Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.8 No.S3, 2019, pp.25-29

DOI: https://doi.org/10.51983/ajcst-2019.8.S3.2110

Performance Improvement of Antilogarithmic Converter Using 28 Regions Error Correction Scheme

A. T. A. Kishore Kumar[1] and R. Seshasayanan[2]
[1]Assistant Professor, Department of ECE, KMM Institute of Technology & Science, Andhra Pradesh, India &
Research Scholar, Sathyabama University, Chennai, India
[2]Associate Professor, Department of ECE, Anna University, Chennai, India
E-Mail: se_sha_sa@yahoo.com, kishorettt@gmail.com
(Received 24 March 2019; Revised 5 April 2019; Accepted 16 April 2019; Available online 24 April 2019)

***Abstract* - Logarithmic conversion is a significant portion of numerous digital signals processing system and other applications. The antilogarithmic transformation presented in this paper is able to support the antilogarithmic conversion of data with the number of bits up to thirty-two. An efficient FPGA hardware implementation of logarithmic operations is an alternative option used in arithmetic operations. In this paper, we implemented an efficient antilogarithmic converter using FPGA. This implementation is compared with 28 regions error correction scheme. The proposed hardware architecture having less area, delay with less error cost. This design is implemented using HDL tool and synthesized using Xilinx CAD tool. The implementation has with respect to existing antilog converter**.
***Keywords:* Antilogarithmic Converter, Logarithmic Number System (LNS), Efficient FPGA, Shift-And-Add Operation**

I. INTRODUCTION

The logarithm and antilogarithm are useful arithmetic concepts used many areas in science and engineering such as digital signal processing (DSP). DSP applications are widely used in audio, 3-D graphics processing, avionics, video, telecommunications and many other portal devices. In order to obtain maximum throughput and power and area reduction there is some tolerance for error.

Linear approximation techniques such as single and multiple-piecewise approaches have been discussed by Mitchell [1] and Hall *et al.,* [2] for converting the logarithmic numbers to the binary formats. Out of these approaches the Mitchell's approach has disadvantage in providing the sufficient accuracy. On the other hand, the multiple- piecewise method of linear approximation proposed by Hall *et al.,* has better advantage in terms of accuracy but difficult in the implementation due to its complexity. Another absence in these two approaches is the hardware realization of their methods. In the approach of Muller [3] the algorithms are suitable to software as well as hardware for computing elementary functions and issues that are related to the accurate floating point implementations. As per the literature, Schulte and Swartz lander [4] have developed the hardware designs by using the functions of reciprocal, square-root, 2x, and log2ðxÞ and has observed better results. These results sounds that it can able to produce exactly rounded results, while the approximations are tested for the circuits of multipliers, higher-order polynomials and Read Only Memories (ROMs). Finally, another proposed method such as Symmetric Bipartite Table Method proposed by Schulte and Stein [5] is also having the capabilities of higher accuracy even by the use of wide range of elementary functions. Here, the approach is designed with the use of carry propagate adder, and a Booth encoder and with its lookup tables.

A logarithmic-number-system (LNS) proposed by Arnold and Walter [6] describes about the use of restricted faithful rounding and its applications. This work also suggests that the model is not suitable for LNS applications. Later they come up with the solution by proposing a new approach which has been designed with the use of ROM. As per the recent studies carried out by Paliouras and Stouraitis [7] investigated that the parameters such as higher computational complexity, power savings are achieved in the digital circuits using LNS even while considering the conversion overhead. In these works, the designs such as antilogarithmic converters (implemented by using ROM and lookup tables) for measuring the power factors have been reported. Also, the reports define that the measured values are evident to state that the power saving capability has been greatly achieved and can be termed as a low-power antilogarithmic converter.

The literature of [1] and [2] defines the algorithm of linear approximation with the absence of ROM or lookup tables in order to minimize the hardware, the same have been reported in this work with the detailed comparison of existing work. It should be understood that the high speed and lower power capabilities can be a self-defeating task with the algorithms which are implemented by the requirement of ROM, lookup table and other complex hardware structures. Few literatures are existed for the implementation of 16-bit logarithmic and antilogarithmic converters by using CMOS circuits without error analysis [8]. Later, higher bits such as 32-bit logarithmic converter with high speed circuits such as fast logarithmic error correcting circuits have been reported [9]. Here, the major work is reported with the consideration of DSP applications for the requirement of low power and high speed parameters. In comparison with these literature works, our work will give an idea about the designs, algorithms for

error correction and antilogarithmic conversion. Along with these results, a detailed analysis has been made on the accuracy and usage estimations for the circuit level implementations in VLSI. These analyses would judge the circuit can be used as a low-power antilogarithmic converter.

