
De Bruijn 그래프에 의한 다중처리기 구성

박 춘 명*

Construction of the Multiple Processing Unit by De Bruijn Graph

Chun-Myoung Park*

요 약

본 논문에서는 De Bruijn 그래프에 기초한 다중처리기 구성 방법에 대해 논의하였다. 유한체 상의 수학적 성질과 그래프의 성질을 사용하여 변환연산자에 대해 논의하였으며, 이들 변환연산자를 이용하여 De Bruijn 그래프의 변환표를 도출하였다. 그리고, 이 변환표로부터 유한체 상의 De Bruijn 그래프를 도출하였다. 제안한 다중처리기는 유한체 상에서의 임의의 소수와 양의 정수에 대해 구성할 수 있으며 고장허용컴퓨팅시스템, 파이프라인 시스템, 병렬처리 네트워크, 스위칭 함수와 이의 회로, 차세대 디지털논리시스템 및 컴퓨터구조 등에 적용할 수 있다.

ABSTRACT

This paper presents a method of constructing the universal multiple processing element unit(UMPEU) by De Bruijn Graph. The proposed method is as following. First, we propose transformation operators in order to construct the De Bruijn graph using properties of graph. Second, we construct the transformation table of De Bruijn graph using above transformation operators. Finally we construct the De Bruijn graph using transformation table. The proposed UMPEU be able to construct the De Bruijn graph for any prime number and integer value of finite fields. Also the UMPEU is applied to fault-tolerant computing system, pipeline class, parallel processing network, switching function and its circuits.

키워드

Graph Theory, De Bruijn Graph, Transformation Operator, Adjacent Matrix,
Fault-Tolerance Computing System, Switching Function etc.

I. 서 론

최근에 각종 디지털시스템 및 컴퓨터 관련 하드웨어는 매우 급속도로 발전되고 있으며 향후 더욱 더 가속화 될 것이다.

디지털시스템 관련 하드웨어는 지금까지의 각종 데이터 처리보다는 훨씬 방대한 데이터 량, 최적의 데이터 압축 및 복원, 초고속 전송, 암호화 및 복호화 등의 복합적이고 고기능의 기술이 요구되고 있다.^[1-3] 따라서 본 논문에서는

이를 효과적으로 해결할 수 있는 방법으로 유한체상의 De Bruijn 그래프^[6-7]에 기초를 둔 다중처리기 구성에 대해 논의 하였다.

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 연구도 병행되고 있다. 이들 연구들은 유한체 상에서 $P=2$ 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 계층 등 과 같은 응용분야에 사용된다. 또한, 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. 본 논문에서는 유한체 상의 수학적 성질과 그래프의 성질로부터 유한체 $GF(P^m)$ 상의 De Bruijn 그래프를 구성하는 방법을 제안하였으며 이를 다중처리기구성에 적용하는 방법에 대해 논의 하였다.

II. 유한체 상의 수학적 성질과 그래프의 성질

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

2.1. 유한체 상의 수학적 성질

유한체란 임의의 소수 P 와 양의 정수 m 에 대하여 P^m 개의 원소를 가지는 유일한 체를 말하며 Galois(Galois, Evariste 1811-1832 : 프랑스의 천재적 수학자)가 발견하여 일명 Galois체(Galois Field)라 불리우고 $GF(P^m)$ 으로 표시한다. 이때 P^m 을 유한체의 위수(order)라고 한다. 한편, 유한체는 기초체(ground field) $GF(P)$ 와 이를 m 차 확대한 확대체(extension field) $GF(P^m)$ 으로 나눌 수 있다. 기초체 $GF(P)$ 의 P 는 1보다 큰 소수로서 $GF(P)$ 상의 원소는 $\{0, 1, 2, \dots, P-1\}$ 이며, 확대체 $GF(P^m)$ 은 $GF(P)$ 상의 m 차 벡터공간(Vector Space)으로 표시할 수 있다. 일반적으로 유한체는 $\{S, +, \cdot, 0, 1\}$ 의 5가지 요소로 표시되며 S 는 원소들의 집합이고 $+$ 와 \cdot 는 S 상의 이항연산(binary operation)이며 0과 1은 각각 가산과 승산에 대한 항등원이며 모든 산술 연산은 mod P 로 처리된다.

