Study of Various Types of Logarithm Multiplier

Ashica Anal [1], Sachin Bandewar [2],
[1,2]*Electronics & Communication Engineering Department,*
RKDF University, Bhopal, India

***Abstract*—The application of digital signal processing (DSP) and image processing, arithmetic operations are essential components. Particularly, digital filters needed to be designed with an efficient multiplier. Nowadays, there are lots of handheld portable battery-operated devices require the hardware efficient and error free computer arithmetic operations. As we aware that multiplier is hardware thirsty and make response slow of any hardware architecture. A lot of research efforts have been directed to design hardware and performance efficient multiplier. The real world is full of applications of Logarithm Number System (LNS) based multiplier which motivates the researchers to design it an efficient LNS. This article shows the importance of the Logarithm numbers and its importance for hardware implementation for multipliers. In this article authors demonstrate various types of logarithm multiplier. This paper try to summarized various types of logarithm multiplier at one platform.**

***Keywords*— Logarithm multiplier, Antilogarithm converter, Logarithm converter, Improved operand decomposition.**

I. Introduction

Arithmetic operation is an important component for Digital Signal Processing (DSP) and Image processing applications [1, 2]. Especially, digital filters needed to be designed with an efficient multiplier [3]. Nowadays, there are lots of handheld portable battery-operated devices require the hardware efficient and error free computer arithmetic operations [4]. As we aware that multiplier is hardware thirsty and make response slow of any hardware architecture. A lot of research efforts have been directed to design hardware and performance efficient multiplier.

The real world is full of applications of Logarithm Number System (LNS) based multiplier which motivates the researchers to design it an efficient LNS. Few of these applications are briefly discussed here. Due to numerous applications of the logarithm multiplier in DSP, it motivates researchers to come forefront to contribute research efforts. In next Section Logarithm Number System, their importance and logarithm multiplier have been introduced.

II. Logarithm Number System

Multiplication and division operations have shown the delay and the requirement of extra hardware. Due to these reasons, most of the weighted binary number arithmetic units may have to compromise at speed and area of circuits. DSP applications have a primary goal to process an operation faster with an efficient hardware which may not be full-filled due to the issues of the binary number system. To overcome these issues, LNS addresses to these issues and overcomes the technical gaps. LNS provides a new approach to speed up the slow multiplication and division with a conventional weighted number, it avoids the issues inherent in RNS. Mathematicians have used logarithms to simplify the mathematical operations like multiplication, division, etc. because these operations can be performed by using addition and subtraction. LNS multipliers are advantageous in terms of speed and accuracy over other multipliers circuits [5, 6]. LNS based on multiplier supports integer or FXP data and FLP data both data types. FXP multiplication is used in DSP applications, Due to its an easier algorithm, faster implementation and a clear understanding. The FLP representation is a suitable choice at the place where decimal notations have essential criteria to represent the non-integer number. The floating-point logarithmic numbers are kept in the following format [7]: Here, first 2-bits are reserved

Flags (2 bits)	Sign (1 bit)	Integer (x bits)	Fraction (y bits)

for flag code of a special exception (like negative, zero, Not a Number (NaN) and $+/-\infty$) [8]. A detailed study of the reported LNS methods, available design algorithms and design limitations is presented in section 3 to 6.

III. Various Methods of LNS Multiplication

Conventionally, various methods of LNS multiplication are divided in two broaden categories: (1) methods that use lookup tables (LUT) and interpolation and (2) methods based on the Mitchell's algorithm [5]. Foreplaning of logarithm-based on multiplication methods is arranged in the organization model given in Figure 1.

IV. LUT and Interpolation Based Logarithmic Multiplication

The most traditional method for logarithmic multiplication is the multiplication get performed by using lookup table. In this method complete possible values of logarithm are stored in ROM [9]. Major disadvantage of this method is the long memory space and may not suitable of ROM usage for long word. Especially, in case of higher than 20 bits LUT are required. Meanwhile, it has suitability for small bits, like 4 to 12 bits. Researchers work to investigated for using small lookup tables is known as bit partitioning [10, 11]. In some research works, bipartite tables are used for table reduction [10–14]. In this table reduction techniques two equally sized LUTs is used in parallel access. It makes design faster in speed

