INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 1)
Available online at www.ijariit.com

High Speed Vedic Multiplier Design using FPGA

B. Shubhaker
shubhaker.subbu@gmail.com
St. Martin's Engineering College, Hyderabad, Telangana

E. Amareswar
subbu_ravi24@yahoo.com
MLR Institute of Technology, Hyderabad, Telangana

ABSTRACT

Vedic mathematics is an ancient system of mathematics, which was formulated by Sri Jagadguru Swami Bharati Krishna Tirthaji (1884 - 1960). After a research of eight years, he developed sixteen mathematical formulae from Atharvana Veda. The sutras (aphorisms) covered each and every topic of Mathematics such as Arithmetic, Algebra, Geometry and Trigonometry, Differential, Integral, etc. The word "Vedic" is derived from the word "Veda" which means the power house of all knowledge and divine. The proposed Vedic multiplier is based on the "Urdhava Triyagbhayam" sutra (algorithm). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. It literally means "Vertically and crosswise". Shift operation is not necessary because the partial product calculation will perform it in a single step, which in turn saves time. 16×16 Vedic multiplier is designed by using Urdhava Tiryakbhyam sutra and using rca adder .The 16 bit Vedic multiplier and array multiplier are designed by using Xilinx Spartan-3E FPGA.

Keywords: *FPGA Kit, DSP, ALU, Multipliers.*

1. INTRODUCTION

The word 'Vedic' was driven from the word 'Veda' which is ancient store-house of all knowledge. Vedic mathematics provides the solution to the problem of long computation time by reducing the time delay needed for the operations to be performed. It has originated from "Atharva Vedas" the fourth Veda. Atharva Veda mainly deals with the branches like engineering, mathematics, sculpture, medicines and all other sciences. Vedic mathematics deals with all areas of mathematics either it is pure or applied. It was rediscovered from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirtha Ji. Vedic mathematics has been formulated on sixteen sutras and thirteen sub-sutras. These sutras offer magical short cut methods toall basic mathematical operations. All the advantages drives from the fact that Vedic mathematics approach is totally different & considered very close to the way a human mind works. Vedic mathematics can be applied to every branch of mathematics including arithmetic, algebra and geometry. The powerful applications of Vedic mathematics are in fields of Digital Signal Processing (DSP), Chip Designing, Discrete Fourier Transform (DFT), High Speed Low Power VLSI Arithmetic and Algorithms and encryption systems. Arithmetic Logic unit (ALU) is the important unit in processors that performs basic arithmetic operations like addition, subtraction, multiplication and logical operations. Today's processors operate at very high clock speeds. Hence it is imperative to have faster additions, multiplications.

2. VLSI DESIGN

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device.

The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors. This is the field which involves packing more and more logic devices to smaller and smaller areas. VLSI circuits can now be put into a small space few millimeters across VLSI circuits are everywhere our computer, our car, our brand new state-of-the-art digital camera, the cell-phones.

VERY LARGE SCALE INTEGRATION

Final step in the development process, starting in the 1980s and continuing through the present, was in the early 1980s, and continues beyond several billion transistor mark transistors as of 2009. In 1986 the first one megabit RAM chips were introduced, which contained more than one million transistors. Microprocessor chips passed the million transistor mark in 1989 and the in 2005.The trend continues largely unabated, with chips introduced in 2007 containing tens of billions of memory transistors.

CHIP DESIGNING

During the desktop PC design era VLSI design efforts have focused primarily on optimizing speed to realise computationally intensive real-time functions such as video compression, gaming, graphics etc. As a result, we have semiconductor ICs that successfully integrated various complex signal processing modules and graphical processing units to meet our computation and entertainment demands. While these solutions have addressed the real-time problem, they have not addressed the increasing demand for portable operation, where mobile phone needs to pack all this without consuming much power. The strict limitation on power dissipation in portable electronics applications such as smart phones and tablet computers must be met by the VLSI chip designer while still meeting the computational requirements. While wireless devices are rapidly making their way to the consumer electronics market, a key design constraint for portable operation namely the total power consumption of the device must be addressed. Reducing the total power consumption in such systems is important since it is desirable to maximize the run time with minimum requirements on size, battery life and weight allocated to batteries. So the most important factor to consider while designing SoC for portable devices is 'low power design'.

Scaling of technology node increases power-density more than expected. CMOS technology beyond 65nm node represents a real challenge for any sort of voltage and frequency scaling Starting from 120nm node, each new process has inherently higher dynamic and leakage current density with minimal improvement in speed. Between 90nm to 65nm the dynamic power dissipation is almost same whereas there is ~5% higher leakage/mm2.Low cost always continues to drive higher levels of integration, whereas low cost technological breakthroughs to keep power under control are getting very scarce. Modern System-on-Chip demand for more power. In both logic and memory, Static power is growing really fast and Dynamic power kind of grows. Overall power is dramatically increasing. If the semiconductor integration continues to follow Moore's Law, the power density inside the chips will reach far higher than the rocket nozzle.

