

대칭적 의사결합처리요소 생성 기법에 의한 결함허용 VLSI 어레이의 신뢰도 향상과 실시간 재구성

(Reliability Improvement and Real-Time Reconfiguration of Fault Tolerant VLSI Arrays Using Symmetrical Pseudo Faulty Processing Elements Generation Technique)

申東錫*, 禹鍾鎬**

(Dong Suk Shin and Chong Ho Woo)

요 약

본 논문에서는 단일트랙 스위치로 구성된 결함허용 VLSI 어레이에서 제한된 하드웨어 자원으로 어레이의 신뢰도를 향상시키기 위하여 대칭적 의사결합처리요소의 생성 기법을 제안한다. 또한 결함허용 VLSI 어레이에 대한 신뢰도를 분석하고 실시간 재구성을 위한 제어논리를 설계하였다. 제안한 기법을 이차원 어레이의 재구성에 적용한 결과 제안한 방법은 동일한 조건의 기존방법보다 높은 신뢰도를 보였으며, 계산식에 의해 분석한 신뢰도의 결과값과 시뮬레이션한 결과값이 거의 유사함을 확인하였다. 그리고 재구성을 위한 제어가 각 처리요소에서 분산되어 수행되므로 재구성시간은 어레이 크기에 의존하지 않으며, 재구성된 어레이는 높은 상호연결의 국부성을 유지하므로 VLSI 어레이의 특징이 그대로 유지된다.

Abstract

In this paper, we propose a symmetrical pseudo faulty processing elements generation technique to improve the overall reliability of arrays with fixed hardware resources on the fault tolerant VLSI arrays based on single-track switches. We have analyzed the reliability of fault tolerant VLSI arrays and designed control logic for real-time reconfiguration. Applying this technique to reconfiguration of VLSI 2-D arrays, we have found that the proposed scheme achieves a higher reliability than the previous methods of similar condition. And we have found that the results of reliability analyzed by mathematic computation are very close to simulated ones. Furthermore, the time overhead for reconfiguration is independent of the array size because the control for reconfiguration is distributively executed by each processing elements. And the proposed scheme has an advantage which maintained properties of VLSI arrays by keeping the locality of interconnections as high as possible even after the reconfiguration.

I. 서 론

VLSI기술의 발전으로 인하여 대규모 프로세서 어레

이가 공학 및 과학분야의 실시간 응용등에 널리 이용되고 있으나 VLSI 어레이 구조가 복잡해짐에 따라 동작중의 어레이에서 영구 혹은 일시적인 결함이 발생될 가능성이 높아지므로, 어레이내의 다수의 처리요소에 결함이 존재하더라도 정확한 동작을 이행하는 결함허용(fault tolerance)이 요구된다^[1].

결함허용은 결함 테스트, 재구성 및 복구의 세단계로 구성되며 재구성 능력은 제공되는 트랙의 수에 의존한다^[2]. 재구성을 위해 추가되는 하드웨어가 많을수

* 正會員, 東明專門大學 電子計算科

(Dept. of Computer Science, Dong Myung Junior College)

** 正會員, 釜山水產大學校 컴퓨터工學科

(Dept. of Computer Eng., National Fisheries Univ. of Pusan)

接受日字: 1996年2月8日, 수정완료일: 1996年4月23日

록 높은 신뢰도를 얻을 수 있으나 공간과 비용의 제한으로 한계가 있다¹⁴⁾. 또한 멀티트랙과 같은 버스형태의 상호연결구조는 VLSI 구현시 이를 위해 많은 칩면적이 소요되며, 어레이에서 수평 혹은 수직방향으로 결합이 연속적으로 발생하면 처리요소간의 연결 거리가 길어져서 클럭속도가 저하되어 시스템성능을 저하시키는 단점이 있다¹³⁾. 따라서 높은 국부성(locality)을 유지하면서 가능한 적은 수의 여분처리요소, 단일트랙스위치 등의 제한된 여분하드웨어자원을 이용하여 높은 신뢰도를 얻을 수 있도록 처리요소사이의 연결을 재구성하여야 한다. 그러나 지금까지 제안된 단일트랙스위치를 이용한 재구성방법들은 단일트랙의 한계로 인하여 재구성 능력이 제한되는 단점이 있어 각 처리요소에서 결합이 발생할 확률이 커지면 낮은 신뢰도가 문제가 된다^{14) 10)}. 한편 어레이의 실시간 결합허용 설계에서 중요하게 고려되어야 할 사항은 결합테스터, 재구성, 복귀까지의 재구성 시간을 최소화 하는 것으로 각 처리요소에서 이웃하는 처리요소의 상태에 관한 지역 정보(local information)만을 이용하면 이를 최소화시킬 수 있다.

본 논문에서는 단일트랙으로 인한 자원부족(resource lack) 문제와 낮은 신뢰도의 한계를 극복할 수 있는 대칭적 의사결합처리요소 생성(Symmetrical Pseudo faulty PE Generation:SPPEG)기법을 이용하여 제한된 하드웨어자원으로 높은 신뢰도를 얻을 수 있는 효율적인 방법을 제안하고, 이의 실시간 재구성을 위한 제어논리를 설계한다. 재구성 방법은 어레이내의 모든 처리요소에서의 자기진단정보(Self Test Function:STF)를 이용하여 처리요소의 결합상태에 따른 여분할당을 행하고, 여기서 구해지는 사상 매트릭스(mapping matrix)에 따라 인덱스사상을 행한 후, 각 처리요소에서의 링크 재결합을 통하여 문제의 크기에 따른 논리어레이를 구성한다. 또한 상호연결의 공간적 국부성과 시간적 국부성을 만족시키기 위해 상호연결 길이가 2로 제한되므로 재구성된 어레이는 높은 국부성이 유지된다.

제안한 방법의 성능을 평가하기 위하여 수학적 방법과 컴퓨터 시뮬레이션 기법을 사용하였다. 성능분석 결과 기존방법에 비해 높은 신뢰도를 보였으며, 특히 각 처리요소의 신뢰도가 0.99인 경우에는 모든 어레이 크기에 대해서 100% 재구성되며, 0.95인 경우에는 어레이 크기가 증가함에 따라 신뢰도는 지수적으로 감소

하였다.

II. 결합허용 VLSI 어레이 모델

VLSI 어레이의 결합허용은 여분하드웨어 자원을 이용하여 잠재적으로 결합을 갖는 물리어레이(physical array)를 무결합의 논리어레이(logical array)로 재구성하는 것이다. 여기서 고려하는 결합허용 VLSI 어레이 모델은 그림 1의 (a)와 같이 $M \times N$ 의 처리요소에 1행 1열의 여분처리요소가 부가된 단일트랙스위치의 상호연결링크로 구성되며, 스위치의 상태는 그림 1의 (b)와 같다.

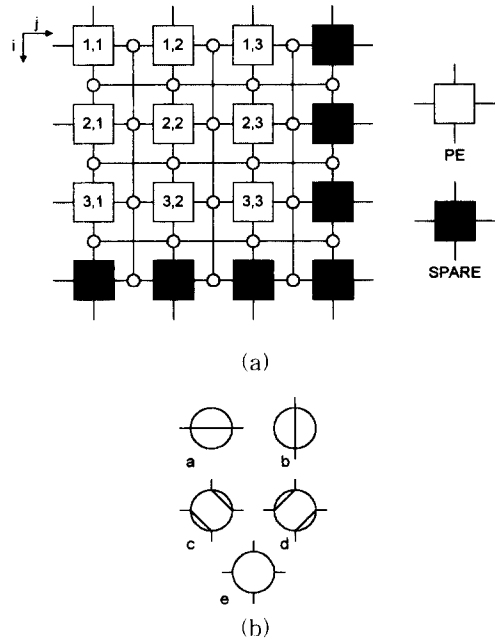


그림 1. (a) 단일트랙 스위치를 이용한 어레이 모델, (b) 스위치의 상태

Fig. 1. (a) The array model based on single-track switches, (b) The status of switches.

본 논문에서는 어레이의 신뢰도를 향상시키기 위하여, [-1,0]과 [0,-1]의 양방향에서 여분할당을 위한 결합처리요소의 상태결정이 이루어지며, 이 두 방향에서 재구성이 동시에 수행된다.

재구성 알고리즘의 전개를 위하여 어레이를 물리어레이와 논리어레이로 구별한다. 물리어레이의 처리요소 집합은 P , ($1 \leq P \leq (M+R) \times (N+C)$)로 정의하고, 논리어레이의 처리요소 집합은 L , ($1 \leq L \leq M \times N$)로 정의한다. 여기서 R 과 C 는 각각 여분행과 열의 수이다.

어레이에서 각 처리요소의 상태는 무결함상태, 결함상태 및 의사결합상태로 구별한다. 의사결합상태는 실제 결함이 있는 상태가 아니라 성공적인 재구성을 위하여 의사적으로 결함상태로 두는 것으로, 이들은 의사수평결합(Pseudo H : PH), 의사수직결합(Pseudo V : PV), 의사수평수직결합(Pseudo HV : PHV) 및 의사수직수평결합(Pseudo VH : PVH) 상태로 구분한다. 결함상태는 수평결함(H0), 수직결함(V0) 및 자체결함(F) 상태로 구분한다. 어레이에서 처리요소의 위치는 물리적 위치와 논리적 위치로 나타내며 각각 (i, j) 와 (i', j') 로 표시한다. 논리어레이는 재구성된 무결함 어레이이며, 결함이 없으면 논리인덱스와 물리인덱스는 같다. 또한 결함처리요소와 이용되지 않은 여분처리요소의 인덱스는 편의상 $(0,0)$ 으로 둔다. 그리고 재구성 과정에서 각 처리요소들간의 높은 국부성이 유지되도록 논리인덱스 (i', j') 가 사상될 수 있는 물리인덱스는 $(i, j), (i, j+1), (i+1, j), (i+1, j+1)$ 로 제한한다.

어레이에서 각 처리요소들은 단일트랙으로 연결되며, $PE(i, j)$ 와 $PE(i, j+1)$ 를 연결하는 수평링크와 $PE(i, j)$ 와 $PE(i+1, j)$ 를 연결하는 수직링크가 있다¹⁶⁾. 처리요소가 무결함상태이면 모든 링크가 인접 처리요소에 접속되며, 처리요소가 결함상태라면 동서 혹은 남북의 링크만 사용한다.

어레이에서 각 처리요소에 대한 결함은 랜덤하게 분포되며, 결함은 처리요소에서만 발생되고, 각 처리요소에는 결함을 판단하기 위한 자체결함체크기능이 있다고 가정한다¹²⁻¹¹⁾. 그리고 결함이 발생된 처리요소는 단지 연결요소(connecting element)로만 사용한다.

