中山大學 软件工程学院

SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

计算机组成原理

授课老师：吴炜滨

大纲

- 定点运算
  - 乘法运算

乘法运算

- 计算机中怎么做二进制的乘法运算？
  - 可以分析一下笔算乘法是怎么做的
  - 把笔算乘法做一定的改进，然后将其用计算机硬件去实现

大纲

- 定点运算
  - 乘法运算
    - 笔算乘法的分析
    - 笔算乘法的改进
    - 原码的乘法运算

笔算乘法的分析

$A = -0.1101 \quad B = 0.1011$

$$
\begin{array}{r}
0.1101 \\
\times\ 0.1011 \\
\hline
1101 \\
1101 \\
0000 \\
1101 \\
\hline
0.10001111
\end{array}
$$

- 乘积的符号心算求得
  - 符号位单独处理
- 乘积的数值部分用乘法规则计算
  - 乘数的某一位决定是否加被乘数
  - 4个位积一起相加
  - 乘积的位数扩大一倍

$A \times B = -0.10001111$

笔算乘法的分析

$A = -0.1101 \quad B = 0.1011$

$$
\begin{array}{r}
0.1101 \\
\times\ 0.1011 \\
\hline
1101 \\
1101 \\
0000 \\
1101 \\
\hline
0.10001111
\end{array}
$$

$A \times B = -0.10001111$

- 计算机中如何模拟？
  - 符号位单独处理
    - 异或电路
  - 乘数的某一位决定是否加被乘数
    - 乘数放移位寄存器，每判断完一次最低位，右移一次
  - 4个位积一起相加
    - 多次累加
    - 位积每次累加前，已有累加结果右移一位
  - 乘积的位数扩大一倍
    - 两个寄存器保存乘积

笔算乘法的改进

$A = -0.1101 \quad B = 0.1011$

$$A \cdot B = A \cdot 0.1011$$

$$= 0.1A + 0.00A + 0.001A + 0.0001A$$

$$= 0.1A + 0.00A + 0.001(A + 0.1A)$$

$$= 0.1A + 0.01[0 \cdot A + 0.1(A + 0.1A)]$$

$$= 0.1\{A + 0.1[0 \cdot A + 0.1(A + 0.1A)]\}$$

右移一位

$$= 2^{-1}\{1 \cdot A + 2^{-1}[0 \cdot A + 2^{-1}(1 \cdot A + 2^{-1}(1 \cdot A + 0))]\}$$

笔算乘法的改进

$$A \cdot B = A \cdot 0.1011$$

$$= 2^{-1}\{1 \cdot A + 2^{-1}[0 \cdot A + 2^{-1}(1 \cdot A + 2^{-1}(1 \cdot A + 0))]\}$$

第一步　被乘数$A + 0$　　①

第二步　右移一位，得新的部分积　　②

第三步　部分积 + 被乘数　　③

⋮

第八步　右移一位，得结果　　⑧

改进后的笔算乘法过程（竖式）

$A = -0.1101$

$B = 0.1011$

部分积	乘数	说明
0.0000 + 0.1101	1011 =	初态，部分积 = 0 乘数为 1，加被乘数
0.1101 0.0110 + 0.1101	1101 =	→ 1，形成新的部分积 乘数为 1，加被乘数
1.0011 0.1001 + 0.0000	1 1110 =	→ 1，形成新的部分积 乘数为 0，加 0
0.1001 0.0100 + 0.1101	11 1111 =	→ 1，形成新的部分积 乘数为 1，加被乘数
1.0001 0.1000	111 1111	→ 1，得结果

小结

- 乘法运算
  - 符号位由异或电路来获得
  - 数值部分可用加和移位实现
    - 数据数值部分位数为n → 加 n 次，移 n 次
- 由乘数的末位决定被乘数是否与原部分积相加
  - 末位为1，加被乘数，否则加0
  - 然后 → 1 位形成新的部分积，同时 乘数 → 1位（末位移丢），空出高位存放部分积的低位。
  - 被乘数只与部分积的高位相加
- 硬件
  - 3个寄存器（X：被乘数，ACC：乘积高位，MQ：乘积低位、乘数），其中2个具有移位功能（ACC，MQ）；1个$n+1$位全加器

