

多量の配線要素を持つ通信処理向け FPGA の専用 CAD システム

室岡 孝宏 高原 厚 筒井 章博 宮崎 敏明
NTT 光ネットワークシステム研究所
〒243-01 神奈川県厚木市森の里若宮 3-1
電話：0462-40-2470
電子メール：murooka@aecl.ntt.co.jp

あらまし 我々は、高速性と柔軟性が要求される通信処理分野をターゲットとした FPGA と、そのアーキテクチャに着目した専用 CAD システムを開発した。本報告では、専用 CAD システムの構成とその処理アルゴリズムについて述べる。本システムは論理合成後のネットリストを入力とし、テクノロジーマッピング、配置配線、プログラムデータの生成という一連の設計環境を提供している。特に、信号の流れに着目した配置処理および、配線構造を意識した高速な配線処理が特徴であり、効率的な回路設計が可能である。

キーワード FPGA、通信処理、CAD、テクノロジーマッピング、配置配線

A dedicated CAD system for telecom-FPGA having rich routing resources

Takahiro Murooka Atsushi Takahara Akihiro Tsutsui
Toshiaki Miyazaki
NTT Optical Network Systems Laboratories
3-1 Morinosato Wakamiya, Atsugi-shi,
Kanagawa Pref., 243-01, JAPAN.
Phone : +81-462-40-2470
E-mail : murooka@aecl.ntt.co.jp

Abstract This paper describes the dedicated CAD system for newly developed telecommunication-oriented FPGA. This CAD system supports the total design environment which is composed of technology-mapping, placement, routing and verification-tools. In the placement and routing algorithms, the properties of our target FPGA structure are well considered to achieve the efficient implementation of given circuits.

key words FPGA,telecommunication,CAD,technology-mapping,placement,routing

1 はじめに

現在、Field Programmable Gate Array (FPGA) [Bro92] が様々な分野で用いられている。通信処理においても様々なプロトコルへの対応を柔軟に行なえるシステムとしての利用が考えられている。我々は、プログラマブル通信装置の構成要素として、通信処理用 FPGA PROTEUS [ONY+94] と専用 CAD システム [筒井95] を開発し、実際の通信処理装置に適用し評価を行なった [林95]。その結果、PROTEUS はトップダウン設計では十分な性能を引き出せないことが明らかになった。そこで、PROTEUS を拡張した新たな通信処理用 FPGA である PROTEUS-Lite [MTK+97] を開発した。PROTEUS-Lite は配線資源を豊富にすることにより、自動設計向けの構造になっている。

本稿では、この様に配線要素を多量に持つ FPGA PROTEUS-Lite の専用 CAD システム (PLCAD) について、そのシステム概要、および処理アルゴリズムについて報告する。PLCAD は論理合成ツールとのインタフェース、テクノロジーマッピング、配置配線、プログラムデータの生成に至る一貫した自動設計/検証環境を提供する。

PROTEUS-Lite の配線資源は PROTEUS と比較して水平/垂直チャンネルの配線トラック数、およびそれらを接続するためのスイッチ数を約2倍に拡張した。そのため、配線処理において膨大な配線資源の組合せから効率的に経路を探索する手法が必要となる。配線処理の高速化手法としては、S.Thaker らによるスイッチモジュールの構成に着目した整数線形計画法による手法 [TWM94] がある。また、最短経路探索の手法としては M.J.Alexander らのスタイナー木を用いた IDOM [AR95] がある。しかし、両者とも XILINX 型の方向性を有しない FPGA を対象としたものであり、特定方向の信号の流れを考慮した階層的な配線構造は考慮されておらず、PROTEUS-Lite に適用するのは難しい。我々は、PLCAD の配線処理に配線資源の構造を強く意識した手法を導入することにより、配線処理の高速化を実現した。

配置処理においては、搭載する回路の信号の流れに着目し、そのトポロジーから初期配置を決定することにより、回路の動作速度の向上を計った。また、高速動作実現のために用いられるハードマクロを含んだ回路を効率的に扱う手法を備えている。

以下では、2章で PROTEUS-Lite の構造的特徴を述べた後、3章で PLCAD の処理の流れと、処理アルゴリズムについて述べる。4章で通信処理回路をもちいて本 CAD システムを評価した結果を示す。5章にまとめを示す。

