JAIST Repository
https://dspace.jaist.ac.jp/

Title	FPGA利用を前提とした最小リンク付加による相互結合網の耐故障化に関する研究
Author(s)	松本，英樹
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1011
Rights	
Description	Supervisor:横田 治夫，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

FPGA 利用を前提とした最小リンク付加による相互結合網の耐故障化に関する研究

指導教官　横田 治夫 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

松本 英樹

1997 年 2 月 14 日

Copyright © 1997 by Hideki Matumoto

目 次

1 序論 **3**

1.1 本研究の背景と目的 3

1.2 本論文の構成 4

2 従来の相互結合網のリンク故障対策 **6**

2.1 動的網 (多段結合網) におけるリンク故障対策 6

2.2 静的網におけるリンク故障対策 7

3 FPGA を用いた静的相互結合網の特徴 **10**

3.1 FPGA の一般的な特徴 10

3.2 ネットワーク構成の柔軟性 11

3.3 書き換えによる耐故障性 12

3.3.1 代替リンクの利用によるリンク故障対策 12

3.3.2 ルーティング/二重化との違い 13

4 FPGA の書き換えによるリンク故障復旧戦略 **14**

4.1 前提とする構成 14

4.1.1 対象とするシステムモデル 14

4.1.2 対象とするリンク故障モデル 16

4.2 リンク故障の検出方法 17

4.2.1 イニシャルな故障の検出 17

4.2.2 動作中に発生した故障の検出 18

4.2.3 データ損失回避 19

4.3 リンク故障復旧戦略 19

4.3.1 集中故障復旧方法 20

4.3.2 分散故障復旧方法 20

4.4 ノードの書き換え 22
4.4.1 各ノードの書き換えを一回に限った場合 22
4.4.2 各ノードの書き換えを複数回許した場合 23

5 代替パスのパターンとその評価 25
5.1 代替パスパターンの生成 25
5.1.1 ネットワークサイズに対する代替リンク数の制約 25
5.1.2 パターンの絞り込み 26
5.2 評価の指標 27
5.2.1 評価の方針 27
5.2.2 故障に対する影響リンク数 27
5.2.3 最大復旧可能リンク数 28

6 具体的な相互結合網における代替パスパターンの評価 30
6.1 二次元 4 隣接トーラス (書き換え一回) 30
6.1.1 乱数によるパターン生成 30
6.1.2 パターンの分類とその考察 34
6.1.3 制限付きパターン生成 38
6.1.4 ネットワークサイズの違いについて 43
6.2 二次元 4 隣接トーラスでの複数回書き換えに対する考察 46

7 考察 47
7.1 信頼性に関する考察 47
7.1.1 信頼性モデル 47
7.1.2 集中故障復旧方式を前提とした場合の信頼性 48
7.1.3 分散故障復旧方式を前提とした場合の信頼性 51
7.2 その他の静的相互結合網に対する考察 53

8 結論 55
8.1 本研究の成果 55
8.2 今後の課題 56

第 1 章

序論

1.1 本研究の背景と目的

近年、インターネットが爆発的に普及し、またマルチメディア化が進む中で、画像や音声といった膨大なデータを計算機で扱う研究が盛んに行なわれるようになり、要求される処理能力は増加の一途をたどっている。

扱うデータ量の増加に伴い高い処理能力が要求され、並列コンピュータにおける相互結合ネットワークの規模は益々拡大している。ネットワークを構成するノードプロセッサやリンク等のハードウェアも増大し、システムの信頼性の面から見れば、これらが故障した場合にもシステムとして正常に動作できる Fault Tolerant Design (耐故障設計) 問題は非常に重要になってくる。プロセッサ技術の飛躍的な進歩によってプロセッサ自身の信頼性は高くなってきたが、プロセッサ間を接続するリンクに関してはコネクタの接触不良など以前としてハードウェア的な故障が多い。

実際にリンクに故障が発生した場合のフォールトトレランス手法としては、各リンクに冗長なリンクを 1 本ずつ追加し、リンク故障時にそれを代替リンクとして使用するというリンクの二重化手法や、故障リンクを迂回しながら目的ノードまでの経路選択を行なうというルーティング手法が一般的である。しかし二重化手法の場合は普段は使わないリンクを追加するため、ハードウェア的なコストは高い。ルーティング手法の場合はアルゴリズムにもよるが、故障リンクを回避するために毎回経路選択を行なうため時間的なコストは高い。また故障回避によって他のリンクやプロセッサの通信負荷は大きくなってしまう。そこで我々はより効果的なリンク故障復旧策を模索する上で FPGA(Field Programmable Gate Array) の持つ「回路の書き換え可能」というユニークな性質に着目した。[11]

FPGA はユーザ自身がデバイスに設計データを書き込むことにより回路を作ることのできる LSI であり、ここ数年で大幅に普及してきた。最近では使用可能ゲート数が 125000 ゲートの FPGA が発売され、大規模、高速なロジック回路の実装が可能になってきた。FPGA の大きな特徴は、何度でも消去/再書き込みが

可能であるという点である。この性質を用いればハードウェアに手を加えずに回路を変更することができる。

FPGA をノードプロセッサとしてネットワークを構成した場合、FPGA に回路データをダウンロードするためのコンフィギュレーションラインはダウンロード後は普通に入出力可能な I/O として用いることができる。従ってリンクに故障が発生した時は故障リンクへの接続をこのラインに繋ぎ換えるるように内部回路を書き換えることでこのラインを代替パスとして用いる。また、コンフィギュレーションラインも故障すると考えた時にその発生箇所によっては復旧処理を行なえない場合がある。この対策として全てのノードに ROM を接続し、データのコンフィギュレーションに必要な処理内容をあらかじめロードしておくことによって、リンク故障検出時に各ノードがローカルに故障復旧できるようなシステムを提案する。このシステムによってコンフィギュレーションのためのラインは必要なくなり、ラインをリンク故障を復旧するための代替パスとしてだけ使用することができる。

これらの手法による特徴は、復旧後ノード間の物理的な接続が失われていないので正常に対する通信性能の低下がないことである。またこの手法はウェーハースケールインテグレーションにも適用可能と思われ、その応用範囲は広い。

ここでリンクの多重故障を考えた場合、代替パスの張り方によって復旧できるリンク故障の最大数は変化し、また同じパスパターンであっても故障するリンクの位置次第で復旧できる故障数は変化する。

そこで本研究においては代替パスの最適な張り方について検討し、そのパスパターンにおけるシステムの信頼性を検討する。信頼性の面では二重化手法との MTTF の比較を行ない、リンクコストを落とした場合に信頼性をどれだけ保てるのか明らかにする。

ここでは本研究室で提案されている DR-net[8, 9, 10] のネットワーク構成に合わせて、n × n のトーラス結合ネットワークを前提に話を展開する。

1.2 本論文の構成

第 1 章では、本研究の背景と目的、及び本論文の構成について述べる。

第 2 章では、従来の相互結合網においてリンク故障対策はどのようにして行なわれてきたのか、またどのような所に問題があるのかについて説明する。

第 3 章では、静的相互結合網のノードプロセッサに FPGA を用いる必然性を FPGA 独自の特徴を元に明らかにしていく。

第 4 章では、FPGA の書き換えによってリンク故障復旧を行なう時にリンク故障どのようにして検出するのか、そして具体的にどのようにして書き換えを行なうのかについて集中、分散、2 つの故障復旧方法に大別して詳しく説明する。

第 5 章では、代替パスパターンを生成する上での制約について述べる。また、生成した代替リンクパターンからどのようにして最適なパスパターンへと絞り込んででいくのか、その評価基準について述べる。

第 6 章では、実際に代替パスパターンを生成し、得られた多数のパターンから評価指標にを用いてパターンの絞り込みを行なう。

第 7 章では、本研究で得られた最適と思われるパスパターンについてマルコフモデルに基づき信頼度 ($MTTF$) を算出し、その他の静的相互結合網と信頼性の比較を行なう。

第 8 章では、本研究で得られた成果と、今後の課題を述べる。

FPGA 実験システムへの実装を検討する。

第 2 章

従来の相互結合網のリンク故障対策

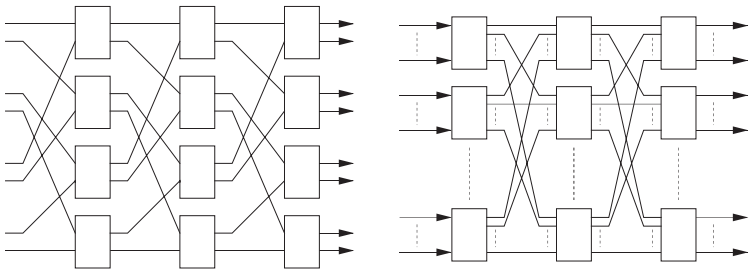
ここではこれまでに良く用いられているリンク故障対策方法について概念的に説明し、そのどこに問題があるのかを言及する。

2.1 動的網 (多段結合網) におけるリンク故障対策

動的ネットワークは、ネットワーク中のアクティブスイッチを切り替えることにより接続を変えることができるのが特徴であり、シングルステージネットワークとマルチステージネットワークの 2 種に大別できる。

シングルステージネットワークにおけるシャッフルエクスチェンジネットワークや、マルチステージにおけるオメガネットワーク (図 2.1(a)) では、入力ターミナルから出力ターミナルまでスイッチを切り替えながら到達できるが、ソースからデスティネーションまでの経路が一通りしか存在しないため、経路のどこかで故障が発生した場合にルーティングによる故障回避が不可能となる。こういった場合の一般的なリンク故障対策としては、冗長な経路を設けておき、リンク故障が発生した場合にスイッチをその経路へと繋ぎ変えることにより復旧を行う。しかしこの冗長なパスは普段は使用されず、また冗長なパスを追加することによってスイッチボックスもそれに対応してポートを追加し切り替えを行なわなければならないため、ハードウェア的に見ればコスト高になってしまう。

ネットワークが閉そくすることなくつねにすべての入出力ターミナル間のパスが確立できるようなクロスネットワーク (図 2.1(b)) に代表される非閉そくネットワークでは、ある入力ターミナルから出力ターミナルへの経路が必ず複数本以上存在するので、ある経路が故障してもスイッチを切り替えながらソースターミナルまでルーティングすることにより、リンク故障を迂回して目的ターミナルへ到達することができる。しかし複数のリンク故障を扱う場合リンクが故障する度にパスが減少しネットワークは閉そくしていく。それによってあるノードや経路に負荷が集中し、通信オーバヘッドが増大してしまう。

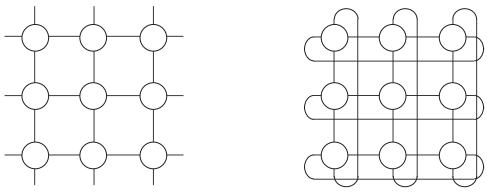


(a) 8×8オメガネットワーク　　(b) クロスネットワーク

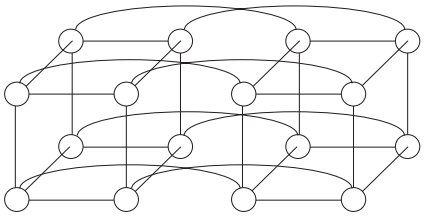
図 2.1: マルチステージネットワーク

2.2 静的網におけるリンク故障対策

静的ネットワークは、プロセッサ間のリンクが固定されていて一定のプロセッサにしか直接接続できないネットワークである。その代表的なものとしては、2 次元ではメッシュ(図 2.2(a))、本研究でも扱うトーラス (図 2.2(b)) などがある。また 4 以上の次元をもつネットワークでは n 次元のハイパーキューブ (図 2.2) があり、INTEL IPSC、FPS T series、Ncube、コネクションマシンなど、従来から多くのマシンで用いられている。



(a) ニアネイバメッシュ　　(b) トーラス



(c) 4次元ハイパーキューブ

図 2.2: 各種の静的ネットワーク

このネットワークにおける故障対策としてはリンクの二重化手法、ルーティング手法が一般的である。リンクの二重化手法は各プロセッサ間に冗長なリンクを一本付加し、リンク故障時にはそのリンクを使用する手法である。しかしこの手法の場合、通常のシステムのリンク本数を n とすると、全てのプロセッサ間に

余分なリンクがn本付加されるため総リンク数は2n本となり、通常のリンク数に対して2倍のコストがかかることになる。

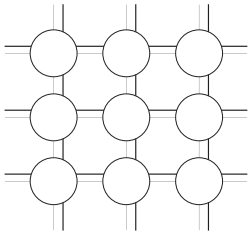


図 2.3: リンクの二重化

ルーティング手法は2.1でも述べたように故障リンクを迂回しながら目的ノードへ到達するという手法であり、本研究室で提案されたDR-net[8, 9, 10] にも採用されている。DR-net 実験システムは5 × 5 のトーラスネットワーク構成となっており、ノードプロセッサとしてTransputerを用いているため、リンク故障対策はソフトウェアによる処理で行なわれている。DR-net の通信プロセッサ間では、ストライピングされたデータに目的ノードアドレス、中継ノードアドレス等の付加情報を加えたデータパケットとして相互通信を行なっている。リンク故障検出後は動的にこの中継ノードアドレスを指定し、その故障リンクを通らないように目的ノードへパケットを配送する。故障リンクの回避例を図2.4に示す。これはリンク故障が1本、2本、3本の時、それぞれ適切な中継ノードを指定することで故障リンクを回避することができることを示している。

しかしこの手法においてもデータの通過経路に偏りが生じたり、通信ノードやリンクへの負荷が増加するなど通信オーバヘッドが大きくなってしまうためあまり好ましい方法とは言えない。

一般的にルーティング手法は複雑なアルゴリズムを要する場合が多く、経路探索に時間がかかるなど、問題も多い。

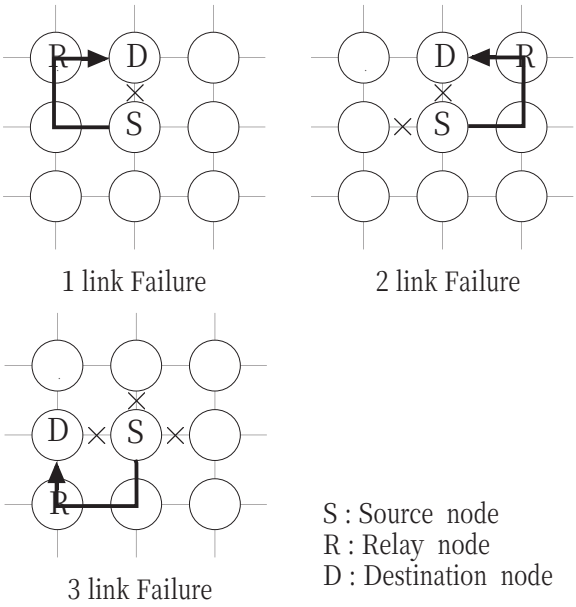


図 2.4: 中継ノード指定によるリンク故障の回避

第 3 章

FPGA を用いた静的相互結合網の特徴

この章では FPGA をノードプロセッサとして相互結合網を構成した場合に従来の相互結合網に比べてどこにオリジナリティがあるのか、またどのように優れているのかについて、FPGA のユニークな特徴を元に明らかにする。

