

다단 상호 연결 네트워크를 위한 효율적인 고장 진단에 관한 연구

A Study on Efficient Fault-Diagnosis for Multistage Interconnection Networks

배 성 환*, 김 대 익**, 이 상 태*, 전 병 실***
(Sung Hwan Bae*, Dae Ik Kim**, Sang Tae Lee*, Byoung Sil Chon***)

요 약

많은 수의 프로세서와 메모리 소자사이의 연결을 위한 구조들은 다단 상호연결 네트워크를 이용해서 구현할 수 있다. 또한 경제성, 처리능력 및 고장 허용면에서의 발전은 자연스럽게 컴퓨터 시스템 성장에 있어서 가장 중요한 요건이 되었다. 그러나 지금까지의 다단 상호연결 네트워크에서의 고장의 허용 방법, 특히 진단에 관한 연구가 미흡하다.

따라서 본 논문에서는 기존의 다단 상호연결 네트워크중에서 generalized cube네트워크를 바탕으로 링크 고착 고착 및 direct와 cross상태에서 스위칭 소자의 고장, 그리고 새로운 broadcast상태에서 고장진단을 포함하는 generalized cube네트워크에 스위칭 소자가 가지는 4가지 상태의 전체적인 진단을 한다. 가정된 고장모델을 검출할 수 있는 테스트 집합을 산출하고 이를 통해서 고장의 검출 및 위치를 결정할 수 있는 효율적인 알고리즘을 제안하고 적용의 예를 보인다.

ABSTRACT

In multiprocessor systems with multiple processors and memories, efficient communication between processors and memories is critical for high performance. Various types of multistage networks have been proposed. The economic feasibility and the improvements in both computing throughput and fault tolerance/diagnosis have been some of the most important factors in the development of these computer systems.

In this paper, we present an efficient algorithm for the diagnosis of generalized cube interconnection networks with a fan-in/fan-out of 2. Also, using the assumed fault model present total fault diagnosis by generating suitable fault-detection and fault-location test sets for link stuck fault, switching element fault in direct/cross states, including broadcast diagnosis methods based on some basic properties of generalized cube interconnection networks. Finally, we illustrate some example.

I. 서 론

지난 여러해 동안 디지털 컴퓨터 네트워크와 병렬처리 시스템에 빠른 발전을 보여 왔다. 또한 최근 VLSI 기술의 발전은 더 많은 수의 프로세서와 메모리 모듈을 가지는 멀티 프로세서와 멀티 컴퓨터의 사용을 가능하게 하였다. 특히 이러한 시스템에서 핵심적인 요소중의 하나는 프로세서들 상호간의 통신 및 프로세서와 메모리 모듈간의 통신을 가능하게 해주는 상호연결 네트워크에 있다고 볼 수 있다.^[1] 지금까지 여러가지 다양한 종류의 다단 상

호연결 네트워크가 제안되었다.^[1]

또한 경제성, 처리능력 및 고장 허용면에서의 발전은 자연스럽게 컴퓨터 시스템 성장에 있어서 가장 중요한 요건이 되었다. 이러한 컴퓨터의 성장에는 네트워크에서 구성요소의 고장시 적절한 대응이 이루어 지지 않는다면 시스템 전반에 관한 고장 혹은 수행능력의 저하를 가져올 수 있다.

고장 진단의 문제는 가정된 고장모델에서 모든 고장에 대하여 적절한 고장 검출 및 위치의 결정을 위한 테스트 집합을 산출함으로써 해결할 수 있다.

다단 상호연결 네트워크에서의 고장진단의 방법들이 여러 논문에서 제안되었는데,^{[2][3][4][5][6][7]} 특히 Wu와 Fen^[7]은 2×2 스위칭 소자의 16개의 가능한 상태를 기준으로 단일 및 다중 고장 검출을 위한 기본적 방법을 제시

*전북대학교 공과대학 전자공학과 박사과정
Dept. of Electronic Eng., Chonbuk Nat'l Univ.

**전북대학교 전기전자회로 합성연구소 연구원
Electrical Circuits and Systems research Institute.

***전북대학교 공과대학 전자공학과 교수
Dept. of Electronic Eng., Chonbuk Nat'l Univ.

접수일자: 1996년 6월 25일

하였다. Huang과 Lombardi[6]는 기존의 Wu와 Feng[7]의 진단 방법중 단일 응답 고장(one-response fault)의 경우에 새로운 방법을 제시하여 모든 경우에 있어서 전체적인 테스트 수를 네트워크의 크기에 관계없이 일정하게 하였다. 또한 Falavarjani와 Pradhan[11]은 2×2 스위칭 소자의 9개 상태를 고려하여 여러 다단 상호연결 네트워크에 적용할 수 있는 테스트 방법을 제시하였다. 이는 각 스테이지 스위칭 소자에 01 혹은 10을 적용할 수 있는 방법을 수학적으로 보이고 이를 이용하여 테스트할 수 있는 방법을 제시하였다.

Agrawal[10]은 제어라인(control line)의 라인 고착 고장과 온-라인 고장 진단의 기술을 제시하였다. 최근에 Lombardi [3]는 baseline네트워크에서 여러개의 고장을 포함하는 고장 진단(검출 및 위치)을 위한 새로운 알고리즘을 제시하였다. 4개의 가능한 전체 조건을 바탕으로 이 알고리즘은 $2(1 + \log_2 N)$ 개의 테스트를 요구한다.

