A Comparative Review and Evaluation of Approximate Adders

Honglan Jiang
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta T6G 2V4, Canada
honglan@ualberta.ca

Jie Han
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta T6G 2V4, Canada
jhan8@ualberta.ca

Fabrizio Lombardi
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, USA
lombardi@ece.neu.edu

ABSTRACT

As an important arithmetic module, the adder plays a key role in determining the speed and power consumption of a digital signal processing (DSP) system. The demands of high speed and power efficiency as well as the fault tolerance nature of some applications have promoted the development of approximate adders. This paper reviews current approximate adder designs and provides a comparative evaluation in terms of both error and circuit characteristics. Simulation results show that the equal segmentation adder (ESA) is the most hardware-efficient design, but it has the lowest accuracy in terms of error rate (ER) and mean relative error distance (MRED). The error-tolerant adder type II (ETAII), the speculative carry select adder (SCSA) and the accuracy-configurable approximate adder (ACAA) are equally accurate (provided that the same parameters are used), however ETATII incurs the lowest power-delay-product (PDP) among them. The almost correct adder (ACA) is the most power consuming scheme with a moderate accuracy. The lower-part-OR adder (LOA) is the slowest, but it is highly efficient in power dissipation.

Categories and Subject Descriptors

B.2.0 [**Arithmetic and Logic Structures**]: General; B.6.1 [**Logic Design**]: Design Styles—*combinational logic*; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*VLSI*; B.8.0 [**Performance and Reliability**]: General

Keywords

Approximate Computing; Adder; Power; Accuracy.

1. INTRODUCTION

As the physical dimensions of CMOS scale down to a few tens of nanometers, it has been increasingly difficult to improve circuit performance and/or to enhance power efficiency. Approximate computing has been advocated as a new approach to saving area and power dissipation, as well as increasing performance at a limited loss in accuracy [3]. While computation errors are in general not desirable, applications such as multimedia (image, audio and video) processing, wireless communications, recognition, and data mining are tolerant to some errors. Due to the statistical/probabilistic nature of these applications, small errors in computation would not impose noticeable degradation in performance [22].

Generally, there are two types of methodologies for improving speed and power efficiency by approximation. The first methodology uses a voltage-over-scaling (VOS) technique for CMOS circuits to save power [4,15,19]. The second methodology is based on redesigning a logic circuit into an approximate version. While the VOS technique is applicable to most circuits for error-tolerant applications, an approximate redesign pertains to the functionalities of different logic circuits. Approximately redesigned adders (simply referred to as approximate adders) are reviewed and a comparative evaluation is performed in this paper.

2. REVIEW

Adders are utilized for calculating the addition (or sum) of two binary numbers. Two common types of adders are the ripple-carry adder (RCA) and the carry lookahead adder (CLA) [9, 21]. In an n-bit RCA, n 1-bit full adders (FAs) are cascaded; the carry of each FA is propagated to the next FA, thus the delay of RCA grows in proportion to n (or $O(n)$). An n-bit CLA consists of n SPGs, which operate in parallel to produce the sum, generate ($g_i = a_i b_i$) and propagate ($p_i = a_i + b_i$) signals, and connected to a carry lookahead generator. For CLA, all carries are generated directly by the carry lookahead generator using only the generate and propagate signals, so the delay of CLA is logarithmic in n (or $O(log(n))$), thus significantly shorter than that of RCA. However, CLA requires larger circuit area and higher power dissipation. The carry lookahead generator becomes very complex for large n. The area complexity of CLA is $O(nlog(n))$ when the fan-in and fan-out of the constituent gates are fixed [16].

Many approximation schemes have been proposed by reducing the critical path and hardware complexity of the accurate adder. An early methodology is based on a speculative operation [16, 23]. In an n-bit speculative adder, each sum bit is predicted by its previous k less significant bits

GLSVLSI'15, May 20–22, 2015, Pittsburgh, PA, USA.

http://dx.doi.org/10.1145/2742060.2743760.

