

拡張容易な CMP シミュレータの電力評価環境構築

佐藤 裕輔[†] 大津 金光[†] 横田 隆史[†] 馬場 敬信[†]

[†]宇都宮大学工学部情報工学科

1 はじめに

近年のプロセッサはマルチコア化が進み、1つのチップ内に多くのハードウェアが凝縮している。並列処理技術の発展により、この豊富なハードウェア資源も有効活用ができるようになりつつある。その一方で電力消費量は増大し、発熱が問題となっているし、モバイル PC や携帯家電などでの長時間稼働への要望もあり、マルチコアプロセッサにおける省電力化技術への需要が高まっている。

アーキテクチャレベルでの省電力化技術研究には、シミュレーションによる事前性能予測が必須だが、現在の CMP (Chip Multi Processor) 電力シミュレータではシミュレータが想定しているハードウェアモデルに変更があった場合にプログラムコードを大幅に書き換えなければならない場合がある。この問題を解消し、CMP におけるアーキテクチャレベルでの省電力化技術を研究するためには、柔軟性があり、あらゆる構成のシステムを構築することができるシミュレータが必要となる。

本稿ではオブジェクト指向言語の基本的な考え方であるクラス化を活用して、拡張性の高い CMP 電力シミュレータの開発を行い、シミュレーション精度の検証を行うことで、省電力化技術研究のための基盤を構築する。

2 電力見積り手法

2.1 ソフトウェアによる電力算出方法

電力を計測する手法は実機計測およびソフトウェアシミュレーションの2種類に分類することができるが、本研究では事前性能予測の手軽さおよび柔軟性の側面からソフトウェアシミュレーションによる手法を前提としている。ソフトウェアシミュレーションによる計測手法では以下の式により消費電力を算出している。

$$P_{active} = \alpha \cdot C_{load} \cdot f \cdot V_{dd}^2 \quad (1)$$

α : 稼働率

C_{load} : 負荷容量

f : クロック周波数

V_{dd} : 電源電圧

2.2 開発基盤の検討

拡張性の高い CMP 電力シミュレータを開発するにあたって、既存のシステムで流用できる部分を検討する。既存のシミュレータを活用することで開発期間を短縮できるというメリットもある。また、本システムに求められることとしては以下の点が挙げられる。

- 近年主流である CMP に対応したマルチコアプロセッサシミュレーションが可能であること
- 省電力化機構検討のために、電力計測機能を搭載していること
- 年々複雑化するハードウェアシステム構成に対応すべく高い拡張性を備えていること

2.2.1 既存シミュレータの特徴

既存のシステムとして、電力計測機能を搭載しているもの、CMP シミュレーションに対応しているものを拡張性の高さに着目しながら特徴を述べる。

Wattch[1] : SimpleScalar[2] をベースに開発された電力計測機能を搭載したシングルプロセッサシミュレータ。SimpleScalar のパッチとして提供されている。数式 1 を元に消費電力を算出する。 C_{load} の部分の算出方法として、4 種類のカテゴリに分類することが大きな特徴である。CMP シミュレーションに対応させるためには大幅な修正が必要となるためシミュレータ自体の拡張性はあまり高くはない。電力算出の方法としては文献 [1] 内で実機計測時との誤差が 10% 以内であるとの記述があり、優秀であるといえる。

また、SimpleScalar とは、近年のスーパースカラプロセッサが行う out-of-order 実行や分岐予測をサポートしたシングルプロセッサシミュレータであり、実験、検証等に幅広く用いられている。

ISIS-SimpleScalar[3] : SimpleScalar をベースに開発されたマルチコアプロセッサシステム用のクラスライブラリ。あらゆるシステム構成を再現することを想定した ISIS のシステムとの接続が可能であるため、拡張性は高いが、電力計測機能は搭載していない。

