第 26 卷第 2 期 电子与信息学报 Vol.26No.2
2004 年 2 月 Journal of Electronics & Information Technology Feb. 2004

嵌入式协处理器初等函数的快速统一实现 [1]

朱樟明 周 端* 杨银堂 徐阳扬*

(西安电子科技大学微电子研究所 西安 710071)
*(西安电子科技大学计算机学院 西安 710071)

摘 要: 该文通过迭代次序的改进和有理近似的引入，提出了改进的 CORDIC 算法. 根据改进的 CORDIC 算法，实现了初等函数的快速统一运算，采用 Verilog-HDL 完成了 16 位数值协处理器 IP 核的设计. 该 IP 核具有规模较小、高速和控制简单的特点，非常适合嵌入式应用.

关键词: 嵌入式协处理器，坐标旋转机，初等函数，统一实现
中图分类号: TN794 **文献标识码**: A **文章编号**: 1009-5896(2004)02-0298-05

The Unified Fast Implementation of Elementary Functions in the Embeded Coprocessor

Zhu Zhang-ming Zhou Duan* Yang Yin-tang Xu Yang-yang*

(*Institute of Microelectronics, Xidian University, Xi'an* 710071, *China*)

*(*School of Computer, Xidian University, Xi'an* 710071, *China*)

Abstract In this paper, the unified CORDIC algorithm is put forward by the improvement of iterative sequence and the presence of rational approximation. The unified algorithm of elementary functions is implemented with high speed. A coprocessor core using the improved algorithm is built by Verilog HDL. The core is very suitable for embedded application due to its small scale, high speed and simple control.

Key words Embeded coprocessor, CORDIC, Elementary functions, Unified implement

1 引言

在现代信号处理和控制领域中，为了对大量数据进行实时处理，并考虑数据计算精度，需要数值协处理器或包含协处理器的微处理器，所以研制高速高精度的嵌入式协处理器具有重要的意义。

实现高速高精度数值协处理器关键在于初等函数的快速高精度运算，初等函数的实现方法有两类：一类是基于多项式迭代，另一类是基于移位相加迭代。前者速度较快，主要部件是乘法器，但是允许精度低，硬件实现规模大，难以嵌入式应用；后者速度较慢，主要部件是加法器和移位器，但是允许精度高，硬件实现规模小，便于嵌入式应用。本文从嵌入式应用出发，引入有理近似，改进迭代次序，提出了改进的 CORDIC 算法，并增加了专门的尾数乘除部件，从而实现初等函数的高速高精度统一运算。

2 坐标旋转机 (CORDIC) 算法

CORDIC 算法最早是由 Volder[1] 提出的，后来 HP 公司的 Walther[2] 等人对它进行了发展和应用， Intel 公司的 80X87 及奔腾系列微处理器都应用了此算法 [3] 。

[1] 2002-10-07 收到， 2003-04-18 改回
教育部跨世纪人才基金项目资助课题

CORDIC 算法是一种不恢复移位相加迭代算法，具体迭代过程如下：

$$\left.\begin{aligned} x_{i+1} &= x_i - d_i y_i 2^{-i} \\ y_{i+1} &= y_i + d_i x_i 2^{-i} \\ z_{i+1} &= z_i - d_i \theta_i \end{aligned}\right\} \tag{1}$$

其中 $d_i = \pm 1$, θ_i 为根据不同旋转坐标系而定的一系列常数，具体值如下：

$$\left.\begin{aligned} \theta_i &= \text{arc tan}2^{-i}, & m = 1 \\ \theta_i &= 2^{-i}, & m = 0 \\ \theta_i &= \text{arc tanh}2^{-i}, & m = -1 \end{aligned}\right\} \tag{2}$$

