# A METHODOLOGY FOR IMPROVEMENT OF ROBA MULTIPLIER FOR ELECTRONIC APPLICATIONS

[1]Angila Rose Daniel and [2]B. Deepa

[1]M.Tech in VLSI and embedded systems, Kerala technical university, India
[2]Assistant professor, EC Department, Kerala technical university, India

## ABSTRACT

*In this paper, propose an approximate multiplier that is high speed yet energy efficient. The approach is to around the operands to the closest exponent of 2. This way the machine intensive a part of the multiplication is omitted up speed and energy consumption. The potency of the planned multiplier factor is evaluated by comparing its performance with those of some approximate and correct multipliers using different design parameters. In this proposed approach combined the conventional RoBA multiplier with Kogge-stone parallel prefix adder. The results revealed that, in most (all) cases, the newly designed RoBA multiplier architectures outperformed the corresponding approximate (exact) multipliers. Thus improved the parameters of RoBA multiplier which can be used in the voice or image smoothing applications in the DSP.*

## KEYWORDS

*Accuracy, approximate computing, efficient, error analysis, high speed multiplier, RoBa architecture, kogge stone adder, DSP processing*

## 1. INTRODUCTION

Multiplier is one in every of the foremost wide used arithmetic knowledge path operation in modern digital design. In the state of art Digital Signal process and graphics applications, multiplication is an important and computationally intensive operation. The multiplication operation is actually gift in several elements of a digital system or computing device, most notably in signal process, graphics and scientific computation. Booth algorithmic program could be a crucial improvement within the style of signed binary multiplication. Later there has been work to switch booth algorithmic program for better performance.

The multiplier factor are often employed in several applications and contributes in upgrading the performance of the application. Our work includes the implementation of IIR notch filter, its output and additionally as finding it's coefficients using the modified booth multiplier. In communication fields, IIR notch filter has important applications therefore implementation of IIR notch filter has got good scope.

The multiplier blocks require intensive computations. There are three major steps to any multiplication. In the opening, the partial product square measure generated. In the second step, the partial product square measure reduced to at least one row of ultimate sums and carries.

within the third step, the ultimate sums and carries square measure additional to come up with the result. A changed booth multiplier factor ought to focus on the subsequent things.

- On reducing the full range of partial product generated. This may include any committal to writing ways or reduction of computation quality of generation partial products.
- A big quantity of delay is consumed find two's complement of multiplicand. So this delay should be reduced.
- The optimization of adder structure. Once partial product generated, they need to be classified and supplemental during a systematic manner intense less delay.

This may contemplate the utilization of correspondence of the method. Next is focus is on the tactic enclosed in adding the 2 operands; the carry propagation ought to be treated with efficiency. Finally for the hardware implementation, appropriate hardware descriptive language ought to be chosen and therefore the code ought to be optimized, synthesized and simulated mistreatment the optimum tool. IIR notch filter involves floating purpose range operations therefore the info has to be outlined. Separate codes to be written for floating point addition, floating point multiplication and floating point division. In finding the filter coefficients, separate perform is needed for pure mathematics performs like circular function. Energy minimization is one of the main design requirements in almost any electronic systems, especially the movable ones like sensible phones, tablets, and different gadgets [1]. It is highly desired to achieve this minimization with minimal performance (speed) penalty[1]. Digital signal process (DSP) blocks are key elements of those transportable devices for realizing numerous transmission applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems [2] .

Therefore, rising the speed and power/energy-efficiency characteristics of multipliers plays a key role in rising the potency of processors. Many of the DSP cores implement image and video processing algorithms where final outputs are either pictures or videos ready for human consumption. This reality permits North American nation to use approximations for rising the speed/energy potency. This originates from the restricted sensory

