RESEARCH ARTICLE OPEN ACCESS

Design of Arithmetic Unit for High Speed Performance Using Vedic Mathematics

Rahul Nimje, Sharda Mungale
Department of Electronics Engineering Priyadarshini College of Engineering Nagpur, India
rahulnimje22@gmail.com
Department of Electronics Engineering, Priyadarshini College of Engineering Nagpur, India
Sharda_37mungale@rediffmail.com

Abstract-
Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculation based on 16 formula (sutras). The word Vedic is desired from word Veda which store house of all knowledge. As the ever increasing demand in enhancing the ability of coprocessor to handle the complex and challenging processor as resulted in integration of number of processor cores into single chip, but still the load on the processor is not less in generic system. This load is reduced by connecting the main processor with co processor, which are designed to work on the specific types of function like numeric computation, signal processing, image processing and arithmetic operation. The speed of arithmetic is of extreme importance and depends greatly on the speed of multiplier. Therefore the technologies are always looking for new algorithm and hardware so as to implement this operation in much optimized way in the terms of area and speed. Vedic Mathematics deals with various branches of mathematics like arithmetic, algebra, geometry etc in computation algorithm of the coprocessor which will reduce the complexity of execution time , area and power consumption etc. The efficiency of Urdhav Triyagbhyam Vedic method for multiplication, strikes a difference of actual process of multiplication, by enabling parallel generation of intermediate product, eliminating unwanted multiplication steps with zeros and scaled to higher bit level. This formula (Sutras) is used to build high speed power efficient multiplier in coprocessor.
This project is to design arithmetic module using the technique of ancient Indian Vedic Mathematics to improve the performance of coprocessor. This project is to design NxN arithmetic module, where A & B are the two N bits inputs of these module and different sections of module are multiplier which is designed by using Vedic algorithm of multiplication named Urdhav Triyagbhyam multiplier and with carry save adder, adder/ subtractor and MAC unit.
Keywords : DSP, Arithmetic Logical Unit, Adder, Multiplication,Vedic Urdhav Triyambakam multiplication algorithm, Vedic Multiplier (VM).

I. Introduction

As many of know that the Computation unit is main unit of any science and technology, which perform different arithmetic operation like as addition, subtraction and multiplication etc. also in some places it perform logical operation such as add, or, invert, x-or etc. Multiplexer are extensively used in Microprocessor, DSP and Communication application. For higher order multiplication, a huge number of adder are used to perform the partial product addition. The need of high speed processor is increasing. Higher throughput arithmetic operation is important to achieve the desired performance in many real time signal and image processing application. Among these units the performance of any processor majorly depends on the time taken by ALU to perform the specified operation. The important fundamental function in arithmetic operation is multiplication. The multiplication based operation such as Multiply and Accumulate (MAC) and inner product are among some of frequently used Computation Intensive (CIAF) Currently implemented in many Digital Signal Processing (DSP) application such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit. As the multiplication operation manage the execution time of most DSP algorithm, so there is a need of high speed multiplier for execution. The core computing process is always a multiplication routing; therefore engineers are constantly looking for new algorithm and hardware to implement technology.
The demand for high speed processing day by day has been increasing as a result of expanding computer and signal processing application. Higher throughput arithmetic operation is important to achieve the desired performance in many real- time signal and image processing application. One of the key arithmetic operations in such application is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing time delay and power consumption are very essential requirement for many application. In any processor the major unit is control unit, ALU and memory read write. ALU is an execution unit which

not only performs the arithmetic operation but also logical operation and therefore ALU is called a heart of Microprocessor, Microcontroller, and CPUs. No technology uses works upon that operation either fully or partially which are performs by ALU. The block diagram of ALU given below in figure (1), where ALU has been implemented on FPGA tool

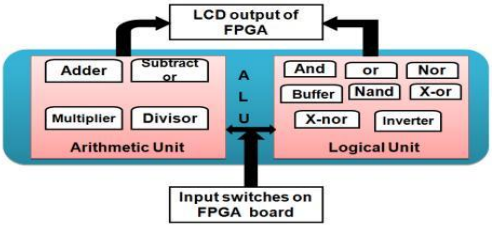


Figure 1: Block Diagram of an ALU

Here the input interface to access ALU module is input switches on FPGA board, and after processing on the data the result can be seen from LCD output of FPGA.

