格子結合型マルチプロセッサシステムの再構成方式に関する研究

東北大学大学院工学研究科
情報工学専攻
沼田 一成

目 次

図目次

第１章

序論

1.1　本研究の背景と目的

情報処理技術の発展により我々を取り巻く環境は急激に変化した．特にコンピュータ技術の発展はめざましい．コンピュータは今や我々の生活のいたるところに入りこみ，もはやそれなしでの生活など考えられないところまでになった．

コンピュータの応用分野の中で最近特に注目されているものに次のようなものがある．

- 科学技術計算等，大規模シミュレーション．
- ニューラルネットワーク等，人工知能．

例えば，分子モデルのシミュレーションを考える．原理的にはニュートンの運動方程式を１つ１つの分子に対して解くということにすぎないが，分子の数が増えてくるにつれ計算量が膨れあがる．また，地球規模の大気の流れをシミュレーションすることを考える．原理的には，流体の基礎方程式を解くということにすぎないが，そのデータ量の巨大さのため計算できていないのが現実である．

もし分子モデルのシミュレーションが計算可能となれば，巨大な実験システムを組み，実際に物質を作らなくとも，コンピュータ上でどのような物性を示すかがわかる．また，大気のシミュレーションが可能となれば，地球規模での天候が予測可能となる．

人間の脳をシミュレーションするニューラルネットワークの問題を考える．人間の脳は140億個以上の細胞から構成され，その１つ１つがネットワーク結合され，細胞１つ１つが単純な演算装置のようなものを持っていると言われる．細胞の構造自体は単純であるが，その数の多さゆえシミュレーションを困難にさせている．

人類はこれらの問題に挑戦するため，日夜コンピュータの改良を試み，進歩させてきた．そして，より速いコンピュータが欲しいという人類の要求は集積回路の技術進歩をもたらし，年々集積回路素子は高密度，高速化された．例えばプロセッサに関しては数年前までは8bitプロセッサが主流であったにが，今や32bitもしくは64bit数百MHz程度のプロセッサが主流である．

しかし単一プロセッサのスピードはもはや限界と言ってもよい．そこで，多数のプロセッサを並列化して用いる方式が近年研究されてきた．

現在，10^2～10^4程度のマルチプロセッサを実装した並列コンピュータが実現されており，そのいくつかは商用マシンとして実際に機能している．

しかし，現在のプロセッサ数では，先に挙げたこれらの問題をモデルを縮小化や単純化したレベルで行われているが，本来求めたいレベルの規模では解くには至っていない．例えば，物質はアボガドロ数に代表されるように，10^{23}程度の分子の集まりであるが，現在の並列マシンでは10^3～10^4程度しか扱えない．これらの問題を解くためには，10^{10}程度のプロセッサが必要であると考えられる[17].

プロセッサが増えてくると故障の問題が無視できなくなってくる．故障率が10^{-9}時間のプロセッサがN個あったとする．t時間後に故障するプロセッサの数nは近似的に次のように表わされる．

$$n = N(1 - e^{-\lambda t})$$

ここで，λは単位時間あたりの故障確率である．これを変形しn個故障するまでの時間を求めると，

$$\begin{aligned} t &= \log(\tfrac{N-n}{N})\tfrac{1}{\lambda} \\ &= \log(1 - \tfrac{n}{N})\tfrac{1}{\lambda} \end{aligned}$$

となる．ここで，nが小さいところでは，

$$t = \tfrac{n}{N\lambda}$$

となる．ここで，Nが10^6程度とすると，1つPEが壊れる確率は10^{-9}であるからこのシステムは1000時間しか正常に動作しない．実際のPEの故障率は10^{-7}程度であるからこの時間はもっと少ない．逆に5%ほど故障してもよいとすると，その20倍の時間動作することになる．つまり，欠陥箇所救済技術，フォールトトレランス(Fault Tolerance)が必要不可欠なものとなる．すなわち故障プロセッサが数個存在したとしてもシステムを稼働できなければ話にならない．

システムのパッケージングの問題もある．これを解決する手段として近年一枚のウェーハ上に実用規模のシステムを構築するWSI(ウェーハスケールインテグレーション,Wafer Scale Integration) が注目されている [20][21]．もしこのようなウェーハ上に超並列システムを構築することが可能であれば，チップ間の配線が無くなるため，クロックのディレイ等が少なくなりシステムの高速化，小電力化，小型化が可能になる．

しかし製造時にウェーハに発生する欠陥は現在の集積回路技術では避けられない問題であり，特にウェーハ口径が大きくなる現在, WSI における歩留まり (Yield) は低く実用には至っていないのが現状である．このため WSI における歩留まりを向上させるためにも欠陥箇所救済技術は必要不可欠である．

現在では従来の VLSI システムに比べて WSI システムは一般に実用化には不利とされているが，VLSI 技術もその線幅は限界に達しており新しい欠陥箇所救済技術の必要性が望まれている．また，現在書換え可能なゲートアレイ FPGA (Field Programmable Gate Array) が注目されつつある [23]．IC 内部でユーザが簡単にロジックや接続形態を変更できるため，WSI の再構成アーキテクチャとして注目されている．

まとめると

- 超並列システムではプロセッサ数が巨大になればなるほど，フォールトトレランスの技術は不可欠である．
- システムを WSI 等にパッケージングしようとすると，ウェーハ上に発生する欠陥の故障救済技術は不可欠である．

前者はシステムが構築された後，後者は構築される前という違いはあるが，いずれにしてもなんらかのフォールトトレランス技術は不可欠である．

超並列システムのプロセッサ間結合形態はシステムを構築する上で重要な問題である．代表的なものとしてハイパーキューブ結合，多段網結合，格子型結合等がある．格子結合型マルチプロセッサシステムは最も基本的で単純なネットワーク結合で，超並列システムの実用化が大いに期待されている．[19]

そこで本研究ではこれらの背景をふまえ格子結合型マルチプロセッサの再構成問題に注目する．これは古くからもっともよく研究されている分野であるにもかかわらず，その効率的な解法は未だ発見されていない．そこで本研究では新たに再帰的シフトという手法を提案し，この問題に取り組む．本手法には以下のような特徴がある．

- ローカルな情報だけで構成できる．

- アレイのサイズ，冗長PE等に制限がなく，超並列システムに向いている．
- 平面構造だけではなく3次元や高次元にも拡張可能である．

本論文では，平面配置が容易であることから最初に，格子結合ネットワークについて述べる．その結果をふまえ，平面配置された格子結合ネットワークを多層重ねた3次元格子結合ネットワークの再構成技術について論じる．

1.2 本論文の構成

本論文の構成は次の通りである．

第1章 序論であり，本研究の背景と目的を述べる．

第2章 本研究でターゲットとしている格子結合マルチプロセッサについて説明し，現在までに提唱されている Kung[5] の冗長化手法および問題点について述べる．

第3章 格子結合マルチプロセッサを再構成するために新たに二つの基本手続き,「バイパス」および「再帰的シフト」を基本とした，FS法およびBS法の2つの再構成手法を提言し，性能評価を行う．これは位置に依存した静的手法でアルゴリズムも確実に停止する．

第4章 アルゴリズムの停止性は保証していないが，位置に依存せず，高い再構成率を得ることができる動的再構成手法のHS法を説明する．

第5章 静的手法，動的手法を様々な角度から性能評価を行い比較検討を行う．

第6章 本手法を階層構造を持つ大規模システムに適用する際の再構成手法および，平面構造である格子結合マルチプロセッサの3次元への拡張について説明し，性能評価を行う．

第7章 本研究の結論・今後の課題について述べる．

第2章

格子結合マルチプロセッサ

2.1 まえがき

格子結合型マルチプロセッサはプロセッサを格子状に単純に配置したもので，メッシュ型やトーラス型がある (図 2.1)．トーラス型はメッシュ型の各 PE 東西南北の端のポートをそれぞれ結合したものであり，格子のサイズを $N \times N$ とすると PE 間の距離は最大で $N/2$ である．これに対してメッシュ型は最大で N と若干大きい．

いずれも基本的なプロセッサの結合形態であるため，現在まで様々な故障救済技術が提案されている．M.Chean ら [3] の分類によると，大きく次の 2 つに分類される．

- 時間冗長 (Time Redundancy)
- ハードウェア冗長 (Hardware Redundancy)

時間冗長とはある PE に欠陥の発生した場合，別の PE がその代用を行なうものである．例えば，R.Negrini ら [9] は欠陥が発生によりスイッチ回路を切り替え，自分自身を 2 度データが流れるようにした時間冗長アーキテクチャを提案している．しかし，このアルゴリズムではクロックのディレイが生じ，PE 間の同期が崩れるため演算速度に対する影響が大きい．そのためパイプライン的な処理を行う高速動作には一般に不向きである．

一方ハードウェア冗長は，あらかじめ通常の PE の他に予備のプロセッシングエレメントを用意しておき，欠陥が発生した場合にその予備を用いる方法で，さらに大きく次のように二つに分類されている．

- ローカル冗長 (Local Redundancy)
- グローバル冗長 (Global Redundancy)

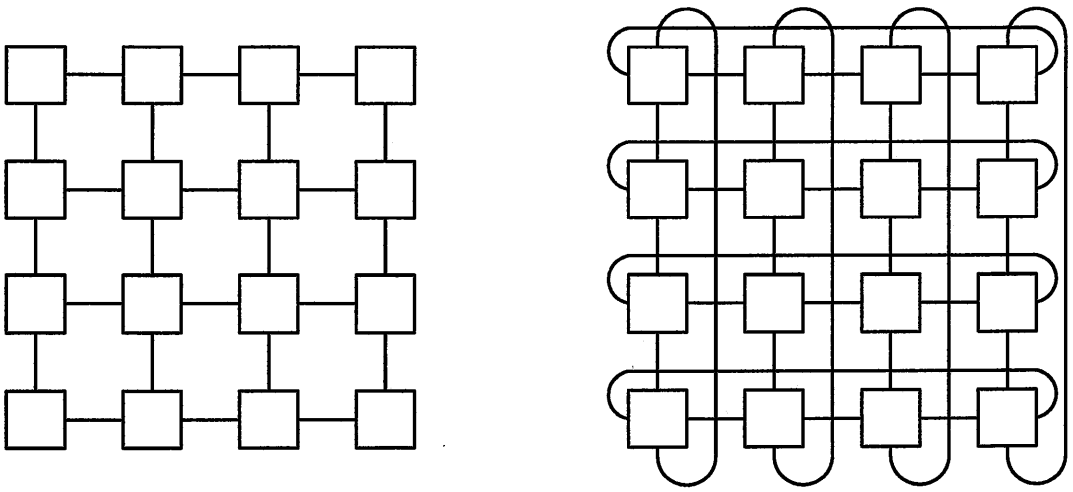


図 2.1　メッシュ結合アレイとトーラス結合アレイ

ローカル冗長とは，PE の予備をローカルに用意しておく方法である．すなわち，ある PE にはある特定の予備しか対応しないため，故障の分布がまばらにしか発生しない場合には極めて不利になる．しかし冗長回路などの構成が簡単にできるという利点がある．例えば A.D. Singh[11] は 4 個の PE の中心部にスペアを置く方法を提案している．

これに対して，グローバル冗長とはスペアをグローバルに用意しておく方法で，もっとも多くの研究がなされている分野の 1 つである．'K out of K+R' 法や 'Harvest' 法がある．前者は，K 個の PE に R 個の冗長な PE を付加して，そこからいかに K 個の PE を得るかを目標とする方式で，後者は故障を含む K 個の PE からいかに大きい PE 集団を得るかを目標とする方式である．

いずれも，次の 2 つのタイプに分類されている．

- 行または列毎等に行うセット切り替え (Set Switched) item PE 毎に切り替えや置き替えを行うプロセッサ切り替え (Processor Switched)

セット切り替えの代表はバイパス (Bypass) で，例えば Kuo ら [6] はグラフを用いて最適なバイパス行および列を検索するアルゴリズムを提案している．プロセッサスイッチはかなり多く研究されており [8][2][18][22] [4][10][12] [13][7]，その代表的なアレイ構造が Kung ら [5] の提唱したスイッチとトラックを用いる $1\frac{1}{2}$ モデルである．本章ではこの Kung らの提唱したモデルについて説明し，その問題点を分析する．

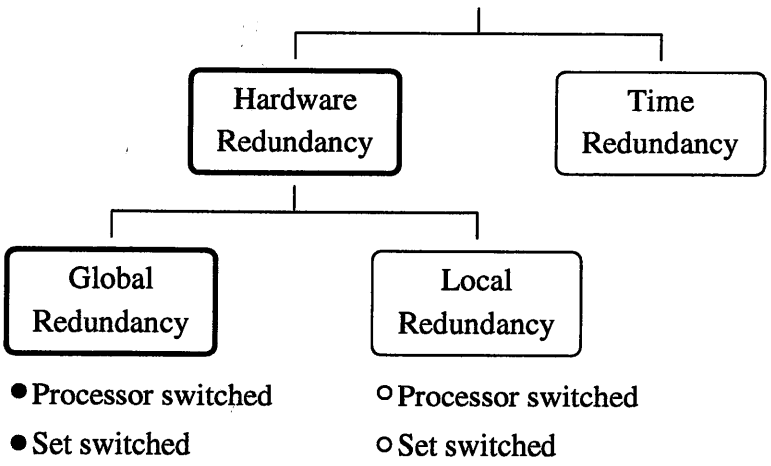


図 2.2 冗長化格子結合型アレイの分類

本章の構成は以下の通りである．第 2.2節で Kung らの提唱したモデルについて述べる．第 2.3節で Kung らの提唱したモデルの結合条件を説明する．第 2.4節で Kung らの提唱したモデルの問題点を述べる．第 2.5節で本手法で何を改善し，何が新しいかを明確にする．

2.2 $T\frac{1}{2}$トラック $-$ Rスペア型格子結合マルチプロセッサ

Kung ら [5] の最初に提唱したモデルは $1\frac{1}{2}$トラック-1 スペア型と呼ばれる．これは，図 2.3に示すように中央部に $N \times N$のアレイを配置し，周辺にそれぞれ 1 行 1 列のスペア PE を配置したものである．PE の構造自体はスペアとそうでないものに違いはない．トラックは一本であるが，後述するように PE 内部をデータをスルーさせるためのトラックが$\frac{1}{2}$だけ必要であるため，$1\frac{1}{2}$ トッラク-1 スペア型と呼ばれる．

また，Kung らは図 2.4のようなモデルを 2 トラックモデルと呼び，$1\frac{1}{2}$トラックモデルは 2 トラックモデルにマッピングできるということを示している．これにより，故障 PE がデータをスルーできるのは一見不自然であるが，$1\frac{1}{2}$トラックモデルを考えれば十分であることがわかる．

本研究ではより一般的にこれらのモデルを次のように定義する．

定義 2.1 T本のトラックに R行 R列の冗長 PE を東西南北に付加し，PE 内部をデータをスルーできるようなモデルを $T\frac{1}{2}-R$モデルと呼ぶ．また，スルーできないモデルを $T-R$ モデルと呼ぶ．

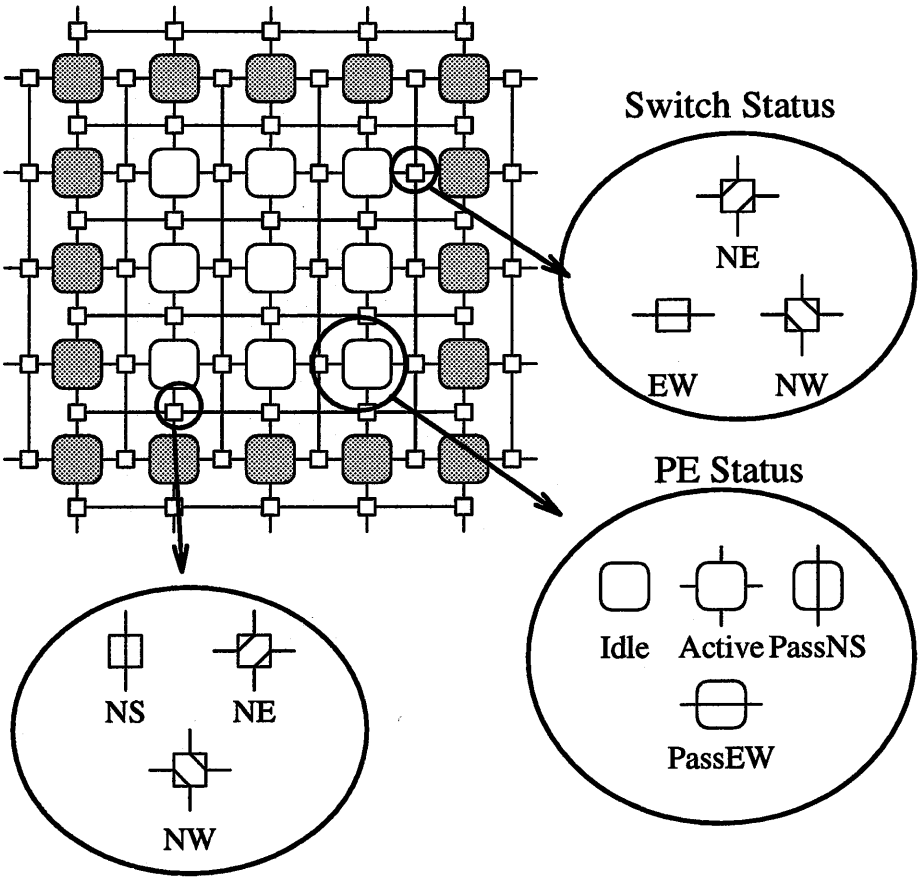


図 2.3 $1\frac{1}{2}$トラック-1 スペアモデル (N=3,R=1,T=1)

定義 2.2 PE の物理アドレスを $[p_i, p_j]$ と呼ぶことにし左上から順番に番号をふる．上方向を北 (N)，右方向を東 (E)，下方向を南 (S), 左方向を西 (W) と呼ぶことにする．

冗長 PE は東西南北に付加することから，全体のサイズは $(N+2R) \times (N+2R)$ となり，PE の物理アドレスは $[1,1]$ から $[N+2R, N+2R]$ までとなる．

2.2.1 PE の構造

PE はそれぞれ NS，EW 方向に結合ポートを持ち後述するスイッチと接続されている．PE には大きくわけて以下に示す 3 つの状態がある．

- アクティブ (**Active**)
- アイドル (**Idle**)

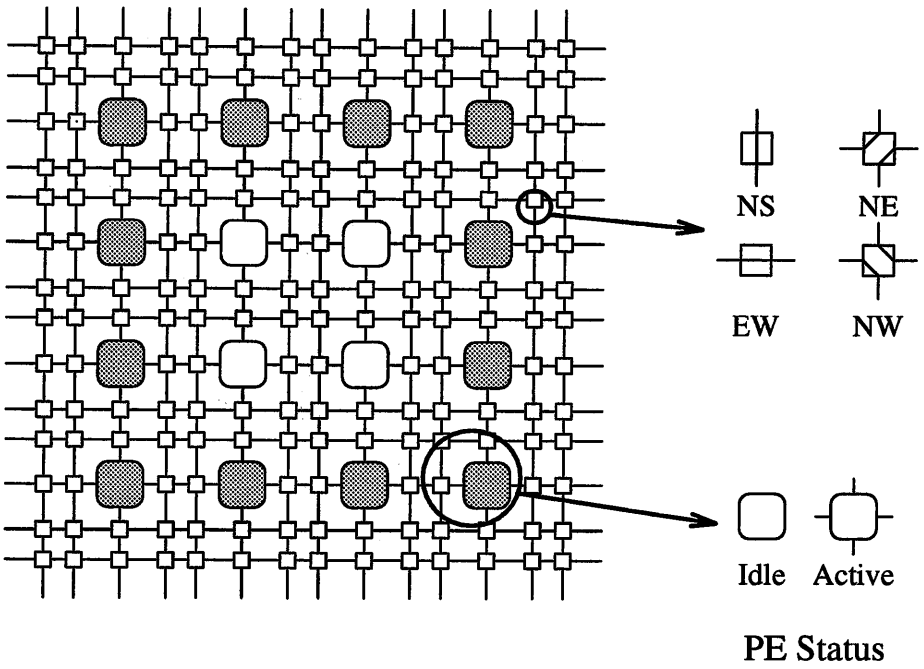


図 2.4 2 トラック-1 スペアモデル (N=2,R=1,T=2)

- バイパス (**Pass**)

ポートが全て結合され，実際に使われている PE をアクティブ (**Active**) な PE と呼ぶことにする．また逆に，全く使用されていない PE をアイドル (**Idle**) な PE と呼ぶことにする．

PE には前述したようにもう 1 つバイパス状態がある．Kung らは PE は故障があるなしにかかわらずデータをスルーできると仮定している．本研究でもこれに従う．バイパスには 2 状態ある．

- 南北方向のバイパス (**PassNS**)

- 東西方向のバイパス (**PassEW**)

図 2.6では，周辺にあるスペアの PE がそれぞれ，南北方向，東西方向のバイパス状態になっている．**PassNSEW** の状態の PE はこの例にはない．

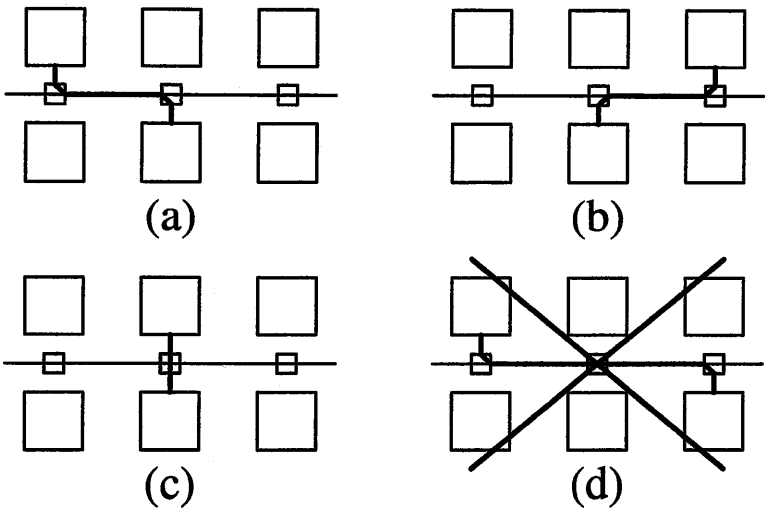


図2.5 PEとトラックの関係

2.2.2 スイッチおよびトラックの構造

$T\frac{1}{2}-R$モデルでは，スイッチはPE間のトラックにそれぞれ1つあり，トラックとPEの4つのポートを結ぶ．すなわち，T個のスイッチがそれぞれのPE間にある．スイッチのステータスはNS,EW,NE,NWの4種類ありPEの接続を切り替えることができる．ただし，NS間のスイッチではEW，EW間のスイッチではNSのステータスをとらないこととする．故にスイッチの状態はそれぞれ3通りになる．(図2.5)

トラックはPE間にT本ずつあり，さらにアレイの外周にもT本ある．$T\frac{1}{2}-R$モデルでは，南北方向と東西方向のトラックの交差点にはスイッチがない．すなわちその地点でデータの流れを切り替えることはできない．$T-R$モデルはスイッチをPE間だけでなくトラックの交差点にも置いており，交差点のスイッチはNS,EW,NE,NWの4通りのステータスを取ることができる．トラックは2つ以上のPE間の接続に占有されることはない．例えば，トラックを1本とすると，図2.6のようにスイッチを切り替え再構成を行うことができる．ただしこの図では冗長PEは東と南方向にのみ置いてある．この例では，$[1,1],[1,2],[1,3],[1,4]$のPEの北のポートのスイッチがNEに，南のポートのスイッチがNWになっており$[2,2]$の故障PEを切り替えて再構成している．このようにスイッチを切り替えPEの再構成を行う．

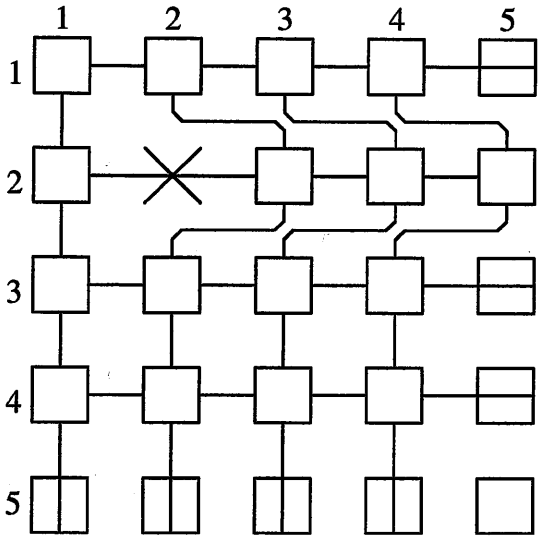


図 2.6　PE 再結合の例

2.3　PE 間の結合法則

2.3.1　物理アドレスと論理アドレス

定義 2.3 再構成システムは PE の物理アドレス $[1,1],...,[N+2R,N+2R]$ の中から $N \times N$のアレイを得ることを目標としている．この $N \times N$のアレイに論理アドレスを $(1,1),...,(N,N)$ と与える．

