ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.11, No.20, 1–6

Low complexity semi–systolic multiplication architecture 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: This paper presents a semi-systolic Montgomery multiplier based on the redundant basis representation of the finite field elements. The proposed multiplier has less hardware and time complexities compared to related multipliers. We also propose a serial systolic Montgomery multiplier that can be applied well in space-limited hardware. Furthermore, a simple inversion based on the proposed scheme is presented.

Keywords: modular multiplication, finite field arithmetic, systolic array

Classification: Integrated circuits

References

[1] H. Wu, M. A. Hasan, I. F. Blake and S. Gao: IEEE Trans. Comput. **51** (2002) 1306. DOI:10.1109/TC.2002.1047755

[2] W. H. Lee, K. J. Lee and K. Y. Yoo: ICCSA LNCS 3045 (2004) 638.

[3] C. Y. Lee, C. W. Chiou and J. M. Lin: J. Electron. Test. **22** (2006) 143. DOI:10.1007/s10836-006-7446-9

[4] C. W. Chiou, C. Y. Lee, A. W. Deng and J. M. Lin: IEICE Trans. Fundamentals **E89-A** (2006) 566. DOI:10.1093/ietfec/e89-a.2.566

[5] W. T. Huang, C. H. Chang, C. W. Chiou and F. H. Chou: IET Info. Secur. **4** (2010) 111. DOI:10.1049/iet-ifs.2009.0160

[6] K. J. Lee and K. Y. Yoo: Integr. VLSI J. **32** (2002) 99. DOI:10.1016/S0167-9260(02)00044-5

1 Introduction

In finite field arithmetic, addition is trivial but multiplication is time-consuming. Other operations such as exponentiation and inversion can be performed using repeated multiplication. As a result, efficient multiplier architectures are important from a system performance point of view. Another crucial factor affecting field arithmetic efficiency is the choice of the basis. Wu et al. [1] proposed a redundant basis (RB) to embed a finite field into a minimal cyclotomic ring with the elegant multiplicative structure of a cyclic group. A number of systolic multipliers over $GF(2^m)$ have been introduced [2, 3, 4, 5]. Recently, Huang et al. [5] proposed a semi-systolic multiplier to reduce both time and space complexities. Chiou et al. [4]

DOI: 10.1587/elex.11.20140713
Received July 23, 2014
Accepted September 1, 2014
Publicized October 3, 2014
Copyedited October 25, 2014

proposed a semi-systolic Montgomery multiplier (MM) with concurrent error detection capability. However, most existing semi-systolic multipliers suffer from several shortcomings, including large time and/or hardware overhead.

In this letter, we propose a low-complexity multiplication algorithm based on the RB and two systolic multipliers over $GF(2^m)$. The proposed scheme can be used as a kernel circuit for multiplication and exponentiation (inversion).

2 MM for finite field

2.1 Bit-parallel (semi) systolic MM

Let β be a primitive nth root of unity in some extension of $GF(2)$. The nth cyclotomic field $GF(2^n)$ over $GF(2)$ is defined to be the splitting field of $x^n - 1$ over $GF(2)$. Then, $GF(2^n)$ is generated by β over $GF(2)$ and any element A of $GF(2^n)$ can be represented as $A = a_0 + a_1\beta + a_2\beta^2 + \ldots + a_{n-1}\beta^{n-1}$, where $a_i \in GF(2)$. Let $GF(2^m)$ be a field that can be embedded in $GF(2^n)$. It has been shown that $GF(2^m)$ is contained in $GF(2^n)$ iff n is odd and m divides the multiplicative order of $2 \bmod n$ [1]. Note that the representation of A is not unique, since $1 + \beta + \beta^2 + \ldots + \beta^{n-1} = 0$. By slightly abusing the terminology, $\{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$ is denoted as a RB for any subfield of $GF(2^n)$ containing $GF(2^m)$ and it forms a cyclic group of order n (i.e., $\beta^n = 1$). Consider the RB for $GF(2^m)$ over $GF(2)$. Let field elements $A, B \in GF(2^m)$ be represented with respect to the RB $\{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$ as $A = \sum_{i=0}^{n-1} a_i\beta^i$ and $B = \sum_{i=0}^{n-1} b_i\beta^i$, where $a_i, b_i \in GF(2)$. Then, the product $T = AB$ is obtained as $T = \sum_{j=0}^{n-1} t_j\beta^j$, where $t_j = \sum_{i=0}^{n-1} a_i b_{\langle j-i \rangle}$. Note that $\langle j - i \rangle$ denotes that $j - i$ is to be reduced modulo n.

