

省電力 MIPS プロセッサ評価ボードへの Linux の移植

茂木 勇[†] 木村 一樹[‡] 砂田 徹也[§] 並木 美太郎[¶]

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

本研究では、省電力 MIPS プロセッサ評価ボードである Geysers-0 on FPGA 環境へ Linux を移植を行う。

2 背景

近年、システム LSI はその急激な高性能化に伴う消費電力の増大という問題に直面している。その省電力化を図るため、細粒度パワーゲーティングという技術を適用した MIPS R3000 ベースの CPU, Geysers-0 の開発が行われた。細粒度パワーゲーティングとは、利用していない回路に対する電力の供給を遮断するパワーゲーティング技術を、ALU, シフトなど各演算器毎に個別に適用したものである。この技術を適用することにより、回路使用時に消費されるダイナミック電力だけでなく、回路不使用時にも消費されてしまうリーク電力も共に削減することができる。本研究では Geysers-0 の動作環境として Xilinx 社製の FPGA ボードである ML501 上に構築された Geysers-0 の評価環境を用いる。先行研究にて実現されたこの環境では、RTL のシミュレーションよりも高速かつ実機に近い動作を実現でき、また FPGA ボードに搭載された SRAM, シリアルポート, 電力評価機構, およびアドレスへのブレイクポイントの設置やクロック単位のステップ実行等のデバッグ機構を利用できる。

3 本研究の目標

本研究では、Geysers-0 on FPGA 環境へ Linux を移植を行い、既存のアプリケーションの評価環境を整えることを目標とする。既存のアプリケーションを Geysers-0 環境上で動作させる場合、C ランタイムライブラリをはじめとした、数値計算や暗号化、データベースや各種アルゴリズムなど、既存の多くのプログラミングライブラリを含めた移植作業を行わなければならない。また、移植にあたっては一般的な C 言語プログラミングの知識の他に、メモリマップの理解と対応するコードの生成、スタックポイントレジスタの設定等 MIPS R3000 及び Geysers-0 アーキテクチャに対するより深いプログラミングへの理解が必要となる。そこで、Linux を Geysers-0 on FPGA 環境へ移植し、その環境に大き

く依存する処理を Linux で吸収する。そしてアプリケーションからは Linux のシステムコール経由でそれらの機能を利用することにより、Linux 用に作成されたアプリケーションの移植作業の手間を大幅に軽減することができると思われる。これにより、幅広く使われている既存の、特にサーバーアプリケーションなどの、パワーゲーティングによる消費電力の削減効果が大きいと考えられるアプリケーションの移植を行い、省電力化の効果を検証することができるようになる。

4 設計

Geysers-0 on FPGA 環境上に Linux システムを構築する。本システムでは、OS として Linux 2.6.30.9, 入力装置としてシリアルポート, ファイルシステムとして tmpfs, C ランタイムライブラリとして uClibc 0.9.30.1 を利用する。また、Linux カーネルは Geysers-0 アーキテクチャに適合するよう修正を行い、シリアルポートと通信するためのデバイスドライバを追加、そして電力評価機構からカウンタの読み出しを行う機能の追加を行う。

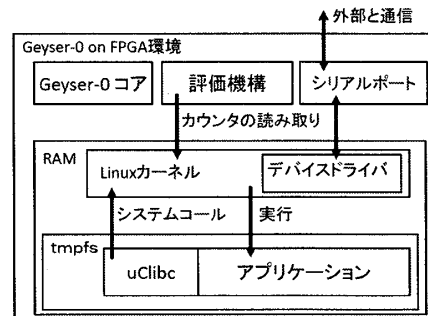


図 1: 本システムの概要

5 実装

Geysers-0 on FPGA 環境への Linux を移植作業は、おおまかに Geysers-0 アーキテクチャへの移植のためのソースコードの修正と FPGA 環境へ移植するためのソースコードの追加に分けられる。

5.1 Geysers-0 アーキテクチャへの移植

Linux ソースコードツリーには MIPS R3000 用のソースコードが存在する。今回の移植ではそのソースコードを利用することになるが、Geysers-0 アーキテクチャでは、ベースとなる MIPS R3000 アーキテクチャからいくつか変更が行われている。すなわちプロセッサ ID レジスタ, コンテキストレジスタの省略, EntryHI レジスタ, ステータスレジスタの動作変更, TLB の数の変更などである。これらの変更点をカバーするため MIPS R3000 用の Linux のソースコードに次に示すような変更を行った。

(1) プロセッサ ID レジスタの代用

Geysers-0 アーキテクチャにおいては、プロセッサ ID レジスタが存在しない。また、Linux では、同一カーネ

Porting Linux to evaluation board for Power-saving MIPS Processor

[†] Isamu MOGI

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

[‡] Kazuki KIMURA

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

[§] Tetsuya SUNATA

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

[¶] Mitaro NAMIKI

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

ルでもある程度異種のプロセッサで動作することができるよう、プロセッサ ID レジスタを利用して FPU の有無や TLB の数などを実行時に確認する。本研究で用いる Linux 環境ではそのような機能は不要と考えられるため、オリジナルソースコードの、プロセッサ ID レジスタを利用して機能を選択している箇所を、Geysers-0 on FPGA 環境へ自動的に適合されるよう修正を行った。

