DESIGN OF APPROXIMATE MULTIPLIER FOR HIGH SPEED APPLICATION

[1] Dr.J. Ajayan,, [2]I.Rinisha Prem Priya,
[1] Associate professor, [2] PG scholar ,
[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. Multiplication is a key fundamental function for many worst case error applications. 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, delay, and area. In existing system a scalable recursive method for the construction of large approximate multipliers with guaranteed WCE was proposed. For configurable architectures (AM, Truncated Array Multiplier (TAM)), all meaningful configurations were considered. The error metrics are evaluated accurately using all test vectors for all considered designs. For truncated multipliers (TMs), the key idea is to remove k least significant bits of the input operands. As a result, a smaller (n−k)-bit multiplier is utilized instead of an accurate n-bit multiplier. Despite the fact that only the WCE, power and area were considered to obtain non dominated design**.**

Index Terms—Approximate computing circuits and systems, circuit synthesis, circuits, computers and information processing.

I.INTRODUCTION:

Approximate circuits are becoming a viable alternative to conventional accurately operating circuits if energy efficiency is crucial and target application is error resilient [1]. This is the case of many, image and video processing circuits that are predominantly composed of adders and multipliers. In order to approximate circuits for a particular application, a designer can either perform a single purpose (ad hoc) approximation or apply some of the circuit approximation methods. We will only deal with functional approximation in which logic function implemented by the original circuits is simplified. 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. If the accurate circuit is not available, it is impossible to obtain real parameters of the approximate circuits and reproduce the results.
(4) Resulting approximate circuits are not available.

(5) Only a few approximate versions created from the accurate circuit are reported, forming thus a sparsely occupied Pareto front.

(6) It is unclear if a given number of test vectors used to evaluate approximate circuits is sufficient for obtaining a trustworthy error quantification if the error is determined using simulation.
(7) A given approximation method is only rarely compared against other approximation methods.
Energy efficiency is a major challenge for current computer systems. Approximate implementations of multipliers are based on various design principles [1]. The major weakness of the manual circuit design

approach, which is clearly dominating in this area, lies in providing only a few different circuit implementations for a given bit width. Many interesting and useful design points thus remain unexplored. Hence, the automated search-based design methods have been developed to provide many approximate designs showing high-quality trade-offs between key design parameters.

II.RELATED WORK

Many researchers are trying to design multipliers which offer the following design targets – high speed, low power consumption, and hence less area and compact VLSI implementation. 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;

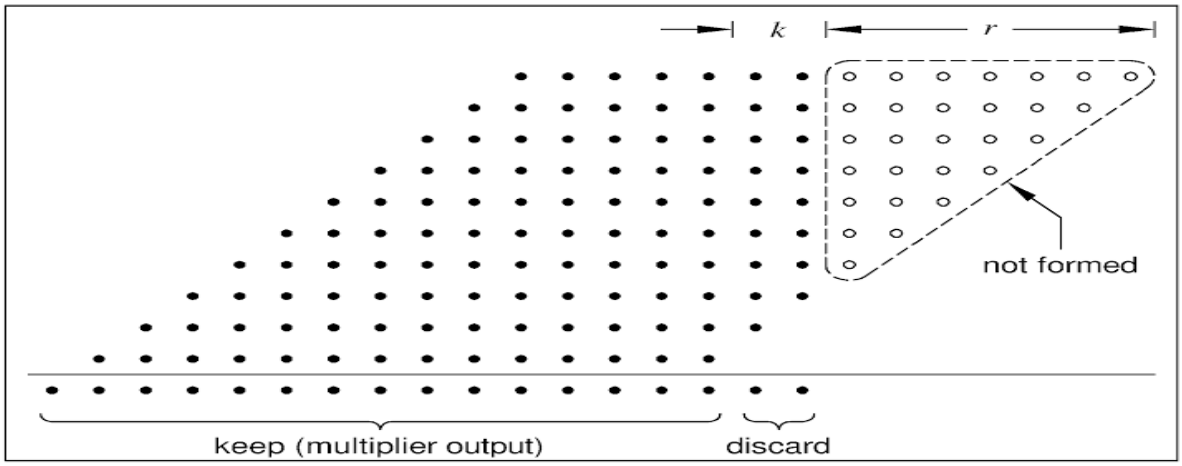
- Using approximate adders, counters, or compressors in the partial product tree to reduce partial products;

- 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.

