Volume: 7 Issue: VII Month of publication: July 2019

DOI: http://doi.org/10.22214/ijraset.2019.7034

Design and Analysis of an Energy-Efficient Accuracy Configurable Adder

Richa Pathak[1], Amit Kumar[2]
[1]*Research Scholar, Microelectronics, Department of Electronics Engineering, Institute of Engineering and technology, Lucknow, Dr APJ Abdul Kalam Technical University, Lucknow India-226021*
[2]*Assistant Professor, Department of Electronics Engineering, Institute of Engineering and technology, Lucknow, Dr APJ Abdul Kalam Technical University, Lucknow India-226021*

Abstract: The approximate adders proposed in literature are dedicated for error- resilient applications only. These adders cannot be used in the domain where conventional adders are used or applications where we need to handle critical data with accuracy. So in this paper we proposed a Simple Accuracy Configurable Adder (SACA) that can be used in two different modes, Accurate Mode and Approximate Mode. The amount of error to be introduced at each stage is controlled with help of MUX. Further we evaluated the performance of Accuracy Configurable Adder in terms of delay, power, PDP, EDP and error characteristics.

SACA is 28.36% faster and consumes 30.11% less power than RCA. PDP is the figure of merit for digital logic circuits. The less is the value of PDP, the more Energy-efficient is circuit. SACA provides 47% less PDP and 62% less EDP as compared to RCA.

Keywords: Simple Accuracy Configurable Adder (SACA), Ripple-carry Adder (RCA), Error-resilient, Approximate Adder, Power Delay Product (PDP).

I. INTRODUCTION

Power restriction is a well-known challenge in advanced VLSI technologies. Low-power techniques for the traditional exact computing paradigm have been already comprehensively studied. A comparatively new trend is approximate computing, where errors are intentionally allowed in exchange for power reduction.

In numerous multimedia applications, such as audio, video and machine learning, occasional small errors are generally acceptable. Such error resilient applications are found in profusion in emerging applications and technologies. These Error-resilient applications are driving force for Approximate Computing. Topical studies show that very high extent of error resilience (≈83%) is prevalent in these applications.

Approximate computing has been extensively investigated at both the software level and hardware level over the past decade. Software techniques used for approximation skip algorithm-level reckoning, whereas hardware techniques revise designs at the circuit level. In approximate computing paradigm at the hardware level, the majority of work has been propounded on arithmetic units. Further, among arithmetic units, adders have pulled in the fervent interest for approximation. The main reason behind this is that in binary arithmetic, adders are the main component to perform arithmetic operations, that is, all arithmetic operations, such as addition, subtraction, multiplication, and division use adders as the basic building block. Besides arithmetic operations, adders are used to perform increment, decrement, and many similar operations. Therefore, being the most widely used fundamental data operators, adders have attracted a noteworthy attention for approximation. Since delay and power of adders increase rapidly with bit-width (N), for a marginal improvement in delay/power, we have to sacrifice an immense amount of power/delay. Consequently, at a micro architecture level of abstraction, adders have become the key delay/power bottleneck of digital systems. One possible way to overcome this situation is to approximate adders, that is, to sacrifice accuracy for delay and/or power. In some applications such as image processing or audio/video compression, the required accuracy might vary during run time. To meet the need for runtime accuracy adjustment, a series of designs are developed to implement accuracy-configurable approximation, which could be reconfigured online to save more power.

In this paper, we propose a new carry-prediction based accuracy configurable adder design: simple accuracy configurable adder (SACA). It is a simple design with considerably less area than CLA, which, to the best of our knowledge, has not been achieved in the past in ACAs. SACA inherits the advantages of all previous carry-prediction-based approaches: no error correction overhead, no data stall, and allowing graceful degradation.

