IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 6, August 2014.
www.ijiset.com
ISSN 2348 – 7968

Area Efficient Vedic Multiplier for Digital Signal Processing Applications

[1]U.Umakantha Reddy, [2]V.SriRami Reddy

[1]Assistant professor, Department of ECE., ACET, Allagadda,
[2]M.Tech (VLSISD) Student, Department of ECE., ACET, Allagadda,

Abstract- ***This paper proposes a method for area efficient fractional fixed point(Q-format) multiplier based on Urdhava Tiryakbhyam of vedic mathematics. Even though conventional or normal Urdhava multipliers carries high speed mathematical operations, they consume more chip area. Hence we proposed a pipelined multiplier architecture in this paper which consumes less chip area. The pipelined multiplier architecture consists of 3 stages. 1st stage consists of the n - bit vedic multiplication unit. 2nd stage consists of partial products and carry. 3rd stage consists of adders and the result of the multiplication. This paper presents the efficiency interms of area for Urdhva Triyakbhyam Vedic methods of both conventional and proposed multipliers. The proposed algorithm is modeled using Verilog, a hardware description language. Implementation has been done for the Xilinx FPGA device, Spartan-3E. The results shows that multiplier implemented using pipelined Vedic multiplication is efficient in terms of area compared to its implementation using normal Urdhava multiplication method .***

Keywords- Q-format; fixed point arithmetic ;Vedic Mathematics; Urdhva Tiryakbhyam Sutra; Booth Multiplier; High Speed.

1. INTRODUCTION

Multiplier [1] is one of the key hardware blocks in designing arithmetic, signal and image processors. Many transform algorithms like Fast Fourier transforms (FFTs), DFT etc make use of multipliers [2], [3], [4]. With advances in technology, many researchers have tried to design multipliers which offer high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. In recent years, high-speed multipliers [5] play an important role while designing any architecture and researchers are still working on many factors to increase the speed of operation of these basic elements. Algorithms for designing high-speed multipliers have been modified and developed for better efficiency [6]. The increased complexity of various applications, demands not only faster multiplier chips but also smarter and efficient multiplying algorithms that can be implemented in the chips. It is up to the need of the hour and the application on to which the multiplier is implemented and what tradeoffs need to be considered. Generally, the efficiency of the multipliers is classified based on the variation in speed, area and configuration. The multiplier architecture can be generally classified into three categories. First is the serial multiplier which emphasizes on hardware and minimum amount of chip area. Second is parallel multiplier (array and tree) which carries out high speed mathematical operations. But the drawback is the relatively larger chip area consumption. Third is serial-parallel multiplier which serves as a good trade-off between the times consuming serial multiplier and the area consuming parallel multipliers. The proposed Vedic multiplier is based on the Vedic Sutras. These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware.

2. VEDIC MATHEMATICS

The Sanskrit word 'Veda' means 'knowledge'. The Vedas consist of a huge number of documents there are said to be thousands of such documents in India, many of which have not yet been translated, which are shown to be highly structured, both within themselves and in relation to each other. Some documents, called 'Ganita sutras' (the name

ISSN 2348 – 7968

'ganita' means mathematics), were devoted to mathematical knowledge. Sri Bharati Krishna Tirtha Maharaj, who is generally considered the doyen of this discipline, in his seminal book Vedic Mathematics, wrote about this special use of sutras [J I). Vedic Mathematics" was the name given by him. He was the person who collected lost formulae from the writings of "Atharwa Vedas" and wrote them in the form of Sixteen Sutras and thirteen sub-sutras. Vedic Mathematics is based on 16 sutras dealing with mathematics related to arithmetic, algebra, and geometry. These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus and applied mathematics of various kinds. The Vedic methods are direct, and truly extraordinary in their efficiency and simplicity. Research is being carried out in many areas, including the effects on children who learn Vedic maths and the development of new, powerful but easy applications of the Vedic sutras in geometry, calculus, computing etc. But the real beauty and effectiveness of Vedic mathematics cannot be fully appreciated without actually practising the system. One can then see that it is perhaps the most refined and efficient mathematical system possible.

2.1 URDHVA TIRYAKBHYAM SUTRA

TheUrdhava Tiryakbhyam [11] is a Sanskrit word which means vertically and crosswire in English. The method is a general multiplication formula applicable to all cases of multiplication. It is based on a novel concept through which all partial products are generated concurrently. Fig. 1demonstrates a 4 x 4 binary multiplication using this method.The method can be generalized for any N x N bit multiplication. This type of multiplier is independent of the clock frequency of the processor because the partial products and their sums are calculated in parallel.

