计算机组成与系统结构

第二章 运算方法和运算器

吕昕晨

lvxinchen@bupt.edu.cn

网络空间安全学院

第二章 运算方法和运算器

2.5 定点运算器组成

- 内部总线
- 定点运算器基本结构

2.6 浮点运算与浮点运算器

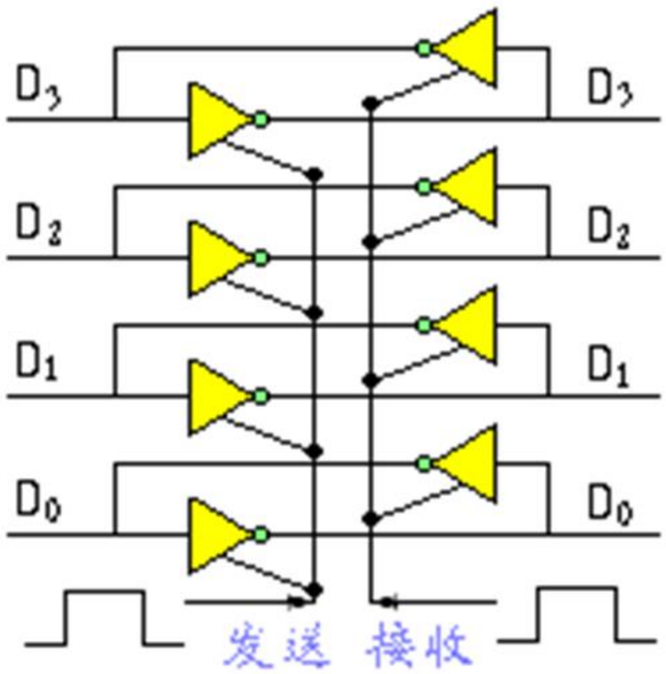
- 浮点加、减法
- 浮点乘、除法
- 浮点运算流水线结构

2.5.3 内部总线

- 内部总线
  - 机器内部各部份数据传送频繁,可以把寄存器间的数据传送通路加以归并,组成总线结构。
  - 分类
    - 所处位置
      - 内部总线（CPU内）
      - 外部总线（系统总线）
    - 逻辑结构
      - 单向传送总线
      - 双向传送总线

带缓冲的双向数据总线

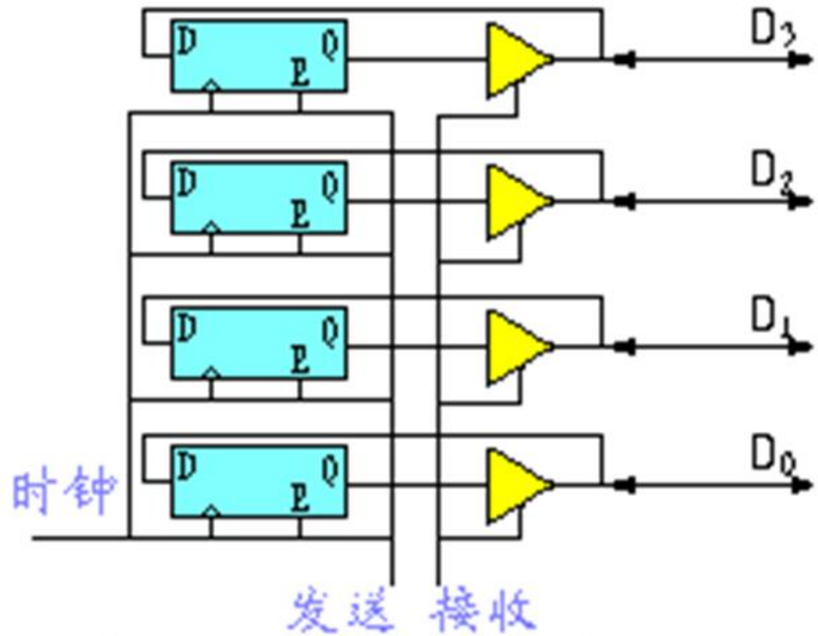
- 三态逻辑电路
  - 高阻态（等效断路）
  - 连通态（输出0/1）
  - 开关功能
  - 数据选择器、总线结构
- 4位带缓冲双向数据总线
  - 8个三态缓冲器（发送、接收）
  - 控制端
    - 发送信号
    - 接收信号



