国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第18卷第3期 1996年9月 Vol. 18 No. 3

# 巨型机中软硬结合的函数计算方案*

张民选

(国防科学技术大学计算机系 长沙 410073)

**摘 要** 本文分别讨论了巨型机中计算 $y/x$、$y/\sqrt{x}$、$a^x$、$\log_a x$、$x^y$ 等函数的算法设计及其误差分析。

**关键词** 巨型机，软硬结合，函数计算

**分类号** TP301.6

# Implemented Schemse with Software-Hardware for Function Computations in Supercomputer

Zhang Minxuan

(Department of Computer, Science NUDT, Changsha, 410073)

**Abstract** In this paper, the algorithms designs and error analysis on the $y/x$、$y/\sqrt{x}$、$a^x$、$\log_a x$、$x^y$functions in supercomputer are discussed.

**Key words** Supercomputer, Software-Hardware, Function Computation.

在流水线向量巨型机中，采用软硬结合、线性流水、向量链接的方法，实现$y/x$、$y/\sqrt{x}$、$a^x$、$\log_a x$、$x^y$ 的计算，获得了高速度、高精度及高性价比的效果。本文着重讨论这些函数的算法设计与误差分析。

为讨论方便，设浮点源操作数 $x$ 的阶码为 $P$、尾数为 $A$(48 位)，形式为 $x=2^P \cdot A=2^P \cdot 0.1a_2a_3\cdots a_{48}$，其中 $a_i$ 为 0 或 1。从源操作数 $x$ 中截去尾数低 36 位，并固定尾数第 12 位为 1，则得 $x_0$。设 $x_0$ 的尾数为 $A_0$ 则 $x_0=2^P \cdot A_0=2^P \cdot 0.1a_2a_3\cdots a_{11}1$。

## 1 $y/x$ 的算法设计和精度分析

### 1.1 倒数近似值指令

求 $x$ 的倒数近似值，采用的计算公式为

* 1996年1月6日收稿

$$1/x = 2^{-P} \cdot 1/A \approx 2^{-P} \cdot (H_0 + T(A_0 - A)) \tag{1}$$

式中 $H_0=1/2(1/A_0+(1/A_0-2^{-13}))(1-2^{-26}/A_0))$，

$$T=\begin{cases}1/(A_0(A_0-2^{-12}))，当(A_0-A)>0 时\\ 1/(A_0(A_0+2^{-12}))，当(A_0-A)\leqslant 0 时\end{cases}$$

在硬件实现中，截取 $T$ 的 $2^{-13}$ 位以上的高位参加运算，乘法器只设 $2^{-1}\sim 2^{-31}$ 位的硬件，则计算公式变为

$$1/x = 2^{-P} \cdot H_1^* = 2^{-P} \cdot [H_0 + T^*(A_0 - A)]^* \tag{2}$$

式中 $T^*=T-2^{-13}\delta_1$，$|\delta_1|<1$；

$[H_0+T^*(A_0-A)]^*=[H_0+T^*(A_0-A)]-2^{-31}\delta_2$，$|\delta_2|<11$。

用 $H_1^*$ 作为 $1/A$ 的近似值，相对误差 $\varepsilon(A)$ 为

$$\varepsilon(A) = |1 - A \cdot H_1^*| = |1 - A(H_0 + T(A_0 - A)) + 2^{-13}\delta_1 A(A_0 - A) + 2^{-31}\delta_2 A|$$
$$= |\varepsilon_1 + \varepsilon_2 + \varepsilon_3| \tag{3}$$

其中 $\varepsilon_1=1-A(H_0+T(A_0-A))$，为公式固有误差，

$\varepsilon_2=2^{-13}\delta_1 A(A_0-A)$，为 $T$ 截断引进的误差，

$\varepsilon_3=2^{-31}\delta_2 A$，为乘法运算截断误差。

当 $|A_0-A|\leqslant 2^{-12}$ 时，有 $|\varepsilon_1|<2^{-25}$，$|\varepsilon_2+\varepsilon_3|<2^{-25}+2^{-28}$。

在乘法器硬件设计中，控制 $\varepsilon_1$ 与 $(\varepsilon_2+\varepsilon_3)$ 的符号，使得公式误差与截断误差相互抵消，使求出的半精度倒数近似值具有 24 位精度，有 $\varepsilon(A)<(2^{-25}+2^{-28})$。

