International Journal of Electrical and Electronics Engineers
Vol. No.7 Issue 02, July-December 2015
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

RANDOM NUMBER GENERATOR USING HIGH SPEED AREA EFFICIENT RNS MODULAR ADDER FOR SECURED CRYPTOGRAPHIC APPLICATION

Anjitha Purushothaman[1], Divya S[2]

[1,2]*Department of Electronics and Communication,*

SREE NARAYANA GURUKULAM College of Engineering, Ernakulam, (India)

ABSTRACT

Random number generators is required extensively by many application like cryptography, numerical analysis, signal processing. Traditional methods for random number generator design is based on linear feedback shift registers. Recently a new interest in residue number systems have increased because of its properties suitable for implementations of fast VLSI systems. Modular adder is the vital component in RNS systems. In this paper a random number generator based on adder is proposed to generate random numbers with good randomness properties desirable for cryptographic applications. Moduli set with the form of (1) is best suitable for multichannel RNS processing. The proposed model offers excellent randomness property which is the basic requirement for network security and also offers better area and delay performance.

Keywords: Residue Number System, Parallel Prefix Adder,Modular Adder,Carry Correction,FPGA

I. INTRODUCTION

The demand for new techniques in network security is increasing with the growth of network services in our world. Cryptography plays an important role in network security and it is a vital tool that provides security against various external and internal threats in the network. Data confidentiality is mainly achieved by means of cryptography. The aim of cryptographic techniques is to secure the information so that only the intended parties can read. For transmitting audio and video signals for cable TV, commercial and sensitive data and video conferencing the speed of the cryptographic module is required to be high. However the traditional software implementation of cryptographic algorithms are not efficient in real time applications.

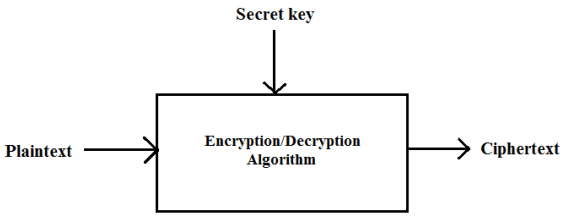

Fig 1 Cryptographic System

The random number generator is a vital cryptographic module widely used for key generation and authentication protocols. The security of such systems completely relies on the excellent randomness property provided by the

generators. Thereby future sequence pattern in the random number sequence cannot be predicted by the observed sequence. The random number generators broadly categorized into true random number generator and pseudo random number generators. The design of cryptographically secure random number generator is extremely difficult. Linear feedback shift register is the traditional method for designing and generating random numbers which uses shift registers. This method has good statistical properties and leads to very efficient hardware implementation. Modular adders can be used to design LFSR for generating random numbers that can offer good randomness property.

Modular adders are the most prime component of residue number system. RNS is an ancient numerical representation system. It is a non weighted numerical representation system and have carry free property in addition and multiplication operations. Modular adder is the key module for RNS based DSP systems. Moduli set with the form of can offer excellent balance among the RNS channels for multichannel RNS processing. This modular adder can be used to design LFSR based random number generator with good randomness property.

II. PREVIOUS WORKS

First a brief survey on modular adders were made and discussed the designing techniques of random number generators based on modular adders. Modular adders can be classified into two types: the general modular adder and the special modular adder and it is based on the form of modulus. In the former adder design the two values A+B and A+B+T should be computed first and one of them is selected as the final output. Bayoumi and Miller [2] proposed a general modular adder for arbitrary modulus by using 2 cascaded binary adders and its delay is the sum of two binary adders. Several modular adders with two binary adders to calculate A+B and A+B+T were proposed subsequently. However this approach offers better delay performance the area consumption is relatively larger and it is twice the binary adder. Reused binary adder configuration [3] is the another type of general modular adder design and was proposed by Dugdale. The drawback of this type of adder is that it will use two operation cycles to perform one modular addition. Subsequently many studies on modular adders were done that have better area and delay performance. A high speed and reduced area modular adder structure for RNS were proposed by Hiasat[6] where any regular carry lookahead based binary adder can be used in the final stage. This structure needs an extra CLA unit to get the carry out bit of A+B+T before the final CLA addition and as a result delay is not reduced significantly. ELMMA [9]algorithm is another popular modular adder design proposed by R.A. Patel. In this adder two carry computation modules for A+B and A+B+T were used and some carry computation units were shared. But in the worst cases almost two independent carry generation modules were used. Dimitris Bakalis [10]proposed fast parallel prefix adder for modulo adders. This architecture is based on parallel prefix carry computation units.