그러나 지금까지의 다단 상호연결 네트워크에서 고장 허용 방법, 특히 진단에 관한 연구가 미흡하다. 따라서 본 논문에서는 기존의 다단 상호연결 네트워크중에서 generalized cube네트워크를 바탕으로 링크 고착 고장 및 direct와 cross상태에서 스위칭 소자의 고장, 그리고 broadcast 상태에서 고장 진단을 포함하는 generalized cube네트워크에 스위칭 소자가 가지는 4가지 상태의 전체적인 진단을 한다.

가정된 고장모델을 검출할 수 있는 테스트 집합을 산출하고 이를 통해서 고장의 검출 및 위치를 결정할 수 있는 효율적인 알고리즘을 소개한다. 나아가서 라인 고착 고장 및 스위칭 소자에 direct와 cross상태에서의 고장을 검출할 수 있는 테스트 수가 네트워크의 크기에 관계없이 일정한 크기를 가진다는 점을 보인다. 또한 기존의 2×2 스위치에서 고장진단에 broadcast에 진단방법을 제안하여 전체적인 고장진단 방법을 제시한다. 그리고 적용의 예를 보인다.

II. 스위칭 소자의 고장모델과 테스트 집합

고장진단의 문제는 가정된 고장모델에서 모든 고장에 관한 적절한 고장검출 및 고장위치 검출을 위한 테스트 집합을 산출함으로써 접근할 수 있다. 그리고 이러한 테스트집합이 최소값 혹은 거의 최소에 집합으로 줄여지게 된다. 따라서 먼저 본 논문에서는 스위칭 소자의 고장 모델을 먼저 제안하고 그러한 고장모델에서 모든 고장에 관한 테스트 집합을 유추한다.

그림 1에서 보여진것 처럼 direct와 cross, broadcast능력을 가지는 스위칭 소자로 구성된 네트워크를 위한 고장 진단 방법을 소개한다.

상호연결 네트워크에서의 고장은 링크 혹은 스위칭 소자에 놓여질 수 있다. 링크에 위치하는 고장은 라인 고착 종류중의 하나로 고려될 수 있는데, stuck-at-zero(s-a-0)

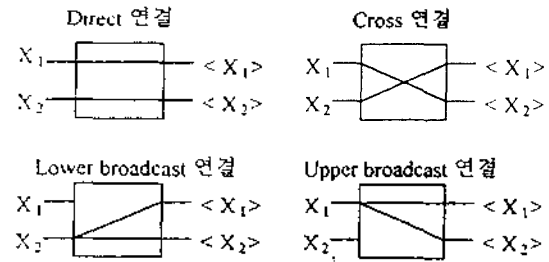


그림 1 스위칭 소자의 4가지 상태
Fig. 1 4-States of switching element

혹은 stuck-at-one(s-a-1)이다. 본 논문에서는 스위칭 소자에 고장종류를 고려하기 위해서 함수적인 접근과 스테이지별 테스트 방식을 사용한다. 일반적으로 두개의 입력라인과 출력라인을 가지는 스위칭 소자는 16개의 상태를 가지는 2×2 교차점 스위칭 매트릭스로 고려될 수 있다.

표 1은 16상태의 집합 S와 관련된 기호화된 표현을 보인다. 하나의 스위칭 소자는 16개 상태중의 하나에 있다고 가정한다. 스위칭 소자가 네트워크 기능을 수행하기 위해서 가정될 수 있는 정당한 상태의 수는 상호연결 네트워크의 요구에 달려있다. 표 2에는 정당한 direct상태(S10)에서 스위칭 소자를 통해 검출될 고장들과 먼저 그것을 검출하기 위한 테스트들이 나열되어 있다. 표 2에 부분(I)에는 링크 고착 고장의 검출이 서술되어있다. 링크라벨의 윗첨자는 고장이 링크 고착 0 또는 1을 일으키는지를 나타낸다. 스위칭 소자 고장의 검출은 부분(II)에 보여진다.

표 1. 2×2 스위칭 소자의 16개 상태의 표현
Table 1. 16-states representation of a 2×2 switching element

상태 이름	스위칭 소자 기호	상태 이름	스위칭 소자 기호
S0		S8	
S1		S9	
S2		S10	
S3		S11	
S4		S12	
S5		S13	
S6		S14	
S7		S15	

스위칭 소자의 고장에 관해서 예를들면 $S_{10}-S_3$ 는 입력 단자에 $(x_1, x_2) = (0, 1)$ 을 적용한다면 출력단자에서 고장 출력값은 정상적인 출력값 $(\langle x_1 \rangle, \langle x_2 \rangle) = (0, 1)$ 과는 다른 $(\langle x_1 \rangle, \langle x_2 \rangle) = (1, 0)$ 일 것이다.

