ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

AN ARCHITECTURE OF $F_{2^{2N}}$ MULTIPLIER FOR ELLIPTIC CURVES CRYPTOSYSTEM

Sarwono Sutikno and Andy Surya

VLSI Research Group
Department of Electrical Engineering
Bandung Institute of Technology
Jl. Ganesha 10, Bandung , Indonesia
Phone: +62-22-2501895, Fax: +62-22-2501895
E-mail: {ssarwono,andy}@vlsi.itb.ac.id

Abstract

Elliptic curves cryptosystem is a potential public key cryptosystem to become the dominant encryption method for information and communication system. This cryptosystem has the same security level compare with other public key cryptosystem, in spite of the relatively short key length that is employed. A short key length makes the encryption and decryption process much faster, lower bandwith for data and more efficient implementation. An implementation of elliptic curves cryptosystem needs a high perfomance finite field arithmetic module. In this paper we will discuss an architecture of finite field $F_{2^{2n}}$ multiplier using normal basis representations. Proposed architecture offers lower computational time and lower complexity architecture compared with other architecture.

1. INTRODUCTION

As the use of internet for business continues to grow, the need of secure and easy to use cryptosystem technologies is becoming more and more critical. Advanced cryptosystem technologies need high performance public key cryptosystems. Elliptic curves cryptosystem is a potential public key cryptosystem to become the dominant encryption method for information and communication system. Elliptic curves cryptosystem was originally developed by Neil Koblitz [1] and Victor S. Miller [2]. Nowadays several public key cryptography schemes using elliptic curves group have been standarized [3].

Using elliptic curves in constructing public key cryptosystem gives several advantages compare with other public key cryptosystems. This cryptosystem use elliptic curves discrete logarithm problem as base of security. Attack on elliptic curves discrete logarithm problem appear to be much harder and more time consuming compare with attack on other hard mathematical problem. Hence elliptic curves cryptosystem can match security of other public key cryptosystem with using short key length. A short key length makes the encryption and decryption process much faster, lower bandwith for data and more efficient implementation.

Performing encryption and decryption process using elliptic curves cryptosystem need point arithmetic computation on selecting elliptic curve and finite field arithmetic computation [4] [5]. Point arithmetic computations need finite field arithmetic computations. Hence implementation of elliptic curves cryptosystem needs a high perfomance computational resource to perform finite field arithmetic .

For finite field representations, prime finite field (F_p) or binary finite field (F_{2^m}) can be used. Binary finite field is more suitable and more practice for hardware implementations rather than prime finite fields. Two common basis can be used for binary finite field are normal basis representation and polynomial basis representation. Normal basis offer several advantages than polynomial basis, therefore we focus on this reperesentation for designing finite field arithmetic computational resource.

Two main operations of finite field arithmetic are addition and multiplication. In normal basis representation, finite field addition is simply perform by bitwise XOR-ing the vector representation. Therefore design of finite field adder with normal basis representation is very simple. Finite field multiplication in normal basis is more complicated so that design of the multiplier is more dificult. Several factors must be considered when designing finite field multiplier: computation time, circuit area and circuit complexity.

In this paper we propose an architecture of finite field F_{2^m} multiplier. This multiplier use finite field with normal basis representation and $m = 2n$ (n odd). Proposed architecture offers lower computational time and lower complexity architecture compared with other architecture.

0-7803-5482-6/99/$10.00 '2000 IEEE

2. NORMAL BASIS REPRESENTATIONS

Finite field F_{2^m} can be viewed as a vector space of dimension m over F_2. There exists a set of m elements $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$ such that each $\alpha \in F_{2^m}$ can be written uniquely in the form

$$\alpha = \sum_{i=0}^{m-1} a_i \alpha_i, \text{ where } a_i \in (0,1) \tag{1}$$

In hardware a field element can be stored in a shift register with dimension m because α can be represented as the bit vector $(a_0, a_1, \ldots, a_{m-1})$.

A normal basis F_{2^m} over F_2 is a basis of the form

$$\{\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{m-1}}\}, \text{ where } \beta \in F_{2^m}$$

β is the generator of the normal basis. Given any element $\alpha \in F_{2^m}$, we can write $\alpha = \sum_{i=0}^{m-1} a_i \beta^{2^i}$, where $a_i \in 0, 1$.

