IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REDUCED AREA AND DELAY UNSIGNED MULTIPLIER USING APPROXIMATE ADDER CELL AND ANALYZE SPEED OF MULTIPLIER AND APPLICATIONS

[1]Haridhar Barinala, [2]T K Kalairasan

1st M.Tech, 2nd Assi Proficer
Electronics and Communication Engineer
Vemu Institute of Technology, Chittoor, India

Abstract: In this paper approximate circuits have been considered for applications that can tolerate some loss of accuracy with improved performance and/or energy efficiency. Multipliers are key arithmetic circuits in many of these applications including digital signal processing (DSP). In this paper, a novel approximate multiplier with a low power consumption and a short critical path is proposed for high-performance DSP applications. This multiplier leverages a newly designed approximate adder that limits its carry propagation to the nearest neighbors for fast partial product accumulation. Different levels of accuracy can be achieved by using proposed approximate adder. In existing method to used two different techniques for the generating the multiplication it used two different blocks for generating the levels of summation is calculated two basic blocks those are OR gates and Approximate Adder. In proposed unsigned multiplier is very fast and gives the better accurate results comparing with existing method. In these adder cell only used less logic gates for comparing with normal full adder that's way to reduced number of logics gates it takes less area and less power consumptions. These types of multiplier are very used for the approximate multiplication applications. Image processing applications, including image sharpening and smoothing, are considered to show the quality of the approximate multipliers in error-tolerant applications. By utilizing an appropriate error recovery scheme, the proposed approximate multipliers achieve similar processing accuracy as exact multipliers, but with significant improvements in power.

Index Terms **-Approximate Adder, Cells, OR gates, DSP, Multiplier.**

I. INTRODUCTION

Approximate Computing (AC) is a wide spectrum of techniques that relax the accuracy of computation in order to improve performance, energy, and/or another metric of merit. AC exploits the fact that several important applications, like machine learning and multimedia processing, do not require precise results to be useful. For instance, [1] we can use a lower resolution image encoder in applications where high-quality images are not necessary. In a data centre, this may lead to large savings in the amount of required processing, storage and communication band width. Nowadays, the majority of our computations are being done either on mobile devices or in large data centres (think of cloud computing). Both platforms are sensitive to power consumption. That is, it would be nice if we can extend the operation time of smartphones and other battery powered devices before the next recharge. We know that applying any lossy compression algorithm (JPEG for example) to a raw image will result in an approximate image. Such compression often comes (by design) with little human perceptible loss in image quality [3]. Also, image encoders usually have tune able algorithmic[7] knobs, like compression level, to trade off image size with its quality. Therefore, strict exactness may not be required and an inaccurate result may be sufficient due to the limitation of human perception. As one of the key components in arithmetic circuits, adders are studied for approximate implementation.

As the carry propagation chain is usually shorter than the width of an adder, the hypothetical adders use a reduced number of less significant input bits to calculate the sum bits. In [15], approximate 4 × 4 and 8 × 8 bit Wallace multipliers are designed by using a carry-in prediction method. Then, they are used in the design of approximate 16 × 16 Wallace multipliers, referred to as AWTM. The AWTM is configured into four different modes by using a different number of approximate 4 × 4 and 8 × 8 multipliers. These approximate multipliers are designed for unsigned operation. Signed multiplication is usually implemented by using a Booth algorithm. Approximate designs have been proposed for fixed width Booth multipliers. The proposed multiplier can be configured into two designs by using OR gates and the proposed approximate adders for error reduction, which can be referred as approximate multipliers M1 and M2 respectively. Different levels of error recovery can also be achieved by using a different number of MSBs for error recovery in both multipliers. As per the analysis, the proposed multipliers have significantly shorter critical paths and lower power dissipation than the traditional Wallace multiplier. [10] Functional and circuit simulations are performed to evaluate the performance of the multipliers. Image sharpening and smoothing are considered as approximate multiplication-based DSP

applications. Experimental results indicate that the proposed approximate multipliers perform well in these error tolerant applications.

The proposed designs can be used as effective library cells for the synthesis of approximate circuits.

II. RELATED WORK

J. Han and M. Orshansky.[1]: Approximate computing has recently emerged as a promising approach to energy-efficient design of digital systems. Approximate computing relies on the ability of many systems and applications to tolerate some loss of quality or optimality in the computed result. By relaxing the need for fully precise or completely deterministic operations, approximate computing techniques allow substantially improved energy efficiency. This paper reviews recent progress in the area, including design of approximate arithmetic blocks, pertinent error and quality measures, and algorithm-level techniques for approximate computing.

