計算機アーキテクチャ 131－6
（1998. 9. 21）

適応デバイス RHW の概要とマッピング手法

山内 宗、中谷 正吾、犬尾 武、梶原 信樹

RWCP 適応デバイス NEC 研究室

〒 216-8555 神奈川県川崎市宮前区宮崎 4-1-1
NEC C&C メディア研究所
yamauchi@ccm.cl.nec.co.jp

あらまし

問題のアルゴリムに最適な構造に再構成可能で、高速な処理が可能な次世代 FPGA として、適応デバイスを開発している。従来の FPGA は、コンピューティング応用に適用するには、性能、機能が不十分である。我々は、従来の FPGA に対して、演算性能、アルゴリズムの高速なマッピング機能を強化し、柔軟性と高速性を合わせもつ適応デバイスの開発を進めている。本報告では、我々が開発した、高速な演算処理を特徴とする次世代 FPGA である適応デバイス RHW の概要、適応デバイス RHW 用配置配線ツールの概要と、その中で用いる多点網配線手法、及びテクノロジーマッピング手法について述べる。

キーワード　再構成可能、FPGA、論理合成、自動配線

Overview of Adaptive Device RHW and Its Mapping Methods

Tsukasa YAMAUCHI, Shougo NAKAYA, Takeshi INUO,, Nobuki KAJIHARA

RWCP Adaptive Devices NEC Laboratory

1-1, Muyazaki 4-Chome, Miyamae-Ku, Kawasaki, Kanagawa 216-8555, JAPAN
NEC C&C Media Research Laboratories
yamauchi@ccm.cl.nec.co.jp

Abstract

Conventional microprocessors show poor performance in real world applications,such as image processing, characteristics extraction from huge number of patterns and real time processing. Therfore we decided to develop a novel reconfigurable adaptive device prototype chip that is a new generation FPGA and achieves high performance by mapping target algorithms directly on its reconfigurable architecure. We have developed a prototype chips, RHW to prove feasibility and performance of proposed architecture. In this paper, we describe overview of the prototype chip and its mapping environment.

key words　reconfigurable computing, FPGA, logic synthesis, routing

1. はじめに

問題のアルゴリムに最適な構造に再構成可能で、高速な処理が可能な次世代FPGA (Field Programmable Gate Array) として、再構成可能適応型デバイス (以下、適応デバイス) を開発している。

実世界の応用では、多量のパタン情報を処理するために、膨大なCPUパワーが必要になる。実時間性が必要になる応用も多く、CPUでは対応不可能な場合がある。現在利用可能な再構成可能なハードウェアとしてはFPGAがあるが、従来のFPGAでは、実世界のこのような応用に適用するには、性能、機能が不十分である。そこで我々は、従来のFPGAに対して、演算性能、アルゴリズムの高速なマッピング機能を強化し、柔軟性と高速性を合わせもつ適応デバイスを目標として開発を進めている。

我々が開発した適応デバイスRHW (Reconfigurable Hardware) は、数種類の機能セルが数多く敷き詰められ、チップ内にプログラマブルな配線リソースが予め用意された特殊な構造となっている。従って、適応デバイスRHWに問題をマップする際には専用のテクノロジーマッピングツール、及び配置配線ツールが必要となる。本報告では、我々が開発した、高速な演算処理を特徴とする次世代FPGAである再構成可能適応デバイスRHWの概要、適応デバイスRHW用配置配線ツールの概要と、その中で用いる多点網配線手法、及びテクノロジーマッピング手法について述べる。

2. 適応デバイスRHW

現在の汎用プロセッサはハードウエア構成は固定で、ソフトウェアにより汎用性を実現している。しかし、問題を逐次実行の命令列で処理するため処理速度が遅い。一方、特定の問題に特化して作られた専用プロセッサは、その問題に対しては最高の性能を有するが、多様な問題や頻繁な仕様変更には対応できない。

RHWでは、ハードウエアの構造を問題に応じて最適に変えることによって、汎用プロセッサの柔軟性と専用プロセッサの高速性の実現を目指す。現在、このような目的に供することができるデバイスとしてFPGA[1]がある。しかし、FPGAはもともとASICの代替として論理回路をマッピングすることを主眼としてきたため、多ビット演算を多用する『コンピュータ応用』には冗長で、性能が不十分である。そこで我々は『コンピュータ応用』に適した適応デバイスを開発することにした。

