Indian Journal of Electronics and Electrical Engineering (IJEEE)
Vol.2.No.1 2014pp1-6
available at: www.goniv.com
Paper Received :05-03-2014
Paper Published:28-03-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

AN IMPROVED LOW LATENCY SYSTOLIC STRUCTURED GALOIS FIELD MULTIPLIER

M.Rajendran, M.Rajendran,
Applied Electronics
Pallavan College Of Engineering
Kanchipuram, India
05rajendran@gmail.com, rajendransusan@gmail.com

ABSTRACT

This paper presents an improved low latency systolic structure for binary multiplication over Galois Field based on irreducible all-one polynomial. The systolic design is a special type of hardware solution because of its ability of pipelining and local connectivity. A cut-set retiming technique is proposed to reduce the duration of the critical-path, to one XOR gate delay in this design. Further the systolic structure can be decomposed into two or more parallel systolic branches, which have the same input operand and share the same input operand registers. Using the improved finite field multipliers, Reed Solomon encoder which uses secure authentication in cryptography applications, is designed. From the implemented hardware synthesis results, the proposed design provides significantly less area and power-delay complexities over the existing designs.

***Index Terms*—**Finite field, all–one polynomial, retiming, memory sharing technique, reed solomon encoder.

I. INTRODUCTION

Finite field multipliers over *GF(2^m)* have wide applications in elliptic curve cryptography (ECC) and error control coding systems [1], [2]. 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. Multipliers for the AOP-based binary fields are simple and regular, and therefore, a number of works have been explored on its efficient realization [6]–[17]. Irreducible AOPs are very often not preferred in cryptosystems for security reasons, and one has to make careful choice of the field order to use irreducible AOPs for cryptographic applications [1], [9]. The AOP-based multipliers can be used for the nearly AOP (NAOP) which could be used for efficient realization of ECC systems [18]. In [13], a bit-parallel AOP-based systolic multiplier has been suggested by Lee *et al*. In a recent paper [15], a low-complexity bit-parallel systolic Montgomery multiplier has been suggested. Very recently [16], an efficient digit-serial systolic Montgomery multiplier for AOP-based binary extension field is presented. The systolic structures for field multiplication have two major issues. First, the registers in the systolic structures usually consume large area and power. Second, the systolic structures usually have a latency of nearly *m* cycles, which is very often undesired for real-time applications. Therefore, in this paper, we have presented a novel register- sharing technique to reduce the register requirement in the systolic structure. Besides, we have proposed a novel cut-set retiming approach to reduce the clock-period.

AOP-based fields could also be used for efficient implementation of Reed-Solomon encoders [19]. Reed-Solomon coding is one of the most important schemes for error detection and correction. The Reed-Solomon codes are called after their discoverers and widely used in digital communication systems. They are constructed and decoded using finite field arithmetic referred as Galois Fields (GF). Thus a real time programmable Reed Solomon coding processor is implemented. The proposed structure is found to involve significantly less area-time-power complexity.

II. ALGORITHM

Let $f(x) = x^{m} + x^{m-1} + \dots + x + 1$ be 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 $\{\alpha^{m-1}, \alpha^{m-2}, .., \alpha, 1\}$ forms the canonical basis, such that an element X in the binary field can be given by

$$X = X_{m-1}\,\alpha^{m-1} + X_{m-2}\,\alpha^{m-2} + \dots + X_{1}\,\alpha + X_{0} \qquad (1)$$

where Xi ε GF(2) for i = m - 1,…., 2, 1, 0

Since α is a root of f(x), we can have f (α) = 0, and

$$\begin{aligned} f(\alpha) + \alpha f(\alpha) &= (\alpha^{m} + \alpha^{m-1} + \dots + \alpha + 1) \\ &\quad + \alpha(\alpha^{m} + \alpha^{m-1} + \dots + \alpha + 1) \\ &= \alpha^{m-1} + 1 = 0 \end{aligned} \qquad (2)$$

Therefore, we have

$$\alpha^{m-1} = 1 \qquad (3)$$

This property of AOP [17] is used to reduce the complexity of field multiplications as discussed in the following.