VLSI DESIGN PROCESS

VLSI technology thus provides a platform for developing systems for various applications. The integrated circuits so developed can be further classified as shown in below figure.

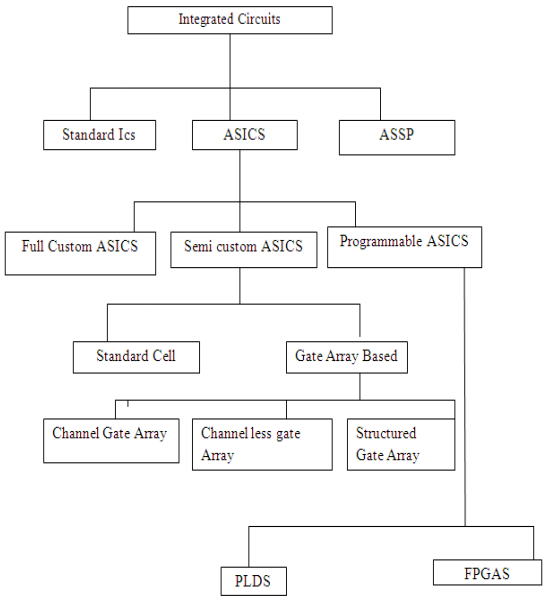


Fig : Classification of IC Design

Today VLSI CMOS technologies deliver individual integrated circuits (ICs) and containing millions of gates, sufficient to implement substantial systems-on-a chip or major subsystems-on-a chip. System-on-chip design may involve the expertise from many fields of electronics such as signal processing, communication, device physics etc. and so on. It is unreasonable to expect the architect of a speech recognition system, for example, to be an expert in device physics as well as in signal processing. The Mead-Conway methodology for integrated-circuit design makes VLSI technology available to such an application designers.

DESIGN DESCRIPTION

VLSI design style mainly uses three domains of design description, viz. the behavioral, the description of the function of the design; the structural, the description of the form of the implementation; and the physical, and the description of the physical implementation of the design. There are many possible representations of a circuit in each description, and judicious choice of representations is important in tool designee. The VLSI designee style is shown in below figure.

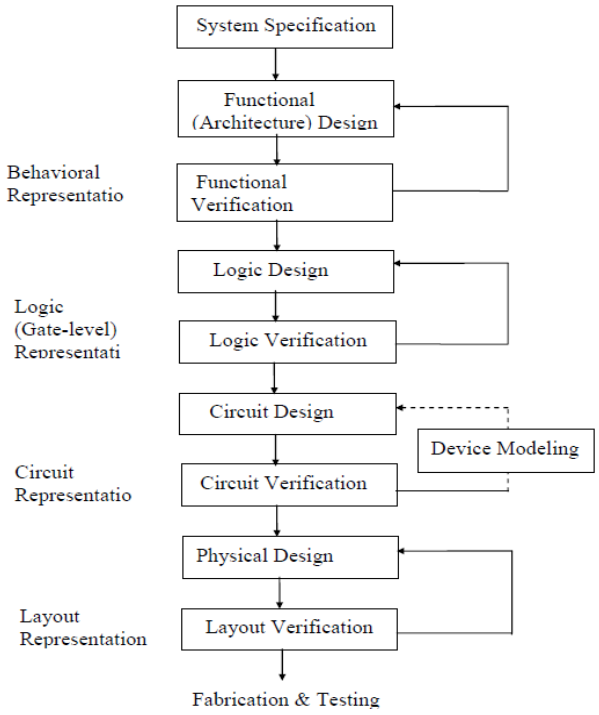


Fig : VLSI Design Style

At the beginning of a design it is important to specify the requirements without unduly restricting the design. The object is to describe the purpose of the design including all aspects, such as the functions to the realised, timing constraints, and power dissipation requirements, etc. Descriptions in block level may show either data flow, control flow, or both. The individual blocks generally correspond to hardware modules.

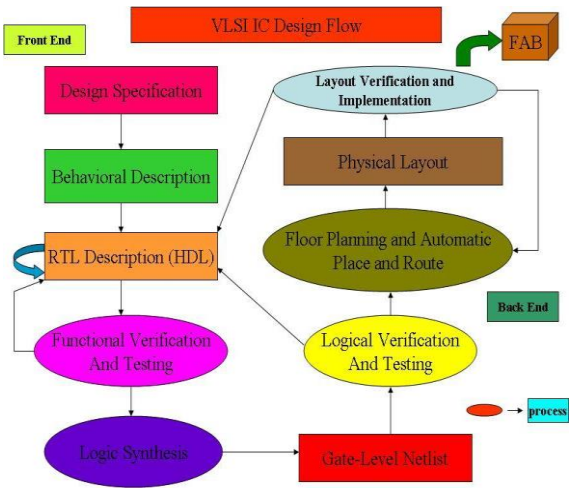


Fig: ASIC Design Flow

The AISE® design flow comprises the following steps: design entry, design synthesis, design implementation, and Xilinx® device programming. Design verification, which includes both functional verification and timing verification, takes places at different points during the design flow See fig 2.5.This section describes what to do during each step. For additional details on each design step, click on a link below the following figure 2.6.