3.1 FPGA の一般的な特徴

大規模、高速なロジック回路を開発する場合に、FPGA(Field Programmable Gate Array) はきわめて有力な選択肢となっている。大規模な回路を FPGA 内に収めることにより以下のような利益が得られる。

- コストダウン
- 動作の高速化
- 消費電力の低減
- 実装面積の縮小
- 部品点数の削減
- 設計能率の向上

FPGA はユーザ自身がデバイスに設計データを書き込むことにより回路を作る。設計から製作までユーザの手元で行なうので、1 個から自由に開発できて、かつ設計終了からチップ納入までの TAT(ターンアラウンド·タイム) がかからない。また、内部回路の消去/再書き込みが可能であり、しかも基盤実装後の再書き込み (インシステム·プログラミング/インサーキット·リコンフィギュレーション) が可能であるので、この機能を用いればハードウェアに手を加えずに回路を変更できるため、多品種少量生産やバージョン·アップに柔軟に対応できるという利点がある。

このように大きな特徴を持つ FPGA をノードプロセッサとして相互結合網を構成した場合、その特徴がどのような点に活かされるのかを従来の相互結合網と対比して以下に詳しく述べる。

3.2 ネットワーク構成の柔軟性

従来の LSI を使用した場合、そのアーキテクチャ構造に合わせた設計がシステム各所に必要となるため、システム全体の設計の自由度が束縛される可能性がある。

例えば現在の DR-net はノードプロセッサとして Transputer を使用しているが、リンクのためのデバイス上のピン位置が固定されているため、通信は東西南北の 4 方向のみでしか行えない (図 3.1参照)。

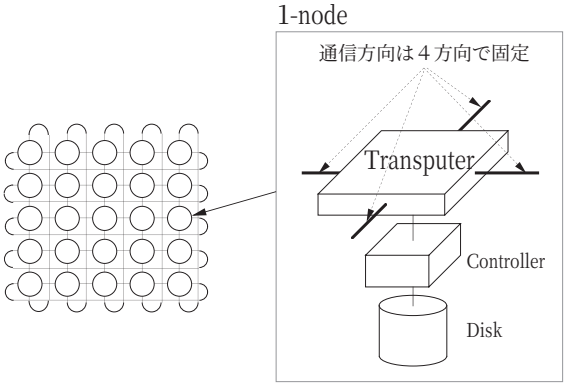


図 3.1: DR-net 実験システムの構成

ノードプロセッサのもつリンク本数が自由に定義できるようになればネットワークの拡張は容易に行うことができるわけであるが、この面に FPGA 特有の「回路構成の柔軟性」を活かすことができる。FPGA は内部の回路構成や I/O ピン配置を自由に設計できるためシステムの設計における柔軟性は非常に高く、ネットワークの拡張も容易に実現することができる (図 3.2参照)。

[11] によれば、DR-net の通信機能のみを FPGA に搭載した場合、5 方向通信では 14800 ゲート、8 方向では 28500 ゲート必要であり、DR-net のノード機能すべてを FPGA で 1 チップ化した場合は 5 方向通信では 4 万数千ゲート、8 方向では 6 万ゲート程度あればよい。FPGA の使用可能ゲート数は年々確実に増加しており、'96 中に使用可能ゲート数が 125000 ゲートの FPGA が市販された。このゲート数は今述べた必要ゲート数を軽く許容しており、今後もゲート数は増え続けるとすれば多次元のハイパーキューブネットワークにも柔軟に対応することが期待される。

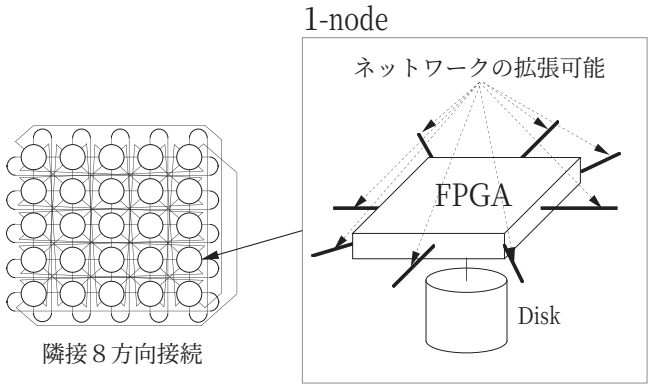


図 3.2: FPGA の柔軟性利用によるネットワークの拡張

3.3 書き換えによる耐故障性

3.3.1 代替リンクの利用によるリンク故障対策

第 3 章でも述べたように、リンク故障処理の手法としては動的、静的どちらのネットワークにおいても故障リンクを迂回しながら目的ノードを目指すというルーティング手法が良く用いられている。この手法の場合、経路選択により迂回したノードやリンクに負荷がかかり、処理上のオーバヘッドが増大してしまうという大きな問題がある。

リンク故障対策として概念的に最も良いのは、リンク故障検出後の耐故障処理によりネットワークの状態を物理的に故障前の状態に戻してしまうことであると思われるが、FPGA のもつ「回路の書き換え可能」という性質を利用することによりこれは実現可能となる。具体的には FPGA に回路データをコンフィギュレーションするためのコンフィギュレーションラインを代替パスとして使用するという方法が考えられる。リンク故障時にはこのラインを代替パスとして使用するようにシステム各ノードの回路を書き換える。ノード間の物理的な接続は失われないため通信性能の低下はないのが特徴である。

しかし後の章でも説明するように、コンフィギュレーションライン自体が故障してしまった場合、次回復旧できるリンクが制限されてしまうという問題がある。

この問題の対策としては各々のノードの外部に ROM を設けダウンロードまでの処理内容を書き込んでおくことにより、書き換え時にシステムを止めることなく、各ノードがローカルに書き換えを行なえるようにするという方法が考えられる。

これらの手法の詳細については第 4 章で述べる。

3.3.2 ルーティング/二重化との違い

ルーティングの場合、ソースからデスティネーションまでは故障リンクを回避して経路選択を行なわなければならない。経路選択するにはそれ専用のアルゴリズムを通す必要があり、しかもそれを故障リンクを回避する度に行なうため故障後のシステムの速度はかなり落ちる。また回避先のノードでの負荷の増大による処理速度の低下や、回避先のリンクでのデータの競合による待ち時間の増加といった事を考慮すると、処理上のオーバヘッドはかなり大きくなるものと思われる。この傾向はリンクが多重故障した場合に更に強くなる。それに比べて FPGA による復旧法では、故障検出後に故障リンクから代替パスへ接続を切り換えるためにリンクの接続情報を書き換えるだけでよい。時間を要するのは書き換えのための再ダウンロードにかかる時間だけで、それ以降は代替パスにより通常と変わらない速度で通信を再開することができる。

次に二重化との違いであるが、リンクを二重化することによる利点はどのリンクで故障が起きても故障リンクから代替リンクへとスイッチを切り換えることによって瞬時に故障復旧可能なことである。また、通常のリンク全ての代替リンクが存在するので、通常のリンク全てが故障してもシステムは動作可能となる。その反面リンクコストは通常リンクの 2 倍に増大する。それに対して本研究で提案する方法では、もともとはネットワークを構成する全ての FPGA にデータをコンフィギュレートするためのコンフィギュレーションラインを代替パスとして用いることができる。これによってノード間を結ぶ代替リンク本数は二重化に比べて半分に抑える事ができるが、救えるリンク故障の数は少なくなる。この件に関しては信頼性についての考察も含め、第 7 章で言及する。

第 4 章

FPGA の書き換えによるリンク故障復旧戦略

FPGA は、ユーザが自分の手元で回路をプログラミングできる LSI デバイスである。特に回路構造を SRAM によって保持するタイプの製品は、「データの書き換え」により何度でも回路をプログラミングすることができる。この「データの書き換え」により、チップ上に自由に回路が構成できることは従来のハードウェアには見られなかった革新的な概念であるといえる。

この章では、FPGA のこの大きな特徴を活かしたリンク故障復旧戦略について詳しく述べることにする。

4.1 前提とする構成

4.1.1 対象とするシステムモデル

設計回路を何度でもダウンロードできるという FPGA 特有の性質を活かしたリンク故障復旧策について詳しく述べる前に、FPGA の開発工程を簡単に説明しておく。図 4.1に、回路図の設計から FPGA へダウンロードするまでの一連の工程を示す。

設計入力 デバイス内部に実現する回路を論理式や回路図、ハードウェア記述言語などを用いて入力する。

ファンクション・シミュレーション ファンクション・シミュレーションでは回路の機能の検証を行なう。ここでのシミュレーションは回路のロジック的な機能を確認するためのものであり、タイミング・シミュレーションは別に行なう。

論理分割 ここで、設計者が入力した回路は、デバイス内部のマクロセルの組み合わせで実現できるように分割される。

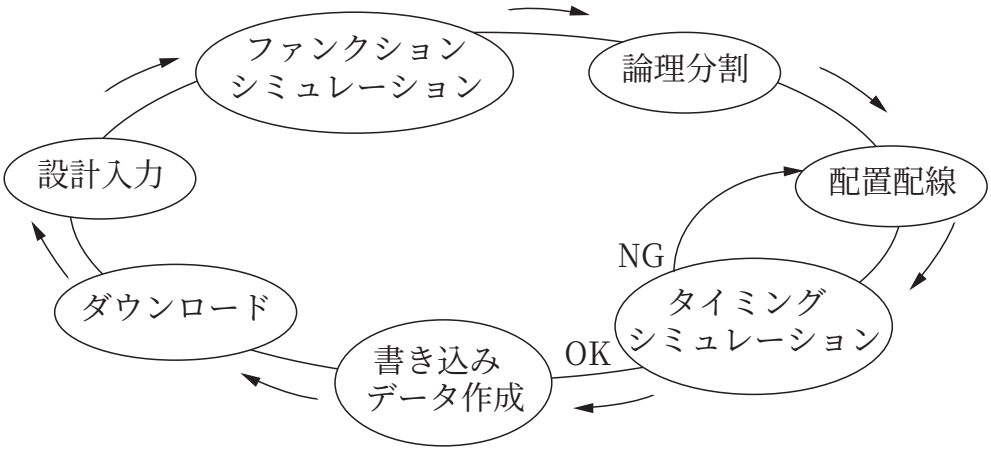


図 4.1: FPGA の開発工程

配置・配線 個々のマクロセルを実際のデバイス上に配置し、その間の配線を行なう。ここでいかに効率良く配線を行なうかによって、できあがる回路の性能が決まる。

タイミング・シミュレーション 配置・配線が終った段階でタイミング・シミュレーションを行なう。配置・配線がどのように行なわれたかによって配線の遅延は大幅に変わってくる。その遅延が回路動作に支障をきたさないことの確認を行なう。ここで問題が見つかった場合は配置・配線をもう一度やりなおすか、手作業で問題の箇所の配線を修正する必要がある。

書き込みデータの作成 ここまでの段階でデバイス上に実現すべき回路はすでに完了している。その回路データをもとに書き込みデータを作成する。

ダウンロード 作成した書き込みデータを専用の書き込み器を用いて FPGA にダウンロードする。

論理設計から FPGA への回路データのダウンロードまでの一連の操作は、パーソナルコンピュータにインストールした CAD ツールを用いて行う。論理回路はパソコンから回路図エディタを用いて入力し、専用 CAD を用いて FPGA への書き込みデータに変換した後、書き込み器経由で FPGA へダウンロードする (図 4.2参照)。

本研究では FPGA をノードプロセッサとして使用し、トーラス結合のネットワークを構成する。その際複数の FPGA に回路データをコンフィギュレーションするために、1 本の配線でこれらをディジチェインする。従って通信リンク以外に全てのノードをつなぐパスができることになる。このラインはコンフィギュレーション後、書き込んだ回路が使用可能な I/O となる。FPGA は回路の再コンフィギュレーションが可能であるので、リンク故障検出時にはシステム各ノードの接続データを書き換えることによってこのラインを代替パスとして用いることができる。

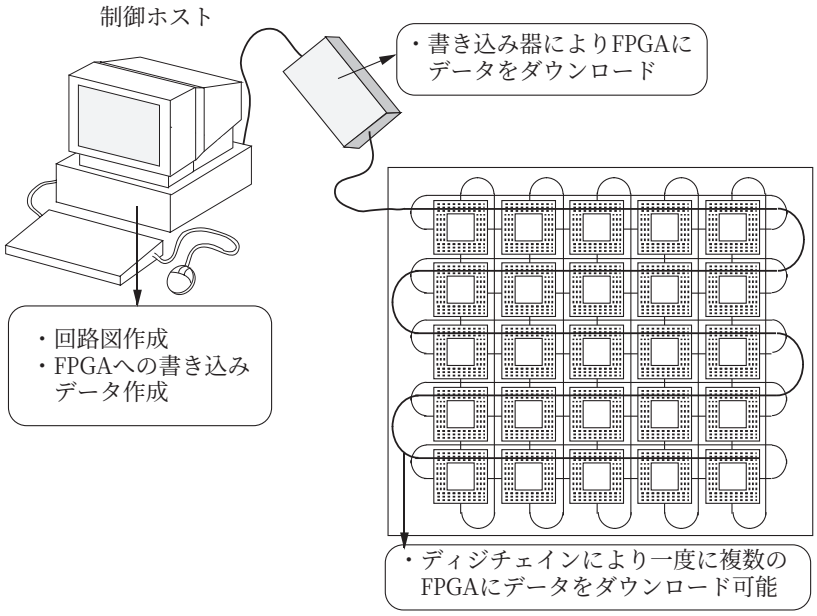


図 4.2: 対象モデルの構成

4.1.2 対象とするリンク故障モデル

ソフトウェアは物理的な特性がなく、このため物理的な故障原因がないのに比べ、ハードウェアには物理的な故障原因が存在する。これはリンクにおいても例外ではなく、様々な故障原因が存在する。次にいくつか故障モデルを挙げる。

物理的な故障 (physical fault) 衝撃による断線、電圧限界によるショート、コネクタの接触不良など

縮退故障 (stuck-at fault) 物理的な故障によって論理ゲートのいくつかのラインが永久に 0 か 1 に固定されてしまう

ビザンチン故障 (Byzantine fault) ラインにノイズが載った場合などに受信器は正しくない (嘘の) 情報を受け取るが、その情報を正しいものと解釈してそのまま処理してしまう

間欠故障 (intermittent fault) システムの稼働サイクルと比較して非常に短い時間だけ存在する一時故障 (temporary fault) の中でも再現性のあるもの

本研究室で提案された Transputer による DR-net 実験システムにおいて一番多いリンク故障は、「コネクタの接触不良」による場合が多い。また、FPGA を用いて実験システムを構成した場合は、ラインの密度が非常に高くなるため、ラインのショートが起こりやすい。

本研究ではモデルの簡単化のため、リンク故障のモデルは「リンクの断線」、あるいは「ピンの接続不良」のような縮退故障を仮定する。

4.2 リンク故障の検出方法

ここでは、データをダウンロードする以前にリンク故障が存在する場合と、システムが動作中にリンクが故障する場合の 2 つの場合におけるリンク故障検出方法を説明する。

4.2.1 イニシャルな故障の検出

コンフィギュレーションラインに故障のない場合

ここでは回路データを FPGA にダウンロードする際、通常リンクに故障がある場合どのようにして検出するのかを説明する。その時コンフィギュレーションラインに故障はないものとし、従ってデータのダウンロードは正常に行なわれるものとする。

