

An efficient parallel systolic array for AB^2 over $GF(2^m)$

Kee-Won Kim^{1a)} and Won-Jin Lee^{2b)}

¹ College of Engineering, Dankook University, Cheonan, Korea

² Institute of Media Content, Dankook University, Yongin, Korea

a) nirkim@gmail.com

b) god7300@dankook.ac.kr (corresponding author)

Abstract: This paper presents an efficient parallel-in parallel-out systolic array for AB^2 over $GF(2^m)$ using the polynomial basis. As compared to existing related systolic arrays, the proposed array gains a significant reduction in hardware complexity. The proposed architecture includes the features of regularity, modularity and local interconnection. Accordingly, it is well suited for VLSI implementation and can be easily applied as a basic architecture for computing an inversion/division operation.

Keywords: cryptography, finite fields, multiplication, systolic array, exponentiation

Classification: Integrated circuits

References

- [1] R. E. Blahut: *Theory and Practice of Error Control Codes* (Addison-Wesley, Reading, 1983).
- [2] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone: *Handbook of Applied Cryptography* (CRC Press, Boca Raton, 1996).
- [3] S. W. Wei: IEEE Trans. Comput. **43** (1994) 226.
- [4] C. L. Wang and J. H. Guo: IEEE Trans. Comput. **49** (2000) 1120.
- [5] C. Y. Lee, E. H. Lu and L. F. Sun: IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **48** (2001) 519.
- [6] K. W. Kim and W. J. Lee: IETE Technical Review **30** (2013) 134.
- [7] S. Y. Kung: *VLSI Array Processors* (Prentice Hall, 1988).
- [8] N. Weste and K. Eshraghian: *Principles of CMOS VLSI Design: a System Perspective* (Addison-Wesley, Reading, 1985).
- [9] STMicroelectronics: STMicroelectronics datasheet (2013)
<http://www.st.com>

1 Introduction

Finite fields $GF(2^m)$ have received a lot of attention in many applications, such as error-correcting codes and cryptography [1, 2]. The important arithmetic operations involved in these applications are multiplication, inversion/division, and exponentiation. These operations can be carried out using a modular AB multiplier or a modular AB^2 multiplier. For example, the

division is performed using multiplication and multiplicative inverse, that is $A/B = AB^{-1}$, while the inverse can be regarded as a special case of exponentiation, because $B^{-1} = B^{2^m-2} = (B(B(B \cdots B(B(B)^2)^2 \cdots)^2)^2)^2$. In order to compute B^{-1} , the AB^2 can be used recursively. However, as an inverse operation is quite time-consuming, a high-speed circuit is preferable for such operations.

For this purpose, parallel-in parallel-out systolic arrays for AB^2 using the polynomial basis representation in $GF(2^m)$ have been proposed by Wei [3] and Wang and Guo [4]. Note that the systolic design in [3] has the bi-directional data flow, while the circuit in [4] has the unidirectional data flow. Lee et al. [5] proposed a systolic array for computing $AB^2 + C$ over $GF(2^m)$ using the irreducible all-one polynomial. Unfortunately, such irreducible all-one polynomials are very rare. Recently, Kim and Lee [6] proposed a low complexity parallel and serial systolic architectures for AB^2 multiplication in $GF(2^m)$. These circuits still have certain shortcomings as regards various applications due to their high circuit complexity and long cell delay. Thus, further research for an efficient power-sum circuit is needed.

In this paper, we propose an efficient parallel-in parallel-out systolic array with unidirectional data flow for AB^2 over $GF(2^m)$ using the polynomial basis representation. As compared to the related circuits in [3, 4, 6], the proposed one gains advantages in terms of chip area and latency.

2 AB^2 Algorithm in $GF(2^m)$

Let A and B be two elements in $GF(2^m)$ with a primitive polynomial G of degree m , where $A = \sum_{j=0}^{m-1} a_j \alpha^j$, $B = \sum_{j=0}^{m-1} b_j \alpha^j$, $G = \alpha^m + \sum_{j=0}^{m-1} g_j \alpha^j$, $F \equiv \alpha^m = \sum_{j=0}^{m-1} g_j \alpha^j$. The coefficients a_i , b_i , and g_i are in $\{0, 1\}$. Each element in $GF(2^m)$ is a residue mod G and all arithmetic operations are performed by taking the results modulo 2.