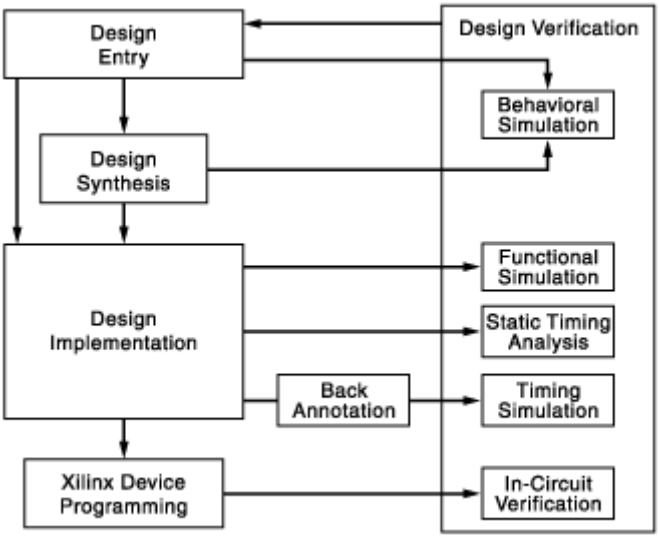


Fig: FPGA Design Flow

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). Contemporary FPGAs have large resources of logic gates and RAM blocks to implement complex digital computations. As FPGA designs employ very fast I/Os and bidirectional data buses it becomes a challenge to verify correct timing of valid data within setup time and hold time. Floor planning enables resources allocation within FPGA to meet these time constraints. FPGAs can be used to implement any logical function that an ASIC could perform.

3. MULTIPLIERS

Multiplication is one of the most Complex Operations within arithmetic processors such as the ALU. Hence it is one of the most complex primitive to be designed in the configurable chip. The selection criteria for various design options. Two Architectures are Configurable serial/parallel Multiplier and Configurable Serial-Parallel Multiplier

Serial multipliers also find applications in system-on-chip (SoC) design. As technology scales, more intellectual property cores and logic blocks will be integrated in a SoC, resulting in larger interconnect area and higher power dissipation. The increase in integration density of the on-chip modules causes the buses connecting these modules to become highly congested. To overcome this problem, new techniques have been evolved recently to have on-chip data transfer in a high speed serial link instead of conventional bus. Figure1 (a) and (b) depict the conventional on-chip bus and alternative on-chip serial-link bus structures, respectively. In Figure1(b), the serializer at the source module converts the parallel outputs to a bit stream that can be transferred in a simple routing network and at the destination module they are converted back to parallel data by the deserializer.

SERIAL-SERIAL MULTIPLIER

The on-chip serial-link is capable of transmitting data at Gb/s so that a chunk of parallel data is available when the destination module finishes the previous computation Under the new on-chip communication paradigm for digital signal processing, it is desirable to have a low complexity data processing unit as the destination module that is able to perform partial computation on the incoming data stream at high speed while the data is being buffered. Figure.2.2 illustrates a potential use of a serial-serial multiplier as a destination module in a SoC with serial-link bus architecture. The low complexity pre computation unit forms part of the serial-serial multiplier and could perform partial computation on the high speed serial bit stream.

The unit doubles as a buffer and eliminates the deserializer. As the data has been partially processed and buffered, the completion of the multiplication can be done at a lower speed with a less complex parallel multiplier.

The challenge in such a scheme lies in reducing the critical path delay of the pre computation unit to that of the deserializer, which usually has bit rate in the order of several Gb/s.
We introduce this new scheme for the design of serial-serial multiplier suitable for SoCs with on-chip serial-link bus architecture. The proposed scheme could also be used as an alternative to embedded multipliers in the future field-programmable gate array (FPGA), where configurable logic blocks (CLBs), embedded multipliers and memory blocks are integrated with serializer / deserializer to facilitate on-chip serial data transfer in order to reduce interconnect complexity.

A serial accumulator developed based on the new design paradigm is proposed to deal with very high-speed data sampling rate of above 4 GHz. The accumulator employs asynchronous counters1 to perform bit accumulation at each bit position of the PP matrix, resulting in low critical path delay and small area, especially for operands with long word length. Asynchronous counter has a low hardware complexity but the outputs are not synchronized with the clock which leads to a timing delay before all output bits of the counter have settled to their final states. The correct output of the counter is read after a timing delay to be analyzed from the timing diagram in Section VI-B. The data dependent counters change states only when the input bit is "1," which leads to low switching power dissipation. The height of the PP matrix after buffering by the asynchronous counters is reduced logarithmically to $[\log_2 n] +1$ before it is further reduced by the CSA tree.