2 PROTEUS-Lite の概要

2.1 全体構成

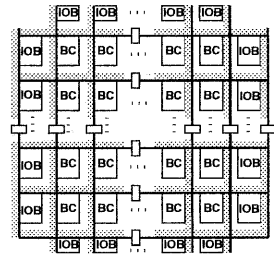


図 1: PROTEUS-Lite の全体構成

PROTEUS-Lite の全体構成を図 1 に示す。格子状の対称的配置 [ATT93] の SRAM 型 FPGA であり、格子状に配線チャンネルを有している。配線スイッチ類はチャンネルの格子点上に配置されている。論理ブロックである基本セル (BC) は各格子に配置されており、外部との接続の I/O (IOB) も縁辺部の格子に配置されている。

図 2 に PROTEUS-Lite のチップ写真と諸元を示す。

諸元	
タイプ	SRAM type
プロセス	0.5 μ CMOS (3 metals)
サイズ	17.1mm \times 15.3mm
I/O 数	136 (双方向)
BC 数	280(28 \times 10)
LUT 数	1120(4LUT/BC)
FF 数	1672
メモリ	223K bit
パッケージ	QFP/CSP/PGA

図 2: PROTEUS-Lite LSI

2.2 基本セルの構成

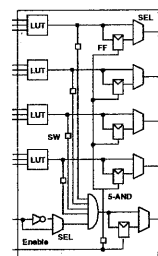


図 3: 基本セルの構成

基本セル (BC: Basic Cell) の構成を図 3 に示す。BC は 4 個の 3 入力 1 出力の Look-Up Table (LUT)、5 入力 AND ゲート (5-AND)、5 つのフリップフロップ (FF) で構成される。FF の使用/不使用はプログラムにより設定可能である。

2.3 階層的配線構造

PROTEUS-Liteの配線資源は二階層で構成されている。第一の階層はBCの内部の論理実現要素と、ローカルラインと呼ばれる隣接する論理要素間を接続する短い配線から構成される。第二の階層はBCの一辺と同じ長さのミドルラインと呼ばれる配線と、チップ一辺の半分の長さを有するロングラインで構成される。それらは乗り換えスイッチ、切り離しスイッチを介して接続できる。この階層では自由度の高い豊富な配線資源を提供している。表1にBC単位の配線要素数を示す。

表 1: BC 単位の配線要素数

スイッチ類	配線要素				
	ロング		ミドル		
	水平	垂直	水平	垂直	
425	6	4	32	16	5

2.3.1 第一階層

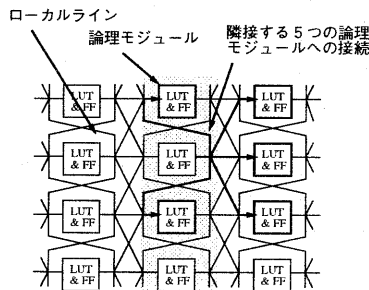


図 4: 第一階層の構成

図4に第一階層の構成を示す。ローカルラインは隣接 LUT を乗り換えスイッチ1個で接続する。1つの LUT の出力は上、下、右上、右下、右に隣接する5つの LUT の入力端子に接続できる。第二階層の配線構成と比較して、ローカルラインで接続される LUT 間の信号伝搬遅延は小さい。ローカルラインを用いることで、通信処理回路に多様されるカウンタ、シフタを容易に実現することができる。

2.3.2 第二階層

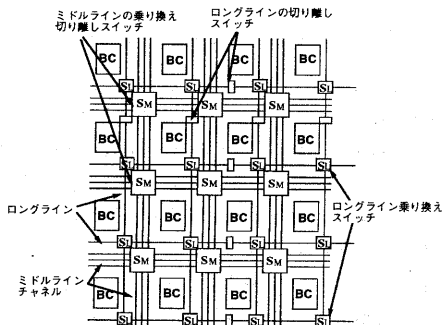


図 5: 第二階層の構造

第二階層の構造を図5に示す。この階層はミドルラインとロングラインで構成される。ミドルラインは垂直、水平

方向に配置され、スイッチモジュール (SM) 内の切り離しスイッチで分割される。ここでは、水平、垂直方向のミドルラインの集合をミドルラインチャンネル、チャンネル内の各ミドルラインをミドルライントラックと呼ぶ。水平方向のチャンネルでは上方から、垂直方向のチャンネルでは左方から順番に整数のトラック番号が付けられている。SM は配線の乗り換えスイッチと切り離しスイッチを含んでいる。ロングラインはチップ中央で切り離しスイッチにより分割されており、複数のミドルラインを切り離しスイッチで接続するよりロングラインを使用する方が伝搬遅延的に有利になる距離に配置された BC 間の接続に使用される。

