FPGAを扱うOSにおける資源管理機構の試作

4M－5

名波秀昇，早川栄一，並木美太郎，高橋延匡

（東京農工大学　工学部）

1. はじめに

マルチメディア時代を迎えて、データ種類の多様化、データの大容量化が進んでいる。そのため、次のような問題が上げられる。

(1)新しいデータ処理方式に対応した専用回路を製作するためには、多くの開発費用や開発期間が必要である。

(2) データ処理に時間がかかる。

これらの問題を解決するための手段として、プロセッサ、メモリを高速化することが上げられる。また、専用回路を製作する方法などがある。しかし、前者は、フォン・ノイマン方式のため、演算速度とメモリアクセス時間で性能限界が決まってしまう。後者は、開発の期間、および、費用が問題となる。

他の解決手段として、再書込み可能なFPGAを計算機に適用して処理を行なう、いわゆるFPGAシステムによるものがある。この方式によれば、データの処理方式の内容を容易に変更でき、柔軟性に富んでいる。また、並列化によるデータ処理の高速化の可能性を秘めていることから有効である。FPGAシステムの設計には、ハードウェアの構成からOSの設計に至るまで多数の選択肢がある。筆者らは、予備実験として、対象となる計算機上にOSを試作した。本稿では試作したOSの資源管理機構について述べる。

2. FPGAシステムの基本設計

2.1 対象とするアプリケーション

対象とするアプリケーションは、ソフトウェアによる実現での処理としては計算処理容量が重いもので、これをハードウェア化したほうが高速化が期待できる処理内容を主な対象とする。ここでは、具体的に、画像にフィルタをかけたり、圧縮するといった画像処理や、浮動小数点演算方式のシミュレーションなどを対象とする。

2.2 FPGAを扱うOSにおける資源管理機構への要求

以下に、資源管理機構への要求を述べる。

(1) 資源の保護を行いたい。

ユーザプログラムから、OSの資源（プロセッサ、メモリ、割込み、外部デバイス、FPGA）への不当なアクセスによって、処理内容や、処理データなどの破壊を防ぐため、保護を行いたい。

(2)ユーザプログラムにはハードウェアを意識させることなくFPGA上の機能を利用できるようにしたい。

従来、計算機上での高級言語を使用したユーザプログラミングにおいて、ハードウェア（レジスタ）への手続きを特に意識していない。ユーザにおけるプログラミングの開発負担を減らすために、FPGA上の処理機能を利用する場合においても、ハードウェアを意識させることなくユーザが目的とする処理が行えるようにしたい。

(3) FPGAで行なう処理内容の決定は、ユーザプログラムから動的に変更できるようにしたい。

新しいデータ処理方式などについて検討をする際、ハードウェアによる実現では、回路の物理的な変更のため、多くの作業時間がかかる。こういった変更をプログラムによって容易に変更できれば、開発期間を短縮できる。

(4) 複数のユーザプログラムが動作するようにしたい。

複数のタスクによる並列処理によって、プロセッサを有効に使用することができる。

3. 設計方針

FPGAを扱うOSにおける資源管理機構の設計における方針を以下に述べる。

(1) 資源はすべてOSで管理する。

資源の管理は、一つの場所で、統一的に行えれば、OSを複雑化することなく構築でき、また、後での変更も容易になると考えた。そこで、FPGAを含むすべての資源はOSで管理することにした。

(2) OSの保護を行う。

資源はすべてOSで管理する。従って、OSの保護が行えれば、不当なユーザアクセスによる資源への破壊を防止することができる。

(3) FPGAの保護を行う。

FPGA上で実現されている処理内容を、OSを介して細かい粒度で使用したり、変更したりすると、処理を行う時間よりもOSにおけるオーバヘッドの方が主要な時間を占めると予想される。しかし、時間的な性能よりも、システムの信頼性の方が重要と考え、FPGAの保護を行うことにする。

4. 設計

4.1 ハードウェアの構成

FPGAシステムを実現するためのハードウェアは、図

The design and proto-type of resource management mechanism in OS deal in FPGA

Hidenori NANAMI, Eiichi HAYAKAWA, Mitarou NAMIKI and Nobumasa TAKAHASHI

Department of Computer Science, Faculty of Technology, Tokyo University of Agriculture and Technology

1 のように、対象とする計算機と FPGA ボードから構成する。FPGA ボードは、Futurebus を介して計算機と接続される。このボードは、FPGA における処理機能を決定するデータを格納する SRAM と、データを入出力するための SRAM などで構成する。データの入出力用の SRAM は、FPGA ボードにおけるバス制御を簡単にするために用意した。