このように定義すると，格子結合型マルチプロセッサシステムの再構成は，論理アドレス (l_i,l_j) と物理アドレス $[p_i,p_j]$ のマッピングと考えることができる．(図 2.7)

定義 2.4 論理アドレスから物理アドレスへの写像を Φで表わす．Φ はあきらかに単射であるが全射ではない．ある論理アドレス (l_i,l_j) の PE が $[p_i,p_j]$ に実際にマッピングされていることを

$$\Phi : (l_i,l_j) \mapsto [p_i,p_j]$$

または，

$$[p_i,p_j] = \Phi(l_i,l_j)$$

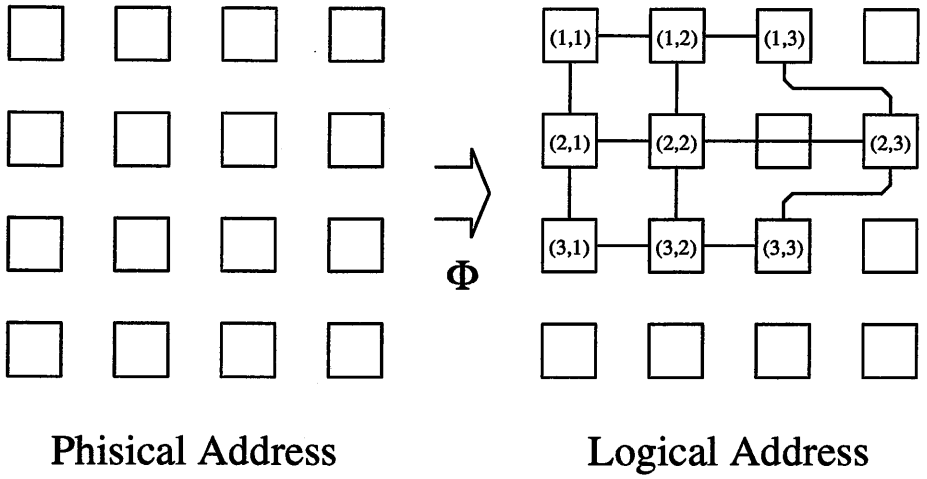


図 2.7 物理アドレスと論理アドレスの関係

と表現する．また写像Φは全単射ではないが，便宜上逆写像をΦ^{-1} で定義する．

前節で述べたように各PEには4つのポートがあり，もしそのPEのステータスが **Active** ならば全てのポートが必ず他の PE と接続している．PE のステータスが **PassNS** または **PassEW** ならば，東西または南北のポートが接続しており，残りのポートは未使用となっている．PE がアイドルならば全てのポートは使用されていない．PE の状態が **Idle** は，Φ^{-1}が存在しないことと同値である．

言い換えると，再構成問題とは,「ある条件のもとで，写像Φ を定めること」と言える．

定義 2.5 PE が矛盾なく結合されているとき「Φが正当である」と呼ぶことにする．

2.3.2 PE ポートの接続条件

補題 2.1 トラック数を 1 とし，任意の PE の物理アドレスを $[p_i, p_j]$ とする．このとき，PE の接続が正当ならば，PE の東のポートは必ず $[p_i-1, p_j+1], [p_i, p_j+1], [p_i+1, p_j+1]$ の 3 つの PE のいずれかと接続する．他のポートも同様である．

証明 PE の東のポートを考える．スイッチのステータスは EW, NE, NW のいずれかである．EW ならば $[p_i, p_j+1]$ であるから題意を満たす．もし，NE ならばその南にあるスイッチに接続される．ここで PE の接続が正当ならば，南のスイッチのステータスは NE でなければならない．故に $[p_i+1, p_j+1]$ と接続されていることになる．スイッチのステータスが NW のときも同様である．□

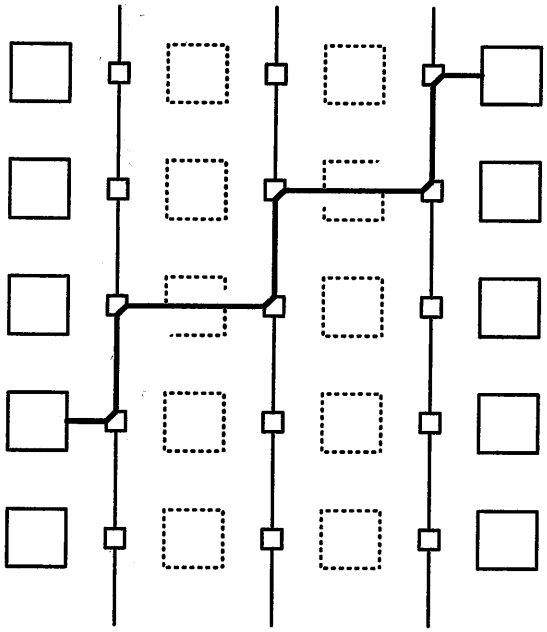


図 2.8 仮の PE

補題 2.2 トラック数Tとし，任意のPEの物理アドレスを$[p_i, p_j]$とする．このとき，PEの接続が正当ならば，PEの東のポートは必ず$[p_i-T, p_j+1], [p_i-T+1, p_j+1], ..., [p_i+T, p_j+1]$の2T+1のPEのいずれかと接続する．他のポートも同様である．

証明 図2.8のように各トラック間に仮のPEがあると考えると，補題2.1よりあきらか．□

定義 2.6 補題2.1，補題2.2より，東のポートは，$[p_i-T, p_j+1], [p_i-T+1, p_j+1], ..., [p_i+T, p_j+1]$のいずれかと接続する．この$i$成分の差に注目し，PEの4つのポートの東のポートの状態をHEと呼ぶことにする．HEは$-T, -T+1, ..., T$のいずれかの値を取る．他のポートについても同様である．とくに，物理アドレス$[p_i, p_j]$のPEというときには$\mathrm{HE}[p_i, p_j]$と書く．論理アドレス(l_i, l_j)のPEというときには$\mathrm{HE}(l_i, l_j)$と書く．

補題 2.3 トラック数1とする．このとき，$P_1[p_i, p_j]$の東のポートが使用されており，かつ$P_2[p_i+1, p_j]$の東のポートが使用されているとする．このとき接続が正当ならば，次のことが成立する．

$$\mathrm{HE}[p_i, p_j] \leq \mathrm{HE}[p_i+1, p_j]$$

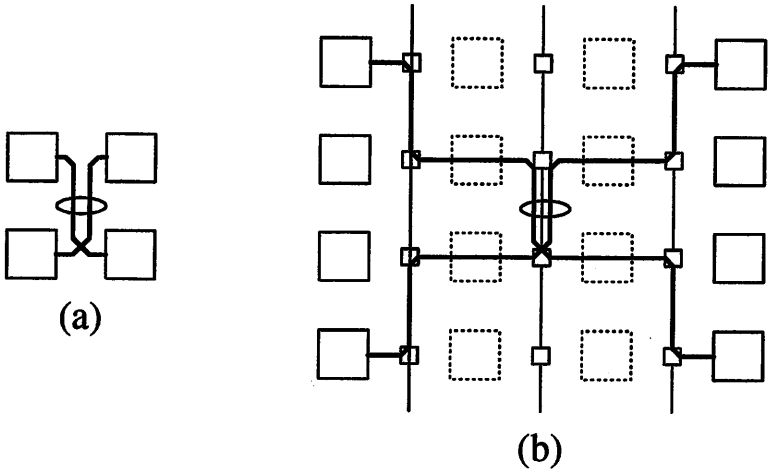


図 2.9 トラックの条件

証明 $\mathrm{HE}[p_i, p_j] > \mathrm{HE}[p_i + i, p_j]$ ならば，図 2.9(a) のようにトラックを共有せねばならない．故に矛盾する．□

補題 2.4 トラック数が T とする．$P_1[p_i, p_j]$ の東のポートが使用されており，かつ $P_2[p_i + 1, p_j]$ の東のポートが使用されているとする．このとき接続が正当ならば，次のことが成立する．

$$
\begin{aligned}
\mathrm{HE}[p_i, p_j] &\leq \mathrm{HE}[p_i + 1, p_j] \\
\mathrm{HE}[p_i, p_j] &\leq \mathrm{HE}[p_i + 2, p_j] + 1 \\
\mathrm{HE}[p_i, p_j] &\leq \mathrm{HE}[p_i + 3, p_j] + 2 \\
&\vdots
\end{aligned}
$$

証明 図 2.9(b) のように仮の PE があると考えれば補題 2.3 よりあきらかである．□

HE, HW, HN, HS とスイッチは密接な関係にある．例えば，HE が決まると，その東のポートのスイッチは表 2.1 のようになる．

2.4 従来の手法およびその問題点

前節までで，本研究で用いる Kung のモデルの PE 間の結合の条件を述べた．本節では，Kung らの提案した補償パス法について説明し，その問題点を指摘する．

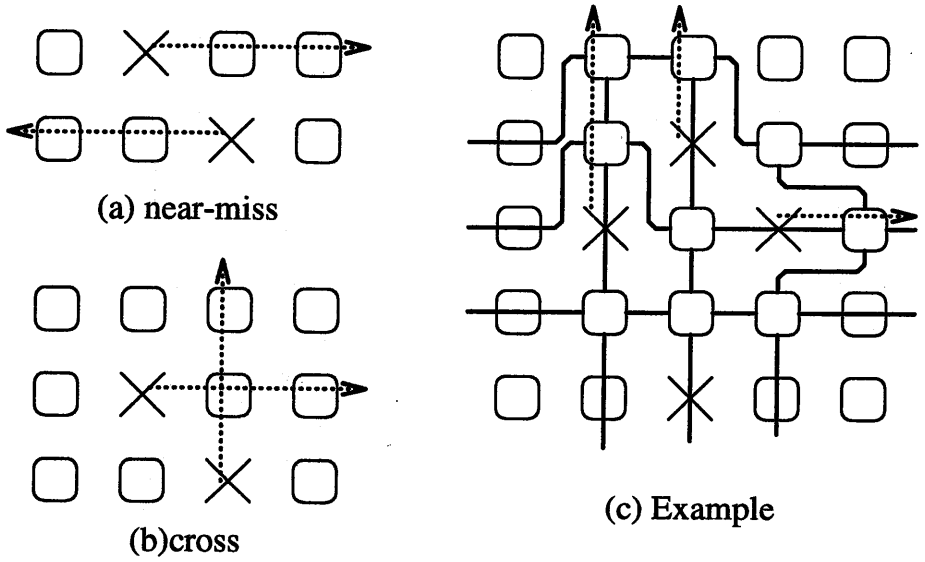


図 2.10　補償パス

2.4.1　補償パス法

S.Y.Kung らは，この問題を解決するために補償パスというアイデアを提唱している．補償パスとは次のように定義される．

定義 2.7 今，$[i,j]$ の PE が故障しているとする，その故障を $[i',j']$ が補償し，$[i',j']$ の故障を $[i'',j'']$ が補償し... となってる場合これを補償パスと呼ぶことにする．

簡単な考察により以下の条件が成立すると $1\frac{1}{2}$ モデルでは故障を救済し再構成することが可能であることがわかる．

1) 補償パスは南北および東西に沿った直線である．

2) 図 2.10(a) に示すように行または列で逆向きの補償パスが隣接しない．これを near-miss と呼ぶ．

3) 図 2.10(b) に示すように補償パスが交差しない．

Kung[5] らはこれらをグラフの組み合せ問題に帰着させ，指数時間で解くアルゴリズムを提唱した．その後 V. P. Roychowdhury[10] らが，多項式時間で解くアルゴリズムを提

唱している．また，J. S. N. Jean[4] らは $2\frac{1}{2}-2$ モデルについて指数時間で解くアルゴリズムを提唱しており，後に一般的な $m\frac{1}{2}-m$ モデルについて多項式時間で解くアルゴリズムを T. Varvarigou[12] らが提唱した．

また，T. Varvarigou[13] らは3トラックモデルに拡張すると条件1の制限がなくなり，より簡単なアルゴリズムが存在することも後に示している．また，高浪 [18][22] らはニューロを用いて組み合せ問題を近似的に解く解法を提案している．

2.4.2 補償パス法の問題点

補償パス法には以下に示すような問題点がある．

- 4方向への補償しか考慮していない．
- 冗長 PE の位置が周辺に1行1列に限られる．
- 十分に PE 資源を活用していない．
- グローバル情報を必要とする．

補償パス法は4方向へのシフトしか考慮していない．故に四隅の PE を使うことができない．例えば図 2.11(a) では NE 方向に補償し，再構成を行っているが，このようなパターンを発見することはない．

また，冗長な PE が周辺に限られ1行1列しか許されない．たとえば，2行2列付加すると，図 2.11(b) のようなパターンも許されることになるが，このような場合は全く別のアルゴリズムを考えなければならない．

最後に，補償パス法は自明な解が存在する場合でも，再構成不能となる可能性がある．例えば図 2.11(c) の中央の PE にはその上下左右全ての方向に故障 PE が存在し，補償パスが存在しないため Kung らの手法では補償することができない．しかし，実際には図 2.11(d) のように再構成することにより，故障を救済することが可能である．すなわち，補償パス法は十分に PE 資源を活用していない．これは現在までに提案された補償パスを用いる再構成法で解を探す方法が故障 PE のみに注目しているために起きる問題である．例えば図 2.11(b) や図 2.11(d) では非故障 PE をも補償することにより，再構成を可能にしている．

一般に故障していない PE を含めて故障救済を考える問題は一見簡単そうだが極めて困難である．例えば図 2.11(d) の中央の PE は前述するアルゴリズムでは N, S, W の方向に

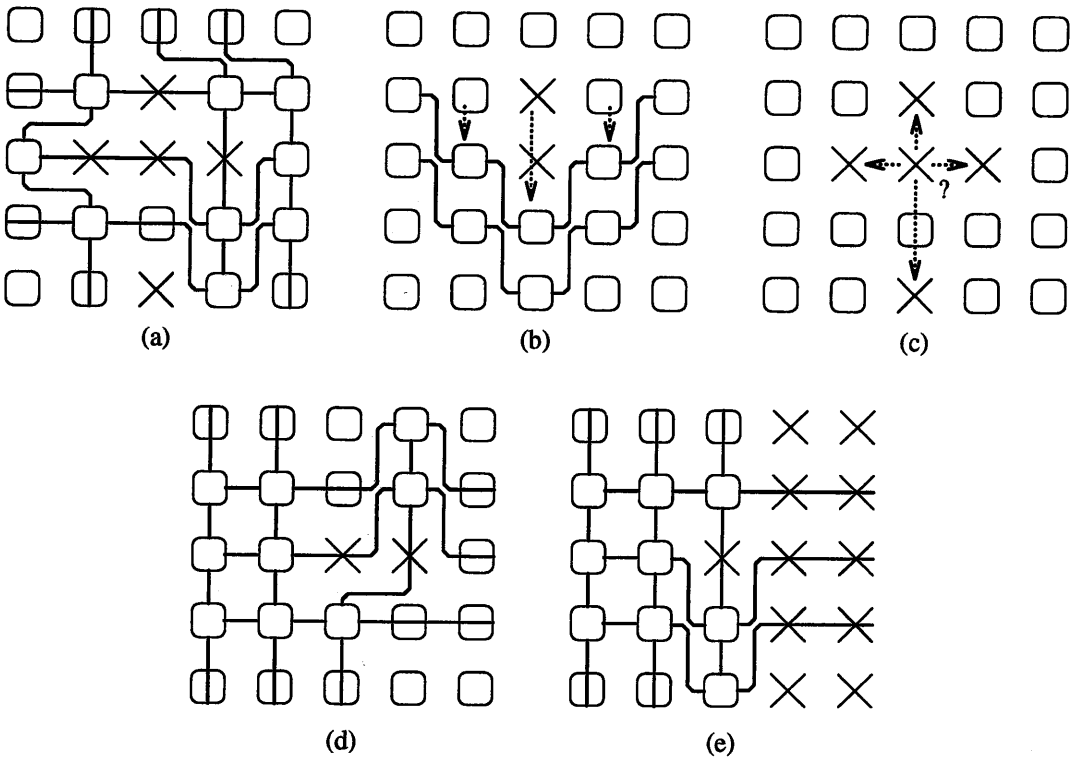


図 2.11　補償パス法で解決できない典型的な故障パターン

補償可能であるが，E 方向には補償できない．しかし実際には図のように NE 方向に補償しても再構成が可能である．また，図 2.11(e) の中央の PE のように W 方向に故障 PE がないにもかかわらず，W 方向に補償してしまうと再構成できないこともある．このように一般に故障 PE 周辺の分布だけを見て PE の補償の方向の可能性の組合せを考えてゆくことはできず，アレイのサイズが大きくなればなるほどこの問題へのアプローチは極めて困難になる．

補償パス法は，全故障情報 (すなわちグローバル情報) を用いて故障 PE を回避するようにスイッチを切り替えねばならない．すなわち全故障の状態を知っているスーパーバイザの存在が不可欠である．しかし，10^6程度からなる超並列マシンを構成することを考えると，全 PE の故障箇所を知るのは極めて困難となる．また，第 1 章で述べたように WSI に構築することを前提とすると，ウェーハのシステムが高密度になるにつれ，外部から故障箇所を検出したり，スイッチの切り替えを操作するのが困難となる．

以上の点が補償パス法に一般に言える欠点である．

2.5 大規模システムに必要とされる再構成法

大規模システムの再構成技術に要求されるものとして以下に示すようなものが挙げられる.

- より少ない情報量.
- 自律再構成.
- より少ないハードウェア量.
- 高い再構成率.

これらのいくつかは互いにトレードオフの関係にあり，全てを満足することは難しい.例えば，次元が$(N+R)^2$の完全結合ネットワークからN^2の格子結合ネットワークを得る手法は自明であるが，リンクの数が増えハードウェア量が激増する．$1\frac{1}{2}$モデルも同様で，Varvarigou ら [13] らは3トラックモデルに拡張し，Kung らの提案した補償パス法の欠点の1つである補償の方向や制限を取り払い，ネットワークのコスト最小経路問題に帰着させ簡単により高い再構成率を得る方法を提案している.

ローカル情報のみで行う手法とは，各PEが自分を含めた近隣のPEの情報のみ用いて，近隣PEとの情報交換によって再構成を行う方法で，大規模アレイに特に有効である．また，WSIのような超高密度システムでは，故障の情報を検査し外部に取り出すのが困難であるため，ローカルな情報で各PEが独立し自律再構成できるのが望ましい.

ローカルな情報を用いた再構成法に関して，例えば，R.Negrini らは Ack と Req の信号のやりとりにより再構成を行うアーキテクチャを提案している [9]．このシステムは本節で説明した Kung らのアーキテクチャと違いポートの結合する範囲を行方向に3，列方向に5としている．このため，PE間の結合のパターンが多く，スイッチやトラックに多くの回路を必要とする.

大規模システムでは集積度も増加するため，付加する冗長回路などはできるだけ少ない方がよい．Kung らの提案したモデルは冗長化回路も単純で，大規模システムに向いていると考えられる．そこで本研究でも Kung らのモデルと同様のモデルを採用することにする．しかし前節で述べたように数個の故障ですら解を発見できないと言う欠点がある.この問題は補償パスを用いるかぎり解決できない．そこで本研究では以下に示す特徴をもつ，新たな再帰的シフトと呼ばれる手法を提言する.

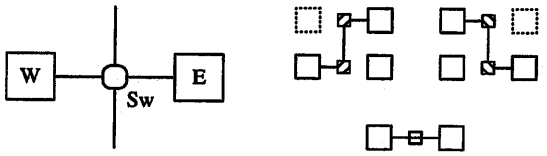
- Kungと同等のアーキテクチャで再構成可能.
- 4方向への補償以外の補償も考える.
- 冗長PEの位置や数に制限を加えない.
- グローバル情報を必要としない.(ローカルな情報だけで自律再構成可能である)
- 3次元への拡張も可能.

これらを,次章以降で詳しく述べる.

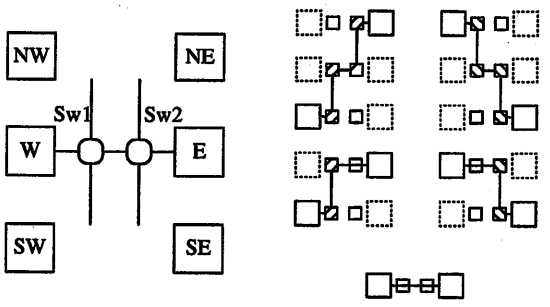
2.6 むすび

本章では,まず,格子結合ネットワークの種類と再構成法の分類を述べた.また本研究でターゲットとするKungらの提唱した冗長化格子結合ネットワークについて述べた.その上で,物理アドレスおよび論理アドレスを定義し,いくつかの拘束条件を述べ,説明を行なった.

また,Kungらが提唱した従来法である補償パスの戦略を示し,その欠点および問題点をあきらかにした.次章以降では,これらの問題点を克服するために本研究で提言する再帰的シフトを用いた再構成手法を説明する.



