

## 細粒度 Power Gating を適用した VLIW 型プロセッサの 実装と評価

石井 義史<sup>†1</sup>      池淵大輔<sup>†1</sup>      天野 英 晴<sup>†1</sup>

VLIW 型プロセッサの演算 SLOT に対して細粒度 Power Gating(PG) を適用し、動的に複数の演算ユニットの sleep/active 状態を切り替えることで漏れ電流を削減し、低消費電力でありながら高性能化を実現する方法を提案する。さらに、細粒度 PG に加えて、粗粒度 PG を VLIW 型プロセッサ内部の SLOT に適用し、細粒度 PG による面積オーバーヘッドの削減を行う。5 SLOT で構成される VLIW 型プロセッサ構成に対して、2 SLOT に粗粒度 PG を、2 SLOT に細粒度 PG を適用し、その効果を 3 つのアプリケーションで評価した。その結果、常温 (25 °C) において、全演算ユニットを sleep させない場合と比較して、5 SLOT 全てを用いる high performance mode 時では 12.6%-29.4%、粗粒度 PG を用いた 2 SLOT を sleep にする low performance mode 時では 29.2%-41.1%、漏れ電力を削減することができた。性能面では、high performance mode 時では、1SLOT で動作する場合に比べ、2.15 倍-2.38 倍、low performance mode 時では、1.19 倍-1.79 倍の性能向上を実現した。

### A VLIW processor with fine-grained power gating

YOSHIFUMI ISHII,<sup>†1</sup> DAISUKE IKEBUCHI<sup>†1</sup>  
and HIDEHARU AMANO<sup>†1</sup>

By switching the sleep mode and active mode dynamically using the fine-grained Power Gating (PG) in computational units of VLIW (Very Long Instruction Word) processor, the leakage current can be suppressed keeping relatively high performance. In order to reduce the overhead of fine-grained PG, a coarse-grained PG is used in a part of computational slots. Here, a VLIW processor with 5 slots is designed, and the coarse-grained PG is applied to 2 slots and fine-grained PG is used for other 2 slots. The leakage current is reduced by 12.6%-29.4% in the high-performance mode using all slots (fine-grained PG is used in 2 slots) compared with the case when all slots are activated. In the low performance mode in which 2 coarse-grained PG slots are always in sleep mode, 29.2%-31.1% of leakage power is reduced. Compared with execution with 1 slot, 2.15-2.38 and 1.19-1.79 times performance improvement is achieved in the high

performance mode and low performance mode, respectively.

#### 1. はじめに

LSI チップの漏れ電流は、90nm 以降、プロセスの進展に伴って急速に増大し、この削減手法は重要な問題となっている。CPU の設計においては、高速だが漏れ電流の大きい低スレッシュホールドレベルのトランジスタをクリティカルパスに利用し、それ以外の部分に漏れ電流の小さい高スレッシュホールドレベルのトランジスタを利用する方法が用いられている<sup>1)</sup>。しかし、CPU の設計では、クリティカルパスに準ずるパスが多いため、漏れ電力削減の効果は十分ではない。

他にも性能低下を伴わず、漏れ電流を削減する手法にパワーゲーティングがある。この手法では、漏れ電流の少ない高スレッシュホールドのトランジスタをパワースイッチとして利用し、利用されていない部分の電源を切り離してスリープ状態にすることにより漏れ電流を削減する。動作対象の回路自体は、通常のスレッシュホールドレベルの高速トランジスタを利用できるため、速度の低下はほとんどない。一方で、回路のスリープ時には、高いスレッシュホールドのパワースイッチがオフとなり、漏れ電流を遮断する。しかし、従来、パワーゲーティングは、アクセラレータ、I/O モジュールやメモリなど、半導体の大きな領域が対象であり、スリープ状態から回復 (ウェイクアップ) するまでに多大な時間を要した。このため、CPU など常に動作しているデバイスに適用することは困難であった。

