Energy Efficient Approximate MAC Unit for High Speed DSP Application

Sumant Mukherjee[1], Saurabh Mitra[2]

[1, 2]Dr. C.V.Raman University, Department of Engineering

Abstract: ***In this paper a new energy efficient MAC unit will be introduced, which will reduce the hardware complexity and make justice with SPAA metrics. Another important issue in digital circuits besides speed, area, power consumption is accuracy. In this paper, our main focus is on performance and accuracy, but we do provide some numbers for the arithmetic units relating to energy and power. This is to provide an estimate of the amount of energy and power consumed by the units we choose to implement.***

Keywords: Approximate half Adder(AHA), Approximate full adder(AFA), Approximate Multiplier, MAC unit, SPAA(Speed, Power, Area, Accuracy)

1. Introduction

The addition and multiplication of two binary numbers is the fundamental and most often used arithmetic operation in microprocessors, digital signal processors, and data-processing application-specific integrated circuits. Therefore, binary adders and multipliers are crucial building blocks in VLSI circuits. The core of every microprocessor, DSP, and data-processing ASIC is its data path. Statistics showed that more than 70% of the instructions perform additions and multiplications in the data path of RISC machines [N01]. At the heart of data-path and addressing units in turn are arithmetic units, such as comparators, adders, and multipliers. Digital multipliers are the most commonly used components in any digital circuit design. Multiplication based operations such as Multiply and Accumulate and inner product are among some of the frequently used Computation-Intensive Arithmetic Functions, currently implemented in many DSP applications such as convolution, fast Fourier transform, filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip. The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Digital signal

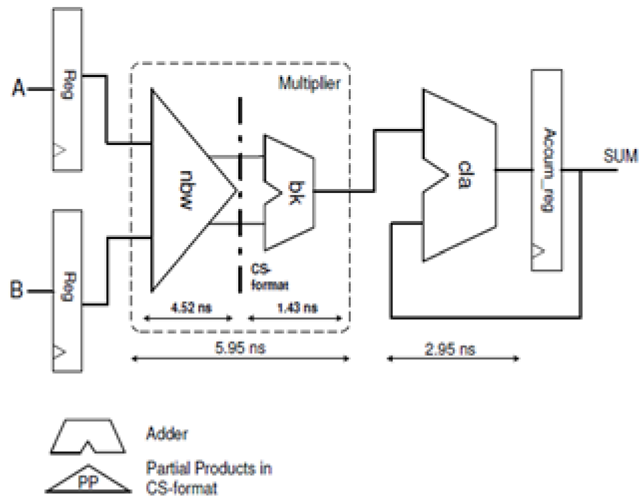


Figure 1: The benchmark MAC unit

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.

2. Literature Review

2.1 Adder Algorithms and Implementations

In nearly all digital IC designs today, the addition operation is one of the most essential and frequent operations. Often, an adder or multiple adders will be in the critical path of the design, hence the performance of a design will be often be limited by the performance of its adders. When looking at other attributes of a chip, such as area or power, the designer will find that the hardware for addition will be a large contributor to these areas.

2.2.Basic Adder blocks

2.2.1 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.

sum = a xor b (1)

carryout = a and b (2)

2.2.2 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.

sum = a xor b xor cin (3)

carryout = a and b + b and cin + a and cin (4)

cin = a and b + (a + b)and cin (5)

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

2.2.3 Partial Full Adder

The Partial Full Adder (PFA) is a structure that implements intermediate signals that can be used in the calculation of the carry bit. Such as delete, propagate and generate.

Table 1: Extended Truth Table for a 1-bit adder

Inputs			Outputs					
carryin	*a*	*b*	*carryout*	*sum*	*g*	*d*	*p*	Carry status
0	0	0	0	0	0	1	0	delete
0	0	1	0	1	0	0	1	propagate
0	1	0	0	1	0	0	1	propagate
0	1	1	1	0	1	0	1	generate/propagate
1	0	0	0	1	0	1	0	delete
1	0	1	1	0	0	0	1	propagate
1	1	0	1	0	0	0	1	propagate
1	1	1	1	1	1	0	1	generate/propagate

generate(g) = a and b (6)

delete(d) = $\bar{a}$ and $\bar{b}$ (7)

propagate(p) = a and b (or a xor b) (8)

sum = p xor carryin (9)

carryout = g or p and carryin (10)

2.2.4 Ripple Carry Adder[14]

In the parallel 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. This is due to the propagation delay in the logic circuitry , which lead to a time delay in the addition process. The carry propagation delay for each full adder is the time between the application of the carryin and the occurrence of the carryout. The parallel adder in which the carryout of each full adder is the carryin to the next more significant adder is called a ripple carry adder.

