INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

VLSI Implementation Of Vedic Mathematics And Its Application In RSA Cryptosystem

Greeshma Liz Jose
Department Of Electronics And Communication
Viswajyothi College Of Engineering And Technology, Vazhakulam, Kerala, India
Sani John
Department Of Electronics And Communication
Viswajyothi College Of Engineering And Technology, Vazhakulam, Kerala, India

Abstract:
Vedic mathematics is the ancient system of Indian mathematics. This paper suggests a method for VLSI implementation of the Vedic multiplication and division algorithms. It also attempts to perform a comparison based on speed, power, and area with conventional multiplication and division algorithms. Hence the vedic systems are better to the conventional systems, the vedic algorithms are used to implement the RSA encryption/decryption systems. RSA is the widely used public key encryption/decryption method. The Vedic RSA enabled the RSA hardware to work as fast as its software counterparts. Simulation is done in Verilog HDLwith Modelsim and Xilinx ISEsoftwares and implementation is on Xilinx Spartan 3E FPGA.

Key words: *Vedic mathematics, RSA cryptography,Verilog HDL, modelsim, Xilinx Spartan 3E*

1.Introduction

The paper describes the implementation of the Vedic mathematical algorithms for multiplication [2] and division [3] in VLSI and comparative study between Vedic methods the conventional methods. The hardware part of RSA public key cryptosystem needs fast division architecture, in order to work as fast as its software counterparts. So the design of the RSA system [1] with Vedic division is also described. Since the Vedic multiplication is supposed to be faster than conventional methods, the conventional multiplier in the system is replaced with a Vedic multiplier as a modification. A comparative study is done in order to estimate the improvements of the Vedic RSA system over the conventional systems.

The history of cryptography begins with secret writings in the Ancient civilizations. The hieroglyphic writings on the tombs of ancient Egypt's [4] and ancient Spartan's Scytale [5], ancient Indian and Hebrew [6, 7] scripts are examples of ancient cryptography. The Caesar cipher [8] is one of the famous substitution ciphering methods. During the period of Second World War electromagnetic machines like enigma [8] were used.

The modern cryptography can be divided into private (symmetric) key encryption methods [9] and public (asymmetric) key encryption methods [10]. Data Encryption Standard [10] – DES is a private key method. To solve the insecurity due to the usage of single key, dual key methods called public key methods were introduced. Diffie – Hell man key exchange [11] is the first approach towards public key methods.

The first successful public key method is RSA cryptosystem, [12,13,14] introduced by Rivest, Shamir, and Adleman in 1977. RSA public key cryptosystem [15] needs fast division architecture, in order to work as fast as its software counterparts. But division is the slowestALUoperation. So it cannot use in RSA systems. Instead of this, modular multiplication [16,17,18] schemes are used. In this paper the fast Vedic division and multiplication methods [19] are used to fasten the RSA system.

2.Multipliers

The Vedic and conventional multipliers are designed using VerilogHDL, simulated and implemented. The details of them are given below.

The Vedic Multiplier

The multiplier is based on an algorithm named as Urdhva Tiryakbhyam (Vertical &Crosswise) of ancient Indian Vedic Mathematics. It is based on a novel concept through which the generation of all partial products can be done and then, the concurrent addition of these partial products can be done.

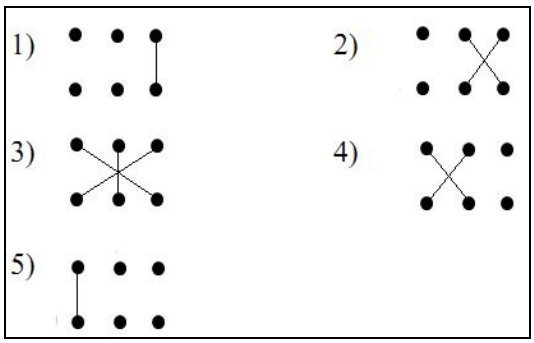

Figure 1: Line Diagram Of Urdhva Tiryakbhyam

Based on this line diagram the hard ware of any multiplier can be formed. The 2*2 and 4*4 multipliers are shown in figure 2 and figure 3.

The Conventional Multipliers

The array multiplier and Booth's multiplier are examples of conventional multiplication algorithms. The array multiplier is the implementation of the pen and paper multiplication. The implementation of the same is shown in the figure 4.

