数字逻辑电路

(2020级本科生课程)

清华大学计算机系

陶品

taopin@tsinghua.edu.cn

办公室：FIT 3－531 （13717813059）

第三章　组合逻辑电路

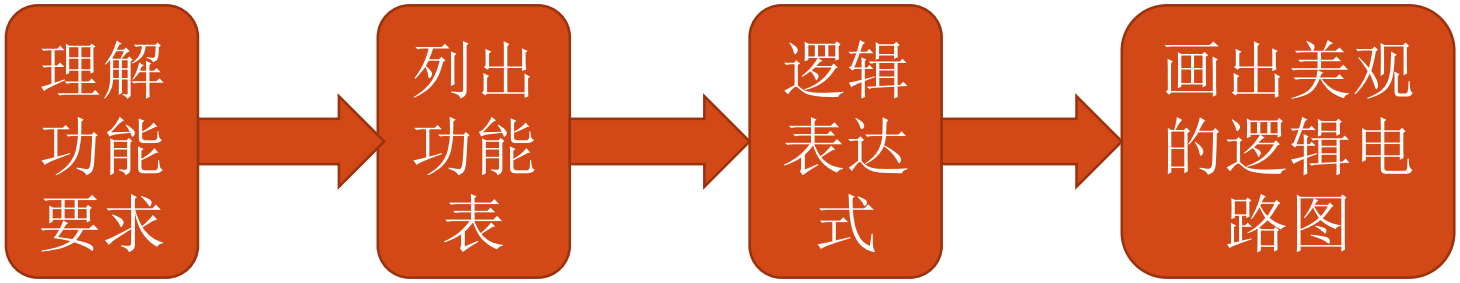
Combinational Logic Circuit

第三章　组合逻辑电路

组合逻辑电路的分析与设计

组合逻辑电路分析

组合逻辑电路设计



3.3 常用的中规模组合逻辑电路

⇨ 3.3.1 译码器

3.3.2 数据选择器

3.3.3 编码器

3.3.4 数据比较器

3.3.5 奇偶校验器

3.3.6 运算器（算数逻辑单元 ALU）

3.3.1 译码器（1）

- 译码器的功能分类：
  - 变量译码器：用来表示输入变量状态的全部组合

    N位输入，2^N输出，

    常见的集成化译码器有2-4、3-8、4-16
  - 码制译码器：如8421码变换为循环码等
  - 显示译码器：控制数码管显示

3.3.1 译码器（2）

●2-4变量译码器

➢（步骤一）定义：2－4译码器是指2输入－4输出的变量译码器。2输入,4输出.对应输入的每一种组合，唯一只有一个输出为“0”.

真值表

输　入		输　出			
A	B	Y_0	Y_1	Y_2	Y_3
0	0	0	1	1	1
1	0	1	0	1	1
0	1	1	1	0	1
1	1	1	1	1	0

3.3.1 译码器（3）

2-4译码器

（步骤二）根据真值表写出输出表达式

真值表

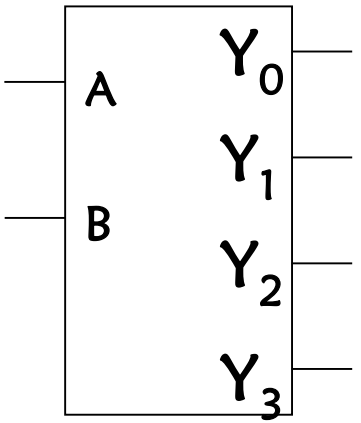
输　入		输　出			
A	B	Y_0	Y_1	Y_2	Y_3
0	0	0	1	1	1
1	0	1	0	1	1
0	1	1	1	0	1
1	1	1	1	1	0

输出表达式

$$\begin{cases} Y_0 = \overline{\overline{AB}} \\ Y_1 = \overline{A\overline{B}} \\ Y_2 = \overline{\overline{A}B} \\ Y_3 = \overline{AB} \end{cases}$$

只用与非门实现

逻辑示意图



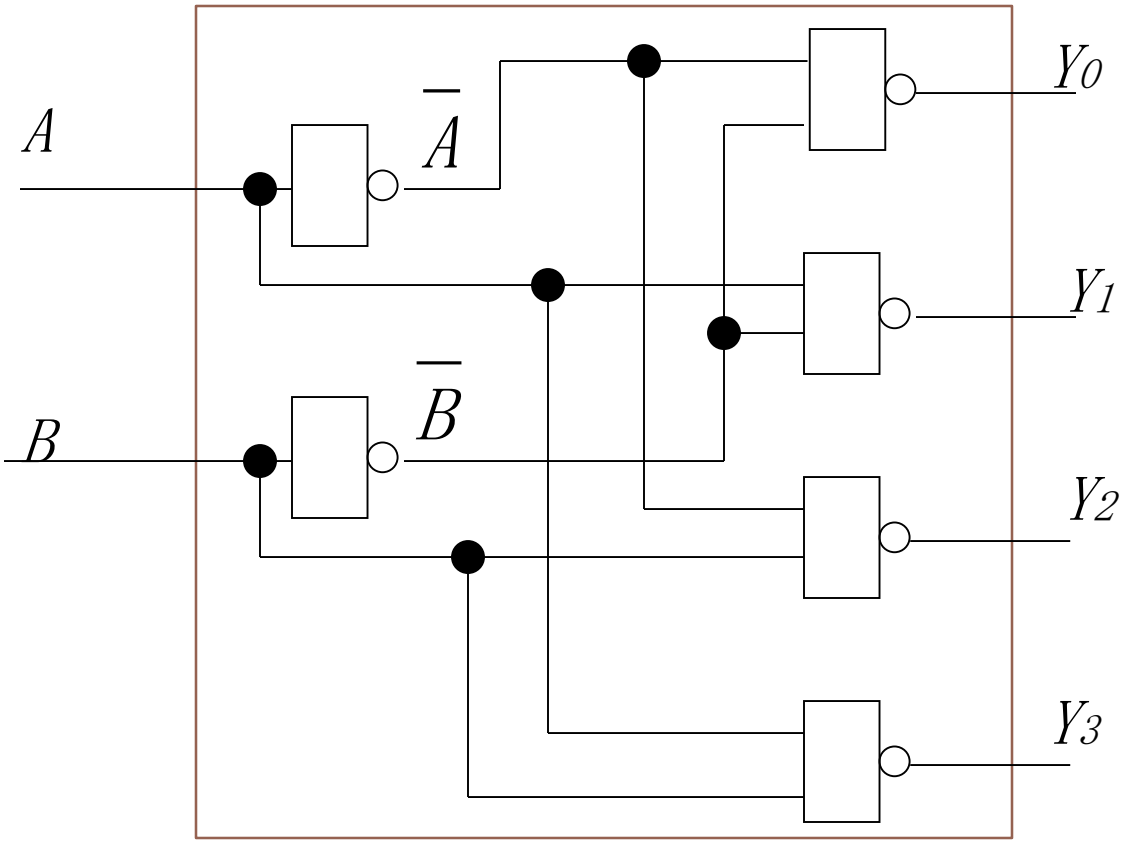
3.3.1 译码器（4）

2-4译码器

（步骤三）按照输出表达式画出逻辑图

输出表达式

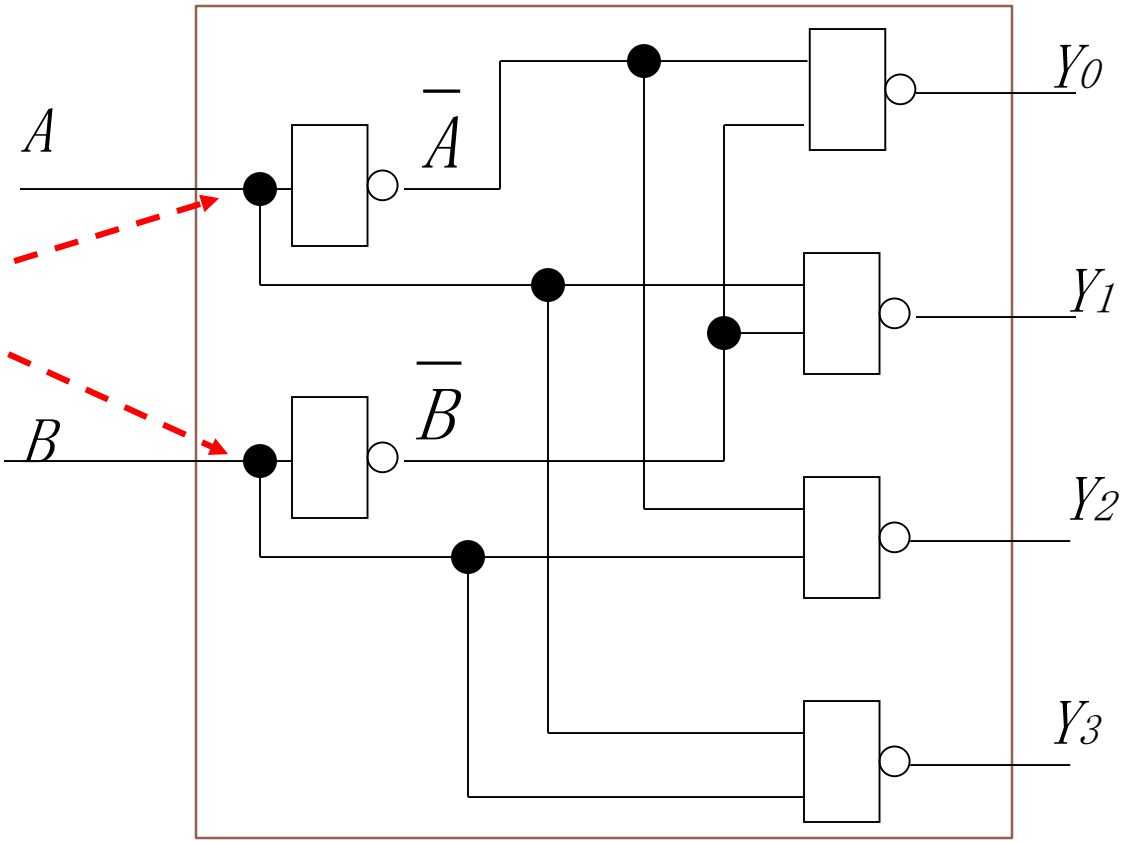
$$\begin{cases} Y_0 = \overline{\overline{A}\,\overline{B}} \\ Y_1 = \overline{A\overline{B}} \\ Y_2 = \overline{\overline{A}B} \\ Y_3 = \overline{AB} \end{cases}$$



有没有什么问题？

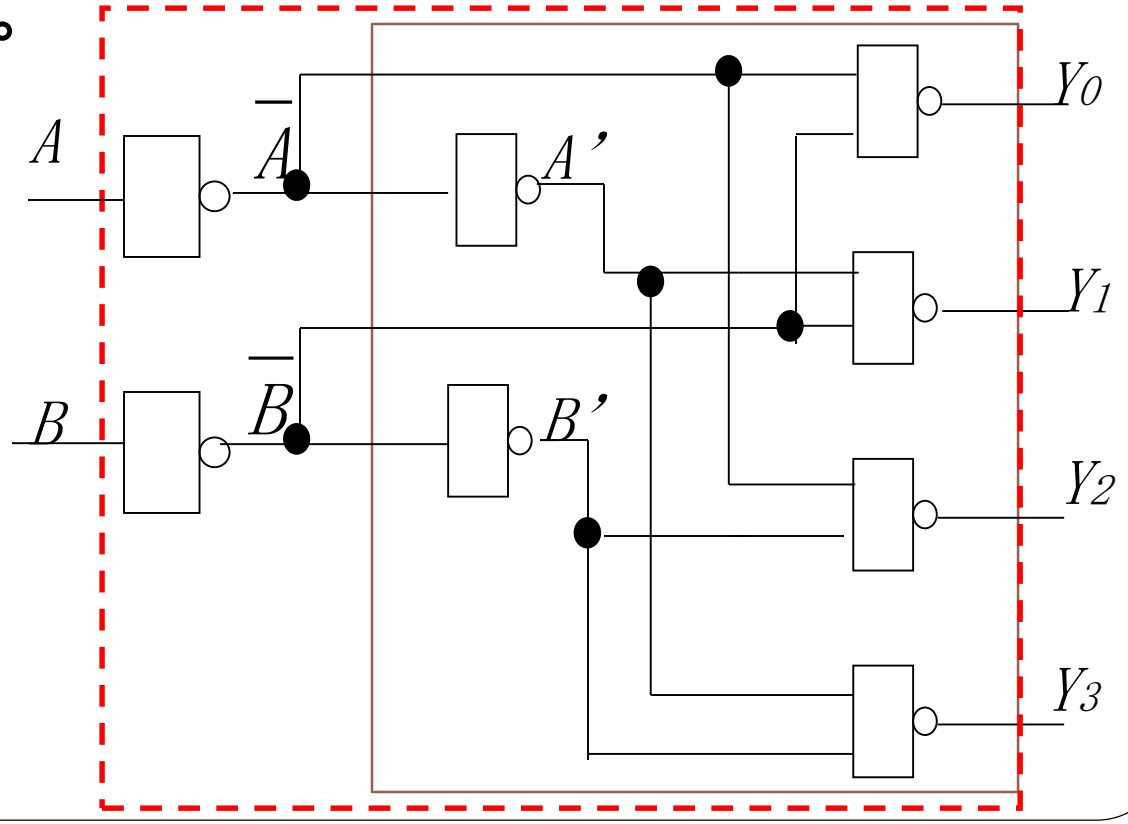
3.3.1 译码器（5）

问题：一个输入有3个负载！



3.3.1 译码器（6）

- 2-4译码器
  - （步骤四）检查可能出现的问题，并修正设计
  - 集成电路的设计原则：一个输入只能按照一个输入负载计算。

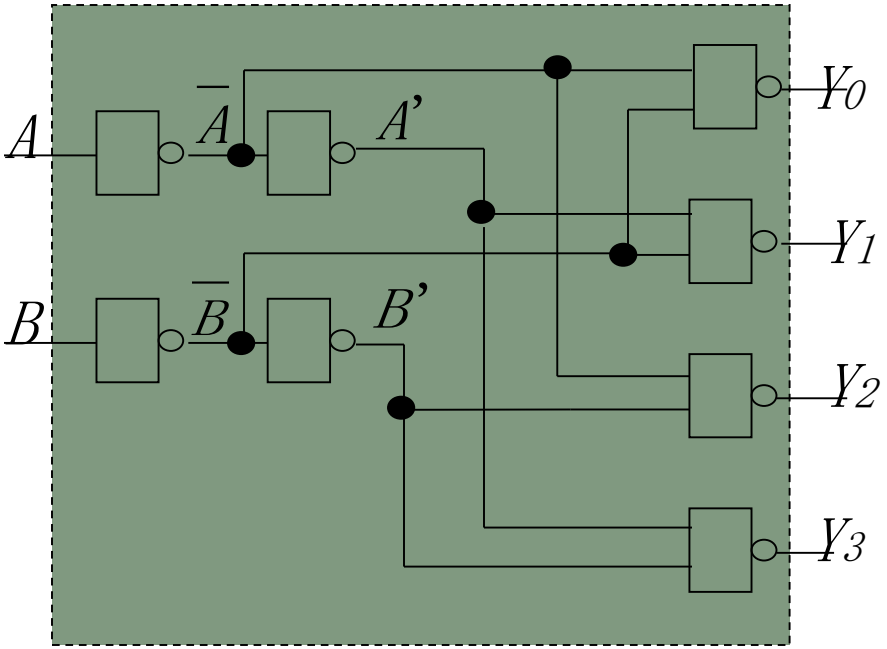


解决办法：增加一级输入缓冲

3.3.1 译码器（7）

- 2-4译码器
  - 集成电路由两部分组成：**输入缓冲部分和译码部分。**

输入缓冲部分使得对外负载只有一个，减轻上一级电路的负担。



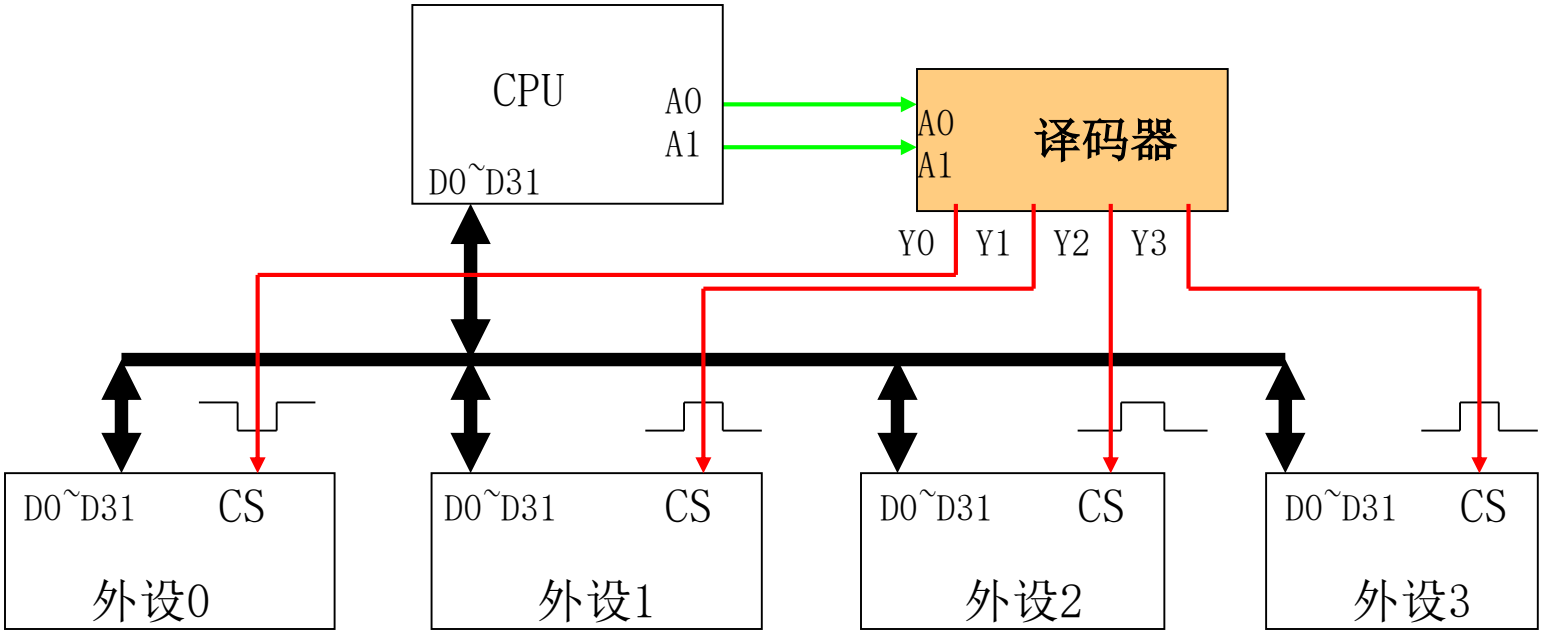
输入缓冲电路　译码逻辑电路

3.3.1 译码器（8）

- 画逻辑图要求：
  - 逻辑图要美观，可读性要好
- 具体注意几个问题：
  - 逻辑图中逻辑门（或逻辑器件）布局要合理，逻辑性强
  - 逻辑图中的连线布局合理，无连接交叉点要少
  - 相接连线的交叉点要画上连接符

3.3.1 译码器（9）

- 2-4译码器的应用举例:CPU控制四个设备



功能级设计要求：

A0=0，A1=0时，外设0工作

A0=1，A1=0时，外设1工作

A0=0，A1=1时，外设2工作

A0=1，A1=1时，外设3工作

信号级设计要求：

A0=0，A1=0时，Y0=0，Y1,Y2,Y3=1

A0=1，A1=0时，Y1=0，Y0,Y2,Y3=1

A0=0，A1=1时，Y2=0，Y0,Y1,Y3=1

A0=1，A1=1时，Y3=0，Y0,Y1,Y2=1

3.3.1 译码器（10）

- 有使能端的2-4译码器
  - 由于2－4译码器的4个输出是2输入的逻辑组合，任何一种组合都会有一个输出有效
  - 要使所有输出无效（输出为高），就需要增加附加逻辑——使能（Enable）

3.3.1 译码器（11）

- 有使能端 $\overline{E}$ 的2-4译码器
  - 在普通的2－4译码器中设置使能端（Enable）

当 $\overline{E}$=0，译码器使能

当 $\overline{E}$=1，译码器禁止

功能表

$\overline{E}$	A	B	Y_0	Y_1	Y_2	Y_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	1	0	1	0	1	1
0	0	1	1	1	0	1
0	1	1	1	1	1	0

3.3.1 译码器（12）

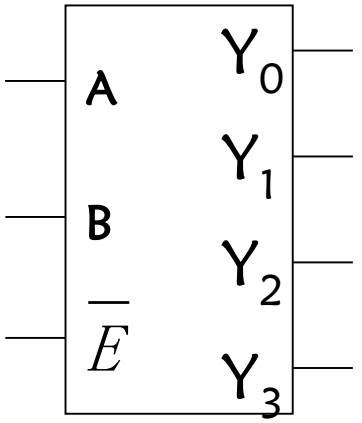
- 有使能端 $\overline{E}$ 的2-4译码器

功能表

$\overline{E}$ A B	Y_0 Y_1 Y_2 Y_3
1 X X	1 1 1 1
0 0 0	0 1 1 1
0 1 0	1 0 1 1
0 0 1	1 1 0 1
0 1 1	1 1 1 0

$$\begin{cases} Y_0 = \overline{\overline{E}\overline{A}\overline{B}} \\ Y_1 = \overline{\overline{E}\overline{A}B} \\ Y_2 = \overline{\overline{E}A\overline{B}} \\ Y_3 = \overline{\overline{E}AB} \end{cases}$$

