计算机组织与体系结构

第九讲

计算机科学与技术学院
舒燕君

Recap

– **MIPS指令集**
  - ✓ 数据类型（整数、浮点数）
  - ✓ 寻址方式（寄存器寻址、立即数寻址、偏移寻址）
  - ✓ 指令格式（**I**类、**R**类、**J**类）
  - ✓操作类型（存取、**ALU**、转移、浮点）

– **CPU**结构
  - ✓ **CU**、**ALU**、寄存器和中断

– 定点运算（移位）

必修实验

必修实验第一次课：10月16日上午10点
硬件实验中心G709

必修实验QQ群：771417971

实验资料：
https://hit-coa.gitlab.io/hit-coa-lab/index.html

第5章　CPU设计与实现

5.1 CPU 的结构

<u>**5.2 运算方法与ALU**</u>

5.3 多级时序系统（X86）

5.4 MIPS CPU的简单实现

5.2 运算方法与ALU

5.2.1 定点运算

5.2.2 浮点四则运算

5.2.3 算术逻辑单元

二、加减法运算

1. 补码加减运算公式

(1) 加法

整数 $[A]_{补} + [B]_{补} = [A+B]_{补}$（mod 2^{n+1}）

小数 $[A]_{补} + [B]_{补} = [A+B]_{补}$（mod 2）

(2) 减法

$$A-B = A+(-B)$$

整数 $[A-B]_{补} = [A+(-B)]_{补} = [A]_{补} + [-B]_{补}$ (mod 2^{n+1})

小数 $[A-B]_{补} = [A+(-B)]_{补} = [A]_{补} + [-B]_{补}$ (mod 2)

连同符号位一起相加，符号位产生的进位自然丢掉

设机器数字长为 8 位（含 1 位符号位）
且 $A = 15$， $B = 24$，用补码求 $A - B$

解：$A = 15 = 0001111$

$B = 24 = 0011000$

$[A]_补 = 0, 0001111$　　　$[B]_补 = 0, 0011000$

$+ [-B]_补 = 1, 1101000$

$[A]_补 + [-B]_补 = 1, 1110111 = [A - B]_补$

$\therefore A - B = -1001 = -9$

练习 1　设 $x = \frac{9}{16}$ $y = \frac{11}{16}$，用补码求 $x+y$

$x + y = -0.1100 = -\frac{12}{16}$　　错

练习 2　设机器数字长为 8 位（含 1 位符号位）
且 $A = -97$，$B = +41$，用补码求 $A - B$

$A - B = +1110110 = +118$　　错

3. 溢出判断

(1) 一位符号位判溢出

参加操作的 两个数（减法时即为被减数和“求补”以后的减数）符号相同，其结果的符号与原操作数的符号不同，即为溢出

硬件实现

最高有效位的进位 $\oplus$ 符号位的进位 = 1　　溢出

如

$$\left.\begin{array}{l} 1 \oplus 0 = 1 \\ 0 \oplus 1 = 1 \end{array}\right\} \text{有 溢出}$$

$$\left.\begin{array}{l} 0 \oplus 0 = 0 \\ 1 \oplus 1 = 0 \end{array}\right\} \text{无 溢出}$$

(2) 两位符号位判溢出

$$[x]_{补'} = \begin{cases} x & 1 > x \geq 0 \\ 4 + x & 0 > x \geq -1 \end{cases} \pmod 4$$

