ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.12, No.11, 1–6

Efficient systolic modular multiplier/squarer for fast exponentiation over $GF(2^m)$

Se-Hyu Choi and Keon-Jik Lee[a)]

School of Architectural, Civil, Environmental and Energy Engineering, Kyungpook National University, Daegu, 702–701, Korea

a) *LeeMeeKael@gmail.com*

Abstract: Using the concept of common components, this letter shows that field multiplication and squaring over $GF(2^m)$ can be efficiently combined, with little hardware overhead. The analysis results show that about 39.23% area-time (AT) complexity is improved when we employ the combined systolic multiplier/squarer instead of implementing the multiplier and the squarer separately in the least significant bit (LSB)-first exponentiation. The proposed architecture features regularity, unidirectional data flow, and local interconnection, and thus is well suited to VLSI implementation.

Keywords: modular multiplication, finite field arithmetic, systolic array

Classification: Integrated circuits

References

[1] C. L. Wang and J. L. Lin: IEEE Trans. Circuits Syst. **38** (1991) 796. DOI:10.1109/31.135751

[2] C. S. Yeh, I. S. Reed and T. K. Troung: IEEE Trans. Comput. **33** (1984) 357. DOI:10.1109/TC.1984.1676441

[3] S. K. Jain, L. Song and K. K. Parhi: IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **6** (1998) 101. DOI:10.1109/92.661252

[4] S. H. Choi and K. J. Lee: IEICE Electron. Express **11** (2014) 20140713. DOI:10.1587/elex.11.20140713

[5] K. J. Lee and K. Y. Yoo: Inf. Process. Lett. **76** (2000) 105. DOI:10.1016/S0020-0190(00)00131-9

1 Introduction

Finite field arithmetic is used to solving many complex calculations, such as those in coding theory and cryptography. Although addition is trivial, multiplication is time-consuming. Modular exponentiation (ME) and inversion can be performed using a sequence of multiplications. As a result, efficient multipliers are important from a system performance point of view. Although several multipliers have been developed with a polynomial basis of $GF(2^m)$, their high space and time complexities are major limitations in cryptographic applications [1, 2, 3, 4]. Thus, further research on efficient multiplication architectures with low space and time complexities is required.

DOI: 10.1587/elex.12.20150222
Received March 2, 2015
Accepted April 22, 2015
Publicized May 19, 2015
Copyedited June 10, 2015

In this letter, we propose a method to increase the hardware utilization by efficiently combining modular multiplication (MM) and modular squaring (MS) into a single unit. Applying the combined systolic multiplier/squarer to LSB-first ME can achieve obvious improvement in AT complexity compared to the method implemented with two distinct multipliers. The proposed scheme can be used as a kernel circuit for both MM and ME.

2 Multiplication and squaring algorithm

2.1 Bit-parallel systolic multiplication/squaring

Let $A = \sum_{j=0}^{m-1} a_j \alpha^j$ and $B = \sum_{j=0}^{m-1} b_j \alpha^j$ be two elements in $GF(2^m)$, $T = \sum_{j=0}^{m-1} t_j \alpha^j$ be the product of A and B and $F = \sum_{j=0}^{m} f_j \alpha^j$ be the irreducible polynomial in standard basis representation, where $f_0 = f_m = 1$. The equation $\alpha^m = \sum_{j=0}^{m-1} f_j \alpha^j$ is used to reduce the product T to a polynomial of degree less than m. To derive a recurrence relation for a systolic multiplication, T can be written as follows:

$$\begin{aligned} T &= AB \bmod F = A \sum_{j=0}^{m-1} b_j \alpha^j \bmod F \\ &= A(b_0 + b_1\alpha + b_2\alpha^2 + \ldots + b_{m-1}\alpha^{m-1}) \bmod F \\ &= b_0 A + b_1(A\alpha \bmod F) + b_2(A\alpha^2 \bmod F) + \ldots + b_{m-1}(A\alpha^{m-1} \bmod F). \end{aligned} \quad (1)$$

or

$$T = (\ldots((Ab_{m-1})\alpha \bmod F + Ab_{m-2})\alpha \bmod F + \ldots + Ab_1)\alpha \bmod F + Ab_0. \quad (2)$$

The above equations lead to two implementations. One implementation is LSB-first multiplication based on (1), the other is most significant bit (MSB)-first multiplication based on (2). In this letter, we construct a low complexity systolic multiplier/squarer using the LSB first scheme. The operations in (1) consist of three steps: modular-reduction, bit-multiplication, and bit-accumulation.

