International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-14 Issue-01 Feb 2025

IMPLEMENTATION OF HIGH SPEED AND AREA EFFICIENT APPROXIMATE MULTIPLIERS

[1]Dr.K.DHANUMJAYA, [2]A SURENDRA REDDY, [3]R.BANUPRIYA

[1]Professor, Dept. of ECE, Audisankara College of Engineering & Technology (Autonomous), Gudur, Tirupati (Dt),AP, India.

[2]Associate professor, Dept. of ECE, Audisankara College of Engineering & Technology (Autonomous), Gudur, Tirupati (Dt),AP, India.

[3]PG Scholar, Dept. of ECE, Audisankara College of Engineering & Technology (Autonomous), Gudur, Tirupati (Dt),AP, India.

Abstract

Multiplication is the basic building block for several DSP processors, Image processing and many other. Over the years the computational complexities of algorithms used in Digital Signal Processors (DSPs) have gradually increased. Approximate computing can decrease the design complexity with an increase in performance and power efficiency for error resilient applications. The partial products of the multiplier are altered to introduce varying probability terms. The architecture of Partial product multiplier mainly consists of some Carry Save Adders, array of AND gates and one Ripple Carry Adder. The proposed approximation is utilized in two variants of 16-bit multipliers. The RTL design of proposed new Partial Product Multiplier and standard partial product multiplier is done using Verilog HDL. The Xilinx ISE design tool is used for FPGA implementation. Comparative result shows the modified design is effective when compared in terms of delay with the standard design.

Keywords: Approximate Carry Adder, Approximate computing, Approximate Multiplier

1. Introduction

Exact and precise models and algorithms are not always appropriate for proficient use in multimedia and image processing operations. The model of approximate calculation relies on entirely relaxing fully exact and completely deterministic building blocks while, designing energy-efficient systems. In digital designs, integer multiplication is one of the fundamental building blocks, which deeply affects the microprocessor and DSP performance. A faster digital circuit is obtained by implementing a speculative (prediction) approach. Speculative digital circuits are based on faster operation by employing a speculative functional unit, which is an arithmetic unit that employs a predictor for the carry signal, without actually waiting for the carry propagation. The speculative unit predicts the carry of the one or more cells used in the digital circuit without waiting for the actual carry propagation to take place. This is similar to a predictor in

the microprocessor. Here we have considered a speculative multiplier which consists of a predictive carry-save reduction tree using three steps: partial products recoding, partial product partitioning and speculative compression. The tree is further comprised of a fast carry-propagate adder and an error rectification circuit. Speculative multipliers have higher speed compared to their conventional counterparts.

In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Research on approximate computing for error tolerant applications is on the rise. Adders and multipliers form the key components in these applications. In approximate full adders are proposed at transistor level and they are utilized in digital signal processing applications. Their proposed employed in fixed-width multiplier designs. Then a constant or variable correction term is added to compensate for the quantization error introduced by the truncated part. Approximation techniques in multipliers focus on accumulation of partial products, which is crucial in terms of power consumption.

2. LITERATURE REVIEW

It Show that approximate circuits have higher performance as compared to precise logic circuits. Many inexact multipliers have been proposed in the literature [4] [6] [7] [8]. These designs employ a truncated multiplication method. In [6], an inexact array multiplier is used, by ignoring selected least significant bits in partial products. A inexact multiplier with correction constant has been proposed in [9]. A variable correction constant inexact multiplier is proposed in [4].This method modifies the correction term according to column n-k-1. If partial products in column n-k-1 are one, then correction factor is increased and, if all partial products in the above column are zero, the correction factor is decreased. In [2], a basic 2x2 multiplier block is suggested for constructing larger multiplier arrays. In all these designs the area was found to be very high. In [10] another approximate multiplier with two approximate [4:2] compressor has been proposed. This multiplier requires lesser area as compared to multipliers using truncation technique however the error percentage was found to be very high.

