AN EFFICIENT ARCHITECTURE OF LEADING ONE DETECTOR

Article *in* International Journal of Pure and Applied Mathematics · February 2018

CITATION
1

READS
3,634

1 author:

Durgesh Nandan
SR University
131 PUBLICATIONS **816** CITATIONS

SEE PROFILE

International Journal of Pure and Applied Mathematics
Volume 118 No. 14 2018, 267-272
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

AN EFFICIENT ARCHITECTURE OF LEADING ONE DETECTOR

Durgesh Nandan[1], Jitendra Kanungo[2], Anurag Mahajan[3]
[1,2]Department of Electronics and Communication Engineering, J.U.E.T, Guna, India
[3]Department of Electronics and Telecommunication Engineering, S.I.T, Pune, India

Abstract: Design of the Leading-One Detector (LOD) is important as they are used for the normalization process in a floating point multiplication, logarithmic multiplication and logarithmic converters for Digital Signal Processing (DSP) applications. The reported LOD design are found to be slower or hardware inefficient. No operations like shifting-and-counting method, bit-by-bit serial evaluation circuits etc. make it possible to design the efficient LOD. Further, an effective technique is required to handle the problem of locating of the leading-one bit with a fast, hardware-efficient and low power LOD circuit. It motivates to explore new approaches. In this paper, the proposed LOD has given a significant gain in terms of area, power and speed. Synthesis results show that the proposed LOD gives 55.02 %, 45.88 %, 33.40 % and 23.16 % less ADP, 62.52 %, 77.78 %, 60.12 % and 59.62 % less energy for 4, 8, 16, and 32 bits architecture respectively in comparison of the reported LOD.

Keywords: Arithmetic circuits, DSP, Leading one detector, Logarithmic converter, Logarithm multiplication.

I. INTRODUCTION

Multiplication and addition are the frequently used components in Digital Signal Processing (DSP) applications. Data analysis shows that an average 40% multiplication and 60% addition operations performed in DSP applications [1]. Especially, Finite Impulse Response (FIR), Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT) techniques need to be designed with an efficient multiplier [2]. But, as it is well- known fact that a multiplier has always been a limiting factor in terms of accuracy, speed and area.
Design of Leading-One Detector (LOD) is important as they are used for the normalization process in a floating point multiplication, logarithmic multiplication, and in logarithmic converter [3-5]. The LOD is used in logarithmic converters to find the position of the leading 'one' bit in the integral and the fractional parts a logarithm operation are determined with the help of LOD. An efficient and low power LODs is a demand for logarithmic converter to perform a DSP operation. A LOD is used as a key component for performing the shifting and normalization process in the floating-point multiplication, floating-point addition and also in binary logarithmic converters [4]. Research is going on to evolve various combinatorial circuits in a constrained space with minimum effort [6]. Researchers have continuously working to develop an efficient architecture for LOD [7-8].

The reported LOD design are found to be slower or hardware inefficient. No operations like shifting-and-counting method, bit-by-bit serial evaluation circuits etc. make it possible to design the efficient LOD. Further, an effective technique is required to handle the problem of locating of the leading-one bit with a fast, hardware-efficient, and low power LOD circuit. It motivates to explore new approaches. Further, the implementation of an efficient architecture of iterative logarithm multiplier is also proposed by using the designed LODs.

The rest of chapter is arranged as follows: Section 2 gives an idea about the systematic growth of LOD. Section 3 gives a brief explanation about the available designs of LOD in literature. Section 4 explores the proposed design. Section 4 explores the results. Finally, the finding of the LOD exploration is concluded in Section 5.

II. LITERATURE REVIEW

In this section, the systematic growth of LOD circuit with the frequently used reversible gates and possible applications of LODs are presented.

2.1 Growth of LODs

Researchers found that LODs are useful components in the area of floating-point normalization, binary logarithms, logarithmic multiplication, logarithm converter and the circuits were the need to find out the position of the leading one bit [9-14].

In logarithm converters, LODs is the first step of the calculation. It is used to find the mantissa or the fractional part of a logarithm. Initially to generated

binary logarithms shifting and counting method is used but it is slow and takes much more clock cycles. In 1992,1993 and 1997 Oklobdzija has reported a short detail of leading-zero detectors (LZDs) which is equal to the leading-one position but does not generate a decoded binary word [7] [15-16]. It is found that these were not suitable for binary logarithms applications because they were hardware inefficient, slow and given an error of one bit.

