情報処理学会論文誌　コンピューティングシステム　Vol. 4　No. 4　24–35 (Oct. 2011)

エラー訂正・検出符号を用いたNetwork-on-Chipの低消費電力化

小島　悠[†1]　松谷宏紀[†1]
鯉渕道紘[†2,†3]　天野英晴[†1]

Network-on-Chip（NoC）においてプロセスの微細化やそれにともなう低電圧化によりビットエラー率が上昇する点が問題となっており，DVFS などによる低消費電力化の大きな妨げになりつつある．そこで，本研究ではエラー耐性技術を用いることでエラー率を抑えつつ電圧を下げる NoC の低消費電力化手法を提案，評価する．まず，現実的なエラー耐性技術を 8 つのパターンに絞り，これをすべて適用可能なルータ構成を提案した．提案ルータでは，エラー検出・訂正処理を通常のルータのルーティング処理と同時に行い，エラーが検出された場合のみ処理をやり直す投機的な実行を採用した．また，空いている入力チャネルバッファを再送に利用することで専用のバッファなしで再送を実現している．評価結果より，電源電圧を下げてエネルギー効率を上げるという点において，ヘッダフリットは各ルータでパリティによるエラー検出を行い，データフリットには end-to-end レベルの CRC によりエラー検出を行う手法が優れており，通常の NoC と比較して転送エネルギーを 45%削減できた．

A Low Power Network-on-Chip Using Error Correction and Detection Codes

Yu Kojima,[†1] Hiroki Matsutani,[†1]
Michihiro Koibuchi[†2,†3] and Hideharu Amano[†1]

As semiconductor technology improves, the supply voltage is lowered and bit-error rate is increased, which introduces a difficulty in further reducing the supply voltage of the chip with maintaining the bit-error rate using low-power techniques, such as DVFS. We propose and evaluate a low-power Network-on-Chip (NoC) structure based on fault-tolerant techniques in order to reduce the supply voltage while maintaining the bit-error rate. We first define eight practical fault-tolerant techniques and propose a router structure that supports all of them. The proposed router speculatively performs the routing computation and the error detection/correction at the same cycle. Also, it performs the link-level retransmission without additional buffer resources by using unoccupied part of input buffers. Evaluation results show that a fault-tolerant technique that uses the link-level parity check for header flits and the end-to-end level CRC error detection for data flits achieves the best energy efficiency. It also shows that the energy consumption for sending a flit is reduced by 45%.

1. はじめに

半導体技術の進歩にともない，メニーコア化が可能となり，並列処理によって高い最大性能を実現する一方，場合によっては，動作電圧と動作周波数を下げて消費電力を抑える手法が一般的になっている．チップ全体の電力を下げるためには，コア間を接続する Network-on-Chip（NoC）も低消費電力化が必要であり，様々な手法が提案されている[1)]．

要求処理速度に応じて動作周波数と電圧を動的に制御する Dynamic Voltage and Frequency Scaling（DVFS）は，CPU コアでは一般的な技術であり，NoC の低消費電力化にも有効である[2)]．一般に，半導体の動的消費電力 P は以下のように表される．

$$P = \frac{1}{2}\alpha C V^2 f \tag{1}$$

ここで，α はスイッチング確率，f は動作周波数，C はキャパシタンス，V は電源電圧である．DVFS は，要求処理速度が低い場合に V と f を下げることにより消費電力を削減する技術である．

しかし，通常の動作においてすでに低い電圧を用いる先端プロセスでは，電源電圧を下げることにより，ソフトエラー率が上昇する傾向にあり[3),4)]，DVFS 技術の適用にとって大きな問題になりつつある．

そのため，チップ全体として，プロセッサ，共有メモリ，NoC 各々にソフトエラー対策が必要となるが，これまで前者 2 つについては様々な耐故障技術が研究されてきた．そこで，本研究では NoC のソフトエラーに焦点をあてる．NoC におけるソフトエラー対策の研究では，各スイッチ内部のフリップフロップに着目して対策を行っていることが多い[5)]．し

†1 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
†2 国立情報学研究所
National Institute of Informatics
†3 総合研究大学院大学
The Graduate University for Advanced Studies

© 2011 Information Processing Society of Japan

かし，NoC ではクロストークにより信号線の値が反転してしまうエラーなども生じるため，そのための対策も必要となる．

クロストークは，2 本の長い信号線が並んで配置された際，片方の信号の変化がもう一方の信号に影響を与え値が反転する問題である．これも同様に，低電圧化により問題が深刻化する[6]．タイル状に並んだ各ローカルコア（あるいは Processing Element）をつなぐ NoC では多数の配線が長い区間にわたって隣接して配線されることが多く，クロストークが発生しやすく，このための対策も重要となってきている．

このクロストークによるソフトエラーに対しては，物理的にチップの配線間の距離を広げることで解決するなどの方策も可能であるが，本研究では，配線間の距離はそのままに，既存の研究[6),7)] と同様にエラー訂正・検出符号を各チャネルに付加するシンプルな方法について検討を行う．つまり，本研究ではロジック回路とワイヤにおけるソフトエラーに対して検出，訂正を行う．そして，エラー耐性技術を用いることでソフトエラーの発生率を抑えつつ電圧を下げる NoC の低消費電力化手法を提案する．さらに各種のエラー耐性技術を検討し，要求されるエラー率に対してどの程度電圧を下げることができるかを評価する．また，このエラー耐性技術の 1 つとして，ルータの余剰なバッファを再送用のバッファとして用いることで，面積オーバヘッドの少ない耐故障ルータを実現する．

本稿の構成は次のとおりである．まず 2 章で関連研究を紹介し，3 章で NoC におけるエラー検出・訂正方式を検討する．次に 4 章で各種エラー耐性技術が実装された耐故障ルータの設計について述べ，5 章でエラー耐性技術ごとのエラー率，電圧削減率をモデル化する．6 章では各種エラー耐性技術を面積，エラー率，消費エネルギーについて評価する．最後に 7 章で本稿をまとめる．

2. 関 連 研 究

DVFS あるいは複数の電源を動的に切り替えて消費電力を削減する方法は NoC に対してもいくつか先行研究が行われている[2),8)]．これらの手法では，NoC が混雑し，高い性能が必要とされる場合には高い電圧と周波数を用い，交換すべきパケットがなくなると電圧あるいは周波数を下げて消費電力を削減する．しかし，これらの論文では電圧を下げたことによるエラー率の悪化は考慮されていない．

プロセス微細化にともない，ソフトエラーは増加する傾向にある[9]．また，微細化にともなう電源電圧の低下も，ソフトエラーを増加させる 1 つの要因となっている[10]．従来，ソフトエラーはメモリのみが対象と考えられてきた．しかし，微細化と半導体の動作周波数の向上にともないロジック回路内部でのソフトエラー率が上昇すると考えられ，50 nm 世代のプロセスではロジック回路と SRAM で起きるエラー率がほぼ等しくなるといわれている[3]．

