Vol.2010-ARC-187 No.13
Vol.2010-EMB-15 No.13
2010/1/29

SmartCoreシステムによるメニーコアプロセッサの信頼性向上手法

佐藤真平[†1] 植原昂[†1]
三好健文[†1,†2] 吉瀬謙二[†1]

我々は，メニーコアプロセッサにおいて，多数のコアの利用と高機能ルータによって性能向上を目指す SmartCore システム (Smart many-core system with redundant cores and multifunction routers) を提案している．本稿では，SmartCore システムの機能の 1 つである多重実行によるプロセッサの信頼性向上について提案する．高機能ルータにおいてパケットの複製，送信先の変更，比較をおこなうことで，複数コアの多重実行を支援し，エラー検出をおこなう．ソフトウェアシミュレータを用いて，多重実行を実現する高機能ルータにおいてパケット比較をおこなう保守的な実装を評価したところ，アプリケーションの性能低下がわずかであったことを確認した．

The SmartCore system to improve the reliabirity on Many-core Processors

Shimpei Sato,[†1] Koh Uehara,[†1] Takefumi Miyoshi[†1,†2] and Kenji Kise[†1]

We have proposed the SmartCore system (Smart many-core system with redundant cores and multifunction routers), which aims at the performance enhancement of many-core processors. In this paper, we propose a method to improve the reliability of many-core processors as one of the function of SmartCore system. In the proposed method, the multifunction router in SmartCore system applies packets duplication, changing their destination, and verification in order to execute programs redundantly and detect their errors. By software simulator, the execution performance is evaluated with some of benchmarks in NAS parallel benchmark. The result shows that the degradation by the proposed method is slightly little.

1. はじめに

プロセッサの性能向上と消費電力の削減を目的に，広い分野にわたり，複数のコアを搭載するマルチコアプロセッサが主流となりつつある．今後は，半導体の集積度の向上により，さらに多くのコアを集積するメニーコアプロセッサへと向かうと考えられる．

一方，LSI の一層の微細化により，ソフトエラー率およびばらつきが増加し，信頼性の低下が深刻化している[1),2)]．ソフトエラーは，メモリや論理回路などでランダムに発生する一時的な誤動作である．微細化が進むことで臨界電荷量が低減すると，中性子などの影響によりビットが反転する．マルチコアやメニーコアプロセッサにおいてはプロセスの微細化とともに集積するトランジスタ数が増加するため，ソフトエラー率の増加が懸念される．同時に，電源，温度の不均一などに起因するチップ内のランダムなばらつきが信頼性の低下を招いている[3)]．

我々は，1 チップに搭載可能なコアが数十から数千コアというオーダーまで増え続けることを想定し，そのような豊富なコアを活用し性能向上を目指す SmartCore システム (Smart many-core system with redundant cores and multifunction routers)[4),5)] を提案している．SmartCore システムはオペレーティング・システムなどのシステムソフトウェアと協調で動作する．メニーコアプロセッサにおけるチップ内ネットワークのルータの支援により，複数のコアによる多重実行を実現し，信頼性の向上，バンド幅の向上，レイテンシの削減，という 3 つの重要項目の改善を目指す．

本稿では，SmartCore システムの機能の 1 つであるプロセッサの信頼性向上を実現する手法を提案する．ソフトウェアシミュレータによる評価より，アプリケーションの性能低下の度合いを調査し，今後の課題を明らかにする．

本稿の構成を以下に示す．2 章で想定するメニーコアアーキテクチャを述べる．3 章で SmartCore システムによる信頼性向上手法を提案する．4 章で SmartCore システムを評価する．5 章で関連研究について述べる．最後に，6 章でまとめる．

2. M-Core アーキテクチャ

本稿では，M-Core アーキテクチャ[6)] を対象に信頼性を向上させる仕組みを実装する．

†1 東京工業大学 大学院情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology
†2 独立行政法人 科学技術振興機構
Japan Science and Technology Agency

