Lower-Error Antilogarithmic Converters using Binary Error Searching Schemes

Chao-Tsung Kuo, Tso-Bing Juang

Abstract*— In this paper, lower-error and ROM-free antilogarithmic converters with multiple regions of piecewise-linear approximation are proposed. By employing Binary Error Searching schemes, the error percent ranges of our proposed antilogarithmic converters could achieve 1.6808%, 0.5681%, 0.137% and 0.098% for 2-region, 4-region, 8-region and 16-region approximations respectively, which can outperform previously proposed methods in the literature. Area comparisons with previously well-known antilogarithmic converter using six-region approximation methods in the literature, our proposed antilogarithmic converter with four-region approximation can provide 1.7x error reduction with only 30% extra hardware overhead under the same delay constraints. These antilogarithmic converters are all designed and synthesized using TSMC 0.18 μm process. Our proposed converters can be applied in the real-time 3-D graphics and DSP computations to ease the tremendous computation efforts.

***Index Terms*— Antilogarithm, Logarithm, Computer arithmetic, very large scale integration (VLSI) design.**

I. INTRODUCTION

Real-time three-dimension (3-D) graphics and Digital Signal Processing (DSP) technologies are the attractive applications for smart Information Technology (IT) consumer electronics products in the amazing smart communication era. Many complex arithmetic calculations such as multiplication, division, reciprocal, square-root and power operations are required in DSP technology. Nowadays, logarithmic number system (LNS) can be used to simplify these complex operations using simple shift-and-add operations. LNS-based computing system contains logarithmic conversion unit, simple calculation unit and antilogarithmic conversion unit to the binary values. Many methods [1-17] about logarithmic conversion to binary system and antilogarithmic converter have been presented in recent years. Under all the tradeoffs considering area, delay time and high accuracy, shift-and-add methods can achieve higher performance instead of complicated multiplication operation using simple shifters and adders/subtractors. Thus, the corresponding antilogarithmic conversions can be easily performed using simple operations based on piecewise-linear approximations of the input data. Since many of previous works are related to logarithmic conversions, it will be useful if the lower error antilogarithmic converters are also proposed to make the overall LNS-based computations more efficient.

Therefore, we will focus on the antilogarithmic conversion in this paper. Mitchell [3] presented a logarithmic and antilogarithmic converter which is based on one-region linear approximation scheme. Mitchell's method is very simple and easy to implement, but it can not provide sufficient accuracy. In order to improve the accuracy of Mitchell's approximation, Hall [4] proposed antilogarithmic converters using two-region and four-region piecewise-linear approximation schemes to improve the linear errors of antilogarithmic converters. However, Hall's approaches could lead to complex hardware implementation due to full bit-accuracy of the input which is used for the approximation. Abed in [5] also proposed two-region, six-region and seven-region linear approximations to reduce the approximation error and improve the area-efficiency, and Kim [6] proposed eight-region piecewise-linear approximation to further reduce more approximation errors. Although Kim's method can reduce effectively the approximation errors, it requires complex hardware implementation due to many regions of approximations. To achieve lower errors of antilogarithmic conversions with tolerable area overheads, in this paper, we will propose lower-error antilogarithmic converters using 2-, 4-, 8-, 16-region shift-and-add approximations. Our proposed technique first partitions the exact antilogarithmic curve into 2-, 4-, 8-, 16- symmetric piecewise-linear regions and then to perform the corresponding conversions. The operations of conversions are obtained by our proposed Binary Error Search Schemes, which cold achieve lower error ranges compared with previous proposed methods in the literature. In addition, our proposed converters have the lowest error compared to previous methods with the same regions of approximations.

The rest of this paper is organized as follows. In Section II, we will review previously reported antilogarithmic conversion methods that are based on 1-, 2-, 4-, 6-, 7- and 8-region approximation schemes. Then our proposed antilogarithmic converters using Binary Error Searching scheme are described in Section III. In section IV, we will give error analysis and comparisons of our methods and others, VLSI hardware implementations are also given. Finally, Section V draws the conclusions.

II. PREVIOUS WORK OF ANTILOGARITHMIC CONVERTERS

In this section, we will review previous proposed antilogarithmic converter, all are based on shift-and-add operations [3-8]. Generally speaking, to perform antilogarithmic conversion, it is based on 2 power of X, which contains any integer portion k and its mantissa m. In Eq. (1), suppose $X=k+m$ is the input for antilogarithmic conversion, which is obtained by logarithmic converter and simple calculation units of LNS. Therefore, by taking 2 power of X, the antilogarithmic value of X is denoted as Y.

Manuscript Received December, 2013.

Chao-Tsung Kuo is with the Department of Electronic Engineering, National Quemoy University, Kinmen 892, Taiwan.

Tso-Bing Juang is with the Department of Computer Science and Information Engineering, National Pingtung Institute of Commerce, Pingtung 900, Taiwan.