한편 유한체 $GF(P^m)$ 상의 원소를 생성 하는데는 기초체 $GF(P)$ 의 존재를 만족하는 특성방정식을 인수분해하여 m 차 기약다항식을 구하고 이를 0으로 하는 한 원시근을 α 로 할 때 원시기약다항식(Primitive irreducible polynomial)을 얻을 수 있으며 이로부터 유한체의 원소들을 생성시킬 수가 있다. 한편, 유한체의 $GF(P^m)$ 내의 원소들은 수학적 성질로부터 계수 a_i 는 $a_i \in GF(P)(i=0, 1, 2, \dots, m-1)$ 이며 이는 P 치를 나타내는 Digit Code로써 표현 할 수 있다. 따라

서 모든 계수들을 Digit Code로써 할당함으로써 $GF(P^m)$ 내의 모든 원소들을 표시할 수 있다. 일반적으로 $GF(P^m)$ 상의 원소들을 P 치 Digit Code로 할당하는 종류는 $P^m!$ 개이다. 이때 $GF(P^m)$ 상의 원소들을 P 치 Digit Code로 할당하는데 필요한 Digit Code 개수는 m 개이고 최고차수계수인 a_{m-1} 을 MSD(Most Significant Digit), 최저차수계수인 a_0 을 LSD(Least Significant Digit)라 한다. 특히, $P=2$ 인 경우의 $GF(2^m)$ 상에서의 원소들은 Bit Code(Binary Digit Code)로 할당할 수 있고 최고차수계수인 a_{m-1} 을 MSB(Most Significant Bit), 최저차수계수인 a_0 을 LSB(Least Significant Bit)라 한다.

본 논문에서 사용한 $GF(P^m)$ 내의 원소들을 Digit Code로 할당하는 알고리즘은 다음과 같다.

[유한체 $GF(P^m)$ 내의 모든 원소들을 Digit Code로 할당하는 알고리즘]

- 단계 1 : 원소 e_0 는 Digit Code 0으로 한다.
- 단계 2 : 원소 e_{K-1} 은 Digit Code를 모두 $(P-1)$ 로 한다.(여기서, $K=P^m$)
- 단계 3 : LSD를 1에서부터 $(P-1)$ 까지 증가시키면서 할당한다.
- 단계 4 : 단계3의 방법을 계속해서 LSD로부터 한 디지털씩 왼쪽으로 시프트(shift) 시키면서 할당한다.
- 단계 5 : 단계4를 MSD까지 계속한다.

이 외에 유용한 수학적 성질은 다음의 참고문헌을 참조하였다.¹⁸⁻⁹⁾

2.2. 그래프의 성질

일반적으로 그래프는 다음 식(2-1)과 같이 표현한다.

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

여기서, V 는 유한개의 비공집합(nonempty set) 노드(node)의 집합이고, E 는 노드 집합에서의 2개의 부분집합(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 V 에 인접한 모든 노드의 집합을 노드 V 의 이웃(neighbour)이라

하고 $N(V)$ 로 표시하며 노드 V 에 의해 일어나는 모서리의 개수를 노드 V 의 깊이(degree)라 하며 $\deg(V)$ 로 표현한다. 특히 그래프G에서 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.^[10-12]