The complexity of special modular adder is much less than that of general modular adder since optimization is possible in special modular adder. The optimization is done according to the modulus. Several architecture for modulo and adder were proposed based on parallel prefix and carry correction.[10][19][20]. Piestrak [4] made a comprehensive study of residue generators and multioper and modular adders. He proposed a design using carry save adder with end around carry and are well suited for VLSI implementation, R.A. Patel [13] first proposed a literature on modulo addition based on carry offset where the carry information of A+B+T is only required to

calculate. The carry information for A+B is obtained by modifying the carries of A+B+T. Even if all the redundant modules of carry computation are eliminated, the structure of carry computation is fixed and can only perform the special modular addition of modulo . In most of the RNS based application ,addition and multiplication intensive systems are used and the main issue is the selection of moduli set accordingly. For such systems residue channels are always expected as many as possible where dynamic range is fixed ie. the wordlength of the residue channels can be reduced inorder to achieve better speed performance. Width of the each channel is also expected as close as possible to get similar critical path delay and thereby fine balance is achieved between each residue channels. The modular adders discussed yet are high performance adders but are not always suitable to construct multichannel RNS that offers fine channel balance ie. It is hard to construct a multichannel that have fine balance with moduli set and . Recently [1] Shang Ma and Jian Ho proposed a modular adder particularly applicable for RNS systems with modulus of the form (1). These adder have outstanding performance in constructing multichannel moduli set with fine balance. L.Li, J Hu and Y Chan recently proposed a general architecture for multiplier. However there were only little discussion and recently a detailed discussion was made [1].

Random number generators are the most vital component used in network security systems like cryptography and encryption techniques. Cryptography is mainly concerned with confidentiality where a message is converted from comprehensible form to incomprehensible form rendering it unreadable by interceptors and eavesdroppers. RNG[7] are basically classified as true and pseudo generators. A common method of producing a pseudo random number generators is to use the output of a linear feedback shift register. Almost all PRNG patterns are reputable and predictable for small cycles. In this paper a new design for random number generator based on modular adder is proposed. This design uses a modular adder which has better area and delay performance. This adder is best suitable for RNS multichannel since a class of modulo is designed instead of a single moduli based on different values of k. Moreover when LFSR design based on adder and conventional modular adder are compared the proposed gives better area and delay performance. Since randomness is the most important requirement in cryptography it is very necessary to design such generator that have good randomness property. This proposed random number generator based on LFSR offers excellent randomness property.

In the rest of the paper a brief introduction of RNS and modular addition are presented in Section II. And Section III introduces the hardware architecture of modulo $2^n - 2^k$ adder. Section IV describes the proposed random number generator design. Performance of different modular adders and LFSR are evaluated and compared in Section V.

III. RNS BASICS & MODULAR ARITHMETICS

RNS arithmetic seems especially suitable for DSP hardware as rapid computation using simple operation of addition subtraction and multiplication can be performed which the basic arithmetic operations of DSP algorithms. RNS arithmetic also has the desirable properties for VLSI implementation of concurrency , modularity and fault tolerant capability.

Residue number system consists of N pairwise relatively prime moduli. A number X is represented as $(|X|_{m1}, |$ where , N>1, GCD , i, j = 1,2,….N and GCD is the greatest common divisor of and . Let A and B be two integers represented by N-tuple word $\{a^0_{RNS}$ and

$\{b_{RNS}^{0}$ respectively in residue number systems. Let ◊ denote the binary operation of addition ,subtraction and multiplication. Then C=A◊B is isomorphic to $C = \{c_{RNS}^{0}, \epsilon$ where ϵ and i . is solely dependent on and , this results in fast, parallel, independent processing within each of the N residue channels.

The modulo m addition for integers A and B in the range of [0,m] is defined as

C = = $\begin{cases} A+B \\ A+B- \end{cases}$ (1)

If C= and the bit width of the modular adder is n bit where n = ie n is the smallest integer no less than . Then eqn (1) can be represented as

C = $\begin{cases} A+B & A+ \\ \langle A+B+T \rangle_{2^n} & A+ \end{cases}$ (2)

Where the correction factor .that is if the carry out bit of A+B+T is '1' then the result of the modular addition is the least significant bits of A+B+T otherwise the result is A+B.

3.1Parallel Prefix Addition

The key element in fast addition of two n-bit operands X and Y is in the reduction of the latency in the carry network. Carry computation can be considered as a prefix problem. This method is widely adopted in binary adder design where each sum bit and carry bit can be calculated with the previous carries and inputs. Prefix based binary adder can be divided into 3 units, the preprocessing unit, prefix computation and sum computation unit.

In the preprocessing unit ,prefix computation is calculated as

$(g_i, p_i) = (a_i b$ (3)

where and represents the carry generation and carry propagation bits respectively. The prefix computation unit is used to compute the carry information used in the sum computation unit. For carry computation group generate and group propagate bits are obtained from and respectively.

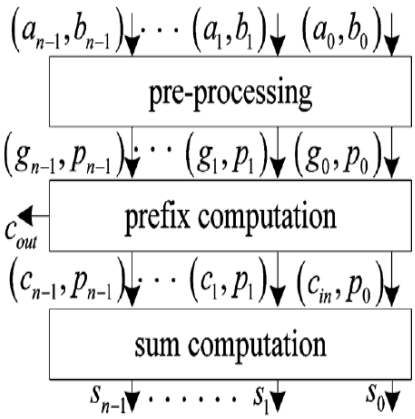

Fig 2. Prefix adder.