MM was proposed originally for efficient integer modular multiplication. Later, it was shown that MM is also applicable to $GF(2^m)$. Instead of computing $AB \bmod G$ in $GF(2^m)$, it computes $ABR^{-1} \bmod G$ in $GF(2^m)$, where G is an irreducible polynomial of degree m, $R = \beta^m$ is a special fixed element of $GF(2^m)$ and $\gcd(R, G) = 1$. Consider MM for $GF(2^m)$ using the RB. Let A and B be two elements in $GF(2^m)$, T be the result of the product AB and these elements are represented in the RB as follows: $A = \sum_{i=0}^{n-1} a_i\beta^i$, $B = \sum_{i=0}^{n-1} b_i\beta^i$, and $T = \sum_{i=0}^{n-1} t_i\beta^i$, where a_i, b_i and $t_i \in GF(2)$. Modular reduction and squaring are more efficient over the RB than in other bases. For multiplying $A = a_0 + a_1\beta + a_2\beta^2 + \ldots + a_{n-1}\beta^{n-1}$ by β, $A\beta = a_0\beta + a_1\beta^2 + a_2\beta^3 + \ldots + a_{n-1}\beta^n$. Since $\beta^n = 1$, $A\beta = a_{n-1} + a_0\beta + a_1\beta^2 + a_2\beta^3 + \ldots + a_{n-2}\beta^{n-1}$. Thus, the multiplication A by β can be obtained using one right cyclic shift of A as $A\beta = \sum_{i=0}^{n-1} a_i\beta^{\langle i+1 \rangle}$.

The multiplicative inverse of β is can be performed as $\beta^{-1} = \beta^{n-1}$. Multiplying A by β^{-1}, we obtain that $A\beta^{-1} = \beta^{-1}(a_0 + a_1\beta + a_2\beta^2 + \ldots + a_{n-1}\beta^{n-1}) = a_1 + a_2\beta + a_3\beta^2 + \ldots + a_0\beta^{n-1}$. Thus, $A\beta^{-1}$ is obtained by one left cyclic shift of A as $A\beta^{-1} = \sum_{i=0}^{n-1} a_i\beta^{\langle i-1 \rangle}$. On the other hand, the squaring of an element A can be optimized owing to the fact that cross terms disappear because they come in pairs and the underlying field is $GF(2)$. Since n is odd and $\beta^n = 1$, $A^2 = a_0 + a_1\beta^2 + a_2\beta^4 + \ldots + a_{n-2}\beta^{2(n-2)} + a_{n-1}\beta^{2(n-1)} = a_0 + a_1\beta^2 + \ldots + a_{\langle (n-1)(n+1)/2 \rangle}\beta^{n-1} + a_{\langle (n+1)/2 \rangle}\beta + a_{\langle 3(n+1)/2 \rangle}\beta^3 + \ldots + a_{\langle (n-2)(n+1)/2 \rangle}\beta^{n-2}$. Thus, the squaring of A is obtained simply by using the subscript operation of the coefficient a_i as $A^2 = \sum_{i=0}^{n-1} a_{\langle i(n+1)/2 \rangle}\beta^i$. Furthermore, A^{2^k} can be obtained easily as $A^{2^k} =$

DOI: 10.1587/elex.11.20140713
Received July 23, 2014
Accepted September 1, 2014
Publicized October 3, 2014
Copyedited October 25, 2014

$\sum_{i=0}^{n-1} a_{\langle i((n+1)/2)^k \rangle}\beta^i$. Note that these properties are useful for constructing efficient low-complexity field arithmetic architectures for $GF(2^m)$ defined by the RB.

A new algorithm to compute the Montgomery product $T = ABR^{-1}$ over the RB can be obtained by $T = AB\beta^{-n} = A(b_{n-1}\beta^{n-1} + b_{n-2}\beta^{n-2} + \ldots + b_1\beta + b_0)\beta^{-n} = b_{n-1}A\beta^{-1} + b_{n-2}A\beta^{-2} + \ldots + b_1A\beta^{-n+1} + b_0A\beta^{-n}$. From this, an iterative procedure for computing ABR^{-1} can be formulated as follows.