其中 $m = 1$ 时为圆坐标系，$m = 0$ 时为直线坐标系，$m = -1$ 时为双曲坐标系。

此外，CORDIC 算法有二次循环和 K 运算两组补充迭代方程，具体迭代如下：

$$\left.\begin{aligned} x_{(i+1),2} &= x_{(i+1),1} - d_i y_{(i+1)} 2^{-i} \\ y_{(i+1),2} &= y_{(i+1),1} + d_i x_{(i+1),1} 2^{-i} \\ z_{(i+1),2} &= z_{(i+1)} - d_i \text{arc tanh}2^{-i} \end{aligned}\right\} \qquad \left.\begin{aligned} x_{n+1} &= K x_n \\ y_{n+1} &= K y_n \\ z_{n+1} &= z_n + 0 \end{aligned}\right\} \tag{3}$$

其中二次循环的条件是 $i = 4, 13, 40, 121, \cdots, l, 3l+1, \cdots$；$K$ 运算的 K 值根据坐标不同而不同，具体如下：

$$\left.\begin{aligned} K &= \prod_{i=0}^{n} (1 + 2^{-2i})^{-1/2}, & m = 1 \\ K &= 1, & m = 0 \\ K &= \prod_{i=1}^{n} (1 - 2^{-2i})^{-1/2}, & m = -1 \end{aligned}\right\} \tag{4}$$

根据上述迭代方程，经过足够的迭代次数后，并通过选择适当的初始值，就可得到乘除，$\sin z$, $\cos z$, $\sinh z$, $\cosh z$ 等函数，再对这些函数进行组合，就可得到其它初等函数 [4−8]。

3 初等函数的快速统一实现

本文从嵌入式应用出发，引入有理近似，改进迭代次序，提出了改进的 CORDIC 算法，统一实现三角函数、双曲函数的快速运算。四则运算不采用 CORDIC 算法，其中乘法采用 Booth 算法，除法采用 SRT 算法。

3.1 三角函数的快速实现

三角函数运算主要包括 $\sin\alpha$, $\cos\alpha$, $\tan\alpha$ 和 $\tan^{-1}\alpha$，本文只对 $\sin\alpha$, $\cos\alpha$ 和 $\tan\alpha$ 的实现过程进行说明，而 $\tan^{-1}\alpha$ 是 $\tan\alpha$ 的反函数，可由 $\tan\alpha$ 的迭代公式逆向推导出 $\tan^{-1}\alpha$ 的迭代公式。

$\tan\alpha$ 的实现采用圆坐标下改进的 CORDIC 算法，保证运算速度和运算精度，具体实现步骤如下：

3.1.1 预处理　CORDIC 算法在圆坐标中进行坐标旋转 $(Z \to 0)$，可实现 $\sin\alpha$, $\cos\alpha$, $\tan\alpha$ 的运算。由于 CORDIC 算法只能计算 $-45° \sim 45°$ 的值，所以在进行 CORDIC 迭代之前进行预处理，将范围扩大至 $-180° \sim 180°$，下面是具体的预处理迭代公式。

$$\left.\begin{aligned} x_{01} &= -\text{sgn}(z_0) y_0 \\ y_{01} &= \text{sgn}(z_0) x_0 \\ z_{01} &= z_0 - \text{sgn}(z_0)\pi/2 \end{aligned}\right\} \qquad \left.\begin{aligned} x &= x_{01} - \text{sgn}(z_{01}) y_0 \\ y &= y_{01} + \text{sgn}(z_{01}) x_{01} \\ z &= z_{01} - \text{sgn}(z_{01})\pi/4 \end{aligned}\right\} \tag{5}$$

$-180^\circ \sim 180^\circ$ 内的任意角度值经以上二次迭代以后，都将映射成 $-45^\circ \sim 45^\circ$ 范围内的值。