(LSBs) ($k < n$). A speculative design makes an adder significantly faster than the conventional design. Segmented adders are proposed in [7,19,27]. An n-bit segmented adder is implemented by several smaller adders operating in parallel. Hence, the carry propagation chain is truncated into shorter segments. Segmentation is also utilized in [1,5,8,10, 12,25], but their carry inputs for each sub-adder are selected differently. This type of adder is referred to as a carry select adder. Another method for reducing the critical path delay and power dissipation of a conventional adder is by approximating the full adder [2,17,20,24]; the approximate adder is usually applied to the LSBs of an accurate adder. In the sequel, the approximate adders are divided into four categories.

2.1 Speculative Adders

As the carry chain is significantly shorter than n in most practical cases, [23] has proposed an almost correct adder (ACA) based on the speculative adder design of [16]. In an n-bit ACA, k LSBs are used to predict the carry for each sum bit ($n > k$), as shown in Fig. 1. Therefore, the critical path delay is reduced to $O(log(k))$ (for a parallel implementation such as CLA, the same below). As an example, four LSBs are used to calculate each carry bit in Fig. 1. As each carry bit needs a k-bit sub-carry generator in the design of [16], $(n-k)$ k-bit sub-carry generators are required in an n-bit adder and thus, the hardware overhead is rather high. This issue is solved in [23] by sharing some components among the sub-carry generators. Moreover, a variable latency speculative adder (VLSA) is then proposed with an error detection and recovery scheme [23]. VLSA achieves a speedup of 1.5× on average compared to CLA.

2.2 Segmented adders

2.2.1 The Equal Segmentation Adder (ESA)

A dynamic segmentation with error compensation (DSEC) is proposed in [19] to approximate an adder. This scheme divides an n-bit adder into a number of smaller sub-adders; these sub-adders operate in parallel with fixed carry inputs. In this paper, the error compensation technique is ignored because the focus is on the approximate design, so the equal segmentation adder (ESA) (Fig. 2) is considered as a simple structure of the DSEC adder. In Fig. 2, $\lceil \frac{n}{k} \rceil$ sub-adders are used, l is the size of the first sub-adder ($l \leq k$), and k is the size of the other sub-adders. Hence, the delay of ESA is $O(log(k))$ and the hardware overhead is significantly less than ACA.

2.2.2 The Error-Tolerant Adder Type II (ETAII)

Another segmentation based approximate adder (ETAII) is proposed in [27]. Different from ESA, ETAII consists of carry generators and sum generators, as shown in Fig. 3 (n is the adder size; k is the size of the carry and sum generators). The carry signal from the previous carry generator propagates to the next sum generator. Therefore, ETAII utilizes more information to predict the carry bit and thus, it is more accurate compared with ESA for the same k. Because the sub-adders in ESA produce both sum and carry, the circuit complexity of ETAII is similar to ESA, however its delay is larger ($O(log(2k))$). In addition to ETAII, several other error tolerant adders (ETAs) have been proposed by the same authors in [26,28,29].

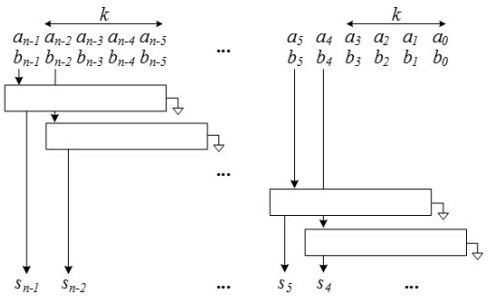


Figure 1: The almost correct adder (ACA). ▭ : the carry propagation path of the sum bit.

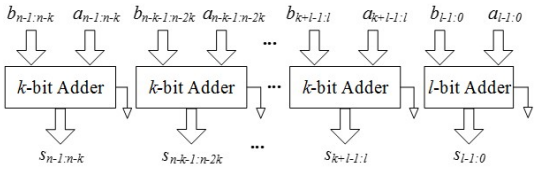


Figure 2: The equal segmentation adder (ESA). k: the maximum carry chain length; l: the size of the first sub-adder ($l \leq k$).

