ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

Implementation of Reduced Area and Power Approximate Adder in VLSI Technology

SHAIK CHINNA BAJI[1], ANJALIDEVI PONAMALA[2]
[1]Assistant professor, MVR College of engineering and technology, Paritala 521180
[2-]Assistant professor, MVR College of engineering and technology, Paritala 521180

ABSTRACT

Many emerging applications are preferring approximate computing because Approximate computing is an efficient approach for error-tolerant applications because it can trade off accuracy for speed. Addition is a key fundamental function for these applications these days approximate computing is employed in many DSP applications where it gives a great flexibility to change the image pixels. In this paper, proposing with a high speed accuracy-configurable adder that also maintains a small design area. The proposed adder is based on the conventional carry look-ahead adder, and its configurability of accuracy is realized by masking the carry propagation at runtime. Compared with the conventional carry look-ahead adder, the proposed experimental result demonstrates the achievement of the original purpose of optimizing area, power and speed simultaneously without reducing the accuracy. In this paper is implemented in Xilinx ISE using VERILOG language and waveforms are observed through simulation
Keywords—Approximate computing configurable adder; carry look ahead adder, A carry-mask able half adder.

-

Applications that have recently emerged (such as image recognition and synthesis, digital signal processing, which is computationally demanding, and wearable devices, which require battery power) have created challenges relative to power consumption, speed and area. Addition is a fundamental arithmetic function for these applications. Most of these applications have an inherent tolerance for insignificant inaccuracies. By exploiting the inherent tolerance feature, approximate computing can be adopted for a tradeoff between accuracy and speed. At present, this tradeoff plays a significant role in such application domains. As computation quality requirements of an application may vary significantly at runtime, it is preferable to design quality configurable systems that are able to tradeoff computation quality and computational effort according to application requirements.

The previous proposals for configurability suffer the cost of the increase in power or in delay. In order to benefit such application, a low-area and high-speed adder for configurable approximation is strongly required. In this paper, we propose a configurable approximate adder, which consumes lesser power than does with a comparable delay and area. In addition, the delay observed with the proposed adder is much smaller than that of with comparable power

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

consumption. Our primary contribution is that, to achieve accuracy configurability the proposed adder achieved the optimization of power and delay simultaneously and with no bias toward either. The design of the proposed CLA by using the carry-maskable half adders in Verilog HDL. Then we evaluated the power consumption, critical path delay, and design area for each of these implementations, Compared with the conventional CLA. This paper also evaluated the quality of these accuracy configurable adders in a real image processing application**.**

II. Literature Survey

Mahdiani et al. [7] proposed a new architecture stating the approximate computing the lower bits with or gates while the MSB bits with accurate adders. Venkatesan et al. [8] paved a path for construction of an circuitry for approximate computing . the approximate design with fixed bit accuracy may lead to wastae of power and area . Kahng et al. [4] proposed an accuracy-configurable adder(ACA), which is based on a pipeline structure. The correction scheme of the ACA proceeds from stage 1 to stage 4, if the most significant bits of the results are required to be correct, all the four stages should be performed. Motivated by the above, Ye et al. [5] proposed adder where approximate sum and accurate sum is calculated and which is mandatory that is chosen Similar to [5], our adder design dosent consists of pipeline structure, but the levels of computation accuracy and to obtain the configurability of accuracy, some multiplexers and additional logic blocks are required in [5].However, the approximate adder design with accuracy scalability is constructed using the CMHA blocks and CLA.

III. Carry Look Ahead Adder

A **carry-look ahead adder** (**CLA**) or **fast adder** is a type of parallel adder used in digital logic. A carry-look beforehand adder advances in pace by way of decreasing the quantity of time required to decide deliver bits. It may be contrasted with the easier, but generally slower, ripple-carry adder (RCA), for which the bring bit is calculated alongside the sum bit, and every degree have to wait till the preceding carry bit has been calculated to begin calculating its very own sum bit and convey bit. The deliver-look beforehand adder calculates one or greater deliver bits before the sum, which reduces the wait time to calculate the end result of the bigger-price bits of the adder. The Kogge–Stone adder (KSA) and Brent–Kung adder (BKA) are examples of this sort of adder. The following are the techniques to get the high speed in the parallel adder to provide the binary addition.

