Energy efficient high-performance approximate adders for imprecision-tolerant signal processing applications

K Komathy Vanitha¹ and S Anila²

¹Assistant Professor, Department of Electronics and Communication Engineering, CSI College of Engineering, Ketti, The Nilgiris, Tamilnadu, India
²Associate Professor, Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology, Coimbatore, Tamilnadu, India
Email: vanitha.kkv@gmail.com

Abstract: The trade-off between Delay and Power consumption has become a major concern as process technology reached less than 10 nm proximity in the modern Very Large-Scale Integration (VLSI) technology. This trade-off can be compensated with accuracy and is vanquished by the development of Approximate Computing (AC). In this paper, six diverse Approximate Adders (AAs) have been proposed based on logic complexity reduction at the transistor level. Simulation results reveal that the Proposed AAs has a significant amount of Power and Delay savings, lesser Power-Delay Product (PDP). The Proposed AAs: PA1, PA3, PA5, PA3 exhibits 12.85 %, 41.59 %, 72.05 %, 1.91 % lesser power than the Existing AAs EAA1, EAA5, EAA6, EAA9 respectively. The Proposed AAs: PA2, PA3 incorporates 37.5 %, 54.5%, of lesser number of transistors compared to Existing AAs: EAA5, EAA9 whereas PA4, PA5 incorporates 40 % of reduction in the number of transistors compared to Existing AAs: EAA6, EAA8. These results are promising for high performance and energy efficient systems for error-resilient applications such as multimedia and signal processing where a slightly degraded output quality is acceptable, which could lead to significant power reduction.

1. Introduction
In the past few decades, the feature size of transistors has decreased exponentially, as governed by Moore’s law [1]. It has become extremely challenging to improve circuit performance in terms of Power and Delay as the physical dimensions of Complementary Metal Oxide Semiconductor (CMOS) shrink to a few tens of nanometers. [2]. A promising class of low-power design techniques, which could be collectively classified under the paradigm of “Approximate Computing”, has received significant attention. Approximate computing has emerged as an attractive paradigm to achieve low power and high-speed designs with reduced complexity These techniques relax the conventional requirement of perfect equivalence between the specification and hardware implementation [3].

Method in which circuits that are not implemented exactly according to the exact design, but are simplified in order to reduce power and delay is called Approximate Computing (AC). The faults that occur in simplified circuits are considered to be tolerable, which is typical of error-tolerant application fields like multimedia and classification and data mining. Applications based on NNs have proven to be highly error resilient [4]. In many applications related to human perception such as image and video processing, the numerical exactness of the output can be relaxed [5] Moore’s law, along with Dennard...
scaling has resulted exponential increase in computational performance at constant energy consumption and transistor cost in the field of digital signal processing. The backbone of different multimedia applications utilized in portable devices is digital signal processing (DSP) blocks [5]. It is known that the computational element Adder is the basic building blocks in multimedia systems [6]. A significant portion of the total power and area is consumed by these computational elements. So, the reduction of power, Time and Area in basic computational elements will definitely have a tremendous impact on the overall reduction of the above parameters of the system.