まずリンク故障の検出方法であるが、FPGA にダウンロードする回路機能の中にリンク故障検出用の回路を盛り込んでおく。図 4.3 に故障検出機能の一例を示し、以下その手順を順番に説明する。

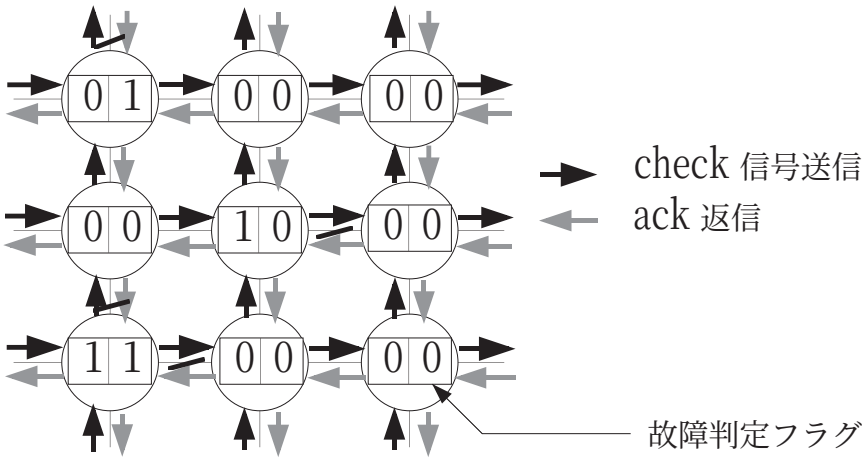


図 4.3: イニシャルな故障の検出

1. まずダウンロードが完了した時点で各ノードは自分の北と東に隣接するノードに 1bit の check 信号を送信する。

2. check 信号を受け取ったノードはすぐに送信側のノードに ACKnowledge(以下 ack と記述) 信号を返信する。

3. 各ノードは故障判定フラグを持ち、リンクの故障によって ack が送信側に返信されない場合は、タイムアウトにより送信側がそのリンクを「故障」と判断し、故障検出フラグを立てる。

対象とするネットワーク構成はトーラス結合のため、check 信号を送信する方向は北と東 (南と西でもよい) の 2 方向とするだけで全てのリンクをチェックすることができる。

各ノードにおいて考えられるリンクの故障状態は

1. リンク故障なし
2. 北側のリンクのみ故障
3. 東側のリンクのみ故障
4. 北側、東側両方のリンクが故障

の 4 状態である。よって故障検出フラグは 2bit 用意し、2 の場合は 01 を、3 の場合は 10 を、4 の場合は 11 を、という具合にフラグを立てれば、どのリンクで故障が発生したのかを判定することができる。

FPGA 内部で検出したこのフラグ情報はダウンロード後フリーとなったコンフィギュレーションラインを利用して制御ホストに出力し、それによって制御ホストは故障の有無を認識することができる。

コンフィギュレーションラインに故障がある場合

コンフィギュレーションラインも通常リンクと同じように故障すると仮定した場合、故障したライン以降の FPGA には回路データをダウンロードできない。従ってそれらの FPGA では書き換えは行なえず、またそれらの FPGA に接続されるリンクにおいて故障が発生しても代替パスによる復旧は不可能となる。

データの書き込み器の直後でラインが故障した場合は、データをダウンロードすることすらできない。この対策として分散故障復旧方式を考える。この詳細については 4.3.2で説明する。

4.2.2 動作中に発生した故障の検出

次に、FPGA への回路データのダウンロード後、システムの動作中にリンク故障が発生した場合にどのように対応するかについて説明する。ここでデータの転送方式は Transputer と同じシリアル転送を仮定する。シリアルデータの 1 ブロックはスタートビット、データビット、ストップビットから構成されるものとする。どのタイミングで故障検出を行なうかについてその手順を以下に示す。

1. 送信側のノードが受信側のノードに check 信号を送信する。
2. 受信側が正常に check 信号を受け取った場合、すぐに送信側にその旨を通知し、それを受けて送信側はデータを送出する。

3. リンク故障により ack 信号が正常に送信側へ返信されない場合、送信側はタイムアウトによりその旨を感知し、データの送出を中止する。

送信側が ack 信号を送出する以前に既にリンクが故障している場合はこの方式により故障検出でき、送出予定のデータが失われることはない。しかし、データの転送中にリンクが故障した場合、故障リンクは検出できるが転送中だったデータは失われてしまう。以下この問題の対策を述べる。

4.2.3 データ損失回避

データ転送中にリンク故障が発生した場合、転送中だったデータは失われてしまう。よってシステムにはデータの損失を避けるための機構が必要となる。

リンク故障時において転送中のデータを保護するために、次のようなプロトコルを考えた。

1. Sender が Receiver からの「返信 ack」を受け取った後データを送出すると同時に、必ずメモリにもデータをロードする。

2. リンク故障により転送中のデータが失われた場合、Receiver はタイムアウトによりデータが損失したことを感知し、故障したリンクの両サイドのノードの接続情報を代替パス側へ書き換える。

3. Receiver は Sender に「データ損失信号」を代替パス経由で返し、Sender はそれを受けてメモリから予備として格納したデータを読み込んで再び送出する。

1 で格納するメモリは外付けメモリを仮定する。FPGA 内部にメモリを設けることは可能であるが、書き換えを行なう際に一旦内部回路がクリアされ、その時一緒に内部メモリの保存内容もクリアされてしまう。また回路規模も大きくなってしまうためメモリは外付けにする必要がある。

リンク故障時、送信側は故障の発生をタイムアウトにより知ることができるが、データが失われたかどうかは分からないため、受信側が送信側にそのことを知らせる必要がある。知らせる手段としては「データ損失信号」を送ればよいのだが、肝心のリンクが故障しているため当然そのままでは送ることができない。よって受信側がデータの損失を感知した時点で故障リンクの両側のノードの接続を代替パスへ切り換えてしまえばよい。そうすることで代替パス経由でデータ損失信号が送信側へ送られ、送信側はメモリに格納しておいた予備データを代替パス経由で再送出することができる。

4.3 リンク故障復旧戦略

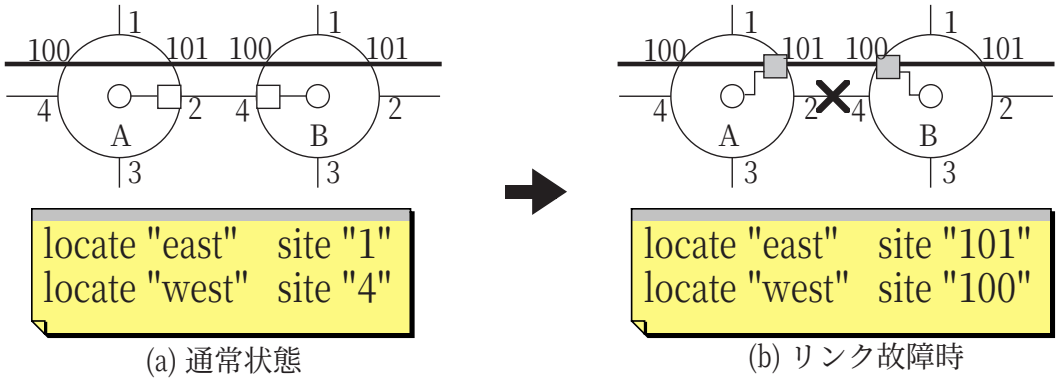
リンク故障の復旧戦略は、制御ホストが集中して復旧処理を行なう「集中方式」と各ノードがローカルに復旧処理を行なう「分散方式」の 2 つに大別することができる。

4.3.1 集中故障復旧方法

この方式ではリンク故障検出後の復旧処理は制御ホストが一括集中して行なう。

図 4.1に示したFPGA の開発工程において、リンク故障時に必要となるのは配置・配線以降の処理である。

具体的に説明すると、配置・配線を行なう際には設計した回路の入出力線を FPGA のどのピンへ割り当てるかを記述した「プリファレンスファイル」が必要となる。各ノードを結ぶリンクのピンの位置はハードウェア設計時に固定されているため変更できないため、通信を行なうための送受信モジュールの入出力線をそのピンに割り当てるようにこのファイルに記述する (図 4.4(a))。



(a) 通常状態　　(b) リンク故障時

図 4.4: Preference File の書き換え

あるリンクに故障が起きた場合は、プリファレンスファイルにおける送受信モジュールの入出力線の接続情報を故障リンクから代替パス側へ書き換え、再び配置・配線からやり直す。FPGA へのダウンロードが完了した時点で、1 本のリンク故障が復旧できたことになる (図 4.4(b))。

制御ホストは故障の判定フラグを受信してから再ダウンロードするまでの一連の復旧処理を全て自動的に行なう。

この方策の利点は、故障の検出から復旧まで制御ホストが一括して処理を行なうため各ノードの構成を単純化できるということである。また注意すべき点は、ダウンロードをやり直す際に各ノードの内部情報が一旦消えてしまうため、復旧処理はデータの送受信を行なっていないタイミングで行なう必要があるということである。

4.3.2 分散故障復旧方法

上で述べた方法における一連のリンクの故障復旧操作は制御ホストが一括集中で行なっていたが、この方式では各ノードがローカルに故障復旧操作を行なう。つまり集中制御していた処理を各ノードに分散したわけである。

では具体的にどのようにするかというと、各 FPGA の外部に ROM を設け、これにリンク故障復旧のための一連の処理内容を書き込んでおく。

図 4.5にこのシステムの概略図を示す。図中の黒い長方形で示してあるのがは ROM であり、各 FPGA にローカルに接続される。

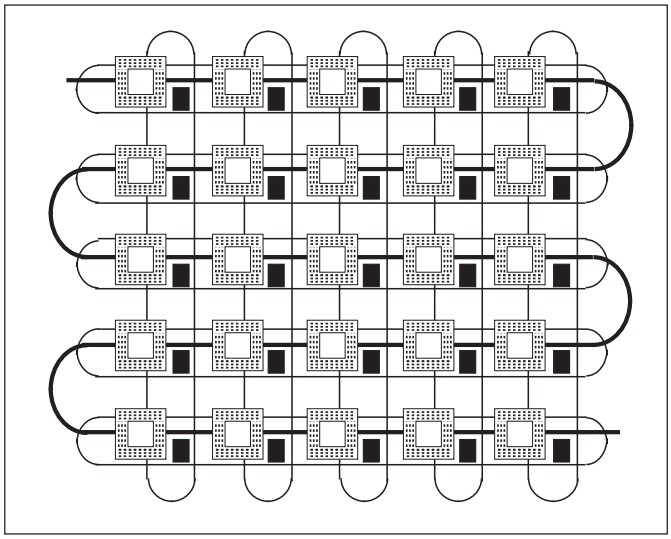


図 4.5: 分散故障復旧システムの概略

リンクに故障が発生した時に接続情報が書き換えられるのはそのリンクの両サイドのノードの Preference File のみである。従って 1 本のリンク故障で 2 個のノードの Preference File が書き換えられる。よってピンの接続を代替パスへと書き換えた Preference File を書き換えパターン分用意し、これらも ROM に格納しておく。

リンク故障検出時には ROM からそのリンクの両サイドのノードの Preference File を読み込む。

ここで、ROM からノードの書き換えに必要な処理を読み込むための回路は FPGA の外からロードしてやる必要がある。なぜならノードを書き換える際に FPGA の内部回路は一旦消去されてしまうからである。具体的には ROM とは別に書き込み専用のデバイスを各々の FPGA に設け、内部回路が消去された時点でそのデバイスが各 FPGA に必要回路をロードする。

その後ダウンロードまでの処理工程を ROM から読み込み、実行することにより FPGA への書き込みデータを作成する。再ダウンロードによりリンク故障の復旧が完了する。

この方式による利点は、各ノードがローカルに書き換えを行なうためいちいち全ノードを書き換える必要がないことである。また集中方式では、コンフィギュレーションラインが故障した場合は故障箇所以降のノードを書き換えることができなくなってしまうが、分散方式では ROM から復旧処理に必要な情報を読み込むことができるため、コンフィギュレーションラインを使用する必要はなく、ラインは故障復旧のみに

使用できることである。よってこの場合ラインで故障が発生しても、故障部分以外のリンクでは故障復旧できることになる。

4.4 ノードの書き換え

4.4.1 各ノードの書き換えを一回に限った場合

ある一本のリンク故障に対して、その両サイドのノードで一回だけ書き換えを行う手法についてその手順を示す。

図 4.6にこの手法の概念図を示す。□で示したのは送受信モジュールであり、各リンクに接続されている。

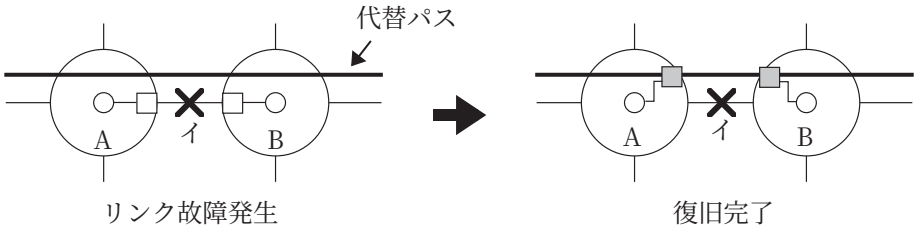


図 4.6: ノードの書き換え 1 回

例えばイのリンクに故障が発生した場合、ノード A、ノード B における送受信モジュールの故障リンクへの接続情報を代替パスへと書き換えることにより復旧が完了する。

この手法は集中、分散どちらの故障復旧法にも適用できる。しかし集中復旧法にはコンフィギュレーションを代替パスとして用いることによる弱点が存在する。

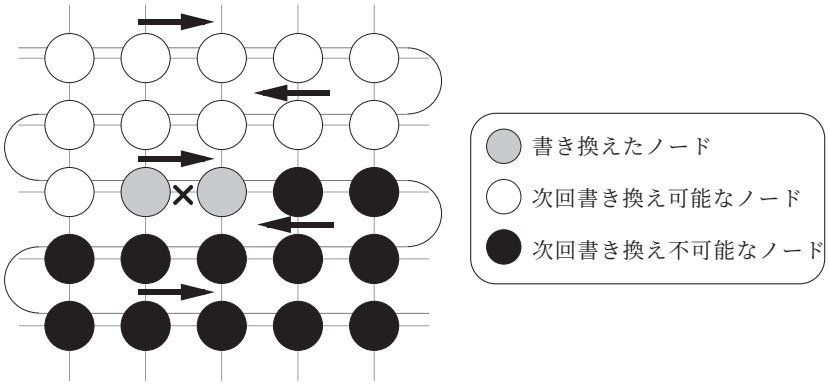


図 4.7: ダウンロード方向の制約による影響

図 4.7においてダウンロードは一番左上のノードから矢印の方向に順番に行なわれるものとする。最初のリンク故障により網掛けしたノードで一回書き換えを行ない故障を復旧した場合、書き換えたノードと、そのノードより後にダウンロードされるノード (黒塗りノード) にはダウンロード用の回路データが届かないため、次回黒塗りノードに接続される 4 本のリンクで故障が発生しても復旧させることができなくなってしまう。

