情報処理学会論文誌　コンピューティングシステム　Vol.5　No.5　23–32 (Oct. 2012)

# VLIW型プロセッサにおけるMixed Power Gatingの研究

石井 義史[1,a)]　王 蔚涵[1]　天野 英晴[1]

受付日 2012年3月28日, 採録日 2012年6月6日

**概要**：半導体プロセスの微細化により，LSI の漏れ電流（リーク）による消費電力の増加が問題になっている．このリークを効果的に削減する方法として，Power Gating（PG）がある．PG では，回路とグランドの間にパワースイッチを挿入し，スイッチを OFF（スリープ）することでリーク電力を削減する．PG 実装方式の 1 つである fine-grained PG を，小規模なプロセッサコアに対し，演算ユニットなど，限られた回路領域に適用することでリークを削減できるが，適用範囲が広くなると，fine-grained PG の性質により面積が増加する．そこで，面積増加の小さい実装方式の coarse-grained PG と組み合わせた Mixed Power Gating という手法を提案し，VLIW 型に拡張したプロセッサに適用したときのリーク削減効果などを調査した．MiBench のアプリケーションで評価した結果，fine-grained PG を適用した回路のみをスリープさせた場合に最大 10.9%，fine-grained PG と coarse-grained PG を適用した回路の両方をスリープさせた場合に最大 28.3%，リーク電力を削減できた．また，VLIW 型でない通常のプロセッサと比較して，前者の場合に最大 2.47 倍，後者の場合に最大 2.02 倍，性能が改善された．一方，PG のオーバヘッドは全体の面積の 5.03%であった．

**キーワード**：低消費電力，パワーゲーティング，VLIW

# A Study of Mixed Power Gating on VLIW Processors

Yoshifumi Ishii[1,a)]　Weihan Wang[1]　Hideharu Amano[1]

Received: March 28, 2012, Accepted: June 6, 2012

***Abstract:*** Power Gating (PG) is an effective way to reduce leakage power that becomes a big issue in LSI designs. There are two ways to implement PG: fine-grained PG and coarse-grained PG. In fine-grained PG, sleep control can be done quickly but the area overhead is large, while coarse-grained PG can be implemented with small area overhead but the long time is needed for power switching. By combining them, we proposed a mixed grain power gating (Mixed PG) and designed a VLIW processor with mixed PG called Geyser-VLIW. Through the evaluation with programs from MiBench, the leakage power at 25°C can be reduced by up to 10.9% in the case of sleeping only fine-grained PG circuits, and it can be reduced by 28.3% in the case of sleeping both fine-grained PG circuits and coarse-grained PG circuits. The performance is improved by up to 2.47 in the former case and improved by 2.02 in the latter case, compared to non-VLIW processors. The area overhead of Mixed PG is 5.03%.

***Keywords:*** low power, Power Gating, VLIW

## 1. 緒論

近年，スマートフォンに代表される性能を重視した携帯端末が登場したことにより高性能な CPU が求められている．しかし，一方で性能に応じて増加する消費電力と発熱はバッテリの寿命や携帯機器特有な密封性の観点から無視できなくなってきた．

CMOS 回路において消費電力は大きくダイナミック電力とリーク電力に分けられるが，プロセスルールの微細化（特に 90 nm 以下）にともなってリーク電力の全体に占める割合が大きくなってきている [1]．

リーク電力を削減する手法としては Dual Vth，基板バ

[1] 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University, Yokohama, Kanagawa 223–0061, Japan
[a)] geyser@am.ics.keio.ac.jp

イアス，Power Gating [2] などが代表的である．この中で Power Gating（PG）は，実装が容易で，動作速度を大きく落とさず，比較的低いオーバヘッドでリーク電力を減らすことができることから利用が進んでいる．この方法では，回路と電源（あるいはグランド）の間にパワースイッチと呼ばれる高スレショッルドで低リークなトランジスタを挿入する．そして，回路への電力供給を，挿入したパワースイッチで制御し，その回路を使用しないときに電力を遮断することでリーク電力を削減する．PG の実現手法には，coarse-grained PG と fine-grained PG があり，スイッチの速度，パワースイッチの切替えによる電力増加，スイッチ，アイソレーションセル，制御回路による面積増加が異なり，適用はこれらを十分考慮する必要がある．

組み込み機器に搭載されるプロセッサでは，特定用途において高い性能が要求されることが多く，特に画像処理，信号処理など並列処理が利用可能な応用分野での性能が重要である．そのため，プロセッサのアーキテクチャ構成を VLIW（Very Long Instruction Word）型に拡張し，1 度に複数の演算処理を実行できるようにすることで，要求される性能を実現する手法が有効である．しかし，VLIW 型の利用により回路規模が大きくなり，面積が増加するため，リーク電力が増大してしまう．VLIW 型のプロセッサは，複数の演算スロットを持つが，並列性の高いアプリケーション以外は，そのすべてが動作している場合は少ない．このため，必要のない場合は，PG を適用してリーク電力を節約し，必要に応じて演算スロットのパワースイッチを入れることで，性能とリーク電力の制御を広い範囲で行うことができる．しかし，PG はオーバヘッドをともなうため，この適用には設計上のトレードオフをよく検討することが必要である．

そこで，本研究では，VLIW 型プロセッサに適した PG 手法として，fine-grained PG と coarse-grained PG を組み合わせた Mixed PG を提案し，MIPS R3000 ベースの VLIW 型プロセッサ Geyser-VLIW に適用してその効果を評価する．

## 2. Geyser-1

### 2.1 Geyser-1 の構成

Geyser-1 [3] は，CPU の演算モジュールに対して fine-grained PG を適用し，ランタイム制御を行うことでリーク電力を減らす手法の適用例として開発された RISC プロセッサである．MIPS R3000 [4] の命令セットアーキテクチャに基づいており，32 個の汎用レジスタと，乗除算の結果を格納するための 64 ビットレジスタを持つ．パイプラインは，IF（Instruction Fetch），ID（Instruction Decode），EX（Exection），MEM（Memory Access），WB（Write Back）の 5 段ステージで構成されている．ALU，Shifter，Multiplier，Divider の 4 つの演算ユニットを持ち，

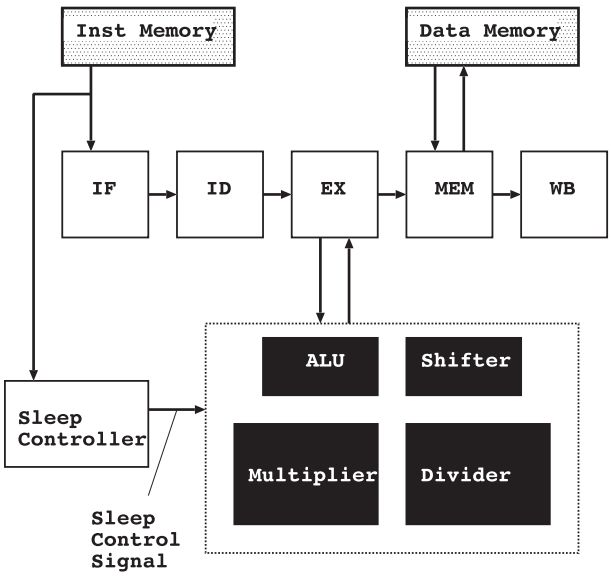



図 1　Geyser-1 の構成
**Fig. 1**　Block Diagram of Geyser-1.