大纲

- 定点运算
  - 乘法运算
    - 原码的乘法运算
      - 运算规则
      - 硬件配置
      - 控制流程

原码一位乘运算规则

- 以小数为例
  - 整数乘法过程相同，将小数点改为逗号即可

设$[x]_{原} = x_0.x_1x_2 \cdots x_n$　　$[y]_{原} = y_0.y_1y_2 \cdots y_n$

$$[x \bullet y]_{原} = (x_0 \oplus y_0).(0.x_1x_2 \cdots x_n)(0.y_1y_2 \cdots y_n)$$
$$= (x_0 \oplus y_0).x^*y^*$$

式中 $x^* = 0.x_1x_2 \cdots x_n$　为 x 的绝对值

$y^* = 0.y_1y_2 \cdots y_n$　为 y 的绝对值

乘积的符号位单独处理 $x_0 \oplus y_0$　　数值部分为绝对值相乘 $x^* \bullet y^*$

原码一位乘运算规则

- 原码一位乘递推公式

$$x^* \bullet y^* = x^*(0.y_1y_2 \cdots y_n)$$

$$= x^*(y_1 2^{-1} + y_2 2^{-2} + \cdots + y_n 2^{-n})$$

$$= 2^{-1}(y_1 x^* + 2^{-1}(y_2 x^* + \cdots 2^{-1}(y_n x^* + 0)\cdots))$$

原码一位乘运算规则

原码一位乘递推公式

- 加法和移位实现乘法

$$x^* \bullet y^* = x^*(0.y_1y_2 \cdots y_n)$$

$$= 2^{-1}(y_1x^* + 2^{-1}(y_2x^* + \cdots 2^{-1}(y_nx^* + 0)\cdots))$$

z_0 z_1 $\cdots$ z_n

$$z_0 = 0$$

$$z_1 = 2^{-1}(y_nx^* + z_0)$$ 由y_n决定是否加上被乘数的数值部分，然后右移一位

$$z_2 = 2^{-1}(y_{n-1}x^* + z_1)$$

$$\vdots$$

$$z_n = 2^{-1}(y_1x^* + z_{n-1})$$ 共n次加法，n次移位

原码一位乘

- 已知机器字长为5位（含1位符号位），$x = -0.1110$，$y = 0.1101$，求$[x \cdot y]_{原}$

解：$[x]_{原} = 1.1110$

$[y]_{原} = 0.1101$

■ 已知机器字长为5位（含1位符号位），$x = -0.1110$，$y = 0.1101$，求$[x \cdot y]_{原}$

解：数值部分

部分积	乘数	说明
0.0000	1101	部分积 初态 $z_0 = 0$
+ 0.1110	=	+ x^*
0.1110		
逻辑右移 0.0111	0110	→ 1，得 z_1
+ 0.0000	=	+ 0
0.0111	0	
逻辑右移 0.0011	1011	→ 1，得 z_2
+ 0.1110	=	+ x^*
1.0001	10	
逻辑右移 0.1000	1101	→ 1，得 z_3
+ 0.1110	=	+ x^*
1.0110	110	
逻辑右移 0.1011	0110	→ 1，得 z_4 → $x^* \cdot y^* = 0.10110110$

