Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

細粒度パワーゲーティングを制御するOSの資源管理方式

砂田徹也[†1] 木村一樹[†1] 近藤正章[†3]
天野英晴[†4] 宇佐美公良[†5]
中村宏[†2] 並木美太郎[†1]

近年, 社会において省エネルギーの要求が高まっており, 実行性能とともに省電力も必要とされる. 本研究は, CPU コア Geyser の細粒度 PG(Power Gating) を OS により制御することで, ハードウェア単体の制御と比べさらなる省電力を実現する. 本 OS は, マルチタスク環境および仮想メモリ管理のほかに省電力機構を備えており, 細粒度 PG の電力的特徴である電力損益分岐点を考慮して, PG のスリープポリシーを OS が変更することにより, 細粒度 PG による電力ロスを削減することで省電力を実現する. 本 OS の省電力機構により, ALU のリーク消費電力を 5%〜20%削減し, その際の OS 制御による性能オーバーヘッドは最大 3%未満となるため, リーク消費エネルギーについても同等の省エネルギー化を実現した.

Resource Management of Operating System for Fine Grain Power Gating Control

Tetsuya Sunata,[†1] Kazuki Kimura,[†1] Masaaki Kondo,[†3]
Hideharu Amano,[†4] Kimiyoshi Usami,[†5] Hiroshi Nakamura[†2]
and Mitaro Namiki [†1]

The power saving is needed with the execution performance in recent years. This paper describes a technique that controls fine grain power gating of Geyser CPU with an OS support for achieving further power savings compared to the hardware based control. The Geyser OS has task management, memory management, and power saving mechanisms with power gating control. By considering the break even point that is an electric power feature of power gating, the power saving mechanism changes the sleep policy of fine grain power gating for reducing the power. The leakage power consumption of ALU has been reduced by 5% to 20% with the power saving mechanism of the Geyser OS. The overhead by the OS becomes less than 3% or less.

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化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.

省電力のための技法としては, ハードウェア, ソフトウェアのそれぞれの視点で様々な研究が行われている. 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. ダイナミック電力とは, 回路が動作する際のトランジスタのスイッチングによって消費される電力であり, リーク電力とは, 回路の動作, 非動作に関係なく回路に電圧がかかることにより発生する電力である. 近年の省電力技術の研究傾向として, ハードウェア, ソフトウェアともにダイナミック電力を削減する技術の研究が多くなされていた. 筆者の研究室でも, ダイナミック電力を削減する代表的な技術である DVFS(Dynamic Voltage and Frequency Scaling) 技術を OS スケジューラによって制御することで, ダイナミック電力の削減を実現している. 一方でリーク電力を削減する技術については, Dual Vth, 基板バイアス, パワーゲーティングなどの技術があるが, これらはハードウェアを中心に展開されており, そのハードウェアの機能を OS をはじめソフトウェア側から制御することで, 省電力を実現した例は少ない. 特に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ため, リーク電力削減のための技術研究が重要となっている.

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技術[6] 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

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

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

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．従来のパワーゲーティングは, 電力供給を停止可能な機能ユニットに対しハードウェアが自律してアイドル期間を予測することで発見し, 実際にパワーゲーティングを行っている．これは, ハードウェアが単体で処理を行うためソフトウェア側の変更が必要なく, またそのソフトウェア処理のオーバーヘッドが発生しないなどの利点が挙げられる．一方で, ソフトウェア側で得られる実行時の情報を基にしてパワーゲーティングを行うことで更なる省電力の可能性があると考えられる．

本研究プロジェクト[1)]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, 空間的粒度としてパワーゲーティングの適用粒度を, 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．また時間的粒度として, パイプラインを流れる各命令ごとにパワーゲーティングを行う．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 アーキテクチャの概要および, その Geyser アーキテクチャ上で動作する OS について解説する．そして, Geyser の省電力特徴である細粒度パワーゲーティングにおける電力的特徴である電力損益分岐点について述べ, その特徴を用いて Geyser OS が備える省電力機構の設計について述べる．評価では, Geyser アーキテクチャのシミュレーション環境上で, Geyser OS の省電力機構の効果について評価する．

