运算方法与运算器

原码一位乘法的实现方案

主讲教师：刘辉

一、原码的手动乘法计算

二、原码一位乘法的逻辑实现

一、原码的手动乘法计算

1.十进制数的乘法过程

设 $x = +25$， $y = -13$，求x × y，其计算是:“绝对值和符号位单独处理”

每一行是一组部分积
每一个称作一个位积

$$
\begin{array}{rrrl}
 & 2 & 5 & (x) \\
\times & 1 & 3 & (y) \\
\hline
 & 7 & 5 & \\
+ \; 2 & 5 & & \\
\hline
3 & 2 & 5 & (z)
\end{array}
$$

符号位：$z_s = x_s \oplus y_s = + \oplus - = -$

所以：x × y= - 325

一、原码的手动乘法计算

2.二进制数的乘法过程

被乘数　$[x]_{原} = x_s x_{n-1} \cdots x_1 x_0$

乘数　$[y]_{原} = y_s y_{n-1} \cdots y_1 y_0$

$[z]_{原}=[x]_{原}\times[y]_{原}$：

符号位：单独处理　$z_s=x_s \oplus y_s$

绝对值：乘法运算

$$|x|\times|y|=x_{n-1} \cdots x_1 x_0 \times y_{n-1} \cdots y_1 y_0$$

一、原码的手动乘法计算

二进制数的乘法过程

设 $x = +1101$， $y = -1011$，原码表示，求x × y，其计算是：“绝对值和符号位单独处理”

$$
\begin{array}{cccccccccc}
 & & & & & 1 & 1 & 0 & 1 & (x) \\
\times & & & & & 1 & 0 & 1 & 1 & (y) \\
\hline
 & & & & & 1 & 1 & 0 & 1 & \\
 & & & & 1 & 1 & 0 & 1 & & \\
 & & & 0 & 0 & 0 & 0 & & & \\
+ & & 1 & 1 & 0 & 1 & & & & \\
\hline
 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & (z)
\end{array}
$$

每一行是一组部分积
每一个称作一个位积

符号位：$z_s = x_s \oplus y_s = 0 \oplus 1 = 1$

所以：$[x \times y]_原 = 1\ 10001111$

二、原码一位乘法的逻辑实现

1.机器算法与人工算法的同异性

（1）计算机不能完全照搬人工算法

原因之一： 机器通常只有n位长,两个n位数相乘,乘积可能为2n位。

原因之二： 只有两个操作数相加的加法器难以胜任将n个位积一次相加起来的运算。

（2）实现思想：多次“加法+移位”

二、原码一位乘法的逻辑实现

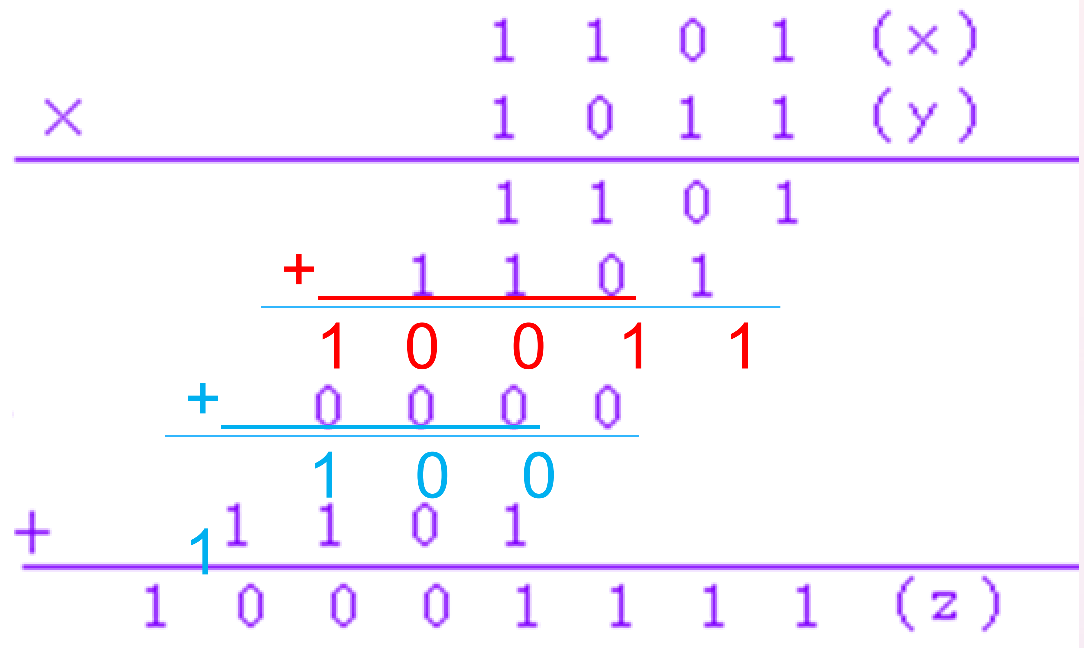
例：按多次“加法-移位”操作完成1011× 1101

原码一位乘法所需电路：

三个寄存器： 初始分别放x，y和部分积

一个加法器： 部分积和x相加

一个计数器： 计数逐位相乘的次数



小结

- 原码的手动乘法计算：部分积相加
- 原码一位乘法的逻辑实现:

计算机实现乘法与手动计算的不同:
机器n位长,n位* n位=2n位;
只有两数相加的加法器。
实现思想：多次“加法+移位”