The Booth's algorithm is an encoding algorithm uses shifting operations

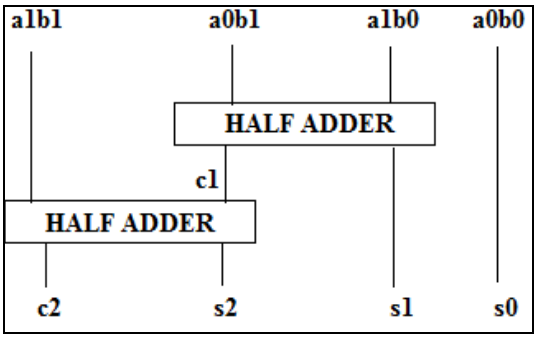

*Figure 2: The 2*2 Vedic Multiplier*

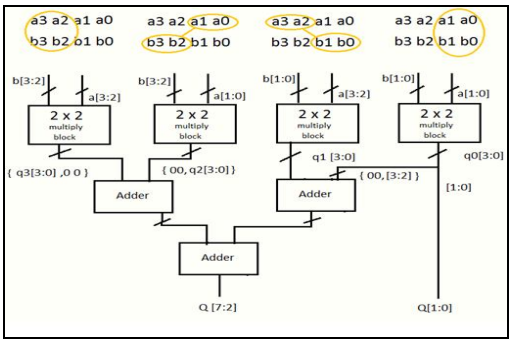

*Figure 3: The 4*4 Vedic Multiplier*

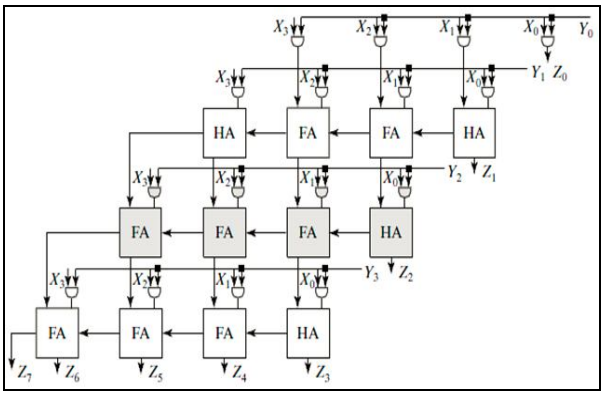

*Figure 4: The 4*4 Vedic Multiplier*

Booth's Algorithm can be represented as below :

Let multiplicand = 4 bits, multiplier = 4 bits, and output = 8 bits.

Product: {0000, Multiplier, 0}

Now, take the two least significant bits of the product and depending on the value proceed with one the following:

00: No changes to product.
01: Add the multiplicand to the left side of the product.
10: Subtract the multiplicand from the left side of the product.
11: No changes to product.
Right shift the product by 1 bit.
Repeat the process x number of times, where x is the number of bits in the multiplicand.

3.Division

The Vedic and conventional divisions are designed, simulated and implemented. The restoring and non- restoring algoriths are the conventional methods used.

3.1.The Vedic Division

Vedic mathematics describes a method called 'Dhvajanka – On the top of the flag' which is a generalized formula for division. It is based on the formula Urdhva-tiryagbhyam. The steps in the Vedic division are shown below:

- The divisor and dividend are arranged in the form shown below. Only leftmost digit of the divisor is left aside. The dividend is separated in two sections right part consisting number of digits equal to digits in the divisor. The divisor is represented by d, divided by X, and quotient by A.
- Only the first digit of the dividend is divided by the left out digit, quotient and remainder of this division are noted.
- During the next iteration remainder from the previous iteration is used with next digit of the dividend. Quotient digits and dividend digits without leftmost digit are multiplied in vertically and crosswise manner. This product is subtracted from the number formed by combination of the remainder and digit of the remainder.
- The number left after subtraction in step 3 is divided by leaving out digit of divisor quotient is noted and the remainder is prefixed with the rest of the digits of the dividend.
- This process is continued till the same number of quotient digits equal to digits in the left part of the dividend is obtained.
- Remainder is obtained by subtraction of right part of the dividend prefixed by last remainder and cross multiplication of quotient and divisor.

The block diagram of the above algorithm is shown in the figure 5.

