Design of Power and Area of Efficient Approximate Multipliers

S. Jaya Simha Kumar [1], P. Sandhya Rani [2]
[1] *Student, MTech(VLSI) Yoga Nanda Institute of Technology and Science*
[2] *Assistant Professor,(Guide) Yoga Nanda Institute of Technology and Science*

Abstract: Energy consumption has become one of the most critical design challenges in integrated circuit design. Arithmetic computing circuits, in particular array-based arithmetic computing circuits such as adders, multipliers, squarer's, have been widely used. Hence, reduction of energy consumption of array- based arithmetic computing circuits is an important design consideration. To this end, designing low-power arithmetic circuits by intelligently trading off processing precision for energy saving in error-resilient applications such as DSP, machine learning and neuromorphic circuits provides a promising solution to the energy dissipation challenge of such systems.
To solve the chip's energy problem, especially for those applications with inherent error resilience, array-based approximate Multipliers circuits (AAMC) circuits that produce errors while having improved energy efficiency have been proposed.

Index Terms – Introduction, Multipliers, Previous works, Proposed multiplier designs, Experimental results, Conclusion

I. INTRODUCTION

A new field of research was set up in recent years to explore ways in which computer systems can be more powerful, faster and less complex by relaxing the right requirement. This field, meant as rough registering, misuses the way that numerous applications are mistake versatile and the blunders in processing are hence either undetectable or worthy. The idea of estimate has seriously been considered, created and applied in software engineering, yet in addition in arithmetic and designing controls. Nonetheless, it has never been applied in the zones in which just exact usage have generally been acknowledged. These days, the architects deliberately acquaint blunders into calculation with fulfill the ceaseless prerequisite for bringing down of intensity utilization

As one of the most encouraging energy-effective registering ideal models that can adapt to ebb and flow difficulties of PC designing, surmised processing has increased a great deal of exploration consideration in the previous scarcely any years. We can recognize two fundamental bearings in rough registering: energy-productive figuring with problematic segments and estimate of frameworks executed on normal stages. In the primary case, the issue is that the specific calculation using nanometer semiconductors gave by late innovation hubs is very costly regarding energy necessities and solid conduct. An open inquiry is the way to adequately and dependably register with countless problematic parts. The second examination heading is inspired by the way that numerous applications (commonly in the zones of media, illustrations, information mining and enormous information preparing) are inalienablyblunder strong. This strength can be misused so that the mistake is traded for enhancements in power utilization, throughput or execution cost

II.MULTIPLIERS

BASICS Multiplication is a mathematical procedure, which is an abbreviated method in which a whole number of times is applied to itself. A number (multiplicand) is added to itself several times to generate a result (product) as indicated by a different number (multiplier).

Digital multiplication consists of three basic steps, these are:-
1)Generation of Partial Product Array
2)Reduction of Partial Product Array
3)Final Addition

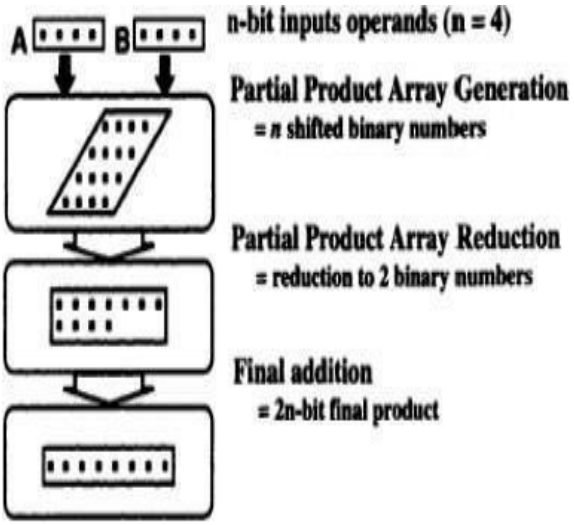


Figure 2.1 Multiplication Flow

III.PREVIOUS WORK

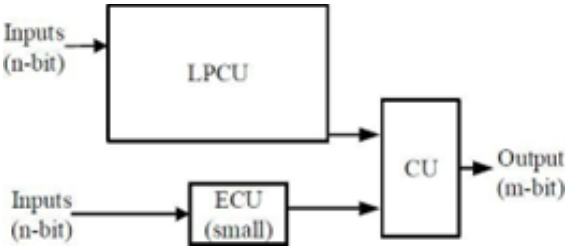
Approximate multipliers:

