International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.02,Issue.19,
December-2013,
Pages:2319-2324

www.semargroups.org,
www.ijsetr.com

Low-Complexity Multiplier for GF (2^m) Based on All-One Polynomials

DASARI SUBBARAO
Associate professor, Siddhartha Institute of Engineering Technology, Ibrahimpatnam, Hyderabad, TS, India,
E-mail: subbu.dasari@gmail.com.

Abstract: This paper presents an area-time-efficient systolic structure for multiplication over *GF* (2^m) 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.

Keywords: All-One Polynomial, Finite Field, Systolic Design.

I. INTRODUCTION

Finite field multipliers over *GF* (2^m) have wide applications in elliptic curve cryptography (ECC) and error control coding systems [2], [3]. 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 [4]–[6]. 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 [7]. 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 [2], [10]. 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. In a bit-parallel AOP-based systolic multiplier has been suggested by Lee et al. Another efficient systolic design is presented. In a recent paper, 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 *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 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. Proposed Structure in Section II .In Section III, Sequential Polynomial Multiplier .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. PROPOSED STRUCTURE

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

A. Basic Systolic Design

For systolic implementation of multiplication over GF (2^m), the operations can be performed recursively. Each recursion is composed of three steps, i.e., modular reduction of bit-multiplication of bit-addition. Equations 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 node M (i) performs an AND operation of a bit of operand B with a reduced form of operand A, according. Node A (i) performs the bit-addition operation according to, as shown in Fig. 1(d), where *Ci* is the partial result available to the node.

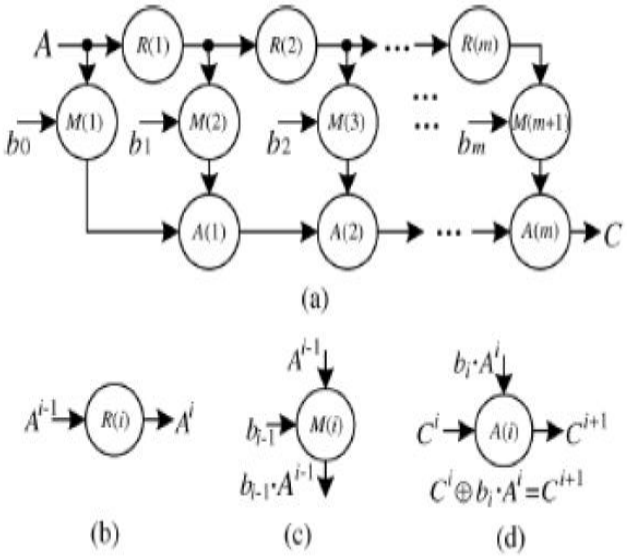

Fig.1. SFG of the algorithm (a) The SFG (b) Function of node R (i) (c) Function of node M (i). (d) Function of node A (i).

Generally, we can introduce a delay between the reduction node and its corresponding bit-multiplication and bit-addition nodes, as shown in Fig. 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 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 to PE[m-1]) not only performs the modular reduction operation 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 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 of Fig. 4) in Fig. 4(e) for m = 4. Note that, one can obtain Ai directly from A0 for $1 \leq i \leq m$, i.e., every PE of the structure of Fig. 3 can have the same input operand A0, and Ai can be obtained from the BSC after A0 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, 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 $\{T_A, T_X\} = T_X$.

