Efficient Linear Array for Multiplication in $GF(2^m)$ Using a Normal Basis for Elliptic Curve Cryptography

Soonhak Kwon, Kris Gaj,
Chang Hoon Kim, and Chun Pyo Hong

[1] Inst. of Basic Science and Dept. of Mathematics, Sungkyunkwan University, Suwon 440-746, Korea
shkwon@math.skku.ac.kr

[2] Dept. of Electrical and Computer Engineering, George Mason University, University Drive, Fairfax, VA 22030, USA
kgaj@gmu.edu

[3] Dept. of Computer and Information Engineering, Daegu University, Kyungsan 712-714, Korea
chkim@dsp.taegu.ac.kr,cphong@daegu.ac.kr

Introduction

∘ Finite field multiplication finds various applications in many cryptographic areas such as ECC and AES.

∘ Arithmetic of $GF(2^m)$ is easily realized in a circuit design using a few logical gates.

∘ A normal basis is widely used because it has some good properties such as simple squaring.

∘ Since the parallel architecture has an area complexity of $O(m^2)$, it is not suited for cryptographic application. On the other hand, a sequential multiplier has an area complexity of $O(m)$ and therefore is applicable for ECC.

◦ Reducing the total delay time of a sequential multiplier is very important.

◦ A normal basis multiplier of Massey and Omura has a parallel-in, serial-out structure and has a quite long critical path delay proportional to $\log_2 m$.

◦ Agnew, Mullin, Onyszchuk, and S.A. Vanstone (1991) proposed a sequential multiplier which has a parallel-in, parallel-out structure, where the critical path delay is significantly reduced from that of Massey and Omura.

∘ Reyhani-Masoleh and Hasan (2003) presented two sequential multipliers using a symmetric property of multiplication of normal elements. These multipliers have the reduced area complexity from that of Agnew et al. with a slightly increased critical path delay.

∘ For the case of a type II ONB, the critical path delay of Reyhani-Masoleh and Hasan is T_A+3T_X while that of Agnew et al. is $T_A + 2T_X$.

∘ Since we are dealing with a sequential (linear) multiplier, even a small increment of critical path delay such as T_X results in a total delay of mT_X where m is the size of a field.

Purpose of this paper

- Our aim in this paper is to present a sequential multiplier using a Gaussian normal basis in $GF(2^m)$ for odd m.
- Since choosing an odd m is a necessary condition for cryptographic purposes and since a low complexity normal basis is frequently a Gaussian normal basis of type (m, k) for low k, our restriction in this paper does not cause any serious problem for practical purposes.
- All the five recommended fields $GF(2^m)$ by NIST for ECC where $m = 163, 233, 283, 409, 571$ can be dealt using our Gaussian normal basis, and the corresponding circuits are easy to construct.

- We will show that

the area complexity of our linear multiplier
$=$ that of Reyhani-Masoleh and Hasan
$<<$ that of Agnew et al.

the critical path delay of ours
$=$ that of Agnew et al.
$\leq$ that of Reyhani-Masoleh and Hasan

Review of the linear normal basis multipliers

Letting $A = \sum_{i=0}^{m-1} a_i \alpha_i$ and $B = \sum_{j=0}^{m-1} b_j \alpha_j$ in $GF(2^m)$, we have the multiplication $C = AB = \sum_{s=0}^{m-1} c_s \alpha_s$ where

$$C = \sum_{i,j} a_i b_j \alpha_i \alpha_j = \sum_{i,j} a_i b_j \sum_{s=0}^{m-1} \lambda_{ij}^{(s)} \alpha_s$$

$$= \sum_{s=0}^{m-1} (\sum_{i,j} a_i b_j \lambda_{ij}^{(s)}) \alpha_s.$$

Therefore, we have the coefficients c_s of $C = AB$ as

$$c_s = \sum_{i,j} a_i b_j \lambda_{ij}^{(s)} = \sum_{i,j} a_i b_j \lambda_{i-s,j-s}^{(0)} = \sum_{i,j} a_{i+s} b_{j+s} \lambda_{ij}^{(0)}$$