Vol.2010-ARC-187 No.13
Vol.2010-EMB-15 No.13
2010/1/29

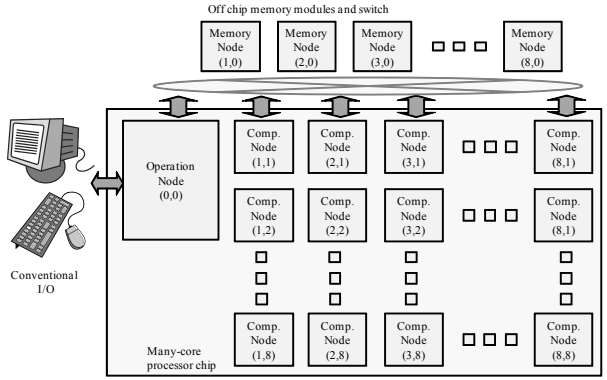


図 1 M-Core アーキテクチャ.

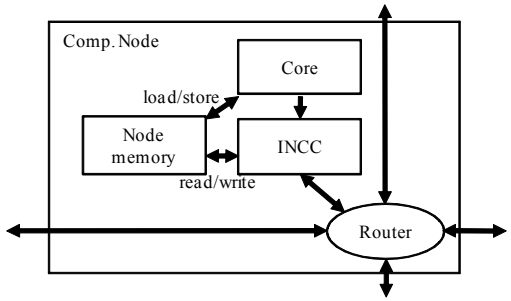


図 2 計算ノードの内部構成.

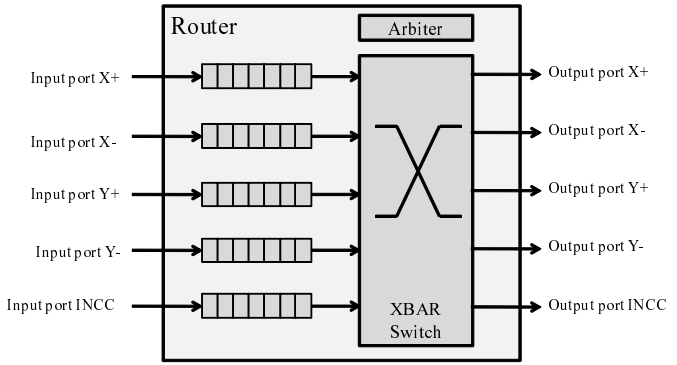


図 3 ルータの内部構造.

図 1 に，M-Core アーキテクチャのモデルを示す．主に，**計算ノード**，**オペレーションノード**，**メモリノード**の 3 種類のノードで構成される．ノードは X 座標と Y 座標との組合せで一意に指定される識別子 (ID) を持つ．本稿では，(X,Y) という ID を持つノードのことを **Node(X, Y)**，と表記する．図 1 の例では，Node(0, 0) がオペレーションノード，Node(1, 0) から Node(8, 0) がメモリノード．Node(1, 1) から Node(8, 8) が計算ノードとなる．オペレーションノードは，様々な I/O が接続されるノードである．メモリノードはオンチップのメモリコントローラで，オフチップのメインメモリに接続されている．それぞれのノードは 2 次元メッシュのネットワークで接続している．

計算ノードの内部構成を**図 2** に示す．計算ノードは**コア**，**ノードメモリ**，**INCC**(Inter-Node Communication Controller)，**ルータ**，で構成される．本稿では，Node(X,Y) のコアやルータを **Core(X, Y)**，**Router(X, Y)**，と表記する．コアは MIPS32 命令セットの RISC プロセッサ，ノードメモリはノードが持つ小規模のメモリである．INCC はノードメモリと他ノードとの DMA 転送を制御するユニットで，ルータに接続されている．コアはロード/ストア命令により自ノードのノードメモリにアクセスする．また，メモリマップド I/O を利用し，INCC に対して DMA 転送を発行する．ノード間のデータ転送は DMA 転送を用いる．

