A Study on Constructing Highly Adder/multiplier Systems over Galois Felds

Chun-Myoung Park
Department of Computer Engineering, Chung-Ju National University
123 Bunji Kumdan-Ri Iryu-Myun Chungju Chungbuk 380-702 Korea
Tel: +82-441-841-5346, Fax: +82-441-841-5340
E-mail: cmpark@gukwon.chungju.ac.kr

Abstract : This paper propose the method of constructing the highly efficiency adder and multiplier systems over finite fie2 degree of

α^k terms, therefore we decrease k into m-1 degree using irreducible primitive polynomial. We propose two method of control signal generation for perform above decrease process. One method is the combinational logic expression and the other method is universal signal generation.

The proposed method of constructing the highly adder/multiplier systems is as following. First of all, we obtain algorithms for addition and multiplication arithmetic operation based on the mathematical properties over finite fields, next we construct basic cell of A-cell and M-cell using T-gate and modP cyclic gate. Finally we construct adder module and multiplier module over finite fields after synthesize α^k generation module and control signal CSt generation module with A-cell and M-cell.

Then, we propose the future research and prospects.

1. Introduction

In many area of digital logic systems and computer application, the arithmetic operation is important role. [1,2] Specially, in modern time, the multimedia and its application fields necessary to complex arithmetic operation and massive data manipulation.

Therefore highly efficiency arithmetic operation and its systems are researched in previous time. In specially, the arithmetic operation is effective analyzed in finite fields or galois fields. The galois filds is used to the mathematical background for encryption/decryption; error correcting code, digital image processing, digital signal processing, switching function of digital logic systems etc.

The following is the previous researches of arithmetic operation and its hardware implementation.

C.C.Wang[3] constructed multiplier over GF(2^m) based on the self-dual normal basis, and C.Ling Wang etc.[4] constructed parallel-in-parallel-out systolic array type multiplier based on normal basis. Also, S.T.J.Fenn etc.[5] Constructed the multiplier based on the dual basis and K.Z.Pekmastzi[6] propose multiplxer-based array multiplier. And G.Drolet[7] propose the small complexity arithmetic circuits.

This paper's construction is as following. Section2 discuss the important mathematical properties of galois fields and section3 discuss construct the adder module over galois fields that imply addition algorithm, basic A-cell. Section4 discuss the multiplier module over galois fields that imply multiplication algorithm, basic M-cell, α^k generation module, control signal CSt generation module, universal control signal CSt generation module. In section5, we summary the proposed highly adder/multiplier over galois fields, and we compare proposed method with earlier method.

Also we prospect future demand research and prospect

2. Mathematical Properties of Galois Fields

In this section, we review the important mathematical properties over galois fields, these mathematical properties used in build up this paper.

Any other mathematical properties except these mathematical properties refer to references.[8,9]

2.1 Finite Fields

Finite fields is defined by any prime number P and integer m, namely galois fields GF(P^m). In generally finite fields is organized by 5-tuple {S,+,•,0,1}, where S is set of elements, + and • are binary operation over S, 0 and 1 are each identity element for addition and multiplication arithmetic operation. Also finite fields are classified into ground fields GF(P) and extension fields GF(P^m). The number of elements over ground fields GF(P), P is the prime number more than 1, are {0,1,2,.....,P-1}.

2.2 Important mathematical properties

The important mathematical properties over galois fields are as following.

<P1> Commutative law
(1) a+b=b+a (2) a· b=b· a ($\forall$ a,b$\in$GF(P^m))
<P2> Associative law
(1) a+(b+c)=(a+b)+c (2) a· (b· c)=(a· b)· c
($\forall$ a,b,c$\in$GF(P^m))
<P3> Distributive law
a· (b+c)= a· b+a· c ($\forall$ a,b,c$\in$GF(P^m))
<P4> Zero element **0** exist.
a+**0**=**0**+a=a ($\forall$ a$\in$GF(P^m))
<P5> Unit element **1** exist.
a· **1**=**1**· a=a ($\forall$ a$\in$GF(P^m))
<P6> Inverse element exist.
(1) additive inverse element.
a+(**-a**)=0
(2) multiplicative inverse element .
a· ($\mathbf{a^{-1}}$)=1 ($\forall$ **-a** , $\mathbf{a^{-1}}\in$GF(P^m))
<P7> 0· a=a· 0=0 ($\forall$ a$\in$GF(P^m)).

