論文 90-27-6-13

GF(2^m)上의 算術演算器시스템 構成 理論

(A Construction Theory of Arithmetic Operation Unit Systems over GF(2^m))

朴 春 明*, 金 興 壽*

(Chun Myoung Park and Heung Soo Kim)

要 約

本 論文에서는 Galois体 GF(P^m)上에서 素数 P가 2이고 m이 陽의 整数인 GF(2^m)上의 元素들간에 算術演算(加算, 減算, 乘算, 除算)을 행하기 위한 算術演算器시스템 構成 方法의 한가지를 提示하였다.

提案된 算術演算器시스템의 構成 方法은 다음과 같다. 우선 Galois体의 数学的 性質로 부터 加算, 減算, 乘算 및 除算 알고리듬을 各各 밝힌다음 이들 알고리듬을 토대로 加算器, 減算器, 乘算器 및 除算器 module들 構成時에 必要한 基本 Cell을 構成한후 이 基本cell을 사용하여 加算器, 減算器, 乘算器 및 除算器 module들을 각각 構成하였다. 그리고 앞의 算術演算器 module들을 合成하기 위해 먼저 分配器 module의 基本cell을 構成한 후 分配器 module을 構成하였으며 이들 分配器module을 사용하여 最終 GF(2^m)上의 算術演算器시스템을 構成하였다.

Abstract

This paper presents a method of constructing an Arithmetic Operation Unit Systems (A.O.U.S.) over Galois Field GF(2^m) for the purpose of the four arithmetical operation (addition, subtraction, multiplication and division) between two elements in GF(2^m).

The proposed A.O.U.S. is constructed by following procedure.

First of all, we obtained each four arithmetical operation algorithms for performing the four arithmetical operations using by mathematical properties over GF(2^m). Next, for the purpose of realizing the four arithmetical unit module (adder module, subtracter module, multiplier module and divider module), we constructed basic cells using the four arithmetical operation algorithms.

Then, we realized the four Arithmetical Operation Unit Modules (A.O.U.M.) using basic cells and we constructed distributor modules for the purpose of merging A.O.U.M. with distributor modules. Finally, we constructed the A.O.U.S. over GF(2^m) by synthesizing A.O.U.M. with distributor modules.

We prospect that we are able to construct an Airthmetic & Logical Operation Unit Systems (A.L.O.U.S.) if we will merge the proposed A.O.U.S. in this paper with Logical Operation Unit Systems (L.O.U.S.).

*正會員, 仁荷大學校 電子工學科
(Dept. of Elec. Eng., Inha Univ.)

接受日字 : 1989年 8月 19日

I. 序 論

現在의 2進論理를 根幹으로한 디지탈시스템 및 컴

퓨터는 2進論理 回路上에서의 回路의 複雜性과 大型化, 演算速度의 制限性, 情報傳送量의 방대함에 따른 情報傳送時間遲延등과 같은 問題点들이 대두되기 시작하였다.[1-4]

따라서 이와같은 問題点들을 해결할 수 있는 概念으로써 多値論理理論이 1970年代 初부터 활발히 研究·進行 되어왔다.

한편, 多値論理理論을 研究, 解析 및 合成하는데 必要한 数学的 概念은 抽象代数学(abstract algebra) 概念이 요구되며 특히 有限体인 Galois 体上에서 多値論理를 容易하게 解析 및 合成할 수 있다.[5-8][12-15] 또한, Galois体 GF(P^m) 上에서 P=2인 경우의 GF(2^m)은 現存의 2進論理 디지탈시스템과 호환성이 가능하다. 즉, 現存의 2進論理 디지탈시스템은 GF(2^m)上에서 m=1인 경우와 같으며 数学的 概念으론 부울代数(Boolean algebra)가 導入된다. 또한, GF(2^m)은 Error-Correcting Code分野를 비롯하여 Digital Signal Processing, Digital Image Processing, Digital Information Processing, Digital Communication의 暗號化 및 解讀化, 般宛宇宙通信 分野등과 디지탈스위칭 理論을 容易하게 展開시킬 수 있다.[9-17]

前術한 여러가지 分野와 多値論理시스템에서 算術演算器시스템 構成은 重要하며 이들에 관한 여러 論文들이 發表되어 왔다.[18-23] GF(2^m)上의 算術演算에서 加算과 減算은 各各 mod2合으로 解析되어 간단한 편이지만 乘算과 除算은 많은 計算過程이 必要하다. 따라서 이러한 計算過程을 効果的으로 수행키 위하여 여러가지의 乘算알고리듬과 乘算器에 관한 研究가 進行되어왔다.[18-20] 또한 이들 乘算器들은 크게 Parallel Processing 형태와 Serial Processing 형태로 分類할 수 있다.

本 論文에서는 GF(2^m)上에서 m의 擴張에 따른 算術演算器시스템 構成 方法중의 한가지를 提示하였다.

本 論文의 叙述過程은 다음과 같다. Ⅱ章에서는 Galois体의 重要한 性質을 論議하고 Ⅲ章에서는 四則演算인 加算, 減算, 乘算 및 除算의 알고리듬을 밝히고 이를 토대로하여 加算器 基本cell[25-26]과 乘算器 基本cell[25-26]을 構成하였다. 그리고 이들 基本cell을 根幹으로 加算器, 減算器, 乘算器 및 除算器 module들을 各各 構成하였다. Ⅳ章에서는 먼저 分配器 module을 構成하기위해 基本cell[24-26]을 構成한후 分配器 module을 構成하였으며 이를 토대로 Ⅲ章에서 構成한 各各의 算術演算器module들을 合成하여 最終의 GF(2^m)上의 算術演算器시스템을 構成하였다. Ⅴ章에서는 Ⅲ章과 Ⅳ章의 內容이 어떻게 適用되는지 例를 들어 説明하였고 마지막 Ⅵ章의 結論에서는 本 論文에서 提案한 GF(2^m)上의 算術演算器 시스템의 特徵을 要約하였으며 앞으로의 展望을 記術하였다.

