ISSN : 2230-7109 (Online) | ISSN : 2230-9543 (Print)
IJECT Vol. 3, Issue 4, Oct - Dec 2012

Radix-8 Booth Encoded Modulo Multiplier

[1]Anuradha. K, [2]Rupesh Kumar Penugonda, [3]Thammishety Narasimharao
[1]Dept. of ECE, Anurag Engineering College, Kodad, AP, India
[2]Dept. of ECE, Sana Engineering College, Kodad, AP, India
[3]Dept. of ECE, Sana Polytechnic, Kodad, AP, India

Abstract
To design an efficient integrated circuit in terms of area, power and speed, has become a challenging task in modern VLSI design field. The encryption and decryption of PKC algorithms are performed by repeated modulo multiplications these multiplications differ from those encountered in signal processing and general computing applications. The Residue Number System (RNS) has emerged as a promising alternative number representation for the design of faster and low power multipliers owing to its merit to distribute a long integer multiplication into several shorter and independent modulo multiplications. The multipliers are the essential elements of the digital signal processing such as filtering, convolution, transformations and Inner products. RNS has also been successfully employed to design fault tolerant digital circuits. The modulo multiplier is usually the noncritical data path among all modulo multipliers in such high-DR RNS multiplier. This timing slack can be exploited to reduce the system area and power consumption without compromising the system performance. With this precept, a family of radix-8 Booth encoded modulo multipliers, with delay adaptable to the RNS multiplier delay, is proposed. In this paper, the radix-8 Booth encoded modulo multipliers whose delay can be tuned to match the RNS delay. In the proposed multiplier, the hard multiple is implemented using small word-length ripple carry adders (RCAs) operating in parallel. The carry-out bits from the adders are not propagated but treated as partial product bits to be accumulated in the CSA tree. The delay of the modulo multiplier can be directly controlled by the word-length of the RCAs to equal the delay of the critical modulo multiplier of the RNS. By combining radix-8 Booth encoded modulo multiplier, CSA and prefix architecture of multiplier, for high speed and low-power is achieved.

Keywords
Blue Noise Half Toning, Error Diffusion, Secret Sharing, Visual Cryptography

I. Introduction
Plain text into cipher text using some key which is not readable format others then it transmitted through channel again cipher text is converted into plain text at receiver end using key , if we are using same key at transmitter and receiver then it is called as public key cryptography , if we are using different keys at transmitter and receiver then it is called as private key cryptography ,but public key cryptography has more advantages and high performance compare to private. R IVEST, Shamir, and Adelman (RSA) and Elliptic Curve Cryptography (ECC) are two of the most well established and widely used Public Key Cryptographic (PKC) algorithms. The encryption and decryption of these PKC algorithms are performed by repeated modulo multiplications [1-3]. These multiplications differ from those encountered in signal processing and general computing applications in their sheer operand size. key sizes of RSA and ECC are typically very high, hence the key multiplication becomes very difficult and the long carry propagation of large integer multiplication is limited the entire system performance, other system come into existence is The residue number system (RNS) RNS has also been successfully employed to design fault tolerant digital circuits [1, 3].

Multipliers are most commonly used in various electronic applications e.g. Digital signal processing in which multipliers are used to perform various algorithms like FIR, IIR etc. Earlier, the major challenge for VLSI designer was to reduce area of chip by using efficient optimization techniques to satisfy MOORE'S law. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today's microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors and today's general-purpose processors. However area and speed are two conflicting constraints.

So improving speed results always in larger areas. Now, as most of today's commercial electronic products are portable like Mobile, Laptops etc. that require more battery backup. Therefore, lot of research is going on to reduce power consumption. So, in this paper it is tried to find out the best solution to achieve low power consumption, less area required and high speed for multiplier operation.

