VLSI Implementation &analysis of area and speed in QSD and Vedic ALU

Bhawna Bishnoi[1] , Ghanshyam Jangid[2]

[1]Department of Electronics and communication , Gyan Vihar University, Jaipur, Raj
[2]Assistant Professor, Department of Electronics and Communication ,Gyan Vihar University , Jaipur, Raj

***Abstract*—**Arithmetic operations in digital signal processing applications suffer from problems including propagation delay and circuit complexity which may occupy larger area. We have two high performance methods among all ALU circuitry. First one is QSD and another one is VEDIC methodology. In QSD number representation allows a method of fast addition/subtraction because the carry propagation chains are eliminated and hence it reduces the propagation time in comparison with common radix 2 system. Here we propose an arithmetic unit based on QSD number system based on quaternary system. The proposed design is developed using VHDL and implemented on FPGA device and results are compared with conventional Vedic arithmetic unit. The implementation of quaternary addition and multiplication results in a fix delay independent of the number of digits. Operations on a small number of digits such as 8, 4, or more, can be implemented with constant delay and less complexity.

Digital signal processing (DSP) is the technology that is omnipresent in almost every engineering discipline. A typical processor devotes a considerable amount of processing time in performing arithmetic operations, particularly multiplication operations. Multiplication is one of the basic arithmetic operations and it requires substantially more hardware resources and processing time than addition and subtraction. In fact, 8.72% of all the instruction in typical processing units is multiplication. The core computing process is always a multiplication routine; therefore, DSP engineers are constantly looking for new algorithms and hardware to implement them. Vedic mathematics is the name given to the ancient system of mathematics, which was rediscovered, from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirthaji. The whole of Vedic mathematics is based on 16 sutras (word formulae) and manifests a unified structure of mathematics.

Keywords— QSD method, Vedic maths, VHDL, Delay, logical operation.

I. INTRODUCTION

Arithmetic operations are widely used and play important role in various digital systems such as computers and signal processors. Designing this Arithmetic unit using QSD number representation has attracted the interest of many researchers. Additionally, recent advances in technologies for integrated circuits make large scale arithmetic circuits suitable for VLSI implementation. However, arithmetic operations still suffer from known problems including limited number of bits, propagation time delay, and circuit complexity. This paper implements a high speed QSD arithmetic logic unit which is capable of carry free addition, borrow free subtraction, up-down count and multiply operations. The QSD addition/subtraction operation employs a fixed number of min terms for any operand size. The designed high speed Multiplier composed of partial product generators and adders which is proposed in [5] is implemented. This paper also proposes a new algorithm for converting binary to QSD and QSD back to binary.

In any processor the major units are Control Unit, ALU and Memory read write. Among these units the performance of any processor majorly depends on the time taken by the ALU to perform the specified operation. Multiplication is an important fundamental function in arithmetic operations. Multiplication based operations such as Multiply and Accumulate (MAC) and inner product are among some of the frequently used Computation Intensive Arithmetic Functions (CIAF) currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. The demand for high speed processing has been increasing as a result of expanding computer and signal processing

applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing application. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of

interest over decades. Reducing the time delay and power consumption are very essential requirements for many applications. This work presents different multiplier architectures. Multiplier based on Vedic Mathematics is one of the fast and low power multiplier. Digital multipliers are the most commonly used components in any digital circuit design. They are fast, reliable and efficient components that are utilized to implement any operation. Depending upon the arrangement of the components, there are different types of multipliers available. Particular multiplier architecture is chosen based on the application. In many DSP algorithms, the multiplier lies in the critical delay path and ultimately determines the performance of algorithm. The speed of multiplication operation is of great importance in DSP as well as in general processor. In the past multiplication was implemented generally with a sequence of addition, subtraction and shift operations. There have been many algorithms proposals in literature to perform multiplication, each offering different advantages and having tradeoff in terms of speed, circuit complexity, area and power consumption. The multiplier is a fairly large block of a computing system. The amount of circuitry involved is directly proportional to the square of its resolution i.e. A multiplier of size n bits has n2 gates. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective. Latency is the real delay of computing a function, a measure of how long the inputs to a device are stable is the final result available on outputs. Multiplier is not only a high delay block but also a major source of power dissipation. That's why if one also aims to minimize power consumption, it is of great interest to reduce the delay by using various delay optimizations. In this thesis work, Urdhva tiryakbhyam Sutra is first applied to the binary number system and is used to develop digital multiplier architecture. This is shown to be very similar to the popular array multiplier architecture. This Sutra also shows the effectiveness of to reduce the NxN multiplier structure into an efficient 2x2 multiplier structures. Nikhilam Sutra is then discussed and is shown to be much more efficient in the multiplication of large numbers as it reduces the multiplication of two large numbers to that of two smaller ones. The proposed multiplication algorithm is then illustrated to show its computational efficiency by taking an example of reducing a 4x4-bit multiplication to a single 2x2-bit multiplication operation. This work presents a systematic design methodology for fast and area efficient digit multiplier based on Vedic mathematics. The Multiplier Architecture is based on the Vertical and Crosswise algorithm of ancient Indian Vedic The functional verification of the QSD & Vedic design is done using Xilinx project navigator and Modelsim simulator. The behavioral model of the design is done using VHDL – VHSIC Hardware description language.