To generate partial products and Booth multipliers, AND gates are used in AND multipliers, to reduce the number of components uses the modified Booth algorithm, there are two kinds of approximate multipliers. For estimated AND-array multipliers variable & constant corrections schemes are suggested. Constant correcting scheme proposes to add one constant in order to offset the truncated error and to add multiple signals to the table of partial product in order to compensate for the variable corrections.

Approximate squarer's:

There was a sequence of estimated squarers. For example, by using constant and variable correction schemes, the design and compensation of the truncation error. By using a hybrid LUT-basedsquarer is suggested.

Approximate multiplier circuits (Array based):



Fig(a)is Error free computing unit and Fig(b) is proposed AAAC model

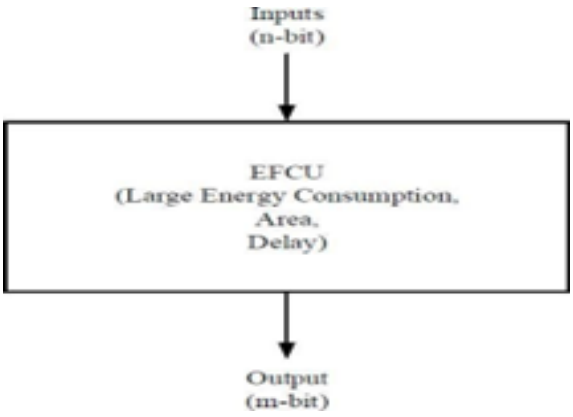


Figure 3.1 (a)

$$E_{ave} = \frac{1}{2^n \cdot N}\sum_{i=1}^{N}|O_{AAAC,i} - O_{EFCU,i}| \quad (1)$$

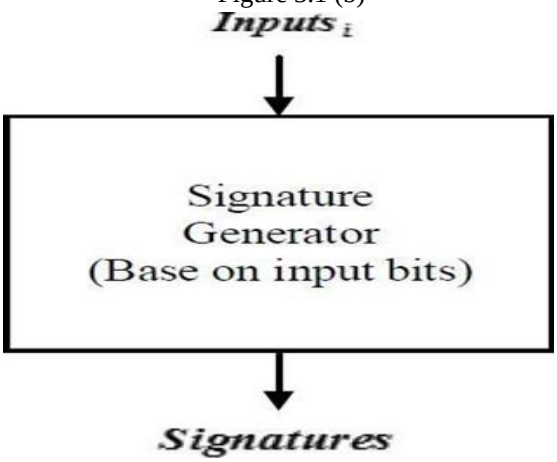
$$E_{max} = \frac{1}{2^n}\max_{i}|O_{AAAC,i} - O_{EFCU,i}| \quad (2)$$

$$E_{ms} = \frac{1}{2^{2n} \cdot N}\sum_{i=1}^{N}(O_{AAAC,i} - O_{EFCU,i})^2 \quad (3)$$

$$E_{BC,i} = O_{EFCU,i} - O_{LPCU,i} \quad (4)$$

$$E_{AC,i} = |E_{BC,i} - Comp_i| \quad (5)$$

Figure 3.1 (b)



Error Metrics:

Error compensation unit model:

Preferably, a particular can be figured by the ECU to totally zero out the mistake for each information design I. Be that as it may, this doesn't fill any need for estimated processing as we are basically re-actualizing the mistake free activity. We present a handy yet broad ECU model, which comprises of a Signature Generator (an) and a K-to-1 Mux (b), as appeared in Figure underneath.

