

## 省電力 MIPS プロセッサを実現する FPGA における OS の開発と評価

砂田 徹也<sup>†1</sup> 木村 一樹<sup>†1</sup> 長井 智英<sup>†2</sup>  
近藤 正章<sup>†3</sup> 天野 英晴<sup>†4</sup> 宇佐美 公良<sup>†5</sup>  
中村 宏<sup>†2</sup> 並木 美太郎<sup>†1</sup>

本研究は、演算器ごとのパワーゲーティング技術を施した MIPS R3000 ベースの CPU コア Geyser を各ユニットのスリープ頻度を計るパフォーマンスカウンタを付加した FPGA により実現した環境において、軽量な OS を実装し、省電力効果の評価を行った。実装した OS は、マルチタスク管理、仮想メモリ管理、省電力機構を備えている。シミュレーション環境と比べ OS 実行時間は、約 600 倍の高速化を実現し、パワーゲーティングによりスリープした演算器のリーク電力は誤差率約 10% の範囲で推算することが可能であり、効率のよい OS 開発環境とともに、実環境に近い状況でプロセッサの性能評価を行うことができた。

### Prototyping of Operating System for Power-saving MIPS Processor Using FPGA and its Evaluation

TETSUYA SUNATA,<sup>†1</sup> KAZUKI KIMURA,<sup>†1</sup> TOMOHIDE NAGAI,<sup>†2</sup>  
MASAAKI KONDO,<sup>†3</sup> HIDEHARU AMANO,<sup>†4</sup> KIMIYOSHI USAMI,<sup>†5</sup>  
HIROSHI NAKAMURA<sup>†2</sup> and MITARO NAMIKI<sup>†1</sup>

This paper describes prototype of OS for processor core Geyser using FPGA based on MIPS R3000 with a fine grain power gating technique to reduce power consumption and evaluate of power by the performance counter that measures the sleep frequency of each unit. The prototype OS has the multi task management, the virtual memory management, and power saving mechanism by power gating control. As the results running prototype OS and benchmark programs, Geyser on FPGA marked about 600 times faster than the simulation, and estimated power consumption of a leakage power with an error rate of about 10% with prototype OS. We achieved an efficient OS development environment, and achieved the environment that evaluated the performance of the processor in a situation near a real environment.

### 1. はじめに

近年、高性能なシステム LSI は、高度情報化社会を支える基盤として広く利用されているが、その性能と消費電力はトレードオフの関係となる。特に、現代社会では省エネルギーの気運が高まっており、性能向上とともに省電力性能も必要とされる。ここで言う電力とは、大きく分けてダイナミック電力とリーク電力に分けられる。最近のプロセッサの傾向としてプロセスルールの微細化が進んでおり、近年の LSI の消費電力は、リーク電力の占める割合が支配的となってきている。このためプロセッサのリーク電力削減技術の確立は、省電力技術において大きなテーマの一つとなっている。

LSI のリーク電力削減の方法については、コンピュータシステムを構築するハードウェアおよびソフトウェアの各分野で取り組まれている。リーク電力の削減技術としてパワーゲーティングや Dual Vth、基盤バイアスなどが挙げられる。上記は、半導体・回路技術による省電力化手法であるが、一方でハードウェアの技術と連携してシステムソフトウェアによるメモリ管理方法<sup>2)</sup> や、スケジューリング方法<sup>3)</sup>、コンパイラ<sup>4)</sup> に着目した省電力方式も多く研究されている。

筆者らは、リーク電力の削減技術としてパワーゲーティング<sup>5)6)</sup> に着目している。パワーゲーティングとは、プロセッサ内の各ユニットに対して電源供給を断ち、その回路をスリープさせることによってリーク電力を削減する技術である。スリープの技術は、ハードウェアによるところが大きいが、そのスリープを制御するタイミングは、システムソフトウェアやコンパイラが行うことによりさらなる省電力効果を見込むことができると考えられる。

本研究プロジェクト<sup>1)</sup>により、パワーゲーティングを細粒度に適用した MIPS R3000 ベース CPU コア Geyser の研究が進められており、回路技術からシステムソフトウェアまでの各設計階層間での協調による省電力の実現を目標としている。細粒度とは、パワーゲーティ

<sup>†1</sup> 東京農工大学  
Tokyo University of Agriculture and Technology  
<sup>†2</sup> 東京大学  
The University of Tokyo  
<sup>†3</sup> 電気通信大学  
The University of Electro-Communications  
<sup>†4</sup> 慶應義塾大学  
Keio University  
<sup>†5</sup> 芝浦工業大学  
Shibaura Institute of Technology

