A Fast Algorithm for Computing Multiplicative Inverses in GF(2^m) Using Normal Bases

TOSHIYA ITOH AND SHIGEO TSUJI

Department of Electrical and Electronic Engineering, Faculty of Engineering, Tokyo Institute of Technology, Tokyo 152, Japan

This paper proposes a fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. Normal bases have the following useful property: In the case that an element \( x \) in GF(2^m) is represented by normal bases, \( 2^k \) power operation of an element \( x \) in GF(2^m) can be carried out by \( k \) times cyclic shift of its vector representation. C. C. Wang et al. proposed an algorithm for computing multiplicative inverses using normal bases, which requires \( (m - 2) \) multiplications in GF(2^m) and \( (m - 1) \) cyclic shifts. The fast algorithm proposed in this paper also uses normal bases, and computes multiplicative inverses iterating multiplications in GF(2^m). It requires at most \( 2 \lceil \log_2(m - 1) \rceil \) multiplications in GF(2^m) and \( (m - 1) \) cyclic shifts, which are much less than those required in the Wang's method. The same idea of the proposed fast algorithm is applicable to the general power operation in GF(2^m) and the computation of multiplicative inverses in GF(q^m) (\( q = 2^n \)).

1. INTRODUCTION

Finite field arithmetic is widely used in various fields, such as coding theory and cryptography and so on. Most of public-key cryptosystems are constructed over finite fields of large order, hence their running-time of encryption and decryption is dominated by multiplication and division. Therefore, it is very important in a practical sense to develop a fast algorithm for carrying out such operations.

This paper proposes a fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. C. C. Wang et al. proposed an algorithm for computing multiplicative inverses in GF(2^m) using normal bases, which requires \( (m - 2) \) multiplications in GF(2^m) and \( (m - 1) \) cyclic shifts. The algorithm proposed in this paper also uses normal bases in GF(2^m), and requires at most \( 2 \lceil \log_2(m - 1) \rceil \) (\([x]\): Gauss' symbol) multiplications in GF(2^m) and \( (m - 1) \) cyclic shifts to compute multiplicative inverses in GF(2^m). It is also shown that the same idea of the proposed algorithm is applicable to the computation of multiplicative inverses in GF(q^m) (\( q = 2^n \)).
2. Preliminaries

Definition (MacWilliams and Sloane, 1979). A normal basis of \( GF(q^m) \) \( (q = 2^n) \) over \( GF(q) \) is a basis of the form

\[
a, a^q, a^{q^2}, \ldots, a^{q^{m-1}},
\]

where "\( a \)" is a non-zero element in \( GF(q^m) \) \( (q = 2^n) \). (\#)

Lemma 1 (Itoh and Tsujii, 1986). Let an element \( x \) in \( GF(q^m) \) \( (q = 2^n) \) be represented by a normal basis (Eq. 1) in the form

\[
x = x_0 a + x_1 a^q + \cdots + x_{m-1} a^{q^{m-1}} = [x_0, x_1, \ldots, x_{m-1}],
\]

where \( \{a, a^q, \ldots, a^{q^{m-1}}\} \) is a normal basis over \( GF(q) \). Then, \( x^{q^k} \) can be computed by \( k \) cyclic shifts of Eq. (2) such that

\[
x^{q^k} = [x_{m-k}, x_{m-k+1}, \ldots, x_{m-1}, x_0, \ldots, x_{m-k-1}].
\]

We call the cyclic shift in Eq. (3) "cyclic shift over \( GF(q) \)" in the rest of this paper.

Lemma 2 (MacWilliams and Sloane, 1979). Every element \( e \) in \( GF(q^m) \) \( (q = 2^n) \) satisfies the identity

\[
e^{q^m} = e.
\]

3. The Wang’s Method

(Wang et al., 1985). A non-zero element \( x \) in \( GF(2^m) \) has an unique multiplicative inverse \( x^{-1} \). Since the non-zero element \( x \) also satisfies Lemma 2, i.e., \( x^{2^m} = x \), \( x^{-1} \) is given by \( x^{-1} = x^{2^m-2} \). Here \( 2^m-2 \) can be represented by \( 2^m-2 = 2^1 + 2^2 + \cdots + 2^{m-1} \), hence \( x^{-1} \) can be computed by

\[
x^{-1} = (x^2)(x^2^2)\cdots(x^{2^{m-1}}).
\]

The following algorithm shows the procedure of computing Eq. (5).

