本节内容

乘法/除法运算

复习方向说明

观察考研大纲要求

【考纲内容】

（一）数制与编码

进位计数制及其相互转换；定点数的编码表示

（二）运算方法和运算电路

基本运算部件：加法器；算术逻辑单元（ALU）

加/减运算：补码加/减运算器；标志位的生成

乘/除运算：乘/除法运算的基本原理；乘法电路和除法电路的基本结构

（三）整数的表示和运算

无符号整数的表示和运算；有符号整数的表示和运算

（四）浮点数的表示和运算

浮点数的表示：IEEE 754 标准；浮点数的加/减运算

根据考研大纲可知：仅要求掌握整数的乘/除法运算，不要求掌握小数的乘/除法

乘法、除法运算学习说明

- 课程讲解顺序
  - 乘法
    - 无符号整数的乘法运算原理
    - 带符号整数的乘法运算原理
  - 除法
    - 无符号整数的除法运算原理
    - 带符号整数的除法运算原理

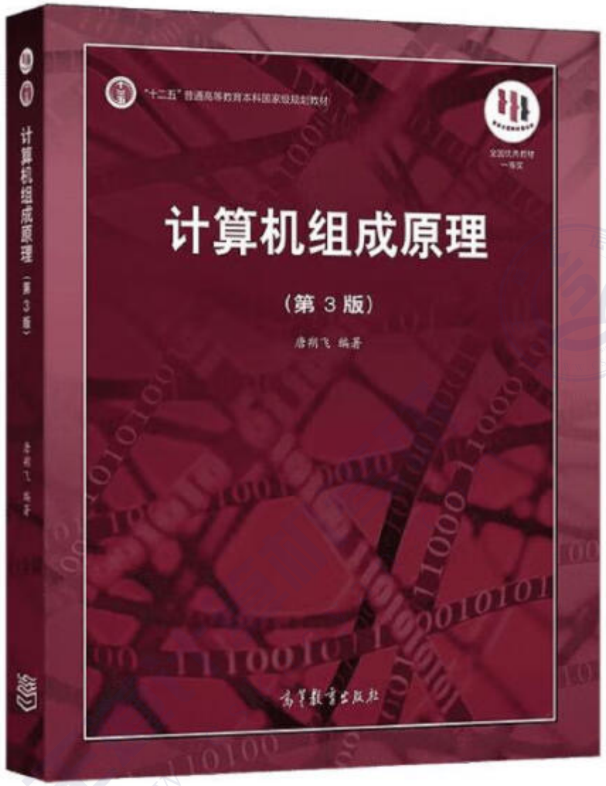
考研大纲的侧重点是“整数”的乘法、除法，对“小数”的乘除法不做要求

在本课程中，我们将尽量贴近考研要求，以“无符号整数、带符号整数”为例，学习乘法/除法原理

- 定点数
  - 定点小数
    - 在现代计算机中，定点小数一般用"原码"表示，用于表示浮点数的"尾数"
  - 定点整数
    - 无符号整数（没有符号位，所有bit都是数值位）
    - 带符号整数（"补码"表示）

→主流教材一般会以定点小数“原码”为例介绍乘法、除法

Ref：唐朔飞老师教材



6.3.3　乘法运算

在计算机中，乘法运算是一种很重要的运算，有的机器由硬件乘法器直接完成乘法运算，有的机器内没有乘法器，但可以按机器做乘法运算的方法，用软件编程实现。因此，学习乘法运算方法不仅有助于乘法器的设计，也有助于乘法编程。

下面从分析笔算乘法入手，介绍机器中用到的几种乘法运算方法。

1. 分析笔算乘法

设 $A=0.1101$，$B=0.1011$，求 $A\times B$。

笔算乘法时，乘积的符号由两数符号心算而得：正正得正。其数值部分的运算如下：

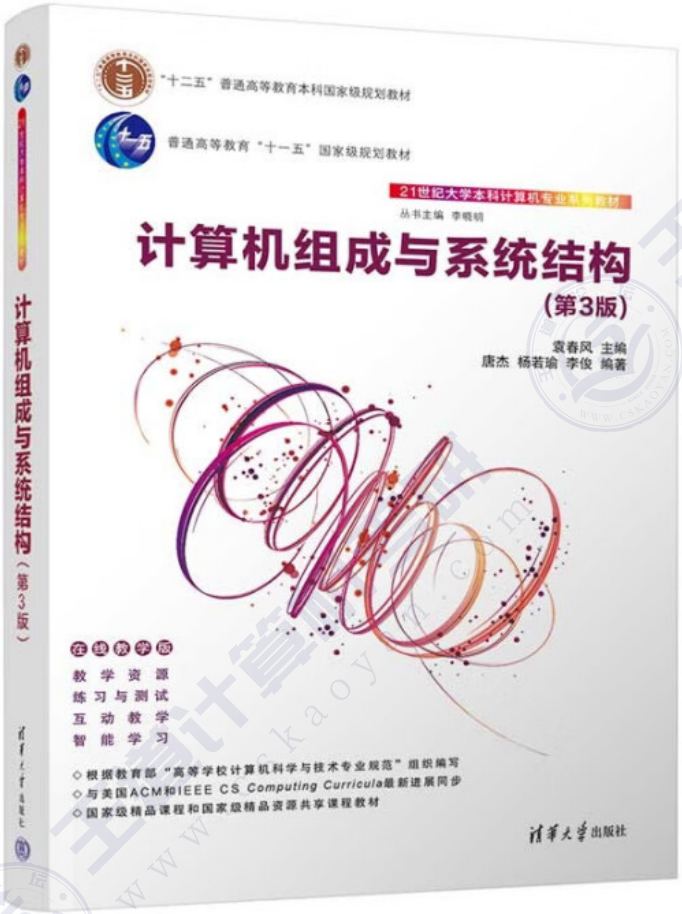
$$
\begin{array}{rll}
 & 0.1101 & \\
\times & 0.1011 & \\
\hline
 & 1101 & \cdots\cdots A\times 2^0 \quad A\text{ 不移位} \\
 & 1101 & \cdots\cdots A\times 2^1 \quad A\text{ 左移 1 位} \\
 & 0000 & \cdots\cdots 0\times 2^2 \quad 0\text{ 左移 2 位} \\
 & 1101 & \cdots\cdots A\times 2^3 \quad A\text{ 左移 3 位} \\
\hline
 & 0.10001111 &
\end{array}
$$

