International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Implementation of Floating point Vedic multiplier by Urdhva Triyagbhyam using VHDL

Shailesh Mishra
Electronics and Communication Engg,
I.E.T., DRMLAU, Ayodhya, India
shaileshm710@gmail.com

Manisha Yadav
Electronics and Communication Engg,
I.E.T., DRMLAU, Ayodhya, India
manishayadav2010@gmail.com

Nupur Kesarwani
Electronics and Communication Engg,
I.E.T., DRMLAU, Ayodhya, India
contact2nupurkesarwani@gmail.com

Abstract: In this paper we proposed the design of high speed Vedic Multiplier using the techniques of Ancient Indian Vedic Mathematics that have been modified to improve performance. Vedic Multiplication Technique is used to implement IEEE 754 Floating point multiplier. For mantissa multiplication we are using Urdhvatriyakbhyam sutra for the underflow and over flow cases are handled. The multiplier's inputs are provided in IEEE 754, 32 bit format. The Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculations based on 16 Sutras. Our work has proved the efficiency of Urdhva Triyagbhyam– Vedic method for multiplication which strikes a difference in the actual process of multiplication. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros and scaled to higher bit levels using Karatsuba algorithm with the compatibility to different data types. The Urdhva tiryakbhyam Sutra is most efficient Sutra (Algorithm), giving minimum delay for multiplication of all types of numbers, either small or large. We implement this multiplier using VHDL. We implement our work by Xilinx ISE tool i.e. responsible for synthesis also. For simulation we are using Modelsim 10.2a.

Keywords: DSP, MAC, Urdhva tiryakbhyam Sutra, and Vedic Mathematics.

1. Introduction:

Multipliers are key components of many high performance systems such as microprocessors, FIR filters, digital signal processors, etc. Performance of a system is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Since multiplication dominates the execution time of most DSP application so there is need of high speed multiplier. Furthermore, it is generally the most area consuming. Hence, optimizing the area and speed of the multiplier is a major design issue. However, speed and area are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area-speed constraints has been designed with fully serial multipliers at one end of the spectrum and fully parallel Multipliers at the other end. These multipliers have moderate performance in both speed and area. Binary floating point numbers multiplication is one of the basic functions used in digital signal processing (DSP) application. The IEEE 754 standard provides the format for representation of Binary Floating point numbers in computers. The Binary Floating point numbers are represented in Single and Double formats. The Single precision format consists of 32 bits and the Double precision format consists of 64 bits. The formats are composed of 3 fields; Sign, Exponent and Mantissa. A typical central processing unit devotes a considerable amount of processing time in implementing arithmetic operations, particularly multiplication operation. Most high performance DSP systems rely on hardware multiplication to achieve high data throughput. Multiplication is an important fundamental arithmetic operation. Performance constraints can also be addressed by applying alternative technologies. A change at the level of design implementation by the insertion of a new technology can often make viable an existing marginal algorithm or architecture. This project deals with the "Design of high speed floating point multiplier using ancient technique". In this project Vedic Multiplication Technique is used to implement IEEE 754 Floating point multiplier. For calculation of mantissa unit The Vedic sutra is used. A change at the implementation level of design by the insertion of a new technology can often make viable an existing marginal algorithm or architecture. Performance constraints can also be addressed by applying alternative technologies.

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate(MAC) and inner product are among some of the frequently used Computation- Intensive Arithmetic Functions(CIAF) currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform(FFT), filtering and in microprocessors in its arithmetic and logic unit [1]. 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 [2]. 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. Reducing the time delay and power consumption are very essential requirements for many applications [2, 3]. This work presents different multiplier architectures. Multiplier based on Vedic Mathematics is one of the fast and low power multiplier.

In this paper we will work on the design of high speed Vedic Multiplier using the techniques of Ancient Indian Vedic Mathematics that have been modified to improve performance. Vedic Multiplication Technique is used to implement IEEE 754 Floating point multiplier. For mantissa multiplication we are using Urdhvatriyakbhyam sutra for the underflow and over flow cases are handled. The multiplier's inputs are provided in IEEE 754, 32 bit format. The Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculations based on 16 Sutras. Our work has proved the efficiency of Urdhva Triyagbhyam– Vedic method for multiplication which strikes a difference in the actual process of multiplication. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros and scaled to higher bit levels using Karatsuba algorithm with the compatibility to different data types.

