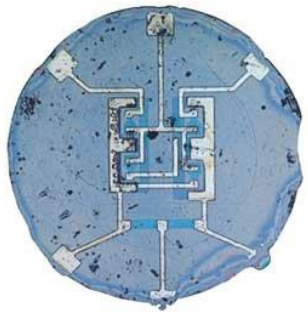
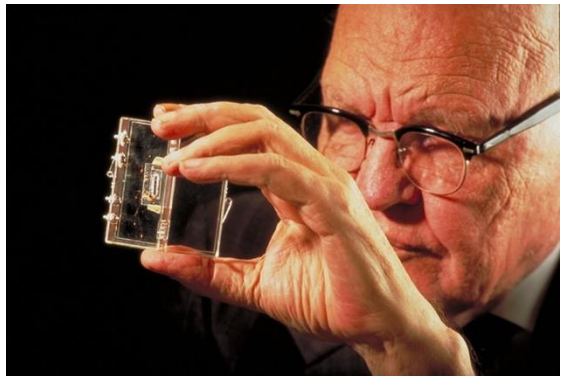
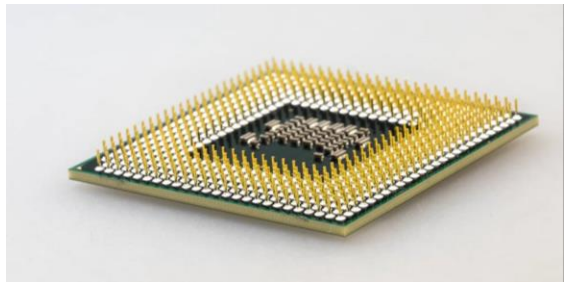
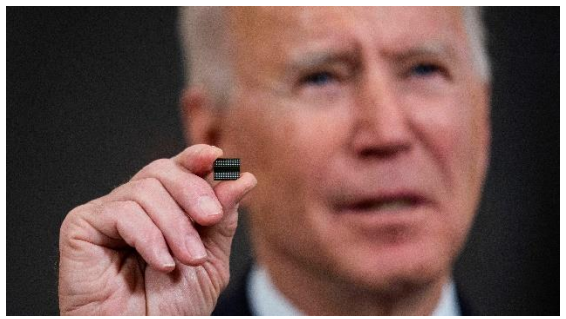
数字逻辑设计

高翠芸
School of Computer Science
gaocuiyun@hit.edu.cn





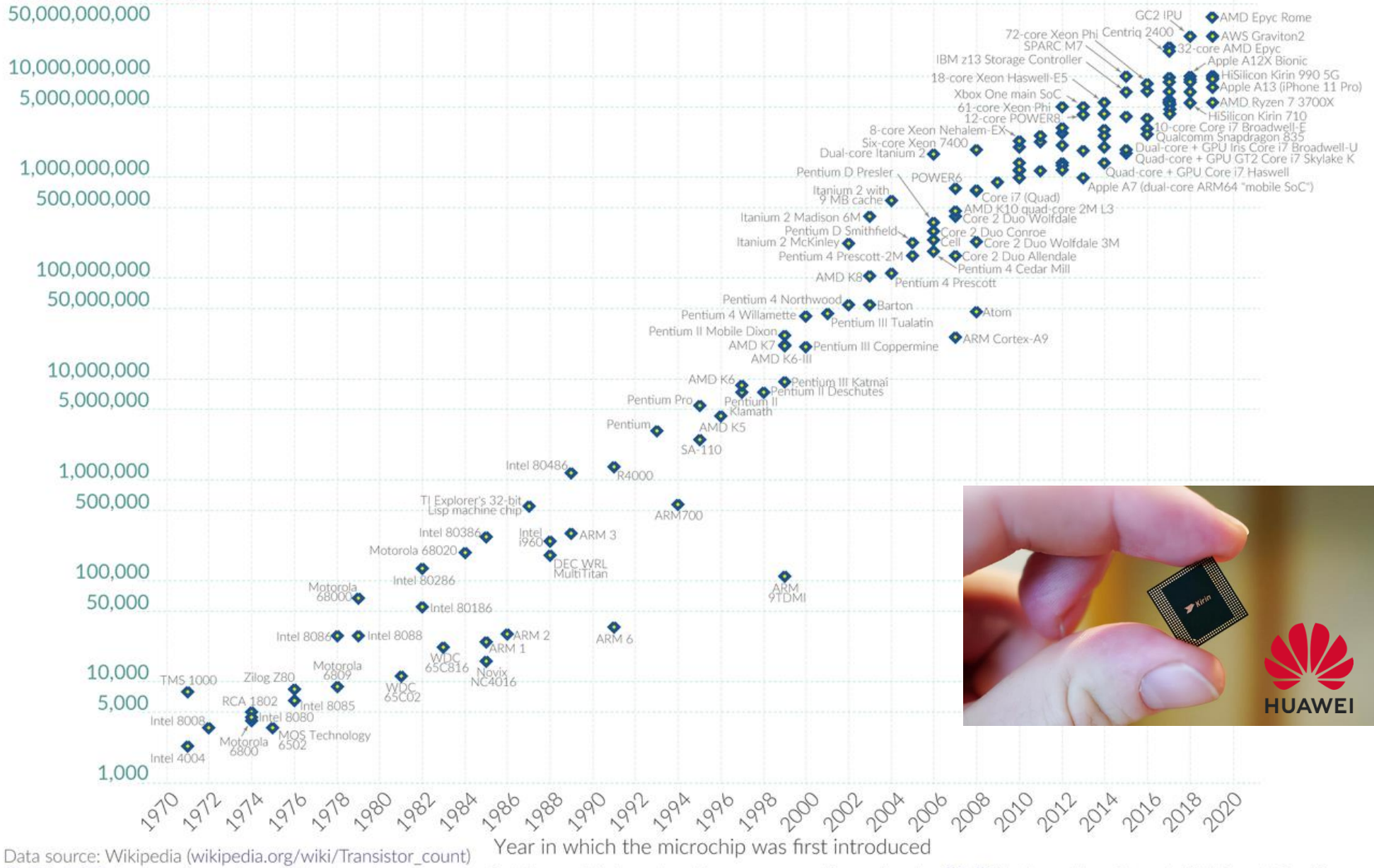




Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

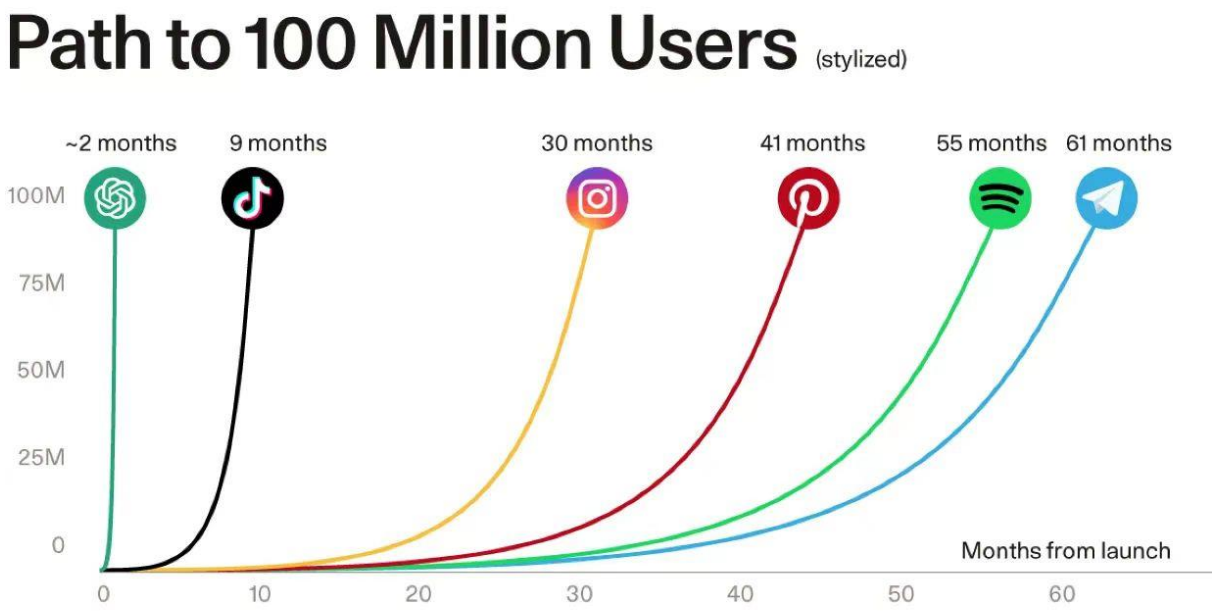


Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

英伟达3个月卖出800吨H100！云服务供应商的大规模H100集群容量即将耗尽，全球陷入GPU短缺——整体算来，全球公司需要约432000张H100。

by 新智元



集成电路的分类

分类	单芯片内集成的逻辑门数量	集成内容	器件封装	需要掌握的内容
小规模（SSI）				
中规模（MSI）				
大规模（LSI）				
超大规模（VLSI）				

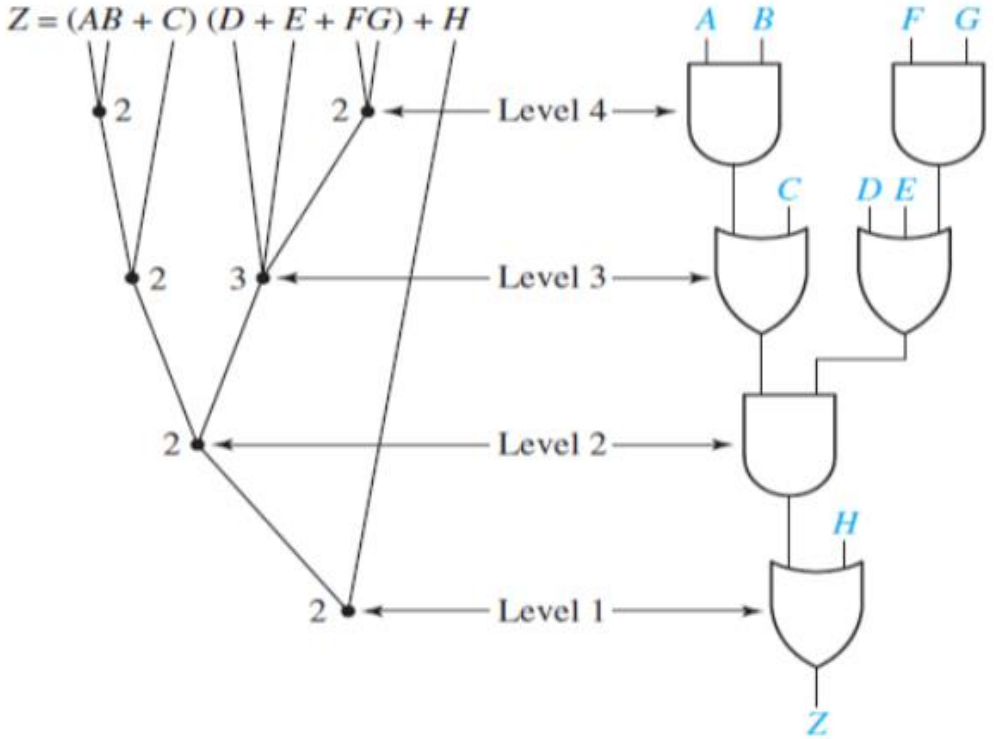
多级门电路

前提：忽略输入端原、反变量的差别.

门的级数——

电路输入与输出之间串联的门的最大数值

- **二级电路**
  - *AND-OR* 电路（积之和）
  - *OR-AND* 电路（和之积）
- **三级电路**
  - *OR-AND-OR* 电路
- **各门没有特定的排列顺序**
- **输出门可以使与门也可以是或门**



