

Review on High Performance Adder Design for Multimedia Applications

Dolly Kaushik¹ and Shweta Agrawal²

¹Research scholar, ²Assistant Professor,

Dept of Electronics and Comm., SRCEM Banmore, Morena, India

Abstract— Multimedia applications on the portable devices are raising exponentially. To achieve high performance, different algorithmic level efforts are done. Along with these, high performance adders are gaining more popularity due to usage in signal processing. Since the multimedia applications produce output used for the human consumption, these applications can accept small amount of error due to limited perception of human sense. Therefore, different approximate adders are developed in the literature. In this paper, an exhaustive literature review is done and then the performance of the existing adder designs is evaluated and compared. These existing designs are implemented and simulated with benchmark input to compute the efficacy of one over the other existing architectures. The designs are modelled on MATLAB and Tanner, simulated with benchmark inputs and then quality and design metrics are evaluated and compared.

Keywords— Digital Signal Processing (DSP), Approximate adders, Image Processing, Integrated Circuits, VLSI, Low Power Design.

I. Introduction

The modern portable devices are employing several multimedia applications [1]. These applications exhibit huge computations to achieve desired output. Therefore, significant research has been carried out to achieve high performance signal processing. Further, the growing number of functions on these gadgets demands VLSI architectures/design which can process signal very efficiently which ultimately increases the complexity of the designs [2]. The complexity of today's design is very high which result in high power and delay in the present devices and it is growing with increasing functionality on the same device. The conventional approach to improve the performance of this design is the device scaling. The scaling approach has reached to its level and the devices cannot be scaled further due to increased effect of process and other variations [3].

In the sub-nanometer designs, the process variation has become so severe that designs without considering it will fail to provide desired output. Further, addition circuit to mitigate the effect of process variation is very costly in terms of power, area and delay such that gain due scaling are less than overhead. Therefore, other design methodology is required to develop designs for the modern gadgets. There are several applications where the approximate results are acceptable such as image/video processing. The relaxation on the accuracy can be exploited to reduce the complexity of the designs [4].

The adder is most basic operation along with multiplier used to performance different signal processing. Even the multiplier also contains adder in it. Therefore, design of

high performance adder may significantly improve the performance of these applications. Along with conventional Ripple carry adder (RCA) [5], several high performance adders such as Carry look-ahead (CLA) [6], Carry select adder (CSL) [7], and Carry Skip adder (CSK) [8] have been developed. This adder provides improved performance at the cost of increased area/power overhead. In order to improve all the three metrics simultaneously, approximate adders are developed for the error tolerant applications.

The several approximate adder architectures include error tolerant adder (ETA-I), ETA-II, ETA-IIM etc. are proposed by Zhu et al. [9]. In ETA-I operands are divided into upper and lower parts where upper part is computed accurately while the lower part approximately. In addition to the approximate adders, accuracy configurable adder is also presented by Khang et al. [10]. This adder provides variable accuracy at the cost of reduced performance. Therefore, accuracy configurable design can be used in wide applications due to exhibiting variable accuracy. The existing adder designs are not efficient and therefore demand more efficient adder architectures.

The rest of the paper first discusses different accurate adders followed by approximate adders and finally compares them by implementing and computing design and quality metrics.

II. Review on accurate adder architectures

This section details different accurate adder architecture such as ripple carry adder, carry select adder, carry skip adder and carry look ahead adder architectures.

2.1 Ripple Carry Adder (RCA)

The RCA [5] is the simplest and accurate adder architecture as shown Fig. 1 where the adder consists of some full adders where each of full adder exist three inputs and two outputs. RCA is very simple in structure wise and area efficient as well. It is constructed by taking number of full adders cascading in series. However, for the large bit numbers it is not very efficient.