III.TRUNCATED ARRAY MULTIPLIER (TAM)

For truncated multipliers (TMs), the key idea is to remove k least significant bits of the input operands. As a result, a smaller (n−k)-bit multiplier is utilized instead of an accurate n-bit multiplier. A 5-bit approximate multiplier is implemented as a truncated carry save adder array in Fig.4.1. In general, an array multiplier consists of n×n cells used to reduce partial products, followed by a single n-bit merging adder for the final summation [5]. Due to the truncation, the cells associated with k least significant bits of the first operand and the cells associated with k least significant bits of the second operand (i.e., k rightmost cells of each row) are omitted [5]. As a result, 2k least significant bits of the final product are always zeros. The accuracy of a TM depends on the bit width (n) and the number of truncated bits (k), where $0 \leq k < n$. The maximum difference between the output of TM (n,k) and a precise multiplier is equal to

$$\mathbf{WCE}_{TM(n,k)} = (2^k - 1)(2^{n+1} - 2^k - 1)..$$



Truncated Array Multiplier

To reduce delay various techniques have been used and in this project two are especially prominent, the Carry-save adder and the Carry look-ahead adder. The area used is actually the same as for a ripple-carry adder after the area optimization algorithm is performed by Xilinx ISE but the speed is increased.

- To reduce the overall area consumption level.
- To optimize the total time consumption process.
- To reduce the error presentation in multiplier output results.
- To improve the accuracy level compare to existing methodology.

IV.SYSTEM MODEL:

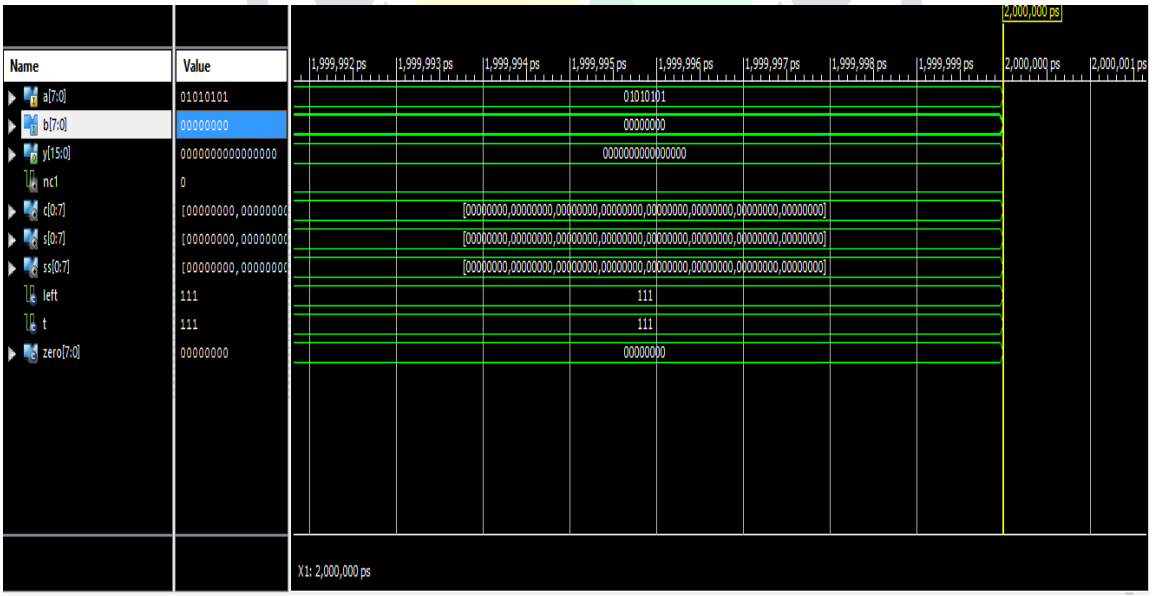
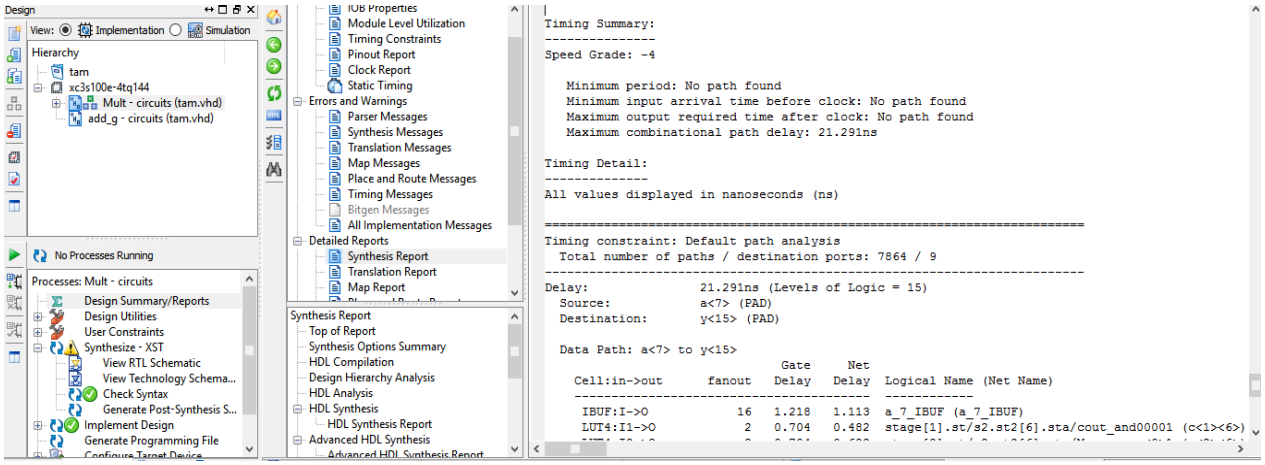