T=1



T=2

$T = 1$

PE status (HE)	
PE_W	Switch
-1	NW
1	NE
0	EW

PE status (HW)	
PE_E	Switch
-1	NE
1	NW
0	EW

$T = 2$

PE status (HE)			
PE_W	PE_{SW}	PE_{NW}	Switch1
$-2_{or}-1$	-	-	NW
0_{or} 1	-	-	EW
2	-	-	NE
Idle	$-2_{or}-1$	-	NW
Idle	-	2	NE

PE status (HW)			
PE_E	PE_{SE}	PE_{NE}	Switch2
-2	-	-	NE
-1_{or} 0	-	-	EW
1_{or} 2	-	-	NW
Idle	-2	-	NW
Idle	-	1_{or} 2	NW

表 2.1　PE のステータスとスイッチステータスとの関係

第 3 章

格子結合型マルチプロセッサの静的自律再構成法

3.1 まえがき

従来格子結合型マルチプロセッサの再構成法は，補償パスの組合わせ問題をいかに解くかを中心に議論されてきた．しかし，補償パスを用いる従来手法は前章で述べたように次に示すような問題がある．

- 再構成可能な故障パターンでも再構成できないことがある．
- 冗長 PE の数が変化した場合に対応できない．
- グローバルな情報を必要とするため WSI に向いていない．

そこで本章では，これらの問題を解決するため，バイパスおよび再帰的シフトという手続きを提唱する．バイパスは格子結合マルチプロセッサの行または列をすべて切り離す手続きである．シフトはある 1 つの PE を任意の方向 (4 方向) の適当な PE に補償する手続きであり，それぞれ以下のような特徴を持つ．

- 冗長 PE の数に依存しない．
- グローバルな情報を必要としない

冗長 PE の数に依存しないというのは，アレイサイズやトラック数と付加する冗長 PE の数が依存しないということで，格子結合ネットワークを構築する際の自由度が増すことを意味する．

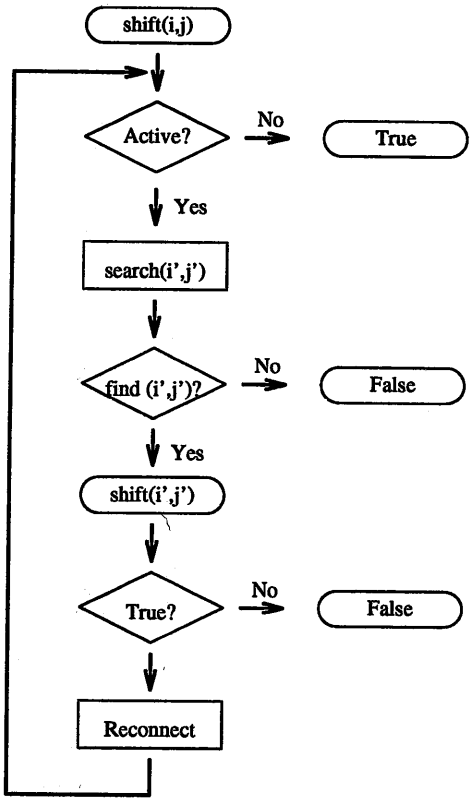


図 3.1　再帰的シフトの概略

本章の構成は以下の通りである．第 3.2節で再帰的シフトの概略を説明する．第 3.3節では本研究で採用した再帰的シフトを用いた自律再構成アルゴリズムの概略を説明する．第 3.4節では，再構成アルゴリズムの 1 つである FS 法を説明し，性能評価を行う．第 3.5節では，同じく再構成アルゴリズムの 1 つである BS 法を説明し，性能評価を行う．

3.2　再帰的シフト

そこで，本研究では再帰的シフトという手法を新たに提言する．再帰的シフトの概略を図 3.1に示す．すなわち，

1) $[i,j]$ の PE が故障していてかつ **Active** とする．この PE はネットワークから切り離されねばならない．そこで，その PE の代りを $[i',j']$ に代行させようとする．この $[i',j']$ が発見できなければ **False** を返しこのアルゴリズムは終了する．

2) $[i',j']$ の PE が **Active** でなければ $[i,j]$ の PE の代行を行うことができる．そこで，

procedure **check PE**

```
begin
  count = 0;
  while( count≤10000 )
  begin
    for i=0 to N+2R do
    for j=0 to N+2R do
    begin
      if status=Active and Fault then
      begin
        select_one until
            ret=Ture or all is False
        begin
          ret=shift-east(i,j);
          ret=shift-south(i,j);
          ret=shift-west(i,j);
          ret=shift-north(i,j);
        end
        if ret=False then
          return False;
      end
    end
    flg=0;
    for i=0 to N+R do
    for j=0 to N+R do
    begin
      if status=Active and Fault then
      flg=1;
    end
    if flg=0 then
      return True;
    count=count+1;
  end
  return False;
end
```

図 3.2　自律再構成アルゴリズム

True を戻り値として返す．そうでなければ，$[i', j']$ は代行できない．そこで，$[i'', j'']$ に代行を要求する．(この呼び出しは再帰的に行われる)

3) $[i', j']$ から **True** を受けとった PE$[i, j]$ は，ポートの結合を変化させ $[i', j']$ につなぎ換え，自分自身を **Active** 以外の状態にする．

本研究では，この $[i', j']$ は東西南北に隣接した 4 つの PE のうちの 1 つとする．また，シフトはシグナルによって行なわれる．すなわちシグナルを受けた PE が発火し，その状態により必要があれば隣接した PE を発火させ，自分自身や近隣の PE の状態やポートの状態を変化させる．

補償パスでは矛盾のない組み合せを完全に調べるパスを探索する．それは周辺まで直進する一本のパスであった．しかし，本手法での再帰的シフトは隣りの PE まで矛盾しないパスを探索するだけである．すなわち，隣りの PE より遠方がどうなっているかは関知せず，各 PE は隣りの PE が単に自分の代行をできるかどうかを知っていればよい．

また，上述の理由により本研究で提言した再帰的シフトやバイパスはローカルな情報の

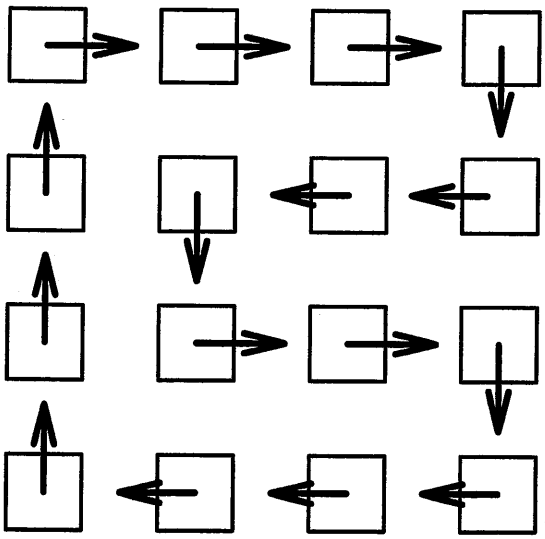


図 3.3　トークンの回覧

みで再構成が可能である．

しかし，補償パス法で説明した通り，基本的にパスの方向は 4 方向ある．再帰的シフトでも同様で $[i', j']$ の PE の候補は 4 つある．残念ながら現在まで，この方向を決定づける明確な方法は発見されていない．そこで，この方向だけを未決定のパラメータとし，様々な方法でシミュレーションを行なった．

3.3　自律再構成アルゴリズム

再帰的シフトを用いる自律再構成アルゴリズムの戦略は次のように書ける．

1) 初期状態として仮配置を行う．

2) 全ての PE を順番にチェックする．

3) PE が **Active** でかつ故障しているならば方向を決定し，再帰的シフトを実行する．

4) 全ての PE の故障が補償されたならば終了する．そうでなければ 2) へ戻る．

複数の PE が同時にシフト手続きを行うとデッドロックに陥いる可能性があるが，これは PE に順番にトークンを渡すことによって回避することができる．このアルゴリズムを

図 3.2に示す．*select_one* と書いてある文は以下の 4 つの文のうち 1 つをある手法で選び実行することを意味する．

これらのアルゴリズムには二つの二重 for 文があるが，この部分をトークンを回覧させながら実行することにより，ローカルな情報だけでアルゴリズムを実行することができる．すなわち，トークンを持っている PE だけがシフト手続きを実行させることができるようにすればよい．トークンの回覧の順番に規則はないので，図 3.3のように回覧させれば効率がよい．

3) のステップにより既に補償されていた PE がまた **Active** になるかもしれない．各 PE は他の PE ががどうなっているかは関知せず自分自身の故障を補償することしか考えていないからである．最初，1 回のシフトにより自分自身は回避されるが，ポートのつけ換えの副作用で他の今まで回避されていたのが再び結合されてしまう可能性がある．このため，全 PE が回避されるまで何度もループを実行する．これで問題は 3) のステップにおける最初の方向を決定づけるところだけとなった．しかし，前章で述べたように補償の方向を決定するのは極めて難しく，有効なアルゴリズムは現在まで提唱されていない．そこで，本研究ではバイパスおよび再帰的シフトをを基本とした自律再構成アルゴリズムとして，以下の 3 つの手法を提言する．

- FS(Four way Shift) 法
- BS(Bypass and Shift) 法
- HS(Heuristic Shift) 法

最初の FS 法は，PE の位置に依存させてシフトの方向を決定する手法であり，次の BS 法は，はじめにバイパスを行いそれによりシフトの方向を一方向に固定する手法である．このためこれらの手法を静的再構成法と呼ぶことにする．最後の HS 法は，シフトの方向が位置に依存しない．このため動的再構成法と呼ばれる．これについては第 4 章で説明する．

3.4 FS(Four way Shift) 法

3.4.1 FS 法の戦略

FS(Four-way Shift) 法とは PE の位置によりシフトの方向を決定するアルゴリズムである [16]．FS 法の戦略は次の通りである．はじめに初期状態として図 3.4のように中央部分

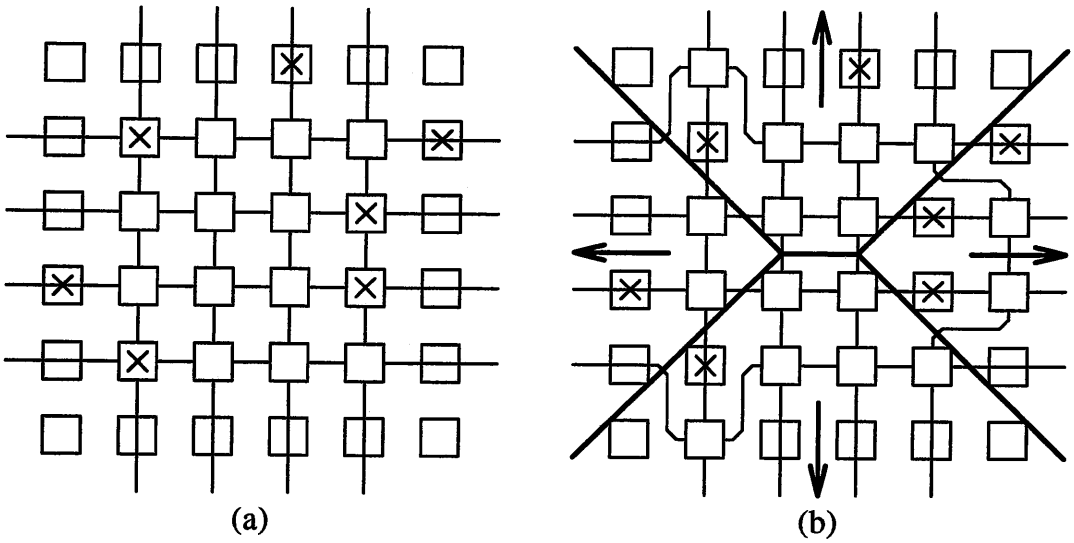


図 3.4　FS(Fourway Shift) 法

に仮の配置をする．シフトの方向は中心から見て北方向は北へ，東方向は東へというように，故障パターンの位置により決定することにする．すなわち PE 自身の物理アドレスからもっとも近い周辺方向へ常にシフトする方法である．故に北または南にシフトする PE は状態として，**PassNS** と，**Active** しかとらない．同様にして，東または西にシフトする PE は **PassEW** と，**Active** の状態にはならない．また，四隅の PE も使用できない．

FS 法の特徴はその単純さにある．シフトの方向が場所により決定されているため，冗長 PE が一行一列であれば，基本的には補償パス法以上の値をとることはない．しかし，本手法はローカルな情報だけで可能であり，かつ冗長 PE の数がいくつあっても構わないと言う特徴をもつ．その点において補償パス法と異なる．

3.4.2　FS 法実現のための PE 間シグナルによる再帰的シフト

FS 法を実現するための再帰的シフトのアルゴリズムを説明する．その例として，東方向へシフトする場合について説明する．注目している PE を図と対応させて PE(*) のように表現する．

step1)

PE が **Idle** あるいは，**PassEW** 状態であるときはシフトは不要である．よって直ちに **True** を返し終了する．それ以外の状態で，これ以上東にシフト不可能な場合は **False** を

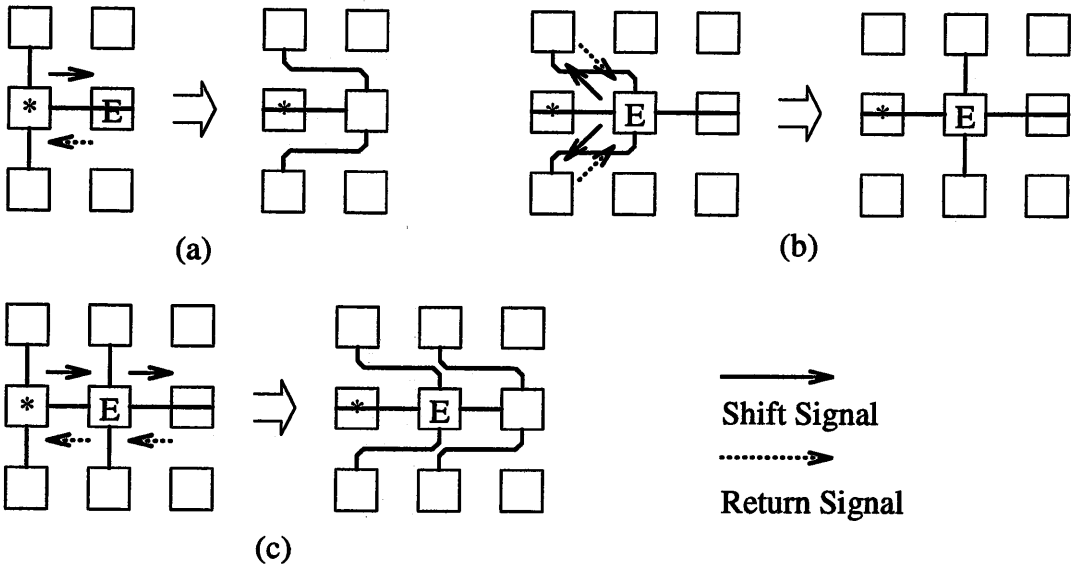


図 3.5　FS 法実現のための再帰的シフト

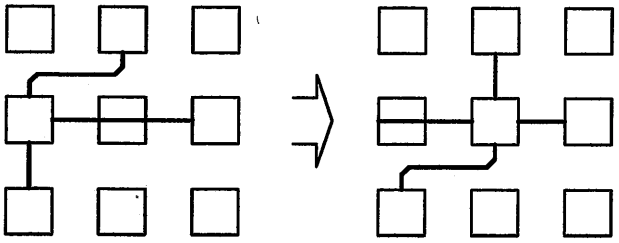


図 3.6　FS 法実現のための再結合

返し終了する.

step2)

HN および HS をそれぞれチェックする. 例えばトラックが 1 のとき, HN $= -1$ であればトラック数が 1 であるため東方向へシフトすることができない. そこでこの場合 PE(NW) に東方向へのシフトシグナルを送る. (図 3.5(b)) トラック数が T ならば, HN $= -T$ のときにシグナルを送る.

step3)

PE(E) に東へのシフトシグナルを送る. **True** が戻り値として確認されると PE(E) は **Idle** かまたは **PassEW** 状態である.

```
procedure shift_to_east(pe)
begin
    if status = Idle or status = PassEW then
        return True;
    if nomore shift to east then
        return False;
    if HN = -T then
        shift_to_east(pe(N));
    else if HS = -T then
        shift_to_east(pe(S));
    shift_to_east(pe(E));
    reconnect_east(pe);
end
```

図 3.7　再帰的シフトのアルゴリズム

```
procedure reconnect_east(pe)
begin
    HS(pe(N)) = HS(pe(N)) + 1;
    HN(pe(S)) = HN(pe(S)) + 1;
    HN(pe(E)) = HN - 1;
    HS(pe(E)) = HS - 1;
    status(pe(E))= Active;
    status = Idle;
end
```

図 3.8　再結合のアルゴリズム

step4)

再結合できるパターンになっていたならポートを繋ぎ換える (再構成). 再結合できるパターンとは，HE が 0 でかつ，東の PE が **Active** かまたは **PassEW** でかつ，HN および HS が$-T+1, -T+2, ..., T-1, T$のいずれかになっていることである．今ポートをつなぎ換えようとしている PE を (L_i, L_j) とすると，$\mathrm{HN}(L_i, L_j+1) = \mathrm{HN}(L_i, L_j) - 1$ とし，$\mathrm{HN}(L_i - 1, L_j)$ の HS を 1 増せばよい．HS も同様である．その後 PE のステータスを **Idle** または **PassEW** にする．

この手続きは正当な接続を保存する．なぜなら，*step2* により HNおよび HSが$-T+1, -T+2, ..., T-1, T$のいずれかになっていることが期待される．*step3* により，東の PE が **Active** かまたは **PassEW** になっていることが期待される．これらを全てクリアした

場合に上のポートを繋ぎ換える手続きを行っても，接続はあきらかに正当である．(図 3.6) 最初の接続が正当であるから，この手続きを何度繰り返しても正当さが崩れることはない．

これらの手法では，各 PE はシステム全体を再構成し補償させようとするのではなく，自分だけを補償しようとする．他の PE が既に補償している PE にぶつかった場合，その PE の補償を無効にしてでも補償しようとする．これを繰り返すことにより，格子結合アレイ中の欠陥 PE を回避できアレイ再構成が実現される．このアルゴリズムを図 3.7および図 3.8 に示す．また，この図では shift を再帰的に呼び出したあとの戻り値のチェックを省いている．本来ならば **False** が返ってきた時点で手続きを直ちに中断する．

3.4.3 FS 法の性能評価

再構成アーキテクチャの性能の指標として様々なものが考えられるが，ここでは格子結合型マルチプロセッサシステムのウェーハ上への実現性を考えてシステムの歩留り (Yield) を中心に性能評価を行なう．本論文では，アレイ歩留りはアレイが構成できる確率として定義し，PE 歩留りは PE が欠陥なく生産される確率として定義している．

再構成アーキテクチャにおいて次の仮定のもとで歩留りの性能評価を行なう．

- 故障を検知する回路および回避を制御する回路は故障しない．
- トラックおよびスイッチは故障しない．
- PE は一定確率で故障する．

この仮定はその他の多くの研究 [5][4][15] でも行われており，再構成アーキテクチャの性能を評価するのには都合がよい．まず，全ての PE 毎にコイン投げを行い，ある確率で故障させる．次にその PE 故障パターンで再構成が成功するかどうかを調べる．本研究では各 100 回の試行を行い，うち再構成ができる回数をカウントし歩留りとした．

図 3.9に 10×10 のアレイの冗長 PE 数 Rを変化させた場合の歩留りを示す．FS 方は $R = 2$ において Kung らの手法とほぼ同じ結果を出すことがわかった．Kung らは $R = 1$ の結果を出していることを考えると，FS 法は Kung の手法に及ばないことがわかる．これから次のような結論が導かれる．

- シフトを能率的に行うには，もっとも近い外周方向へシフトするだけでは不充分である．すなわち，遠い外周方向へのシフトが重要である．

以上のことを考え，次節において新たな手法を提案する．

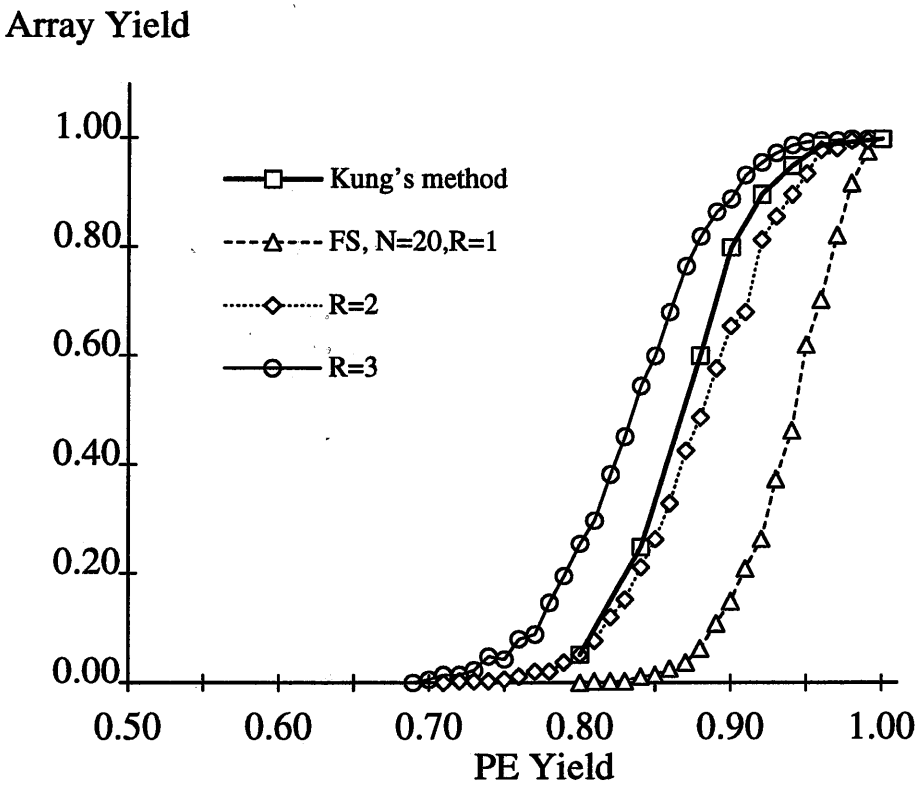


図 3.9 $N = 10, T = 1$ における FS 法の歩留り

3.5 BS(Bypass and Shift) 法

3.5.1 BS 法の戦略

BS(Bypass and Shift) 法とはシフトとバイパスを組み合わせたアルゴリズムである．BS 法の戦略は次の通りである．はじめに，$N + 2R$ の列から $2R$列をバイパスする．これにより，アレイは $(N + 2R) \times N$ となる．次に，図 3.10のように各 PE をもっとも北側を迂回するように仮の結合をする．その後，仮の PE の結合をチェックし，故障している場合は南方向に一段づつシフトする方法である．最初にもっとも北側を迂回するように結合されているので，南方向のシフトだけを考えてやればよいことがわかる．すなわち，BS 法では最初にバイパスすることにより，シフトの方向が 4 方向から 1 方向に定まる．

BS 法では列方向にバイパスを取り入れているため，行方向のスイッチの構造が単純になる．これについては第 5 章で説明する．

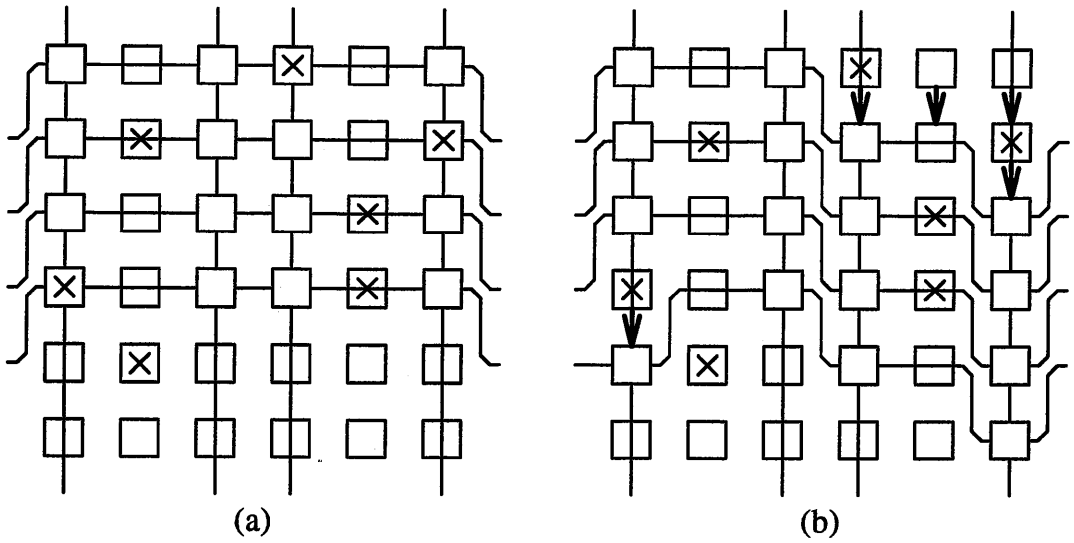


図 3.10　BS(Bypass and Shift) 法

3.5.2　PE 間シグナルによるバイパス

バイパスアルゴリズムは単に 1 列または 1 行の PE を全て切り離す方法である．このアルゴリズムの利点として，故障が比較的集中して発生した際に効果を発揮することが挙げられる．しかし，故障が同一行，あるいは同一列において少ない場合も 1 行あるいは 1 列をバイパスしてしまい，PE の利用効率が悪くなる可能性がある．例えば Kuo と Fuch[6] は，行および列における PE 故障数を調べ，グラフ理論による組合せを解析することにより最適なバイパス行および列を選択している．しかしこの方法では計算量が多いことやグローバル情報を必要とするためバイパスの単純さの効果を失う．そこで，ここでは単に欠陥のある PE の数の多い行 (列) から順番にバイパスする手法を用いる．

このアルゴリズムは次に示す 6 つのステップにより実現できる．

1) $[1, p_j](p_j = 1, 2, ..., N + 2R)$ の PE は自分が故障しているならば 0 をそうでなければ 1 を $[2, j]$ の PE へ送る．

2) $[p_i, p_j](p_j = 1, 2, ..., N + 2R)$ の PE は故障しているならば 送られてきたデータに 1 をプラスして $[p_i + 1, p_j]$ の PE の送る．

このアルゴリズムにより $[N + 2R, p_j](p_j = 1, 2, ..., N + 2R)$ の PE (最下段の PE) はその列にある全欠陥 PE の数 n_j を知ることができる．(図 3.11)

次に

3) $[N+2R,1]$ の PE は n_1 を $[N+2R,2]$ へ送る.

4) $[N+2R,p_j]$ の PE は 送られてきたデータと n_j を比較し大きい方を $[N+2R,p_j+1]$ の PE に 送る.

このアルゴリズムにより $[N+2R,N+2R]$ の PE(最下段, 最右の PE) は $n_{\max} = \max(n_1,n_2,...)$ を知ることができる.

次に

5) $[N+2R,N+2R]$ の PE は $n_{\max}$ と n_{N+2R} とを比較し等しければその列をバイパスし $n_j=-1$ として $[N+2R,N+2R-1]$ の PE へ 0 を送る. そうでなければ $n_{\max}$ を送る.

6) $[N+2R,p_j]$ の PE は 送られてきたデータと n_j とを比較し等しければその列をバイパスし $n_j=-1$ として $[N+2R,p_j-1]$ の PE へ 0 を送る. そうでなければ送られてきたデータをそのまま送る.

この最後のアルゴリズムで $[N+2R,1]$ の PE がデータを受け取ったところで一回目のバイパスが終了する (図 3.11). 複数列のバイパスを行いたい時は, 必要なだけ最後の 3,4,5,6 のアルゴリズムを実行する. これらの処理はその流れの方向などから, 現在どのフェーズにあるのかは自明である. 故にバイパスシグナルおよび, 隣接 PE に送るパラメータのみで行うことができる. この様子を図 3.11に示す.

3.5.3 アルゴリズム停止性の証明

BS 法や FS 法のアルゴリズムが停止することを証明する. これらのアルゴリズムをΨと呼ぶことにする.