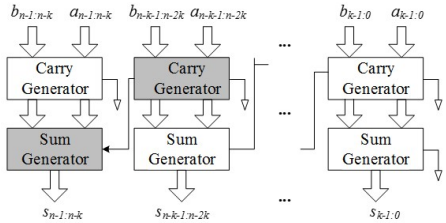


Figure 3: The error-tolerant adder type II (ETAII) [27]: the carry propagates through the two shaded blocks.

2.2.3 Accuracy-Configurable Approximate Adder

An accuracy-configurable approximate adder (ACAA) is proposed in [7]. As accuracy can be configured at runtime by changing the circuit structure, a tradeoff of accuracy versus performance and power can be achieved. In an n-bit adder, $\lceil \frac{n}{k} - 1 \rceil$ $2k$-bit sub-adders are required. Each sub-adder adds $2k$ consecutive bits with an overlap of k bits, and all $2k$-bit sub-adders operate in parallel to reduce the delay to $O(log(2k))$. In each sub-adder, the half most significant sum bits are selected as the partial sum. An error detection and correction (EDC) circuit is used to correct the errors generated by each sub-adder. The accuracy configuration is implemented by the approximate adder and its EDC with a pipelined architecture. For the same k, the carry propagation path is the same for each sum bit as in ACAA and ETAII; hence they have the same error characteristics.

2.2.4 The Dithering Adder

The dithering adder [18] starts by dividing a multiple-bit adder into two sub-adders. The higher sub-adder is an accurate adder and the lower sub-adder consists of a conditional upper bounding module and a conditional lower bounding module. An additional "Dither Control" signal is used to configure an upper or lower bound of the lower

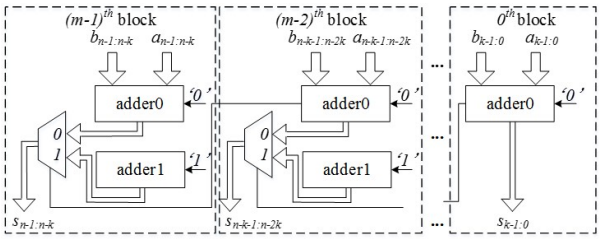


Figure 4: The speculative carry selection adder (SCSA).

sum and carry into the higher accurate sub-adder, resulting in a smaller overall error variance.

2.3 Carry Select Adders

In the carry select adders, several signals are commonly used: generate $g_j = a_j b_j$, propagate $p_j = a_j \oplus b_j$, and $P^i = \prod_{j=0}^{k-1} p_j^i$. $P^i = 1$ means that all k propagate signals in the i^{th} block are true.

2.3.1 The Speculative Carry Select Adder (SCSA)

The SCSA is proposed in [1]. An n-bit SCSA consists of $m = \lceil \frac{n}{k} \rceil$ sub-adders (window adders). Each sub-adder is made of two k-bit adders: adder0 and adder1, as shown in Fig. 4. Adder0 has carry-in "0" while the carry-in of adder1 is "1"; then the carry-out of adder0 is connected to a multiplexer to select the addition result as a part of the final result. Thus, the critical path delay of SCSA is $t_{adder} + t_{mux}$, where t_{adder} is the delay of the sub-adder ($O(log(k))$), and t_{mux} is the delay of the multiplexer. SCSA and ETAII achieve the same accuracy for the same parameter k, because the same function is used to predict the carry for every sum bit. Compared with ETAII, SCSA uses an additional adder and multiplexer in each block and thus, the circuit of SCSA is more complex than ETAII.

2.3.2 The Carry Skip Adder (CSA)

Similar to SCSA, an n-bit carry skip adder (CSA) [8] is divided into $\lceil \frac{n}{k} \rceil$ blocks, but each block consists of a sub-carry generator and a sub-adder. The carry-in of the $(i+1)^{th}$ sub-adder is determined by the propagate signals of the i^{th} block: the carry-in is the carry-out of the $(i-1)^{th}$ sub-carry generator when all the propagate signals are true ($P^i = 1$), otherwise it is the carry-out of the i^{th} sub-carry generator. Therefore, the critical path delay of CSA is $O(log(2k))$. This carry select scheme enhances the carry prediction accuracy.