2. Related Work:

Sushma S. Mahakalkar et al, they worked on the fundamental and the core of all the Digital Signal Processors (DSPs) are its multipliers and the speed of the DSPs is mainly determined by the speed of its multiplier. IEEE floating point format was a standard format used in all processing elements since Binary floating point numbers multiplication is one of the basic functions used in digital signal processing (DSP) application. In that work VHDL implementation of Floating Point Multiplier using ancient Vedic mathematics is presented. The idea for designing the multiplier unit is adopted from ancient Indian mathematics "Vedas". The Urdhva Tiryakbhyam sutra (method) was selected for implementation since it is applicable to all cases of multiplication. Multiplication of two no's using Urdhva Tiryakbhyam sutra is performed by vertically and crosswise, crosswise means diagonal multiplication and vertically means straight above multiplication and taking their sum. The feature is any multi-bit multiplication can be reduced down to single bit multiplication and addition using this method. On account of these formulas, the carry propagation from LSB to MSB is reduces due to one step generation of partial product [1].

Aritra Mitra et al, proposed a Vedic Multiplication Technique which used to implement Floating point multiplier. The Urdhvatriyakbhyam sutra will be used for the multiplication of Mantissa. The underflow and over flow cases will be handled. The inputs to the multiplier in 32 bit format. The multiplier is designed in VHDL or VERILOG and simulated using Modelsim [3].

Bhagyashree Hardiya et al, worked on multiplication of the floating point numbers described in IEEE 754 single precision valid. Floating point multiplier is done using VHDL .Implementation in VHDL(VHSIC Hardware Description Language) is used because it allow direct implementation on the hardware while in other language they have to convert them into HDL then only can be implemented on the hardware. In floating point multiplication, adding of the two numbers is done with the help of various types of adders but for multiplication some extra shifting is needed. This floating point multiplication handles various conditions like overflow, underflow, normalization, rounding. In this work they use IEEE rounding method for perform the rounding of the resulted number. This work reviews the implementation of an IEEE 754 single precision floating point multiplier developed by many researchers [4].

3. Methodology:

DSP applications essentially require the multiplication of binary floating point numbers. The IEEE 754 standard provides the format for representation of Binary Floating point numbers [1, 2]. The Binary Floating point numbers are represented in Single and Double formats. The Single consist of 32 bits and the Double consist of 64 bits. The formats are composed of 3 fields; Sign, Exponent and Mantissa. The Figure 3.1 shows the structure of Single and Double formats of IEEE 754 standard. In case of Single, the Mantissa is represented in 23 bits and 1 bit is added to the MSB for normalization, Exponent is represented in 8 bits which is biased to 127, actually the Exponent is represented in excess 127 bit format and MSB of Single is reserved for Sign bit. When the sign bit is 1 that means the number is negative and when the sign bit is 0 that means the number is positive. In 64 bits format the Mantissa is represented in 52 bits, the Exponent is represented in 11 bits which is biased to 1023 and the MSB of Double is reserved for sign bit.

The performance of Mantissa calculation Unit dominates overall performance of the Floating Point Multiplier. This unit requires unsigned multiplier for multiplication of 24x24 BITs. The Vedic Multiplication technique is chosen for the implementation of this unit. This technique gives promising result in terms of speed and power [6].The Vedic multiplication system is based on 16 Vedic sutras or aphorisms, which describes natural ways of solving a whole range of mathematical problems. Out of these 16 Vedic Sutras the Urdhva-triyakbhyam sutra is suitable for this purpose. In this method the partial products are generated simultaneously which itself reduces delay and makes this method fast. The method for multiplication of two, 3 BITs

number is shown Figure 3.4. Consider the numbers A and B where A = a2a1a0 and B = b2b1b0. The LSB of A is multiplied with the LSB of B:

s0=a0b0;

Then a0 is multiplied with b1, and b0 is multiplied with a1 and the result are added together as:

c1s1=a1b0+a0b1;

Here c1 is carry and s1 is sum. Next step is to add c1 with the multiplication results of a0 with b2, a1 with b1 and a2 with b0.

c2s2=c1+a2b0+a1b1 + a0b2;

Next step is to add c3 with the multiplication results of a1 with b2 and a2 with b1.

c3s3=c2+a1b2+a2b1;

Similarly the last step

c4s4=c3+a2b2;

Now the final result of multiplication of A and B is c4s4s3s2s1s0.

