動的再構成可能プロセッサのシステムレベルからの段階的設計法に関する研究

北　道　淳　司　　会津大学コンピュータ理工学部上級准教授

はじめに

近年、チップ内部の構成をシステムの動作中に変更することのできる動的再構成可能プロセッサが、NEC エレクトロニクス[1]、IP フレックス[2]、SONY など多くの半導体メーカ、大学の研究機関から提案・発表されている。

動的再構成可能プロセッサは、必要に応じて動作を変更することが可能であるので、大規模なシステムを少ないハードウエアリソースにより実現する可能性があり、ネットワーク通信システムにおけるサーバシステムでのネットワークスイッチ機構において増大するプロトコルの種類に対応するスイッチング機能を実現することが可能であり、携帯端末においてより省電力（一般に省電力のデバイスは演算能力は劣る）のハードウエアリソースを用いても、操作する種類ごとにシステム構成を変更させ多種類の機能を実現するなど、今後ネットワークの基幹システムから端末システムにいたるまで、主要なハードウエア素子として注目されている。

しかし、それぞれの動的再構成可能プロセッサは、演算機能の粒度、切り替えの機構などそれぞれ異なり、それぞれ得意とするアプリケーションが異なる。今後、開発すべきシステムのために動的再構成可能プロセッサの新規開発や、多種の動的再構成可能プロセッサからなるシステム LSI 開発においては、既存の個々の動的再構成可能プロセッサ用あるいは ASIC 用の開発環境では対応することができない。

本研究では、以降で述べる手順により、システムレベル設計言語 SystemC[3]とハードウエア記述言語 VerilogHDL を用いて、システムレベルから RTL（レジスタ転送レベル）までの動的再構成可能プロセッサの仕様記述法および段階的設計法を提案した。

2 提案する段階的設計法

2-1 準備

プロセッサのシステムレベルからレジスタ転送レベルまでのプロセッサ設計のシステムレベルとして命令レベルでの動作を扱い、レジスタ転送レベルとして、1 クロックにおける各レジスタ間のデータ転送を扱う。また、これらの中間的なレベルとして、たとえばパイプラインアーキテクチャを採用する場合、非パイプライン制御（たとえばマルチサイクルアーキテクチャ）を採用したモデルを作成し、データパスの正当性検証に利用する。

動的再構成可能プロセッサでは、動的に再構成されるハードウエアリソースの粒度を考慮し、各レベルでの動的再構成機能を設計する。

動的再構成可能プロセッサは、その再構成機能がさまざまであり、ある特定のプロセッサに依存することなく、多様なアーキテクチャに対応可能な設計が可能なように対応させる。

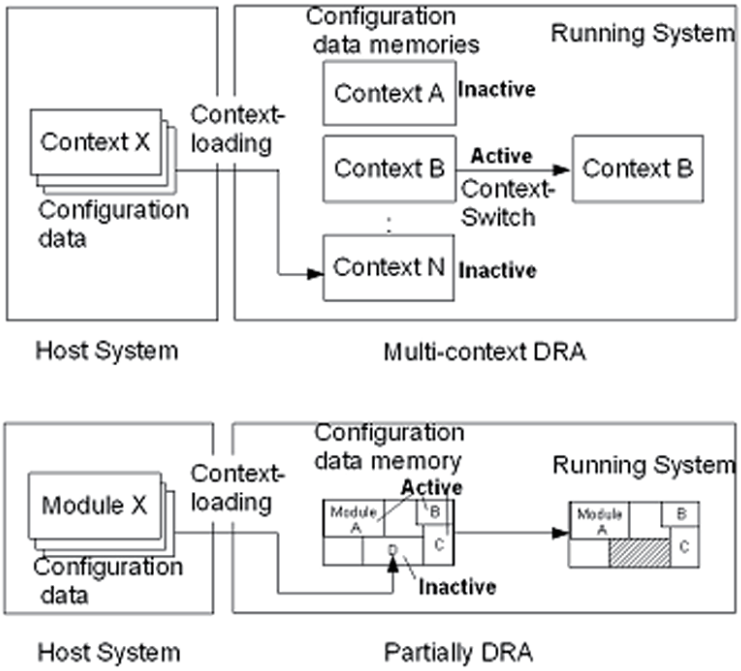

図１　２種類の動的再構成可能アーキテクチャ

具体的には、マルチコンテキスト型アーキテクチャ（図１の上図）、部分再構成型アーキテクチャ（図２の下図）および、ハードウエアリソースが潤沢にある場合の通常のリコンフィギャラブルアーキテクチャへのマッピングを検討する必要がある。

