Unit 3-5 组合逻辑电路案例

数字逻辑设计

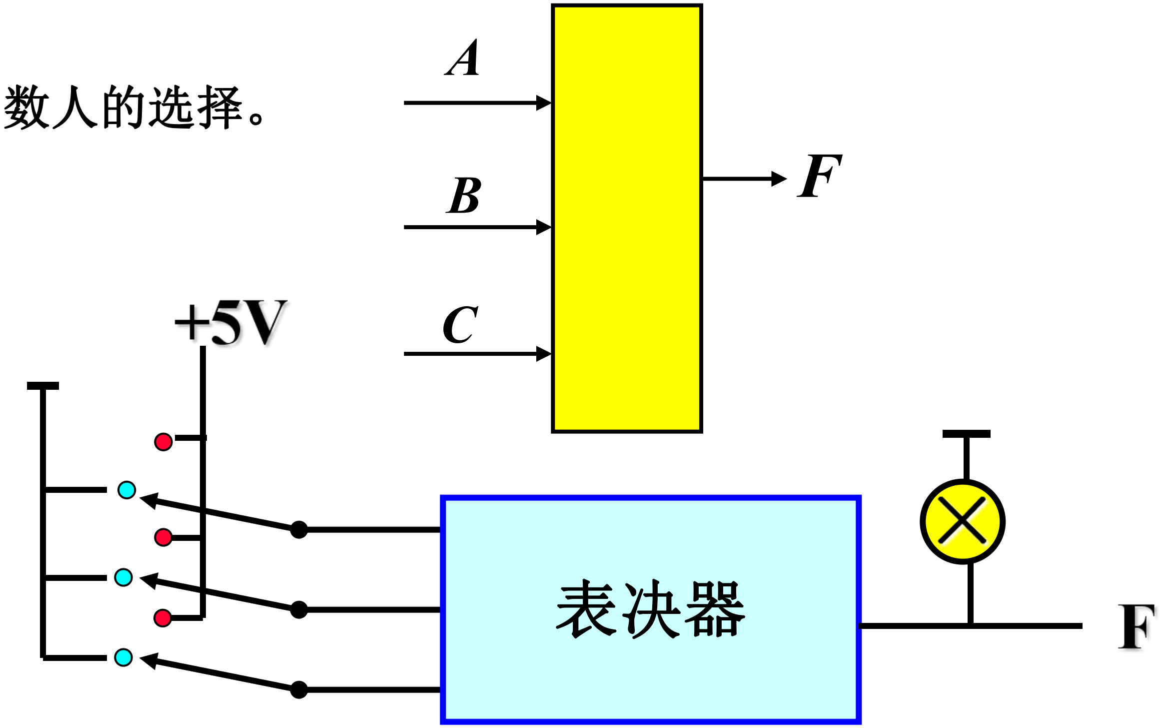
Digital Logic Design

张春慨
School of Computer Science
ckzhang@hit.edu.cn

Example 1

三人表决器设计

少数服从多数，结果为多数人的选择。

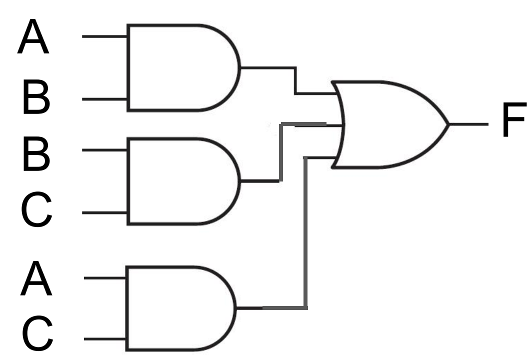


真值表

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

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

$$F=AB+AC+BC$$



Example 2

举重比赛裁判电路设计

- 一个主裁判，两个副裁判
- 比赛结果用红、绿两只灯显示 ⇨

> - 两灯都亮：成功
> - 只有红灯亮：需讨论
> - 其他：未成功

规则

1. 红绿两只灯都亮：
   - 三个裁判均按下自己的按钮；
   - 两个裁判（其中有一个是主裁判）按下自己的按钮；
2. 只红灯亮：
   - 两个裁判（均是副裁判）；
   - 只一个主裁判按下自己的按钮；
3. 其它情况，红绿灯都灭

Example 2

Truth Table

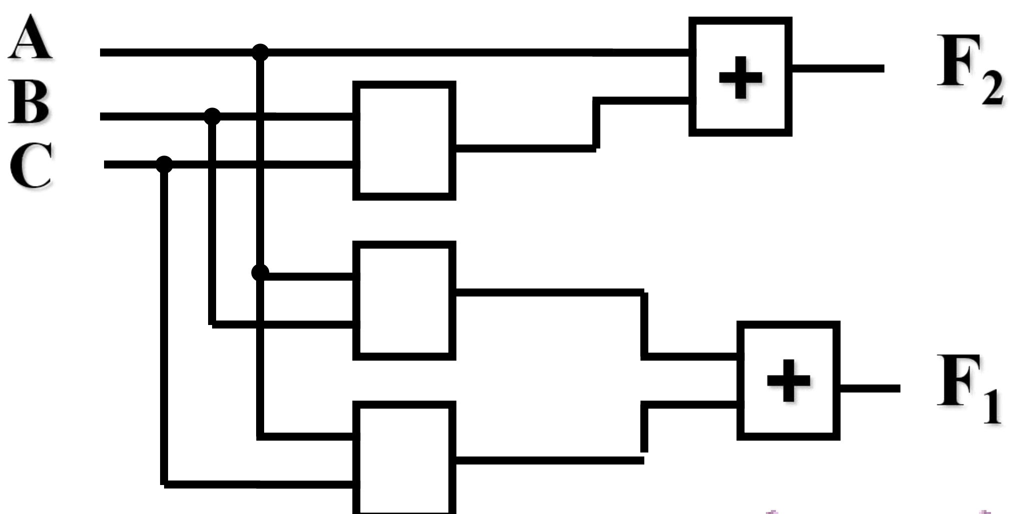
A B C 主付付	F_2 F_1 红 绿
0 0 0	0 0
0 0 1	0 0
0 1 0	0 0
0 1 1	1 0
1 0 0	1 0
1 0 1	1 1
1 1 0	1 1
1 1 1	1 1