ルータの内部構図を**図 3** に示す．ルータは隣接する 4 つのルータおよび INCC と接続している．データ転送の単位であるパケットは，入力線を経由して入力バッファに格納され，クロスバを通り，適切な方向へと出力される．各入力ポートには 1 サイクルあたりの転送の単位であるフリットをいくらか格納するバッファを備える．構成をシンプルにするために仮想チャネル，出力バッファは持たない．ルータ-ルータ間，ルータ-INCC 間の接続はフリットのビット幅と等しく，入出力には別の信号線を利用するために全二重の通信が可能である．スイッチング方式はワームホールスイッチング，ルーティング方式は XY 次元順ルーティング，フロー制御は Xon/Xoff を採用する．ルータ内はパイプライン化されておらず，経路選択，アービトレーション，伝送を 1 サイクルで完了する[7]．したがって，ルータ間及びルータ-INCC 間は 1 サイクル/1 ホップとする．

3. SmartCore システムによる信頼性の向上

3.1 概　　要

SmartCore システムは，オペレーティング・システムなどのシステムソフトウェアと協調動作し，高機能ルータによる複数コアの多重実行によって，メニーコアプロセッサにおける信頼性の向上，チップ内ネットワークのバンド幅，レイテンシの改善を目指す仕組みである[4),5)]．本稿では，SmartCore システムの機能の 1 つである信頼性向上の手法を提案する．

図 4 を用いて，信頼性向上の仕組みを述べる．SmartCore では，高機能ルータを用いたパケットレベルでの信頼性向上を目指す．図の例では，Node(3, 2) と Node(4, 2) を用いて 2 重実行をおこない，DMR(Dual Modular Redundant) を実現する．高機能ルータにおいて Node(3, 2) と Node(4, 2) が送信するパケットを比較することでエラーを検出し，信頼性の向上を達成する．エラーが検出された場合は，実行を停止し再実行するなどの適切な回

Vol.2010-ARC-187 No.13
Vol.2010-EMB-15 No.13
2010/1/29

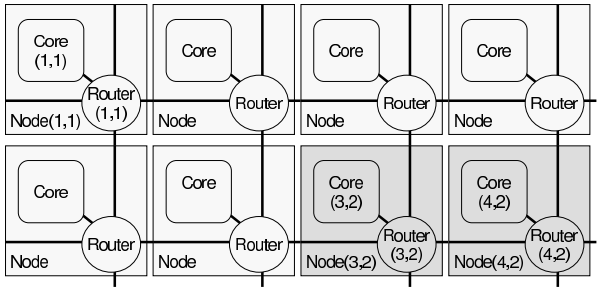


図 4 SmartCore システムによる信頼性向上.

復処置が必要となる．効率よく実行を継続するための適切なチェックポイントの設定などは今後の課題である．

3.2 信頼性向上を実現する高機能ルータ

アプリケーションを実行するために本来必要とされるノードを**マスターノード**，多重実行のために追加されたノードを**ミラーノード**とする．マスターノードとミラーノードで多重実行をおこない，信頼性の向上を達成するために，以下の 3 つの機能を高機能ルータで実現する．

- パケットの複製
- パケットの送信先変更
- パケットの比較

アプリケーションは，マスターノードに対してのみ通信を発生させる．したがって，マスターノードとミラーノードの多重実行を維持するために，マスターノードのルータにおいて受信するパケットを複製し，ミラーノードに送信する．ミラーノードから送信されるパケットをマスターノードのルータにおいて比較し，エラー検出をおこなう．そのために，ミラーノードのルータは送信するパケットの送信先をマスターノードへと変更する．

図 5 に，信頼性向上を実現するルータの内部構造を示す．ルータには，自身がマスターノードかミラーノードかの情報を持つ．また，マスターノードであれば対応するミラーノードの ID，ミラーノードであれば対応するマスターノードの ID の情報を持つ．これらの情報は，システムソフトウェアによってミラーノードの導入時に設定される．

