論　文

複数コアリンクを用いた低遅延オンチップトポロジーに関する研究

河野　隆太[†a)]　藤原　一毅[††b)]　松谷　宏紀[†,††c)]　天野　英晴[†,††d)]
鯉渕　道紘[††,†††e)]

The Study of Low-latency On-chip Topology using Multiple Core Links

Ryuta KAWANO[†a)], Ikki FUJIWARA[††b)], Hiroki MATSUTANI[†,††c)],
Hideharu AMANO[†,††d)], and Michihiro KOIBUCHI[††,†††e)]

あらまし　近年のメニーコア・アーキテクチャでは，コアの数は増加の一途を辿っており，コア間の通信遅延が並列アプリケーションに与える影響が益々大きくなってきている．コア間の通信にはパケット・ネットワーク構造が広く用いられるため，コア間トポロジーが通信遅延に大きく影響する．これまでの我々の研究で，通信遅延を削減するために，規則的なルータ間トポロジーにランダムに選択したルータ間リンクを付加する方法が有効であることが分かっている．そこで，本研究ではこの研究を拡張し，単一コアとランダムに選択した近隣のルータとの間にリンクを追加し，コアからのリンクを複数とする方法を提案・評価する．フリットレベルのネットワークシミュレーションの結果，ランダムコアリンクを用いた我々のトポロジーは，従来のトポロジーに比べ，平均遅延を最大 45%減少させた．更に．フルシステム CMP シミュレーションの結果，NAS 並列ベンチマークのアプリケーション実行時間を最大 10.9%向上させることが分かった．

キーワード　チップ内ネットワーク，トポロジー，相互結合網

1. まえがき

近年のメニーコア･計算機アーキテクチャでは，コア数が増加の一途を辿っている．そのため，コア間の通信遅延がアプリケーションに与える影響が益々大きくなってきている．コア間の通信にはパケット転送を用いたネットワーク構造（Network-on-Chip, NoC）[1] が広く用いられるため，コア間ネットワークトポロジーが通信遅延に大きく影響する．

従来，チップ内ネットワークのトポロジーの研究はトポロジーをルータを頂点としたグラフにモデル化し，そのグラフの直径，平均距離，チップへのレイアウトの改良に焦点をあてることが多かった．しかし，この生成グラフに基づくネットワークトポロジーでは，全ての end-to-end 通信遅延に対して，最初と最後の 1hop，すなわちコア–ルータ間の遅延を別途加算することになる．この遅延がオーバーヘッドとなるため，ルータ間トポロジーの最適化による遅延削減の効果は頭打ちとなる．

一方，最近の我々の報告において，オフチップでの相互接続網における低遅延トポロジーを提案している [2]．この報告では，ホストからの複数リンクをショートカットとして用いることで，従来のルータ間トポロジーの限界を超えた end-to-end 通信遅延の削減効果が得られることが分かっている．更に，ホストからの各リンクの接続先ルータをランダムに選択することで，効率的に遅延を削減できることも示されている．

ノード数を N としたとき，2 ノード間の距離（最小経由ノード数）の平均が $\log N$ に比例するグラフはスモールワールドグラフと呼ばれ，そのようなグラフが平均距離を小さくすることなどの性質をスモールワールド性と呼ぶ．また，スモールワールドグラフの

† 慶應義塾大学大学院理工学研究科，横浜市
Graduate School of Science and Technology, Keio University, Hiyoshi, 3–14–1 Kohoku-ku, Yokohama-shi, 223–8522 Japan
†† 国立情報学研究所，東京都
National Institute of Informatics, 2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430 Japan
††† 総合研究大学院大学複合科学研究科，神奈川県
School of Multidisciplinary Sciences, SOKENDAI, Shonan Village, Hayama, Kanagawa-ken, 240–0193 Japan
a) E-mail: kawano@am.ics.keio.ac.jp
b) E-mail: ikki@nii.ac.jp
c) E-mail: matutani@arc.ics.keio.ac.jp
d) E-mail: hunga@am.ics.keio.ac.jp
e) E-mail: koibuchi@nii.ac.jp

一つに，ノード間を不規則に接続したランダムグラフがある [3]．先述の我々の報告 [2] では，ホスト間トポロジーをランダムグラフに近づけることで，スモールワールド性を用いてホスト間の経由ルータ数を減らし，通信遅延の削減を図っている．

そこで，我々はこの成果をチップ内ネットワーク向けに応用した低遅延トポロジーを提案する．すなわち，既存のルータ間トポロジーにおいて，コアから異なるランダムに選択した近隣ルータへリンクを直接接続し，従来のローカルなコア–ルータ間の接続と合わせてコアからのリンクを複数とする．

規則的なルータ間トポロジーでのルーティング手法には，Mesh トポロジーでの次元順ルーティングなどの単純なルーティング手法が一般的に使用される．一方，不規則なルータ間トポロジーでは，up/down ルーティングなどのトポロジーに依存しないルーティング手法が使われる．このようなルーティング手法では規則的なトポロジーのルーティング手法とは異なり，ルーティングテーブルの保持が必要となる．オフチップネットワーク向けの従来提案であるルータ間ランダムリンクの追加手法 [4] をそのままチップ内ネットワークに適用した場合，ルータ間が不規則なトポロジーとなるため，ルーティングテーブルの保持によるルータの面積増加や消費電力の増加などのオーバーヘッドが発生する．一方，本研究の提案は “コア–ルータ間” にリンクを追加する手法であり，ルータ間トポロジーは既存のものが維持されるため，既存のルーティング機構を利用でき，ルータでのオーバーヘッドを最小限に抑えられる．すなわち，“コア–ルータ間” にランダムリンクを追加することにより，既存のルータ間トポロジーを維持しつつコア間をランダムトポロジーに近づけ，スモールワールド性を用いた通信遅延の削減を図る．