Ⅱ. Galois体의 重要한 数學的 性質

〈P1〉 다음 式(1)을 因数分解하여 m次 既約多項式을 구하고 이 既約多項式을 0으로 하는 한 根을 α로 할때 式(2)와 같은 原始既約多項式을 얻을 수 있다.

$$X^{P^m}-X=X(X-1)(X^{P^m-2}+X^{P^m-3}+\cdots+X+1)=0 \quad (1)$$

여기서 P는 素数, m은 陽의 整数

$$F(\alpha)=\sum_{i=0}^{m-1}a_i\alpha^i$$

$$=a_{m-1}\alpha^{m-1}+a_{m-2}\alpha^{m-2}+\cdots+a_1\alpha+a_0 \quad (2)$$

여기서 α는 P를 法으로 하는 整数次 $Z_P(=GF(P))$의 元素를 係数로 하는 m次 原始既約多項式의 根이고 $a_i\in Z_P$ $(i=0,1,\cdots p^m-1)$이다.

또한 式(2)를 벡터공간(Vector space)으로 表示하면 다음 式(3)과 같고 특히 P=2인 경우에 비트벡터공간(Bit Vector space)이라 한다.[23]

$$\underline{F(\alpha)}=(a_{m-1},a_{m-2},\cdots a_1,a_0) \quad (3)$$

여기서 a_i $(i=m-1,m-2,\cdots1,0)$는 α^i의 係数이다.

〈P2〉 GF(P^m)上에서의 元素生成은 式(2)를 0으로 하는 한 根 α의 冪乘으로 구해지며 元素의 갯수는 P^m개이다. 이 內容을 式으로 表示하면 다음 式(4)와 같다.

$$GF(P^m)=\underbrace{\{0,\alpha^1,\alpha^2,\cdots,\alpha^{P^m-2}=\alpha^{-1},\alpha^{P^m-1}=1\}}_{P^m개} \quad (4)$$

〈P3〉 逆元의 存在

(1) $a+(-a)=0$인 加法에 관한 逆元 $-a$가 存在한다. $(\forall a\in GF(P^m))$

(2) $a\cdot a^{-1}=1$인 乘法에 관한 逆元 a^{-1}이 存在한다. $(\forall a\in GF(P^m))$

〈P4〉 $(a+b)^{P^m}=a^{P^m}+b^{P^m}=a+b(\forall a,b\in GF(P^m))$

〈P5〉 $\alpha^i\cdot\alpha^j=\alpha^{i+j(\mathrm{mod}P^m-1)}$

단, $i+j(\mathrm{mod},P^m-1)$은 $i+j\equiv r(\mathrm{mod}\,P^m-1)$, $0\le r\le P^m-1$이다.

以上의 数學的 性質과 그 외의 本 論文의 展開에 必要한 数学的 性質은 参考文献[5-8][11-15]에서 引用하였다.

Ⅲ. 算術演算알고리듬 및 算術演算器module 構成

1. 加算알고리듬 및 加算器module 構成

1) 加算알고리듬

彼加算元素, 加算元素 및 加算後元素를 各各 e_i, e_j 및 e_a라하고 이들을 비트벡터공간으로 表現한 것을 各各 $\underline{e_i}(a_v)$, $\underline{e_j}(b_v)$ 및 $\underline{e_a}(A_v)$라 하면 두 元素 e_i와 e_j의 加算은 다음 式(5)와 같다.

$$e_i \oplus e_j = \underline{e_i}(a_v) \oplus \underline{e_j}(b_v) = \underline{e_a}(A_v) \qquad (5)$$

여기서 $\begin{cases} i, j = 2^m-1, 2^m-2, \cdots, 1, 0 \\ a_v, b_v, A_v = GF(2) = \{0, 1\}\ (V = m-1, m-2, \cdots, 1, 0) \\ \oplus\ ; \text{mod } 2\ 合 \end{cases}$

式(5)에서 본바와 같이 $A_v = a_v \oplus b_v$이다. 따라서 b_v를 加算時의 制御入力으로 사용하면 加算後元素 A_v는 b_v값에 따라 a_v값을 그대로 유지하거나 2의 補数를 취한 값이 되고 이는 mod2合의 数学的 性質과 같다. 이 內容을 式으로 表示하면 다음 式(6)과 같다.

$$A_v = \begin{cases} a_v & \text{iff } b_v = 0 \\ \overline{a_v} & \text{iff } b_v = 1 \end{cases} \qquad (6)$$

여기서 $\begin{cases} a_v, b_v, \bar{a}_v \in GF(2) = \{0, 1\} \\ V = m-1,\ m-2, \cdots, 1, 0 \end{cases}$

이제 앞의 內容과 式(5)및 (6)을 토대로 $GF(2^m)$ 上의 加算알고리듬을 세우면 다음과 같다.

[節次 1] 彼加算元素 e_i와 加算元素 e_j를 各各 비트벡터공간으로 表現한 $\underline{e_i}(a_v)$와 $\underline{e_j}(b_v)$로 表示한다.

[節次 2] 加算元素 $\underline{e_j}(b_v)$를 制御入力으로 사용하여 b_v의 값이 1) "0"이면 彼加算元素 $\underline{e_i}(a_v)$의 해당 비트값을 그대로 유지하고 2) "1"이면 2의 補数를 취한다.

[節次 3] 節次2를 행한후의 結果가 最終 加算後의 元素 $\underline{e_a}(A_v)$가 된다.

2) 加算器module 構成

(1) 加算器 基本 cell(A-cell) 構成

1)節의 式(6)을 토대로 $GF(2^m)$上의 加算器 基本 cell을 構成하면 그림 1 과 같고 A-cell이라 한다.

(2) 加算器module 構成

加算알고리듬과 加算器 基本cell을 사용하여 $GF(2^m)$上의 加算器module을 構成하면 다음 그림2와 같다.

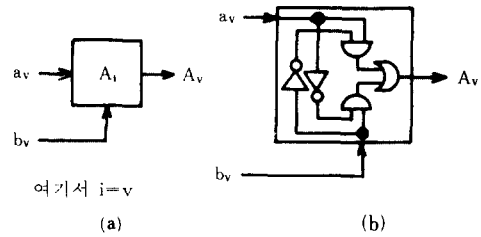


그림 1. GF(2^m)上의 加算器 基本cell(A-cell)
Fig. 1. Basic cell(A-cell) of adder over GF(2^m).
(a) symbol of basic A-cell.
(b) internal circuits of basic A-cell.

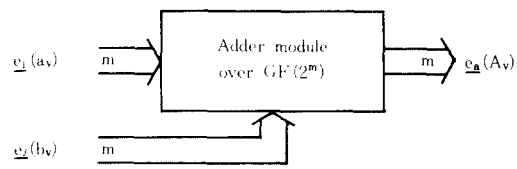


그림 2. GF(2^m)上의 加算器module
Fig. 2. An adder module over GF(2^m).