II. LITERATURE REVIEW

We review a few representative works on ACA design and show the relation with our method. These designs can be generally categorized into two groups: error-correction-based configurations [1]–[3] and carry-prediction-based configurations [4], [5]. The main idea of an error-correction-based approach [1]–[3] is shown in Fig. 1. The scheme starts with an approximate adder (the dashed box), where the carry chain is shortened using separated sub-adders with truncated carry-in. In order to reduce the truncation error, the bit-width in some sub-adders contains redundancy. For example, subadder2 calculates the sum for only bits 8 and 9, but it is an 8-bit adder using bit [9 : 2] of the addends, 6 bits of which are redundant. Even with the redundancy, there is still residual error which is detected and corrected by additional circuits. In Fig. 1, the errors of subadder2 must be corrected by error-correction2 before the errors of subadder3 are rectified by error-correction3. As such, the configuration progression always starts with small accuracy improvements. The redundancy and error detection/correction incur large area overhead. Since the error correction circuits are usually pipelined, an accurate computation may take multiple clock cycles and could stall the entire data path, depending on the addend values. The frame work of carry-prediction-based methods [4],[5] is shown in Fig. 2. These schemes start with an accurate adder design, which is formed by chaining a set of sub-adders. Each sub-adder comes with a fast but approximated carry prediction circuit. By selecting between the carry-out from sub-adder or carry prediction, the overall accuracy can be configured to different levels. Such an approach does not need error detection/correction circuitry. Moreover, the configuration of higher bits is independent of lower bits. This leads to fast convergence or graceful degradation in the progression of configurations. In GDA [4], the sub-adders are CRA designs, while the carry-prediction circuit is similar to the carry look-ahead part of CLA. Furthermore, its carry prediction can be configured to different accuracy levels. However, the complicated carry prediction induces large area overhead. The RAP-CLA scheme [5] uses CLA for its baseline, where the carry-ahead of each bit is computed directly from the addends of all of its lower bits. Its carry prediction reuses a part of the look-ahead circuit rather than building extra dedicated prediction circuitry, and hence is more area-efficient than GDA. But its baseline is much more expensive than GDA. Our design is a carry-prediction-based approach. Its sub-adders are CRA instead of expensive CLA as in RAP-CLA. Its carry prediction also reuses part of the sub-adders rather than having dedicated prediction circuitry. As such, it avoids the disadvantages of both GDA and RAP-CLA.

