VLSI HARDWARE MODELING OF MULTIFUNCTION GF ARCHITECTURE FOR CRYTOGRAPHIC DEVICES

RAMYA P, KEERTHI. N.
Chadalwada Ramanamma Engineering College

***Abstract*—**this paper presents a hardware structure for polynomial binary-to-residue number system (PRNS) conversion using parallel field structures.This structure is based only on polynomial multipliers along with GF architecture. Thisconcisework is motivated by the existing RNS binary-to-RNS converters,which are particular inefficient for larger values of n. The experimentalresults obtained for a given dynamic range suggest that theprojected conversion structures are able to drasticallyprogress theforward conversion efficiency, with greater successful hardware modeling. And GF (Galois Field) can highly increase proper selectionand utilization of the polynomial functionsfor high end applications like Cryptography.The proposed logistic technique is simulated and verified by Xilinx tools along with Virtex – 5 FPGA board.

***Index Terms*—RNS,Polynomial multifunction, cryptography, Xilinx tools and Virtex -5 FPGA.**

I. INTRODUCTION

In general advanced and high speed digital system designs includes an application of cryptography,error correction coding, computer arithmetic and logic, DSP and many more which depends on the competent insight of arithmetic over finitefields of the form which is represented as GF(2^n) such that n is an integer. Also it is very familiar that Cryptographic applicationsas per security reason they are of special kind of digital entities and they necessitatehuge integer operands.Competent field multiplication with large operands is crucialfor achieving a gratifying cryptosystem performance, sincemultiplication is the most time- and area-consuming operation.As a result, there is a require for increasing the speed of cryptosystemsalong with the implementation of modular arithmetic with the least possible areaconsequence. An understandable approach to achieve this would be duringparallelization of their operations.

It is very familiar that the Residue Number System (RNS) has modular nature by means whit offers the prospective for swift, parallel reckoning since that it is a carry-free calculation system, also it is to be noted that RNS is a non-weighted number system, which uses remainders to represent numbers. The basic arithmetic operations (add, subtract, and multiply) are easily implemented in RNS and data computations are implemented over operands that are extensively shorter than the resulting RNS Dynamic Range. Typical applications for RNS are in VLSI Digital Signal Processing (DSP), Cryptography, Network Security, High end filtering, convolutions, correlations, and Fast Fourier Transform computations etc.

Basically the RNS modulus set is set up by defining the moduli of (m_i)relatively positive prime integers. A number **P**isrepresented in RNS by its residues $\mathbf{p_i = <X>m_i}$, where $\mathbf{p_i}$is theremainder of the division of **X** by$\mathbf{m_i}$. and then to implement complete RNS system Conversion from weightednumber system to RNS (binary to-RNS or forward conversion), and vice versa (RNS-to-binary or reverse conversion) is required. At the very beginning the development of proposed system on RNS was mainly persistent on the three modulus set {2^n 1, 2^n, 2^n+ 1}, but today the research has been extended to dynamic range of prime integers also such as {2^n k, 2^n, 2^n+ k}, k Z^+ such that the implementation of such method can drastically improves the circuit performance.

The literature survey reveals the fact that various techniques are already proposed like serial method, full parallel method or serial-parallel technique etc, to reduce the weight representation of the number systems, which is the root cause of occupying lot of system memory and consumes un wanted power consumption and finally the architectures for high end process applications becomes much more slower. Therefore a new memory-less standard forward conversion structure fora DR of m = qn-bit, using {2n +/- k} moduli, is proposed, consideringn ≥ 2. The projected approach divides the qninput bits into qinput sets,and computes the particular residue value using modular additionsand constant multiplications so that the idea of constant multipliers does not lead to any excess memory consumption.
A novel method for integrating residue arithmetic in a dual-field Montgomery modular multiplication algorithm for integers and for polynomials in GF (2^n) presented in this paper. The mathematical circumstances that need to be satisfied for a valid RNS/PRNS amalgamation are verified. The plagiaristic architecture is extremely parallelizable and adaptable, as it supports

P RAMYA is a Student of VLSI System Design at
Chadalawada Ramanamma Engineering College, Tirupati (A.P), India.
(Phone: +91 8892717943; email: ramya.p9999@gmail.com.)