3.2. The Conventional Division

Restoring division follows the same method as the pen and paper long division algorithm. In the long division algorithm, the divisor is compared to the left digits of the dividend. If the divisor is bigger than the dividend numbers being compared, then a 0 in appended to the quotient and divisor is shifted to the right to compare with bigger dividend digits. If the divisor is smaller than the dividend, then the divisor being compared is subtracted from the dividend and the result is stored as a reminder, while the number of times the divisor can go into the dividend is appended to the quotient.

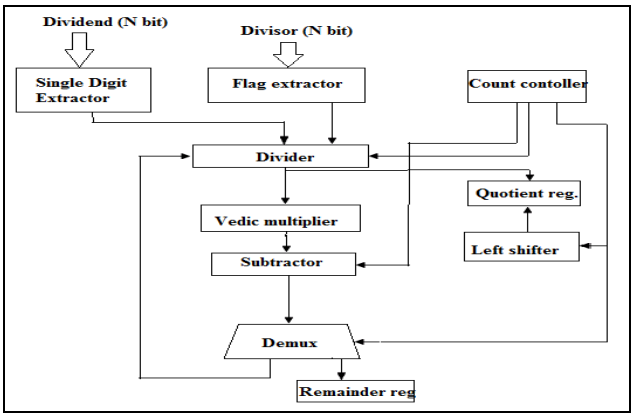

Figure 5:The Block Diagram Of Vedic Divider

The algorithm to perform non-restoring division is as follows:

- Shift remainder left 1 bit.
- If remainder is negative, add Divisor to the left half of the remainder. Shift quotient left 1 bit.
- If remainder is positive, subtract Divisor from the left half of the remainder. Shift quotient left 1 bit and add 1.
- Repeat for number of bits in divisor.
- Correction step: If remainder is negative, add divisor to the remainder to obtain the correct value.

4.The RSA Cryptosystem

RSA is one of the safest standard algorithms, based on public-key, for providing security in networks. The following section describes the RSA algorithm and the figure 6 shows the block diagram of the RSA architecture.

4.1. The RSA Algorithm

The steps of a basic RSA algorithm are shown below.

- Select any prime numbers p, q
- Compute n = p*q
- Compute phi = (p-1) * (q-1)
- Select e, such that 1< d < phi, and gcd (phi, d) = 1
- Find d such that e*d = 1 mod phi
- Public Key = {d, n}
- Private Key = {e, n}

For any plaintext m < n,

Encryption, c = (mod n).

Decryption, m = (mod n).

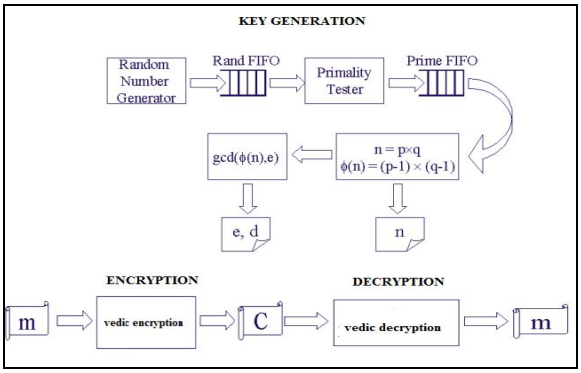

Figure 6: The Block Diagram Of RSA Crypto System

4.2.The Block Diagram Description

In the above diagram 'm' is the message, which ciphered to 'c' using the encryption algorithm. The cipher text 'c' is the input to decryption module which deciphered it into the original message 'm'. The upper section shows the key generation. RSA has two keys – a public key 'd' and private key 'e'. Public key is chosen and private key is generated. The 'p' and 'q' are two prime numbers, 'n' is product of the above prime numbers and 'phy' is (p-1* q-1). The vedic encryption and decryption means the encryption and decryption algorithms modified by ancient Indian Vedic multiplication and division, which discussed in the earlier sections.

The RSA cryptographic architecture consists of the following structures.

- Pseudo Random Number Generator
  A 'Linear Feedback Shift Register' (LFSR) is used as a Pseudo Random Number Generator. The figure 7 shows the structure of the LFSR.
- Primality Tester
  Output of pseudo random number generator should be check for prime and if not ignore the same and generate next number and the process continues. The RAND FIFO and PRIME FIFO are two registers which are used to store random numbers and prime numbers respectively.
- Greatest Common Divisor (GCD) Finder
  After primality test 'p' and 'q' are tested for 'greatest common divisor' (GCD) condition. In mathematics, the greatest common divisor (gcd) of two or more integers (at least one of which is not zero), is the largest positive integer that divides the numbers without a remainder.The most famous method to find the gcd is using 'Euclid's algorithm'.
- Private Key Generator
  It uses 'phi' and 'd' to compute private key 'e' needed for decryption. 'e', 1< e< phi, the private key is computed from e and phi as e*d = 1 mod phy.