本稿では，DVFS や複数電源の動的切替えの適用法を提案しているのではなく，NoC において要求されるエラー率を維持したまま，電源電圧を下げてこれらの手法を実現するためのエラー耐性技術の検討と，そのためのルータ構成の提案を目的としている．DVFS や複数電源の動的切替え手法自体は，従来の研究のどの手法と組み合わせることも可能である．

エラー訂正，検出符号を用いることで，転送エネルギーの削減と高信頼性を両立する既存の研究として，文献 6)，7) があげられる．しかし，両者は，(1) 符号の付加にともなうルータ全体の面積オーバヘッドの定量的な評価が行われておらず，かつ，(2) NoC の動作に致命的な影響を及ぼすヘッダ情報と，アプリケーション・データ情報の区別なく符号化しているため，分離によるさらなる両立の向上の余地がある．

そこで，本研究では，これらの 2 点の調査に加えて，供給電圧などにより変動するチップのエラー率に応じて，様々なエラー検出・訂正符号を扱うことができる耐故障ルータの構成を提案し，レイアウトデータを含めて評価を行う．

3. NoC におけるソフトエラー耐性技術

NoC におけるソフトエラー（以後，単にエラーと呼ぶ）耐性技術は，1) エラー検出・訂正方式，2) パケット・フリットへの適用箇所，3) 適用方式（end-to-end・switch-to-switch）の 3 つの設計要素がある．これらにより NoC のスループット，消費エネルギー，レイテンシなどの性能が決定される[11]．

3.1 エラー検出・訂正方式

NoC におけるシンプルなエラー検出・訂正符号として，エラー検出にはパリティ符号や CRC 符号をフリットに対して付加する方法，エラー訂正には水平垂直パリティやハミング符号を用いる方法があげられる．

3.2 パケットへの適用範囲

NoC においてデータ（メッセージ）はパケットとして転送される．パケットはヘッダ部とデータ部に分けられる．パケットのヘッダにはパケット長，送信先，送信元，順序番号などの制御情報が格納されている．そのため，ヘッダの情報にエラーが生じた場合，NoC の動作に致命的な影響を及ぼすことが考えられる．たとえば，ヘッダのエラーによって，パケットの誤配，ACK/NACK の不達によるデッドロックなどネットワーク全体の不具合を引き起こす可能性がある．このため，ヘッダとデータには異なったエラー耐性技術を用いる

© 2011 Information Processing Society of Japan

ことで性能向上する場合がある可能性がある．

3.3　適用方式（end-to-end・switch-to-switch）

エラー検出・訂正は，end-to-end で行う方式と，switch-to-switch [*1] で 1-hop 進むごとに行う方式の 2 つが考えられる．

end-to-end 方式では，パケットの目的地でエラー検出を行い，エラーが検出された場合は NACK を送信し，出発地から再送を行う．エラーがなければ，送信元に対して ACK パケットを送信する．end-to-end 方式ではパケット単位でエラー検出・訂正する方法と，フリット単位でエラー検出・訂正する方法の両方が利用可能である．

一方，switch-to-switch 方式でエラー検出・訂正を行う場合は，フリット単位で行う．これは，多くの NoC ではレイテンシ削減のため，ワームホールスイッチングを採用しており，それぞれのルータではクロックごとにフリットが通過して次のルータに転送されるため，パケット単位でエラーを検出することが困難であるためである．switch-to-switch 方式では，ルータ単位でデータの再送が必要となるため，1 つ手前のルータにおいてデータを蓄えておく必要がある．この方式に関しては後に詳しく検討を行う．

3.4　検討する各方式

ヘッダにおける制御情報のエラーは，深刻なパケット配送問題を引き起こす．一方，それ自体の情報量は多くない．そこで，ヘッダフリットに対しては switch-to-switch でエラー検出・訂正を行うこととする．データに対しては switch-to-switch と end-to-end の両方が可能である．本稿で検討する各方式の組合せを**表 1** に示す．表中の SS は switch-to-switch を表し，EE は end-to-end である．CRC 符号については最大連続した 8 つまでの誤り検出ができる CRC8 符号を用いることとした．

表 1　本稿で検討するエラー検出・訂正方式の組合せ
Table 1　Combination list of error detection and correction methods considered in the paper.

	Header Flit	Data Flit
Pattern1	パリティ（SS）	パリティ（SS）
Pattern2	ハミング（SS）	ハミング（SS）
Pattern3	パリティ（SS）	ハミング（EE）
Pattern4	ハミング（SS）	ハミング（EE）
Pattern5	パリティ（SS）	水平垂直パリティ（EE）
Pattern6	ハミング（SS）	水平垂直パリティ（EE）
Pattern7	パリティ（SS）	CRC8（EE）
Pattern8	ハミング（SS）	CRC8（EE）

⋆1 本稿では，ルータとスイッチ（switch）を同義で用いている．

4.　耐故障ルータの設計

3 章で選択した 8 つのパターンのエラー検出・訂正を実現する耐故障ルータを提案する．ここでは，典型的な 2 次元メッシュ・トーラス構造に用いる 5 入出力のルータを取り扱うが，同様の構成は入出力数に関係せず実現可能である．

4.1　故障検出・訂正ユニットの検討

基本的なパイプライン型ルータにおけるパケット処理は 3 つのステップに分けることができる．パケットが入力ポートに到着すると，まず，入力ポートにおいてパケットのヘッダから宛先アドレスを解読し出力ポートを計算する（Routing Computation: RC）．次に，出力仮想チャネルおよびクロスバを割り当てる（Virtual-Channel and Switch Allocation: VA/SA）．そして，クロスバスイッチ上のデータ転送を行う（Switch Transfer: ST）ことで，パケットは出力ポートへ転送される．

ここでは，NoC を流れるパケットのエラーが NoC 全体に及ぼす影響を考える．通信中にヘッダにエラーが生じた場合，RC ステージにおいてそのパケットの送信先やパケット長に関する情報を誤検出される恐れがある．

そこで，エラー検出・訂正ユニットをルーティングを行う各入力チャネルに実装した．クロスバや出力ポートなどパケットのヘッダが動作に影響を及ぼさない部分ではエラー検出・訂正を行わない．さらに，仮想チャネル数にかかわらず，1 本の物理チャネルには，1 サイクルあたり 1 フリットの転送のみ可能であるため，同一物理チャネル内の仮想チャネルどうしでエラー検出・訂正回路を共有することで面積効率を上げることもできる．

なお，end-to-end 方式を用いる場合，5 ポートのうちローカルコアと接続されている 1 ポートのみにエラー検出符号のエンコーダ・デコーダを実装すればよいが，switch-to-switch 方式では各ホップでエラー検出・訂正を行うため，各ポートにエラー検出・訂正ユニットが必要となる．しかし，end-to-end の場合，ヘッダフリットのエラー検出・訂正符号とデータフリットのエラー検出・訂正符号をそれぞれ独立して持たなくてはならないが，switch-to-switch ではそれら 2 つを区別せず 1 つのエラー検出・訂正符号で対処できるため，面積が抑えられる．