N KEERTHI, Assistant Professor, ECE Dept
Chadalawada Ramanamma Engineering College, Tirupathi (A.P), India.
(Phone: +919502079107; email: keerthi499 @gmail.com.)

binary-to-RNS/PRNS and RNS/PRNS-to-binary conversions, Mixed Radix Conversion (MRC) for integers and polynomials, dual-field Montgomery multiplication, and dual-field modular exponentiation and inversion in the same hardware.

DESIGN DEVELOPMENT

To promote low power high performance applications of RNS, Various dynamic macro Techniques are modeled on VLSI. The VLSI architectures developed as follows. The VLSI architecture proposed with this articlefor use of RNS is to reduce on chip memory consumption adopts the macro selection condition which results in to dynamic implementation of major techniques. The literature survey revealed that various state of work has been proposed fir polynomial based multiplication for Cryptography to overcome the limitations of RNS systems.

Montgomery's algorithm is one of the fields of art for dual-field implementations. The Montgomery architectures perform well for RSA key word lengths, by processing word-size data, since RSA key sizes (512, 1024, 2048, etc.) are always multiples of word size. However, in Error Correction Code technique, key sizes are not integer multiples of word size, it sense that, if these architectures were to be used in Error Correction Code techniques, they would require more clock cycles for their implementation thus more power consumption. The above mentioned problem can be over ruled with suggesting as architecture configured at bit-level entity.

The above technique are modeled and implemented by targeting the area and power consumption, the next proceeding section will conclude all about proposed scheme. This proposal is also realized into field programmable gate array (FPGA) prototyping system using Xilinx Viitex-5 unit. The maximum operating frequency of this design is more than 500 MHz.

TABLE 1
SUMMARY OF DESIGN CONSIDERATIONS

Sno	Design consideration	Selection
1	Compiler	Xilinix14.4Vivado
2	Programming Language	Verilog
3	FPGA	Virtex -5
4	Interface	USB

II. MODELING OF GF (2^N) MULTIFUNCTION RNS TECHNIQUES

RNS Representation:
An RNS is defined by a set of relatively prime integers called the moduli. The moduli-set is denoted as $\{m_1, m_2 \ldots m_n\}$ where $m_k$is the k^{th} modulus. Each integer can be represented as a set of smaller integers called the residues. The residue-set is denoted as $\{r_1, r_2, \ldots, r_n\}$ where r_k is the k^{th} residue. The residue $r_k$is defined as the least positive remainder when is divided by the modulus. This relation can be symbolically written based on the congruence: X mod $m_k = r_k$The same congruence can be written in an alternative notation as: $|X|_{m_k} = r_k$ The RNS is capable of uniquely representing all integers that lie in its dynamic range. The dynamic range $\{m_1, m_2 \ldots m_n\}$ is determined by the moduli-set and denoted as

$$M = \prod_{i=1}^{n} m_i$$

The RNS provides exceptionaldepiction for all integers in the range between 0 and. If the integer is greater than , the RNS representation repeats itself. Therefore, more than one integer might have the same residue representation.It is important to accentuate that the moduli have to be relatively prime to be able to exploit the full dynamic range

Mathematical Modeling of RNS conversion:

Allowing for a binary representation of X, with 4n-bit ofDynamic Range, it is required to compute in order toattain the residue modulo {2n k} of X.