PARALLEL/PARALLEL MULTIPLIER

In serial/parallel multiplier algorithm is one design "serial" components points to reduce silicon chip area. Two unsigned fixed point numbers represented by m, n bits can be:

$$a(m) = a_{m-1} \ldots\ldots a_0$$
$$b(n) = b_{n-1} \ldots\ldots b_0$$

The double word length product Q (m+n) is

$$Q(m+n) = \sum \sum a^i b^i 2^{i+j}$$

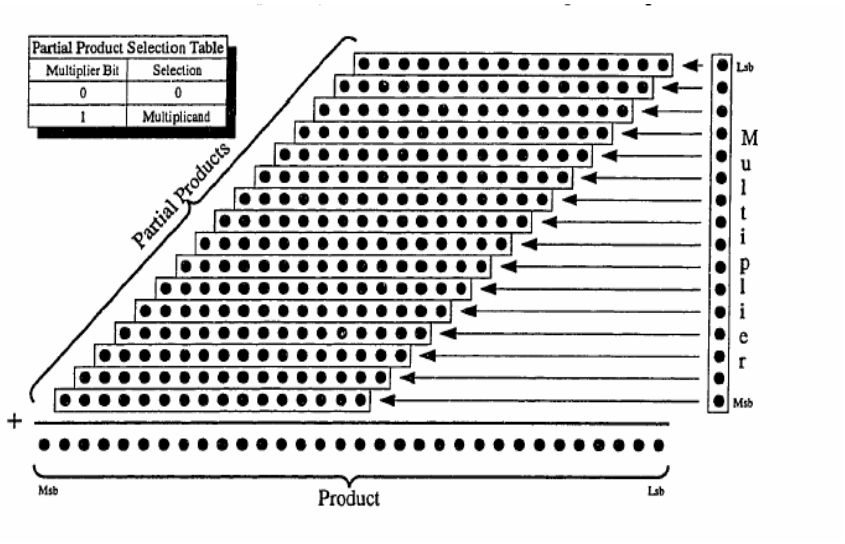
Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets–high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation. The common multiplication method is "add and shift "algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages. Further by combining both modified Booth algorithm and Wallace Tree technique we can see advantage of both algorithms in one multiplier. However with increasing parallelism, the amount of shifts between the partial products and intermediate sums to be added will increase which may result in reduced speed, increase in silicon area due to irregularity of structure and also increased power consumption due to increase in interconnect resulting from complex routing. On the other hand "serial-parallel "multipliers compromise speed to achieve better performance for area and power consumption. The selection of a parallel or serial multiplier actually depends on the nature of application. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.

DIFFERENT MULTIPLIERS

As we know in multiplication operation there are two operands, one is multiplicand and other is multiplier. In binary number system we do multiplication by using different type of multiplier. A binary multiplier uses the simple shift and adds operation. There are many multipliers introduced in digital electronics. Some of them are as follows:

ARRAY MULTIPLIER

An array multiplier is shown in below Fig 3.9 is a parallel multiplier which does shift and adds all at once. This multiplier is called an array because it has array of adders. An array multiplier also uses shift and adds operation as in binary multiplier but it adds the partial products parallel. The following figure shows the 4x4 array multiplier. Digital Multiplication entails a sequence of additions carried out on partial products the method by which this partial product array is summed to give the final product is the key distinguishing factor amongst the numerous multiplication schemes.



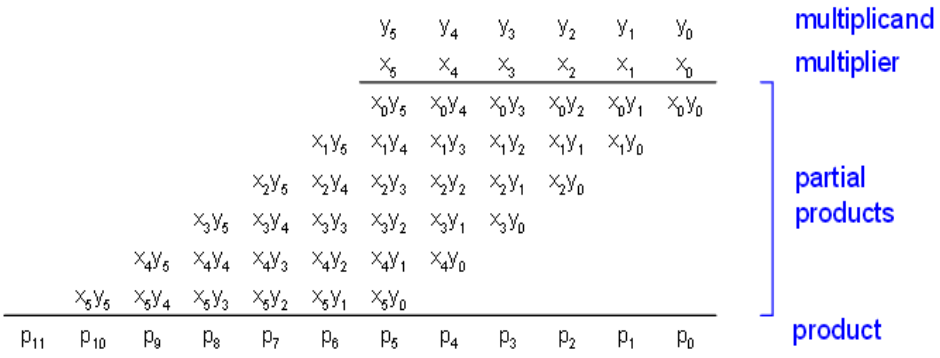


Fig: Array Multiplication

Advantages

- Minimum Complexity
- Easy Scalable
- Easy Pipelined
- Regular Shape &Easy Place & Route

Disadvantages

- High Power Consumption
- More Digital Gates &Large Chip Area

Booth Multiplication

Booth Multiplication Algorithm

Booth algorithm gives a procedure for multiplying binary integers in signed –2's complement representation.