それぞれに fine-grained PG が適用されており，独立にスリープ状態の制御ができる．**図 1** に Geyser-1 の構成の概略を示す．図中の Sleep Controller が各演算ユニットのスリープ制御を行う．

### 2.2 Geyser-1 における PG

従来の PG は CPU やキャッシュ，I/O モジュールなどチップ上の大きな面積を単位とし，これらを利用していない期間のリーク電力削減を目的としていた．しかし，大きな単位で電源を ON/OFF するため，電源が OFF であるスリープモードと，電源がオンの動作モード間の遷移にはマイクロ秒単位の時間を要した．

これに対して，Geyser-1 では，それぞれのゲートごとに仮想グラウンドを設けて，これらを一定数接続し，グラウンドとの間にスリープトランジスタを設ける．この方法は，専用ゲートライブラリを必要とするが，スリープトランジスタ数を調整することで，スリープモードに遷移する時間と動作モードに遷移する時間をナノ秒単位に短縮できる．この手法を用いれば，演算器レベルの細かい単位で，PG が可能になり，CPU が動作中でも，利用していないモジュールのリーク電流を削減できる．この手法を従来の Power Gating と区別して fine-grained PG と呼ぶ．Geyser-1 ではこれを演算ユニットに対して適用し，命令を読み出した直後に利用する演算ユニットを判定して，wake-up を行い，命令実行後にスリープさせている．

### 2.3 Geyser-1 の評価結果

実機評価より，Geyser-1 は最大 60 MHz で動作し[*1]，漏

[*1] Geyser-1 は，キャッシュを外付けとしたため，実機動作周波数が 60 MHz になったが，コア部は 200 MHz で動作するように設計されている．

表 1　fine-grained PG の面積オーバヘッド
**Table 1**　Overhead of the fine-grained PG.

| 演算器名 | 面積 | オーバヘッド面積 | 割合 |
|---|---|---|---|
| ALU | 3,594.8 | 386.8 | 10.76% |
| シフタ | 3,294.8 | 344.4 | 10.45% |
| 乗算器 | 27,782.8 | 2,027.6 | 7.30% |
| 除算器 | 28,263.6 | 1,516.4 | 5.37% |
| Geyser-1 | 128,981.6 | 7,911.6 | 6.13% |

れ電流を 25°C で 7%，80°C で 24%削減できることが分かった．また，fine-grained PG の面積オーバヘッドは**表 1** のようになった．

以上の結果から，fine-grained PG を演算ユニットに適用することで，CPU のリーク電力を削減できることが確認された．しかし，Geyser-1 は，MIPS R3000 という小規模なプロセッサコアをもとに設計されており，用途によっては演算性能が不足する．より高い演算性能を実現するための一手法として VLIW 化が考えられるが，fine-grained PG をすべての演算ユニットに用いた場合，そのオーバヘッドが無視できなくなる．そこで，本論文では Geyser-1 を VLIW 化した場合の PG 手法を検討し，必要な場合には，高い演算性能を実現する一方で，面積オーバヘッドとリーク電力を抑える方法を検討する．

## 3. Geyser-VLIW

### 3.1 VLIW 型プロセッサの構成

VLIW 型プロセッサは，スーパスカラ型と比べて制御回路が簡単で，容易に多数の命令を発行できるため，信号処理用プロセッサ DSP（Digital Signal Processor）では一般的な方法である．ここでは，Geyser-1 を拡張した VLIW 型プロセッサ Geyser-VLIW を設計する．Geyser-VLIW では，一般的な VLIW 型プロセッサと同様に，ロード，ストア，演算，分岐などの命令複数個からなる命令語（長命令）を構成する．この長命令を構成する個々の命令をスロット命令と呼び，スロット命令 1 つを実行する機構をスロットと呼ぶ．VLIW プロセッサの設計にあたってはまずスロット数を決める必要がある．ここではスロット数を 5 とした．これは DSP などで用いられている VLIW プロセッサよりも小さいが，本提案手法はスロット数が増えれば，より効果的である．このためまず小規模なシステムで解析を行うこととした．

次に，各スロットについて「スロットの種類」を決定した．スロットの種類には，演算のみを行う「演算スロット」と，メモリの読み書きを行う「メモリアクセススロット」がある．演算とメモリアクセスを同じ並列度で実行できるようにするため，演算スロット数を 3 スロット，メモリアクセススロット数を 2 スロットとした．そして，slot1〜slot2 にメモリアクセススロットを，slot3〜slot5 に演算スロットを割り当てた．Geyser-VLIW 全体の構成を**図 2** に示す．

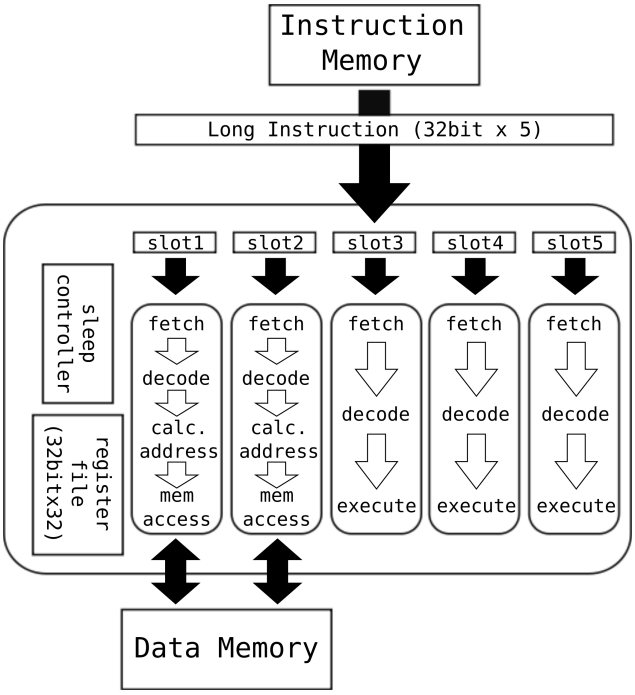



図 2　Geyser-VLIW の構成
**Fig. 2**　Structure of Geyser-VLIW.