2.1 基本構成

図1 (a) にRHWの全体図を示す。メイン部は、主にデータパス系をマッピングする部分でチップの大半を占める。その周辺に、制御系論理回路をマッピングするのに適した周辺部があり、最も外側にチップ外部と信号を授受するI/O部がある。この構造は次の理由による。適応デバイスにおいて、効率的な処理をするためには、まず、マッピングすべきアルゴリズムのデータフローに可能なかぎり忠実なデータパスを構成し、データパス上には必要な場所に必要な演算器を配置する。こうするとデータフロー制御や演算の種類を逐次指定するための複雑な制御系は不要になる。従って、制御系はデータ数のカウントや、タイミングの調整といった簡単なもので済むため少ない面積でよく、チップの大部分は可能なかぎり並列性を出して高速な処理を実現するためのデータパスに割り当てられる。また、データパスは遅延が少なくなるように他の回路を挿入せずまとめて配置し、制御系はその周辺に配置する。以上のような実装を効率的に行うためには図1(a)のような構成が適している。図1(b)はRHWのより詳細なブロック構造を示したものである。メイン部、および左右の周辺部は9ビットのALUから成り、上下周辺部には種々のビット長のALUが配置されている。

コンピュータ応用では、データパスの大部分は多ビット演算で占められる。すなわちビット毎に異なる複雑な論理演算はほとんど出てこない。このため、メインALUは各ビットを構成する機能セルを結合して多ビット演算に特化し、小面積化を図っている。一方、周辺ALUは、多様で複雑な論理演算を使用する制御系がマッピングされるため、各ビットを独立な機能セルとして使用することができるようになっている。多ビット演算用機能セル (A-Cell) は複雑な論理演算用機能セル (R-Cell) に比べて必要なコンフィギュレーションメモリ量が約1/5で足りるためA-Cellの大きさは非常にコンパクトである。RHWではA-Cellがチップの大部分を占めるので、通常のFPGAのようにチップ全面にフル機能を持つセル (R-Cellに相当) を配置するのに比べて、セル集積度を上げることができる。

2.2 機能セル

図2に、RHWの一つの機能セルで実現できる機能を示す。機能セルはプログラムによって、算術演算に適した算術コンフィギュレーション (a) と論理機能が充実した論理コンフィギュレーション (b) のいずれかに設定できる。図2(a)は全加算器に機能拡張のためのロジックを付加した形になっており、これによって使用頻度の高い加減算器や乗算器の構成要素等を効率的に作ることができる。図2(b)は論理コンフィ

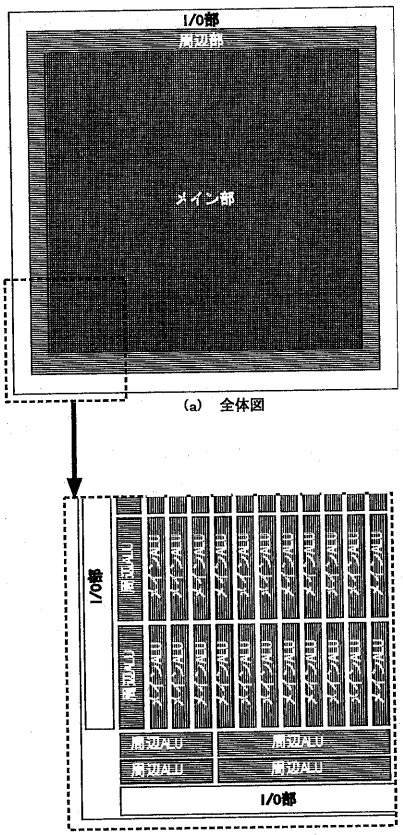

図 1: RHW 全体構成図

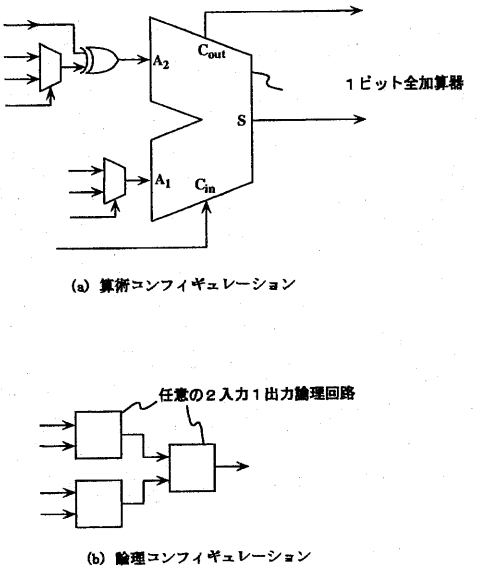

図 2: 1 つのセルで実現可能な機能

ギュレーション時の機能のうちもっとも有用な表現を示したもので、これは複雑なランダムロジックを効率的に構成するのに有用である。また、後述するように、この表現を前提とする高速なテクノロジ・マッピング手法が存在し、アルゴリズムの高速なマッピングにも有利である。