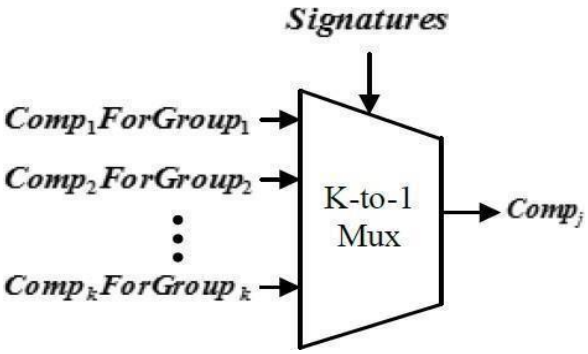


Figure 3.1 (c)

Theoretically, for a given information design I, the mark generator delivers a few marks that encode certain fundamental data about the data sources.

IV.PROPOSED MULTIPLIER DESIGN

4.1 Booth Multiplier Fundamentals:

For high-speed applications standard multipliers are used and the Radix-4Adapted Booth Multipliers are most commonly used. Figure shows the key blocks of a changed booth Radix-4 multiplier. The Radix 4 Booth Algorithm is used by the encoding block to encode the multiplier B

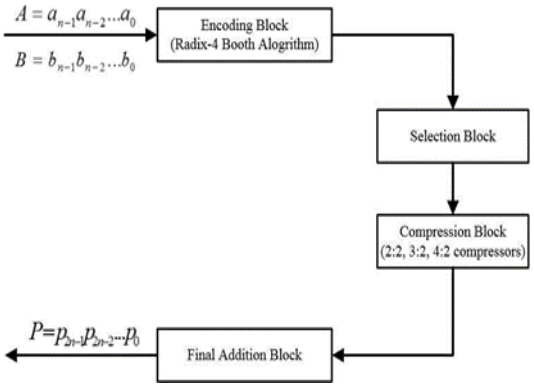


Figure 4.1: Error-Free Booth multiplier blocks.

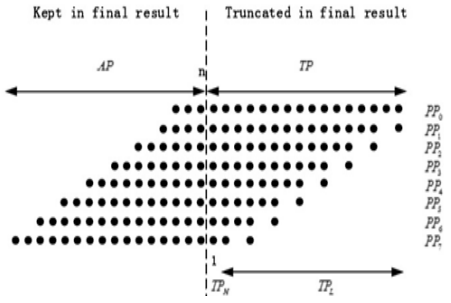


Figure 4.1.1: Partial product for n=16 fixed width Booth multiplier

4.2 Signal generator:

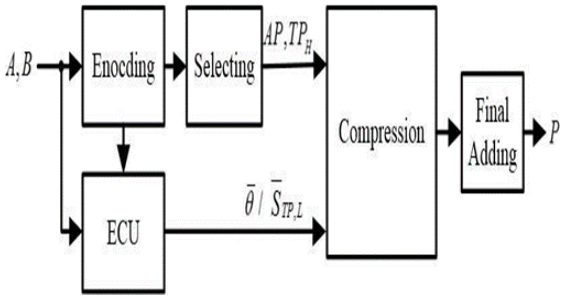


Figure 4.2: Blocks and schematics of Signature Generator

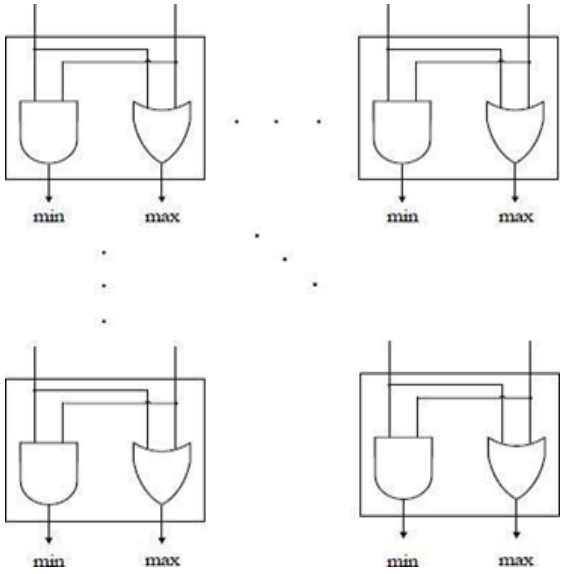


Figure 4.2.1 Blocks of sorting network.