$[x]_{原} = 1.1110$

$[y]_{原} = 0.1101$

原码一位乘

- 已知机器字长为5位（含1位符号位），$x = -0.1110$，$y = 0.1101$，求$[x \cdot y]_{原}$

$$[x]_{原} = 1.1110 \qquad [y]_{原} = 0.1101$$

  - 乘积的符号位

$$x_0 \oplus y_0 = 1 \oplus 0 = 1$$

  - 乘积的数值部分由两数绝对值相乘而得

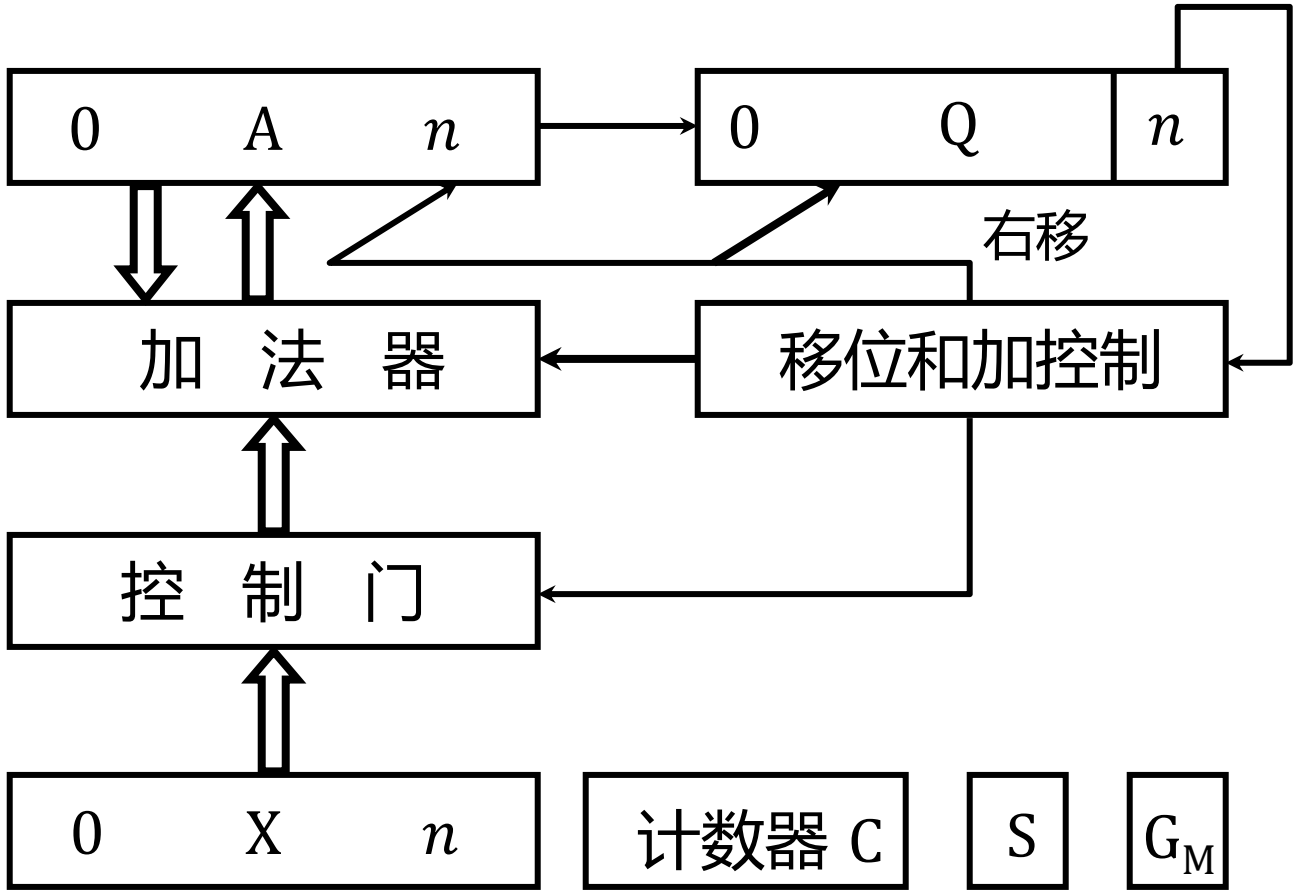
$$x^* \cdot y^* = 0.1\,0\,1\,1\,0\,1\,1\,0 \qquad \text{则 } [x \cdot y]_{原} = 1.1\,0\,1\,1\,0\,1\,1\,0$$

原码一位乘

- 原码一位乘的特点
  - 绝对值参与运算
  - 逻辑移位
    - “符号位”上的数字不代表符号，是低位数值部分相加以后向高位的进位
  - 用移位的次数判断乘法是否结束
    - 如果操作数的数值部分位数为n，则移位n次