2.3.3 The Gracefully-Degrading Accuracy-Configurable Adder (GDA)

An accuracy-configurable adder, referred to as the gracefully-degrading accuracy-configurable adder (GDA), is presented in [25]. Control signals are used to configure the accuracy of GDA by selecting the accurate or approximate carry-in using a multiplexer for each sub-adder. The delay of GDA is determined by the carry propagation and thus by the control signals to multiplexers.

2.3.4 The Carry Speculative Adder

Different from SCSA, the carry speculative adder (CSPA) in [12] contains one sum generator, two internal carry generators (one with carry-0 and one with carry-1) and one carry predictor in each block. The output of the i^{th} carry predictor is used to select carry signals for the $(i+1)^{th}$ sum generator. l input bits (rather than $k, l < k$) in a block are used in a carry predictor. Therefore, the hardware overhead is reduced compared to SCSA.

2.3.5 The Consistent Carry Approximate Adder

The consistent carry approximate adder (CCA) [10] is also based on SCSA. Likewise, each block of CCA comprises adder0 with carry-0 and adder1 with carry-1. The select signal of a multiplexer is determined by the propagate signals, i.e., $S_i = (P^i + P^{i-1})SC + \overline{(P^i + P^{i-1})}C_{out}^{i-1}$, where C_{out}^{i-1} is the carry out of the $(i-1)^{th}$ adder0, and SC is a global speculative carry (referred to as a consistent carry). In CCA, the carry prediction depends not only on its LSBs, but also on the higher bits. The critical path delay and area complexity of CCA are similar to SCSA.

2.3.6 The Generate Signals Exploited Carry Speculation Adder (GCSA)

In [5], the generate signals are used for carry speculation. GSCA has a similar structure as CSA. The only difference between them is the carry selection; the carry-in for the $(i+1)^{th}$ sub-adder is selected by its own propagate signals rather than its previous block. The carry-in is the most significant generate signal g_{k-1}^i of the i^{th} block if $P^i = 1$, or else it is the carry-out of the i^{th} sub-carry generator. The critical path delay of GCSA is $O(log(2k))$ due to the carry propagation. This carry selection scheme effectively controls the maximal relative error.

2.4 Approximate Full Adders

2.4.1 The Lower-Part-OR Adder (LOA)

LOA [17] divides an n-bit adder into an $(n-l)$-bit more significant sub-adder and an l-bit less significant sub-adder. For the less significant sub-adder, its inputs are simply processed by using OR gates (as a simple approximate full adder). The more significant $(n-l)$-bit sub-adder is an accurate adder. An extra AND gate is used to generate the carry-in signal for the more significant sub-adder by ANDing the most significant input bits of the less significant sub-adder. The critical path of LOA is from the AND gate to the most significant sum bit of the accurate adder, i.e., approximately $O(log(n-l))$. LOA has been utilized in a recently-proposed approximate floating-point adder [14].

2.4.2 Approximate Mirror Adders (AMAs)

In [2], five AMAs are proposed by reducing the number of transistors and the internal node capacitance of the mirror adder (MA). The AMA adder cells are then used in the LSBs of a multiple-bit adder. However, the critical paths of AMA1-4 are longer than LOA because the carry propagates through every bit. As for AMA5, the carry-out is one of the inputs; thus, no carry propagation exists in the LSBs of an approximate multiple-bit adder.

2.4.3 Approximate Full Adders using Pass Transistors

Three approximate adders (AXAs) based on XOR/XNOR gates and multiplexers (implemented by pass transistors) have been presented in [24]. Several approximate complementary pass transistor logic (CPL) adders have been pro-

posed by reducing the number of transistors in the accurate CPL adder [20]. Significant area and power savings have been obtained for both types of approximate designs.

3. COMPARATIVE EVALUATION

In the evaluation, SCSA is selected as a typical carry select adder and LOA is considered as a representative design using approximate full adders. All adders and sub-adders are implemented as CLA in this paper.