2. 減算알고리듬 및 減算器module 構成

Mod2의 数学的 性質에 의해 減算은 加算과 같다. 따라서 GF(2^m)上에서 두 元素간의 減算알고리듬은 加算알고리듬과 같고 減算器 基本cell과 減算器module은 各各 그림1, 2와 同一하다.

3. 乘算알고리듬 및 乘算器module 構成

1) 乘算알고리듬

彼乘算元素, 乘算元素 및 乘算後元素를 各各 e_i, e_j 및 e_m이라 하고 이를 비트벡터공간으로 表現한 것을 各各 $\underline{e_i}(a_v)$, $\underline{e_j}(b_v)$ 및 $\underline{e_m}(M_v)$라 하면 두 元素 e_i와 e_j의 乘算은 다음 式(7)과 같다.

$$e_i \cdot e_j = \underline{e_i}(a_v) \cdot \underline{e_j}(b_v) = \underline{e_m}(M_v) \qquad (7)$$

한편, 彼乘算元素, 乘算元素 및 乘算後元素의 原始既約多項式을 各各 $F(\alpha) = \sum_{i=0}^{m-1} a_i \alpha^i$, $G(\alpha) = \sum_{j=0}^{m-1} b_j \alpha^j$와 $H(\alpha) = \sum_{k=0}^{m-1} M_k \alpha^k$라 하면 식(7)은 다음 式(8)과 같이 다시 쓸 수 있다.

$$\begin{aligned} F(\alpha) \cdot G(\alpha) &= \left(\sum_{i=0}^{m-1} a_i \alpha^i\right) \cdot \left(\sum_{j=0}^{m-1} b_j \alpha^j\right) \\ &= \sum_{i=0}^{m-1} \left(\sum_{j=0}^{m-1} b_j\right) a_i \alpha^{i+j} \\ &= \sum_{i+j=0}^{2m-2} a_i b_j \alpha^{i+j} \end{aligned} \qquad (8)$$

여기서 $\begin{cases} a_i, b_j \in GF(2) = \{0, 1\} \\ i, j = m-1, m-2, \cdots, 1, 0 \end{cases}$

여기서 r=i+j라 하면 式(8)은 다음 式(9)와 같고 이는 H(α)와 같아야 한다.

$$F(\alpha)\cdot G(\alpha)=\sum_{r=0}^{2m-2} a_i b_j \alpha^r$$
$$=H(\alpha)$$
$$=\sum_{k=0}^{m-1} M_k \alpha_k \qquad (9)$$

따라서 α^r의 r는 $m\le r1\le 2m-2$ 部分과 $0\le r2\le m-1$ 部分으로 分割할 수 있으며 α^{r1}項을 数学的 性質로부터 α^{r2}項으로 表現하여 α^k項과 일치시킬 수가 있다.

또한 이들 α^{r1}項들이 3)節의 乘算器module중 制御入力 C_L 生成module의 入力이 되고 이 C_L에 의해서 最終 乘算後元素 $\underline{e_m}(M_v)$를 얻는다. 위 內容을 式으로 表示하면 다음 式(10)과 같고 乘算過程을 나타내면 다음 그림 3 과 같다.

$$\sum_{r=0}^{2m-2} a_i b_j \alpha^r=\sum_{r1=m}^{2m-2} a_i b_j \alpha^{r1}+\sum_{r2=0}^{m-1} a_i b_j \alpha^{r2}$$
$$=\underbrace{\sum_{r1=m}^{2m-2} R_{r1}\alpha^{r1}}_{\text{制御入力}C_L\text{生成 module의 入力部分}}+\underbrace{\sum_{r2=0}^{m-1} R_{r2}\alpha^{r2}}_{\text{制御入力}C_L\text{에 따라 最終乘算後의 結果가 될 部分}}$$
$$=\underbrace{\sum_{k=0}^{m-1} M_k \alpha^k}_{\text{最終乘算後結果}} \qquad (10)$$

여기서 $\begin{cases} R_{r1}=a_i\cdot b_j\,(r1=i+j=2m-2,\,2m-3,\cdots,\\ \quad m+1,\,m)\\ R_{r2}=a_i\cdot b_j\,(r2=i+j=m-1,\,m-2,\cdots,1,0)\end{cases}$

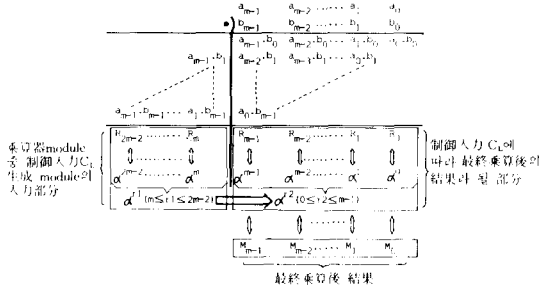


그림 3. GF(2^m)上의 乘算過程
Fig. 3. Multiplication process over GF(2^m).

이제 앞의 內容과 式(8), (10) 및 그림 3 을 토대로 GF(2^m)上의 乘算알고리듬을 세우면 다음과 같다.

[節次 1] 彼乘算元素 e_i와 乘算元素 e_j를 各各 비트벡터공간으로 表現한 $\underline{e_i}(a_v)$와 $\underline{e_j}(b_v)$로 表示한다.

[節次 2] 式(10)과 같이 α^{r1}項과 α^{r2}項을 各各 구한다.

[節次 3] α^{r1}項을 α^{r2}項으로 表現하여 制御入力 C_L을 구한다.

[節次 4] 節次3에서 구한 C_L의 값이 1) "0"이면 해당 α^{r2}項의 비트값을 그대로 유지하고 2) "1"이면 2의 補数를 취한다.

[節次 5] 節次4를 행한후의 結果가 最終 乘算後元素 $\underline{e_m}(M_v)$가 된다.

2) 乘算器 module 構成

(1) 乘算器 基本cell(M-cell)과 α^r生成module 構成

식(8)과 (10)을 토대로 乘算器 基本cell(M-cell)과 α^r 生成module을 構成하면 各各 그림4, 5와 같다. 이때 乘算器 基本cell은 內部에 加算器 基本cell인 A-cell을 포함하고 있으며 α^r生成module도 α^{r1}部分과 α^{r2}部分으로 나누어진다.