2. Geyser アーキテクチャの概要

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．また Geyser アーキテクチャは OS を実際に動作させることをひとつの目標としており, そのための特権レジスタ CP0 や, アドレス変換機構 TLB といったシステムモードを有している．Geyser の全体構成を図 1 に示す． 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．

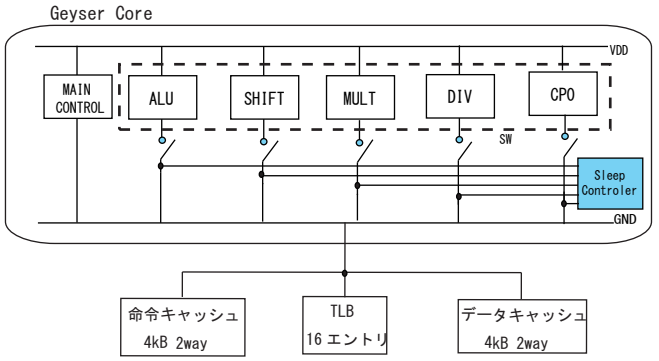

図 1 Geyser アーキテクチャの全体構成

2.1 細粒度パワーゲーティングとソフトウェアインタフェース

Geyser アーキテクチャは省電力のための機能として, リーク電力の削減技術として注目されているパワーゲーティング技術を有する．特に Geyser は, このパワーゲーティングを細粒度に適用した細粒度パワーゲーティングによりリーク電力削減を実現する．細粒度は, 以下の二つの粒度を言う．

- 時間的粒度: パイプラインを流れる各命令ごとに
- 空間的粒度: 演算器 (ALU, SHIFT, MULT, DIV) ごとに

またこれらの粒度をソフトウェア側から制御するために, Geyser アーキテクチャの特権レジスタである PGStatus レジスタを有する．PGStatus レジスタは図 2 として示される．PGStatus レジスタは, パワーゲーティング対象である各演算器ごとに, 以下に示す三つのスリープポリシーを定めることができる．

- 動的にパワーゲーティング (通常ポリシー)
- キャッシュミス時にのみスリープ
- 常にアクティブ (パワーゲーティングなし)

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

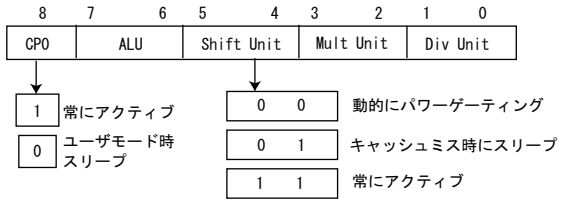

図 2 PGStatus レジスタ

OS は特権レジスタである PGStatus レジスタを操作することで, Geyser の細粒度パワーゲーティングのスリープポリシーを制御することが可能である.

2.2 Geyser アーキテクチャの実装

ここまで示してきた Geyser アーキテクチャは, 表 1 に示されるように三つの環境で実現され, それぞれの環境でそれぞれに特徴を持つ. 表に示されるように, 各環境ごとに一長一短であるが, ハードウェア構成が決定されれば FPGA や実機を使うのが, 実行速度や実ハードウェアを用いているため良いといえる. 逆に Geyser アーキテクチャなどハードウェアそのものに変更を加える可能性が高い場合は, 開発時に柔軟に構成が変更可能な RTL/Gate Sim 環境を用いるのが良い. また電力評価についても, 現状 FPGA において取得可能な情報のみでは電力評価は難しいが, その他の環境と組み合わせることで可能となると考えられる.