Algorithm 1.

S1. \( y := x \)
S2. \( \text{for } k := 1 \text{ to } m-2 \text{ do} \)
S3. \( \text{begin} \)
S4. \( z := y^2 \) (one cyclic shift)
S5. \( y := z x \) (multiplication in \( GF(2^m) \))
S6. \( \text{end} \)
S7. \( y := y^2 \) (one cyclic shift)
S8. \( \text{write } y \)
MULTIPLICATIVE INVERSSES IN FINITE FIELDS

By Lemma 1 and Algorithm 1, computing Eq. (5) requires \((m - 2)\) multiplications in \(GF(2^m)\) and \((m - 1)\) cyclic shifts over \(GF(2)\).

4. PROPOSED FAST ALGORITHM IN \(GF(2^m)\)

By Lemma 1, we have the following theorem.

**Theorem 1.** Let \(x\) be a non-zero element in \(GF(2^m)\) \((m = 2^r + 1)\). Then, there exists an algorithm for computing \(x^{-1}\), which requires

- number of multiplications in \(GF(2^m)\): \(NM = \log_2(m - 1) = r\),
- number of cyclic shifts over \(GF(2)\): \(NS = m - 1 = 2^r\).

**Proof.** Represent \(2^m - 2\) in binary form:

\[
2^m - 2 = (1, 1, \ldots, 1, 0), \quad \text{where} \quad m - 1 = 2^r,
\]

and define the following symbols to simplify the notation:

\[
> t = (1, 1, \ldots, 1), \quad (7)
\]
\[
\wedge t = (1, 1, \ldots, 1), \quad (8)
\]
\[
\& t = 2^r, \quad (9)
\]
\[
\uparrow t = 2^{2t}. \quad (10)
\]

Let \(M_t\) and \(S_t\) be the number of multiplications in \(GF(2^m)\) and cyclic shifts over \(GF(2)\) to compute \(x^\wedge t\) \((1 \leq i \leq r)\), respectively. Since \(x^\wedge t = (x^\wedge (t - 1))(x^\wedge (t - 1))\), we have \(M_t = M_{t - 1} + 1\) and \(S_t = S_{t - 1} + 2^{t - 1}\), where \(M_0 = S_0 = 0\). Hence \(M_r = r\) and \(S_r = 2^r - 1\), and thus

\[
NM = M_r = r = \log_2(m - 1), \quad (11)
\]
\[
NS = S_r + 1 = 2^r = m - 1, \quad (12)
\]

because \(x^{-1} = (x^\wedge r)^2\).

Theorem 1 can be described by

**Algorithm 2.**

1. \(y := x\)
2. for \(k := 0\) to \(r - 1\) do
3. begin
4. \(z := y^{2^k}\) \((2^k\) cyclic shifts)
5. \(y := yz\) \((multiplication \ in \ GF(2^m))\)
6. end
7. \(y := y^2\) \((multiplication \ in \ GF(2^m))\)
8. write \(y\).

The following theorem is the generalization of Theorem 1.
THEOREM 2. Let \( x \) be a non-zero element in \( \text{GF}(2^m) \). Then, there exists an algorithm for computing \( x^{-1} \), which requires

- number of multiplications in \( \text{GF}(2^m) \):
  \[
  \text{NM} = \left\lfloor \log_2(m - 1) \right\rfloor + H_w(m - 1) - 1 \leq 2\left\lfloor \log_2(m - 1) \right\rfloor,
  \]
- number of cyclic shifts over \( \text{GF}(2) \): \( \text{NS} = m - 1 \),

where \( \left\lfloor \cdot \right\rfloor \) = Gauss' symbol and \( H_w(\cdot) \) = Hamming weight.

Proof. \( x^{-1} \) can be computed by

\[
  x^{-1} = (x^{(m - 1)})^2.
\]

Suppose that \( m - 1 \) is represented by

\[
  m - 1 = \sum_{s=1}^{\ell} 2^{k_s}, \quad \text{where} \quad k_1 > k_2 > \cdots > k_{\ell},
\]

and so we have

\[
  x^{-1} = \{(x^{k_1}) \circ e_1 (x^{k_2}) \circ e_2 \cdots (x^{k_{\ell}}) \circ e_{\ell}\}^2,
\]

where \( e_s = \sum_{s=s+1}^{\ell} 2^{k_s} \) and \( e_s = 0 \).