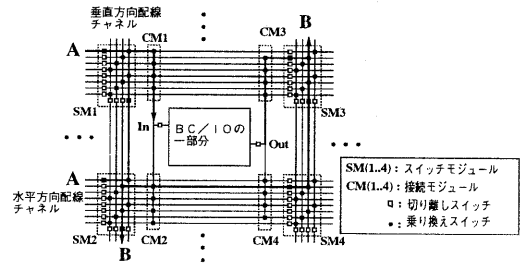


図 6: BC 周辺の配線構造

図6にBC周辺のミドルライントラック、乗り換えスイッチ、切り離しスイッチの構成を簡略化して示す。BC、IOB の入出力配線は上下の水平方向ミドルラインチャンネルと接続モジュール (CM1、2、3、4) を介して接続される。各 CM 中の乗り換えスイッチは BC への入出力が異なる番号のミドルライントラックに配置されている。たとえば、CM1 では奇数番のトラックへの乗り換えスイッチが配置され、CM2 ではそれらが偶数番のトラックに配置される。一方、水平、垂直方向のミドルラインを接続するスイッチモジュール (SM1、2、3、4) では全て同じパターンで乗り換えスイッチが配置されている。この対称構造は配線の接続性の確認処理を容易にし、配線の経路探索処理を高速にする。例えば、上下の水平チャンネルのトラック A が選ばれた場合、チップ上のどの BC 間でも CM1 を介して接続できる。また、垂直チャンネルのトラック B が選ばれた場合、CM3 を介する事でチップ上の全ての BC の出力と接続できる。

乗り換えスイッチは単一の N-MOS トランジスタで構成されるパッシブ素子である。そのため、配線遅延を増加させるので、信号伝搬遅延の少ない配線経路を得るには乗り換えスイッチを少なく使用した経路を探索する必要がある。また、SM には乗り換えスイッチの他に切り離しスイッチが配置されている。切り離しスイッチは同一方向の隣接するミドルライントラックの接続/切り離しの制御を行なう。また、切り離しスイッチは信号を増幅するバッファを有している。

3 PLCAD

3.1 PLCAD 概要

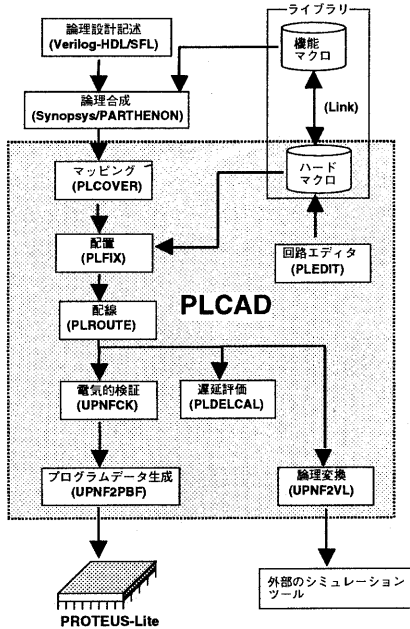


図 7: PLCAD の処理の流れ

PROTEUS-Lite の専用 CAD システムである PLCAD の処理の流れを図 7 に示す。PLCAD は既存の論理合成ツールにより生成されたネットリストを入力とし、テクノロジマッピング (PLCOVER)、配置 (PLFIX)、配線 (PLROUTE) 処理を行なうことにより、PROTEUS-Lite に対応したネットリストを出力する。また、手設計が可能で視覚的な回路エディタ (PLEDIT) も提供している。設計された論理回路は Verilog-HDL に変換 (UPNF2VL) し、シミュレーションを行なえる。配線処理が終了したネットリストは電気的な正当性の検証 (UPNFCK) の後、FPGA へロードするプログラムデータへ変換 (UPNF2PBF) される。配置、配線が終了したネットリストを、ハードマクロとして保存し、再利用できる。

