

다중 메시의 꼬인 큐브에 대한 임베딩

(Embedding Multiple Meshes into a Twisted Cube)

김 속 연 [†]

(Sook-Yeon Kim)

요약 꼬인 큐브는 병렬처리 시스템의 상호연결망으로서 널리 알려진 하이퍼큐브와 많은 면에서 비슷하면서도 절반 정도의 지름을 가지는 등 개선된 망 성질들을 가지므로 각광받아 왔다. 본 논문에서는 크기 $2^n \times 2^m$ 인 메시의 복사본 2^{n-1} 개가 꼬인 큐브에 노드 중복없이 연장을 1로 임베딩될 수 있음을 보인다 ($1 \leq n \leq m$). 여기서 확장율은 m 이 짝수이면 1이고 홀수이면 2이다.

키워드 : 꼬인 큐브, 메시, 임베딩, 연장율, 확장율

Abstract The twisted cube has received great attention because it has several superior properties to the hypercube that is widely known as a versatile parallel processing system. In this paper, we show that node-disjoint 2^{n-1} meshes of size $2^n \times 2^m$ can be embedded into a twisted cube with dilation 1 where $1 \leq n \leq m$. The expansion is 1 for even m and 2 for odd m .

Key words : twisted cube, mesh, embedding, dilation, expansion

1. 서론

그래프 임베딩은 병렬 구조간의 시뮬레이션이나[1,2], 병렬처리시스템의 병렬 알고리즘 수행[3,4], VLSI 칩 설계[5-7] 등을 위해서 필수적인 기술이다. 또한 그래프 임베딩은 작업을 병렬처리 시스템에 할당하는 데에도 활용되어 왔다. 하나의 작업을 구성하는 부작업들(subtasks)과 그들간의 통신을 그래프 G 로 나타내고 병렬 처리 시스템의 프로세서들과 그들간의 연결 상태를 그래프 H 로 나타낸다. 그런 후 그래프 G 를 그래프 H 에 효율적으로 임베딩하면 부작업들은 병렬처리시스템에 효과적으로 할당될 수 있다. 그런데 통상 다수 개의 작업들이 동시에 하나의 병렬 처리 시스템에서 수행되므로 다수 개의 작업을 나타내는 그래프 G 를 임베딩하는 것이 필요하다 [8,9]. 다수 개의 작업을 나타내는 그래프 G 는 다수 개의 연결요소를 가지는 그래프로 볼 수 있다. 특히 그래프 G

에 속하는 다수 개 연결 요소가 서로 동형인 경우는 병렬 처리 시스템의 고장에 대한 신뢰도를 높이는 방안으로 활용될 수도 있다. 왜냐하면 한 작업을 병렬 처리시스템에서 중복 수행하는데 적용될 수 있기 때문이다.

주어진 그래프 G 에 대해서 $V(G)$ 와 $E(G)$ 를 그래프 G 의 노드 집합과 에지 집합이라고 하자. 주어진 두 개의 그래프 G 과 H 에 대해서 그래프 G 의 그래프 H 에 대한 임베딩은 노드 집합 $V(G)$ 의 노드 집합 $V(H)$ 에 대한 단사 함수 ψ 이다($\psi: V(G) \rightarrow V(H)$). 여기서 그래프 G 를 손님 그래프 (guest graph)라 하고 그래프 H 를 주인 그래프 (host graph)라고 한다. 어떤 작업을 나타내는 그래프의 복사본 x 개를 노드 중복없이 임베딩한다고 함은 x 개의 연결요소(connected component)를 가지는 손님 그래프를 임베딩함을 의미한다.

임베딩의 성능을 측정하는 척도에는 연장율(dilation)과 확장율(expansion) 등이 있다. 주인 그래프가 프로세서간의 연결 상태를 나타낼 때 임베딩의 연장율은 통신 지연(communication delay)을 측정하는 척도가 되고 확장율은 프로세서 이용도(processor utilization)를 측정하는 척도가 된다. 손님 그래프 G 의 에지 (u, v) 에 대해서 에지 (u, v) 의 연장율은 두 노드 $\psi(u)$ 와 $\psi(v)$ 간의 최단 거리이다. 임베딩 ψ 의 연장율은 에지 집합 $E(G)$ 에 속한 에지들의 연장율 중 최댓값이다. 확장율은 주인 그래프와 손님 그래프의 노드 개수의 비율 $|V(H)|/|V(G)|$ 이다.