In 1998, Lee and Sartori have proposed the modular LOD to find out leading one for binary logarithm but it has a complex circuit to produce have integer value [8]. In 1999, SanGregory *et al. [*17] have used a bit-by-bit serial evaluating circuits to compute the binary logarithm. However, this technique fails to help for small values of *N*, such as *N* equal to 1 because it activates the critical path in these serial circuits, and it limits the maximum operating frequency. In 2001, Schmookler and Nowak [18] have presented a review report on the leading zero anticipator, leading one anticipator and leading one predictor that are key components of floating-point arithmetic and Floating Point (FLP) normalization. In 2003, Abed and Sifred have proposed a new design of LODs which was faster and hardware efficient [12]. They have designed fast serial and parallel LOD circuit approach, which was based on the dividing of input binary word into groups. It effectively works on the leading-one for small as well as large binary words with high speed of circuit execution. In again 2006, Abed and Sifred have provided two new methods to design LODs: (a) For high speed and (b) for low power and hardware efficiency [4]. First design has solved the problem of locating the leading one position in binary words with maintaining high speed and the second approach has less hardware and power in comparison to the first design. In 2013, Kunaraj and Seshasayanan have proposed [5] the novel design of LODs with efficient hardware and high speed in comparison of the reported architecture. They have proposed a gate level efficient architecture of LODs. But, there has been a need to explore new approaches to perform better than of the reported existing approaches.
The main motive of this paper is to design an efficient architecture of LOD. So, at first try to understand what is LOD, how it work and its architecture. Detail about LOD is briefly describes in next section.

III. LEADING ONE DETECTOR

High performance and speed are the main concerns to design the LOD. It is used for the normalization process in floating point multiplication, logarithmic multiplication and logarithmic converters as a useful component [9-14]. LOD is used in logarithmic converters to find the integer portion and the fractional part of the binary logarithm. The 4-bit and 16-bit LOD circuits proposed by Kunaraj & Seshasayanan in 2013 are shown in Figure 1 [5]. These are found to be recent designs. For proper understanding of reported LOD design quality and limitations, it needs to discuss the reported architecture. 4-bit LOD input evaluates from the LSB ‘d0’ to the MSB ‘d3’. Suppose it has a taken “1001” as an input for 4-bit LOD then found output a “0001” and ‘1’ as a zero flag in place of “1000” for a given architecture by Kunaraj and Seshasayanan in 2013. So, it has been shown that given output is wrong. 16-bit LOD can be made by using 4-bit LOD. 16-bit LOD input evaluates from the LSB ‘d0’ to the MSB ‘d16’. Suppose a16-bit LOD input “1000000000000001” then output is found “0000000000000001” and ‘0’ as a zero flag in place “1000000000000000” for given architecture by Kunaraj and Seshasayanan in 2013. Again it gives wrong output. In Figure 1, if an input “i3i2i1i0” consider as an input “i0i1i2i3” and a output “o3o2o1o0” as a output “o0o1o2o3” for 4-bit LOD, then for input “1001” found output “1000” and ‘0’ as a zero flag. That is an expected output. This circuit is based on the serial and parallel LOD approach. For detailed analysis of the reported LOD circuits designed refer by Kunaraj & Seshasayanan [5]. It has been found there is a scope to design the hardware efficient and faster LOD. So, it is challenging to design a low power, hardware efficient and high-speed LOD.

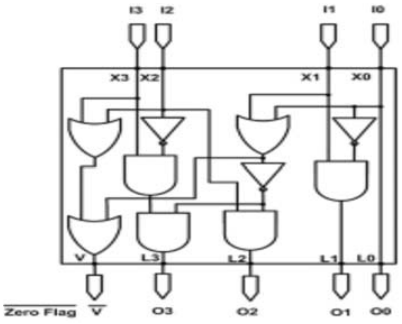

Figure 1: 4 bit LOD circuit [5]

IV. PROPOSED DESIGN

The proposed Iterative Logarithm Multiplier (ILM) consists of logarithm converter, anti- logarithm converter and the arithmetic circuit. As it is known that, LOD is the one of a component of logarithm converter. In this section, the design of LOD, reversible logic

implementation of LOD and an efficient architecture of ILM are discussed.

4.1 Proposed design of LOD

The proposed LOD design approach is based on the LOD circuit given by Abed and Sifred [4]. Some changes are carried out to make it hardware efficient such as: Replacement of 2*1 multiplexer with 2-input 'AND' gate and serially data process for an evaluation of output in place of parallel and serial data processes. Here, parallel data processing is not used for higher bit LODs. It saves flag bit overhead but there is penalty of considerable cost of worst-case delay. Here, design of 4 bits LOD, 8 bits LOD, 16 bits LOD and 32 bits LOD are given.

(a) Proposed 4 bits LOD

The proposed design of 4-bit LOD circuit is shown in Figure 2. It has serial connection that evaluates in arrange from the MSB 'd3' to the LSB 'd0' as compared to the reported LSB 'd0' to the MSB d3. The proposed architecture of the 4-bit LOD is simple in comparison as well as with less hardware requirement. It is easily observed that the proposed design is efficient in comparison of the reported design [4-5].