(2) コンテキストレジスタの代用

コンテキストレジスタが存在しない。コンテキストレジスタは、上位 4 ビットに現在利用している TLB ページオフセットが格納可能であり、また TLB 関連の例外発生時の仮想アドレスの上位 20 ビットが格納される。Linux においては、TLB ページオフセットはコンテキストレジスタを利用してない。また、例外発生時の仮想アドレスは、BadVAddr レジスタの内容を加工することにより取得することができる。

(3) EntryHI レジスタの変更点

TLB 関連の例外発生時に EntryHI レジスタの仮想アドレスビットが更新されない。これも BadVAddr レジスタで代用した。

(4) ステータスレジスタ変更点

ステータスレジスタの、コプロセッサ利用可能性ビットが利用できない。Linux においては、そのビットをカーネルスタック設定関連の用途に利用しているため、その処理を代行するコードを追加した。

(5) TLB の数の変更点

MIPS R3000 アーキテクチャには 64 個の TLB が存在する。Geysers-0 の TLB の個数はそれとは違い 16 個であるが、前述により、MIPS 用 Linux には利用する TLB 数を起動時に動的に決定することができるため、R3000 用の TLB 操作のコードをそのまま利用することができた。

5.2 Geysers on FPGA 環境への移植

Geysers on FPGA 環境には、出入口装置としてシリアルポート、主記憶として 16MB の SRAM、電力評価を行う機構が存在する。これらの機能を有するボード依存コードを Linux ソースコードに追加した。

(1) 主記憶

MIPS 用 Linux カーネルは、有効な RAM が存在するアドレスを通知することで、ページやアロケータやカーネル内部で使うメモリ確保を行うアドレスを自動で構成する機能が存在する。Geysers on FPGA 環境では物理アドレス 0x00000000 - 0x01000000 までに有効な RAM が存在するため、それをカーネルに通知するコードを追加した。

(2) シリアルポート

シリアルポートを利用するために Linux 用のデバイスドライバを作成した。

(3) 電力評価機構へのインターフェース

Geysers on FPGA に搭載されている電力評価機構をユーザープログラムから利用するためのシステムコールを新たに追加した。

6 評価

移植によるソースコードの変更行数は、Geysers-0 アーキテクチャ移植のための MIPS 依存コードの変更が 250 行程度、Geysers-0 on FPGA 環境への対応のために追加したコードが 300 行程度、その他若干行となった。

Geysers-0 へ適用される細粒度パワーゲーティング技術により、ALU、シフト、乗算器、除算器、CPO の各回路について、回路が利用されていない場合電力が供給されない状態になる。そのような状態をスリープサイクルと呼ぶ。評価ボードへ移植した Linux を用い、Linux のブートプロセス、カーネルモードでの行列演算プログラム、ユーザーモードでの行列演算プログラムの各々について、電力評価機構のカウンタを用いて実行時の ALU のスリープサイクル数を計測した。計測結果を表 1 に示す。表の sleep 2, sleep 16 等の行はそれぞれ、2 サイクル連続スリープサイクルが起きた回数、16 サイクル連続のスリープサイクルが起きた回数を表す。

表 1: ALU のスリープサイクル数

cycle	boot	kernel app	user app
sleep 1	7471	24009	24096
sleep 2	394755	2710	9404
sleep 6	2043368	206914	238269
sleep 11	48	12	12
sleep 12	2293159	69877	113563
sleep 16	2315963	112479	173452
sleep 40	3290	49	114
sleep total	77759385	3911193	5615198
total	296690259	19902414	26349947
sleep / total	0.262	0.197	0.213

7 考察

ブートプロセスでのスリープサイクルの割合は、他の二つより大きい。これは行列演算が高い頻度で ALU を利用するため、妥当な結果だと考えられる。また、ユーザーモードでの行列計算アプリケーションがカーネルモードのものよりわずかにスリープサイクルの割合が高いのは、ユーザーモードではページフォルト等の割り込み処理によって行列演算が何度か中断されたためのだと考えられる。

8 おわりに

本研究では、省電力 MIPS プロセッサ評価ボードへ Linux の移植を行い、また回路のスリープ状況の計測、電力評価のための考察を行った。今後の課題として、様々なライブラリファイルの移植を行い、より大規模かつ実際に利用されているプログラムの動作確認、評価を行うことがあげられる。

参考文献

[1] 木村一樹, 砂田徹也, 長井智英, 関直臣, 近藤正章, 天野英晴, 宇佐美公良, 中村宏, 並木美太郎. 省電力 MIPS プロセッサコア評価のための計算機システムの FPGA による試作. 情報処理学会「システムソフトウェアとオペレーティング・システム」第 111 回研究報告, OS-111, No.34, pp.1-8 (2009)

[2] 光澤敦, 林和則, 田中浩一. Mach マイクロカーネルシステムの MIPS アーキテクチャへの移植. 情報処理学会「システムソフトウェアとオペレーティング・システム」第 75 回研究報告, OS-75, No.56, pp.19-24 (1997)