AREA EFFICIENT SYSTOLIC ARCHITECTURE FOR ALL-ONE POLYNOMIAL MULTIPLIER

Nhivashini.G, Sheebha.B

Abstract—**Irreducible polynomial algorithm for modular multiplication with a large modulus has been widely used for error control coding in secured data communication. This paper presents an area-time efficient bit-parallel systolic multiplication architecture over GF (2^m) based on irreducible all-one polynomial. It is further shown that the systolic structure can be decomposed into two or more parallel systolic branches where the pair of systolic branches has the same input operand, and they can share the same input operand registers This circuit is constructed by identical cells, each of which consists of one two-input AND gate, one two-input XOR gate and Bit shift cell. The proposed architecture is well suited to VLSI systems due to their regular interconnection pattern and modular structure. From the field programmable gate array synthesis results provides less area-delay and power-delay complexities over the best of the existing designs.**

Index terms— **Irreducible Polynomial, systolic array, finitefield**

I.INTRODUCTION

Finite field multipliers over GF (2^m) have widely as error-control coding in digital communication and storage system. [1],[2]. Multiplication in polynomial basis offer higher scalability and does not require a basis conversion. Efficient hardware design for polynomial-based multiplication is therefore important for real-time applications[3]-[5]. All-one polynomial (AOP) is one of the classes of polynomials considered suitable to be used as irreducible polynomial for efficient implementation of finite field multiplication. For the bit parallel systolic architecture, the number of gates can drastically reduced by selecting special irreducible polynomial[6]-[7].Using appropriate retiming and optimization of implementation of the logic functions in the PEs, therefore derive an efficient bit-parallel systolic design of finite field multiplier to be used for RS codec[8]-[12].

In this paper, register sharing technique to reduce the circuit complexity in the systolic design .

The proposed algorithm helps by reducing the latency and register complexity. Cut-set retiming allows introducing number of delays on all the edges in one direction of any cut-set of signal flow-graph(SGF). This technique is used for the digital circuits to minimize the critical path.

The rest of the paper is organized as follows. In Section II, summarize the previous work on systolic bit-parallel structure algorithm. Results and discussion of Low Latency systolic structure in Section III. Hardware complexities are compared in the Section IV. Details of Reed-Solomon Encoder based on finite field multiplier using irreducible All-One Polynomial(AOP) in Section V. Conclusions are presented in Section VI.

II.ALGORIHTM

Let $f(x) = x^m+x^{m-1}+\ldots+x+1$ is an irreducible AOP of degree m over GF (2). As a requirement of irreducible AOP for GF (2^m), (m+1) is prime and 2 is the primitive modulo (m+1).

The set $\{\alpha^{m+1},\alpha^{m-2},\ldots.\alpha,1\}$ forms the polynomial basis (where α is a root of f(x)), such that an element X of the binary field can be given by

$$X=X_{m-1}\alpha^{m-1}+X_{m-2}\alpha^{m-2}+\ldots+X_1\alpha+X_0$$

Where $X_i \in GF(2)$ for i=m-1… 2, 1, 0. since α is a root of f(x), we can have $f(\alpha)=0$,and

$$f(\alpha)+\alpha f(\alpha)=(\alpha^m+\alpha^{m-1}+\ldots.+\alpha+1)+\alpha(\alpha^m+\alpha^{m-1}+\ldots.+\alpha+1)$$
$$=\alpha^{m+1}+1=0$$

Therefore, we have

$$\alpha^{m+1}=1$$

This property of AOP is used to reduce the complexity of field multiplications as discussed in the following.

$$A=\sum_{j=0}^{m} a_j\alpha^j \quad B=\sum_{j=0}^{m} b_j\alpha^j \quad A=\sum_{j=0}^{m} c_j\alpha^j$$

Where a_j, bj, cj $\in GF(2)$, for $0<= j <= m-1$, and $a_m=0$, $b_m=0$, $c_m=0$ If C is the product of elements