III. 재구성 알고리즘

1. 기본 개념

재구성은 여분할당, 인덱스사상 및 링크 재결함으로 이루어지며, 여분할당은 결함처리요소를 여분처리요소로 대체하는 과정으로 정의된 규칙에 의해 각 처리요소의 결함상태가 결정되며, 이에따라 사상경로(mapping path)가 형성된다. 사상경로의 형태는 세 종류, 즉 수직사상경로, 수평사상경로 및 혼성사상경로가 있으며 이는 결함처리요소의 여분처리요소로의 할당에서 그 논리인덱스가 수정되어야할 물리어레이상의 경로를 의미한다. 그림 2에서 처리요소 C1이 수직결함 상태이면 이는 여분처리요소 S1에 의해 대체되며,

C1에서 S1에 이르는 수직사상경로가 형성된다. 만일 처리요소 C1이 수평결함 상태이면 이는 여분처리요소 S2에 의해 대체되며, C1에서 S2에 이르는 수평사상경로가 형성된다. 만일 C1이 수평결함 상태이고 C3가 수직결함 상태이면 C1을 위해서는 S2가 C3를 위해서는 S3가 결함을 대체하기 위해서 사용되며, C3에서 사상경로의 교차가 일어나며, 이 사상경로를 혼성사상경로(hybrid mapping path)라 한다.

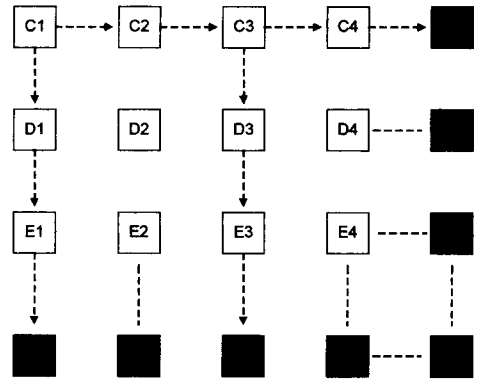


그림 2. 사상경로의 형태
Fig. 2. The types of mapping paths.

그림 1의 어레이 모델에서 $[-1, 0]$ 방향의 재구성시 결함처리요소의 상태를 결정하기 위한 과정은 다음과 같이 어레이의 각 열의 밑에서 위로 동시에 수행된다(그림 3의 a참조).

- (1) $PE(i, j)$ 가 결함처리요소이고, $1 \leq i \leq M+R, 1 \leq j \leq N$ 인 경우
 - ① $PE(i, j)$ 가 그 열에서 첫 번째 결함이면 수직결함(V0) 상태가 된다
 - ② $PE(i, j)$ 가 그 열에서 첫 번째 결함이 아니면 수평결함(H0) 상태가 된다.
- (2) $PE(i, j)$ 가 결함처리요소이고, $1 \leq i \leq M+R, j = N + C$ 인 경우
 - ① $PE(i, j)$ 의 좌측($PE[i, k]$)에 수평결함 상태의 결함처리요소가 존재하면 수직결함 상태가 된다.
 - ② $PE(i, j)$ 의 좌측($PE[i, k]$)에 수평결함 상태의 결함처리요소가 존재하지 않으면 $PE[i, j]$ 는 자체결함이 된다.
 - ③ 여기서 k 는 $1 \leq k \leq j-1$ 이다.

또한 $[0, -1]$ 방향은 그림 3과 같이 $[-1, 0]$ 방향의

결합처리요소 상태결정 과정에서 $[-1,0]$ 의 $[1,1]$ 을 축으로 모든 동작들이 서로 대칭된다. 따라서 $[0,-1]$ 방향의 재구성에서 결합처리요소의 상태를 결정하기 위한 과정은 어레이의 각 행의 우측에서 좌측으로 동시에 수행되며(그림 3의 b참조), 위의 과정 (1), (2)에서 $H0$ 는 $V0$ 로, $V0$ 는 $H0$ 가 되며 그 인덱스만 바꾸어 주면 된다. 그림에서 화살표는 사상경로를 나타낸다.

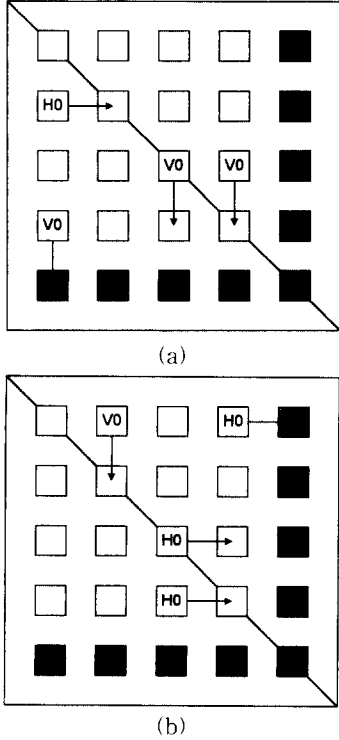


그림 3. 재구성에서 각 방향의 대칭성 (a) $[-1,0]$ 방향과 (b) $[0,-1]$ 방향

Fig. 3. The symmetry of reconfiguration in (a) $[-1,0]$ and (b) $[0,-1]$ direction.

결합처리요소의 상태에 따른 여분할당으로 사상경로가 형성되면 이에따라 사상 매트릭스가 구성된다. 사상 매트릭스는 수평 매트릭스(H_{pi})와 수직 매트릭스(V_{pi})에 의해 표현되며, 이의 각 원소는 물리어레이에서 각 처리요소내의 H와 V플래그 값이다. 사상 매트릭스는 인덱스사상에서 이용되며 논리어레이 구성을 위한 인덱스 정보를 포함하고 있다. $PE(i,j)$ 가 수평결합이면 $PE(i,k)$, $j < k \leq N+C$,의 H는 1로 되며, $PE(i,j)$ 가 수직결합이면 $PE(k,j)$, $i < k \leq M+R$,의 V는 1이 된다.

인덱스사상은 물리어레이로부터 요구되는 논리어레이를 구성하기 위한 논리인덱스의 물리인덱스로의 사

상(mapping)으로 인덱스사상 함수 f 로 정의한다. 성공적인 재구성에서는 모든 $PE(i',j') \in L$ 에 대한 물리어레이의 무결합처리요소 $PE(i,j) \in P$, 즉 $f(i',j') = (i,j)$ 가 존재한다는 것을 의미한다. 인덱스가 이동되는 사상경로의 형태는 수평사상과 수직사상으로 제한되며, 사상 매트릭스 H_{pi} 및 V_{pi} 의 값에 따라 인덱스가 이동된다. 즉, j 열에서의 수직사상은 $f(1,j') \rightarrow f(2,j') \rightarrow f(3,j') \dots \rightarrow f(M,j')$ 이다. $[-1,0]$ 방향의 재구성에서는 수평사상이 수행된 후 수직사상이 수행되며, $[0,-1]$ 방향의 재구성에서는 그 반대로 수직사상이 수행된 후 수평사상이 수행된다. 수평사상은 다음과 같고 수직사상도 이와같은 방법으로 이행된다¹⁾¹⁰⁾.

- (1) $PE(i,j)$ 의 $H=1$ 이면, $f(i',j-1') = (i,j)$
- (2) $PE(i,j)$ 의 $H=0$ 이고 $PE(i,j+1)$ 의 $H=0$ 이면, $f(i',j') = (i,j)$
- (3) $PE(i,j)$ 의 $H=0$ 이고 $PE(i,j+1)$ 의 $H=1$ 이면, $f(0,0) = (i,j)$

링크 재결합은 물리어레이상의 모든 수평링크와 수직링크상의 스위치에 의해 이루어지며, 스위치요소에서의 경로배정은 동서남북 방향의 인접한 처리요소의 상태에 따라 결정된다.

한편 S.Y. Kung¹⁴⁾이 제안한 단일트랙을 이용한 재구성 알고리즘은 보상패스의 교차(intersect)를 허용하지 않으므로 성공적인 재구성을 위해서는 두 수직경로(수평경로)가 중복(overlap)되지 않는 순서보존(order-preserving)이 요구된다. 즉 모든 수직경로(수평경로)가 원래의 순서를 유지해야 하므로 물리어레이의 어떤 행에서 논리인덱스 j' 의 왼쪽에 있는 처리요소의 논리인덱스는 인덱스사상 후에도 j' 보다 작으며, 오른쪽에 있는 처리요소의 논리인덱스는 인덱스사상 후에도 j' 보다 크야 한다. 따라서 혼성사상경로가 존재하는 경우에는 순서보존의 위배로 인한 상호연결자원의 부족으로 재구성이 불가능하다. 그림 4에서 물리어레이 $PE(1,4)$ 의 사상경로는 교차되므로 논리어레이 $PE(1,3)$ 과 $PE(2,3)$ 을 연결하는(a 표시) 상호연결 링크와 논리어레이 $PE(1,4)$ 와 $PE(2,4)$ 를 연결하는(b 표시) 상호연결 링크의 부족으로 인하여 재구성 불가능하다.

단일트랙을 이용한 재구성에서 이러한 제한은 결과적으로 낮은 신뢰도를 초래하므로 수직 혹은 수평경로의 교차를 허용하는 융통성있는 여분할당이 필요하다.

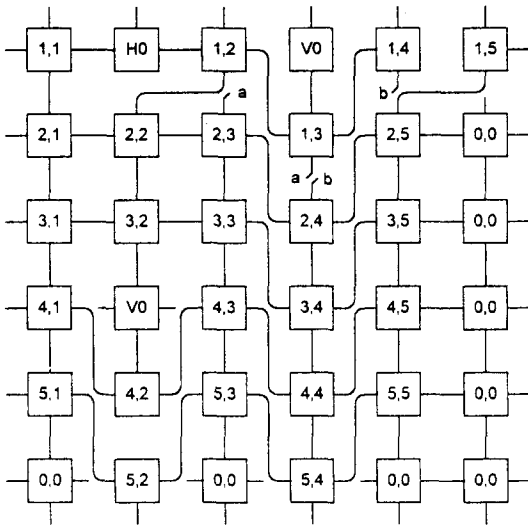


그림 4. 혼성사상경로로 인한 자원부족
Fig. 4. The resource lack by hybrid mapping path.

2. 의사결합처리요소의 생성

단일트랙을 이용한 VLSI 어레이의 재구성은 인덱스 사상 후에도 모든 수직 혹은 수평경로가 원래의 순서를 유지하는 순서보존이 요구된다. 그러나 혼성사상경로가 존재하는 경우에는 재구성이 불가능하다. 본 논문에서는 의사결합처리요소의 개념을 이용하여 사상경로의 교차를 허용하는 융통성 있는 여분할당을 행하여 단일트랙으로 인한 자원부족 문제를 해결하고 높은 신뢰도를 얻을 수 있는 SPPEG 기법을 제안한다.