A. K. Verma, P. Brisk, and P. Ienne[2]: Adders are one of the key components in arithmetic circuits. Enhancing their performance can significantly improve the qualityof arithmetic designs. This is the reason why the theoretical lower bounds on the delay and area of an adder have been analysed, and circuits with performance close to these bounds have been designed. In this paper, we present a novel adder design that is exponentially faster than traditional adders; however, it produces incorrect results, deterministically, for a very small fraction of input combinations. We have also constructed a reliable version of this adder that can detect and correct mistakes when they occur. This creates the possibility of a variable-latency adder that produces a correct result very fast with extremely high probability; however, in some rare cases when an error is detected, the correction term must be applied and the correct result is produced after some time. Since errors occur with extremely low probability, this new type of adder is significantly faster than state-of-the-art adders when the overall latency is averaged over many additions.

III. EXISTING METHOD

a) The Approximate Adder

The design of a new approximate adder is shown. This adder operates on a set of pre-processed inputs. The input pre-processing (IPP)is based on the commutativity of bits with the same weights in different addends. For example, consider two sets of inputs to a 4-bit adder: i) A = 1010, B = 0101 and ii) A = 1111, B = 0000. Clearly, the additions in i) and ii) produce the same result. In this process, the two input bits Ai Bi = 01 is equal to Ai Bi = 10 (with i being the bit index) due to the commutativity of the corresponding bits in the two operands. The basic rule for the IPP is to switch Ai and Bi if Ai = 0 and Bi = 1 (for any i), while keeping the other combinations (i.e., Ai Bi = 00, 10 and 11) unchanged. By doing so, more 1's are expected in A and more 0's are expected in B. If A˙i B˙ i are the ith bits in the pre-processed inputs, the IPP functions are given by:

$$A_i = A_i + B_i \qquad (1)$$

$$B_i = A_i B_i \qquad (2)$$

Equations (1) and (2) compute the propagate and generate signals used in a parallel adder such as the carry lookahead adder (CLA). The proposed adder can process data in parallel by reducing the carry propagation chain. Let A and B denote the two input binary operands of an adder, S be the sum result, and E represent the error vector. Ai , Bi , Si and Ei are the ith least significant bits of A, B, S and E, respectively. A carry propagation chain starts at the ith bit when B˙i = 1, A˙i+1 = 1, B˙i+1 = 0. In an accurate adder, Si+1 is 0 and the carry propagates to the higher bit. In the proposed approximate adder, Si+1 is set to 1 and an error signal is generated as Ei+1 = 1. This stops the carry signal from propagating to the higher bits. Hence, a carry signal is produced only by the generate signal, i.e., Ci = 1 only when B˙i = 1, and it only propagates to the next higher bit, i.e., the (i +1)th position. The logical functions are given by

$$S_i = B_{i-1} + B_i A_i, \qquad (3)$$

$$E_i = B_i B_{i-1} A_i . \qquad (4)$$

By replacing A˙i and B˙i using (1) and (2) respectively, the logic functions with respect to the original inputs are given by

$$S_i = (A_i \oplus B_i) + A_{i-1}B_{i-1}, \qquad (5)$$

$$E_i = (A_i \oplus B_i) A_{i-1}B_{i-1}, \qquad (6)$$

Where i is the bit index, i.e., i = 0, 1,⋯ , n for an n-bit adder.

Let A−1 = B−1 = 0 when i is 0, thus, S0 = A0 ⊕ B0 and E0 = 0. Also, Ei = 0 when Ai−1 or Bi−1 is 0. Consider an n-bit adder, the inputs are given by A = An−1 ⋯ A1A0 and B = Bn−1 ⋯ B1B0, the exact sum is S = Sn−1 ⋯S1S0. Then, Si can be computed as Si + Ei and thus, the exact sum of A and B is given by

$$S = S + E. \qquad (7)$$

In (7) '+' means the addition of two binary numbers rather than the 'OR' function. The error E is always nonnegative and the approximate sum is always equal to or smaller than the accurate sum. This is an important feature of this adder because an additional adder can be used to add the error to the approximate sum as a compensation step. While this is intuitive in an adder design, it is a particularly useful feature in a multiplier design as only one additional adder is needed to reduce the error in the final product.

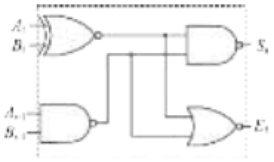

Figure-1: the approximate adder cell.