ングの適用粒度を、CPU コア内の演算器レベルまで細かくし、従来のマルチコアにおけるコアごとの制御よりも、より細かい制御を実現する。本研究プロジェクトでは実際に Geyser を試作しており、省電力効果を測定している。筆者の所属する研究室では、本研究プロジェクトにおいて、システムソフトウェアレベルの省電力技術の研究を行っている。先行研究<sup>10)</sup>において、Geyser のシミュレーション環境における Geyser OS の実装と評価を行っている。また、Geyser シミュレーション環境の FPGA への移行を実現<sup>9)</sup>した。シミュレーション環境では大きな問題のあった膨大なシミュレーション時間を短縮すると同時に I/O の充実を目的に、FPGA を用いることにより OS の開発環境を実現した。また FPGA においてパフォーマンスカウンタを実装し、消費電力を推算し、方式の有用性を評価する環境を実現している。

本稿では、CPU コアとしての Geyser の概要および、上記 FPGA 環境 (以下、Geyser on FPGA) の概要について示し、試作した Geyser on FPGA で動作する Geyser OS について述べる。そして評価として、シミュレーションと Geyser on FPGA における Geyser OS の実行時間や、電力見積もりの精度について述べる。

## 2. Geyser の概要

Geyser は、省電力を目的とした MIPS R3000 アーキテクチャをベースとしたプロセッサコアである。Geyser の実装は、おもに本研究プロジェクトの回路設計およびアーキテクチャグループによって行われた。筆者はシステムソフトウェアグループとして、CP0 や TLB などの OS の動作と深く関係のある部分の設計に携わり、シミュレーション環境において OS の試作と電力評価を行っている。Geyser の省電力技術の特徴として、細粒度パワーゲーティングを適用している。

### 2.1 細粒度パワーゲーティング

Geyser の備えるパワーゲーティングは、従来のパワーゲーティングと比べて非常に粒度の細かい制御を実現している。パワーゲーティングの対象は、Geyser が備える演算器である ALU, SHIFT, MULT, DIV および CPU と OS とのインターフェースの役割を持つ CP0 のそれぞれである。図 1 に、Geyser の構成と細粒度パワーゲーティングの構成図を示す。各演算器に対して、動作状態を切り替えるスイッチを挿入し、そのスイッチをスリープコントローラによって切り替えることで、パワーゲーティング制御を行っている。

#### 2.1.1 OS による細粒度パワーゲーティング制御インターフェース

細粒度パワーゲーティングを制御するために、OS に与えられているインターフェースとして、CP0 レジスタにパワーゲーティング制御用の PGStatus レジスタを有している。

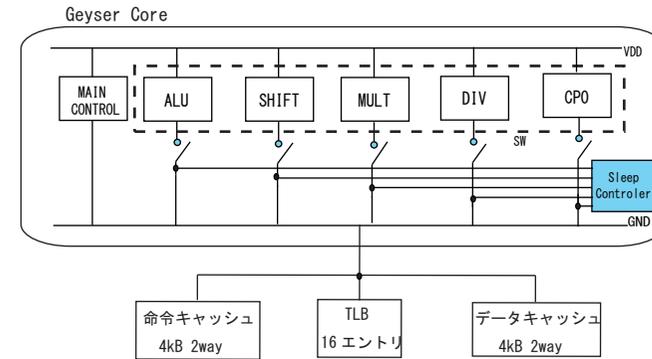


図 1 Geyser の構成と細粒度パワーゲーティング

PGStatus レジスタは、OS によってパワーゲーティングの対象となる各演算器に対して通常とは異なる以下の三つのスリープポリシーを適用する際に用いる。

- 動的にパワーゲーティング (通常)
- キャッシュミス時のみスリープ
- 常に動作 (パワーゲーティングを行わない)

図 2 に、PGStatus レジスタを示す。動的にパワーゲーティングというポリシーは、実行される各命令ごとに必要とする演算器を判別し、スリープ制御を行う。キャッシュミス時のみスリープというポリシーは、キャッシュミスによって約 20 サイクルの遅延が発生するため、スリープ期間の推測が可能であるポリシーとしてスリープ制御を行う。

### 2.2 Geyser の電力評価

従来の Geyser は、Verilog によるシミュレーションによって実現されていた。その際筆者

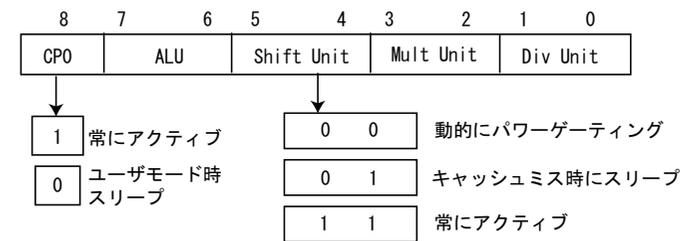


図 2 PGStatus レジスタ

は、本研究室において開発されてきた組込み向け OS「開闢 (かいもん)」<sup>8)</sup> を Geysler に向けて移植した Geysler OS を開発している<sup>10)</sup>。Geysler OS は、タスク管理、例外・割り込み管理、システムコールなどの機能を備えている。Geysler OS を用いていくつかのベンチマークプログラムをマルチタスクで動作させた結果、細粒度パワーゲーティングにより、総消費エネルギーで約半減、リーク電力においては、使用頻度の高い ALU や SHIFT では大きな削減効果を得られないものの、使用頻度の低い DIV のような演算器では 40%～最大 85% の削減を実現している。以下図 3、図 4 にシミュレーション時における Geysler の省電力効果について示す。

