ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 5, Issue, 12, pp. 6241-6246, December, 2015

International Journal of
DEVELOPMENT RESEARCH

Full Length Research Article

LOW-COMPLEXITY MULTIPLIER FOR GF (2m) BASED ON ALL-ONE POLYNOMIALS

[1,*]Pardhasarathy, T. and [2]Vijayabhaskar, C.

[1]M. Tech (VLSI), 11F6D5707, SIETK, Puttur, India
[2]Department of ECE, SIETK, Puttur, India

ARTICLE INFO

Article History:
Received 26th September, 2015
Received in revised form
29th October, 2015
Accepted 16th November, 2015
Published online 30th December, 2015

Key Words:
All-one polynomial,
Finite field, Systolic design.

ABSTRACT

This paper presents an area-time-efficient systolic structure for multiplication over GF(2m) based on irreducible all-one polynomial(AOP). We have used a novel cut-set retiming to reduce the duration of the critical-path to one XOR gate delay. It is further shown that 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. From the application-specific integrated circuit and field-programmable gate array synthesis results we find that the proposed design provides significantly less area-delay and power-delay complexities over the best of the existing designs.

INTRODUCTION

Finite field multipliers over ($GF2^m$) 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 akernel circuit for field exponentiation, inversion, and division architectures.

Corresponding author: Pardhasarathy, T.
M. Tech (VLSI), 11F6D5707, SIETK, Puttur, India.

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.*. Another efficient systolic design is presented. In a low-complexity bit-parallel systolic Montgomery multiplier has been 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 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 neighbouring PEs to reduce the register complexity but also helps reducing the latency. Cut-set retiming allows introducing 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 ($GF2^m$) 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.

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 $GF2^m$,(m+1) is prime and 2 is the primitive modulo(m+1).The set $\{\alpha^{(m-1)}, \alpha^{(m-2)}, \ldots\ldots, \alpha, 1\}$ forms the polynomial basis (where α is a root of $f(x)$ such that an element _ of the binary field can be given by

$$X=X_{m-1}\,\alpha^{(m-1)}+X_{m-2}\,\alpha^{(m-2)}+\ldots\ldots+\mathbf{X_1}\,\alpha+X_0 \quad (1)$$

Where $X \in$ GF(2) for i=m-1,…..,2,1,0.

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

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

$$\alpha^{m+1}=0 \quad (3)$$

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

$$X=X_0+\mathbf{X_1}\,\alpha+\ldots\ldots\ldots\ldots\ldots+ X_m\,\alpha^{(m)}$$

where Xi $\in$ GF(2)

And set $\{\alpha^{(m)}, \alpha^{(m-1)}, \ldots\ldots, \alpha, 1\}$ is the extended polynomial.

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

Where aj,bj and cj $\in$ GF(2) for $0 \le j \le m-1$ and

a_m=0,b_m=0 and c_m=0.

If c is a product of elements of A and B, then we can have

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

Which can be decomposed to form

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

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

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

Where Xi is given by

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

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

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

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

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

Where

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

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

$$A^{i+1}=[\,a_0^{i+1}+a_1^{i+1}.\alpha+\ldots\ldots\ldots+a_m^i.\alpha^{m+1}] \quad (11a)$$

Where

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

$$a_j^{i+1}=a_{j-1}^i,\text{for } 0 \le l \le j\text{-}1 \quad (11c)$$

It is also possible to extend (11) further to obtain A^{i+1}

Directly from A^i for $1 \le l \le m$, such that

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

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.

Proposed Structure

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

Basic Systolic Design

For systolic implementation of multiplication over, 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).

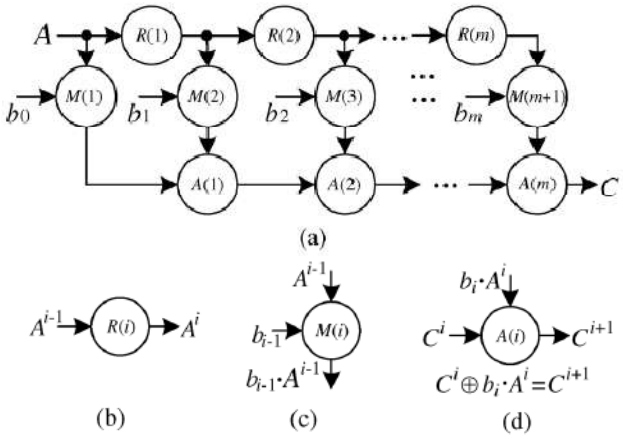


Fig. 1. SFG of the algorithm. (a) The SFG. (b) Function of node. (c)Function of node. (d) Function of node

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 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 with a reduced form of operand , according to (8). Node A(i)performs the bit-addition operation according to (7), as shown in Fig. 1(d), where 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,T_X and 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.