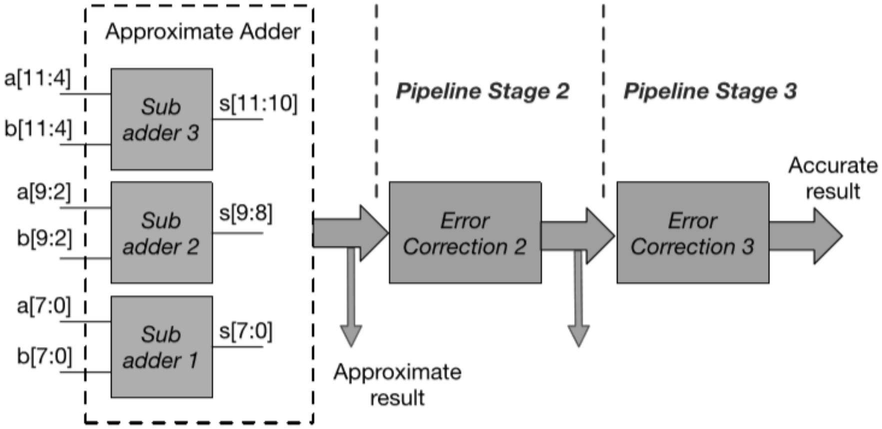

Fig. 1 Error-correction-based configurable adder

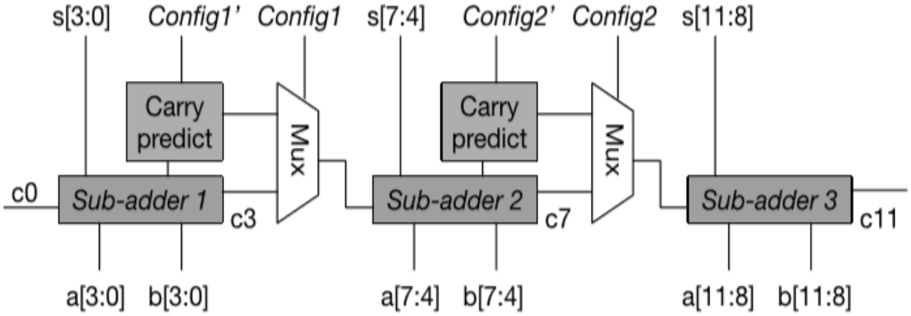

Fig. 2 Carry-prediction-based configurable adder

III.SIMPLE ACCURACY CONFIGURABLE ADDER DESIGN

A. Preliminaries

An N-bit adder performs operation on two addends $A = (a_N, a_{N-1}, \dots. a_i, \dots. a_1)$ and $B = (b_N, b_{N-1}, \dots. b_i, \dots. b_1)$. For bit i, its carry-in is c_{i-1} and its carry-out is c_i. Defining the carry generate bit $g_i = a_i . b_i$, propagate bit $p_i = a_i \oplus b_i$, and kill bit $k_i = \overline{a}_i . \overline{b}_i$, the conventional full adder computes the sum s_i and carry c_i according to

$$s_i = p_i \oplus c_{i-1} \tag{1}$$

$$c_i = g_i + p_i . c_{i-1} \tag{2}$$

A gate level schematic of a conventional full adder is provided in Fig. 4(a). By applying equation (2) repetitively, we get

$$c_i = g_i + p_i . g_{i-1} + \dots\dots + g_1 \prod_{k=2}^{i} p_k + c_0 \prod_{k=1}^{i} p_k \tag{3}$$

This equation implies that c_i can be obtained directly from g and p of all bits, without waiting for the c of its lower bits to be calculated.

B. Simple Accuracy Configurable Adder

In Accuracy-Configurable, an N-bit adder is composed of K segments of L-bit sub- adders, where $K = \lfloor N/L \rfloor$(see Fig.2). Each sub-adder is almost the same as CRA except that the MSB of a sub-adder, which is bit i, provides a carry prediction as

$$c_i^{pr} = g_i$$

The LSB of the higher bit sub-adder, which is bit i +1, its carry-out c_{i+1} can be obtained either by the traditional

$$c_{i+1} = g_{i+1} + p_{i+1} . c_i, \tag{4}$$

or by predicting the carry as

$$c_{i+1} = g_{i+1} + p_{i+1} . c_i^{pr} = g_{i+1} + p_{i+1} . g_i \tag{5}$$

The selection between the two options is conceived using MUXes as in Fig. 3 and the multiplexer (MUX) selection result is denoted as $\hat{c}_i$. Comparing (5) with (3), we can see that the carry prediction is a pruning-based approximation to carry calculation. Therefore$\hat{c}_i$, can be configured to either accurate mode or approximate mode, that is

$$\hat{c}_i \leftarrow \begin{cases} c_i^{pr}, & if\ approximate\ mode \\ c_i, & if\ accurate\ mode \end{cases} \tag{6}$$

It should be noted that the carry prediction c_i^{pr} reuses g_i in an existing full adder instead of introducing an additional dedicated circuit as in [46] or Fig.4. This prediction scheme makes an unpretentious modification to the traditional full adder, as shown in Fig. 4(b). One can connect $\hat{c}_i$ to its higher bit $i + 1$ to calculate both carry $c_{i+1}$and sums_{i+1}, as in GDA [4] and RAP-CLA [5]. We suggest an improvement over this approach by another simple change as in Fig. 4(c), where S_{i+1} is based on c_i instead of $\hat{c}_i$. Such approach can help reduce the error rate in outputs when an incorrect carry is propagated. Because the sum keeps accurate and the carry will not be propagated when addends are exactly the same. Moreover, out of all four configurations of sum/carry calculation by approximate/accurate carry-in, the most meaningful way is to have sum bit calculated by accurate carry and make carry bit configurable. Therefore, sum $s_{i+1}$is calculated directly by accurate carry c_i without the option of c_i^{pr} . Applying this in ACA as in Fig. 3, in the approximation mode, computing S_{j+1} from c_j can still limit the critical path to be between cprdt $i - 1$ and S_{j+1}, but has higher accuracy than computing $S_{j+1}$from $\hat{c}_j$. Compared to sum computation in GDA and RAP-CLA, this technique improves accuracy with almost no additional overhead. Compared to CRA, the overhead of ACA is merely the MUXes, which is almost the minimum possible for configurable adders.

