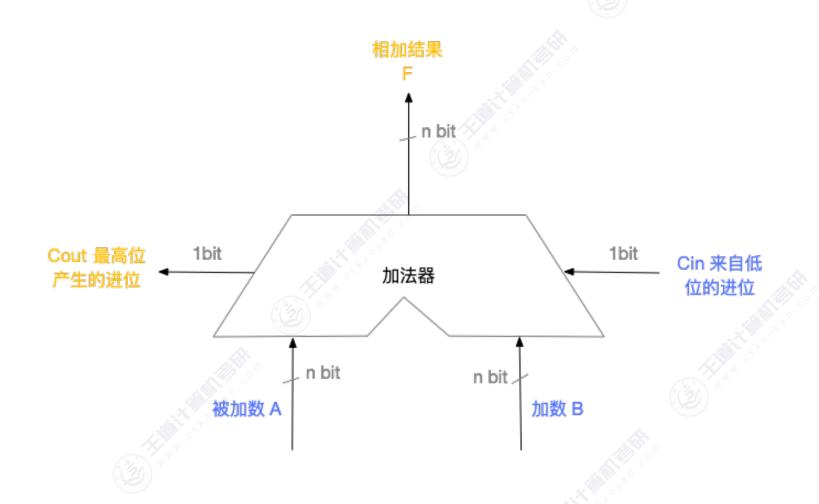


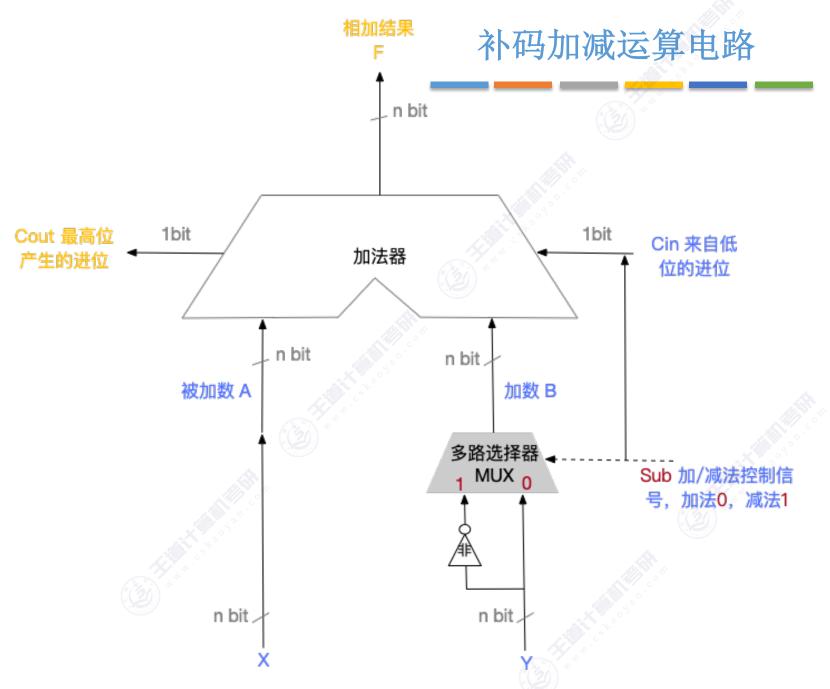
n bit 加法器



例:

A=1000,B=0111,Cin=0 则 F=1111,Cout=0

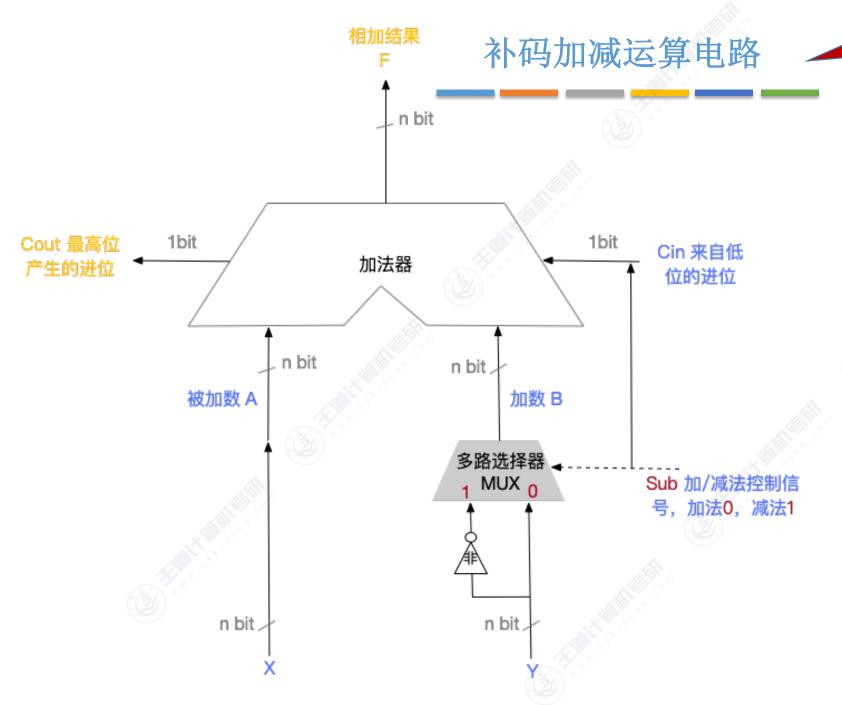
A=1000,B=0111,Cin=1 则 F=0000,Cout=1



例: 4bit补码, X=3, Y=4。X_补=0011, Y_补=0100

n bit<mark>补码 X + Y</mark>,按位相加即可

n bit<mark>补码 X – Y</mark>: 将减数Y全部 按位取反,末位+1,得到[-Y]_补, 减法变加法



也可用于计算无符 号数加减运算

例: 无符号数 X=8, Y=7 用4bit表示, X=1000B, Y=0111B

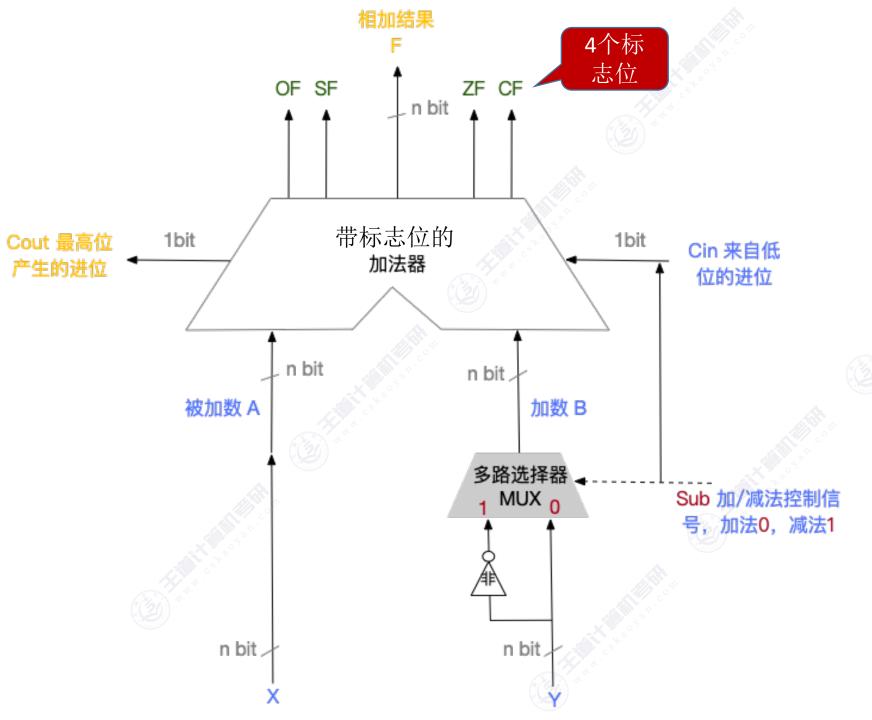
X+Y = 1111B = 15D X-Y = 1000 + (1000+1) = 10001 = 1D 运算结果只保留低四位,最高位进 位丢弃