Linear Type II Optimal Normal Basis Multiplier by Agnew et al. (1991)

Example

$m=5, \quad A, B \in GF(2^5)$

$A = [a_0, a_1, a_2, a_3, a_4]$

$B = [b_0, b_1, b_2, b_3, b_4]$

$C = AB = [c_0, c_1, c_2, c_3, c_4]$

$$c_s = \sum_{i.i} a_{i+s} b_{j+s} \lambda_{ij}^{(0)}$$

where

$$\lambda_{ij}^{(0)} = 1 \quad \text{iff} \quad 2^i \pm 2^j \equiv \pm 1 \pmod{2m+1}$$

$$0 \quad \text{otherwise}$$

Idea of the circuit by Agnew et al.

Example

$$c_0 = \underline{a_1 b_0} + (a_0+a_3)\, b_1 + (a_3+a_4)\, b_2 + (a_1+a_2)\, b_3 + (a_2+a_4)\, b_4$$

$$c_1 = a_2 b_1 + \underline{(a_1+a_4)\, b_2} + (a_4+a_0)\, b_3 + (a_2+a_3)\, b_4 + (a_3+a_0)\, b_0$$

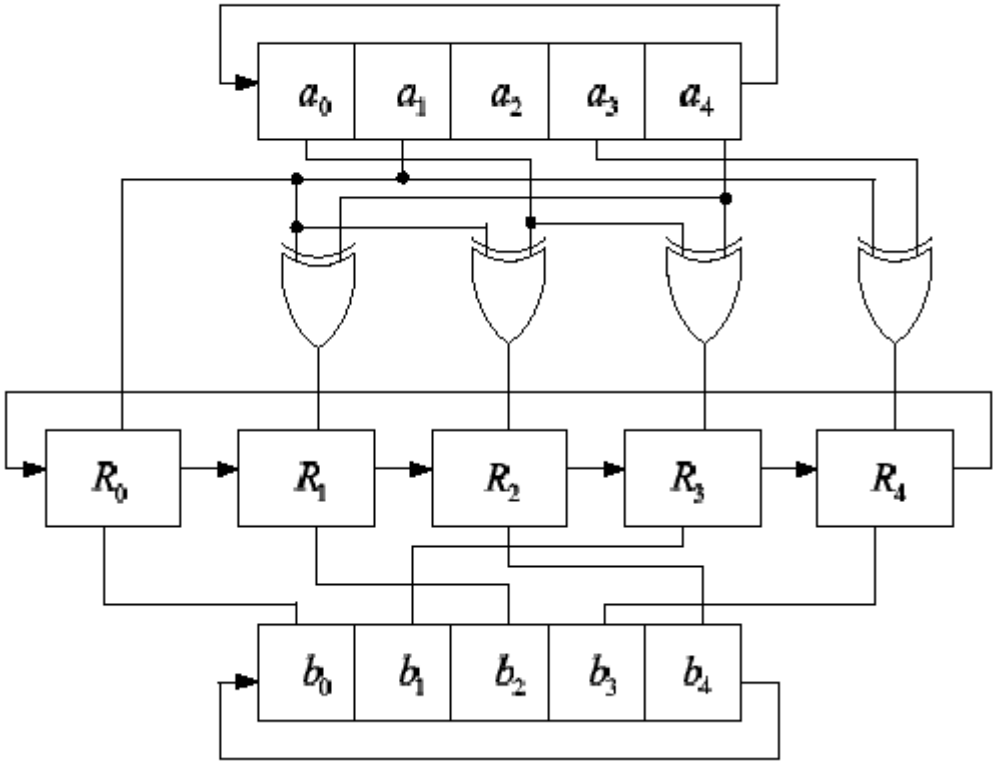
$$c_2 = a_3 b_2 + (a_2+a_0)\, b_3 + \underline{(a_0+a_1)\, b_4} + (a_3+a_4)\, b_0 + (a_4+a_1)\, b_1$$