의사결합처리요소는 여분할당에서 혼성사상경로가 존재하는 경우에 순서보존을 만족시키기 위한 수단으로 사용되며, 네 종류가 있다. [-1,0] 방향의 재구성에서는 PH, PV 및 PHV가 생성되고, [0,-1] 방향의 재구성에서는 PH, PV 및 PVH가 생성된다. PH와 PHV는 자신의 상태를 수평결합 상태로 세트하며(즉, H0=1), PV와 PVH는 자신의 상태를 수직결합으로 세트한다(즉, V0=1). 위의 그림 4에서 물리어레이의 PE(2,3)과 PE(2,5)를 각각 수직결합 상태의 의사결합처리요소와 수평수직결합 상태의 의사결합처리요소로 두면 재구성 가능하다.

의사결합처리요소의 생성은 물리어레이상에서 수평결합과 수직결합이 발생하는 위치에 의존하므로 [-1,0] 방향의 재구성시 의사결합처리요소의 생성은 물리어레이의 PE(k,j)가 수직결합이고 PE(i,d)가 수평결합일 때 이의 생성이 요구된다. 여기서 $1 \leq k \leq i, d+1 \leq j \leq N$

$+C, k \leq i \leq M$ 및 $1 \leq d \leq j-1$ 혹은 $1 \leq k < i, d+1 \leq j \leq N, k \leq i \leq M$ 및 $1 \leq d \leq j-1$ 이다. [0,-1] 방향의 재구성시 의사결합처리요소의 생성은 물리어레이 PE(i,k)가 수평결합이고 PE(l,j)가 수직결합일 때 이의 생성이 요구된다. 여기서 $1 \leq k \leq j, 1+1 \leq i \leq M+R, k \leq j \leq N$ 및 $1 \leq i \leq i-1$ 혹은 $1 \leq k < j, 1+1 \leq i \leq M, k < j \leq N$ 및 $1 \leq i \leq i-1$ 이다.

3. 알고리즘 설계

제한된 알고리즘은 어레이내의 임의의 처리요소에서 결함이 발생하면 그 결함상태에 의해 구성되는 사상경로를 따라 인접한 처리요소로 결함상태를 전송하며, 그 경로상의 무결합처리요소는 결함처리요소로부터 결함상태에 관한 정보를 받아 인덱스사상에서 이용될 자신의 플래그(H 또는 V) 값을 변화시켜 링크 재결합을 통하여 재구성된다.

1) 재구성 알고리즘의 구성요소

재구성은 [-1,0] 방향과 [0,-1] 방향에서 한 클럭 시간동안 동시에 수행되며, 먼저 [-1,0] 방향으로 재구성이 이루어진다. 이 방향에서 재구성불가능할 경우 SF신호가 세트되어, [0,-1] 방향의 재구성이 이루어진다. 만일 후자의 방향에서도 재구성이 불가능 할 경우 FF(fatal failure)신호가 세트되어 재구성 불가를 나타낸다. 재구성 알고리즘의 전체적인 구조는 다음과 같다.

Reconfiguration algorithm

Input : 각 처리요소 PE [ij] 는 결함여부를 STF [ij] 에 가지고 있다 /* Fault PE : STF [ij] = 1, Fault free PE : STF [ij] = 0 */

Output : 각 처리요소는 인덱스사상의 결과를 H [ij] 및 V [ij] 에 진다. 즉, H [ij] 가 1이면 수평사상에 속한다는 것을 나타내고, V [ij] 가 1이면 수직사상임을 나타낸다.

I: 결합처리요소 상태 집합 (I1: 수직결합, I2: 수평결합, I3: 자체결합)

A: 의사결합처리요소 상태 집합 (A1: 의사수평결합처리요소, A2: 의사수직결합처리요소, A3: 의사수평수직결합처리요소, A4: 의사수직수평결합결합처리요소)

procedure RECONFIGURATION
begin

```

for all 1 ≤ i ≤ M+R, 1 ≤ j ≤ N+C do in parallel
    H0&V0_SETTING /* 결합처리요소의 상태 결정 및 수평, 수직사상 세트 */
    PS_SETTING /* 의사결합생성 유도신호 세트 */
    BB_SETTING /* 수평 또는 수직결합상태의 의사결합처리요소 생성 */
    BF_SETTING /* 수직 또는 수평결합상태의 의사결합처리요소 생성 */
    PB_SETTING /* 수평수직 또는 수직수평결합상태의 의사결합처리요소 생성 */
    
```

```
SF&FF_SETTING /* 재구성 방향 및 재구성 불가 판별 */
all for
end RECONFIGURATION.
```

그림 5는 알고리즘에서 요구되는 제어신호 명세를 기술하기 위한 예로서 물리레이 PE(3,2)와 PE(2,5) 및 PE(6,2)가 각각 결합인 경우의 [-1,0] 방향에서 재구성된 7×7 어레이이다. 각 결합처리요소의 상태를 결정하기 위한 과정에서 PE(6,2)와 PE(2,5)는 결합여부를 나타내는 STF가 1이고 그 옆에서 첫 번째 결합이므로 수직결합이 되며, 또한 동일 열에 존재하는 결합처리요소의 상태를 결정하기 위하여 PE(k,2)와 PE(k1,5)로, $1 \leq k \leq 5$, $1 \leq k1 < 2$, 자신들의 결합을 알리는 신호가 필요하다. 이를 CF라 한다. 따라서 PE(3,2)는 자신의 STF가 1이고 CF가 1이므로 수평결합 상태가 된다. 알고리즘에서 수평결합과 수직결합은 H0와 V0신호로 나타내며, H0=1이면 수평결합, V0=1이면 수직결합이다. 이 결합처리요소들을 대치하기 위한 여분할당을 위해 H0와 V0신호는 동일행과 열로 전파되며, 인덱스사상에서 이용되는 사상 매트릭스를 구성하기 위해 각 처리요소에 H와 V플래그가 필요하다. 그러므로 PE(6,2)는 PE(7,2)와 여분처리요소 PE(8,2)의 V플래그를 1로 셋트하며, PE(3,2)는 PE(i,k)의, $3 \leq k \leq 8$, H플래그를 1로 셋트한다. 한편 물리레이 PE(3,5)에서 사상경로가 교차되므로 2장에서 설명한 의사결합처리요소의 생성이 요구된다. 이러한 사상경로의 교차는 PE(i,j)가 수평결합이고 자신의 우측이나 우측상단에 수직결합 상태의 결합처리요소가 존재하면 발생한다. 이 수직결합 상태를 알리는 신호가 EB이며, 자신의 좌측과 좌측하단의 처리요소에 자신 이 수직결합임을 알린다. 즉 PE(i,j)가 수직결합이면 처리요소 PE(k,l), $i \leq k \leq M+R$ 및 $1 \leq l \leq j-1$,의 EB가 1이된다. 따라서 사상경로의 교차가 일어난 지점을 알려서 의사결합처리요소의 생성을 유도하는 신호가 PS, IPS 및 DB이다. 그림과 같이 수평결합처리요소 PE(i,j)의 우측상단에 수직결합처리요소 PE(k,l)이, $1 \leq k \leq i-1$, $j+1 \leq l \leq N$, 존재할 경우 PE(k,j)의 처리요소는 의사수평결합처리요소가 되어야 하므로, PE(i,j)가 수평결합 인 것을 PE(k,j)에 알려야 한다. 이 신호가 DB이다. 마찬가지로 PE(3,5)에서의 사상경로 교차로 인해 PE(4,4), PE(4,6)에도 의사결합처리요소의 생성이 요구된다. 이를위한 신호가 PS와 IPS이다. PE(i,j)의 윗행(PE(i-1,j))에 있는 처리요소의 H와 V플래그

가 1이거나, H플래그가 1이고, 수직결합인 경우는 사상경로의 교차를 의미하므로, 의사결합처리요소의 생성을 위해 PE(i,j)에 PS신호가 생성된다. 그리고, 이 PS신호에 의해서 IPS신호가 PE(i,k)에, $j+1 \leq k \leq N+C$, 전달된다. 따라서, PE(4,4)는 의사수직결합처리요소(PV)가 되며, PE(4,6)은 의사수평수직결합처리요소(PHV)가 된다. 알고리즘에서 의사수평결합처리요소는 BB, 의사수직결합처리요소는 BF, 의사수평수직결합처리요소는 PB신호로 나타낸다.

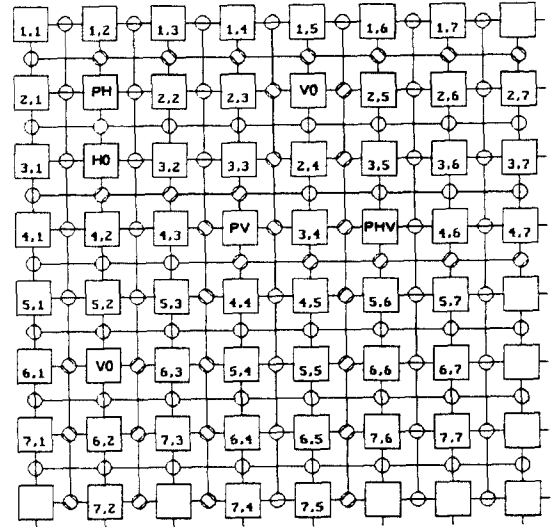


그림 5. 결합에 따른 재구성의 예
Fig. 5. The example of reconfiguration in fault pattern.

그림 5에서 기존의 결합외에 PE(4,3)과 PE(7,3)의 수평결합과 수직결합이 각각 추가되었을 경우, 알고리즘에 의해 PE(2,3)이 의사수평결합처리요소가 되어야 하나 이미 PE(2,2)가 의사수평결합처리요소이므로 PE(2,3)이 의사결합처리요소가 되는 것을 방지해야 재구성이 가능하다. 이를위해 이미 의사결합처리요소가 존재한다는 신호 VB가 필요하다. RB신호는 PE(i,j)의 H0가 1인 처리요소에서 PE(i,k)의, $1 \leq k < j$, 처리요소로 보내며, 의사수평결합처리요소의 생성시에 우측에 이미 수평결합이 존재할 경우 [-1,0] 방향의 재구성 불가를 알리기 위한 것으로 SF신호를 셋트한다. DD신호는 여분처리요소임을 알리는 신호이다. 이상의 [-1,0] 방향의 재구성에서 PE(i,j)가 의사결합처리요소가 되는 조건은 다음과 같다. 여기서 \wedge 는 논리 AND이다.

