



# Energy Efficient Multiplier for High Speed DSP Application

Sumant Mukherjee<sup>1</sup>, Dushyant Kumar Soni<sup>2</sup>

<sup>1</sup>Dr. C.V.Raman University, Department of Engineering, Kota, Bilaspur,(C.G.)

<sup>2</sup>Dr. C.V.Raman University, Department of Engineering, Kota, Bilaspur,(C.G.)

<sup>1</sup>[sjsumant@gmail.com](mailto:sjsumant@gmail.com), <sup>2</sup>[dushyantsoni.soni@gmail.com](mailto:dushyantsoni.soni@gmail.com)

*Abstract: A multiplier is one of the important hardware blocks in most digital and high performance systems such as microprocessors. With improving in technology, many researchers have tried and are trying to design multipliers which offer high speed, low power consumption and less area. However area and speed are two most important constraints. In this paper we propose Energy Efficient approximate multiplier using AHA and AFA. Due to this area reduces upto 35% and carry propagation delay also reduces.*

*Keywords :- Approximate half Adder(AHA), Approximate full adder(AFA), Approximate Multiplier, MAC unit, SPAA(Speed, Power, Area, Accuracy), LUT(Look up Table)*

## I. Introduction

Multipliers are one the most important component of many systems. In high speed digital signal processing (DSP) and image processing multiplier play a vital role. Multipliers and adders are the key element of the arithmetic units as they lie in the critical path. With the recent advances in technology, many researchers have tried to implement increasingly efficient multiplier. They aim at offering low power consumption, high speed and reduced delay. Digital signal Processing (DSP) is finding its way into more applications [19], and its popularity has materialized into a number of commercial processors [18]. Digital signal processors have different architectures and features than general purpose processors, and the performance gains of these features largely determine the performance of the whole processor. Basic operation found in MAC is the

binary addition. Besides of the simple addition of two numbers, addition forms the basis for many processing operations, from counting to multiplication to filtering. But also simpler operations like incrimination and magnitude comparison based on binary addition. Therefore, binary addition is the most important arithmetic

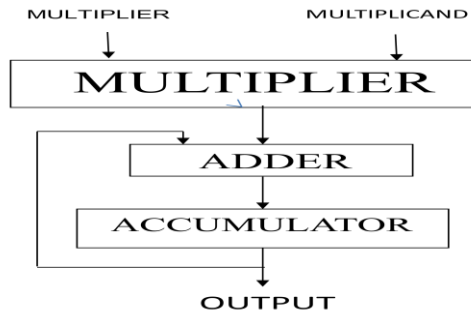


Figure 1 MAC unit

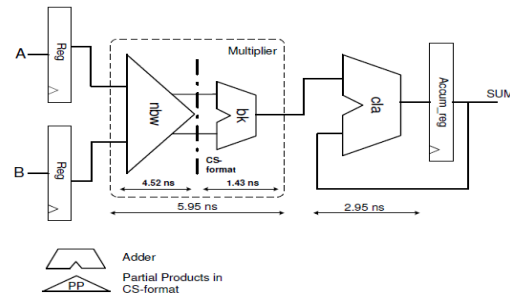


Figure 2 The Benchmark MAC unit

operation. It comparison based on binary addition. It is also a very critical one if implemented in hardware because it involves an expensive carry-propagation step, the evaluation time of which is dependent on the operand word length.

## II. Literature Review

In this part we begin with the basic building blocks used for addition and multiplication, then go through different algorithms.

