

メタレベル最適化計算機システム YAWARA のシミュレーション環境 — スレッドエンジンシミュレータの実装 —

青木 隆行† 古川 文人‡ 月川 淳† 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 ‡宇都宮大学ベンチャー・ビジネス・ラボラトリー

1 はじめに

我々は、プログラムの実行状況に応じて自律的に最適化を行うことで性能向上を目指す、メタレベル最適化計算機システム YAWARA^[1] を提案している。そして、このシステムの評価のために、YAWARA の実行エンジンのシミュレータ開発を行っている。この実行エンジンは、スレッドエンジン (Thread Engine, TE) と呼ぶ VLIW プロセッサを複数並べたマルチプロセッサシステムである。本稿では、このシミュレータ開発の第一段階として作成した、単一 TE のクロックレベルシミュレータについて説明する。

2 YAWARA

YAWARA では、アプリケーションの実行と並行して実行プロファイル情報の取得を行う。そして、得られた情報をもとに、マルチスレッド化を含むアプリケーションの最適化を行うことで性能向上を目指す。このような一連のフィードバック処理や、処理全体の資源管理は全て一様に構成した TE で行う。

TE は、最大で 4 つの RISC 命令を同時に実行する。ここで RISC 命令は、PISA (Portable Instruction Set Architecture)^[2] で定義される命令と同一である。TE では、VLIW 命令の上位 2 ビット (バンドル長指定フィールド) の符号により同時に実行する RISC 命令数を指定することが可能である。これにより、不要な nop 命令によるメモリ消費の削減を実現する。

3 スレッドエンジンシミュレータ

3.1 シミュレーション環境

YAWARA シミュレーション環境は、マルチ TE のシミュレータとマルチスレッドコードを生成するシステムソフトウェアから構成される。現在、この環境の基盤となる単一 TE システム実験環境 CHA-MEN を開発している。

図 1 に、単一 TE システム実験環境 CHA-MEN を示す。CHA-MEN は、単一 TE シミュレータと、このためのシングルスレッドコードを生成する言語処理系^[3] から構成される。本言語処理系は、C 言語で記述された評価プログラムのソースコードを入力とし、TE 上で命令レベルの並列実行を行うために命令コードのスケジューリングを行い、シミュレータの入力であるメモリイメージを出力する。単一 TE シミュレータは、メモリイメージを入力として、HW 情報ファイル、命令

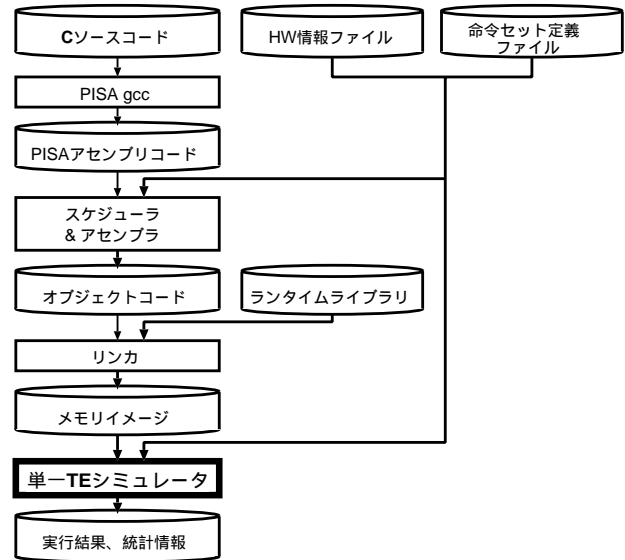


図 1: 単一 TE システム実験環境 CHA-MEN

セット定義ファイルから設定されるパラメータに基づいたシミュレーションを行い、実行結果、統計情報を出力する。

3.2 設計方針

本シミュレータでは、単一 TE におけるプログラム実行のクロックレベルシミュレーションを行うことにより TE の詳細な性能評価を可能とする。シミュレータ構築に費す時間を短縮するために、既存のプロセッサシミュレータをもとにして、最小限の変更を施すことで独自のシミュレータを開発する。さらに、YAWARA システム全体のシミュレータへの拡張を考慮し、変更・拡張が容易な構造を実現する。そこで、本研究では、C++ で記述され、可読性に優れた既存のプロセッサシミュレータ SimCore/Alpha RealScalar Version 0.9.30^[4] をベースとして選択した。そして、実行する RISC 命令を PISA に変更し、可変長の VLIW 命令を実行する機能の実装、SimpleScalar との計算結果の比較を目的として SimpleScalar と同一のシステムコールの実装を行った。さらに、柔軟なシミュレーションを実現するために、シミュレーションパラメータを容易に設定可能とし、パラメータとして設定される HW 制約の違反を検出する機能を実装した。また、プログラムの特定部分を評価するために、独自の統計情報出力機能の実装を行った。

3.3 モジュール構造

図 2 に、本シミュレータのモジュール構造を示す。本シミュレータでは、RealScalar の設計方針を継承し、それぞれの構成要素を独立した機能を持つモジュールとして実現することで、シミュレータへの変更や、新し

Simulation Environment for Meta-Level Optimizing Computer System YAWARA — Implementation of Thread Engine Simulator —
†Takayuki Aoki, Atsushi Tsukikawa, Kanemitsu Ootsu, Takashi Yokota, and Takanobu Baba, Department of Information Science, Faculty of Engineering, Utsunomiya University
‡Fumihito Furukawa, Venture Business Laboratory, Utsunomiya University