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

$$F_2 = A + BC$$

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

$$F_1 = AB + AC$$

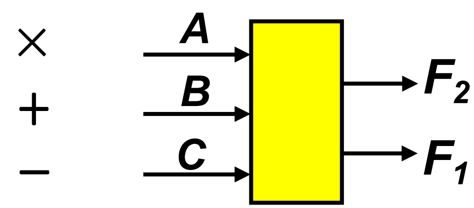


Example 3

用**与或非门**设计一个操作码形成器，当按下×、＋、－各个操作键时（**同时按无效**），要求分别产生乘法、加法、减法的操作码01、10和11

Truth Table

A (×)	B (+)	C (−)	F_2	F_1
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	×	×
1	0	0	0	1
1	0	1	×	×
1	1	0	×	×
1	1	1	×	×



Constraint: $AB=0$

$BC=0$

$AC=0$

$\overline{A}BC=0$

$A\overline{B}C=0$

$AB\overline{C}=0$

$ABC=0$

Example 3

Some Examples

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

$F_2 = B + C$

$F_2 = (B'C')'$

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

$F_1 = A + C$

$F_1 = (A'C')'$

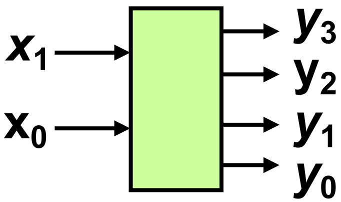
Some Examples

Example 4

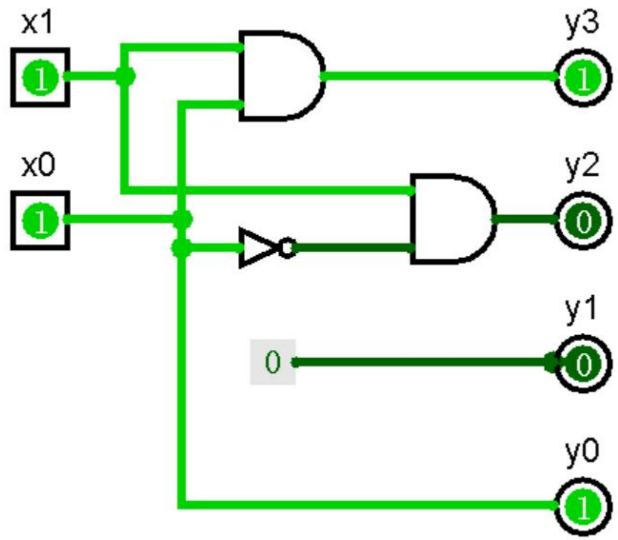
X is 2-bit binary, Design a circuit to realize $Y=X^2$

真值表

x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1



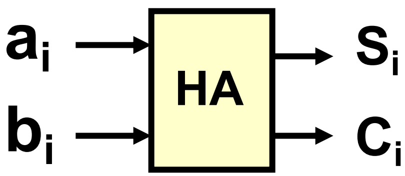
$$\begin{cases} y_3 = x_1 x_0 \\ y_2 = x_1 \bar{x}_0 \\ y_1 = 0 \\ y_0 = x_0 \end{cases}$$



Example 5

半加器（Half adder）

功能：对两个1位二进制数执行相加运算

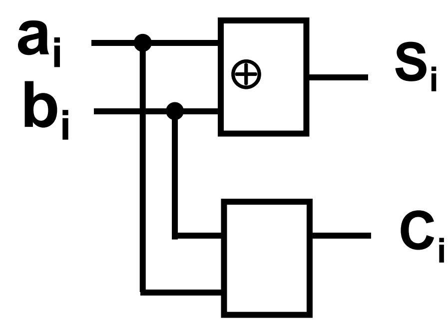


$$S_i = a_i \oplus b_i$$

$$C_i = a_i b_i$$

真值表

a_i	b_i	S_i	C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

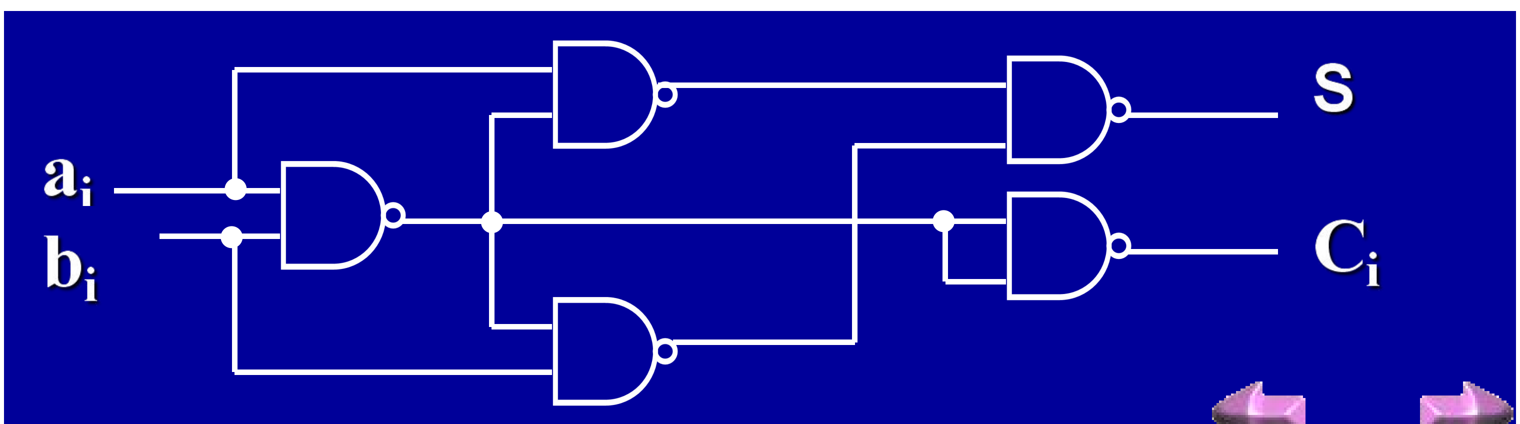


Example 5

Some Examples

利用单一逻辑门与非门实现半加器

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



Some Examples

全加器（Full adder）

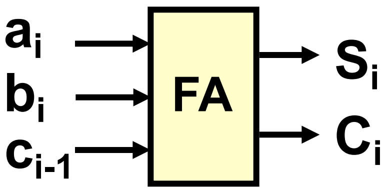
$$
\begin{array}{rcccccl}
 & 1 & 0 & 1 & 1 & \ldots\ldots\ldots\ldots & A \\
 & 1 & 1 & 1 & 0 & \ldots\ldots\ldots\ldots & B \\
+ & & & & 0 & \ldots\ldots\ldots\ldots & C_{i-1} \\
\hline
 & & & & & \ldots\ldots\ldots\ldots & S_i
\end{array}
$$