$$[x]_{补'} + [y]_{补'} = [\,x + y\,]_{补'} \pmod 4$$

$$[x - y]_{补'} = [x]_{补'} + [-\,y]_{补'} \pmod 4$$

结果的双符号位 相同　　未溢出　　00, ×××××

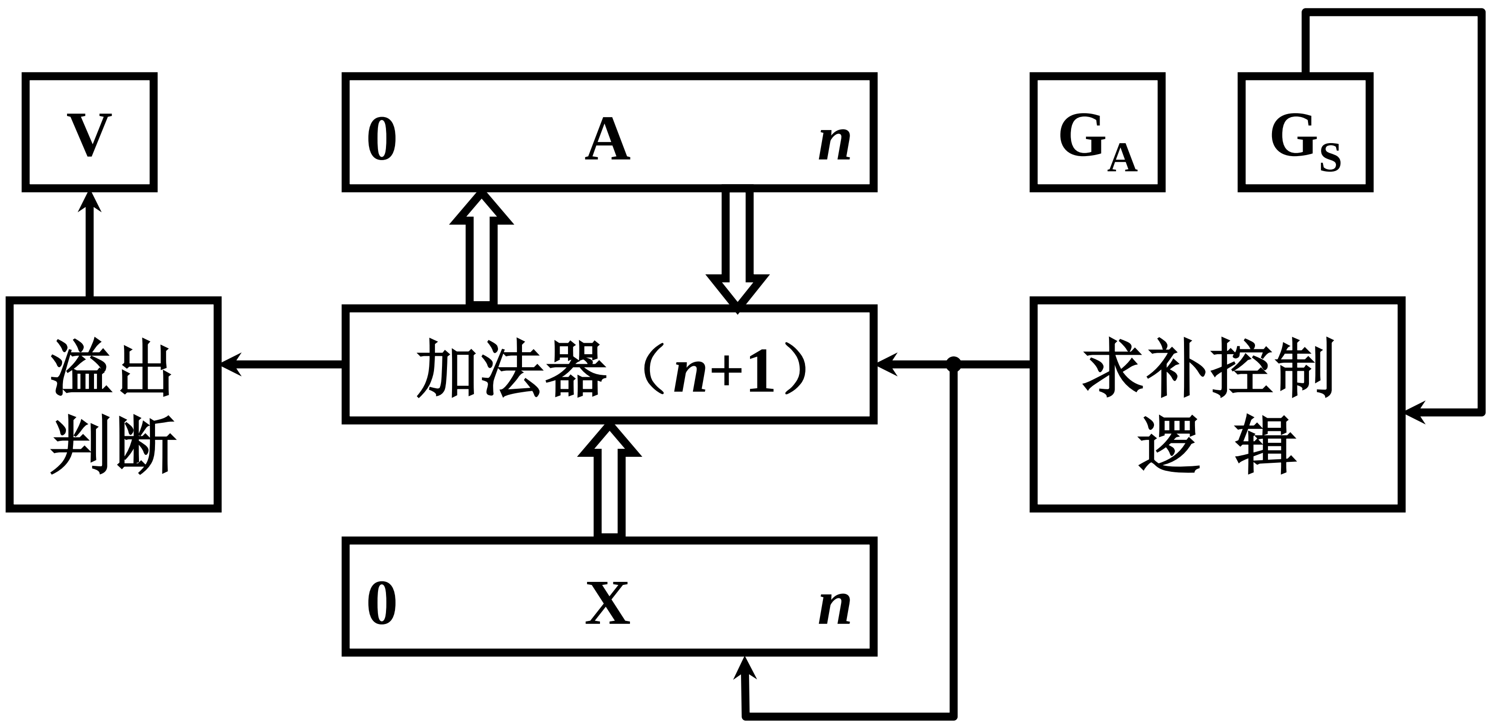
11, ×××××

结果的双符号位 不同　　溢出　　10, ×××××

01, ×××××

最高符号位 代表其 真正的符号

4. 补码加减法的硬件配置



A、X 均 n+1 位

用减法标记 G_S 控制求补逻辑

三、乘法运算

1. 分析笔算乘法

$A = -0.1101 \qquad B = 0.1011$

$A \times B = -0.10001111$ 乘积的符号心算求得

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

✓ 符号位单独处理

✓ 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

2. 笔算乘法改进

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

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

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

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

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

右移一位

$$= \underbrace{\boxed{2^{-1}}\{A + 2^{-1}[0 \cdot A + 2^{-1}\underbrace{(A + \underbrace{2^{-1}\underbrace{(A+0)}_{\text{①}}}_{\text{②}})}_{\text{③}}]\}}_{\text{⑧}}$$