2.3 プロトタイプチップ RHW1

RHW プロトタイプチップとして、0.35 μ セルベース IC にて RHW1 を開発した (図 3)。RHW1 は 6 × 20 個のメイン ALU を内蔵する。RHW1 は、機能検証を主目的としたため、集積度を追及していないが、配置配線の結果からこの二倍以上の集積化の見通しを得ている。テスト用に 18 ビット加算器を多段に接続した回路をマッピングしたところ、126MHz までの動作を確認した。RHW1 は全コンフィギュレーションに 13K バイトのデータを要するが、32 ビット並列でデータ転送し、システムクロックの速度でコンフィギュレーションが可能なため、非常に高速にコンフィギュレーションすることが出来、約 34 μ 秒 (126MHz 動作) で全書き換えが終了する。同等の回路規模の FPGA (Xilinx XC4013XL) の約 3.3m 秒に比べて 100 倍程度高速である。これは、ビデオレートで送られてくる各画像に対し、コンフィギュレーションを書き換えながら複数の処理を施すような応用にも対応可能で、既存 FPGA では達成困難な速度である。

適応デバイスを用いた応用として、高次自己相関アルゴリズムを用いた画像認識システムを開発中である。コアのアルゴリズムを RHW1 へ実装するために、高次相関アルゴリズムの回路を設計し、VHDL で動作の検証を終えた。この回路は、演算器の数、配置、データフローを最適化して 35 個の特徴ベクトル成分を並列に計算できるようにしたもので、各画素を 1 クロックで処理できる。これは、RHW1 を 66MHz で動作させた場合、DEC ALPHA (400MHz) の 34 倍の処理速度となる。

3. 適応デバイス用マッピング手法

静的適応デバイス RHW1 は、柔軟な構造を持つプログラマブルなハードウェア上に問題をダイレクトにマッピングすることにより、高速に処理を行なうことを目的としている。従って、RHW1 の処理能力は、問題をマッピングする手法に大きく依存する。また、

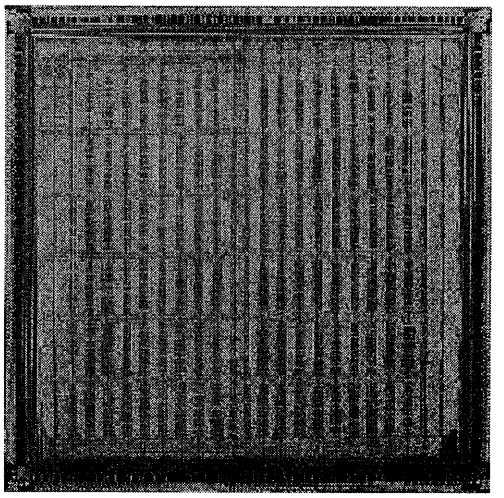

図 3: RHW1 チップ写真 ($14.9 \times 14.9mm^2$)

RHW1 をコンピュータとして使うことが主目的であるので、LSI を設計する時の様に何時間もマッピングのツールが走る必要があるという状況は好ましくない。我々は、プログラム入力からコンフィギュレーション・データ作成まで 5 分以内を目標として開発をしている。

マッピングの流れを図 4に示す。コンパイラは、高級言語で記述されたプログラムを RTL レベルの記述 (VHDL 記述) に変換するもので、CAD の分野では機能合成と呼ばれている。機能合成のツールは、ハードウェア・リソースの制約に従って、データパスを生成し、演算器やレジスタ等のスケジューリングや割り当てを決定するものであり、我々は NEC で研究開発された Cyber[2] を用いることにした。論理合成は、RTL 記述からチップに合わせたゲートレベルの記述を出力するが、高速に論理合成を完了するための新たなテクノロジーマッピング手法を考案した。また、適応デバイス RHW1 は、予めチップ上に用意されたセルや配線リソースを用いて回路をマップするので、そのための専用の配置配線ツールを開発する必要がある。