(1) パケットの複製のための機構

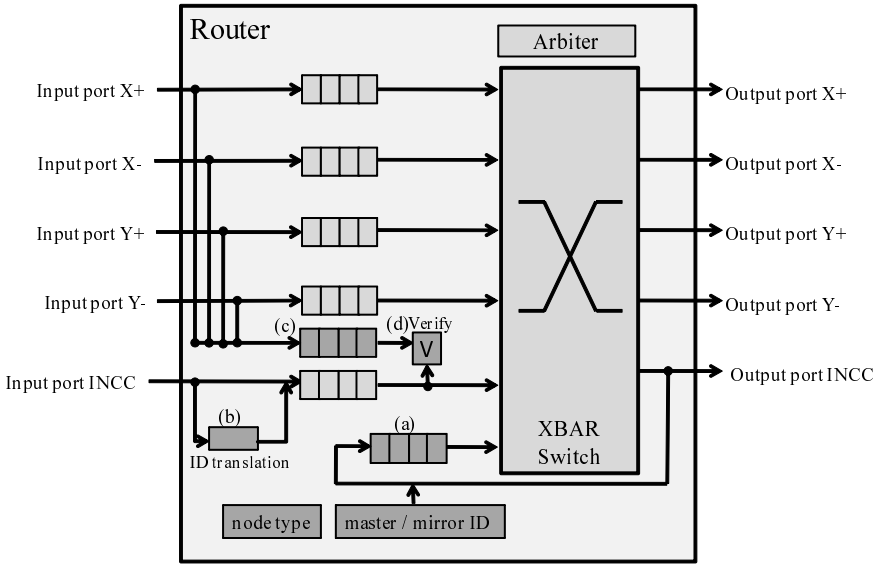


図 5 信頼性向上を実現する高機能ルータの内部構造.

ルータにおいて，複製したパケットのためのバッファ(図 5 中 (a)) を追加する．自身がマスターノードである場合は，INCC へパケットを出力するとともに，送信先をミラーノードへ変更した複製パケットをこのバッファへ格納する．追加したバッファのためにクロスバを 6 入力 5 出力の構成とする．本手法においては，デッドロックを回避するために複製したパケットを格納するバッファにおいてはカットスルー方式のスイッチングをおこなう．

また．この手法によるパケットの複製の場合，マスターノードとミラーノードでパケットの受信タイミングが異なる．受信タイミングを一致させるためには，5) において提案されているサーキットスイッチ方式のネットワークを用いて複製したパケットを送信する手法など，マスターノードのルータからミラーノードへの送信レイテンシが一定であることが保証される経路の確保が必要となる．

受信タイミングのずれが大きくなる場合，ミラーノードにおいてパケット送受信のオーバーラップが発生し，多重実行が維持できなくなることがある．提案手法では，後述のパケットの比較のための機構を用いてこれを解決する．

(2) パケットの送信先を変更する機構

Vol.2010-ARC-187 No.13
Vol.2010-EMB-15 No.13
2010/1/29

送信先を変更するために，ID の変換テーブル (図 5 中 (b)) を追加する．INCC から送信されるパケットの送信先情報を監視し，変換テーブルを参照することによりパケットの送信先を変更する．自身がミラーノードの場合，INCC から送信されるパケットの送信先はマスターノードに変更される．テーブルの情報はシステムソフトウェアによって設定される．

(3) パケットの比較のための機構

ミラーノードから送信されたパケットを格納するためのバッファ(図 5 中 (c)) を追加する．ミラーノードにおいて，パケットの送信先をマスターノードに変更する際に，ミラーノードからのパケットであることを示す識別子を付加する．マスターノードでは，その識別子によりミラーノードからのパケットを判別する．ミラーノードからのパケットの場合，追加したバッファにそのパケットを格納する．

マスターノードの INCC から送信されるパケットは，入力バッファにおいてミラーノードからのパケットを待ち合わせる．マスターノードのパケットは，ミラーノードからのパケットがバッファに格納されると，比較器 (図 5 中 (d)) で順次パケットを比較し, エラーを検出する．

本手法では，マスターノードからのパケット送信は，ミラーノードからのパケットを待ち合わせた後に送信されることが保証されるため，前述のミラーノードにおけるパケットのオーバーラップは発生しない．

3.3 パケットのレベルでエラー検出をおこなう SmartCore システムの利点

高機能ルータを中心にしてコアの単位で多重実行をおこない，かつパケットのレベルでエラーを検出する．したがって，パケットを送信するノードであれば信頼性を向上させることができる．そのため，チップ上のキャッシュなど実装の詳細に依存せず，多重実行するノードを柔軟に選択できるという利点がある．