補題 3.1 アルゴリズムΨにおける状態遷移は一方通行である.

証明 アルゴリズムΨで, ポートの状態 (すなわちΦ) の変化は, 再結合アルゴリズムの中でのみ行われる. この置き換えは, 必ず特定方向へアドレスがシフトしている. したがって, 状態遷移は一方通行で決して戻ることはない. □

補題 3.2 アルゴリズムΨにおいて, ポートが交差することはない.

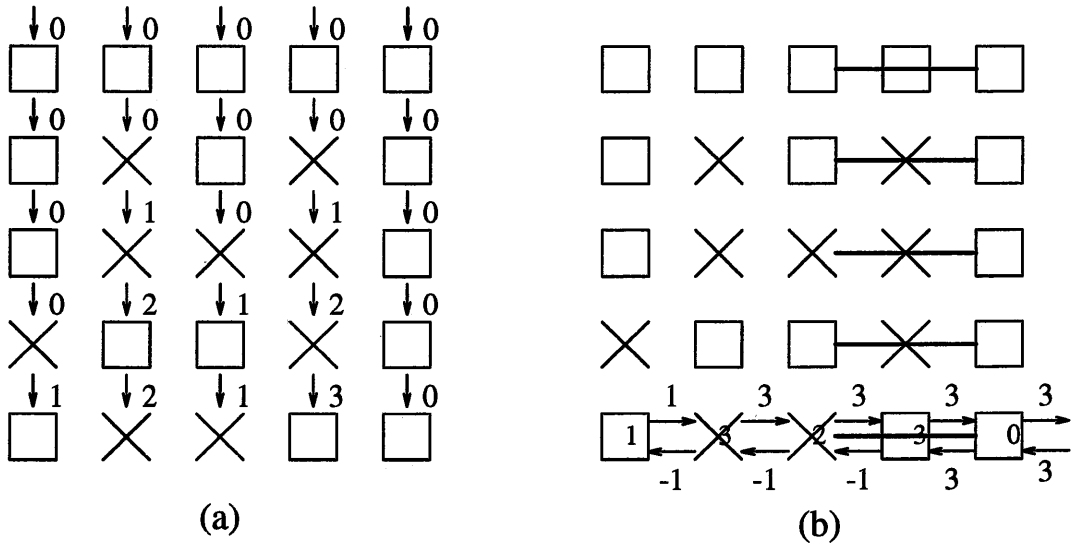


図 3.11　バイパス

証明

回避アルゴリズムが **True** を返すのは，その PE が **Idle** または **PassNS** のときだけである．したがって，再結合シグナルの発生時点で，必ず南の PE は **Idle** または **PassNS** で東西のポートは未使用である．ポートの状態の変化はポートの結合していない $[p_i+1, p_j]$ と，$\Phi(l_i, l_j \pm 1)$ のポートを接続させるだけであるから，途中で交差の要素が入ることはない．□

補題 3.3 回避アルゴリズムは停止する．

証明　今，$[p_i, p_j]$ がシグナルを受けたとする．もし，ポートの状態により再度 $[p_i, p_j]$ がシグナルを受ければループとなり回避アルゴリズムは停止しない．

回避アルゴリズムは物理アドレスで見ると自分より北に位置する PE にシグナルさせる可能性があるが，論理アドレスで考えると自分より北に位置する PE をシグナルを送るにはポートがクロスして接続されている必要がある．これは補題 3.2 により否定されている．また，シグナルの方向は一方通行であるので同じ道を戻って来ることはない．□

定理 3.1 アルゴリズム Ψ は停止する．

証明　補題 3.3 より回避アルゴリズムは必ず停止し，また補題 3.1 より状態は常に南方向へと変化する．アレイのサイズが有限であるので Ψ は必ず停止する．□

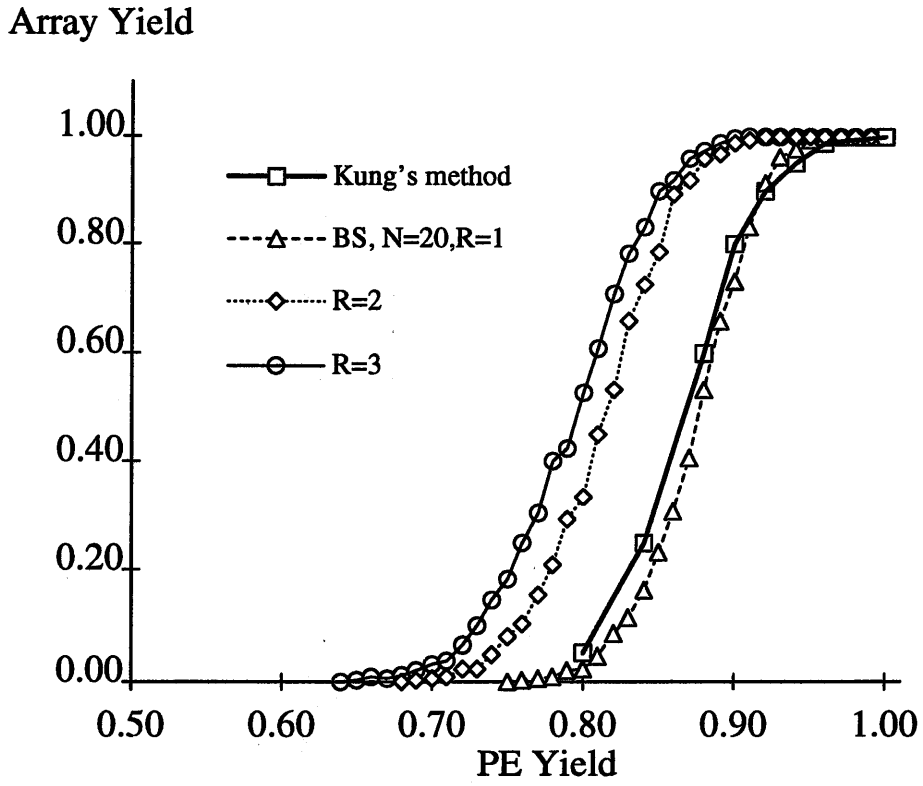


図 3.12　$N = 10, T = 1$ における BS 法の歩留り

3.5.4　BS 法の性能評価

図 3.12に 10×10 のアレイにおいて冗長 PE を変化させた場合の歩留りを示す．BS 法は $R = 1$ において Kung らの手法とほぼ同様の結果を得ることができることがわかる．このように BS 法は，ローカルな情報のみを用いて Kung と同程度の自律再構成が可能であることがわかった．

また，図 3.12はある程度以上冗長な行および列を付加してもそれに比例して歩留りが改善されないことを示している．この理由としては以下のようなことが考えられる

- 冗長な PE を増やすことにより故障する PE も増える．PE が増えれば当然故障数の増加にもつながる．
- 必要とするアレイサイズに比較して数多くの冗長な PE を付加しても，トラック数が 1 本または 2 本と少ないため冗長 PE を利用できないことから歩留りが改善されない．

3.6 むすび

格子結合マルチプロセッサの自律再構成アルゴリズムとして再帰的シフトを提案した．また，再帰的シフトを用いた再構成手法として，FS 法，BS 法を提案した．FS 法はシフトの方向を位置に依存させ 4 方向に固定したもので，再構成率はそれほど高くない．BS 法は，まずバイパスを行いシフトの方向を南方向に固定し，Kung と同程度の再構成率を得ることができた．また，これら FS，BS の再帰的シフトのアルゴリズムが停止することを証明した．

しかし，歩留りは，Kung と同程度であり，PE の利用効率がよいとは言えない．これは両手法とも方向を固定しているためと考えられる．そこで，次章では PE の位置に依存しない HS 法と呼ばれる新たな手法を提案し，再構成率の改善をはかる．

第4章

格子結合型マルチプロセッサの動的自律再構成法

4.1 まえがき

前章では，位置に依存した格子結合マルチプロセッサの自律再構成法を説明した．本章では，FS法のようにPEの位置に依存せず，PE全体を広く使えるような手法を提案する．

例えばFS法ではシフトの方向が固定であるため，別方向にあるかもしれない解を発見できない場合がある．また，BS法では東西方向は単純にバイパスしてしまうため多くの正常PEをも切り離し，東西方向にシフトすることによって得られるかもしれない解を切り捨てている．しかし，シフトの方向を決定するのは，その問題の見た目の平易さに比べ非常に困難な問題である．仮に全PEについて検索したとすると，そのオーダーは$4^{N\times N}$となりとても現実時間では解くことができない．そこで何らかの工夫が必要となる．そこで本章では乱数を用いた手法を提案する．乱数を用いているため，BS法やFS法のように停止性の証明はできない．また，解を発見しなかったときに，解が本当に存在しないとは言えない．

シフトの方向が固定だったFS法やBS法に比べて，HS法ではPEに新たに東西南北にデータをスルーする状態 (**PassEWNS**) を加える必要がある．また，再帰的シフトのアルゴリズムも若干改良が必要である．このため，FS法やBS法に比べるとハードウェア量が多くなる．

また，Kungらは，$1\frac{1}{2}$トラックモデルは2トラックモデルにマッピング可能であることを示している．本章ではPEに東西南北にデータをスルーできる状態を付加しても2トラックモデルにマッピング可能であることも示す．

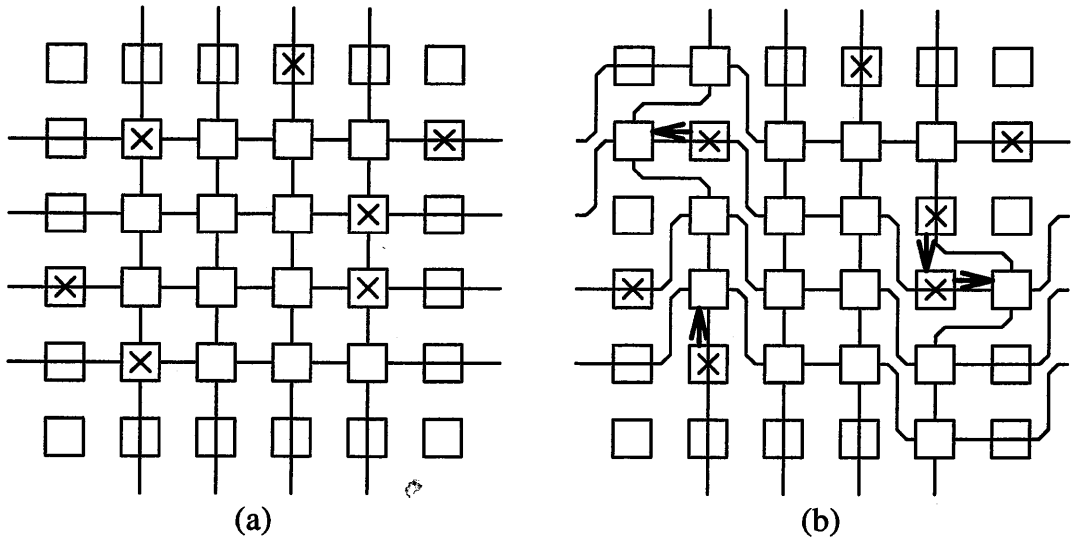


図 4.1　HS(Heuristic Shift) 法

本章の構成は以下の通りである．第 4.2節では位置に依存しない再構成手法である，HS 法について述べる．第 4.3節では本章で提案する HS 法の性能評価を行う．第 4.4節では，PE に新たなステータスを付加しても 2 トラックモデルにマッピング可能であることを示す．

4.2　HS(Heuristic Shift) 法

4.2.1　HS 法の概要

HS(Heuristic Shift) 法では，最初に FS 法と同様に中央部分に仮の配置をした後に，図 4.1のように，シフトの方向を試行錯誤的にランダムに選択する．よって，HS 法のシフトの方向の選択は，ある場合には正しい選択で，ある場合には間違った選択となるが，試行を繰り返すうちに正しい解を発見する可能性が高くなる．このように試行錯誤的にシフトの方向を選択するという考えかたは単純ではあるが，故障パターンに依存しないため，ローカルな情報のみでシフトを行うのに極めて有効である．また結果的に多くの組み合せを調べるので，複雑な形への拡張も容易である．

HS 法では，このループがいつまでたっても終わらない可能性があるので．本研究ではループの最大数を 10000 回とし，この数以上にシフト手続が発生した場合は回避できないものとして再構成を中止することにした．このように HS 法は時間がかかるという欠点

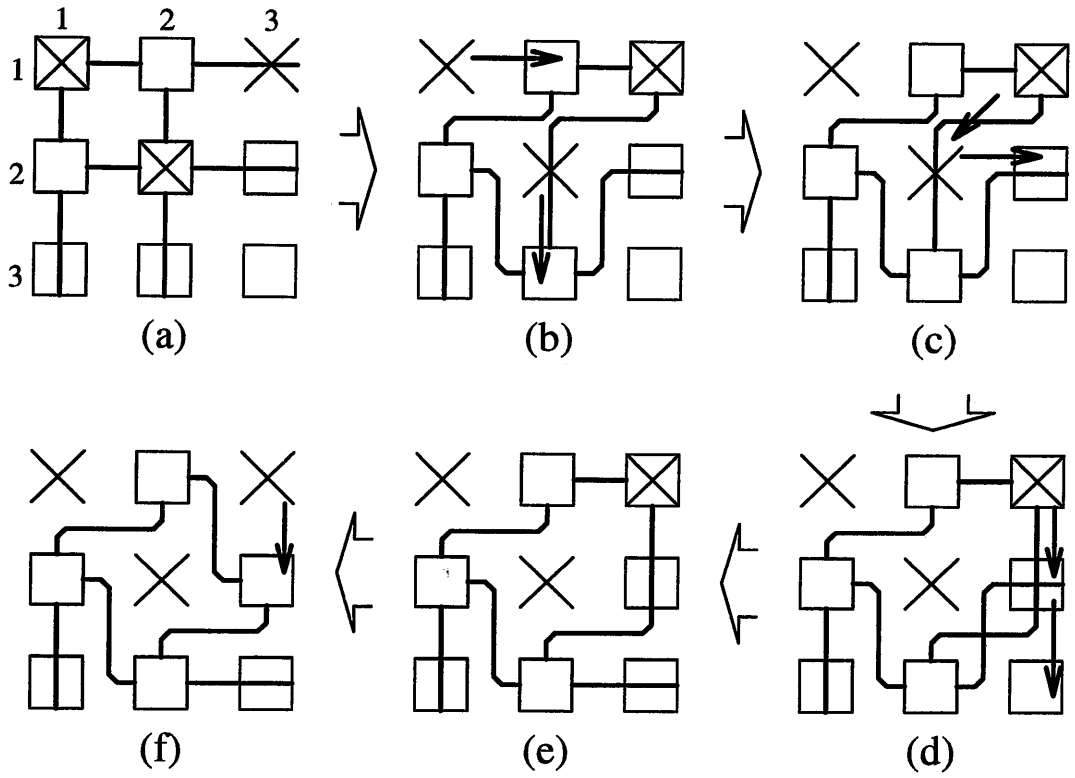


図 4.2　HS 法に現われる結合パターン

をもつ．また解があっても発見しない可能性もある．

4.2.2　PE ステータスの拡張

シフトの方向が固定の FS 法や BS 法と違って，HS 法にはこれまでにはないパターンが現れる．例えば，図 4.2(a) に示すように $[1,1],[2,2],[1,3]$ の PE が故障しており最初に $[1,1],[2,2]$ のようにシフトし，次に $[1,3]$ の PE が南方向へシフトしようと試みようとしたとする図 4.2(b)．しかし，HS が -1 となっており，そのままではシフトすることができない．そこで，$[2,2]$ の PE に東方向へのシフトシグナルを送り，南のポートの値 HS を 0 にする必要がある (図 4.2(c))．HS 法では，このようにシフトシグナルを送る PE が必ずしも東西南北の 4 つの PE の 1 つとは限らない．

シフトシグナルを受けとった，$[2,2]$ の PE は **Active** ではないが，**PassEW** となっており東方向のシフトを代行することができない．HS 法では，**Pass** 状態になっている PE をもシフトさせねばならない．

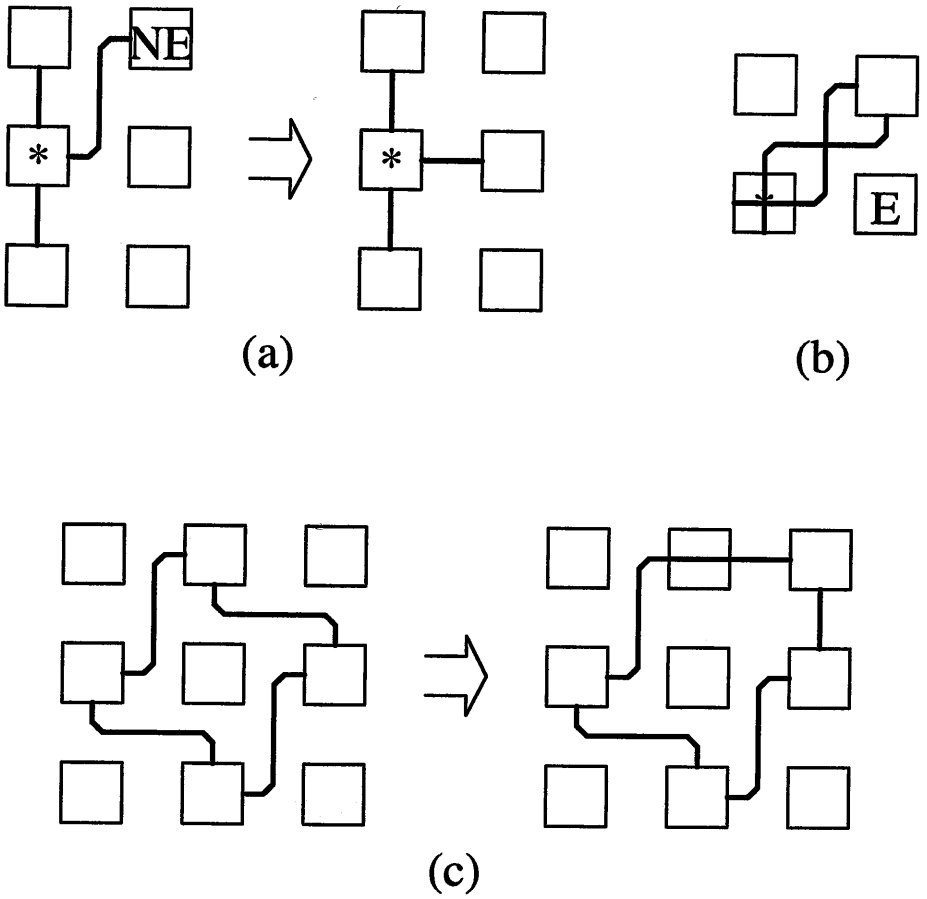


図 4.3　HS 法実現のための再帰的シフト

そこでさらに東方向の [2,3] の PE.へシフトシグナルを送る．[2,3] は **PassEW** であるからこれを受理し，[2,2] の PE はポートをつなぎ換える．しかし，この PE は **PassNS** であるから，正当性を守るためには図 4.2(d) のように [2,3] の PE のステータスを **PassEWNS** にしてやらなければならない．ここで [1,3] の PE の南のポートの値 HS が 0 になる．

この **PassEWNS** の状態は前章までで説明した，補償パス法や FS 法，BS 法には出てこない状態である．すなわち，HS 法では，今迄の手法に加え **PassEWNS** の状態が必要となる．

次に [1,3] は南の PE[2,3] に南方向へのシフトシグナルを送る．[2,3] は **Active** ではないが，東西にもデータをスルーしているので [1,3] の代行をすることはできない．そこで，[2,3] はさらに南方向へシフトしようと試みる．

南のポートの値 HS は−1 であるから，ここで先に述べた手法により，[2,3] は [3,2] へ

東方向へのシフトを送ることになる．すると同様の理由により [3,2] の PE が東へシフトするためには，[2,3] へさらに南方向へのシフトを送る必要があり，結果的に無限再帰呼び出しになってしまう．そこで，PE のステータスが **PassEWNS** の場合はシフトさせる方向のポートが 0 でなくとも，そのまま南方向の PE へシフトを行う．その後，図 4.2(e) のように **PassEWNS** の **PassEW** 成分だけ南方向へシフトさせてやればよい．

この段階で始めて [2,3] が受理できる状態になり，[1,3] の PE はポートをつなぎ換えることができる．

まとめると，HS 法を実現するには，

- PE ステータスとして **PassEWNS** の状態が必要．
- シフト前にポートのチェック手続きが余分に必要．
- Pass 状態でもシフトの可能性がある．
- Pass 状態をシフトさせる場合のポートの再結合の変更が必要．

が FS 法や BS 法などに比べて余分に必要となる．

4.2.3 PE 間シグナルによるシフトの改良

FS 法を変更した PE 間シグナルを次に説明する．以下にその説明するため，東方向にシフトする例について説明する．注目している PE を図と対応させて PE(*) のように表現する．

step1)

PE が **Idle** あるいは，**PassEW** 状態であるときはシフトは不要である．よって直ちに **True** を返し終了する．それ以外の状態で，これ以上東にシフト不可能な場合は **False** を返し終了する．

step2)

PE の状態が **PassNS** 以外のときは，PE は HE の状態をチェックし，もし HE = 0 でなければ HE = 0 になるようその PE にシフトシグナルを送る．例えば図 4.3(a) の例では PE(NE) に南方向へのシフトシグナルを送る．

ただし，図 4.3(b) のようなパターンの場合この手法を用いると無限ループに陥いる．よってこのときに限り何もしないことにする．

```
procedure shift_to_east(pe)
begin
    if status = Idle or status = PassEW then
        return True;
    if nomore shift to east then
        return False;
    if status = Active then
    begin
        if HE = -1 then
            shift_to_south(pe(NE));
        else if HE = 1 then
            shift_to_north(pe(NE));
    end
    if HN = -1 then
        shift_to_east(pe(NW));
    else if HS = -1 then
        shift_to_east(pe(SW));
    shift_to_east(pe(E));
    reconnect_east(pe);
end
```

図 4.4 HS 法実現のための再帰的シフトのアルゴリズム

step3)

HN および HS をそれぞれチェックする．例えばトラックが 1 のとき，HN $= -1$ であればトラック数が 1 であるため東方向へシフトすることができない．そこでこの場合 PE(NW) に東方向へのシフトシグナルを送る．(図 4.3(d)) トラック数が T ならば，HN $= -T$ のときにシグナルを送る．

step4)

PE(E) に東へのシフトシグナルを送る．これは FS 法と同様である．

step5)

再結合できるパターンになっていたならポートを繋ぎ換える (再構成)．

これらの手法は FS 法同様，各 PE はシステム全体を再構成し補償させようとするのではなく，自分だけを補償しようとする．他の PE が既に補償している PE にぶつかった場合，その PE の補償を無効にしてでも補償しようとする．これを繰り返すことにより，格子結合アレイ中の欠陥 PE を回避できアレイ再構成が実現される．再構成以外のアルゴリズムを図 4.4に示す．この図では shift を再帰的に呼び出したあとの戻り値のチェックおよ

```
procedure reconnect_east(pe)
begin
    HS(pe(N)) = HS(pe(N)) + 1;
    HN(pe(S)) = HN(pe(S)) + 1;
    HN(pe(E)) = HN - 1;
    HS(pe(E)) = HS - 1;
    if status = Active then
    begin
        status(pe(E))= Active;
        status = Idle;
    end
    else if status = PassNS then
    begin
        if status(pe(E)) = PassEW then
            status(pe(E)) = PassNSEW;
        else
            status(pe(E)) = PassNS;
        status = Idle;
    end
    else if status = PassNSEW then
    begin
        if status(pe(E)) = PassEW then
            status(pe(E)) = PassNSEW;
        else
            status(pe(E)) = PassNS;
        status = PassEW;
    end
end
```

図 4.5 HS 法実現のための再構成のアルゴリズム

び無限ループのチェックを省いている．本来ならば **False** が返ってきた時点で手続きを直ちに中断する．また再構成のアルゴリズムを図 4.5に示す．

あるステップに各 PE のシフト方向が重なると，故障のない PE をもシフトされることになる．例えば，図 4.6は最初に東に，次に南にシフトが生じた例である．このようにシフトは解を発見するまで行われるので，従来の補償パス法では生じないシフトパターンを発見する可能性がある．

