An Efficient Systolic Multiplier for GF (2^m) Based On All-One Polynomial

Neethu Johny[1], Mrs.Georgina Binoy Joseph[2]

[1]Electronics and Communication, KCG College of Technology
Chennai, Tamil Nadu - 600097 , India

[2]Electronics and Communication, KCG College of Technology
Chennai, Tamil Nadu - 600097 , India

Abstract

Finite field multipliers over GF(2^m) are widely used in applications like elliptic curve cryptography (ECC) and error control coding systems. These real time applications require an efficient hardware design for polynomial-based multiplication. In this paper, an efficient systolic structure for multiplication over GF(2^m) based on irreducible all-one polynomial (AOP) is designed. The paper presents a cut-set retiming technique to reduce the critical-path to one XOR gate delay. This technique is highly useful for pipelining digital circuits to reduce the critical path. This paper also presents a novel register- sharing technique and merging of systolic structures to reduce the register requirements and the latency in the systolic structure.

Keywords: *AOP, systolic, cut set retiming, register share technique*

1.Introduction

A low power, area efficient architecture is very important in high speed VLSI implementation. Finite field arithmetic is an area which involves operations like addition, subtraction, multiplication, division on finite fields. Thus the area efficient design of finite field multipliers is of very important in VLSI.

Many approaches and architectures have been proposed to perform multiplication in GF(2^m). In those implementations, much architecture applied the systolic array concept. In general, a non-systolic architecture has global signals. Therefore, if m becomes large, propagation delay also increases. A systolic architecture, however, does not suffer from the problem. This is because the systolic architecture consists of replicated basic cells and each basic cell is connected with its neighbouring cells through pipelining, i.e., there are no global signals. Consequently, the systolic architecture is a better choice than the non-systolic architecture for a high-speed VLSI implementation.

The most commonly used basis representations are dual, normal, and standard basis. Multipliers using the dual and normal basis representations require a basis conversion, in which complexity heavily depends on the irreducible polynomial. In contrast, multipliers that use the standard basis do not require a basis conversion; they are therefore more efficient from the point of view of irreducible polynomial selection and hardware optimization.

Though systolic structures are widely used for field multiplication, it has 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, this paper presents a novel register- sharing technique to reduce the register requirement in the systolic structure. The proposed algorithm not only facilitates sharing of registers by the neighbouring PEs to reduce the register complexity but also helps reducing the latency.

2. 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 $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$ 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 $X_i \varepsilon$ GF(2) for i= m-1,….,2,1,0.
Since α is a root of f(x), we can have f (α) = 0 and

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

Therefore, we have $\alpha^{m-1} = 1$ (3)

This property of AOP is used to reduce the complexity of field multiplications as discussed in the following. Any element in given by (1) in polynomial basis representation can be represented as, X= $x_0+x_1 \alpha +\ldots..+ x_m \alpha^m$; where xi ε GF(2) and $\{\alpha^m , \alpha^{m-1} ,\ldots.,\alpha ,1\}$ is the extended polynomial basis. Similarly, if, A, B, C ε GF(2^m), they can be represented by the extended polynomial basis as

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

$$C = \sum_{j=0}^{m} c_j \alpha^j \tag{4}$$

where a_j, b_j , and c_j ε GF(2), for $0\leq j \leq m-1$, and $a_m=0$, $b_m = 0$ and $c_m =0$.
If C is the product of elements A and B, then we have

$$C= A.B \bmod f(\alpha) \tag{5}$$

This can be decomposed to a form

$$C= \sum_{i=0}^{m} b_i (\alpha^i .A \bmod f(\alpha)) \tag{6}$$

The above equation can be expressed as a finite field accumulation

$$C = \sum_{i=0}^{m} X_i \tag{7}$$

where Xi is given by

$$Xi= bi . A^i \tag{8a}$$

For $A^0 = A$, and $A^i = [\alpha^i . A \bmod f(\alpha)]$. Thus A^i can be obtained from A as

$$A^i= a_{m-1} \alpha^m +a_{m-i-1} \alpha^{m-1} +\ldots..+ a_{m-i+2} \alpha + a_{m-i+1} \tag{8b}$$

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

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

The partial product generation and modular reduction are performed according to the above equations.