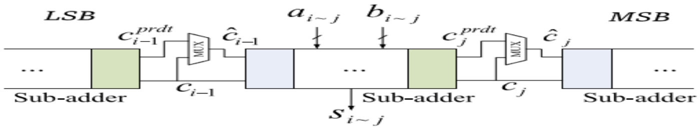

Fig. 3 Design of SACA

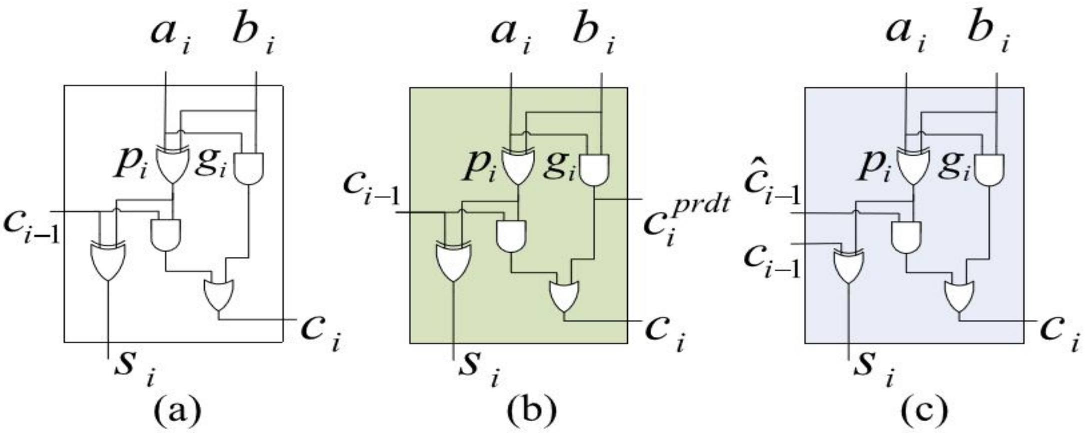

Fig. 4 (a) Conventional full adder. (b) Carry-out selectable full adder. (c) Carry-in configurable full adder.

IV. MATRICES BASED GENERALIZED APPROACH FOR ERROR ANALYSIS

To smooth the progress of the error analysis and make it generic for any AA, we redefine the aforementioned method using matrix theory. This step would allow us to do the error analysis of any AA by simply swapping three matrices obtained from the truth table of the specified AA. The following steps describe this approach:

1) We start by forming a 1x8 matrix $M = [m_1, m_2, \ldots, m_8]$, such that $m_i = 1$ if $Cout$ is "1" AND the case is a Success else $m_i = 0$ to discard the failure cases for all possible 8 cases of the truth table.
2) In a similar way, we declare another 1x8 matrix $K = [k_1, k_2, \ldots, k_8]$, such that k_i= 1 if C_{out} is "0" AND the case is a Success else k_i = 0.
3) Similarly, we obtain a third matrix $L = [l_1, l, \ldots, l_8]$, such that $l_i = 1$ if the case is a Success else $l_i = 0$. These three matrices will remain constant irrespective of the number of bits of a particular AA.
4) The next step is to evaluate probabilities of occurrence of all 8 possible outcomes of the truth table and use them to form the 1x8 Input Probability Matrix (IPM) as follow:

$$IPM = [P(\overline{A_i}).P(\overline{B_i}).P(\overline{C_c} \cap S), P(\overline{A_i}).P(\overline{B_i}).P(C_c \cap S), P(\overline{A_i}).P(B_i).P(\overline{C_c} \cap S), P(\overline{A_i}).P(B_i).P(C_c \cap S), P(A_i).P(\overline{B_i}).P(\overline{C_c} \cap S), P(A_i).P(\overline{B_i}).P(C_c \cap S), P(A_i).P(B_i).P(\overline{C_c} \cap S), P(A_i).P(B_i).P(C_c \cap S)] \tag{7}$$