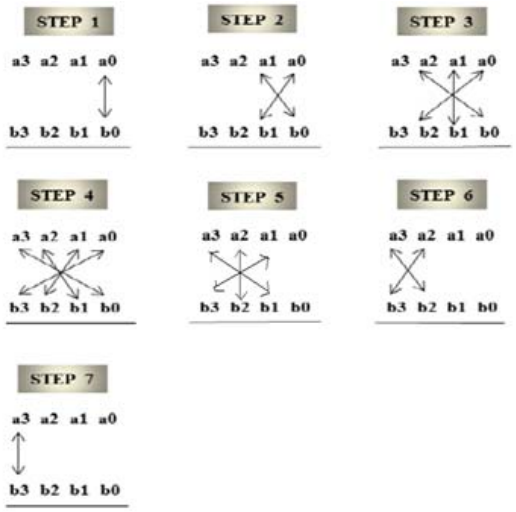

Figure.1 multiplication of two 4-bit numbers of using Urdhva Tiryakbhyam.

The line diagram in fig. 1 illustrates the algorithm for multiplying two 4-bit binary numbers $a_3a_2a_1a_0$ and $b_3b_2b_1b_0$. The procedure is divided into 7 steps and each step generates partial products. Initially as shown in step 1 of fig. 2, the least significant bit (LSB) of the multiplier is multiplied with least significant bit of the multiplicand (vertical multiplication). This result forms the LSB of the product. In step 2 next higher bit of the multiplier is multiplied with the LSB of the multiplicand and the LSB of the multiplier is multiplied with the next higher bit of the multiplicand(crosswire multiplication). These two partial products are added and the LSB of the sum is the next higher bit of the final product and the remaining bits are carried to the next step. For example, if in some intermediate step, we get the result as 1101, then 1 will act as the result bit(referred as rn) and 110 as the carry (referred as cn). Therefore cn may be a multi-bit number. Similarly other steps are carried out as indicated by the line diagram. The important feature is that all the partial products and their sums for every step can be calculated in parallel. Thus every step in fig. 2 has a corresponding expression as follows

r0=a0b0+1'b0+1'b0+1'b0
c1r1=a1b0+a0b1+1'b0+1'b0
c2r2=c1+a2b0+a1b1+a0b2+1'b0
c3r3=c2+a3b0+a2b1+a1b2+a0b3
c4r4=c3+a3b1+a2b2+a1b3+1'b0
c5r5=c4+a3b2+a2b3+1'b0+1'b0
c6r6=c5+a3b3+1'b0+1'b0+1'b0

ISSN 2348 – 7968

With c6r6r5r4r3r2rlrO being the final product. Partial products are calculated in parallel and hence the delay involved is just the time it takes for the signal to propagate through the gates.

3. IMPLEMENTATION OF VEDIC MULTIPLIER

3.1 CONVENTIONAL MULTIPLIER:

The hardware architecture of existing conventional multiplier is shown in below figure, which is 4bit multiplier.this uses completely combinational circuits.

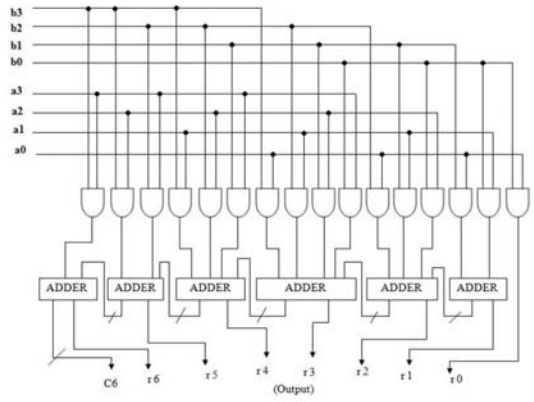

Fig. 2.Hardware architecture of 4 X 4 conventional Urdhva Tiryakbhyam multiplier.

3.2THE PROPOSED ALGORITHM:

The pseudo code of the proposed multiplication algorithm is described. The algorithm is broadly divided in three stages namely the initialization, pre-processing and processing. The proposed multiplication algorithm is for n-bit multiplication using basic shifting, multiplication and addition operations.

3.2.1 PSEUDO CODE

(a) Initialization

Positive edge clock (elk) and negative edge reset (rst_n) is initialized.