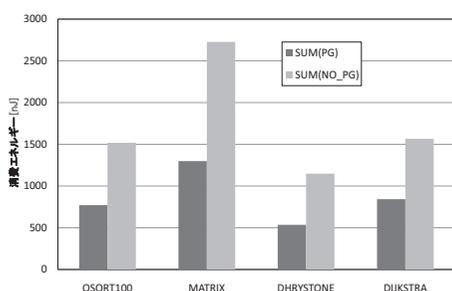


図 3 演算器全体での総エネルギー (RTL/ゲートレベルシミュレーション)

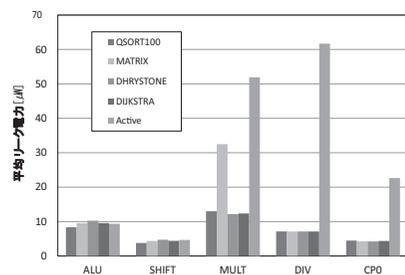


図 4 各演算器のリーク電力削減効果 (RTL/ゲートレベルシミュレーション)

### 2.3 シミュレーション環境の問題点

上記評価を取る際には、Verilog によって RTL/ゲートレベルの各シミュレーションを行った。しかし、シミュレーション環境には以下に示す問題点がある。

#### (1) シミュレーション時間

シミュレーションに要した時間は、Geysler の目標周波数である 200MHz から予想される実時間と比べて、RTL シミュレーションで数万倍、ゲートレベルシミュレーションにおいて数千万倍に達した。これは OS やプログラムの規模が大きくなればその開発において非常に問題となってくる。

#### (2) メモリと I/O

シミュレーションではメモリはすべて RTL 記述によるレジスタアレイで実現される。これはメモリのアクセスコストが極めて小さく、キャッシュメモリなど本来ならばより

高速であるはずの記憶領域と差別化されないため、OS 開発においてはメモリ管理や評価において実計算機的环境と大きく異なる環境となる。また I/O もないため、ユーザインタフェースや実行時におけるデータの送受信機能などの実現も困難である。

## 3. Geysler on FPGA の概要

シミュレーション環境において発生した各問題を解決するために、FPGA を用いて Geysler の環境を実現する。FPGA 上に Geysler のコアを移植し、各種のメモリコントローラや I/O モジュールを接続することにより、Geysler の環境をシミュレーションよりもより高速に、より現実的な実メモリ・実 I/O を用いて実現することができる。

また FPGA において Geysler の特徴である省電力効果を測定するために、パワーゲーティングにより演算器がスリープする状況を観測し、それを記録することで消費電力の削減量を見積もるパフォーマンスカウンタを備える。

### 3.1 Geysler on FPGA の全体構成

図 5 に、今回先行研究<sup>9)</sup> において設計された Geysler on FPGA の全体構成を示す。Geysler をコアとして、各種メモリや I/O を接続する。FPGA の詳細な設計については、先行研究を参考いただきたい。

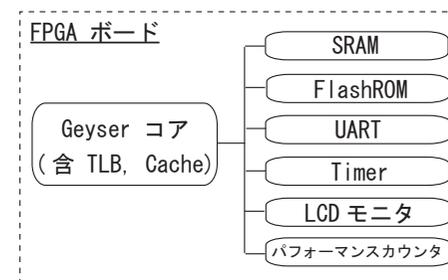


図 5 Geysler on FPGA の全体構成

#### 3.1.1 パフォーマンスカウンタ

Geysler on FPGA では Geysler コアの実現やメモリ、I/O の充実だけでなく、パフォーマンスカウンタを備える。これは、後述する電力評価の際に必要な、各演算器のスリープ状況をはじめ、既存の CPU に備えられるようなカウンタを備える。現行のパフォーマンスカウンタは以下の値取得を可能としている。