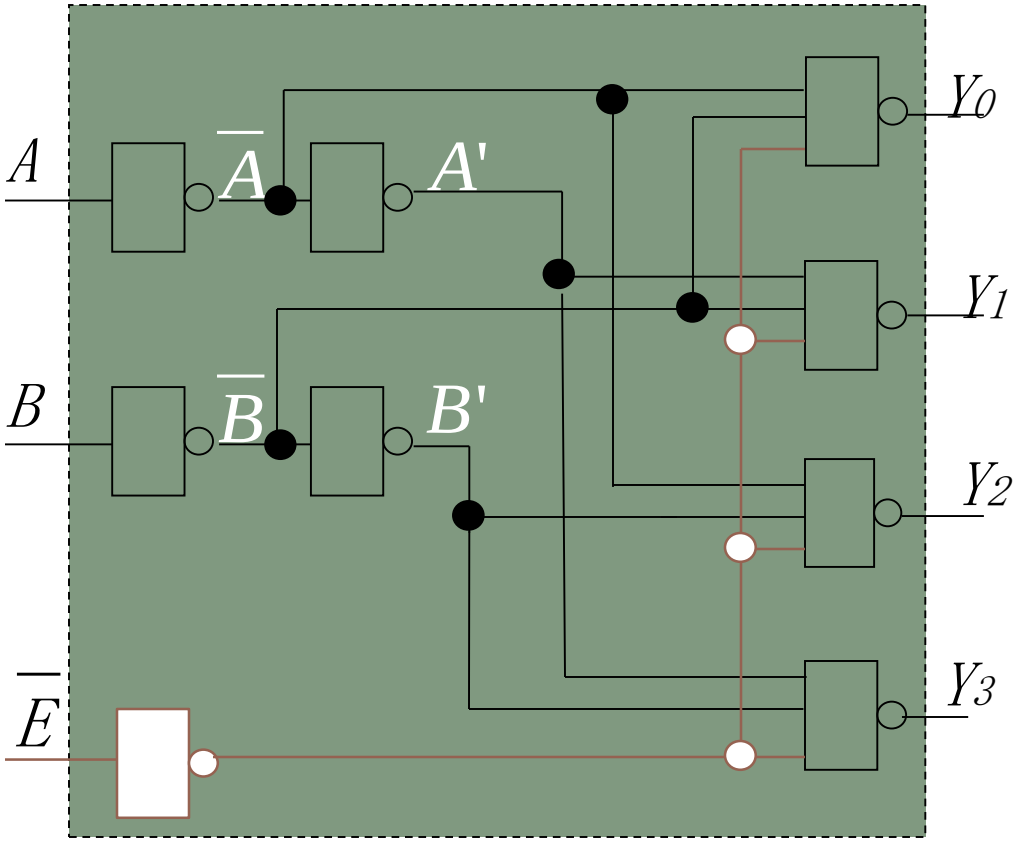
逻辑示意图



3.3.1 译码器（13）

- 有使能端$\overline{E}$的2-4译码器
  - 根据表达式画出逻辑图

$$\begin{cases} Y_0 = \overline{\overline{E}\overline{A}\overline{B}} \\ Y_1 = \overline{\overline{E}\overline{A}B} \\ Y_2 = \overline{\overline{E}A\overline{B}} \\ Y_3 = \overline{\overline{E}AB} \end{cases}$$



3.3.1 译码器（14）

- 译码器使能端 $\overline{E}$ 的作用
  - 在集成电路中增加控制使能(Enable)端$\overline{E}$，是电路设计中常用的技术，使得集成电路更加灵活、可靠。
  - 灵活：用于扩展
  - 可靠：用于选通

3.3.1 译码器（15）

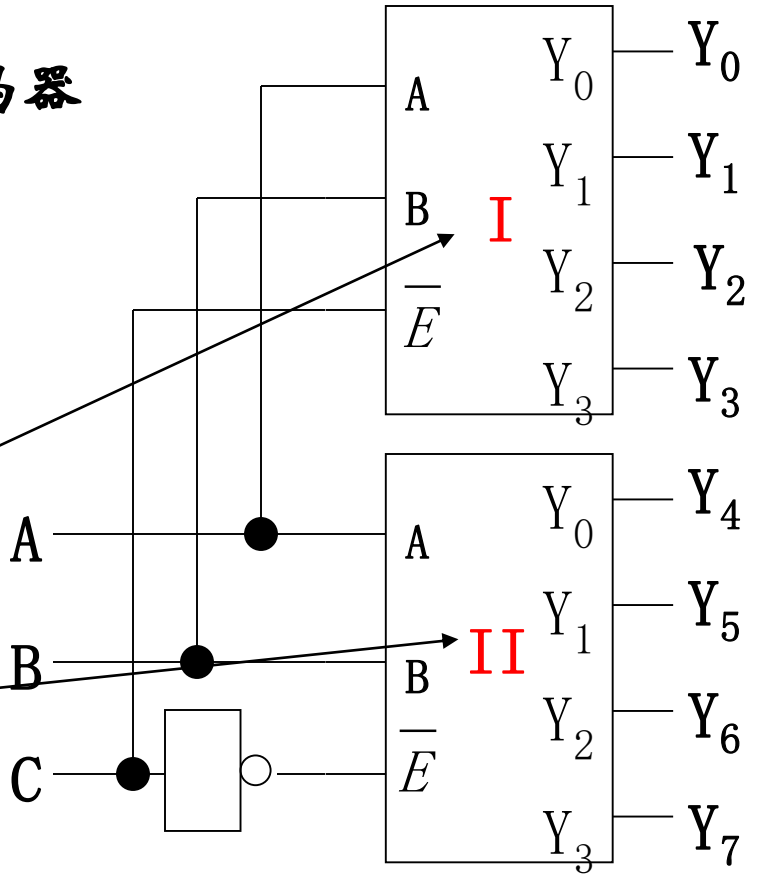
- 译码器使能端$\overline{E}$的作用
  - 用于多片扩展（作用一）：

例：用两片带使能端的2-4译码器组成3-8译码器

高位输入C用作选片，A、B用于选中片内译码。

C＝0选中片I，

C＝1选中片II。

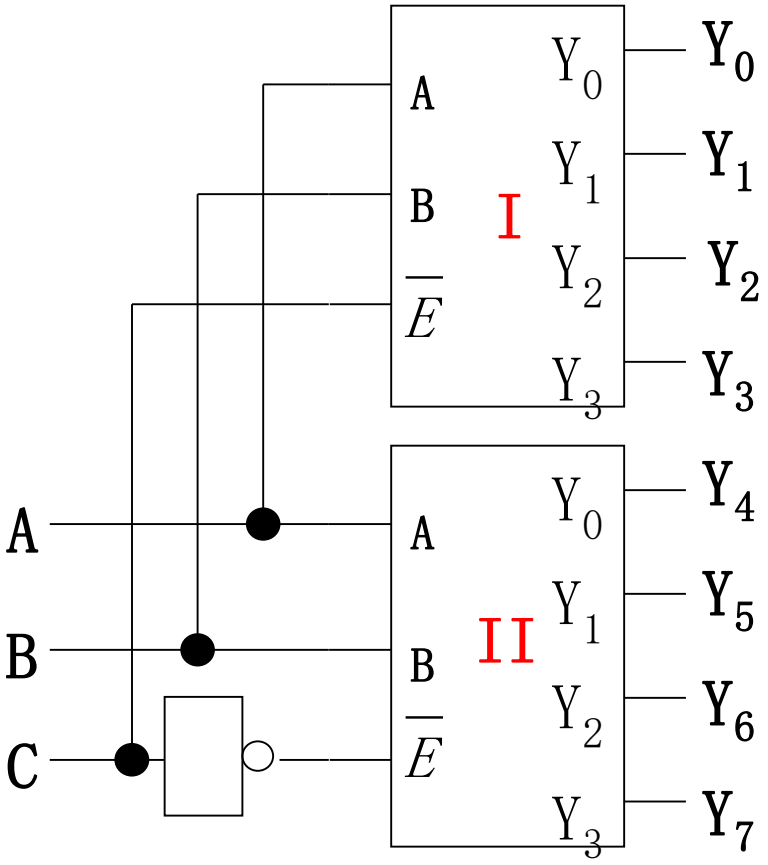


3.3.1 译码器（16）

- 用两片2-4译码器组成3-8译码器的真值表

真　值　表

输　入	输　出
A B C	Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7
0 0 0	0 1 1 1 1 1 1 1
1 0 0	1 0 1 1 1 1 1 1
0 1 0	1 1 0 1 1 1 1 1
1 1 0	1 1 1 0 1 1 1 1
0 0 1	1 1 1 1 0 1 1 1
1 0 1	1 1 1 1 1 0 1 1
0 1 1	1 1 1 1 1 1 0 1
1 1 1	1 1 1 1 1 1 1 0



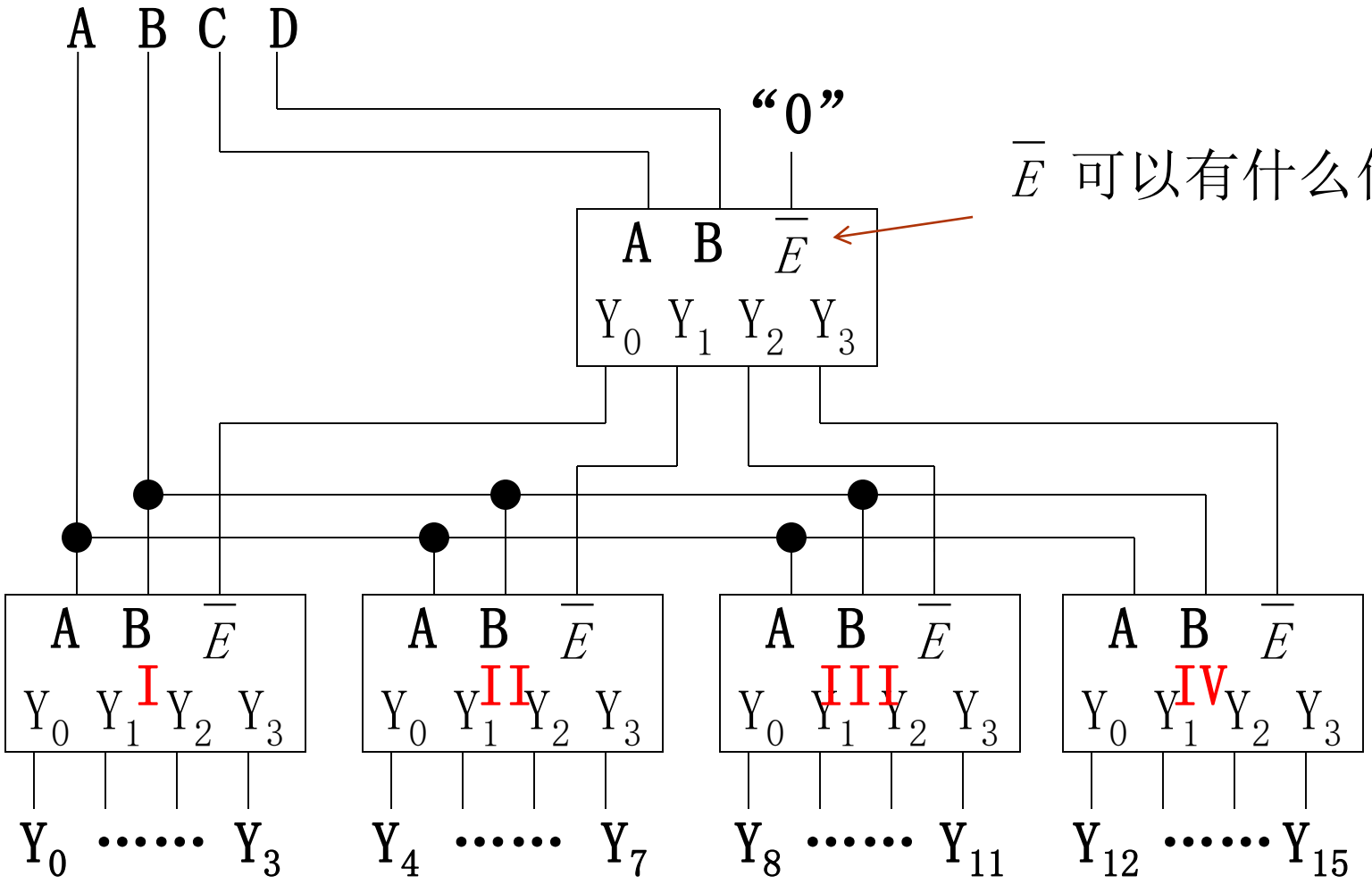
3.3.1 译码器（17）

- 用2－4译码器构成4－16译码器
  - 需要使用 ？片2－4译码器

答案：5片

问题：32－2^{32}译码器，如何搭建？

3.3.1 译码器（18）



$\overline{E}$ 可以有什么作用？

3.3.1 译码器（19）

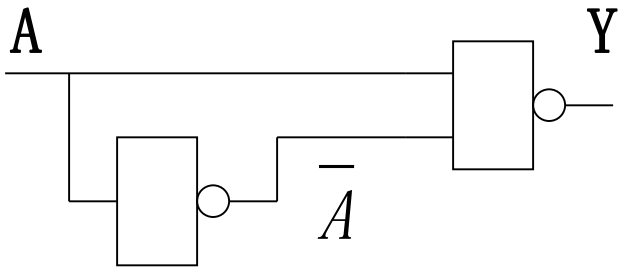
- $\overline{E}$ 用作选通（作用二）
  - 为什么需要选通？

    针对门电路的传输延迟造成的竞争、冒险问题提出的。

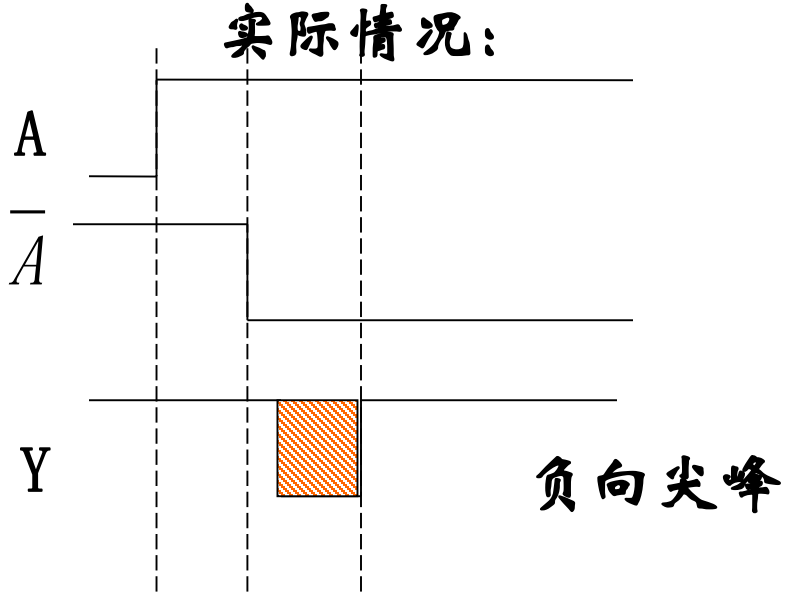
3.3.1 译码器（20）

- 门电路的传输延迟造成会竞争、冒险

二输入与非门的输入为 A 和$\overline{A}$ 时，$\overline{A}$ 滞后于A ，则Y会出现尖峰信号（与非门上升沿有尖峰）



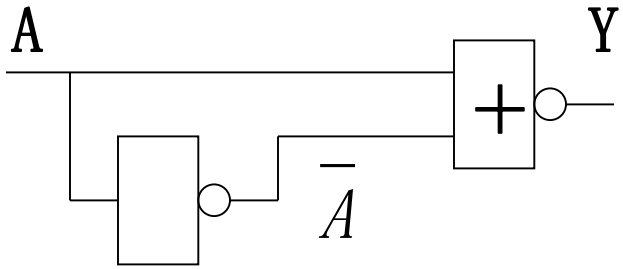
理想情况：$Y = \overline{A \bullet \overline{A}} = "1"$