Reg [n-2:0] = O. (Temporary Registers)
P [n-l :0] = O. (Partial Products)
Cy [n-3:0] = O. (Carry)
Load enable (Iden) = O.
Operation enable (open) = O.

(b) Preprocessing

Input n-bit binary numbers a and b
a[n-l :0] = Multiplicand
b[n-l :0] = Multiplier
Load Enable (Iden) = 1.
Operation enable (open) = O.
temp_b [n-l:0] = Right shift of b.

(c) Preprocessing

Operation enable (open) = O.

1. Right shift the multiplier b [n-l :0] n*2 times
2. Multiply a and b w r t expressions
3. Add [n-2:0] = {p (0) + P (1) + ... + P (n-2) + p (n-l)} + cy (n-3:0).
4. Result = right shift add [n-(4*multiple of 4)] n*2 times.
5. Return the result.
6. END.

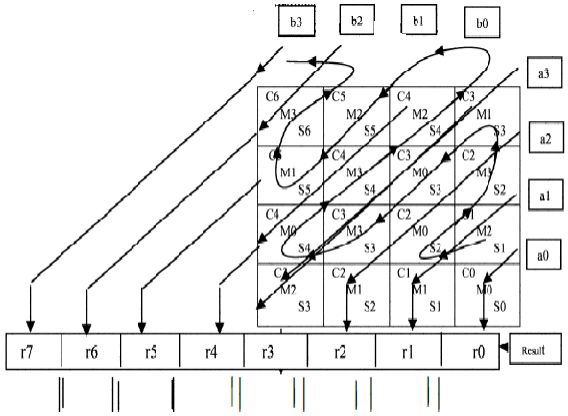

Fig. 3 Proposed Vedic multiplier architecture for 4-bit multiplication.

3.2.2 ANALYSIS OF ALGORITHM

The proposed multiplication algorithm is hierarchical in structure i.e., top-down approach. In our algorithm, we consider nibble as leaf cell for the higher bit multiplications design. Thus, using basic nibble multiplication unit, the higher bit multiplication can be built which will makes modification easier. Hence debugging is easy and efficient in area and speed. In the conventional method, for n-bit multiplication

ISSN 2348 – 7968

n*4 AND gates are used for implementation. The proposed multiplication algorithm uses n
AND gates for n-bit multiplication. Thus, area consumption is less. The method uses sequential circuits in the design that used for different frequency ranges. Speed is increased by reducing the delay in sequential circuits. The above figure shows Proposed way of nibble multiplication by Urdhva Tiryakbhyam Sutra. For the case, nibble multiplication of a [3:0] and b[3:0]. The analysis is as follows. It takes n*2 clock cycles for n-bit multiplication. In nibble
multiplication 8 clock cycles are required to get the result. In the 151 clock cycle, a[O] is multiplied with b[O] and a[l], a[2], a[3] are multiplied with reg[O], reg[l], reg[2] respectively. After getting the partial products p[3:0], added with carry using simple adder. The add [0] is feed to the result and add[2: 1] acts
as carry. For next clock cycles, b[3:0] is right shifted by 1 bit and multiplied with corresponding a[3:0] bits. Then partial products are added with previous carry using simple adder and add [0] is feed to the result and previous addition result is right shifted by 1 bit. Likewise, for each clock cycle the method follows. After 7th clock cycle result is generated and stored in 8 bit register. In this way the multiplier operation was performed. In this paper we extend the 4- multiplication to 16-bit multiplication and 32-bit multiplication for Q15 and Q31 numbers respectively.

4. FIXED POINT ARITHMETIC

An N-bit fixed point number can be interpreted as either an integer or a fractional number. Integer fixed point is difficult to use in processors due to possible overflow. For e.g. In a 16-bit processor for signed integers the dynamic range is from -2^{15} to 2^{15}-1 i.e. 32768 to 32767. If 500 are multiplied by 800 the result is 40000 which is an overflow. In order to overcome this situation fractional fixed point representation also known as Q-format is used.