### 1.2 倒数迭代指令

源操作数 $x$、$h$ 均为规格化浮点数，且 $x$、$h$ 互为近似倒数，求倒数迭代因子 $(2-x\cdot h)$，硬件实现的公式为

$$C = (2 - 2^{-48}) - (x \cdot h - 2^{-48}) \tag{4}$$

指令实现中，$x\cdot h$ 采用截断乘的舍入方案，计算结果具有 48 位精度，记 $C^*$ 为 $C$ 的计算结果，则有 $C^*=C+2^{-48}\delta$，$0\leqslant\delta<1$。

### 1.3 除法序列和精度

在巨型机中，求 $f=y/x$ 的指令序列，有两种类型。

①先求半精度商，再求全精度商

$Z_1=(/H)X$，　　半精度倒数；

$H_1=Z_1(*F)Y$，　　半精度商；

$C=Z_1(*I)X$，　　迭代因子；

$H_2=H_1(*R)C$，　　全精度商。

②先求全精度倒数，再求全精度商

$Z_1=(/H)X$，　　半精度倒数；

$C=Z_1(*I)X$，　　迭代因子；

$Z_2=Z_1(*F)C$，　　全精度倒数；

$H_2=Z_2(*R)Y$，　　全精度商。

为了使计算结果不出现 $x/x\neq 1$ 的情况，设计中已保证 $-2^{-48}<(1-x/x)<2^{-47}$。

## 2 $y/\sqrt{x}$的算法设计和精度分析

### 2.1 平方根倒数指令

求函数 $f(x)=1/\sqrt{x}$ 的近似值，采用如下计算公式

$$f(x)=1/\sqrt{x}=\begin{cases}2^{-P/2}\cdot 1/\sqrt{A}, & P\text{ 为偶数},\\ 2^{-\left(\frac{P-1}{2}+1\right)}\cdot\sqrt{2}/\sqrt{A}, & P\text{ 为奇数}。\end{cases}$$

令

$$H=\begin{cases}1/\sqrt{A}, & P\text{ 为偶数},\\ \sqrt{2}/\sqrt{A}, & P\text{ 为奇数}\end{cases}$$

$$H\approx H_1=\begin{cases}H_0+T(A_0-A), & P\text{ 为偶数},\\ (\sqrt{2}H_0)+(\sqrt{2}T)\cdot(A_0-A), & P\text{ 为奇数}。\end{cases}\tag{5}$$

式中 $H_0=1/2(1/\sqrt{A_0}+1/\sqrt{A_0-2^{-13}}-2^{-13}\cdot T)$

$$T=\begin{cases}2^{12}(1/\sqrt{A_0-2^{-12}}-1/\sqrt{A_0}), & (A_0-A)>0;\\ 2^{12}(1/\sqrt{A_0}-1/\sqrt{A_0+2^{-12}}), & (A_0-A)\leqslant 0。\end{cases}$$

硬件实现时，$T$ 截取 $2^{-13}$位以上的高位，0 舍 1 入后参加运算，运算器只设 $2^{-1}\sim 2^{-31}$ 位的硬件，则计算公式为

$$H_1^*=\begin{cases}[H_0+T^*(A_0-A)]^*, & P\text{ 为偶数},\\ [(\sqrt{2}H_0)+(\sqrt{2}T)^*(A_0-A)]^*, & P\text{ 为奇数}。\end{cases}\tag{6}$$

此处 $T^*=T-2^{-14}\cdot\delta_1$，$|\delta_1|<1$；

$\sqrt{2}\cdot T^*=\sqrt{2}T-2^{-14}\cdot\delta_1$，$|\delta_1|<1$；

$H_1^*=H_1-2^{-31}\cdot\delta 2$，$|\delta_2|<11$。

用 $H_1^*$ 作为 $1/\sqrt{A}$ 的近似值，相对误差 $\varepsilon(A)$ 为

$$\begin{aligned}\varepsilon(A)&=|1-\sqrt{A}\cdot H_1^*|\\ &=|1-\sqrt{A}\cdot[H_0+T\cdot(A_0-A)]+2^{-14}\cdot\delta_1\cdot\sqrt{A}\cdot(A_0-A)+2^{-31}\cdot\delta_2\cdot\sqrt{A}|\\ &=|\varepsilon_1+\varepsilon_2+\varepsilon_3|。\end{aligned}\tag{7}$$