The corresponding conversion is shown in Eq. (2), where k is an integer part and m is a fraction part, respectively. Mitchell [3] proposed the antilogarithmic conversion based on one-region approximation, he approximates 2^m in the m intervals as a straight line (1+m) and its equation is shown in Eq. (3), denoted as Y_{mitchell}'. The error of Mitchell's method denoted as $Error_{\text{Mitchell}}$ is given in Eq. (4). For example, letting X=3.5=3+0.5, here we take k=3 and m=0.5. Y is equal to $2^{3.5}=2^3\times2^{0.5}=8\sqrt{2}\approx 8\times1.4142=11.3136$. In Eq. (3), Mitchell approximates Y as $2^3\times(1+0.5)$, it is equal to 12. Thus $Error_{\text{Mitchell}}$ =12-11.3136=0.6864. Mitchell's method compared with actual values is shown in Fig. 1.

$$X = k + m, \quad k \text{ is any integer and } 0 \le m < 1 \tag{1}$$

$$Y = Anti\log_2(X) = 2^X = 2^k 2^m \tag{2}$$

$$Y_{mitchell}' = 2^{X'} = 2^k(1+m), \quad 0 \le m < 1 \tag{3}$$

$$Error_{Mitchell} = 2^k \times ((1+m) - 2^m), \quad 0 \le m < 1 \tag{4}$$

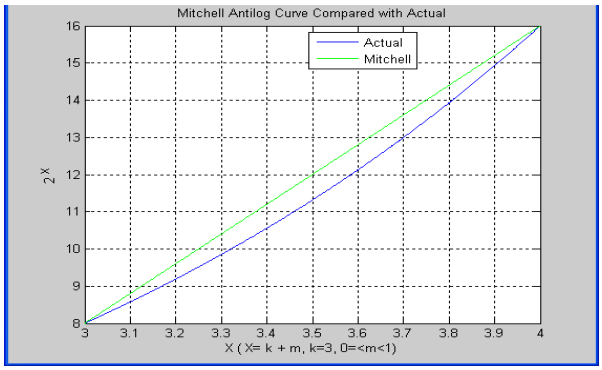

Fig. 1 Mitchell's approximations compared with the actual values

In the following, we will compare every error percent ranges for each method. The bit-width of k and m is 6 bits and 26 bits, respectively, thus Q6.26 format is adopted throughout the work and the comparisons, as shown in Fig. 2. All error percent ranges are obtained by Matlab, k is fixed to 3 and the fractional part m=0 to 1, i.e., X=3 to 4. The error percent range is the summation of the maximum positive and negative error percents. The maximum positive (negative) error percent is defined as the maximum absolute values of the positive (negative) errors which are divided by the actual corresponding antilogarithmic values × 100%. According to (4), we can obtain that the maximum positive error percentage of Mitchell's method = $(((1+m)-2^m)/2^m) \times 100\%$, and the maximum negative error percent is zero, thus the error percent range of Mitchell's method is equal to $(((1+m)-2^m)/2^m) \times 100\%$. The error percentage of Mitchell's one-region antilogarithmic converter is shown in Fig. 3, where the error percent range is 6.1476%.

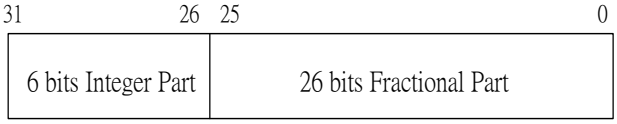

Fig. 2 Q6.26 format adopted in our work and comparisons

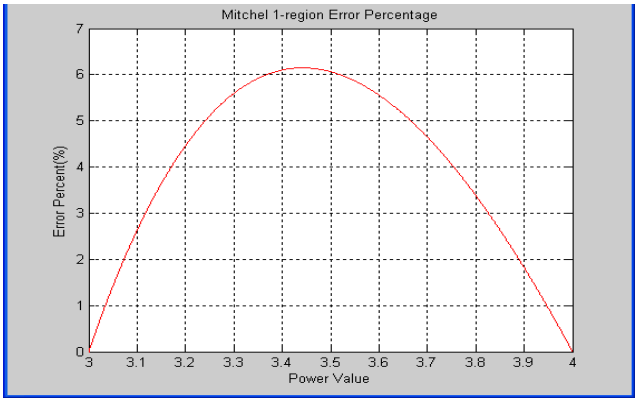

Fig. 3 Error percents of Mitchell's method

Hall [4] proposed two-region and four–region schemes for the antilogarithmic conversions, using multiple piecewise-linear approximations with full bit-width (i.e., 26 bits) of the input to improve the approximation error of antilogarithmic converter, demanding more area cost and complex hardware. The mathematical modeling of Hall's proposed four-region antilogarithmic converter is shown in Eq. (5).

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k[m + (\frac{1}{4})*(1-m) + (\frac{3}{4})], 0 \le m < 0.25; \\ 2^k[m + (\frac{13}{128})*(1-m) + (\frac{55}{64})], 0.25 \le m < 0.5; \\ 2^k[m + (\frac{9}{128})m + (\frac{7}{8})], \quad 0.5 \le m < 0.75; \\ 2^k[m + (\frac{35}{128})m + (\frac{23}{32})], \quad 0.75 \le m < 1.00; \end{cases} \tag{5}$$

The error percent range of Hall's four-region conversion method is 0.7768%, which is shown in Fig. 4.