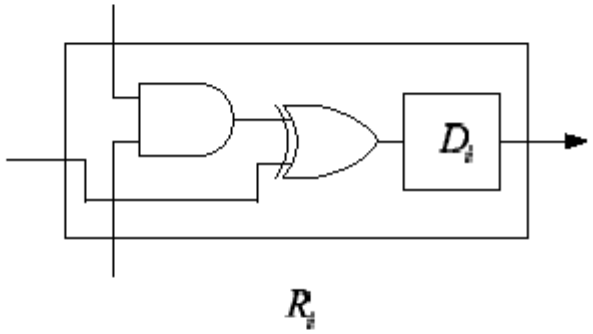
$$c_3 = a_4 b_3 + (a_3+a_1)\, b_4 + (a_1+a_2)\, b_0 + \underline{(a_4+a_0)\, b_1} + (a_0+a_2)\, b_2$$

$$c_4 = a_0 b_4 + (a_4+a_2)\, b_0 + (a_2+a_3)\, b_1 + (a_0+a_1)\, b_2 + \underline{(a_1+a_3)\, b_3}$$

Linear Normal Basis Multiplier by Agnew et al.

Example

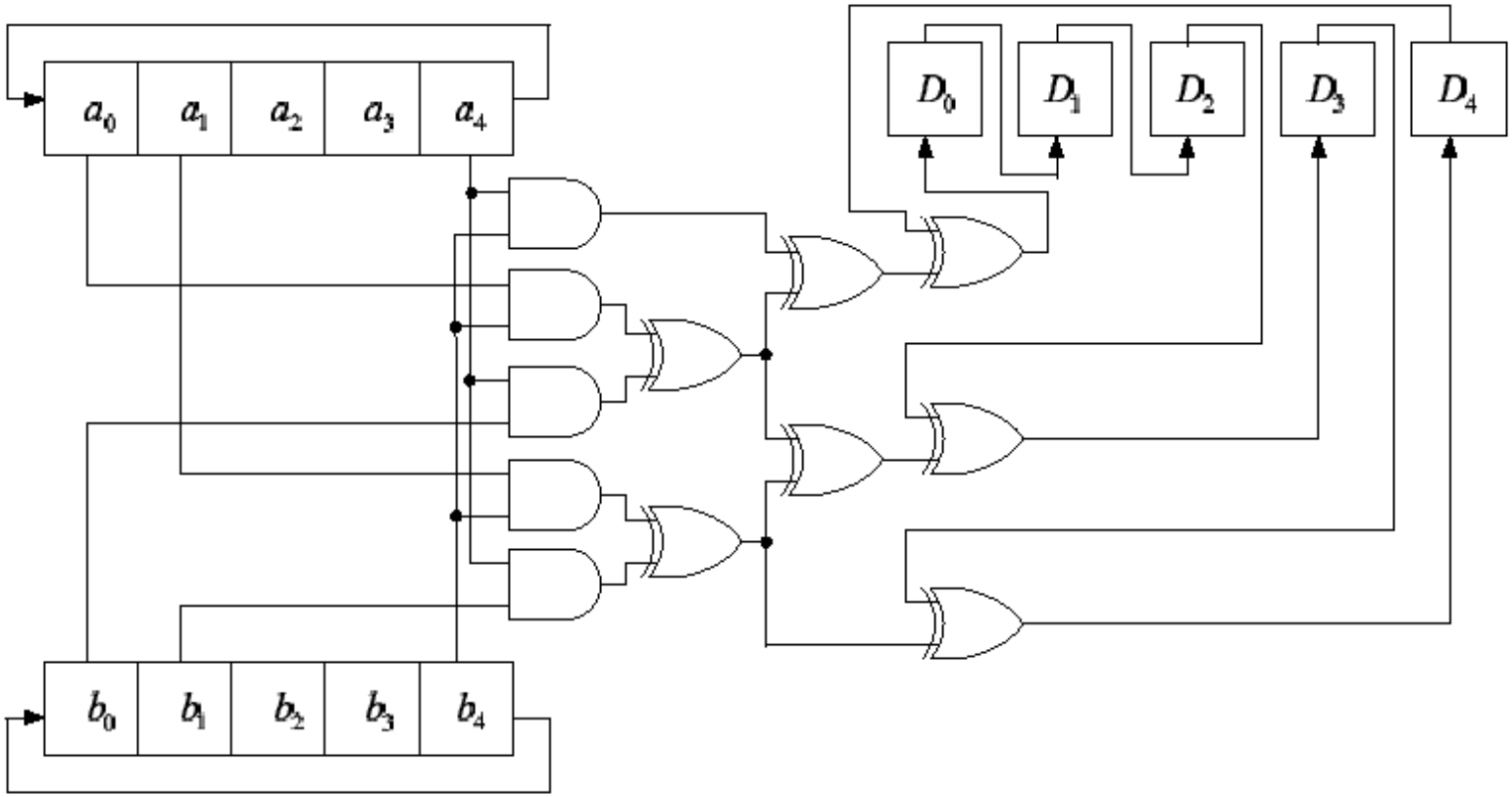




Critical path delay = T_A + **2** $\mathbf{T_X}$

Area = 15 A_D + **9** $\mathbf{A_X}$ + 5 A_A

Multiplier by Reyhani-Masoleh and Hasan (2003)



Critical path delay = T_A + **3** $\mathbf{T_X}$

Area = 15 A_D + **7** $\mathbf{A_X}$ + 5 A_A

Gaussian normal basis of type k in $GF(2^m)$

Let m, k be positive integers such that $p = mk + 1$ is a prime $\neq 2$. Let $K = \langle \tau \rangle$ be a unique subgroup of order k in $GF(p)^{\times}$. Let β be a primitive pth root of unity in $GF(2^{mk})$. The following element

$$\alpha = \sum_{j=0}^{k-1} \beta^{\tau^j}$$

is called a Gauss period of type (m, k) over $GF(2)$.

Let $ord_p 2$ be the order of $2 \pmod{p}$ and assume $gcd(mk/ord_p 2, m) = 1$.

Gaussian normal basis of type k in $GF(2^m)$

Then it is well known that α is a normal element in $GF(2^m)$. That is, letting $\alpha_i = \alpha^{2^i}$ for $0 \leq i \leq m-1$, $\{\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{m-1}\}$ is a basis for $GF(2^m)$ over $GF(2)$.

It is called a Gaussian normal basis of type k or (m, k) in $GF(2^m)$.

Among the five binary fields recommended by NIST,

- $m = 233$ is the only case where a type II ONB exists.

On the other hand, the lowest complexity Gaussian normal basis for the rest of the fields are

- type 4 Gaussian normal basis when $m = 163, 409$,
- type 6 Gaussian normal basis when $m = 283$,
- and type 10 Gaussian normal basis when $m = 571$.

Our new multiplication algorithm

1. $A = \sum_{i=0}^{m-1} a_i \alpha_i$ and $B = \sum_{i=0}^{m-1} b_i \alpha_i$ are loaded in m-bit registers respectively. intermediate values $D_0, D_1, \cdots, D_{m-1}$ of the multiplication are all set to zero.

2. For $t = 0$ to $m - 1$, do the following;

$$y_{s,s+t} + D_{s+t} \longrightarrow D_{s+t+1},$$

   where the above computation is done in parallel for all $0 \leq s \leq m - 1$.

