International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 05, May 2016
www.ijates.com
ijates
ISSN 2348 - 7550

PERFORMANCE OF DADDA MULTIPLIER USING CARRY SELECT ADDER

S.Shenbagam[1] , V.Vandhana[2]

[1]PG scholar, [2]Assistant Professor, Department of ECE,
Sasurie Academy of Engineering,Coimbatore, (India)

ABSTRACT

Approximate adders have been considered as a potential alternative for error tolerant applications to some accuracy for gains in other circuit based metrics such as power, area and delay. Existing approximate adder designs have shown advantages in improving many of these operational features. Now a days, low power applications play an important task in designing of VLSI based digital circuits. They demand to investigate different techniques to reduce power consumption in digital circuits while maintains computational throughput. Many methods are implemented to reduce the power dissipation. In a multiplier, most contribution of power consumption is due to generation and reduction of partial products. Among multipliers, Dadda multiplier shows enhanced performance in terms of power, area and delay than other multipliers. Dadda implementation can minimizes the number of adder stages required to perform the summation of partial products. In this paper, a new design technique for row compression of dadda multipliers is presented. To estimate the PSNR value of any image, matrix multiplication is needed which in turn is produced from dada multiplier with carry select adder.

Keywords: Approximate adders, Carry Select Adder, Dadda Multiplier , ETA11, Matrix Multiplication.

I. INTRODUCTION

1.1 Approximate Adders

As an important arithmetic module, the adder plays a key role in determining the speed and power consumption of a digital signal processing 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. As the physical dimensions of CMOS scale down to a few tens of nanometres, it has been increasingly difficult to improve circuit performance and 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. Approximate computing is motivated by the large and growing class of applications that demonstrate inherent error resilience, such as DSP, multimedia (images/audio/video),graphics, wireless communications, and emerging workloads such as recognition, mining, and synthesis. 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 nature of these applications, small errors in computation would not impose noticeable degradation in performance .

In a group of techniques that could be collectively classified as approximate computing, wherein the requirement of exact numerical or Boolean equivalence between the specification and implementation of a computing platform is relaxed in order to achieve improvements in performance or energy efficiency. These applications usually process large, redundant data sets that contain significant noise, by utilizing statistical or probabilistic computations. The requirement of numerical exactness on their outputs is relaxed due to several factors: (i) the limited perceptual capability of humans (e.g., audio, video, graphics), (ii) a golden result is difficult to defined or does not exist or (iii) users are willing to accept less-than-perfect results. Approximate computing in hardware is based on designs of hard- ware building blocks whose implementation does not exactly match the specification.

Approximate adders have been proposed by using a reduced number of transistors and by truncating the carry propagation chain for a speculation-based operation An approximate speculative designs achieve a better performance in terms of area, power and delay compared to conventional (exact) adders. New metrics and simulation- based approaches have been proposed to model and evaluate approximate adders according to specific computational features. Monte Carlo or exhaustive simulation approaches have been employed to acquire data for analysis. This class of approaches are however time-consuming and require building functional models of the approximate designs. To improve efficiency, a mathematical characterization of the arithmetic accuracy of approximate adders is then required for a better understanding of the design prior to a simulation based evaluation.

In addition to generic metrics (such as the error rate (ER)), application specific measures (ASMs) such as the peak signal-to-noise ratio (PSNR) for image processing are well suited in practice. Without an approach to modeling the relationship between the generic metrics and the ASMs, extensive programming and simulation efforts are required to obtain the ASMs for assessing the impact and the potential of approximate computing in different applications. Therefore, an effective approach to obtain or estimate the ASMs from generic error metrics is needed however, there no formal methodologies or analytical approaches for these purposes.

1.2 Multipliers

Multipliers are often found in the critical path of signal processors. The multiplication is a slow operation and the improvement in the multiplier performance usually leads to higher operating speed. Historically, technology scaling has been used to improve the performance at a reduced energy budget. Unfortunately, technology scaling cannot sustain the constant power density because of the threshold voltage limitation. Therefore, the circuit implementation techniques and energy-delay tradeoff become critical.