A. Basic systolic design

ISSN: 2278 – 1323

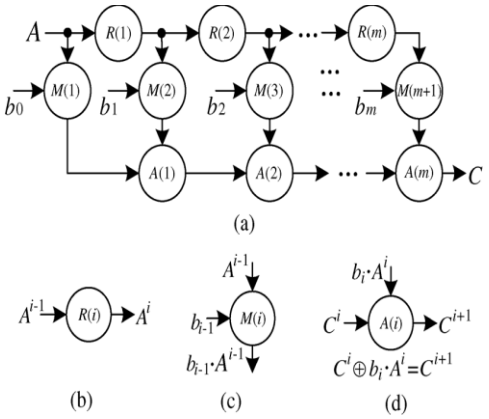

Fig. 1.SFG of the algorithm(a)The SFG.(b)Function of node R(i)(c) Function of node M(i)(d) Function of node A(i)

Generally, we can introduce a delay between the reduction node and its corresponding bit-multiplication and bit-addition nodes, as shown in fig 1 such that the critical-path is not larger than, where the propagation delay of AND gate and XOR gate, respectively. In this section, however, we introduce a novel cut-set retiming to reduce the critical-path of a PE. It is observed that the node performs only the bit-shift operation according to and therefore it does not involve any time consumption. Therefore, we introduce a critical-path which is not larger than to derive the basic design of a systolic multiplier, we have shown the formation of PE of the retimed SFG. It can be observed that the cut-set retiming allows to perform a reduction operations, bit-addition, and bit-multiplication concurrently, so that the critical path is reduced.

The systolic implementation of multiplication over GF (2^m), the operations of (1), (2) and (3) can be performed recursively. Each recursion is composed of three steps, i.e., modular reduction of (3), bit-multiplication of (2), and bit-addition of (1). Equations of (1), (2) and (3) can be shown below

$$C = \sum_{i=0}^{m} Xi \quad \ldots\ldots\ldots\ldots \quad (1)$$

$$X_i = b_i * A^i \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$A^{i+1} = a_0^{i+1} + a_0^{i+1} + a_1^{i+1}.\alpha + \ldots\ldots\ldots + a_m^{i+1}.\alpha^m \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

The systolic implementation of multiplication over GF (2^m), shown below in fig 2.

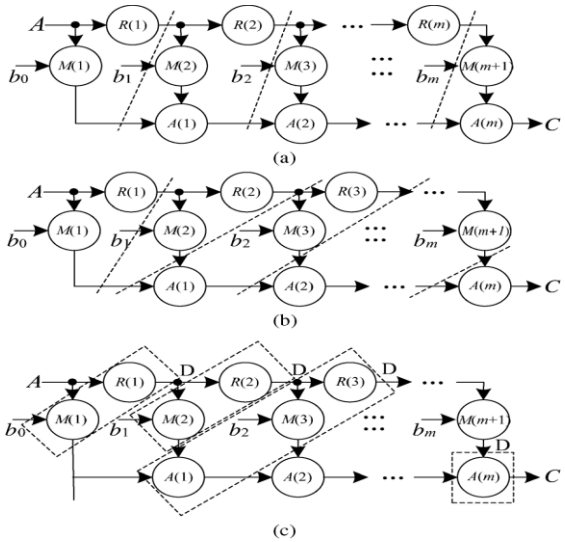

Fig.2.Cut-set retiming of the SFG.(a)Cut-set retiming in a general way.(b)Cut-set retiming.(c)Formation of PE."D" denotes unit delay.

B. Low-Latency Register Sharing Systolic Structure

For irreducible AOP, m is an even number. Therefore , let l and P be two integers such that(m+1)=lP+r, where r is an integer in the range $0 \leq r \leq l$.For example, if we choose P=m/2, then l=2, r=1 is

$$C = \sum_{i=0}^{m/2} Xi + \sum_{i=\frac{m}{2}+1}^{m} Xi$$

The systolic structure is decomposed into two parallel systolic branches. The upper branch consists of [(m/2)+2]PEs and the lower branch consists of[(m/2)+1]Pes and a delay cell. Besides, an addition-cell(AC) is required to perform the final addition of the outputs of the two systolic arrays. The latency of the structure is only[(m/2)+3]cycles.