### A. Basic Adder blocks

#### 1. Half Adder

The Half Adder (HA) is a combinational circuit with two binary input and two binary outputs such as sum and carryout. The equation (1) and (2) are the Boolean equations for sum and carryout, respectively.

$$\text{sum} = a \text{ xor } b \tag{1}$$

$$\text{carryout} = a \text{ and } b \tag{2}$$

#### 2. Full Adder

The Full Adder (FA) is a combinational circuit that adds two bits and a carry and outputs a sum bit and a carry bit. Equation (3) , (4) and (5) are the Boolean equations for the full adder sum and full adder carryout, respectively. In both those equations cin means carryin.

$$\text{sum} = a \text{ xor } b \text{ xor } \text{cin} \tag{3}$$

$$\text{carryout} = a \text{ and } b + b \text{ and } \text{cin} + a \text{ and } \text{cin} \tag{4}$$

$$\text{cin} = a \text{ and } b + (a + b)\text{and } \text{cin} \tag{5}$$

From the above equations we see that sum and carryout is depends on carry in.

*B. Basic Multiplication Schemes*

Multiplication hardware often consumes much time and area compared to other arithmetic operations. Digital signal processors use a multiplier/MAC unit as a basic building block [5] and the algorithms they run are often multiply-intensive. A multiplication operation can be broken down into two steps:

- 1) Generate the partial products.
- 2) Accumulate (add) the partial products.

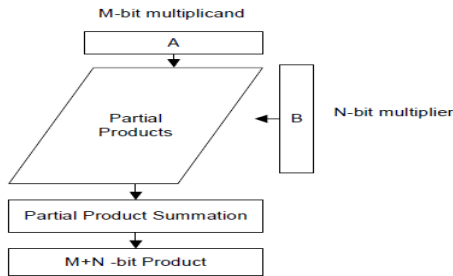


Figure 3 Generic Multiplier Block Diagram

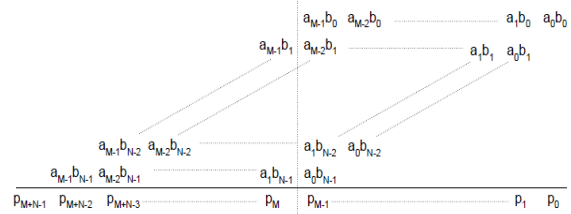


Figure 4 Partial product array

*1. Array Multiplier*

Each multiplicand is multiplied by a bit in the multiplier, generating N partial products. Each of these partial products is either the multiplicand shifted by some amount, or 0. The generation of partial products consists of simple AND'ing of the multiplier and the multiplicand.

*2. Tree Multiplier*

The tree multiplier reduces the time for the accumulation of partial products by adding all of them in parallel, whereas the array multiplier adds each partial product in series. The tree multiplier commonly uses CSAs to accumulate the partial products.

*2.1 Wallace Tree*

The reduction of partial products using full adders as carry-save adders (also called 3:2 counters) became generally known as the "Wallace Tree" [14]. Figure shows an example of tree reduction for an 8\*8-bit partial product tree.