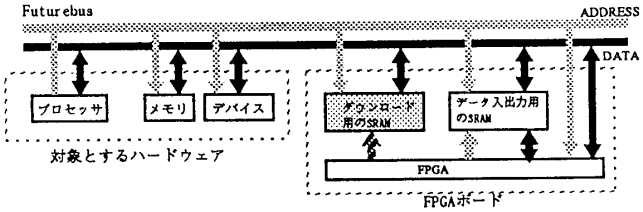

図1　ハードウェアの構成

4.2　OS の構成

2.2 で述べた管理機構への要求より、OS は図 2 に示す 5 つに大きく分けることができる。本 OS の特徴を以下に示す。

(1) OS を通して、ユーザから FPGA への操作は行われる。

FPGA への操作に関して、ユーザにはライブラリとして提供し、OS への呼出し（以下 SVC:Super Visor Call）によって行う。

(2) ユーザ機能拡張部で、FPGA への操作は行われる。

FPGA への管理を集中的に行うことにより、保守性、拡張性が向上する。新しい機能はここに登録される。

(3) 通信・同期・協調管理部で、FPGA の排他処理を行う。

FPGA への複数のアクセスを防止するため、セマフォによる排他処理を行う。

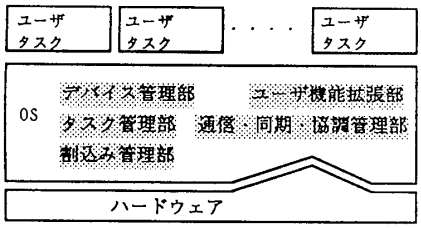

図2　OSの構成図

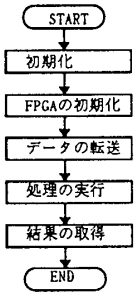

図3　処理フロー

4.3　FPGA へのアクセス機構

(1) SVC 型を採用

FPGA へのアクセス機構のモデルとして、表 1 に示す 3 つの方法の検討を行った。方式 (1) は SVC によるオーバヘッドが大きく、また、細かい粒度の利用には不利である。また、方式 (2) は、要求にあるシステムの保護を完全に行うことができない。そこで、SVC によって FPGA への操作を行うことにより、OS、FPGA の保護が行え、かつ、(1) よりオーバヘッドを小さくすることが可能な SVC 型を採用する。

(2) 処理機能の利用方法

FPGA 上に実現する処理機能を利用する場合の処理フローを図 3 に示す。

＜初期化＞

FPGA への操作を行うための処理ルーチン、および、FPGA の処理機能を決定するデータをユーザ機能拡張部に登録する

＜FPGA の初期化＞

FPGA の処理機能を決定するデータをダウウンロード用の SARM に転送し、FPGA に読み込ませる。

＜処理データの転送＞

処理を行うデータをデータ入出力用の SRAM か、主記憶メモリ上のレジスタに転送する。

＜実行＞

FPGA が実行する命令をレジスタに転送する。

＜結果データの取得＞

実行の終了は、実行終了フラグ、あるいは、割込みで検出し、処理結果をユーザに戻す。

表1　FPGAアクセス機構のモデルの検討

項目 ＼ 方式	方式 (1) クライアント／サーバ型	方式 (2) 専用プロセッサ型	方式 (3) SVC型
不正なアクセスの回避が可能	○	×	○
システムコールによるオーバヘッド	×	○	△
FPGAの利用粒度が細かい時の対応	×	○	×
ハードウェアを意識させない	○	×	○

5. 実現

資源管理機構の開発において、OS は UNIX を用い、言語処理系は BSD 系のクロスの C コンパイラ、アセンブラを用いて行った。現段階において、割込み管理機構を実現している。その行数と、コンテキスト切り替えにかかる時間の測定結果を以下に示す。なお、対象としたハードウェアの仕様を表 2 に示す。

C 言語　　：約 1200
アセンブラ：約 400
コンテキスト切り替えにかかる時間：約 73.8usec

表2　対象としたハードウェアの仕様

プロセッサ	R3081E
主メモリ	16M×16
I/Fデバイス	RS-232C (2チャンネル) Ethernet
採用バス	Futurebus+
備考	簡易ROMモニタ搭載

6. おわりに

今後は、試作した OS の評価を行い、FPGA システムとしての実現可能性の検討を行う。

参考文献

[1] 中原雅彦，中川正樹，高橋延匡「URR 浮動小数点演算コプロセッサのアーキテクチャの検討と言語 C 処理系 cat による利用方式」，情報処理学会第 36 回全国大会，1989

[2] 斎藤正伸，多田好克「ソフトウェアの部分的なハードウェア化による高速化技法の研究」，第 36 回プログラミングシンポジウム，1995