FPGA Implementation of Area Efficient Finite Field Redundant Multipliers

B. S. T. Ragu[1] S. Maragatharaj[2] N. Vijayanantham[3]
[1,3]PG Student (VLSI Design) [2]Assistant Professor
[1,2,3]Department of Electronics & Communication Engineering
[1,2,3]Knowledge Institute of Technology, Salem, Tamilnadu, India

Abstract— Redundant Basis (RB) multipliers over Galois Field (GF) is a basic operation in modern cryptographic systems. Because of their negligible hardware budget for squaring and modular reduction. An improved recursive decomposition algorithm has been applied for Redundant Basis multiplication to get the high-throughput digit-serial presentation. Over effective forecast of Signal-Flow Graph (SFG) of the planned algorithm, an extremely regular Processor-Space Flow-Graph (PSFG) is obtained. The analysis and synthesis outcomes confirm the efficiency of planned multipliers over the present ones. The synthesis outcomes for Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC) consciousness of the planned designs and present designs are compared. The high-throughput structures are the best among the corresponding designs, for Field Programmable Gate Array and ASIC implementation.

Key words: Application-Specific Integrated Circuit, Galois Field, Finite Field Multiplication, Field Programmable Gate Array, High-Throughput, Digit-Serial, Redundant Basis

ABBREVIATIONS

- Elliptic Curve Cryptography (ECC)
- Ultra-Wide Band (UWB)
- Redundant Binary Modified Partial Product Generator (RBMPPG)
- Add-Accumulation (AA)
- Error-Correcting Word (ECW)
- Field Programmable Gate Array (FPGA)

I. INTRODUCTION

The advancement of VLSI in the electronics industry has attained a phenomenal development during the recent past mainly due to the fast advances in integration technologies and extensive systems design. The number of submissions of integrated circuits in high-performance to the beginning of VLSI. Typically, the required computational power of these applications is the powerful force for the fast growth of this field. One of the most significant characteristics of information facilities is their increasing need for very high dispensation power and bandwidth.

The traditional way of applying the consistent algorithms is software, consecutively on general-purpose processors or on digital-signal processors. Never the less, in some cases the time limitations cannot be met with instruction set processors, and exact hardware must be considered, that is, circuits exactly designed for performing those complex algorithms they implement the specific computation primitives of the algorithms and profit from their characteristic parallelism. Distant from the Application-Specific Integrated Circuits (ASICs) solution, additional technology at hand for developing exact circuits is constituted by Field-Programmable Gate Arrays (FPGA).

Finite field multiplication above Galois Field is a basic operation regularly encountered in modern cryptographic structures such as the Elliptic Curve Cryptography (ECC) and error control coding. Elliptic Curve Cryptography:

- It is applicable for encryption, decryption and security purpose.
- Security provided by the keys of small size.

Moreover, multiplication over a finite field can be used additionally to perform other field operations, e.g., division, exponentiation, and inversion. Effective digit-level serial/parallel designs are desirable to get high throughput finite field multiplication above Galois Field based on Redundant Basis.

Galois Field is very important in many application examples: Coding theory and cryptosystem. It is represented in Polynomial, Triangular, Dual and Normal basis. Maximum of the real-time applications, hence, need hardware implementation of finite field arithmetic operations for the welfares like low-cost and high-throughput rate.

- Correct data transmitted by every second.
- We aim at presenting effective digit-level serial/parallel designs for high throughput finite field multiplication above Galois Field based on Redundant Basis.

Digital serial multipliers used in the special purpose processors. Digit pipelining can be used to maintain high data rates.

K..Aerts, et.al [1] describes Elliptic curve cryptography (ECC) offers high security with shorter keys than the other public-key cryptosystem and it has been successfully used in security critical embedded systems. It is the first hardware implementation that uses the recently introduced lambda coordinates and the Galbraith-Lin-Scott (GLS) technique with fast endomorphisms.

