

論文93-30B-5-1

# 다중 프로세서 시스템을 위한 블로킹없는 하이퍼큐브 구조와 루팅 알고리즘

## (Nonblocking Hypercube Architectures And Routing Algorithms For Multiprocessor Systems)

崔 相 昉\*

(Sang Bang Choi)

### 要 約

본 논문에서는 임의의 순열을 회선교환방식에서 구현하기 위한 진보된 하이퍼큐브 구조와 루팅 알고리즘을 제안하였다. 만약 하나의 링크쌍(한개의 전이중 통신선로)이 임의의 한 차원에 추가되면 하이퍼큐브가 재배열 가능하다는 것을 증명하였다. 두 이웃 노드간의 연결이 두개의 링크쌍으로 구성된다면, 하이퍼큐브는 두개의 독립된 순열을 동시에 다룰 수 있다는 것을 증명하였다.

### Abstract

In this paper, we develop enhanced hypercube architectures and routing algorithms to realize arbitrary permutations in circuit switching. We prove that a hypercube is rearrangeable if one additional pair of links (one full-duplex communication line) is provided in only one dimension of connections. We also prove that if each connection between two neighboring nodes consists of two pairs of links, the hypercube can handle two independent permutations simultaneously.

### 1. 서론

병렬처리 시스템에서 여러 PE(processing element)간의 통신경로를 제공하기 위해 다양한 상호연결망(interconnection network)들이 제안되었다. 여러 상호 연결망 중에서 하이퍼큐브(hypercube 또는 Boolean n-cube)는 높은 시스템 활용도(avail-

ability)와 알고리즘의 병렬수행 능력 때문에 최근 몇 년간 주목을 받아 왔다. 하이퍼큐브는 규칙성(regularity), 대칭성(symmetry), 강한 연결성(strong connectivity)으로 인하여 다양한 알고리즘의 병렬처리가 용이하다. 또한 오늘날의 다중 프로세서 시스템에 필수적인 고장 허용(fault-tolerant)능력도 가지고 있다. 하이퍼큐브 구조를 갖는 다중 프로세서 시스템을 위한 여러가지 알고리즘이 개발되었다.

하이퍼큐브에서 PE간 통신을 실현하기 위한 여러 가지 루팅 알고리즘(routing algorithm)이 제안되었다. 1대1 전송, 동시전송, 고장이 있는 경우, 고장이 없는 경우 등 여러 상황에 대한 루팅 알고리즘이

\*正會員, 仁荷大學校 電子工學科

(Dept. of Elec. Eng., Inha Univ. )

(※ 이 연구는 인하대학교 산업과학기술연구소 '91 학년도 자체학술연구비 지원에 의해 수행되었음.)

接受日字: 1992年 10月 29日

개발되었으나, 회선교환(circuit switching)방식 하이퍼큐브의 효과적인 라우팅 알고리즘과 순열 구현능력(permutation capability)에 대한 지식의 부족으로 대부분의 연구들이 패킷교환(packet switching)방식 하이퍼큐브에 대하여 행해졌다. <sup>(1-4)</sup> 본 논문의 관심은 회선교환방식하에서 임의의 순열을 실현할 수 있는 재배열 가능한(rearrangeable) 하이퍼큐브를 설계하는 것이다. 만약 하나의 링크쌍(link pair; 한 개의 전방향 통신선로)이 임의의 한 차원(dimension)에 추가되면 하이퍼큐브가 재배열 가능하다는 것을 증명하였다. 두 이웃 노드(node)간의 연결이 두 개의 링크쌍으로 구성된다면, 하이퍼큐브는 두개의 독립된 순열을 동시에 다룰 수 있다는 것을 증명하였다. 이러한 환경하에서는 메세지가 출발점(source)에서 도착점(destination)까지 경로를 따라 전달되는 동안 중간노드의 버퍼링(buffering)은 허용되지 않는다.

재배열 가능한 연결망의 장점은 실현할 수 있는 연결패턴이 다양하여 여러가지 일반적인 문제를 쉽게 처리할 수 있다. 프로세서가 고장인 경우에는 고장난 프로세서를 발견해서 다른 정상적인 프로세서로 대체하는 것이 비교적 수월하다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 하이퍼큐브의 정의, 위상학적(topological) 특성, 하이퍼큐브의 재배열 가능성에 관한 내용이 논의된다. 3장에서는 하이퍼큐브를 등가의 다단(multistage) 그래프 모델로 변환시키고 이중 링크 하이퍼큐브라 불리는 세가지 재배열 가능한 하이퍼큐브 구조를 유도해 낸다. 마지막으로 4장에서 결론을 맺는다.

## II. 하이퍼큐브의 위상 및 재배열성

### 1. 하이퍼큐브의 정의 및 위상학적 특성

이 장에서는 그래프 이론의 관점에서 하이퍼큐브를 연구하고, 위상학적 특성을 알아본다. 우선 하이퍼큐브를 무방향 그래프  $Q_n = (V, A)$ 로 간주한다. 여기서 아래첨자  $n$ 은 하이퍼큐브의 크기를,  $V$ 는 노드들의 집합을,  $A$ 는 링크의 집합을 나타낸다. 즉  $A \subseteq V \times V$ .

[정의 1]  $G = (V, A)$ 를  $0, 1, \dots, |V| - 1$ 로 번호가 붙여진 노드를 갖는 무방향 그래프라고 하자.  $G^d = (V^d, A^d)$ 를 그래프  $G$ 의 복사본이라 하고,  $G^d$ 의 노드는  $|V|, |V| + 1, \dots, 2|V| - 1$ 로 번호를 붙인다.  $G^i = G + G^d$ 로 표현되는 그래프  $G = (V, A)$ 의 이중화 동작(twofold operation)에 의해  $G_i = (V_i, A_i)$ 이 만들어지며, 여기서  $V_i = V \cup V^d$ 이고  $A_i =$

$A \cup A^d \cup \{ (v, |V| + v) \mid \forall v \in V \}$ 이다.

