数字电路与逻辑设计

第 4 章 组合逻辑电路

张江山

zhangjs@hust.edu.cn

信息工程系

4.1 组合逻辑电路的分析

4.2 组合逻辑电路的设计

4.3 组合逻辑电路中的竞争和冒险

4.4 常用组合逻辑集成电路

4.5 组合可编程电路

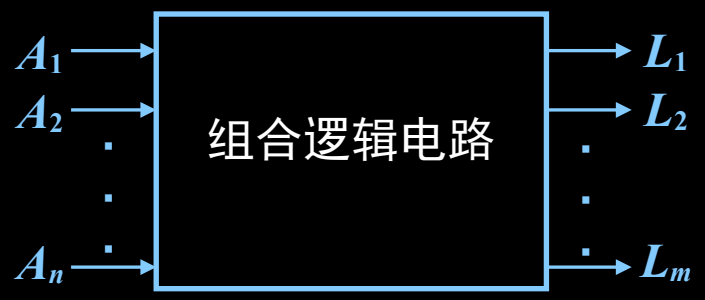
4.6 用 Verilog HDL 描述组合逻辑电路

教学要求

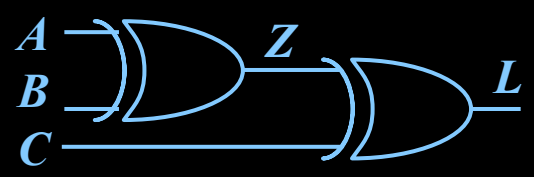
1. 掌握组合逻辑电路的分析方法和设计方法
2. 掌握组合逻辑电路出现竞争和冒险的原因及消除方法
3. 根据 MSI 器件数据手册，掌握典型集成电路器件的应用方法
4. 掌握 PLD 的表示方法及应用方法

数字逻辑电路
- 组合逻辑电路——无状态记忆功能
- 时序逻辑电路——有状态记忆功能

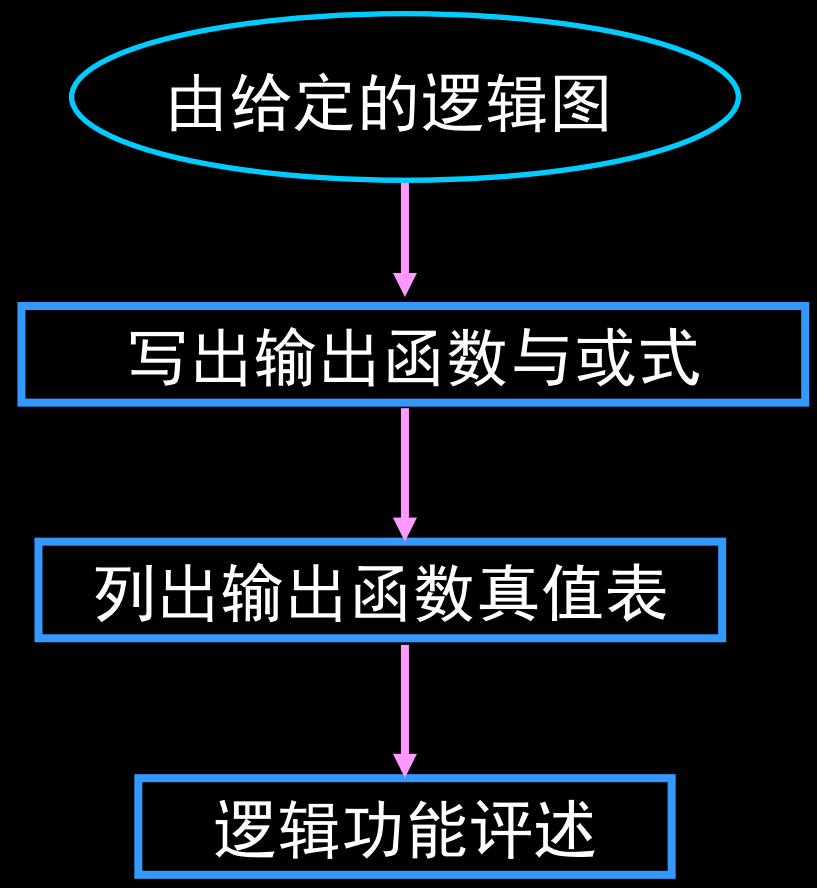
- 组合逻辑电路：输出仅取决于此时的输入，电路无状态
- 特点：不含状态记忆元件，电路无状态记忆功能



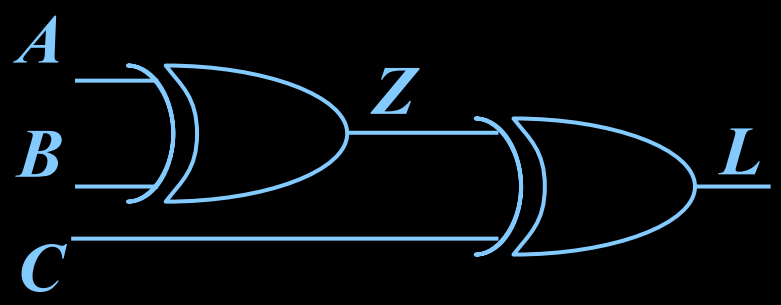
$$L_i = f_i(A_1, A_2, \ldots, A_n), i = \{1, 2, \ldots, m\}$$



- 逻辑电路分析：电路→功能



例：逻辑电路如图，分析其功能



(1) 写出输出函数与或式

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

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

$$= \sum m\,(1, 2, 4, 7)$$

(2) 列真值表

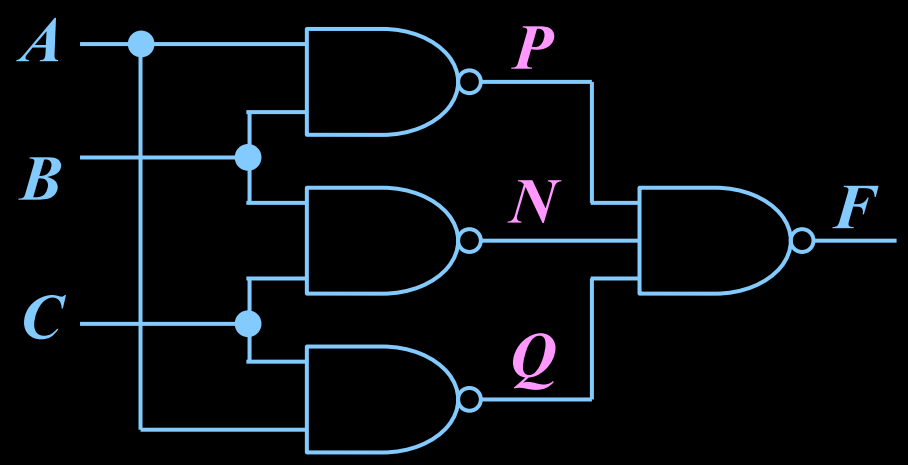
A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(3) 功能评述

三位输入有奇数个 1 时，L 为 1，否则为 0，是三位奇校验电路。

也是偶校验位生成电路，即：L 校验位要确保 $ABCL$ 四位有偶数个 1

例：逻辑电路如图，分析其功能



(2) 列真值表

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(1) 写出输出函数与或式

$$F=\overline{\overline{AB}\cdot\overline{BC}\cdot\overline{AC}}=\underline{AB}+\underline{BC}+\underline{AC}$$

(3) 功能评述

三位输入有 2 位及更多位为 1 时，输出 1，否则输出 0

即三输入多数表决器

例：逻辑电路如图，分析其功能

(1) 写出输出函数与或式

$$X = A$$

$$Y = \overline{P \bullet Q}$$

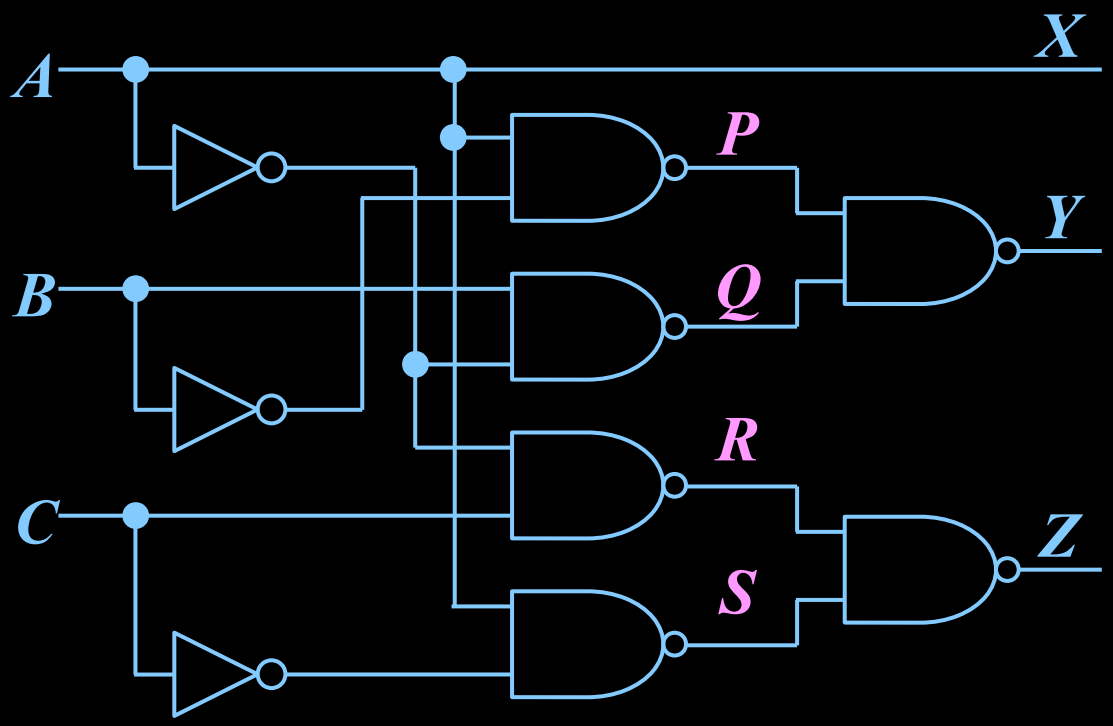
$$= \overline{\overline{A \cdot \overline{B}} \cdot \overline{\overline{A} \cdot B}}$$

$$= A \bullet \overline{B} + \overline{A} \bullet B$$

$$Z = \overline{R \bullet S}$$

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

$$= \overline{A} \bullet C + A \bullet \overline{C}$$



例：逻辑电路如图，分析其功能

(1) 写出输出函数与或式

$$X = A$$

$$Y = \overline{P \cdot Q}$$

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

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

$$Z = \overline{R \cdot S}$$

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

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

(2) 列真值表

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

(3) 功能评述

输入三位有符号数原码，最高位为符号位，0 为正，1 为负。

输出正数的原码，负数的反码。

4.2 组合逻辑电路的设计

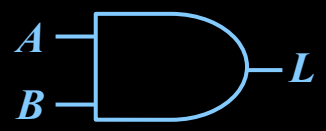
- 逻辑电路设计：给定功能，设计电路，又称逻辑电路综合
- 组合逻辑电路设计一般步骤

1. 逻辑抽象：根据实际问题的因果关系确定输入、输出变量，并定义逻辑状态的含义
2. 根据逻辑描述列出真值表
3. 由真值表写出逻辑表达式，并按器件要求进行化简、变换
4. 检查避免竞争冒险，绘制逻辑电路图

4.2.1 组合逻辑电路中的竞争冒险

- 竞争：逻辑门电路两输入信号，理论上同时发生反相变化，在实际电路中因这些信号经由的逻辑门数量各不相同，使该变化存在时间差异，由此导致产生错误脉冲的现象。

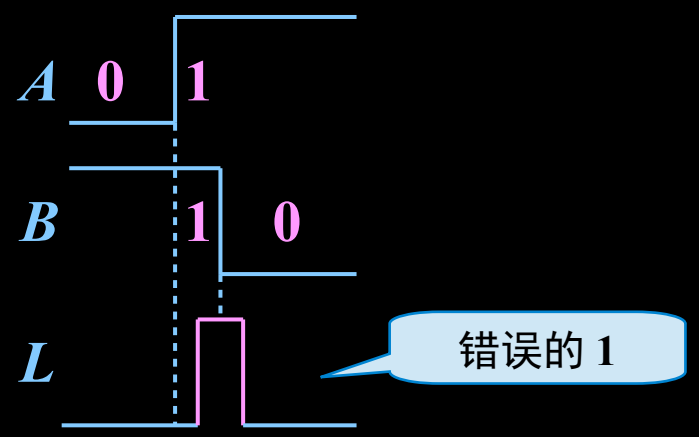
例如：$L = A \cdot B$



若 $A = 0$，$B = 1$，则 $L = 0$

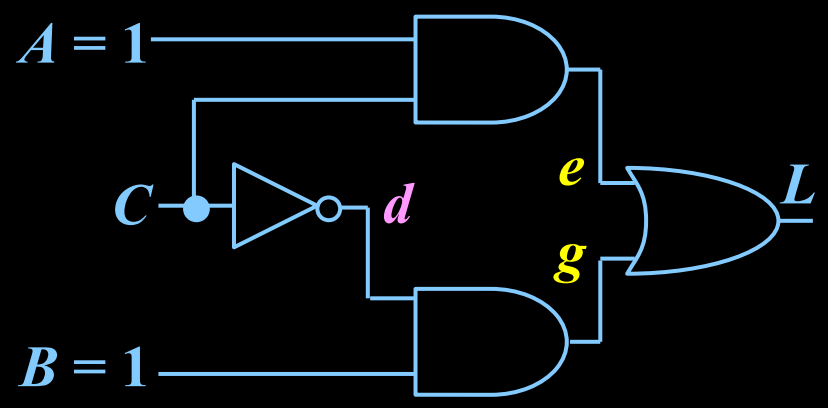
当 A 和 B 同时反相，即：A 由 0 变为 1，同时 B 由 1 变为 0，理论上这种输入信号的同时反相，不会影响与门输出 $L = 0$ 的结果。

但因前级电路时延差异，若使 B 的变化滞后于 A，则会输出错误的 1

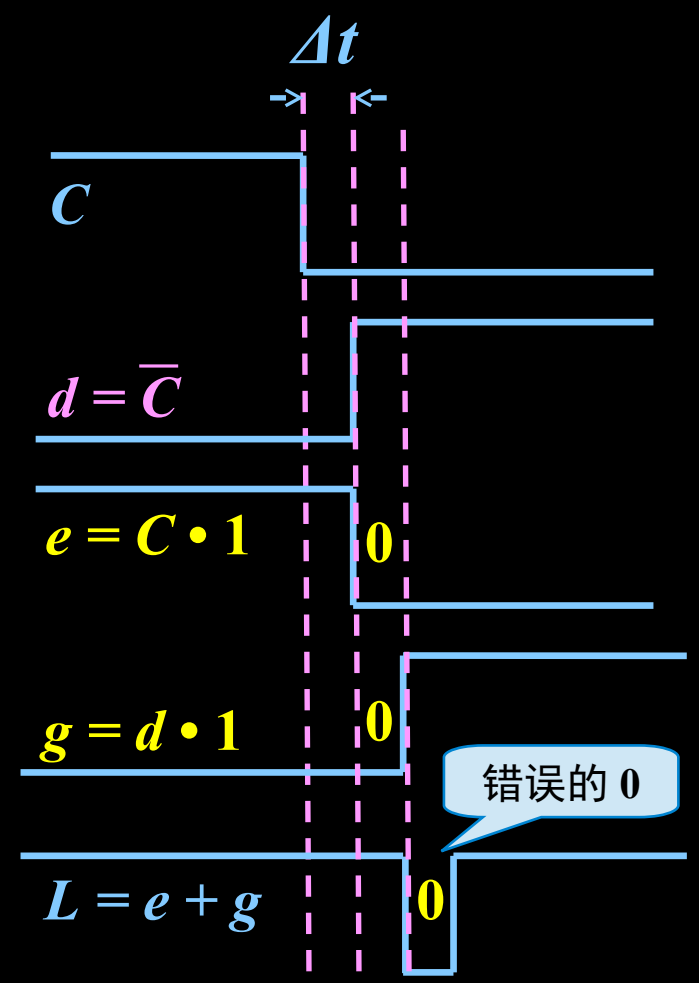


- **冒险**：由输出函数中的互反变量引起的信号竞争，在特定输入时，使输出 $L=\overline{C}\cdot C$ 或 $L=\overline{C}+C$ 出现错误的 1 或 0，即因互反变量信号的竞争导致冒险现象。可变换输出函数来避免电路的冒险现象。

如下二选一电路，设各逻辑门时延都为 Δt，在 $A=B=1$ 时，互反变量 C 和 $\overline{C}$ 的变化会产生竞争冒险现象



$L=A\,C+B\,\overline{C}$



$A=B=1$ 时，代入 L 函数得 $L=C+\overline{C}=1$

理论上此时 C 的变化不影响 $L=1$

实际上 C 和 $\overline{C}$ 存在竞争，使 L 产生错误的 0

1. 代数法检查竞争冒险

注意：须保持原函数式不变，否则电路结构就变了

(1) 检查函数式中是否存在可能引起竞争冒险的互反变量，如 X 和 $\overline{X}$

(2) 穷举其他变量的值，分别代入原函数式后得：

- $F = X + \overline{X}$ 则会产生错误 0
- $F = X \cdot \overline{X}$ 则会产生错误 1

例：试判断 $F = \overline{A}\,\overline{C} + \overline{A}\,B + A\,C$ 描述逻辑电路是否产生竞争冒险

可能引起竞争冒险的互反变量：$\overline{A}$ 和 A，$\overline{C}$ 和 C

考察 A 和 $\overline{A}$：穷举 B 和 C 的值代入 F

BC 为 00 时，$F = \overline{A}$

BC 为 01 时，$F = A$

BC 为 10 时，$F = \overline{A}$

BC 为 11 时，$F = A + \overline{A}$

$B = 1, C = 1$ 时，变化 A 会产生错误 0

考察 C 和 $\overline{C}$：穷举 A 和 B 的值代入 F

AB 为 00 时，$F = \overline{C}$

AB 为 01 时，$F = 1$

AB 为 10 时，$F = C$

AB 为 11 时，$F = C$

变化 C 不会产生竞争冒险

2. 卡诺图法检查并消除与或式逻辑电路的竞争冒险

(1) 作卡诺图

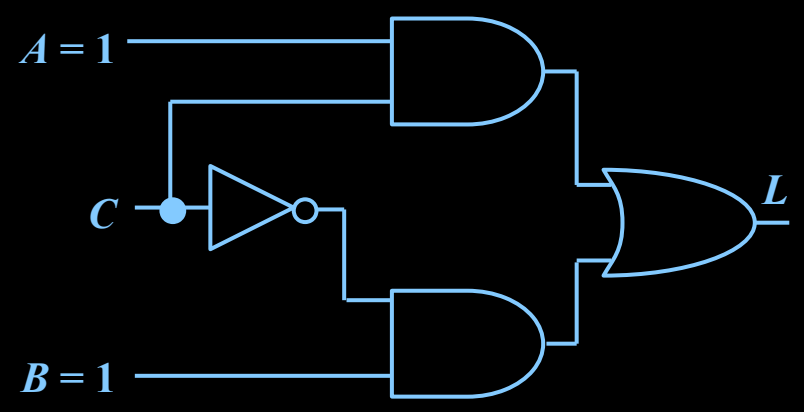
注意：须保持原函数式不变，否则电路结构就变了

(2) 画出与或式中各与项对应的卡诺圈

(3) 若有卡诺圈相切，则该电路在切点处，互反因子会引起竞争冒险

(4) 新增覆盖上述切点的冗余圈，可消除对应互反因子引起的竞争冒险

例：二选一选择器 $L = A\,C + B\,\overline{C}$



A \ BC	00	01	11	10
0	0	0	0	1
1	0	1	1	1

相切

$A = B = 1$ 时，$L = C + \overline{C}$

C 变化会产生错误 0

$L = A\,C + B\,\overline{C} + A\,B$

$A = B = 1$ 时，$L = C + \overline{C} + 1$

C 变化不再产生竞争冒险

2. 卡诺图法检查并消除与或式逻辑电路的竞争冒险

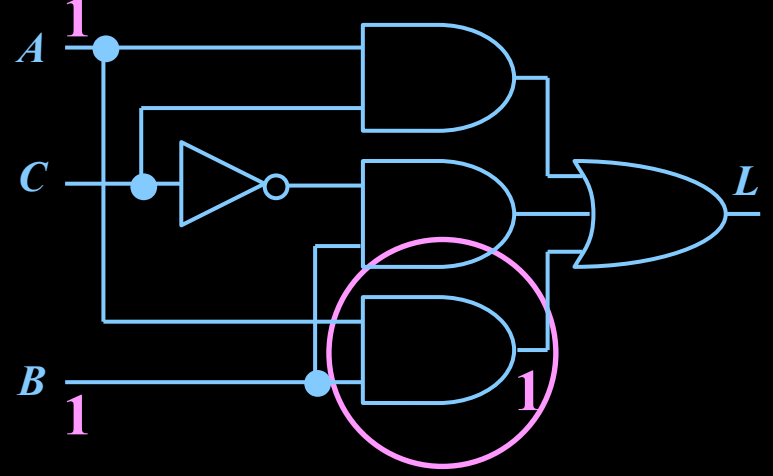
(1) 作卡诺图

(2) 画出与或式中各与项对应的卡诺圈

(3) 若有卡诺圈相切，则该电路在切点处，互反因子会引起竞争冒险

(4) 新增覆盖上述切点的冗余圈，可消除对应互反因子引起的竞争冒险

例：二选一选择器 $L = A\,C + B\,\overline{C}$



A \ BC	00	01	11	10
0	0	0	0	1
1	0	1	1	1

相切

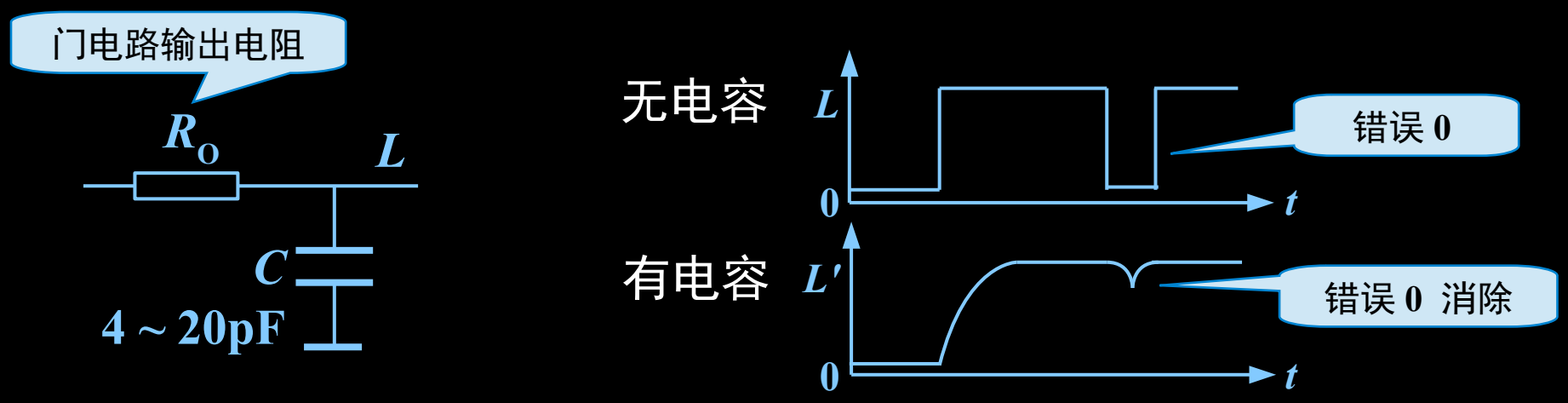
$L = A\,C + B\,\overline{C} + A\,B$

$A = B = 1$ 时，$L = C + \overline{C} + 1$

C 变化不再产生竞争冒险

3. 输出端并联电容器

若逻辑电路工作速度较慢时，为消去竞争冒险，可在输出端并联一电容，使输出波形上、下沿变化较缓，对窄负跳变脉冲起平波作用



例：用与非门设计一个三输入表决器，其中 A 输入具有否决权

1. 逻辑抽象

分析要求，确定输入输出变量

输入：A, B, C　　0 反对，1 赞成

输出：F　　0 否决，1 通过

2. 列真值表

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

要求用与非门实现，变换成“与非”形式

$$F = \overline{\overline{AB + AC}}$$

$$= \overline{\overline{AB} \cdot \overline{AC}}$$

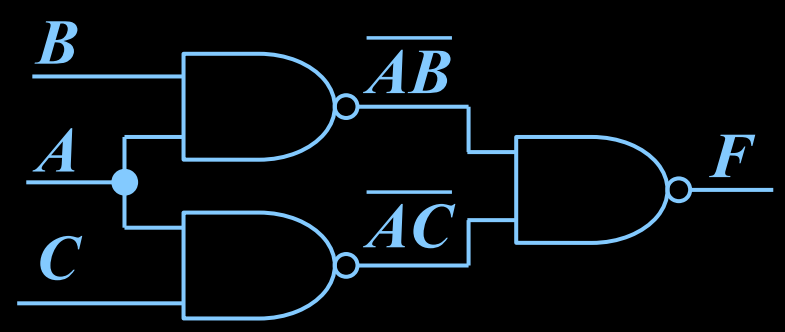
无互反变量
无竞争冒险

3. 写表达式并化简　$F = AB + AC$

A \ BC	00	01	11	10
0	0	0	0	0
1	0	1	1	1

F 的卡诺图

4. 画逻辑电路图



例：试设计一个码转换电路，将 4 位格雷码转换为二进制自然码。

解：(1) 逻辑抽象

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

(2) 列真值表

真值表

输入格雷码 $G_3\ G_2\ G_1\ G_0$	输出二进制 $B_3\ B_2\ B_1\ B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

(3) 用卡诺图化简输出函数和变换

B_3

G_3G_2 \ G_1G_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$B_3 = G_3$

B_2

G_3G_2 \ G_1G_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$\overline{G}_3\ G_2$

$G_3\ \overline{G}_2$

无相切圈
无竞争冒险

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

真值表

输入格雷码 $G_3\ G_2\ G_1\ G_0$	输出二进制 $B_3\ B_2\ B_1\ B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

4.2.2 组合逻辑电路的设计过程

B_1 G_1G_0

G_3G_2	00	01	11	10	
00	0	0	1	1	$\overline{G}_3\,\overline{G}_2\,G_1$
01	1	1	0	0	$\overline{G}_3\,G_2\,\overline{G}_1$
11	0	0	1	1	$G_3\,G_2\,G_1$
10	1	1	0	0	$G_3\,\overline{G}_2\,\overline{G}_1$

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

B_0 G_1G_0

G_3G_2	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

无相切圈，无竞争冒险

无相切圈，无竞争冒险

$$B_0 = \overline{G}_3\,\overline{G}_2\,\overline{G}_1\,G_0 + \overline{G}_3\,\overline{G}_2\,G_1\,\overline{G}_0 + \overline{G}_3\,G_2\,\overline{G}_1\,\overline{G}_0 + \overline{G}_3\,G_2\,G_1\,G_0 +$$
$$G_3\,G_2\,\overline{G}_1\,G_0 + G_3\,G_2\,G_1\,\overline{G}_0 + G_3\,\overline{G}_2\,\overline{G}_1\,\overline{G}_0 + G_3\,\overline{G}_2\,G_1\,G_0$$

真值表

输入格雷码 $G_3\,G_2\,G_1\,G_0$	输出二进制 $B_3\,B_2\,B_1\,B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

4.2.2 组合逻辑电路的设计过程

$$B_3 = G_3$$

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

$$= G_3 \oplus G_2 = B_3 \oplus G_2$$

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

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

$$= \overline{G}_3\,(G_2 \oplus G_1) + G_3\,(G_2 \odot G_1)$$

$$= G_3 \oplus G_2 \oplus G_1 = B_2 \oplus G_1$$

$$B_0 = \overline{G}_3\, \overline{G}_2\, \overline{G}_1\, G_0 + \overline{G}_3\, \overline{G}_2\, G_1\, \overline{G}_0 + \overline{G}_3\, G_2\, \overline{G}_1\, \overline{G}_0 + \overline{G}_3\, G_2\, G_1\, G_0 +$$

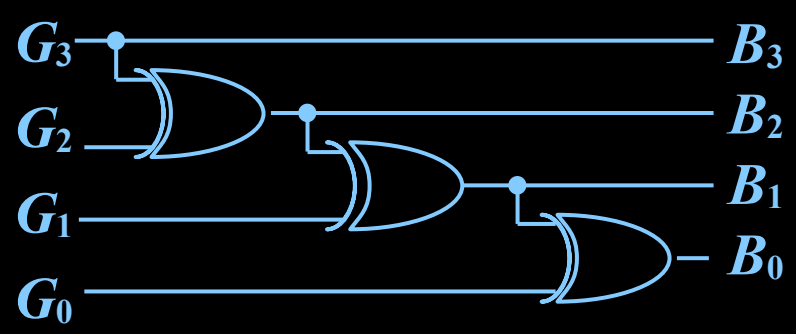
$$G_3\, G_2\, \overline{G}_1\, G_0 + G_3\, G_2\, G_1\, \overline{G}_0 + G_3\, \overline{G}_2\, \overline{G}_1\, \overline{G}_0 + G_3\, \overline{G}_2\, G_1\, G_0$$

$$= \overline{G}_3\, \overline{G}_2\,(G_1 \oplus G_0) + \overline{G}_3\, G_2\,(G_1 \odot G_0) + G_3\, G_2\,(G_1 \oplus G_0) + G_3\, \overline{G}_2\,(G_1 \odot G_0)$$

$$= (G_3 \odot G_2)(G_1 \oplus G_0) + (G_3 \oplus G_2)(G_1 \odot G_0)$$

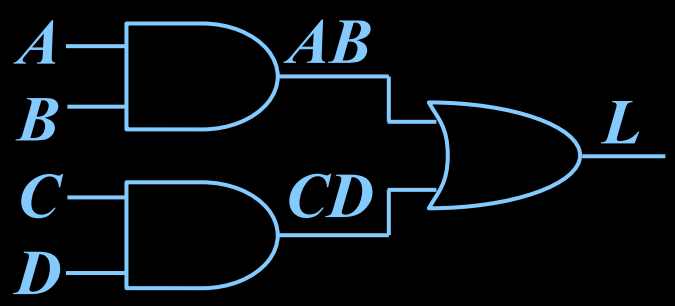
$$= G_3 \oplus G_2 \oplus G_1 \oplus G_0 = B_1 \oplus G_0$$