Let $A_i = A_{i-1}\alpha \bmod F$. From (1), A_i at step i $(1 \le i \le m)$ is obtained from A_{i-1} recursively as

$$\begin{aligned} A_i &= \left(\sum_{j=1}^{m} a_{i-1,j-1}\alpha^{j-1}\right)\alpha \bmod F = \sum_{j=1}^{m} a_{i-1,j-1}\alpha^j \bmod F \\ &= a_{i-1,m-1}\sum_{j=1}^{m} f_{j-1}\alpha^{j-1} + \sum_{j=1}^{m-1} a_{i-1,j-1}\alpha^j, \end{aligned} \quad (3)$$

where $A_0 = A$. Therefore, each basic cell at step i computes

$$a_{i,j} = a_{i-1,j-1} + a_{i-1,m-1} f_{\langle j \rangle}, \quad m \ge j \ge 1, \quad (4)$$

where $a_{i,0} = 0$ for $0 \le i \le m-1$, $a_{0,j} = a_j$ for $m-1 \ge j \ge 0$, and $\langle j \rangle$ denotes that j is to be reduced modulo m.

From (1), let $T_i = T_{i-1} + b_{i-1}A_{i-1}$. Then, we have T_i at step i $(1 \le i \le m)$ as

$$\begin{aligned} T_i &= \sum_{j=1}^{m} t_{i,j-1}\alpha^{j-1} = \sum_{j=1}^{i} b_{j-1}A\alpha^{j-1} = \sum_{j=1}^{m} t_{i-1,j-1}\alpha^{j-1} + b_{i-1}A\alpha^{i-1} \\ &= \sum_{j=1}^{m} t_{i-1,j-1}\alpha^{j-1} + b_{i-1}\sum_{j=1}^{m} a_{i-1,j-1}\alpha^{j-1}. \end{aligned}$$

where $T_0 = 0$. Thus, the product T at step i can be represented in a recursive manner as

DOI: 10.1587/elex.12.20150222
Received March 2, 2015
Accepted April 22, 2015
Publicized May 19, 2015
Copyedited June 10, 2015

$$t_{i,j-1} = t_{i-1,j-1} + b_{i-1}a_{i-1,j-1}, \quad m \geq j \geq 1, \tag{5}$$

where $t_{0,j} = a_{i,0} = 0$ for $0 \leq i,\ j \leq m-1$, and $a_{0,j} = a_j$ for $m-1 \geq j \geq 0$.

ME is a crucial part of modern cryptographic algorithms. It is typically performed using a series of MS and MM operations based on the binary method, and thus, can be time-consuming. Similar to the above mentioned MM method, the binary ME method has two commonly used schemes: LSB-first ME and MSB-first ME, where LSB and MSB refer to the LSB and MSB of the exponent E. Note that LSB-first ME can process MS and MM simultaneously. In this letter, we consider a systolic multiplier/squarer that can compute MS and MM concurrently based on the properties of LSB-first ME, as shown in [5].

The common components to perform concurrent implementation of MS and MM are extracted during ME. By computing these components only once, the area-time complexity can be reduced. The field multiplication, $T = AB \bmod F$, and field squaring, $S = AA \bmod F$, can be expressed as

$$\begin{aligned} T &= AB \bmod F = A(b_0 + b_1\alpha + b_2\alpha^2 + \ldots + b_{m-1}\alpha^{m-1}) \bmod F \\ &= b_0A + b_1(A\alpha \bmod F) + b_2(A\alpha^2 \bmod F) + \ldots + b_{m-1}(A\alpha^{m-1} \bmod F). \end{aligned} \tag{6}$$

$$\begin{aligned} S &= AA \bmod F = A(a_0 + a_1\alpha + a_2\alpha^2 + \ldots + a_{m-1}\alpha^{m-1}) \bmod F \\ &= a_0A + a_1(A\alpha \bmod F) + a_2(A\alpha^2 \bmod F) + \ldots + a_{m-1}(A\alpha^{m-1} \bmod F). \end{aligned} \tag{7}$$