표 2. 정당한 direct상태(S_{10})에서 테스트 입력과 출력값
Table 2. Fault inputs and outputs in valid direct state(S_{10})

고 장		테스트		출력값			
		X_1	X_2	정상		고장	
				$\langle X_1 \rangle$	$\langle X_2 \rangle$	$\langle X_1 \rangle$	$\langle X_2 \rangle$
부분 I. 링크 고착 고장	$X_1^0, \langle X_1^0 \rangle$	1	0	1	0	0	0
	$X_1^1, \langle X_1^1 \rangle$	0	0	0	0	1	0
	$X_2^0, \langle X_2^0 \rangle$	0	1	0	1	0	0
	$X_2^1, \langle X_2^1 \rangle$	0	0	0	0	0	1
부분 II. 스 위 칭 소 자 고 장	$S_{10}-S_0$	0	1	0	1	-	-
	$S_{10}-S_1$	0	1	0	1	1	-
	$S_{10}-S_2$	0	1	0	1	-	1
	$S_{10}-S_3$	0	1	0	1	1	1
	$S_{10}-S_4$	0	1	0	1	-	0
	$S_{10}-S_5$	0	1	0	1	1	0
	$S_{10}-S_6$	0	1	0	1	-	φ
	$S_{10}-S_7$	0	1	0	1	1	φ
	$S_{10}-S_8$	0	1	0	1	0	-
	$S_{10}-S_9$	0	1	0	1	φ	-
	$S_{10}-S_{11}$	0	1	0	1	φ	1
	$S_{10}-S_{12}$	0	1	0	1	0	0
	$S_{10}-S_{13}$	0	1	0	1	φ	0
	$S_{10}-S_{14}$	0	1	0	1	0	φ
	$S_{10}-S_{15}$	0	1	0	1	φ	φ

표 2에서 '-'는 정의되지 않은 출력값을, ' φ '는 0 과 1이 동시에 단자에 나타나는 논리적으로 에러있는 출력값들을 의미한다. 표를 통해서 '-'와 ' φ '에 0과 1의 임의에 할당은 정상적인 출력과 고장 출력 사이에 영향을 끼치지 않는다. 표 2로부터 두개의 테스트 $(x_1, x_2) = (0, 1)$ 과 (x_1, x_2)

표 3. 정당한 direct상태(S_5)에서의 테스트 입력과 출력값
Table 2. Fault inputs and outputs in cross state(S_5)

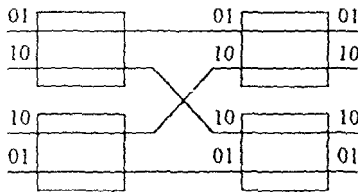
고 장		테스트		출력값			
		X_1	X_2	정상		고장	
				$\langle X_1 \rangle$	$\langle X_2 \rangle$	$\langle X_1 \rangle$	$\langle X_2 \rangle$
부분 I. 링크 고착 고장	$X_1^0, \langle X_1^0 \rangle$	1	0	0	1	0	0
	$X_1^1, \langle X_1^1 \rangle$	0	0	0	0	0	1
	$X_2^0, \langle X_2^0 \rangle$	0	1	1	0	0	0
	$X_2^1, \langle X_2^1 \rangle$	0	0	0	0	1	0
부분 II. 스 위 칭 소 자 고 장	S_5-S_0	0	1	1	0	-	-
	S_5-S_1	0	1	1	0	1	-
	S_5-S_2	0	1	1	0	-	1
	S_5-S_3	0	1	1	0	1	1
	S_5-S_4	0	1	1	0	-	0
	S_5-S_6	0	1	1	0	-	φ
	S_5-S_7	0	1	1	0	1	φ
	S_5-S_8	0	1	1	0	0	-
	S_5-S_9	0	1	1	0	φ	-
	S_5-S_{10}	0	1	1	0	0	1
	S_5-S_{11}	0	1	1	0	φ	1
	S_5-S_{12}	0	1	1	0	0	0
	S_5-S_{13}	0	1	1	0	φ	0
	S_5-S_{14}	0	1	1	0	0	φ
	S_5-S_{15}	0	1	1	0	φ	φ

$= (1, 0)$ 이 모든 고장을 검출하기에 충분하다. x_1 과 x_2 의 테스트 벡터는 각각 01과 10이다.

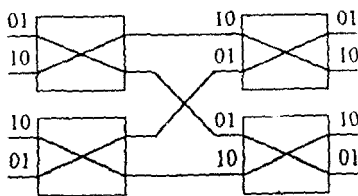
간편한 참조를 위해서 앞으로 $t=01$ 로 \bar{t} 는 10으로 정의한다. 스위칭 소자의 정당한 cross상태(S5)에 관해서는 표 3에 보인다.

Ⅲ. 고장 검출

네트워크를 위한 테스트 집합은 Feng와 Wu의 효율적인 알고리즘을 이용한다. 이는 네트워크에 각 스위칭 소자가 각기 두개의 테스트 벡터(t 와 \bar{t})를 가지고 라벨된 두개의 입력 라인을 가지도록 각 링크를 라벨하는 것이다. 연결 경로는 스위칭 소자를 테스트될 정당한 상태로 놓음으로서 설정될 수 있다. 그림 2에 보여진 스위칭 소자의 direct와 cross상태(S_{10} 과 S_{3})를 위해서는 두개의 테스트 단계가 필요하다.



(a) 페이즈 1 테스트(Phase 1 test)



(b) 페이즈 2 테스트(Phase 2 test)

그림 2 기본적 네트워크에서의 테스트 집합과 응답
Fig. 2 Test set and response for a basic network

테스트 페이즈 1과 2에서는 네트워크에 모든 스위칭 소자에 대한 상태를 각각 테스트한다. 크기가 N인 네트워크를 위한 효율적인 테스트 집합을 산출하기 위한 알고리즘은 다음과 같다.

단계 1: 테스트 벡터 $t=01$ 을 가지고 네트워크 왼쪽의 가장 위에 터미널 링크를 라벨 한다.