4.2　耐故障ルータの構成

基本的な 3 サイクル・ワームホールルータを基にした耐故障ルータの詳細を**図 1** に示す．Error Correction: EC/Error Detection: ED Unit がエラー訂正・検出ユニットであ

© 2011 Information Processing Society of Japan

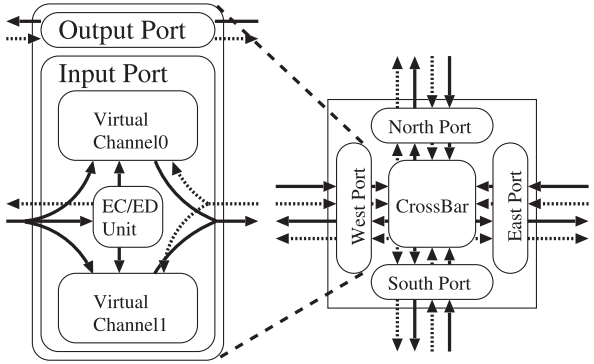

図 1　耐故障ルータの構造（ローカルコアへの入出力ポートは略）
Fig. 1　A fault-tolerant router structure (input and output pores between local cores are omitted).

り，実線の矢印がフリットの流れを示し，破線の矢印が ACK/NACK の流れを示している．ACK/NACK は switch-to-switch でのエラー検出（Pattern 1，3，5，7）でのみ必要であり，switch-to-switch でエラー訂正を用いる場合は必要ない．

まず，ルータに入力されたデータは入力ポートを通り，各仮想チャネルとエラー検出・訂正ユニットに送られる．そこで，各フリットにエラーがない場合はクロスバを通過し，出力ポートに送られ次のルータへ送られる．

次に，本ルータにおける入力ポートの詳細を**図 2** に示す．

各仮想チャネルでは入力されたヘッダの宛先アドレスから出力チャネルを計算する．エラー訂正ユニットではフリットをレジスタに保存し，次のクロックでエラー検出・訂正を行って，エラーのないフリットを FIFO に保存する．

エラー検出・訂正ユニットを RC ステージの手前に実装すると新たにステージが増えレイテンシが増加するが，今回の設計ではエラー検出・訂正のために新たにステージを設けず VA/SA ステージと並行して計算するため，レイテンシに対するオーバヘッドが発生しないというメリットがある．さらに，仮想チャネルどうしでエラー検出・訂正ユニットを共有するため，面積オーバヘッドも抑えることができる．

本方式では，仮想チャネルで行う出力チャネルの計算は，エラー検出・訂正をしていないヘッダフリットを用いるため，ルーティング処理の投機実行をしていることになる．つまり，RC ステージの投機実行が失敗した場合には，パケットのヘッダ情報にソフトエラーが発生し，ヘッダ・フリットのエラー訂正，あるいは再送により遅延が生じる．

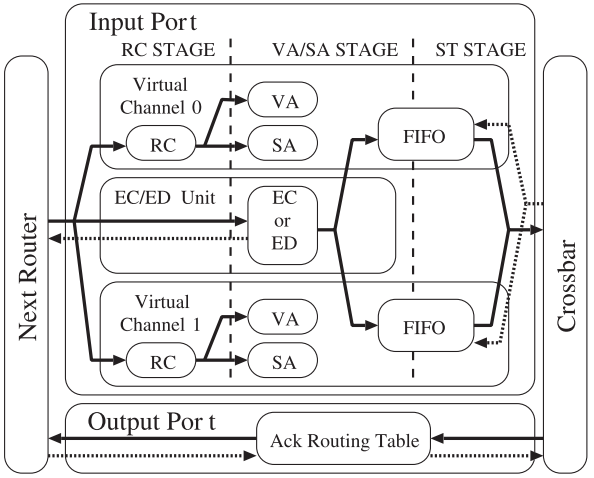

図 2　耐故障ルータにおける入力ポート・出力ポートの構造
Fig. 2　An input and output port structure on a fault-tolerant router.

しかし，この遅延に関わる処理は，ヘッダフリットに対して同一のエラー検出，訂正符号を用いた従来のほとんどの耐故障 NoC の研究と本提案手法は同等である．また，これら従来の研究では，訂正，検出処理のためのパイプラインステージを追加し，ルータ内パケット転送遅延が恒常的に増加する．つまり，無負荷時の遅延が増加する．これに対し，本方式では，その訂正，検出オーバヘッドを最小限にするために，（ルーティング情報に誤りが含まれていないと仮定して）投機的に実行を行う．そのため，ルーティング情報にソフトエラーが含まれていない場合のルータ内パケット転送遅延を削減することができるという利点がある．

switch-to-switch でエラー検出を行う場合は，データの再送が必要となるため，隣接ルータにおいてデータを蓄えておく必要がある．この目的で，各ルータに再送のための専用バッファを設ける方法が提案されている[11]が，これはオーバヘッドが大きい．そこで本稿では，Reliable Router[12]の再送機構のように，パケット転送に利用されていない FIFO の一部を再送に利用することにした．

このために，1) 隣接する送信ルータにおいて FIFO に残った転送済みのフリットを消さずに残しておき，2) 受信ルータにおいては受信フリットにエラーが生じた場合，隣接する送信ルータに NACK，無事に受信できた場合，ACK を返す（図 2）．送信ルータは NACK

© 2011 Information Processing Society of Japan

を受信した場合，消さずに FIFO に残しておいたフリットを再送し，ACK を受信した場合，その部分の FIFO 領域を次のパケット送信のために解放する．

具体的な動作は以下のとおりである．FIFO に入力されたフリットは VA/SA が完了し次第出力され，クロスバを通過し，次のルータへ送られる．この際に，クロスバにおいてフリットの送信元の FIFO と送信先のルータを図 2 の ACK Routing Table に記録する．そして，次のルータの入力ポートにおいてフリットごとにエラー検出がされた後に，NACK あるいは ACK を送信元ルータの FIFO に返す．送信先のルータから返信される ACK/NACK は ACK Routing Table を参照することで，送信元の FIFO に送られる．本実装では FIFO にリードポインタとライトポインタに加えて，ACK ポインタがあり，ACK を受け取っていないフリットが他のフリットで上書きされないようになっている．ACK が到着した場合は該当するフリットが格納されていた FIFO 内部の ACK ポインタをインクリメントし，NACK の場合は再送を行う．ACK あるいは NACK の到着には，FIFO からフリット送出後，4 サイクル必要となる．以上のルータの状態遷移を**表 2** にまとめる．