[†] 정 회 원 : 한경대학교 컴퓨터공학과 교수
entpentp@paran.com
논문접수 : 2008년 12월 4일
심사완료 : 2009년 5월 3일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 시스템 및 이론 제37권 제2호(2010.4)

한편 병렬 처리 분야에서 매우 유명한 상호연결망인 하이퍼큐브는 단순한 구조, 작은 지름, 정규 분지수, 대칭성, 고장 감내성 등의 유용한 성질들을 가지고 있다. 하이퍼큐브의 변형으로서 제안된 꼬인 큐브(twisted cube)는 하이퍼큐브와 많은 면에서 비슷하면서도 절반 정도의 지름을 가지는 등 개선된 망 성질들을 가지므로 각광 받게 되었다[10-12]. 완전이진트리를 꼬인 큐브에 임베딩하는 연구 결과가 발표되었고[13] 꼬인 큐브가 pancyclic이면서[12] 에지-pancyclic임도 알려졌다[14]. 고장인 꼬인 큐브에 Hamiltonian 사이클이나 경로를 임베딩하는 연구 결과들도 발표되었다[15-18]. 꼬인 큐브의 t/k -diagnosability는 하이퍼큐브와 같고[19] 짝수인 $m \geq 0$ 에 대하여 크기가 2×2^m 인 메쉬는 연장을 1과 확장을 1로 꼬인 큐브에 임베딩된다[9]. 또한 짝수 $m \geq 2$ 에 대하여 크기 4×2^m 인 메쉬의 복사본 두 개가 꼬인 큐브에 연장을 1, 확장을 1로 임베딩됨도 알려졌는데 이 연구결과는 크기가 4×2^m 인 메쉬가 연장을 1, 확장을 2로 임베딩됨을 의미하기도 한다[9]. 이와 더불어 최근에 m 이 짝수일 경우에 크기 $2^n \times 2^m$ 인 메쉬가 꼬인 큐브에 연장을 1, 확장을 2^{n-1} 로 임베딩되고 m 이 홀수일 경우엔 연장을 1 확장을 2^n 으로 임베딩됨이 알려졌다($1 \leq n \leq m$)[20]. 그러나 양변의 길이가 모두 8 이상인 메쉬의 복사본 다수개가 꼬인 큐브에 연장을 1로 임베딩되는지는 알려진 바가 없다.

본 논문에서는 크기 $2^n \times 2^m$ 인 메쉬의 복사본 2^{n-1} 개가 꼬인 큐브에 노드 중복없이 연장을 1로 임베딩될 수 있음을 보인다($1 \leq n \leq m$). 제시하는 임베딩의 확장율은 m 이 짝수이면 1이고 홀수이면 2이다. 2 절에서는 꼬인 큐브와 메쉬를 정의하고 3 절에서는 메쉬를 꼬인 큐브에 재귀적으로 임베딩함으로써 크기 $2^n \times 2^m$ 인 메쉬의 복사

본 2^{n-1} 개가 꼬인 큐브에 연장을 1로 임베딩될 수 있음을 보인다. 4 절에서는 결론을 맺는다.

2. 꼬인 큐브와 메쉬의 정의

이 절에서는 꼬인 큐브와 메쉬를 정의한다. 꼬인 큐브를 정의하기 위해서 먼저 패러티 함수를 소개한다.

정의 1. 길이가 p 인 이진스트링 $u = u_{p-1}u_{p-2} \dots u_1u_0$ 에 대하여 연산자 \oplus 가 exclusive-or일 때 패러티 함수 $\theta(u, i)$, $0 \leq i \leq p$, 는 $u_i \oplus u_{i-1} \oplus \dots \oplus u_1 \oplus u_0$ 이다.

정의 1의 패러티 함수를 이용하여 p -차원 꼬인 큐브 Q_p 를 재귀적으로 정의한다. 임의의 비트 u_i 의 보수를 u'_i 라 하자.

