

論文96-33A-2-19

의사결합처리요소를 이용한 단일트랙 이차원 시스톱릭 어레이에서 재구성율의 향상

(Improvement of Reconfiguration rate Using Pseudo Faulty Processing Elements on the Single Track 2-D Systolic Array)

申 東 錫 *, 禹 鍾 鎬 **

(Dong Suk Shin and Chong Ho Woo)

요 약

시스톱릭 어레이의 재구성에서는 각 처리요소(PE)에서 결함이 연속적으로 발생할 경우 논리적으로 인접한 처리요소간의 긴 연결길이로 인하여 클럭 속도가 저하되어 어레이의 처리율이 감소되는 단점이 있다. 따라서 어레이가 재구성된 후에도 높은 상호연결의 국부성이 유지되어야 하며, 재구성이 간단한 경로배정소자로 구현되어야 한다. 그러나 이러한 국부성과 단순성의 조건으로 인해 재구성능력이 저하되므로 본 논문에서는 의사결합처리요소의 개념을 이용하여 두가지 조건을 만족시키면서 높은 재구성율을 얻을 수 있는 효율적인 방법을 제안한다. 의사결합처리요소의 개념을 적용하여 동일한 조건의 기존 방법보다 높은 재구성율을 얻었으며, 컴퓨터 시뮬레이션 결과 어레이 크기가 5와 8이고 N개의 결함이 발생한 경우, 재구성율은 각각 97%, 84%가 되었다.

Abstract

In reconfiguration of systolic arrays, a potential disadvantage is that, in the PRESENCE of consecutive faulty PE's, logically connected PE's may be far apart, requiring the reduction of clock speed and thus reducing throughput of the array. Thus it is fundamental to keep locality of interconnections as high as possible even after reconfiguration and to make reconfiguration implemented in the simple routing devices. However requirements of locality and simplicity mean that reconfiguring capability is limited. This paper deals with the issue of developing efficient method for reconfiguration of 2-D systolic arrays which can be achieved high reconfiguration rate, with the two conditions satisfying using concept of pseudo faulty processing element. Applying this concept to reconfiguration of systolic array, we have found that the proposed scheme achieves a higher reconfiguration rate than the previous design of similar condition. The simulation shows that reconfiguration rates are 97%, 84% when N faults occurs on the $N \times N$ array in case of $N=5, 8$ respectively.

* 正會員, 東明專門大學 電子計算科
(Dept. of Computer Science., Dong Myung
Junior College)

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

接受日字: 1995年6月18日, 수정완료일: 1996年1月25日

** 正會員, 釜山水産大學校 컴퓨터工學科

I. 서론

VLSI 기술의 진보에 따라 동일한 다수의 처리요소들이 규칙적으로 연결된 대규모 프로세서 어레이가 널리 이용되고 있다. 그러나 구현되는 VLSI 어레이 구조가 복잡해짐에 따라 실행시간 동안에 그 어레이의 모든 처리요소들이 항상 정확하게 동작한다고 보장하는 것은 불가능하기 때문에 동작중의 신뢰성(reliability)을 보장하는 문제가 제기되고 있다^[1]. 한편, 시스토크 어레이는 서로 인접하고 있는 처리요소들과 통신하는 상호연결의 공간적 국부성(spatial locality)과 한 처리요소에서 다른 처리요소로의 신호 전달을 위해 하나의 단위시간 지연을 필요로 하는 시간적 국부성(temporal locality)을 지닌다. 따라서 시스토크 어레이에서의 결합허용을 위한 재구성은 재구성된 어레이에서 데이터나 계산된 결과들이 각 처리요소간에 전달되는 시간들이 일정하도록 하기 위한 지연과 타이밍이 중요한 요소로서 고려되어야 한다. 이를 위해 재구성된 어레이의 처리요소 사이에 가능한 한 높은 상호연결의 국부성(locality)이 유지되어야 하며, 재구성이 간단한 경로배정 소자로서 구현되어야 한다^[2]. 그러나 국부성과 단순성의 조건은 결국 재구성율의 저하를 초래한다.

지금까지 제안된 대부분의 재구성 방법은^[1,2,7,8,9] 다중트랙을 이용한 버스 연결 구조를 갖고 있으므로 칩 면적이 증가되며, 어레이에서 수평 혹은 수직 방향으로 결합이 연속적으로 발생하면 처리요소간의 연결 거리가 길어져서 클럭 속도가 저하되는 단점이 있다. VLSI 구현시 상호연결 링크가 많은 칩 면적을 차지하므로^[2] 각 수평·수직 링크를 따라 하나의 통신 경로를 두어 인접한 처리요소간에 데이터가 전달되게 하면 칩 면적을 줄일 수 있으며, 클럭 속도의 변형은 처리요소간의 수평·수직 상호연결 거리를 일정하게 제한함으로써 해결할 수 있다.

