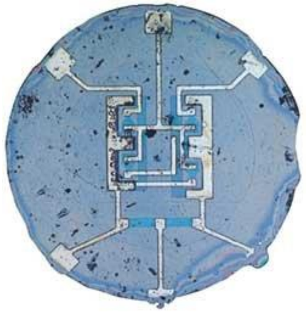
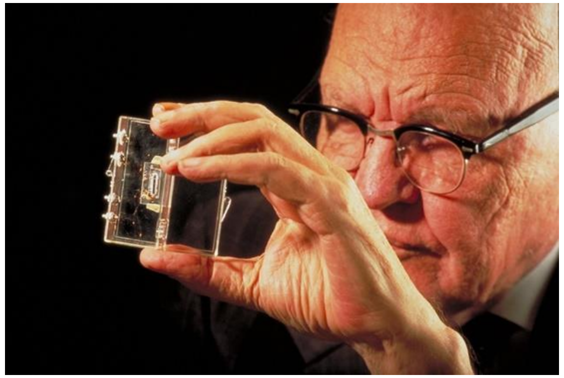
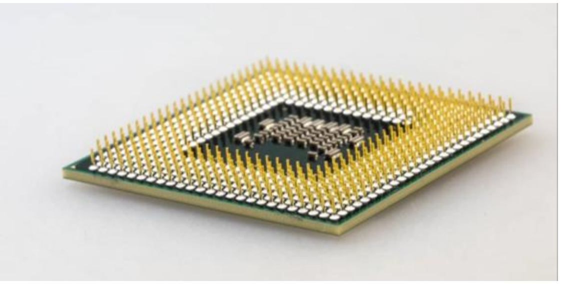
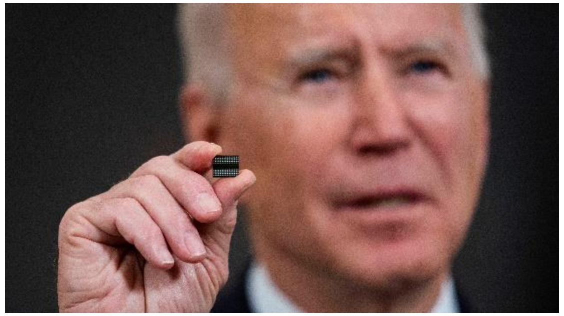
数字逻辑设计

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





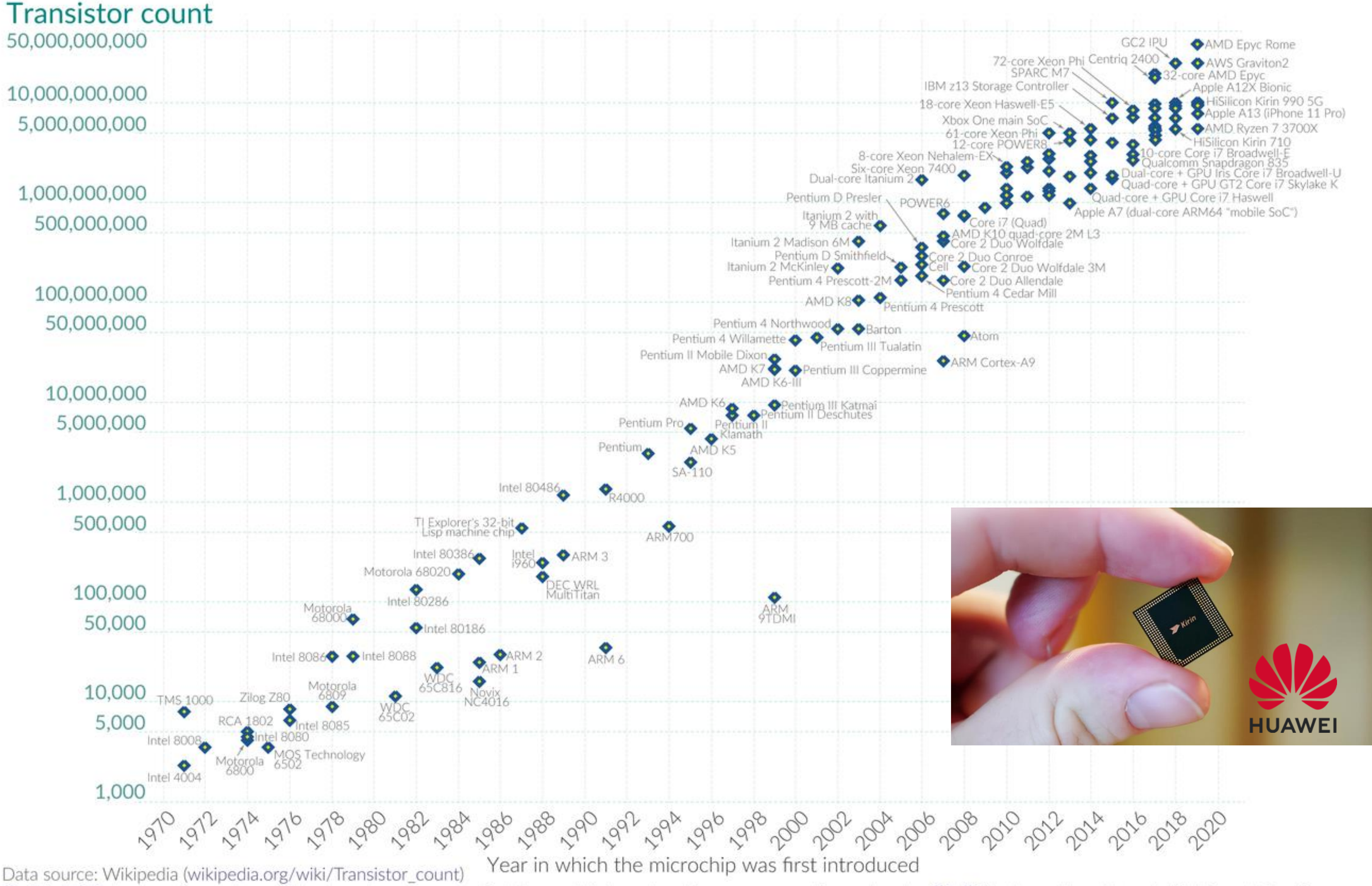




Moore's Law: The number of transistors on microchips doubles every two years

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.

