RETRACTED CHAPTER: Optimized Lower Part Constant-OR Adder for Multimedia Applications

Mahendra Vucha and A. L. Siridhara

Abstract Power consumption and speed of computing systems depend on their arithmetic modules such as adder, subtractor, and multiplier. So, the need for high speed, error tolerance, and power efficiency nature of few applications has been improved by developing approximate adders. Increasing the effectiveness of integrated circuits by making the trade-off between accuracy and cost has got significant importance. A systematic methodology for optimizing the architecture of approximate adders has been proposed and called optimized lower part constant-OR adder (LOCA). In this article, the approximate adders are designed by redesigning its logic circuit, implemented on reconfigurable architectures, and then compared with traditional adder architectures. The proposed architecture outperforms its contemporary architectures in terms of hardware and accuracy.

Keywords Approximation · Stochastic computing · Error metrics · Hardware trade-off

1 Introduction

VLSI systems rely on three major parameters, namely power consumption, delay, and space occupied (area). All these parameters must be optimized and kept controlled while designing the system. Computing architectures may face problems if these parameters are not maintained properly. In general, optimizing all these three parameters is not applicable for every system architecture, but the designers could balance them based on application requirements. For example, the design of ATMs strictly adheres to response to the inputs and transaction speed, where optimization required to the delay compared to power and area. For efficient computations, designers need

The original version of this chapter was retracted: The retraction note to this chapter is available at https://doi.org/10.1007/978-981-16-6605-6_67

M. Vucha (✉) · A. L. Siridhara
Department of Electrnonics and Communication Engineering, MLR Institute of Technology, Hyderabad, India

P. Karrupusamy et al. (eds.), *Sustainable Communication Networks and Application*, Lecture Notes on Data Engineering and Communications Technologies 93,
https://doi.org/10.1007/978-981-16-6605-6_19

to minimize the complexity while optimizing the size of a system. Adder is one of the main components of arithmetic circuits, and analysts in the domain of approximate computing have paid attention to adders. There are two methods to approach adders called stochastic computing and approximate computing. Stochastic computing uses binary bitstream where the value of bitstream referred to as a stochastic number (SN) is encoded as 0s and 1s. The major disadvantage of stochastic computing is that it assumes bitstreams are independent, but this assumption does not hold if it fails. Approximate computing is a low-power means for digital signal processing applications and brings a trade-off between performance and accuracy. Approximate techniques may have some errors where no individual errors are recognized but only average errors can systematically predict the impact of error in the output. On the past decade, approximate computing is chosen for adders both at the software level and hardware level. Adders have significant importance in digital operations and signal processing. The approximation adders are the segmented adders, where the *m*-bit adder can partition into *k*-bit sub-adders. Carry select adders—where the multiple sub adders used, carry look-ahead adder, equal segmentation adder, Exact adder approximate full adders—where full adder is approximated. The design of all these approximate adders is to limit the carry generated by the adders. The length of carrying propagation in an *N*-bit conventional adder is similar to Log 2 *N*.

The (lower part OR adder (LOA) presented in literature [1] is shown in Fig. 1, and it is called OR adder, because it is observed that the adders are divided into sub-adders which are said to be *m*-bit one-half adders and the remaining part has OR gates [2–5]. So, the half adder is named as a higher sub-adder which consists of an (n_h–1)-bit-exact adder, and the 'n_l' of OR gates is represented as lower part sub-adder which consists of (0–n_l–1) bits [6–9]. A carry signal for an accurate adder is generated using an extra AND gate [10–14]. Since the approximation is restricted to the least significant bits, the magnitude of errors is limited [15–17]. This is the major advantage of LOA when compared with other architectures such as equal segmentation adder (ESA) [18, 19].