It is easy to check that

$$\alpha^{m+1} = \sum_{j=0}^{m-1} (g_{m-1}g_j + g_{j-1})\alpha^j + g_{m-1}g_0 \equiv F' = \sum_{j=0}^{m-1} g'_j \alpha^j. \quad (1)$$

It is well known that the power 2 operation of element B in $GF(2^m)$ can be represented as

$$B^2 = b_{m-1}\alpha^{2(m-1)} + b_{m-2}\alpha^{2(m-2)} + \cdots + b_1\alpha^2 + b_0. \quad (2)$$

The $R = AB^2 \bmod G$ can be expressed as

$$\begin{aligned} R &= \sum_{j=0}^{m-1} Ab_j \alpha^{2j} \\ &= \sum_{j=2}^{m-1} Ab_j \alpha^{2j} + Ab_1 \alpha^2 + Ab_0 \\ &= \left[\sum_{j=2}^{m-1} Ab_j \alpha^{2(j-1)} + Ab_1 \right] \alpha^2 + Ab_0, \end{aligned} \quad (3)$$

where $R = \sum_{j=0}^{m-1} r_j \alpha^j$.

Further expanding the last summations over j in (3), we obtain the following recursion for R :

$$T^{(0)} = 0 \quad (4)$$

$$T^{(i)} = [T^{(i-1)} \alpha^2 + Ab_{m-i}] \bmod G, \text{ for } i = 1, 2, \dots, m \quad (5)$$

$$R = T^{(m)}, \quad (6)$$

where $T^{(i)} = \sum_{j=0}^{m-1} t_j^{(i)} \alpha^j$.

Assume that $T^{(i)}$ can be represented as the sum of $C^{(i)}$, $D^{(i)}$, and $E^{(i)}$, such as

$$T^{(i)} = C^{(i)} + D^{(i)} + E^{(i)}, \quad (7)$$

where $C^{(i)} = \sum_{j=0}^{m-1} c_j^{(i)} \alpha^j$, $D^{(i)} = \sum_{j=0}^{m-1} d_j^{(i)} \alpha^j$, and $E^{(i)} = \sum_{j=0}^{m-1} e_j^{(i)} \alpha^j$.

With F , F' , and (7), we can re-express the i th loop operation as follows:

$$\begin{aligned} T^{(i)} &= [T^{(i-1)} \alpha^2 + b_{m-i} A] \bmod G \\ &= [[C^{(i-1)} + D^{(i-1)} + E^{(i-1)}] \alpha^2 + b_{m-i} A] \bmod G \\ &= [C^{(i-1)} + D^{(i-1)} + E^{(i-1)}] \alpha^2 \bmod \alpha^m \\ &\quad + (c_{m-1}^{(i-1)} + d_{m-1}^{(i-1)} + e_{m-1}^{(i-1)}) \alpha^{m+1} + (c_{m-2}^{(i-1)} + d_{m-2}^{(i-1)} + e_{m-2}^{(i-1)}) \alpha^m + b_{m-i} A \\ &= [C^{(i-1)} \alpha^2 \bmod \alpha^m + b_{m-i} A] + [D^{(i-1)} \alpha^2 \bmod \alpha^m + u_{i-1} F'] \\ &\quad + [E^{(i-1)} \alpha^2 \bmod \alpha^m + v_{i-1} F] \\ &= C^{(i)} + D^{(i)} + E^{(i)}, \end{aligned} \quad (8)$$

where $u_{i-1} = c_{m-1}^{(i-1)} + d_{m-1}^{(i-1)} + e_{m-1}^{(i-1)}$ and $v_{i-1} = c_{m-2}^{(i-1)} + d_{m-2}^{(i-1)} + e_{m-2}^{(i-1)}$.