(1) PH 생성조건

- ① $DB_{ij} \wedge EB_{ij} \wedge H'_{ij} \wedge RB'_{ij}$ 혹은
- ② $CF_{ij} \wedge PS_{ij+1} \wedge H'_{ij} \wedge RB'_{ij}$

(2) PV 생성조건

- ① $PS_{ij+1} \wedge CF'_{ij} \wedge H'_{ij} \wedge V'_{ij}$ 혹은
- ② $IPS_{ij} \wedge RB_{ij} \wedge CF'_{ij}$

(3) PHV 생성조건

- ① $PS_{ij+1} \wedge V'_{ij} \wedge H'_{ij} \wedge RB'_{ij}$ 혹은
- ② $H0_{ij} \wedge EB'_{ij} \wedge IPS_{ij}$

2) 여분할당 및 사상 매트릭스 구성

여기서는 폴리어레이 PE(i,j)의 결합여부를 나타내는 STF와 CF에 의해 결합처리요소의 상태를 결정하고, 결합처리요소의 상태에 따른 여분할당과 인덱스사상에서 이용되는 사상 매트릭스의 구성을 위한 H와 V플래그의 값이 세트된다. 또한 이 과정의 수행중에 자신의 결합상태를 알리는 EB 및 RB신호등이 인접한 처리요소에 전달된다. 이를위한 처리과정이 H0&V0_SETTING이다.

표 1. 논문에 사용된 플래그와 기호
Table 1. The flags and symbols used in paper.

[-1,0]방향의 재구성, $1 \leq i \leq M-R$, $1 \leq j \leq N-C$			[0,-1]방향의 재구성, $1 < i \leq M-R$, $1 \leq j \leq N-C$		
기호	의미	신호 흐름	기호	의미	신호 흐름
HO	수평 결합	→	H0	수평 결합	→
VO	수직 결합	↓	VO	수직 결합	↓
CF	하단의 결합 출현 여부	↑	CF	우측의 결합 출현 여부	←
RB	우측의 수평결합 출현 여부	←	RB	하단의 수직결합 출현 여부	↑
EB	우측과 우측상단의 수직 결합 출현 여부	←	EB	하단과 좌측하단의 수평 결합 출현 여부	↑
DB	PH 생성 유도신호	↑	DB	PV 생성 유도신호	←
PB	PHV 생성 조건	→	PB	PVH 생성 조건	←
PS	PH, PV 생성 유도신호	→	PS	PV, PH 생성 유도신호	↓
IPS	PHV, PV 생성 유도신호	→	IPS	PVH, PH 생성 유도신호	↓
BB	PH 생성 조건	←	BB	PV 생성 조건	←
BF	PV 생성 조건	→	BF	PH 생성 조건	←
VB	BB, PB의 출현 여부	→	VB	BB, PB의 출현 여부	↓
SF	[-1,0]방향의 재구성 불가	all	IF	재구성 불가	↑
DD	우측 SPARE 여부(N·C)	↑	LD	하단 SPARE 여부(M·R)	←
flag	의미	신호 흐름	flag	의미	신호 흐름
H	수평사상을 위한 플래그	→	H	수평사상을 위한 플래그	→
V	수직사상을 위한 플래그	↓	V	수직사상을 위한 플래그	↓
STF	결합 여부	STF	결합 여부	STF	결합 여부

한편 [0,-1] 방향의 재구성에서 PE(i,j)가 의사결합처리요소가 되는 조건은 [1,0] 방향의 재구성에서 PE(i,j)가 의사결합처리요소가 되는 조건과 대칭되므로 PH는 PV, PV는 PH, 그리고 PHV는 PVH의 생성 조건이 된다. 그리고 각 조건에서 H'는 V', H0는 V0, V'는 H'가 되며 대칭조건에 따른 인덱스만 바꾸어 주면 된다. 표 1은 재구성에서 요구되는 제어신호와 플래그를 종합한 것이다. 각 기호는 위에서 기술한 재구성에서 사용되는 제어신호를 나타내며, 이 신호에 의해 각 처리요소의 H와 V플래그의 값이 결정된다. 신호의 흐름은 그림 1의 (i,j)좌표에서의 방향을 나타낸다.

procedure H0&V0_SETTING

```

if SF가 0일 경우 /* [-1,0]방향의 재구성 */
if (우측여분이 아닌 경우) /* j ≠ N·C */
if (PE [k,j]로 부터 결합(CF신호)이 없고 PE [i,j] 자신이 결합일 경우) /* k = i-1-M+R */
자신의 상태 = I1 /* V0 [i,j] = 1 */
PE [i,j] 아래의 수직사상을 위한 PE [k,j]의 V플래그를 세트 시킨다. /* k = i-1-M+R */
PE [i,j] 자신의 결합 전달을 위해 PE [k,j]로 CF신호를 보낸다. /* k = i-1 */
PE [k,l]에 PE [i,j]의 I1상태와 PE [k+1,j]의 V플래그 존재를 알리기 위한 신호 EB를 보낸다. /* k = i-M+R, l = 1-j-1 */
end if
if (PE [k,j]로 부터 결합(CF신호)이 있고 자신PE [i,j]가 결합일 경우) /* k = j+1-M+R */
자신의 상태 = I2 /* H0 [i,j] = 1 */
PE [i,j] 우측의 수평사상을 위한 PE [i,k]의 H플래그를 세트한다. /* k = j+1-N·C */
PE [i,j] 좌측의 같은 행(PE [i,l])에 I2가 존재한다는 신호 RB를 보낸다. /* l = 1-j-1 */
PE [k,j]에 PE [i,j]가 I2 상태임을 알리는 신호 DB를 보낸다. /* k = 1-i-1 */
end if
else /* j = N·C */
if (PE [i,j] 자신의 좌측에 I2가 존재(H플래그)할 경우) /* H [i,j] = 1 */
자신의 상태 = I1
PE [i,j] 아래의 수직사상을 위한 PE [k,j]의 V플래그를 세트 시킨다. /* k = i+1-M·R */
PE [k,l]에 PE [i,j]의 I1상태와 PE [k,j]의 V플래그 존재를 알리기 위한 신호 EB를 보낸다. /* k = i+1-M+R, l = 1-j-1 */
else
자신의 상태 = I3
end if
end if
else /* [0,1]방향의 재구성 */

```

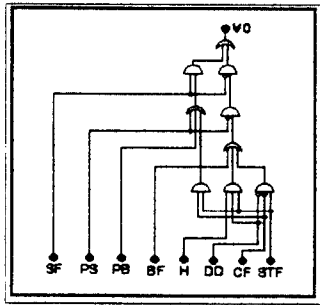
```

if (하단여분이 아닌 경우) /* i ≠ M+R */
  if (PE [i,k] 로 부터 결합(CF신호)이 없고 PE [ij] 자신이
    결합일 경우) /* k = j+1~N+C */
      자신의 상태 = I'2 /* H0 [ij] = 1 */
    PE [ij] 우측의 수평사상을 위한 PE [k,j] 의 H플래그를
      셋트 시킨다. /* k = j+1~N+C */
    PE [ij] 자신의 결합 전달을 위해 PE [i,k] 로 CF신호를
      보낸다. /* k = 1~j-1 */
    PE [i,k] 에 PE [ij] 의 I'2상태와 PE [i,k+1] 의 H플래그
      존재를 알리기 위한 신호 EB를 보낸다.
      /* k = j~N+C, l = 1~i-1 */
  end if
  if (PE [i,k] 로 부터 결합(CF신호)이 있고, 자신PE [ij] 가
    결합일 경우) /* k = j+1~N+C */
      자신의 상태 = I'1 /* V0 [ij] = 1 */
    PE [ij] 하단의 수직사상을 위한 PE [k,j] 의 V플래그를
      셋트한다. /* k = i+1~M+R */
    PE [ij] 상단의 같은 열(PE [i,j])에 I'1이 존재한다는
      신호 RB를 보낸다. /* l = 1~i-1 */
    PE [i,k] 에 PE [ij] 가 I'1 상태를 알리는 신호 DB를
      보낸다. /* k = 1~j-1 */
  end if
else /* i = M+R */
  if (PE [ij] 자신의 상태에 I'1이 존재(V플래그)할 경우)
    /* V [ij] = 1 */
    자신의 상태 = I'2
    PE [ij] 우측의 수평사상을 위한 PE [i,k] 의 H플래그를
      셋트 시킨다. /* k = j+1~N+C */
  
```

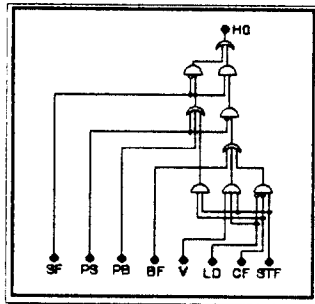
```

PE [l,k] 에 PE [ij] 의 I'2상태와 PE [i,k] 의 H플래그 존재를
  알리기 위한 신호 EB를 보낸다.
  /* k = j+1~N+C, l = 1~i-1 */
else
  자신의 상태 = I'3
end if
end if
end if
end if
end H0&V0_SETTING.
  
```

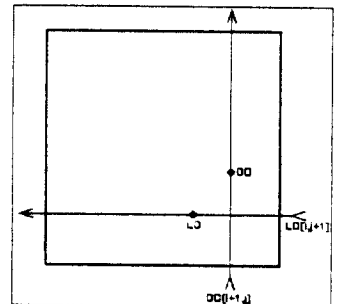
그림 6은 PE(i,j)에서 이 과정을 이행하는 제어논리이다. DD와 LD는 물리어레이에서 각각 여분처리요소를 구별하기 위한 신호로서 [-1,0] 방향의 재구성에서 $i=M+R, j=N+C$ 에 위치하는 처리요소의 DD 입력값은 1이고, [0,-1] 방향의 재구성에서는 이 처리요소의 LD 입력값이 1이 된다. 제어논리에서 입력될 기호 I[i], I[i+1], I[i-1], II[i+1] 및 II[i-1] 등은 동일 열, I[j], I[j+1], I[j-1], II[i+1] 및 II[j-1] 등은 동일 행을 나타내며, I 기호는 [-1,0] 방향의 재구성, II는 [0,1] 방향의 재구성에서 전달되는 신호를 의미한다. 그리고 신호의 내용을 나타내는 설명에서 \wedge 은 논리 AND, \vee 은 논리 OR이다.