3.1.2 Z 迭代 在 CORDIC 算法中，X, Y 迭代过程只是用到 Z 迭代产生的方向因子 d_i，所以可先对 Z 进行迭代，并将方向因子 d_i 存放在移位寄存器中。从迭代过程看，Z 迭代就是对 Z 进行如下过程的分解：

$$Z = \sum_{i=1}^{n} d_i \tan^{-1}(2^{-i}) + Z_{n+1} \tag{6}$$

其中 $d_i = \mathrm{sgn}(Z_i)$，可以证明 $Z_{n+1} < 2^{-n}$。

3.1.3 有理近似 CORDIC 算法在计算三角函数时的初始值为 $X = 1, Y = 0$，本文利用 Z 迭代 n 次产生 $Z_{N+1}(Z_{N+1} \to 0)$，根据 Maclaurin 公式，作如下有理近似：

$$\tan(Z_{n+1}) = Z_{n+1} \tag{7}$$

可知该近似的最大误差 $(Z = 45^\circ)$ 约为 $R_n \le Z_{n+1}^3/18$，当 $n = 20$ 时，$R_n \le 8.036 \times 10^{-21}$，$Z_{n+1} = 0.000000524929843$。

3.1.4 X, Y 迭代 由于 tan=sin/cos，所以取初值 $Y_1 = Z_{n+1}, X_1 = 1$，进行如下的 X, Y 迭代。

$$\left.\begin{aligned} x_{i+1} &= x_i - d_i y_i 2^{-i} \\ y_{i+1} &= y_i + d_i x_i 2^{-i} \end{aligned}\right\} \tag{8}$$

3.1.5 结果运算 根据有理近似，当迭代次数为 20 次时，误差 $R_n \le 8.036 \times 10^{-21}$。迭代 20 次后，得到 y_{21} 和 x_{21}，从而 $\sin z = K y_{21}, \cos z = K x_{21}, \tan z = y_{21}/x_{21}$。经验证，运算结果的误差 $R_n \le 8.036 \times 10^{-21}$。

3.2 双曲函数的快速实现

双曲函数主要是实现指数函数和对数函数的运算，本文只对对数函数的实现进行说明，由于指数和对数互为反函数，其实现过程可由对数函数的迭代公式逆向推导而得。此外，平方根函数也可快速实现。

对数函数的实现是采用双曲坐标下的 CORDIC 算法，但对迭代次序进行合理改变，舍弃了二次循环运算，使控制变得更加简单，并引入了有理近似，具体步骤如下：

3.2.1 X, Y 迭代 由于对数函数的实现是利用 CORDIC 迭代所产生的 $\tanh^{-1}(y/x)$，并且 $d_i = \mathrm{sgn}(y_i)$，所以就先进行 X, Y 迭代，具体迭代如下：

$$\left.\begin{aligned} x_{i+1} &= x_i - d_i y_i 2^{-i} \\ y_{i+1} &= y_i - d_i x_i 2^{-i} \end{aligned}\right\} \tag{9}$$

其中 $x_0 - y_0 = 2$，迭代所产生的方向因子 $d_i = \mathrm{sgn}(y_i)$ 存放于移位寄存器中。

3.2.2 有理近似 根据 Maclaurin 公式，取有理近似：

$$\tanh y_{N+1} = y_{N+1}/x_{N+1} \tag{10}$$

由于 $\tanh^{-1}2^{-i} < 2^{-i} < \tan^{-1}2^{-i}$，如要达到三角函数的精度，就必须增加迭代次数。当 $N = 25$ 时，就能够达到精度要求。

3.2.3 Z 迭代 由于 tanh=sinh/cosh，所以取有理近似 $Z_1 = y_{N+1}/x_{N+1}$，根据移位寄存器中的方向因子，作 Z 迭代，具体迭代如下：

$$Z_{i+1} = Z_i + d_i \tanh^{-1}2^{-i} \tag{11}$$

3.2.4 结果运算　根据 CORDIC 算法，$\ln(W) = 2Z_{26}$，其中 $x_0 = W + 1, y_0 = W - 1$。运算精度与三角函数的精度一致。其他对数函数可根据下面的公式实现。

$$\log_n x = \log_n e \cdot \ln x \tag{12}$$

平方根函数的实现与对数函数的实现过程相似，但是平方根函数的实现过程与 Z 没有关系，所以只需要 X, Y 迭代和结果运算 (K 运算)。当迭代次数为 25 次，就可满足扩展精度实数要求，具有较高的速度。