第一步　被乘数A + 0

第二步　右移 一 位，得新的部分积

第三步　部分积 + 被乘数

⋮

第八步　右移 一 位，得结果

3. 改进后的笔算乘法过程（竖式）

部 分 积	乘 数	说 明
0 . 0 0 0 0 + 0 . 1 1 0 1	1 0 1 1	初态，部分积 = 0 乘数为 1，加被乘数
0 . 1 1 0 1 0 . 0 1 1 0 + 0 . 1 1 0 1	1 1 0 1	→1，形成新的部分积 乘数为 1，加被乘数
1 . 0 0 1 1 0 . 1 0 0 1 + 0 . 0 0 0 0	1 1 1 1 0	→ 1，形成新的部分积 乘数为 0，加 0
0 . 1 0 0 1 0 . 0 1 0 0 + 0 . 1 1 0 1	1 1 1 1 1 1	→ 1，形成新的部分积 乘数为 1，加 被乘数
1 . 0 0 0 1 0 . 1 0 0 0	1 1 1 1 1 1 1	→1，得结果

小结

- 乘法 运算可用 加和移位实现
  $n = 4$，加 4 次，移 4 次
- 由乘数的末位决定被乘数是否与原部分积相加，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，具有移位功能
　　　1 个全加器

4. 原码乘法

(1) 原码一位乘运算规则

以小数为例

$$\text{设}[x]_{原} = x_0 . x_1 x_2 \quad \cdots \quad x_n$$

$$[y]_{原} = y_0 . y_1 y_2 \quad \cdots \quad y_n$$

$$[x \cdot y]_{原} = (x_0 \oplus y_0).(0.x_1 x_2 \cdots x_n)(0.y_1 y_2 \cdots y_n)$$

$$= (x_0 \oplus y_0).\, x^* y^*$$

式中 $x^* = 0.x_1 x_2 \cdots x_n$ 为 x 的绝对值

$y^* = 0.y_1 y_2 \cdots y_n$ 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 $x^* \cdot y^*$

(2) 原码一位乘递推公式

$$x^* \cdot y^* = x^*(0.y_1y_2 \ \ldots \ y_n)$$

$$= x^*(y_12^{-1}+y_22^{-2}+ \ \cdots \ + y_n2^{-n})$$

$$= \underbrace{2^{-1}(y_1x^*+2^{-1}(y_2x^*+ \ \cdots \underbrace{2^{-1}(y_nx^* + \underbrace{0}_{z_0})}_{z_1} \cdots))}_{z_n}$$

$$z_0 = 0$$

$$z_1 = 2^{-1}(y_nx^*+z_0)$$

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

$$\vdots$$

$$z_n = 2^{-1}(y_1x^*+z_{n-1})$$

例6.21 已知 $x = -0.1110$ $y = 0.1101$ 求$[x \cdot y]_原$

解： 数值部分的运算

	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	$\rightarrow 1$， 得 z_1
	+ 0.0000		+ 0
	0.0111	0	
逻辑右移	0.0011	1011	$\rightarrow 1$， 得 z_2
	+ 0.1110		$+ x^*$
	1.0001	10	
逻辑右移	0.1000	1101	$\rightarrow 1$， 得 z_3
	+ 0.1110		$+ x^*$
	1.0110	110	
逻辑右移	0.1011	0110	$\rightarrow 1$， 得 z_4