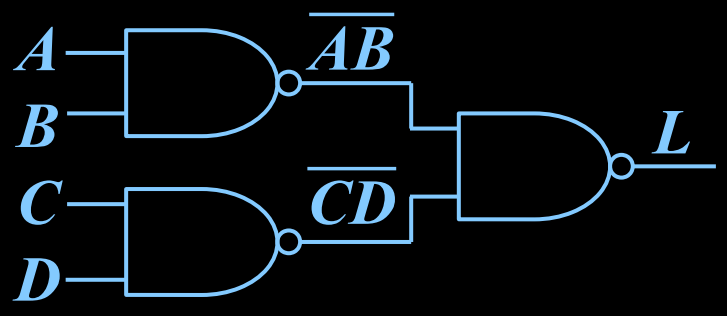
(4) 画出逻辑图



- 单输出电路（单逻辑函数式）优化
  - 除逻辑函数式化简优化外，还须考虑实际逻辑门集成电路的特性。
  - 如 CMOS 集成电路的与门、或门，一般是由与非、或非后接非门实现，故与非门、或非门相对于与门、或门，时延更小，成本更低。

$$L = AB + CD$$
$$= \overline{\overline{AB} \cdot \overline{CD}}$$

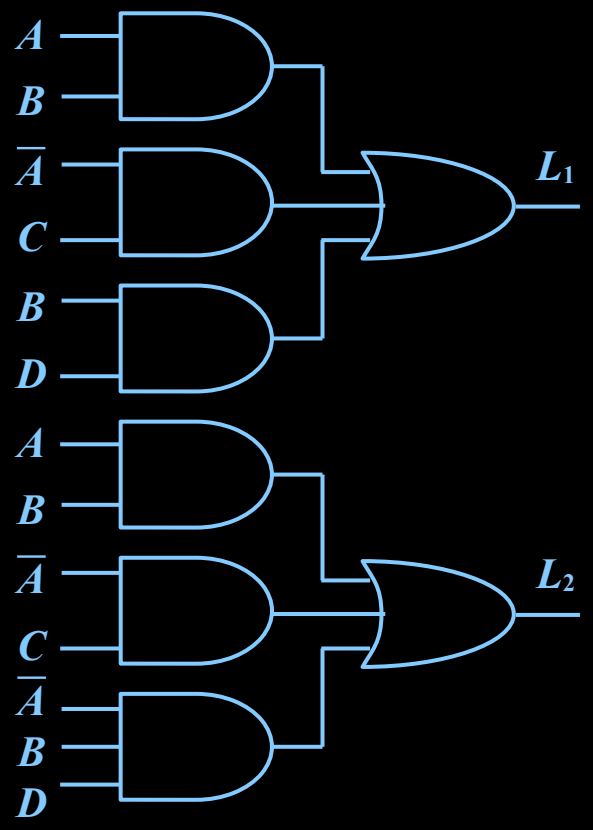


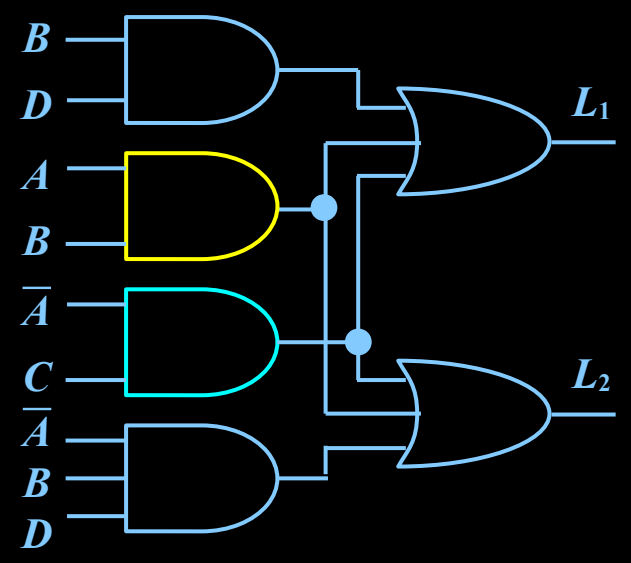


- 多输出电路（多逻辑函数式）优化
  - 除对各输出函数式化简等优化外，还须考虑各式是否能共享相同逻辑门

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

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





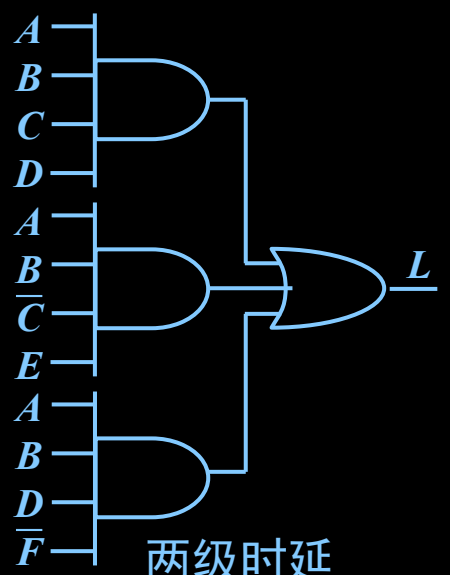
- 多级逻辑门时延电路优化
  - 一般与或、或与式是两级时延电路，适于实现变量较少的电路
  - 变量较多时，对逻辑门的扇入数要求较多，器件成本较高
  - 可用提取公因子、函数分解等变换，降低对逻辑门扇入数的要求

例：用扇入数不大于 3 的逻辑门实现 $L = ABCD + AB\overline{C}E + ABD\overline{F}$

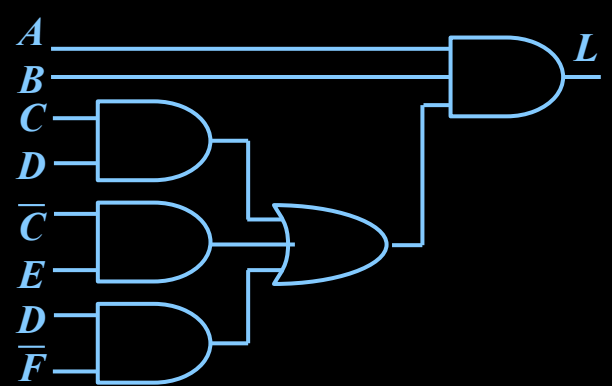
解：(1) 提取公因子得：$L = AB(CD + \overline{C}E + D\overline{F})$

(2) 函数分解得：$P = AB, Q = CD, R = \overline{C}E, S = D\overline{F}$;

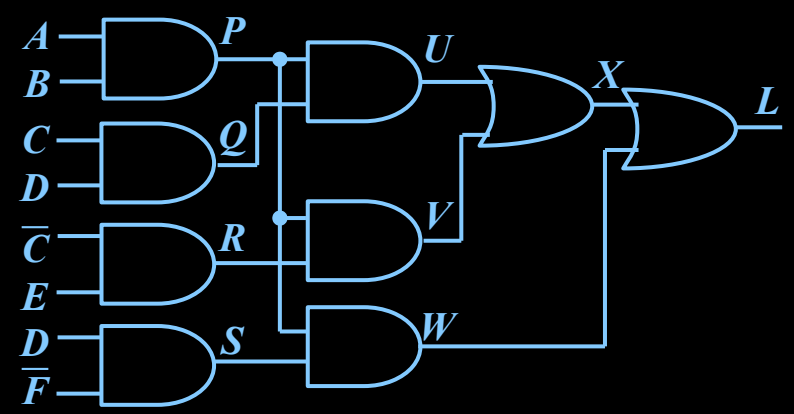
$U = PQ, V = PR, W = PS; X = U + V; L = X + W$



两级时延



三级时延



四级时延

4.4 若干典型组合逻辑集成电路

1. 典型集成电路考虑周到，功能齐全，值得学习和借鉴

(1) 具有各种使能、控制输入，提供各种信息输出

(2) 具有扩展、级联的功能，通用性强

2. 学习典型组合逻辑集成电路的重点是应用

(1) 快速阅读器件功能表，掌握器件逻辑功能

(2) 根据器件功能灵活应用

1. 编码器的定义与工作原理

编码：赋予二进制代码特定含义的过程称为编码

编码器：输入待编码有效信号，输出对应的二进制代码

普通编码器：输入的有效信号须是唯一的，否则输出为无效码

优先编码器：可输入多个有效信号，输出其中优先级最高的码

1. 编码器的定义与工作原理

(1) 4-2 线普通编码器

设输入高电平有效

Y_1 AB \ CD	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	0	0	0	0
10	1	0	0	0

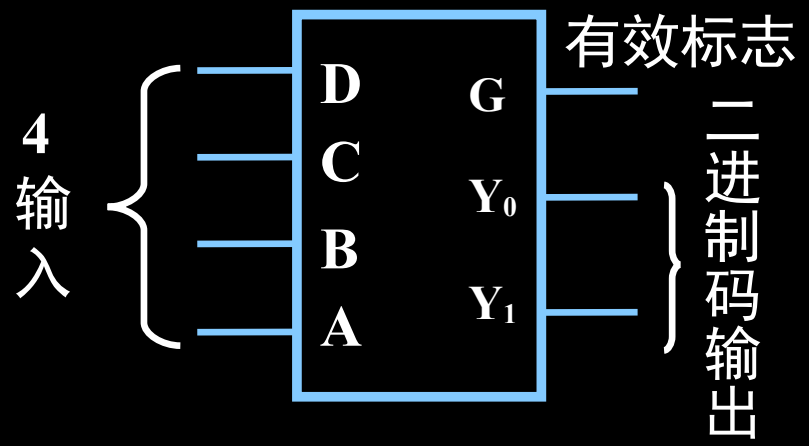
Y_0 AB \ CD	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	0

G AB \ CD	00	01	11	10
00	0	1	0	1
01	1	0	0	0
11	0	0	0	0
10	1	0	0	0

真值表

输入				输出		
A	B	C	D	Y_1	Y_0	G
1	0	0	0	1	1	1
0	1	0	0	1	0	1
0	0	1	0	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	0	0
others				0	0	0

逻辑框图



输出函数表达式：

$$Y_1 = A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}\,\overline{D}$$

$$Y_0 = A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D}$$

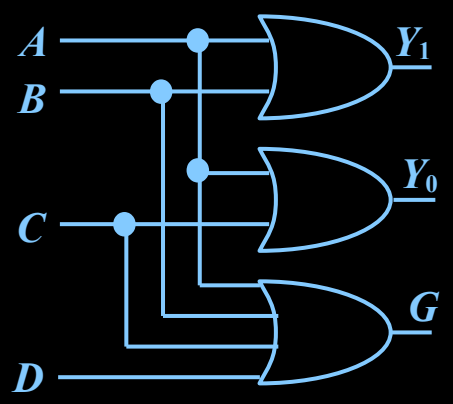
$$G = A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D$$

若输入多个有效信号，如 B 和 C 同时为 1 时， $G=0$，输出无效码 00

1. 编码器的定义与工作原理

(1) 4-2 线普通编码器

设输入高电平有效



真值表

输入				输出		
A	B	C	D	Y_1	Y_0	G
1	0	0	0	1	1	1
0	1	0	0	1	0	1
0	0	1	0	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	0	0
others				d	d	d

若按含无关项设计，输入信号须满足无关条件

$$AB + AC + AD + BC + BD + CD = 0$$

Y_1

$A\,B$ \ $C\,D$	00	01	11	10
00	0	0	d	0
01	1	d	d	d
11	d	d	d	d
10	1	d	d	d

Y_0

$A\,B$ \ $C\,D$	00	01	11	10
00	0	0	d	1
01	0	d	d	d
11	d	d	d	d
10	1	d	d	d

G

$A\,B$ \ $C\,D$	00	01	11	10
00	0	1	d	1
01	1	d	d	d
11	d	d	d	d
10	1	d	d	d

输出函数表达式：

$$Y_1 = A + B$$

$$Y_0 = A + C$$

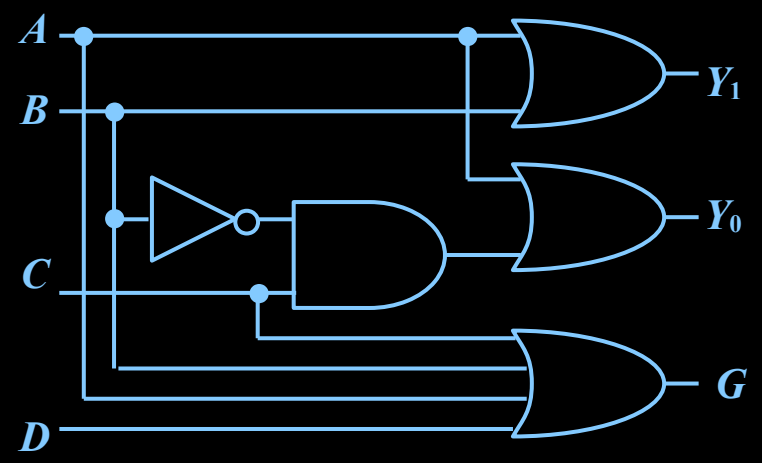
$$G = A + B + C + D$$

若输入不满足无关条件，如 B 和 C 同时为 1 时， $G = 1$，输出错误有效码 11

1. 编码器的定义与工作原理

(2) 4-2 线优先编码器

设输入高电平有效， A, B, C, D 优先序由高至低



真值表

输入				输出		
A	B	C	D	Y_1	Y_0	G
1	x	x	x	1	1	1
0	1	x	x	1	0	1
0	0	1	x	0	1	1
0	0	0	1	0	0	1
0	0	0	0	0	0	0

Y_1

A B \ C D	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Y_0

A B \ C D	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

G

A B \ C D	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

输出函数表达式：

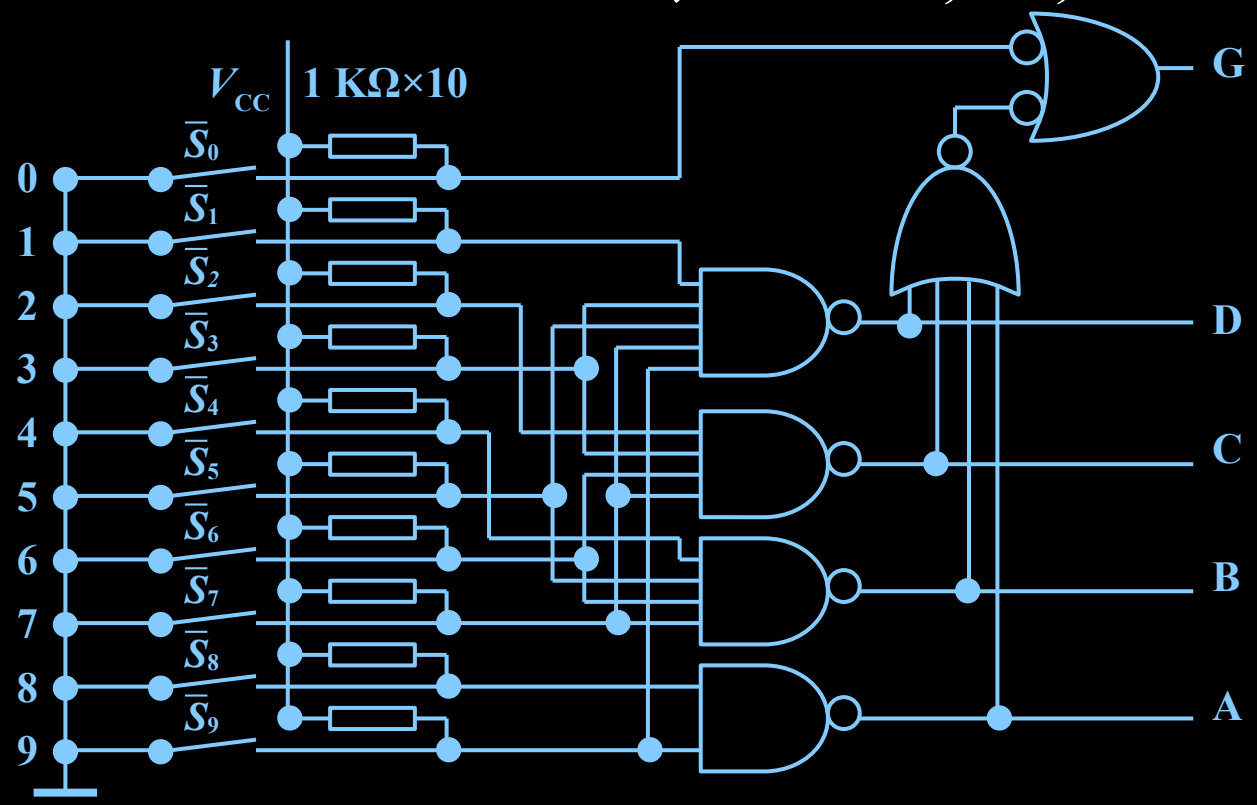
$$Y_1 = A + B$$

$$Y_0 = A + \overline{B}\,C$$

$$G = A + B + C + D$$

若输入多个有效信号，如 B 和 C 同时为 1 时， $G = 1$，输出 B 的有效码 10

例：分析图示电路功能，其中 $\overline{S}_0, \ldots, \overline{S}_9$ 分别为 0 , … , 9 十个数字按键



解：(1) 写出输出逻辑函数式

$$A=\overline{\overline{S}_9\overline{S}_8}$$

$$B=\overline{\overline{S}_7\overline{S}_6\overline{S}_5\overline{S}_4}$$

$$C=\overline{\overline{S}_7\overline{S}_6\overline{S}_3\overline{S}_2}$$

$$D=\overline{\overline{S}_9\overline{S}_7\overline{S}_5\overline{S}_3\overline{S}_1}$$

$$G=\overline{\overline{S}_0}+\overline{\overline{A+B+C+D}}$$

$$=\overline{S}_0+A+B+C+D$$

(2) 无键按下，输出 $ABCD$ 为 0000 , $G=0$

(3) 单键按下

仅 $\overline{S}_0=0$ 时，输出 $ABCD$ 为 0000 , $G=1$

仅 $\overline{S}_1=0$ 时，输出 $ABCD$ 为 0001 , $G=1$

……

仅 $\overline{S}_9=0$ 时，输出 $ABCD$ 为 1001 , $G=1$

(4) 多键按下

如仅 $\overline{S}_1=\overline{S}_2=0$ 时，输出 $ABCD$ 为 0011 , $G=1$

故该电路是数字键盘 8421BCD 普通编码器

$ABCD$ 是 8421BCD 码输出

$G=1$ 标志代码输出有效码，否则输出无效码

$$G=\overline{\overline{S}_0}+A+B+C+D$$

$$A=\overline{\overline{S}_9}\,\overline{\overline{S}_8}$$

$$B=\overline{\overline{S}_7}\,\overline{\overline{S}_6}\,\overline{\overline{S}_5}\,\overline{\overline{S}_4}$$

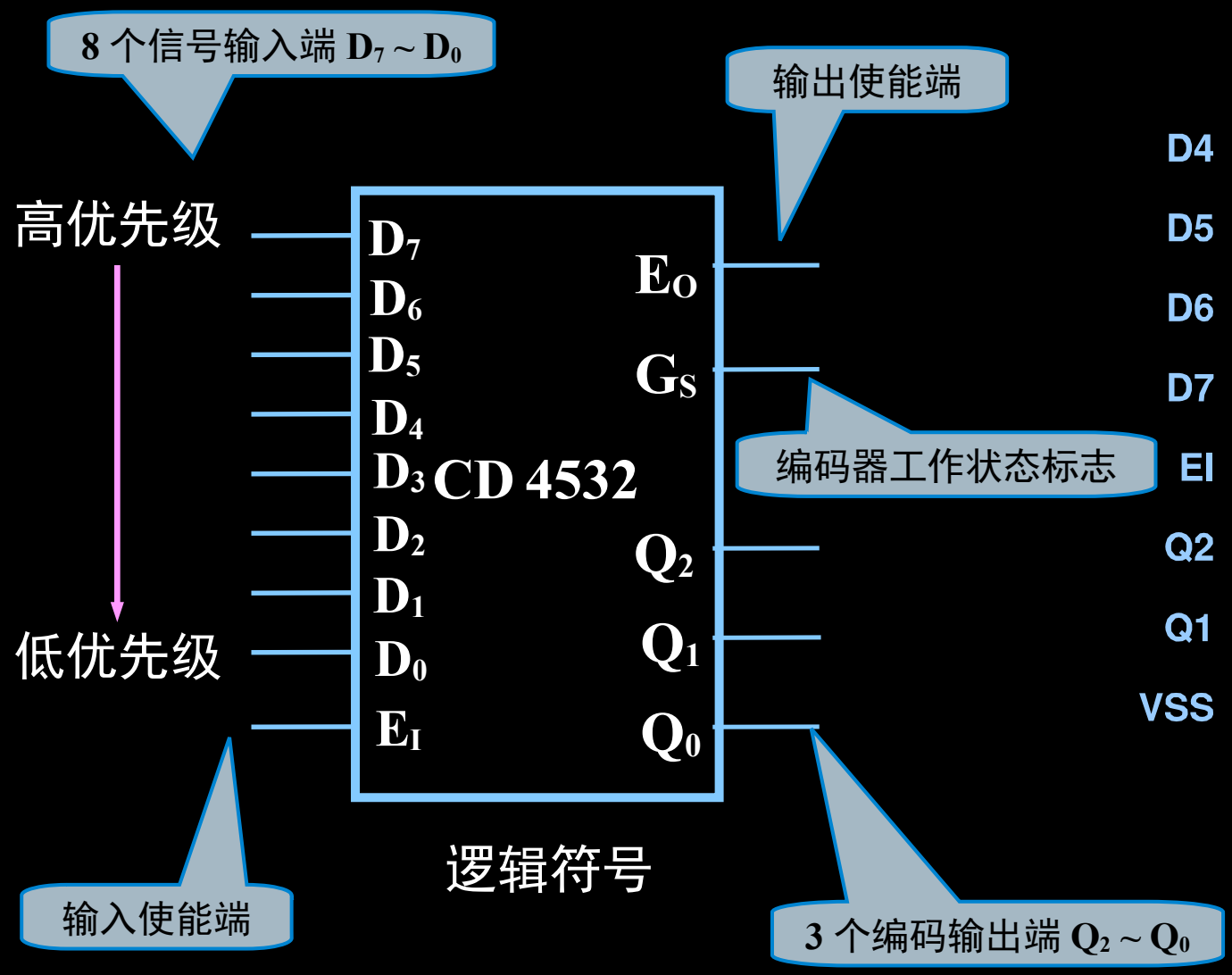
$$C=\overline{\overline{S}_7}\,\overline{\overline{S}_6}\,\overline{\overline{S}_3}\,\overline{\overline{S}_2}$$

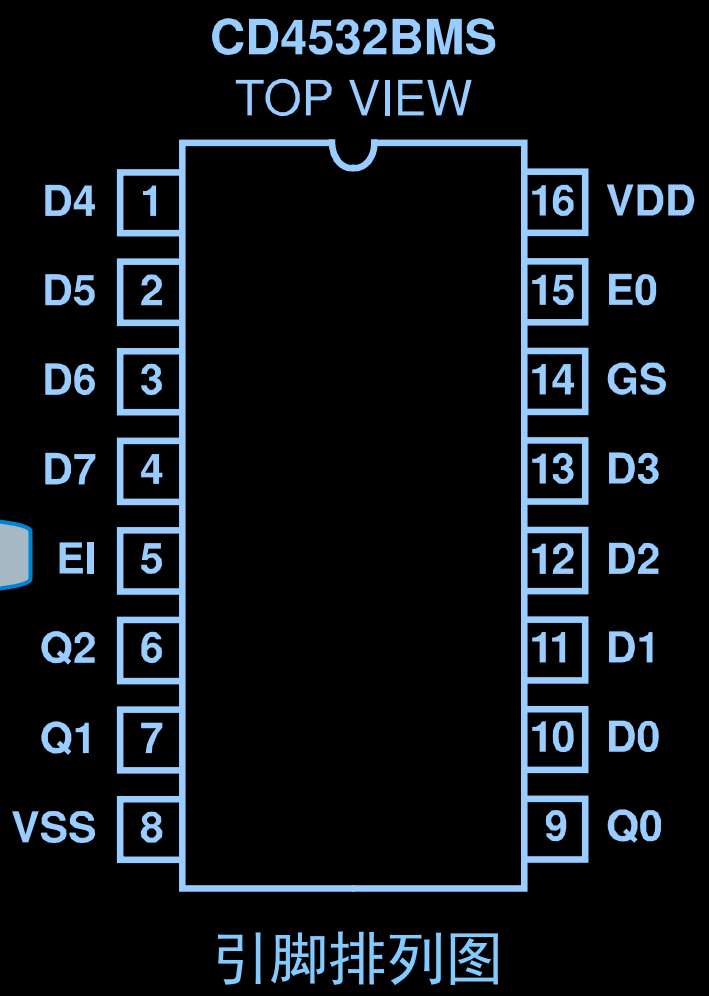
$$D=\overline{\overline{S}_9}\,\overline{\overline{S}_7}\,\overline{\overline{S}_5}\,\overline{\overline{S}_3}\,\overline{\overline{S}_1}$$

键盘输入 8421BCD 编码器 (输入低电平有效)

输入										输出				
$\overline{S}_9$	$\overline{S}_8$	$\overline{S}_7$	$\overline{S}_6$	$\overline{S}_5$	$\overline{S}_4$	$\overline{S}_3$	$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	A	B	C	D	G
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0	1	0	0	0	1	1
1	1	1	1	1	1	1	0	1	1	0	0	1	0	1
1	1	1	1	1	1	0	1	1	1	0	0	1	1	1
1	1	1	1	1	0	1	1	1	1	0	1	0	0	1
1	1	1	1	0	1	1	1	1	1	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1	0	1	1	0	1
1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	0	0	1	1

2. 集成电路编码器：优先编码器 CD4532





4.4.1 编码器

优先编码器 CD4532 功能表　　输入 H 有效

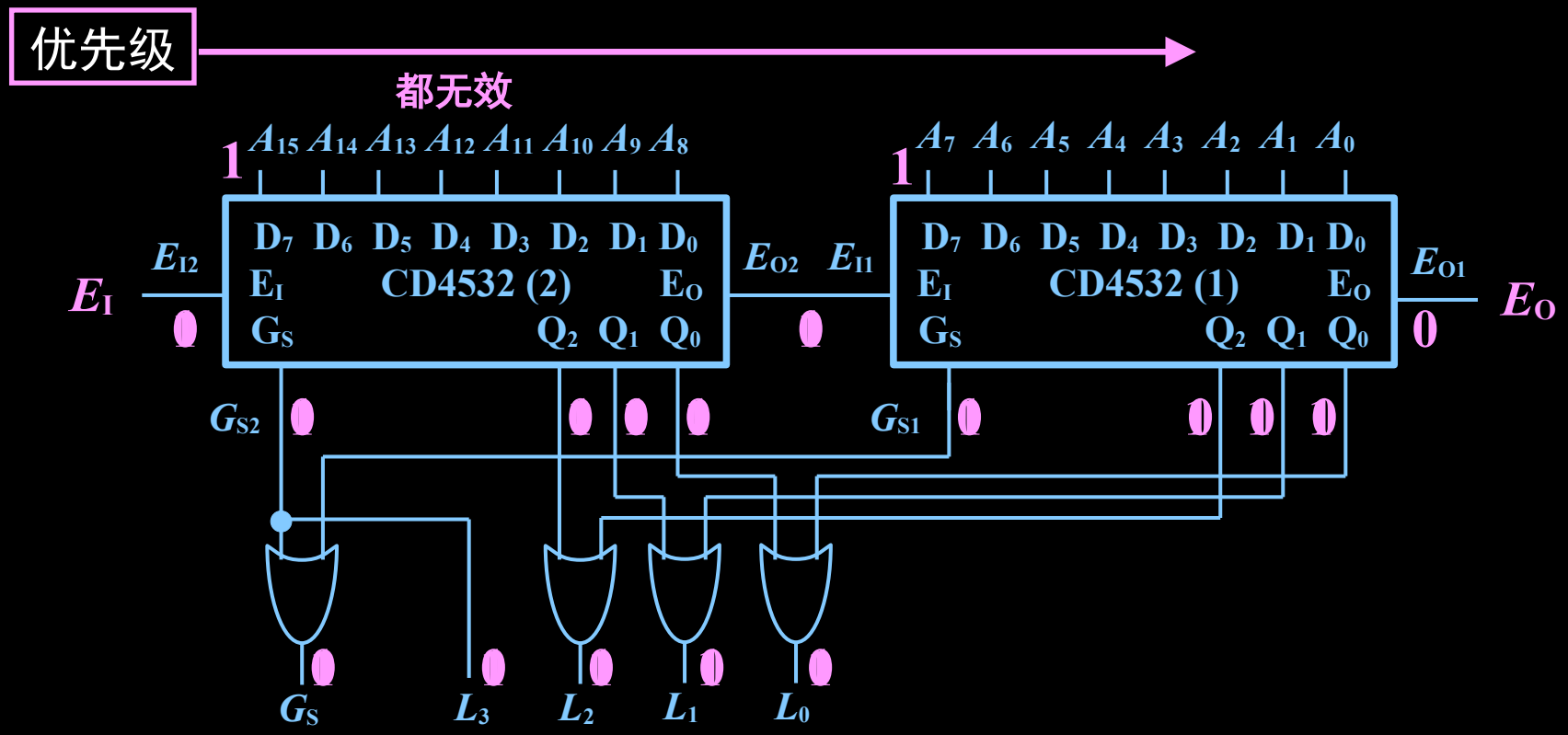
输入									输出				
E_I	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0	G_S	E_O
L	x	x	x	x	x	x	x	x	L	L	L	L	L
H	L	L	L	L	L	L	L	L	L	L	L	L	H
H	H	x	x	x	x	x	x	x	H	H	H	H	L
H	L	H	x	x	x	x	x	x	H	H	L	H	L
H	L	L	H	x	x	x	x	x	H	L	H	H	L
H	L	L	L	H	x	x	x	x	H	L	L	H	L
H	L	L	L	L	H	x	x	x	L	H	H	H	L
H	L	L	L	L	L	H	x	x	L	H	L	H	L
H	L	L	L	L	L	L	H	x	L	L	H	H	L
H	L	L	L	L	L	L	L	H	L	L	L	H	L

E_I：输入使能；　E_O：输出使能；　G_S：编码器工作状态标志

优先级传递：$E_I = 1$，$D_7 \ldots D_0$ 全无效，　$E_O = 1$，允许下级编码，否则不允许

例：二片 CD4532 构成 16-4 线优先编码器

解： $G_S = G_{S2} + G_{S1}$, $L_3 = G_{S2}$, $L_i = Q_{i2} + Q_{i1}$ ($i = 0, 1, 2$)