Approximate multiplier design which utilizes prediction units for the carry signal and also has lesser error percentage as compared to [11]. SFUs (Speculative Functional Units) are prediction circuits that can be considered as black box entities which are faster than their non -speculative counterparts, independently of the particular implementation [8]. Hence approximate multipliers using SFUs also aim to achieve delay improvements, at the same time introducing less power and area overheads. This multiplier utilises Carry Save Adder (CSA) tree [12] for partial product reduction, wherein the carry outputs are propagated rather than being preserved thereby reduces the delay. Popular CSA schemes include Wallace tree and Dadda multiplier.

Wallace tree [1] [9] result in long and irregular wires along the columns to connect to the CSA. The wire capacitance in turn increases the delay and energy of the multiplier and the wires are difficult to layout. Dadda refined Wallace's method by introducing a counter placement strategy that requires few numbers of counters in the reduction stage but at the cost of larger Carry propagate Adder (CPA) [2] [9]. The delay from an input to an output in a full adder is not the same. This delay is dependent on a particular transition (0-to-1, 1-to-0). Therefore it is also possible to come up with different realizations of a full adder wherein a specific signal path is favored with respect to the others and has been designed in such a way that a signal propagation of this path takes a minimal amount of time [3]. The CSA scheme which takes care of this delay in transition is Three Dimensional Scheme (TDM) [3], where partial product array is represented in space and time. This is followed by a speculative adder [5].

Many multipliers that are of high-speed, low power and standard in design are used in multiplication process. Numerous efforts have been used to decrease the number of partial products generated in a multiplication process. One of the efforts is Partial product multiplier. This work aims at designing and implementation of 8-bit Partial product multiplier using VHDL language. A Partial product multiplier is an upgraded version of tree based multiplier architecture. Partial product multiplier use carry saves addition algorithm to decrease the latency. Speed of Partial product multiplier can be improved by using compressor methods. By minimizing the number of partial products we used half adder and full adder in 8-bit multiplier. In this Partial product multiplication process, we also used carry save adder in the last stage to accumulate the last bits. In this work 8*8-bit Partial product multiplier construction is examined and simulated in XILINX software. To implementation and simulation of 16-bit Partial product multiplier we used XILINX ISE Design suite 14.7 software. In this 8-bit Partial product multiplier circuit our main goals are to decrease the area of multiplier circuit and increase the speed of multiplier

3. EXISTING SYSTEM

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better.

Figure 1 depicts the overall topology of an exact 4: 2 compressor that is having five inputs with three outputs, and that is formed by two complete adders that are cascaded. The precise 4:2 compressor's inputs are A1, A2, A3, A4, and CIN, while the outputs are COUT, CARRY, and SUM.

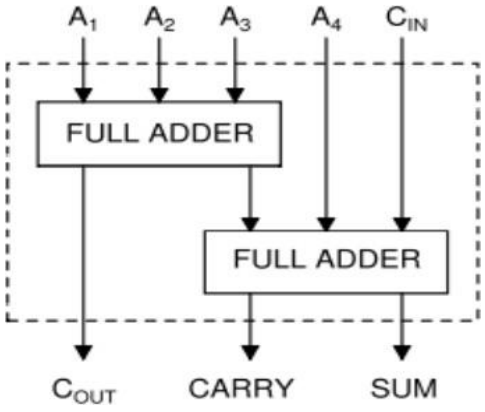

Fig.1: conventional 4 bit compressor design

This study proposes an alternate architecture for multipliers with more than three stages of cascaded compressors as a method of improving the hardware utilisation of the proposed design. The high performance with area-efficient compressor topology requires a XOR, AND, and 2 OR logical gates along with a MUX. OR and AND gates each utilize six CMOS based transistors. This research suggests and implements a design that affects the number of transistors reduction by leveraging NAND and NOR gates. Although the SUM and CARRY obtained by the modified design are not identical to those provided by the suggested 4: 2 compressor architecture, the mistake is eliminated by cascading the compressor in multiples of 2.

By approximating the 4: 2 compressors the output count can be reduced to two. To approximate the outcome, COUT is removed. When the input combination is 1111, an error occurs. The CARRY and SUM are both set to 11 when the input bits are 1111, resulting in a one-bit error.