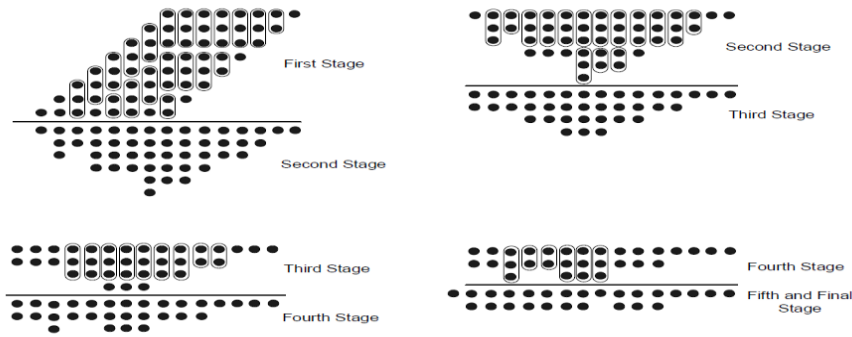


Figure 5 Wallace Tree for an 8 \* 8-bit partial product tree

### 3. Baugh–Wooley Algorithm

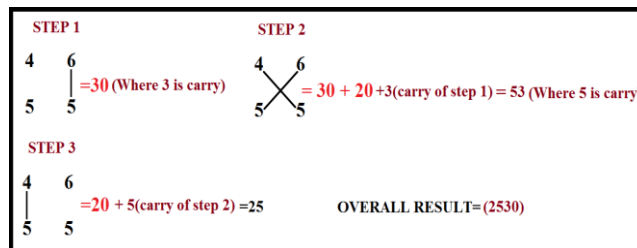
The baugh-Wooley Multiplication Algorithm is an efficient way to handle the sign bits. This technique has been developed in order to design regular multipliers suited for 2’s complement numbers. The Baugh-Wooley (BW) algorithm is a relatively straightforward way of doing signed multiplications.

### 4. Vedic Multiplication

Vedic mathematics is part of four Vedas (books of wisdom) of Indian culture. The Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system.

#### 4.1 Urdhva– Triyagbhyam (Vertically & Crosswise)

Urdhva tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means “Vertically and Crosswise”.



### III. Problem Identification

From the adder architecture we understand that the carry propagation is the main issue. In the ripple carry adder the carry out of each stage is connected to the carryin of the next stage. The sum and carryout bits of any stage cannot be produced, until some time after the carryin of that stage occurs. The time for this implementation of the adder is expressed in below Equation, where tRCAcarry is the delay for the carryout of a FA and tRCAsum is the delay for the sum of a FA.

$$\text{Propagation Delay (tRCAProp)} = (N - 1) \cdot \text{tRCAcarry} + \text{tRCAsum}$$

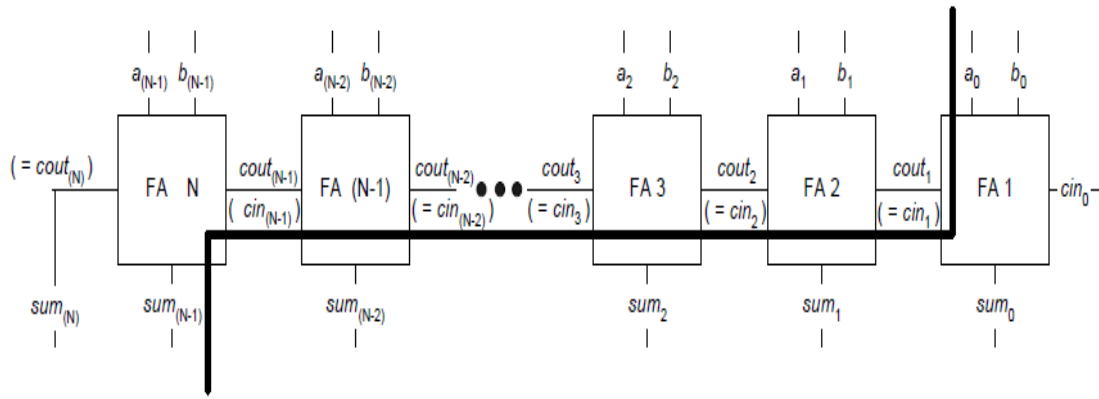


Figure 6 Critical Path for an N-bit Ripple Carry Adder

In the multiplier, after partial product we again have to add that partial product by using adders. So if we want to speed up MAC unit we have to minimize carry propagation delay.

#### IV. Proposed Architecture of 8 Bit approximate adder

Here we proposed a new architecture of half adder and full adder as we know for 8 bit addition there is total 7 full adder and 1 half adder is require. But in proposed approach we propose a new novel 8 bit architecture where we can put some error on lsb bit of adder. Here in approximate half and full adder there is no any carry generation unit. So on first LSB bit we are using proposed approximate half adder and on second LSB bit we use one approximate full adder for next third bit there is no any carry generate so there is no need to use one full adder so at the place of full adder we are using one half adder and after that we use 5 full adder. So as we can see with small error generation we can reduce the hardware requirement and we can make justice with SPAA matrices.