On normal basis representation, addition and squaring are simple operation. Addition of field elements is simply perfomed by bitwise XOR-ing the vector representation and takes only one clock cycle. Squaring is perfomed by using linear operator property in F_{2^m}, thus

$$\alpha^2 = \sum_{i=0}^{m-1} a_i \beta^{2^i+1} = \sum_{i=0}^{m-1} a_{i-1} \beta^{2^i} \tag{2}$$

Squaring can be performed by simple rotation of the vector representation. This operation can be easily implemented using cylic shift register.

Multiplication in normal basis is more complicated. We will discuss more deeply about this multiplication and the implementation of this operation in the next section.

3. F_{2^M} OPTIMAL NORMAL BASIS MULTIPLIER

Let $A = (a_0, a_1, \ldots, a_{m-1})$, $B = (b_0, b_1, \ldots, b_{m-1})$ be arbitrary elements in F_{2^m}, and let $C = A \cdot B = (c_0, c_1, \ldots, c_{m-1})$. Then

$$C = \sum_{k=0}^{m-1} c_k \beta^{2^k} = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_i b_j \beta^{2^i} \beta^{2^j} \tag{3}$$

We can derive above equations to determine c_k as follow

$$c_k = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_{i+k} b_{j+k} \lambda_{ij}^{(0)} \tag{4}$$

Equations (4) can be implemented using Massey-Omura multiplier. Massey Omura use a logic circuit with inputs **A** and **B** to compute c_0. Using same logic circuit, we can rotate **A** and **B** k times to compute c_k. Normal basis representation said optimal (Optimal Normal Basis/ONB) if $C(N) = m-1$. $C(N)$ denote the number of nonzero terms in $\lambda_{ij}^{(0)}$ and determine complexity of logic circuit for multiplications. Figure 1 illustrate F_{2^6} multiplier using Massey Omura structure.

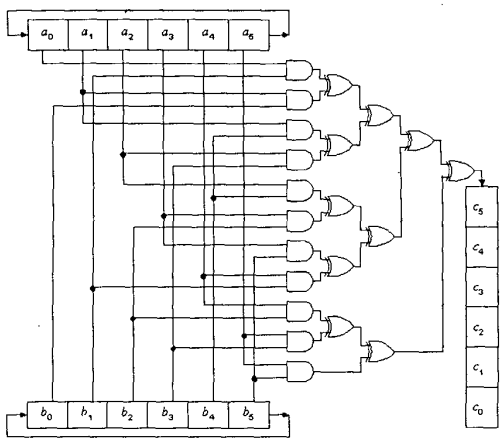

Figure 1: ONB F_{2^6} multiplier (Massey Omura)

Agnew et. al. [8] proposed an alternative architecture. Equations 4 can be written as

$$c_k = \sum_{j=0}^{m-1} b_{j+k} \sum_{i=0}^{m-1} \lambda_{ij}^{(0)} a_{i+k} \tag{5}$$

Let

$$F_j^{(k)} = b_{j+k} \sum_{i=0}^{m-1} \lambda_{ij}^{(0)} a_{i+k} \tag{6}$$

For fixed j and and variable k, the functions F are related by the cyclic permutation of the subscript. Also

$$c_k = \sum_{j=0}^{m-1} F_j^{(k)}, k = 0, 1, \ldots, m-1. \tag{7}$$

The functions $F_j^{(k)}$ are referred to as terms. For non negative integers t, let

$$F_j^{(k)}(t) = b_{j+k+t} \sum_{i=0}^{m-1} \lambda_{ij}^{(0)} a_{i+k+t} \tag{8}$$

Let $A_1, A_2, \ldots, A_m$ and $B_1, B_2, \ldots, B_m$ be cells of cyclic shift registers **A** and **B**, respectively. Define logic cells $C_i, i = 0, 1, 2, \ldots, m-1$ as follow. In cells C_{k_j} let there be a logical circuit which will compute the expression

$$T_{k_j}(t) = \tilde{B}_{j+k_j}(t) \sum_{i=0}^{m-1} \lambda_{ij}^{(0)} \tilde{A}_{i+k_j}(t) \tag{9}$$

where $\bar{A}_p(t)$ and $\bar{B}_q(t)$ are the contents of cells A_p and B_q of **A** and **B**, respectively, at time t. This cell also contain a storage register R_k which can store previously calculated results and can add its contents $\bar{R}_k$ to the value of T_k calculated above.