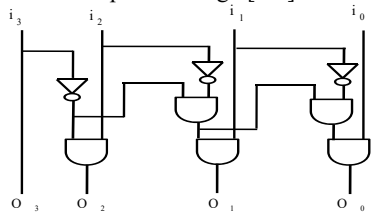

Figure 2: The proposed 4 bit LOD circuit

(b) Proposed 8, 16 and 32 bits LOD

The proposed 8, 16 and 32 bits LOD having a serial connection that evaluates in arrange from the MSB 'dn-1' to the LSB 'd0' in comparison of reported LSB d0 to the MSB dn-1. Each circuit is an extension of the proposed 4-bit LOD. A design of 32-bit LOD circuit is shown in Figure 3.

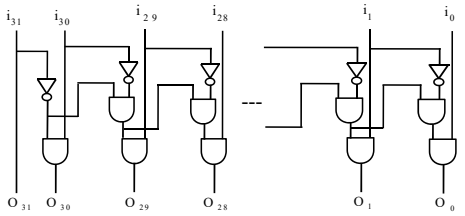

Figure 3: The proposed 32 bit LOD circuit

V. RESULTS

The proposed LOD design has *N* bit input samples and *N* bit output samples. Where, '*N*' is a positive integer number. It has involved a simple 'AND' logic gate and 'NOT' logic gate. The detailed analysis of hardware complexity, error analysis of implemented logic and synthesis results are discussed in next Sub-sections.

5.1 Theoretical hardware analysis

The theoretical analysis of hardware complexity in terms of the basic gates and D-Flip Flops (D-FF) for the proposed structures and reported structures of 4-bits, 8-bits, 16-bits and 32-bits LODs are listed in Table 1. It reduces the two AND gates and two OR gates for 4-bit architecture, 38 less AND gates and 41 less OR gates, 16 less NOT gates and 32 less D-FFs for 16-bit architecture and 95 less AND gates and 94 less OR gate, 41 less NOT gate and 64 less D-FF for 32-bit architecture in comparison of the reported LOD given by Abed and Siferd in 2006 [4]. Here, one thing is noticeable that in the reported LOD, there is no architecture for 8-bit LOD is found. So, no comparison is made for it. Further, in comparison of reported LOD given by Kunaraj & Seshasayanan 1 more AND gates and 3 less OR gate are required for 4-bit architecture, 3 less AND gates, 7 less OR gate and 1 less NOT gate are required for 8-bit architecture, 7 less AND gates, 15 less OR gate and 1 less NOT gate are required for 16-bit architecture and 12 less AND gates, 30 less OR gate and 1 less NOT gate are required for 32-bit architecture with the proposed architecture.

5.2 Synthesis results

The reported and the proposed LOD architectures for 4, 8, 16, and 32-bits by using at 90 nm CMOS technology node have also been synthesized. The net list is generated by the 'Synopsys Design Compiler'. The area, power, and timing constraints are compared with the reported LOD as listed in Table 2. Area-Delay Product (ADP) and Energy for the proposed and the reported structures are listed in Table 2.

The proposed LOD gives 55.02 %, 45.88 %, 33.40 % and 23.16 % less ADP, 62.52 %, 77.78 %, 60.12 % and 59.62 % less energy for 4, 8, 16, and 32-bits architecture respectively than of the reported LOD given by Kunaraj & Seshasayanan [5].

VI. CONCLUSION

In this paper, LOD architecture has been proposed. It has been observed that an important gain in terms of area, power and delay is also obtained. Synthesis results show that the proposed LOD gives 55.02 %, 45.88 %, 33.40 % and 23.16 % less ADP, 62.52 %, 77.78 %, 60.12 % and 59.62 % less energy for 4, 8, 16 and 32 bits architecture respectively in comparison of the reported LOD [5]. LOD Design and its importance of logarithmic multiplication are formed to be useful components in DSP applications.

Table 1: Hardware complexities for LOD

Structure	Reported [4]			Reported [5]				Proposed			
	4 bit	16 bit	32 bit	4 bit	8 bit	16 bit	32 bit	4 bit	8 bit	16 bit	32 bit
AND	7	67	156	4	16	36	73	5	13	29	61
OR	2	41	94	3	7	15	30	---	---	---	---
NOT	3	31	72	3	8	16	32	3	7	15	31
D-FF	---	32	64	---	---	---	---	---	---	---	---

Table 2: Synthesis results for the proposed LOD architecture and the reported LOD architectures.