(a) 带有缓冲器的双向数据总线

带锁存器的双向数据总线

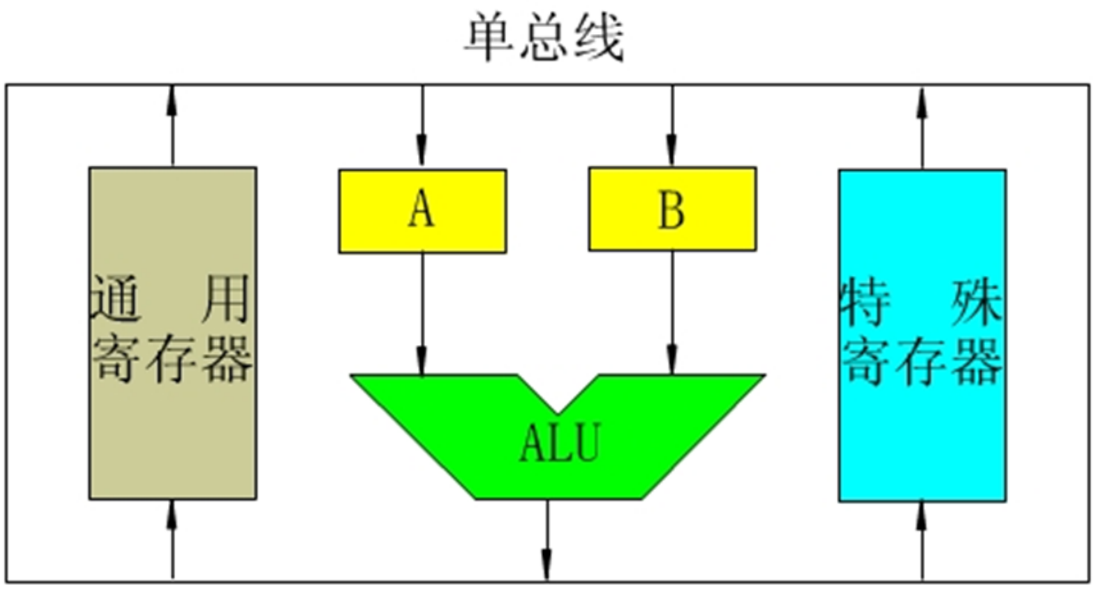
- 三态逻辑电路
- DE触发器
  - D触发器
    - 时钟上升沿将D端存入
    - Q端可输出存入数据
    - 寄存器
  - 使能端口：E（控制D端输入）
- 功能
  - 当E=1时，接收功能（三态缓冲关）
  - 当E=0时，发送（三态缓冲开）