따라서,  $0$  과  $|V| - 1$  사이의 수,  $k$  에 의해 번호 붙여진 원래 그래프  $G$ 의 각 노드들에 대응하는 복사된 그래프  $G^d$ 의 노드  $v^d$ 는  $|V| + k$ 로 번호를 붙이고, 두 노드간의 새로운 링크  $(v, v^d)$ 는 집합  $A_i$ 에 더한다.

[정의 2] Boolean  $n$ -cube  $Q_n = (V, A)$ 는 다음과 같이 재귀적(recursive)으로 정의된다.

1.  $0$ -cube,  $Q_0$ , 는 링크가 없는 단일 노드로 정의된다.

$$2. Q_n = Q_{n-1} \cup Q_{n-1}^d$$

$n$ -cube의 가장 중요한 성질은 작은 차원의 큐브로부터 재귀적으로 구성될 수 있다는 점이다. 역으로,  $n$ -cube는 재귀적으로 작은 차원의 서브큐브(subcube)로 분리될 수 있다. 하이퍼큐브의 특성은 참고문헌 [5] 에 설명되어 있다.

통신망으로서의 하이퍼큐브에서는 인접하는 모든 노드 간의 무방향 링크  $(v_i, v_j) \in A$ 는 방향이 서로 반대인 두개의 통신링크, 즉 전방향 통신선로(full-Duplex communication link)로 볼 수 있다. 이러한 하이퍼큐브 모델은 방향성 그래프로 간주될 수 있고,  $Q_n = (V, E)$ 로 나타낸다. 여기서  $E$ 는 방향성 링크의 집합  $E = \{ (v_i \rightarrow v_j), (v_j \rightarrow v_i) \mid \forall (v_i, v_j) \in A \}$ 이다.

[정의 3] Boolean  $n$ -cube  $Q = (V, E)$ 에서 한 링크의 양끝 노드의 이진 번지가  $k$  번째 비트 (bit)만 틀린 경우 이런 방향성 링크를  $k$ -D(dimension) 링크라 하며, 그 두 노드사이를 연결하는 방향이 반대인 두개의 링크를  $k$ -D 링크쌍이라 한다. 본 논문에서 제안된 이중 링크 하이퍼큐브 구조에서는 두개의  $k$ -D 링크쌍이 있을 수 있다. 그 두 노드들 사이의 모든  $k$ -D 링크쌍들을  $k$ -D 연결이라 부른다.

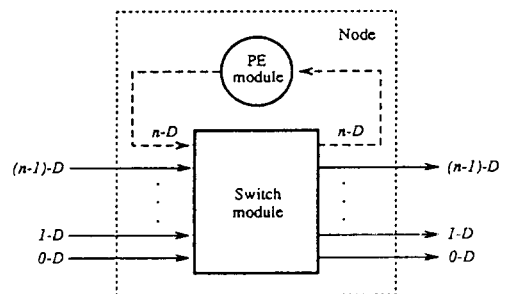


그림 1. PE 모듈과 스위치 모듈로 구성된 노드  
Fig. 1. A node constructed with PE and switch modules.

k-D 연결의 집합은  $E_k = \{(v_i \rightarrow v_j) \in E \mid v_i \oplus v_j = 2^k\}$ 로 나타낸다. 따라서 E 는  $E_k$ 의 합집합이며 k는 0부터 n - 1까지의 값을 가진다. 하이퍼큐브에서 각 노드들은 n개의 연결을 가지며, 각각은 0-D, 1-D, ..., (n - 1)-D 연결이다. 모든 노드들은 그림 1에서 처럼 두개의 모듈(PE 모듈과 스위치 모듈)로 구성되어 있다. 그림에서 PE 모듈과 스위치 모듈간의 연결은 편의상 n-D 연결이라 부른다.

## 2. 하이퍼큐브의 재배열성

N 개의 PE로 구성된 다중 프로세서 시스템에서 PE들이 동시에 서로 통신하려 한다고 생각해 보자. PE를 0에서 N - 1까지의 정수로 나타내면 그러한 통신형태는  $[0, 1, \dots, N - 1]$ 의 순열로 표현된다.  $[0, 1, \dots, N - 1]$ 의 순열은 자기자신으로의 1대1 대응이며 다음과 같이 표현된다.

$$P = (\pi : PE_i \rightarrow PE_j \mid i, j = 0, 1, \dots, N - 1)$$

N = 4 인 경우 순열  $[3, 2, 0, 1]$ 은  $\pi(PE_0) = PE_3$ ,  $\pi(PE_1) = PE_2$ ,  $\pi(PE_2) = PE_0$ ,  $\pi(PE_3) = PE_1$ 의 통신 형태를 나타낸다. N개의 PE를 가진 연결망에서는 N! 개의 순열이 가능하다.

PE간의 통신은 요구되는 순열을 구현해 줄 수 있는 상호 연결망을 통하여 이루어진다. 만일 기존의 연결에 영향을 주지않고 새로운 연결을 만들 수 있는 루팅 알고리즘이 존재하면 상호 연결망은 엄밀하게 블로킹이 없다(strictly nonblocking)고 한다. 따라서 엄밀히 블로킹이 없는 연결망에서는 임의로 발생하는 출발점-도착점 연결 요구는 양쪽 PE가 사용되지 않는 한 그 알고리즘에 의해 구현될 수 있다.<sup>[6]</sup> 어떤 네트워크에서 필요한 경우 기존의 연결을 재배열하여 모든 순열을 구현할 수 있으면 그 네트워크는 재배열 가능하다고 한다.<sup>[6]</sup> 하이퍼큐브가 재배열 가능하다면 모든 순열에 대하여 링크가 중첩되지 않는(link-disjoint) 경로가 존재하여야한다.

상호 연결망의 성능은 통신 지연과 순열 능력에 의해서 측정될 수 있다. 회선교환방식에서는 중간 프로세서에서의 큐잉(queueing) 지연이 없고 전기적 전파 지연은 무시할 수 있으므로, 상호연결망의 성능은 대부분 순열 능력에 의해 결정된다. 병렬처리 시스템의 성능은 R/C 비율에 의해 결정된다.<sup>[7]</sup> 여기서 R은 작업수행 시간이고, C는 그 작업에 의해 생기는 통신비용(communication cost)이다. 그 비율이 매우 낮으면 병렬처리는 효율적이며, 반대로 높으면 병렬처리는 효과적으로 활용된다. 회선교환방식에서는 요구되는 연결 형태를 상호 연결망이 지원하는한 각 작업의 크기와는 관계없이 그 비는 매우 높다. 따

라서 회선교환에서 재배열 가능한 연결망은 매우 효율적인 병렬처리를 제공한다.

