Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Design and Implementation of DMAS and DM Architectures for Cryptography

Nagaraj B
P.G. Student, E&C & DBIT,
Bangalore, India

Babitha S
Asst. Professor, E&C & DBIT,
Bangalore, India

Abstract— Communication security is very important in day to day life. The proposed algorithms and architectures operate on Galois Fields GF of the form GF (p) for integers and GF (2^n) for polynomials. Alternative number representation systems such as Residue Number System for integers and Polynomial Residue Number System for polynomials are employed., as well as a VLSI architecture of a dual-field residue arithmetic Montgomery multiplier are presented. An analysis of input/output conversions to/from residue representation, along with the proposed residue Montgomery multiplication algorithm, reveals common multiply-accumulate data paths both between the converters and between the two residue representations. A versatile architecture is derived that supports all operations of Montgomery multiplication like input/output conversions, Mixed Radix Conversion for integers and polynomials, in the same hardware. Detailed comparisons with state-of-the-art implementations prove the potential of residue arithmetic exploitation in dual-field modular multiplication.

Keywords— Computation in finite fields, Computer arithmetic, Montgomery multiplication, Parallel arithmetic and logic structures.

I. INTRODUCTION

Everyday transactions from few years ago required some physical existence have been replaced by electronic applications. Users may now use friendly, fast and safe interfaces to perform easily numerous tasks, varying from money transfers using the web and remote health checks to e-learning and e-commerce. Being exposed in an unprecedented number of threats and frauds, safe connectivity for all network-based systems has now become a predicate necessity. The science of cryptography provides the necessary tools and means towards this direction. Cryptographic hardware and software play now a dominant role in e-commerce, mobile phone communications, military applications, and private emails, digital signatures for e-commerce, ATM cards, and web banking, maintenance of health records and so on.

The efficient realization of arithmetic over finite fields of the form GF (2^n) , where $n \epsilon Z$ and $n \geq 1$, or the form GF (p) where p a prime. Cryptographic applications form a special case, since, for security reasons, they require large integer operands. Efficient field multiplication with large operands is crucial for achieving a satisfying cryptosystem performance, since multiplication is the most time and area-consuming operation. Therefore, there is a need for increasing the speed of cryptosystems employing modular arithmetic with the least possible area penalty.

An obvious approach to achieve this would be through parallelization of their operations. In recent years, RNS and PRNS have enjoyed renewed scientific interest due to their ability to perform fast and parallel modular arithmetic. Using RNS/PRNS, a given path serving a large data range is replaced by parallel paths of smaller dynamic ranges, with no need for exchanging information between paths. As a result, the use of residue systems can offer reduced complexity and power consumption of arithmetic units with large word lengths. On the other hand, RNS/PRNS implementations bear the extra cost of input converters to translate numbers from a standard binary format into residues and output converters to translate from RNS/PRNS to binary representations.

A new methodology for embedding residue arithmetic in a dual field Montgomery modular multiplication algorithm for integers in GF(p) and for polynomials in GF(2^n) is presented. The mathematical conditions that need to be satisfied for a valid RNS/PRNS incorporation are examined. The derived architecture is highly parallelizable and versatile, as it supports binary-to-RNS/PRNS and RNS/PRNS-to-binary conversions, Mixed Radix Conversion (MRC) for integers and polynomials, dual-field Montgomery multiplication in the same hardware.

A scalable Montgomery multiplier design methodology for GF(p) is introduced in order to obtain hardware implementations. This design methodology allows using a fixed-area modular Multiplication circuit for performing multiplication of unlimited precision operands. The design tradeoffs for best performance in a limited chip area were also analyzed. The design approaches as to obtain a scalable hardware module. Furthermore, the scalable multiplier described here is capable of performing multiplication in both types finite fields GF(p) and GF(2^n),i.e., it is a scalable and unified multiplier. This show that a unified architecture for multiplication module which operates both in GF(p) and GF(2^n) can be designed easily without compromising scalability, time and area efficiency.