QUATERNARY SIGNED DIGIT NUMBER

Quaternary digit {0,1,2} represented by 2- bit equivalent in binary whereas QSD represented are by 3-bit 2's compliment notation.

Each number can be represented by-

$$D = \sum_{i}^{n} x_i 4^i,$$

where xi can be any value of set {3,2,1,0,1,2,3} for producing appropriate decimal representation. A QSD negative value is the QSD complement of the QSD positive value i.e. 3=-3, 2=-2, 1=-1.

It offer the advantage of reduced circuit complexity both in the term of transistor count and interconnection.QSD number uses 25% less space than BSD number and it can be verified by theorem described as under QSD number save 25% storage compared to BSD: To represent a numeric value Nlog4N number of QSD and 3log2N binary bits are required whereas for the samelog2N BSD digits and 2log2N binary bits are required in BSD representation. Therefore, QSD saves ¼ of the storage used by BSD. So the proposed of the QSD adder is better than RBSD adder in term of the number of gates , input connection and delay though both perform addition within constant time. Proposed design has the advantages of the both parallelism as well as reduced gates complexity needed.

II. DESIGN ALGORITHM OF QSD ADDER

In QSD number system carry propagation chain are eliminated which reduce the computation time substantially, thus enhancing the speed of the machine [31]. As range of QSD number is from -3 to 3, the addition result of two QSD numbers varies from -6 to +6 [30]. Table I depicts the output for all possible combinations of two numbers. The decimal numbers in the range of -3 to +3 are represented by one digit QSD number. As the decimal number exceeds from lower digit position to higher digit position QSD number representation is used [37]. QSD numbers allow redundancy in the number representations. The same decimal number can be represented

in more than one QSD representations. So we choose such QSD represented number which prevents further rippling of carry. To perform carry free addition, the addition of two QSD numbers can be done in two steps [4]:from this range, more than one digit of QSD number is required. For the addition result, which is in the range of -6 to +6, two QSD. In the two digit, QSD result the LSB digit the sum bit and MSB digit represent the carry bit. To prevent this carry bit to propagate: digits are needed. In the two digits QSD result the LSB digit represents the sum bit and the MSB digit represents the carry bit. To prevent this carry bit to propagate

Step 1: First step generates an intermediate carry and Intermediate sum from the input QSD digits i.e., addend and augend.

Step 2: Second step combines intermediate sum of current digit with the intermediate carry of the lower significant digit.

Table1.The intermediate carry and sum between-6to+6

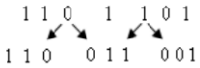
Sum	*QSD represented number*	*QSD coded number*
-6	$\bar{2}2, \bar{1}\bar{2}$	$\bar{1}\bar{2}$
-5	$\bar{2}3, \bar{1}\bar{1}$	$\bar{1}\bar{1}$
-4	$\bar{1}0$	$\bar{1}0$
-3	$\bar{1}1, 0\bar{3}$	$\bar{1}1$
-2	$\bar{1}2, 0\bar{2}$	$0\bar{2}$
-1	$\bar{1}3, 0\bar{1}$	$0\bar{1}$
0	00	00
1	$01, 1\bar{3}$	01
2	$02, 1\bar{2}$	02
3	$03, 1\bar{1}$	$1\bar{1}$
4	10	10
5	$11, 2\bar{3}$	11
6	$12, 2\bar{2}$	12

TECHNIQUE OF CONVERSION FROM BINARY NUMBER TO QSD NUMBER

1-digit QSD can be represented by one 3-bit binary equivalent as follows

$$3 = 101$$
$$\bar{2} = 110$$
$$\bar{1} = 111$$
$$0 = 000$$
$$1 = 001$$
$$2 = 010$$
$$3 = 011$$