Depending on the bits of B or A, the repeated operations involve multiplying A by α and then taking mod F. Note that the common components, $A\alpha^i \bmod F$ $(0 \leq i \leq m-1)$ exist in (6) and (7), and thus by computing the common components once can lead to improved performance when performing the LSB-first ME. Similar to (4) and (5), the recursive equation of (7) at step i $(1 \leq i \leq m)$ can be represented as

$$a_{i,j} = a_{i-1,j-1} + a_{i-1,m-1}f_{\langle j \rangle}. \tag{8}$$

$$s_{i,j-1} = s_{i-1,j-1} + a_{i-1}a_{i-1,j-1}, \quad m \geq j \geq 1, \tag{9}$$

where $s_{0,j} = a_{i,0} = 0$ for $0 \leq i,\ j \leq m-1$, and $a_{0,j} = a_j$ for $m-1 \geq j \geq 0$. Then, we merge (4), (5), (8), and (9) to yield the recursive equation of a combined multiplier/squarer as

$$\begin{aligned} &a_{i,j} = a_{i-1,j-1} + a_{i-1,m-1}f_{\langle j \rangle}. \\ &t_{i,j-1} = t_{i-1,j-1} + b_{i-1}a_{i-1,j-1}. \\ &s_{i,j-1} = s_{i-1,j-1} + a_{i-1}a_{i-1,j-1}, \quad 1 \leq i \leq m \text{ and } m \geq j \geq 1, \end{aligned} \tag{10}$$

where $t_{0,j} = s_{0,j} = a_{i,0} = 0$ for $0 \leq i,\ j \leq m-1$, and $a_{0,j} = a_j$ for $m-1 \geq j \geq 0$.

A parallel systolic multiplier/squarer with $m = 4$ based on (10) is shown in Fig. 1(a), where m^2 basic cells of Fig. 1(b) are used. One 1-bit latch (denoted by "•") is placed at each horizontal link and diagonal link, and two 1-bit latches are placed at each vertical link. The basic cell at position (i, j) performs the logic operations described in (10). The initial positions and index points of each input are as follows. First, a_i and b_i $(0 \leq i \leq m-1)$ enter index $[i+1, m]$ from the left side and flow in the direction of $[0, -1]$. Next, the values t_j and s_j $(0 \leq j \leq m-1)$ enter index $[1, j+1]$ from the top and their computation results flow in the direction of $[1, 0]$, where $t_j = s_j = 0$. Then, f_j $(0 \leq j \leq m-1)$ enters index $[1, \langle j+m \rangle]$ from

DOI: 10.1587/elex.12.20150222
Received March 2, 2015
Accepted April 22, 2015
Publicized May 19, 2015
Copyedited June 10, 2015

the top and flows in the direction of $[1, 0]$. Next, a_j $(0 \leq j \leq m-1)$ enters index $[1, j+1]$ from the top, and is computed using the partial values generated by the previous row to give new partial values that are passed on to the next row, and then flows in the direction of $[1, 1]$. The final results, T_m and S_m, emerge in parallel from the bottom row of the array after m iterations. Each basic cell consists of three 2-input AND gates and three 2-input XOR gates, as shown in Fig. 1(b). The (i, j) cell receives $a_{i-1,j-1}$ as its input from the $(i-1, j-1)$-th cell; $t_{i-1,j-1}$, $s_{i-1,j-1}$, and f_j, from the $(i-1, j)$-th cell; and a_{i-1} and b_{i-1} from the $(i, j+1)$-th cell. The critical path delay of this structure is the total delay of one 2-input AND gate and one 2-input XOR gate. In Fig. 1(a), the top inputs t_j, s_j, f_{j+1}, and a_j $(0 \leq j \leq m-2)$ are staggered by one clock cycle relative to t_{j+1}, s_{j+1}, $f_{\langle j+2 \rangle}$, and a_{j+1}, respectively, and the left side inputs a_{i+1} and b_{i+1} $(0 \leq i \leq m-2)$ are staggered by two clock cycles relative to a_i and b_i, respectively.