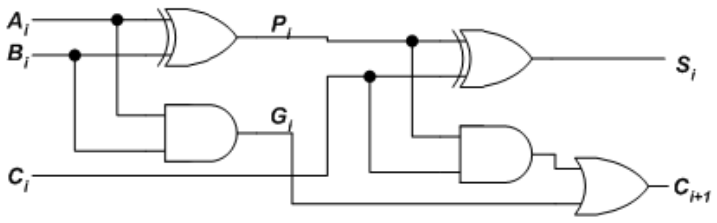

Fig 1: 1 Bit Carry Look Ahead Adder

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

1. Consider the full adder circuit shown above figure. If we define two variables as the first step is carry generate Gi and carry propagate Pi then,

$$P_i = A_i \oplus B_i$$

$$Gi = Ai\ Bi$$

2. Then the next step is carry generation Ci +1 = Gi + Pi Ci

3. The last step is post processing sum output t can be expressed as

$$Si = Pi \oplus Ci$$

Where Gi is a carry generate which produces the carry when both Ai, Bi are one regardless of the input carry. Pi is a carry propagate and it is associate with the propagation of carry from Ci to Ci +1.The carry output Boolean function of each stage in a 4 stage carry-Look ahead adder can be expressed as

$C_{i+1}=G_i+P_i(C_{i-1})$

$$C_{0+1} = G0 + P0\ Cin$$

$$C_{1+1} = G1 + P1\ G0 + P1\ P0\ Cin$$

$$C_{2+1} = G2 + P2\ G1 + P2\ P1\ G0 + P2\ P1\ P0\ Cin$$

$$C_{3+1} = G3 + P3\ G2 + P3\ P2\ G1 + P3\ P2\ P1\ G0 + P3\ P2\ P1\ P0\ Cin$$

The cla is said to the additional combinational circuit which helps in calculating the next carry by which the delay can be reduced and the generated carry is driven into the full adder blocks as with the inputs, as shown below.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

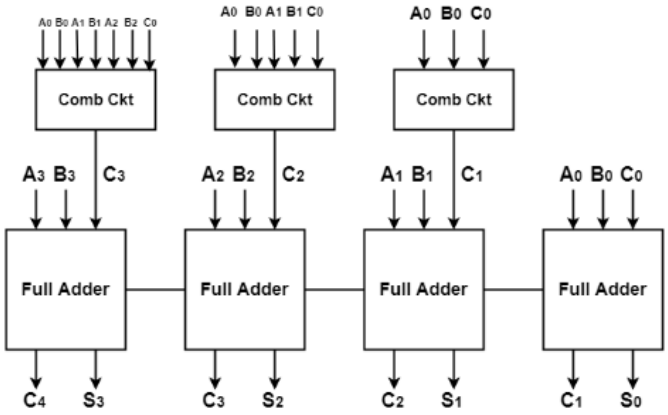

Fig 2: 4 Bit Carry Look Ahead Adder

IV. ACCURACY-CONFIGURABLE ADDER

The working of the cla can be explained in three steps in first step the propagation and generation functions are calculated, followed by calculating carry in second step and in final step calculation of sum is done.

