D_6m_6 + D_7m_7$$

•当 $\overline{E}=1$ 时, $Y=0$ 。

•当 $\overline{E}=0$ 时

Y=1的情况共有八种情况, 可以写出对应表达式:

(1) $S_2S_1S_0D_0=0001$ (5) $S_2S_1S_0D_4=1001$

(2) $S_2S_1S_0D_1=0011$ (6) $S_2S_1S_0D_5=1011$

(3) $S_2S_1S_0D_2=0101$ (7) $S_2S_1S_0D_6=1101$

(4) $S_2S_1S_0D_3=0111$ (8) $S_2S_1S_0D_7=1111$

$$Y = \overline{S_2}\overline{S_1}\overline{S_0}D_0 + \overline{S_2}\overline{S_1}S_0D_1 + \overline{S_2}S_1\overline{S_0}D_2$$

$$+ \overline{S_2}S_1S_0D_3 + S_2\overline{S_1}\overline{S_0}D_4 + S_2\overline{S_1}S_0D_5$$

$$+ S_2S_1\overline{S_0}D_6 + S_2S_1S_0D_7$$

$$Y = \sum_{i=0}^7 D_i m_i$$

输 入				输 出	
使能 E	选 择			Y	\overline{Y}
	S ₂	S ₁	S ₀		
1	X	X	X	L	H
0	0	0	0	D ₀	$\overline{D_0}$
0	0	0	1	D ₁	$\overline{D_1}$
0	0	1	0	D ₂	$\overline{D_2}$
0	0	1	1	D ₃	$\overline{D_3}$
0	1	0	0	D ₄	$\overline{D_4}$
0	1	0	1	D ₅	$\overline{D_5}$
0	1	1	0	D ₆	$\overline{D_6}$
0	1	1	1	D ₇	$\overline{D_7}$

4.4.4 数值比较器

Comparators

本节了解

数值比较器：对两个1位数字进行比较（ A 、 B ），以判断其大小的逻辑电路。 Determine whether two numbers are equal

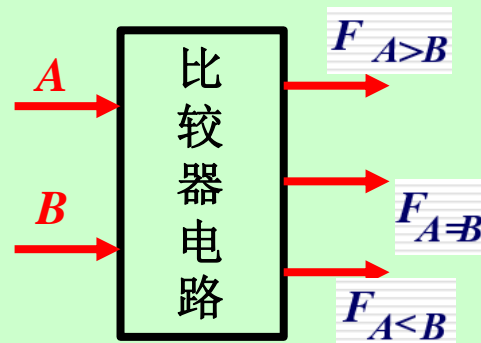
1. 1位数值比较器(设计)

输入：两个一位二进制数 A 、 B 。

输出： $F_{A>B}=1$ ，表示 A 大于 B

$F_{A<B}=1$ ，表示 A 小于 B

$F_{A=B}=1$ ，表示 A 等于 B



1, If $A=1$ and $B=0$, number A is greater than number B ;

2, If $A=0$ and $B=1$, number A is less than number B ;

3, If $A=B$, number A is equal to number B .