$$\begin{aligned}\langle X\rangle_{2^n-k} &= \langle 2^{3n}X_{[4n-1:3n]} + 2^{2n}X_{[3n-1:2n]} + \\ &\quad + 2^{n}X_{[2n-1:n]} + X_{[n-1:0]}\rangle_{2^n-k} \\ &= \langle 2^{3n}X_3 + 2^{2n}X_2 + 2^{n}X_1 + X_0\rangle_{2^n-k} \\ &= \langle k^3X_3 + k^2X_2 + kX_1 + X_0\rangle_{2^n-k}\end{aligned}$$

Where X [k: l] represents the bits l to k of the integer X.
Similarly, the residue calculation modulo {2n + k}, isachieved as

$$\begin{aligned}\langle X\rangle_{2^n+k} &= \langle 2^{3n}X_{[4n-1:3n]} + 2^{2n}X_{[3n-1:2n]} + \\ &+2^nX_{[2n-1:n]} + X_{[n-1:0]}\rangle_{2^n+k} \\ &= \langle 2^{3n}X_3 + 2^{2n}X_2 + 2^nX_1 + X_0\rangle_{2^n+k} \\ &= \langle -k^3X_3 + k^2X_2 - kX_1 + X_0\rangle_{2^n+k}\ .\end{aligned}$$

Taking into deliberation the meticulous cases of modulo{2n 1} and {2n + 1}, conversion from binary-to-RNS can be performed as per given calculation.

$$\begin{aligned}\langle X\rangle_{2^n-1} &= \langle N_3 + N_2 + N_1 + N_0\rangle_{2^n-1} \\ \langle X\rangle_{2^n+1} &= \langle -N_3 + N_2 - N_1 + N_0\rangle_{2^n+1}\end{aligned}$$

PRNS Representation:

Analogous to RNS, a PRNS is definite through a set of a finite pair-wise relatively prime polynomial which can be denoted by the dynamic range of the PRNS. In PRNS each and every polynomial has a unique PRNS representation:

MONTGOMERY MULTIPLICATION:

i.GF (p) arithmetic

Field elements in GF (p) are integers in the range of 0 to p-1 and modulo arithmetic is performed the relevant algorithm is as shown in figure1 which do not contain any division process.

Input: a, b, p, R, R^{-1} /* $a, b < p$
Output: $c \equiv abR^{-1} \mod p$, /* $c < 2p$
1: $s \leftarrow a \cdot b$
2: $t \leftarrow s \cdot \left(-p^{-1}\right) \mod R$
3: $u \leftarrow t \cdot p$
4: $v \leftarrow s + u$
5: $c \leftarrow v/R$

Figure1: Montgomery Multiplication process

ii. GF (2^n) arithmetic:

Field elements in GF (2^n)are polynomials represented as binary vectors of a predefined and fixed dimension, relative to a given polynomial basis the algorithm will work as shown in the figure2.

Input: $a_{\mathcal{T}}, b_{\mathcal{T}}, \left(-p^{-1}\right)_{\mathcal{B}}, Q_{\mathcal{A}}^{-1}, p_{\mathcal{A}}$, /* $a, b < 2p$
Output: $c_{\mathcal{T}}$, /* $c < 2p$ and $c \equiv abQ^{-1} \mod p$
1: $s_{\mathcal{T}} \leftarrow a_{\mathcal{T}} \cdot b_{\mathcal{T}}$
2: $t_{\mathcal{B}} \leftarrow s_{\mathcal{B}} \cdot \left(-p^{-1}\right)_{\mathcal{B}}$
3: $t_{\mathcal{A}} \leftarrow t_{\mathcal{B}}$ /* base conversion step
4: $u_{\mathcal{A}} \leftarrow t_{\mathcal{A}} \cdot p_{\mathcal{A}}$
5: $v_{\mathcal{A}} \leftarrow s_{\mathcal{A}} + u_{\mathcal{A}}$
6: $c_{\mathcal{A}} \leftarrow v_{\mathcal{A}} \cdot Q_{\mathcal{A}}^{-1}$
7: $c_{\mathcal{B}} \leftarrow c_{\mathcal{A}}$ /* base conversion step

Figure2: Polynomial Multiplication Process

iii. The Proposed Dual-Field Montgomery Multiplication Algorithm

To enhance the design vitality the above two algorithms are embedded into each other to design a Dual-Field Montgomery Multiplication unit and the algorithm is shown in the following figure3. The only difference is that integer additions/subtractionsand multiplications are replaced by polynomial ones. And also the degree of input and output polynomials are both less than n the reason it allows the construction of a modular exponentiationalgorithm.

