

# De Bruijn 그래프에 기초한 다중처리기구성

박춘명\*

\* 충주대학교 전기전자 및 정보공학부 컴퓨터공학전공

## A Construction of Multiple Processing based on De Bruijn Graph

Chun-Myoung Park\*

\* School of Electrical Electronic & Information Engineering, Computer Engineering Major, Chungju National University

E-mail : cmpark@gukwon.chungju.ac.kr

### 요 약

본 논문에서는 De Bruijn 그래프에 기초한 다중처리기구성의 한 가지 방법을 제안하였다. 제안한 방법에서는 유한체상의 수학적 성질과 그래프의 성질을 사용하여 변환연산자를 제한하였으며, 이들 변환연산자를 이용하여 De Bruijn 그래프의 변환표를 도출하였다. 그리고, 이 변환표로부터 유한체상의 De Bruijn 그래프를 도출하였다. 제안한 다중처리기는 유한체상의 임의의 소수와 양의 정수에 대해 구성할 수 있으며 고장허용컴퓨팅시스템, 파이프라인 시스템, 병렬처리 네트워크, 스위칭 함수와 이의 회로, 차세대 디지털논리시스템 및 컴퓨터구조 중의 하나인 다차디지털논리시스템 등에 적용할 수 있으리라 전망된다.

### 1. 서론

최근에 멀티미디어 H/W와 S/W에 기반을 둔 여러 분야가 매우 급속도로 발전되고 있으며 21C에는 더욱 더 활용 및 적용이 요구될 것이다. 특히 멀티미디어 H/W는 지금까지의 각종 데이터 처리보다는 훨씬 방대한 데이터량, 최적의 데이터 압축 및 복원, 초고속 전송, 암호화 및 복호화 등의 복합적이고 고기능의 기술이 요구되고 있다.[1-4]

따라서 본 논문에서는 이를 효과적으로 해결할 수 있는 방법으로 유한체상의 De Bruijn 그래프에 기초를 둔 다중처리기의 한가지 방법을 제안하였으며, De Bruijn 그래프를 바탕으로 한 병렬처리구조에는 Linear array, Ring, CBT(Complete Binary Tree), TM(Tree Machine), SE(Suffle-Exchange), CCC(Cube Connected Cycle) 등과 같은 구조가 소개되었으며 이를 고장허용컴퓨팅시스템에 적용하는 염구도 병행되었다.

이들 연구들은 유한체상에서  $P=2$ 인  $GF(2^m)$ 에 기초를 둔 2진(2치)논리 De Bruijn 그래프(BDBG : Binary De

Bruijn Graph)를 바탕으로 한 처리기들이며 이 BDBG는 pipe-line 계층, Multiplex 계층, ASCEND와 DESCEND 계층등 과 같은 응용분야에 사용되며 이들 형태는 SISO(Serial Input Serial Output) 형태, SIPO(Serial Input Parallel Output) 형태, PISO(Parallel Input Serial Output) 형태, PIPO(Parallel Input Parallel Output) 형태 및 HIHO(Hybrid Input Hybrid Output) 형태로 크게 분류할 수 있다.

따라서 본 논문에서는 유한체의 수학적 성질과 그래프의 성질로부터 유한체  $GF(P^m)$ 상의 De Bruijn 그래프를 구성하는 방법을 제안하였으며 이를 다중처리기구성에 적용하는 한가지 방법을 제안하였다.

본 논문의 서술과정은 다음과 같다.

2장에서는 본 논문을 전개하는데 필요한 수학적 성질과 그래프의 성질에 대해 논의하였으며, 3장에서는 2장에서서의 내용을 토대로 유한체  $GF(P^m)$ 상의 De Bruijn 그래프를 구성하기 위한 각종 변환연산자에 대해 논의하였다.