- Encryption and Decryption
  The encryption block computes cipher c = m^d (mod n), and the decryption block deciphers the cipher back into message as m= c^e (mod n). This block requires a multiplier and a divider. The conventional and Vedic methods for multiplication and divisions are used here.

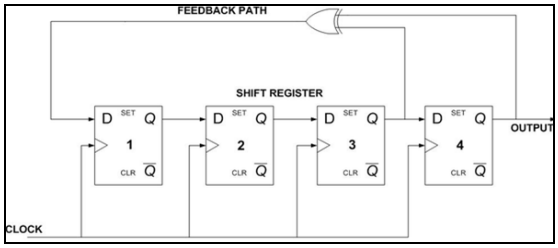

Figure 7: The Linear Feedback Shift Register

5.Results And Comparisons

ModelsimSE 6.2c and Xilinx ISE8.1i are the softwares used for the simulation. After simulation all the designed systems were implemented on the Xilinx Spartan 3E FPGA. The FPGA kit used for the implementation is Xilinx Spartan 3E (family), XC3S50 (device), ft 256 (Package), -4 (speed grade).
The waveforms from the Modelsim are shown below.

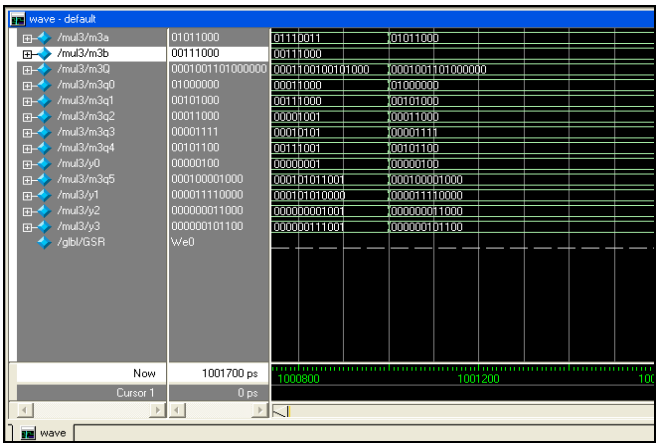

Figure 8: .The Simulation Result Of Vedic Multiplier

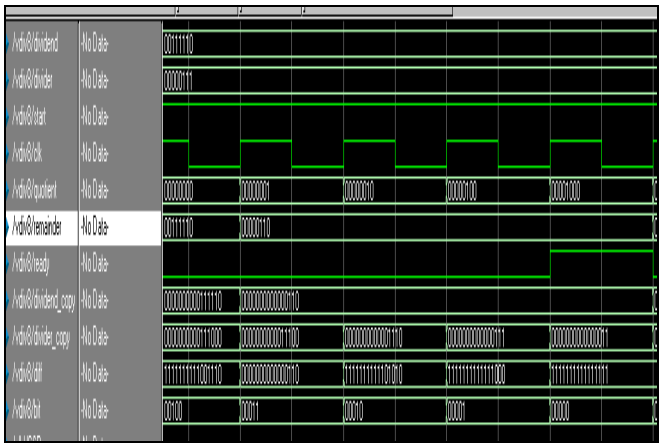

Figure 9: .The Simulation Result Of Vedic Divider

The implementation results includes the design summery
from the Xilinx ISE software. Using these results the comparisons in speed, power, and area are performed between conventional and Vedic methods. The results will be shown in the table 1.