Sullivan과 Bashkow<sup>[8]</sup>는 간단한 1대1과 동시 전송 루팅 알고리즘을 그들의 하이퍼큐브 시스템(C HPP - Columbia Homogeneous Parallel Processor)에 사용하였다. Valient<sup>[4]</sup>는 두개의 패킷이 동시에 같은 링크를 통과함이 없이  $O(\log_2 N)$  시간 안에 목적지에 이를 수 있는 분산된 무작위(distributed randomized) 알고리즘을 제안했다. Johnson과 Ho<sup>[1]</sup>는 하이퍼큐브로 구성된 다중 프로세서에서 네가지 다른 통신 문제를 다루기 위해 신장 트리(spanning tree)를 사용한 통신 그래프를 소개했다. 위의 연구들은 패킷교환방식에 초점을 두고 있다. 그러나 회선교환방식의 하이퍼큐브의 재배열 가능성에 관해서는 연구가 거의 없었다. Szymanski<sup>[6]</sup>는 임의의 순열을 구현하기 위한 회선교환방식 하이퍼큐브의 능력을 연구하고 하이퍼큐브가  $n \leq 3$ 에 대해 재배열 가능하다는 것을 증명했다. 그러나  $n \geq 4$ 인 경우에 대해서는 재배열 가능한지 아닌지 증명도 반증도 되어 있지 않다. Qruc<sup>[9]</sup>가 제안한 재귀적 카르노 도(recursive Karnaugh maps)를 이용한 경로 설정 알고리즘은 큐브로 연결된 네트워크에서 일부 순열(transposition, 사이클 등)의 경로를 설정하는 데 사용될 수 있다. 본 논문의 목적은 임의의 순열을 모두 실현할 수 있는 하이퍼큐브 구조와 루팅 알고리즘을 고안하는 것이다.

## III. 재배열 하이퍼큐브 구조

### 1. 하이퍼큐브의 다단 네트워크로의 변환

이중 링크 하이퍼큐브라고 부르는 재배열 가능한 하이퍼큐브 구조를 이끌어 내는 주된 과정은 하이퍼큐브를 등가의 다단 상호 연결망으로 변환시키고 그것을 분석하는 것이다. 이렇게 변환시키는 이유는 다단 상호연결망의 재배열 가능성은 많이 연구된 문제이고, 또 하이퍼큐브 같은 정적 네트워크 보다 다루기 쉽기 때문이다.

[정의 4] Boolean n-cube,  $Q_n = (V, E)$ , 에서 자체 연결(self-connection)의 집합은 각 노드로부터 자신의 노드로 연결된 방향성 링크들의 집합이고,  $E_s = \{(v_i \rightarrow v_i) \mid v_i \in V\}$ 로 표기 한다. 자체 연결을 포함한 k-D 연결은  $E_k$ 와 구별하기 위해서  $E_k = E_k \cup E_s$ 로 표기한다.

자체 연결의 집합  $E_s$ 는  $E_n$ 으로 나타내어지는 노드 내부의 n-D 연결 집합과 구분되어야 한다.  $E_s$ 는 가상의 링크인데 반해  $E_n$ 는 노드내에 실제로 존재하는

링크이다. 재배열 가능한 하이퍼큐브와 루팅 알고리즘을 고안하기 위해 하이퍼큐브  $Q_n = (V, E)$ 를  $2n$ 개의 이분 그래프(bipartite graph)로 다음과 같이 변환시킨다.

1. 각 차원  $k$  ( $k = n-1, n-2, \dots, 0$ )에 대하여,  $V_{1,k}$ 와  $V_{2,k}$ 로 표기되는 노드 집합  $V$ 의 복사본을 두개 만든다. 원래의 노드  $v_i \in V, 0 \leq i \leq N - 1$ , 에 대하여 복사된 두 노드,  $v_i^{1,k} \in V_{1,k}$ 와  $v_i^{2,k} \in V_{2,k}$ 가 존재한다. 그리고  $k$ -D 링크의 연결관계를  $V_{1,k}$ 와  $V_{2,k}$ 에 사상(mapping)하여 방향성 이분 그래프  $B_k = (V_{1,k}, V_{2,k}, E_k)$ 를 만든다. 여기서  $V_{1,k}$ 는  $B_k$ 의 입력 노드집합 이고,  $V_{2,k}$ 는 출력 노드 집합이다. 그리고,  $E_k$ 는 모든 노드의 자체 연결을 포함하는  $k$ -D 연결의 방향성 링크 집합이다. 따라서,  $Q_n$ 에서 노드  $v_i \in V$  로부터 노드  $v_j \in V$  로  $k$ -D 링크가 존재한다면,  $B_k$ 에는 이에 대응하는  $v_i^{1,k} \in V_{1,k}$ 에서  $v_j^{2,k} \in V_{2,k}$ 로의 방향성 링크가 존재한다. 또한 모든  $i$ 값에 대하여  $v_i^{1,k}$ 에서  $v_i^{2,k}$ 로의 방향성 링크가 존재한다(자체 연결). 따라서  $k$ -D 연결에 대하여 하이퍼큐브  $Q_n$ 에서 이분 그래프  $B_k$ 로의 사상  $m_k$ 는 다음과 같이 나타낼 수 있다.

