4.2 加法器

4.2.1 二进制数的算术运算

一、两个绝对值之间的运算

运算规则，加法："逢二进一"；减法："借一当二"。

加、减、乘、除。

1001+0101=?　　1001−0101=?

1011×0101=?　　11001÷101=?

乘法运算可以通过连续的加法运算来实现；除法运算可以通过连续的减法运算来实现。如 9×3, 9÷3.

4.2 加法器

4.2.1　二进制数的算术运算

一、原码、反码和补码

（1）原码=符号---数值。（‘0’正‘1’负）

+1010101的原码？ - 1010101的原码？

（2）反码表示

正数的反码=原码；
负数的反码=符号---数值按位取反。

+1010101的反码？ - 1010101的反码？

4.2 加法器

4.2.1 二进制数的算术运算

一、原码、反码和补码

（3）补码表示

正数的补码=原码；
负数的补码=反码+1。

+1010101的补码？ - 1010101的补码？

-1的8位二进制原码、反码、补码？

4.2 加法器

4.2.1 二进制数的算术运算

一、原码、反码和补码

补码运算规则：补码的补码是原码；两数补码之和等于两数之和的补码。即

$[补码]_{补码}=原码$；

$[X_1]_{补码}+[X_2]_{补码}=[X_1+X_2]_{补码}$

减法运算电路结构复杂，运算速度慢；相对而言，加法电路结构简单，运算速度快。故常将减一个数变为加上一个负数，负数用补码表示就可以将减法变为加法运算。

4.2 加法器

4.2.1 二进制数的算术运算

用补码运算： 1101-1010=？ 0110-1001=？

注意：在进行补码运算时，必须在其相应位数表示的数值范围内进行，否则会产生错误的运算结果。

综上所述，二进制数的加、减、乘、除运算都可以通过加法运算来实现。

实现加法运算的器件称为加法器，它是计算机重要的组成单元。

4.2 加法器和数值比较器

4.2.2 加法器

一、半加器和全加器

1. 半加器（Half Adder）

两个 1 位二进制数相加。

$A_i + B_i = S_i$ (和) $\rightarrow C_i$ (进位)

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

函数式

$$S_i = \overline{A}_i B_i + A_i \overline{B}_i$$
$$= A \oplus B$$
$$C_i = A_i B_i$$

半加器（Half Adder）

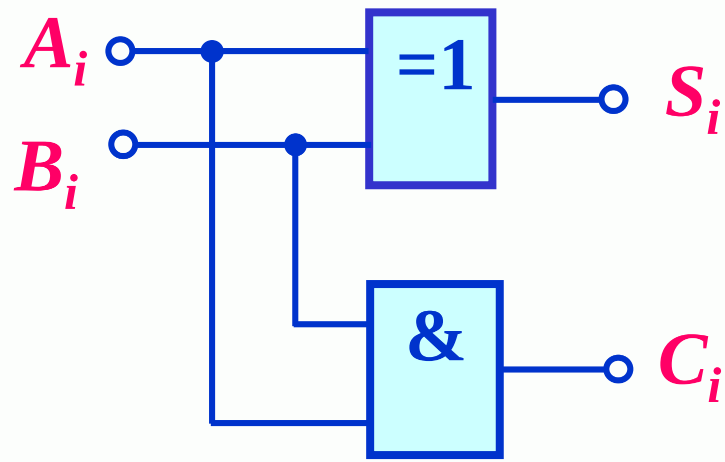
函数式

$$S_i = \overline{A_i}B_i + A_i\overline{B_i}$$

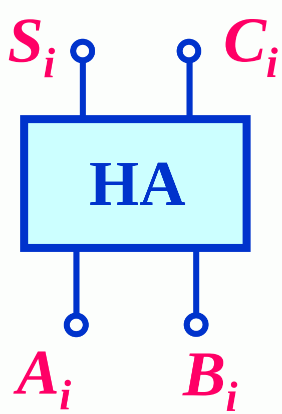
$$= A \oplus B$$

$$C_i = A_iB_i$$

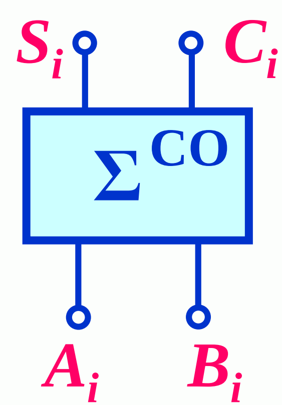
逻辑图



曾用符号



国标符号



2. 全加器（Full Adder）

两个 1 位二进制数相加，考虑低位进位。

$A_i + B_i + C_{i-1}$ (低位进位)