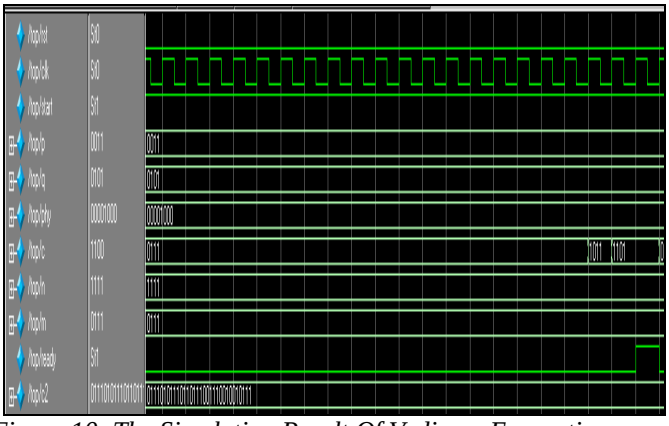

Figure 10: The Simulation Result Of Vedicrsa Encryption.

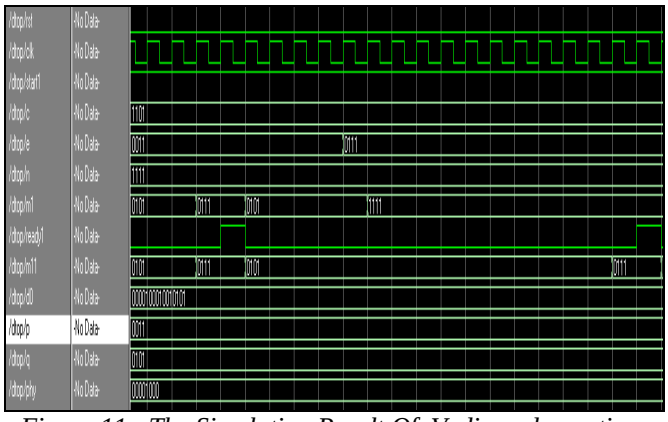

Figure 11: The Simulation Result Of Vedicrsadecryption

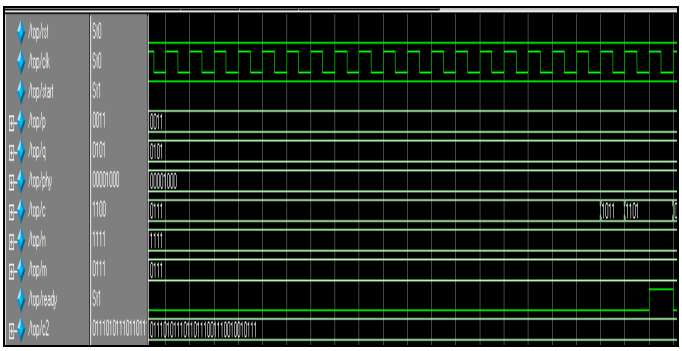

Figure 12: Simulation Result Of Modified Vedic RSA Encryption

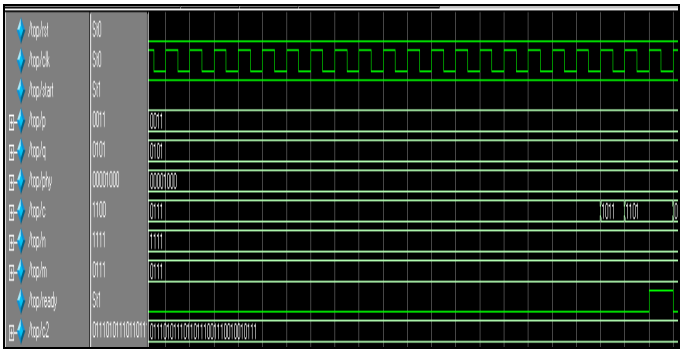

Figure 13: Simulation Result Of Modified Vedic Rsadecryption

The array and Booth's multiplier and restoring and non-restoring division algorithms are also designed for the purpose of comparison. The RSA encryption and decryption is first designed using the conventional division and multiplication. Then the division is replaced with the Vedic division andresults compared. As a modification the Vedic multiplier is used and the results compared.

Method	Comparison Between 8 Bit Multipliers		
	Delay in ns	No. of Slices Used (Area)	Power Consumption in m W
Vedic	26.99	71	18
Array	39.32	96	24
Booth's	42.43	119	32

Table 1: The Comparison Results Of multiplication

Method	Comparison Between 8 Bit Divisions		
	Delay in ns	No. of Slices Used (Area)	Power Consumption in m W
Vedic	7.63	35	12
Restoring	42.59	60	36
Non-restoring	44.356	68	39