この解決策としては、2 本目以降のリンク故障が黒塗りノードエリアで発生した時は、ダウンロードを後ろ (この場合一番右下のノード) からやり直すようにする。制御ホストは故障がどこにあるのか認識できるので、書き込み器から遠い順に故障リンクを復旧していく。こうすることで複数の故障が発生した場合でも復旧することができるようになる。

4.4.2 各ノードの書き換えを複数回許した場合

4.4.1でも述べたように、集中故障復旧方式においては故障検出後、再ダウンロードの順番を考慮することによりノードは何度でも書き換えることが可能となる。また分散故障復旧方式においては、各々のノードが各自保有する ROM からローカルにダウンロード用の回路データを読み込むことができるのでもともとノードの書き換えは何度でも行なうことができる。

ここでノードの書き換えを複数回許した場合のリンク故障復旧戦略についてその手順を述べる。図 4.8にこの手法の概念図を示す。

図 4.8において、イのリンク故障復旧後、ロ、又はハのリンクで故障が発生した場合、ロに関してはノード A が、そしてハに関してはノード B が 2 度目の書換えを行なっている。

この方式の利点は、複数回の書き換えによりこれまで復旧できなかったリンクも復旧できるため、単純に復旧可能なリンク数が増えることである。5 × 5 の構成においては 1 回しか書き換えられない場合は最大で 12 本復旧できるが、複数回 (今の場合は 2 回だけ) 書き換えれる場合は倍の 24 本の故障を普及することができる。

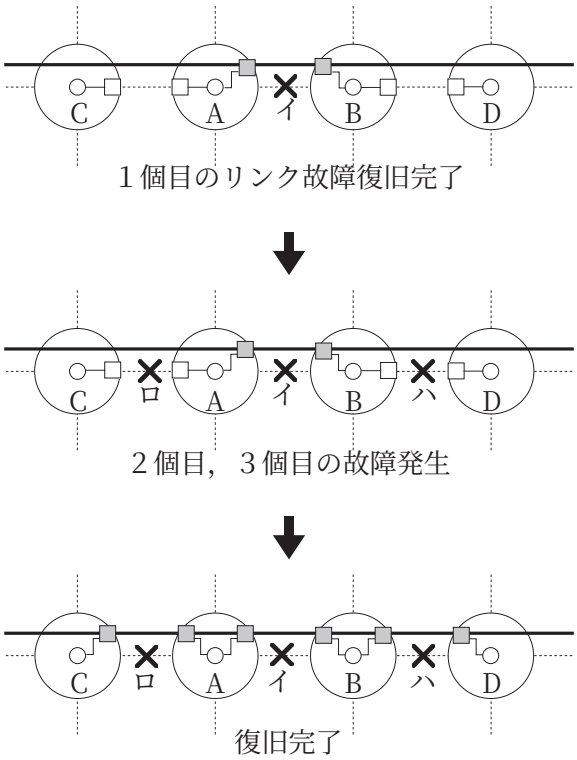


図 4.8: ノードの書き換え複数回

第 5 章

代替パスのパターンとその評価

4 章で説明したように、リンク故障の復旧には複数の FPGA に回路データをコンフィギュレーションするためのコンフィギュレーションラインや、これとは別に新たに設けたラインを代替パスとして利用する。しかしこの代替パスの張り方次第で復旧可能なリンク数が変わってくる。よってどのパスパターンが最適となるのか明らかにしなければならない。

この章ではまず実際に代替パスを張る上での制約について説明する。そしてそれを踏まえてパスを張った際に、数多く存在するパスパターンからどのようにして最適なパスパターンへと絞り込んでいくのか、その基準について説明する。

5.1 代替パスパターンの生成

5.1.1 ネットワークサイズに対する代替リンク数の制約

4.1.1でも述べたように、FPGA をノードプロセッサとしてネットワークを構成する時、複数の FPGA に回路データをコンフィギュレーションするために用いるコンフィギュレーションラインを代替パスとして利用することができる。

すべての FPGA に回路データをコンフィギュレートするためにコンフィギュレーションラインは各ノードを必ず一回通過することになるので、リンク故障の復旧にこのラインを代替パスとして考えた場合、2 つのノードを結ぶ代替リンクはコンフィギュレートしたノードと同じ数だけ存在することになる。また、このラインによって全てのノードが接続されるので、あるノードからその他すべてのノードへのパスが必ず 1 本存在することになるので、最初の故障がどのリンクに発生してもそれを復旧させることができる。

図 5.1に 5 × 5 のトーラス結合ネットワークに対する代替パスパターンの一例を示す。この場合ノード数は 25 個であり、代替リンク本数は 25 本である。

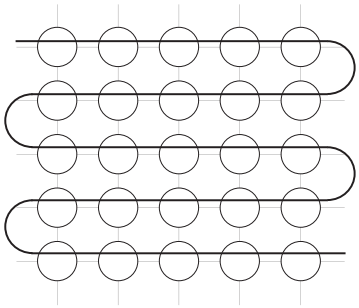


図 5.1: 代替パスパターンの一例

5.1.2 パターンの絞り込み

普通、コンフィギュレーションラインの張り方は各ノードを 1 度だけ通過するという条件さえ満たしていれば任意でかまわない。しかしリンク故障時の代替パスとして利用することを前提とした場合、任意に張るというわけにはいかなくなる。なぜなら、ラインの張り方次第で復旧できるリンク数が変化するなど、故障復旧に関して特性が変わるからである (この話については後の章で詳しく述べる)。

よってどのようなパターンで、あるいは法則でラインを張れば最適パターンと言えるのかを明らかにする必要がある。とはいってもパスパターンは単純に考えてもノード数 n の相互結合網において n!通りも存在するためパターンの絞り込みを行なう必要がある。以下その絞り込み方法を説明する。

トポロジカルに同一なパターンの除去

トーラス結合のネットワークの場合、ずらしたり反転したりすると同一となるパターンが多数存在する。図 5.2に、トポロジカルに同一な 4 つのパターンの例を示す。

pattern A を基準に考えると、pattern B は A を左右、又は上下に反転したもの、pattern C は A を横方向にずらしたもの、そして pattern D は A を回転したものであり、どのパターンも同一であるとみなすことができる。

よってこういった同一パターンを除去してパターンを絞り込んでいく。

評価による絞り込み

トポロジカルな絞り込みで得られた代替パスパターンについて、システムの信頼性の面での優劣を判断するために何らかの評価指標が必要となってくる。よってある評価指標を設け、その指標に基づいて更にパターンを絞り込んでいくことにする。次節にその詳細を述べる。

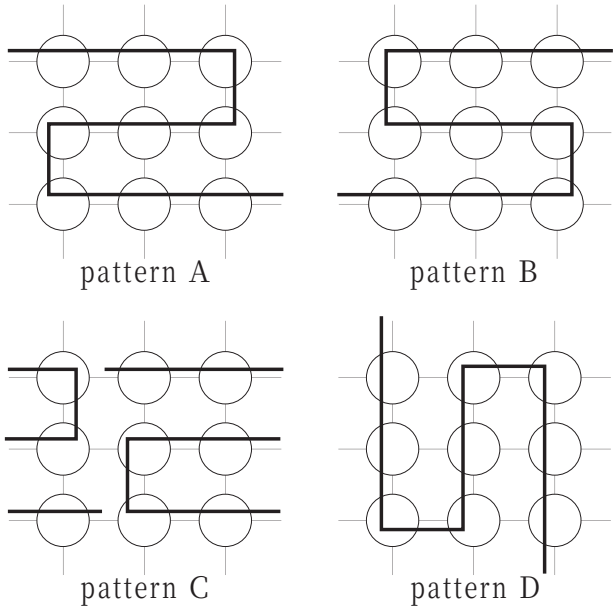


図 5.2: 同一パスパターンの除去

5.2 評価の指標

5.2.1 評価の方針

5.1.2で述べたように、n!通り存在する代替パスパターンの中から最適なパスパターンを絞り込んでいく上で、トポロジカルな絞り込みには限界がある。また故障復旧率の面では、パターンが変わることによって復旧できる故障数は変わってくる。よってどのパターンが最適であるかを判断するためには何らかの評価指標が必要となってくる。

本研究では以下に述べる 2 つの評価指標を中心に、様々な代替パスパターンを検討、評価していく。

5.2.2 故障に対する影響リンク数

あるリンクで故障が発生し、ノードの書き換えによりそれを復旧した場合、代替リンクのパターンによっては次回故障が発生した場合に復旧できないリンクが存在する。そういったリンクの数を「影響リンク数」という名の一つの評価指標とする。

「影響リンク数」を定義する上で、図5.3に示す5 × 5 のトーラス結合ネットワークを例として、その概念を示す。

ノード 0-5 間のリンク故障に対し、その故障リンクに接続されるノード 0 とノード 5 で代替パスを利用するように 1 回書き換えを行なう。それによってノード 0-5 間のパスは確保され、データの送受信が可能と

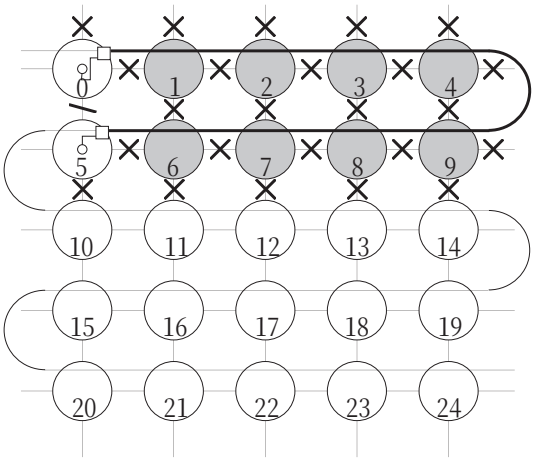


図 5.3: 影響リンク

なる。しかし代替パスの太線部分はノード 0-5 間のデータ送受信によって占有されており、次回のリンク故障に対して網掛けしたノードで書き換えを行なった場合、この占有が崩れ、ノード 0-5 間で通信ができなくなってしまう。従って網掛けのノードで書き換えは行なえないので、最初の書き換えを行なったノード 0、5 に接続されるリンクだけでなく、網掛けしたノードに接続される全てのリンク、即ち×印を付けたリンクで次回リンク故障が発生した場合、復旧できない。この×印の付いたリンクを「影響リンク」と呼び、ある 1 本のリンク故障に対する影響リンクの総数を「影響リンク数」と呼ぶことにし、これを様々な代替パスパターンを評価する際の一つの大きな指標とする。

ここでは最大影響リンク数、最小影響リンク数、平均影響リンク数を測定した。例えば 5 × 5 のトーラスの場合、総リンク数 50 本に対して最大影響リンク数は 49 となるが、これはリンク故障を一本しか復旧できないことを意味する。また、システムの信頼性を考えた場合、平均影響リンク数が少ない代替パスパターン程良いパターンであると言える。なぜなら、一個目のリンク故障が 50 本のリンクそれぞれに等しい確率で発生するとすると、平均影響リンク数が多い程、1 本のリンク故障に対してその他のリンクが影響を受ける確率が高くなるからである。言い替えれば故障しても復旧できないリンクが増加するということであり、従ってシステムの信頼性も低くなるものと推測される。信頼性については 7 章で詳しく分析する。

5.2.3 最大復旧可能リンク数

ある 1 本のリンク故障に対し、ノードの書き換えによりこれを復旧した際に次回故障が起きた時に復旧できないリンク、つまり影響リンクが存在することは 5.2.2で述べた。ここでは逆に、ノードの書き換えによって復旧できるリンクは何本存在するのかということを考える。

ネットワークにおいて幾つのリンク故障を復旧できるかという問題は重要な評価指標となりうる。よっ

て、ある代替リンクパターンにおいて最大で何個リンク故障を復旧できるか、その個数を「最大復旧可能リンク数」とし、これも一つの評価指標とする。

5 × 5 の相互結合網を例に、最大復旧可能なリンク故障のパターンを図 5.4に示す。

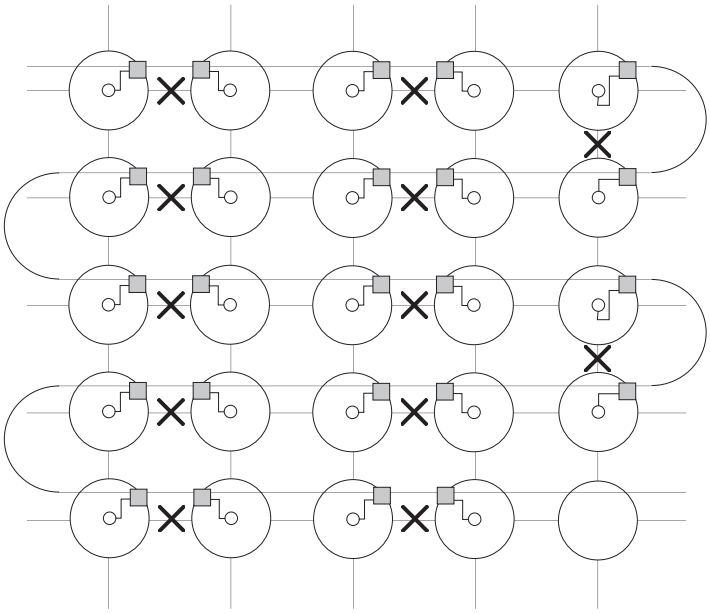


図 5.4: 最大復旧可能リンク数

この場合、最大で 12 個のリンク故障を復旧できる。

第 6 章

具体的な相互結合網における代替パスパターンの評価

この章では、ネットワーク構成を 5 × 5 の二次元 4 隣接トーラスに限定し、実際に様々なパターンの代替パスを張りめぐらしてそのパターンの妥当性を評価する。

6.1 二次元 4 隣接トーラス (書き換え一回)

6.1.1 乱数によるパターン生成

数多く存在する代替パスパターンにおいてどんなパターンが良いのかを探るために、0 から 24 までの数字をランダムに並べ替え、平均影響リンク数、及び最大復旧可能リンク数を測定した。5.1.2で述べたように、ノードの「ずらし」によって同一となるパターンは多数存在する。よってそのようなパターンを除外するために開始ノードを固定した。

今回ノードの番号付けは図6.1のように行ない、開始ノードは 0 に固定してランダムに数字列を出力した。

出力した結果を解析するにあたり、番号順に全てのノードを線で結びパターンを形成した。しかし出来上がったパターンは非常に見にくく、規則性があるのかないのか判断が難しいため、表現法を図 6.2のように簡略化した。この場合、乱数によってランダムに生成された数字列は

```
0,9,19,6,2,3,7,17,14,15,10,22,8,12,5,18,24,11,4,20,1,23,13,16
```

である。次に各ノードを何番めに通過したのか示すために、各ノード番号の下に通過順番を示す番号を割り振っていく。

```
0, 9,19, 6, 2, 3, 7,17,14,15,10,22, 8,12, 5,18,24,11, 4,20, 1,23,13,16,21
0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24
```