また，本稿ではパケットの複製や比較をマスターノードのルータにおいておこなう手法を提案したが，マスターノードやミラーノード以外のノードにおいてパケットの複製や比較を行うことも可能である．この場合，性能を意識しつつ信頼性を向上させることができるが，デッドロックを回避するためにパケットのルーティングなどを十分に考慮する必要がある．

4. 評　　価

メニーコアプロセッサシミュレータの SimMc Version 1.0.0[6] に信頼性向上のための SmartCore システムを実装し，多重実行によるアプリケーションの性能がどの程度低下するか調査する．

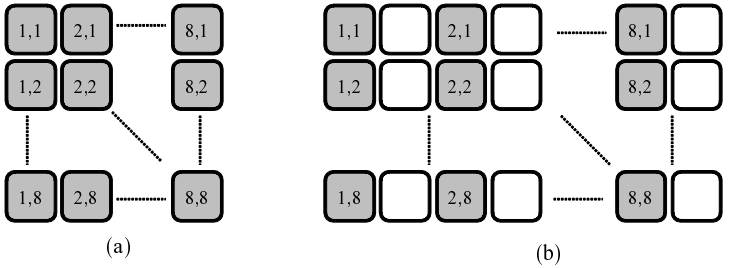


図 6 アプリケーションを実行するノードの配置.

4.1 評 価 環 境

ベンチマークは，NAS Parallel Benchmark の中から cg, ft, is, lu の 4 本を使用する．問題のサイズは S とし，使用するノード数は 8×8 の 64 ノードとする．

SimMc のパラメータを以下に列挙する．

- コアは MIPS ISA のシングルサイクルのプロセッサ
- オンチップネットワークのトポロジは 2 次元メッシュ
- INCC は DMA が発行されると次のサイクルからパケットの送信を開始する．また，1 サイクルに 1 フリット送信する．
- 1 パケットは 10 フリットを最長とし，転送量が大きい場合は複数のパケットに分割して転送する．先頭の 2 フリットにヘッダとアドレス，続く 8 フリットにデータが格納される．1 フリットに格納されるデータは 4 バイトとする．
- ルータは 1 サイクルに 1 フリット伝送する．ルーティングは XY 次元順ルーティング，フロー制御は Xon/Xoff 方式を採用する．

今回の評価では，ルータの入力バッファおよび SmartCore システムのために新しく追加したバッファのエントリ数は，理想化して十分に大きいものとした．これは，パケット比較のためにマスターノードとミラーノードが待ち合わせることよる性能低下を明確にするためである．

4.2 多重実行による性能の変化

アプリケーションの性能は，**図 6** の (a) のように 8×8 の領域にノードを配置した場合をベースとして，16×8 の領域に (b) のようにノードを配置した場合 (Area 2x) と，SmartCore システムを導入し，(b) の配置の各ノードの右側にある白色のノードをミラーノードとして

Vol.2010-ARC-187 No.13
Vol.2010-EMB-15 No.13
2010/1/29

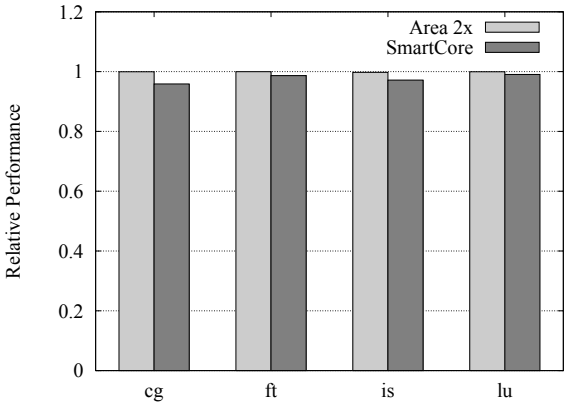


図 7 ベースのノード配置に対する相対性能.

動作させた場合 (SmarCore) を比較する.

図 7 に，ベースのノード配置に対する Area 2x と SmartCore の相対性能をベンチマークごとに示す．ベースに対する性能の低下は Area 2x ではすべてのベンチマークで 1%以下となった．SmartCore の場合は，cg で最も性能が低下し 4.1%であった.