$= S_i$ (和) $\rightarrow C_i$ (向高位进位)

真值表

A	B	C_{i-1}	S_i	C_i	A	B	C_{i-1}	S_i	C_i
0	0	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	1	1	1

标准与或式

$$S_i = \overline{A_i}\,\overline{B_i}C_{i-1} + \overline{A_i}B_i\overline{C_{i-1}} + A_i\overline{B_i}\,\overline{C_{i-1}} + A_iB_iC_{i-1}$$

$$C_i = \overline{A_i}B_iC_{i-1} + A_i\overline{B_i}C_{i-1} + A_iB_i\overline{C_{i-1}} + A_iB_iC_{i-1}$$

全加器（Full Adder）

卡诺图

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

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

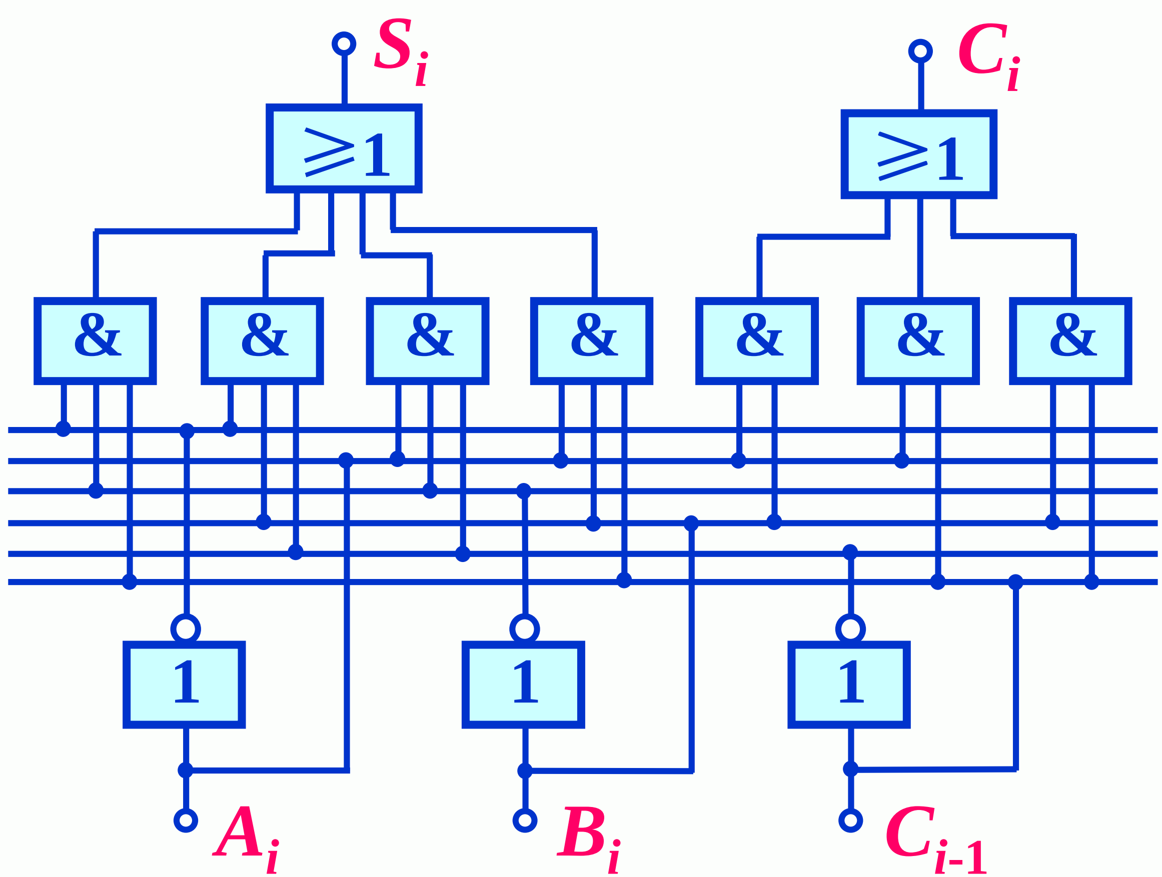
最简与或式

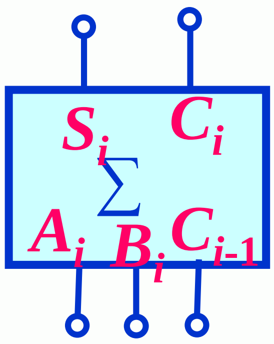
圈 “1”
$$\begin{cases} S_i = \overline{A}_i\overline{B}_iC_{i-1} + \overline{A}_iB_i\overline{C}_{i-1} + A_i\overline{B}_i\overline{C}_{i-1} + A_iB_iC_{i-1} \\ C_i = A_iB_i + A_iC_{i-1} + B_iC_{i-1} \end{cases}$$