그리고 4장에서는 3장의 내용을 바탕으로 유한체  $GF(P^m)$ 상의 De Bruijn 그래프의 변환표를 구성하였으며

5장에서는 4장의 내용을 기초로 하여 최종 GF(P<sup>m</sup>)상의 De Bruijn 그래프에 바탕을 둔 다중처리기의 특징을 기술하였으며 향 후 연구과제로서 이를 적용하는 분야의 연구와 앞으로의 전망을 기술하였다.

## 2. 유한체의 수학적 성질과 그래프의 성질

본 장에서는 본 논문을 전개하는데 필요한 유한체상의 중요한 수학적 성질과 그래프의 성질에 대해 논의한다.

### 2-1. 유한체의 수학적 성질

유한체란 임의의 소수 P와 양의 정수 m에 대하여 P<sup>m</sup>개의 원소를 가지는 유일한 체를 말하며 Galois(Galois, Evariste 1811-1832 : 프랑스의 천재적 수학자)가 발견하여 일명 Galois체(Galois Field)라 불리우고 GF(P<sup>m</sup>)으로 표시한다. 이때 P<sup>m</sup>을 유한체의 위수(order)라고 한다.

한편, 유한체는 기초체(ground field) GF(P)와 이를 m차 확대한 확대체(extension field) GF(P<sup>m</sup>)으로 나눌 수 있으며, 기초체 GF(P)의 P는 1보다 큰 소수로써 GF(P)상의 원소는 {0,1,2, ..., P-1}이며, 확대체 GF(P<sup>m</sup>)은 GF(P)상의 m차 벡터공간(Vector Space)으로 표시할 수 있다.

일반적으로 유한체는 {S, +, •, 0, 1}의 5가지 요소로 표시되며 S는 원소들의 집합이고 +와 •는 S상의 이항연산(binary operation)이며 0과 1은 각각 가산과 승산에 대한 항등원이며 모든 산술연산은 modP로 처리된다.

한편 유한체 GF(P<sup>m</sup>)상의 원소를 생성 하는데는 기초체 GF(P)의 존재를 만족하는 특성방정식을 인수분해하여 m차 기약다항식을 구하고 이를 0으로 하는 한 원시근을 α로 할 때 원시기약다항식(Primitive irreducible polynomial)을 얻을 수 있으며 이로부터 유한체의 원소들을 생성시킬 수 가 있다.

한편, 유한체의 GF(P<sup>m</sup>) 내의 원소들은 수학적 성질로부터 계수 a<sub>i</sub>는 a<sub>i</sub> ∈ GF(P)(i=0, 1, 2, ..., m-1)이며 이는 P치를 나타내는 Digit Code로써 표현 할 수 있다. 따라서 모든 계수들을 Digit Code로써 할당함으로써 GF(P<sup>m</sup>) 내의 모든 원소들을 표시할 수 있다.

일반적으로 GF(P<sup>m</sup>)상의 원소들을 P 치 Digit Code로 할당하는 종류는 P<sup>m</sup>! 개이다. 이때 GF(P<sup>m</sup>)상의 원소들을 P 치 Digit Code로 할당하는데 필요한 Digit Code 개수는 m개이고 최고차수계수인 a<sub>m-1</sub>을 MSD(Most Significant Digit), 최저차수계수인 a<sub>0</sub>을 LSD(Least Significant Digit)라 한다.

특히, P=2인 경우의 GF(2<sup>m</sup>)상에서의 원소들은 Bit Code(Binary Digit Code)로 할당할 수 있고 최고차수 계수인 a<sub>m-1</sub>을 MSB(Most Significant Bit), 최저차수 계수인 a<sub>0</sub>을 LSB(Least Significant Bit)라 한다.

본 논문에서 채택한 GF(P<sup>m</sup>)내의 원소들을 Digit Code로 할당하는 알고리즘은 다음과 같다.

[유한체 GF(P<sup>m</sup>)내의 모든 원소들을 Digit Code로 할당하는 알고리즘]

Step 1 : 원소 e<sub>0</sub>는 Digit Code 0으로 한다.