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

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

Some Examples

Example 6



S_i

a_i \ $b_i c_{i-1}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

C_i

a_i \ $b_i c_{i-1}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

a_i	b_i	C_{i-1}	S_i	C_i
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

$$S_i = \bar{a}_i\bar{b}_i c_{i-1} + \bar{a}_i b_i \bar{c}_{i-1} + a_i \bar{b}_i \bar{c}_{i-1} + a_i b_i c_{i-1}$$

$$=(\bar{a}_i\bar{b}_i + a_i b_i)\, c_{i-1} + (\bar{a}_i b_i + a_i \bar{b}_i)\, \bar{c}_{i-1}$$

$$=\overline{(a_i \oplus b_i)}\, c_{i-1} + (a_i \oplus b_i)\, \bar{c}_{i-1}$$

$$= a_i \oplus b_i \oplus C_{i-1}$$

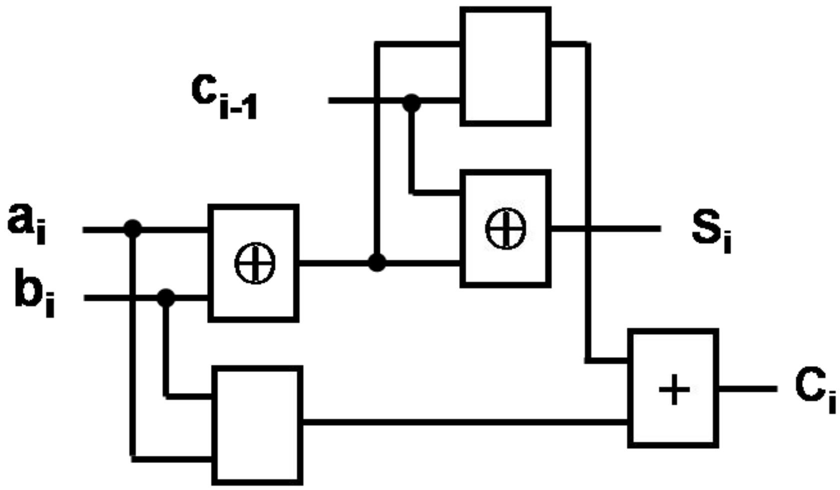
$$C_i = (a_i \oplus b_i)\, C_{i-1} + a_i b_i$$