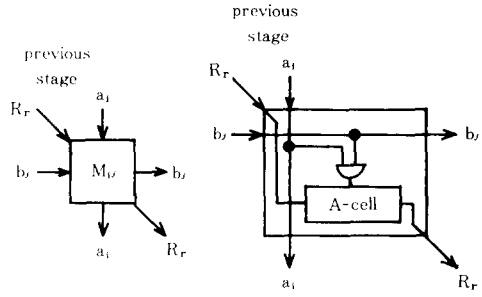


그림 4. GF(2^m)上의 乘算器 基本cell(M-cell)
Fig. 4. Basic cell (M-cell) of multiplier over GF(2^m).
(a) symbol of basic M-cell,
(b) internal ciruits of basic M-cell.

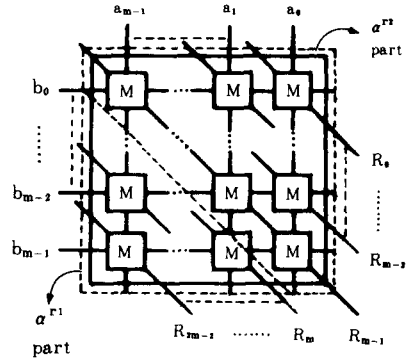


그림 5. GF(2^m)上의 α^r生成module
Fig. 5. α^r generation module over GF(2^m).

(2) 制御入力C_L生成module 構成

制御入力 $C_L(L=m-1, m-2, \cdots, 1, 0)$은 式(10)의 α^{r1} 項으로 부터 구할 수 있다. 즉, $\sum_{r1=m}^{2m-2} R_{r1}\alpha^{r1}$을 α^{r2}項으로 表現해 이들을 mod2合 함으로써 쉽게 구할 수 있으며 이 內容을 式으로 表示하면 다음 式(11)과 같고 이때 制御入力 C_L의 갯수는 m개이다.

$$\sum_{r1=m}^{2m-2} R_{r1}\alpha^{r1} = \sum_{r1=m}^{2m-2} R_{r1}\left(\sum_{L=0}^{m-1}\alpha^{L}\right) \tag{11}$$

(3) 乘算器module 構成

乘算器module은 (1)節의 α^r生成module과 (2)節의 制御入力C_L生成module을 合成함으로써 구할 수 있는데 이때 C_L의 값이 1) "0"이면 式(10)의 M_k 값은 R_{r2} 값을 유지하고 2) "1"이면 R_{r2}값에 2의 補数를 취한 값이 된다. 이 內容을 式으로 表示하면 다음 式(12)와 같다

$$M_k = \begin{cases} R_{r2} & \text{iff } C_L=0 \\ \overline{R}_{r2} & \text{iff } C_L=1 \end{cases} \tag{12}$$

여기서 $\begin{cases} M_k, C_L, R_{r2}, \overline{R}_{r2} \in GF(2)=\{0,1\} \\ k, L, r_2 = m-1,\ m-2, \cdots 1, 0 \end{cases}$

그런데 式(12)는 加算器module을 表現한 式(6)과 數學的으로 내용이 同一하다. 따라서 이 部分은 加算器module을 그대로 사용할 수 있으며 이 內容을 토대로 乘算器module을 構成하면 다음 그림6과 같다.

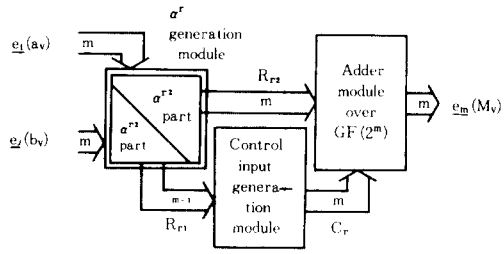


그림 6. $GF(2^m)$上의 乘算器module
Fig. 6. An multiplier module over $GF(2^m)$.

4. 除算알고리듬 및 除算器module 構成

1) 除算알고리듬

彼除算元素, 除算元素 및 除算後元素를 各各 e_i, e_j 및 d_d라 하고 이를 비트벡터공간으로 表現한 것을 各各 $\underline{e_i}(a_v)$, $\underline{e_j}(b_v)$ 및 $\underline{e_d}(D_v)$라 하면 두 元素 e_i와 e_j의 除算은 다음 式(13)과 같다.

$$\begin{aligned} e_i/e_j &= \underline{e_i}(a_v)/\underline{e_j}(b_v) = e_i \cdot e_j^{-1} \\ &= \underline{e_i}(a_v)\cdot\underline{e_j}(b_v^*) = \underline{e_d}(D_v) \end{aligned} \tag{13}$$

여기서 e_j^{-1}은 e_j의 逆元이며 b_v^*는 b_v에 대한 逆元 비트벡터 공간이다.

한편, 彼除算元素, 除算元素 및 除算後元素의 原始既約多項式을 各各 $F(\alpha)$, $G(\alpha)$및 $H(\alpha)$라 하면 식(13)은 다음 式(14)와 같이 쓸 수 있다.

$$\begin{aligned} F(\alpha)/G(\alpha) &= F(\alpha)\cdot[G(\alpha)]^{-1} \\ &= H(\alpha) \end{aligned} \tag{14}$$

여기서 $[G(\alpha)]^{-1}$ 은 $G(\alpha)$에 대한 乘法逆元生成多項式(multiplicative inverse element generation polynomial)이다.

II章의 数学的 性質 P3과 式(4)로 부터 다음 式(15)를 구할 수 있다.

$$\beta = \alpha^{-1} = \alpha^{2^m-2} \tag{15}$$

또한 数学的 性質 P2에 의해서 $F(\alpha)=0$의 α를 冪乘하여 $GF(2^m)=\{0, \alpha^1, \alpha^2, \cdots, \alpha^{2^m-2}, \alpha^{2^m-1}\}$을 얻고 $G(\beta)=0$의 β를 冪乘하여 $GF(2^m)=\{0, \beta^1, \beta^2, \cdots, \beta^{2^m-2}, \beta^{2^m-1}\}$을 얻을 수 있다. 이때 $\alpha^i\cdot\beta^i=1$ $(i\neq 0, i=1, 2, \cdots, 2^m-1)$이 된다.