M.F. Albrawy, et.al [2] presents Elliptic Curve Cryptography (ECC) offers a smaller key size without sacrificing security level. A brief survey on applying the main equation of Elliptic Curve (EC) with different values of the coefficients a and b. The Value of a and b implemented on FPGA according to correlation results between plaintext image and ciphertext image on MATLAB. This EC equation will be applied to an ultra-wide band (UWB) system to secure transmission of data in a wireless channel.

Jiafeng Xie, et.al [3] describes By identifying suitable cut-sets, we have improved the PSFG properly and performed efficient feed-forward cut-set retiming to derive three novel multipliers which not only involve significantly less time complexity than the present ones but also require fewer area and less power consumption related to the others. Both theoretical analysis and synthesis results confirm the efficiency of proposed multipliers over the existing ones.

Xiaoping Cui, et.al [4] states a new Redundant Binary modified partial product generator (RBMPPG) is proposed; it removes the extra error-correcting word (ECW) and hence, it saves one redundant basis partial product (RBPP) accumulation stage. Therefore, the projected RBMPPG generates fewer partial product rows than a conventional RB MBE multiplier.

P.S Mallick, et.al [5] proposed multiplier reduces the power consumption by skipping the unwanted switching activity when the multiplicand operand consists of a number of zeros. This work approximation the power, delay, and area of fixed-width multipliers and presences that the planned array multiplier consumes less power and reduced delay compared to the conventional fixed-width array multipliers

J.Xie**,** et. al [6] presents the Multiplication above is a basic field operation which is frequently encountered in elliptic curve cryptography and error control coding. A number of architectures have been projected for polynomial-basis finite-field multiplication above in hardware platforms. Systolic structures are compressed and can be used in resource-constrained schemes, but cannot be used for high-speed users due to their small throughput rates.

G. Drolet [7] describes the general architecture is additionally optimized for equally spaced polynomials, trinomials, and penta nomials. Finite field arithmetic I operations are used in error control coding and VLSI testing. The basic arithmetic operations above the finite field GF addition is easily understood using m two-input XOR gates with multiplication is expensive in terms of gate count and time delay.

P. K. Meher [8] explains Novel systolic and super-systolic architectures are presented for polynomial basis multiplication over GF. The hardware complexities of planned super-systolic designs are closely three times that of the present bit-parallel structures but suggestion very high throughput associated with the others for large values of. For the field orders, the projected structures offer, respectively, ten and eleven times more throughput than the others.

N.R.Murthy, et.al [9] explains the efficient computation of the arithmetic processes in finite fields is linked to the specific ways in which the field essentials are represented. The common field representations are a polynomial basis representation and a normal basis representation. It is well known appropriate for hardware implantation.

C.Y. Lee [10] describes Cryptographic functions, such as key conversation, signing, and verification, require an important amount of computations in the finite field GF. The excellent of the representation plays a significant role in determining the complexity of a finite field arithmetic unit. In a normal basis multiplication scheme implemented in bit-parallel fashion using m identical logic blocks.

H.Wu et. al [11] describes the bit-parallel systolic structures with bit-level pipelining suggestion very high throughput rates at higher hardware cost. The digit-serial and serial/parallel systolic designs offer scalability of hardware and throughput; the numbers of pipelining latches among the PEs in the present structures add substantial complexity to the complete area and time complexity of the system.

L. Song, et. al [12] States to design DSP data paths for finite field arithmetic multiplication operation. To integrate finite field multiplication into DSP data path, one can each combine finite field multiplier with the present binary multiplier or add a distinct data path for finite field arithmetic. One combined binary and finite field arithmetic data path architecture have been proposed and Shared data path has the advantage of decreasing bus load.

I. Blake, et.al [13] presents the Finite fields occur frequently in error-correcting codes and cryptography. Inside the integrated circuit, components are denoted by binary digits. The complexity relating both times as well as an area of employing all the circuits' dependence on the representation.

K. K. Parhi, et.al [14] gives effective outcome among identified bit-parallel multipliers. The non-conventional base has practical properties such as cost-free squaring an efficient software implementation and a simple base conversion to a polynomial basis. The representing the field essentials with respect to the polynomial basis, we consider bit-parallel architectures for multiplication above the finite fields GF.