Any element X in GF(2^{m}) given by (1) in polynomial basis representation can be represented as, $X = x_{0} + x_{1}\alpha + \dots + x_{m}\alpha^{m}$, where xi ϵ GF(2), and $\{\alpha^{m}, \alpha^{m-1}, \dots, \alpha, 1\}$ is the extended polynomial basis[17]. Similarly, if A, B, C ϵ GF(2^{m}), they can be represented by the extended polynomial basis as

$$A = \sum_{j=0}^{m} a_{j}\,\alpha^{j}, \quad B = \sum_{j=0}^{m} b_{j}\,\alpha^{j}, \quad C = \sum_{j=0}^{m} c_{j}\,\alpha^{j} \qquad (4)$$

where a_{j}, b_{j}, and c_{j} ϵ GF(2), for $0 \le j \le m - 1$, and $a_{m} = 0$, $b_{m} = 0$

and $c_{m} = 0$.

If C is the product of elements A and B, then we have

$$C = A.\,B \bmod f(\alpha) \qquad (5)$$

This can be decomposed to a form

$$C = \sum_{i=0}^{M} b_{i}\,(\alpha^{i}\,.\,A \bmod f(\alpha)\,) \qquad (6)$$

Equation (6) can be expressed as a finite field accumulation

$$C = \sum_{i=0}^{M} X_{i} \qquad (7)$$

where Xi is given by

$$Xi = bi.\,A^{i} \qquad (8a)$$

for $A^{0} = A$, and $A^{i} = [\,\alpha^{i}\,.\,A \bmod f(\alpha)\,]$ and using (3) Ai can be obtained from A as

$$A^{i} = a_{m-1}\,\alpha^{m} + a_{m-i-1}\,\alpha^{m-1} + \dots + a_{m-i+2}\,\alpha + a^{m-i+1} \qquad (8b)$$

Such that Ai + 1 can be obtained from A^{i} recursively as

$$A^{i+1} = \alpha.\,A^{i} \bmod f(\alpha) \qquad (9)$$

The partial product generation and modular reduction are performed according to (8) and (9) respectively. The additions of the reduced polynomials are performed according to (7).

Equation (9) can be expressed as

$$A^{i+1} = [\,a_{0}{}^{i}.\,\alpha + a1^{i}.\,\alpha^{2} + \dots + am^{i}.\,\alpha^{m+1}] \bmod f(\alpha) \qquad (10a)$$

where

$$A^{i} = \sum_{j=0}^{M} a_{j}{}^{i}\,\alpha^{j} \qquad (10b)$$

Substituting (3) into (10a), A^{i+1} can be obtained as

$$A^{i+1} = a0^{i+1} + a1^{i+1}.\,\alpha + \dots + am^{i+1}.\,\alpha^{m} \qquad (11a)$$

where

$$a_{0}{}^{i+1} = a^{i}{}_{m} \qquad (11b)$$

$$a_{j}{}^{i+1} = a_{ij} - 1, \quad \text{for} \quad 1 \le j \ge m - 1 \qquad (11c)$$

It is also possible to extend (11) further to obtain A^{i+l} directly from A^{i} for $1 \le l \ge m$, such that

$$a_j^{i+1} = \begin{cases} a^i_{m-j+j+1}, & \text{for } 0 \le j \ge l-1 \\ a^i_{j-i,} & \text{otherwise} \end{cases} \qquad (12)$$

We have used the above equations to derive the proposed linear systolic structure based on a novel cut-set retiming strategy and register-sharing technique.

III. BASIC SYSTOLIC STRUCTURE

For systolic implementation of multiplication over GF (2^m), the operations of (7), (8) and (11) can be performed recursively. Each recursion is composed of three steps, i.e., modular reduction of (11), bit-multiplication of (8), and bit-addition of (7). Equations of (7), (8) and (11) can be represented by the SFG (shown in Fig. 1) consisting of m modular reduction nodes R(i) and m addition nodes A(i) for $1 \le i \ge m$, and (m + 1) multiplication nodes M(i) for $1 \le i \ge m+1$.

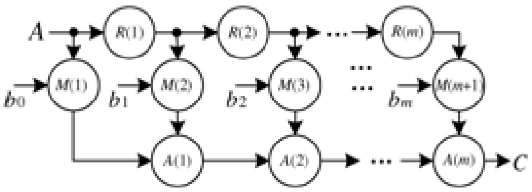

Fig. 1. Signal flow graph

Node R(i) perform the modular reduction of degree by one according to (11). Node M(i) performs an AND operation of a bit of operand B with a reduced form of operand A , according to (8). Node A(i) performs the bit-addition operation according to (7).

