Volume 3, Issue 12, December 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

GF(2^m) Based Polynomials Multiplier Using An Efficient Systolic Multiplier

S. Nazeer Hussain, D. Himaja Reddy, A. Maheswara Reddy
Assistant Professor, Dept: ECE,
AITS Kadapa.
Kadapa (DT), India

Abstract—This paper presents an area-time-efficient systolic structure for multiplication over GF(2^m) based on irreducible all-one polynomial (AOP). A novel cut-set retiming to reduce the duration of the critical-path to one XOR gate delay is used. Also the systolic structure can be decomposed into two or more parallel systolic branches, where the pair of parallel systolic branches has the same input operand, and they can share the same input operand registers. The field-programmable gate array synthesis results shows that the proposed design provides significantly less area-delay complexities over the best of the existing designs.

Keywords:

I. INTRODUCTION

Finite field multipliers over GF(2^m) have wide applications in elliptic curve cryptography (ECC) and error control coding systems. Polynomial basis multipliers are popularly used because they are relatively simple to design, and offer scalability for the fields of higher orders. Efficient hardware design for polynomial-based multiplication is therefore important for real-time applications. 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. Irreducible AOPs are not abundant. They 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. The AOP-based multipliers can be used for the nearly AOP (NAOP) which could be used for efficient realization of ECC systems. AOP-based fields could also be used for efficient implementation of Reed-Solomon encoders. Besides, the AOP-based architectures can be used as a kernel circuit for field exponentiation, inversion, and division architectures.

Systolic design is a preferred type of specialized hardware solution due to its high-level of pipeline ability, local connectivity and many other advantageous features. A bit-parallel AOP-based systolic multiplier has been suggested by Lee *et al.*. A low-complexity bit-parallel systolic Montgomery multiplier has been recently suggested. Very recently, 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. The proposed algorithm not only facilitates sharing of registers by the neighboring PEs to reduce the register complexity but also helps reducing the latency. Cut-set retiming allows to introduce certain number of delays on all the edges in one direction of any cut-set of a signal flow-graph (SFG) by removing equal number of delays on all the edges in the reverse direction of the same cut-set .When all the edges are in a single direction, one can introduce any desired number of delays on all the edges of any cut-set of an SFG. Therefore, this technique is highly useful for pipelining digital circuits to reduce the critical path. In this paper, we have proposed a novel cut-set retiming approach to reduce the clock-period. The proposed structure is found to involve significantly less area-time-power complexity compared with the existing designs. The rest of this paper is organized as follows. The proposed algorithm for finite field multiplication over GF(2^m) based on AOP is derived in Section II. In Section III, the proposed structure is presented. In Section IV, we have listed the complexities and compared them with those of the existing structures. Finally the conclusion is given in Section V.

II. ALGORITHM

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

$$X = X_{m-1}\alpha^{m-1} + X_{m-2}\alpha^{m-2} + \cdots + X_1\alpha + X_0 \qquad (1)$$

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

$$f(\alpha)+\alpha f(\alpha)=(\alpha^m+\alpha^{m-1}+\cdots+\alpha+1)+\alpha(\alpha^m+\alpha^{m-1}+\cdots+\alpha+1)=\alpha^{m+1}+1=0. \quad (2)$$

$$\alpha^{m+1}=1. \quad (3)$$

Therefore, we have

This property of AOP 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 $X = x_0 + x_1\alpha + \ldots + x_m\alpha^m$, where $x_i \varepsilon$ GF(2), and $\{\alpha^m, \alpha^{m-1}, \alpha^{m-2}, \ldots, \alpha, 1\}$ is the extended polynomial basis. Similarly, if A,B,C ε GF(2^m), they can be represented by the extended polynomial basis as

$$A=\sum_{j=0}^{m}a_j\alpha^j, \quad B=\sum_{j=0}^{m}b_j\alpha^j, \quad C=\sum_{j=0}^{m}c_j\alpha^j \quad (4)$$

where a_j, b_j, and $c_j \varepsilon$ 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 can have

$$C = A \cdot B \bmod f(\alpha) \quad (5)$$

which can be decomposed to a form

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

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

$$C=\sum_{i=0}^{m}X_i \quad (7)$$

where X_i is given by

$$X_i = b_i \cdot A^i \quad (8a)$$

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

$$A^i=a_{m-i}\alpha^m+a_{m-i-1}\alpha^{m-1}+\cdots+a_{m-i+2}\alpha+a_{m-i+1}. \quad (8b)$$

