GF(p^m) Multiplication Using Polynomial Residue Number Systems

Abstract

GF(p^m) multiplication is computed in two stages. Firstly, the polynomial product is computed modulus a highly factorisable degree S polynomial, $M(x)$, with $S \geq 2.m - 1$. This enables the product to be computed using a Polynomial Residue Number System (PRNS). Secondly, the result is reduced by the irreducible polynomial, $I(x)$, over which GF(p^m) is defined. Suitable choices for S, $M(x)$ and $I(x)$ are discussed and an iterative method for the factorisation of $x^T - k$ polynomials, $k \in$ GF(p), is presented. Finally, multi-dimensional PRNS is proposed to solve the upper limit constraint on m, which is dependent on p.

1 Introduction

GF(p^m) multipliers are required in a number of applications. For instance, to realise cyclic convolutions using GF(p^m) spectra [3, 4] and for error-correction [4, 5], cryptographic, and multi-valued logic (MVL) systems. The three conventional ways to perform GF(p^m) multiplication use standard basis [10], normal basis [6], or dual basis multipliers [7]. Dual basis solutions are most area-efficient, but all three methods require at least m^2 general multiplications, mod p. There is also the need for continual reduction by an irregular polynomial modulus, requiring some fixed multiplications and additions, mod p. This reduction will occur concurrently or interleaved, causing increased latency, decreased throughput and impaired design symmetry. In contrast, Polynomial Residue Number Systems (PRNS) [9] decompose polynomial products (PPs) into a small number of autonomous products, mod p, performed in parallel without any irregular reduction by a polynomial modulus. There is the added task of conversion to and from the polynomial residue domain, but for special PRNS this simplifies to the complexity of efficient Number Theoretic Transforms (NTTs), mod p [1]. It will be shown how this PRNS technique can be applied to the GF(p^m) multiplier. The independence of the small products introduces a capacity for fault-tolerance not evident in conventional GF(p^m) schemes [14] and enables highly-parallel, high-speed implementations.

Extended Galois Fields, GF(p^m), are defined over irreducible polynomials, $I(x)$, of order m, and elements in GF(p^m) are computed as PPs, mod $I(x)$ over GF(p). If, instead, the GF(p^m) multiplication is performed mod $M(x)$, where $M(x)$ is factorisable and of a suitably high degree, a PRNS decomposition can be used [9]. A final reduction, mod $I(x)$, realises the GF(p^m) product. This paper highlights suitable choices for $M(x)$ and $I(x)$. Firstly, a method to determine the factorisation of $x^T - k$, $k \in$ GF(p), over GF(p), is outlined. This method will aid in choosing $M(x)$ and $I(x)$. The GF(p^m) multiplier is then described as a combination of PRNS and modular reduction, using suitable $I(x)$ and $M(x)$. Finally, as

$M(x)$ is only fully factorisable over GF(p) into distinct degree-one factors (FFD) if m is below an upper limit determined by p, this 'upper limit' on m is extended using multi-dimensional PRNS.

2 Factorising $x^T - k$, $k \in$ GF(p), Over GF(p)

This section describes a way to determine the degrees of the distinct factors of $x^T - k$, $k \in$ GF(p).
It can be verified that d_1 distinct roots of k exist in GF(p) which zero $G(x) = x^T - k$, where,

$$d_1 = \gcd(T, p-1) \quad \text{if} \quad d_1 | ((p-1)/O_p(k)), \qquad d_1 = 0 \quad \text{otherwise} \tag{1}$$

where $O_p(k)$, the "order of k over GF(p)", implies $k^{O_p(k)} = 1$ over GF(p), with $k^w \neq 1$ for $w < O_p(k)$.

