ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

Pipelined Floating Point Multiplier Based On Vedic Multiplication Technique

Irine Padma B.T[1], Suchitra. K[2]

M Tech Student, Department of Applied Electronics and Instrumentation, Lourdes Matha College of Science and Technology, Kerala, India.

Asst. Prof, Department of Electronics and Communication, Lourdes Matha College of Science and Technology, Kerala, India.

ABSTRACT: To represent very large or small values, large range is required as the integer representation is no longer appropriate. These values can be represented using the IEEE 754 standard based floating point representation. Multiplying floating point numbers is a critical requirement for DSP applications involving large dynamic range. The paper describes the implementation and design of IEEE 754 Pipelined Floating Point Multiplier based on Vedic Multiplication Technique. The inputs to the multiplier are provided in IEEE 754, 32 bit format. The Urdhva Triyakbhyam sutra is used for the multiplication of mantissa. The underflow and overflow cases are handled.

KEYWORDS: Floating Point Numbers, IEEE 754, Vedic Multiplication, Urdhva Triyakbhyam sutra, Underflow, Overflow, Pipelining.

I. INTRODUCTION

Digital multipliers are the core components of all the digital signal processors (DSPs) and the speed of the DSP is largely determined by the speed of its multipliers. As multiplication dominates the execution time of the most DSP algorithms, there is a need of high speed multiplier. DSP applications essentially require the multiplication of binary floating point numbers. Floating point numbers are one possible way of representing real numbers in binary format. Floating-point numbers are widely adopted in many applications due to their dynamic representation capabilities. These applications perform vast amount of image transformation operations which require many multiplication and division operation. Floating-point representation is able to retain its resolution and accuracy compared to fixed-point representations. To represent very large or small values, large range is required as the integer representation is no longer appropriate. These values can be represented using the IEEE-754 standard based floating point numbers. The IEEE-754 standard provides representation of Binary Floating point numbers in Single and Double formats. The Single consist of 32 bits and the Double consist of 64 bits. FPGA usage for the implementation of Floating Point Number implementations rather than microprocessor based structures will be the best choice due to parallel processing capability, re-programmability, and higher speed.

This paper describes the implementation of pipelining in the design of floating-point multiplier using VHDL and its synthesis for a Xilinx Virtex-II FPGA using Xilinx's Integrated Software Environment (ISE) 12.1. Pipelining is one of the popular methods to realize high performance computing platform. Pipelining is a technique where multiple instruction executions are overlapped. In the top-down design approach, four arithmetic modules: addition, subtraction, multiplication, and division: are combined to form the floating-point ALU. The pipeline modules are independent of each other.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

A. Floating Point Numbers

Fixed point numbers suffer from limited range and accuracy. For a given word length both fixed point and floating point representations give equal distinct numbers. The difference is that in fixed point representation the spacing between the numbers is equal, so smaller numbers when truncated or rounded give a much larger error than the larger numbers. However floating point representation gives different spacing between numbers. We get denser distances between numbers when the number is small and sparser distance for larger numbers. So the absolute representation error increases with larger numbers.

A floating point number consists of two parts plus a sign. The Mantissa is the part of a floating point number that represents the magnitude of the number. The Exponent represents the number of places that the decimal point (or binary point) is to be moved. Let us consider a decimal number which, in integer form, is 241,506,800. When the integer is expressed as a floating point number, it is normalized by moving the decimal point to the left of all the digits so that the mantissa is a fractional number and the exponent is the power of ten. Therefore, the floating point representation is,

$0.2415068 \times 10\ 10^9$

For binary floating point numbers, the format is defined by ANSI/IEEE standard 754-1985.

B. IEEE 754 Standard

Multiplication of Binary floating point numbers 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. The Binary Floating point numbers are represented in three forms: Single-precision, Double-precision and Extended-precision formats. Single-precision floating point numbers have 32 bits, double-precision numbers have 64 bits, and extended-precision numbers have 80 bits.

The formats are composed of 3 fields; Sign, Exponent and Mantissa. The standard format for the representation of floating point number is

$$(-1)^S\ 2^E\ (b_0.b_1b_2\ldots\ldots\ldots\ldots\ldots\ldots..b_{p-1}).$$

framework makes it possible to offer different security levels for different functionalities of different applications.