J. H. Kim^[3] 등은 재구성 가능한 이차원 시스토크 어레이를 위한 $O(N^2)$ 의 시간 복잡도를 갖는 계층적 재구성 방법을 제안했다. 여기서는 단지 한 행에서 하나의 결합처리요소를 포함하는 이차원 시스토크 어레이에 대해서만 고려하였고, 실제로 한 행에서 연속 결합을 허용하기 위해서는 허용될 결합의 수 만큼의 여분의 열을 요구한다. J. Bruck^[5] 등은 단일트랙 스위치를 이용한 이차원 어레이에서 결합처리요소의

수에 의존하는 $O(|F|^2)$ 의 시간 복잡도를 갖는 효율적인 다항 시간 알고리즘을 제안하였다. 그러나 결합처리요소의 여분처리요소로의 할당이 단지 수평·수직 방향으로만 이루어지기 때문에 재구성율이 낮은 단점이 있다. 沼田一成^[6] 등은 $M \times N$ 어레이에 각각 R과 C개의 여분의 행과 열을 갖는 어레이 구조에서 결합처리요소를 가장 많이 포함하고 있는 열을 C개 바이패스한 후 시프트를 통해 재구성하는 알고리즘을 제안하였다. 이 방법은 재구성은 간단하지만 바이패스로 인한 처리요소 이용율이 낮을 뿐만아니라 최대의 결합처리요소를 갖는 열의 결정이 용이하지 않다.

본 논문에서는 위의 사항을 고려하여 단일트랙 스위치를 이용한 $N \times N$ 의 시스토크 어레이에 1행 1열의 여분처리요소를 부가한 시스토크 어레이에서 국부성과 단순성을 만족시키면서 높은 재구성율을 얻을 수 있는 효율적인 재구성 방법을 제안한다. 어레이에서 상호연결의 공간적 지역성과 시간적 지역성을 만족시키기 위해 상호연결 길이를 2로 제한시켜 높은 상호연결의 국부성을 유지하며, 단일트랙으로 인한 재구성율의 저하 문제를 해결하기 위해 의사결합처리요소(pseudo faulty PE)를 이용한다. 재구성 방법은 각 처리요소에서 결합을 체크하는 STF(Self Test Function)의 자기진단정보를 근거로 수평·수직 매트릭스를 구성하고 이들의 값에 따라 인덱스사상을 행한 후, 각 처리요소에서의 링크 재결합을 통하여 문제의 크기에 따른 논리어레이를 구성한다. 재구성을 위한 제어는 각 처리요소에서 이웃하는 처리요소의 상태에 관한 지역 정보(local information)만을 이용하여 각 처리요소에서 동시에 수행되므로 어레이의 재구성에 소요되는 시간은 어레이의 크기에 의존되지 않는다. 제안한 방법의 유효성을 검증하기 위하여 여러가지 평가 기준을 토대로 기존의 방법과 비교·분석하였다.

II. 시스토크 어레이의 재구성

이차원 시스토크 어레이의 재구성은 제한된 여분을 이용하여 잠재적으로 결합이 존재하는 물리어레이(physical array)를 무결함인 논리어레이(logical array)로 구성하는 것이다.

1. 어레이 모델

어레이 모델은 그림 1과 같이 $N \times N$ 의 처리요소

(nonspare)에 1행 1열의 여분처리요소 및 단일트랙 스위치로 구성된 상호연결 링크로 구성되어 있다.

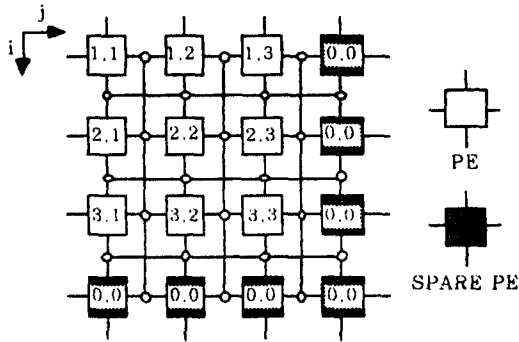


그림 1. 재구성 가능한 시스토크 어레이 모델
Fig. 1. The Reconfigurable Systolic Array Model.

재구성 알고리즘의 전개를 위하여 어레이를 물리어레이와 논리어레이로 구별한다. 어레이의 크기가 N인 경우 물리어레이를 구성하는 처리요소의 집합은 P, ($1 \leq P_{i,j} \leq N+1$)으로 정의하고, 논리어레이를 구성하는 처리요소의 집합을 L, ($1 \leq L_{i,j} \leq N$)로 정의한다. 물리어레이는 $(N+1) \times (N+1)$ 로 구성되며 전체 여분의 처리요소의 수는 $2N+1$ 이다. 어레이에서 처리요소의 위치는 물리적 위치와 논리적 위치로 나타내며 각각 $[i,j]$ 와 (i',j') 로 표시한다. 또한 결함처리요소와 이용되지 않은 여분처리요소의 인덱스는 편의상 (0,0)으로 둔다. 각 처리요소는 결함의 유무에 관계없이 데이터를 상하좌우로 전달 가능하며, 결함처리요소는 연산은 하지 않고 단지 데이터를 전달하는 연결요소로만 이용한다. 재구성 동안에 어레이에서 무결함처리요소는 사용상태(Use)로, 결함처리요소는 각각 수평결함상태(H0)와 수직결함상태(V0)로 구분한다. 그리고 의사결함처리요소는 의사수평결함상태(Pseudo H : PH), 의사수직결함상태(Pseudo V : PV) 및 의사수평수직결함상태(Pseudo HV : PHV)로 구분한다. 의사결함처리요소는 실제 결함이 있는 처리요소가 아니라 성공적인 재구성을 위하여 의사적인 결함상태로 두는 것이다. 재구성을 위한 처리요소의 상태는 그림 2의 (a)와 같다.