$$\mu_k : Q_n \rightarrow B_k$$

2. 이분 그래프  $B_k$ 의 출력 노드 집합  $V_{2,k}$ 가  $B_{k-1}$ 의 입력 노드집합  $V_{1,k-1}$ 과 일치하도록 직렬로 연결한다. 여기서  $n - 1 \geq k \geq 1$ 이다. 따라서 노드  $v_i^{2,k} \in V_{2,k}$ 와 노드  $v_i^{1,k-1} \in V_{1,k-1}$ 은 하나의 동일한 노드가 된다. 이와 같이 얻어진 그래프를  $G_n$ 으로 나타낸다.
3.  $V_{2,0}$ 에 대하여  $G_n$ 과 대칭되는 복사본을 만들고 이것을  $G'_n$ 으로 나타낸다.  $G_n$ 의 입력(출력) 노드는  $G'_n$ 의 출력(입력) 노드가 된다. 또한  $G_n$ 의 출력 노드집합  $V_{2,0}$ 는  $G'_n$ 의 입력 노드집합  $V'_{2,0}$ 와 동일한 집합이다.  $G'_n$ 의 각 이분 그래프는  $B'_k = (V'_{2,k}, V'_{1,k}, E'_k), k = 0, 1, \dots, n - 1$ 로 나타내며  $V'_{1,k}$ 는  $G_n$ 에서  $V_{1,k}$ 의 복사본 이고,  $V'_{2,k}$ 는  $G_n$ 에서  $V_{2,k}$ 의 복사본이다. 그 결과로써 얻은 그래프가 그림 2에 나타나 있다.

그림 2의 합성된 연결망  $G_n \cdot G'_n$ 은 다음과 같이 표기한다. 여기서 “ $\cdot$ ”와 “ $\prod$ ”는 이분 그래프의 직렬 연결을 나타낸다.

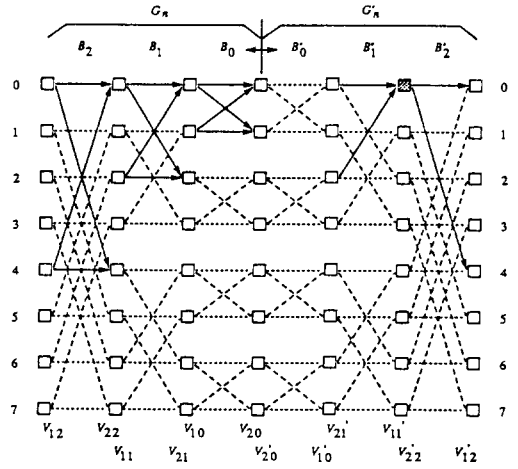


그림 2. 하이퍼큐브의 변환된 그래프 모델  
Fig. 2. Transformed graph model of hypercube.

$$\begin{aligned}
 G_n \cdot G'_n &= (V_{1,n-1}, V_{2,n-1}, E_{n-1}) \cdot (V_{1,n-2}, V_{2,n-2}, E_{n-2}) \\
 &\quad \cdots (V_{1,0}, V_{2,0}, E_0) \cdot (V'_{2,0}, V'_{1,0}, E'_0) \\
 &\quad \cdot (V'_{2,1}, V'_{1,1}, E'_1) \cdots (V'_{2,n-1}, V'_{1,n-1}, E'_{n-1}) \\
 &= 3(V_{1,i}, V_{2,i}, E_i) \cdot \prod (V'_{2,j}, V'_{1,j}, E'_j) \\
 &= \prod B_k \cdot \prod B'_j \quad (1)
 \end{aligned}$$

[정의 5] 그림 2에서 긴 점선은 각 차원에서 실제로 연결된 방향성 링크를 나타내고, 짧은 점선은 자체 연결(한 노드에서 자기 자신으로의 방향성 링크)을 의미한다. 따라서 전자를 실제(real) 링크라 부르고 후자를 가상(pseudo) 링크라 부른다.

[정의 6] 변환된 그래프의  $B_k$  또는  $B'_k$ 에서 실제 링크쌍 ( $v_i \rightarrow v_i \oplus 2^k$ )와 ( $v_i \oplus 2^k \rightarrow v_i$ )의 complement는 가상 링크쌍 ( $v_i \rightarrow v_i$ )와 ( $v_i \oplus 2^k \rightarrow v_i \oplus 2^k$ )으로 정의 된다.

변환된 그래프는 다음과 같은 특성을 가진다. 특성 2에서 부터 5까지는  $B'_k$ 에 대해서도 성립된다.

특성 1 : 이 그래프는  $V_{2,0}$  또는  $V'_{2,0}$ 에 대해 대칭이다.

특성 2 : 모든 이분 그래프  $B_k, 0 \leq k \leq n - 1$ , 는  $2^{n-1}$ 개의 서로 분리된 부분 그래프(subgraph)로 구성되어 있다.

특성 3 :  $B_k$ 의 각 부분 그래프는 이웃하는 두 노드  $v_i$ 와  $v_i \oplus 2^k$ 를 연결하는  $k$ -D 링크쌍에 대응한다.

특성 4 : 특성2와 3으로부터  $k$ -D 연결을 구성하는  $2^{n-1}$ 개의 링크쌍들은 이에 대응하는 이분 그래프  $B_k$ 에 사상된다.

특성 5 : 모든 이분 그래프들은 서로 동일형태(isomorphism)이다. 따라서 변환된 그래프는 하이

퍼큐브의 대칭성 (노드 대칭과 링크 대칭)을 그대로 유지한다.

2. 이중 링크 하이퍼큐브 구조 1

그림 2는  $N = 2^n$  입력/출력과  $2 \times 2$  스위치로 이루어진  $2n$  단(stage)을 가진 다단 상호연결망의 그래프 모델로 볼 수 있다. 이 모델에 대하여 두가지의 해석이 가능하다. 첫번째 해석에서 그래프의 노드는 연결망의 통신 선로를 나타내고, 링크는 연결망에서의 스위칭 소자를 나타낸다. 따라서 이분 그래프내의 각 부분 그래프 상에서 교차된 두 링크와 평행한 두 링크는  $2 \times 2$  스위치(그림 2의  $B_2, B_1, B_0$ 에서 실선으로 표시)로 생각할 수 있다. 두 번째 해석은 그래프의 노드를  $2 \times 2$  스위치로 보고 링크는 스위치의 입출력선으로 보는 것이다 (그림 2의  $G'_n$ 에서 색칠된 노드 및 실선으로 표시된 링크).