단계 2: 라벨된 터미널 링크는 위로부터 아래로 0, 1, ..., $m-1$ 로 표시하고 다음 라벨 되지않은 m개의 라벨을 위로부터 아래로 $m, m+1, \dots, 2m-1$ 로 표시한다고 가정한다.(단, $1 \leq m \leq N$ 이고, m은 2의 지수승이다.)

$0 \leq i \leq m-1$ 에 대해서 터미널 링크 $m+i$ 를 $L(i)$ 로 라벨한다.

(여기서 $L(i)$ 는 터미널 링크 i에 할당된 테스트 벡터이고, $L(i)$ 는 보수이다.)

단계 3: 좌측면에 라벨 되지않은 터미널 링크에 대해서 모든 N개의 터미널 이 라벨될때까지 단계 2를 반복한다.

Generalized cube네트워크에서 만약 스테이지 i ($0 \leq i \leq l, l = \log_2 N - 1$)에 N개 입력 라인을 $l-1$ 비트로 표시하다면, 다음과 같다.

페이즈 1 입력($P_l P_{l-1} \dots P_0$) $_0 = \sigma \beta_l \beta_{l-1} \dots \beta_1$
 여기서 $\sigma(P_l P_{l-1} \dots P_0) = (P_{l-1} P_{l-2} \dots P_0 P_l)$
 $\beta_{(k)}(P_l P_{l-1} \dots P_0) = (P_l \dots P_{k+1} P_0 P_{k-1} \dots P_1 P_k)$

페이즈 2 입력($P_l P_{l-1} \dots P_0$) $_{\bar{0}} = \bar{\sigma} \beta_l \beta_{l-1} \dots \beta_1$
 여기서 $\bar{\sigma}(P_l P_{l-1} \dots P_0) = (P_{l-1} P_{l-2} \dots P_0 \bar{P}_l)$
 $\beta_{\bar{k}}(P_l P_{l-1} \dots P_0) = (P_l \dots P_{k+1} P_0 P_{k-1} \dots P_1 \bar{P}_k)$

Ⅳ. 고장 위치

Generalized cube네트워크에서 단일 고장에 위치를 결정하는 문제는 크게 다음 세가지의 단계(가, 나, 다)로 구성될 수 있다. 하나는 링크에서 단일 고장 위치를 결정하는 문제이고, 두번째는 스위칭 소자에 direct와 cross상태에서 고장을 위치시키는 문제이다. 마지막으로 broadcast상태에서 고장 위치를 검출하는 문제이다. 따라서 4개의 상태를 가지는 스위칭 소자를 위한 전체적인 고장 진단을 위한 구성도는 그림 3와 같다.

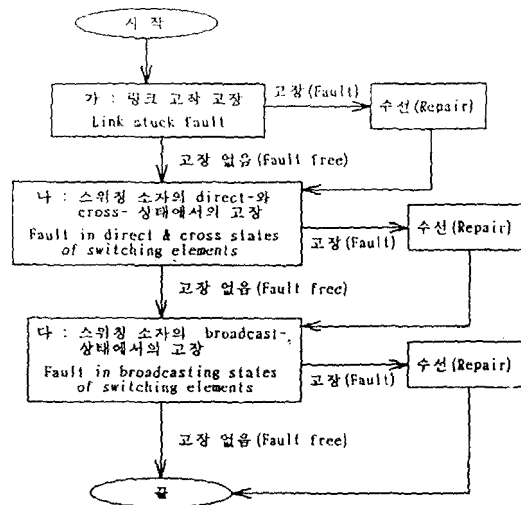
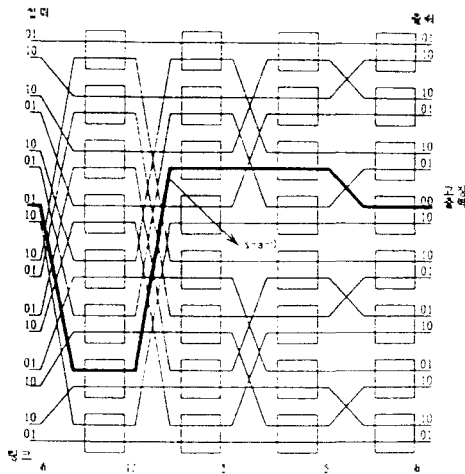


그림 3 고장 진단의 전체적인 구성도
Fig. 3 Total construction diagram of fault diagnosis

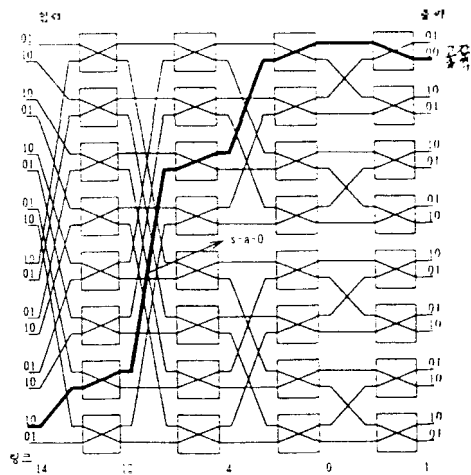
가. 링크 고착 고장(Link stuck fault)

하나의 경로에서 링크 집합(link set)을 계산하는 방법은 [5],[6]에서 보여진다. 단일 고장 검출을 위해서 유도된 테스트 집합은 링크에서의 단일 고착 고장을 위치 시키기 위해서 사용될 수 있다.

