

溢出的判断



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

溢出的概念



- **溢出(Overflow)** : 运算结果超出了 数据表示范围

例如: `short i = 23456;`

`short j = 23456; //short 最大值32767`

`short k = i + j; //此时k 为-18624`

加法溢出的判断



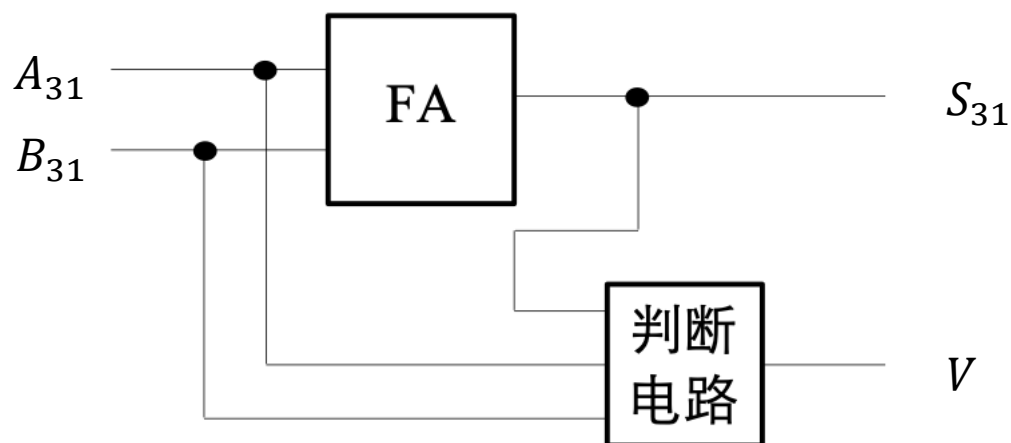
- 符号相异的数相加、符号相同的数相减, 不可能发生溢出
- 符号相同的数相加、或符号相异的数相减、可能发生溢出

操作	操作数 A	操作数 B	显示溢出的结果
$A + B$	≥ 0	≥ 0	< 0
$A + B$	< 0	< 0	≥ 0
$A - B$	≥ 0	< 0	< 0
$A - B$	< 0	≥ 0	≥ 0

方法一：符号位判断

$$V = A_{31}B_{31}\overline{S_{31}} + \overline{A_{31}}\overline{B_{31}}S_{31}$$

FA：表示加法器



缺点：速度慢，在运算结果产生后才能判断溢出

补码加法的几种情况及其溢出检测

最高位进位 C_n ，次高位进位 C_{n-1}

无溢出

$$\begin{array}{r} 0\ 1\ 0\ 1\ 0\ 1 \\ +\ 0\ 0\ 1\ 0\ 0\ 0 \\ \hline 0\ 1\ 1\ 1\ 0\ 1 \end{array}$$

$C_n = 0, C_{n-1} = 0$

无溢出

$$\begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 1\ 0\ 0\ 0 \\ \hline 1\ 1\ 0\ 1\ 1\ 0\ 1 \end{array}$$

$C_n = 1, C_{n-1} = 1$

正溢出

$$\begin{array}{r} 0\ 1\ 0\ 1\ 0\ 1 \\ +\ 0\ 1\ 1\ 0\ 0\ 0 \\ \hline 1\ 0\ 1\ 1\ 0\ 1 \end{array}$$

$C_n = 0, C_{n-1} = 1$

负溢出

$$\begin{array}{r} 1\ 0\ 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 1\ 0\ 0\ 0 \\ \hline 1\ 0\ 1\ 1\ 1\ 0\ 1 \end{array}$$