3.3.1 译码器（21）

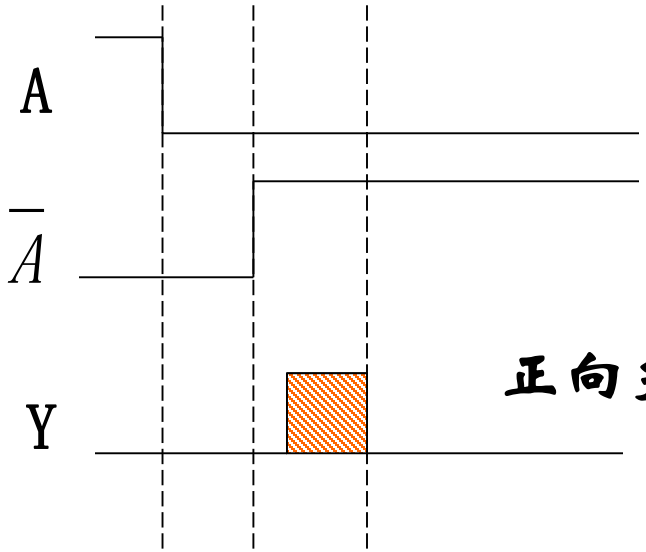
- 门电路的传输延迟造成会竞争、冒险

二输入或非门的输入为 A 和 $\overline{A}$ 时，$\overline{A}$ 滞后于A，则Y会出现尖峰信号（或非门下降沿有尖峰）



理想情况：$Y = \overline{A + \overline{A}} = "0"$

实际情况：



3.3.1 译码器（22）

- $\overline{E}$端用于选通

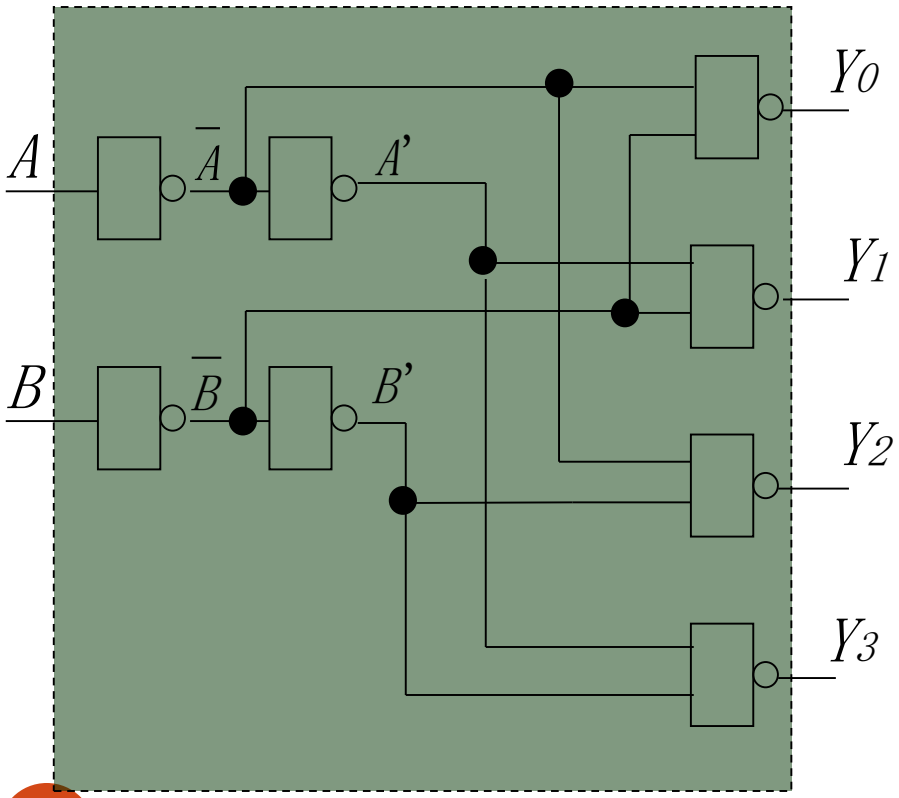
  2－4译码器中设置二级缓冲，目的是均衡负载，

  但是由于信号传输的延迟，会在输出端产生“0”重叠(Overlap)和尖峰信号(有些书中称为毛刺，英文词为：Spike, Glitch)。

  为消除尖峰和重叠，增加了 $\overline{E}$

3.3.1 译码器（23）

- 若无使能端，延迟产生尖峰和零重叠问题
  - 若A B同时到来：从11变到00

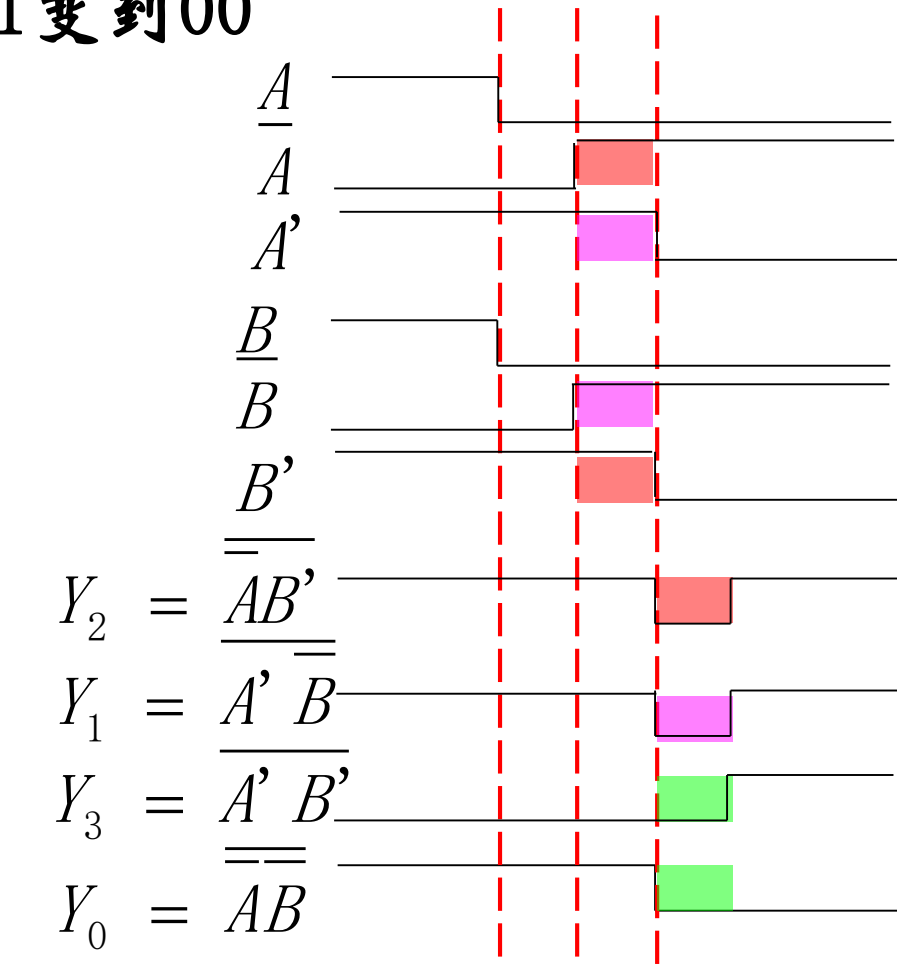


$$Y_2 = \overline{AB'}$$

$$Y_1 = \overline{A'\,B}$$

$$Y_3 = \overline{A'\,B'}$$

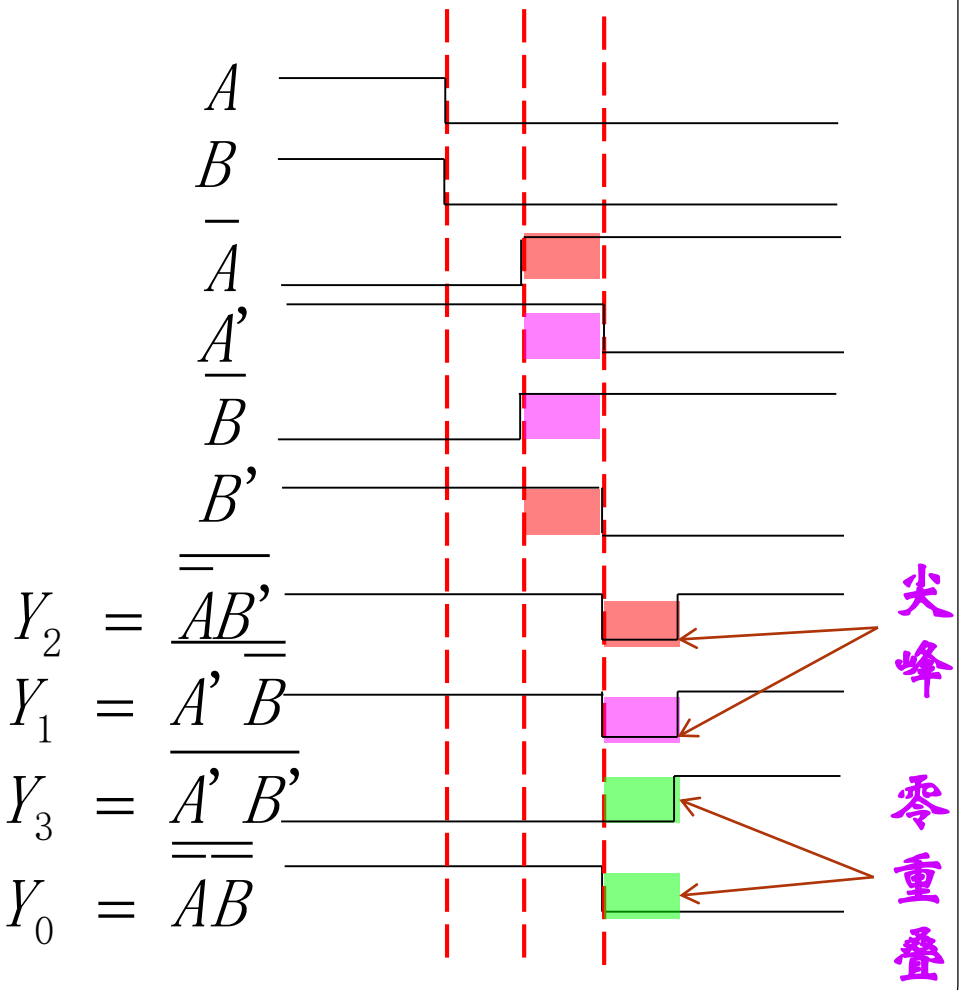
$$Y_0 = \overline{\overline{AB}}$$



3.3.1 译码器（24）

若A B同时到来(无偏移Skew)。从功能表上分析，A B从“11”变到“00”时，输出应从Y_3=0变成Y_0=0，Y_1Y_2保持为“1”。

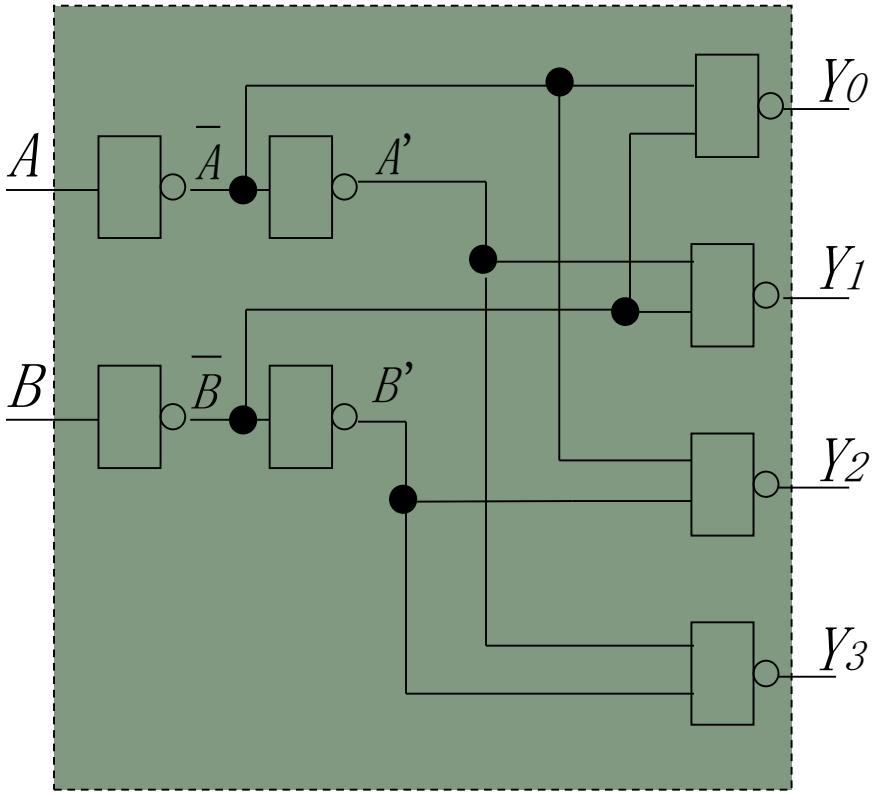
但是，由于门的传输延迟，造成Y_1,Y_2上出现了尖峰，同时，Y_3,Y_0有一段时间同时为“0”，即零重叠。



3.3.1 译码器（25）

- 当A B从“11”变到“00”时，输出应从Y_3=0变成Y_0=0。

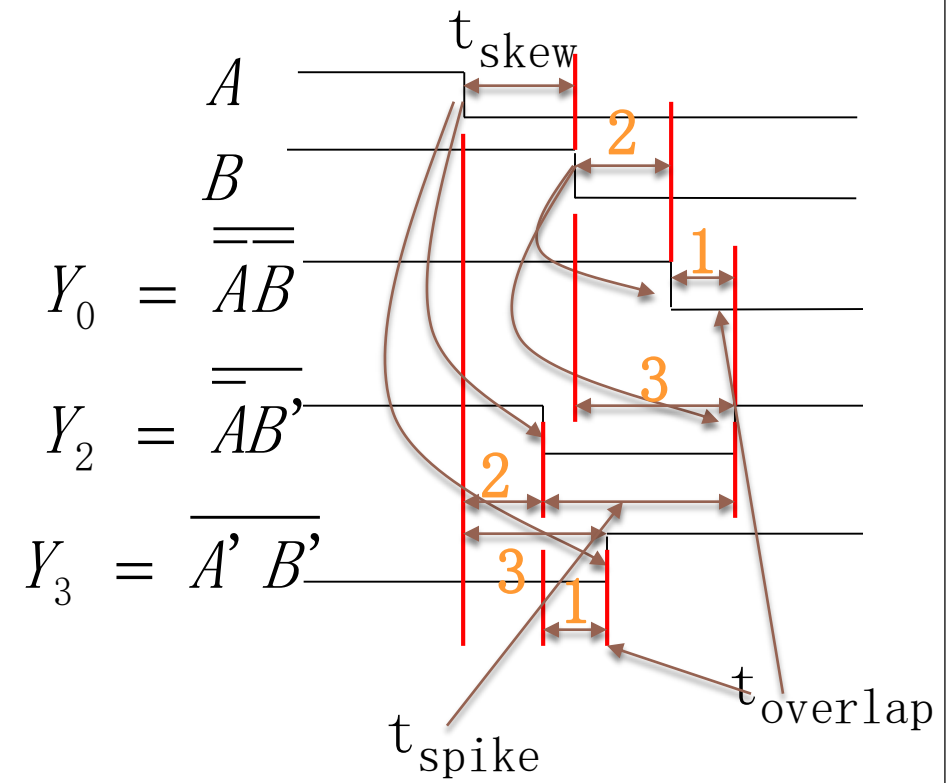
假设A B不能同时到来，存在偏移(Skew)，导致尖峰信号更宽。



$$Y_0 = \overline{\overline{AB}}$$

$$Y_2 = \overline{\overline{AB'}}$$

$$Y_3 = \overline{A'\,B'}$$



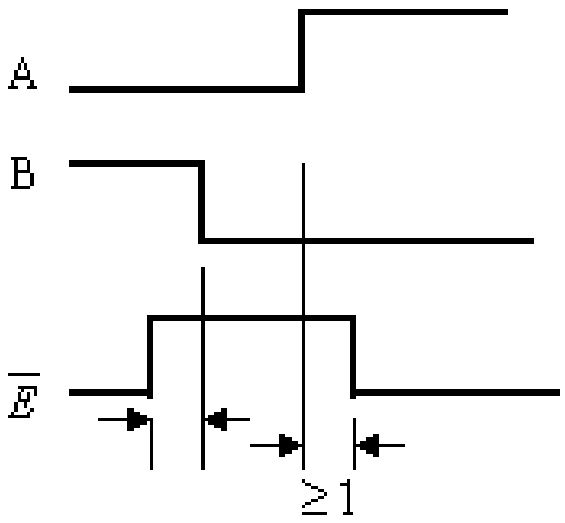
t_{spike}加宽、两处出现零重叠

$t_{overlap}$=1级延迟

$t_{spike} = t_{skew}$+1级延迟

3.3.1 译码器（26）

- 用使能端可以消除延迟产生尖峰和零重叠
  - 在A B变化期间，输出是不稳定的，可能会出现尖峰信号。加一个能覆盖输入变化的正脉冲（$\overline{E}=1$），使得A B变化期间强制 Y_0-Y_3=1，即可消除输出端的干扰。

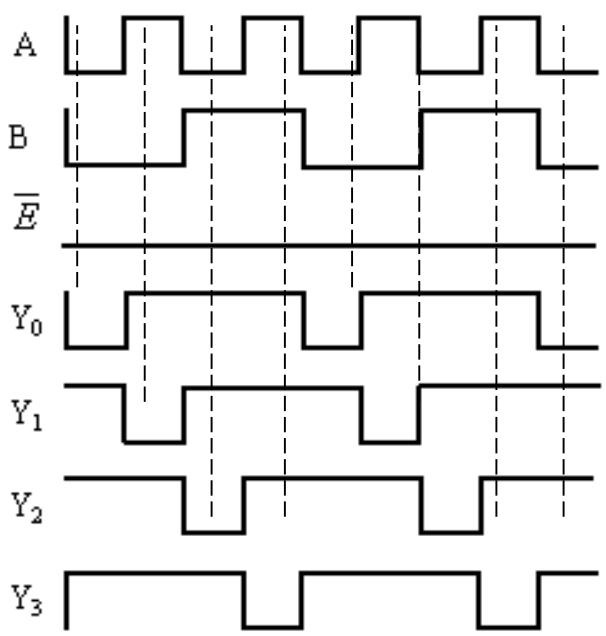


抑制尖峰和零重叠的使能正信号应先于（或同时）译码器的变量输入变化前到来，正信号撤除应滞后于变量输入的变化(至少滞后1级缓冲的延迟)。
但也不能太宽，否则速度会慢。

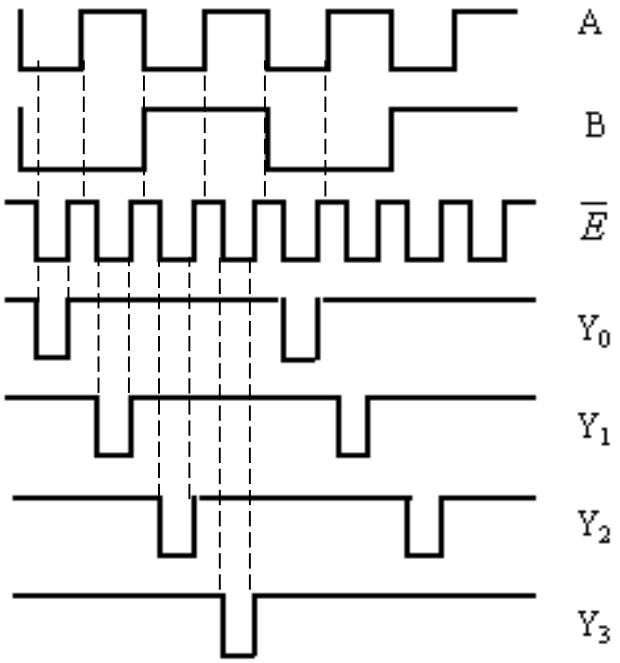
3.3.1 译码器（27）

使用 $\overline{E}$ 来抑制零重叠和尖峰，译码器的输出波形变窄了。

不使用 $\overline{E}$



使用 $\overline{E}$

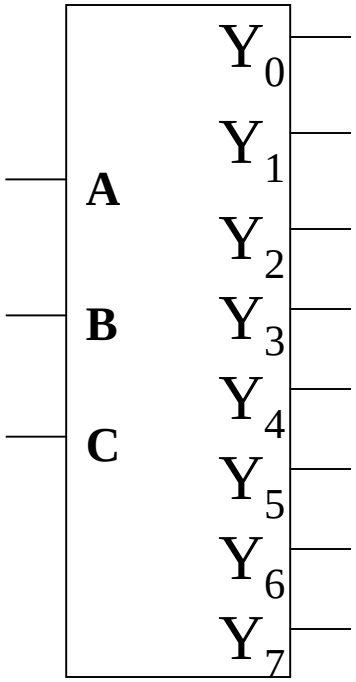


3.3.1 译码器（2/8）

- 3-8译码器
  - 定义：3－8译码器是指3输入－8输出的变量译码器。
  - 逻辑示意图

3－8译码器
逻辑示意图

C为最高位，
A为最低位



3.3.1 译码器（29）

3-8译码器

真值表和逻辑表达式

真　值　表

输　入	输　出
A B C	Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7
0 0 0	0 1 1 1 1 1 1 1
1 0 0	1 0 1 1 1 1 1 1
0 1 0	1 1 0 1 1 1 1 1
1 1 0	1 1 1 0 1 1 1 1
0 0 1	1 1 1 1 0 1 1 1
1 0 1	1 1 1 1 1 0 1 1
0 1 1	1 1 1 1 1 1 0 1
1 1 1	1 1 1 1 1 1 1 0

输出表达式

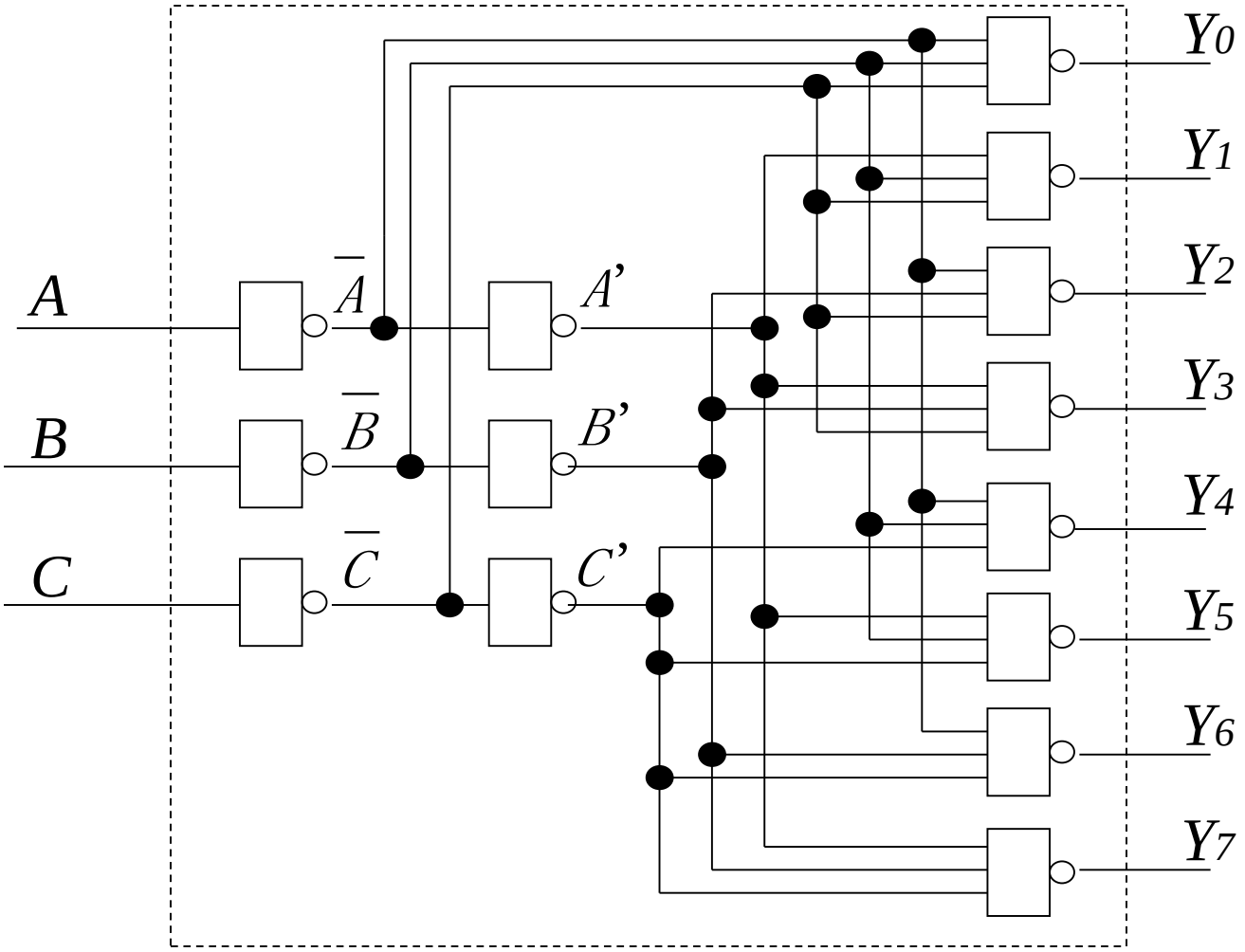
$$\begin{cases} Y_0 = \overline{\overline{A}\overline{B}\overline{C}} \\ Y_1 = \overline{\overline{A}\overline{B}C} \\ Y_2 = \overline{\overline{A}B\overline{C}} \\ Y_3 = \overline{\overline{A}BC} \\ Y_4 = \overline{A\overline{B}\overline{C}} \\ Y_5 = \overline{A\overline{B}C} \\ Y_6 = \overline{AB\overline{C}} \\ Y_7 = \overline{ABC} \end{cases}$$