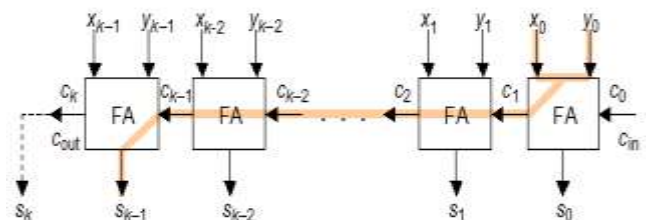


Fig. 1: Architecture of RCA

The major limitation of the RCA is its delay large propagation delay. The worst case propagation delay occurs when carry moves from LSB to MSB and makes RCA as slowest among the existing adders. Consequentially we can say that it is simple and energy efficient as well but slow in speed. Mostly, conventional circuits designed with ripple carry adder where performance is not the prime issue and area is the major factor.

2.2 Carry Select Adder (CSL)

In order increase the speed of slow RCA, commonly used adder is CSL [7]. This type of conventional adder provides efficient speed to the circuit. The architectural diagram of CSL adder is shown in Fig. 2. By the use of this technique say carry select adder, divides large RCA bits into small RCA blocks. This small RCA blocks have two carry paths: one for logical zero and one for logical one. Thereafter carry propagation at high RCA become selection, consequently reduces significantly delay at the cost of increased hardware over RCA.

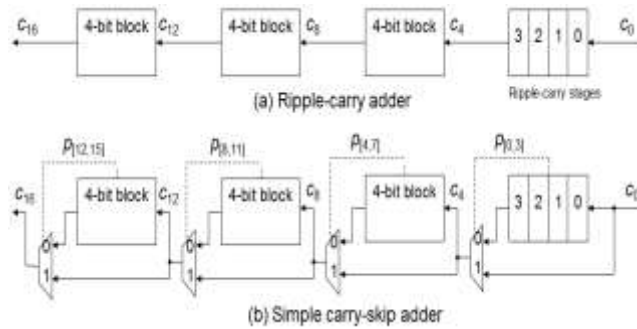


Fig. 3: Architecture of CSK

2.3 Carry Look-ahead Adder

The concept of carry-look-ahead adder (CLA) [6] is to determine the carry required in the MSB position in advance to reduce the carry propagation delay. Since this adder reduces the propagation time, it significantly improves the performance of the adder with litter area and power overhead of the adder. The CLA computes the carry-in for the different group of sub-adders and make used of this carry-in signal to reduce the dependency thus reduces the delay of the addition. They generate and propagate signal for the given binary signal is given by the equations 3.1 and 3.2. which can be easily implemented by the simple logic gates.

$$g_i = x_i \cdot y_i \tag{3.1}$$

$$p_i = x_i \oplus y_i \tag{3.2}$$

$$a_i = \text{not}(x_i + y_i) \tag{3.3}$$

$$t_i = (x_i + y_i) \tag{3.4}$$

The logical block diagram of the CLA is shown in the Figure 3.4. It allows independent carry generation for each bits and consists of the simple AND and OR logic.

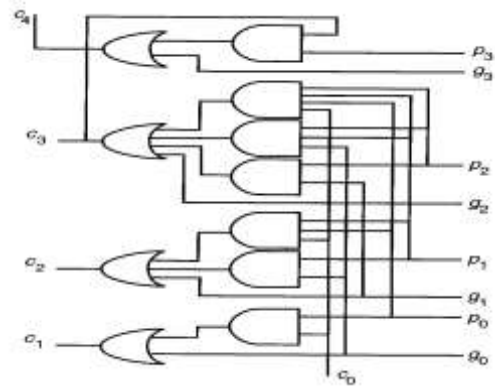


Fig. 3: Architecture of CLA.

III. Approximate adder architectures

This section provides different approximate adders energy efficient kernels, given that can be used to efficiently compute the smoothed pixel.

3.1 Scope of Approximate Adders

The existing simple RCA cannot be used due to its poor performance as it has very large carry propagation while other fast adders such as CLA, CSL and CSK are area and power inefficient. Therefore, there is requirement of the adder which provides all the three parameters to be efficient. Moreover, there are applications where, 100% accuracy is not required i.e. these applications can accept output with small error. For these applications approximate adder can be designed such that there is significant improvement in design metrics with small introduced error. To compute the error produced by the approximate adder, different error metrics are developed and are detailed in the next subsection.