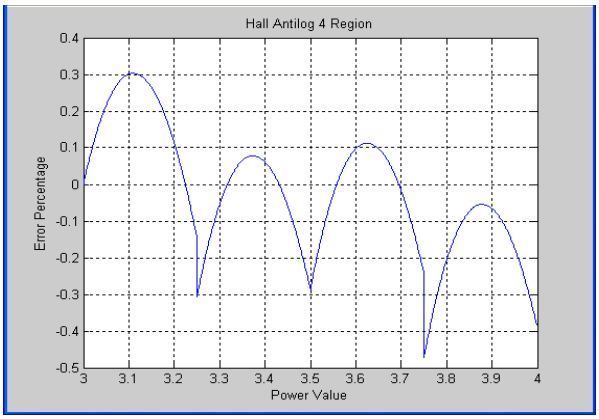

Fig. 4 Error percents of Hall's four-region method

Abed [5] proposed two-region, six-region and seven-region of piecewise-linear approximations of antilogarithmic converter to reduce the linear approximation error. It should be noted that $m_{7MSBits}$ ($m_{9\ MSBits}$) is denoted as the 7 (9) most significant bits of the fraction m. The mathematical modeling of Abed's proposed six-region and seven-region approximations of antilogarithmic converter is shown in Eq. (6) and (7) respectively.

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k[m+(\frac{3}{16})\overline{m}_{7MSBits}+(\frac{103}{128})+(2^{-7})+(2^{-10}+2^{-11})], 0 \le m < 0.0625; \\ 2^k[m+(\frac{3}{16})\overline{m}_{7MSBits}+(\frac{103}{128})], 0.0625 \le m < 0.375; \\ 2^k[m+(\frac{3}{16})\overline{m}_{7MSBits}+(\frac{103}{128})+(2^{-7})], 0.375 \le m < 0.5; \\ 2^k[m+(\frac{3}{16})m_{7MSBits}+(\frac{103}{128})+(2^{-7})], \quad 0.5 \le m < 0.625; \\ 2^k[m+(\frac{3}{16})m_{7MSBits}+(\frac{103}{128})], \quad 0.625 \le m < 0.875; \\ 2^k[m+(\frac{3}{16})m_{7MSBits}+(\frac{103}{128})+(2^{-7})], \quad 0.875 \le m < 1; \end{cases} \quad (6)$$

where $\overline{m}_{7MSBits}$ =1-$m_{7MSBits}$-(1/128).

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k[m+(\frac{1}{4})\overline{m}_{9MSBits}+(\frac{95}{128})+(2^{-8})+(2^{-8}+2^{-11})], 0 \le m < 0.125; \\ 2^k[m+(\frac{1}{4})\overline{m}_{9MSBits}+(\frac{95}{128})+(2^{-8})+(2^{-8})], 0.125 \le m < 0.25; \\ 2^k[m+(\frac{1}{8})\overline{m}_{9MSBits}+(\frac{107}{128})+(2^{-7})+(2^{-9})], 0.25 \le m < 0.375; \\ 2^k[m+(\frac{107}{128})+(2^{-4})+(2^{-6})+(2^{-8})+(2^{-10})], \quad 0.375 \le m < 0.5; \\ 2^k[m+(\frac{107}{128})+(2^{-4})+(2^{-6})], \quad 0.5 \le m < 0.625; \\ 2^k[m+(\frac{1}{8})m_{9MSBits}+(\frac{107}{128})], \quad 0.625 \le m < 0.75; \\ 2^k[m+(\frac{1}{4})m_{9MSBits}+(\frac{95}{128})+(2^{-8})], \quad 0.75 \le m < 1; \end{cases} \quad (7)$$

where $\overline{m}_{9MSBits}$ =1-$m_{9MSBits}$-(1/512).

Simulation result of error percents of Abed's six-region antilogarithmic converter is shown in Fig. 5. The maximum positive error of Abed's proposed six-region antilogarithmic converter is 0.9572% and the maximum negative error is -0.5786%, respectively, that is, the error percentage is 1.5358%.

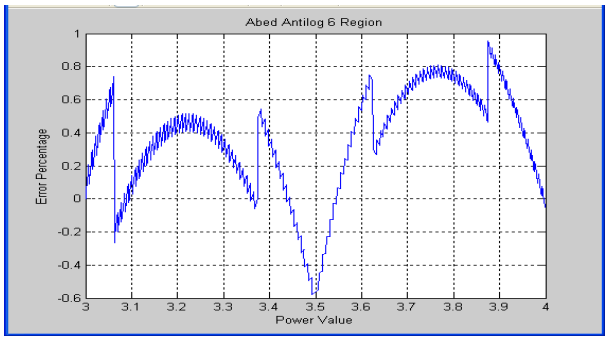

Fig. 5 Error percent of Abed's six-region method

Kim [6] proposed eight-region conversion schemes with full bit-width (i.e., 26 bits) of the input to further improve the approximation errors. Although Kim's method can achieve lower errors than other methods proposed in [3-5], but it will lead to complex hardware implementation due to use of full bit-width of the inputs for approximations.