International Journal of Electrical and Electronics Engineers
Vol. No.7 Issue 02, July-December 2015
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

$$\begin{cases} (G_{i:i}^0, P_{i:i}^0) = (g_i, p_i) \\ (G_{i:k}^l, P_{i:k}^l) = (G_{i:j+1}^{l-1}, P_{i:j+1}^{l-1}) \bullet (G_{j:k}^{l-1}, P_{j:k}^{l-1}) \\ \qquad = (G_{i:j+1}^{l-1} + P_{i:j+1}^{l-1} G_{j:k}^{l-1}, P_{i:j+1}^{l-1} P_{j:} \end{cases} \quad (4)$$

Where i=0,1,….n-1, 0 , and l represents the stage. There are several well known binary prefix addition structures such as Sklansky, Brent Kung, Kogge Stone, Han Carlson. There structures are usually called prefix trees. After prefix computation carries are obtained i= 0,1,2…..n for bit and computed as

$$\{c_i : \quad (5)$$

In the sum computation unit the carries from prefix computation unit and partial sum from the preprocessing unit are used together to compute the final sum .

$$s_i = p_i \oplus c_i \quad i = 0,1, \quad .(6)$$

IV. NOVEL ADDER

The adder structure used for the design of random number generator is shown in fig…. it is of modulus and composed of four units , preprocessing unit, carry generation, carry correction and sum computation unit. Generally this adder structure can be divided into two general binary adders A1 and A2 as shown in fig…. with carry correction and sum computation unit. This is based on the characteristics of correction T for modulus . any existing prefix structures can be used to compute the carries of A+B+T, . and by correcting the carries we can obtain the final carries used in the final stage. Thus final modular addition result is obtained from and partial sum information from the preprocessing units. The main interesting feature of this architecture is that it avoids the calculation of carry information for A+B+T and A+B separately. Thereby area and delay can be reduced significantly and offers flexible tradeoff between area and delay.

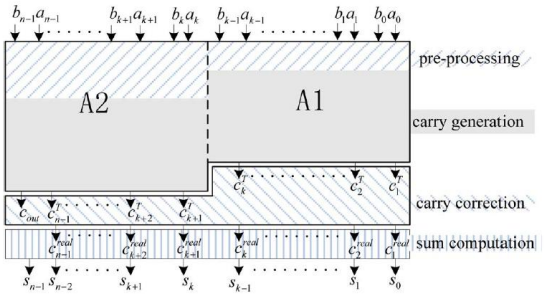

Fig 3. Adder structure.

4.1Pre Processing Unit

This unit computes carry generation and carry computation bits for every bit I, .When modulus *m*= ,then the correction factor is given as $T = : . =$. The binary representation of T is 00….001 00….001. the computation of A+B+T can be performed by A1 and A2 where

International Journal of Electrical and Electronics Engineers
Vol. No.7 Issue 02, July-December 2015
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

A1 and A2 are used for lower-k bits and higher n-k bits addition respectively. Let = 00…001, = 00…001 and the binary representation of A and B are a_{n-1} and b_{n-} respectively.

The operation of adder A1 and A2 can be given as

$$\begin{cases} S_{A1} = a_{k-1} \dots . a_0 + b_{k-1} \dots . b_0 + \\ S_{A2} = a_{n-1} \dots . a_k + b_{n-1} \dots . b_k + T_i \end{cases} \quad (7)$$

Where is the carry out bit of adder A1. This LSB bits of is '1' and all others are '0'. This A1 can be treated as a k-bit adder with lowest carry in bit since is one of the input of A1. Since the LSB bit of is '1' it is considered for the carry generation and carry propagation bits and are computed as

$$\begin{cases} (g_0, p_0) = (a_0 + b_0, \overline{a_0 \oplus b_0}) & i = \\ (g_i, p_i) = (a_i b_i, a_i \oplus b_i) & i = 1,2, \dots . k \end{cases} \quad (8)$$

The second adder A2 adds the constant and the carry out bit from adder A1. So it can be regarded as a 3-input adder with lowest carry in bit. The 3-inputs are and . In this design the number of inputs is reduced from three to two for adder A2 by using Simple Carry Save adder (SCSA). The first stage of SCSA computes for

$$(g_i', p_i') = (a_i b_i, a_i \oplus b_i) \quad (9)$$

This are treated as the inputs of the second stage in SCSA. The carry generation and carry propagation bits for are obtained from this second stage from and . Thus the final output for preprocessing unit are:

$$\begin{cases} (g_k, p_k) = (p_k', g_k') & i = k \\ (g_i, p_i) = (p_i' g_{i-1}', p_i' \oplus g_{i-1}') & i = k+1, \dots . , n-1 \end{cases} \quad (10)$$

