

ビットシリアル PCA の構成法に関する研究

生産科学研究科 永本 太一

ユビキタス時代を迎え、フォンノイマン型コンピュータは生活に多大な影響を与え社会の隅々まで浸透している。フォンノイマン型コンピュータの発展の鍵はその汎用性と柔軟性にあり、ハードウェアとソフトウェアからなる構造が汎用性の、メモリ上のデータ表現やオブジェクトインスタンスを `malloc` や `new` 操作などによって動的に生成できるヒープ管理が柔軟性の源となっていると考えられる。

PCA (Plastic Cell Architecture) は、この汎用性と柔軟性に着目して提案された新しいコンピューティングアーキテクチャである。フォンノイマン型コンピュータがプログラム論理をベースとするのに対し、PCA は布線論理をベースとし、動作中の回路が新たな回路を生成して協調動作することができるという特徴がある (以降このことを動的再構成機能と呼ぶ)。PCA はルーティングネットワークの中にメモリと FPGA をちりばめたアーキテクチャでメモリ上のデータ間の加工・転送を CPU を介さずに FPGA 上の論理回路とルーティングネットワークにより直接かつ並列的に行うことができる。

PCA に関連する研究として、NTT の研究グループは PCA の最初の LSI である PCA-1 を製造し、特徴的な機能のすべてが正しく動作することを確認した。さらに同グループは性能向上を目指した二つ目の LSI である PCA-2 を製造している。PCA-2 の設計に際しては、PCA-1 の評価結果に基づくアーキテクチャおよび設計手法の改良と先端プロセスの利用による性能向上が図られている。以上のように PCA アーキテクチャの改良に関する研究は盛んに行われている。しかし、PCA-1、PCA-2 は粒度の小さい再構成可能単位を 2 次元メッシュ上に均質に配置した構造をとっており自由度が高いことが利点であるが、配線を構成する場合にも LUT を使うため配線が長くなると、遅延が大きくなる、集積度が低くなるという欠点がある。これは、粒度が小さすぎるためであり、これを解決するには再構成要素の改善に関する研究が必要となる。

一方、アプリケーションに関する研究には、PCA の最大の特徴である動的再構成機能を積極的に使った研究が盛んに行われており、トラヒックの変動などに合わせて負荷を分散させる適応負荷分散処理などの研究が行われている。しかし、PCA には動的再構成機能を使うための `malloc` または `new` 操作のような領域管理法が未だ提供されていないため領域管理は人手によって行わなければならない。つまり、動的再構成に使う空き領域はあらかじめ設計者が確保しておかなければならないという問題がある。これは、資源の払い出し要求が同時に複数発生する布線論理の世界と資源の管理という考えが合致しないところに未だ有力な領域管理法が提案されない原因がある。動的再構成機能を十分に活用するためには領域管理機構の研究が重要となる。本研究は、PCA アーキテクチャのさらなる性能向上をねらうものであり、特に、遅延が少なく集積度の高い再構成要素と領域管理機構に焦

点を当てる。

本論文では大規模なアプリケーションとして IMBE と呼ばれる音声に特化した音声符号化処理を選択してそのリファレンスモデルを C 言語によるソフトウェアとして作成するとともに、非同期ビットシリアル方式での開発が可能な設計環境である QROQS を用いて、できるだけ高性能になるよう非同期パイプラインの動作を整理し、性能を表すパラメータを考慮しながら PCA 用の実装を進めた。次に正しく動作していることを確認するためにリファレンスモデルの結果と比較を行いながら、試行錯誤を進め PCA 用の典型的な構造パターンを抽出することに成功した。この典型的な構造パターンは 3 種類のステートマシンと入出力スイッチからなり、ステートマシンはコントロール部、データサイズ判定部、演算部の 3 種類必要であることがわかった。この調査をもとに PCA 可変部として組み込むためにその実装方法を検討した。

一方、領域管理機構に関しては圧力の概念を基本とした新しい領域管理方を提案しその詳細を述べた。この領域管理法を実現するために考案した 3 種類のコマンドセットとその評価方法について述べている。最も優れていることがわかったコマンドセット 2 については完成できるまでに発生した問題の関係などを含め詳細な議論を行った。この考案により管理者のない全く新しい領域管理法のための基本的な動作を明らかとすることができた。

圧力によって得られた形のわからない領域に回路を構成するためには回路を増殖させながら配置する必要がある。これも PCA に存在した大きな問題であった。これに対しわずか 3 つの増殖ルールを適切な順序で適用すれば任意の回路構造を作れることをグラフの操作をベースに証明した。またこの証明の中で増殖順序も決定される。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場合にも、ハードウェア量の増大を十分に抑えることができると考えられる。

以上によって NTT の PCA-1, PCA-2 と比べてはるかに完成度の高い PCA アーキテクチャを得ることができたと考えている。