ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

Design and Implementation Of Modulo Multiplication by Using Radix-8 and Prefix adders

Supriya Sarkar[1] , G.Dilip[2] , Sanghita Deb[3]

M Tech Student[VLSI], Dept .of ECE,RKDF Institute Of Science And Technology, Bhopal, India[1]

M Tech, Dept. of ECE, Jayamukhi institute of technology and science, warangal, India[2]

B.E Student, Dept. of CSE, N.H College of engineering, BAMU, Aurangabad, M.S, India [3]

ABSTRACT*:* 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. This modulo multiplication is frequently used in digital processing systems at the encryption and decryption of PKC(public key cryptography) algorithms are performed by repeated modulo multiplications these multiplications differ from those encountered in signal processing and general computing applications. In this method we are using radix-8M.B.E technique for partial product generator and modulo addition by prefix adders. It is over comes the previous methods delay problem.

Keywords*: M.B.E, modulo arithmetic , radix-8, prefix adders, PKC.*

I.INTRODUCTION

Lossless communication with high secrecy and security is require in present scenario In order to achieve secrecy and security generally we prefer cryptographic technique .Basically there are two types of cryptographic techniques named as public key cryptography , private key cryptography. The cryptography technique converts 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].

A RNS is defined by a set of pair-wise co-prime moduli,$\{L_1,L_2,\ldots L_N\}$such that any integer X within the dynamic range (DR) i.e. $\prod_{i=1}^{N} L_i$,is represented as a N-tuple$\{x_1,x_2\ldots x_N\}$, where x_i is the residue of X modulo [4]-[6].RNS multipliers based on generic moduli have been reported in [1]-[7]. However, special moduli of forms 2^n or $2^n\pm1$ are preferred over the generic moduli due to the ease of hardware implementation of modulo arithmetic functions as well as system-level inter modulo operations, such as RNS-to-binary conversion and sign detection [7]-[9]. The most popular of these special moduli sets is the triple moduliset, $\{2^n-1,2^n,2^n+1\}$ which however has a DR of only 3n bits. It is obvious that the DR of an existing moduli set can be extended by appending many small word- length moduli or a few large word-length moduli. It has been shown that the speed of RNS processor is increasingly dominated by the residue arithmetic operation rather than theone-time forward or reverse conversion [10]

This paper focuses on the design space exploration of arithmetic operation in one of the two special moduli, i.e.,The modulo 2^n-1multiplier design. The Montgomery modulo multiplication, while computing the modular productwithout trial division, is modulus-independent and incapable ofexploiting number theoretic properties of modulo2^n-1arithmeticforcombinational circuit simplification. The propertiesof modulo 2n -1 arithmetic were most effectively exploited for the full adder based implementation of modulomultiplier, the multiplier bits were not encoded, which

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

lead to higher implementation area and longer partial product accumulation time. In [10] and [11], 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, whichismore aggressive than the radix-4 Booth encoding. However, in the radix-8 Booth encoded modulo 2^n-1multiplication, not all modulo-reduced partial products can be generated using the bitwise circular-left-shift operation and bitwise inversion. Particularly, the hard multiple $|+3X|2^n$-1is to be generatedby 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 formodulo 2^n-1multipliers 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 modulo2^n-1multipliers 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 carryoutbits 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 delayof the modulo2^n-1multiplier can be directly controlled by the word-length of the RCAs to equal the delay of the critical modulo multiplier of the RNS. By means of modulo 2^n-1arithmetic properties, we show that the compensation constant that negates the effect of the bias introduced in this processcanbe pre-computed and implemented by direct hardwiring with no delay over head for all feasible combinations of nand k. It is shown that the proposed multiplier lowers the area and powerdissipation of the radix-4 Booth encoded modulo 2^n-1multiplier under the delay constraints derived from various high dynamic range RNS multipliers.