Error_STAGE はエラーが発生した場合にだけ使われるステージである．エラーを検出するのは，図 2 の VA/SA STAGE である．ヘッダフリットでエラーが検出された場合，再度 RC および VA/SA をやり直す必要がある．エラー訂正方式では訂正したフリットを使い，再度 RC，VA/SA を行う．一方で，エラー検出方式では，RC の結果を破棄し，エラーの検出されたフリットは FIFO に格納せずに破棄し，再送要求を出す．

また，リンク間再送をする場合（switch-to-switch でエラー検出を行う場合）は ACK_STAGE が存在する．これはすべてのフリットを送信し終わった後，ACK フリットを受信するステージである．ACK ステージには 4 サイクルの時間がかかり，このステージの後に次のパケットの受信を開始するため，性能オーバヘッドが発生する．しかし，Pattern3, Pattern5，Pattern7 に関してはヘッダフリットのみのエラー検出であるため，データフリットが 4 フリット以上連なっている場合はヘッダフリットの ACK がデータフリットを送信している間に届くため，ACK_STAGE は隠蔽され，性能オーバヘッドは発生しない．

Pattern7 と Pattern8 では end-to-end で CRC8 によるエラー検出を行っているため，end-to-end での ACK，NACK の送信が必要であり，1 フリットパケットを用いて ACK/NACK を送信する．再送バッファには各ローカルコア（Processing Element）内部にあるバッファを利用することとした[⋆1]．

ACK，NACK パケットを送る際にはデッドロックを考慮し，送信されてきた仮想チャネルとは異なるチャネルに ACK，NACK のパケットを流す．ACK を生成するユニットはノードに対する入出力ポートに付加する形で実装した．また，ACK を送信するネットワークが輻輳する可能性を考慮し，一時的に ACK/NACK フリットを保存する FIFO を設けた．

表 2　switch-to-switch 方式におけるパケット転送の状態遷移
Table 2　State transition of packet transfers on a switch-to-switch method.

	エラー訂正 （リンク間再送なし）	エラー検出 （リンク間再送あり）
RC_STAGE	投機的	投機的
VSA_STAGE	投機的 エラー訂正	投機的 エラー検出
ST_STAGE	変更なし	変更なし
ACK_STAGE	なし	送信した ACK の受信
Error_STAGE	ヘッダの場合 RC，VA/SA の再実行	RC，VA/SA は中止 し再送要求

5. 耐故障 NoC のエラー・エネルギー解析

前章で提案したルータを用いることで，エラー耐性技術を実装していない通常のルータを用いた場合と比較して，同等のエラー率を保ちながら低電圧化を行うことができる．

どのような Pattern がどの程度のエラー率・電圧で有効になるかを調べるためには電圧とエラー率の関係をモデル化する必要がある．正規分布に基づく電圧ノイズを V_N，その標準偏差を σ_n とした場合，以下の式で電圧 Vdd とワイヤのエラー率 ϵ の関係を示すことができる[13),14)]．

$$\epsilon = Q(\frac{Vdd}{2\sigma_n}) \tag{2}$$

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy \tag{3}$$

この ϵ を 1-hop，1-bit あたりのエラー率として考えて各エラー訂正手法におけるエラー訂正能力 $E[i](\epsilon)$ を考える．ただし，i には表 1 の各手法の番号が入るものとする．このようにした場合，$E[original](\epsilon) > E_i(\epsilon)$（$i$ は Pattern1..8 の各対象の方式）となる．

5.1 エラーモデル

本解析では，**表 3** に示したパケットフォーマットを仮定する．

⋆1 ローカルコアには十分な量のバッファがあると想定し，そのバッファを利用できることとした．

© 2011 Information Processing Society of Japan

表 3　パケットフォーマット
Table 3　Packet format.

項目	パラメータ
フリットサイズ	データ長（64 ビット）と 符号幅（0-8 ビット）の和
ヘッダ・フリット内の 制御情報	計 16 ビット（目的地，出発地， シーケンス番号，パケット長）

表 4　各種エラー検出・訂正方式のデータ幅と符号長
Table 4　Data width and code length of each error detection and correction methods.

	Code 幅 （Header）	Data 幅 （Header）	Code 幅 （Data）	Data 幅 （Data）
Original	0	16	0	64
Pattern1	1	16	1	64
Pattern2	5	16	7	64
Pattern3	1	16	7	64
Pattern4	5	16	7	64
Pattern5	1	16	1	64
Pattern6	5	16	1	64
Pattern7	1	16	8	64
Pattern8	5	16	8	64

フリットサイズは 64 ビット以上を想定しているため，ヘッダフリットには，48 ビットの未使用領域が生じる．そこで，この未使用領域にヘッダフリットの使用領域のエラー訂正，検出符号を格納することとした．一方，ボディフリットには，64 ビット幅のデータと 0-8 ビットの符号が格納されていることとした．

また，本解析では各フリットのデータ（Data 幅）および符号部分（Code 幅）のビット幅を**表 4** に示すとおり設定した．

Header の Data 幅は宛先アドレスなどから成るルーティング情報のビット数であり，Code 幅はこれを符号化する際の符号長である．同様に Data の Code 幅は 64-bit データフリットの符号長である．なお，ヘッダの Code 部分はヘッダフリットの未使用領域に格納するものとする．なお，Pattern 5，6 の水平垂直パリティでは，各フリットにおける垂直パリティビットは 1 ビットであるが，別途，パケットごとに各フリットの水平パリティビットを格納した 1 つのフリットが必要となる．

さらにビットあたりのエラー率を ϵ と定義してエラーモデルを作成した．通常の NoC における 1 パケットが宛先に到着するまでのエラー率 $E_{Original}$ は以下のようになる．

なお，Data Width（Header Flit の有効データとその符号部の幅）を HDW，Data Width（Data Flit とその符号部の幅）を DDW とする．また，パケットの長さを P_{Length} とする．

まず，データフリットが宛先に到着するまでにビットエラーが生じる確率（$Ed_{Original}$）を求める．

$$\begin{aligned} \epsilon_{ee} &= 1-(1-\epsilon)^{hop} \\ Ed_{Original} &= 1-(1-\epsilon_{ee})^{DDW} \\ &\approx \epsilon \cdot hop \cdot DDW \end{aligned} \tag{4}$$

ϵ_ee は end-to-end で 1-bit のワイヤがエラーを起こす確率である．次に，ヘッダフリットが宛先に到着するまでにビットエラーが生じる確率（$Eh_{Original}$）を求める．

$$\begin{aligned} \epsilon_{ee} &= 1-(1-\epsilon)^{hop} \\ Eh_{Original} &= 1-(1-\epsilon_{ee})^{HDW} \\ &\approx \epsilon \cdot hop \cdot HDW \end{aligned} \tag{5}$$