SESC[4] : 電力モデルを搭載した CMP シミュレータ。温度見積り機能をはじめとして非常に多くの機能を実装しているため、内部構造が複雑になっている。

2.2.2 基盤シミュレータの決定

本稿では、ISIS-SimpleScalar をベースにして Wattch を参考にした電力計測機能を実装することで拡張が容易な CMP 電力シミュレータを開発する。ISIS-SimpleScalar では、あらゆる構成のシステムをシミュレートすることが可能な ISIS のシステムとの接続が可能であるため汎用性が高く、近年主流となっているマルチプロセッサシステムでのシミュレーションにも対応している。拡張が容易という点では目標とするシミュレータを目指すのに適している。Wattch は先にも述べたとおり、電力モデルとしては優秀であり、他の多くの論文でも Wattch の電力モデルを実装して評価を行っている。本稿でも電力モデルの部分のみ流用することとした。SESC の拡張性を高めるという方法で同様の目的を達成できるとも考えられるが、開発期間の観点から考えれば ISIS-SimpleScalar をベースに Wattch の電力モデルを実装するというアプローチのほうが優れていると考えられる。

Constructing of Power Estimation Environment in Extendable Simulator for CMP

[†]Yusuke Sato, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba

Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

3 CMP 電力シミュレータの設計と実装

3.1 設計方針

ISIS-SimpleScalar では, SimpleScalar の多くをクラス化することによりオブジェクトのインスタンスを生成するだけで簡単に複数のモジュールを用意できるようになっている. 今回開発するシミュレータにおいてもマルチコアプロセッサ化に対応するためにプロセッサコアを表すクラスを中心に拡張をしていく.

3.2 電力計測モジュールのクラス化

Wattch は図 1 のように SimpleScalar のハードウェアアクセス回数計測し, その結果を元に電力計算モジュールで最終的な電力を算出する仕組みになっている.

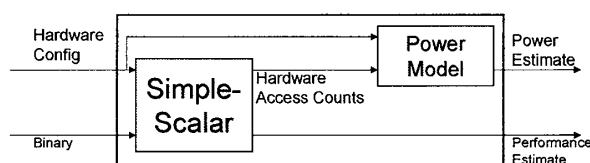


図 1: Wattch システムの概要

Wattch では一般的なスーパースカラプロセッサについて内部のユニットを, 配列構造, CAM (Content Addressable Memory), 組合せ回路, クロック回路の 4 つに分類して静電容量 C_{load} を決定している. 今回の実装にあたっては, Wattch のハードウェアユニットの分類方法および電力計算式はそのまま移植することとした.

Wattch は SimpleScalar が提供するプロセッサの動作モデルを記述した部分にハードウェアカウンタを取り付けて稼働率 α を測定している. 今回の実装においても同様の方法で ISIS-SimpleScalar の内部にハードウェアカウンタを追加し, 稼働率 α を測定することとした.

ISIS-SimpleScalar のプロセッサコアを表すクラスに対してハードウェアカウンタを追加し, 電力計算用の関数についてもプロセッサコアを表すクラスに追加した. これにより, マルチコアシミュレーションにおいて並列に稼働している他のコアの影響を受けず, 各コアごとにそれぞれ独立した電力統計の出力を可能にした.

3.3 出力される統計データ

オリジナル Wattch では出力される統計データは各ハードウェアユニットで個別に確かめることができる. また, プロセッサコア内部の各ハードウェアユニットに対してアクセス回数を出力することもできる. 今回実装するシミュレータでもこれらの統計を各コアごとに出力するようにした.