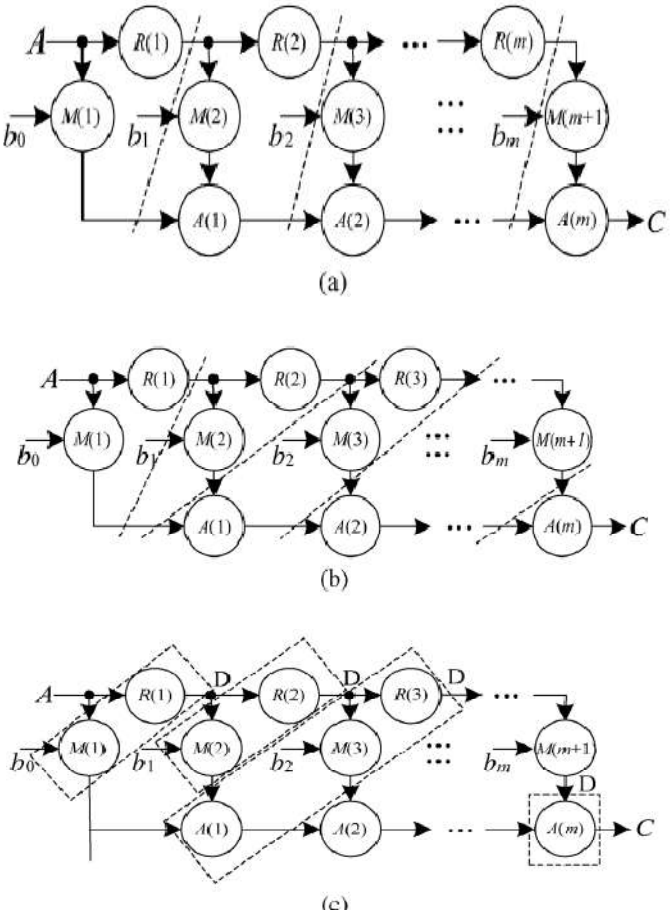


Fig. 2. Cut-set retiming of the SFG. (a) Cut-set retiming in a general way. (b)Proposed cut-set retiming. (c) Formation of PE. "D" denotes unit delay

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 of PE[m+1] 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 -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, Fig. 4. Structure of PEs. (a) Internal structure of a regular PE. (b) Internal structure of PE [0] of Fig. 4. (c)An example of AND cell for .(d) Structure of the AC. (e) Structure of BSC where. (f) Alternate structure of a regular PE. (g) Alternate structure of PE[0].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 $\max\{T_A,T_X\}=T_X$.

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.