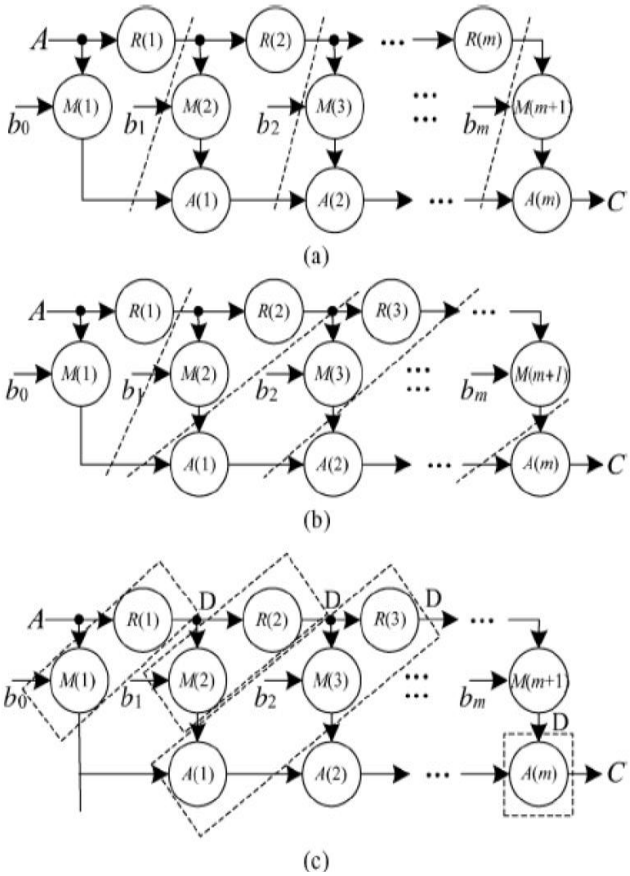

Fig.2. Cut-set retiming of the SFG (a) Cut-set retiming in a general way (b) Proposed cut-set retiming. (c) Formation of PE. "D" denotes unit delay.

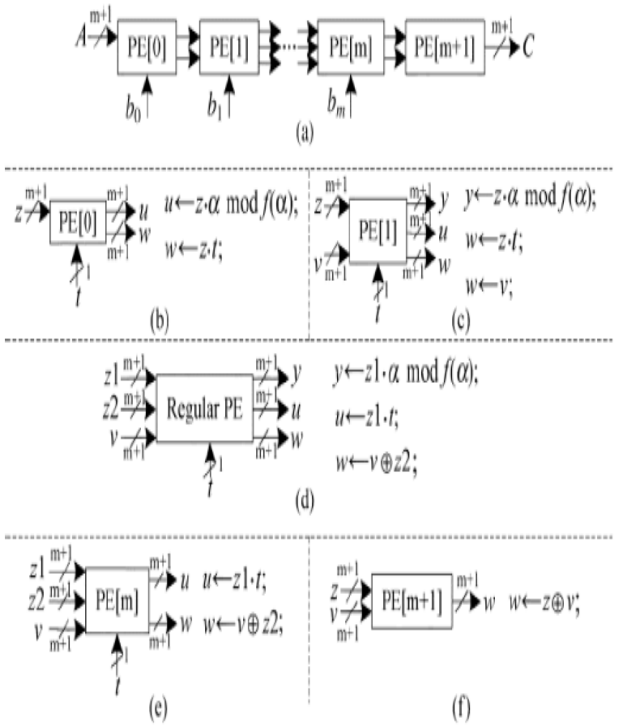

Fig.3. Proposed systolic structure (a) Systolic design (b) Function of PE. (c) Function of PE. (d) Function of regular PE (from PE to PE [m-1]). (e) Function of PE[m]. (f) Function of [m+1].