2. Why Approximate Adders AAs?
One of the most basic arithmetic circuits in a machine is the adder, where the binary integers added together. The Ripple Carry Adder (RCA) and the Carry Look Ahead adder are two basic designs (CLA). The time it takes for a CLA is important. In N or O (log (n)), approximately logarithmic an RCA’s delay is slightly shorter.[1] The simple CLA structure is inefficient for an adder with a width of 32 bits or more, due to the wide fan in and fan out of the constituent gates, which reduces speed and increases circuit area and power consumption. As a result, several levels of look ahead structures have been proposed to construct a wide width adder [7], also known as a parallel adder. Taking advantage of certain applications' error tolerance, such as multimedia, image processing, and machine learning, a number of recent papers have suggested designing estimated adders that sometimes produce unreliable results in return for reduced delay and power consumption. [8]. An approximate multi bit adder, for example, can ignore the carry chain and thus perform addition operations in parallel across all of its sub adders. To ensure a safe operation, the chip must operate within a predetermined power budget. To avoid excessive power consumption, a portion of the chip must stay inactive, this inactive portion is referred to as “Dark Silicon” [9]. A study has shown that the area of dark silicon may reach up to more than 50% for an 8-nm technology [9]. ITRS projections have anticipated that by 2020, developers would look up to 90% of dark silicon, implying that just 10% of the IC are valuable at some random time when high working frequencies are applied [10]. This suggests an increasing challenge to enhance circuit overall performance and power efficiency through the usage of usual technologies. New design methodologies have been investigated to address this issue, including multi-core architectures, heterogeneous integration, and AC [11]. The use of probabilistic imprecise arithmetic or deterministic approximate logic commonly used in soft additions; however, it dissipates little power [12]. Due to its accumulative existence over several cycles, the implementation of the segmentation method causes a major error in meta function computation [13]. Approximate logic has been used to build bio-inspired lower part OR adder (LOA) The LOA is the most time consuming and hence Improved dynamic segmentation with multi cycle error correction approach (DSEC) enhances accuracy over a wide voltage range to resolve the above problem. Logic complexity reduction for adders at bit level provides better power savings over conventional low power design techniques. Five unique simplified implementations of Mirror Adders have been proposed based on logic complexity reduction [14]. Different methods were proposed based on logic complexity reduction at the transistor level [13]. The Existing methods has a greater number of transistors resulting in higher power consumption and delay.

3. Existing and Proposed Approximate adders AAs
In this paper, six diverse Approximate Full Adders are proposed based on Boolean /Logic function reduction at transistor level. In comparison to existing designs, the proposed designs incorporate minimum number of transistors and provide better performance in terms of Power, Delay and PDP. The existing AAs: EAA1, EAA2, EAA3, EAA4, EAA5, EAA6, EAA7, EAA8, EAA9, EAA10 and the Proposed AAs:PA1, PA2, PA3, PA4, PA5, PA6 are described in the subtopics below.
3.1. Existing Approximate adders (EAA1- EAA10)
Two different methods have been considered in the design of Existing AAs [13],[14]. First method is designed such that sum is dependent on the Carry output and in the second method sum is independent of Carry output. The outputs of the various Existing AAs have been summarized in table 1.

Table 1. Truth table for Existing Approximate Adders.

| Inputs | Conventional adder Output | EAA1 | EAA2 | EAA3 | EAA4 | EAA5 | EAA6 | EAA7 | EAA8 | EAA9 | EAA10 |
|--------|---------------------------|------|------|------|------|------|------|------|------|------|-------|
| A      | B                          | C    | Sum  | Carry| Sum  | Carry| Sum  | Carry| Sum  | Carry| Sum  | Carry |
| 0      | 0                          | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 0      | 0                          | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 0      | 0                          | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 0      | 1                          | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 0      | 1                          | 1    | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 1      | 0                          | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 1      | 0                          | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 1      | 1                          | 0    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     |
| 1      | 1                          | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1     |

*x* - Denotes the erroneous output

3.2. Proposed approximate adders (PA1- PA6)
To develop better approximate adders, two approaches have been proposed, in the first method approximation is done in the sum alone and carry remains constant. In the Second method approximation in carry alone is done and Sum remains constant. Six approximate adders have been proposed and explained in the following subsections

3.2.1. Proposed Approximate Adder PA1
In the Proposed Adder, out of 8 input combinations exact output is obtained for 5 input combinations in the sum and for 6 input combinations in the Carry. The Boolean functions of PA1 are as in equations (1), (2)

\[ \text{Sum}_{PA1} = (B' + C)A \]  
\[ \text{Cout}_{PA1} = B \]

3.2.2. Proposed Approximate Adder: PA2
In the Proposed Adder PA2 out of 8 input combinations exact output is obtained for 6 input combinations in the sum and exact output is obtained for all the 8 input combinations in case of carry. The Boolean functions of PA2 are as in equations (3), (4)

\[ \text{Sum}_{PA2} = BC + \text{Cout}' \]  
\[ \text{Cout}_{PA2} = AB + (A + B)C \]