4.2.4 無限ループの存在

前節で述べた HS 法のアルゴリズムを実行すると，図 4.7(a) のようなパターンが現われ

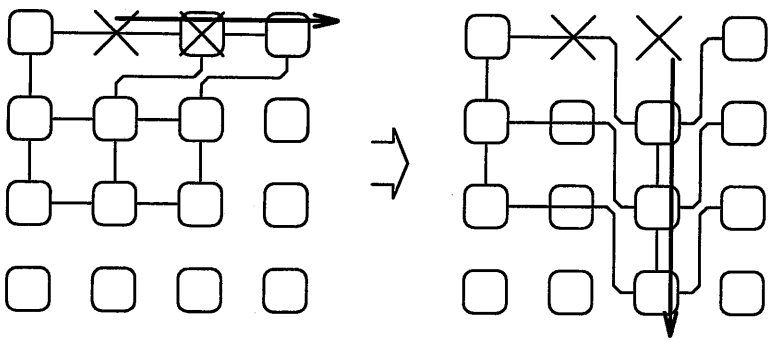


図 4.6 斜めへのシフト例

ることがある．例えば，図 4.2の最終状態の形がこれである．

今，図 4.7(a) の [2, 1] の PE に東方向へのシフトシグナルを送ったとする．東のポートの状態が+1 であるため，この PE は 0 に戻そうと [3, 2] の PE に北方向へのシフトシグナルを送る．[3, 2] の PE は次にさらに，北方向のポートを 0 に戻そうとし，[2, 3] へ西方向のシフトシグナルを送る．このように次々とシフトが発生し最終的には，[2, 1] に東方向へのシフトが発生する現象が発生し，無限ループに陥いってしまう．このようなパターンを劫と呼ぶことにする．このため，最大シフト数が 10000 を越えてしまい，シフトのパス上にこのパターンが存在した場合は **False** となってしまう．

これを解決するために，この場合に限り PE(*) の南西の PE に北方向のシグナルを送る前に東方向へのシフトシグナルを送ることにより，図 4.7(b) のようにうまくシフトすることができる．

無限ループ検出は PE にマークをつけておくことで簡単に発見できる．すなわち，再帰的シフトのシグナルを送る前にマークをつけておき，シグナルを受けとった時にマークがついていたら，それは劫であるとして処理を行えばローカル情報だけで検出することができる．

4.3 HS 法の性能評価

図 4.8に Kung らのグラフ理論を用いた格子結合型マルチプロセッサと，本論文で提案した再帰シフトに基づいた場合のシステム構成率 (Array Yield) の比較を示す．横軸は PE の歩留りで全体の何割が故障したかを示す．また縦軸はアレイが再構成できたかどうかを示すアレイ全体の歩留りである．

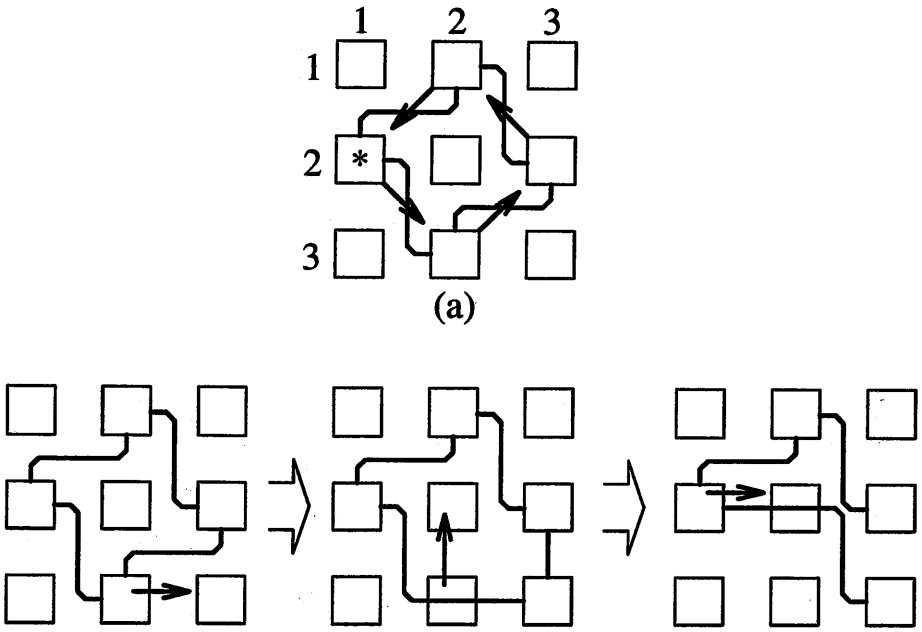


図 4.7　劫 (無限ループ)

このように HS 法は PE の歩留りが高いところにおいて従来法に比べ効果的であることがわかる．これは，従来法では PE の故障数が少ないときですら，解の発見が不可能なパターンが多いことを意味している．すなわち，非故障 PE の補償が重要であることがわかる．逆に PE の歩留りが低くなると，非故障 PE を補償する必要が必然的に少なくなるため，ほぼ同等か，場合によっては組み合せ問題を解いた方が良い結果を得る．

また，高い再構成率を得ることができるその他の理由としては

- 新たな状態 **PassEWNS** を導入した．
- 無限ループに陥りやすい状態を排除した．

等があげられる．例えば，**PassEWNS** を考慮しない場合，図 4.9のように再構成率が悪くなることがわかっている．

4.4　2 トラックモデルへのマッピング

HS 法では，PE の状態として東西南北にデータをスルーする状態をも許している．S.Y. Kung らは $1\frac{1}{2}$ トラックモデルは 2 トラックモデルに置き換えられることを示した．そこ

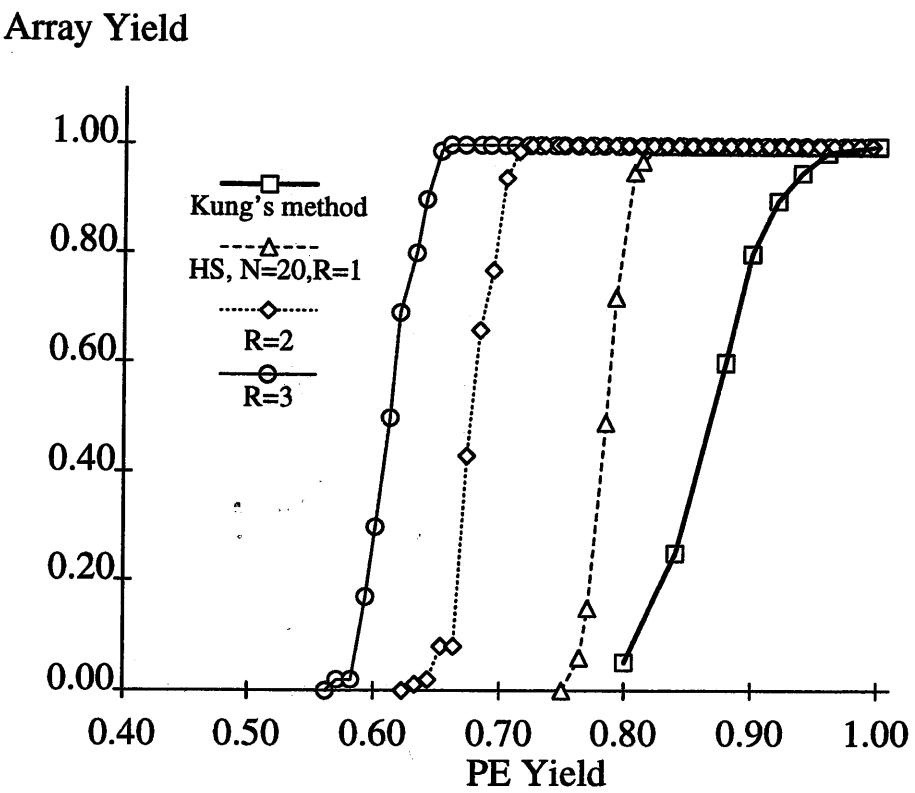


図 4.8　$N = 10, T = 1$ における HS 法の歩留り

で，本節では PE のステータスを拡張しても 2 トラックモデルに平面配置可能であることを示す．

補題 4.1 ステータスが **Active** である任意の二つの PE，$P_1[p_{1i}, p_{1j}]$, $P_2[p_{2i}, p_{2j}]$ を考え，その論理アドレスを $L_1(l_{1i}, l_{1j})$, $L_2(l_{2i}, l_{2j})$ とすると以下が成立する．

$p_{1i} \leq p_{2i}$ ならば $l_{1i} \leq l_{2i}$

$l_{1i} \leq l_{2i}$ ならば $p_{1i} \leq p_{2i}$

不等号が逆の場合および j についても同様である．

証明　PE の各ポートは i または j 方向に増加または減少するしかないのであきらかである．
□

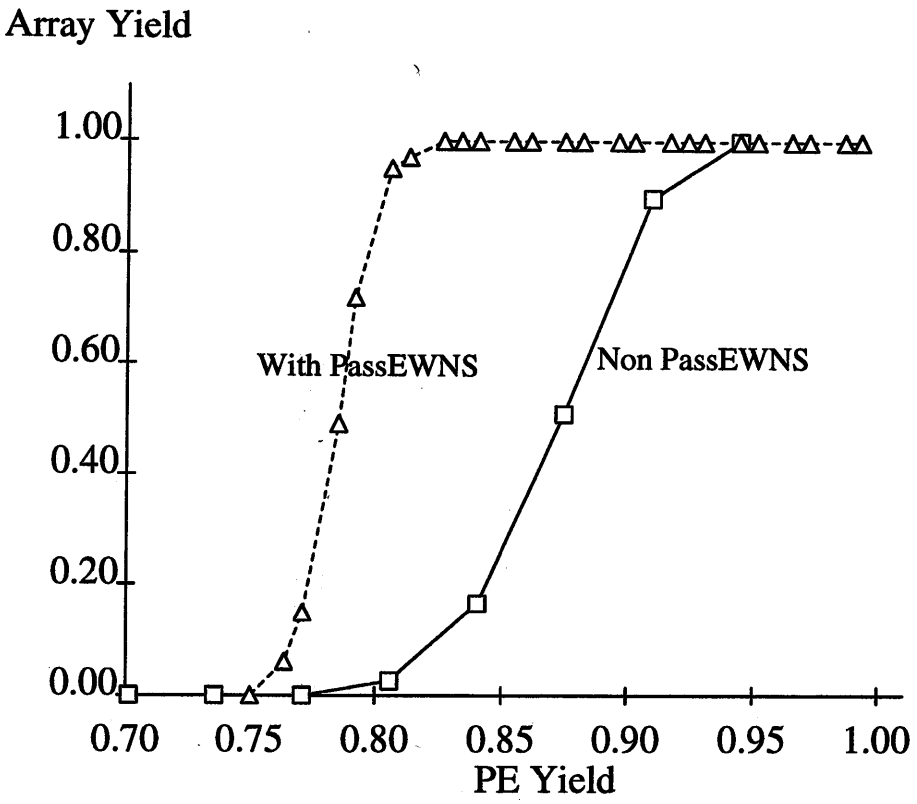


図 4.9 **PassEWNS** を考慮しなかった場合の歩留りの比較

補題 4.2 ステータスが **PassEWNS** の PE, $P[p_i, p_j]$ を考える．PE の東西南北の PE の論理アドレスを L_E, L_W, L_S, L_N とすると，そのうちの 2 つの PE は同じものである．

証明 東西南北の PE が L_E, L_W, L_S, L_N と全て違うものと仮定する (図 4.10(a))．補題 2.2 より $l_{Ni} \leq l_{Wi}$ である．また同様に，$l_{Si} \geq l_{Wi}$ である．しかし，l_{Ni} と l_{Si} はそのアドレスが 1 しか違わない．故に $l_{Ni} = l_{Wi}$ または $l_{Si} = l_{Wi}$ である．同様にして，$l_{Nj} = l_{Wj}$ または $l_{Nj} = l_{Ej}$ である．仮に $l_{Ni} = l_{Wi}$ とすると L_W の西のポートまたは，L_E の東のポートが L_N と接続する必要がある．故に $l_{Nj} < l_{Wj}$ である．$l_{Nj} = l_{Sj}$ であるから，$l_{Sj} < l_{Wj}$ である (図 4.10(b))．しかし，L_W の南のポートと L_S の東のポートを接続するには，トラックをクロスしなければならない．しかし，スイッチのステータスは前述したように，3 通りであるから矛盾する．$l_{Ni} = l_{Ei}$ としても同様である．すなわち，このようなパターンは存在しない．これは，図 4.10(a) のようなパターンを考えたから矛盾したのであって，図 4.10(c) のようなパターンであれば矛盾しない．

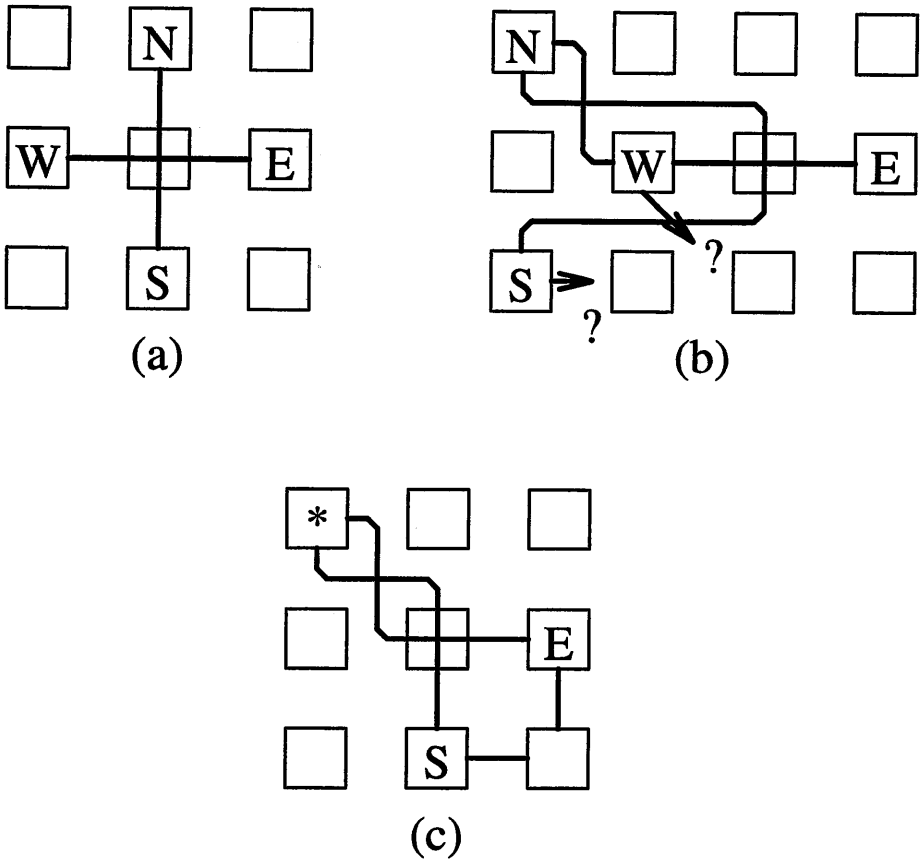


図 4.10　**PassEWNS** のマッピング

補題 4.3 $T\frac{1}{2}$モデルは，$T+2$ モデルに平面配置可能である．

証明　例えば PE のステータスに応じて図 4.11の配置するように決めておけば，図 4.12右上のようにあきらかにマッピング可能である．□

定理 4.1 $T\frac{1}{2}$モデルは，$T+1$ モデルに平面配置可能である．

証明　図 4.12右下のように各 PE の N 方向にあるトラックを取り去り，丸印のところの N 方向のバイパスを 1 つ北にシフトさせる．PE 間にあるトラックは常にT本以下となるのは自明であるから，これは矛盾なく行うことができる．同様にして，各 PE の W 方向にあるトラックを取り去ることができる．よって図 4.12左下のように$T+1$ モデルに平面配置可能である．□

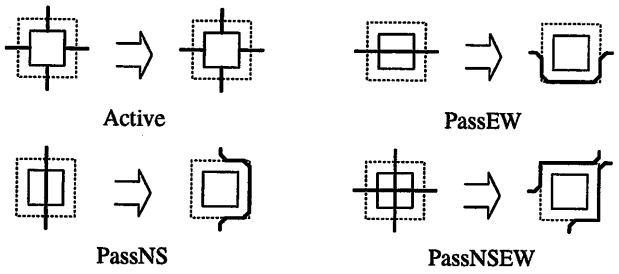


図 4.11 PE のマッピング

4.5 むすび

格子結合ネットワークを再構成する手段として位置に依存しない HS 法を示し性能評価を行った．HS 法は，次のような特徴を持つ．

- ローカルな情報だけで再構成可能である．
- 高い再構成率を得ることができる．
- 必要とする冗長化回路は従来のモデルと同等かそれ以下である．

逆に言えば，現在まで提案されている手法は PE を有効にしてなかったとも言える．しかし，HS 法はアルゴリズムに乱数を導入しているので，解を確実に発見もしくは発見不能と答えることができない．さらに，BS 法や FS 法が確実に停止することがわかっているのに対し，HS 法は停止性が明確でない．また，BS 法は列方向にシフトしないので，ハードウェアを簡略化できるという特徴を持ち，これだけではどちらが良いとはいえない．

しかし，いづれも本手法は冗長 PE 等に制限がないため，超並列システムを構築する場合に最適な手法を選び，最的な冗長 PE を選ぶことが可能である．故に他の多くの手法に比べアドバンテージを持つものと考える．

最後に，図 4.13に 20×20 の格子結合型アレイに 1 行 1 列スペアプロセッサを追加し，トラック数を 1 とした冗長格子結合型アーキテクチャ(N=10,R=1,T=1) の再構成例および，2 行 2 列スペアプロセッサを加えトラック数を 2 とした冗長格子結合型アーキテクチャ($N = 10, R = 2, T = 2$) の例をあげる．

前者は故障 PE が 30 で冗長 PE の総数が 44 個，後者は故障数が 70 で冗長 PE の総数が 96 個であるから，その冗長 PE の大多数が故障してもシフト可能であることがわかる．

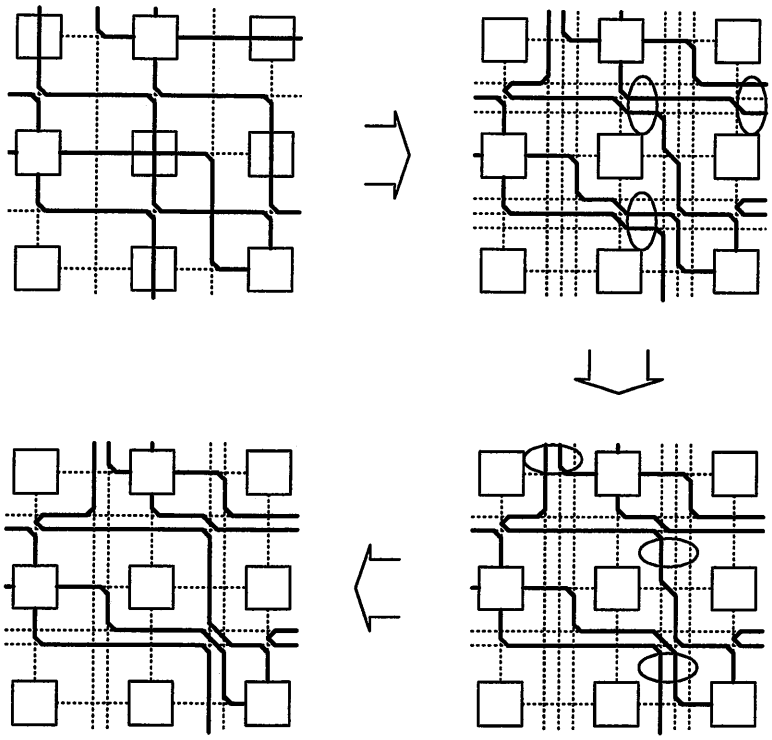
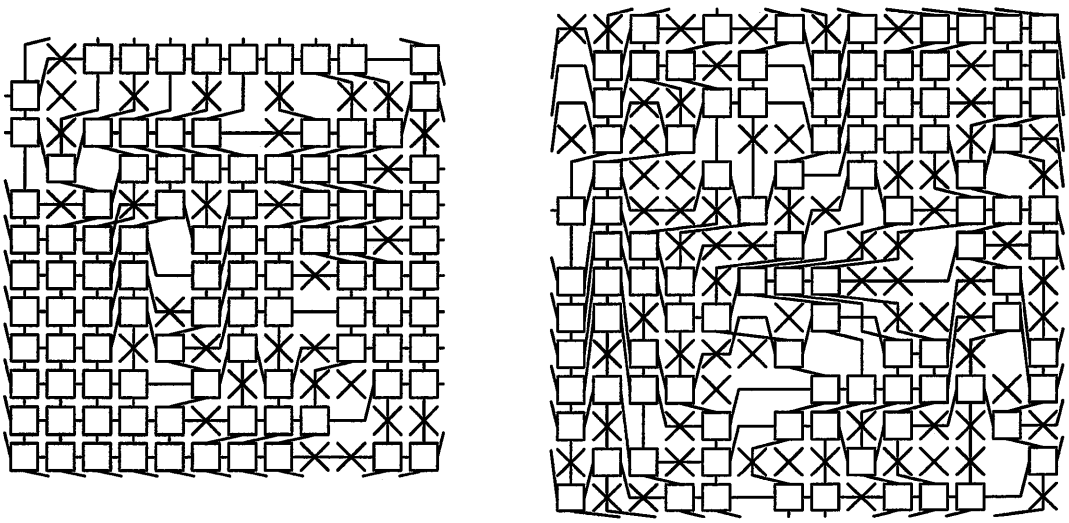


図 4.12　$1\frac{1}{2}$トラックモデルの 2 トラックモデルへのマッピング



N=10,R=1,T=1
Fault=30

N=10,R=2,T=2
Fault=70

図 4.13　HS 法の再構成例

第5章

格子結合マルチプロセッサの再構成法の総合評価

5.1 まえがき

前章までで，格子結合マルチプロセッサの静的再構成法としてFS法およびBS法，動的再構成法としてHS法を提言した．本章ではこれらの手法および従来手法の性能の評価および比較検討を行う．本章の構成は次の通りである．

第5.2節では各タイプの歩留りについて比較検討を行う．第5.3節では各タイプの再構成不能となる最少故障数を示す．第5.4節では各タイプのハードウェア量について比較検討を行う．第5.5節では各タイプの計算量について比較検討を行う．

5.2 歩留りの比較

本節では前章までに提言した静的再構成法，動的再構成法，従来手法の歩留りを比較し，どの手法が超並列マシンに向いているかを比較検討する．歩留りの評価方法としては，横方向にPEの歩留り (どのくらいの確率で生きているか)，縦方向にアレイの歩留り (どのくらいの確率で再構成できるか) をとる．

図5.1にトラックを1とし，10×10 に1行1列冗長PEを付加した際の各タイプの歩留り，Kungらの手法 [5] による歩留り，およびVarvarigouらの手法 [13] を示す．Kungらの手法は，$1\frac{1}{2}$トラックモデルにおける結果である．Varvarigouらの手法は，3トラックモデルに1行1列冗長PEを付加したもので，本研究における，$2\frac{1}{2}$モデルに相当し，いずれも故障PEにしか注目しておらず，グローバルな情報を必要とする．

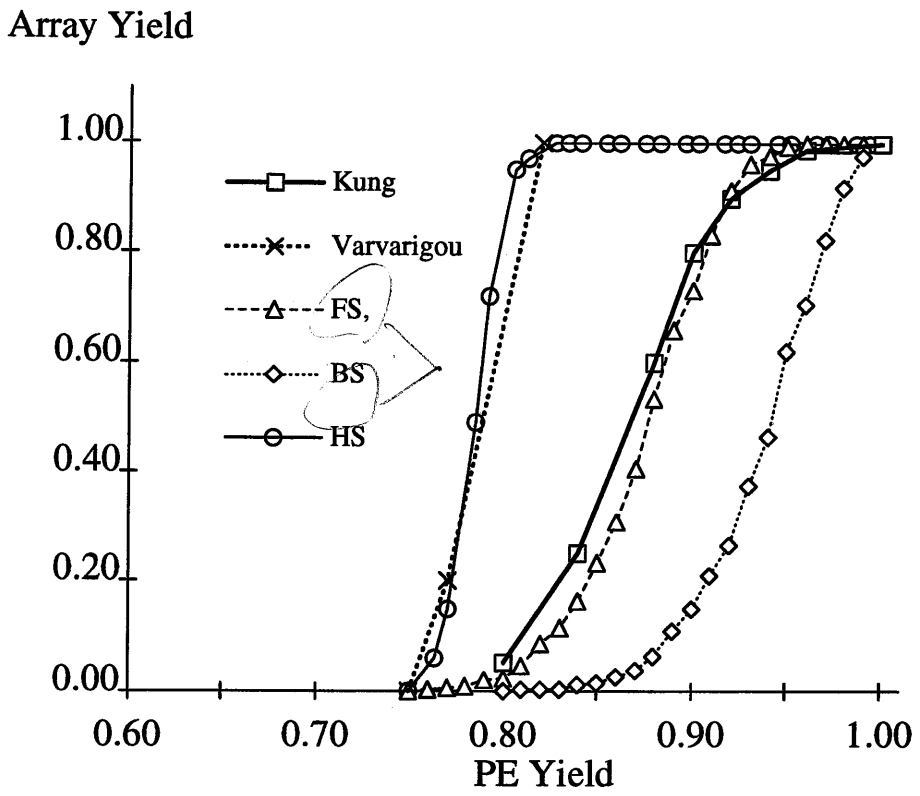


図 5.1 $N = 10, T = 1$ における各タイプの歩留り

この図から HS 法は Varvarigou らはの手法とほぼ同様の歩留りを得ることができることがわかる．Varvarigou らの手法は，3トラック用いており，HS 法2がトラックで実現できることを考えると，HS 法はより少ないハードウェア量でかつ同等の歩留りを得ている．