3.2 マッピング処理

マッピング処理の入力ネットリストは二種類の形態に対応している。一つは、2 入力 AND、OR 等と FF で記述されたネットリスト、もう一つは、3 入力 1 出力の LUT で表現できる 256 種類の論理部品と FF で記述されたネットリストである。いずれも単相同期回路として設計されたものでなくてはならない。前者のネットリストでは、PLCOVER は論理をノード、配線をエッジとした有向グラフに展開し、3 入力のノードとなる様に被覆するマッピング処理 [Bro92] を行なう。この時、インバータ、固定値については、その直近の LUT の論理に含める FPGA 固有の処理をする。後者のネットリストは既に PROTEUS-Lite の LUT

の論理形態であるため、被覆処理は行なわず、上記の FPGA 固有の処理を行なう。

入力ネットリストでは、論理合成時に使用されたハードマクロは端子のみの実態の無いモジュールとして扱われる。PLCOVER ではハードマクロと他の回路の接続を検証する処理のみを行なう。

3.3 配置処理

配置処理は配線の局所集中を避け、配線長が最小となるように Simulated Annealing 法を用いて決定する。ただし、PROTEUS-Lite は配線資源の構造が信号の流れの方向を考慮したものとなっているため、初期配置として、ランダム配置を与えたのでは、構造を生かした効率的な配置とはならない。そこで、信号の流れの方向を考慮するため、PLFIX ではチップ左側を入力、右側を出力として論理回路をレベルソートして初期配置を決定する。その後、ネットの始点終点の距離の総和をコストの評価値とする Simulated Annealing を行ない配置を最適化する。また、LUT の接続関係から PROTEUS-Lite が持つ階層的な配線構造を利用して直結可能なものはローカルラインで接続可能な LUT に配置する様に制約条件を設けている。

ハードマクロは物理的な大きさが LUT / BC と異なるために、Simulated Annealing での配置位置の入れ替え処理で LUT / BC と同一に取り扱う事が難しい。そこで、マクロの配置については、チップの領域を配置するマクロの大きさに応じた格子上の領域に分割し、その格子上のみを移動させることにしている。

3.4 配線処理

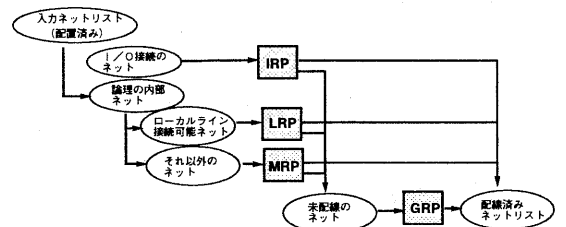


図 8: PLROUTE の配線処理の流れ

図 8 に配線処理の流れを示す。配線処理は I/O 端子優先配線処理 (IRP)、ローカルライン配線処理 (LRP)、ミドルライン配線処理 (MRP)、未配線処理 (GRP) の 4 つの処理からなる。入力ネットリストの配線部分は、I/O 接続ネット、論理の内部ネットに分割する。最初に I/O 接続ネットを IRP で処理する。次に論理の内部ネットからローカルライン接続可能ネットを抽出し、LRP を行ない、IRP、LRP の引き残しを含めた残りのネットに対して MRP を行なう。ここまでの配線処理で発生した未配線は GRP での経路探索を行なう。

3.4.1 I/O 端子優先配線処理 (IRP)

IRPでは、I/O と内部論理を接続するネットを優先的に配線処理する。これは、I/O との配線は外部の論理回路のタイミング的な制約を強く受けるので、配線の不要な引き回しによる遅延の増加を避けるためである。配線処理の対象には遅延を最小にするために、全ての階層の配線資源を対象として経路探索を行なう。ここでの経路探索手法はダイクストラ法を基本としている。効率的な配線資源の使用と、経路探索のための計算量の削減のために経路探索領域を制限し、冗長な配線資源を探索空間から除くことにより処理速度を向上させた。

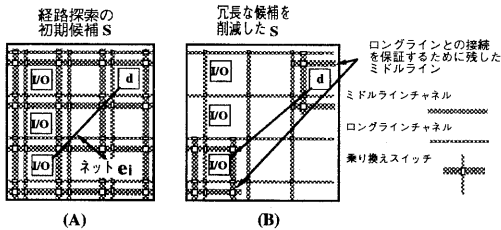


図9: IRPでの配線候補の削減

図9に処理例を示す。ネット(e_i)の始点と終点を囲む領域を S とする。 S の一辺がミドルラインを切り離しスイッチを使用して直列に接続した場合より、ロングラインを使用する方が信号伝搬遅延が小さい場合、 e_i の始点と終点でBCへの接続に必要なミドルラインを残して、それ以外のミドルラインを配線候補から除く。図9は2辺共にロングラインを使用した方が有利になる例を示している。この処理により、ダイクストラ法の経路探索に必要な計算量を少なく押える事ができると共に、ミドルラインとロングラインを効率的に組合せた経路を得ることができる。