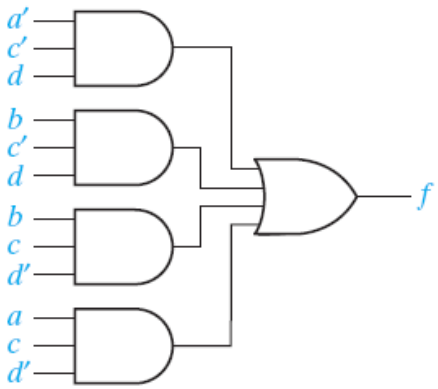
多级门电路

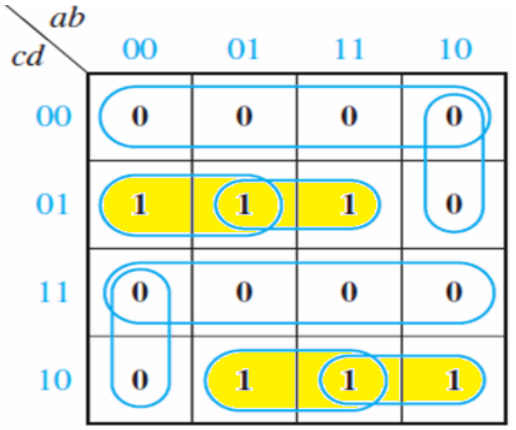
1. 二级电路

AND-OR 电路（积之和）

$$f = a'c'd + bc'd + bcd' + acd'$$

5个门, 16 个输入端

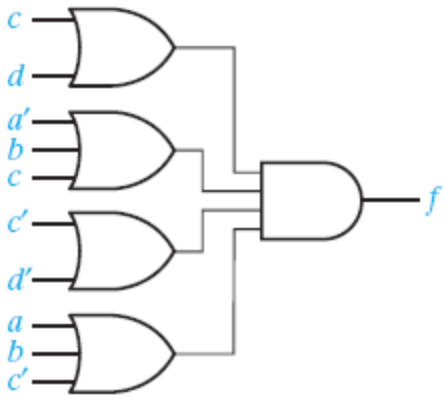




OR-AND 电路（和之积）

$$f = (c + d)(a' + b + c)(c' + d')(a + b + c')$$

5个门, 14 个输入端



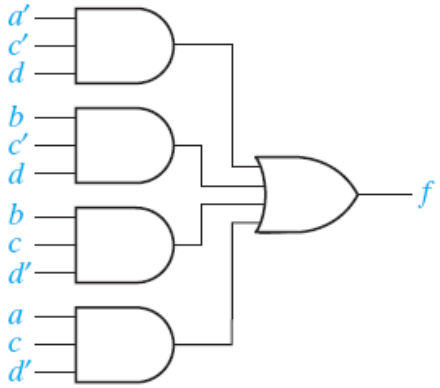
多级门电路

1. 二级电路

AND-OR 电路（积之和）

$$f = a'c'd + bc'd + bcd' + acd'$$

5个门, 16 个输入端

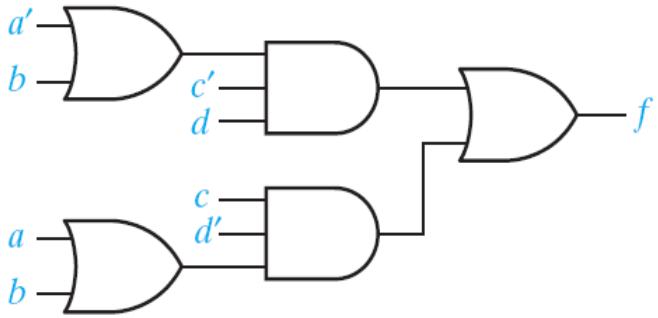


2. 三级电路

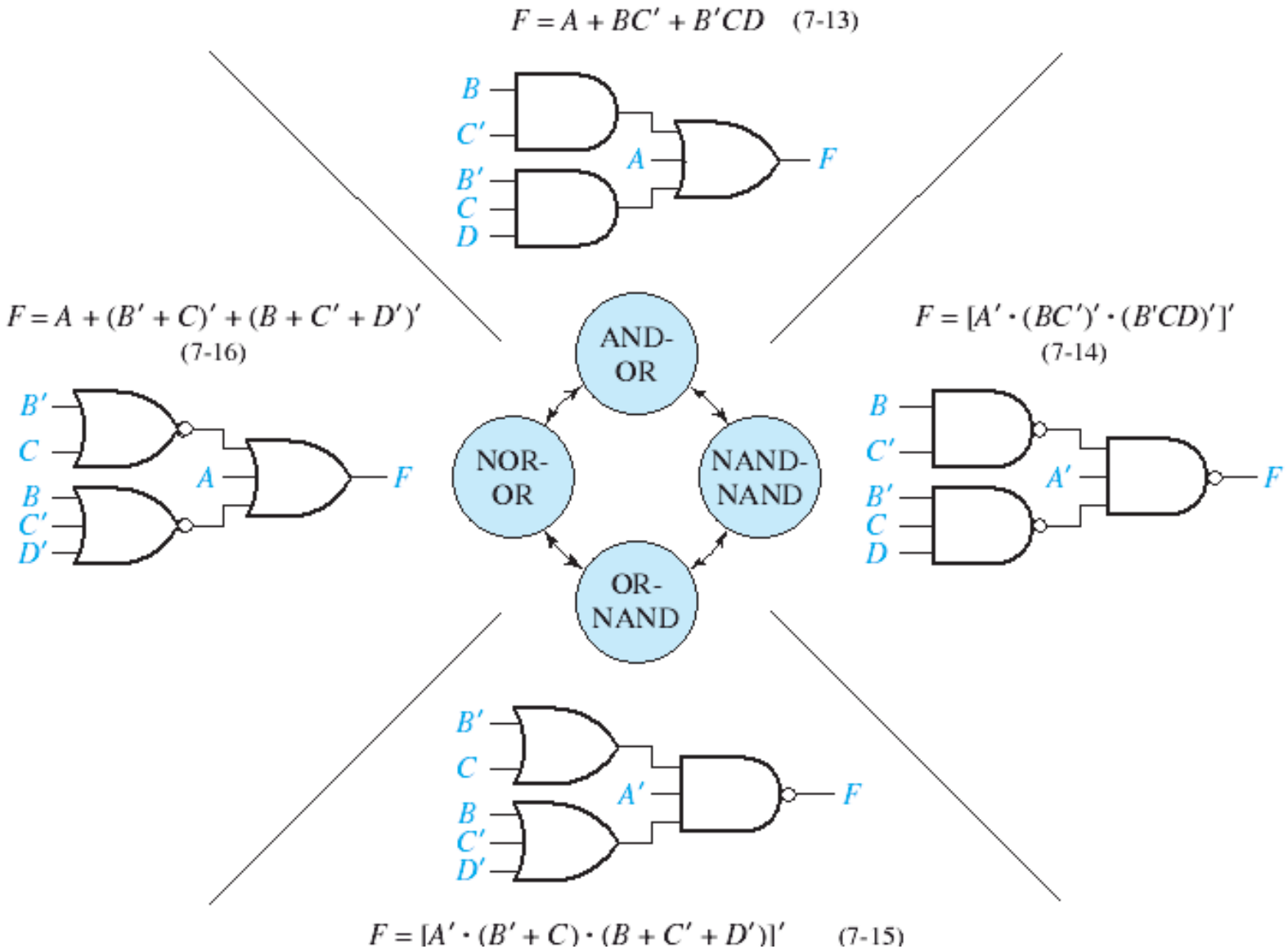
OR-AND-OR 电路

$$f = c'd(a' + b) + cd'(a + b)$$

5个门, 12 个输入端

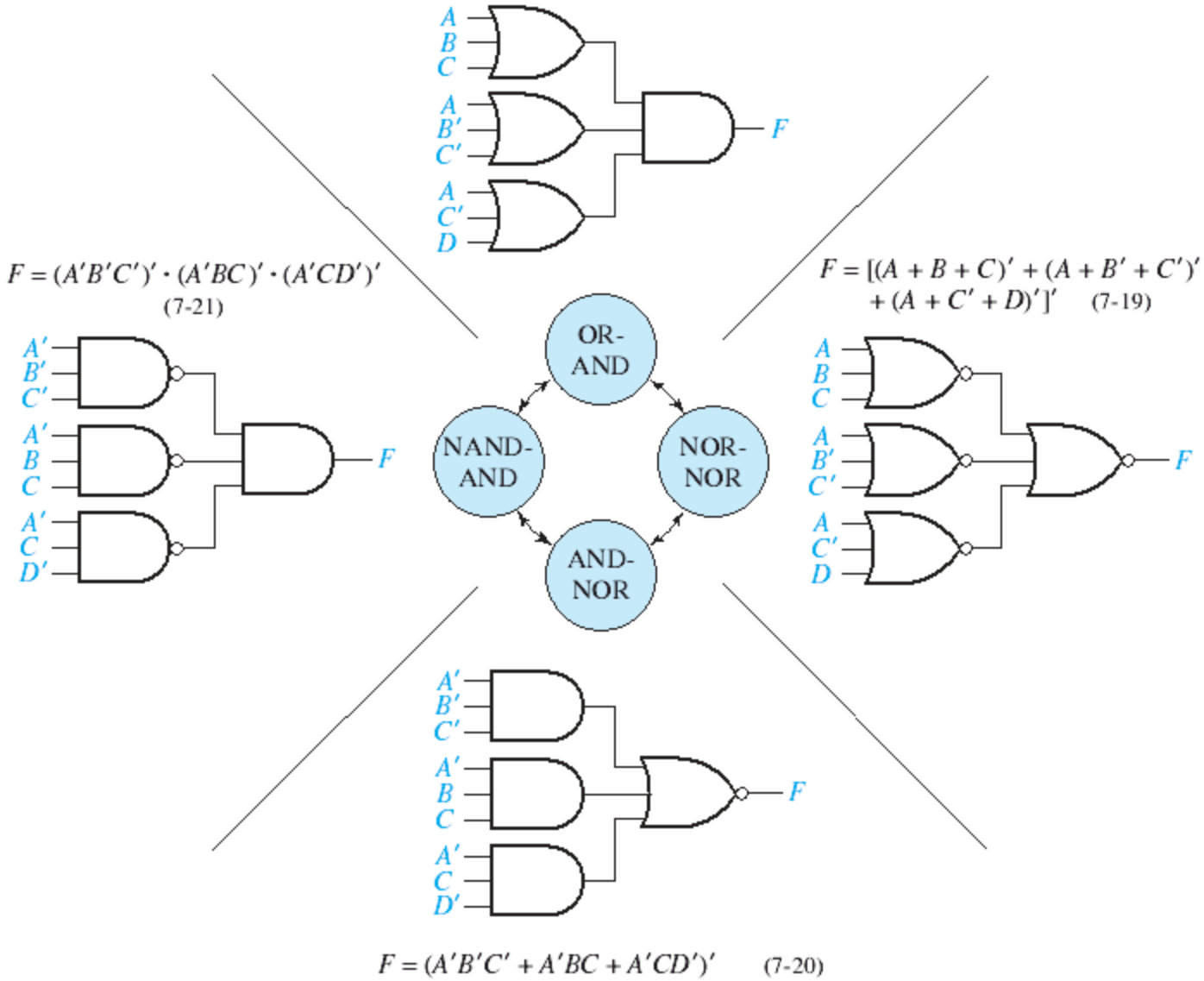


二级门电路的8种基本形式——1



二级门电路的8种基本形式——2

$$F = (A + B + C)(A + B' + C')(A + C' + D) \qquad (7\text{-}18)$$



$F = (A'B'C')' \cdot (A'BC)' \cdot (A'CD')'$ (7-21)

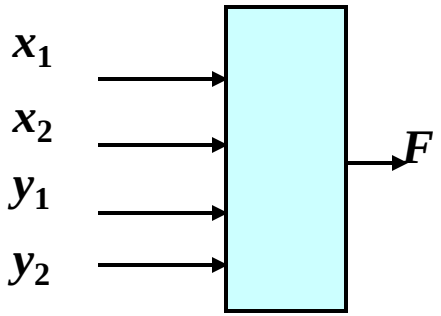
$F = [(A + B + C)' + (A + B' + C')' + (A + C' + D)']'$ (7-19)

$$F = (A'B'C' + A'BC + A'CD')' \qquad (7\text{-}20)$$

多级门电路设计实例

- 设计组合电路，对输入的2个二进制数 $X=X_1X_2$和 $Y=Y_1Y_2$ 比较，当$X>Y$, 输出$F=1$；否则，$F=0$.

① 确定输入输出



② 真值表

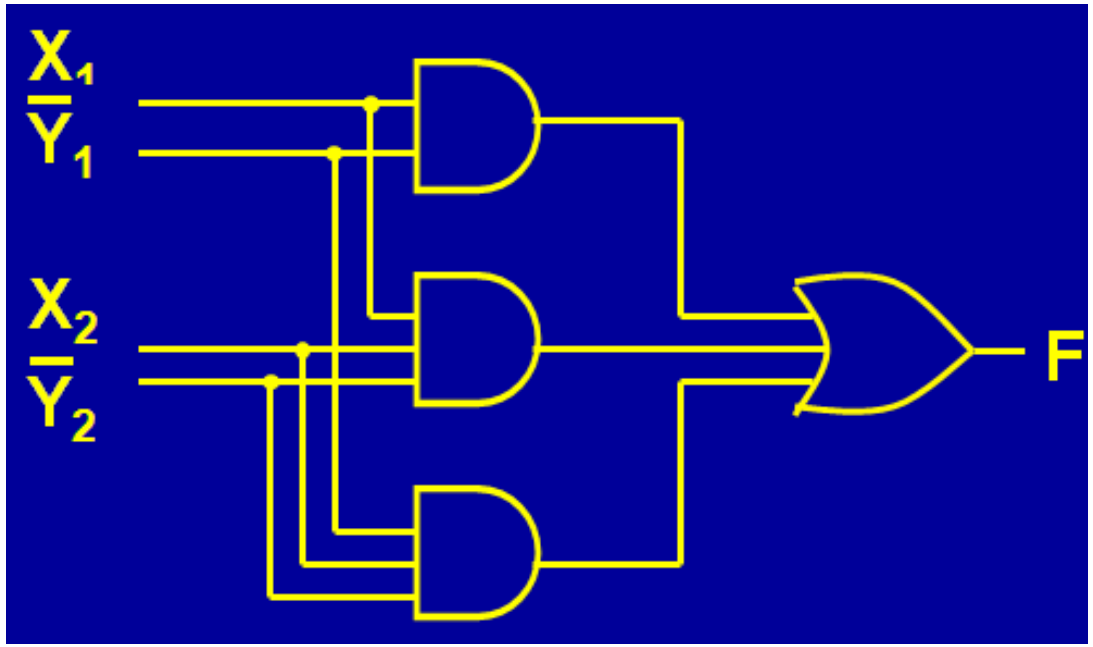
X_1 X_2 Y_1 Y_2	F	X_1 X_2 Y_1 Y_2	F
0 0 0 0	0	1 0 0 0	1
0 0 0 1	0	1 0 0 1	1
0 0 1 0	0	1 0 1 0	0
0 0 1 1	0	1 0 1 1	0
0 1 0 0	1	1 1 0 0	1
0 1 0 1	0	1 1 0 1	1
0 1 1 0	0	1 1 1 0	1
0 1 1 1	0	1 1 1 1	0

多级门电路

③ 最简二级与或电路

$$F = X_1\bar{Y}_1 + X_2\bar{Y}_1\bar{Y}_2 + X_1X_2\bar{Y}_2$$

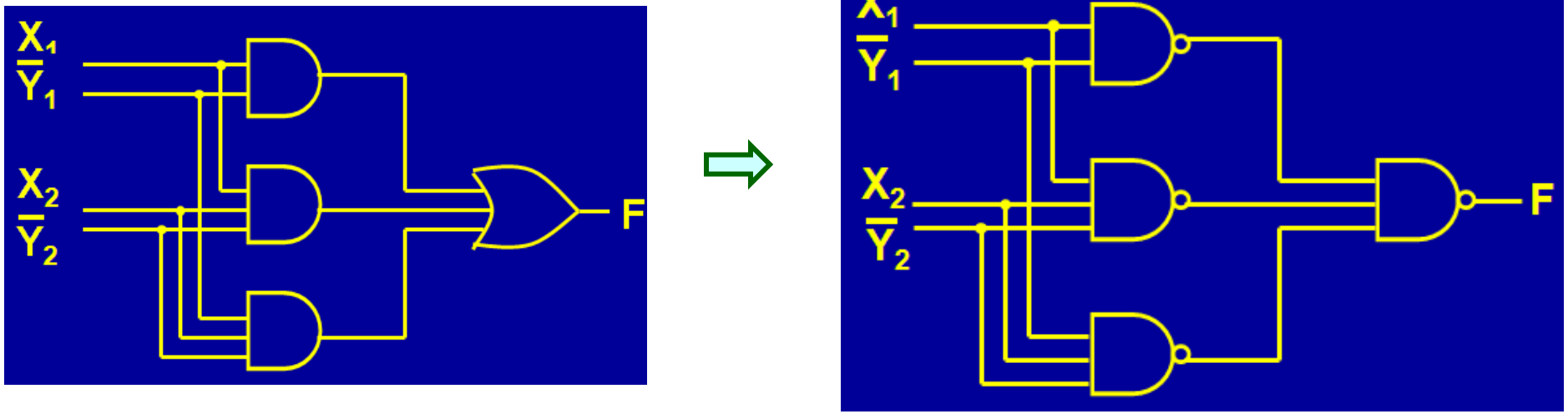
X_1X_2 \ Y_1Y_2	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0



多级门电路

④ 采用单一逻辑门(与非门)设计

$$F = X_1\bar{Y}_1 + X_2\bar{Y}_1\bar{Y}_2 + X_1X_2\bar{Y}_2 = \overline{\overline{(X_1\bar{Y}_1)}\ \overline{(X_2\bar{Y}_1\bar{Y}_2)}\ \overline{(X_1X_2\bar{Y}_2)}}$$



二级门电路的设计

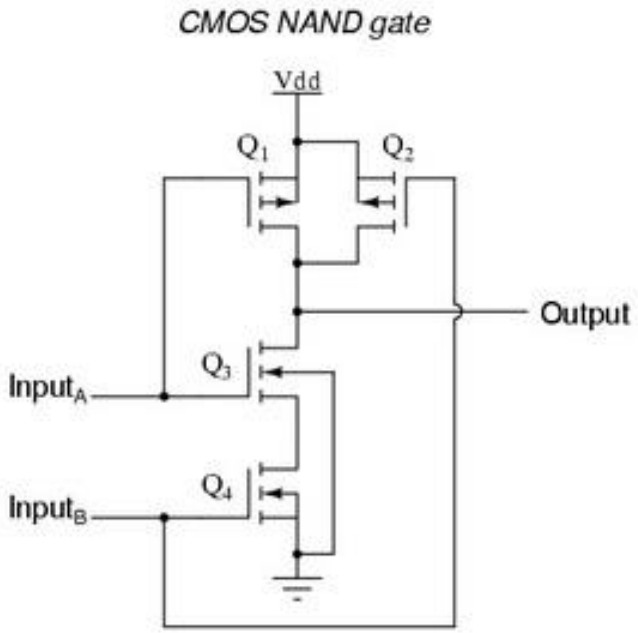
任何逻辑都可以用二级门电路实现

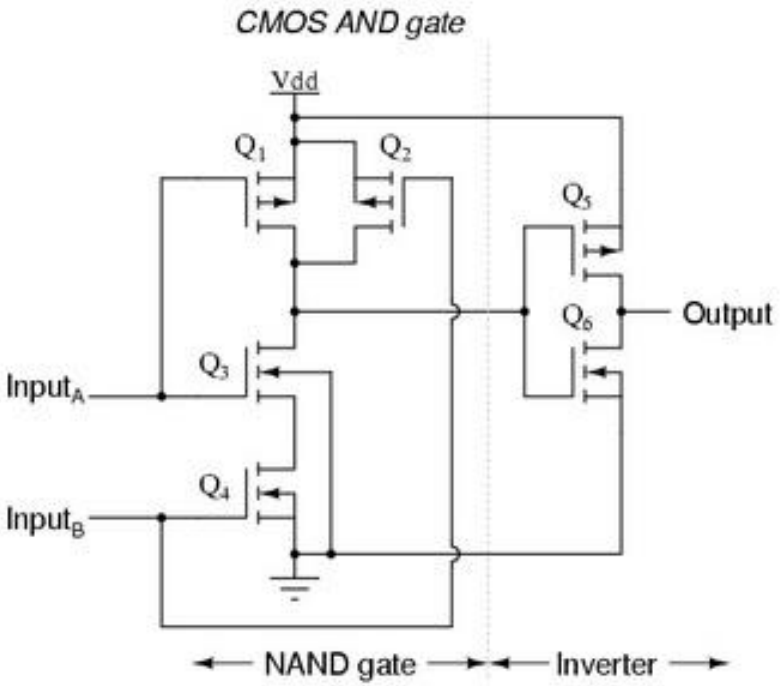
$$F(X,Y,Z)=\sum_{XYZ}(1,6,7)=\prod_{XYZ}(0,2,3,4,5)$$

$$F'(X,Y,Z)=\sum_{XYZ}(0,2,3,4,5)=\prod_{XYZ}(1,6,7)$$

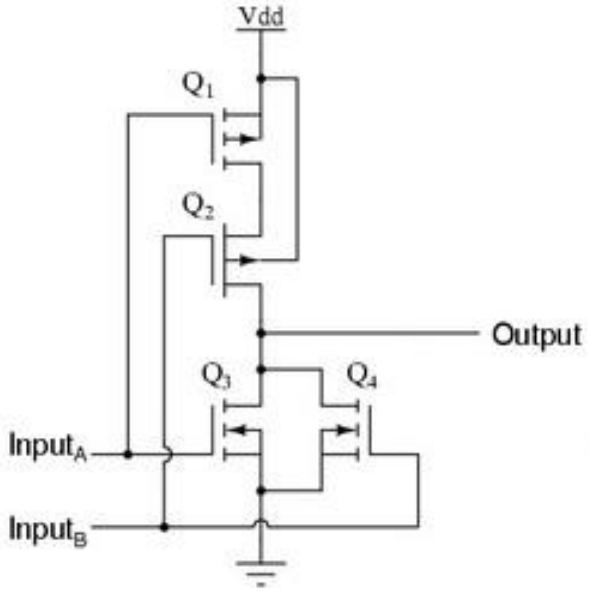
NAND and **NOR** gates:

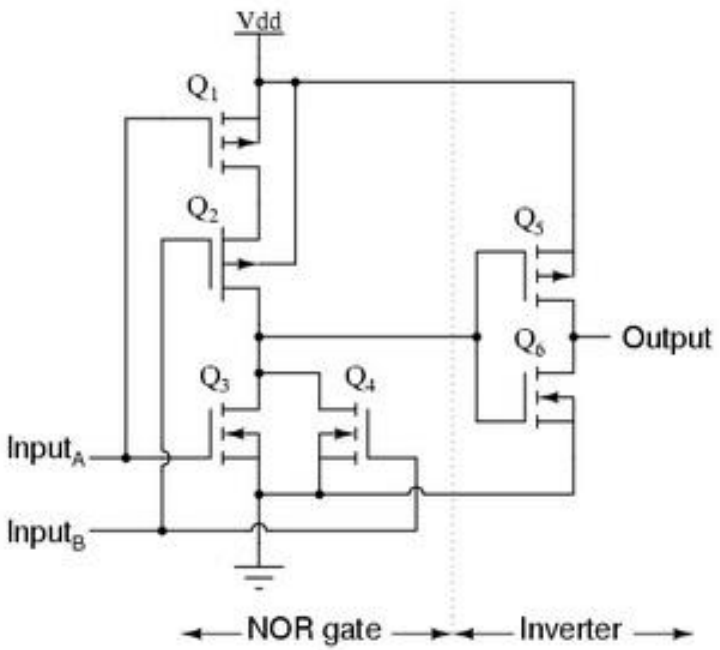
相比与门、或门——速度更快；价格便宜；使用的器件更少





与非门/或非门节省空间和门延迟。



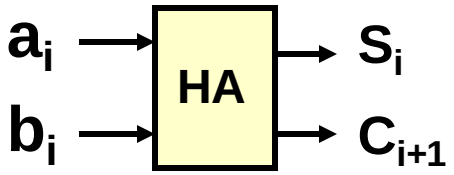


加减法器和OC门

- 半加器
- 全加器
- 多位加法器
- 全减器
- OC门

半加器（Half Adder）

功能：对两个1位二进制数执行相加运算

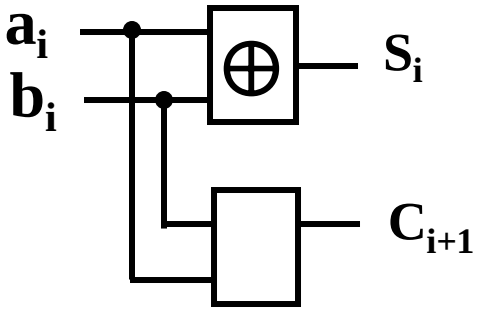


$$S_i = a_i \oplus b_i$$

$$C_{i+1} = a_i b_i$$

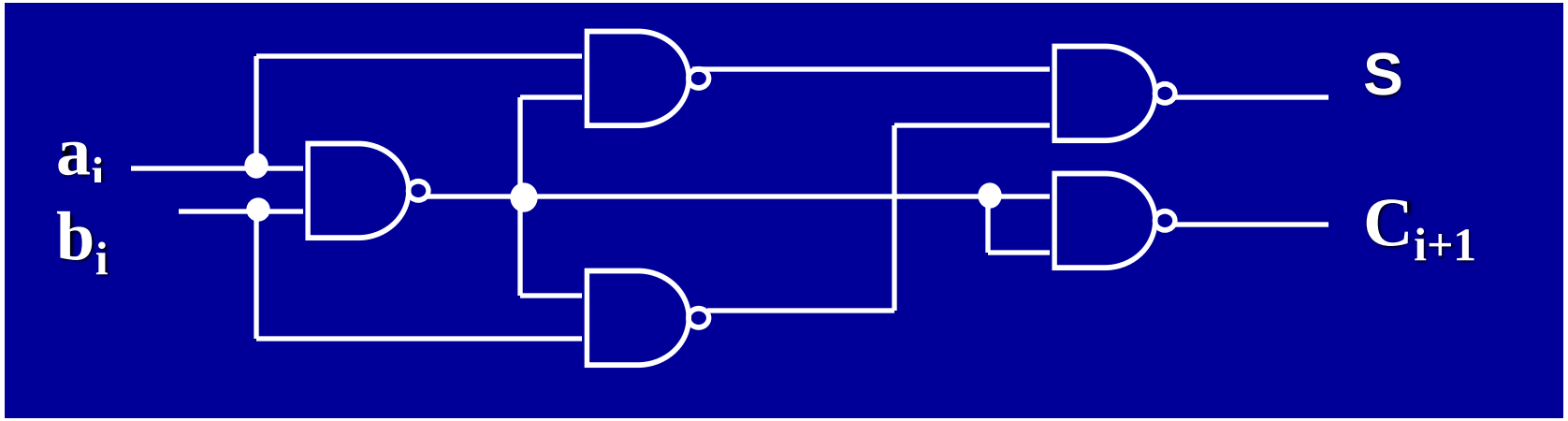
真值表

a_i	b_i	S_i	C_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



利用单一逻辑门与非门实现半加器

$$
\begin{cases}
S_i = \overline{a}_i b_i + a_i \overline{b}_i = \overline{a}_i b_i + a_i \overline{b}_i + a_i \overline{a}_i + b_i \overline{b}_i \\
\quad = a_i (\overline{a}_i + \overline{b}_i) + b_i (\overline{a}_i + \overline{b}_i) = a_i \overline{a_i b_i} + b_i \overline{a_i b_i} \\
\quad = \overline{\overline{a_i \overline{a_i b_i}} \ \overline{b_i \overline{a_i b_i}}} \\
C_{i+1} = \overline{\overline{a_i b_i}}
\end{cases}
$$