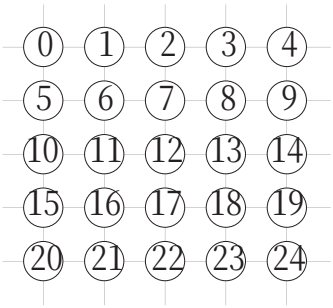
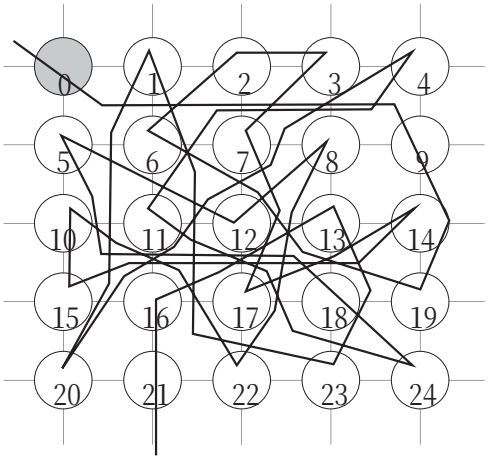


図 6.1: ノードの番号付け

この通過順番号を5 ×5 のボックスの対応するノードの位置に入れていけば図6.2の右の簡略図が得られる。この簡略図を用いてランダムパスパターンの解析を行なう。



0	20	4	5	18
14	3	6	12	1
10	17	13	22	8
9	23	7	15	2
19	24	11	21	16

図 6.2: パターン表現の簡略化

図6.3にランダムパスパターンの一部を示す。平均影響リンク数、最大復旧可能リンク数を二大評価指標とし、評価値の悪い順にパターン A、B、C、D と名前を付けた。

考察

この 4 つの代替パスパターンそれぞれにおいて、平均影響リンク数、最大復旧可能リンク数を基に、パスの張り方に規則性が存在するのか考察してみる。

評価値の悪いパターン A と評価値の良いパターン D を比較して気づくことは、パターン A では、東西南北の 4 方向に隣接するノードを結ぶパスの数がパターン D に比べて極端に少ないということである。各パターン毎に東西南北 4 方向に隣接するパスの数をまとめた (表 6.1)。

0	20	4	5	18
11	3	6	12	1
10	16	13	22	7
9	23	8	15	2
17	24	14	21	19

パターンA

平均影響リンク数　　：34.28 本
最大復旧可能リンク数：5 個

0	7	6	16	19
10	11	15	20	4
23	18	1	17	24
13	14	2	9	21
8	5	3	12	22

パターンB

平均影響リンク数　　：29.08 本
最大復旧可能リンク数：9 個

0	4	1	14	19
13	12	15	16	18
23	17	9	8	24
20	6	2	10	21
7	5	3	11	22

パターンC

平均影響リンク数　　：24.04 本
最大復旧可能リンク数：9 個

0	3	4	6	9
21	16	17	19	22
23	15	18	20	24
13	10	11	12	14
1	2	5	7	8

パターンD

平均影響リンク数　　：19.16 本
最大復旧可能リンク数：12 個

図 6.3: ランダムパターン

表 6.1: パターン毎の隣接パスの数

パターン	隣接パス	合計
A	(4-5)(9-10)(10-11)(23-24)	4 本
B	(1-2)(2-3)(6-7)(10-11)(13-14)(21-22)(23-24)	7 本
C	(2-3)(4-5)(5-6)(8-9)(10-11)(12-13)(15-16) (18-19)(20-21)(21-22)(23-24)	11 本
D	(0-1)(1-2)(2-3)(3-4)(4-5)(6-7)(7-8)(8-9) (10-11)(11-12)(13-14)(15-16)(16-17)(17-18) (19-20)(21-22)(23-24)	17 本

その結果、パターン A、B、C、D と評価値が良くなっていくにつれ、隣接パスの数が増加していくのが分かる。この傾向を逆に言えば、評価値が悪くなるにつれノードを一つ跳ばし、あるいは二つ跳ばしで結合するパターンや、斜め方向に隣接するノードを結合するパターン (以下、跳び結合、斜め結合と呼ぶ) が多く見られる (図 6.4参照)。よって平均影響リンク数をできるだけ少なく、また最大復旧可能リンク数をできるだけ多くなるように代替パスを張るためには「斜め結合」及び「跳び結合」を用いず、東西南北いずれかの方向に隣接するようにノードを結合していけばよいと推測できる。

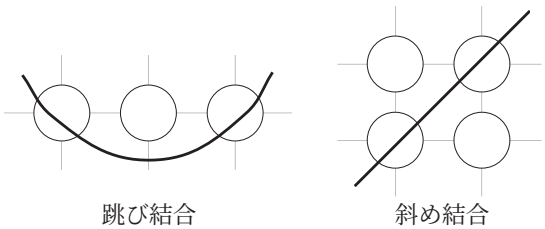


図 6.4: 悪い結合パターン

また、3×3 から 10×10 の構成までのトーラスネットワークにおいてランダムに数字を繋いで平均影響リンク数を求めていき、その中で平均影響リンク数が一番低くなったパターンの値を表 6.2に示した。

ランダムパターン生成において今回採用したアルゴリズムは以下の通りである。

1. ランダム関数により数字列を出力し、そのパターンにおける平均影響リンク数及び最大復旧可能リンク数を測定する。

2. この数字列をランダムに並べ換え、再び評価指標を測定し、前の値よりも良かったらこの数字列を基準値として採用する。

表 6.2: 各ネットワーク規模における平均影響リンク数

$n \times n$ トーラス	平均影響リンク数	最大復旧可能リンク数	理論最大復旧可能リンク数
3×3	11.00	4	4
4×4	15.50	8	8
5×5	19.16	12	12
6×6	22.90	18	18
7×7	26.76	24	24
8×8	30.38	31	32
9×9	35.00	38	40
10×10	39.79	45	50

3. 評価指標の値が一定値に収束するまで 2 の操作を繰り返す。

しかしランダムに数字を並べ換えているため、平均影響リンク数がある一定値 (Local Minimum) から下がっていない可能性がある。また、ネットワーク規模が大きくなるにつれ最小値が一定値に収束しない。よって表 6.2における値を各ネットワーク構成における最小値であるという様には断定はできないが、パターンを絞り込む際の一つの基準とはなり得る。ただし 8×8 以降は最大復旧可能リンク数が理論値と一致していないため、基準として用いることができるのは 7×7 の構成の値までとする。今後 7×7 の構成まででパターンを絞り込む時にはこの値を一つの基準とする。

6.1.2 パターンの分類とその考察

6.1.1でパターンを観察した結果、斜め結合、一つ跳ばし結合を多用している代替リンクパターンでは平均影響リンク数、最大復旧可能リンク数共に悪い傾向にあることが分かった。そこで、斜め結合、一つ跳ばし結合を強調した代替パスパターン、また逆に接続方向を東西南北の 4 方向のみに限定した代替パスパターンにおいて、その傾向の信憑性を確認する。

ここでは図 6.5に示した 13 通りのパターンにおいて評価を行なう。評価指標として、平均影響リンク数、影響リンク数の標準偏差、最大リンク故障復旧数、最小リンク故障復旧数をとり、各パターン毎にデータをまとめた (表 6.3)。

考察

表 6.3を見ると、最大復旧可能リンク数が最大の「12」となるパターンというのは、東西南北の 4 方向に隣接するノードのみを代替パスで結合していったパターンであり、パターン 3 のような斜め結合や、パター

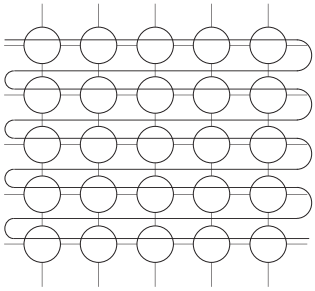
ン 8 のような一つ跳ばし結合を含まないことがわかる。

また、斜め結合や一つ跳ばし結合を多用するパターンでは、平均影響リンク数も大きくなっているのがわかる。4.4.1でも少し述べたように、一本のリンク故障における最小影響リンク数は 6 本であり、これは 2 つの隣接ノードを代替パスが直接結ぶ場合にのみ最小となる。それ以外の結合パターンでは影響リンク数は飛躍的に増加し、パターン 5、6、7 では平均影響リンク数は総リンク数 (この場合 50 本) の 50%以上にもなることがわかる。

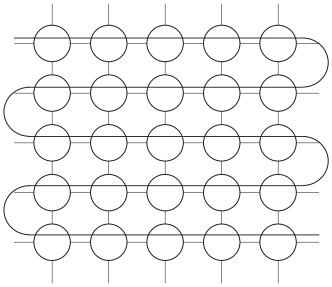
最大復旧可能リンク数が「12」を超えるパターンはここでは存在していないが、その理由はリンク故障の復旧方針に依存する。表 6.3におけるデータはノードの書き換えを一回と仮定して採ったものであり、従ってこの場合書き換えを行なったノードの両サイドのリンクに次回故障が起きても復旧できないことになる。よって、パターン 14 のように結合パターンに斜め結合や一つ跳ばし結合を用いず全てのノードに関して東西南北のいずれかの方向のみに代替パスを張ったパターンでは、リンク一つおきに故障復旧することになる。今対象としているのは 5 × 5 のトーラス結合網であり、その代替パス距離は 24 であるので、復旧可能なリンクの数はその半分の「12」ということになる。

表 6.3: 各パターンにおける評価結果

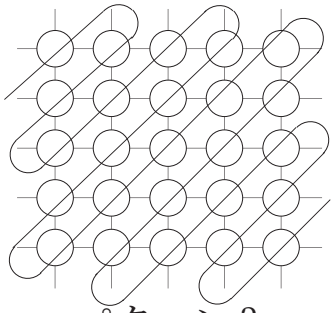
パターン	平均影響リンク数	標準偏差	最大復旧可能リンク数	最小復旧可能リンク数
1	19.80	10.097	10	2
2	19.80	10.471	12	2
3	23.00	9.771	8	3
4	22.04	10.112	12	2
5	30.48	9.830	5	1
6	30.48	9.388	5	1
7	26.24	9.438	5	1
8	21.20	9.656	5	1
9	20.64	10.081	9	1
10	19.80	10.449	12	1
11	21.00	10.816	12	2
12	20.88	10.153	12	2
13	19.16	9.516	12	2



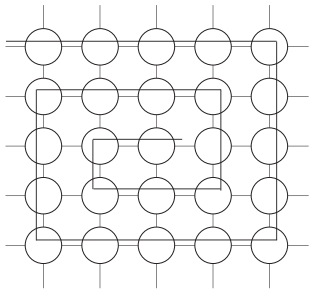
パターン 1



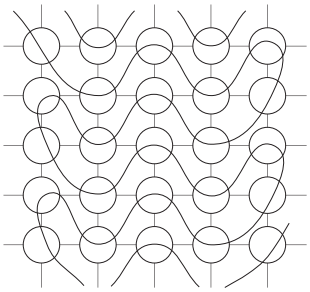
パターン 2



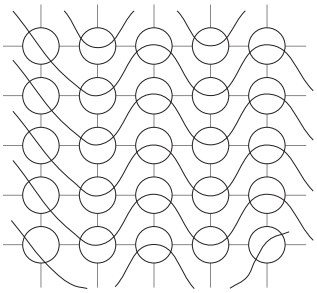
パターン 3



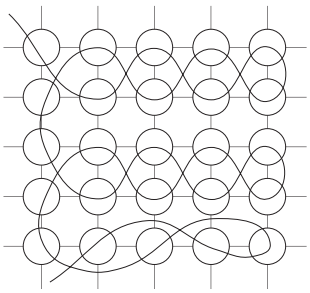
パターン 4



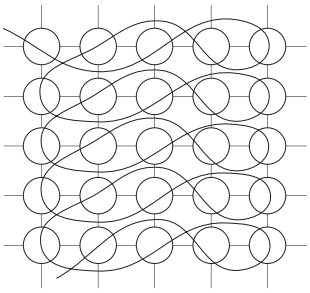
パターン 5



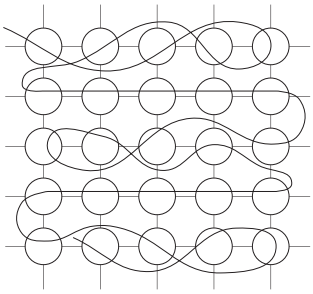
パターン 6



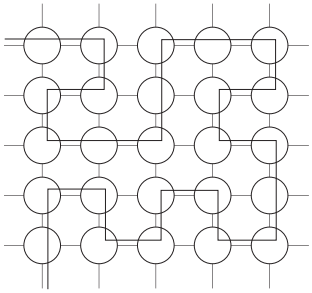
パターン 7



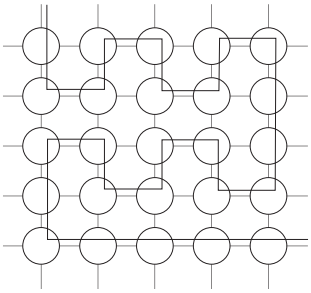
パターン 8



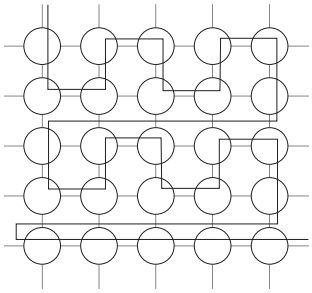
パターン 9



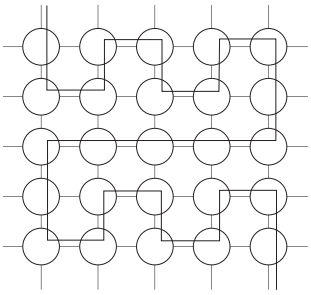
パターン 10



パターン 11



パターン 12



パターン 13

図 6.5: 結合則強調パターン

6.1.3 制限付きパターン生成

6.1.2で述べたように、ノードに代替パスを張る際に斜め結合、一つ跳ばし結合を多用した場合、それらの結合を含まないパスパターンに比べ平均影響リンク数は飛躍的に増大し、復旧可能なリンク数も少なくなることがわかった。

これらのことから予測できるのは、代替パスの張り方としては結合パターンに「斜め結合」、「跳び結合」を用いず、東西南北の 4 方向いずれかに隣接するノード同士を繋ぐようにすれば良いということである。そうすることによって平均影響リンク数を抑えることができ、またネットワークの上限値までリンク故障を復旧することができる。

パターンの絞り込み

よってここでは 5 × 5 のトーラス結合網を対象に、「接続方向を東西南北 4 方向結合に限る」という条件の下、一筆書きで得られる代替パスパターンを全て網羅した。また、5.1.2でも述べたようにパターンの絞り込みを行なうため、ここでも開始ノードは「0」に固定した。

またこの接続形態の場合、最大復旧可能リンク数は必ず上限値の「12 本」となるので、パターンの絞り込みには平均影響リンク数を用いた。具体的には平均影響リンク数が 5 × 5 のネットワークにおける最小値と思われる「19.16」となるものを抜粋した。

その結果、平均影響リンク数「19.16」、最大復旧可能リンク数「12」となる結合形態限定パスパターンは全部で 512 通り得られた。ここで、開始ノード 0 から採り得る方向は 4 方向あるが、対象とするのはトーラス結合のためノード「0」以降で採り得るパスパターンもネットワークの回転、あるいは反転により全て一致する。よって考慮するパターン数は 1 方向だけで良く、従ってパターン数は 1/4 の 128 通りに絞り込むことができる。