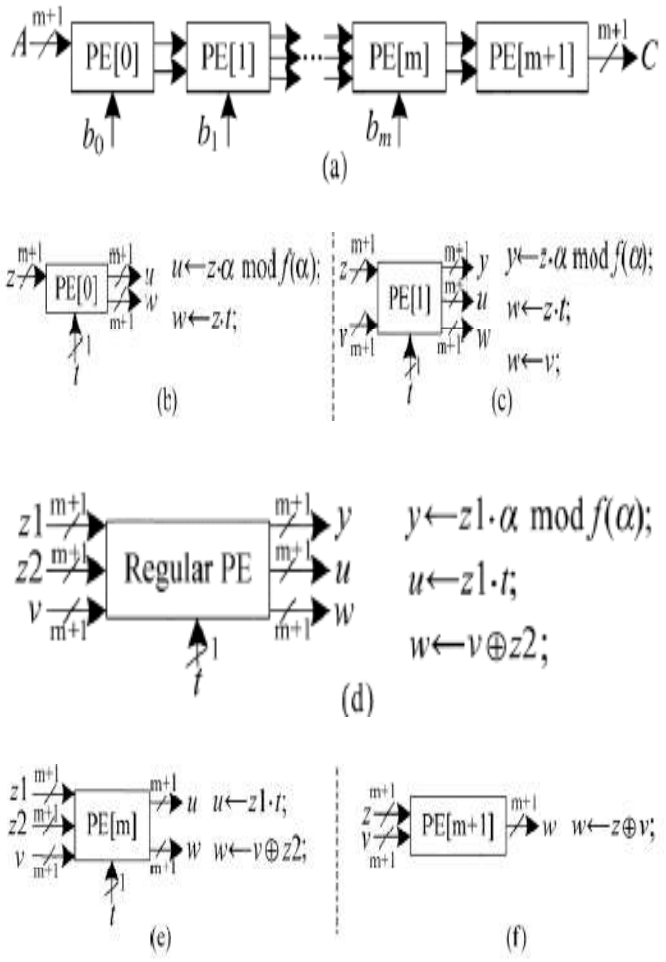


Fig. 3. Proposed systolic structure. (a) Systolic design. (b) Function of PE[0].(c) Function of PE[1]. (d) Function of regular PE (from PE[2] to PE[m-1] .(e) Function ofPE[m] . (f) Function of PE[m+1]

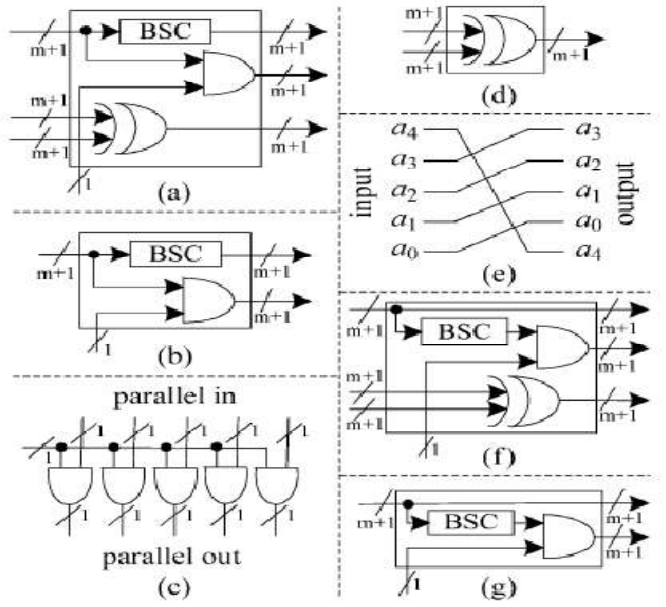


Fig. 4. Structure of PEs. (a) Internal structure of a regular PE. (b) Internal structureof PE[0] of Fig. 4. (c) An example of AND cell for m=4 (d) Structure ofthe AC. (e) Structure of BSC where m=4. (f) Alternate structure of a regularPE. (g) Alternate structure of PE[0]

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+l)=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, (7) can be rewritten as.

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

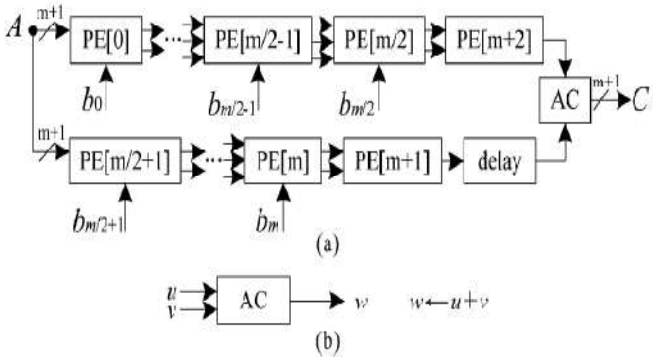


Fig. 5. Proposed low latency systolic structure. (a) The systolic structure. (b)Function of the AC

