ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

Reduced-Latency Algorithm for Finite Field Inversion in GF(2^m)

Walid Mahmoud

Aassistant Professor, Communications and Networks Engineering, Prince Sultan University Riyadh, Kingdom of Saudi Arabia

Abstract: In this letter, we propose a novel reduced-latency finite field inversion algorithm for binary extension fields GF(2^m) using normal basis representation. A similar approach to that in Itoh-Tsujii inversion algorithm is used, however, the latency is significantly reduced for the time required to perform the necessary multiplications for inversion, which is a function of the binary length of the extension degree $(m-1)$ of the concerned field. The latency of our proposed finite field inversion algorithm is always comparable to the best case scenario in Itoh-Tsujii inversion algorithm for any given extension degree m, or equivalently, for any given GF(2^m).

Keywords: Finite field inversion, Fermat's little theorem, normal basis representation, binary extension fields GF(2^m).

I. INTRODUCTION

In binary extension fields GF(2^m) the elements are binary vectors of size m, the extension degree of the concerned field, with different interpretations depending on the used representation [1, 2]. In particular, targeting reduced-latency inversion architectures in such fields is of paramount importance to academic researchers in the literature [3, 4, 5].

Using Fermat's field inversion approach, given any nonzero element $\beta \in$ GF(2^m) with the extension degree m, its inverse is given by

$$\beta^{-1} = \beta^{2^m-1} = \beta^{2^1} \times \beta^{2^2} \times \ldots \times \beta^{2^{m-1}}, \quad (1)$$

that is computed using $(m-2)$ field multiplications by using square-multiply inversion algorithm, *which is the incurred latency in using such algorithm*. Note that $2^i - th$ powers are free execution-time operations in using normal basis representation. For example, given $m = 6$ and the element $\beta \in$ GF(2^6), its inverse β^{-1} is computed using 4 field multiplications as shown in Fig. 1.

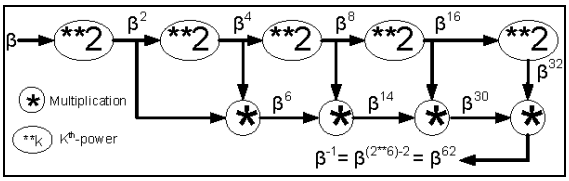

Fig. 1 Square-Multiply Inversion Algorithm

Itoh-Tsujii algorithm (ITA [6]) is also Fermat's-based, however, it uses another way to approach the inverse that is computed with $[\ell(m-1) + \omega(m-1) - 2]$ multiplications, *which is the incurred latency in using ITA algorithm*. Note that $\ell(m-1)$ is the binary length and $\omega(m-1)$ is the Hamming weight operators of the extension degree m subtracted by 1. For example, given $m = 12$ and the element $\beta \in$ GF(2^{12}), its inverse β^{-1} is calculated with 5 field multiplications as shown in Fig. 2.

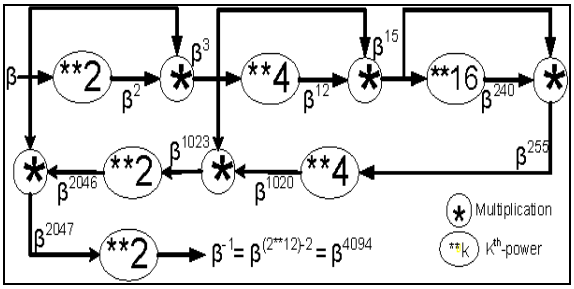

Fig. 2 ITA Inversion Algorithm

II. PROPOSE ALGORITHM

Our proposed reduced-latency field inversion algorithm is shown in Fig. 3. Certainly, when both *for* loops in the algorithm are executed in tandem, the inverse is obtained after the q^{th} iterate, which is equivalent to the execution time of $q = \ell(m-1)$ binary extension field multiplications. This value also represents the incurred latency in using our proposal.

