Rounding Technique Analysis for Energy Efficient Approximate Multiplier Design with approximate compressors

Firdous Amreen [1] **K. Niranjan Reddy [2]**

[1]PG Scholar, Dept of ECE, CMR Institute of Technology, Hyderabad, Telangana.
[2]Associate Professor, Dept of ECE, CMR Institute of Technology, Hyderabad, Telangana

ABSTRACT: Approximate computation is one of the better adapted effective data processing for error-resistant applications such as signal and image processing, computer vision, machine learning , data mining, etc. Approximate computation limits the precision that is appropriate, because the expense of growing the circuit characteristics depends on the task. The target accuracy is the criterion for managing the trade between between the accuracy and the features of the circuit under the supervision of the circuit builder. Within this job, the rounding methodology is used as an effective means of regulating this exchange.The simulation results for the three chosen technologies indicate a substantial improvement in the circuit characteristics in terms of strength, range, speed and energy for the proposed multiplier relative to their counterparts. Output data rounding pattern and recurrence frequency for rounded values have been added as two important things to monitor the degree of precision for increasing set of data at a minimal hardware expense.

Keywords—Data Processing, Digital Arithmetic, Approximate computing, Energy efficient, Hi-performance, Rounding Technique.

I.INTRODUCTION

The simulation results for the three chosen technologies indicate a substantial improvement in the circuit characteristics in terms of strength, range, speed and energy for the proposed multiplier relative to their counterparts. Output data rounding pattern and recurrence frequency for rounded values have been added as two important things to monitor the degree of precision for increasing set of data at a minimal hardware expense.On this point, we see countless work with a major tradeoff on precision and power-delay-energy. The basic building blocks of the multiplier are partial product production, partial product reduction and packaging. This paper introduces a rounding methodology as a modern approach for input blocks prior to partial product production. Accuracy curve, as a metric, plays a vital role in regulating and reducing the error rate to be significant depending on the method. Similar algorithms are applied at various multiplier block speeds. Input block has a 16bit and 32bit rounding strategy depending on the degree of accuracy. Produced partial products are divided into either active productsInactive partial products are both zeros and thus do not need to be included in the compressor reduction phase. This greatly decreases the amount of incomplete goods for compression. The isolation of active partial goods, which takes much of the time, has been minimized and has a positive effect on the latency of the circuit. The compressed output block includes both precise compressor blocks and estimated compressors centered on OR gates to ensure precision and reduce hardware.

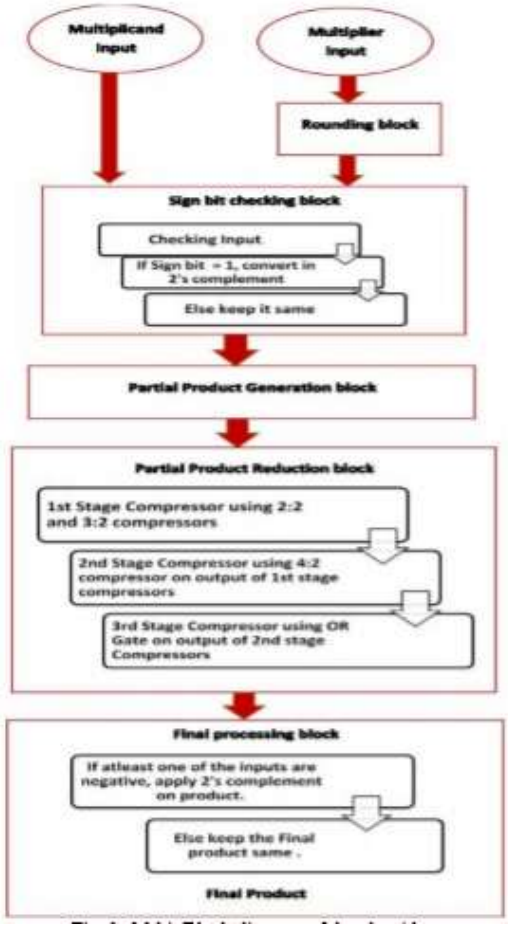
II.RELATED STUDIES:

In estimated multipliers, significant experiments focused largely on minimizing or truncating partial goods. There are various strategies for

applying indirect computing that can be divided into two types: Hardware-level strategies with fewer precise and extremely effective energy components; and software-level techniques that limit measurements or memory accesses to increase efficiency at the cost of results precision. A high-speed, energy-efficient multiplier based on input rounding in the form of 2n has recently been proposed in[1]. This strategy increased pace and energy use significantly (up to 65 per cent) As the computationally expensive multiplication element has been omitted. [2] provides a hardware architecture with reconfigurable kernels and an overflow-resilient limiter when interpreted at the hardware level. Recent literature contains several methods which relax a single part of a computing device (e.g. a functional unit[3]) for improved design. Authors in [4] suggested an indirect configuration of multipliers with an error distribution that reduces the propagation time and increases energy efficiency. None of the work focussed on pre-computing results. This paper introduces a rounding of input data before provisional implementation of multipliers and reveals a substantial reduction I

III. III. PROPOSED Architecture OF MULTIPLIER APPROXIMATES

A. Block diagram The key concept behind the suggested approximate multiplier is to make multiplication use of rounded data. The proposed algorithm applies a rounding technique to the partial product generation before moving the data on. Illustration. 1 Displays the specification map for the suggested system for the estimated construction of multipliers. The multiplier is rounded first from the two inputs (Multiplicand and Multiplier) by going through the rounding tube. Until multiplication begins, the



Sign bit of both inputs is stored, and the output sign of the multiplication value is calculated depending on the input signs. The correct sign is applied on the test at the last point. In case of combining negative numbers the corresponding input blocks should be translated into the complement of their 2. In With N-bit input, partial products (half products) are generated in traditional multipliers. But the partial products produced in the rounding technique are the mixture of active and inactive component products. The active partial products are, which have "1" as the Multiplier coefficient. As a consequence it produces a complete Multiplicand row after rounding. Inactive partial products are also the lines of complete 0's. Thus, in the reduction process, there is no need to cover them.

B. Rounding data

TABLE 1: ROUNDING ALGORITHM (16-BIT)

Accurate Bit Position	Approximate Bit Position
bit0	bit1
bit1	bit1
bit2	bit1
bit3	bit4
bit4	bit4
bit5	bit4
bit6	bit7
bit7	bit7
bit8	bit7
bit9	bit10
bit10	bit10
bit11	bit10
bit12	bit13
bit13	bit13
bit14	bit15
bit15	bit15

A = 0001 0011 1000 1000

B = 0000 0011 1111 1111

Br = 0000 0100 1001 0010

15 13 10 7 4 1

Leads to active partial product rows

Rounding data on inputs requires a significant obligation to ensure consistency. It can be reported with a simple assumption, that rounding lower bits results in less error relative to rounding higher bits. The suggested algorithm has thus allocated rounding weights with respect to the magnitude of the bit location. From fig to fig. 2, Proposed approach reveals 16-bit error location curve which gives lower bits less weight and higher bits more weight. There is a slight error difference between exact position of the bit and rounded position of the bit.

There is a corresponding rounded bit value assigned for each precise bit as seen in Table 1. Error gap decreases with increases in the value of the bit position. Illustration. 3 Specifies an scenario where inputs are 'A' and 'B' and inputs are rounded to 'Br.' "Rounding Technique" basically checks for a '1' in the 'X' bit position and assigns '1' with or without a small error to the respective 'Y' bit position

C. Partial product reduction

The reduction of partial products is a stage in which partial products are compressed using different types of compressors. The proposed algorithm provides consistency to the the number of rows of component items. For an alternative 16-bit architecture seen in Fig. 6 Reduces partial products to 6 rows known as active partial products. N-bit inputs are combined, like all conventional approaches, to produce partial N / N products. As the number of bits increases, the length of Partial products decreases with O(N2) in terms of the computational complexity. The proposed algorithm supplies 16-bit O(N6) and 32-bit O(N113) computational complexity.