III. 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

[1] $LR[\text{digit stream}]$: 한개 디지털 왼쪽 로테이트

$$LR[D_{m-1}D_{m-2} \dots D_1D_0] = [D_{m-2}D_{m-3} \dots D_1D_0D_{m-1}]$$

[2] $RR[\text{digit stream}]$: 한개 디지털 오른쪽 로테이트

$$RR[D_{m-1}D_{m-2} \dots D_1D_0] = [D_0D_{m-1}D_{m-2}D_{m-3} \dots D_1]$$

[3] $CLR[\text{digit stream}]$: MSD에 대한 modP 보수의 왼쪽 로테이트

$$CLR[D_{m-1}D_{m-2} \dots D_1D_0] = [D_{m-2}D_{m-3} \dots D_1D_0D_{m-1}']$$

[4] $CRR[\text{digit stream}]$: LSD에 대한 modP 보수의 오른쪽 로테이트

$$CRR[D_{m-1}D_{m-2} \dots D_1D_0] = [D_0'D_{m-1}D_{m-2}D_{m-3} \dots D_1]$$

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

3.2. modP 변환연산자

[1] $LR(K) \bmod P[\text{digit stream}]$: modP(MSD+K)의 한개 디지털 왼쪽 로테이트

$$LR(K) \bmod P[D_{m-1}D_{m-2} \dots D_1D_0] = [D_{m-2}D_{m-3} \dots D_1D_0 \bmod P(D_{m-1}+K)]$$

[2] $RR(K) \bmod P[\text{digit stream}]$: modP(MSD+K)의 한개 디지털 오른쪽 로테이트

$$RR(K) \bmod P[D_{m-1}D_{m-2} \dots D_1D_0] = [\bmod P(D_0+K)D_{m-1}D_{m-2} \dots D_2D_1]$$

[3] $CLR(K) \bmod P[\text{digit stream}]$: (MSD+K)에 대한 modP 보수의 왼쪽 로테이트

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

[M] $CRR(K) \bmod P[\text{digit stream}]$: (LSD+K)에 대한 modP 보수의 오른쪽 로테이트

$$CRR(K) \bmod P[D_{m-1}D_{m-2} \dots D_1D_0] = [(D_0+K)' D_{m-1}D_{m-2} \dots D_2D_1]$$

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

3.3. 역 변환연산자

3.1.절의 일반연산자에서 $LR(\text{digit stream})$ 변환연산자와 $RR(\text{digit stream})$ 변환연산자는 서로 역 변환 관계에 있으며, $CLR(\text{digit stream})$ 변환연산자와 $CRR(\text{digit stream})$ 변환연산자는 서로 역 변환 관계에 있다. 또한, 3.2.절의 modP 변환연산자에서 $LR(K1) \bmod P[\text{digit stream}]$ 변환연산자와 $RR(K2) \bmod P[\text{digit stream}]$ 변환연산자는 서로 역 변환 관계에 있으며, $CLR(K1) \bmod P[\text{digit stream}]$ 변환연산자와 $CRR(K2) \bmod P[\text{digit stream}]$ 변환연산자는 서로 역 변환 관계에 있으며 이때 $K1+K2=P$ 의 조건을 갖는다.

즉, 위의 내용을 정돈하면 다음과 같다.

- $LR[\text{digit stream}]^{-1} = RR[\text{digit stream}]$
- $RR[\text{digit stream}]^{-1} = LR[\text{digit stream}]$
- $CLR[\text{digit stream}]^{-1} = CRR[\text{digit stream}]$
- $CRR[\text{digit stream}]^{-1} = CLR[\text{digit stream}]$
- $LR(K1) \bmod P[\text{digit stream}]^{-1} = RR(K2) \bmod P[\text{digit stream}]$
- $RR(K2) \bmod P[\text{digit stream}]^{-1} = LR(K1) \bmod P[\text{digit stream}]$
- $CLR(K1) \bmod P[\text{digit stream}]^{-1} = CRR(K2) \bmod P[\text{digit stream}]$
- $CRR(K2) \bmod P[\text{digit stream}]^{-1} = CLR(K1) \bmod P[\text{digit stream}]$

IV. De Bruijn 그래프의 인접행렬과 변환표

본 장에서는 앞의 3장의 내용을 토대로 De Bruijn 그래

프^[10-12]를 구성하기 위해 먼저 유한체 GF(P^m)상의 인접행렬을 도출하고 이를 바탕으로 De Bruijn 변환표를 구성한다. 인접행렬에서의 loop는 자기 자신의 원소를 의미하며, 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 m이 2인 GF(3²)상의 De Bruijn 그래프의 인접행렬을 구성하면 다음 표 4-1과 같다.