3. Adder

3.1 Addition Algorithm

We put any two element over $GF(P^m)$, $F(\alpha)=\sum_{i=0}^{m-1} a_i\alpha^i$ and $G(\alpha)=\sum_{j=0}^{m-1} b_j\alpha^j$, and $A(\alpha)=\sum_{k=0}^{m-1} A_k\alpha^k$ that is the element after add them. Then we represent relationship among these elements as following.

$$F(\alpha)+G(\alpha)=\sum_{i=0}^{m-1} a_i\alpha^i+\sum_{j=0}^{m-1} b_j\alpha^j=\sum_{i,j=0}^{m-1}(a_i+b_j)\alpha^i$$

$$=\sum_{k=0}^{m-1} A_k\alpha^k=A(\alpha) \qquad (3\text{-}1)$$

where, $a_i, b_j, A_k \in GF(P)=\{0,1,\ldots,P-1\}(i,j,k=0,1,\ldots,m-1)$, $A_k=a_i+b_j$, Σ and + means modP summation.

Also, we represent above expression(3-1) to vector space, it is expression(3-2).

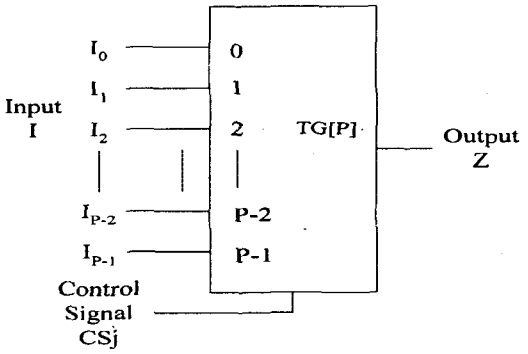
$F(\alpha)=\underline{F(\alpha)}=[a_{m-1}, a_{m-2}, \ldots\ldots, a_1, a_0]=\underline{F(\alpha)}[a_V]$
$G(\alpha)=\underline{G(\alpha)}=[b_{m-1}, b_{m-2}, \ldots\ldots, b_1, b_0]=\underline{G(\alpha)}[b_V]$
$A(\alpha)=\underline{A(\alpha)}=[A_{m-1}, A_{m-2}, \ldots\ldots, A_1, A_0]=\underline{A(\alpha)}[A_V]$
$F(\alpha)+G(\alpha)=\underline{F(\alpha)}[a_V]+\underline{G(\alpha)}[b_V]=\underline{A(\alpha)}[A_V]$ (3-2)

Where, $a_V, b_V, A_V \in GF(P)(V=0,1,\ldots, m-1)$

3.2 Basic A-cell

In order to construct adder, first we construct basic adder cell(A-cell) using data selector T-gate and modP cyclic gate. The following expression(3-3) represent T-gate operation and Fig.3-1 depict T-gate, expression(3-4) represent modP cyclic gate operation and Fig.3-2 depict modP cyclic gate.

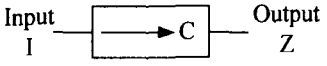
$$Z=I_i \quad \text{iff} \quad I_i=CS_j \qquad (3\text{-}3)$$



Where, I_i, Z, $CS_j \in GF(P)$ and i,j=0,1,.....,P-1

Fig.3-1. The block diagram of T-gate.

$$Z=I^{\rightarrow C}=(I+C) \text{ modP} \qquad (3\text{-}4)$$



Where, 1≤C≤P-1(C=integer)

Fig.3-2. The block diagram of modP cyclic gate.

As we see above contents, because of $A_k=a_i+b_j(I=j=k)$, A_k is obtained as following. The coefficient a_i use as T-gate input after passing modP cyclic gate, also b_j use as T-gate control signal. Therefore we construct A-cell Fig.3-3 and its characteristic operation is expression (3-5).

$$A_k=a_i^{\rightarrow b}{}_j=(a_i+b_j) \text{ modP} \qquad (3\text{-}5)$$

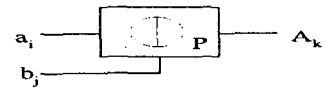


Fig.3-3. The block diagram of A-cell.

3.3 Adder module

We construct the adder module(A-module) using above section 3.1 and 3.2 . The Fig.3-4 shows block diagram of A-module.