只用非实的出达与门现输表式

3.3.1 译码器（3O）

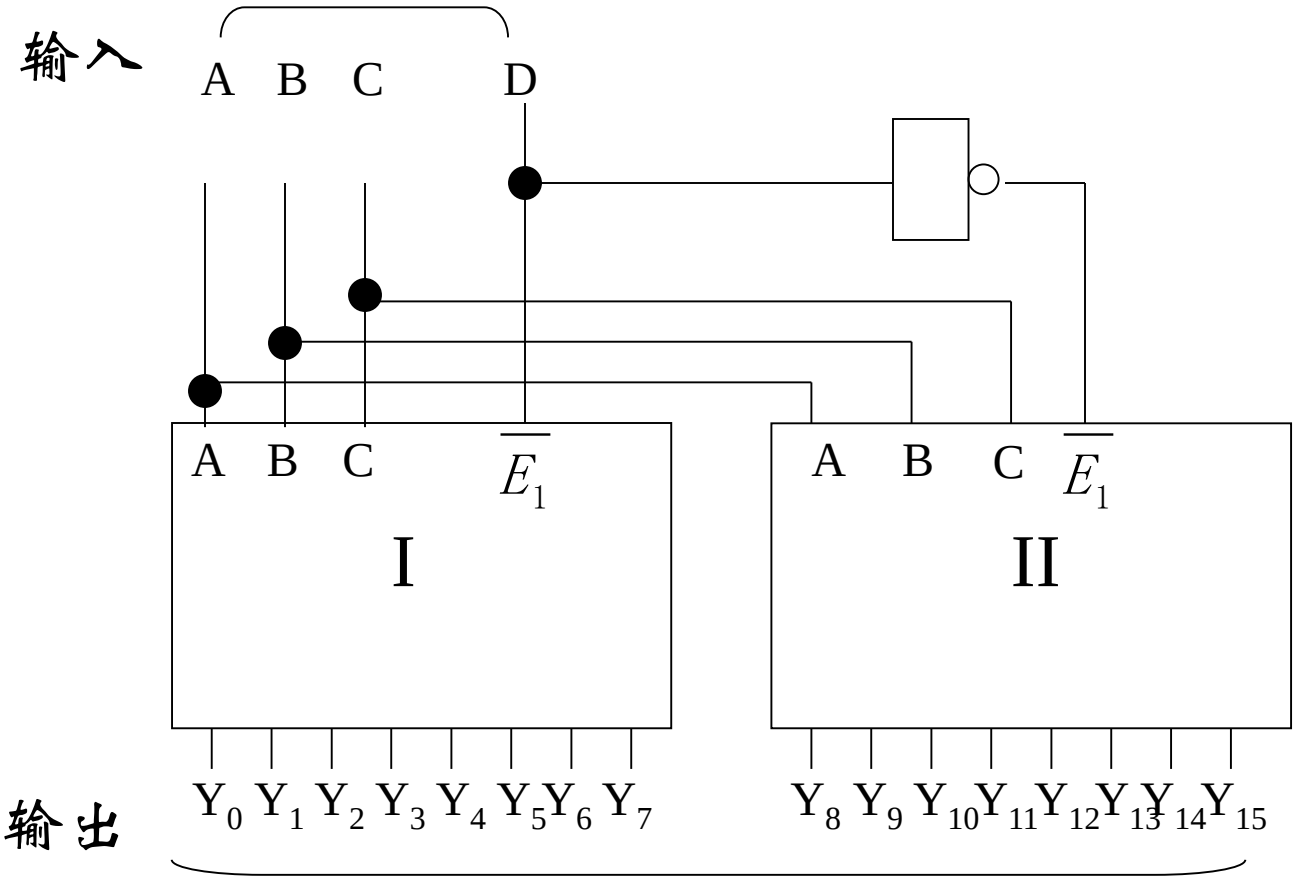
- 3-8译码器：按照输出表达式画出3－8译码器的逻辑图

$$
\begin{cases}
Y_0 = \overline{\overline{A}\,\overline{B}\,\overline{C}} \\
Y_1 = \overline{A\overline{B}\,\overline{C}} \\
Y_2 = \overline{\overline{A}B\overline{C}} \\
Y_3 = \overline{AB\overline{C}} \\
Y_4 = \overline{\overline{A}\,\overline{B}C} \\
Y_5 = \overline{A\overline{B}C} \\
Y_6 = \overline{\overline{A}BC} \\
Y_7 = \overline{ABC}
\end{cases}
$$



3.3.1 译码器（31）

- 用3-8译码器扩展成4-16译码器



3.3.1 译码器（32）

- 有多个使能端的译码器件——典型器件
  - 器件一： 74LS(HCT，HC）139

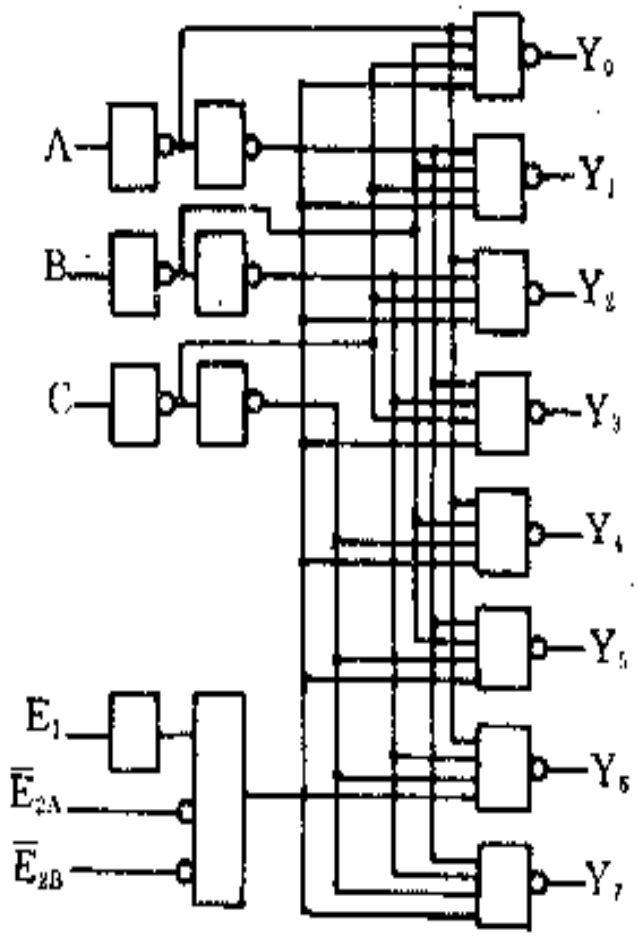
    功能：双2-4译码器
  - 器件二：74LS(HCT，HC）138

    功能：3-8 译码器（3个使能端）
  - 器件三：74 LS (HCT，HC）154

    功能：4-16 译码器（2个使能端）

功能表

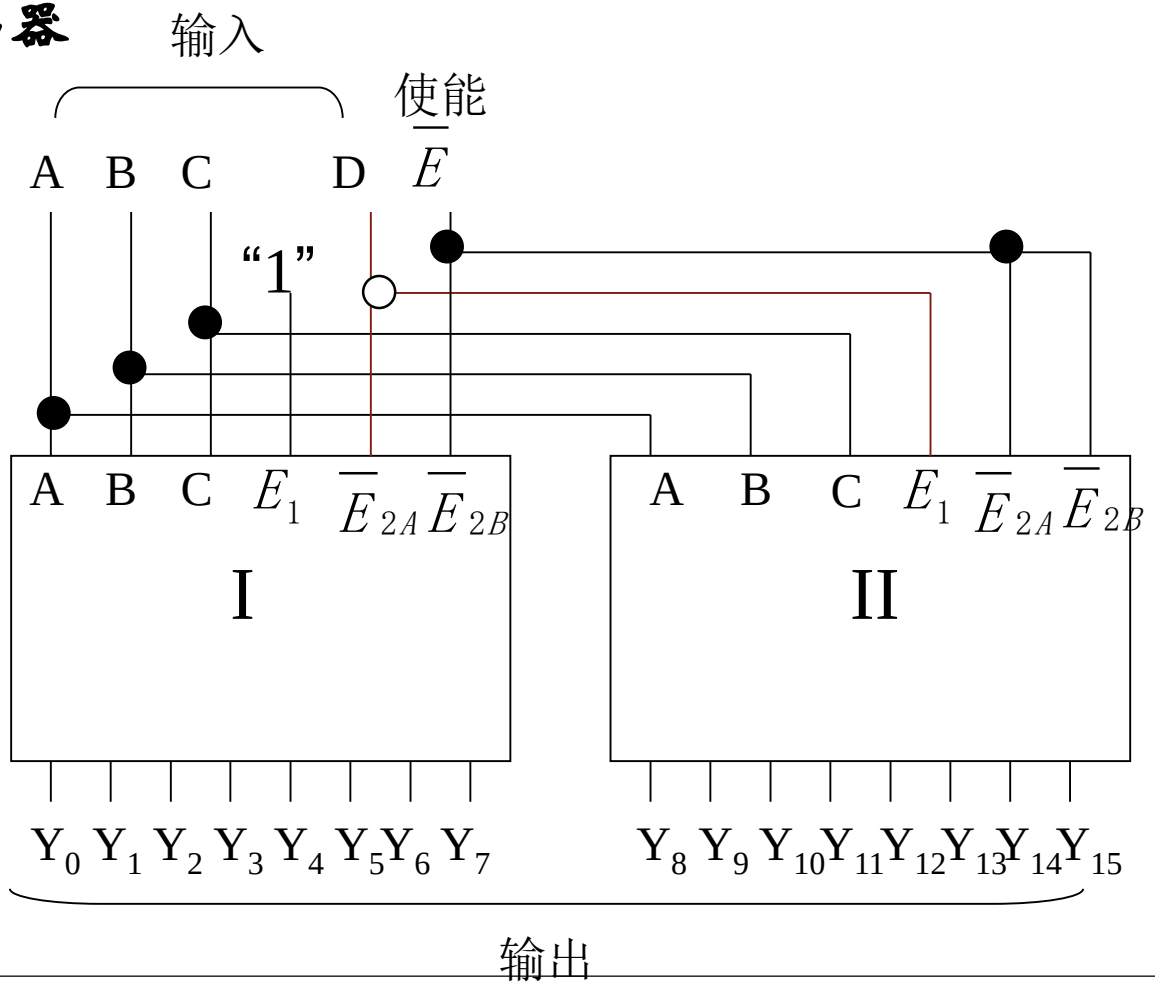
输入					输出							
E_1	$\overline{E}_{2A}+\overline{E}_{2B}$	A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
×	1	×	×	×	1	1	1	1	1	1	1	1
0	×	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	1	0	0	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	1	1	0	1	1	1	0	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0



4-2　三输入变量译码器的逻辑图

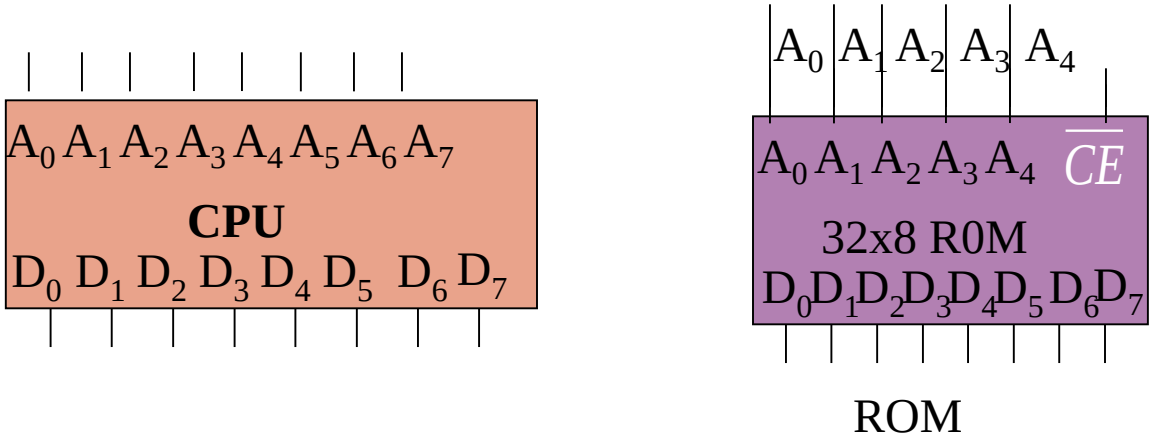
3.3.1 译码器（33）

- E用作扩展：具有多个使能端的3-8译码器扩展为4-16译码器

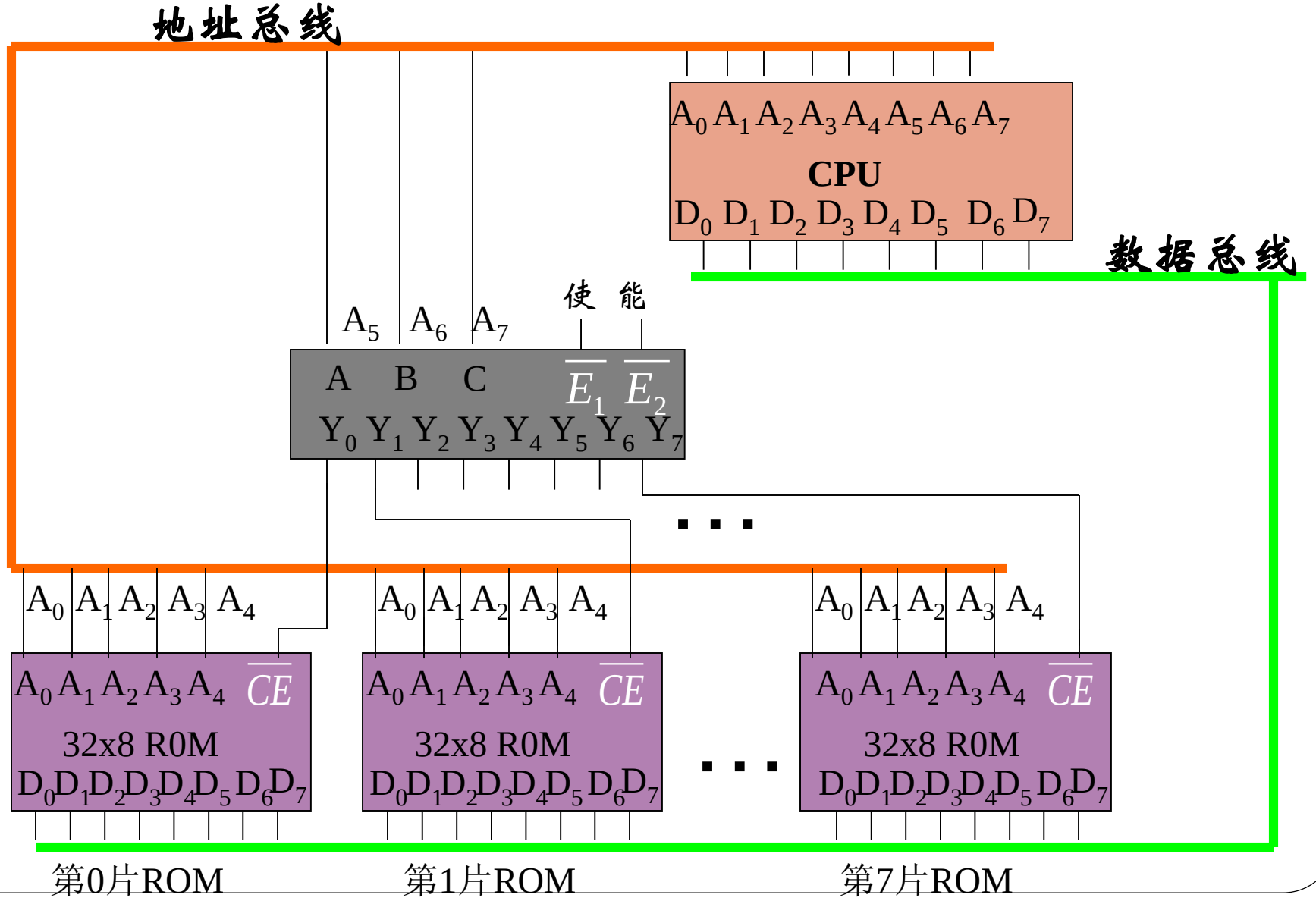


3.3.1 译码器（34）

- 用3－8译码器分配地址区
  - CPU的地址空间：A_7~A_0共有256个地址空间
  - 每个ROM有32个地址空间

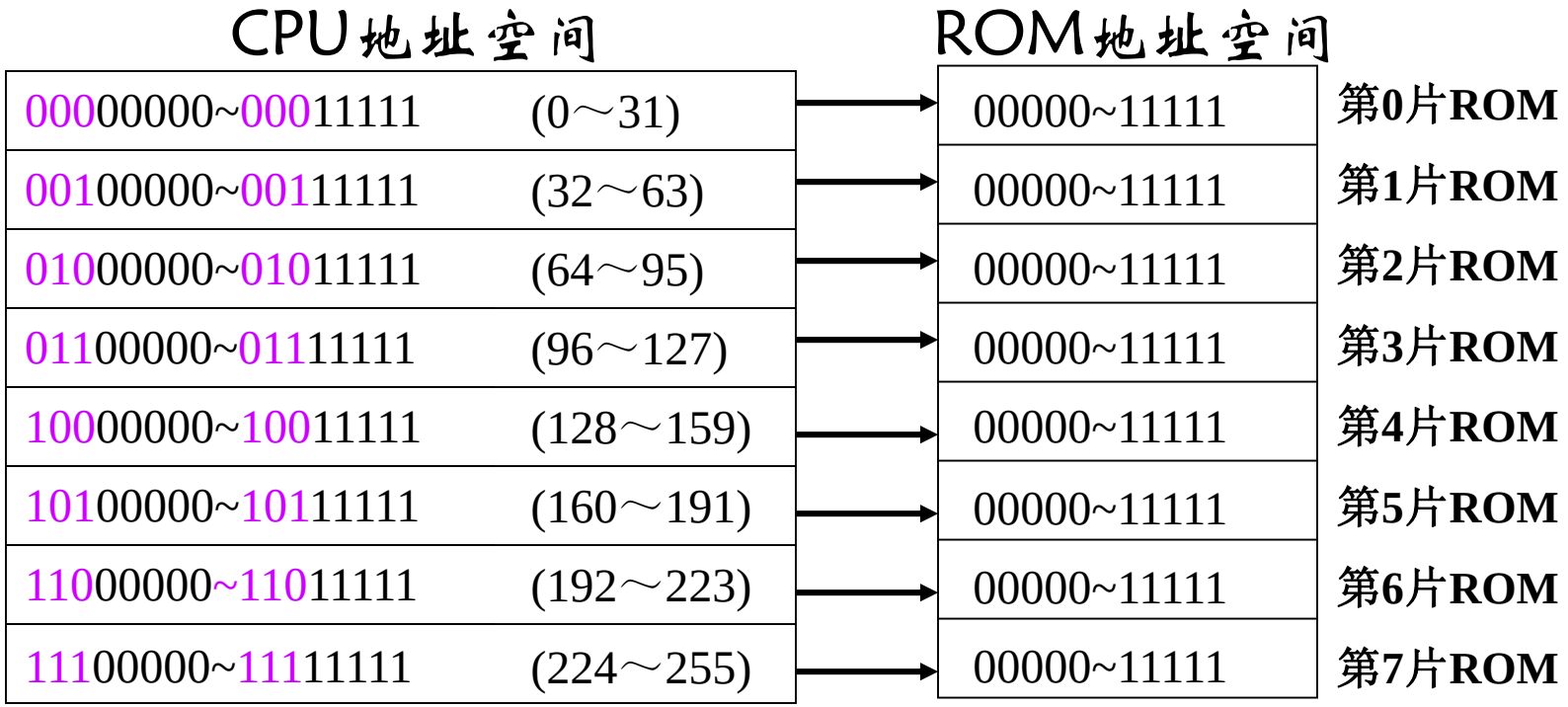


用3－8译码器分配地址区(1)



用3－8译码器分配地址区（2）

地址空间的对应关系如图：



3.3.1 译码器（35）

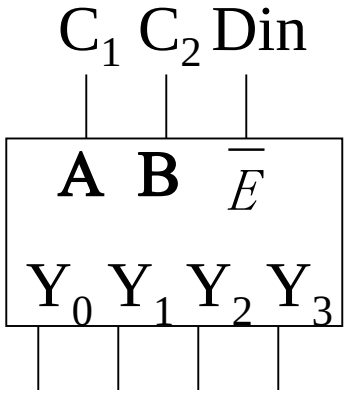
- 用译码器完成地址分配

地址线有10位，可以表示2^{10}=1K个地址空间；
地址线有20位，可以表示2^{20}=1M个地址空间；
地址线有30位，可以表示2^{30}=1G个地址空间；
32位地址可以表示4G地址；
16M存储器需要24位地址。

3.3.1 译码器（36）

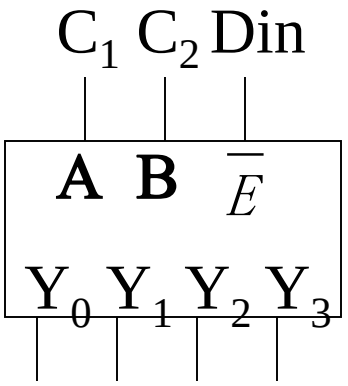
- 译码器用作数据分配器（Demultiplexer）
  - 数据分配：将输入数据在地址控制下连接到多个输出通道

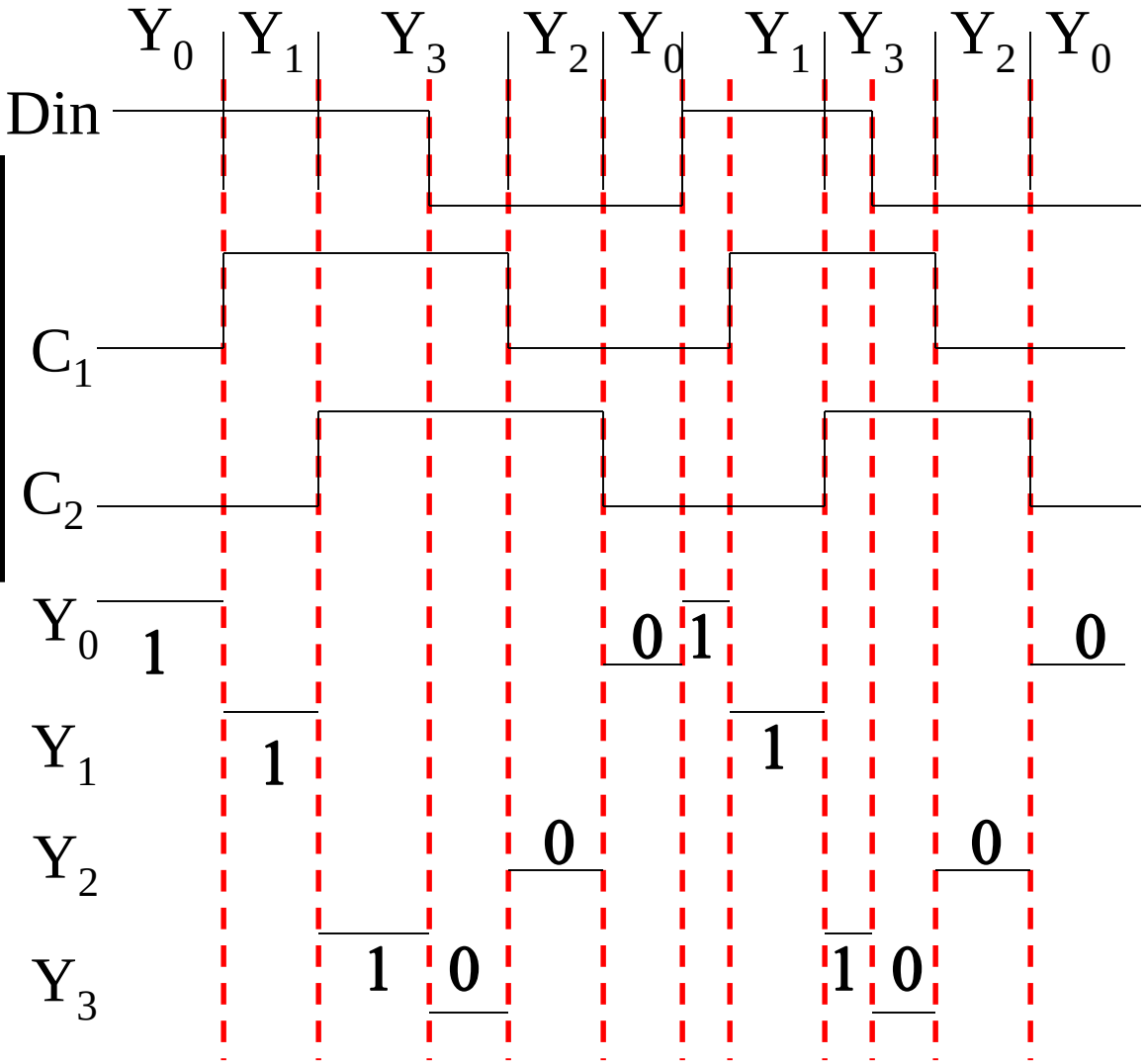
Din	C_1	C_2	Y_0	Y_1	Y_2	Y_3
0/1	0	0	0/1	1	1	1
0/1	1	0	1	0/1	1	1
0/1	0	1	1	1	0/1	1
0/1	1	1	1	1	1	0/1