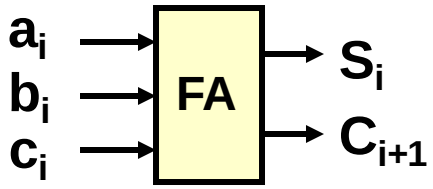
全加器（Full Adder）

$$
\begin{array}{rl}
 & 1\ 0\ 1\ 1 \ldots\ldots\ldots\ A \\
 & 1\ 1\ 1\ 0 \ldots\ldots\ldots\ B \\
+ & \qquad 0 \ \ldots\ldots\ldots\ C_i \\
\hline
 & \qquad\qquad \ldots\ldots\ldots\ S_i
\end{array}
$$

$A = a_3\, a_2\, a_1\, a_0 = 1011$

$B = b_3\, b_2\, b_1\, b_0 = 1110$

全加器表示



a_i	b_i	C_i	S_i	C_{i+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

S_i

a_i \ b_ic_{i-1}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

C_{i+1}

a_i \ b_ic_i	00	01	11	10
0	0	0	1	0
1	0	1	1	1

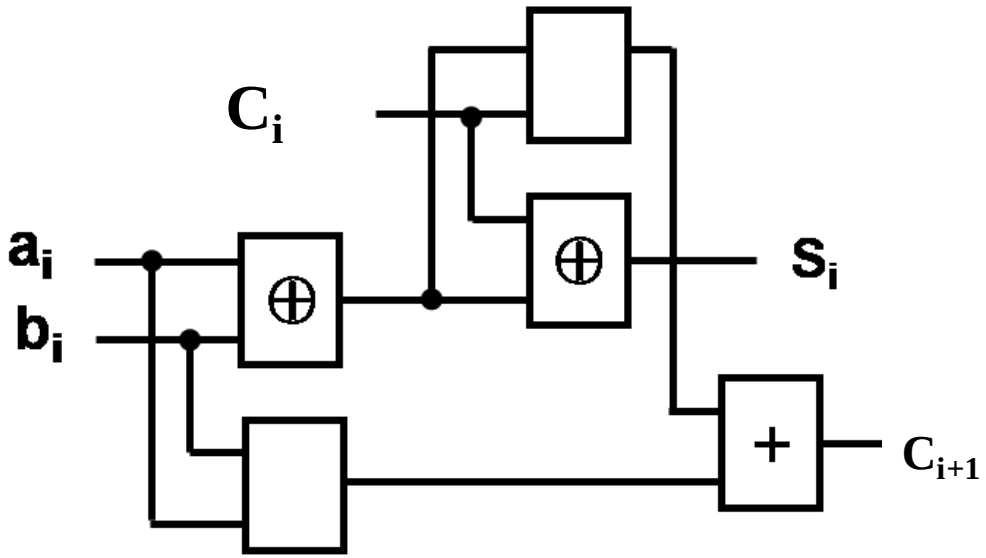
$$S_i = \bar{a}_i\bar{b}_ic_i + a_i\bar{b}_i\bar{c}_i + a_i\bar{b}_i\bar{c}_i + a_i b_i c_i$$
$$=(\bar{a}_i\bar{b}_i + a_i b_i)\, c_i + (a_i\bar{b}_i + a_i\bar{b}_i)\,\bar{c}_i$$
$$=\overline{(a_i \oplus b_i)}\, c_i + (a_i \oplus b_i)\,\bar{c}_i$$
$$= a_i \oplus b_i \oplus C_i$$

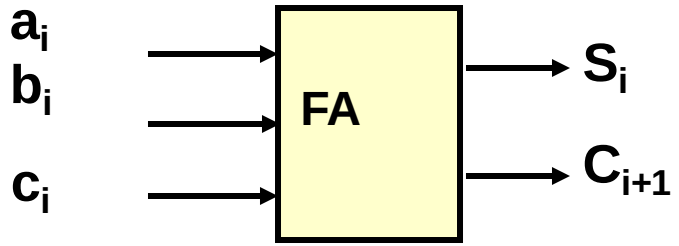
$$C_{i+1}=(a_i \oplus b_i)\, C_i + a_ib_i$$

全加器逻辑表示

- **solution 1：**

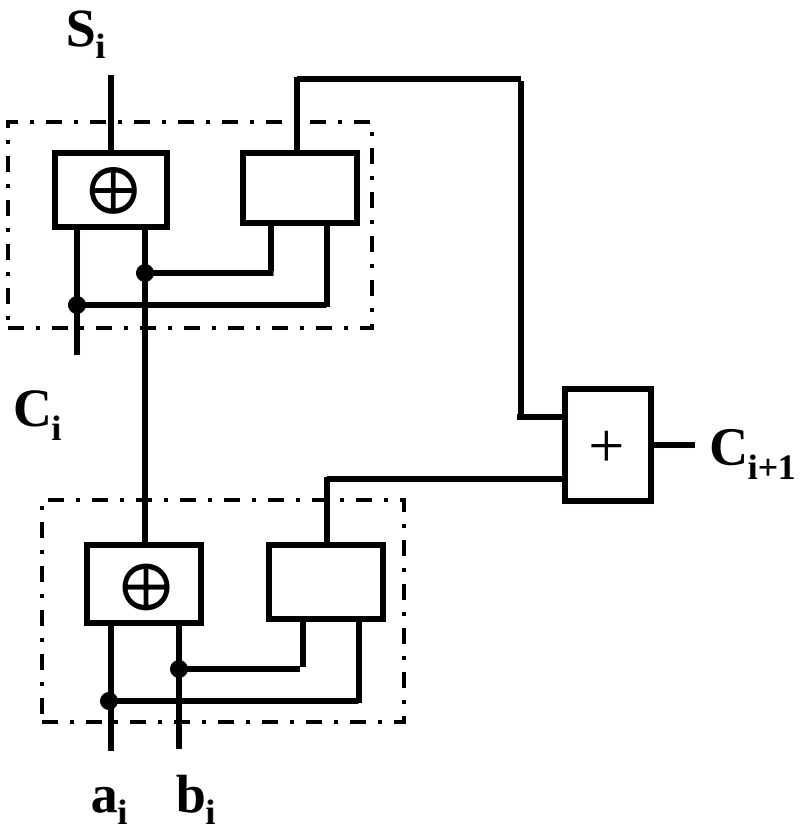
$$\begin{cases} S_i = a_i \oplus b_i \oplus C_i \\ C_{i+1} = (a_i \oplus b_i)\, C_i + a_i b_i \end{cases}$$





全加器逻辑表示（2）

- solution 2



$$S_i = a_i \oplus b_i \oplus C_i$$

$$C_{i+1}=(a_i \oplus b_i)\, C_i + a_i b_i$$

$$\begin{cases} S_i = a_i \oplus b_i \\ C_{i+1}= a_i b_i \end{cases}$$

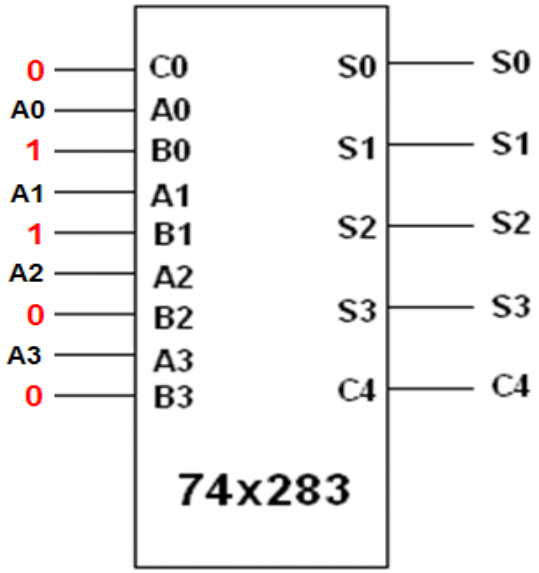
全加器的应用

典型芯片

- **74LS82：** 2-bit adder
- **74LS283：** 4-bit adder

二进制数 $A_3\ A_2\ A_1\ A_0$	余三码 $S_3\ S_2\ S_1\ S_0$	二进制数 $A_3\ A_2\ A_1\ A_0$	余三码 $S_3\ S_2\ S_1\ S_0$
0 0 0 0	0 0 1 1	1 0 0 0	1 0 1 1
0 0 0 1	0 1 0 0	1 0 0 1	1 1 0 0
0 0 1 0	0 1 0 1	1 0 1 0	×
0 0 1 1	0 1 1 0	1 0 1 1	×
0 1 0 0	0 1 1 1	1 1 0 0	×
0 1 0 1	1 0 0 0	1 1 0 1	×
0 1 1 0	1 0 0 1	1 1 1 0	×
0 1 1 1	1 0 1 0	1 1 1 1	×

应用——余3码产生器



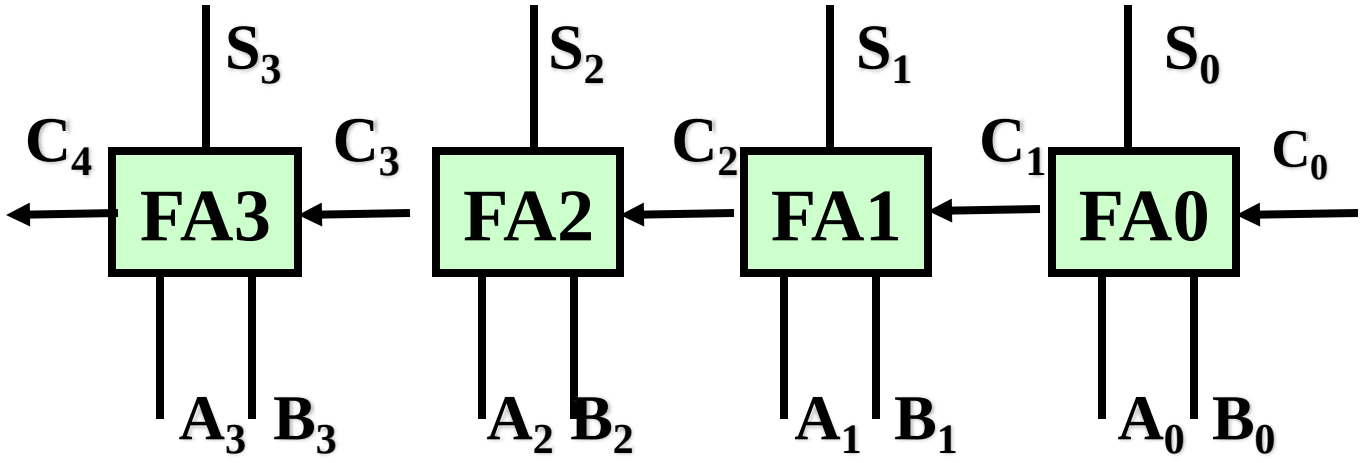
$A_3A_2A_1A_0$: 输入 8421 BCD码

$S_3S_2S_1S_0$：输出余3 码

S = A+0011

4位并行加法器

（1）串行进位



$A = A_3A_2A_1A_0 = 1011$

$B = B_3B_2B_1B_0 = 1110$

$S_i = a_i \oplus b_i \oplus C_i$

$C_{i+1} = (a_i \oplus b_i)\, C_i + a_i b_i$

- 优点：线路简单
- 缺点：串行进位，运算速度慢
- 关键：进位形成时间
- 解决方案：改串行进位为并行进位

4位并行加法器

P83 （2） 超前进位

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

$A= A_3A_2A_1A_0 =1011$

$B= B_3B_2B_1B_0 =1110$

$C_{i+1}=P_iC_i + G_i$

$P_i= A_i \oplus B_i$

$G_i= A_iB_i$

——进位迭代公式

$C_1= P_0C_0+G_0$

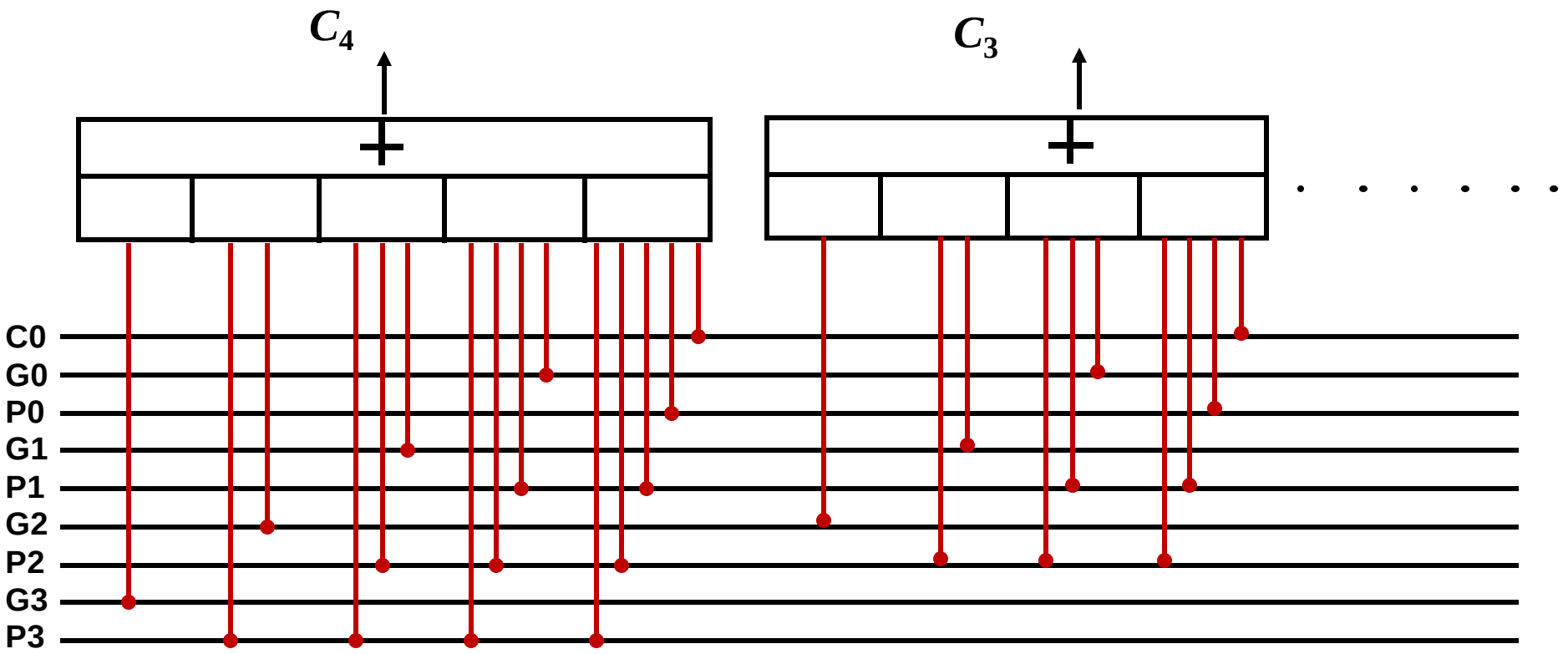
$C_2= P_1C_1+G_1 = P_1P_0C_0+P_1G_0+G_1$

$C_3= P_2C_2+G_2 = P_2P_1P_0C_0+P_2P_1G_0+P_2G_1+G_2$

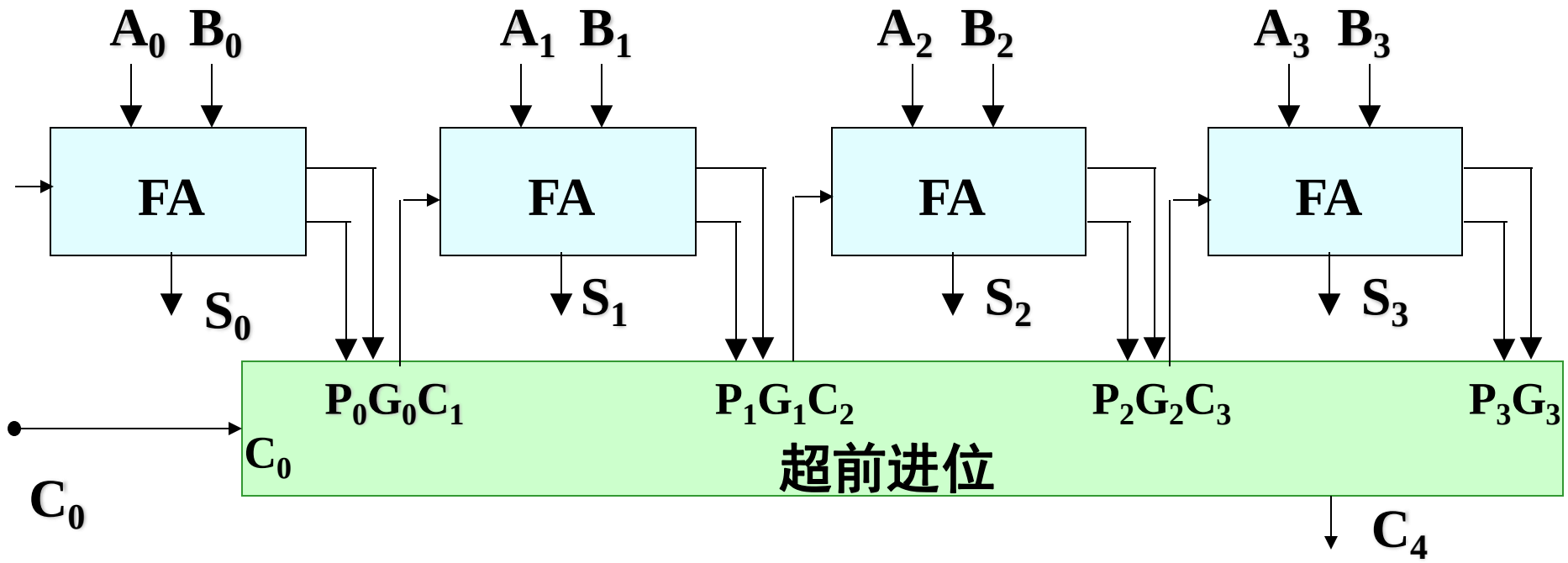
$C_4= P_3C_3+G_3 = P_3P_2P_1P_0C_0+P_3P_2P_1G_0+P_3P_2G_1+P_3G_2+G_3$