Rapid multiplication algorithm for the simple digital circuits has been carried out by Hall *et al.*, [11]. These works are related to the addition or subtraction of logarithms, computation of approximate logarithms with the use of binary approximation antilogarithmic functions. This algorithm can be utilized for the implementation of parallel digital filters but might not be suitable for array multiplier rather the computation of single products, due to its complexity in design. In the literature a 32-bit unique binary-binary logarithmic converter is available which was designed by Abed and Sifred [12] using CMOS technology. It can able to calculate the single clock cycles using approximate algorithm and can be utilized for the implementation of pure combinational logic circuit designs. The logarithmic converter having a logarithmic shifter and the fast Leading One Detector (LOD) can be operated with high speed and also consumes less area. Still, minor errors can be introduced with the utilization of logarithmic approximation techniques. Many other works are available as per the literature related to the converters implemented using combinational logic circuits and antilogarithmic functions of single clock cycle. One of such approach is carried out by Abed and Sifred [13] for the design of 32-bit antilogarithmic converter using the CMOS VLSI circuits, which generates data for Digital Signal Processing (DSP) applications.

The remainder of this paper is organized as follows. Section 2 will give the existing work done in antilogarithmic converters. The mathematical expression for the error correction discussed in section 3. FPGA implementation of area efficient lower error antilogarithmic converters using constant compensation schemes are given in Sect. 4. Section 5 discuss about the result. Conclusions and future work will be drawn in Section 6.

II. EXISTING WORK

In previous years, various antilogarithmic approximation methods were proposed, all are based on piecewise-linear approximation. ROM-based methods can achieve fast and more accurate conversions; however, the area costs for ROM-based methods may become tremendously higher while the bit width of inputs increases. On the other hand, using polynomial approximations will reduce the area costs while sacrificing the accuracy and speed. Compared with these two kinds of implementations, shift-and-add methods can be used to achieve better design trade-offs between accuracy, area consumptions and speed. The common characteristic of these antilogarithm converters is to use shift-and-add operations instead of complex operations such as multiplications to reduce the area costs.

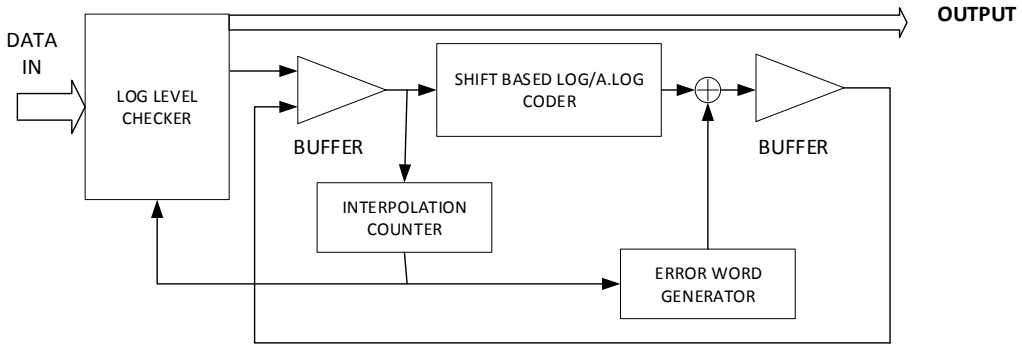

Fig. 1 Log cum antilog coder

III. MATHEMATICAL EXPRESSION

In 1962, Mitchell proposed an algorithm based on simple add and shift operation for multiplication and division to gain speed by using the piecewise linear approximation. Generally speaking, the calculation of antilogarithmic converter is based on 2 power of X. Equations (1) to (4) is used to show the operations of the antilogarithmic conversion. Suppose $A = p + m$ is the input value of antilogarithmic converter, where p is denoted as the integer part and m is denoted as the decimal fraction part.