1. 译码器的定义与功能

译码：编码的逆过程，将二进制码翻译成代表某一特定含义的有效信号

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：将输入代码转换成与之唯一对应的有效信号

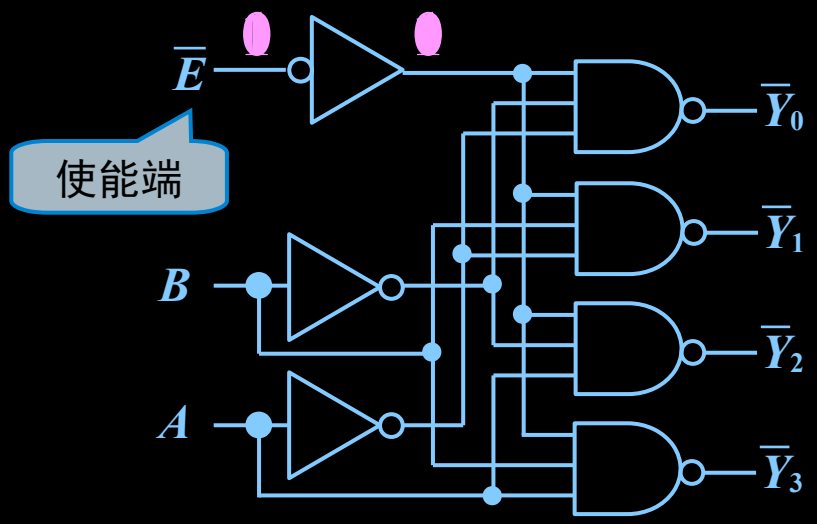
代码变换器：将一种代码转换为另一种代码

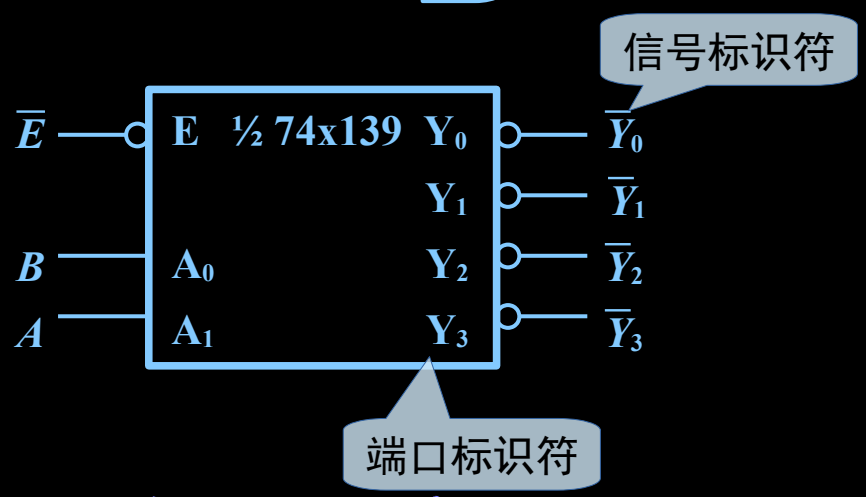
常见译码器
- 地址译码器
- BCD 译码器
- 显示译码器

2. 地址译码器

(1) 74HC139

2-4 线地址译码器





$\overline{E}=1 \quad \overline{Y}_3=\overline{Y}_2=\overline{Y}_1=\overline{Y}_0=1$ 无效

$\overline{E}=0 \quad \overline{Y}_0=\overline{\overline{A}\cdot\overline{B}} \quad \overline{Y}_1=\overline{\overline{A}\cdot B}$

$\overline{Y}_2=\overline{A\cdot\overline{B}} \quad \overline{Y}_3=\overline{A\cdot B}$

$\overline{Y}_0=\overline{m}_0 \quad \overline{Y}_1=\overline{m}_1 \quad \overline{Y}_2=\overline{m}_2 \quad \overline{Y}_3=\overline{m}_3$

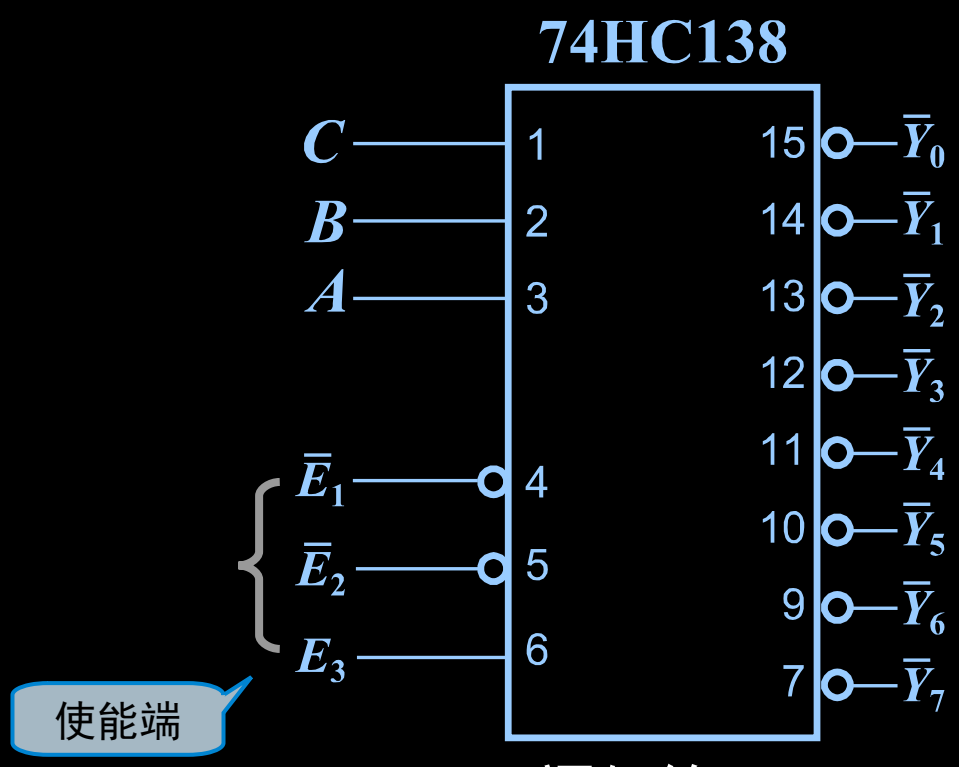
2-4 线二进制译码器功能表

输入			输出			
$\overline{E}$	A	B	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$
1	x	x	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

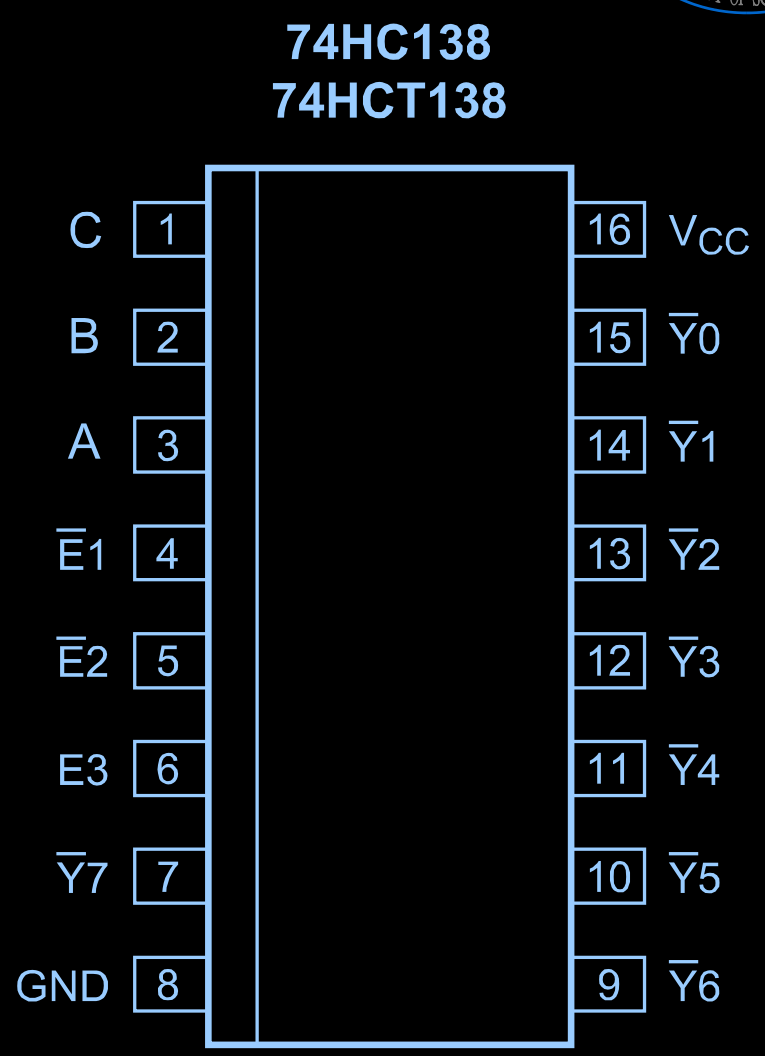
2. 地址译码器

(2) 74HC138

3-8 线地址译码器



逻辑符号



引脚排列图

$E_3 \bullet \overline{\overline{E}}_2 \bullet \overline{\overline{E}}_1 = 1$ 使能

2. 地址译码器

(2) 74HC138

$E = E_3\,\overline{\overline{E}}_2\,\overline{\overline{E}}_1 = 0$

$\overline{Y}_i = 1\ (i = 0 \sim 7)$，无效

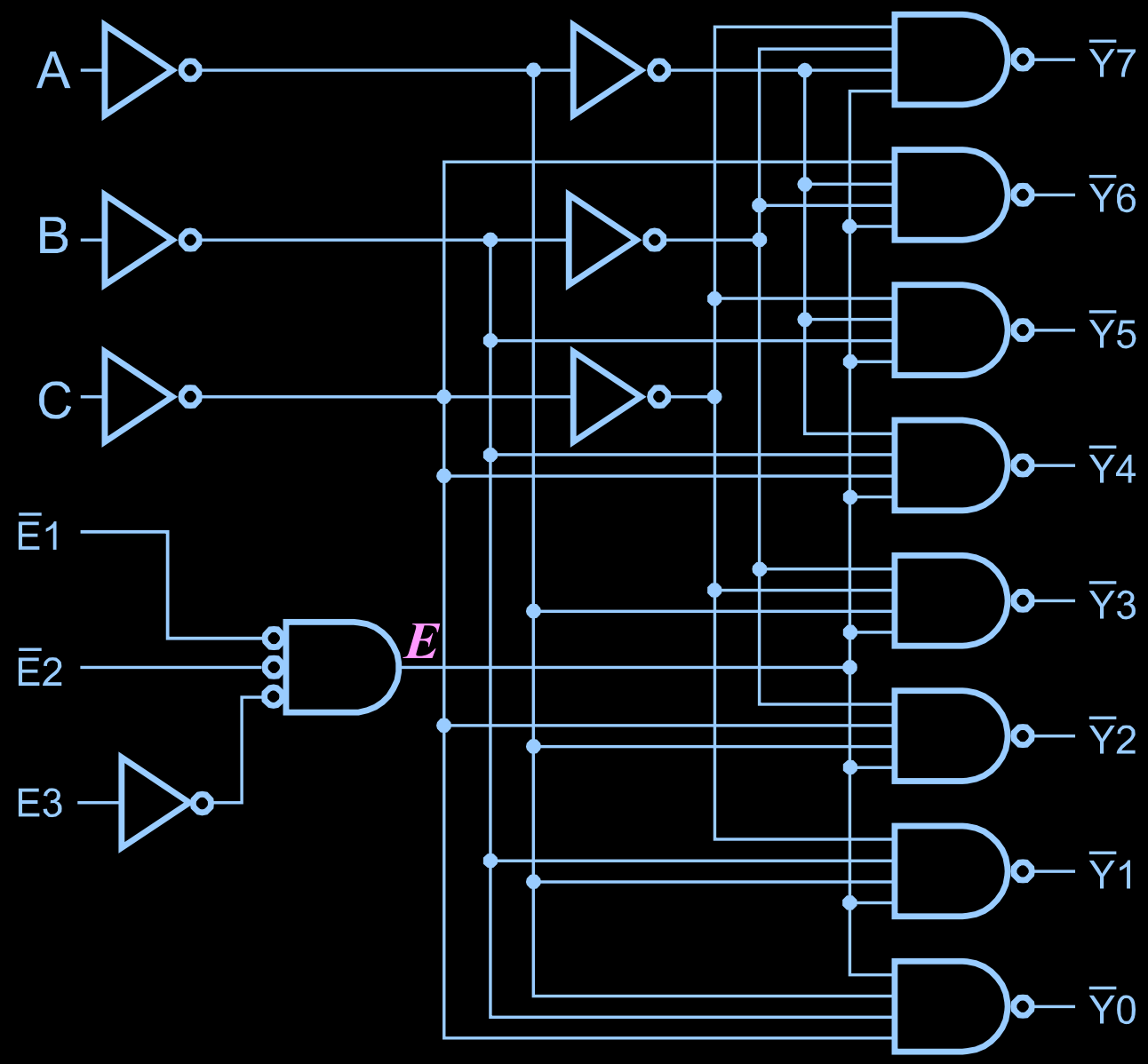
$E = E_3\,\overline{\overline{E}}_2\,\overline{\overline{E}}_1 = 1$

$\overline{Y}_0 = \overline{\overline{A}\cdot\overline{B}\cdot\overline{C}} = \overline{m}_0$

$\overline{Y}_1 = \overline{\overline{A}\cdot\overline{B}\cdot C} = \overline{m}_1$

$\cdots$

$\overline{Y}_7 = \overline{A\cdot B\cdot C} = \overline{m}_7$



2. 地址译码器

(2) 74HC138

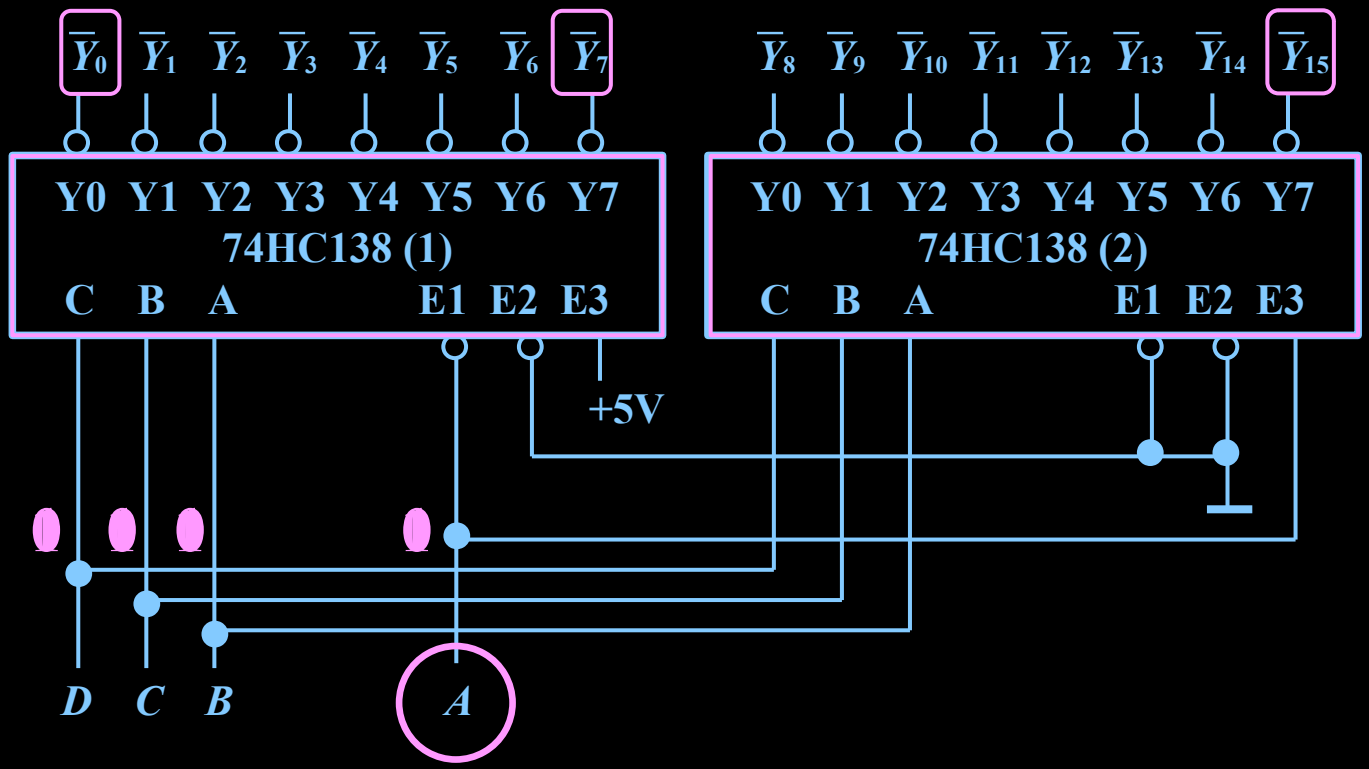
74HC138 集成译码器功能表

E_3	$\overline{E}_2$	$\overline{E}_1$	A	B	C	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
x	H	x	x	x	x	H	H	H	H	H	H	H	H
x	x	H	x	x	x	H	H	H	H	H	H	H	H
L	x	x	x	x	x	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

2. 地址译码器

(2) 74HC138

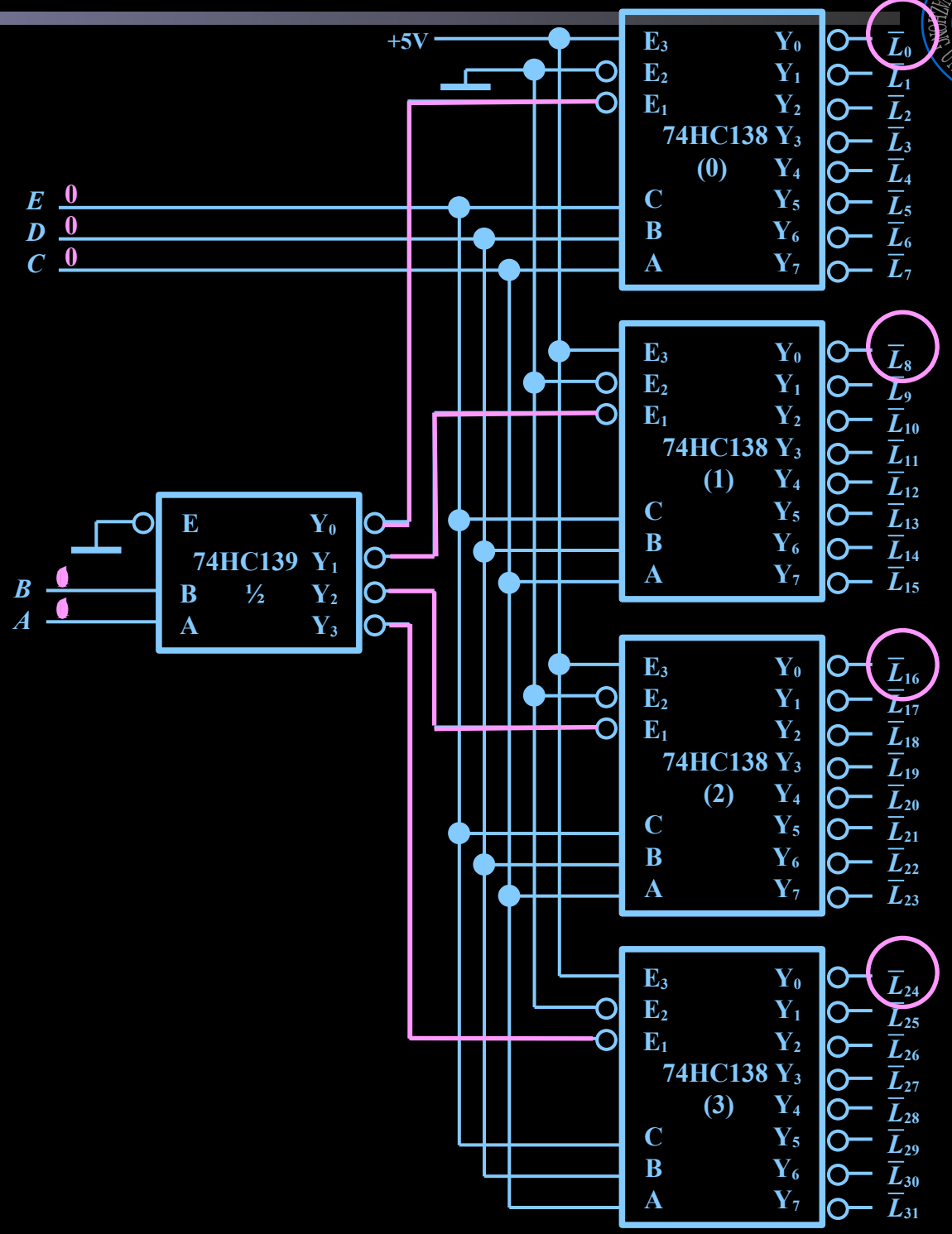
例：用两片 3-8 译码器扩展为 4-16 译码器



2. 地址译码器

(2) 74HC138

例：用 74x139 和 74x138 实现 5-32 线译码器



2. 地址译码器

(2) 74HC138

例：用 74x138 实现逻辑函数 $L=\overline{A}\bullet\overline{C}+A\bullet B$

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

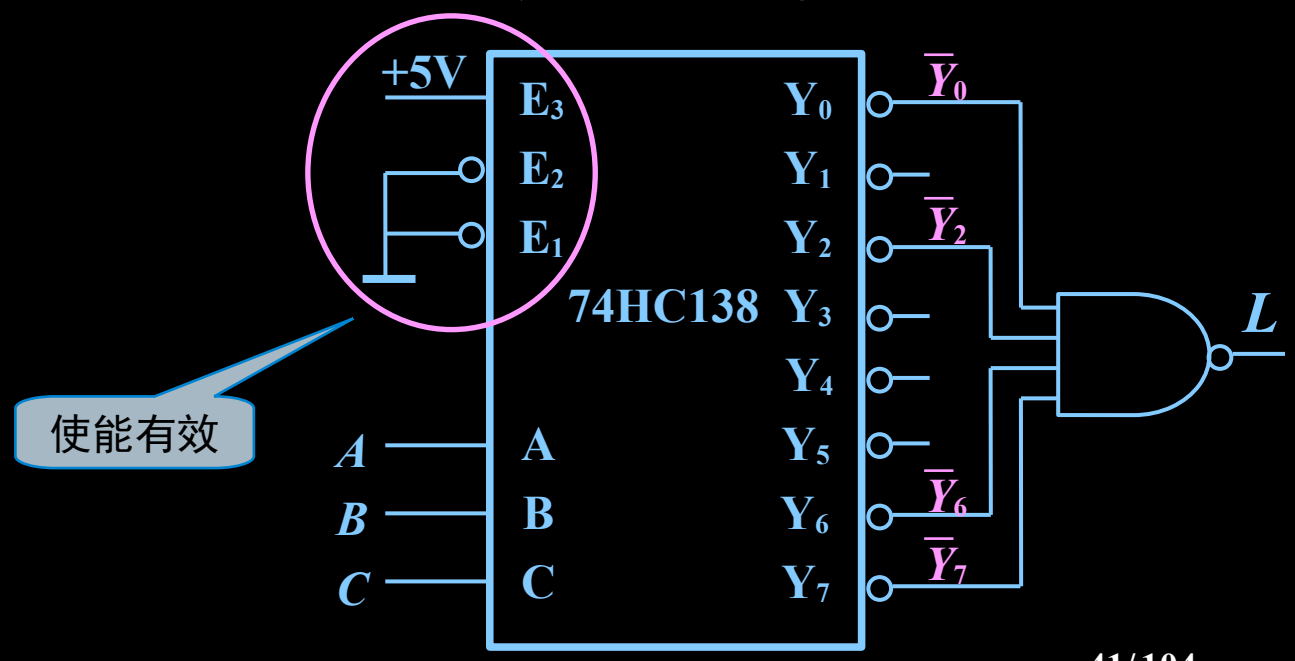
$$=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,B\,\overline{C}+A\,B\,\overline{C}+A\,B\,C$$

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

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

$$=\overline{\overline{Y}_0\cdot\overline{Y}_2\cdot\overline{Y}_6\cdot\overline{Y}_7}$$

设计电路如图：



2. 地址译码器

(2) 74HC138

例：用 74138 和与非门实现 1 位全减器

$$\begin{array}{rcccc} & \bullet & \bullet & \bullet & \\ & 1 & 0 & 1 & 0 \\ - & 0 & 1 & 1 & 1 \\ \hline & 0 & 0 & 1 & 1 \end{array}$$

解：设输入借位 G_I，被减数 A，减数 B；输出本位差 D，借位 G_O

列真值表

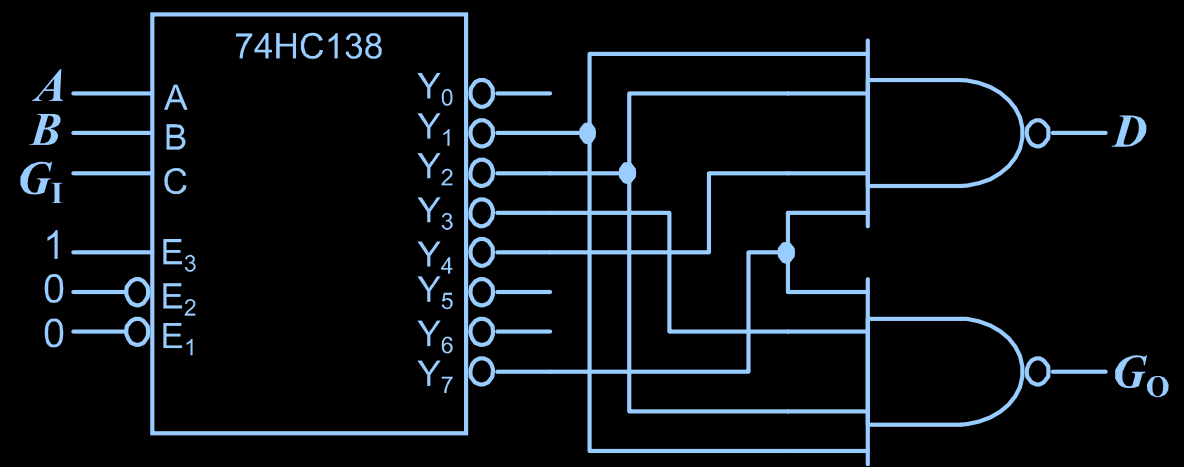
A	B	G_I	D	G_O
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

从真值表得到表达式：

$$D = m_1 + m_2 + m_4 + m_7 = \overline{\overline{m_1} \cdot \overline{m_2} \cdot \overline{m_4} \cdot \overline{m_7}}$$

$$G_O = m_1 + m_2 + m_3 + m_7 = \overline{\overline{m_1} \cdot \overline{m_2} \cdot \overline{m_3} \cdot \overline{m_7}}$$

设计电路如图：

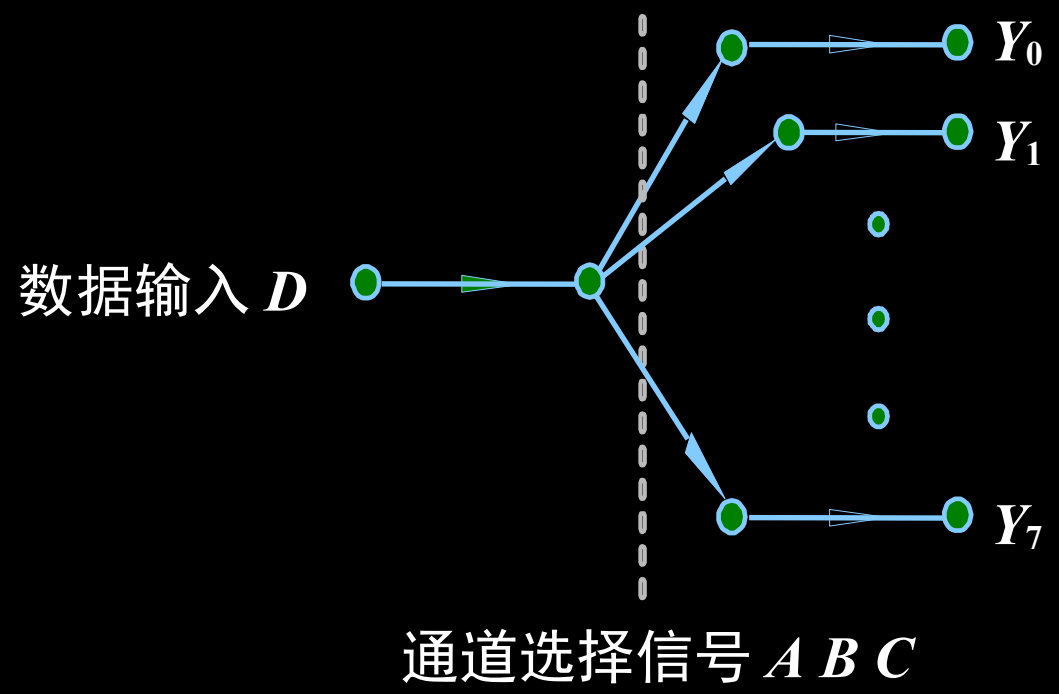


2. 地址译码器

(2) 74HC138

例：用 74HC138 实现 1-8 线数据分配器

解： 1-8 线数据分配器是能将 1 路数据分时送至 8 路通道上的逻辑电路，又称为多路解复用器（ DMUX ），功能类似于多输出的单刀多掷开关。



$A\ B\ C = 000$ ， $Y_0 = D$

$A\ B\ C = 001$ ， $Y_1 = D$

…

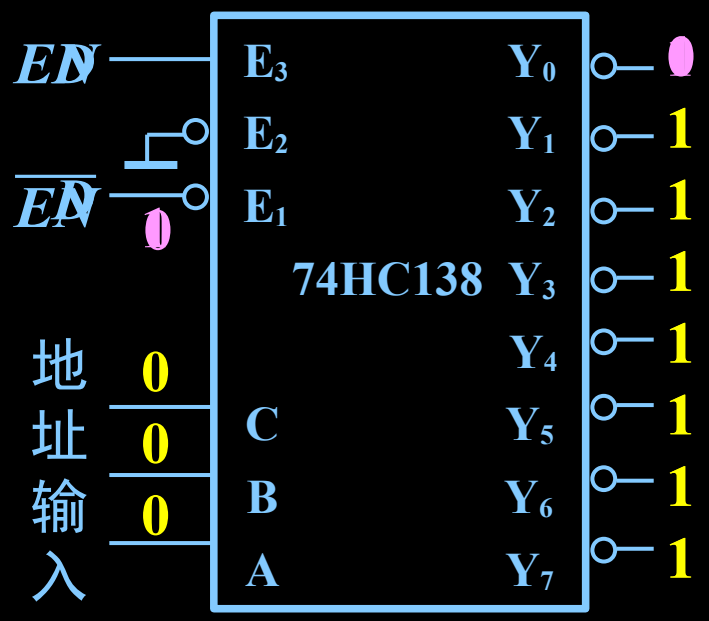
$A\ B\ C = 111$ ， $Y_7 = D$

2. 地址译码器

(2) 74HC138

例：用 74HC138 实现 1-8 线数据分配器

数据分配器



$EN = 1$，使能

74HC138 译码器组成数据分配器功能表

$\overline{E}_1$	A	B	C	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
D	0	0	0	D	1	1	1	1	1	1	1
D	0	0	1	1	D	1	1	1	1	1	1
D	0	1	0	1	1	D	1	1	1	1	1
D	0	1	1	1	1	1	D	1	1	1	1
D	1	0	0	1	1	1	1	D	1	1	1
D	1	0	1	1	1	1	1	1	D	1	1
D	1	1	0	1	1	1	1	1	1	D	1
D	1	1	1	1	1	1	1	1	1	1	D

思考：若 D 从 E_3 输入，则对应输出信号如何变化？

例：试用门电路设计具有低电平使能控制的 1–4 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

解：

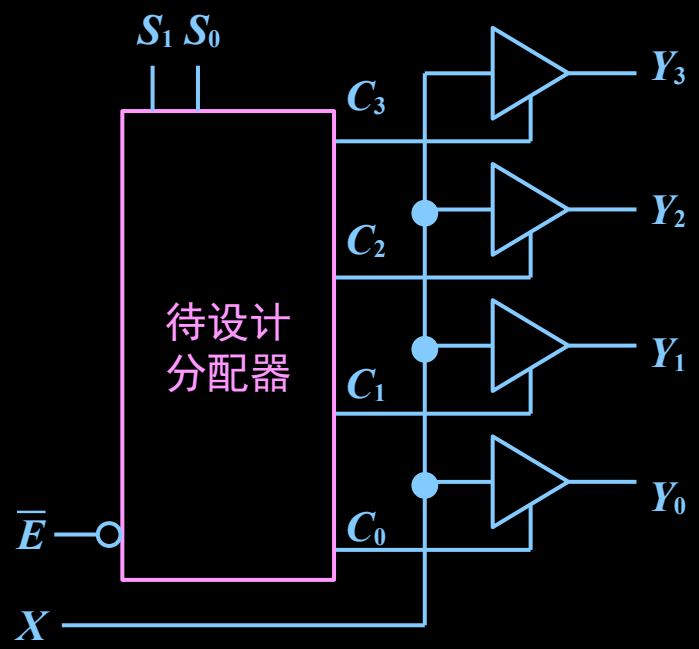
(1) 逻辑抽象

输入： ?