The basic principle used for multiplication is to evaluate partial products and accumulation of shifted partial products. In order to perform this operation number of successive addition operation is required. Therefore one of the major components required to design a multiplier is Adder. Adders can be Ripple Carry, Carry Look Ahead, Carry Select, Carry Skip and Carry Save [1-3]. A lot of research work has been done to analyze performance of different fast adders. Encoding is an effective method which greatly increases the speed of our algebra. We propose the first-ever family of low-power radix-8 Booth encoded multipliers. In the proposed multiplier, the hard multiple is generated using small word-length ripple carry adders (RCAs) [1] operating in parallel. The carry-out bits from the adders are not propagated but treated as partial product bits to be accumulated in the CSA tree [6]. The effect of the RCA word-length, on the time complexities of each constituent component of the multiplier is analyzed qualitatively and the multiplier delay is shown to be almost linearly dependent on the RCA word-length. Consequently, the delay of the multiplier can be directly controlled by the word-length of the RCAs. By means of modulo arithmetic properties, we show that the compensation constant that negates the effect of the bias introduced in this process can be precomputed and implemented by direct hardwiring with no delay overhead for all feasible combinations of and it is shown that the proposed multiplier lowers power dissipation of the radix-4 Booth encoded multiplier.

This paper focuses on the design space exploration of arithmetic operation in one of the two special moduli, i.e., the modulo 2^n-1 multiplier design. The Montgomery modulo multiplication, while computing the modular product without trial division, is modulus-independent and incapable of exploiting number theoretic properties of modulo 2^n-1 arithmetic for combinational circuit simplification. The properties of modulo 2^n-1 arithmetic were most effectively exploited for the full adder based implementation of modulo multiplier, the multiplier bits were not encoded, which

lead to higher implementation area and longer partial product accumulation time. In [10], the radix-4 Booth encoding algorithm was employed to reduce the number of partial products to [n/2]+1and [n/2], respectively. The shorthand notations [a] and [a] and denote the smallest integer greater than or equal to 'a' and the largest integer smaller than or equal to 'a', respectively. With higher radix Booth encoding, the number of partial products is reduced by more than half and consequently, significant reduction in silicon area and power dissipation is feasible [11]. The radix-8 Booth encoding reduces the number of partial products to [n/3]+1, which is more aggressive than the radix-4 Booth encoding. However, in the radix-8 Booth encoded modulo 2^n-1 multiplication, not all modulo-reduced partial products can be generated using the bitwise circular-left-shift operation and bitwise inversion. Particularly, the hard multiple |+3x| 2^n-1 is to be generated by an n-bit end-around-carry addition of X and 2X.The performance overhead due to the end-around-carry addition is by no means trivial and hence, the use of Booth encoding for modulo 2^n-1 multipliers have been restricted to only radix-4 in literature. In this paper, we propose the first-ever family of low-area and low-power radix-8 Booth encoded modulo 2^n-1 multipliers whose delay can be tuned to match the RNS delay closely. In the proposed multiplier, the hard multiple is generated using small word-length ripple carry adders (RCAs) operating in parallel. The carryout bits from the adders are not propagated but treated as partial product bits to be accumulated in the CSA tree. The effect of the RCA word length, k on the time complexities of each constituent component of the multiplier is analyzed qualitatively and the multiplier delay is shown to be almost linearly dependent on the RCA word-length. Consequently, the delay of the modulo 2^n-1 multiplier can be directly controlled by the wordlength of the RCAs to equal the delay of the critical modulo multiplier of the RNS. By means of modulo 2^n-1 arithmetic properties, we show that the compensation constant that negates the effect of the bias introduced in this process can be pre-computed and implemented by direct hard wiring with no delay overhead for all feasible combinations of n and k. It is shown that the proposed multiplier lowers the area and power dissipation of the radix-4 Booth encoded modulo 2^n-1 multiplier under the delay constraints derived from various high dynamic range RNS multipliers.