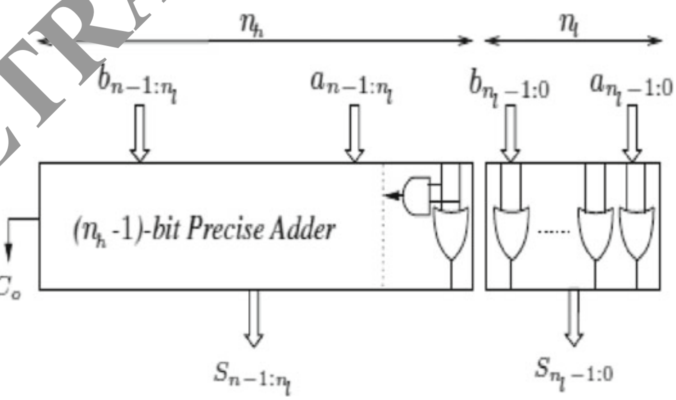


Fig. 1 Existing structure of LOA

The presented LOA is the slowest, but it is highly efficient at its computations. In this article, a method is presented to improve the LOA systematically by considering architectural templates from [1] and then implement all possible combinations to study its efficiency and propose an optimized lower part constant-OR adder (OLOCA) as reduced hardware architecture**.**

2 Optimized Lower Part Constant-OR Adder Architecture

The proposed optimal architecture can be obtained through the incorporation of the following sequential steps.

Step 1: Analyze the error metrics to value hardware quantifying the standard architecture.

Step 2: Consider LOA as a hardware template where the number of OR gates depends on the number of inputs.

Step 3: Implement mean square error (MSE) which is very minimum for OLOCA compared with any other error metric.

2.1 Error Metrics

Since an approximation technique has been adopted for this architecture, this approximation may generate errors in the output of a system which is not desired. In order to reduce the error, error metrics are preferred, and they play a major role in evaluation of different architecture in different fields. The quality of the approximate adders can be evaluated using these error metrics and shows the balance between error and cost of hardware. Error magnitude can be quantified with several metrics. Some of the error metrics are average error (μ), standard deviation (σ), mean square error (MSE), mean absolute error (MAE), root mean square (RMS), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (SMAPE). Error is observed as difference between approximate outcome and actual outcome, that is $\varepsilon = \tilde{S} - S$, where $\tilde{s}$ = Approximate outcome and S = Actual outcome. Error metrics can be calculated using the formulas below.

$$\text{Average Error}(\mu) = E[\varepsilon]$$

$$\text{Standard Deviation}(\sigma) = \sqrt{E\left[(\varepsilon - \mu)^2\right]}$$

$$\text{Mean Square Error(MSE)} = E\left[\varepsilon^2\right] = \mu^2 + \sigma^2$$

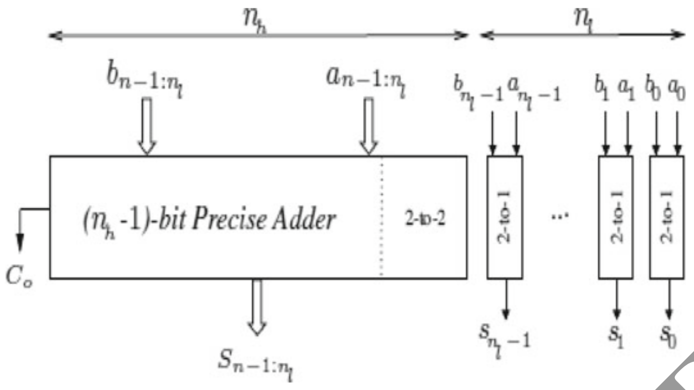


Fig. 2 Internal hardware of LOA

$$\text{Mean Absolute Error(MAE)} = E[|\epsilon|],$$

where E is the expectation operator.

2.2 Architecture

The template architecture is considered from literature and shown in Fig. 2. The architecture template should figure out the efficiency of hardware architecture and delay, where 'A' and 'D' denote the area and delay of the architecture, respectively. In the approximation technique, many samples have been analyzed to consider the best one. Using the unit gate model, not only OR gates, more gates like AND, OR, and NAND are placed, and the combinations of their value are also noted to state that non-similar two input gates XOR and XNOR have more area and delay.

As literature and discussions state that error versus hardware cost trade-off is very efficient in LOA and found to be the best architecture among the existing approximate adders. Evaluation of the general template of LOA allows division of sub-adder into 'n_l' 2-to-1 logic blocks as shown in Fig. 2 and single 2-to-2 logic block which generates the carry for the accurate adder using AND gate by receiving the inputs of the adder at bit position n_l. The higher sub-adder is an accurate adder (exact adder), where XOR results in sum, and AND results in carry. The error metrics and unit gate characteristics of 2-to-1 blocks and 2-to-2 blocks of the architecture are stated in Tables1 and 2, respectively.