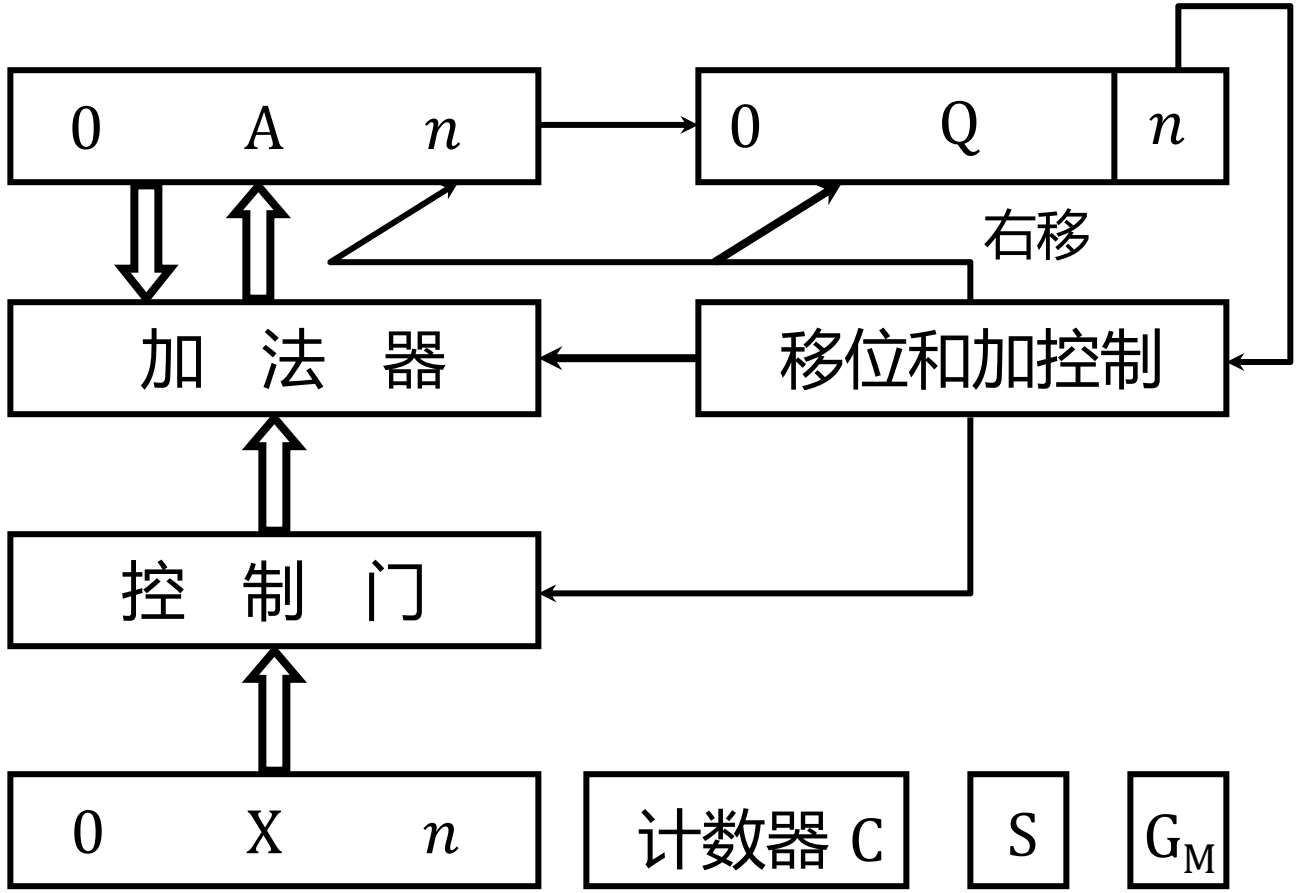
原码一位乘的硬件配置

- 寄存器A、X、Q、加法器均 $n+1$ 位，计算过程中：
  - A：部分积的高位，最高位非符号位，而是低位数值部分相加后向高位的进位
  - X：被乘数的原码
  - Q（MQ）：乘数的原码、部分积的低位
- 计数器C
  - 计数器值=移位次数=数值部分位数=n
  - 每移位一次，计数器值减1
- S：乘积符号
  - 值=被乘数和乘数的符号位进行异或
- G_M：乘法标志