Geyser-VLIW は，まず，外部の Instruction Memory から長命令をフェッチし，プロセッサ内部で 5 つのスロット命令に分割し，それぞれのスロット命令を対応するスロットに格納する．

メモリアクセススロットは，fetch，decode，execute，memory access，write back の 5 段ステージパイプラインで構成される．スロットに入った命令は，execute ステージでアクセスを行うメモリのアドレスを計算する．その後，memory access ステージで，外部の Data Memory とデータのやりとりを行い，最後に write back ステージでレジスタファイルにデータの書き込みを行う．

一方，slot3〜slot5 に割り当てた演算スロットでは，fetch，decode，execute，write back の 4 段ステージパイプラインで構成される．演算スロットに入ったスロット命令は，execute ステージでスロットに割り当てられた演算ユニット（3.1.1 項を参照）を使用して演算を行い，write back ステージで，レジスタに対し演算結果の書き込みを行う．

図 2 において，メモリアクセススロットは slot1〜slot2 に，演算スロットは slot3〜slot5 にそれぞれ対応している．図中では，write back ステージのブロックは省略している．

#### 3.1.1 演算ユニットの割当て

命令セットにある演算命令すべてを実行するためには ALU，シフタ，乗算器，除算器の 4 種類の演算ユニットが必要である．すべてのスロットが全種類の演算を実行できるように設定すると，面積が大きくなるため，スロットご

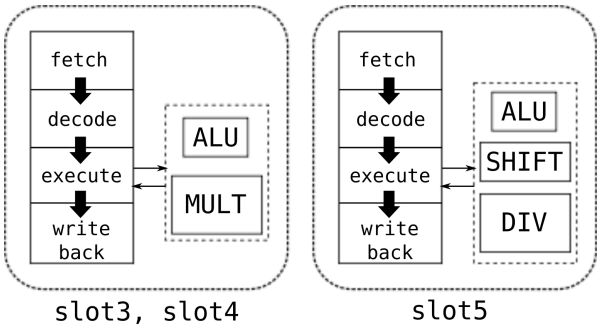



図 3　各演算スロットの演算ユニットの割当て
**Fig. 3**　Assignment of computational units to each slot.

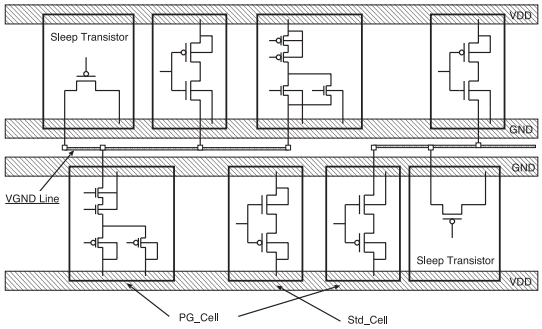



図 4　fine-grained PG（Cell-based 型）の実装
**Fig. 4**　Cell-based fine-grained PG.

とに，実行できる演算の種類を変えて割り当てる．多くのアプリケーションでは，ALU での演算が最も多く，乗算がこれに次ぎ，シフタと除算器は特定のアプリケーションを除き，使用頻度が低い．Geyser-VLIW では，演算スロットに割り当てた slot3〜slot5 のうち，**図 3** のように，slot3 と slot4 には ALU と乗算器を，slot5 には ALU，シフタ，除算器を割り当てた．

### 3.2　Mixed Power Gating

Geyser-VLIW に対しても，Geyser-1 と同様に演算ユニットすべてに fine-graind PG を適用し，命令に応じてスリープ制御を行うことで，リークを削減する方法が考えられる．しかし，実際のプロセッサ上で動かすアプリケーションのコードを解析すると，一部の並列性の高いアプリケーションを除いて，VLIW 型プロセッサ中のスロットすべてが命令で埋まる可能性は高くない．よって，演算ユニットだけでなく，演算ユニットを含めたスロット全体をスリープさせたほうが，より大きなリーク電力の削減が見込めるはずである．そこで，いくつかのスロットに対しては，スロット全体に対して PG を適用するように決めた．

スロット全体に PG を適用するうえで，Geyser-1 で適用した fine-grained PG では問題がある．fine-grained PG は，高速にスリープ状態の ON/OFF を切り替えることができるが，表 1 で示したように，パワースイッチ挿入による面積増加が大きい．回路規模が大きくなれば，パワースイッチの挿入が多くなる傾向にあるため，面積増加が大きくなる．そこで，面積オーバヘッドの小さい，coarse-grained PG を併用することを考えた．この場合，coarse-grained PG でスリープ状態になったスロットは，プログラム実行中動作することができないが，全体として大きなリーク電流削減効果を得ることができる．

まず，5 つのスロットのうち，いくつかをスロット全体に PG を適用するスロットとし，残りのスロットでは Geyser-1 同様にそれぞれの演算器単位に PG を適用する．スロット全体に PG を適用する場合は coarse-grained PG を用いプログラム実行中は完全にスリープ状態とする．一方，それぞれの演算器に PG を適用する場合は，fine-grained PG を用い，Geyser-1 同様に動作中に利用していない演算器をスリープ状態にする．この手法を Mixed Power Gating（Mixed PG）と呼ぶ．対象とするアプリケーションが十分な並列性を持たないか，あるいは電力消費の削減が優先される場合は，いくつかのスロットを coarse-grained PG を使ってスリープ状態にし，スロット数を減らして動作させる．動作中のスロットの演算器にも fine-grained PG が適用されるので，リーク電力を可能な限り抑えることができる．十分な並列性を持つアプリケーションでかつ性能が重視される場合は，全スロットを動作させることで，高い性能を実現できる．すなわち Mixed PG は，リーク電力の削減と性能向上を広い範囲で制御することが可能である．Mixed PG は，1 つの CPU 内で coarse-grained PG と fine-grained PG をスロットごとに使い分けることを意味し，1 つのモジュールに対して coarse-grained PG と fine-grained PG を併用することではない．1 つのモジュールに対する併用は，ノイズや仮想 GND レベルの上昇の問題があり，今後の課題である．