3.1 Error Characteristics

To evaluate the accuracy of approximate adders, analytical and simulation-based approaches have been proposed [6, 11, 13, 18, 22]. In this paper, Monte Carlo simulation is performed. The error rate (ER, the probability of producing an incorrect result), the normalized mean error distance (NMED, the normalization of mean error distance (MED) by the maximum output of the accurate adder) and the mean relative error distance (MRED, the average value of all possible relative error distances (REDs)) are used to assess the error characteristics of the approximate designs. Error distance (ED) and RED are calculated as: $ED = |M' - M|$ and $RED = \frac{ED}{M}$, where M' is the approximate result and M is the accurate result [11]. MED is the mean of all possible EDs.

The functions of the approximate adders ($16-$bit) are implemented in MATLAB and simulated with 10^8 random input combinations. The error measures in ER, NMED and MRED are obtained. Fig. 5 shows the simulation results where each adder's name is followed by the value of its parameter k. For ACA, ETAII and ESA, k is the size of the sub-adder, while k is the size of the less significant adder for LOA (as implemented by OR gates).

The NMED and MRED values of the approximate adders with data sorted by the MRED are shown in Fig. 5(a). The logarithms (base 10) of the NMED and MRED are plotted, and the vertical axis is labeled by negative numbers. In this figure, ETAII-k represents ETAII-k, SCSA-k and ACAA-k because they have the same carry propagation chain for each sum bit and hence, the same error characteristics (ER, NMED and MRED). The NMED and MRED show the same trend, so we only consider MRED in the comparison. Fig. 5(b) shows the comparison of ER and MRED of the approximate adders with data sorted by ER.

Among these approximate adders, ETAII-6 has the smallest MRED, while ESA-3 has the largest. LOA (shown in different patterns in Fig. 5(b)) has a structure different from the other approximate adders. Its higher part is totally accurate, while the approximate part is less significant. Therefore, the MRED of LOA is rather small but its ER is very large. The information used to predict each carry in ESA is rather limited, so its MRED and ER are the largest (excluding LOA). Specifically, a lower ER usually indicates a smaller MRED; a larger k normally means a lower ER and MRED for all the approximate adders (except for LOA). Compared with ETAII, SCSA and ACAA (represented by ETAII in Fig. 5), ACA gives slightly higher ER and MRED for the same k due to the shorter carry propagation chain (thus, less information is used for predicting the carry bits).

In summary, ETAII-6, SCSA-6 and ACAA-6 are the most accurate adders among the compared designs since they have the smallest ER and MRED. ESA-3 is the least accurate design in terms of ER and MRED.

Table 1: Circuit Characteristics Comparison of ETAII, SCSA and ACAA.

Adder Type	Delay (ps)	Area (um^2)	Power (uW)
ETAII-4	560	254	68.4
SCSA-4	370	394	117.6
ACAA-4	700	272	73.6

3.2 Circuit Characteristics

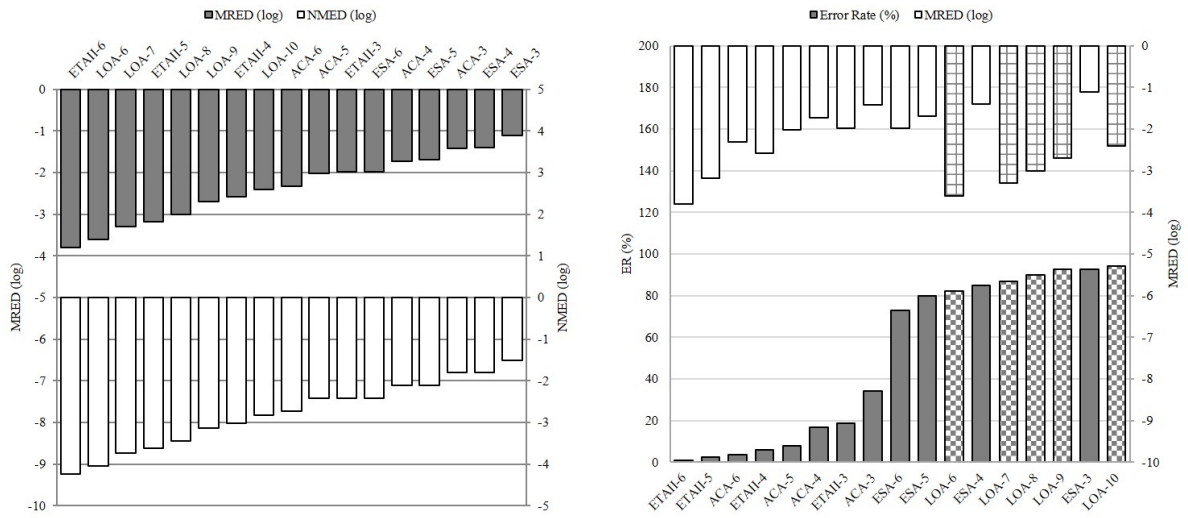
To assess circuit characteristics, the considered 16-bit approximate adders and the accurate CLA are implemented in VHDL and synthesized using the Synopsys Design Compiler based on an STM 65-nm process with a supply voltage of 1.35 V; delay, area and power are then obtained. Among ETAII, SCSA and ACAA (with the same error characteristics when the same parameter k is selected), SCSA incurs the largest area and power dissipation because two sub-adders and one multiplexer are utilized in each block, and ACAA is the slowest because of its long critical path ($2k$) (Table 1). The block of ETAII (a carry generator and a sum generator) is significantly simpler than those of SCSA and ACAA. Therefore, ETAII has a shorter delay than ACAA and consumes less power and area than SCSA, thus ETAII is selected for circuit comparisons.