3. After mth iteration, we have $D_i = c_i$ for all $0 \leq i \leq m - 1$, where $AB = \sum_{i=0}^{m-1} c_i \alpha_i$

At the first cycle $(t = 0)$,

$$D_1 = y_{00}, D_2 = y_{11}, \cdots, D_0 = y_{m-1,m-1}.$$

When $t = 1$,

$$D_2 = D_1 + y_{01} = y_{00} + y_{01},$$
$$D_3 = D_2 + y_{12} = y_{11} + y_{12},$$
$$\cdots,$$
$$D_1 = D_0 + y_{m-1,0} = y_{m-1,m-1} + y_{m-1,0}.$$

Finally, at m*th* $(t = m - 1)$ *cycle,*

$$\begin{aligned} D_0 &= D_{m-1} + y_{0,m-1} \\ &= y_{00} + y_{01} + \cdots + y_{0,m-1} = c_0, \\ D_1 &= D_0 + y_{10} \\ &= y_{11} + y_{12} + \cdots + y_{10} = c_1, \\ &\cdots\cdots \\ &\cdots\cdots \\ D_{m-1} &= D_{m-2} + y_{m-1,m-2} \\ &= y_{m-1,m-1} + y_{m-1,0} + \cdots + y_{m-1,m-2} = c_{m-1}. \end{aligned}$$

Multiplier for a Gaussian normal basis of type 2 (= Optimal Normal Basis of type 2)

Example 1 (m=5, k=2)

Step 1: Computing intermediate tables K

$\{\tau$=-1, 1$\}$ a multiplicative subgroup of order k of GF(p=mk+1)

	K_0	K_1	K_2	K_3	K_4	K'_0	K'_1	K'_2	K'_3	K'_4
s=0	1	2	4	8	5	2	3	5	9	6
s=1	−1	−2	−4	−8	−5	0	−1	−3	−7	−4

$K_t = \tau^s 2^t$ $\qquad$ $K'_t = 1 + \tau^s 2^t$

Multiplier for a Gaussian normal basis of type 2 (= Optimal Normal Basis of type 2)

Example 1 (m=5, k=2)

Step 2: Computing matrix (λ_{ij})

$0 \le t \le 4$ $\qquad$ $0 \le t \le 4$

	K_0	K_1	K_2	K_3	K_4	K'_0	K'_1	K'_2	K'_3	K'_4
s=0	1	2	4	8	5	2	3	5	9	6
s=1	−1	−2	−4	−8	−5	0	−1	−3	−7	−4

λ_{ij} = 1 iff $K'_j \equiv K_i \bmod km+1$ $\qquad$ $\lambda_{31}=1$

0 otherwise

Multiplier for a Gaussian normal basis of type 2 (= Optimal Normal Basis of type 2)

Example 1 (m=5, k=2)

Step 3: Writing equations for c_s

$$(\lambda_{ij}) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$c_0 = \underline{(a_3 + a_4)b_2} + a_1 b_0 + (a_1 + a_2)b_3 + (a_0 + a_3)b_1 + (a_2 + a_4)b_4$$

$$c_1 = (a_4 + a_0)b_3 + \underline{a_2 b_1} + (a_2 + a_3)b_4 + (a_1 + a_4)b_2 + (a_3 + a_0)b_0$$

$$c_2 = (a_0 + a_1)b_4 + a_3 b_2 + \underline{(a_3 + a_4)b_0} + (a_2 + a_0)b_3 + (a_4 + a_1)b_1$$