Energy efficient parallel multiplier design requires the exploration of multiplication algorithms, technology constraints and circuit implementation techniques. The parallel multipliers consist of three main computational blocks namely partial product generation, partial product reduction and final addition. The simplest way of partial product generation is AND operation . High speed multiplication is performed using Column Compression Multipliers such as Wallace and Dadda Multipliers.

II. EXISTING METHOD

2.1 Error Tolerant Adder Type 11 (Eta11)

The ETAII is also based on the truncation of the carry propagation chain and the segmentation of a full-sized adder .Compared to the ESA, the predicted carry input for each segmented k-bit sub-adder is generated by k LSBs. The ETAII has an improved accuracy compared to the ESA, because it uses more information to predict the carry when the same k is used.

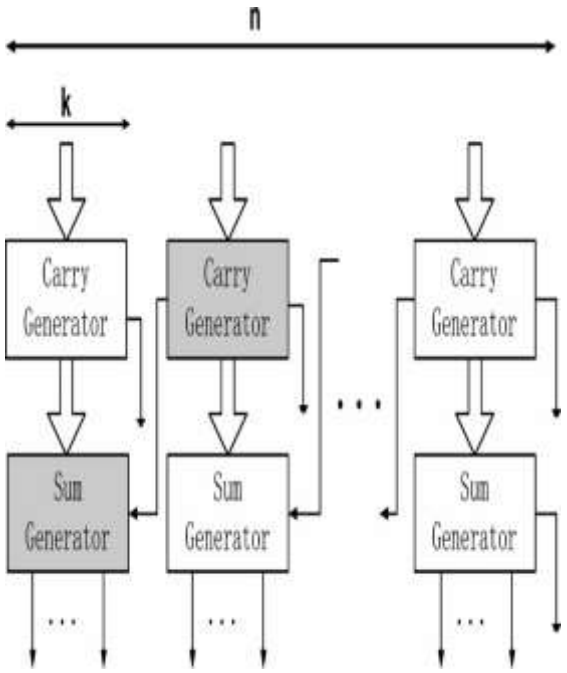

Fig 1 Error Tolerant Adder (Eta11)

2.2 Carry Select Adder

In the so-called carry select addition (CSA) an n-bit adder is first divided into sub-adders (also known as "window adders"); each sub-adder consists of two k-bit adders: adder0 and adder1. The only difference between the two k-bit adders is the carry input; the carry of adder0 is "0" while it is "1" for adder1. The output of the ith sub-adder is selected from adder0 and adder1 based on the carry out signal generated by the $(i-1)$th sub-adder. The carry out of each sub-adder is generated based on the k-bit in the sub-adder rather than all previous bits. Therefore, the carry selection process is still approximate and faster than a traditional carry selection scheme. Even though the CSA and the ETAII have different circuit implementations, they share a similar functionality if their sub-adders have the same length. The CSA and the ETAII generate the same carry signal for each sub-adder (or the Sum Generator in the ETAII) even though by different circuit implementations.