So to convert n-bit binary data to its equivalent q-digit QSD data, we have to convert this n-bit binary data into 3q-bit binary data. To achieve the target, we have to split the 3rd, 5th, 7thbit…. i.e. odd bit (from the LSB to MSB) into two portions. But we cannot split the MSB. If the odd bit is 1 then, it is split into 1 & 0 and if it is 0 then, it is split into 0 & 0. An example makes it clear, the splitting technique of a binary number (1101101)2is shown below:



So we have to split the binary data (1) q− times (as example, for conversion of 2-bit quaternary number, the splitting is 1 time; for converting 3-digit quaternary number the split is 2-times and so on). In each such splitting one extra bit is generated. So, the required binary bits for conversion to it's QSD equivalent (n) = (Total numbers of bits generated after divisions) – (extra bit generated due to splitting).

$$n = 3q - \{1 \times (q-1)\}$$
$$= (2q+1)$$

So, number of bits of the binary number should be 3, 5, 7, 9 etc for converting it to its equivalent QSD number. Now every 3-bit can be converted to its equivalent QSD according to the equation. The following two examples as given below will help to make the things clear

III. VEDIC INTRODUCTION

As all of us know that the Computation unit is main unit of any technology, which performs different arithmetic operations like as addition, subtraction and multiplication etc. also in some places it performs logical operations also like as and, or, invert, x-or etc. which is dominant feature in the digital domain based applications. ALU is the execution unit which does not only performs Arithmetic operations but also Logical operations. And that's why ALU is called as the heart of Microprocessor, Microcontrollers, and CPUs. No technology can exist, without those operations which are performed by ALU. Every technology uses works upon those operations either fully or partially which are performed by ALU. The block diagram of ALU is given below, where ALU has been implemented on FPGA tool

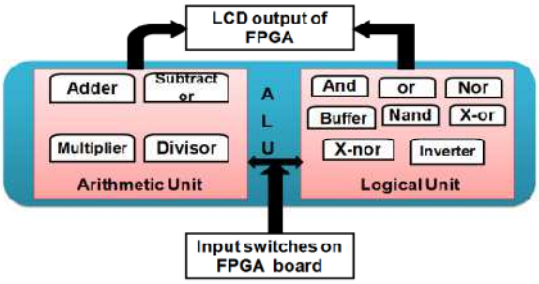
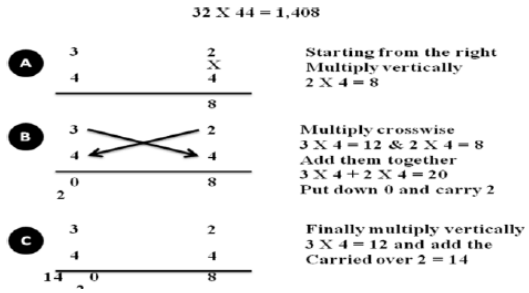


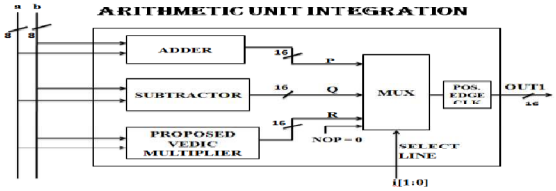
Figure 1.1 Block Diagram of ALU Here the input interface to access ALU module is input switches on FPGA board, and after processing on the data the result can be seen from LCD output of FPGA. For multiplication purpose vedic Urdhva Triyambakam multiplication scheme has been used. Urdhva Triyakbhyam Sutra is a general multiplication formula

applicable to all cases of multiplication. It literally means "Vertically and Crosswise". To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (32 × 44). The conventional methods already know to us will require 16 multiplications and 15 additions. An alternative method of multiplication using Urdhva Triyakbhyam Sutra is shown in following figure. The Vedic multiplication algorithm for 2 digit decimal numbers is shown below:-

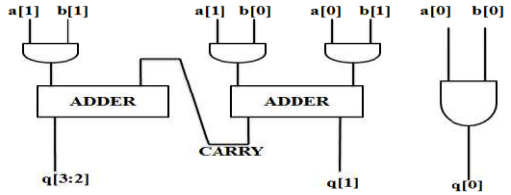


On the basis of this the conventional vedic Multiplier hardware has been designed which is shown below for 4x4 Bit, Using the same approach N-Bit Multiplier can be introduced.