近年、小さな回路領域に対して高速にスリープおよび回復操作を行うことのできる細粒度パワーゲーティング (細粒度 PG) の研究が進んでおり<sup>2)</sup>、この手法を演算ユニットに対して用いた CPU Geyser<sup>3)4)</sup> が実装されている。汎用 CPU である Geyser は演算ユニットは、加減算を行う CLU、シフトユニット、乗算器、除算器の 4 つを対象としていた。しかし、この手法は、多くの演算ユニットを持つ VLIW 型 CPU に適用した場合、より効果的であることが予想される。そこで、本報告では、Geyser で用いた細粒度 PG を VLIW 型プロセッサへ適用し、その効果を評価する。

<sup>†1</sup> 慶應義塾大学 理工学部  
Faculty of Science and Technology, Keio University

## 2. 細粒度動的 PG

### 2.1 細粒度 PG の実現手法

従来のパワーゲーティングはコアやモジュールなどチップ上の大きな面積を単位としていた<sup>5)6)</sup>。しかし、大きな単位で電源をオン、オフするため、電源がオフ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、Geysler では、図 1 に示すように、それぞれのゲート毎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、仮想グラウンドを持ったパワーゲーティング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シャットダウン時間と動作モードに遷移するウェイクアップ時間をナノ秒単位に短縮することができる。この技術を用いれば、演算器レベルの細かい単位で、パワーゲーティングが可能になる。この技術を従来のパワーゲーティングと区別して細粒度パワーゲーティング (細粒度 PG) と呼ぶ。

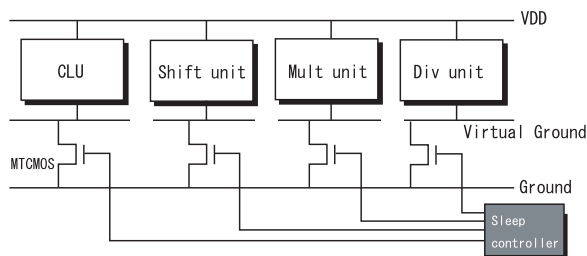


図 1 パワーゲーティング回路の概要

図 2 は細粒度 PG における実際のセル配置を示している。従来のパワーゲーティングとは異なり、一つの仮想グラウンドラインを数個程度のセルが共有している。一つグラウンドラインとスリープトランジスタの集合で一つのユニットが構成される。ユニットは、単一のスリープ制御信号でスリープモードと動作モードの切り替えが制御される。高速なモード切り替えのためには、スリープトランジスタの数と配置を適切に行う必要があり、これについては Pinnacle<sup>7)</sup> および Geysler-0 の設計、実装により、その回路上の設計手法がほぼ確立している。

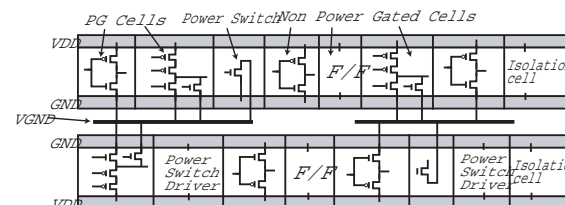


図 2 細粒度 PG の仮想グラウンドアーキテクチャ

パワーゲーティングはモード遷移に一定のエネルギーが必要である。また、スリープモードに入ってもリーク電流を遮断するまでには、一定の時間がかかる。このため、モード遷移の頻度が大きいと、逆にエネルギーロスによって消費電力が大きくなってしまいう可能性がある。スリープモードに遷移して一定期間経過後、エネルギーロスよりも、削減したエネルギーの方が大きくなる地点をブレイクイーブンポイントと呼ぶ。ブレイクイーブンポイントを超えてスリープを継続できれば消費電力を削減できていることになる。

Geysler では、MIPS R3000 互換の 5 段パイプラインを装備しており、その実行 (EX) ステージの演算ユニットを対象として、細粒度パワーゲーティングを実装している。命令フェッチユニットに簡易命令デコーダが装備されており、これにより必要な演算ユニットがウェイクアップされる。ID ステージを命令が通過している間で、各ユニットはウェイクアップされ、演算命令を実行後、すみやかにスリープ状態に移行する。