4位并行加法器

（2） 超前进位



4位并行加法器



$$P_i = A_i \oplus B_i \qquad G_i = A_i B_i$$

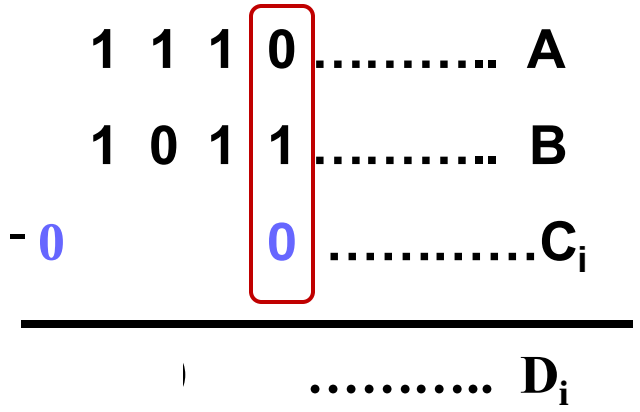
$$C_1 = P_0 C_0 + G_0$$

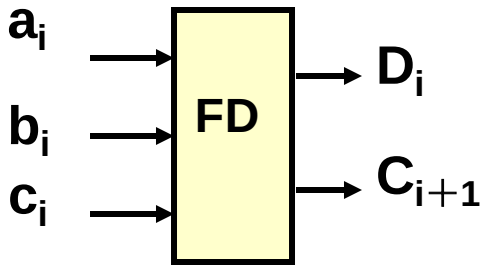
$$C_2 = P_1 C_1 + G_1 = P_1 P_0 C_0 + P_1 G_0 + G_1$$

$$C_3 = P_2 C_2 + G_2 = P_2 P_1 P_0 C_0 + P_2 P_1 G_0 + P_2 G_1 + G_2$$

$$C_4 = P_3 C_3 + G_3 = P_3 P_2 P_1 P_0 C_0 + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3$$

全减器（Binary Full Subtracter）





A= $a_3\,a_2\,a_1\,a_0$ =1110

B = $b_3\,b_2\,b_1\,b_0$=1011

真值表

a_i	b_i	C_i	D_i	C_{i+1}
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

全减器——例

真值表

a_i	b_i	C_i	D_i	C_{i+1}
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

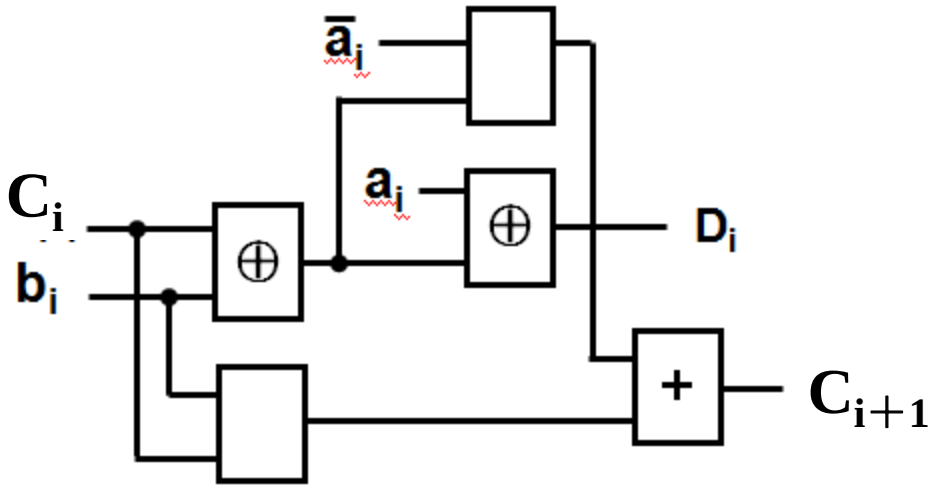
D_i

a_i \ b_ic_i	00	01	11	10
0	0	1	0	1
1	1	0	1	0

C_{i+1}

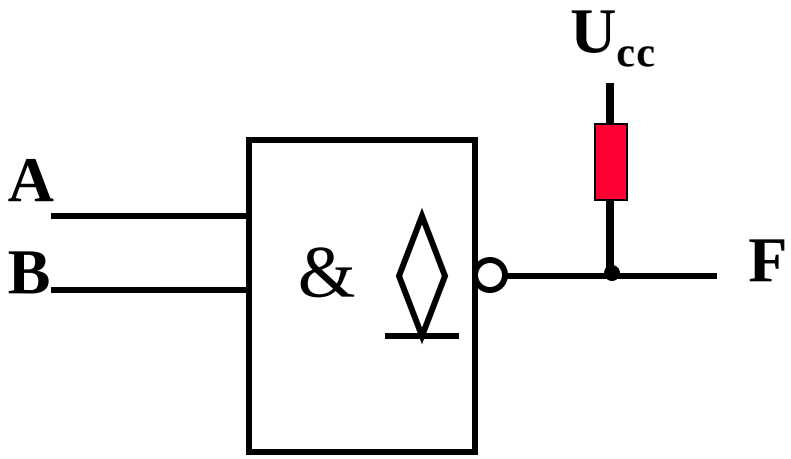
a_i \ b_ic_i	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$$\begin{cases} D_i = a_i \oplus b_i \oplus C_i \\ C_{i+1} = (C_i \oplus b_i)\,\bar{a_i} + C_i b_i \end{cases}$$



OC门（集电极开路门：Open Collector Gate）

$F = \overline{AB}$

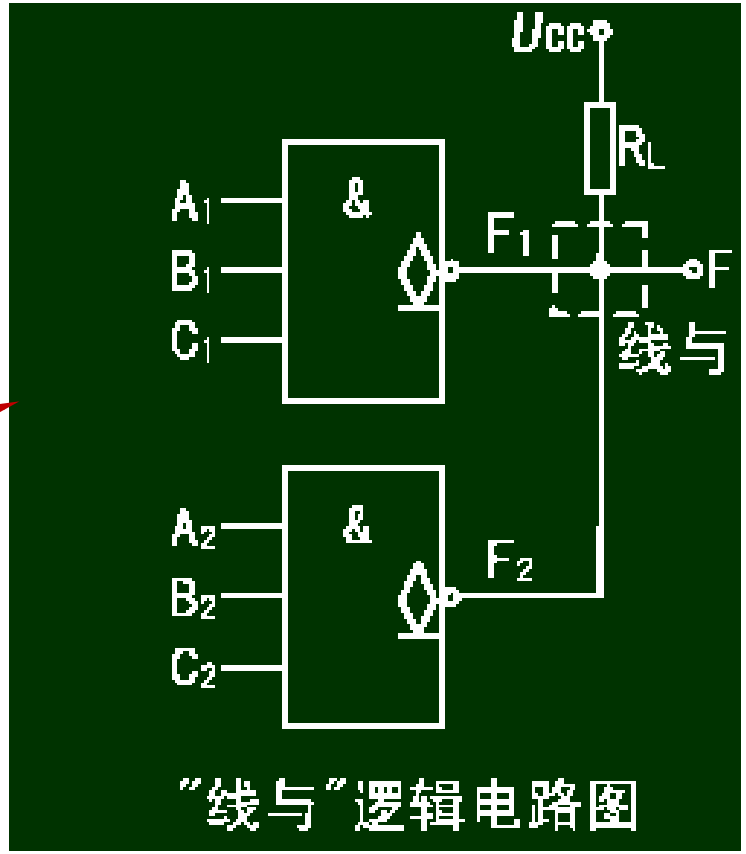


- 几个OC门的输出端可以直接互连：“线与”
- 使用时必须加负载/上拉电阻

OC门&线与

$$F=F_1 \cdot F_2=\overline{A_1B_1C_1} \cdot \overline{A_2B_2C_2}$$

不使用OC门，
需要2个与非
门、1个与门



”线与”逻辑电路图

小　　结

- 半加器
- 全加器
- 多位加法器
- 全减器
- OC门

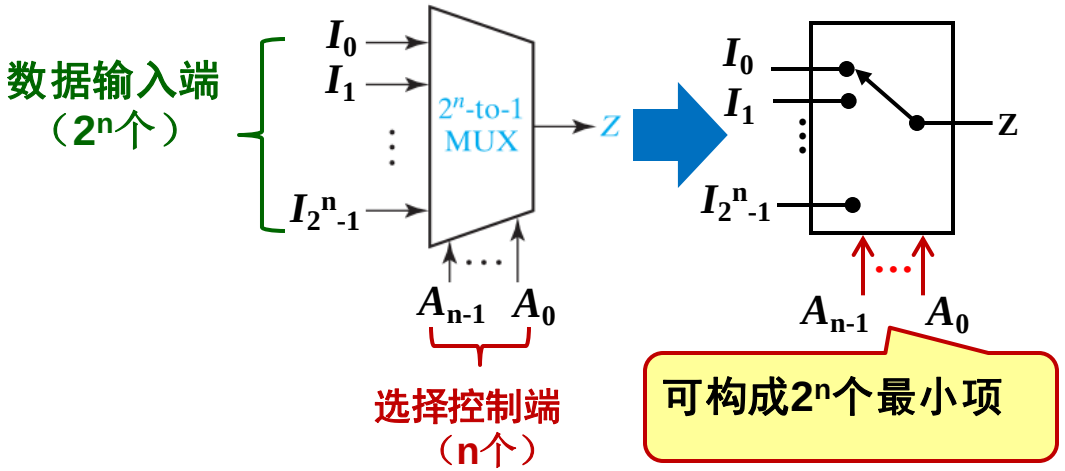
Unit 7 组合逻辑元件

- **多路复用器**(multiplexers)
- **三态器件**(Three-state Buffer)
- **译码器(**Decoders)
- **编码器(**Encoders)
- **奇偶校验器**
- **比较器**
- **只读存储器**(ROM)
- **利用MSI设计组合逻辑电路**

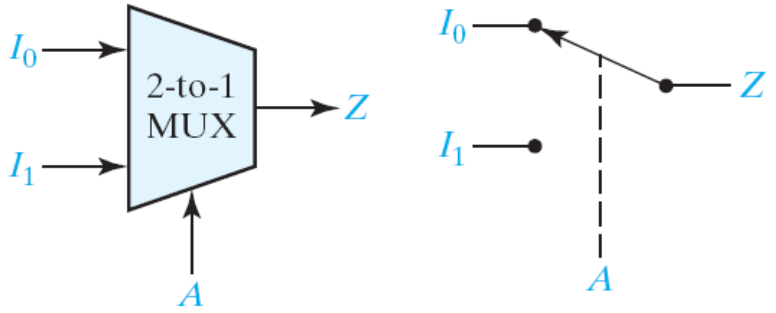
Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

数据选择器/多路开关



2选1数据选择器



$$Z = A' I_0 + A I_1$$

$$Z = \sum_{k=0}^{2^n-1} m_k I_k$$

控制端最小项m_k的序号K，指向了第K路数据输入端I_k。

m_k —— n 个控制变量的最小项

I_k ——第 k 路数据输入

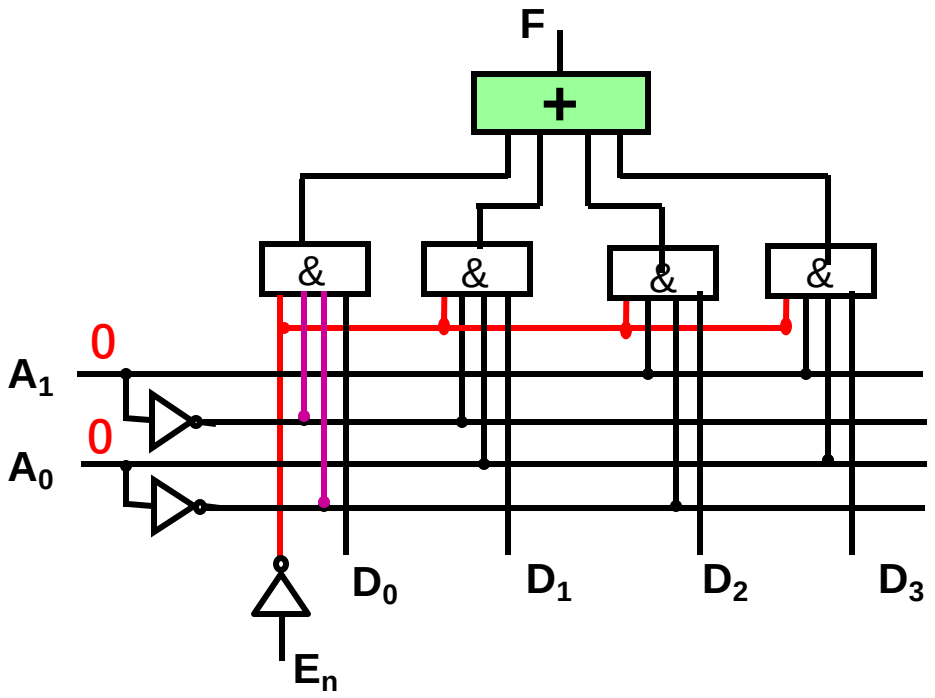
数据选择器的功能：

① 从多路输入中选择一个送往输出端（2ⁿ选1）；

② 选择哪一路输入送到输出端由控制信号决定；

用途：实现多通道的数据传送；

4选1数据选择器

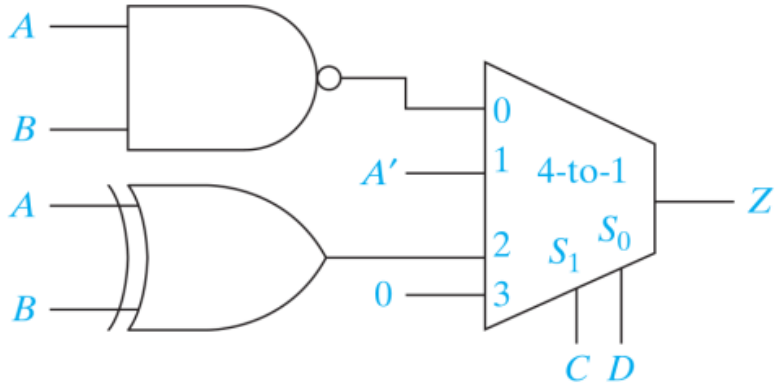


E_n	A_1	A_0	F
1	×	×	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

功能表

$$F=\overline{E}_n(D_0\overline{A}_1\overline{A}_0+D_1\overline{A}_1A_0+D_2A_1\overline{A}_0+D_3A_1A_0)$$

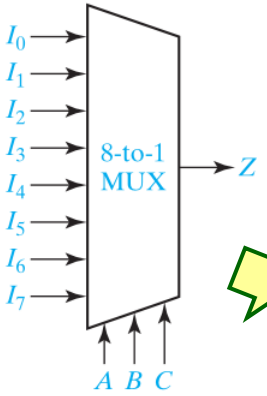
- 典型应用——实现常规逻辑函数



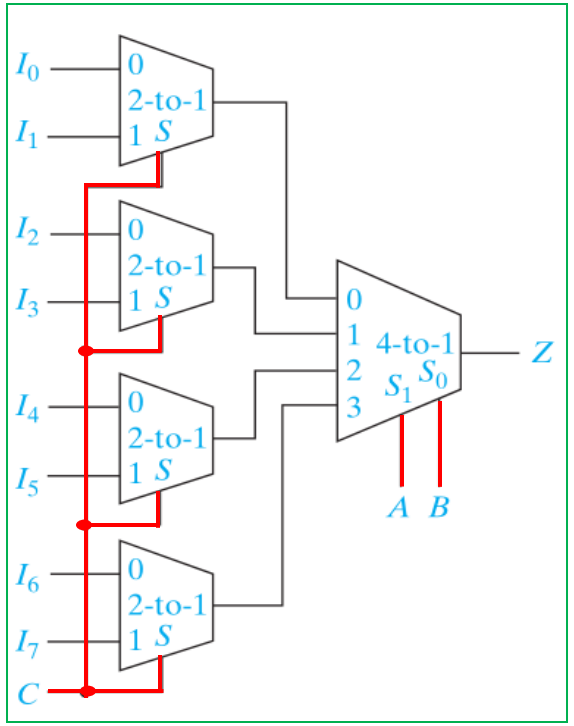
$$Z=\overline{C}\,\overline{D}(\overline{A}+\overline{B})+\overline{C}D\overline{A}+C\overline{D}(A\overline{B}+\overline{A}B)+CD(0)$$