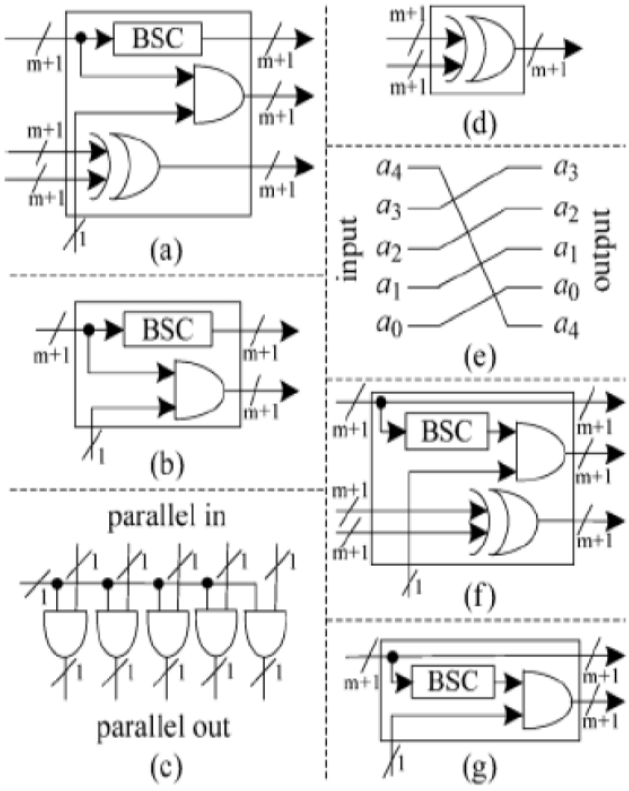

Fig.4. Structure of PEs (a) internal structure of a regular PE. (b) Internal structure of PE [0] of Fig. (c) An example of AND cell for m = 4. (d) Structure of the AC. (e) Structure of BSC where m = 4. (f) Alternate structure of a regular PE. (g) Alternate structure of PE [0].

III. SEQUENTIAL POLYNOMIAL MULTIPLIER

Even with the hRAIK method combinatorial multiplier with long bit sizes are still quite slow and not as small as it is desirable for mobile devices or even wireless sensor nodes. Common approaches use smaller combinatorial multiplication units and serialize the multiplication. Actually the original iterative Karatsuba multiplier (IKM) approach was presented as solution for this purpose. It uses smaller combinatorial multiplication blocks, and applies them repeatedly following the Karatsuba method in order to perform a larger polynomial multiplication the IKM design for a 233 bit multiplication unit presented in the starting point for the investigation concerning improved IKM design. It consists of three main parts: The selection logic selects and combines the factors of the partial multiplication the partial multiplier performs the partial multiplication within one clock cycle, and the accumulation logic computes the final product by accumulating the partial products. The number of clock cycles depends on the size of the segmentation. For our 233 bit ECC design we are considering IKM configurations with Table 5: Area and timing of combinatorial multipliers in 0.25µm CMOS128, 64, and 32 bit partial combinatorial multiplier. These designs require 3, 9, and 27 clock cycles respectively.

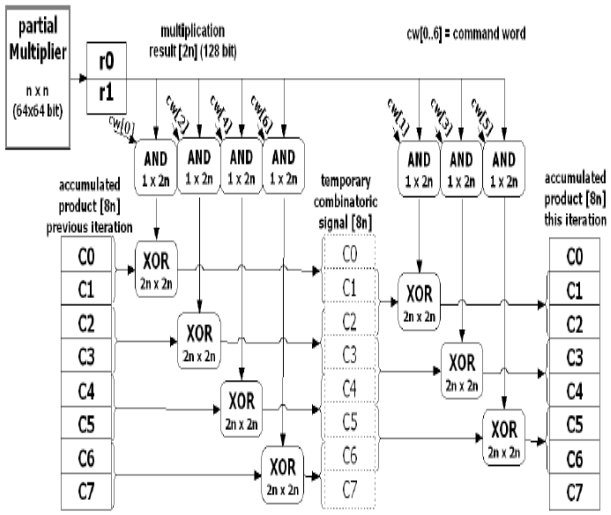

Fig.5. Configurable structure for the accumulation of partial products in the IKM process with embedded reduction operation this design allows only storing four registers (c0 to c3) in flip-flops instead of eight

For the hardware-IKM described the selection and accumulation blocks of the multiplier become large with higher segmentation. This is due to a complicated data path that indeed reduces the number of total executed XOR operations, but leads to an irregular data path structure. We solved the issue by implementing a data path that does not reduce the number of operations but has a much more regular structure and thus requires less silicon area. In the accumulation of the four segments IKM, seven different positions are possible. The positions can be represented by a seven bit command word which is generated by a small controller block. The value of this

command word depends on the current clock cycle of the multiplication. The data path is organized as shown in Fig.5. If a command bit is set, the partial product is forwarded to the corresponding XOR operation, otherwise the XOR operation is performed with zeros what results in no change at the relative position. Because of the overlapping XOR operations it is necessary to perform this process in two stages. The intermediate result after the first stage is not stored but is forwarded directly to the second stage. The result of the second stage is stored in registers, and used again in the next iteration. An additional benefit of this method is that the longest path is only one AND and two XORs. The selection process is done in the same way. Small control words determine the XOR operations that have to be executed. The results for the new selection and accumulation approach are listed in Table 2 and are compared to the original method. The results clearly show that the influence of the number of segments on area consumption is no longer that significant.