式中 $\varepsilon_1$ 为公式误差，$|\varepsilon_1|\leqslant 2^{-26}$；

$\varepsilon_2$ 为 $T$ 截断引进的误差，$|\varepsilon_2|<2^{-26}$；

$\varepsilon_3$ 为运算器截断引进的误差，$|\varepsilon_3|<2^{-26}$。

硬件实现时，通过对 $\delta_1$、$\delta_2$ 符号的控制，使公式误差与截断误差相抵消，求出的半精度平方根倒数近似值达到了 25 位精度，即有 $\varepsilon(A)<2^{-25}$。当 $P$ 为奇数时，也有同样结论。

### 2.2 平方根倒数迭代指令

源操作数 $x$ 和 $h$ 均为规格化数，且 $x$、$h$ 互为近似倒数。平方根倒数迭代因子 $C=(3-x\cdot h)/2$。硬件实现的公式为：

$$C=[(3-2^{-48})-(x\cdot h-2^{-48})]/2。\tag{8}$$

指令实现中，采用与舍入乘指令相同的补偿、舍入方案，计算结果 $C^*$ 与 $C$ 的关系为 $C^*=C+(2^{-49}+2^{-52})\delta$，$|\delta|\leqslant 1$。

### 2.3 $\sqrt{x}$、$1/\sqrt{x}$、$y/\sqrt{x}$的指令序列和精度

求$\sqrt{x}$只需四条指令便可实现。其指令序列为

$Z_1=(/Q)X$，　　半精度平方根倒数；

$H_1=Z_1(*R)X$，　　半精度平方根；

$C=Z_1(*Q)H_1$，　　迭代因子；

$H_2=C(*R)H_1$，　　全精度平方根。

若将第四条指令中的$H_1$用$Z_1$代替，则求得$1/\sqrt{x}$的结果。再用$y$乘$1/\sqrt{x}$，则得$y/\sqrt{x}$的结果。

$\sqrt{x}$结果的相对误差$\varepsilon(x)$为

$$\varepsilon(x)=|1-(1/\sqrt{x})\{(xZ_1+\delta_1)Z_1^2[1/2(3-Z_1(Z_1\cdot x+\delta_1)+\delta_2]+\delta_3\}|$$

$$=|\varepsilon_1(x)-[(3-2xZ_1^2)/(2\cdot\sqrt{x})]\delta_1-\sqrt{x}Z_1\delta_2-1/\sqrt{x}\delta_3| \qquad (9)$$

式中　$\varepsilon_1(x)$为迭代公式误差，$0\leqslant\varepsilon_1(x)<3/2\cdot\varepsilon_0^2(x)$

已知$|\varepsilon_0(x)|<2^{-25}$；则$0\leqslant\varepsilon_1(x)<2^{-50}+2^{-51}$

$\delta_1$为第二条指令的计算误差，$|\delta_1|<2^{-49}+2^{-50}$；

$\delta_2$为迭代因子的计算误差，$|\delta_2|<2^{-49}+2^{-52}$；

$\delta_3$为第四条指令的计算误差，$|\delta_3|<2^{-49}+2^{-50}$；

从而可知，$\varepsilon(x)<2^{-46}$，即平方根结果具有46位精度。容易推得平方根倒数结果也具有46位精度。当源操作数为“0”时，无需特殊处理就可获得正确结果。

## 3　$a^x$、$\log_a x$、$x^y$的算法设计与精度分析

### 3.1　$2^A$指令

源操作数$A$为64位的定点数补码，最低16位为小数，高48位为整数。当整数大于$-2^{14}$时，结果送全“0”，当整数大于$(2^{13}-2)$时，产生浮点错条件。

设　$A=a_0a_1a_2\cdots a_{47}\cdot a_{48}\cdots a_{63}$，则

$$2^A=2^{a_0a_1a_2\cdots a_{47}\cdot a_{48}\cdots a_{63}}=2^P\cdot(2^{Y_h}\cdot 2^{Y_l}) \qquad (10)$$

式中　$P=\begin{cases}a_0a_0a_{33}a_{34}\cdots a_{47}, & P\leqslant 2^{13}-2;\\ 1100\cdots 0, & P>2^{13}-2;\end{cases}$