From Table 1, it is clear that consideration of MSE error metric and replacing OR gates in 2-to-1 blocks and OR-AND in the first bit of higher sub-adder is the best selection. MSE has strictly positive values (non-negative).

Table 1 Error metrics and unit gate characteristics of 2-to-1 blocks

	μ	σ^2	MSE	A	D
AND	–3/4	3/16	3/4	1	1
OR	–1/4	3/16	1/4	1	1
Buffer	–1/2	1/4	1/2	0	0
Cte-0	–1	1/2	3/2	0	0
Cte-1	0	1/2	1/2	0	0

Table 2 Error metrics and unit gate characteristics of 2-to-2 blocks

	μ	σ^2	MSE	A	D
Half adder	0	0	0	3	2(1)
OR_AND	¼	3/16	1/4	2	1(1)
Cte-1_AND	½	1/4	1/2	1	0(1)
Buffer_AND	0	1/2	1/2	1	0(1)

Table 2 represents all possible combinations of 2-to-2 blocks in the higher sub-adder block which eliminate maximum error values. In this case, half adder is used while it is having standard deviation value as zero, average error and MSE as zero, area as 3, and delay 2.

The data which is distributed parallelly, every bit is uncorrelated along with this error metrics, is to be analyzed and measured as a function of error metrics of each block. The block which contains better parameters can be chosen as optimal architecture. So, the overall error is concluded as the combination of error of each block with corresponding weight,

$$\varepsilon_T = \sum_{i=0}^{n_l} \varepsilon_i 2^i$$

$$\mu_T = \sum_{0}^{n_l} \mu_i 2^i$$

$$\sigma_T^2 = \sum_{0}^{n_l} \sigma_i^2 2^{2i}$$

$$\text{MSE}_T = \sum_{0}^{n_l} \sigma_i^2 2^{2i} + \left(\sum_{0}^{n_l} \mu_i 2^i \right)^2$$

where μ_i and σ_i^2 are average error and variance of error associated with block in bit position i.

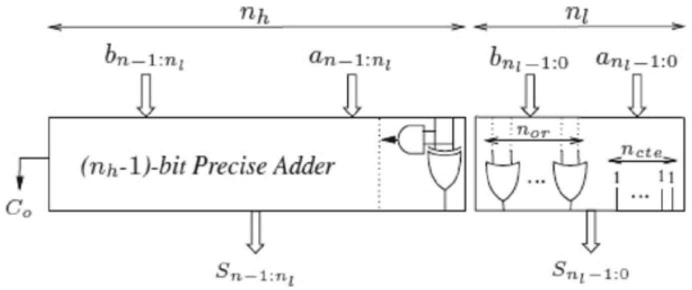


Fig. 3 Proposed system of LOCA: $n_l = n_{cte} + n_{or}$

3 Optimized LOCA

The LOCA can have various optimization methods based on the metrics values shown in Tables 1 and 2. In data processing and image processing applications, MSE is considered as one of the important error metrics because it measures the average of the errors that is the average difference between the approximate results to the accurate result. So, the proposed optimized architecture based on the mean square error metric surely brings the optimum and errorless computations in real-time applications. The various combinations of optimized lower part constant-OR adder architecture can be evaluated by including both lower sub-adder block and higher sub-adder block.

The upper bits (higher sub-adder) produce a high error rate as compared to lower bits (lower sub-adder). So, this article has concentrated on higher sub-adder rather than lower sub-adder. As seen in Table 2, the best higher sub-adder is OR_AND and half adder. Although it does not improve the delay by replacing the OR_AND with half adder, it improves the area. By fixing the higher sub-adder to a half adder, it is observed that the average error is considered as zero or positive, coming to the lower sub-adder block having a zero or negative average error. So, a higher sub-adder block is used with small μ (C-1) or small (OR). Therefore, the optimal architecture of the lower sub-adder consists of OR gates followed by 1's blocks in the lower bits where hardware complexity of structural design is reduced while optimizing area and delay and hence called OLOCA (Fig. 3).

The proposed LOCA architecture is verified with an optimal number of OR gates and results in optimal value at nor = log2 (8/3). The integer numbers nor = 1 and nor = 2 produce the same MSE, but nor = 2 gives a better STD, and hence, this architecture is named OLOCA. The various error formulas in terms of architecture parameters are as shown (Table 3).