정의 2. p -차원 꼬인 큐브 Q_p 의 각 노드는 길이가 p 인 이진스트링으로 표현된다. 꼬인 큐브 Q_1 은 두 개의 노드 0과 1을 가지는 완전그래프이다. 홀수 $p \geq 3$ 에 대해서 p -차원 꼬인 큐브 Q_p 는 네 개의 부큐브 Q_{p-2}^{00} , Q_{p-2}^{01} , Q_{p-2}^{10} , Q_{p-2}^{11} 로 구성된다. 여기서 Q_{p-2}^{ij} , $i, j \in \{0,1\}$, 는 Q_{p-2} 와 동형이다. 노드 집합 $V(Q_p)$ 는 $\cup_{i,j \in \{0,1\}} V(Q_{p-2}^{ij})$ 이고 에지 집합 $E(Q_p)$ 는 $\dot{\cup}(\cup_{i,j \in \{0,1\}} E(Q_{p-2}^{ij}))$ 이다. 꼬인 큐브 Q_p 의 두 노드 $u = u_{p-1}u_{p-2} \dots u_1u_0$ 와 $v = v_{p-1}v_{p-2} \dots v_1v_0$ 에 대하여 다음 세가지 조건 중 하나를 만족하면 $(u, v) \in \dot{E}$ 이다.

1. $u = v'_{p-1}v'_{p-2}v'_{p-3} \dots v_1v_0$
2. $u = v'_{p-1}v'_{p-2}v_{p-3} \dots v_1v_0$ 이고 $\theta(u, p-3) = 0$
3. $u = v_{p-1}v'_{p-2}v'_{p-3} \dots v_1v_0$ 이고 $\theta(u, p-3) = 1$

정의 2에 따르면 p -차원 꼬인 큐브 Q_p 는 2^p 개의 노드와 $p2^{p-1}$ 개의 에지를 가진다. 그림 1에 꼬인 큐브 Q_5 가 그려져 있다[17]. 앞으로 꼬인 큐브 Q_p 를 간단히 큐브 Q_p

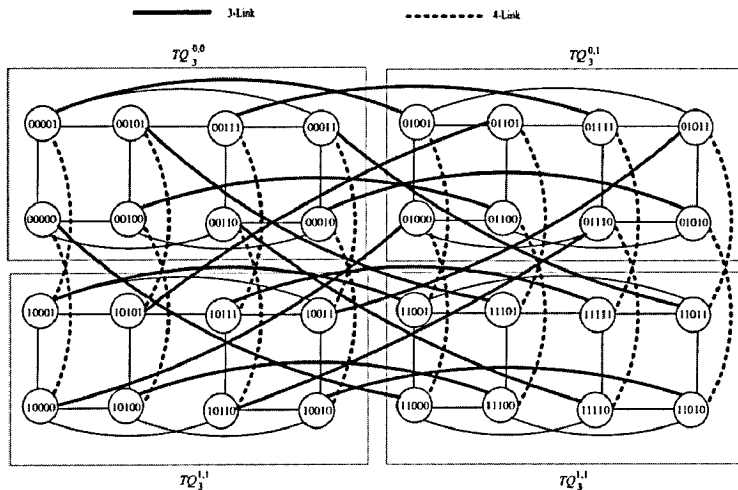


그림 1 꼬인 큐브 Q_5

라고도 하고 꼬인 큐브의 노드를 큐브노드라고도 하겠다.
 이제 메쉬를 정의한다. 그림 2에 크기 3×4 인 메쉬가 나타나 있다.

