

Linux における動的再構成可能プロセッサの実行管理機構の試作

高島 和幸[†] 佐藤 未来子[‡] 磯部 泰徳[‡] 並木 美太郎[§]

Kazuyuki TAKABATAKE[†] Mikiko SATO[‡] Hironori ISOBE[‡] Mitaro NAMIKI[§]
 東京農工大学工学部情報コミュニケーション工学科[†] 東京農工大学大学院工学府情報工学専攻[‡]
 東京農工大学大学院共生科学技術研究院[§]

1 はじめに

近年、デジタル機器の多機能化や高精細化、多様化が急速に進み、メディア処理向けのプロセッサの開発頻度が增大しており、このコストの低減が重要な課題となっている。そのため、単一のプロセッサ上で様々なメディア処理を実現することで開発頻度を抑えるべく、メディア処理に合わせて構成を動的に変更できる動的再構成可能プロセッサ (DRP) が研究開発されている。DRP は、図 1 に示すように 2 次元アレイ状に多数の演算器やメモリを配置することで並列演算が可能で、マルチメディア処理や各種のデータ処理、繰り返し構造の多いデータ処理において有効である [1]。また、図 2 のように汎用プロセッサと DRP を連携させることにより、汎用のアクセラレータを実現可能である。

本研究では、汎用 OS である Linux を用いた環境において、ユーザプログラムから DRP の制御と管理を行うシステムを構築する。DRP の例として、Flexible Engine/Generic ALU Engine (FE-GA) [2] を使用する。動的再構成の構成情報であるコンフィギュレーションデータの管理、汎用プロセッサと DRP 間のデータ転送のための API を用意し、Linux 上でユーザプログラムから DRP を動作させるための環境を目指す。

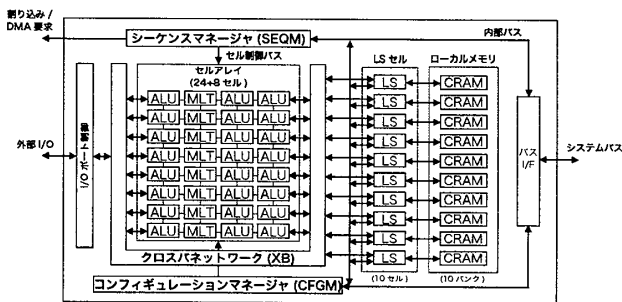


図 1: 動的再構成プロセッサの例

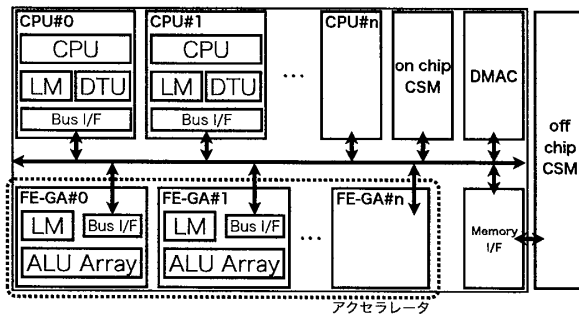


図 2: DRP 連携の例

2 問題分析

汎用プログラムから FE-GA を利用するための基本手順は以下のとおりである。

- 1) FE-GA の初期化

内部状態のリセットや割り込み設定など、FE-GA を使用するための初期化・事前処理を行う。
- 2) コンフィギュレーションデータの転送

コンフィギュレーションデータ (CFGD) はコンフィギュレーションマネージャ (CFGM) と呼ばれるユニットに書き込む。
- 3) コンフィギュレーション設定

複数のコンフィギュレーションを切り替えることで、処理内容を高速に変更することができる。ここでは使用したい処理に合わせてコンフィギュレーションの面を指定する。
- 4) 入力データ転送

FE-GA では演算データの一時格納のため最大 16KB×10 バンクのローカルメモリがあり、外部メモリから入力データを転送する。
- 5) FE-GA の起動

FE-GA を起動して演算を開始させる。
- 6) 出力データ転送

演算結果をローカルメモリから外部メモリへ転送する。
- 7) 3) へ戻り、次の処理のためにコンフィギュレーションを切り替える。

これらの処理を行うためには、ユーザはすべて明示的にアドレス等を指定してコンフィギュレーション設定や、データの転送先/転送元を指定する必要がある。FE-GA の内部構造によく精通していなければならない。

Prototyping of a Program Management System for Reconfigurable Processor with Linux

[†] Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

[‡] Department of Computer and Information Sciences, The Graduate School at Tokyo University of Agriculture and Technology

[§] Institute of Symbiotic Science and Technology, The Graduate School at Tokyo University of Agriculture and Technology

