Power and Speed Efficiency of Multipliers

VEDIKA.S[1] and DR.S.UMA MAHESWARI[2]

[1]M.E Research Scholar Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore, Email: vedikascit@gmail.com

[2] Professor Electronics and Communication Engineering, Coimbatore Institute of Technology,Coimbatore Email: umamaheswari@cit.edu.in

***Corresponding Author**
VEDIKA .S
M.E Research Scholar, Electronics and Communication Engineering,
Coimbatore Institute of Technology, Coimbatore,
Email: vedikascit@gmail.com

ABSTRACT In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Approximate computing can decrease the design complexity with an increase in performance and power efficiency for error resilient applications. This brief deals with a new design approach for approximation of multipliers. The partial products of the multiplier are altered to introduce varying probability terms. Logic complexity of approximation is varied for the accumulation of altered partial products based on their probability. Meanwhile, a significant proportion of the power consumption of an adder is due to the glitches that are caused by the carry propagation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption can be achieved. The partial product addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) and normal addition method is applied based on 4:2 compressor. This is to preserve its correctness since the higher order bits play a more important role than the lower order bits.

Keywords: low power, low error, multipliers, speed

I. INTRODUCTION

Arithmetic units such as adders and multipliers are key components in a logic circuit. The speed and power consumption of arithmetic circuits significantly influence the performance of a processor. High-performance arithmetic circuits such as Carry Look Ahead Adders (CLAs) and Wallace tree multipliers have been widely utilized.[1] However, traditional arithmetic circuits that perform exact operations are encountering difficulties in performance improvement.

Approximate arithmetic that allows a loss of accuracy can reduce the critical path delay of a power consumption and area overhead. Thus, approximate arithmetic is advocated as an approach to improve the speed, area and power efficiency of a processor due to the error-resilience of some algorithms and applications. As a significant arithmetic module, the multiplier has been redesigned to lots of approximate version. [2]The often conflicting advantages and disadvantages of these designs make it difficult to select the most suitable approximate multiplier for a specific application.

Thus, approximately redesigned multipliers are reviewed in this paper and a comparative evaluation is performed by considering both the error and circuit characteristics. Energy minimization is one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is highly desired to achieve this minimization with minimal performance (speed) penalty.

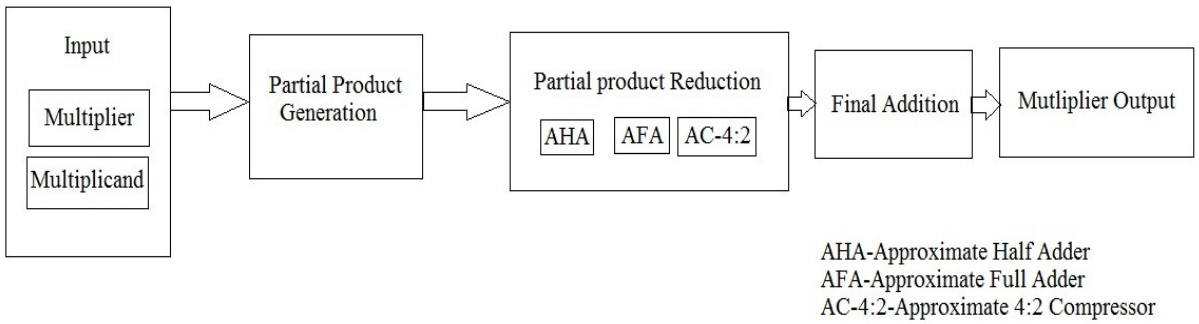

Figure 1: power and area efficient of approximation multiplier

Digital Signal Processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems. [3]Therefore, improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors.

Applying the approximation to the arithmetic units can be performed at different design abstraction levels including circuit, logic, and architecture levels, as well as algorithm and software layers. [4]The approximation may be performed using different techniques such as allowing some timing violations (e.g., voltage over scaling or over clocking) and function approximation methods (e.g., modifying the Boolean function of a circuit) or a combination of them . In the category of function approximation methods, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested.

II. RELATED WORK

The author Kartikeya Bhardwaj, Pravin S. Mane and J¨org Henkel proposed a bit-width aware approximate multiplication algorithm for optimal design of our multiplier. [5]We employ a carry-in prediction method to reduce the critical path. It is further augmented with hardware efficient pre computation of carry-in. We also optimize our multiplier design for latency, power and area using Wallace trees. Accuracy as well as LPA design metrics are used to evaluate our approximate multiplier designs of different bit-widths, ***i.e***. 4 × 4, 8 × 8 and 16 × 16.