PROPOSED ARITHMETIC UNIT:

Our Proposed 32x32 bit Arithmetic Unit is shown in the following:-

CPUs. No technology uses works upon that operation either fully or partially which are performs by ALU. The block diagram of ALU given below in figure (1), where ALU has been implemented on FPGA tool

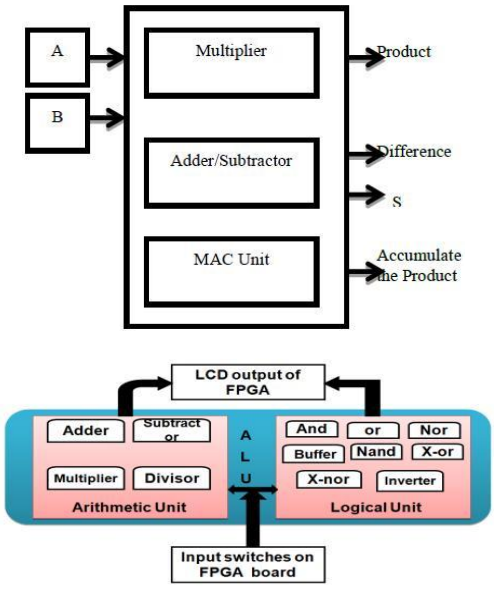


Figure 1: Block Diagram of an ALU

Here the input interface to access ALU module is input switches on FPGA board, and after processing on the data the result can be seen from LCD output of FPGA.

PROPOSED ARITHMETIC UNIT:

Our Proposed 32x32 bit Arithmetic Unit is shown in the following:-

FPGAs are well suited to datapath designs come across in digital filtering applications which encountered in digital filtering applications which operations on a single device. In particular, multiple multiply-accumulate (MAC) units [2] [3] may be implemented on a single FPGA, which provides comparable performance to general-purpose architectures which have a single MAC unit.

This work focuses on using variable precision multiplication approach with adder applicable to FPGAs. The word sizes can be chosen to balance the size of the implementation, which is limited by the FPGA density, against the numerical precision. Larger word sizes are possible if the number of MAC units per chips is reduced. The increase in density of FPGAs in the future will surely expand the design space available to the designer, and make such constraints less severe

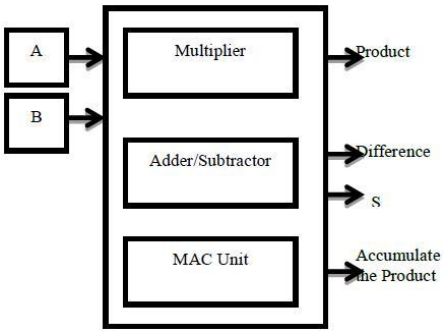


Figure 2: Arithmetic Unit

Here the A and B are the two 32 bit inputs of our Arithmetic Unit. And other sections of the design are self-explanatory. We have not given much importance on the designing of the adder/subtractor circuits as they are not listed as the modules which consumes high amount of area and power in ALU. But after a deep we found that in general, carry ripple adders can be used when they meet timing constraints because they are compact and easy to build. At word lengths of 32, tree adders are distinctly faster.

MAC UNIT:

Due to ever-increasing IC i.e. integrated circuit fabrication capabilities, the future of FPGA technology promises both higher densities and higher speeds. Many FPGA families are based on memory technology, so the improvements in those areas should associate with FPGA evolution.

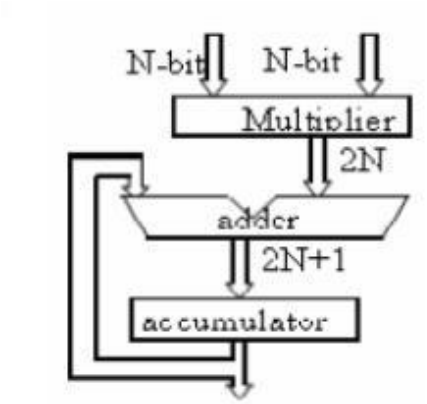


Figure 3: Basic structure of MAC Unit

The basic structure of MAC Unit is shown in fig. 2 it has two N-bit inputs (i.e A and B) given to multiplier which produces 2N bits as output then an adder is used to add the present and previous bits output of multiplier and accumulator, so the Output bits are usually 2N+1 bits. For floating point numbers based MAC where rounding is required at least once or twice is referred to as fused multiply–add (FMA) or fused multiply– accumulate (FMAC).