3.2 Error Metric for approximate adders [9]

There are number of error metrics developed to compute the error/accuracy of the designs. The approximate adder uses some terminologies of error metric which are as given as follows:

- **Overall error:** The overall error (OE) can be expressed by equation given below
 $OE = [R_C - R_a]$
 where R_a is approximate and R_C represent correct result.
- **Accuracy (ACC):** The accuracy of the given results quantifies the amount of correctness over the desired value. It is expressed as:
 $ACC = (1 - (OE/R_C)) \times 100\%$
 The accuracy can have value from 0% to 100%.
- **Minimum acceptable accuracy (MAA):** In error tolerant application some error is allowed but the designs should produce error below an acceptable level. The MAA represent the maximum error an application can tolerate.
- **Acceptance probability:** Acceptance probability (AP) represents the probability of a design that provides results of accuracy higher than MAA and can be expressed as
 $AP = P(ACC > MAA)$
 Where the value of AP ranges from 0 to 1

3.3 Error Tolerant Adder (ETA-I)

The architecture diagram of the ETA [9] is shown in the Fig. 4, which shows that ETA-I is divided in two parts: accurate and inaccurate parts. The accurate part contains some most significant bits (MSBs) whereas the inaccurate part contains few least significant bits (LSBs). Since the higher order bits play more important role than the lower order bits' normal addition method is applied for accurate part to preserve its correctness and special strategy is adopted for the inaccurate part. Further its accurate part is implemented with any of accurate adders such as RCA, CSK, CSL, or CLA. Its carry is grounded and the accuracy of accurate part is too strong.

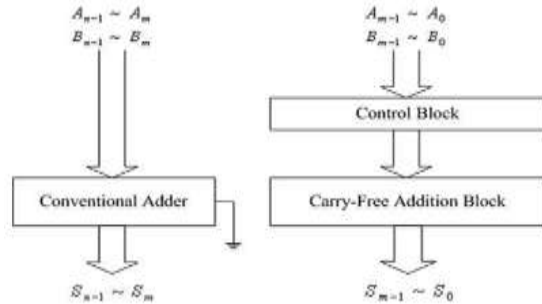


Fig. 4: Architecture diagram of ETA-I.

Another side the inaccurate part is implemented with the combination of two blocks; a control block and a carry-free addition block (CFAB). The control block is made up by and-or logic to generate the control signals, for the working mode of the carry free addition block. And the CFAB is made up by modified XOR gate where three additional transistors are added to original XOR circuit and with additional control signal. The working principle of the ETA-I can be better understood via an example as shown in Fig. 5. In this, we take two 16-bit input data as, X= "1011001110011010" (45978) and Y= "0110100100010011" (26899).

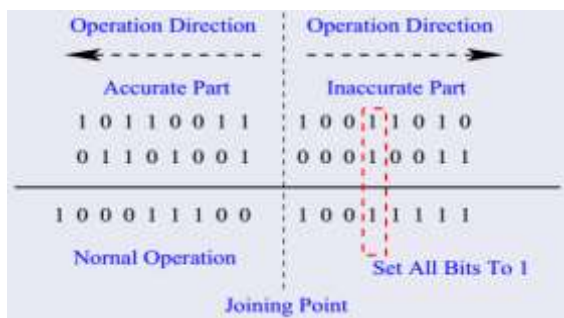


Fig. 5: Working of ETA-I.

The input bits are divided into two parts and the addition starts from the segmentation point in the two opposite direction in parallel. For the MSBs accurate addition is done from right to left to preserve its correctness and for inaccurate part no carry signal will be generated. To reduce error in overall addition a method is utilized in the approximate part in which carry will not be generated and forward. It checks both bits of the operand from right to left direction and if both bits are not logic '1', normal addition

is done without carry which can be achieved by XOR operation.