재구성을 용이하게 하기 위해 각 처리요소에 자신의 상태 및 인접한 처리요소의 상태에 관한 정보를 나타내는 플래그를 두어 이들의 값에 따라 수평·수직 매트릭스 및 의사결함처리요소를 생성한다.

본 논문에서는 제안한 재구성 방법의 분석을 용이하게 하기 위해 다음과 같이 가정한다.

- ① 어레이의 각 처리요소에 대한 결함은 랜덤하게 분포된다.
- ② 결함은 처리요소에서만 발생되고^[2-7], 각 처리요소에 결함을 판단하기 위한 자체결함체크기능이 있다.
- ③ 결함처리요소는 단지 연결요소(connecting element)로만 사용한다.

2. 단일트랙 스위치

단일트랙은 인접한 두 처리요소 사이의 수평, 수직 링크를 따라 단지 하나의 통신 경로를 갖는것을 의미한다.^[3] 어레이에서 PE $[i,j]$ 와 PE $[i,j+1]$ 을 연결하는 경로를 수평링크, PE $[i,j]$ 와 PE $[i+1,j]$ 을 연결하는 경로를 수직링크라 부른다. 각 처리요소에는 4개의 링크가 존재하며, 처리요소의 상태에 따라 링크의 사용 형태가 다르기 때문에 각 처리요소에서 링크의 상태는 자신과 인접한 처리요소의 상태에 따라 결정된다. 어떤 처리요소의 상태가 사용중이라면 모든 링크가 이웃 처리요소에 접속된다. 그러나 처리요소가 수평결함상태 또는 수직결함상태라면 동서 혹은 남북의 링크만 사용된다. 그리고 재구성된 어레이에서 높은 상호연결의 국부성을 유지하기 위하여 논리인덱스의 영역은 내개의 물리적으로 인접한 처리요소들로 제한되며, 논리어레이의 PE (i',j') 가 사상될 수 있는 물리어레이의 처리요소 집합은 $[i,j]$, $[i,j+1]$, $[i+1,j]$ 및 $[i+1,j+1]$ 이다.

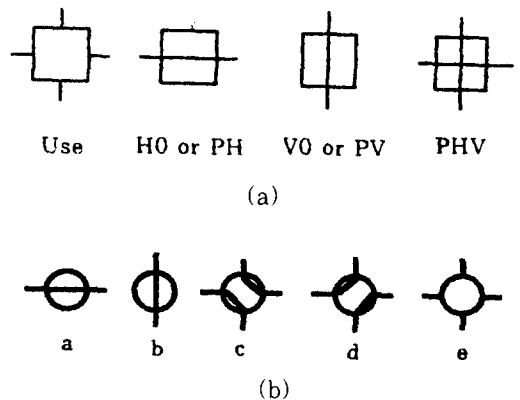


그림 2. (a) 처리요소의 상태, (b) 스위치 상태
Fig. 2. (a) The PE's status, (b) The status of switches.

스위치는 각 처리요소의 동서남북 방향에 위치하며 인접 처리요소간에는 스위치를 통하여 연결된다. 스위치는 전송되는 데이터의 방향을 변경시킬 수 있으며, 어레이를 재구성하기 위한 제어정보도 스위치를 통하여 전달된다. 경로배정은 스위치 요소에서의 스위칭 상태에 의해 결정되며, 각 처리요소의 수평, 수직링크를 결정하기 위한 스위치 요소의 상태는 그림 2의 (b)와 같다.

III. 재구성 알고리즘

재구성 알고리즘은 각 처리요소의 결합에 관한 자기진단정보(STF)와 인접한 처리요소의 상태에 관한 정보를 이용하여 각 결합처리요소에 대한 결합상태를 결정하고, 이 상태에 따라 수평·수직 매트릭스를 구성하며, 의사결합처리요소를 생성한 후 인덱스사상과 상호 연결 링크의 재결합을 통하여 어레이를 재구성한다.

1. 결합처리요소의 상태결정

결합처리요소의 상태결정 과정은 어레이의 각 열의 밑에서 위로, 즉 $[-1,0]$ 방향으로 동시에 수행되며, 결합처리요소의 상태는 $PE[i,j]$ 의 결합여부를 나타내는 자기진단정보의 값과 $PE[i+k,j]$, $i < k \leq N+R$ 로 부터의 결합신호 CF에 의해 결정된다. 각 처리요소에서의 자기진단정보에 의해 $STF=1$ 이면 결합처리요소, $STF=0$ 이면 무결합처리요소이다. 그리고 수평결합상태는 H0, 수직결합상태는 V0로 표시하며, $H0=1$ 이면 수평결합상태, $V0=1$ 이면 수직결합상태이다. 결합처리요소의 상태결정 과정은 다음과 같다.