$Y_h=2^{Y_h}=2^{0\cdot a_{48}a_{49}\cdots a_{55}}$；

$Y_l=2^{Y_l}=2^{0\cdot a_{56}a_{57}a_{63}\cdot 2^{-8}}$，

$2^A$的截断误差$\delta(A)$为：

$$\delta(A)=|2^A-2^P[(Y_h+2^{-49}\delta_1)(Y_l+2^{-49}\delta_2)+2^{-49}\delta_3]|$$

$$=|-2^P(2^{-49}\delta_1Y_l+2^{-49}\delta_2Y_h+2^{-49}\delta_3)| \qquad (11)$$

已知，$|\delta_1|\leqslant 1$，$|\delta_2|\leqslant 1$，$|\delta_3|<1$，从而有：$\delta(A)<|-2^P(2^{-48}+2^{-49})|$。则$2^A$的计算结果具有47位精度。

### 3.2　$\log_2 x$指令

源操作数$x$为规格化的浮点数，实际用于计算的操作数$x^*$为$2^P\cdot 0.1a_2a_3\cdots a_{11}$，采用如下公式计算：

$$\log_2 x^* = \begin{cases} P + \log_2 0.1a_2a_3\cdots a_{11}, & P \text{ 为正偶数或小于等于 } 0; \\ (P-1) + \log_2 1.a_2a_3\cdots a_{11}, & P \text{ 为正奇数}; \end{cases}$$
$$= P^* + Y_0 \tag{12}$$

式中 $P^*$：为源操作数阶码值或阶码值减 1 后化成的规格化浮点数；

$Y_0$：为 $\log_2 0.1a_2a_3\cdots a_{11}$ 或为 $\log_2 1.a_2a_3\cdots a_{11}$，是规格化的浮点数。

该指令在浮加部件执行，$Y_0$ 查表得到，$P^*$ 用硬件实现。计算结果具有 48 位精度，即 $|\log_2 x^* - (P^* + Y_0)| \leqslant 2^{-48}$。

为适应计算 $x$ 全精度对数算法的需要，当 $1-2^{-12} < x < 1+2^{-11}$ 时，将 $\log_2 x$ 指令的计算结果强置为全“0”。

### 3.3 指数 $a^x$ 的算法与精度

求函数 $e^x$、$10^x$、$2^x$ 的值，统称为求 $a^x$ 的值，采用的计算公式是

$$a^x = 2^{x\cdot\log_2 a}$$
$$= 2^{2^{-16}[x\cdot 2^{16}\log_2 a}\text{取整数}\cdot 2^{2^{-16}\{x\cdot 2^{16}\log_2 a\}}\text{取小数} \tag{13}$$

其计算步骤为

① $s = x\cdot 2^{16}\log_2 a$；

② $N = \mathrm{FIX}(s)$，取整数部分；

③ $y_1 = 2^{2^{-16}N}$，用 $2^x$ 指令计算；

④ $s_0 = \mathrm{FLT}(N)$，化成浮点；

⑤ $t = s - s_0$，取小数部分；

⑥ $y_2 = 2^{2^{-16}t} = C_0 + C_1 t + C_2 t^2$；

⑦ $a^x = y_1 y_2$。

计算结果的相对误差 $\varepsilon(x)$

$$\varepsilon(x) = |1/a^x[a^x - (2^{2^{-16}N}(1-\varepsilon_2)(1-\varepsilon_3)2^{2^{-16}t-\varepsilon_1 x\log_2 a})]|$$
$$= |1 + 2^{-\varepsilon_1 x\log_2 a}\cdot(\varepsilon_2 + \varepsilon_3 - \varepsilon_2\varepsilon_3 - 1)|$$
$$= |1 + (1+\varepsilon_1)(\varepsilon_2 + \varepsilon_3 - \varepsilon_2\varepsilon_3 - 1)|$$
$$= |\varepsilon_2 + \varepsilon_3 - \varepsilon_1| \tag{14}$$

式中 $(1+\varepsilon_1) = 2^{-\varepsilon_1 x\log_2 a}$

$\varepsilon_1$：为第一步乘法引进的相对误差；

$\varepsilon_2$：为 $2^x$ 指令引进的相对误差，$\varepsilon_2 < 2^{-47}$；

$\varepsilon_3$：为多项式计算引进的相对误差，$\varepsilon_3 < 2^{-48}$。

可知 $2^x$、$10^x$、$e^x$ 的计算结果具有 46 位精度，即 $\varepsilon(x) < 2^{-46}$。

### 3.4 对数 $\log_a x$ 的算法与精度

利用换底公式 $\ln x = \log_2 x\cdot\ln 2$，

$$\lg x = \log_2 x\cdot\lg 2。 \tag{15}$$

可以在计算 $\log_2 x$ 的基础上，容易求得 $\ln x$、$\lg x$。计算 $\log_2 x$ 的公式为

$$\log_2 x = \log_2 x^* + \log 2[(1+t)/(1-t)]; \tag{16}$$
$$t = (x - x^*)/(x + x^*)$$