$$\begin{aligned} T_{-1} &= 0 \\ T_i &= T_{i-1} + b_{n-i-1}A\beta^{-i-1} \\ T &= T_{n-1}, \end{aligned} \tag{1}$$

for $i = 0, 1, \ldots, n-1$. After n iterations, T is obtained, where $T_i = t_{i,n-1}\beta^{n-1} + t_{i,n-2}\beta^{n-2} + \ldots + t_{i,1}\beta + t_{i,0}$. Assuming that $A_i = \sum_{j=0}^{n-1} a_{i,j}\beta^j$ is the ith intermediate value, Equation (1) can be reformulated as the following recursive equations.

$$A_i = A_{i-1}\beta^{-1} \tag{2}$$

$$T_i = T_{i-1} + b_{n-i-1}A_{i-1}, \tag{3}$$

where $T_i = \sum_{k=0}^{i} b_{n-i-1}A\beta^{-i-1}$, $A_{-1} = A\beta^{-1}$, $T_{-1} = 0$ and $0 \le i \le n-1$. Note that from (2) and (3) it is evident that $\deg(T_{n-1}) \le n-1$. To reduce the critical path delay, the operation of $T_i = T_{i-1} + b_{n-i-1}A_{i-1}$ in (3) can be reorganized as follows: $C_i = b_{n-i-1}A_{i-1}$ and $T_i = T_{i-1} + C_{i-1}$ and thus the final result $T = T_{n-1} + C_{n-1}$, where $C_{-1} = 0$ and $C_i = \sum_{j=0}^{n-1} c_{i,j}\beta^j$.

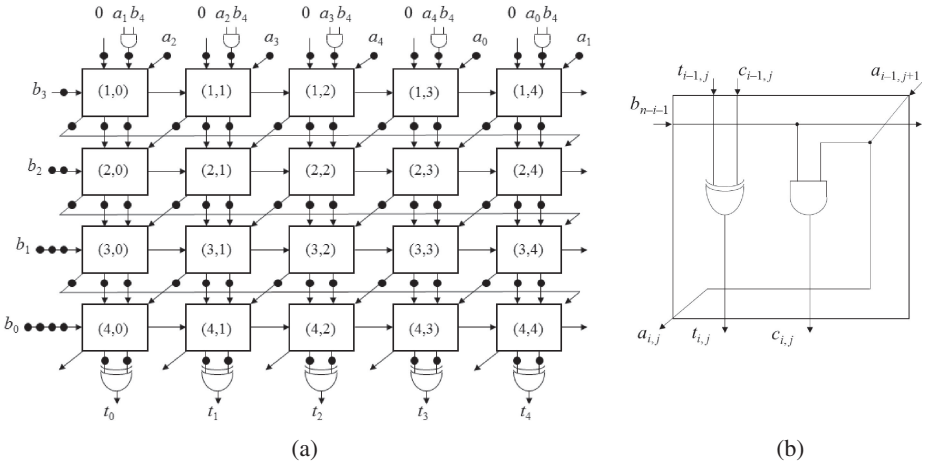

Fig. 1. (a) Proposed multiplier in $GF(2^4)$ (b) Circuit of the (i, j) cell

For simplicity, the binary field $GF(2^4)$ is used to illustrate the systolic multiplier architecture over the RB, where $GF(2^4)$ can be embedded in the minimal cyclotomic field $GF(2^5)$. Based on the proposed algorithm, the hardware architecture of the semi-systolic multiplier is shown in Fig. 1(a), where $(n-1) \times n$ basic cells, n AND gates and n XOR gates are used and "•" denotes a 1-bit latch. In Fig. 1(b), the basic cell at position (i, j) performs the following logic operations ($1 \le i \le n-1$ and $0 \le j \le n-1$): $a_{i,j} = a_{i-1,j+1}$; $c_{i,j} = b_{n-i-1} \cdot a_{i-1,j+1}$; $t_{i,j} = t_{i-1,j} \oplus c_{i-1,j}$. In Fig. 1(a), the cell at position (i, j) receives $a_{i-1,j+1}$ from the cell at position $(i-1, j+1)$ of the previous row and computes $c_{i,j}$ and $t_{i,j}$, respectively.