例6.21 结果

① 乘积的符号位 $x_0 \oplus y_0 = 1 \oplus 0 = 1$

② 数值部分按绝对值相乘

$$x^* \cdot y^* = 0.\,1\,0\,1\,1\,0\,1\,1\,0$$

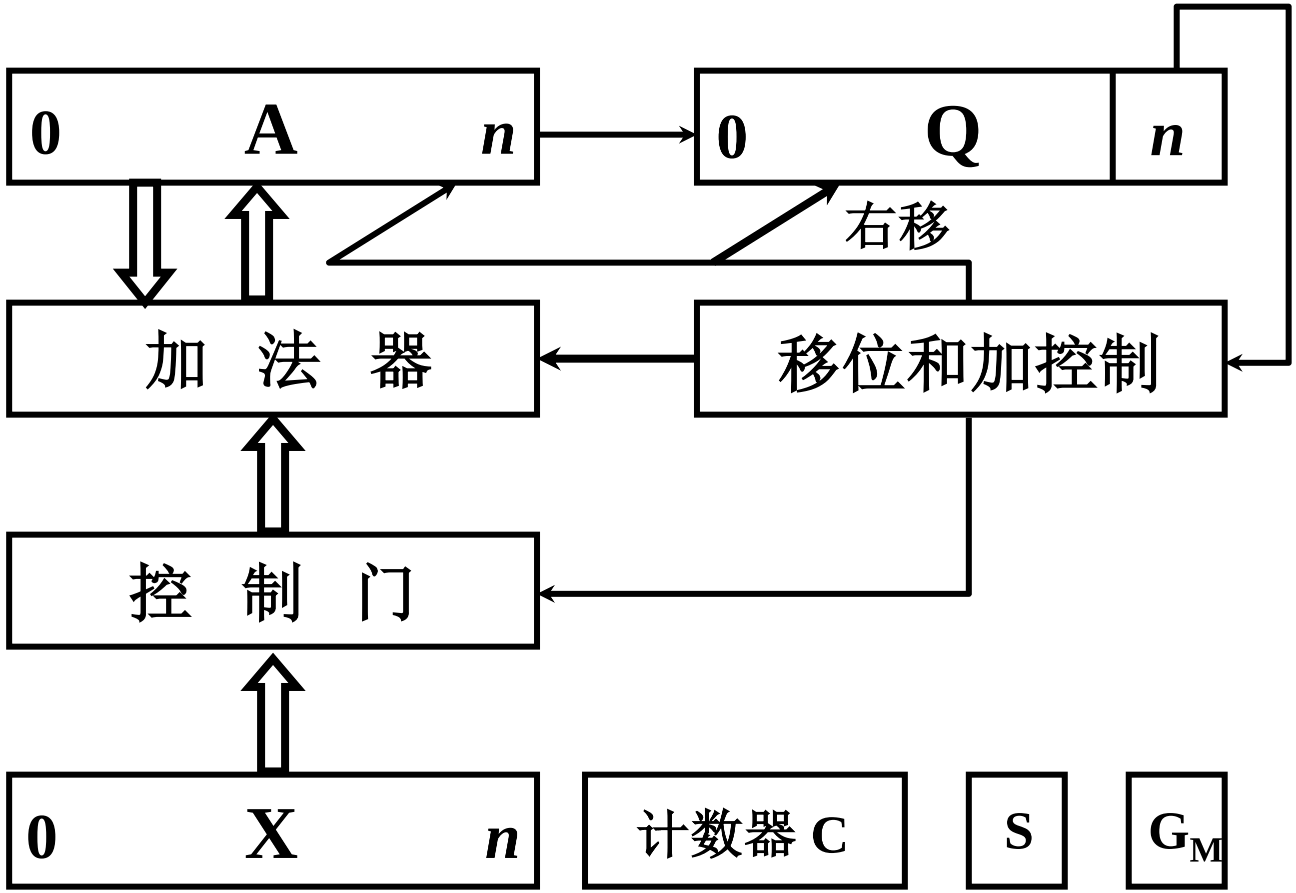
则 $[x \cdot y]_{原} = 1.\,1\,0\,1\,1\,0\,1\,1\,0$

特点　绝对值运算

用移位的次数判断乘法是否结束

逻辑移位

(3) 原码一位乘的硬件配置



A、X、Q 均 *n*+1 位

移位和加受末位乘数控制

5.2.2 浮点四则运算

一、浮点加减运算

$$x = S_x \cdot 2^{j_x} \qquad y = S_y \cdot 2^{j_y}$$

1. 对阶

(1) 求阶差

$$\Delta j = j_x - j_y = \begin{cases} = 0 & j_x = j_y & \text{已对齐} \\ > 0 & j_x > j_y & \begin{cases} x \text{ 向 } y \text{ 看齐} & S_x \leftarrow 1,\ j_x - 1 \\ y \text{ 向 } x \text{ 看齐} & \checkmark\ S_y \rightarrow 1,\ j_y + 1 \end{cases} \\ < 0 & j_x < j_y & \begin{cases} x \text{ 向 } y \text{ 看齐} & \checkmark\ S_x \rightarrow 1,\ j_x + 1 \\ y \text{ 向 } x \text{ 看齐} & S_y \leftarrow 1,\ j_y - 1 \end{cases} \end{cases}$$