The paper is organized as follows. Section II describes the radix-8 Booth encoding algorithm for modulo multiplication.Proposed radix-8 booth encoded modulo multiplierdesign is described in Section III. In Section IVProposed adder structure,In Section V. simulation results, paper is concluded in Section VIand Fallowed by Acknowledgment and references.

II. RADIX-8BOOTH ENCODEDMODULO 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 orley 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 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. 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 $X = \sum_{i=0}^{n-1} x_i \cdot 2^i$ nd $Y = \sum_{i=0}^{n-1} y_i \cdot 2^i$ represent the multiplicand and the multiplier of the modulo 2^n-1multiplier, respectively. The radix-8 Booth encoding algorithm can be viewed as a digit set conversion of four consecutive overlappingmultiplierbitsy_{3i}+2$y_{3i+1}y_{3i}(y_{3i-1})$to a signed digit, d_i,d_iЄ [-4,4] for i=0,1……⌊n/3⌋The digit set conversion is formally expressed as

$$d_i=y_{3i-1}+y_{3i}+2y_{3i+1}-4y_{31+2}$$

Where$Y_{-1}=Y_n=Y_{n+2}=Y_{n+1=}0$

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 twoin radix_4 .but if we need to reduce the partial products further means you need to go for the radix-8 the general

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

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, d_i, 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 I

Modulo-Reduced Multiples for the Radix-8 Booth Encoding

d_i	pp_i	$\|d_i X\|_{2^n-1}$
0	000....000	0..0
+1	$x_{n-1-3i}\, x_{n-2-3i} \ldots x_0\, x_{n-1} \ldots x_{n-3i}$	X
+2	$x_{n-2-3i}\, x_{n-3-3i} \ldots x_0\, x_{n-1} \ldots x_{n-1-3i}$	CLS(x,1)
+3	+3X	$\|+3X\|_{2^n-1}$
+4	$x_{n-3-3i}\, x_{n-4-3i} \ldots x_0\, x_{n-1} \ldots x_{n-2-3i}$	CLS(X,2)
-4	$\bar{x}_{n-3-3i}\bar{x}_{n-4-3i} \ldots \bar{x}_0\bar{x}_{n-1} \ldots \bar{x}_{n-2-3i}$	CLS($\bar{x}$,2)
-3	-3X	$\|-3X\|_{2^n-1}$
-2	$\bar{x}_{n-2-3i}\bar{x}_{n-3-3i} \ldots \bar{x}_0\bar{x}_{n-1} \ldots \bar{x}_{n-1-3i}$	CLS($\bar{x}$,1)
-1	$\bar{x}_{n-1-3i}\bar{x}_{n-2-3i} \ldots \bar{x}_0\bar{x}_{n-1} \ldots \bar{x}_{n-3i}$	$\bar{x}$
-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

1) 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:

$$|-X|_{2^n-1} = 2^n - 1 - X = \bar{X}.$$

2) Property 2:For any nonnegative integer, the periodicity of an integer power of two over Modulus can be stated as follows:

$$\left|2^{n\cdot s+i}\right|_{2^n-1} = \left|\left|2^{n\cdot s}\right|_{2^n-1} \cdot \left|2^i\right|_{2^n-1}\right|_{2^n-1} = \left|2^i\right|_{2^n-1}.$$

Property 2 ensures that the modulo 2^n-1reduction of binary exponents can be implemented with no logic cost. As a corollary, the modulo 2^n-1reduction 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.

3) Property 3: For j<n

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

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

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

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-left-shift. 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 4X-X.