DOI: 10.1587/elex.11.20140713
Received July 23, 2014
Accepted September 1, 2014
Publicized October 3, 2014
Copyedited October 25, 2014

The initial positions and index points of each inputs are as follows: b_i $(0 \le i \le n-2)$ enters index $[n-i-1, 0]^T$ from the left side and flows in the direction $[0, 1]^T$, where T denotes the transpose operator. $a_{\langle j-3\rangle}$ $(0 \le j \le n-1)$ enters index $[1, j]^T$ from the top and flows in the direction $[1, -1]^T$, where $a_{0,j+1} = a_{\langle j-3\rangle}$. a_j $(2 \le j \le n-1)$ also enters index $[j, n-1]^T$ from the right side and flows in the direction $[1, -1]^T$, where $a_{j-1,n} = a_j$. The values t_j and c_j $(0 \le j \le n-1)$ enter index $[1, j]^T$ from the top, respectively and are computed with the partial products generated by the previous row to give new partial products that are passed on to the next row, and then flow in the direction $[1, 0]^T$, where $t_{0,j} = t_j = 0$ and $c_{0,j} = c_j = a_{\langle j+1\rangle} \cdot b_{n-1}$. The result T is obtained from the bottom row of the array after $n-1$ iterations.

It can be seen from Fig. 1(a) that $a_{i,0}$ generated on the left side of the ith row enters the right side cell of the $(i+1)$th row (i.e., one left cyclic shift). In Fig. 1(b), the basic cell consists of one 2-input AND gate and one 2-input XOR gate, and the cell at position (i, j) receives $a_{i-1,j+1}$ as its input from the $(i-1, j+1)$th cell, $t_{i-1,j}$ and $c_{i-1,j}$ from $(i-1, j)$th cell, and b_i from $(i, j-1)$th cell, respectively. In Fig. 1(a), the left side input b_i (resp., the right side input a_{j+1}) is staggered by one clock cycle relative to b_{i+1} (resp., a_j), where $n-3 \ge i \ge 0$ and $1 \le j \le n-2$.

2.2 Bit-serial systolic MM

By projecting Fig. 1(a) in the east direction (projection vector $[0, 1]^T$ and schedule vector $[2, 1]^T$) and retiming by the cut-set systolization techniques [6], a new one-dimensional serial systolic multiplier can be derived. The result is shown in Fig. 2(a), where "•" denotes a 1-bit latch. This multiplier consists of $n-1$ identical basic cells, one 2-input AND gate and one 2-input XOR gate, where the functions of the basic cell are depicted in Fig. 2(b).

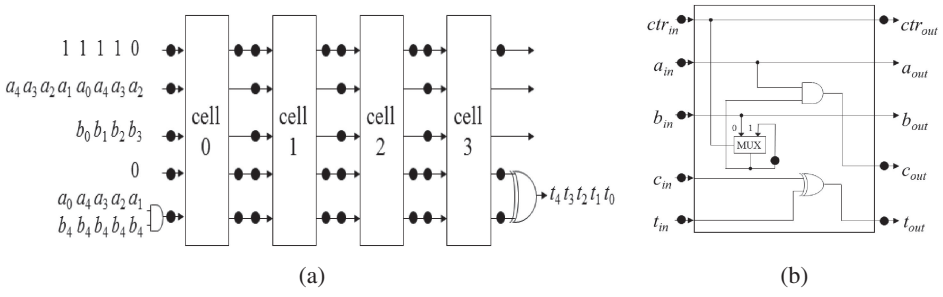

Fig. 2. (a) Proposed multiplier in $GF(2^4)$ (b) Circuit of the basic cell

Note that according to the projection, the input values other than B enter the left side of the array in a serial form, while the coefficients of B should stay inside the array, i.e., b_{n-i-2} $(0 \le i \le n-2)$ should remain at ith cell to be ready for the execution. It is possible to incorporate an additional one 2-to-1 MUX and one 1-bit latch into each cell in Fig. 2(a), so that b_i may also enter the array serially with the most significant bit first at the same time as the control sequence ctr. The multiplier of Fig. 2 is controlled by a control sequence $ctr = 011\ldots1$ of length n. When $ctr = 0$ enters the ith cell, b_{n-i-1} also enters that cell, and then its loading operation occurs, for $1 \le i \le n-1$. The basic cell of Fig. 2(b) consists of one 2-input AND

DOI: 10.1587/elex.11.20140713
Received July 23, 2014
Accepted September 1, 2014
Publicized October 3, 2014
Copyedited October 25, 2014

gate, one 2-input XOR gate, one 2-to-1 MUX, and nine 1-bit latches, and its critical path delay is one 2-input AND gate and one 2-to-1 MUX delays. If the input data come in continuously, this multiplier produces output results at a rate of one per $m+1$ clock cycles with a latency of $3m+1$ clock cycles. The result t_j $(0 \leq j \leq n-1)$ emerges from the right side of the array in serial form with the least significant bit first.