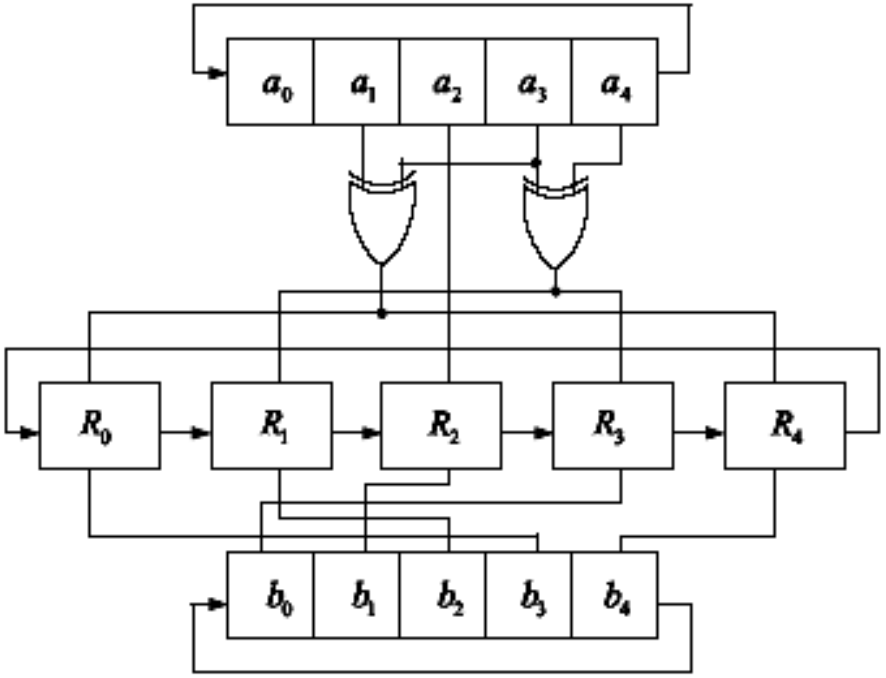
$$c_3 = (a_1 + a_2)b_0 + a_4 b_3 + (a_4 + a_0)b_1 + \underline{(a_3 + a_1)b_4} + (a_0 + a_2)b_2$$

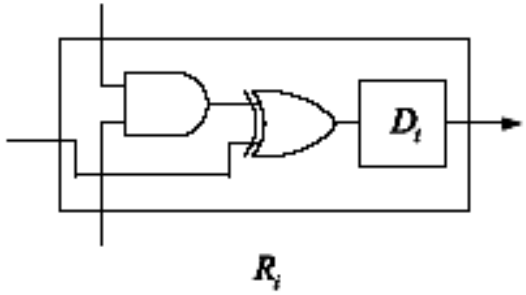
$$c_4 = (a_2 + a_3)b_1 + a_0 b_4 + (a_0 + a_1)b_2 + (a_4 + a_2)b_0 + \underline{(a_1 + a_3)b_3}$$

Multiplier for a Gaussian normal basis of type 2 (= Optimal Normal Basis of type 2)

Example 1 (m=5, k=2)

Step 4: Constructing a corresponding circuit





Multiplier for a Gaussian normal basis of type 4

Example 2 (m=7, k=4)

Step 1: Computing intermediate tables K

$\{\tau$=12, 28, 17, 1$\}$ a multiplicative subgroup of order k of GF(p=km+1)

	K_0	K_1	K_2	K_3	K_4	K_5	K_6	K'_0	K'_1	K'_2	K'_3	K'_4	K'_5	K'_6
s=0	1	2	4	8	16	3	6	2	3	5	9	17	4	7
s=1	12	24	19	9	18	7	14	13	25	20	10	19	8	15
s=2	28	27	25	21	13	26	23	0	28	26	22	14	27	24
s=3	17	5	10	20	11	22	15	18	6	11	21	12	23	16

$K_t = \tau^s 2^t$ $\qquad$ $K'_t = 1 + \tau^s 2^t$

Multiplier for a Gaussian normal basis of type 4

Example 2 (m=7, k=4)

Step 2a: Computing matrix (λ_{ij})

	K_0	K_1	K_2	K_3	K_4	K_5	K_6	K'_0	K'_1	K'_2	K'_3	K'_4	K'_5	K'_6
s=0	1	2	4	8	16	3	6	2	3	5	9	17	4	7
s=1	12	24	19	9	18	7	14	13	25	20	10	19	8	15
s=2	28	27	25	21	13	26	23	0	28	26	22	14	27	24
s=3	17	5	10	20	11	22	15	18	6	11	21	12	23	16

$\lambda_{52}=1$

$\lambda_{ij} = 1$ iff $K'_j \equiv K_i \bmod km+1$

0 otherwise

Multiplier for a Gaussian normal basis of type 4

Example 2 (m=7, k=4)