With (8), we obtain the following recursion for R :

$$C^{(0)} = 0, D^{(0)} = 0, \text{ and } E^{(0)} = 0 \quad (9)$$

$$C^{(i)} = C^{(i-1)} \alpha^2 \bmod \alpha^m + b_{m-i} A \quad (10)$$

$$D^{(i)} = D^{(i-1)} \alpha^2 \bmod \alpha^m + u_{i-1} F' \quad (11)$$

$$E^{(i)} = E^{(i-1)} \alpha^2 \bmod \alpha^m + v_{i-1} F \quad (12)$$

$$R = C^{(m)} + D^{(m)} + E^{(m)}. \quad (13)$$

The coefficients of $C^{(i)}$, $D^{(i)}$, and $E^{(i)}$ can be rewritten in terms of recurrence equations:

$$c_j^{(i)} = c_{j-2}^{(i-1)} + b_{m-i} a_j, \quad (14)$$

$$d_j^{(i)} = d_{j-2}^{(i-1)} + u_{i-1} g'_j, \quad (15)$$

$$e_j^{(i)} = e_{j-2}^{(i-1)} + v_{i-1} g_j, \quad (16)$$

where $0 \leq j \leq m-1$ and $c_{-2}^{(i-1)} = c_{-1}^{(i-1)} = d_{-2}^{(i-1)} = d_{-1}^{(i-1)} = e_{-2}^{(i-1)} = e_{-1}^{(i-1)} = 0$, for $1 \leq i \leq m$.

Equations (14), (15), and (16) can be computed independently and have a similar computation form. Therefore, we can sequentially compute three equations using one processing element. If we design a systolic architecture using above equations, the required latency is $4m + 3$ clock cycles. If 2-bit

Algorithm 1. The proposed AB^2 algorithm

Input : A, B, F, F'

Output : $R = AB^2 \bmod G$

Step 1: $c_j^{(0)} = d_j^{(0)} = e_j^{(0)} = 0$, for $0 \leq j \leq m-1$
 Step 2: $c_{-1}^{(i-1)} = d_{-1}^{(i-1)} = e_{-1}^{(i-1)} = c_{-2}^{(i-1)} = d_{-2}^{(i-1)} = e_{-2}^{(i-1)} = 0$, for $1 \leq i \leq m$
 Step 3: for $i = 1$ to m do begin
 Step 4: $u_{i-1} = c_{m-1}^{(i-1)} + d_{m-1}^{(i-1)} + e_{m-1}^{(i-1)}$, $v_{i-1} = c_{m-2}^{(i-1)} + d_{m-2}^{(i-1)} + e_{m-2}^{(i-1)}$
 Step 5: for $k = m/2 - 1$ to 0 do begin
 Step 6: $c_{2k+1}^{(i)} = c_{2k-1}^{(i-1)} + b_{m-i}a_{2k+1}$, $c_{2k}^{(i)} = c_{2k-2}^{(i-1)} + b_{m-i}a_{2k}$
 Step 7: $d_{2k+1}^{(i)} = d_{2k-1}^{(i-1)} + u_{i-1}g'_{2k+1}$, $d_{2k}^{(i)} = d_{2k-2}^{(i-1)} + u_{i-1}g'_{2k}$
 Step 8: $e_{2k+1}^{(i)} = e_{2k-1}^{(i-1)} + v_{i-1}g_{2k+1}$, $e_{2k}^{(i)} = e_{2k-2}^{(i-1)} + v_{i-1}g_{2k}$
 Step 9: end for
 Step 10: end for
 Step 11: for $k = m/2 - 1$ to 0 do begin
 Step 12: $r_{2k+1} = c_{2k+1}^{(m)} + d_{2k+1}^{(m)} + e_{2k+1}^{(m)}$, $r_{2k} = c_{2k}^{(m)} + d_{2k}^{(m)} + e_{2k}^{(m)}$
 Step 13: end for
 Step 14: return R

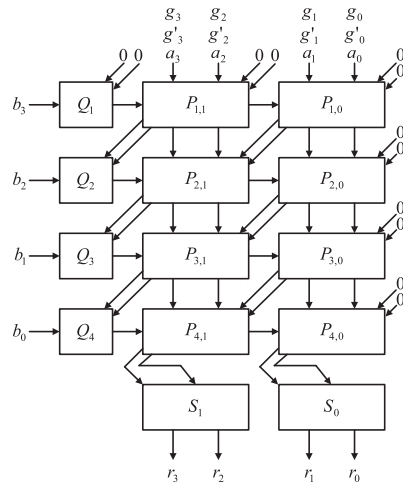


Fig. 1. A two-dimensional SFG array for AB^2 in $GF(2^4)$.