The above is achieved by running two processes in tandem (*or two processors in hardware-mapping*), whereby the first process calculates the binary extension field multiplications that depend on $\ell(m-1)$ term, and the second process calculates the binary extension field multiplications that depend on $\omega(m-1)$ term, as evident from the proposed reduced-latency inversion architecture in Fig. 4.

Since both processes are running concurrently, this approach renders inverse calculation independent of the execution time required for performing the binary extension field multiplications that depend on $\omega(m-1)$ term for any extension degree m. Thus, the latency is significantly reduced to the execution time of $q = \ell(m-1)$ binary extension field multiplications. However, the total number of multiplications required for inversion is same that required by ITA algorithm.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

Input: a nonzero $\beta \in GF(2^m)$, the extension degree m
Output: $b^2 = \beta^{-1} \in GF(2^m)$
Initial: Represent $(m-1)$ as $(1m_{q-2}...m_1m_0)_2$, $\delta = \beta$
Initial: $t = [t_0t_1...t_{q-2}]$, if $m_0 = 0 : b = 1$, else : $b = \beta$
for $i = 0$ to $q-2$ do
  $\delta = \delta \times \delta^{2^{2^i}}$
  $t_i = \delta$
end for
for $i = 1$ to $q-1$ do
  if $t_{i-1} \neq 0$ then
    if $m_i = 1$ then
      $a = (t_{i-1})^{2^{\sum_{j=0}^{i-1} m_j 2^j}}$
    else
      $a = 1$
    end if
  end if
  $b = b \times a$
end for
return b^2

Fig. 3 Reduced-Latency Inversion in GF(2^m)

For example, given $m = 16$, the inverse of a nonzero element $\beta \in$ GF(2^{16}) is computed using our proposed reduced-latency architecture in GF(2^m) as shown in Fig. 5.

From Fig. 5, despite the use of 6 multipliers, the latency is equivalent to the execution time of 4 binary extension field multiplications, i.e., $q = \ell(m-1) = \ell(15)$=4. This is because multipliers pointed to by the curved-lines (2, 3) are running concurrently. For this specific example, the latency of ITA algorithm is equivalent to the execution time of 6 binary extension field multiplications.

Our proposed field inversion algorithm achieve its utmost competitive advantage in comparison with ITA algorithm for the extension degrees m, or equivalently in the binary extension fields GF(2^m), in which $(m-1) = 2^k - 1$ for any positive integer k (i.e., $m = 2^k$). Such m values represent the worst cases in using ITA algorithm. This is because the number of binary extension field multiplications those necessary for inversion and the corresponding latency is maximized, and is given by the value $(2k-2)$ for the given k as above.

III. CONCLUSIONS

In this letter, we proposed a novel reduced-latency field inversion algorithm, along with its architecture, for binary extension fields GF(2^m) using normal basis representation for the field elements. However, it can be easily modified for use with other extension fields and representation bases.

The followed inversion approach is similar to that used in Itoh-Tsujii inversion algorithm, however, the latency is significantly reduced to the execution time of a number of multiplications equal to the binary length of the extension degree $(m-1)$ of the concerned GF(2^m). Unlike the case in Itoh-Tsujii inversion algorithm, the latency in our algorithm is fixed and it is independent of the Hamming weight in $(m-1)$ of the concerned binary extension field.