$$\text{Thus} \qquad G(x) = x^T - k = (x - r_{1,1}).(x - r_{2,1}).\ldots.(x - r_{d_1,1}).G_1(x) \tag{2}$$

where $r_{i,1} \in$ GF(p) are zeroes of $x^T - k$, and $d(G_1(x)) = T - d_1$, ($d(*)$ means "the degree of $*$").
Similarly, $G(x)$ contains d_2 distinct roots of k in GF(p^2), where,

$$d_2 = \gcd(T, p^2-1) \quad \text{if} \quad d_2 | ((p^2-1)/O_p(k)), \qquad d_2 = 0 \quad \text{otherwise} \tag{3}$$

These d_2 roots include the d_1 roots over GF(p). Thus,

$$G(x) = x^T - k = (x - r_{1,1}).(x - r_{2,1}).\ldots.(x - r_{d_1,1}).(x - r'_{1,1}).(x - r'_{2,1}).\ldots.(x - r'_{d_2-d_1,1}).G_2(x) \tag{4}$$

where $r_{i,1} \in$ GF(p), $r'_{i,1} \in$ GF(p^2) $\notin$ GF(p), and $d(G_2(x)) = T - d_2$. The roots, $r'_{i,1}$, are always from length-2 conjugate sets, and can be combined in pairs to form $(d_2 - d_1)/2$ quadratics. Thus,

$$G(x) = x^T - k = (x - r_{1,1}).(x - r_{2,1}).\ldots.(x - r_{d_1,1}).(x^2 - r_{1,2}).(x^2 - r_{2,2}).\ldots.(x^2 - r_{(d_2-d_1)/2,2}).G_2(x) \tag{5}$$

where the $r_{i,2}$ are polynomials in x with coefficients $\in$ GF(p), and $d(r_{i,2}) \leq 1$.
Consider factorisation over GF(p^3), where the new roots are combined to form $(d_3 - d_1)/3$ cubics,

$$G(x) = x^T - k = \prod_{i=1}^{d_1}(x - r_{i,1}). \prod_{i=1}^{(d_2-d_1)/2}(x^2 - r_{i,2}). \prod_{i=1}^{(d_3-d_1)/3}(x^3 - r_{i,3}).G_3(x) \tag{6}$$

The $r_{i,3}$ are quadratics in x, with coefficients $\in$ GF(p), $d(r_{i,3}) \leq 2$, and $d(G_3(x)) = T - d_2 - d_3 + d_1$. The d_3 roots contain the d_1 roots but not the d_2 roots, as GF(p^3) is an extension of GF(p), not of GF(p^2).
In general, d_t distinct T^{th}roots of k exist in GF(p^t), where,

$$d_t = \gcd(T, p^t - 1) \quad \text{if} \quad d_t | ((p^t - 1)/O_p(k)), \qquad d_t = 0 \quad \text{otherwise} \tag{7}$$

By computing d_t from $t = 1$ up to $t = T$,

$$G(x) = x^T - k = \prod_{t=1}^{T}\left(\prod_{i=1}^{d'_t/t}(x^t - r_{i,t})\right).G_T(x) \quad \text{where } d'_t = d_t - \sum_{i|t, i<t} d'_i, \qquad \prod_{i=1}^{0} = 1 \tag{8}$$

and $d(r_{i,t}) \leq t - 1$. There may be less than T distinct zeroes of $x^T - k$ in GF extensions $1 \leq t \leq T$. (8) specifies the distinct factorisation of $x^T - k$, $k \in$ GF(p), over GF(p). (Only the degrees of the factors

have been determined, not the coefficients). In particular,

$$\begin{gathered} G(x) \text{ is fully factorisable into distinct degree-one factors (FFD) over GF}(p), \text{ iff} \\ d_1 = T \qquad \Rightarrow \qquad T|((p-1)/O_p(k)). \end{gathered} \tag{9}$$

$$G(x) \text{ is irreducible over GF}(p) \text{ iff } d_t = 0 \text{ for } 1 \leq t < T. \tag{10}$$

3 GF(p^m) Multiplier

Consider the GF(p^m) multiplication,

$$C'(x) = \langle A(x).B(x) \rangle_{I(x)} = \langle C(x) \rangle_{I(x)} \tag{11}$$