### 3.3　Mixed Power Gating の実装

ここでは，fine-grained PG は Cell-based 型で実装し，coarse-grained PG は Ring-based 型で実装する．両者を簡単に説明する．

#### 3.3.1　fine-grained PG の実装

Cell-based 型の実装方式は，スタンダードセルを拡張して仮想 GND と GND を用意し，**図 4** に示すように，パワースイッチを論理セルと混在させて配置する方式である．

Cell-based 型の実装方式では，sleep の ON/OFF 状態の切替えを高速に実行できる利点があり，このために Geyser-1 はこの Cell-based 方式を用いている．一方，表 1 に示したようにパワースイッチ挿入などによる面積オーバヘッドが大きく，専用の仮想 GND を持つ Cell を必要とする問題点がある．

#### 3.3.2　coarse-grained PG の実装

coarse-grained PG は，現在多くの携帯機器用の SoC

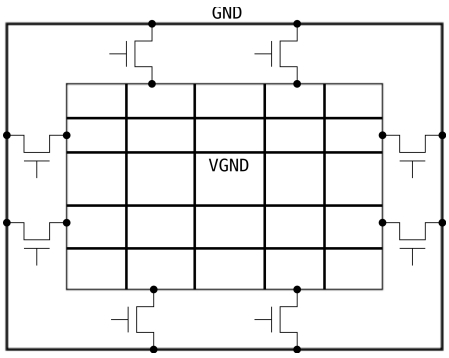



図 5　coarse-grained PG の実装
**Fig. 5**　Ring based coarse-grained PG.

表 2　PG 適用パターン
**Table 2**　Patterns for PG application.

| | slot1 | slot2 | slot3 | slot4 | slot5 |
|---|---|---|---|---|---|
| パターン 1 | none | CPG | FPG | CPG | FPG |
| パターン 2 | none | CPG | FPG | FPG | FPG |

（System-on-a Chip）で利用されている方法で，**図 5** に示す Ring-based 型を用いる場合が多い．この方法では，パワースイッチを PG 適用回路の周りに配置し，その周りを通常の電源（VDD）あるいはグランド（GND）のリングで囲む [5]．レイアウト上，ある大きなブロックに電源供給を行うためには，元々 Ring を作る場合が多い．このため，元々存在する Ring を利用してその間にパワースイッチを入れる Ring-based 型は，面積オーバヘッドが少なく，実装の方法によってはスリープ時のリーク電流を fine-graind 型よりも大きく減らすことができる．しかし，大きな面積の電源ドメインに対してパワースイッチを ON/OFF することから，ノイズの発生が問題となり，通常スリープ，wake-up は時間をかけて行う．したがって，Geyser-1 で用いるように命令を読み出してその内容に応じて ON/OFF することには向いていない．

#### 3.3.3　PG の適用パターン

Mixed PG の適用には，全体に coarse-grained PG を適用するスロットと演算器単位に fine-grained PG を適用するスロットを適切に選択しなければならない．PG の適用パターンを考える場合，以下の点を考慮した．まず，coarse-grained PG を適用したスロットをすべてスリープさせた状態でも残りのスロットで 4 種類すべての演算ユニットが使用できなければならない．また，Geyser-1 を上回る性能を得るためにつねに 2 スロットは動作可能とした．設計の最初に，coarse-grained PG を適用するスロット数を 2 および 1 のそれぞれ 2 つのパターンで予備評価を行った．これを**表 2** に示す．以後，表あるいは図中では fine-grained PG を FPG，coarse-grained PG を CPG と表記する．なおこの評価は 4.1 節の表 4 に示す設計環境を用いた．

表 3　各動作モードにおける Sleep 制御
**Table 3**　Sleep control for each operational mode.

| | Unit Sleep モード | Slot Sleep モード |
|---|---|---|
| FPG スロット | 演算器に対し Sleep 制御 | 演算器に対し Sleep 制御 |
| CPG スロット | Active 状態 | スロット全体を Sleep |

ここで，coarse-grained PG を適用したスロットを CPG スロットと呼び，fine-grained PG を適用したスロットを FPG スロットと呼ぶ．パターン 1 は，slot2，slot4 が CPG スロットであり，パターン 2 は，slot2 のみが CPG スロットである．

CPG スロット，FPG スロットの制御には様々な方法が考えられるが，VLIW への適用は本研究が最初であることから，ここでは単純な方法を用いることにした．すなわち，FPG スロットは Geyser-1 同様にフェッチした命令に応じて対応した演算器のみが ON になり，演算終了後すぐにスリープモードに遷移する．CPG スロットはアプリケーションに応じて全体をスリープするかどうかを決め，実行時には状態を変えない．今回，複数 CPG スロットがあっても，両方とも ON か OFF かで状態を揃える，このことで，制御回路は簡単になる．ここで，CPG スロットを ON にした状態を Unit Sleep モード，OFF にした状態を，Slot Sleep モードと呼ぶ．Slot Sleep モードは省電力を重視した動作状態であり，Unit Sleep モードは性能を重視した動作状態である．モードの切替えはコンパイラあるいは OS によりアプリケーション実行前に行われる．モードにかかわりなく FPG スロットでは，Geyser-1 同様に，命令に応じて演算器単位の ON/OFF が行われる点に注意されたい．

動作モードごとの Sleep 制御をスロットの種類別にまとめると，**表 3** のようになる．

### 3.4　面積オーバヘッドとリーク電力の予備評価

まず，予備評価として，CPG スロット数の影響を調べるため，論理合成によって得られた回路の面積，リーク電力から，各 PG パターンの面積増加と ON/OFF 時のリーク電力を算出し，比較を行った．本節の評価は方式検討のための予備評価であるため，リーク電力はスリープ時のスタティックな値とし，ON/OFF 時のオーバヘッドを想定していない．Synopsys 社の Power Compiler（RTL Syntesis 2009.06-SP5 中の電力評価用ツール）を用いて行った．その結果を**図 6**，**図 7** に示す．温度は 25°C とした．その他の評価条件は 4.1 節の表 4 と同一である．

図 6 では，PG を適用していない場合のプロセッサ全体の面積を評価基準（0%）とし，各 PG パターンを適用したときの面積との差分を面積オーバヘッドとした．図 7 では，PG を適用しない場合のプロセッサのリーク電力を基準に，各 PG パターンを適用した場合のプロセッサのリーク電力との差分を計算し，リーク削減率を求めて比較を

