Design Of Approximate Multiplier With Tradeoff In Power And Area

[1]I.Rinisha Prem Priya, [2]Dr.J. Ajayan
[1]PG scholar, [2]Associate professor,
[1]Dept of Electronics & Communication Engineering, [1]SNS College of Technology, Coimbatore -35, India.

Abstract - Approximation circuit's offer superior performance (delay and area) compared to traditional circuits at the cost of computational accuracy. Three stages can be identified in a multiplier: partial product generation, partial product reduction, and final addition. The state-of-the-art approximate multipliers that provide the best trade-off between quality and other design parameters such as power and delay. The proposed approximate multiplier achieves a improved area and delay. By improving delay and area the performance of BAM multiplier is improved. Approximate unsigned multipliers are comparatively evaluated for both error and circuit characteristics.

Keywords - Approximate computing circuits and systems,broken array multiplier,approximate multipliers.

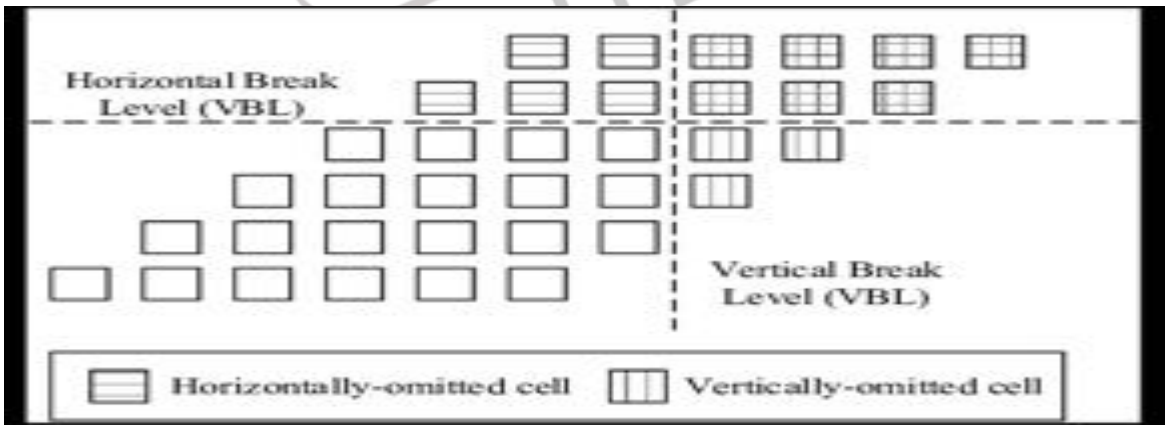
INTRODUCTION

Multipliers are the speed of multiply operation is of great importance in digital signal processing. In the past multiplication was generally implemented through a sequence of addition, subtraction, and shift operations. Multipliers play an important role in today's various applications.. Three stages can be identified in a multiplier: partial product generation, partial product reduction, and final addition. Four main methods are used for the design of approximate multipliers:

- Approximation in generating partial products based on a simpler structure;
- Approximation in the partial product tree by ignoring some partial products (truncation), dividing the partial products into several modules and applying an approximation in the less significant modules, or composing complex approximate multipliers from simple approximate multipliers;
- Using approximate adders, counters, or compressors in the partial product tree to reduce partial products;
- Using search-based methods to perform approximation on the gate level or in more complex cells. In the sequel, we briefly introduce the state-of-the-art approximate multipliers that provide the best tradeoff between quality and other design parameters such as power, delay, and area.