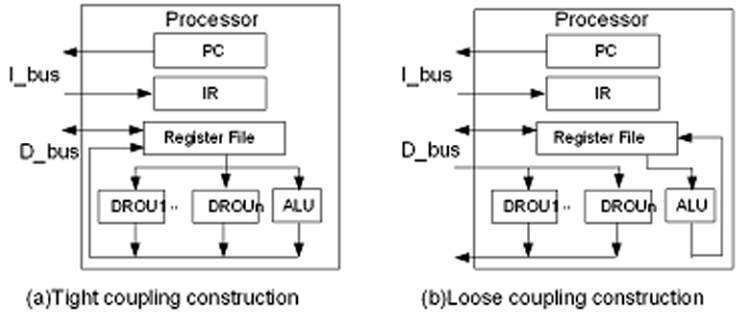

図２　２種類の動的再構成可能プロセッサアーキテクチャ

2-2 各レベルでのモデリング

設定した様々な設計レベルにしたがい、動的再構成可能プロセッサのシステムレベルからレジスタ転送レベルまでのモデルを設計する。具体的にはMIPS 命令セットアーキテクチャ[7]を基に、演算機能を動的に生成する機能を追加した。システムにおいて、自由にモジュールを動的に生成削除するような構成のモデルではなく、動的再構成可能モジュールを生成削除可能な特別なモジュールを設定し（図２における DROU)、その中で複数の動的再構成可能モジュールを生成削除可能とするようにモデリングした。

このモデリング方法によって、システムにおいて動的である部分と動的でない部分を分離し、それらのインターフェース（接続するポートの数や種類）を確定(静的な設計)することにした。このことにより以降で述べる段階的設計支援が可能となる。本研究においては、浮動小数点演算を採用したが、さまざまな演算機能に対して適用可能である。動的再構成機能のモデリングのために、従来研究においてモジュール、チャネル、ポートの動的な生成削除のモデリングを可能とする Dynamic Module Library[4,5]を用いて効率的にモデリングを行うことができた。

2-3 段階的設計支援機能の実現

各レベルでのモデルを参考に、下位レベルの設計あるいは設計の一部を合成可能な箇所を抽出した。システムにおいて動的に変更されない機能および動的再構成を管理する機能は、システムの動作中常に存在するようにすべきである。この部分は従来の設計手法を用いて設計支援が可能である。動的に生成削除される動的再構成可能モジュールは、特定の機能モジュール内でのみ生成削除が可能であるようにしたので、そのエリアは、同時に生成される動的再構成可能モジュールのエリアの和で与えられ、最高動作周波数はそれらの動作周波数の最悪値で与えられる。システムのパフォーマンスを向上させるには、これらの値を決定する動的再構成可能モジュールを最適化する、あるいは、より上位の設計レベルから再設計を行うという設計手法が考えられる。これらの設計支援に関しては今後の課題である。

2-4 モデルの正当性の検証

上記手順で開発したモデルの正当性を、テストベンチを作成しシミュレーションによって検証する。動的再構成プロセッサのシミュレーションに要する時間、必要メモリなどを計測し、直接具体的なレベルで設計を行うより、高速なシミュレーションにより、命令レベル、RT レベルそれぞれのレベルで効率的な検証を行うことが可能であった。

3 まとめと今後の課題

以上のように､システムレベルから RT レベルまでの段階的な動的再構成可能プロセッサのためのモデリング手法を提案した。本手法では、モジュールを動的再構成するための Dynamic Module Library を用いて、各レベルにおいて自然にプロセッサアーキテクチャを表現することが可能であり、 SystemC および Dynamic Module Library の機能により各レベルにおいて高速なシミュレーションによるシステムの動作の正当性検証を行うことができた。今後、これらの設計を既存の CAD ツールを用いて実際のデバイス上に実装するのは今後の課題である。

段階的設計を支援する方法の概要についても本研究で提案することができたが、これらの詳細の検討、支援システムの開発に関しても今後検討を行う。

【参考文献】

[1] H. Nakano, T. Shindo, T. Kazami, and M. Motomura, ``Development of Dynamically Reconfigurable Processor Lsi," NEC Technical Journal, Vol.56, No.4, pp.99-102, 2003.

[2]T.Toyo, H.Watanabe, and K.Shiba. ``Implementation of Dynamically Reconfigurable Processor DAP/DNA-2," Proc, of IEEE VLSI-TSA Int. Sympo. on VLSI Design, Automation & TEST, pp.321-322, 2005.

[3]IEEE, ``IEEE Standard SystemC Language Reference Manual," http:// standards.ieee. org /getieee/1666/, 2006.

[4]K. Asano, J. Kitamichi, K. Kuroda, ``Dynamic Module Library for System Level Modeling and Simulation of Dynamical Reconfigurable Systems," Journal of Computers, Vol.3, Issue 2, pp.55-62, Academy Publisher, 2008.

[5]J. Kitamichi, K. Ueda, and K. Kuroda,``A Modeling of a Dynamically Reconfigurable Processor using SystemC," 21st International Conference on VLSI DESIGN 2008,pp.91-96, 2008.

[6]Xilinx,``Vertex-5 FPGA Configuration User Guide, http://www.xilinx.com/support/ documentation/user_guides/ug191.pdf, UG191(v3.6), 2009.

[7]G. Kane, ``MIPS RISC Architecture -R2000/R3000-," Prentice-Hall, 1988

〈発 表 資 料〉

題　　名	掲載誌・学会名等	発表年月
Proposal of Instruction Level Modeling for Dynamically Reconfigurable Processor using SystemC	The 17th IFIP/IEEE International Conference on Very Large Scale Integration(VLSI-SOC 2009)	2009 年 10 月 （発表予定）