3.3.1 译码器（37）

Din	C_1	C_2	Y_0	Y_1	Y_2	Y_3
0/1	0	0	0/1	1	1	1
0/1	1	0	1	0/1	1	1
0/1	0	1	1	1	0/1	1
0/1	1	1	1	1	1	0/1



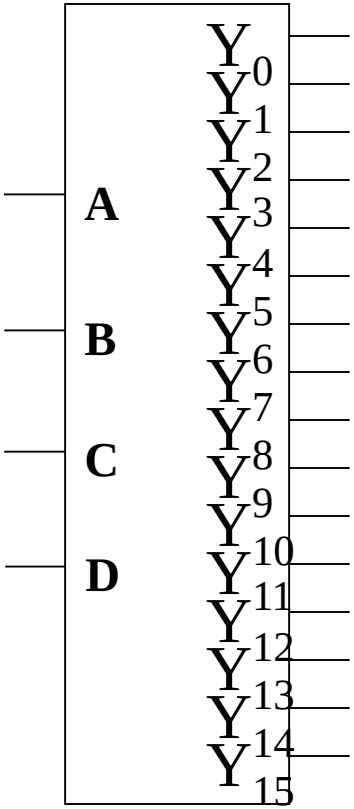


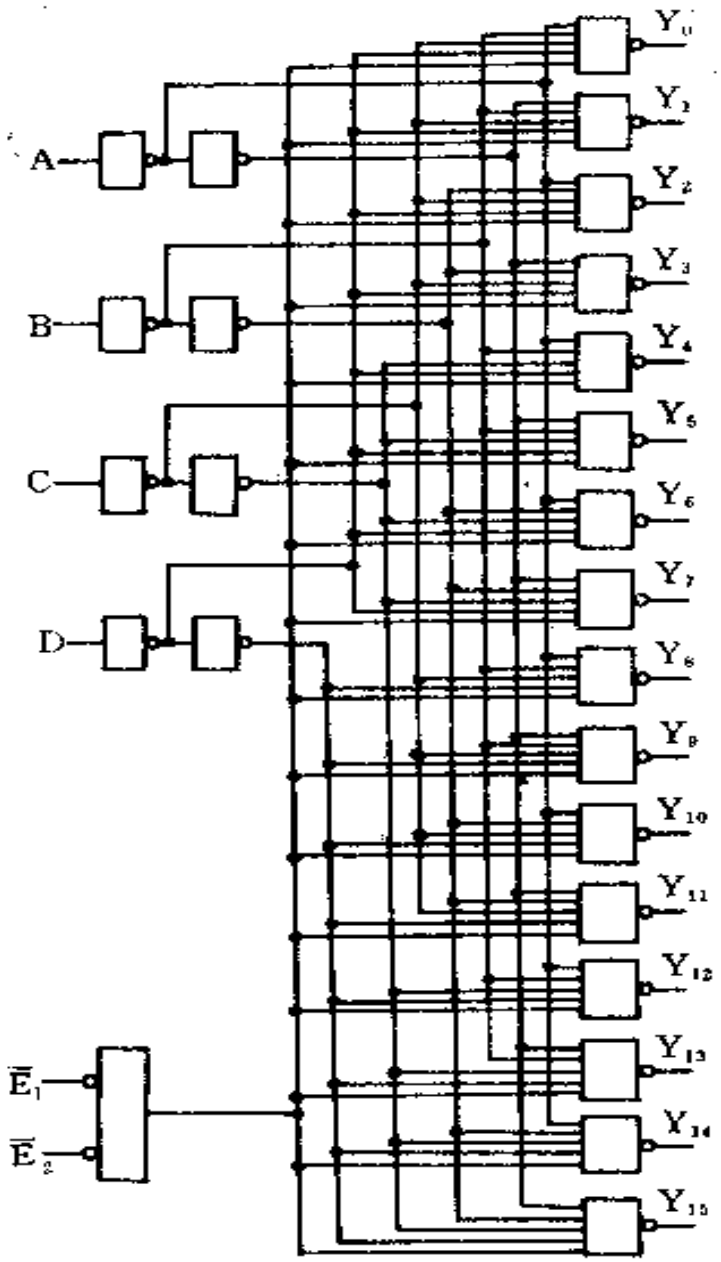
3.3.1 译码器（39）

- 4-16译码器
  - 定义：4－16译码器是指4输入－16输出的变量译码器。
  - 逻辑示意图

4－16译码器逻辑示意图

D为最高位，A为最低位





功　　能　　表

$\overline{E}_1$	$\overline{E}_2$	A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	×	×	×	×	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	×	×	×	×	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	×	×	×	×	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

图 4-3　四输入变量译码器逻辑图

3.3.1 译码器（40）

4-16译码器

存在的问题

- 缓冲门的负载较大：第一级缓冲门(反变量）负载9个负载，第二级缓冲门（原变量）8个负载
- 使能端与门的负载有16个，必须在制造芯片时增大驱动能力

3.3.1 译码器（41）

当输入变量数增大

- 当译码器的输入变量数N增大时，用单级译码器不能实现
  - ✓ 译码部分与非门的输入端数会增多：输入端数为N+1(使能端)个。
  - ✓ 二级Buffer的每个Buffer的输出负载加重
    - ❖ 负载:第一级为$2^{N-1}+1$，第二级为2^{N-1}，使能端为2^N
    - ❖ 例如，当N＝11时，每个译码门至少有12个输入，第一级缓冲门有1025个负载，第二级缓冲门有1024个负载，这是不可实现的。
  - ➢ 采用多级译码技术可以减少负载： 用在大容量存储器片内的译码结构。

3.3.1 译码器（41）

- 多级译码

考察4－16变量译码器

$$Y_0 = \overline{\overline{A}\overline{B}\overline{C}\overline{D}} = \overline{(\overline{A}\overline{B})(\overline{C}\overline{D})} \qquad Y_8 = \overline{A\overline{B}\overline{C}\overline{D}} = \overline{(A\overline{B})(\overline{C}\overline{D})}$$

$$Y_1 = \overline{\overline{A}\overline{B}\overline{C}D} = \overline{(\overline{A}\overline{B})(\overline{C}D)} \qquad Y_9 = \overline{A\overline{B}\overline{C}D} = \overline{(A\overline{B})(\overline{C}D)}$$

$$Y_2 = \overline{\overline{A}\overline{B}C\overline{D}} = \overline{(\overline{A}\overline{B})(C\overline{D})} \qquad Y_{10} = \overline{A\overline{B}C\overline{D}} = \overline{(A\overline{B})(C\overline{D})}$$

$$Y_3 = \overline{\overline{A}\overline{B}CD} = \overline{(\overline{A}\overline{B})(CD)} \qquad Y_{11} = \overline{A\overline{B}CD} = \overline{(A\overline{B})(CD)}$$

$$Y_4 = \overline{\overline{A}B\overline{C}\overline{D}} = \overline{(\overline{A}B)(\overline{C}\overline{D})} \qquad Y_{12} = \overline{AB\overline{C}D} = \overline{(AB)(\overline{C}D)}$$

$$Y_5 = \overline{\overline{A}B\overline{C}D} = \overline{(\overline{A}B)(\overline{C}D)} \qquad Y_{13} = \overline{AB\overline{C}D} = \overline{(AB)(\overline{C}D)}$$

$$Y_6 = \overline{\overline{A}BC\overline{D}} = \overline{(\overline{A}B)(C\overline{D})} \qquad Y_{14} = \overline{ABC\overline{D}} = \overline{(AB)(C\overline{D})}$$

$$Y_7 = \overline{\overline{A}BCD} = \overline{(\overline{A}B)(CD)} \qquad Y_{15} = \overline{ABCD} = \overline{(AB)(CD)}$$

3.3.1 译码器（41）

多级译码

令$E=\overline{\overline{A}\overline{B}}, F=\overline{A}\overline{B}, G=\overline{\overline{A}B}, H=AB$

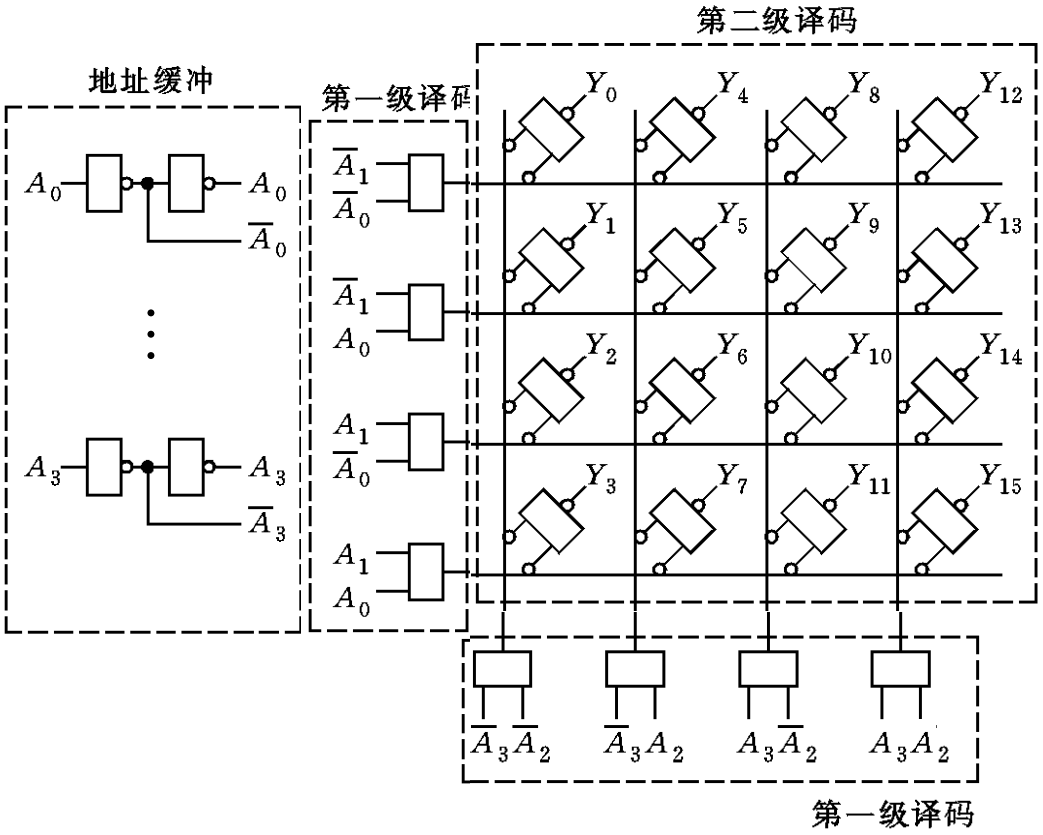
$W=\overline{\overline{C}\overline{D}}, X=\overline{C}\overline{D}, Y=\overline{\overline{C}D}, Z=CD$，则有：

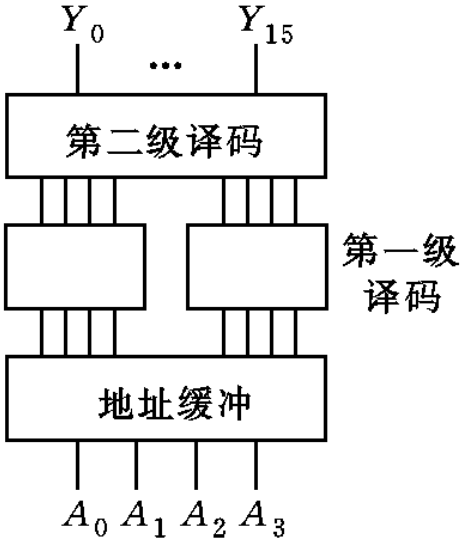
$Y_0=\overline{\overline{\overline{\overline{ABCD}}}}=\overline{EW}$	$Y_4=\overline{\overline{\overline{ABC}\overline{D}}}=\overline{EX}$	$Y_8=\overline{\overline{\overline{A}BCD}}=\overline{EY}$	$Y_{12}=\overline{\overline{AB}CD}=\overline{EZ}$
$Y_1=\overline{\overline{\overline{\overline{ABCD}}}}=\overline{FW}$	$Y_5=\overline{\overline{\overline{ABC}\overline{D}}}=\overline{FX}$	$Y_9=\overline{\overline{\overline{A}BCD}}=\overline{FY}$	$Y_{13}=\overline{A\overline{B}CD}=\overline{FZ}$
$Y_2=\overline{\overline{A}\overline{\overline{BCD}}}=\overline{GW}$	$Y_6=\overline{\overline{A}\overline{\overline{BC}\overline{D}}}=\overline{GX}$	$Y_{10}=\overline{\overline{A}\overline{\overline{B}CD}}=\overline{GY}$	$Y_{14}=\overline{A\overline{BCD}}=\overline{GZ}$
$Y_3=\overline{\overline{A}\overline{\overline{BCD}}}=\overline{HW}$	$Y_7=\overline{\overline{A}\overline{\overline{BC}\overline{D}}}=\overline{HX}$	$Y_{11}=\overline{\overline{A}\overline{\overline{B}CD}}=\overline{HY}$	$Y_{15}=\overline{ABCD}=\overline{HZ}$

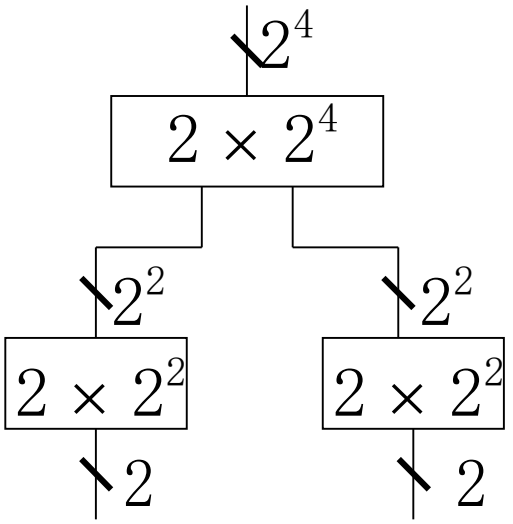
- ✓A,B,C,D和它们的反变量，都出现8次，说明它们共有8个负载。
- ✓考察E,F,G,H,W,X,Y,Z，每个出现4次，意味着它们共有4个负载。
- ✓说明经过变换后，负载数降低了一半。

二级译码

用两级译码电路实现4-16译码器



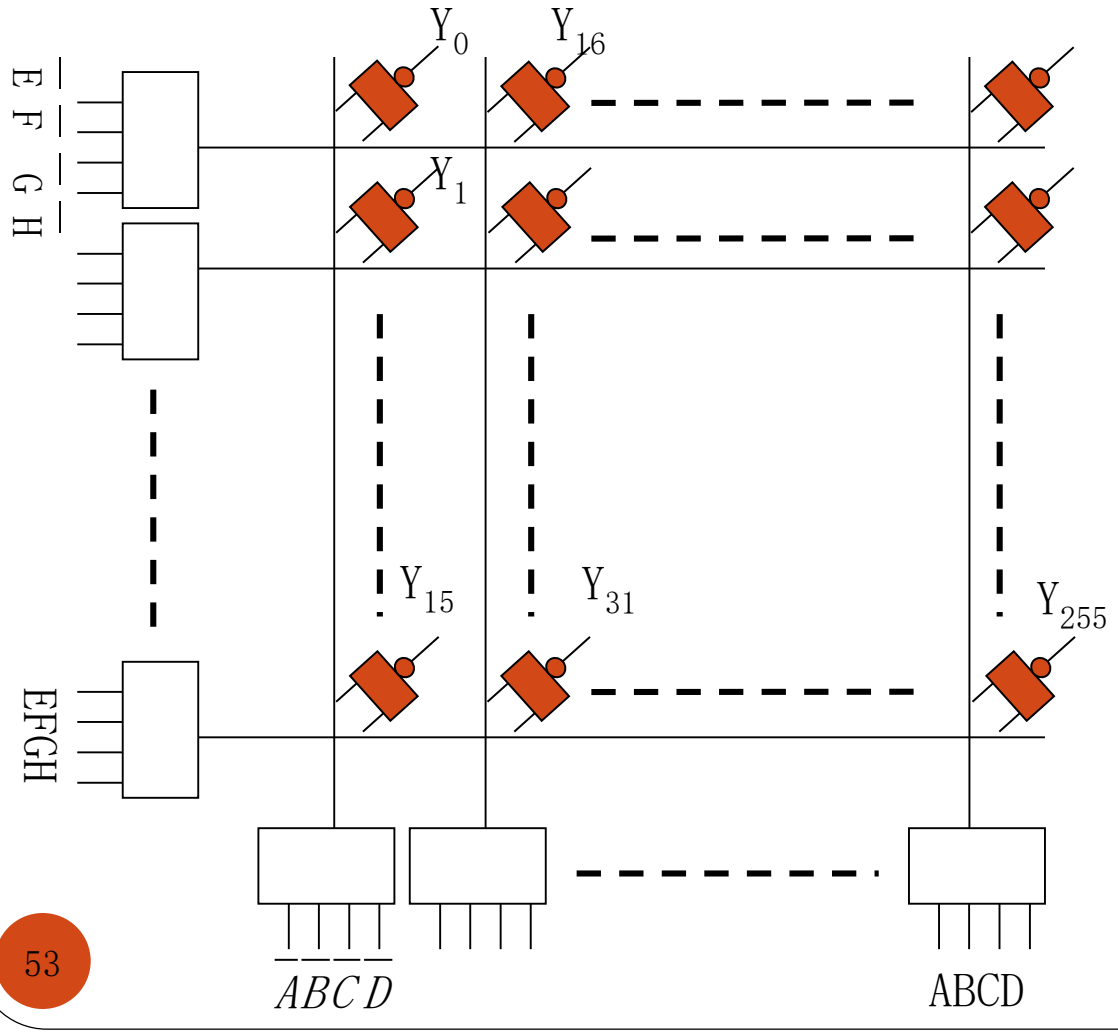


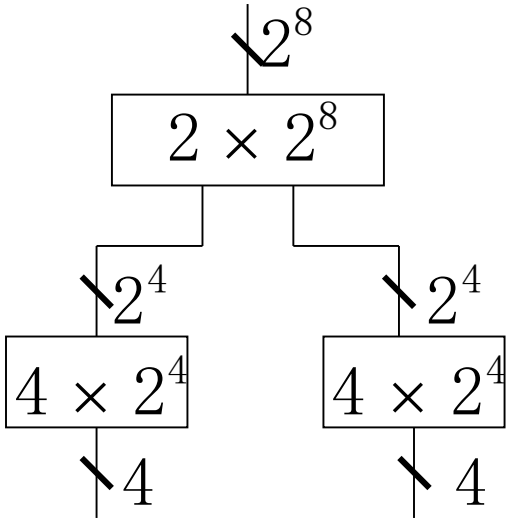


(2X2^2表示2输入与门4个，2X2^4表示2输入与非门16个)