Step 2 : 원소 e<sub>K-1</sub>은 Digit Code를 모두 (P-1)로 한다.(여기서, K=P<sup>m</sup>)

Step 3 : LSD를 1에서부터 (P-1)까지 증가시키면서 할당한다.

Step 4 : Step3의 방법을 계속해서 LSD로부터 한 Digit씩 좌측으로 시프트(shift) 시키면서 할당한다.

Step 5 : Step4를 MSD까지 계속한다.

예) GF(3<sup>2</sup>)상의 원소들을 제안한 알고리즘에 의하여 Digit Code로 할당하면 다음과 같다. 이경우는 P=3이고 m=3이므로 원소 rpt수는 9개이고 이들 원소들을 3치 Digit Code로 할당하는데 필요한 Digit Code 개수는 2개이며 앞의 알고리즘에 의하면 다음 표2-1과 같다.

표 2-1. GF(3<sup>2</sup>)상의 원소들을 Digit Code로 할당  
Table 2-1. Digit code assignment of elements over GF(3<sup>2</sup>).

Elements	Coefficient		Step
	MSD a <sub>i</sub>	LSD a <sub>0</sub>	
e <sub>0</sub>	0	0	Step 1
e <sub>1</sub>	0	1	Step 3 Step 4 및 Step 5
e <sub>2</sub>	0	2	
e <sub>3</sub>	1	0	
e <sub>4</sub>	1	1	
e <sub>5</sub>	1	2	
e <sub>6</sub>	2	0	Step 2
e <sub>7</sub>	2	1	
e <sub>8</sub>	2	2	

이 외에 유용한 수학적 성질은 다음의 참고문헌을 참조하였다.[5-7]

### 2-1. 그래프의 성질

일반적으로 그래프는 다음 식(2-1)과 같이 표현한다.

$$G(V, E) \tag{2-1}$$

여기서, V는 유한개의 비공집합(nonempty set) 노드(node)의 집합이고, E는 노드 집합에서의 2개의 부집합(subset)의 모서리(edge)의 집합이다.

또한, |V|는 그래프의 위수(order)라 하며 노드의 개수이고, |E|는 그래프의 크기(size)라 하며 모서리의 개수이다.

그리고, 노드 V에 인접한 모든 노드의 집합을 노드 V

의 이웃(neighbour)이라 하고  $N(V)$ 로 표시하며 노드  $V$ 에 대해 일어나는 모서리의 개수를 노드  $V$ 의 깊이(degree)라 하며  $\deg(V)$ 로 표현한다. 특히 그래프  $G$ 에서 일어나는 경우는  $\deg_G(V)$ 로 표시한다.

한편, 오직 사이클릭(cyclic)을 구성하는 위수  $n$ 의 그래프를 사이클릭  $C_n$ 으로 표시하고, 사이클릭이 없는 그래프인 경우를 비사이클릭(acyclic)이라 하며, 트리는 연결된 비사이클릭 그래프의 일종이다.

본 논문에서는 유한체  $GF(P^m)$ 상의 De Bruijn 그래프는 쌍방향성이며 사이클릭 성질을 갖는 그래프이다.

이 외에 그래프의 유용한 성질은 다음의 참고문헌을 참조하였다.[8,9]

### 3. 유한체 $GF(P^m)$ 상의 De Bruijn 그래프를 구성하기 위한 변환연산자

본 논문에서는 2장의 내용에 바탕을 두고 유한체  $GF(P^m)$ 상의 De Bruijn 그래프 구성을 위한 변환연산자에 대해 논의한다.