$\overline{E}, S_1, S_0$

输出： ?

C_3, C_2, C_1, C_0



(2) 列真值表

输入			输出			
$\overline{E}$	S_1	S_0	C_3	C_2	C_1	C_0
0	0	0	0	0	0	1
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0
1	x	x	0	0	0	0

(3) 写出 4 个三态门控制端的逻辑函数表达式

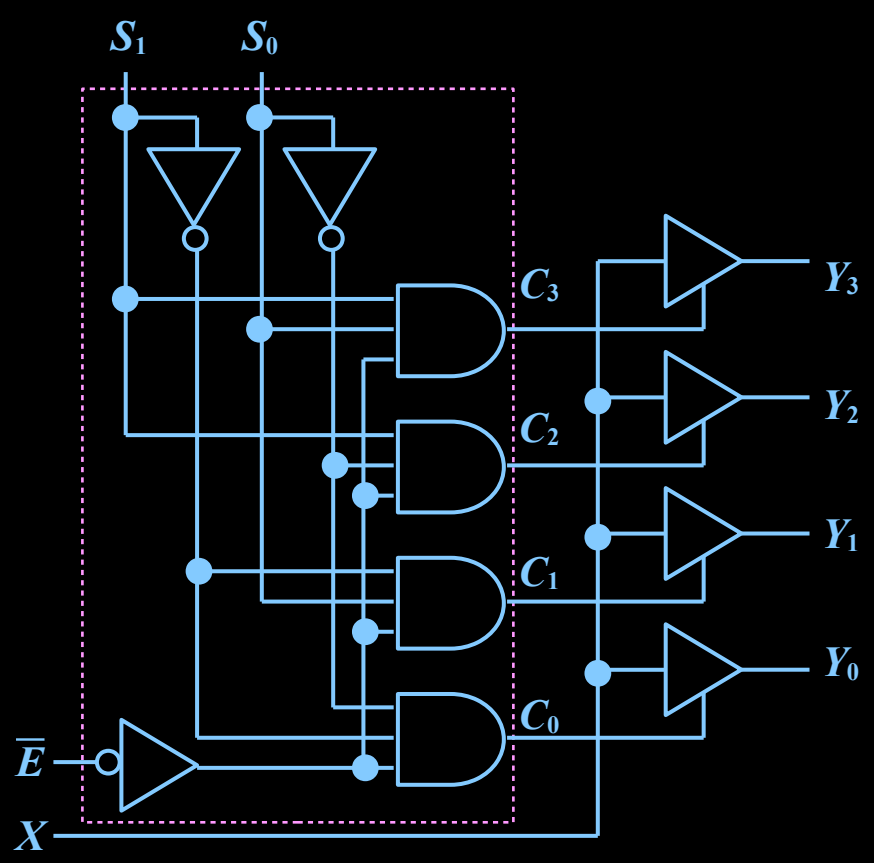
$$C_3 = \overline{\overline{E}}\, S_1 S_0$$

$$C_2 = \overline{\overline{E}}\, S_1 \overline{S}_0$$

$$C_1 = \overline{\overline{E}}\, \overline{S}_1 S_0$$

$$C_0 = \overline{\overline{E}}\, \overline{S}_1 \overline{S}_0$$

例：试用门电路设计具有低电平使能控制的 1–4 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



(3) 写出 4 个三态门控制端的逻辑函数表达式

$$C_3 = \overline{\overline{E}}\, S_1 S_0$$

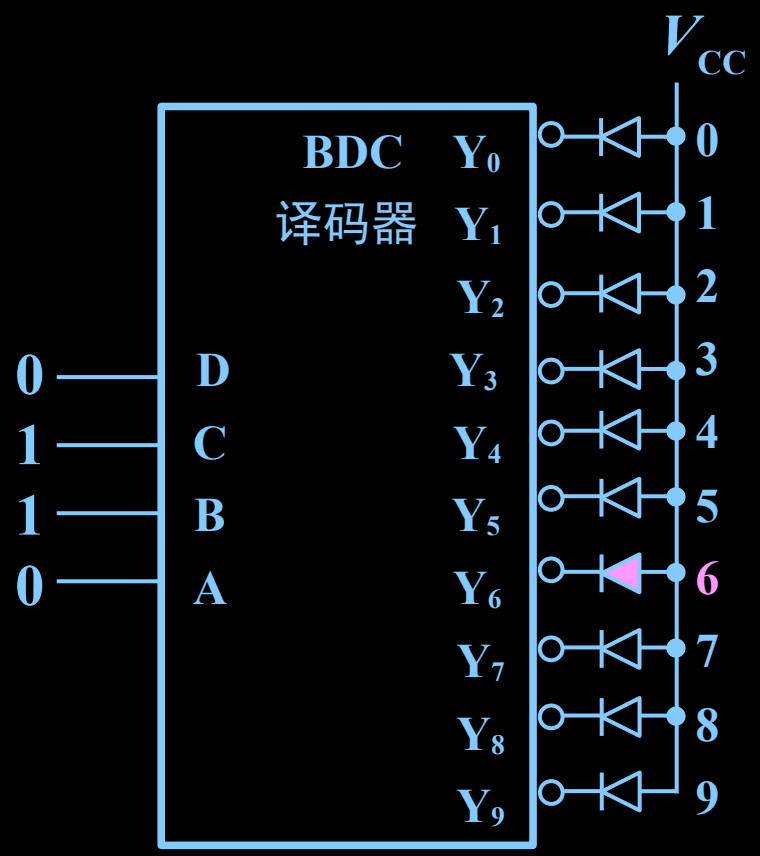
$$C_2 = \overline{\overline{E}}\, S_1 \overline{S}_0$$

$$C_1 = \overline{\overline{E}}\, \overline{S}_1 S_0$$

$$C_0 = \overline{\overline{E}}\, \overline{S}_1 \overline{S}_0$$

3.BCD 译码器：二～十进制译码器，将 8421BCD 码译成 10 个状态

典型芯片： 74HC42



74HC42 译码器功能表

A	B	C	D	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$	$\overline{Y}_8$	$\overline{Y}_9$
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	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	0	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	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

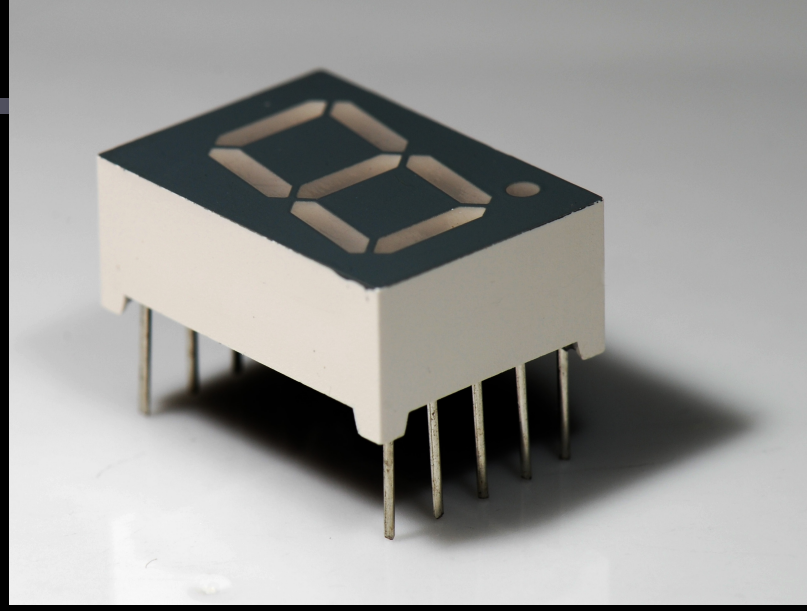
BCD 码之外的 6 个伪码 (1010~1111)， Y_0 ~Y_9 均为高电平

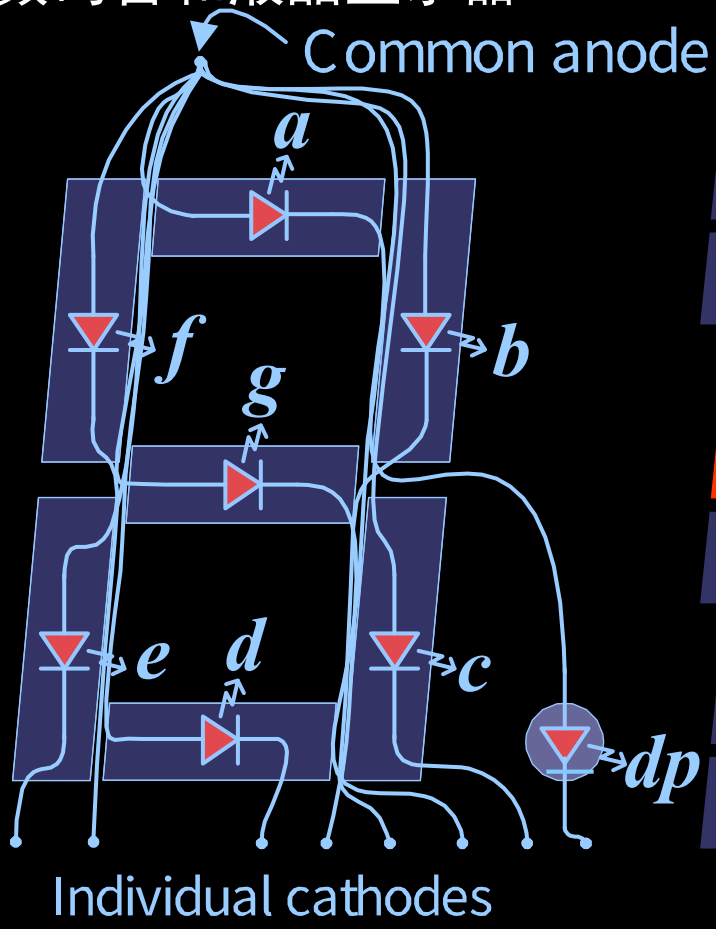
4. 显示译码器

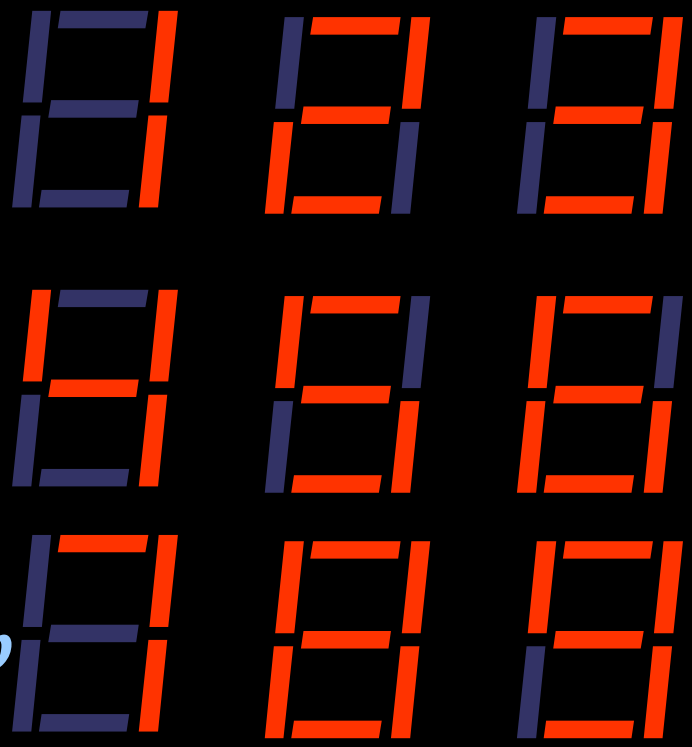
数码显示译码器： 8421BCD 转段码

数码显示器：用来显示数字或符号

如： LED 数码管和液晶显示器



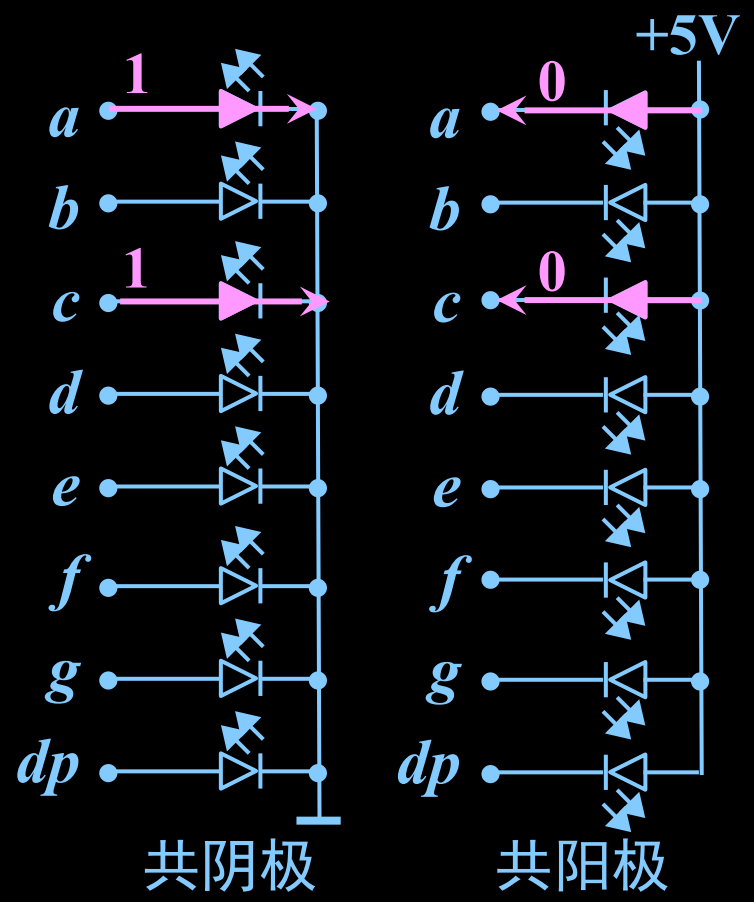


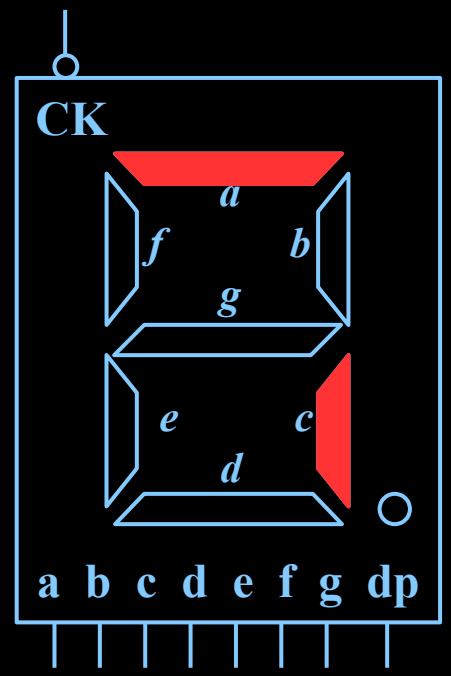


LED (Light Emitting Diode)： 发光二极管的缩写

常用的 LED 数码管有七段、八段等，每段对应一个发光二极管

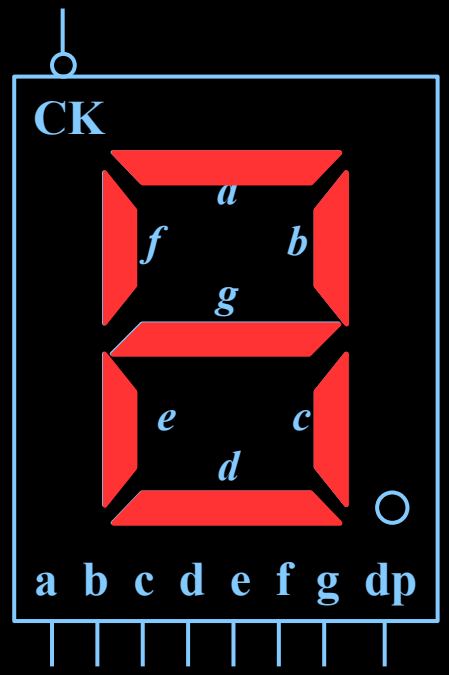
LED 数码管有共阳极和共阴极两种

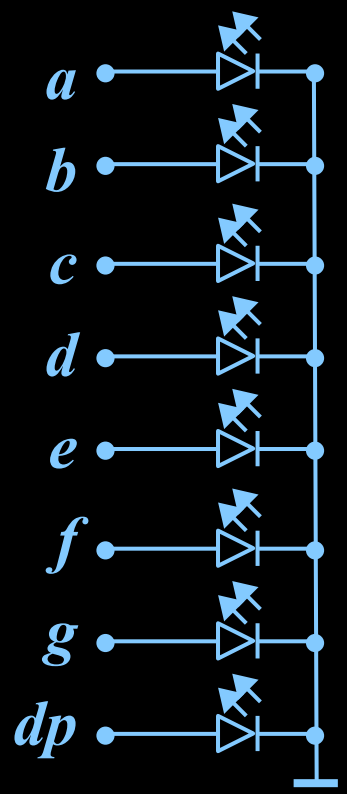




为使 LED 数码管显示不同数字，须输入对应段码 (或称字型码)

如：共阴极 LED 显示器段码，当引脚为高电平时， LED 发光



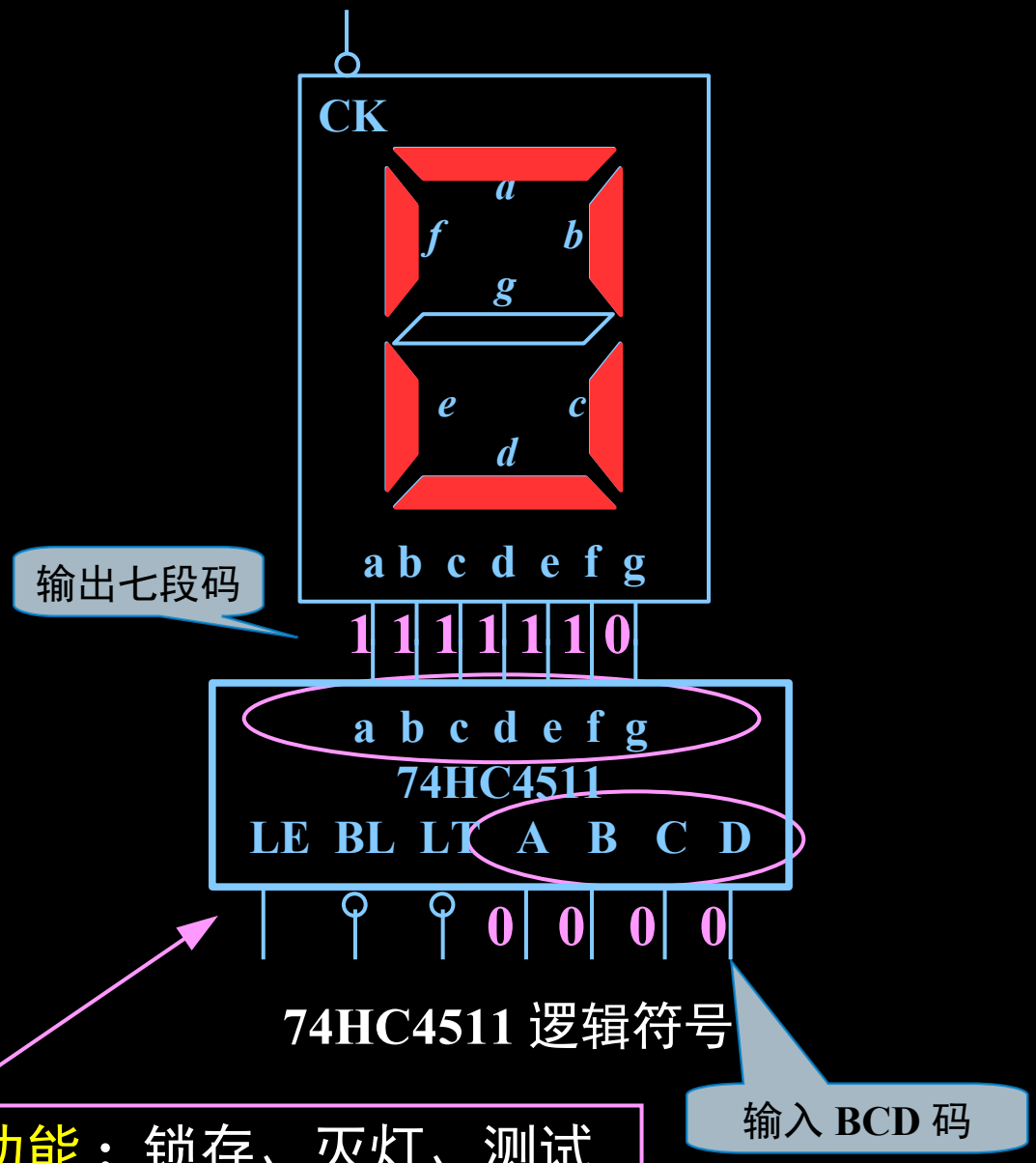


显示	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*db*	段码
0	1	1	1	1	1	1	0	0	FC_H
1	0	1	1	0	0	0	0	0	60_H
2	1	1	0	1	1	0	1	0	DA_H

74HC4511： CMOS 七段显示译码器

74HC4511 译码器基本功能表

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1



74HC4511 逻辑符号

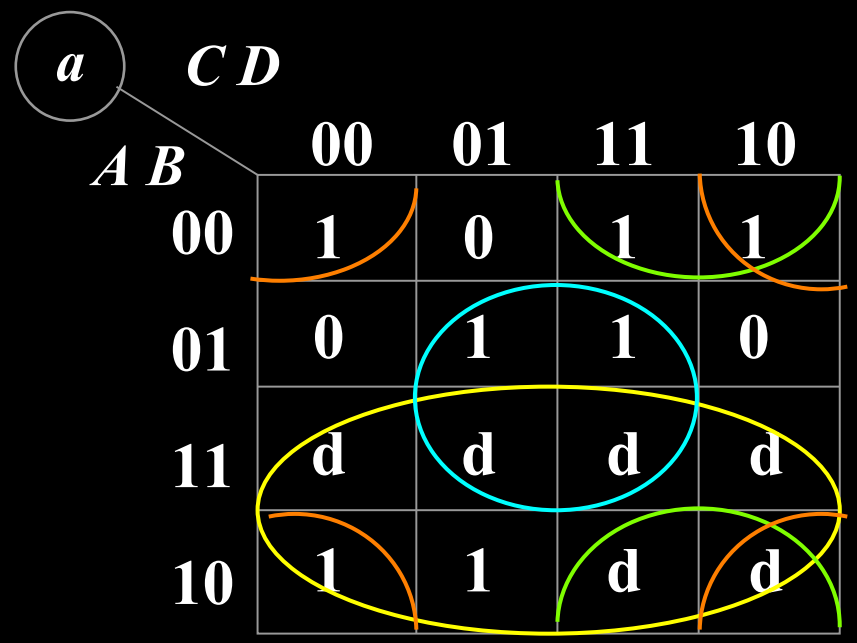
辅助功能：锁存、灭灯、测试

74HC4511： CMOS 七段显示译码器

基本功能电路设计

74HC4511 译码器基本功能表

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1



无关条件： $AB + AC = 0$

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

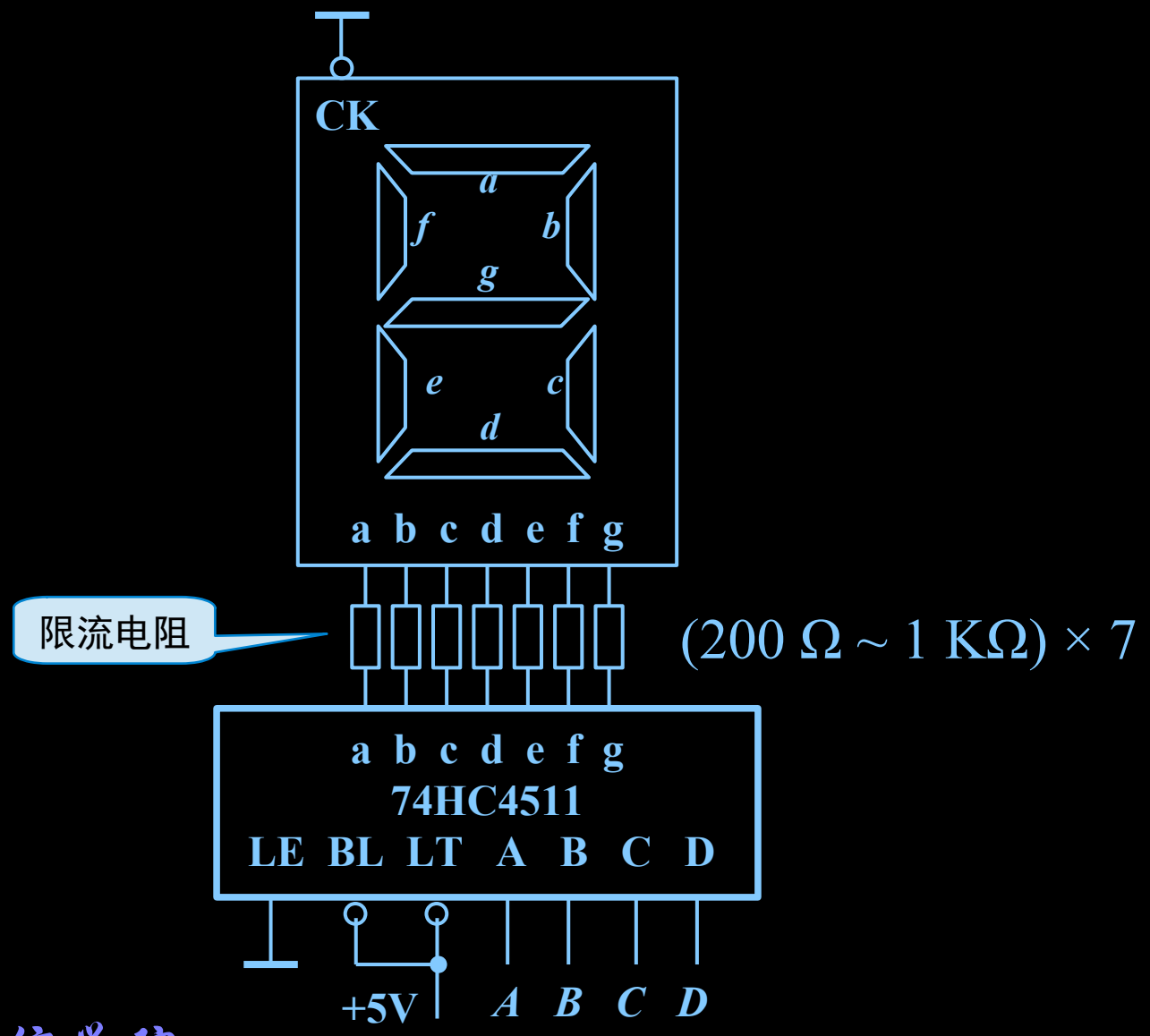
CMOS 七段显示译码器 74HC4511 功能表

功能	输入							输出							字形
	$\overline{LT}$	$\overline{BL}$	LE	A	B	C	D	a	b	c	d	e	f	g	
0	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0
1	1	1	0	0	0	0	1	0	1	1	0	0	0	0	1
2	1	1	0	0	0	1	0	1	1	0	1	1	0	1	2
3	1	1	0	0	0	1	1	1	1	1	1	0	0	1	3
4	1	1	0	0	1	0	0	0	1	1	0	0	1	1	4
5	1	1	0	0	1	0	1	1	0	1	1	0	1	1	5
6	1	1	0	0	1	1	0	0	0	1	1	1	1	1	6
7	1	1	0	0	1	1	1	1	1	1	0	0	0	0	7
8	1	1	0	1	0	0	0	1	1	1	1	1	1	1	8
9	1	1	0	1	0	0	1	1	1	1	0	0	1	1	9