二级译码

用两级译码实现8-256译码器

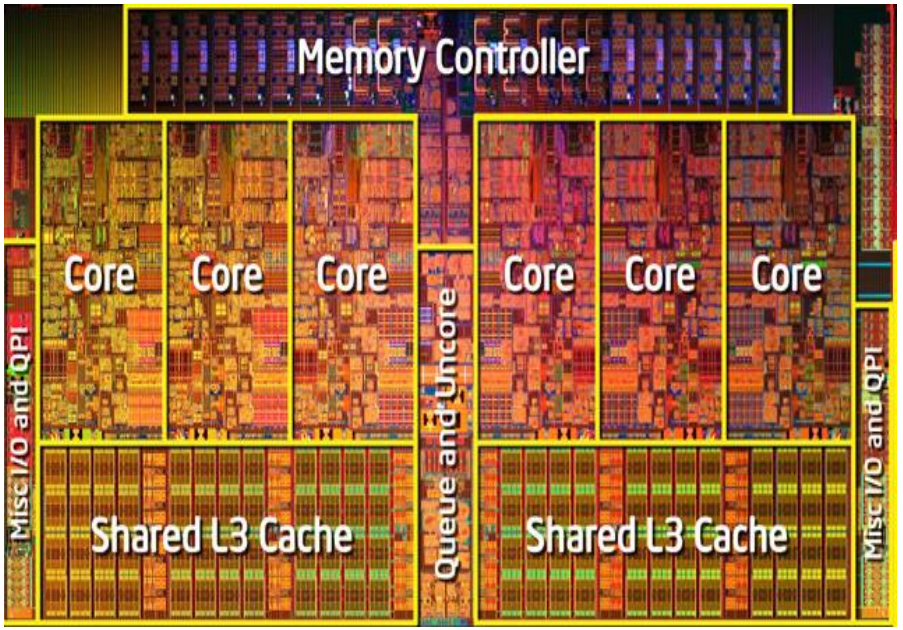




- 负载：每个反变量为8+1，原变量为8。每个与门16个负载。
- 负载：如果用一级译码，则每个原变量负载为128，每个反变量负载为129。

大容量存储器的地址译码

地址线有10位，可以表示210＝1K个地址空间；
地址线有20位，可以表示2^{20}＝1M个地址空间；
地址线有30位，可以表示2^{30}＝1G个地址空间；
32位地址可以表示4G地址；
16M存储器需要24位地址。



3.3.1 译码器（42）
——变量译码器小结

- 译码器的功能分类：
  - 变量译码器：用来表示输入变量状态的全部组合,N位输入，2^N输出。
    - 2-4译码器：设计，存在的问题：竞争与冒险
      使能端，作用：扩展、消除竞争与冒险
    - 3-8译码器：应用：地址分配，数据选择
    - 多级译码器：二级译码：4-16译码器和8-256译码器

3.3.1 译码器（43）

- 译码器的功能分类：
  - 变量译码器
  - 码制译码器：如8421码变换为循环码等
  - 显示译码器：控制数码管显示

3.3.1 译码器（44）

- 码制译码器：将一种编码变换为另外一种编码的逻辑电路。

  二—十进制译码器

3.3.1 译码器（45）

- 码制译码器：将一种编码变换为另外一种编码的逻辑电路。

  二—十进制译码器：

  十进制的二进制编码（二进制编码的十进制数，也叫BCD编码：Binary-Coded to Decimal, BCD）

  - 1、不完全译码的BCD译码器
  - 2、完全译码的BCD译码器

3.3.1 译码器（46）

- 8-4-2-1 码表示十进制数

D C \ B A	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	X	X	X	X
10	8	9	X	X

十进制数	8421码
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

3.3.1 译码器（47）

- 不完全译码的BCD 译码器的功能表

	A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	0	1	1	1	1	1	1	1	1
2	0	1	0	0	1	1	0	1	1	1	1	1	1	1
3	1	1	0	0	1	1	1	0	1	1	1	1	1	1
4	0	0	1	0	1	1	1	1	0	1	1	1	1	1
5	1	0	1	0	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	1	1	1	0	1	1	1	1	1	1	1	0	1	1
8	0	0	0	1	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

（a）功能表

3.3.1 译码器（48）

- 不完全译码的BCD译码器逻辑化简

当ABCD＝0101~1111时，$Y_{0\sim9}$均为任意值，$Y_{0\sim9}$表达式为

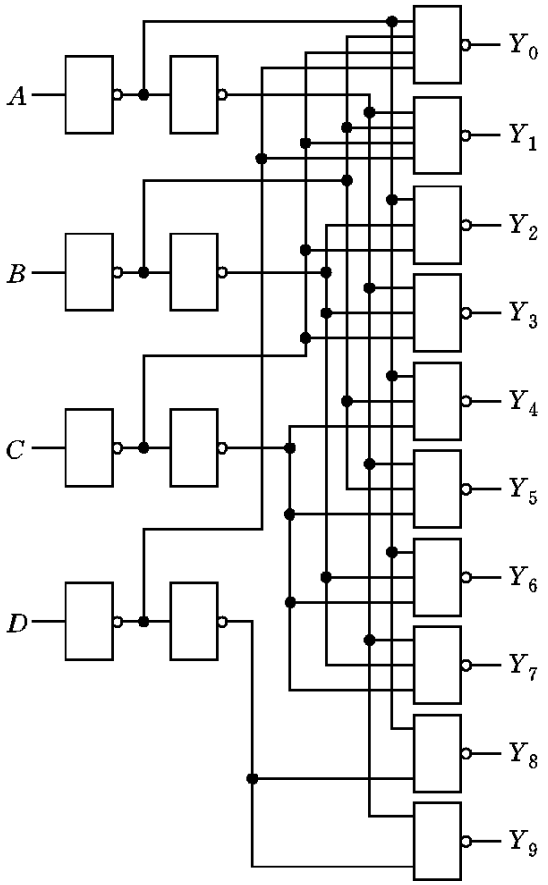
D C \ B A	00	01	11	10
00	Y_0=0	Y_1=0	Y_3=0	Y_2=0
01	Y_4=0	Y_5=0	Y_7=0	Y_6=0
11	X	X	X	X
10	Y_8=0	Y_9=0	X	X

$$\begin{cases} Y_0 = \overline{\overline{A}\overline{B}\overline{C}\overline{D}} \\ Y_1 = \overline{\overline{A}\overline{B}\overline{C}D} \\ Y_2 = \overline{\overline{A}\overline{B}C} \\ \\ Y_8 = \overline{\overline{A}D} \\ Y_9 = \overline{AD} \end{cases}$$

3.3.1 译码器（49）

- 不完全译码的BCD译码器逻辑图

$$
\begin{cases}
Y_0 = \overline{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}} \\
Y_1 = \overline{A\overline{B}\,\overline{C}\,\overline{D}} \\
Y_2 = \overline{\overline{A}\,\overline{\overline{B}\,\overline{C}}} \\
\\
Y_8 = \overline{\overline{A}D} \\
Y_9 = \overline{A\overline{D}}
\end{cases}
$$



(b) 逻辑图

3.3.1 译码器（50）

- **完全译码的BCD译码器**

  当输入ABCD出现0101~1111时，译码器输出$Y_{0\sim9}$均为“1”，$Y_{0\sim9}$表达式为

	A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	0	1	1	1	1	1	1	1	1
2	0	1	0	0	1	1	0	1	1	1	1	1	1	1
3	1	1	0	0	1	1	1	0	1	1	1	1	1	1
4	0	0	1	0	1	1	1	1	0	1	1	1	1	1
5	1	0	1	0	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	1	1	1	0	1	1	1	1	1	1	1	0	1	1
8	0	0	0	1	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
不用	0	1	0	1	1	1	1	1	1	1	1	1	1	1
	1	1	0	1	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	1	1	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	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1

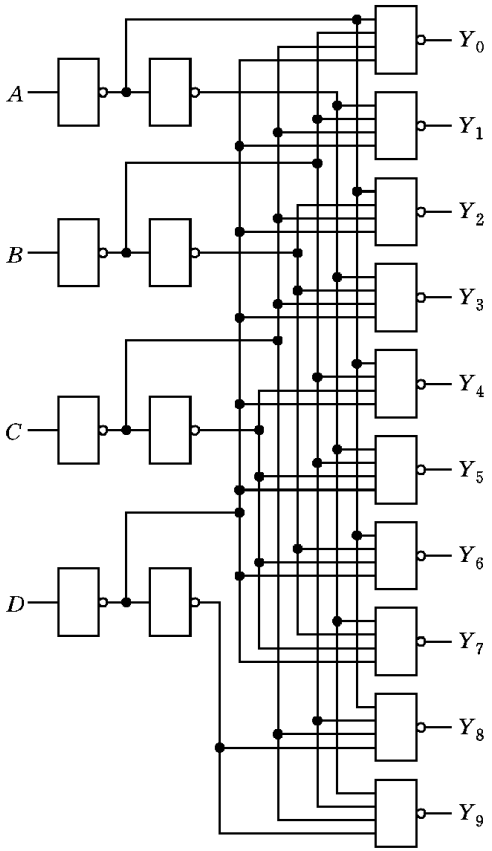
(a) 功能表

$$\begin{cases} Y_0 = \overline{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}} \\ Y_1 = \overline{\overline{A}\,\overline{B}\,\overline{C}\,D} \\ \\ Y_9 = \overline{\overline{A}\,B\,\overline{C}\,D} \end{cases}$$

3.3.1 译码器（51）

- 完全译码的BCD译码器逻辑图

$$\begin{cases} Y_0 = \overline{\overline{A}\overline{B}\overline{C}\overline{D}} \\ Y_1 = \overline{\overline{A}\overline{B}\overline{C}D} \\ \\ Y_9 = \overline{\overline{A}\overline{B}\overline{C}D} \end{cases}$$



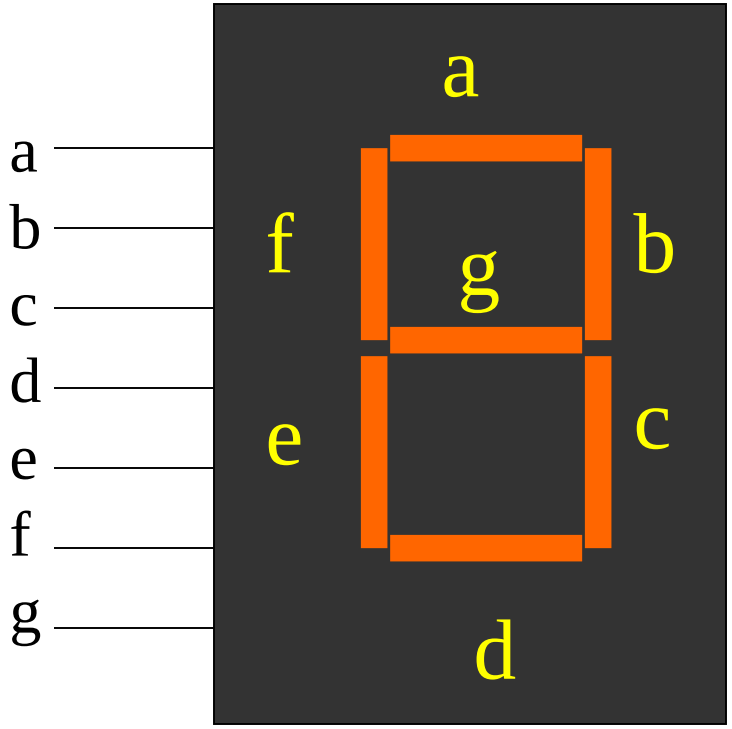
（b）逻辑图

3.3.1 译码器（52）

- 译码器的功能分类：
  - 变量译码器
  - 码制译码器
  - 显示译码器：控制数码管显示

3.3.1 译码器（53）

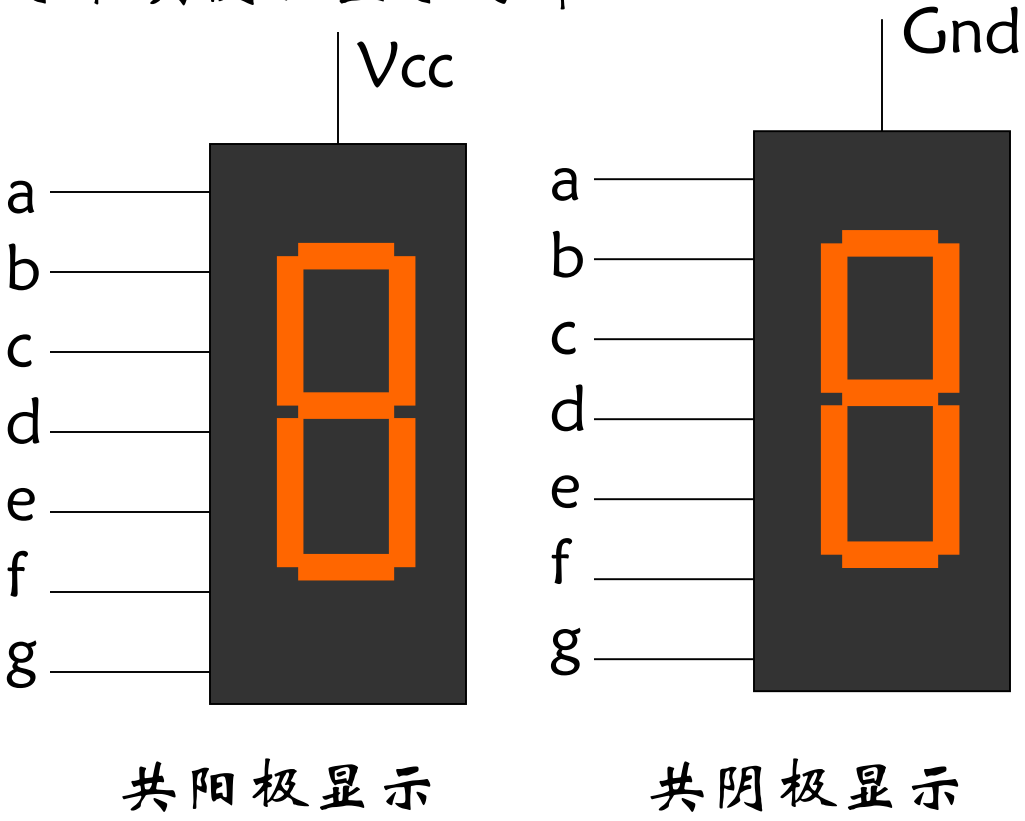
- 显示译码器
  - 一个七段数码管有7个控制端输入a～g，分别对应与数码管的7段。有些数码管是8个输入，在右下方有一个小数点。



3.3.1 译码器（54）

显示译码器

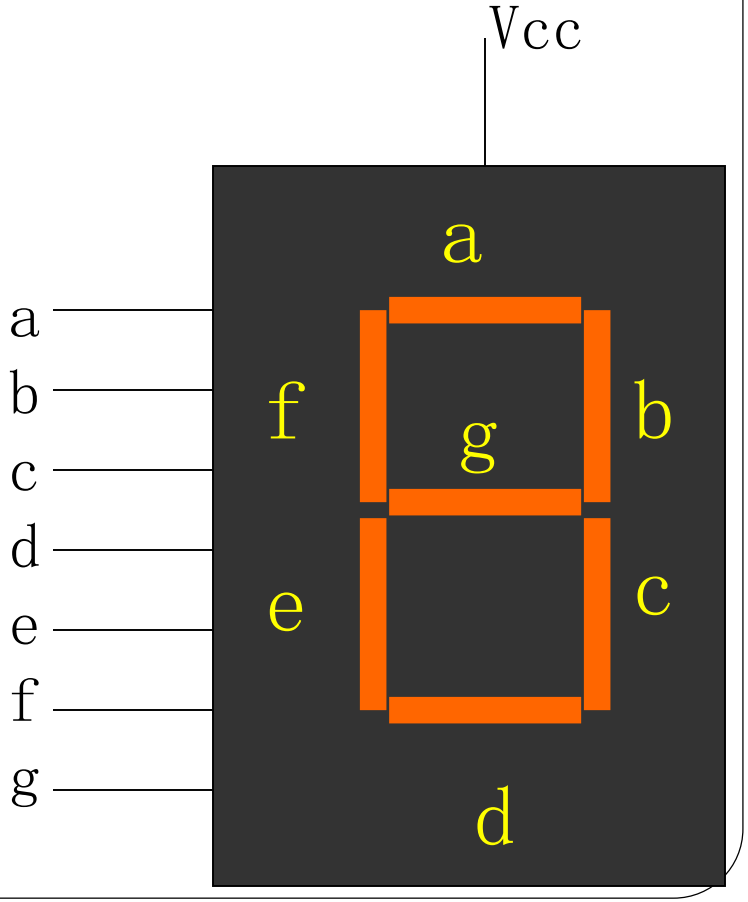
- 七段数码管分为共阳极显示和共阴极显示两种
- 一般来说，共阳极显示的数码管有一个管脚为“低”，则对应的段点亮，为“高”则灭。
- 共阴极显示的数码管有一个管脚为“高”，则对应的段点亮，为“低”则灭。
- 我们教材用共阳极显示数码管讲解



3.3.1 译码器（55）

显示译码器

- 当a=0时，a段亮，a=1时，a段灭。
- 其它变量相同。
- 7段数码管可以显示从0～9的数字。要显示0时，g=1,其它变量=0；显示2时，a,b,g,e,d=0,其它变量=1。
- 显示0～9中的任何一个分别对应于a～g的一组编码



3.3.1 译码器（56）

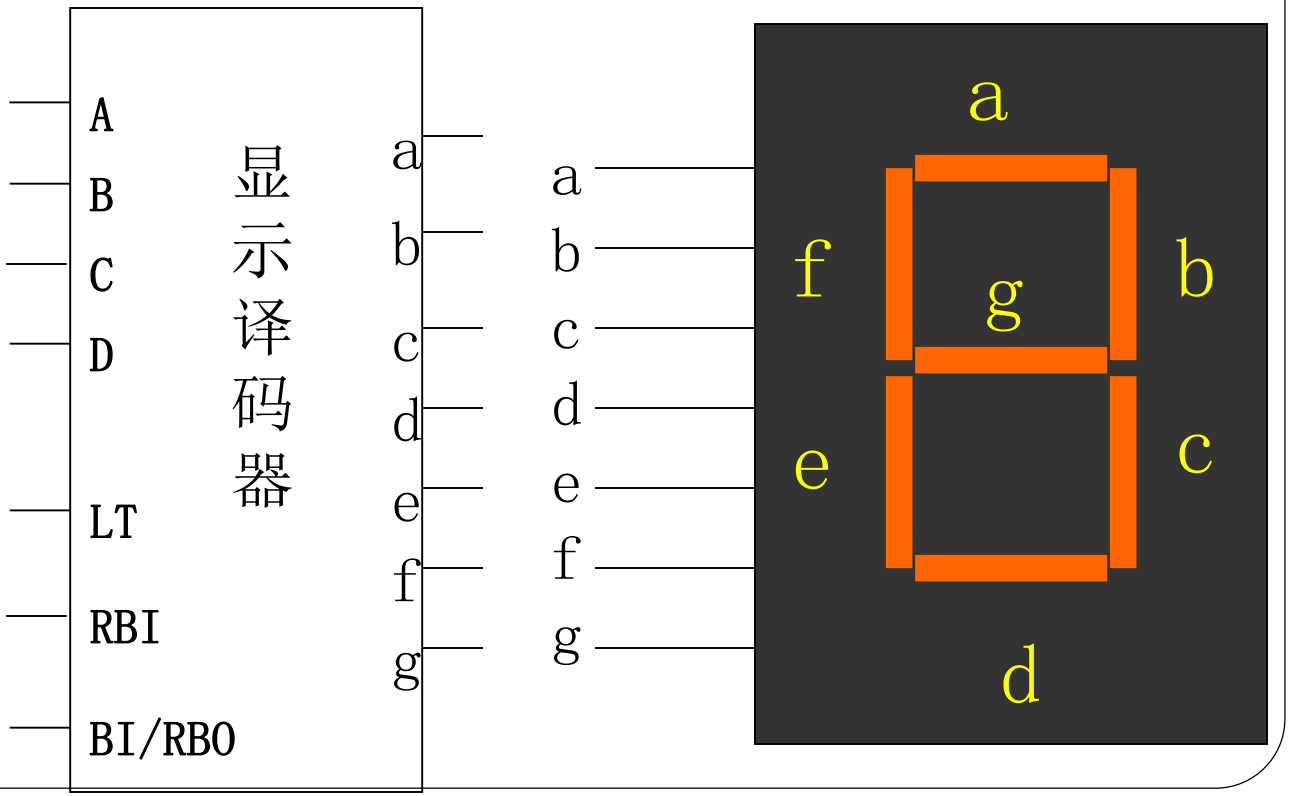
■ 显示译码器

□ 数字逻辑电路中用BCD码表示十进制数

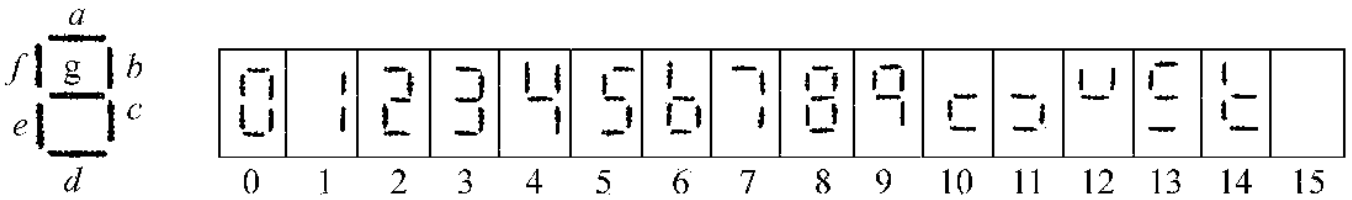
- 在BCD码和7段数码管编码之间需要一个显示译码器。

- 将BCD码转换为数码管对应的编码

- 显示译码器的实质是码制译码器



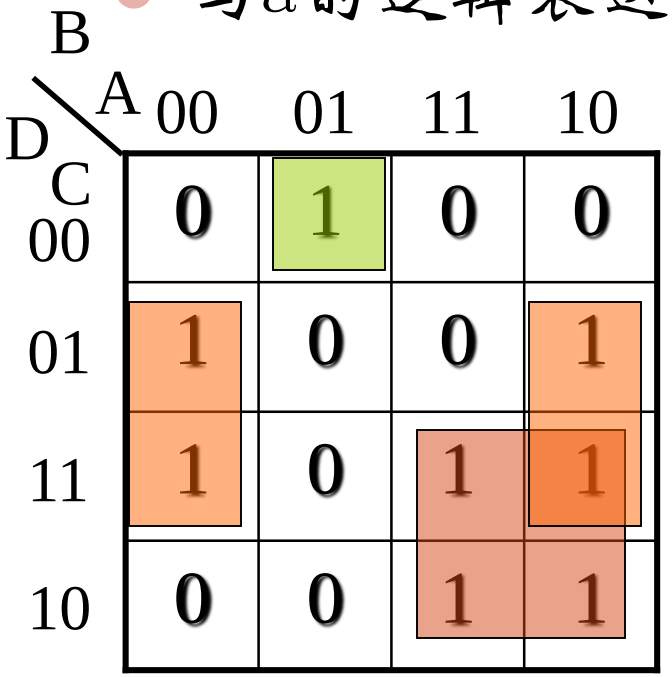
3.3.1 译码器（57）



显示数字	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	1	0	0	1	1	1	1
2	0	1	0	0	0	0	1	0	0	1	0
3	1	1	0	0	0	0	0	0	1	1	0
4	0	0	1	0	1	0	0	1	1	0	0
5	1	0	1	0	0	1	0	0	1	0	0
6	0	1	1	0	1	1	0	0	0	0	0
7	1	1	1	0	0	0	0	1	1	1	1
8	0	0	0	1	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	1	1	0	0
A	0	1	0	1	1	1	1	0	0	1	0
B	1	1	0	1	1	1	0	0	1	1	0
C	0	0	1	1	1	0	1	1	1	0	0
D	1	0	1	1	0	1	1	0	1	0	0
E	0	1	1	1	1	1	1	0	0	0	0
F	1	1	1	1	1	1	1	1	1	1	1