Step 2b: Computing matrix (λ_{ij})

$$(\lambda_{ij}) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Multiplier for a Gaussian normal basis of type 4

Example 2 (m=7, k=4)

Step 3: Writing equations for c_s

$$c_0 = \underline{(a_2 + a_5)b_3} + a_{0256}b_1 + a_{1456}b_6 + (a_2 + a_6)b_4 + a_{1345}b_2 + a_1b_0 + a_{1236}b_5$$
$$c_1 = (a_3 + a_6)b_4 + \underline{a_{1360}b_2} + a_{2560}b_0 + (a_3 + a_0)b_5 + a_{2456}b_3 + a_2b_1 + a_{2340}b_6$$
$$c_2 = (a_4 + a_0)b_5 + a_{2401}b_3 + \underline{a_{3601}b_1} + (a_4 + a_1)b_6 + a_{3560}b_4 + a_3b_2 + a_{3451}b_0$$
$$c_3 = (a_5 + a_1)b_6 + a_{3512}b_4 + a_{4012}b_2 + \underline{(a_5 + a_2)b_0} + a_{4601}b_5 + a_3b_3 + a_{4562}b_1$$
$$c_4 = (a_6 + a_2)b_0 + a_{4623}b_5 + a_{5123}b_3 + (a_6 + a_3)b_1 + \underline{a_{5012}b_6} + a_4b_4 + a_{5603}b_2$$
$$c_5 = (a_0 + a_3)b_1 + a_{5034}b_6 + a_{6234}b_4 + (a_0 + a_4)b_2 + a_{6123}b_0 + \underline{a_6b_5} + a_{6014}b_3$$
$$c_6 = (a_1 + a_4)b_2 + a_{6145}b_0 + a_{0345}b_5 + (a_1 + a_5)b_3 + a_{0234}b_1 + a_0b_6 + \underline{a_{0125}b_4}$$