3 本研究の目標

前述の問題点を解決するため、ユーザプログラムからFE-GAを利用しやすい環境を提供する。理想的にはハードウェアの内部構造を意識せず、関数を呼び出すようにしてFE-GAを実行できるようにしたい。本研究においては、コンフィギュレーションデータがあらかじめ用意されていることを前提とする。

4 システムの概要

本システムのモデルを図3に示す。図中で破線になっている部分が今回作成する部分である。デバイスドライバとAPIから成り、次の機能を提供する。

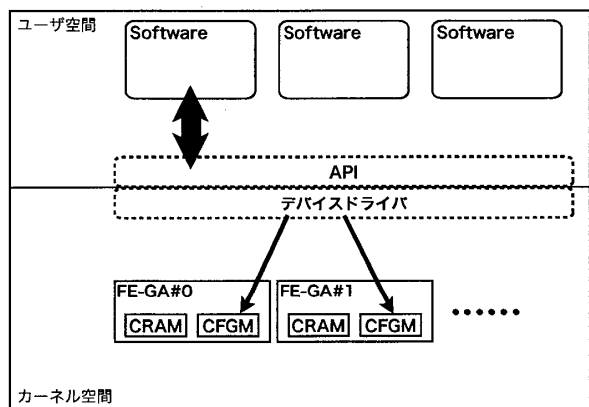


図3: システムの概要

- コンフィギュレーションデータのロード
FE-GAのコア番号と、ユーザ空間上にあるコンフィギュレーションデータのアドレスを引数として、指定したFE-GAにコンフィギュレーションデータをロードする。
- FE-GAの実行
FE-GAのコア番号と、コンフィギュレーションの指定、入力データ、出力データを引数として、FE-GA上で、入力データに対して指定されたコンフィギュレーションの処理を行い、その結果を出力データに格納する。

4.1 プログラミングモデル

イメージとして、以下のようにAPIを利用する。

```
drp_load(0, myCFGD);  
drp_exec(0, FFT, input, output);
```

このようにユーザプログラムからFE-GAの内部をほとんど意識しないプログラムスタイルを提供する。

5 本システムの設計

前述のプログラムスタイルを実現するために、内部でどのような処理を行うか述べる。

5.1 コンフィギュレーションデータのロード

コンフィギュレーションデータのロードでは、指定された番号のFE-GAのコンフィギュレーションマネージャに、コンフィギュレーションデータを引き渡す。

5.2 FE-GAの実行

まず、入力データを入力バッファに格納する。次にFE-GAの実行の準備として、初期化を行い、指定されたコンフィギュレーションに切り替える。

残りの処理は入出力のデータサイズに応じて異なり、いずれにしても処理が終了した場合はそのことをユーザプログラムに通知する。

- FE-GAのローカルメモリサイズより小さい入力バッファのデータをすべてローカルメモリに格納できるので、FE-GAの実行は1回だけ行う。結果は出力バッファに格納する。出力バッファのデータを指定した変数に転送し、処理を終了する。
- FE-GAのローカルメモリサイズより大きい入力バッファのデータがすべてローカルメモリに格納できないため、同じコンフィギュレーションで繰り返しデータ処理を行う必要がある。入力バッファにデータがなくなるまでFE-GAの実行を繰り返し、結果は出力バッファに格納していく。出力バッファのデータを指定した変数に転送し、処理を終了する。

6 おわりに

本論文では、Linux上でユーザプログラムからDRPを実行・制御するための機構について述べた。DRPの内部構造を隠蔽することにより、ユーザが容易にDRPを用いることが可能である。また、演算器の個数やビット幅は異なるものの、構成が比較的似ている動的再構成可能プロセッサであるDAPDNA-2(IPFlex)やDRP-1(NEC)に対しても本システムが適用できるのではないかと考えられる。

参考文献

- [1] 石川浩行, 清水翔, 荒川豊, 山中直明, 斯波康祐, 「並列リコンフィギュラブルプロセッサDAPDNA-2を用いた集合被覆問題の高速解法」, 『電子情報通信学会技術研究報告. RECONF, リコンフィギュラブルシステム』 Vol.107, No.418 pp.67-72, 2008
- [2] Hiroaki Shikano, Masaki Ito, Kunio Uchiyama, Toshihiko Odaka, et al. "Software-Cooperative Power-Efficient Heterogeneous Multi-Core for Media Processing", Proceedings of the 2008 conference on Asia and South Pacific design automation (ASP-DAC 2008), pp.736-741, 2008
- [3] 本間雅行, 戸川望, 柳沢政生, 大附辰夫, 佐藤真琴, 「再構成型プロセッサFE-GAへのフィルタマッピングとその自動化手法」, 『電子情報通信学会技術研究報告. CAS, 回路とシステム』 Vol.107, No.100 pp.67-72, 2007