On the other hand, if both bits are '1', a control logic is generated and from this bit position onward, all the remaining lower sum bits are kept to logic '1' and the search operation is terminated. An example showing addition process using ETA-1 as shown in Fig. 5 shows that accurate adder produces output of 72877 while the ETA-1 produces 72863. The error introduced is less than 1%. Thus, the ETA reduces the delay by nearly half. The major limitation of the ETA is its poor accuracy for the small input i.e. it exhibits large error when small range input is applied. Therefore, ETA-1 is not suitable for the application which can have input data of any value as the ETA-1 adder is input data dependent design.

3.4 Sloppy Adder

Albicocco et al. [11] presented a sloppy adder that computes approximate sum with reduced complexity. In the proposed sloppy adder, least significant bits are computed with approximate logic due to their small contribution in the overall sum while the MSBs are evaluated in accurate manner to maintain the quality. In the approximation logic, author used OR logic to compute least significant some bits. The resulting architecture is shown in Fig. 6.

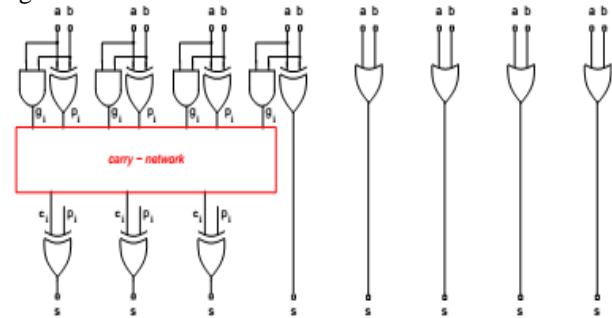


Fig. 6: 8-bit sloppy k = 4 adders

3.4 Accuracy Configurable Adder (ACA)

In order to improve the wide applicability of the approximate designs, an accuracy configurable adder [11] is presented which can configure the accuracy at run-time. This adder provides approximate sum during normal operation but can provide accurate results at the cost of performance power overhead. The approximate part of the ACA utilizes sub-adders to compute the partial bits of sum. The overall approximate sum is extracted in the following manner:

1. All bits of least significant sub-adders are considered.
2. Upper half of each sub-adder except least significant one bits are considered.
3. All bits are concatenated to achieve the overall sum.

In this adder error occurs only when there is carry transmission from one partial adder to another. The addition of error detection and correction to this approximate adder will make it as accuracy configurable adder. The EDC logic for this proposed adder is very simple and can be implemented by few AND gates. The

architecture of the accuracy configurable adder is shown in Fig. 7.

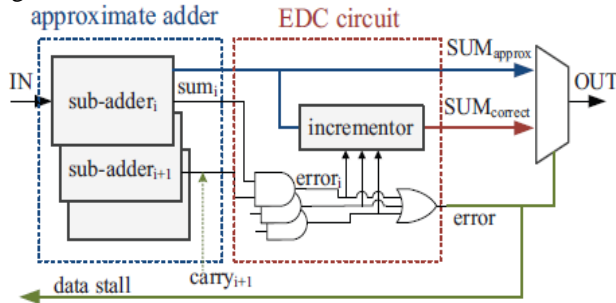


Fig. 7: ACA adder architecture

The adder when will produce incorrect result; it will generate a carry flag. In order to compute the accurate result, the error detected by the EDC will be added to the approximate part to compute the accurate sum. The approximate or accurate sum will be selected with the help of multiplexor.

The prime advantage of the ACA is to provide variable accuracy output while the major drawback is its large area overhead. Therefore, if the area of the ACA if can be reduce, it will provide an adder which is good in all respects.

IV. EXPERIMENTAL RESULT & ANALYSIS

In order to evaluate the quality metrics [9], [10], MATLAB tool is used to model the different accurate and approximate adder architectures and simulated. On the other hand, to evaluate the design metrics designs are implemented on Tanner schematic editor. Finally, the design metrics such as area, power and delay are extracted for the proposed and existing designs and compared.