4.1 Q-FORMAT REPRESENTATION

In general any Q-format representation is denoted by Q*m.n*, where *m* is the number of bits to represent integer, *n* denotes number of bits to represent fractional part and the total number of bits is given by N = *m*+*n*+1 for signed numbers. For e.g. Q4.11 format signifies that a total of 16 bits are required to represent a fractional number in which 4 bits are reserved for the integer part and 11 bits for the fractional part and 1 bit indicates sign. Special cases of Q-format consist of zero bits to represent the integer part. Q0.15 (Q15) and Q0.31 (Q31) are two such formats for 16 bit and 32 bit representations respectively. As there is no integer part the fractional number has a range between 1 and -1. Therefore the products of such numbers also lie between 1 and -1. This property is best suited for implementing multipliers as the bit length of the product is same as the input bit length and thus Q-format finds its application in digital signal processing hardware.

An N-bit number in Q*m.n* format is represented as follows.

$a_{n+m}a_{n+m-1}\ldots\ldots\ldots a_n.a_{n-1}\ldots\ldots\ldots\ldots a_1a_0$

Here the '.' between an.an$_{-1}$ represents the fixed point and
value of (1) is given by, $(a_{n+m}2^{N-1}+a_{n+m-1}2^{N-2}\ldots \ldots+a_2 2^2+a_1 2+a_0)2^{-n}$

When we want to convert a fractional number in the range of the desired Q*m.n* format, we multiply it with 2^n. The resultant value is truncated or rounded off to the nearest
integer. Therefore a small amount of precision loss is involved which reduces as the number of bits representing the fractional part increases. We prefer rounding technique since its error bias in both positive and negative direction is same. Therefore the rounded value will be more precise. For e.g. Conversion of 0.2625 to Q15 format is done by multiplying it with 2^{15} which equals to 8601.6 which when rounded gives 8602. This is stored as 0010000110011010 in a16 bit memory location. The most significant bit indicates sign of the number. If it is negative then 2's complement method is followed to store the number. Thus a fraction is converted to an integer in a Q-format and the choice of the decimal point lies entirely in the hands of the programmer. In general a Q*m.n* format has a resolution of 2^{-n} and its dynamic range lies between -2^m to 2^m-2^{-n}. Therefore as the number of bits for fractional representation increases the resolution increases and as the number of bits for integer part increases the dynamic range increases. The resolution of Q15 format is 2^{-15}, and for Q31format it is 2^{-31}. Therefore a number

ISSN 2348 – 7968

represented in Q31 format has higher resolution and is more precise than the one in Q15 format. In this paper we mainly concentrate on Q15 and Q31 formats since they are best suited for implementing multipliers for DSP applications.

4.2 Q-FORMAT MULTIPLICATION

When two Q15 numbers are multiplied their product is 32 bits long as illustrated in Figure1. The product has a redundant or extended sign bit. Since the product stored in memory should also be a Q15 number we left shift the product by one bit and the most significant 16 bits (including sign bit) is stored in the memory. Figure. 1 demonstrates multiplication of two Q15 format numbers. The process remains same for Q31 format wherein after left shifting the product by one bit, the most significant 32 bits are stored in the memory. Similarly for Q31 numbers also. Therefore with Q-format, multiplications of two fractional numbers can be carried out by using integer multiplications. Integer multiplications consume less area and are faster compared to floating point multipliers which is the major advantage of Qformat representation

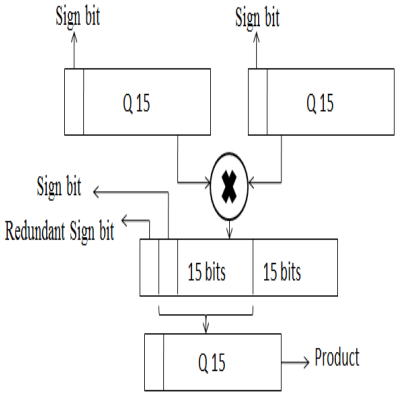

Figure 4. Multiplication of two Q15 format numbers yielding the product in Q15 formats itself.

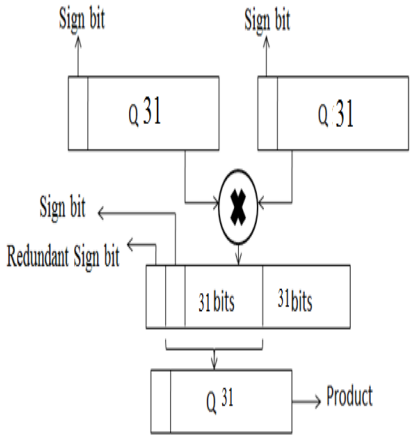

Figure 5.Multiplication of two Q31 format numbers yielding the product in Q31 format itself.