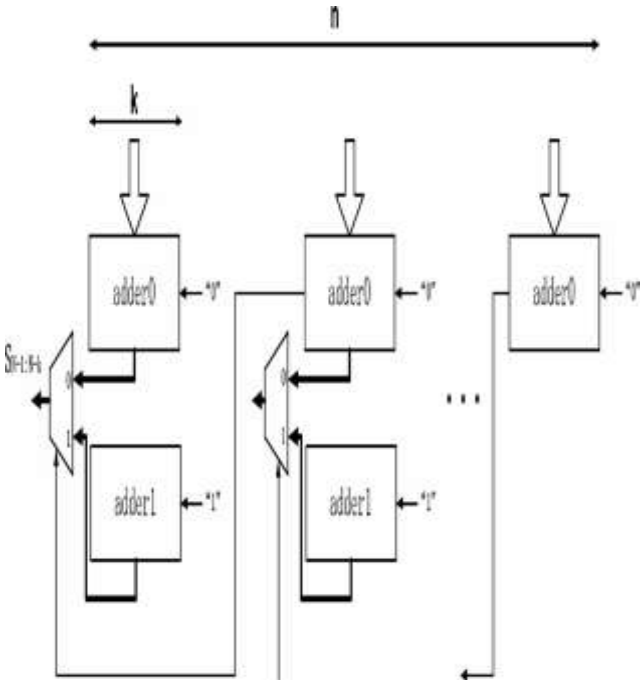

Fig 2 Carry Select Adder

TABLE 1 Comparision Of Adders

	ACA	ESA	ETAII	SCSA
K	10	10	6	10
DELAY ps	580	490	460	370
AREA ucm^2	398	394	254	234
POWER µW	117.6	88.9	68.4	45.5
ER%	4.74	2.7	1.8	0.05

III. PROPOSED METHOD

3.1 Dadda Multiplier

In a popular multiplication scheme the array, the summation proceeds in a more regular, but slower manner, to obtaining the summation of the partial products .Using this scheme only one row of bits in the matrix is eliminated at each stage of the summation.

In a parallel multiplier the partial products are generated by using array of AND gates. The main problem is the summation of the partial products, and it is the time taken to perform this summation which determines the maximum speed at which a multiplier may operate. The Dadda scheme essentially minimizes the number of adder stages required to perform the summation of partial products. This is achieved by using full and half adders to reduce the number of rows in the matrix number of bits at each summation stage.

Dadda multipliers are a refinement of the parallel multipliers presented by Wallace. Dadda multiplier consists of three stages. The partial product matrix is formed in the first stage by N2 AND stages. In the second stage, the partial product matrix is reduced to a height of two. Dadda replaced Wallace Pseudo adders with parallel (n, m) counters. A Parallel (n, m) counter is a circuit which has n inputs and produce m outputs which provide a binary count of the ONEs present at the inputs. A full adder is an implementation of a (3, 2) counter which takes 3 inputs and produces 2 outputs. Similarly a half adder is an implementation of a (2, 2) counter which takes 2 inputs and produces 2 outputs.

In Dadda multipliers that reduce the number of rows as much as possible on each layer, Dadda multipliers do as few reductions as possible. Because of this, Dadda multipliers have less expensive reduction phase, but the numbers may be a few bits longer, thus requiring slightly bigger adders. In a parallel multiplier, the terms yi ^ (xn-1 - . . . x0) are known as the partial products and are generated using an array of AND gates. For a parallel multiplier, the shifting term 2i is inherent in the wiring and does not require any explicit hardware. Thus the main problem is the summation of the partial products, and it is the time taken to perform this summation which determines the maximum speed at which a multiplier may operate.

Consider the process of multiplication of two binary numbers, each composed of n bit, as been based on obtaining the sum of v summands. These summands are obtained, in the simplest schemes, by shifting left the multiplicand by 1, 2, 3,….(n-1) places, and multiplying it by the corresponding bits of the multiplier. In this situation v = n. Now the number of summands can be made less than n by using some multiples of the multiplicand, on the basis of two or more multiplier digits.

Hence a proposed architecture can be developed by L Dadda, which works on the principle of reducing the number of summands. This architecture is based on the use of logical blocks called it as parallel (n, m) counters, these are combinational networks with m outputs and n(≤ 2m) inputs. The m outputs, considered as a binary number, codify the number of « ones» present at the inputs.

3.2 Algorithm

1.Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding N2 results.

2.Reduce the number of partial products to two layers of full and half adders. For this, Dadda reduction scheme uses the following algorithm.

a)Let d1 = 2 and dj+1 = [3.dj / 2], where dj is the matrix height for the j-th stage from the end. Find the largest j such that at least one column of the matrix has more than dj bits.

b)Employ (3, 2) and (2, 2) counters to obtain a reduced matrix with no more than dj elements in any column.

c)Until a matrix with only two rows is generated. Let j = j-1 and repeat step b

3.Group the wires in two numbers, and add them with a conventional adder.