#### 3-1. 일반 변환연산자

- [GOP 1] LR[*digit stream*] : one digit left rotate  
 $LR[D_{m-1}D_{m-2} \dots D_1D_0]=[D_{m-2}D_{m-3} \dots D_1D_0D_{m-1}]$
- [GOP2] RR[*digit stream*] : one digit right rotate  
 $RR[D_{m-1}D_{m-2} \dots D_1D_0]=[D_0D_{m-1}D_{m-2}D_{m-3} \dots D_1]$
- [GOP3] CLR[*digit stream*] : left rotate of modP complement for MSD  
 $CLR[D_{m-1}D_{m-2} \dots D_1D_0]=[D_{m-2}D_{m-3} \dots D_1D_0D_{m-1}']$
- [GOP4] CRR[*digit stream*] : right rotate of modP complement for LSD  
 $CRR[D_{m-1}D_{m-2} \dots D_1D_0]=[D_0'D_{m-1}D_{m-2}D_{m-3} \dots D_1]$

위 일반연산자에서  $D_i \in \{0,1, \dots, P-1\}$ 이고  $D_i'$ 의 modP 보수(complement)이다.

#### 3-2. modP 변환연산자

- [MOP1] LR(K)modP[*digit stream*] : one digit left rotate of modP(MSD+K)  
 $LR(K)modP[D_{m-1}D_{m-2} \dots D_1D_0]=[D_{m-2}D_{m-3} \dots D_1D_0modP(D_{m-1}+K)]$
- [MOP2] RR(K)modP[*digit stream*] : one digit right rotate of modP(MSD+K)  
 $RR(K)modP[D_{m-1}D_{m-2} \dots D_1D_0]=[modP(D_0+K)D_{m-1}D_{m-2} \dots D_2D_1]$
- [MOP3] CLR(K)modP[*digit stream*] : left rotate of modP complement for (MSD+K)

$$CLR(K)modP[D_{m-1}D_{m-2} \dots D_1D_0]=[D_{m-2} \dots D_2D_1D_0(D_{m-1}+K)']$$

- [MOP4] CRR(K)modP[*digit stream*] : right rotate of modP complement for (LSD+K)  
 $CRR(K)modP[D_{m-1}D_{m-2} \dots D_1D_0]=[(D_0+K)' D_{m-1}D_{m-2} \dots D_2D_1]$

위 변환연산자에서  $D_i \in \{0,1, \dots, P-1\}$ 이고  $D_i'$ 는  $D_i$ 에 대한 보수이다. 또한,  $0 \leq K \leq P-1$ (K=integer)이다.

#### 3-3. 역 변환연산자

3-1절의 일반연산자에서 LR[*digit stream*] 변환연산자와 RR[*digit stream*] 변환연산자는 서로 역 변환 관계에 있으며, CLR[*digit stream*] 변환연산자와 CRR[*digit stream*] 변환연산자는 서로 역 변환 관계에 있다.

또한, 3-2절의 modP 변환연산자에서 LR(K1)modP[*digit stream*] 변환연산자와 RR(K2)modP[*digit stream*] 변환연산자는 서로 역 변환 관계에 있으며, CLR(K1)modP[*digit stream*] 변환연산자와 CRR(K2)modP[*digit stream*] 변환연산자는 서로 역 변환 관계에 있으며 이때  $K1+K2=P$ 의 조건을 갖는다.

즉, 위의 내용을 정돈하면 다음과 같다.

$$\begin{aligned} LR[*digit stream*]]^{-1} &= RR[*digit stream*]] \\ RR[*digit stream*]]^{-1} &= LR[*digit stream*]] \\ CLR[*digit stream*]]^{-1} &= CRR[*digit stream*]] \\ CRR[*digit stream*]]^{-1} &= CLR[*digit stream*]] \\ LR(K1)modP[*digit stream*]]^{-1} &= RR(K2)modP[*digit stream*]] \\ &= RR(K2)modP[*digit stream*]]^{-1} \\ &= LR(K1)modP[*digit stream*]] \\ CLR(K1)modP[*digit stream*]]^{-1} &= CRR(K2)modP[*digit stream*]] \\ &= CRR(K2)modP[*digit stream*]]^{-1} \\ &= CLR(K1)modP[*digit stream*]] \end{aligned}$$

#### 4. De Bruijn 그래프의 인접행열과 변환표

본 장에서는 앞의 3장의 내용을 토대로 De Bruijn 그래프[10-12]를 구성하기 위해 먼저 유한체  $GF(P^m)$ 상의 인접행열을 도출하고 이를 바탕으로 De Bruijn 변환표를 구성한다.