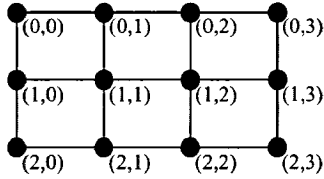


그림 2 크기 3×4 인 메쉬

정의 3. 크기 $N \times M$ 인 메쉬는 다음과 같은 노드 집합 V 와 에지 집합 E 를 가진다.

$$V = \{(i, j) | 0 \leq i < N, 0 \leq j < M\}$$

$$E = \{((i, j), (i', j)) | (i, j) \in V, (i', j) \in V, i + 1 = i'\}$$

$$\cup \{((i, j), (i, j')) | (i, j) \in V, (i, j') \in V, j + 1 = j'\}$$

특별한 언급이 없는 한 본 논문에서 다루는 모든 임베딩의 주인 그래프는 꼬인 큐브이고 연장율은 1이다. 따라서 크기 $N \times M$ 인 메쉬의 복사본 γ 개가 확장을 e 로 임베딩된다고 함은 $NMr -$ 노드 손님 그래프가 $NMre -$ 노드 꼬인 큐브에 연장율 1로 임베딩됨을 의미한다.

3. 메쉬의 꼬인 큐브에 대한 재귀적 임베딩

이 절에서는 메쉬를 꼬인 큐브에 임베딩하는 재귀적인 방법을 제시함으로써 크기 $2^n \times 2^m$ 인 메쉬의 복사본 2^{n-1} 개가 m 이 짝수(홀수)이면 확장을 1(확장을 2)로 임베딩됨을 보인다($1 \leq n \leq m$). 이 절의 재귀적인 방법은 하나의 메쉬를 임베딩하는 재귀적인 방법[20]을 확장한 것이다.

그림 3은 크기 $A \times B$ 인 메쉬가 확장을 h 로 임베딩될 때 크기 $2A \times B$ 인 메쉬의 복사본 두 개를 확장을 h 로 임베딩하는 방법을 보여준다. 그림 3(a)에는 크기 $A \times B$ 인 메쉬의 상하가 구분되도록 그려져 있다. 그림 3(a)와 같은 크기 $A \times B$ 인 메쉬가 확장을 h 로 임베딩된다고 하자. 그러면 크기 $A \times B$ 인 메쉬의 각 노드는 $\log_2 ABh$ 비트의 주소를 갖게 된다. 그림 3(b)에는 크기 $A \times B$ 인 메쉬 두 개를 상하 대칭적으로 붙여 형성한 크기 $2A \times B$ 인 메쉬가 그려져 있다. 크기 $A \times B$ 인 메쉬 두 개를 붙일 때 추가된 에지들은 곡선으로 그려져 있다. 그림 3(b)의 크기 $2A \times B$ 인 메쉬의 각 노드에는 이미 $\log_2 ABh$ 비트의 주소가 할당되어 있는데 크기 $A \times B$ 인 메쉬 두 개를 대칭적으로 붙였으므로 추가된 에지의 양끝 노드의 주소는 서로 같게 된다. 이제 그림 3(b)의 크기 $2A \times B$ 인 메쉬의 각 노드의 주소의 앞에 두 비트씩 덧붙이되 상반부 노드들에는 모두 00을 하반부

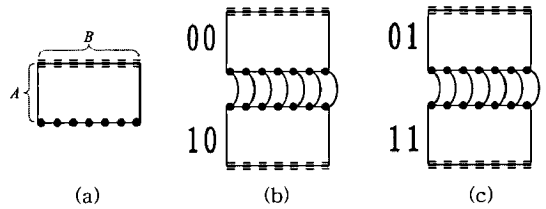


그림 3 재귀적으로 임베딩하는 방법: (a) 크기 $A \times B$ 인 메쉬의 임베딩, (b) 크기 $2A \times B$ 인 메쉬의 첫번째 임베딩 (c) 크기 $2A \times B$ 인 메쉬의 두번째 임베딩

노드들에는 모두 10을 덧붙인다. 그러면 크기 $2A \times B$ 인 메쉬의 각 노드는 $(\log_2 ABh + 2)$ 비트의 서로 다른 주소를 갖게 된다. 이 주소대로 크기 $2A \times B$ 인 메쉬를 꼬인 큐브에 임베딩하면 크기 $2A \times B$ 인 메쉬에서 인접한 노드들은 인접한 큐브노드들에 대응함을 쉽게 확인할 수 있다.

