文章编号:1671-4598(2017)10-0221-03 DOI:10.16526/j.cnki.11—4762/tp.2017.10.056 中图分类号:TP39 文献标识码:A

基于 FPGA 的浮点指数函数算法研究与实现

史雄伟，王 成，张春雷，陈乃奎
（北京广利核系统工程有限公司，北京 100094）

摘要：基于 FPGA 的核电站仪控设备中涉及大量浮点指数运算，而常用的 CORDIC 算法和线性逼近法等存在计算范围小、计算精度不高等问题，对 FPGA 硬件实现指数函数的方法进行研究，并提出一种改进的级数近似法；该方法对输入进行预处理，将输入分解后采用查找表和泰勒级数展开结合的方法，在展开很少项数的情况下快速收敛，发挥查找表法和级数近似法的优势，提高算法的运算精度和效率；在 Matlab 环境下对改进算法的有效性进行仿真验证，且采用 Verilog 语言进行编程实现，在 Microsemi 公司的 IGLOO2 系列 FPGA 上进行具体算法性能验证；Matlab 仿真和 FPGA 验证结果均表明，改进的级数近似法能够大幅增大指数函数的自变量输入范围，并提高计算精度。

关键词：指数函数；FPGA；级数近似法

Algorithm Research and Implementation of Float Point Exponential Function Based on FPGA

Shi Xiongwei, Wang Cheng, Zhang Chunlei, Chen Naikui
(China Techenergy Co., Ltd., Beijing 100094, China)

Abstract: A large number of floating-point exponential calculations are involved in nuclear power plant instrumentation based on FPGA, and the methods for hardware implementation of exponential function are studied, an improved series approximation algorithm is proposed, aiming at solving the problems such as small calculation scale, low precision existing in CORDIC algorithm and linear approximation algorithm. The lookup table and series approximation algorithm are combined in the proposed algorithm, with input data splitting into two part. It takes advantage of the lookup method and the linear approximation, and can work even using a few expansion series. The validity of the improved algorithm is simulated in Matlab, and algorithm is programmed using Verilog and verified on the IGLOO2 series FPGA of Microsemi Corporation. The Matlab simulation result and the implementation result on FPGA demonstrates that this method can expand the calculation capacity and with high accuracy.

Keywords: exponential function; FPGA; series approximation method

0 引言

在核电站仪控系统中，需要采集堆芯外核检测仪表的数据以实现对核反应堆的实时监控。对于一部分核测设备，需要进行指数、对数等超越函数的算法运算才能得到实际物理值。FPGA 凭借其并行定制计算的高效性和可重构的灵活性，在核电站仪控设备中得到越来越广泛的应用。然而 FPGA 本身对指数函数的直接计算缺乏有效地支持，而且通用、高效的超越函数计算 IP 核也较少[1]，因此有必要研究运算高效、精度高、计算范围大的浮点指数函数硬件实现方法。

在对 CORDIC 算法、分段线性逼近法等常用算法进行性能分析的基础上，针对其计算范围小、精度差等问题，提出一种改进的级数近似法，与查表法结合实现浮点指数函数，能有效提高计算精度和计算效率。通过 Matlab 仿真分析和 Microsemi 的 IGLOO2 系列的 FPGA 器件的硬件实现，对算法进行性能测试评估。实验结果表明，相比于 CORDIC 算法、分段线性逼近法，本方法在计算精度和收敛域范围方面的性能更好。而且本方法能适应于多种常见超越函数的计算，适用性较好。

1 现有算法分析

目前在 FPGA 中硬件实现指数函数常用的方法有 CORDIC 算法、分段线性逼近法、查表法、级数近似法等[2-6]。

1.1 CORDIC 算法实现

CORDIC 即坐标旋转数字计算方法是一种数值逼近算法，最初应用于解决三角学问题，通过多次迭代将一些在硬件实现中既耗时又占用资源的运算转换为简单的移位、加减运算，便于硬件实现，后来被广泛用于多种初等函数的运算中（包括三角函数、乘除法运算、指数运算、对数运算等）[3-6]。