4.1 Simulation results on MATLAB

The error metrics is shown in Table 1, which describes that 8-bit proposed adder exhibits nearly same characteristics as that of ETA-IIM and ACA under mode=0. These error metric reflect that the proposed adder can be effectively employed in the applications where ACA and the ETA-IIM can be used. Thus, proposed adder is suitable for different image and video processing applications. Similarly, the error metrics for higher bit-width adder 16-bit and 32-bit adder are extracted and compared.

Table 1: Comparison of 8-bit adder error metrics

Error Metrics	ACA			ETA2M	ETA1
	m=0	m=1	m=2		
Mean (μ)	7.6	7.56	0	7.6	2.98
MSE	357.2	426.9	0	357.2	14.13
Std (σ)	18.89	20.66	0	18.89	3.76
poe	0.19	0.11	0	0.19	0
cofact	0.4	0.366	0	0.4	0.79

The error metrics of the existing adder are tabulated in Table 2. The simulation results show that ETA1 exhibits

good error metrics but provides very poor design metrics. Whereas the error metrics of ETA2M is the very poor over the all existing designs.

Table 2: Error metrics of 16-bit adder.

Error Metrics	ACA			ETA-IIM	ETA-1
	m=0	m=1	m=2		
Mean	127.88	126.15	0	20448	61233
MSE	4.79×10^5	5.0×10^5	0	2.29×10^7	6.32×10^8
Std (σ)	692.6	707.6	0	4790	2.5×10^4
poe	0.057	0.03	0	0.4818	1
cofact	0.184	0.178	0	0.426	2.43

Thus, it observed from the simulation results that ETA-I provides good design metrics over all the existing adder architectures whereas the quality metrics are comparable to the accuracy configurable adder architecture.

4.2 Design metrics on Tanner

For estimating the design metrics of proposed and all the existing adders they are implemented on the Tanner v14.1 and simulated with 45nm technology file. Table 3 shows the design metric parameters for all the 8-bit conventional and approximate adders.

Table 3: Design metrics of various 8-bit adder.

Adder Arch.	#Tran	Power (mw)	Delay (ns)	PDP (nJ)
ETA-I	212	1.68	0.178	299.04
ETA-IIM	224	0.0076	0.163	1.239
ACA	336	0.0099	0.163	1.614

On comparing all the design metric of different adders through Table 3 we conclude that ETA-IIM requires minimum energy over ACA. Thus, we can say that ETA-IIM and ACA exhibits nearly same delay whereas ETA-I requires minimum area over the existing adder architectures. Fig. 8 compares the area of different adder architecture where the ETA-I shows small area over the all existing approximate adder architectures.

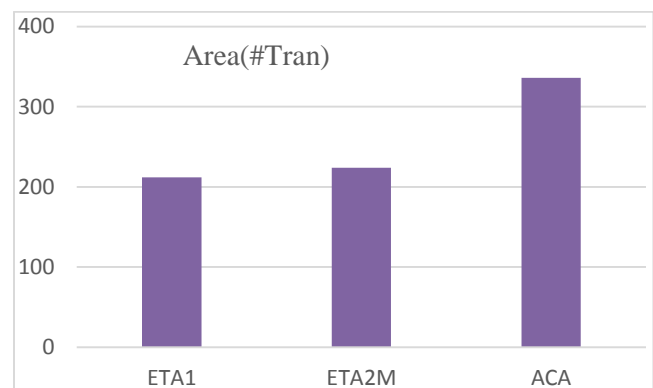


Fig. 8: Area of various 8-bit adder architectures.

Similarly, Fig. 9 compares the delay of different adder architecture where ETA-IIM and ACA exhibits same delay which is smaller over ETA-I.

Similarly, the design metrics for the 16-bit and 32-bit adders are computed as shown in Table 4. The design metrics show that ACA adder architecture requires very small area, power and delay over the existing adder architectures. The energy consumption (power delay product) is least for the ACA adder over all the existing adders.