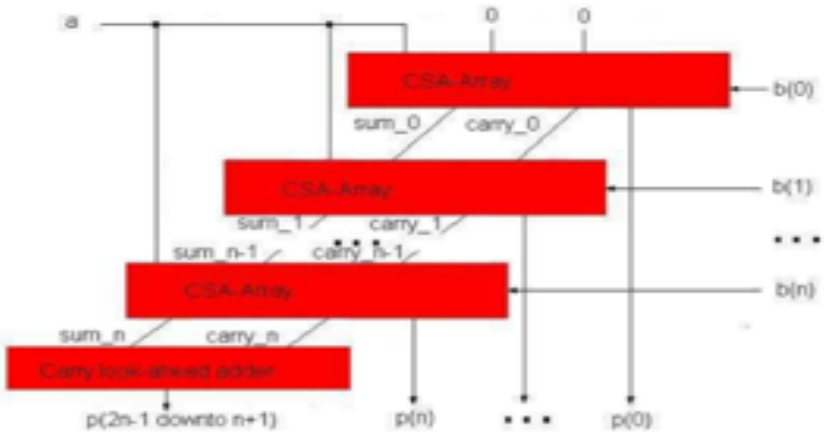
BROKEN-ARRAY MULTIPLIER (BAM),

In the BAM some of the carry-save adders are removed from an array multiplier. The omitted cells are specified using two parameters: the horizontal break level (h) and vertical break level (v), where $0 \leq h < n$ and $h \leq v < 2n$. An example of a BAM is shown in Fig. In the case that the vertical break level is 2× of the horizontal break level (i.e., v =2h), a structure similar to TM with k =v is obtained. As shown in our example, however, BAM preserves more carry-save adder cells [7].



Since the reduction of carry-save adders can be done in both the directions, the accuracy of BAM (n, h,v) depends on the three parameters. The multiplier and multiplicand are in this case labeled a and b respectively. The partial products are formed, either by AND'ing the appropriate bits or by an alternative method. In this multiplier the alternative method is used. Every CSA-array represents a partial product and therefore n CSA-arrays are needed to make a full multiplier where n is the width of the multiplicand. This way of ordering the CSA-arrays is called a chain. Notice the extra input b(i) compared to a normal CSA; This input serves the same purpose as the multiplier bit in the before mentioned alternative method and in this case as a gating parameter8 for a. Every CSA-array has three input numbers. The outputs of the first CSA

are used as inputs in the second CSA along with a and so on. Since the first CSA array has no previous CSA-arrays it has two 0's and of course a as input. This means that the first CSA-array can actually be replaced by very simple logic.



Carry-save adder (CSA) multiplier

As shown in Figure it is possible to combine the CSA-array with a Carry look–ahead adder. Each CSA-array outputs one bit giving the least significant half of the final product, while the final Carry look-ahead adder outputs the most significant half of the final product.

RELATED WORKS

Approximate circuits are becoming a viable alternative to conventional accurately operating circuits if energy efficiency is crucial and target application is error resilient. Approximate circuits such as adders and multipliers suitable for a particular application. We will only deal with functional approximation in which logic function implemented by the original circuits is simplified. Other approximation techniques enabling power reduction such as voltage over scaling are not considered in this paper. Unfortunately, almost all papers dealing with circuit approximation show some of the following features that are undesirable from a practical point of view:

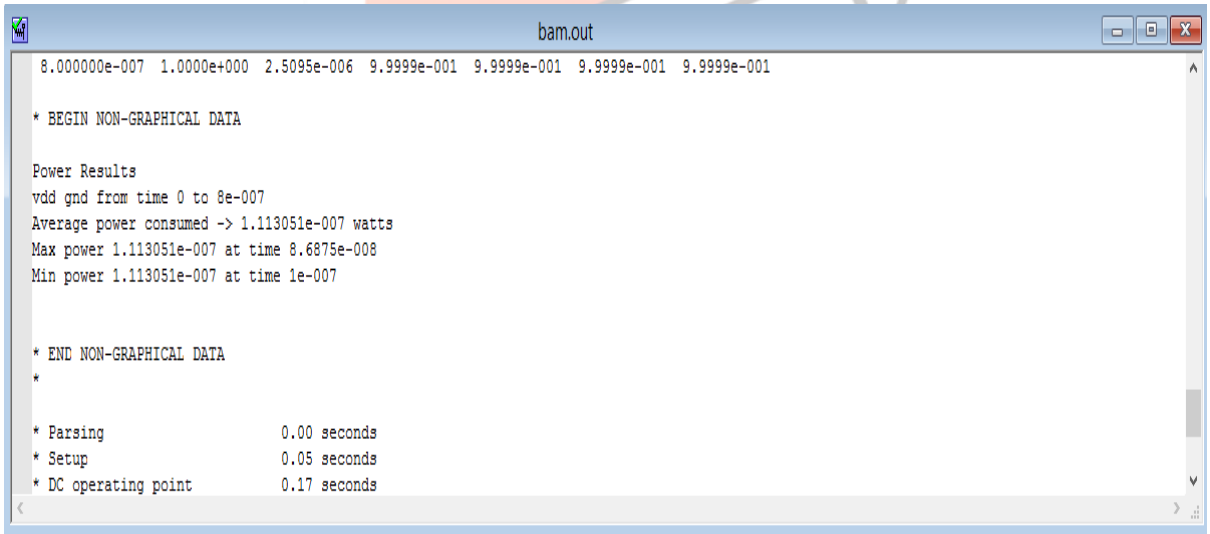
(1) The approximation method is described, but a corresponding software implementation is not available.