Wallace Tree Multiplier

Several popular and well-known schemes, with the objective of improving the speed of the parallel multiplier, have been developed in past. Wallace introduced a very important iterative realization of parallel multiplier. This advantage becomes more pronounced for multipliers of bigger than 16 bits.

In Wallace tree architecture, all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. Another set of counters then reduces this new matrix and so on, until a two-row matrix is generated. The most common counter used is the 3:2 counter which is a Full Adder. The final results are added using usually carry propagate adder. The advantage of Wallace tree is speed because the addition of partial products is now O (logN). A block diagram of 4 bit Wallace Tree multiplier is shown in below. As seen from the block diagram partial products are added in Wallace tree block. The result of these additions is the final product bits and sum and carry bits which are added in the final fast adder (CRA).

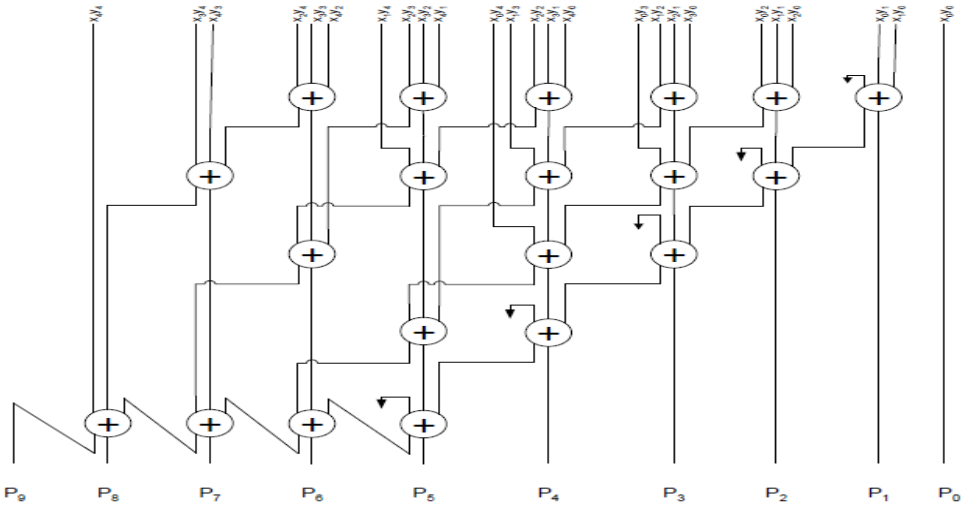


Fig: Wallace tree Architecture

Since Wallace Tree is a summation method, it can be used in conjunction with array multiplier of any kind including Booth array. The diagram below shows the implementation of 8 bit squarer using the Wallace tree for compressing the addition process. Above fig 3.10 is a Wallace tree Architecture.

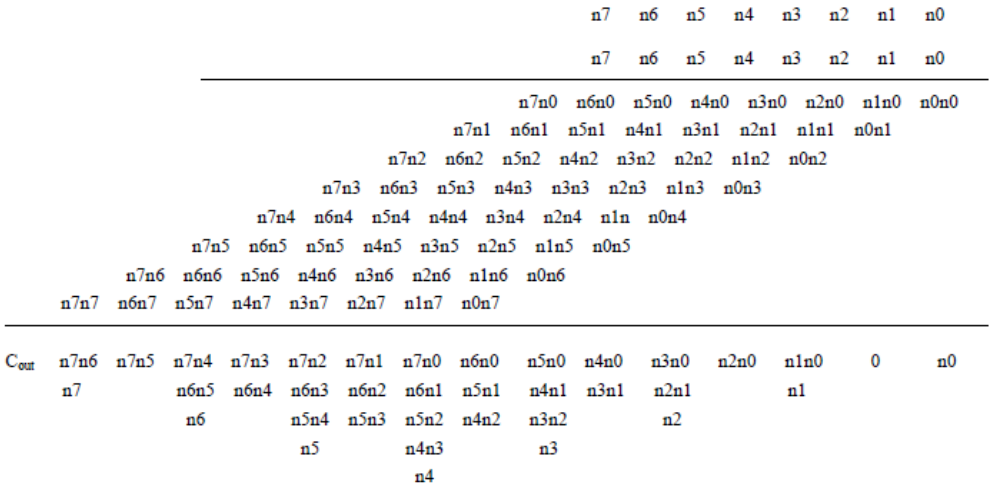


Fig: Operation of 8 bit square

Advantages

- Each layer of the tree reduces the number of vectors by a factor of 3:2
- Minimum propagation delay.
- The benefit of the Wallace tree is that there are only O (log *n*) reduction layers, but adding partial products with regular adders would require O (log *n*) ^{2} times.

Disadvantages

- Wallace trees do not provide any advantage over ripple adder trees in many FPGAs.
- Due to the irregular routing, they may actually be slower and are certainly more difficult to route.
- Adder structure increases for increased bit multiplication.