逻辑图

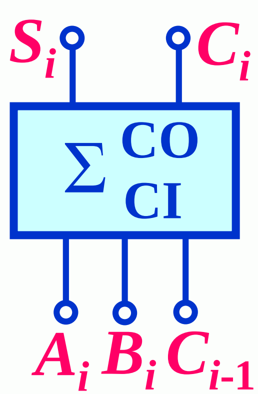
曾用符号

(a) 用与门、或门和非门实现



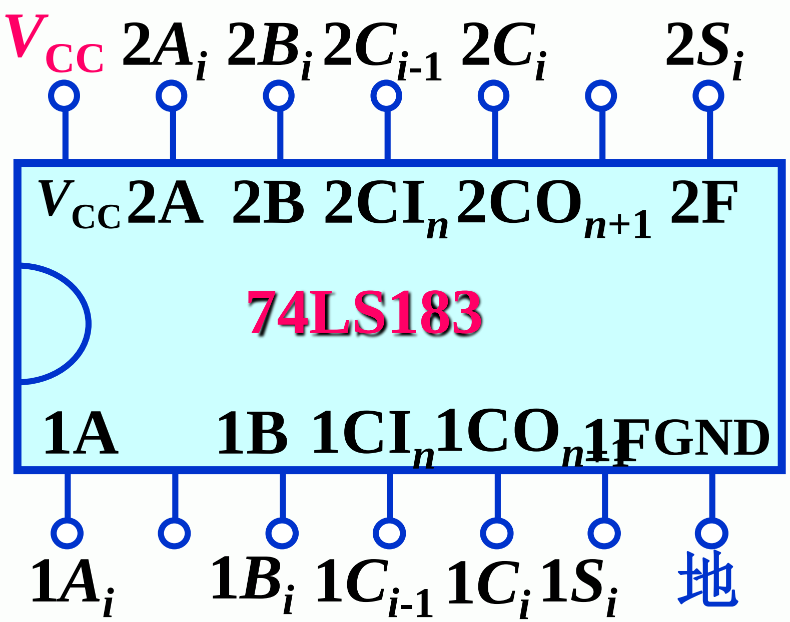


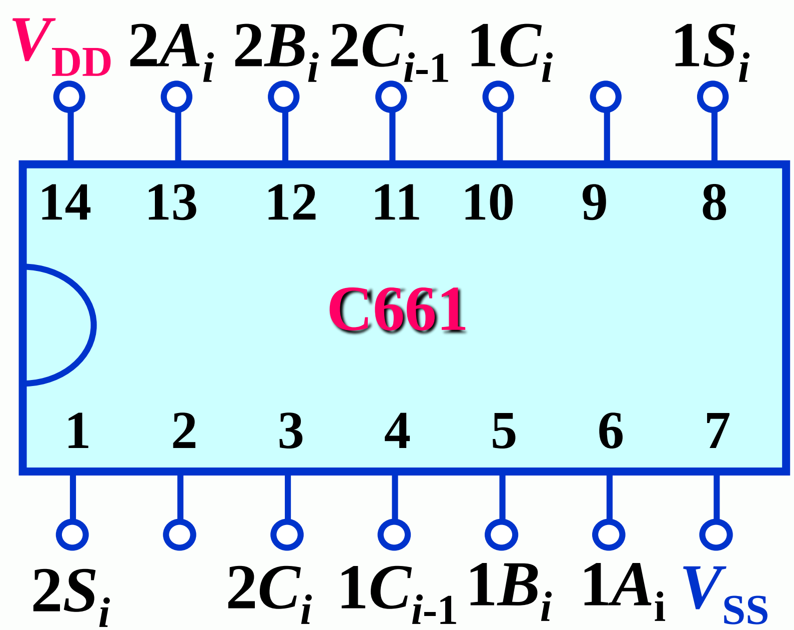
国标符号



3. 集成全加器

双全加器 { TTL：74LS183 / CMOS：C661 }

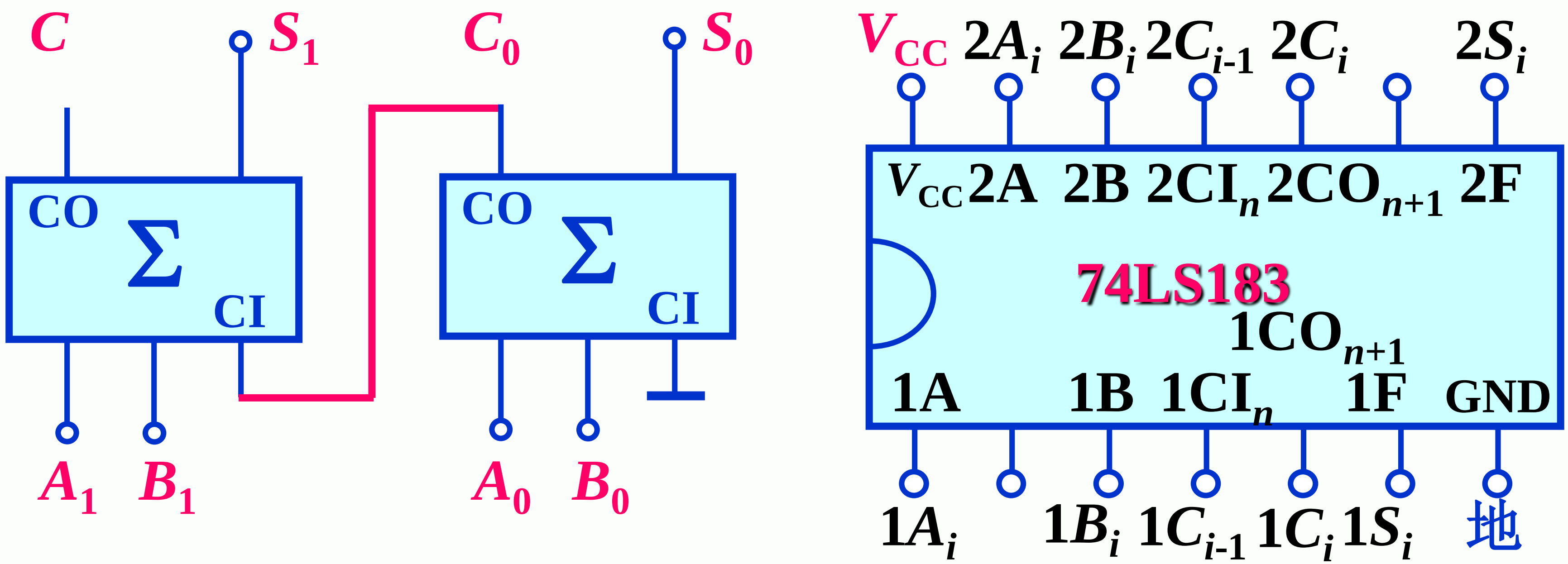