Table 4.2. Compensation for input groups of 16x16 multiplier

Case		
1	1	0.9853
2	2	1.1259
3	2	1.0188
4	1	0.8580
5	0	0.4001

4.3 Proposed Full-Width Booth Multiplier:

Inexact full-width multipliers, I. e., ones that inexact exact nxn Booth multipliers by yielding a full-width 2n-bit estimated item, are additionally helpful for some pragmatic applications. The introduced fixed-width configuration can be promptly stretched out to encourage full-widthactivity with the distinction being that for this situation we might want to approximate

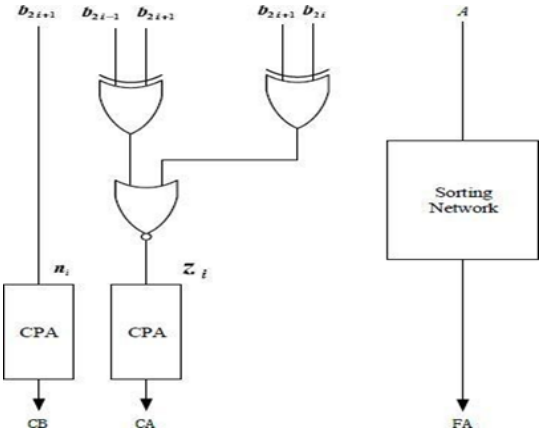


Figure 4.3: Design blocks of the Complete proposed multipliers

V.EXPERIMENTAL RESULTS

By using Verilog HDL, we implement our 16bit fixed Booth multiplier and squarer are designed using synopsis Design Complier with commercial 90nm CMOS technology and standard cell library. By using 2:2, 3:2, 4:2 compressors we compressed the generated all partial products

Approximate multipliers Formula for accuracy analysis:

$$\frac{|E_{existing} - E_{prposed}|}{E_{existing}} \times 100\%$$

Comparison with Various Multipliers:

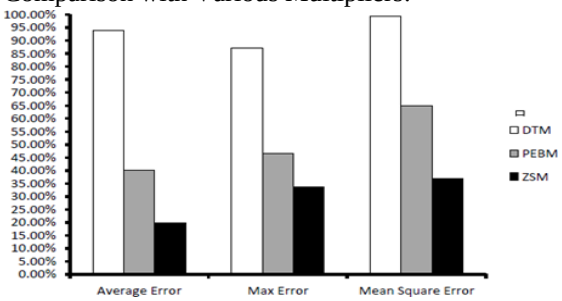


Figure 5.1: Error reduction of the proposed 32x32 fixed-width Boothmultiplier over DTM, PEBM, ZSM

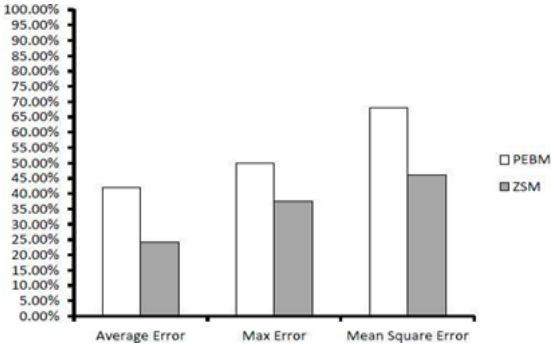


Figure 5.2: Error reduction of the proposed 16x16 full-width Booth multiplier overPEBM, ZSM

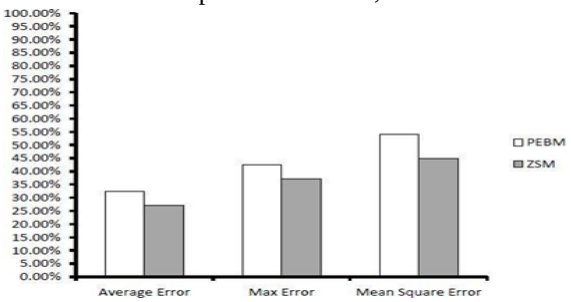


Figure 5.3: Error reduction of the proposed 32x32 full-width Booth multiplier overPEBM, ZSM