$I(x)$ is irreducible over GF(p), $d(I(x)) = m$, and $d(A(x), B(x)) < m$. This product can be performed using a PRNS stage followed by a Reduction stage:

3.1 PRNS Stage

As $d(C(x)) < 2.m-1$, $C(x)$ can be embedded in a polynomial ring, mod $M(x)$, where $d(M(x)) \geq 2.m-1$, without requiring any reduction of $C(x)$, mod $M(x)$. Thus,

$$C(x) = A(x).B(x) = \langle A(x).B(x) \rangle_{M(x)} \tag{12}$$

where $d(M(x)) = S \geq 2.m-1$.
$M(x)$ will be chosen to factorise over GF(p) as follows,

$$M(x) = \prod_{i=0}^{n-1} q_i(x) \tag{13}$$

where $q_i(x) \not| q_j(x) \quad i \neq j$, i.e. the $q_i(x)$ factors are mutually prime, mod p. Using PRNS techniques,

$$C(x) = \text{CRTP}(\langle C(x) \rangle_{q_0(x)}, \langle C(x) \rangle_{q_1(x)}, \ldots, \langle C(x) \rangle_{q_{n-1}(x)}) \tag{14}$$

where the residue products, $\langle A(x).B(x) \rangle_{q_i(x)}$, are computed independently, and 'CRTP' implies 'the Chinese Remainder Theorem for Polynomials' [9]. $M(x)$ is FFD over GF(p) if $n = S$, with $d(q_i(x)) = 1 \ \forall \ i$. A FFD $M(x)$ is particularly useful as the residue products occur over GF(p). As p is prime, there are $p-1$ mutually prime degree-one polynomials over GF(p). Therefore,

$$\text{An M(x) which is FFD over GF}(p) \text{ exists } \underline{\text{only}} \text{ if} \qquad 1 \leq S = d(M(x)) \leq p-1. \tag{15}$$

Also $S \geq 2.m-1$ to avoid reduction of $A(x).B(x)$, mod $M(x)$. Thus, remembering that p is prime,

$$\text{A FFD PRNS implementation of GF}(p^m) \text{ multiplication is only feasible if } m \leq (p-1)/2. \tag{16}$$

Solutions for $m > (p-1)/2$ will be examined later. For systems that satisfy (16), a choice of FFD $M(x)$ is evident, where $2.m-1 \leq S \leq p-1$. The $q_i(x)$ are of the form $(x - j_i)$, where $j_i \in \{1, 2, \ldots, p-1\}$

and $j_i \neq j_l$ for $i \neq l$. Often in the literature [3, 11, 12], $M(x)$ is chosen as $x^S \pm 1$, where $M(x)$ is FFD, to implement cyclic, or skew-cyclic convolutions. This allows the use of the Agarwal-Cooley Cyclic Convolution (CC) algorithm, [3], where the factors of S are mutually prime, to decompose the system into smaller CC PRNS. Alternatively, one can use the Generalised Number Theoretic Transform to realise the PRNS [2]. To obtain FFD $M(x)$ of the form $M(x) = x^S \pm 1$, and, more generally, $x^S - \alpha$, $\alpha \in$ GF(p), S is further restricted as stated in (9), where $S = T$. From (9), S may be substantially larger than $2.m - 1$. Also, S may not possess many, if any, prime factors. In such cases, solutions with $M(x)$ other than $x^S - \alpha$ may be preferable.

3.2 Reduction Stage

The second stage of the multiplier reduces $C(x)$ by $I(x)$ to obtain the GF(p^m) product, $C'(x)$. Hence,

$$C'(x) = \langle C(x) \rangle_{I(x)} \qquad \text{where } d(C(x)) < S \text{ and } d(C'(x)) < m \tag{17}$$

The complexity of this reduction depends on the form of $I(x)$. Let us define,

$$I(x) = x^m - k \qquad \text{where } k \in \text{GF}(p) \tag{18}$$

The reduction, mod $I(x)$, requires only $S - m$ fixed multiplications and additions, mod p. To achieve this simplified reduction, an $I(x) = x^m - k$, $k \in$ GF(p), must be found which is irreducible over GF(p). $I(x)$ polynomials for GF(p^m) multipliers can be determined using (8) and (10), and are shown in Table 1 (dimension 1). $I(x)$ are not always found, for example, no irreducible polynomial of the form $x^m - k$, $k \in$ GF(p), exists for $p = 11$ and $m = 4$.