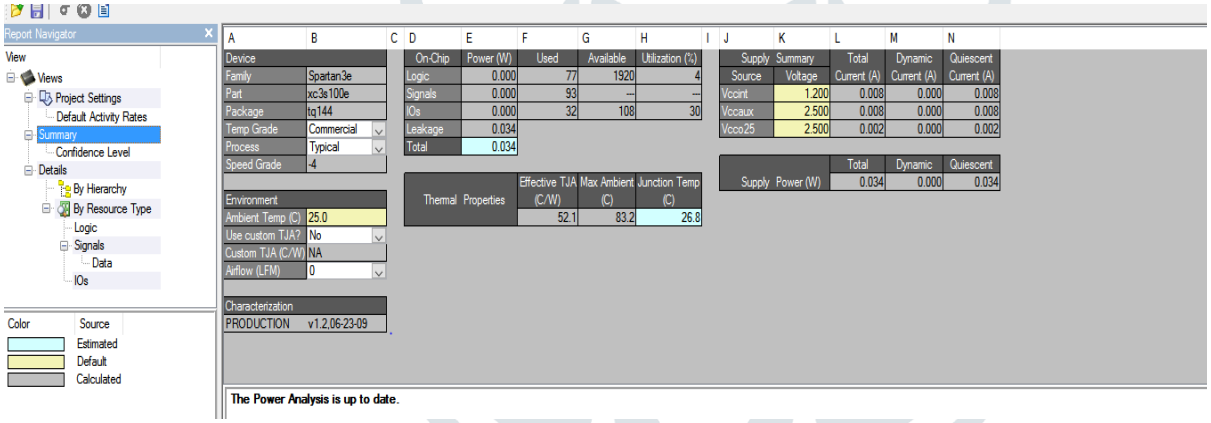
Fig shows a and b is the input of the multiplier C in is the carry and sum is the final adder output. A 5-bit approximate multiplier is implemented as a truncated carry save adder array. In general, array multiplier consists of n × n cells (i.e., the carry-save adder array) used to reduce partial products, followed by a single *n*-bit merging adder (a ripple carry adder in our example) for the final summation.