また，オフチップネットワークでは長距離配線による遅延はスイッチング遅延に比べて小さく，end-to-end の通信遅延に対する影響は限定的である．一方，NoC においては，長距離配線はネットワークの動作周波数を決定づけるため，end-to-end の通信遅延に対する影響は無視できなくなる．そこで，本研究では，各追加リンクの配線長を一定以下に抑える制限を課すことで，オフチップ向けの我々の提案 [4] を NoC 向けに適応させることを図る．

以後，**2.** において関連研究を述べ，**3.** において，二次元 NoC におけるランダムコアリンクを用いたトポロジーの提案及び end-to-end 通信遅延の解析を行う．更に **4.** において，ランダムコアリンクトポロジーを三次元 NoC に拡張する提案を行う．**5.** においては，ネットワークシミュレーションによる遅延及びスループットの測定を行う．**6.** ではフルシステムシミュレーションによるアプリケーション実行性能の評価を行い，**7.** でランダムコアリンクに関する議論を行った後，**8.** においてまとめと今後の課題を述べる．

2. 関 連 研 究

2.1 低遅延通信技術

チップ内ネットワークにおいて，これまで 1 サイクル，あるいは 2 サイクルでパケットを転送可能な低遅延ルータ [5]，トラヒックパターン，負荷に応じて混雑を避ける経路を動的に選択する適応型ルーティング，ワームホールスイッチングの利用など，種々の低遅延通信技術が設計されてきた．特にワームホールスイッチングの利用により，長いパケットの end-to-end 通信遅延に対する経由（ルータ）ホップ数が与える影響を抑えることができるようになった．

しかし，パケットサイズは通常，極めて短い場合が多い．例えば，TRIPS では，On-Chip Network (OCN) におけるトラヒックはメモリ転送が多く，キャッシュの line size である 64-Byte 転送は 5-flit パケットに，データ転送要求などの制御用の通信は 1-flit パケットとして転送する．更に，Operand Network (OPN) には演算データが流れるが，90% のパケットは 99-bit 以下であり，1-flit パケットに収まる [6], [7]．このような場合，通信遅延を削減するためには，トポロジー自体の改良が重要となる．

2.2 NoC トポロジーとレイアウト

直径，平均距離の小さなトポロジーとそのレイアウトについては，様々な提案が成されている．図 1 に典型的なトポロジーを示す．各丸頂点はコア，黒四角頂点はルータを示す．**3.3**，**4.2** のグラフ解析では，ルータ間トポロジーに 2-D MESH（図 1(a)），2-D TORUS（図 1(b)），H-TREE（図 1(c)），HYPER-CUBE（図 1(d)）を採用し，**5.** のネットワークシミュレーション，**6.** のフルシステムシミュレーション，及び **7.2** のグラフ解析ではルータ間トポロジーに 2-D MESH を採用した．

これらの規則的なトポロジーに加えて，ランダムトポロジーについても議論がなされている．ランダムグラフを用いたネットワークが，そのスモールワールド性により，従来の規則的なトポロジーに比べて直径

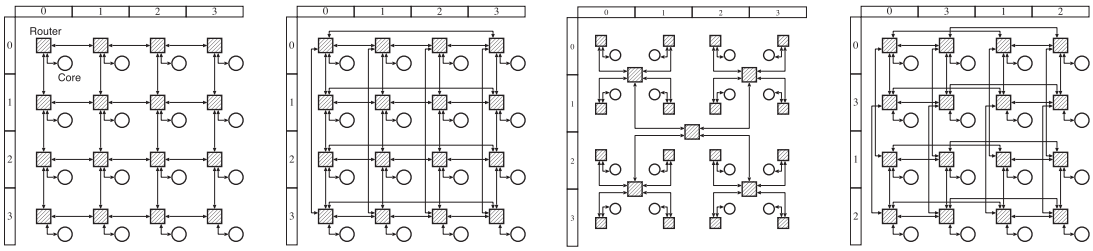

図 1　典型的な接続網の二次元レイアウト（16 コア）
Fig. 1　Two-dimensional layouts of typical interconnects (16-core).

や平均最短距離を小さくできることが報告されている [4]．このようなネットワークは通信遅延の削減効果に対して配線長が大きくなる傾向があるが，配線長を抑えるようなレイアウトの工夫が提案されており [8]，その上で十分な性能が期待される [9]．FPGA の配線についてもスモールワールド性の効果が報告されている [10]．

また，物理配線を減らしつつ低遅延化・高帯域化を実現するために 60GHz 無線を用いるチップ内ネットワークが提案されている [11]．ランダムトポロジーに加えて，Flattened Butterfly [12] に代表される高次元ネットワークではチップ内レイアウトにおいて長距離リンクが生じる．60GHz 無線リンクは，チップ内において物理距離が離れたルータ間を直接接続する場合に効果的である．

更に，チップの三次元実装 [13], [14] が登場しつつある．複数枚のウェハまたはダイを垂直方向に重ね合わせることで個々のチップサイズを小型化できるため，実装面積や配線長の削減が期待されている．このような状況を背景に，2005 年頃より三次元 IC 向けの NoC の研究が報告されている [15], [16]．