It is observed that the two systolic branches in fig 3 share the same input operand A, and the PE(from PE[2] to PE[m/2-1] is shown in fig 4 where the structure consists of [(m/2)+2]Pes and an AC. The circuit of its regular PE(from PE[2] to PE[m/2-1]) is shown is 4©. It combines two regular Pes of fig 3 together by sharing one input-operand-transfer. The other Pes need some minor modifications, as shown in fig.4(b),(d) and (e), respectively

ISSN: 2278 – 1323

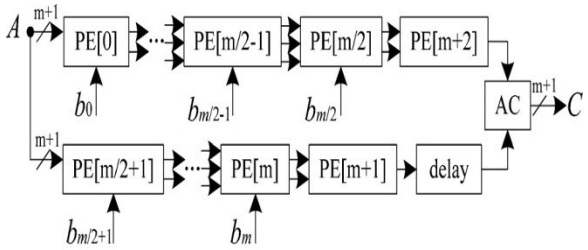

Fig. 3. Low Latency Systolic Structure

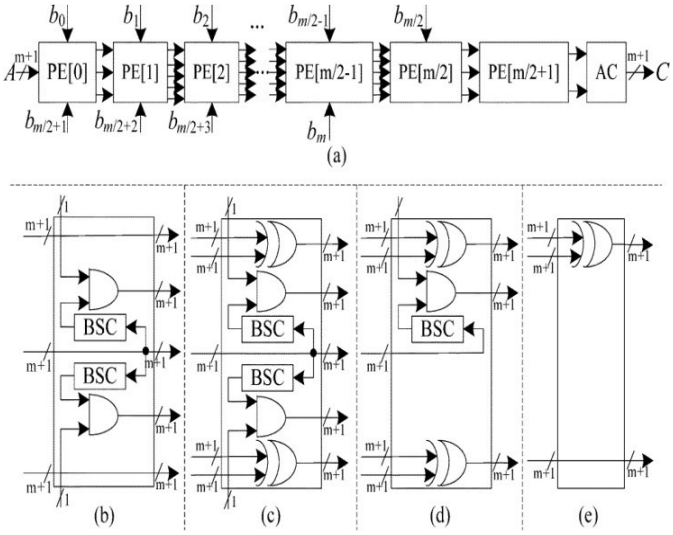

Fig. 4.Low Latency register-sharing systolic structure. (a)Structure of PE[1].(C)Structure of a regular PE(from PE[2] to PE[m/2-1].(d)Structure of PE[m/2+1].

Besides, the latency of structure is [(m/2+3)] cycles, while the duration of the cycle period of a regular is Tx. The systolic structure is further is decomposed then,

$$C = \sum_{i=0}^{\frac{m}{4}-1} Xi + \sum_{i=m/4}^{\frac{m}{2}-1} Xi + \sum_{i=m/2}^{\frac{3m}{4}-1} Xi + \sum_{i=3m/4}^{m} Xi$$

Following the same approach as the one used to derive the structure of Fig. 3, it consists of four systolic branches shown in Fig. 5(a). The design of Fig. 5(b) requires only [(m/4)+4]cycles of latency.

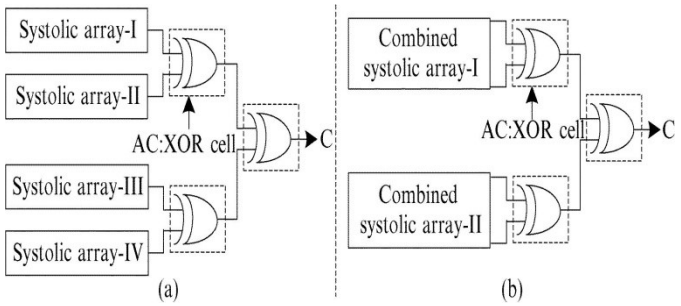

Fig. 5.Improved low-latency systolic structure.(a)The proposed systolic array merging(b)Improved systolic Structure

The systolic structure in Fig.3 requires [(m/2)+2]Pes and one AC. Each of the regular PE consists of 2(m+1) XOR gates and 2(m+1)AND gates. Besides AC requires (m+1) XOR gates. The latency of the design is [(m/2)+3] cycles, where the duration of the clock-period is Tx. The structure of Fig. 5 requires nearly the same logic-elements as that of Fig. 4. But its latency is [(m/4)+4]cycles.