式(14)에서 본 바와 같이 除算에서는 $G(\alpha)$에 대한 乘法逆元生成多項式인 $[G(\alpha)]^{-1}$을 구하여 이를 $F(\alpha)$에 곱해주면 된다. 따라서 $[G(\alpha)]^{-1}$은 다음 式(16)과 같이 表示할 수 있다.

$$[G(\alpha)]^{-1} = \left(\sum_{j=0}^{m-1} b_j\alpha^j\right)^{2^m-2} \tag{16}$$

한편 式(16)에서 $Q=2^m-2$라 하면 式(17)과 같이 自乘(Sguaring)꼴로 分割할 수 있다.[20]

$$Q = 2^m-2 = 2^1+2^2+2^3+\cdots+2^{m-1} \tag{17}$$

따라서 式(16)은 다음 式(18)과 같다.

$$\begin{aligned} [G(\alpha)]^{-1} &= [G(\alpha)]^{2^1}\cdot[G(\alpha)]^{2^2}\cdot[G(\alpha)]^{2^3}\cdots[G(\alpha)]^{2^{m-1}} \\ &= \sum_{j=0}^{m-1} b_j^*\alpha^j \end{aligned} \tag{18}$$

여기서 b_j^*는 逆元의 係数이며 b_j의 組合으로 이루어진다.

이제 앞의 內容과 式(14), (18)을 토대로 $GF(2^m)$ 上의 除算알고리듬을 세우면 다음과 같다.

[節次 1] 彼除算元素 e_i와 除算元素 e_j를 各各 비트벡터공간으로 表現한 $\underline{e_i}(a_v)$와 $\underline{e_j}(b_v)$로 표시한다.

[節次 2] 除算元素의 原始既約多項式 $G(\alpha)$에 대한 乘法逆元生成多項式 $[G(\alpha)]^{-1}$과 逆元비트벡터공간 b_v^*로 表示된 $\underline{e_j}(b^*)$를 구한다.

[節次 3] 節次2에서 구한 $\underline{e_j}(b_v)$를 乘算 알고리듬의 乘算元素로 한다.

[節次 4] 以後 부터는 乘算알고리듬의 節次 2 以下와 同一하다.

2) 乘法逆元生成module 構成 및 除算器module構成

(1) 乘法逆元生成 module 構成

1)節의 內容을 토대로 乘法逆元生成module 構成은 다음과 같다. 먼저 GF(2^m)上에서 m의 擴張에 따른 原始既約多項式 G(α)를 선택한후 G(α)에 대한 乘法逆元生成 多項式 [G(α)]^-1을 구하여 b*를 구하고 이를 PLA(programmable logic array)를 사용하여 構成한다.

(2) 除算器module 構成

(1)節의 乘法逆元生成module의 出力 b*를 GF(2^m)上의 乘算器module의 乘算元素 入力에 연결함으로써 除算器module을 構成할 수 있으며 이 內容을 그림으로 나타내면 다음 그림7과 같다.

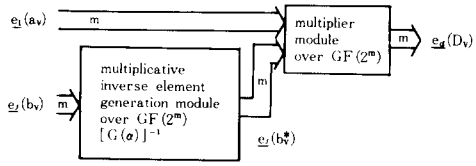


그림 7. GF(2^m)上의 除算器module
Fig. 7. A divider module over GF(2^m).

Ⅳ. GF(2^m)上의 算術演算器시스템 構成

1. 分配器module(Distributor module)構成

算術演算時 加算, 減算, 乘算 및 除算을 수행키 위해 加減乘除의 各 算術演算器module을 선택하기 위한 部分이 必要하다. 따라서 이를 행하기 위해서 本論文에서는 分配器module을 構成하였다.

1) 分配器module D1 構成

(1) module D1의 基本cell(D1-cell) 構成

彼算術演算元素인 e1(av)는 1)加算과 減算일때는 加算器module의 入力으로 2)乘算과 除算일때는 α^r生成module의 入力으로 사용한다. 그러므로 이를 수행키 위한 制御入力 T1과 Pass Transistor G_{D1i}(i=0, 1)로써 module D1의 基本cell을 構成할 수 있다. 위 내용을 식으로 表示하면 다음 式(19)와 같고 이 式을 토대로 D1-cell을 構成하면 그림8과 같으며 이에 대한 眞理値表는 表1과 같다.

$$a_i = \begin{cases} y_{0i} \text{ if } T1=0 \Rightarrow \text{加算器module} \\ y_{1i} \text{ if } T1=1 \Rightarrow \text{乘算器module} \end{cases} \quad (19)$$

여기서 i=m−1, m−2, ⋯, 1, 0

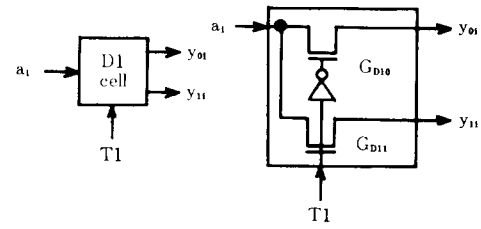


그림 8. Module D1의 基本cell(D1-cell)
Fig. 8. Basic cell(D1-cell) of module D1.
(a) symobol of D1-cell,
(b) internal circuits of D1-cell.

표 1. Module D1의 基本cell(D1-cell)에 대한 眞理値表
Table 1. Truth table for basic cell(D1-cell) of module D1.

a_i	T1	G_{D10} Y_{0i}	G_{D11} Y_{1i}
a_i	0	ON a_i	OFF x
a_i	1	OFF x	ON a_i

where, x→nonpass

(2) 分配器module D1 構成

(1)節의 內容을 토대로 分配器module D1을 構成하면 다음 그림 9와 같다.

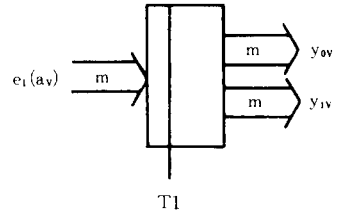


그림 9. 分配器module D1
Fig. 9. Distributor module D1.

이때 分配器module D1을 構成하는 D1-cell의 갯수는 m개이다.

2) 分配器module D2 構成

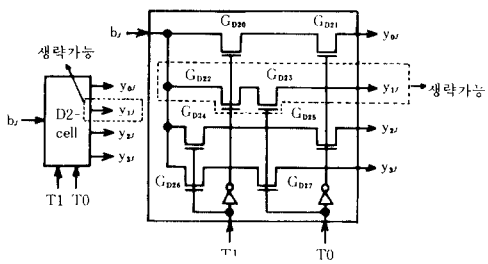
(1) module D2의 基本cell(D2−cell) 構成

算術演算元素인 e2(bv) 1)加算과 減算일때는 加算器module의 入力으로 2)乘算일때는 α^r生成module의 入力으로 3)除算일때는 乘法逆元生成module의 入力으로 사용한다. 그러므로 이를 수행키 위한 制御入

力 T1과 T0 및 Pass Transistor G_{D2i} (i=0, 1, …7)로써 module D2의 基本cell을 構成할 수 있다. 위 내용을 식으로 表示하면 다음 式(20)과 같고 이 式을 토대로 D2-cell을 構成하면 그림10과 같으며 이에대한 眞理値表는 表 2 와 같다.

$$b_j = \begin{cases} y_{0j} \text{ if } T1T0=00 \Rightarrow \text{加算器module} \\ y_{1j} \text{ if } T1T0=01 \Rightarrow \text{減算器module(생략가능)} \\ y_{2j} \text{ if } T1T0=10 \Rightarrow \alpha^r \text{生成module} \\ y_{3j} \text{ if } T1T0=11 \Rightarrow \text{乘法逆元生成module} \end{cases} \quad (20)$$

한편, GF(2^m)上에서의 減算은 加算과 같으므로 y_{1j} 部分은 생략해도 된다.