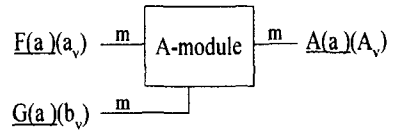


Fig. 3-4. The block diagram of adder module

4. Multiplier Module

There are 2m − 2 term of α for any two element multiplication over Galois Fields, that time we convert α term of α^k, m≤k≤2m-2, into less standard basis representation α term less than m-1 degree using irreducible primitive polynomial. Next we obtain the result that multiply two element after sum each α term.

We named Mod F(X) for this processing.

[Definition 4-1] Let $\delta[(a_0, a_1, \ldots, a_{m-2}, a_{m-1}),(b_0, b_1, \ldots, b_{m-2}, b_{m-1})]=M_k$, mapping function δ is binary operation, $\delta: GF(P^m)\times GF(P^m)\rightarrow GF(P)$. Where M_k is the kth product result of $(a_0, a_1, \ldots, a_{m-2}, a_{m-1})$ and $(b_0, b_1, \ldots, b_{m-2}, b_{m-1})$, and $a_i, b_j \in GF(P)(i,j=0,1,\ldots,m-1)$ and $0\le k\le 2m-2$.

Also mapping relationship is decided by selection irreducible primitive polynomial.

4.1 Multiplication algorithm

We put any two element over $GF(P^m)$, $F(\alpha)=\sum_{i=0}^{m-1} a_i\alpha^i$ and $G(\alpha)=\sum_{j=0}^{m-1} b_j\alpha^j$, and $M(\alpha)=\sum_{k=0}^{m-1} M_k\alpha^k$ that is the element after multiply them. Then we represent relationship among these elements as following.

$$F(\alpha)\bullet G(\alpha)=\sum_{i=0}^{m-1} a_i\alpha^i \bullet \sum_{j=0}^{m-1} b_j\alpha^j = a_{m-1}(\sum_{j=0}^{m-1} b_j\alpha^j)\,\alpha^{m-1+j}+$$

$$a_{m-2}(\sum_{j=0}^{m-1} b_j\alpha^j)\,\alpha^{m-2+j}\ldots+a_1(\sum_{j=0}^{m-1} b_j\alpha^j)\,\alpha^{1+j}+a_0(\sum_{j=0}^{m-1} b_j\alpha^j)\,\alpha^j$$

$$= \sum_{i,j=0}^{2m-2} a_i\, b_j\, \alpha^{i+j} \qquad (4\text{-}1)$$

where, $a_i, b_j \in GF(P)(i,j,k=0,1,\ldots,m-1)$, + and Σ are modP summation, • is mod P product.

As we see the expression(4-1), we partition α^k term into $m \le k1 \le 2m-2$ and $0 \le k2 \le m-1$. This is represent in expression(4-2).

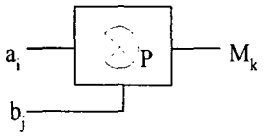
$$F(\alpha) \bullet G(\alpha) = \sum_{k1=m}^{2m-2} a_i\, b_j\, \alpha^{k1} \bullet \sum_{k2=0}^{m-1} a_i\, b_j\, \alpha^{k2} = \sum_{k=0}^{m-1} M_k \alpha^k$$
$$= M(\alpha) \qquad (4\text{-}2)$$

where, $k1 = a_i\, b_j (k1 = i+j = m, m+1, \ldots, 2m-2)$
$k2 = a_i\, b_j (k2 = i+j = 0, 1, \ldots, m-1)$

The other hand, these α^{k1} terms are used in input of control signal CSt.

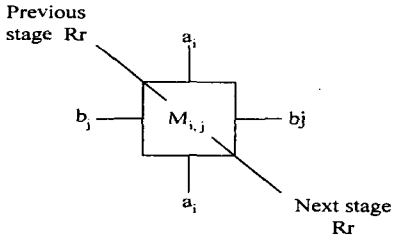
4.2 ModP multiplication gate and M-cell

This section discuss the modP multiplication processing device that is constructed by using T-gate, namely modP multiplication gate, it is depicted in Fig.4-1. And we construct basic M-cell using by modP multiplication gate and adder basic cell A-cell, it is depicted in Fig.4-2.