5) Now, we can just use the dot product between M, K and IPM matrices to obtain the carry out probability:

$$P(C_n \cap S) = [IPM].[M] \tag{8}$$

$$P(\overline{C_n} \cap S) = [IPM].[K] \tag{9}$$

6) Steps 4 and 5 are repeated $N - 1$ times in case of a N bit adder. For every iteration, we use the evaluated carry probabilities from current stage to estimate the ones for the next stage as explained while describing Equation 6.
7) After $N - 1$ iterations, the probability of success and error can be evaluated by dot product as follow:

$$P(S) = [IPM].[L] \tag{10}$$

$$P(E) = 1 - P(S) \tag{11}$$

V. SIMULATION RESULTS

We implemented SACA using S- Edit of Tanner EDA Tool on 32nm technology parameters and supply voltage of 0.9Volts. We then analyzed the delay and power of implanted adders with the help to Tspice and obtained output waveforms with the help of W-Edit. For error analysis we have used MATLAB.

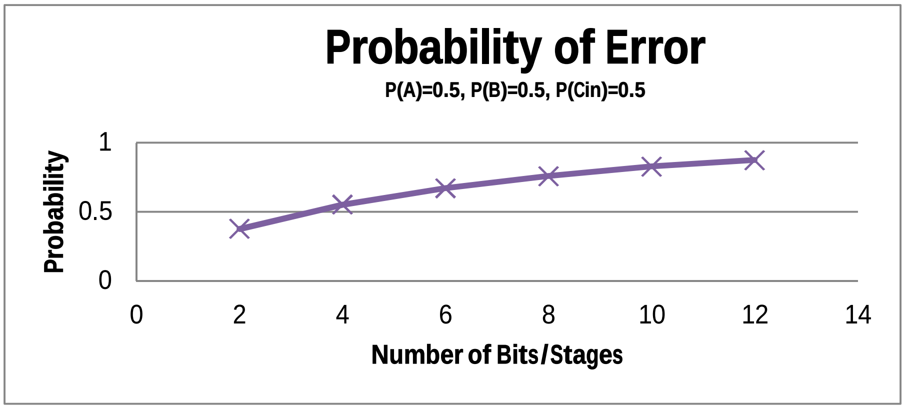

Fig. 5 P(E) when P(A)=0.5, P(B)=0.5, P(Cin)=0.5

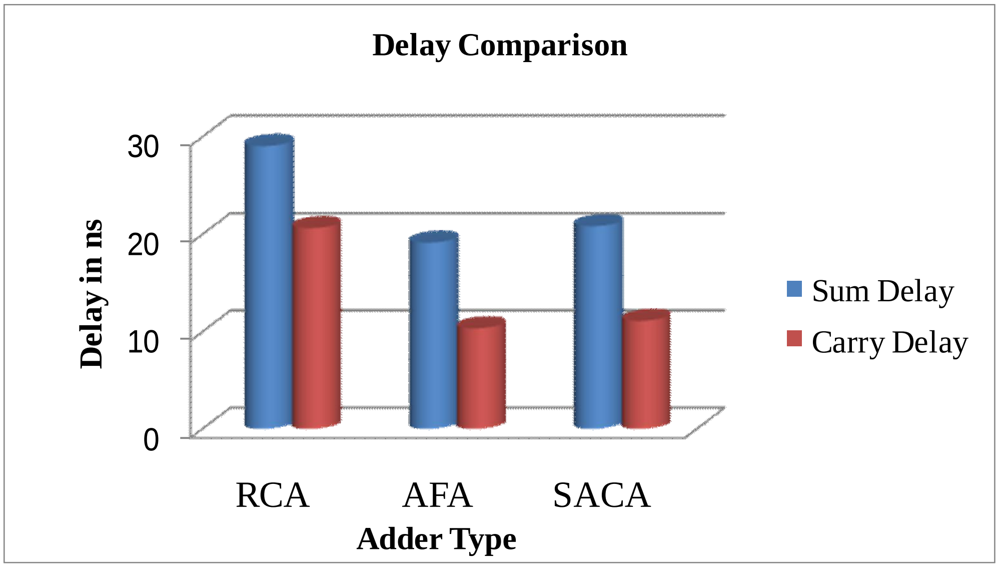

Fig. 6 Delay Analysis

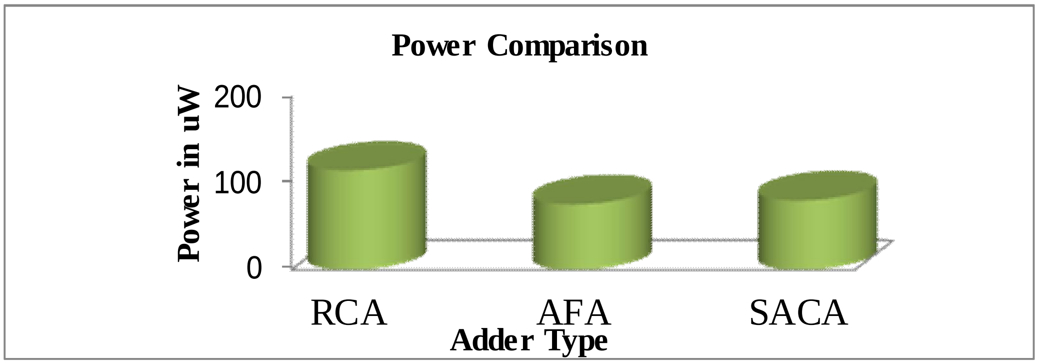

Fig. 7 Power Analysis

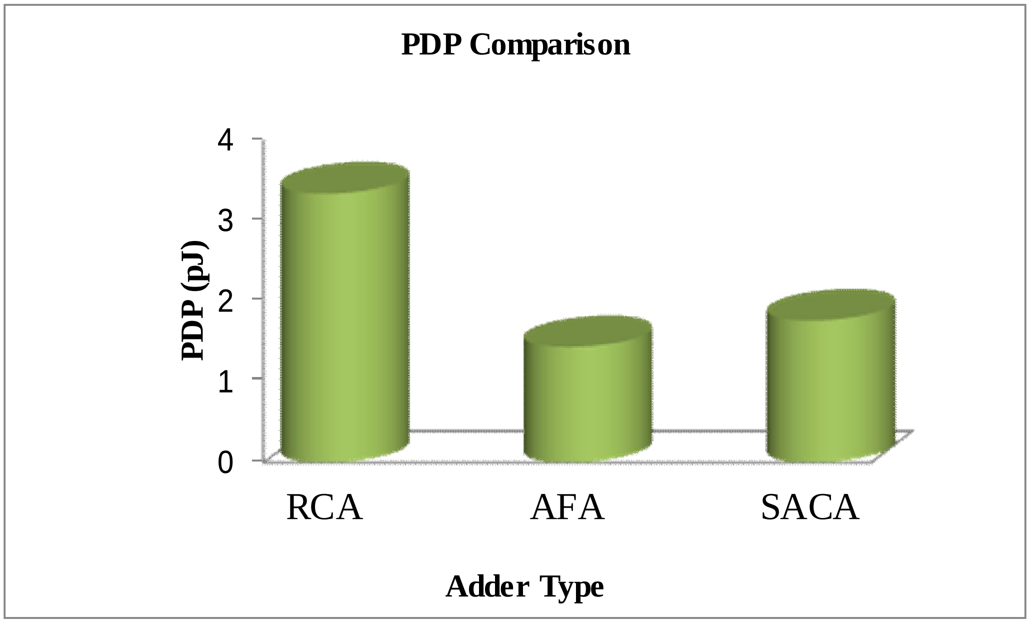

Fig. 8 PDP Analysis

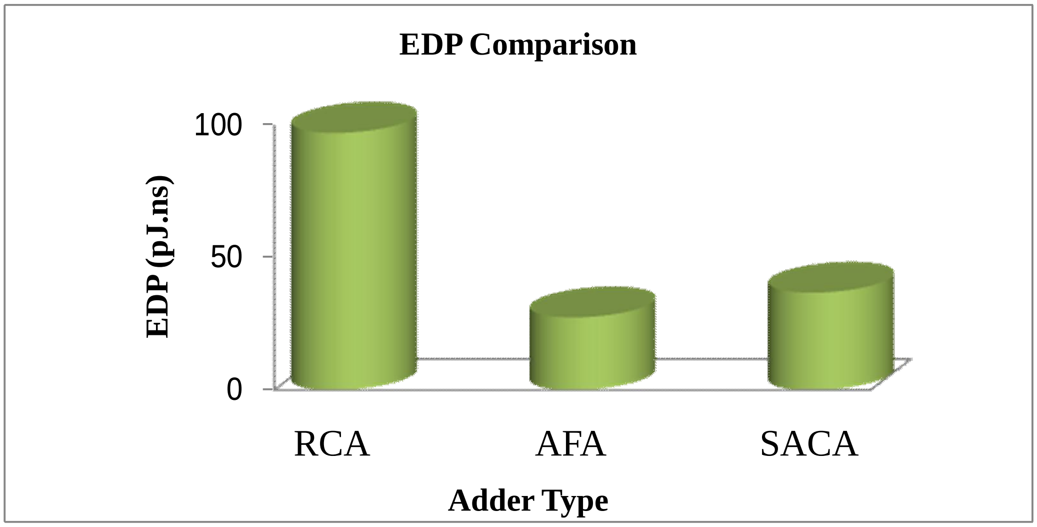

Fig. 9 EDP Analysis