II. RELATED WORKS

In this paper [1] "An algorithmic and architectural study on Montgomery exponentiation in RNS," by F. Gandino, F. Lamberti, G. Paravati, J. Bajard, and P. Montuschi, the design opportunities offered by well-known computer arithmetic techniques are studied, with the aim of developing an efficient arithmetic cell architecture. Furthermore, since the use of efficient RNS bases with a low Hamming weight are being considered with ever more interest, four additional cell architectures specifically tailored to these bases are developed and the tradeoff between benefits and drawbacks is carefully explored. An overall comparison among all the considered algorithmic approaches and cell architectures is presented, with the aim of providing the reader with an extensive overview of the Montgomery exponentiation opportunities in RNS.

In this paper [2] "Hardware-fault attack handling in RNS-based Montgomery multipliers" by D. Schinianakis and T. Stourai tis, hardware-fault attacks, serves as a prominent threat against secure cipher implementations, are deliberately introduced during the operation of cryptographic hardware so that, based on the faulty outputs, secret keys may be recovered. This work focuses on the RSA-CRT algorithm, which, although famous and widely exploited, is known to be vulnerable to hardware-fault attacks. Most of the counter measures, proposed in the literature for this algorithm, are based on number theory techniques that apply at a protocol level. In these cases, security is offered at the cost of extra operations in the RSA-CRT protocol. Unlike these solutions, this work examines the security potential offered by hardware implementations. It attempts to prove that the use of a well-designed, residue-arithmetic, Montgomery multiplier overcomes hardware-fault attack threats, with no need to alter the basic RSA-CRT protocol.

In this paper [3] "Montgomery multiplication using polynomial residue arithmetic," by D. Schinianakis, A. Skavantzos, and T. Stouraitis, a methodology for incorporating polynomial residue arithmetic in Montgomery multiplication algorithm for polynomials in GF(2^n) is been presented. The mathematical conditions that need to be satisfied, in order for this incorporation to be valid are examined and performance results are given in terms of the field characteristic n, the number of moduli elements L, and the moduli word-length w. The proposed architecture is highly parallelizable and flexible, as it supports Polynomial-to-PRA and PRA-to-Polynomial conversions, and also Chinese Remainder Theorem for polynomials, Montgomery multiplication, and Montgomery exponentiation in the same hardware.

In this paper [4] "A RNS Montgomery multiplication architecture," by D. Schinianakis and T. Stouraitis, RNS Montgomery multiplication architecture is a new hardware architectures for Montgomery modular multiplication algorithm is designed. With all the parameters optimized for minimum latency, and this architecture performs only for single Montgomery multiplication in approximately 2n clock cycles, where n is the size of operands in bits. For n clock cycles in this architecture latency is more. Here, they proposed two new hardware architectures that are able to perform the same operation in approximately n clock cycles with almost the same clock period.

In this paper [5] "An improved RNS Montgomery modular multiplier" by Y. Tong-jie, D. Zi-bin, Y. Xiao-Hui, and Z. Qian-jin, here, this paper presents an improved RNS modular multiplication for large operands. Modular multiplication arithmetic plays an important role in public key cryptography. The algorithm uses the Montgomery's method for the Chinese Remainder Theorem, and is performed using a Residue Number System. The number of modular multiplication in the improved one is reduced by percent (n is the number of modulus) comparing to the Bajard's method. Proper hardware architecture for this algorithm is proposed. Finally this work has been verified by modeling it in verilog-HDL.

In this paper [6] "Word-based Montgomery modular multiplication algorithm for low-latency scalable architectures" by M..D. Shieh and W.-C. Lin, this paper presents new word based Montgomery modular multiplication algorithm which can be used to achieve a low-latency scalable architecture for efficient hardware implementations. They showed how to relax the data dependency in conventional word-based algorithms so that a latency of exactly one cycle can be obtained regardless of the chosen word size w ($w > 1$). With the presented operand reduction scheme, the proposed scalable architecture can operate at high speeds and suitable data paths can be chosen for specific applications. Complexity analysis shows that the proposed architecture. Experimental results demonstrate that design has area, speed, and flexibility advantages over related schemes.