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

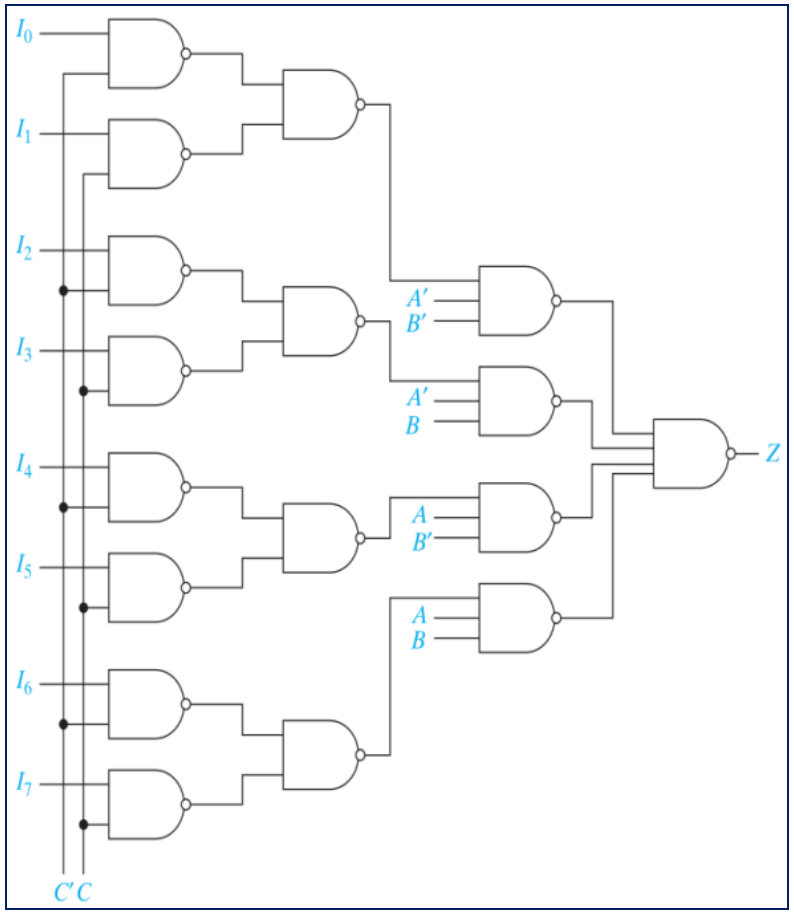
8选1数据选择器



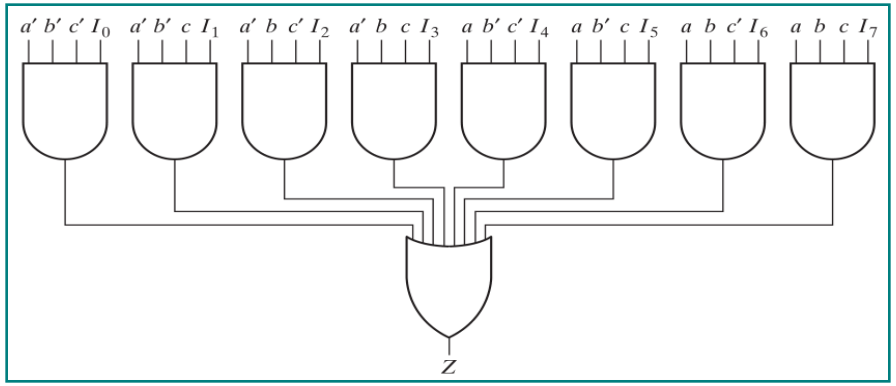
数据选择器级联实现



单一逻辑门实现



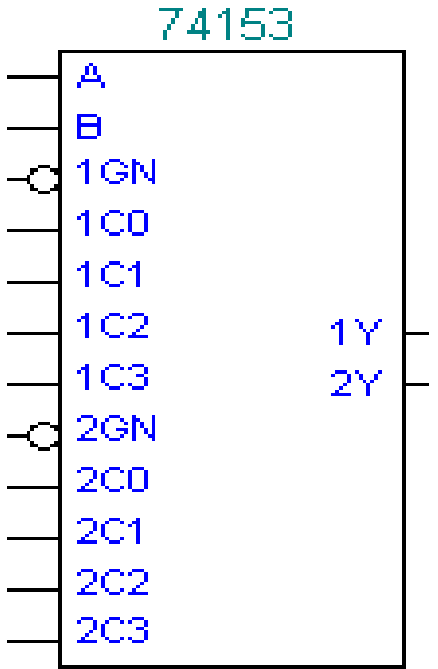
二级电路实现

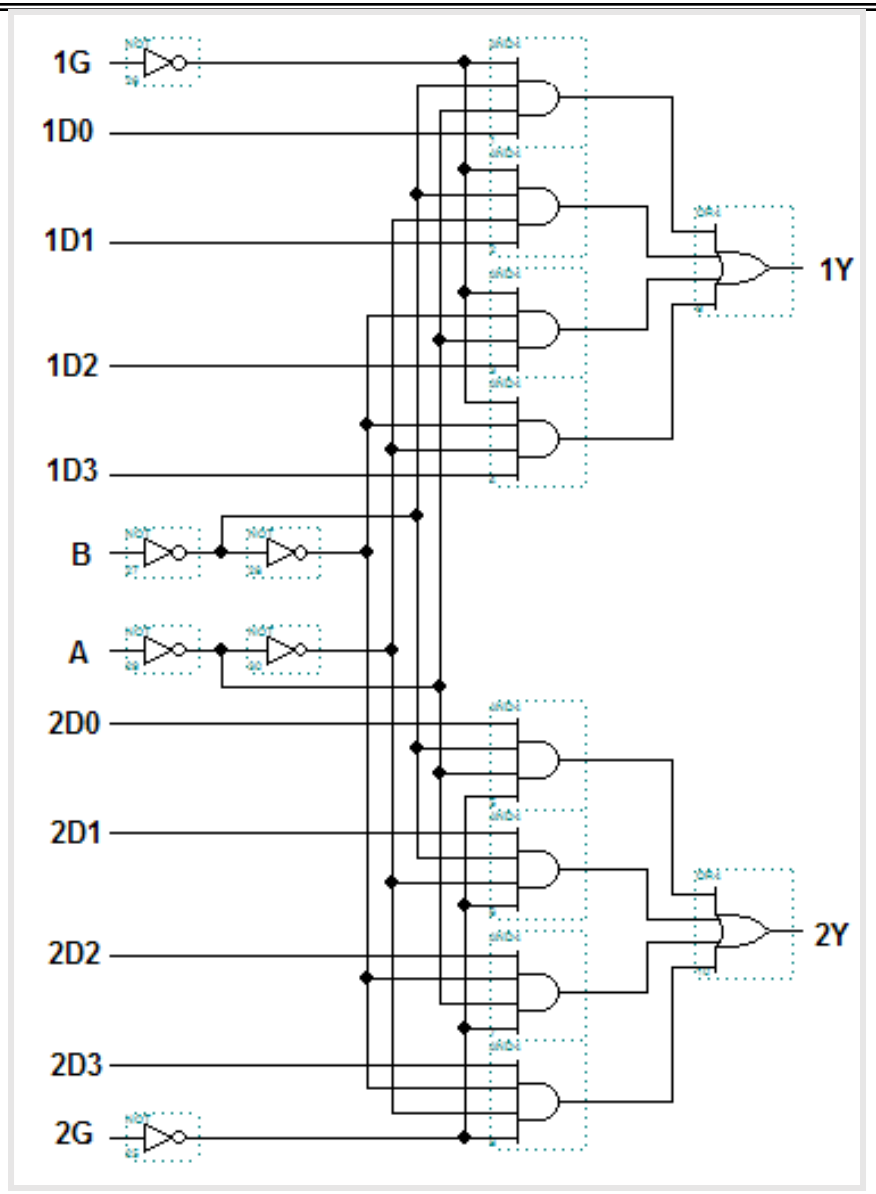


$$Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 + AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

双4选1典型器件74LS153

1Gn	2Gn	A	B	1Y	2Y
1	1	×	×	0	0
0	0	0	0	$1C_0$	$2C_0$
0	0	0	1	$1C_1$	$2C_1$
0	0	1	0	$1C_2$	$2C_2$
0	0	1	1	$1C_3$	$2C_3$





Unit 7 组合逻辑元件

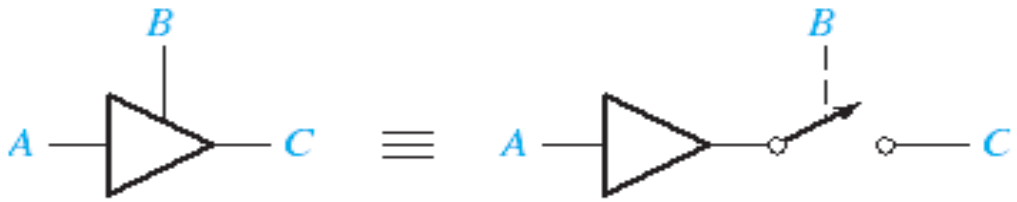
- **多路复用器**(multiplexers)
- **三态器件(Three-state Buffer)**
- **译码器(**Decoders)
- **编码器(**Encoders)
- **奇偶校验器**
- **比较器**
- **只读存储器**(ROM)
- **利用MSI设计组合逻辑电路**

三态门(Three-State Buffers)

三态——

- **0**
- **1**
- **Z: 高阻态**

- 包括三态恒等门、三态非门、三态与非门等，**缓冲器**（驱动门）。
- 用途之一:可用来增强输出驱动能力



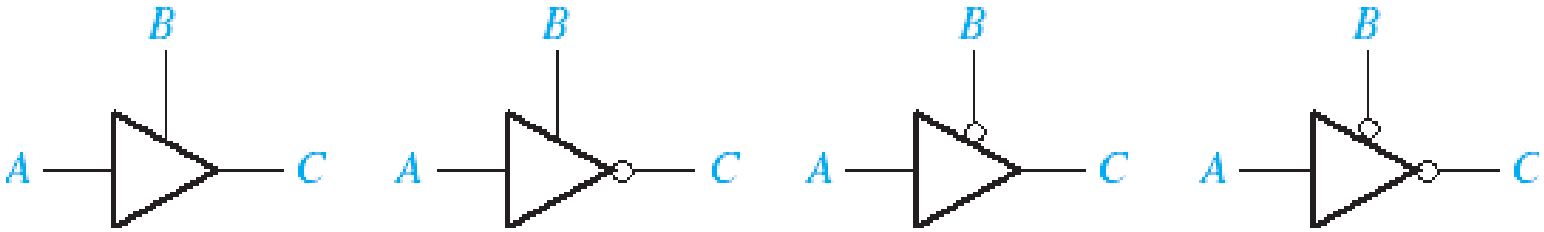
三态门（恒等）

B：使能端，高电平有效

真值表

B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

三态门(Three-State Buffers)



B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

(a)

B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

(b)

B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

(c)

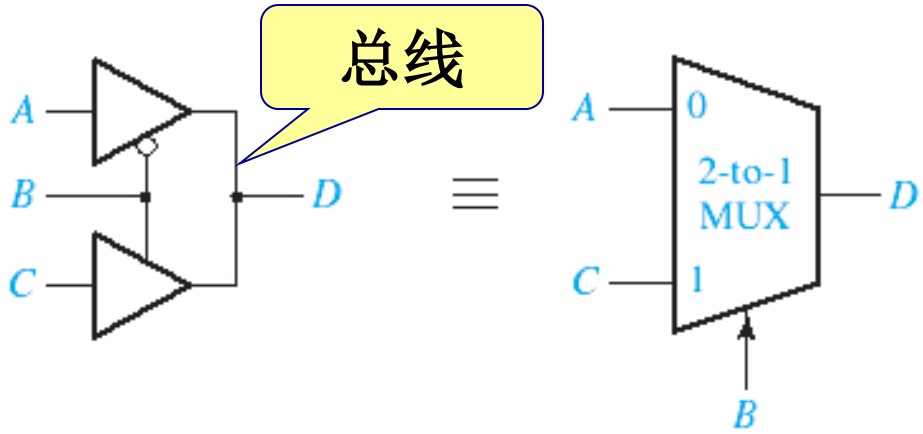
B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

(d)

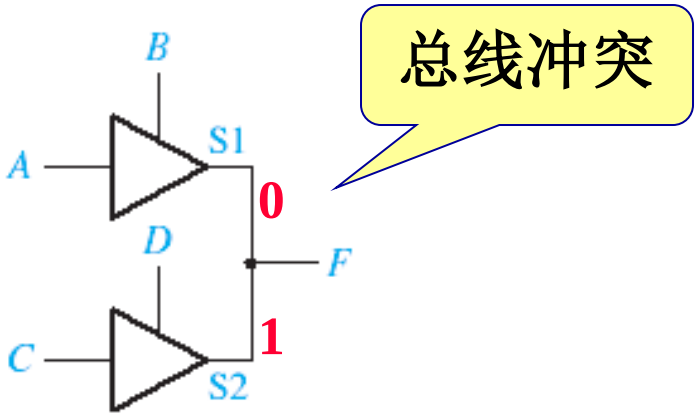
高阻态：电阻很大，相当于开路

高阻态相当于该门同与它连接的电路处于断开的状态。（实际电路中不可能去断开它）

三态门



D=B'A+BC



S_1 \ S_2	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

三态门

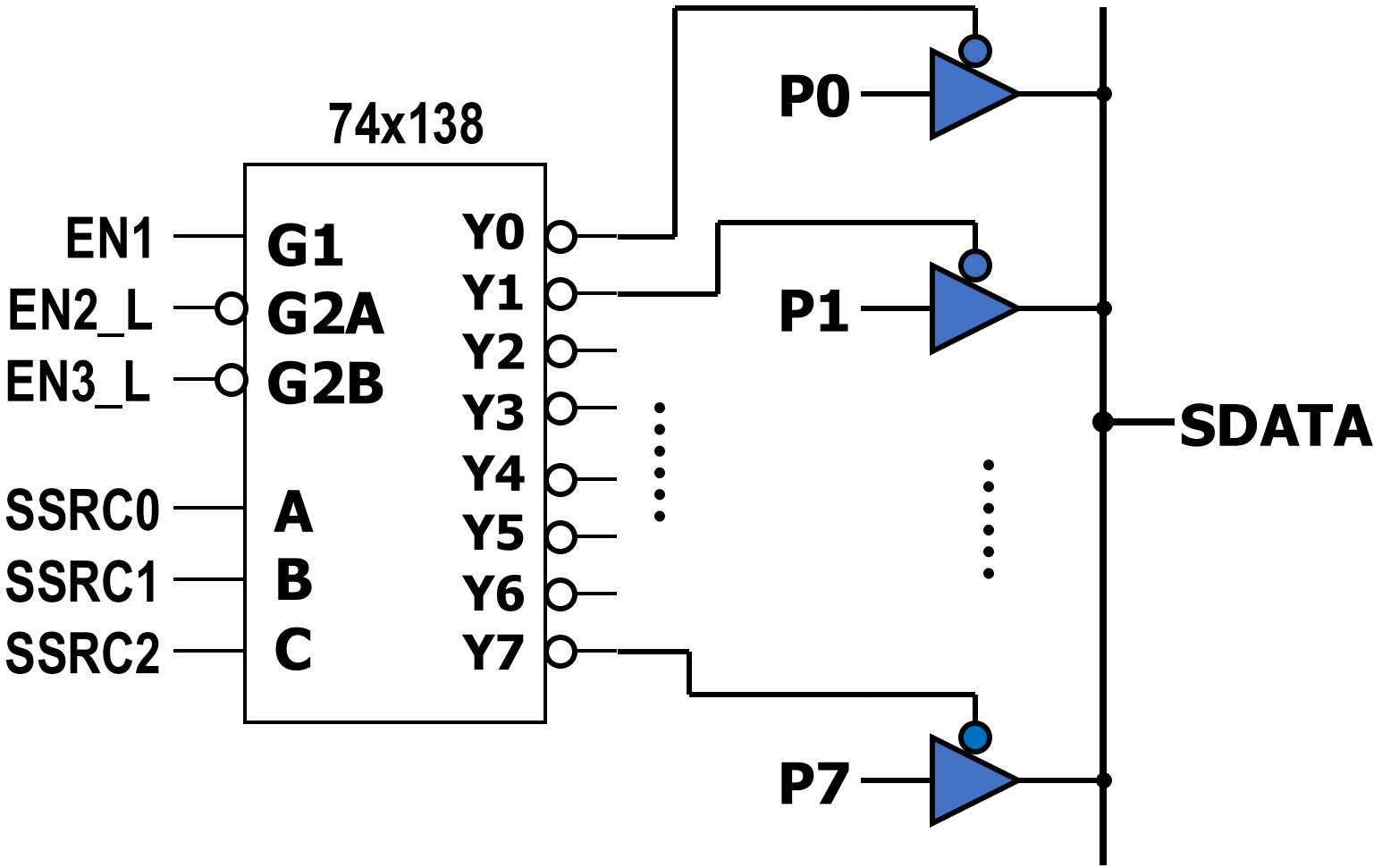
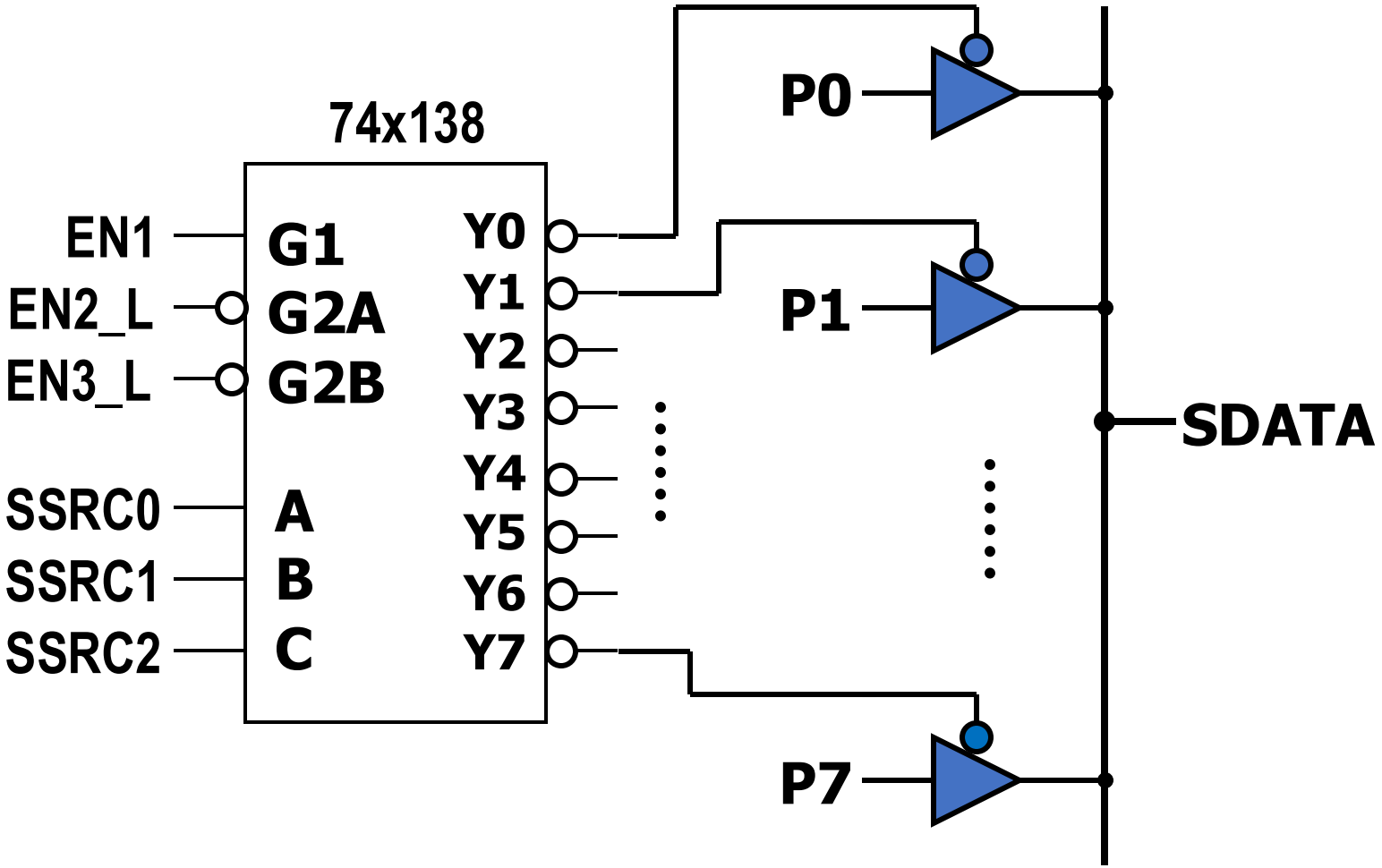


图 8个信号源共享1根三态总线/同线

三态器件允许多个信号源共享单个“总线”（同线），但线上每次仅一个器件“谈话”