本研究では，これらの多岐にわたる二次元及び三次元 NoC に適用することができるコア–ルータ間リンクをランダムに追加するトポロジー技術を以降の章で探求する．

3.　二次元ランダムコアリンク・トポロジー

3.1　二次元ランダムコアリンクの接続

我々が提案する二次元 NoC トポロジーにおけるランダムコアリンクの接続を図 2 に示す．この図においてランダムコアリンクを点線で示しており，通常のルータ間トポロジーに対し，各コアからランダムに選択したルータに対し追加リンクを接続する．

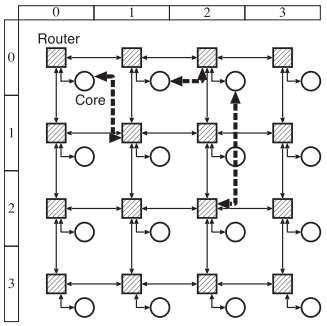

図 2　ランダムコアリンクの接続 (4x4 MESH)
Fig. 2　Adding random-core links (4x4 MESH).

コア当りの追加リンクの本数 (x) は各コア間で等しく，一つのコアからのリンクは互いに異なるルータに繋がるものとする．また，コア–ルータ間のリンク追加に伴うルータの増加ポート数は各ルータ間で等しいとする．

更に，追加リンクの長さを制限することによる性能の変化を調べるため，トポロジー内のランダムコアリンクの長さの上限である接続半径 (y) を設定した．ここで，図 1 や図 2 のように各ルータの位置を示す座標を導入し，隣接 2 ルータ間の距離（コア長と呼ぶ）を 1 とする．以降の説明では，任意の 2 ルータ間のマンハッタン距離の単位としてコア長を用いる．そして，y は，各コアのローカルルータ（ランダムリンクを用いない場合に各コアが接続しているルータ）と，そのコアが接続可能なルータとの間のマンハッタン距離（単位はコア長）の上限を表す．

3.2　ランダムコアリンクを用いた通信手法

ランダムコアリンクを用いたトポロジーを用いて通信を行うためには，コアからのパケット送信時に，使用するコアリンクを複数の中から一つ選択する必要がある．そこで，パケットの送信先に応じて，最もホッ

プ数を小さくできるようなコアリンクを選択することで，遅延削減を図る．このようなコアリンクの選択を行うためには，コア上にリンク選択のためのルーティング処理機構が必要となる．

ただし，本研究では，文献[2] やツリーベースのトポロジーに関する文献[17] と異なり，各コアは（経路上の中間ルータとして）他のコアから送信されたパケットを別のコアへ転送することは行わない．これは，(1) 文献[2] によると，コア上でのパケット転送による end-to-end 通信遅延の削減効果が限定的であるという点，(2) 文献[17] によると，中間ルータとしてのパケット転送処理をさせる場合，コアのハードウェア量が相応に増加してしまうという点からの判断である．

また，単一コアが同時に送信するパケットは一つのみとする．すなわち，送信先の異なる複数のパケットが，互いに異なるコアリンクを用いて同時にコアから送信されることはないものとする．同様に，単一コアが同時に受信するパケットは一つのみとする．

以上のようなランダムコアリンクを用いた通信手法は，以降の章においても同様に用いる．

3.3 解　　析

本章ではグラフ解析を用いることにより，従来のトポロジーと複数ランダムコアリンクを用いたトポロジーの比較評価を行う．

ネットワークトポロジーの生成スクリプトは Ruby で記述し，疑似乱数を生成することで，ランダムコアリンクを生成した．そして，そのように生成したネットワークトポロジーに対し，R 言語と igraph ライブラリを用いた Zero-load 遅延の解析を行った．Zero-load 遅延として，経路上におけるリンク及びルータの遅延を考慮した，コア間の最小遅延の最大値及び平均値を求めた．

以後，評価対象のトポロジーは”TOPOLOGY-x-y”と表記する．この表記は，TOPOLOGY がルータ間トポロジー，x がコア当りのランダムリンクの本数，y がランダムコアリンクの長さの上限を表す．$x = 0$ のとき，存在するコア–ルータ間のリンクは，同じ二次元座標に位置するコア–ルータ間のローカルなリンクのみである．

本章では，グラフ解析を用いた遅延評価の対象となるルータ間トポロジーとして，2D-MESH, 2D-TORUS, H-TREE, HYPERCUBE の四つを採用した．各トポロジーのレイアウトについては，直感的な方法（図 1）に基づいて行う．すなわち，トポロジーを複雑に折り畳んだ配置や配線の斜め配置を行わない，通常の NoC で広く用いられる配置・配線の手法を採用する．また，ルータ間トポロジーが H-TREE の場合は，コアからのランダムリンクの接続先として，ハブとなっているルータは選ばないようにした．これは，ハブルータをランダムコアリンクの接続先として選んだ場合，ポート数増加とそれに伴う面積増加が懸念されるためである．

あるコアから別のコアに 1 フリット転送するとき，コアからローカルルータ，ローカルルータからコアへの転送遅延を 1[cycle]，ルータを通過する遅延を 2[cycles]，そしてルータ間の配線での転送遅延を 1 コア長当り 1[cycle / コア長] とした．

3.3.1 ランダムコアリンク数

はじめに，各コアにおけるランダムリンク数 (x) を増加させた場合の平均 Zero-load 遅延の変化について解析する．

図 3〜6 は 16, 64 コアの NoC トポロジーにおける

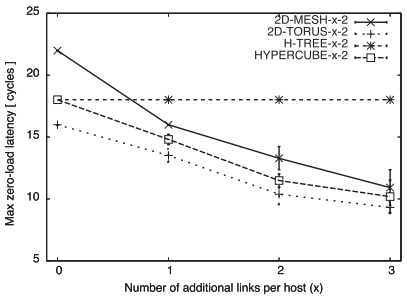

図 3　16 コア，$y = 2$ における各トポロジーのランダムコアリンク数と最大 Zero-load 遅延

Fig. 3　Max zero-load latency versus the number of random-core links for each topology (16 cores, $y = 2$).