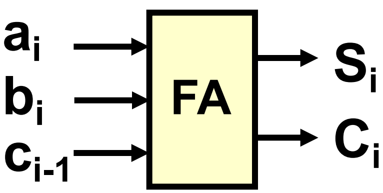
Some Examples

Example 6

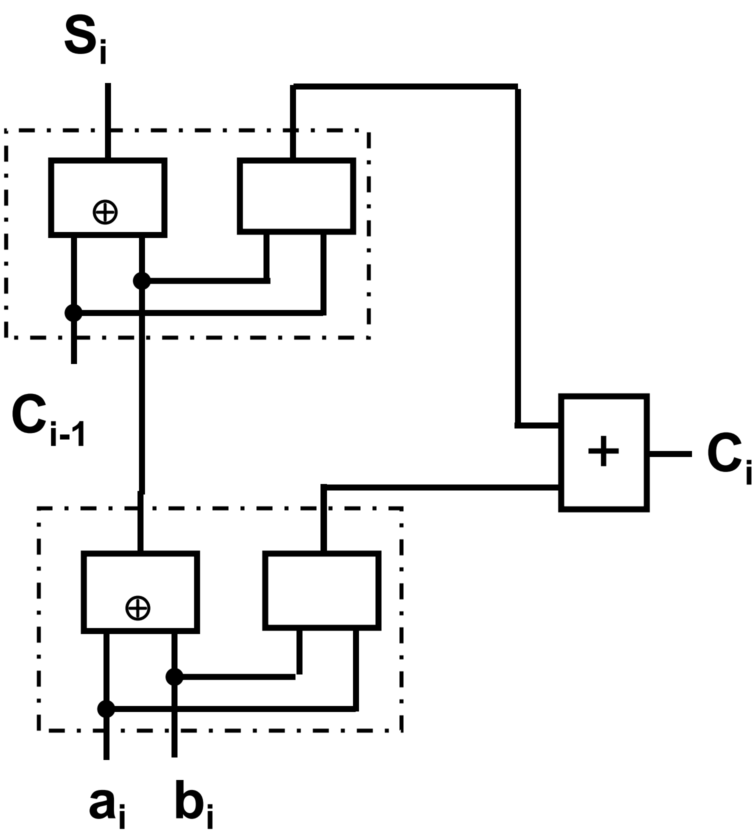
- **solution 1:**

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





Some Examples



- **solution 2：**

$$S_i = a_i \oplus b_i \oplus C_{i-1}$$

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

$$\begin{cases} S_i = a_i \oplus b_i \\ C_i = a_i b_i \end{cases}$$

Some Examples

Example 6

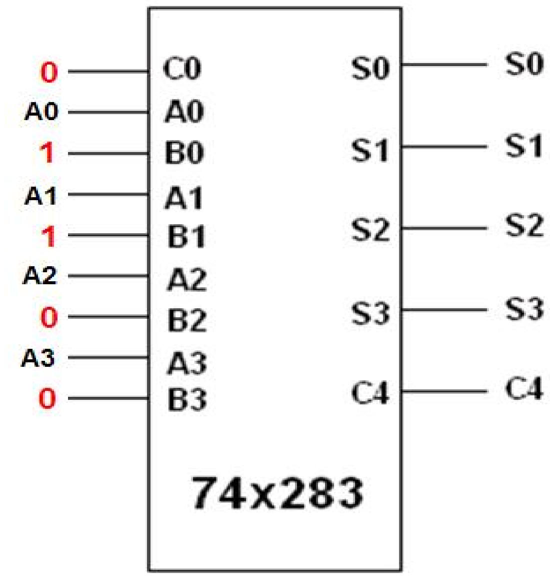
全加器（Full adder）

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

Example 7

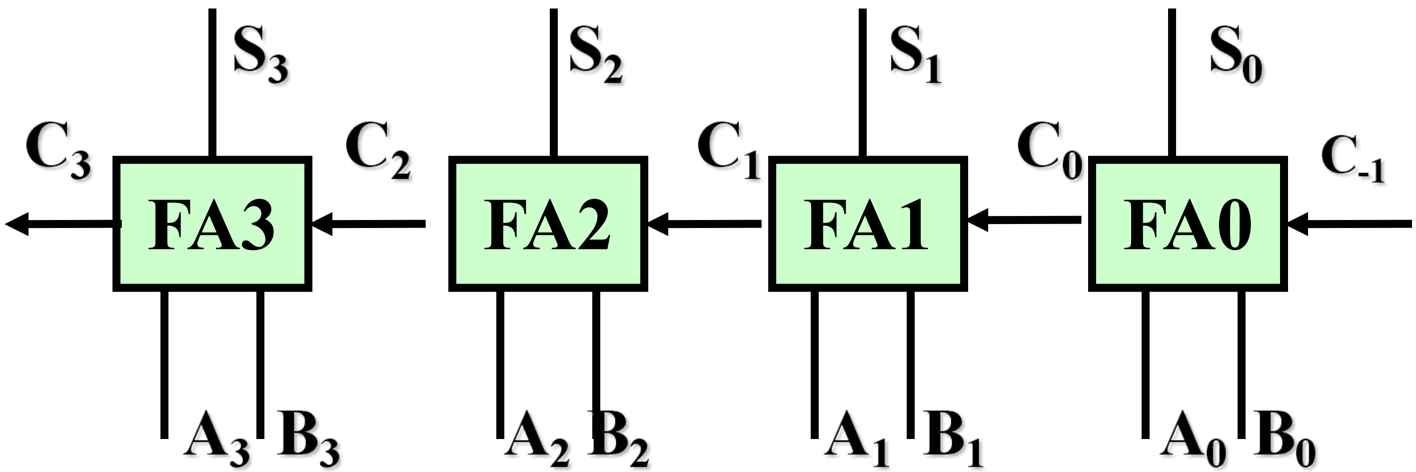
Some Examples

4位并行加法器

$S_i = a_i \oplus b_i \oplus C_{i-1}$

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

（1）串行进位



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

Example 7

Some Examples

（2） 超前进位

$A = A_3A_2A_1A_0 = 1011$

$B = B_3B_2B_1B_0 = 1110$

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

$$C_i=P_iC_{i-1} + G_i$$
$$P_i= A_i \oplus B_i$$
$$G_i= A_iB_i$$

——进位迭代公式

$$C_0= P_0C_{-1}+G_0$$

$$C_1= P_1C_0+G_1 = P_1P_0C_{-1}+P_1G_0+G_1$$

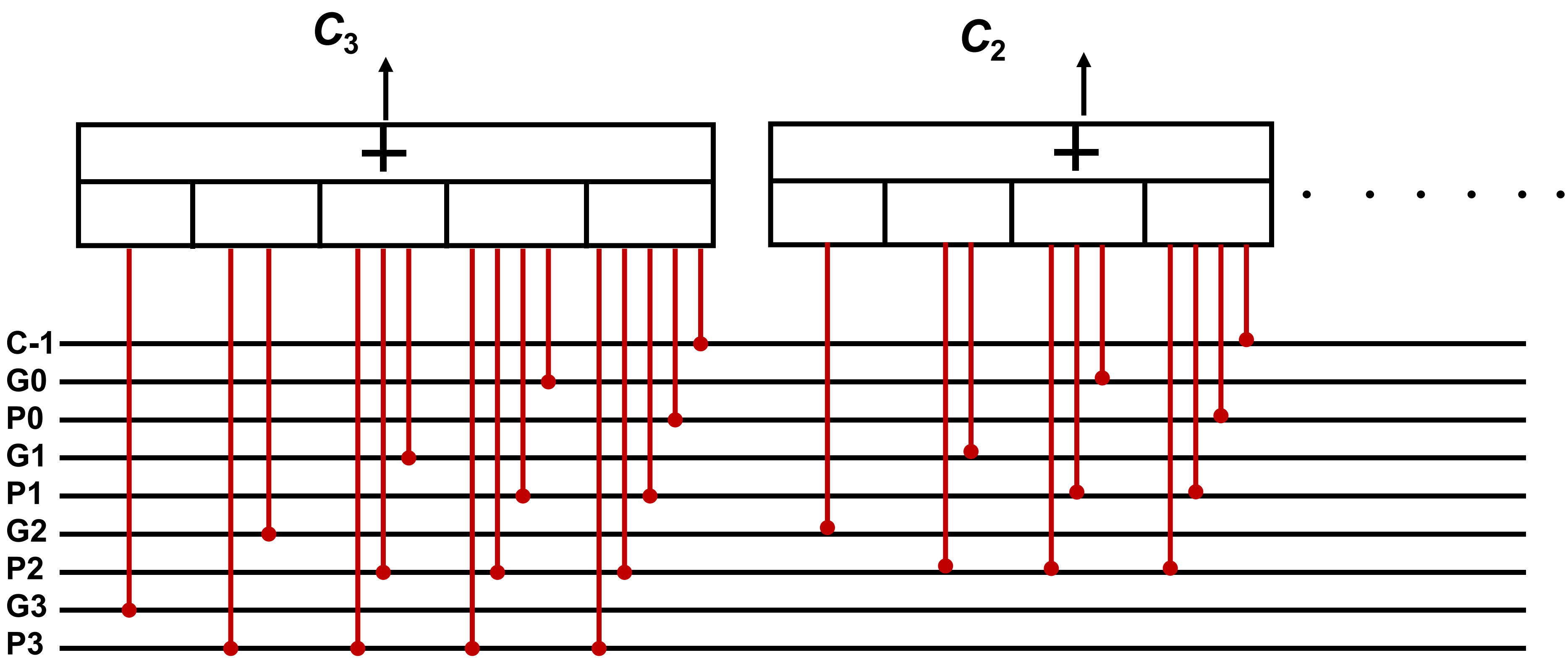
$$C_2= P_2C_1+G_2 = P_2P_1P_0C_{-1}+P_2P_1G_0+P_2G_1+G_2$$