For better understanding, along with design, an example is explained with input values A, B and Br (from fig. 6(right)). Multiplier data ‘ B 'is first rounded to ‘ Br'. Inputs are then combined to get N×N partial products. According to rounding of multiplier data, N×N partial products is a mixture of active and inactive partial products. Multiplier of ‘ 1 'as coefficient, after rounding, produces a full row of Multiplicand as a consequence as shown in fig. 6. Therefore, inactive partial products are the whole zero values line which had "0" as the coefficient on the Multiplier. Thus, there is no need to cover them in the reduction process. In addition, inactive partial goods could only inc

It has prompted us to first delete, all unused partial goods before packaging. This method plays important role in rising power, region and time use and in effect increasing its efficiency. The active partial products are compressed and packed using three levels of compression. In the 1st step, partial items are compressed using full adders and half adders. The performance of 1st stage compression is further compressed using a 4:2 compressor while inputs are 16bit while for 32bit, 9:2 compressor

Illustration. 6 (right) shows corresponding operations on an illustration. The output of the

2nd stage compressor is eventually packed using OR gate to get a final product. Conventionally complete adders are used instead of OR. The whole purpose of using OR gate instead of full adder is to reduce area and energy consumption significantly

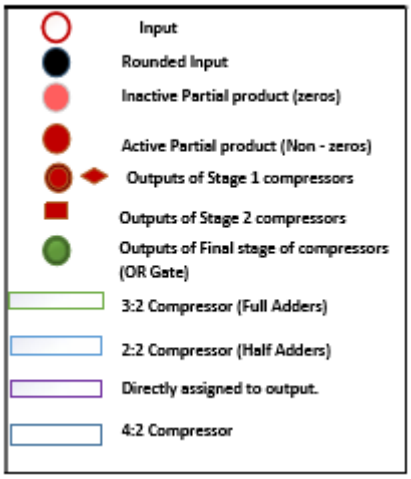
IV. ROUNDING ERROR ANALYSIS

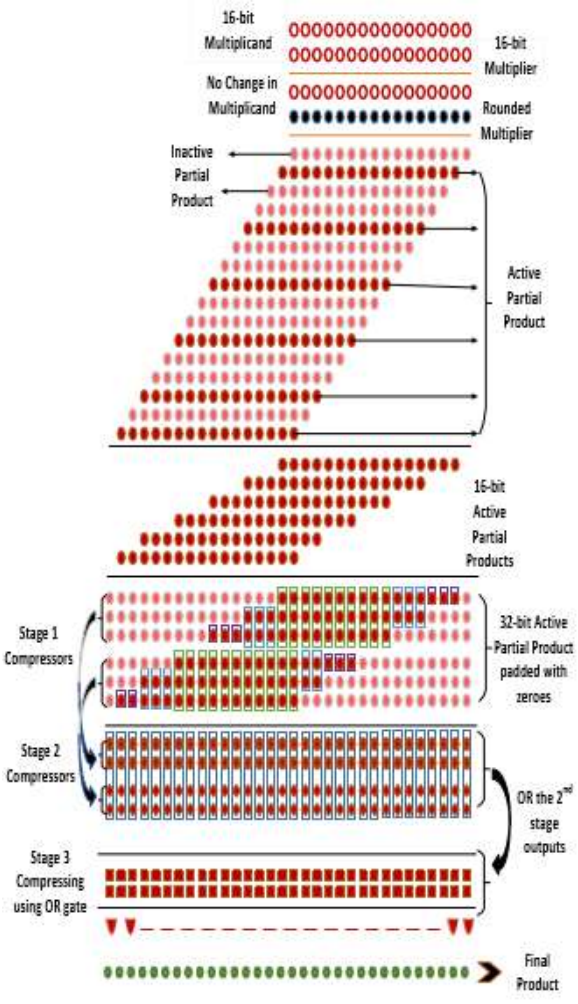
As seen in previous sections, only one Input (Multiplier) is rounded. As an instance let us find 16-bit multiplier to evaluate rounding error. From given inputs Multiplicand and

TABLE 2: ROUNDED VALUES AND THEIR OCCURRENCES RATE

Rounded value (DEC)	# of Occurrence of Rounded value	Rounded value (DEC)	# of Occurrence of Rounded value	Rounded value (DEC)	# of Occurrence of Rounded value
0	1	8338	1029	33936	1029
2	7	9216	21	33938	7203
16	7	9218	147	40960	9
18	49	9232	147	40962	63
128	7	9234	1029	40976	63
130	49	9344	147	40978	441
144	49	9346	1029	41088	63
146	343	9360	1029	41090	441
1024	7	9362	7203	41104	441
1026	49	32768	3	41106	3087
1040	49	32770	21	41984	63
1042	343	32784	21	41986	441
1152	49	32786	147	42000	441
1154	343	32896	21	42002	3087
1168	343	32898	147	42112	441
1170	2401	32912	147	42114	3087
8192	3	32914	1029	42128	3087
8194	21	33792	21	42130	21608
8208	21	33794	147		
8210	147	33808	147		
8320	21	33810	1029		
8322	147	33920	147		
8336	147	33922	1029		