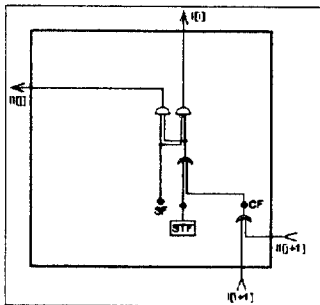
(a) V0 제어논리
(a) V0 control logic



(b) H0 제어논리
(b) H0 control logic



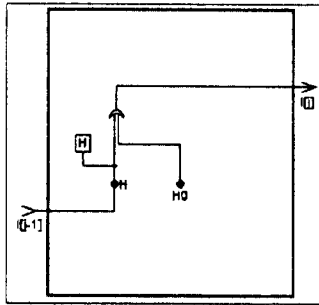
(c) LD와 DD 제어논리
(c) LD & DD control logic



(d) CF 제어논리
(d) CF control logic

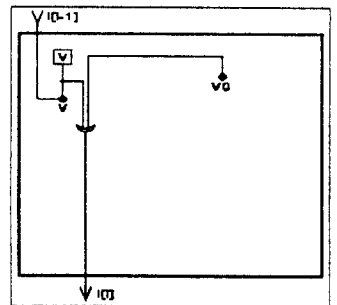
$$I[i+1] : CF[i+1,j] \vee STF[i+1,j]$$

$$II[j+1] : CF[i,j+1] \vee STF[i,j+1]$$



(e) H 플래그와 신호 위한 제어논리
(e) H Flag & Signal control logic

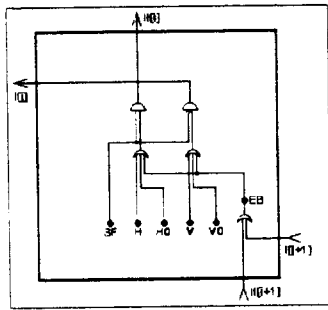
$$II[j-1] : H0[i,j-1] \vee H[i,j-1]$$



(f) V 플래그와 신호를 위한 제어논리
(f) V Flag & Signal control logic

$$I[i-1] : V[i-1,j] \vee V0[i-1,j]$$

그림 6. (계속)
Fig. 6. Continued

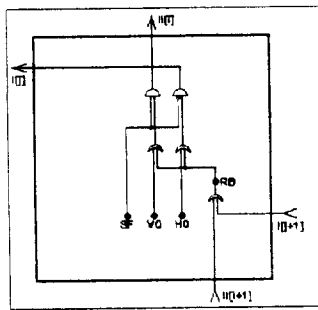


(g) EB 제어논리

(g) EB control logic

$$I[j+1] : V0[i+1] \vee V[i+1] \vee EB[i,j+1]$$

$$II[i+1] : H0[i+1,j] \vee H[i+1,j] \vee EB[i+1,j]$$

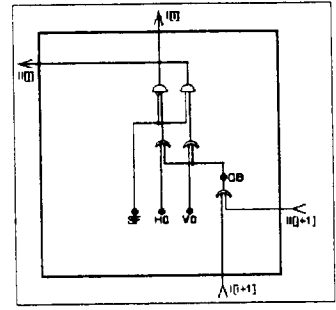


(h) RB 제어논리

(h) RB control logic

$$I[j+1] : H0[i,j+1] \vee RB[i,j+1]$$

$$II[i+1] : V0[i+1,j] \vee RB[i+1,j]$$

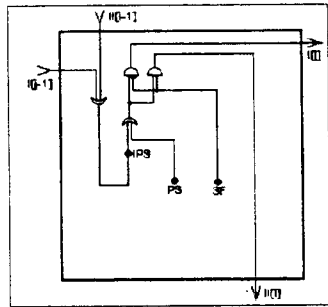


(i) DB 제어논리

(i) DB control logic

$$I[i+1] : H0[i+1,j] \vee DB[i+1,j]$$

$$II[j+1] : V0[i,j+1] \vee DB[i,j+1]$$

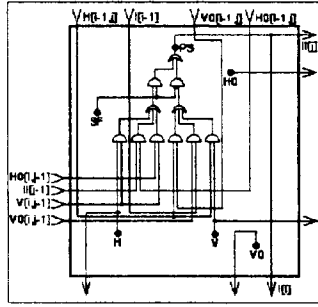


(j) IPS 제어논리

(j) IPS control logic

$$I[j-1] : IPS[i,j-1] \vee PS[i,j-1]$$

$$II[i-1] : IPS[i-1,j] \vee PS[i-1,j]$$

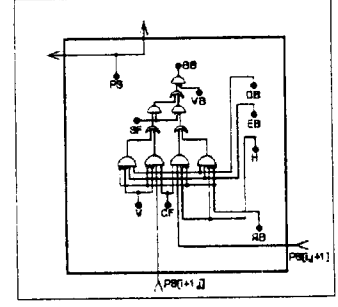


(k) PS 제어논리

(k) PS control logic

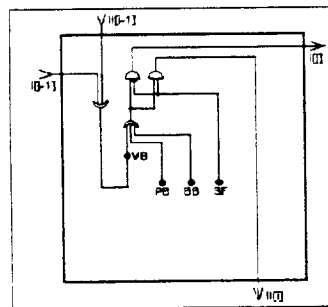
$$I[i-1] : (H[i-1,j] \wedge V0[i-1,j]) \vee (V0[i,j-1] \wedge PS[i-1,j]) \vee (H[i-1,j] \wedge V[i-1,j])$$

$$II[j-1] : (V[i,j-1] \wedge H0[i,j-1]) \vee (H0[i-1,j] \wedge PS[i,j-1]) \vee (V[i,j-1] \wedge H[i,j-1]) \text{end PS_SETTING.}$$



(l) BB 제어논리

(l) BB control logic

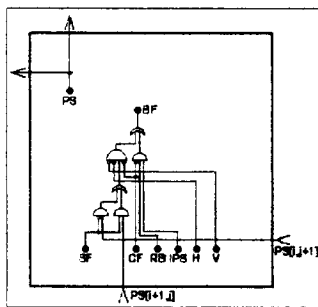


(m) VB 제어논리

(m) VB control logic

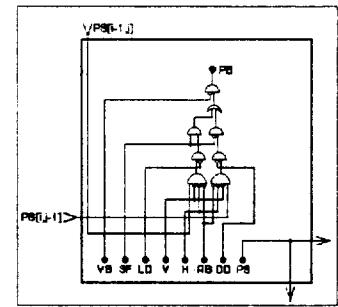
$$I[j-1] : PB[i,j-1] \vee BB[i,j-1] \vee VB[i,j-1]$$

$$II[i-1] : PB[i-1,j] \vee BB[i-1,j] \vee VB[i-1,j]$$



(n) BF 제어논리

(n) BF control logic

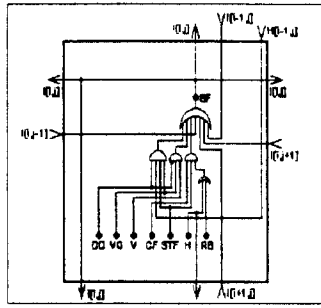


(o) PB 제어논리

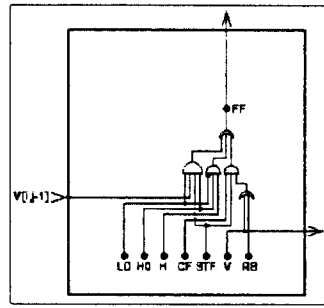
(o) PB control logic

그림 6. (계속)

Fig. 6. Continued



(p) SF 제어논리
(p) SF control logic



(q) FF 제어논리
(q) FF control logic

그림 6. 재구성성을 위한 제어논리(a ~ q)
Fig. 6. The controlling logic for reconfiguration in PE(i,j).(a ~ q)

3) 의사결합생성 유도신호

의사결합처리요소를 생성하기 위한 유도신호는 DB, PS 및 IPS이며, 자신이 수평결합이고, 자신의 우측이나 우측상단에 수직결합 상태의 결합처리요소가 존재하면 사상경로의 교차로 인한 자원부족을 해결하기 위해 생성된다. DB신호는 자신의 우측상단에 수직결합 상태의 결합처리요소가 존재할 때 생성되며, 3.2의 여분할당 단계에서 처리된다. 의사결합생성 유도신호 PS와 IPS를 위한 처리과정은 PS_SETTING이다. 이를 위한 제어논리는 그림 6의 (j)와 (k)이다.

procedure PS_SETTING
begin

```

if SF가 0일 경우 /* [-1,0] 방향의 재구성 */
  if (PE [ij] 윗행에(PE [i-1,k]) T2가 존재(H플래그)하고,
      PE [ij] 위(PE [i-1,j])의 상태가 T1인 경우이거나,
      PE [ij] 바로 앞(PE [ij-1])에 T1이 존재하고, PE [i-1,j]에 의사결합생성 유도신호(PS신호)가 존재할 경우이거나,
      PE [ij] 윗행 (PE [i-1,k]에 T2가 존재(H플래그)하고, PE [ij] 위(PE [i-1,j]에 T1이 존재(V플래그)할 경우)* 1 ≤ k ≤ N·C */
    PE [ij]에 의사결합생성 유도신호 PS가 발생된다.
    PE [ij] 우측(PE [i,k])에 의사결합생성 유도신호가 존재한다는 신호 IPS를 보낸다. /* k = j+1~N·C */
  end if
else /* [0,-1] 방향의 재구성 */
  if (PE [ij] 좌측열에(PE [k,j-1]) T1이 존재(V플래그)하고,
      PE [ij] 좌측(PE [i,j-1])의 상태가 T2인 경우이거나,
      PE [ij] 바로 상단(PE [i-1,j])에 T2가 존재하고, PE [i,j-1]에 의사결합생성 유도신호(PS신호)가 존재할 경우이거나,
      PE [ij] 좌측열 (PE [k,j-1]에 T1이 존재(V플래그)하고, PE [ij] 좌측(PE [i,j-1]에 T2가 존재(H플래그)할 경우)* 1 ≤ k ≤ M·R */
    PE [ij]에 의사결합생성 유도신호 PS가 발생된다.
    PE [ij] 하단(PE [k,j])에 의사결합생성 유도신호가 존재한다는 신호 IPS를 보낸다. /* k = i+1~M·R */
  end if
end if
end SETTING
    
```

4) 의사수평결합 및 의사수직결합처리요소의 생성

이 과정은 [-1,0] 방향의 재구성에서는 의사수평결합처리요소, [0,-1] 방향에서는 의사수직결합처리요소를 생성한다. 이를 위한 처리과정은 BB_SETTING이며, 제어논리는 그림 6의 (l)와 (m)이다.