假如不是全部EN线有效，则没有一个三态缓冲器能被使能，此时SDATA上的逻辑值是“未定义”，悬空信号的实际电压值依赖于电路细节。

三态门



对典型的三态器件，进入高阻态比离开高阻态快。也会使得系统中产生冲突（fighting）

图 8个信号源共享1根三态同线

三态门

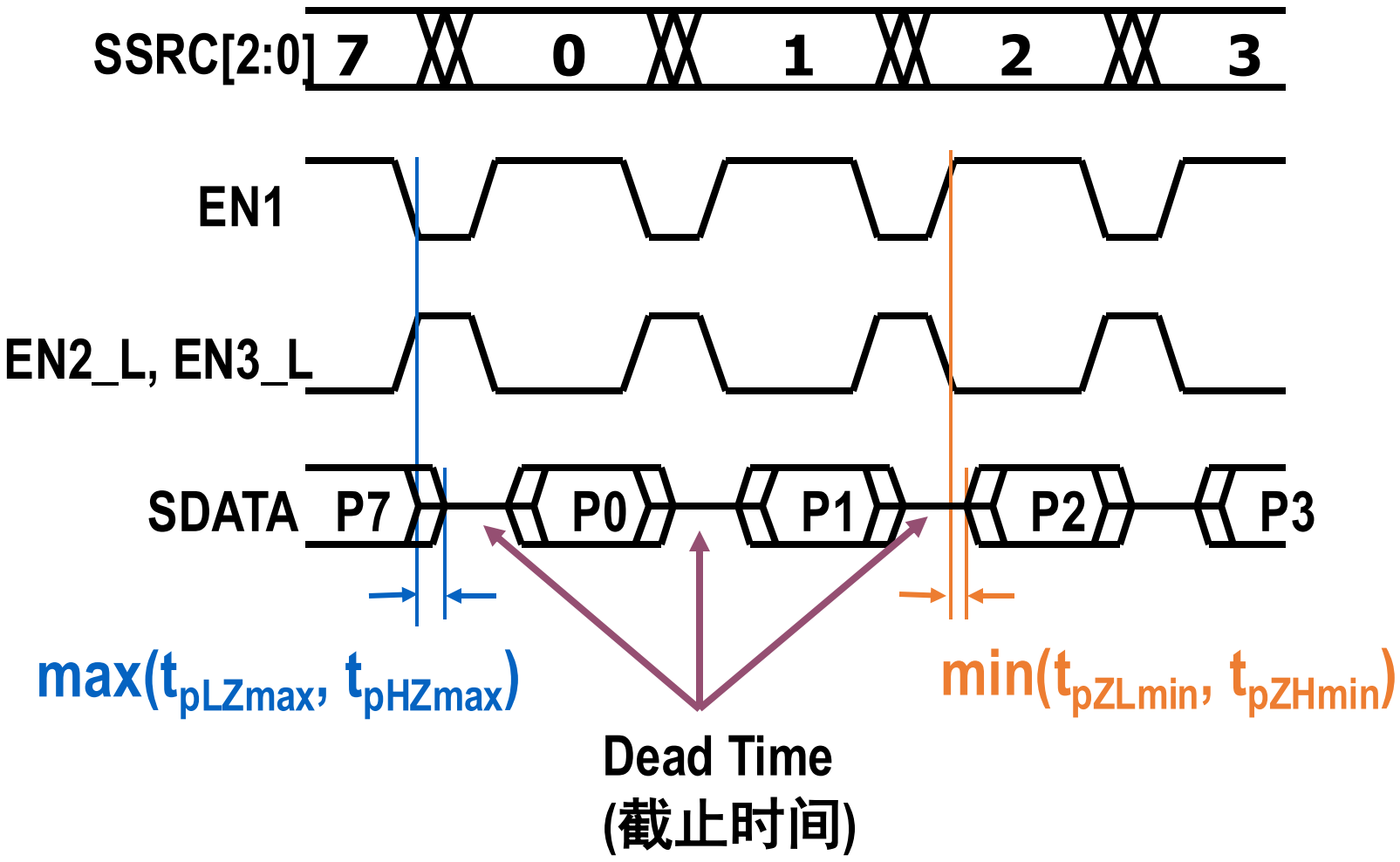
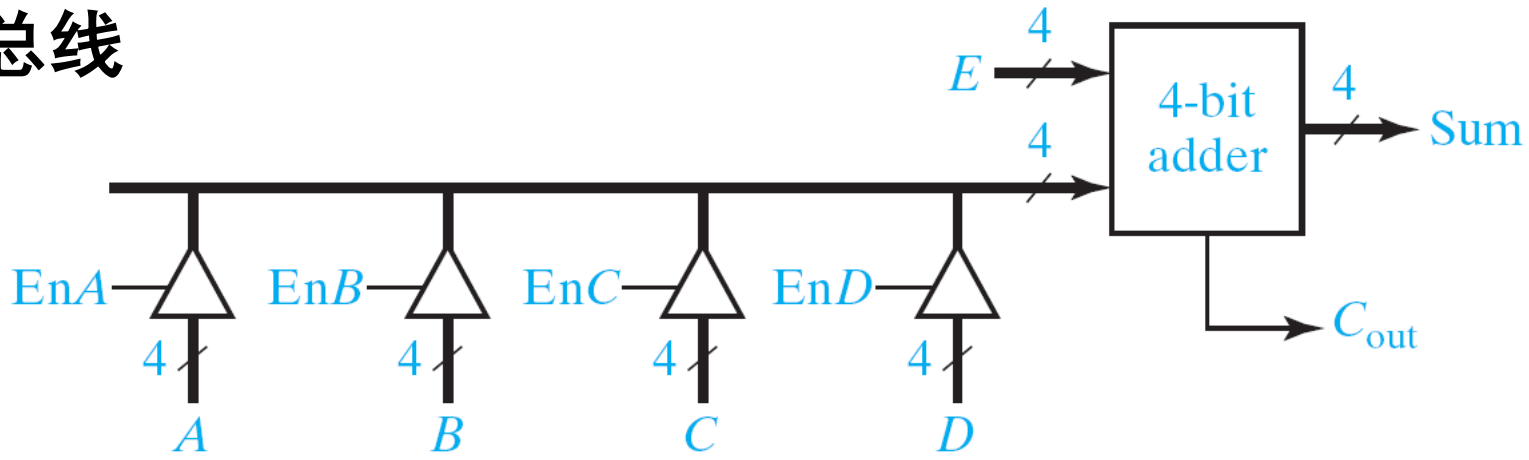


图 三态同线定时图

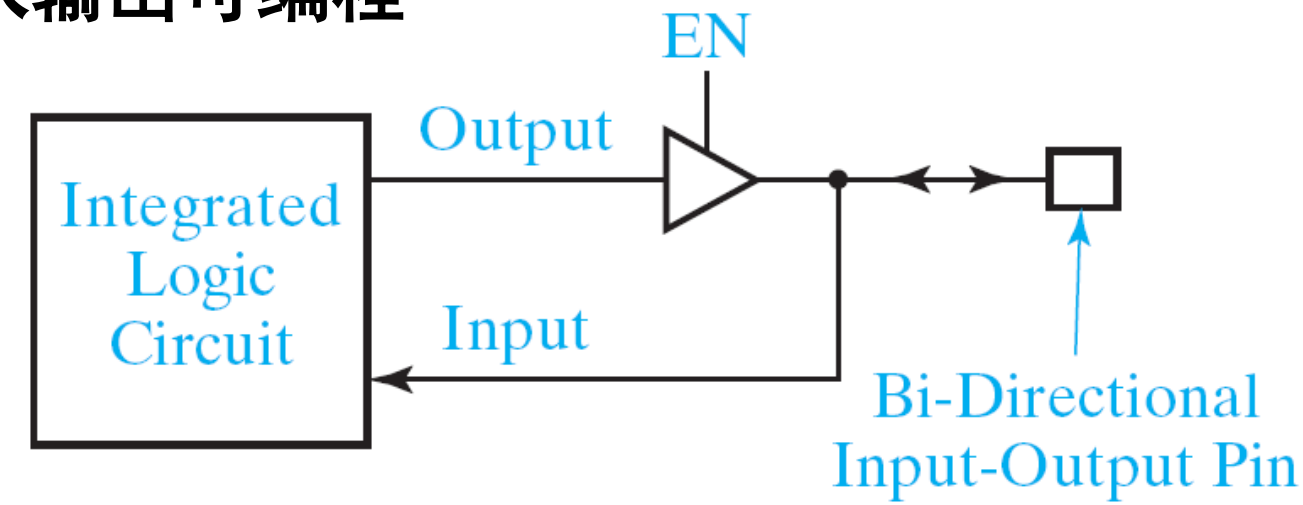
使用三态器件唯一真正安全的方法是**设计逻辑控制**，以保证同线上有一段**截止时间**（dead time），在此期间不应有任何器件驱动同线。

三态门应用

- 三态总线

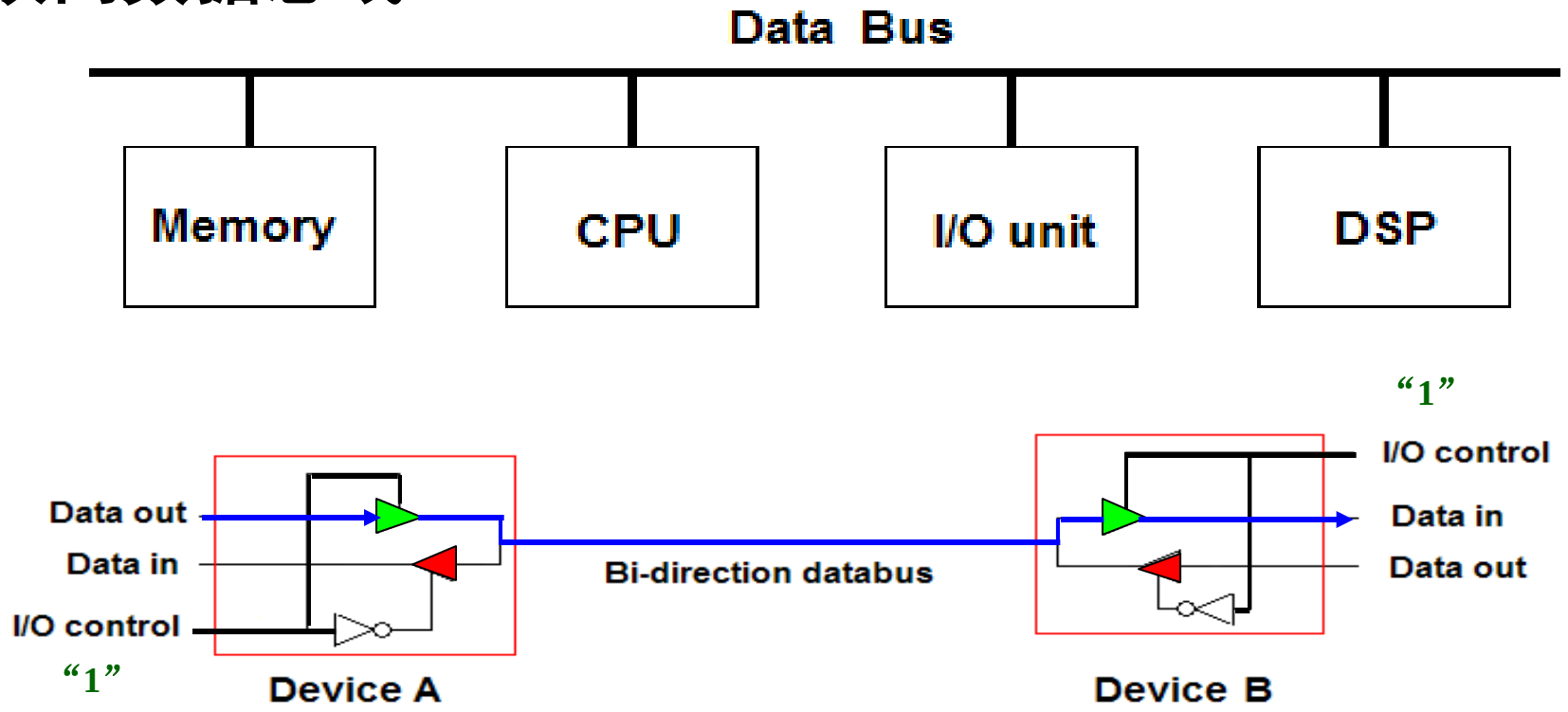


- 管脚输入输出可编程



三态门应用——续

- 双向数据总线



三态门应用——续

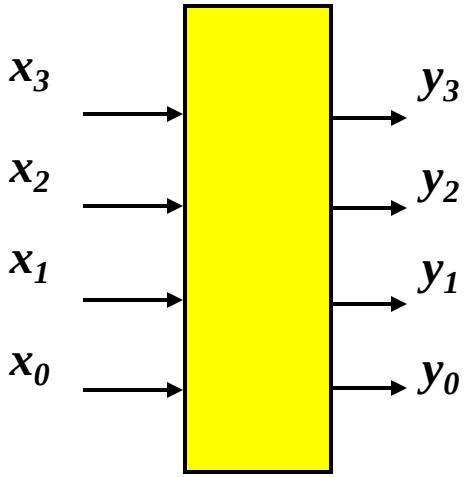
内存里的一个存储单元

- 读写控制线处于低电位时，可以写入；
- 读写控制线处于高电位时，可以读出
- 但是不读不写，就要用高阻态

三态门的应用

- $X=X_3X_2X_1X_0$为8421BCD码，设计一个MOD 5选择电路，要求选择那些能被5整除的数输出。

①真值表（ F为控制信号）



X_3 X_2 X_1 X_0	F	X_3 X_2 X_1 X_0	F
0 0 0 0	1	1 0 0 0	0
0 0 0 1	0	1 0 0 1	0
0 0 1 0	0	1 0 1 0	×
0 0 1 1	0	1 0 1 1	×
0 1 0 0	0	1 1 0 0	×
0 1 0 1	1	1 1 0 1	×
0 1 1 0	0	1 1 1 0	×
0 1 1 1	0	1 1 1 1	×

三态门的应用——续

② 化简

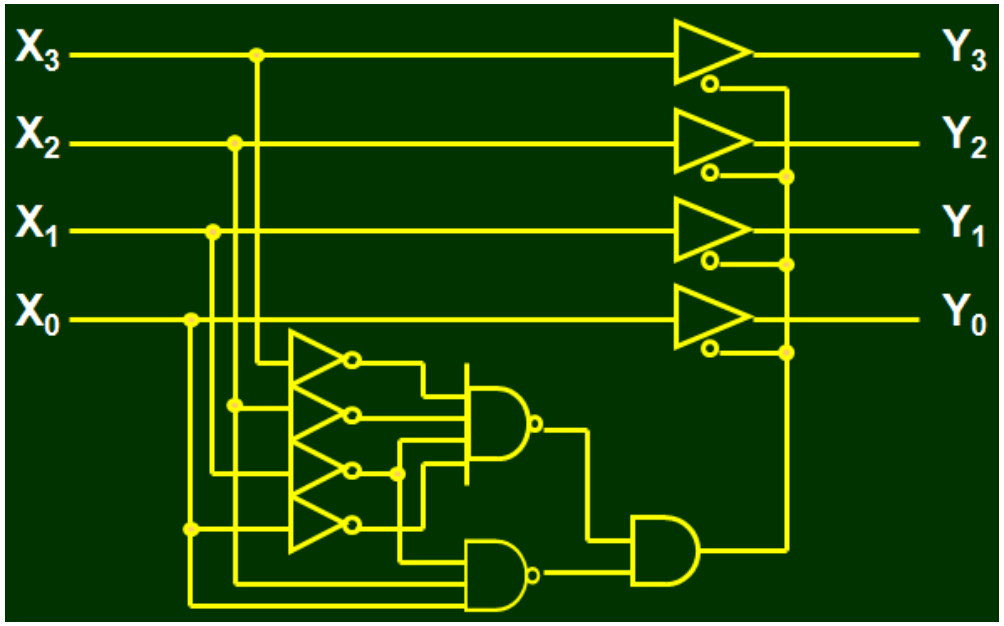
x_3x_2 \ x_1x_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	×	×	×	×
10	0	0	×	×

$$\overline{F} = (\overline{X_2\bar{X}_1X_0})\ (\overline{\bar{X}_3\bar{X}_2\bar{X}_1\bar{X}_0})$$

$$F=\overline{X_2\bar{X}_1X_0+\bar{X}_3\bar{X}_2\bar{X}_1\bar{X}_0}$$

$$= \overline{(X_2\bar{X}_1X_0)}\ \overline{(\bar{X}_3\bar{X}_2\bar{X}_1\bar{X}_0)}$$

③ 逻辑图

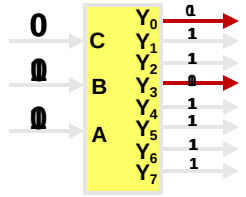
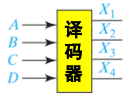
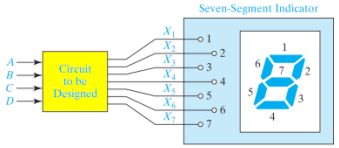


Unit 7 组合逻辑元件

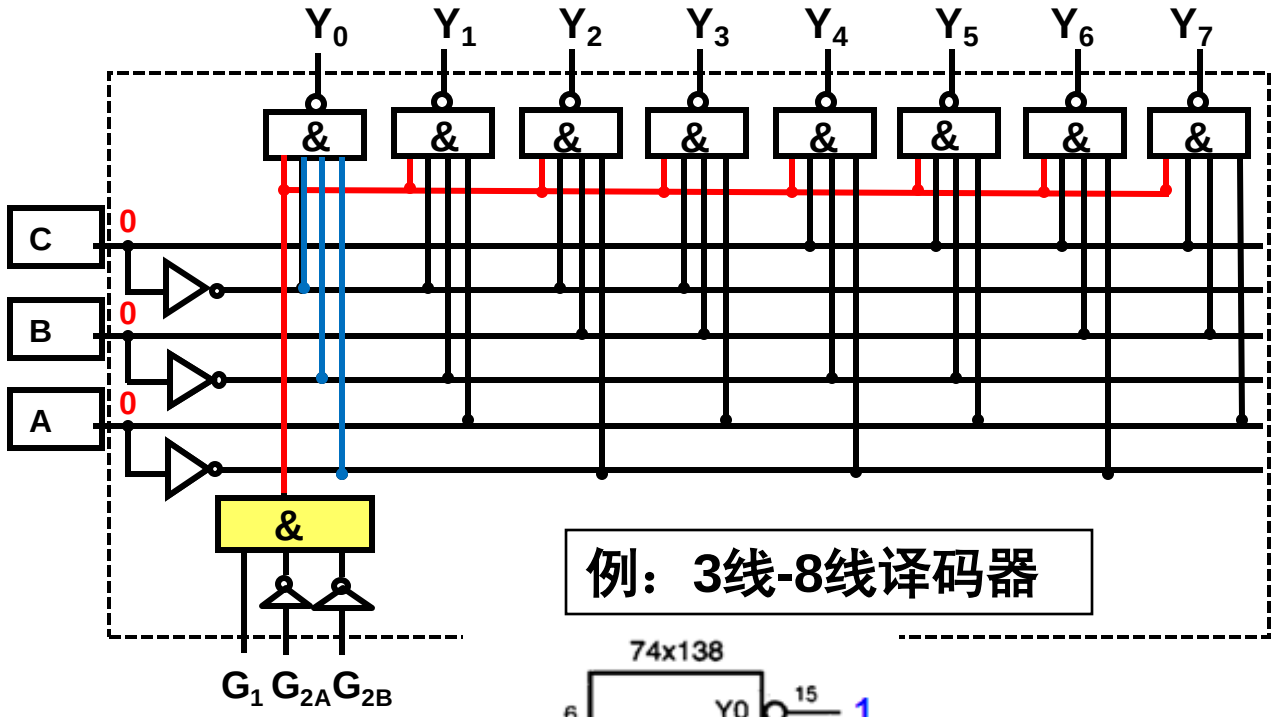
- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

译码器及分类

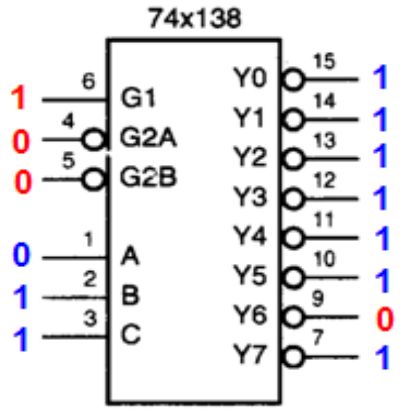
- 特点：多输入、多输出的组合逻辑电路
- 功能：将一种编码转换为另一种编码

分类	特点	译码演示
二进制译码器	• n 位二进制码 • N位（$N=2^n$），每根输出线都与一个输入最小项唯一对应（输出线编号值=最小项编号值） • 每个最小项输入，只能使 N 根输出线中的一个输出有效 ⟶ N（$N=2^n$）中取一译码器，也称最小项译码器。	 （3 线-8 线译码器）
代码转换译码器	从一种编码转换为另一种编码 （例如：8421BCD码→余3码）	
显示译码器	将输入的编码信号转换为十进制码或其它特定编码，用来驱动显示器件显示相应的文字符号。	