Activity skills of mortals in observant a picture or a video. In addition to the image and video process applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the sys-tem. Being able to use the approximate computing pro-vides the designer with the ability of making tradeoffs between the accuracy and the speed as well as power/energy consumption [2], [5]. Applying the approximation to the arithmetic units can be performed at different design abstraction levels including circuit, logic, and architecture levels, as well as algorithm and software layers [2]. The approximation may be performed using different techniques such as allowing some timing violations (e.g., voltage over scaling or over clocking) and function approximation methods (e.g., modifying the Boolean operate of a circuit) or a combination of them [4],[5].In the class of perform approximation ways, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested. In this paper, focus on proposing a high-speed low power/energy yet approximate multiplier appropriate for error resilient DSP applications. The projected approximate multiplier factor, which is also area efficient, is constructed by modifying the conventional multiplication approach at the algorithm level assuming rounded input values. Here call this rounding-based approximate (RoBA) multiplier. The projected multiplication approach is applicable to each signed and unsigned multiplications that three optimized architectures are given. The efficiencies of those structures are assessed by examination the delays, power and energy consumptions, energy-delay products (EDPs), and areas with those of some approximate and accurate (exact) multipliers. The rest of this paper is organized as follows. Section II discusses the related works about approximate multipliers. In section III, the characteristics of the proposed approximate multiplier compared with the accurate and approximate multipliers, and also its effectiveness in image processing applications are studied. Section V describes the proposed architecture Finally, the conclusion is drawn in Section VI. Future scope is described in section VII.

## 2. LITERATURE SURVEY

In this section, describe some prior research that focuses on approximate multiplier factors style as they relate to our proposed approach in approximate computing. Gupta et al. proposed many approximate adder styles that, by removing a number of the logic utilized in a conventional mirror adder, achieved improved power, area, and performance [3]. Mandeni et al. propose a bio-inspired approach, where the addition results within the multiplier were approximated by the use of OR gates for the lower part of the inputs [4]. At the same time, new metrics and methodologies are proposed in [5], [6],[7] for evaluation of modeling of approximate adders. In [3], an approximate multiplier and an approximate adder based on a technique named broken-array multiplier (BAM) were proposed. By applying the BAM approximation method of [3] to the conventional modified Booth multiplier, thus by an approximate signed Booth multiplier was presented in [5]. The approximate multiplier provided power consumption savings form 28% to 58.6% and area reductions from 19.7% to 41.8% for different word lengths in comparison with a regular Booth multiplier. Kulkarnietal. [6] suggested an approximate multiplier factor consisting of variety a number of 2 × 2 inaccurate building blocks that saved the power facility by 31.8%–45.4% over an accurate multiplier. An approximate signed 32-bit multiplier for speculation purposes in pipelined processors was designed in [7].

In [8], an error-tolerant multiplier, which calculated the approximate output by dividing the multiplication into one correct and one approximate part, was introduced, in which the accuracies for different bit widths were reported. In the case of a 12-bit multiplier factor, a power saving of more than 50% was reported. In [9], two approximate 4:2 compressors for utilizing in a regular Dada multiplier were designed and analyzed. The use of approximate multipliers in image process applications, which leads to reductions in power consumption, delay, and transistor count compared with those of an exact multiplier design, has been discussed in the literature. In [10], an accuracy-configurable multiplier architecture (ACMA) was suggested for error-resilient systems. To extend its turnout, the ACMA made use of a technique known as carry in prediction that worked on supported a precomputation logic. When compared with the exact one, the proposed approximate multiplication resulted in nearly 50% reduction within the latency by reducing the critical path. Also, Bhardwaj et al. [11] presented an approximate Wallace tree multiplier (AWTM). Again, it invoked the carry-in prediction to cut back the essential path. In this work, AWTM was used in a real-time benchmark image application showing about 40% and 30% reductions in the power and area, respectively, without any image quality loss compared with the case of victimization AN correct Wallace tree multiplier factor (WTM) structure.

In [12], approximate unsigned multiplication and division based on an approximate logarithm of the operands have been proposed. In the proposed multiplication, the summation of the approximate logarithms determines the result of the operation. Hence, the multiplication is simplified to some shift and adds operations. In [13], a method for increasing the accuracy of the multiplication approach of [12] was proposed. It was based on the decomposition of the input operands. This method considerably improved the average error at the price of increasing the hardware of the approximate multiplier by about two times.

In [16], a dynamic segment method (DSM) is presented, which performs the multiplication operation on an m-bit segment starting from the leading one bit of the input operands. A dynamic range unbiased multiplier (DRUM) multiplier, which selects an m-bit segment starting from the leading one bit of the input operands and sets the least significant bit of the truncated values to one, has been proposed in [17]. In this structure, the truncated values are multiplied and shifted to left to generate the final output. In [18], an approximate 4 ×4 WTM has been proposed that uses an inaccurate 4:2 counters.

Most of the previously proposed approximate multipliers are depended on either modifying the structure or complexity reduction of a specific accurate multiplier. In this paper, similar to [12], the difference between proposed work and [18] is that, although the principles in both works are merely similar for unsigned numbers, the mean error of this proposed approach is smaller. In addition, here suggest some approximation techniques when the multiplication is performed for signed numbers.