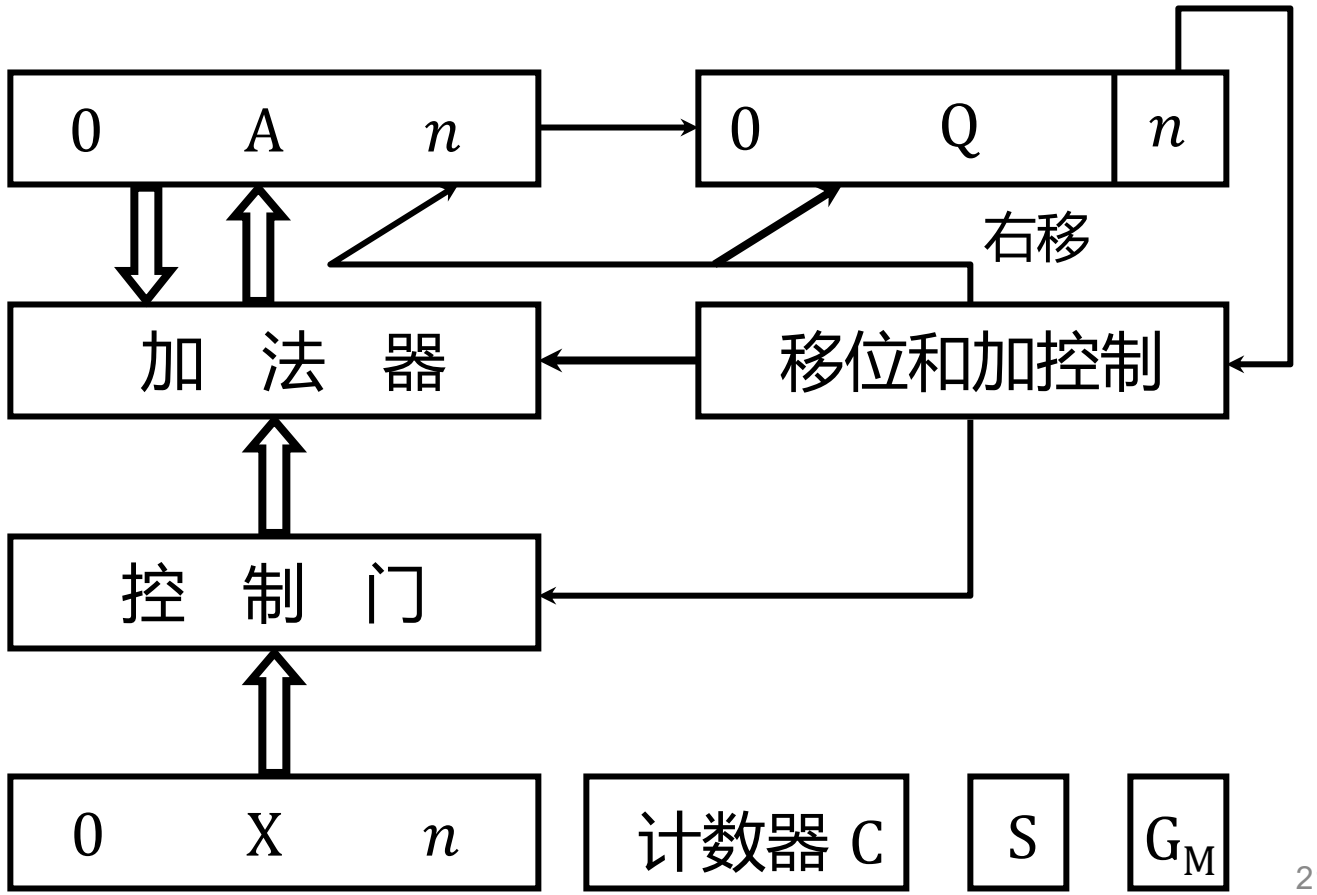
原码一位乘的硬件配置

- 移位和加受末位乘数控制
  - 末位乘数为1，控制门打开，将被乘数送至加法器和部分积进行累加，然后A、Q右移一位
  - 末位乘数为0，控制门关闭，不送被乘数， A、Q直接右移一位
    - 用移位次数来控制乘法的结束，而不用加法次数



原码一位乘控制流程

- 准备：被乘数原码 → X，乘数原码 → Q， n → C， A清零作为初始部分积
- 求积符： $X_0 \oplus Q_0 \rightarrow S$
- 取绝对值： $0 \rightarrow X_0$， $0 \rightarrow Q_0$
- $Q_n = 1$?
  - Y：[A] + [X] → A
- A、Q同时逻辑右移一位
- [C] – 1 → C
- C = 0?
  - Y：结束
  - N：回到判断Q_n=1?



谢谢！