procedure BB_SETTING
begin

```

if SF가 0일 경우 /* [-1,0] 방향의 재구성 */
  if (PE [ij] 아래에(PE [k,j]) T2가 존재(DB신호)하고, PE [ij] 우측(PE [i,l])에서 EB신호가 오며, PE [ij] 좌측(PE [i,m])이나 우측(PE [i,l])에 T2가 존재(H플래그, RB신호)하지 않고, PE [ij] 좌측에 Δ1, Δ3가 존재(VB신호)하지 않을 경우)
    /* k = i+1~M·R, l = j+1~N·C, m = l-j-1 */
    자신의 상태 = Δ1
    PE [ij] 우측의 수평사상을 위한 PE [i,k]의 H플래그를 세트한다. /* k = j+1~N·C */
    PE [ij] 우측의 같은 행(PE [i,l])에 Δ1이 존재한다는 신호 VB를 보낸다. /* l = j+1~N·C */
  end if
  if (PE [ij] 하단(PE [k,j])에 결합이 존재(CF신호)하고, PE [ij]의 바로 우측(PE [i,j+1])에서 의사결합생성 유도신호 (PS신호)가 오고, PE [ij] 자신이 T2 상태가 아니면서, PE [ij] 좌측(PE [i,m])이나 우측(PE [i,l])에 T2가 존재 (H플래그, RB신호)하지 않고, PE [ij] 좌측에 Δ1, Δ3가 존재(VB신호)하지 않을 경우)
    /* k = i+1~M·R, l = j+1~N·C, m = l-j-1 */
    자신의 상태 = Δ1
    PE [ij] 우측의 수평사상을 위한 PE [i,k]의 H플래그를 세트한다. /* k = j+1~N·C */
    PE [ij] 우측의 같은 행(PE [i,l])에 Δ1이 존재한다는 신호 VB를 보낸다. /* l = j+1~N·C */
  end if
else /* [0,-1] 방향의 재구성 */
  if (PE [ij] 우측에(PE [i,k]) T1이 존재(DB신호)하고, PE [ij] 하단(PE [l,j])에서 EB신호가 오며, PE [ij] 상단(PE [m,j])이나 하단(PE [l,j])에 T1이 존재(H플래그, RB신호)하지 않고, PE [ij] 상단에 Δ2, Δ4가 존재(VB신호)하지 않을 경우)
    /* k = j+1~N·C, l = i+1~M·R, m = l-i-1 */
    
```

```

자신의 상태 =  $\Delta 2$ 
PE [ij] 하단의 수직사상을 위한 PE [kj]의 V플래그
를 세트한다. /* k = i+1~M+R */
PE [ij] 하단의 같은 열(PE [i,l])에  $\Delta 2$ 가 존재한다는
신호 VB를 보낸다. /* l = i+1~M+R */
end if
if (PE [ij] 우측(PE [i,k])에 결함이 존재(CF신호)하고, PE
[i,j]의 바로 하단(PE [i+1,j])에서 의사결합생성 유도
신호 (PS신호)가 오고, PE [ij] 자신이 I1 상태가 아
니면서, PE [ij] 상단(PE [l,m])이나 하단(PE [l,j])
에 I1이 존재(V플래그, RB신호)하지 않고, PE [ij] 상
단에  $\Delta 2, \Delta 4$ 가 존재(VB신호)하지 않을 경우)
/* k = j+1~N+C, l = i+1~M+R, m = l-i-1 */
자신의 상태 =  $\Delta 2$ 
PE [ij] 하단의 수직사상을 위한 PE [kj]의 V플래그
를 세트한다. /* k = i+1~M+R */
PE [ij] 하단의 같은 열(PE [i,l])에  $\Delta 2$ 가 존재한다는
신호 VB를 보낸다. /* l = i+1~M+R */
end if
end if
end BB_SETTING.

```

5) 의사수직결합 및 의사수평결합처리요소의 생성
이 과정은 [-1,0] 방향의 의사수직결합처리요소와
[0,-1] 방향의 의사수평결합처리요소를 생성한다. 이
의 과정은 BF_SETTING이며, 이를위한 제어논리는
그림 6의 (n)이다.

```

procedure BF_SETTING
begin
if SF가 0일 경우 /* [ 1,0] 방향의 재구성 */
if (PE [ij] 하단(PE [k,j])에 결함이 존재(CF신호)하지 않
고, PE [ij]의 바로 우측(PE [i,j+1])에서 의사결합생
성 유도신호 (PS신호)가 오고, PE [ij]의 같은 행에 I
2가 존재(H플래그)하지 않으며, PE [ij]가 I1의 상태
가 아니고, PE [ij]의 열에 I1이 존재(V플래그)하지
않을 경우) /* k = i+1~M+R */
자신의 상태 =  $\Delta 2$ 
PE [ij] 아래의 수직사상을 위한 PE [k,j]의 V플래그
를 세트 시킨다. /* k = i+1~M+R */
end if
if (PE [ij] 좌측(PE [i,m])에 의사결합생성 유도신호 PS가 존
재(PS신호)하고, PE [ij] 우측(PE [i,l])에 I2상태가
존재(RB신호)하고, PE [ij] 하단(PE [k,j])에 결함(CF
신호)이 없을 경우)
/* k = i+1~M+R, l = j+1~N+C, m = l-j-1 */
자신의 상태 =  $\Delta 2$ 
PE [ij] 아래의 수직사상을 위한 PE [k,j]의 V플래그
를 세트 시킨다. /* k = i+1~M+R */
end if
else /* [0,-1] 방향의 재구성 */
if (PE [ij] 우측(PE [i,k])에 결함이 존재(CF신호)하지 않고,
PE [ij]의 바로 하단(PE [i+1,j])에서 의사결합생성 유
도신호 (PS신호)가 오고, PE [ij]의 같은 열에 I1이 존
재(V플래그)하지 않으며, PE [ij]가 I2의 상태가 아니
고, PE [ij]의 행에 I2가 존재(H플래그)하지 않을 경
우) /* k = j+1~N+C */
자신의 상태 =  $\Delta 1$ 

```

```

PE [ij] 우측의 수평사상을 위한 PE [i,k]의 H플래그를
세트 시킨다. /* k = j+1~N+C */
end if
if (PE [ij] 상단(PE [m,j])에 의사결합생성 유도신호 PS가 존
재(PS신호)하고, PE [ij] 하단(PE [l,j])에 I1상태가
존재(RB신호)하고, PE [ij] 우측(PE [i,k])에 결함(CF
신호)이 없을 경우)
/* k = j+1~N+C, l = i+1~M+R, m = l-i-1 */
자신의 상태 =  $\Delta 1$ 
PE [ij] 우측의 수평사상을 위한 PE [i,k]의 H플래그
를 세트 시킨다. /* k = j+1~N+C */
end if
end if
end BF_SETTING.

```

6) 의사수평수직결합처리요소 및 의사수직수평결합 처리요소의 생성

이 과정은 [-1,0] 방향의 의사수평수직결합처리요
소와 [0,-1] 방향의 의사수직수평결합처리요소를 생
성한다. 이의 과정은 PB_SETTING이며, 이를위한 제
어논리는 그림 6의 (o)이다.

```

procedure PB_SETTING
begin
if SF가 0일 경우 /* [-1,0] 방향의 재구성 */
if (PE [ij] 바로 좌측(PE [i,j-1])에 의사결합생성 유도신호
(PS신호)가 오고, I1이 존재(V플래그)하지 않으며, 좌측
(PE [i,m])이나 우측(PE [i,l])에 I2가 존재(H플래
그, RB신호)하지 않을 경우이거나, PE [ij] 자신의 상태
가 I2의 상태이고, 좌측(PE [i,m])에서 의사결합생성
유도신호(PS신호)가 오고, PE [ij]에 EB신호가 올 경
우)
/* l = j+1~N+C, m = l-j-1 */
if (PE [ij] 좌측(PE [i,m])에  $\Delta 1, \Delta 3$ 가 존재(VB신호)
하지 않을 경우) /* m = l-j-1 */
PE [ij] 아래의 수직사상을 위한 PE [k,j]의 V플래
그를 리셋 시킨다. /* k = i+1~M+R */
자신의 상태 =  $\Delta 3$ 
PE [ij] 우측의 수평사상을 위한 PE [i,k]의 H플
래그를 세트한다. /* k = j+1~N+C */
PE [ij] 좌측의 같은 행(PE [i,l])에 I2가 존재한
다는 신호 RB를 보낸다. /* l = l-j-1 */
PE [ij] 우측의 같은 행(PE [i,l])에  $\Delta 3$ 이 존재한
다는 신호 VB를 보낸다. /* l = j+1~N+C */
end if
end if
else /* [0,-1] 방향의 재구성 */
if (PE [ij] 바로 상단(PE [i-1,j])에 의사결합생성 유도신호
(PS신호)가 오고, I2가 존재(H플래그)하지 않으며, 상단
(PE [m,j])이나 하단(PE [l,j])에 I1이 존재(V플래
그, RB신호)하지 않을 경우이거나, PE [ij] 자신의 상태
가 I1의 상태이고, 상단(PE [m,j])에서 의사결합생성
유도신호(PS신호)가 오고, PE [ij]에 EB신호가 올 경
우)
/* l = i+1~M+R, m = l-i-1 */
if (PE [ij] 상단(PE [m,j])에  $\Delta 2, \Delta 4$ 가 존재(VB신호)하
지 않을 경우) /* m = l-i-1 */
PE [ij] 우측의 수평사상을 위한 PE [i,k]의 H플래

```

```

그를 리셋트 시킨다. /* k = j+1~N+C */
자신의 상태 = Δ4
PE [ij] 하단의 수직사상을 위한 PE [k,j] 의 V플래그
그를 세트한다. /* k = i+1~M+R */
PE [ij] 상단의 같은 열(PE [l,j] )에 l1이 존재한다면
신호 RB를 보낸다. /* l = 1~i-1 */
PE [ij] 하단의 같은 열(PE [l,j] )에 Δ4가 존재한다면
신호 VB를 보낸다. /* l = i+1~M+R */
end if
end if
end if
end PB_SETTING.
    
```