The carry out bit of SCSA, is required to compute the carry out bit of A+B+T, . It is calculated as

$$c_{SCSA} = a_{n-1} b \quad .(11)$$

4.2 Carry Generation Unit

This unit uses any existing prefix structure to compute the carries of A+B+T from carry generation and carry propagation bits of pre processing unit. The carry out bit of SCSA is not involved in the prefix computation . is combined with the carry out bit of prefix tree and determines the carry out bit of A+B+T, .

$$c_{out} = c_{SCSA} + c_n^T = c_{SCSA} + G_{n-1:0}$$
$$= c_{SCSA} + G_{n-1:1} + P_{n-1:1} G_{n-1:1}$$
$$= c_{SCSA} + G_{n-1:1} + P \quad (12) \quad .$$

4.3 Carry Correction Unit

The final carries used in the final sum computation stage for each bit is obtained from the unit. In this design the carries for A+B is is obtained by correcting the carries of A+B+T. Hence area is reduced. The relation between and is derived where and are the carry outputs of prefix trees when the lowest carry in is '0' and '1' respectively.

The relationship is given as

= (13)

Where $P_{i:}$ and . This means that can be determined from by simple logic operation. This is the foundation of carry correction for this modular adder. The carry bit of A+B can be obtained with twice carry correction of A+B+T. Whether carry correction is performed or not depends on the carry our bit of A+B+T, .

Carry correction for A1

Since binary representation of T is 00…01 00…01, can be regarded as carry bits of the is modified to determine the carry bits of

= (14)

The carries of A+B+T is corrected under the condition of = 0 . A 2 to 1 MUX is used to perform this action and is used as the control signals. While and are input signals and output is the result of first correction denoted as (i=0,1,….n-2)

$$C_{i+1}^{ci} = \overline{C_{out}} C_{i+1}^{T-1} + C_{out} C_{i+1}^{T-1}$$

From eq (3) = +

= .(15)

Carry correction for A2

is the carry information obtained after the correction of adder A1. Then second correction is performed based on and the carry bits of second correction be . is the correction result of when =0. Otherwise is the carry output of A+B+T. This is the final carry information needed in the sum computation unit.

Second carry correction is performed under the condition that the lowest carry in bit of adder A2 is a constant '1' ie is '1' which is the carry out bit of A1. The propagation bits used in carry correction unit should be computed by and .

$$p_k^{"} = p_k' \oplus c_k^{ci} = \overline{p_k} \oplus c_k \quad i = k$$

$$p_i^{"} = p_i \quad i = k+1,.. \qquad (16)$$

Let group be the group propagate carries then

$$P_{i:k}^{"} \qquad (17)$$

When i=k+1,k+2….n-2 the carries after second correction are

$$c_{i+1}^{real} = \left(\overline{P_{i:k}^{"}} + c_{out}\right) c_{i+1}^{c_1} = \left(\overline{P_{i:k+1} p_k^{"}} + c_{out}\right) c_{i+1}^{c_1}$$

$$= c_{i+1}^{T} (c_{out} + \overline{P_{i:0}}) \left(\overline{P_{i:k+1} p_k^{"}} + \right. \qquad (18)$$

Substituting eqn (15) and (16) in (17), we get

$$c_{i+1}^{real} = c_{i+1}^{T} (c_{out} + \overline{P_{i:0}}) \left(\overline{P_{i:k+1} \left(c_k^{c_1} \oplus \overline{p_k}\right)} + c_{out}\right)$$

$$= c_{i+1}^{T} \left(c_{out} + \left(\overline{P_{i:k+1}} + c_k^{c_1} \overline{p_k} + \overline{c_k^{c_1}} p_k\right) \times (c_{out} + \overline{P_{i:0}})\right)$$

$$= c_{i+1}^{T} \left(c_{out} + \overline{P_{i:0} P_{i:k+1}} + \overline{p_k} c_k^{c_1} + \overline{P_{i:k+1} c_k^{c_1}} p_k + \overline{P_{k-1:0} c_k^{c_1}} p_k\right)$$

$$= c_{i+1}^{T} \left(c_{out} + \overline{P_{i:k+1}} + \overline{p_k} c_k^{c_1} + p_k \overline{P_{k-1}} \ldots\ldots\ldots\ldots (19)\right.$$

Substituting (16) into (19)

$$c_{i+1}^{real} = c_{i+1}^{T} \left(c_{out} + \overline{P_{i:k+1}} + \overline{P_{k-1:0}} \overline{p_k} c_k^{T} + \overline{P_{k-1:0}} p_k \overline{c_k^{T}}\right)$$

International Journal of Electrical and Electronics Engineers
Vol. No.7 Issue 02, July-December 2015
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

$$= c_{i+1}^{T}\left(c_{out} + \overline{P_{1:k+1}} + \overline{P_{k-1:0}}(p_k \right. \quad (20)$$