最後に，それぞれの式からパケットが宛先に到着するまでにビットエラーが生じる確率（$Ep_{Original}$）を求める．

$$\begin{aligned} Ep_{Original} &= 1-(1-Ed_{Original})^{P_{Length}} \\ &\quad \cdot(1-Eh_{Original}) \\ &\approx \epsilon \cdot hop \cdot (HDW+DDW) \end{aligned} \tag{6}$$

式 (6) は十分 ϵ が小さいときの近似式であり，パケットのエラー率は ϵ に比例することが分かる．

同様に，エラー訂正を行う Pattern3〜6 のモデルについても以下のとおり検討する．まず，データフリットが宛先に到着するまでにビットエラーが生じる確率（$Ed_{[i]}$）を求める．また，添え字 i は Pattern3〜6 の番号に対応する．

$$\begin{aligned} \epsilon_{ee} &= 1-(1-\epsilon)^{hop} \\ Ed_{[i]} &= 1-\left\{(1-\epsilon_{ee})^{DDW}\right. \\ &\quad \left.+DDW\cdot\epsilon_{ee}\cdot(1-\epsilon_{ee})^{DDW-1}\right\} \\ &\approx \epsilon^2\cdot hop^2\cdot {}_{DDW}\mathrm{C}_2 \end{aligned} \tag{7}$$

次に，ヘッダフリットが宛先に到着するまでにビットエラーが生じる確率（$Eh_{[i]}$）を求

© 2011 Information Processing Society of Japan

める．

$$E_{flit} = 1 - \left\{(1-\epsilon)^{HDW} + HDW \cdot \epsilon \cdot (1-\epsilon)^{HDW-1}\right\}$$
$$Eh_{[i]} = 1 - (1 - E_{flit})^{hop} \approx \epsilon^2 \cdot hop \cdot {}_{HDW}\mathrm{C}_2 \quad (8)$$

最後に，それぞれの式からパケットが宛先に到着するまでにビットエラーが生じる確率（$Ep_{[i]}$）を求める．

$$Ep_{[i]} = 1 - (1 - Ed_{[i]})^{P_{Length}} \cdot (1 - Eh_{[i]}) \approx \epsilon^2 \cdot hop \cdot (hop \cdot {}_{DDW}\mathrm{C}_2 + {}_{HDW}\mathrm{C}_2) \quad (9)$$

式 (6) と同様に式 (9) は十分 ϵ が小さいときの近似式であり，ϵ^2 に比例する．このようにしてすべての方式におけるモデルを構築し近似式を求めた．

なお，本解析にはエラー訂正・検出符号によりデータ・フリットの大きさが Pattern ごとに異なるが，本解析では符号も含むデータ・フリットのエラー率を示している．

ビットエラーの発生確率（BER: Bit Error Rate）に対し，エラー耐性技術によって防ぎきれず実際にパケットエラーが生じる確率（PER: Packet Error Rate）を**図 3** に示す．エラー率が低い点（10^{-7} 以下）では近似式を利用して計算を行った．図 3 を見ると当然のことながら original に比べてエラー訂正・検出を行った各 Pattern の PER は低くなっている．Pattern 間の差は少ないが，なかでも Pattern7，8 が最もエラー訂正能力が高いことが分かる．

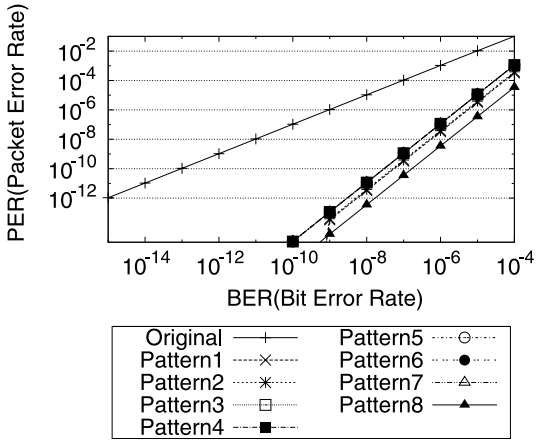

図 3　ビットエラー（BER）に対し，実際にパケットエラーが生じる確率（PER）
Fig. 3　The probability of packet errors for each bit error rate (BER).

5.2　電圧比の導出

通常の NoC およびそれぞれのエラー検出・訂正方式を用いた NoC において，フリットをエラー率 E で転送するために必要な電圧を求める．5.1 節で求めたエラーモデルより，フリットエラー率が E のときの 1 ホップあたりのビットエラー率 ϵ_i（i は $Pattern1, 2, .., 8$ の各対象の方式）は次のように表せる．

$$E = E_{original}(\epsilon_{original}) = E_i(\epsilon_i) \quad (10)$$
$$\epsilon_{original} = E^{-1}_{original}(E) \quad (11)$$
$$\epsilon_i = E_i^{-1}(E) \quad (12)$$

また，エラー耐性技術を実装していない通常の NoC の電源電圧を $Vdd_{original}$ とし，エラー耐性技術を実装した場合を Vdd_i とすると，前式を用いて以下のように電圧とフリットエラー率の関係が得られる[14]．

$$\frac{Vdd_{original}}{2\sigma_N} = Q^{-1}(\epsilon_{original}) \quad (13)$$
$$\frac{Vdd_i}{2\sigma_N} = Q^{-1}(\epsilon_i) \quad (14)$$

これらの式を変形し，σ_N を消すことで同じフリットのエラー率の場合の通常の NoC に対するエラー耐性技術を用いた NoC の電圧比の式が得られる．

$$\frac{Vdd_i}{Vdd_{original}} = \frac{Q^{-1}(\epsilon_i)}{Q^{-1}(\epsilon_{original})} \quad (15)$$

式 (15) より電圧比が求められる．すなわち，エラー耐性技術を導入したことで，エラー率を維持したままどれだけ電圧を下げることができるか分かる．

5.3　エネルギーの計算

パケットを送信元から宛先まで転送するための消費エネルギー（En）は，ホップ数を hop，パケット長を L フリット，ACK/NACK パケット長を ACK フリット，データ幅を W ビットとするとき次式で計算できる．

© 2011 Information Processing Society of Japan

$$En = \Big\{ En_{SWtoSW} \cdot W \cdot (hop - 1) + En_{SWtoNode} \cdot W \cdot 2 + En_{link} \cdot W \cdot hop \Big\} \cdot (L + ACK) \quad (16)$$

ここで，En_{SWtoSW} および $En_{SWtoNode}$ は，それぞれ 1-bit あたりのルータ–ルータ間の転送に要するエネルギーと 1-bit あたりのルータ–ノード間の転送に要するエネルギーであり，符号化によるエネルギーを含んでいる．ここでは En_{SWtoSW} と $En_{SWtoNode}$ は等しいものとする．

また，各リンクにおける転送エネルギー（En_{link}）は次式から求めた．

$$En_{link} = dV^2 C_{wire}/2 \quad (17)$$

ただし，d を 1 ホップあたりの平均距離，V を動作電圧，C_{wire} を配線容量とする．ここで，リンク間における配線容量はおよそ 2 pF/cm とした[15]．また，1 ビットあたりのリンク間の距離を 1 mm とし，電圧を 1.1 V とすると En_{link} は 0.121 pJ となる．