思考

- 如何用全加器实现两位二进制加法。

$$A_1A_0 + B_1B_0 = CS_1S_0$$

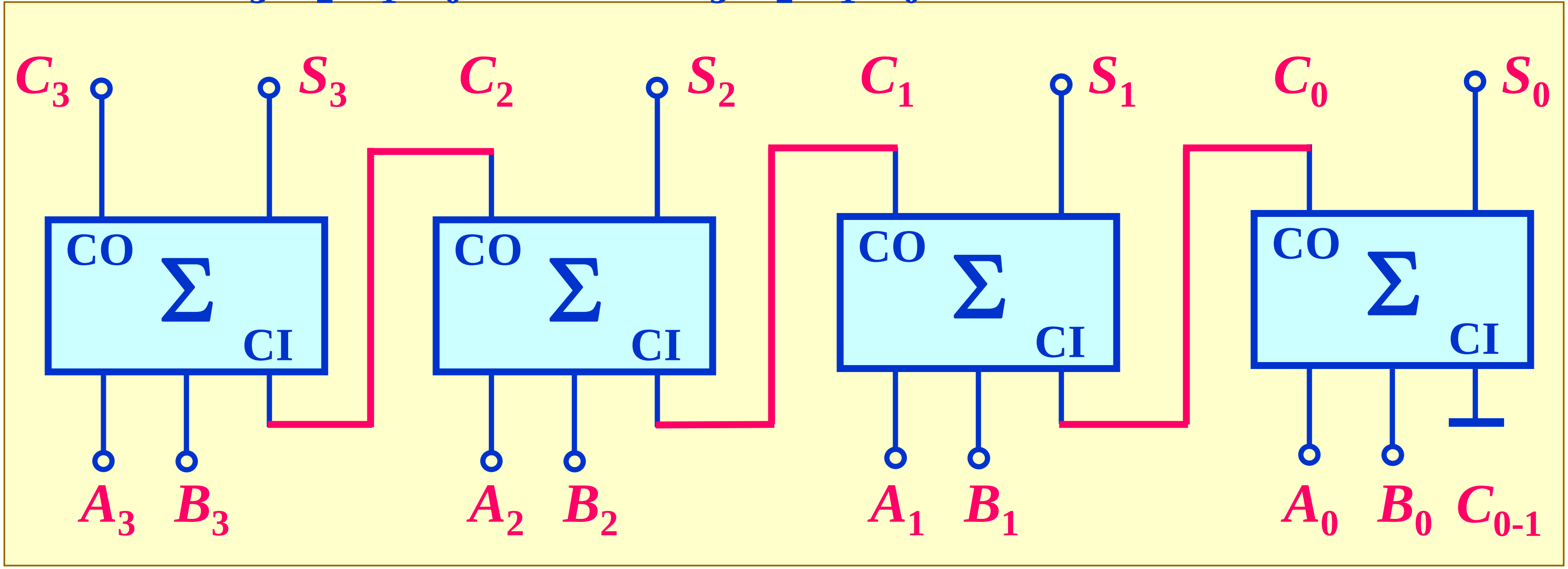


二、加法器（Adder）

实现多位二进制数相加的电路

1. 4 位串行进位加法器

$A = A_3A_2A_1A_0 \quad B = B_3B_2B_1B_0$



特点：
- 电路简单，连接方便
- 速度低 = 4 t_{pd}