行った．リーク削減率は，評価の節で利用した回路シミュレータで求めた値を用いた．

期待したとおり，PG パターン 1 の構成をとると，CPG スロットが多いため，動作モード間のリーク電力に大きな差が生まれ，Slot Sleep モードでのリーク電力削減が大きくなる．一方，PG パターン 2 では，FPG が演算ユニットにのみに適用されたスロットが多いため，Slot Sleep モードでのリーク電力削減は小さくなってしまうが，Unit Sleep モードのときのリーク電力をより小さくすることができる．したがって，動かすアプリケーションやシステムの，プロセッサに対する性能要求が大きく偏っている場合，つまりアイドル時間が長くとれる機会が多い場合は PG パターン 1 が，安定して高い計算性能を要求される場合は PG パターン 2 が適していることが分かった．

### 3.5　VLIW 用実行コードの生成

設計した Geyser-VLIW 上でアプリケーションを動作させるためには，Geyser-VLIW 用のコンパイラを作成して，アプリケーションのソースコードを実行コードに変更する必要がある．

今回は，既存の MIPS プロセッサ用の C Compliler である mips-gcc を用いて，ソースコードをアセンブラコードに変換し，そのアセンブラコード中の命令を Geyser-VLIW のスロットに割り当てることで，Geyser-VLIW 用の実行コードを作成した．Geyser-VLIW 用の長命令は，Geyser-1 の命令 32 bit を 5 スロット分持つため，160 bit となる．

具体的には以下のような流れで，実行コードを生成する．

(1) C 言語で記述されたアプリケーションのソースコードを mips-gcc を用いて MIPS のアセンブラコードに変換
(2) Slot Scheduler を用いて，Geyser-VLIW 用のアセンブラコードに変換
(3) アセンブラコードから，Geyser-VLIW 用のアセンブラを用いて実行コードを生成

上記のコード生成フローにおいて，命令の割当てを決めるために専用の Scheduler を作成して，コード生成環境を構築した．Slot Scheduler は，それぞれの命令の依存関係を調べ，並列に実行できるかどうかを判断し，並列に実行可能であれば，その命令が実行できるスロットに命令を割り当てる．また，Geyser-VLIW では，動作モードごとに使用可能なスロット数が異なるため，動作モード別に実行コードを用意する必要がある．具体的には，Slot Scheduler を用いて Geyser-VLIW 用のアセンブラコードに変換する際に，Slot Scheduler にオプションを与えることで，Unit-Sleep 用のコード，あるいは Slot-Sleep 用のコードを生成する．

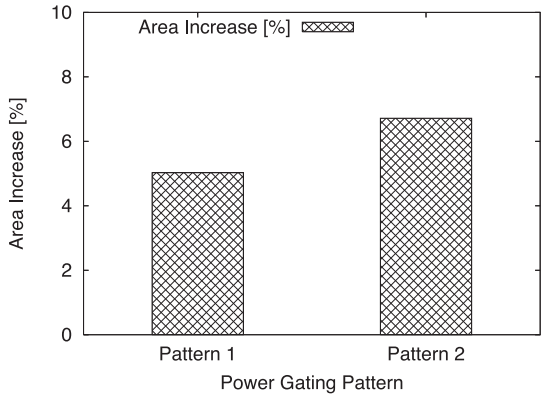



図 6　面積オーバヘッド
**Fig. 6**　Area overhead of Pattern 1 and 2.

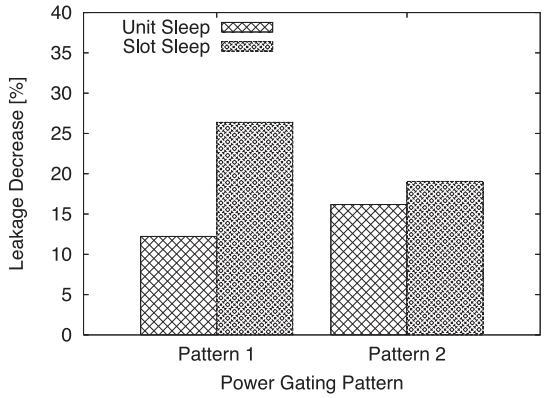



図 7　リーク削減効果
**Fig. 7**　Leak reduction of Pattern 1 and 2.

## 4.　評価

本章では，2 つの CPG スロットを持つ「PG パターン 1」を適用した Geyser-VLIW の設計とプログラム動作時の性能と ON/OFF におけるオーバヘッドを含んだリーク電力評価を報告する．

### 4.1　設計・評価環境

Geyser-VLIW は，Geyser-1 と同じプロセスを用いており，Veirlog HDL で記述されている．今回の評価は CPU 部のみであることから，メモリは理想的なものを想定しており，キャッシュのミスヒットなどの影響は反映されていない．設計で使用したテクノロジライブラリ，シミュレーションや論理合成に使用したツール（CAD）を**表 4** に示す．これらはすべて Geyser-1/2 [3], [6] で用いたものと同じである．

Geyser-VLIW は，Geyser-1/2 と同じく e-shuttle 65 nm

表 4　設計・評価環境
**Table 4**　Design tools and used process.

| | |
|---|---|
| Technology | Fujitsu e-shuttle 65 nm |
| Cell Library | CS202SZ 1.2 V |
| RTL Simulator | Cadence NC-Verilog 08.20-s002 |
| Circuit Simulator | Synopsys hsim 2010.12 |
| Synthesis Tool | Synopsys RTL Synthesis 2009.06-SP5 |

CS202SZ cell library を用いて標準電圧である 1.2 V において 200 MHz で動作するように設計されている．fine-grained PG の wake-up 遅延も 5 nsec 以内になるようにスリープトランジスタの個数，制御回路が設計されている．本章の評価はすべて 200 MHz での動作を想定している．

また，今回の評価では，キャッシュを想定せず，命令，データともに 1 クロックでアクセスできるメモリ中に格納されているという理想のメモリを仮定している．キャッシュミス時は，すべての演算器をスリープ状態とすることができるため，小規模なミス率の高いキャッシュを想定すると，リーク電力を大きく削減した結果が得られる．しかし，組み込みシステムの多くのプログラムではキャッシュミスを頻発することは少ないため，最も厳しい条件ということで理想のメモリを想定した．キャッシュを装備した現実的な構成では，ミス率に応じてより大きなリーク電力を削減することが可能である．

評価用アプリケーションとしては，Geyser-1 の評価に使われた，組み込みプログラムのベンチマーク集 MiBench [7] を利用した．MiBench は，利用分野ごとに package として各プログラムが分類されており，automobile package から quick sort（qsort），consumer device package からメディア処理を代表して jpeg encoder（jpeg），network package から dijkstra，security package から blowfish，telecommunication package から 32-bit Cyclic Redundancy Check（crc32）の 5 つのアプリケーションを，シミュレータを用いて Geyser-VLIW 上で動作させ，性能とリーク電力を測定した．