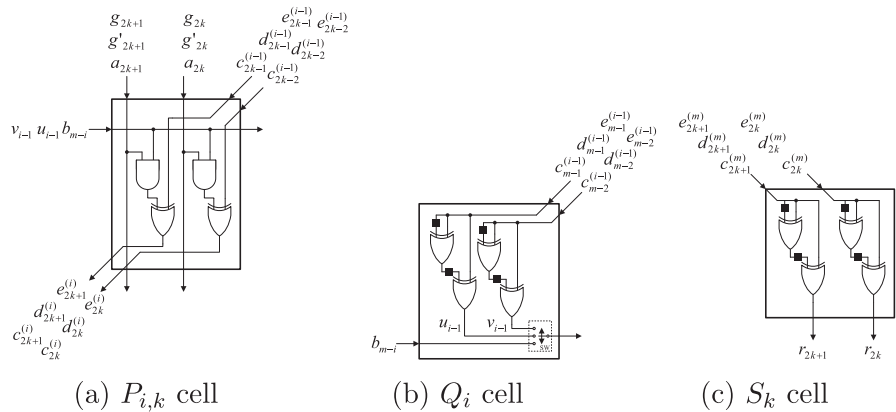


Fig. 2. The detailed circuits.

parallel systolic architecture such as Wang and Guo [4] is adopted, we can reduce the latency. By above equations, we can derive the Algorithm 1.

In Algorithm 1, $c_{2k+1}^{(i)}$ and $c_{2k}^{(i)}$ in Step 6 can be computed in parallel because there is no data dependency. Similarly, $d_{2k+1}^{(i)}(e_{2k+1}^{(i)})$ and $d_{2k}^{(i)}(e_{2k}^{(i)})$ of Step 7(Step 8) can be processed independently. Moreover, since Steps 6, 7,

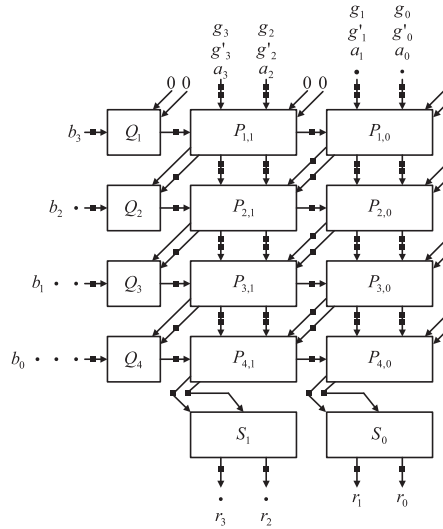


Fig. 3. The proposed systolic array for AB^2 in $GF(2^4)$.

and 8 have an identical computation structure, we can sequentially compute these steps using one processing element.

3 The proposed systolic array for computing AB^2

Fig. 1 shows a two-dimensional signal flow graph (SFG) array to implement Algorithm 1 for AB^2 in $GF(2^m)$. It consists of $m \times m/2$ P cells, m Q cells, and $m/2$ S cells, where these basic cells are illustrated in Fig. 2. The i th row of array realizes the i th iteration from Step 3 to Step 10 of Algorithm 1. As shown in Fig. 1, A , F' , and F enter the array from the top in sequence, and B from the left. When input data pass through the array, the coefficients of R emerge from the bottom row.