t_{pd} — 1位全加器的平均传输延迟时间

2. 超前进位加法器

作加法运算时，各个位置上的进位信号由输入二进制数直接产生。

$$C_0 = A_0B_0 + (A_0 + B_0)C_{0-1}$$

$$C_1 = A_1B_1 + (A_1 + B_1)C_0$$

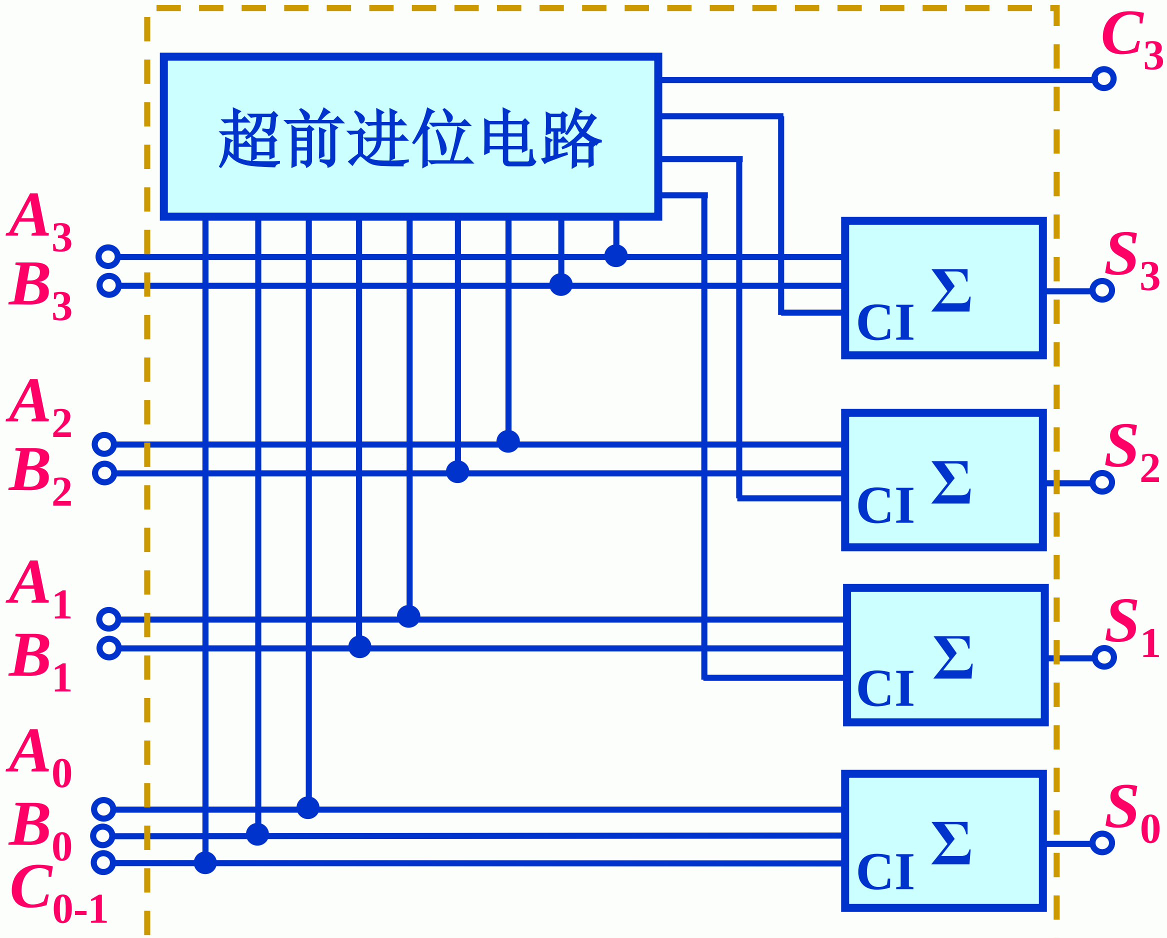
$$= A_1B_1 + (A_1 + B_1)\left[A_0B_0 + (A_0 + B_0)C_{0-1}\right]$$

