

## 増殖による回路の構成方法

矢野 智史<sup>†</sup> 樋口 隼人<sup>†</sup> 永本 太一<sup>†</sup> 柴田裕一郎<sup>†</sup> 小栗 清<sup>†</sup>

<sup>†</sup>長崎大学, 長崎市

E-mail: [tyano@pca.cis.nagasaki-u.ac.jp](mailto:tyano@pca.cis.nagasaki-u.ac.jp)

**あらまし** 最大3本のリンクを持つノードとリンクからなる任意の連結グラフは、単一のノードから初めて3つの増殖ルール（隣接部分へのノードの挿入、隣接リンクの挿入、隣接ノードのリンクの交換）を逐次適用することによって構成することができる、ということを示した。この構成方法では、細胞増殖によって生物組織が形成されるように、基本要素（ノード）の増殖により回路全体が構成されていくことになる。

**キーワード** グラフ理論, 動的再構成デバイス, 増殖

## Construction Method for a Circuit by Multiplication

Satoshi YANO<sup>†</sup>, Hayato HIGUCHI<sup>†</sup>, Taichi NAGAMOTO<sup>†</sup>, Yuichiro SHIBATA<sup>†</sup>, and Kiyoshi OGURI<sup>†</sup>

<sup>†</sup> Graduate School of Science and Technology, Nagasaki University Bunkyo 1-14, Nagasaki-shi, Nagasaki, 852-8521 Japan

E-mail: [tyano@pca.cis.nagasaki-u.ac.jp](mailto:tyano@pca.cis.nagasaki-u.ac.jp)

**Abstract** It shows that a connected graph which is composed of nodes which with max three links can be constructed to apply three multiplication rules(insert a node to the next, insert link to the next, exchange each one link of two connected node) successively. This construct method constructs a whole circuit by the multiplication of base elements(nodes) like tissues are built by the cell proliferation.

**Key words** Graph Theory, Dynamic Reconfigurable Device, Multiplication

### 1. まえがき

計算機の汎用性はプロセッサとメモリによるプログラム論理により達成され、その性能は今日に至るまでの半導体集積技術の向上に支えられてきた。しかしながら、その動作周波数の向上はひとつの壁を迎えつつあり、多くのプロセッサが単一コアによる性能向上ではなくマルチコアによるスループット向上を目指すようになってきている。一方、PLD [1] や FPGA [2], [3] と呼ばれるプログラム可能なハードウェアが注目されている。これらはプログラム論理ではなく、より高速な布線論理に汎用性を与えるものであり、DRP [4] などの自律的な性質をもつ動的再構成アーキテクチャの登場により計算機の新しい道が見出されようとしている。

汎用であるということは即ち新たな動作を与える余地が残されているということであり、自由に使用できるリソースを自律的に管理する能力をもっているということである。現行の動的再構成アーキテクチャはこの点でフォンノイマンアーキテクチャにはまだ及ばない。PCA(Plastic Cell Architecture) [5], [6] は、布線論理に十分な柔軟性を与えることを目指した LSI アーキ

テクチャである。一様に並んだ可変部とそれらを管理しネットワークを形成するための固定部の協調動作により回路を改変、増殖させることができる。しかし増殖機能については解決すべき問題が残されており、PCA にとって最大の課題となっている。そこで本稿では生物の細胞増殖を真似た回路構成法を新たに考え、その基礎となる問題の設定とその解について述べる。

### 2. グラフを用いた問題の設定

PCA は固定部と可変部からなる PCA セルを二次元メッシュ状に並べた構成を持ち、隣り合った PCA セル同士が通信しながら回路を構成していく。回路間の通信は主に固定部に組み込まれたコマンド群によってなされ、そのうち回路の追加、削除に関するコマンドにより新規回路を要求したセルの直近に空きスペースが作られ、そこに回路が作成される。これは生物の細胞増殖を模した無理のない回路の増殖を目指した結果であり、領域の確保に関する調停部を必要としない点で有利である。

以上に述べた増殖機能を前提とし、任意の回路を構成するにはどのようにすればよいかを明らかにすることを本論文のテーマとする。

## 2.1 ノード

回路はノードが多数接続されたものであるとし、ノードを1個ずつ追加していくことで、ある1つの回路全体が構成されるものとする。

## 2.2 初期状態

図1のようにループを1つ持つノードを初期状態とし、この状態から回路を増殖構成していくこととする。このノードは回路の外界を表しており、リンクの一方が外界からの入力、他方が外界への出力を表している。