(b) 带有锁存器的4位双向数据总线

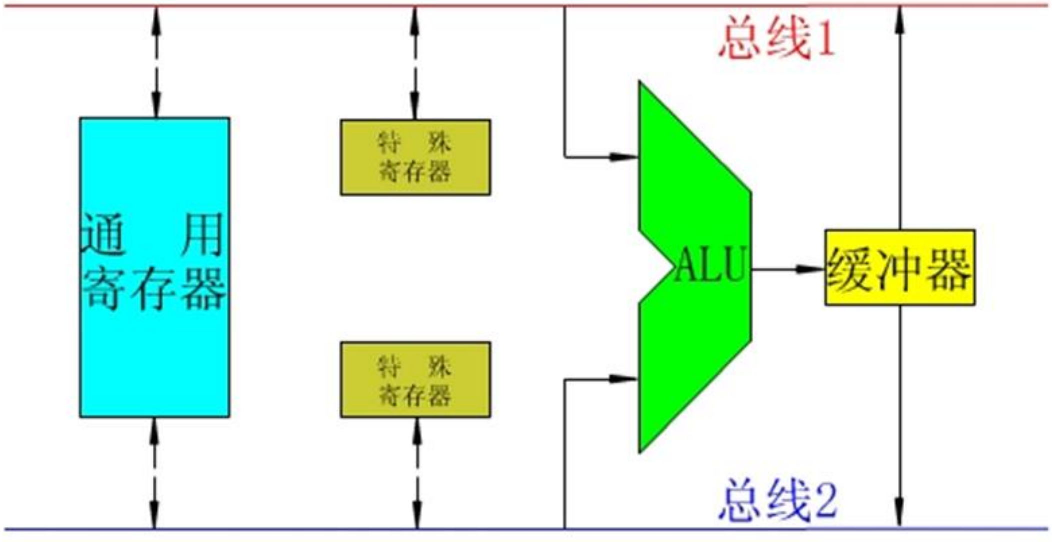
单总线结构运算器

- 特点：所有部件连接到同一总线上，控制简单
- 性能
  - 同一时刻仅允许一个操作数出现在总线上
  - 数据存入A、B寄存器：2个时钟周期
  - 结果输出：1个时钟周期



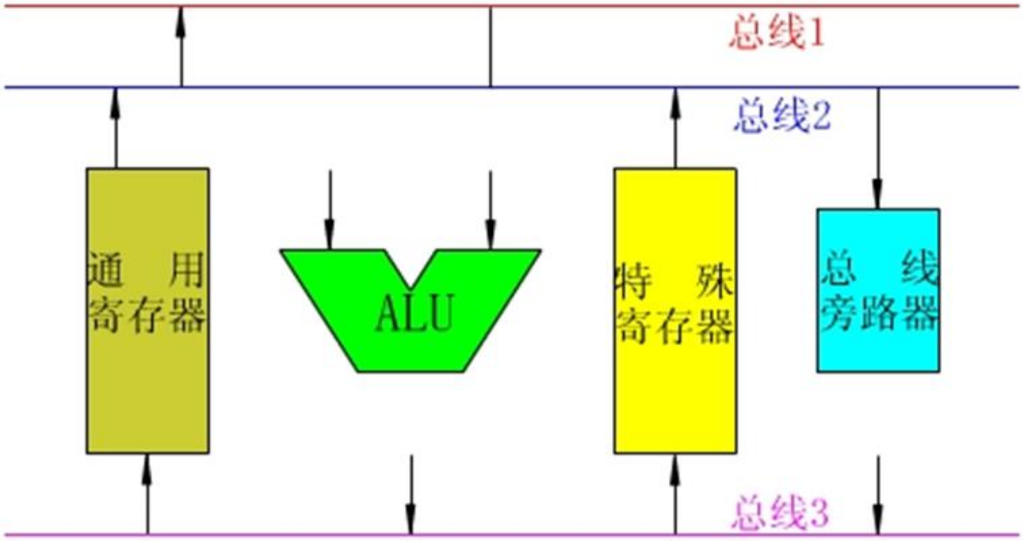
双总线结构运算器

- 特点：ALU输入端由不同总线连接
- 性能
  - 两个操作数可同时送入ALU
  - 数据输入：1个时钟周期
  - 结果输出：1个时钟周期



三总线结构运算器

- 特点：ALU输入端、输出端由不同总线连接
- 性能
  - 1个时钟周期，进行输入输出
  - 选通脉冲，考虑ALU延迟
  - 总线旁路器：操作数不需要修改