When $i =$. Thus we get

$$c_{i+1}^{real} = c_{i+1}^{T}\left(c_{out} + \overline{P_{k-1:0}}(p_k \in \right. \quad (21)$$

Therefore combining all equations, the carry bits required by the modular adder are given as

$$c_{i+1}^{real} =$$

$$\begin{cases} c_{i+1}^{T}(c_{out} + \overline{P_{1:0}}) & i = 0,1,\dots,k-1 \\ c_{i+1}^{T}\left(c_{out} + \overline{P_{k-1:0}}(p_k \oplus c_k^{T})\right) & i = k \\ c_{i+1}^{T}\left(c_{out} + \overline{P_{1:k+1}} + \overline{P_{k-1:0}}(p_k \oplus c_k^{T})\right) & i = k+1,\dots,n-2 \end{cases} \quad (22)$$

4.4 Sum Computation Unit

This unit computes the final result for modular adder. It is same as that in prefix based binary adder. is used for final computation of sum with respect to . If is the carry bit of A+B, otherwise it is the carry bit of A+B+T. the partial sum bits of A+B and A+B+T are both required in the final sum computation.

Let and be the partial sum of A+B and A+B+T respectively

$$\begin{cases} p_0^0 = \overline{p_0}, p_0^1 = p_0 & i = 0 \\ p_k^0 = \overline{p_k}, p_k^1 = p_k & i = k \\ p_i^0 = p_i^1 = p_i & i = 1,\dots,k-1,k+1,\dots. \end{cases} \quad (23)$$

Hence

$$s_0 = \overline{c_{out}}p_0^0 + c_{out}p_0^1 = \overline{c_{out}p_0} + c_{out}p_0 = c_{out} \oplus \overline{p_0}$$

$$s_k = c_k^{real} \oplus (\overline{c_{out}}p_k^0 + c_{out}p_k^1) = c_k^{real} \oplus (\overline{c_{out}p_k} + c_{out}p_k \quad :$$

(24)

When $i = 1,\dots,k-1,k$

s_i (25)

Therefore the sum bits for all i are,

$$s_i = \begin{cases} c_{out} \oplus \overline{p_0} & i = 0 \\ c_k^{real} \oplus c_{out} \oplus \overline{p_k} & i = k \\ c_i^{real} \oplus p_i & i = 1,\dots,k-1,k+1,\dots, \end{cases} \quad (26)$$

This and can be obtained at the same time. Therefore there is no extra delay.

International Journal of Electrical and Electronics Engineers
Vol. No.7 Issue 02, July-December 2015
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