- サイクルカウンタ
- 各演算器のスリープ頻度情報

各演算器のスリープ頻度は、「 $N$  サイクルのスリープが  $M_N$  回」という情報を取得する。

### 3.2 Geysers on FPGA の実現環境

Geysers on FPGA の実現には、Xilinx 社の Virtex5 を搭載した FPGA ボード ML501 を用いた。メモリデバイスには、ML501 オンボードの SRAM(1MB)、FlashROM(32MB) のそれぞれを RAM、ROM として用いる。入出力装置として、RS-232C およびキャラクタ LCD を備える。動作クロックは、ボード上のクロックリソースの制約から、16.5MHz である。

本研究では、この FPGA ボードを用いて以下に示すプログラムのブートロード環境および Geysers OS on FPGA を開発している。電力評価についても FPGA ボードに実装したパフォーマンスカウンタをもとに取得する。

### 3.3 ブートローダ環境

従来の Geysers on FPGA では、FlashROM に予めプログラムを記憶させておくことにより、FlashROM からブートを行うことが可能であった。そこで、ソフトウェア開発における柔軟性を高めるために、FPGA のメモリ内にプログラムを転送する方法として、今回 XMODEM/SUM をプロトコルとしたブートローダを作成した。メモリ内の任意の位置に、PC からバイナリ形式のプログラムを XMODEM/SUM プロトコルを用いて転送が可能である。

## 4. Geysers OS on FPGA の試作の概要と全体構成

本項では、上記 Geysers on FPGA 環境で動作する Geysers OS の試作について述べる。最初にブートロード環境について示し、Geysers OS が持つ機能を示す。

### 4.1 Geysers OS on FPGA の設計方針

Geysers OS は、筆者の研究室で開発されてきた組み込み向け OS「開聞」をもとに試作した。「開聞」<sup>8)</sup> は、軽量なタスク管理、例外管理などを備えた OS となっている。「開聞」は過去に MIPS プロセッサ (Vr4300) へと移植が行われており、Geysers がベースとしている R3000 とは若干異なるものの MIPS アーキテクチャである点は変わらない。また「開聞」はリアルタイム OS としての動作を行うことも可能であり、今後の発展的な部分として考えている。そこで筆者は、Geysers の評価を行うための試作 OS として「開聞」を参考とした。

### 4.2 Geysers OS on FPGA の基本構成

Geysers OS on FPGA の基本構成を図 6 に示す。Geysers OS on FPGA は、以下に示す

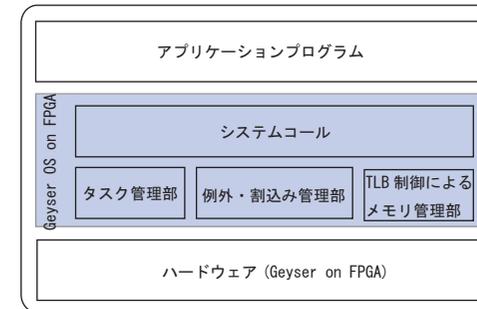


図 6 Geysers OS on FPGA の基本構成

機能を実現する。ただし、OS としての実用性よりも OS 実行時のプロセッサ Geysers を評価することが Geysers OS on FPGA の主な目的である。

- マルチタスク OS
- 仮想メモリ管理
- 省電力制御

Geysers OS on FPGA は、上記各機能を実現するために、

- 例外・割り込み管理部
- システムコール
- タスク管理部
- TLB 制御によるメモリ管理部

のそれぞれの管理部を有している。

## 5. Geysers OS on FPGA における省電力制御

Geysers OS on FPGA の特徴として、マルチタスクや仮想メモリを管理するとともに、Geysers の特徴である細粒度パワーゲーティングを制御するための省電力制御を行う点にある。以下に Geysers OS on FPGA における省電力制御手法について述べる。

### 5.1 細粒度パワーゲーティングの問題点とブレークイーブンポイントの定義

パワーゲーティングによるスリープ、動作状態の遷移には一定の電力が必要となり、またスリープ状態に切り替わっても瞬時にリーク電力が下がるわけではなく、徐々に下がっていくタイムラグがある。ここで、常に動作状態であった場合に消費する電力よりも状態遷移によるオーバーヘッドを考慮してそれでもスリープ状態へ移行した方が、消費電力が削減できる

点 (サイクル数) をブレイクイーブンポイント (BEP: Break Even Point) と呼ぶ。BEP は、温度と演算器の特性によって異なり、ある演算器においてスリープ期間を  $t$ 、温度を  $T$  とした時、スリープ遷移におけるエネルギーオーバーヘッドを  $E_{sleepOH}(T)$ 、スリープから動作状態に遷移する際のエネルギーオーバーヘッドを  $E_{wakeupOH}(T)$ 、スリープすることによって削減されるエネルギー  $E_{sleep}(t, T)$  とすると、スリープと動作状態の遷移において、

$$E_{sleep}(t, T) < E_{sleepOH}(T) + E_{wakeupOH}(T)$$

となる時の、スリープ期間  $t$  を  $BEP(t_{bep})$  と定義する。  $t_{bep}$  を越えるスリープ期間を得られないときは、スリープしないと比べて消費エネルギーは増加する。