予備評価とは異なり，ここで測定したリーク電力とは，「ゲートリーク電流やサブスレッショルドリーク電流などが原因の消費電力（スタティック）」と「パワースイッチ周りのセルなどの ON/OFF 充放電による消費電力のオーバヘッド」の 2 種類の電力を含んだものと定義する．

評価は以下のように行っている．coarse-grained PG については，Synopsys 社の回路シミュレータ hsim を用いて演算器の面積とスリープトランジスタ数より，スリープ時，アクティブ時の漏れ電流を見積もった．しかし，fine-grain PG については，アプリケーションの実行によって，スリープトランジスタの ON/OFF のタイミングが異なり，ON/OFF にともなう電力損失と，スリープ時の漏れ電流削減効果が違ってくる．hsim などの回路シミュレータは，実行時間が長いため，アプリケーション実行時の CPU のリーク電流を直接測定することは現実的でない．そこで，本章のリーク電力評価は Geyser [3], [6] で用いたのと同じ方式を用いている．以下，この手法を解説する．fine-grained PG についても，hsim を用いてアクティブ時，スリープ時の漏れ電流，パワースイッチの ON/OFF 時に消費される電力を測定することができる．このデータより，演算器ごとに，スリープ状態が続いたクロック数に対して，削減できるリーク電力を調べて表を作る．スリープクロック数が BET よりも大きければ，表の値はプラスとなり，電力削減の効果が得られるが，そうでなければマイナスとなり，スリープトランジスタの ON/OFF により電力を余分に消費することになる．

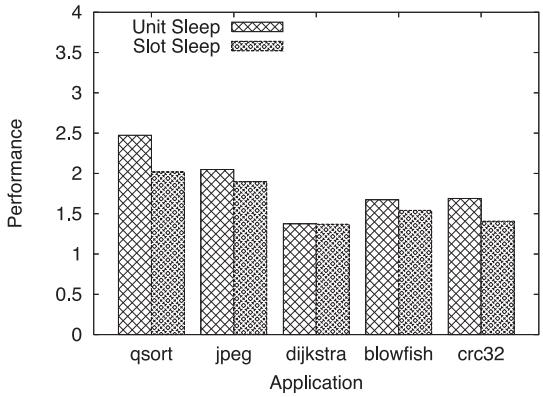



図 8　性能改善率
**Fig. 8**　The relative performance.

次にアプリケーションを Geyser-VLIW で実行した際に，各演算器がスリープ状態に連続して存在したクロック数についての統計データをとる．これには Cadence 社の論理シミュレータ ncverilog を用いて，アプリケーション全体をシミュレーションする．実行中に採取したデータに基づき，連続してスリープしたクロック数に対し，それが生じた回数を記録した表を作成する．この 2 つの表からアプリケーション実行時のリーク電力の削減効果を算出することができる．

### 4.2　評価結果

#### 4.2.1　性能改善

**図 8** に，Geyser-1 で評価用アプリケーションを実行した場合と比較して，Geyser-VLIW で実行することによる，性能の改善率を示す．なお，Geyser-VLIW は，Unit Seep モードで動作すれば，Geyser 同様に，PG 機構を用いることでの性能低下は生じない．

図 8 では，横軸に評価アプリケーション，縦軸に性能改善率を示す．性能改善率が 1.0 以上であれば，その評価アプリケーションにおいて，Geyser-1 よりも短い実行時間で，そのアプリケーションを完了できることになる．

Geyser-VLIW では，5 スロットの VLIW 型に拡張したことにより，Geyser-1 に比べて最大で 5 倍，性能が改善される可能性があるが，実際は並列性の制限によりそこまで性能は向上できない．図 8 の結果から，Geyser-1 と比較して，5 スロットの VLIW 型に拡張することで，Unit Sleep モードで最大 2.47 倍，Slot Sleep モードで最大 2.02 倍に性能が改善されることが分かった．しかし，個々のアプリケーションの性能改善率をみると，最大 5 並列の命令処理が可能な動作モードである，Unit Sleep モードにおける性

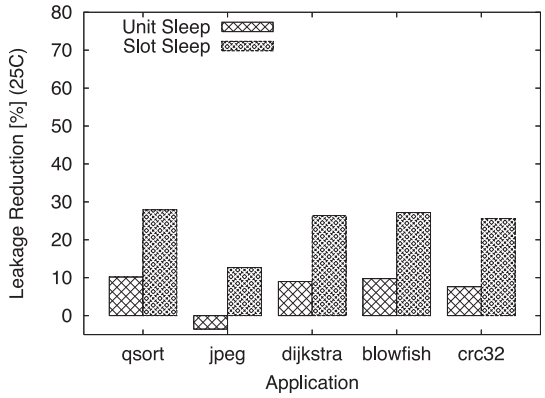



図 9　リーク削減率（25°C）
**Fig. 9**　The Leakage Reduction Ratio (25°C).

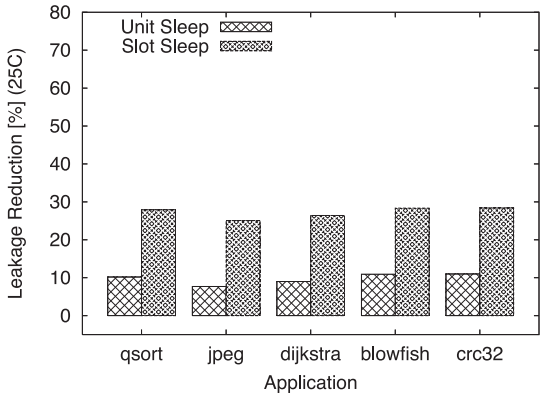



図 10　リーク削減率（25°C，BET 考慮）
**Fig. 10**　The Leakage Reduction Ratio (25°C, considering BET).