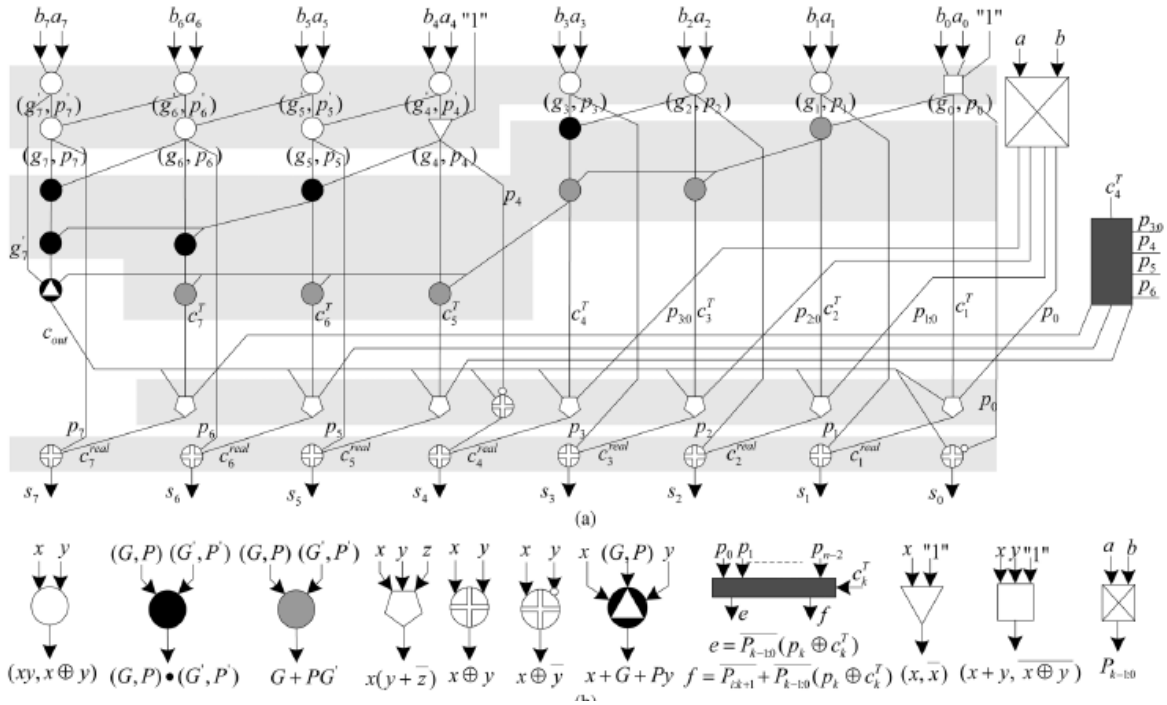

Fig.3.Modulo Adder

V.THE PROPOSED SYSTEM

The data confidentiality in secured communication system is achieved by means of cryptography. This security is maintained by keeping the secret key used for data encryption and decryption confidential. The secret keys should be extremely strong enough so that attackers and eavesdroppers could not predict out and break the cipher text and misuse it.. Therefore we require strong keys. The keys are usually generated by simple random number generators. And the random numbers generated must have excellent randomness properties. Fig 4 shows a conventional random number generator based on linear feedback shift register.

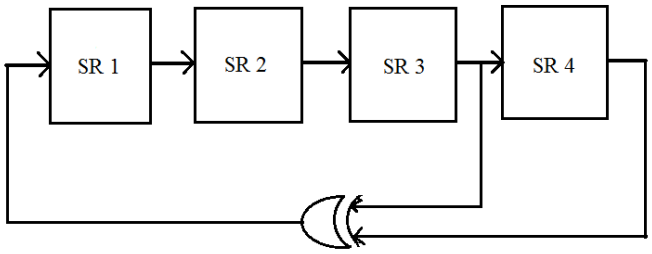

Fig 4. Conventional LFSR

The generator is uses XOR based feedback. The input of shift register is the linear function of previous states. Fig shows the proposed design for random number generator that have excellent randomness properties. In this design the XOR based feedback is replaced by modular adder.

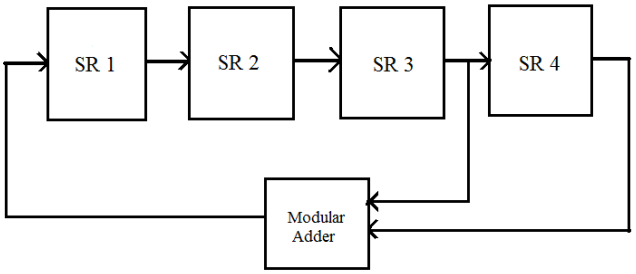

Fig 5 Proposed Random Number Generator.

Whenever the sum exceeds the modulus, the adder produces an exactly different result as sum. By keeping the moduli sets used in the design of modular adders confidential we can produce extremely strong cipher texts rendering it unreadable by interceptors and eavesdroppers. The moduli set is very suitable in constructing balanced multichannel with fixed dynamic range and similar critical path delay. So by employing modular adder we could get generator with excellent randomness property, large dynamic range and better performance which is very suitable for cryptography application.

VI. FPGA IMPLEMENTATION AND PERFORMANCE COMPARISON

6.1 FPGA Implementation

To understand the effectiveness of the proposed design, it is implemented on FPGA of device family SPARTAN 3E. this id synthesized in Xilinx 13.3 version. The simulation result for modular adder and random number generator are shown.

Table shows the device utilization summary and timing report for various adder and the proposed adder.

Fig 6 and Fig 7 shows the simulation waveform of modular adder and random number generator.

Area and delay of the adder influence the area and delay of proposed LFSR design. So it is required to reduce the factors of adder so that we can improve the performance of LFSR correspondingly. In [2] where two binary adders are used to get A+B and A+B+T simultaneously. Since two adders are used the area requirement is much greater than other adders. In [3] where single binary adder is used to compute the result but requires twice the clock cycles. Delay is much increased in this

Table. 1. Logic utilization and Timing report

Modular adders	No of Slices (Available 4656)	No of 4 i/p LUTs (Available 9312)	Delay (ns)
Bayoumi [2]	17	29	14.687
Dugdale [3]	25	41	33.735.
2^(n)-2^(n-2)-1 [13]	12	21	11.831.
2^n-2^k-1	19	33	14.878

International Journal of Electrical and Electronics Engineers
Vol. No.7 Issue 02, July-December 2015
www.arresearchpublication.com
IJEEE
ISSN 2321 - 2055