通常のネットはファンアウトがあるので、分岐点の決定処理が必要である。図10にIRPでの分岐処理を示す。分岐点の集中を避けるため、スイッチモジュールの構成を基に、一つの配線要素から分岐可能な分岐数の上限を定めた。

1. 与えられたネットをファンアウト先別に分割し、ファンアウトの集合を作成する。
2. ファンアウトの集合をマンハッタン距離の降順にソートする。
3. ファンアウトの集合から、最長のファンアウト f_1 を取り出す。
4. f_1 についてIRPの配線処理で経路を決定し、経路情報に加える。
5. 経路が決定した f_1 を集合から削除する。
(ここで、次に長いファンアウトが f_1 になる。) ファンアウトの集合が空であるなら、処理終了。
6. 集合から f_1 を取り出す。
7. これまでに決定した経路の中で f_1 の入力側に最も近く、且つ、分岐制限を越えない配線要素へ f_1 の出力側を移動する。
4の処理へ戻る。

図10: ファンアウトの分岐処理

3.4.2 ローカルライン配線処理 (LRP)

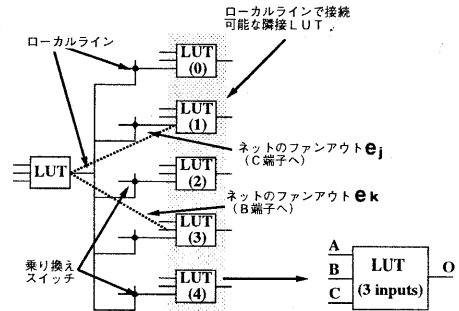


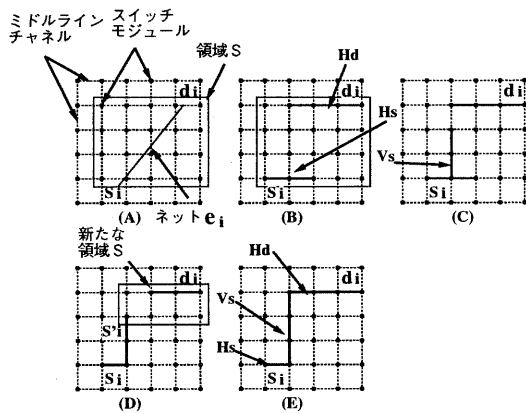
図11: ローカルラインの配線処理

LRPでは、ローカルラインで隣接LUTを接続する処理を行なう。図11に処理例を示す。ネットのファンアウト e_j はLUT(1)のC端子へ、ファンアウト e_k はLUT(3)のB端子へ接続されている。 e_j はローカルラインのみにより接続可能であるために、一意に経路が決定される。一方、 e_k はLUTの位置関係からはローカルライン接続可能であるが、B端子に接続されているため、ローカルラインのみでは接続できない。そこで、LUT内の論理を変更しB端子とC端子の入れ換えを行なうことにより、 e_j と同様のローカルラインのみで接続可能となる。この端子の入れ換え処理により他のネットのファンアウトがローカルラインのみでは配線不可能になる場合があるが、これらは後の配線処理(MRP、GRP)で経路を決定する。

3.4.3 ミドルライン配線処理 (MRP)

MRPは、PROTEUS-Lite中で最も大きな自由度を有するミドルラインを使用した経路探索を行なう。MRPの経路探索手法は、線分探索法[D.H69]を基本にしてミドルラインの構造に合わせた改良を行なったものである。

従来の線分探索法はネットの始点/終点の双方から上下左右の4方向に最長線分を探索し経路を決定している。PROTEUS-Liteはミドルラインの配線資源を十分に有しているため、線分探索の方向を、ネットの始点と終点を対角に有する長方形の辺に則した2方向のみに限定している。また、配線の折れ曲がり避けるために、始点方向からのみ探索を行なう。ここでは、ネットの始点と終点から、それぞれ最長の水平線分を探索し、次に水平線分の双方に接続可能な垂直線分を探索する。最長線分/接続可能線分の探索には、テーブル表現された使用可能な乗り換えスイッチの配置情報を連鎖的に参照することで、効率的な探索が行なえる。