4 硬件和工艺实现结果

4.1 硬件系统结构

在本文设计中，所有操作数 (二进制编码的十进制数－BCD 、整数、浮点数) 首先变换为扩展实数类型，计算完成后再转换为目标操作数，所以在用 Verilog HDL 实现算法时，数据路径如图 1 所示，并采用微程序控制方式。

在图 1 的数据路径中，可分为尾数数据路径和指数数据路径。尾数数据路径的宽度为 68 位，除 64 位扩展精度的浮点尾数外，还包括溢出位、警戒位、舍入位和粘接位。指数数据路径的宽度为 16 位。尾数算术逻辑单元 (ALU) 中除加减和逻辑运算外，还包括 Booth 尾数乘法器、SRT 尾数除法器。为了提高运算速度和体系结构设计的方便，常数 ROM 分为两部分组成，尾数常数 ROM 和指数常数 ROM，其中尾数常数 ROM 为 64×68，指数常数 ROM 为 64×16，即共有 64 个常数。

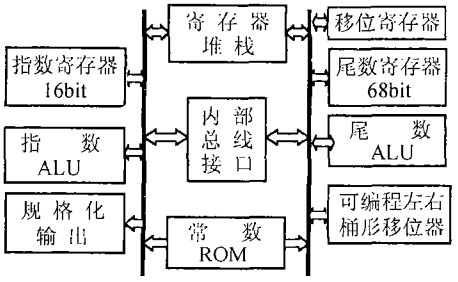


图 1　协处理器数据路径

内部总线接口是尾数和指数之间的调整，但所有的调整都是在计算前和计算后进行的，在运算过程中指数数据路径和尾数数据路径没有相互影响，从而提高运算速度。

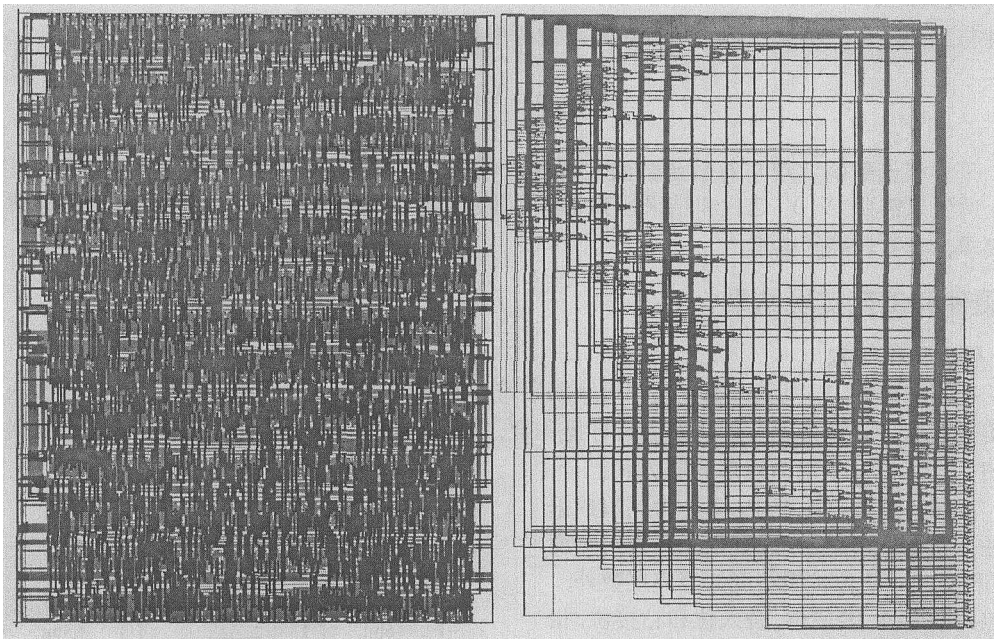
4.2 硬件和工艺实现结果

基于算法改进的浮点协处理器 IP 核采用 Verilog-HDL 实现，采用 Modelsim 进行仿真，采用 Synopsys DC 进行逻辑综合，采用 Cadence SE 进行布局布线，工艺采用 CSMC 0.6μm 的 SPDM 工艺。由于该 IP 核中的总线宽度为 84 位，布线通道的面积将大于电路单元的面积，所以采用 SPDM 工艺布局布线所得的面积偏大。图 2 是 68 位尾数加法器的 Schematic 和 Layout，其中 Layout 的面积为 0.569×0.786mm^2。

采用该 IP 核的 8087 兼容芯片面积为 5.1×6.8mm^2。本文采用仿真测试和 FPGA 测试方法，在临时实数的精度要求下，各函数指令的执行时间如表 1 所示，并与 Intel 公司 80X87 进行比较，速度比 80X87 有显著提高。

表 1　各函数指令执行时间 (时钟周期)

函数指令	实现功能	执行时间周期 (典型值)	
		本文	80X87
FPTAN	$Y/X = \tan(Z)$	260	450
FPATAN	$Z = \tan^{-1}(Y/X)$	310	650
FL2X	$Z = Y \bullet \log_2 X$	400	950
FL2XP1	$Z = Y \bullet \log_2(X + 1)$	405	850
F2XM1	$Z = 2^X - 1$	355	500



(a) 工艺实现结果 (b) 硬件实现结果

图 2 68 位尾数加法器的硬件和工艺实现结果

5 结论