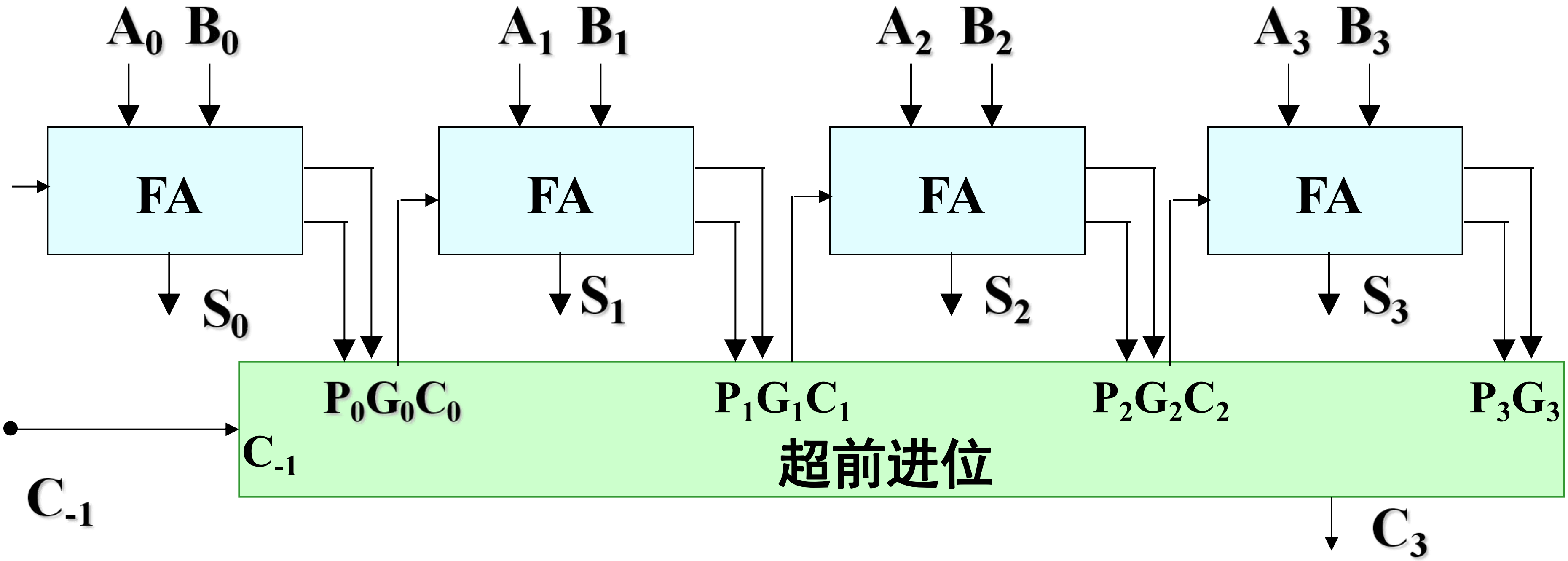
$$C_3= P_3C_2+G_3 = P_3P_2P_1P_0C_{-1}+P_3P_2P_1G_0+P_3P_2G_1+P_3G_2+G_3$$

Example 7

Some Examples

（2） 超前进位





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

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

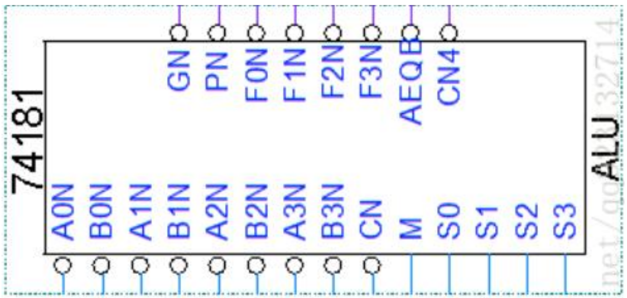
$$C_1 = P_1 C_0 + G_1 = P_1 P_0 C_{-1} + P_1 G_0 + G_1$$

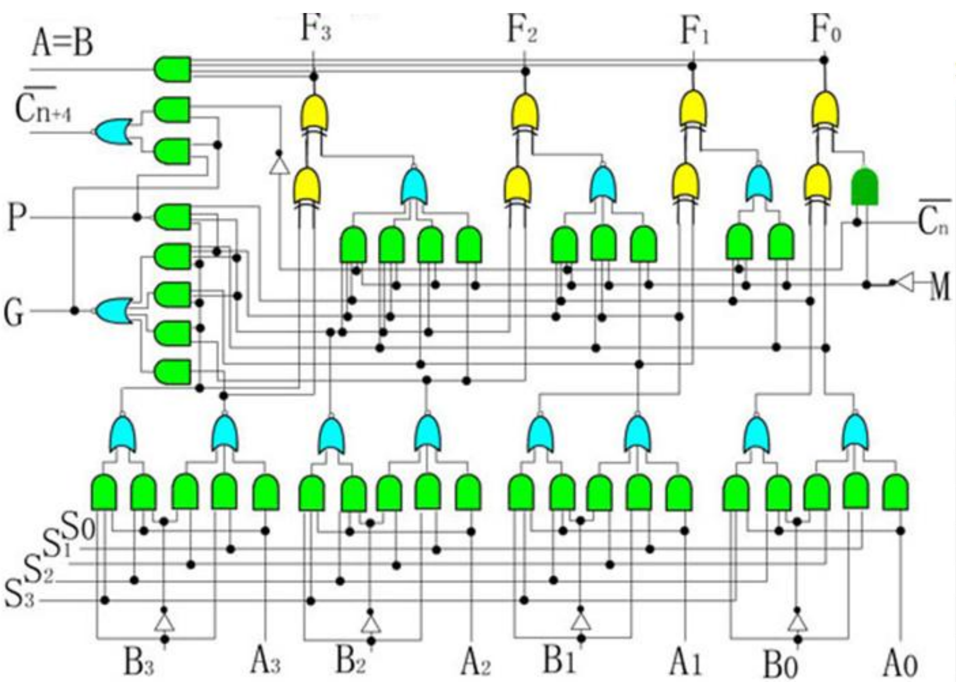
$$C_2 = P_2 C_1 + G_2 = P_2 P_1 P_0 C_{-1} + P_2 P_1 G_0 + P_2 G_1 + G_2$$