4. DESIGN OF HIGH PERFORMANCE VEDIC MULTIPLIER

Vedic Multiplier Architecture

This section introduces multiplication operation using *Vedic IXI Methodology* and then illustrates architecture of 2x2 multiplier modules and finally architecture of multiplier. *Urdhava-Tiryakbhayam* sutra has been used for multiplication purpose. The fig. 1 explains multiplication of two decimal numbers using IXI technique:

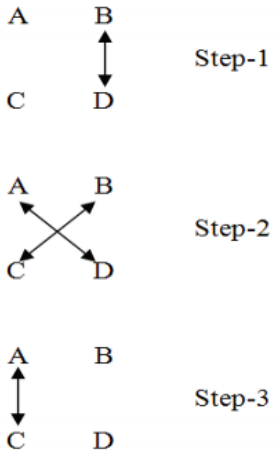


Fig: IXI Methodology for Multiplication

In this method initially multiplication of the rightmost digit of multiplier is performed with the rightmost digit of the multiplicand giving the LSB of the product term as shown in step-1 of fig. 1. Then multiplication of the leftmost digit of the multiplicand with the rightmost digit of the multiplier and the rightmost digit of the multiplicand with the leftmost digit of multiplier is performed and then added. Thus forming the middle part of the product term as in step-2 of fig.1. At the last step, in step-3 the leftmost part of the multiplicand is multiplied with the leftmost part of the multiplier forming the leftmost part of the product term. In this way the multiplication process is carried out. Similar logic of cross multiplication and addition can be extended to implement any number of bits. Each iteration gives the coefficient of the final product.

16-bit Vedic Multiplier

This architecture consists of four 8 bit multipliers used for calculating the partial products. Next, the results of these 8 bit multipliers are adjusted using concatenation operation to have all the partial product terms of equal bit-length. The partial product of right most multiplier is concatenated with the partial product of leftmost multiplier and the partial products of middle two multipliers are concatenated with 8 zeroes each. All the numbers obtained after concatenations are added together using the single carry save adder. At the end, the sum obtained from the ripple carry adder is concatenated with the LSB partial product of right most multiplier to get the required final product.

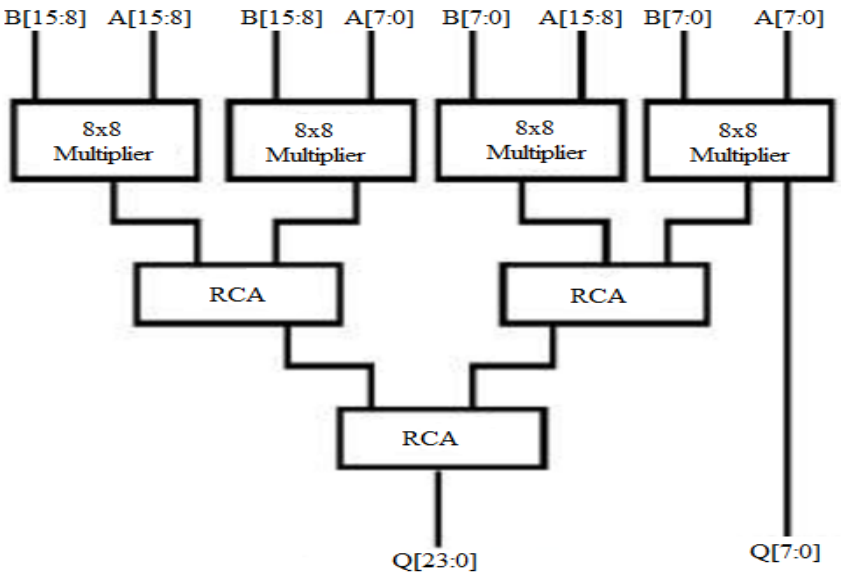


Fig: Architecture of 16-bit Multiplier

Array Multiplier

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array. In array multiplier, consider two binary numbers A and B, of m and n bits.

There are a summands that are produced in parallel by a set of an AND gates. n x n multiplier requires n (n-2) full adders, n half-adders and n2 AND gates. Also, in array multiplier worst case delay would be (2n+1) td.