(a) (b)

그림 10. Module D2의 基本cell(D2-cell)
Fig. 10. Basic cell(D2-cell) of module D2.
(a) symbol of D2-cell.
(b) internal circuits of D2-cell.

표 2. Module D2의 基本 cell(D2-cell)에 대한 眞理値表
Table 2. Truth table for basic cell(D2-cell) of module D2.

b_j	T1 T2	G_{D20} G_{D21} Y_{0j}	G_{D22} G_{D23} Y_{1j}	G_{D24} G_{D25} Y_{2j}	G_{D26} G_{D27} Y_{3j}
b_j	0 0	ON ON b_j	ON OFF x	OFF ON x	OFF OFFx
b_j	0 1	ON OFF x	ON ON b_j	OFF OFFx	OFF ON x
b_j	1 0	OFF ON x	OFF OFFx	ON ON b_j	ON OFF x
b_j	1 1	OFF OFFx	OFF ONx	ON OFF x	ON ON b_j

생략가능

where, x→nonpass

(2) 分配器 module D2 構成

(1)節의 內容을 토대로 分配器module D2를 構成하면 다음 그림11과 같다.

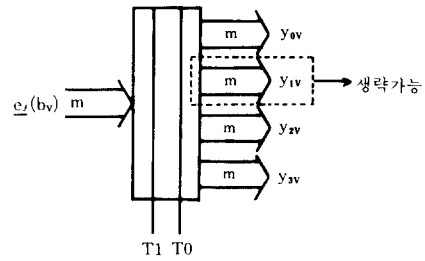
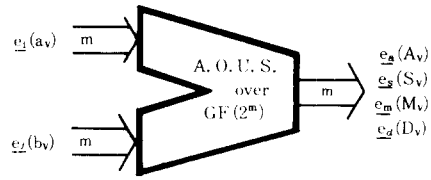


그림11. 分配器module D2
Fig. 11. Distaibutor module D2.

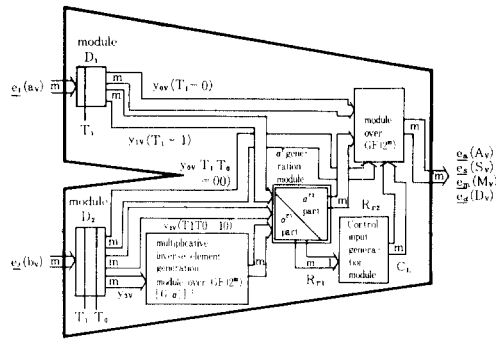
이때 分配器module D2를 構成하는 D2-cell의 갯수는 m개이다.

2. GF(2^m)上의 算術演算器시스템 構成

1節의 分配器module D1, D2와 Ⅲ章의 加算器module, 乘算器module 및 除算器module을 合成하여 GF(2^m)上의 算術演算器시스템을 構成하면 다음 그림12와같다.



(a) symbol of A. O. U. S. over GF(2^m).



(b) internal structure of A. O. U. S. over GF(2^m).

그림12. GF(2^m)上의 算術演算器시스템
Fig. 12. Arithmetic Operation Unit Systems over GF(2^m).

V. 適用例

이 章에서는 Ⅲ章과 Ⅳ章의 內容이 어떻게 適用되어 GF(2^m)上의 算術演算器시스템을 構成하는지 例를 들어 說明한다.

例) GF(2³)上의 算術演算器시스템을 構成하면 다음과 같다.

GF(2³)上에서의 原始旣約多項式으로써 다음 式(21)을 선택한다.

$$F(x)=x^3+x+1 \tag{21}$$

1. 加算器module 構成

Ⅲ章의 1節 內容과 그림2를 토대로 GF(2³)上의 加算器module을 構成하면 다음 그림13과 같다.

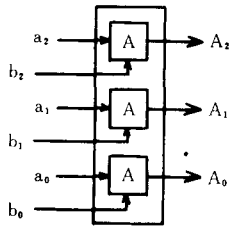


그림13. GF(2³)上의 加算器module
Fig. 13. An adder module over GF(2³).

2. 乘算器module 構成

式(21)을 0으로 하는 한 根을 α라 하면 다음 식(22)를 얻는다.

$$\alpha^3 \triangleq \alpha+1 \tag{22}$$

한편, 制御入力 C_L은 Ⅲ章의 3-2)-(2)節의 內容과 式(10), (11)로 부터 $3 \le r1 \le 4$이므로 $\alpha^4=\alpha^2+\alpha$이고 α^3은 식(22)와 같다. 따라서 制御入力 $C_L(L=0,1,2)$은 各各 다음 식(23), (24) 및 (25)와 같다.

$$C_2=\alpha^4=R_4 \tag{23}$$

$$C_1=\alpha^4 \oplus \alpha^3=R_4 \oplus R_3 \tag{24}$$

$$C_0=\alpha^3=R_3 \tag{25}$$

위 內容과 그림6을 토대로 GF(2³)上의 乘算器module을 構成하면 다음 그림14와 같다.

3. 除算器module 構成

除算器module 構成에서는 乘法逆元生成module構成이 重要하므로 먼저 Ⅲ章의 4節 內容과 式(17), (18)로

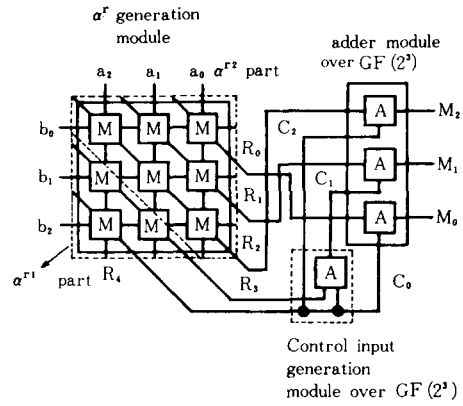


그림14. GF(2³)上의 乘算器module
Fig. 14. A multiplier module over GF(2³).