C. Single Precision Binary Floating Point Numbers

In the standard format for a single-precision binary number, the sign bit (S) is the left-most bit, the exponent (E) includes the next eight bits, and the mantissa or fractional part (F) includes the remaining 23 bits. The IEEE 754 format for Single Precision Floating Point Number is shown in Fig.1.

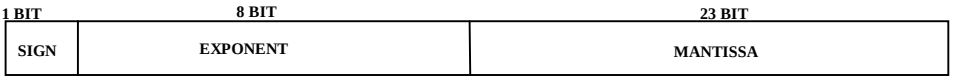


Fig.1: IEEE 754 format for Single-Precision Floating Point Number.

In the mantissa or fractional part, the binary point is understood to be the left of the 23 bits. Effectively, there are 24 bits in the mantissa because in any binary number the left-most (MSB) bit is always a 1. Therefore, this 1 is understood to be there although it does not occupy an actual bit position.

The eight bits in the exponent represent a biased exponent, which is obtained by adding 127 to the actual exponent. The purpose of the bias is to allow very large or very small numbers without requiring a separate sign bit for the exponents. The biased exponent allows a range of actual exponent values from -126 to + 128.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

Let us consider a binary number 1011010010001. First, it can be expressed as 1 plus a fractional binary number by moving the binary point 12 places to the left and then multiplying by the appropriate power of two.

$1011010010001 = 1.011010010001 \times 2^{12}$

Assuming that this is a positive number, the sign bit (S) is 0. The exponent, 12, is expressed as a biased exponent by adding it to 127 (12+127 = 139). The biased exponent is expressed as the binary number 10001011. The mantissa is the fractional part (M) of the binary number, .011010010001. The floating point representation of the binary number is

S	E	M
0	10001011	01101001000100000000000

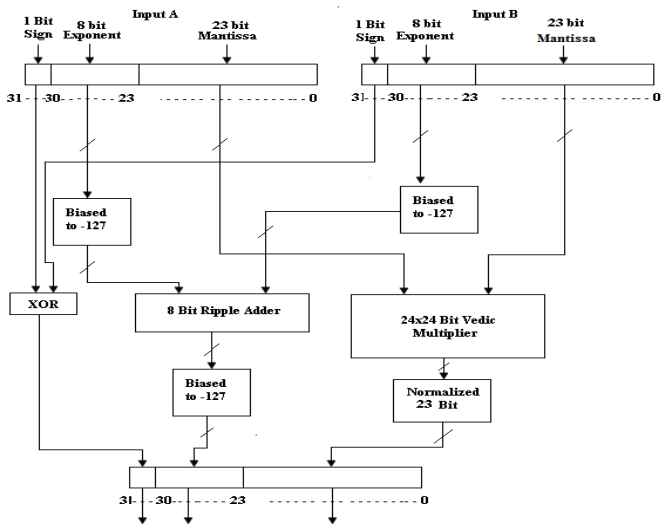


Fig.2: Architecture of Single Precision Floating Point Multiplier.

The multiplier for the floating point numbers represented in IEEE 754 format can be divided in four different units:

- Mantissa Calculation Unit
- Exponent Calculation Unit
- Sign Calculation Unit
- Control Unit

The Mantissa Calculation Unit requires a 24 bit multiplier. The Vedic Multiplication technique is chosen for the implementation of this unit. 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 Exponent Calculation Unit is implemented using 8 BIT Ripple Carry Adder. The advantages of ripple carry adder in addition to its implementation ease are low area and simple layout. The Control Unit raises the flag when NaN, Infinity, zero, underflow and overflow cases are detected.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

The control unit raises appropriate flag accordingly when the cases occurs. The various cases and its constituent flags are:

If e = 255 and f ≠ 0, then NaN
If e = 255 and f = 0, then Infinity
If 0 < e < 255, then Number is $(-1)^S \ 2^{(e-127)} \ (1 \cdot f)$ (Normalized numbers)
If e = 0 and f ≠ 0, $(-1)^S \ 2^{(-126)} \ (0 \cdot f)$ (Denormalized numbers)
If e = 0 and f = 0, then zero.