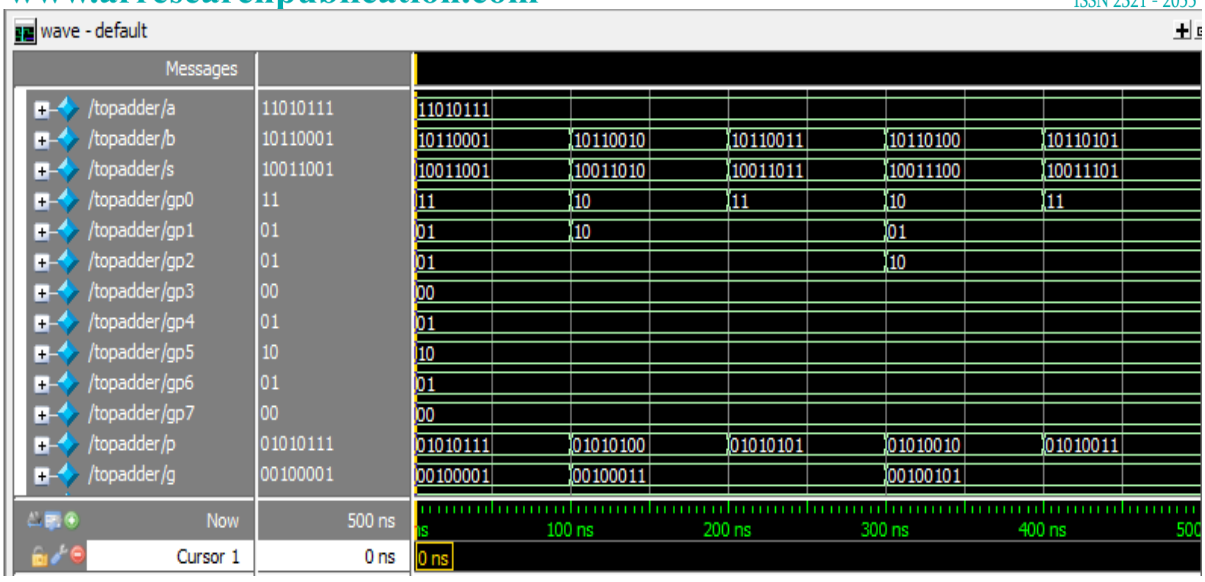

Fig6 . Simulation of Modular Adder.

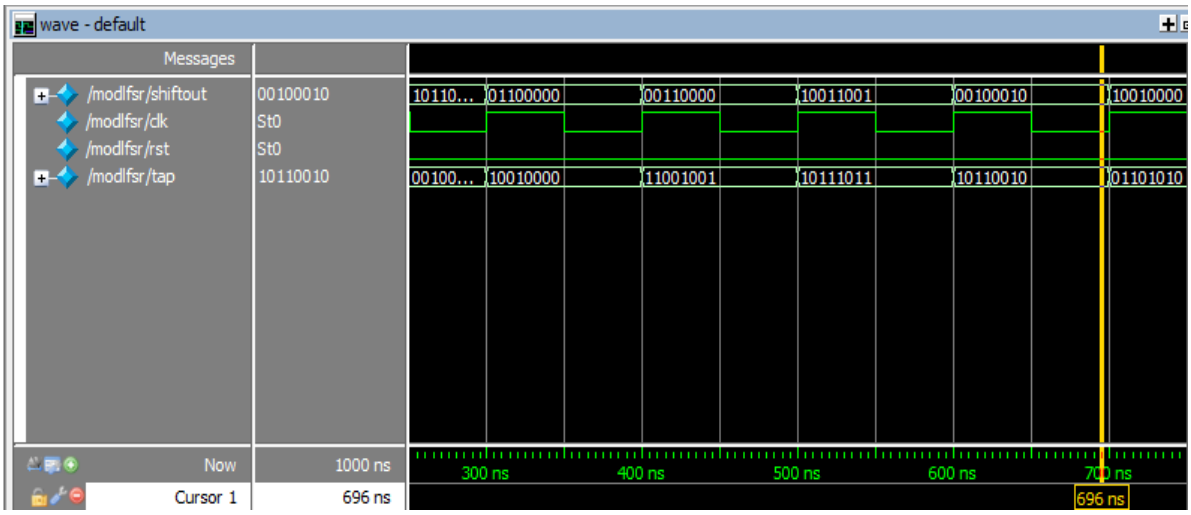

Fig7 . Simulation of Proposed Random Number Generator

scheme. An another scheme of modulo which is a special case of our adder has relatively small delay and give better area delay performance. The adder adder offers a difference set of moduli for difference values of 'k'. That is it is a general design architecture for different modulus. Therefore some optimizations and extra design are applied for such purpose making it suitable for multichannel RNS application. However even if these optimizations are done it still offers better area and delay performance compared to [2][3] [13]. Thus by employing this adder fastest and large dynamic range LFSR can be obtained with excellent randomness property.

VII. CONCLUSION

In this paper a new design approach for random number generator using modular adder is proposed. The proposed design consists of shift registers and modular adders. And modular adder is constructed of four units, preprocessing, carry computation, carry correction and sum computation unit.Since the modular adder use twice

carry corrections instead of carry computation improved the area and timing in VLSI implementation and reduces the redundant units of parallel computation of A+B+T and A+B in the traditional adder. Hence comparison shows the LFSR designed using offer better area and delay performance when compared with traditional adders. The modulus with the form of facilitates the construction of RNS channels with large dynamic and more balanced complexity among each residue channels.