4.2. De Bruijn 변환표

본 절에서는 앞의 4.1.절의 내용을 토대로 유한체 GF(P^m) 상의 De Bruijn 변환표를 구하면 다음과 같다. 예를 들어 GF(3^m)에서 m이 2인 GF(3²) 상의 De Bruijn 변환표를 구성하면 다음 표 4-2와 같다. 여기서, 유한체 GF(3^m) 상에서의 LR(0)mod3 변환연산자와 RR(0)mod3 변환연산자의 결과는 동일하다.

표 4-2. GF(3²)상의 De Bruijn 변환표
Table 4-2. The De Bruijn transformation table over GF(3²).

	(00)	(01)	(02)	(10)	(11)	(12)	(20)	(21)	(22)
LR(0) mod3	(00)	(10)	(20)	(01)	(11)	(21)	(02)	(12)	(22)
RR(0) mod3	(00)	(10)	(20)	(01)	(11)	(21)	(02)	(12)	(22)
LR(1) mod3	(01)	(11)	(21)	(02)	(12)	(22)	(00)	(10)	(20)
RR(1) mod3	(10)	(20)	(00)	(11)	(21)	(01)	(12)	(22)	(02)
LR(2) mod3	(02)	(12)	(22)	(00)	(10)	(20)	(01)	(11)	(21)
RR(2) mod3	(20)	(00)	(10)	(21)	(01)	(11)	(22)	(02)	(12)

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

V. De Bruijn 그래프 구성

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

GF(5)상의 De Bruijn 그래프는 그림 5-1과 같으며, GF(2⁴), GF(3²), GF(3³)상의 De Bruijn 그래프 구성은 각각 다음 그림 5-2, 5-3, 5-4와 같다.

표 4-1. GF(3²)상의 De Bruijn 그래프의 인접행렬
Table 4-1. The De Bruijn adjacent matrix over GF(3²).

	(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

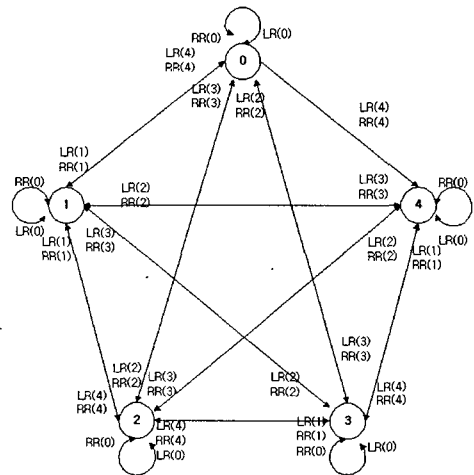


그림 5-1. GF(5)상의 De Bruijn 그래프 구성
Fig. 5-1. The De Bruijn graph construction over GF(5).

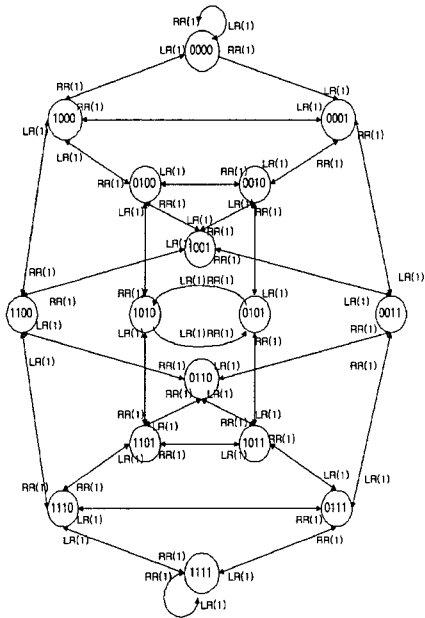


그림 5-2. $GF(2^4)$ 상의 De Bruijn 그래프 구성
Fig. 5-2. The De Bruijn graph construction over $GF(2^4)$.