For multiplication purpose Vedic Urdhva Triyambakam multiplication scheme has been used. Urdhva Triyakbhyam Sutra (formula) is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". To demonstrate this multiplication scheme, let us consider multiplication of two decimal numbers (21 x 32). The conventional methods for calculation already know to us it is a time consuming method and for the present condition of requirement we need speed, so because of that an alternative method of multiplication using Urdhav Triyakbhyam Sutra is shown in following figure(4). The Vedic multiplication algorithm for 2 digit number is shown below

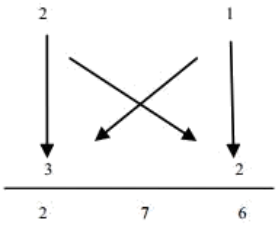


Figure 4: Multiplication of 21*32 using Urdhav Triyakbhyam Sutra

The multiplication of two 2-digit decimal numbers 21 and 32 is shown in figure (4). The least significant digit 1 of multiplicand is multiplied vertically by least significant digit 2 of the multiplier, so we get their product 2 and set it down as the least significant part of the answer. Then 2 and 2, 1 and 3 are multiplied crosswise, add the two, get 7 as the sum and set it down as the middle part of the answer, and then 2 and 3 is multiplied vertically, get 6 as their product and put it down as the last the left hand most part of the answer.

We get: 21 x 32 = 276

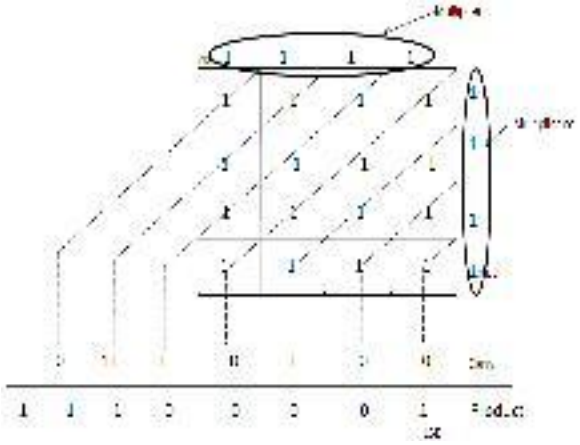


Figure 5: Multiplication of two 4-bit binary numbers by Urdhva tiryagbhyam sutra

Urdhva tiryagbhyam Sutra is used for two decimal number multiplication. This Sutra is used in binary multiplication as the 4-bit binary numbers to be multiplied and are written on two consecutive sides of the square as shown in the figure (5) and square is divided into rows and columns where as each row/column corresponds to one of the digit of either a multiplier or a multiplicand and hence, each bit of the multiplier has a small box common to a digit of the multiplicand. Each bit of the multiplier is then independently multiplied (logical AND) with every bit of the multiplicand and the product is written in the common box. All the bits lying on a crosswise dotted line are added to the previous carry so that least

II. OBJECTIVE :

The main objective of this thesis is to design and synthesis of an ALU with high speed, which can be used in any processor application. The ever increasing demand in enhancing the ability of processor to handle the complex and challenging processes has resulted in the integration of number of processor cores into one chip still the load on processor is not less and the load on processor increases. This load is reduced by supplementing the main coprocessor which is designed to work on specific types of functions like numeric computations, graphics, signal processing, etc. The speed of ALU depends on the multipliers. In algorithm and structure levels and numerous multiplication techniques have been developed to improve the efficiency of multiplier which focus on reducing the partial product and their additions and get a high efficiency of result. But in this case principle behind the multiplication remains same, so the use of Vedic mathematics for multiplication strikes difference in actual process of calculation.