4 GF(p^m) Multipliers where $m > (p-1)/2$

From (16), if $m > (p-1)/2$, FFD $M(x)$, with $S \geq 2.m-1$, do not exist. [11, 12] present a 2-dimensional (2-D) decomposition of a 1-D PP, mod $x^S \pm 1$, to overcome the $m > (p-1)/2$ constraint. These techniques will be applied to the GF(p^m) multiplier and then generalised to the multi-dimensional case.

4.1 Two-Dimensional PRNS

It will be shown that an $m-1$ degree PP can always be expressed as a 2-D PP where the first and second dimensions compute $m_1 - 1$ and $m_2 - 1$ degree PPs, respectively, with $m_1.m_2 \geq m$. Let,

$$A(x) = \sum_{i=0}^{m-1} a_i.x^i \quad \text{and} \quad B(x) = \sum_{i=0}^{m-1} b_i.x^i \tag{19}$$

By choosing $m_1 < m$, where $m_2 = \lceil (m-1)/m_1 \rceil$, then,

$$m_1.m_2 \geq m \tag{20}$$

p	m	possible k	Dimension	m_1	m_2	m_3
5	2	2,3	1	2	-	-
5	3	-	-	-	-	-
5	4	2,3	2	2	2	-
5	5	-	-	-	-	-
5	6	-	-	-	-	-
5	7	-	-	-	-	-
5	8	2,3	3	2	2	2
7	2	3,5,6	1	2	-	-
7	3	2,3,4,5	1	3	-	-
7	4	-	-	-	-	-
7	5	-	-	-	-	-
7	6	3,5	2	2	3	-
11	2	2,6,7,8,10	1	2	-	-
11	3	-	-	-	-	-
11	4	-	-	-	-	-
11	5	2,3,4,5,6,7,8,9	1	5	-	-
13	2	2,5,6,7,8,11	1	2	-	-
13	3	2,3,4,6,7,9,10,11	1	3	-	-
13	4	2,5,6,7,8,11	1	4	-	-
13	4	”	2	2	2	-
29	2	2,3,8,10,11,12,14,15,17,18,19,21,26,27	1	2	-	-
29	3	-	-	-	-	-
29	4	2,3,8,10,11,12,14,15,17,18,19,21,26,27	1	4	-	-
61	2	2,6,7,8,10,11,17,18,21,23,24,26,28,29,30, 31,32,33,35,37,38,40,43,44,50,51,53,54,55,59	1	2	-	-

Table 1: Irreducible Polynomials, $(I(x))$, for use in N-D PRNS GF(p^m) General Multipliers

By assigning $y = x^{m_1}$,

$$A(x) = A(x,y) = \sum_{i_1=0}^{m_1-1} \left(\sum_{i_2=0}^{m_2-1} a_{i_1+m_1.i_2}.y^{i_2} \right).x^{i_1} = \sum_{i_1=0}^{m_1-1} A_{i_1}(y).x^{i_1} \tag{21}$$

$B(x,y)$ is similarly defined. Therefore,

$$C(x) = C(x,y) = \sum_{i_1=0}^{m_1-1} A_{i_1}(y).x^{i_1}. \sum_{i_1=0}^{m_1-1} B_{i_1}(y).x^{i_1}$$

which is a degree $m_1 - 1$ PP in x with y polynomial coefficients. $M_1(x)$ is then chosen so that,

$$C(x,y) = \langle C(x,y) \rangle_{M_1(x)} \tag{22}$$

where $S_1 = d(M_1(x)) \geq 2.m_1 - 1$.