IV. RETIMING TECHNIQUE

Generally, we can introduce a delay between the reduction node and its corresponding bit-multiplication and bit-addition nodes, such that the critical-path is not larger than (T_A + T_X),where the T_A and T_X refer 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 to T_X. It is observed that the node R(i) performs only the bit-shift operation according to (11), and therefore it does not involve any time consumption.

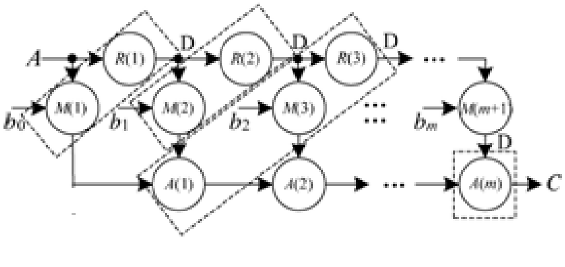

Fig. 2. The formation of PE of the retimed SFG

Therefore, we introduce a critical-path which is not larger than T_X. The basic design of a systolic multiplier, 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 to max{ $T_{A,}$ T_M, T_R }, where T_A, T_M and T_R are, respectively, the computation times of the bit-addition nodes, bit-multiplication nodes, and reduction nodes.

The basic design of systolic multiplier thus derived is shown in Fig. 3. It consists of (m+2) PEs, and the functions of the PEs are shown in Fig. 3. During each cycle period, the regular PE not only performs the modular reduction operation according to (11), but also performs the bit-multiplication and bit-addition operations concurrently.

The regular PE consists of three basic cells, e.g., the bit-shift cell (BSC), the AND cell, and the XOR cell. The AND cell, and the XOR cell correspond to the node M(i), and node A(i) of the SFG of Fig. 1, respectively.

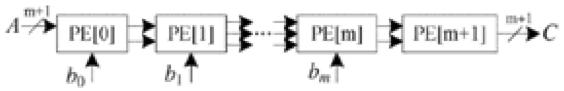

Fig. 3. Basic systolic design

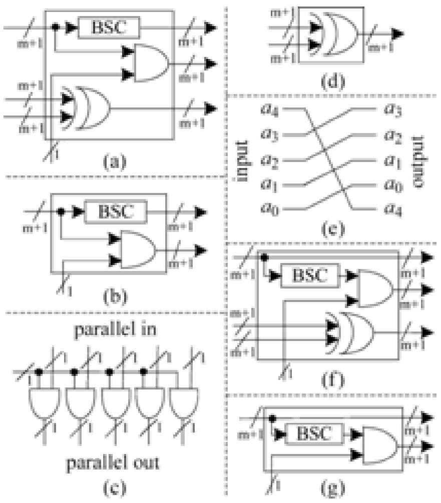

Fig. 4. Structure of PEs. (a) Internal structure of a regular PE. (b) Internal structure of PE[0]. (c) An example of AND cell for m=4. (d) Structure of the AC. (e) Structure of BSC where m=4. (f) Alternate structure of a regular PE. (g) Alternate structure of PE[0].

The structure of PE[1] consists of an AND cell and a BSC. Each XOR cells and AND cells in the PE consists of (m+1) number of gates working in parallel. The PE[m+1] of the systolic structure consists of only an XOR cell, as shown in Fig. 4(d), which performs bit-by-bit XOR operations of its pair of m-bit inputs. The BSC in the PE performs the bit-shift operation according to (11). Therefore, we can change the circuit-designs of Fig. 4(a) and (b) into the form of Fig. 4(f) and (g), respectively. Besides, according to (11), the operation of node R(i) does not involve any area and time-consumption. Therefore, the minimum duration of clock-period of a regular PE amounts to max{ T_A, T_x }. The proposed systolic design yields the first output of desired product (m+2) cycles after the first input is fed to the structure, while the successive outputs are available in each cycle.

V. MEMORY SHARING TECHNIQUE

For irreducible AOP, m is an even number. Therefore, let *l* and *P* be two integers such that *(m + 1) = lP + r*, where is an integer in the range $0 \le r \ge l$. For example, if we choose P = m / 2, then *l = 2*, r = 1, (7) can be rewritten as

$$C = \sum_{i=0}^{m/2} X_i + \sum_{i=m/2+1}^{m/2} X_i \qquad (13)$$

As shown in (13), one of the sum contains [(m/2)+1] partial products while the other has m/2 partial products. Based on (13), the systolic structure of Fig. 4 could be modified to a form shown in Fig. 5, which consists of two 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 structure has the PEs of the same complexity as those in Fig. 3, but the latency of structure is only [(m/2)+3] cycles.