G. Drolet [15] terms the binary arithmetic operations are much more concentrated than finite field arithmetic operations. Shared data path will continuously add speed and power consumption above to those DSP algorithms with leading number of binary arithmetic operations. Users can power it on or off as desirable and operate it with preferred clock frequency, supply voltage, etc., based on the computation strength and time criticalness.

II. METHODOLOGY

Redundant basis multiplier (RB) has a main impact on the presentation of the arithmetic circuits. A well-organized recursive decomposition arrangement for digit-level RB multiplication and founded on that for resulting parallel algorithms for high throughput digit-serial multiplication. The existing system of using two different high-speed architectures i.e., regular 2-dimensional Signal-flow Graph (SFG) array consumes power and introduces delay. Proposed system introduces a 1-dimensional Processor-space Flow Graph (PSFG) to offset the use of two multipliers and thereby improve the throughput rate. The proposed digit-serial multiplier include less power- delay complexities than the corresponding existing designs.

A. Signal Flow Graph (SFG)

Redundant basis (RB) multiplication can be represented by the two-dimensional SFG. It consisting of Q parallel arrays, where each array consists of (P-1) bit-shifting nodes(S-node), P-multiplication nodes (M-nodes) and (P-1) addition nodes (A-nodes).

There are two types of S nodes (S-I node and S-II node). S-I node performs circular bit-shifting by one position and S-II node performs circular bit-shifting by Q-positions for the degree reduction requirement. Each of the M-nodes performs an AND operation of a bit of serial input operand A with the bit-shifted form of operand B, while each of the A nodes performs a XOR operation.

The desired product word is obtained after the addition of the Q-parallel output of the arrays. The functions of nodes of PSFG are the same as those of corresponding nodes in the SFG and except an extra add-accumulation (AA) node to execute the accumulation operation for Q-cycles to yield the desired result.

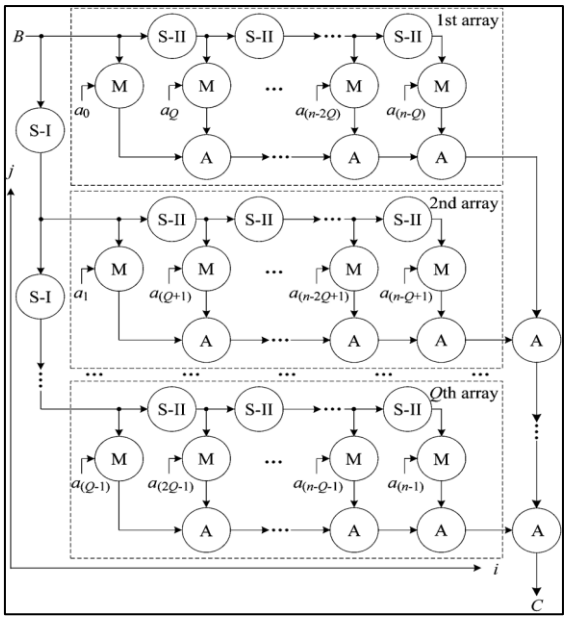

Fig. 1: Signal Flow Graph (SFG)

Assuming x to be a primitive n-th root of unity, elements in finite fields GF(2^m) can be represented in the form:

$$A = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \quad (1)$$

Where $a_i \in$ GF (2), for $0 \le i \le n-1$, such that the set $\{1, x^2, \ldots, x^{n-1}\}$ is defined as the RB for finite field elements, where n is a positive integer not less than m.

Where $a_i \in$ GF (2), for $0 \le i \le n-1$, such that the set $\{1, x^2, \ldots, x^{n-1}\}$ is defined as the RB for finite field elements, where n is a positive integer not less than m.

For a finite field GF(2^m), when (m+1) is prime and 2 is a primitive root modulo (m+1), there exists a type-I optimal normal basis (ONB), where is the element of GF (2^m), and n=m+1.