最初的 CORDIC 算法基本原理为通过一系列固定的、与运算基数有关的角度的不断偏摆以逼近所需的旋转角度。1971 年，J. S. Walther 提出了统一的 CORDIC 算法，引入工作模式参数 m：$m=1$ 为圆周系统、$m=0$ 为线性系统、$m=-1$ 为双曲系统），采用一个 CORDIC 迭代方程统一表示 3 种系统：

$$x_{i+1} = x_i - m\delta_i y_i \cdot 2^{-i}$$

$$y_{i+1} = y_i + \delta_i x_i \cdot 2^{-i}, i = 0,1,2,\cdots,N-1$$

$$z_{i+1} = z_i - \delta_i \theta_i$$

其中：$\theta = \begin{cases} \arctan 2^{-i}, & m=1 \\ 2^{-i}, & m=0 \\ \arctan h 2^{-i}, & m=-1 \end{cases}$，

收稿日期：2017-04-12； **修回日期**：2017-05-03。
基金项目：国家科技重大专项(2011ZX06004-030)。
作者简介：史雄伟(1985-)，男，河北石家庄人，硕士，工程师，主要从事复杂数字信号处理和实现方向的研究。

$$\delta_i = \begin{cases} sign(z_i), \text{旋转模式} \\ -sign(x_i \cdot y_i), \text{向量模式} \end{cases}$$

选择不同参数可得到不同的计算结果，如表 1 所示。

表 1 CORDIC 算法的参数选择及结果

CORDIC	旋转模式($z_N -> 0$)	向量模式($y_N -> 0$)
圆周系统 ($m=1$)	$x_N = \frac{1}{K_c}(x_0 \cos z_0 - y_0 \sin z_0)$ $y_N = \frac{1}{K_c}(y_0 \cos z_0 + x_0 \sin z_0)$	$x_N = \frac{1}{K_c}\sqrt{x_0{}^2 + y_0{}^2}$ $z_N = z_0 + \arctan(y_0/x_0)$
线性系统 ($m=0$)	$x_N = x_0$ $y_N = y_0 + x_0 z_0$	$x_N = x_0$ $z_N = z_0 + y_0/x_0$
双曲系统 ($m=-1$)	$x_N = \frac{1}{K_h}(x_0 \cosh z_0 - y_0 \sinh z_0)$ $y_N = \frac{1}{K_h}(y_0 \cosh z_0 + x_0 \sinh z_0)$	$x_N = \frac{1}{K_h}\sqrt{x_0{}^2 - y_0{}^2}$ $z_N = z_0 + arctanh(y_0/x_0)$

其中：$K_c = \prod_{i=0}^{N-1} \frac{1}{\sqrt{1+2^{-2i}}}, K_h = \prod_{i=0}^{N-1} \frac{1}{\sqrt{1-2^{-2i}}}$

下面介绍利用 CORDIC 算法的双曲旋转法实现自然指数函数运算。

由于指数函数 $e^x = \sinh x + \cosh x$ ，采用双曲系统的旋转模式可以计算 $\sinh x$ 和 $\cosh x$，初始值设置为：$x_0 = K_h$，$y_0 = 0$，$z_0 = x$。

需要注意的是在双曲系统下要保证迭代序列收敛，需要从第 4 项开始每隔 $3n+1$ 项必须重复一次，即 $n = 1,2,3,4,4,5,6,\cdots,13,13,14,\cdots,40,40,\cdots$，重复迭代的位置为：$i = 4, 13, 40, \cdots, k, 3k+1, \ldots$

根据文献 [6]，自变量的收敛范围仅为 (−1.1182, 1.1182)，函数的输入范围受到极大限制。

$$|z_{in}| \leqslant \arctanh(2^{-N}) + \sum_{i=1}^{N} \arctanh(2^{-i})$$

即 1.1182，解决的办法是增加 i 为负数的迭代，改进的迭代公式为：

当 $i > 0$ 时，
$$\begin{aligned} x_{i+1} &= x_i + \delta_i y_i \cdot 2^{-i} \\ y_{i+1} &= y_i + \delta_i x_i \cdot 2^{-i} \\ z_{i+1} &= z_i - \delta_i \arctanh(2^{-i}) \end{aligned}$$

当 $i <= 0$ 时，
$$\begin{aligned} x_{i+1} &= x_i + \delta_i y_i \cdot (1 - 2^{i-2}) \\ y_{i+1} &= y_i + \delta_i x_i \cdot (1 - 2^{i-2}) \\ z_{i+1} &= z_i - \delta_i \arctanh(1 - 2^{i-2}) \end{aligned}$$

$$K_h = \prod_{i=-M}^{0} \frac{1}{\sqrt{1-(1-2^{-2^{-i+1}})^2}} \cdot \prod_{i=1}^{N-1} \frac{1}{\sqrt{1-2^{-2i}}}$$