$$P_i = A_i \oplus B_i,\ G_i = A_i \cdot B_i, \quad (1)$$

$$C_i = G_i + P_i \cdot C_{i-1}, \quad (2)$$

$$S_i = P_i \oplus C_{i-1}. \quad (3)$$

Where *i* is denoted the bit position from the least significant bit. Note that owing to reuse of the circuit of A*i* *XOR* B*i* for S*i* generation, here P*i* is defined as A*i* *XOR* B*i* instead of A*i* *OR* B*i*. Because C0 is equal to G0, if G0 is 0, C0 will be 0. From (2), we find that C1 is equal to G1 when C0 is 0. In other words, if G0 and G1 are equal to 0, C0 and C1 will be 0. By expanding the above to *i*, C*i* will be 0 when G0, G1, … , G*i* are all 0. This means that the carry propagation from C0 to C*i* is masked. From (3), we can obtain that S*i* is equal to P*i* when C*i*-1 is 0. From the perspective of approximate computing, if G is controllable and can be controlled to be 0, the carry propagation will be masked and S (=P) can be considered as an approximate sum. In other words, we can obtain the selectivity of S between the accurate and approximate sum if we can control G to be A *AND* B or 0. Evidently, we can achieve selectivity by adding a select signal. Figure 3(a) is a conventional half adder and Fig. 3(b) is a half adder to which the select signal has been added. Compared with the conventional half adder, we add a signal named "M_X" as the select signal and use a 3-input AND gate to replace the 2-input one. When M_X = 1, the function of G is the same as that of a conventional half adder; when M_X = 0, G is equal to 0. Consider the condition when the inputs A*i* and B*i* are both 1, when M_X*i* = 1, the accurate sum S*i* and carry C*i* will be 0 and 1 ({C*i*, S*i*} = {1, 0}); when M_X0 , M_X1 , … , M_X*i* are all 0, S*i*

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

is equal to P*i* (= A*i* *XOR* B*i* = 0) as an approximate sum and C*i* is equal to 0 ({C*i* , S*i*} = {0, 0}) as discussed above. Here {,} denotes concatenation. This means that the difference between the accurate and approximate sum is 2.

Toward better accuracy results for the approximate sum, we use an OR function instead of an XOR function for P generation when M_X = 0. A 2-input XOR gate can be implemented by using a 2-input NAND gate, a 2-input OR gate, and a 2-input AND gate. An equivalent circuit of the conventional half adder is shown in Fig. 4. This is called a carry maskable half adder (CMHA). The dashed frame represents the equivalent circuit of a 2-input XOR (M_X = 1). We can obtain the following: P is equal to A *XOR* B, and G is equal to A *AND* B when M_X = 1; when M_X = 0, P is equal to A *OR* B and G is 0. Thus, M_X can be considered as a carry mask signal.

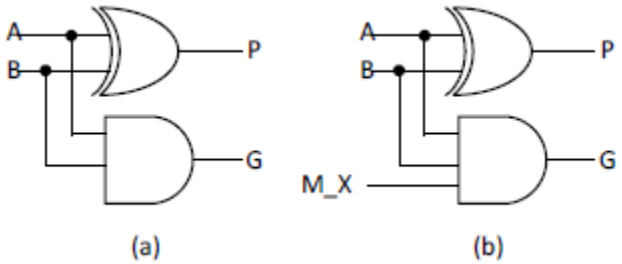

Fig.3. (A) An Accurate Half Adder, And (B) A Half Adder with a Select Signal.

Consider an n-bit CLA, whose half adders for G and P signals preparation are replaced by CMHAs. In this case, an n bit carry mask signal for each CMHA is required. To simplify the structure for masking carry propagation, we group four CMHAs and use a 1-bit mask signal to mask the carry propagation of the CMHAs in each group. The structure of a group with four CMHAs is shown in Fig. 5 as an example. A3- 0, B3-0, P3-0, and G3-0 are 4-bit-length signals and represent {A3, A2, A1, A0}, {B3, B2, B1, B0}, {P3, P2, P1, P0}, and {G3, G2, G1, G0,}, respectively. M_X0 is a 1-bit signal and is connected to the four CMHAs to mask the carry propagation simultaneously. When M_X0 = 1, P3-0 = A3-0 *XOR* B3-0, and G3-0 = A3-0 *AND* B3- 0; when M_X0 = 0, P3-0 = A3-0 *OR* B3-0, and G3-0 = 0. We proposed an accuracy-configurable adder by using CMHAs to mask the carry propagation.

