# 第 2 章 运算方法和运算器

主要内容：

- 数据与文字的表示方法
- 定点的加、减法运算
- 定点的乘法运算
- 定点的除法运算
- 定点的运算器的组成
- 浮点运算方法和浮点运算器

# 2.2 定点加法、减法运算

## 2.2.1 补码加法

- 补码加法规则

两个相加的数无论正负，其和的补码等于两数补码之和：$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

例12 设X=+1001，Y=+0101，用补码求Z=X+Y。

解：$[X]_{补}=01001$，$[Y]_{补}=00101$；（P27例11）

$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

$=01001+00101=01110$

符号位一同参与运算

故：X+Y=01110

例13 设X=0.1001，Y=-0.0101，用补码求Z=X+Y。

解：$[X]_{补}=0.1001$，$[Y]_{补}=1.1011$；

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$
$$=0.1011+1.1011$$
$$=0.0110$$

故：$X+Y=+0.011$

计算前，需明确字长。根据补码模的含义，超过字长的进位需抛弃！

# 2.2.2 补码减法

- 补码减法规则

两个相减的数无论正负：

$$[X-Y]_{补} = [X+(-Y)]_{补} = [X]_{补} + [-Y]_{补}$$

注意：$[-Y]_{补}$为$[Y]_{补}$的机器负数,转换方法:将连同符号位一起变反,末位加1。

例14　$x=+1101$, $y=+0110$,求 $x-y$。

（P28例14）

解:

$[x]_{补}=01101$　　$[y]_{补}=00110$,　　$[-y]_{补}=11010$

$$\begin{array}{lr}
[x]_{补} & 01101 \\
+[-y]_{补} & 11010 \\
\hline
[x-y]_{补} & 100111
\end{array}$$

$x-y=+0111$

## 补码加减运算规则小结

- 参加运算的操作数用补码表示。
- 符号位参加运算。
- 若指令操作码为加，则两数直接相加；若操作码为减，则将减数连同符号位一起变反加1后再与被减数相加。
- 运算结果用补码表示。

补码使得加、减可以统一处理

例15 $x=+1011$, $y=+1001$,求 $x+y$。

（P28例15）

[解:]

$[x]_{补}=01011 \quad [y]_{补}=01001$

$$
\begin{array}{lr}
 & [x]_{补} \quad 01011 \\
+ & [y]_{补} \quad 01001 \\
\hline
 & [x+y]_{补} \quad 10100
\end{array}
$$

两个正数相加的结果成为负数——错误！

结果超出了表示范围，产生溢出！

Q：如何判断溢出？

## 2.2.3 溢出判断

- **基本规律**：两个异号数相加或两个同号数相减不会发生溢出；只有两个同号数相加或两个异号数相减才可能发生溢出。
  - 正溢：运算结果为正且大于所能表示的最大正数；
  - 负溢：运算结果为负且小于所能表示的最小负数；

- 溢出判断法：
  ① 采用一个符号位判断（最高有效位判断法）
  ② 采用双符号位法（变形补码法）

# 1、单符号位（最高有效位）判断法

两个补码数相加、减时，若最高数值位向符号位送的进位值与符号位送向更高位进位不相同，则运算结果溢出。

溢出的逻辑表达式为（**P30**）：

$$V = \overline{C_f} \bullet C_0 + C_f \bullet \overline{C_0}$$

此逻辑表达式可用异或门实现

# 2、双符号位法（变形补码法,模4补码法）

变形补码定义：

$$[x]_{补}=2^{n+2}+x$$

- 变形补码的符号用两位来表示，正数为00，负数为11。
- 变形补码的两个符号位都可以参与运算，运算结果根据两个符号位是否一致来判断是否溢出。

溢出的逻辑表达式：

（P30）

$$V=\overline{S_{f1}}\bullet S_{f2}+S_{f1}\bullet\overline{S_{f2}}$$

“01”表示正溢，“10”表示负溢，最高符号永远表示结果的正确符号。

例16 x ＝＋01100, y ＝＋01000,求 x ＋ y 。

解: $[x]_补 = 001100$, $[y]_补 = 001000$

（P28例17）

$$
\begin{array}{lr}
 & [x]_补 \quad 00\ 1100 \\
+ & [y]_补 \quad 00\ 1000 \\
\hline
 & 01\ 0100
\end{array}
$$

- 两个符号位不一致,结果溢出。

例17 x ＝－0.1100, y ＝-0.1000,求 x ＋ y 。

解: $[x]_补 = 11.0100$, $[y]_补 = 11.1000$

$$
\begin{array}{lr}
 & [x]_补 \quad 11.0100 \\
+ & [y]_补 \quad 11.1000 \\
\hline
 & 10.1100
\end{array}
$$