第二章 运算方法和运算器

2.5 定点运算器组成

- 内部总线
- 定点运算器基本结构

2.6 浮点运算与浮点运算器

- 浮点加、减法
- 浮点乘、除法
- 浮点运算流水线结构

浮点加减法运算

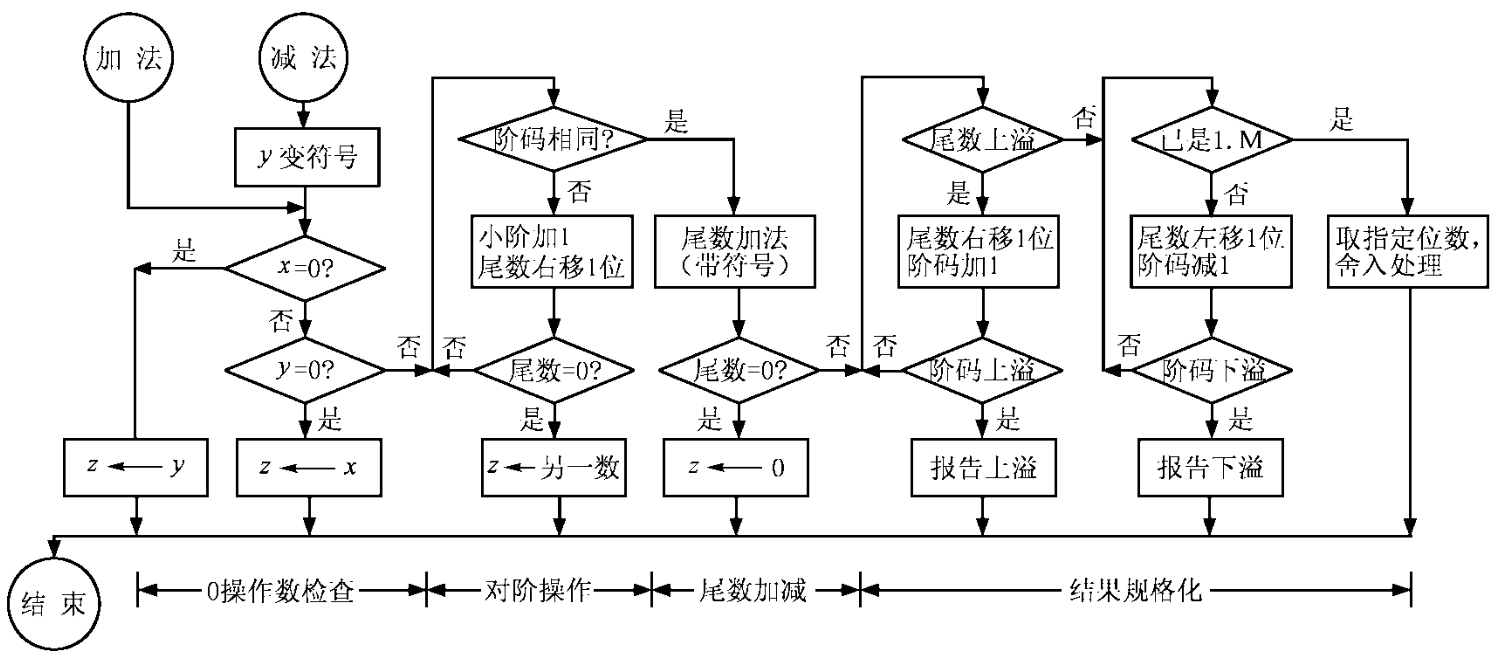
- 设有两个浮点数 x 和 y，它们分别为
  - $x = 2^{Ex} \cdot M_x$
  - $y = 2^{Ey} \cdot M_y$
- Ex 和 Ey 分别为数 x 和 y 的阶码
- M_x 和 M_y 为数 x 和 y 的尾数

- 两浮点数进行加法和减法的运算规则是
- $x \pm y = (M_x 2^{Ex - Ey} \pm M_y) 2^{Ey}$
- 设 $Ex <= Ey$
- 不失一般性，统一成较大数阶码