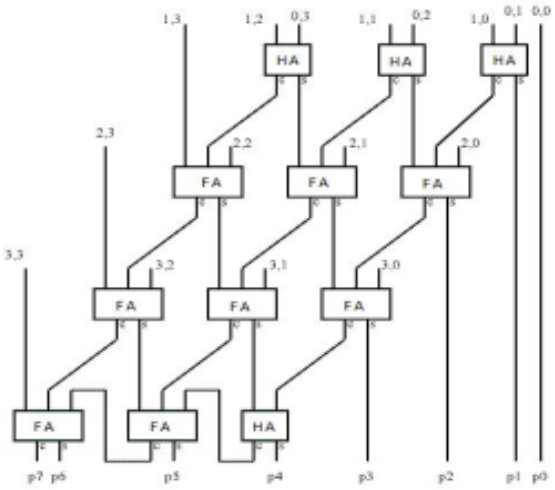


Figure.1.Array Multiplier

The single core processor is heart and brain of a computer, enhancing the ability of single core to handle the complex and challenging processes has led to design of MP (Multi Core Processor). The load on the processor is reduced by supplementing the main processor with Co-Processors. Coprocessors design is application specific, they are used as add on with the main processor in field of applications like Signal Processing, Image Processing, Servers, and Graphics etc. Most DSP tasks require real-time processing; it must perform these tasks speedily while minimizing Cost and Power. Although multiplication techniques such as 'Booth Algorithm Multiplier' have been effective over conventional 'Array Multiplication' technique, their disadvantage of time consumption has not been completely removed. The multiplication algorithms differ in the means of 'partial product generation' and 'partial product addition. The array multiplier has linear time complexity i.e. O(n) therefore delay degrades the performance for multipliers having larger operand sizes. Also it has poor space complexity O(n2), as it requires approximately n2 cells to produce multiplication. Therefore as the operand size grows, the circuit takes larger area and power. Vedic Mathematics shows its application in fast calculations, trigonometry, three dimensional coordinate geometry, solution of plane and spherical triangles, linear and non-linear differential equations, matrices and determinants, log and exponential. Vedic Mathematics provides unique solutions in several instances where trial and error method is available at present. In this paper we present implementation of 16×16, multiplier design using Array of Array technique. The 16 bit array multiplier is implemented using four 4-bit array multipliers.

Comparison

Delay Comparison between Vedic Multiplier and Array Multiplier

Multiplier Type	Vedic Multiplier	Array Multiplier
Delay(ns)	27	47.72

5. FIELD PROGRAMMABLE GATE ARRAY

INTRODUCTION OF FPGA (FIELD PROGRAMMABLE GATE ARRAY)

Field-programmable gate array (FPGA) is a semiconductor device that can be configured by the customer or designer after manufacturing—hence the name "field-programmable". To program an FPGA you specify how you want the chip to work with a logic circuit diagram or a source code in a hardware description language (HDL). FPGAs can be used to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications.

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like a one-chip programmable breadboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.
For any given semiconductor process, FPGAs are usually slower than their fixed ASIC counterparts. They also draw more power, and generally achieve less functionality using a given amount of circuit complexity.

But their advantages include a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. Vendors can also take a middle road by developing their hardware on ordinary FPGAs, but manufacture their final version so it can no longer be modified after the design has been committed.

POWER DISSIPATION IN FPGAS

The programmability and flexibility of FPGAs make them less power-efficient than custom ASICs for implementing a given logic circuit. The FPGA configuration circuitry and configuration memory consume silicon area, producing longer wire lengths and higher interconnect capacitance. Programmable routing switches attach to the pre-fabricated metal wire segments in the FPGA interconnect and add to the capacitive load incurred by signals. To be sure, most dynamic power in an FPGA is consumed in the programmable routing fabric. Likewise, static power is proportional to total transistor width. The interconnect comprises a considerable fraction of an FPGA's transistors and therefore dominates leakage. Consequently, the interconnection fabric should be a prime target for FPGA power optimization.

Certainly, power can be addressed through process technology changes, at the hardware architecture level, or at the circuit level. Virtex-5 FPGAs, for example, contain .diagonal. Interconnect resources; allowing connections to be made with fewer routing conductors and reduced interconnect capacitance. At the transistor level, Virtex-4 and Virtex-5 FPGAs employ triple-oxide process technology for leakage mitigation. Three possible oxide thicknesses are available for each transistor, depending on its speed, power, and reliability requirements. The proliferation and increased use of hard IP blocks, such as DSPs and processors, can also lower power consumption versus implementing such functionality in the standard FPGA fabric. An overview of approaches for optimizing FPGA power can be found.

It is also possible to reduce power without incurring any costly hardware or process changes. Power issues can be tackled through power-driven CAD algorithms and design flows.

6. ADVANTAGES, DIS-ADVANTAGES AND APPLICATIONS

ADVANTAGES

- Since the partial products and their sums are calculated in parallel, the multiplier is independent of clock frequency of processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of clock frequency.
- Vedic multiplier has less number of gates required for given 8 × 8, 16× 16... Multipliers hence power dissipation is very small hence low power consumption i, e., power efficient.
- Vedic multiplier has greatest advantage as compared to other multipliers over gate delays and regularity of structures.
- Highest speed among conventional multiplier.
- It has higher throughput operations.

DIS-ADVANTAGES

- Due to Urdhva tiryakbhyam structure, the system suffers from a carry propagation delay in case of large numbers.
- As the number of bits increases above 32 or 64 bits the propagation delay in calculating RHS part of the algorithm also increases significantly.

7. APPLICATIONS

- Because of high speed of Vedic multiplication ALU utilizes this algorithm to give reliable output.
- Vedic multiplier is a low power VLSI system design.