表 1 Geyser アーキテクチャ実装環境の特徴

	実行速度	実メモリ	実 I/O	構成変更の容易さ	電力評価環境
Geyser 実機	○	○	○	×	○
RTL/Gate Sim	×	×	×	○	○ (但し Simlation)
FPGA	△	○	○	△	△

3. GeyserOS の設計

3.1 Geyser OS の基本設計と目的

Geyser OS とは, これまで示してきた Geyser アーキテクチャで動作するオペレーティングシステムの名称であり, 現在 RTL/ゲートシミュレーションおよび Geyser on FPGA 上で動作している.

Geyser OS は, Geyser のシステムモードで操作される CP0 や TLB を制御する. ただし, OS としての実用性よりも OS 実行時のプロセッサ Geyser を評価することが Geyser OS の主な目的である. そして, この Geyser OS に省電力機構を組み込むことにより, OS による細粒度パワーゲーティング制御方式を実現する. 本稿で提案する省電力機構は, スケジューラ部で動作し, またパフォーマンスカウンタへのアクセスが必要となるため, GeyserOS はタスク管理部および各 I/O へのアクセス手段を持つ必要がある. Geyser OS はタスク管理, メモリ管理といった基本機能を提供し, それらを基に提案する省電力機構を実現することで, 省電力を実現することを目的とする.

3.2 Geyser OS の基本構成

Geyser OS の基本構成を図 3 に示す. Geyser OS は, 以下に示す機能を実現することで, OS として基本的な機能を提供する.

- マルチタスク OS
- 仮想メモリ管理
- 省電力制御

Geyser OS は, 上記各機能を実現するために,

- タスク管理部
- 例外・割込み管理部
- システムコール
- 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. 特に GeyserOS の特徴である省電力制御は, タスク管理部におけるタスクスケジューラ部において, タイムスライスごとに BEP ミスを計算することで省電力を実現するため, タスク管理部は特に GeyserOS に特化した構成となる. 省電力

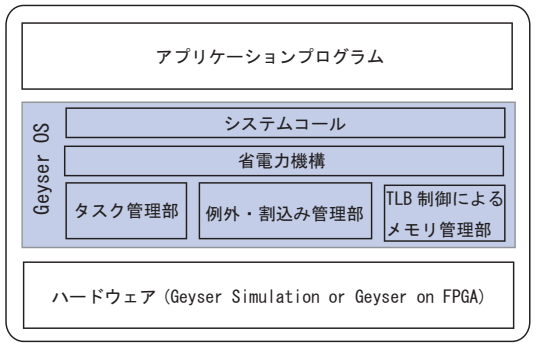

図 3 Geyser OS の基本構成

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

機構の詳細は, 後述する.

Geyser OS は CP0 のパワーゲーティング効果の検証を行うためユーザモードへの遷移およびユーザモードでの正常動作が可能である OS として開発を行う. また TLB ミスの際にも, 正しく例外処理を行える実装とする. 割込みは, FPGA によって実装したタイマを用いて, Geyser on FPGA において Geyser に割込みをかけ, それを Geyser OS によってハンドリングする.

4. 細粒度パワーゲーティングの電力的特徴

ここで, Geyser アーキテクチャの省電力的特徴である細粒度パワーゲーティングの電力的特徴について述べ, 後述する省電力機構の基となる BEP(Break Even Point) ミス率について定義する.

4.1 細粒度パワーゲーティングによる電力遷移

パワーゲーティングを行うことにより Geyser アーキテクチャは省電力を実現する. パワーゲーティングはある機能ユニットに対して, 電源供給を ON/OFF する技術であるので, パワーゲーティング実行時の電力遷移は図 4 に示されるようになる. 各領域は, 以下に示す電力となる.