## 3. BASIC ALGORITHM FOR BINARY MULTIPLICATION

A Binary number is utilized in digital physics or in an exceedingly personal systems or totally different electronic devices to carryout multiplication of two numbers represented in binary format. It's engineered mistreatment binary adders. The foremost basic technique involves generating a collection of partial merchandise, so summing, this method is akin to the plan of action that is instructed to lower classes students to find long multiplication on base ten integers, however has been changed here for application to a base-2 (binary) numeral system. The steps for binary multiplication are expressed as given:

- If the given number digit is one, the number is derived down and it provides the merchandise.
- If the number digit is zero then it have a tendency to get a product that is additionally zero.

For planning such a number circuit there should always have the electronic equipment to carry out or do the sub-sequent four things:

i. It could be capable of recognizing whether or not a little is zero or one.
ii. It have to be capable of shifting the left partial product.
iii. It have to be compelled to be capable of adding all the partial-products to provide the merchandise as an addition of the partial merchandise.
iv. It must examine sign bits and if they are similar, the sign of the merchandise are going to be a Positive illustration and if the sign bits are opposite then the merchandise are going to be negative [20].

The sign little bit of the merchandise that has been keep with the higher than criteria have to be displayed together with the merchandise. From the higher than discussion it will observe that it's not necessary to attend till all the partial merchandise is shaped before polishing off the adder. In truth the addition of the partial merchandise is allotted as before long as a partial product is formed.

## 4. ROBA MULTIPLIER ARCHITECTURE

The motive behind this approximate multiplier is to make use of the ease of operation of power n (2^n ).To elaborate on the process of the approximate multiplier, first, let us denote of the input of A and B rounded value by Ar and Br, respectively. The multiplication of A by B can be write as

$$A \times B = (Ar - A) \times (Br - B) + Ar \times B + Br \times A - Ar \times Br \text{ ----1}$$

Key observation is to facilitate the multiplications of Ar * Br, Ar * B, and Br X A may be implemented just by the operation of shifting which is publicized in the equation (1). The hardware implementation (Ar – A) X (Br – B), however, is rather complex. The weight of this term in the concluding result, depends on differences of the exact numbers from their rounded ones, is typically small. Hence, it is proposed to omit this part from (Ar -A) x (Br – B), helping simplify the multiplication operation shown in the equation (2).

Hence, to perform the multiplication method, the following expression is used

$$A \times B = Ar \times B + Br \times A - Ar \times Br \text{ ---- 2}$$

While both values lead to same effect on the accuracy of the multiplier, selecting the larger one (expect for the value p=2) leads to a smaller hardware implementation for determining the nearest rounded value. It originates from the detail that the number in the composition of 3*2^p considered as do not care in the both rounding process up and down manner, and smaller logic expressions may be achieved. With the help of accurate and approximate equation the proposed architecture can be designed. Figure 5.1 provides the detail block diagram for the RoBA multiplier which is applicable for the two processing such as unsigned multiplication, signed multiplication.

If the operation is for unsigned multiplication the sign detector and sign set is disabled which can speed up the multiplication process. The two inputs are provided to the detector block which detects MSB of the input and it is provided to the sign set block to denoted signed or unsigned multiplication. Rounding and shifter are worn to reduce the operands value to the nearest power of 2 and it can be shifted with the help of barrel shifter. There are 3 levels of shifter for the following terms obtained in the approximate equation. The kogge stone adder is used to add the two functions from the shifter. The sign can be set with the help detector block.

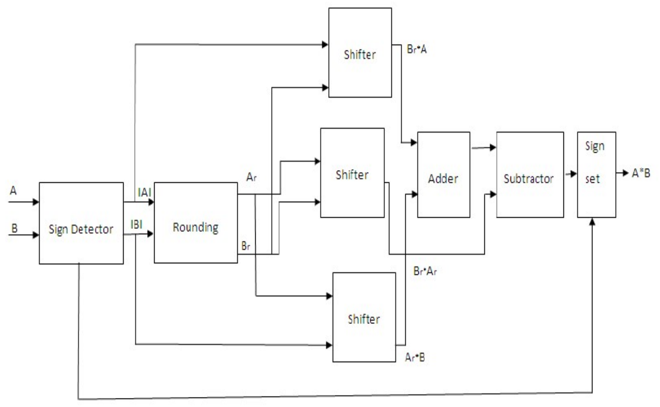

Figure 4.1: RoBA Multiplier Architecture [1]