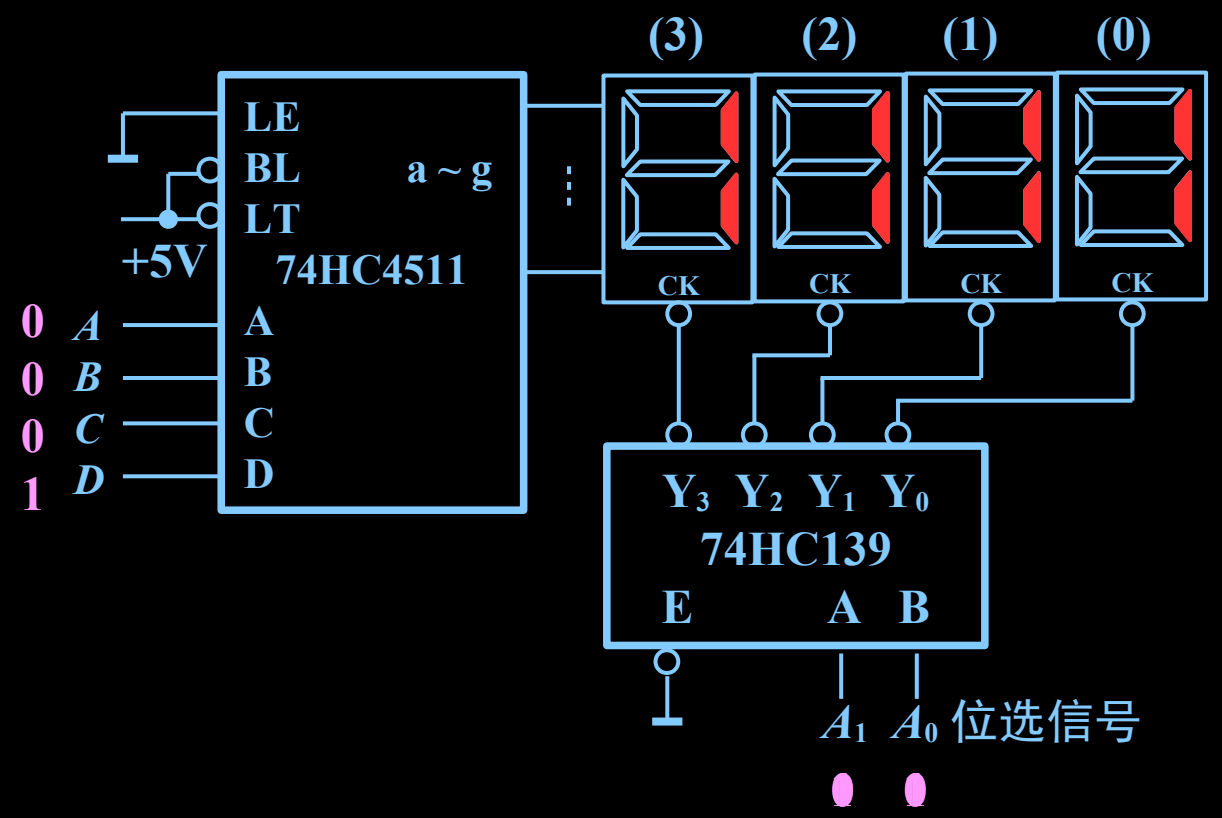
CMOS 七段显示译码器 74HC4511 功能表 (续)

功能	输入							输出							字形
	$\overline{LT}$	$\overline{BL}$	LE	A	B	C	D	a	b	c	d	e	f	g	
10	1	1	0	1	0	1	0	0	0	0	0	0	0	0	熄
11	1	1	0	1	0	1	1	0	0	0	0	0	0	0	熄
12	1	1	0	1	1	0	0	0	0	0	0	0	0	0	熄
13	1	1	0	1	1	0	1	0	0	0	0	0	0	0	熄
14	1	1	0	1	1	1	0	0	0	0	0	0	0	0	熄
15	1	1	0	1	1	1	1	0	0	0	0	0	0	0	熄
测	0	x	x	x	x	x	x	1	1	1	1	1	1	1	8
灭	1	0	x	x	x	x	x	0	0	0	0	0	0	0	熄
锁	1	1	1	x	x	x	x	*	*	*	*	*	*	*	*

显示译码器与显示器的实际连接方式

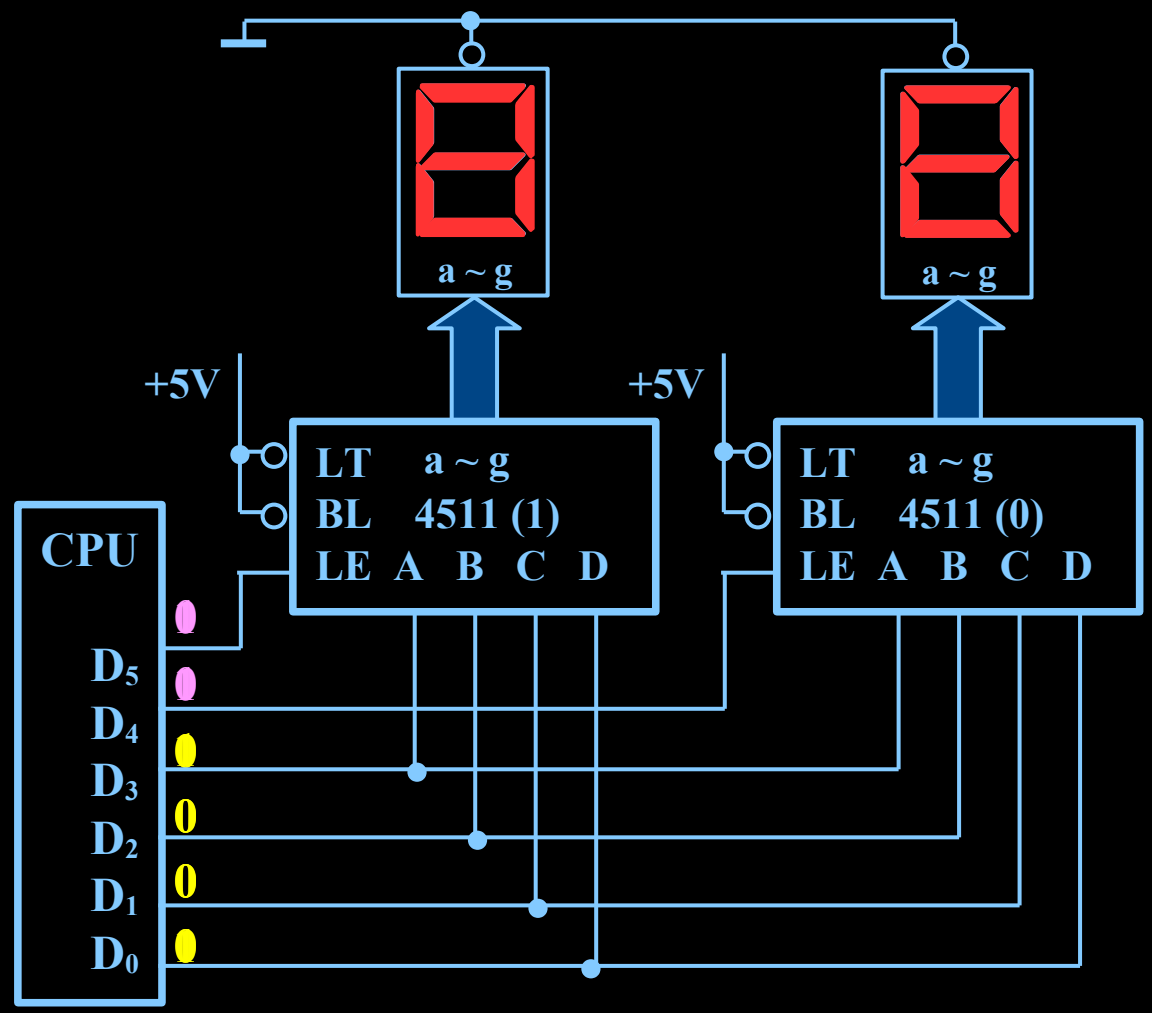


例：由译码器、显示译码及七段显示器构成的 4 位动态显示电路



- 位选择信号 A_1 , A_0 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 *A B C D* 分别在 4 个显示器上显示
- 人眼暂留时间，即满足 25 Hz < f_c < 100 Hz 可看到稳定数字

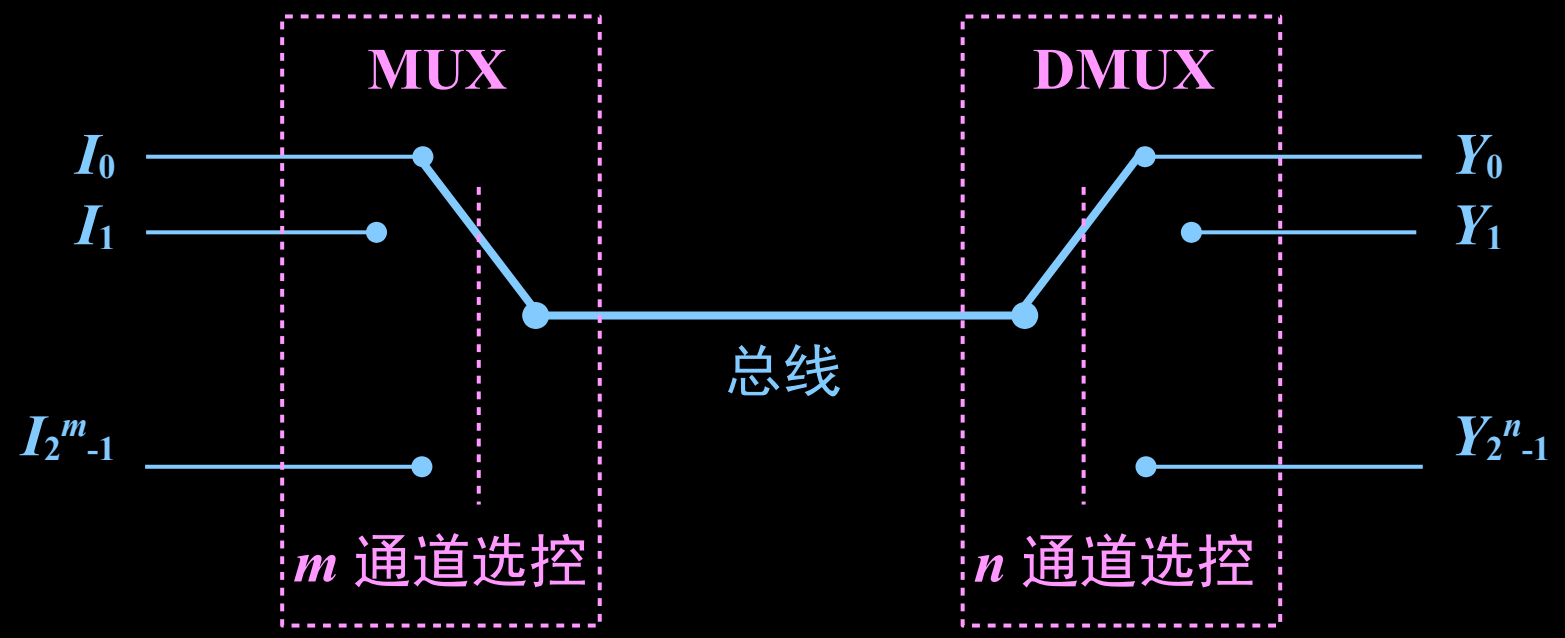
例： 74HC4511 锁存功能示例



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

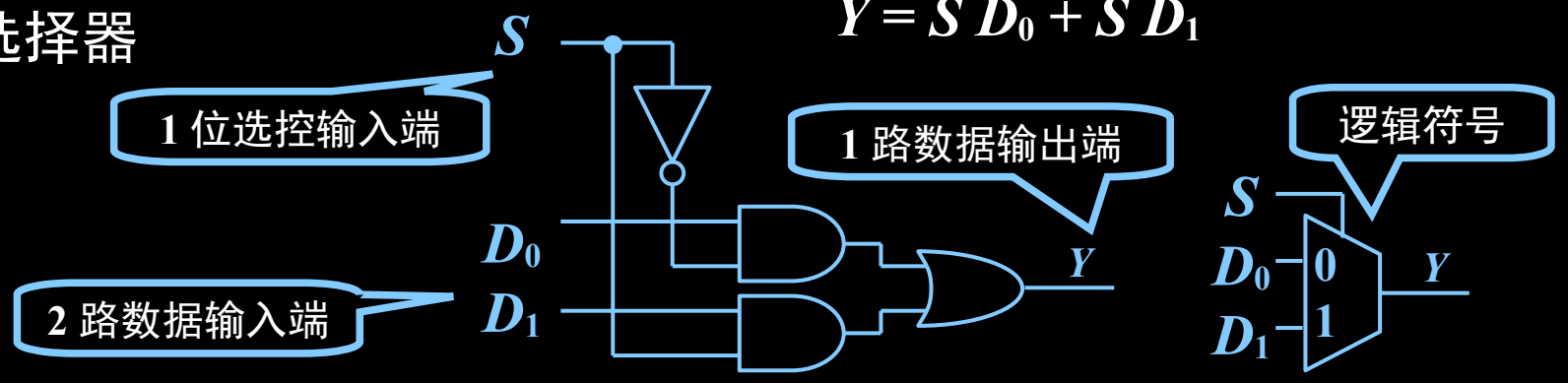
- 功能类似于多输入的单刀多掷开关，又称多路复用器（MUX）
- 在通道选控信号作用下，分时传输多通道数据至公共通道
- 常与多路分配器 / 解复用器（DMUX）一起实现通信总线控制



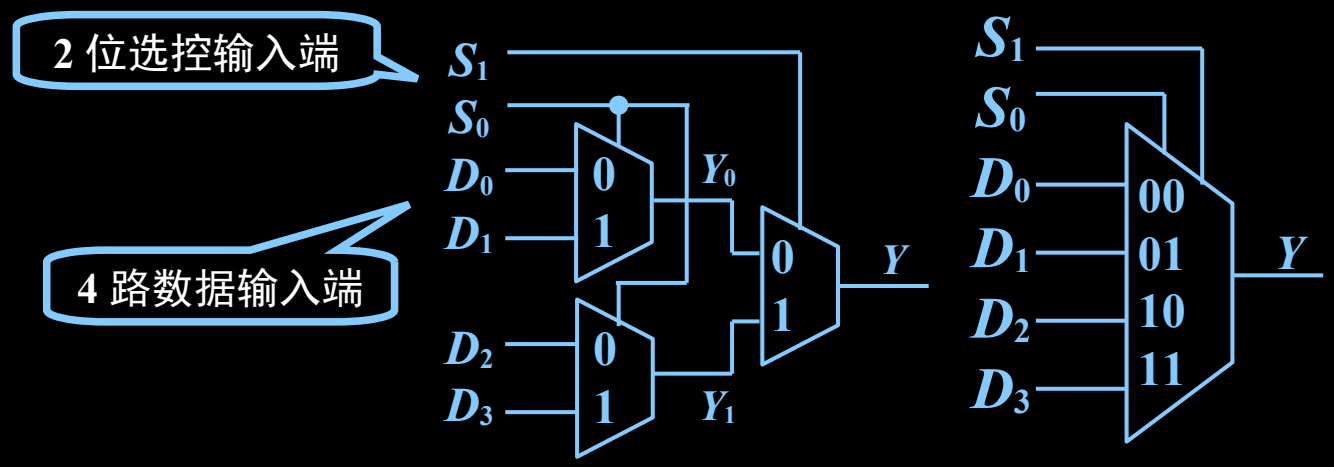
1. 数据选择器的定义与功能

2 选 1 数据选择器

$$Y = \overline{S}\,D_0 + S\,D_1$$



4 选 1 数据选择器：可由 3 个 2 选 1 数据选择器构成



真值表

选控输入		输出
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

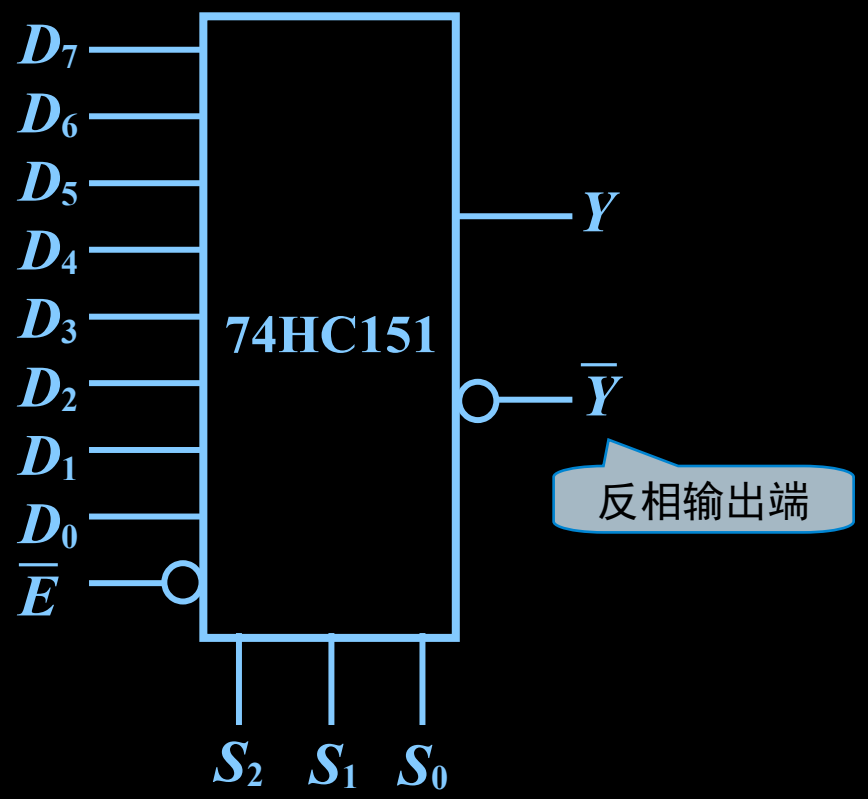
$$Y = \overline{S}_1\,\overline{S}_0\,D_0 + \overline{S}_1\,S_0\,D_1 + S_1\,\overline{S}_0\,D_2 + S_1\,S_0\,D_3$$
$$= m_0\,D_0 + m_1\,D_1 + m_2\,D_2 + m_3\,D_3$$

1. 数据选择器的定义与功能

74HC151： 8 选 1 集成数据选择器，使能有效后输出表达式为：

$$Y=\bar{S}_2\bar{S}_1\bar{S}_0D_0+\bar{S}_2\bar{S}_1S_0D_1+\bar{S}_2S_1\bar{S}_0D_2+\bar{S}_2S_1S_0D_3+S_2\bar{S}_1\bar{S}_0D_4+S_2\bar{S}_1S_0D_5+S_2S_1\bar{S}_0D_6+S_2S_1S_0D_7$$

$$Y=\sum_{i=0}^{7}m_iD_i$$



74HC151 逻辑符号

功能表

输入				输出
$\bar{E}$	S_2	S_1	S_0	Y
1	x	x	x	0
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7

1. 数据选择器的定义与功能

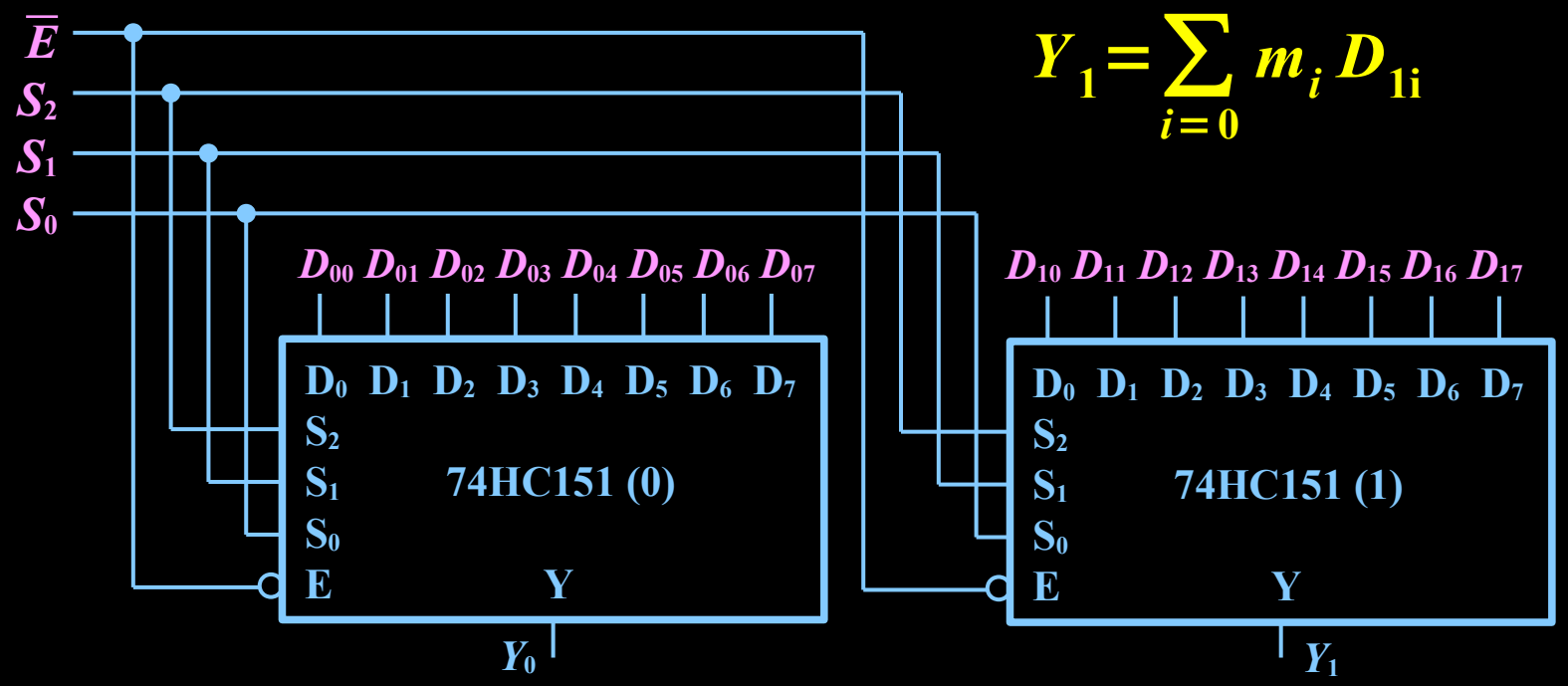
例： 74HC151 位扩展，实现 2 位通道 8 路选 1 路的数据选择器

使能有效后输出表达式为：

$$Y_0=\sum_{i=0}^{7} m_i D_{0i}$$

$$Y_1=\sum_{i=0}^{7} m_i D_{1i}$$

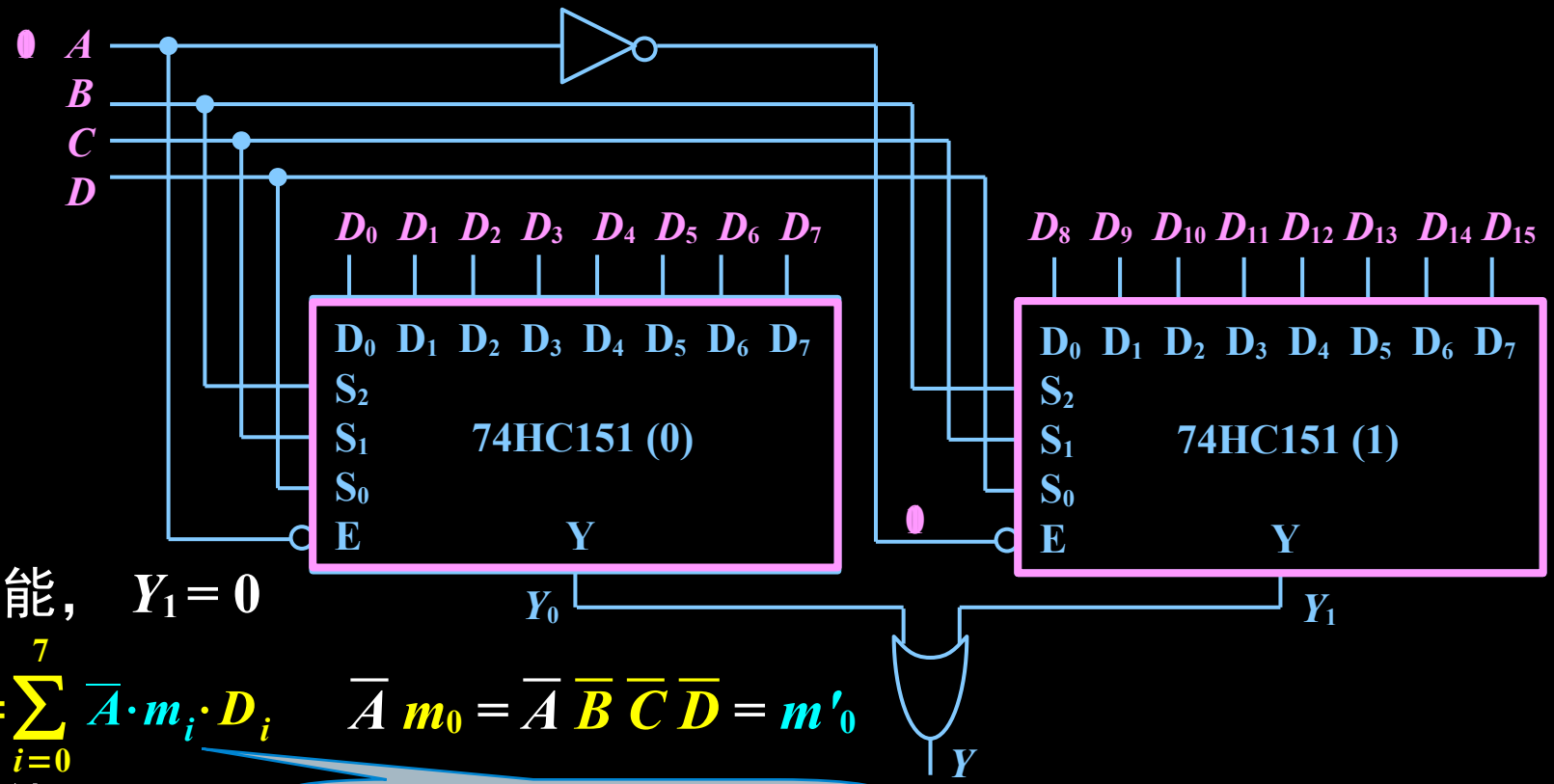
$S_2 S_1 S_0$	Y_1	Y_0
0 0 0	D_{10}	D_{00}
0 0 1	D_{11}	D_{01}
…….		
1 1 0	D_{16}	D_{06}
1 1 1	D_{17}	D_{07}



1. 数据选择器的定义与功能

例： 74HC151 字扩展，实现 16 路选 1 路的数据选择器

解：



$A=0$，片 0 使能， $Y_1=0$

$$Y=Y_0+Y_1=Y_0=\sum_{i=0}^{7}\overline{A}\cdot m_i\cdot D_i \qquad \overline{A}\,m_0=\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}=m'_0$$

m_i 是 B, C, D 三变量的最小项

$A=1$，片 1 使能， $Y_0=0$

$$Y=Y_0+Y_1=Y_1=\sum_{i=0}^{7}A\cdot m_i\cdot D_{i+8} \qquad A\,m_0=A\,\overline{B}\,\overline{C}\,\overline{D}=m'_8$$

m'_i 是 A,B,C,D 四变量的最小项

$$Y=Y_0+Y_1=\sum_{i=0}^{15}m'_i\cdot D_i$$

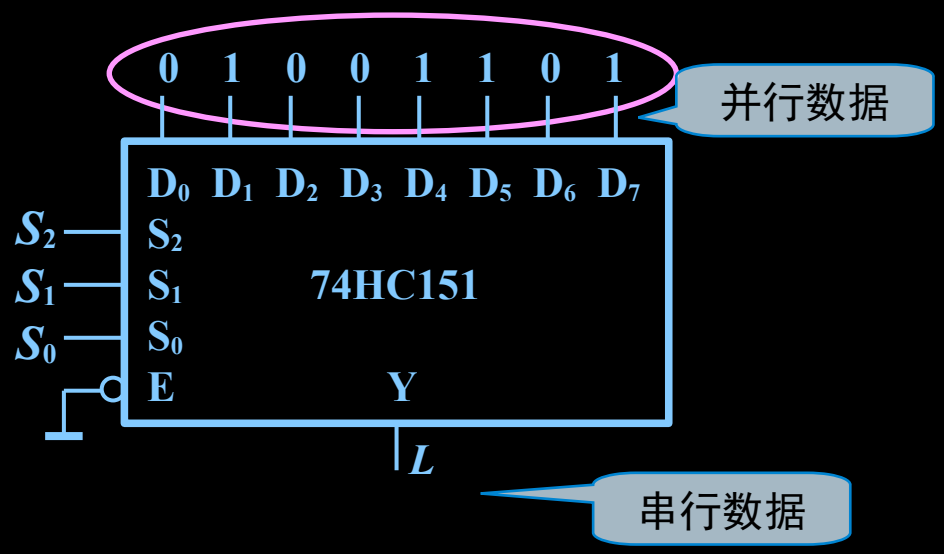
1. 数据选择器的定义与功能

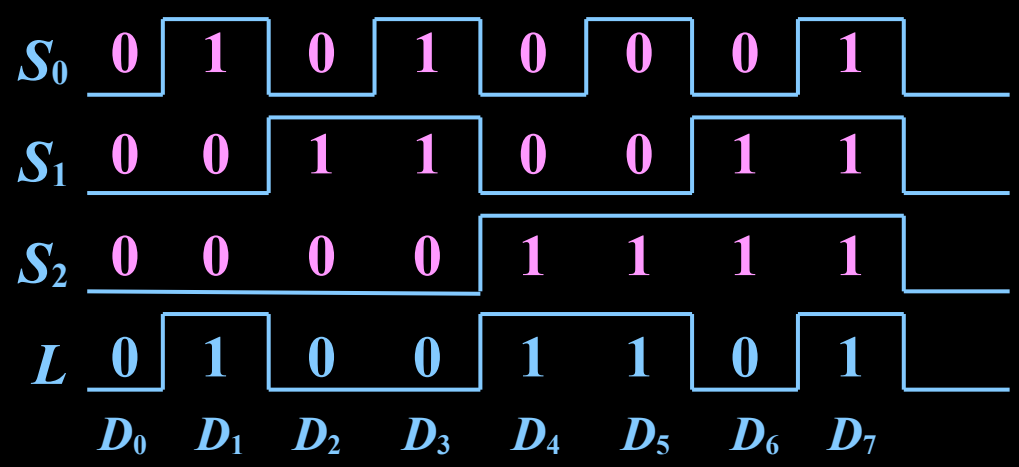
例： 74HC151 实现并串转换

并行数据：多根线上的多位数据

串行数据： 1 根线上分时表示的多位数据

如图，控制 $S_2\ S_1\ S_0$ ，可将 D_i 数据串行输出





2. 数据选择器实现逻辑函数

(1) 选控端数 n 与目标函数变量数相同：

- 将目标函数的所有变量分配给 MUX 各控制端
- 写出目标函数的标准与或式
- 若上式含某最小项 m_i，则选择器对应的待选端 D_i 为 1，否则为 0

(2) 选控端数 n 小于目标函数变量数：

- 选目标函数中的 n 个变量分配给 MUX 各选控端
- 写出目标函数的标准与或式，并提取选控端最小项因子 m_i
- 对比选择器输出函数 $Y=\sum_{i=0}^{2^n-1} m_i D_i$，便可得到 D_i