From above analysis we found out that SACA is 28.36% faster and consumes 30.11% less power than RCA. PDP is the figure of merit for digital logic circuits. The less is the value of PDP, the more Energy-efficient is circuit. ACA provides 47% less PDP and 62% less EDP as compared to RCA.

VI. CONCLUSION

In this paper we proposed SACA i.e. simple accuracy configurable adder and also presented a comparative analysis with Ripple Carry Adder (RCA) and Approximate Full Adder (AFA) in terms of parameters like Error Rate, Power, Delay, PDP, EDP. To overcome the limitations of existing analytical techniques for error rate evaluation, we have presented a generic matrices based recursive approach for error analysis which can be used for the error evaluation of any multi-bit approximate adder.

To conclude we can say that the values for AFA and SACA for different parameters are almost comparable but AFA is slightly better than SACA. But SACA is very much efficient when compared to RCA. AFA achieves better performance than SACA at the cost of Error. SACA provides us the flexibility to be used in accurate mode or approximate mode because its error rate is run time configurable. So in highly error resilient applications we should use AFA but in all other applications where we want to have control over the accuracy we should use SACA.

REFERENCES

[1] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proc. Design Autom. Conf. (DAC), 2012, pp. 820–825.

[2] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in Proc. Design Autom. Conf. (DAC), 2015, pp. 1–6.

[3] V. Benara and S. Purini, "Accurus: A fast convergence technique for accuracy configurable approximate adder circuits," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, Jul. 2016, pp. 577–582.

[4] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration oriented approximate adder design and its application," in Proc. Int. Conf. Comput.-Aided Design (ICCAD), 2013, pp. 48–54.

[5] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," IEEE Trans. Circuits Syst. II, Exp. Briefs, to be published.

[6] Sunil Dutt, Sukumar Nandi, and Gaurav Trivedi, "Analysis and Design of Adders for Approximate Computing", ACM C Vol. 17, No. 2, Article 40. Publication date: December 2017.

[7] Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders." IEEE Trans. Comput., 62(9):1760{1771, 2013.

[8] K. Du, P. Varman, and K. Mohanram. High performance reliable variable latency carry select addition. In Proceedings of the 2012 Design, Automation Test in Europe Conference Exhibition (DATE 2012), pages 1257–1262, March 2012.