浮点加减法运算算运算步骤

- 操作数检查;
- 比较阶码并完成对阶（小阶向大阶对齐）;
- 尾数求和运算;
- 结果规格化;
- 舍入处理

$$x \pm y = (M_x 2^{Ex-Ey} \pm M_y) 2^{Ey}$$



回顾：溢出判断（双符号位）

- 补码采用双符号位，为了对溢出进行判断
- 00　为正　　11　为负
- 01　上溢　　10　下溢

$x=+011, y=+110$,求$[x+y]_{补}$和$[x-y]_{补}$,并判断是否溢出。

$[x]_{补}=00011, [y]_{补}=00110, [-y]_{补}=11010$

$[x+y]_{补}=[x]_{补}+[y]_{补}=01001$, 结果上溢。

$[x-y]_{补}=[x]_{补}+[-y]_{补}=11101$, 结果正确,为$-3$。

浮点加减法示例一（1）

[例28] 设x = 2^{010}×0.11011011,y=-2^{100}×0.10101100，求x+y

1、0操作数检查（非0）

2、对阶：阶码对齐后才能加减。规则是阶码小的向阶码大的数对齐；

- 若△E = 0,表示两数阶码相等,即E x = E y；
- 若△E>0,表示E x >E y；
- 若△E<0,表示E x >E y 。
- 当E x ≠E y 时,要通过尾数的移动以改变E x 或E y ,使之相等。

- $[x]_{浮}$→Ex=00010，Mx=0.11011011
- $[y]_{浮}$→Ey=00100，My=1.01010100（补码）；
- 阶差=$[Ex]_{补}$-$[Ey]_{补}$=00010-00100=00010+11100(取反加一)=11110
- 即阶差为-2，Mx右移两位，Ex加2。
- $[x]_{浮,对阶}$→Ex=00100，Mx=0.00110110(11)，阶码精度

浮点加减法示例一（2）

3、尾数相加（带符号位）

- 0. 0 0 1 1 0 1 1 0 (11)

  \+ 1. 0 1 0 1 0 1 0 0

  1. 1 0 0 0 1 0 1 0 (11)

4、结果规格化

- 规则
  - 尾数右移1位，阶码加1
  - 尾数左移1位，阶码减1
- 左规处理，结果为1.00010101(10)，阶码为00011

Ex=00100

浮点加减法示例一（3）

- 舍入处理（对阶和向右规格化时）
  - 就近舍入(0舍1入):类似”四舍五入”,丢弃的最高位为1,进1
  - 朝0舍入:截尾
  - 朝＋∞舍入:正数多余位不全为”0”,进1;负数,截尾
  - 朝－∞ 舍入:负数多余位不全为”0”,进1;正数,截尾