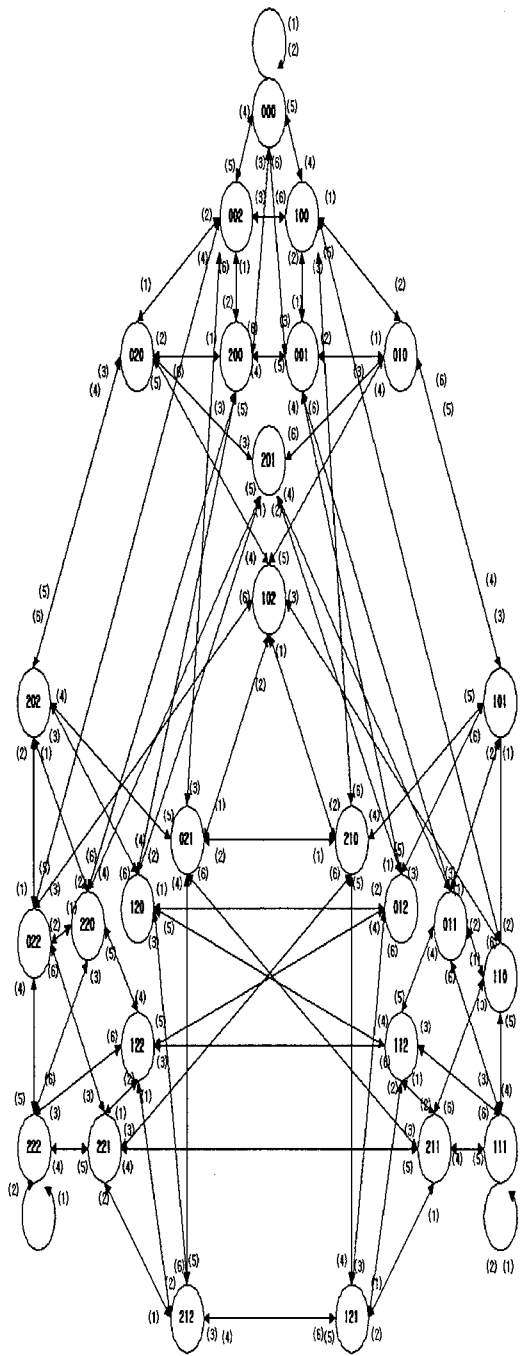


그림 5-4. $GF(3^3)$ 상의 De Bruijn 그래프 구성
Fig. 5-4. The De Bruijn graph construction over $GF(3^3)$.

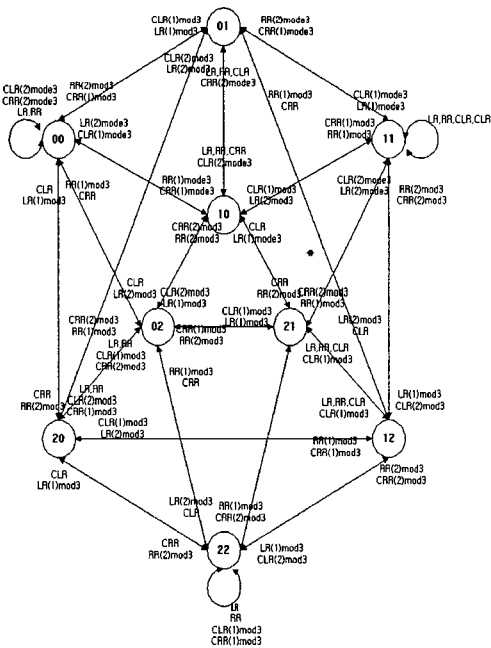


그림 5-3. $GF(3^2)$ 상의 De Bruijn 그래프 구성
Fig. 5-3. The De Bruijn graph construction over $GF(3^2)$.

VI. 결론

본 논문에서는 유한체 $GF(P^m)$ 상의 De Bruijn 그래프를 구성하여 이를 다중처리기 구성에 적용하는 한가지 방법을 제안하였다.