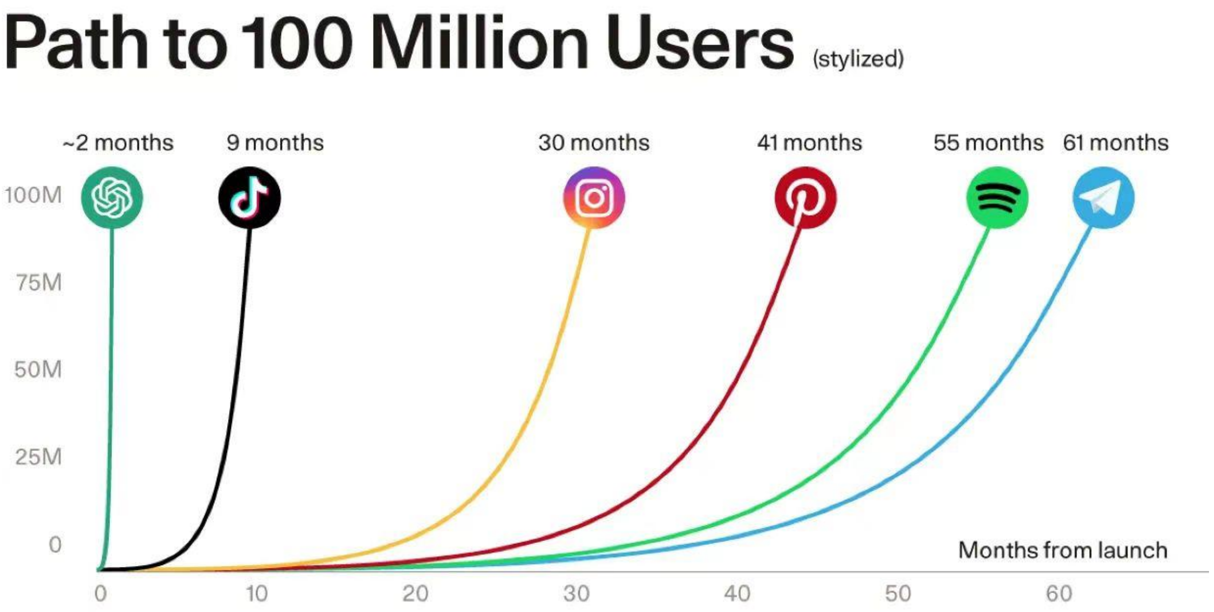


Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

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

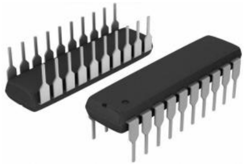
英伟达3个月卖出800吨H100！云服务供应商的大规模H100集群容量即将耗尽，全球陷入GPU短缺——整体算来，全球公司需要约432000张H100。

by 新智元



WHEN EVERYONE DIGS FOR GOLD
SHOVELS $99
$
$
NVIDIA.
Microsoft
G
$
Meta
SELL
SHOVELS
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）	<10 gates	逻辑门、触发器等		① 学会查阅器件资料； ② 典型集成电路芯片的功能、外特性； ③ 能熟练运用并完成设计要求。
中规模（MSI）	10~100 gates	译码器、计数器、加法器等模块		
大规模（LSI）	100~10000 gates	存储器、微处理器或复杂的数字系统	 	① 了解典型PLD集成芯片的功能和特性； ② 能够使用HDL语言完成逻辑设计；
超大规模（VLSI）	>10000 gates			