3.2.3. Proposed Approximate Adder PA3
In the Proposed Adder PA3 out of 8 input combinations, exact output is obtained for 6 input combinations in the sum and exact output is obtained for all the 8 input combinations in case of Carry. The Boolean functions of PA3 are as in equations (5), (6)
\[ \text{Sum}_{PA3} = (A + B)\text{Cout}' \]  
\[ \text{Cout}_{PA3} = AB + (A + B)C \]

### 3.2.4. Proposed Approximate Adder PA4

The Proposed Adder PA4 introduces two errors in Sum and one error in carry. The Boolean functions of PA4 are as in equations (7), (8)

\[ \text{Sum}_{PA4} = (A + C)\text{Cout}' \]  
\[ \text{Cout}_{PA4} = AC + B \]

### 3.2.5. Proposed Approximate Adder PA5

In the Proposed Adder out of 8 input combinations exact output is obtained for 6 input combinations in the sum and for 7 input combinations in the Carry. The Boolean functions of PA5 are as in equations (9), (10)

\[ \text{Sum}_{PA5} = (A + B)\text{Cout}' \]  
\[ \text{Cout}_{PA5} = AB + C \]

### 3.2.6. Proposed Approximate Adder PA6

In the Proposed Adder exact output is obtained for all the 8 input combinations in case of Carry and exact output is obtained for 7 input combinations in the sum. Compared to all the above proposed methods PA6 has less probability of error. The Boolean functions of PA6 are as in equations (11), (12)

\[ \text{Sum}_{PA6} = (A + B)C' + AB + A'B'C \]  
\[ \text{Cout}_{PA6} = AB + (A + B) \]

The truth table for the Proposed approximate adders are summarized in table 2

| Inputs  | A  | B  | Cin | Sum | Carry | Conventional adder Output | Proposed AAs Output |
|---------|----|----|-----|-----|-------|---------------------------|---------------------|
| PM1     | 0  | 0  | 0   | 0   | 0     | 0                         | 0                   |
| PM2     | 0  | 0  | 1   | 0   | 0     | 0                         | 1                   |
| PM3     | 0  | 1  | 0   | 0   | 1     | 0                         | 0                   |
| PM4     | 0  | 1  | 0   | 1   | 0     | 0                         | 0                   |
| PM5     | 0  | 1  | 1   | 1   | 0     | 0                         | 0                   |
| PM6     | 0  | 1  | 1   | 1   | 1     | 0                         | 0                   |
| PM7     | 0  | 1  | 1   | 1   | 1     | 0                         | 0                   |

\(\times\) - Denotes the erroneous output

### 4. Results and Comparisons:

The designs of the proposed approximate adders are analyzed in terms of Power, Delay, Power Delay Product (PDP) and Area in terms of number of transistors. The Existing AAs and the Proposed AAs are simulated in Tanner v2019.2 and then the Power consumption and Delay is calculated using T-spice tool of Tanner.
4.1. Analysis and Comparison of the Proposed AAs:
Because of the vast number of transistors, the conventional mirror adder has the highest power consumption. The comparison of Power, Delay, Area, and the number of errors are given in Table 3 for the Existing AAs (EAA1–EAA10), and Proposed AAs (PA1–PA6). From the results, it is observed that the proposed adder PA1 that has 5 errors has reduced number of transistors, lesser power, and delay compared with the Existing AAs: EAA1, EAA3. There exists two errors in the sum of proposed AAs: PA2, PA3 and consumes less power and delay and it has reduced number of transistors compared to Existing AAs: EAA5, EAA9. The proposed adders PA4, PA5 has three errors and consumes less power compared with EAA6 and EAA8 and with a trade-off in delay. The proposed adder PA6 has one error in the sum with reduced number of transistors with trade-off in the power.