1位数值比较器

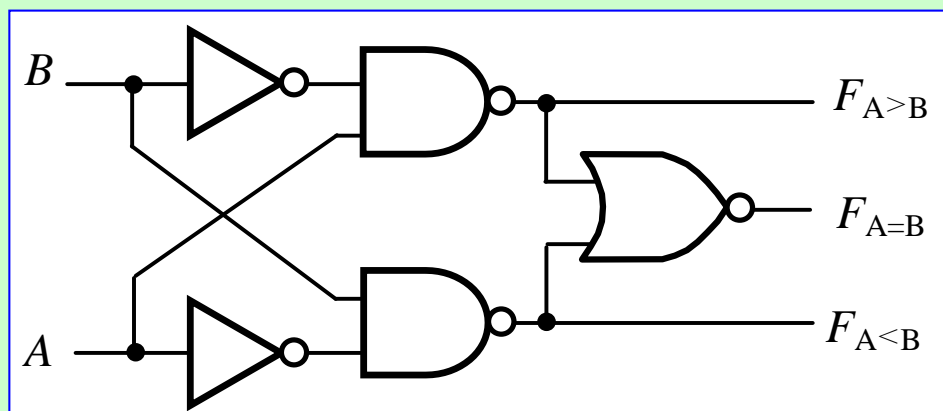
一位数值比较器真值表

输 入		输 出		
A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

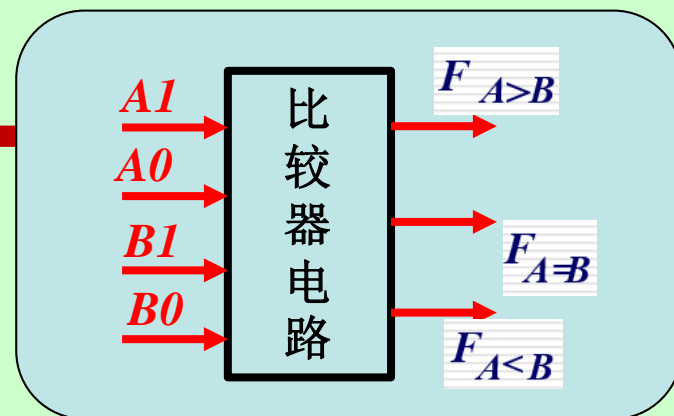
$$F_{A>B} = A \bar{B}$$

$$F_{A<B} = \bar{A} B$$

$$F_{A=B} = \bar{A} \bar{B} + AB$$



2、2 位数值比较器:



比较两个2 位二进制数的大小的电路

输入：两个2位二进制数 $A=A_1A_0$ 、 $B=B_1B_0$

能否用1位数值比较器设计两位数值比较器？

用一位数值比较器设计多位数值比较器的原则

当高位 (A_1 、 B_1) 不相等时，无需比较低位 (A_0 、 B_0)，高位比较的结果就是两个数的比较结果。

当高位相等时，两数的比较结果由低位比较的结果决定。

真值表

输 入				输 出		
A_1	B_1	A_0	B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_1 > B_1$		\times		1	0	0
$A_1 < B_1$		\times		0	1	0
$A_1 = B_1$		$A_0 > B_0$		1	0	0
$A_1 = B_1$		$A_0 < B_0$		0	1	0
$A_1 = B_1$		$A_0 = B_0$		0	0	1