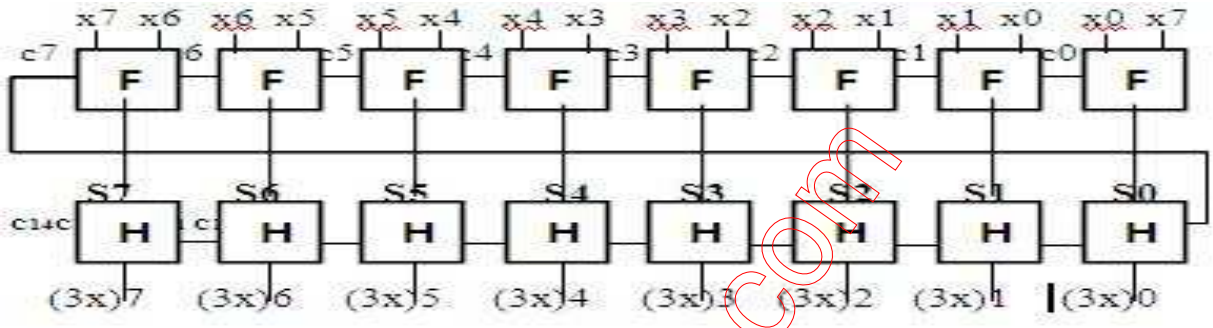

Fig 1: Generation of $|3X|_{2^n-1}$ using two -bit RCAs.

The abovetechnique 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, thelatearrivalof 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 MULTIPLIERDESIGN

Generation of Partially-Redundant Hard Multiple Let and be added by a group ofbit 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

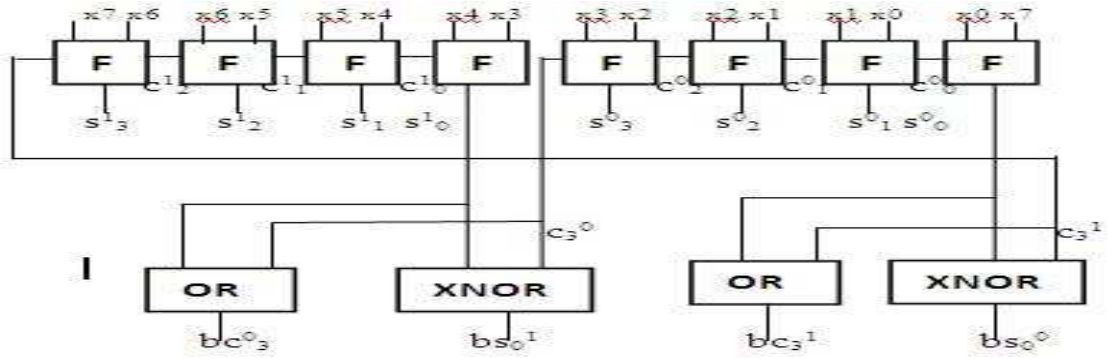

Fig 2: Generation of Proposed $|3X|_{2^n-1}$

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

0	0	0 0	1	0	0	0 0	1	B+0
x_7	x_6	x_5 x_4	$\overline{x_4}$	x_3	x_2	x_1 x_0	$\overline{x_0}$	B+X
x_6	x_5	x_4 x_3	$\overline{x_3}$	x_2	x_1	x_0 x_7	$\overline{x_7}$	B+2X
x_5	x_4	x_3 x_2	$\overline{x_2}$	x_1	x_0	x_7 x_6	$\overline{x_6}$	B+4X

Fig 3: Generation of partially-redundant simple multiples

	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
X						d_2	d_1	d_0
	pp_{07}	pp_{06}	pp_{05} q_{01}	pp_{04}	pp_{03}	pp_{02}	pp_{01} q_{00}	pp_{00}
	pp_{17}	pp_{16}	pp_{15}	pp_{14} q_{10}	pp_{13}	pp_{12}	pp_{11}	pp_{10} q_{11}
	pp_{27} q_{20}	pp_{26}	pp_{25}	pp_{24}	pp_{23} q_{21}	pp_{22}	pp_{21}	pp_{20}
	0	0	1	0	0	0	1	0