2. 数据选择器实现逻辑函数

例：用 4 选 1 数据选择器实现 $L = AB + A\overline{C} + BC$

解： 4 选 1 选择器有 2 个选控端，设 $S_1 = A$， $S_0 = B$

选择器输出函数： $Y = \overline{A}\,\overline{B}D_0 + \overline{A}BD_1 + A\overline{B}D_2 + ABD_3$

n 个选控端的选择器

输出函数 $Y = \sum_{i=0}^{2^n-1} m_i D_i$

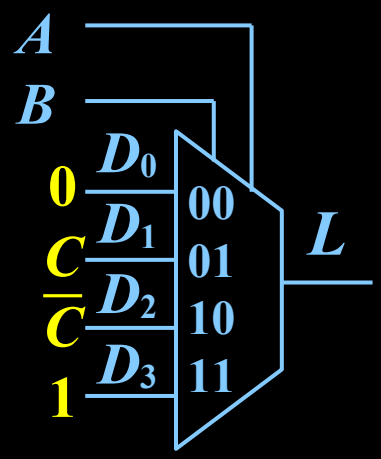
写出 L 的标准与或式：

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

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

提取选控端 A 和 B 的最小项因子得： $L = AB + A\overline{B}\,\overline{C} + \overline{A}BC$

故 $D_0 = 0$, $D_1 = C$, $D_2 = \overline{C}$, $D_3 = 1$ 时， $Y = L$



2. 数据选择器实现逻辑函数

例：用 2 选 1 数据选择器和必要的逻辑门实现 $L(A, B, C) = \Sigma m(3, 4, 6, 7)$

解： 2 选 1 选择器有 1 个选控端，设 $S = A$

n 个选控端的选择器

输出函数 $Y = \sum_{i=0}^{2^n-1} m_i D_i$

选择器输出函数： $Y = \overline{A} D_0 + A D_1$

L 已是标准与或式，即 $L = \overline{A} B C + A \overline{B} \overline{C} + A B \overline{C} + A B C$

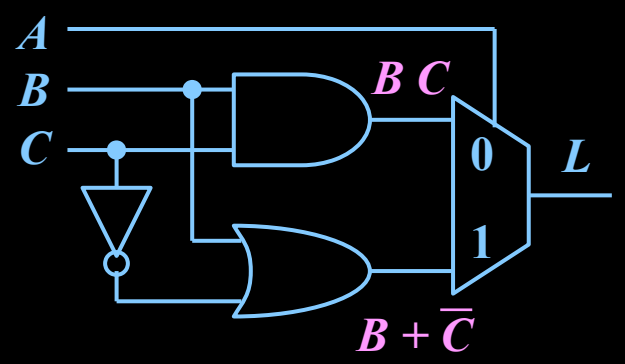
提取选控端 A 的最小项因子：

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

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

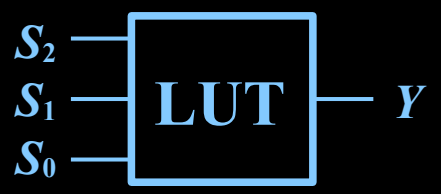
$$= \overline{A} B C + A(B + \overline{C})$$

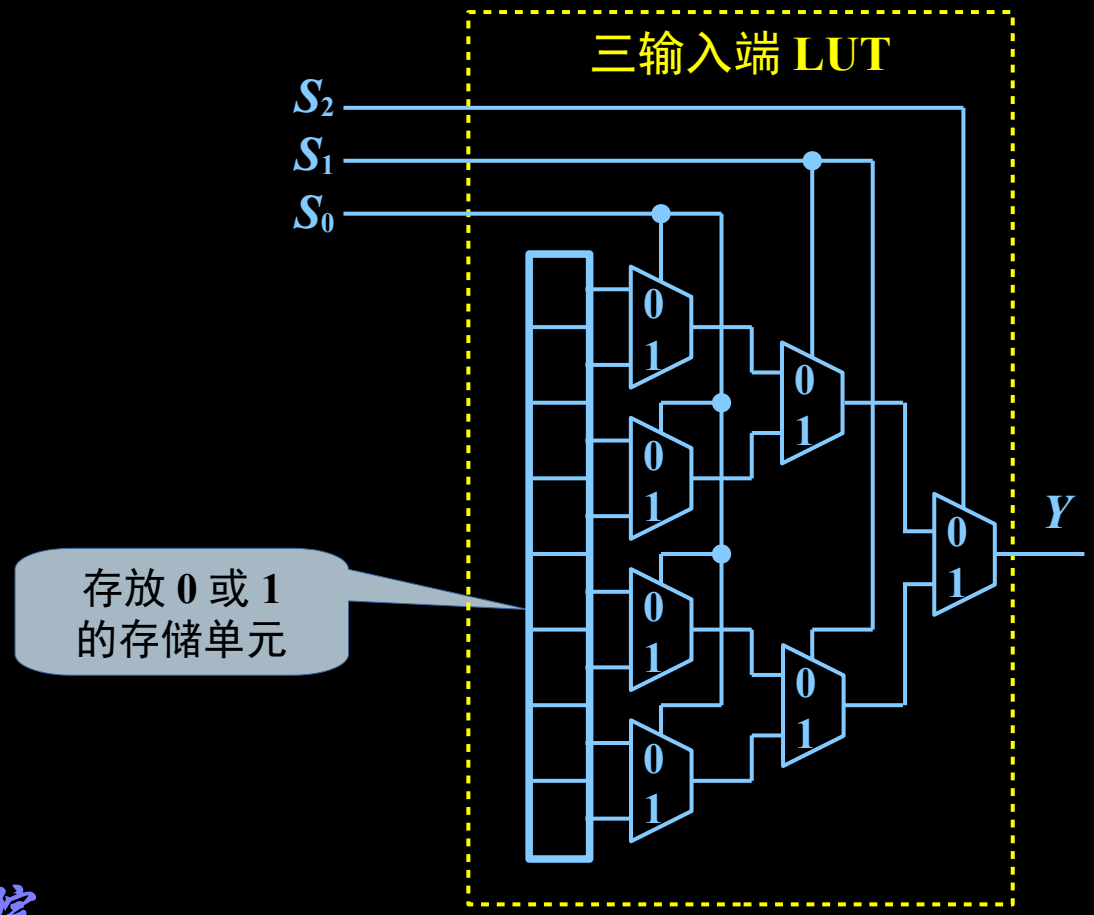
故 $D_0 = B C$, $D_1 = B + \overline{C}$ 时， $Y = L$



3. 数据选择器构成查找表 LUT

- FPGA 基本单元的逻辑块是查找表 LUT
- LUT 实质是存储器加选择器，以真值表的形式实现逻辑函数
- LUT 输入端，即为内部选择器的选控端





3. 数据选择器构成查找表 LUT

例：用 3 输入端 LUT 实现逻辑函数 $L(A, B, C) = \Sigma m(3, 4, 6, 7)$

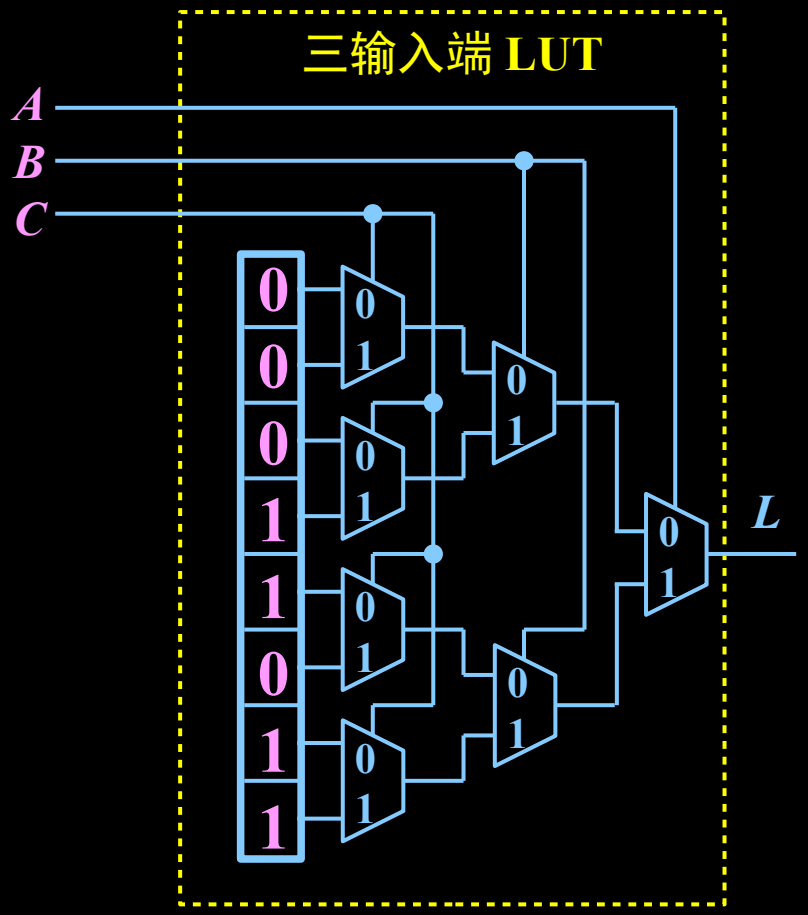
解： 3 输入端 LUT 由 8 选 1 选择器和 8 位存储器组成

设 3 输入端分配为 $S_2 = A$, $S_1 = B$, $S_0 = C$

LUT 输出函数为 $Y = \sum_{i=0}^{7} m_i D_i$

因标准与或式 $L(A, B, C) = \Sigma m(3, 4, 6, 7)$

故 8 位存储器应配置为 1101_1000_B



数值比较器：对两个数字进行比较，以判断其大小的逻辑电路

1. 一位数值比较器

输入：两个一位二进制数 A， B

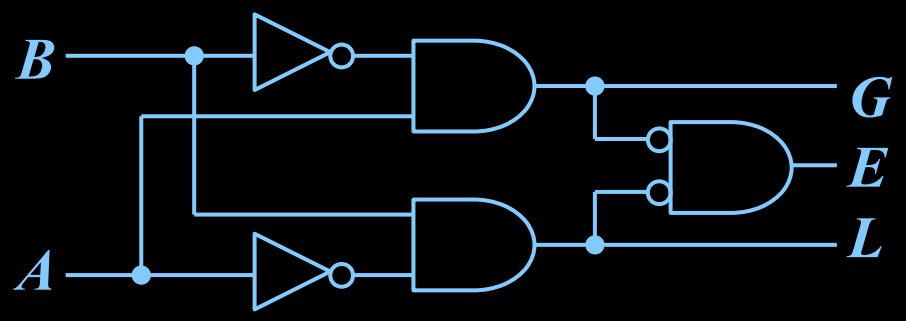
输出： $G=1$，表示 A 大于 B

$L=1$，表示 A 小于 B

$E=1$，表示 A 等于 B

真值表

输入		输出		
A	B	G	L	E
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1



输出表达式

$$G=A\overline{B},\quad L=\overline{A}B$$

$$E=\overline{A}\,\overline{B}+AB$$

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

$$=\overline{G+L}$$

$$=\overline{G}\cdot\overline{L}$$

2. 两位数值比较器

比较两个 2 位二进制数 $A = A_1A_0$， $B = B_1B_0$， 输出 G , L , E

可用一位数值比较器设计两位数值比较器

设高位 A_1 和 B_1 比较结果输出至 G_1 , L_1 , E_1

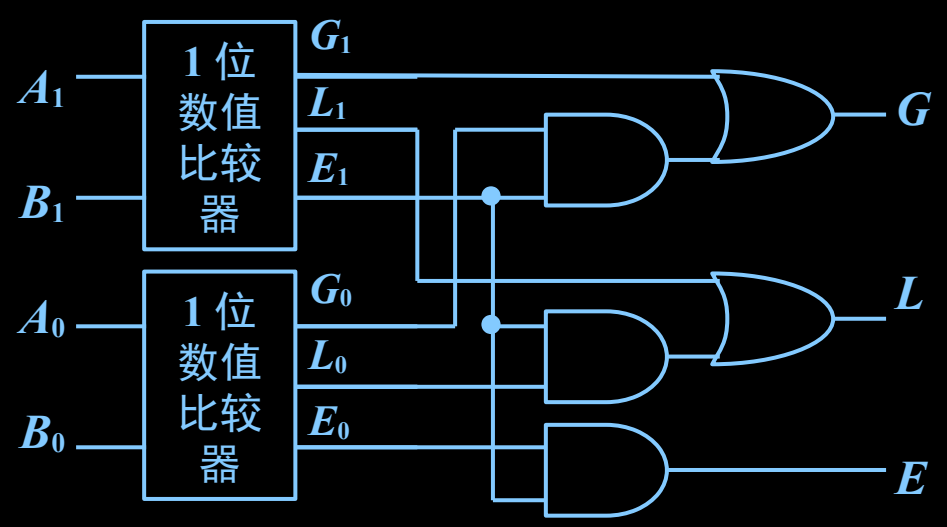
而低位 A_0 和 B_0 比较结果输出至 G_0 , L_0 , E_0

由高、低位比较结果，可得两位数比较器输出为：

$$G = G_1 + E_1 \bullet G_0$$

$$L = L_1 + E_1 \bullet L_0$$

$$E = E_1 \bullet E_0$$



3. 集成数值比较器

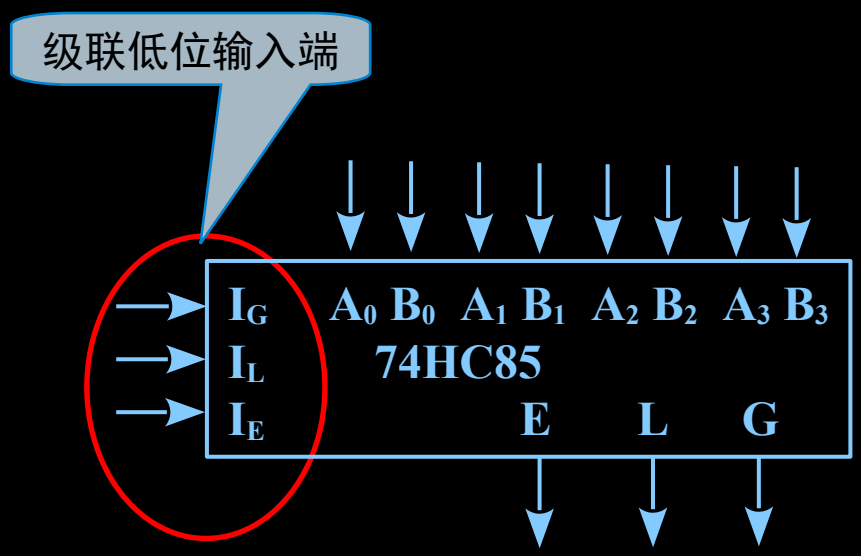
(1) 74HC85 四位数值比较器

其原理和两位数值比较器相同

输入： $A = A_3A_2A_1A_0$， $B = B_3B_2B_1B_0$

当 $A \neq B$，输出取决 A, B 比较

当 $A = B$，输出取决 I_G, I_L, I_E



74HC85 的示意框图

$A = B$ 时输出与输入关系

输入			输出		
I_G	I_L	I_E	G	L	E
1	0	0	1	0	0
0	1	0	0	1	0
x	x	1	0	0	1
1	1	0	0	0	0
0	0	0	1	1	0

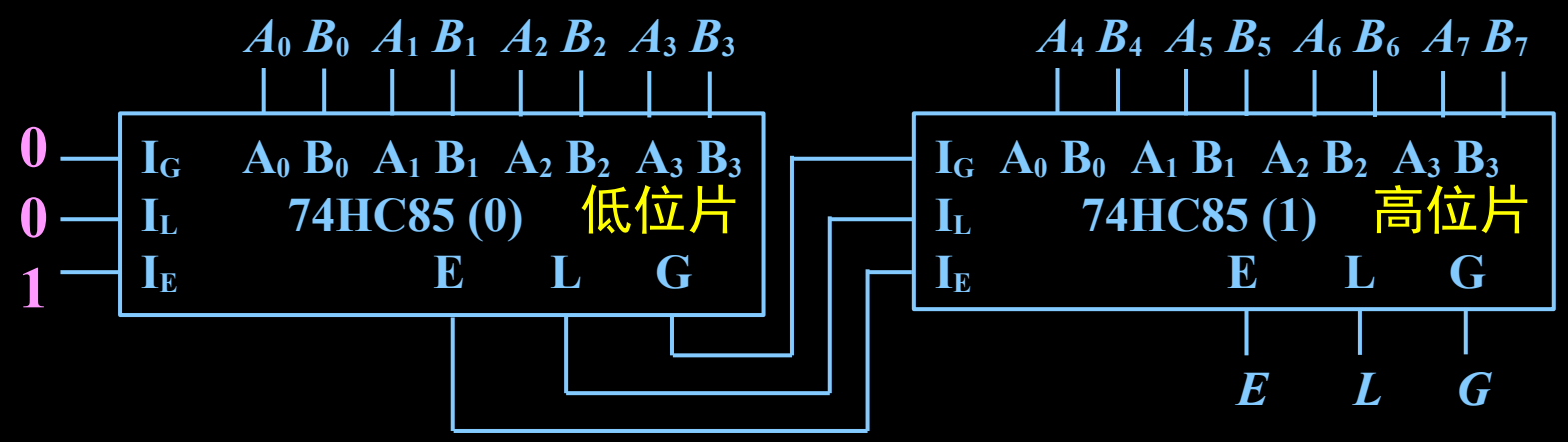
非级联时， $I_G = 0$, $I_L = 0$, $I_E = 1$

3. 集成数值比较器

(2) 集成数值比较器的扩展

例： 2 片 74HC85 串联扩展实现 8 位数值比较器

输入： $A = A_7A_6A_5A_4A_3A_2A_1A_0$ $B = B_7B_6B_5B_4B_3B_2B_1B_0$



若高位片两数相等，结果取决于低位片的输出，故会附加 1 级时延

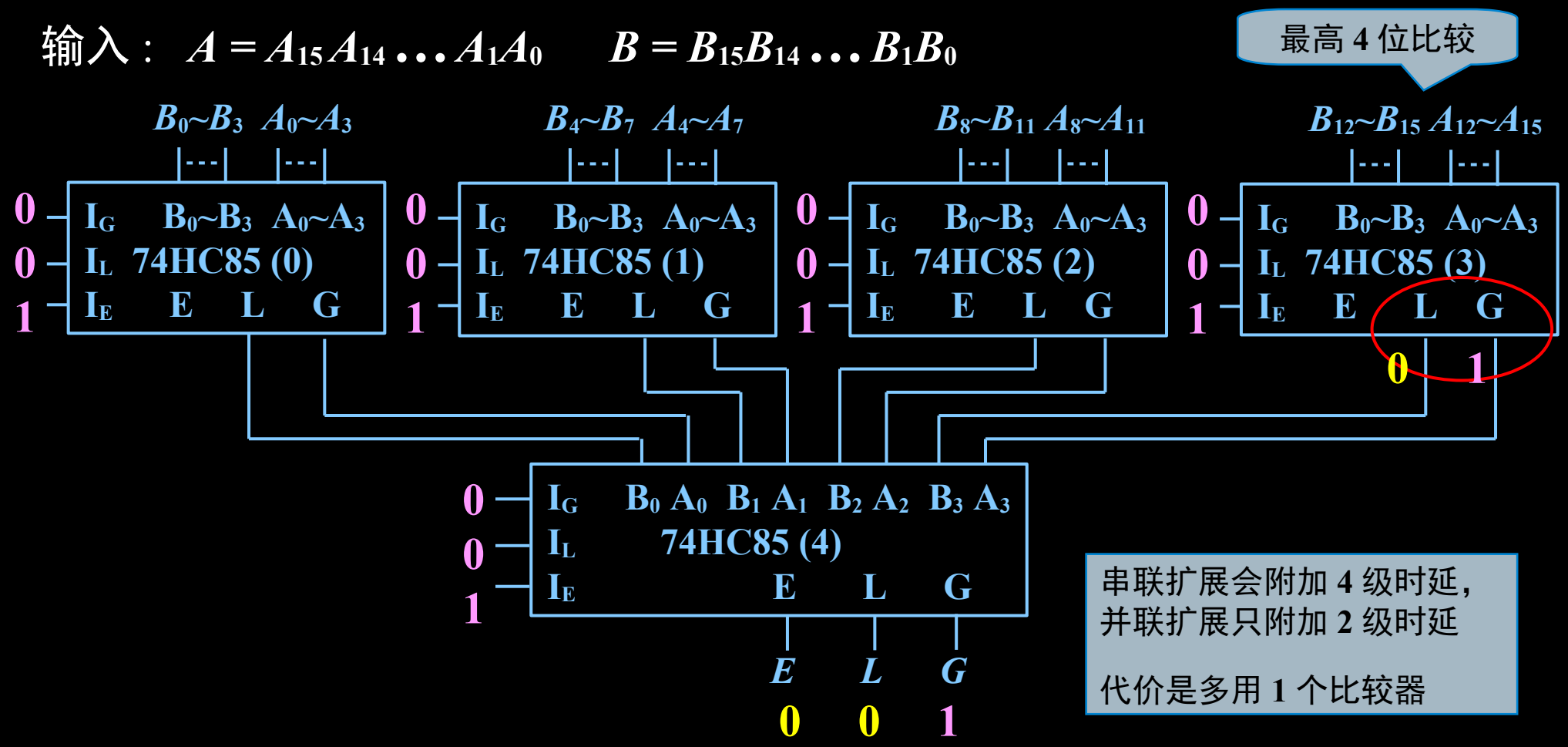
串联越多，附加时延越大，且时延取决于操作数的值，不利于时序控制

3. 集成数值比较器

(2) 集成数值比较器的扩展

例： 5 片 74HC85 并联扩展实现 16 位数值比较器

输入： $A = A_{15}A_{14}\ldots A_1A_0$ $B = B_{15}B_{14}\ldots B_1B_0$



串联扩展会附加 4 级时延，
并联扩展只附加 2 级时延

代价是多用 1 个比较器

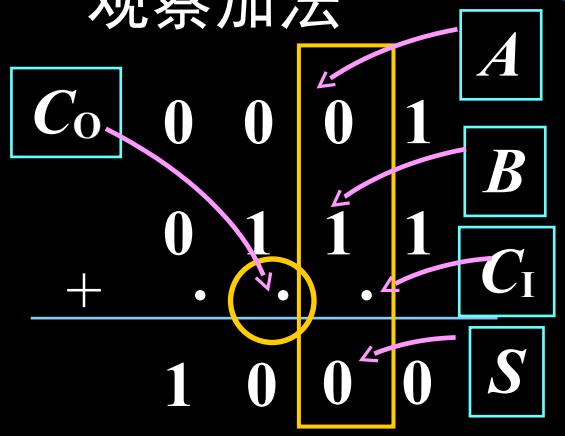
1. 半加器和全加器

加法器分为半加器和全加器两种

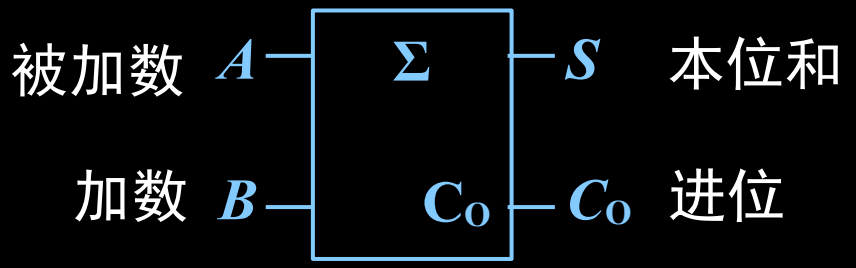
半加：在两个 1 位二进制数相加时，不考虑低位来的进位的相加

全加：在两个 1 位二进制数相加时，考虑低位进位的相加

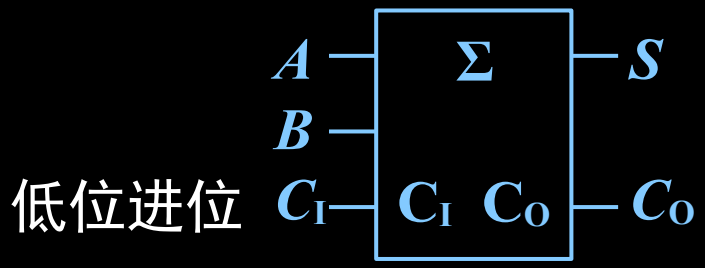
观察加法



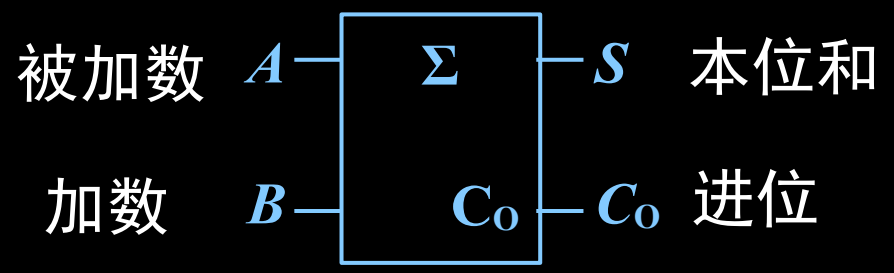
半加器



全加器



(1) 半加器



真值表

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

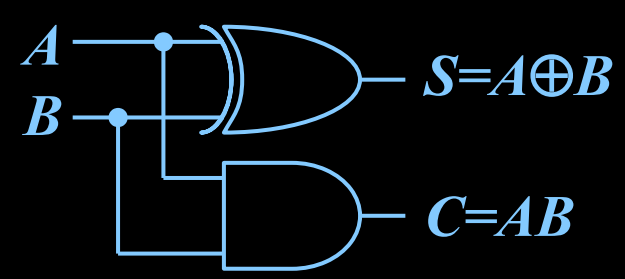
从真值表可得函数表达式

$$S = A\overline{B} + \overline{A}B$$

$$= A \oplus B$$

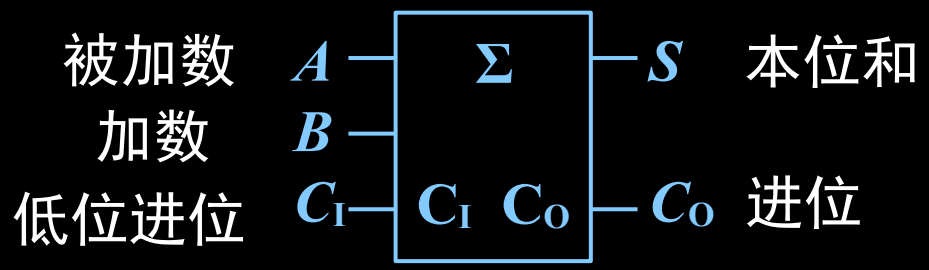
$$C = AB$$

电路如图



当 n 位数值相加时，最低位加法就是半加

(2) 全加器



真值表

A	B	C_I	S	C_O
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

从真值表可得函数表达式，

并进行多输出端综合化减

$$S=\overline{A}\,\overline{B}\,C_I+\overline{A}\,B\,\overline{C}_I+A\,\overline{B}\,\overline{C}_I+A\,B\,C_I$$

$$=\overline{A}\,(\overline{B}\,C_I+B\,\overline{C}_I)+A\,(\overline{B}\,\overline{C}_I+B\,C_I)$$

$$=\overline{A}\,(B\oplus C_I)+A\,(B\odot C_I)$$

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

$$C_O=\overline{A}\,B\,C_I+A\,\overline{B}\,C_I+A\,B\,\overline{C}_I+A\,B\,C_I$$

$$=\overline{A}\,B\,C_I+A\,\overline{B}\,C_I+A\,B$$

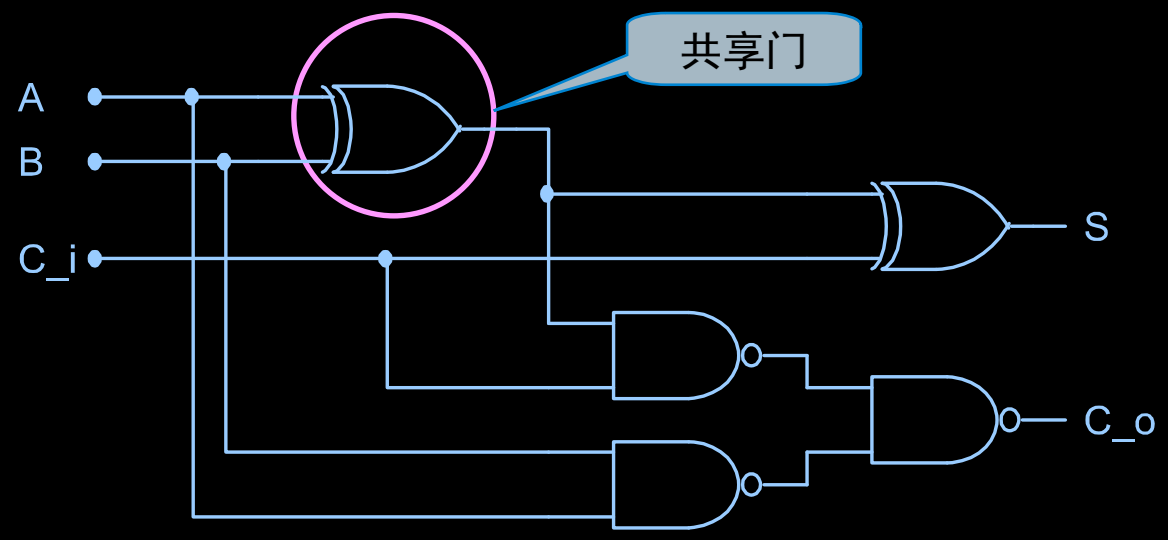
$$=(\overline{A}\,B+A\,\overline{B})\,C_I+A\,B$$

$$=(A\oplus B)\,C_I+A\,B$$

(2) 全加器

$S = A \oplus B \oplus C_I$

$C_O = (A \oplus B)\, C_I + A\, B$

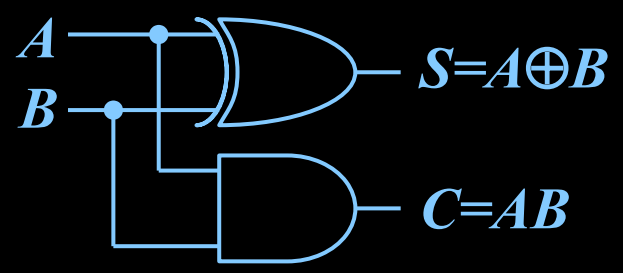


(2) 全加器

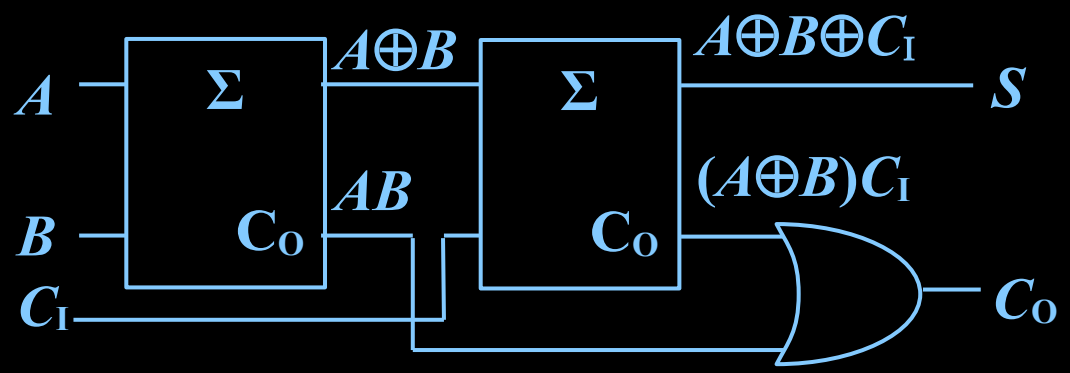
$S = A \oplus B \oplus C_I$

$C_O = (A \oplus B)\, C_I + A\, B$

半加器电路



由半加器可构成全加器电路



(2) 全加器

全加器真值表及应用

A	B	C_I	S	C_O
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

ABC 有奇数个 1 时 *S* 为 1

ABC 有偶数个 1 时 *S* 为 0

S 可作为三位二进制码的奇偶校验器

C_O 还有啥功能？

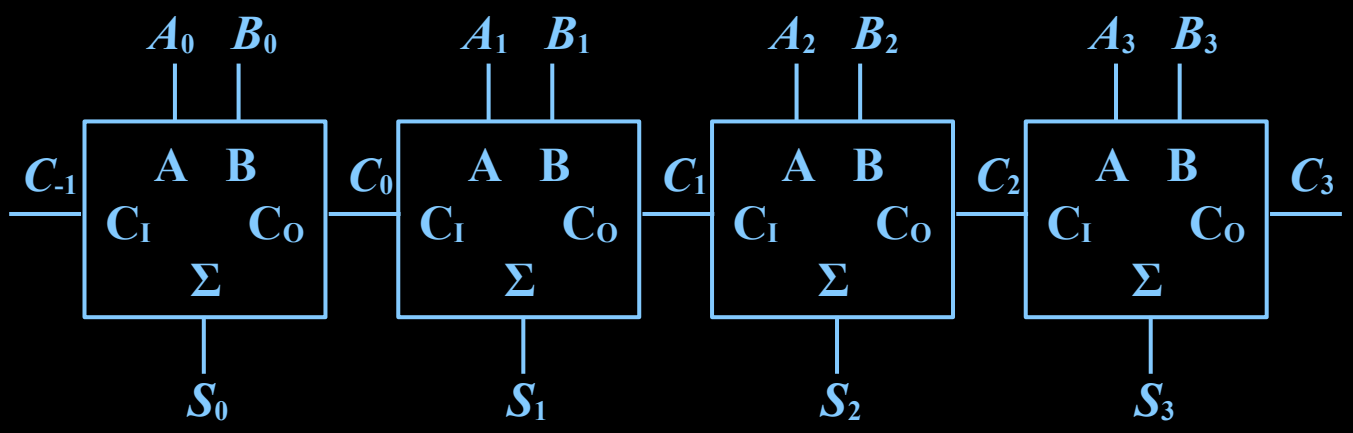
三输入多数表决器

2. 多位数加法器

(1) 串行进位加法器

$$\begin{array}{r} A_3\ A_2\ A_1\ A_0 \\ B_3\ B_2\ B_1\ B_0 \\ +\ C_3\ C_2\ C_1\ C_0\ C_{-1} \\ \hline S_3\ S_2\ S_1\ S_0 \end{array}$$