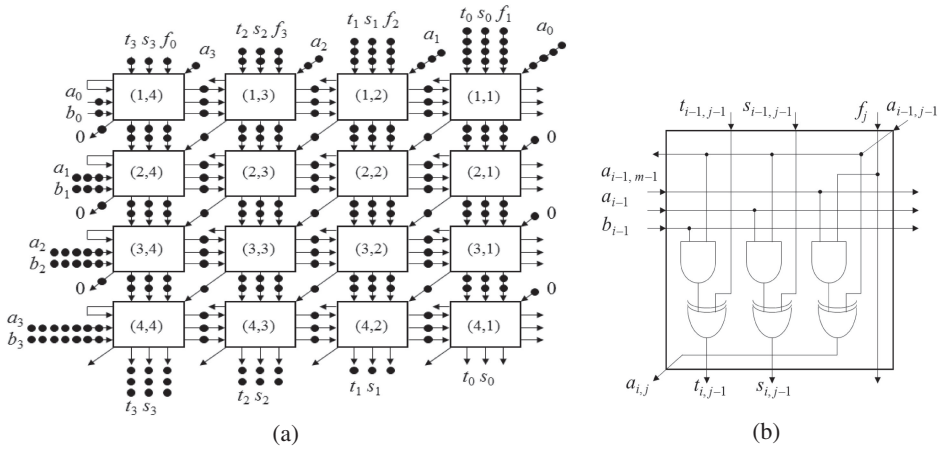


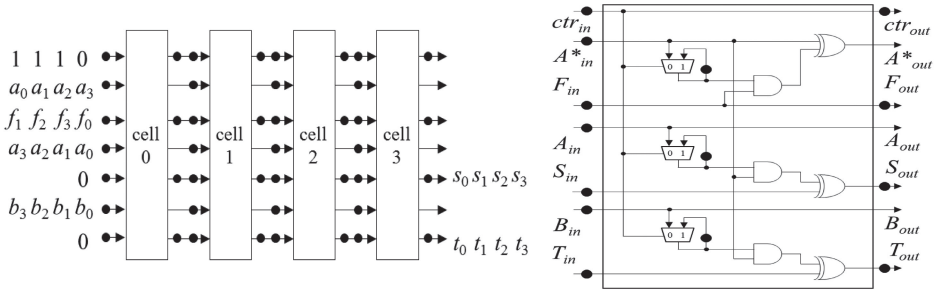
Fig. 1. (a) Proposed multiplier (b) Circuit of cell (i, j)

2.2 Bit-serial systolic multiplication/squaring

By projecting Fig. 1(a) in an eastward direction and retiming using the cut-set systolization techniques [1, 5], the new one-dimensional serial systolic array in Fig. 2(a) can be derived. This array consists of m basic cells; the functions of these cells are shown in Fig. 2(b). This circuit is controlled by a control sequence $ctr = 011\ldots1$ of length m. Because the MSB of each partial result A_i (denoted by A^*) is required to execute (10), one 2-to-1 MUX and one 1-bit latch are added to each cell in Fig. 2(a). When ctr is in logic 0, the MSB of each A_i is loaded into each cell.

Note that according to the projection, input values other than A and B enter the left side of the array in a serial form, while each bit of A and B must remain inside the systolic array, i.e., a_i and b_i $(0 \leq i \leq m-1)$ must remain in the i-th cell to be ready for execution. It is possible to incorporate an additional two 2-to-1 MUXes and two 1-bit latches into each cell, as shown in Fig. 2(a), so that a_i and b_i may also enter the array serially with LSB-first at the same time as ctr. That is, when $ctr = 0$ enters the i-th cell, a_i and b_i also enter that cell, and the loading operation of a_i and b_i occurs. The basic cell in Fig. 2(b) consists of three 2-input AND gates, three 2-input XOR gates, three 2-to-1 MUXes, and fourteen 1-bit latches; its critical path

DOI: 10.1587/elex.12.20150222
Received March 2, 2015
Accepted April 22, 2015
Publicized May 19, 2015
Copyedited June 10, 2015



(a) (b)

Fig. 2. (a) Proposed multiplier (b) Circuit of the basic cell

delay is the total delay of one 2-input AND gate, one 2-input XOR gate and one 2-to-1 MUX. The results t_j and s_j $(0 \le j \le m-1)$ emerge from the right side of the array in serial form with MSB first.