对数的计算步骤为

① $y_1=\log_2 x^*$；

② $t=\begin{cases}(x-x^*)/(x+x^*), & y_1\neq 0;\\ (x-1.0)/(x+1.0), & y_1=0;\end{cases}$

③ $y_2=C_1t+C_3t^3$；

④ $\log_2 x=y_1+y_2$；

⑤ $\ln x=\log_2 x\cdot\ln 2$；$\lg x=\log_2 x\cdot\lg 2$；

$\log_a x$ 计算结果的相对误差 $\varepsilon_a(x)$ 为：

$$\begin{aligned}\varepsilon_a(x) &= |(\log_a x-\log_2 x(1+\varepsilon_2(x))(1+\delta_a)\log_a 2)/\log_a x|\\ &= \left|\varepsilon_a-\varepsilon_1\cdot\log_2 x^*/\log_2 x-\varepsilon_2\cdot\log\left|\frac{1+t}{1-t}\right|/\log_2 x\right|\end{aligned}\tag{17}$$

式中 $\varepsilon_1$：为 $\log_2 x$ 指令的相对误差，$|\varepsilon_1|<2^{-48}$

$\varepsilon_2$：为多项式计算结果的相对误差，$|\varepsilon_2|<2^{-48}+2^{-49}$

$\delta_a$：为 $\log_a 2$ 常数的相对误差，$|\delta_a|<2^{-49}$

可知 $\log_2 x$、$\ln x$、$\lg x$ 的计算结果，具有 46 位精度，即 $\varepsilon_a(x)<2^{-46}$，$a=2$，e，10。

### 3.5 幂函数 $x^y$ 的算法与精度

幂函数 $x^y$，在 $x>0$ 时，有 $x^y=2^{y\cdot\log_2 x}$ 用 $\log_2 x$ 和 $2^z$ 的计算方法组合起来计算幂函数，其计算结果的相对误差为

$$\begin{aligned}\varepsilon(x) &= |1-2^{-y\log_2 x}\cdot(1+\varepsilon_2)2^{y(\log_2 x(1+\varepsilon_1))}|\\ &= |1-2^{\varepsilon_1 y\log_2 x}(1+\varepsilon_2)|\end{aligned}\tag{18}$$

其中 $\varepsilon_1$ 为计算 $\log_2 x$ 的相对误差，$|\varepsilon_1|<2^{-46}$；

$\varepsilon_2$ 为计算 $2^z$ 的相对误差，$|\varepsilon_2|<2^{-46}$；

当 $|y\cdot\log_2 x|\leqslant 1$ 时，$x^y$ 的计算结果具有 44 位精度，当有

$$2^{n-1}<|y\cdot\log_2 x|\leqslant 2^n,\quad n=2,3,\cdots,13。$$

则有 $\varepsilon(x)<2^{-(45-n)}$，可知在最坏情况下，幂函数的计算结果至少有 32 位精度。

## 参 考 文 献


1 张民选，王久林. 流水线向量机浮点运算精度控制. 计算机工程与科学，1995，(4)

2 张民选，李晓梅. 软硬结合的迭代除法方案及其精度分析. 国防科技大学学报，1989，(1)

3 Wang W F. Fast Hardware-Based Alogrithms for Elementary Function Computations Using Rectangular Multipliers. IEEE Trans Comput，1994，43(3)

4 Wang D，Hynn M J. Fast division using accurate quotient approximations to reduce the number iterations. in proc. Tenth IEEE Symp. Comput. Arith，1991：191－201

5 Koren I，Zinaty O. Evaluating elementary functions in a numerical coprocessor based on rational approximations. IEEE Trans. Comput. 1990. 39：1030～1037

6 Tang P T P. Table-lookup algorithms for elementary functions and their error analysis. Argonne Nat. Lab. Rep.，MCS-P194－1190，1991



（责任编辑 张 静）