Input: $a_{\mathcal{T}}, b_{\mathcal{T}}, \left(-p^{-1}\right)_{\mathcal{B}}, Q_{\mathcal{A}}^{-1}, p_{\mathcal{A}}$, /* $a, b < 2p$
Output: $c_{\mathcal{T}}$, /* $c < 2p$ and $c \equiv abQ^{-1} \mod p$
1: $s_{\mathcal{T}} \leftarrow a_{\mathcal{T}} \odot b_{\mathcal{T}}$
2: $t_{\mathcal{B}} \leftarrow s_{\mathcal{B}} \odot \left(-p^{-1}\right)_{\mathcal{B}}$
3: $t_{\mathcal{A}} \leftarrow t_{\mathcal{B}}$ /* base conversion step
4: $u_{\mathcal{A}} \leftarrow t_{\mathcal{A}} \odot p_{\mathcal{A}}$
5: $v_{\mathcal{A}} \leftarrow s_{\mathcal{A}} \oplus u_{\mathcal{A}}$
6: $c_{\mathcal{A}} \leftarrow v_{\mathcal{A}} \odot Q_{\mathcal{A}}^{-1}$
7: $c_{\mathcal{B}} \leftarrow c_{\mathcal{A}}$ /* base conversion step

Figure3: Proposed embedded algorithm

Proposed architecture for dual field polynomial modular multiplication presumptuous conversion isbased on a parallel technique. In this techniqueboth technique i and ii are embedded into each This architecture exhibits the modular reduction of each calculation with integrating GF(2^n) and GF(p) modulo unit, instead of adding allterms and reducing then iteratively at the conclusion. The suggested architecture is as shown in figure4.

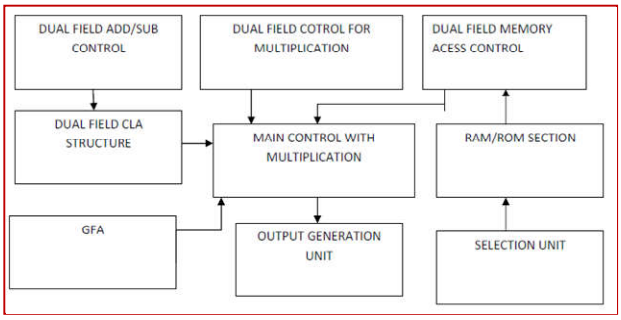

Figure4: proposed architecture

The key design of this architecture that the dual field i.e., embedded of GF(2^n) and GF(p) in to single architecture and are associated in the design description which facilitates the logic transfer from one stage to other for example one bit length to other bit length can be easily avail without changing the complete architecture. The implementation of dual field lead to this proposed system. As shown in figure4.

Front-End Modeling:

This phase of implementation contains the following stages simulation using Xilinx 14.4 Vivado suite, synthesis using Xilinx 14.4 XST and verifying on Virtex – 5 FPGA board.

III. SIMULATION AND SYNTHESIS RESULTS

The Verilog RTL Description of the above article is simulated and synthesized using Xilinx14.4 (ISE-Simulator), implementation of all the above macro encoding techniques are successfully synthesized and verified on Virtex -5 FPGA board and the results are shown in figure5, figure6.