II. Radix-8 Booth Encoded Modulo Multiplication Algorithm

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is the standard technique used in chip design, and provides significant improvements over the "long multiplication" technique. Recoding of binary numbers was first hinted at by Booth four decades ago. Mac or ley proposed a modification of Booth's algorithm a decade after. The modified Booth's algorithm (radix-4 recoding) starts by appending a zero to the right of x0 (multiplier LSB). Triplets are taken beginning at position x-1 and continuing to the MSB with one bit overlapping between adjacent triplets. If the number of bits in X (excluding x-1) is odd, the sign (MSB) is extended one position to ensure that the last triplet contains 3 bits. In every step we will get a signed digit that will multiply the multiplicand to generate a partial product entering the Wallace reduction tree. The radix-8 Booth encoding reduces the number of partial products to which is more aggressive than the radix-4 Booth encoding. However, in the radix-8 Booth encoded modulo 2n-1 multiplication, not all modulo-reduced partial products can be generated using the bitwise circular-left-shift operation and bitwise inversion. Particularly,the hard multiple |+3X|2^n-1 is to be generated by an n - bit end-around-carry addition of X and 2X. When applying Booth encoding to a k-bit digit, the resulting encoded digit value is in the range [- 2k+2,2k-2]. Or radix 8, k=3 and the encoded multiplier digit is in the range increases the complexity of the design

Let $X=\sum_{i=0}^{n-1} xi.2i$ and $Y=\sum_{i=0}^{n-1} yi.2i$ represent the multiplicand and the multiplier of the modulo 2^n-1 multiplier, respectively. The radix-8 Booth encoding algorithm can be viewed as a digit set conversion of four consecutive over lapping multiplier bits y3i+2y3i+1y3i(y3i-1) to a signed digit, di,di Є [-4,4] for i=0,1…… [n/3] The digit set conversion is formally expressed as

$$di=y_{3i-1}+y_{3i}+2y_{3i+1}+4y_{3i+2}$$

where $y_{-1}=y_n=y_{n+2}=y_{n-1}$

As by the modified booth algorithms we notice That The partial products will not be decreased for radix -2 and where as the partial products will be divided by factor of two in radix_4 .but if we need to reduce the partial products further means you need to go for the radix-8 the general implementation of radix_8 is differs from our contest because our architecture itself performing the modulo operation .so that, the detailed description towards the proposed method is disused below

Radix-8 table summarizes the modulo-reduced multiples of X for all possible values of the radix-8 Booth encoded multiplier digit, di, where CLS(X, J) denotes a circular-left-shift of X by j bit positions. Three unique properties of modulo 2^n-1 arithmetic that will be used for simplifying the combinatorial logic circuit of the proposed modulo multiplier design are reviewed here.

Table 1: Modulo-Reduced Multiples for the Radix-8 Booth Encoding

d_i	pp_i	$\lvert d_iX\rvert_{2^n-1}$
0	000….000	0..0
+1	$x_{n-1-3i}\ x_{n-2-3i}\cdots x_0\ x_{n-1}\cdots x_{n-3i}$	X
+2	$x_{n-2-3i}\ x_{n-3-3i}\cdots x_0\ x_{n-1}\cdots x_{n-1-3i}$	CLS(x,1)
+3	+3X	$\lvert+3X\rvert_{2^n-1}$
+4	$x_{n-3-3i}\ x_{n-4-3i}\cdots x_0\ x_{n-1}\cdots x_{n-2-3i}$	CLS(X,2)
-4	$\bar{x}_{n-3-3i}\ \bar{x}_{n-4-3i}\cdots\bar{x}_0\ \bar{x}_{n-1}\cdots\bar{x}_{n-2-3i}$	CLS($\bar{x}$,2)
-3	-3X	$\lvert-3X\rvert_{2^n-1}$
-2	$\bar{x}_{n-2-3i}\ \bar{x}_{n-3-3i}\cdots\bar{x}_0\ \bar{x}_{n-1}\cdots\bar{x}_{n-1-3i}$	CLS($\bar{x}$,1)
-1	$\bar{x}_{n-1-3i}\ \bar{x}_{n-2-3i}\cdots\bar{x}_0\ \bar{x}_{n-1}\cdots\ \bar{x}_{n-3i}$	$\bar{x}$
-0	1111…….1111	1..1

From the below properties of modulo arithmetic we can notice that hard ware implementation of modulo multiplier can be reduced. The design architecture reviewed here

Property 1: The modulo 2^n-1 reduction of –X can be implemented as the -bit one's complementation of the binary word as follows:

$$\left|2^jX\right|_{2^n-1}=\sum_{i=0}^{n-j-1}x_i\cdot 2^{i+j}+\sum_{i=n-j}^{n-1}x_i\cdot 2^{i+j-n}=CLS(X,j)$$

$l\text{-}X|_{2n-1}=2^n-1-X=X$……

Property 2: For any nonnegative integer, the periodicity of an integer power of two over

Modulus can be stated as follows:

$|2^{n.8+i}|_{2n-1} = |2^{n.8}|_{2n-1} .|2^{i}|_{2n-1} |_{2n-1} = |2^{i}|_{2n-1}$.....