4. PROPOSED SYSTEM

In exact equipment circuits, in opposition to programming approximations, offer transistors decrease, bring down unique and spillage control; bring down circuit postponement, and open door for scaling back. Inspired by the constrained research on surmised multipliers, contrasted and the broad research on inexact adders, and expressly the absence of estimated strategies focusing on the incomplete item age, we exclude the age of some fractional items, hence diminishing the quantity of halfway items that must be gathered, we diminish the range, power, and profundity of the amassing tree. An 8-bit unsigned multiplier is used for illustration to describe the proposed method in approximation of multipliers.

High-speed multipliers are becoming increasingly used in a variety of computing applications, including computer graphics, scientific calculations, and image processing, and so on. The multipliers speed determines how fast the processors run, and designers are currently concentrating on achieving high speed while utilising minimal electricity. The multiplication architectural style including of three main layers such as: partial product outputs production or calculation, partial product minimization/reduction process, and final adding the obtained two rows of results with general adder like RCA, PPA etc. The PPR is responsible for a significant portion of the multiplier time, energy, and area overhead. Compressors are frequently built to produce partial

products in order to reduce delay and improve performance.

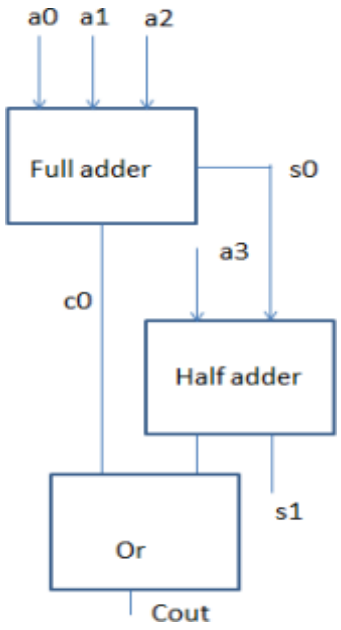

Fig: 2 proposed approximate compressor.

To analyse compressor, we are now building an 8 bit dadda and 16 bit dadda multiplier. Below is an illustration of an 8-bit dadda multiplier. The suggested multiplier offers a faster speed and better accuracy than the present multiplier.

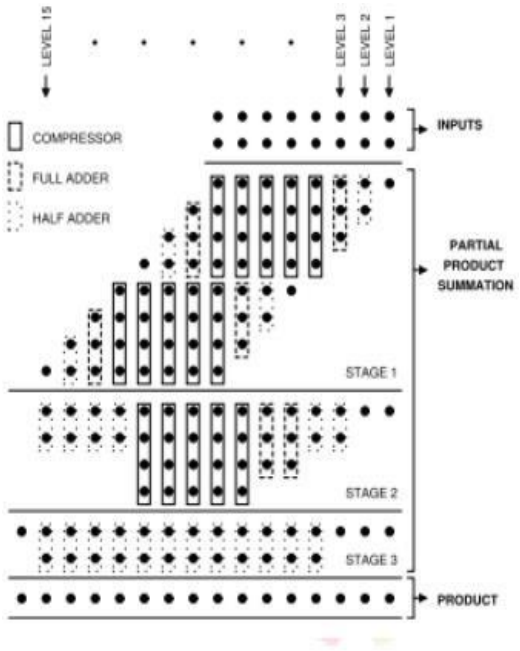

Fig.3: approximate multiplier for 8 x 8 bit

The 16-bit dadda multiplier is created by combining proposed compressors with an identical compressor that already exists. Due to proposed design, exactness is somewhat increased compared to the current multiplier, and the multiplier s performance is enhanced compared to the existing multiplier because of the suggested compressor architecture.

5. RESULTS AND DISCUSSIONS

The proposed and the existing multiplier designs are developed using Verilog HDL for 16 bits, respectively. The simulation waveform of exact multiplier 16-bits is shown in fig.4. The simulation results of waveforms for implemented design using test bench. In this, we

observe that when the based on the input signals and, for every set of input values, the output values are generated which are the approximate results that make use of the proposed compressor for the reduction of partial products in the multiplication process.