$PE[i,j]$ 가 결합을 갖는 처리요소이고

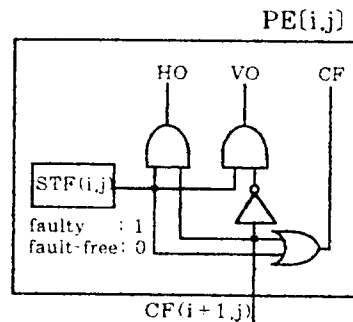
- ① CF가 0이면, $PE[i,j]$ 는 수직결합상태가 되어 V0 및 CF는 1로 세트된다.
- ② CF가 1이면, $PE[i,j]$ 는 수평결합상태가 되어, H0 및 CF는 1로 세트된다.
- ③ 여기서 CF는 $PE[i+k,j]$ 로 부터의 결합신호이며, $k=1, 2, \dots, N+R$ 이다.

결합처리요소의 상태결정 과정에서 H0, V0 및 CF의 값을 세트시키기 위한 제어논리와 H0와 V0의 값에 따른 각 처리요소의 상태는 그림 3과 같다.

2. 수평·수직 매트릭스의 구성과 인덱스사상

결합처리요소의 상태가 결정되면, 즉 H0와 V0, 이 결합처리요소를 여분처리요소로 대체하기 위한 사상경

로(mapping path)가 형성되며, 이 경로를 따라 수평·수직 매트릭스가 구성된다. 사상경로의 형태는 세종류, 즉 수직사상경로, 수평사상경로 및 혼성사상경로가 있다. 그림 4는 각 처리요소에서의 결합상태 결정에 따른 사상경로의 형태이다. 임의의 처리요소가 수평결합상태이면 동일 행의 오른쪽 여분처리요소(S2)에 의해 대체되며, 그 결합위치에서 S2에 이르는 수평사상경로가 형성된다. 수직결합상태도 이와 유사하다. 그림에서 물리어레이 $PE[1,3]$ 은 수평사상경로에 위치하면서 자신은 수직결합상태인 경우이며, $PE[1,3]$ 에서 사상경로의 교차가 일어난다. 이 사상경로를 혼성사상경로라 한다. 이 경로로 인해 자원부족 문제가 발생되며 의사결합처리요소의 생성이 요구된다.



(a)

H0	V0	status of each PE's
0	0	Nonfaulty
0	1	Vertical Faulty
1	0	Horizontal Faulty
1	1	no occurrence

(b)

그림 3. (a) H0, V0와 CF를 위한 제어논리, (b) 각 처리요소의 상태

Fig. 3. (a) The Controlling logic for H0, V0 and CF, (b) The Status of each PE's.

수평·수직 매트릭스의 구성을 위해 H0와 V0신호는 동일 행과 열로 전달되며, 매트릭스의 각 원소는 물리어레이에서 각 처리요소내의 H와 V플래그 값이다. 이 매트릭스는 인덱스사상에서 이용되며 논리어레이 구성을 위한 인덱스 정보를 포함하고 있다. $PE[i,j]$ 가 수평결합이면 $PE[i,k]$, $j < k \leq N+C$ 의 수평 매트릭스 원소인 H가 1로 되며, $PE[i,j]$ 가 수직결합이면 $PE[k,j]$, $i < k \leq N+R$ 의 수직 매트릭스 원소

인 V가 1이 된다.

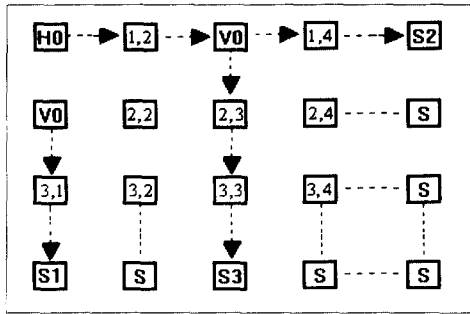


그림 4. 사상경로의 형태
Fig. 4. The type of mapping paths.

인덱스사상은 요구되는 논리어레이를 구성하기 위한 논리인덱스의 물리인덱스로의 사상으로 $\phi(i',j') = [i,j]$ 로서 기술한다. 논리어레이는 재구성된 무결합어레이이며, 성공적인 재구성에서는 모든 $PE(i',j') \in L$ 에 대한 물리어레이의 무결합처리요소 $PE[i,j] \in P$, 즉 $\phi(i',j') = [i,j]$ 가 존재한다는 것을 의미한다. 물리어레이에 결합이 없으면 논리인덱스와 물리인덱스는 동일하며 수평·수직 매트릭스의 값에 따라 인덱스가 이동된다. 인덱스가 이동되는 형태는 수평사상과 수직사상으로 제한되며, 수평사상경로에 따른 수평사상이 우선적으로 결정된 후, 수직사상경로상의 수직사상이 결정된다. 수평사상과 수직사상은 다음과 같다.

- (1) 수평사상
 - ① PE [i,j]의 H=1 이면, $\phi(i',j-1') = [i,j]$
 - ② PE [i,j]의 H=0 이고 PE [i,j+1]의 H=0 이면, $\phi(i',j') = [i,j]$
 - ③ PE [i,j]의 H=0 이고 PE [i,j+1]의 H=1 이면, $\phi(0,0) = [i,j]$
- (2) 수직사상
 - ① PE [i,j]의 V=1 이면, $\phi(i'-1,j') = [i,j]$
 - ② PE [i,j]의 V=0 이고 PE [i+1,j]의 V=0 이면, $\phi(i',j') = [i,j]$
 - ③ PE [i,j]의 V=0 이고 PE [i+1,j]의 V=1 이면, $\phi(0,0) = [i,j]$

그림 5는 논리인덱스의 물리인덱스로의 인덱스사상의 사상 결과이며, 점선은 사상경로이다.