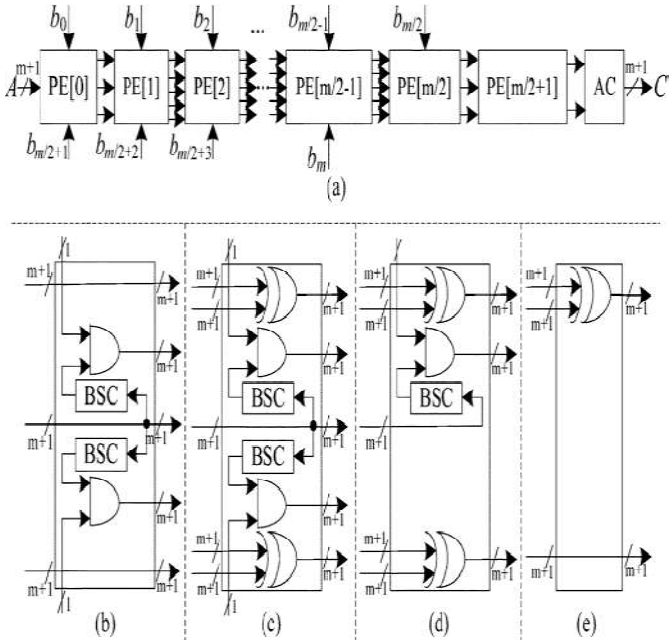


Fig. 6. Low-latency register-sharing systolic structure. (a) The systolic structure. (b) Structure of PE[1]. (c) Structure of a regular PE (from PE[2] to). (d) Structure of . (e) Structure of

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 (m/2)+2 of PEs and the lower branch consists of (m/2)+1PEsand 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 inFig.5(The structure has the PEs of the same complexity as those in Fig. 3, but the latency of structure is only (m/2)+3cycles.

It is observed that the two systolic branches in Fig. 5 share the same in put operand , 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 need some minor modifications, as shown in Fig. 6(b), (d) and (e), respectively.

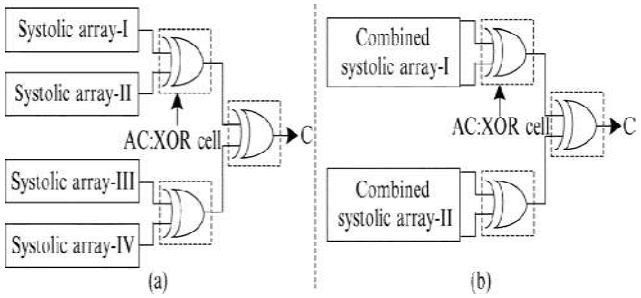


Fig. 7. Improved low-latency systolic structure. (a) The proposed systolic arraymerging. (b) Improved systolic structure

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

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

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)+4cycles of latency. When is a large number, and can be chosen as(15)to obtain an optimal realization.

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

Hardware 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 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.

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 (see Figs. 6 and 7) and the existing designs of [11]–[16] are listed in Table I.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.

The digit-serial structures of [12] and [16] yield one product word in m/l and (2m/l-1) clock-periods, respectively, while the proposed structure produces one product word in every clock-period. 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 (see Fig. 7) 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 are listed in Table II.

Table 1. Area and Time Complexity

Design	AND	MUX	XOR(3-INPUT)	XOR(2INPUT)	Register	Latency	Criticalpath
[11]	$(m+1)^2$	$(m+1)^2$	0	0	$2(m+1)^2$	m+2	T_A+T_F#
[13]	$(m+1)^2$	0	0	$(m+1)^2$	$4(m+1)^2$	m+1	T_A+T_x
[14]	m^2	0	0	2m	$(5m^2+7m-6)/2$	m/2+1	T_A+T_{3x} *
[15]	$(m+1)^2$	0	$(m^2-m)/2$	$(m+1)^2$	$3(m+1)^2$	m+1	T_A+T_x
[16]	Lm	2m	0-m	Lm	2m+m/L-L-1	2m/L-1	$TA+[L-1+\log_2 L]T_X$
[12]	Lm	0	0	L(2m-1)	2m	m/L	$T_M+T_A+(\log_2 L)T_X$
FIG.6	$(m+1)^2$	0	0	$(m+1)^2$	$(5/2xm^2)+(13/2xm)+4$	m/2+3	T_x
FIG.7	$(m+1)^2$	0	0	$(m+1)^2$	$(5/2xm^2)+(1/2xm)+7$	m/4+4	T_x

T_F : Delay of a T flipflop
T3x: Delay of 3-input Xor gate
T_m: The delay of a 2:1 mux

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

Table 2. Comparison of area and time complexity for m=20

Designs	Area	Act	Power	ADP	PDP	Throughput
15	19115	0.17	4.158	3250	0.71	1
16	2463.9	1.35	0.537	3326	0.72	1/5
Fig.7	17871	0.13	3.893	2323	0.52	1