4 Result and Discussion

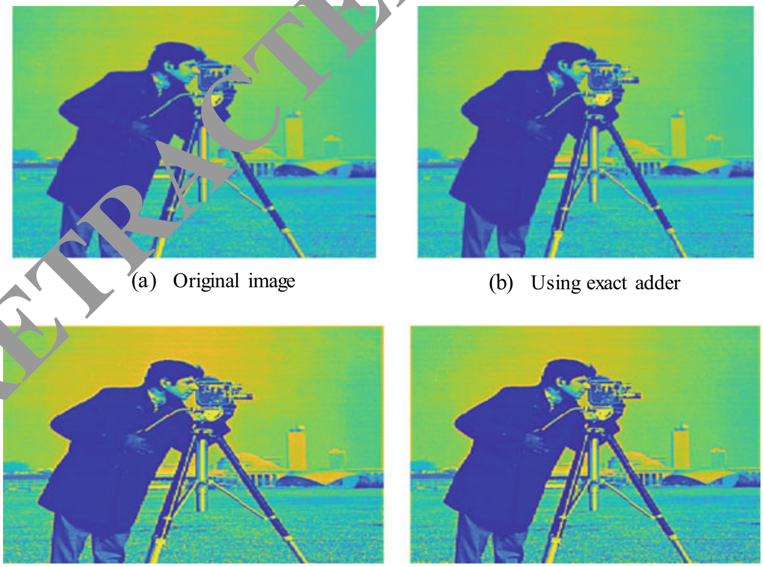
The LOCA found very significant image processing applications to improve the sharpness of an image. Structural similarity of an image can be the quality metric

Table 3 Formulas of error metrics, area, and delay

Parameter	LOA	OLOCA
μ	1/4	$-3/16\ 2^{n}{}_{1}$
σ^{2}	$1/4^{n}{}_{1} - {}_{1/16}$	$53/7684^{n}{}_{1}$
MSE	$1/4\ 4^{n}\ {}_{1}$	$5/48\ 4^{n}{}_{1} - 1/6$
MAE	$3/8\ 2^{n}{}_{1} - 3/8$	$15/64\ 2^{n}{}_{1} - 3/42^{-n}{}_{1}$
A	$(n_h - 1).\ A_{\mathrm{FA}} + A_{\mathrm{AND}} + (n_l + 1).\ \mathrm{A}_{\mathrm{OR}}$	$(n_h - 1).\ A_{\mathrm{FA}} + A_{\mathrm{HA}} + (n_1 - n_{\mathrm{cte}}).\ A_{\mathrm{OR}}$
D	$(n_h - 1).t_c + T_{\mathrm{AND}}$	$(n_h - 1).t_c + T_{\mathrm{AND}}$

used to measure the similarity between two images. Multimedia applications like animation programs where pixels are added in a picture can utilize the OLOCA addition operator and reduce the error rate. JPEG compression is used for saving storage space and transmission bandwidth for digital images. Reducing the data correlation by converting it from the time domain to the frequency domain is the strategy behind JPEG compression. The human eye is less sensitive to high frequencies. The LOCA techniques are implemented on a MATLAB simulation environment, and the results are shown in Fig. 4. The error metrics for JPEG images having 8-bit data size are summarized in Table 4.

The proposed OLOCA and ripple carry adder architectures are also implemented using Verilog HDL with targeted FPGA device xc7a100t-3-csg324, and the design parameters like area and speed of the architectures are tabulated in Table 5. From


(a) Original image

(b) Using exact adder

(c) Image with two OR gates with n_l=8

(d) Image with two OR gates with n_l=7

Fig. 4 Simulated images processed using OLOCA

Table 4 Simulation results of 8-bit

Error technique		$n_l = 2$	$n_l = 3$	$n_l = 4$	$n_l = 5$
MAE	LOA	1.38 s	2.88	5.87	11.87
	OLOCA	0.75	1.78	3.70	7.48
MSE	LOA	4.00	16.00	63.93	255.90
	OLOCA	1.50	6.53	26.50	106.57
STD	LOA	1.99	3.99	7.99	16.00
	OLOCA	0.97	2.06	4.18	8.40
ADP	LOA	26.82	19.19	13.19	7.95
	OLOCA	27.00	18.89	12.24	6.74

Table 5 Design parameters of 8-bit adder architectures

Parameter	RCA	OLOCA
Number of XOR gates	16	7
Delay	2.523 ns	0.761 ns
Speed	0.4 GHz	1.3 GHz

Table 5, it has been proved that the proposed architecture has presented optimum performance both in terms of area (number of gates) and speed of computations.