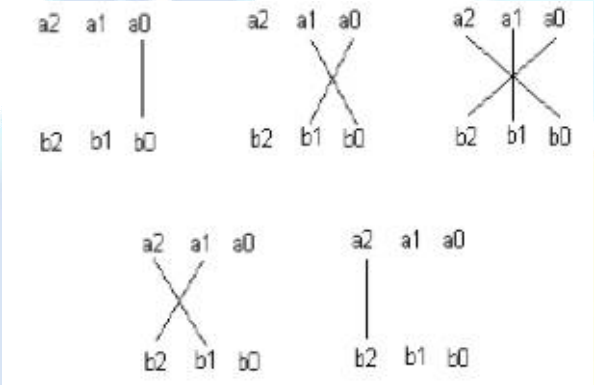

Fig 1: The Vedic Multiplication method

For Multiplier, first the basic blocks, that are the 2x2 bit multipliers have been made and then, using these blocks, 4x4 block has been made by adding the partial products using carry save adders and then using this 4x4 block, 8x8 bit block, 16x16 bit block and then finally 32 x 32 bit Multiplier as shown in figure 3.5 has been made [5].

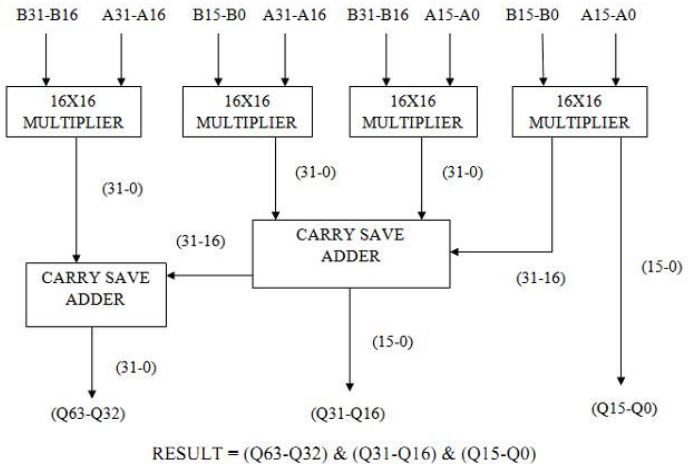

Fig 2: 32X32 Bits proposed Vedic Multiplier

The design starts first with Multiplier design, that is 2x2 bit multiplier as shown in figure 3.6. Here, "Urdhva Tiryakbhyam Sutra" or "Vertically and Crosswise Algorithm"[3] for multiplication has been effectively used to develop digital multiplier architecture. This algorithm is quite different from the traditional method of multiplication, which is to add and shift the partial products.

To scale the multiplier further, Karatsuba – Ofman algorithm can be employed [4]. Karatsuba-Ofman algorithm is considered as one of the fastest ways to multiply long integers. It is based on the divide and conquer strategy. A multiplication of 2n digit integer is reduced to two n digit multiplications, one (n+1) digit multiplication, two n digit subtractions, two left shift operations, two n digit additions and two 2n digit additions.

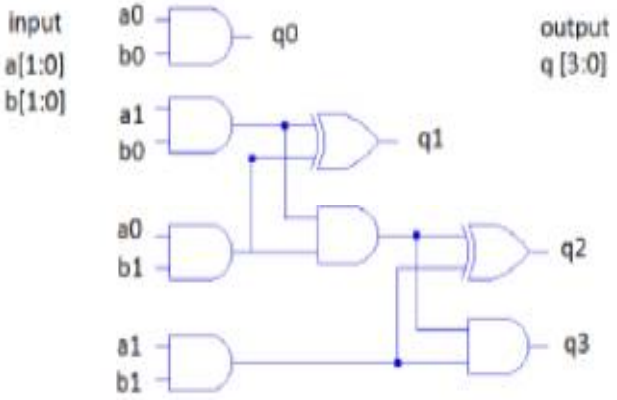

Fig 3: Hardware Realization of 2x2 block

The proposed multiplications were implemented using two different coding techniques viz., conventional shift & add and Vedic technique for 4, 8, 16, and 32 bit multipliers. It is evident that there is a considerable increase in speed of the Vedic architecture.