図1 初期状態  
Fig.1 Initial State

## 2.3 無向グラフ

回路は当然入出力があるので有向グラフとして考える必要があるが、リンクの向きは最初は考えずに議論を進められることが分かったので、当面無向グラフで考えることとする。

## 2.4 ノードの種類

ノードの次数を大きくすると問題が複雑になるので、できるだけ小さい次数のノードだけを考えることにする。任意の機能を実現するには1入力1出力、1入力2出力、2入力1出力の3種類(図2)あればよい。この3種類のパターンを用いればそれ以上の入出力をもつ回路も構成できるからである。

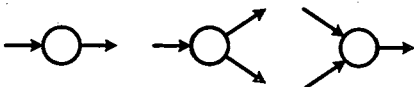


図2 ノードの入出力数  
Fig.2 The Number of Inputs and Outputs of a Node

また、リンクの向きは考えないので、1入力1出力のノードは次数2のノード、1入力2出力のノードと2入力1出力のノードは次数3のノードとなる(図3)。次数1のノードでは

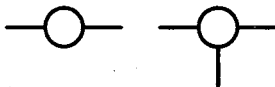


図3 ノードの種類  
Fig.3 Kind of Nodes

連結したグラフが作れないので、そのようなノードは考えないこととする。

## 2.5 ルールの適用範囲

回路の増殖は、全体が見渡せる立場で行うのではなくあくまで局所的な情報を使って行わなければならない。すなわち、着目点となるノードと、そのノードに隣接しているノードの情報のみで増殖を行うこととする。隣接している範囲はできるだけ狭いことが望ましい。また、着目点となったノードおよび隣接ノードが持つ情報による増殖が完了すると、着目点を移動し、同様に増殖を続けていくこととする。

## 2.6 問題の定義

ここまでで、我々が明らかにしたいことは、“次数2のノードを最低でも1個含む次数2または次数3のノードからなる任意の連結グラフを図1の初期状態から始めて、局所的な増殖ルールを適用することによって作ることができるか”ということとなる、あるいは“次数2のノードを最低でも1個含む次数2または次数3のノードからなる任意の連結グラフに逆ルールを逆順序で適用して、図1の初期状態にすることができるか”となる。

## 3. 増殖ルール

2.6の解となるルールおよび逆ルールを示す。

増殖ルールは、

- ノードの挿入
- リンクの追加
- リンクの交換

の3つがあればよい。以下に、この3つのルールの詳細と各々の逆ルールを示す。

“ノードの挿入”は、あるノード間に張られたリンク上に新しいノードを1つ挿入することである。“ノードの挿入”の逆は、次数2のノードを削除することができる“ノードの削除”である。図4は、ノード1とノード2の間のリンクのノードAを挿入・削除している。

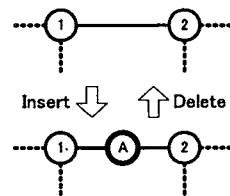


図4 ノードの挿入・削除  
Fig.4 Insert, Delete a Node

“リンクの追加”は、隣接する2つのノードの次数が共に2である場合、2つのノード間にもう1本リンクを追加することである。“リンクの追加”の逆は、多重辺のリンクを1本削除する“リンクの削除”である。図5は、隣接する次数2のノードAとノードBにリンクを追加・削除している。

“リンクの交換”は、隣接する2つのノードの次数が共に3である場合、2つのノードをつなぐリンク以外のリンクのうちそれぞれ1つずつ交換することである。交換するリンクの選び方によって、 $2 \times 2 = 4$ 通りの交換ができる。“リンクの交

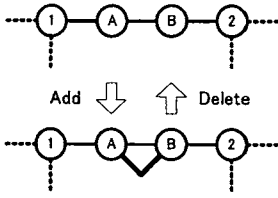


図5 リンクの追加・削除  
Fig. 5 Add, Delete a Link

換”の逆は、交換したリンクを元に戻すだけなので“リンクの交換”である。図6は隣接する次数3のノードAとノードBにおけるリンクの交換を示している。(a)で交換しているのは、ノードAのノード1へのリンクとノードBのノード2へのリンク、(b)で交換しているのは、ノードAのノード3へのリンクとノードBのノード2へのリンク、(c)で交換しているのは、ノードAのノード1へのリンクとノードBのノード4へのリンク、(d)で交換しているのは、ノードAのノード3へのリンクとノードBのノード4へのリンクである。