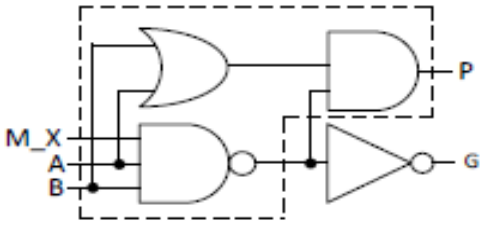

Fig 4. A Carry-Maskable Half Adder.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

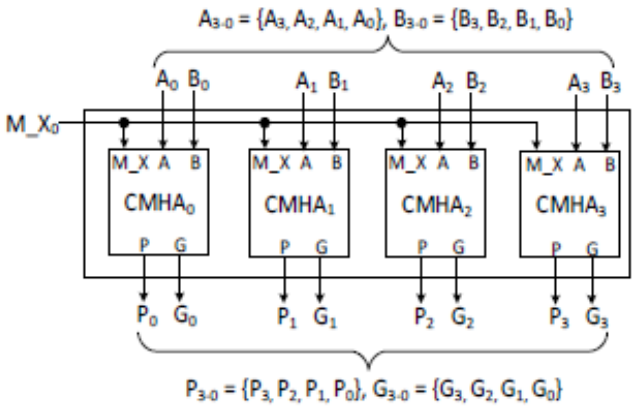

Fig.5. Structure of a group with four CMHAs.

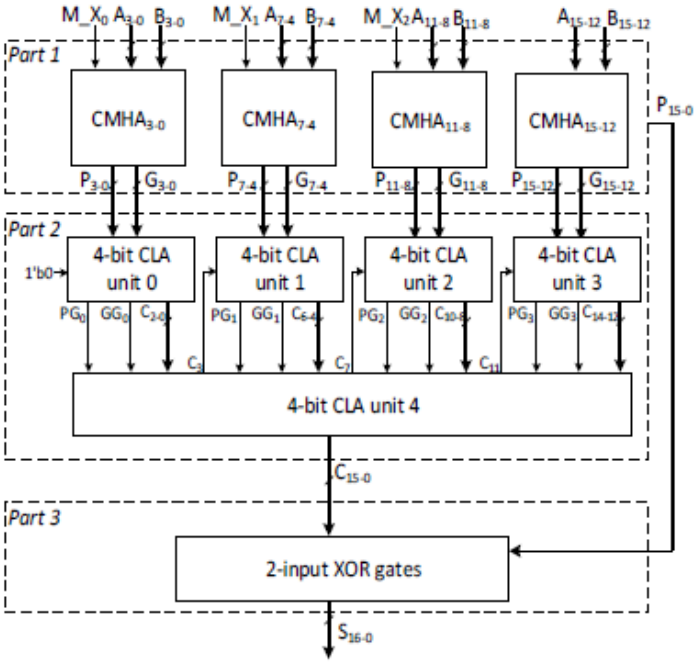

Fig 6. Structure of the proposed 16-bit adder.

The design structure of the proposed 16-bit adder is flaunted in Fig. 6 as. Four groups (CMHA3-0, CMHA7-4, CMHA11- 8, and CMHA15-12) are employed to produce the propogation and generation signals each of 16 bit . Each group comprises of four CMHAs There is no mask signal for CMHA15-12 in this design ; therefore, accurate P15-12 (= A15-12 *XOR* B15-12) and G15-12 (= A15-12 *AND* B15-12) are always obtained. P15-0 and G15-0 are the outputs from Part 1 and are connected to Part 2. Note that P15-0 is also connected to Part 3 for

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

sum generation. In Part 2, four 4-bit carry look-ahead units (unit 0, 1, 2, 3) generate four PGs (PG0, PG1, PG2, and PG3), four GGs (GG0, GG1, GG2, and GG3), and 12 carries (C2-0, C6- 4, C10-8, and C14-12) first, and then the carry look-ahead unit 4 generates the remaining four carries (C3, C7, C11, and C15) by using the PGs and GGs. C15-0 is the output of Part 2 and is connected to Part 3. The fifteen 2-input XOR gates in Part 3
Generate the sum.