### Table 3. Comparison of Existing and Proposed AAs.

| Existing AAs | Number errors in Sum and Carry | Area /Number of Transistors | Power in nW | Delay in ns | Power Delay product (PDP) in J |
|--------------|--------------------------------|-----------------------------|-------------|-------------|-------------------------------|
| EAA1         | 5                              | 10                          | 149.08      | 100.2       | 149.38                        |
| EAA2         | 4                              | 12                          | 299.05      | 150.29      | 449.44                        |
| EAA3         | 5                              | 8                           | 168         | 149         | 250.32                        |
| EAA4         | 1                              | 18                          | 442         | 24.52       | 108.38                        |
| EAA5         | 2                              | 22                          | 395         | 99.94       | 394.76                        |
| EAA6         | 3                              | 18                          | 404         | 50.5        | 204.02                        |
| EAA7         | 1                              | 36                          | 266         | 25.52       | 67.88                         |
| EAA8         | 3                              | 18                          | 457         | 40.62       | 185.63                        |
| EAA9         | 2                              | 28                          | 264         | 123.5       | 326.04                        |
| EAA10        | 1                              | 36                          | 442         | 24.51       | 108.33                        |

| Proposed AAs | Number errors in Sum and Carry | Area /Number of Transistors | Power in nW | Delay in ns | Power Delay product (PDP) in J |
|--------------|--------------------------------|-----------------------------|-------------|-------------|-------------------------------|
| PA1          | 5                              | 8                           | 131         | 23.6        | 30.92                         |
| PA2          | 2                              | 16                          | 296         | 22          | 65.12                         |
| PA3          | 2                              | 16                          | 259         | 22.6        | 58.53                         |
| PA4          | 3                              | 12                          | 279         | 100         | 279.00                        |
| PA5          | 3                              | 12                          | 190         | 73          | 138.70                        |
| PA6          | 1                              | 32                          | 450         | 24.8        | 111.60                        |

4.2. Graphical Representation
Figure 1 represents the Power, figure 2 represents the Delay and figure 3 represents the Area in terms of Number of transistors for various existing/proposed adders. PA6 has the highest power consumption of all the proposed adders whereas PA1 has the lowest power consumption due to reduced number of transistors. From Fig 3 it is evident that the Proposed AAs have reduced number of transistors which will pave way for area reduction.
4.3. Quantitative Accuracy Metrics

The quantitative Accuracy Metrics such as Error Rate (ER), Error Distance (ED), Pass Rate (PR), Mean Error Distance (MED), Normalized Error Distance (NED) of the Proposed Approximate adders are calculated as given in Equation (13), (14), (15), (16), (17), and the values are represented in table 4.

\[
ER = \frac{\text{Number of Approximate outputs}}{\text{Total Number of outputs}}
\]

\[
ED = \text{Accurate output} - \text{Approximate Output}
\]

\[
PR = \frac{\text{Number of Accurate outputs}}{\text{Total Number of outputs}}
\]

\[
MED = \frac{\text{Average of the Error Distance}}{\text{Total Number of outputs}}
\]

\[
NED = \frac{MED}{\text{Maximum possible error}}
\]
Table 4. Quantitative Error Metrics of Proposed AAs.

| Proposed AAs | ED | ER | PR | MED | NED |
|--------------|----|----|----|-----|-----|
| PA1          | 5  | 0.63 | 0.375 | 0.48 | 0.06 |
| PA2          | 2  | 0.25 | 0.75 | 0.24 | 0.03 |
| PA3          | 2  | 0.25 | 0.75 | 0.24 | 0.03 |
| PA4          | 3  | 0.38 | 0.625 | 0.36 | 0.045 |
| PA5          | 3  | 0.38 | 0.625 | 0.36 | 0.045 |
| PA6          | 1  | 0.13 | 0.875 | 0.09 | 0.011 |

5. Conclusion
This study presents six imprecise or approximate adders that can be used to balance Power and Delay in error-tolerant DSP systems. The proposed Adders exhibits higher efficiency in terms of Power, Delay and PDP compared to existing approximate adders. Reduction in the number of transistors of the proposed adders design would considerably reduce the circuit area. Hence these proposed approximate adders can be used for error resilient applications such as multimedia and signal processing which lead to high computing applications.