演算器の中で利用頻度が高いものは、ブレイクイーブンポイントに達せずにモード遷移が頻発して、エネルギーロスを大きくしてしまう可能性がある。このような状況では、スリープに移行せず、アクティブな状態をキープの方がエネルギー効率が良い。ブレイクイーブンポイントは、温度に敏感であり、このような制御はハードウェアによる自動的なパワーゲーティングの制御では実現できない。そこで、Geysler ではソフトウェア制御によって、ケースバイケースのパワーゲーティング制御を行うため、PG 制御情報付き命令と、PG 制御レジスタの二つの機構を持っている。

## 3. VLIW 型プロセッサ Geysler-VLIW の設計

本節では、Geysler を VLIW 型プロセッサに拡張した Geysler-VLIW の設計について述べる。まず、Geysler-VLIW で実行される命令形式について述べた後、プロセッサ本体の内部構造について説明し、最後に Geysler-VLIW の Sleep 機構について説明する。

3.1 命令形式

Geyser-VLIW の命令は, SLOT と呼ばれる, 32bit で構成される短い命令を 5 つつなげた, 160bit で構成される. 各 SLOT に四則演算命令や論理演算, 分岐命令をいれることでこれらの演算を実行できる. 今回設計した Geyser-VLIW には, 5 つの SLOT が存在し, それぞれに実行できる命令が以下のように決まっている.

- SLOT 0: ALU 演算, シフト演算, 除算, branch/jump 命令
- SLOT 1: ALU 演算, 乗算
- SLOT 2: ALU 演算, 乗算
- SLOT 3: メモリに対するデータの Read/Write
- SLOT 4: メモリに対するデータの Read/Write

対応する SLOT に入れる命令は, 上記で示した, 各 SLOT での実行可能命令と対応している必要がある.

各 SLOT が実行する 32bit の命令フォーマットは, MIPS R3000<sup>8)</sup> の命令セットと同じ形式を採用した. 演算を実行すると自動的に細粒度 PG によりスリープ状態になる Geyser と異なり, Geyser-VLIW では, PG 用の命令を実行することによって, SLOT をスリープ状態にする.

3.2 Geyser-VLIW の内部構造

図 3 に, Geyser-VLIW の内部構造を示す.

Geyser-VLIW は, 5 つの SLOT Pipeline で構成され, Program Counter(PC) は SLOT0 に実装されている. そのため, instruction cache(icache) から fetch してくる命令のアドレスの計算は SLOT0 で行われる.

SLOT1-2 は 4 段 pipeline 構成の演算専用 SLOT であり, 演算ユニットとして ALU と Multiplier を持っている. Multiplier は 3 ステージ構成で pipeline 化されている. このため, 同一スロットで続けて乗算を実行することができる.

SLOT3-4 は 5 段 pipeline 構成のメモリアクセス専用 SLOT であり, メモリに対するデータの読み書き (load 命令/store 命令) が実行できる. execute stage においてアクセスするメモリのアドレス計算を行い, memory read/write stage において外部にある data cache(dcache) にアクセスし, データのやりとりを行う.

以上の構成により 2 つのメモリアクセス SLOT を用いて, 1 度に 2 つの data を読み出し, 2 つの演算専用 SLOT で演算を行い, 演算結果を 2 つのメモリアクセス SLOT によってメモリに書き戻すことができる. この構成は, 画像処理をはじめとするマルチメディア処理の

