

大域ディペンダブル情報基盤プロジェクト

～ディペンダブルアーキテクチャグループ 田中・坂井研究室～

田中 英彦 坂井 修一

情報理工学系研究科電子情報学専攻

あらまし コンピュータシステムにとって処理速度・消費電力とともに重要なことがディペンダビリティである。われわれはマイクロプロセッサを対象として、システムLSIにマルチプロセッサと再構成可能ユニットを組み込み、高性能・省電力・ディペンダビリティの3者をバランス良く向上させるプロセッサアーキテクチャの研究を行っている。今年度は、チップマルチプロセッサ部の構造と機構を検討し、電力シミュレーションフレームワークを開発し、キャッシュ方式の提案などを行った。これらの有効性について、シミュレーションによって検証した。

1. はじめに

コンピュータシステムにとって処理速度・消費電力とともに重要なことがディペンダビリティである。本研究ではマイクロプロセッサレベルでのディペンダビリティを対象とし、システムLSIに再構成可能ユニットを組み込み、CPUと機能分散することで、性能とディペンダビリティの両面を向上させるプロセッサアーキテクチャ（コデザインを含む）の研究を行う。

本年度はその2年目として、チップマルチプロセッサ部の基本構成と処理方式について研究し、シミュレータを実装してベンチマークによってその初期的評価を行った。

2. 対象とするマイクロプロセッサ

ここでは、図1に示すような近未来のシステムLSIを考える。本システムLSIのアーキテクチャは、PDAからPC,サーバまで広範な用途を想定している。

本アーキテクチャはチップマルチプロセッサと再構成可能ユニット（RU、Reconfigurable Unit）から成

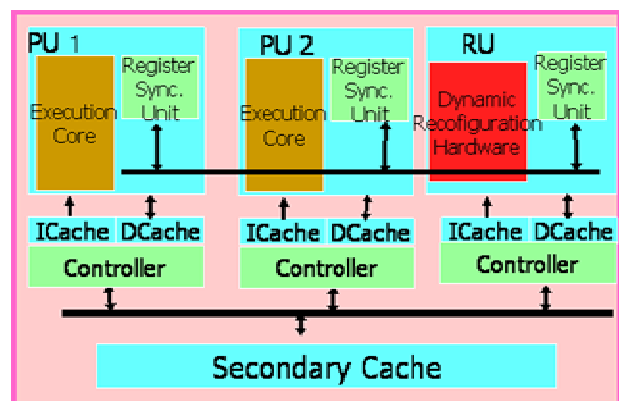


図1. ディペンダブルプロセッサLSI

り、プロセッサ多重利用による性能とディペンダビリティの向上、RUによるアプリケーション適応型のディペンダビリティ提供とPUとの機能分散、ディペンダブル・共有メモリプロトコルの実現、アプリケーション適応コード生成、多重実行コード生成、段階的デグレレーションなどがこの上で実現される。

ここでは、本LSIの中でチップマルチプロセッサ部の基本構成とその処理方式について検討する。

3. チップマルチプロセッサ NEKO

われわれは、投機的マルチスレッディングを行うチップマルチプロセッサ NEKO を考案し、シミュレーシ

ョン評価を行った。本研究で対象とするシステム LSI では、数値計算ではなく整数計算やメディア計算が処理の中心となるため、性能向上は単純な並列化では得られず、スレッド投機処理が重要な技術となる。一方で省電力を考えれば投機処理の抑制が必要な場合もあり、ディペンダビリティの観点からはマルチスレッド多重処理も考える必要がある。このように、スレッド投機の制御は、性能・省電力・ディペンダビリティのトレードオフ問題となる。

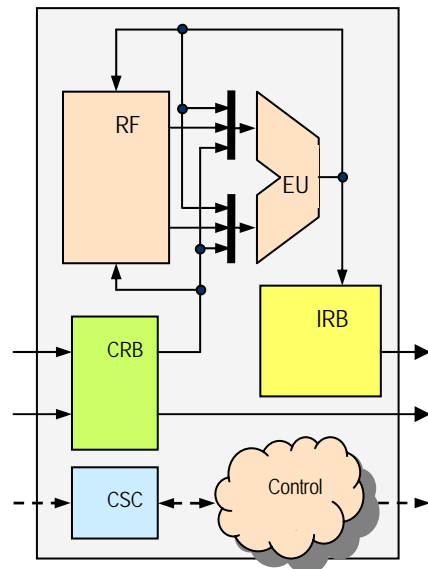
われわれはまず、NEKO に最適なレジスタ間通信アーキテクチャを考案し、評価した[1]。また、スレッド投機のための予測器の提案・評価を行った[2]。さらに、コンパイラの手法として、スレッド長の最適化[4]、投機処理支援[5]、値投機支援[6]などの技術を開発し、評価した。成果を以下に要約する。