本節では、RHW1 用自動マッピングツールを構成する、新たに考案したテクノロジーマッピング手法、自動配置配線ツールとその中で用いる多点網配線手法について述べる。

3.1 RHW1 向きテクノロジーマッピング手法

RHW1 は、問題を直接プログラマブルなハードウェア上にマップして高速実行することを目的としている。そのためには、機能合成等の上位のツールが生成したハードウェア記述を、RHW1 のセルに適合する様にテクノロジーマッピングする必要がある。また、マッピングに要する時間は、従来型計算機におけるコンパイル時間に相当するので、数分間程度に抑える必要がある。

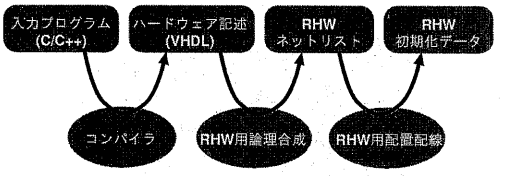

図 4: RHW1 におけるマッピングの流れ

早稲田大学と共同で、RHW1 の機能セル構造の特徴を生かした高速テクノロジーマッピング手法を開発した。適応デバイス RHW1 の機能セル (図 2) は、高速演算器を構成するのに適した構造となっていると同時に、コンフィギュレーションを変更することで、任意の 2 入力 2 段の論理関数が実現可能である。この特徴を生かすと、ノードの論理的な意味を考えず、トポロジーのみに着目したマッピングが可能となり、処理時間を大幅に短縮出来る [3]。具体的に説明すると、一つの論理素子で表すことが出来る部分グラフは 3 通りあるので、これらを用いて予め 2 入力に分解された組合せ回路 (ネットリスト) の入力側から面積最小になる様にカバーして行く。この手法を評価したところ、Look-Up Table ベースの FPGA を対象とした従来の手法 (UCB の MIS-pga 等) と比較して論理合成に要する処理時間が格段に高速化 (40 ～ 60 倍) された。また、論理段数 / 面積のトレードオフ (最適解からの許容幅を指定) を考慮したマッピング手法も考案 / 評価を行なった結果、1 ～ 2 割程度論理段数 (即ち遅延) が改善された。

厳密には、RHW1 のセルは任意の 2 入力 2 段の論理関数よりも多機能な表現能力を有している。そこで、RHW1 のセルを任意の 2 入力 2 段の論理関数とみなす方法 (Tree(1)) と、RHW1 のセルの全表現能力を用いる方法 (Tree(1,2,3)) の双方を比較した評価結果を図 5に示す。これを見ると、Tree(1,2,3) の方が多少セル数を削減出来るが、処理速度が数十倍も遅くなるので、総マッピング時間 5 分以内を目標とした我々の要求には Tree(1) の方が適していると考えられる。但し、予め RHW1 用のコンフィギュレーション・データを用意することが可能で、少しでもセル数を圧縮したい場合を考え、Tree(1,2,3) もコンパイルオプションで選択可能にしておく。

Circuit	#block		Time (sec.)	
	Tree(1)	Tree(1,2,3)	Tree(1)	Tree(1,2,3)
C1355	172	171	0.947	17.84
C17	5	5	0.012	0.026
C1908	265	263	2.262	44.39
C2670	536	531	8.932	220.3
C3540	541	522	10.924	446.9
C432	98	98	1.093	25.24
C499	90	90	0.499	6.886
C5315	930	914	13.24	476.6
C6288	856	803	15.795	1461.5
C7552	1185	1162	19.664	670.3
C880	168	165	1.676	40.90
apex6	313	308	3.009	46.14
b1	5	5	0.012	0.046
cc	35	35	0.114	0.481
cm138a	15	15	0.020	0.080
cm151a	23	21	0.130	1.630