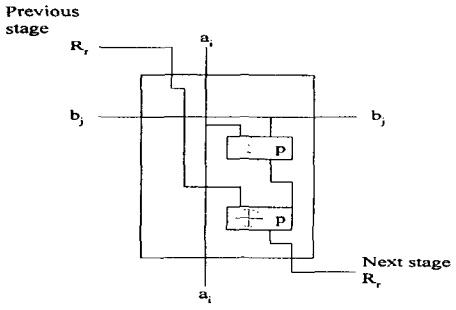


Where, $a_i, b_j, M_k \in GF(P)$

Fig. 4-1. The block diagram of modP multiplication Gate.



(a) symbol



(b) internal circuit

where, $a_i, b_j, R_s \in GF(P)$

Fig.4-2. Basic M-cell.

4.3 α^r generation module

The α^r generation module can be constructed by using M-cell, it is represented in Fig.4-3.

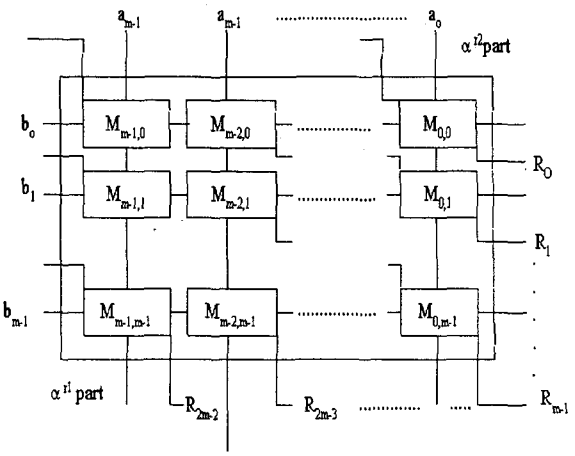


Fig. 4-3. α^r generation module

4.4 Control signal CSt generation module

The α^k term is generated in $m \le k1 \le 2m-2$ and $0 \le k2 \le m-1$, we can obtain multiplication result between two element using modP sum α^{k2} with result after decrease m-1 degree using irreducible primitive polynomial. Therefore α^{k1} term is defined according to α^{k2}, we named α^{k2} to control signal CSt(t=0,1,2,...,m-1).

This paper propose two algorithm of generating control signal CSt.

4.4.1 Combinational method

[STEP1] we select the proper irreducible primitive polynomial.

[STEP2] we construct basic control digit code $BCD_w(QQQ\ldots Q)$ of α^k. Where w=m,m+1,...,2m-2 and $Q \in GF(P)$.

[STEP3] final control signal CSt is obtained as following we disregard Q except corresponding R_{k2} and modP sum after each modP multiply.

The drawback of this algorithm in according to selected irreducible primitive polynomial. Therefore, in using this algorithm, we select irreducible primitive polynomial type $X^m+(P-1)X^{m-1}+(P-1)X^{m-2}+ \ldots\ldots + (P-1)X+(P-1)$.

4.4.2 Universal control signal CSt generation module

This proposed algorithm's advantage is usage of any irreducible primitive polynomial. That is not change basic control signal generation module, only input each α term coefficient in change the selected irreducible primitive polynomial. We named this algorithm as universal control signal CSt generation module.

This universal control signal CSt generation module operate modF(X). In order to obtain this function, we input coefficient of irreducible primitive polynomial to shift register, and shift each coefficient to next stage shift register in case of multiply α term in each time.

The Fig.4-4 depicted universal control signal CSt generation module.

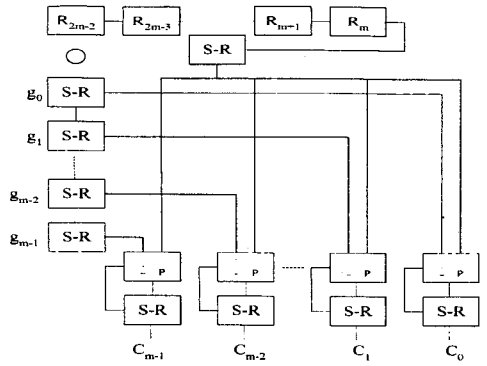


Fig.4-4. Universal control signal CSt generation module.