7) 재구성 방향 및 불가 판별

이 과정에서는 [-1,0] 방향의 재구성에서 재구성이 불가능 할 경우 SF가 1이 되어 [0,-1] 방향으로의 재구성이 이루어지며, [0,-1] 방향에서도 재구성이 불가능 할 때 FF가 1로 된다. SF는 [-1,0] 방향의 재구성 동안에 재구성 불가능 조건, 즉 한 행(열)에서 두 개 이상의 수평결함(수직결함) 상태의 결함처리요소가 발생하든지 혹은 j=N+C의 여분처리요소에서 자신, PE(i,j),은 수직결함이고, 자신의 상단, PE(i-k,j) (1 ≤ k ≤ i-1),에 수직결함이 존재하면서 자신의 상단 좌측으로, PE(i-1,l) (1 ≤ l ≤ N), 수평결함이 존재하면 세트된다. 이의 처리과정은 SF&FF_SETTING이며, 그림 (p)는 SF, (q)는 FF에 대한 제어논리이다.

```

procedure SF&FF_SETTING
begin
if SF가 0일 경우 /* [-1,0] 방향의 재구성 */
if (PE [ij] 에 [-1,0] 방향의 재구성 불가(SF신호)가 오거나,
PE [ij] 하단에 결함이 존재(CF신호)하고, PE [ij]
자신이 결함이면 PE [ij] 좌측(PE [i,m] )이나
우측(PE [i,l] )에 l2가 존재(H플래그, RB신호)할 경우이거나,
우측 여분이 아니면 l1 상태이고, V플래그가 세트일
경우이거나, 우측여분이면서 PE [ij] 바로 상단(PE
[i-1,j] )에 l2가 존재(H플래그)하고 l1이 존재(V플래그)하며
자신이 결함일 경우) /* l = j+1~N+C, m = 1~j-1 */
[-1,0] 방향의 재구성 불가능 신호 SF를 세트한다.
end if
else /* [0,-1] 방향의 재구성 */
if (PE [ij] 우측에 결함이 존재(CF신호)하고, PE [ij]
자신이 결함이면 PE [ij] 상단(PE [m,j] )이나 하단(PE
[l,j] )에 l1이 존재(V플래그, RB신호)할 경우이거나,
하단 여분이 아니면 l2 상태이고, H플래그가 세트일
경우이거나, 하단여분이면서 PE [ij] 바로 좌측(PE [ij-1]
)에 l1이 존재 (V플래그)하고 l2가 존재(H플래그)하며
자신이 결함일 경우) /* l = i+1~M+R, m = 1~i-1 */
[0, -1] 방향의 재구성 불가능 신호 FF를 세트한다.
end if
end if
end SF&FF_SETTING.
    
```

그림 6의 (p)에서 I [ij] 는 $(CF [ij] \wedge STF [ij] \wedge (H [ij] \vee RB [ij])) \vee (DD' [ij] \wedge VO [ij] \wedge V [ij]) \vee (DD [ij] \wedge H [i-1,j] \wedge VO [ij])$ 이다.

8) 링크 재결합

재구성에서 경로배정은 스위치에 의해 수행되는 스위칭 함수이다. 모든 스위치는 두개의 인접한 처리요소들에 의해 제어되며, 그림 1의 (b)와 같은 상태를 가진다. 수평링크상에 위치하는 스위치는 a, c, d 및 e의 상태를 가지며, 수직링크상에 위치하는 스위치는 b, c, d 및 e의 상태를 가진다. 수평링크상에 위치하는 스위치의 상태(HSW)를 결정하기 위한 과정은 다음과 같다.

$$\begin{aligned}
 SW_a &\leftarrow (((VO'_{ij} \wedge V'_{ij} \wedge VO'_{i,j+1} \wedge Vi_{j+1}) \vee (Vi_{j+1} \wedge V_{i,j+1})) \vee ((PSi_{j+1} \vee IPSi_{j+1})) \wedge SF_{ij}) \\
 SW_c &\leftarrow ((V'_{ij} \wedge V_{i+1}) \vee (VO'_{ij} \wedge V'_{ij} \wedge VO_{i+1})) \\
 SW_d &\leftarrow (((V_{ij} \wedge V'_{ij-1}) \vee (VO_{ij} \wedge VO'_{i,j-1} \wedge V'_{i,j-1})) \wedge ((PS'_{i,j+1} \wedge IPS'_{i,j+1}) \vee SF'_{ij})) \\
 SW_e &\leftarrow (VO_{ij} \wedge VO_{i,j-1})
 \end{aligned}$$

또한 수직링크상에 위치하는 스위치의 상태(VSW) 결정을 위한 과정은 아래와 같다.

$$\begin{aligned}
 SW_b &\leftarrow (((HO'_{ij} \wedge H'_{ij} \wedge HO'_{i-1,j} \wedge Hi_{i-1,j}) \vee (Hi_{i-1,j} \wedge Hi_{i-1,j})) \vee ((PSi_{i-1,j} \vee IPSi_{i-1,j})) \wedge SF'_{ij}) \wedge ((IPS'_{i,j-1} \wedge PS'_{i,j-1}) \wedge TF'_{ij}) \\
 SW_c &\leftarrow ((H'_{ij} \wedge Hi_{i-1,j}) \vee (HO'_{ij} \wedge H'_{ij} \wedge HO_{i-1,j})) \\
 SW_d &\leftarrow (((Hi_{ij} \wedge H'_{i-1,j}) \vee (HO_{ij} \wedge HO'_{i-1,j} \wedge H'_{i-1,j})) \wedge ((PS'_{i-1,j} \wedge IPS'_{i-1,j}) \vee SF'_{ij}) \vee ((PSi_{i-1,j} \vee IPSi_{i-1,j} \vee PSi_{ij} \vee IPSi_{ij}) \wedge TF_{ij})) \\
 SW_e &\leftarrow (HO_{ij} \wedge HO_{i-1,j})
 \end{aligned}$$

여기서 TF는 $TF_{ij} \leftarrow (IPS_{i,N+1} \vee PS_{i,N+1}) \wedge H'_{i,N+1} \vee RB_{ij}$ 이다. 그리고 [0,-1] 방향에서의 스위치 제어는 3.1에서 기술한 대칭조건이 적용되어 수평링크상에 위치하는 스위치 상태는 VSW의 조건을 이용하며 SW_b 는 SW_a , SF'_{ij} 는 SF_{ij} 가 된다. 마찬가지로 수직링크상에 위치하는 스위치 상태는 HSW의 조건을 이용하며 SW_a 는 SW_b , SF_{ij} 는 SF'_{ij} 가 된다.

IV. 시뮬레이션 결과 및 고찰

재구성 문제에 대한 정확성과 효율성을 검증하기 위하여 컴퓨터 시뮬레이션 기법과 수학적인 해석방법을

사용하였다. 성능평가를 위하여 결합은 처리요소에서만 일어난다고 가정하였으며, 각 처리요소에서 결합이 랜덤하게 발생한다고 가정하여, 각 처리요소에서의 결합율과 결합의 수에 따른 신뢰도를 측정하였다. 그림 7은 PE(1,3), PE(1,6), PE(3,2), PE(3,6) 및 PE(4,3)이 결합처리요소인 경우의 시뮬레이션 예로 재구성이 완료된 후의 논리어레이를 나타낸다.

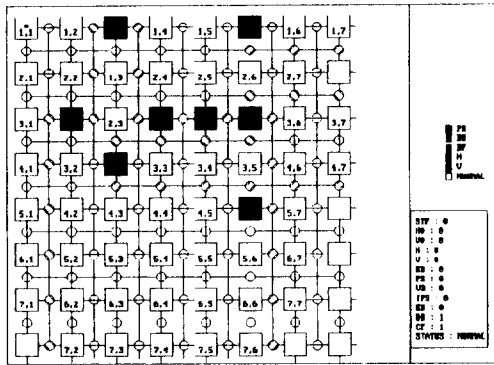


그림 7. 재구성된 7x7 VLSI 어레이
Fig. 7. The reconfigured 7x7 VLSI array.

여기서 물리적으로 인접한 두 처리요소사이의 거리를 단위길이라 가정하면 논리적으로 인접한 처리요소의 수평거리와 수직거리는 단위길이이며, 수평수직 혹은 수직수평거리는 2가 되어 재구성된 어레이 사이에서도 높은 국부성이 유지됨을 알 수 있다.

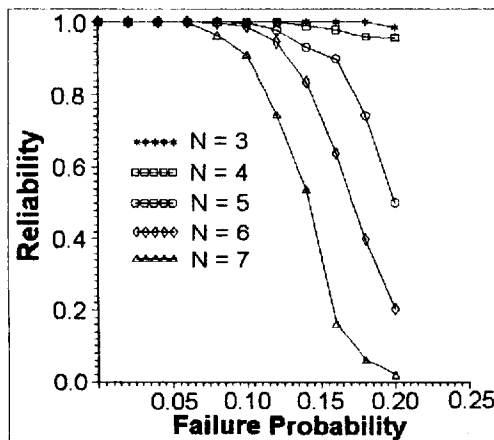


그림 8. 결합율에 따른 평균 신뢰도
Fig. 8. The average reliability on failure probability.

그림 8은 각 처리요소에서의 결합율을 0.00 ~ 0.20로 변화시켜가면서 시뮬레이션한 결과이다. 이것은 이

차원 어레이의 평균 신뢰도를 나타낸 것으로 그래프에서 보듯이 처리요소에서의 결합율이 증가함에 따라 신뢰도가 지수적으로 감소하는 것을 알 수 있다.

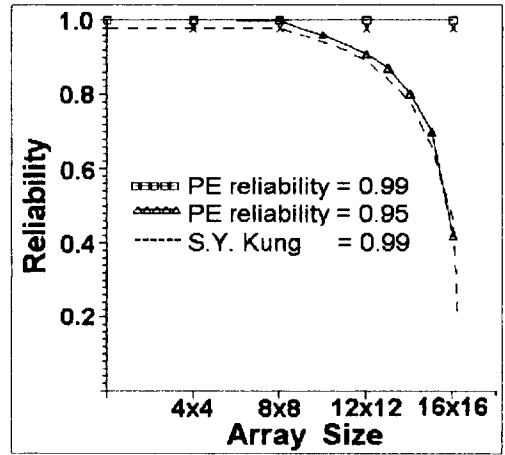


그림 9. 어레이 크기에 따른 신뢰도
Fig. 9. Reliability v.s. array size.

그림 9는 각 처리요소의 신뢰도가 0.99인 경우 S.Y. Kung^[4]이 제안한 방법과 본 논문에서 제안한 방법에 대해서 각각의 물리어레이 크기에 따른 어레이의 신뢰도를 비교한 것이다. 제안한 방법은 처리요소의 신뢰도가 0.99인 경우 100% 재구성 가능하며, 0.95인 경우 Kung이 제안한 재구성 방법과 거의 비슷한 신뢰도를 보이고 있어, 제안한 방법이 제한된 하드웨어 자원으로 높은 재구성율을 얻을 수 있음을 알 수 있다. 한편 Kung은 어레이의 크기가 증가함에 따라 신뢰도가 급격히 떨어지는 것을 해결하기 위해 전체 어레이를 일정한 크기로 분할(partition)하여 재구성하였으나(X 표시) 제안한 방법과 거의 유사한 신뢰도를 나타낸다.