Multiplier, Multiplier feedback is rounded. For 16-bit values ranging from 0 to 65535, a set of rounded values are obtained using rounding algorithm as discussed in fig. 2 from section III. Illustration. 4 maps the relationship between

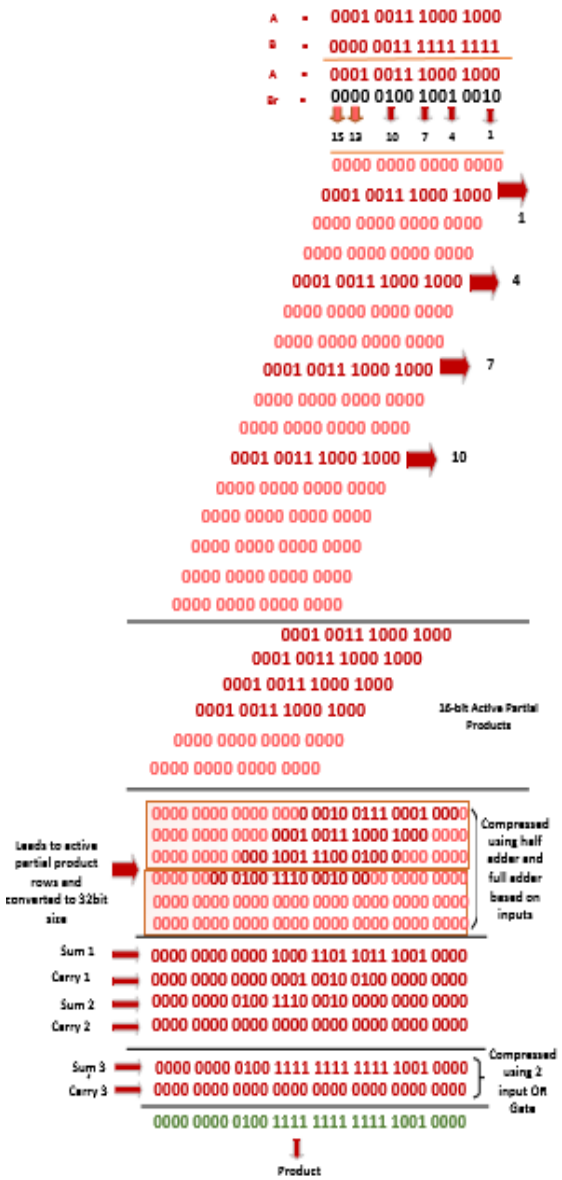




rounded values and its occurrences. The phase pattern obtained, shows

how similar the rounded values are with each other. Many of the instances,

phase sizes are low which infers as lesser error and thus more precision. Just in one scenario, we see higher step size which can cause marginally higher error. This emphasises us effective data processing is performed before the multiplication to primarily work on this field with a few additional hardware to obtain improved accuracy



Therefore, with this detailed analysis, a method. We then measured the probability of occurrence of rounded values for numbers ranging from 0 to 65535. Yellow lines in fig. 4, provides very strong example about how rounded values are spread with the probabilities. Area until 9360 rounded values have lower probability than provides higher accuracy. Even, area from rounded value 40960 onwards will have reduced chance. Area at the centre, with higher probability gives an opportunity to change rounding pattern to get better accuracy. Further studies and research will be mainly focusing on these areas to optimise algorithm with the expense of extra hardware. Table 2 includes number of instances of rounded values against input values.