采用0舍1入法处理，得到1.00010110。

$\boxed{1.00010101(10)}$

- 溢出判断和处理
  - 阶码上溢，一般将其认为是＋∞和－∞ 。
  - 阶码下溢，则数值为0。

阶码符号位为00，不溢出。得最终结果为

$$x+y = 2^{011}\times(-0.11101010)$$

浮点加减法示例二

[例29] 设$x = 10^{Ex} \times Mx = 10^{2} \times 0.3$,

$y = 10^{Ey} \times My = 10^{3} \times 0.2$, 求 $x+y=?$ $x-y=?$

解： Ex=2, Ey=3, Ex<Ey, 对阶时小阶向大阶看齐。

$$x+y = (Mx \cdot 10^{Ex-Ey} + My) \times 10^{Ey}$$

$$= (0.3 \times 10^{2-3} + 0.2) \times 10^{3}$$

$$= 0.23 \times 10^{3} = 230$$

$$x-y = (Mx \cdot 10^{Ex-Ey} - My) \times 10^{Ey}$$

$$= (0.3 \times 10^{2-3} - 0.2) \times 10^{3}$$

$$= -0.17 \times 10^{3} = -170$$

单选题 1分

设置

$x=0.1101 \times 2^{01}$ ； $y=-0.1010 \times 2^{11}$

尾数和阶符都采用补码表示，都采用双符号位表示法，就近舍入规则。 求$x+y=?$

A. 0.1011×2^{01}

B. 0.1011×2^{10}

C. 0.1101×2^{01}

D. 0.1101×2^{10}

提交

习题求解过程

$[x]_{浮}$=0001，00.1101

$[y]_{浮}$=0011，11.0110

阶差=1110， 即为-2

Mx应当右移2位，

$[x]_{浮}$=0011，00.0011（01）

尾数和为11.1001（01）

左规11.0010（10），阶码减1为0010

舍入（就近舍入）11.0011 丢弃10

$x+y=-0.1101*2^{10}$

浮点乘法和除法运算

- 设有两个浮点数 x 和 y：

$$x = 2^{E_x} \cdot M_x$$

$$y = 2^{E_y} \cdot M_y$$

- $x \times y = 2^{(E_x + E_y)} \cdot (M_x \times M_y)$
- $x \div y = 2^{(E_x - E_y)} \cdot (M_x \div M_y)$
- 乘除运算分为四步
  ① 0操作数检查
  ② 阶码加减操作
  ③ 尾数乘除操作
  ④ 结果规格化和舍入处理

浮点乘法和除法运算示例一

[例30] 设有浮点数 $x = 2^{-5} \times 0.0110011$, $y = 2^{3} \times (-0.1110010)$

- 求$[x \times y]_{浮}$。
- 阶码用4位移码表示,尾数(含符号位)用8位补码表示。
- 要求用补码完成尾数乘法运算,运算结果尾数保留高8位(含符号位),并用尾数低位字长值处理舍入操作。

[解:]

阶码采用双符号位,尾数原码采用单符号位,则有

$[M_x]_{原}=0.0110011$, $[M_y]_{原}=1.1110010$

$[E_x]_{补}=11011$, $[E_y]_{补}=00011$

$[x]_{浮}=11011,0.0110011$, $[y]_{浮}=00011,1.1110010$

(1) 求阶码和：$[E_x]_{补}+[E_y]_{补}=11011+00011=11110$ (补码形式-2)

(2) 尾数乘法运算可采用原码阵列乘法器实现，即有

$[M_x]_{原}\times[M_y]_{原} = [0.0110011]_{原}\times[1.1110010]_{原}$

$= [1.0101101,0110110]_{原}$

(3) 规格化处理：乘积不是规格化的数，需要左规。尾数左移1位变为 1.1011010,1101100 , 阶码变为11101 (-3)。

(4) 舍入处理：尾数为负数，取高位字长，按舍入规则舍去低位字长，故尾数为1.1011011 。

最终相乘结果为 $[x\times y]_{浮}=11101,1.1011011$

其真值为 $x\times y=2^{-3}\times(-0.1011011)$

浮点乘法和除法运算示例二

- [例31] 设基数R=10, $x=10^{Ex}\times Mx=10^{2}\times 0.4$, $y=10^{Ey}\times My=10^{3}\times 0.2$, 用浮点法求$x\times y=?$ $x\div y=?$

- 解:
  - $Ex=2, Ey=3, Mx=+0.4, My=+0.2$
  - $x\times y=10^{(Ex+Ey)}\times(Mx\times My)=10^{2+3}\times(0.4\times 0.2)$ $=8000$
  - $x\div y=10^{(Ex-Ey)}\times(Mx\div My)= 10^{2-3}\times(0.4\div 0.2)$ $=0.2$