(2) 对阶原则

小阶向大阶看齐

例如 $x = 0.1101 \times 2^{01}$　　$y = (-0.1010) \times 2^{11}$

求 $x+y$

解： $[x]_{补} = 00, 01; 00.1101$　　$[y]_{补} = 00, 11; 11.0110$

1. 对阶

① 求阶差 $[\Delta j]_{补} = [j_x]_{补} - [j_y]_{补}$

$$
\begin{array}{rr}
 & = 00, 01 \\
+ & 11, 01 \\
\hline
 & 11, 10
\end{array}
$$

阶差为负（-2） $\therefore S_x \longrightarrow 2$　$j_x + 2$

② 对阶 $[x]_{补'} = 00, 11; 00.0011$

2. 尾数求和

$$
\begin{array}{lll}
 & [S_x]_{补'} & = 00.0011 \\
+ & [S_y]_{补} & = 11.0110 \\
\hline
 & & \ 11.1001
\end{array}
$$

对阶后的$[S_x]_{补'}$

$\therefore [x+y]_{补} = 00, 11; 11.1001$

3. 规格化

(1) 规格化数的定义

$$r = 2 \qquad \frac{1}{2} \le |S| < 1$$

(2) 规格化数的判断

$S>0$	规格化形式	$S<0$	规格化形式
真值	$0.1\times\times\cdots\times$	真值	$-0.1\times\times\cdots\times$
原码	$0.\boxed{1}\times\times\cdots\times$	原码	$1.\boxed{1}\times\times\cdots\times$
补码	$\boxed{0.1}\times\times\cdots\times$	补码	$\boxed{1.0}\times\times\cdots\times$
反码	$0.1\times\times\cdots\times$	反码	$1.0\times\times\cdots\times$

原码　不论正数、负数，第一数位为1

补码　符号位和第 一数位不同

特例

$$S = -\frac{1}{2} = -0.100\cdots 0$$

$$[S]_{原} = 1.100\cdots 0$$

$$[S]_{补} = \boxed{1.1}00\cdots 0$$

$\therefore$ $[-\frac{1}{2}]_{补}$ 不是规格化的数

$$S = -1$$

$$[S]_{补} = \boxed{1.0}00\cdots 0$$

$\therefore$ $[-1]_{补}$ 是规格化的数

例如 $x = 0.1101 \times 2^{01}$　　$y = (-0.1010) \times 2^{11}$

求 $x+y$

解：$[x]_{补} = 00, 01; 00.1101$　　$[y]_{补} = 00, 11; 11.0110$

1. 对阶

① 求阶差 $[\Delta j]_{补} = [j_x]_{补} - [j_y]_{补}$

$$\begin{array}{rr} & = 00, 01 \\ + & 11, 01 \\ \hline & 11, 10 \end{array}$$

阶差为负（-2）　$\therefore S_x \longrightarrow 2$　$j_x + 2$

② 对阶　$[x]_{补'} = 00, 11; 00.0011$

2. 尾数求和

$$\begin{array}{rll} & [S_x]_{补'} & = 00.0011 \\ + & [S_y]_{补} & = 11.0110 \\ \hline & & 11.1001 \end{array}$$

对阶后的$[S_x]_{补'}$

$\therefore [x+y]_{补} = 00, 11; 11. 1001$

(3) 左规

尾数左移一位，阶码减 1，直到数符和第一数位不同为止

上例 $[x+y]_{补} = 00, 11; 11.1001$

左规后 $[x+y]_{补} = 00, 10; 11.0010$

$$\therefore x + y = (-0.1110) \times 2^{10}$$

(4) 右规

当 尾数溢出（ >1）时，需 右规

即尾数出现 01. ×× …×或 10. ×× …×时

尾数右移一位，阶码加 1

$$x = 0.1101 \times 2^{10} \quad y = 0.1011 \times 2^{01}$$

求 $x+y$（除阶符、数符外，阶码取 3 位，尾数取 6 位）

解： $[x]_{补} = 00, 010; 00.\ 110100$

$[y]_{补} = 00, 001; 00.\ 101100$

① 对阶

$$[\Delta j]_{补} = [j_x]_{补} - [j_y]_{补} \quad \begin{array}{r} = \ 00, 010 \\ +\ 11, 111 \\ \hline \boxed{1}00, 001 \end{array}$$