$$F_{A>B} = (A_1 > B_1) + (A_1 = B_1)(A_0 > B_0)$$

$$F_{A<B} = (A_1 < B_1) + (A_1 = B_1)(A_0 < B_0)$$

$$F_{A=B} = (A_1 = B_1)(A_0 = B_0)$$

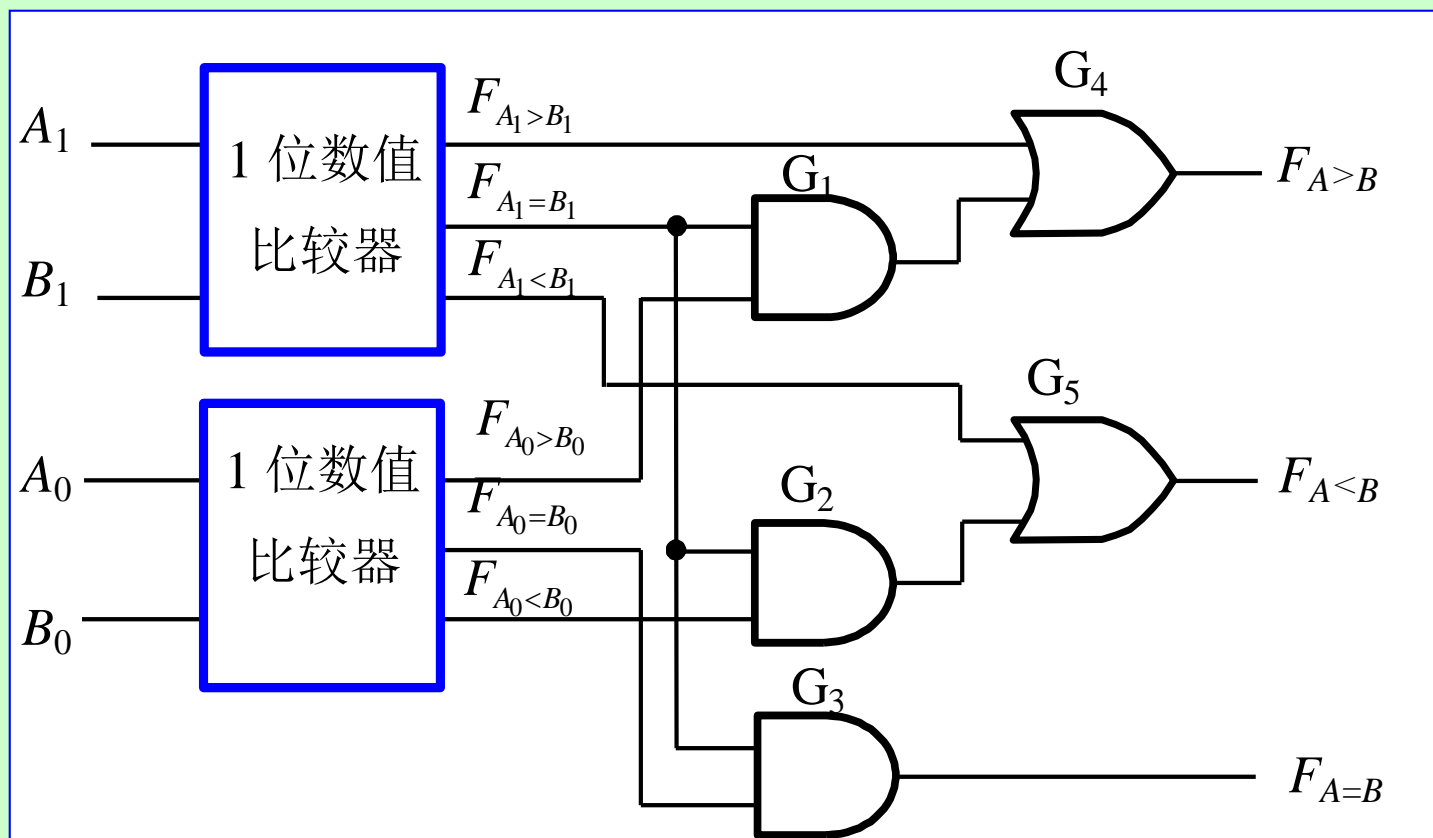
注意：上述不是真正的逻辑函数表达式，只示意逻辑关系。

$$F_{A>B} = (A_1 > B_1) + (A_1 = B_1)(A_0 > B_0)$$

$$F_{A=B} = (A_1 = B_1)(A_0 = B_0)$$

$$F_{A<B} = (A_1 < B_1) + (A_1 = B_1)(A_0 < B_0)$$

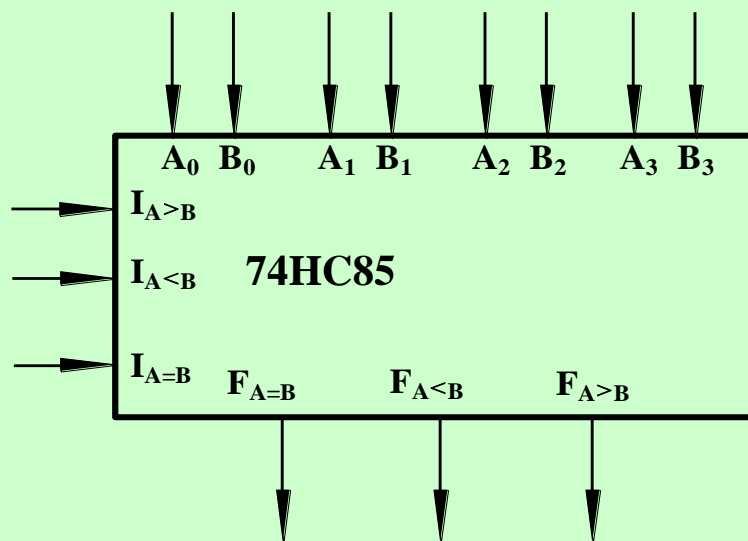
两位数值比较器逻辑图



3、集成数值比较器

(1.) 集成数值比较器74HC85的功能

74HC85是四位数值比较器，其工作原理和两位数值比较器相同。



74HC85的示意框图