Table 2: The Comparison Results Of Division

Method	Comparison Between RSA Encryptions		
	Delay in ns	No. of Slices Used (Area)	Power Consumption in m W
Vedic	76.14	185	212
Restoring	584.07	4257	428
Non-restoring	489.21	2095	374
Array	88.21	206	246
Booth's	92.34	219	255

Table 3: The Comparison Results Of RSA Encryption

Method	Comparison Between RSA Encryptions		
	Delay in ns	No. of Slices Used (Area)	Power Consumption in m W
Vedic	17.47	50	14
Restoring	94.92	236	78
Non-restoring	67.25	135	65
Array	21.7	74	17
Booth's	23.92	97	19

Table 4: The Comparison Results Of RSA Decryption

6.Conclusion

The Vedic multiplier using the vertical crosswise algorithm and Vedic division using at the top of the flag algorithm are designed, simulated and implemented. The new architectures are then compared with the existing algorithms. The result is shown in the tables above. The Vedic multiplication is faster, uses very less number of slices, and consumed lower power than the array or Booth's multiplier. Similarly, the Vedic division also area efficient and consumes lower power. The delay of the Vedic divider is much lower than both the existing algorithm.

The RSA algorithm implemented using Vedic division is improved in speed, power, and in area utilization. The Vedic multiplication fastens the system to a level at which the hardware can work in synchronization with the software counterparts.

Since the system is supposed to implement on the Xilinx Spartan 3E FPGA kit, the key size and prime number width can't increase beyond the 4 bits. The improved key size and higher bit primes results in a much secure RSA system.

7.References

1. R.G. Kaduskar,"A New Architecture for RSA Algorithm Using Vedic Mathematics" IEEE fourth international conference on emerging trends, 2011 pp 233-237.
2. S. Kumaravel, Ramalatha Mariimuthu, VLSI Implementation of High Performance RSA Algorithm Using Vedic Mathematics",IEEE conference on computer intelligence 2007.
3. Prabir Saha, Arindam Banerjee,"Vedic Divider: Noval Architecture for High Speed VLSI Applications",IEEE international symposium on electronic system design 2011, PP 126-128.
4. Stamatios V. "A Primer on Cryptography in Communication", in IEEE cummuniction magazine,April 2006,pp 147-151.
5. Antonio Mazzeo,"Scanning the Issue", poceedings of IEEE, Feb.2006,Vol.94,No.2,pp 343-345.
6. Ayushi Jagid "Cryptography", international journal for scientific and engineering research,Nov. 2010, Vol 1 , Issue 2.
7. Varsha Sahni, Jaspreet Kaur,"Encryption Standards of Cryptography",UNIASCIT, 2012, Vol 2,pp 225-228.
8. David B. Newman,Jim K. Omura, "Public Key Management for Network Security",in IEEE network magazine, April 1987,Vol 1, pp 11-15.
9. Peter Guttman David Naccale , "What is Cryptography",in IEEE security and Privazy magazine, June 2006.
10. Amos Beimel, Benny Chor,"Secret Sharing with Public Reconstruction",
11. transaction on information theory,Sept 1998,pp 1887-1895.
12. Diffie, W., and Hellman, M. New directions in cryptography. IEEE Trans. Inform.Theory IT-22,PP 644-654 ,Nov. 1976.
13. R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, February 1978,21 (2), pp. 120-126.
14. Chiranth E,Chakravarthy H,"implementation of rsa Cryptosystem using verilog" international Journal for Scientific and Engineering Research,May 2011.
15. Vibhor Garg, V. Arunachalam,"Architectural analysis of RSA cryptosystem on fpga" international Journal for computer application, July 2011.
16. Ankit Anand,Pushkar Praveen ,"implementation of rsa algorithm on fpga", international Journal for Scientific & Engineering Research,July 2011
17. Jin Hua Hong,Cheng Wen Wu "Cellular array Modular multiplier for fast RSA",IEEE transactions on VLSI systems, June 2003
18. Atsushi Miyamoto,Naufumi Homma"Systematic Design of RSA Processors Based on High Radix Montgomery Multipliers" ",IEEE transactions on VLSI systems , july 2011
19. Ming Der Shieh,jun Hong Chen,"A new modular exponentiation architecture for efficient design of RSA Cryptosystem" ",IEEE transactions on VLSI systems,Sept.2008.
20. Swami Sri Bharath, Krsna Tirathji, "Vedic Mathematic" Motilal Banarsidas , Varanasi(India),1986.