IV. PROPOSED VEDIC ALU MODULE



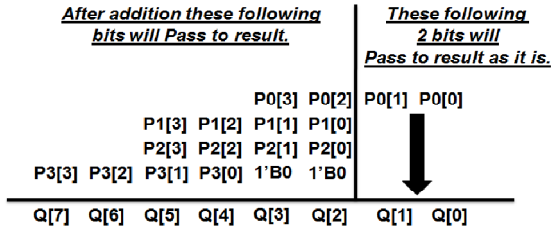
Here a and b are the two 8 bit inputs of our Arithmetic Unit. And other sections of the design are self-explanatory. For 2-Bit multiplication Conventional Vedic multiplication Hardware has been used. As at 2 bit level multiplication we have not to worry about the carry propagation path.



Here inputs are a [1:0] and b[1:0], and output is q[3:0].

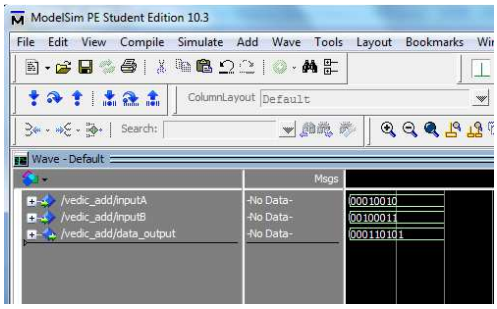
Fig:-Two Bit Vedic Multiplier

Diagram for Unique addition tree structure for partial product addition for 4 bit is given in the following:-