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）

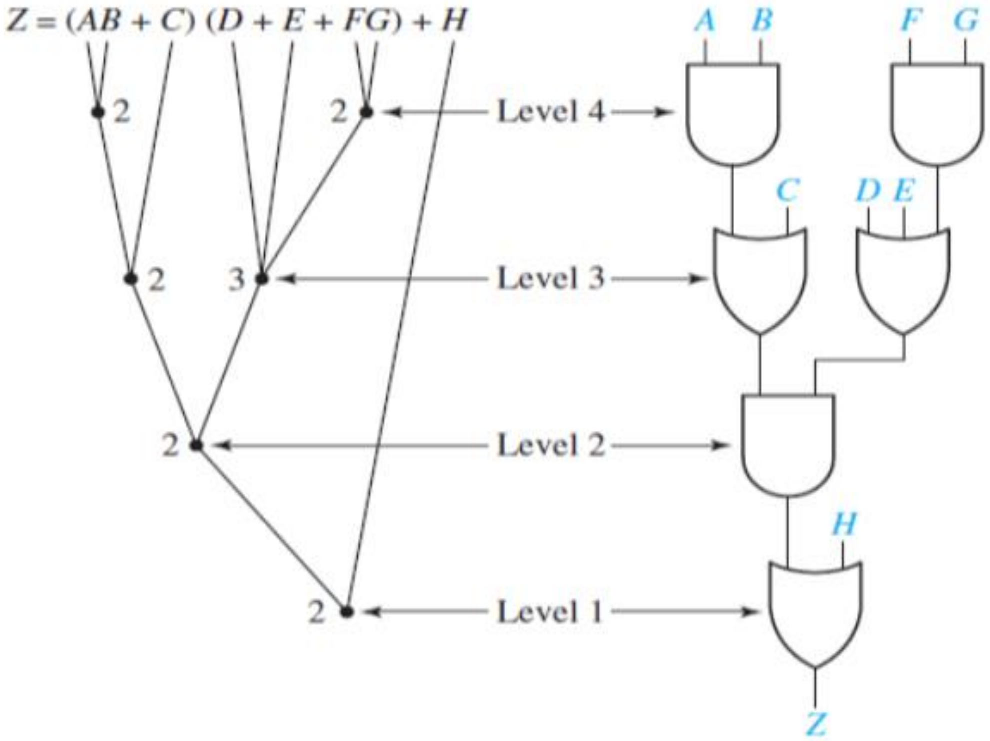
  OR-AND 电路（和之积）

- 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

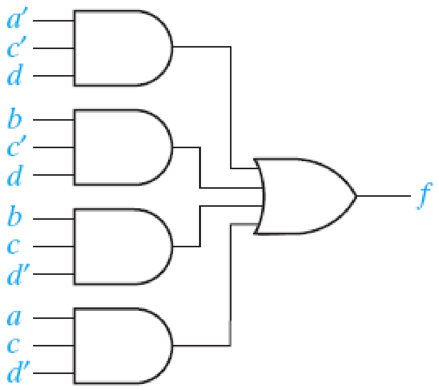
AND-OR 电路（积之和）

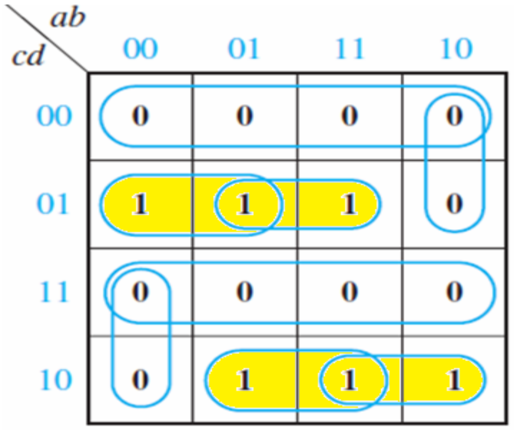
$$f = a'c'd + bc'd + bcd' + acd'$$

OR-AND 电路（和之积）

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

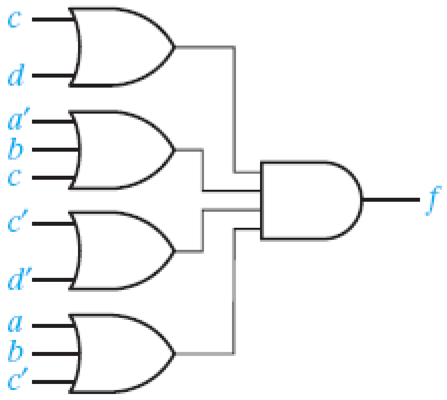
5个门, 16 个输入端





cd \ ab	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	0	0	0	0
10	0	1	1	1

5个门, 14 个输入端



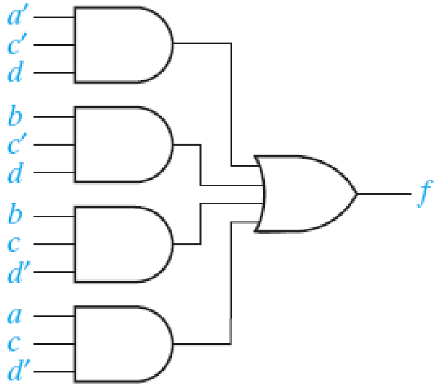
多级门电路

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

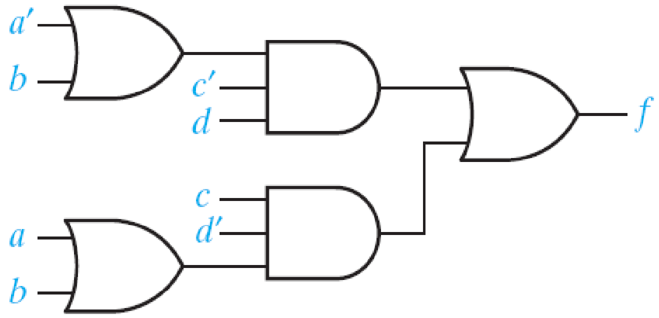


2. 三级电路

OR-AND-OR 电路

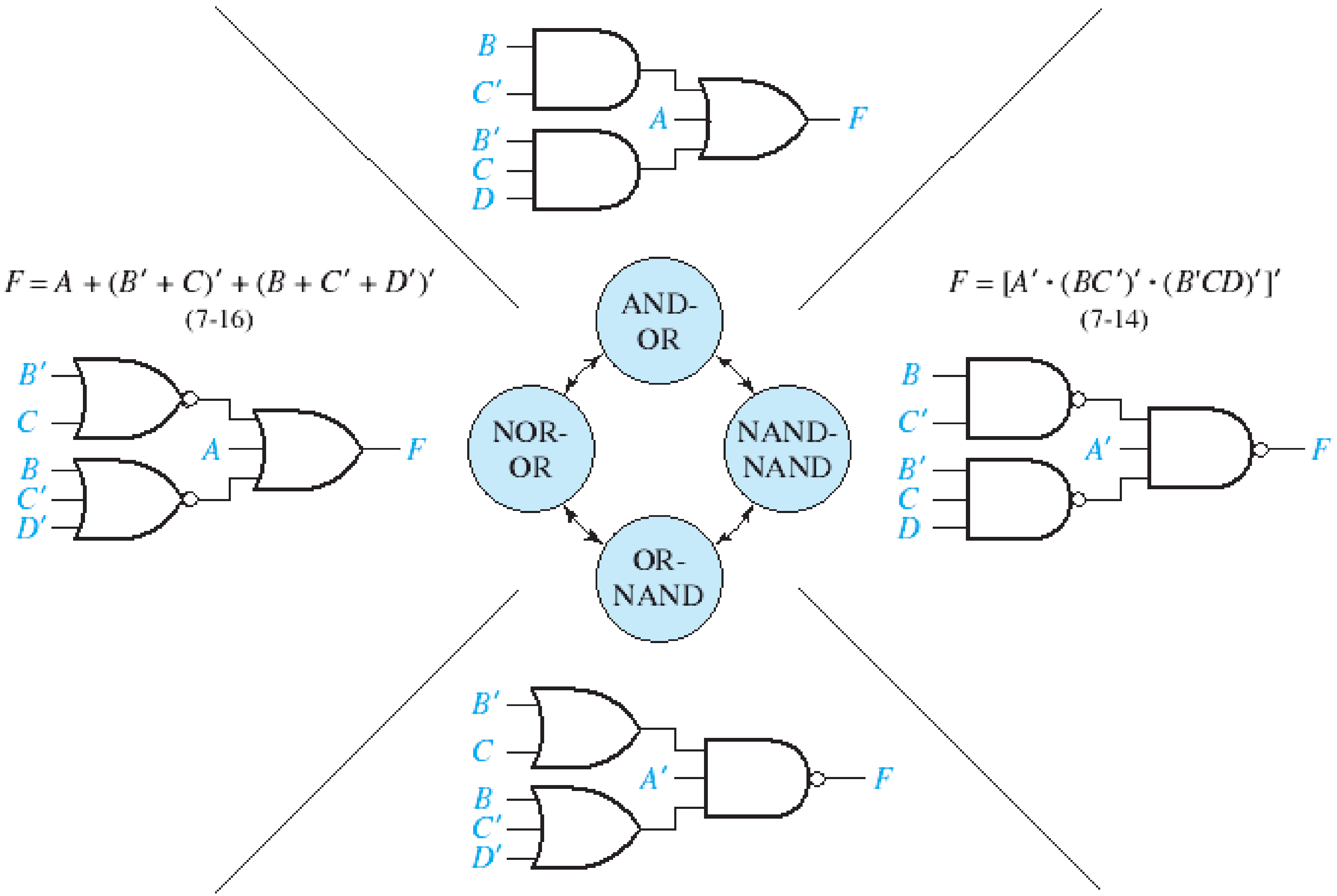
$$f = c'd(a' + b) + cd'(a + b)$$

5个门, 12 个输入端



二级门电路的8种基本形式——1

$F = A + BC' + B'CD$ (7-13)



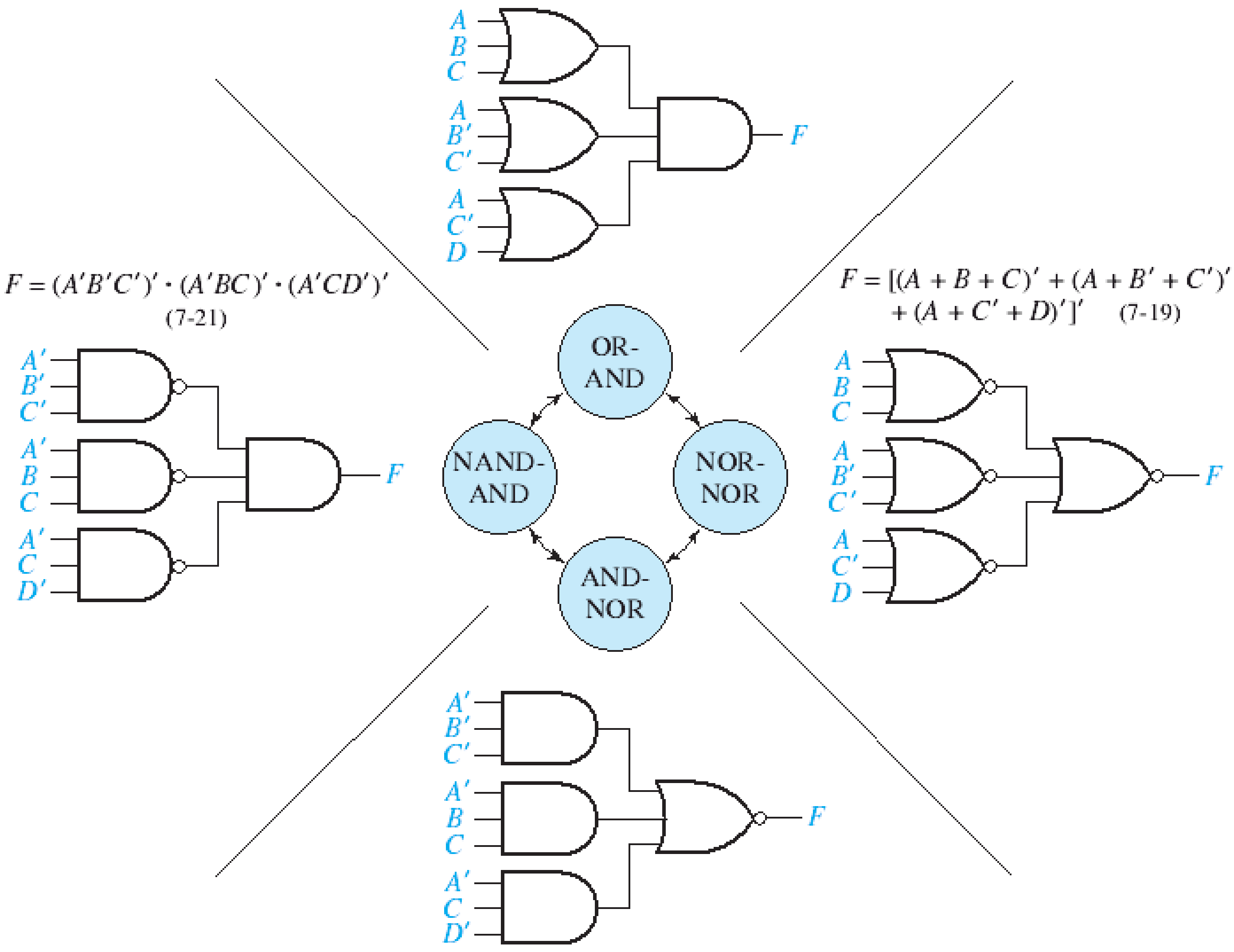
$F = [A' \cdot (BC')' \cdot (B'CD)']'$ (7-14)