Table1. Area and timing of combinatorial multipliers in 0.25µm CMOS

	CPM			hCKM			hRAIK		
m	[mm^2]	[ns]	[nWs]	[mm^2]	[ns]	[nWs]	[mm^2]	[ns]	[nWs]
64	0.477	3.3	3.36	0.176	6.2	0.84	0.170	6.0	0.78
128	2.070	3.9	18.64	0.555	7.9	3.52	0.537	7.6	3.31
256	9.000	4.6	105.1	1.714	9.8	14.01	1.636	9.1	12.51

Table2. Area consumption in mm^2 of selection and accumulation tasks for 233 bit IKM compared to the original method.

	Selection	Accumulation	Summation sel. + acc.	Original method
2 segment	0.05	0.08	0.13	0.15
4 segment	0.05	0.09	0.14	0.39
8 segment	0.06	0.10	0.16	0.59

Table3. GF (2^m)-multipliers tailored for B-163, B-233, and B-571

Size [bit]	Segments	Size core mul	Cycles	Area [mm^2]	Power [mW]	Energy [nWs]
163	2	96	3	0.79	47.9	4.31
163	4	48	9	0.45	31.6	8.53
163	8	24	27	0.35	18.5	14.99
233	2	128	3	1.17	64.5	5.80
233	4	64	9	0.62	42.9	11.58
233	8	32	27	0.44	22.8	18.47
571	2	320	3	4.35	277.6	25.0
571	4	160	9	2.10	141.8	38.3
571	8	80	27	1.31	82.9	67.14

Tailored ECC core multiplier with intention to apply the multiplier in a particular ECC design, we made a further modification of the accumulation logic: we integrated the reduction inside the multiplier. The reduction must be performed to transform the long product to an equivalent m bit element inside the field GF (2^m). It corresponds to the modulo operation in prime fields. Traditionally, the reduction is performed after the multiplication is finished, i.e. after the nine partial multiplication steps were performed. Instead, we perform a reduction after every iteration step. Thus, the partial results c4, c5, c6, and c7, shown in Fig.5, do not need to be stored. In case of a 256 bit multiplier it saves 255 flip-flops. For the 233 bit B-233 curve with four-segment multiplier, which requires nine clock cycles for the polynomial multiplication in GF(2233) the silicon area is 0.62mm2 measured for the 0.25 µm CMOS technology.

IV. 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 2(*m*+1) AND gates in a pair of AND cells. Besides, the AC requires (*m*+1) XOR gates. Moreover, (2.5m^2+6.5*m*+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.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 and the existing designs of [11] are listed in Table 4. 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 and 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.

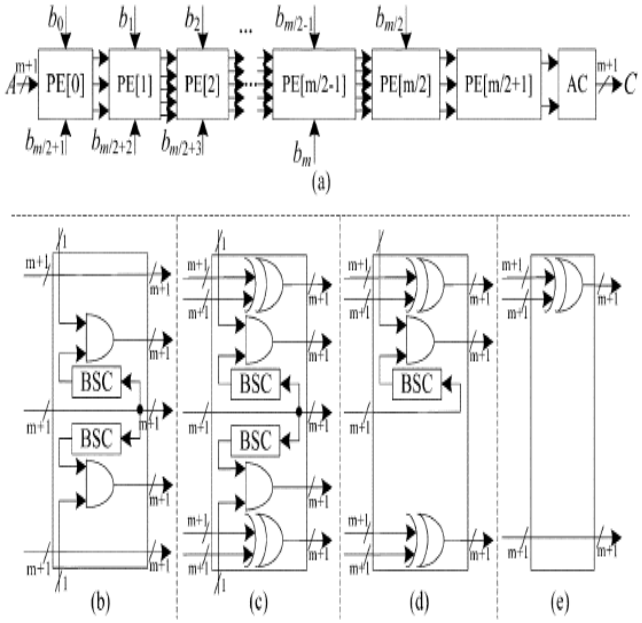

Fig.6. Low-latency register-sharing systolic structure (a) the systolic structure (b) Structure of PE [1] (c) Structure of a regular PE (from PE [2] to PE [*m*/2-1]). (d) Structure of PE [*m*/2]. (e) Structure of PE [*m*/2+1].