Let A, B $\in$ GF (2^m) be expressed in RB representation as

$$A = \sum_{i=0}^{n-1} a_i x^i \quad (2)$$

$$B = \sum_{i=0}^{n-1} b_i x^i \quad (3)$$

Where $a_i, b_i \in$ GF (2^m).

Let C be the product of A and B, which can be expressed as

$$c_i = \sum_{i=0}^{n-1} b_{(i-j)n}\ a_j \quad (4)$$

Where $(i-j)n$ denotes modulo n reduction. In the newly proposed RB multiplier both operands A and B are disintegrated into a number blocks to achieve digit-serial multiplication and after that partial product corresponding to these blocks are added together to obtain the desired product word.

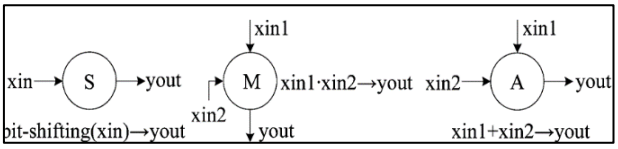

Fig. 2. Functional description of S- node, M–node, and A-node

The functional description of S-node performs bit shifting operation and M-node performs multiplication operation and A- node performs addition operation in a parallel realization of RB multiplication. All operations are performed in the signal flow graph.

B. Processor Space Flow Graph (PSFG)

The functions of nodes of PSFG are the same as those of corresponding nodes in the SFG and except an extra add-accumulation (AA) node to execute the accumulation operation for Q-cycles to yield the desired result.

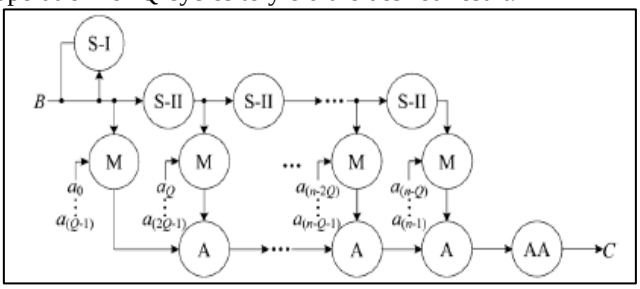

Fig. 3: Processor Space Flow Graph (PSFG)

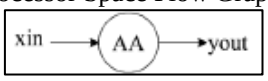

Fig. 4: Add-Accumulation Node

C. Cut-Set Retiming Of Processor Space Flow Graph

A retiming is a conversion technique used to change the locations of delay elements in a circuit without affecting the input/output characteristics of the circuit. When using the retiming the clock period of the circuit should be reduced and delay also reduced.

1) Types of Finite Fields

a) Polynomial Basis

If R is a commutative ring, then a polynomial in the indeterminate x over R is an expression of form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

Where $a_i \in R, \forall i \in \{0, 1 \ldots n\}$. The element a_i is called the coefficient of xi in f(x). The largest integer m (if any) such that $a_m \neq 0$ is the degree of f (x). It is denoted degree (f) and is called the leading coefficient.

If all the coefficients of f(x) are equal to 0 then f (x) is called the zero polynomial and its degree defined to be equal to 0 to $-\infty$. The zero-degree polynomials are also called constant polynomials.

A monic polynomial is a polynomial whose leading coefficient is equal to 1. The polynomial ring R[x] is the ring formed by the set of all polynomials in the indeterminate with coefficients in R.

The two operations are the standard polynomial addition and multiplication, with coefficient arithmetic performed in R. The additive identity element 0 is the zero polynomial. The multiplicative identity element 1 is the monic constant polynomial.

b) Normal Basis

The characteristic two is that the squaring operation in NB simply a cyclic shift of the coordinates of elements. It is useful for computing large exponentiations and multiplication inverses.

Multiplication table of normal basis is symmetric so suitable for hardware implementation. For many values of m, the finite field GF ((2^m)) has an optimal basis illustration as well as the polynomial illustration described.