$A= p+m$, $B= \text{Antilog}_2(A) = 2^A = 2^p 2^m$ where p is any integer; $0 \le m < 1$ (1)

$B'_{\text{Mitchell's 1-region}} = 2^p(1+m)$, $0 \le m < 1$ (2)

$\text{Error}_{\text{Mitchell's 1-region}} = 2^p x((1+m) - 2m)$, $0 \le m < 1$ (3)

$$\text{Percent Error}_{\text{Mitchell's1-region}} = \frac{\text{ErrorMitchell's 1-region}}{B}$$

$$= \frac{((1+m)-2m)}{2^m} \text{ X } 100\ \%\ 0 \le m < 1 \quad (4)$$

Equations (1) to (4) is used to show the operations of the antilogarithmic conversion. Suppose $A = p + m$ is the input value of antilogarithmic converter, where p is denoted as the

integer part and *m* is denoted as the decimal fraction part. Mitchell proposed a logarithmic and antilogarithmic converter with one-region linear approximation scheme, where he approximated 2*m* in *m* intervals as a straight line (1 + *m*) as shown in Eq. (2). *B* Mitchell'1-region is denoted as the approximation value adopted by Mitchell's method. The error of Mitchell's one region method is given in the Eq. (3). Taking a short example, letting input $A = 0.6 = 0 + 0.6$, here we take $p = 0$ and $m = 0.6$. The output *B* equals to $20.6 = 20 \times 20.6 = 20.6 \fallingdotseq 1.5157$. In Eq. (2), Mitchell approximated output *B* as $20 \times (1 + 0.6)$, it is equal to 1.6. The term "Error" in Eq. (3) is obtained by subtracting the approximate and the real value. And, the term "Percent Error" in Eq. (4) is indicated as the error divided by the real value and then be multiplied by 100%. Therefore, the maximum error and the percent error by Mitchell's method is 0.0843 and 6.1476%, respectively. It can be observed that Mitchell's method is simple for implementation, but the accuracies are less high.

To reduce the approximation error, area cost and delay time as minimum as possible, the proposed method with multiple regions of constant compensation could achieve better performance than previously reports in the literature. In Sect. 3, area-efficient and highly-accurate antilogarithmic converters using multiple regions of constant compensation schemes will be proposed.

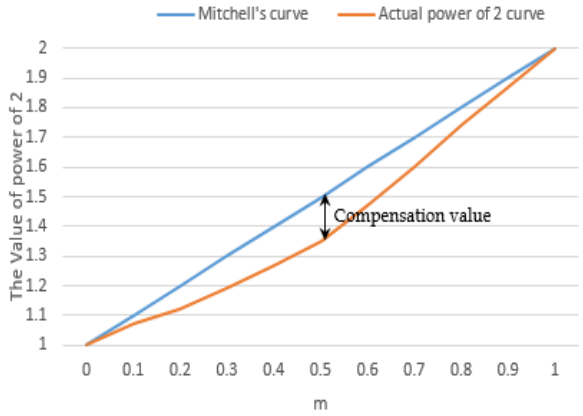

Fig. 2 The curve of compensation values between 2^m and 1+m

IV. PROPOSED METHODS AND FPGA IMPLEMENTATION OF AREA EFFICIENT LOWER ERROR ANTILOGARITHMIC CONVERTERS BYUSING 28 REGIONS ERROR CORRECTIONSCHEME

Based on the data analysis of antilogarithmic conversion usingMitchell's method and actual antilogarithmic conversion, we found that there is a possibility of adding the common correction terms in a certain range for error minimization in logarithm arithmetic.

Here, a process to add the difference of approximated values and actual value will not give much accurate result.

For an error minimization, we can sub-divide the entire line in multiple regions and add the mean difference of correction terms in a certain define range. But, it increases the number of sub-sections a hardware penalty. The input format of x for antilogarithmic conversion can be supposed like x=0.m-1 m2 m-3m-4 up to m-26.

Here we use mantissa's 7 most significant bits (MSBS) should be used for adjustment to obtain a simple correcting circuit. The mathematical formulation of proposed antilogarithmic conversion and error correction value is shown in Eq. (5) and (6).