네트워크가 그림 4에서 보여지는 페이즈 1과 페이즈 2에 테스트 응답을 가진다고 가정하면, 고장이 없는 출력과 테스트 출력을 비교하면 페이즈 1 링크집합 {6, 12, 5, 5, 6} 경로와 페이즈 2 링크집합 {14, 12, 4, 0, 1} 경로에서 페이즈 1에서는 링크(0110)₄에 페이즈 2에는 링크(0001)₄에 고장출력 00이 있다. 이 두 링크집합을 교차시켜보면 링크(1100)₁에 고장을 위치 시킬 수 있으며 이는 0에 고착되어 있다.



(a) 페이즈1 테스트(Phase1 test)



(b) 페이즈2 테스트(Phase2 test)

그림 4 링크 고착 고장
Fig. 4 Link stuck fault

나. 함수적 접근 방식을 이용한 스위칭 소자 고장

스위칭 소자의 고장은 표 1에 보여진 16개 상태의 어느 하나의 결과일 수 있다. 네트워크에서 단일 스위칭 소자 고장은 여러 고장 출력 형태가 있는데 검출 페이즈의 응답 형태에 따라서 고장은 다음과 같이 네가지 경우로 분류될 수 있다.

이러한 분류는 Feng와 Wu의 방법에 따른다. 첫째, 단지 하나의 고장 출력이 나타나는 단일 응답 고장으로, 고장 출력은 페이즈 1 또는 페이즈 2에 있다. 둘째, 두개의 고장출력이 있는데, 분류된 두개의 응답 고장중 하나는 페이즈 1 테스트에 있고, 다른 하나는 페이즈 2 테스트에 있다. 셋째, 분류되지 않은 두개의 응답 고장출력이 있는데 두개의 고장출력이 모두 페이즈 1 또는 페이즈 2의 터미널 출력에 있다. 넷째, 여러개의 응답 고장으로서 두개의 고장 출력보다 더 있는 경우이다. 이 네가지 경우를 각각 generalized cube네트워크에 적용하면 결과는 다음과 같다.

(1)경 우 1

단일 응답 고장의 고장위치 검출과 종류는 네트워크 크기에 관계없이 10개의 테스트에 의해서 결정될 수 있다.

(2)경 우 2

이 경우에서 스위칭 소자는 단지 각 정당한 상태에서 하나의 고장 출력을 가졌다. 경우 1에서는 스위칭 소자에 direct와 cross상태에서 단지 하나의 고장출력이 나타나지만 경우 2에서 이 두 집합에 가능한 고장 출력을 표 4에 나타냈는데, 6개의 부분 A, B, C, D, E, F로 구성된다. 경우 2는 36개의 가능한 상태 결합이 있는데, 각 경우에 많아야 경우 A는 8개, 경우 B와 C는 10개, 경우 D와 E는 12개, 경우 F는 4개의 테스트로 결정될 수 있다.

표 4. 경우 2에서 고장출력 형태

Table 4: Faulty output pattern in case 2

부분 경우	고장출력	
	1	2
A	01 또는 10 이진벡터	01 또는 10 이진벡터
B	01 또는 10 이진벡터	φφ
C	01 또는 10 이진벡터	--
D	φφ	--
E	φφ	φφ
F	--	--

(3)경 우 3 과 4

고장 경로에 스위칭 소자의 집합을 계산할 수 있고 이러한 집합에 교집합은 고장 스위칭 소자를 검출할 수 있

나. 경우 3과 4의 두 경우에서 고장을 검출하고 위치시키기 위해서 단지 4개의 테스트가 필요하다. 경우 3과 4의 고장 위치와 종류는 많아야 8개의 테스트에 의해서 결정될 수 있다.

다. 스테이지별 테스트

그림 1에서 보여진 것과 같은 4개의 상태를 가지는 Generalized cube네트워크에서 broadcast상태에서 고장 진단은 가와 나에의 결과를 이용하는 스테이지별 테스트를 통해서이다.

Generalized cube네트워크에서 각 스테이지의 스위칭 소자 $m(0 \leq m \leq N/2)$ 에 broadcast출력값은 출력단자 m 과 $Cube_i(m) (0 \leq i \leq n-1)$ 을 통해서 알 수 있다.

증명: Generalized cube네트워크가 N 개의 입력과 N 개의 출력을 가진다고 가정할 때 $n = \log_2 N$ 개에 스테이지를 가진다. 각 스테이지는 $m(0 \leq m \leq N/2-1)$ 개 스위칭 소자를 가졌고, $Cube_i(m) (0 \leq i \leq n-1)$ 로 연결되어 있다.

여기서 $Cube_i(P_{n-1}, \dots, P_1, P_0) = P_{n-1} \dots P_{i+1} \overline{P_i} \dots P_1 P_0$ 이고 스테이지 $i (0 \leq i \leq n-1)$ 의 출력은 출력단자의 m 과 $Cube_i(m)$ 을 조사함으로써 알 수 있다. 따라서 스테이지 i 를 broadcast상태로 놓고 다른 스테이지는 direct상태로 놓을 때 스테이지 i 의 스위칭 소자에 상태는 출력단 m 과 $Cube_i(m)$ 을 조사함으로써 알 수 있다.

만약 $N=16$ 개 입력단을 가지는 generalized cube네트워크에서 스테이지 2의 broadcast출력값은 출력단 $m(0 \leq m \leq 7)$ 과 $Cube_2(m)$ 을 조사함으로써 알 수 있다. 그림 5에 broadcast 출력값의 예를 보인다.