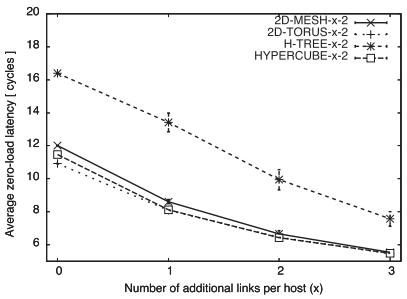

図 4　16 コア，$y = 2$ における各トポロジーのランダムコアリンク数と平均 Zero-load 遅延

Fig. 4　Average zero-load latency versus the number of random-core links for each topology (16 cores, $y = 2$).

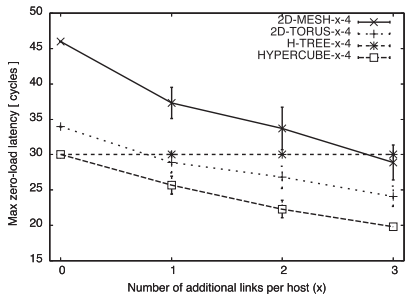

図 5　64 コア，$y = 4$ における各トポロジーのランダムコアリンク数と最大 Zero-load 遅延

Fig. 5　Max zero-load latency versus the number of random-core links for each topology (64 cores, $y = 4$).

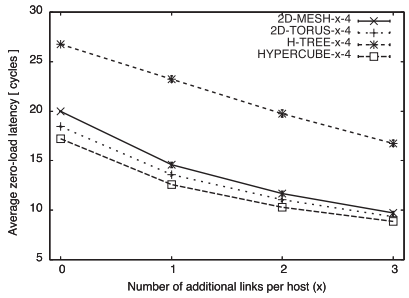

図 6　64 コア，$y = 4$ における各トポロジーのランダムコアリンク数と平均 Zero-load 遅延

Fig. 6　Average zero-load latency versus the number of random-core links for each topology (64 cores, $y = 4$).

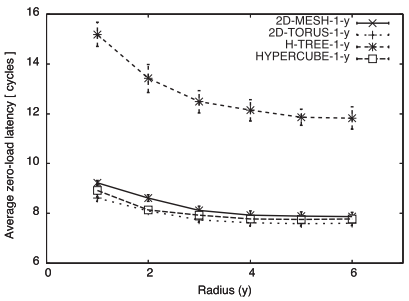

図 7　16 コア，$x = 1$ における各トポロジーのランダムコアリンク数と平均 Zero-load 遅延

Fig. 7　Average zero-load latency versus the radius (y) for each topology (16 cores, $x = 1$).

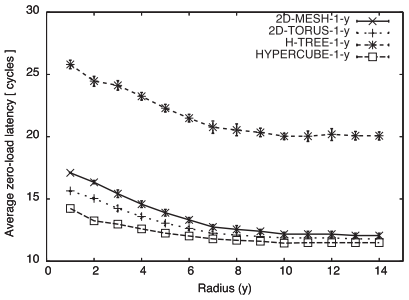

図 8　64 コア，$x = 1$ における各トポロジーのランダムコアリンク数と平均 Zero-load 遅延

Fig. 8　Average zero-load latency versus the radius (y) for each topology (64 cores, $x = 1$).

1 コア当りのランダムリンク数と最大・平均 Zero-load 遅延の関係を示している．ここで，ランダムリンクの接続先として許されるローカルコアからのコア距離 (y) を，16 コアでは 2, 64 コアでは 4 とした．また，各プロット値では，乱数を変えて 10 個のトポロジーを生成し，それぞれの計測値の平均値と標準偏差をとっている．

また，各図で各折れ線のラベルの変数（ここでは x）は横軸に対応しており，以後のグラフ解析の結果を示す各図でも同様とする．

これらの図より，最大 Zero-load 遅延についてはルータ間が H-TREE の場合を除く全ての場合で遅延を削減でき，平均 Zero-load 遅延については，全てのルータ間トポロジーにおいて遅延削減の効果が得られることが分かった．特に 2D-MESH の場合は，64 コアのトポロジーにおいて 1 コア当りランダムリンクを 3 本付加することにより，ランダムリンクを用いる前のトポロジーと比べて最大，平均の Zero-load 遅延をそれぞれ 37%，51%削減できた．

また，H-TREE が最大 Zero-load 遅延を減らすことができないのは，ランダムリンクの接続先であるルータが，H-TREE の階層構造の末端に位置するため，リンク付加による遅延削減の効果が局所的となってしまうことが原因と考えられる．

3.3.2　ランダムリンクの接続半径

ここでは，各コアにおけるランダムリンクのコア距離での接続半径 (y) を増加させたときの遅延の変化について解析する．この場合，接続半径が増えるほど接続先の候補となるルータ数が増えるため，コア間の遅延が小さくなることが予想される．