Circuit	#block		Time (sec.)	
	Tree(1)	Tree(1,2,3)	Tree(1)	Tree(1,2,3)
cm152a	9	9	0.123	0.738
cm42a	17	17	0.016	0.058
cm82a	6	6	0.014	0.082
cm85a	16	16	0.133	0.899
cu	30	29	0.126	1.007
dalu	686	674	8.567	347.1
i5	199	194	0.804	10.65
i6	246	246	1.962	10.85
majority	5	5	0.025	0.150
parity	5	5	0.193	0.872
rot	311	302	1.733	53.09
vda	448	439	9.349	124.5
x1	186	182	1.723	25.39
x2	26	26	0.102	0.924
x3	461	438	4.346	84.91
x4	237	235	1.342	15.09
total	8125	7931	108.9	4135.6

図 5: テクノロジーマッピング評価結果

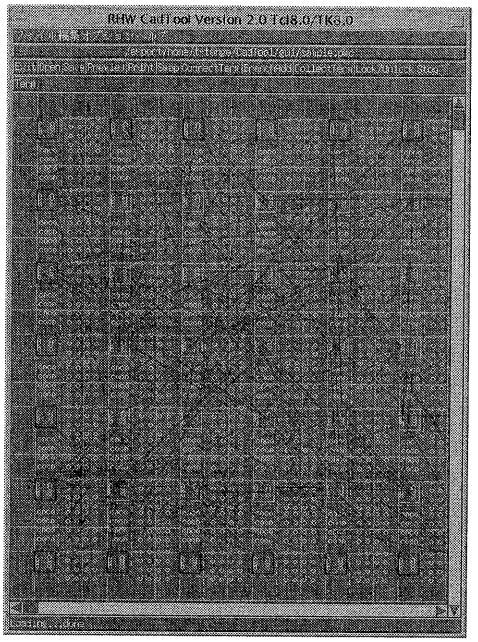

図 6: RHW1 用配置・配線ツール

3.2 RHW1 用自動配置配線ツール

RHW1 は、機能セルが数多く敷き詰められた構造となっている。従って、配置処理は、ゲートアレイ等の LSI とは異なり、離散的な座標に存在する機能セルリソースの中から適切な場所を選ぶ処理になる。自動配置処理については、入力ネットリストに対してグラフ分割手法を適用し、2 分割された部分ネットリストを RHW1 の左右に配置し、更にその部分ネットリストを 2 分割して上下に配置するという処理を、全セルが配置されるまで再帰的に繰り返す手法を用いた ([6][7][8])。

配線処理に関しても、RHW1 はゲートアレイ等の LSI とは異なり、チップ内に予めプログラマブルな配線リソースを有しているので、それらの配線リソース上で配線径路を求めるという処理になる。自動配線処理については、RHW1 の配線リソースを、重み付き有向グラフで表現し、そのグラフ上で、必要な端子間の最短径路を Dijkstra の手法を用いて探索する手法と次に説明する多点網配線手法 (MPFA 法) を用いた。

また、(1) 手動配置、(2) 配置結果の確認、(3) 配置結果の修正、(4) 手動配線、(5) 配線結果の確認、(6) 配線結果や未配の修正、(7) 性能改善等の目的のために、GUI を用いた配置 / 配線処理と修正処理の機能も付加した。このツールが動いている際の実際の画面表示を図 6に示す。

3.3 多点網配線手法

適応デバイス RHW1 用に、品質の高い配線径路を求める新たな多点網配線手法を考案した (MPFA 法)。1 対 1 の配線であれば、Dijkstra の手法を用いて最短径路を求めるのが良い。しかし、通常の論理回路では、ひとつの信号源に対して、その信号を受けとる端子が複数存在する多点網を構成していることが多い。

多点網の配線径路を求める手法としては、(1) 端子対を順番に選び出してその端子対間の最短径路を求めることにより、最終的に網全体の配線径路を求める方法、(2) 初期解として端子間の最短径路長を枝の重みとした端子だけのグラフを作り、次にそのグラフの最小全域木を求め、そのグラフの枝を元の最短径路に戻してから再び最小全域木を求めた後、余分な枝葉を取り除く方法 (KMB 法)、(3) 信号の流れる向きに着目して、信号源端子から目的端子までの距離を短くすることに注力した Path-Folding Arborescence (PFA) 法、等の手法が提案されている [4]。

しかし、(1) の手法は、個々の端子対間の距離 (コスト) の最小化のみに着目しており、多点網全体のコスト削減になるとは限らず、また端子対を選ぶ順番にもコストが大きく依存するという問題点がある。また、(2) の手法 (KMB 法) は、信号の流れる向きを考慮せずに総配線長のみを最小化してしまうため、実際の回路における信号の伝搬遅延削減に結び付かないという問題がある。(3) の手法 (PFA 法) は、端子やグラフのノードを同時に 2 つずつしか考慮しないため、配線径路のパターンによっては、無駄な配