The simulation results show that we obtain a mean accuracy of 99.85% to 99.965%. Single cycle implementation of AWTM gives almost 24% reduction in latency. We achieve significant reduction in power and area, ***i.e***. Up to 41.96% and 34.49% respectively that clearly demonstrates the merits of our proposed AWTM design. Finally, AWTM is used to perform a real time application on a benchmark image. We obtain up to 39% reduction in power and 30% reduction in area without any loss in image quality.

The author Yuke Wang, Yingtao Jiang and Edwin Sha proposed we present an improved architecture for a multiplexer-based multiplication algorithm Also throughout intensive HSPICE simulation, it has been shown in this paper that due to smaller internal capacitance, multiplexer-based array multiplier outperforms the modified Booth multiplier in both speed and power dissipation by 13% to 26%. [6]In addition, we demonstrate that using area -efficient full adder circuits (SERF and 1OT 1111) can help reduce the overall routing capacitance, resulting in less power consumption on multipliers built upon those adder circuits. Therefore, multiplexer-based multiplier following suggested architecture along with area efficient full adder circuits can be used for low power high performance parallel multiplier designs.

The author V. Sai Priya and R. Suga Priya proposed approximation of a multiplier offers an effective approach to obtain low hardware utilization. The adaptive filter used to convert the value in binary. As constrained by a low error rate, all three designs have a very high accuracy, so both truncation and approximation are used in the multiplier design to further reduce power, area and delay. [7]In proposed two techniques are used Tabu search algorithm and scheduling algorithm. The designs are used to improve the error rate and increase the performance. This approximation technique enables the parameters like high area and power savings while retaining high accuracy. We explored product perforation on a large set of multiplier architectures, evaluating its impact on different architectures and error bounds. Finally, these techniques are scalable, offering better results as the multiplier's bit width increases. An approximate multiplier that trades off accuracy and energy has been proposed. [8]The proposed method of approximate multiplier takes m consecutive bits (i.e., an m-bit segment) of an n-bit operand either starting from the MSB or ending at the LSB and applies these two segments that include the leading ones from two operands.

The author Venkatachalam, Hyuk Jae Lee and Seok-Bum Ko The partial product generation and partial product accumulation circuits. Radix-4 partial product generation and accumulation approximation is proposed which remarkably enhances the performance. [9]The proposed approximate booth multiplier achieves 41% area reduction and 49% power reduction compared to an exact booth multiplier. Also, it has better area, power and error metrics compared to existing works on approximate multipliers. [10]The proposed multiplier is evaluated with an image processing application- in Discrete Cosine Transform (DCT) encoding part of JPEG compression and found to perform almost similar to exact multiplication unit.

III. EXISTING METHOD

The multiplier comprises three steps: generation of partial products, partial products reduction tree, and finally, a vector merge addition to produce final product from the sum and carry rows generated from the reduction tree. Second step consumes more power. The approximation is applied in reduction tree stage.

Using OR gate in the accumulation of column wise generate elements in the altered partial product matrix provides exact result in most of the cases. The probability of error (Perr) while using OR gate for reduction of generate signals in each column. As can be seen, the probability of misprediction is very low. As the number of generate signals increases, the error probability increases linearly. However, the value of error also rises. To prevent this, the maximum number of generate signals to be grouped by OR gate is kept at 4. For a column having m generate signals, _m/4_ OR gates are used.

In 4-2 compressor, three bits are required for the output only when all the four inputs are 1, which happens only once out of 16 cases. This property is taken to eliminate one of the three output bits in 4-2 compressor. To maintain minimal error difference as one, the output “100" (the value of 4) for four inputs being one has to be replaced with outputs “11" (the value of 3). For Sum computation, one out of three XOR gates is replaced with OR gate. Also, to make the Sum corresponding to the case where all inputs are ones as one, an additional circuit x1 · x2 · x3 · x4 is added to the Sum expression. This results in error in five out of 16 cases. Carr y is simplified as in (4).

$W1 = x1 \cdot x2$

$W2 = x3 \cdot x4$

$Sum = (x1 \oplus x2) + (x3 \oplus x4) + W1 \cdot W2$

$Carr\ y = W1 + W2.$

IV. PROPOSED METHOD

We proposed the error-tolerant design can be a potential solution to this problem. By sacrificing some accuracy, the ETAC can attain great improvement in both the power consumption and speed performance. In a conventional partial product adder circuit, the delay is mainly attributed to the carry propagation chain along the critical path, from the least significant bit (LSB) to the most significant bit (MSB). Meanwhile, a significant proportion of the power consumption of an adder is due to the glitches that are caused by the carry propagation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption can be achieved. We first split the overall partial products into two parts: an accurate part that includes several higher order bits and the inaccurate part that is made up of the remaining lower order bits. The length of each part need not necessary be equal. The addition process starts from the middle joining point of the two parts.