If the output is negative the error value is calculated by inverting the output equation and it is added with binary value of 1. It supposed to be noted that contrary to the previous work where the approximate result is lesser than the exact result, the final result calculated by the RoBA multiplier may be either larger or lesser than the exact result depending on the magnitudes of Ar and Br com-pared with those of A and B , respectively. Note that if one of the operands (say A) is lesser than its equivalent rounded value while the other operand (say B ) is larger than its equivalent rounded value, then the approximate result are larger than the precise result. Because the term (An r − A) × (B r − B) will be neglected. Since the differentiation between (1) and (2) is precisely this product, the approximate result becomes higher than the exact one. Similarly, if both A and B are larger or both are lesser than Ar and Br, then the approximate result is less-er than the exact result. Hence, before the multiplication operation starts, the values of both input are absolutes and the output sign of the result are based on the inputs signs be determined and then the operation should be performed for unsigned numbers and, at the final stage, the proper sign be applied to the unsigned result.

## A. Structure Level Design Of RoBA Multiplier

From the equation 1 and 2 the structure level implementation of the multiplier were designed. The inputs are represented in the format of two's complement. First, the signs of the inputs are determined, and for each negative value, the unconditional value is generated. Next, the rounding block extracts the nearest value for each unconditional value in the form of 2^n. The bit width of the output of this block is n (the most significant bit of the absolute value of an n-bit number is zero for two's complement format).To determine the closest price of input A, the operands square measure misreckoning off to the facility of two with the assistance of misreckoning criteria. There are four cases for selecting final rounded of value from the original input values there are discussed below

i. Ar is high and Br is low.
ii. Ar is low and Br is high.
iii. Ar is high and Br is high.
iv. Ar is low and Br is low.

By selecting the case one, the approximate result is larger when observed with exact result. From the case two and three, the approximate result is somewhat larger than the accurate result in contrast with case one. For case four, the approximate result is lower than the exact result. The program should be slightly modified for each one of the cases. The rate or error is extremely low down for case one and four in contrast with other two cases. The error rate is the important factors that should be considered while designing the approximate multiplier. The distance between exact and inexact results for the approximate multiplier is calculated before calculating the error rate of the rounding based approximate multiplier. The hardware architectures of the sign detector, rounding, barrel shifter, kogge stone, subtract or and the sign set modules. The RTL architecture for RoBA multiplier is shown in Fig 4.1. The sign set block is used to negate the output if the final output is negative valued. To negate values, which have the representation of two's complement, the corresponding circuit based on X+1 should be used. To speed up negation operation, one may skip the incrementation process in the negating phase by accepting its associated error. As will be seen later, the impact on the error de-creases when an input width increases.

If the negation is performed exactly (approximately), the implementation is called signed RoBA (S-RoBA) multiplier [approximate S-RoBA (AS-RoBA) multiplier]. If the inputs are always positive, to speed up and decrease the power consumption, the sign detector and sign set blocks are omitted from the architecture, providing  with the architecture called unsigned RoBA (U-RoBA) multiplier.

## 5. PROPOSED SYSTEM DESCRIPTION

To improve the performance of conventional  multipliers we use the parallel adders in the place of adder circuits in the multiplier's architecture. From different parallel prefix adders, among that it obtained that kogge-stone adder is the best. The delay comparison for all the adders for 16, 32 and 64 bit input data width were selected the TSMC 90 nm technology. As expected, Kogge Stone Adder has the best performance among all the adders for all the input data width considered. The Kogge Stone Adder (KSA) has regular layout which makes them favored adder within the electronic technology. Another reason the KSA is the favored adder is because of its minimum fan-out or minimum logic depth. As a results of that, the KSA becomes a fast adder but has a large area. The delay of KSA is capable log (24) which is the number of stages for the "o" operator. The KSA has the area (number of "o" operators) of (n*log2^n) where n is the number of input bits.