図 8 に，SmartCore システム利用した場合，マスターノードがミラーノードのパケットを待ち合わせたサイクル数ごとの累積分布を示す．横軸はミラーノードからのパケットを待ち合わせたサイクル数，縦軸はあるサイクル数を待った通信回数の和である．図 8 より，ほとんどの通信はミラーノードからのパケットを待たずに行われていることがわかる.

以上より，SmartCore によるミラーノードのパケットの待ち合わせが，アプリケーションの性能に与える影響がわずかであることが確認できた.

5. 関 連 研 究

マルチコアプロセッサの空間的冗長性を利用し，高性能化とディペンダビリティ向上を達成する方式に Slipstream プロセッサ[8] がある．通常の命令列を 1 つのコアで動作させ，同時に，同様の挙動を示す短い命令列を異なるコアで動作させることで，高性能化とディペンダビリティの向上を達成する．この方式は，コアの同期のために低レイテンシの特別なコア間通信の仕組みを必要とするため，我々が想定している大規模なメニーコアアーキテクチャにおける利用は困難である.

マルチコアプロセッサの空間的冗長性を利用し，エラー検出および回復をおこなう手法に Loose Lock-stepped システム[9),10)] がある．この手法では，コア，キャッシュおよび主記憶をいくつかのグループに分割し，冗長実行させる．I/O のレベルでエラー検出する．主記憶を冗長実行させる点，比較をプロセッサのチップ外でおこなっている点が SmartCore システムとは異なる.

複数の CPU に同一の動作をさせることで故障を検出する技術として，PowerPC 750GX に採用されている Lockstep[11)] がある．チェック用のプロセッサを追加し，プロセッサの入出力のレベルでエラー検出する．2 つのプロセッサは，システムの起動時からクロックレベルで完全に同期する．SmartCore システムとは，冗長実行させるコアを柔軟に変更することが出来る点で異なっている.

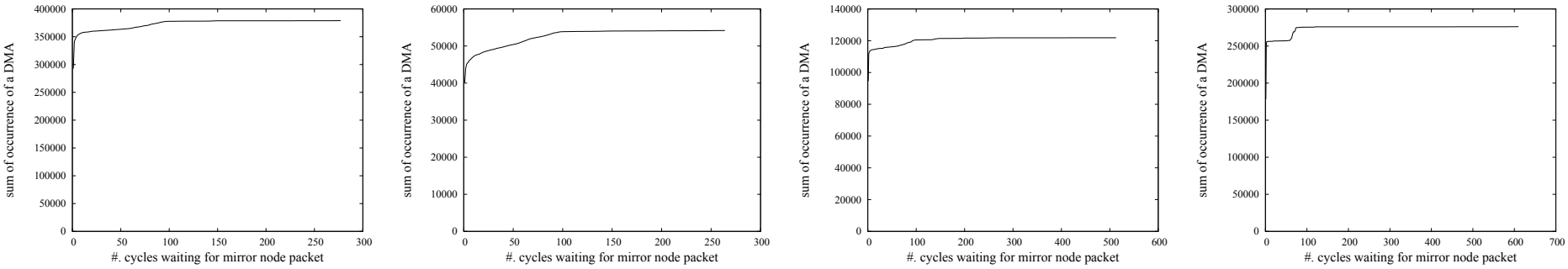
6. まとめと今後の課題

我々はメニーコアプロセッサにおける信頼性の向上，バンド幅の向上，レイテンシの削減という 3 つの重要項目の改善を目指す SmartCore システムを提案している．本稿では，SmartCore システムの機能の 1 つであるプロセッサの信頼性向上を実現する手法を提案した．アプリケーションの実行のために本来必要とされるノードをマスターノード，多重実行のために SmartCore システムが導入するノードをミラーノードとする．提案手法では，高機能ルータにおいて (1) マスターノードが受信するパケットを複製しミラーノードに送信する．(2) ミラーノードから送信されるパケットの送信先をマスターノードに変更する．(3) ミラーノードから届くパケットを待ち合わせ，マスターノードにおいてパケットの比較，エラー検出をおこなう．という 3 つの機能により信頼性の向上を達成する.