簡単のため，前式にはエラーによる再送エネルギーを含めていないが，以降の評価では再送エネルギーも考慮している．

6. 評　　価

本評価では，まず，ルータにおけるエラー訂正機構のエネルギー，ハードウェアの面積オーバヘッドを示し，次に，これらの結果から要求されるエラー率に対応して必要となる電源電圧を評価する．最後に，これらをふまえたパケット転送で必要となるフリットあたりのエネルギーを示す．エラーの発生頻度に関しては，関連研究[7] で扱っている範囲とほぼ同様（10^{-6}〜10^{-15}）を対象とした．エラーの発生頻度に関しては，対象デバイス，動作環境に依存するため，ある程度の範囲に対して評価を行う必要がある．そこで，5 章において文献 14) より展開した式 (13)〜(15) に基づき，これらの各エラーレートに関して相対的に削減可能な電圧，エネルギーを求める．

6.1 評 価 環 境

本章における評価環境を**表 5** に示す．

メッシュトポロジにおける次元順ルーティングの場合，仮想チャネルがなくても，デッドロックフリーを実現することができる．しかし，現状，多くのメッシュトポロジを持つ実 NoC は，性能向上あるいはプロトコルデッドロックを避けるために多数の仮想チャネルを持っていることも考慮し，すべてのプロトコルにおいて，平等に仮想チャネルを 2 本とした．

表 5　評価環境
Table 5　Simulation environments.

項目	パラメータ
トポロジ	2 次元メッシュ
ルーティング	次元順ルーティング
仮想チャネル数	2 本（エラー検出・訂正回路は共有）
ルータのバッファ容量（FIFO）	8 フリット分
パケット長	6（うち，ヘッダが 1）
フリットサイズ	データ幅（64 ビット）と 符号幅（0-8 ビット）の和

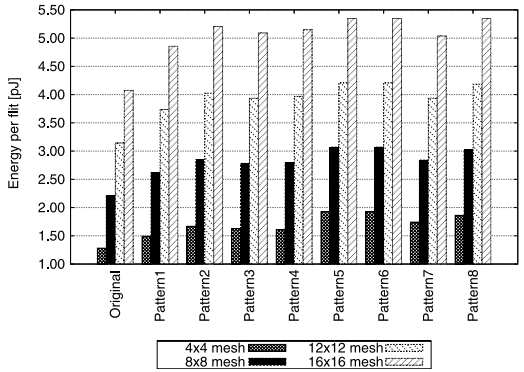

図 4　1 フリットあたりの転送エネルギー（通常電圧時）
Fig. 4　Packet energy of a single flit (normal power supply).

6.2 通常電圧時の消費エネルギー

4 章で示したオンチップルータを Verilog-HDL を用いて設計し，これらを NANGATE 45 nm の CMOS プロセスにより Design Compiler 2007.12-SP3 を用いて論理合成，SoC Encounter 8.1 を用いて配置配線を行った．この設計データに基づき，パケット転送エネルギー，面積オーバヘッドを算出した．

まず各エラー検出・訂正方式における 1 フリットあたりの通常電圧時（1.1 V）の消費エネルギーを算出した．様々なネットワークサイズにおける消費エネルギーを**図 4** に示す．評価には k-ary n-mesh におけるユニフォームトラフィックの平均ホップ数が，$hop = k \times n/3$ となること[12] を利用し，計算を行った．なお，この消費エネルギーには，エラー検出の手

© 2011 Information Processing Society of Japan

法を用いた場合のエラーによる再送エネルギーも含まれている．また，Pattern7，8 では ACK パケットの消費エネルギーも含まれている．

Original がエラー検出・訂正をいっさい行わないベースラインとなる NoC の転送エネルギーである．Pattern1 はエネルギーオーバヘッドが最小で 4 × 4 mesh では 16%のエネルギーオーバヘッドであった．一方，Pattern6 ではエネルギーオーバヘッドが最大であり 4 × 4 mesh で 39%に達した．end-to-end 方式では，使われている符号方式によって，リンク間のビット幅が決まる（表 4）．そのため，end-to-end でのエラー検出・訂正機構のオーバヘッドは，単純なエンコード・デコードによるエネルギー消費だけではなく，データ幅の増加にともなうオーバヘッドが存在する．Pattern3 と Pattern4，Pattern5 と Pattern6，Pattern7 と Pattern8 はそれぞれ end-to-end で同じエラー検出・訂正方式を採用しているため，各グループごとで似た結果になっている．また，Pattern5，6 ではルータのノードに対する出力ポートに水平垂直パリティ符号でエラー訂正するため，一時的にデータを 1 パケット分保存するバッファが必要となる．このため，ホップ数が短い場合はエネルギーオーバヘッドが大きくなる．同様に，Pattern7，8 ではノードに対する出力に ACK を返信するユニットを実装しているため，ホップ数が短い場合はエネルギーオーバヘッドが大きくなる．

6.3　ルータの面積オーバヘッド

ルータの面積オーバヘッドの評価結果を**図 5** に示す．エラー訂正・検出機構を導入することで，ルータ面積は最大 41%，最小でも 25%大きくなった．

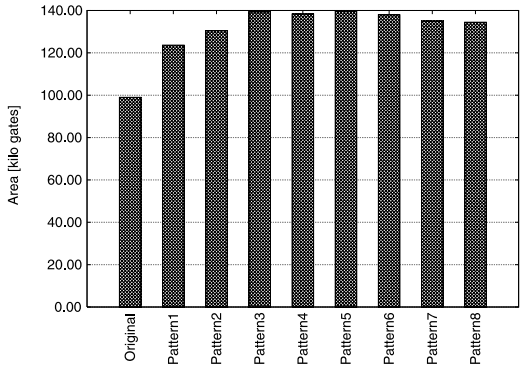

図 5　耐故障ルータの面積オーバヘッド
Fig. 5　Area overhead of fault-tolerant routers.