6. References
[1] Moore G E 1965 Cramming more components onto integrated circuits *Electronics* **38** 114
[2] Han J and Orshansky M 2013 Approximate Computing: An Emerging Paradigm for Energy-Efficient Design 18th IEEE European Test Symposium (ETS)
[3] Debabrata Mohapatra, Vinay K, Chippa, and Raghunathan and Kaushik Roy 2011 Design of Voltage-Scalable Meta-Functions for Approximate Computing Design Automation and Test in Europe-Grenoble
[4] Vojtech, Shakib Sarwar, Sekanina, ZdenekVasicek and Kaushik Roy 2016 Design of Power Efficient Approximate Multipliers for Approximate Artificial Neural Networks CCAD’ pp 1607-10
[5] HassaanSaadat, HarisJavaidb and Sri Parameswaran 2019 Approximate Integer and Floating-Point Dividers with Near Zero Error Bias DAC ’19 Proceedings of the 56th Annual Design Automation Conference pp 1-6
[6] Yuan-Ho Chen 2015 Area-Efficient Fixed Width Squarer with Dynamic Error Compensation Circuit IEEE Transactions on Circuits and Systems II Express Briefs pp 851 – 55
[7] Vasicek Z and Sekanina L 2015 Evolutionary approach to approximate digital circuits design IEEE Transactions on Evolutionary Computation 19 pp 432–444
[8] Mahdiani H R, Ahmadi A, Fakhraie S M and Lucas C 2010 Bio-inspired Imprecise computational blocks for efficient VLSI implementation of soft computing applications IEEE Trans. Circuits Syst. pp 57850–62
[9] Esmailzadeh H, Blem E, Amant R S, Sankaralingam K, and Burger D 2011 Dark silicon and the end of multicore scaling Proc. of IEEE International Symposium on Computer Architecture pp 365–376
[10] Semiconductor Industry Association 2011 *International technology roadmap for semiconductors* (ITRS)
[11] Shafique M, Garg S, Henkel J and Marculescu D 2014 The EDA Challenges in the dark silicon era: Temperature reliability and Variability perspectives Proc. DAC pp 1–6
[12] onglan H,Jiang, Jie Han, Abrizo F and Lombardi 2010 A comparative review and Evaluation of approximate adders Transactions on Circuits and Systems **57** pp 850-62.
[13] Anushaa G and Deepa P 2020 Microprocessors and Microsystems Design of approximate adders and multipliers for error tolerant image processing *Microprocessors and Microsystems* 72

[14] Gupta V, Mohaptra D, Raghunathan A and Roy K 2013 Low power digital signal processing using approximate adders *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* pp 3287–97.

[15] Venkataramani S, Chakradhar S T, Roy K and Raghunathan A 2015 Approximate computing and the quest for computing efficiency 52nd ACM/EDAC/IEEE Design Automation Conference

[16] Hardavellas N, Ferdman M, Falsafi B and Ailamaki 2011 Toward dark silicon in servers *IEEE Micro* 31 pp 6–15.

[17] Vinay Kumar, Ankit Singh, Shubham Upadhyay and Binod Kumar 2019 Power-Delay-Error Efficient Approximate Adder for Error-Resilient Applications *Journal of Circuits, Systems, and Computers* 28 1950171

[18] Jothin R, Peer Mohamed M and Vasanthanayaki C 2020 High performance compact energy efficient error tolerant adders and multipliers for 16-bit image processing applications, *Microprocessors and Microsystems* 78 103237

[19] Thiruvenagadom C, Palanivelan M, Senthil Kumar and Jayasankar T 2020 Low power approximate adder based repetitive iteration cord (LP-ARICO) algorithm for high-speed applications *Microprocessors and Microsystems* 78 103260