線径路を生成するという問題点がある。

そこで、新たな多点網配線手法を考案した。まず、信号源から各々の端子への最短径路を求め、それらの最短径路の重なり具合を、枝の重みと重なっている最短径路の本数を用いて評価し、重なりの大きい複数の最短径路を共通化していくことにより、総配線長が短く且つ、信号源から端子への距離が最短になっている配線径路を求める手法である (Multiple Path-Folding Arborescence (MPFA) 法と呼ぶ。この手法を用いることにより、PFA 法よりも伝搬遅延や配線の静電容量が少なく、高速動作に適した多点網配線径路を求めることが出来る。

本節では、従来の多点網配線手法と考案した MPFA 法の詳細について述べる。

3.4 従来の多点網配線手法

- KMB 手法

  1. 端子だけのグラフ A を作る (端子間のエッジは最短径路長のコストを持つ)
  2. グラフ A の MST(最小全域木) を求める (グラフ B と呼ぶ)
  3. グラフ B の枝を実際の最短径路に戻す (グラフ C と呼ぶ)
  4. グラフ C の MST を求める (グラフ D と呼ぶ)
  5. グラフ D の中の余分な枝を取り除く (葉が端子ではないものを取り除く)

- PFA (Path-Folding Arborescence) 手法

  信号のソースとなる端子を n_0 とする。
  $G=(V,E)$, $\{n_0,p,s\} \subseteq V$
  $minpath_G(n_0,p)$
  $= minpath_G(n_0,s) + minpath_G(s,p)$
  の時、p が s を支配 (dominate) すると言う。言い替えると、n_0 から p までの最短径路として s を通るものが存在すれば、p は s を支配する。また、MaxDom(p, q) は、ノード p, q に支配され且つ $minpath_G(n_0, MaxDom(p,q))$ が最大となるノードを指すこととする。

  1. $m = MaxDom(p,q)$
  2. p,q を m と置き換える
  3. n_0 だけになるまで、ステップ 1、2 を繰り返す
  4. $MaxDom(p,q)$ から p,q の最短径路を求めて最終的な径路を求める

3.5 MPFA 手法

PFA 手法がノードを 2 個ずつまとめていくことによって生じる問題点を解決したのが、我々の考案した MPFA (Multi-Path-Folding Arborescence) 手法である。

MPFA 手法の説明をする前に、各記号について次の通り定義する。

グラフを G、ノードのセットを V、枝のセットを E、端子のセットを N、信号源端子を n_0、目的端子のセットを T で表す。
$G=(V,E)$, $N \subseteq V$, $N = \{n_0\} \cup T$

1. 信号源端子から、目的端子への最短径路を列挙 (最短径路セット M)

   $$M = \{minpath_G(n_0,t) | t \in T\}$$

   この場合、$minpath_G(n_0,t)$ は、信号源端子 n_0 から目的端子 t までの最短径路を構成する枝の集合を表す。

   $$minpath_G(n_0,t) \subseteq E$$

2. 列挙した最短径路集合の中で最短径路が重複している部分の重複コストを計算

   $$C = \{cost_m(G,M,n_0,T) | m \in M\}$$

   以下に、$cost_m$ の計算について示す。

   (a) 異なった目的端子への最短径路が 2 本以上重なっている枝を探す
   $E = \{e \in m \wedge e \in m' | m = minpath_G(n_0,t), m' = minpath_G(n_0,t'), t \neq t', \{t,t'\} \subseteq T\}$

   (b) その枝の重みと (重なっている本数 -1) の積を、その枝の重複度 (overlap score) とする
   $overlap\_score(e)$
   $= (overlap\_num(e) - 1) \times weight(e)$

3. 信号源端子から枝を辿って、重複度 0 の枝に到達するまでの各枝の重複度の総和 (これを重複コストと呼ぶ) が最も大きくなる径路を探す。

   $$overlap\_cost(m) = \sum_{e \in m} overlap\_score(e)$$

4. その径路の端点 (重複度 0 の枝との境界) となる頂点 p を求める

5. 信号源端子からの最短径路が頂点 p を通ることが可能な目的端子集合 U を求める

6. 頂点 p から、集合 U の端子への最短径路を求める (求める配線径路の部分解)