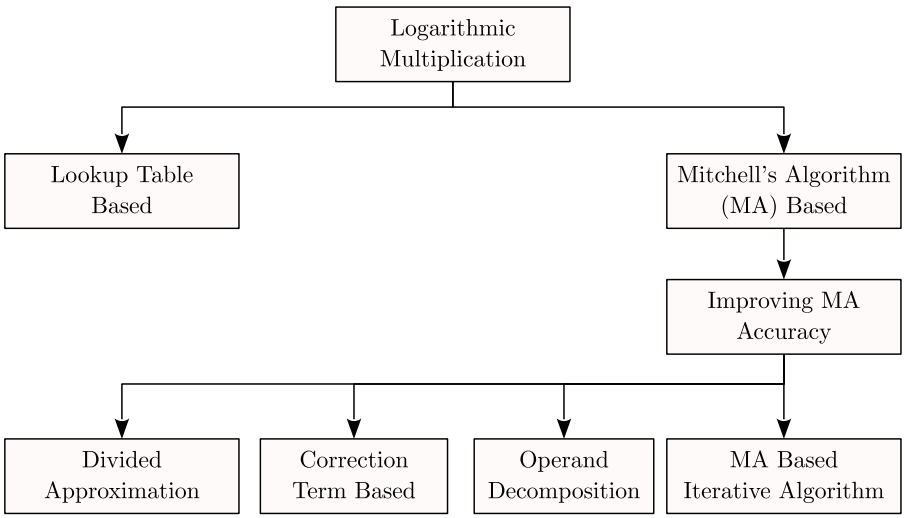

Fig. 1. Organization Model of Logarithmic Multiplication

and hardware efficient. This approach is useful for 10-12 bits range multiplication process. However, the extension of the bipartite table, is known as the multipartite table had suffered increment in size. Its limits the approach to only 13-bits [15].

In spite of the attempts at reducing the amount of memory to store lookup tables, there is strong interest in Mitchell's algorithm, which completely eliminates the use of lookup tables.

V. Mitchell's Multiplication Algorithm

In 1962, Mitchell proposes an algorithm based on binary algorithm which provides method for computer multiplication and division [5]. At Mitchell's algorithm simply add and shift operation is required to multiply any two numbers. For ease understanding of Mitchell's algorithm, we can go step by step for multiplication of two inputs name as A and B. Mitchell's algorithm shows some error due to his fraction part of logarithms. At Mitchell's method the error lies between zero to 11.1% and average error is 3.85% [16], so we can say that the maximum possible error (MPE) is around 11.1% and it occur when both of the fraction parts are equal to 0.5 [16]. For minimizing error in future, it is must necessary to understand step by step error analysis of Mitchell's multiplication algorithm.

A. Divided Approximation (DA) Based Correction

In Mitchell's logarithm that uses a piecewise linear curve and producing larger errors was later improved [17–31]. In these methods, authors have suggested a conversion method to achieve high accuracy with lower delay and area costs.

1) Hall's Correction Coefficients: Hall's algorithm [32] is uses all bits in the mantissa for adjustment. Therefore, requires more adders due to the slope multiplied by the mantissa. Mantissa was divided into four sub parts and applying linear piecewise approximation. It reduced the maximum error percentage of logarithmic multiplication from 11.1 % in Mitchell's method to 1.3%.

2) SanGregory's Correction Coefficients: SanGregory's proposed correcting algorithm for making logarithm multiplication fast. It uses only mantissa's four MSB for adjustment of concatenated result [26]. It has two region conversion for improving accuracy of Mitchell's logarithm at the cost of small hardware over-head in from of ROM circuits [26]. The proposed algorithm has performed four basic operations:

1) Determination of the leading one bit,
2) Re-alignment of all bits,
3) Generation of characteristics bit and
4) Adjustment of concatenated result.

Its two region conversion methods is performed using only combinational logic and requires no multiplications.

3) Abed and Siferd's Correction Coefficients: Abed and Siferd developed correction algorithm that required trade-off between the accuracy, speed and complexity. The equations for the 2–region correcting logarithm algorithm Maximum percent error for 2–region correcting logarithm is range over -0.9299 to 0.5544 and -0.5631 to 1.3310 for antilogarithm equations. The maximum error using the 6–region antilogarithm approximation ranges over -0.5786 to 0.9572. A higher approximation region demands more area and complex circuitry. Therefore, Juang's use a two-region conversion method to achieve high accuracy with low area and complexity of circuit.