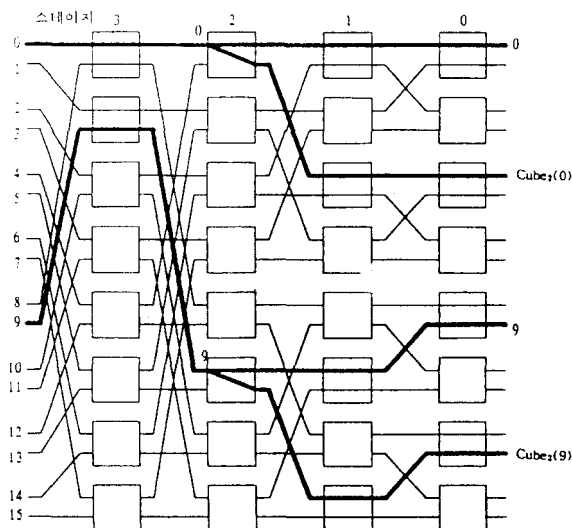


그림 5 Generalized cube 네트워크의 Upper-broadcast 출력
Fig. 5 Broadcast output in Generalized cube network

Generalized cube네트워크에서 스위칭 소자에 upper(혹은 lower) broadcast상태의 고장위치는 $2 \times \log_2 N$ 테스트에 의해서 결정될 수 있다.

증명: 가와 나에서의 링크코착고장과 스위칭 소자의 direct와 cross상태 테스트후 generalized cube네트워크에 각 스테이지의 스위칭소자 $m(0 \leq m \leq N/2)$ 의 broadcast출력값은 출력단자 m 과 $Cube_i(m) (0 \leq i \leq n-1)$ 을 통해서 알 수 있으므로 테스트를 위해서 산출된 테스트 벡터를 입력단에 적용하여 고장위치를 결정할 수 있다. 총 테스트의 수는 $2 \times \log_2 N$ 이다.

Generalized cube네트워크에서 broadcast 상태에서 고장 진단은 가와 나에의 결과를 이용하는 스테이지별 테스트

표 5. Lower broadcast상태에서 테스트 입력과 출력값
Table 5. Test inputs and outputs in lower-broadcast state

고 장	테스트		출력값				
	X_1	X_2	정상		고장		
			$\langle X_1 \rangle$	$\langle X_2 \rangle$	$\langle X_1 \rangle$	$\langle X_2 \rangle$	
스 위 칭 소 자 고 장	S_3-S_0	0	1	1	1	-	-
		1	0	0	0	-	-
	S_3-S_1	0	1	1	1	1	-
		1	0	0	0	0	-
	S_3-S_2	0	1	1	1	-	1
		1	0	0	0	-	0
	S_3-S_4	0	1	1	1	-	0
		1	0	0	0	-	1
	S_3-S_5	0	1	1	1	1	0
		1	0	0	0	0	1
	S_3-S_6	0	1	1	1	-	φ
		1	0	0	0	-	φ
	S_3-S_7	0	1	1	1	1	φ
		1	0	0	0	0	φ
	S_3-S_8	0	1	1	1	0	-
		1	0	0	0	1	-
S_3-S_9	0	1	1	1	φ	-	
	1	0	0	0	φ	-	
S_3-S_{10}	0	1	1	1	0	1	
	1	0	0	0	1	0	
S_3-S_{11}	0	1	1	1	φ	1	
	1	0	0	0	φ	0	
S_3-S_{12}	0	1	1	1	0	0	
	1	0	0	0	1	1	
S_3-S_{13}	0	1	1	1	φ	0	
	1	0	0	0	φ	1	
S_3-S_{14}	0	1	1	1	0	φ	
	1	0	0	0	1	φ	
S_3-S_{15}	0	1	1	1	φ	φ	
	1	0	0	0	φ	φ	

표 6. Upper broadcast 상태에서 테스트 입력과 출력값
Table 6. Test inputs and outputs in upper broadcast state

고 장	테스트		출력값			
	X_1	X_2	정상		고장	
			$\langle X_1 \rangle$	$\langle X_2 \rangle$	$\langle X_1 \rangle$	$\langle X_2 \rangle$
스 위 칭 소 자 고 장	$S_{12}-S_0$	0 1	1 0	0 1	0 1	- -
	$S_{12}-S_1$	0 1	1 0	0 1	0 1	1 0
	$S_{12}-S_2$	0 1	1 0	0 1	0 1	- 0
	$S_{12}-S_3$	0 1	0 1	0 1	0 1	1 0
	$S_{12}-S_4$	0 1	1 0	0 1	0 1	- 1
	$S_{12}-S_5$	0 1	1 0	0 1	0 1	1 0
	$S_{12}-S_6$	0 1	1 0	0 1	0 1	- φ
	$S_{12}-S_7$	0 1	1 0	0 1	0 1	φ φ
	$S_{12}-S_8$	0 1	1 0	0 1	0 1	- -
	$S_{12}-S_9$	0 1	1 0	0 1	0 1	φ φ
	$S_{12}-S_{10}$	0 1	1 0	0 1	0 1	1 0
	$S_{12}-S_{11}$	0 1	1 0	0 1	0 1	φ φ
	$S_{12}-S_{13}$	0 1	1 0	0 1	0 1	φ 1
	$S_{12}-S_{14}$	0 1	1 0	0 1	0 1	0 φ
	$S_{12}-S_{15}$	0 1	1 0	0 1	0 1	φ φ

를 통해서이다. 테스트 알고리즘은 다음과 같다. 먼저 generalized cube 네트워크가 N개의 입력과 N개의 출력을 가진다고 가정할 때 $n(=\log_2 N)$ 개의 스테이지를 가지게 된다.