An optimal basis is given an alternative way of defining multiplication on the elements of a field. It is less insightful than polynomial multiplication. It is in practice much more efficient. The value of m-determines which type shall be used.

c) Dual Basis

In linear algebra, given a vector V with a basis B of vectors indexed by an index set its dual set is a set B* of vectors in the dual space V* with the same index set 1such that B and form a biorthogonal system.

The dual set is always linearly independent but does not necessarily span V*. If it does span V*, then B* is known as the dual basis for the basis B.

Representing the indexed vector sets as and being biorthogonal means that the elements pair to 1 if the indexes are equal and to zero otherwise symbolically calculating a dual vector in V* on a vector in the original space V.

2) Binary Field

A field that contains binary numbers. It may refer to the storage of binary numbers for designing purposes, or to a field that is capable of holding any pieces of information including data, text, graphics images, voice, and video.

3) Prime Fields

A finite field or Galois field (so-named in the integrity of Évariste Galois) is a field that covers a finite number of elements. As with any field, a finite field is a set on which the operations of multiplication, addition, subtraction and division are defined and satisfy certain basic rules.

4) Extension Field

A field (F, +, *) consists of a set F with two binary operations + and *, with an additive identity element 0 and a multiplicative identity element 1 and the field extension satisfy the finite field elements such as

– (F, +, *) is a commutative ring.
– All non-zero elements of F have a multiplicative converse.

III. Results and Discussion

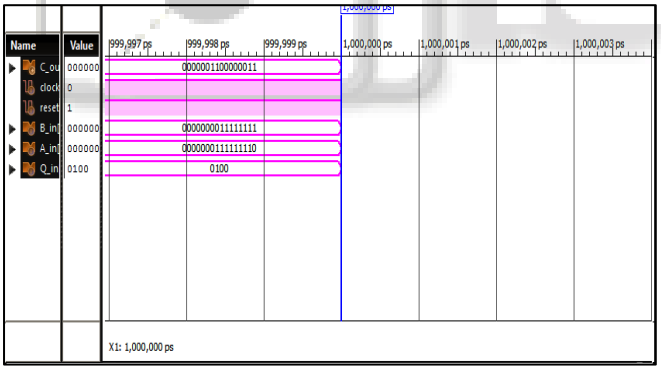

Fig. 5: Simulation Results for Signal Flow graph

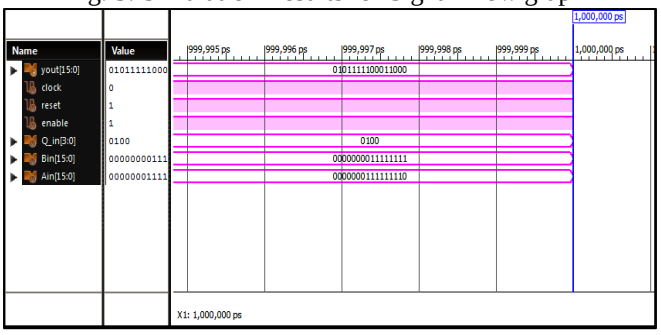

Fig. 6: Simulation Results for Processor Signal Flow graph

The figure 4 and 5 shows the output of RB Redundant basis multiplier which is implemented using two methods namely SFG and PSFG respectively. Simulation results show the correct operation of the multiplier operation. The comparative analysis of the existing system (SFG) and proposed system (PSFG) are shown in Table 1. It shows that the multiplier implemented using PSFG gives better performance in terms of reduced power and delay than that of multiplier designed using SFG.

Parameter	SFG	PSFG
Total Delay	7.2ns	7.0ns
Power	0.052w	0.052w
Power Delay Product	0.374w	0.364w
Area	48353µm^2	39782µm^2

Table 1: Comparison of SFG and PSFG

IV. Conclusion

The modified recursive decomposition for RB multiplication implemented using PSFG is written using Verilog and simulated using Xilinx ISE. Based on the comparative analysis, it is inferred that the multiplier implemented using PSFG consumes lower power and has less delay than that of the multiplier designed using SFG. Further, by the suitable projection of SFG and identifying suitable cut-sets for feed-forward cut-set retiming, high speed and throughput could be achieved. Redundant Basis (RB) multipliers output by using Process Space Flow Graph (PSFG) describes the reduce delay and increase the speed compared to the Signal Flow Graph (SFG). Moreover, efficient structures with low register count have been resulting from area-constrained implementation and particularly for implementation in FPGA platform where registers are not abundant and using various real-time applications environment.