4) Juang's Correction Coefficients: Juang et. al [22] proposed a two-region bit level manipulation schemes to achieve high accuracy with area, time and efficient hardware implementation. The maximum error using the Juang *et al.* 2–region logarithm approximation ranges over 0 to 0.0319 and ranges over -0.60 to 1.72 for antilogarithm converter.

B. Correction Term–Based Methods

The correction term-based methods analyzed error generated by the Mitchell algorithms and analysis used for a method which improves the accuracy. It can be achieved

by adding a correction term either to final product or the logarithm summation [5, 33]. Mitchell's developed analytical expressions for error and be added error to the product result. For Mitchell's Error Correction (MEC) carryover bit from the mantissa part to the integer part determines which one of the following two equations are to be used for correction:

$$P_{\text{M.E.C.}} = P_{\text{M.A.}} + 2^{k_1+k_2} x_1 x_2 \quad \text{where } x_1 + x_2 < 1 \tag{1}$$

$$P_{\text{M.E.C.}} = P_{\text{M.A.}} + 2^{k_1+k_2} y_1 y_2 \quad \text{where } x_1 + x_2 \geq 1 \tag{2}$$

C. Operand Decomposition

The operand decomposition is independent approach of minimizing error and applicable to all previous logarithmic multiplication approaches [16]. Suppose, for multiplying two n–bit binary numbers X and Y at first, the operand X and Y are decomposed into the following four operands A, B, C and D. where decomposed operands are calculated using the following equations:

$$A = X + Y \tag{3}$$

$$B = X \cdot Y \tag{4}$$

$$C = \overline{X} \cdot Y \tag{5}$$

$$D = X \cdot \overline{Y} \tag{6}$$

The product is computed from the decomposed operands using the following property.

$$X \cdot Y = (A \cdot B) + (C \cdot D) \tag{7}$$

The decomposition increases the number of 0 bits and hence decreases the switching power in the multiplication operation. Due to increases 0 bits, decreases the chances of a carryover from the mantissa part. Nandan *et al.* extended operand decomposition concept at 2017 and 2018 [34, 35].

D. Iterative Logarithmic Multiplier

Iterative logarithmic approximation is based on the correction terms, calculated immediately after the calculation of the product which avoids the comparison of the sum of mantissas with '1'. In this way, high level of parallelism can be achieved by the principle of pipelining, thus the basic block for multiplication requires less logic resources and increasing the speed of the multiplier with error correction circuits. Durgesh *et al.* proposed efficient architecture of iterative logarithm multiplier at 2017 and 2018 [36].

VI. Conclusion

This article shows the importance of the Logarithm numbers and its importance for hardware implementation for multipliers. In this article authors demonstrate various types of logarithm multiplier. Authors try to summarized various types of logarithm multiplier at one platform.

References

[1] S. Z. M. Naziri, R. C. Ismail, and A. Y. M. Shakaff, "The design revolution of logarithmic number system architecture," in *2014 2nd International Conference on Electrical, Electronics and System Engineering (ICEESE)*, Dec 2014, pp. 5–10. [Online]. Available: https://doi.org/10.1109/ICEESE.2014.7154603

[2] A. Klinefelter, J. Ryan, J. Tschanz, and B. H. Calhoun, "Error-energy analysis of hardware logarithmic approximation methods for low power applications," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 2361–2364. [Online]. Available: https://doi.org/10.1109/ISCAS.2015.7169158

[3] I. Kouretas, C. Basetas, and V. Paliouras, "Low-power logarithmic number system addition/subtraction and their impact on digital filters," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2196–2209, Nov 2013. [Online]. Available: https://doi.org/10.1109/TC.2012.111

[4] Z. Abid, D. A. El-Dib, and R. Mudassir, "Modified operand decomposition multiplication for high performance parallel multipliers," *Journal of Circuits, Systems and Computers*, vol. 25, no. 12, p. 1650149, 2016. [Online]. Available: https://doi.org/10.1142/S0218126616501498

[5] J. N. Mitchell, "Computer multiplication and division using binary logarithms," *IRE Transactions on Electronic Computers*, vol. EC-11, no. 4, pp. 512–517, Aug 1962. [Online]. Available: https://doi.org/10.1109/TEC.1962.5219391

[6] D. K. Kostopoulos, "An algorithm for the computation of binary logarithms," *IEEE Transactions on Computers*, vol. 40, no. 11, pp. 1267–1270, Nov 1991. [Online]. Available: https://doi.org/10.1109/12.102831