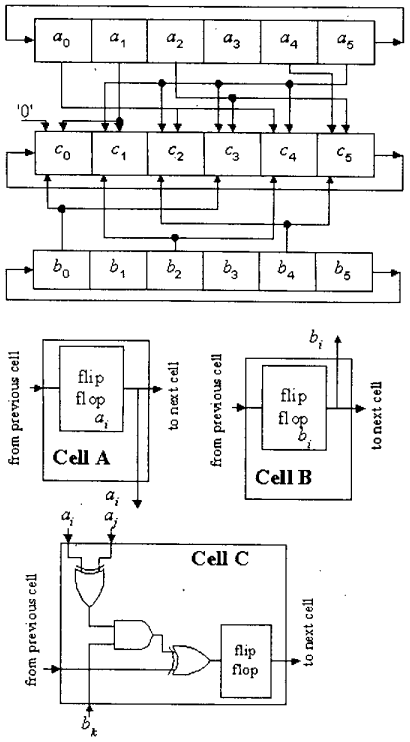

Figure 2: ONB F_{2^6} multiplier (Agnew et. al.)

Figure 2 illustrate register organization and cell structure of F_{2^6} multiplier using Agnew et. al.'s architecture. The system is initialized by loading the values a_i and b_j in the respective cells of A_i and B_j, respectively, and the register R of cells C_i are loaded with zero for $i = 0, 1, \ldots, m-1$. At time t, for $t = 0, 1, \ldots, m-1$, the term $T_k(t)$ is calculated in cell $C_k(t)$ using the current contents of the **A** and **B** registers. The current contents of R_k are XOR'ed with $T_k(t)$ and the results are stored in register $R_{k+1 \bmod n}$. At the end of time $t = m-1$, the register of R_k contain c_k, $k = 0, 1, \ldots, m-1$.

4. EXTENSION FIELD

Finite field normal basis F_{2^k} over F_2 can be extended to produce normal basis $F_{2^{mk}}$ over F_{2^m}. Suppose m and k are positive integers such that gcd(m, k)=1. To produce optimal normal basis $F_{2^{mk}}$ over F_{2^m} when given an optimal normal basis F_{2^k} over F_2 proceed as follows.

Let $K = F_{2^k}$ and $M = F_{2^m}$ be viewed as subfields of $MK = F_{2^{mk}}$. Let $N = \{\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{k-1}}\}$ be a normal basis for K over F_2 viewed as a subfield of MK. Let $\beta_i = \beta^{2^i}, 0 \le i \le k-1$. Since gcd($m, k$)=1 then N is lineary independent over M and is an optimal normal basis for MK over M. Then the set $X = \{\sum_{i=0}^{m-1} a_i \beta^{2^i} : a_i \in M\}$ has cardinality 2^{mk}, and hence is MK.

If

$$\alpha = a_0\beta_0 + a_1\beta_1 + \cdots + a_{k-1}\beta_{k-1} \tag{10}$$

then

$$\alpha^2 = a_{k-1}^2\beta_0 + a_0^2\beta_1 + \cdots + a_{k-2}^2\beta_{k-1} \tag{11}$$

Hence, in the coordinate representations for $F_{2^{mk}}$ over F_{2^m} with respect to N, addition and squaring operations can be perfomed efficiently. Addition is perfomed by adding each the individual coordinates. Squaring is perfomed by squaring each the individual coordinates together with a cyclic shift of the coordinate themselves.

Because of the optimality of the assumed basis for F_{2^k} over F_2, general multiplication is relatively efficient given efficient multiplier for F_{2^m}. We use this idea by setting $k = 2$ and $m = n$ (n odd) to design $F_{2^{2n}}$ optimal normal basis multiplier.

5. $F_{2^{2N}}$ OPTIMAL NORMAL BASIS MULTIPLIER

Based on extension field idea, we proposed a hardware architecture to perform $F_{2^{2n}}$ (n odd) optimal normal basis multiplier. Using modification on architecture proposed by Agnew et. al. can be obtained faster computational time and lower complexity architecture.