ACT=(Critical path)X(cycles number of required to obtain a result) digit size L=4

Table 3. FPGA synthesis result of proposed and existed designs

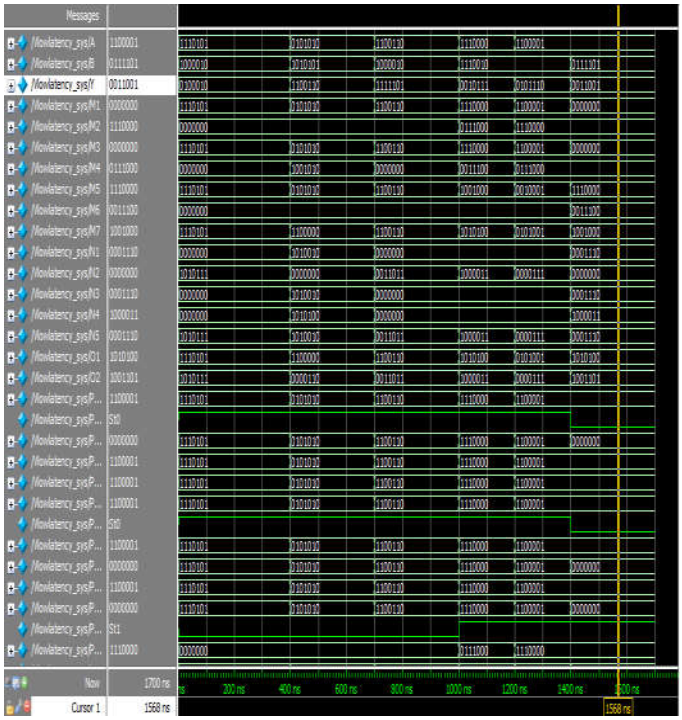
Designs	LE	ACT	PC	ADP	PDP
15	1764	4.2	103.12	7409	433
16	185	35.7	71.62	6607	2557
FIG.7	1736	3.6	94.67	6250	339

Besides, we have synthesized the proposed design (see Fig. 7) and the designs of [15] and [16] for M=20 and implemented on an Altera FPGA: Cyclone-IIEP2C15AF256A7 using Quartus II 9.0. From the synthesis result, as shown in Table III, we find

that the proposed design has lower ADP and less PDP than the existing ones.

Simulation Results

Low latency systolic structure simulation results are shown in Fig.8



Conclusion

Efficient systolic design for the multiplication over GF(2^m)based on irreducible AOP is 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 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. Moreover, our proposed design can be extended to further reduce the latency.

REFERENCES

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

[2] Fan, H. and Hasan, M.A. 2006. "Relationship between montgomeryand shifted polynomial basis multiplication algorithms," *IEEETrans. Computers*, vol. 55, no. 9, pp. 1202–1206, Sep. 2006.

[3] Wang, C.L. and Lin, J.L. 1991. "Systolic array implementation of multipliersfor finite fields ," *IEEE Trans. Circuits Syst.*, vol. 38, no. 7,pp. 796–800, Jul. 1991.

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

[5] Kim, C.H., Hong, C.P. and Kwon, S. 2005. "A digit-serial multiplier forfinite field ," *IEEE Trans. Very Large Scale Integr. (VLSI)Syst.*, Vol. 13, no. 4, pp. 476–483, 2005.

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

[7] Wu, H. 2008. "Bit-parallel polynomial basis multiplier for new classes offinite fields," *IEEE Trans. Computers*, vol. 57, no. 8, pp. 1023–1031, Aug. 2008.

[8] Fenn, S., Parker, M.G., Benaissa, M. and Taylor, D. 1997. "Bit-serial multiplication in using all-one polynomials," *IEE Proc. Com. Digit. Tech.*, vol. 144, no. 6, pp. 391–393.

[9] Chang, K.Y., Hong, D. and Cho, H.S. 2005. "Low complexity bit-parallel multiplier for defined by all-one polynomials using redundantre presentation," *IEEE Trans. Computers*, vol. 54, no. 12, pp.1628–1629, Dec. 2005.

[10] Kim, H.S. and Lee, S.W. 2007. "LFSR multipliers over defined by all-one polynomial," *Integr., VLSI J.*, vol. 40, no. 4, pp. 571–578.

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

****