The number of LUTs and slices required for the Vedic Multiplier is less and due to which the power consumption is reduced [6]. Also the repetitive and regular structure of the multiplier makes it easier to design. And the time required for computing multiplication is less than the other multiplication techniques.

An Overflow or Underflow case occurs when the result Exponent is higher than the 8 BIT or lower than 8 BIT respectively. Overflow may occur during the addition of two Exponents which can be compensated at the time of subtracting the bias from the exponent result. When overflow occurs the overflow flag goes up. The under flow can occur after the subtraction of bias from the exponent, it is the case when the number goes below 0 and this situation can be handled by adding 1 at the time of normalization. When the underflow case occur the under flow flag goes high.

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

4. Result and Discussion:

We have taken two inputs 'A' and 'B' as a multiplier and multiplicand these are floating point signed value we are perform multiplier using Vedic Algorithm between these inputs and will be stored in other output port which we have taken as 'Z' all operations are performing on positive edge of clock.

For case I we take value of 'A' is 134.0625 and value of 'B' is -2.25. Here 'A' is unsigned floating pint number and 'B' is Signed Floating Point Number. Now we have to convert value of 'A' to binary format after normalize we get 1.00001100001x2^7 then we have to convert it into IEE-32 floating point format then we get 0 10000110 00001100001000000000000 then convert it into hexadecimal format we get 0x43061000. Now we have to convert value of 'B' to binary format after normalize we get -1.001x2^1 then we have to convert it into IEE-32 floating point format then we get 1 10000000 00100000000000000000000 then convert it into hexadecimal format we get 0xC0100000. After multiplication using Vedic Multiplier we get 0xC396D200 the value of this hexadecimal no. is -301.640625 fig 4 shows the simulation result of this data.

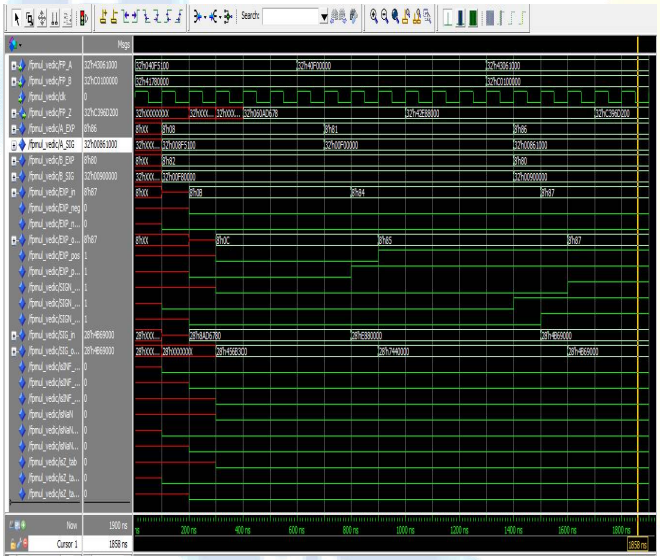

Fig 4. Simulation Result of Case I

For case II we take value of 'A' is -14.5 and value of 'B' is -0.375. Here 'A' is signed floating pint number and 'B' is also a signed Floating Point Number. Now we have to convert value of 'A' to binary format after normalize we get -1.1101x2^3 then we have to convert it into IEE-32 floating point format then we get 1 10000010 11010000000000000000000 then convert it into hexadecimal format we get 0xC1680000. Now we have to convert value of 'B' to binary format after normalize we get -1.1x2^-2 then we have to convert it into IEE-32 floating point format then we get 1 01111101 10000000000000000000000 then convert it into hexadecimal format we get 0xBEC00000. After multiplication using Vedic Multiplier we get 0x40AE0000 the value of this hexadecimal no. is 5.4375 fig 5 shows the simulation result of this data.