Fig 4: Modulo-reduced partial products and CC for $|X.Y|_{2^n-1}$

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

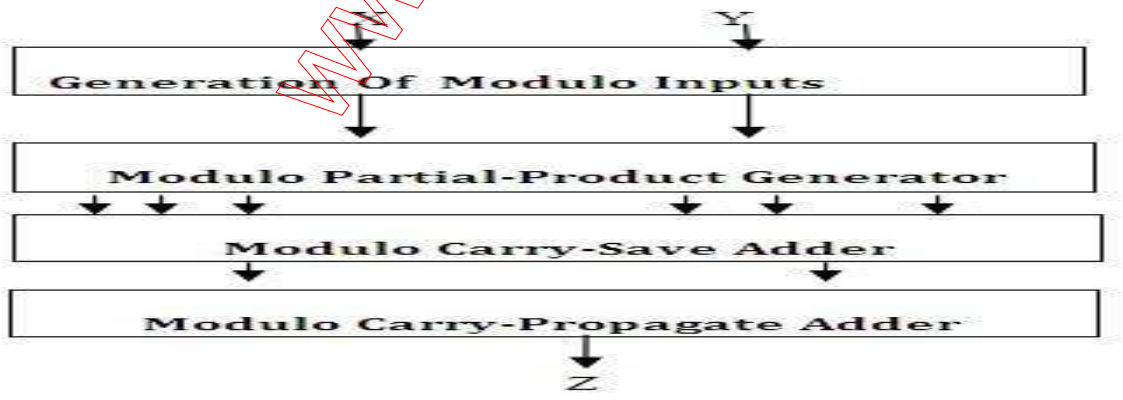

Fig 5: Modulo (2^n-1) multiplier architecture

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.

Radix-8 Booth encoded modulo 2^n-1 multiplication with partially –redundant partial products