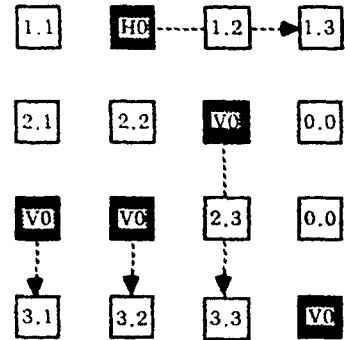
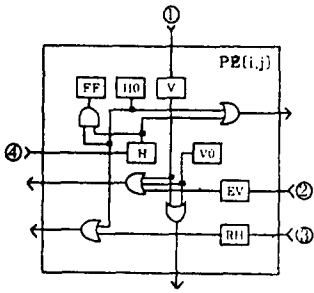


그림 5. 사상경로와 인덱스사상의 결과
Fig. 5. The mapping path and result of index mapping.

한편 결합처리요소의 결합상태가 결정되면 인접한 무결합 처리요소는 결합처리요소로 부터 결합상태에 관한 정보를 받아서 재구성을 위해 자신의 플래그 값을 변화시키며, 의사결합처리요소 생성시에 이용될 인접한 처리요소의 상태에 관한 정보를 나타내는 값, RH 및 EV도 동시에 결정된다. 물리어레이 PE [i,j]에서 이들의 값을 결정하기 위한 동작은 다음과 같다.

```
Controlling logic for H, V, RH, EV, and FF in
reconfiguration process
begin /* ^ : logical AND, v : logical OR */
for all 1 ≤ i ≤ N+R, 1 ≤ j ≤ N+C do in parallel
    Hij ← H0i,j-1 v Hi,j-1
    Vij ← V0,i-1,j v Vi-1,j
    RHij ← H0i,j-1 v RHi,j-1
    EVij ← V0i,j+1 v Vi,j+1 v EVi,j+1
    FFij ← H0ij ^ Hij
all for
end
```

EV의 값은 V0가 1 또는 V가 1인 처리요소에서 좌측하단의 처리요소로 전달된다. 즉 PE [i,j]가 수직 결합이면 PE [k,l], $i \leq k \leq N+R$ 및 $1 \leq l \leq j-1$, 처리요소의 EV가 1이 된다. 그리고 PE [i,j]의 H0가 1이면 $1 \leq k \leq j-1$ 인 처리요소의 RH [i,k]의 값이 1로 세트된다. 한 행에서 두 개 이상의 처리요소에서 수평 결합상태가 존재하면 재구성 불가능(FF: fatal failure)이 된다. 그림 6은 H0, V0를 이용하여 H, V와 RH, EV 및 FF의 값을 결정하는 제어논리이다.



- ① $V0_{i-1,j} \vee V_{i-1,j}$
- ② $EV_{i,j+1} \vee V0_{i,j+1} \vee V_{i,j-1}$
- ③ $H0_{i,j+1} \vee RH_{i,j+1}$
- ④ $H0_{i,j-1} \vee H_{i,j-1}$

그림 6. H, V, RH, EV 및 FF를 위한 제어논리
Fig. 6. The Controlling Logic for H, V, RH, EV and FF.

3. 의사결합처리요소의 생성

상호연결의 높은 국부성을 유지하기 위하여 논리인덱스의 영역은 네개의 물리적으로 인접한 처리요소로 제한된다. 그러나 한 행에서 연속적인 결합이 발생할 경우 비록 여분 할당이 성공적으로 이루어진 경우라 하더라도 상호연결 자원의 부족으로 인하여 재구성성이 불가능한 경우가 존재한다. 이러한 상호연결 자원의 부족은 그림 4에서 설명한 혼성사상경로로 인해 논리적 열(행)의 구성에서 물리인덱스의 열(행)의 순서가 역전되어 발생한다. 이로인한 재구성율의 저하를 해결하기 위하여 의사결합처리요소를 이용한다.

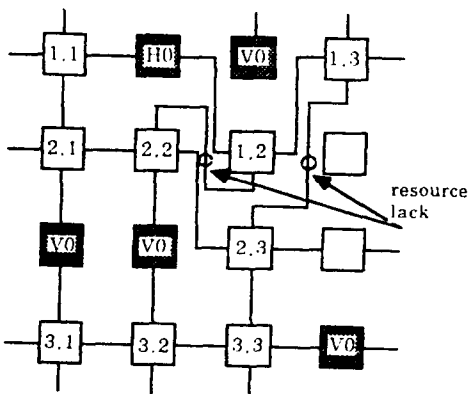


그림 7. 인덱스사상 후의 자원부족
Fig. 7. The Resource lack after index mapping.

그림 7은 자원부족(resource lack)의 예이다(O 표시). 여기서 PE [2.2]를 의사수평결합상태로, PE

[3.4]를 의사수평수직결합상태로 두면 위의 문제는 해결된다.

의사결합처리요소에서 의사수평결합상태의 처리요소와 의사수평수직결합상태의 처리요소는 자신의 H0를 1로 세트하며, 의사수직결합상태의 처리요소는 자신의 V0를 1로 세트한다. 어레이에서 각 처리요소가 의사결합처리요소로 전환되는 조건은 다음과 같다.