한편 그림 3(c)에도 크기 $A \times B$ 인 메쉬 두 개를 상하 대칭적으로 붙여 형성한 크기 $2A \times B$ 인 메쉬가 그려져 있다. 그림 3(c)의 메쉬의 각 노드에는 이미 $\log_2 ABh$ 비트의 주소가 할당되어 있는데 이 주소의 앞에 두 비트씩 덧붙이되 상반부 노드들에는 모두 01을 하반부 노드들에는 모두 11을 덧붙인다. 이 주소대로 그림 3(c)의 메쉬를 임베딩하면 크기 $2A \times B$ 인 메쉬에서 인접한 노드들은 인접한 큐브노드들에 대응함을 쉽게 확인할 수 있다.

그림 3(b)와 그림 3(c)와 같이 크기 $2A \times B$ 인 메쉬의 복사본 두 개를 $(\log_2 ABh + 2)$ -차원 꼬인 큐브에 임베딩하면 손님 그래프의 노드 수는 $4AB$ 이고 주인 그래프의 노드수는 $4ABh$ 이다. 따라서 확장을 h 이다. 여기서 크기 $A \times B$ 인 메쉬의 복사본 두 개의 각 노드는 서로 다른 큐브노드에 대응한다. 왜냐하면 첫 두 비트는 그림 3(b)와 그림 3(c)와 같고 나머지 비트들은 크기 $A \times B$ 인 메쉬를 임베딩할 때 사용한 주소이기 때문이다. 결과적으로 크기 $A \times B$ 인 메쉬가 확장을 h 로 임베딩되면 크기 $2A \times B$ 인 메쉬의 복사본 두 개가 확장을 h 로 임베딩된다.

앞에서 제시한 재귀적인 임베딩 방법의 예가 그림 4에 나타나 있다. 그림 4(a)는 크기 2×4 인 메쉬가 확장을 1로 임베딩됨을 보여준다. 그림 4(a)의 메쉬 두 개를 상하 대칭적으로 붙인 뒤 상반부에는 00을 하반부에는 10을 붙인 결과가 그림 4(b)에 나타나 있다. 그림 4(a)의 메쉬 두 개를 상하 대칭적으로 붙인 뒤 상반부에는 01을 하반부에는 11을 붙인 결과가 그림 4(c)에 나타나 있다. 그림 4는 크기 2×4 인 메쉬가 확장을 1로 임베딩될 때 크기 4×4 인 메쉬의 복사본 두 개가 확장을 1로 임베딩될 수 있음을 보여준다. 이러한 재귀적인 임베딩

방법으로부터 다음과 같은 소정리를 얻을 수 있다.

소정리 1. 크기 $N \times M$ 인 메쉬가 확장율 e 로 임베딩 된다면 크기 $2^i N \times M$ 인 메쉬의 복사본 2^i 개는 확장율 e 로 임베딩된다.

증명. 정수 i 에 관한 수학적 귀납법으로 증명하겠다. $i=0$ 일 때 본 소정리가 성립한다. 본 소정리가 $i=j$ 일 때 성립한다고 가정하고 $i=j+1$ 일 때 성립함을 보이겠다. 크기 $2^j N \times M$ 인 메쉬의 복사본 2^j 개가 확장율 e 로 임베딩된다고 가정하자. 크기 $2^j N \times M$ 인 메쉬의 복사본들을 $T_j(k)$, $0 \leq k < 2^j$, 라 하자. 그러면 메쉬 $T_j(k)$ 의 각 노드는 $\log_2(2^{2j} NMe)$ 비트의 주소를 가지고 있다. 메쉬 $T_j(0)$ 두 개를 그림 5의 맨 왼쪽 메쉬와 같이 상하 대칭적으로 붙여 크기 $2^{j+1} N \times M$ 인 메쉬 $T_{j+1}(0)$ 을 형성하도록 에지들을 추가한다. 그리고 각 노드의 $\log_2(2^{2j} NMe)$ -비트 주소의 앞에 두 비트씩 덧붙이되 상반부의 노드들에는 모두 00을 하반부의 노드들에는 모두 10 붙인다. 그러면 메쉬 $T_{j+1}(0)$ 의 각 노드들은 서로 다른 새로운 큐브노드 주소를 갖게 되는데 이 주소는 길이가 $\log_2(2^{2j+1} NMe)+2$ 이다. 이 주소대로 $T_{j+1}(0)$ 을 꼬인 큐브에 임베딩하면 메쉬 $T_{j+1}(0)$ 에서 인접한 노드들은 인접한 큐브노드들에 대응함을 쉽게 확인해 볼 수 있다.