Each $P_{i,k}$ cell (Fig. 2 (a)), for $1 \leq i \leq m$ and $0 \leq k \leq m/2 - 1$, consists of two 2-input AND gates (AND_2) and two 2-input XOR gates (XOR_2) and sequentially performs Steps 6, 7, and 8 in Algorithm 1. Each Q_i cell (Fig. 2 (b)), for $1 \leq i \leq m$, is composed of four XOR_2 , one 3×1 switch (SW), and four 1-bit latches ($LATCH$) (denoted by '■' in figures). It performs Step 4 in Algorithm 1. In other words, it computes u_{i-1} and v_{i-1} using data from $P_{i-1,m/2-1}$ cells, for $2 \leq i \leq m$. In addition, it propagates b_{m-i} , u_{i-1} , and v_{i-1} to i th row $P_{i,k}$ cells, for $1 \leq i \leq m$ and $0 \leq k \leq m/2 - 1$. The SW device is used for selecting one of three signals, i.e. b_{m-i} (for the $(3l-1)$ th clock cycle), u_{i-1} (for the $(3l)$ th clock cycle), and v_{i-1} (for the $(3l+1)$ th clock cycle), where $l \geq 1$. Each S_k cell (Fig. 2 (c)), for $0 \leq k \leq m/2 - 1$, consists of four XOR_2 and four 1-bit latches and performs Step 12 in Algorithm 1. In other words, it computes r_{2k+1} and r_{2k} using data from $P_{m,k}$, for $0 \leq k \leq m/2 - 1$. After execution of S cells, the result R will emerge from the bottom of S cells.

By applying the cut-set systolization techniques [7] to the SFG array in Fig. 1, we can derive a parallel-in parallel-out systolic array for AB^2 in $GF(2^m)$ as shown in Fig. 3. It employs $m \times m/2$ P cells, m Q cells, $m/2$ S cells, and $O(m^2)$ latches. Since the architecture is regular, its chip area can

Table I. Comparison of systolic arrays for AB^2 in $GF(2^m)$.

Multiplier	Wei [3]	Wang-Guo [4]	Kim-Lee [6]	Fig. 3		
Cell No.	m^2	$m^2/2$	$(m^2 - m)/2$	$P : m^2/2, Q : m, S : m/2$		
Cell complexity				P	Q	S
AND ₂	3	6	6	2	0	0
XOR ₂	1	0	6	2	4	4
XOR ₃	1	0	0	0	0	0
XOR ₄	0	2	0	0	0	0
Latch	13	17	19	7	6	6
3 × 1 SW	0	0	0	0	1	0
Area	$146m^2$	$110m^2$	$118m^2 - 118m$	$42m^2 + 136m$		
Cell	T_{AND2}	T_{AND2}	$2T_{XOR2}$	T_{XOR2}		
Dealy	$+T_{XOR3}$	$+T_{XOR4}$		$+T_{3 \times 1 SW}$		
Latency	$4m$	$2m + m/2$	$2m + m/2 - 1$	$2m + m/2 + 4$		
Time	$124m$	$77.5m$	$60m - 24$	$90m + 144$		
AT product	$18104m^3$	$8525m^3$	$7080m^3$	$3780m^3$		
			$-9912m^2 + 2832m$	$+18288m^2 + 19584m$		

be estimated based on the cells and latches.

4 Conclusions

In CMOS VLSI technology, each gate is composed of several transistors [8]. We assume that AND_2 , XOR_2 , $LATCH$, and $3 \times 1 SW$ consist of 6, 8, 8, and 16 transistors, respectively. Also, for a further comparison of time complexity, we adopt the practical integrated circuits in [9] and the following assumptions are made: $T_{AND_2}=7$, $T_{XOR_2}=12$, $T_{LATCH}=13$, and $T_{3 \times 1 SW}=24$ where T_{GATE_n} denotes the propagation delay of an n -input gate. Also, we assume that one XOR_3 and one XOR_4 are constructed using two XOR_2 and three XOR_2 .

In this paper, we have presented an efficient parallel-in parallel-out systolic array for AB^2 in $GF(2^m)$. Table I lists some important parameters of the proposed array and the related arrays in [3, 4, 6]. Accordingly, the proposed array saves about 78%, 53%, and 43% (for $m \geq 100$) area-time product as compared to Wei's [3], Wang-Guo's [4], and Kim-Lee's [6], respectively. As compared to the related circuit in [3, 4, 6], the proposed array gains a significant reduction in hardware complexity. Also, the simplicity, regularity and modularity of our proposed architecture allow for easy extension and make this design for implementation using VLSI technologies, particularly for cryptographic applications.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript. This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program.