この 128 通りのパスパターンを更に絞り込む。得られた 128 通りのパスパターンを分類したところ、図 6.8 のようなツリー状になった。このツリーを見れば分かるように、ノード「0 → 1」の次に採り得るノードパターンは「6 → 5」あるいは「21 → 20」の 2 通りである。この「0 → 1 → 6 → 5」と「0 → 1 → 21 → 20」までのパスパターンは線対象であるので、それ以降のパスパターンも反転すれば一致するものと考えることができる。よって考慮すべきパスパターンはさらに半分になって 64 通りとなる。

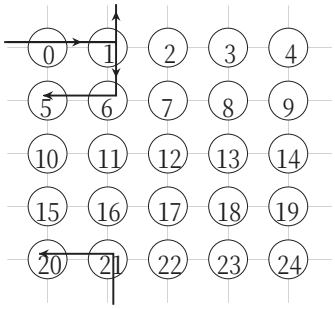


図 6.6: 反転によるパターンの一致

図 6.7において、パターン 1 とパターン 2 は一見違ったパターンに見えるが、パターン 1 を反転してずらすとパターン 2 と同一パターンとなる。このように「反転」、「ずらし」によって一致するパスパターンを除外すると、最終的に考慮するパターンは 36 通りとなった。

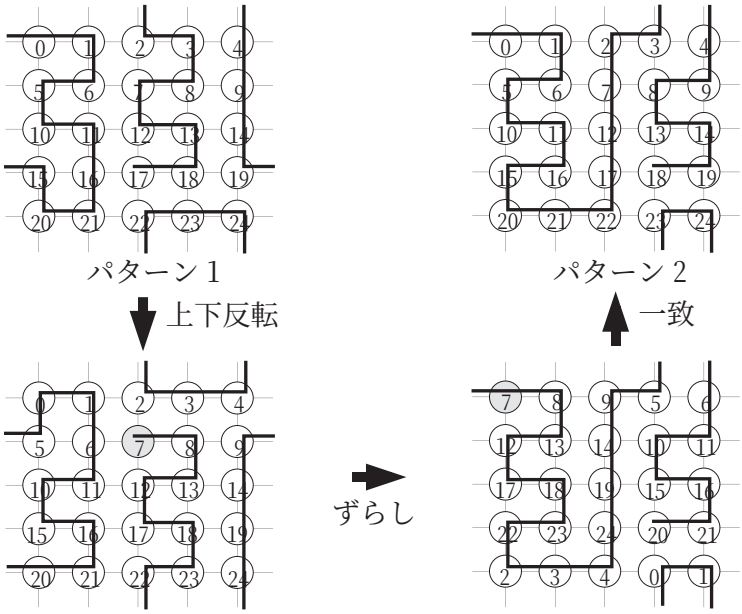


図 6.7: 反転、ずらしによる同一パターンの除去

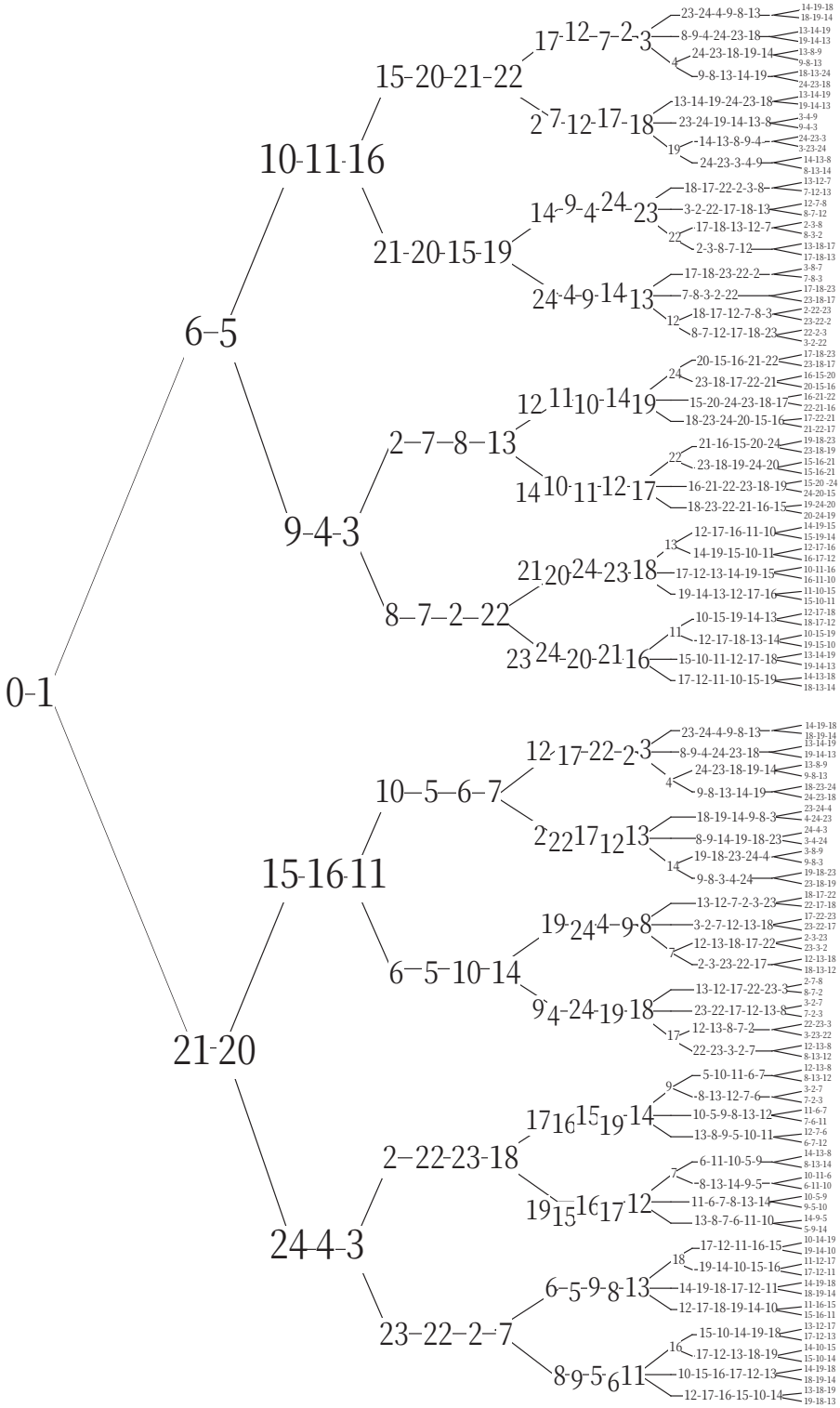

0-1
6–5
21-20
10-11-16
9-4-3
15-16-11
24-4-3
15-20-21-22
21-20-15-19
2–7–8–13
8–7–2–22
10–5–6–7
6–5–10-14
2–22-23-18
23–22–2–7
17-12-7-2-3
2-7-12-17-18
14-9-4-24-23
24-4-9-14-13
12-11-10-14-19
14-10-11-12-17
21-20-24-23-18
23-24-20-21-16
12-17-22-2-3
2-22-17-12-13
19-24-4-9-8
9-4-24-19-18
17-16-15-19-14
19-15-16-17-12
6-5-9-8-13
8-9-5-6-11
23-24-4-9-8-13
8-9-4-24-23-18
24-23-18-19-14
9-8-13-14-19
13-14-19-24-23-18
23-24-19-14-13-8
14-13-8-9-4
24-23-3-4-9
18-17-22-2-3-8
3-2-22-17-18-13
17-18-13-12-7
2-3-8-7-12
17-18-23-22-2
7-8-3-2-22
18-17-12-7-8-3
8-7-12-17-18-23
20-15-16-21-22
23-18-17-22-21
15-20-24-23-18-17
18-23-24-20-15-16
21-16-15-20-24
23-18-19-24-20
16-21-22-23-18-19
18-23-22-21-16-15
12-17-16-11-10
14-19-15-10-11
17-12-13-14-19-15
19-14-13-12-17-16
10-15-19-14-13
12-17-18-13-14
15-10-11-12-17-18
17-12-11-10-15-19
18-19-14
18-19-14
13-14-19
19-14-13
13-8-9
9-8-13
18-13-24
24-23-18
13-14-19
19-14-13
3-4-9
9-4-3
24-23-3
3-23-24
14-13-8
8-13-14
13-12-7
7-12-13
12-7-8
8-7-12
2-3-8
8-3-2
13-18-17
17-18-13
3-8-7
7-8-3
17-18-23
23-18-17
2-22-23
23-22-2
22-2-3
3-2-22
17-18-23
23-18-17
16-15-20
20-15-16
16-21-22
22-21-16
17-22-21
21-22-17
19-18-23
23-18-19
15-16-21
15-16-21
15-20 -24
24-20-15
19-24-20
20-24-19
14-19-15
15-19-14
12-17-16
16-17-12
10-11-16
16-11-10
11-10-15
15-10-11
12-17-18
18-17-12
10-15-19
19-15-10
13-14-19
19-14-13
14-13-18
18-13-14
14-19-18
18-19-14
13-14-19
19-14-13
13-8-9
9-8-13
18-23-24
24-23-18
23-24-4
4-24-23
24-4-3
3-4-24
3-8-9
9-8-3
19-18-23
23-18-19
18-17-22
22-17-18
17-22-23
23-22-17
2-3-23
23-3-2
12-13-18
18-13-12
2-7-8
8-7-2
3-2-7
7-2-3
22-23-3
3-23-22
12-13-8
8-13-12
12-13-8
8-13-12
3-2-7
7-2-3
11-6-7
7-6-11
12-7-6
6-7-12
14-13-8
8-13-14
10-11-6
6-11-10
10-5-9
9-5-10
14-9-5
5-9-14
10-14-19
19-14-10
11-12-17
17-12-11
14-19-18
18-19-14
11-16-15
15-16-11
13-12-17
17-12-13
14-10-15
15-10-14
14-19-18
18-19-14
13-18-19
19-18-13

図 6.8: tree

パターンの解析

今得られた36通りのパターンに共通して言えることは、4つのノードを「コの字型」に接続するパスが多く存在するということである。前にも述べたように影響リンクが最小の「6本」となるのは、隣接する2つのノードを結ぶリンクに沿って張った代替パスを用いてリンク故障を復旧した場合である。では隣接する2ノード間に直接代替パスが張られていないリンクにおいて故障が発生した場合の影響リンク数をできるだけ少なく抑え、平均影響リンク数を最小にするためにはどんなパスを張ればよいのか?

n × n のネットワークを2 × 2 のネットワークの集合と考えた場合、最小規模の2 × 2 のネットワークで平均影響リンク数が最小となる代替パスパターンを、n × n のネットワークの中で組合わせれば、n × n においても平均影響リンク数は最小となるものと推測できる。2 × 2 のネットワークで考えられるパスパターンは図6.10に示す3パターンしかない。各々のパスパターンにおける影響リンク数を表6.4にまとめた。この表を見ても分かるように、パターン1から3の中で平均影響リンク数が最も少ないのはコの字型のパターン1(平均影響リンク数7.25) である。

しかしコの字型をただ単純に何の規則性もなく組み合わせるだけではいけないことは図6.5のパターン10、11、12、13の平均影響リンク数を見れば明らかである。ではこの4つのパターンの違いは何か?

パターンの絞り込みで得られた36通りのパターンを解析した結果、パスの張り方に興味深い規則性が存在することが分かった。それは、どのパターンもノード0を出発してからコの字型に湾曲を続け、ちょうど11本目のリンクから必ず14本目まで直進するということである。言い替えれば、この直進ラインを堺としてネットワークがちょうど半分に分割される。ネットワークを半分に分割することによる効果は、一方の領域で発生するリンク故障がもう一方の領域に影響を及ぼさないということである。

この規則性がここで得られた36通りの制限付きパターンにおいてのみあてはまるのかどうか確かめるために、ランダムに生成したパターンにおいて平均影響リンク数が最小の「19.16」、最大復旧可能リンク数が最大の「12」となったパターンを用いて検証を行なった。

その結果、ランダムパターンにおいても同様の規則性が存在することがわかった。

図6.9において、ランダム、制限付きどちらのパターンにおいてもネットワークは網掛けのノード列を堺として半分に分割されているのが分かる。しかしこのランダムパターンは「コの字型」ばかりで構成されているわけではなく、「斜め結合」や「跳び結合」も多数用いて構成されている。ここで新たに分かることは、最適なパスパターンを張る時の制約としては結合パターンによるよりも、ネットワークを半分に分割するようにパスをはるという制約の方が強いということである。

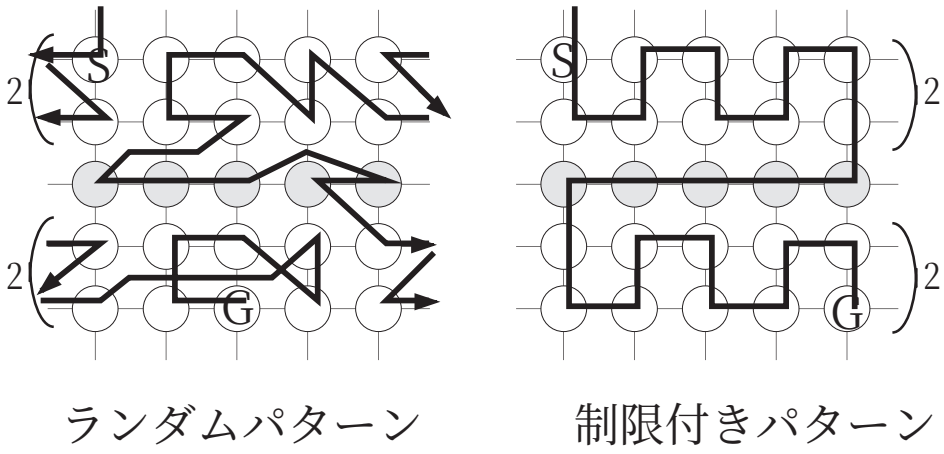


図 6.9: 5×5 の構成におけるパスパターンの規則性

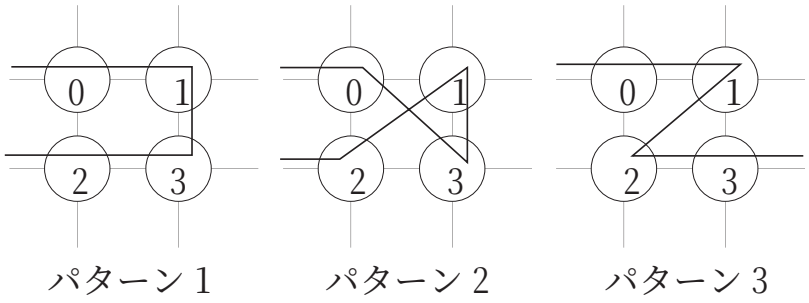


図 6.10: 2 × 2 ネットワークにおけるパスパターン

表 6.4: 各パターンにおける影響リンク数

パターン	影響リンク数				平均影響リンク数
	0-1 間故障	0-2 間故障	1-3 間故障	2-3 間故障	
1	6	11	6	6	7.25
2	9	11	6	9	8.75
3	6	9	9	6	7.5

6.1.4 ネットワークサイズの違いについて

本研究では主にネットワーク構成を 5 × 5 として話を展開してきたが、この規則性が 5×5 の構成についてのみ言えるのか、それとも一般的な n×n の構成についても言えるのか検証する。まず図 6.11に示した 3×3 の構成において、ランダムに生成したパターン、制限付きで生成したパターンどちらにおいても平均影響リンク数は最小の「11」、最大復旧可能リンク数は最大の「4」である。その結果、3×3 のパターンにおいても網掛けのノード列によってネットワークは半分に分割された。