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

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

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

$$A^{i+1}=\left[a_0^i\cdot\alpha+a_1^i\cdot\alpha^2+\cdots+a_m^i\cdot\alpha^{m+1}\right] \bmod f(\alpha) \quad (10a)$$

$$A^i=\sum_{j=0}^{m}a_j^i\alpha^j. \quad (10b)$$

Where

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

$$A^{i+1}=a_0^{i+1}+a_1^{i+1}\cdot\alpha+\cdots+a_m^{i+1}\cdot\alpha^m \quad (11a)$$

$$a_0^{i+1}=a_m^i \quad (11b)$$

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

Where

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

$$a_j^{i+l}=\begin{cases}a_{m-l+j+1}^i, & \text{for } 0\le j\le l-1\\ a_{j-l}^i, & \text{otherwise.}\end{cases} \quad (12)$$

such that

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. PROPOSED STRUCTURE

In this section, we derive a basic systolic design followed by the proposed register sharing structure.

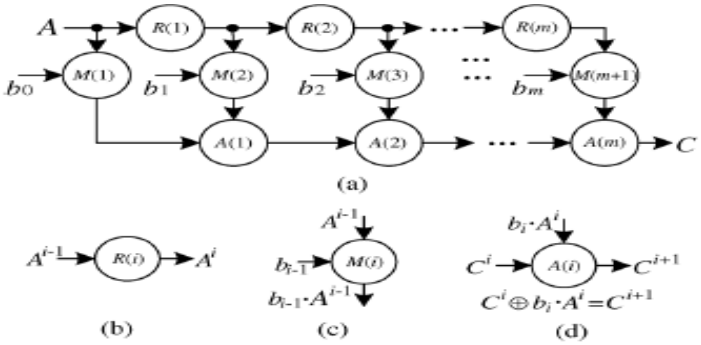


Fig. 1. SFG of the algorithm. (a) The SFG. (b) Function of node R(i). (c) Function of node M(i). (d) Function of node A(i) .

A. Basic Systolic Design

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 $1 \leq i \leq m$, and (m+1) multiplication nodes M(i) for $1 \leq i \leq m+1$. The functions of these nodes are shown in Fig. 1(b)–(d). 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 (7), as shown in Fig. 1(d), where C^i is the partial result available to the node.

Generally, we can introduce a delay between the reduction node and its corresponding bit-multiplication and bit-addition nodes, as shown in Fig. 2(a), 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. Therefore, we introduce a critical-path which is not larger than T_X, as shown in Fig. 2(b). To derive the basic design of a systolic multiplier, we have shown the formation of PE of the retimed SFG in Fig. 2(c). 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 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 (from PE[2] to PE[m-1]) not only performs the modular reduction operation according to (11), but also performs the bit-multiplication and bit-addition operations concurrently. The detail circuit of a regular PE is shown in Fig. 4.

The regular PE, as shown in Fig. 4(a), 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. The structure of PE[1] of Fig. 3 is shown in Fig. 4(b). It 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. Fig. 4(c) shows an example of AND cell for m = 4. The PE[m+1] of the systolic structure in Fig. 3 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). We have shown an example of the structure of BSC (of PE[1] of Fig. 4) in Fig. 4(e) for m = 4. Note that according to (12), one can obtain A^i directly from A^0 for $1 \leq i \leq m$, i.e., every PE of the structure of Fig. 3 can have the same input operand A^0, and A^i can be obtained from the BSC after A^0 is fed as input. 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$\{T_A, T_X\} = T_X$.