한편 그림 5의 오른쪽에서 세번째 메쉬와 같이 메쉬 $T_j(0)$ 두 개를 상하 대칭적으로 붙여 크기 $2^{j+1} N \times M$ 인 메쉬 $T_{j+1}(2^j)$ 을 형성하도록 에지들을 추가한다. 그리고 각 노드의 $\log_2(2^{2j} NMe)$ -비트 주소의 앞에 두 비트씩 덧붙이되 상반부(하반부)의 노드들에는 모두 01(11)을 붙인다. 이 주소대로 $T_{j+1}(2^j)$ 을 꼬인 큐브에 임베딩하면 메쉬 $T_{j+1}(2^j)$ 에서 인접한 노드들은 인접한 큐브노드들에 대응함을 쉽게 확인해 볼 수 있다.

그림 5에서 보는 바와 같이 메쉬 $T_j(0)$ 외에 나머지

복사본 메쉬 $T_j(k)$, $1 \leq k < 2^j$ 에 대해서도 같은 방법으로 $T_{j+1}(k)$ 와 $T_{j+1}(2^j+k)$ 를 임베딩한다. 그러면 모든 k , $0 \leq k < 2^{j+1}$ 에 대해서 메쉬 $T_{j+1}(k)$ 를 $(\log_2(2^{2j+1} NMe)+2)$ -차원 꼬인 큐브에 임베딩하게 된다. 손님 그래프의 노드 수는 $2^{2(j+1)} NM$ 이고 주인 그래프의 노드수는 $2^{2(j+1)} NMe$ 이므로 확장율은 e 이다. 또한 모든 k , $0 \leq k < 2^{j+1}$ 에 대해서 메쉬 $T_{j+1}(k)$ 의 각 노드는 서로 다른 큐브노드에 대응한다. 왜냐하면 첫 두 비트는 그림 5와 같고 나머지 비트들은 메쉬 $T_j(k)$ 를 임베딩할 때 사용한 주소이기 때문이다. 결과적으로 크기 $2^{j+1} N \times M$ 인 메쉬의 복사본 2^{j+1} 개가 확장율 e 로 임베딩된다. □

소정리 1과 기존의 연구결과[9]로부터 다음과 같은 소정리를 얻을 수 있다.

소정리 2. 짝수인 m 에 대하여 크기 $2^n \times 2^m$ 인 메쉬의 복사본 2^{n-1} 개는 확장율 1로 임베딩된다($1 \leq n \leq m$).

증명. 홀수인 $p \geq 1$ 에 대하여 크기 $2 \times 2^{p-1}$ 인 메쉬는 확장율 1로 임베딩된다[9]. 그러면 소정리 1에 의해서 홀수인 $p \geq 1$ 에 대하여 크기 $(2^i \cdot 2) \times 2^{p-1}$ 인 메쉬의 복사본 2^i 개는 확장율 1로 임베딩된다($i \geq 0$). i 를 $n-1$ 으로 치환하고 p 를 $m+1$ 로 치환하면 본 소정리를 얻을 수 있다. □

소정리 2는 한쪽 변이 2의 짝수승 2^m 인 메쉬들만 다루었고 다음 정리에서 나머지 메쉬들도 함께 다룬다.

정리 1. 크기 $2^n \times 2^m$ 인 메쉬의 복사본 2^{n-1} 개는 m 이 짝수이면 확장율 1로 m 이 홀수이면 확장율 2로 임베딩된다($1 \leq n \leq m$).