図 7, 8 は，16 コア及び 64 コアの NoC トポロジーにおけるランダムリンクの接続可能半径と平均 Zero-load 遅延の関係を示している．**3.3.1** と同様に，各プロット値は乱数の異なる 10 個のトポロジーについて，平均値と標準偏差を求めている．ここで，追加ランダムリンク数 (x) を 1 とした．4x4, 8x8 のコア配列

における最長のコア間距離はそれぞれ 6, 14 であるため，4x4 のコア配列では $y = 6$ のとき，8x8 のコア配列では $y = 14$ のとき，完全にランダムなコアリンクの接続が実現されている．

これらの図において，半径の増加に伴い遅延は単調に減少しているが，接続可能半径の増加に伴い，遅延削減の効果は次第に穏やかになっている．例えば，ルータ間 MESH, 64 コアにおいてランダムコアリンクを用いない場合と比べた平均遅延の削減率は，$y = 4$ で 27%，$y = 6$ で 33% であるのに対し，完全にランダム接続である $y = 14$ で 40% にとどまる．

よって，コアリンクの長さ（接続半径）をある程度小さくする設計でも十分に Zero-load 遅延を削減する効果が得られることが分かった．

4. ランダムコアリンク・トポロジーの三次元 NoC への拡張

本章では，これまでに提案したランダムコアリンクトポロジーを三次元 NoC に拡張する提案を行う．

4.1 三次元 NoC におけるランダムコアリンクトポロジーの接続

三次元 NoC トポロジーにおけるランダムコアリンクの接続を図 9 に示す．この図において，各コアから異なるチップ上のルータを含む全てのルータからランダムに選択したルータに対し点線で示したランダムコアリンクを接続している．

3.1 で示したとおり，各コア間での追加リンクの本数 (x) と各ルータ間での増加ポート数は等しく，コアからのリンクは互いに異なるルータに繋がるものとする．

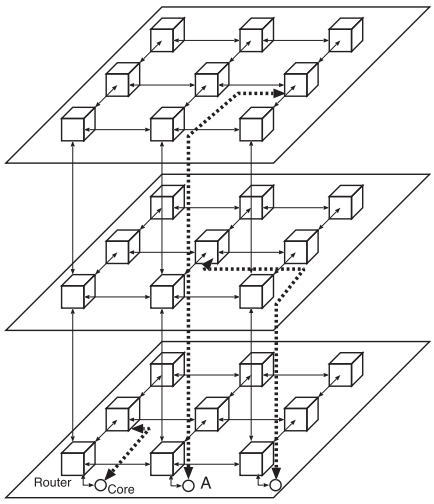

図 9　ランダムコアリンクの接続 (3x3x3 MESH)
Fig. 9　Adding random-core links (3x3x3 MESH).

また，接続半径 y については，ルータ間のマンハッタン距離をチップ上の二次元方向（x, y 方向）のみ考慮し，垂直方向（z 方向）は考慮しないものとする．すなわち，チップ間のマンハッタン距離を 0 とし，チップ上のマンハッタン距離のみを用いて接続半径を定義する．これは，チップ間の距離がチップ上のルータ間距離に比べて著しく小さいためである．例えば，図 9 において，“A” とラベル付けしたコアからのランダムコアリンクは，垂直方向の距離を 0 とするため，長さは 2[コア長] となる．このような定義のもとで，ランダムコアリンクの接続半径を設定した．

4.2 解　　析

本章では **3.3** と同様に，グラフ解析を用いたトポロジーの評価を行う．

この章における評価対象のトポロジーの表記は，”TOPOLOGY-*d*-*x*-*y*” とする．この表記は，TOPOLOGY がチップ上のルータ間トポロジー，d がチップの積層枚数，x がコア当りのランダムリンクの本数，y がランダムコアリンクの長さの上限（コア長）を表す．

本解析では，積層する各チップのルータ間トポロジーは全て同じとした．ルータ間トポロジーには，**3.3** と同様に 2D-MESH, 2D-TORUS, H-TREE, HYPER-CUBE の四つを採用した．また，チップ間の接続は，図 9 のようにポイント・ツー・ポイントの接続方式を用いた．すなわち，上下に隣り合ったチップ間で同じ x, y 座標に位置するルータ同士を接続した．例えば，チップ内のルータ間トポロジーが MESH の場合，チップ間のルータが直線状につながり，最終的に三次元 MESH を形成する．なお，図 9 では，図の複雑化を防ぐため，3x3 MESH 上の手前 3 コアについてのみチップ間リンクを表示し，それ以外のコアについては省略している．

ルータ及び配線の遅延は **3.3** と同様とした．ただし，例外として，チップ間のルータ間配線における遅延は 1[cycle] とした．

図 10, 11 は 16 コアを 4 枚積層した NoC トポロジーにおける 1 コア当りのランダムリンク数と最大・平均 Zero-load 遅延の関係を示している．接続半径は $y = 2$ とした．これらの図より，二次元 NoC にランダムコアリンクを適用した場合と同様に，三次元 NoC

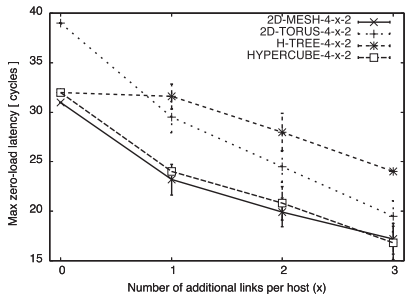

図 10　16x4 コア，$y = 2$ における各トポロジーのランダムコアリンク数と最大 Zero-load 遅延
Fig. 10　Max zero-load latency versus the number of random-core links for each topology (16x4 cores, $y = 2$).