硬件实现时，首先计算出 K_h 的值，然后给定 $x_0 = K_h$，$y_0 = 0$，$z_0 = x$，进行多次迭代后，z 趋近于 0，指数函数的计算结果为 $x+y$。

1.2 分段线性逼近法

分段线性逼近法是将查找表与多项式逼近相结合，保存各段多项式的系数，计算速度快，消耗资源少，易于硬件实现[7]，广泛应用于人工神经网络、图像处理等非线性函数的计算过程[8]。

分段线性逼近法采用多段直线来逼近曲线，创建查找表保存各段直线的系数，计算时根据输入选择合适的直线段逼近非线性函数。每个直线段表示为 $y = k_i x + b_i$，根据每一个分段区间逼近基点处直线与非线性函数的值相同，可以求得各段直线段的 k 值和 b 值。

基于分段线性逼近法在 FPGA 中实现指数函数的主要步骤为：

步骤 1：按照一定的分段规则将指数函数的输入区间进行划分，求取各个直线段的逼近基点。本文根据需求定义输入区间为 [−10，10]，采用均分输入区间的方法，以 s 为步进针对 [−10，10] 区间划分为 $20/s$ 个区间。

步骤 2：根据相邻 2 个逼近基点的指数函数值，计算各个直线段 $y = k_i x + b_i$ 中的 k、b 值。为方便硬件实现直接使用，需要将 k、b 值转换成相应的数据格式。一般采用最小二乘法拟合得到每段的 k、b 值，并将其存储在 FPGA 的片内 RAM 中。

步骤 3：使用硬件描述语言实现硬件设计。首先要计算的指数函数自变量 x，选择正确的 k、b 值，然后再进行一次乘法和加法操作即得到计算结果，其计算速度较快。

1.3 查表法原理

查表法是根据计算精度要求，先将指数函数的输入值分为若干个点，分别求出对应点的函数值，并将结果写入 ROM/RAM 等存储器中。然后通过适当的运算将输入变量与存储器地址对应起来，再计算时直接查表得到结果。

用查表法实现函数计算，虽然没有输出延时，但其所需存储单元随着函数输入域的增大或是计算精度的提高呈指数形式增加，资源消耗巨大[2]，甚至是不可行的。

1.4 级数近似法原理

级数近似法利用泰勒展开式将超越函数的复杂运算转换为基本的乘法和加减法运算，再根据计算精度对级数进行截取。级数近似法需要的乘法器、加法器较多，硬件实现复杂，资源消耗较大。

根据泰勒公式可以得到指数函数 e^x 的幂级数展开式为：

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \frac{x^n}{n!}, \quad -\infty < x < \infty$$

该级数在整个定义域内收敛，但是只有在 0 点附近收敛速度较快，当 x 取值较大时，需要的展开项较多，运算量大，计算速度变慢，不适合在 FPGA 中直接使用。

2 改进算法设计

针对现有算法存在精度不高、计算范围小、消耗资源多等问题，提出一种改进的级数近似法，采用查找表与级数近似法相结合实现浮点函数计算，有效提高计算精度和计算效率。

2.1 改进级数近似法的原理

针对泰勒级数近似法的缺点进行改进，将输入数据 x 进行预处理，将其转换到 0 点附近的区域，采用较少的展开项满足误差需求。原理为：

$x = a + c$，其中 a 为整数部分，c 为小数部分。但是在 c 取值接近 1 时，需要的迭代步骤仍然较多，进一步，令 $x = a + b_1 * 2^{(-1)} + b_2 * 2^{(-2)} + \cdots + b_n * 2^{(-n)} + c$ 其中，a 为整数部分，$b_1 \sim b_n$ 分别 $2^{-1} \sim 2^{-n}$ 的二进制系数，c 为接近与 0 的余项。$c < 2^{(-n)}$，能够保证只采用幂级数展开的前几项即可满足精度要求。最终结果 $e^x = e^a \cdot e^b \cdot e^c$。其中，$b = b_1 * 2^{(-1)} + b_2 * 2^{(-2)} + \cdots + b_n * 2^{(-n)}$。