2.2.5 Carry Look Ahead Adder[15]

In the case of the parallel adder , the speed with which an addition can be performed is governed by the time required for the carries to propagate or ripple through all the stages of the adder. The look ahead carry adder speeds up the process by eliminating this ripple carry delay. It examines all the input bits simultaneaously and also generates the carry in bits for all the stages simultaneously.

2.3 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.

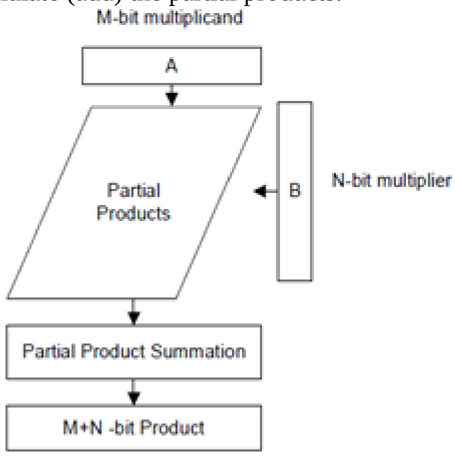


Figure 2: Generic Multiplier Block Diagram

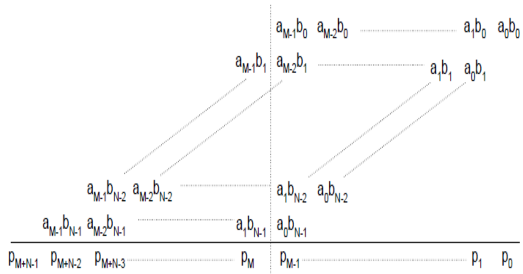


Figure 3: Partial product array for an M *N multiplier

2.3.1 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. This is illustrated in Fig for an M * N multiplies operation. The generation of partial products consists of simple AND'ing of the multiplier and the multiplicand.

2.3.2 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.

2.3.2.1 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.

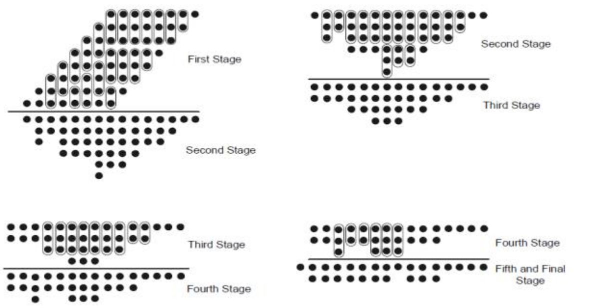


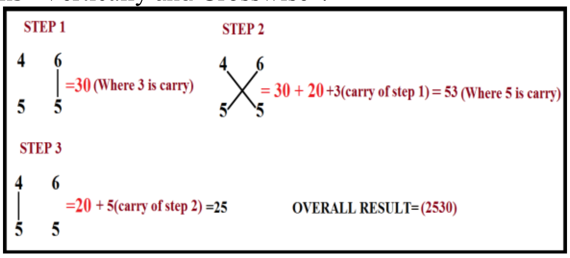
Figure 4: Wallace Tree for an 8 * 8-bit partial product tree

2.3.3 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

2.3.3.1 Urdhva– Triyagbhyam (Vertically & Crosswise)

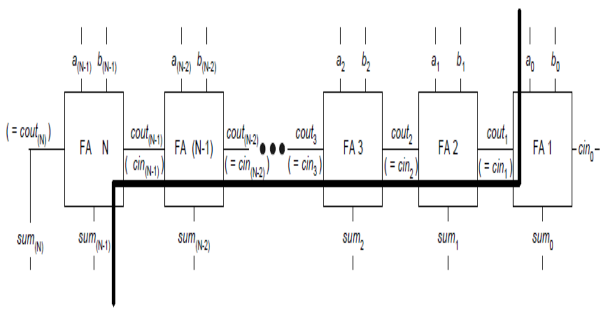
Urdhva tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and Crosswise".