- 領域 A: アクティブリーク電力

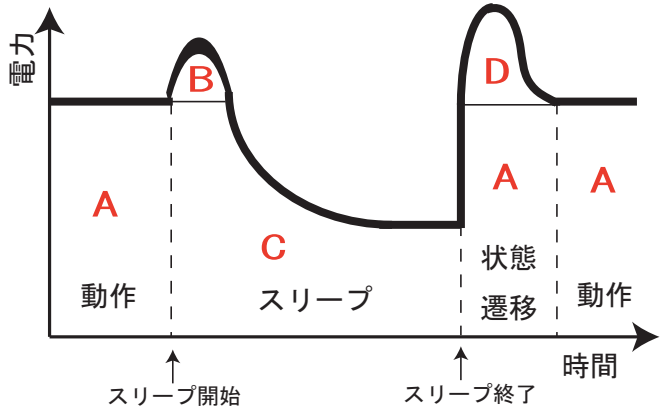

図 4 細粒度パワーゲーティングにおける電力遷移

表 2 細粒度パワーゲーティングのブレークイーブンポイント [単位 : サイクル]

	25 ℃	65 ℃	100 ℃	125 ℃
ALU	124	38	18	12
SHIFT	160	50	22	14
MULT	118	44	44	34
DIV	58	14	6	2

- 領域 B: アクティブ→スリープ遷移時のオーバーヘッドリーク電力
- 領域 C: スリープ時リーク電力 (過渡現象による曲線)
- 領域 D: スリープ→アクティブ遷移時のオーバーヘッドリーク電力

ここで, 細粒度パワーゲーティングにおける電力の特徴として, 領域 B/D に示されるオーバーヘッドリーク電力である. オーバーヘッドに対してここで電力損益分岐点 (BEP: Break Even Point) を定義する.

4.2 BEP および BEP ミス率の定義

前述したように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) を電力損益分岐点と呼ぶ. 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 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, スリープしないときと比べて消費エネルギーは増加する. また BEP は, 演算器ごとに異なり, さらに演算器の動作温度においても異なる. 表 2 に, 演算器と動作温度の違いによる BEP の変化を示す. ここで BEP ミスというのは, 細粒度パワーゲーティングによって発生するオーバーヘッドがスリープによって得られる電力削減を越えてしまった場合, つまり細粒度パワーゲーティングを行うことによって逆に電力ロスしてしまう状況として定義する. 図 5 に, BEP を満たす状況および BEP ミスする状況について示す. この図では, 上の図では, WakeUp のタイミングが BEP を越えたところで起きているため, 結果電力削減

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

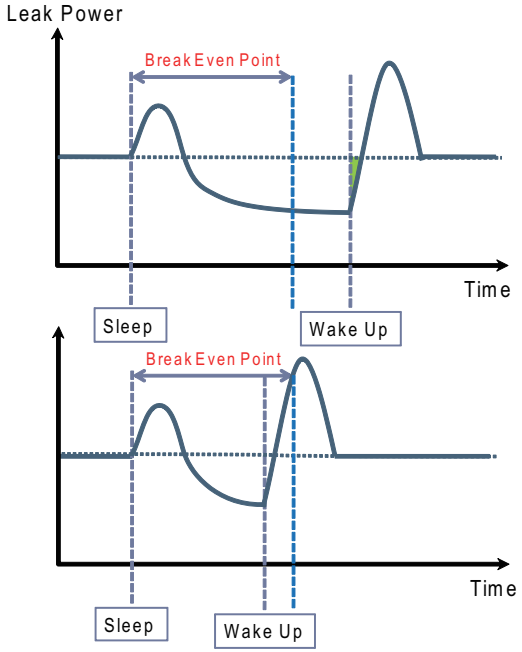

図 5 BEP について (上: BEP を満たす PG 下: BEP ミスする PG)

を実現している. 一方で下の図は, WakeUp が早すぎるためにパワーゲーティングによるスリープ期間の確保ができず, 電力ロスしている例である.

BEP ミス率は, ある期間におけるスリープサイクルに対して実際にミスしたスリープサイクル数の割合として定義される. ある期間における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, BEP ミス率は, 以下の式として定義される.

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

$$BEP_{MissRate} = \frac{\sum_{t_i \in S} t_i}{t_{sleepAll}} \quad (1)$$

つまり, BEP ミス率が高ければ高いほど, 細粒度 PG において電力のロスが大きくなることを意味する. Geyser OS における省電力機構は, 細粒度パワーゲーティングのスリープポリシーを OS が制御することで BEP ミス率を削減する方針の基に設計する.