Figure 7 Proposed Approximate Half Adder

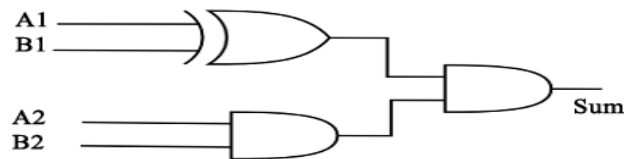


Figure 8 Proposed Approximate Full Adder

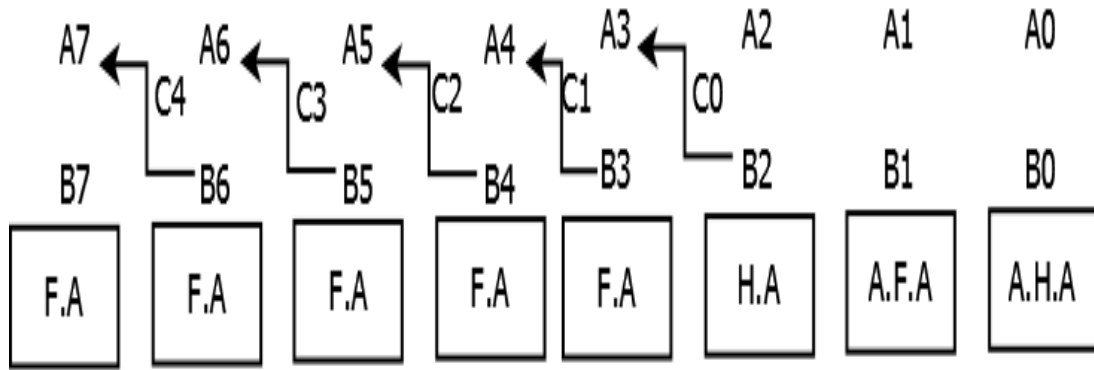


Figure 9 Proposed Architecture of 8 Bit Approximate Adder

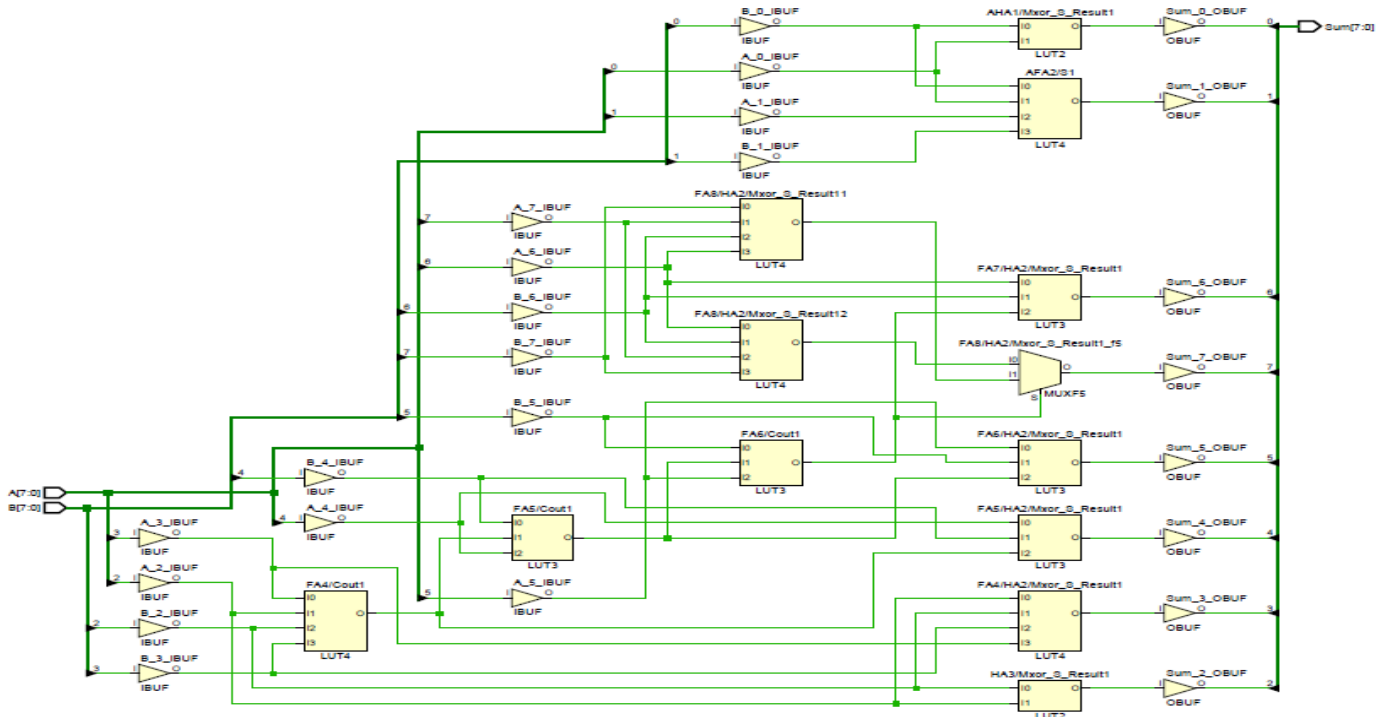


Figure 10 Implementation Detail of proposed 8 Bit approximate adder

### V. Proposed Architecture of 8 Bit approximate multiplier

In this section we present over proposed 8 bit multiplier. This multiplier is a combination of accurate and approximate 4 bit multiplier. For generation of this multiplier we are using the divide and concrete approach in which we design one 4 bit approximate multiplier where we are using normal multiplication approach but at the time of final addition we are using my own approximate half and full adder logic. Due to this approach there is reduction in hardware stricture of 4 bit multiplier.