$$C_3 = P_3 C_2 + G_3 = P_3 P_2 P_1 P_0 C_{-1} + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3$$

ALU

74LS181 ALU是主要进行算术和逻辑运算的电路，可以作为处理器进行运算的核心部件。它对两个4位操作数进行逻辑或者算术运算等。





***M*=0， addition**

***M*=1， logic operation**

74ls181芯片总共有22个引脚。

数据引脚

- 8个数据输入端，A0m、A1n、A2n、A3n，B0n、B1n、B2n、B3n，（其中A3和B3是高位）。
- 4个二进制输出端F0、F1、F2、F3，以四位二进制形式输出运算的结果。
- CN端处理进入芯片前进位值，CN4记录运算后的进位。
- GN先行进位产生端。PN先行进位传递函数。

控制引脚

- 4个控制端，S0、S1、S2、S3，控制两个四位输入数据的运算，例如加、减、与、或。
- M控制芯片的运算方式，包括算术运算和逻辑运算。

功能表

S3	S2	S1	S0	M = H Logic Functions	M = L Arithmetic Operations	
					/Cn = H	/Cn = L
L	L	L	L	F = /A	F = A	F = A plus 1
L	L	L	H	F = /(A + B)	F = A + B	F = (A + B) plus 1
L	L	H	L	F = (/A)B	F = A + /B	F = (A + /B) plus 1
L	L	H	H	F = 0	F = minus 1 (2s Comp)	F = ZERO
L	H	L	L	F = /(AB)	F = A plus A(/B)	F = A plus A(/B) plus 1
L	H	L	H	F = /B	F = (A + B) plus A(/B)	F = (A + B) plus A(/B) plus 1
L	H	H	L	F = A xor B	F = A minus B minus 1	F = A minus B
L	H	H	H	F = A(/B)	F = A(/B) minus 1	F = A(/B)
H	L	L	L	F = /A+ B	F = A plus AB	F = A plus AB plus 1
H	L	L	H	F = /(A xor B)	F = A plus B	F = A plus B plus 1
H	L	H	L	F = B	F = (A + /B) plus AB	F = (A + /B) plus AB plus 1
H	L	H	H	F = AB	F = AB minus 1	F = AB
H	H	L	L	F = 1	F = A plus A	F = A plus A plus 1
H	H	L	H	F = A + /B	F = (A + B) plus A	F = (A + B) plus A plus 1
H	H	H	L	F = A + B	F = (A + /B) plus A	F = (A + /B) plus A plus 1
H	H	H	H	F = A	F = A minus 1	F = A