3.3.1 译码器（58）

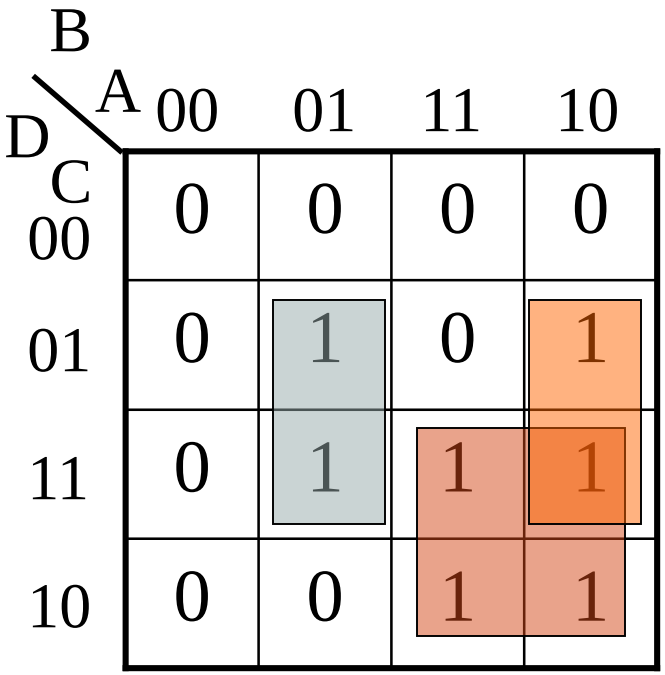
- 显示译码器
  - 写出各个变量的逻辑表达式
    - 写a的逻辑表达式



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

3.3.1 译码器（59）

- 显示译码器
  - 写出各个变量的逻辑表达式



$$b = BD + A\overline{B}C + \overline{A}BC$$

3.3.1 译码器（60）

- **显示译码器**

用同样的方法写出各个变量的逻辑表达式

$$a = BD + \overline{AC} + A\overline{BCD}$$

$$b = \overline{BD} + A\overline{BC} + \overline{A}BC$$

$$c = \overline{AB}\,\overline{C} + \overline{CD}$$

$$d = A\overline{BC} + \overline{A}BC + ABC$$

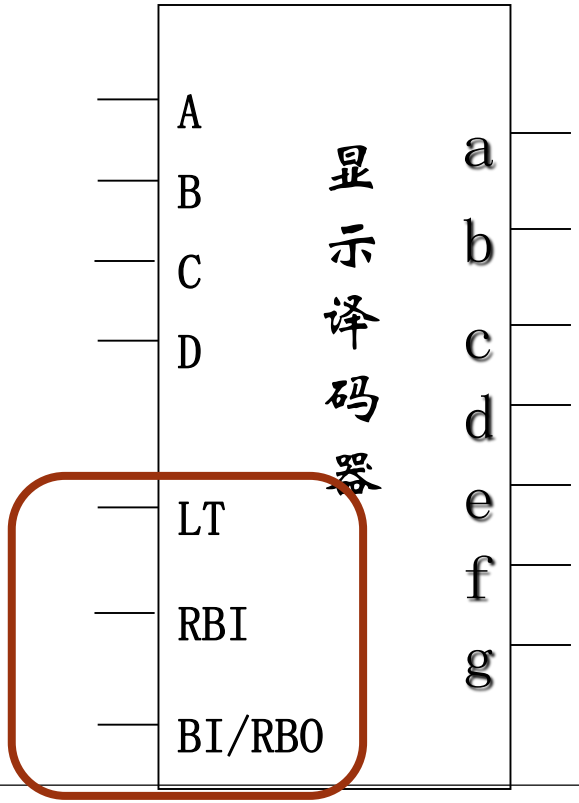
$$e = A + \overline{BC}$$

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

$$g = \overline{BCD} + ABC$$

3.3.1 译码器（61）

- 显示译码器
  - 显示译码器控制功能输入：LT, RBI, BI/RBO
  - 主要用于对数码管的测试和其他一些应用。



3.3.1 译码器（63）——小结

- 译码器的功能分类：
  - 变量译码器
  - 码制译码器
  - 显示译码器：控制数码管显示

3.3 常用的中规模组合逻辑电路

3.3.1 译码器

⇨ 3.3.2 数据选择器

3.3.3 编码器

3.3.4 数据比较器

3.3.5 奇偶校验器

3.3.6 运算器（算数逻辑单元 ALU）

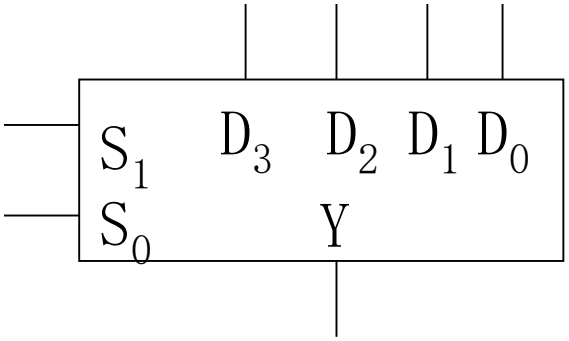
3.3.2 数据选择器(1)

- 数据选择器
  - 在选择控制的信号作用下，能从多个输入数据中选择一个或多个作为输出。
  - 多输入单输出数据选择器
  - 多输入多输出数据选择器

3.3.2 数据选择器(2)

- 4选1数据选择器
  - 4输入，1输出，2个选择控制

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

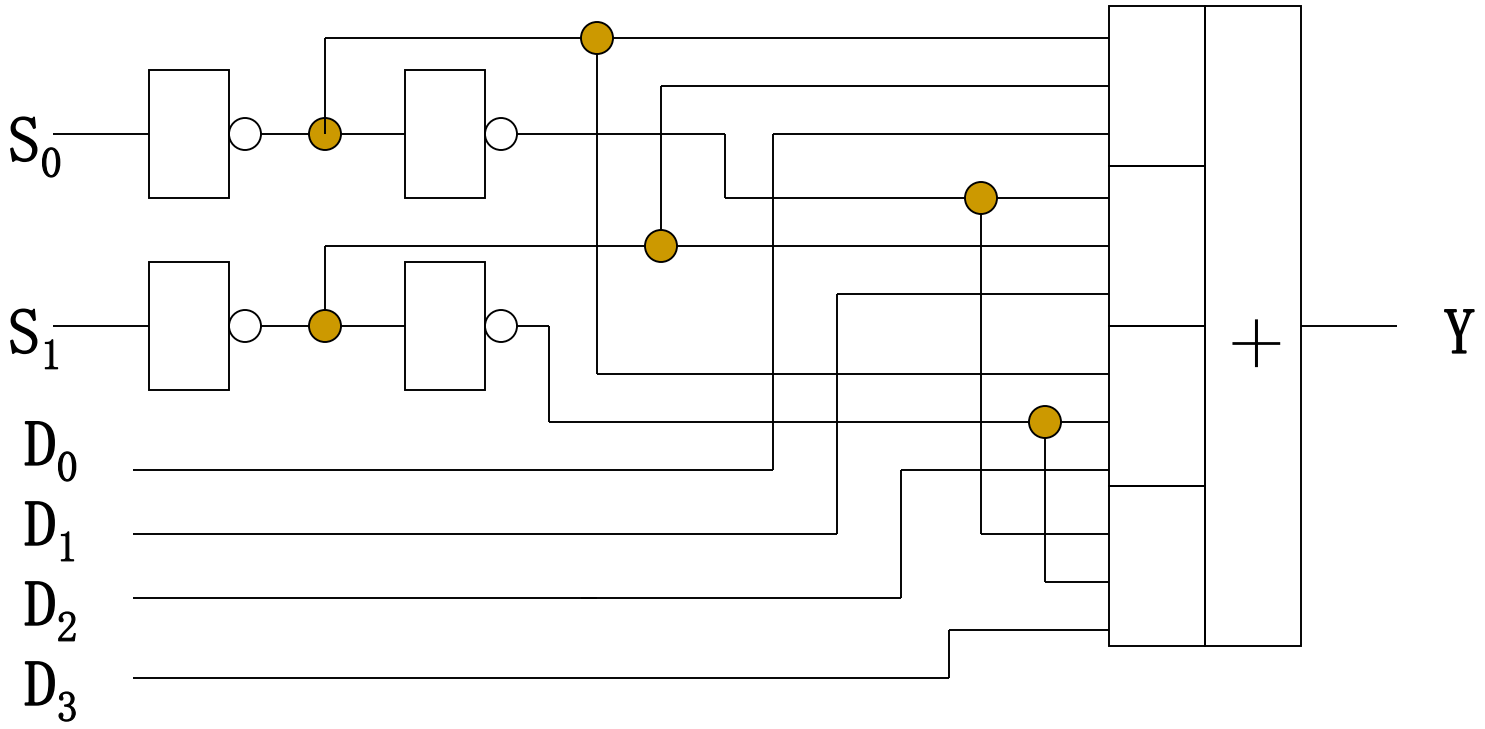


$$Y = \overline{S_0}\,\overline{S_1}D_0 + S_0\overline{S_1}D_1 + \overline{S_0}S_1D_2 + S_0S_1D_3$$

3.3.2 数据选择器(3)

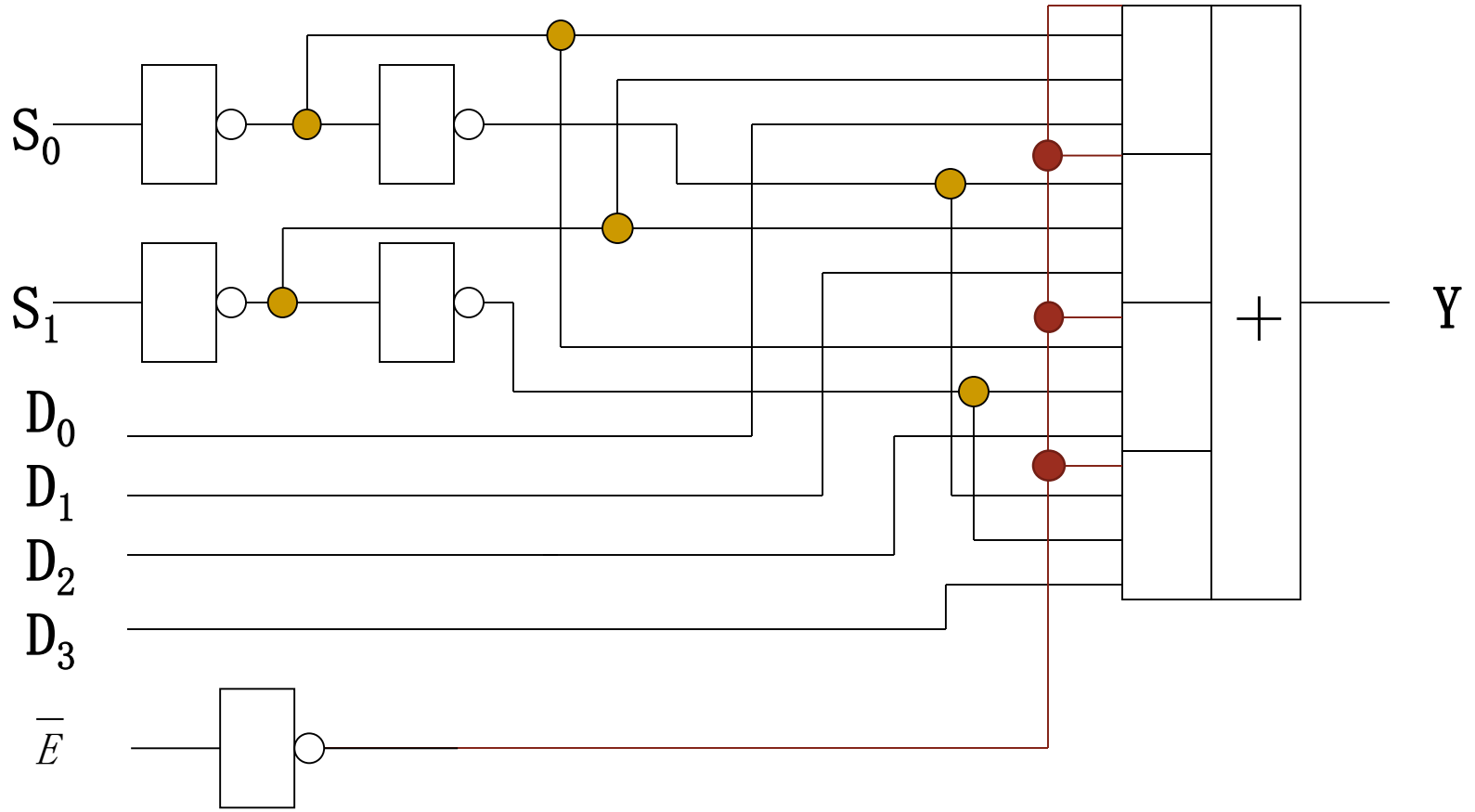
- 4选1数据选择器逻辑图

$$Y = \overline{S_0}\,\overline{S_1}D_0 + S_0\overline{S_1}D_1 + \overline{S_0}S_1D_2 + S_0S_1D_3$$



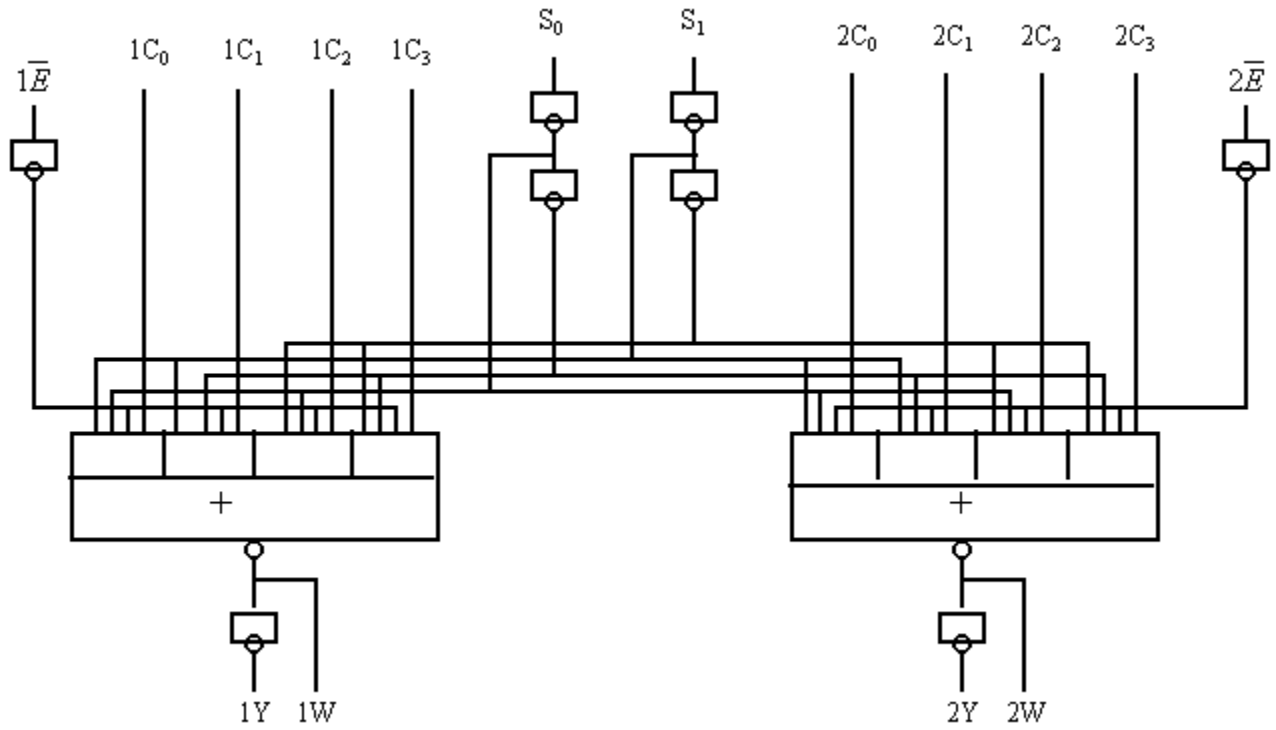
3.3.2 数据选择器(4)

- 4选1数据选择器逻辑图
  - 如果设计使能端，需要加在什么地方？



3.3.2 数据选择器(5)

- 有使能端的双4选1数据选择器
  - 注意输出结构，提供1正1反两个输出



选择器扩展:用双4选1选择器扩展成16选1选择器

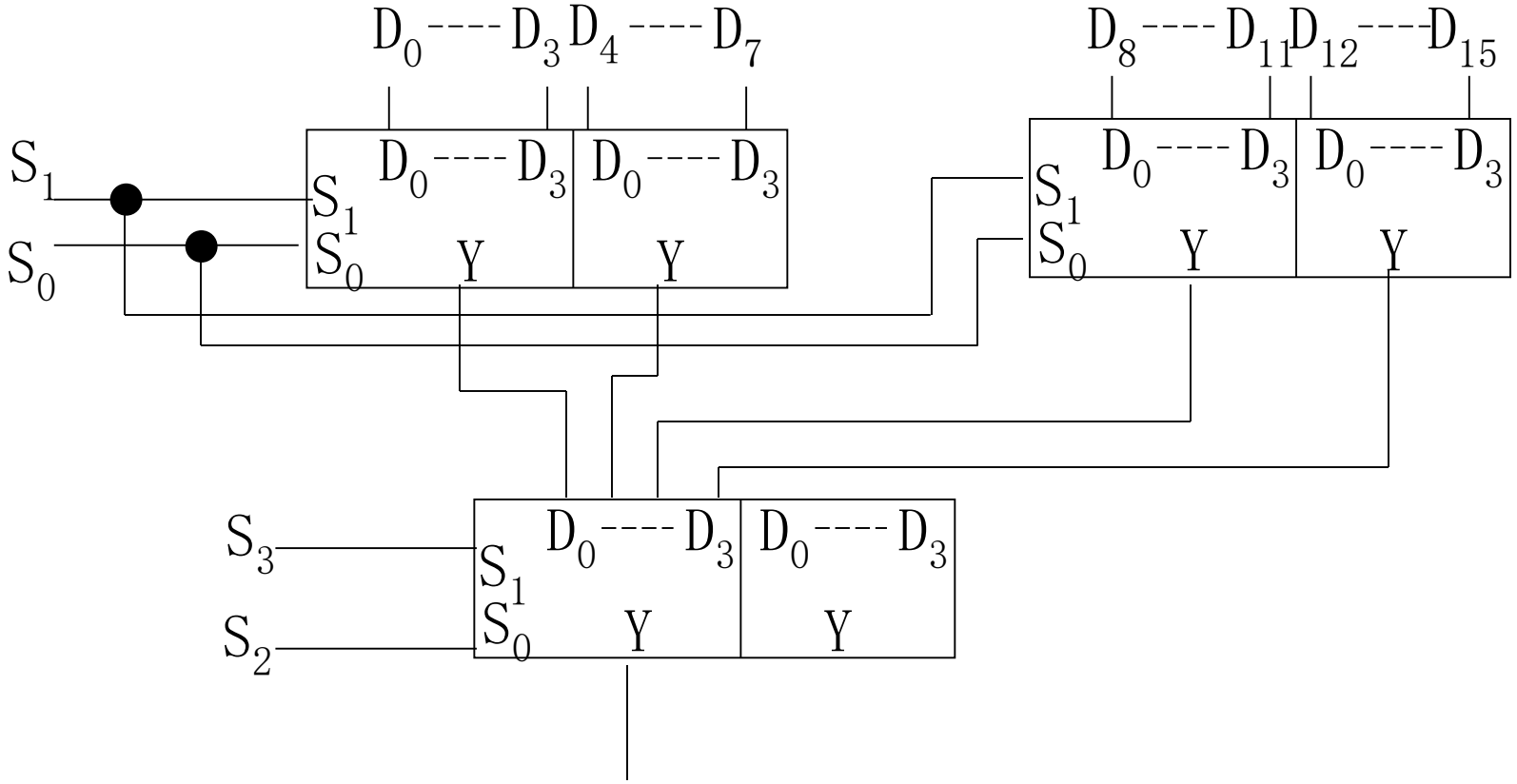
16选1功能表

两种不同的扩展方案，从功能表上分析，可以先选低两位，也可以先选高两位。

S_3	S_2	S_1	S_0	Y
0	0	0	0	Y_0
		0	1	Y_1
		1	0	Y_2
		1	1	Y_3
0	1	0	0	Y_4
		0	1	Y_5
		1	0	Y_6
		1	1	Y_7
1	0	0	0	Y_8
		0	1	Y_9
		1	0	Y_{10}
		1	1	Y_{11}
1	1	0	0	Y_{12}
		0	1	Y_{13}
		1	0	Y_{14}
		1	1	Y_{15}