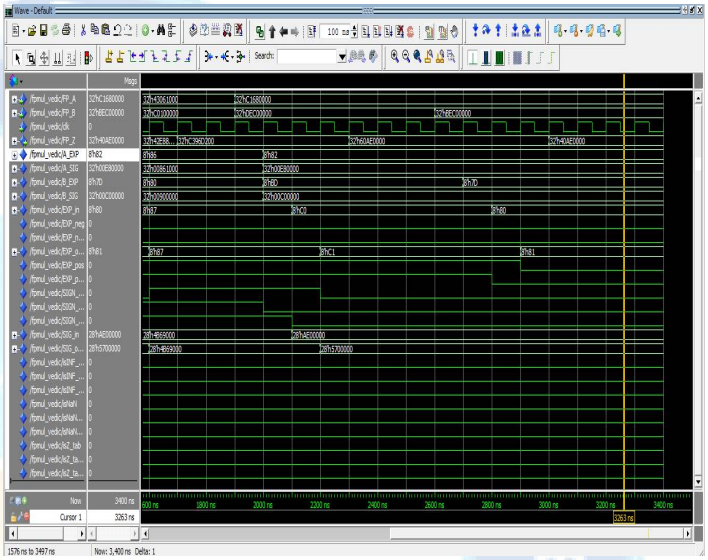

Fig. 5. Simulation Result of Case II

Fig 6 shows the RTL of our code, fig 7 shows the internal RTL.