D. Vedic Mathematics

Vedic Mathematics is the name given to the ancient Indian system of mathematics that was rediscovered in the early twentieth century from ancient Indian sculptures (Vedas). The word 'Vedic' is derived from the word 'Veda' which means the store-house of all knowledge. It mainly deals with Vedic mathematical formulae and their application to various branches of mathematics. It was re-introduced to the world by Jagadguru Swami Bharati Krishna Tirtha Sankaracharya Maharaja. It refers to the technique of Calculations based on a set of 16 Sutras, or aphorisms, as algorithms and their upa-sutras or corollaries derived from these Sutras. It is more coherent than modern mathematics. The algorithms based on conventional mathematics can be simplified and even optimized by the use of Vedic Sutras. These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral), and applied mathematics of various kinds.

The simplicity of Vedic Mathematics means that calculations can be carried out mentally though the methods can also be written down. It offers a fresh and highly efficient approach to mathematics covering a wide range - starts with elementary multiplication and concludes with a relatively advanced topic, the solution of non-linear partial differential equations. But the Vedic scheme is not simply a collection of rapid methods; it is a system, a unified approach. It extensively exploits the properties of numbers in every practical application.

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya-Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus.

Vedic mathematics is not only a mathematical wonder but also it is logical. That's why Vedic Mathematics has such a degree of eminence which cannot be disapproved. Due these phenomenal characteristic, Vedic Mathematics has already crossed the boundaries of India and has become a leading topic of research abroad. Vedic Mathematics deals with several basic as well as complex mathematical operations.

E .Urdhva-Triyakbhyam Sutra

Urdhva Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done and then, concurrent addition of these partial products can be done. Thus parallelism in generation of partial products and their summation is obtained.

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. Thus any multi-bit multiplication can be reduced down to single bit multiplication and addition using this method. Moreover, the carry propagation from LSB to MSB is reduced due to one step generation of partial product.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

II. MULTIPLIER DESIGN

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 [8]. The aim is to pipeline each unit a sufficient number of times in order to maximize speed of the Multiplier. As the number of pipelining stages increases, speed as well as throughput also increases.

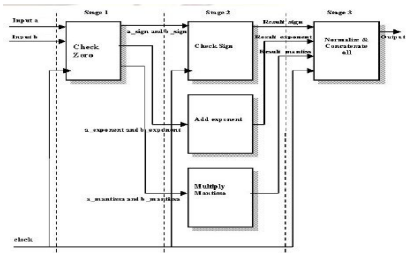


Fig.3: A 3-stage Pipelined Floating Point Multiplier.

A. 2X2 Vedic Multiplier

In 2x2 bit multiplier, the multiplicand has 2 bits each and the result of multiplication is of 4 bits. So in input the range of inputs goes from (00) to (11), output lies in the set of (0000, 0001, 0010, 0011, 0100, 0110, 1001). By using Urdhva Tiryakbhyam, the multiplication takes place as illustrated in Fig.2. Here multiplicands a and b are taken to be (10) both. The first step is the vertical multiplication of LSB of both multiplicands, and second step is the crosswise multiplication and addition of the partial products. Then Step 3 involves vertical multiplication of MSB of the multiplicands and addition with the carry propagated from Step 2.

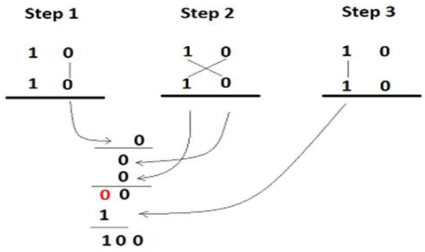


Fig 4: 2X2 Multiplication using Urdhva Triyakbhyam Sutra.

The hardware realization of 2x2 multiplier blocks is illustrated in Fig.4. Let the two 2-bit binary numbers be a1, a0 and b1, b0, then the product of these terms are

q0= a0.b0
q1=(a1*bo) +(a0*b1)
q2=(a1*b1) +Ci
q3=C_0

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

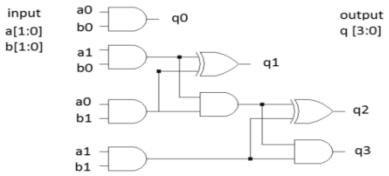


Fig.5: Hardware Realization of 2X2 block.

B. 4X4 Vedic Multiplier

The block diagram of 4x4 Vedic Multiplier is shown in Figure 6.This multiplier is modelled using structural style of modeling using VHDL. In this paper first a 2x2 Vedic Multiplier is implemented using the above mentioned method. The 4x4 Vedic Multiplier is designed using four 2x2 Vedic Multipliers. After that 8x8 Vedic Multiplier is implemented using four 4x4 Vedic Multipliers. The 16x16 Vedic Multiplier is made using four 8x8 Vedic Multipliers. Finally, the 32x32 Vedic Multiplier is designed using four 16x16 Vedic Multipliers.