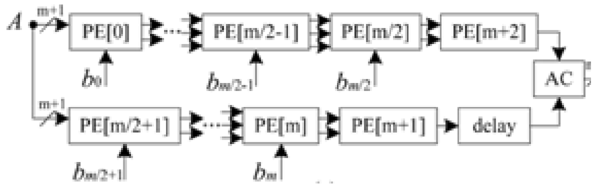

Fig. 4. Low latency systolic structure

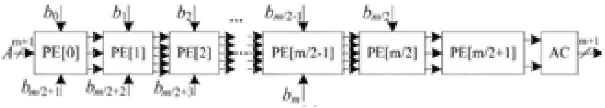

Fig. 5. Low latency register sharing systolic structure

It is observed that the two systolic branches in Fig. 5 share the same input operand A, and the PEs in both the branches perform the same operation except the last PE in each of the branches. Therefore, we present an efficient structure using the register-sharing technique as shown in Fig. 6, where the structure consists of [(m/2)+2] PEs and an AC. It combines two regular PEs of Fig.5(a) together by sharing one input-operand-transfer. Thus, the whole structure requires only [$2.5m^2 + 6.5m + 4$] bit-registers, while the structure of Fig. 4 requires [$3m^2 + 5m + 2$] bit-registers. Besides, the latency of structure is [(m/2) + 3] cycles, while the duration of cycle period of a regular PE is still T_X.

VI. IMPROVED LOW LATENCY SYSTOLIC STRUCTURE

We may further decompose the design in Fig. 6. For example, if we choose P = m/4, then l = 2, r = 1, (7) can be rewritten as

$$C = \sum_{i=0}^{m/4-1} X_i + \sum_{i=m/4}^{m/2-1} X_i + \sum_{i=m/2}^{3m/4-1} X_i + \sum_{i=3m/4}^{m} X_i \qquad (14)$$

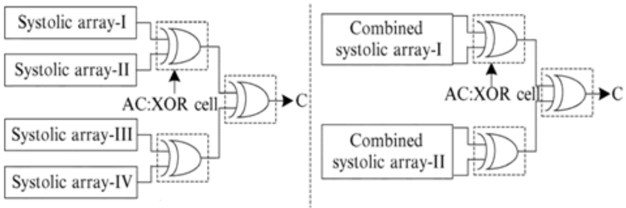

Fig. 7. Improved systolic structure

Following the same approach as the one used to derive the structure of Fig. 5, we can have the design in Fig. 7, where it consists of four systolic branches. Similarly, following the approach presented to derive the structure of Fig. 6 from Fig. 5, we may have the design shown in Fig. 7. The design of Fig. 7 requires only [(m/4) + 4] cycles of latency. When m is a large number, l and P can be chosen as to obtain optimal realization.

$$l = P = [m + 1] \qquad (15)$$

VII. REED SOLOMON ENCODER DESIGN

Reed-Solomon codes have a widespread use to provide error protection especially for burst errors. This feature has been an important factor in adopting RS codes in many practical applications such as wireless communication system, cable modem, computer memory.

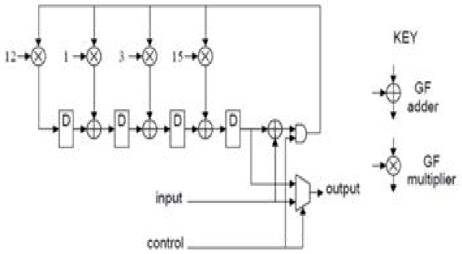

Fig. 8. Reed Solomon encoder

This thesis proposes an area efficient, low energy, high speed architecture for a Reed-Solomon RS(255,239) decoder based on Decomposed Inversionless Berlekamp-Massey Algorithm, where the error locator and evaluator polynomial can be computed serially. In the proposed architecture, a new scheduling of t finite field multipliers is used to calculate the error locator and evaluator polynomials to achieve a good balance between area, latency, and throughput. This architecture is tested in two different decoders. The first one is a two parallel decoder, as two parallel syndrome and two parallel Chien search are used. The second one is a serial decoder, as serial syndrome and Chien search are used. In our architectures we have investigated hardware area, throughput, and energy per symbol and we did a good optimization between the latency, throughput, and energy per symbol while maintaining a small area.