IV. PROPOSED SYSTEM

The proposed system we are Attaining very fast digital devices with reduced power utilization is an important interest to the VLSI circuit designers and manufacturers. For the most part computation functions are carried out using the multiplier, where it is found to be more power consuming component in the electronic circuits. Eventually, the operation of multiplication has been carried out by the process of shift and add method. Due to the enhancement among various adders, which paved the way for the increase in execution rate of the multipliers. Parallel multiplication algorithms often use combinational circuits and don't contain feedback structures. The circuit is developed utilizing Verilog and functions were validated based on the simulations obtained utilizing Xilinx. In this project, the improvement in approximate unsigned multiplier using the Modified Approximate Full Adder concepts is done.

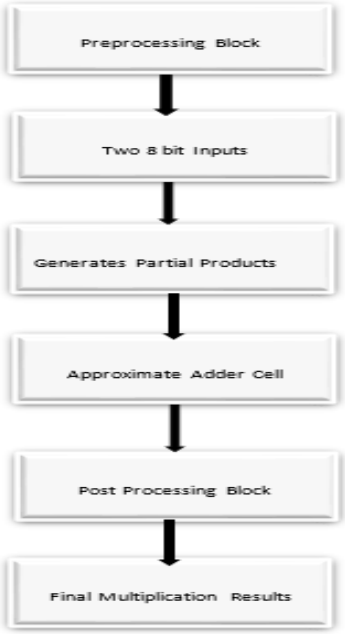

Figure-2: Flow of Proposed Method

Arithmetic and logic unit has been the most significant unit in any electronic devices. In the recent advancement, for an arithmetic and logic unit to be significant it needs to have an efficient algorithmic operation such as Multiplications and addition. Multipliers are a sublime unit in the digital environment. They take part in a critical role in the applications of digital signal processing and image processing. For a Multiplier to be effective and efficient in terms of performance which needs to be high speed or consume less power or to be time effective. Reduction in delay makes the overall computation time decrease. Area and delay are the two major conflicting constraints. Overall optimization must be made for an efficient multiplier.

Multiplication can be done serially or parallel. Theoretically multiplication can be done by repeated addition. Consider multiplication A x B where A is multiplier and B is multiplicand, if we add B to itself A times the sum will be the product of A x B. Practically this process is very slows on ever been used.

This method in valves generating intermediate products and then adding them properly taking into account the weigh to fetch bit while moving from LSB to MSB. The following steps are used in manual calculation:

a) Start with LSB of the multiplier, if it is 1 the multiplicand will be the first partial product. If LSB is 0, put 0 sin the current partial product.

b) Analyse next bit of the multiplier and generate the partial products as per the first rule. The new partial product will be placed one bit left to that of previous partial product. Repeat this step till last bit of the multiplier covered.

c) Ad dell the partial products to get final product. The previous method is slightly modified while implementing in a computer by the hardware using parallel binary adder. The partial product is immediately added after each cycle instead of adding them at the end. The partial product is shifted right by one bit after each cycles that multiply and can be added straight to next cycle.

1) Modified Approximate Full Adder Cell

A modified approximate full adder has been used in the initial stages of the entire architecture. There are exactly 5 number of modified approximate full adders in the design. This is used in the area of conventional full adders to reduce the gate counts using approximation concept.

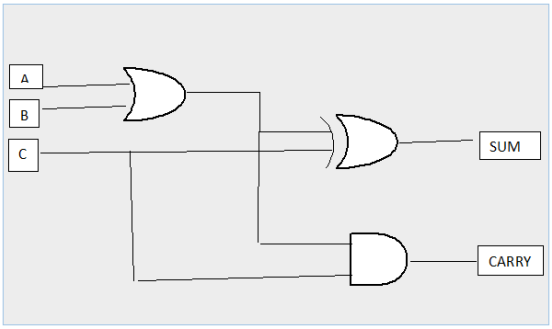

Figure -3: Logic Structure of the Modified 3-bit Approximate Full adder.

TABLE-1: Truth Table for Approximation Full adder

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

The modified approximate full adder uses only one OR gate, one EX-OR gate and one AND gate in total to obtain the desired function of a full adder thereby reducing the number of gate counts and other parameters like power, area. These modified approximate full adders after its functioning produces 5 equivalent sum outputs and 5 equivalent carry outputs.

2) Approximate Adder Cell with Unsigned Multiplier

In these proposed method to implement the approximate multiplier is done based on approximate adder cell for reducing time delay, path delay of number of ports and also observed the area of the multiplier for the unsigned techniques.

