

2K-3

Just-In-Time コンパイラを用いた動的解析に基づく細粒度 PG 制御

和田 基† 佐藤 未来子† 天野 英晴‡ 近藤 正彰†† 中村 宏†† 並木 美太郎†
 † 東京農工大学 ‡ 慶応義塾大学 †† 東京大学

1 はじめに

回路技術の微細化に伴い、プロセッサの消費電力に占めるリーク電力の割合は増加傾向にある。パワーゲーティング技術 (PG 技術) を用いることで、このリーク電力を削減することが可能である。

本研究で用いる『Geyser』プロセッサは、この PG 技術を細粒度に施しており、4 つの演算器 (ALU, Shift, Mult, Div) への電力供給を 1 クロック単位で制御することができる。しかし、電力供給の遮断・開始時には通常より大きなオーバーヘッド電力が発生するため、短い期間の電源遮断 (スリープ) は消費電力を増加させてしまう。電源遮断によって得られる省電力効果とオーバーヘッド電力が釣り合うスリープ長を、損益分岐点 (Break Even Point) と定義する。BEP より短い時間単位でのスリープは消費電力を増加させるため、BEP 未満の時間のスリープが発生しないよう PG を制御することが大事であるが、課題として BEP の高い温度依存性がある。

我々の先行研究では、コンパイラによってプログラムの事前解析を行い、PG 制御コードを含むオブジェクトファイルを生成する手法が提案された [1]。しかし事前解析では、条件分岐の偏りやチップ温度などの動的な要因を考慮した PG 制御を行うことは困難である。

そこで筆者では、JIT コンパイラを持つインタプリタ型言語処理系を対象とした、事前解析を必要とせず、実行時に動的な要因を考慮した解析処理を行う PG 制御手法を提案した [2]。本原稿では、その更なる考察についてまとめ、今後の展望につなげる。

2 Geyser の PG 制御インタフェース

Geyser プロセッサは MIPS をベースとして開発されたプロセッサである。命令セットとして MIPS I を採用しており、Geyser ではここに R 形式命令の拡張として PG 制御命令が追加されている。

MIPS の定める R 形式命令は、op コードとして (000000)₂ を持つ。Geyser プロセッサでこの命令を実行すると、パイプライン中 ID ステージで使用演算器を判定、電力供給を開始し、EXE ステージが終了すると供給を停止す

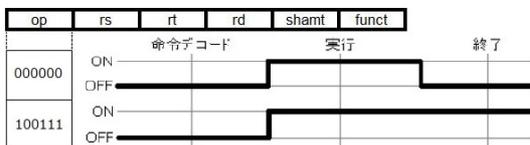


図 1: PG 制御命令実行時の電源遷移

る。しかし、op コードを専用の PG 制御ビット (100111)₂ にすることで、本来の演算を行った後、図 1 に示すよう電源供給を継続しアクティブ状態を維持することができる。Geyser においては、この PG 制御命令をオブジェクトコード中の R 形式命令と置き換えることによって、BEP 未満のスリープを抑えるよう PG を制御する。

3 提案した PG 制御手法

3.1 全体構成

インタプリタ型言語処理系が JIT コンパイラを持つとき、プログラムの実行は、ソースコード (もしくは中間コード) の逐次解釈処理による実行と JIT コンパイラの生成したネイティブコード (以後 JIT コード) の実行を切り替えながら行われる。提案手法では JIT コード中に 2 章で述べた PG 制御命令を配置し JIT コード実行時の消費電力を抑えることで、システム全体の省電力化を図る。

提案手法の全体構成を図 2 に示す。PG 制御は、JIT コンパイル直後に行われる事前解析処理と、JIT コードの実行直前に行われる実行直前処理の 2 段階の処理で行われる。

3.2 事前解析処理

JIT コードを予測実行順に走査し、各演算器の使用間隔 (アイドルサイクル) を命令数単位で解析する。分岐命令がある場合は、分岐先を双方に等確率で振る。

3.3 実行直前処理

事前解析処理で解析したアイドルサイクルと現在のチップ温度から算出された BEP とを比較し、BEP が下回る位置に PG 制御命令を配置して電力的に不利となるスリープを抑制する。

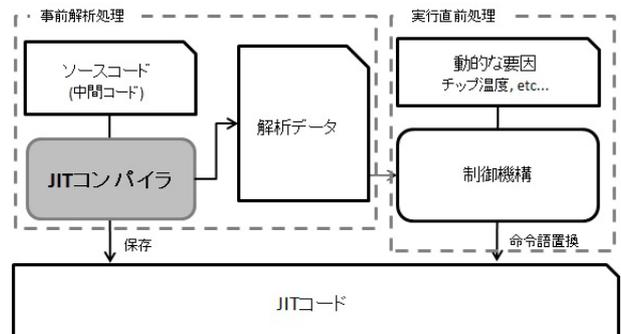


図 2: 提案手法の全体構成

4 評価実験と考察

4.1 評価方法

提案手法を実装した Dalvik VM を AndroidEmulator (QEMU) 上で動作させ、ベンチマーク実行による評価実験を行った。プロセッサアーキテクチャは MIPS32rev2 を用い、Geysers の PG モデルに対応するように PG 制御命令と命令列トレースによるスリープシミュレーション機構を追加した。シミュレーションによって得られたスリープ分布より、平均リーク電力を算出した [3]。また、総実行命令数に占める JIT コードの命令数 (JIT コード実行率) も同時に計測した。また、チップ温度として 25 °C, 65 °C, 100 °C, 125 °C の 4 パターンの固定値を用いてそれぞれ実験を行った。