[7] H. Fu, O. Mencer, and W. Luk, "Fpga designs with optimized logarithmic arithmetic," *IEEE Transactions on Computers*, vol. 59, no. 7, pp. 1000–1006, July 2010. [Online]. Available: https://doi.org/10.1109/TC.2010.51

[8] J. Detrey and F. de Dinechin, "A vhdl library of lns operators," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, Nov 2003, pp. 2227–2231 Vol.2. [Online]. Available: https://doi.org/10.1109/ACSSC.2003.1292376

[9] M. G. Arnold and S. Collange, "A real/complex logarithmic number system alu," *IEEE Transactions on Computers*, vol. 60, no. 2, pp. 202–213, Feb 2011. [Online]. Available: https://doi.org/10.1109/TC.2010.154

[10] F. J. Taylor, R. Gill, J. Joseph, and J. Radke, "A 20 bit logarithmic number system processor," *IEEE Transactions on Computers*, vol. 37, no. 2, pp. 190–200, Feb 1988. [Online]. Available: https://doi.org/10.1109/12.2148

[11] D. M. Lewis, "Interleaved memory function interpolators with application to an accurate lns arithmetic unit," *IEEE Transactions on Computers*, vol. 43, no. 8, pp. 974–982, Aug 1994. [Online]. Available: https://doi.org/10.1109/12.295859

[12] D. Das Sarma and D. W. Matula, "Faithful bipartite rom reciprocal tables," in *Proceedings of the 12th Symposium on Computer Arithmetic*, July 1995, pp. 17–28. [Online]. Available: https://doi.org/10.1109/ARITH.1995.465381

[13] H. Hassler and N. Takagi, "Function evaluation by table look-up and addition," in *Proceedings of the 12th Symposium on Computer Arithmetic*, July 1995, pp. 10–16. [Online]. Available: https://doi.org/10.1109/ARITH.1995.465382

[14] M. J. Schulte and J. E. Stine, "Symmetric bipartite tables for accurate function approximation," in *Proceedings 13th IEEE Sympsoium on Computer Arithmetic*, July 1997, pp. 175–183. [Online]. Available: https://doi.org/10.1109/ARITH.1997.614893

[15] F. de Dinechin and A. Tisserand, "Multipartite table methods," *IEEE Transactions on Computers*, vol. 54, no. 3, pp. 319–330,

March 2005. [Online]. Available: https://doi.org/10.1109/TC.2005.54

[16] V. Mahalingam and N. Ranganathan, "Improving accuracy in mitchell's logarithmic multiplication using operand decomposition," *IEEE Transactions on Computers*, vol. 55, no. 12, pp. 1523–1535, Dec 2006. [Online]. Available: https://doi.org/10.1109/TC.2006.198

[17] K. H. Abed and R. E. Siferd, "Cmos vlsi implementation of a low-power logarithmic converter," *IEEE Trans. Comput.*, vol. 52, no. 11, pp. 1421–1433, Nov. 2003. [Online]. Available: https://doi.org/10.1109/TC.2003.1244940

[18] K. H. Abed and R. E. Siferd, "Cmos vlsi implementation of a low-power logarithmic converter," *IEEE Transactions on Computers*, vol. 52, no. 11, pp. 1421–1433, Nov 2003. [Online]. Available: https://doi.org/10.1109/TC.2003.1244940

[19] K. H. Abed and R. E. Siferd, "Vlsi implementation of a low-power antilogarithmic converter," *IEEE Transactions on Computers*, vol. 52, no. 9, pp. 1221–1228, Sep. 2003. [Online]. Available: https://doi.org/10.1109/TC.2003.1228517

[20] K. H. Abed and R. E. Siferd, "Vlsi implementations of low-power leading-one detector circuits," in *Proceedings of the IEEE SoutheastCon 2006*, March 2006, pp. 279–284. [Online]. Available: https://doi.org/10.1109/second.2006.1629364

[21] J. Y. L. Low and C. C. Jong, "Unified mitchell-based approximation for efficient logarithmic conversion circuit," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1783–1797, June 2015. [Online]. Available: https://doi.org/10.1109/TC.2014.2329683.