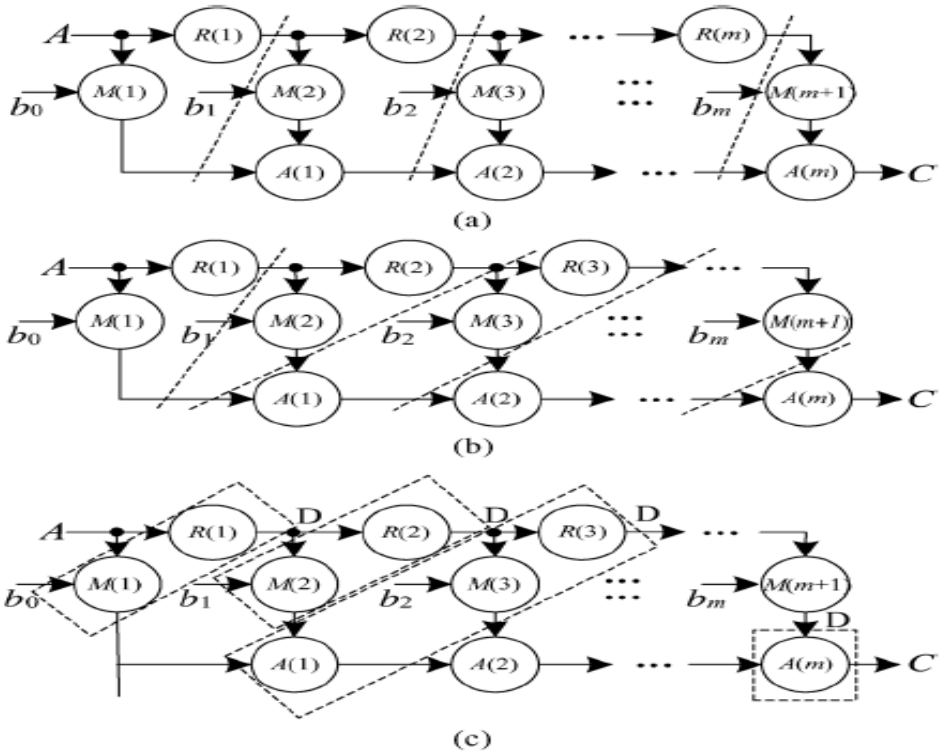


Fig. 2. Cut-set retiming of the SFG. (a) Cut-set retiming in a general way. (b) Proposed cut-set retiming. (c) Formation of PE. "D" denotes unit delay.

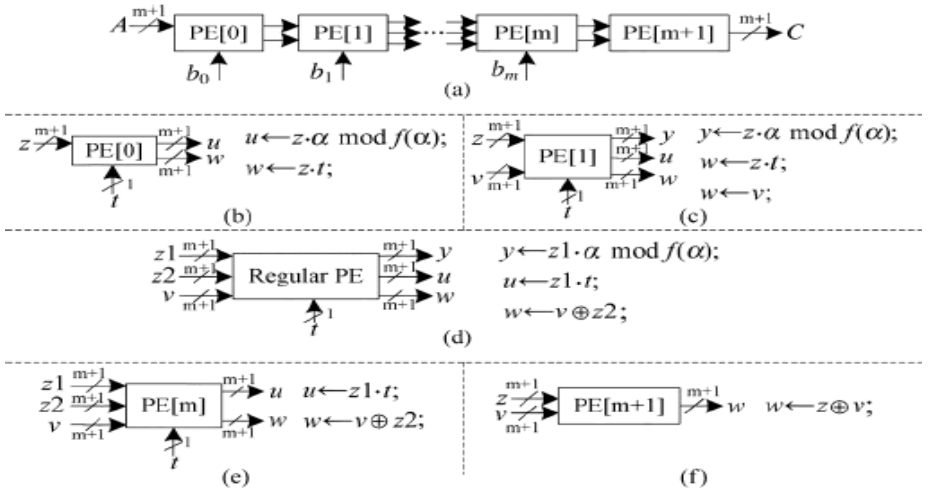


Fig. 3. Proposed systolic structure. (a) Systolic design. (b) Function of PE[0]. (c) Function of PE[1]. (d) Function of regular PE (from PE[2] to PE[m-1]). (e) Function of PE[m]. (f) Function of [m+1].