4.2 CaffeineMark v3 による評価

CaffeineMark v3 より Sieve, Method, Loop ベンチマークを用いて評価実験を行った結果 (図 3), 平均 6%, また 100 °C における Sieve で最大 22% の平均リーク電力削減に成功した。Sieve と Loop では平均リーク電力の削減に成功したが、Method は全温度で平均リーク電力が増加した。アイドルサイクルの解析値が実際に発生したアイドルサイクルよりも短く、さらに両者の間に BEP がある場合、実際には得となるはずのスリープを誤って抑制し平均リーク電力を増加させてしまう。JIT コード中に分岐命令が多いと確率的な要素が増えるため、アイドルサイクルの解析精度が落ちてしまう。チップ温度以外の要因を考慮しアイドルサイクル解析の精度を高めることで、Method においても省電力効果が見込める。

4.3 SPECJVM2008 による評価

SPECJVM2008 より, compress と crypto/rsa ベンチマークを用いた評価実験を行った。図 4 に平均リーク電力と JIT コード実行率を示す。compress では最大 6.3%, 平均 2.2% の平均リーク電力を削減できたが、crypto/rsa では全ての温度下で平均リーク電力が平均 11.1% 増加する結果となった。双方について、JIT コード実行率はそれぞれ 24%, 3.0% となった。

compress では平均リーク電力の大半が ALU によるものだが、crypto/rsa においては Mult の平均リーク電力が

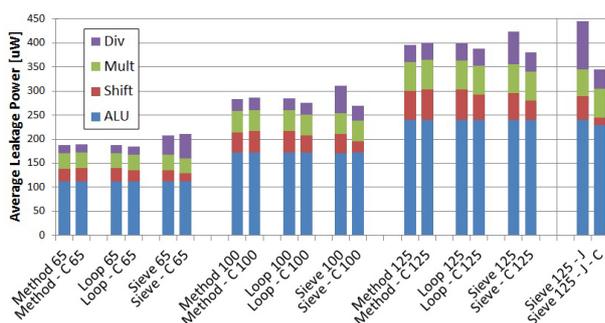


図 3: CaffeineMark による評価結果

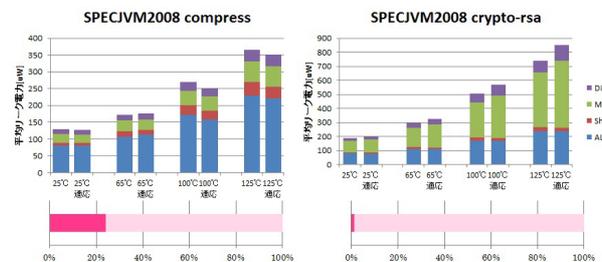


図 4: SPECJVM2008 による評価結果

主要項となっている。crypto/rsa では多倍長演算の乗算を多く行い非常に短い間隔で Mult が使用されるので、BEP を下回るスリープが多発してしまうことが原因である。crypto/rsa の JIT コード実行率は低く、しかも JIT コンパイルの対象となったクラスのほぼ全てが Java クラスライブラリ内のクラスであったため、多倍長演算の処理は JIT コンパイルの対象になっておらず、提案手法で消費電力を削減することは不可能である。従来の事前解析を行う手法 [1] と本提案手法を組み合わせることで、crypto/rsa のように JIT コード実行率が低いベンチマークにおいても省電力化を実現することができる。

また、JIT コード実行率が低いことは提案手法が関与できる時間が短いことを示唆しており、直感的には全体に与える影響も小さくなると考えられる。しかし本実験では、JIT コード実行率の低い crypto/rsa がより大きな影響を全体に与えている。JIT コードの実行回数や長さなどの実行率以外の要因が関与している可能性がある。

5 おわりに

本研究では JIT コンパイラに着目し、解析処理を実行時に行うことによって動的な要因を考慮可能な PG 制御を Dalvik VM 上で実現した。また CaffeineMark, SPECJVM2008 による評価実験において、本手法の有効性を示すとともに、今後の課題を明らかにした。

今後の展望として、チップ温度以外の動的な要因を考慮するよう提案手法を改良し評価実験を行う。また Geysers 実プロセッサ上に DalvikVM を移植し、実際の消費電力評価を行いたい。

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機構「CREST」における研究領域「革新的電源制御による次世代超低消費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] 薦田登志矢, 佐々木宏, 近藤正章, 中村宏, “リーク電力削減のためのコンパイラによる細粒度スリープ制御”, SACSIS 2009, pp.11-18 (May 2009).
- [2] Motoki Wada, Jun Tsukamoto, et al. “Dalvik VM JIT Compiler for Fine-Grained Power Gating Control, In the Proceedings of COOL Chips XVI, 2013.
- [3] 中田光貴, 白井利明, 香嶋俊裕, 武田清大, 宇佐美公良, 関直臣, 長谷川揚平, 天野英晴, “ランタイムパワーゲーティングを適用した回路での検証環境と電力見積もり手法の構築”, 電子情報通信学会信学技報 VLD2007-111, pp.37-42 (Jan 2008).