8. RESULT

A 16x16 Vedic multiplier using Urdhva–Tiryakbhyam Sutra in binary are implemented using VHDL language. The entire code is completely synthesizable. The synthesis is done using Xilinx Synthesis Tool (XST) available with Xilinx ISE14.7.Implementation of an Array multiplier is also done and both the multipliers are compared to know the speed and delay. The hardware output of a 16X16 Vedic multiplier is dumped in to a FPGA (Field Programmable Gate Array) and the output is shown.

9. CONCLUSION

The 16 bit Vedic multiplier and array multiplier proved to be efficient in terms of speed. These architectures have regular structures. So, they can be realized easily on the silicon chip. The array multiplier is more efficient then Vedic multiplier. The reduced number of computations in multiplication due to adjusting using concatenation operation and one carry save adder only. The 16 bit Vedic multiplier and array multiplier are designed by using Xilinx Spartan-3E FPGA.

10. FUTURE SCOPE

The 16 bit vedic multiplier is done using Urdhva-Tiryakbhyam Sutra. This project can be further developed i.e. the speed and power of the device can be increased, delay can be decreased by implementing or designing the multiplier by using other sutras such as Nikhilam and Anurupye sutras are such vedic sutras which can reduce the delay, power and hardware requirements for multiplication of numbers.

11. REFERENCES

[1] Jagadguru Swami Sri Bharati Krishna Tirtha Ji Maharaja, "Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.
[2] Poornima M, Shivaraj Kumar Patil, Shivukumar, Shridhar K P, Sanjay H, "Implementation of Multiplier using Vedic Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 2, Issue-6, May 2013.
[3] G. Vaithiyanathan, K.Venkatesan, S,Sivaramakrishnan, S.Siva, S.Jayakumar, "Simulation And Implementation of Vedic multiplier using VHDL code", International Journal of Scientific &Engineering Research, Vol. 4, Issue 1, January 2013.
[4] R.K.Bathija, R.S.Meena, S.Sarkar, Rajesh Sahu, "Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics", International Journal of Computer Applications, Vol. 59, No. 6, December 2012.
[5] Ramachandran.S, Kirti.S.Pande, "Design, Implementation and Performance Analysis of an Integrated Vedic multiplier Architecture", International Journal of Computational Engineering Research (IJCER), Vol. 2, Issue No. 3, pp. 697-703, May-June 2012.
[6] G.Ganesh Kumar, V.Charishma, "Design of High Speed Vedic Multiplier using Vedic Mathematics Techniques", International Journal of Scientific and Research Publications, Vol. 2, Issue 3, March 2012.
[7] Kabiraj Sethi, Rutuparna Panda, "An Improved Squaring Circuit for Binary Numbers", International Journal of Advance Computer Science and Applications, Vol. 3, No. 2, pp. 111-116, 2012.
[8] Dilip Kumar, Abhijeet Kumar, Siddhi, "Hardware Implementation of 16 * 16 bit Multiplier and Square using Vedic Mathematics", International Conference on Signal, Image and Video Processing (ICSIVP), 2012.
[9] Ramalatha M, Thanushkodi K, Deena Dayalan K, Dharani P, "A Novel Time and Energy Efficient Cubing Circuit using Vedic Mathematics for Finite Field Arithmetic", International Conference on Advances in Recent Technologies in Communication and Computing, pp.873-875, 2009.
[10] Himanshu Thapliyal, Saurabh Kotiyal, M.B Srinivas, "Design and analysis of a novel parallel square and cube architecture based on ancient Indian Vedic Mathematics", Circuits and Systems, 48th Midwest Symposium, pp. 1462, Vol. 2, 7-10 Aug. 2005.
[11] Jagadguru Swami, Sri Bharati Krsna Tirthji Maharaja," Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.
[12] A. Karatsuba, Yu. Ofman (1962). "Multiplication of Many-Digital Numbers by Automatic Computers". Proceedings of the USSR Academy of Sciences 145: 293–294.
[13]Mrs. M. Ramalatha, Prof. D. Sridharan, "VLSI Based High Speed Karatsuba Multiplier for Cryptographic Applications Using Vedic Mathematics", IJSCI, 2007
[14] Thapliyal H. and Srinivas M.B. "High Speed Efficient N x N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Engineering, Computing and Technology, 2004, Vol.2.
[15] Prashant Nair, Darshan Paranji, S. S. Rathod," VLSI Implementation of Matrix-Diagonal Method of Binary Multiplication". Proceedings of SPIT-IEEE, Vol. 2, 55, 2007
[16] M. Ramalatha, K. Deena Dayalan, P. Dharani, S. Deborah Priya," High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques ", ACTEA 2009
[17] Abhijit Asati and Chandrashekhar "A High-Speed, Hierarchical 16×16 Array of Array Multiplier Design", IMPACT 2009.
[18] Kevin Biswas, "Multiplexer Based Array Multipliers," A Ph.D. Dissertation, University of Windsor, Electrical and Computer Engineering, Apr. 2005.