### 5.2 省電力機構の概要

Geyser OS on FPGA では、パワーゲーティングの動作状態遷移に伴う BEP に満たないスリープが増えることによる消費エネルギーの増加を低減するために、BEP の値と、Geyser on FPGA が提供するパフォーマンスカウンタを用いて、実行時に省電力を行う機構を備える。

BEP は、本研究プロジェクトの先行研究<sup>7)</sup> によって既知の値であるとする。Geyser on FPGA のパフォーマンスカウンタは、第 3.1.1 項に示すように、各演算器ごとのスリープ頻度を取得可能である。BEP を越えないスリープサイクル数を低減させる目的から、全スリープサイクルにおいて、BEP を越えないスリープ頻度の割合がある閾値  $BEP_{threshold}$  を越える場合に、スリープポリシーを変更する。

全スリープサイクルを  $t_{sleepAll}$ 、 $i$  サイクルのスリープ期間を  $t_i$  とすると、

$$T = \{t_i | i < bep\}$$

$$\frac{\sum_{t_i \in T} t_i}{t_{sleepAll}} < BEP_{threshold} \rightarrow changeSleepPolicy$$

となる。上記の機構を実現するために、システムコール、タスク管理部と連携して省電力を実現する。

### 5.3 Geyser OS on FPGA のシステムコール

Geyser OS on FPGA の各機能および特徴である省電力機構を実現するために、システムコールとして、表 1 を実装した。

ユーザタスクは Geyser on FPGA の特徴でもあるパフォーマンスカウンタをシステムコールを介して制御可能である。パフォーマンスカウンタの制御として、

- カウンタ値の初期化
- カウンタの測定開始/停止/出力

表 1 システムコールの一覧

I/O 関連		タスク管理関連	
システムコール名	処理	システムコール名	処理
sys_printf	UART へ出力	Init_Task	初期タスクの生成
sys_gets	UART から入力	Create_Task	タスクの生成
Timer_Init	タイマの初期化	Delte_Task	タスクの消去
Timer_Set	タイマのセット	Suspen_Task	タスクを実行可能状態に戻す
Timer_Stop	タイマの停止	Resume_Task	タスクを実行可能状態に遷移
		Yield	強制的にスケジューラを呼び出す

パフォーマンスカウンタ関連	
システムコール名	処理
Start_PerCntr	モニタリングの開始
Stop_PerCntr	モニタリングの停止
Print_PefCntr	モニタリングの結果を出力
Read_PefCntr	モニタリングの結果を読み出す
Init_PefCntr	モニタリング結果の初期化
Set_PGStatus	PGStatus 制御インタフェース

を実現している。また実際に、パワーゲーティングのスリープポリシーを変更する際は、Set\_PGStatus システムコールを使用する。Set\_PGStatus システムコールは制御する演算器 unit と、スリープポリシー policy を引数に与える次の仕様となっている。

Set\_PGStatus(unit, policy)

### 5.4 タスク管理部

タスク管理部は、マルチタスクにおけるタスクの生成、消去およびスケジューリングを実現する。第 5.2 節で示した省電力機構を動作させるタイミングとして、スケジューラにおけるタスクスイッチ時を考える。

- (1) スケジューラ開始
- (2) パフォーマンスカウンタからスリープ頻度を取得
- (3) if: BEP を越えないスリープ頻度が、ある閾値より多い → スリープポリシー変更
- (4) タスクスイッチ
- (5) スケジューラ終了

ただし、本稿で試作した省電力機構は、ALU を使う傾向が強い、MULT を使う傾向が強いといった、タスクごとの演算器の使用特性を考慮していないため、タスクスイッチで演算器の使用特性が全く異なるタスクを実行するときは、本省電力機構の適用は難しくなる。解決策として、タスクごとの演算器使用特性を実行時に取得し、演算器の使用特性の類似度によっ

てタスクをグループ分けしてそれぞれで制御することで、省電力効果を高めることができると考えられる。

### 5.5 TLB 制御によるメモリ管理部

Geysler on FPGA では、TLB を備えており、ユーザプログラムは TLB を介して物理メモリにマッピングされる。Geysler OS on FPGA は、TLB を制御し、仮想メモリ管理を行う。図 7 に、Geysler on FPGA における OS のメモリ管理の概要図について示す。Geysler on