증명. m 이 짝수이면 소정리 2에 의해서 본 정리가 성립하므로 m 이 홀수인 경우만 증명하면 된다. m 이 홀수이면 크기 $2^n \times 2^{m-1}$ 인 메쉬의 복사본 2^{n-1} 개는 소정리 2에 의해서 확장율 1로 임베딩된다. 그런데 크기 $2^n \times 2^m$ 인 메쉬는 크기 $2^n \times 2^{m-1}$ 메쉬에 연장율 1과 확장율 2로 임베딩된다. 따라서 크기 $2^n \times 2^m$ 인 메쉬의 복사본 2^{n-1}

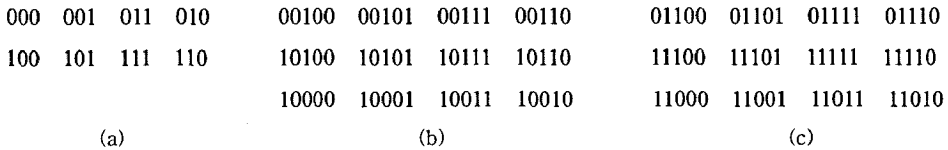


그림 4 재귀적으로 임베딩하는 방법의 예: (a) 크기 2×4 인 메쉬의 임베딩 (b) 크기 4×4 인 메쉬의 첫번째 임베딩 (c) 크기 4×4 인 메쉬의 두번째 임베딩

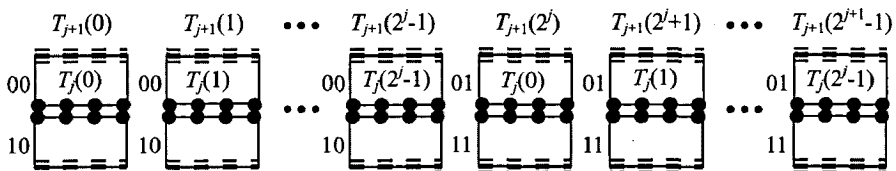


그림 5 메쉬 $T_{j+1}(k)$, $0 \leq k < 2^{j+1}$, 의 임베딩

개는 확장을 2로 꼬인 큐브에 임베딩된다. □ □

정리 1로부터 다음과 같은 따름정리들을 얻을 수 있다.

따름정리 1. 크기 4×2^m 인 메시의 복사본 두 개는 m 이 짝수이면 확장을 1로 m 이 홀수이면 확장을 2로 임베딩된다($m \geq 2$).

따름정리 1에 제시된 내용 중 m 이 짝수인 경우에 대해서는 Lai와 Tsai에 의해서 이미 알려진 바 있다[9].

따름정리 2. 크기 $2^n \times 2^m$ 인 메시의 복사본 2^{n-1} 개는 m 이 짝수이면 큐브 $Q_{2^{n+m-1}}$ 에 m 이 홀수이면 $Q_{2^{n+m}}$ 에 임베딩된다($1 \leq n \leq m$).

4. 결론

본 논문에서는 크기 $2^n \times 2^m$ 인 메시의 복사본 2^{n-1} 개가 m 이 짝수(홀수)이면 꼬인 큐브 $Q_{2^{n+m-1}}$ (꼬인 큐브 $Q_{2^{n+m}}$)에 연장을 1로 임베딩될 수 있음을 보인다($1 \leq n \leq m$). 여기서 m 이 짝수이면 확장은 1이고 홀수이면 2이다. 한편 꼬인큐브를 비롯하여 교차큐브나 피비우스 큐브를 포함하는 BC 그래프[19]에 대해서도 본 논문의 연구결과가 확장될 수 있는지 밝히는 것은 향후 연구 과제이다.

참 고 문 헌

[1] B. Monien, H. Sudborough, "Embedding one inter-connection network in another," pp.257-282, Springer-Verlag/Wien, 1990. Computing Supplementum 7: Computational Graph Theory.

[2] A. Rosenberg, "Issues in the study of graph embeddings," *Lecture Notes in Computer Science*, Springer-Verlag, New York, vol.100, pp.150-176, 1981.

[3] F. Berman, L. Snyder, "On mapping parallel algorithms into parallel architectures," *J. Parallel Distrib. Comput.*, vol.4, no.5, pp.439-458, 1987.

[4] V. Chaudhary, J.K. Aggarwal, "Generalized mapping of parallel algorithms onto parallel architectures," *Proc. Int'l Conf. Parallel Processing*, pp. 137-141, Aug. 1990.