V.Extension:

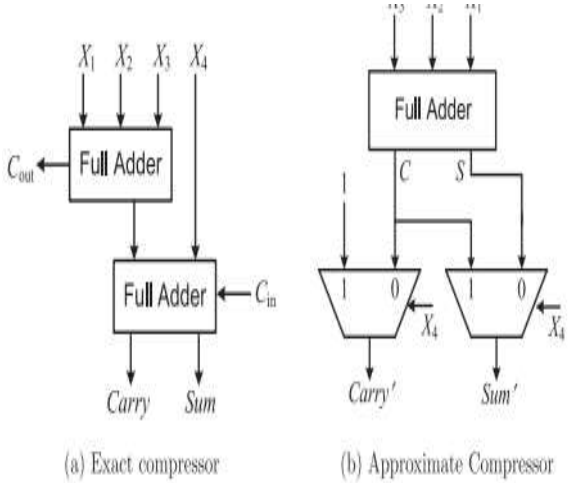
The compressor collects the input Cin from a previous block of order one binary bit lower in significance, and generates outputs Cout and Carry of order one binary bit higher in significance. The block diagram of identical 4–2 compressor is seen in Fig. 1a. An improved configuration of a 4–2 compressor proposed in [9] is used to compare the output of estimated compressors in this article. The following Boolean Eqs. (1) – (3) represent the operation of this compressor.

In this project at stage 4 we used carry propagate adder used with exact compressors an extension we did approximate compressor this we can reduce delay of the approximate rounding multiplier.

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in} \quad (1)$$

$$C_{out} = (X_1 \oplus X_2)X_3 + \overline{(X_1 \oplus X_2)}X_1 \quad (2)$$

$$Carry = (X_1 \oplus X_2 \oplus X_3 \oplus X_4)C_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)}X_4 \quad (3)$$



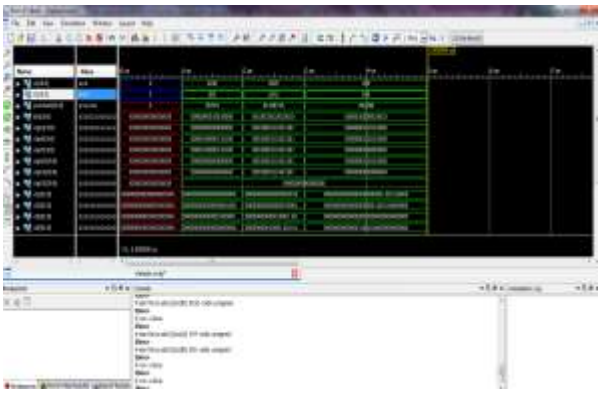
(a) Exact compressor (b) Approximate Compressor

VI.RESULTS

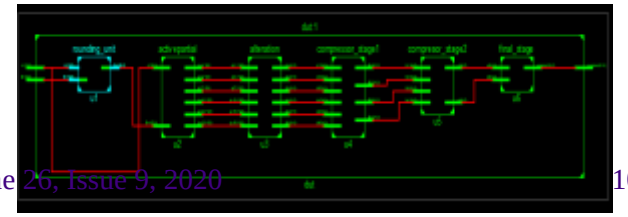
The developed project is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level net list mapped to a specific technology library. Here in this Spartan 3E family, many different devices were available in the Xilinx ISE tool. In order to synthesis this design the device named as "XC3S500E" has been chosen and the package as "FG320" with the device speed such as "-5".

This design is synthesized and its results were analyzed as follows:

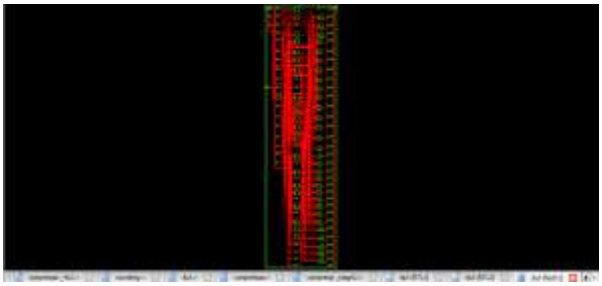
SIMULATION RESULTS **:**



RTL SCHEMATIC DIAGRAM



TECHNOLOGICAL SCHEMATIC



DESIGN SUMMARY:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	106	2400	4%
Number of fully used LUT-FF pairs	0	106	0%
Number of bonded IOBs	64	102	62%