$$\vdots$$

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

即各进位信息均是输入A、B和C_{0-1}的函数。

集成芯片 CMOS：CC4008
TTL：74283 74LS283

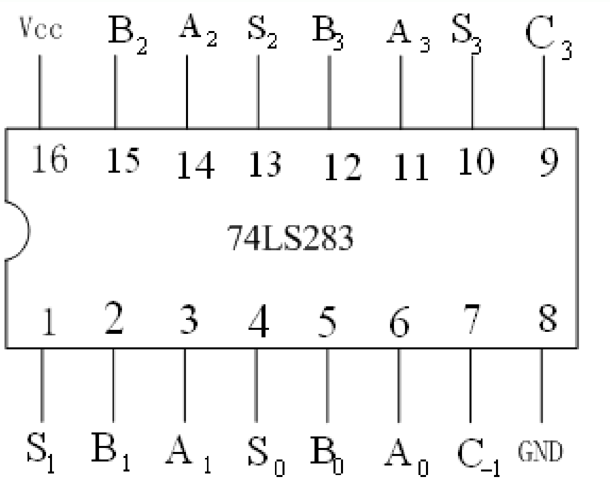


逻辑结构示意图

优点：运算快

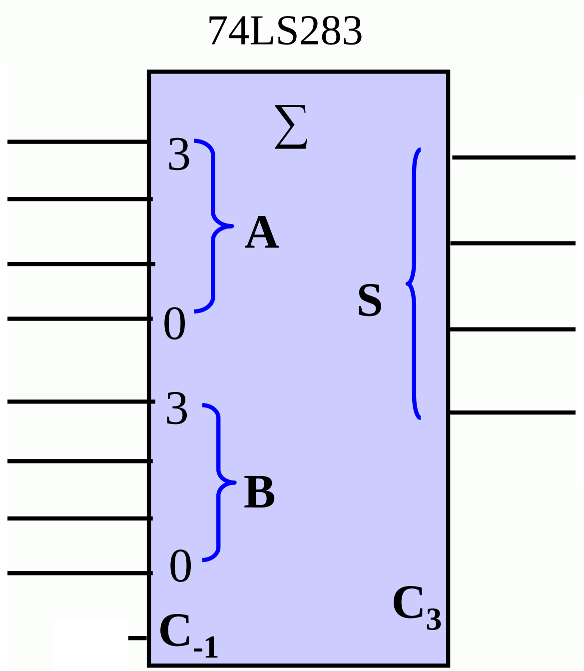
缺点：电路复杂

加法器应用—代码转换



4位二进制超前进位加法器

8421BCD—余3码



BCD码到余3码的转化电路