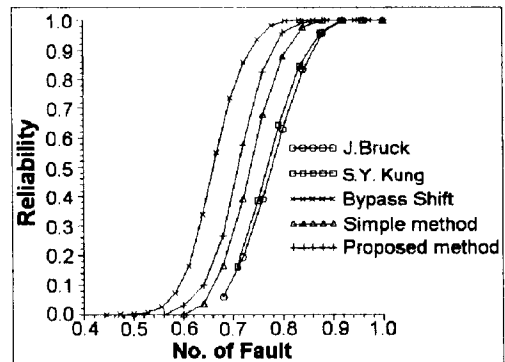


그림 10. 5x5어레이에 대한 각 방법의 비교
Fig. 10. The comparison of each method to array size(5x5).

그림 10은 5×5 어레이에서 결합처리요소의 수를 변화시키면서 어레이의 신뢰도를 분석한 결과이다. 여기서 어레이의 신뢰도는 각 결합의 수를 10,000회 랜덤하게 분산시켜 측정하였다. 沼田一成^[8]이 제안한 Bypass Shift의 재구성 모델은 여분처리요소를 상하 좌우에 배치한 (N+2R)×(N+2C)이며, J.Bruck^[9]와 S.Y Kung^[4] 및 Simple method^[10]의 재구성 모델은 본 논문과 같은 (N+R)×(N+C)이다. J. Bruck의 방법은 여분처리요소를 위와 아래에 배치한 경우이고, S.Y Kung과 Simple method는 여분처리요소를 아래와 오른쪽에 배치한 경우이다. 결합처리요소의 수에 대한 신뢰도에서 제안한 방법은 N개의 결합이 존재하는 경우 99% 이상의 높은 신뢰도를 보인다. BS방법과 동일한 조건은 아니지만 제안한 방법이 비교적 높은 신뢰도를 얻을 수 있다는 것을 알 수 있다.

제안한 방법에 대한 시뮬레이션의 정확성을 검증하기 위하여 시뮬레이션한 결과값과 수학적 신뢰도 해석의 결과값을 비교하였다. 수학적 신뢰도 분석을 간단하게 하기 위하여 어레이에서 PE(i,j), PE(i1,j), PE(i,j1) 및 PE(i1,j1), 1≤i≤M+R and 1≤j≤N+C, 에 결합이 분포된다고 가정한다. 이와같은 결합패턴은 [-1,0] 및 [0,-1] 방향의 재구성에서 재구성 불가능 조건이다. 전체 어레이에서 결합처리요소 4개가 분포되는 가지수 TN은 $\binom{M+R}{i} \binom{N+C}{j} C_4$ 이고 위의 가정과 동일한 위치상에 결합이 분포되는 가지의 수 FN은 다음식과 같다.

$$FN = \sum_{i=1}^{N+C-1} i \times \binom{N+C}{i} C_2$$

그러나 [-1,0] 방향의 재구성에서 결합처리요소의 열 j1이 j=N+C에 위치하고 j와 j1의 물리인덱스 차가 1보다 크면 재구성 가능하며, [0,-1] 방향의 재구성에서 결합처리요소의 행 i/01가 i=M+R에 위치하고 i와 i1의 물리인덱스 차가 1보다 크면 재구성 가능하다. 이것을 B라 두고 순차적인 수식으로 구해보면

$$B = \sum_{i=1}^{N+C-2} i \times 2 \times (N+C-1) - (N+C-2)^2 = \sum_{i=1}^{N+C-2} (3i^2 - i + 1)$$

가 된다. 따라서 어떤 방향에서도 재구성 불가능한 가지의 수 FF는

$$FF = FN - B = \left(\sum_{i=1}^{N+C-1} i \times \binom{N+C}{i} C_2 \right) - \sum_{i=1}^{N+C-2} (3i^2 - i + 1)$$

이 되므로 어레이의 신뢰도는

$$Reliability = \frac{TN - FF}{TN} = \frac{\binom{M+R}{i} \binom{N+C}{j} C_4 - \left[\sum_{i=1}^{N+C-1} i \times \binom{N+C}{i} C_2 - \sum_{i=1}^{N+C-2} (3i^2 - i + 1) \right]}{\binom{M+R}{i} \binom{N+C}{j} C_4}$$

가 된다. 표 2는 수학적 신뢰도 분석의 결과값과 컴퓨터 시뮬레이션한 결과값을 비교한 것이다. 여기서 해석적인 값과 실제 시뮬레이션한 값의 차이가 0.3% 정도이므로 제안한 방법의 시뮬레이션 결과값이 타당성이 있다고 볼 수 있다.

표 2. 시뮬레이션 결과와 수학적 분석 결과 비교

Table 2. The comparison of simulation and analytic results.

어레이크기	시뮬레이션	수학적분석
4×4	0.98468	0.98791
5×5	0.99429	0.99518
6×6	0.99724	0.99761
7×7	0.99849	0.99865
8×8	0.99911	0.99917

V. 결론

본 논문에서는 VLSI 어레이의 재구성에서, 대칭적 의사결합처리요소의 생성 기법을 이용하여 제한된 하드웨어 자원의 효율적인 이용과 연속적인 결합으로 인한 자원부족 문제를 해결하고, 실시간 재구성을 위한 제어논리를 설계하였다.

제안한 방법의 유효성을 검증하기 위하여 재구성 모델에 대한 신뢰도를 분석하였으며, 이를위해 수학적해석 방법과 컴퓨터 시뮬레이션 기법을 사용하였다. 이러한 수학적 분석 결과값들과 컴퓨터 시뮬레이션한 결과값들을 비교해 보면 두 결과들이 매우 유사하여 제안한 방법이 타당성이 있음을 알 수 있다. 어레이 크기를 변화시키면서 신뢰도를 평가한 결과 동일한 조건의 기존 방법에 비해 높은 신뢰도를 보였으며, 특히 각 처리요소의 신뢰도가 0.99인 경우에는 모든 어레이 크기에 대해서 100% 재구성된다. 각 처리요소에서의 결합율과 어레이의 크기를 변화시키면서 시뮬레이션한 결과를 살펴보면, N개의 결합이 발생할 경우 어레이의 크기가 증가할수록 신뢰도는 급격히 감소하는데 이는

결합처리요소 수의 증가에 비해 재구성 능력은 단일트랙 스위치로 인하여 제한되기 때문이다. 한편 재구성된 논리어레이에서 상호연결의 길이는 2로 제한되어 높은 국부성을 유지하며, 재구성을 위한 재료가 각 처리요소에서 동시에 수행되므로 어레이의 재구성에 소요되는 시간은 어레이의 크기에 의존하지 않는다.

본 논문에서는 처리요소의 결함율만으로 성능평가를 행하였으나 실제 웨어퍼상에 구현되어지는 처리요소, 스위치 및 트랙등이 차지하는 면적을 고려한 수율(yield)에 대한 재구성 평가와 메쉬(mesh)구조 뿐만 아니라 트리구조나 다른 유용한 위상(topology) 등에서도 적용될 수 있는 재구성 기법에 대한 연구가 더 진행되어져야 할 것으로 생각된다.

참 고 문 헌

- [1] J.A. Abraham, P. Banerjee, C.Y. Chen, W.K. Fuchs, S.Y. Kuo, and A.L.N. Reddy, "Fault Tolerance Techniques for Systolic Arrays," *IEEE Computer*, pp. 65-75, 1987.
- [2] R. Negrini, M.G. Sami, and R. Stefanelli, "Fault Tolerance Techniques for Array Structures Used in Supercomputing," *Computer*, Vol. 19, No. 2, pp. 403-412, 1986.
- [3] T.A. Varvarigou, et al., "Reconfiguring Processor Arrays Using Multiple-Track Models : The 3 Track 1 Spare Approach," *IEEE Trans. Computer*, Vol. 42, No. 11, pp. 1281-1293, 1993.
- [4] S.Y. Kung, C.W. Chang, and C.W. Jen, "Real-Time Reconfiguration for Fault-Tolerant VLSI Array Processors," *Proc. Real-Time Syst. Symp.*, pp. 46-54, 1986.
- [5] M.G. Sami and R. Stefanelli, "Reconfigurable Architectures for VLSI Processing Arrays," *Proceedings of the IEEE*, Vol. 74, NO. 5, 1986.
- [6] S.Y. Kung, S.N. Jean and C.W. Chang, "Fault-Tolerance Array Processor Using Single-Track switch," *IEEE Trans Computer*. Vol. 38, No. 4, pp. 501-514, 1989.
- [7] J.H. Kim and S.M. Reddy, "On the Design of fault-Tolerant Two-Dimensional Systolic Arrays for Yield Enhancement," *IEEE Trans.Computer*. Vol. 38, No. 4, pp. 515-525, 1989.
- [8] Issei NUMATA and Susumu HORIGUCHI, "WSI Implementation of Mesh-Connected Multiprocessor Systems," *電子情報通信學會, D-I. Vol.176-D-I. No. 10*, pp 514-543, 1993.
- [9] V.P. Roychowdhury and J. Bruck, "Efficient Algorithms for Reconfiguration in VLSI/WSI Arrays," *IEEE Trans Computer*. Vol. 39, No. 4, pp. 480-489, 1990.
- [10] Dong Suk Shin, et al., "Design of Reconfigurable 2-D systolic Array Using Pseudo Faulty Processing," *JTC-CSCC '95*, pp. 495-498, 1995.
- [11] F. LOMBARDI, "Reconfiguration of VLSI Arrays by Covering," *IEEE Trans. Computer-Aided Design*. Vol. 8, No. 9, pp. 952-965, 1989.
- [12] 신동석, 우중호, "의사결합처리요소를 이용한 단일트랙 이차원 시스토크 어레이에서 재구성율의 향상," *전자공학회지, 제 33권, A편, 제 2호*, pp.163-172. 1996
- [13] I. Koren, et al., "Yield and Performance Enhancement Through Redundancy in VLSI and WSI Multiprocessor Systems," *Proc. of IEEE*, Vol. 74, No. 5, pp 699-711, 1986.
- [14] M.A. Breuer, et al., "On Area and Yield Considerations for Fault-Tolerant VLSI Processor Arrays," *IEEE Trans. Computer*, Vol. C33, No. 1, pp. 21-27, 1984.

저 자 소 개

申東錫(正會員) 第30卷 A編 第3號 參照
현재 동명전문대학 전자계산과
조교수

禹鍾鎬(正會員) 第27卷 A編 第3號 參照
현재 부산수산대학교 컴퓨터공학과
교수