변환된 그래프를 분석하기 위해 우선 첫번째 해석을 적용한다. 그러면 합성된 연결망  $G_n$ 은 Banyan 네트워크로 간주할 수 있다.  $G_{n,0} = \prod_{i=1}^n B_i$ 를  $B_0$ 가 없는  $G_n$ 이라 하면, 합성된 연결망  $G_{n,0} \cdot G'_n$ 은  $(2n - 1)$ 단 Benes 네트워크와 위상이 동일하다. Benes 네트워크는 재배열가능한 연결망임이 잘 알려진 사실이고, 모든 순열을 구현하기 위한 루팅 알고리즘은 Waksman<sup>[10]</sup>에 의해 제안되었다. 재배열 가능한  $2(\log N) - 1$ 단 네트워크의 루팅 알고리즘은 참고문헌 [11]에서도 다루었다. 이러한 종류의 루팅 알고리즘은 일종의 루핑(Looping) 알고리즘이다.

[정리 1] 임의의 한 차원의 연결에서는 하나의 링크쌍을 가지고 다른 모든 차원에서는 두개의 링크쌍을 가진 하이퍼큐브는 어떤 순열도 실현할 수 있다.

[증명] 그림 2에서 사용되는 각 스위칭단은 하이퍼큐브에서 해당되는 차원의 링크 사용을 나타낸다.  $n - 1, n - 2, \dots, 1$  차원에 해당하는 스위칭단을 두번 사용되었다( $G_{n,0}$ 에서 한번  $G'_n$ 에서 한번). 반면에 0-D에 해당하는 스위칭단은  $B'_0$ 에서 한번 사용되었다. 따라서 0-D 연결에 하나의 링크쌍을, 다른 모든 차원에는 두개의 쌍을 갖는 하이퍼큐브는 모든 순열에 대해 블로킹없이 루트를 제공한다. 변환된 그래프의 특성 5로부터 0-D대신 어떤 차원을 사용할 수 있다. Q.E.D.

루팅 알고리즘에 의해  $G_{n,0} \cdot G'_n$  상에 설정 경로가 하이퍼큐브상으로 사상되는 경우 필요없는 사이클(cycle: 출발점과 도착점이 같은 폐경로) 또는 가상 링크가 존재할 수 있다.  $G_{n,0} \cdot G'_n$ 에서 설정된 경로에서 연속적으로 반복되는 노드들은 하이퍼큐브에 사

상된 후 가상 링크의 부분이되고 서로 떨어져 반복된 두 노드사이의 경로는 하이퍼큐브에서 사이클 부분이 된다.

[정리 2]  $G_{n,0} \cdot G'_n$  상에 설정된 경로에서 서로 떨어져서 반복되는 노드들은 항상  $B'_0$ 에 대칭으로 생긴다.

[증명]  $G_n \cdot G'_n$ 에 대하여 증명할 것이다. 그러나,  $G_{n,0} \cdot G'_n$ 에 대해서도 동일하게 성립하는 것을 쉽게 알 수 있다.  $G_n$ 과  $G'_n$ 은 유일 경로(unique-path)연결망이므로,  $G_n$  또는  $G'_n$ 에서 임의의 두 입출력 노드 사이는 유일한 경로가 있다.  $G_n \cdot G'_n$ 에서 모든 경로는  $V_{2,0}$ (또는  $V'_{2,0}$ )를 통과한다. 반복되는 두 노드  $v_i$ 와  $v'_i$  사이의 부분 경로(subpath)가 노드  $v_k \in V_{2,0}$ (또는  $v'_k \in V'_{2,0}$ )를 통과한다고 하자. 가정한  $v_i$ 와  $v_k$ 사이의 부분 경로는  $G_n$ 의 유일 경로 특성에 의해 유일해야 한다.  $v'_k$ 와  $v'_i$ 사이의 부분 경로도 같은 이유로 유일하다.  $v_k = v'_k$ 이고,  $G'_n$ 과  $G_n$ 은 서로 대칭이므로  $v_i$ 와  $v_k$ 사이의 부분 경로 및  $v'_k$ 와  $v'_i$ 사이의 부분 경로는  $v_k$ (또는  $v'_k$ )에 대하여 대칭이다. 다시말하면  $v_i$ 와  $v'_i$ 사이의 모든 중간 노드들은  $V_{2,0}$ (또는  $V'_{2,0}$ )에 대해 대칭이다.

Q.E.D.

제일 바깥쪽의 반복된 두 노드를 찾아서 그 사이의 부분 경로를 제거한 후 하이퍼큐브로 사상하면 사이클이 없는 경로를 얻게 된다. 여기서 제거되는 사이클은 가상 링크를 일부 포함할 수도 있다. 마지막으로 남아있는 연속적으로 반복되는 노드들을 제거하여야 한다. 하이퍼큐브를 위한 완전한 루팅 알고리즘은 다음 세 단계로 이루어진다.

- (1) 주어진 순열에 대하여 재배열 루팅 알고리즘을  $G_{n,0} \cdot G'_n$ 에 적용한다.
- (2) 위에서 구한 각각의 경로에서 사이클을 제거한다.
- (3) 연속적으로 반복되는 노드들로 이루어진 가상 링크를 제거한다.

위 루팅 절차에 의하여 얻어진 경로는 많아야  $2 \log_2 N - 2$ 개의 노드와  $2 \log_2 N - 1$ 개의 링크를 통과한다. 제안된 이중 링크 하이퍼큐브에서 필요한 총 링크의 갯수는  $2|E| - N$ 이다. 이것은 원래의 하이퍼큐브의 약 두배이다. 위에서 기술한 루팅 알고리즘의 결과는 재배열 다단 상호연결망에 대한 것과 같다. 다음절에서는 재배열 다단 상호 연결망보다 경제적이고 우수한 이중 링크 하이퍼큐브 및 루팅 알고리즘을 소개한다.

3. 이중 링크 하이퍼큐브 구조 2

이절에선 노드와 링크에 대한 해석을 바꿔서 분석

한다. 즉 그래프  $G_n \cdot G'_n$ 에서 노드는  $2 \times 2$  스위치로 간주되고 링크는 스위치의 입출력선으로 간주된다. 스위치에 연결된 각각 두개의 입력선과 출력선중 하나는 실제 링크이지만 다른 하나는 가상 링크이다.