If $M_1(x)$ is FFD over GF(p), the PP in x can be performed using a PRNS, with all operations except residue products, performed over GF(p). The residue products are defined as follows,

$$C_j^*(y) = A_j^*(y).B_j^*(y) \qquad 0 \leq j < S_1 \tag{23}$$

where $Z_j^*(y)$ implies the residue of $Z(x,y)$, mod $q_j(x)$, and $M_1(x) = \prod_{j=0}^{s-1} q_j(x)$. These are degree $m_2 - 1$ PPs in y, with GF(p) coefficients. $M_2(y)$ is then chosen so that,

$$C_j^*(y) = \left\langle C_j^*(y) \right\rangle_{M_2(y)} \tag{24}$$

where $S_2 = d(M_2(y)) \geq 2.m_2 - 1$.

If $M_2(y)$ is FFD over GF(p), the PP in y can be performed using a PRNS, with all residue products performed over GF(p). Therefore the PP, $C(x) = A(x).B(x)$, can be embedded in two, nested PRNS,

$$C(x) = C(x,y) = \Big\langle \langle A(x,y).B(x,y) \rangle_{M_1(x)} \Big\rangle_{M_2(y)} \tag{25}$$

and, from (11),

$$C'(x) = \Big\langle \Big\langle \langle A(x,y).B(x,y) \rangle_{M_1(x)} \Big\rangle_{M_2(y)} \Big\rangle_{I(x)} \tag{26}$$

All operations in the computation of $C(x)$ occur in GF(p) iff $M_1(x)$ and $M_2(y)$ are FFD over GF(p). This is true if S_1 and $S_2 \leq p-1$. As $S_1 \geq 2.m_1 - 1$ and $S_2 \geq 2.m_2 - 1$,

$$m_1, m_2 \leq (p-1)/2 \tag{27}$$

and, combining (20) and (27),

$$1 \leq m \leq (p-1)^2/4 \tag{28}$$

Thus, a 2-D PRNS GF(p^m) multiplier, using only GF(p) products, can be implemented, for $1 \leq m \leq (p-1)^2/4$, (as long as $x^m - k$ is irreducible over GF(p), $k \in$ GF(p)).

4.2 N-Dimensional PRNS

The FFD PRNS technique can be generalised to the N-D case, given,

$$m_1.m_2.\ldots.m_N \geq m \tag{29}$$

$$m_i \leq (p-1)/2 \qquad \text{for } 1 \leq i \leq N \tag{30}$$

enabling multiplication over GF(p^m) for,

$$1 \leq m \leq [(p-1)/2]^N \tag{31}$$

Consider multiplication over GF(5^8). The minimum value of N, for which (31) is satisfied is 3. Thus, FFD, N-D, PRNS can be used to implement a GF(5^8) multiplier, where $N \geq 3$. For instance, (29) and (30) are satisfied by choosing $m_1 = m_2 = m_3 = 2$. Moreover, $I(x) = x^8 - 3$ is irreducible over GF(5) so the final $I(x)$ reduction is simplified. Finally, (30) specifies a lower limit on p for which the method of this paper is feasible, as, for $p = 2$ or 3, the m_i cannot be > 1, rendering a reduction in m impossible. Table 1 shows examples where N-D PRNS is necessary (dimension > 1).

5 GF(p^m) Multiplication Using GF(p^{m_2}) Operations, $m_2|m$

This section notes a subset of the N-Dimensional PRNS of Section 4. With reference to (20) and (21), if $m_1.m_2 = m$, the polynomials in y, $A_{i_1}(y)$ and $B_{i_1}(y)$, are elements of the sub-field, GF(p^{m_2}), and GF(p^{m_2}) is defined over $I_2(y)$.

$$\text{if} \quad I(x) = x^m - k \quad \text{then} \quad I_2(y) = y^{m_2} - k \qquad I(x), I_2(y) \text{ irreducible over GF}(p) \tag{32}$$