7. 集合 U の端子を削除し、頂点 p を代わりの端子とみなす

8. 信号源端子と、新たに端子とみなすことになった頂点 p 以外に端子がなければ、信号源端子と残った端子 (頂点 p) の間の最短径路を求めて、多点網配線処理が終了する。信号源端子以外に 2 個以上の端子が残っていた場合は、ステップ 1 に戻る

図 7〜図 10に、 MPFA 手法を用いて多点網配線径路を求めている様子を示す。

同じ問題に対して PFA 手法と MPFA 手法を適用した際の違いについて図 11に示す。 PFA 手法はノードを 2 個ずつまとめていくので、配線結果が 2 分木の形になり易いが、我々が考案した MPFA 手法は同時に複数の配線径路を束ねることが可能なので、よりコンパクトな配線結果が得らる。従って、 PFA 手法で生じる問題点が MPFA 手法においては解決されていると言える。

また、実装を工夫することにより、各ノードのラベリングを一度するだけで、後はインクリメンタルな更新のみで全ノード間の配線径路を求めることが出来るので、高速に配線処理を行なうことが出来、短時間にマッピングを終了するという RHW1 の目的にも適した手法と言える。

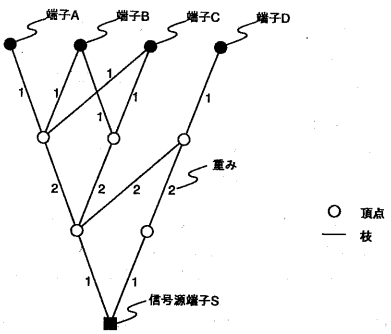

図 7: MPFA による配線処理 (1)

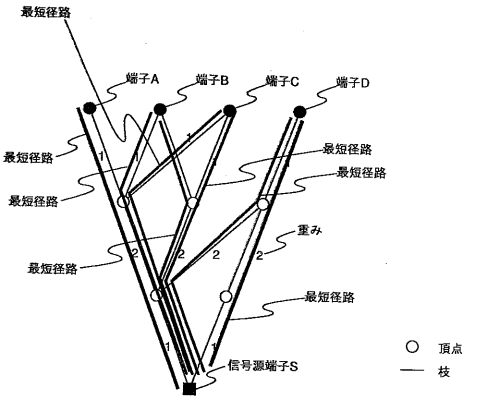

図 8: MPFA による配線処理 (2)

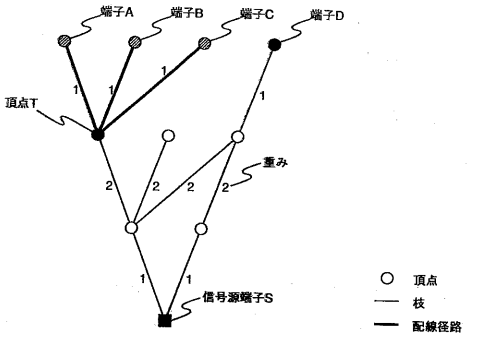

図 9: MPFA による配線処理 (3)

3.6 MPFA 手法の評価結果

一様分布の乱数を用いてランダムに重みを生成した 2 次元メッシュ上に、同じく個数や配置をランダムに決めた多端子間の配線を MPFA 手法と KMB 手法で比較評価した結果を表 1に示す。これを見ると、 KMB 法と比較して、総配線長が長いが、これは、 MPFA 手法が信号源から各端子への配線が最短になること

を保ちつつ多点網を生成するため、乱数で信号源端

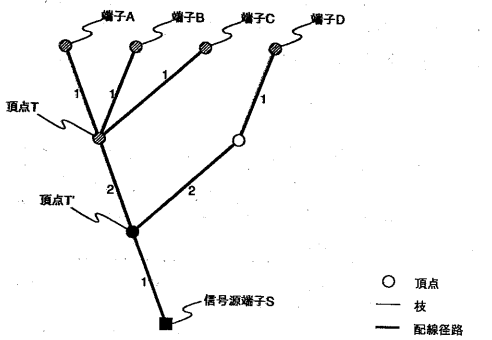

図 10: MPFA による配線処理 (4)