ANCIENT VEDIC METHODS**:**

The Sanskrit word "Veda" means house of knowledge and this gift that the Indian gave to the world, thousand of year ago nearly in 1500 BC and this knowledge now currently employed in our global silicon chip technology of Engineering. The use of Vedic mathematics lies in the fact that it reduces the typical calculations was a time consuming method so develop a conventional mathematics to very simple ones. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works, so typical to typical calculation can be performed by normal person and hence Vedic mathematic provides techniques to solve operation with large magnitude number easily. Vedic Mathematics is a methodology of arithmetic rules that allow more efficient speed implementation and hence it is a very interesting field and presents some effective algorithms which can be applied to various branches of science and technology such as computing, quadratic equation, factorization and spherical geometry.

Urdhva Tiryakbhyam Sutra:

The given Vedic multiplier based on the Vedic multiplication formulae (Sutra). "Urdhva" and "Tiryakbhyam" words are derived from Sanskrit literature. This Sutra has been traditionally used for the multiplication of two numbers. Urdhav means "Vertically" and "Tiryakbhyam" means crosswise. In proposed work; we will apply the same ideas to make the proposed work compatible with the digital hardware. Urdhva Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It means "Vertically and crosswise". The digits on the two ends of the line are multiplied and the result is added with the previous carry and if there are more lines in one step, all the results are added to the previous carry. And hence the least significant digit of the number thus obtained acts as one of the result digits and the rest act as the carry for the next step. Initially carry will be taken will be taken to as zero. The visualize line diagram for multiplication of two 4-bit numbers is as shown in figure (6)

The application of "Urdhva Tiryakbhyam Sutra" using line diagram for the multiplication is shown in Figure

(6). The digits on the two ends of the line are multiplied and the result is added with the previous carry shown in above process. When there are more lines in one step, all the results are added to the previous carry and the least significant digit of the number thus obtained acts as one of the result digits and the rest act as the carry for the next step. Initially the carry is taken to be as zero thus extends this Sutra to binary number system.

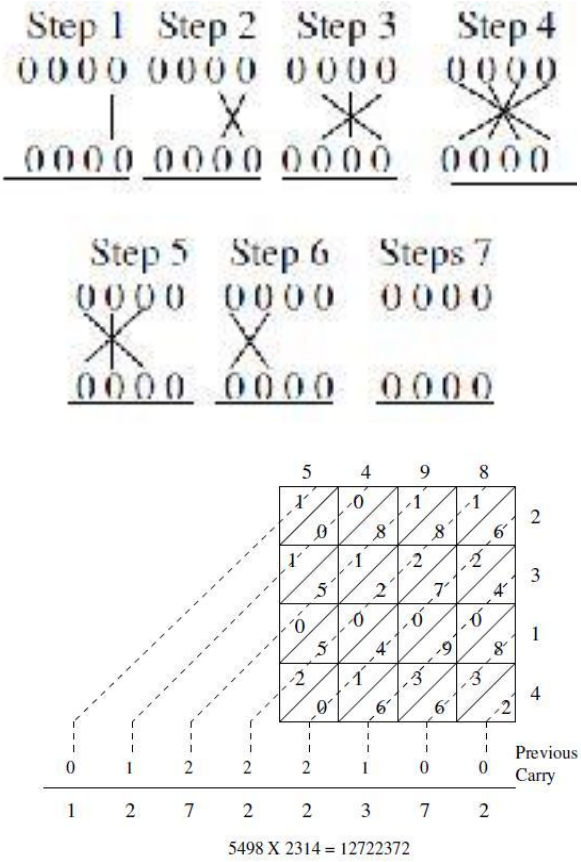


Figure 7: Alternative way of multiplication by Urdhva Tiryakbhyam Sutra.