Antilog (A) = $2^A = 2^P x 2^x$ (1+x+error correction value, $0 \le x < 1$ (5)

Error correction value = $c = \pm\Sigma 2^{-i}$, where i is any positive integer value (6)

We can observe that only adder/subtractor and simple combinational logic unit were used in hardware implementation. It should be noted that there is not any shifter and look-up table structure be used in our proposed antilogarithmic converters. In practical hardware realization, once the accuracy is given, the compensation values can be obtained by the proposed method at one time and thus the real VLSI circuit implementations can be found. Therefore, the complexity of the searching algorithm will not compromise the performance of VLSI circuit implementations of antilogarithmic converters.

TABLE I CONDITIONS OF ADDING THE CORRECTED VALUES M_3=0 ($0 \le M_{-1} \le 0.25$) FOR 28 REGIONS

m_3=0 (0≤m_3≤0.25) ($m_{-4}m_{-5}m_{-6}m_{-7}$)	**Corrected value**					
	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
0000	0	0	0	0	0	0
0001	0	0	0	0	1	0
0010	0	0	0	1	0	0
0011	0	0	0	1	1	0
0100	0	0	1	0	0	1
0101	0	0	1	0	1	1
0110	0	0	1	1	0	1
0111	0	0	1	1	1	1
1000	0	1	0	0	0	1
1001	0	1	0	0	1	0
1010	0	1	0	1	0	0
1011	0	1	0	1	1	0
1100	0	1	1	0	0	1
1101	0	1	1	1	0	0
1110	0	1	1	1	0	0
1111	0	1	1	1	0	0

TABLE II CONDITIONS OF ADDING THE CORRECTED VALUES M_3=1 ($0.25 \le M_{-1} \le 1$) FOR 28 REGIONS

m_3=1 (0.25≤m_-1≤1) (m_-4m_-5m_-6m_-7)	Corrected value					
	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
0000	0	1	1	1	0	0
0001	0	1	1	1	0	0
0010	0	1	1	1	1	1
0011	1	0	0	0	0	1
0100	1	0	0	1	0	0
0101	1	0	1	0	0	0
0110	1	0	0	1	1	1
0111	1	0	0	0	1	1
1000	1	0	0	0	0	0
1001	0	1	1	1	0	0
1010	0	1	1	0	0	0
1011	0	1	0	1	0	0
1100	0	1	0	0	0	0
1101	0	0	1	0	1	1
1110	0	0	0	1	1	0
1111	0	0	0	0	0	0

The following is the pseudo code used to obtain optimal compensation coefficient.

Pseudo code to obtain optimal compensation coefficient

Input: R (given R regions starting with region 1 to R)
Output: c_i (R values for c_i)
for each region I (1 i R) do
forc_i= -1 to 1 step 1/1024 do
searching the values of $c_i$which can produce the minimum percent errors end for
end for
set the values of c_i into the combination of power of 2.

V. RESULTS AND DISCUSSION

We use VHDL to design the proposed antilogarithmic converters. The 28 regions converters design is coded and simulated using CAD tool. Using Xilinx CAD tool the code is synthesized and shown in fig.5 and fig.6. The device summery also shown in table III for both error correction scheme.

TABLE III DEVICE UTILIZATION SUMMERY

Device Utility	Selected Device 3s500efg320-5
	28 regions
Number of slices out of 4656	15
Number of 4 input LUTs out of 9312	27
Number of IOBS out of 232	23
Maximum path delay in ns	13.331

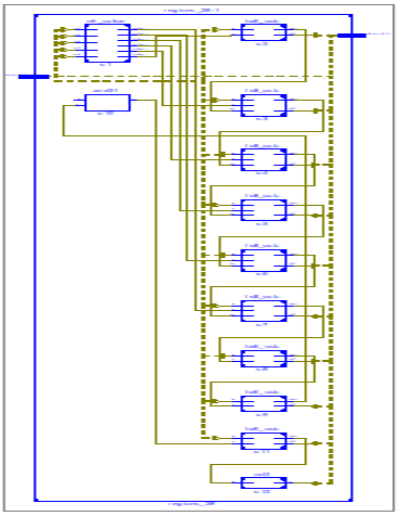

Fig. 3 RTL view for 28 region error correction scheme

The proposed work is compared with existing work and the table 4 gives the overall comparisons