二进制译码器举例——3线-8线译码器



例：3线-8线译码器



典型芯片：74LS138

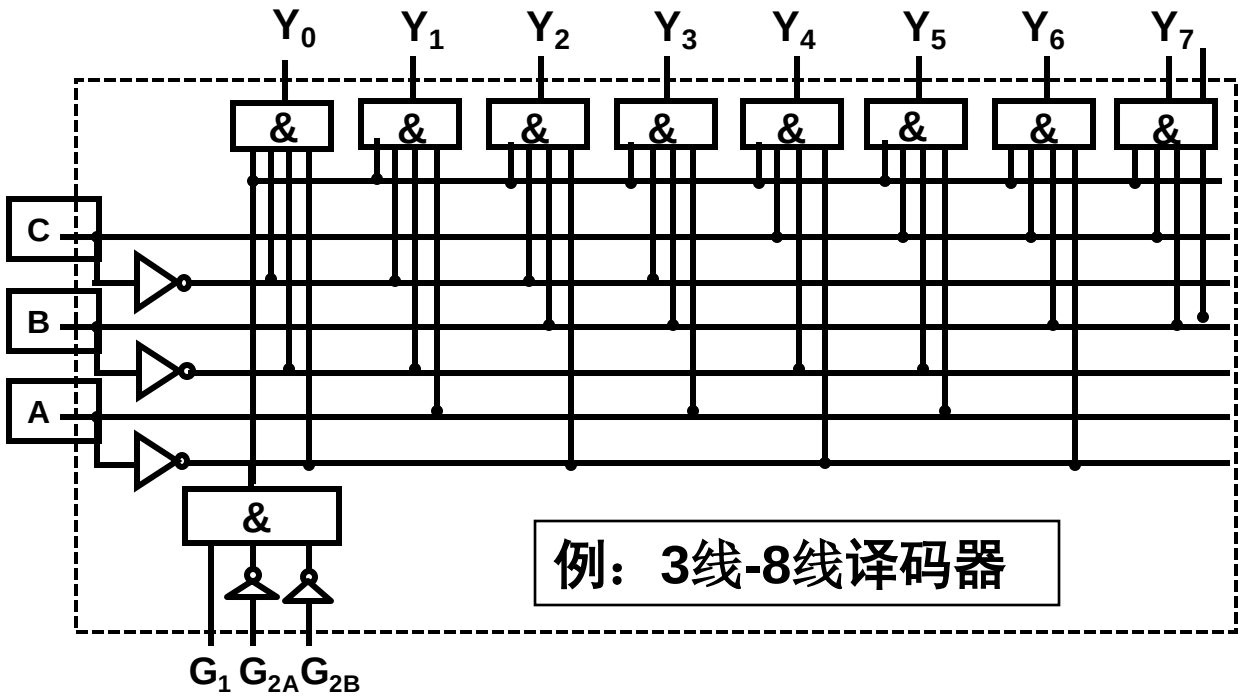
使能端			输入			译码输出							
G_1	G_{2A}	G_{2B}	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

译码器输出：低电平有效

$y_i=\bar{m}_i=M_i$

3线–8线译码器

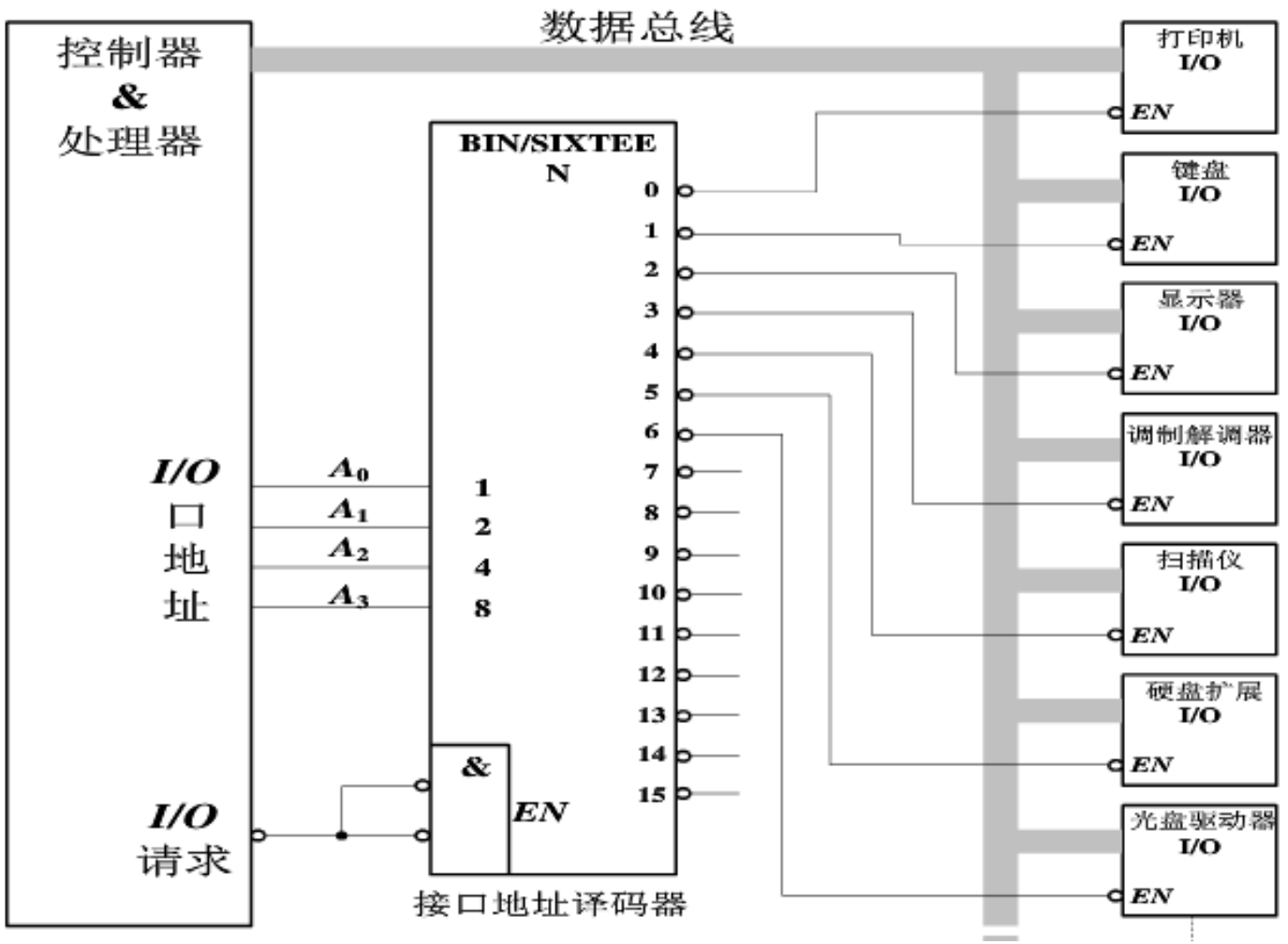
译码器输出：高电平有效 ⇨ $y_i=m_i$



使能端			输入			译码输出							
G_1	G_{2A}	G_{2B}	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	X	X	X	X	X	0	0	0	0	0	0	0	0
X	1	X	X	X	X	0	0	0	0	0	0	0	0
X	X	1	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	1	0
1	0	0	1	1	1	0	0	0	0	0	0	0	1

二进制译码器的典型应用——地址译码

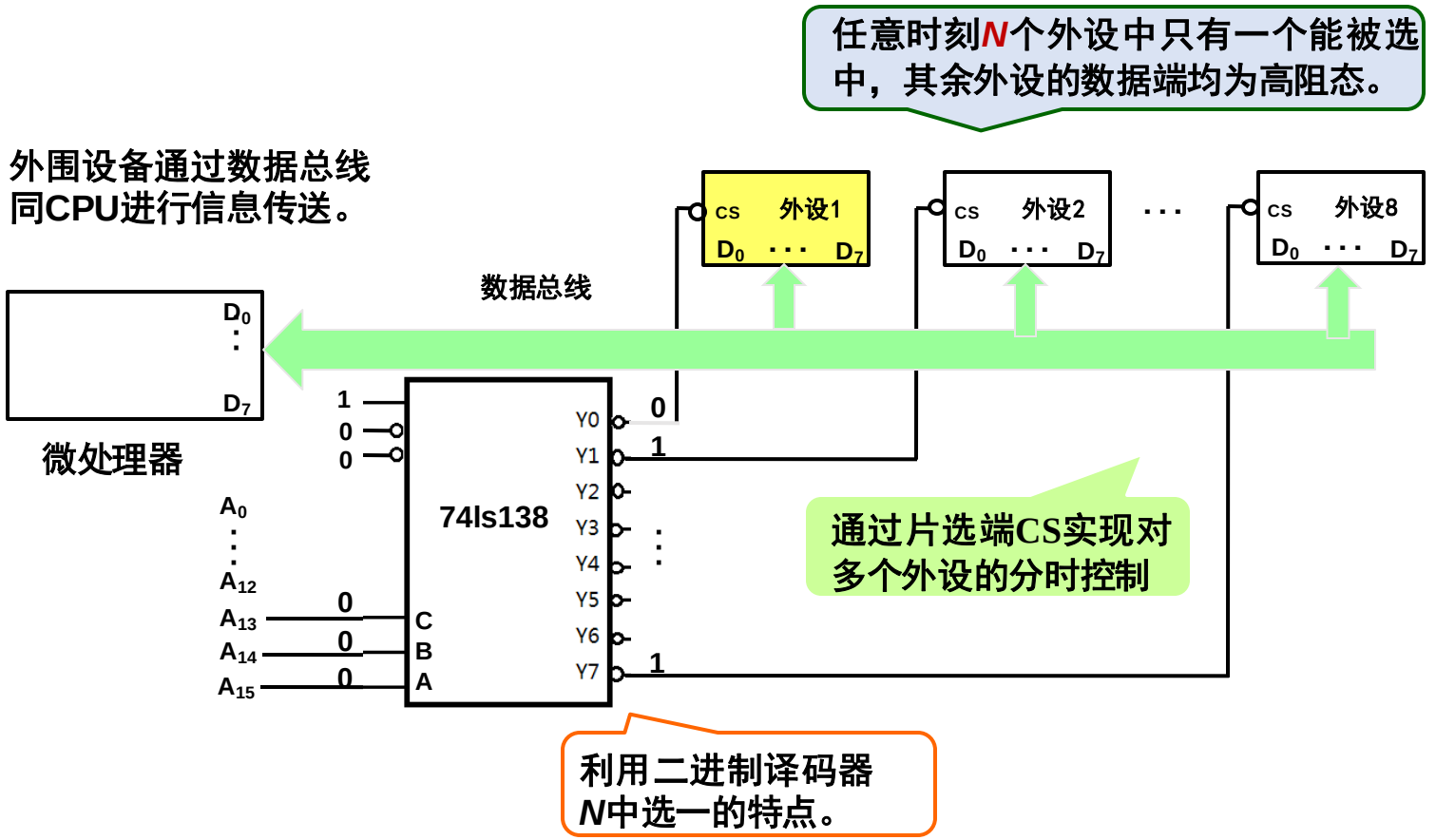
- **微处理器的地址译码**
  - CPU输出一个**存储器地址**到地址总线上，这个地址怎样才能选中一个存储单元，这就是地址译码器要解决的问题。
  - **地址译码器：**输入为数字量（即地址），根据输入的数字量在多个输出端中选一个有效。



二进制译码器的典型应用——地址译码

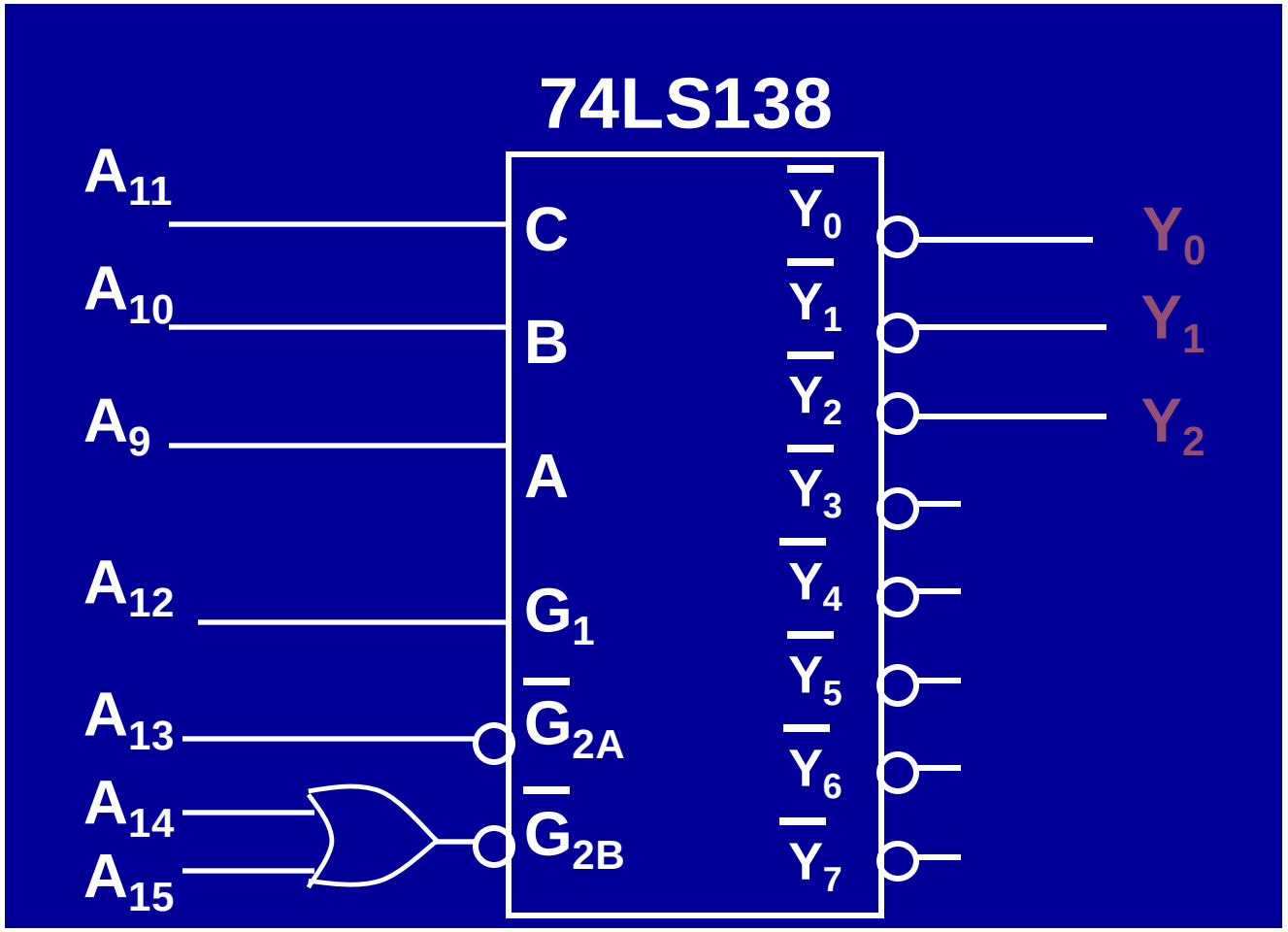
- 微处理器的地址译码

*假设A0—A7连接到各个外设的低8位地址线。



地址译码

- 图示电路的整个地址译码范围？各个外设的地址译码范围？

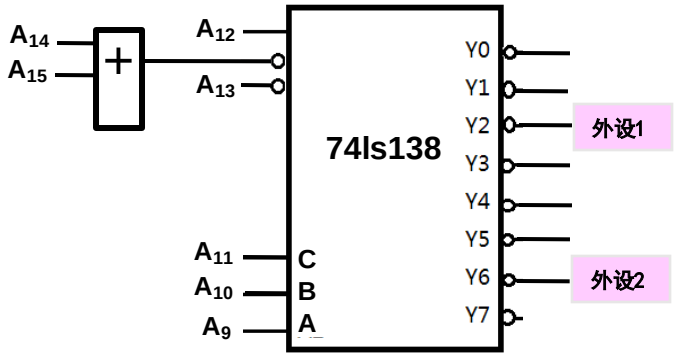


设置

1 整个地址译码范围为： [填空1]H— [填空2]H

2 外设1的地址译码范围为：[填空3]H—[填空4]H

3 外设2的地址译码范围为：[填空5]H—[填空6]H



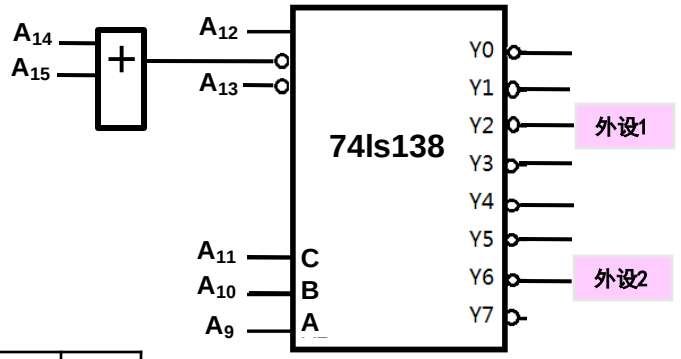
作答

地址译码例题

地址译码

图示电路的整个地址译码范围？

各个外设的地址译码范围？



整个译码器的地址译码范围：

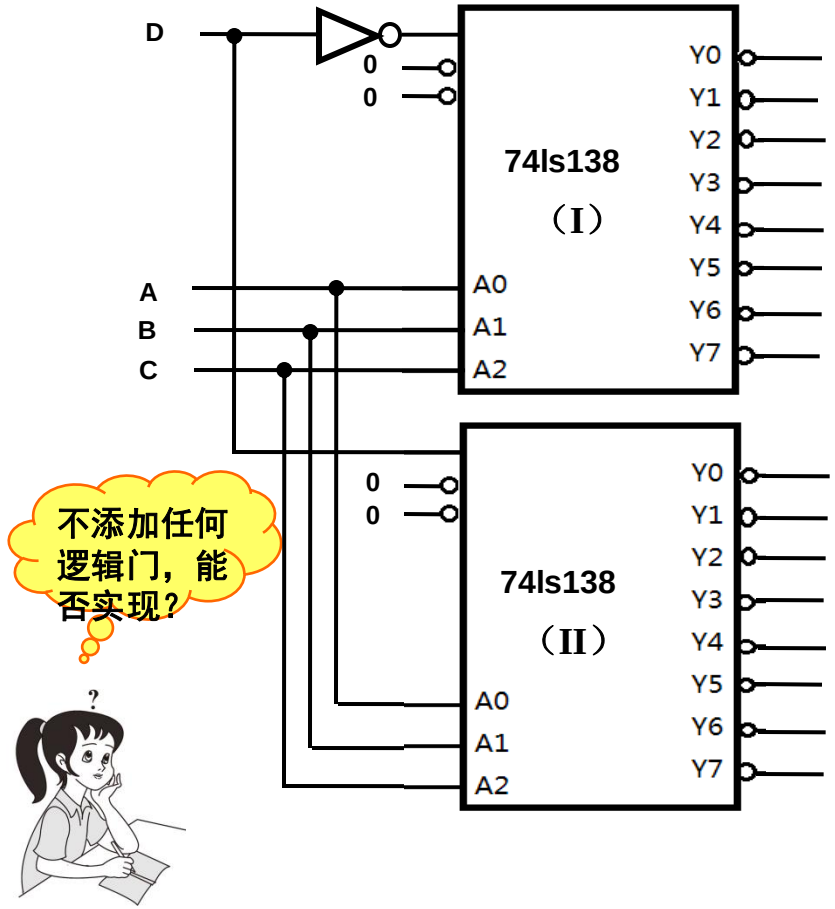
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	0	0	1	0	0	0	X	X	X	X	X	X	X	X	X	最小取值 1000H
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	最大取值 1FFFH

外设1的地址译码范围：

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	0	0	1	0	1	0	X	X	X	X	X	X	X	X	X	最小取值 1400H
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	最大取值 15FFH

二进制译码器的典型应用——译码器级联

- 3线-8线译码器扩展为4线-16线译码



（I）

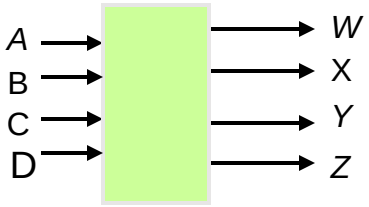
输入				译码输出							
D	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0

（II）

输入				译码输出							
D	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
1	0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1	1	1	1	1
1	0	1	0	1	1	0	1	1	1	1	1
1	0	1	1	1	1	1	0	1	1	1	1
1	1	0	0	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	0