また図 5.2にトラックを2とし，10×10 に2行2列冗長 PE を付加した際の各タイプの歩留まりおよび Jean らの手法 [4] の結果を示す．ただし，FS 法はこの図では割愛している．Jean らの手法は補償パスの組み合せ問題を近似して解いているもので，BS 法と比べてもかなりの部分の PE を有効に使っていないことがわかった．また，Kung らに代表される補償パスを用いるやりかたは，故障 PE 以外の PE をシフトさせることがないため，故障 PE 数が比較的少ないところでの再構成率がよくない．HS 法は乱数を用いているのにもかかわらず，Kung らの手法に比べはるかによい結果を得ている．またグラフの傾きも急峻で十分な回数の試行錯誤の後，解をほぼ間違いなく発見していると考えられる．

次に HS 法において，冗長 PE を1行1列に固定し，アレイサイズを変化させた場合のアレイ歩留りの変化を図 5.3に示す．この図から，アレイサイズが小さい方が歩留りがよ

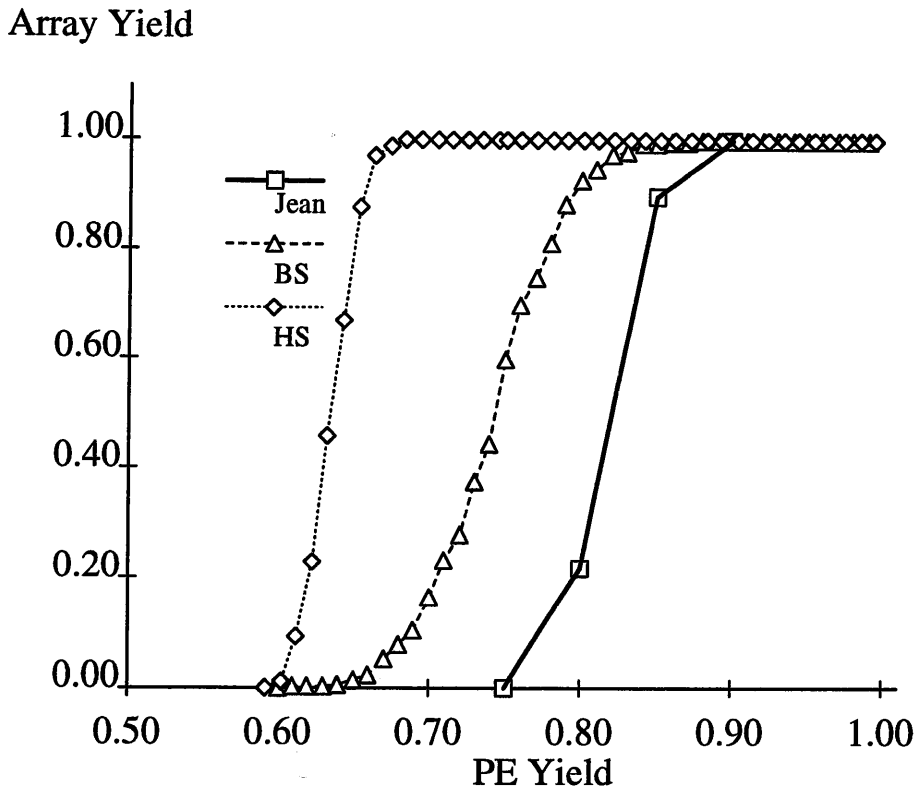


図 5.2 $N = 10, T = 2$ における各タイプの歩留り

いことがわかる．これは次のような理由による．

- アレイサイズが巨大になるにつれ，故障 PE の数が増える．
- 全体 PE が N^2のオーダで増えるにかかわらず，冗長 PE は Nのオーダでしか増加しない．

故に，巨大なアレイに冗長化を施すより，小さいアレイに冗長化を施したほうが再構成率がよいことがわかる．

5.3 再構成不能となる最少故障数の比較

本節では，$T = 1, R = 1$ のとき各タイプにおいて，最低いくつ故障すれば再構成不能となるかを考察する．

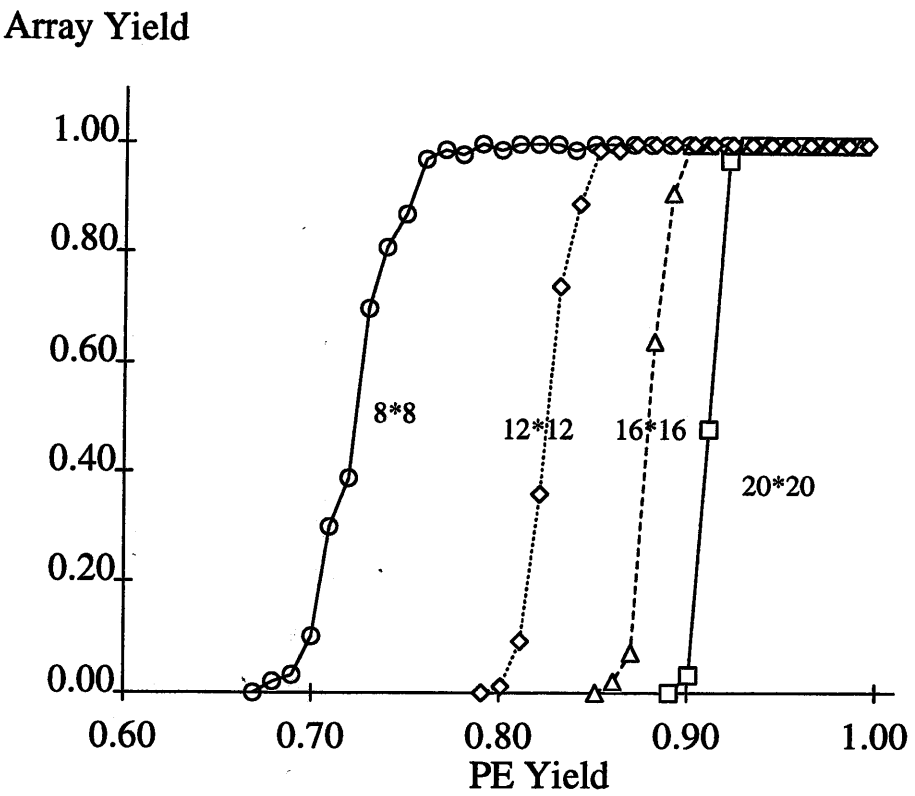


図 5.3 $HS, R = 1, T = 1$ における各タイプの歩留り

補題 5.1 Kung らの提唱した補償パスを用いる手法では最悪 PE が 5 個故障した場合再構成不能となる.

証明 故障が 4 つまでは必ず矛盾なく補償パスを選べる. しかし, 図 5.4(a) に示すように故障が 5 つ発生した場合は補償パスを矛盾なく選べない. 故に補償パス法では, 再構成不能となる最少故障は 5 である. □

補題 5.2 FS 法は最悪 2 個故障した場合に再構成不能となる.

証明 FS 法では外周に向って決められた方向にシフトするため, 図 5.4(b) に示すように 2 つ以上の故障 PE がそのパス上にあるとき再構成不能となる. 故に最少個数は 2 つである. □

補題 5.3 BS 法は最悪 9 個故障した場合に再構成不能となる.

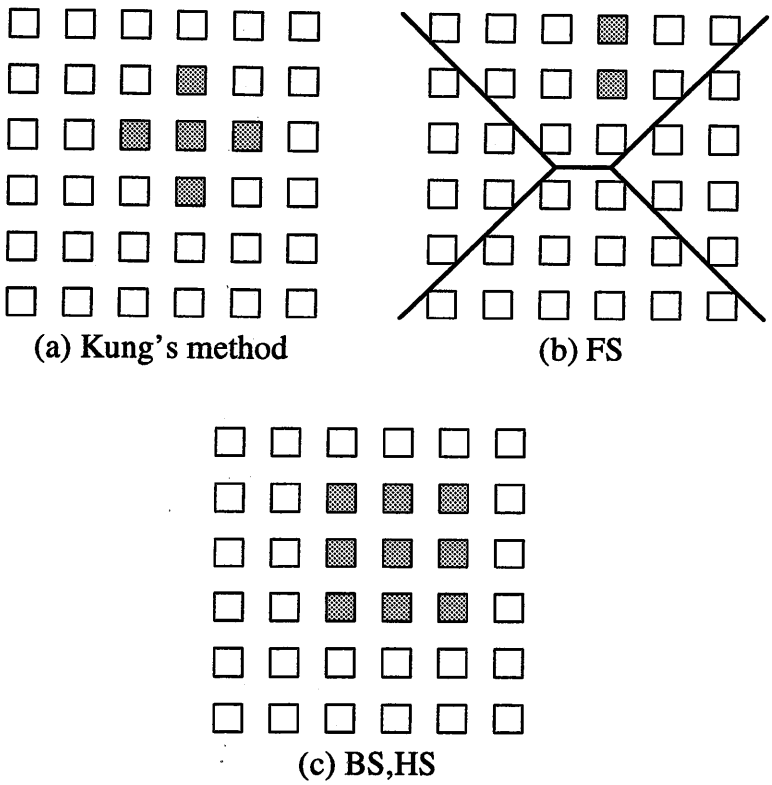


図 5.4 再構成不能となる最少故障数 $T=1, R=1$

証明 BS 法では南北方向にしかシフトしない．1 つの列には 2 個の冗長 PE しかないため，列の PE が 3 つ以上故障した場合は再構成不能である．しかし，バイパスを故障の多いところから行うため 3 個故障した列が少なくとも 3 つ必要である．故に図 5.4(c) に示すように 9 が最少個数である．□

補題 5.4 BS 法で再構成できる故障パターンは，HS 法でも再構成できる．

証明 最初のバイパスを東西方向へのシフトと考えればあきらかである．□

補題 5.5 HS 法において論理アドレス (l_i, l_j) の PE は，物理アドレス $[p_i+1, p_j+1]$ およびそれを中心とした周囲 8 箇所のいずれか，すなわち，$[p_i, p_j], [p_i+1, p_j], [p_i+2, p_j], [p_i, p_j+1], [p_i+1, p_j+1], [p_i+2, p_j+1], [p_i, p_j+2], [p_i+1, p_j+2], [p_i+2, p_j+2]$ のいずれかしか存在できない．

証明 今，$(n, 0)$ の PE を P_0，その i 成分を p_{0i}，$(n-1, l_j)$ の PE を P_1，その i 成分を p_{1i}，$(n-2, l_j)$ の PE を P_2，その i 成分を p_{2i} と呼ぶことにする．P_0 が，$[n-1, 0]$ に存在

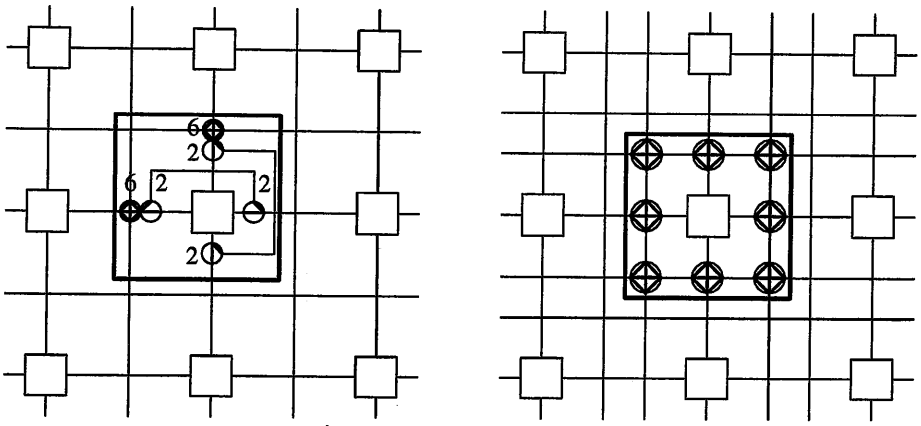


図 5.5 $1\frac{1}{2}$トラックモデルおよび，2トラックモデル

したとすると．補題 4.1より，P_1の PE の物理アドレスの i 成分は P_0のそれより小さい．同様にして，P_2の PE の物理アドレスの i 成分は P_1のそれより小さい．故に，

$$\underbrace{p_{i0} > p_{1i} > p_{2i} > \cdots > p_{ni}}_{n \text{ times}} \tag{5.1}$$

が成立する．よって，$p_{ni} < 0$ となり矛盾する．j方向および，増加方向についても同様である．□

補題 5.6 HS 法は最悪 9 個故障した場合に再構成不能となる．

証明 補題 5.4より BS 法の解は HS 法の一部であるから，9 個のときに再構成不能となるパターンを示せば十分である．図 5.4(c) に示すように 9 個故障すると，補題 5.5より，中央の PE はどこにも移動することができない．よって，9 個が最少個数である．□

従来法は 5 つ故障しただけで再構成不能に陥いる可能性があるのに対して，BS 法，HS 法は 9 個故障しなければ再構成不能にならない．これからも，本論文の手法の優位性が導かれる．また BS 法は 3 個 PE が故障した列が 3 つ存在した場合に再構成不能となるが，HS 法は 9 個集中して故障した場合にのみ再構成不能となる．これらの点からも HS 法が高い再構成率を得ることが示された．

5.4 ハードウェア量の比較

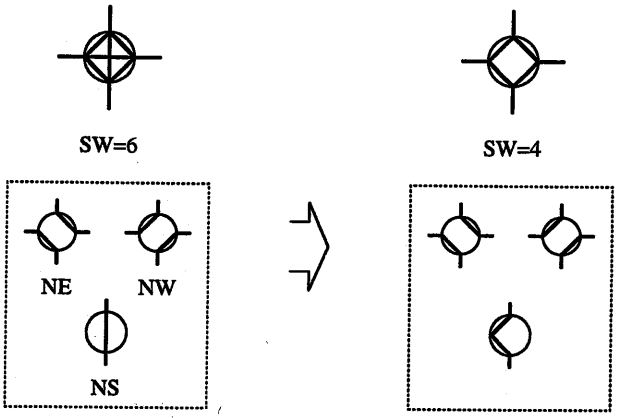


図 5.6 スイッチの簡略化

格子結合ネットワークにおいてスイッチとトラックの量は極めて重要なポイントである．本研究で述べた歩留りはスイッチやトラックの故障を無視しているが，実際にはこれらの部分の故障をも考える必要がある．また，WSI などにインプリメントする際には，スイッチやトラックの占める面積が問題となりウェーハ全体に配置できる PE の数をも制限しうる．

そこで，ここでは一般性を持たせるため，図 5.5のように 2 トラックモデルと $1\frac{1}{2}$トラックモデルに各手法をインプリメントした際，どのくらいスイッチやトラックが必要であるかを考える．

スイッチ量は太線の中のスイッチの数を単純にカウントしたものとする．例えば，図 5.5(a) では 20，(b) では 48 である．

トラックの量は図の太線で囲んだ領域を一辺の長さが 1 の四角形と仮定し，どのくらいの長さがあるかで評価する．トラックの間隔は全て均等とし，バイパスに要するトラックの長さは 1 とし，PE の大きさは無視することにする．例えば，図 5.5(a) では 6，(b) では 6 である．

しかし，スイッチの数はもう少し減少することができる．例えば，EW の状態をとらないと仮定すると，図 5.6のようにスイッチは 4 にまで減少させることができる．そこで，本節では各モデルの条件によりどのくらいスイッチやトラックが必要かを議論する．

5.4.1 $1\frac{1}{2}$トラックモデルのハードウェア量

HS 法および補償パス法は，全ての PE を上下または左右バイパスするためのトラック

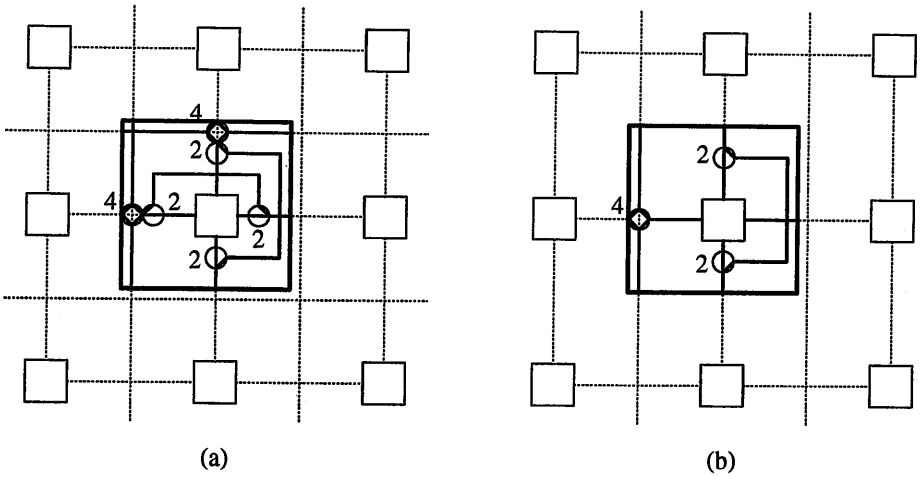


図5.7 $1\frac{1}{2}$トラックモデルに必要なスイッチおよびトラック

およびそのスイッチが必要であるため，図5.7(a)のように16個のスイッチが必要となる．トラックはバイパス部の長さを1と仮定し，PEの大きさを0とすると，6必要である．また，静的再構成法のようにシフトの方向が決定している場合は一方行のみのバイパスを考えればよいのでさらにハードウェアを軽減できる．例えばBS法では，列方向にシフトしないため南北間のスイッチを減らすことができる．行方向にバイパスしないため東西方向のバイパス用のトラックが不要である．よって，図5.7(b)のようにスイッチ数8，トラック数4まで減少させることができる．

5.4.2 2トラックモデルのハードウェア量

次に2トラックモデルについて考える．従来の補償パス法は，南北方向，東西方向へバイパスさせるためのスイッチを考慮し，図5.8(a)のようにすればよい．ここで左上のスイッチは切り替える必要がないため不要であることに注意する．故に23個のスイッチおよび6のトラックが必要である．

次に静的再構成法の場合を考える．この場合$1\frac{1}{2}$と同様，図5.8(b)のように簡略化できる．この場合，スイッチの個数が8，トラックが$4\frac{1}{3}$のトラックがあれば十分である．このようにBS法は従来手法の半分のトラックで実現できる．

次に東西南北にデータをスルーさせる動的再構成法を考える．HS法の2トラックへのマッピングは図4.12に示している．これを実現するには，図5.8(c)のようにスイッチが，30個トラックが6必要となる．このようにスイッチが増える理由は，**PassNSEW**を実

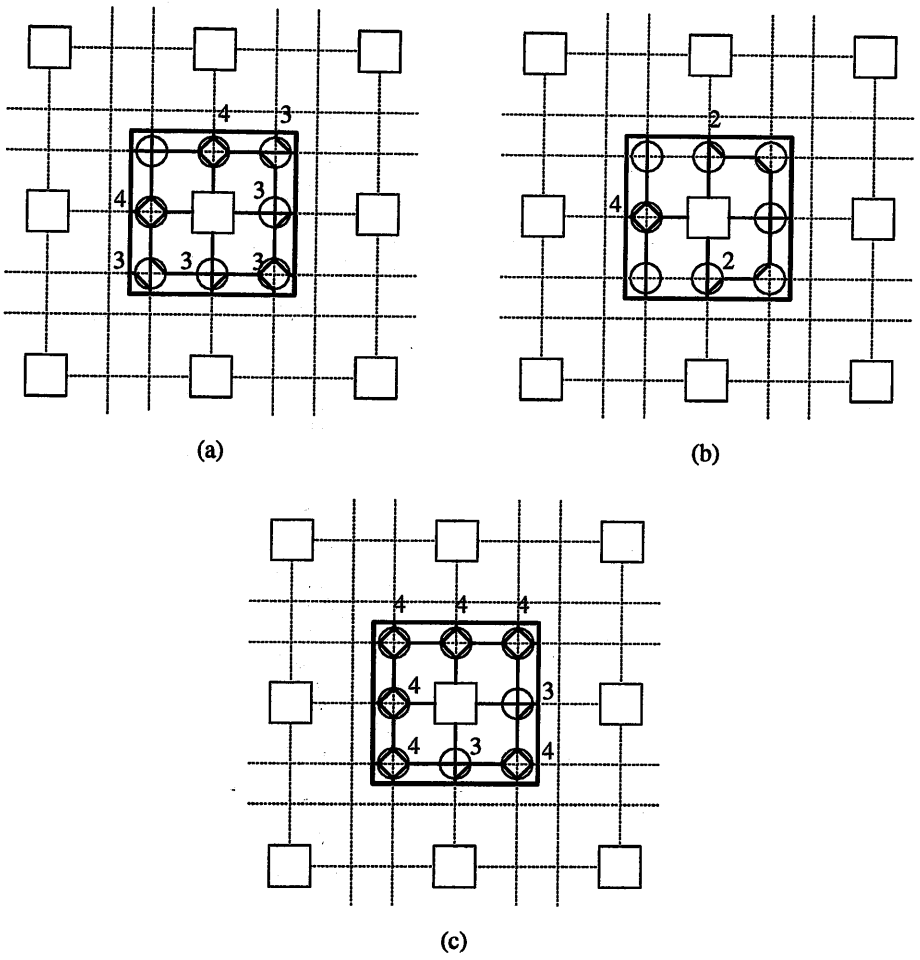


図5.8 2トラックモデルに必要なスイッチおよびトラック

現するためだけに使用されるスイッチが増えるためである．しかし，トラック配置を $1\frac{1}{2}$ モデルのように工夫すれば，十分少ないスイッチで実現可能である．

各タイプのハードウェア量の比較として，2トラックタイプと $1\frac{1}{2}$ タイプにマッピングした際のトラック量とスイッチ量を表5.1に示す．FS，BSでは2トラックの方がハードウェアを少なくできるが，HS法や従来の補償パス法は $1\frac{1}{2}$ トラックの方がよい．これは，HSを2トラックにマッピングする場合，東西または南北にスルーさせるときにしか使わないチャネルが増えるためである．しかし，HSでも $1\frac{1}{2}$ トラックならば十分ハードウェア量は少ない．これより，シフトの方向が固定であるFS法およびBS法は2トラック，そうでないものは $1\frac{1}{2}$ トラックを用いたほうがよいことがわかる．

Type	2-tracks		1-1/2track	
	# of sw	track length	# of sw	track length
Kung	23	6	16	6
FS	8	4-1/3	8	4
BS	8	4-1/3	8	4
HS	30	6	16	6