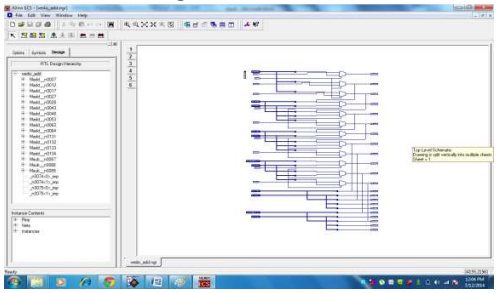


V. SIMULATION RESULT & ANALYSIS

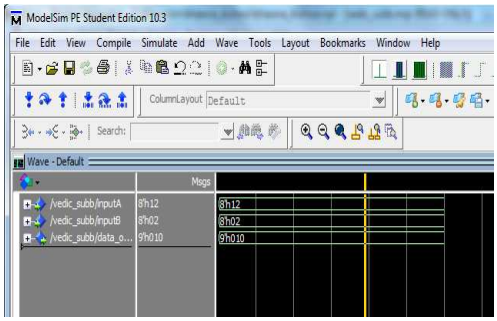
1. VEDIC ADDER



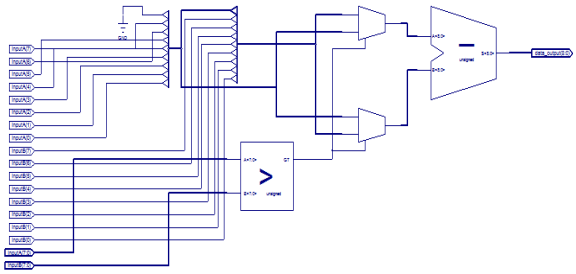
2. RTL design of Vedic 8 bit adder



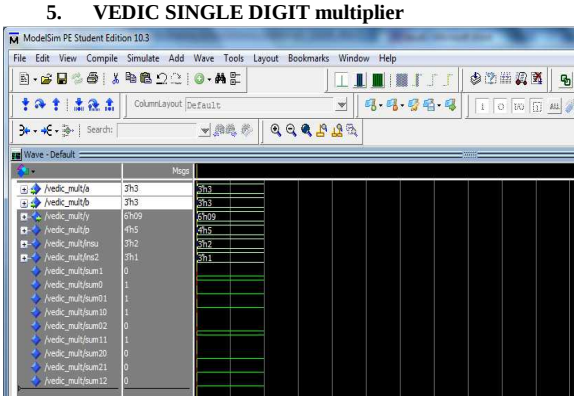
3. VEDIC SUBTRACTOR



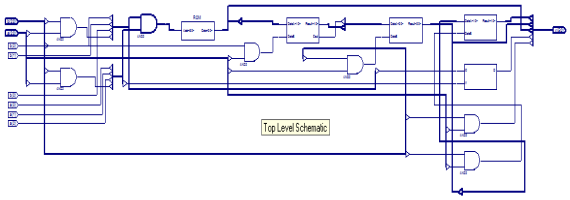
4. RTL design of vedic 8 bit subtractor



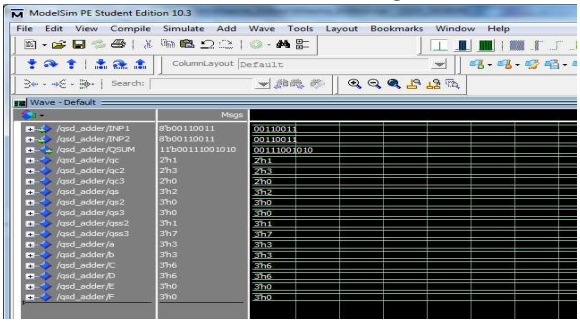
5. **VEDIC SINGLE DIGIT multiplier**



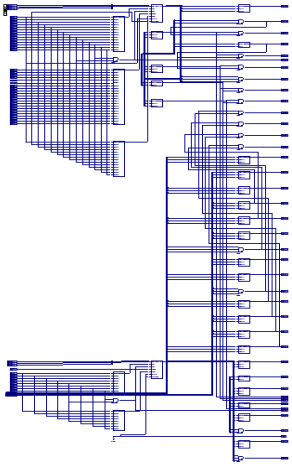
6. **RTL of single digit Vedic multiplier**



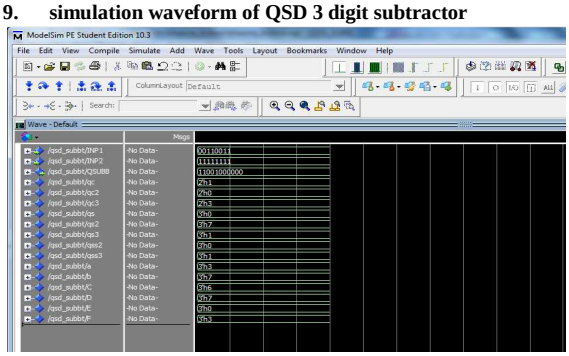
7. **Simulation waveform of QSD 3 digit adder**



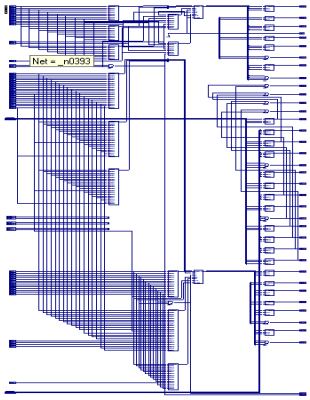
8. **RTL Design of QSD 3 digit adder**



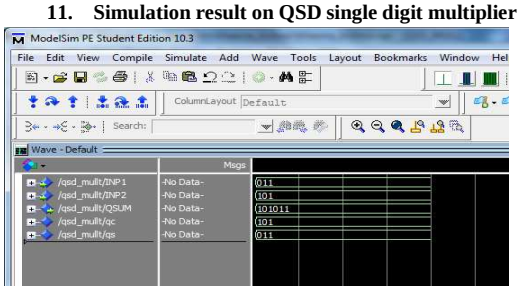
9. **simulation waveform of QSD 3 digit subtractor**



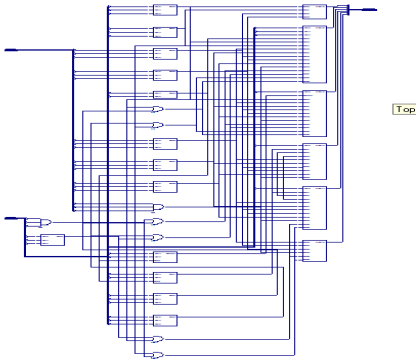
10. **RTL of QSD 3 digit subtractor**



11. **Simulation result on QSD single digit multiplier**



12. **RTL of QSD single digit multiplier**



Comparison on VEDIC & QSD

S. No.	Operation	Types	AREA			Timing
			Gate density	LUT	Slices	
1	adder	QSD	1225	172	91	12.728ns
		VEDIC	8891	118	66	40.232ns
2	Subtractor	QSD	1344	190	100	13. 183ns
		VEDIC	10231	230	165	42.042ns
3	multiplier	QSD	126	21	11	11.580ns
		VEDIC	120	20	11	14.196ns

VI. CONCLUSION

I have studied more about Quaternary signed digit no system and VEDIC procedure for implementing ALU.. Vedic concept utilization is increase the speed of calculation and QSD concept increase the comfort of calculation so thesis focus on comparison study on QSD ALU and VEDIC ALU.