- ① PE [i,j]의 V0와 H가 1이거나 V와 H가 1인 경우 : 이 조건은 자신이 수직결합상태이고 자신의 왼쪽에 수평결합상태가 존재하거나, 자신이 수직사상경로상에 위치하고 있고 자신의 왼쪽에 수평결합상태가 존재하는 경우이다.
- ② PE [i,j]의 EV와 H가 1인 경우 : 이 조건은 자신이 수평결합상태이고 자신의 오른쪽 위에 수직결합상태가 존재하는 경우이다.

이 두가지 조건에서 ①과 ②의 조건에 따른 의사결합처리요소의 생성과정은 각각 그림 8, 그림 9와 같다.

```

procedure Pseudo PE generation in case of V0=1, H=1 or V=1, H=1
in PE[i,j]
begin
  if((PE[i,j].V=1 or PE[i,j].V0) and PE[i,j].H=1)
  begin
    k=i
    while(PE[k,j].H = 1)
      k=k+1
      begin
        if(PE[k,j-1].CF=1 and PE[k,j-1].H = 0 and
PE[k,j-1].RH=0)
          PE[k,j-1].H0=1 /* PH의 생성 */
        else
          PE[k,j-1].V0=1 /* PV의 생성 */
        endif
      end
      begin
        if(PE[k,j+1].V0=0 and PE[k,j+1].V=0 and
PE[k,j+1].H=0
and
PE[k,j+1].RH=0)
          PE[k,j+1].H0=1 /* PHV의 생성 */
        elseif(PE[k,j+1].V0=0 and PE[k,j+1].RH=1)
          PE[k,j+1].V0=1 /* PV의 생성 */
        elseif(PE[k,j+1].V0=1 and PE[k,j+1].RH=0)
          PE[k,j+2].H0=1 /* PHV의 생성 */
        endif
      end
    endwhile
  end
end
end
    
```

그림 8. 조건 1의 의사결합처리요소 생성
Fig. 8. The pseudo PE's generation of case 1 in PE [i,j].

```

procedure Pseudo PE generation in case of H0=1 and EV=1
in PE[i,j]
begin
  if((PE[i,j].H0=1 and PE[i,j].EV=1)
  begin
    k=i
    while(PE[k,j].EV=1 and k>0)
    begin
      k=k-1
      if(PE[k,j].H=0 and PE[k,j].RH=0)
        PE[k,j].H0=1 /* PH의 생성 */
      endif
    end
  endwhile
end
end
end
    
```

그림 9. 조건 2의 의사결합처리요소 생성
 Fig. 9. The pseudo PE's generation of case 2 in PE [i,j].

4. 링크 재결합

링크의 재결합은 동서남북 방향의 인접한 처리요소의 상태와 자신의 상태에 따라 이루어지며, 이들의 상태에 따라 관련 스위치의 상태가 결정된다. 표 1은 수평링크에 위치하는 스위치 제어를 위한 처리요소의 상태이다. 각 스위치의 상태는 표 1에서 정의된 각 처리요소의 상태에 따라 결정되며 표 2와 같다. 또한 수직링크에 대한 링크의 재결합도 이와같은 방법으로 이행된다.

표 1. 처리요소의 상태와 여분할당

Table 1. The PE's status and mapping path.

0	fault-free and not on vertical mapping path
1	fault-free and on vertical mapping path
2	faulty and on vertical mapping path
3	faulty and not on vertical mapping path

표 2. 스위치의 상태

Table 2. The status of switch.

PE[i,j] \ PE[i,j+1]	0	1	2	3
0	a	c	c	a
1	d	a	d	d
2	d	c	e	d
3	a	a	c	x

물리어레이 PE [1,2], PE [1,3], PE [3,1] PE [3,2] 및 PE [4,4]가 결함인 경우, 재구성이 완료된 후의 논리어레이는 그림 10과 같다. 물리적인

로 인접한 두 처리요소 사이의 거리를 단위길이로 가정하면 논리적으로 인접한 두 처리요소, PE [x,y]와 PE [i,j], 사이의 거리는 $d((x,y)(i,j)) = |x-i| + |y-j| \leq 1$ 이다. 따라서 논리적으로 인접한 처리요소의 수평거리와 수직거리는 단위길이가 1이며, 수평수직 거리는 2가 된다.

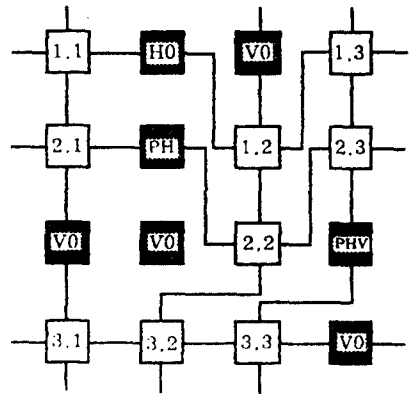


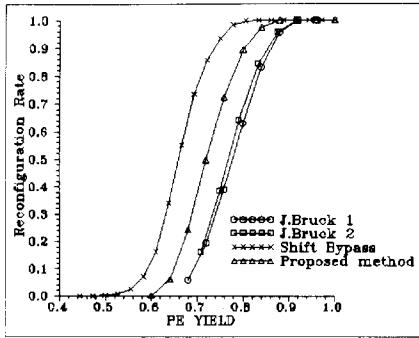
그림 10. 재구성된 3x3 이차원 시스토픽 어레이.
 Fig. 10. The reconfigured 3x3 2-D systolic array.