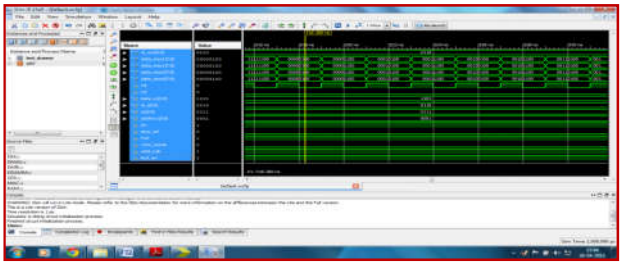

Figure5: Simulation output

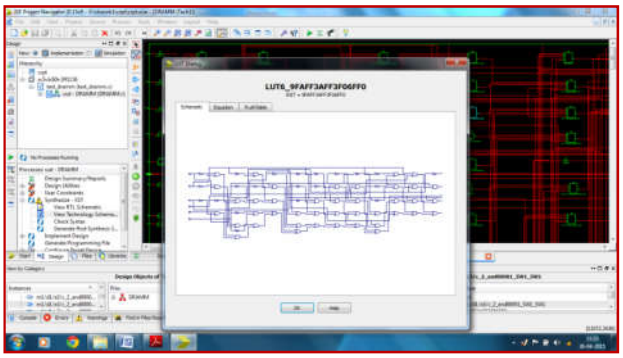

Figure6: Synthesis output

The Synthesized report is summarized in the following Table2.

TABLE 2

SUMMARY OF SYNTHESIS REPORT

S.No	Parameter	Quantification
1	Slice Logic utilization	< 5%
2	Slice logic Distribution	88%
3	IO utilization	12%
4	Specific feature Factor	3%
5	Total Logic Delay(nS)	12.646
6	Total Offset Delay(nS)	25.577
7	Total path Delay(nS)	27.577
8	Real time Compilation(S)	34
9	Total memory (KB)	394220

IV. CONCLUSION

In this paper, we have presented VLSI hardware modeling of dual field GF architecture for high speed application like Cryptography. In addition, their part is usual to increasein potential technology of power and area usage.This paper has realized with Xilinx tools along with Virtex -5 FPGA. Such designs are suggested to exhibits a competitive performance with current work.

ACKNOWLEDGMENT

We would like to thank Dr.V.THRIMURTHULU, for his outstanding support and also we would like to especially express gratitude EDUPLUS-IERC team for their technical advices.

REFERENCES

[1] Multifunction Residue Architecturesfor CryptographyDimitrios Schinianakis*, Member, IEEE*, and Thanos Stouraitis*, Fellow, IEEE*
[2] D. Hankerson, A. Menezes, and S. Vanstone*, Guide to Elliptic Curves Cryptography*. New York, NY, USA: Springer-Verlag & Hall/CRC, 2004.
[3] J.-P. Deschamps*, Hardware Implementation of Finite-Field Arithmetic*. New York, NY, USA: McGraw-Hill, 2009.
[4] R. Lidl and H. Niederreiter*, Introduction to Finite Fields and Their Applications*. New York, NY, USA: Cambridge Univ. Press, 1986.
[5] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, pp. 120–126, Feb. 1978.
[6] S. Kawamura, M. Koike, F. Sano, and A. Shimbo, "Cox-Rower architecture for fast parallel Montgomery multiplication," in *EUROCRYPT' 00: Proc. 19th Int. Conf. Theory and Application of Cryptographic Techniques*, 2000, pp. 523–538.
[7] J.-C. Bajard and L. Imbert, "Brief contributions: A full RNS implementation of RSA," *IEEE Trans. Comput.*, vol. 53, no. 6, pp. 769–774, Jun. 2004.
[8] D. Schinianakis, A. Fournaris, H. Michail, A. Kakarountas, and T. Stouraitis, "An RNS implementation of an elliptic curve point multiplier," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 6, pp. 1202–1213, Jun. 2009.
[9] A. Halbutoğullari and Ç. K. Koç, "Parallel multiplication in using polynomial residue arithmetic," *Design, Codes and Cryptography*, vol. 20, no. 2, pp. 155–173, Jun. 2000.