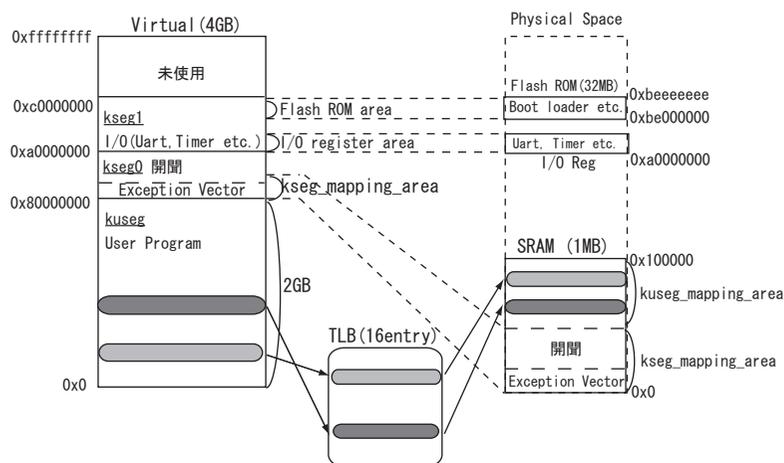


図 7 TLB 制御によるメモリ管理

FPGA は、SRAM、I/O Register、FlashROM を備えており、この中で実際に Geysler OS が管理するのは、Read/Write 可能な SRAM の領域である。SRAM の領域をカーネル領域とユーザ領域に分ける。Geysler OS 自身はカーネル領域に配置され、OS によってユーザ領域の空き領域を管理する。ユーザプログラムの実行時に TLB ミスが発生した際に、Geysler OS は、管理する空き領域から 1 ページをユーザプログラムに割り当て、処理をユーザプログラムに戻すことで、TLB ミスハンドリングを行う。

## 6. 電力評価方法

本項では、Geysler on FPGA による電力評価方法について述べる。最初に、FPGA で計測することのできる電力について述べ、実際の評価方法について示す。

### 6.1 Geysler on FPGA で計測可能な電力

図 8 に、細粒度パワーゲーティングにおいて評価対象となる電力を示し、表 9 に、FPGA によって現在評価可能な電力を示す。シミュレーションでは、全領域にわたって電力評価が

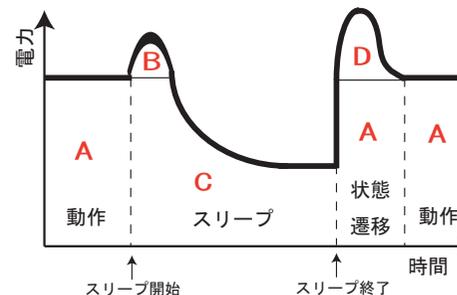


図 8 実験

電力内訳		一週 (領域)	VDECシミュ レーション	FPGA	電力評価に 必要なデータ	
総電力	ダイナミック 電力	アクティブ時	A, B	○	×	スイッチング確率 Power Compiler
		スリープ時 (=0)	—	—	—	—
	リーク電力	アクティブ時	A, B	○	×	スイッチング確率 Power Compiler
		スリープ時	C, D	○	○	スリープ頻度情報 電力解析スクリプト

図 9 実験結果

可能であった。しかし、FPGA においてアクティブ時の電力を求めるトランジスタのスイッチング確率を取得することはできないため、FPGA において評価可能な電力は、演算器の使用頻度で評価することが可能であるスリープ時のリーク電力 (領域 C, D) となる。

### 6.2 スリープ時リーク電力の取得方法

スリープ時リーク電力を求めるためには、細粒度パワーゲーティングの対象である各演算器に対して、動作/スリープの頻度情報を取得する必要がある。これは、Geysler on FPGA のパフォーマンスカウンタによって取得可能である。

$i$  サイクルスリープ時の消費電力  $P_i$  は Synopsys 社の HSIM というツールで取得可能であり、本研究プロジェクトにおいて取得は完了している。また、1000 サイクル以上のスリープ  $P_{i>1000}$  については、1000 サイクルスリープの電力と同等とみなし、 $P_{i>1000} = P_{1000}$  とする。この時、全スリープ期間を  $T_{all}$ 、 $T_i$  のスリープ期間が現れた回数  $n_i$  は、FPGA のパフォーマンスカウンタによって取得する。これらの情報より、スリープ期間の出現頻度は、

$$R_i = T_i * n_i / T_{all}$$

となる。この  $R_i$  から、スリープ時のリーク電力の合計  $P_{sleep}$  は、

$$P_{sleep} = \sum_i (P_i \times R_i)$$

となる。

上記計算式に基づいて、各演算器の使用頻度からスリープ時のリーク電力評価を行う。

7. 評価

本項では、評価環境について述べ、Geysler OS on FPGA 動作時の実行時間および電力評価を行う。また Geysler OS on FPGA によって細粒度パワーゲーティングと OS とのインタフェースである PGStatus レジスタを制御し、スリープポリシーを変更した際の電力評価について述べる。