From above, we can observe that Mitchell's method is simple but suffers from higher errors. Hall's approaches can improve approximation errors but will lead to complex hardware implementation. Abed's methods produces larger errors compared with Kim's [6]. However, Kim's approaches also require a complex hardware implementation. Under the tradeoffs of lowering approximation errors and area efficiency, we expect to propose lower-error antilogarithmic converters with multiple regions based on shift-and-add operations. In Section III, we will describe our proposed lower-error antilogarithmic converters using Binary Error Searching schemes.

III. PROPOSED LOWER-ERROR ANTILOGARITHMIC CONVERTERS USING BINARY ERROR SEARCHING SCHEMES

Taking *X* segments of the straight line *am*+*b* where *X* is the number of the regions. In the straight line, *a* is the slope of the line, *b* is the constant coefficient, and *m* is divided uniformly into 2, 4, 8 and 16 regions, respectively, from 0 to 1. Taking 4-region example, *m* is divided into [0, 0.25), [0.25, 0.5), [0.5, 0.75) and [0.75, 1.00), respectively. The 4-region proposed antilogarithmic conversion compared with Mitchell's method is shown in Fig. 6. It can be shown that our proposed 4-region antilogarithmic conversion will be more close to the actual conversion than Mitchell's methods.

The proposed piecewise-linear approximation is approximated by the following Eq. (8).

$$Y' = 2^k 2^{m'} = 2^k (am+b) \quad (8)$$

$$Error\ Percent = (\frac{(a\,m+b\,)-2^m}{2^m})*100\%, \quad 0 \le m < 1 \quad (9)$$

The error percent is represented by Eq. (9). Considering the feasible implementation on digital circuits and the searching of optimal coefficients of *a* and *b*, our proposed methods for searching coefficient of linear equation $a_i\ m + b_j$ are based on Eq. (10), i.e.

$$a_i = 2^{-m}, b_j = 2^{-n}, \quad 0 \le m \le 8, 0 \le n \le 11, \ m,n\ is\ integer \quad (10)$$

The algorithm of obtaining optimal coefficients min*a* and min*b* in the proposed four-region piecewise-linear antilogarithmic converter is shown in the Fig. 7. In the algorithm, firstly we set the initial value and range of a_i, b_j and *m*. In the mean time, we set the initial optimal coefficients (min*a*, min*b)* to (a_0, b_0), where a_0 and b_0 is the initial value of coefficient range of *a* and *b*, respectively. We take a_i to range from 0 to 2 by step 1/256, b_j also ranging from 0 to 1 by step 1/2048. The next step is to search the minimum coefficients of *a* and *b* using minimizing error percent calculating (i.e., error percent () shown in Eq. (9)). Finally, the optimal coefficients (*mina*, *minb*) can be obtained. The coefficients of the proposed algorithm can be obtained by Matlab software.

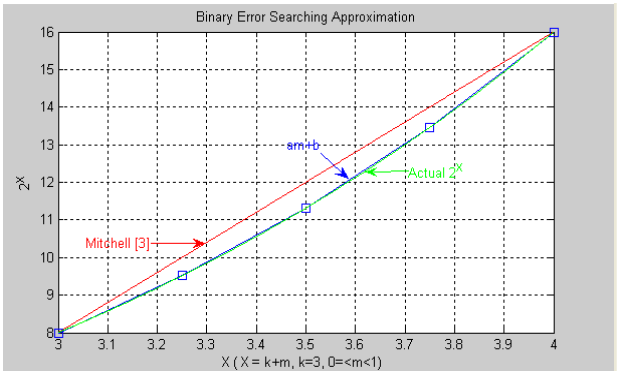

Fig. 6 Four-Region Binary Error Searching Approximation

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

1:[Input] a_i, b_j, m, where i, j=0,1,2,....2048.
2:Decide the numbers of regions X.
3:Decide the range of m in the X-region approximation.
4:Set initial value (min$a$,min$b$) = (a_0, b_0)
5: *for* b_j = 0 to 1 step 2^{-11}
6: *for* a_i = 0 to 2 step 2^{-8}
7 : *If* percent error(a_{i+1}, b_j)< percent error(a_i, b_j),
8: *then* (min$a$,min$b$) = (a_{i+1}, b_j).
9: *End* of a_i
10 : *End* of b_j
11 : [*Output*] (min$a$,min$b$).

Fig. 7 Our Proposed Algorithm using Binary Error Searching Schemes to Obtain the coefficients of am+b

Using our proposed algorithm shown in Fig. 7, the optimal coefficients of *mina* and *minb* for different regions of antilogarithmic conversion can be easily obtained. Therefore, 2-, 4-, 8-, and 16-region antilogarithmic conversion based on our proposed algorithmic can be modeled as Eq. (11) to (14), respectively, where $m_{dMSBits}$ is denoted the first d most significant bits after the point of m.