Property 2 ensures that the modulo 2^n-1 reduction of binary exponents can be implemented with no logic cost. As a corollary, the modulo 2^n-1 reduction of the product of a binary word X and an integer power of two, 2j, is equivalent to CLS(X, J). This property can be formally expressed as Property 3.

In Table above ,the modulo 2^n-1 reduction for di $\in \{\pm1,\pm2,\pm3,\pm4\}$ are replaced by simple bitwise inversion and bitwise circular-left-shift of X using Properties 1 and 3, respectively.

MBA reduces the number of partial products by a factor of two, without requiring a pre-adder to produce the partial products. In general, there will be [n+2/2] partial a product where 'n' is the operand length. Here, it reduces the number of partial Products by half but requires a carry propagate add to produce the "3M" multiplier, before the partial products are generated. The implementation of radix-8 partial products is two types Soft multiple and hard multiple, soft multiple can be generated by simple bitwise inversion and bitwise circular leftshift. For redundant 2 we performing CLS by one position as well as for 4 performing CLS by two positions as we discussed in above properties, for redundant '0' partial product will '0',for '1' same multiplicand number will present Where as we can't generate hard multiple i.e.3, 5, 7…etc using CLS or Any other shifting operations. So we need to design 3X in two Ways i.e., 2X+X or 4X-X if we design 3X using 4X-X hard ware Circuitry will increase, another way to implement is 2X+X.Hense the hard ware circuit will decrease compare to 4XX.

0	0	0 0	1	0	0	0 0	1	B+0
x_7	x_6	x_5 x_4	$\bar{x}_4$	x_3	x_2	x_1 x_0	$\bar{x}_0$	B+X
x_6	x_5	x_4 x_3	$\bar{x}_3$	x_2	x_1	x_0 x_7	$\bar{x}_7$	B+2X
x_5	x_4	x_3 x_2	$\bar{x}_2$	x_1	x_0	x_7 x_6	$\bar{x}_6$	B+4X

Fig. 1: Generation of |3X |2n-1 using two –bit RCAs

The above technique for computation involves two -bit carry-propagate additions in series such that the carry propagation length is twice the operand length In the worst case, the late arrival of the may considerably delay all subsequent stages of the modulo multiplier. Hence, this approach for hard multiple generation can no longer categorically ensure that the multiplication in the modulo channel still falls in the noncritical path of a RNS multiplier. In what follows, we propose a family of low-power and low-area modulo multipliers based on the radix- 8 Booth encoding, which allows for an adaptive control of the delay to match the delay of the critical modulo channel of a RNS multiplier.

III. Proposed Radix-8 Booth Encoded Modulo Multiplier Design

Generation of Partially-Redundant Hard Multiple Let and be added by a group of bit RCAs such that there is no carry propagation between the adders. Fig. 2 shows this addition for and, The idea is to form the 3M multiple in a partially redundant form by using a series of small length adders with no carry propagation between the adders. This causes fast generation of 3X operation with require for specified application and area and power will reduced dramatically with our proposed adder structure.