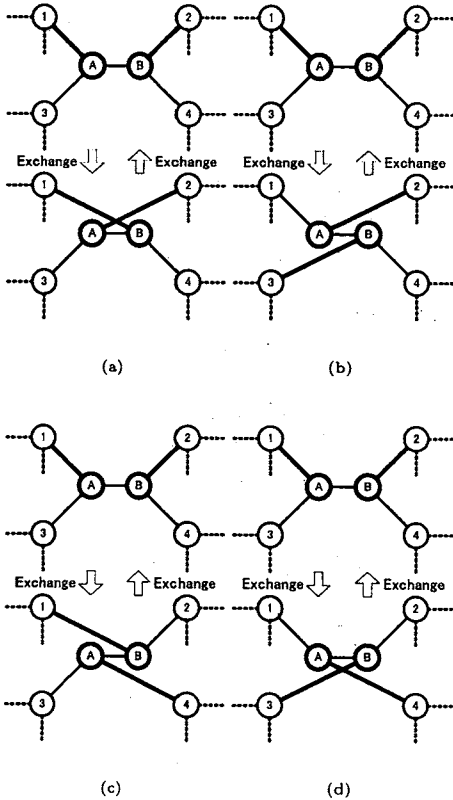


図6 リンクの交換  
Fig. 6 Exchange Links

また、この“リンクの交換”を用いることによってループを作ることができる。図7は、ノードAとノードBが多重辺になっており、ノードAのノードBへのリンクとノードBのノード

2へのリンクを交換している。こうすることによってノードBにループをつなぐことができる。

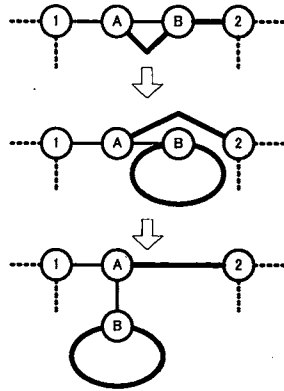


図7 ループ  
Fig. 7 Loop

以上よりまとめると、増殖ルールの逆ルールは、

- ノードの削除
- リンクの削除
- リンクの交換

となる。

#### 4. 証明

次数2のノードを最低でも1個含む次数2または次数3のノードからなる任意の連結グラフに3.章で示した逆ルールを順番に適用して、図1の初期状態にすることができることを証明する。

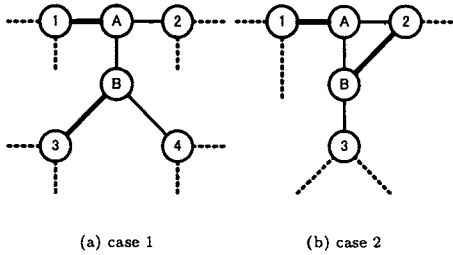
先に、次数2のノードは図4より、“ノードの削除”で消すことができるので、以降で対象とするグラフは次数3のノードのみからなるグラフとして一般性を失われない。

まず、“すべてのノードは主経路上に持って行くことができる”ということを証明する。初期状態(図1)のノードの端から端までつながった経路を任意に1本選び、主経路(図8)と呼ぶことにする。主経路を選ぶことでノードは、主経路上のノ

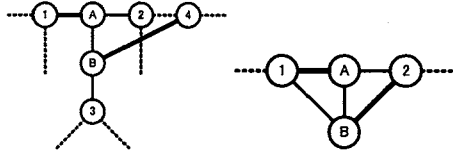


図8 主経路  
Fig. 8 Main Route

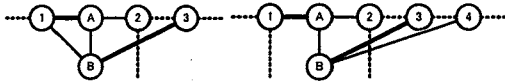
ドと主経路上にないノードに分かれる。主経路上にないノードのうち主経路上のノードに隣接しているノードを考える。以下に示すように、“リンクの交換”を適用することで主経路上に持って行くことができる。そのノードの接続のされ方は、以下の6通り(図9)である。そして、いずれの場合も太線のリンクを交換することによってノードBを主経路上に持って行くことができる。例として、図9の(a)のノードBを主経路上に



(a) case 1 (b) case 2



(c) case 3 (d) case 4



(e) case 5 (f) case 6

図 9 6 通りの場合

Fig. 9 6 cases

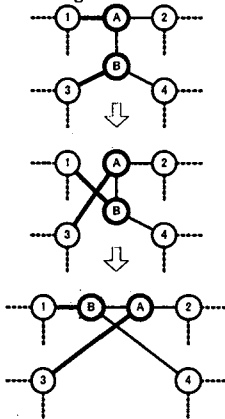


図 10 場合 1

Fig. 10 case 1

持って行く過程を図 10 に示す。以上を繰り返すことによって、主経路上にない全てのノードを主経路上に持って行くことができる。

次に、“主経路上のノードはすべて多重辺を持つノードにすることができる”ということを証明する。主経路上にノードを持って行くことで多重辺を持つノードができ、削除できる場合があるが、たいていのノードは図 11 のようにリンクが交差している。ここでは、リンクの交差をすべて取り除くことができることを示す。図 12 は、ノード A とノード B が主経路上で隣