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k \times [m-(\frac{45}{256}) \times m_{7MSBits} + \frac{256}{256}], & 0 \le m < 0.5; \\ 2^k \times [m+(\frac{45}{256}) \times m_{7MSBits} + \frac{211}{256}], & 0.5 \le m < 1; \end{cases} \quad (11)$$

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k \times [m-(\frac{1}{4}) \times m_{7MSBits} + \frac{1024}{1024}], & 0 \le m < 0.25; \\ 2^k \times [m-(\frac{1}{16}+\frac{1}{32}) \times m_{7MSBits} + \frac{985}{1024}], & 0.25 \le m < 0.5; \\ 2^k \times [m-(\frac{1}{16}) \times m_{7MSBits} + \frac{906}{1024}], & 0.5 \le m < 0.75; \\ 2^k \times [m-(\frac{1}{4}+\frac{1}{16}) \times m_{7MSBits} + \frac{753}{1024}], & 0.75 \le m < 1; \end{cases} \quad (12)$$

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k \times [m-(\frac{71}{256}) \times m_{9MSBits} + \frac{2048}{2048}], & 0 \le m < 0.125; \\ 2^k \times [m-(\frac{55}{256}) \times m_{9MSBits} + \frac{2033}{2048}], & 0.125 \le m < 0.25; \\ 2^k \times [m-(\frac{18}{256}) \times m_{9MSBits} + \frac{1996}{2048}], & 0.25 \le m < 0.375; \\ 2^k \times [m-(\frac{17}{256}) \times m_{9MSBits} + \frac{1940}{2048}], & 0.375 \le m < 0.5; \\ 2^k \times [m+(\frac{7}{256}) \times m_{9MSBits} + \frac{1845}{2048}], & 0.5 \le m < 0.625; \\ 2^k \times [m+(\frac{30}{256}) \times m_{9MSBits} + \frac{1729}{2048}], & 0.625 \le m < 0.75; \\ 2^k \times [m+(\frac{55}{256}) \times m_{9MSBits} + \frac{1580}{2048}], & 0.75 \le m < 0.875; \\ 2^k \times [m+(\frac{85}{256}) \times m_{9MSBits} + \frac{1370}{2048}], & 0.875 \le m < 1; \end{cases} \quad (13)$$

$$Y' = 2^k 2^{m'} = \begin{cases} 2^k \times [m-(\frac{38}{128}) \times m_{9MSBits} + \frac{1024}{1024}], & 0 \le m < 0.0625; \\ 2^k \times [m-(\frac{34}{128}) \times m_{9MSBits} + \frac{1022}{1024}], & 0.0625 \le m < 0.125; \\ 2^k \times [m-(\frac{31}{128}) \times m_{9MSBits} + \frac{1020}{1024}], & 0.125 \le m < 0.1875; \\ 2^k \times [m-(\frac{24}{128}) \times m_{9MSBits} + \frac{1010}{1024}], & 0.1875 \le m < 0.25; \\ 2^k \times [m-(\frac{21}{128}) \times m_{9MSBits} + \frac{1004}{1024}], & 0.25 \le m < 0.3125; \\ 2^k \times [m-(\frac{15}{128}) \times m_{9MSBits} + \frac{989}{1024}], & 0.3125 \le m < 0.375; \\ 2^k \times [m-(\frac{10}{128}) \times m_{9MSBits} + \frac{974}{1024}], & 0.375 \le m < 0.4375; \\ 2^k \times [m-(\frac{6}{128}) \times m_{9MSBits} + \frac{960}{1024}], & 0.4375 \le m < 0.5; \\ 2^k \times [m+(\frac{1}{128}) \times m_{9MSBits} + \frac{932}{1024}], & 0.5 \le m < 0.5625; \\ 2^k \times [m+(\frac{5}{128}) \times m_{9MSBits} + \frac{914}{1024}], & 0.5625 \le m < 0.625; \\ 2^k \times [m+(\frac{12}{128}) \times m_{9MSBits} + \frac{879}{1024}], & 0.625 \le m < 0.6875; \\ 2^k \times [m+(\frac{18}{128}) \times m_{9MSBits} + \frac{846}{1024}], & 0.6875 \le m < 0.75; \\ 2^k \times [m+(\frac{26}{128}) \times m_{9MSBits} + \frac{798}{1024}], & 0.75 \le m < 0.8125; \\ 2^k \times [m+(\frac{30}{128}) \times m_{9MSBits} + \frac{772}{1024}], & 0.8125 \le m < 0.875; \\ 2^k \times [m+(\frac{40}{128}) \times m_{9MSBits} + \frac{702}{1024}], & 0.875 \le m < 0.9375; \\ 2^k \times [m+(\frac{47}{128}) \times m_{9MSBits} + \frac{649}{1024}], & 0.9375 \le m < 1; \end{cases} \quad (14)$$

We will present the simulation results of error percents and VLSI implementations for our proposed lower-error antilogarithmic converters in the next section.

IV. SIMULATION RESULTS AND VLSI IMPLEMENTATIONS

We simulate proposed two-region, four-region, eight-region and sixteen-region piecewise-linear antilogarithmic converters by Matlab software. The simulation of our proposed two-region error percentage which compared with Mitchell antilogarithmic converter approximation is shown in Fig. 8, where we can find the maximum positive error of proposed two-region antilogarithmic converter is 1.6138%, and the maximum negative error is -0.067%. The total error percentage is 1.6808%, using the 7 MSBs of the input for approximation.