지금까지는 $P=2$ 인 $GF(2^m)$ 상의 2진(2치)논리 De Bruijn 그래프(BDBG)를 구성하여 이를 처리기 구성에 적용하였다.

본 논문에서 제안한 유한체 $GF(P^m)$ 상의 다중처리기는 P 가 어떤 값이건 De Bruijn 그래프를 구성할 수 있는 특징을 가진다.

제안한 $GF(P^m)$ 상의 De Bruijn 그래프를 바탕으로 한 다중처리기는 차세대 디지털논리시스템 중의 하나인 다치논리시스템 구성시에 원소들의 자유로운 변환과 스위칭 회로에 적용할 수 있으리라 사료된다.

향 후 연구과제로는 이를 바탕으로 한 유한체 $GF(P^m)$ 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. 그리고 향후 각종 진보된 스위칭함수 구성 등에 적용 및 응용할 수 있으리라 전망된다.

참고문헌

- [1] S. Mirta, N. R. Saxena, and E. J. McCluskey, "Efficient Design Diversity Estimation for Combinational Circuits," IEEE Trans. Comput. Vol.53,NO.11, pp.1493-1496, Nov. 2004.
- [2] V. Zyuban, D. Beutel, V. Srinivasan, M. Gschwind, P. Bose, P. N. Strenski, and P.G.Emma, "Intergrated Analysis of Power and Oerformance for Pipelined Microprocessors," IEEE Trans. Comput. Vol.53, NO.8, pp.1004-11016, Aug. 2004.
- [3] B. R. Childers and J. W. Davidson, "Custom Wide Counterflow Pipelines for high-performance Embedded Applications," IEEE Trans. Comput. Vol.53, NO.2, pp.141-158, Feb. 2004.
- [4] F. Balarin et al., *Hardware-Software Co-design of*

Embedded Systems: The Polis Approach, Kluwer Academic Press, Boston, June 1997.

- [5] R. K. Gupta, *Co-Synthesis of Hardware and Software for Digital Embedded Systems*, vol. 329, Kluwer Academic Publishers, Boston, Aug. 1995.
- [6] B.Arazi, "On the Sythesis of De-Bruijn Sequences", INFORMATION and CONTROL 49., pp.81-90, 1981.
- [7] A. Lempel, "On a Homomorphism of the DE- Bruijn Graph and its Applications to the Design of Feedback Shift Registers", IEEE Trans. Comput. Vol.C-19, NO12, pp.1204-1209, Dec. 1970.
- [8] I. F. Blake, *Algebraic Coding Theory : History and development*, Down, Hutchinson & Ross, Inc., Stroudburg, Pennsylvania, 1973.
- [9] R. Lidi and G. Piltz, *Applied Abstract Algebra*, Spring-Verlg, Inc., 1984.
- [10] E. Artin, *Galois Theory*, NAPCO Graphic arts, Inc., Wisconsin. 1971.
- [11] R. Gould, *Graph Theory*, The Bejamin/Cummings Publishing Company, Inc., 1988.
- [12] M. Gondran and M. Minoux, *Graph and Algorithms*, A Wiley-interscience publication, Toronto, 1984.

저자소개

박춘명(Chun-Myoung Park)



1994년 2월 인하대학교 대학원
전자공학과(공학박사)
1995년 9월 ~ 2005년 현재
충주대학교 전기·전자 및 정보
공학부 컴퓨터공학과 교수

1984년 ~ 2005년 현재 IEEE Computer Society Member
1984년 ~ 2005년 현재 대한전자공학회 정회원
2005년 현재 한국해양정보통신학회 학회지편집위원
2002년 ~ 2003년 UCI(University of California, Irvine) ICS
와 CECS(Center for Embedded Computer
Systems) Visiting Scholar

※ 관심분야 : 차세대 디지털논리시스템 및 컴퓨터 구조, 임베디드시스템, 마이크로프로세서 응용, 유비쿼터스컴퓨팅시스템, 멀티미디어시스템 등