단계 1: 스테이지 $i(=n-1)$ 를 정당한 upper(혹은 lower) broadcast 상태로 놓고 다른 스테이지는 direct 상태로 놓는다.

단계 2: 네트워크의 입력단에 알고리즘에 의해서 산출된 테스트 벡터 $i=01$ $i=10$ 을 적용한다.

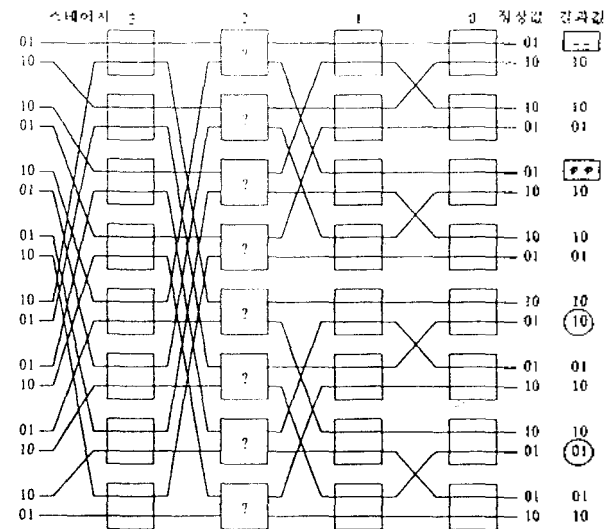
단계 3: 스테이지 i 를 스테이지 $i-1$ 로 이동한다.

단계 4: 스테이지 i 가 0가 아니면 다시 단계 1로 간다.

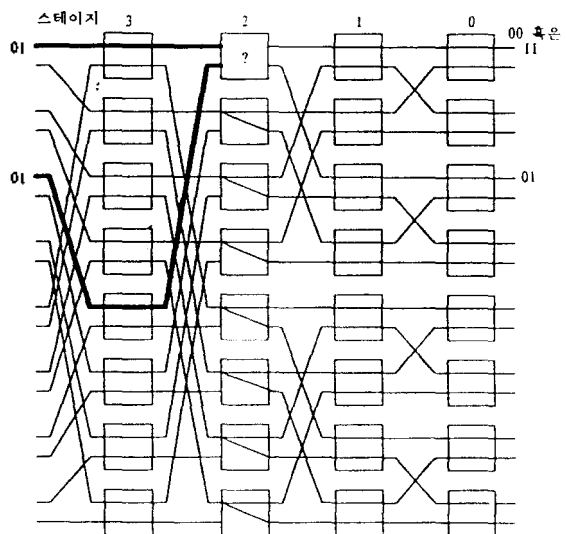
스테이지별 테스트에 의해서 검출된 고장은 각 스테이지에 하나 이상의 고장을 검출할 수 있다. 스테이지별 테

스트에서 ' $\varphi\varphi$ '와 ' $--$ '를 구별하기 위한 방법은 고장이 나타난 스위칭 소자에 입력단에 같은 테스트 벡터(01 혹은 10)를 적용한다. 따라서 출력단자의 출력값이 00 또는 11인 경우 ' $\varphi\varphi$ '와 ' $--$ '를 구별하기 위한 추가적인 2개의 테스트를 필요로 한다.

각 스테이지마다 고장을 포함하는 경우에도 ' $\varphi\varphi$ ' 혹은 ' $--$ '를 구별하기 위한 추가적인 테스트를 필요로 하게 된다. 따라서 각 스테이지마다 고장이 있는 최악의 경우 $4 \times \log_2 N$ 개의 테스트를 필요로 한다. 그러므로 스테이지별 테스트를 이용한 broadcast의 테스트의 총수는 upper와



(a) 스테이지 2의 upper-broadcast 테스트(Upper-broadcast test in stage 2)



(b) ' $\varphi\varphi$ '와 ' $--$ '를 구별하기 위한 테스트(Test for differentiating ' $\varphi\varphi$ ' and ' $--$ ')

그림 6 Generalized cube 네트워크의 broadcast 테스트
Fig. 6 Broadcast test in Generalized cube network

lower broadcast의 모든 경우를 포함 최대 $8 \times \log_2 N$ 이다. broadcast상태에서 테스트 입력과 출력값을 표 5와 6에 보였다.

만약 $N=16$ 개의 입력단을 가지는 generalized cube네트워크에서 스테이지 2의 broadcast 테스트를 그림 6에 보인다. 출력단의 결과를 이용하여 스테이지 2의 스위칭 소자 0과 5가 고장이다. 스테이지 2의 스위칭 소자 0는 " φ "와 "--"을 구별하기 위한 추가적인 테스트후 표 5를 이용하여 고장 종류를 결정할 수 있다.