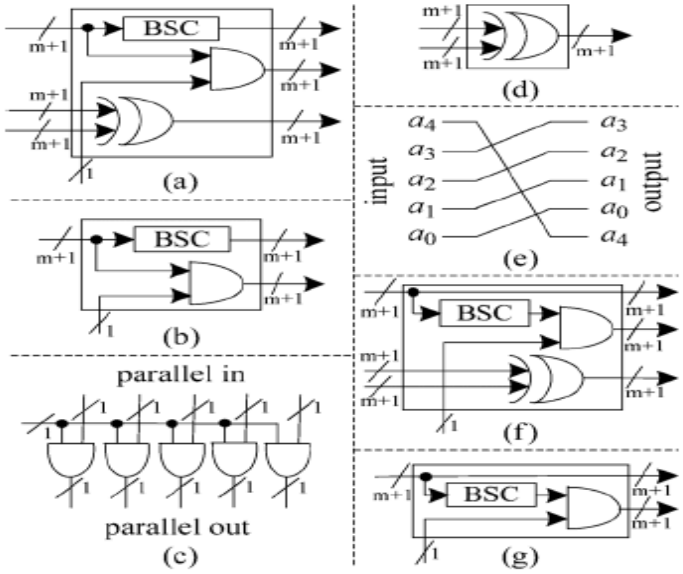


Fig. 4. Structure of PEs. (a) Internal structure of a regular PE. (b) Internal structure of PE[0] of Fig. (c) An example of AND cell for m = 4. (d) structure of the AC. (e) Structure of BSC where m = 4. (f) Alternate structure of a regular PE. (g) Alternate structure of PE[0].

B. Shared-Register Low-Latency 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 $0 \le r \le l$. For example, if we choose P = m/2, then l = 2, r = 1, (7) can be rewritten as

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

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, as shown in Fig. 5(b). 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.

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. The circuit of its regular PE (from PE[2] to PE[m/2-1]) is shown in Fig. 6(c). It combines two regular PEs of Fig. 5(a) together by sharing one input-operand- transfer. The other PEs needs some minor modifications, as shown in Fig. 6(b), (d) and (e) respectively. The function of AC is the same as that in Fig. 5. 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.

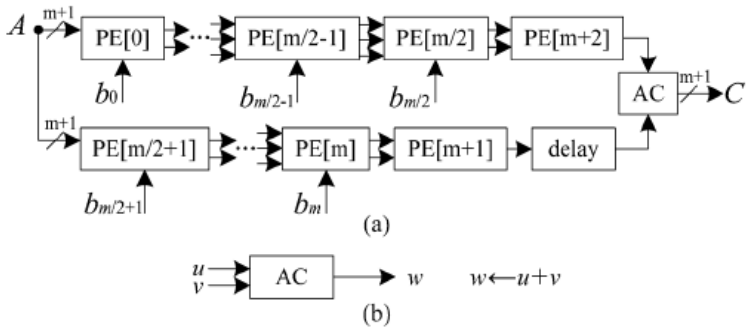


Fig. 5. Proposed low latency systolic structure. (a) The systolic structure. (b) Function of the AC