- 两个符号位不一致,结果溢出。

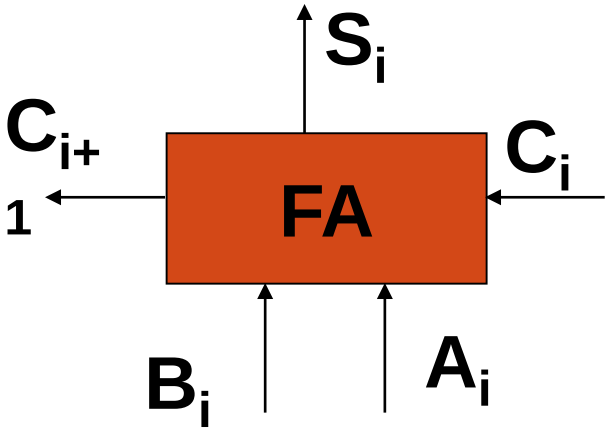
# 2.2.4 基本的二进制加减法器

## 1、1位加法器设计

$$Si = \overline{A_i}\,\overline{B_i}C_i + \overline{A_i}B_i\overline{C_i} + A_i\overline{B_i}\,\overline{C_i} + A_iB_iC_i \quad = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = \overline{A_i}B_iC_i + A_i\overline{B_i}C_i + A_iB_i\overline{C_i} + A_iB_iC_i$$

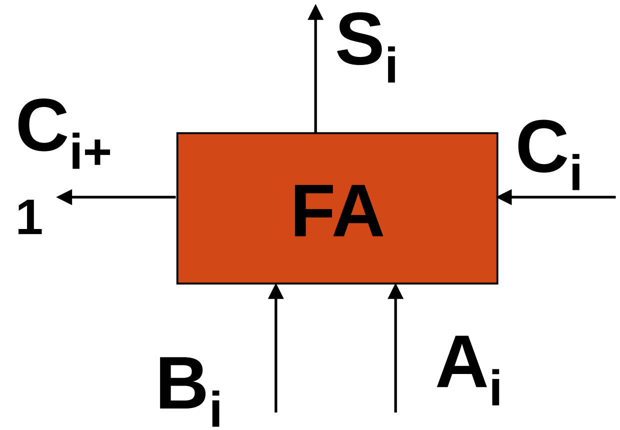
$$= A_iB_i + (A_i \oplus B_i)\,C_i$$

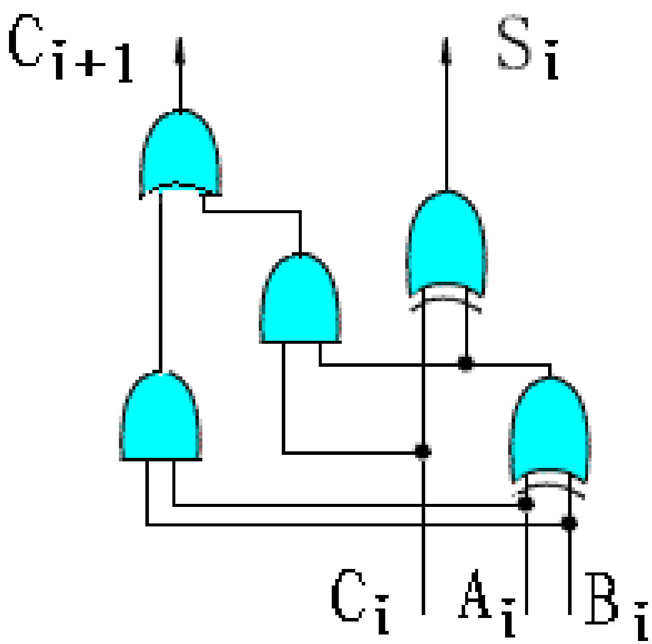



| 输入 | | | 输出 | |
|---|---|---|---|---|
| $A_i$ | $B_i$ | $C_i$ | $S_i$ | $C_{i+1}$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$$S_i = A_i \oplus B_i \oplus C_i$$