Pattern1 が最も小さい理由は，ビット幅が小さくて済むためである．一方，Pattern5，6 はビット幅が小さいにもかかわらず，面積が大きい．これは，ルータのノードに対する出力ポートに，水平垂直パリティ符号でエラー訂正するために一時的にデータを 1 パケット分保存するバッファが必要となるためである．なお，評価に用いたルータは，単純なアービタ，ルーティングテーブルを用いないソースルーティングなど基本的な構成であり，より高性能なオンチップルータの場合にはオーバヘッドは相対的に小さくなると考えられる．

6.4　要求されるパケットエラー率と電源電圧

まず，通常の NoC を用いたとき，要求されるパケットエラー率（PER）に対して必要な電源電圧を求め，これを $Vdd_{original}$ とする．同様にエラー耐性技術を用いた NoC についても要求される PER に対して必要な電源電圧を求め，Vdd_{coded} とする．4 × 4 mesh NoC についてこの電圧比 $Vdd_{coded}/Vdd_{original}$ を算出し，1 パケットが転送されるときの要求エラー率を横軸にとって**図 6** に示す．この比が小さいほど，同じエラー率を実現する場合に，エラー耐性技術によって電圧を下げることができた割合が大きいことを示す．なお，グラフ中の Pattern3，4，5，6 の結果は重なって表示されている．

次に平均ホップ数が増加した場合の Vdd とエラー率の関係を見るため，16 × 16 mesh の NoC におけるエラー率を同様に評価し，**図 7** に示す．

図 6 および図 7 で，Pattern1，2 に対して Pattern3，4，5，6 の比率が大きくなってし

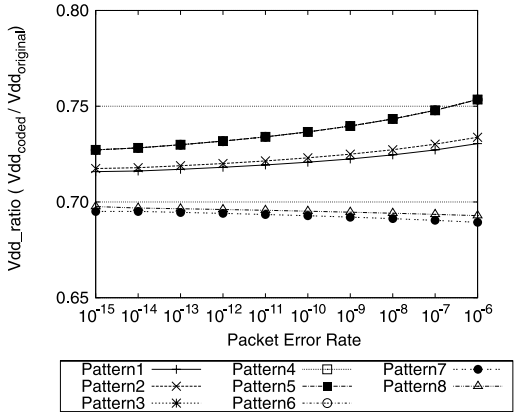

図 6　要求 PER に対する電圧比（4 × 4 mesh NoC）
Fig. 6　Ratio of power supply for each required PER (4 × 4 mesh NoC).

© 2011 Information Processing Society of Japan

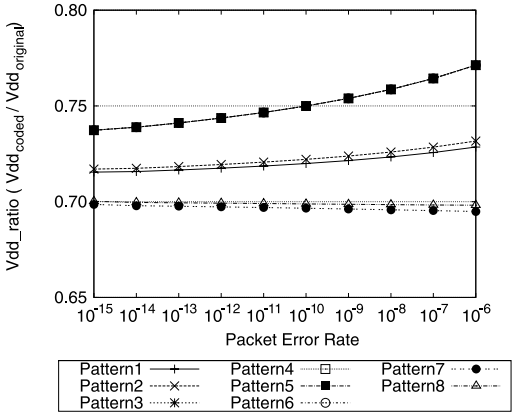

図 7　要求 PER に対する電圧比（16 × 16 mesh NoC）
Fig. 7　Ratio of power supply for each required PER (16 × 16 mesh NoC).

まっている原因はデータフリットに対して end-to-end でのみ 1 誤り訂正の符号を用いているためである．Pattern1，2 は各ルータスイッチで 1 誤り検出・訂正をしているため，より電圧を下げることが可能となる．一方，Pattern7，8 は end-to-end で CRC8 を利用しているため，8-bit までのバーストエラーを検出でき，より電圧を下げることが可能となる．また，パリティやハミングなどのエラー検出・訂正方式では，要求されるエラー率が小さくなるにつれて，高い電圧が要求されるが，CRC8 ではバーストエラーに対応しているため，要求されるエラー率が小さくなってもほぼ一定の電圧まで下げることができる．

図 6 と図 7 を比較すると，16 × 16 mesh の方が Pattern3 から Pattern6 までに関しては要求される電圧が高くなっている．これは平均ホップ数が増加しているためである．一方で，各ホップでエラー検出を行う Pattern1，2 および CRC8 を用いたバーストエラーに対する耐性のある Pattern7，8 は，要求電圧がほとんど変化しないことが分かる．

6.5　エラー率とエネルギー削減効果

各エラー耐性方式における 1 パケットあたりの通信エネルギーを求め，電圧とエラー率の関係から各エラー率におけるエネルギー比を求めた．**図 8** に 4 × 4 mesh 構成，**図 9** に 16 × 16 mesh 構成の NoC において 1 パケットを転送するのに必要なエネルギーをそれぞれ示す．

エネルギーは電圧比の 2 乗に比例するため，エラー検出・訂正機構によるエネルギーオーバヘッドを考慮しても，いずれの方式もエネルギーを 30%以上削減することができている．また，4 × 4 mesh において，エラー率が 10^{-12} 以下では Pattern1 のハミング符号を用いたリンク再送方式が最もエネルギーを削減できた．この手法はエラー検出のための符号ビッ

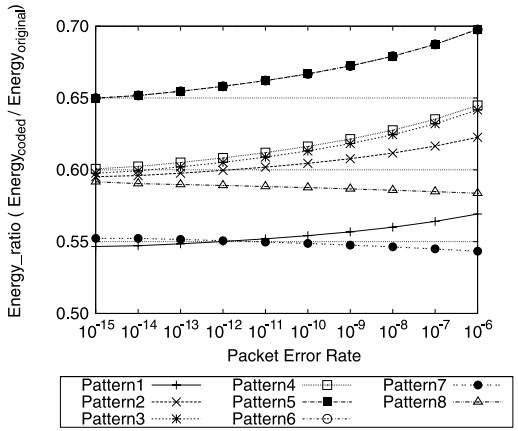

図 8　要求 PER に対するエネルギー比（4 × 4 mesh NoC）
Fig. 8　Energy ratio for each required PER (4 × 4 mesh NoC).

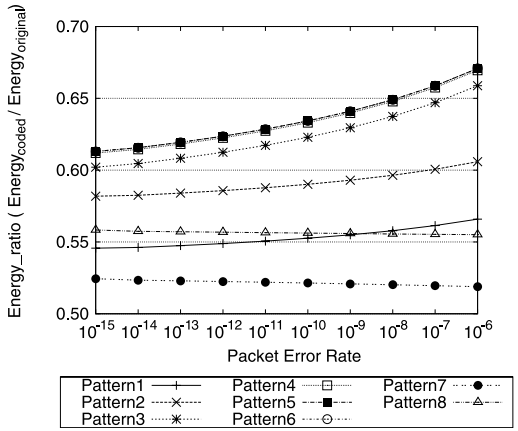

図 9　要求 PER に対するエネルギー比（16 × 16 mesh NoC）
Fig. 9　Energy ratio for each required PER (16 × 16 mesh NoC).