[5] S. L. Bezrukov, J. D. Chavez, L. H. Harper M. Röttger, U.-P. Schroeder, "The congestion of n-cube layout on a rectangular grid," *Discrete Math*, vol.213, no.1-3, pp.13-19, Feb. 2000.

[6] A. Matsubayashi, "VLSI layout of trees into grids of minimum width," *IEICE Trans. Fundamentals*, vol.E87-A, no.5, pp.1059-1069, May 2004.

[7] A. Patel, A. Kusalik, C. McCrosky, "Area-efficient VLSI layouts for binary hypercubes," *IEEE Trans. Computers*, vol.49, no.2, pp.160-169, Feb. 2000.

[8] Q. Dong, X. Yang, J. Zhao and Y. Y. Tang, "Embedding a family of disjoint 3D meshes into a crossed cube," *Information Sciences*, vol.178, Issue 11, pp.2396-2405, June 2008.

[9] C.-J. Lai, C.-H. Tsai, "Embedding a family of

meshes into twisted cubes," *Information Processing Letters*, vol.108, Issue 5, pp.326-330, Nov. 2008.

[10] P. A. J. Hilbers, M. R. J. Koopman, J. L. A. van de Snelcheut, "The twisted cube," in *PARLE: Parallel Architectures and Languages Europe, Parallel Architectures*, vol.1, Springer, Berlin, pp.152-158, 1987.

[11] S. Abraham, K. Padmanabhan, "The twisted cube topology for multiprocessors: a study in network asymmetry," *J. Parallel Distrib. Comput.*, vol.13, Issue 1, pp.104-110, 1991.

[12] C.-P. Chang, J.-N. Wang, L.-H. Hsu, "Topological properties of twisted cube," *Information Sciences*, vol.113, pp.147-167, 1999.

[13] E. Abuelrub, S. Bettayeb, "Embedding complete binary trees into twisted hypercubes," *Proc. of the Int. Conf. on Computer Applications in Design, Simulation and Analysis*, Washington, D.C. March 10-12, pp.1-4, 1992.

[14] J. Fan, X. Lin, X. Jia, R. W. H. Lau, "Edge-pancyclicity of twisted cubes," *ISAAC 2005, Lecture Notes in Comput. Sci.*, vol.3827, pp.1090-1099, 2005.

[15] W.-T. Huang, J. J. M. Tan, C.-N. Hung, L.-H. Hsu, "Fault-tolerant hamiltonicity of twisted cubes," *J. Parallel Distrib. Comput.*, vol.62, no.4, pp.591-604, 2002.

[16] M.-C. Yang, T.-K. Li, J. J. M. Tan, L.-H. Hsu, "On embedding cycles into faulty twisted cubes," *Information Sciences*, vol.176, no.6, pp.676-690, 2006.

[17] J.-S. Fu, "Fault-free Hamiltonian cycles in twisted cubes with conditional link faults," *Theoretical Computer Science*, vol.407, no.1-3, pp.318-329, Nov. 2008.

[18] J. Fan, X. Lin, Y. Pan, X. Jia, "Optimal fault-tolerant embedding of paths in twisted cubes," *J. Parallel Distrib. Comput.*, vol.67, no.2, pp.205-214, 2007.

[19] J. Fan, X. Lin, "The t/k-diagnosability of the BC Graphs," *IEEE Trans. Computers*, vol.54, no.2, pp.176-184, 2005.

[20] S.-Y. Kim, "Embedding a mesh of size $2^n \times 2^n$ into a twisted cube," *The Transactions of the Korea Information Processing Society A*, vol.16-A, no.4, pp.223-226, 2009.



김 숙 연

1991년 연세대학교 전산학과(학사). 1993년 한국과학기술원(KAIST) 전산학과(공학석사). 1998년 한국과학기술원(KAIST) 전산학과(공학박사). 1998년 3월~2004년 2월 한국전자통신연구원(ETRI) 선임연구원. 2004년 3월~현재 한경대학교 컴퓨터공학과 교수. 관심분야는 병렬처리, 그래프 임베딩, 상호연결망, 네트워크 등