5. IMPLEMENTATION AND RESULTS

The algorithm is designed for Q15 and Q31 multipliers (which use proposed multiplier architecture) using Verilog-HDL. Simulation is done using Isim on xilinx ise13.2, Synthesis and Implementation is done using Xilinx Spartan-3E FPGA Board of device family xc3s500e-4fg320.Further the Q-format multipliers were also implemented using normal(conventional) urdhava multiplier.

A. Simulation Results:

The design was simulated using Isim on Xilinx ISE 13.2 version.

For Q15 format multiplication as shown in fig.

Input1 =-0.75 = 1010 0000 0000 0000

Input2 =- 0.25 = 1100 0000 0000 0000

Output = 0.1875 =0001 1000 0000 0000

For Q31 format multiplication as shown infig.7,

Input1=- 0.666666=10101010101010101011000001000010

Input2=0.333333=00101010101010101010100111110111 11

Output= 1110 0011 1000 1110 0011 1100 1001 1110 whose value is -0.22222177777439355850219726556 25.But the actual value of the product is -0.222221777778. Therefore precision loss is involved in this multiplication and is found to be 3.60644E-12 which is less than the resolution of Q31 representation i.e. 2^{-31}.

ISSN 2348 – 7968

Thus it provides 32 bit accurate product whichh is acceptable for most of the DSP applications.

The subsequent figures show the simulation and Technical Schematic of Q15 and Q31 multiplications.

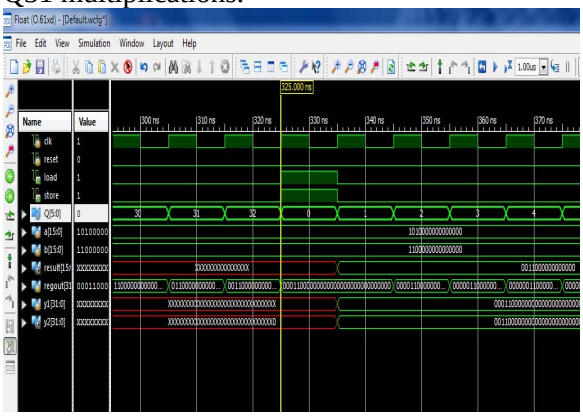

Figure 6. Simulation results for Q15 multiplier**.**

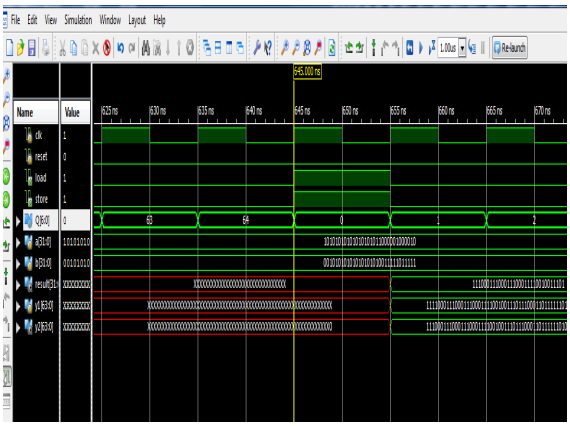

Figure 7. Simulation results for Q31 multiplier.

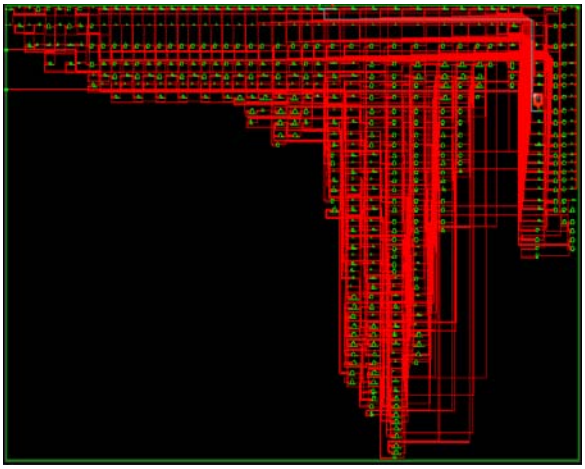

Figure 8. Technology view of Q15 multiplier using Vedic mathematics.