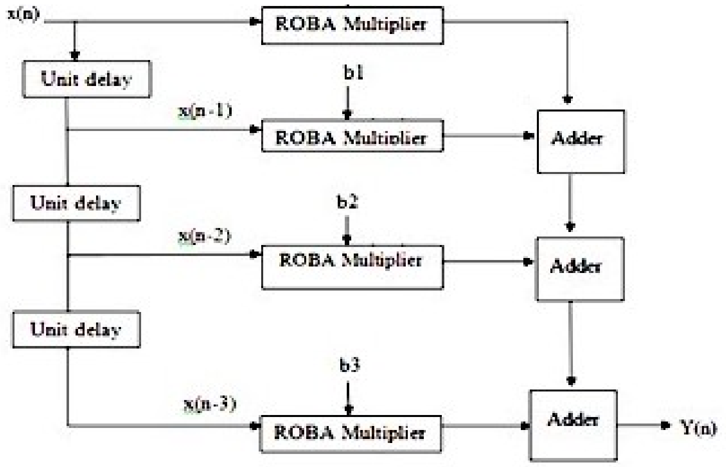

Figure 5.1: Proposed System architecture

So this work proposed a high efficient RoBA multiplier modified with kogge stone adder. The ROBA multiplier, which had high accuracy, was based on rounding of the inputs to the form of 2^n. In Fig.6.1 First, the signs of the inputs are determined, and for each negative value, the absolute value is generated. Next, the rounding block extracts the nearest value for each absolute value in the form of 2^n. Having determined the rounding values, using three shifter blocks, the products Ar × Br, Ar × B, and Br × A were calculated. The input bit width of the shifter blocks is n, while their outputs are 2^n. A single 2^n-bit kogge stone adder is used to calculate the summation of Ar × B and Br × A. The output of this adder and the result of Ar × Br are the inputs of the subtractor block whose output is the absolute value of the output of the proposed multiplier. Finally, if the sign of the final multiplication result should be negative, the output of the subtractor will be negated in the sign set block. To negate the values, two's complement representation was used.

## A. Simulation Results And Comparisons

From the previous results obtained that RoBA is an energy efficient multiplier. To improve its parameters here proposed that combining it with parallel prefix adders. The obtained simulation outputs and their performance details are described below.

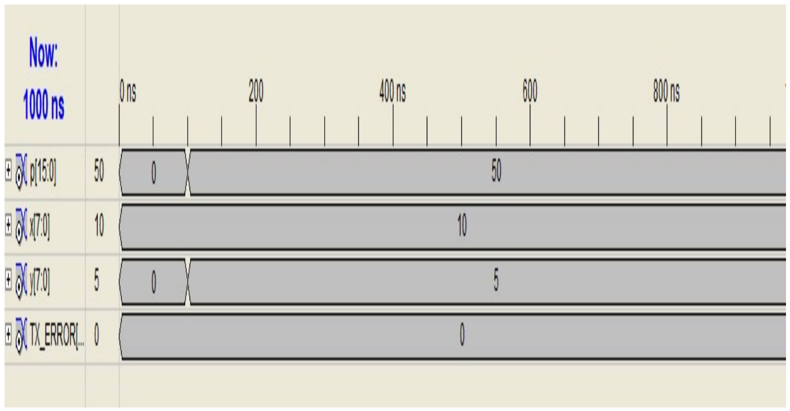

Figure 5.2: output of proposed system

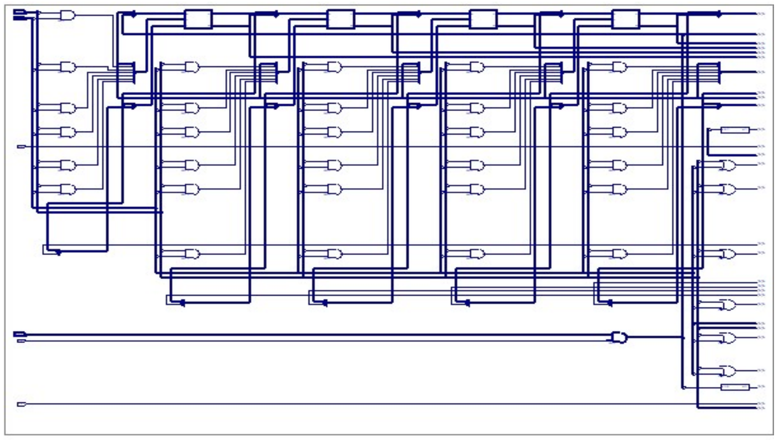

Figure 5.3: Technology schematic of proposed system

The technology schematic and corresponding outputs of proposed system are shown in figure 5.2 and 5.3.

| | Conventional RoBA multiplier | Proposed RoBA multiplier |
|---|---|---|
| LUT | 108 | 51 |
| NO. OF SLICES | 56 | 20 |
| CLOCK | 8.8 NS | 5.6 NS |

Table 1: Comparison Results

Then compared the results of conventional RoBA multiplier with the proposed high efficient low energy newly designed RoBA multiplier. Obtained results were discussed in the table 5.1 shown below.