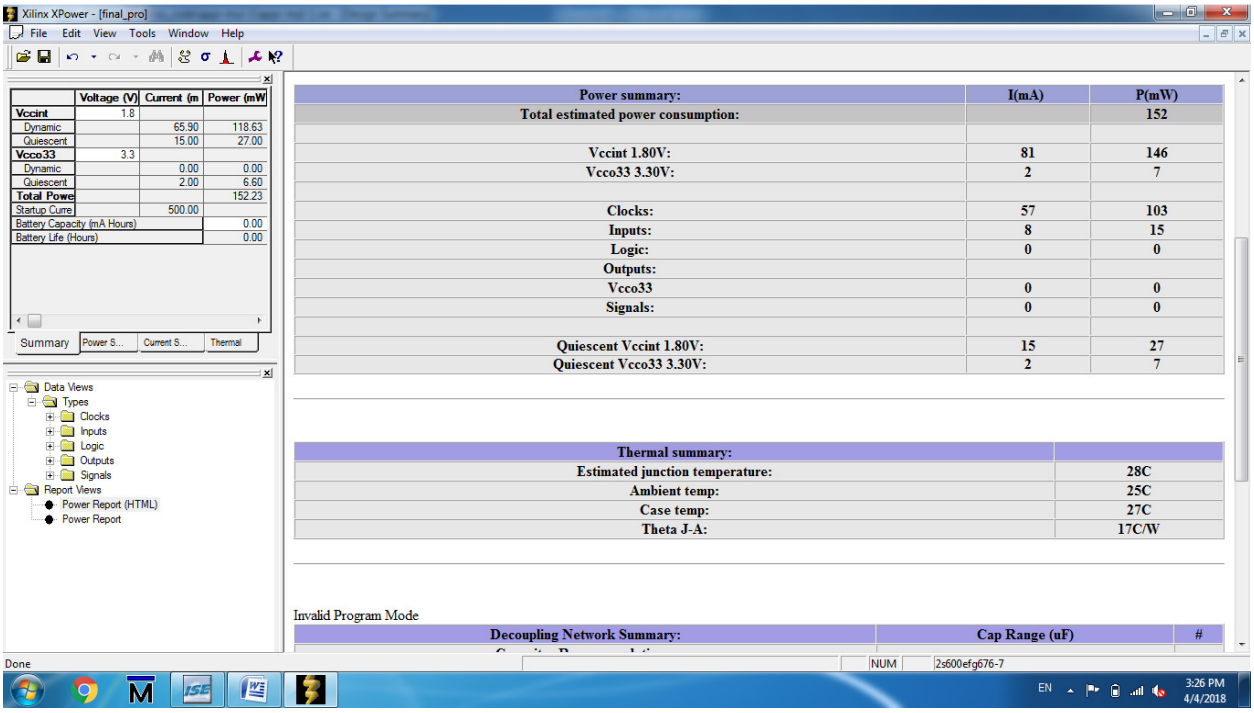

Figure: 2 ETAC based power and speed consumption

v. EXPERIMENTAL RESULT

The partial product addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) and normal addition method is applied based on 4:2 compressor. This is to preserve its correctness since the higher order bits play a more important role than the lower order bits.

The lower order partial product bits of the multiplier (inaccurate part) require a special addition mechanism. No carry signal will be generated or taken in at any bit position to eliminate the carry propagation path. To minimize the overall error due to the elimination of the carry chain, a special strategy is adapted, and can be described as follow:

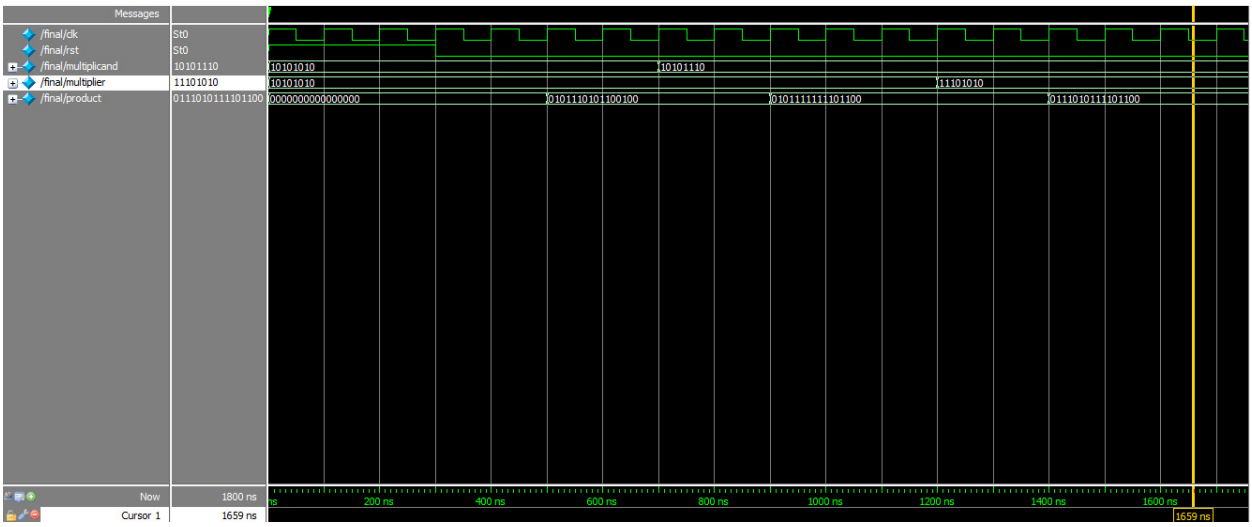

Figure: 3 The simulation result for LSB to MSB Approach

1) 2^0 to 2^5 column values is considered for approximation.
2) 2^6 to 2^14 column values are added based on existing computing mechanism using FAs, Has and 4:2 compressor.
3) In the LSB part a0, 0 is taken as product (0) while the columns 2^1 and 2^2 output product are generated by using 2 and 3 input OR gate respectively.
4) 2^3 and 2^4 columns uses 4-input and 5-input OR gate to produce product (3) and product (4).
5) The 2^4 columns 5-input OR gate along with 2 3-input OR gate of 2^5 column determine the carry propogation to the MSB part.
6) It is analyzed based on the following equation.

Product [0] = p0 [0];
Product [1] = p0 [1] | p1[0];
Product [2] = p0 [2] | p1[1] | p2[0];
Product [3] = p0 [3] | p1[2] | p2[1] | p3[0];
r1 = p0 [4] | p1 [3] | p2[2] | p3[1] | p4[0];
r2 = p0 [5] | p1[4] | p2[3];
r3 = p3 [2] | p4[1] | p5[0];
Product [4] = r1;
Product [5] = r2 | r3;

TABLE I COMPARISION OF RESULTS

Comparison Table:

Parameters	Existing Method	Proposed Method
Slices	78	66
FFs	64	64
LUTs	134	118
Gate Count	1956	1579
Power(mW)	197	152
Delay(ns)	8.664	10.047

VI. CONCLUSION

The proposed efficient approximate multipliers, partial products of the multiplier are modified using generate and propagate signals. Approximation is applied using simple OR gate for altered *generate* partial products. Approximate half-adder, full-adder, and 4-2 compressor are proposed to reduce remaining partial products. Two variants of approximate multipliers are proposed, where approximation is applied in all n bits in Multiplier1 and only in $n-1$ least significant part in Multiplier2.

Multiplier1 and Multiplier2 achieve significant reduction in area and power consumption compared with exact designs. They are also found to have better precision when compared to existing approximate multiplier designs. The proposed multiplier designs can be used in applications with minimal loss in output quality while saving significant power and area.

REFERENCES

1. Suganthi Venkatachalam and Seok-Bum Ko, Design of Power and Area Efficient Approximate Multipliers, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 2017:25(5):1782-1786.
2. Kartikeya Bhardwaj, Pravin S. Mane, Power- and Area-Efficient Approximate Wallace Tree Multiplier for Error-Resilient Systems, IEEE conference, 2014:5(1):1027-1036.
3. Bharat Garg and G. K. Sharma, Low Power Signal Processing via Approximate Multiplier for Error-Resilient Applications, IEEE International Conference, 2016:546-551.
4. Sunghyun Kim and Youngmin Kim, High-Performance and Energy-Efficient Approximate Multiplier for Error-Tolerant Applications, IEEE conference, 2017:278-279.
5. K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, Design of low-error fixed-width modified booth multiplier, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 2004:12(5):522-531.
6. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, Energy-efficient approximate multiplication for digital signal processing and classification applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*2015:23(6):1180-1184.
7. N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 2010:18(8):1225-1229.
8. 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*, 2013:32(1):124-137.
9. J. Park, J. H. Choi, and K. Roy, "Dynamic bit-width adaptation in DCT: An approach to trade off image quality and computation energy," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2010:18(5):787-789.
10. S. L. Lu, "Speeding up processing with approximation circuits," *Computer*, 2004:37(3):67–73.