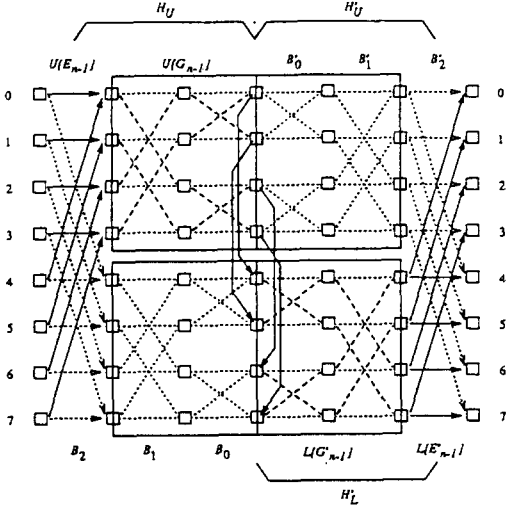


그림 3. 변환된 그래프에서 가능한 여러 라우팅 경로  
Fig. 3. Alternative routing paths in transformed graph.

$G_{n-1} = \prod_{i=0}^{n-2} B_i$ 이고  $G'_{n-1} = \prod_{i=0}^{n-2} B'_i$ 라고 표기한다.  $G_{n-1}$ 의 위쪽과 아래쪽의 부분 그래프는 각각  $U[G_{n-1}]$  과  $L[G_{n-1}]$  로 표시한다(그림 3 참조).  $U[E_{n-1}]$  은  $U[G_{n-1}]$  에 접하는  $E_{n-1}$ 의 부분집합을 나타내고,  $L[E_{n-1}]$  은  $L[G_{n-1}]$  에 접하는  $E_{n-1}$ 의 부분집합을 나타낸다. 그래프  $G'_{n-1}$ 에 대하여도 동일한 표기법을 사용할 수 있다. 그러면 그래프  $G_n \cdot G'_n$ 은 다음과 같이  $H_U, H'_U, H_L, H'_L$  네개의 부분 네트워크로 나뉘지며, 두개의 합성 연결망을 구성할 수 있다.

$$H_U \cdot H'_U = U[E_{n-1}] U[G_{n-1}] \cdot U[G'_{n-1}] U[E'_{n-1}] \quad (2)$$

$$H_L \cdot H'_L = L[E_{n-1}] L[G_{n-1}] \cdot L[G'_{n-1}] L[E'_{n-1}] \quad (3)$$

$H_U \cdot H'_U$ 와  $H_L \cdot H'_L$ 은  $N$  입력과  $N$  출력을 가진 두개의 독립적인 재배열 가능 연결망으로 간주할 수 있다. 그래프  $G_n \cdot G'_n$ 의 가장 왼쪽의 노드( $V_{1,n-1}$ )와 가장 오른쪽의 노드( $V'_{1,n-1}$ )는 입력과 출력 단자로 간주한다.

[정리 3] 모든 차원에 두개의 링크쌍을 가진 하이퍼큐브에서 임의의 한 차원에 대해 분리된 두개의 서브큐브  $Q^U_{n-1}$ 과  $Q^L_{n-1}$  중 하나를 사용하면 모든 순열을 실현할 수 있다.

[증명] 부분 그래프  $H_U \cdot H'_U$ 와  $H_L \cdot H'_L$ 은  $(n-1)$ -D 연결에 의해 연결된 두개의 서브큐브에 해당한다( $Q^U_{n-1} \cup Q^L_{n-1}$ ).  $H_U \cdot H'_U$ 와  $H_L \cdot H'_L$ 은 각각  $2^n$  노드의 모든 순열에 대해 재배열 가능하다. 변환된 그래프의 특성 5로부터 임의의 차원에 대해 분리된 어떤  $Q_{n-1}$  하나를 사용하여도 ( $2n$  종류의 서브큐브 가능) 모든 순열을 실현할 수 있다. Q.E.D.

[정리 4] 모든 차원에 두개의 링크쌍을 갖는 하이퍼큐브는 독립된 두 순열을 동시에 실현할 수 있다.

[증명] 정리 3으로부터 곧바로 증명이 된다. 즉, 독립된 두 순열을 위하여 임의의 한 차원에 대하여 분리된 두개의  $Q_{n-1}$ 을 각각 사용하면 그 두 순열을 동시에 실현할 수 있다. Q.E.D.

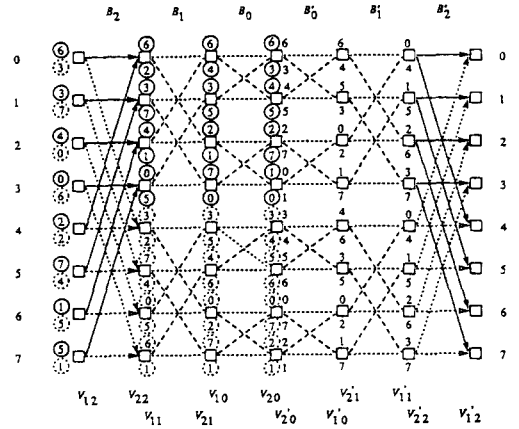


그림 4. 두 순열 [6, 3, 4, 0, 2, 7, 1, 5] 와 [3, 7, 0, 6, 2, 4, 5, 1] 이  $H_U \cdot H'_U$  와  $H_L \cdot H'_L$ 에 각각 적용된 결과

Fig. 4. Two permutations: [6, 3, 4, 0, 2, 7, 1, 5] for  $H_U \cdot H'_U$  and [3, 7, 0, 6, 2, 4, 5, 1] for  $H_L \cdot H'_L$ .

앞절에서 설명한 바와 같이  $H_U \cdot H'_U$  또는  $H_L \cdot H'_L$ 상에 설정된 경로에서 몇개의 반복되는 중간 노드들이 있을 수 있다. 그 노드들은 하이퍼큐브에서 사이클이나 가상링크를 만드므로 앞절과 동일한 절차에 의해 제거하여야 한다. 그림 4는 두 순열 [6, 3, 4, 0, 2, 7, 1, 5] 와 [3, 7, 0, 6, 2, 4, 5, 1] 를

$H_U \cdot H'_U$  와  $H_L \cdot H'_L$  에 각각 적용한 것이다.