III. PROPOSED DESIGN OF DMAS AND DM

A Dual-field Full Adder (DFA) cell is basically a Full Adder (FA) cell consists of half adder and equipped with a field select signal (f_{sel}), that controls the operation mod.

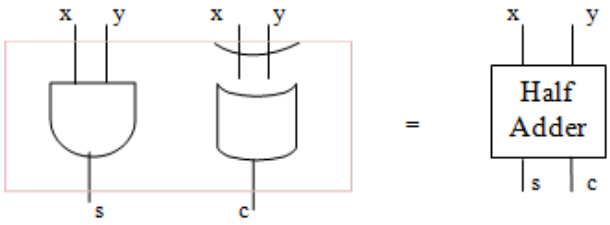

Fig.1 Half Adder

When f_{sel}= 0, the carry output is forced to 0 and the sum outputs the XOR operation of the inputs. As already mentioned, this is equivalent to the addition operation in GF(2n). When f_{sel} = 1, GF(p) mode is selected and the cell operates as a normal FA cell.

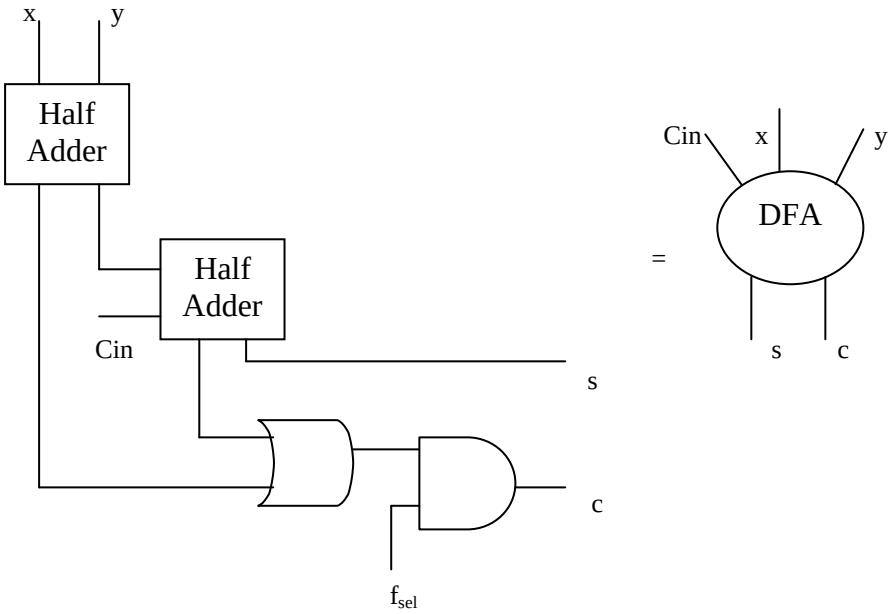

Fig.2 Dual Field Full Adder (DFA)

Dual-field adders in various configurations (carry-propagate, carry-skip, etc) can be mechanized by utilizing DFA cells. In the proposed implementation, 3-level, CLA with 4-bit Carry Look ahead Generator (CLG) groups are employed. An example of a 4-bit dual-field CLA is shown below. The GAP modules generate the signals pi = xi XOR yi , gi = xi AND yi , ai = xi OR yi , and AND gates along with a fsel signal control whether to eliminate carries or not. The carry-look ahead generator is an AND -OR network.