Structure of cell on registers **A**, **B** and **C** are modified to implement $F_{2^{2n}}$ multiplier. Each cells contain two register to store each individual coordinates and width of the registers **A**, **B** and **C** are n. F_{2^2} optimal normal basis multiplier is used in each cell of register **C** to implement proposed architecture. Figure 3 illustrate F_{2^2} optimal normal basis multiplier used in cell of register **C**

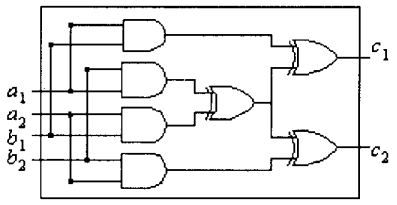

Figure 3: F_{2^2} multiplier

Figure 4 illustrates register organization and cell structure of F_{2^6} multiplier using proposed architecture. The network is considered to operate with the same operation as

Agnew et. al.'s architecture. Using extension field idea, complexity of inteconection between registers **A** and **B** to register **C** can be reduced from $2m-1$ to $2n-1$ ($m=2n$). Also computation time can be reduced from m to $m/2$.

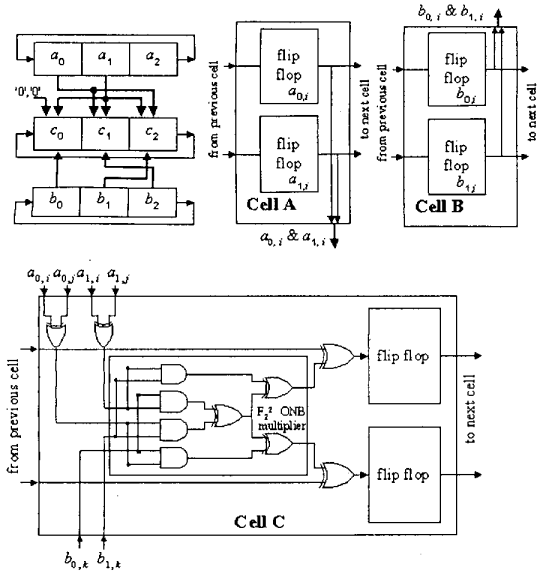

Figure 4: ONB F_{2^6} multiplier (proposed)

6. CONCLUSION

Table 1 shows comparison of proposed architecture compare with Massey Omura's structure and Agnew et. al.'s architecture. The proposed architecture offer lower computational time and lower complexity architecture compare with other architectures.

	Proposed	Massey Omura	Agnew
Computation time	m/2	m	m
Complexity	m-1	2m-1	2m-1
Area:			
FF	3m	3m	3m
XOR	7m/2	2m-2	2m
AND	2m	2m-1	m

Table 1: Comparison (m=2n)

Proposed architecture can be implemented efficiently to obtain high perfomance finite field computational resource. The computational resource can be used to implement elliptic curves cryptosystem scheme. As an example elliptic curves cryptosystem using finite field $F_{2^{166}}$ ($n=83$) arithmetic chip with key length 166 bit has the same security with current public key cryptosystem with key length 1024 bit.

Future work we use the proposed architecture to implement finite field arithmetic chip using SCMOS standard cell technology. The finite field arithmetic chip can be used to implement practical implementation elliptic curves cryptosystem.

7. ACKNOWLEDGMENT

The authors would like to acknowledge the support of RUT VII Grant through KMNRT and LIPI (contract no 40/SP /RUT/1999)

8. REFERENCES

[1] N. Koblitz, "Elliptic Curves Cryptosystems", *Mathematics of Computation,* 48 (1987), 203-209.

[2] V.S. Miller, "Use of Elliptic Curves in Cryptography", *Advances in Cryptology: Proc Crypto '85,* Springer LNCS (1986), 417-426.

[3] IEEE P1363, "Standard Specifications for Public Key Cryptography", 1999. Institute of Electrical and Electronics Engineers, Inc, 1999.

[4] A. Menezes, *Elliptic Curve Public Key Cryptosystems,* Kluwer Academic Publishers, Boston, 1993.

[5] A. Menezes, *Applications of Finite Fields,* Kluwer Academic Publishers, Boston, 1993.

[6] R. Mullin, I. Onyszchuk and S. Vanstone, "Optimal Normal Bases in GF(p^n)", *Discrete Applied Mathematics,* 22 (1988/89), 149-161.

[7] G. Agnew, T. Beth, R. Mullin and S. Vanstone, "Aritmethic Operations in GF(2^m)", *Journal of Cryptology,* 6 (1993), 3-13.

[8] G. Agnew, R. Mullin, I. Onyszchuk and S. Vanstone, "An Implementation for a Fast Public Key Cryptosystem", *Journal of Cryptology,* 3 (1991), 63-79.