각 노드  $v_i^{1,1} \in V_{1,1}$ 는  $V_{1,2}$ 로부터 두 토큰(token)  $p(v_i^{1,2})$ 와  $\pi(v_i^{2,2})$ 를 받는다. 그림으로 부터 얻어진 경로에서 사이클과 가상링크를 제거하면 이중링크 3-D 하이퍼큐브에서 두 순열을 실현할 수 있는 완전한 경로를 얻을 수 있다.

4. 이중링크 하이퍼큐브 구조 3

그래프  $H_U \cdot H'_U$ 에서,  $H'_U$ 상에 설정된 경로는  $H'_L$ 상으로 직접 사상될 수 있다(1대1 사상). 만일  $U [G_{n-1}]$ 의 출력이  $L [G'_{n-1}]$ 의 입력으로 직접 연결된다면,  $H_U$ 와  $H'_L$ 은 또다른 재배열 가능한 연결망으로 간주할 수 있으며 다음과 같이 나타낸다.

$$H_U \cdot 2L [E'_{n-1}] \cdot H'_L = U [E_{n-1}] U [G_{n-1}] \cdot 2L [E'_{n-1}] \cdot L [G'_{n-1}] L [E'_{n-1}] \quad (4)$$

$$H_L \cdot 2U [E'_{n-1}] \cdot H'_U = L [E_{n-1}] L [G_{n-1}] \cdot 2U [E'_{n-1}] \cdot U [G'_{n-1}] U [E'_{n-1}] \quad (5)$$

여기서  $2L [E'_{n-1}]$ 은  $H_U$ 에서  $H'_L$ 으로 향하는 두개의  $(n-1)$ -D 방향성 링크를 나타내고  $2U [E'_{n-1}]$ 은  $H_L$ 에서  $H'_U$ 으로 향하는 두개의  $(n-1)$ -D 방향성 링크를 나타낸다.  $H_U \cdot 2L [E'_{n-1}] \cdot H'_L$ 은 그림 3에 나타나 있다. 다음 기술에서는 표기를 간단히 하기위해  $2L [E'_{n-1}]$ 이나  $2U [E'_{n-1}]$ 를 빼기로 한다.

토큰  $\pi(i)$ 가  $H_U$ 의  $U [V_{2,0}]$ 에서 이미 도착점에 도달하였으면  $H'_U$ 에서  $H'_L$ 으로 사상할때 도착점에 두번 도달하게 되므로 사이클을 형성한다. 따라서 이러한 부분 경로는 사상에서 제외하여야 한다. 사상후 가상 링크들을 없애기 위해  $H_U \cdot H'_L$ 에서 얻어진 경로에서 연속적으로 반복된 노드들을 제거해야 한다. 따라서 루팅 알고리즘은 다음 세단계로 이루어진다.

- 1) 주어진 순열에 대하여 재배열 루팅 알고리즘을  $H_U \cdot H'_U$ 에 적용.
- 2)  $H'_U$ 에 설정된 경로를  $H'_L$ 으로 사상.
- 3) 가상링크 제거.

위 루팅 절차에 의하여 얻어진 경로는 많아야  $2 \log N$  개의 노드와  $2 \log N + 1$  개의 링크를 통과한다.

[ 정리 5 ] 하이퍼큐브에서 임의의 한 차원의 연결에 한 링크쌍을 추가하면 모든 순열을 실현할 수 있다.

[ 증명 ]  $U [G_{n-1}]$  과  $L [G'_{n-1}]$  사이에는 중복된 노드들이 없다. 따라서  $(n-2)$ -D 부터 0-D 연결에 속하는 모든 링크들은 단 한번 사용된다. 만일  $U$

$[G_{n-1}]$  과  $L [G'_{n-1}]$  에 해당하는 두 서브큐브  $QU_{n-1}$ 와  $QL_{n-1}$ 사이의  $(n-1)$ -D 연결에 한 링크쌍이 추가 된다면, 그 연결에서는 각 방향으로 두개의 링크가 존재한다. 그림 3에 나타난것 처럼 아래쪽 절반의 입력단자  $L [V_{1,n-1}]$  ( $Q'_{n-1}$ 을 구성하는  $N/2$  노드)에 있는 모든 토큰은  $(n-1)$ -D 연결상의 한 링크를 통해 위쪽 노드  $U [V_{1,n-2}]$  ( $Q'_{n-1}$ 에 해당하는 노드)로 전송된다.  $U [G_{n-1}]$ 에서 루팅이 이루어진 후에  $U [V_{2,0}]$  ( $Q^U_{n-1}$ ) 상의 각 노드에 있는 두개의 토큰은 먼저와 반대방향의 두 링크를 통하여  $L [V'_{2,0}]$  ( $Q^L_{n-1}$ ) 상의 대응 노드로 전송된다. 마지막으로  $L [V'_{1,n-1}]$  ( $Q^L_{n-1}$ ) 노드상의 두 토큰중 하나만이  $U [V'_{1,n-1}]$  ( $Q^U_{n-1}$ )로 전송된다. 따라서 루팅을 달성하기 위해서는  $(n-1)$ -D 연결만이 두개의 링크쌍을 필요로 하고 다른 차원의 연결들은 한개의 링크쌍을 필요로 한다. 변환된 그래프의 특성 5로 부터  $(n-1)$ -D 연결

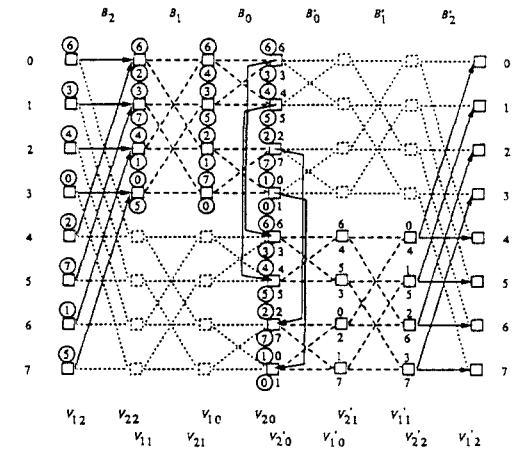


그림 5.  $H_U \cdot H'_L$ 에 적용된 순열 [6, 3, 4, 0, 2, 7, 1, 5]

Fig. 5. A permutation [6, 3, 4, 0, 2, 7, 1, 5] in  $H_U \cdot H'_L$ .