IV. 컴퓨터 시뮬레이션 및 고찰

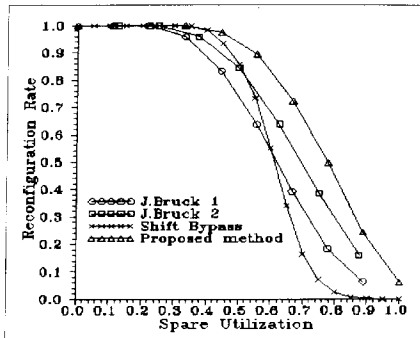
어레이의 재구성 문제에 대한 정확성을 검증하기 위하여 어레이의 크기에 따른 컴퓨터 시뮬레이션을 수행하였다. 어레이에서 각 결함의 수를 랜덤하게 분산시켜서 10,000회 수행하여 재구성율을 측정하였다. J. Bruck와 沼田一成 등의 재구성 방법들과 본 논문에서 제안한 방법에 대한 결함처리요소의 수에 따른 재구성율과 여분처리요소 이용율에 따른 재구성은 그림 11과 같다. 물리어레이의 크기가 5x5인 경우 결함처리요소의 수에 대한 재구성율에서 제안한 방법은 N개의 결함이 존재하는 경우 97% 이상의 높은 재구성율을 보인다. J. Bruck-1은 여분처리요소를 위와 아래에 배치한 경우이고, J. Bruck-2는 여분처리요소를 아래와 오른쪽에 배치한 경우이다. 그리고 沼田一成의 Shift Bypass는 2행 2열의 여분처리요소를 아래와 오른쪽에 배치한 것이다. 물리어레이의 크기가 8x8인 경우에 각 재구성 방법들과 제안한 방법에 대한 결함처리요소의 수에 따른 재구성율과 여분처리요소 이용율에 따른 재구성율은 그림 12와 같다. 결함처리요소의 수에 따른 재구성율에서 제안한 방법은 N개의 결함이 발생한

경우 84% 이상의 재구성율을 보였다.

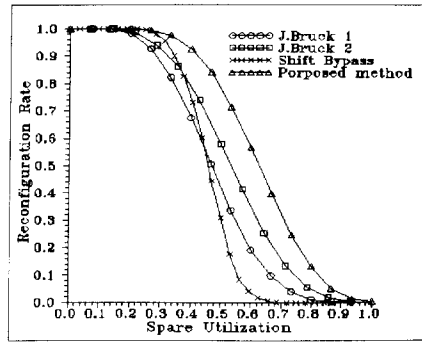
제한한 재구성 방법의 효율성을 검증하기 위해 재구성율과 여분처리요소 이용율, 상호연결 링크의 복잡도, 여분하드웨어로 인한 면적복잡도 등과 같은 평가척도를 이용한다.



(a) reconfiguration rate versus number of faults



(b) reconfiguration rate versus spare utilization



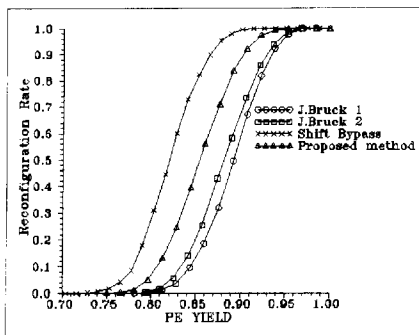
(b) reconfiguration rate versus spare utilization

그림 12. 어레이 크기 8×8에 대한 각 방법의 비교
Fig. 12. The comparison of each method to array size(8×8).

그림 11과 12에서 Shift Bypass방법이 가장 높은 재구성율을 얻을 수 있으나 2행 2열의 여분처리요소를 부가했기 때문이며, 가장 결함이 많이 포함된 열을 바이패스 하여 재구성하므로 여분처리요소의 이용율은 낮다. 동일한 조건에서 본 논문과 비교하면 제한한 방법이 J. Bruck의 방법에 비해 높은 재구성율과 여분처리요소 이용율을 보인다. 상호연결 링크의 복잡도는 인접한 처리요소간의 거리로서 평가되며, 이는 상호연결의 국부성을 나타낸다.

그림 10에서 논리적으로 인접한 두 처리요소 사이의 거리는 2를 초과하지 않으므로 높은 상호연결의 국부성을 나타낸다. 여기서 PE [1,2]와 PE [2,3]은 물리어레이의 서로다른 행에 위치하지만 논리어레이에서는 동일한 행에 속한다. 이러한 두 개의 수평, 수직으로 연결된 논리적으로 인접한 처리요소간의 거리는 2단위가 증가한다. 그러나 PE [2,1]과 PE [4,1]과 같은 두 개의 수직으로 연결된 논리적으로 인접한 처리요소간의 거리는 증가하지 않는다. 이는 결함처리요소를 데이터를 전달하여 주는 연결요소로 이용하기 때문이다. 또한 논리적으로 수평으로 연결된 인접 처리요소간에도 수평, 수직으로 연결된 인접 처리요소간의 거리는 2단위 증가하지만 두 개의 수평으로 연결된 논리적으로 인접한 처리요소간의 거리는 증가하지 않는다. 여분하드웨어의 면적복잡도를 상호연결의 링크여분(link redundancy)만으로 평가하면 결함허용을 고려하지 않은 시스토크 어레이에 비하여 재구성으로 인한 수평링크와 수직링크가 요구되므로 50% 증가된다. 칩면적의 복잡도와 처리요소간의 국부성을 고려하면 단

그림 11. 어레이 크기 5×5에 대한 각 방법의 비교
Fig. 11. The comparison of each method to array size(5×5).