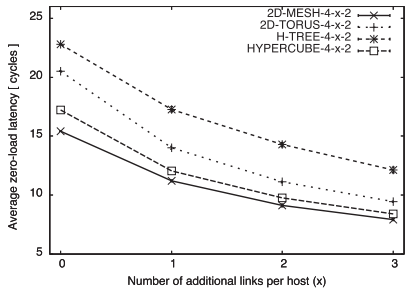

図 11　16x4 コア，$y = 2$ における各トポロジーのランダムコアリンク数と平均 Zero-load 遅延
Fig. 11　Average zero-load latency versus the number of random-core links for each topology (16x4 cores, $y = 2$).

におけるランダムコアリンクによる遅延削減の効果が大きいことが分かる．例えば，ルータ間トポロジーが MESH の場合，1 本のランダムコアリンク追加による平均遅延削減率は，16x4 コアの場合 27% となっている．このように大幅に通信遅延を削減できるのは，チップ間方向のリンク長が 0 であり，ランダムコアリンクによりチップ間方向へのホップ数が大幅に削減できたためであると考えられる．

5. ネットワークシミュレーション

本章では，ランダムコアリンクのネットワーク性能への影響を調べるため，通信遅延を評価した．

5.1 シミュレーション環境

複数ランダムコアリンクを用いたトポロジーの性能評価のために，マルチコアプロセッサのフルシステムシミュレータである GEM5 [18] を使用した．GEM5 は，C++ 及び Python で実装されており，プロセッサの詳細なシミュレーションに加え，NoC を対象にしたネットワークシミュレーション機能も有している．

表 1　ネットワークパラメータ
Table 1　Network parameter.

Router latency	3 cycles
Link latency	1 cycle
Switching	Wormhole
Flit size	128 bit
Number of VCs	3
Buffer size per VC	4-flit
Topology of routers	MESH
Routing	Dimension Order Routing

パケットサイズは 1-flit/5-flit とした．その他のネットワークパラメータを表 1 に示す．発生するトラヒックに応じて，各コアは独立してネットワークにフリットを注入するものとした．**3.2** に示したとおり，ランダムコアリンクを用いたトポロジーにおいて経路として用いるコアリンクを選択する際は，最小ホップ数となるようなリンクを選ぶこととした．

5.2 ネットワーク性能の評価

本章では，ネットワークシミュレーションによるランダムコアリンクを用いたトポロジーの評価を行う．

ルータ間トポロジーは，二次元 NoC トポロジーである 4x4 MESH（16 コア），8x8 MESH（64 コア）と，三次元 NoC トポロジーである 4x4x4 MESH（64 コア）の三つを採用し，追加リンク数 $x = 0, 1, 3$ の三つについて評価を行った．また，接続半径は $y = 2$(4x4 MESH), 4(8x8 MESH), 2(4x4x4 MESH) とした．

図 12〜17 は，ランダムに宛先を選択する Uniform トラヒックと，合成トラヒックである Bit complement トラヒックを，先述のネットワークに注入した場合のシミュレーション結果である．縦軸はフリットが生成されてから宛先コアに到達するまでのコア間の平均遅延を，横軸は各コアの送信フリットレートである Injection rate を示す．

これらの図から，コアリンク数の増加に伴い，低負荷時の遅延が削減できることが分かる．例えば，ルータ間 8x8 MESH，$y = 4$ の場合，ランダムコアリンクを 3 本付加することにより，Uniform Traffic では 45%，Bit complement Traffic では 38%の遅延削減効果が得られた．また，4x4x4 MESH, $y = 2$ の場合，ランダムコアリンク 3 本の付加により，Uniform Traffic では 40%，Bit complement Traffic では 43%の遅延削減効果が得られた．

しかし，図 13，図 15，図 17 に示すように，特に Bit complement Traffic において，ランダムリンクを

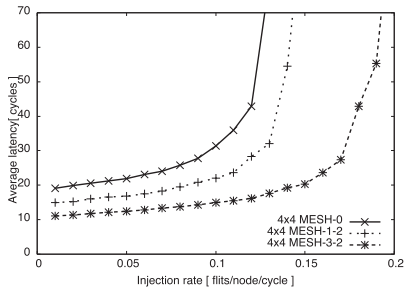

図 12　16 コアにおけるネットワーク性能（16 コア，Uniform トラヒック）

Fig. 12　Average latency vs. accepted traffic (16 cores, Uniform traffic).

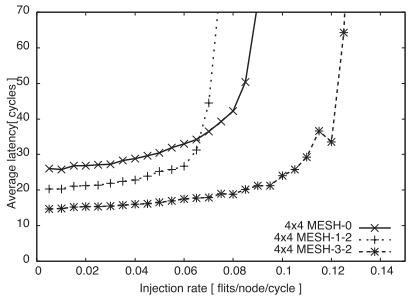

図 13　16 コアにおけるネットワーク性能（16 コア，Bit complement トラヒック）

Fig. 13　Average latency vs. accepted traffic (16 cores, Bcomp traffic).

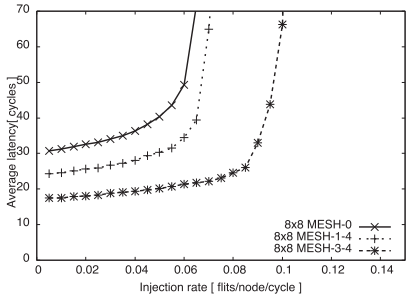

図 14　64 コアにおけるネットワーク性能（64 コア，Uniform トラヒック）

Fig. 14　Average latency vs. accepted traffic (64 cores, Uniform traffic).