所以　$A\times B=+0.10001111$

可见，这里包含着被乘数 A 的多次左移，以及 4 个位积的相加运算。

若计算机完全模仿笔算乘法步骤，将会有两大困难：其一，将 4 个位积一次相加，机器难以实现；其二，乘积位数增长了一倍，这将造成器材的浪费和运算时间的增加。为此，对笔算乘法进行改进。

Ref：袁春风老师教材



3.3.3 原码乘法运算

原码作为浮点数尾数的表示形式，需要计算机能实现定点原码小数的乘法运算。根据每次部分积是一位相乘得到还是两位相乘得到，可以有原码一位乘法和原码两位乘法，根据原码两位乘法的原理推广，可以有原码多位乘法。

1. 原码一位乘法

用原码实现乘法运算时，符号位与数值位分开计算，因此，原码乘法运算分为两步。

（1）确定乘积的符号位。由两个乘数的符号异或得到。

（2）计算乘积的数值位。乘积的数值部分为两个乘数的数值部分之积。

原码乘法算法描述如下：已知$[x]_{原}=X_0.X_1\cdots X_n$，$[y]_{原}=Y_0.Y_1\cdots Y_n$，则$[x\times y]_{原}=Z_0.Z_1\cdots Z_{2n}$，其中$Z_0=X_0\oplus Y_0$，$Z_1\cdots Z_{2n}=(0.X_1\cdots X_n)\times(0.Y_1\cdots Y_n)$。

可以不管小数点，事实上在机器内部也没有小数点，只是约定了一个小数点的位置，小数点约定在最左边就是定点小数乘法，约定在右边就是定点整数乘法。因此，两个定点小数的数值部分之积可以看成是两个无符号数的乘积。

下面是一个手算乘法的例子，以此可以推导出两个无符号数相乘的计算过程。

$$
\begin{array}{rl}
0.1011 & \text{被乘数 } X=0.X_1X_2X_3X_4=0.1011 \\
\times 0.1101 & \text{乘数 } Y=0.Y_1Y_2Y_3Y_4=0.1101 \\
\hline
1011 & \cdots\cdots X\times Y_4\times 2^{-4} \\
0000 & \cdots\cdots X\times Y_3\times 2^{-3} \\
1011 & \cdots\cdots X\times Y_2\times 2^{-2} \\
1011 & \cdots\cdots X\times Y_1\times 2^{-1} \\
\hline
0.10001111 &
\end{array}
$$

由此可知，$X\times Y=\sum_{i=1}^{4}(X\times Y_i\times 2^{-i})=0.10001111$。

Ref：2026王道书

2.2.4 定点数的乘除运算

1. 定点乘法运算

（1）乘法运算的基本原理

命题追踪 ▶▶ 用加法、移位指令实现乘法运算的原理（2020、2024）

命题追踪 ▶▶ 用软/硬件实现乘法指令的速度对比（2020）

原码乘法的特点是符号位与数值位是分开求的，原码乘法运算分为两步：① 乘积的符号位由两个乘数的符号位“异或”得到；② 乘积的数值位是两个乘数的绝对值之积。两个定点数的数值部分之积可视为两个无符号数的乘积。下面是两个无符号数相乘的手算过程。

$$
\begin{array}{rll}
0.1101 & & \text{被乘数 } X=0.x_1x_2x_3x_4=0.1101 \\
\times\ 0.1011 & & \text{乘数 } Y=0.y_1y_2y_3y_4=0.1011 \\
\hline
1101 & \text{-------- } X\times y_4\times 2^{-4} & X\times 1 \text{ 右移 4 位} \\
1101 & \text{-------- } X\times y_3\times 2^{-3} & X\times 1 \text{ 右移 3 位} \\
0000 & \text{-------- } X\times y_2\times 2^{-2} & X\times 0 \text{ 右移 2 位} \\
1101 & \text{-------- } X\times y_1\times 2^{-1} & X\times 1 \text{ 右移 1 位} \\
\hline
0.10001111 & &
\end{array}
$$

定点小数的“原码”表示，本质上是“符号位+无符号数”