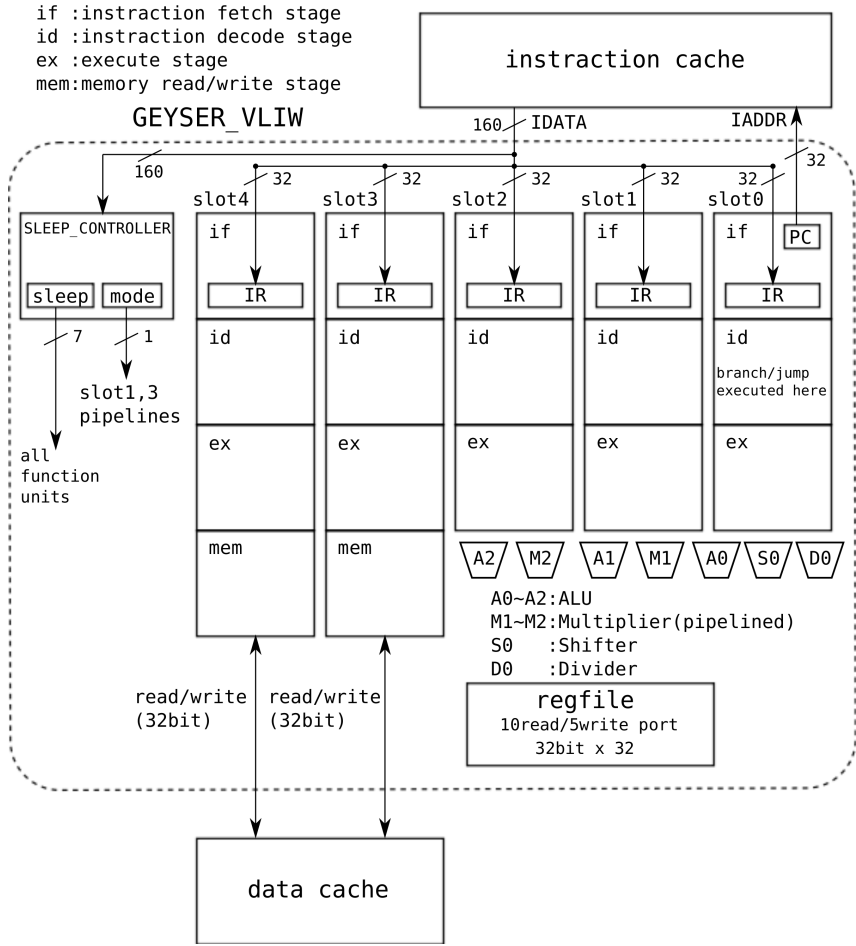


図 3 Geyser-VLIW の内部構造

高速化を主に念頭に置いている.

3.3 Sleep 機構

Geyser-VLIW の Sleep 機構について述べる前に, 本プロセッサで用いられている, 細粒度 PG (fine-grained PG, FG-PG) と粗粒度 PG (coarse-grained PG, CG-PG) にそれぞれ

について、長所と短所を述べる。その後、本論文で提案する、Power Gating の、FG-PG と CG-PG の組み合わせた新手法について説明する。

FG-PG は power switch を gate level で挿入して sleep 可能な機構を構成するため、power switch の切り替えの細かい制御が可能であり、Break Even Time よりも長い時間 sleep させることができれば、その差分だけ電力を得ることができる。しかし、power switch の挿入による面積オーバーヘッドが存在するため、FG-PG を実装することによってチップ面積の増大を招く可能性がある。

CG-PG は、仮想電源 (またはグラウンド) リングを設け、PG 対象となる回路の電源 (またはグラウンド) の供給元を仮想電源 (またはグラウンド) リングに変更し、仮想電源 (またはグラウンド) リングと電源 (またはグラウンド) リングとの間に power switch を挿入し、sleep 可能な機構を構成する。回路内部に power switch を挿入しないため、面積のオーバーヘッドは小さい。しかし、CG-PG の場合、power switch の on/off を切り替える電力オーバーヘッドが大きく、スリープ、アクティブのモード変換にも時間が掛かるため、細かい時間間隔での power switch の切り替えには向かない。

VLIW 型プロセッサは pipeline が複数存在し、複数の演算ユニットを持つため、実装した場合、演算ユニット部の面積が大きくなりやすい。リーク削減を行うために、FG-PG のみで Sleep 機構を実装してしまうと、更なる面積の増加を招く恐れがある。そこで、VLIW 型プロセッサにおいて、FG-PG と面積オーバーヘッドの小さい CG-PG を組み合わせて実装する手法を考案した。具体的には、FG-PG を SLOT0, SLOT2 の 2 つの演算ユニットに適用し、CG-PG を SLOT1, SLOT3 に適用した。それぞれの sleep 制御は、図 3 で示したように、Sleep Controller によって制御される。

Sleep 機構について、FG-PG が適用された演算ユニットの sleep 制御は Sleep Controller の出力線 sleep によって行われる。CG-PG が適用された SLOT に関しては、動作モードという概念を考え、Sleep Controller の出力線 mode によって power switch の制御が行われる。