A circuit with larger area is likely to consume more power, so only power and delay are considered in the comparison. Fig. 6 shows the delay and power of the approximate adders with ascending delay (Fig. 6(a)) and power (Fig. 6(b)) from left to right. Obviously, the accurate CLA has the longest delay among all adders, but not the highest power dissipation. LOA (shown in different patterns) is the slowest, but it is very power efficient compared with the other approximate adders. With the same k, ESA is the fastest (when k is 3 or 4) and most power efficient due to its simple segment structure. ETAII is the slowest excluding LOA, and ACA incurs the largest power consumption due to its complex speculation circuit. Among all the approximate adders shown in Fig. 6, ESA-3 is the fastest, while LOA-8 is the slowest. The delays of all ACAs are very close (less than 400 ps), and all the LOAs have a delay larger than 600 ps. For ESA, a smaller k leads to a smaller delay and power dissipation, a larger k also shows significantly larger values of these metrics.

Since a smaller delay does not always imply lower power dissipation, the power-delay-product (PDP) is used as a joint metric to evaluate the circuit characteristics of the approximate adders. Fig. 7 shows in ascending order the PDPs of the approximate adders from left to right. ESA-3 has the smallest PDP, while the accurate CLA has the largest value. LOA-10 and LOA-9 have moderate PDPs (due to large delay and low power dissipation). In terms of PDP, the approximate adders can be classified into three classes: ESA-3 and ESA-4 have the smallest PDP with less than 10 fJ, then ESA-5, LOA-10 and ACA-3 with around 20 fJ, and the PDPs of the other approximate adders are larger than 20 fJ and less than 45 fJ.