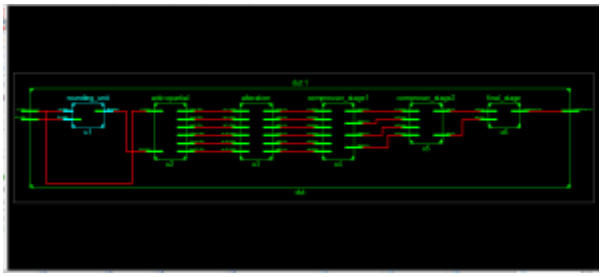
TIMING SUMMARY:

```
Timing Summary:
---------------
Speed Grade: -2

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 20.459ns

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 5720 / 31
-------------------------------------------------------------------------
Delay:               20.459ns (Levels of Logic = 14)
  Source:            B<4> (PAD)
  Destination:       product<28> (PAD)
```

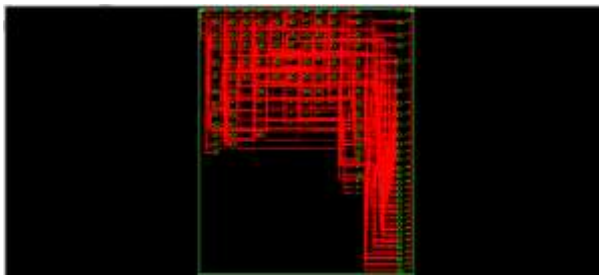
EXTENSION RESULTS

SIMULATION RESULTS

RTL BLOCK DIAGRAM:



TECHNOLOGICAL SCHEMATIC:



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	51	2400	2%
Number of fully used LUT-FF pairs	0	51	0%
Number of bonded IOBs	64	102	62%

DESIGN SUMMARY:

```
Timing Summary:
---------------
Speed Grade: -2

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 9.043ns

Timing Details:
---------------
All values displayed in nanoseconds (ns)

--------------------------------------------------------------------------------
Timing constraint: Default path analysis
  Total number of paths / destination ports: 331 / 30
--------------------------------------------------------------------------------
Delay:               9.043ns (Levels of Logic = 5)
  Source:            B<7> (PAD)
  Destination:       product<14> (PAD)
```

TIMING SUMMARY:

COMPARISON TABLE

MODULE	DELAY
ROUNDING MULTIPLIER WITH EXACT MUTIPLIER	20.43ns
ROUNDING MULTIPLIER WITH APPROXIMATE MUTIPLIER	**9.043ns**

VII.CONCLUSION

In this paper, an accuracy-configurable adder without suffering the cost of the increase in power or in delay for configurability was proposed. The proposed adder is based on the conventional CLA, and its configurability of accuracy is realized by masking the carry propagation at runtime. The experimental results demonstrate that the proposed adder delivers significant power savings and speedup with a small area overhead than those of the conventional CLA. Furthermore, compared with previously studied configurable adders, the experimental results demonstrate that the proposed adder achieves the original purpose of delivering an unbiased optimized result between power and delay without sacrificing accuracy. It was also found that the quality requirements of the evaluated application were not compromised.

VI. REFERENCES

[1] S. Cotofana, C. Lageweg, and S. Vassiliadis, “Addition related arithmetic operations via controlled transport of charge”, IEEE Transactions on Computers, vol. 54, no. 3, pp. 243-256, Mar. 2005.

[2] V. Beiu, S. Aunet, J. Nyathi, R. R. Rydberg, and W. Ibrahim, “Serial Addition: Locally Connected Architectures”, IEEE Transactions on Circuits and Systems-I: Regular papers, vol. 54, no. 11, pp. 2564-2579, Nov. 2007.

[3] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Quality programmable vector processors for approximate computing”, 46th Annual IEEE/ACM International Symposium on Micro architecture (MICRO), pp. 1-12, Dec. 2013.

[4] A. B. Kahng, and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs”, IEEE/ACM Design Automation Conference (DAC), pp. 820-825, Jun. 2010.

[5] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, “On Reconfiguration- Oriented Approximate Adder Design and Its Application”, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 48- 54, Nov. 2013.

6] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-Power digital signal processing using approximate adders”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124-137, Jan. 2013.

7] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft-Computing applications”, IEEE Transactions on Circuits and Systems I: Regular papers, vol. 57, no. 4, pp. 850-862, Apr. 2010.

[8] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, “MACACO: modeling and analysis of circuits for approximate computing”, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 667-673, Nov. 2011.

[9] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008

10] J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders”, IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771, Sep. 2013.