用全加器实现的串行进位加法器

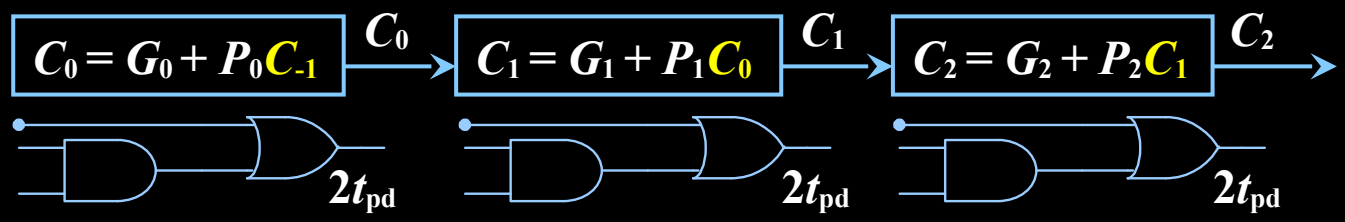


必须前级产生进位后，后级加法才有效，位数越多，速度越慢。

定义第 n 位信号 $C_n = \underbrace{A_n B_n}_{G_n} + \underbrace{(A_n \oplus B_n)}_{P_n} C_{n-1}$， $S_n = \underbrace{A_n \oplus B_n}_{P_n} \oplus C_{n-1}$

设中间变量 G_n 和 P_n： $G_n = A_n B_n$， $P_n = A_n \oplus B_n$， 逻辑门时延为 t_{pd}

则 $C_n = G_n + P_n C_{n-1}$， $S_n = P_n \oplus C_{n-1}$



得到 C_2 的时延为 $3 \times 2 \times t_{pd} + t_{pd}$

得到 C_n 的时延为 $(n+1) \times 2 \times t_{pd} + t_{pd}$

2. 多位数加法器

(2) 超前进位加法器

4 位超前进位产生电路

中间变量 $G_n = A_n B_n$, $P_n = A_n \oplus B_n$

$C_n = G_n + P_n C_{n-1}$

$C_0 = G_0 + P_0 C_{-1}$

$C_1 = G_1 + P_1 C_0$

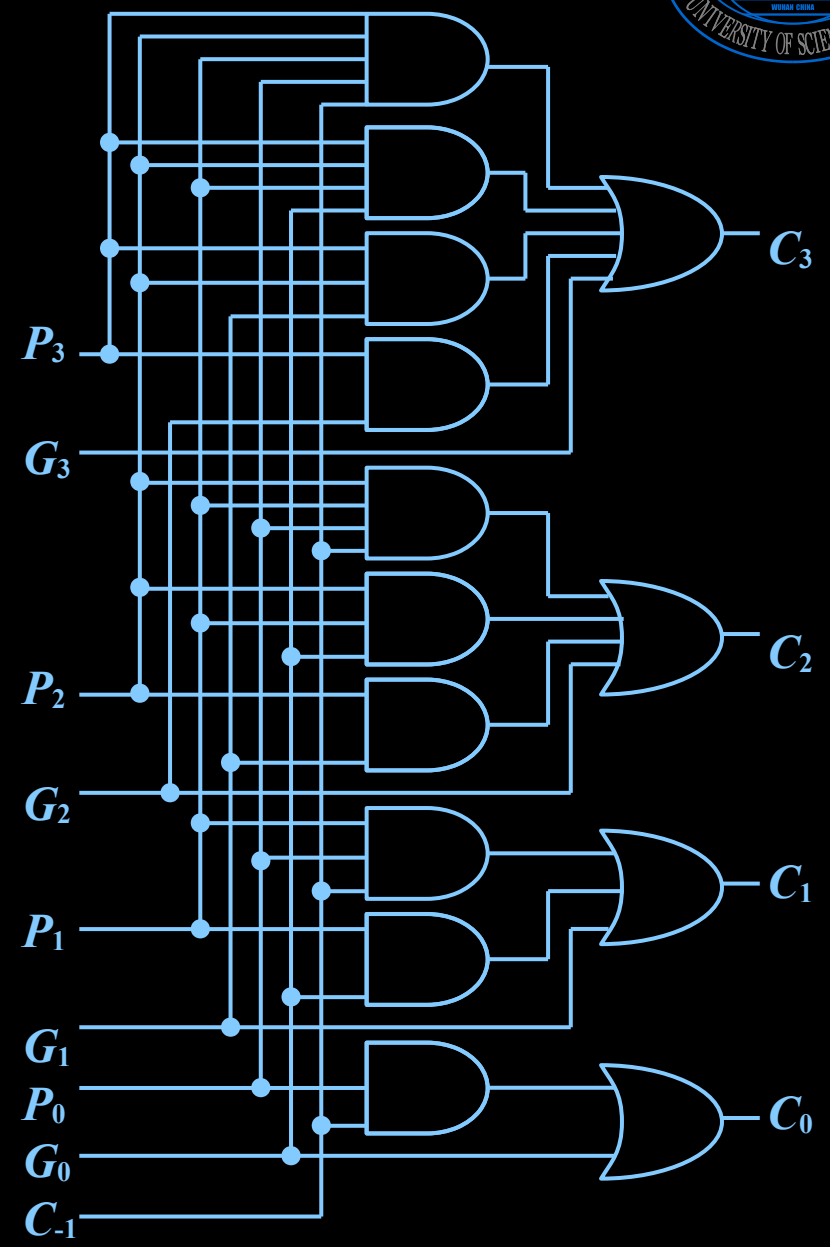
$= G_1 + P_1 G_0 + P_1 P_0 C_{-1}$

$C_2 = G_2 + P_2 C_1$

$= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1}$

$C_3 = G_3 + P_3 C_2$

$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1}$

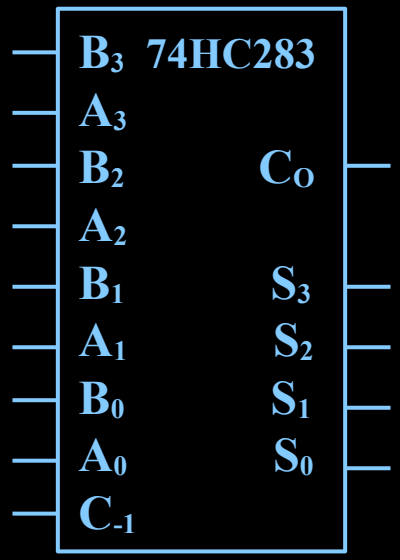


得到 C_n 只需 3 t_{pd}

2. 多位数加法器

(2) 超前进位加法器

集成 4 位加法器 74LS283 结构如图



$G_n = A_n B_n$, $P_n = A_n \oplus B_n$

$S_n = P_n \oplus C_{n-1}$

B3
A3
B2
A2
B1
A1
B0
A0
C-1

P3 G3
P2 G2
P1 G1
P0 G0
C-1

超前进位产生电路

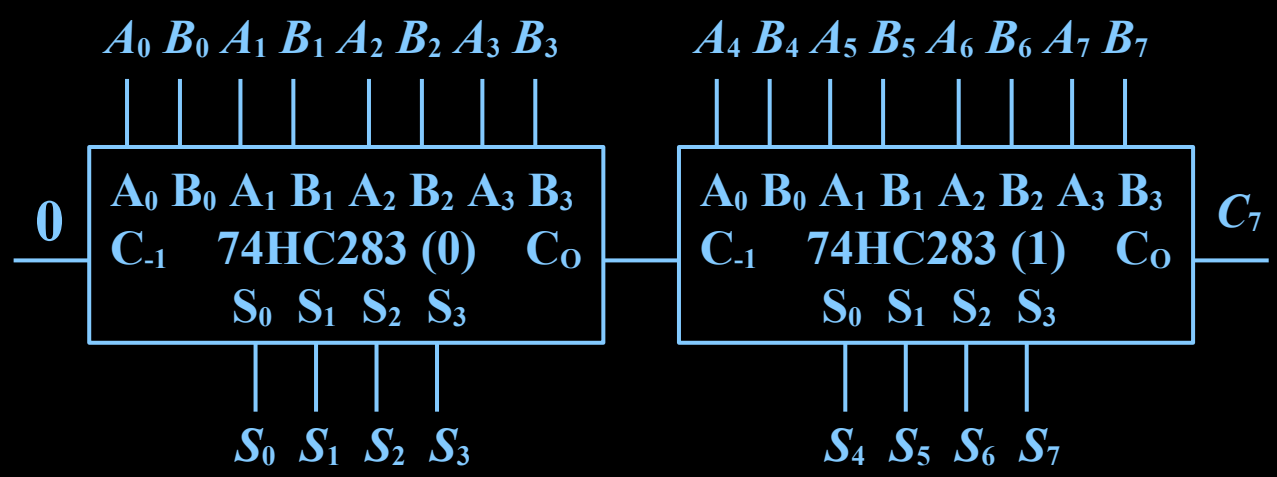
C3 C2 C1 C0

CO
P3 S3
P2 S2
P1 S1
P0 S0

2. 多位数加法器

(2) 超前进位加法器

例：用两片 74LS283 构成一个 8 位二进制数加法器



在片内是超前进位，而片与片之间是串行进位

2. 多位数加法器

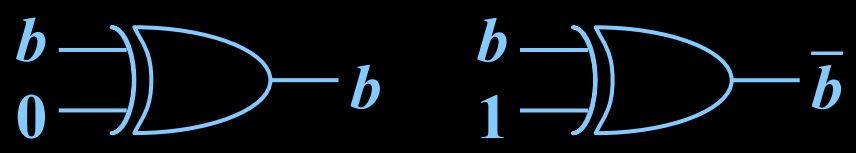
(2) 超前进位加法器

例：用 74HC283 实现 4 位无符号数减法，并输出差的原码

解：用加法器实现减法，即 $x_B - y_B = x_B + (\sim y_B + 1_B)$

- 无符号数之差有正有负，但输出的 4 位数只表示值位，需扩展符号位
  - 若差为正，四位输出是其绝对值
  - 若差为负，四位输出是其绝对值关于 2 的补，须再求补，得其绝对值
- 可采用异或门由借位（扩展的符号位）控制是否对差的值位求补

$b \oplus 0 = b$，　　$b \oplus 1 = \overline{b}$



- 可利用 74HC283 的进位输入 C_{-1}，实现值位取反后加 1，得到绝对值

2. 多位数加法器

(2) 超前进位加法器

例：用 74HC283 实现 4 位无符号数减法，并输出差的原码

解：借位分析

(1) $x - y \geq 0$ 时，如 $5 - 1 = 4$

$0101_B - 0001_B = 0101_B + 1110_B + 1_B$

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ \ 1\ 1\ 1\ 1 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$

(2) $x - y < 0$ 时，如 $11 - 14 = -3$

$1011_B - 1110_B = 1011_B + 0001_B + 1_B$

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ +\ 0\ 0\ 1\ 0 \\ \hline 0\ 1\ 1\ 0\ 1 \end{array}$$