- ・ リング結合のトポロジーによって効率的で低コストのレジスタ間通信が実現されることが示された(図2)。本方式はハードウェアを無限に投入した理想的な方式と比較して 93.5%の性能を得た。
- ・ スレッド投機をする環境で、レジスタデータの同期を行う方式を考案した。
- ・ スレッド投機用に Dual-Length Path-Based Predictor を考案し、有効性をシミュレーション評価によって示した。
- ・ コンパイラが行うスレッド分割について、制御フローグラフのノードではなくエッジをスレッド境界とする手法を提案し、初期的評価を行って有効性を検証した。これによって最大で 14%を超える性能向上がみられた。
- ・ ハードウェアコストのかからない効率的な値投機方式について提案・評価を行い、最適な投機方式を考案した。具体的には、コンパイラが静的に命令を分類し、値予測の候補から除外する命令を選択することで総予測数を削減することでこれを実現する。

4. 電力シミュレータ PRESTO

省電力の検証のために、アーキテクチャレベルの電力シミュレータである PRESTO の研究開発を行った。PRESTO では、さまざまなアーキテクチャへの適用性を高めるために、電力モデルに、テクノロジレイヤ、回路レイヤ、動的挙動レイヤの3階層を設け、それぞれでチューニングを行いことができるようにしている(図3)。

今年度は、基本素子に PRESTO をあてはめて SPICE との整合性をとるとともに、SRAM などいくつかの例に適用してその有効性を予備評価した[6]。



RF	<i>Register File</i>
EU	<i>Execution Unit</i>
IRB	<i>Intermediate Register Buffer</i> - Indexed by physical reg. no. - Temporarily holds register values (speculatively) generated by the PU
CRB	<i>Communication Register Buffer</i> - Indexed by logical reg. no. - Holds register values either generated or propagated by the previous PU
CSC	<i>Communication Scoreboard</i> - Indexed by logical reg. no. - Holds status of registers in comm.

図2 NEKOにおけるレジスタ間通信方式

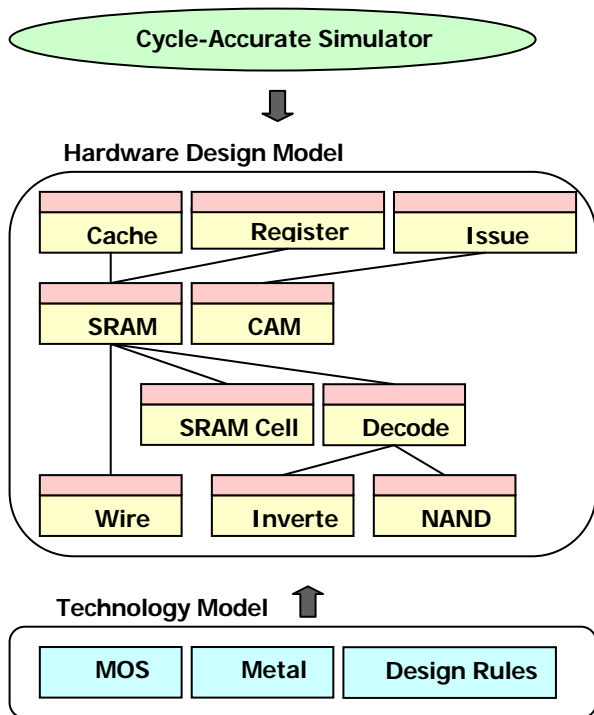


図3 PRESTOの構成

5. Way Variable Cache

省電力でディペンダブルなキャッシュは、本アーキテクチャで最も必要なものの一つである。われわれは、負荷に応じて動的にアクティブなウェイ数を変更するセットアソシアティブキャッシュを考案し、方式設計を行い、シミュレーションによってこれを評価した。

その結果、提案した方式によってアクティブなウェイ数を平均約60%に抑えながら、99.1%の性能が維持できることが確認された(図4)[7]。

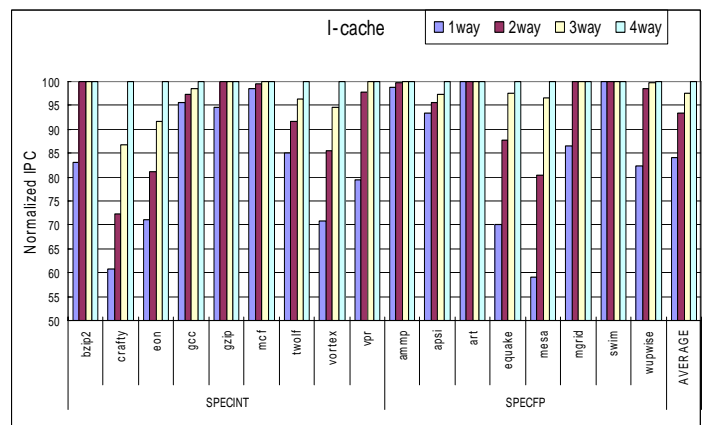
6. NEKOのキャッシュ方式

NEKOの一次キャッシュは、通常の共有メモリマルチプロセッサのキャッシュと比較すると、投機的マルチスレッディングを実現する点が異なっている。

われわれはNEKOの一次キャッシュのコントローラを機能設計し(図5)、ハードウェア量と処理の複雑さを見積もった。その結果、従来型のキャッシュのコントローラと比較して、クリティカルパスのレーテン

```
//collects  $N_{L2ACCESS}$ ,  $N_{TOTALACCESS}$  for the current window
if( $\frac{N_{L2ACCESS}}{N_{TOTALACCESS}} > Threshold1$  && way_current < way_max)
    way_next = way_current + 1;
else if( $\frac{N_{L2ACCESS}}{N_{TOTALACCESS}} < Threshold2$  && way_current > 1)
    way_next = way_current - 1;
else
    way_next = way_current;
```