(2) An implementation of the accurate circuit is not available.

(3) The quality of approximation and other parameters of approximate circuits are expressed relatively to parameters of the original circuit.

(4) Implementations of the approximate circuits are not available

POWER ANALYSIS:



The Fig shows the power report of the new proposed design obtained in Tanner. It contains the total power consumed for the proposed approximate multiplier**.**

WAVEFORM

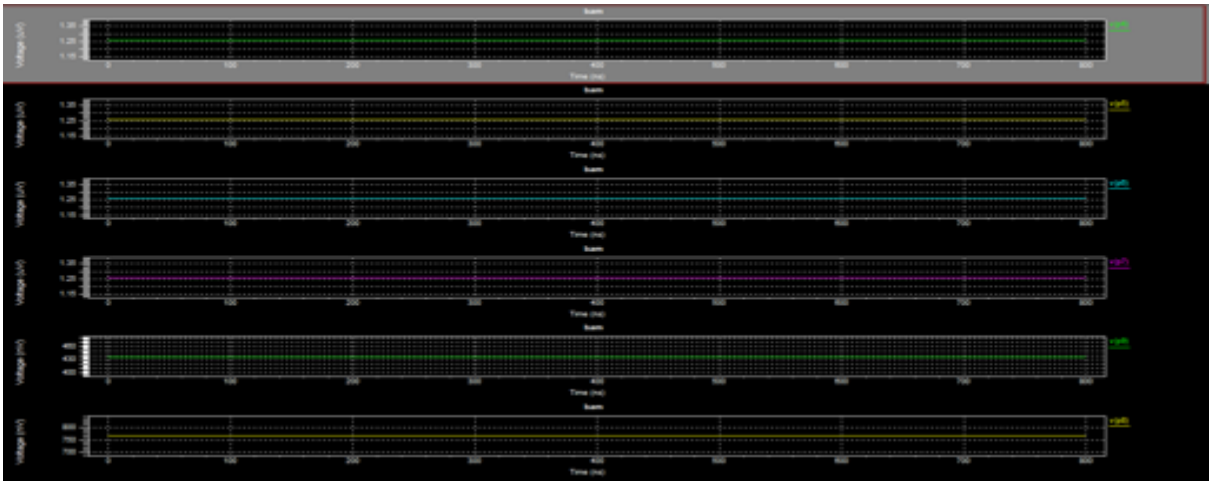


Fig shows the(waveform of broken array multiplier) shows the output result of the BAM. It contains the total number of various hardware resource utilized for the new proposed technique for multiplier.

CONCLUSION

A multiplier has a significant impact on the speed and power dissipation of an approximate processor. Precise results are not always required in many algorithms. Error-tolerant algorithms and applications have promoted the development of approximate multipliers to tradeoff accuracy for speed, implementation area and/or power efficiency. The proposed work has just analyzed the delay and power of approximate multiplier.

REFERENCES

[1] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits,*J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 4, 2017, Art. no. 60.

[2] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "EvoApproxSb: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Proc. DATE*, Mar. 2017, pp. 258–261.

[3] C. Yu and M. Ciesielski, "Analyzing imprecise adders using BDDs—A case study," in *Proc. ISVLSI*, Jul. 2016, pp. 152–157.

[4] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficien approximate wallace tree multiplier for error-resilient systems," in *Proc. 15th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2014, pp. 263–269.

[5] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in *Proc. DAC*, Jun. 2015, pp. 1–6

IJEDR