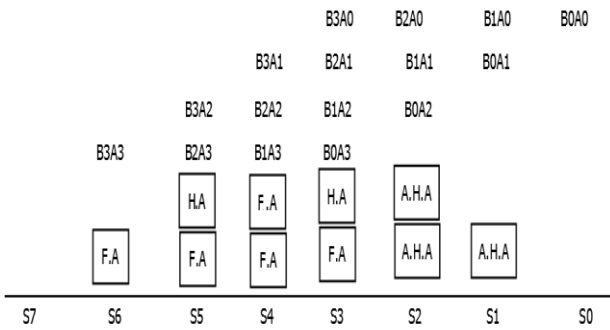


Figure 11 Proposed Approximate 4 Bit Multiplier

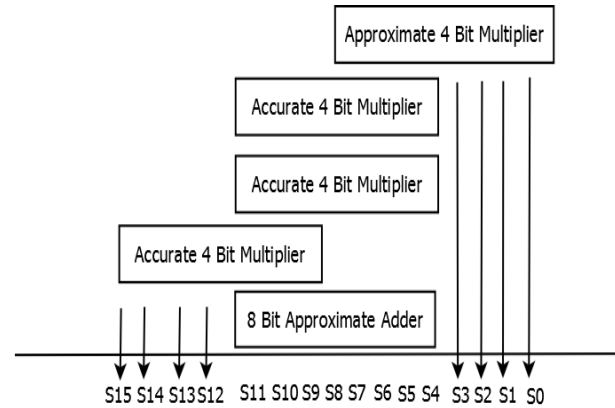


Figure 12 Proposed Approximate 8 Bit Multiplier

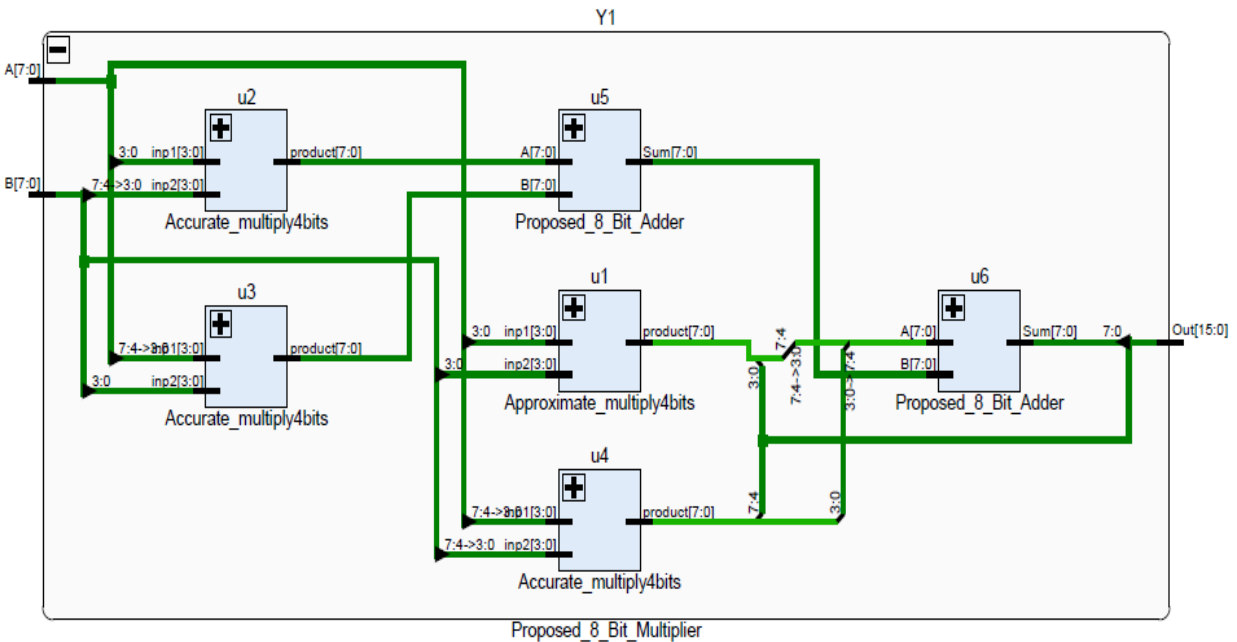


Figure 13 Implementation Detail of proposed 8 Bit approximate multiplier

### VI. Result & Hardware Analysis

Approximate Multiplier Accuracy Level = 87%

The FPGA comparison analysis of proposed and accurate are shown below, here hardware analysis is done on Vertex 6 FPGA which is 45nm based technology.