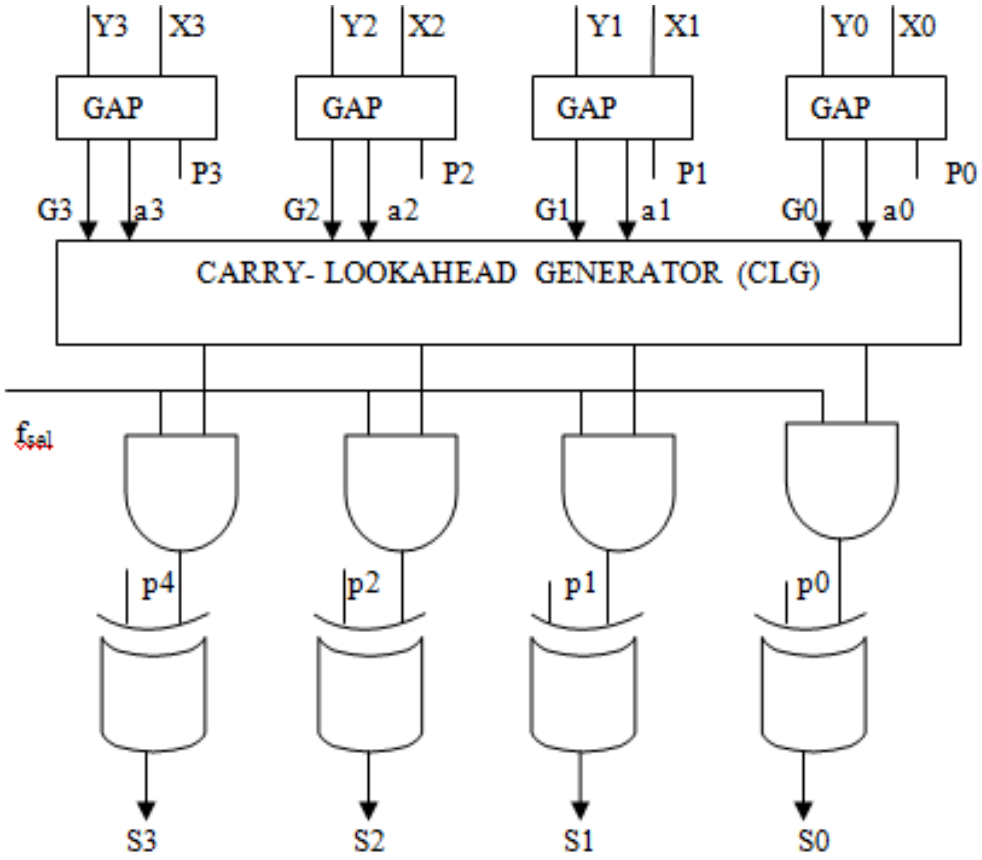

Fig.3 Carry-Look ahead Adder

The MRC-based algorithm avoids the evaluation of the γ factor of the basis of the proposed RNS-based Montgomery multiplication algorithm. The derived algorithm is identical to Algorithm; however the BC algorithm is now based on the modified version of MRC. Comparing the previous approach employing, which requires L (L-1)/2 modular multiplications, the optimized MRC requires only L-2 modular multiplications. The methodology is further extended for the case of GF (2^n).