注：+ 是或的意思 /是非 plus是加 xor是异或

Example 8

Some Examples

全减器（Binary Full Subtracter）

1 1 1 0 .... A

1 0 1 1 .... B

\- 0 0 .... C_{i-1}

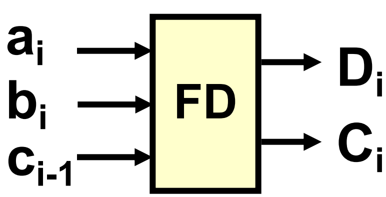
.... D_i

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

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

真值表

a_i	b_i	C_{i-1}	D_i	C_i
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



Some Examples

Example 8

真值表

a_i	b_i	C_{i-1}	D_i	C_i
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

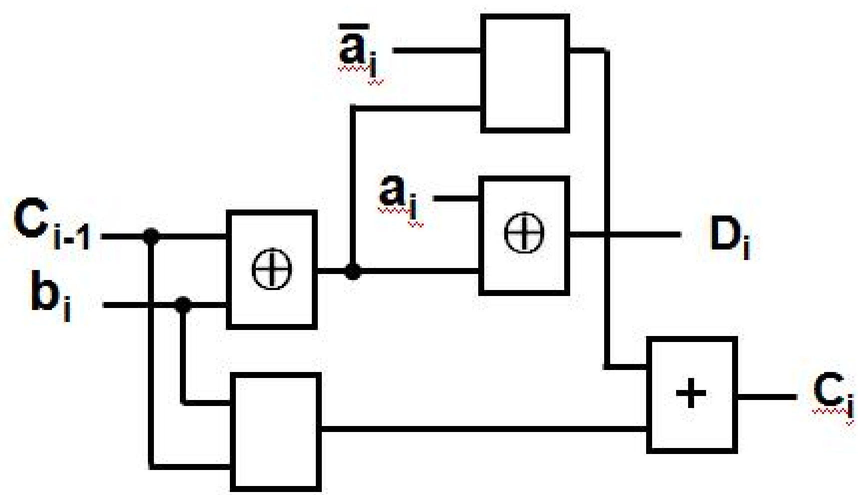
D_i

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

C_i

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

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



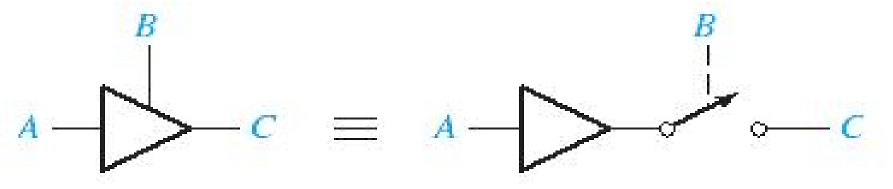
Some Examples

Example 9

三态门（Three-State Buffers）

三态——

- **0**
- **1**
- **Z: 高阻态**



三态门（恒等）

B：使能端，高电平有效

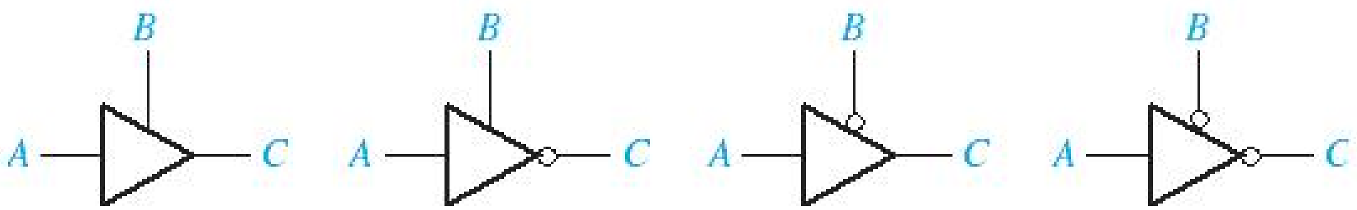
真值表

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

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

Some Examples

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)

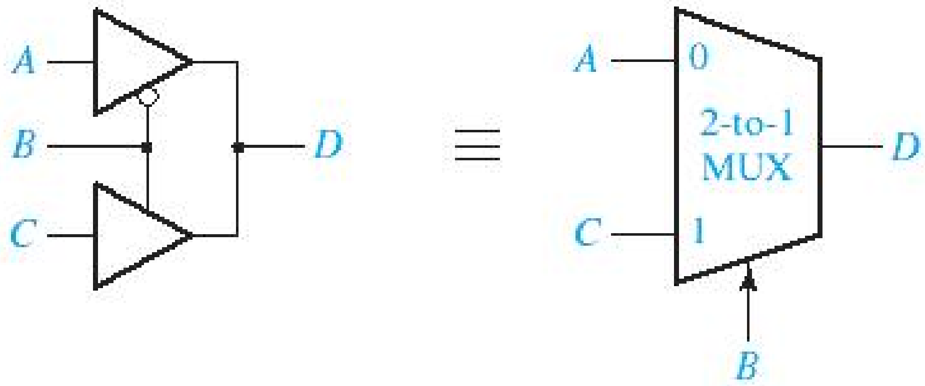
B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

(d)