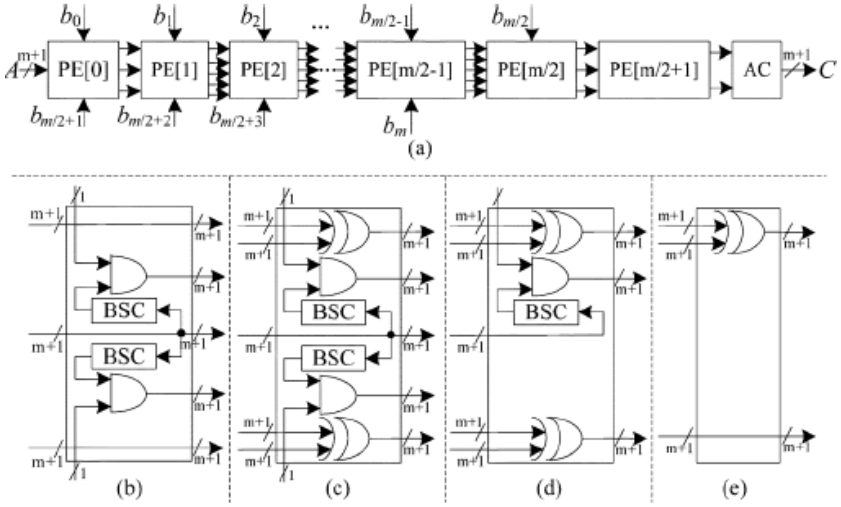


Fig. 6 Low-latency register-sharing systolic structure. (a) The systolic structure. (b) Structure of PE[1]. (c) Structure of a regular PE (from PE[2] to PE[m/2-1]). (d) Structure of PE[m/2]. (e) Structure of PE[m/2 + 1].

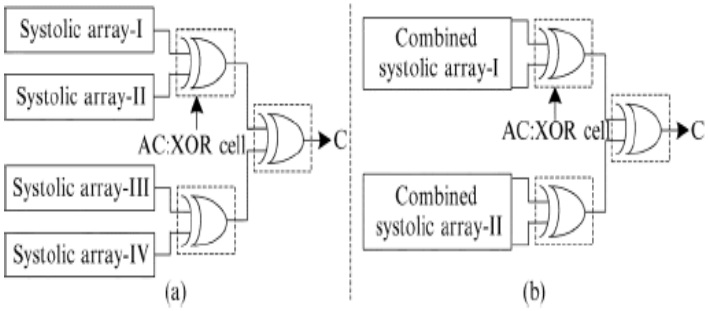


Fig. 7 Improved low-latency systolic structure. (a) The proposed systolic array merging. (b) Improved systolic structure.

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

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

TABLE I
Area and Time complexity

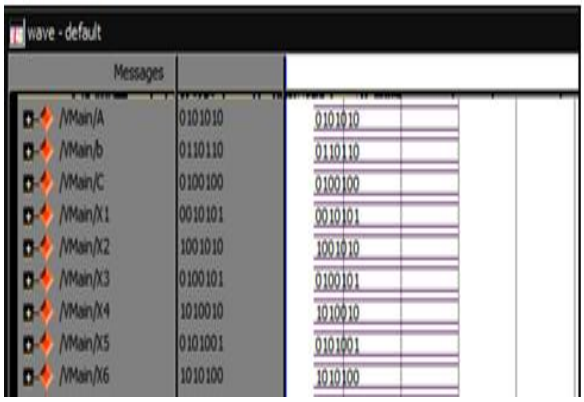
Design	AND	MUX	XOR (2-input)	XOR (3-input)	Latency	Critical-path (nS)
Existing bit parallel design	(m)	0	(m-1)	0	m+1	T_X+T_A
Proposed Fig. 6	(m+1)	0	(m)	0	(m/2)+3	T_X
Proposed Fig. 7	(m+1)	0	(m)	0	(m/4)+4	T_X

Following the same approach as the one used to derive the structure of Fig. 5, we can have the design in Fig. 7(a), 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(b). the design of Fig. 7(b) requires only [(m/4) + 4]cycles of latency. When m is a large number, l and P can be chosen as l=P=(m+1) to obtain an optimal realization

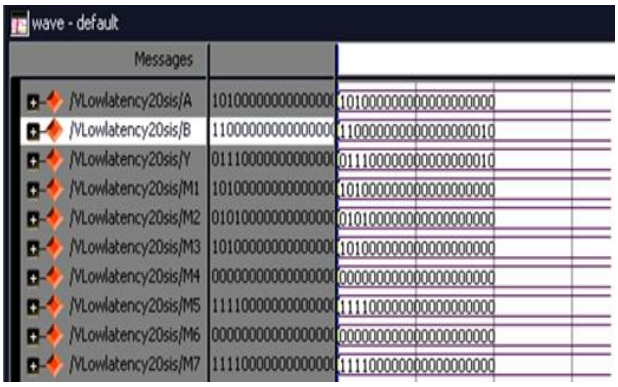
IV. HARDWARE AND TIME COMPLEXITY

The proposed structure (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. 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 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 (Figs. 6 and 7) and the existing design are listed in Table I.