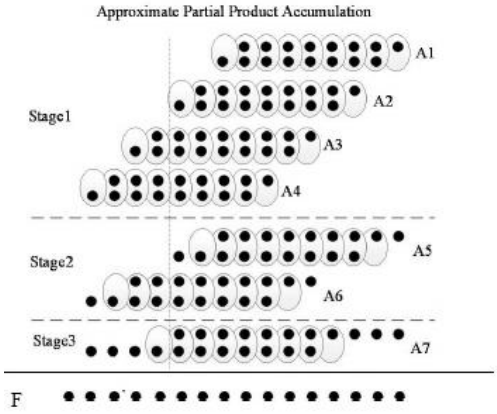

Figure-4: Approximate unsigned multiplier logic representation

When observed the approximate multiplier for the reduced the partial products in four stages each stage we applying the approximate adder cell for the multiplier partial products, in this to generate the sum and carry products in the approximate adder cell sum part is forward to next stage and carry term is transferred for next stage of the partial product terms as shown figure 4 that will be as finally to generate the multiplication results in term of F.

V. RESULTS

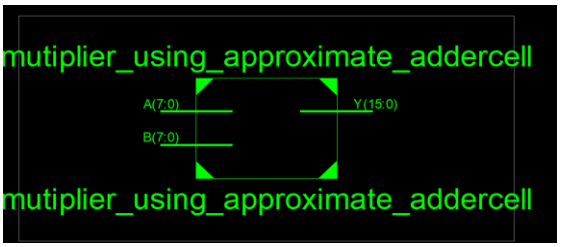

Figure-5: Synthesis Results for Top level RTL for unsigned multiplier

Unsigned multiplier synthesis is shown figure 5 .it contains the Two 8 bit input signals A and B and one multiplier output Y it contains 16 bits length.

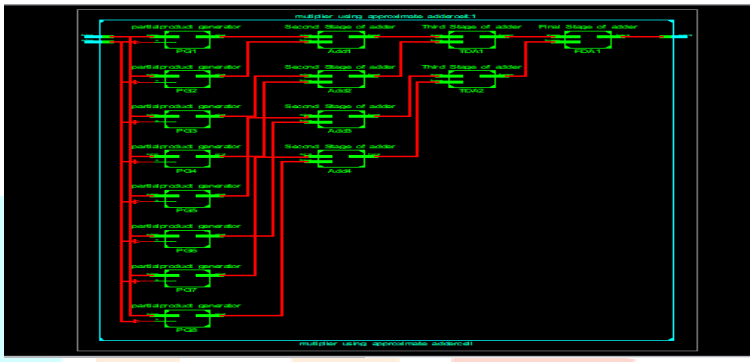

Figure-6: Synthesis Results for RTL Logic Level for unsigned multiplier

The unsigned 8bit multiplier synthesis RTL logic is shown in figure-6 we observed this figure to display all logics of generated partial products and reduction of partial products for three stages. In each stage to observed the logic report.

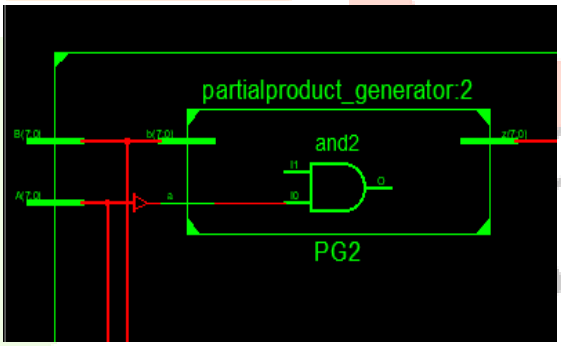

Figure-7: Partial product RTL logic Level.

The figure-8 shows synthesis RTL logic for Partial Product Generator for logic-2.

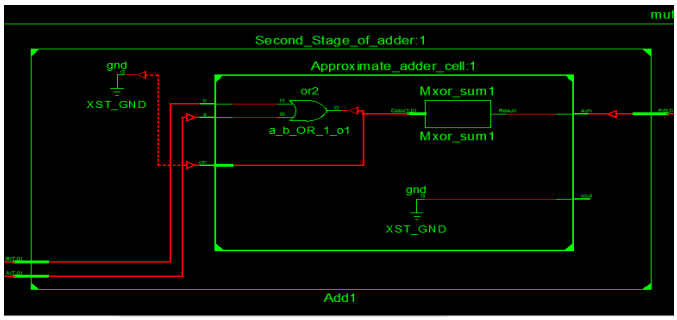

Figure-9: Synthesis RTL logic For Second stage of adder

To observed Second stage of partial product reduction of Synthesis RTL logic in figure-9. In this we see approximate adder cell logic it contains OR logic and XOR logic in shown the above figure.