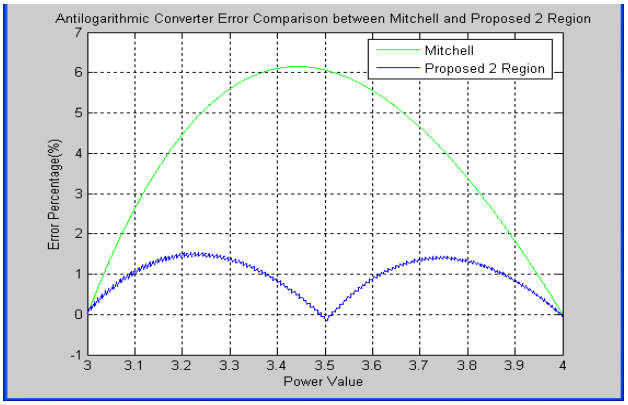

Fig. 8 Error Percentage of Our Proposed 2-region antilogarithmic converter

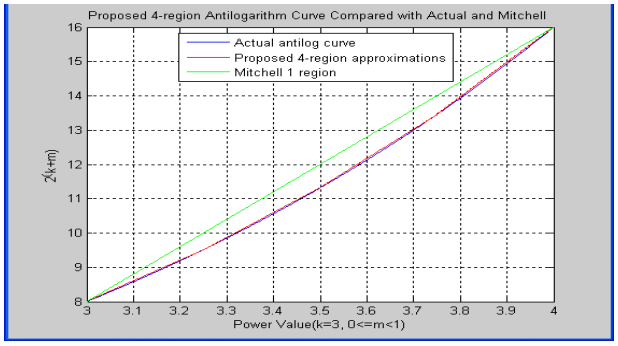

Fig. 9 Proposed 4-region piecewise-linear Antilog curve

Fig. 9 is the approximation curve for our proposed 4-region antilogarithmic conversion method compared with Mitchell's [3] and the actual conversion. We can observe that using our proposed methods, the antilogarithmic conversion will be very close to the actual conversion. The error percentage of proposed four-region antilogarithmic converter is shown in Fig. 10, where we can find the maximum positive error of proposed work is 0.4810%, and the maximum negative error is -0.08710%, respectively. The total error percentage is 0.5681%, using 7 MSBs of the input for approximation.

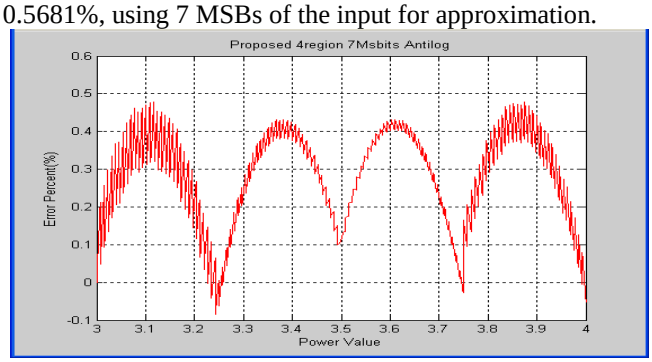

Fig. 10 Error Percentage of our proposed 4-region antilogarithmic converter

The error percentage of proposed eight-region antilogarithmic converter is shown in Fig. 11, where we can find the maximum positive error of proposed work is 0.133%, and the maximum negative error is -0.004%. The total error percentage is 0.137%, using 9 MSBs of the input for approximation. It can be observed that according to Table I, our proposed eight-region piecewise-linear approximation is lower to Kim's [6] eight-region method. And the hardware complexity will also be simpler than Kim's [6] method due to its full bit accuracy.

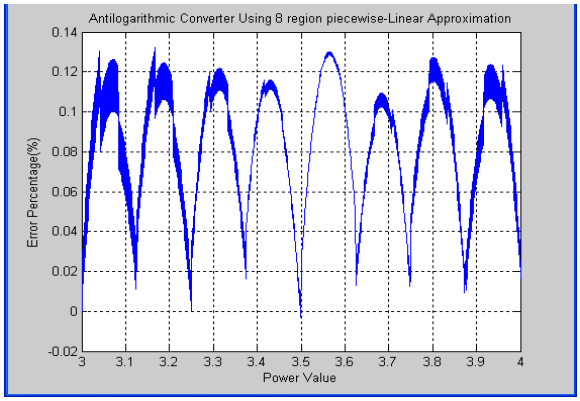

Fig. 11 Error Percentage of our proposed 8-region antilogarithmic converter

The simulation of error percentage of proposed sixteen-region antilogarithmic converter is shown in Fig. 12, where we can find the maximum positive error of proposed work is 0.058%, and the maximum negative error is -0.04%. The total error percentage is 0.098%, using the same 9 MSBs of the input for approximation.