3.3.1 各 SLOT における PG の適用

SLOT0, SLOT2 は、Geysler と同様に FG-PG を用いている。したがって、各演算ユニットの sleep 制御を行うための power switch がそれぞれの演算ユニットとグラウンド (GND) に挿入されており、この power switch の入力に Sleep Controller の sleep 出力線に接続されている。SLOT0, SLOT2 は sleep 命令で演算ユニットを指定して、スリープ状態に遷移させる。スリープした演算ユニットを利用する場合、その演算ユニットに対してあらかじめ active

命令を実行して、ウェイクアップさせる必要がある。

一方、SLOT1, 3 では、SLOT 全体に対して仮想 GND リングを設け、GND リングとの間に power switch を設けることによって、CG-PG を実現している。power switch の入力に Sleep Controller の mode 出力線を接続することで、SLOT1, 3 の sleep 制御を行う。

Geysler-VLIW には、動作モードと呼ばれる概念が存在する。動作モードには 2 種類あり、High Performance Mode(HP モード) と Low Performance Mode(LP モード) がある。

HP モードでは、全 SLOT がアクティブ状態になっている。このとき、Sleep Controller から出力線 mode を通して、VGND リングと接続されている power switch の入力には、High が入力されており、全 5SLOT で演算が可能である。実行プログラムの命令並列性が大きく、高速に実行させる場合、sleep\_mode 命令を SLOT0 で実行することで、この動作モードに移行できる。

LP モードでは、SLOT1, SLOT3 の sleep 状態になっており、SLOT0, SLOT2, SLOT4 のみ命令実行が可能な状態となる。高速に動作させる必要のない場合に SLOT0 において、sleep\_mode 命令を実行することで、この動作モードに以降し、プロセッサ全体のリーク電力を抑えることができる。

4. Simulation 評価

まず初めに評価環境について説明する。その次に、その評価環境を用いることで得た結果を、電力、性能それぞれに関して報告する。

4.1 評価環境

まず、図 3 を元に、RTL の Geysler-VLIW のプロセッサ及び Test Bench のコードを Verilog で記述し、VLIW 用に作成したプログラム (アプリケーション) を NCVerilog 上でシミュレーション実行可能な環境を構築した。

VLIW 用に作成したプログラムは、MIPS のアセンブラ言語を用いて記述される。以下にその一例を示す。

```

nop          | nop          | ori $18, $0, 124 | ori $17, $0, 0   | ori $16, $0, 0x1234
loop: lw      $8, 0($17) | lw      $9, 4($17) | nop            | nop            | nop
          lw      $10, 8($17) | lw      $11, 12($17) | nop           | nop           | nop
          lw      $12, 16($17) | lw      $13, 20($17) | add $24, $16, $8 | add $25, $16, $9 | nop
          lw      $14, 24($17) | lw      $15, 28($17) | add $26, $16, $10 | add $27, $16, $11 | nop
          sw      $24, 0($17) | sw      $25, 4($17) | add $28, $16, $12 | add $29, $16, $13 | nop
          sw      $26, 8($17) | sw      $27, 12($17) | add $30, $16, $14 | add $31, $16, $15 | nop
          sw      $28, 16($17) | sw      $29, 20($17) | addi $18, $18, -1 | addi $17, $17, 32 | nop
          sw      $30, 24($17) | sw      $31, 28($17) | nop            | nop            | bne $18, $0, loop
          nop          | nop          | nop            | nop            | nop

```

SLOT ごとに行われる命令は 'I' で区切られ、左から順に SLOT4, 3, 2, 1, 0 で実行される

命令に対応する。

上記のような形式によって、評価アプリケーションとして DCT(離散コサイン変換), Alpha Blenda, Sepia を作成した。DCT では, greyscale の 8x8 の計 64data の計算を行う。Alpha Blenda では, greyscale 画像 (8x8) の二つの入力画像を用いて計算を行う。Sepia では, 24bit の RGB data を持った 8pixel 分の計算を行った。なお, アプリを作成した後, sleep 命令, sleep\_mode 命令

表 1 25 における各演算ユニットの BreakEvenTime (1cycle:5ns)