(22) can be interpreted as a PRNS in x, over $M_1(x)$, where all coefficients are $\in$ GF(p^{m_2}). The subsequent PRNS in y is not performed. This change in base field allows the GF(p^m) multiplier to be implemented using GF(p^{m_2}) hardware. In [13], linear convolution of complex data is performed by embedding data in a polynomial ring. For the GF(p^m) multiplier of this paper, a similar scenario occurs when $m_2 = 2$ and $M_1(x) = x^{m_1.m_2} - k = \mathrm{I(x)}$. GF($p^2$) is suitable for complex arithmetic but, unlike [13], it is not QRNS, as $I_2(y)$ is irreducible. Similarly, $M_1(x) = I(x)$ is not suitable for PRNS as it, too, is irreducible. Although GF(p^m) computation contains sub-field linear convolution, it is not amenable to the solution of [13].

6 Prime-Factor Multi-Dimensional Techniques

Another multi-dimensional decomposition exists. The Agarwal-Cooley algorithm decomposes a 1-D PP, mod $x^S - \alpha$, into a N-D PP, mod $x^{S_1} - \alpha$, mod $x^{S_2} - \alpha$, . . etc. (The FFD modulus must be of the form $x^S - \alpha$, $\alpha \in$ GF(p), $S \geq m$, for this method to work). For the 2-D case,

$$y = x^{m_1} \qquad z = x^{m_2} \qquad \text{where } m_1.m_2 \geq m, \text{ and } \gcd(m_1, m_2) = 1 \tag{33}$$

The PP is decomposed, mod $x^{m_1.m_2} - \alpha$, into two nested PPs, mod $z^{m_1} - \alpha$ and $y^{m_2} - \alpha$, respectively. However, this technique is of <u>no</u> benefit for cases where $m > (p-1)/2$ as, from (9), $x^{m_1.m_2} - \alpha$ is only FFD if $m_1.m_2 | p-1$. If $m > (p-1)/2$, then $m_1.m_2 \not| p-1$. This suggests a hybrid solution, where the N-Dimensional method of Section 4 reduces the problem so that all $m_i \leq (p-1)/2$. Prime-factor techniques may then be preferable for further decomposition.

7 A Comparison with Conventional GF(p^m) Multipliers

In the following, only 1-D PRNS is assessed, assuming the PRNS is defined over a FFD $M(x)$, mod p, where conversion to the PRNS requires two S-point NTTs, mod p, (for $A(x)$ and $B(x)$) and, for conversion from the PRNS to $C(x)$, one S-point Inverse NTT, mod p. Also, S residue products, mod p, are required, and for the final reduction by $I(x)$, $S - m$ fixed mults and adds, mod p. Each NTT normally requires S^2 fixed mults and adds, mod p but, for efficient NTTs [15], this figure approaches the equivalent of S general mults, mod p. Using this figure, the total operation count is,

$$4.S \text{ general mults, mod } p, \qquad S - m \text{ fixed mults, mod } p, \qquad \text{and} \qquad S - m \text{ adds, mod } p$$

In comparison, an 'optimal' dual-basis multiplier, defined over an irreducible trinomial [7], requires,

$$m^2 \text{ general mults, mod } p, \qquad 2.m \text{ fixed mults, mod } p, \qquad \text{and} \qquad m^2 \text{ adds, mod } p$$

Comparing general mults only, the PRNS design becomes competitive when $4.S \leq m^2$. Assuming $S = 2.m - 1$, this requires $m \geq 8$. The PRNS has the equivalent of a general multiplier count less than $O(m^2)$, as opposed to the dual-basis multiplier, which requires $O(m^2)$ general multipliers, and one can

expect the PRNS to compete for large m and p. The elimination of interleaved modular reduction gives the PRNS multiplier a lower latency and higher throughput potential than conventional solutions and the residue form allows the incorporation of redundant-residue-based fault-tolerance [8, 14]. One can evisage a Reed-Solomon (RS) encoder/decoder over GF(p^m), where the inherent PRNS-based GF(p^m) multipliers are, themselves, protected by a shorter-length RS code over GF(p).