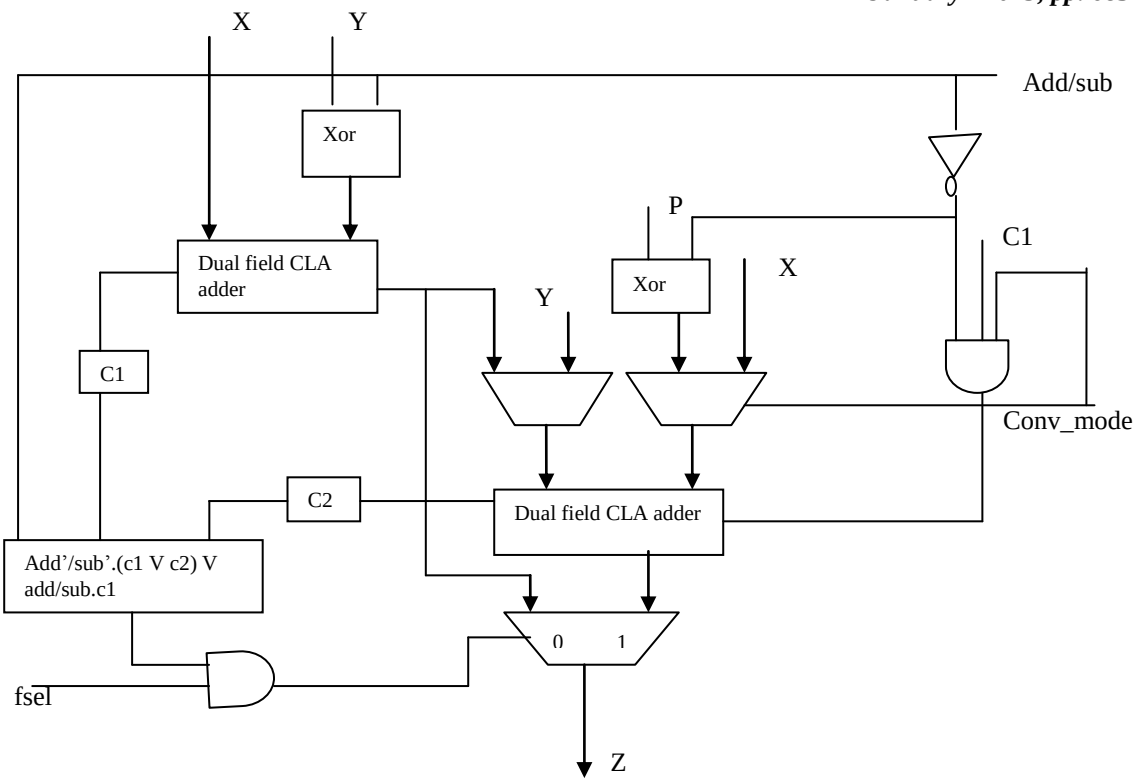

Fig. 4 DMAS structure

A parallel tree multiplier which is suitable for high-speed arithmetic requires little modification to support both fields. Regarding input operands, either integers or polynomials, partial product generation is common for both fields. An AND operation among all operand bits. Consequently, the addition tree that sums the partial products must support both formats. In GF(2^n) mode, if DFA cells are used, all carries are eliminated and only operations are performed among partial products. In GF(P) mode, the multiplier acts as a conventional tree multiplier. A 4 4-bit example of the proposed dual-field multiplier (DM) with output in carry-save format is depicted.