4 動作検証

動作検証では, オリジナルである SimpleScalar ベースの Wattch との誤差を検証するため, 各ハードウェアユニットのアクセス回数に注目して検証を行った. SimpleScalar3.0d ベースの Wattch と今回実装を行った ISIS-SimpleScalar1.1.1 ベースの Wattch とでそれぞれをデフォルトのパラメータで使用して比較を行った. $P_{active} = \alpha \cdot C_{load} \cdot f \cdot V_{dd}^2$ の電力計算式のうち, 静電容量 C の値をそのまま移植しているため, 動作周波数 f および電源電圧 V_{dd} を同じパラメータで動作さ

表 1: SimpleScalar ベースの Wattch との誤差

	test-math	test-fmath	test-printf	test-llong	test-lswlr	平均誤差
rename	1.77%	4.20%	11.23%	8.64%	4.10%	5.99%
bpred	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
window	14.72%	14.55%	20.15%	19.29%	10.60%	15.86%
lsq	2.89%	3.23%	5.92%	3.35%	2.68%	3.61%
regfile	8.78%	10.53%	1.02%	9.55%	29.64%	11.90%
icache	1.59%	3.75%	13.84%	10.60%	3.57%	6.67%
dcache	2.07%	2.24%	2.64%	1.22%	0.51%	1.74%
dcache2	2.43%	1.61%	0.45%	5.38%	0.72%	2.12%
alu	1.24%	2.04%	5.06%	3.34%	1.77%	2.69%
resultbus	2.31%	1.37%	2.72%	0.24%	9.36%	3.20%

せたときには稼働率 α が電力を決定する要素になる. 稼働率 α の値は各ハードウェアユニットへのアクセス回数と密接な関係がある.

検証には SimpleScalar に付属している test-math, test-fmath, test-printf, test-llong, test-lswlr の 5 つのテストプログラムを用いた. リネームユニット (rename), 分岐予測機 (bpred), 命令ウィンドウ (window), ロードストアキュー (lsq), レジスタファイル (regfile), L1 インストラクションキャッシュ (icache), L1 データキャッシュ (dcache), L2 キャッシュ (dcache2), ALU (alu), リザルトバス (resultbus) についてアクセス回数を出力し, 各ハードウェアユニットでのアクセス回数の差を SimpleScalar ベース Wattch のアクセス回数で除算を行い誤差を算出した.

結果は表 1 のようになり, いくつかのユニットを除いては特に問題無いことが確かめられた. 命令ウィンドウとレジスタファイルに関しては誤差率が特に大きい. オリジナルの SimpleScalar とオリジナルの ISIS-SimpleScalar の統計データで比較してみるとサイクル数やキャッシュアクセス回数などその他の部分についても出力結果に差が出てしまっているため, このような結果になってしまったのでは無いかと考えられる.

5 おわりに

ISIS-SimpleScalar をベースに Wattch で採用している電力測定方式を実装することで拡張容易な CMP シミュレータでの電力評価環境を構築した. 実装したシミュレータの精度を確かめるために動作検証を行った. 大部分のハードウェアユニットのアクセス回数に関しては, 電力シミュレータとして実用に耐えうる誤差の範囲内であったが, 一部のハードウェアユニットに対しては大きな誤差が発生してしまった. 今後, これらの誤差が妥当なものかどうかをさらに検証した上で, 共有バスなど CMP 特有のハードウェアユニットに関する電力消費を検討し, 実用に耐えうる CMP 電力シミュレータを開発することが課題である.

謝辞 本研究は, 一部日本学術振興会科学研究費補助金 (基盤研究 (B)18300014, 同 (C)19500037, 同 (C)20500047) および宇都宮大学重点推進研究プロジェクトの援助による.

参考文献

- [1] David Brooks, Vivek Tiwari, Margaret Martonosi "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations" in Proc of ISCA, June 2000, pp.83-94
- [2] SimpleScalar LLC, <http://www.simplescalar.com/>.
- [3] 葉袋俊也, "ISIS-SimpleScalar の実装", 慶應義塾大学大学院理工学研究科修士論文, 2005 年 3 月.
- [4] J. Renau, B. Fraguera, J. Tuck, W. Liu, M. Prvulovic, "SESC Simulator", Jan. 2005; <http://sesc.sourceforge.net/>.