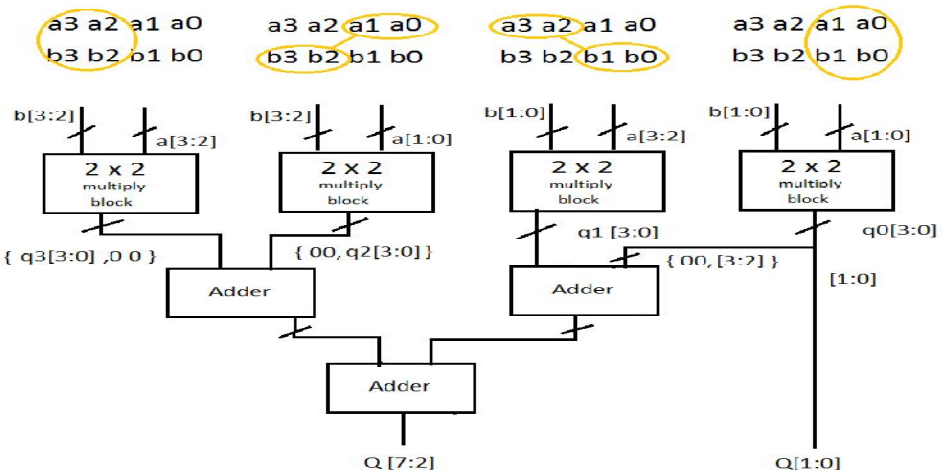


Fig.6: 4x4 Vedic Multiplier block.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

C. 32X32Vedic Multiplier

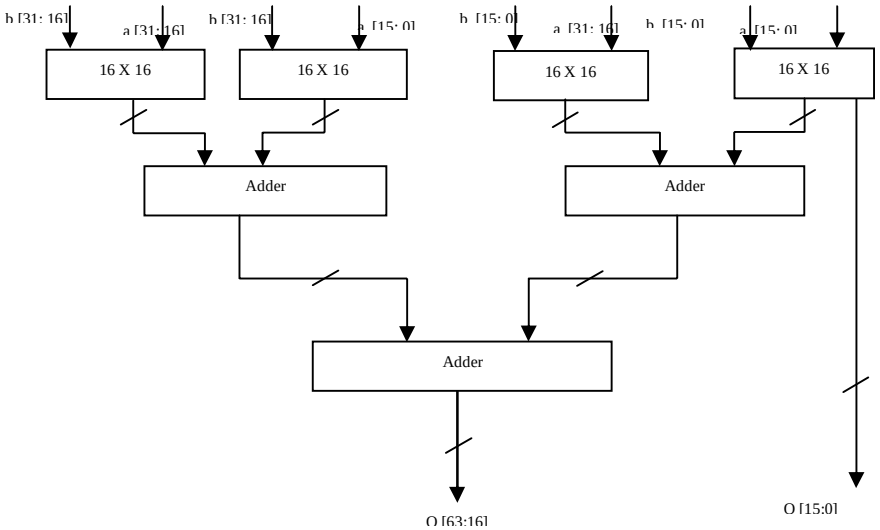
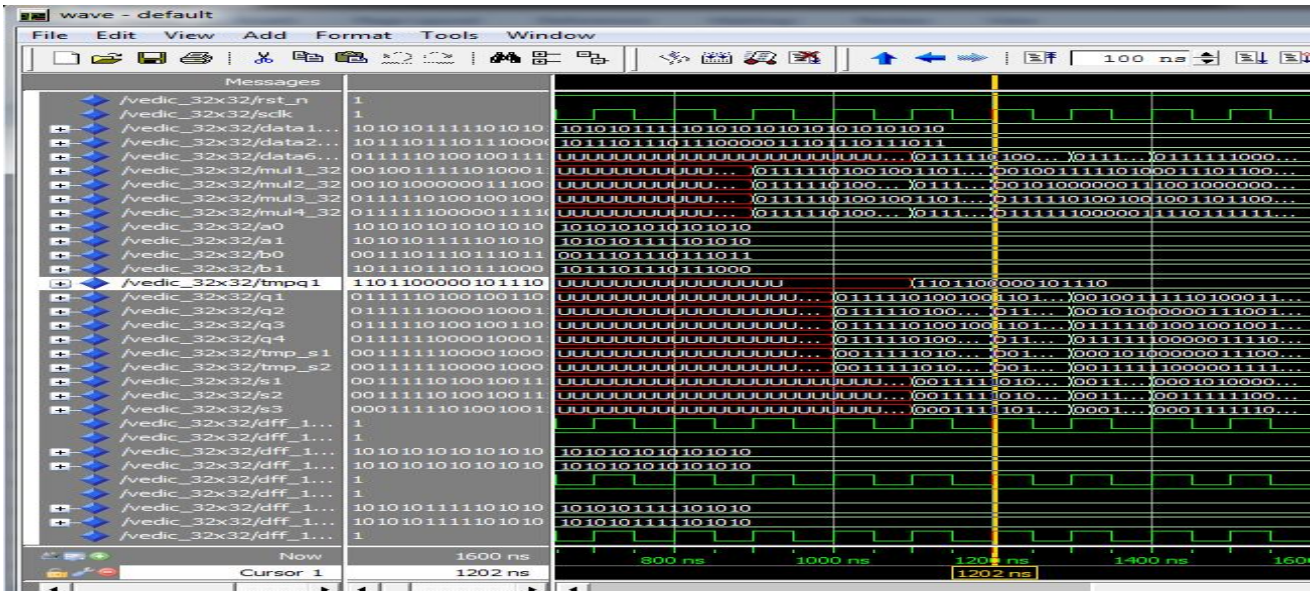