Equation 2.9 can be expressed as

$$A^{i+1} = [a_0^i. \alpha + a_1^i . \alpha^2 + \ldots + a_m^i . \alpha^{m+1}] \bmod f(\alpha) \tag{10a}$$

where

$$A^i = \sum_{j=0}^{m} a_j^i \alpha^j \tag{10b}$$

Substituting 2.3 into 2.10a, A^{i+1} can be obtained as

$$A^{i+1} = a_0^{i+1} + a_1^{i+1} . \alpha + \ldots + a_m^{i+1} . \alpha^m \tag{11a}$$

where

$$a_0^{i+1} = a^i_m \tag{11b}$$

$$a_j^{i+1} = a^i_{j-1} \tag{11c}$$

By using the above equations, we can derive the proposed linear systolic structure.

3.Systolic Design

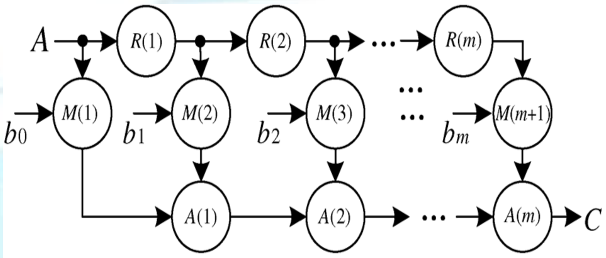

Fig.1 Signal Flow Graph

Figure 1 shows the signal flow graph based on the algorithm described in section 2. For systolic implementation of multiplication over GF (2^m), the operations of equations (7), (8) and (11) have to be done recursively. Each recursion consists of three steps, i.e., modular reduction, bit- multiplication, and bit-addition. Equations (7), (8) and (11) can be represented by the SFG shown in Fig 1. It consists of m modular reduction nodes R(i) and m addition nodes A(i) for $1 \leq i \leq m$, and (m+1) multiplication nodes M(i) for $1\leq i\leq m+1$. Node R(i) performs 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 equation (7).

4. Cut Set Retiming

Retiming is a transformation technique used to change the locations of delay elements in a circuit without affecting the input/output characteristics of the circuit. A cut set is a set of edges that can be removed from the graph to create 2 disconnected sub graphs. Critical path is defined to be the path with the longest computation time among all paths that contain zero delay. The lower bound on the clock period of the circuit can be achieved by retiming.

Figure.2 shows cut-set retiming of the SFG. Here the cut-set retiming technique reduces 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. From the figure, it can be observed that the cut-set retiming allows to perform a reduction operation, 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. Thus the critical-path is not larger than T_X (same as T_A).

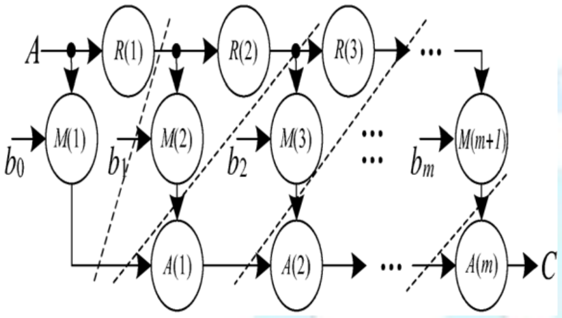

Fig.2 Cut set Retimed Signal Flow Graph

5. Proposed Systolic Structure

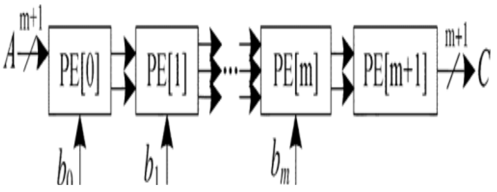

Fig.3 Proposed Systolic Array

The basic design of a systolic multiplier thus derived is shown in Fig.3. It consists of (m+2) PEs and the structure of the PEs are shown in Fig.4. During each cycle period, the regular PE (from PE [2] to PE [m-1]) not only performs the modular reduction operation according to equation (11), but also performs the bit-multiplication and bit-addition operations concurrently.

The internal structure of PE0/PE1, general PE, AND cell, XOR cell and BSC are given respectively in Figure 4.