인접행열에서의 loop는 자기 자신의 원소를 의미하

며, 1은 해당 행과 열의 원소가 서로 모서리로 연결됨을 의미하며, 0은 해당 행과 열의 원소가 서로 연결되지 않음을 나타낸다.

**4-1. 인접행렬 구성**

예를 들어  $GF(3^m)$ 에서  $m$ 이 2인  $GF(3^2)$ 상의 De Bruijn 그래프의 인접행렬을 구성하면 다음 표4-1과 같다.

표 4-1.  $GF(3^2)$ 상의 De Bruijn 그래프의 인접행렬  
Table 4-1. The De Bruijn adjacent matrix over  $GF(3^2)$ .

	(00)	(01)	(02)	(10)	(11)	(12)	(20)	(21)	(22)
(00)	loop	1	1	1	0	0	1	0	0
(01)	1	0	0	1	1	1	1	0	0
(02)	1	0	0	1	0	0	1	1	1
(10)	1	1	1	0	1	0	0	1	0
(11)	0	1	0	1	loop	1	0	1	0
(12)	0	1	0	0	1	0	1	1	1
(20)	1	1	1	0	0	1	0	0	1
(21)	0	0	1	1	1	1	0	0	1
(22)	0	0	1	0	0	1	1	1	loop

그리고,  $GF(5^m)$ 에서  $m$ 이 1인  $GF(5)$ 상의 De Bruijn 그래프의 인접행렬을 구성하면 다음 표4-2와 같다.

표 4-2.  $GF(5)$ 상의 De Bruijn 그래프의 인접행렬  
Table 4-2. The De Bruijn adjacent matrix over  $GF(5)$ .

	(0)	(1)	(2)	(3)	(4)
(0)	loop	1	1	1	
(1)	1	loop	1	1	1
(2)	1	1	loop	1	1
(3)	1	1	1	loop	1
(4)	1	1	1	1	loop

**4-2. De Bruijn 변환표**

본 절에서는 앞의 4-1절의 내용을 토대로 유한체  $GF(P^m)$ 상의 De Bruijn 변환표를 구하면 다음과 같다.

$GF(3^m)$ 에서  $m$ 이 2인  $GF(3^2)$ 상의 De Bruijn 변환표를 구성하면 다음 표4-3과 같으며, 유한체  $GF(3^m)$ 에서의  $LR(0) \bmod 3$  변환연산자와  $RR(0) \bmod 3$  변환연산자의 결과는 동일하다.

또한,  $GF(5^m)$ 에서  $m$ 이 1인  $GF(5)$ 상의 De Bruijn 그래프의 인접행렬을 구성하면 다음 표4-4와 같다.

표 4-3.  $GF(3^2)$ 상의 De Bruijn 변환표  
Table 4-3. The De Bruijn transformation table over  $GF(3^2)$ .

	(00)	(01)	(02)	(10)	(11)	(12)	(20)	(21)	(22)
$LR(0) \bmod 3$	(00)	(10)	(20)	(01)	(11)	(21)	(02)	(12)	(22)
$RR(0) \bmod 3$	(00)	(10)	(20)	(01)	(11)	(21)	(02)	(12)	(22)
$LR(1) \bmod 3$	(01)	(11)	(21)	(02)	(12)	(22)	(00)	(10)	(20)
$RR(1) \bmod 3$	(10)	(20)	(00)	(11)	(21)	(01)	(12)	(22)	(02)
$LR(2) \bmod 3$	(02)	(12)	(22)	(00)	(10)	(20)	(01)	(11)	(21)
$RR(2) \bmod 3$	(20)	(00)	(10)	(21)	(01)	(11)	(22)	(02)	(12)

표 4-4.  $GF(5)$ 상의 De Bruijn 변환표  
Table 4-4. The De Bruijn transformation table over  $GF(5)$ .