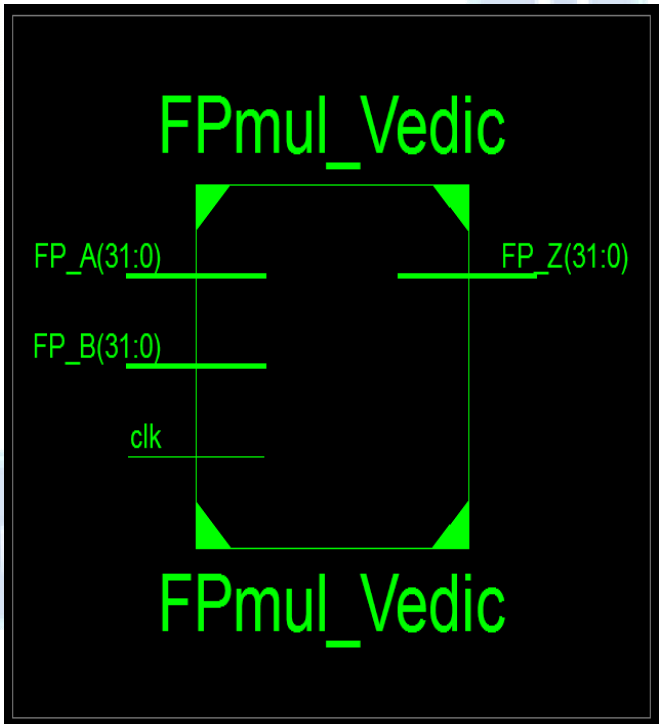

Fig 6. Main RTL

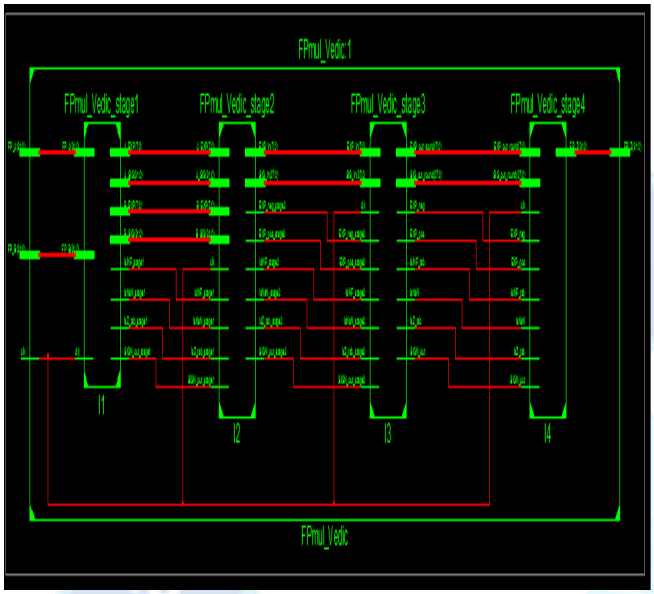

Fig 7. Internal RTL

5. Conclusion:

The Floating Point numbers are the basic necessity in the current scenario of digital design based systems. Hence we implemented, the design of Floating point number in IEEE32 bit format, on Spartan 3E- XC3S250-5-CP132. The design is based on Vedic method of multiplication. The worst case propagation delay in the Optimized Vedic floating point multiplier case is 4.788 ns. It is therefore seen that the Vedic floating point number multipliers are much faster than the conventional multipliers. This gives us method for hierarchical floating point multiplier design. So the design complexity gets reduced for inputs of large no of bits and modularity gets increased. Urdhva tiryakbhyam sutra algorithm is been used which can reduce the delay and hardware requirements for multiplication of Floating point numbers. FPGA based simulation and Synthesis of this floating point multiplier shows that hardware realization of the Vedic mathematics algorithms is easily possible. The high speed multiplier algorithm exhibits improved efficiency in terms of speed.

References:

[1] Sushma S. Mahakalkar et al, "Review on floating point multiplier using ancient techniques" IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331 PP 01-04.

[2] Tariquzzaman et al, " FPGA implementation of 64 bit RISC processor with Vedic multiplier using VHDL" IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331 PP 12-16.

[3] Aritra Mitra et al , "Design of Floating Point Multiplier based on Vedic Multiplication Technique" School of Electronics Engineering Electronics And Telecommuncation Kalinga Institute of Industrial Technology BHUBANESWAR-751024.

[4] Bhagyashree Hardiya et al "Implementation Of Floating Point Multiplier Using VHDL" Technology and Engineering (BEST: IJMITE) Vol. 1, Issue 3, Dec 2013, 199-204 © BEST Journals.

[5] S Venkateswara Reddy et al, "Design And Implementation Of 32 Bit Multipier Using Vedic Mathamatics" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization)Vol. 2, Issue 8, August 2013.

[6] Remadevi R, "Design and Simulation of Floating Point Multiplier Based on VHDL" Remadevi R / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 2, March - April 2013, pp.283-286.

[7] Dinesh Kumar et al, "Simulation And Synthesis Of 32-Bit Multiplier Using Configurable Devices" International Journal of Advances in Engineering & Technology, Jan. 2013. ©IJAET ISSN: 2231-1963.

[8] Poornima M et al, "Implementation of Multiplier using Vedic Algorithm" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013.

[9] MD. Belal Rashid et al, "VLSI Design and Implementation of Binary Number Multiplier based on Urdhva Tiryagbhyam Sutra with reduced Delay and Area"

[10] Assistant Professor et al, "Implementation Of Fixed And Floating Point Division Using Dhvajanka Sutra" International Journal of VLSI and Embedded Systems-IJVES Vol 04, Issue 02; March - April 2013.

[11] Anurag, R. Sharma, " Load Forecasting by using ANFIS", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020.

[12] R. Sharma, Anurag, " Load Forecasting using ANFIS A Review", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020.

[13] R. Sharma, Anurag, " Detect Skin Defects by Modern Image Segmentation Approach, Volume 20, Issue 1, 2020.

[14] Anurag, R. Sharma, " Modern Trends on Image Segmentation for Data Analysis- A Review", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020.

[15] Korra Tulasi Bai et al, "A New Novel Low Power Floating Point Multiplier Implementation Using Vedic Multiplication Techniques" Korra Tulasi Bai, J. E. N. Abhilash / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 4, Jul-Aug 2013, pp.1801-1804.

[16] Mr. Dharmendra Madke et al, "Review Paper on High Speed Karatsuba Multiplier and Vedic Mathematics Techniques" International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 12, December 2013.

[17] Krishnaveni D et al, "VLSI Implementation Of Vedic Multip-Lier With Reduced Delay" International Journal of Advanced Technology & Engineering Research (IJATER) National Conference on Emerging Trends in Technology (NCET-Tech).