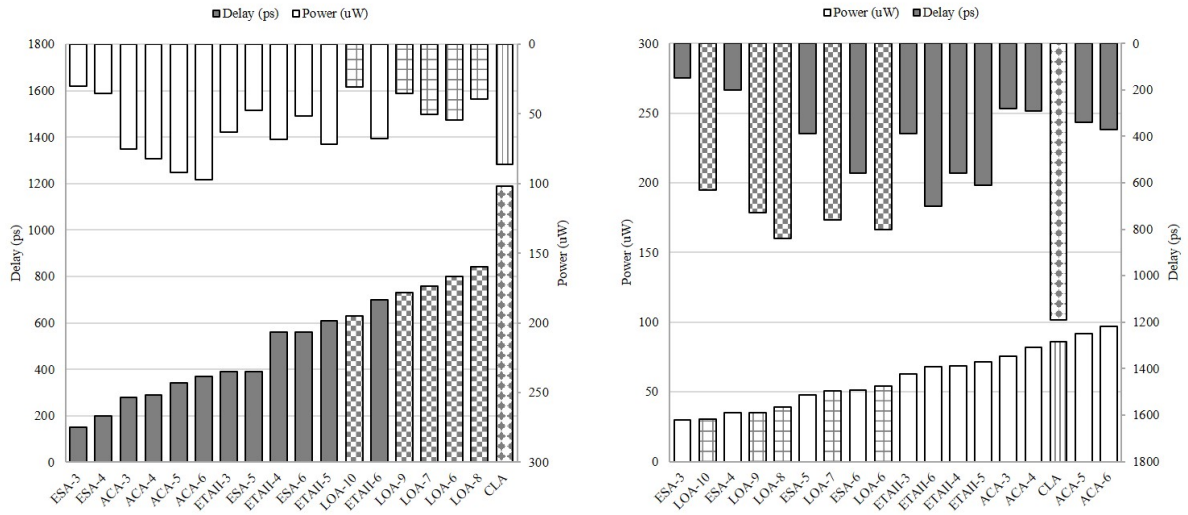
4. DISCUSSION AND CONCLUSION

In this paper, current approximate adders are reviewed; their error and circuit characteristics are evaluated. Fig. 8 shows the MRED and PDP of the approximate adders in



(a) (b)

Figure 5: A comparison of error characteristics of approximate adders with data sorted on (a) MRED and (b) ER.



(a) (b)

Figure 6: A comparison of delay and power of approximate adders with data sorted on (a) delay and (b) power.

a two-dimension (2-D) plot. ESA-3 and ESA-4 have rather small PDP but a considerably large MRED; ETAII-6, LOA-6 and ETAII-5 are the opposite, i.e., they have a small MRED but a large PDP. These approximate adders do not show the best tradeoff, but they can be used for special applications where either hardware efficiency or high accuracy is required. ESA-5, LOA-10, ETAII-3 and ACA-4 show moderate MRED and PDP.

Overall, ESA is the most hardware-efficient design but it is also the least accurate. ETAII, SCSA and ACAA have the same accuracy when their parameters are the same, whereas ETATII shows the smallest PDP among them. ACA is the most power consuming design with a moderate accuracy, while LOA is the slowest but it is highly power efficient among all approximate adders.

5. REFERENCES

[1] K. Du, P. Varman, and K. Mohanram. High performance reliable variable latency carry select addition. In *DATE*, pages 1257–1262, 2012.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. Low-power digital signal processing using approximate adders. *IEEE Trans. CAD*, 32(1):124–137, 2013.

[3] J. Han and M. Orshansky. Approximate Computing: An Emerging Paradigm For Energy-Efficient Design. In *ETS*, pages 1–6, Avignon, France, 2013.

[4] R. Hegde and N. Shanbhag. Soft digital signal processing. *IEEE Trans. VLSI Syst.*, 9(6):813–823, 2001.

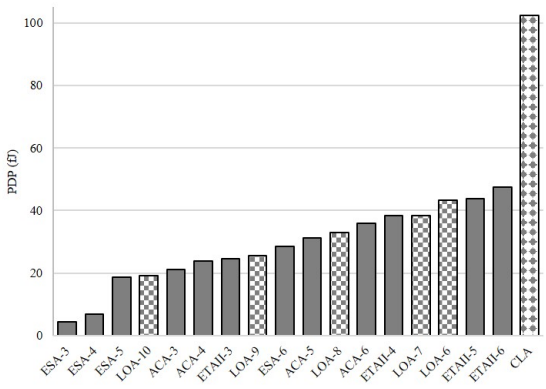


Figure 7: Sorted PDP of the approximate adders.

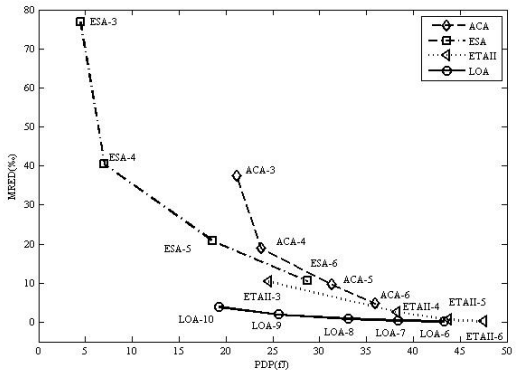


Figure 8: MRED and PDP of the approximate adders.