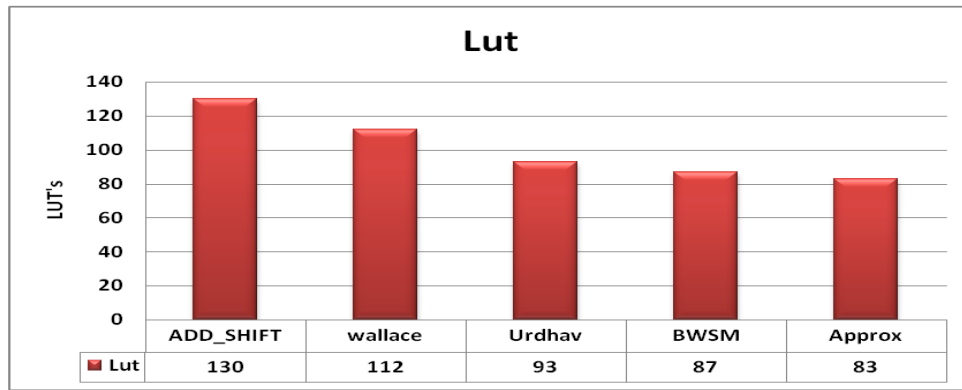


Figure 14 Comparison analysis of Luts of Accurate and Proposed Multiplier

From the above graphs we can see that 35% reduction in logic block is achieved

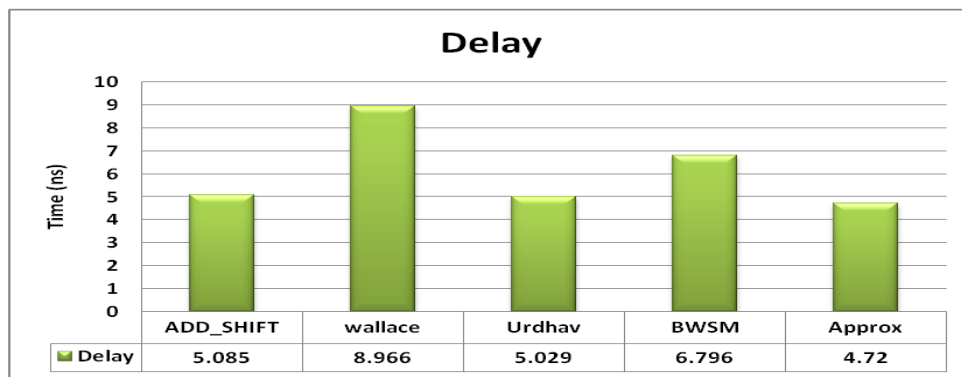


Figure 15 Comparison analysis of delay of Accurate and Proposed Multiplier

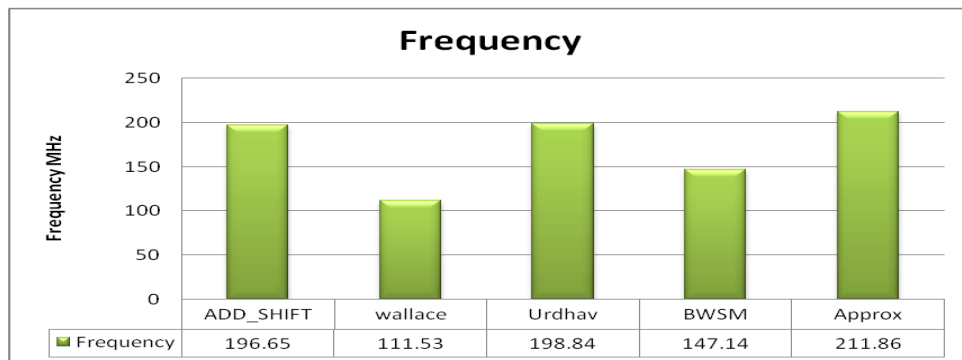


Figure 16 Comparison analysis of frequency of an accurate and Proposed Multiplier



## VII. Conclusion

In this paper, the implementation and analysis of a Approximate multiplier architecture is proposed. The comparison result also shows that a significant reduction in the area is achieved. The overall area, delay and frequency analysis are presented and compared. From the results we can depict that approximately up to 25% to 35% of reduction at all levels are achieved. So due to this we use approximation, which will minimize delay. The results obtained prove that the proposed architecture is more efficient than the conventional one in terms of area and delay. This design is particularly useful in computation-intensive applications which are robust to small errors in computation. The potential applications of this approximate Multiplier fall mainly in areas where there is no strict requirement on accuracy or where super-low power consumption and high speed performance are more important than the accuracy. One example of such applications is in the DSP application for portable devices such as cell phones and laptops.