3 Analysis and conclusion

We obtained the area of the gates, multiplexer, and latch along with their worst-case intrinsic delays pertaining to unit drive-strength from the "SAMSUNG STD 150 0.13 µm 1.2 V CMOS Standard Cell Library" databook.

Using these data we estimated the time and area complexities of the proposed and related works. Table I summarizes the time and area requirements of the cells used in our analysis, where G_n denotes the n-input logic cell G.

Table I. Time and area requirements for the cells used

	AND_2	OR_2	XOR_2	MUX_2	Latch
Time (ns)	0.094	0.105	0.167	0.141	0.157
Area (transistor count)	6.68	6.68	12.00	12.00	16.00

To demonstrate the efficiency of the proposed method, we measure AT complexity of each work and then calculate the improvement. ME can be performed by a series of simultaneous MM and MS in order to implement a fast exponentiator: by the combined multiplier/squarer or by two multipliers in parallel. In Table II, the proposed architecture is compared with Fig. 4 in [1] and the modified Fig. 2 in [2]. Note that one 2-to-1 MUX and two 1-bit latches are additionally required for each cell of Fig. 2 in [2] if the coefficients of B are inputted serially (i.e., LSB-first) into the array at the same time as ctr. Our method performs MM and MS at a time using one multiplier for the efficient implementation of the LSB-first ME, and thus requires m cells. On the other hand, [1] and [2] compute MM and MS separately using two multipliers, and thus need $2m$ cells, respectively. The basic cell of [1] (resp., [2]) consists of three (resp., three) 2-input AND gates, two (resp., two) 2-input XOR gates, two (resp., two) 2-to-1 MUXes, and ten (resp., twelve) 1-bit Latches. The total cell area is listed in Table II. As can be seen from Table II, the proposed bit-serial systolic multiplier/squarer has better advantages in terms of the area, time, AT, and latency over other methods.

DOI: 10.1587/elex.12.20150222
Received March 2, 2015
Accepted April 22, 2015
Publicized May 19, 2015
Copyedited June 10, 2015

Table II. Comparison of the serial systolic arrays for parallel computation of MM and MS

Multipliers	Two separate multipliers based on Wang-Lin [1]	Two separate multipliers based on Yeh et al. [2]	Our combined multiplier/squarer
Algorithm	MSB	LSB	LSB
# cells	$2m$	$2m$	m
Throughput	$1/m$	$1/m$	$1/m$
# control signal	1	2	1
Latency	$3m$	$3m$	$3m-1$
Cell area			
AND_2	$6m$	$6m$	$3m$
XOR_2	$4m$	$4m$	$3m$
MUX	$4m$	$4m$	$3m$
Latch	$20m$	$24m$	$14m$
# transistors	$456.08m$	$520.08m$	$316.04m$
Cell time			
Delay	0.569	0.402	0.402
Total delay	$1.707m$	$1.206m$	$1.206m-0.402$
AT complexity	$778.53m^2$	$627.22m^2$	$381.14m^2-127.05$
AT Improvement	51.04%	39.23%	-

Specifically, the comparison results show that the AT complexity of our method is improved by approximately 51.04% and 39.23% compared to Wang-Lin and Yeh et al.'s methods, respectively. The proposed parallel (resp., serial) multiplier/squarer produces the results at a rate of one per 1 (resp., m) cycles with a latency of $3m-1$ (resp., $3m-1$) cycles. Note that the parallel systolic architecture has better throughput but a much higher hardware cost than a serial systolic architecture. Thus, the serial architecture is attractive in resource-constrained cryptographic applications.

This study proposes a new systolic combined multiplier/squarer for efficient modular exponentiation, which is the crucial operation in the finite field arithmetic. The proposed method utilizes the common components in the simultaneous computation of both MM and MS. So, it can be used in the efficient computation of the LSB-first ME over $GF(2^m)$. The effectiveness of our study is demonstrated by the reduced AT complexity compared with the related work. The proposed architecture involves unidirectional data flow, and is highly regular and nearest-neighbor connected. It is thus well-suited for VLSI implementation. The proposed multiplier/squarer can be used for cryptographic applications such as asymmetric watermarking for geographic information system (GIS) vector map management.

DOI: 10.1587/elex.12.20150222
Received March 2, 2015
Accepted April 22, 2015
Publicized May 19, 2015
Copyedited June 10, 2015