$F = A + (B' + C)' + (B + C' + D')'$ (7-16)

$F = [A' \cdot (B' + C) \cdot (B + C' + D')]'$ (7-15)

二级门电路的8种基本形式——2

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



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

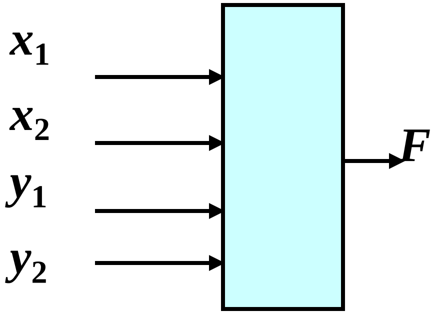
$$F = [(A + B + C)' + (A + B' + C')' + (A + C' + D)']' \qquad (7\text{-}19)$$

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

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



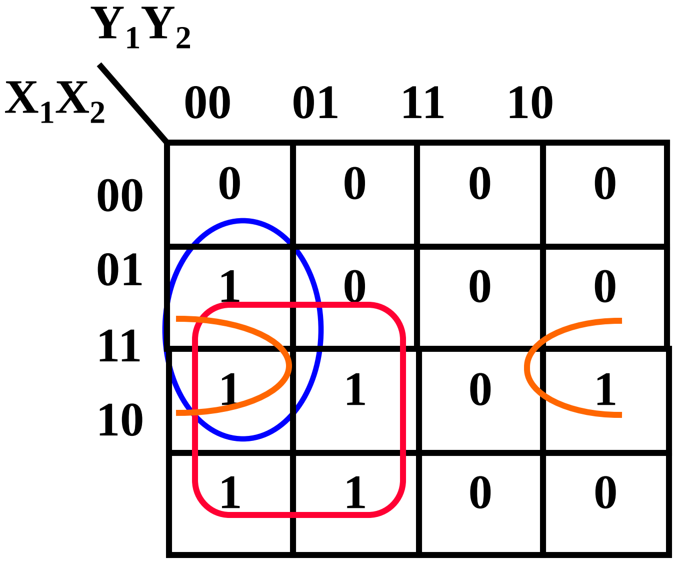
② 真值表

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

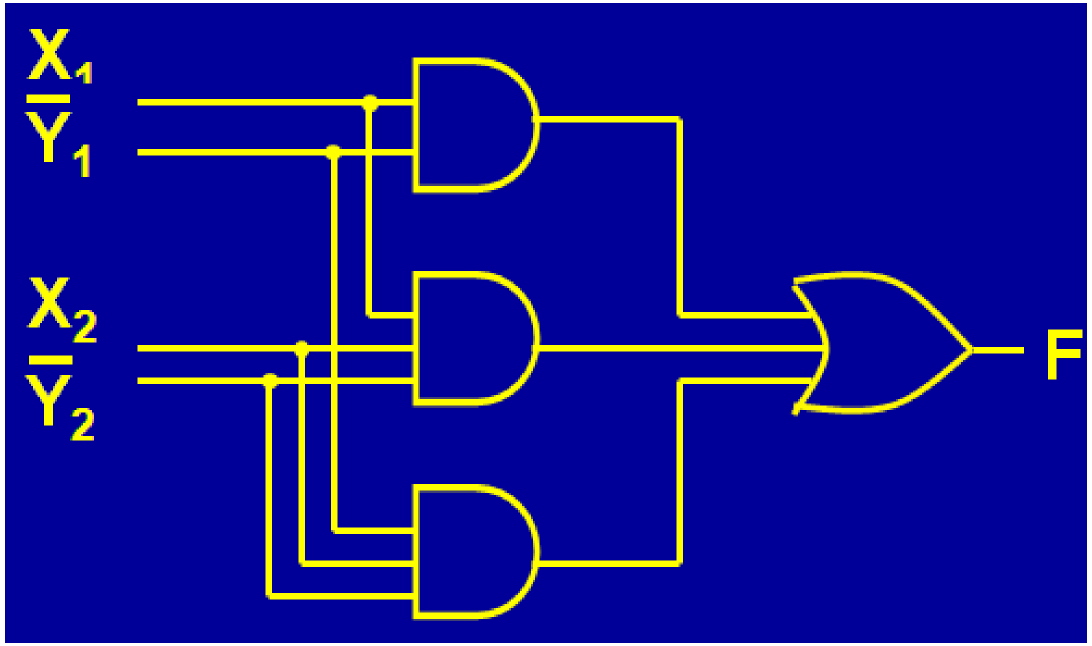
多级门电路

③ 最简二级与或电路

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



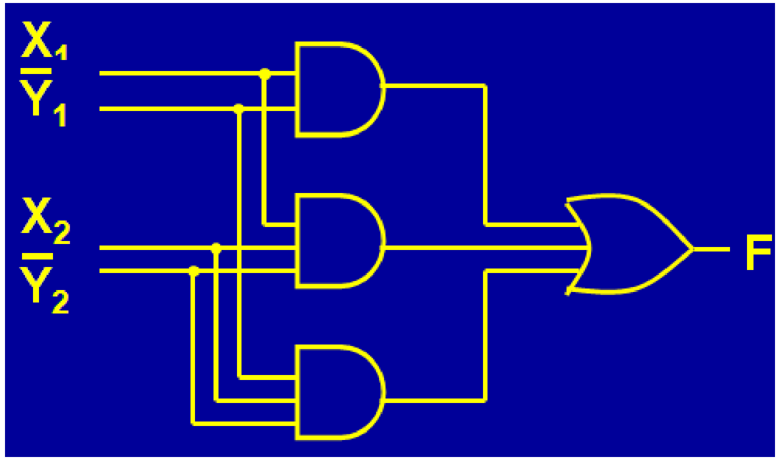
X_1X_2 \ Y_1Y_2	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