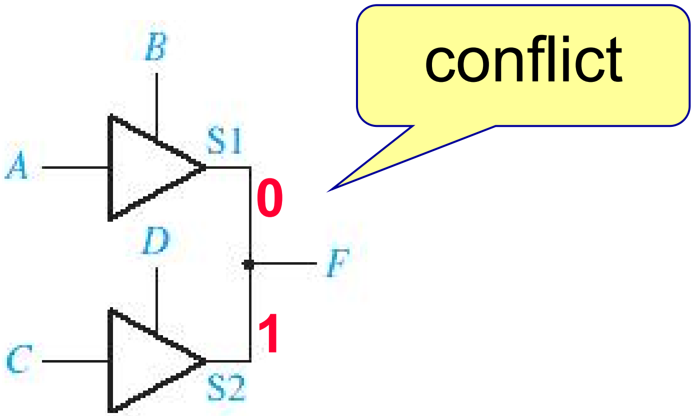
理解三态门——

- **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。（实际电路中你不可能去断开它）**

Three-State Buffers



D=B'A+BC

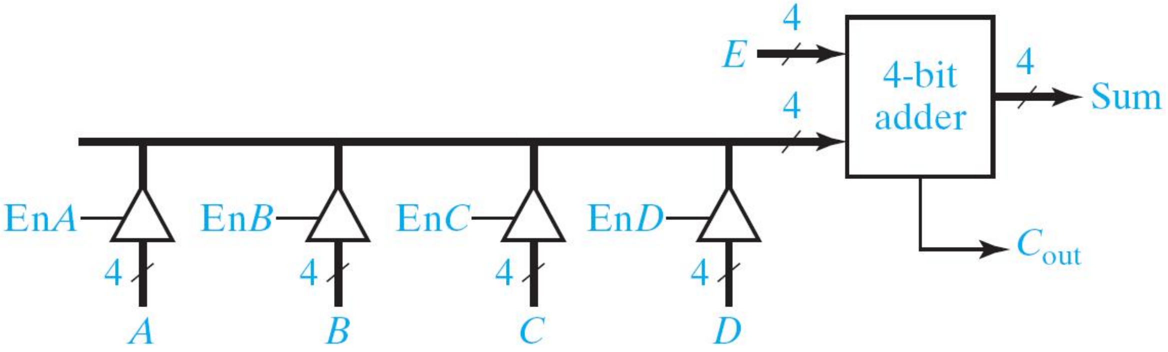


S_1 \ S_2	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

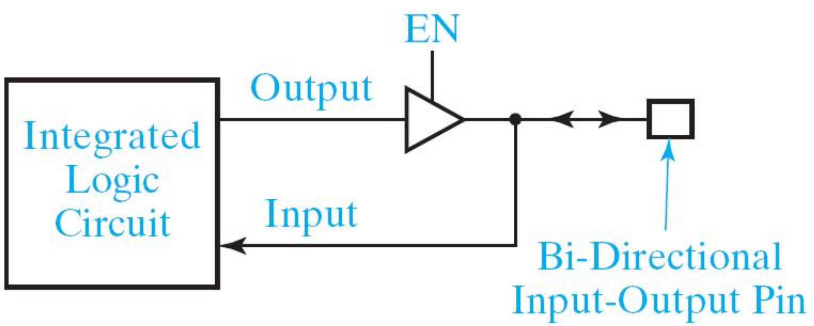
Three-State Buffers

应用

- 三态总线



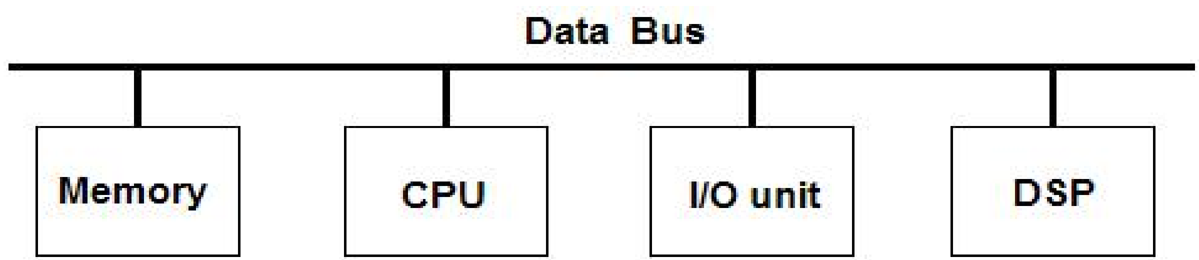
- 管脚输入输出可编程

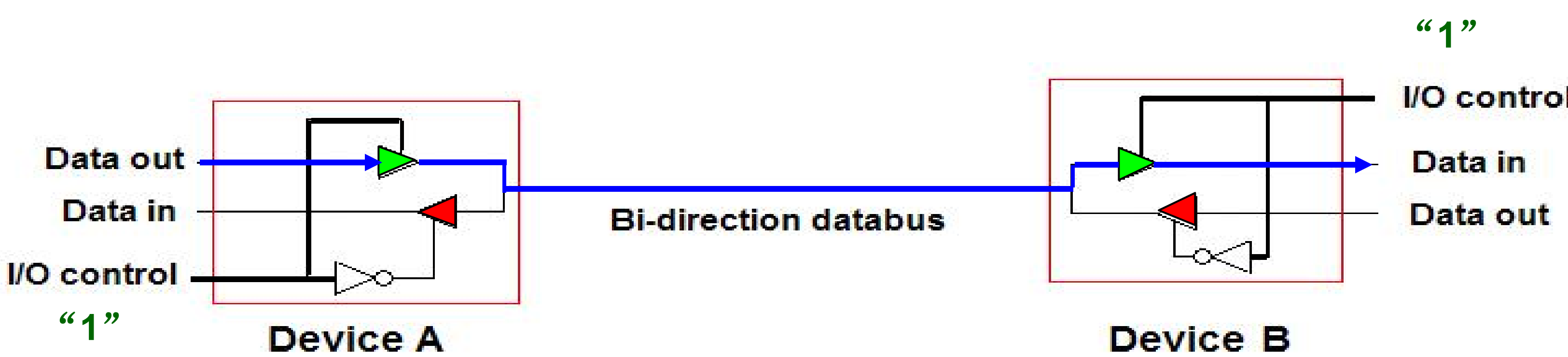


Three-State Buffers

应用

- 双向数据总线





Three-State Buffers

理解三态门——

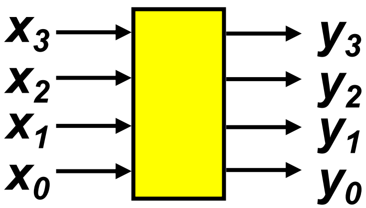
内存里的一个存储单元

- 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

Three-State Buffers

Example

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	×

Three-State Buffers

② 化简

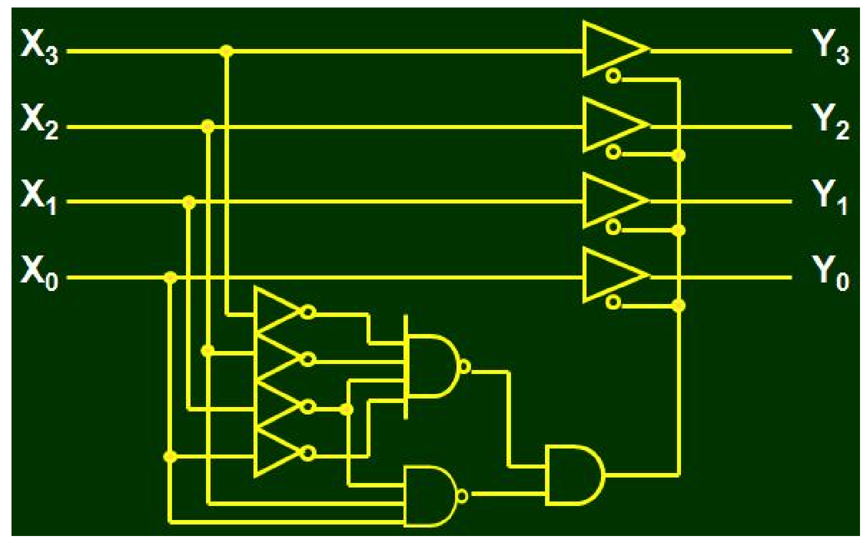
x_3x_2 \ x_1x_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	×	×	×	×
10	0	0	×	×

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

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

③ 逻辑图

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



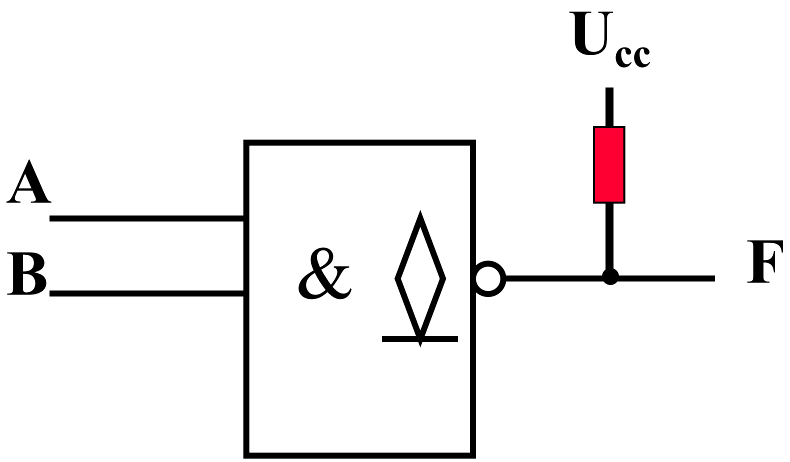
Example 10

Some Examples

OC门 (Open Collector Gate)

- 几个OC门的**输出端**可以**直接互连**
- 使用时必须加**负载电阻**

$$F=\overline{AB}$$



Example 11

Some Examples

OC门 (Open Collector Gate)

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