(a)ウェイ切り替えアルゴリズム



(b)シミュレーション結果

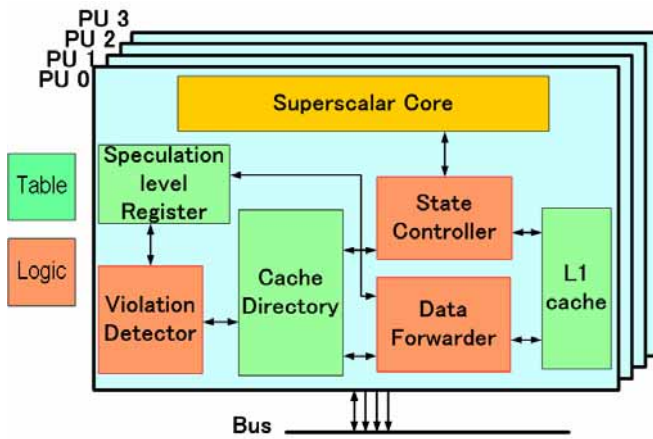
図4 Way Variable Cacheの効果

シが11%の増加で抑えられ、遅延時間としては、F04にして22段以下で抑えられることが示された[8]。

7. まとめ

今年度は、ディペンダブルLSIの主要な要素であるチップマルチプロセッサについて、性能と電力の両面から、アーキテクチャとコンパイラについて検討し、いくつかの有効な方式を開発することに成功した。

今後は、信頼性・安全性を向上させつつ性能・電力面も満足させるようなアーキテクチャ、コンパイラ的设计が課題となる。また、前年度に研究したRUとの組合せがさらなる課題となっている。



(a) プロセッサ内部構成

ユニット名 (動作)	Logic Delay [FO4]	Total Delay [FO4] (ディレクトリへの アクセス時間を含む)	Total Delay [ps] @90nm
Data Forwarder (Owner Probing)	2.5	13.9	500
Data Forwarder (Data Transfer)	7.3	7.3	263
Violation Detector	7.2	18.6	668
State Controller (状態遷移)	9.8	21.2	763

(b) レーテンシ見積もり

図5 キャッシュコントローラ

発表文献

- [1] Niko Demus Barli, Daisuke Tashiro, Chitaka Iwama, Shuichi Sakai, and Hidehiko Tanaka, A Register Communication Mechanism for Speculative Multithreading Chip Multiprocessors, Symposium on Advanced Computing Systems and Infrastructures (SACSIS) 於国立情報学研究所, pp.275-282 (May 2003).
- [2] Niko Demus Barli, Luong Dinh Hung, Hideyuki Miura, Shuichi Sakai and Hidehiko Tanaka, A Dual-Length Path-Based Predictor for Thread Prediction, International Workshop on Innovative Architecture 2003 at Kauai, Hawaii, USA, (Jan. 2003).
- [3] 田代 大輔、バルリ ニコ デムス、坂井 修一、

田中 英彦、スレッド投機実行におけるエッジに着目したスレッド分割手法、第145回計算機アーキテクチャ研究発表会、於慶應義塾大学日吉キャンパス、pp.67-72 (May 2003).

[4] Hideyuki Miura, Luong Dinh Hung, Chitaka Iwama, Daisuke Tashiro, Niko Demus Barli, Shuichi Sakai, and Hidehiko Tanaka, Compiler-Assisted Thread Level Control Speculation, Euro-par 2003 at Klagenfurt, Austria, pp. 603-608 (Aug. 2003).

[5] 飯塚大介、バルリ ニコ・デムス、坂井修一、田中英彦、値予測の軽量効率化方式の提案と評価、情報処理学会 ACS 論文誌、Vol. 44, No. SIG6 (ACS 1), pp.65-75, (June 2003).

[6] 岩間智女、ルオン ディン フォン、バルリ ニコ デムス、坂井修一、田中英彦、The Design of PRESTO: A Framework For Architecture Level Power Estimation, 第146回計算機アーキテクチャ(ARC)研究会 (SWOPP 松江 2003), pp.103-108 (Aug. 2003).

[7] Luong Dinh Hung, 岩間智女, Niko Demus Barli, 坂井修一, 田中英彦、Way-variable Caches for Static Power Reduction、情報処理学会 計算機アーキテクチャ(ARC)研究会デザインガイア 於北九州国際会議場 (Nov. 2003).

[8] Yoshimitsu Yanagawa, Luong Dinh Hung, Chitaka Iwama, Niko Demus Barli, Shuichi Sakai, and Hidehiko Tanaka, Complexity Analysis of a Cache Controller for Speculative Multithreading Chip Multiprocessors, Proc. HiPC2003, (Dec. 2003).