	(0)	(1)	(2)	(3)	(4)
$LR(0) \bmod 5$	(0)	(1)	(2)	(3)	(4)
$RR(0) \bmod 5$	(0)	(1)	(2)	(3)	(4)
$LR(1) \bmod 5$	(1)	(2)	(3)	(4)	(0)
$RR(1) \bmod 5$	(1)	(2)	(3)	(4)	(0)
$LR(2) \bmod 5$	(2)	(3)	(4)	(0)	(2)
$RR(2) \bmod 5$	(2)	(3)	(4)	(0)	(2)
$LR(3) \bmod 5$	(3)	(4)	(0)	(1)	(3)
$RR(3) \bmod 5$	(3)	(4)	(0)	(1)	(3)
$LR(4) \bmod 5$	(4)	(0)	(1)	(2)	(4)
$RR(4) \bmod 5$	(4)	(0)	(1)	(2)	(4)

**5. De Bruijn 그래프 구성**

본 장에서는 3장과 4장의 내용을 토대로 유한체  $GF(P^m)$ 상의 De Bruijn 그래프를 구성하면 다음과 같다.

$GF(2^4)$ 와  $GF(3^2)$ 상의 De Bruijn 그래프 구성은 각각 다음 그림 5-1과 5-2와 같다. 그리고  $GF(3^3)$ 상의 De Bruijn 그래프 구성은 그림 5-3과 같으며  $GF(5)$ 상의 De Bruijn 그래프는 그림 5-4와 같다.

이 외의 임의의  $P$ 와  $m$ 에 대해서도 앞에서 논의한 2장의 유한체  $GF(P^m)$ 내의 Digit Code 할당 알고리즘과 3장의 변환연산자, 4장의 인접행렬과 변환표로부터 쉽게 구성할 수 있다.

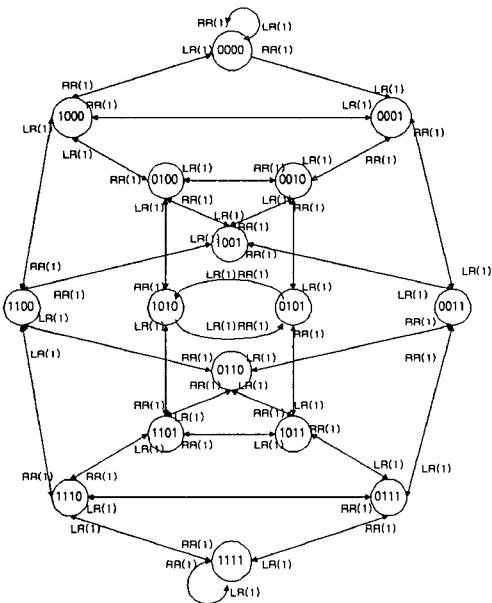


그림 5-1.  $GF(2^4)$ 상의 De Bruijn 그래프 구성  
Fig. 5-1. The De Bruijn graph construction over  $GF(2^4)$ .