$C_n = 1, C_{n-1} = 0$

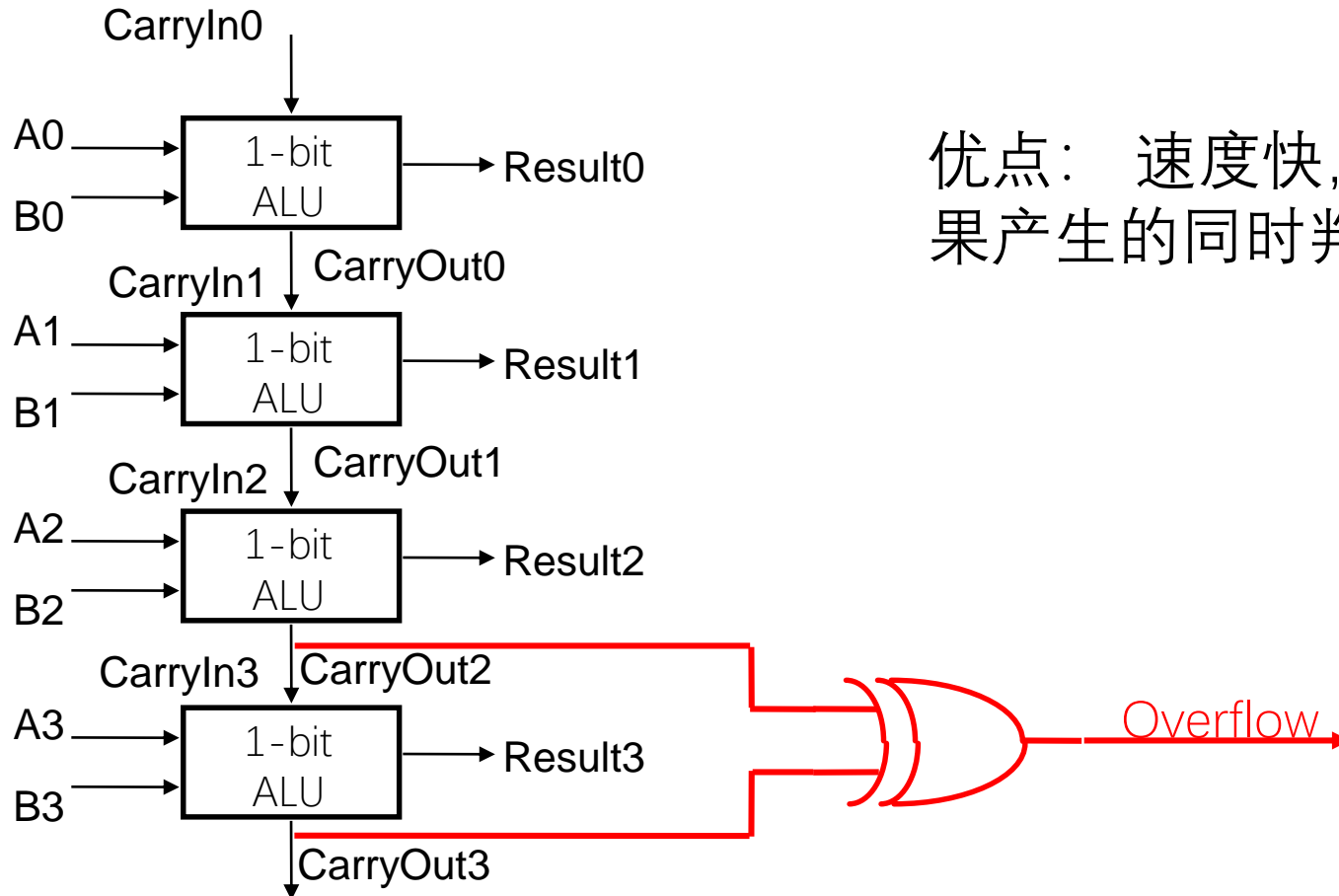
方法二：进位位相异或

溢出信号V对应的真值表

C_n	C_{n-1}	V
0	0	0
1	1	0
1	0	1
0	1	1

$$V = C_n \oplus C_{n-1}$$

N位 ALU: $\text{Overflow} = \text{CarryIn}[N-1] \text{ XOR } \text{CarryOut}[N-1]$



优点：速度快，在运算结果产生的同时判断溢出

溢出概念及其检测方法之三

- 将运算数的符号位设置为00（正数）或11（负数）
- 如果结果的符号位不是00或11，而是01或10则溢出

$$\begin{array}{r}
 \begin{array}{cccccc}
 0 & 0 & 1 & 1 & 1 & \\
 \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \\
 0 & 0 & 1 & 1 & 1 & 7 \\
 0 & 0 & 1 & 1 & 1 & 3 \\
 + & 0 & 0 & 0 & 1 & 1 \\
 \hline
 0 & 1 & 0 & 1 & 0 & -6
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{cccccc}
 1 & 0 & & & & \\
 \swarrow & \swarrow & & & & \\
 1 & 1 & 1 & 0 & 0 & -4 \\
 1 & 1 & 0 & 1 & 1 & -5 \\
 + & 1 & 1 & 1 & 1 & 7 \\
 \hline
 1 & 0 & 1 & 1 & 1 &
 \end{array}
 \end{array}$$

小结

- 溢出及其判断电路
- 计算机内部的运算单元（ALU）
 - 无符号数、有符号数的加减法使用相同的ALU
 - 溢出判断：只适用于补码、不适用于无符号数
- C编译器忽略溢出信号

作为程序员，无论是无符号加法、有符号加法，都要注意运算结果是否溢出，防止程序错误