Unit	ALU	Shifter	Multipilier	Divider
BET [cycle]	136	180	111	80

#### 4.2 電力評価結果

Geysner-1 の演算ユニット (ALU, Shifter, Multiplier, Divider) のリークデータを元に, Geysner-VLIW のリーク電力比を導出した。その結果をアプリケーションごとに表 2 から表 2 に示す。

リーク電力比は, 全機能ユニットが active であった場合のリーク電力を 100%とした時に, sleep 命令によって BET が得られる演算ユニットを sleep させた時のリーク電力の大きさの割合を表す。

表 2 DCT (high performance mode, 実行時間:524 cycle)

Unit	25	65	100	125
slot2.alu	100%	100%	100%	100%
slot2.mult	100%	100%	100%	100%
slot1.alu	100%	100%	100%	100%
slot1.mult	100%	100%	100%	100%
slot0.alu	100%	100%	100%	100%
slot0.shift	100%	100%	100%	100%
slot0.div	47.3%	43.0%	19.2%	11.7%

次に, 各アプリケーションにおいて, 機能ユニット全体として, どの程度リーク電力を削減できたかを知るために, 常温 (25 ) における全演算ユニットでのリーク電力削減率を求めた。その結果を表 8 に示す。

表 8 により, HP モードにおいて最大 27.2%, LP モードにおいて最大 41.1%, 全機能ユニットのリーク電力を削減できることがわかった。

表 3 DCT (low performance mode, 実行時間:949 cycle)

Unit	25	65	100	125
slot2.alu	100%	100%	100%	100%
slot2.mult	100%	100%	100%	100%
slot1.alu	35.4%	18.5%	9.04%	6.19%
slot1.mult	36.2%	23.7%	12.0%	7.81%
slot0.alu	35.4%	18.5%	9.04%	6.19%
slot0.shift	100%	100%	100%	100%
slot0.div	52.0%	25.3%	11.1%	2.16%

表 4 Alpha Blenda (high performance mode, 実行時間:194 cycle)

Unit	25	65	100	125
slot2.alu	100%	100%	100%	100%
slot2.mult	100%	100%	100%	100%
slot1.alu	100%	100%	100%	100%
slot1.mult	100%	100%	100%	100%
slot0.alu	100%	100%	100%	100%
slot0.shift	100%	100%	100%	100%
slot0.div	76.1%	60.4%	55.2%	38.7%

表 5 Alpha Blenda (low performance mode, 実行時間:315 cycle)

Unit	25	65	100	125
slot2.alu	100%	100%	100%	100%
slot2.mult	100%	100%	100%	100%
slot1.alu	63.0%	35.2%	22.1%	14.3%
slot1.mult	62.1%	35.5%	27.0%	18.8%
slot0.alu	100%	100%	100%	100%
slot0.shift	100%	100%	100%	100%
slot0.div	57.3%	62.6%	31.2%	18.9%

#### 4.3 性能評価結果

それぞれのアプリケーションをシミュレーション上で実行し, 実行完了までにかかったサイクル数を調べ, 性能評価を行った。その結果を, 表 9 に示す。

表 9 により, HP モードで最大 2.38 倍, LP モードで最大 1.79 倍, 性能向上を実現できたことがわかった。1

表 6 Sepia (high performance mode, 実行時間:131 cycle)

Unit	25	65	100	125
slot2.alu	100%	100%	100%	100%
slot2.mult	100%	100%	100%	100%
slot1.alu	100%	100%	100%	100%
slot1.mult	100%	100%	100%	100%
slot0.alu	100%	100%	100%	100%
slot0.shift	100%	100%	100%	100%
slot0.div	76.1%	60.4%	55.2%	38.7%

表 7 Sepia (low performance mode, 実行時間:174 cycle)

Unit	25	65	100	125
slot2.alu	100%	100%	100%	100%
slot2.mult	100%	100%	100%	100%
slot1.alu	91.6%	51.5%	34.5%	27.9%
slot1.mult	87.8%	50.3%	35.8%	30.9%
slot0.alu	100%	100%	100%	100%
slot0.shift	100%	100%	100%	100%
slot0.div	79.3%	59.7%	59.5%	42.3%