図 11 交換後の主経路

Fig. 11 Main Route after Exchanging Links

接しており、ノード A とノード B のリンクが交差している。この場合、ノード A のノード 1 へのリンクとノード B のノード 2 へのリンクを交換すると、交差が解ける。もし、図 12 のノ

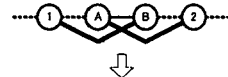
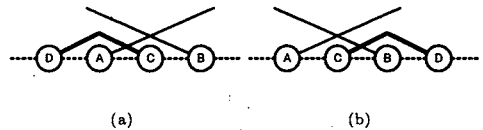


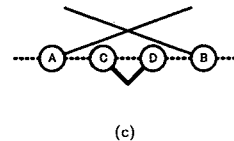
図 12 リンクの交差 1

Fig. 12 Cross of Links No.1

ド A とノード B の間にノード (ノード C) があるとする (図 13)。ノード C のリンク先のノードは以下の 3 通りに分類できる。(a) はノード C のリンク先であるノード D がノード A の外、(b) はノード B の外、(c) はノード A とノード B の間である。(a) の場合はノード A とノード C、(b) の場合はノード A



(a) (b)



(c)

図 13 リンクの交差 2

Fig. 13 Cross Links No.2

とノード C との間で同じことを着目する 2 つのノードが隣接するまで繰り返す、隣接するノードが見つければ図 12 のようにして交差を解く。以上を繰り返すことによって、図 14 のような交差のない主経路にすることができる。



図 14 交差のない主経路

Fig. 14 Main Route without Cross Links

最後に、“多重辺を持つノードは削除することができる”ことを証明する。図 15 は、多重辺のノードであるノード A とノード B を削除する過程を示している。まず、“リンクの削除”によってノード A とノード B をつなぐ 2 本のリンクのうちの 1 本を削除する。これで、ノード A もノード B も次数が 2 になるのであるとは“ノードの削除”によって 1 つずつ消すことができ

る。よって、多重辺を持つノードは削除できることがわかった。

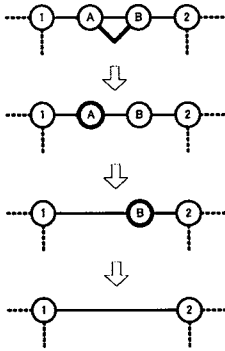


図 15 多重辺のノードの削除

Fig. 15 Delete Nodes with Multiedge

以上より、逆ルールを適用することによって任意のトポロジを初期状態（図 1）に戻すことができることがわかった。よって、増殖ルールの正当性が証明された。

## 5. むすび

本論文では、増殖による回路の構成方法として、グラフを用いた回路モデル上で実現し得る任意のトポロジを 3 つのルールで構成できることを確かめた。これにより PCA 上で任意のトポロジを構成する際、どのような順序でどのように接続していくかを定義できる。これは PCA の最大の特徴である増殖による動的再構成機能を実現する上でなくてはならないものである。また、ルールの数はわずか 3 つしかないため PCA の固定部のコマンドとして組み込むには十分に小さく実現でき、ハードウェア量の増大を抑えることができると考えられる。増殖ルールはいくつか考えることができるが、ルール数とルール適用コスト、すなわち操作回数はおおよそトレードオフの関係にあるため、今後は本論文で示したルールに対して適用コストを含めた評価を行い、PCA 上での実現を目指す。

## 文 献

- [1] 平成 13 年度特許出願技術動向調査分析報告書 “プログラマブル・ロジック・デバイス技術,” 特許庁, (2002).
- [2] S.D. Brown, R.J. Francis, J. Rose, and Z.G Vranesic, “Field-Programmable Gate Array,” Kluwer Academic Publishers, (1992).
- [3] S.M. Trimberger, “Field-Programmable Gate Array Technology,” Kluwer Academic Publishers, (1994).
- [4] Masato Motomura, “A Dynamically Reconfigurable Processor Architecture,” Micr oprocessor Forum, Oct. (2002)
- [5] K. Nagami, K. Oguri, T. Shiozawa, H. Ito, and R. Konishi, “Plastic Cell Architecture: Towards Reconfigurable Computing for General-Purpose,” in Proc. FCCM '98, pp.68-77, Apr. (1998).
- [6] 小栗 浩, “布線論理による新しい汎用情報処理アーキテクチャ PCA,” bit, (連載) Vol.32, No.1, pp.27-35, Vol.32, No3, pp.54-62, Vol.32, No.7, pp.51-59, Jan.-Jul. (2000).