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




对于$A_i$ 、$B_i$和 $C_i$三个输入

$S_i$的时延为：3T*2=6T

$C_{i+1}$的时延为：3T+2T=5T

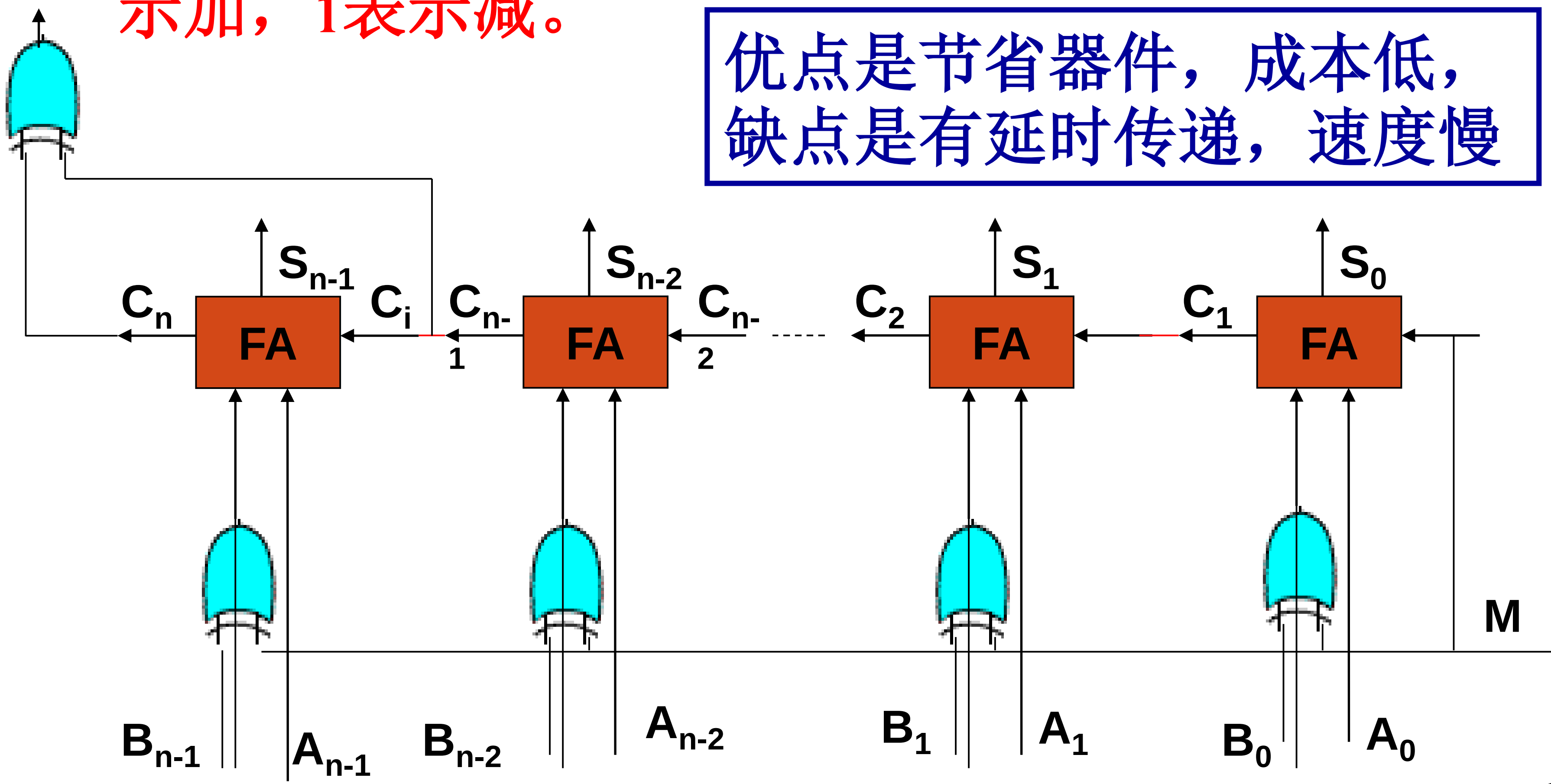
一位全加器FA逻辑电路图

Q：如何形成n位加法器？

# 2、n位加法器设计

- n位加法器可由多个1位加法器级联实现（行波进位加法器）。
- 补码减法器可由加法器实现。方式控制线M，0表示加，1表示减。

优点是节省器件，成本低，
缺点是有延时传递，速度慢



整个n位加法器的时延分析：

产生各FA的求和项时延：3T ；（相对于$A_i$ 、$B_i$ 输入）

$C_1$的时延为：5T （相对于FA三个输入）

$C_n$的时延为：2T*(n-1)=2(n-1)T；（相对于$C_1$ 输入）

V(溢出)的时延：3T（相对于$C_n$ 输入）

整个器件的时延为上述各项和（2n+9）T