For the purpose of multiplication algorithm, let us consider the multiplication of two 4-bit binary numbers

a3 a2 a1 a0 and b3 b2 b1 b0. As getting the result of this multiplication would be more than 4 bits of number we get and express it as … r4 r3 r2 r1 r0. As in the last case, the digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and carry. This carry will be added in the next step and hence the process goes on. If the condition present is, more than one line are there in one step, then all the results are added to the previous carry and in each step, least significant bit acts as the result bit and all the other bits act as carry. Let us take one simple example, if in some intermediate step, we will get 011, then 1 will act as result bit and 01 as the carry. Thus we will get the following expressions shown below:

r0 = a0b0; (1) c1r1= a1b0+a0b1; (2)
c2r2= c1+a2b0+a1b1 + a0b2; (3)
c3r3= c2+a3b0+a2b1 + a1b2 + a0b3; (4) c4r4= c3+a3b1+a2b2 + a1b3; (5)
c5r5= c4+a3b2+a2b3; (6) c6r6= c5+a3b3 (7)

C6r6r5r4r3r2r1r0 being the final product get. Hence this is the standard general mathematical formula which is applicable to all cases of multiplication purpose. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array. So for large number this is not an efficient algorithm for the multiplication of as a lot of propagation delay will be involved in such cases of mathematical calculation. To overcome this type of problem Nikhilam Sutra come in exist will present an efficient method of multiplying two large numbers.

VEDIC MULTIPLICATION ALGORITHM:

The Vedic Sutras:

Vedic mathematics is based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. These Sutras along with their brief meanings are enlisted below alphabetically.

1) (Anurupye) Shunyamanyat – If one is in ratio, the other is zero.
2) Chalana-Kalanabyham – Differences and Similarities
3) Ekadhikina Purvena – By one more than the previous one.
4) Ekanyunena Purvena – By one less than the previous one.
5) Gunakasamuchyah – The factors of the sum is equal to the sum of the factors.
6) Gunitasamuchyah – The product of the sum is equal to the sum of the product.
7) Nikhilam Navatashcaramam Dashatah – All from 9 and the last from 10.
8) Paraavartya Yojayet – Transpose and adjust.
9) Puranapuranabyham – By the completion or Non-completion
10) Sankalana-vyavakalanabhyam – By addition and by subtraction.
11) Shesanyankena Charamena – The remainders by the last digit.
12) Shunyam Saamyasamuccaye – When the sum is the Same that sum is zero.
13) Sopaantyadvayamantyam – The ultimate and twice the penultimate.
14) Urdhva-Tiryagbyham – Vertically and crosswise.
15) Vyashtisamanstih – Part and Whole.
16) Yaavadunam – Whatever the extent of its deficiency.

These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral), and applied various kind of mathematic calculation. As mentioned earlier, all these Sutras were renovated from ancient Vedic texts early in the last century by Sri Bharati Krsna Tirtha. Many Sub-sutras were also discovered at the same time which is not discussed here.

III. Quantitative Results

Following table shows the area and timing constraint of proposed Vedic Multiplier at different bit levels.

NBit Multiplier	Number of LUT uses as logic	Number of occupied Slices	Number of bounded IOBs	Maximum Combinational path delay(ns)
4-bit	28	16	16	11.443ns
8-bit	64	37	80	6.03ns

IV. Synthesis and Simulation

1.RTL schematic of proposed vedic multiplier:

IV. Result

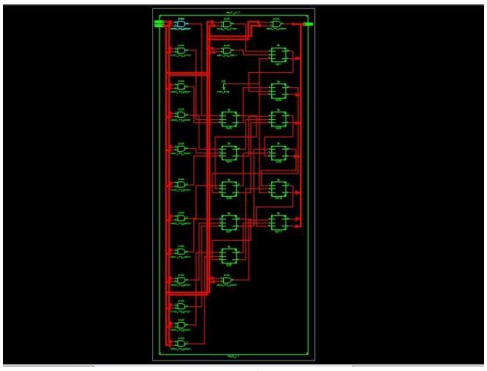
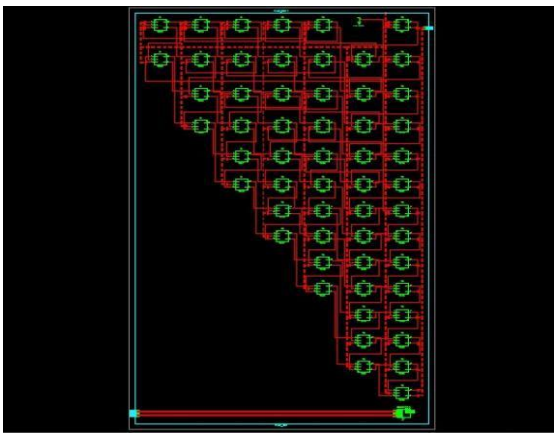
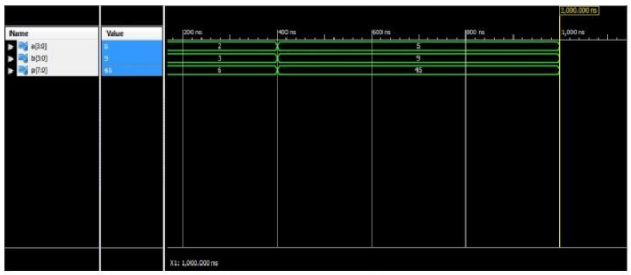
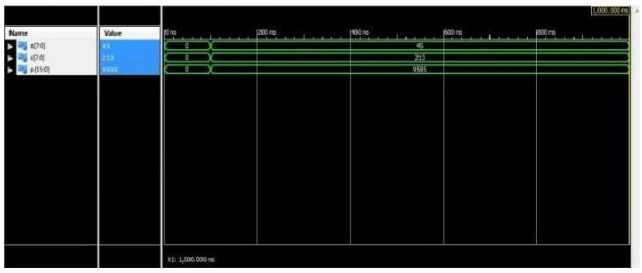


Figure 2: RTL schematic of proposed Vedic multiplier







We have proposed a new technique to deign Vedic multiplier using 4x4 bit and 8x8 bit Vedic multiplier designed in
VHDL (Very High Speed Integrated Circuit Hardware Description Language). It gives us method for hierarchical multiplier design and clearly specify a computational advantage offered by Vedic method. The computation path delay proposed 8x8 bit Vedic multiplier is found to be **6.03ns**. Logical Synthesis and Simulation was done in XilinxISE 13.1 Project Navigator and Simulated in Xilinx package. The performance of circuit is evaluated on the Xilinx device family Spartan3E.

The RTL schematic of 8x8 bit vedic multiplier comparises of 4x4 bit vedic multiplier as shown in figure

REFERENCES

[1] Vedic Maths Sutras - Magic Formulae [Online]. Available: http://hinduism.about.com/library/weekly/extra/blvedicmath utras.htm.

[2] S.J. Jou, C.Y.Chen, E.C. Yang, and C.C.Su(1997), "A pipelined multiplier-accumulator using a high speed, low power static and dynamic full adder design", IEEE journal of Solid-state circuits, vol.32, no.1, Jan.1997,pp.114-118.

[3] Xiaoping Hung, Wen-Jung Liu, and Belle W. Y. Wei. A High Performance CMOS Redundant Binary Multiplication and Accumulation (MAC) Unit, *IEEE* Transactions On Circuits and Systems, 41(1):33--39, January 1994

[4] P. R. Cappello, editor. VLSI Signal Processing. (IEEE Press, 1984).

[5] V.Vamshi Krishna, S. Naveen Kumar "High Speed,Power and Area efficient Algorithms for ALU using Vedic Mathematics"International Journal of Scientific and Research Publication,Volume 2, Issue 7 jully 2012.

[6] Mr. Abhishek Gupta, Mr. Amit Jain, Mr. Anand Vardhan Bhalla, Mr.Utsav Malviya " International Journal Of Engineering Reserch and Application (IJERA)issn:2248-9622 Volume.2 Issue5. September-October 2012"

[7] Pushpalata Verma, K.K.Mehta "Implementation of an Efficient Multiplier based on vedic Mathematics Using EDA Tool" International Journal of Engineering and Advanced Technology(IJEAT) ISSN:2249-8958, Volume-1, Issue5-June 2012.

[8] M. Ramalath, K.Deena Dayalan, S.Deborah Priya " High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques" IEEE Jully 15-17,2009.

[9] Abhishek Gupta "arithmetic Unit Implementation Using Delay Optimized Vedic Multiplier with BIST Capability" International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 5, May 2012.

[10] Shanthala S, S.Y. Kulkarni "VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique" European Journal of scientific Research Volume.30 No.4 (2009).