REFERENCES

[1] Shang Ma, Jian Hao Hu and Chen Hao, "A novel modulo $2^n - 2^k - 1$ adder for residue number system" IEEE Transactions On Circuits And Systems—I: Regular Papers. vol. 60, no. 11, pp. 2962–2972, May. 2013

[2] M. Bayoumi, G. Jullien, and W. Miller, "A VLSI implementation of residue adders," IEEE Trans. Circuits Syst., vol. CAS-34, no. 3, pp. 284–288, Mar. 1987.

[3] M. Dugdale, "VLSI implementation of residue adders based on binary adders," IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process., vol. 39, no. 5, pp. 325–329, May 1992.

[4] S. J. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," IEEE Trans. Comput,, vol. 43, no. 1, pp. 68–77, Jan. 1994.

[5] A. A. Hiasat, "High-speed and reduced-area modular adder structures for RNS," IEEE Trans. Comput,, vol. 51, no. 1, pp. 84–89, Jan. 2002.

[6] Cerda J.C., Martinez C.C., Corner J.M. and Hoe, "An Efficient FPGA Random Number Generator Using LFSR and Cellular Automata", IEEE Trans. Circuits & Systems,pp.912-915, Aug 2012.

[7] Erkek E and Tuncer T., "The implementation of ASG and SG Random Number Generator", IEEE International Conference on Science and Engineering,pp 363-367,July 2013.

[8] Liang W. and Long Jing, "A Cryptographic Algorithm Based on Linear Feedback Shift Register", IEEE International Conf on Computer Applications and Systems Modeing, vol. 15, pp 526-529, Oct 2010.

[9] R. A. Patel, M. Benaissa, N. Powell, and S. Boussakta, "ELMMA: A new low power high-speed adder for RNS," in Proc. IEEE Workshop on Signal Processing Systems, Oct. 2004, pp. 95–100.

[10] E. Vassalos, D. Bakalis, and H. T. Vergos, "Modulo arithmetic units with embedded diminished-to-normal conversion," in Proc. 14th Euromicro Conf. Digital System Design (DSD), 2011, pp. 468–475

[11] R. A. Patel and S. Boussakta, "Fast parallel-prefix architectures for modulo addition with a single representation of zero," IEEE Trans. Comput., vol. 56, no. 11, pp. 1484–1492, Nov. 2007.

[12] S. H. Lin and M. H. Sheu, "VLSI design of diminished-one modulo adder using circular carry selection," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 9, pp. 897–901, Sep. 2008.

[13] R. A. Patel, M. Benaissa, and S. Boussakta, "Fast modulo addition: A new class of adder for RNS," IEEE Trans. Comput., vol. 56, no. 4, pp. 572–576, Apr. 2007.

[14] R. A. Patel, M. Benaissa, N. Powell, and S. Boussakta, "Novel power-delay-area-efficient approach to generic modular addition," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1279–1292, Jun. 2007.

[15] P. M. Matutino, R. Chaves, and L. Sousa, "Arithmetic units for RNS moduli and operations," in Proc. 13th Euromicro Conf. Digital System Design: Architecture, Methods and Tools (DSD), 2010, pp. 243–246.

[16] E. Vassalos, D. Bakalis, and H. T. Vergos, "Modulo arithmetic units with embedded diminished-to-normal conversion," in Proc. 14th Euromicro Conf. Digital System Design (DSD), 2011, pp. 468–475.

[17] P. Patronik, K. Berezowski, S. J. Piestrak, J. Biernat, and A. Shrivastava, "Fast and energy-efficient constant-coefficient FIR filters using residue number system," in Proc. Int. Symp. Low Power Electronics
and Design (ISLPED), 2011, pp. 385–390.

[18] S. Ma, J. H. Hu, L. Zhang, and L. Xiang, "An efficient RNS parity checker for moduli set $\{2^n - 1, 2$ and its applications,"
Sci. in China, Ser. F: Inform. Sci., vol. 51, no. 10, pp. 1563–1571, Oct. 2008.

[19] G. Jaberipur and S. Nejati, "Balanced minimal latency RNS addition for moduli set $\{2^n$,," in Proc. 18th Int. Conf. Systems, Signals and Image Processing (IWSSIP), 2011, pp. 1–7.

[20] H. T. Vergos and C. Efstathiou, "A unifying approach for weighted and diminished-1 modulo addition," IEEE Trans. Circuits Syst. II,Exp Briefs, vol. 55, no. 10, pp. 1041–1045, Oct. 2008

[21] H. Vergos, "On the design of efficient modular adders," J. Circuits, Syst., and Comput., vol. 14, no. 5, pp. 965–972, Oct. 2005.

[22] R. Zimmermann, "Binary Adder Architectures for Cell-Based VLSI and their Synthesis," Ph.D. dissertation, Integrated Syst. Lab., Swiss Federal Inst. of Technol., Zurich, 1997.