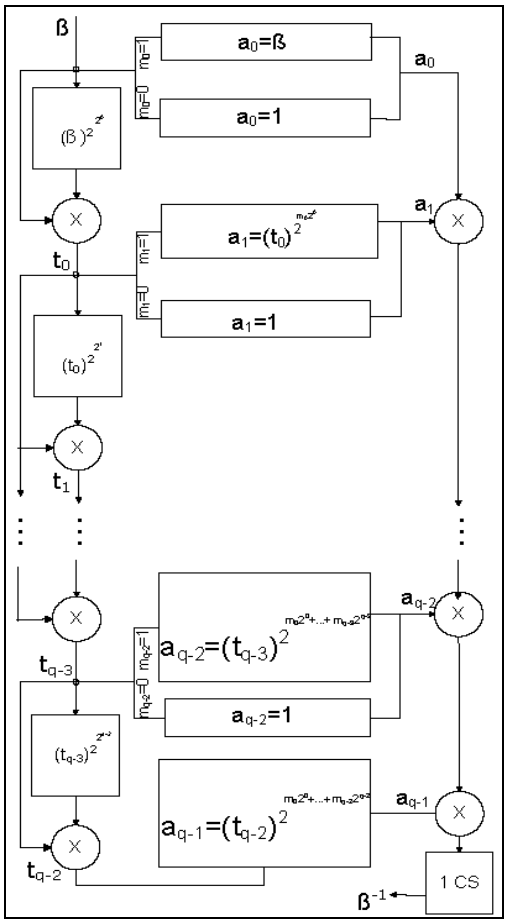

Fig. 4 Proposed - Inversion Architecture

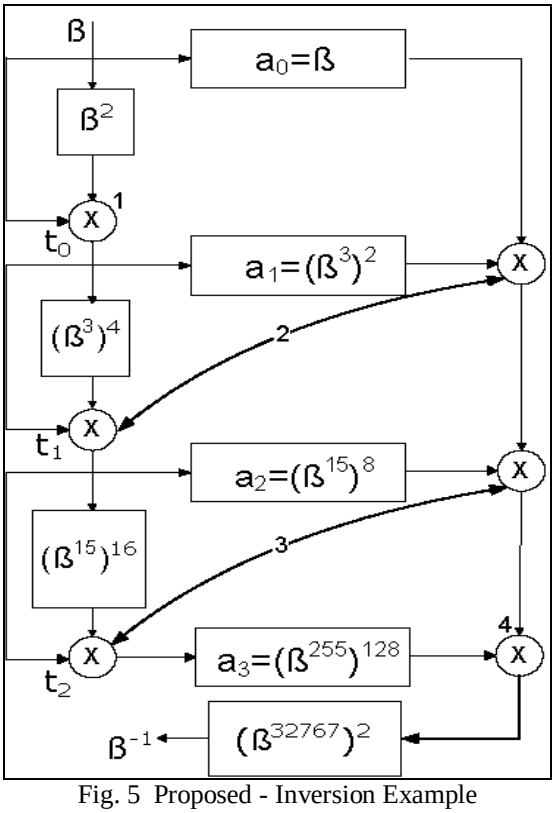

Fig. 5 Proposed - Inversion Example

ACKNOWLEDGMENT

The author would like to warmly thank peer reviewers for their help in reviewing this manuscript.

REFERENCES

[1] I. P1363/D1, Standard specifications for public-key cryptography, draft version 1st~ed. ``http://grouper.ieee.org/groups/1363/'': IEEE standards documents, Nov. 2009.

[2] D. Hankerson, J. Menezes and S. Vanstone, Guide to Elliptic Curve Cryptography, New York, USA: Springer-Verlag Inc., 2004.

[3] A. Dinh, R. Bolton and R. Mason, "A low latency architecture for computing multiplicative inverses and divisions in GF(2^m)," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, issue 8, pp. 789-793, Aug. 2001.

[4] M. Jing, J. Chen, Z. Chen and Y. Chen, "Low Complexity Architecture for Multiplicative Inversion in GF(2^m)," IEEE Asia Pacific Conference on Circuits and Systems, pp. 1492-1495, Dec. 2006.

[5] Q. Deng, X. Bai, L. Guo and Y. Wang, "A fast hardware implementation of multiplicative inversion in GF(2^m)," Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, pp. 472-475, Jan. 2009.

[6] T. Itoh and S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverses in GF(2^m) Using Normal Basis," Information and Computation Journal, vol. 78, pp. 171-177, 1988.