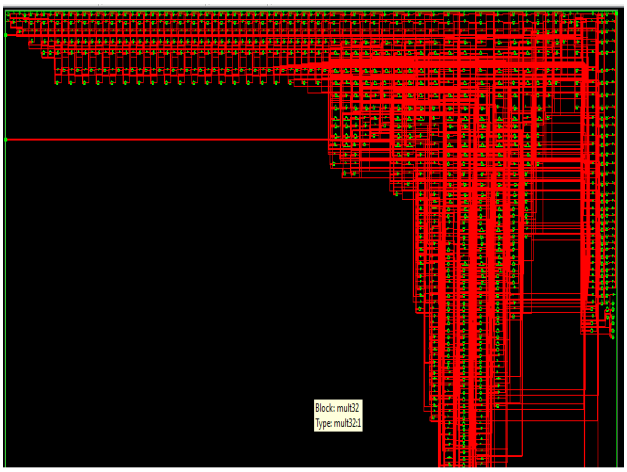

Figure 9. Technology view of Q31 multiplier using Vedic mathematics.

COMPARISON OF 16 BIT Q15-FORMAT MULTIPLIERS

	4-input LUT usage	Frequency (MHz)
Normal urdhava multiplier	537/9312	236.18
Proposed urdhava multiplier	194/9312	91.929

COMPARISON OF 32 BIT Q31-FORMAT MULTIPLIERS

	4-input LUT usage	Frequency (MHz)
Normal urdhava multiplier	1303/9312	158.90
Proposed urdhava multiplier	426/9312	56.667

ISSN 2348 – 7968

6. CONCLUSION

The paper proposed a pipelined multiplier architectue for signed q-format multiplications using Urdhava Tiryakbhyam method of vedic mathematics. Since Q-format representation is widely used in Digital Signal Processors the proposed multiplier can substantially reduces the occupied chip area of multiplier which is the basic hardware block. The synthsis results had shown that they occupy less area and are moderate speed compared with normal (conventional) Urdhava multipliers.Future work of this project is in the direction of implementing serial-parallel architecture for multiplier to increase the speed of multiplier with small incrase in chip area compared with proposed multiplier.

REFERENCES

[I] H.Guilt, "Fully Iterative Fast Array for Binary Multiplication", Electronics Letters, vol. 5, p. 263, 1969

[2] Vadiraj Sagar, Shripad Sagar, Sudhindracharya, Vedavyas Mathad, Subhash Kulkarni "Vhdl Implementation of Vedic Mathematical Sutras" Department of Electronics & Communication, P DA College of Engineering.

[3] "Lifting Scheme Discrete Wavelet Transform Using Vertical and Crosswise Multipliers" Anthony O'Brien and Richard Conway, ISSC, 2008,Galway, June 18-19.

[4] Zhijun Huang, Milos D. Ercegovac, "High-Performance Lefl-to-Right Array Multiplier Design," arith, pp.4, 16th IEEE Symposium on Computer Arithmetic (ARITH-16 '03), 2003

[5] Ramalatha, M Dayalan, K D Dharani, P Priya, and S Deborah, "High speed energy efficient ALU design using Vedic multiplication techniques", ICACTEA, 2009. pp. 600-3, Jul 15-17, 2009.

[6] H ThapJiyal, M B Srinivas, and H R Arabnia, "Design and Analysis of a VLSI Based High Performance Low Power Parallel Square Architecture", in Proc. Int. Con! Algo. Math. Compo Sc., Las Vegas, pp. 72-6, Jun. 2005.

[7] Harpreet Singh Dhillon Abhijit Mitra, "A Digital Multiplier Architecture using Urdhava Tiryabhyam Sutra of Vedic Mathematics", Indian Institue of Technology, Guwahatti.

[8] Purushottam D. Chidgupkar and Mangesh T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing on 8085/8086", Global J. of Engng. Educ., Vol.8 No.2 © 2004 UICEE Published in Australia.

[9] Himanshu Thapliyal and Hamid R. Arabnia, "A Time-Area- Power Efficient Multiplier and Square Architecture Based on Ancient Indian VediC", Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[10] M.M. Mano, "Computer system Architecture", Englewood Cliffs, NJ: Prentice-hall,1982

[11] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja," Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.

[12] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced- Bit Multiplication Algorithm for Digital Arithmetics ", International Journal of Computational and Mathematical Sciences, 2008.

V. SriRami Reddy was born in A.P, India. He received B.TECH degree in **Electronics and Communication Engineering** from Jawaharlal Nehru technological university Anantapur in 2011. Presently he is pursuing M.Tech VLSI system Design in ACET, Allagadda. He research interests include FPGA Implementation, Low Power Design.