a) **DELAY ANALYSIS**

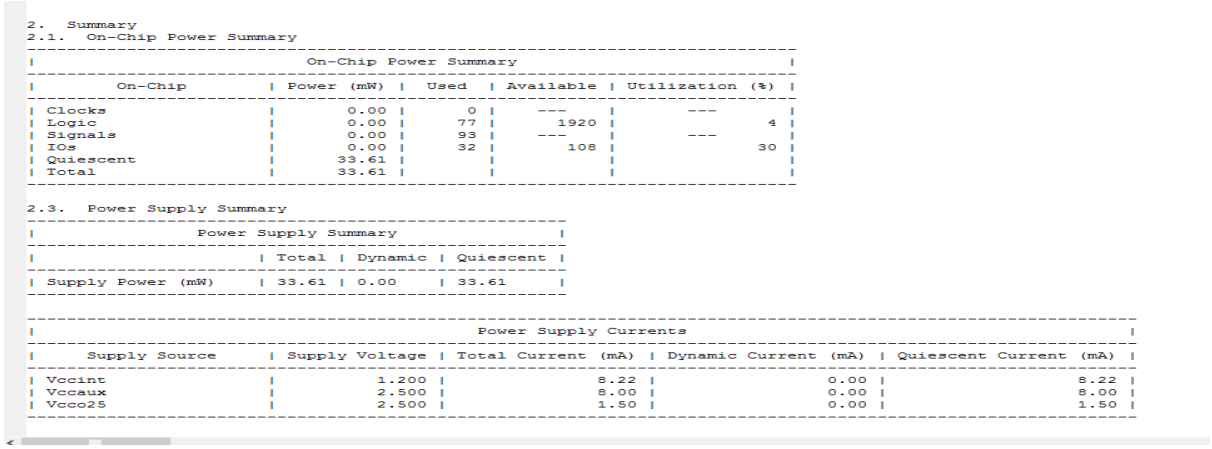


The Fig shows the delay analysis of approximate multiplier obtained in Xilinx ISE. It contains the total combinational path delay of the design. Due to the combinational path delay of the proposed system is reduced compared with the existing method.

b) **POWER ANALYSIS**



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



The Fig shows the power supply of AM and supply current of proposed approximate multiplier which is obtained by Xilinx ISE.

CONCLUSION:

The proposed approximate multiplier achieves a improved area and delay using Xilinx ISE. By improving delay and area the performance of multiplier is improved. Approximate unsigned multipliers are comparatively evaluated for both error and circuit characteristics. Among the considered approximate multipliers, truncation on part of the partial products is an effective way to reduce circuit complexity. When truncation is not used, multipliers approximated in the partial product tree tend to have a poor accuracy and moderate hardware consumption. To reduce delay various techniques have been used and in this project two are especially prominent, the Carry-save adder and the Carry look-ahead adder. The area used is actually the same as for a ripple-carry adder after the area optimization algorithm is performed by Xilinx ISE but the speed is increased. Hence the delay and power is analysed using xilinx ISE.

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] A. Chandrasekharan, M. Soeken, D. Große, and R. Drechsler, "Approximation-aware rewriting of AIGs for error tolerant applications," in Proc. ICCAD, Nov. 2016, pp. 1–8.
[5] Z. Vasicek, V. Mrazek, and L. S. Brno, "Towards low power approximate DCT architecture for HEVC standard," in Proc. DATE, Mar. 2017, pp. 1576–1581.
[6] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. ICCAD, Nov. 2011, pp. 667–673.
[7] 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. I, Reg. Papers , vol. 57, no. 4, pp. 850–862, Apr. 2010.
[8] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient approximate wallace tree multiplier for error-resilient systems," in Proc. 15th Int. Symp. Quality Electron. Design (ISQED), Mar. 2014, pp. 263–269.
[9] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," in Proc. DAC, Jun. 2015, pp. 1–6.