3. Problem Identification

From the adder circuit 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.

Propagation Delay (tRCAprop) = (N - 1) . tRCAcarry + tRCAsum



.

Figure 1 : 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.

4. Proposed Multiply-Accumulate Unit Design and Implementation

The Multiply-Accumulate (MAC) unit performs the Multiply instruction and the MAC instruction, which are essential for all DSP processors. To improve the speed of the multiplication operation is to improve the partial product generation step. This can be done in two ways:

1) Generate the partial products in a faster manner.
2) Reduce the number of partial products that need to be generated.

Here we represent the implementation details of my proposed 8 bit arithmetic unit , 8 bit multiplier unit and 8 bit MAC unit.

4.1 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.

Figure 6: Proposed Approximate Half Adder

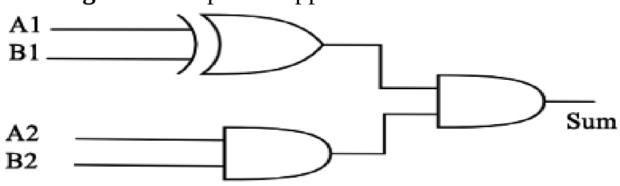


Figure 7: Proposed Approximate Full Adder

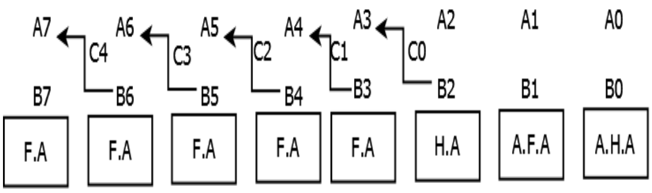


Figure 8: Proposed Architecture of 8 Bit Approximate Adder

4.2 Proposed Architecture of 8 Bit approximate multiplier

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

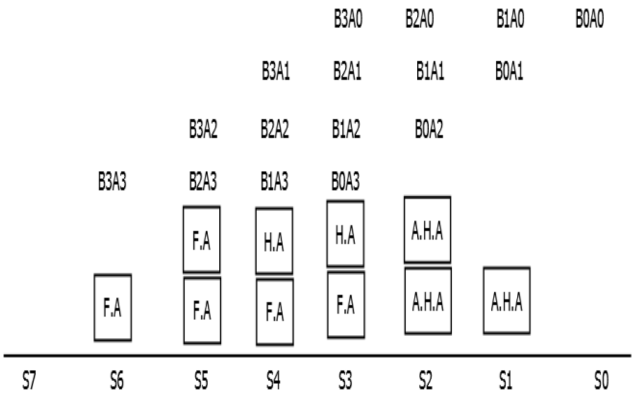


Figure 9: Proposed Approximate 4 Bit Multiplier

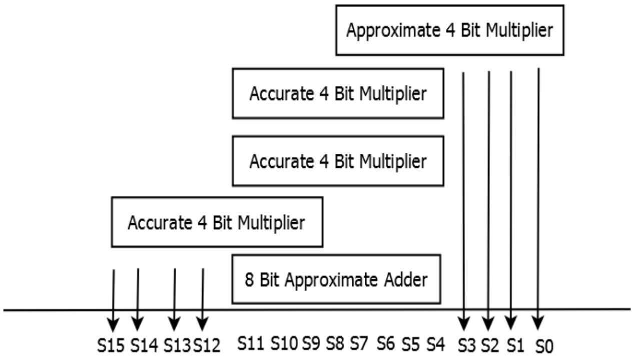


Figure 10: Proposed Approximate 8 Bit Multiplier

4.3 Proposed Architecture of 8 Bit approximate MAC unit

We proposed 8 Bit Multiplier Accumulator unit which is combination of accurate and approximate logic unit. Here we are using 8 bit approximate multiplier unit which is combination of 4 bit multiplier and one 16 bit adder which is combination of one Approximate half adder , three approximate full adder, one accurate half adder and 11 accurate full adder.