5. Geyser OS における省電力機構の設計

5.1 Geyser OS の省電力機構概要

図 6 に, Geyser OS における細粒度 PG 制御機構の概要を示す. 本省電力機構は GeyserOS のタスクスケジューラによって実現され, 個々のタスクの演算器使用特性に基づいて制御を行う. 本省電力機構は, GeyserOS のタスクスケジューラ部と連携して動作する. 省電力機構は, 入力として前述した BEP ミス率を受け取り, 出力としてタスクコンテキストに対してスリープポリシーを与える. そして, スケジューラによって決定されたタスクのコンテキストからスリープポリシーを取得し, 省電力機構によって PGStatus レジスタにスリープポリシーを設定することで動作する.

5.2 BEP ミス率を取得するパフォーマンスカウンタ

ここで, BEP ミス率を求めるためのパフォーマンスカウンタの設計について述べる. BEP ミス率を求めるためには前節で示したように細粒度パワーゲーティングにおけるスリープイベントをカウントする機構が必要である. パフォーマンスカウンタによって計測可能な値を以下に示す.

- クロックサイクルカウンタ
- PG におけるスリープ期間とその回数 (N サイクルスリープが M_N 回)
- 全スリープサイクル数 $t_{sleepAll}$ $(= \sum_{N=1} M_N)$

BEP ミス率の計算には, PG におけるスリープ期間とその回数および全スリープサイクル数を用いる. 例として BEP が 100 サイクルであった場合, 式 (1) は以下のように示される.

$$BEP_{MissRate} = \frac{\sum_{N=1}^{100} M_N}{t_{sleepAll}}$$

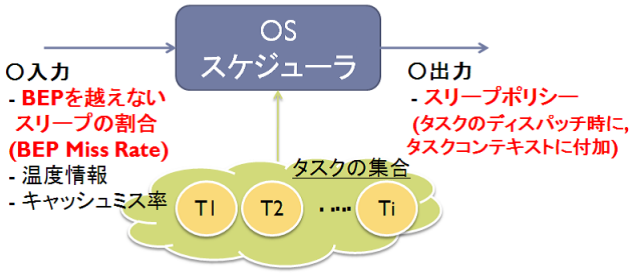

図 6 細粒度 PG 制御機構の概要

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

このパフォーマンスカウンタは, Geyser アーキテクチャの実装環境である RTL/ゲートレベルシミュレーションおよび Geyser on FPGA によって実現されている.

5.3 省電力機構の全体構成

本システムを実現する省電力機構の全体構成を図 7 に示す. 本省電力機構は, スケジューラにおけるタイムスライス時に実行され, 以下の順序で実行される.

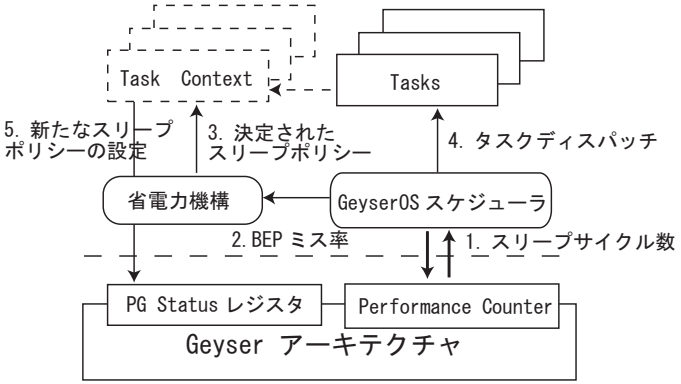

図 7 省電力機構の全体構成

(1) パフォーマンスカウンタよりスリープ期間とその回数を取得
メモリマップされたパフォーマンスカウンタから値を読みだす.