表 5.1 各タイプに必要なハードウェア量

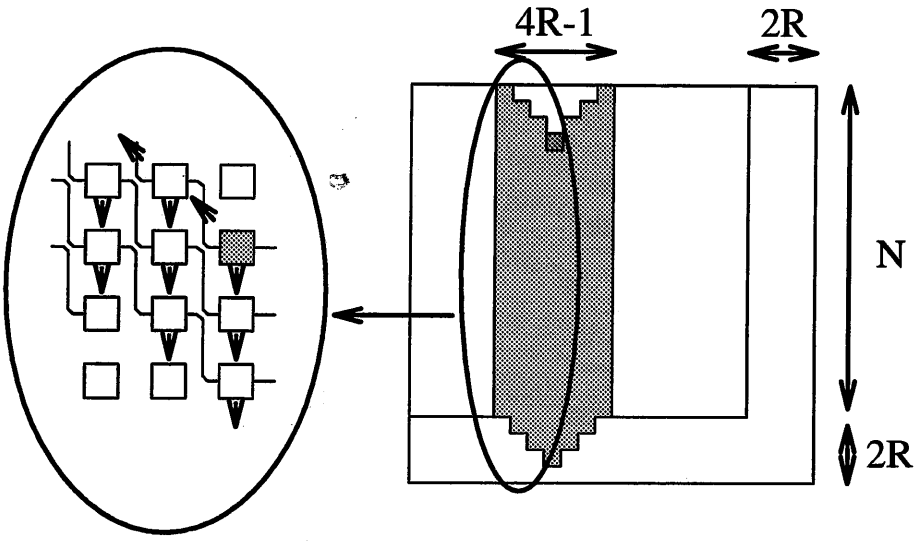


図 5.9 再帰的シフトの計算量

5.5 FS 法，BS 法の計算量

BS 法はバイパスとシフトから構成される．アレイのサイズを $N \times N$ とすると，バイパスはアレイサイズや故障数にかかわらず各列の故障数をカウントするのに N，バイパスする列を決定するのに $2N$ 必要だから，冗長 PE が R のときは全体で，

$$2RN$$

必要である．次にシフト部について考える．シフトは全体の状態によりシグナルが送られる PE の数が変化する．ワーストケースは図 5.9 に示すような場合で，このとき $N \times (4R-1)$ の PE に再帰的シフトが送られる．故にシフトの回数は最悪でも，

$$(4R-1)N$$

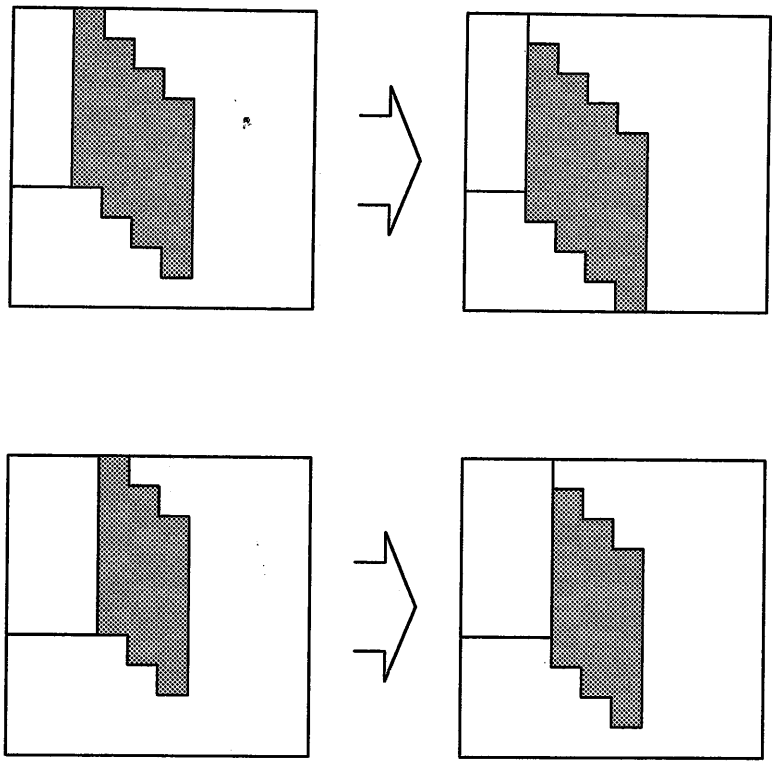


図5.10 シフトによって移動するPEの数

となる．

次にループについて評価する．1回のシフトで移動するPEの数は図5.10のように，移動する列の数で決定できる．しかし初期配置した時点で，各PEは$2R$回しか移動できない．よってシフトの回数は$2N^2R$で押えられる．1回のシフトでPEが1つ動いたとすると，$2N^2R$回シフトが発生する．$(4R-1)N$個PEが動いたとすると，$2N^2R/((4R-1)N)$回シフトが発生する．すなわち一回のシフトでいくつPEが動いたとしてもループを含めたシフトの回数は，最悪でも

$$2N^2R$$

となる．これはあきらかに，全PEが移動したと仮定した場合に等しい．また，同様に考えによりFS法も

$$N^2R$$

となることがわかる．故に，FS，BS法のオーダとしては，

$$O(N^2R)$$

Type	歩留り	ハードウェア量	計算量
Kung	中	多	大
FS	低	少	小
BS	中	少	小
HS	高	多	-

表 5.2 各タイプの総合評価

となる．一方補償パスを用いる手法は，Roychowdhury[10] らが，故障数 $|F|$，$R=1$ のときに，

$$O(|F|^2)$$

で解くアルゴリズムを提案している．この手法では，$|F|$ が増加するにつれ遅くなることがわかる．例えば，$N=20$，$R=1$ とすると，$|F|=4$ までは，Roychowdhury らの手法が早い．しかし，それを越えると Roychowdhury らの手法は 2 乗のオーダで遅くなる．しかし本アルゴリズムは一定時間で押えられており，Roychowdhury の手法に比べ故障数の多いところで有利である．

5.6 むすび

格子結合マルチプロセッサの自律再構成法の総合評価について述べた．評価は，歩留り，最少故障数，ハードウェア量，計算量について行った．

- 歩留り
  HS 法がもっともよい結果を得ることができ，続いて補償パス法，BS 法と続くことがわかった．

- ハードウェア量
  BS 法，FS 法のようにシフト方向を固定するとハードウェア量を半分程度に減少できる．また HS 法は 2 トラックモデルより $1\frac{1}{2}$ モデルにマッピングしたほうがよい．

- 再構成不能となる最少故障数
  歩留りが高い HS 法が 9 個ともっとも高い値を示すことを証明した．また，補償パス法は 5 個と冗長 PE を有効に使っていないこともわかった．

- 計算量

  BS法，FS法のオーダは全PEをシフトする可能性があることから，$O(N^2R)$ であるため故障が増えても一定量で押えられる．一方補償パス法は故障数 $|F|$ で効いてくるため，故障が増える毎に遅くなる．HS法は，そもそも停止するかどうかもわからないためオーダの評価は不可能である．

以上の結果から格子結合型マルチプロセッサの自律再構成法としてはHS法が有効であると考えられ，超並列コンピュータをインプリメントするのに適していることがわかる．しかし，ハードウェア量の点から見るとBS法は，HS法に比べ半分程度までスイッチを低減でき，トラック量も少ないため，一概にはHS法が良いとは言えない．これは一種のトレードオフであり定量的に評価するには，PE，スイッチ，トラックの占める面積を評価する必要がある．これは今後の課題である．最後に各タイプの評価を表5.2にまとめる．

第6章

階層型冗長構成法および3次元格子結合マルチプロセッサ

6.1 まえがき

先に述べたように，格子結合型マルチプロセッサの自律再構成法として，HS法が有力であることがわかった．しかし，アレイサイズは 10×10 から 20×20 程度であり超並列システムを考える場合は少くともアレイサイズ 100×100 を考察する必要がある．

また，前節の結果から小さいアレイに冗長化を施した方が効果的であることがわかっており，100×100 のアレイに冗長化を施しても再構成率はさほど期待できない．そこで本節では，階層化を用いたより大きなアレイの構築について論じる．

10^6規模の超並列マシンを構築した場合を考えてみると，2次元格子結合ネットワークでは，1000×1000 のアレイとなりその径は巨大なものとなる．しかし，3次元では，$100 \times 100 \times 100$ となり，同じプロセッサ数で径を小さくすることができる．

またパッケージングの面でも近年WSIにインプリメントしようという動きがさかんで，例えばMichael L. らは，3次元スタック構造のニューロコンピュータを提案している [1]．また，R.J.Wojnarowski ら [14] は，2D ウェーハを積み上げたハイブリッド型WSIを提案している．

本章ではこれらの背景をふまえ，前章までに提案した2次元格子結合プロセッサの冗長化技法を3次元に応用することを考える．

本章の構成は以下の通りである．第6.2節では大規模システム構築の場合の階層化技術について論じる．第6.3節では，3次元冗長化格子結合ネットワークについて述べる．第6.4節では，3次元冗長化を行った場合の結合条件を示す．第6.5節では，再帰的シフトを

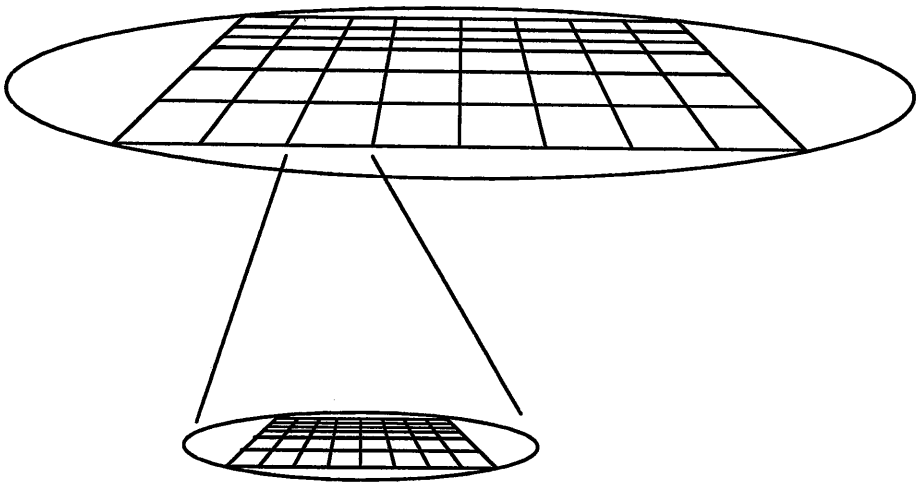


図 6.1　格子結合ネットワークの階層化

3 次元に拡張する．第 6.6節では，3 次元 HS 法について述べる．第 6.7節は，3 次元冗長化格子結合ネットワークの性能評価である．

6.2　階層型冗長構成法

図 6.1に本論文における階層化の一例を示す．この図では，$(N+R) \times (N+R)$ のアレイの各成分 1 つ 1 つを $(N+R) \times (N+R)$ のアレイで構成し，全体で $N^2 \times N^2$サイズのアレイを得る方式である．階層化は，小さいサイズのアレイの積み重ねることにより巨大なサイズのアレイを作るので，小さいアレイを評価することにより巨大サイズの性能も評価できるという利点がある．

そこで，比較的小さなアレイサイズ 10×10 において，HS が比較的高い再構成性能を達成することがわかったので，本論文ではこの HS 構成法を階層化超並列システムに適用する．図 6.2にシステムを 2 階層にした場合のアレイ歩留りを示す．グラフは HS の N=10,T=1,R=1 および，T=10,T=1,R=2 の場合から読みとり求めた．また，参考として階層化を施さない，N=100, T=1, R=44 の BS 法の値を記した．階層化を施したのと，そうでないとの違いがはっきりとわかる．このように，超並列システムでは，階層化が重要な役割を演じることが確認できた．

6.3　$1\frac{1}{2}$トッラク-1 スペア型 3 次元格子結合マルチプロセッサ

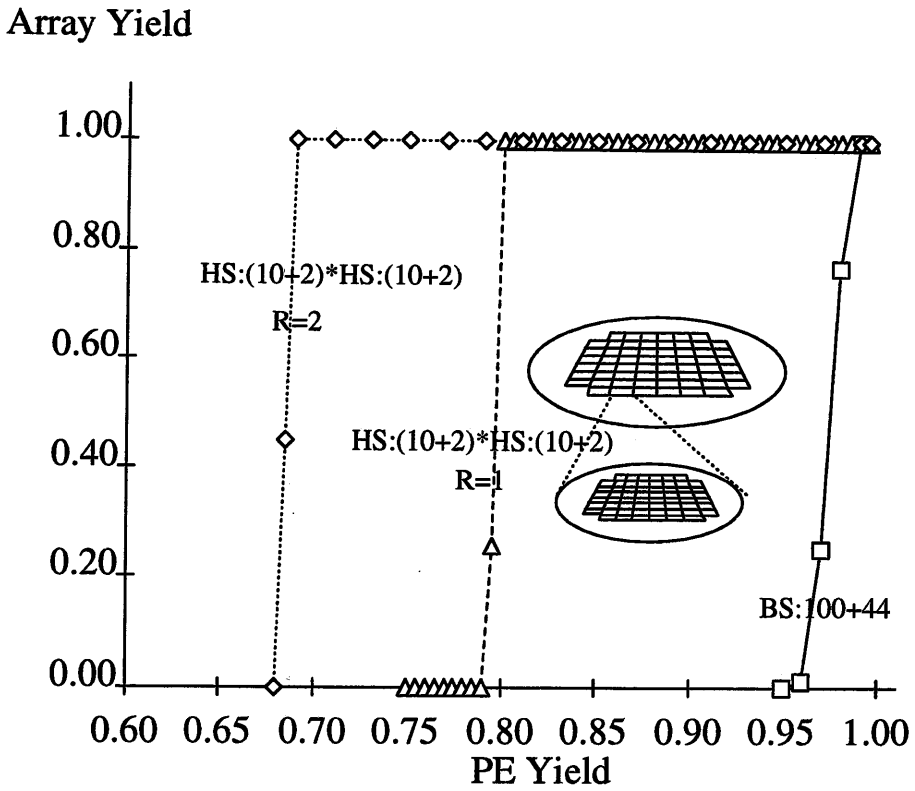


図 6.2 $N = 10$ の HS アレイを階層配置した場合の歩留り

本論文で考察する3次元格子結合アレイは，図6.3のように，$N \times N$の PE および，それを取り囲む R行または列の予備 PE からなる平面にインプリメントした格子結合型アーキテクチャを $N + 2R$枚積みあげたモデルで，2次元において歩留りの高かった HS 法の適用を考え，性能評価を行う．

定義 6.1 図6.3のように PE を3次元の格子点に配置し，立方体の面にあたる部分に予備の PE を R 行 R 列配置したものを2次元同様，$T\frac{1}{2}$トラック-R スペア型と呼ぶことにする．

PE は東西南北および上下に6つのポートを持ち，スイッチでトラックに接続されている．2次元アレイと同様，物理アドレスと論理アドレスを次のように定義する．

定義 6.2 PE に最下層の左上から行方向に対して，$[0,0,0],[1,0,0],...,[N+2R-1,N+2R-1,0]$，その上段に，$[0,0,1],...,[N+2R-1,N+2R-1,1]$ と物理アドレスをつけ，$[p_i,p_j,p_k]$ で現わす．2次元と同様に $(0,0,0),....,(N-1,N-1,N-1)$ と論理アドレスを

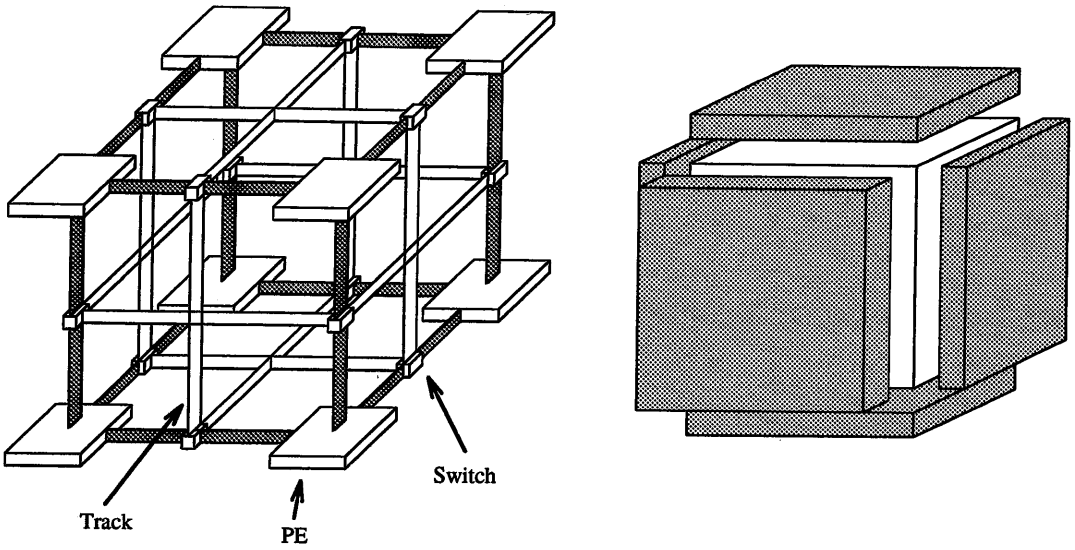


図 6.3　3 次元冗長化格子結合アレイ

つけ，$[l_i, l_j, l_k]$ であらわす．水平面に対しての方向を 2 次元同様 N, E, W, S であらわし，鉛直方向を U, D であらわす．

すなわち，この $(N+2R) \times (N+2R) \times (N+2R)$ 個の PE から，$N \times N \times N$個の PE を得ることが目標である．

6.3.1　PE の構造

2 次元同様，PE のステータスを

- アクティブ (**Active**)
- アイドル (**Idle**)
- バイパス (**Pass**)

で現わす．

2 次元同様，ポートが全て結合され，実際に使われている PE をアクティブな PE と呼ぶことにする．また逆に，全く使用されていない PE をアイドルな PE と呼ぶことにする．また故障のあるなしにかかわらず PE はデータをスルーされることが可能と仮定する．するとバイパスの状態としては，

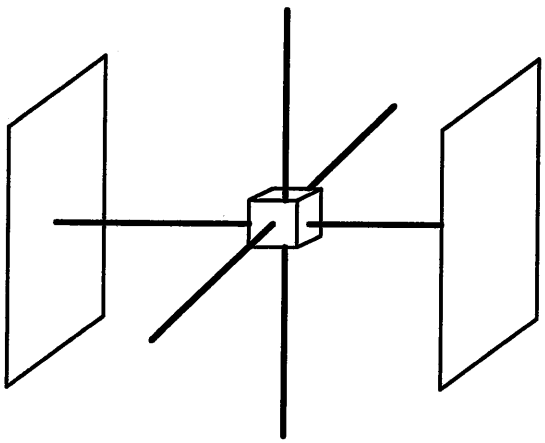


図6.4 トラックおよびスイッチの構造

- 南北方向のバイパス (**PassNS**)
- 東西方向のバイパス (**PassEW**)
- 上下方向のバイパス (**PassUD**)
- 東西南北方向のバイパス (**PassEWNS**)
- 東西上下方向のバイパス (**PassEWUD**)
- 南北上下方向のバイパス (**PassNSUD**)
- 南北上下方向のバイパス (**PassNSEWUD**)

の7通りが考えられる.

6.3.2 スイッチおよびトラックの構造

PEにはポートが6つありそれぞれトラックにスイッチを経て結合される. トラックはPEのNS, EW, UDそれぞれの方向に対して, 図6.4のように直交するように2本置く. すなわち, $T\frac{1}{2}$モデルでは各PE間に$2T$本のトラックが存在することになる. 2次元のときと同様, 東西, 南北, 上下のそれぞれのトラックの交点にはスイッチを置かない. また, PEと接続する直交点にスイッチを置き, データの流れを切り替えることができるようにする. スイッチは各PEをとりまくように6個存在し, 各PEのもつ6本のポートと接続される. スイッチには次のような条件を与える.

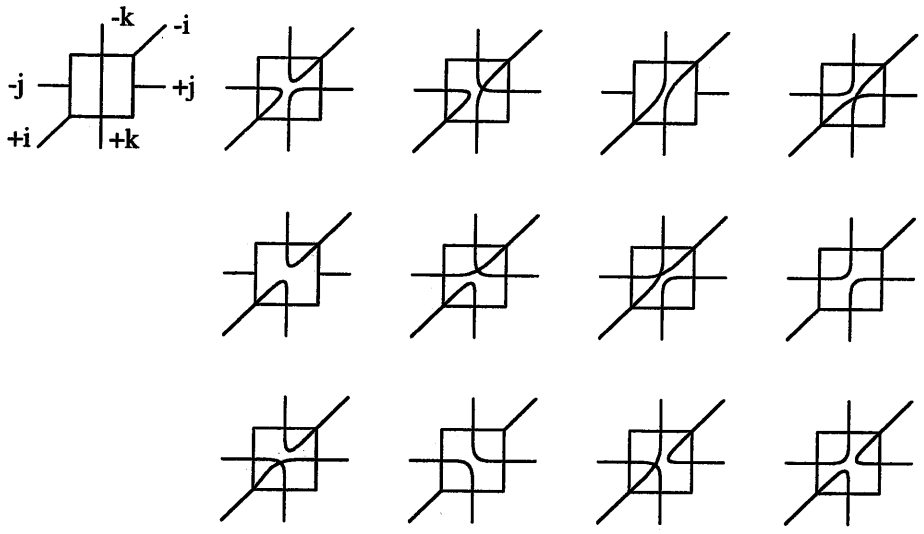


図 6.5 スイッチステータス

定義 6.3 UD 方向間にあるスイッチのステータスは次のような条件を満すようにしか選べない.

- N 方向と S 方向, または, E 方向と W 方向は結合しない.
- N と S, E と W 方向は結合できない.

これらの条件を満足するスイッチの状態は, 図 6.5に示すような 13 通りになる.

6.4 3 次元格子結合ネットワークの結合条件

2 次元のときと同様, 接続が正当であるためには次のことが成立する必要がある.

補題 6.1 トラック数を 1 とする. PE の接続が正当ならば, 物理アドレス $[p_i, p_j, p_k]$ の PE の E 方向のポートは $[i-1, j+1, k-1], [i, j+1, k-1], [i+1, j+1, k-1], ..., [i+1, j+1, k+1]$ の 9 つの PE としか接続できない.

証明 E方向のポートが $[i, j+1, k]$ と接続しているならば, 補題は成立するのでそれ以外の状態を考えればよい. E方向のポートがスイッチを経て北方向のトラックと接続しているとする. スイッチの定義から, 図 6.6の A のトラックは $[i+1, j+1, k]$ の PE に接続

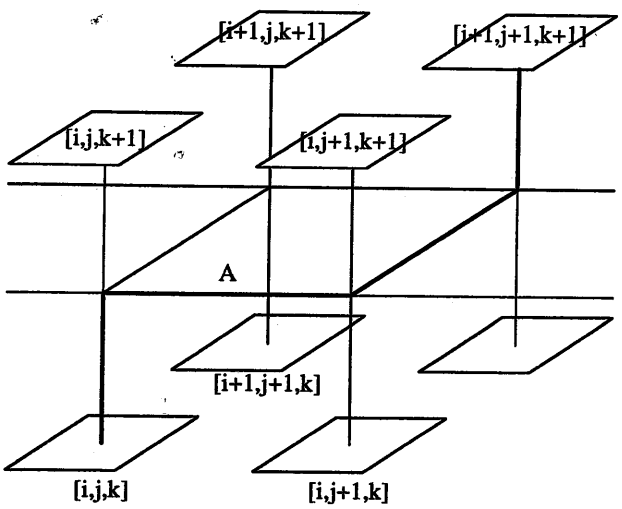


図 6.6　トラックの接続条件

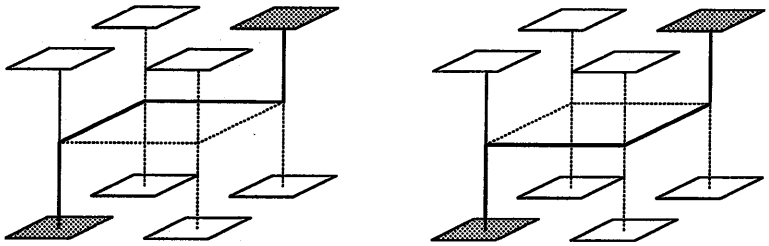


図 6.7　PE 間のルーティング

するか，上下方向のトラックに切り替えることしかできない．前者なら補題をみたす．後者の場合はどちらのトラックへ接続されていようとも，結局 $[i+1, j+1, k+1]$ もしくは $[i+1, j+1, k-1]$ と接続できない．同様の手法をあらゆる場合に適用することにより補題が成立することがわかる．□

補題 6.2 トラック数を T とする．PE の接続が正当ならば，PE の物理アドレスを $[p_i, p_j, p_k]$ の E 方向のポートは $[i-T, j+1, k-T], [i-T+1, j+1, k-T], [i-T+2, j+1, k-T], ..., [i+T, j+1, k+T]$ の $(2T+1)^2$ の PE としか接続できない．

証明　2 次元の場合同様，仮の PE があると考えるとあきらか．□

定義 6.4 補題 6.1，補題 6.2 より，E 方向のポートは，i 成分，k 成分がそれぞれ，$-1, 0, 1$ だけ違うところと接続している．この差の成分に注目し，PE$[i, j, k]$ の E 方向のポートの

状態を3次元のベクトル形式 HE $= (d_i, 0, d_k)$ とあらわす．他のポートについても同様である．

補題 6.3 ステータスが **Use** である任意の二つの PE，$P_1[p_{1i}, p_{1j}, p_{1k}]$, $P_2[p_{2i}, p_{2j}, p_{2k}]$ を考え，その論理アドレスを $L_1(l_{1i}, l_{1j}, l_{1k})$, $L_2(l_{2i}, l_{2j}, l_{2K})$ とすると以下が成立する．

$p_{1i} \leq p_{2i}$ならば $l_{1i} \leq l_{2i}$

$l_{1i} \leq l_{2i}$ならば $p_{1i} \leq p_{2i}$

不等号が逆の場合および j, kについても同様である．

証明 補題4.1と同様，PE の各ポートは i または j方向に増加または減少するしかないのであきらか．□

補題 6.4 $P[i,j,k]$ の北のポートと $P[i+1,j+1,k+1]$ の南のポートが接続しているとする．結合経路は図6.7のように2通りあるが，どちらを経由してもよい．