1. e_i は与えられたネットである。 s_i は e_i の始点であり、 d_i は終点である。
2. e_i を囲む領域 S を作成する。(A)
3. s_i を始点とする最長の水平線分 H_s を探索する。この線分はミドルラインを切り離しスイッチのみを使用して直列に接続したものである。(B)
4. H_s と同様に d_i からの最長線分 H_d を探索する。(B)
5. 以下の条件で最長の垂直線分 V_s を探索する。(C)
 - (a) V_s は H_s に接続され、 H_d に最も近い線分である。
 - (b) V_s は d_i に最も近い線分である。
 条件を満たす V_s が存在した場合は次のステップに進む。存在しなければ、処理を中止する。
6. 5の結果に対応して以下の2つの処理を行なう。
 - (a) V_s が H_d と接している場合
 $s_i \rightarrow H_s \rightarrow V_s \rightarrow H_d \rightarrow d_i$ が最短経路となる (E)
 処理終了
 - (b) V_s が H_d と接していない場合
 s_i を V_s の終点に移動し、 s'_i として再度2からの処理を行なう。(D)

図 12: MRP の線分探索処理

図 12 に MRP の配線探索手順を示す。ここでは、分岐の無いネットの処理手順を示す。我々はこの経路探索を”Longest line Search Heuristics”(LSH) と呼ぶ。LSH は最悪でも、領域 S 内の全てのスイッチモジュールを探索するだけで経路を発見することができ、ダイクストラ法と比較して計算量は少ない。尚、MRP ではファンアウトを有するネットの分岐点を IRP と同じ手法で決定している。

3.4.4 未配線処理 (GRP)

ここまでの配線処理は配線経路が発見される事を保証していない。GRP はここまでの処理で経路が確定していないネットに対して経路探索を試みる。未配線ネット発生の主な原因は、MRP で使用できる領域 S 内の配線資源の枯渇である。そこで GRP では未使用の配線資源に加え、未使用の LUT も配線資源として流用する。

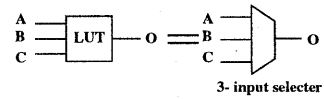


図 13: LUT のセレクタへの転用

図 13 に示す様に PROTEUS-Lite の LUT はプログラムで端子を選択可能な 3 入力のセレクタと同一に取り扱う。GRP ではダイクストラ法で経路探索を行なう。この段階では、候補となる配線資源が十分少なくなっており、ダイクストラ法でも計算量は増加しないためである。ファンアウトの分岐点の決定は IRP と同様である。

3.5 手動設計ツール

PROTEUS-Lite 内のリソースを図的に表現し、人手で論理、配線の編集が可能な回路エディタ (PLEDIT) を準備した。

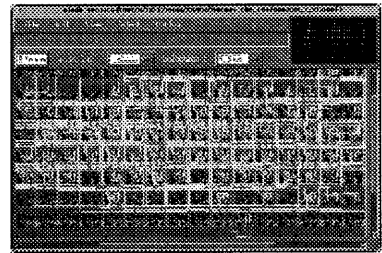


図 14: PLEDIT の画面

PLEDIT の編集中のイメージを図 14 に示す。PLEDIT は、手設計による論理回路の設計、配置処理、配線処理後の回路編集等をサポートする。また、ハードマクロと呼ばれる論理部品を作成できる。

3.6 検証ツール

PLCAD では設計した論理回路を PROTEUS-Lite に搭載する前に論理検証、遅延検証、電気的な接続検証を行なうためのツールが準備されている。

3.6.1 論理検証

設計した論理回路の検証は、論理シミュレーションを実行することで行なう。PLCAD の内部ネットリストは PROTEUS-Lite に特化した UPNF と呼ばれる形式で記述されている。この UPNF を Verilog-HDL に変換するツール (UPNF2VL) を用意し、市販シミュレータによる論理検証をサポートしている。

3.6.2 遅延検証

PROTEUS-Lite に搭載した論理回路の性能を見積もるために静的な遅延評価を行なう PLDELICAL を準備した。FPGA の配線遅延は用いられる配線の種別、および組合せ