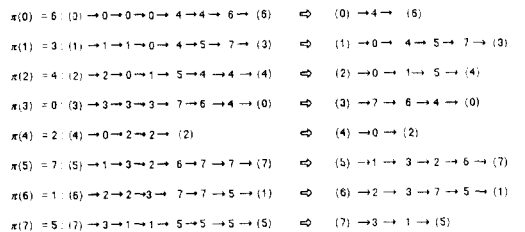


그림 6.  $H'_U$ 에서  $H'_L$ 으로의 사상과 가상링크 제거  
Fig. 6. Mapping from  $H'_U$  to  $H'_L$  and removing pseudo links.

대신 다른 연결을 사용할 수도 있다. Q.E.D.

그림 5는 그림 4의  $H_U \cdot H'_U$  상에서 사용된 순열을  $H_U \cdot H'_L$ 에 적용한 것이다. 그림 6의 좌측은  $H'_U$ 에서  $H'_L$ 로 사상한 결과이다. 그림에서 첫번째 괄호안의 수는 출발점 PE의 번지이고 마지막 괄호안의 수는 도착점 PE의 번지이다.  $p(4)$ 는 사이클을 형성함으로 사상되지 않았다. 그림의 우측은 가상 링크를 제거한 결과이다.

#### IV. 결론

임의의 순열을 구현하기 위해서 우리는 세가지 이중 링크 하이퍼큐브 구조와 루팅 알고리즘을 제안했다. 두번째 이중 링크 하이퍼큐브 모델은 가장 많은 링크가 필요하지만 두개의 독립적인 순열을 위한 루트를 동시에 제공한다. 세번째 이중 링크 하이퍼큐브는 세가지 중에 가장 경제적인 구조이나 가장 긴 통신 경로가 필요하다. 이들 루팅 알고리즘의 주된 단점은 출발점으로 부터 도착점까지 최단 경로를 제공할 수 없다는 것이다. 설정 경로의 최대 길이는 하이퍼큐브 구조 3에서  $2\log_2 N + 1$  이다. 하이퍼큐브에서 가능한 최단거리 경로  $\log_2 N$  과 비교하면 약 두배 길다. 그러나 회선교환방식에서 링크의 길이는 시스템 성능에서 무시할 수 있다. 이유는 전기적 전송 지연을 제외하면 중간 노드에서의 시간 지연이 없기 때문이다.

본 논문에서 제안된 하이퍼큐브 구조의 하드웨어 구현은 경제적이지만 중앙 집중식 루팅 제어를 필요로 한다. 병렬처리 컴퓨터 시스템에서 루팅 알고리즘이 집중 제어를 요구하는 것은 전체적인 성능에 장애가 될 수 있다. 그러나 병렬처리 알고리즘에서 작업들이 분산되어서 PE간에 수행되는 경우, 그 작업을 위한 PE간의 통신 형태는 일반적으로 고정되어 작업이 끝날 때까지 유지된다. 그래프를 이용하여 분산 및 병렬처리 시스템을 위한 알고리즘을 표현하는 경우 일반적으로 선형배열(linear array), 트리(tree), 망사(mesh)등의 위상을 사용하는 것으로도 알 수 있다. 따라서 연결망이 재배열 가능하면 루팅 정보(routing information)와 함께 수행되어야 할 작업을 각 PE로 분배하면, 매우 적은 초기부담(initial overhead)으로 높은 성능을 얻을 수 있다. 결론적으로 이상과 같이 제안된 이중 링크 하이퍼큐브 구조는 병렬처리와 배열 프로세서(array processor)에서 모든 순열에 대하여 블로킹 없는 통신을 가능케하는 홀

륭한 구조이다.

#### 參 考 文 獻

- [1] S. L. Johnsson and C. -T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Comput.*, vol. 38, pp. 1249-1268, Sept. 1989.
- [2] P. Ramanathan and K. G. Shin, "Reliable broadcast in hypercube multicomputers," *IEEE Trans. Comput.*, vol. C-37, pp. 1654-1657, Dec. 1988.
- [3] H. Sullivan and T. R. Bashkow, "A large scale, homogeneous, fully distributed parallel machine, I," *Proc. IEEE 1977 Comput. Arch.*, pp. 105-117, 1985.
- [4] L. G. Valiant, "A scheme for fast parallel communication," *SIAM J. Comput.*, vol. 11, pp. 350-361, May 1982.
- [5] Y. Saad and M.H. Schultz, "Topological properties of hypercubes," *IEEE Trans. Comput.*, vol. C-37, pp. 867-872, July 1988.
- [6] V. E. Benes, "Mathematical theory of connecting networks and telephone traffic," New York: Academic, 1965.
- [7] H. S. Stone, "High-performance computer architecture," Reading, MA: Addison-Wesley, 1990.
- [8] T. Szymanski, "On the permutation capability of a circuit-switched hypercube," *Proc. 1989 Int. Conf. Parallel Processing*, vol. I, pp. 103-110, Aug. 1989
- [9] A. Y. Oruc and M. Mittal, "Setup algorithms for cube-connected parallel computers using recursive Karnaugh maps," *IEEE Trans. Comput.*, vol. 40, pp. 217-221, Feb. 1991.
- [10] A. Waksman, "A permutation network," *J. ACM*, vol. 15, pp. 159-163, Jan. 1968.



- [11] K. Y. Lee, "On the rearrangeability of  $2(\log_2 N) - 1$  stage permutation networks," *IEEE Trans. Comput.*, vol. C-34, pp. 412-425, May 1985.

---

著 者 紹 介



崔 相 昉 (正會員)

1954年 9月 11日生. 1981年 2月 한양대학교 전자공학과 졸업. 1988年 3月 University of Washington 졸업(공학석사). 1990年 8月 University of Washington 졸업(공학박사). 1981年~1986年 금성정보통신(주)근무. 1991年~ 현재 인하대학교 전자공학과 조교수. 주관심분야는 컴퓨터 구조, 병렬 및 분산처리 시스템, Fault-tolerant computing 등임.