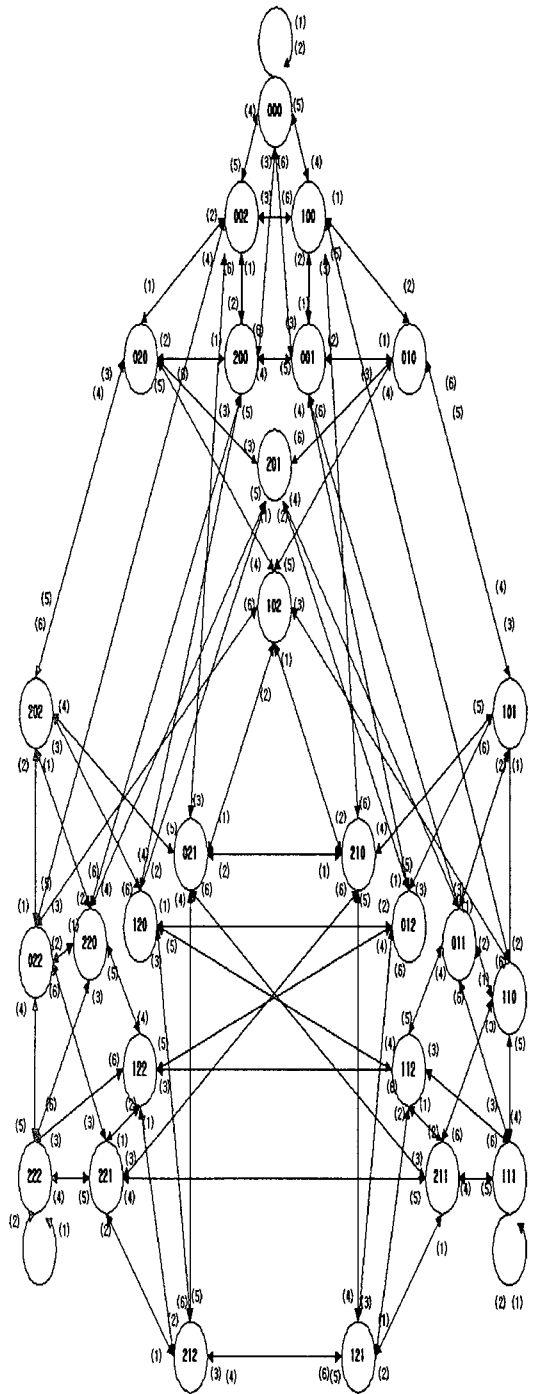


그림 5-3.  $GF(3^3)$ 상의 De Bruijn 그래프 구성  
Fig. 5-3. The De Bruijn graph construction over  $GF(3^3)$ .

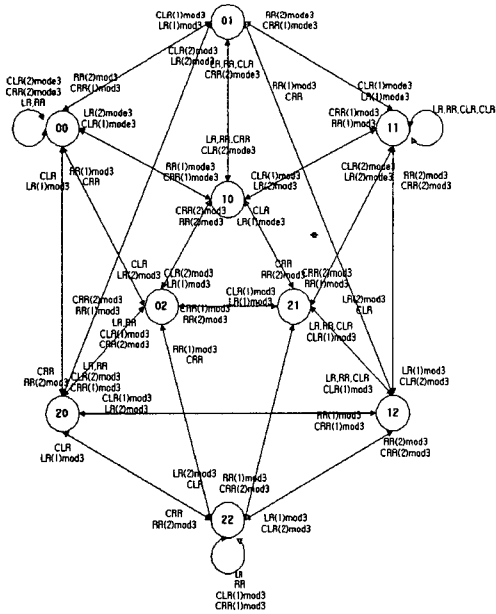


그림 5-2.  $GF(3^2)$ 상의 De Bruijn 그래프 구성  
Fig. 5-2. The De Bruijn graph construction over  $GF(3^2)$ .

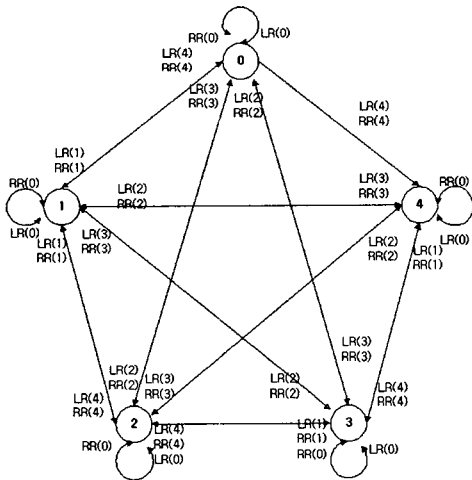


그림 5-4. GF(5)상의 De Bruijn 그래프 구성  
Fig. 5-4. The De Bruijn graph construction over GF(5).