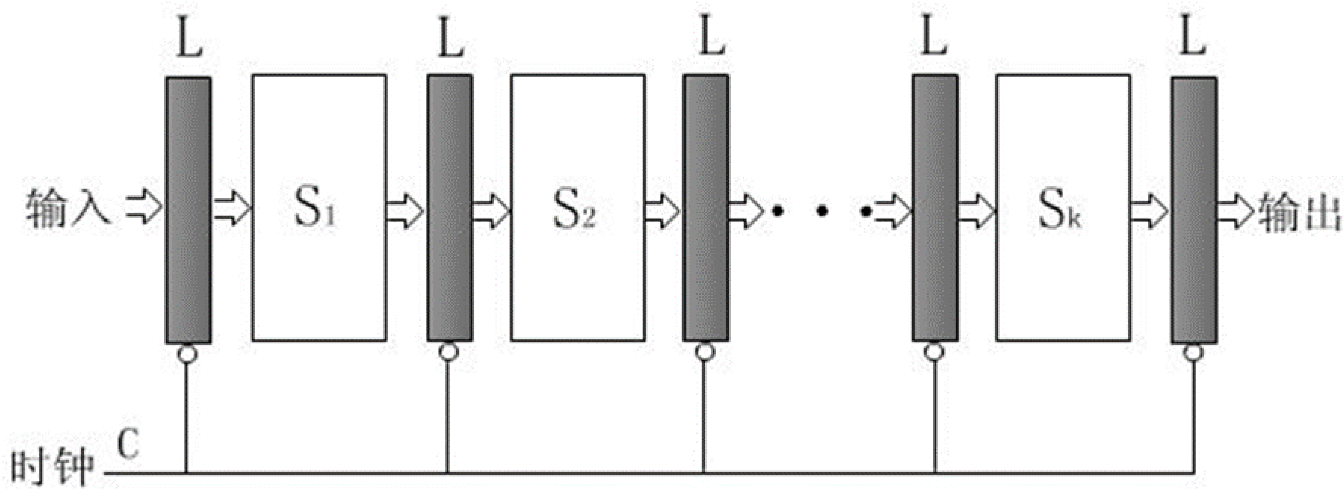
浮点运算流水线

- 提高并行性的两个渠道：
  - 空间并行性
    - 增加冗余部件，如增加多操作部件处理机和超标量处理机
  - 时间并行性
    - 改善操作流程如：流水线技术

流水线技术原理

- 在流水线中必须是连续的任务，只有不断的提供任务才能充分发挥流水线的效率
- 把一个任务分解为几个有联系的子任务。每个子任务由一个专门的功能部件实现
- 在流水线中的每个功能部件之后都要有一个缓冲寄存器，或称为锁存器（各级流水独立）
- 流水线中各段的时间应该尽量相等，否则将会引起“堵塞”和“断流”的现象
- 流水线需要有装入时间和排空时间，只有当流水线完全充满时，才能充分发挥效率