值位求补 ⟹ 0011

无符号数减法的借位是最高进位之反

借位可用于差的符号位扩展

$x - y \geq 0$ 时，借位为 0，即差为正数

$x - y < 0$ 时，借位为 1，即差为负数

值位求补后，得到原码形式的值位

2. 多位数加法器

(2) 超前进位加法器

例：用 74HC283 实现 4 位无符号数减法，并输出差的原码

解：电路如图

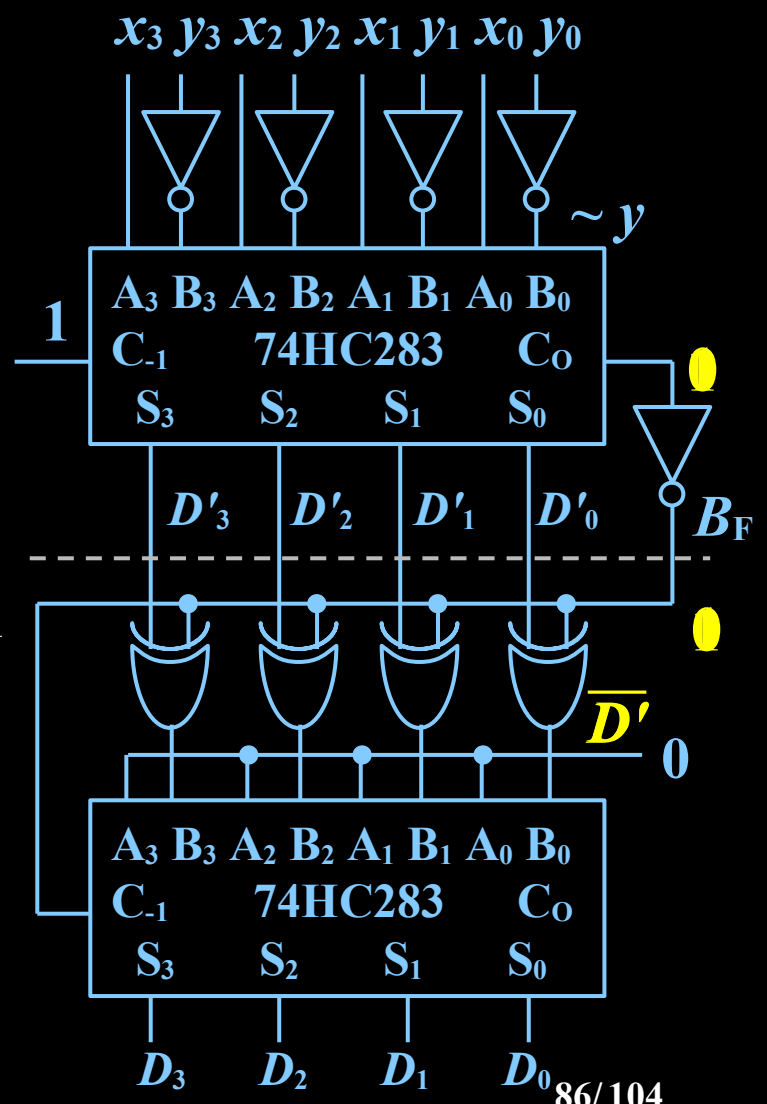
$D' = x - y = x + (\sim y + 1)$，是差的值位

$C_O = 1$ 时，借位 $B_F = \overline{C}_O = 0$，差为正

D' 为正数原码值位，故 $D = D' + 0 + 0$

$C_O = 0$ 时，借位 $B_F = \overline{C}_O = 1$，差为负

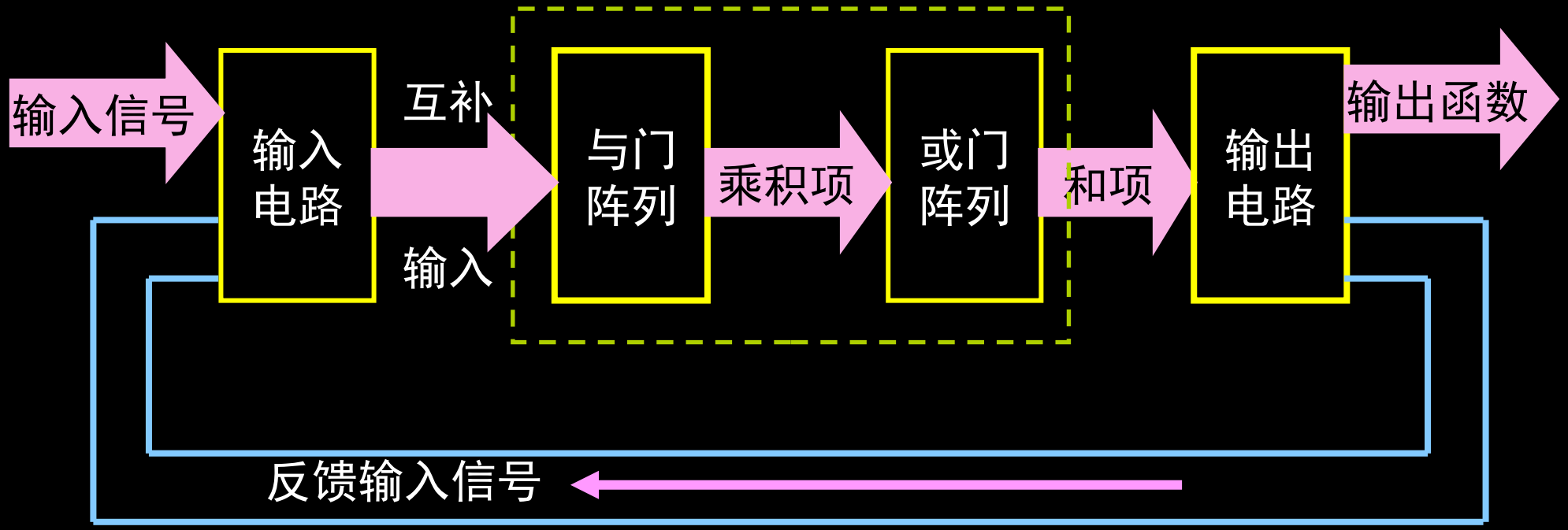
D' 为负数补码值位，故原码值位 $D = \sim D' + 0 + 1$



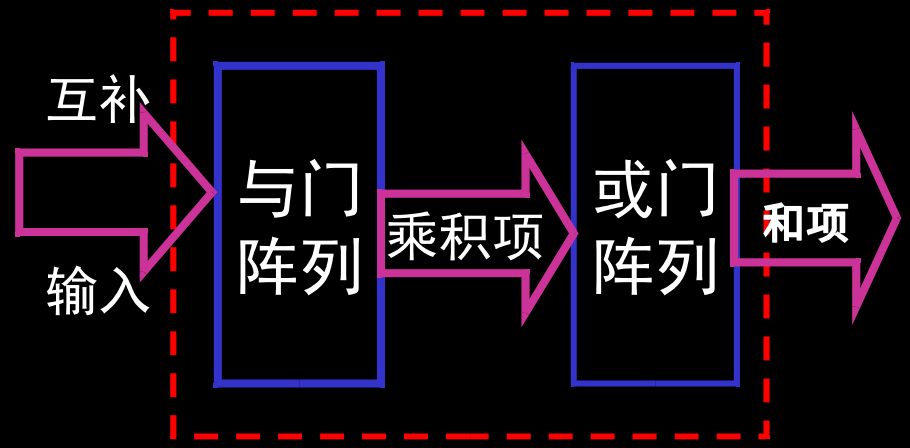
- **PLD： Programmable Logic Device**

可编程逻辑器件是一种可由用户编程设置逻辑功能的器件

1. PLD 的结构

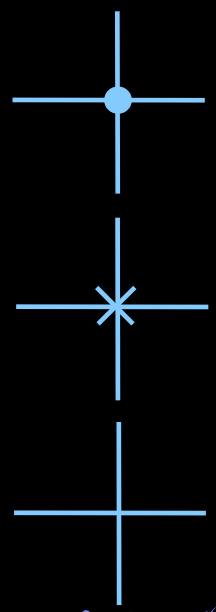


2. PLD 的表示方法



框图

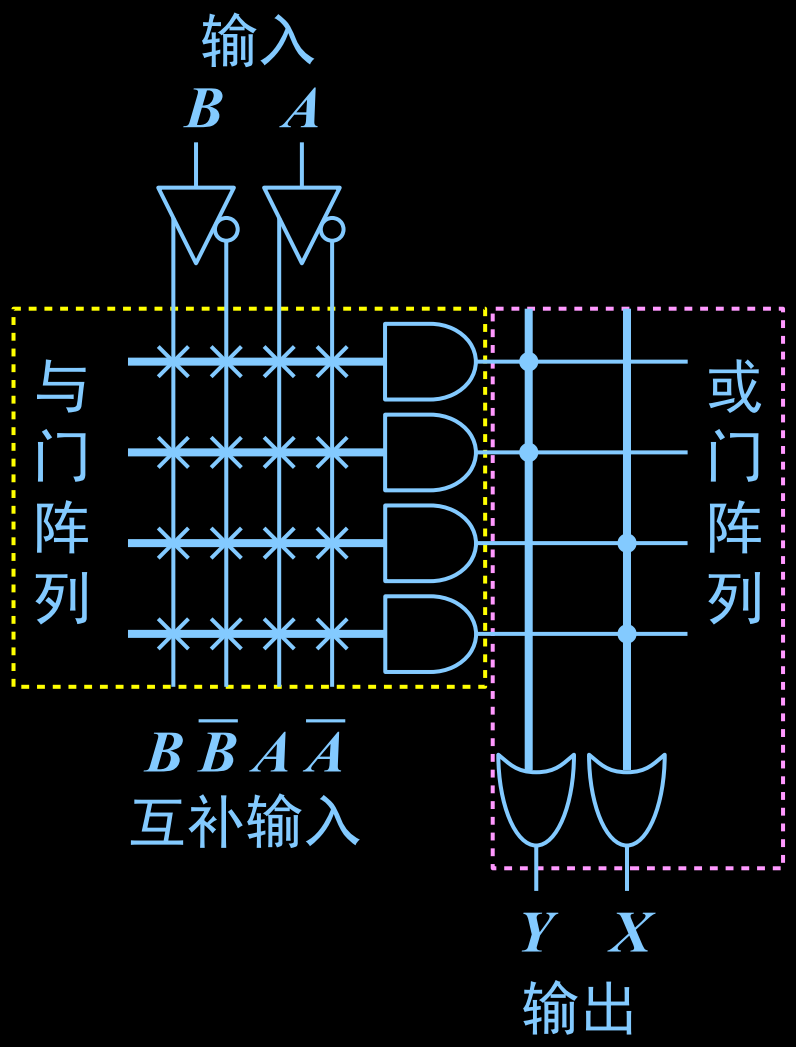
(1) 连接的方式



硬线连接单元

被编程接通单元

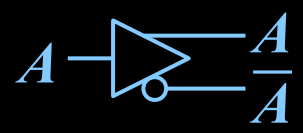
被编程擦除单元



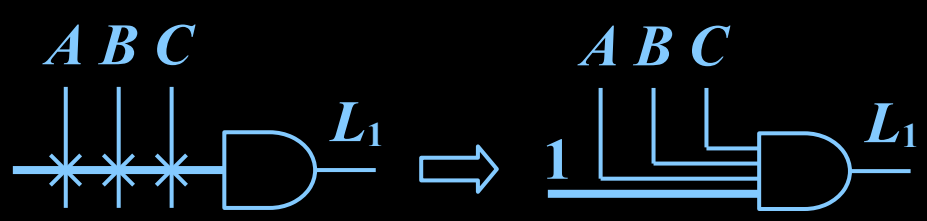
基本电路结构

2. PLD 的表示方法

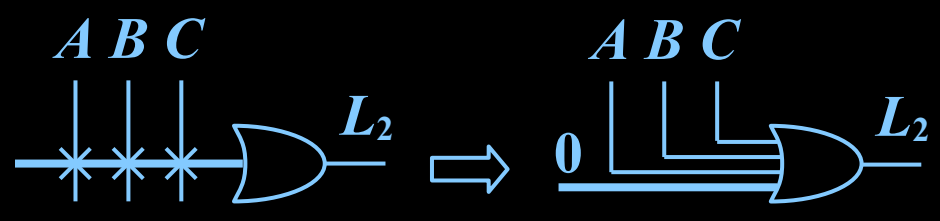
(2) 基本门电路的表示方式



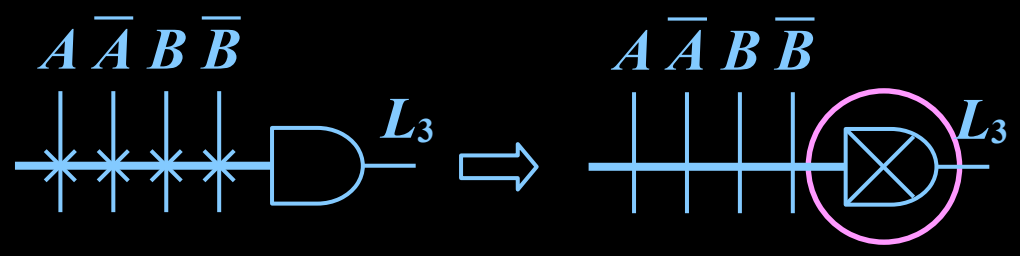
输入缓冲器



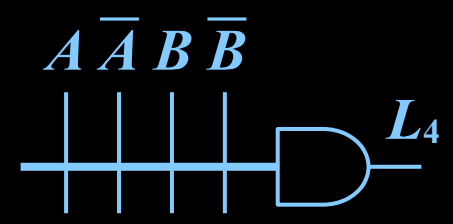
与门：$L_1 = A \cdot B \cdot C$



或门：$L_2 = A + B + C$

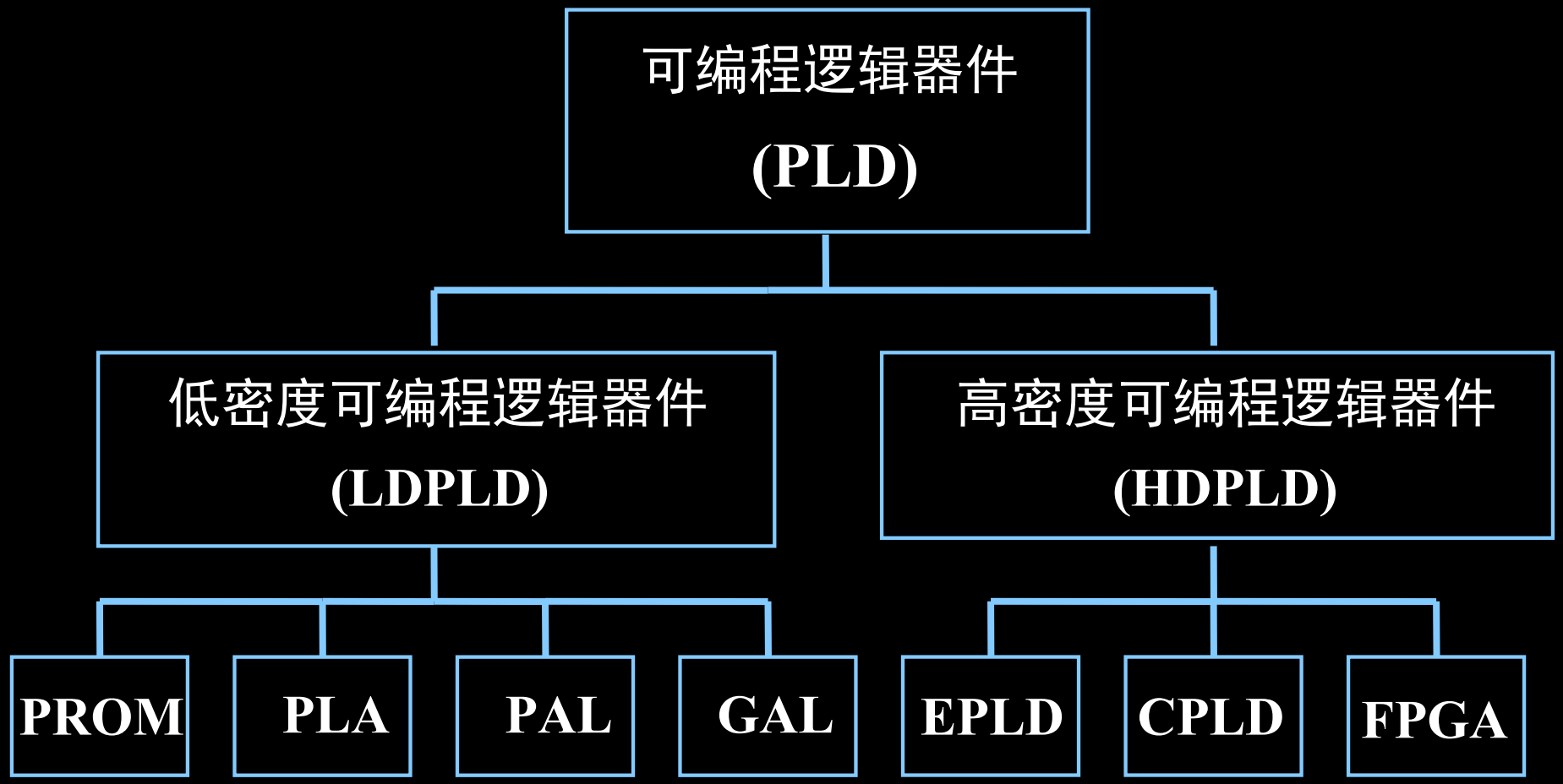


输出恒为 0 的与门：$L_3 = A\,\overline{A}\,B\,\overline{B} = 0$



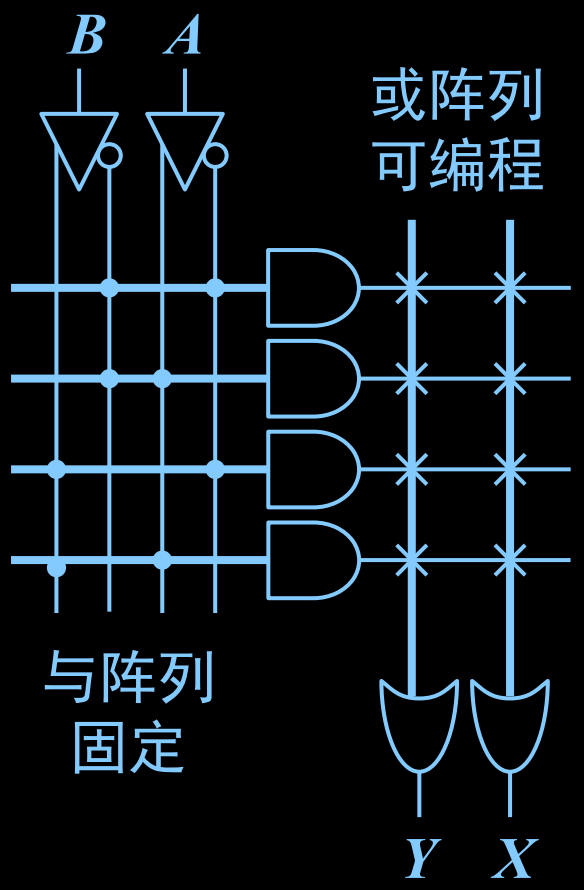
输出恒为 1 状态：$L_4 = 1$

3. PLD 的分类

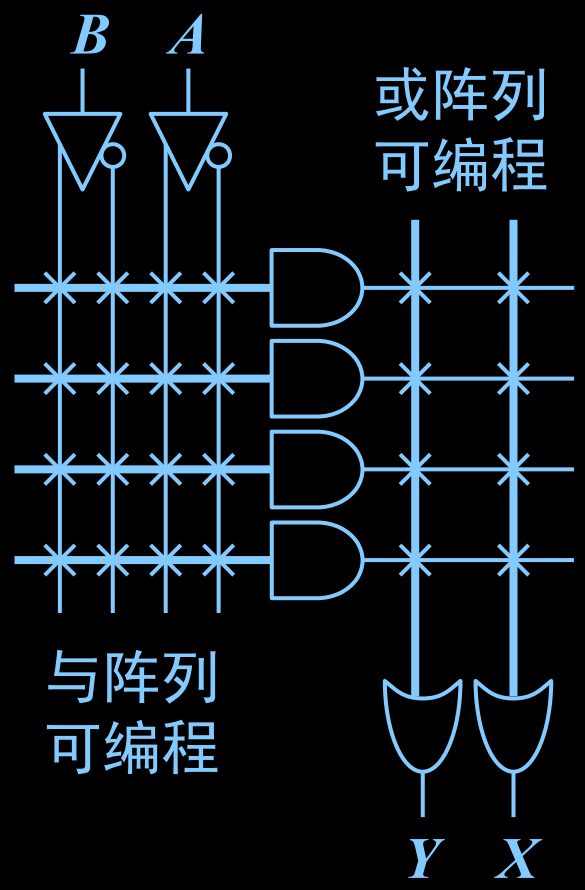

可编程逻辑器件
(PLD)
低密度可编程逻辑器件
(LDPLD)
高密度可编程逻辑器件
(HDPLD)
PROM
PLA
PAL
GAL
EPLD
CPLD
FPGA

3. PLD 的分类

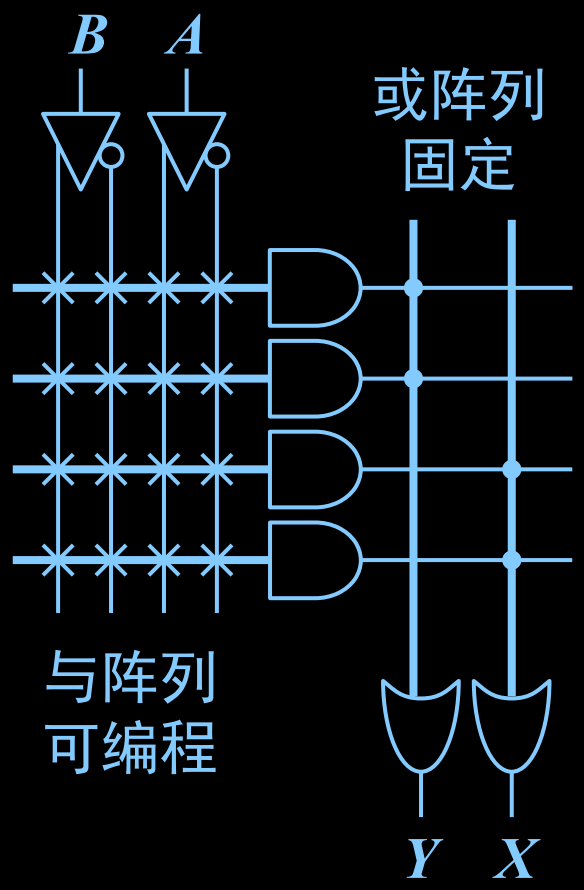
与阵列固定，或阵列可编程 (PROM)



与阵列、或阵列均可编程 (PLA)



与阵列可编程，或阵列固定 (PAL 和 GAL 等)

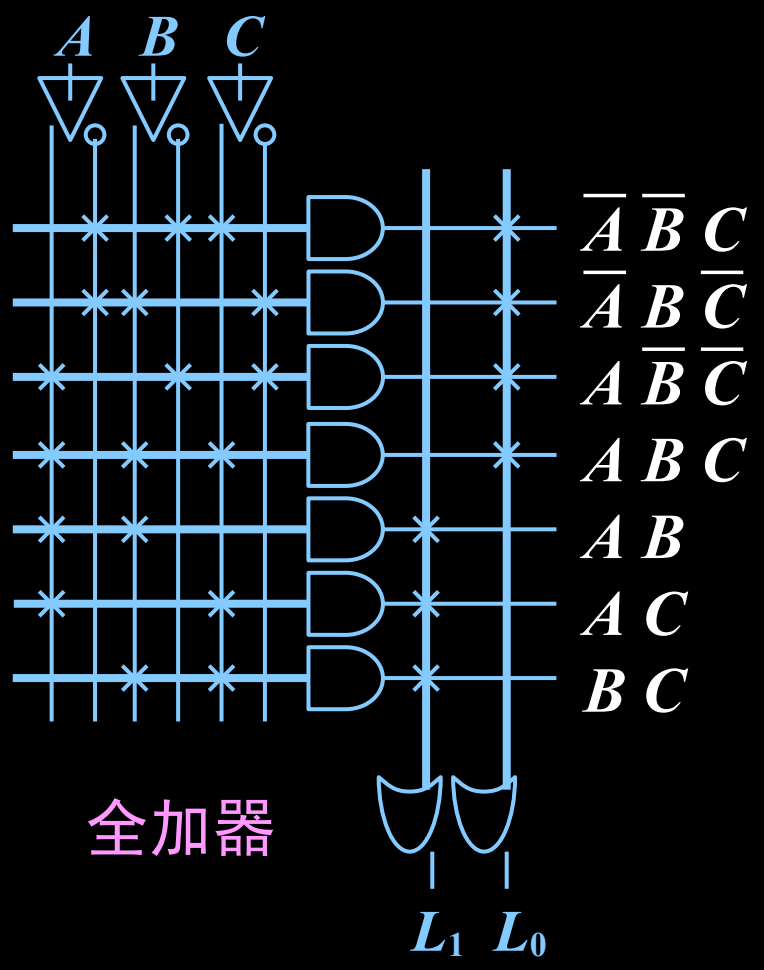


1. 可编程逻辑阵列 PLA：与、或阵列都可以编程

例：如图由 PLA 构成的逻辑电路，试写出其逻辑表达式，并描述功能

$$L_0 = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,B\,C$$

$$L_1 = A\,B + A\,C + B\,C$$



真值表　　和　进位

A	B	C	L_0	L_1
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

2. 可编程阵列逻辑 PAL

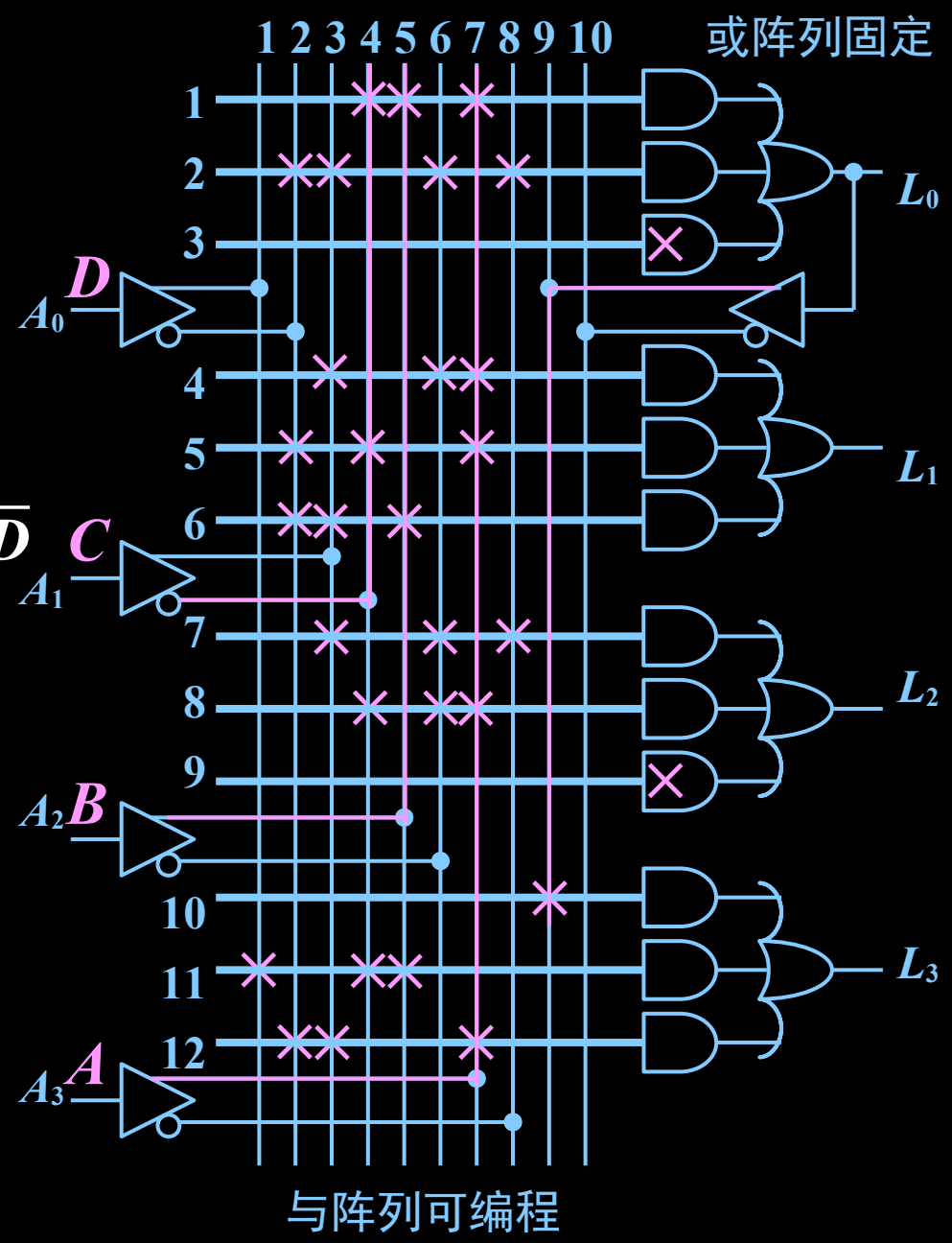
例：由 PAL 实现

$$L_0 = \underline{A\,B\,\overline{C}} + \underline{\overline{A}\,\overline{B}\,C\,\overline{D}}$$

$$L_1 = A\,\overline{B}\,C + A\,\overline{C}\,\overline{D} + B\,C\,\overline{D}$$

$$L_2 = \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C}$$

$$L_3 = \underbrace{A\,B\,\overline{C} + \overline{A}\,\overline{B}\,C\,\overline{D}}_{L_0} + B\,\overline{C}\,D + A\,C\,\overline{D}$$



组合逻辑电路的行为级描述一般使用：

- assign 结构：连续赋值语句
- always 结构：过程赋值语句
- 条件语句（ if-else ）
- 多路分支语句（ case-endcase ）
- for 循环语句等

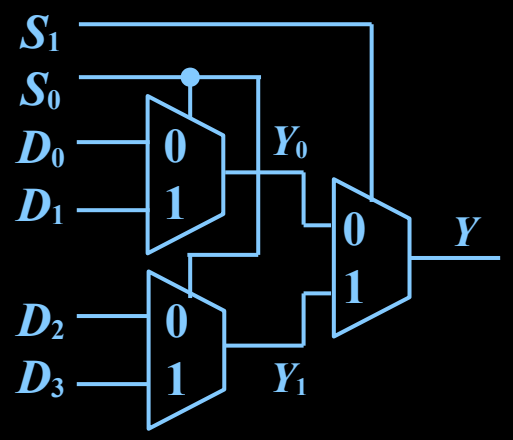
```
if (condition_expr1) true_statement1;
else if (condition_expr2) true_statement2;
else if (condition_expr3) true_statement3;
...
else default_statement;

for (initial_assignment; condition; step_assignment)
  statement;
```

```
case (case_expr)
  item_expr1: statement1;
  item_expr2: statement2;
  ...
  default: default_statement;
endcase
```

casex：条件含无关项 x

casez：条件含高阻态 z

例： 4 选 1 数据选择器的行为建模




```
module mux4to1(output reg Y, input[3:0] D, input[1:0] S);
  always @(D, S)
    if (S == 2'b00)      Y = D[0];
    else if (S== 2'b01)  Y = D[1];
    else if (S== 2'b10)  Y = D[2];
    else                 Y = D[3];
endmodule
```

4.6.1 组合逻辑电路的行为级建模

例：具有使能端 $\overline{E}$ 的 4 选 1 数据选择器的行为建模。当 $\overline{E}=0$ 时，数据选择器工作，$\overline{E}=1$ 时，禁止工作，输出为 0

```
module mux4to1_bh(output reg Y, input[3:0] D, input[1:0] S, input _E);
  always @(D, S, _E)
  begin
    if (_E == 1)  Y = 0; // _E = 1时，输出为0
    else                 // _E = 0时，选择器工作
      case (S)
        2'd0: Y = D[0];
        2'd1: Y = D[1];
        2'd2: Y = D[2];
        2'd3: Y = D[3];
      endcase
  end
endmodule
```

例：具有高电平使能的 3-8 线译码器的行为建模

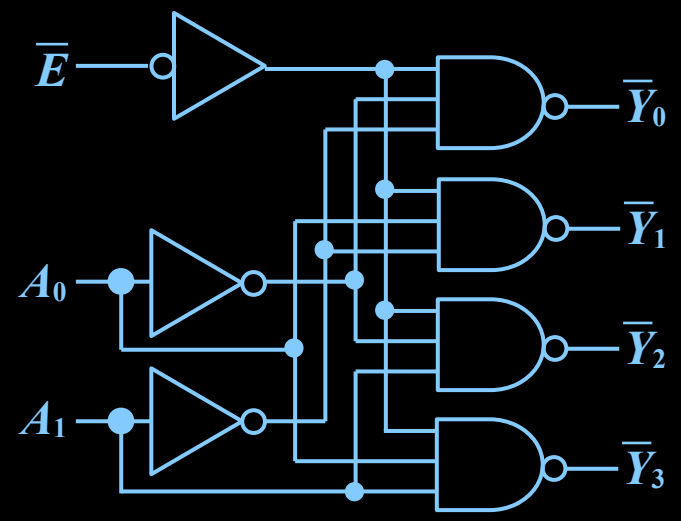
```
module encoder3to8_bh(output reg[7:0] _Y,
                      input[2:0] A, input E);
  integer k;                        // 声明一个整型变量 k
  always @(A, E)
  begin
    _Y = 8'b1111_1111;              // 设译码器输出的默认值
    for(k = 0; k <= 7; k = k + 1) // 对 8 位输出分别进行处理
      if ((E == 1) && (A == k))
        _Y[k] = 0;                  // 当 _E=1 时，根据 A 进行译码
      else
        _Y[k] = 1;                  // 处理使能无效或输入无效的情况
  end
endmodule
```

例：用条件运算符实现 2 选 1 数据选择器的数据流和行为建模

```
module mux2x1_df(output L, input A, B, SEL);
  assign L = SEL ? A : B;
endmodule
```

```
module mux2x1_df(output reg L, input A, B, SEL);
  always @( A, B, SEL ) // 用 always 语句和条件运算符建模
    L = SEL ? A : B;
endmodule
```

例： 2-4 线译码器的数据流建模




```
module decoder_df(output[3:0] _Y, input[1:0] A, input _E);
  assign _Y[0] = ~(~A[1] & ~A[0] & ~_E);
  assign _Y[1] = ~(~A[1] & A[0] & ~_E);
  assign _Y[2] = ~(A[1] & ~A[0] & ~_E);
  assign _Y[3] = ~(A[1] & A[0] & ~_E);
endmodule
```

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

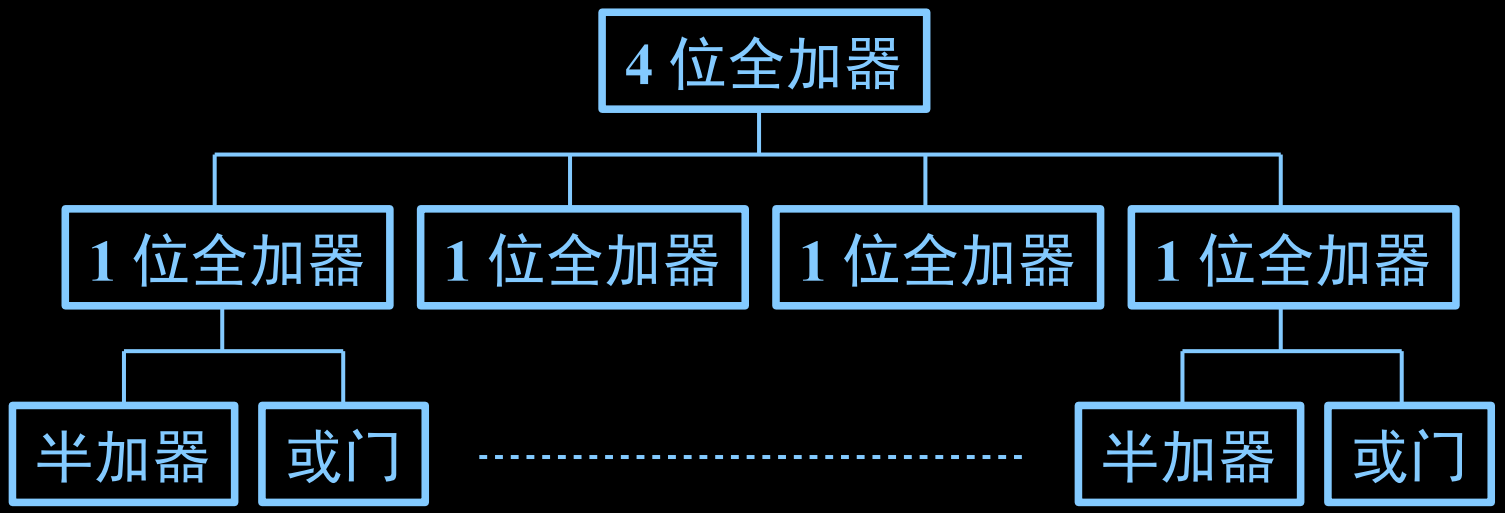
$$\overline{Y}_1 = \overline{\overline{A}_1 A_0 \overline{E}}$$

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

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

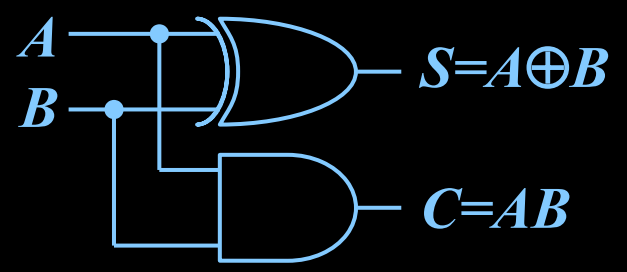
分层次的电路设计：

将两个或多个模块组合起来描述电路逻辑功能的方法



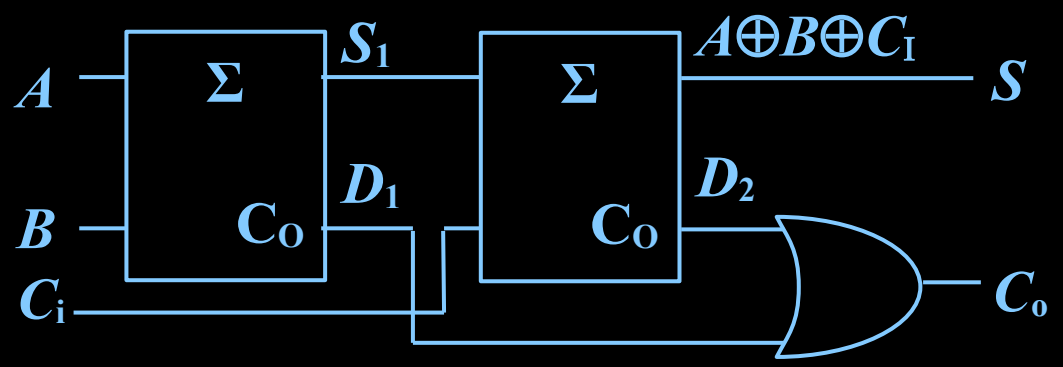
1 位半加器

```
module halfadder(output S, C, input A, B);
  xor g1(S, A, B);
  and g2(C, A, B);
endmodule
```

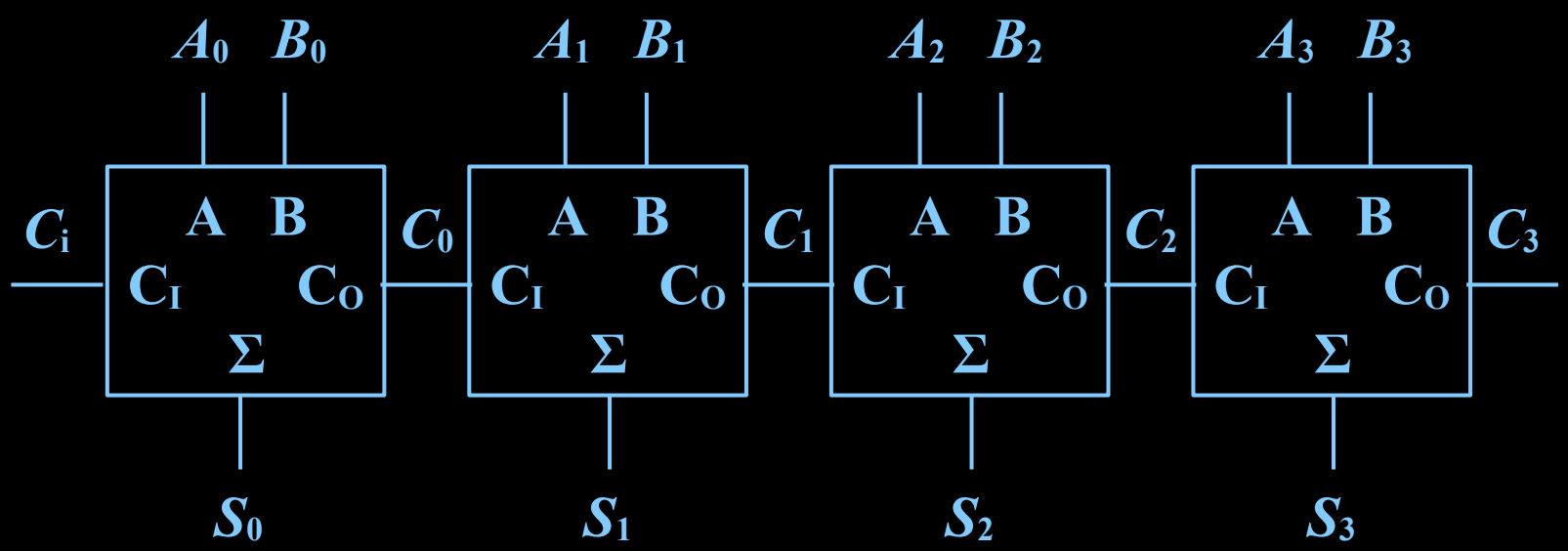


1 位全加器

```
module fulladder(output S, Co, input A, B, Ci);
  wire S1,D1,D2; // 内部节点信号
  halfadder HA1 (S1,D1,A,B);
  halfadder HA2 (S,D2,S1,Ci);
  or g1(Co,D2,D1);
endmodule
```



4 位全加器




```
module adder_4bit(output[3:0] S, output C3, input[3:0] A, B, input Ci);
  wire C0,C1,C2;  // 内部进位信号
  fulladder FA0 (S[0],C0,A[0],B[0],Ci), FA1 (S[1],C1,A[1],B[1],C0),
            FA2 (S[2],C2,A[2],B[2],C1), FA3 (S[3],C3,A[3],B[3],C2);
endmodule
```

作业

康华光教材 7 版

4.1.2

4.2.2

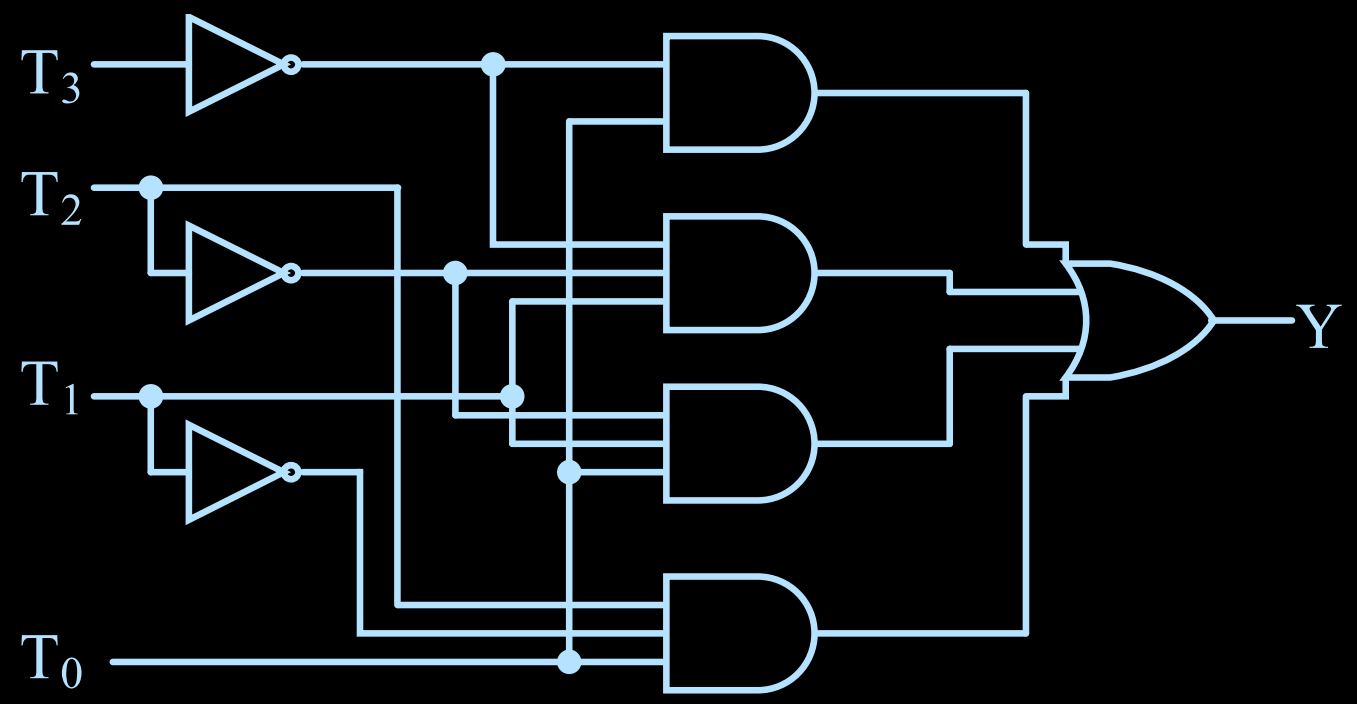
4.2.9

4.3.3

4.4.8

1. 用代数法化简函数 F 为最简与或式

$$F(A,B)=A\oplus B\odot\overline{\overline{A}\cdot\overline{B}}$$

$$F(A,B,C,D,E)=A\cdot B+A\cdot\overline{D}\cdot E+A\cdot\overline{C}+\overline{B\odot C}$$

2. 分析逻辑电路功能