により異なるために計算が複雑になる。PLDELCALでは、切り離しスイッチにより分離される領域内での配線の組合せごとに遅延評価関数を用意し、それらによる計算結果を積算する方法 [KTMF97] で算出しているの、効率的な遅延計算が行なえる。実チップへ搭載した論理回路の動作速度と PLDELCAL の結果を比較したところ、平均で4%程度の誤差であった。

3.6.3 電氣的接続の検証

手動での論理回路設計では配線のショートや未配線等の設計誤りが起こりうる。これらの設計誤りにより、回路が正常に動作しないばかりか、PROTEUS-Lite 自体をも破壊する。そこで、配線のショート、不完全な配線を検出するツール (UPNFCK) を準備した。UPNFCK は UPNF 形式のネットリストを読み込み、配線の電氣的な妥当性の確認を行なう。この検証が終了した後に PROTEUS-Lite に論理を搭載するためのプログラムデータを UPNF2PBF を使用して生成する。

4 評価

実際の通信処理に使用される回路をサンプルとし、PLCAD の自動設計ツールを使用して PROTEUS-Lite への回路搭載実験を実施した。通信回路の Verilog-HDL 記述から Synopsys 社のツールで論理合成を行ない、PLCAD でマッピング、配置、配線を行なった。PLCAD システムは全て C++ で記述されており、SunOS 上で実行可能になっている。我々は、実行環境に SS20 を使用した。

表 2: サンプル回路のマッピング結果

回路名	処理前		処理後		処理時間
	#GT	#FF	#LUT	#FF	
fcount	67	13	45	13	0.3
hec	109	48	85	48	0.6
cellgen	187	54	136	54	1.2
stm1gen	308	93	228	93	2.6
sigg	863	209	612	209	13.7

#GT: 論理規模 #FF: FF 数
#LUT: LUT 数 処理時間 (秒)

表 2 にサンプル回路の PLCOVER でのマッピング結果を示す。入力のネットリストは 2 入力の AND、OR 等の基本ゲートで構成されたものである。マッピングの結果、平均で 1.5 個程度の基本ゲートを 1 つの LUT に搭載している。PLCOVER では FF は最適化しないので、マッピング前後で変化しない。処理時間はサンプル中最大の sigg でも 13.7 秒であり、高速である。

表 3: 配置処理結果

回路名	処理前		処理後	処理時間
	#LUT	#FF	#BC	
fcount	45	13	21	58
hec	85	48	24	61
cellgen	136	54	38	66
stm1gen	228	93	60	73
sigg	612	209	219	98

#LUT: LUT 数 #FF: FF 数
#BC: BC 数 処理時間 (秒)

表 3 に PLFIX による配置処理の結果を示す。処理時間では、サンプル中で最大の sigg でも 98 秒と高速に処理している。

表 4: 配線処理の各段階での配線率

回路名	配線処理段階			
	IRP	LRP	MRP	GRP
fcount	27	37	38	0
hec	38	39	23	0
cellgen	6	37	57	0
stm1gen	15	32	53	0.2
sigg	7	32	61	0.1

数値は処理率 (%)

表 4 は PLROUTE の各段階で配線処理されたネットのファンアウト数の割合を示している。ここで、LRP、MRP で処理されるファンアウト数は与えられたサンプル回路中の約 80% である。LRP は平均で 35% のファンアウトを処理している。これは、PLFIX の配置処理にローカルラインの構造を考慮した効果である。hec では MRP で処理されるファンアウトは 23% と少ない。これは、回路規模に比較して I/O が多く 38% のファンアウトが IRP で処理されているからである。

表 5: 配線の使用状況

回路名	配線の種別				
	LL	HM	VM	HL	VL
fcount	227	249	49	10	2
hec	403	235	119	0	4
cellgen	634	434	264	0	6
stm1gen	1120	1060	405	1	10
sigg	2860	3858	1417	27	9

LL: ローカルライン HM: 水平ミドルライン
VM: 垂直ミドルライン HL: 水平ロングライン
VL: 垂直ロングライン

表 5 に各配線種別の使用量を示す。ローカルラインの使用量が多いのは、BC や IOB に配線する場合、必ずローカ

ルラインを通過しなくてはならない事と、IRPで配線する場合でもローカルラインは2本必要であるためである。

表 6: 配線の階層構造を考慮しない処理との比較

回路	配線処理			
	PLROUTE			PLDIJK
回路名	処理時間	高速化率	遅延比較	処理時間
fcount	35.8	12.7	0.46	457
hec	37.1	4.2	0.98	157
cellgen	39.5	8.5	0.91	330
stm1gen	71.7	7.2	1.03	513
sigg	244.6	12.0	0.94	2933