在 n 确定后，根据 IEEE754 标准浮点数的表示方法，可以方便求得 a，$b_1 \sim b_n$，和 c. 以 n 取 3 为例，$x = 1.3125$ 时，浮点表示为 0x3FA80000，即：

符号位（1bit）	指数位（8bit）	尾数位（23bit）
0	01111111	01010000000000000000000

则 a=1，$b1$=0，$b2$=1，$b3$=0，c=0.00010000000000000000000 (0.0625)

硬件实现时，根据所需精度确定 n 的取值和泰勒级数展开的项数，并将 e^a 的值事先计算出来做成查找表，a=0，1，2，…，88（单精度浮点数能表示的数值范围决定 a 最大值为 88），同理将 e^b 也分别计算出来存入查找表中备用。

基于该算法在 FPGA 中实现指数函数的计算，主要步骤为：

步骤 1：将输入值 x 进行预处理，拆分成整数部分 a，并通过移位等操作得到 b_1，b_2，b_3，…b_n 和小数余项 c。

步骤 2：根据 a 的结果查找 e^a 的值，根据 b_1，b_2，b_3，… b_n 查找 e^b 的值。

步骤 3：根据泰勒级数展开式计算 e^c 的值。

步骤 4：计算最终结果 $e^x = e^a \cdot e^b \cdot e^c$。

本方法采用级数近似和查找表结合的方法，能够在整个定义域内都得到满意的精度。

2.2　性能分析

经过理论分析可得，展开式项数越多，计算精度越高，但是消耗的资源会越大，为了适合硬件实现，以展开至 x^4 项进行分析。

其次确定 n 的数值，$b_1 \sim b_n$ 不存在，即 n=0 的情况，即只将数据分解为整数部分和小数部分，最大相对误差是 0.01049。n = 1，2，3 的情况下最大相对误差分别为：1.19277e−004，3.15153e−006，9.05842e−008；仿真结果与理论分析一致，n 的取值越大，最后得到的小数余项 c 越小，展开项数一致的情况下精度越高。综上所述为了平衡资源和精度，FPGA 硬件实现时取 n=3，展开至 x^4 项进行分析。

下面对提出的改进算法与 CORDIC 算法、分段线性逼近算法在 matlab 环境下进行精度性能分析比较。参考文献 [6]，CORDIC 算法负向迭代次数为 −5 时收敛域范围扩展到（−12.42644，12.42644），从 −8 次迭代，收敛域扩展到（−22.82194，22.82194）。由于硬件资源等限制，采用正向迭代 22 次进行分析[4]。对于分段线性逼近法，取 x 的范围与 CORDIC 算法的收敛域一致，步长为 0.01，各拟合直线段的 k 值和 b 值均采用最小二乘法拟合得出。图 1 和图 2 分别是 3 种算法在输入范围为（−12，12）和（−22，22）时 3 种算法的相对精度曲线图。

对于 CORDIC 算法，负向迭代次数为 −5 时，最大相对误差为 7.43625e−006，负向迭代次数为 −7 时，最大相对误差增大为 6.71834e−004。由于单精度浮点数能够表示的精度有限，当不能足够精确地表示 K_h 时，计算结果会有较大的误差。收敛域越大，i 为负数的迭代次数越多，计算的 K_h 的误差会越大，需要在收敛域和精度之间寻找平衡。

分段线性逼近法的在输入范围为（−12，12）和（−22，22）时，根据步长得到划分的段数分别为 2 400，4 400，最大相对误差分别为 −6.02285e−006，−7.46858e−006。

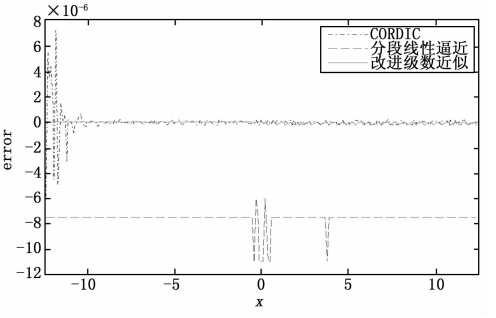

图 1　相对误差曲线图

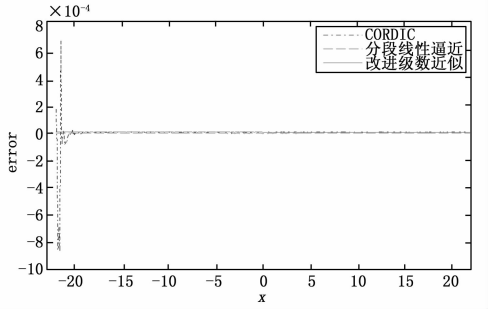

图 2　误差曲线图

改进级数近似法在整个输入范围内使用的参数一致，最大相对误差为 9.05842e−008，比 CORDIC 算法和分段线性逼近法精度都高，而且输入范围增大。经过仿真分析，改进级数近似法在输入范围（−86，88.7）内都能够保证运算的高精度，并且所消耗的资源不随输入范围的变化而改变。