III RESULTS AND DISCUSSION FOR m=6

The Low Latency Systolic structure has been designed has been coded in verilog and simulated in Modelsim6.4c for m=7.The simulation result for the low latency systolic structure shows in Fig, 6

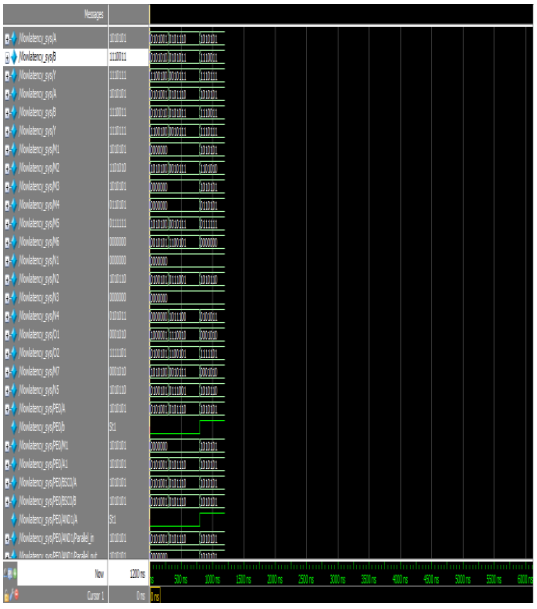

Fig. 6 Low latency Systolic structure simulation result.

ISSN: 2278 – 1323

V.HARDWARE AND TIME COMPLEXITY

The proposed structure (see Fig. 4) requires [(m=2)+2] PEs and one AC. Each of the regular PEs consists of 2(m + 1) XOR gates in a pair of XOR cells and 2(m + 1) AND gates in a pair of AND cells. Besides, the AC requires (m + 1) XOR gates. Moreover, (2:5m^2+ 6:5m + 4) bit-registers are required for transferring data to the nearby PE. The latency of the design is [(m=2) + 3] cycles, where the duration of the clock-period is T_X. The structure of Fig. 5 requires nearly the same gate-counts as that of Fig. 4. But its latency is [(m=4) + 4] cycles. The number of Logic Utilization, Logic Distribution, LUT and gate count of Fig.3 and Fig.4 are listed in Table I and Table II.

TABLE I
LOW LATENCY SYSTOLIC ARCHITECTURE

Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	42	7,168	1%
Logic Distribution			
Number of occupied Slices	21	3,584	1%
Number of slices containing only related logic	21	21	100%
Number of slices containing unrelated logic	0	21	0%
Total number of 4 input LUTs	42	7,168	1%
Number of bonded IOBs	21	141	14%
Total equivalent gate count for design	294		
Additional JTAG gate count for IOBs	1,008		

TABLE II
REGISSTER SHARING SYSTOLIC ARCHITECTURE

Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	35	7,168	1%
Logic Distribution			
Number of occupied Slices	18	3,584	1%
Number of slices containing only related logic	18	18	100%
Number of slices containing unrelated logic	0	18	0%
Total number of 4 input LUTs	35	7,168	1%
Number of bonded IOBs	21	141	14%
Total equivalent gate count for design	210		
Additional JTAG gate count for IOBs	1,008		

The proposed design (see Fig. 4) has been coded in Verilog and syn- thesized by Xilinx 13.2 Spartan 3 for m=6. has lower ADP and less PDP than the existing ones.

V.REED SOLOMON ENCODER

A systolic multiplier based on Reed Solomon Encoder Application will be designed is shown in Fig. 7. The increasing application of cryptographic algorithms to ensure secure communications across virtual networks has led to an ever-growing demand for high performance hardware implementations of the encryption methods. In our paper, the proposed design has Finite Field Galois Field multipliers. Using this Multiplier we will do the RS Encoder Design.