5 Conclusion

The hardware architectures and their performance of adders have got significant importance in most of the computing architectures. In this article, an architecture called optimized lower part constant-Or adder architecture has improved the computing speed of addition operations compared to traditional adder architectures. The proposed architecture would balance the cost of hardware and accuracy while reducing the hardware complexity of existing architecture and proved that the proposed architecture showed significant optimization in both area and speed.

References

1. D. Ayad, N. Ardalan, G.O. Alberto, Systematic design of an approximate adder: the optimized lower-part constant-OR adder (2018)
2. J. Satori, R. Kumar, Stochastic computing. Found. Trends Electron. Design Autom. **5**(3), 153–210 (2011)
3. S. Mittal, A survey of techniques for approximate computing. ACM Comput. Surv. **48**(4), 62-1–62-33 (2016)

4. A.B. Kahng, S. Kang, Accuracy-configurable adder for approximate arithmetic designs, in Proceedings of the 49th Annual Design Automation Conference (DAC) (2012), pp. 820–825
5. D. Mohapatra, V.K. Chippa, A. Raghunathan, K. Roy, Design of voltage-scalable meta-functions for approximate computing, in Proceedings of the Design, Automation and Test in Europe (2011), pp. 1–6
6. S. Ajmera, M. Vucha, A. Kokkula, High speed architecture for orthogonal code convolution**,** in Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS, 2017)
7. N. Zhu, W.L. Goh, K.S. Yeo, An enhanced low-power high-speed adder for error tolerant application, in Proceedings of the 12th International Symposium Integerates Circuits (2009), pp. 69–72
8. M. Vucha, A.L. Siridhara, High speed cryptography architecture for health ınformation exchange. Int. J. Adv. Trends Comput. Sci. Eng. **8**(4), (2019)
9. T. Kalyani, S. Monika, B. Naresh, M. Vucha, Accident detection and alert system. Int. J. Innov. Technol. Exploring Eng. (IJITEE) **8**(4S2), (2019). ISSN: 2278–3075
10. S.-L. Lu, Speeding up processing approximate circuits, Computer **37**(3), L 67–73 (2004)
11. T. Anuradha, K.A. Manjusha, R. Karthik, M. Vucha, A.L. Siridhara, Design and ımplementation of an audio parser and player. J. Eng. Appl. Sci. **12**(20), 5301–5306 (2017)
12. D. Esposito, D. De Caro, A.G.M. Strollo, variable latency speculative parallel prefix adders for unsigned and signed operands. IEEE Trans. Circuits Syst. I, Reg. Papers **63**(8), 1200–1209 (2016)
13. H. Jiang, J. Han, F. Lombardi, A comparative review and evaluation of approximate adders, in Proceedings of the 25th Edition Great Lakes Symp VLSI (GLSVLSI) (2015), pp. 343–348
14. A.L. Siridhara, M. Vucha, T. Ravinder, Performance evaluation of parallel multipliers. J. Eng. Appl. Sci. **12**, 5186–5189 (2017)
15. A. Najafi, M. Weiβbrich, G.P. Vaya, A. Garica-Ortiz, A fair comparison of adders in stochastic regime, in Proceedings of the 27th International Symposium Power Timing Modeling, Optim. Simulation(PATMOS) (2017), pp. 1–6
16. M. Vucha, A. Rajawat, Design and VLSI ımplementation of systolic array architecture for matrix multiplications. Int. J. Comput. Appl. **26**(3), 18–22 (2011)
17. M. Vucha, L.S. Varghese, Design space exploration of DSP techniques for hardware software co-design: an OFDM transmitter case study. Int. J. Comput. Appl. **116**(20), 29–33 (2015)
18. P. Gurjar, R. Solanki, P. Karshivan, M. Vucha, VLSI ımplementation of adders for high speed ALU. Int. J. Comput. Appl. **29**(10), 11–15 (2011)
19. M.C. Sudeep, M. Sharath Bimba, M. Vucha, Design and FPGA implementation of high speed vedic multiplier. Int. J. Comput. Appl. **116**(20), 6–9 (2014)