阶差为 +1 $\quad \therefore S_y \longrightarrow 1,\ j_y+1$

$\therefore [y]_{补'} = 00, 010; 00.\ 010110$

② 尾数求和

$$\begin{array}{rl} [S_x]_{补} = & 00.\ 110100 \\ +\ [S_y]_{补'} = & 00.\ 010110 \\ \hline & 01.\ 001010 \end{array}$$

对阶后的 $[S_y]_{补'}$

尾数溢出需右规

③ 右规

$$[x+y]_{补} = 00, 010; 01.001010$$

右规后

$$[x+y]_{补} = 00, 011; 00.100101$$

$$\therefore x+y = 0.100101 \times 2^{11}$$

4. 舍入

在 对阶 和 右规 过程中，可能出现 尾数末位丢失 引起误差，需考虑舍入

(1) 0 舍 1 入法

(2) 恒置“1”法

$$x = (-\frac{5}{8}) \times 2^{-5} \quad y = (\frac{7}{8}) \times 2^{-4}$$

求 $x-y$（除阶符、数符外，阶码取 3 位，尾数取 6 位）

解： $x = (-0.101000) \times 2^{-101}$ $\quad$ $y = (0.111000) \times 2^{-100}$

$[x]_补 = 11, 011; 11.011000$ $\quad$ $[y]_补 = 11, 100; 00.111000$

① 对阶

$$[\Delta j]_补 = [j_x]_补 - [j_y]_补 = \begin{array}{r} 11, 011 \\ +\ 00, 100 \\ \hline 11, 111 \end{array}$$

阶差为 -1 $\quad$ $\therefore S_x \longrightarrow 1, \quad j_x + 1$

$\therefore [x]_{补'} = 11, 100; 11.101100$

② 尾数求和

$$
\begin{array}{r}
[S_x]_{补'} = 11.\ 101100 \\
+\ \ [-S_y]_{补} = 11.\ 001000 \\
\hline
\boxed{1}10.\ 110100
\end{array}
$$