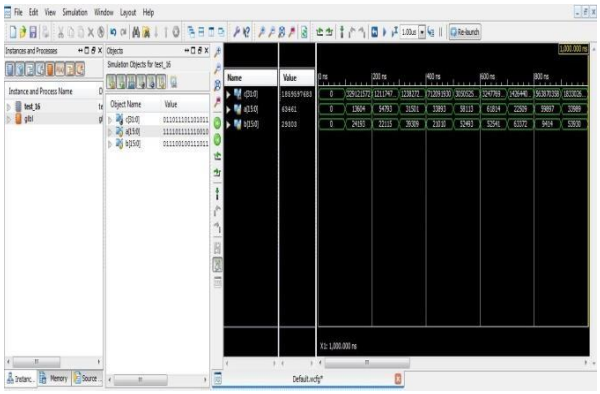

Figure 4: Simulation result of the 16-bit exact multiplier

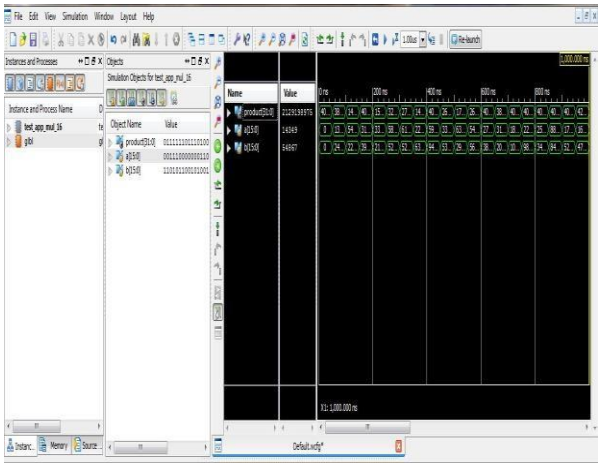

Figure 5: Simulation result of the 16-bit approximate multiplier

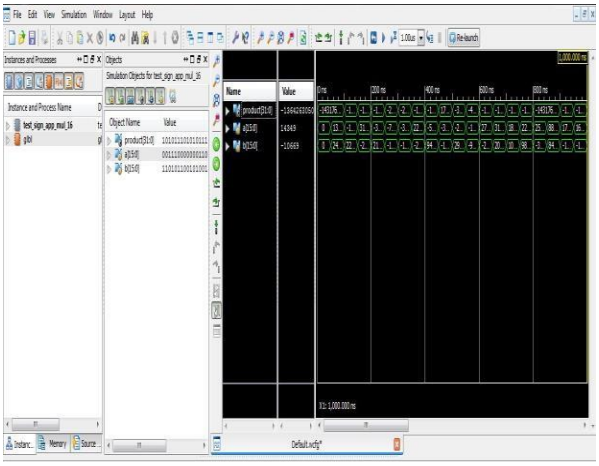

Figure 6: Simulation result of the 16-bit signed approximate multiplier

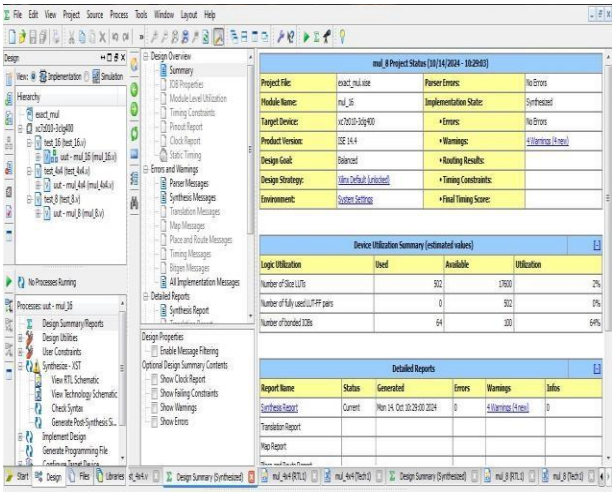

Figure 7: Summary report of the 16-bit exact multiplier

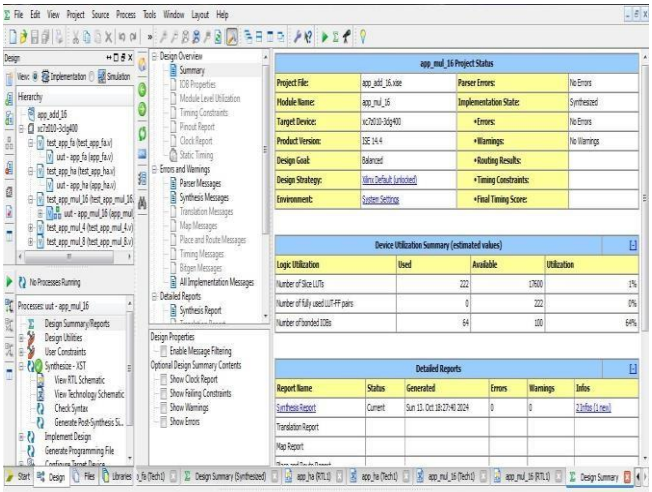

Figure 8: summary report of the 16-bit approximate multiplier

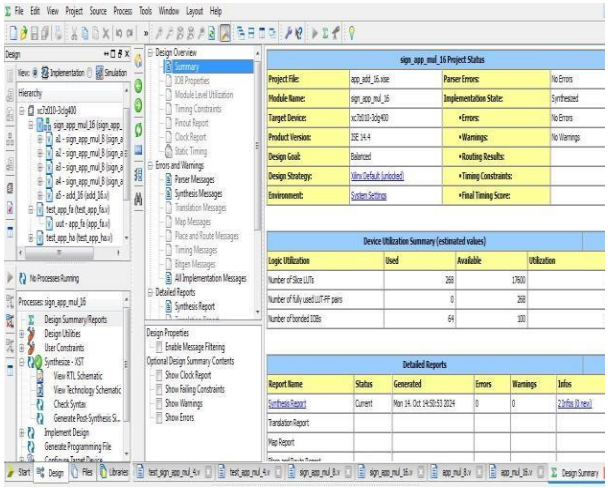

Figure 9: Summary report of the 16-bit signed approximate multiplier

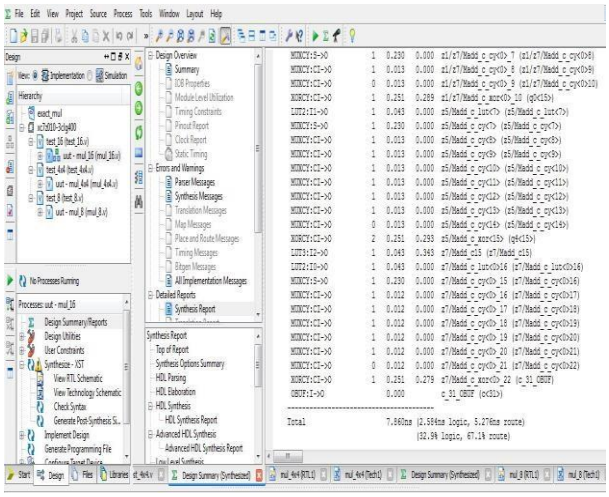

Figure 10: Delay report of the 16-bit exact multiplier

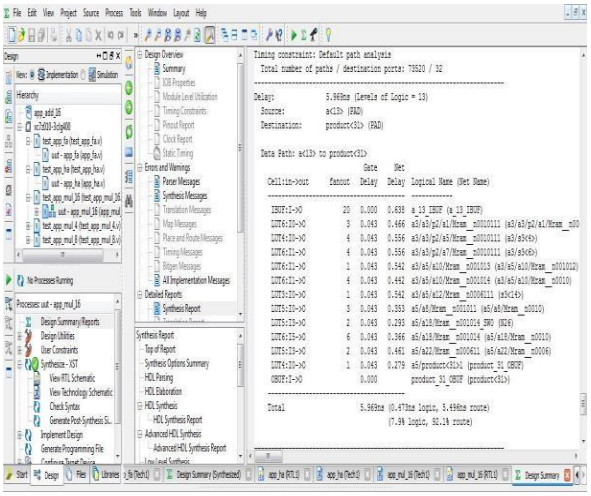

Figure 11: Delay report of the 16-bit approximate multiplier

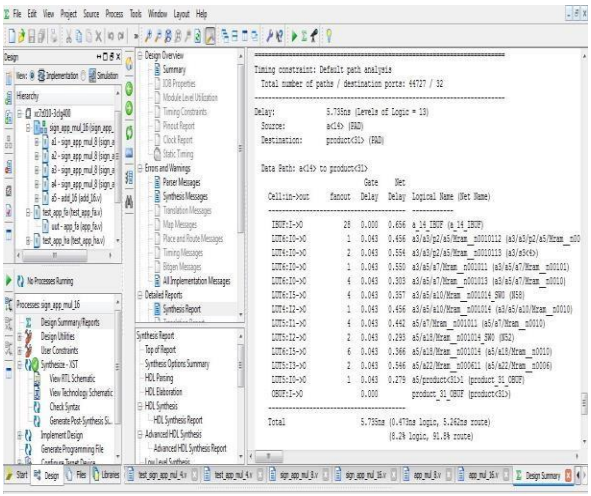

Figure 12: Delay report of the 16-bit signed approximate multiplier

Table 1: Comparison of exact and approximate multipliers

	AREA (LUT)	DELAY (nano seconds)
16-bit Exact Multiplier	502	7.860
16-bit Approximate Multiplier	222	5.969
16-bit Signed Approximate	268	5.735

Multiplier		

Table 1 describes the comparison of performance parameters such as Area and Delay between multipliers implemented with Existing and multipliers using proposed compressor. Based on these findings, we claim that the suggested method's duration is minimized by deploying an innovative