From the above table it's clearly understood that when proposed a RoBA multiplier with parallel prefix adders its performance was increased. The results shown that the LUTs and no. of slices are reduced in this proposed system. The computations time also less for proposed one. It has 5.6 ns for computation time.

## 6. CONCLUSION

In this paper, multipliers for low power applications were enforced. Three basic algorithms namely Booth, Wallace and RoBA multipliers were implemented on Xilinx 8.31 version. The Power, Delay and space area unit calculated for these algorithms. In this paper, we proposed a high-speed yet energy efficient approximate multiplier called RoBA multiplier. The projected number, which had high accuracy, was based on rounding of the inputs in the form of 2^n. In this approach, the computational intensive part of the multiplication was omitted improving speed and energy consumption at the price of a small error. The projected approach was applicable to each signed and unsigned multiplications. In this proposed approach, combined the conventional RoBA multiplier with Kogge- stone parallel prefix adder.

The efficiencies of the projected multipliers were evaluated by scrutiny them with those of some correct and approximate multipliers mistreatment totally different style parameters. The results revealed that, in most (all) cases, the newly de-signed RoBA multiplier architectures outperformed the corresponding approximate (exact) multipliers. The no. of slices, LUTS and computation time are less required in this proposed system when compared with conventional RoBA approach. Therefore efficacy of the proposed approximate multiplier is efficiently applicable in image processing applications, i.e., image sharpening and smoothing.

## 7. FUTURE SCOPE

This low power, fast and area efficient multiplier can be used for FIR filter design, MAC design as an extension to this paper. The hardware implementations of the approximate multiplier including one for the unsigned and two for the signed operations can be done. It can be downloaded into FPGA for further improvements and observations

## REFERENCES

[1] Zendegani, Reza, Mehdi Kamal, Milad Bahadori, Ali Afzali-Kusha and Massoud Pedram. "RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25 (2017): 393-401.

[2] M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," IEEE Trans. Cir-cuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2016.

[3] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approx-imate adders," IEEE Trans. Comput.-Aided Design In-tegr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

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

[5] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghu-nathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. Int. Conf. Com-put.-Aided Design, Nov. 2011, pp. 667–673.

[6] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, "New approximate multiplier for low power digital signal processing," in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADS), Oct. 2013, pp. 25–30.

[7] D. R. Kelly, B. J. Phillips, and S. Al-Sarawi, "Ap-proximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf.Design Archit. Signal Image Process., 2009, pp. 97–104.

[8] 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 Cir-cuits (EDSSC), Dec. 2010, pp. 1–4.

[9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans.Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[10] K. Bhardwaj and P. S. Mane, “ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip,” in Proc. 8th Int.Workshop Reconfigur-able Commun.-Centric Syst.-Chip, 2013, pp. 1–6.

[11] 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. Quality Electron. Design (ISQED), 2014, pp. 263–269.

[12] J. N. Mitchell, “Computer multiplication and divi-sion using binary logarithms,” IRE Trans. Electron. Comput., vol. EC-11, no. 4, pp. 512–517, Aug. 1962.

[13] V. Mahalingam and N. Ranganathan, “Improving accuracy in Mitchell’s logarithmic multiplication using operand decomposition,” IEEE Trans.Comput., vol. 55, no. 12, pp. 1523–1535, Dec. 2006.

[14] Nangate 45nm Open Cell Library, accessed on 2010. [Online]. Available: http://www.nangate.com/

[15] H. R. Myler and A. R. Weeks, The Pocket Hand-book of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.

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

[17] S. Hashemi, R. I. Bahar, and S. Reda, “DRUM: A dynamic range unbiased multiplier for approximate ap-plications,” in Proc. IEEE/ACMInt. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 418–425.

[18] C.-H. Lin and I.-C. Lin, “High accuracy approx-imate multiplier with error correction,” in Proc. 31st Int. Conf. Comput. Design (ICCD), 2013, pp. 33–38.

[19] A. B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs,” in Proc. 49th Design Autom. Conf. (DAC), Jun. 2012, pp. 820–825.

[20] 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, Sep. 2013.