Reordering the terms in Eq. (15),

\[
  x^{-1} = \{(x^{k_1})(x^{k_{s-1}}) \cdots ((x^{k_2})(x^{k_1}) \circ k_2) \circ k_3 \cdots \} \circ k_{\ell}\}^2.
\]

Let \( M(k_1) \) and \( S(k_1) \) be the number of multiplications in \( \text{GF}(2^m) \) and cyclic shifts over \( \text{GF}(2) \) to compute \( x^{k_1} \), respectively. Here we have \( M(k_1) = k_1 \) and \( S(k_1) = 2^{k_1} - 1 \). Since every term \( x^{k_s} (2 \leq s \leq \ell) \) is already computed in the procedure of computing \( x^{k_1} \) (see proof of Theorem 1), we have \( \text{NM} = k_1 + n - 1 \) and \( \text{NS} = 2^{k_1} - 1 + \sum_{s=2}^{n} 2^{k_s} + 1 \). By the the fact that \( \left\lfloor \log_2(m - 1) \right\rfloor = k_1 \) and \( H_w(m - 1) = n \), thus

\[
  \text{NM} = \left\lfloor \log_2(m - 1) \right\rfloor + H_w(m - 1) - 1 \leq 2\left\lfloor \log_2(m - 1) \right\rfloor,
\]

\[
  \text{NS} = \sum_{s=1}^{n} 2^{k_s} = m - 1.
\]

A result similar to that of Theorem 2 has been found independently by S. A. Vanstone (1987).

5. Example

The following example confirms Theorem 2:
Let $x$ be a non-zero element in $\text{GF}(2^{11})$. Here $x^{-1}$ is given by $x^{-1} = x^{2^{11} - 2} = x^{2046}$, hence $x^{-1}$ can be computed by the following procedure:

1. $(x^2)^2 = x^2$ : 1 cyclic shift over $\text{GF}(2)$
2. $x^2 x = x^3$ : 1 multiplication in $\text{GF}(2^{11})$
3. $(x^3)^2 = x^{12}$ : 2 cyclic shifts over $\text{GF}(2)$
4. $x^{12} x^3 = x^{15}$ : 1 multiplication in $\text{GF}(2^{11})$
5. $(x^{15})^{2^2} = x^{240}$ : 4 cyclic shifts over $\text{GF}(2)$
6. $x^{240} x^{15} = x^{255}$ : 1 multiplication in $\text{GF}(2^{11})$
7. $(x^{255})^{2^2} = x^{1020}$ : 2 cyclic shifts over $\text{GF}(2)$
8. $x^{1020} x^3 = x^{1023}$ : 1 multiplication in $\text{GF}(2^{11})$
9. $(x^{1023})^2 = x^{2046}$ : 1 cyclic shift over $\text{GF}(2)$

Observing the above procedure, the number of multiplications in $\text{GF}(2^{11})$ and cyclic shifts over $\text{GF}(2)$ are as follows:

- 4 multiplications (in $\text{GF}(2^{11})$) in S2, S4, S6, and S8.
- 10 cyclic shifts (over $\text{GF}(2)$) in S1, S3, S5, S7, and S9.

On the other hand, since $[\log_2(11 - 1)] = 3$ and $H_\omega(11 - 1) = 2$, we have $\text{NM}(= [\log_2(11 - 1)] + H_\omega(11 - 1) - 1) = 4$ and $\text{NS}(= 11 - 1) = 10$, and this example confirms Theorem 2.

6. PROPOSED FAST ALGORITHM IN $\text{GF}(q^m)$ ($q = 2^n$)

This section shows a fast algorithm for computing multiplicative inverses in $\text{GF}(q^m)$ ($q = 2^n$) using normal bases.