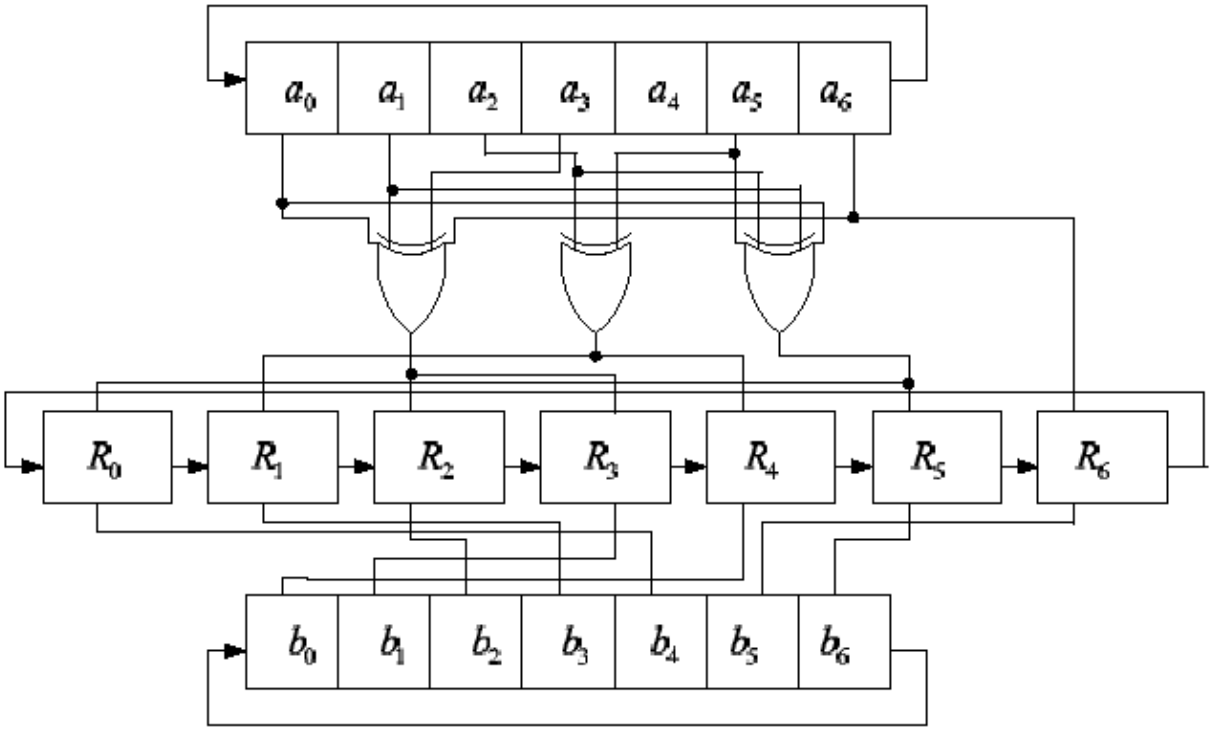
where

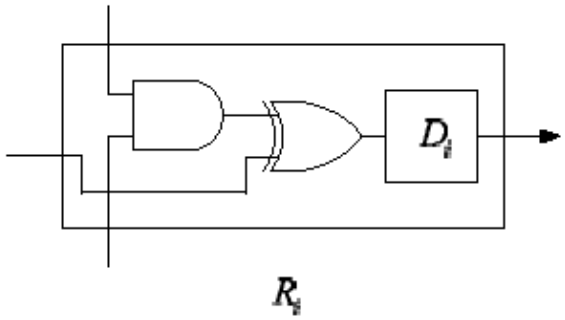
$$a_{ijkl} = a_i + a_j + a_k + a_l$$

Multiplier for a Gaussian normal basis of type 4

Example 2 (m=7, k=4)

Step 4: Constructing a corresponding circuit





Comparison with previously proposed architectures

	Critical path delay (Type II ONB case)	AND	XOR (Type II ONB case)	flip-flop
Massey and Omura [7]	$\leq T_A + \lceil \log_2 (mk) \rceil T_X$ $(T_A + \lceil \log_2 (2m) \rceil T_X)$	C_N	$\leq C_N - 1$ $(2m-2)$	$2m$
Agnew et al. [1]	$\leq T_A + (1 + \lceil \log_2 k \rceil) T_X$ $(T_A + 2T_X)$	m	$\leq C_N$ $(2m-1)$	$3m$
Reyhani-Masoleh and Hasan [3]	$\leq T_A + (1 + \lceil \log_2 (k+2) \rceil) T_X$ $(T_A + 3T_X)$	m	$\leq \frac{1}{2}(C_N + 1) + \lfloor \frac{m}{2} \rfloor$ $(\frac{3m-1}{2})$	$3m$
This paper	$\leq T_A + (1 + \lceil \log_2 k \rceil) T_X$ $(T_A + 2T_X)$	m	$\leq m + \frac{m-1}{2}(k-1)$ $(\frac{3m-1}{2})$	$3m$

Conclusions

- We proposed a low complexity sequential normal basis multiplier over $GF(2^m)$ for odd m using a Gaussian normal basis of type k.
- We presented a general method of constructing a circuit arrangement of the multiplier and showed explicit examples for the cases of type 2 and 4 Gaussian normal bases.
- Among the five binary fields, $GF(2^m)$ with $m = 163, 233, 283, 409, 571$, recommended by NIST for ECC, our examples cover the cases $m = 163, 233, 409$ since $GF(2^{233})$ has a type II ONB and $GF(2^{163}), GF(2^{409})$ have a Gaussian normal basis of type 4.

- Our general method can also be applied to other fields $GF(2^{283})$ and $GF(2^{571})$ since they have a Gaussian normal basis of type 6 and 10, respectively.
- Compared with previously proposed architectures of the same kinds, our multiplier has a superior or comparable area complexity and delay time.
- Thus it is well suited for many applications such as VLSI implementation of elliptic curve cryptographic protocols.