7.1 評価環境

評価のためのハードウェア環境は、本稿で Geysler on FPGA の実現環境として述べた ML501 である。ソフトウェアは、試作した Geysler OS on FPGA を用いて、クイックソート 100 要素、行列乗算 100 × 100 を動作させた際の評価を行う。パフォーマンスカウンタにおけるモニタ対象は、今回は ALU とする。

パフォーマンスカウンタのカウント期間は実際に XMODEM ロードによってカーネルプログラムをロードした時から、カーネルの実行を終了する期間である。Geysler OS on FPGA によるパフォーマンスカウンタの API を用いて、取得を行っている。

Geysler on FPGA との比較対象として、Cadence 社の NC-Verilog を用いて Geysler をシミュレーションする環境を挙げる。

以下に、FPGA の電源投入から実行終了までの流れを以下に示す。

- (1) FPGA に電源投入
- (2) FlashROM から、Geysler のコンフィギュレーションデータをロード
- (3) FlashROM から、XMODEM ブートローダを SRAM 内にロード
- (4) XMODEM ブートローダを用いて、Geysler OS on FPGA およびユーザプログラムをロード
- (5) Geysler OS on FPGA のセットアップ
- (6) ユーザモードを遷移し、ユーザプログラムの実行
- (7) 実行終了

この流れの中で、パフォーマンスカウンタを制御するシステムコールを用いて各評価データを取得する。

7.2 実行速度の比較

Geysler on FPGA と、Verilog による RTL シミュレーション環境における実行速度の比較を行う。評価結果を表 2 に示す。

表 2 各ベンチマークの実行時間およびサイクル数

QSORT	サイクル数	時間 [sec]	MATRIX	サイクル数	時間 [sec]
FPGA	779,141	0.047	FPGA	1,684,323	0.102
VDEC シミュ	169,223	13.848	VDEC シミュ	1,036,755	65.342

実行時間において、各ベンチマークで約 300 倍、約 640 倍の高速化を実現しており、FPGA の利点が強く表れている。OS の開発・実行環境として FPGA の優位性を示す結果となった。

7.3 FPGA 環境とシミュレーション環境における電力評価

ここでは、FPGA 環境とシミュレーション環境における電力評価の結果を示す。評価のための PGStatus によるスリープポリシーとして、次の二つのスリープポリシーの評価を行った。なお、原稿執筆時点において、Geysler on FPGA のキャッシュ動作はデバッグ中となっており、デバッグ終了後、Geysler OS on FPGA において設計した省電力機構に基づくスリープポリシーの評価を行う予定である。

(1) 実行開始から、スリープポリシー：動的にパワーゲーティングで固定

まずスリープポリシーを動的にパワーゲーティングとした時のスリープ時電力の評価結果を図 10、図 11 に示す。

各ベンチマーク共に、Geysler on FPGA とシミュレーションで誤差は最大 10%程度に収まっており、前述の実行速度の大幅な向上を実現している環境において、充分精度が高いといえる。

(2) 実行開始から、スリープポリシー：キャッシュミス時のみスリープで固定

図 12、図 13 に、スリープポリシーを動的に PG およびキャッシュミス時のみスリー

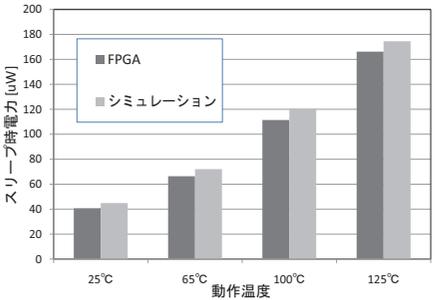


図 10 OS + Qsort × 2 の電力評価結果 (動的に PG)

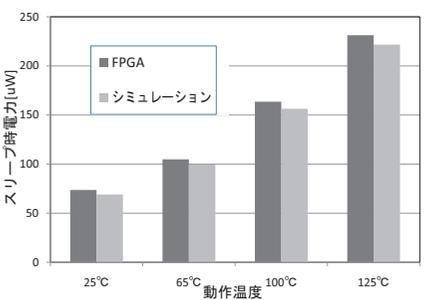


図 11 OS + Matrix × 2 の電力評価結果 (動的に PG)

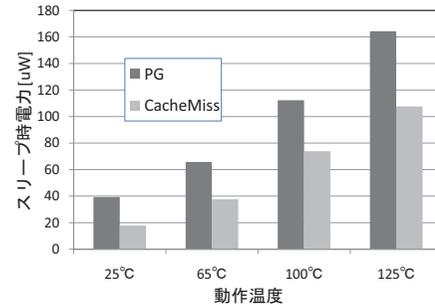
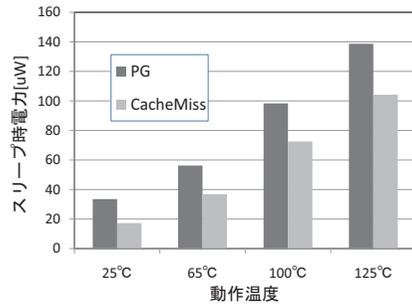