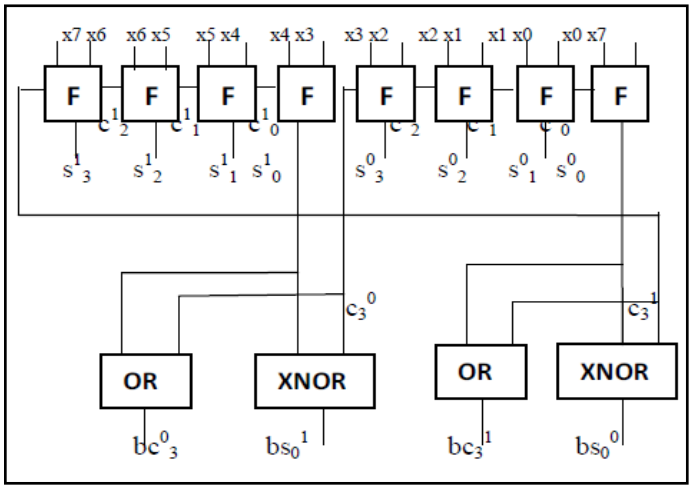

Fig. 2: Generation of Proposed $|3X|_{2n-1}$

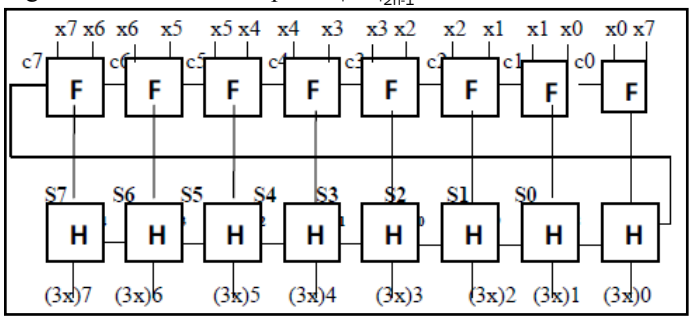

Fig. 3: Generation of Partially- Redundant Simple Multiples

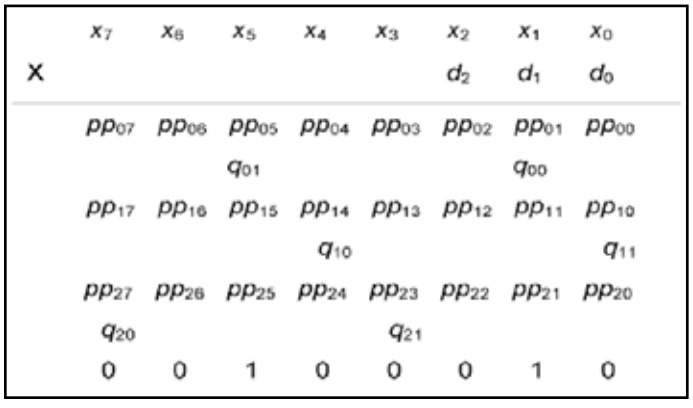

Fig. 4: Modulo-Reduced Partial Products and CC for $|X.Y|_{2n-1}$

From the above we can notice that after generation if partial products along with the intermediate carry generation with number that to be added to the addition process in order to archive the modulo 2^n-1 multiplication , our architecture basic elements are both encoder, both selector adder CSA adder and the parallel prefix adder are used for addition.

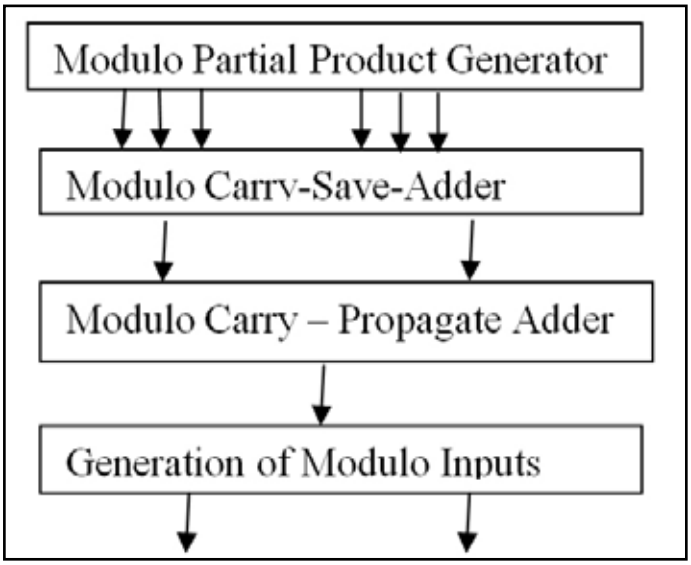

Fig. 5: Modulo (2^n-1) Multiplier Architecture

Our proposed technique should perform multiplication for both hard multiple and soft multiple we can't predict whether the soft multiple or hard multiple our proposed technique should perform hard multiple and soft multiple multiplication depends upon requirements.