In addition, an application of the proposed scheme is to compute the inverse of any element in $GF(2^m)$. Inversion is a special case of exponentiation and it can be obtained as $B^{-1} = B^{2^m-2}$, where the exponent $E = 2^m - 2$ can be represented with a vector representation $[e_{m-1}e_{m-2}\ldots e_1e_0]$. The inversion $T = B^{2^m-2}$ over the RB can be computed as $T_{i+1} = (T_i \cdot T_i) \cdot B$ for $i = 1$ to $m-2$ with initially $T_1 = B$. Finally, the result $T_m = (T_{m-1} \cdot T_{m-1})$. This method shows that the inversion contains $m-2$ multiplications and $m-1$ squarings. Note that the squaring can be easily obtained by the subscript operation. Hence, the inverse element can be calculated using $m-2$ stages of the proposed multiplier.

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 structure and the related structures. The notations T_{GATEn} and A_{GATEn} denote the delay and area of the n-input cell, respectively. Table I summarizes the time and area requirements for the cells used in our analysis.

Table I. Cells used for evaluation of time and area

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

Note: MUX denotes a 2-to-1 multiplexer.

To demonstrate the efficiency of the proposed method, we measure the area-time (AT) complexity of each work and then calculate the improvement. From Table II, we can see that the semi-systolic multiplier of Fig. 1 obtains obvious area, time, and AT advantages over other multipliers.

In detail, the comparison results show that the AT complexity of the proposed semi-systolic multiplier is improved by approximately 62%, 56%, and 38% compared to Lee et al., Chiou et al., and Huang et al.'s multipliers, respectively. The proposed parallel (resp., serial) multiplier produces the results at a rate of one per 1 (resp., $m+1$) cycles with a latency of $m+1$ (resp., $3m+1$) cycles using $O(m^2)$ (resp., $O(m)$) area complexity. Note that the parallel semi-systolic architectures have better throughput but much higher hardware cost than the serial systolic architecture of Fig. 2.

This work presents an efficient multiplication algorithm for computing the modular multiplication, which is the crucial operation in the finite field arithmetic.

DOI: 10.1587/elex.11.20140713
Received July 23, 2014
Accepted September 1, 2014
Publicized October 3, 2014
Copyedited October 25, 2014

Table II. Complexity comparison of semi-systolic multipliers

Multipliers	Lee et al. [3]	Chiou et al. [4]	Huang et al. [5]	Proposed	
				Fig. 1	Fig. 2
# cells	m^2	$(m+1)m$	m^2	$m(m+1)$	m
Throughput	1	1	1	1	$1/(m+1)$
Latency	m	$m+1$	m	$m+1$	$3m+1$
Area complexity					
AND_2	$2m^2$	$2m^2+2m$	$2m^2$	m^2+2m+1	$m+1$
XOR_2	$2m^2$	0	$2m^2$	m^2+2m+1	$m+1$
XOR_3	0	m^2+m	0	0	0
MUX	0	0	0	0	m
Latch	$3.5m^2-0.5m$	$3.5m^2+3.5m$	$3.5m^2-0.5m$	$3.5m^2+3.5m+2$	$9m$
Total transistors	$93.36m^2-8m$	$93.36m^2+93.36m$	$93.36m^2-8m$	$74.68m^2+93.36m+3$	$174.68m+2$
Time complexity					
Cell delay	0.68	0.59	0.42	0.32	0.67
Total delay	$0.68m$	$0.59m+0.59$	$0.42m$	$0.32m+0.32$	$2.01m+0.67$
AT complexity	$63.48m^3-5.44m^2$	$54.62m^3+109.24m^2+54.62m$	$39.02m^3-3.34m^2$	$24.20m^3+54.44m^2+31.22m+0.97$	$351.11m^2+121.06m+1.34$
Improvement of Fig. 1					
Area	20%	20%	20%	-	-
Time	53%	46%	24%	-	-
AT	62%	56%	38%	-	-

The proposed scheme exploits the characteristics of the MM and the RB to construct a low-complexity systolic multiplication architecture. In particular, the serial systolic multiplier is attractive for space-constrained applications. The proposed architectures have the features of regularity, modularity, concurrency, and unidirectional data flow and thus are well suited for VLSI implementation.

DOI: 10.1587/elex.11.20140713
Received July 23, 2014
Accepted September 1, 2014
Publicized October 3, 2014
Copyedited October 25, 2014