4.5 Multiplier

This section discuss constructing the multiplication module over galois fields. We can construct this multiplication module in merging α^r generation module with control signal generation module CSt. Where, final multiplication result $M_k(k=0,1,.....,m-1)$ between any two element over galois fields obtain R_{r2} of α^r mod P cyclic corresponding to control sgnal CSt. This is represented in exprssion (4-3).

$$M_k = R_{r2}^{\rightarrow CSt} \qquad (4\text{-}3)$$

Then, the expression(4-3) is the same as expression in adder module. Therefore we use the adder module in this part, this block diagram depicted in Fig.4-5.

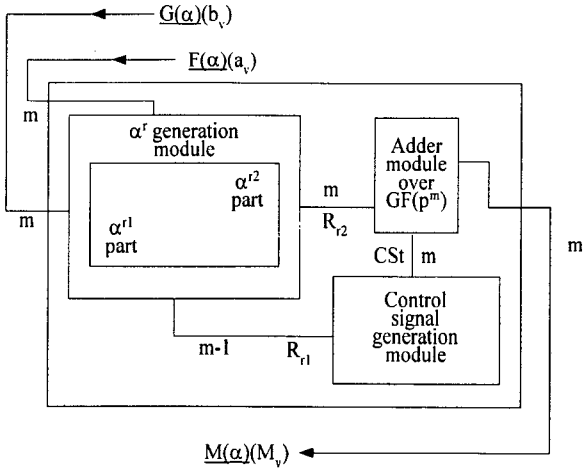


Fig.4-5. The multiplier over galois fields.

5. Conclusion

This paper propose the method of constructing the highly efficiency adder and multiplier systems over finite fields. The proposed highly adder/multiplier systems is more regularity, extensibility and modularity than any other research.

Also, the proposed highly efficiency adder and multiplier systems is fabricated in VLSI type easily.

The future demand research is the other arithmetic operation subtracter and divider, also need to constructing AOU(Arithmetic Operation Unit) in order to processing the four basic arithmetic operation. And we demanded more improvement ALOU(Arithmetic & Logical Operation Unit). The proposed highly efficiency adder and multiplier systems is able to apply modern multimedia hardware systems. The following table5-1 represented several item that compare proposed highly adder/multiplier over finite fields with any other research result.

Table 5-1. The comparison table

Comparison item	C.C. Wang	C-Ling etc.	S.T.J. Fenn etc.	This paper
Basis	SDNB	SB	DB	NB
I/O Type	SISO	SIPO	P-I/O	P-I/O
AND	3m	$2m^2$	$2m^2$	2m
OR	2m	$2m^2$	$2m^2$	m
# of control signal	2m-1	2m-1	2m-2	m-1
Overall Type	M-O	S-A	S-A	S-A
Regularity/ Extensibility	●	◎	0	0

Remarks : SISO : Serial Input Serial Output
SIPO : Serial Input Parallel Output,
P-I/O : Parallel I/O
I/O : Input/Output
SDNB : Standard Dual Normal Basis
SB : Standard Basis
NB : Normal Basis
DB : Dual Basis
M-O : Massey-Omura
S-A : Systolic Array
0 : Available ◎:some available
● : Disable

References

[1] D.Green, *Modern Logic Design*, Addison-Wesley company,1986.

[2] K. Hwang, *Comptuer Arithmetic principles, architecture, and design*,john Wiley & Sons,1979.

[3] C.C.Wang,"an algorithm to design finite field multipliers using a self-dual normal basis", IEEE Trans. Compt., vol.38,no.10,pp.1457-1460,Oct.1989.

[4] C.Ling and J.Lung,"Systolic array implementation of multipliers for finite fields $GF(2^m)$", IEEE Trans. Cir. & Sys., vol.38,no.7,pp.796-800,Jul.1991.

[5] S.T.J.Fenn, M.Benaissa and D.Taylor,"$GF(2^m)$ multiplication and Division over dual basis", IEEE Trans. Comput., vol.45,no.3,pp.319-327,Mar.1996.

[6] K.Z.Pekmastzi,"multiplxer-based array multipliers", IEEE Trans. Comput.,vol.48.no.1,pp.15-23,Jan.1999.

[7] G.Drolet,"A new representation of elements of finite fields $GF(2^m)$ yields small complexity arithmetic circuits,' IEEE Trans. Comput.,vol.47.no.9,pp.938-946,Sep.1998.

[8] E. Artin, *Galois Theory*, NAPCO Graphics arts, Inc., Wisconsin.1971.

[9]R.Lidi and H.Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press,1986.