Dadda generalized and extended Wallace's results by noting that a full adder can be thought of as a circuit, which counts the number of ones in the input and outputs that number in 2-bit binary form. Using such a counter, Dadda postulated that, at each stage, only minimum amount of reduction should be done in order to reduce the partial product matrix by a factor of 1.5. Dadda's method requires the same number of levels as that of Wallace method.

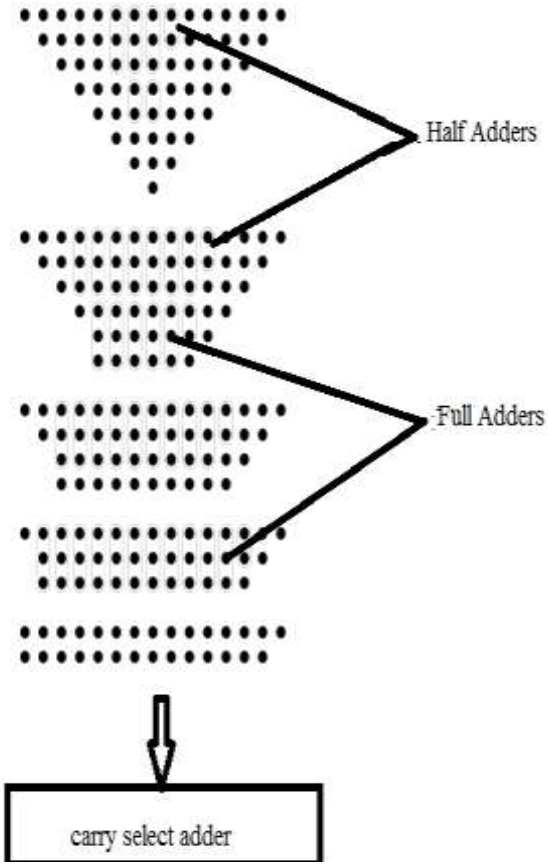

Fig 3 8×8 Dadda Multiplier

3.3 Matrix Multiplication

For approximate matrix multiplication suggests itself : pick a random subset of s columns of A to form an m × s matrix S; form an s × p matrix R out of the corresponding columns of B. Then, intuitively, it follows from that the product SR is an estimator (entry by entry) of the product AB; the variance remains to be worked out. Our contribution to the above and is two-fold. First, instead of picking columns uniformly at random, we pick them according to some "more in teresting" probability distribution. In general, we pick a column with probability proportional to its length squared, which is a measure of the amount of "information" the column contains. Second, before including a column in the sample, we scale it in order to compensate for the columns that are not picked.

With these two improvements, interesting bounds can be proven for the error of the approximations. This approach for approximating matrix multiplication has obvious advantages. It is conceptually simple, it can be easily implemented and it can be generalized to approximate the product of more than 2 matrices. Also, since the "heart" of the algorithm involves matrix multiplication of smaller matrices, it can use any algorithms that exist in the literature for performing the desired matrix multiplication.

3.4 PSNR

This section presents an image processing application using the 8x8 bit multipliers. An image where G is a matrix given by:

$$G = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}.$$

sharpening algorithm is considered and functionally implemented The processed image quality is measured by the peak signal noise ratio (PSNR); the PSNR quantifies the maximum possible power of signal and the power of an image with loss of accuracy following an additional process, such as compression and/or approximate computation. The PSNR is usually used to measure the quality of a reconstructive process involving information loss and is defined by the mean square error (MSE).