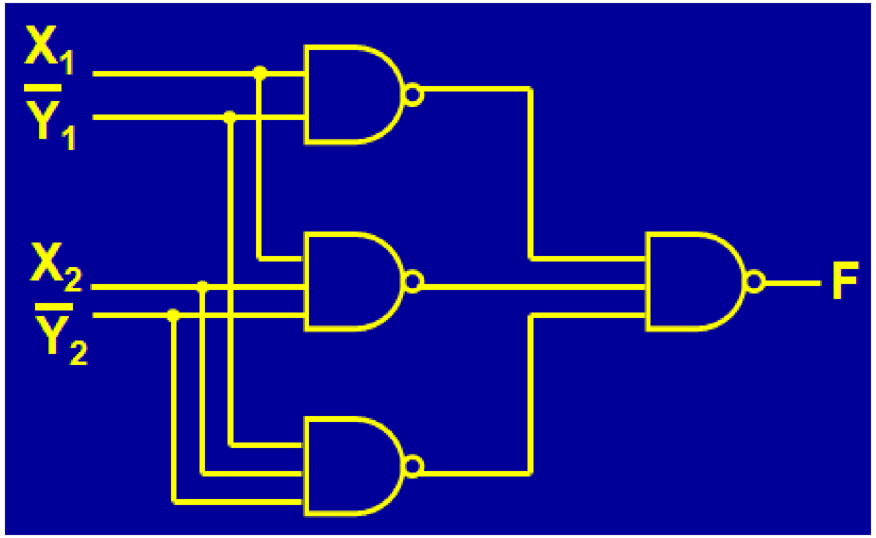


多级门电路

④ 采用单一逻辑门(与非门)设计

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





二级门电路的设计

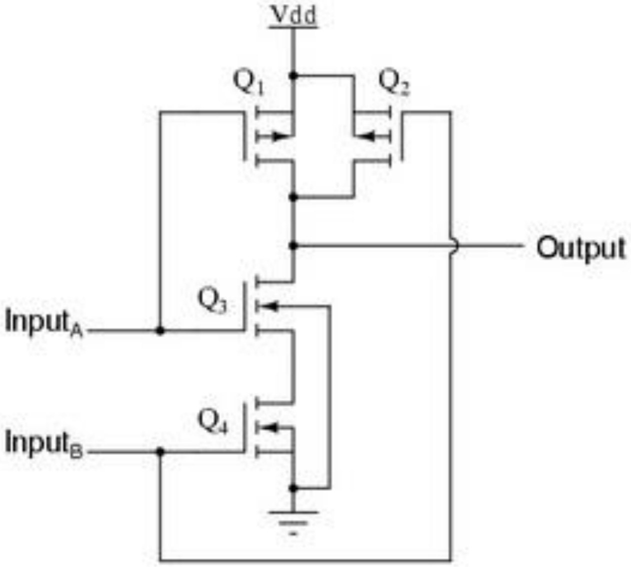
任何逻辑都可以用二级门电路实现

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

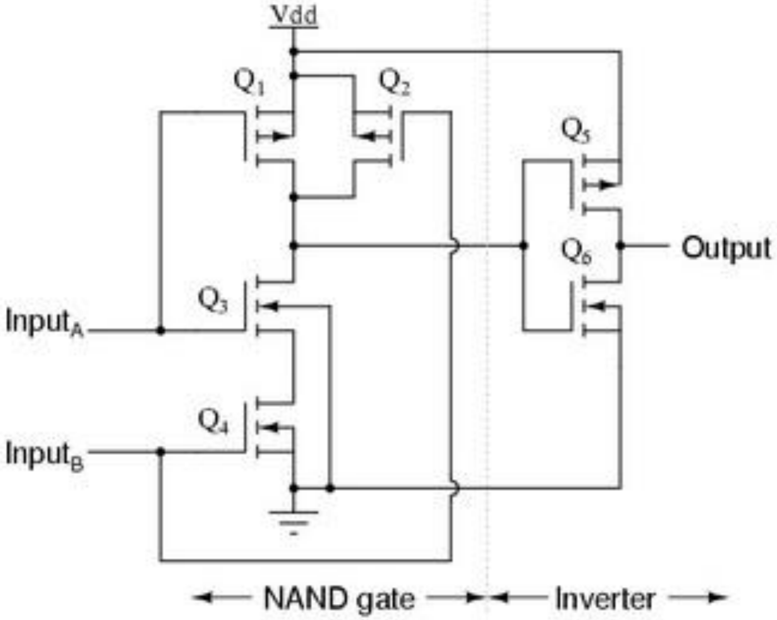
$$F'(X,Y,Z)=\sum_{XYZ}(0,2,3,4,5)=\prod_{XYZ}(1,6,7)$$