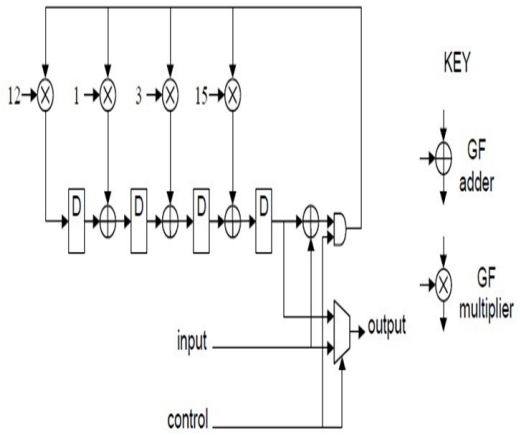

Fig. 7 Future Enhancement Module Diagram

VI.CONCLUSION

Efficient systolic design for the multiplication over GF (2^m) based on irreducible AOP is discussed. We have derived a low-latency bit-parallel systolic multiplier. Compared with the existing systolic structures for bit-parallel realization of multiplication over GF (2^m), the discussed one is found to involve less area and Logic Utilization. From ASIC and FPGA synthesis results we find that the design involves significantly less gate count than the existing designs. Moreover proposed design can be extended to further to reduce the latency and also

REFERENCE

[1]. S. B. Wicker and V. K. Bhargava, Eds., Reed-Solomon codes and their applications. Piscataway, NJ: IEEE Press, 1994.

[2]. I. S. Hsu, T. K. Truong, L. J. Deutsch, and I. S. Reed, "A comparison of VLSI architecture of finite field multipliers using dual, normal, or standard

ISSN: 2278 – 1323

bases," IEEE Trans Computers, vol. 37, no. 6, pp. 735–739, June 1988.
[3]. S. Fenn,M.G. Parker,M. Benaissa, and D. Taylor, "Bit-serial multiplication in GF (2^m) using all-one polynomials," IEE Proc. Com. Digit. Tech., vol. 144, no. 6, pp. 391–393, 1997.
[4].C.-Y. Lee, E.-H. Lu, and J.-Y. Lee, "Bit-parallel systolic multipliers for GF (2^m) fields defined by all-one and equally spaced polynomials," IEEE Trans. Computers, vol. 50, no. 6, pp. 385–393, May 2001.
[5]. Y.-R. Ting, E.-H. Lu, and Y.-C. Lu, "Ringed bit-parallel systolic multipliers over a class of fields GF (2^m)," Integr., VLSI J., vol. 38, no. 4, pp. 571–578, 2005.
[6]. C. H. Kim, C.-P. Hong, and S. Kwon, "A digit-serial multiplier for finite field GF (2^m) ," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 4, pp. 476–483, 2005.
[7]. K.-Y. Chang, D. Hong, and H.-S. Cho, "Low complexity bit-parallel multiplier for GF (2^m) defined by all-one polynomials using redundant representation," IEEE Trans. Computers, vol. 54, no. 12, pp. 1628–1629, Dec. 2005.
[8]. 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 (2^m) ," IEEE Trans. Computers, vol. 54, no. 9, pp. 1061–1070, Sep. 2005.
[9]. H. Fan and M. A. Hasan, "Relationship between GF (2^m) Montgomery and shifted polynomial basis multiplication algorithms," IEEE Trans. Computers, vol. 55, no. 9, pp. 1202–1206, Sep. 2006.
[10]. H.-S. Kim and S.-W. Lee, "LFSR multipliers over GF (2^m) defined by all-one polynomial," Integr., VLSI J., vol. 40, no. 4, pp. 571–578, 2007.
[11]. H. Wu, "Bit-parallel polynomial basis multiplier for new classes of finite fields," IEEE Trans. Computers, vol. 57, no. 8, pp. 1023–1031, Aug. 2008.
[12]. M. Sandoval, M. F. Uribe, and C. Kitsos, "Bit-serial and digit-serial GF (2^m) montgomery multipliers using linear feedback shift registers," IET Comput. Digit. Tech., vol. 5, no. 2, pp. 86–94, 2011.

ISSN: 2278 – 1323