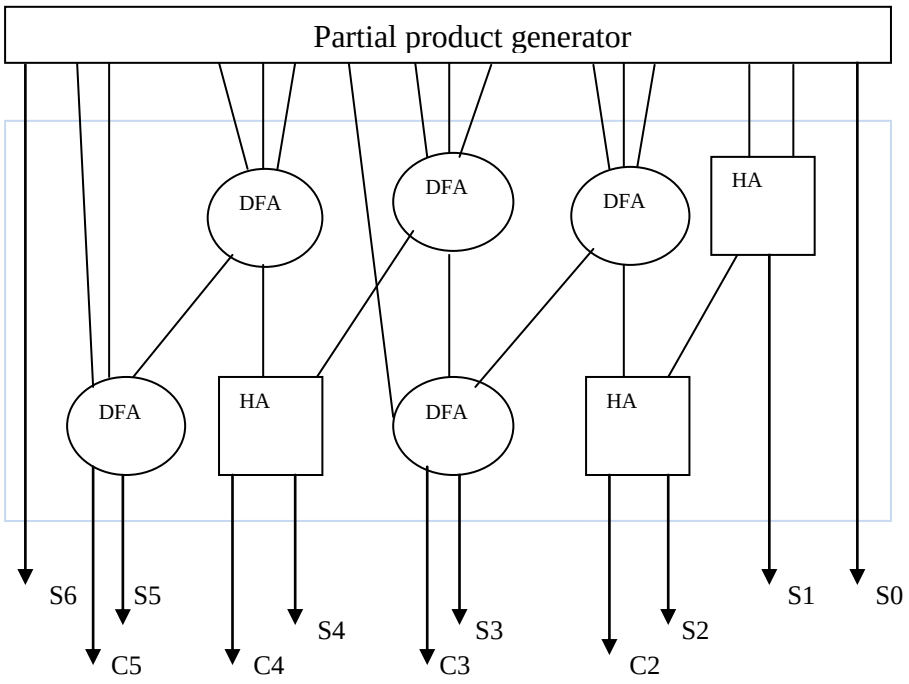

Fig .5 Dual field multiplier

Efficient field multiplication with large operands is crucial for achieving a satisfying system performance, since multiplication is the most time- and area-consuming operation. Cryptographic applications form a special case, since, for security reasons, they require large integer operands.

IV. EXPERIMENTAL RESULTS

A dual-field full-adder (DFA) cell is basically a full-adder (FA) cell consists of half adder and the behavior is similar to normal full adder. The Dual field full adder wave form is shown below. The operation is mainly depends on $f_{sel.}$ The dual field of $G(2^n)$ and G(p) can be performed easily in this method.

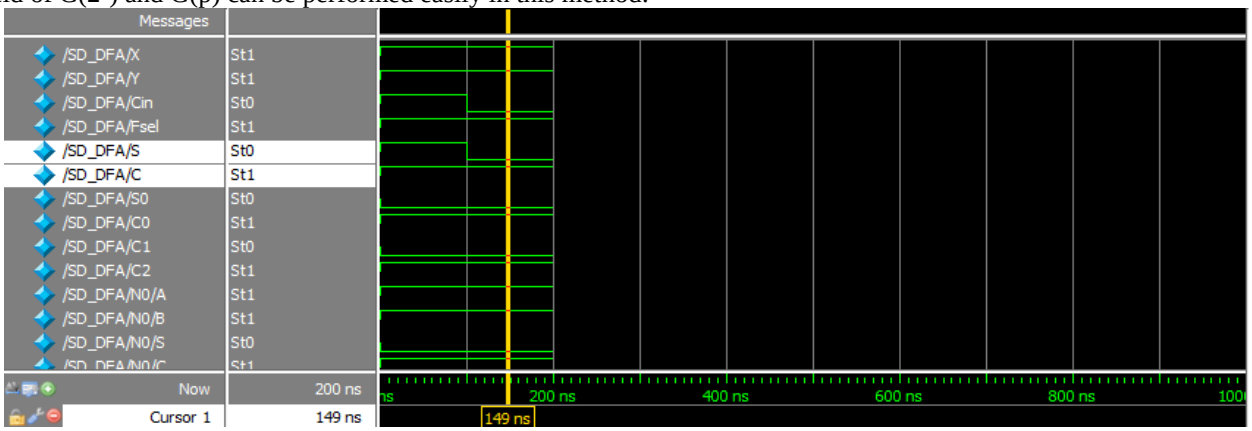

Fig.6 Dual Field Full Adder

An example of a 4-bit dual-field CLA is shown below. The GAP modules generate the signals pi = xi XOR yi , gi = xi AND yi , ai = xi OR yi , and AND gates along with a fsel signal control whether to eliminate carries or not. The carry-look ahead generator is an AND -OR network. This CLA is useful to design the DMAS whose waveform is shown below.