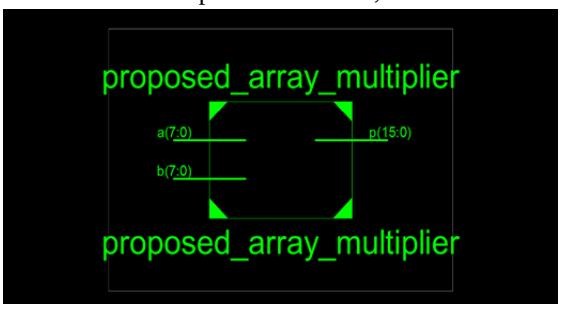


Figure 5.4: RTL top level

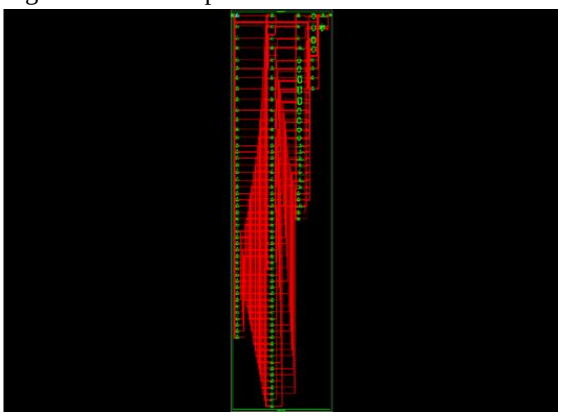


Figure 5.5: RTL Schematic approximate multiplier

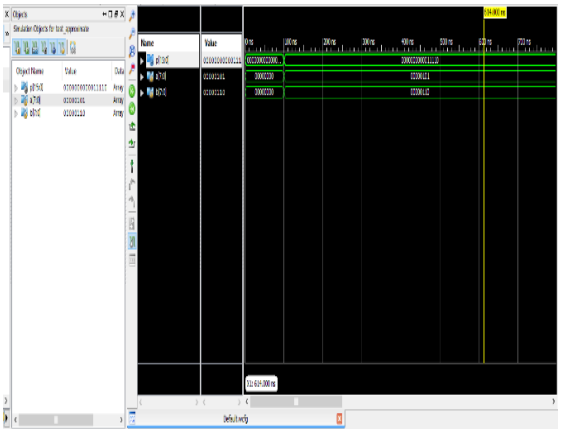


Figure 5.6: Simulation for approximate multiplier result

VI.CONCLUSION

Our proposed fixed width Booth multiplier of 16 bit requires 44.85% of less energy and 28.33% less area when compared to normal or traditional accurate fixed width Booth multiplier. In addition to this when we compared with appropriate traditional devices, we also reduce the overall average error of 11.11%, mean square error of 25% and maximum error of 28.11%. With respect to normal accurate devices our16x16 bit proposed approximate squarer decreases the maximum error of 21.67%, average error of 18.18%, mean square error of 31.25%. By introducing extra don't cares and signatures we reduce error and cost of the design even more. When operated in full width mode the proposed multipliers and squarer's also achieved improvement in error significantly.

REFERENCE

[1] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. CoRR, abs/1412.7062, 2014.

[2] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan. Analysis and characterization of inherent application resilience for approximate computing. In DAC '13, pages 113:1–113:9, 2013.

[3] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. ICDAR, 2011.

[4] K.-L. Du and M. Swamy. Neural Networks Softcomputing Framework. Springer London, 2006.

[5] Z. Du, K. Palem, A. Lingamneni, O. Temam, Y. Chen, and C. Wu. Leveraging the error resilience of machine-learning applications for designing highly energy efficient accelerators. In ASP-DAC '14, 2014.

[6] G. Hinton, L. Deng, D. Yu, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6):82–97, Nov 2012.