Fig.7: 32X32 Vedic Multiplier block.

III. SIMULATION RESULTS



Simulation Result of 32x32 Vedic Multiplier.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 3, Special Issue 5, July 2014

International Conference On Innovations & Advances In Science, Engineering And Technology [IC - IASET 2014]

Organized by

Toc H Institute of Science & Technology, Arakunnam, Kerala, India during 16th - 18th July -2014

IV. CONCLUSION

The paper shows the efficient use of Vedic multiplication method in order to multiply two floating point numbers. Here we introduce the concept of pipelining so that lesser number of LUTs verifies that the hardware requirement is reduced, thereby reducing the power consumption without compromising delay so much.

V. SCOPE OF FUTURE WORK

An improvement in multiplication speed by using new techniques can greatly improve system performance. This project can be extended for the reconfigurable architecture. Double Precision Floating Point Multiplier can be implemented.

VI. REFERENCES

[1]. Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics Sixteen Simple Mathematical Formulae from theVeda,"1965.
[2]. Kavita Khare, R.P.Singh, Nilay Khare,"Comparison of pipelined IEEE-754 standard floating point multiplier with unpipelined multiplier" Journal of Scientific & Industrial Research Vol.65, pages 900-904 November 2006.
[3]. Brian Hickmann, Andrew Krioukov, and Michael Schulte, Mark Erle,"A Parallel IEEE 754 Decimal Floating-Point Multiplier," In 25th International Conference on Computer Design ICCD, Oct. 2007.
[4]. Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho Dadgour, "Multiplier design based on ancient Indian Vedic Mathematics" International SoC Design Conference,2008.
[5]. Metin Mete ÖZBLEN, Mustafa GÖK, "A Single/Double Precision Floating-Point Multiplier Design for Multimedia Applications," Journal of Electrical & Electronics Engineering. Vol.9, Number 1, pp 827-831 2009.
[6]. Al-Ashrafy, M.; Salem, A.; Anis, W, "An efficient implementation of floating point multiplier," Electronics Communications and Photonics Conference (SIECPC), 2011.
[7]. Neelima Koppala, Rohit Sreerama, Paidi Satish "Performance Comparison of Fast Multipliers Implemented on Variable Precision Floating Point Multiplication Algorithm", International Journal of Computer Applications in Engineering Sciences, Vol II, Issue II, June 2012.
[8]. Aniruddha Kanhe, Shishir Kumar Das and Ankit Kumar Singh, "Design And Implementation Of Low Power Multiplier Using Vedic Multiplication Technique," International Journal of Computer Science and Communication (IJCSC) Vol. 3, No. 1, January-June 2012, pp. 131-132.
[9]. Riya Saini, R.D.Daruwala "Efficient Implementation of Pipelined Double Precision Floating Point Multiplier" International Journal of Engineering Research and Applications, Vol. 3, Issue 1, January -February 2013, pp.1676-1679.