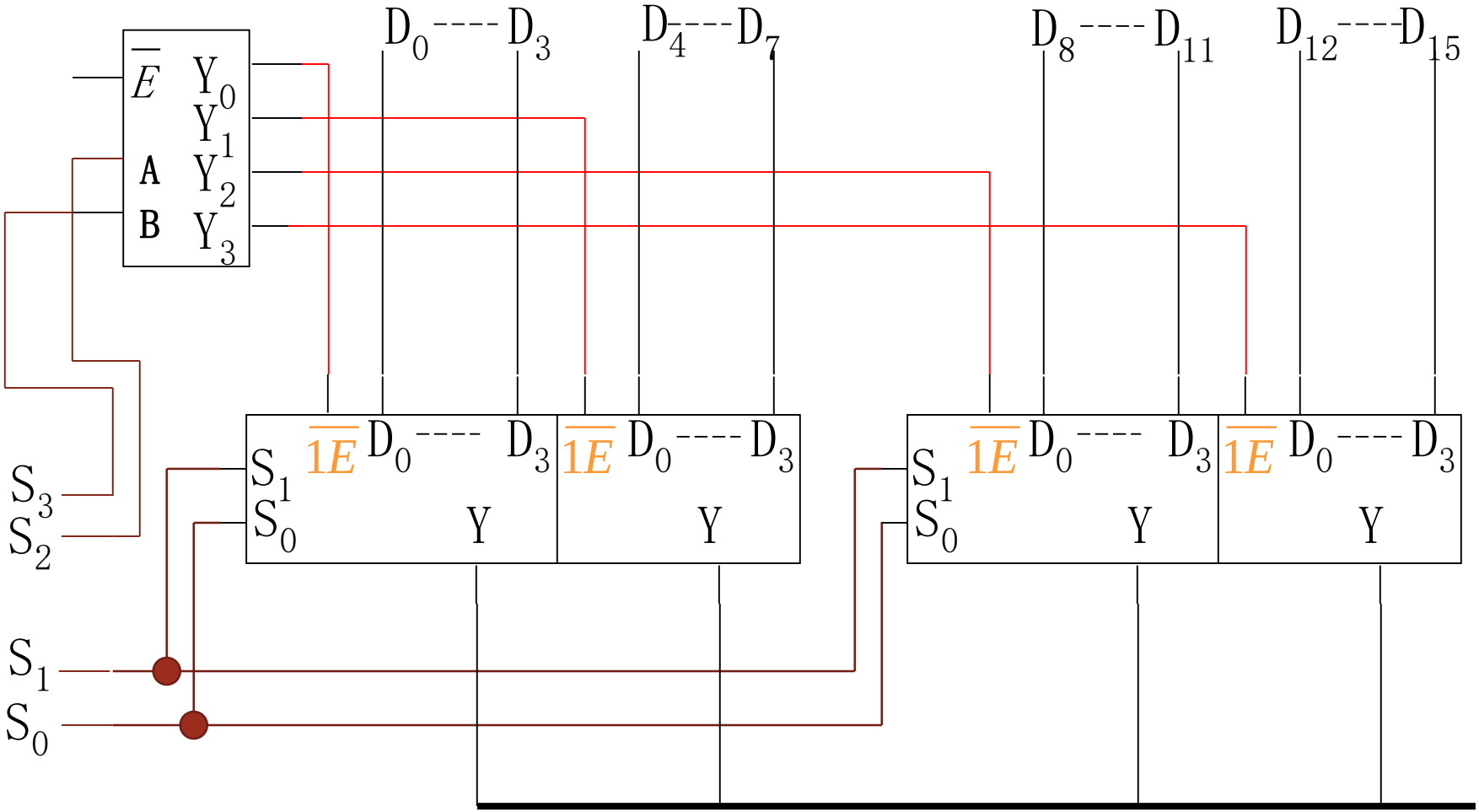
选择器扩展： 用双4选1选择器(无使能端)扩展成16选1选择器

两级选择结构



逻辑结构：S_1S_0控制第一层选择，S_3S_2控制第二层选择。

选择器扩展： 用双4选1选择器(有使能端)扩展成16选1选择器



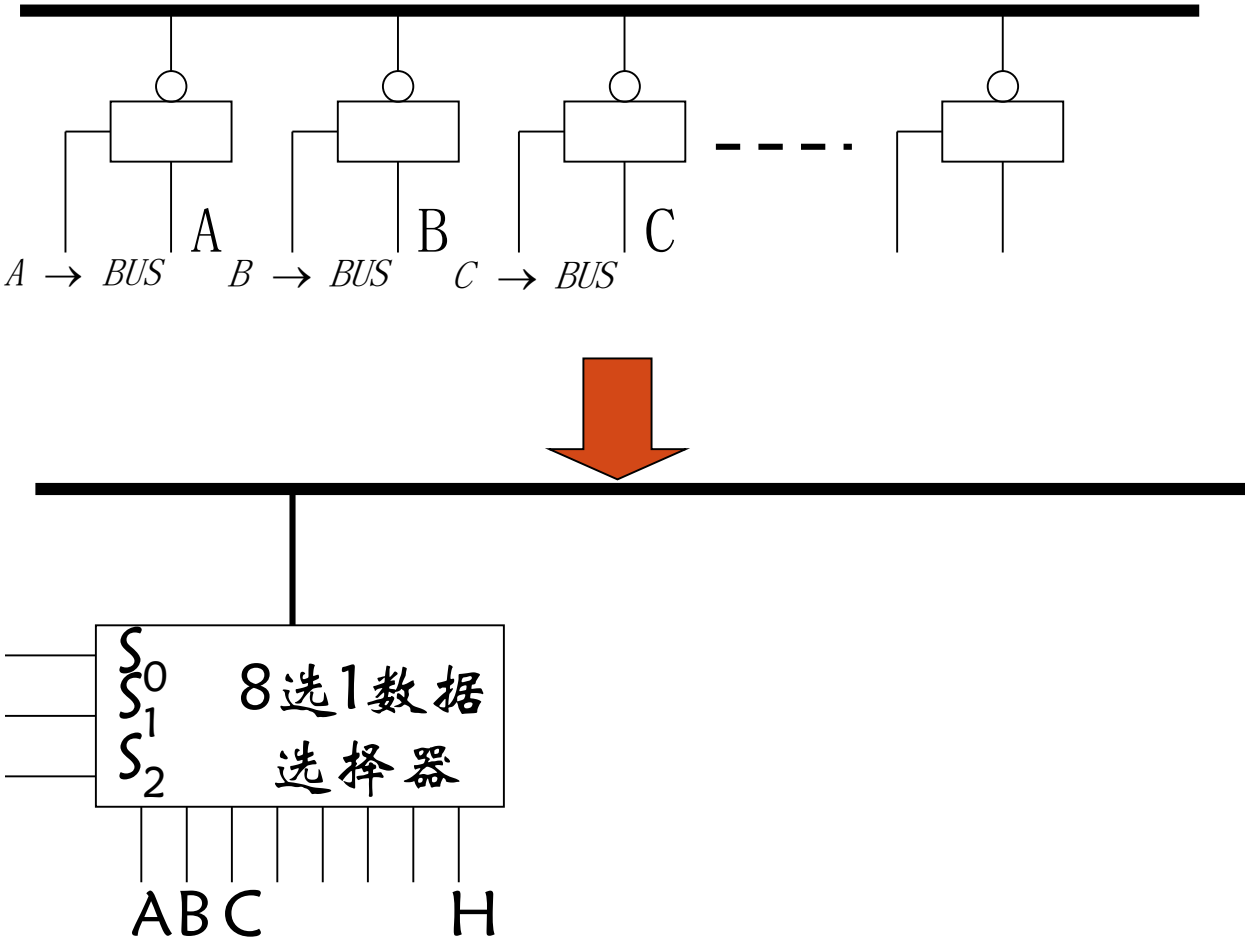
高两位控制端经译码后分别控制数据选择器的使能端E，以实现扩展。输出级是三态门，因此可以“线与”。

3.3.3 数据选择器(6)

- 同学们想一想，还有没有其它扩展方法？

3.3.2 数据选择器(7)

- 数据选择器用于总线发送控制



3.3.2 数据选择器(8)

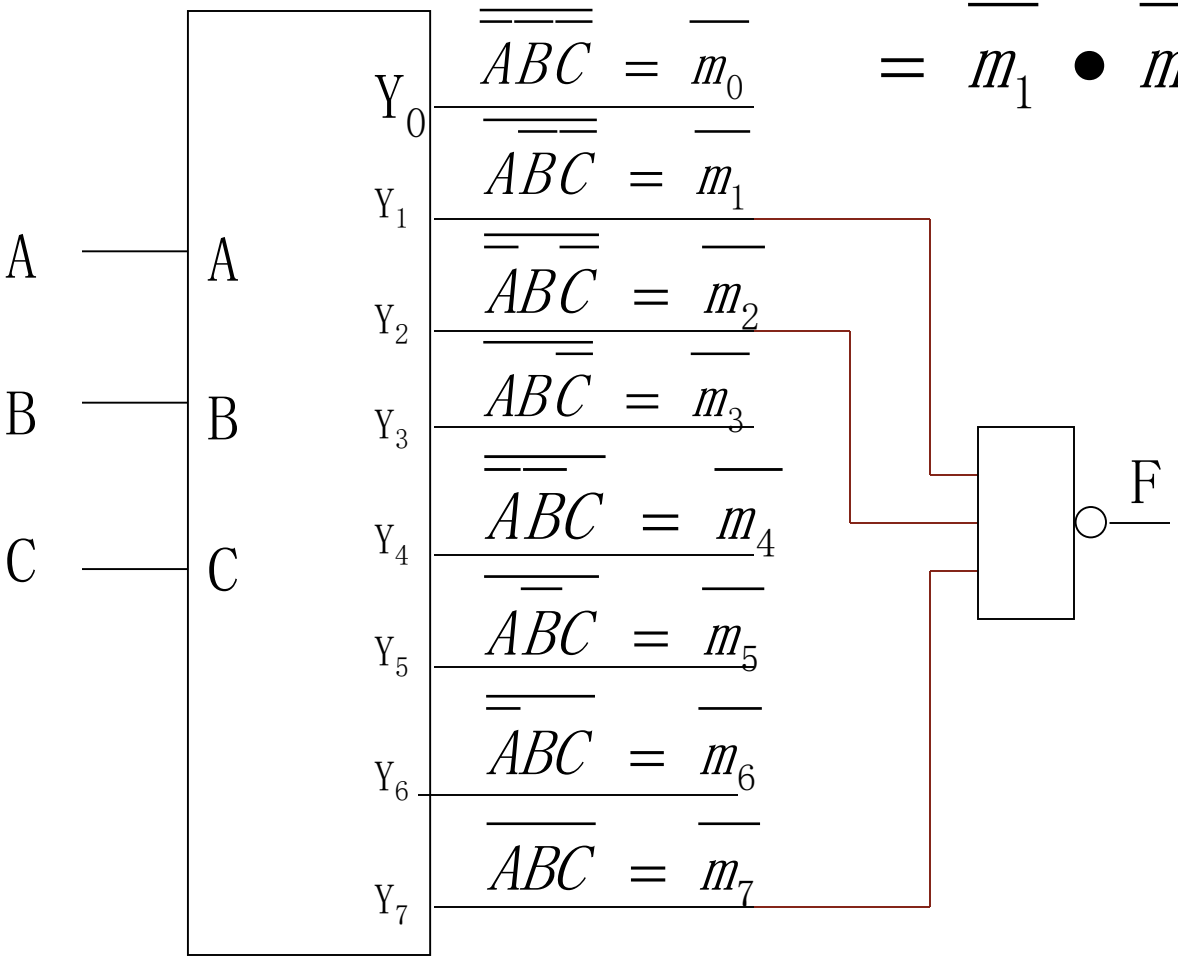
- 译码器与数据选择器实现逻辑函数
  - 译码器：可以看成是N个输入变量组成的2^N个最小项。
  - 如果再加一级与非门，可组成“与非-与非”逻辑，也可表达“与-或”逻辑。即可用译码器实现“与-或”逻辑函数。
- 如何用译码器实现如下函数：

$$F = A\overline{B}\overline{C} + \overline{A}B\overline{C} + ABC = m_1 + m_2 + m_7$$

3.3.2 数据选择器(9)

$$F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + A\overline{B}C + ABC = m_1 + m_2 + m_7$$

$$= \overline{m_1} \bullet \overline{m_2} \bullet \overline{m_7}$$



3.3.2 数据选择器(10)

- 译码器与数据选择器实现逻辑函数
  - 数据选择器：逻辑结构就是与-或表达式。
    - 如：4选1选择器：

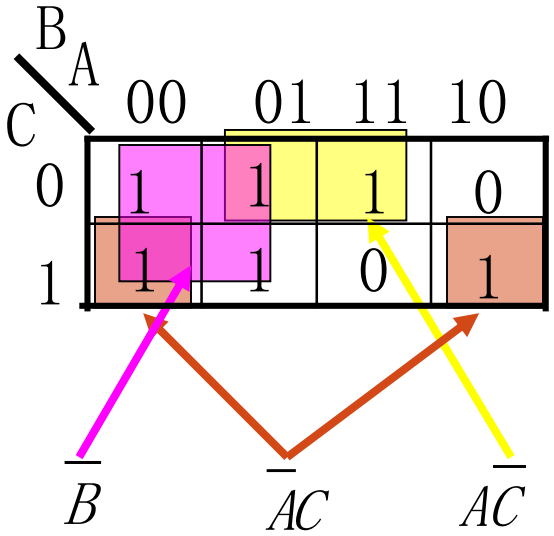
$$Y = \overline{S_0}\,\overline{S_1}D_0 + S_0\overline{S_1}D_1 + \overline{S_0}S_1D_2 + S_0S_1D_3$$

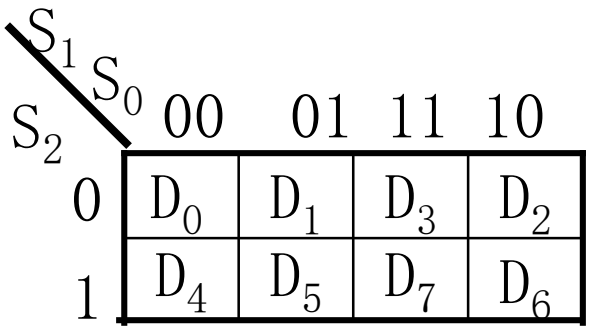
  - 数据选择器可以看成是N个控制端的2^N个最小项和2^N个输入组成的“与-或”表达式。选择某些输入为“1”，就是选中这些最小项组成逻辑函数。

3.3.2 数据选择器(11)

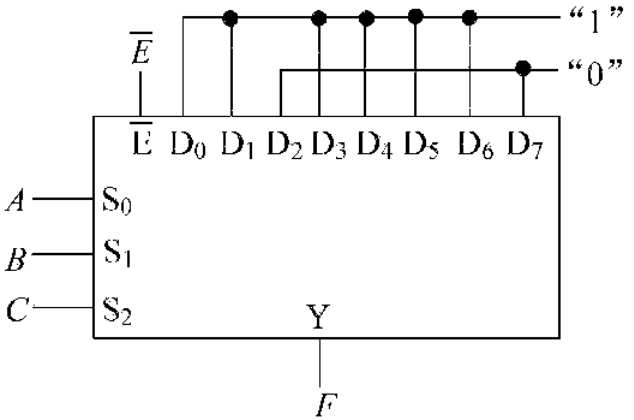
- 译码器与数据选择器实现逻辑函数
  - 用八选一数据选择器实现3变量函数：

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





八选一数据选择器功能的卡诺图



(c) 八选一数据选择器实现三变量函数的连接图

注意对应关系

3.3.2 数据选择器(12)

- 译码器与数据选择器实现逻辑函数
  - 8选1数据选择器可以实现4变量函数：

例：$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN$

思：3个变量用在选择控制端

总共4个变量，3个用在选择端，另一个变量怎么办？

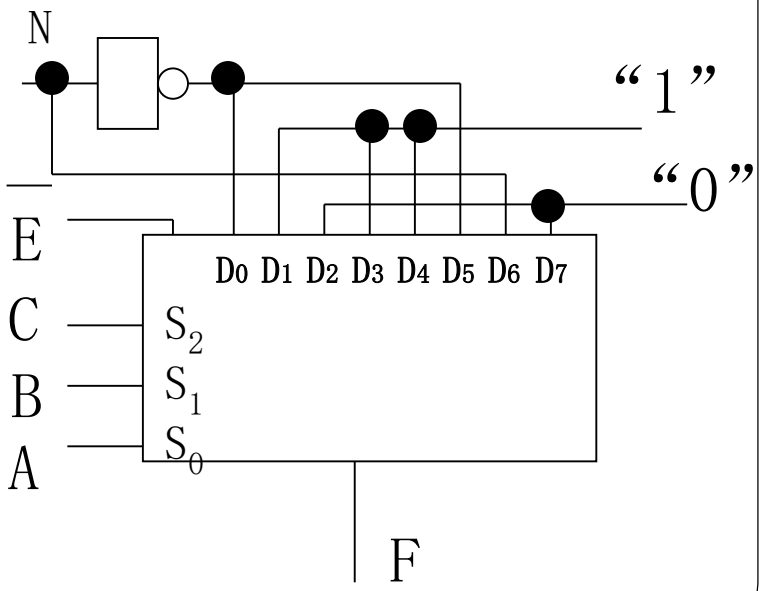
1个变量在数据输入端！

3.3.2 数据选择器(13)

- 译码器与数据选择器实现逻辑函数：8选1数据选择器可以实现4变量函数 $F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN$

令A=S_0, B=S_1, C=S_2

S_0 S_1 S_2	输入	函数F的值
0 0 0	D_0	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = \overline{N}$
1 0 0	D_1	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = 1$
0 1 0	D_2	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = 0$
1 1 0	D_3	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = 1$
0 0 1	D_4	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = 1$
1 0 1	D_5	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = \overline{N}$
0 1 1	D_6	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = N$
1 1 1	D_7	$F = A\overline{C} + \overline{B}\overline{N} + \overline{A}CN = 0$



- 作业：4.3, 4.5, 4.6, 4.10, 4.15, 4.22

4.3 分析图 4-83 的所示逻辑电路，列出逻辑表达式及功能表，说明电路实现的逻辑功能。

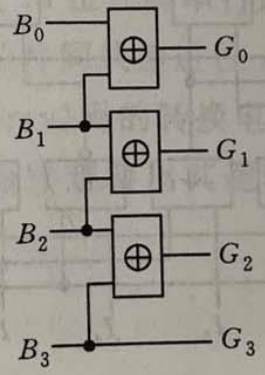


图 4-83

4.5 分析图 4-85 所示逻辑电路，列出 $K=1$，$K=0$ 时，输出的逻辑表达式，写出功能表，说明电路的逻辑功能。

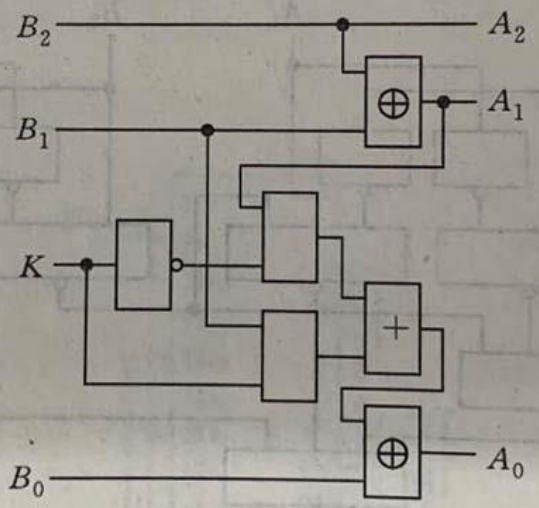


图 4-85

4.6　分析图 4-86 所示逻辑电路，写出输出的逻辑表达式，说明电路实现的逻辑功能。

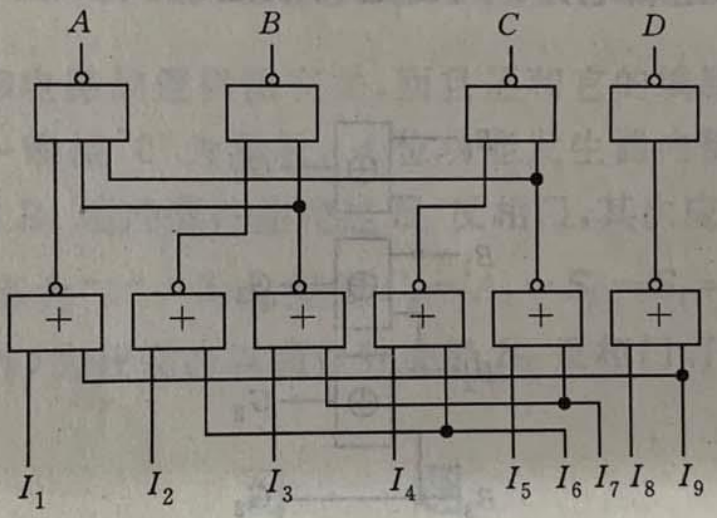


图　4-86

4.10　在宿舍只有一盏灯，同住在此宿舍的 3 位同学要求在各自床头安装开关均能独立地控制灯的关或开。用最少的门电路设计一个控制电路满足 3 位同学的要求。

4.15 用 3 个 2 输入 4 输出变量译码器实现一个非完全译码的 BCD 译码器(不再用其他门)。

4.22 设计一个 10 选 1 数据选择器,要求:

(1) 用门电路实现;

(2) 用集成电路实现——只能用一块 8 选 1 数据选择器和一块 4 位 2 选 1 数据选择器。