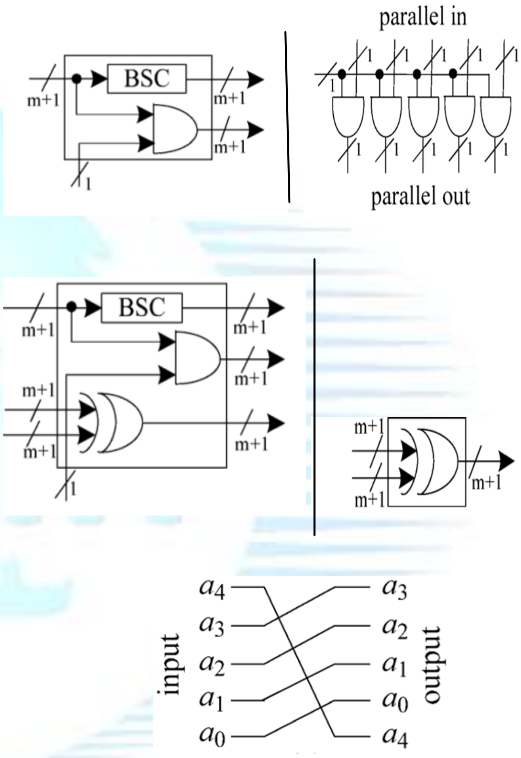

Fig.4 Internal Structure of PEs

6. Shared Register Low Latency Systolic Structure

The product term in equation (7) can be split into two terms like

$$C = \sum_{i=0}^{m/2} X_i + \sum_{i=m/2+1}^{m} X_i \qquad (12)$$

In the above equation, first term contains m/2 +1 partial products and the other has m/2 partial products. Thus the systolic design can be divided in to two systolic branches, and an addition cell is required to perform the final addition. The structure of PEs is of same as given in figure 4. It is seen that the latency of the structure is only (m/2 + 3) cycles.

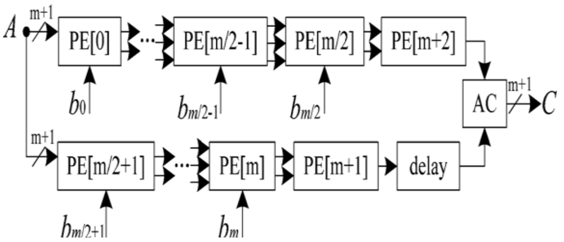

Fig.5 Low Latency Systolic Structure

In figure 5, the two systolic branches share the same input operand and the PEs in both branches does the same operation, a more efficient structure can be made using the register sharing technique. It combines two regular PEs together by sharing one input operand. The whole structure requires only ($2.5m^2$+6.5m+4) bit registers. The latency of the structure is (m/2) + 3 cycles.

As the PEs in two branches perform the same operation, the two PES can be combined together into a single PE, which leads into more efficient structure as shown in figure 6.

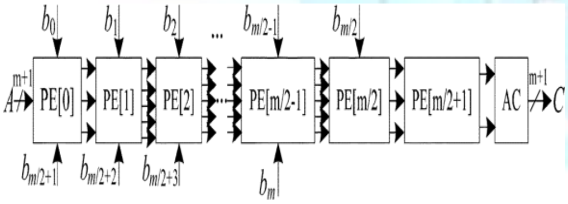

Fig.6 Register Sharing Systolic structure

7. Modified Systolic Structure

The proposed systolic structure can be further modified by decomposing the product term in equation (7) in to four systolic branches as given in equation (13). The corresponding design is shown in figure 7.

$$C = \sum_{i=0}^{m/4-1} X_i + \sum_{i=m/4}^{m/2-1} X_i + \sum_{i=m/2}^{3m/4-1} X_i + \sum_{i=3m/4}^{m} X_i \qquad (13)$$

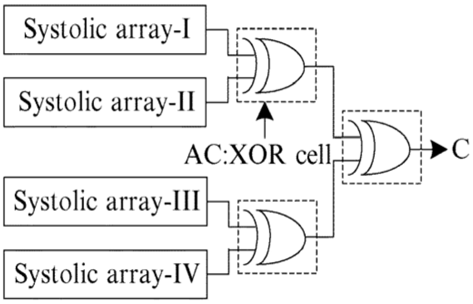

Fig.7 Modified Systolic Structure

Again using the register share technique, the two systolic branches can be merged in to a single branch. The combined systolic structure is given in figure 8. All PEs make use of the basic internal structures shown in figure 4. This design requires only (m/4) + 4 cycles of latency.