I have implemented both alu on VHDL with the help of Xilinx tool and simulation with modelsim. It was tested and verified from previous result. I have mention simulation waveform and synthesis result along with screen shot, I also shown comparison chart between QSD and VEDIC for area and timing delay.

Table was shown the brief result of area and timing and I have conclude that QSD addition and subtraction having less dense area with less timing delay while vedic is no more efficient. In the meanwhile multiplication in Vedic is having good result in terms of area while timing delay is far occupancy.

Overall QSD is having better response over VEDIC in arithmetic unit. But we talk about logical part it would be same for both.

Future work

Arithmetic circuit is the backbone of digital system and DSP unit. Mostly used element of the application is ALU (Addition, subtraction, Multiplication). In this thesis I have proposed a comparison model for QSD and Vedic alu and have studied about them.I found the qsd alu having better performance then Vedic as it was shown in table 6.1 in chapter 6. I also have shown result waveform in same chapter. This work was done in 3 digit qsd unit or 8 bit input stream for addition and subtraction. While I have focused only single digit multiplication for QSD and Vedic.

In future the comparison study can be extent up to more input and multiplication process can be extent up to multiple digits And QSD multiplier can be compare with all digital multiplier.

This RTL design cab be implement on Nano scale 45nm TSMC library file and obtain the power performance with lower timing delay.

REFRENCES

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

[2] Himanshu Thapliyal and Hamid R. Arabnia, “A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics”, Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[3] E. Abu-Shama, M. B. Maaz, M. A. Bayoumi, “A Fast and Low Power Multiplier Architecture”, The Center for Advanced Computer Studies, The University of Southwestern Louisiana Lafayette, LA 70504.

[4] Harpreet Singh Dhillon and Abhijit Mitra, “A Reduced-Bit Multiplication Algorithm for Digital Arithmetics”, International Journal of Computational and Mathematical Sciences 2;2 © www.waset.org Spring 2008.Shamim Akhter, “VHDL Implementation of Fast NXN Multiplier Based on Vedic Mathematics”, Jaypee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[6] Charles E. Stroud, “A Designer‟s Guide to Built-In Self-Test”, University of North Carolina at Charlotte, ©2002

Kluwer Academic Publishers New York, Boston, Dordrecht, London, Moscow.
[7] Douglas Densmore, "Built-In-Self Test (BIST) Implementations An overview of design tradeoffs", University of Michigan EECS 579 – Digital Systems Testing by Professor John P. Hayes 12/7/01.
[8] Shripad Kulkarni, "Discrete Fourier Transform (DFT) by using Vedic Mathematics", report, vedicmathsindia.blogspot.com, 2007.
[9] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja,"Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.Himanshu Thapliyal, Saurabh Kotiyal and M. B Srinivas, "Design and Analysis of A Novel Parallel Square and Cube Architecture Based On Ancient Indian Vedic Mathematics", Centre for VLSI and Embedded System Technologies, International Institute of Information Technology, Hyderabad, 500019, India, 2005 IEEE.
[11] Himanshu Thapliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad-500019, India. Abhijeet Kumar, Dilip Kumar, Siddhi, "Hardware Implementation of 16*16 bit Multiplier and Square using Vedic Mathematics", Design Engineer, CDAC, Mohali.
[13] Himanshu Thapliyal and M.B Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics", IEEE, 2005.
[14] "Spartan-3E FPGA Starter Kit Board User Guide", UG230 (v1.1) June 20, 2008.
[15] Deming Chen, Jason Cong, and Peichan Pan, "FPGA Design Automation: A Survey", Foundations and Trends in Electronic Design Automation Volume 1 Issue 3, November 2006.
[16] Ken Chapman, "Initial Design for Spartan-3E Starter Kit (LCD Display Control)", Xilinx Ltd 16th February 2006.
[17] Michael L. Bushnell and Vishwani D. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Kluwer Academic Publishers, 2002.
[18] R. Bencivenga, T. J. Chakraborty and S. Davidson, "The Architecture of the Gentest Sequential Test Generator", in Proc. of the Custom Integrated Circuits Conference, pp. 17.1.1–17.1.4, May 1991.
[19] E. B. Eichelberger, E. Lindbloom, J. A. Waicukauski and T. W. Williams, "Structured Logic Testing", Englewood Cliffs, New Jersey, Prentice-Hall, 1991.
[20] M. Abramovici and C. Stroud, "BIST-Based Test and Diagnosis of FPGA Logic Blocks", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.9, No.1, pp159-172, 2001.