③ 右规

$$[x-y]_{补} = 11,\ 100;\ 10.\ 110100$$

右规后

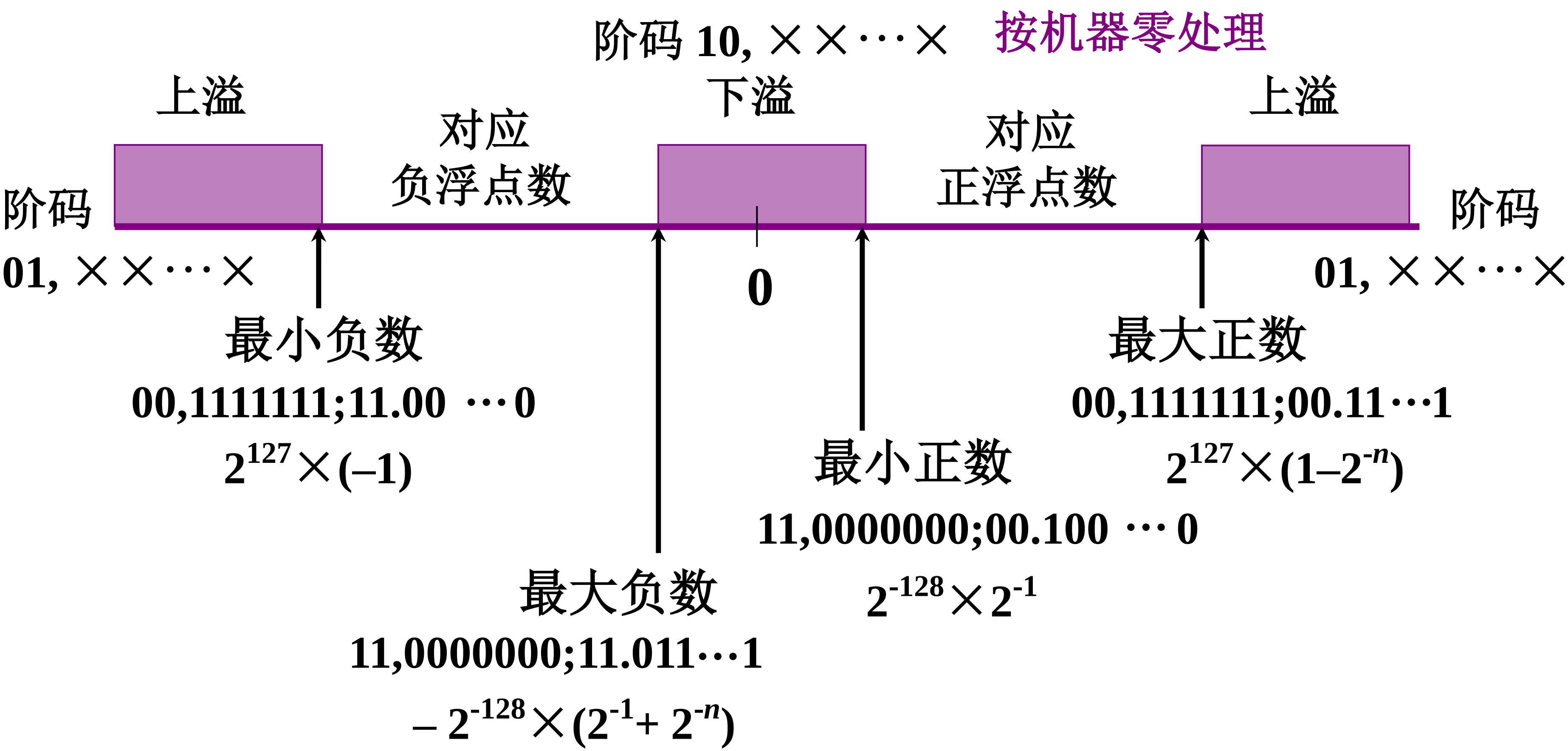
$$[x-y]_{补} = 11,\ 101;\ 11.\ 011010$$

$$\therefore\ x-y = (-0.100110)\times 2^{-11}$$

$$= (-\frac{19}{32})\times 2^{-3}$$

5. 溢出判断

设机器数为补码，尾数为 规格化形式，并假设阶符取 2 位，阶码的数值部分取 7 位，数符取 2 位，尾数取 *n* 位，则该 补码 在数轴上的表示为



二、浮点乘除运算

$$x = S_x \cdot 2^{j_x} \qquad y = S_y \cdot 2^{j_y}$$

1. 乘法

$$x \cdot y = (S_x \cdot S_y) \times 2^{j_x + j_y}$$

2. 除法

$$\frac{x}{y} = \frac{S_x}{S_y} \times 2^{j_x - j_y}$$

3. 步骤

(1) 阶码采用 补码定点加（乘法）减（除法）运算

(2) 尾数乘除同 定点 运算

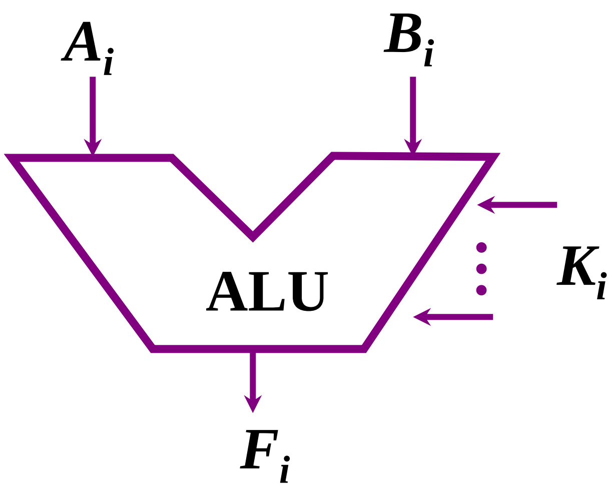
(3) 规格化

4. 浮点运算部件

阶码运算部件，尾数运算部件

5.2.3 算术逻辑单元

一、ALU 电路



组合逻辑电路

K_i 不同取值

F_i 不同

四位 ALU 74181

$M = 0$ 算术运算

$M = 1$ 逻辑运算

$S_3 \sim S_0$ 不同取值，可做不同运算