[22] T. Juang, S. Chen, and H. Cheng, "A lower error and rom-free logarithmic converter for digital signal processing applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 12, pp. 931–935, Dec 2009. [Online]. Available: https://doi.org/10.1109/TCSII.2009.2035270

[23] T. Juang, P. K. Meher, and K. Jan, "High-performance logarithmic converters using novel two-region bit-level manipulation schemes," in *Proceedings of 2011 International Symposium on VLSI Design, Automation and Test*, April 2011, pp. 1–4. [Online]. Available: https://doi.org/10.1109/VDAT.2011.5783555

[24] C. Liu, S. Ou, K. Chang, T. Lin, and S. Chen, "A low-error, cost-efficient design procedure for evaluating logarithms to be used in a logarithmic arithmetic processor," *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1158–1164, April 2016. [Online]. Available: https://doi.org/10.1109/TC.2015.2441696

[25] S. L. SanGregory, C. Brothers, D. Gallagher, and R. Siferd, "A fast, low-power logarithm approximation with cmos vlsi implementation," in *42nd Midwest Symposium on Circuits and Systems (Cat. No.99CH36356)*, vol. 1, Aug 1999, pp. 388–391 vol. 1. [Online]. Available: https://doi.org/10.1109/MWSCAS.1999.867287

[26] S. Paul, N. Jayakumar, and S. P. Khatri, "A fast hardware approach for approximate, efficient logarithm and antilogarithm computations," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 2, pp. 269–277, Feb 2009. [Online]. Available: https://doi.org/10.1109/TVLSI.2008.2003481

[27] C. Kuo and T. Juang, "A lower error antilogarithmic converter using novel four-region piecewise-linear approximation," in *2012 IEEE Asia Pacific Conference on Circuits and Systems*, Dec 2012, pp. 507–510. [Online]. Available: https://doi.org/10.1109/APCCAS.2012.6419083

[28] T.-B. Juang, H.-L. Kuo, and K.-S. Jan, "Lower-error and area-efficient antilogarithmic converters with bit-correction schemes," *Journal of the Chinese Institute of Engineers*, vol. 39, no. 1, pp. 57–63, 2016. [Online]. Available: https://doi.org/10.1080/02533839.2015.1070692

[29] D. Nandan, J. Kanungo, and A. Mahajan, "An efficient vlsi architecture design for antilogarithmic converter by using the error correction scheme," in *International Conference on Signal Processing (ICSP 2016)*, Nov 2016, pp. 1–5. [Online]. Available: https://doi.org/10.1049/cp.2016.1445

[30] C.-T. Kuo and T.-B. Juang, "Area-efficient and highly accurate antilogarithmic converters with multiple regions of constant compensation schemes," *Microsystem Technologies*, vol. 24, no. 1, pp. 219–225, Jan 2018. [Online]. Available: https://doi.org/10.1007/s00542-016-3238-z

[31] E. L. Hall, D. D. Lynch, and S. J. Dwyer, "Generation of products and quotients using approximate binary logarithms for digital filtering applications," *IEEE Trans. Comput.*, vol. 19, no. 2, pp. 97–105, Feb. 1970. [Online]. Available: http://dx.doi.org/10.1109/T-C.1970.222874

[32] D. J. Mclaren, "Improved mitchell-based logarithmic multiplier for low-power dsp applications," in *IEEE International [Systems-on-Chip] SOC Conference, 2003. Proceedings.*, Sep. 2003, pp. 53–56. [Online]. Available: https://doi.org/10.1109/SOC.2003.1241461

[33] D. Nandan, J. Kanungo, and A. Mahajan, "An efficient vlsi architecture design for logarithmic multiplication by using the improved operand decomposition," *Integration*, vol. 58, pp. 134–141, 2017. [Online]. Available: https://doi.org/10.1016/j.vlsi.2017.02.003

[34] Durgesh Nandan and Kanungo, Jitendra and Mahajan, Anurag, "An error-efficient gaussian filter for image processing by using the expanded operand decomposition logarithm multiplication," *Journal of Ambient Intelligence and Humanized Computing*, Jul 2018. [Online]. Available: https://doi.org/10.1007/s12652-018-0933-x

[35] D. Nandan, J. Kanungo, and A. Mahajan, "An efficient vlsi architecture for iterative logarithmic multiplier," in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb 2017, pp. 419–423. [Online]. Available: https://doi.org/10.1109/SPIN.2017.8049986

[36] D. Nandan, J. Kanungo, and A. Mahajan, "An efficient architecture of iterative logarithm multiplier," *International Journal of Engineering & Technology*, vol. 7, no. 2.16, p. 24, 2018.