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

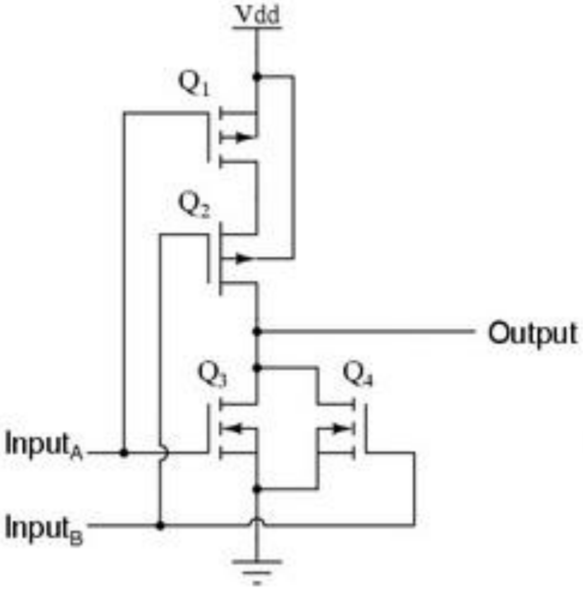
CMOS NAND gate



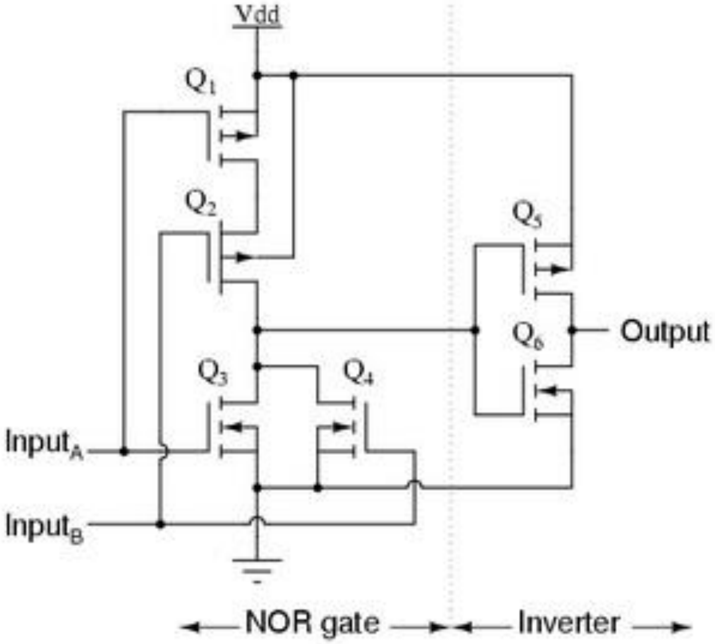
CMOS AND gate



CMOS NOR gate



CMOS OR gate



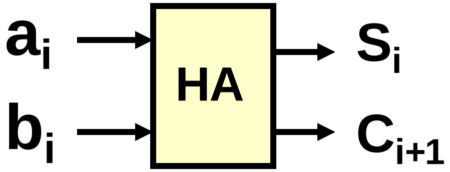
与非门/或非门节省空间和门延迟。

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

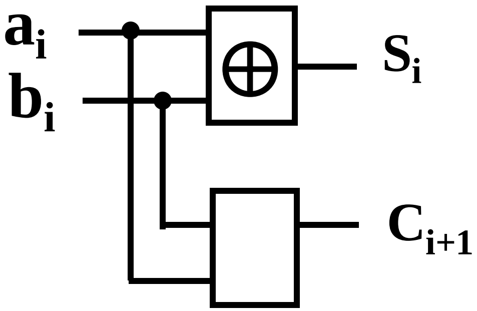


$$S_i = a_i \oplus b_i$$

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

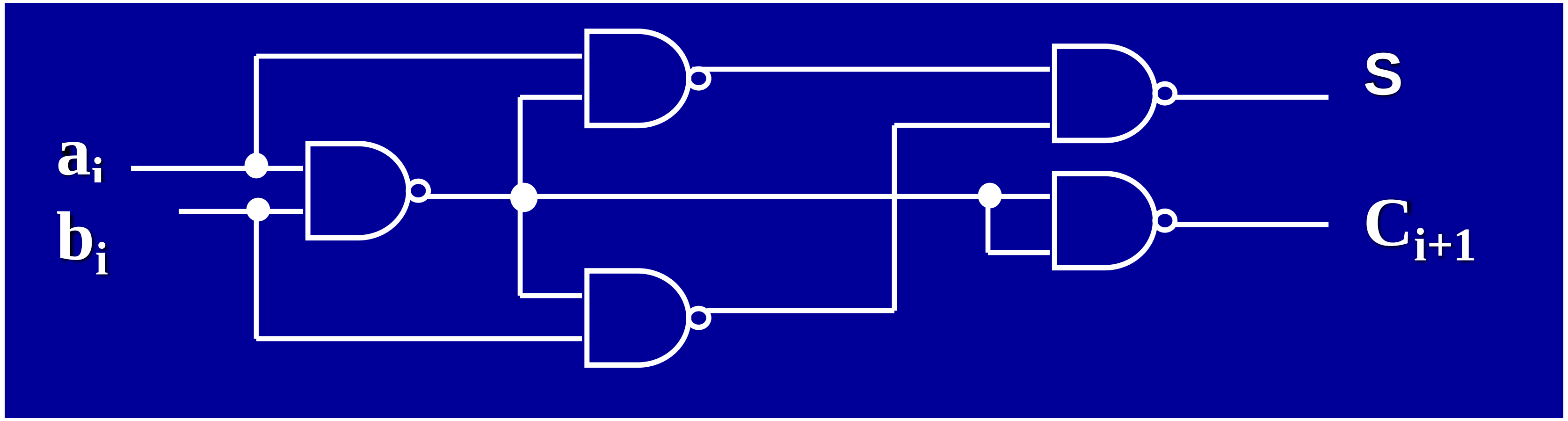
真值表

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

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



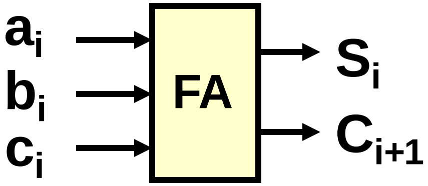
全加器（Full Adder）

```
   1 0 1 1 ……….. A
   1 1 1 0 ……….. B
+        0 ………. Cᵢ
─────────────────────
           ……….. Sᵢ
```

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

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

全加器表示

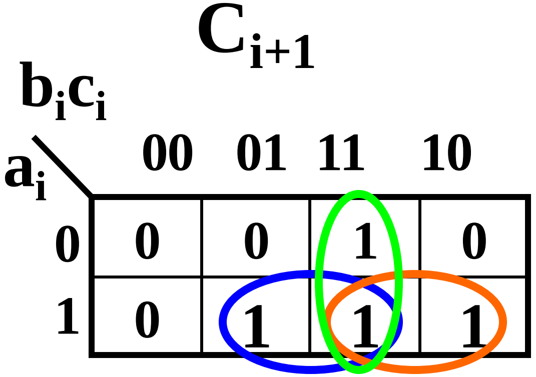


a_i	b_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

$$S_i = \bar{a}_i\bar{b}_ic_i + a_i\bar{b}_i\,\bar{c}_i + a_i\,\bar{b}_i\bar{c}_i + a_i\,b_i\,c_i$$

$$=(\bar{a}_i\bar{b}_i + a_i\,b_i)\,c_i + (a_i\bar{b}_i + a_i\,\bar{b}_i)\,\bar{c}_i$$

$$=\overline{(a_i \oplus b_i)}\,c_i + (a_i \oplus b_i)\,\bar{c}_i$$

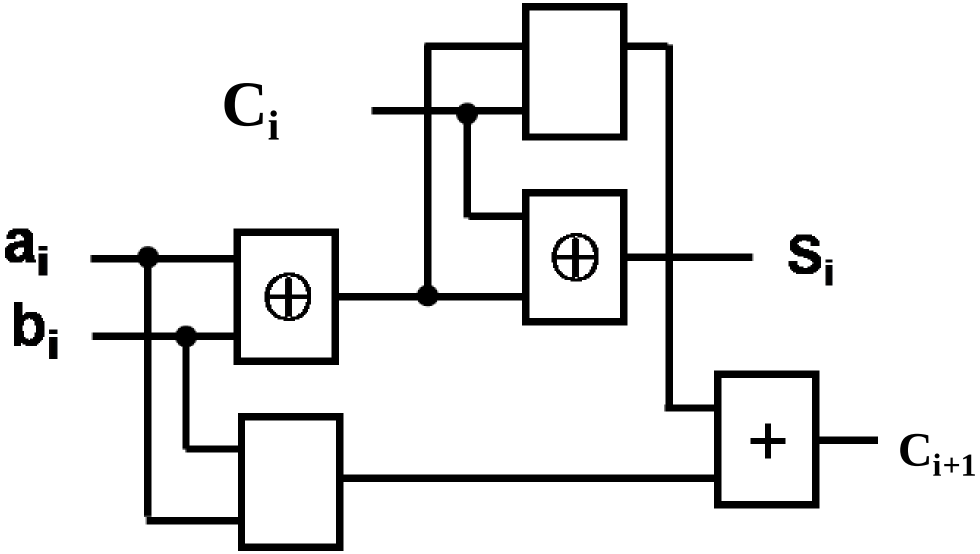
$$= a_i \oplus b_i \oplus C_i$$

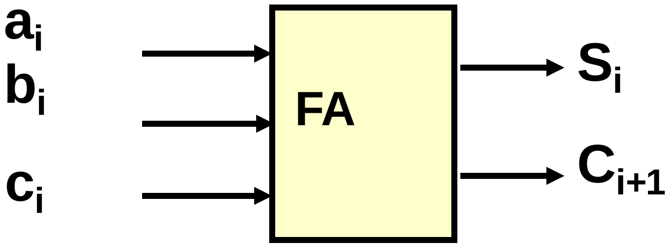
$$C_{i+1}=(a_i \oplus b_i\,)\,C_i + a_ib_i$$

全加器逻辑表示

- **solution 1：**

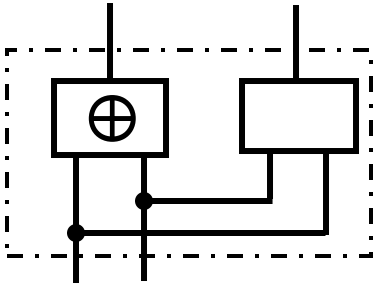
$$\begin{cases} S_i = a_i \oplus b_i \oplus C_i \\ C_{i+1} = (a_i \oplus b_i)\, C_i + a_i b_i \end{cases}$$