処理時間の単位は(秒)

配線リソースの構造を考慮しない配線処理と比較を行なうために、配線資源の第二階層を単一アルゴリズム(ダイクストラ法)で処理する PLDIJK を準備し、PLROUTE との比較を行なった結果を、表 6 に示す。処理時間は SS20 での実行に要した時間(秒)であり、高速化率は PLDIJK との処理速度の比である。また、遅延比は搭載した論理回路の性能を PLDELCAL を使用して評価した結果を PLDIJK の結果を 1 として規格化したものである。

第二階層のミドルラインを高速な MRP で処理を行なう PLROUTE は PLDIJK と比較して平均で 8 倍程度高速になっている。配線処理結果の性能評価では PLDIJK と同程度か、高速化されている事がわかる。これは、PLDIJK は同一種類の配線は同じ重みとして経路を探索しているの、配線乗り換えの手段である、乗り換えスイッチ、切り離しスイッチを利用する際の信号伝搬遅延の違いを考慮していないためである。このことから、配線資源の構造を考慮した配線処理手法は配線処理効率、処理された論理回路の性能に効果的であると言える。PLCAD 全体の処理時間では、最大の sigg でも約 356 秒と短時間である。

5 まとめ

豊富な配線資源を階層的に持つ通信処理用 FPGA である PROTEUS-Lite の専用 CAD システム (PLCAD) について述べた。PLCAD は RTL 記述からのトップダウン設計環境を提供している。PLCAD の各ツールは PROTEUS-Lite の構成を強く意識した処理を行なう。これにより、FPGA の CAD の中で最も大きな計算量が必要な配線処理を、構成を考慮しない場合に比べ 8 倍程度の高速化を達成できた。今後は、搭載した論理回路の動作速度向上を目的に、配置処理、配線処理アルゴリズムの改善を行なっていきたい。

参考文献

- [AR95] M.J. Alexander and G. Robins. New Performance-Driven FPGA Routing Algorithms. In *Proc. of ACM / IEEE DAC*, pp. 562-567, 1995.
- [ATT93] AT&T Microelectronics. Optimized reconfigurable cell array (ORCA) series Field-Programmable Gate Arrays. *Advance Data Sheet*, 1993.
- [Bro92] S.D. Brown, R.J Francis, J. Rose and Z.G. Vranesic. Field-Programmable Gate Arrays. *Kluwer Academica*, 1992.
- [D.H69] D.Hightower. A Solution to Line-Routing Problems on the Continuous Plane. *Proc. of 6th Design Automation Workshop*, pp. 1-24, 1969.
- [KTMF97] M. Katayama, A. Takahara, T. Miyazaki, and K. Fukami. Delay Calculation Method for SRAM-based FPGAs. *IEICE Trans. Fundamentals*, Vol. E80-A, No. 9, pp. 60-65, 1997.
- [MTK⁺97] T. Miyazaki, A. Takahara, M. Katayama, T. Murooka, T. Ichimori, K. Fukami, A. Tsutsui, and K. Hayashi. CAD-oriented FPGA and Dedicated CAD System for Telecommunications. In *Proc. of Field-Programmable Logic (FPL)*, 1997.
- [ONY⁺94] N. Ohta, H. Nakada, K. Yamada, A. Tsutsui, and T. Miyazaki. PROTEUS: Programmable Hardware for Telecommunication Systems. In *Proc. of IEEE International Conference on Computer Design (ICCD)*, pp. 178-183, 1994.
- [TWM94] S. Thakur, D.F. Wong, and S. Muthukrishnan. Algorithms for FPGA Switch Module Routing. In *Proc. of ACM / SIGDA Workshop on FPGAs*, January 1994.
- [筒井 95] 筒井章博, 宮崎敏明, 中田広, 山田一久, 太田直久. 通信処理用 FPGA 専用 CAD システム. '95 回路とシステム 野井沢ワークショップ, pp. 73-78, 1995.
- [林 95] 林一博, 山田一久, 筒井章博, 宮崎敏明. フレキシブルトランスポートシステムの新パラダイム. CS 研究会, March 1995.