TABLE IV COMPARISONS OF PRESENT ERRORS FOR THE PROPOSED CONVERTERS USING 28 REGIONS CONSTANT COMPENSATION SCHEMES METHOD

Items	Hall *et al.*, (1970)	Abed and Siferd (2003)	Proposed work
Region no	4	6	28
Maximum positive error	0.3032%	0.9572%	0.59%
Maximum negative error	-0.4736%	-0.5786%	-0.01%
Percent error range	0.7768%	1.5358%	0.609%

Since the number of partition regions for antilogarithmic conversion increases, the percent errors for approximation can be reduced with increased hardware costs. It should be noted that the hardware cost of antilogarithmic conversion for Mitchell's method is the lowest with highest percent errors. Since we expect to obtain lower approximations with reduced hardware implementations, Although the number of partition regions for our proposed antilogarithmic approximations is higher than previous methods in the literature, the delay/area costs can be reduced since the delay/area of the proposed implementations can be optimized by synthesis software only using constants for approximations with the input numbers. The percent errors of proposed converters could achieve 1.8319%, and 0.609% for 28-region constant compensation schemes, respectively. It can be seen that the proposed antilogarithmic converters can achieve the lowest percent errors compared with other methods in the literature.

VI. CONCLUSION

In this paper, we have proposed area-efficient and highly-accurate hardware architecture of antilogarithmic converters by using 28 regions error correction schemewith multiple regions of constant compensation schemes. The proposed

convertersusing FPGAs can achieve area-efficient and lower-delay with reducedpercent errors. The complexity of converters is very simpleto implement in FPGAs. It can be applied to digital camera and computation-intensive applications to reduce the computation efforts. We believe that the proposed antilogarithmic converter may be used to ease the area and time-consuming multiplications for real-time DSP applications. It can be applied to digital camera and computation-intensive applications to reduce the computation efforts.

REFERENCES

[1] J. N. Mitchell Jr., "Computer Multiplication and Division Using Binary Logarithms", *IRE Trans. Electronic Computers*, Vol. 11, pp. 512-517, Aug. 1962.

[2] 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.

[3] J. M. Muller, *Elementary Functions: Algorithms and Implementation*, Birkhauser, 1997.

[4] M. J. Schulte and E.E. Swartzlander Jr., "Hardware Designs for Exactly Rounded Elementary Functions", *IEEE Trans. Computers*, Vol. 43, No. 8, pp.964-973, Aug. 1994.

[5] M.J. Schulte and J.E. Stine, "Symmetric Bipartite Tables for Accurate Function Approximation", *Proc. 13th Symp. Computer Arithmetic*, pp. 175- 183, 1997.

[6] M.J. Arnold and C. Walter, Unrestricted Faithful Rounding Is Good Enough for Some LNS Applications, *Proc. 15th IEEE Symp. Computer Arithmetic*, pp. 237-246, June 2001.

[7] V. Paliouras and T. Stouraitis, "Low-Power Properties of the Logarithmic Number System", *Proc. 15th IEEE Symp. Computer Arithmetic*, pp. 229-236, June 2001.

[8] K.H. Abed and R.E. Siferd, "CMOS VLSI Implementation of 16-Bit Logarithm and Anti-Logarithm Converters", *Proc. IEEE Midwest Symp. Circuits and Systems*, pp. 776-779, Aug. 1999.

[9] K.H. Abed and R.E. Siferd, "CMOS VLSI Implementation of a 32-Bit Logarithmic Converter", *IEEE Trans. Computers*, submitted.

[10] S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego, Calif.: California Publishing, 1997.

[11] 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 Transactions on Computers*, Vol. C-19, No. 2, pp. 97-105, February 1970.

[12] K. H. Abed and R. E. Sifred, "CMOS VLSI Implementation of a Low-Power Logarithmic Converter," *IEEE Transactions on Computers*, Vol. 52, No. 11, pp. 142 1-1433, November 2003.

[13] K. H. Abed and R. E. Sifred, "VLSI Implementation of a Low-Power Antilogarithmic Converter," *IEEE Transactions on Computers*, Vol. 52, No. 9, pp. 1221-1228, September 2003.