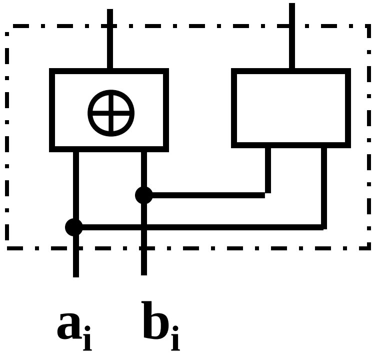




全加器逻辑表示（2）

- **solution 2**





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

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

$$\Uparrow$$

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

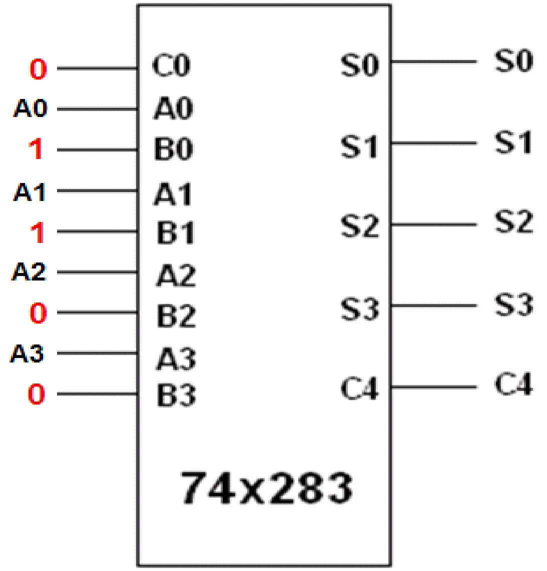
全加器的应用

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



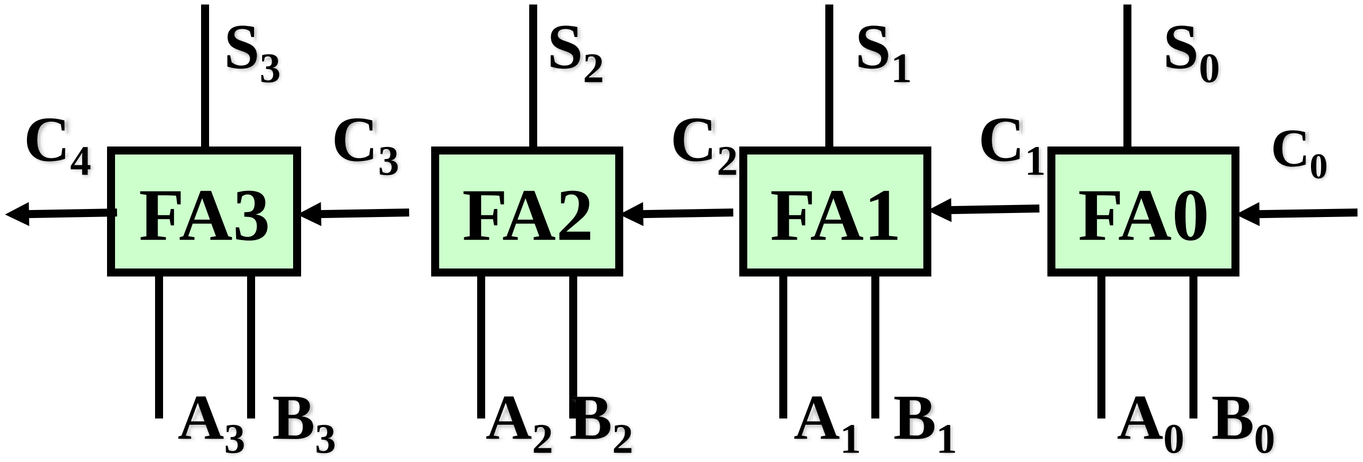
$A_3A_2A_1A_0$: 输入 8421 BCD码

$S_3S_2S_1S_0$: 输出余3 码

S = A+0011

4位并行加法器

（1）串行进位



$A= A_3A_2A_1A_0 =1011$

$B= B_3B_2B_1B_0 =1110$

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

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

- 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

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

$\mathrm{A}=\mathrm{A}_3\mathrm{A}_2\mathrm{A}_1\mathrm{A}_0=1011$

$\mathrm{B}=\mathrm{B}_3\mathrm{B}_2\mathrm{B}_1\mathrm{B}_0=1110$

$C_{i+1}=P_iC_i+G_i$ ——进位迭代公式

$P_i= A_i \oplus B_i$

$G_i= A_iB_i$

$$C_1=\mathrm{P}_0C_0+G_0$$

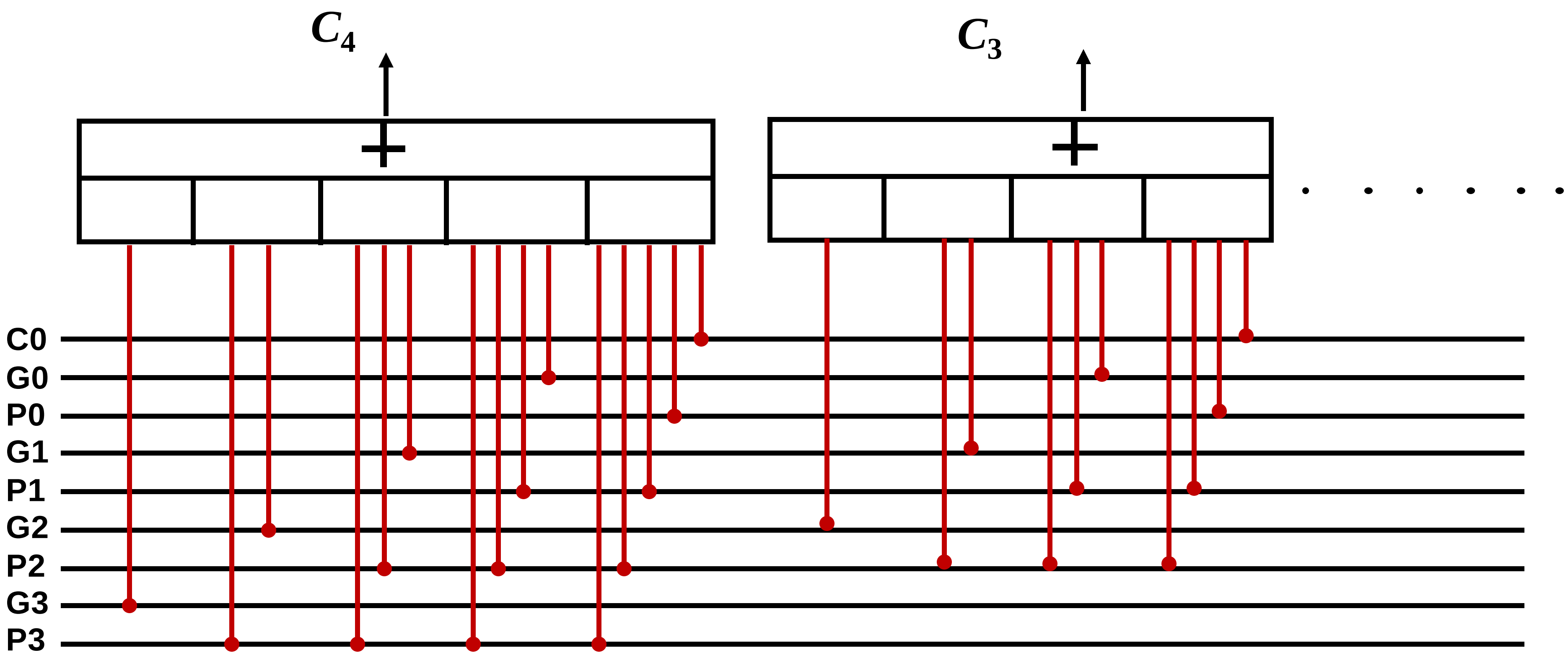
$$C_2=\mathrm{P}_1C_1+G_1=\mathrm{P}_1\mathrm{P}_0C_0+\mathrm{P}_1G_0+G_1$$

$$C_3=\mathrm{P}_2C_2+G_2=\mathrm{P}_2\mathrm{P}_1\mathrm{P}_0C_0+\mathrm{P}_2\mathrm{P}_1G_0+\mathrm{P}_2G_1+G_2$$