부터 다음 式(26)의 乘法逆元生多項式을 구하여 PLA로 構成하면 그림15와 같고 逆元비트벡터공간 $b_v^*(v=0,1,2)$를 GF(2³)上의 乘算器module의 乘算元素 入力에 연결하므로써 GF(2³)上의 除算器module을 구할 수 있다.

$$\begin{aligned}[G(\alpha)]^{-1}&=[G(\alpha)]^6\\&=(b_2\cdot b_0\oplus b_2\oplus b_1)\alpha^2\oplus(b_1\cdot b_0\oplus b_2)\oplus(b_2\cdot b_1\oplus b_2\oplus b_1\oplus b_0)\\&=b_2^*\alpha^2+b_1^*\alpha+b_0^*\end{aligned} \tag{26}$$

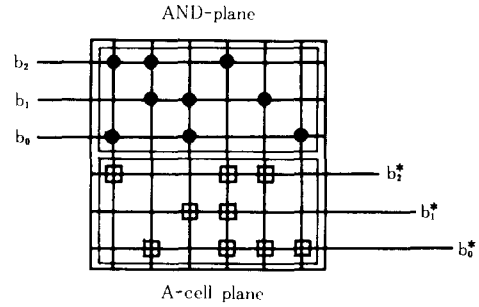


그림15. GF(2³)上의 乘法逆元生成 module
Fig. 15. A multiplicative inverse element generation module over GF(2³).

4. 分配器module 構成

이제 앞에서 구한 加算器module, 乘算器module 및 除算器module을 合成하여 最終의 GF(2³)上의 算術演算器시스템을 構成하기 위해 分配器 module D1 및 D2를 Ⅳ章의 內容과 그림9 및 10을 토대로하여 構成하면 各各 다음 그림16, 17과 같다.

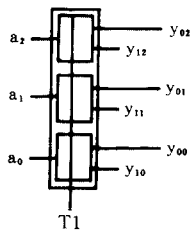


그림16. GF(2³)上의 分配器 module D1
Fig. 16. A distributor module D1 over GF(2³).

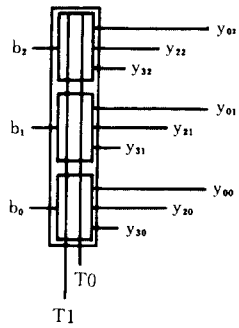


그림17. GF(2³)上의 分配器module D2
Fig. 17. A distributor module D2 over GF(2³).

5. GF(2³)上의 算術演算器시스템 構成

GF(2³)上의 算術演算器시스템을 構成하기 위해 지금까지 構成한 加算器module, 乘算器module, 除算器module을 分配器module D1과 D2로 合成하면 다음그림18과 같다.

Ⅵ. 結 論

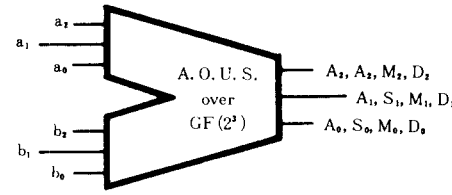
本 論文에서는 有限体인 Galois体 중에서 P=2인 GF(2ᵐ)上의 算術演算器시스템 構成 方法중의 한가지를 提案하였다.

本 論文에서 提示한 算術演算器시스템의 構成 節次는 다음과 같다.

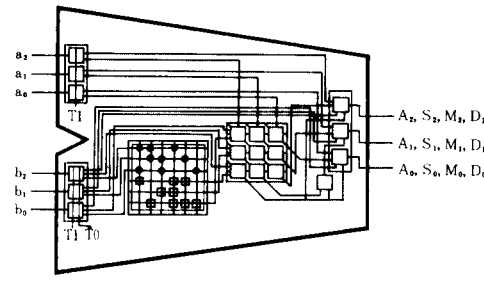
[節次 1] Galois体의 数学的 性質로 부터 四則演算인 加減乘除의 알고리듬을 各各 구한다.

[節次 2] 이들 알고리듬을 토대로 各各 加算器 基本cell(A-cell)과 乘算器 基本cell(M-cell)을 構成한다.

[節次 3] 이들 基本cell을 사용하여 加算器module, 乘算器module 및 除算器module을 構成한다.



(a) symbol of A. O. U. S. over GF(2³),



(b) internal structure of A. O. U. S. over GF(2³).

그림18. GF(2³)上의 算術演算器시스템
Fig. 18. Arithmetic Operation Unit Systems over GF(2³).

[節次 4] 節次3에서 구한 各各의 module을 合成하기 위하여 먼저 分配器module 構成時 心要한 基本cell인 D1-cell과 D2-cell을 構成한 후 이들을 사용하여 分配器module D1과 D2를 構成한다.

[節次 5] 節次4의 分配器module과 節次3의 算術演算器module들을 合成하여 最終 GF(2ᵐ) 上의 算術演算器시스템을 構成한다.

한편, 本 論文에서 提案한 算術演算器시스템은 다음과 같이 6개의 module로써 構成된다.

1. 加算器module
2. α^r生成module
3. 制御入力 C_L生成module
4. 乘法逆元生成module
5. 分配器module D1
6. 分配器module D2

또한, 乘算器module과 除算器module은 各各 다음과 같은 module들을 合成함으로써 構成된다.

乘算器module=(α^r生成module)+(制御入力C_L生成module)+(加算器module)

除算器module=(乘法逆元生成module)+(乘算器module)

위에서 본바와 같이 加算器module은 加算, 減算, 乘

算, 除算의 어떤 演算을 하더라도 항상 사용된다.

그리고 加算器module內의 A-cell갯수는 m개, α^r生成module內의 M-cell갯수는 m^2개이고 分配器 module D1과 D2內의 D1-cell과 D2-cell갯수는 各各 m개이다.

本 論文에서 提案한 GF(2^m)上의 算術演算器시스템의 特徵을 要約하면 다음과 같다.

提案한 算術演算器시스템은 module들의 合成으로 構成되므로 m의 擴張에 따른 算術演算器시스템은 各 module을 m에 따라 擴張만 하면 構成할 수 있다.

또한, 加算器module은 A-cell로써, α^r生成module은 M-cell로써 構成되므로 역시 m의 擴張에 따른 加算器module과 α^r生成module을 쉽게 構成할 수 있다.