CORDIC 算法易于硬件实现，但是收敛范围有限，需要增加 n 为负数的迭代以扩展收敛域，且收敛速度与数据的表示精度成反比，当精度要求较高时，算法的迭代次数较多，计算延迟会增大。分段线性逼近法原理简单，但是计算精度与分段数量相关，当自变量范围较大、计算精度较高时，要有分段数量会很大。改进级数近似法在不增加资源消耗的前提下，在很大输入范围内都能够保持高运算精度。

3　算法的 FPGA 实现

为对改进级数近似法的计算精度、计算速度、资源消耗及适用范围进行评估，将上述设计在 Microsemi 公司的 IGLOO2 系列的 FPGA M2GL150 上实现，该芯片拥有多达 240 个 mathblock 资源和丰富的 LUT 和逻辑资源。

在硬件实现时，CORDIC 算法从 −5 开始负向迭代，正向迭代次数设置为 22，能够保证计算相对精度在 10^{-6} 级别。分段线性逼近法计算范围与 CORDIC 的收敛域保持一致，步长为 0.01。改进级数近似法 n 取 3，即保证小数项 c 小于 0.125。展开项为 5 项，即计算到 x^4 项。为保证运算速度，所有算法均采用流水线结构。

经过理论分析和仿真结果以及 FPGA 的硬件实现，可以得到 3 种算法的性能对比如表 2 所示。

CORDIC 算法可以使用简单的移位和累加代替乘法，实现简单，但是要保证精度需要迭代的次数较多，而且计算范围有限。

（下转第 231 页）

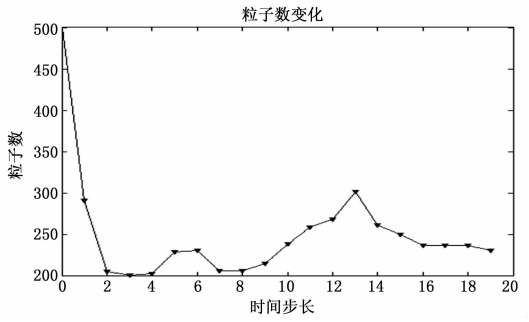

图 5　粒子数变化情况

中的复杂度降低，复杂度变小。

4　结论

提出了一种自适应遗传粒子滤波算法，并将其应用于滚动轴承的性能衰退趋势预测。AGPF 算法采用遗传算法代替传统粒子滤波中的重采样方法，并根据粒子数与滤波误差方差之间的关系，自适应调节滤波过程中的粒子数，解决了粒子退化及计算过程复杂度问题。将 AGPF 算法应用于滚动轴承性能衰退趋势预测，分别对 PF、GPF 和 AGPF 算法的滤波效果进行了验证，结果表明，AGPF 算法在确保预测精度的前提下降低了计算的复杂度，更适合于轴承的性能衰退趋势预测。

参考文献：

[1] 潘宏侠，门吉芳. 粒子滤波在轴承故障振动信号降噪中的应用 [J]. 振动、测试与诊断，2011，31 (3)：354－356.

[2] 胡建旺，张　淼. 自适应采样数粒子滤波算法 [J]. 军械工程学院学报，2009，21 (3)：55－58.

[3] 阙子俊. 基于 UKF 的轴承剩余寿命预测方法研究 [J]. 仪器仪表学报，2016，37 (9)：2036－2043.

[4] Li N P，Lei Y G. An Improved exponential model for predicting remaining useful life of rolling element bearings [J]. IEEE Transactions on Industrial Electronics，2015，62 (12)：7762－7773.

[5] Qian Y N，Yan R Q. Remaining useful life prediction of rolling bearings using an enhanced particle filter [J]. IEEE Transactions on Instrumentation and Measurement，2015，64 (10)：2696－2707.