© 2011 Information Processing Society of Japan

トが少なくかつ各ルータでエラー検出を行うためエラー検出能力が高い．一方，4×4 mesh において 10^{-12} よりエラー率が高い場合と，16×16 mesh においては Pattern7 がすべての方式の中で最小であり，通常の NoC に対して 55%を下回るエネルギーで動作する．

評価の結果，エラー率を保ったままで，電源電圧を下げてエネルギー効率を上げるという点では Pattern7，すなわちヘッダフリットは各ルータでパリティによる検出を行い，end-to-end でデータフリット部について CRC8 によりエラー検出を行う手法が優れていることが分かった．しかし，Pattern1，すなわちヘッダとデータともに各ルータでパリティを用いる方法も Pattern7 に近い効率を実現し，かつハードウェアオーバヘッドが少ない方法であり，ルータのハードウェアを増やしたくない場合は有利である．

7. ま と め

本稿ではエラー耐性技術を用いることで，エラー率を抑えつつ電圧を下げる NoC の低消費電力化手法を提案した．まず，現実的なエラー耐性技術を 8 つのパターンに絞り，これをすべて適用可能なルータ構成を提案した．このルータは，通信レイテンシを抑えるため，エラー検出・訂正処理を通常のルータのルーティング処理と同時に行い，エラーが検出された場合のみ処理をやり直す投機的な実行を採用した．また，空いている入力チャネルバッファを再送に利用することで専用のバッファを用いない低コストな再送機構を実現している．

評価結果より，電源電圧を下げてエネルギー効率を上げるという点において，ヘッダフリットは各ルータでパリティによるエラー検出を行い，データフリット部には end-to-end の CRC8 により検出を行う手法が優れていることが分かった．このとき通常の NoC と比較してエネルギーを 45%削減でき，面積オーバヘッドは 36.4%となった．一方，ヘッダとデータともに各ルータでパリティを用いる方法も面積オーバヘッドが少ない（25%）割に高いエネルギー削減を実現できる方法であることが分かった．

謝辞　本研究の一部は，科研費（若手研究（B）22700061）の援助による．

参 考 文 献

1) Lee, K., Lee, S.-J. and Yoo, H.-J.: Low-Power Network-on-Chip for High-Performance SoC Design, *IEEE Trans. Very Large Scale Integration* (*VLSI*) *Systems*, Vol.14, No.2, pp.148–160 (2006).
2) Beigne, E., Clermidy, F., Miermont, S. and Vivet, P.: Dynamic Voltage and Frequency Scaling Architecture for Units Integration within a GALS NoC, *Proc. International Symposium on Networks-on-Chip* (*NOCS'08*), pp.129–138 (2008).
3) Shivakumar, P., Kistler, M., Keckler, S.W., Burger, D. and Alvisi, L.: Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic, *Proc. International Conference on Dependable Systems and Networks* (*DSN'02*), pp.389–398 (2002).
4) Kobayashi, H., Kawamoto, N., Kase, J. and Shiraish, K.: Alpha Particle and Neutron-Induced Soft Error Rates and Scaling Trends in SRAM, *Proc. International on Reliability Physics Symposium* (*IRPS'09*), pp.206–211 (2009).
5) Park, D., Nicopoulos, C., Kim, J., Vijaykrishnan, N. and Das, C.R.: Exploring Fault-Tolerant Network-on-Chip Architectures, *Proc. International Conference on Dependable Systems and Networks* (*DSN'06*), pp.93–104 (2006).
6) Bertozzi, D., Benini, L. and DeMicheli, G.: Error Control Schemes for On-Chip Communication Links: The Energy-Reliability Tradeoff, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.24, No.6, pp.818–831 (2005).
7) Pande, P.P., Ganguly, A., Feero, B., Belzer, B. and Grecu, C.: Design of Low Power and Reliable Networks on Chip Through Joint Crosstalk Avoidance and Forward Error Correction Coding, *Proc. International Symposium on Defect and Fault Tolerance in VLSI Systems* (*DFT'06*), pp.466–476 (2006).
8) Matsutani, H., Koibuchi, M., Wang, D. and Amano, H.: Adding Slow-Silent Virtual Channels for Low-Power On-Chip Networks, *Proc. International Symposium on Networks-on-Chip* (*NOCS'08*), pp.23–32 (2008).
9) Hazucha, P. and Svensson, C.: Impact of CMOS Technology Scaling on The Atmospheric Neutron Softerror Rate, *IEEE Trans. Nuclear Science*, Vol.47, No.6, pp.2586–2594 (2000).
10) Chandra, V. and Aitken, R.: Impact of Technology and Voltage Scaling on the Soft Error Susceptibility in Nanoscale CMOS, *Proc. International Symposium on Defect and Fault Tolerance of VLSI Systems* (*DFT'08*), pp.114–122 (2008).
11) Murali, S., Theocharides, T., Vijaykrishnan, N., Irwin, M.J., Benini, L. and DeMicheli, G.: Analysis of Error Recovery Schemes for Networks on Chips, *IEEE Design and Test of Computers*, Vol.22, No.5, pp.434–442 (2005).
12) Dally, W.J. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).
13) Hegde, R. and Shanbhag, N.: Toward Achieving Energy Efficiency in Presence of Deep Submicronnoise, *IEEE Trans. Very Large Scale Integration Systems*, Vol.8, No.4, pp.379–391 (2000).
14) Gebali, F., Elmiligi, H. and El-Kharashi, M.W.: *Networks-on-Chips: Theory and Practice*, CRC Press (2009).
15) ITRS: *International Technology Roadmap for Semiconductors 2007 Edition Interconnect* (2007), available from ⟨http://www.itrs.net/⟩.

© 2011 Information Processing Society of Japan

(平成 23 年 1 月 27 日受付)
(平成 23 年 7 月 5 日採録)

小島　悠

平成 20 年慶應義塾大学理工学部情報工学科卒業．平成 22 年同大学大学院理工学研究科開放環境科学専攻修士課程修了．同年ソニー（株）入社．現在，カメラ信号処理技術の開発・設計に従事．

松谷　宏紀（正会員）

平成 16 年慶應義塾大学環境情報学部卒業．平成 20 年同大学大学院理工学研究科開放環境科学専攻博士課程修了．博士（工学）．現在，慶應義塾大学理工学部情報工学科専任講師．平成 21 年度より 22 年度まで日本学術振興会特別研究員 SPD．計算機アーキテクチャ，オンチップネットワークの研究に従事．

鯉渕　道紘（正会員）

平成 12 年慶應義塾大学理工学部情報工学科卒業．平成 15 年同大学大学院理工学研究科開放環境科学専攻博士課程修了．博士（工学）．平成 14 年度より 16 年度まで日本学術振興会特別研究員．現在，国立情報学研究所准教授，総合研究大学院大学複合科学研究科情報学専攻准教授（兼任）．ハイパフォーマンスコンピューティングとインターコネクトに関する研究に従事．IEEE Computer Society Japan Chapter Young Author Award 2007，平成 19 年度情報処理学会論文賞受賞．IEEE，電子情報通信学会各会員．

天野　英晴（正会員）

昭和 56 年慶應義塾大学工学部電気工学科卒業．昭和 61 年同大学大学院工学研究科電気工学専攻博士課程修了．工学博士．現在，慶應義塾大学理工学部情報工学科教授．計算機アーキテクチャの研究に従事．

© 2011 Information Processing Society of Japan