表 8 25 における全演算ユニットでのリーク電力削減率

Mode	DCT	Alpha Blenda	Sepia
high	12.6%	27.2%	24.7%
low	29.2%	41.1%	29.4%

表 9 性能改善率

	Geysers-1	Geysers-VLIW (high)	改善率 (high)	Geysers-VLIW (low)	改善率 (low)
DCT	1129 cycle	524 cycle	2.15	949 cycle	1.19
Alpha Blenda	427 cycle	194 cycle	2.20	315 cycle	1.36
Sepia	312 cycle	131 cycle	2.38	174 cycle	1.79

5. ま と め

演算ユニット数の多い VLIW 型プロセッサに注目し、その演算ユニットに対して細粒度 Power Gating を適用し、その効果を評価した。細粒度 Power Gating では、細かい時間間隔で sleep の制御が行うことが可能なため、Break Even Time 以上 sleep させることができ

れば、その差分だけリーク電力を削減することが可能である。しかし、この手法には、power switch の挿入による面積オーバーヘッドが存在するため、多くの回路に細粒度 Power Gating を行うとチップ面積の増大を招く可能性がある。そこで、面積のオーバーヘッドの小さい、Power Gating の別の手法である粗粒度 Power Gating と組み合わせる方法を提案した。今回は、5SLOT 構成の VLIW 型プロセッサ Geysers-VLIW を考案し、実装を行った。全 5SLOT 中、2SLOT に対し細粒度 Power Gating を、別の 2SLOT に対し粗粒度 Power Gating をそれぞれ適用した。細粒度 Power Gating を適用した 2SLOT が持っている機能ユニットは、sleep 命令によって粗粒度 Power Gating を適用した 2SLOT は sleep\_mode 命令によって、sleep/active 状態の切り替えが実行される。

Geysers-VLIW で 3 つのアプリケーションをモード別に行なった結果、全機能ユニットのリーク電力を、high-performance mode 時に最大で 27.2%、low-performance mode で最大 41.1%削減することができた。また、性能に関して、実行サイクル時間で Geysers-1 と比較した結果、high-performance mode 時に最大で 2.38 倍、low-performance mode 時に最大で 1.79 倍、性能向上が実現できた。

これらの結果から、VLIW 型プロセッサに対し、Power Gating を適用することで、リーク電力を抑えつつ、高い性能を持つプロセッサを実現することができる。

今後は、Geysers-VLIW の論理合成に基づく評価を行うのに加えて、細粒度 Power Gating と粗粒度 Power Gating の面積オーバーヘッドを定量的に比較し、この両者を組み合わせたことの利点を評価して行きたい。

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。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を通じ、シノプシス株式会社・日本ケイデンス株式会社の協力で行われた。

参 考 文 献

- 1) M.Keating, D.F. and et. al.: *Low Power Methodology Manual: For System-on-Chip Design*, New York: Springer (2007).
- 2) T.Kashima, K.Usami and et. al.: Architecture Design for Low-Power Multiplier applying Run time Power Gating (in Japanese), *VLD*, Vol.73, pp.7-12 (2006).
- 3) T.Kashima, K.Usami and et. al.: Architecture Design for Low-Power Multiplier applying Run time Power Gating (in Japanese), *VLD*, Vol.73, pp.7-12 (2006).
- 4) T.Kashima, K.Usami and et. al.: Architecture Design for Low-Power Multiplier applying Run time Power Gating (in Japanese), *VLD*, Vol.73, pp.7-12 (2006).
- 5) M.Ishikawa and et. al.: A 4500 MIPS/W, 86μA Resume-Standby, 11μA Ultra-Standby

- Application Processor for 3G Cellular Phones, *IEICE Trans. on Electronics*, Vol.E88-C No.4, pp.528–535 (2005).
- 6) Y.Kanno: Hierarchical Power Distribution with 20 Power Domains in 90-nm Low-Power Multi-CPU Processor, *ISSCC2006*, Vol.9999???, pp.540–541 (2006).
  - 7) Usami, K. and Ohkubo, N.: A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals, *Proceedings of the International Conference on Computer Design (IC CD'06)* (2006).
  - 8) Gerry Kane: *mips R2000/3000*, Prentice Hall (1987).