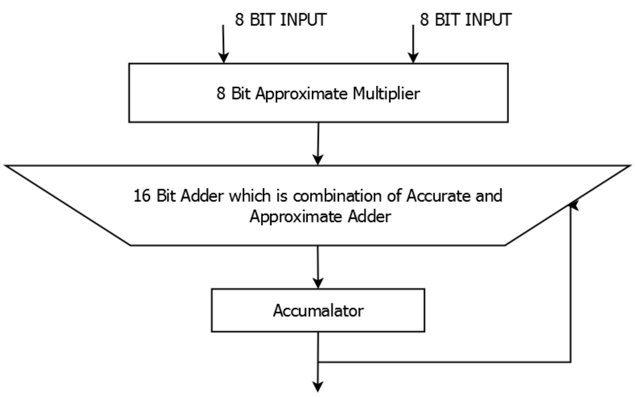


Figure 11: Approximate 8 bit MAC unit

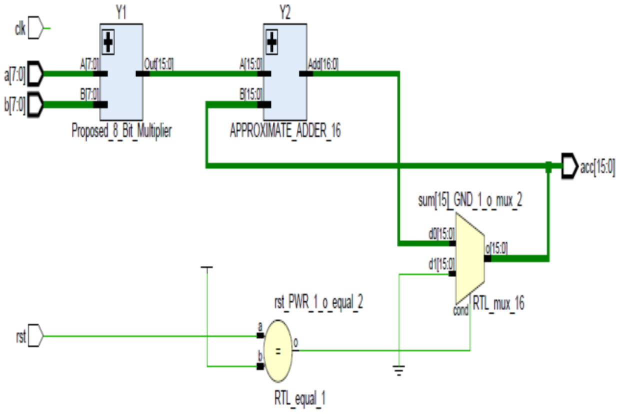


Figure 12: Architecture of Approximate 8 Bit MAC Unit

5. Result & Analysis

Through proposed MAC unit we generate the output image and compare it with accurate sobel edge output image. Here we using some parameter and those parameters are :

- PSNR
- SSIM[16]
- GMSD[17]

Generated output result of all parameter are shown in below:

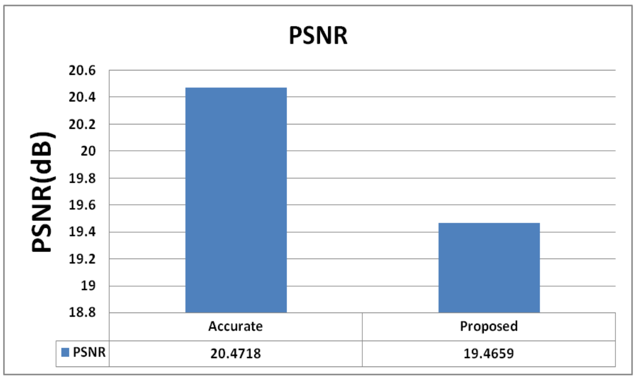


Figure 13: Comparison analysis of PSNR of Accurate and Proposed MAC unit

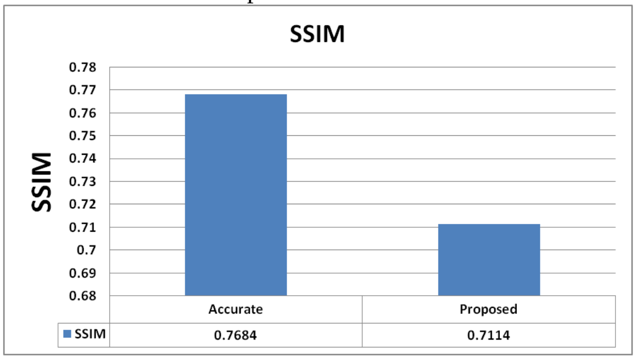


Figure 14: Comparison analysis of SSIM of Accurate and Proposed MAC unit

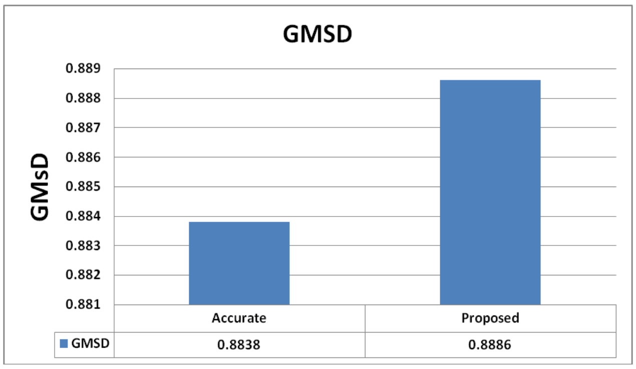


Figure 15: Comparison analysis of GMSD of Accurate and Proposed MAC unit

5.1 Hardware Analysis

Approximate MAC Unit Accuracy Level = 90%
The FPGA comparison analysis of proposed and accurate are shown below, here hardware analysis is done on Vertix 6 FPGA which is 45nm based technology.

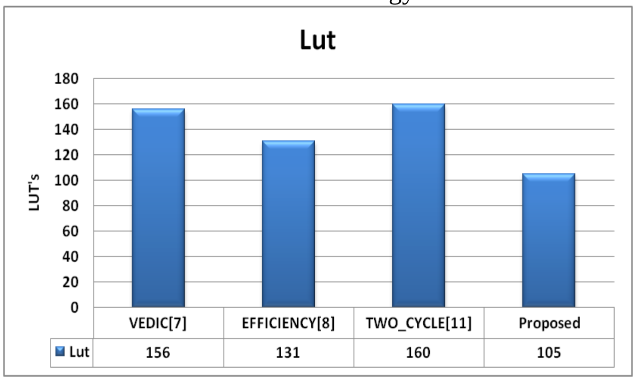


Figure 16: Comparison analysis of Luts of Accurate and Proposed MAC unit

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

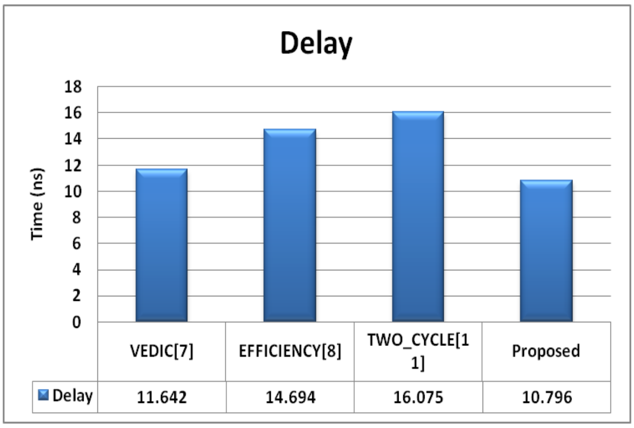


Figure 17: Comparison analysis of delay of Accurate and Proposed MAC unit

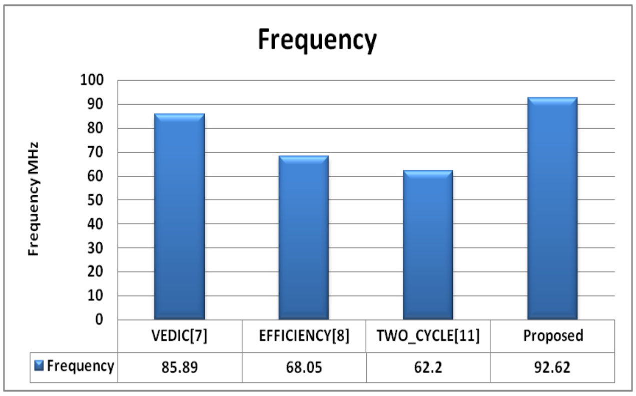


Figure 18: Comparison analysis of frequency of an accurate and Proposed MAC unit

6. Conclusion

This paper present a approximate MAC [4, 5, 7] unit. Using approximates half and full adder we create 8 and 16 bit adder. Which is use in 8 bit multiplier and 8 bit mac unit. For image quality analysis we use one application which is known as sobel edge detection. There is small degradation in image quality which is tolerable by human eye. The overall area and 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 potential applications of this approximate MAC unit 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 Prossing 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-Kumg 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 E_cient 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