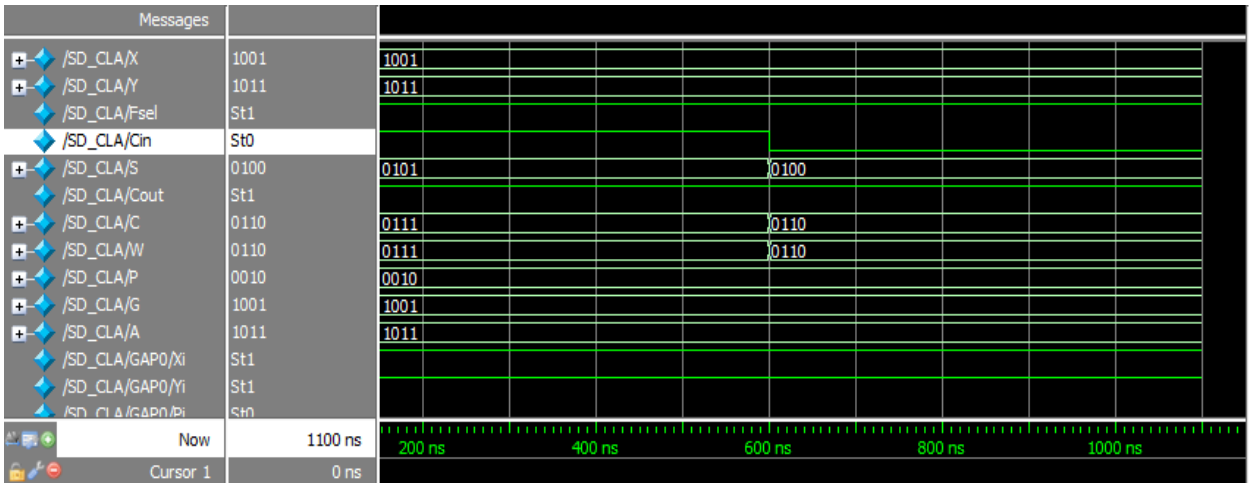

Fig.5 CLA waveform

The work presented in this paper the Dual-field Full Adder consists of When f_{sel}= 0, the carry output is forced to 0 and the sum outputs the XOR operation of the inputs. When f_{sel} = 1, GF(p) mode is selected and the cell operates as a normal FA cell. The duel full adder result is shown below.

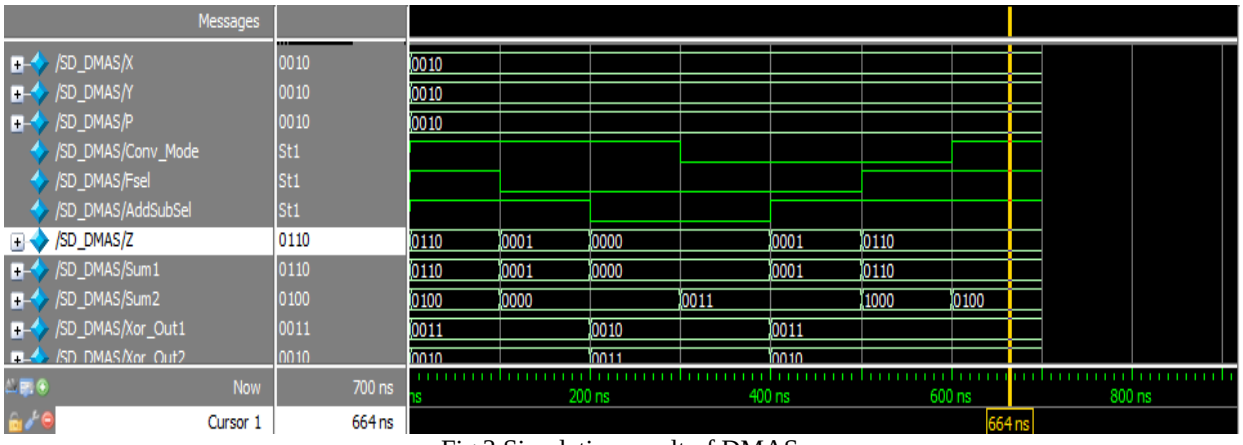

Fig.3 Simulation result of DMAS

The work presented in this paper, the DM multiplier which is suitable for high-speed arithmetic requires little modification to support both fields. A 4 4-bit example of the proposed dual-field multiplier (DM) with output in carry-save format is depicted.