It can be seen that the proposed design outperforms the existing designs. The proposed designs are coded in Verilog, simulated using ModelSim 6.4c and are synthesised using Spartan-3E FPGA.



Simulation result for m=6



Simulation result for m=20

The comparison table II gives the comparison between area and time complexity of the existing and proposed designs for m value of 6

TABLE II
Comparison of area and time complexity for m = 6

.Design	LUTs	Delay (ns)	Critical-path duration (ns)
Existing bit parallel design	30	9.014	7.022
Proposed Fig. 4 (systolic)	35	9.239	5.753
Proposed Fig. 5 (low latency)	35	8.118	5.753
Proposed Fig. 6 (register sharing)	21	7.105	5.753

Table III gives the comparison between the area and time complexities of the existing and proposed designs for the value of m equal to 20.

TABLE III:Comparison of area and time complexity for m=20

Design	LUTs	Delay (ns)	Critical-path duration (ns)
Existing bit parallel design	380	16.401	7.280
Proposed Fig. 5 (low latency)	399 (2,835)	13.798	5.753
Proposed Fig. 6 (register sharing)	399 (2,772)	13.826	5.753
Proposed Fig. 7 (improved low latency register sharing)	378	9.626	5.753
Proposed Fig. 7 (further improved register sharing	315	9.430	5.753

V CONCLUSION

Efficient systolic design for the multiplication over GF(2^m) based on irreducible AOP is proposed. By novel cut-set re-timing 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 realization of multiplication over GF(2^m), the proposed one is found to involve less area, shorter critical-path and lower latency. From FPGA synthesis results we find that the proposed design involves significantly less ADP than the existing designs. Moreover, out 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] C. H. Kim, C.-P. Hong, and S. Kwon, "A digit-serial multiplier for finite field ," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 476–483, 2005.

[3] 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.

[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] 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.

[6] C.-Y. Lee, E.-H. Lu, and J.-Y. Lee, "Bit-parallel systolic multipliers for fields defined by all-one and equally spaced polynomials,"*IEEE Trans. Computers*, vol. 50, no. 6, pp. 385–393, May 2001.

[7] 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.

[8] www.doc.ic.ac.uk.

[9] S. Talapatra, H. Rahaman, and J. Mathew, "Low complexity digit serial systolic Montgomery multipliers for special classes of GF(2^m) ", IEEE trans. Very Large Scale Integr. (VLSI) Syst., vol.18, no. 5, pp. 847-852, May 2010.

[10] 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.

ACKNOWLEDGMENT

S Nazeer Hussain, born in Kadapa, A.P., India in 1989. He received B.Tech Degree in Electronics and Communication Engineering from J.N.T University Anantapur, India. He received M.Tech (VLSI SYSTEM DESIGN) Degree from Annamacharya Institute of Technology and Sciences, Rajampet, A.P., India. His research interests include VLSI, Digital Signal Processing and Digital Design.

D Himaja Reddy, born in Kadapa A.P.,India. She received B.Tech Degree in Electronics and Communication Engineering from SV University Tirupati A.P., and India. She received M.Tech (VLSI SYSTEM DESIGN) Degree from Annamacharya Institute of Technology and Sciences, Rajampet, A.P., India. Her research interest is in VLSI Technology.

A Maheswara Reddy, born in Kadapa A.P., India. He received B.Tech Degree in Electronics and Communication Engineering from J.N.T University Anantapur . He received M.Tech (EMBEDDED SYSTEMS) Degree from J.N.T University Hyderabad, A.P.,India. His research interest is in Digital Signal Processing, Embedded Systems.