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

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

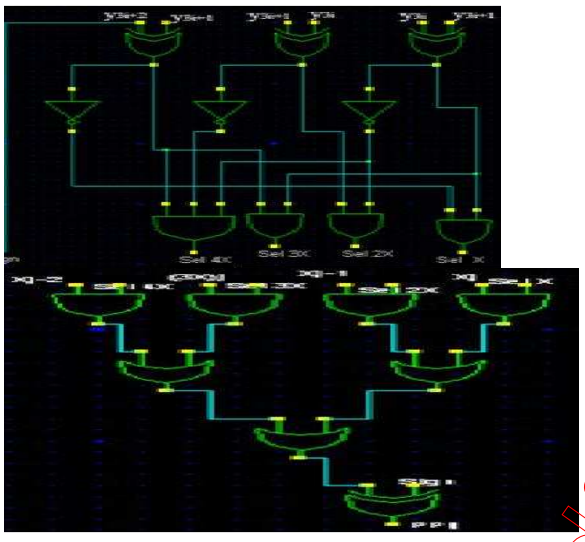

Fig 6: (a) Booth encoder (b) Booth selector

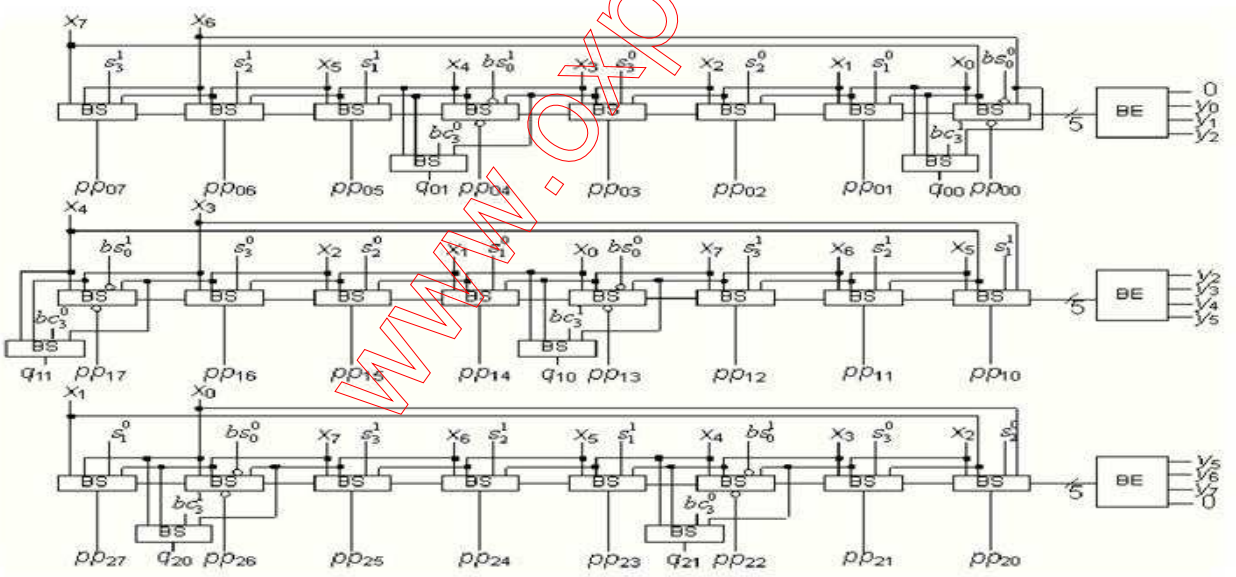

Fig 7: Modulo-reduced partial product generation

IV.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 2^n-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.

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

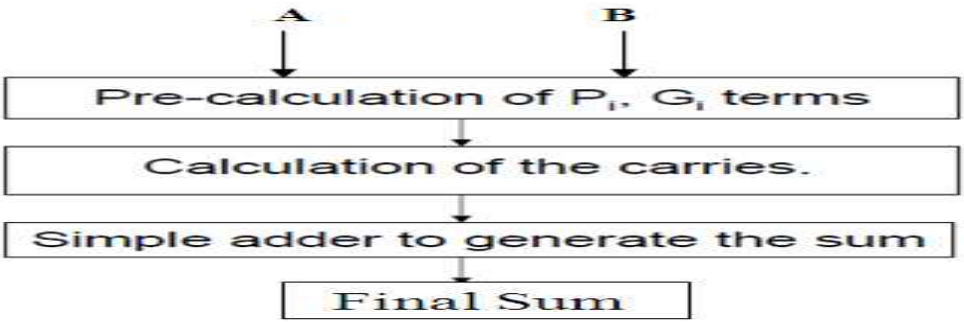

Fig 8:8-bit parallel-prefix structure

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

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.
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 carry-outs for the final sum. A few of them are as follows.
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.

V. SIMULATION RESULTS

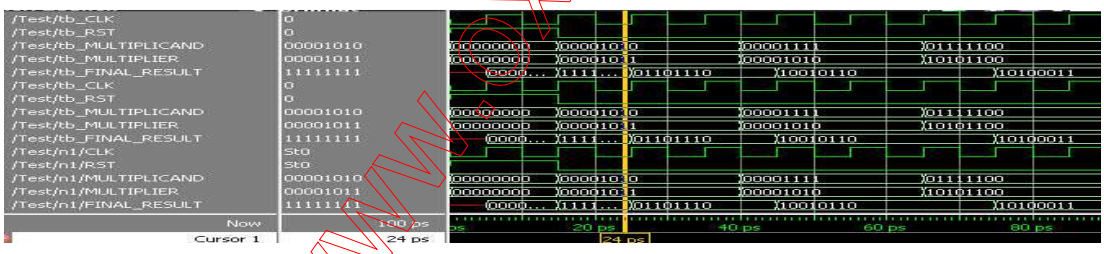

Fig 9: Simulation Results (8-Bit)

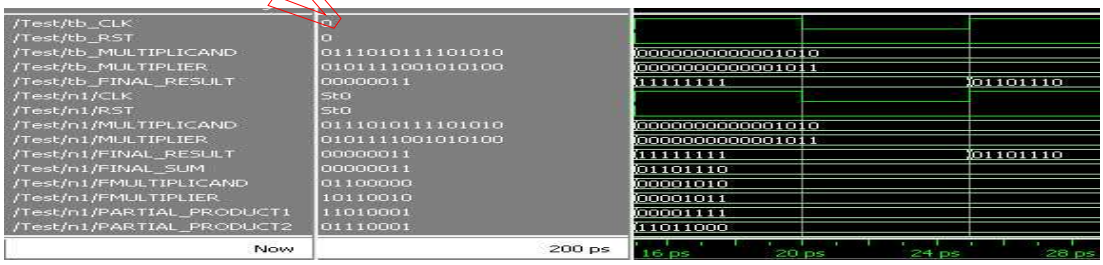

Fig 10: Simulation Results (16-Bit)