V. RESULTS

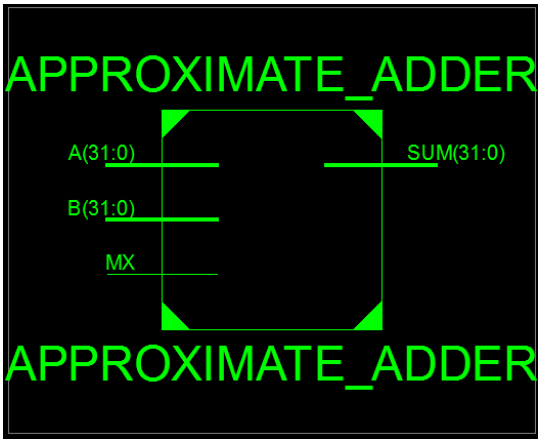

Fig7: RTL schematic view

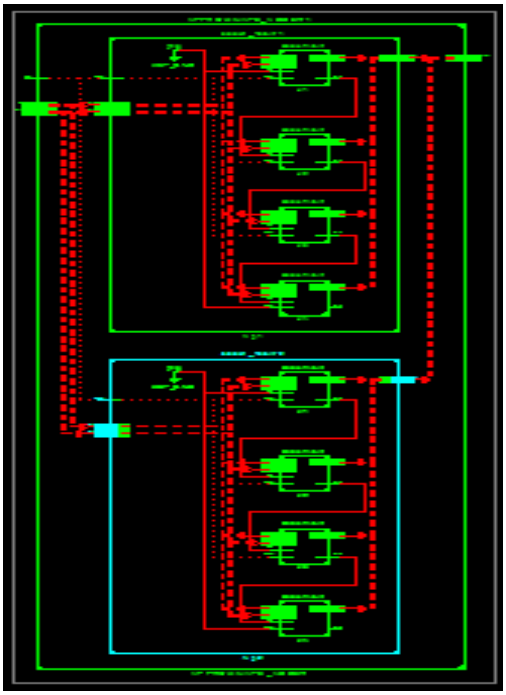

Fig8: Intenal structure RTL schematic view

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

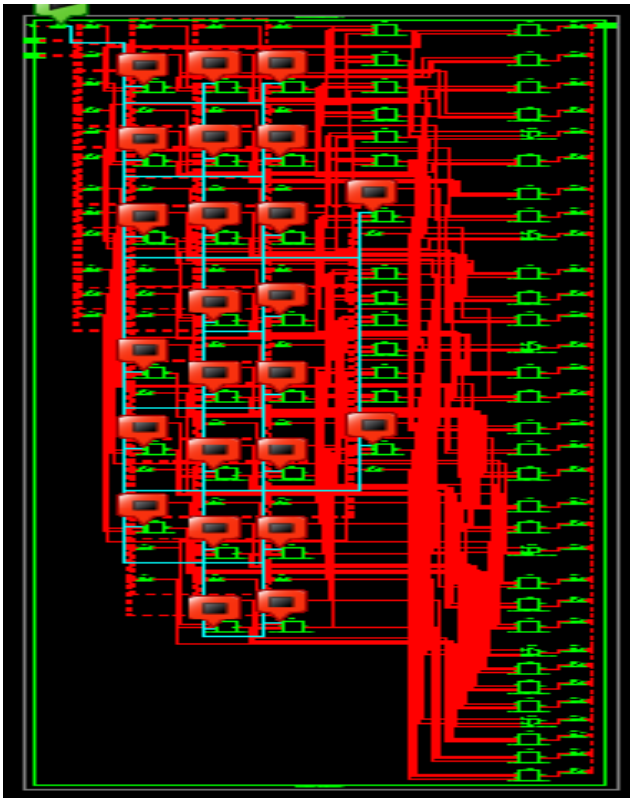

Fig9 : View technology schematic

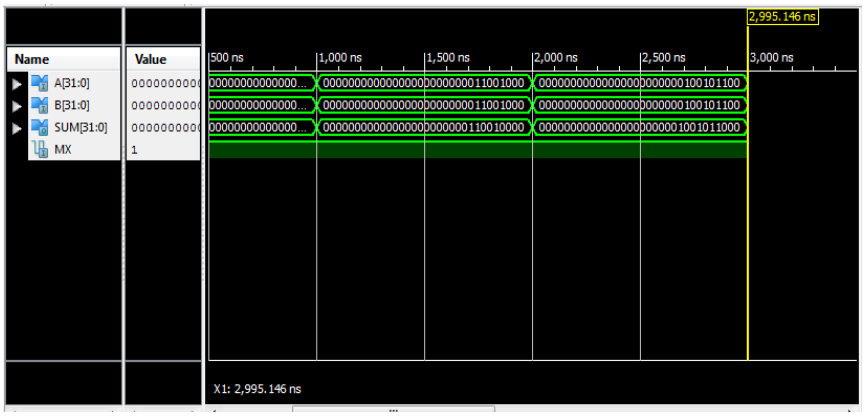

Fig10: simulated wave forms

Parameter	Conventional CLA Design	Approximate Adder Design
No of LUTs	64	44
Delay (ns)	44.366	2.590

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

Table 1: parameter comparison table

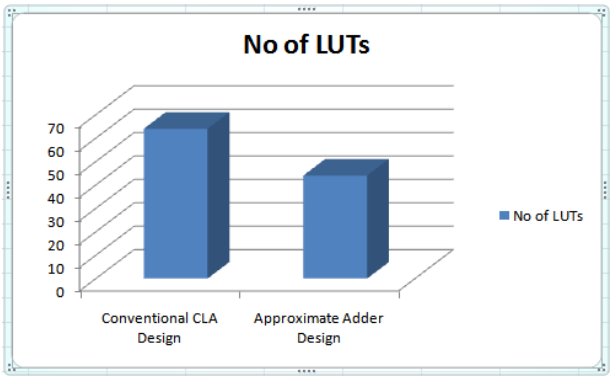

Fig11: LUT Comparison Bar Graph

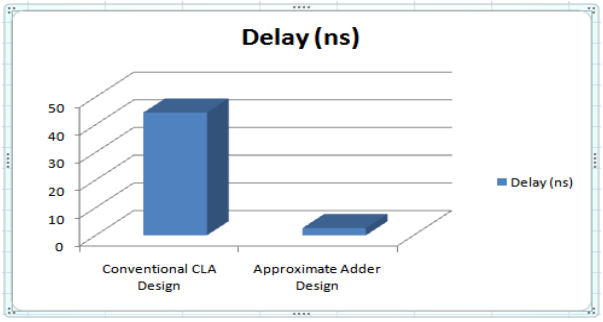

Fig12: Parameter Comparison Bar Graph

IV. CONCLUSION

In this paper, The proposed configurable carry look ahead adder is better than the existing traditional carry look ahead adder those results are shown in table 1.The proposed adder design 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 speedup with a small area and less power 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 area, power and delay without sacrificing accuracy. It was also found that the quality requirements of the evaluated application were not compromised.

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.

ISSN: 2278-4632
Vol-10 Issue-9 No.03 September 2020

[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 Microarchitecture (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 Comptuer-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]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.