Structure	DAT (ns)	Area (μm^2)	Power (μW)	ADP ($\mu m^2 * \mu s$)	% gain in ADP	Energy (n J)	% gain in EPS
Reported [4]							
4-bit	3.75	86.71	4.973	0.325	--------	18.650	--------
16-bit	5.11	2129.82	84.751	10.883	--------	433.077	--------
32-bit	5.37	4270.06	101.391	22.930	--------	544.469	--------
Reported [5]							
4-bit	3.04	121.41	4.285	0.369	--------	13.026	--------
8-bit	3.85	287.54	10.249	1.107	--------	39.458	--------
16-bit	4.56	677.56	19.809	3.089	--------	90.329	--------
32-bit	5.20	1494.62	37.193	7.772	--------	193.403	--------
Proposed							
4-bit	2.95	56.24	1.655	0.166	55.02	4.882	62.52
8-bit	3.40	176.45	2.578	0.599	45.88	8.7652	77.78
16-bit	4.30	478.54	8.376	2.057	33.40	36.016	60.12
32-bit	4.90	1218.96	15.937	5.972	23.16	78.091	59.62

References

[1] F. Pappalardo, G. Visalli, and M. Scarana, "An application-oriented analysis of power/precision tradeoff in fixed and floating-point arithmetic units for VLSI processors," in Proc. IASTED Conf. Circuits, Signals, and Systems, pp. 416 – 421, 2004.

[2] 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, pp. 2196–2209, 2013.

[3] H. Suzuki, H. Morinaka, and H. Makino, Y. Nakase, K. Mashiko, T. Sumi, "Leading-Zero Anticipatory Logic for High Speed Floating Point Addition," IEEE J. Solid-State Circuits, vol. 31, no. 8, pp. 1157-1164, 1996.

[4] K.H. Abed, and R.E. Sifred, "VLSI Implementations of Low-Power Leading-One Detector Circuits," Proceedings of the IEEE Southeast Conference, pp. 279-284, 2006.

[5] K. Kunaraj, and R. Seshasayanan, "Leading one detectors and leading one position detectors-an evolutionary design methodology", Canadian journal of electrical and computer engineering, vol. 36, no.3, pp. 103-110, 2013.

[6] V.K. Vassilev, D. Job, and J.F. Miller, "Towards the automatic design of more efficient digital circuits," Proceedings in the Second NASA/DoD Workshop on Evolvable Hardware, pp.151-160, 2000.

[7] V.G. Oklobdzija, "An implementation algorithm and design of a novel leading zero detector circuit," The Twenty-Sixth Asilomar Conference on Signals, Systems and Computers, pp.391-395, 1992.

[8] P. Lee, and A. Sartori, "Modular leading one detector for logarithmic encoder," Electronics Letters, vol.34, pp. 727, 1998.

[9] P. Akhter, S. Bandewar, and Durgesh Nandan, "Logarithmic Multiplier: An Analytical Review," International Journal of Engineering Research, vol. 5, pp: 721-723, 2016.

[10] Durgesh Nandan, J. Kanungo, and A. Mahajan, "An efficient VLSI architecture design for logarithmic multiplication by using the improved operand decomposition," Integration, the VLSI Journal, vol. 58, pp. 134–141, 2017.

[11] Durgesh Nandan, J. Kanungo, and A. Mahajan, "An efficient VLSI architecture for Iterative Logarithmic Multiplier," IEEE 4th International conference on signal processing and integrated networks (SPIN), pp. 419-423, 2017.

[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.1421-1433, 2003.

[13] Durgesh Nandan, J. Kanungo, and A. Mahajan, "An Efficient VLSI architecture design for antilogarithmic converter by using the error correction scheme," IETE International conference on signal processing (ICSP), 2016.

[14] Durgesh Nandan, A. Mahajan, and J. Kanungo, "An efficient antilogarithmic converter by using 11-regions error correction scheme", IEEE 4th International signal processing, Computing and Control (ISPCC), 2017.

[15] V.G. Oklobdzija, "Algorithmic design of a hierarchical and modular leading zero detector circuit," Electronics Letters, vol.29, no.3, 1993.

[16] V.G. Oklobdzija, "Comments on Leading-Zero Anticipatory Logic for high Speed Floating Point Addition," *IEEE J. Solid-State Circuits*, pp. 292-293, 1997.

[17] S.L. SanGregory, R.E. Siferd, C. Brother, and D. Gallagher, "Low-Power Logarithm Approximation with CMOS VLSI Implementation," Proc. IEEE Midwest Symp. Circuits and Systems, pp. 381-391, 1999.

[18] M.S. Schmookler, K.J. Nowka, "Leading zero anticipation and detection-a comparison of methods," Proceedings in 15th IEEE Symposium on Computer Arithmetic, pp.7-12, 2001.