6. 결론

본 논문에서는 유한체 GF(P<sup>m</sup>)상의 De Bruijn 그래프를 구성하거 이를 다중처리기 구성에 적용하는 한 가지 방법을 제안하였다.

지금까지는 P=2인 GF(2<sup>m</sup>)상의 2진(2차)논리 De Bruijn 그래프(BDBG)를 구성하여 이를 처리기 구성에 적용하였다.

본 논문에서 제안한 유한체 GF(P<sup>m</sup>)상의 다중처리기는 P가 어떤 값이건 De Bruijn 그래프를 구성할 수 있는 특징을 가진다.

제안한 GF(P<sup>m</sup>)상의 De Bruijn 그래프를 바탕으로 한 다중처리기는 차세대 디지털논리시스템 중의 하나인 다치논리시스템 구성시에 원소들의 자유로운 변환과 스위칭회로에 적용할 수 있으리라 사료되며, 향후 연구과제로는 이를 바탕으로한 유한체 GF(P<sup>m</sup>)상의 조합논리시스템과 순차논리시스템의 구성에 관한 연구가 기대되며 현재 연구진행 중에 있다.

또한, 제안한 다중처리기는 다치논리시스템을 채택한 컴퓨터시스템에서 고장허용컴퓨팅시스템(fault-tolerant computing system)에 적용할 수 있으리라 기대된다.[13]

그리고 향후 각종 진보된 임베이드컴퓨터시스템 [14-15]과 멀티미디어시스템에 적용 및 응용을 이용할 수 있으리라 전망된다.

참고문헌

- [1] Murphy, *Multimedia Devices*, Prentice-Hall, 1999.
- [2] A. Karm, *Multimedia Modeling*, World science, 1998.
- [3] C.M.Park, "A Study on Constructing Highly Adder/Multiplier Systems over Galois Fields," ITC-CSCC2000, vol.1, pp.318-321, 11-13 July 2000.
- [4] C.M.Park etc., "A Construction of Multimedia Hardware Arithmetic Devices," DCS-2000, vol.1, pp.230-234, 9 Dec. 2000.
- [5] I.F.Blake, *Algebraic Coding Theory:History and development*, Down, Hutchinson & Ross,Inc., Stroudburg,Pennsylvania,1973.
- [6] R.Lidi and G.Piltz, *Applied Abstract Algebra*, Springer-Verlag, Inc., 1984.
- [7] E.Artin, *Galois Theory*,NAPCO Graphic arts, Inc.,Wisconsin.1971.
- [8] R.Gould, *Graph Theory*, The Bejamin/Cummings Publishing Company, Inc., 1988.
- [9] M.Gondran and M.Minoux, *Graph and Algorithms*, A Wiley-interscience publication, Toronto, 1984.
- [10] B.Arazi, "On the Sythesis of De-Bruijn Sequences", INFORMATION and CONTROL 49., pp.81-90, 1981.
- [11] A.Lempel, "On a Homomorphism of the DE-Bruijn Graph and its Applications to the Design of Feedback Shift Registers", IEEE Trans. Comut. Vol.C-19, NO12, pp.1204-1209, Dec. 1970.
- [12] M.A.Sridhar and C.S.Reghavendra, "Fault-Tolerant Networks based on the De Bruijn Graph", IEEE Trans. Comut. Vol.41,NO.11, pp.1167-1174, Oct. 1991.
- [13] M.D.Ercegovac and T.Lang, *Digital Systems and Hardware/Firmware Algorithms*, John Wiley & Sons, Inc., Canadae, 1985.
- [14] F. Balarin et al., *Hardware-Software Co-design of Embedded Systems: The Polis Approach*, Kluwer Academic Press, Boston, June 1997.
- [15] R.K.Gupta, *Co-Synthesis of Hardware and Software for Digital Embedded Systems*, vol. 329, Kluwer Academic Publishers, Boston, Aug. 1995.