[6] 吴　昊，孙晓燕，郭玉堂. 保持粒子多样性的非退化粒子滤波方法研究 [J]. 电子学报，2016，44 (7)：1734－1741.

[7] Li N P，Lei Y G，Liu Z Y，et al. A particle filtering－based approach for remaining useful life predication of rolling element bearings [A]. 2014 IEEE Conference on Prognostics and Health Management (PHM) [C]. IEEE，2014.

(上接第 223 页)

表 2　3 种算法的性能对比

算法	运算周期数	资源	计算范围	最大相对误差
CORDIC	36clk	DFF:1970 RAM1K18:0 MACC:0	[－12,12]	$7.58*10^{-6}$
分段线性逼近	23clk	DFF: 853 RAM1K18:8 MACC:2	[－12.,12]	$-6.54*10^{-5}$
改进级数近似法	78clk	DFF:2306 RAM1K18:1 MACC:24	[－86.2,88.7]	$1.658*10^{-7}$

分段线性逼近法实现简单，仅使用查找表和一次乘法和加法即可计算结果，查找表的大小与计算范围和步长相关，计算范围越大、步长越小则需要的查找表越大。在步长较大精度差，步长小时占用的资源多，而且查找表耗费的时间也会增长。

改进级数近似法需要使用乘法器，硬件实现相对复杂，消耗了较多的硬件 mathblock 资源和逻辑资源，并且计算速度也相对较慢。但是通过改进的级数近似法能够大幅加大计算范围，而且计算精度在输入范围内均保证高精度，比 CORDIC 算法高一个数量级，优势明显。

4　结论

本文对当前硬件实现指数函数的方法进行了研究，并以增大计算范围、提高计算精度为目的，提出一种改进的级数近似法。该方法采用查找表和级数近似法相结合的方式，将输入数据进行预处理，在展开很少项数的情况下快速收敛，既保证了计算精度，又大幅提高了指数函数的计算范围。在 Matlab 环境下对 CORDIC 算法、分段线性逼近法和改进级数近似法进行了仿真分析，并采用 Verilog 语言，在 Microsemi 公司的 IGLOO2 系列的 FPGA M2GL150 上验证了各算法的性能，实验结果表明，改进的级数近似法虽然消耗较多的资源和计算时间，但是在计算范围和计算精度方面都有明显的优势，能够更大范围地满足工程应用需求。文中提出的改进方法已经在某核电站仪控平台实现过程中得到应用，该算法也适用于其他基于 FGPA 的系统实现中。

参考文献：

[1] 王少军，张启荣，彭　宇，等. 超越函数 FPGA 计算的最佳等距分段线性逼近方法 [J]. 仪器仪表学报，2014，35 (6)：1209－1216.

[2] 赵海燕，周晓方，周　电. 对数/指数算法的改进及其 VLSI 实现 [J]. 计算机工程与应用，2007，43 (7)：104－107.

[3] 牟胜梅，李兆刚. 一种面向 FPGA 的指/对数函数求值方法 [J]. 计算机工程与应用，2011，47 (33)：59－61.

[4] 黄晓可，刘洛琨，汪　涛，等. 基于改进 SF－CORDIC 的指数和对数函数求值算法 [J]. 计算机应用与软件，2014，31 (2)：279－282.

[5] 刘小会，许　蕾，刘海颖，等. 基于 CORDIC 改进算法的反正切函数在 FPGA 中的实现 [J]. 计算机技术与发展，2013，23 (11)：103－107.

[6] Hu X B，Harber R G，Bass S C. Expanding the range of convergence of the CORDIC algorithm [J]. IEEE Transactions on Computers，1991，40 (1)：13－21.

[7] 杨　阳，王永纲，都军伟. Piecewise 算法计算超越函数及其 FPGA 实现 [J]. 核电子学与探测技术，2010，30 (6)：755－758.

[8] Florea C，Vertan C. Piecewise linear approximation of logarithmic image processing models for dynamic range enhancement [J]. IEEE Transactions on Power Systems，2011，26 (4)：2581－2583.

[9] 郭邵忠，许瑾晨，陈建勋. 一种改进的超越函数的通用算法 [J]. 计算机工程. 2012，38 (15)：31－34.

[10] 张玲玲，李克俭，蔡启仲. 基于 FPGA 自主控制浮点加减乘除控制器设计 [J]. 计算机测量与控制，2014，22 (10)：2941－2943.