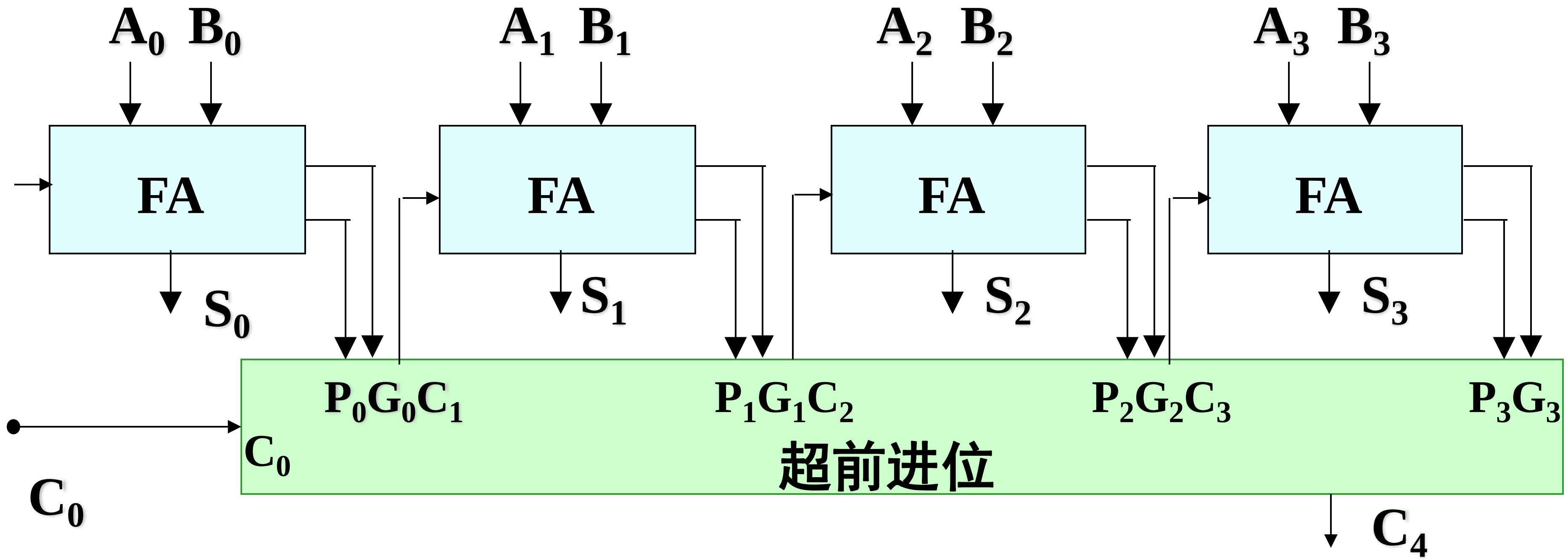
$$C_4=\mathrm{P}_3C_3+G_3=\mathrm{P}_3\mathrm{P}_2\mathrm{P}_1\mathrm{P}_0C_0+\mathrm{P}_3\mathrm{P}_2\mathrm{P}_1G_0+\mathrm{P}_3\mathrm{P}_2G_1+\mathrm{P}_3G_2+G_3$$

4位并行加法器

（2） 超前进位



4位并行加法器



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

$$C_1 = P_0C_0 + G_0$$

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

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

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

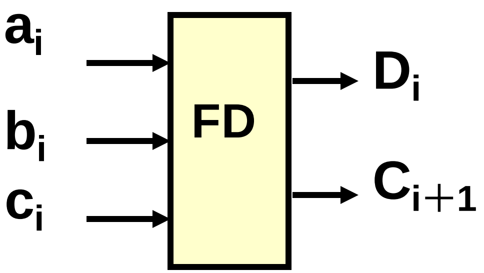
全减器（Binary Full Subtracter）

1 1 1 0 .... A

1 0 1 1 .... B

- 0 0 ....C_i

.... D_i



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

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

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

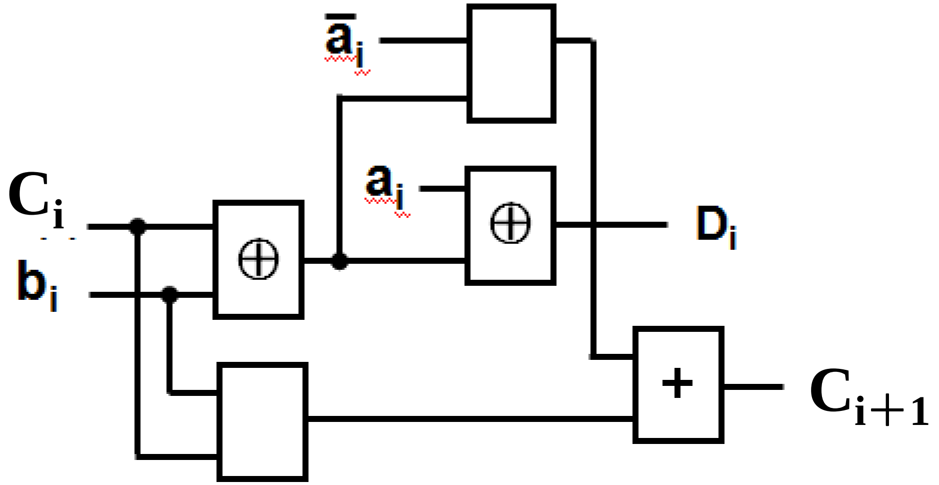
$$\begin{cases} D_i = a_i \oplus b_i \oplus C_i \\ C_{i+1} = (C_i \oplus b_i)\,\overline{a_i} + C_i b_i \end{cases}$$

D_i

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

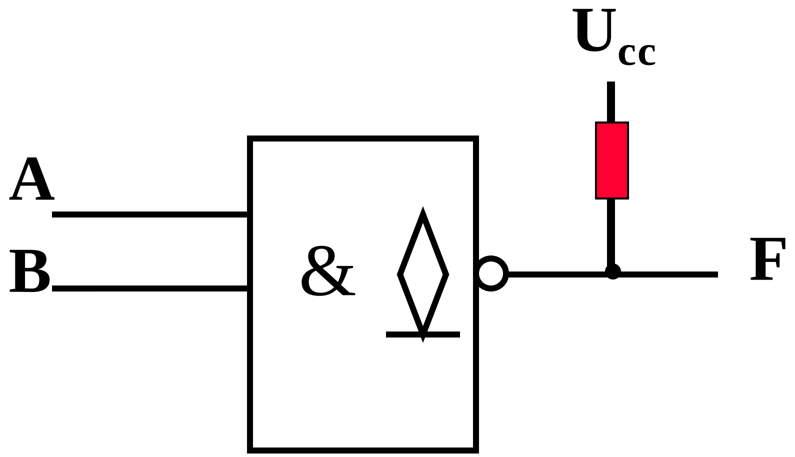
C_{i+1}

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



OC门（集电极开路门：Open Collector Gate）

$F=\overline{AB}$

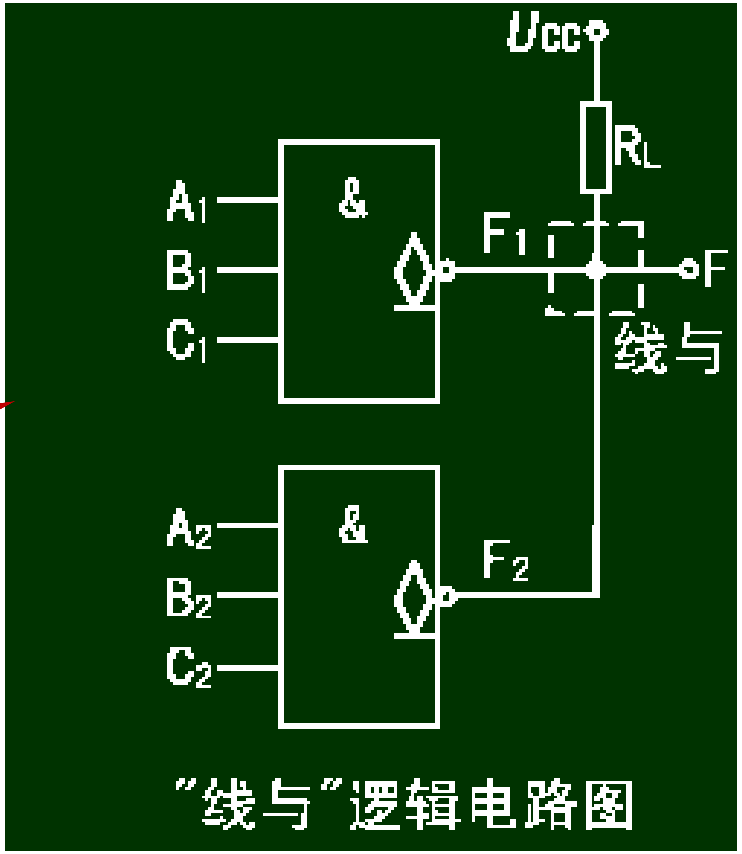


- 几个OC门的输出端可以直接互连：“线与”
- 使用时必须加负载/上拉电阻

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