能改善率が，qsort，jpeg では 2.0 倍以上であるが，他の 3 つのアプリケーションでは，1.3〜1.7 倍の改善率にとどまっている．これらのアプリケーションでは，Unit Sleep モードの性能改善率が，最大 3 並列の命令処理が可能な Slot Sleep モードの場合とほとんど変わらない結果となっている．この原因は，これらのアプリケーションにおいて並列に実行できる部分（命令）が少ないためであり，Slot Sleep モードを利用してリーク電力を削減すべきアプリケーションであるといえる．

#### 4.2.2　リーク削減効果

各アプリケーションを Geyser-VLIW で実行した際のリーク電力を測定し，リーク削減率を求めた．その結果を示したものが**図 9** である．予備評価と同様に，PG 回路を用いないハードウェア構成におけるリーク電力を基準としている．測定温度は 25°C とした．図 9 のグラフでは，それぞれのアプリケーションを実行したときのリーク削減率が，動作モードごとに示されている．リーク削減率が 0%よりも大きければ，リーク電力を削減できたことになり，逆にマイナスの値となれば，ON/OFF 時のオーバヘッドによってリーク電力が増加したことを表す．

図 9 のグラフの，jpeg を除く他の 4 つのアプリケーションでのリーク削減率をみると，Slot Sleep モードでは，CPG スロット全体のスリープと FPG スロットの演算器単位のランタイムスリープ制御により，25%以上リーク電力を削減できており，また，FPG スロットのスリープ制御のみを行う Unit Sleep モードにおいても，Geyser-1 と同程度のリーク削減率（10%前後）が得られている．

jpeg でのリーク削減率をみると，1 Slot Sleep モードでも，12.6%となっており，他のアプリケーションと比較して削減率が小さくなっている．さらに，Unit Sleep モードでは，削減率がマイナス（–3.5%）になっている．これは FPG を適用した演算ユニットの auto-sleep 制御により，演算ユニットの ON/OFF におけるオーバヘッドがリーク電力の減少を上回ってしまったことを意味する．そこで，次にこのオーバヘッドの削減を試みる．

#### 4.2.3　BET 考慮時のリーク削減効果

FPG スロットにおけるランタイム PG は，ON/OFF の切替時にエネルギーオーバヘッドが存在する．各演算ユニットは，一定の時間スリープさせないと，スリープさせたことによるリーク電力の削減を，切替え時のオーバヘッドが上回ってしまう．削減したリーク電力と，切替えオーバヘッドの等しくなる時間を Break Even Time（BET）と呼ぶ．各演算器は，BET 以上の時間，PG 回路をスリープさせる必要がある．これを考慮しないと，たとえば jpeg などのアプリケーションでは，面積の大きい Multiplier や Divider に対するパワースイッチの ON/OFF を頻繁に行うことによる切替え電力がリーク電力の節約効果を上回り，全体の電力を増加させる結果となる．

そこで，元のアプリケーションのアセンブラコードを用いて，BET 解析を行い，BET を考慮したアプリケーションのコードを作成し，Geyser-VLIW で実行させ，リーク削減率の評価を再び行った．各演算命令には keep bit を持たせ，実行終了後もスリープモードに遷移しないようにした．この bit をスケジュール時に制御することで，BET よりも短い時間ではスリープさせないようにした．keep bit は元の R3000 の命令フィールドで利用していない部分を使って実装されており，命令長を拡張する必要はない．

**図 10** に，BET を考慮したときのリーク電力の削減率を示す．jpeg では，面積が大きな乗算器と除算器の ON/OFF が頻繁に行われオーバヘッドが大きかった．このため，BET を考慮しない場合では，PG の電力オーバヘッド以上のリーク電力の削減効果が得られず，逆に削減リーク電力が増加した．しかし，BET を考慮して ON/OFF を制御した結果，BET を下回る ON/OFF を約 30%削減することができた．このことで，jpeg 実行時のリーク電力を，Unit Sleep モードの場合においても 7.69%削減することができた．また，Slot Sleep モードでは 25.1%削減することができ，リーク電力をより大きく削減できた．最終的には，BET を考慮した場合では，常温（25°C）におけるリーク

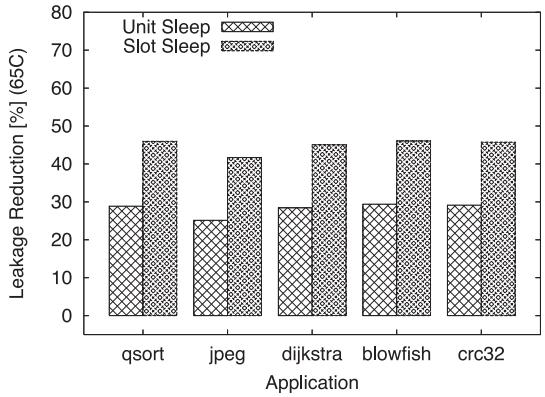



図 11　リーク削減率（65°C，BET 考慮）
**Fig. 11**　The Leak Reduction Ratio (65°C, considering BET).

電力を Unit Sleep モードで最大 10.9%，Slot Sleep モードで最大 28.3%削減できることが分かった．しかし，その他のアプリケーションでは，もともとリーク電力の削減効果の大きい乗算器と除算器の利用頻度が少なかったため，さほど効果はあがっていない．

ただし，今回の BET を考慮した ON/OFF 制御はアセンブラのコードの中の命令の間隔を測る簡単なツールによるものであり，ループにまたがった間隔の制御や，スケジュールを行って間隔を保つことはやっていない．この点についてはすでに多くの関連研究 [8], [9] があり，これらを取り入れることは今後の課題である．

#### 4.2.4　高温時のリーク削減効果

リーク電力は，高温になればなるほど大きくなる性質を持つため，高温状態における各アプリケーションのリーク削減率を測定した．その結果が，**図 11** である．高温状態（65°C）では，Unit Sleep モード時に最大 29.4%，Slot Sleep モードにおいて最大 46.1%のリーク電力を削減できた．

## 5.　関連研究

coarse-grained PG は，リーク電流の削減のため，長期間利用していないモジュールをスリープさせる方法として，最近のシステム LSI では一般的に用いられている [5]．一方，さらにリーク電流を削減するため，プロセッサの演算ユニットに対して fine-grained PG を積極的に適用する研究も，本論文のベースとなった Geyser-1 [3]，Geyser-2 [6] 以外にもいくつか行われている．

Youssef らは，プログラムの振舞いを一定の時間セグメントでトレースし，演算モジュールのスリープ可能な時間を予測する方法を提案している [10]．Lungu らもモニタの結果に基づき PG の質を保証する手法を提案している [11]．さらに Loop-Directed Mothballing（LDM）は，実行時にループを解析し，演算器単位でループ内の利用率を記録して 2 つのスレッショルドレベルを用いて PG を制御する [12]．これらの手法が実行時の挙動を観測することで，PG を行うかどうかを判定しているのに比べて，コンパイル時に演算器の利用間隔を予測し，PG を行うかどうかを判定する手法 [8], [9] も提案されている．