IV. SIMULATION RESULT

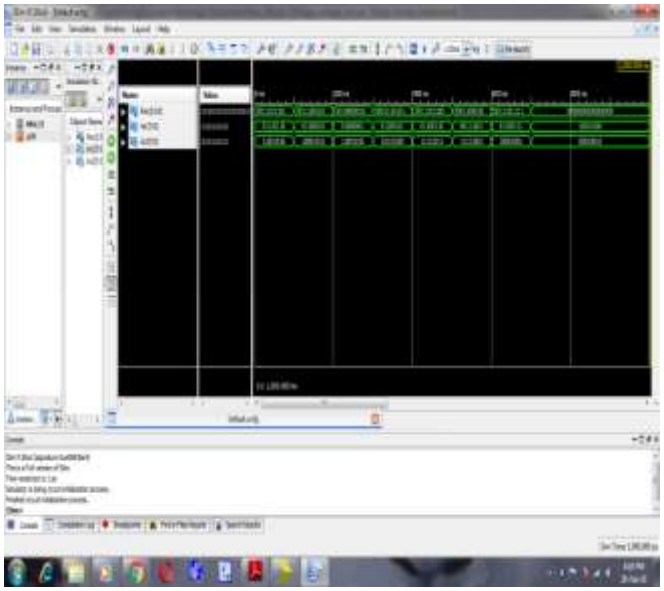

Fig 4 Multiplication Wave of Dadda Multiplier

International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 05, May 2016
www.ijates.com
ijates
ISSN 2348 - 7550