A. Radix-8 Booth Encoded Modulo 2^n-1 Multiplication with Partially –Redundant Partial Products

The above architecture consists of Booth Selector (BS) and Booth Encoder (BE) .These two are the important elements of our architecture, the functionality of booth encoder is to select the partial redundant bit for given multiplicand grouping it will generate SEL X, SEL 2X,SEL 3X, SEL4X and sign depends on given input group that will be fed to the input of booth selector. Along with the booth encoder inputs booth selector consists of X, 2X, 3X, 4X as inputs depends upon condition any one of output will be gives as input to the XOR gate. Generally XOR gate deals with complement generation based on sign give by the booth encoder.

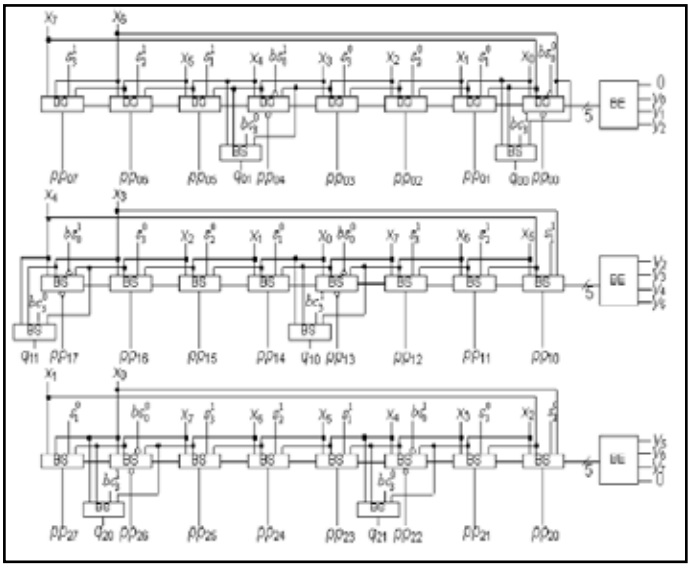

Fig. 6: Modulo-Reduced Partial Product Generation

B. Proposed Adder Structure

After generation of partial products we need to add the partial products, intermediate carry terms and constant number in order to achieve the modulo 2n-1 multiplication For that we are adding with CSA and parallel prefix adder Structure were used in our proposed architecture .detailed description towards the adder structure will described.

A parallel prefix adder can be seen as a 3-stage process namely Pre-computation, Prefix and Post-computation:

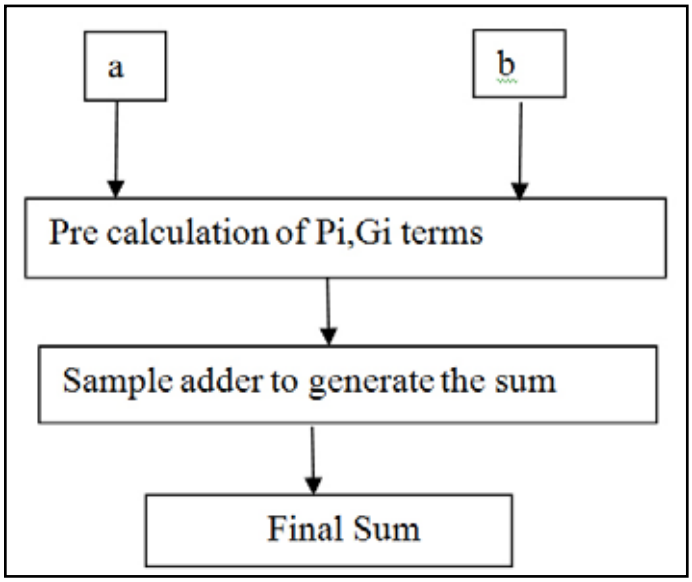

Fig. 8: 8-Bit Parallel-Prefix Structure

C. Pre-Computation

In pre-computation stage, each bit computes its carry generate (g)/propagate (p) signals and a temporary sum as below. These two signals are said to describe how the Carry-out signal will be handled.

D. Prefix

In the prefix stage, the group carry generate/propagate signals are computed to form the carry chain and provide the carry-in for the adder below. Various signal graphs/architectures can be used to calculate the carryouts for the final sum. A few of them are as follows.