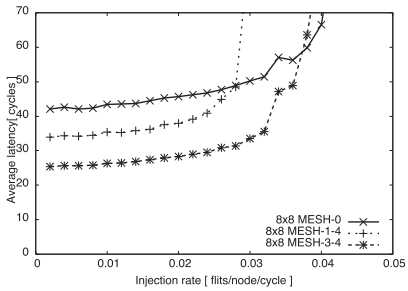

図 15　64 コアにおけるネットワーク性能（64 コア，Bit complement トラヒック）

Fig. 15　Average latency vs. accepted traffic (64 cores, Bcomp traffic).

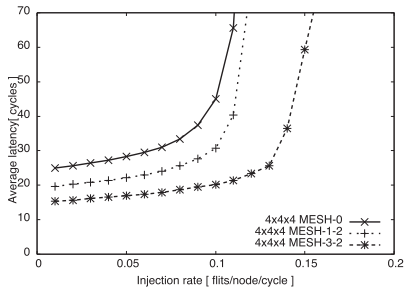

図 16　16x4 コアにおけるネットワーク性能（16x4 コア，Uniform トラヒック）

Fig. 16　Average latency vs. accepted traffic (64 cores, Uniform traffic).

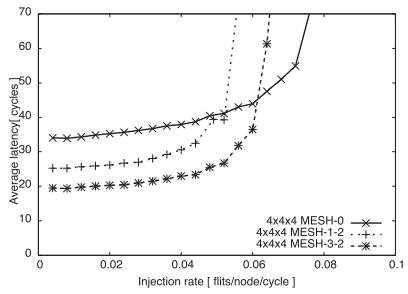

図 17　16x4 コアにおけるネットワーク性能（16x4 コア，Bit complement トラヒック）

Fig. 17　Average latency vs. accepted traffic (64 cores, Bcomp traffic).

付加することによりスループットが減少している．これは，今回の評価におけるルーティングが一つの最短経路に転送経路を限定するようなルーティングであるため，リンク付加によりルータ間トポロジーにおけるロードバランシングが悪化していることが原因と考えられる．このような問題を解決する手法として，複数の最短経路及び非最短経路を許すようなルーティングを実装し，ロードバランシングを図ることが考えられる．このようなルーティング手法の実装及び評価は今後の課題である．

6. フルシステムシミュレーション

本章では，ランダムコアリンクトポロジーに対する並列アプリケーション評価を行う．

6.1 シミュレーション環境

評価のため，フルシステムシミュレータとして

C	C	C	C
L	L	L	L
L	L	L	L
D	D	D	D

(a) 4x4 MESH

C	C	C	C	C	C	C	C
L	L	L	L	L	L	L	L
L	L	L	L	L	L	L	L
L	L	L	L	L	L	L	L
L	L	L	L	L	L	L	L
L	L	L	L	L	L	L	L
L	L	L	L	L	L	L	L
D	D	D	D	D	D	D	D

(b) 8x8 MESH

C	L	L	C
L	L	L	L
L	L	L	L
D	L	L	D

(c) 4x4x4 MESH (1 chip)

図 18　各トポロジーにおけるタイル配置
Fig. 18　Tile placement on each topology.

表 2　フルシステムシミュレーションのパラメータ
Table 2　Parameters of full-system simulation.

Processor	X86_64
L1 I/D cache size	32 KB (line:64B)
L1 cache latency	1 cycle
L2 cache bank size	256 KB (assoc:8)
L2 cache latency	6 cycles
Memory size	2 GB
Memory latency	160 cycles

GEM5 [18] を使用した．評価に用いたプロセッサ，キャッシュ，及びメモリの構成は表 2 に示す．プロトコルは MOESI directory とし，制御/データパケットのサイズはそれぞれ 1-flit/5-flit とした．その他のネットワークパラメータは表 1 と同様とし，ランダムコアリンクの経路選択手法は **5. 1** と同様とした．

6. 2　アプリケーション評価

本章のアプリケーション評価の対象となるルータ間トポロジーとして，4x4 MESH（16 コア），8x8 MESH（64 コア），4x4x4 MESH（64 コア）の三つを採用した．各トポロジーにおけるアーキテクチャのタイル配置は図 18 に示す．この図において，“C” は CPU，“L” は L2 キャッシュ，“D” はディレクトリコントローラを表す．CPU に示すタイルには L1 命令/ データキャッシュが内包されている．各タイルは 1 ルータをローカルにもち，ルータ間で MESH を構成している．また，4x4x4 MESH については，図 18(c) のようにタイル配置を行ったチップを 4 枚積層することでトポロジーを構成し，チップ間はポイント・ツー・ポイントで接続した．

ベンチマークとして NAS 並列ベンチマーク [19] の七つのプログラムを使用した．スレッド数は各トポロジーがもつ CPU の数に合わせた．

図 19〜21 にアプリケーション実行時間の評価結果を示す．この図では，ランダムコアリンクを追加しない場合 ($x = 0$) の実行時間を基準としてランダムコアリンクを用いたトポロジーの実行時間を正規化している．また，右端の “Ave.” の凡例に，七つのアプリケーション実行時間の平均を示している．

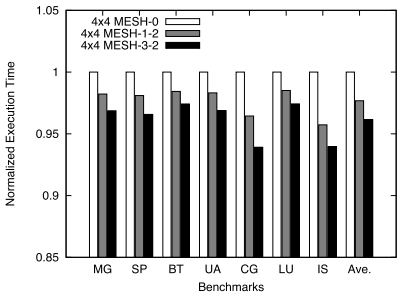

図 19　4x4 MESH（16 コア）におけるアプリケーション実行時間
Fig. 19　Execution time of applications on 4x4 MESH (16 cores).