## References

1. Leem, L.; Hyungmin Cho; Bau, J.; Jacobson, Q.A.; Mitra, S, "ERSA: Error Resilient System Architecture for probabilistic applications," *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010* , vol., no., pp.1560,1565, 8-12 March 2010
2. Ning Zhu; Wang-Ling Goh; Kiat-Seng Yeo, "An enhanced low-power high-speed Adder For Error-Tolerant application," *Integrated Circuits, ISIC '09. Proceedings of the 2009 12th International Symposium on* , vol., no., pp.69,72, 14-16 Dec. 2009
3. Kahng, A.B.; Seokhyeong Kang, "Accuracy-configurable adder for approximate arithmetic designs," *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE* , vol., no., pp.820,825, 3-7 June 2012
4. Rudagi, J M; Ambli, Vishwanath; Munavalli, Vishwanath; Patil, Ravindra; Sajjan, Vinaykumar, "Design and implementation of efficient multiplier using Vedic Mathematics," *Advances in Recent Technologies in Communication and Computing (ARTCom 2011), 3rd International Conference on* , vol., no., pp.162,166, 14-15 Nov. 2011
5. Abdelgawad, A.; Bayoumi, M., "High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications," *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on* , vol., no., pp.3199,3202, 27-30 May 2007
6. Mottaghi-Dastjerdi, M.; Afzali-Kusha, A.; Pedram, M., "BZ-FAD: A Low-Power Low-Area Multiplier Based on Shift-and-Add Architecture," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.17, no.2, pp.302,306, Feb. 2009
7. Tung Thanh Hoang; Sjalander, M.; Larsson-Edefors, P., "A High-Speed, Energy-Efficient Two-Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.57, no.12, pp.3073,3081, Dec. 2010
8. Lomte, R.K.; Bhaskar, P.C., "High Speed Convolution and Deconvolution Using Urdhva Triyagbhyam," *VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on* , vol., no., pp.323,324, 4-6 July 2011
9. Abdelgawad, A., "Low power multiply accumulate unit (MAC) for future Wireless Sensor Networks," *Sensors Applications Symposium (SAS), 2013 IEEE* , vol., no., pp.129,132, 19-21 Feb. 2013
10. Saokar, S.S.; Banakar, R. M.; Siddamal, S., "High speed signed multiplier for Digital Signal Processing applications," *Signal Processing, Computing and Control (ISPCC), 2012 IEEE International Conference on* , vol., no., pp.1,6, 15-17 March 2012
11. Gandhi, D.R.; Shah, N.N., "Comparative analysis for hardware circuit architecture of Wallace tree multiplier," *Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on* , vol., no., pp.1,6, 1-2 March 2013
12. Prakash, A.R.; Kirubaveni, S., "Performance evaluation of FFT processor using conventional and Vedic algorithm," *Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on* , vol., no., pp.89,94, 25-26 March 2013
13. Itawadiya, A.K.; Mahle, R.; Patel, V.; Kumar, D., "Design a DSP operations using vedic mathematics," *Communications and Signal Processing (ICCSP), 2013 International Conference on* , vol., no., pp.897,902, 3-5 April 2013

14. Khan, S.; Kakde, S.; Suryawanshi, Y., "VLSI implementation of reduced complexity wallace multiplier using energy efficient CMOS full adder," *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on* , vol., no., pp.1,4, 26-28 Dec. 2013
15. Yu-Ting Pai; Yu-Kung Chen, "The fastest carry lookahead adder," *Field-Programmable Technology, 2004. Proceedings. 2004 IEEE International Conference on* , vol., no., pp.434,436, 28-30 Jan. 2004
16. ZhouWang; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P., "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on* , vol.13, no.4, pp.600,612, April 2004 doi: 10.1109/TIP.2003.819861
17. Xue, W.; Zhang, L.; Mou, X.; Bovik, A., "Gradient Magnitude Similarity Deviation:A Highly Efficient Perceptual Image Quality Index," *Image Processing, IEEE Transactions on* , vol.PP, no.99, pp.1,1
18. Itawadiya, A.K.; Mahle, R.; Patel, V.; Kumar, D., "Design a DSP operations using vedic mathematics," *Communications and Signal Processing (ICCSP), 2013 International Conference on* , vol., no., pp.897,902, 3-5 April 2013
19. Saokar, S.S.; Banakar, R. M.; Siddamal, S., "High speed signed multiplier for Digital Signal Processing applications," *Signal Processing, Computing and Control (ISPCC), 2012 IEEE International Conference on* , vol., no., pp.1,6, 15-17 March 2012