(a) reconfiguration rate versus number of faults

일트랙이 다중트랙보다 유리하며, 처리요소에만 결함이 발생한다는 가정에도 타당성이 있다. 그러나 전체 처리요소의 수에 비해 상대적인 여분처리요소의 수가 작기 때문에 어레이의 크기가 증가하면 상대적으로 재구성율이 낮아진다. 따라서 그림 1의 재구성 모델에서, 본 논문에서 고려한 $[-1, 0]$ 방향의 재구성뿐만 아니라 $[1, 1]$ 을 축으로 대각선에 대칭되는 $[0, -1]$ 방향의 재구성을 동시에 고려하면 $[-1, 0]$ 방향에서 재구성이 불가능한 경우에도 $[0, -1]$ 방향에서 재구성이 될 수 있으므로 보다 높은 재구성율을 얻을 수 있을 것이다.

VI. 결 론

시스토크 어레이의 재구성시, 처리요소들에서 결함이 연속적으로 발생할 경우 논리적으로 인접한 처리요소간의 연결길이가 길어져서 동작클럭의 속도가 저하되어 어레이의 처리율이 감소되므로 재구성된 어레이는 높은 상호연결의 국부성이 유지되어야 하며, 재구성이 간단한 경로배정소자로 구현되어야 한다. 이러한 국부성과 단순성의 조건은 어레이의 재구성율을 저하시키므로 본 논문에서는 $N \times N$ 의 처리요소에 1행 1열의 여분처리요소 및 단일트랙 스위치로 구성된 시스토크 어레이에서 의사결합처리요소의 개념을 이용하여 두가지 조건을 만족시키면서 높은 재구성율을 얻을 수 있는 효율적인 방법을 제안하였다.

어레이의 재구성은 ①자신의 상태와 인접한 처리요소의 상태를 나타내는 값이 제어논리에 의해 결정되어 ②상상경로에 따른 수평·수직 매트릭스가 구성되고 ③자신과 인접한 처리요소의 상태에 관한 정보와 수평·수직 매트릭스를 근거로 의사결합처리요소를 생성한 후, ④인덱스상 및 링크 재결합을 통하여 이루어진다. 특히 의사결합처리요소의 개념을 이용한 결과 제한된 여분하드웨어 자원을 보다 효율적으로 사용하여 상상경로의 교차로 인한 자원부족 문제를 해결하고, 높은 재구성율을 얻을 수 있었다. 또한 어레이에서 상호연결의 공간적 국부성과 시간적 국부성을 만족시키기 위해 상호연결 길이를 2로 제한시켜 높은 국부성을 유지한다. 그리고 재구성을 위한 제어가 각 처리요소에서 인접한 처리요소의 상태에 관한 지역정보만을 이용하여 각 처리요소에서 동시에 수행되므로 어레이의 재구성에 소요되는 시간은 어레이의 크기에 의존하지 않는다.

여러 가지 평가기준을 토대로 기존의 대표적인 J.

Bruck와 沼田一成 등의 재구성 방법과 비교, 분석하였다. 컴퓨터 시뮬레이션 결과 어레이의 크기가 5와 8일 때 재구성율은 각각 97%, 84%가 되었다. 제안한 방법과 J. Bruck 방법의 재구성율을 비교하면 제안한 방법의 재구성율이 어레이의 크기가 5인 경우에는 13%, 8인 경우에는 26%정도 높다.

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," Vol. 19, No. 2, pp. 403-412, 1986.
- [3] M.G. Sami and R. Stefanelli, "Reconfigurable Architectures for VLSI Processing Arrays," *Proceedings of the IEEE*, Vol. 74, NO. 5, 1986.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] F. LOMBARDI, "Reconfiguration of VLSI Arrays by Covering," *IEEE Trans.*

Computer-Aided Design. Vol. 8, No. 9, pp. 952-965, 1989.

[9] P.K. Rhee, et al., "A Novel Reconfiguration Schemes for 2-D Processor Arrays," *Int'l Conf. Computer Aided Design*, pp. 230-233, 1989.

[10] I. Koren, et al., "Yield and Performance Enhancement Through Redundancy in VLSI and WSI Multi processor Systems," *Proc. of IEEE*, Vol. 74, No. 5,

pp 699-711, 1986.

[11] 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.

[12] 신동석, 이수진, 우종호, "결합허용 이차원 시스토릭 어레이에서 재구성율의 향상에 관한 연구," *대한전자공학회 하계종합학술대회 논문집*, 제18권, 제1호, pp 349-352, 1995

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 컴퓨터공학
과 교수