V. 결 론

2개의 입력과 출력단을 가지는 스위칭 소자로 구성된 generalized cube네트워크에서의 고장진단을 위한 효율적인 알고리즘을 제시하였다. 본 논문에서는 Wu와 Feng이 제시한 2×2 스위칭 소자의 가능한 16개의 모든 상태를 고려하여 이를 다단 상호연결 네트워크중에서 generalized cube네트워크에 적용하여 라인 고착 고장과 스위칭 소자의 direct와 cross상태에서의 고장 진단 방법을 보였고, 여기에 broadcast상태에 고장진단을 포함하여 스위칭 소자의 4가지 상태에서 고장위치 및 종류를 진단할 수 있는 효율적인 방법을 제시하였다. 라인 고착 고장 및 스위칭 소자의 direct 및 cross상태에서 단일 고장을 검출하기 위한 테스트의 수는 네트워크에 크기에 관계없이 4이고, 고장위치 및 종류를 결정하기 위한 테스트의 수는 12이다.

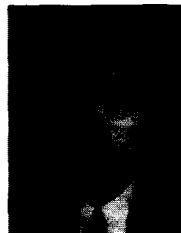
스위칭 소자의 broadcast상태에서의 테스트는 스테이지별 테스트를 이용하였는데, upper와 lower broadcast의 테스트 총수는 모든 경우를 포함해서 최소 $4 \times \log_2 N$ 그리고 최대 $8 \times \log_2 N$ 이다. 향후 4×4 스위칭 소자로 구성된 다단상호연결 네트워크에서의 고장진단에 관한 연구가 필요하다.

참 고 문 헌

1. K. Hwang, Advanced Computer Architecture: Parallelism Scalability, Programmability, McGraw-Hill, 1993.
2. H. J. Siegel, Interconnection Networks for Large-Scale Parallel Processing, McGraw-Hill, 1992.
3. F. Lombardi, C. Feng and W. K. Huang, "Detection and Location of Multiple Faults in Baseline Interconnection Networks," IEEE Trans. Comput., vol. 41, pp. 1340-1344, Oct. 1992.
4. G. J. Lipovski and M. Malek, Parallel Computing: Theory and Comparisons, New York: Wiley-Interscience, 1987.
5. C. Wu and T. Feng, "On a class of multistage interconnection networks," IEEE Trans. Comput., vol. C-29, pp. 694-702, Aug. 1980.
6. F. Lombardi and W. K. Huang, "On the constant diagnosability of baseline Interconnection Networks," IEEE Trans. Comput., vol. C-39, no. 12, pp. 1485-1488, 1990.

7. C. Wu and T. Feng, "Fault diagnosis for a class of multistage interconnection networks," IEEE Trans. Comput., ol. C-30, no. 10, pp. 743-758, Oct. 1981.
8. V. Cherkassky, E. Oppor and M. Malek, "Reliability and fault diagnosis analysis of fault tolerant multistage interconnection networks," in Proc. FTCS, pp. 243-253, 1984.
9. E. Oppor and M. Malek, "Real-time diagnosis of banyan networks," in Proc IEEE Real-Time Syst. Symp., pp. 27-36, 1982.
10. Dharma P. Agrawal, "Testing and fault tolerance of multistage interconnection networks," IEEE Comput., pp. 41-53, Apr. 1982.
11. K. M. Falavarjani and D. K. Pradhan, "Fault-diagnosis of parallel processor interconnection networks," in Proc. FTCS, pp. 209-212, 1981.
12. V. P. Nelson, "Fault-tolerant computing: fundamental concepts," IEEE Comput., vol. 23, no. 7, pp. 19-25, Jul. 1990.
13. A. D. Singh and H. Y. Youn, "A modular fault-tolerant binary tree architecture with short links," IEEE Trans. Comput., vol. 40, no. 7, pp. 882-890, Jul. 1991.

▲배 성 환(Sung Hwan Bae) 1970年 10月 12日生
 1993年 2月: 전북대학교 공과대학 전자공학과 졸업(공학사)
 1995年 2月: 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)
 1995년~현재: 전북대학교 대학원 전자공학 박사과정
 ※주관심분야: 병렬처리, 메모리 테스트



▲김 대 익(Dae Ik Kim) 1969年 1月 23日生
 1991年 2月: 전북대학교 공과대학 전자공학과 졸업(공학사)
 1993年 2月: 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)
 1996년 8월: 전북대학교 대학원 전자공학 박사과정 졸업(공학박사)
 1996년~현재: 전북대학교 전기전자회로 합성연구소 연구원
 ※주관심분야: VLSI설계, 메모리 테스트



▲이 상 태(Sang Tae Lee) 1954年 12月 6日生



1977年 2月: 아주대학교 전자공학과 졸업(공학사)

1993年 2月: 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1993年~현재: 전북대학교 대학원 전자공학 박사과정

1981年~1985年: 국제상사 컴퓨터 사업본부 근무

1985年~현재: 표준과학연구원 전산센터근무

※주관심분야: 병렬처리, 광대역통신망, 지능망

▲전 병 실(Byoung Sil Chon) 1945年 2月 14日生



1967年 2月: 전북대학교 공과대학 전기공학과 졸업(공학사)

1969年 2月: 전북대학교 대학원 전자공학 석사과정 졸업(공학석사)

1974年: 전북대학교 대학원 전자공학(공학박사)

1979年: 미국 Univ. of Nortre Dame 전기공학과 객원교수

1986年: 전북대학교 전자계산소장

1994年: 전북대학교 도서관장

1971年~현재 전북대학교 공과대학 전자공학과 교수

※주관심분야: 병렬컴퓨터, 지능망, ATM, VLSI 설계