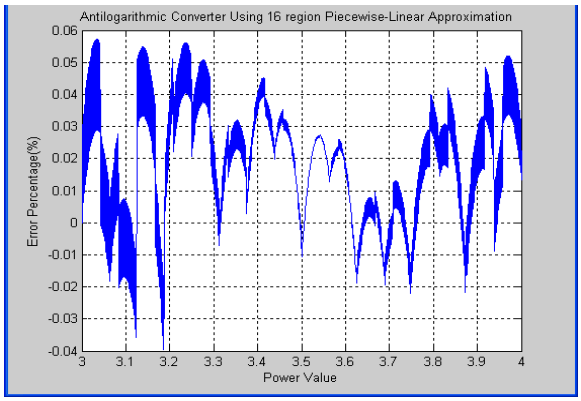

Fig. 12 Error Percentage of our proposed 16-region antilogarithmic converter

Table I: Comparisons of our proposed two-region, four-region, eight-region and sixteen-region method with previous methods

Items	Hall[4]	Hall[4]	Abed[5]	Abed [5]	Kim[6]	**Proposed**	**Proposed**	**Proposed**	**Proposed**
Range of *X*	[3 : 4)	[3 : 4)	[3 : 4)	[3 : 4)	[3 : 4)	**[3 : 4)**	**[3 : 4)**	**[3 : 4)**	**[3 : 4)**
Number of Regions	2	4	6	7	8	**2**	**4**	**8**	**16**
Fraction Bits	All Bits (26 Bits)	All Bits (26Bits)	7 Bits	9 Bits	All Bits (26 Bits)	**7 Bits**	**7 Bits**	**9 Bits**	**9 Bits**
Max. Positive Error (%)	1.5042%	0.3032%	0.9572%	0.3477 %	0.082%	**1.6138%**	**0.4810 %**	**0.133%**	**0.058%**
Max. Negative Error (%)	1.1155%	-0.4736 %	-0.5786 %	-0.2232 %	-0.07%	**-0.067%**	**-0.0871 %**	**-0.004%**	**-0.04%**
Total Error Range (%)	2.6197%	0.7768 %	1.5358 %	0.5709%	0.152%	**1.6808%**	**0.5681 %**	**0.137%**	**0.098%**

Comparisons of our proposed two-region, four-region, eight-region and sixteen-region antilogarithmic converters with previous methods are shown in Table I. We can obtain that the proposed two-region antilogarithmic conversion method attains the lower error percentage compared with Hall's 2-region approximation. The proposed four-region antilogarithmic conversion method also attains the lowest error percentage compared with Hall's 4-region and Abed's 6-region approximation. Meanwhile, the proposed eight-region antilogarithmic conversion method also attains the lowest error percentage compared with Abed's 7-region and Kim's 8-region approximation. And in the last column of Table I, we can find the total error range for our sixteen-region antilogarithmic conversion is only 0.098%.

VLSI implementations of our proposed 2-, 4-, 8- and 16-region antilogarithmic conversion are tabulated in Table II. All are realized using TSMC 0.18 μm CMOS technology.

Table II. VLSI Implementation of proposed antilogarithmic conversion

Method / Items	Proposed Work 2 region	Proposed Work 4 region	Proposed Work 8 region	Proposed Work 16region
Area (μm^2)	4993	6639	11819	18811
Delay (ns)	10	10	10	10

Although Hall's method in four-region linear approximation produces a lower error than Abed's six-region conversion, but it leads to complex hardware due to full bit-accuracy of input. Therefore, we simply compare our four-region work with Abed's six-region using the same 7 MSBs of the input for approximation. The proposed four-region antilogarithmic converter is realized on silicon using TSMC 0.18 μm CMOS technology, and the comparison is tabulated in Table III, which shows our proposed four-region antilogarithmic converter only costs 30% additional hardware (i.e., 6639 μm^2 vs. 5119 μm^2) with 1.7x reduction (i.e., 0.5681% vs. 1.5358% shown in Table III) in error percent range under the same delay constraints.

Table III. VLSI Implementation Comparison with Abed's 6-region conversion

Methods / Items	Abed[3] 6-region	**Proposed Work**
Area (μm^2)	5119	**6639**
Delay (ns)	10	**10**

V. CONCLUSIONS

In this paper, we have proposed lower-error antilogarithmic converters based on 2-egion, 4-region, 8-region and 16-region piecewise-linear approximations using Binary Error Searching schemes. The proposed technique provides lower errors, which can outperform previously proposed methods in the literature. In error and area comparisons with Abed's six-region methods in the literature, our proposed four-region antilogarithmic converter only cost 30% additional hardware with 1.7x reduction under the same delay constraints. The proposed 2-, 4-, 8-, 16- region antilogarithmic converters can be used to the overall LNS-based computation for 3-D computer graphics and Digital Signal Processing applications.