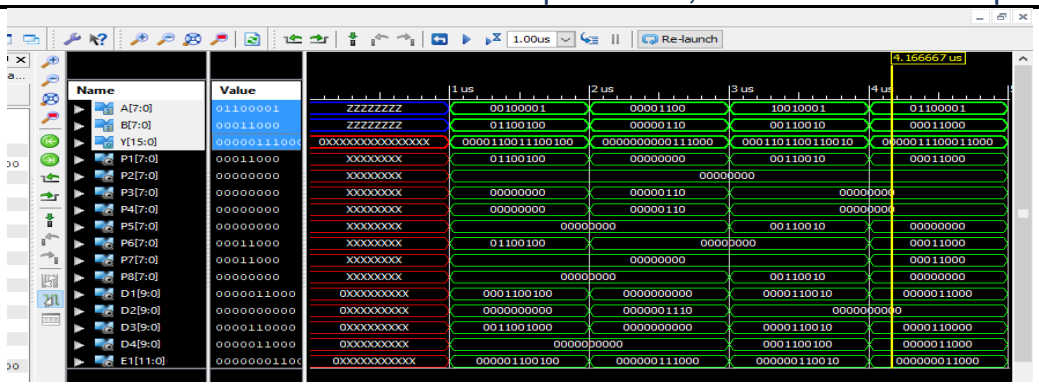

Figure-10: Simulation results for unsigned multiplier

Table-2: comparison table unsigned multipliers

Parameters	Existing Method	Proposed Method
Number of slice LUTs	65	7
Number of Fully used LUT-FF pairs	0	0
Number of Bounded IOBs	32	26
Logic	0.583	0.098
Route Delay(ns)	4.345	0.659
Total number of paths / destination ports	881 / 16	16 / 9

VI. CONCLUSION

This paper proposes a high-performance and low-power approximate partial product accumulation tree for a multiplier using a newly designed approximate adder. The proposed approximate adder cell is implemented using basic logic gates that is XOR, AND and OR gates it will be increase the performance. By using these approximate adder cell in multiplier for partial product reductions apply this logic in multiplier product generation. The proposed approximate multipliers improve over previous approximate designs especially in accuracy. While previous designs focus on reducing both delay and power with often unsatisfying accuracy, the proposed designs achieve excellent delay and power reductions with a high accuracy. The application of the proposed multipliers to image sharpening and smoothing has shown that the proposed designs are very competitive in performance with their accurate counterpart.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emergingparadigm for energy-efficient design," in Proc. 18th IEEE Eur. TestSymp., May 2013, pp. 1–6.

[2] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.

[3] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. Design,Automat. Test Eur., Mar. 2008, pp. 1250–1255.

[4] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in Proc. 12th Int. Symp. Integr.Circuits, Dec. 2009, pp. 69–72.

[5] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation ofsoft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[6] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," inProc. IEEE/ACM Int. Symp. Low Power Electron. Design, Aug. 2011, pp. 409–414.

[7] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proc. Design Automat. Conf., Jun. 2012,pp. 820–825.

[8] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. Design, Automat. TestEur. Conf. Exhib., Mar. 2012, pp. 1257–1262.

[9] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 62,no. 9, pp. 1760–1771, Jun. 2012.

[10] J. Huang, J. Lach, and G. Robins, "A methodology for energy-quality tradeoff using imprecise hardware," in Proc. Design Automat. Conf.,Jun. 2012, pp. 504–509.

[11] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in Proc. IEEE/ACMInt. Conf. Comput.-Aided Design, Nov. 2012, pp. 728–735.

[12] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," inProc. IEEE/ACM Int. Conf. Comput.-Aided Design, Nov. 2011, pp. 667–673.

[13] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits,"ACM J. Emerg. Technol. Comput. Syst., vol. 13, no. 4, 2017, Art. no. 60.

[14] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.

[15] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient approximate wallace tree multiplier for error-resilient systems," in Proc.15th Int. Symp. Qual. Electron. Design, Mar. 2014, pp. 263–269.

[16] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. ElectronDevices Solid-State Circuits, Dec. 2010, pp. 1–4.

[17] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processingand classification applications," IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[18] Y.-H. Chen and T.-Y. Chang, "A high-accuracy adaptive conditionalprobabilityestimator for fixed-width booth multipliers," IEEE Trans.Circuits Syst. I, Reg. Papers, vol. 59, no. 3, pp. 594–603,Mar. 2012.

[19] B. Shao and P. Li, "Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 4, pp. 1081–1090, Apr. 2015.