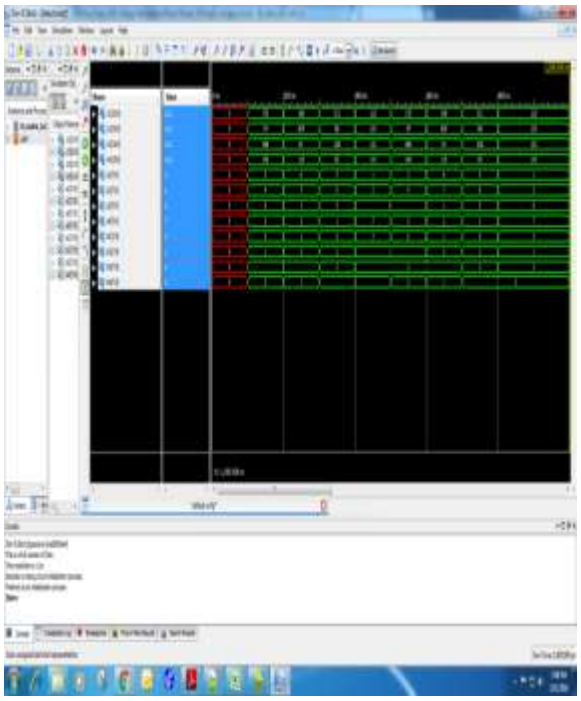

Fig 5 Matrix Multiplication

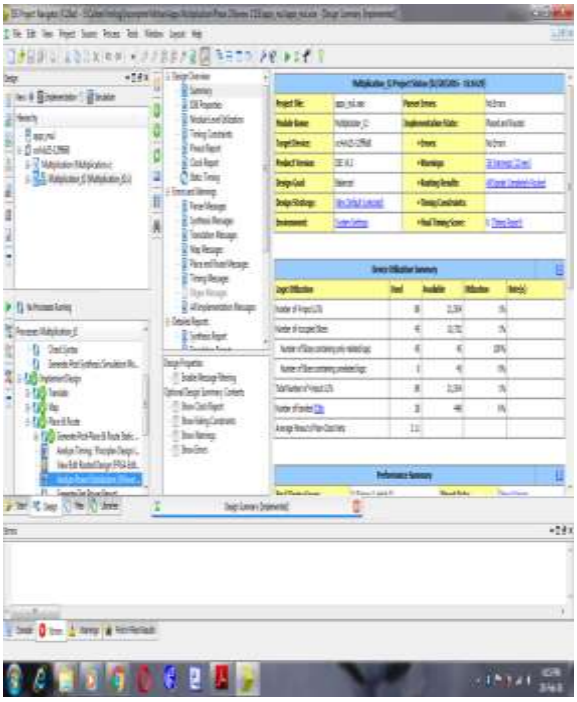

Fig 6 Area Utilisation

Fig 7 Power Analysis

V. CONCLUSION

In this paper, an analytical framework has been proposed for characterizing approximate adder designs. This framework consists of models for the evaluation of three different types of approximate adders targeting several error metrics. In this paper dada multiplier is performed withy carry select adder The comparison result shows that the modified one reduces the number of half adders by 80%.. But Wallace and Modified Wallace reduction use more gates for their reduction than Dadda multiplier. CSA is introduced in the final carry propagation path of the multipliers. From all the comparison results we can conclude that the Dadda multiplier with CSA in the final carry propagation path is more efficient.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their detailed comments and suggestions which helped to improve the quality of the paper. Finally, I would like to thank my family members for their continuous motivation and support.

REFERENCES

[1] Cong Liu, Jie Han, Member, IEEE, and Fabrizio Lombardi, Fellow, IEEE 2015,."*An Analytical Framework for Evaluating the Error Characteristics of Approximate Adders ".*

[2] P.R. Cappello and K Steiglitz, "*A VLSI layout for a pipe-lined Dadda multiplier,*" ACM Trans. Computer Systems, pp. 157-174, 1983.

[3] S. Nakamura, "*Algorithms for iterative array multiplication*," IEEE Trans. Computers, vol. 35, pp. 713-719, 1986.

[4] D.G. Crawley and G.A.J. Amaratunga, "8 *x 8 bit pipelined Dadda mul- tiplier in CMOS*," IEEE Proc., vol. 135, pp. 231-240, 1988.

[5] B. Maden and C.G. Guy, "*Parallel architectures for high speed multipliers,*" Proc. ISCAS'89, pp. 142-145, 1989.s

[6] R. A. Horn, "*The hadamard product*," in Proc. Symp. Appl. Math, vol. 40, 1990, pp. 87–169.

[7] R. Hegde and N. Shanbhag, "*Soft digital signal processing," Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on, vol. 9, no. 6, pp. 813–823, Dec 2001.

[8] M. Breuer, "*Intelligible test techniques to support error-tolerance,"* in Test Symposium, 2004. 13th Asian, 2004, pp. 386–393.

[9] S.-L. Lu, "*Speeding up processing with approximation circuits*," Com- puter, vol. 37, no. 3, pp. 67–73, 2004.

[10] M. S. Lau, K.-V. Ling, and Y.-C. Chu, "*Energy-aware probabilistic multiplier: design and analysis,"* in Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems. ACM, 2009, pp. 281–290.Transactions on, vol. 18, no. 4, pp. 517–526, April 2010.

[11] Hore and D. Ziou, "*Image Quality Metrics: PSNR vs. SSIM*," in 2010 20th International Conference on Pattern Recognition. IEEE, Aug. 2010, pp. 2366–2369.

[12] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "*Design of voltage-scalable meta-functions for approximate computing,"* 2011 Design, Automation & Test in Europe, pp. 1–6, Mar. 2011.

[13] P. Varman, and K. Mohanram, "*High performance reliable variable latency carry select addition,"* in Design, Automation Test in Europe Conference Exhibition (DATE), 2012, 2012, pp. 1257–1262.

[14] J. Huang, J. Lach, and G. Robins, "*A methodology for energy-quality tradeoff using imprecise hardware*," in Proceedings of the 49th Annual Design Automation Conference. ACM, 2012, pp. 504–509.

[15] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "*Low-power digital signal processing using approximate adders,"* Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 32, no. 1, pp. 124–137, 2013.

[16] J. Han and M. Orshansky, "*Approximate Computing: An Emerging Paradigm For Energy-Efficient Design,"* in ETS'13, Proc. of the 18th IEEE European Test Symposium, Avignon, France, May 2013.

[17] J.Y. Lee, H.L. Garvin, and C.W. Slayman, "*A high-speed high-density silicon 8 x 8-bit parallel multiplier,"* IEEE J. Solid-state Circuits,2013.