REFERENCES

1. T. A. Brubaker and J. C. Becker, "Multiplication using logarithms implemented with read-only memory," *IEEE Trans. Computers*, vol.C-24, no..8, pp.761-766, Aug. 1975.
2. M. J. Shulte and AJ. E. E. Swartzlander, "Hardware Designs for exactly rounded elementary functions," *IEEE Trans. Computers*, vol.43, no..8, pp.964-973, Aug. 1994
3. J. N. Mitchell, Jr., "Computer Multiplication and Division Using Binary Logarithms," *IRE Trans. on Electronic Computers*, vol. EC-11, pp. 512-517, August 1962.
4. E.L. Hall, D.D. Lynch, and S.J.Dwyer III, "Generation of Products and Quotients Using Approximate Binary Logarithms for Digital Filtering Applications," *IEEE Trans. Computers*, vol.19, pp.97-105, Feb. 1970
5. K. H. Abed and R. E. Siferd, "CMOS VLSI Implementation of a Low-Power Antilogarithmic Converter," *IEEE Trans. on Computers*, Vol. 52, No. 11, pp. 1421–1433, Nov. 2003.
6. Hyejung Kim, Byeong-Gyu Nam, Ju-Ho Sohn, Jeong-Ho Woo and Hoi-Jun Yoo, "A 231-MHz, 2.18-mW 32-bit Logarithmic Arithmetic Unit for Fixed-Point 3-D Graphics System," *IEEE Journal of Solid State Circuits*, vol. 41, no. 11, pp.2 373-2381, Nov. 2006
7. Suganth Paul, Nikhil Jayalumar, and Sunil P. Khatr, "A Fast Hardware Approach for Approximate, Efficient Logarithm and Antilogarithm Computations," *IEEE Trans. On Very Large Scale Integration*, vol. 17, no. 2, pp. 269-277, Feb. 2009
8. Dongdong Chen, Yu Zhang, Daniel Teng, Khan Wahid, Moon Ho Lee and Seok-Bum Ko, "A New Decimal Antilogarithmic Converter," *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 445-448, May 2009.
9. Tso-Bing Juang, Sheng-Hung Chen and Huang-Jia Cheng, "A Lower-Error and ROM-Free Logarithmic Converter for Digital Signal Processing Applications" *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 12, pp. 931-935, Dec. 2009.
10. R. Gutierrez and J. Valls, "Low Cost Hardware Implementation of Logarithm Approximation," *IEEE Trans. On Very Large Scale Integration*, vol. 19, no. 12, pp. 2326-2330, Dec. 2011
11. J.-A. Pineiro, "Algorithm and architecture for logarithm, exponential, and powering computation," IEEE Transaction on Computers, vol. 53, no. 9, pp. 1085–1096, Sep. 2004.
12. P. T. P. Tang, "Table-lookup algorithms for elementary functions and their error analysis," *Proc. 10th Symp. Comput. Arithmetic*, pp. 232–236. Jun. 1991.
13. M. J. Schulte and J. E. Stine, "Approximating elementary functions with symmetric bipartite tables," *IEEE Transactions on Computers*, vol. 48, no. 8, pp. 842–847, Aug. 1999.
14. J. E. Stine and M. J. Schulte, "The symmetric table addition method for accurate function approximation," *Journal of VLSI Signal Processing Systems*, vol. 21, pp. 167–177, Jun. 1999.
15. M. Combet, H. V. Zonneveld and L. Verbeek, "Computation of the base two logarithm of binary numbers, *IEEE Transactions on Electronic Computers*, Vol. 14, No. 6, pp. 863–867, June 1965.
16. Tso-Bing Juang, Pramod Kumar Meher and Kai-Shiang Jan, "High-Performance Logarithmic Converters Using Novel Two-Region Bit-Level Manipulation Schemes," *Proc. of 2011 VLSI-DAT (VLSI Symposium on Design, Automation, and Testing)*, pp.390-393, April 2011.
17. PALIOURAS AND T. STOURAITIS, "LOW-POWER PROPERTIES OF THE LOGARITHMIC NUMBER SYSTEM," PROC. 15TH IEEE SYMP. COMPUTER ARITHMETIC, PP. 229-236, JUNE 2001

AUTHORS PROFILE

Chao-Tsung Kuo received the PhD degree of the Electrical Engineering from National Chung Cheng University, Taiwan, in Jan. 1999. He is currently an assistant professor at National Quemoy University since Aug. 2008. His current research area includes VLSI design, High-Speed Arithmetic units design and Phase-Locked Loop design

Tso-Bing Juang received the PhD degree of the Computer Science and Engineering from National Sun Yat-Sen University, Taiwan, in Dec. 2004. He served as a lecturer and assistant professor at Ta Jen University during Aug. 1998 to Jan. 2006 prior to joining the department of Computer Science and Information, National Pingtung Institute of Commerce (NPIC), Taiwan in Feb. 2006, where now he is currently an Associate Professor since Aug. 2010. He has received the Best Tutoring Award and Outstanding Teaching Award by NPIC in 2010 and 2011, respectively. Since Dec. 2011, he was the visiting researcher at the Centre for High-Performance Embedded Systems (CHiPES), Nanyang Technological University (NTU) in Singapore under financial support by National Science Council in Taiwan.

Dr. Juang was a receiver of Xerox Best Thesis Award in Taiwan for the contributions of his master thesis in 1995. He has co-authored one textbook about full-custom IC design (in Chinese) and authored more than 40 research papers in referred international and domestic journals and conferences. He was the principal investigator and the co-investigator of 7 research projects supported by National Science Council in Taiwan since 2007. Also he was the principal investigator of 3 industry-academy research projects supported by the Ministry of Education in Taiwan, in 2007, 2008 and 2009, respectively. He served as the peer reviewers for IEEE journals such as IEEE Transactions on Circuits and Systems (I) and (II), IEEE Transactions on Computers, and IEEE Transactions on VLSI Systems.