い機能の追加が容易に行える構造を実現している。本シミュレータは、大きく分けてプロセッサモジュール部 (scalar_chip), メインメモリ (main_memory), システムの設定 (system_config), 統計情報の出力部 (evaluation_result), 例外処理ユニット (exceptions), デバック情報の出力部 (debug) の 6 つのモジュールから構成される。さらに、プロセッサモジュール部はキャッシュシステム (Icache, Dcache, memory_system), 分岐予測器 (Pcgen, Bpred), VLIW 命令処理部 (instruction_s), システムコール処理部 (system_manager), レジスタファイル (architecture_state), プロセッサインターフェース (chip_wire) から構成される。ここで、exceptions は、HW 制約違反やプログラム実行中に発生するエラーを検出し、例外処理を行うためにオリジナルの RealScalar に対して、本シミュレータに新たに実装したモジュールである。

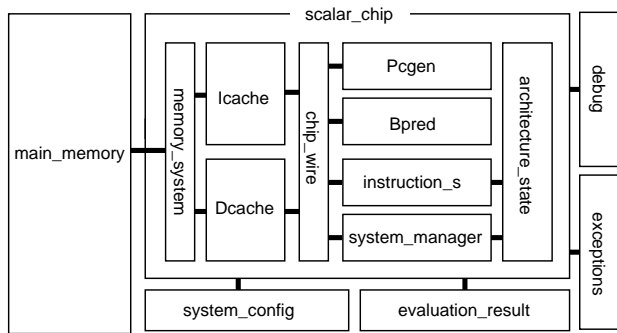


図 2: シミュレータのモジュール構造

3.4 HW 構成と実行サイクル数の設定

RealScalar に変更を加えることで、HW 情報ファイル、命令セット定義ファイルの 2 つのファイルからシミュレーションパラメータを容易に設定可能とした。HW 情報ファイルからは、キャッシュサイズ、キャッシュブロックサイズ、キャッシュレイテンシ、キャッシュミスペナルティ、整数演算ユニット数、浮動小数点演算ユニット数、レジスタ数を設定する。命令セット定義ファイルからは、各 RISC 命令の実行サイクル数を設定する。

3.5 HW 制約違反検出機能の実装

本シミュレータでは、パラメータとして HW 資源の制限を設定可能とするため、処理中の命令が制限を超えていないかチェックする必要がある。そこで、処理中の命令から制限違反を検出し、違反した場合に例外処理を行う機能を追加した。制限違反の種類として、レジスタ数制限違反、機能ユニット数制限違反に加え、同一レジスタへの複数の書き込みがあり、これらが検出された際には、例外処理としてシミュレーションを強制終了する。

3.6 統計情報出力機能の実装

プログラムの特定の部分を評価するとき、その部分のみの統計情報を抜き出し評価することが必要となる。本シミュレータでは、break 命令により統計情報を出力する機能を加えた。ここで、break 命令で統計情報を得る場合には、その命令の処理のためのサイクル数

が加わることや、キャッシュ内の命令の配置が変化するという問題がある。そこで本シミュレータでは、このようなプローブ効果を排除した独自の機能を新たに追加した。実装の容易さを考慮して、VLIW 命令中の特定のフィールドに情報を付加することで、その VLIW 命令の処理終了時にシミュレーション中止、レジスタダンプ、統計情報の出力を行う。図 3 に、TE 命令形式を示す。各ビットを 1 にセットすることで、これらの機能の 1 つまたは複数を同時に利用することができる。

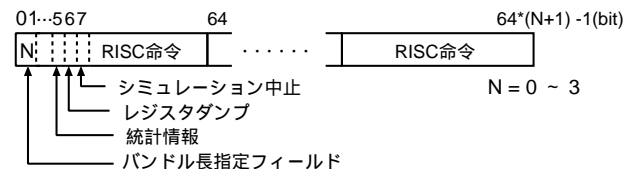


図 3: スレッドエンジン命令形式

3.7 シミュレータの動作検証

シミュレータの動作検証として、VLIW 形式にスケジューリングしていないプログラムにおいて、エラトステネスのふるい、Nクイーン問題、ライフゲーム、SPECint95 の 129.compress, 130.li について、SimpleScalar と同一の計算結果が得られることを確認した。また、VLIW 形式にスケジューリングしたプログラムでは、エラトステネスのふるい、Nクイーン問題について、正しい実行結果が得られることを確認した。

4 おわりに

本稿では、メタレベル最適化計算機システム YAWARA のシミュレータ開発の第一段階として作成した、単スレッドエンジンのクロックレベルシミュレータについて説明した。今後は、SPECint95, SPECfp95 などの大規模プログラムによる本シミュレータのテストを行う予定である。そして、マルチ TE 化への検討を行い、変更・拡張が容易な構造で、マルチ TE シミュレータを実現する予定である。

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)14380135, 同 (C)16500023, 若手研究 14780186) の援助による。

参考文献

- [1] Takanobu Baba, et al.: YAWARA: A Meta-Level Optimizing Computer System, International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems, pp.148-153 (2004.1)
- [2] Doug Burger, T.Austin: "The SimpleScalar Tool Set, Version 2.0," Univ. of Wisconsin-Madison Computer Sciences Department Technical Report #1342 (1997.6)
- [3] 月川 淳ほか: メタレベル最適化計算機システム YAWARA のシミュレーション環境 — PISA をベースとした VLIW アセンブラの開発 —, 情報処理学会第 67 回全国大会 (2005.3)
- [4] SimCore Project Homepage, <http://www.yuba.is.uec.ac.jp/~kis/SimCore/>