証明 図6.7の A を使用しないと不可能な結合は $[i,j+1,k]$ と $[i,j,k+1]$ であるが，これは論理アドレスと物理アドレスの大小関係が補題6.3に反する．同様にして，$[i,j+1,k]$ と $[i+1,j,k]$ なども否定される．すなわち，どちらの経路を選んでも他の結合経路を塞ぐことはない．□

6.5 3次元再帰的シフト

2次元と同様に次のように3次元再帰的シフトの戦略を示す．

1) $[i,j,k]$ の PE が故障していてかつ **Active** とする．この PE はネットワークから切り離されねばならない．そこで，その PE の代りを $[i',j',k']$ に代行させようとする．この $[i',j',k']$ が発見できなければ **False** を返しこのアルゴリズムは終了する．

2) $[i',j',k']$ の PE が **Active** でなければ $[i,j,k]$ の PE の代行を行うことができる．そこで，**True** を戻り値として返す．そうでなければ，$[i',j']$ は代行できない．そこで，$[i'',j'',k'']$ に代行を要求する．(この呼び出しは再帰的に行われる)

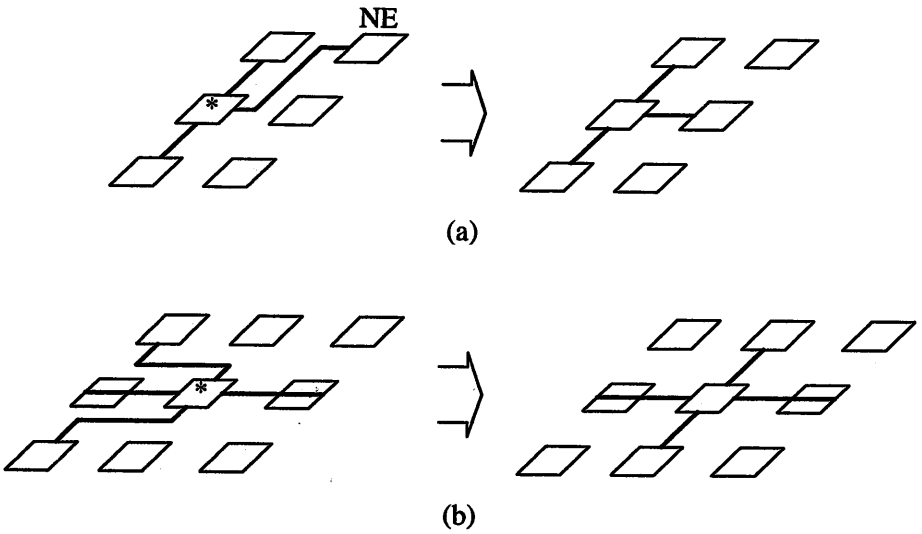


図 6.8　3 次元 HS 法

3) $[i', j', k']$ から **True** を受けとった PE$[i, j, k]$ は，ポートの結合を変化させ $[i', j', k']$ に繋ぎ換え，自分自身を **Active** 以外の状態にする．

2 次元同様，本研究ではこの $[i', j', k']$ を隣接した 6 つの PE とする．自律再構成のアルゴリズムの戦略として，2 次元同様，再帰的シフトを用いる自律再構成アルゴリズムの戦略を以下に示す．

1) 初期状態として仮配置を行う．

2) 全ての PE を順番にチェックする．

3) PE が **Active** でかつ故障しているならば方向を決定し，再帰的シフトを実行する．

4) 全ての PE の故障が補償されたならば終了する．そうでなければ 2) へ行く．

6.6　3 次元 HS 法

本研究では，3 次元における再構成法として 2 次元で評価が高かった HS 法を用いた．2 次元の場合と同様，中央部に仮の配置を行う．また，ループの上限を 10000 回とし，この数以上にシフト手続きが発生した場合は再構成を中止することにする．ここでは 2 次元同様東方向にシフトする場合を説明する．

step1)

PEが**Idle**あるいは，**PassEW**状態であるときはシフトは不要である．よって直ちに**True**を返し終了する．それ以外の状態で，これ以上東にシフト不可能な場合は**False**を返し終了する．

step2)

PEの状態が**PassNS**や**PassUD**以外のときは，PEはHEの状態をチェックする，もしHE = (0,1,0)でなければHE = (0,1,0)になるようそのPEにシフトシグナルを送る．例えば図6.8(a)の例ではPE(NE)に南方向へのシフトシグナルを送る．

step3)

HN, HS, HU, HDをそれぞれチェックする．例えばトラックが1のとき，HN = −1であればトラック数が1であるため東方向へシフトすることができない．そこでこの場合PE(NW)に東方向へのシフトシグナルを送る．(図6.8(b)) トラック数がTならば，HN = $-T$のときにシグナルを送る．

step4)

PE(E)に東へのシフトシグナルを送る．

step5)

再結合できるパターンになっていたならポートをつなぎ換える(再構成)．

HS法はこのように若干の変更だけで3次元に適用することができる．

6.7 3次元格子結合マルチプロセッサの再構成法の性能評価

以上のアルゴリズムを用いてシミュレーションを行った．図6.9は$R = 1$としアレイサイズを変化させ，故障PEの数を変化させ再構成確率を求めている．試行回数は各故障数に対して500回である．横軸にPE 1つ当りの歩留り，縦軸にアレイの歩留りを記している．

この図からわかる通り，従来の方式程度の冗長化回路で同程度の歩留まりが得られることがわかる．これは本アルゴリズムが3次元アーキテクチャに対しても有効に働くことを示している．

しかし，付加する冗長PEの数には大きな差がある．$(8+2)^3$すなわち1000個のPEから512個のPEを得ることになり，全PEの半分近くが冗長なPEになっている．これに

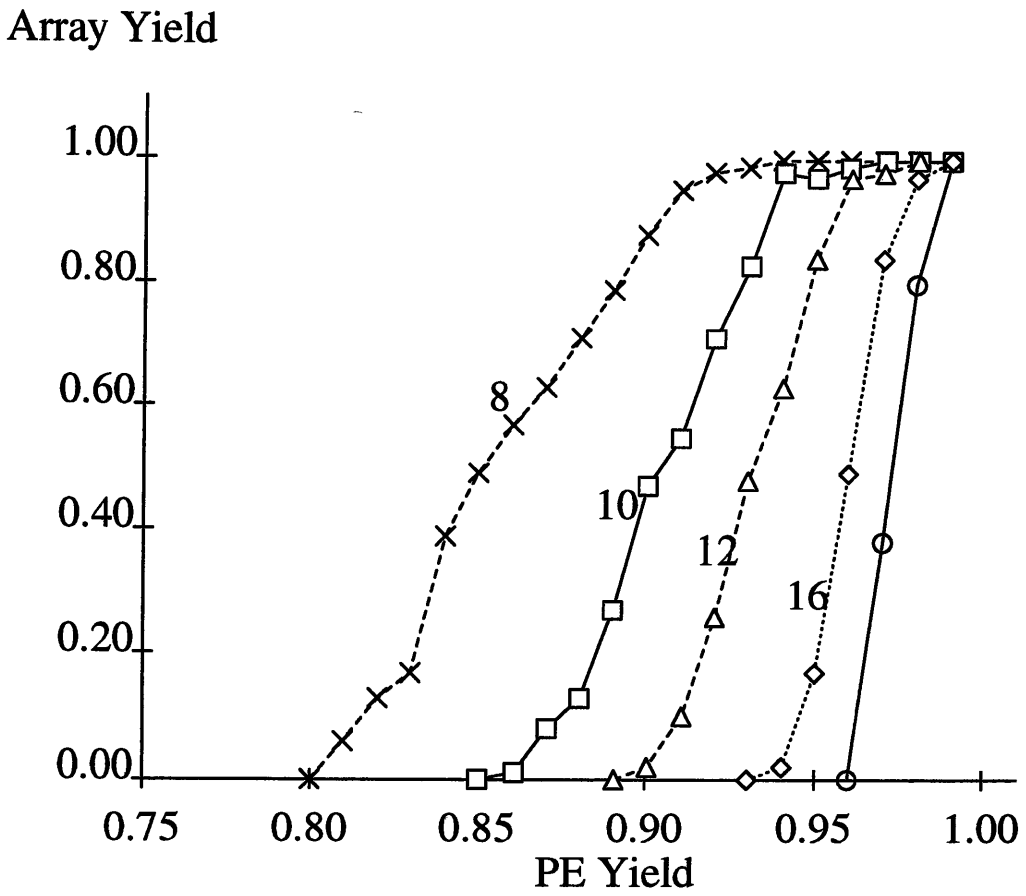


図 6.9 $(N+2)^3$アレイの歩留り

対して，2次元では，同じ400～500程度のPEを得る場合，例えば，$(22+2)^2$のPEの576個のPEから484個のPE得ることで実現でき，この場合冗長なPEは92個であり全体の16%程度が冗長PEに過ぎない．3次元ウェーハの場合はアレイサイズが3乗のオーダで効いてくるので，冗長PEの割合が2次元にくらべ大きくなるという問題がある．

6.8 むすび

小さな冗長化アレイを階層化し，大規模アレイを構築する手法，従来の平面アーキテクチャを拡張して3次元アーキテクチャにした場合のアルゴリズムおよびその適応について述べた．

同じサイズの，HS法で階層化を施したものと，BS法で非階層化のものを比較し，階層化が大規模アレイに向いていることを示した．

2次元を拡張した3次元アーキテクチャでトラック，スイッチの接続条件を求め，2次元のHS法の概念を適用し，アレイ歩留まりを求めた．

しかし，3次元アレイの歩留まりは，まだ低いと考えられる．そこで，従来のような1トラック型ではなく2トラック型へ拡張を行うことによりさらに歩留まりが向上すると思われる．

しかし，3次元アーキテクチャで，これを実現するのにはより多くの冗長化回路を必要とする．そのため，パラメータを換え，シミュレーションをより細かく行ない，より定量的な結果を出す必要があると思われる．

第 7 章

結論

7.1 まえがき

本研究の目的は，2次元および3次元格子結合ネットワークにおける，再構成アルゴリズムの確立であった．2章で述べた本研究の特色，すなわち，

- Kung と同等のアーキテクチャで再構成可能である．
- 4 方向への補償以外の補償も考える．
- 冗長 PE の位置や数に制限を加えない．
- グローバル情報を必要としない．(ローカルな情報だけで自律再構成可能である)

がどのように実現されているか，および本研究の成果および今後の構想を述べる．

7.2 本研究の成果

- 再構成率の改善
  Kung らにより提唱された $1\frac{1}{2}$ トラック 1 スペアモデルにおいて大きく再構成率を高める手法をいくつか提言した．HS 法は再構成率が高いが，BS 法に比べるとハードウェア量が BS 法に比べ若干多い．これらを定量的に評価するには PE，トラック，スイッチの占める面積を考慮する必要がある．これは今後の課題である．さらにこの手法は Varvarigou らが提唱した 3 トラックを用いるものとほぼ同等の再構成率を得ることができることがわかった．同等またはそれ以下の冗長量で，高い再構成率

を得ていることになる．これは，超並列コンピュータを実現する際に非常に有利である．

- 補償範囲の拡大

  Kungらの手法ではPEは4方向のいずれかにしかシフトが可能でなく，PEの存在できる範囲は5箇所に限られている．これは，$1\frac{1}{2}$の資源を有効に利用していない．本研究で述べた再帰的シフトはPEの物理アドレスに依存しないので，冗長PEが$2R$あれば，$(2R)^2$箇所に移動することが可能である．故に高い再構成率を得ることができる．

- 冗長PEの制限

  本研究で述べた手法は冗長PEがいくつあっても構わない．また，HS法BS法は冗長PEがどこにあっても構わない．すなわち初期配置はどうなっていてもよい．しかし，補償パス法で周辺に冗長PEがなければいけなかった．これはただちに図7.1に示すような冗長PEの配置が可能であることを示す．このようにすることによりさらなる巨大アレイの構築が容易になると考えられる．これの応用としてある1つのPEの移動できる範囲をあらかじめ決定しておくことにより，トーラス型への拡張も可能である．

- ローカル情報の利用

  本研究で提案したバイパスおよび再帰的シフトは全て近隣情報のみで行われておりグローバル情報は一切不要である．

- 階層化冗長化構成法

  階層化を施すことにより，10^4規模の超並列マシンの再構成アーキテクチャを構築することができることが示せた．さらに非階層化のモデルと比較し階層化が重要であることを示した．

- 3次元への拡張

  HS法やBS法は3次元へそのまま応用できる．2次元で非故障PEをもシフトする本手法は3次元においても有効であると考えられる．

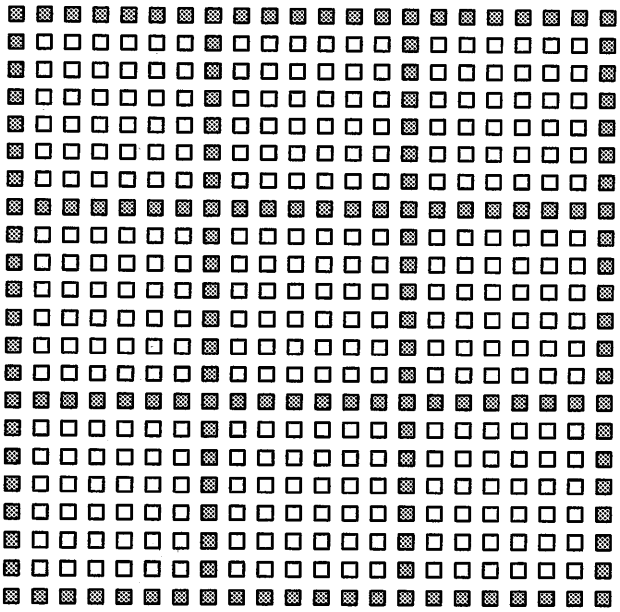


図 7.1 冗長 PE を分散させた冗長化格子結合型アレイ

7.3 今後の構想

2 次元および 3 次元の格子結合ネットワークの再構成問題に取り組んで来た．しかし，3 次元の分野はまだ 2 次元ほど研究されていない．本手法での解析は簡略化した HS 法を応用しただけにすぎず，より定量的な解析が必要である．またさらにその上を行く多次元へのアプローチも行っていきたい．例えば，近年注目されているネットワークの 1 つであるハイパーキューブなどに応用できる可能性がある．

今回はシフトの方向を 4 方向に限定しているが，トラック数を増すことにより斜めへのシフトも可能となる．現在までに提案されている手法のほとんどは非故障 PE のシフトを考えておらず，その点をも考えた他の再構成モデルへの適用も考えていくことが必要である．

さらに，可能であれば，シフトの方向を決定づける要因をつきとめ，非決定アルゴリズムでない高速な手法を開発していきたい．またそうでないならば，不可能であることを理論的に解明していく必要があるであろう．

7.4 むすび

フォールトトレランス技術の研究は昔から行われているが，格子結合ネットワークの再構成技術が論じられ始めたのは30年前にWSIの構想が打ち出されてからである．その後，LSI技術の発展によりWSI構想は一時途絶えたが，近年また注目されている．

その中で格子結合ネットワークの再構成法として，Kungらの手法が寄与したものは大きい．その問題は解決されたかに見えたが，Roychowdhuryら[10]により，その問題点が指摘された．そして，その問題の単純さに比べ，具体的な解決法は現在まで提案されていなかった．

本研究では非決定アルゴリズムを用いたり，シフトの方向を限定するなどして，比較的現実的な時間で解を求めるアルゴリズムを提言した．しかし，未だ完全な解決法は求まっていない．

謝辞

本研究を行うにあたり，御指導，御助言を戴いた東北大学工学部 阿曽弘具教授，成富敬博士，東北大学情報処理教育センター 大町真一郎博士，東北大学大型計算機センター 孫寧博士に感謝いたします．また，本論文をまとめるに際し貴重な御意見を頂いた，西関隆夫教授，亀山充隆教授に深く感謝いたします．

また，北陸先端科学技術大学院大学特別研究生としての間，ご指導頂いた北陸先端科学技術大学院大学 木村正行教授，堀口進教授，阿部亨助教授，下平博助教授らに心から感謝します．

また，半導体の現状についてコメントいただいた，日本電気株式会社ＬＳＩ事業本部 森健彦氏，松下電気株式会社半導体研究センター 福田大氏，研究のため環境整備に携わっていただいた，松下通信工業情報システム事業部 宮下重博氏，ＮＴＴ情報システム本部 粟津辰功氏，研究についての相談に懇切丁寧に応対してくださった，東京大学工学部電気工学科 白石知之氏，株式会社富士通研究所 和田祐二氏，大阪大学基礎工学部情報工学科 松下誠氏，奈良先端科学技術大学院大学情報科学研究科 城和貴氏らに感謝いたします．

さらに，活発な討論をしていただき，御助言を下さった北陸先端科学技術大学院大学情報科学研究科の井口寧氏，山森一人氏，當山孝義氏，村田真一郎氏，岩田英明氏，山田隆弘氏はじめ，堀口研究室，阿部研究室の皆様および東北大学工学部情報工学科の後藤英明氏および阿曽研究室の皆様に感謝いたします．

これからも不惜身命の思いで研究に取り組んでいきたいと思うとともに，計算機環境の整備に携わってくれた方々を始め研究環境を礎から支えてくださった多くの方々に厚く御礼を申しあげます．

参考文献

[1] Michael L Campbell, Scott T Toborg, and Scott L Taylor. 3-d wafer stack nuero-computing. *IEEE Int'l Conf. on Wafer Scale Integration*, pp. 66–74, 1993.

[2] M Chean and J A B Fortes. Fuss: A reconfiguration scheme for fault-tolerant processor arrays. *Int'l Workshop Hardware Fault Tolerance in Multiprocessors*, pp. 30–32, June 1989.

[3] M chean and Jose A B Fortes. A taxonomy of reconfiguration techniques for fault-tolerant processor arryas. *Computer*, Vol. 23, No. 1, pp. 55–69, 1990.

[4] J S N Jean, H C Fu, and S Y Kung. Yield enhancement for wsi array processors using two-and-half-track switches. *IEEE Int'l Conf. on Wafer Scale Integration*, pp. 243–250, January 1990.

[5] S Y Kung, S N Jean, and C W Chan. Fault - tolerant array processors using single - track switches. *IEEE Trans. Computers*, Vol. 38, No. 4, April 1989.

[6] S Y Kuo and W K Fuch. Efficient spare allocation for reconfigurable arrays. *IEEE Design and Test*, pp. 24–31, February 1987.

[7] Stephen Y H Su Minghsien Wang, Michal Cutler. Reconfiguration of vlsi/wlsi mesh array processors with two-level redundancy. *IEEE Trans. Comput*, Vol. 38, No. 4, April 1989.

[8] R Negrini and R Stefanelli. Algorithms for self-reconfiguration of wafer-scale regular arrays. *Proc. ICCAS, IEEE, Beijing*, 1985.

[9] R Negrini and R Stefanelli. Time redundancy in wsi arrays of processing elements. *Proc. 1st Int. Conf. on Supercomputing Systems, St.Petersburg*, December 1985.

[10] V P Roychowdhury, J Bruck, and T Kailath. Efficient algorithms for reconfiguration in vlsi/wsi arrays. *IEEE Trans. Computers*, Vol. 39, No. 4, April 1990.

[11] A D Singh. Interstitial redundancy: An area efficient fault-tolerance scheme for large area vlsi processor arrays. *IEEE Trans. Computers*, Vol. 34, pp. 448–461, May 1985.

[12] T A Varvarigou, V P Roychowdhury, and T Kailath. A polynomial time algorithm for reconfiguring multiple-track models. *IEEE Trans. Computers*, Vol. 42, No. 4, April 1993.

[13] T A Varvarigou, V P Roychowdhury, and T Kailath. Reconfiguring processor arrays using multiple-track models: The 3-track-1-spare approach. *IEEE Trans. Computers*, Vol. 42, No. 11, November 1993.

[14] R J Wojnarowski, R A Fillion, B Gorowitz, and R Saia. Three dimensional hybrid wafer scale integration using the ge high density interconnect technology. *IEEE Int'l Conf. on Wafer Scale Integration*, pp. 309–317, 1993.

[15] 沼田一成, 堀口進. 格子型結合マルチプロセッサの再構成アーキテクチャ. 信学会技報, Vol. **CPSY91-67**, , January 1992.

[16] 沼田一成, 堀口進, 木村正行. メッシュ結合ネットワークフォールトトレランスアーキテクチャ. 情報処理学会全国大会, Vol. 4Q-11, pp. 87–88, October 1991.

[17] 馬場敬信. 超並列マシンへの道. 情報処理, Vol. 32, No. 4, pp. 348–364, April 1991.

[18] 高浪五男, 久長穣, 井上克司. 周辺に予備を持つメッシュ結合プロセッサ配列の再構成のための結合切り替え. 信学会技報, Vol. **WSI92-7**, , August 1992.

[19] 小柳滋. 超並列マシンの実現技術. 情報処理, Vol. 32, No. 4, pp. 365–376, April 1991.

[20] 堀口進. ウェハ規模超密度集積回路について. *Hybrids*, Vol. 6, No. 1, pp. 16–21, 1990.

[21] 堀口進. Wsi デバイスの研究開発動向. 電子材料, Vol. 30, No. 5, pp. 16–23, 1991.

[22] 渡部徹, 高浪五男, 渡辺孝博. 4 辺に予備を持つ格子状結合高並列計算機の再構成法のニューラルネット解法. 信学会技報, Vol. **WSIA94-4**, , March 1994.

[23] 身次茂. Fpga の現状と将来. 情報処理, Vol. 35, No. 6, pp. 505–510, June 1994.

公表目録

発表論文

[1] S.Horiguchi, I.Numata, M.Kimura,
"Self-Reconfigurable Algolithm Of WSI Sorting Network",
IEEE International Conference on Wafer Scale Integration pp.249-255 (Jan. 1991)

[2] 沼田一成, 堀口進,
"格子結合型マルチプロセッサシステムの自律再構成法",
電子情報通信学会論文誌 (D-I), J76-D-I, pp.531-540, (Oct. 1993)

[3] 沼田一成, 堀口進,
"格子結合型マルチプロセッサシステムの WSI 構成法",
電子情報通信学会論文誌 (D-I), J77-D-I, pp.121-129, (Feb. 1994)

[4] S.Horiguchi, I.Numata,
"Self-Reconfiguration Architecture for Mesh Arrays",
IEEE International Workshop on Defect and Fault Tolerance on VLSI Systems pp.212-220 (Oct. 1994)

[5] 沼田一成, 堀口進,
"再帰的シフトを用いた格子結合型マルチプロセッサの再構成法",
電子情報通信学会論文誌 (D-I), 寄稿中

学会発表等

[1] 沼田一成, 堀口進, 木村正行,
"フォールトトレランスソーティングネットワーク",

平成 2 年度電気関係学会東北支部連合大会, 1H9, (Aug. 1990)

[2] 沼田一成, 堀口進, 木村正行,
"フォールトトレランスソーティングネットワークアーキテクチャ",
情報処理学会東北支部研究会, 2-1-5, (Sep. 1990)

[3] 沼田一成, 堀口進, 木村正行,
"WSI ソーティングネットワークのフォールトトレランス性能",
情報処理学会東北支部研究会, 3-1-3, (May. 1991)

[4] 沼田一成, 堀口進, 木村正行,
"メッシュ結合ネットワークフォールトトレランスアーキテクチャ",
情報処理学会全国大会, 4Q-11, (Oct. 1991)

[5] 沼田一成, 堀口進, 木村正行,
"フォールトトレランスメッシュ結合ネットワークの評価",
情報処理学会東北支部研究会, 3-2-3, (Dec. 1991)

[6] 沼田一成, 堀口進,
"格子結合型マルチプロセッサの再構成アーキテクチャ",
電子情報通信学会技術研究報告, CPSY91-6, (Jan. 1992)

[7] 沼田一成, 堀口進,
"格子結合型マルチプロセッサの再構成法",
ウェーハスケール集積システム研究会, WSI92-6, (Aug. 1992)

[8] 沼田一成, 堀口進,
"格子結合型マルチプロセッサシステムの自律再構成アルゴリズム",
ウェーハスケール集積システム研究会, WSI93-1, (Jun. 1993)

[9] 沼田一成, 堀口進,
"非決定アルゴリズムを用いた格子結合型ネットワークの再構成アルゴリズム",
情報処理学会全国大会, 1T-8, (Oct. 1993)

[10] 沼田一成, 堀口進,
"ウェーハスタック構造格子結合型マルチプロセッサの再構成法",
ウェーハスケール集積システム研究会, WSI93-11, (Nov. 1993)

[11] 沼田一成, 堀口進,
"補償パスを用いない格子結合ネットワークの再結合法", 情報処理学会全国大会, 2L-5, (Oct. 1994)