図 12 OS + Qsort × 2 : 動的に PG vs キャッシュミス時のみスリープ  
図 13 OS + Matrix × 2 : 動的に PG vs キャッシュミス時のみスリープ

ブとした時の、スリープ時電力の評価結果を示す。なお、本評価はシミュレーション環境において取得したものである。キャッシュミス時のみスリープとした際に、スリープ時電力が約 50% 削減できていることがわかる。しかし、スリープ時電力は削減できるが、その分そのユニットがアクティブである期間が長くなるので、アクティブ時の電力が大きくなるので、注意が必要である。

## 8. おわりに

本稿では、省電力 MIPS プロセッサである Geysler を実現する Geysler on FPGA において動作するブートローダおよび Geysler OS on FPGA の試作を行い、OS 実行時の評価データを取得した。Geysler on FPGA において、Geysler OS を用いることによってマルチタスクおよび仮想メモリ管理を実現した。また OS によるパフォーマンスカウンタの制御と PGStatus レジスタの制御も実現し、パフォーマンスカウンタと PGStatus の制御およびスケジューラとの連携により省電力を実現する機構の設計を行った。評価として、実行速度は Geysler on FPGA は、シミュレーションと比較して 600 倍程度の高速度を実現し、OS の開発・実行環境において FPGA が有効である点を示した。また、電力評価においては、シミュレーションとの誤差が最大で 10%にも満たない程度となり、FPGA でスリープ時のリーク電力を見積もる有効性を示した。

今後の課題として、まず Geysler 実機による評価データを取得し、シミュレーションおよび FPGA における電力評価が有効かどうかを示す。その上で、OS 開発・実行環境に有利な FPGA においてアクティブ時の電力を見積もる環境を構築する方式を設計し、FPGA にお

ける電力評価において、今回設計した OS による省電力機構の試作、評価を行っていく。その過程で、独自 OS である Geysler OS on FPGA から、Linux など一般によく知られる OS の Geysler on FPGA への移植を行う。

**謝辞** 本研究は東京大学大規模集積システム設計教育研究センター (VDEC) を通し、株式会社半導体理工学研究センター、富士通株式会社、松下電器産業株式会社、NEC エレクトロニクス株式会社、株式会社ルネサステクノロジ、株式会社東芝の協力で行われたものである。

本研究は、科学技術振興機構「JST」の戦略的創造研究推進事業「CRSET」における研究領域「情報システムの超低電力化を目指した技術革新と統合化技術」の研究課題「革新的電源制御による次世代超低電力高性能システム LSI の研究」によるものである。

## 参考文献

- 1) 中村宏 他: 革新的電源制御による超低消費電力高性能システム LSI の構想, 情報処理学会研究報告 ARC-173, pp.79-84 (2007).
- 2) Pratap Ramamurthy et al.: Performance-directed Energy Management using BOS, ACM SIGOPS Operating Systems Review, Vol.41, pp.66-77 (2007).
- 3) Andreas Merkel et al.: Task Activity Vectors: A New Metric for Temperature-Aware Scheduling, Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008, pp.1-12 (2008).
- 4) Chinnakrishnan S. Ballapuram et al.: Exploiting Access Semantics and Program Behavior to Reduce Snoop Power in Chip Multiprocessors, Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-XIII), pp.60 - 69 (2008)
- 5) Zhigang Hu et al.: Microarchitectural Techniques for Power Gating of Execution Units, Proceedings of the 2004 International Symposium on Low Power Electronics and Design (ISLPED 04). pp32-37 (2004).
- 6) Y.Kanno : Hierarchical Power Distribution with 20 Power Domains in 90-nm Low-Power Multi CPU Processor, ISSCC2006, pp.540-541 (2006).
- 7) 白井利明 他: ランタイムパワーゲーティングを適用した MIPS R3000 プロセッサの実装設計と評価, 電子情報通信学会技術研究報告 VLD2007-112, pp.43-48, (2008)
- 8) 堀口 努, 萱嶋 志門, 並木 美太郎: 組込み用 OS『開聞』の MIPS プロセッサへの移植, 情報処理学会研究報告 OS-87, pp.57-64 (2001).
- 9) 木村 一樹, 並木 美太郎 他: 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, 情報処理学会研究報告 OS-111 (2009).
- 10) 砂田 徹也, 並木 美太郎 他: 省電力 MIPS プロセッサにおける OS の試作とシミュレーションによる電力評価, 情報処理学会研究報告 OS-108, pp.163-170 (2008).