编码转换译码器

- 例：设计一个译码器，
- 将输入的4位二进制数转换为典型格雷码



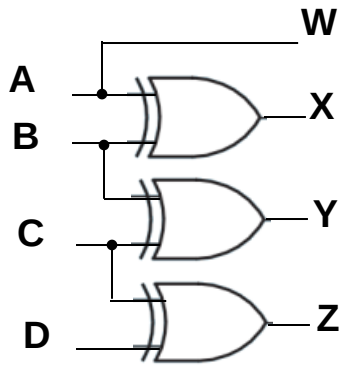
CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

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

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

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

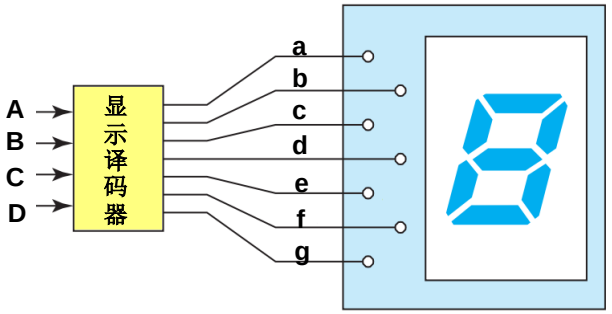
$$W = A$$
$$X = A \oplus B$$
$$Y = B \oplus C$$
$$Z = C \oplus D$$



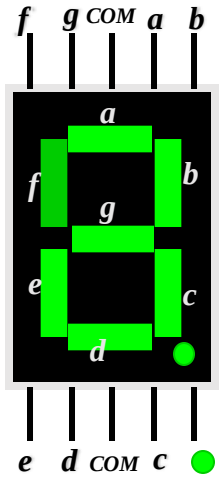
ABCD	WXYZ	ABCD	WXYZ
0000	0000	1000	1100
0001	0001	1001	1101
0010	0011	1010	1111
0011	0010	1011	1110
0100	0110	1100	1010
0101	0111	1101	1011
0110	0101	1110	1001
0111	0100	1111	1000

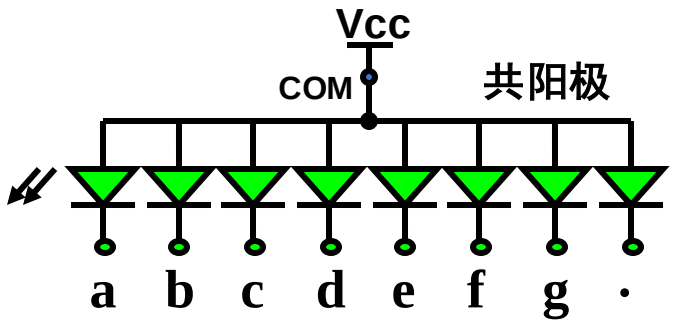
七段显示译码器

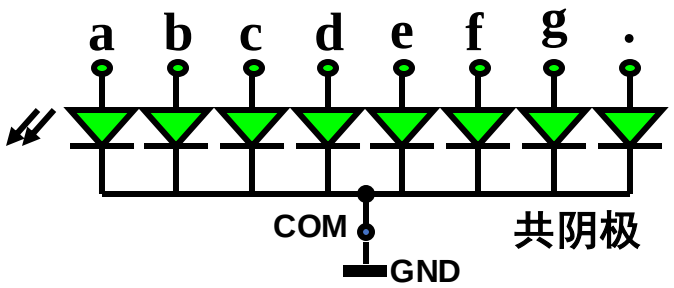
显示译码器:与显示器件（如数码管）配合，将输入代码转换为十进制码或特定编码，并在显示器件上显示相应的字形



七段数码管



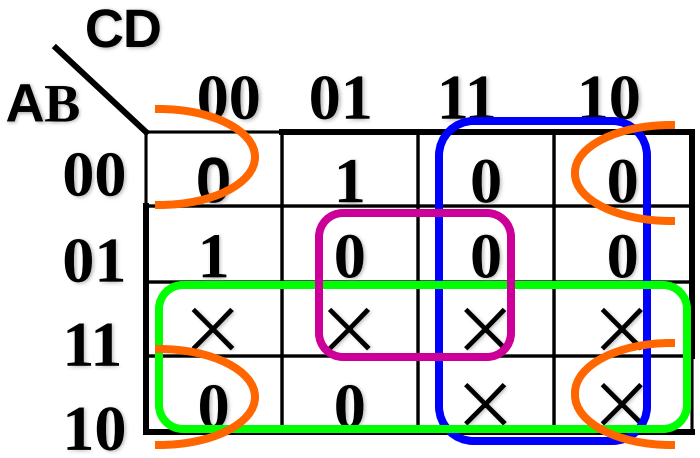




8421BCD码驱动的共阴极七段数码管显示译码器功能表

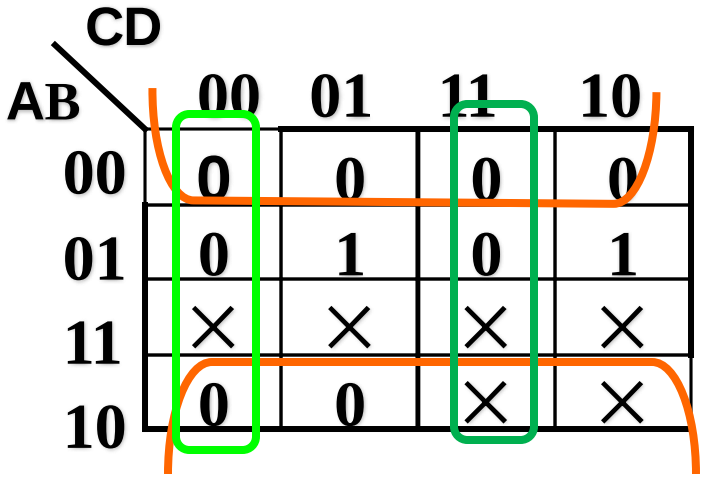
输入				译码输出						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

a



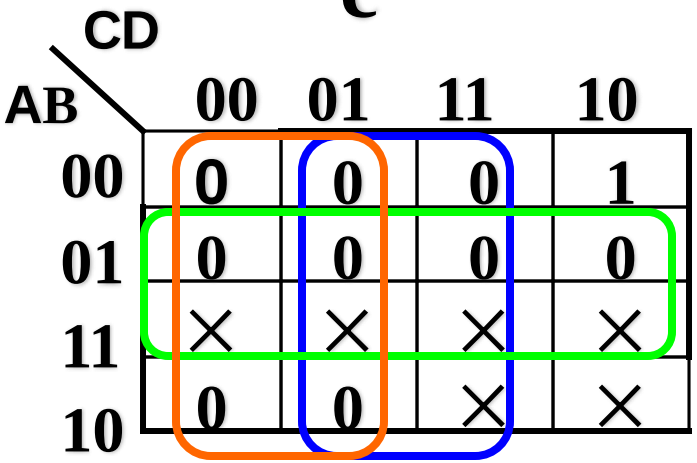
a=A+C+BD+BD

b



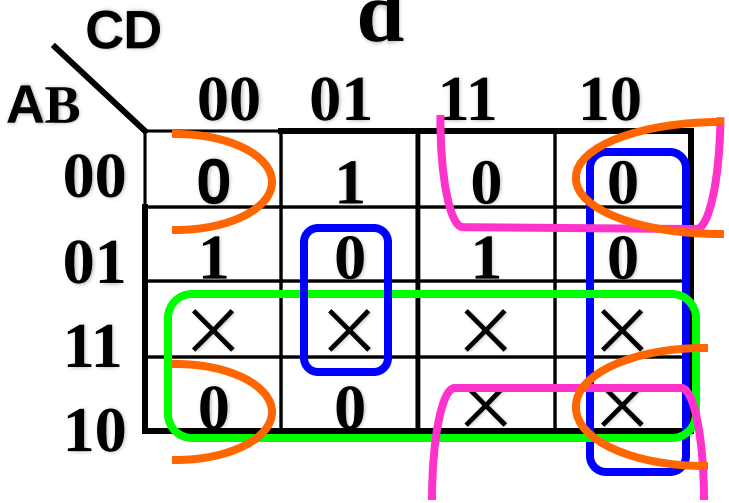
b=B+CD+CD

c



c=B+C+D

d



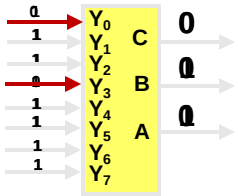
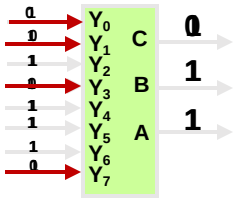
d=A+CD+BC+BD+BCD

Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路

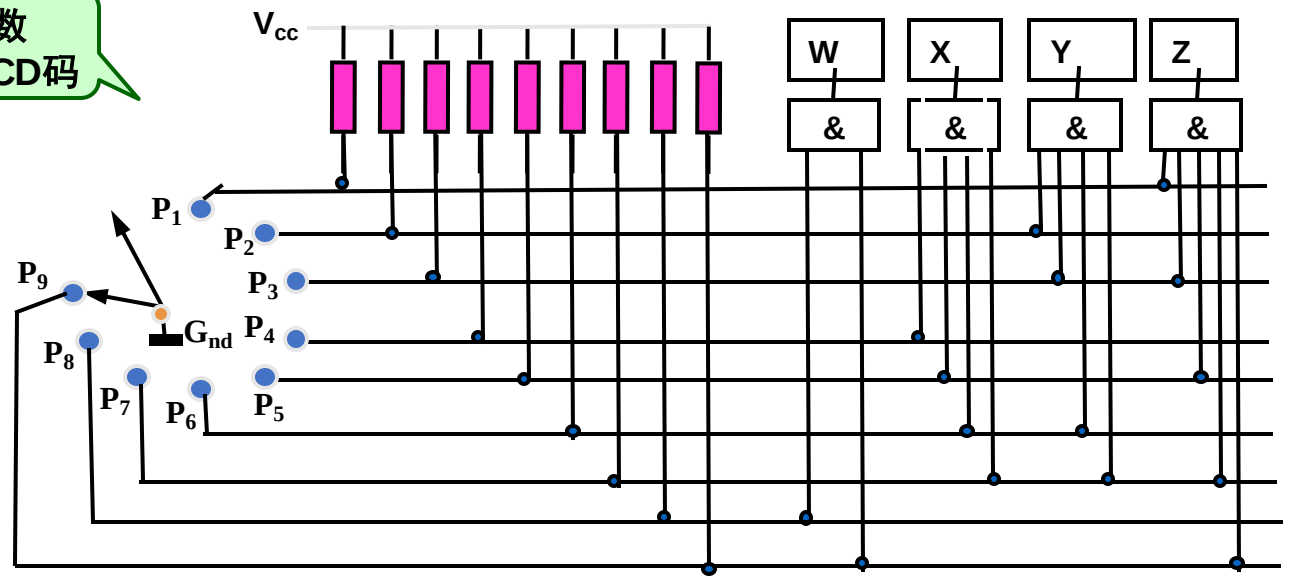
编码器

- 特点：多输入、多输出的组合逻辑电路
- 功能：将二进制码按照一定规律编排，使其具有特定含义，与译码器互逆。

常用编码器	特点	编码演示
普通编码器（二进制编码器）	输入：N位，任何时刻N 根输入线中只能有一个输入有效，N（$N=2^n$）中取一。 输出：n 位二进制码	 （8 线–3 线编码器）
优先编码器	允许同时输入两个以上的有效编码输入信号，优先编码器能按照预先设定的优先级别，只对其中优先级最高的输入进行编码。	 （8 线–3 线优先编码器）

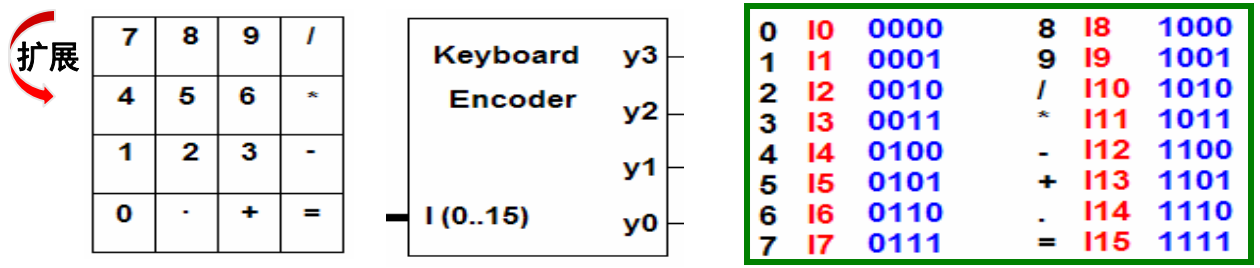
键盘编码器

输入：十进制数
输出：8421BCD码



键盘编码器功能表

P_9…… $.P_1$	按键	WXYZ
111111111	0	0000
111111110	1	0001
111111101	2	0010
111111011	3	0011
111110111	4	0100
111101111	5	0101
111011111	6	0110
110111111	7	0111
101111111	8	1000
011111111	9	1001



$W=(P_8 \cdot P_9)'$　　$X=(P_4 \cdot P_5 \cdot P_6 \cdot P_7)'$

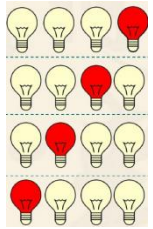
$Y=(P_2 \cdot P_3 \cdot P_6 \cdot P_7)'$　　$Z=(P_1 \cdot P_3 \cdot P_5 \cdot P_7 \cdot P_9)'$

键盘编码器

4:2编码器

计算机配有四个外部设备：声卡(A0)，硬盘驱动器(A1)，鼠标(A2)，网卡(A3)，B_0、B_1为编码输出。

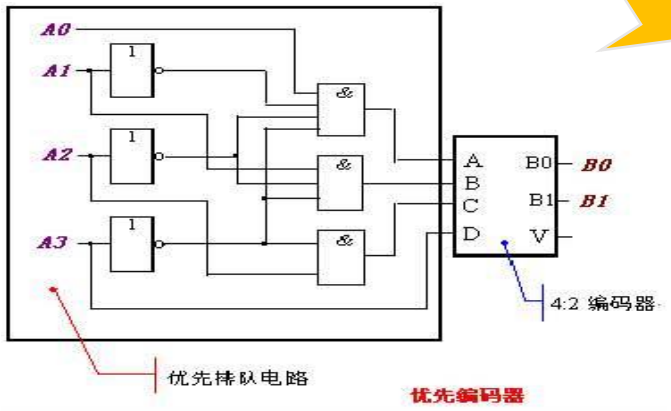
抢答器输出



A_3	A_2	A_1	A_0	B_1	B_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

某一时刻只允许输入一个编码信号，如A_1（A_1=1）向 CPU 请求传送数据，CPU 根据接收的编码 B_1B_0 = 01，启动硬盘驱动器，开始传送数据。

普通编码器：无法避免错误输入（同时输入多路有效信号），容易造成混乱。



4:2优先编码器

A_3	A_2	A_1	A_0	B_1	B_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

$$A = A0\ \overline{A1}\ \overline{A2}\ \overline{A3}$$
$$B = A1\ \overline{A2}\ \overline{A3}$$
$$C = A2\ \overline{A3}$$
$$D = A3$$

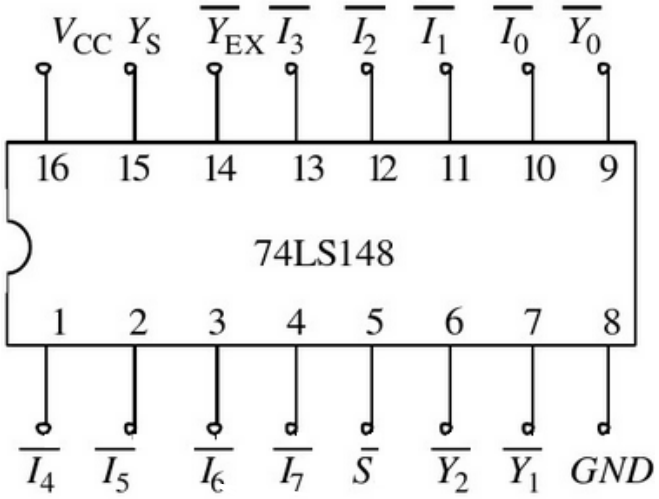
二进制编码器：

- 可以对2^n个输入对象编码
- 只需n个输出端（每个对象获得一个n位编码）
- 编码具有唯一性

优先编码器：

- 允许同时输入多路有效信号
- 按照预先设定的优先级，只对其中优先级最高的输入进行编码。

编码器典型芯片74LS148



输入和输出均为低电平被有效。

标志位：
0：编码输出；1：非编码输出

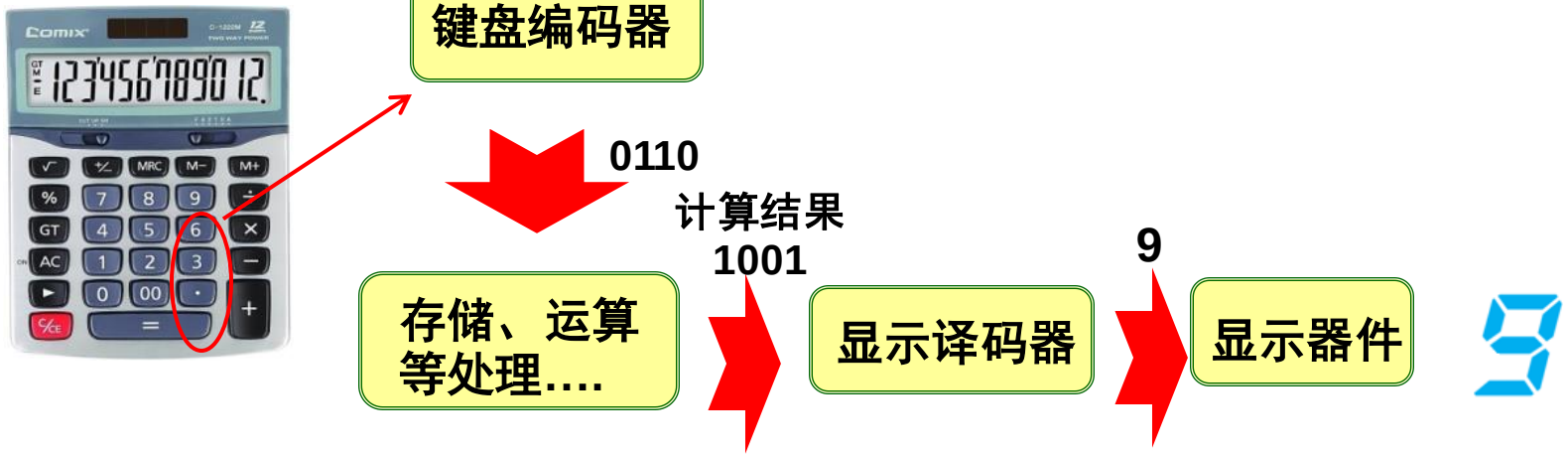
输入使能

输出使能

8线-3线优先编码器

输入									输出				
$\overline{S}$	$\overline{I}_7$	$\overline{I}_6$	$\overline{I}_5$	$\overline{I}_4$	$\overline{I}_3$	$\overline{I}_2$	$\overline{I}_1$	$\overline{I}_0$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$	$\overline{Y}_{EX}$	Y_S
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	×	×	×	×	×	×	×	0	0	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	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	1

编码器与译码器的实际应用



Unit 7 组合逻辑元件

- 多路复用器(multiplexers)
- 三态器件(Three-state Buffer)
- 译码器(Decoders)
- 编码器(Encoders)
- 奇偶校验器
- 比较器
- 只读存储器(ROM)
- 利用MSI设计组合逻辑电路