流水线原理



- 设过程段 S_i所需的时间为τ_i,缓冲寄存器的延时为τ_l,线性流水线的时钟周期定义为

$$\tau = \max\{\tau_i\} + \tau_l = \tau_m + \tau_l$$

- 流水线处理的频率为 $f = 1/\tau$。

流水线浮点运算器

$A = a \times 2^{P}$,　　$B = b \times 2^{q}$

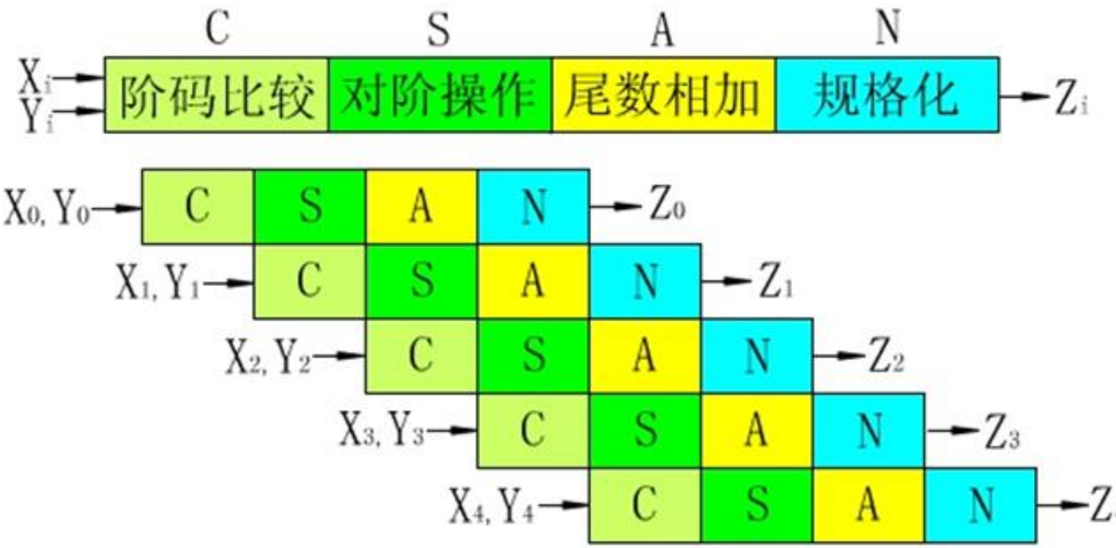
在4级流水线加法器中实现上述浮点加法时，分为以下操作：

(1) 求阶差

(2) 对阶

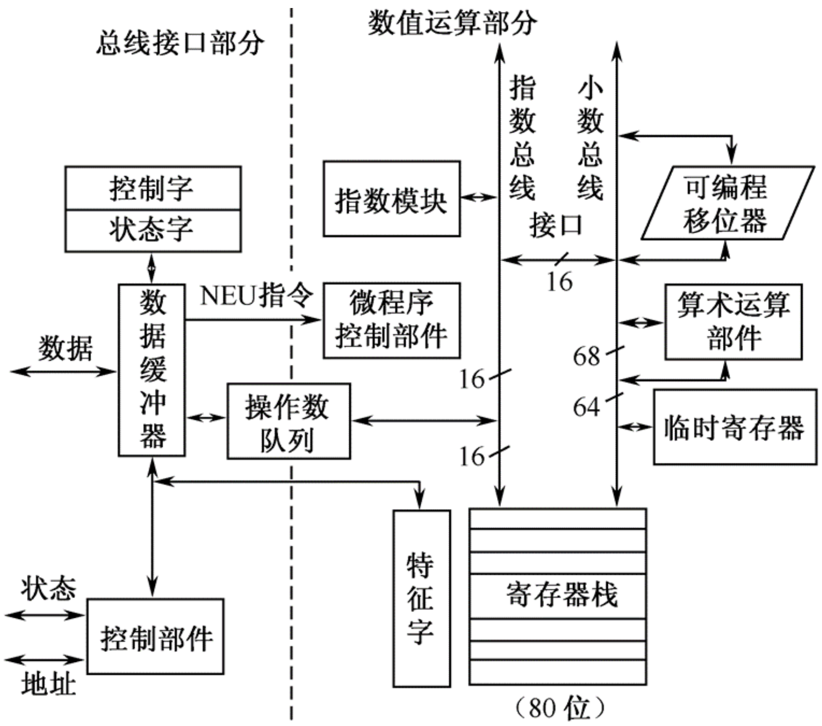
(3) 相加

(4) 规格化



浮点运算器实例

- 浮点运算器实例
  - CPU之外的浮点运算器（数学协处理器）如80287
    - 完成浮点运算功能，不能单用
    - 可以和80386或80286异步并行工作
    - 高性能的80位字长的内部结构。有8个80位字长以堆栈方式管理的寄存器组
    - 浮点数格式符合IEEE标准



课后作业

第二章作业（2）

2.7、2.8、2.9、2.10