V. 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 CSAtree is developed,The resultant propagated carry and sum from CSA adder is fed to parallelfex 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

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics andInstrumentation Engineering
Vol. 2, Issue 5, May 2013

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.

Future Work:

Montgomery modular multiplication algorithm is a well-known method that is employed in efficient modular multiplication architectures and therefore is widely used in GF (p) elliptic curve applications.

he complexity of Montgomery multiplier makes the testing process a big challenge. A methodology for developing testing modules is introduced in. Including a self-testing block in the multiplier's system willbe beneficial and will reduce the time and effort for testing. A self-testing block will perform Montgomery multiplication of hardwired numbers and compare the result with predefined values. A flag bit can be used to indicate an error.

Power dissipation study of the design is also needed in the context of power differential attack. This type of attack on a cryptographic system tries to deduce parameters of the system by observing system's power dissipation. This study would be applicable to show the adequacy of this design approach to hw-power devices, such as portable computers.

More study need to be done to see the effect of applying re-timing technique to radix-2 design, and how the re-timing will affect the performance of the design. Some investigations need to be done to show how the radix-4 design presented in this text can be extended to cover the unified architecture as presented in .The integration of multiplication and exponentiation can be included as part of a hardware co-processor.

REFERENCES

[1] R. Rivest, A. Shamir, and 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.ofComput.*, vol.48, no. 177, pp. 203–209, Jan. 1987.
[4] National Institute of Standards and Technology [Online].Available:http://csrc.nist.gov/publications/Pubs SPs.html[5] A.K.Lenstra and E. R. Verheul, "Selecting cryptographic key sizes,"*J. Cryptol.*, vol. 14, no. 4, pp. 255–293, Aug. 2001.
[5] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques," *IEEProc. Comput.and Dig.Techniq.*, vol. 151, no. 6, pp. 402–408, Nov. 2004.
[6] C. McIvor,M.McLoone, and J. V. McCanny, "Hardware elliptic curve cryptographic processors over G F (p)," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 9, pp. 1946–1957, Sep. 2006.
[7] J. C. Bajard and L. Imbert, "A full RNS implementation of RSA," *IEEE Trans. Comput. – Brief Contributions*, vol. 53, no. 6, pp. 769–774, Jun.2004
[8] H,NozakiM.Motoyama,A.Shimbo,andS.Kawamura ,"implementation of RSA algorithm based on RNS Montgomerymultiplication, "I;e.proc.France,May 2001.pp.364-376 ".
[9] T.Stourait's and V,paliouras,"considering the alternatives low power design ,"EEE circuitDevicesMag,. Vol17, no. pp.22-29.
[10] S.Pontarelli, G.C.Cardarilli,M.Re.and A,Salsano,"Totally fault tolerant RNS based FIRfilter, "in proc.14th IEEE int.on-line testing symp., Rhodes,Greece,jun2008,pp.192-194.

BIOGRAPHY

Supriya Sarkar was born in Tripura, India . He received the B.E degree from SRTMU , Nanded, Maharashtra, India in 2006, he has six year teaching experince , presently pursuing his M.Tech from Rajib Gandhi Proudyogiki Vishwavidyalaya , Bhopal, India. His research interests include VLSI and embedded systems.

Dilip.g was born in A.P, India. He received the M Tech from JNTU , Hyderabad. Presently he is working as a vlsi design engineer .

Sanghita Deb was born in Tripura, India. She is a BE Student of Dr.Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India.