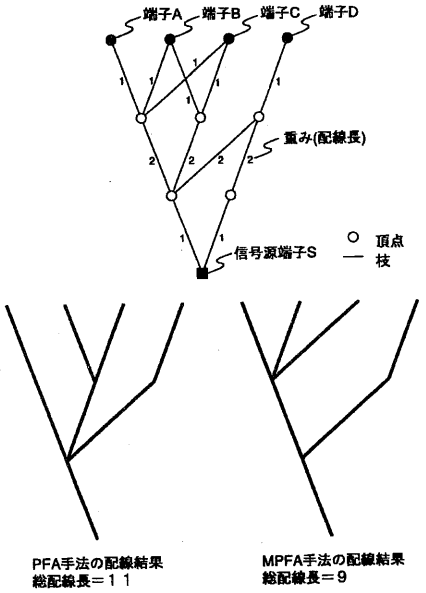

図 11: 従来の PFA 手法と提案した MPFA 手法の比較

子を決めると不利な配置になることが多いためである。処理時間は MPFA 手法の方が 10 ～ 20 倍ほど高速である。これは、MPFA 手法が重みの更新をインクリメンタルに行なえるという実装上の利点と、KMB 手法が用いている最小全域木を求める処理が重いことが要因である。

適応デバイス RHW のマッピングツールは高速性が重要であり、その点では、MPFA 手法が有利である。総配線長に関しては、MPFA 手法は信号源端子の位置の影響を大きく受けるので、配置ツールの結果が重要であり、配置配線、そして論理合成のツールまで含めた統合的な連携動作が高性能化のために必要であると考えられる。

表 1: MPFA 手法と KMB 手法の評価結果

	MPFA 手法		KMB手法	
メッシュサイズ	平均処理時間	平均総配線長	平均処理時間	平均総配線長
100x100	0.128 sec	2435	2.932 sec	1686.5
200x200	0.75 sec	7012.6	17.173 sec	4569.7
500x500	5.672 sec	13405.7	72.971 sec	8333.9

UltraSPARC2(300MHz)使用
重みは1～20の乱数

4. まとめと今後の課題

適応デバイス RHW 及びマッピングツールの開発状況に関して報告した。適応デバイス RHW1 は、126MHz のクロックで動作することを確認した。現在、RHW1 の改良版として、3 倍以上の高集積化を目指した RHW2 を開発中である。

配置・配線ツールに関しては、GUI を用いた手動配置・配線と基本的な自動配置・配線機能が完成し、新たに考案した MPFA 法も実装されているので、今後は、より一層多点網配線能力を充実すると同時に、自動配線の配線率、配線品質の改善を狙う。

論理合成については、本報告で述べたテクノロジーマッピング手法を他の機能 / 論理合成ツールと連携動作させ、C 言語記述からゲートレベルのネットリストの作成までを自動化する方向で完成させる。

また、これ以外のツールとしては、生成されたコンフィギュレーション・データが RHW1 チップに電気的なダメージを与えないかどうかをチェックするツールや、RHW1 チップにロードしないで機能検証をするためのシミュレータ等の開発も必要と考えられる。

参考文献

[1] http://www.xilinx.com/xilinx.htm

[2] Wakabayashi, K. and Tanaka, H., 'Global Scheduling Independent of Control Dependencies Based on Condition Vectors', 29th DAC, pp112-pp115, (1992)

[3] 荒 宏視、戸川 望、柳澤 政生、大附 辰男、"ツリー構造を持つ論理ブロックを対象としたテクノロジーマッピング手法"、信学技報、VLD97-104(1997)

[4] Alexander, M.J. and Robins, G., 'New Performance-Driven FPGA Routing Algorithms', IEEE Trans. Computer-Aided Design, Vol.15 No.12, pp.1505-1517, Dec. (1996)

[5] Cong, J. and Madden, P.H., 'Performance-Driven Routing with Multiple Sources', IEEE Trans. Computer-Aided Design, Vol.16 No.4, pp.410-419, April (1997)

[6] Kernighan, B.W. and Lin, S., 'An Efficient Heuristic Procedure for Partitioning Graphs','The Bell System Technical Journal, 49, 2 , pp.291-307, Feb. (1970).

[7] Fiduccia, C.M. and Mattheyses, R.M., 'A Linear-Time Heuristic for Improving Network Partitions', Proc. of 19th DAC, pp.175-181, (1982)

[8] Edahiro, M. and Yoshimura, T., 'New Placement and Global Routing Algorithms', Proc. of 27th DAC,pp.642-645, (1990)