한편, 乘算器module 構成時의 制御入力C_L生成module도 GF(2^m)上의 原始既約多項式으로 부터 간단히 구할 수 있다. 그리고 除算器module 構成은 乘法逆元生成多項式 $[G(\alpha)]^{-1}$만 구하여 乘法逆元生成module을 構成한후 이를 乘算器module의 入力으로 사용하여 構成할 수 있다. 마지막으로, 最終 GF(2^m)上의 算術演算器시스템은 分配器module들로써 合成하여 쉽게 構成할 수 있다.

以上의 內容을 綜合하면 現在 사용하고 있는 디지탈시스템 및 스위칭理論을 그대로 適用시킬 수 있으므로 기존의 컴퓨터를 GF(2^m)上에서 m의 擴張에 따른 多値컴퓨터로 應用할 수 있다고 展望된다. 또한 論現演算器시스템(Logical Operation Unit Systems)을 構成해 本 論文에서 提示한 算術演算器시스템과 合成한다면 多値컴퓨터의 算術·論理演算器 시스템(Arithmetic & Logical Operation Unit Systems)을 構成할 수 있으리라 展望 및 硏究가 기대되며 序論에서 언급한 여러가지 分野에도 應用될 수 있으리라 생각된다. 그리고 本 論文에서 提案한 GF(2^m) 上의 算術演算器시스템은 기존의 素子로써 쉽게 構成할 수 있으며 P≠2인 경우에도 쉽게 擴張할 수 있으리라 사료된다.

參 考 文 獻

[1] K.C. Smith, "The prospects for multivalued: a technology and applications view," IEEE Trans. Compt., vol. C-30, pp. 619-634, Sep. 1981.

[2] E.J. McCluskey, "A discussion of multiple valued logic circuits," The 12th Int. Symp. on Multiple-Valued Logic, pp. 200-205, Paris, France, 25-27, May. 1982.

[3] S.L. Hurst, "Multiple-valued-its status and its future," IEEE Trans. Compt., vol. C-33, pp. 1160-1179, Dec. 1984.

[4] K.C. Smith, "Multiple-valued logic: a tutorial and appreciation," Computers, pp. 17-27, Apr. 1988.

[5] G. Birkhoff and T.C. Bartee, Modern Applied Algebra, McGraw-hill book company, N.Y., 1970.

[6] E. Artin, Galois Theory, NAPCO Graphic arts, Inc., Wisconsin. 1971.

[7] J.B. Fraleigh, A First Course in Abstract Algebra, Addison-Wesley publishing Company, Inc., Philippines. 1982.

[8] R. Lidi and G. Pilz, Applied Abstract Algebra, Spring-Verlag, Inc., N.Y., 1984.

[9] 金興壽, 朴春明, "Bit Code 割當에 依한 GF(2^m)上의 多値論理函數 構成 理論," 1986年 5月 大韓電子工學會 論文誌 第23卷 第3號. pp. 295-308.

[10] 朴春明, 金興壽, "GF(2^m)上의 多値論理順次머시인 構成 理論," 1987年 9月 大韓電子工學會 論文誌 第24卷 第5號, pp. 823-832.

[11] I.F. Blake, Algebraic Coding Theory: History and Development, Down, Hutchinson & Ross, Inc., Stroudburg, Pennsylvania, 1973.

[12] B. Benjauthrit and I.S. Reed, "Galois switching functions and their applications," IEEE Trans. Compt., vol. C-25, pp. 78-86, Jan. 1976.

[13] D.K. Pradhan, "A theory of galois switching functions," IEEE Trans. Compt., vol. C-27, pp. 239-248, May. 1978.

[14] H.S. Kim, "A construction of multiple-valued switching functions by galois field," Ph.D. dissertation, Inha Univ., Inchon, Korea.

[15] I. Takahashi, "Switching functions constructed by galois fields," Inform. Contr., vol. 48, pp. 95-108, Jan. 1981.

[16] S. Lin and D.J. Costello, Error Control Coding, Prentice-Hall, Inc., 1983.

[17] R.E. Blahut, Fast Algorithms for Digital Signal Processing, Addison-Wesley Publishing Company, Inc., 1985.

[18] B.A. Laws and C. K. Rusforth, "A cellular-array multiplier for GF(2^m)," IEEE Trans. Compt., short-notes, pp. 1573-1578, Dec. 1971.

[19] C.S. Yeh, I.S. Reed and T.K. Trung, "Systolic multiplier for finite fields GF (2^m)," IEEE Trans. Compt., vol. C-33,

pp. 357-360, Apr. 1984.

[20] C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch,. J.K. Omura and I.S. Reed, "VLSI architecture for computing multiplications and inverse in GF(2^m),' IEEE Trans. Compt., vol. C-34, pp. 709-717, Aug. 1985.

[21] 박춘명, 김홍수 外, "GF(3^m)상의 승산기 구성 이론," 1988年度 大韓電子工學會 秋季綜合學術大會 論文集 vol. 11 no. 1, 1988年 11月 26日, pp. 329－333.

[22] 박춘명, 김홍수外, "GF(3^m)상의 역원생성기 구성 이론," 1989年度 大韓電子工學會 夏季綜合學術大會 論文集 vol. 12. no. 1, 1989年7月7日 pp, 436－439.

[23] 朴春明, 金興壽 外, "GF(3^m)上의 乘算器 및 逆元生成器 構成," 1990年 5月 大韓電子工學會 論文誌 第27卷 第5號, pp. 103－112.

[24] M.D. Ercegovac and T. Lang, Digital Systems and hardware/firmware algorithms, John Wiley & Sons, Inc., Canada, 1985.

[25] K. Bromley, Sun-Yuan Kang and E. Swartzlander, International Conference on SYSTOLIC Arrays, Computers Society Press, N.Y., 1985.

[26] S.Y. Kung, VLSI ARRAY PROCESSORS, Prentice-Hall, Inc., 1988.

著 者 紹 介

朴 春 明(正會員)

1955年 12月 4日生. 1983年 2月 인하대학교 전자공학과 졸업(공학사). 1986年 2月 인하대학교 대학원 전자공학과 졸업(공학석사). 1986年 9月～현재 인하대학교 대학원 전자공학과 박사학위 과정중. 주관심분야는 다치논리이론·함수구성 및 회로 설계, 컴퓨터구조 및 VLSI설계, Coding Theory, DSP & DIP 등임.

金 興 壽 (正會員) 第27卷 第5號 參照
현재 인하대학교 전자공학과 교수