(2) BEP ミス率を計算し, 省電力機構に渡す

(3) 現在実行中のタスクコンテキストに, ある方針において決定されたスリープポリシーの設定

(4) 次タスクディスパッチ

(5) ディスパッチされたタスクコンテキストからスリープポリシーを PGStatus レジスタを用いて設定

省電力機構によって決定されるスリープポリシーとして, GeyserOS は, 「BEP ミス率が越えるスリープの割合を削減する」という方針をとる. BEP ミス率を削減する手段として, PGStatus レジスタを用いたスリープポリシーの変更を OS により行うことで, 細粒度パワーゲーティングの粒度を制御する. そこで各スリープポリシーは, 以下の図で示すように三つの閾値 $Thd_{PG->CMiss}$, $Thd_{CMiss->PG}$, $Thd_{CMiss->NoPG}$ を定めることで変更する. また PG 無しから毎サイクル PG へは, 一度のタイムスライスにより遷移すること

表 3 評価環境

評価環境	内容
実行環境	NC-Verilog によるシミュレーション
ベンチマーク	Quick Sort Matrix Dhrystone
動作温度	25 ℃/100 ℃
閾値 $Thd_{PG->CMiss}$	80
閾値 $Thd_{CMiss->PG}$	30
閾値 $Thd_{CMiss->NoPG}$	80

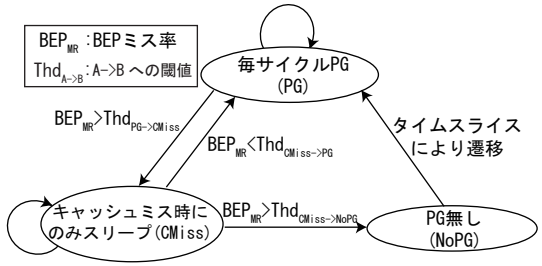

図 8 スリープポリシーの遷移条件

とする. 本省電力機構は, パフォーマンスカウンタを用いて各タイムスライスごとに BEP ミス率を計算し, スリープポリシーを実行時に動的に変更することで, 省電力を実現する.

6. 評　　価

6.1 評 価 環 境

評価環境を表 3 に示す. Geyser アーキテクチャの実現環境には, NC-Verilog を用いた RTL/ゲートレベルシミュレーションを用いる. また電力評価は, アクティブリーク電力を求めるときは HSIM および Power Compiler を用い, スリープリーク電力を求めるときには, HSPICE および本研究プロジェクトで作成されたスクリプトを用いて算出する.

また評価項目として, ALU のリーク消費電力およびリークエネルギーを評価する. ここで ALU を評価対象としたのは, ベンチマークプログラム三種に対して ALU の使用頻度が

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

基本的に高く, 本 OS の省電力機構によってスリープポリシーを変更される可能性が高いためである. また動作温度の 25 ℃/100 ℃は, 実行開始から終了までずっとその間で動作したと仮定した場合のシミュレーションによる評価となり, 実際の Geyser 実機においてこのように温度変化するわけではない.

また評価内における NoControl, PGControl の表記はそれぞれ以下のとおりである.