References

[1] K.Aerts, B.Gövem, K.Järvinen, I.Verbauwhede, N.Mentens, "A Fast and Compact FPGA Implementation of Elliptic Curve Cryptography Using Lambda Coordinates," Springer, Cham, vol. 9646, pp. 63-83, April 2016.

[2] M.F Albrawy., A.E Taki El-Deen, M.E. Abo-Elsoaud, "Implementation of Elliptic Curve Crypto-System to Secure Digital Images over Ultra-Wideband Systems Using FPGA," Springer, Cham, vol. 533, October 2016.

[3] Jiafeng Xie, Pramod Kumar Meher and Zhi-Hong Mao, "High- Throughput Finite Field Multipliers Using Redundant Basis for FPGA," Steady Papers, IEEE Transactions on Circuits and Systems—I, vol.62, no.1, January 2015.

[4] Xiaoping Cui, Weiqiang Liu,Xin Chen, Earl E. Swartzlander, Jr., Life Fellow and Fabrizio Lombardi, Fellow, " A Modified Partial Product Generator for Redundant Binary multipliers", IEEE Transactions on Computer, 2015.

[5] S.Balamurugan and P.S Mallick., "Fixed-Width Multiplier Circuits Using Column Bypassing and Decomposition Logic Techniques," International Journal of Electrical Engineering and Informatics – vol. 7, no. 4, December 2015

[6] Xie, J.He and P.K.Meher, "Low latency systolic montgeomery multiplier for finite field GF(2m) based on penta nomials", "IEEE.Trans.Large scale integer.(VLSI) syst. , vol. 21, no. 2, pp. 385-389, February 2013.

[7] G.Drolet P.K..Meher, "On efficient implementation of accumulation the finite field over GF(2m) and its

applications," IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 17, no. 4, pp. 541–550, 2009.
[8] P. K. Meher, "Systolic and super-systolic multipliers for finite field GF(2m) based on irreducible trinomials," IEEE Trans. Circuits Syst.I, Reg. Papers, vol. 55, no. 4, pp. 1031–1040, May 2008.
[9] N. R. Murthy and M. N. S. Swamy, "Cryptographic applications brahma qupta-bha skara equation," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 7, pp. 1565–1571, October 2006.
[10] C.-Y. Lee, J.-S. Horng, I.-C. Jou, and E.-H. Lu, "Low-complexity bit-parallel systolic montgomery multipliers for special classes of GF(2m)," IEEE Trans.Comput., vol. 54, no. 9, pp. 1061–1070, September 2005.
[11] H.Wu, M.A.Hasan, I.F.Blake, and S.Gao, "Finite field multiplier using redundant representation," IEEE Trans.comput., vol. 51, no. 11, pp. 1306-1316, November 2002.
[12] L. Song, K. K. Parhi, I. Kuroda, and T.Nishitani, "Hardware/software code sign of finite field data path for low-energy Reed-Solomn codecs," IEEE Trans. Very Large Scale Integer.(VLSI) Syst., vol. 8, no. 2, pp.160–172, April 2000.
[13] I. Blake, G. Seroussi, and N.P.Smart, "Elliptic Curves in Cryptography," ser. London Mathematical Society Lecture Note Series. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[14] L. Song and K.K. Parhi, "Low-energy digit-serial/ parallel finite field multipliers," J. VLSI Digit. Process., vol. 19, pp. 149–C166, 1998.
[15] G. Drolet, "A new representation of elements of finite fields GF(2m) yielding small complexity arithmetic circuits," IEEE Trans.comput, vol. 47, no. 9, pp. 938–946, 1998.