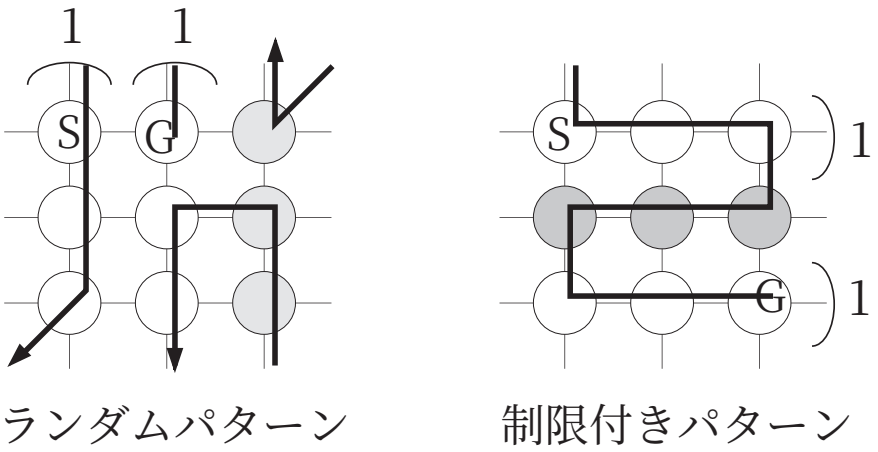


図 6.11: 3×3 の構成におけるパスパターンの規則性

次に n=7 のネットワーク構成について同様に調べた。図 6.12 において、ランダムに生成したパターンの最小平均影響リンク数は「26.918」となり、制限付きで生成したパターンの最小平均影響リンク数は「26.837」となった。制限付きパターンにおいてはこの値以下にはならなかったが、ランダムパターンにおいては測定する度に最小値が変わり、一定しなかった。これは 6.1.1でも述べたように、乱数を用いることによって「local minimum」に値が落ち込むことによる影響と思われる。従ってこの値を最小値と断定することはできないが、この値においてもネットワークは網掛けのノード列によって二分割されていることが分かった。

また、7×7 の構成で復旧できるリンク故障の最大数は「24」であるが、ランダムなパターン生成においては最大数が「23」の場合の平均影響リンク数が「24」の場合の平均影響リンク数を下回った (26.755)。この場合においても「二分割の法則」は当てはまるが、東西南北に隣接するパスの数が少なくなった。

これらのことから推測できることは、

1. 平均影響リンク数を最小とするためには、ネットワークがあるノード列によって二分割されるようにパスを張らなければならない。

2. 最大復旧可能リンク数をそれぞれのネットワーク構成における MAX 値にするためには、パスは制限

付きで張る方がよい。

ということである。最適なパスを張る際に、1 は十分条件となり、2 は必要条件となるものと推測できる。

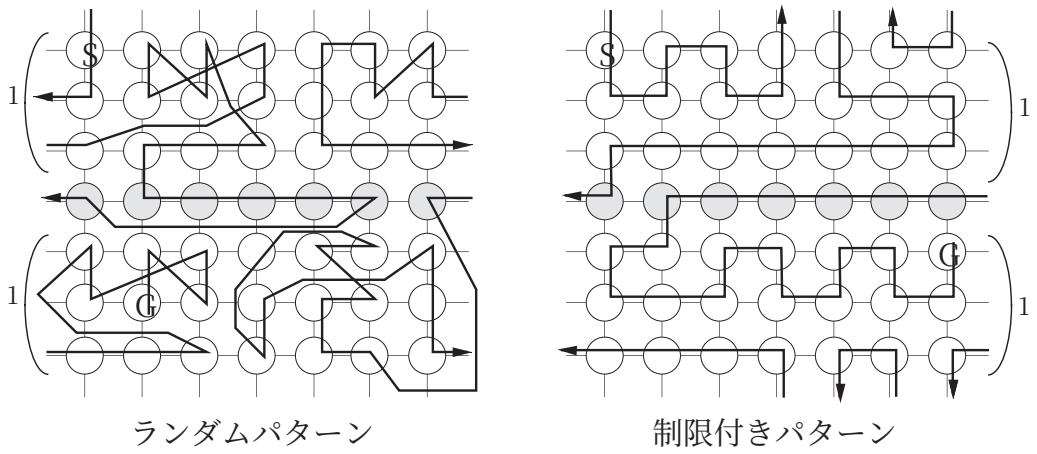


図 6.12: 7×7 の構成におけるパスパターンの規則性

では n×n の構成で n が偶数の場合はどのような規則性が存在するだろうか? n が 4 と 6 の場合について調べてみた。その結果、4×4 の場合にはネットワークはあるノード列によって 1:2 に分割されることが分かった。また、6×6 の構成についてはあるノード列によってネットワークが 2:3 に分割されることが分かった。このことから、n が偶数のネットワークにおいては n が大きくなるにつれ、つまりネットワークの規模が大きくなるにつれノード列による分割比は 1:1 に近付く傾向にあることが分かった (図 6.13、図 6.14 参照)。

これらの傾向は n が 3、4、5、6、7 の 5 種類のネットワークに言えただけであり、全ての n に対して言えるとは限らないが、ここではこの傾向を元に得られたパターンを最適パターンとして扱うものとする。

ここで得られた最適パターンの信頼性の評価は第 7 章で行なう。

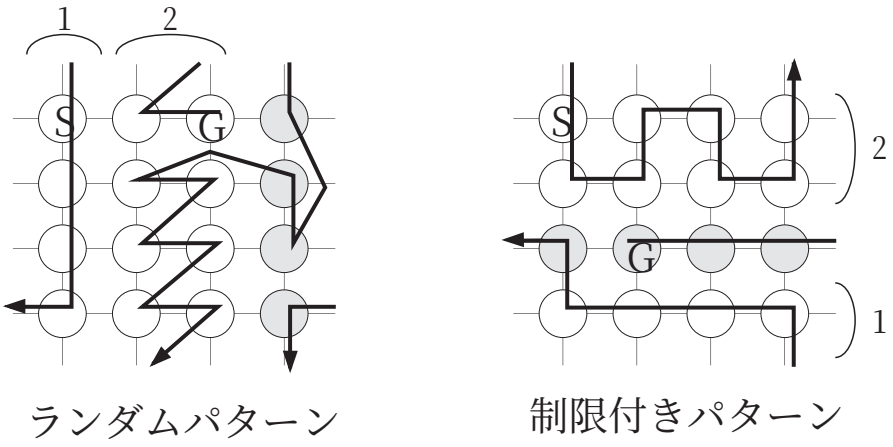


図 6.13: 4×4 の構成におけるパスパターンの規則性

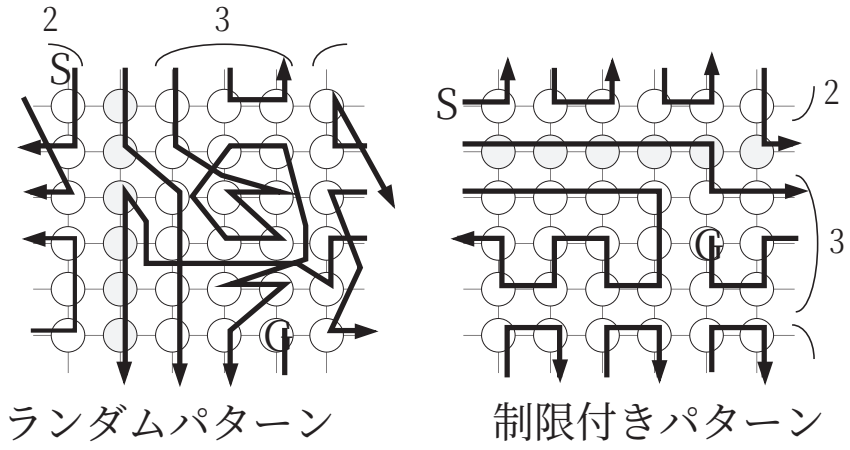


図 6.14: 6×6 の構成におけるパスパターンの規則性

6.2 二次元 4 隣接トーラスでの複数回書き換えに対する考察

ノードの書き換えは複数回行なうことによって復旧可能なリンク数は増加し、システムの信頼性も向上する。

ノードの書き換えを複数回行なうためには、分散故障方式においては初めからノードは複数回書き換え可能であるため問題はないが、集中故障復旧方式ではコンフィギュレーションラインとは別に代替ラインを 1 本設ければよいことは 4.3.1、4.3.2で述べた。

この方式においてパスパターンの絞り込みを行なう際には、複数書き換えにおける平均影響リンク数を用いる必要はなく 1 回だけ書き換える場合の平均影響リンク数をそのまま用いればよい。なぜなら、ノードの書き換えを複数回行なうことによって平均影響リンク数は減少するが、どのリンクにおいてもその値は一様に減少するからである。

しかし信頼性を算出する場合はこの限りでなく、2 度書き換えた場合の平均影響リンク数を考慮しなければならない。

第 7 章

考察

多数存在する代替パスパターンから最適なパスパターンを絞り込むために、5.2.2で「平均影響リンク数」という評価指標を設けた。この値はパターンを絞り込む際の基準として、あるいは単純にパスパターンの善し悪しを判断する時の目安として使うことはできたが、厳密にシステムの信頼性について言及する時の目安として使えるかどうかは分からない。

従ってシステムの信頼性の評価はMTTF のような一般的な評価指標を導入する必要がある。その上で評価指標として用いた影響リンク数が計算した信頼性にどれだけ見合うのかを考察する。

7.1 信頼性に関する考察

7.1.1 信頼性モデル

本研究におけるリンク故障復旧策はコンフィギュレーションラインを代替パスとして故障を回避する方法であり、故障リンクの修理は考えていない。従って修復システムとしてではなく縮退システムとしてマルコフモデルを元に、システムの $MTTF_{system}$を求める。

リンク総数をN、リンク一本の故障率を$\lambda(=1/MTTF_{link})$、あるリンクが i 番目に故障した場合に全てのノード間の接続が失われていない確率を c_iとすると、システムのモデル図は図7.1のようになる。リンク故障数に関するカバレッジは全てのリンク故障パターンの中からノード間の接続が失われないパターン数を数えあげることにより算出できる。

1 つ目の故障は、どのリンクが故障しても代替パスによって必ず復旧できるので c_1は 1 となる。図中の S_iはシステム内に i 本のリンク故障が存在し、なおかつ全てのノード間の接続が失われていない状態、S_F はノード間の接続が 1 本でも失われてしまった状態を表す。n は最大 n 個のリンク故障でもノード間の接続は失われていないことを示す。

$MTTF$の計算は、DR-net のディスク故障に関する信頼性の評価でも行なわれており [8, 9, 10] 同じ非修

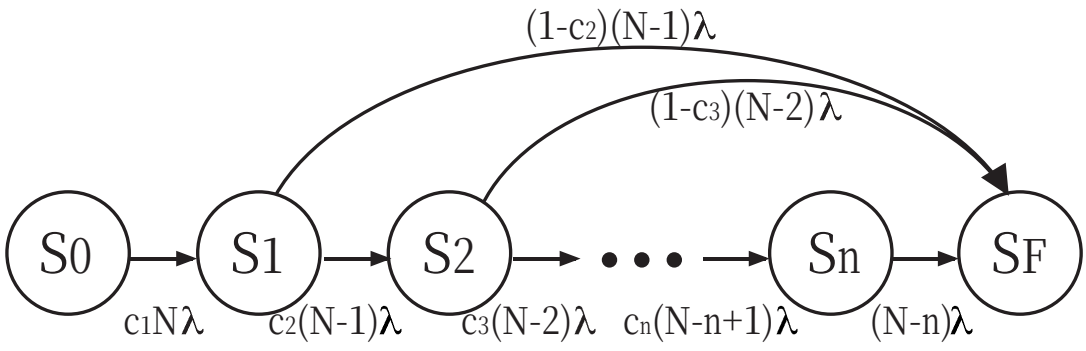


図 7.1: マルコフモデル

理系の縮退システムとして計算されているので、ここではその計算式をそのまま用いて $MTTF$ を計算することにする。以下その計算式を示す。

$$
\begin{aligned}
MTTF &= \int_0^\infty R(t)dt \\
&= \sum_{i=0}^{n} \int_0^\infty p_i(t)dt \\
&= \frac{1}{N\lambda} + \sum_{i=1}^{n} \left(\prod_{j=1}^{i} c_j \right) \frac{1}{(N-i)\lambda}
\end{aligned} \tag{7.1}
$$

7.1.2 集中故障復旧方式を前提とした場合の信頼性

集中故障復旧方式では、コンフィギュレーションラインに故障はないものと仮定し、従って代替パスも故障しないものと仮定する。一般の N について c_i を求めるのは困難なので、ここでは 3 × 3 の構成について考える。ここでは平均影響リンク数が比較的高い「渦巻きパターン」と、6.1.3で得られた規則性を基にしてパスを張った「最適パターン」と、リンク故障復旧において一番信頼性が高いと思われる「二重化パターン」の 3 種類のパターンについてそれぞれカバレッジを求めた (表 7.1,7.2,7.3)。3 × 3 では総リンク数 N は 18、そして最大で 4 個のリンク故障を復旧できるため n=4 となる。これらにもとづき (7.1) 式を用いて MTTF を算出すると

$$
\begin{aligned}
MTTF_{convolution} &= \frac{0.136}{\lambda} \\
MTTF_{best} &= \frac{0.137}{\lambda} \\
MTTF_{double} &= \frac{0.165}{\lambda}
\end{aligned}
$$

となる。

表 7.1: 渦巻きパスパターンのカバレッジ (c_i)

リンク故障数	全リンク故障パターン数	全ノード間通信が可能なパターン数	カバレッジ (c_i)
1	18	18	1.00
2	306	198	0.353
3	4896	271	0.0554
4	73440	144	0.00196

表 7.2: 最適パスパターンのカバレッジ (c_i)

リンク故障数	全リンク故障パターン数	全ノード間通信が可能なパターン数	カバレッジ (c_i)
1	18	18	1.00
2	306	206	0.327
3	4896	235	0.0480
4	73440	120	0.00163

また、代替パスを張らない場合、つまりリンク故障の復旧を行なわない場合は $n = 0$ であり、これを式 (7.1) に代入してこの場合の $MTTF$ を計算すると、

$$\begin{aligned} MTTF_{no-path} &= \frac{1}{N\lambda} \\ &= \frac{0.0556}{\lambda} \end{aligned}$$

となる。

$MTTF$ を長い順にまとめたのが表 7.4である。平均影響リンク数との対応を見るためにこの値も並べて示す。

考察

最初に評価指標として用いた「平均影響リンク数」の有効性を検証する。

二重化パターンに対して最適パターンの $MTTF$ は約 5/6 倍になるが、平均影響リンク数は約 6/5 倍とはなっていない (約 2 倍)。従って「平均影響リンク数」はパターンを絞り込む際の 1 つの目安としては十分用いることができるが、信頼性の評価として用いることはできない。

次に本研究で得られた最適パターンと、リンクのコストは高いが信頼性的には一番高い二重化パターン

表 7.3: 全二重化パターンのカバレッジ

リンク故障数	全リンク故障パターン数	全ノード間通信が可能なパターン数	カバレッジ (c_i)
1	18	18	1.00
2	306	198	0.647
3	4896	1116	0.228
4	73440	1320	0.0180