- NoControl: スリープポリシーを「動的に PG」に固定
- PGControl: スリープポリシーを OS により動的に変更

6.2 評価: ALU のリーク消費電力とリーク消費エネルギー

図 9 に, ALU のリーク消費電力を示す. OS による制御の結果, 制御前と比べて 5〜最大

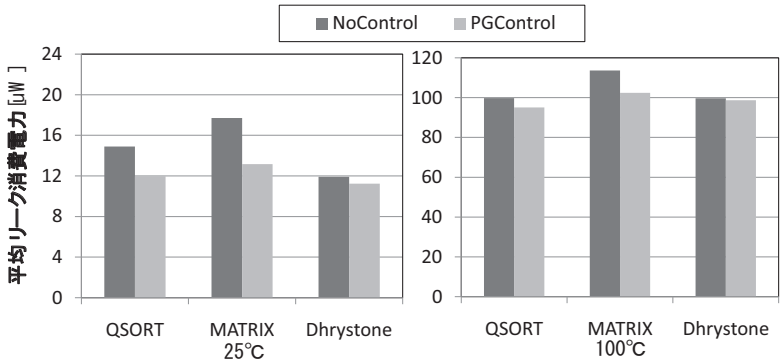

図 9 ベンチマークプログラムごとの平均リーク電力

25%の平均リーク電力削減を実現した. これは, OS の制御により BEP ミスを起こす ALU のスリープを削減できたためであると考えられる. また温度が 25 ℃から 100 ℃へ上昇したことにより, 平均リーク電力全体が増大し, さらに温度上昇により BEP が短くなるため BEP ミス率が減少することから, OS による制御の効果はあまり出ていない.

同様にリーク消費エネルギーを図 10 に示す. 省電力機構におけるオーバーヘッドが非常に小さいため, リーク消費電力と同様に 5%〜25%程度のエネルギー削減を実現した.

6.3 評価結果に対する考察

まず, 本稿で設計および実装を行った省電力機構に対する考察を行う. OS によりスリープポリシーを変化させた際の電力評価として, パワーゲーティングの効果が期待できるリー

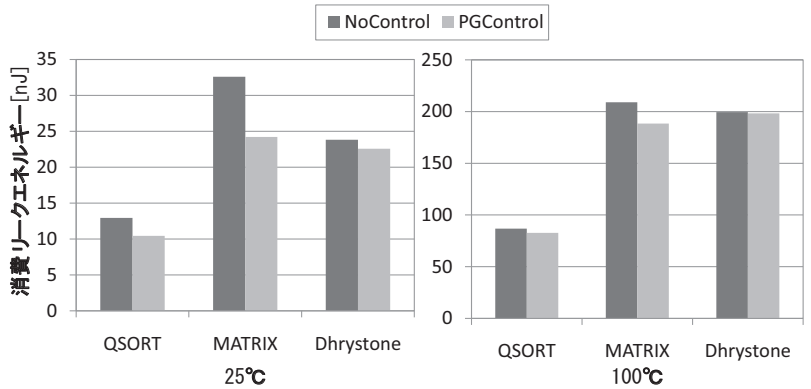

図 10 リーク消費エネルギー

ク電力について評価を行った. その結果, 最大で Matrix を 100 ℃で実行した際の 25%, 少ないもので DhryStone を 100 ℃で実行した時の 5% ほどのリーク電力削減を実現した. OS の省電力機構による BEP ミス率の変化に関して図 11 に示す. 図を見ると, OS の省電力

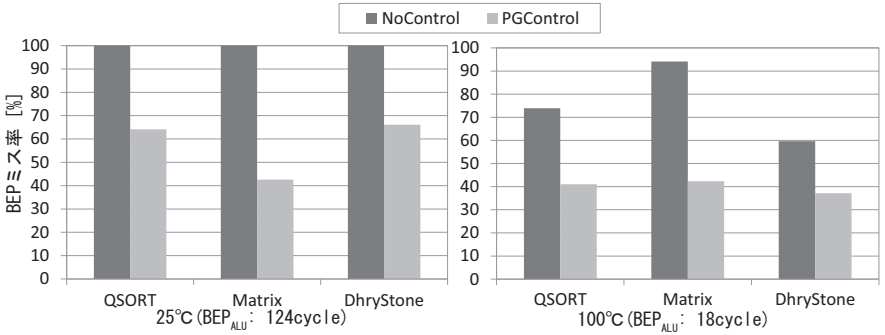

図 11 OS の省電力機構による BEP ミス率の変化

機構により BEP ミス率を大きく減少させることを実現しており, リーク消費電力を削減できたと言える. 25 ℃時の BEP ミス率が各ベンチマークプログラムで 100%となっているが, これは 25 ℃時の ALU の BEP が 124 サイクルと非常に長く, 特に様々なプログラムで使用

Vol.2010-ARC-189 No.8
Vol.2010-OS-114 No.8
2010/4/22

表 4 省電力機構の OS オーバーヘッド

ベンチマーク	NoControl[cycle]	PGControl[cycle]	オーバーヘッド [%]
Dhrystone	398931	400598	0.41
Matrix	367857	371908	1.10
QSort	174227	179105	2.80

されることの多い ALU では, そのスリープサイクル数を満たすことが難しいためである.