无符号整数的加法**/**减法也可用该 电路实现

n bit无符号数 X + Y,按位相加即可

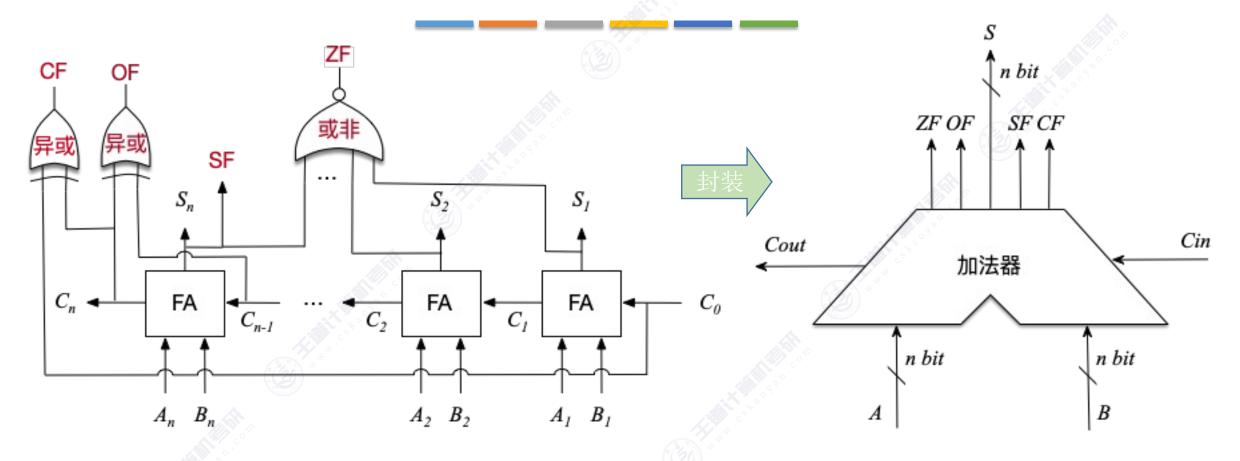
n bit<mark>无符号数 X – Y</mark>:将减数Y全部按位取反,末位+1,减法变加法

王道考研/CSKAOYAN.COM



- OF (Overflow Flag)溢出标志,用于判断带符号数加减运算是否溢出。OF=1溢出;OF=0未溢出
- SF (Sign Flag) 符号标志, 用于判断<mark>带符号数</mark>加减运算结 果的正负性。SF=1 结果为 负; SF=0 结果为正
- ZF (Zero Flag)零标志,用 于判断加减运算结果是否为0。
 ZF=1 表示结果为0; ZF=0 表示结果不为0
- CF (Carry Flag)进位/借位标志,用于判断无符号数加减运算是否溢出。CF=1溢出;CF=0未溢出

回顾: 标志位的生成



 $OF = C_n \oplus C_{n-1}$ ——即 最高位的进位 \oplus 次高位的进位 。反映<mark>带符号数</mark>加减运算是否溢出。 $SF = S_n \longrightarrow U$ 就是取运算结果的最高位(符号位)。反映<mark>带符号数</mark>加减运算的正负性。 $ZF = \overline{S_n + \dots + S_2 + S_1}$ —— 仅当运算结果所有 bit 全0时,ZF 才为1,此时表示运算结果为0。 $CF = C_{out} \oplus C_{in} = C_n \oplus C_0$ ——反映<mark>无符号数</mark>加减运算是否溢出。

回顾: 补码加减法的溢出判断

设机器字长为8位(含1位符号位),A=15,B=-24,求 $[A+B]_{\stackrel{}{\mathbb{A}}}$ 和 $[A-B]_{\stackrel{}{\mathbb{A}}}$

$$C = 124$$
,求 $[A+C]$ 补和 $[B-C]$ 补

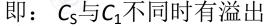
$$[A+C]_{\uparrow h} = 0,0001111 + 0,11111100 = 1,0001011$$
 真值-117
[$B-C]_{\uparrow h} = 1,1101000 + 1,0000100 = 0,1101100$ 真值+108

方法二:采用一位符号位,根据数据位进位情况判断溢出

符号位的进位 $C_{\rm s}$ 最高数值位的进位 $C_{\rm l}$

上溢 0 1 下溢 1 0





处理"不同"的逻辑符号: 异或⊕

溢出逻辑判断表达式为 $V=C_S \oplus C_1$

若V=0,表示无溢出; V=1,表示有溢出。

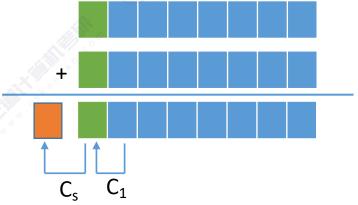
异或逻辑:不同为1,相同为0

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$



回顾: 无符号数加法/减法的溢出判断

手算判断溢出的方法: n bit 无符号整数表示范围 0~2n-1, 超出此范围则溢出

计算机判断溢出的方法:

无符号数加法的溢出判断:最高位产生的进位=1时,发生溢出,否则未溢出。

无符号数减法的溢出判断:减法变加法,最高位产生的进位=0时,发生溢出,否则未溢出。



△ 公众号: 王道在线



b站: 王道计算机教育



抖音:王道计算机考研