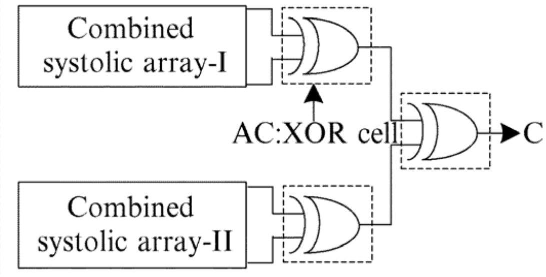

Fig.8 Combined Systolic Structure

8. Results

The proposed design and the modified design have been coded in Verilog and the simulation results are shown in the following graphs.

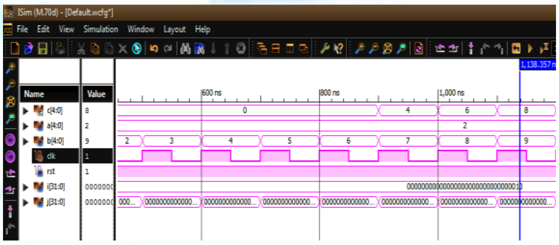

Fig.10 Simulation result of proposed structure (m=4)

The simulation output shows a latency of 6 cycles which is equal to (m+2) cycles.

The simulation result of low latency structure is shown in figure 11.

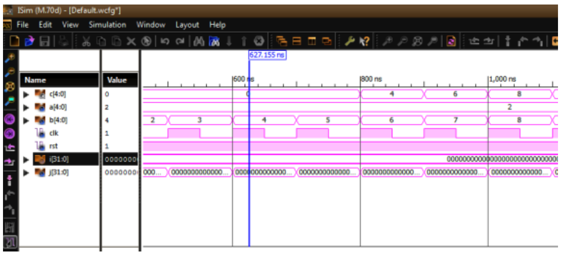

Fig.11 Simulation results of low latency structure (m=4)

The result shows a latency of 5 cycle, which is equal to (m/2+3) cycles. The latency less when compared with the proposed systolic design.

Furthermore, the output of modified systolic structure in fig.8 (for m=20), require a latency of only 9 cycles compared to 22 cycles in the proposed design.

Table 1- Area and time complexities when m=4

Design	LUTs utilization factor	Latency	Registers Used	Critical Path
Proposed systolic array (Fig.3)	3%	6	59%	Tx
Low latency systolic array (Fig.5)	2%	5	59%	Tx
Register shared systolic array (Fig.6)	2%	5	55%	Tx

The table 1 shows the synthesis results of various designs. The improvements are significant when m is a large number.

9. Conclusion

Modified systolic design for the multiplication over $GF(2^m)$ based on irreducible AOP using register sharing technique is proposed. Using cut-set retiming we have been able to reduce the critical path to one XOR gate delay and using register sharing technique, we achieved a low-latency bit-parallel systolic multiplier. Compared with the existing systolic structures for bit-parallel realization of multiplication over $GF(2^m)$, the proposed design is found to involve less area, shorter critical-path and lower latency.

References

[1] C. Paar, “Low complexity parallel multipliers for Galois fields based on special types of primitive polynomials,” in *Proc. IEEE Int. Symp. Inform. Theory*, 1994, p. 98.

[2] C. H. Kim, C.-P. Hong, and S. Kwon, “A digit-serial multiplier for finite field GF(2m) ,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 476–483, 2005.

[3] 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.

[4] 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.

[5] 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

[6] 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.

[7] J. Xie , P.K Meher and J.He “Low Complexity Multiplier for GF(2m) Based on All-One Polynomials”, *IEEE Trans, Very large Scale Integr,(VLSI)*,2011

[8] 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.

[9] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley, 1999.

[10] P. K. Meher, “Systolic and non-systolic scalable modular designs of finite field multipliers for Reed-Solomon Codec,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 6, pp. 747–757, Jun. 2009.

[11] 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.

[12] 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.