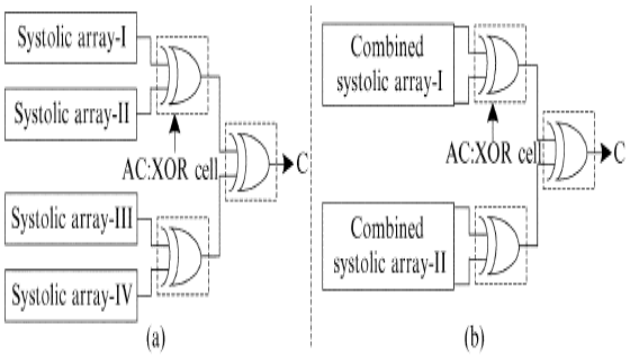

Fig.7. Improved low-latency systolic structure (a) The proposed systolic array merging (b) Improved systolic structure

Table 4. Area and Time Complexity

Design	AND	MUX	XOR (3-input)	XOR (2-input)	Register	Latency	Critical-path
[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	$(m^2-m)/2$	$2m$	$(5m^2+7m-6)/2$	$m/2+1$	T_A+T_{3X}*
[15]	$(m+1)^2$	0	0	$(m+1)^2$	$3(m+1)^2$	$m+1$	T_A+T_X
[16]‡	Lm	$2m$	0	Lm	$2m+m/L-L-1$	$2m/L-1$	T_M†$+T_A+(\log_2 L)T_X$
[12]‡	Lm	0	0	$L(2m-1)$	$2m$	m/L	$T_A+[(L-1)+\log_2 L]T_X$
Fig. 6	$(m+1)^2$	0	0	$(m+1)^2$	$(5/2)m^2+(13/2)m+4$	$m/2+3$	T_X
Fig. 7	$(m+1)^2$	0	0	$(m+1)^2$	$(5/2)m^2+(1/2)m+7$	$m/4+4$	T_X

*T_{3X}: delay of a 3-input XOR gate.

; T_F: the delay of a T flip flop.

‡: For the digit-serial structure, L is the digit size.

† T_M: the delay of a 2:1 MUX.

Table 5. Comparison of Area and Time Complexity for M =20

designs	area (μm^2)	ACT (ns)	Power (mW)	ADP	PDP	throughput
[15]	19115	0.17	4.178	3250	0.71	1
[16]*	2463.9	1.35	0.537	3326	0.72	1/5
Fig. 7	17871	0.13	3.893	2323	0.51	1

ACT = (critical−path) × (cycles number required to obtain a result).

*Digit-size $L = 4$.

Table6. FPGA Synthesis Result of Proposed and Existing Designs

designs	LE	ACT (ns)	PC (mW)	ADP	PDP
[15]	1764	4.2	103.12	7409	433
[16]‡	185	35.7	71.62	6607	2557
Fig. 7	1736	3.6	94.27	6250	339

Power consumption (PC).

Logic element (LE).

‡Digit-size $L = 4$.

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 and digit-serial systolic structure. The average computation time (ACT), area and power consumption (at 100 MHz frequency) thus obtained are listed in Table 5. 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. Besides, we have synthesized the proposed design (see Fig. 7) and the designs] for m = 20 and implemented on an Altera FPGA: Cyclone-II EP2C15AF256A7 using Quartus II 9.0. From the synthesis result, as shown in Table 6, we find that the proposed design has lower ADP and less PDP than the existing ones.

V. 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.

VI. REFERENCES

[1] Jiafeng Xie, Pramod Kumar Meher, and Jianjun He, **"**Low-Complexity Multiplier for *GF* (2^m) Based on All-One Polynomials", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 1, January 2013.

[2] 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.

[3] 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.

[4] 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.

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

[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] C. Paar, "Low complexity parallel multipliers for Galois fields *GF* ($(2^m)^4$) based on special types of primitive polynomials," in Proc. IEEE Int. Symp. Inform. Theory, 1994, p. 98.

[8] 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.

[9] 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.

[10] 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.
[11] 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.
[12] 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.