NAS Parallel Benchmark を用いたシミュレーションにより，(3) の待ち合わせによるアプリケーションの性能低下を評価した結果，SmartCore システムによる多重実行の性能低下は最大で 4.1%であった.

今後の課題として，次のことが挙げられる.

- 今回の評価において理想化したバッファのエントリ数を現実的なエントリ数とした場合に，デッドロックなどが発生せずに多重実行が行えることを調査する.
- エラーのモデルを設定し，実際にエラー検出を行い信頼性を計測する.
- エラーからの効率的な回復手法を検討する.
- 本稿では 2 重実行のみを実装した．さらなる信頼性向上のために 3 重実行をおこなう手法を検討する.
- 本稿では，パケットの複製，比較をマスターノードでおこなった．マスターノード，ミ



(a) cg (b) ft (c) is (d) lu

図 8 パケットの待ち合わせサイクル数ごとの発生回数の累積分布.

ラーノード以外のノードにおいてパケットの複製，比較をおこなうより柔軟な構成を検討する．

謝辞 本研究の一部は，科学技術振興機構・戦略的創造研究推進事業 (CREST) 「アーキテクチャと形式的検証の協調による超ディペンダブル VLSI」の支援による．ルータの基礎から実装に至るまで丁寧に指導していただきました電気通信大学の吉永努教授，東京大学の松谷宏紀博士に感謝いたします．

参 考 文 献

1) 佐藤寿倫，舟木敏正：マルチコアプロセッサのための電力・性能間トレードオフを考慮したディペンダビリティ選択法，情報処理学会論文誌， Vol.49, No.6, pp.2005–2015 (2008).
2) 佐藤寿倫，舟木敏正：性能・消費電力・信頼性の間のトレードオフを考慮出来るマルチ・クラスタ型コア・プロセッサ，電子情報通信学会技術研究報告. CPSY, コンピュータシステム， Vol.107, No.276, pp.39–44 (2007).
3) 入江英嗣，杉本 健，五島正裕，坂井修一：動的タイミング・エラー検出のための「書き込み保証バッファ」の評価，情報処理学会研究報告, 2007-ARC-173， Vol.2007, No.55, pp.73–78 (2007).
4) 吉瀬 謙二, 植原 昂, 佐藤真平：メニーコアプロセッサのディペンダビリティ向上と高性能化を目指す SmartCore システム，情報処理学会研究報告 2008-ARC-180， Vol.2008, No.101, pp.49–52 (2008).
5) 佐藤真平，植原 昂，吉瀬謙二：メニーコアプロセッサのオンチップネットワーク性能を向上させる SmartCore システム，先進的計算基盤システムシンポジウム SACSIS2009 論文集，pp.27–35 (2009).
6) Uehara, K., Sato, S., Miyoshi, T. and Kise, K.: A Study of an Infrastructure for Research and Development of Many-Core Processors, *Workshop on Ultra Performance and Dependable Acceleration Systems held in conjunction with PDCAT'09*, pp.414–419 (2009).
7) R. Mullins, A. West and S. Moore: Low-Latency virtual-channel routers for on-chip networks, *In ASP-DAC '06*, pp.164–169 (2006).
8) Sundaramoorthy, K., Purser, Z. and Rotenberg, E.: Slipstream processors: Improving both Performance and Fault Tolerance, *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, Vol.35, No.11, pp.257–268 (2000).
9) Aggarwal, N., Ranganathan, P., Jouppi, N.P. and Smith, J.E.: Configurable Isolation: Building High Availability Systems with Commodity Multi-core Processors, *ISCA '07: Proceedings of the 34th annual International Symposium on Computer Architecture*, pp.470–481 (2007).
10) Aggarwal, N., Jouppi, N.P., Smith, J.E., Ranganathan, P. and Saluja, K.K.: Implementing High Availability Memory with a Duplication Cache, *Proceedings of the 41st Annual International Symposium on Microarchitecture (MICRO-41)* (2008).
11) PowerPC 750GX Lockstep Facility: IBM Application Note (2008).