compressor approach. Integrating pre-existing compressors with creative compressors which are later proposed a trade-off between area overhead and computational delay is built between existing and later implemented 16 bit multiplication algorithms.

CONCLUSION

Here a high speed approximate multiplier design has been proposed. Partial product circuitry architectures have been studied as innovative paradigm for reducing resource utilization for DSP systems. In this paper, the 4:2 compressors based partial product multiplier architecture which uses both truncation and approximation of compressor is studied. The designs functionality have been verified using Xilinx ISE design suite 14.7. A comparison of the proposed design with conventional approximate multiplier showed that it has faster operation. The synthesis and simulation results showed that the proposed multiplier design gives better improvement in delay, lesser resource utilization as compared to multiplier without optimisation. The performance of the approximate multiplier can further be improved by considering don't- care conditions and further by using variable latency adder instead of almost correct adder.

REFERENCES

[1] C. Wallace, "A suggestion for a fast multiplier," IEEE Transaction onElectronic Computers, Vol. EC13, pp.14-17.

[2] R. S. Waters, E. E. Swartzlander, "A reduced complexity Wallacemultiplier reduction," IEEE Transactions on Computers, vol. 59, no.8, pp.1134- 1137, 2010.

[3] Shahzad Asif and Yinan Kong, "Low Area Wallace Multiplier," VLSIDesign, vol. 2014.

[4] M. J. Rao, S. Dubey, "A high speed and area efficient Booth recodedWallace tree multiplier for Fast Arithmetic Circuits," in Asia Pacific Conferenceon Postgraduate Research in Microelectronics and Electronics(PrimeAsia), Hyderabad, India, 5-7 Dec. 2012, pp.220-223.

[5] Karthick, S. Karthika and S. Valannathy, "Design and Analysis ofLow Power Compressors," International Journal of AdvancedResearch in Electrical, Electronics and instrumentation Engineering,YoU, Issue 6, Dec. 2012.

[6] S. T. Oskuii, P. G. Kjeldsberg, "Power optimized partial productreduction interconnect ordering in parallel multipliers," NORCHIP,Aalborg, 19-20 Nov. 2007, pp.1-6.

[7] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate

multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[8] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[9] 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.

[10] Lombardi, with his team published a transaction work in IEEE on New metrics for the reliability of approximate and probabilistic adders, in Sep. 2013.

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

[12] G. Zervakis, S. Xydis, K. Tsoumanis, D. Soudris, and K. Pekmestzi, Hybrid approximate multiplier architectures for improved poweraccuracy trade-offs, in Proc. IEEE/ACM Int. Symp. Low Power Electron. Des. (ISLPED), Rome, Italy, Jul. 2015, pp. 79–84