また, 温度ごとのリーク電力を考えると, 25 ℃では 10〜20[uW] 程度のリーク電力であったのに対し, 100 ℃での評価では, 90〜120[uW] 程度と, 温度によるリーク電力の違いが大きく分かる結果となった. 実行時の動作温度を考慮した制御は本省電力機構では今後の課題となっているが, 将来的に省電力機構へ導入したい.

またこの際の OS オーバーヘッドは, 表 4 に示すとおり, 最大 3% 程度となり, 低オーバーヘッドで省電力効果が得られているといえる. これらの評価により, 本稿で提案した, OS による細粒度パワーゲーティング制御手法が有効であることがわかる. 本結果はシミュレーションによる評価であり, 実 I/O, 実メモリを考慮した実機での評価が必要である.

7. おわりに

7.1 本研究の成果

本研究では, 省電力 MIPS プロセッサである Geyser の特徴である細粒度パワーゲーティングを, OS によって効果的に制御する手法について述べた. Geyser アーキテクチャ上で実行可能な OS は, 割込み/例外管理やタスク管理, システムコールを備えており, マルチタスク環境, 仮想メモリ管理, および省電力機構を実現する.

本研究の特徴である, OS によって細粒度 PG を効果的に制御することで, ハードウェアの自律制御と比べてさらなる省電力を実現する手法について述べた. 本 OS の特徴である省電力機構は, Geyser が提供している細粒度 PG の 3 種類のスリープポリシーを変更するためのインタフェースを操作することによって省電力を実現する. 本省電力機構は, マルチタスクを実現するタスク管理部において, タイムスライスごとに, BEP ミス率を減少させるようにスリープポリシーを実行時に決定することで, OS によって細粒度 PG を制御し, 省電力を実現する. 本 OS の省電力機構により, 省電力機構を用いない場合と比べて, リーク電力で 5〜20%の省電力化を実現した. またその際の OS オーバーヘッドは最大で 3%を越えない程度であり, リーク消費エネルギーについても同様に削減した.

7.2 今後の課題

今後の課題として, まず Geyser の実機による評価を行うことにより今回提案した省電力手法を評価する. そのうえで, BEP が動作温度によって大きく変動することを考慮して, タスクの実行順序を制御することにより一つの演算器の温度上昇を抑えることで, 省電力を実現する機構の設計, 評価などが挙げられる.

また, 研究プロジェクトとして他のグループ, 主にコンパイラとの連携について進めていくことで, さらに省電力化を実現する可能性があると考えられる.

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) K.Flautner, et al., “Drowsy Caches: Simple Techniques for Reducing Leakage Power”, In Proc. the 29th ISCA, pp.148-157 (2002)
3) S.Kaxiras, et al., “Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power”, In Proc. the 28th ISCA, pp.240-251 (2001)
4) Z.Hu, et al., “Microarchitectural Techniques for Power Gating of Execution Units” In Proc. ISLPED’04, pp.32-37 (2004)
5) K.Usami and N.Ohkubo, “A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signal”, In Proc. ICCD2006 (2006)
6) Zhigang Hu, Alper Buyuktosunoglu, Viji Srinivasan, Victor Zyuban, Hans Jacobson, Pradip Bose, IBM T.J. Watson Reserch Center “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 Aug).
7) Soumyaroop Roy et al, “A Compiler-Based Leakage Reduction Technique by Power-Gating Functional Units in Embedded Microprocessors” (2006)
8) Pratap Ramamurthy, Ramanathan Palaniappan, “Performance-directed Energy Management using BOS”, ACM SIGOPS Operating Systems Review, Volume 41, Issue 1, pp.66-77 (2007 Jan).