VIII. AREA AND TIME COMPLEXITY

The proposed structure (see Fig. 6) 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. The latency of the design is [(m/2)+ 3] cycles, where the duration of the clock-period is T_X. The structure of Fig. 7 requires nearly the same gate-counts as that of Fig. 6. But its latency is [(m/4)+ 4] cycles. The number of gates, latency and critical-path of the proposed designs are listed in Table I.

TABLE I
AREA AND TIME COMPLEXITIES

Design	Registers	Latency	Critical path
Basic systolic structure	$2(m + 1)^2$	m + 2	$T_A + T_F$
Low latency register sharing structure	$(5 / 2 \times m^2) + (13 / 2 \times m) + 4$	m / 2 + 3	T_X
Improved low latency systolic structure	$(5 / 2 \times m^2) + (1 / 2 \times m) + 7$	m / 4 + 4	T_X

It can be seen that the proposed design outperforms the existing designs. Although slightly more registers than that in [11] are used, proposed design requires shorter latency and lower critical-path than the other as well as the MUX gates. Besides, as shown in Fig. 7, the proposed design can be extended further to obtain a more efficient design for high-speed implementation, especially when m is a large number.

The proposed design has been coded in VHDL and synthesized by Synopsys Design Compiler using TSMC 90-nm library for m = 20 along with the bit-parallel systolic design of [15] and digit-serial systolic structure of [16]. The average computation time (ACT), area and power consumption (at 100 MHz frequency) thus obtained. The proposed design has at least 28.5% less area-delay product (ADP) and 28.2% lower power-delay product (PDP) compared to the existing ones.

IX. CONCLUSION

An improved efficient systolic design for the multiplication over GF(2^m) based on irreducible AOP and Reed Solomon application are proposed. By novel cut-set retiming we have been able to reduce the critical path to one XOR gate delay and by sharing of registers for the input-operands in the PEs, we have derived a low-latency bit-parallel systolic multiplier. Compared with the existing systolic structures for bit-parallel and bit-serial realization of multiplication over GF(2^m), the proposed one is found to involve less area, shorter critical-path and lower latency. From ASIC and FPGA synthesis results we find that the proposed design involves significantly less ADP and PDP than the existing designs. Besides, our proposed design can be extended to further reduce the latency.

REFERENCES

[1] M. Ciet, J. J. Quisquater, and F. Sica, “A secure family of composite finite fields suitable for fast implementation of elliptic curve cryptography,”in *Proc. Int. Conf. Cryptol. India*, 2001, pp. 108–116.

[2] 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.

[3] C.-L.Wang and J-L. Lin, “Systolic array implementation of multipliers for finite fields GF(2^m),” *IEEE Trans. Circuits Syst.*, vol. 38, no. 7, pp. 796–800, Jul. 1991.

[4] B. Sunar and C. K. Koc, “Mastrovito multiplier for all trinomials,”*IEEE Trans. Comput.*, vol. 48, no. 5, pp. 522–527, May 1999.

[5] 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.

[6] C. Paar, "Low complexity parallel multipliers for Galois fields GF(2^m) based on special types of primitive polynomials," in *Proc. IEEE Int. Symp. Inform. Theory*, 1994, p. 98.

[7] 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.

[8] 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.

[9] 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.

[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] P. K. Meher, Y. Ha, and C.-Y. Lee, "An optimized design of serial-parallel finite field multiplier for GF(2^m) based on all-one polynomials," in *Proc. ASP-DAC*, 2009, pp. 210–215.

[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.

[13] 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.

[14] 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.

[15] 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.

[16] S. Talapatra, H. Rahaman, and J. Mathew, "Low complexity digit serial systolic montgomery multipliers for special class of GF(2^m)," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 5, pp. 847–852, May 2010.

[17] T. Itoh and S. Tsujii, "Structure of parallel multipliers for a class of fields GF(2^m)," *Inform. Computation*, vol. 83, no. 1, pp. 21–40, 1989.

[18] C. Negre, "Quadrinomial modular arithmetic using modified polynomial basis," in *Proc. ITCC*, 2005, pp. 550–555.

[19] Z. Chen, M. Jing, J. Chen, and Y. Chang, "New viewpoint of bit-serial/ parallel normal basis multipliers using irreducible all-one polynomial," in *Proc. ISCAS*, 2006, pp. 1499–1502.