[5] J. Hu and W. Qian. A new approximate adder with low relative error and correct sign calculation. In *DATE*, in press, 2015.

[6] J. Huang, J. Lach, and G. Robins. A methodology for energy-quality tradeoff using imprecise hardware. In *DAC*, pages 504–509, 2012.

[7] A. B. Kahng and S. Kang. Accuracy-configurable adder for approximate arithmetic designs. In *DAC*, pages 820–825, 2012.

[8] Y. Kim, Y. Zhang, and P. Li. An energy efficient approximate adder with carry skip for error resilient neuromorphic vlsi systems. In *ICCAD*, pages 130–137, 2013.

[9] I. Koren. *Computer Arithmetic Algorithms, 2nd edition*. A K Peters/CRC Press, Natick, 2001.

[10] L. Li and H. Zhou. On error modeling and analysis of approximate adders. In *ICCAD*, pages 511–518, 2014.

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

[12] I. Lin, Y. Yang, and C. Lin. High-performance low-power carry speculative addition with varible latency. *IEEE Trans. VLSI Syst.*, in press, 2014.

[13] C. Liu, J. Han, and F. Lombardi. An analytical framework for evaluating the error characteristics of approximate adders. *IEEE Trans. Comput.*, 2014.

[14] W. Liu, L. Chen, C. Wang, M. ONeill, and F. Lombardi. Inexact floating-point adder for dynamic image processing. In *IEEE-NANO*, pages 239–243, 2014.

[15] Y. Liu, T. Zhang, and K. Parhi. Computation error analysis in digital signal processing systems with overscaled supply voltage. *IEEE Trans. VLSI Syst.*, 18(4):517–526, 2010.

[16] S.-L. Lu. Speeding up processing with approximation circuits. *Computer*, 37(3):67–73, 2004.

[17] 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 Trans. Circuits Syst.*, 57(4):850–862, 2010.

[18] J. Miao, K. He, A. Gerstlauer, and M. Orshansky. Modeling and synthesis of quality-energy optimal approximate adders. In *ICCAD*, pages 728–735, 2012.

[19] D. Mohapatra, V. Chippa, A. Raghunathan, and K. Roy. Design of voltage-scalable meta-functions for approximate computing. In *DATE*, pages 1–6, 2011.

[20] D. Nanu, R. P. K., D. Sowkarthiga, and K. S. A. Ameen. Approximate adder design using cpl logic for image compression. *International Journal of Innovative Research and Development*, 3(4):362–370, 2014.

[21] B. Parhami. *Computer Arithmetic: Algorithms and Hardware Designs, 2nd edition*. Oxford University Press, New York, 2010.

[22] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan. Macaco: Modeling and analysis of circuits for approximate computing. In *ICCAD*, pages 667–673, 2010.

[23] A. K. Verma, P. Brisk, and P. Ienne. Variable latency speculative addition: A new paradigm for arithmetic circuit design. In *DATE*, pages 1250–1255, 2008.

[24] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi. Approximate XOR/XNOR-based adders for inexact computing. In *IEEE-NANO*, pages 690–693, 2013.

[25] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu. On reconfiguration-oriented approximate adder design and its application. In *ICCAD*, pages 48–54, 2013.

[26] N. Zhu, W. L. Goh, G. Wang, and K. S. Yeo. Enhanced low-power high-speed adder for error-tolerant application. In *SOCC*, pages 323–327, 2010.

[27] N. Zhu, W. L. Goh, and K. S. Yeo. An enhanced low-power high-speed adder for error-tolerant application. In *ISIC 2009*, pages 69–72, 2009.

[28] N. Zhu, W. L. Goh, and K. S. Yeo. Ultra low-power high-speed flexible Probabilistic Adder for Error-Tolerant Applications. In *SOCC*, pages 393–396, 2011.

[29] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong. Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing. *IEEE Trans. VLSI Syst.*, 18(8):1225–1229, 2010.