これらの研究は，BET を満足するように fine-grained PG を制御する方法を提案しており，VLIW のスロットに対して 2 種類の PG を適用して制御する本論文の提案手法とは，目的が異なっている．これらの研究で提案された fine-grained PG の制御手法のうちコンパイラによる手法は，本論文の提案手法においても，fine-grained PG を用いたスロット制御に適用することができる．これは今後の課題である．

本論文の提案手法と最も近いのは，Maro らの提案 [13] であり，この手法では DEC Alpha の演算ユニットを浮動小数点演算ユニット 1 つと，2 つの整数演算命令ユニットのクラスタに分離して，それぞれ利用率を計測して PG をするかどうかを決めていく．この手法は本論文で提案した corase-grained PG を用いたスロットの制御手法に近い．しかし，本論文での提案手法は，fine-grained PG のスロットを組み合わせて用いている点でこれとは異なっている．また，Maro らの提案手法は各クラスタがそれぞれ命令発行能力を持っている点で強力なプロセッサを想定しており，この点でも異なっている．

## 6.　結論

Geyser-1 を VLIW 型プロセッサに拡張した Geyser-VLIW を設計し，これに対して fine-graind PG と coarse-grained PG を併用する Mixed Power Gating を適用した．性能，電力，面積オーバヘッドの評価を行った結果，Mixed Power Gating の適用により，面積は 5.03%増加したが，Unit Sleep モードで最大 2.47 倍，Slot Sleep モードで最大 2.02 倍，性能を改善できた．また，リーク電力を，常温（25°C）で Slot Sleep モードを用いて最大 28.3%削減できた．温度が高温（65°C）の場合は，最大 46.1%削減可能である．以上の結果から，Geyser-VLIW では，動作モードを使い分けることで，広い範囲で電力と性能の制御が可能であるといえる．CPU の命令レベルで fine-grained PG を行う試みの実現例は Geyser が初めてであり，VLIW への適用は本研究以外は文献に登場していない．

現在，Geyser-VLIW のスケジューラは gcc をフロントエンドとしており，Trace scheduling などの VLIW プロセッサ用のスケジュールが行われていない．また，スリープ間隔を大きくするように，スロットの割当てを変えるなどのスケジュールも未検討であり，これらの適用により性能の向上と，リーク電力の削減が見込まれる．また，スロット数が大きい場合，提案手法はさらに有効であると考えられるが，この評価については，今後の課題である．

本論文の提案手法は，リーク電力の削減を目的としており，ダイナミックな電力を含めた全電力についての検討は行っていない．VLIW 型プロセッサは大きなデータパス

を持つため，ダイナミック電力の削減はオペランドアイソレーション，クロックゲーティングなど古典的な手法を用いる必要があり，さらに電圧制御，周波数制御の組合せも有効であると考えられる．全体としての VLIW 型の電力削減は今後の課題である．

**謝辞**　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「CREST」における研究領域「情報システムの超低消費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．

## 参考文献

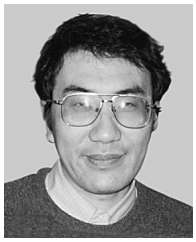
[1] Lee, D., Blaauw, D. and Sylvester, D.: Gate oxide leakage current analysis and reduction for VLSI circuits, *IEEE Trans. Very Large Scale Integration* (*VLSI*) *Systems*, Vol.12, No.2, pp.155–166 (2004).
[2] Long, C., Xiong, J. and Lui, Y.: Techniques of Power-gating to Kill Sub-Threshold Leakage, *IEEE Asia Pacific Conference*, pp.952–955 (2006).
[3] Ikebuchi, D. et al.: Geyser-1: A MIPS R3000 CPU core with fine grain runtime power gating, *Proc. A-SSCC2009*, pp.281–284 (2009).
[4] Kane, G.: *Mips R2000/3000*, Prentice Hall (1987).
[5] Keating, D.F.M. et al.: *Low Power Methodology Manual: For System-on-Chip Design*, Springer, New York (2007).
[6] Lei, Z., Ikebuchi, D., et al.: Geyser-2: The 2nd prototype CPU with fine-grained run-time power gating, *Design Automation Conference* (*ASP-DAC*), *2011 16th Asia and South Pacific*, pp.87–88 (2011).
[7] Guthaus, M.R., Ringenberg, J.S., et al.: MiBench: A free, commercially representative embedded benchmark suite, *Proc. 4th Annual IEEE International Workshop on Workload Characterization* (2001).
[8] Zhang, W., Kandemir, M., Vijaykrishnan, N., Irwin, M. and De, V.: Compiler support for reducing leakage energy consumption, *Design, Automation and Test in Europe Conference and Exhibition, 2003*, pp.1146–1147, IEEE (2005).
[9] Roy, S., Ranganathan, N. and Katkoori, S.: A Framework for Power-Gating Functional Units in Embedded Microprocessors, *IEEE Trans. Very Large Scale Integration* (*VLSI*) *Systems*, Vol.17, No.11, pp.1640–1649 (2009).
[10] Youssef, A., Anis, M. and Elmasry, M.: Dynamic standby prediction for leakage tolerant microprocessor functional units, *39th Annual IEEE/ACM International Symposium on Microarchitecture* (*MICRO'06*), IEEE Computer Society (2006).
[11] Lungu, A., Buyuktosunoglu, A. and Sorin, D.: Dynamic power gating with quality guarantees, *Proc. 14th ACM/IEEE international symposium on Low power electronics and design*, pp.377–382, ACM, New York, NY, USA (2009).
[12] Court, P.K.C.A.: Loop-Directed Mothballing: Power Gating Execution Units Using Runtime Loop Analysis, *IEEE Micro*, Vol.31, No.6, pp.29–38 (2011).
[13] Maro, R. and Bai, R.Y.: Dynamically Reconfiguring Processor Resources to Reduce Power Consumption in High-Performance Processors, *Proc. 1st Int'l Workshop Power Aware Computer Systems*, pp.97–111 (2001).

**石井 義史**

2012 年慶應義塾大学大学院理工学研究科修士課程修了．現在，ルネサスエレクトロニクス（株）に所属．

**王 蔚涵**

2011 年慶應義塾大学理工学部情報工学科卒業．現在，同大学大学院修士課程．低電力アーキテクチャの研究に従事．



**天野 英晴**　（正会員）

1958 年生．1981 年慶應義塾大学理工学部電気工学科卒業．1986 年同大学大学院理工学研究科電気工学専攻博士課程修了．現在，同大学理工学部情報工学科教授．工学博士．計算機アーキテクチャの研究に従事．