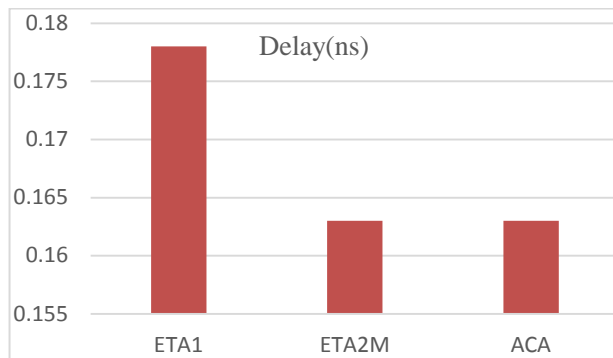


Fig. 9: Delay of 8-bit adder architectures.

Table 4: Design metrics of 16 and 32 bit adders

Adder Architectures		# Tran	Power (uw)	Delay (ns)	PDP (fJ)
ETA-I	16-bit adder	426	3.3	0.344	1.1352
ETA-IIM		448	0.143	0.43	0.06149
ACA		672	0.197	0.331	0.065207
ETA-I	32-bit adder	854	6.31	0.678	4.27818
ETA-IIM		896	0.244	0.666	0.146156
ACA		1344	1.14	0.66	0.75924

It can be observed from Table 4 that ACA adder requires least energy consumption for both 16-bit and 32-bit adder over the existing accurate and approximate adder architectures. Thus we can say ACA adder provides significant improvement in area, power, delay and PDP over all existing adders.

V. CONCLUSION

The high performance requirement by the different portable devices exhibiting multimedia applications can be achieved by designing approximate adder. This paper presents an exhaustive literature review on different kinds of approximate adder and accessed their performance in terms of area power and delay. These adders exhibit different tradeoff between these metrics.

REFERENCES

- [1]. Melvin A. Breuer and Haiyang Zhu, "Error-tolerance and multi-media," in Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2006.
- [2]. Neil. H. E. Weste, "Principle of CMOS VLSI Design," Addison-Wesley 1998.
- [3]. International Technology Roadmap for Semiconductors [Online]. Available: <http://public.itrs.net/>

- [4]. T. Y. Hsieh, K. J. Lee and M. A. Breuer, "Reduction of detected acceptable faults for yield improvement via error-tolerance," in Proceedings Design, Automation and Test European Conference Exhibition, pp. 1-6, 2007.
- [5]. M. Fawaz, N. Kobrosli, J. Rizakallah, M. Mansour, Ali Chehab, A. Kayssi and H. Hajj, "Energy minimization feedback loop for ripple carry adders," International conference on digital object identifier, pp. 1-2, 2010.
- [6]. M. Morrison and R. Meana, "Design of a novel reversible ALU using an enhanced logic carry look ahead adder," 11th IEEE conference on digital object identifier, pp. 1436-14406, 2011.
- [7]. B. Ram kumar and Harish M. Kittur., "Low power and area efficient carry select adder," IEEE Transaction on VLSI system, Vol. 20, No. 2, 2012.
- [8]. Michael J. Schulte, Kai Chirca, John Glossner, Haoran Wang, Suman Mamidi, Pablo Balzola and Stamatis Vassiliadis, "A low power carry skip adder with fast saturation," 15th IEEE International Conference on Application-Specific Systems, Architectures and Processors 2004.
- [9]. Ning Zhu, Wang Ling Goh and Kiat Seng Yeo, "An enhanced low-power high-speed Adder for Error-Tolerant application," in Integrated Circuits, ISIC '09, Proceedings of the 2009 12th International Symposium on, pp. 69-72, 2009.
- [10]. A. Kahng and S. Kang, Accuracy-con_gurable adder for approximate arithmetic designs," in Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE, june 2012, pp. 820{825.
- [11]. P. Albicocco, G. Cardarilli, A. Nannarelli, M. Petricca, and M. Re, Imprecise arithmetic for low power image processing," in Signals, Systems and Computers (ASILOMAR), 2012, pp. 983{987.