**Theorem 3.** Let $x$ be a non-zero element in $\text{GF}(q^m)$ ($q = 2^n$). Then, there exists an algorithm for computing $x^{-1}$, which requires

- number of multiplications in $\text{GF}(q^m)$:
  \[
  \text{NM}_1(m) = \lceil \log_2(m - 1) \rceil + H_\omega(m - 1),
  \]
- number of cyclic shifts over $\text{GF}(q)$: $\text{NS}_1(m) = m - 1$,
- number of multiplications in $\text{GF}(q)$ ($q = 2^n$):
  \[
  \text{NM}_2(n) = \lceil \log_2(n - 1) \rceil + H_\omega(n - 1) - 1,
  \]
- number of cyclic shifts over $\text{GF}(2)$: $\text{NS}_2(n) = n - 1$,

where $\lceil \cdot \rceil$ = Gauss' symbol and $H_\omega(\cdot)$ = Hamming weight.

**Proof.** For a non-zero element $x$ in $\text{GF}(q^m)$ ($q = 2^n$), $x^{-1}$ is given by $x^{-1} = x^{q^m - 2}$. Here $q^m - 2$ can be decomposed by

\[
q^m - 2 = (q - 2) \sum_{i=0}^{m-1} q^i + \sum_{j=1}^{m-1} q^j.
\] (19)
For the simplicity of the notation, define \( a = \sum_{i=0}^{m-1} q^i \) and \( b = \sum_{j=1}^{m-1} q^j \), and we have \( x^{-1} = y^{(p-2)}z \), where \( y = x^a \) and \( z = x^b \). Note that \( y = zx \). Applying the similar procedure in Section 4 (proof of Theorem 2), \( \text{NM}_1(m) \) (number of multiplications in \( \text{GF}(q^m) \)) and \( \text{NS}_1(m) \) (number of cyclic shifts over \( \text{GF}(q) \)) to compute \( z \) and \( y \) are

\[
\text{NM}_1(m) = \lfloor \log_2(m - 1) \rfloor + H_w(m - 1),
\]

\[
\text{NS}_1(m) = m - 1.
\]

Since \( y \) is norm of \( x \) (Lidl and Niederreiter, 1983), \( y \) is an element of \( \text{GF}(q) \). Hence, \( y^{q-2} = y^{-1} \), and by Theorem 2, \( \text{NM}_2(n) \) (number of multiplications in \( \text{GF}(q) \) \( (q = 2^n) \)) and \( \text{NS}_2(n) \) (number of cyclic shifts over \( \text{GF}(2) \)) to compute \( y^{-1} = y^{q-2} = y^{2^n-2} \) are

\[
\text{NM}_2(n) = \lfloor \log_2(n - 1) \rfloor + H_w(n - 1) - 1,
\]

\[
\text{NS}_2(n) = n - 1.
\]

### 7. Conclusions

A fast algorithm for computing multiplicative inverses in \( \text{GF}(2^m) \) using normal basis has been proposed. This algorithm requires at most \( 2 \lfloor \log_2(m - 1) \rfloor \) multiplications in \( \text{GF}(2^m) \) and \( (m - 1) \) cyclic shifts over \( \text{GF}(2) \) to compute multiplicative inverses, which are much less than those required in the Wang's method. The algorithm proposed in this paper computes multiplicative inverses in \( \text{GF}(2^m) \) by iterating multiplications in \( \text{GF}(2^m) \) and cyclic shifts over \( \text{GF}(2) \) in turn. Hence, the number of cyclic shifts is reduced to \( \lfloor \log_2(m - 1) \rfloor + H_w(m - 1) \), using special hardware which carries out \( k \) cyclic shifts over \( \text{GF}(2) \) in one machine cycle (Itoh and Tsujii, 1986). It has been also shown that the same idea of the proposed algorithm in \( \text{GF}(2^m) \) is applicable to the computation of multiplicative inverses in \( \text{GF}(q^m) \) \( (q = 2^n) \). It is clear that the computation of multiplicative inverses in \( \text{GF}(p^m) \) \( (p: \text{odd prime}) \) can be carried out similarly in the case of \( \text{GF}(q^m) \) \( (q = 2^n) \) (Itoh and Tsujii, 1986).

Furthermore, an idea similar to the proposed algorithm can be applied to general power operation in \( \text{GF}(2^m) \), which was pointed out by S. A. Vanstone (1987).

### Acknowledgments

The authors wish to thank Professor S. A. Vanstone of the University of Waterloo for his valuable discussions.

Received July 8, 1987; accepted January 12, 1988
REFERENCES


VANSTONE, S. A. (1987), Unpublished manuscript, lecture at NTT.