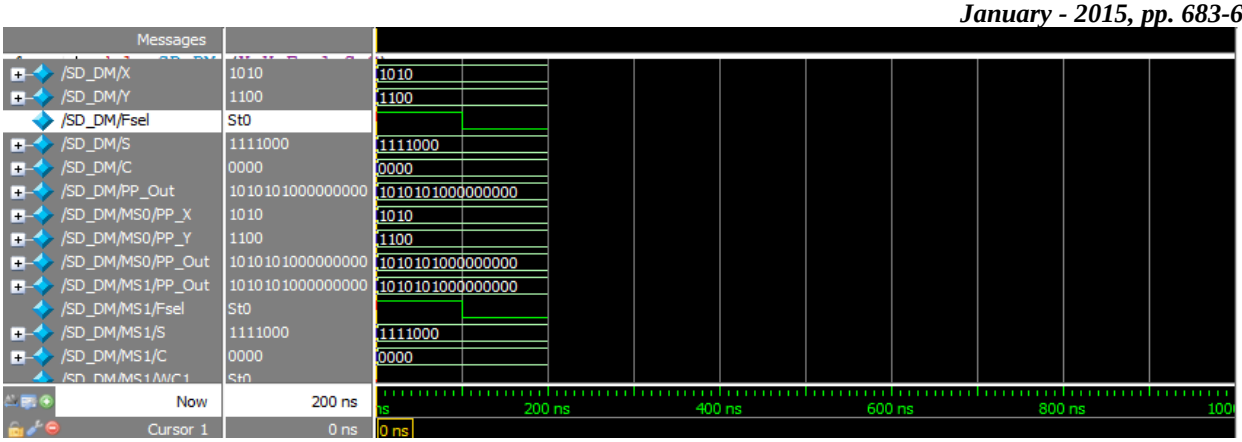

Fig.4 Result of DM

V. CONCLUSIONS

The design methodology for incorporating RNS and PRNS in MMM in GF(p) or GF(2n) respectively was subsequently presented. An analysis of input/output conversions to/from residue representation, along with the proposed residue Montgomery multiplication algorithm, revealed common multiply-accumulate data paths both between the converters and between the two residue representations. A novel versatile architecture was derived that supports all operations of MM in GF(p) and GF(2n), input/output conversions, MRC for integers and polynomials, dual-field modular exponentiation and inversion in the same hardware. Detailed comparisons with state-of-the-art implementations proved the potential of residue arithmetic exploitation in dual-field modular multiplication.

ACKNOWLEDGMENT

Wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files, which have been used in the preparation of this paper.

REFERENCES

[1] F. Gandino, F. Lamberti, G. Paravati, J. Bajard, and P. Montuschi, *"An algorithmic and architectural study on Montgomery exponentiation in RNS,"* IEEE Trans. Comput., vol. 61, no. 8, pp. 1071–1083, 2012.

[2] D. Schinianakis and T. Stouraitis, *"Harware-fault attack handling in RNS-based Montgomery multipliers,"* in Proc. IEEE Int. Symp. Circuits and Systems, 2013.

[3] D. Schinianakis, A. Skavantzos, and T. Stouraitis, *"Montgomery multiplication using polynomial residue arithmetic,"* in Proc. IEEE Int. Symp. Circuits and Systems, 2012, pp. 3033–3036

[4] D. Schinianakis and T. Stouraitis, *"A RNS Montgomery multiplication architecture,"* in Proc. IEEE Int. Symp. Circuits and Systems, 2011, pp. 1167–1170.

[5] Y. Tong-jie, D. Zi-bin, Y. Xiao-Hui, and Z. Qian-jin, *"An improved RNS Montgomery modular multiplier*," in Proc. 2010 Int. Conf. Computer Application and System Modeling (ICCASM), 2010, vol. 10, pp. V10-144–V10-147.

[6] N. Guillermin, *"A high speed coprocessor for elliptic curve scalar multiplications over,"* in Cryptographic Hardware and Embedded Systems, CHES 2010, 2010, pp. 48–64, Lecture Notes in Computer Science 6225

[7] M.-D. Shieh and W.-C. Lin, *"Word-based Montgomery modular multiplication algorithm for low-latency scalable architectures,"* IEEE Trans. Comput., vol. 59, no. 8, pp. 1145–1151, aug. 2010.

[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.