表 7.4: 各パターンの MTTF

構成	MTTF	平均許容リンク故障数	平均影響リンク数
二重化パターン	0.165MTTF	2.97	6.00
最適パターン	0.137MTTF	2.47	11.00
渦巻きパターン	0.136MTTF	2.45	11.44
代替パスなし	0.0556MTTF	-	-

との信頼性の比較を行なう。

7.4において、代替パスを張らない場合の $MTTF$ に対して二重化パターンの $MTTF$ は約 3 倍となっている。これはリンク二重化することによりシステムの信頼度は 3 倍向上することを意味する。また最適パターンにおいては約 2.5 倍向上する。これにより 1 本のラインを代替パスとして用いた場合、全てのリンクを二重化する半分のコストでその約 83%の信頼性を獲得できることになる。

よって本研究で提案した手法は、コストよりも信頼性を重視する場合には二重化に比べて若干劣ることになるが、コストを重視した場合に二重化の半分のリンクコストで 83%の信頼性が得られれば十分有用な復旧手段であると言えるのではないだろうか。

7.1.3 分散故障復旧方式を前提とした場合の信頼性

分散故障復旧方式によってリンク故障の復旧を行なう場合は、4.3.1で述べたようなコンフィギュレーションラインの占有問題を心配しなくてもよいため、代替パスの故障も考慮して信頼性を計算する必要がある。

対象とするネットワーク構成はここでも 3 × 3 のトーラス結合網とした。代替パスの故障も考慮したマルコフモデルを図 7.2に示す。ここで通常リンク本数は 18 本、代替リンク本数は 9 本であるので総リンク数 N は 27 である。そのうち 8 本のリンクが切れても各ノード間の接続は失われないので n=8 となり、カバレッジは c_8 まで計算する必要がある。

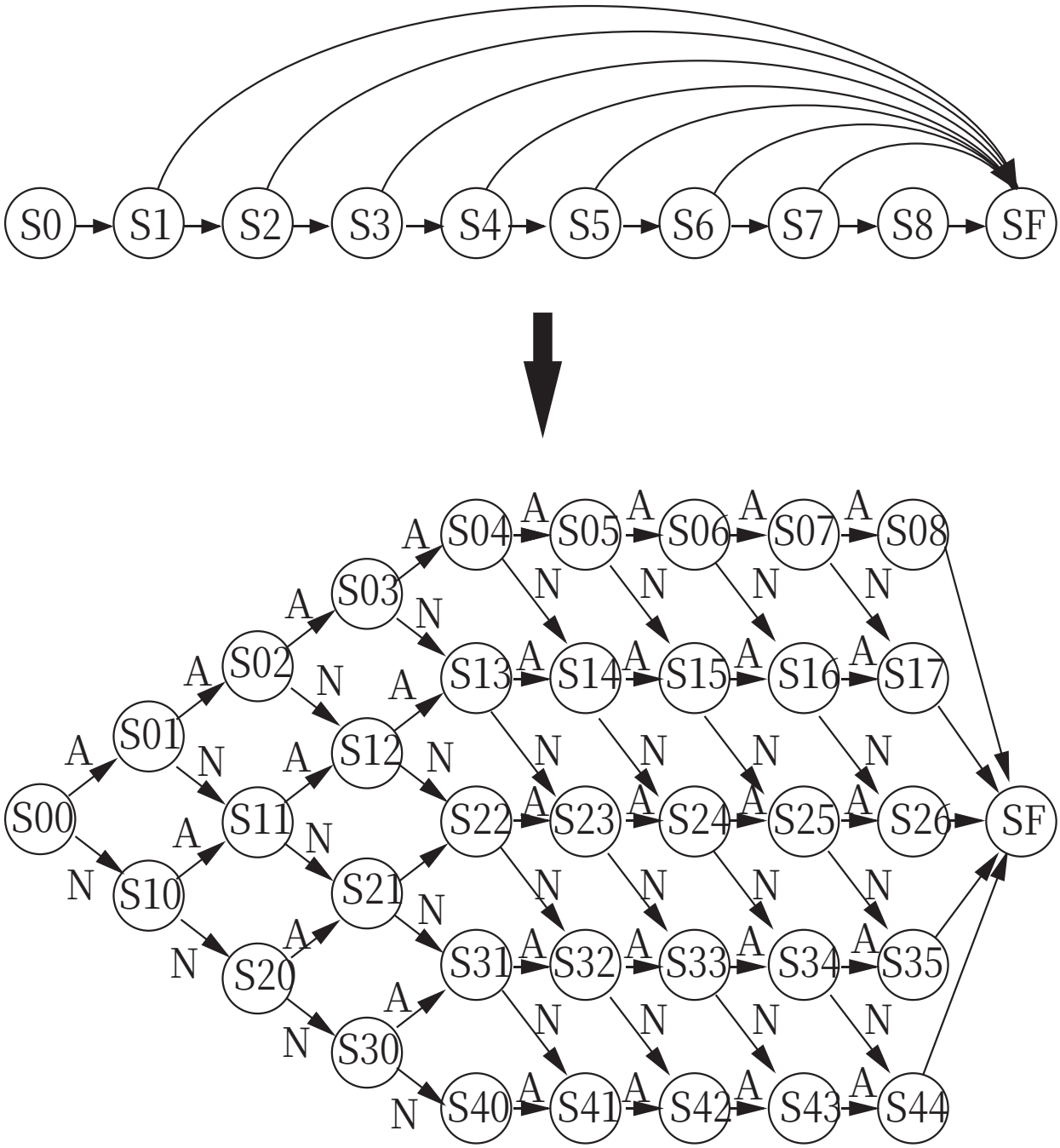
また、このモデルの場合代替として用いるリンクと通常のリンクとでは故障した時に回りのノードに及ぼす影響が違うため、代替リンクが切れた場合と通常リンクが切れた場合とに場合分けして考えなければならない。そうすることにより、マルコフモデルは図 7.2の下側に示したような複雑な形になる。代替パスが故障した時は「A」の方向に、そして通常リンクが故障した時は「N」の方向に遷移する。

図において、S_{ij} はシステム内に i 本の通常リンク故障と j 本の代替リンク故障が発生した時に全てのノード間の接続が失われていない状態を示す。

この図からカバレッジを計算する時は、代替リンクが故障した時のカレバッジと通常リンクが故障した時とに場合分けして求めたカバレッジを、一列分全部足し合わせることにより求めることができる。

例えば c_2 を計算したいとする。c_2 は「リンクが 2 本切れてもノード間の接続が失われない確率」であり、これを求めるには S02,S11,S20 への全遷移パターンを求め、その中で 2 本切れても大丈夫なパターン数を数え上げ、それぞれについてカバレッジを計算してそれらを足し合わせればよい。

現時点では、時間の都合上まだカバレッジの計算が終わっていないので、分散故障復旧方式における信頼性 ($MTTF$) を評価できない。これについては今後の課題としたい。



A : Additional link failure
N : Normal link failure

図 7.2: 分散故障復旧方式におけるマルコフモデル

7.2 その他の静的相互結合網に対する考察

従来のトーラスネットワークにおいて、リンクのコストは各ノード間で一定ではない。図 7.3で太線部分のリンクはネットワークの両端のノードを接続するリンクであり、その他のリンクに比べて数倍長くなっているのが分かる。

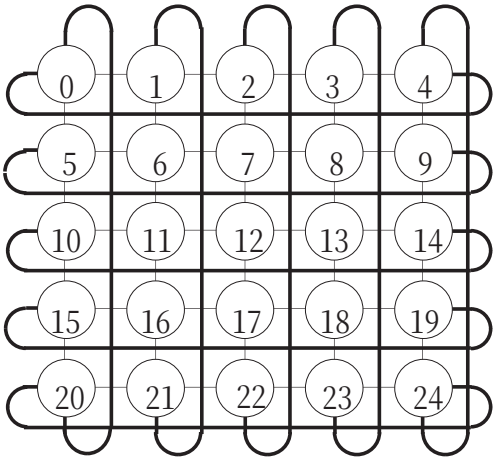


図 7.3: 従来のトーラス結合網

本研究においてリンクの障害率はどのリンクにおいても一定と仮定したが、実際この太線部分のリンクの障害率はその他のリンクに比べて高くなると思われる。従ってリンクの長さは全てのリンクにおいて一定とすべきである。この問題の解決策の一つとして、[7] で「folded torus」が提案されている。

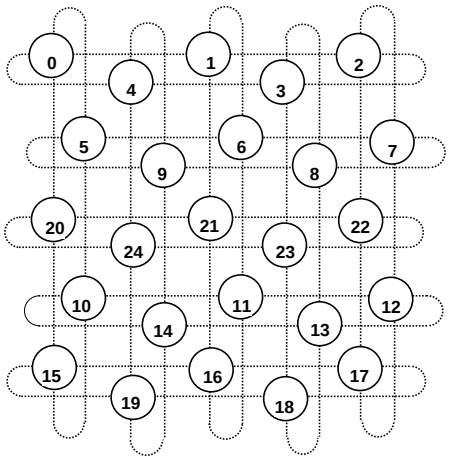


図 7.4: folded torus system

図7.4のようにノードの配置を変えることで各ノード間のリンクコストは同一になる。図をみても分かるように各ノードに隣接する4つのノードは番号的に従来のトーラス結合網と変わらない。従ってリンク故障に関するシステムの信頼性も従来のトーラスと変わりはない。

第 8 章

結論

8.1 本研究の成果

本研究では、FPGA をノードプロセッサとして使用した n × n のトーラス結合網において、FPGA の「回路の書き換え可能」という特徴を活かしたこれまでにない全く新しい概念のリンク故障復旧方法を提案し、その信頼性を評価した。

リンク故障復旧戦略としては、故障の検出、通知、そして復旧を制御ホストで一括管理して行なう「集中リンク故障復旧方式」と、各 FPGA に ROM を接続し、書き換えに伴う一連の処理をロードしておくことで各ノード毎にローカルに復旧処理を行なうことができる「分散リンク故障復旧方式」を考えた。どちらの方式においてもリンクの故障復旧には 1 本の代替ラインを用い、リンク故障が発生した時は故障リンクの両サイドのノードにおけるリンクの接続情報を故障リンクから代替リンクへ書き換え、その情報を再び FPGA にダウンロードし直すことにより故障が復旧できることを示した。

また代替パスの張り方次第で復旧できるリンク数が変化するなど故障復旧に関して特性が変わるため、最適な代替パスの張り方を明らかにする必要があった。そこで数多く存在する代替パスパターンの中から最適なパスパターンを絞り込んでいくために、5 × 5 のトーラスネットワークを対象にしてトポロジカルな絞り込みと評価指標による絞り込みを行なった。ここでは、リンク故障を復旧するためにノードで一回書換えを行なった場合、代替リンクのパターンによっては次回復旧できないリンクが存在し、そういったリンクの数を「影響リンク数」という名の一つの評価指標としてパターンの絞り込みを行なった。

そして絞り込みによって得られた 36 通りのパスパターンを解析した結果 n×n のトーラスネットワークにおける最適パターンの張り方の規則性を発見することができた。

最後に 3 × 3 のネットワークを対象に、得られた最適パターンと静的相互結合網においてリンク故障における信頼性が最も高いと思われる二重化パターンとの MTTF による信頼性の比較を行なった。コンフィギュレーションラインは故障しないとする集中故障復旧方式を前提として MTTF を求め、二重化の半分の

リンクコストでその 83%の信頼性を獲得できることを明らかにした。

現時点では 3×3 の構成における最適パスパターンの信頼性しか計算していないが、ネットワークの規模が大きくなればなるほど最適パスを張ることの意義が増大し、システムの信頼性にも大きく影響してくるものと考えられる。

8.2 今後の課題

今後の課題としては、信頼性の面で分散故障復旧方式を前提とした場合の $MTTF$ を算出し、集中故障復旧方式との信頼性の比較を行なう必要がある。

ネットワークの規模も 3 × 3 や 5 × 5 といった特定の規模ではなく一般的な規模で比較できることが望ましい。ネットワーク構成も 4 隣接のトーラスだけでなく、8 隣接やハイパーキューブなど、様々なネットワークトポロジについても検討が必要である。

また、本研究で提案した手法を実際に FPGA 実験装置に実装し動作確認することも必要である。

謝辞

本研究を行なうに当たり、終始熱心な御指導を賜わりました横田治夫助教授に心から感謝致します。

また、適切な御助言を頂きました日比野靖教授に深く感謝致します。

さらに、貴重な御意見を頂きました横田研究室、日比野研究室の皆様を初め、多くの方々の御援助に対し深く感謝申し上げます。

参考文献

[1] Dilip D.Kandlur,*Member,IEEE*,and Kang G.Shin,*Fellow,IEEE* Traffic Routing for Multicomputer Networks with Virtual Cut-through Capability. In *IEEE Trans.Computers*,Vol.41,No.10,pp.1257-1270,Oct 1992.

[2] NATHANIEL J.DAVIS IV,STUDENT MEMBER,IEEE,WILLIAM TSUN-YUK HSU,AND HOWARD JAY SIEGEL,SENIOR MEMBER,IEEE. Fault Location Techniques for Distributed Control Interconnection Networks. In *IEEE Trans.Computers*,Vol.C-34,No.10,pp.902-910,Oct 1985.

[3] Andreas Bauch and Erik Maehle. Reconfiguration in Octagonal Mesh-Based Multicomputer Systems with Distributed Checkpointing. IEEE,1995

[4] JIAN LU and TAIICHI YUASA. A New Algorithm for 2D Convolution on Mesh-connected SIMD Computers. Transaction of Information Processing Society of Japan,Vol.37,No.12,Dec 1996.

[5] S.-Y.LEE AND J.K.AGGARWAL. Parallel 2-D Convolution on a Mesh Connected Array Processor. In *Proc.*IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,VOL.PAMI-9,NO.4,pp.590-595,July 1987.

[6] Frang,Z.,Li,X.and Ni,L.M. On the Communication Complexity oc Generalized 2-D Convolution on Array Processor. *IEEE Trans.Comput.*,Vol.38,No.2,pp.184-194,1989.

[7] William J.Dally. A VLSI Architecture for Concurrent Data Structures. Kluwer Academic Publishers,1987.

[8] 横田治夫. RAID のネットワーク上への展開と信頼性向上. 信学技法 CPSY 93-11, 電子情報通信学会,Apr 1993.

[9] 味松康行, 横田治夫. 並列ディスクシステムのパリティグループの構成の変化と信頼性の比較. 信学技法 FTS95-344, 電子情報通信学会,Aug 1995.

[10] 味松康行. ディスクシステムの高速化および信頼性向上に関する研究. Master's thesis, 北陸先端科学技術大学院大学,1995.

[11] 浅野俊雄. FPGA 利用による相互結合ネットワークの柔軟構成化に関する研究. Master's thesis, 北陸先端科学技術大学院大学,1995.

[12] 末吉敏則. 教育への FPGA 応用例.In 情報処理,page 519-529,Jun 1994.

[13] 末吉敏則他. 上級コース向きマイクロプロセッサ DLX-FPGA の設計と実績.In 信学技法, page 17-24,Sep 1994