E. Post-Computation

In the post-computation stage, the sum and carry-out are finally produced. The carry-out can be omitted if only a sum needs to be produced.

IV. Conclusion

In conclusion, a new approach for multiplication modulo (2^n-1) is proposed. Similar to the binary multiplier, the generation of the partial products is accomplished by Radix-8 modified booth algorithm. The CSA tree is applied to reduce the speed for compression of column size from N to two. To completely utilize the unequal delay of a full adder, an algorithm for delay optimization of the CSA tree is developed, The resultant propagated carry and sum from CSA adder is fed to parallel adder to achieve modulo multiplier output. The proposed approach of 16-bit modulo multiplier exhibits superior performance, in terms of either speed of hardware requirement, in comparison with a recent counterpart for the same purpose. In addition, the proposed multiplier modulo (2^n- 1) shows an extremely regular structure and is very suitable for VLSI implementation.

References

[1] R. Rivest, A. Shamir, L. Adleman,"A method for obtaining digital signatures and public key cryptosystems", Commun. ACM, Vol. 21, No. 2, pp. 120–126, Feb. 1978.

[2] V. Miller,"Use of elliptic curves in cryptography", In Proc. Advances in Cryptology-CRYPTO'85, Lecture Notes in Computer Science, 1986, Vol. 218, pp. 417–426.

[3] N. Koblitz,"Elliptic curve cryptosystems", Mathemat. of Comput., Vol. 48, No. 177, pp. 203–209, Jan. 1987.

[4] National Institute of Standards and Technology [Online]. Available: http://www.csrc.nist.gov/publications/PubsSPs.html

[5] A. K. Lenstra, E. R. Verheul,"Selecting cryptographic key sizes", J. Cryptol., Vol. 14, No. 4, pp. 255–293, Aug. 2001.

[6] C. McIvor, M. McLoone, J. V. McCanny,"Modified Montgomery modular multiplication and RSA exponentiation techniques", IEE Proc. Comput. and Dig. Techniq., Vol. 151, No. 6, pp. 402–408, Nov. 2004.

[7] C. McIvor, M.McLoone, J. V. McCanny,"Hardware elliptic curve cryptographic processors over", IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 9, pp. 1946–1957, Sep. 2006.

[8] D. M. Schinianakis, A. P. Fournaris, H. E. Michail, A. P. Kakarountas, and T. Stouraitis,"An RNS implementation of an __ Elliptic curve point multiplier", IEEE Trans. Circuits Syst. I, Reg. Papers, Vol. 56, No. 6, pp. 1202–1213, Jun. 2009.

[10] J. C. Bajard, L. Imbert,"A full RNS implementation of RSA", IEEE Trans. Comput. – Brief Contributions, Vol. 53, No. 6, pp. 769–774, Jun. 2004.

[11] H. Nozaki, M. Motoyama, A. Shimbo, S. Kawamura, "Implementation of RSA algorithm based on RNS Montgomery multiplication", in Proc. Workshop on Cryptographic Hardware and Embedded Systems, Paris, France, May 2001, pp. 364–376.

Anuradha.K Completed her M.Tech in Electronics and Communications from J.N.T.U, Hyderabad in the year 2011 with distinction. Her interesting subjects are Electronic Devices & Circuits, Pulse and Digital Circuits, V.L.S.I Design and S.T.L.D. Presently she is working as Assistant Professor in E.C.E department in Anurag Engineering College, Kodad, Nalognda Dist, A.P.

Rupesh Kumar Penugonda obtained his B.Tech in ECE from JNTU, Hyderabad in the year 2009, and doing his Master's Degree in "VLSI design" from JNTU, Hyderabad and doing his academic project in "Algorithm of binary image labeling and parameter extracting based on FPGA". At now he has been working as Assistant Professor in the department of Electronics and Communications Engineering at Sana Engineering College, Kodad. His are of interest includes Electronic Measuring Instruments, Antenna Wave Propagation and Digital Image Processing.

Thammishety Narasimharao obtained his B.Tech in ECE from JNTU, Hyderabad in the year 2011 with Distinction. He has done his academic project on "Automatic Flight Control Systems" using the language VHDL. His area of interests are Control Systems, Electronic Devices and Circuits, Micro Controllers and VLSI design.