从嵌入式应用出发，引入有理近似，改进迭代次序，提出了改进的 CORDIC 算法，并增加了专门的尾数乘除部件，从而实现初等函数的高速高精度统一运算。基于改进的 CORDIC 算法，采用微程序控制方式，完成了浮点协处理器 IP 核。结果表明，速度比 80X87 有显著的提高，规模较小，控制简单，非常适合嵌入式应用。

参 考 文 献

[1] Volder J E. The CORDIC trigonometric computing technique. *IRE Trans. on Electronics Computers*, 1959, EC-8(3): 330–334.

[2] Walther J S. A unified algorithm for elementary functions. AFIPS 1971, Proc. Spring Joint Computer Conf, Boston, 1971: 379–385.

[3] Rafi Nave. Implementation of transcendental on a numerics processor. *Microprocessing and Microprogramming*, 1983, 17(11): 221–225.

[4] 沈绪榜. 超大规模集成系统设计. 北京：科学出版社， 1992: 326–337.

[5] Wong W F, *et al.*. Fast hardware-based algorithms for elementary functions computations using rectanggular multipliers. *IEEE Trans. on Comput.*, 1994, C-43(3): 381–386.

[6] International Business Machines Co. PowerPC 740, PowerPC 750 RISC Processor User's Manual, Feb, 1999.

[7] Schulte M J, Stine J E, Wires K E. High-speed reciprocal approximaitons. IEEE Proc. Circuit and System, Los Angeles, 1998: 1183–1187.

[8] Schulte M J, Stine J E. Symmetric bipartite tables for accurate function approximation. Proc. 13th Symp. Computer Arithmetic, Chicago, 1997: 175–181.

朱樟明： 男， 1978 年生，博士生，研究方向为 VLSI 及高速混合信号集成电路设计、集成电路低压低功耗设计.

周 端： 女， 1957 年生，副教授，研究方向为计算机体系结构及 VLSI 设计.

杨银堂： 男， 1962 年生，西安电子科技大学微电子所所长，教授，博士生导师，研究方向为 VLSI 技术、深亚微米模拟集成电路设计、新型半导体器件和电路设计.