8 Conclusion

Multiplication over GF(p^m) has been implemented using PRNS, followed by a final reduction by the irreducible polynomial, $I(x)$, over which GF(p^m) is defined. It is shown how, by appropriate choice of PRNS and $I(x)$, all operations occur over GF(p), and reduction by an irregular polynomial modulus is eliminated. In comparison to conventional solutions, potential throughput is enhanced, and a high-speed VLSI implementation is possible which is symmetric, parallel and fault-tolerant. For large m one can also expect a reduction in area. An iterative method for finding suitable $M(x)$ and $I(x)$, of the form $x^T - k$, $k \in$ GF(p) has also been presented. Solutions for large m and small p are possible using multi-dimensional PRNS, and an equation evaluating the minimum PRNS dimensionsionality is given. It is noted that prime-factor PRNS may be beneficial for $m \leq (p-1)/2$ but not applicable for $m > (p-1)/2$. The multiplier can be incorporated in spectral-based cyclic convolvers, and also has application to error-correction and multi-valued logic-based systems.

References

[1] J.H.McClellan,C.M.Rader, **Number Theory in Digital Signal Processing,** Prentice Hall, '79

[2] J-B.Martens, "Polynomial Products by Means of Generalized Number Theoretic Transforms", *IEEE Trans on Acoustics, Speech and Signal Processing*, Vol 32, No 3, pp 668-670, June '84

[3] R.E.Blahut, **Fast Algorithms for Digital Signal Processing,** Reading, Addison-Wesley, '85

[4] R.E.Blahut, "Algebraic Fields, Signal Processing, and Error Control", *Proc of IEEE*, Vol 73, No 5, pp 874 - 893, May '85

[5] R.Lidl,H.Niederreiter, **Introduction to Finite Fields and their Applications,** Cambridge Univ Press, '86

[6] A.Pincin, "A New Algorithm for Multiplication in Finite Fields", *IEEE Trans on Computers*, Vol 38, No 7, pp 1045 - 1049, July '89

[7] M.Wang,I.F.Blake, "Bit Serial Multiplication in Finite Fields," *SIAM J. Disc. Math*, Vol 3, No 1, pp 140 - 148, Feb '90

[8] A.Shiozaki,T.K.Truong,K.M.Cheng,I.S.Reed, "Fast Transform Decoding of Nonsystematic Reed-Solomon Codes", *IEE Proc-E*, Vol 137, No 2, pp 139 - 143, March '90

[9] A.Skavantzos,F.J.Taylor, "On the Polynomial Residue Number System", *IEEE Trans on Signal Processing*, Vol 39, No 2, pp 376-382, Feb '91

[10] C-L Wang,J-L Lin, "Systolic Array Implementation of Multipliers for Finite Fields GF(2^m)", *IEEE Trans on Circuits and Systems*, Vol 38, No 7, pp 796-800, July '91

[11] A.Skavantzos,N.Mitash, "Computing Large Polynomial Products using Modular Arithmetic," *IEEE Trans on Circuits and Systems - II*, Vol 39, No 4, pp 252 - 254, April '92

[12] A.Skavantzos,N.Mitash, "Implementation Issues of 2-Dimensional Polynomial Multipliers for Signal Processing Using Residue Arithmetic," *IEE Proc-E*, Vol 140, No 1, pp 45 - 53, Jan '93

[13] A.Skavantzos,T.Stouraitis, "Polynomial Residue Complex Signal Processing," *IEEE Trans on Circuits and Systems - II*, Vol 40, No 5, pp 342 - 344, May '93

[14] M.G.Parker,M.Benaissa, "Fault-Tolerant Linear Convolution using Residue Number Systems," *Proc of ISCAS '94, London*, Vol 2, pp 441 - 445, May '94

[15] M.G.Parker,M.Benaissa, "VLSI Structures for Bit Serial Modular Multiplication Using Basis Conversion," *IEE Proc-Comput.Digit.Tech*, Vol 141,No 6,pp 381-390, Nov '94