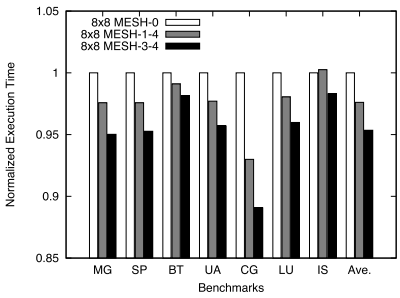

図 20　8x8 MESH（64 コア）におけるアプリケーション実行時間
Fig. 20　Execution time of applications on 8x8 MESH (64 cores).

これらの図より，ランダムコアリンクによりほとんどのアプリケーションにおいて実行時間が短縮できていることが分かる．ランダムコアリンクを 3 本追加することによる実行時間の平均削減率は，4x4 MESH で 3.9%，8x8 MESH で 4.6%，4x4x4 MESH で 3.6%となっている．しかし，実行時間の削減率はアプリケーションによって異なる．例えば，トポロジーを 8x8

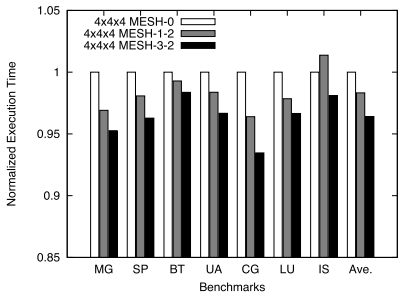

図 21　4x4x4 MESH（64 コア）におけるアプリケーション実行時間
Fig. 21　Execution time of applications on 4x4x4 MESH (64 cores).

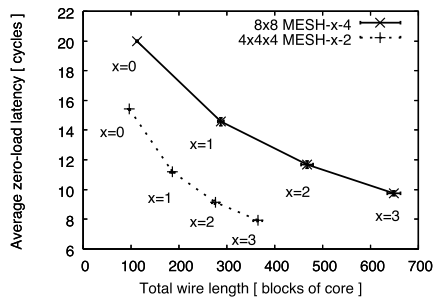

図 22　8x8 MESH と 4x4x4 MESH（64 コア）における総配線長と平均 Zero-load 遅延
Fig. 22　Average zero-load latency versus total wire length on 8x8 MESH and 4x4x4 MESH (64 cores).

表 3　8x8 TORUS と 8x8 MESH-1-2（64 コア）における総配線長と平均 Zero-load 遅延
Table 3　Total wire length and average zero-load latency on 8x8 TORUS and 8x8 MESH-1-2 (64 cores).

	8x8 TORUS	8x8 MESH-1-2
Total wire length	224 blocks	214 blocks
Average zero-load latency	18.5 cycles	16.4 cycles

MESH としてランダムコアリンクを 3 本追加することにより，CG の実行時は実行時間を 10.9%改善できているが，IS の実行時はランダムコアリンクによる改善率が 1.9%となり，ほとんど実行時間に差がなかった．このような結果は，実行アプリケーションによって通信の頻度や偏りに違いがあることが原因であると考えられる．

7. 議　　論

7.1 長いコアリンクによる動作周波数への影響

文献 [20] によると，一般的な 65nm プロセスにおけるメタル配線（中間層）に対し，配線遅延が最小になるようにリピータバッファを挿入した場合，1mm 当りの配線遅延は 41[ps] であると報告されている．NoC におけるコア長（ルータ間距離）を 3mm とした場合 [1]，接続半径を 2, 4（単位はコア長）としたときの最大配線遅延は，それぞれおよそ 0.24, 0.49[ns] となる（コア長が 3mm よりも短い場合，最大配線遅延はこれよりも更に小さくなる）．これらの配線遅延は，ルータの入力及び出力側の両方でバッファリングを行い，1GHz 以下の比較的低い動作周波数で動作するオンチップルータにおいてはほとんど問題にならないと考えられる．

よって，**5.** などで想定した接続半径 2, 4（コア長）のランダムコアリンクは現実的に可能である．

7.2 二次元と三次元ランダム・コアリンクトポロジーの比較

本章ではランダムコアリンクを用いた二次元トポロジーと三次元トポロジーの性能比較を行う．

図 22 に 64 コアにおける二次元 NoC（8x8 コア）と三次元 NoC（4x4x4 コア）の総配線長対平均 Zero-load 遅延の性能比較を示す．ここで，ルータ間トポロジーは MESH とし，追加ランダムコアリンク数 (x) を 0〜3 まで変化させている．接続半径は $y = 4$(8x8 MESH)，2(4x4x4 MESH) とした．Zero-load 遅延の解析手法は **3.3** などと同様とした．

この図より，三次元 NoC にランダムコアリンクを適用する方が，総配線長の増加に対する遅延の削減の度合が高いことが分かる．三次元 NoC はチップ間方向の接続を生かして総配線長やホップ数を減らすことができるため，元々二次元 NoC よりも総配線長に対する遅延性能が良いが，更にランダムコアリンクを用いると，チップ間に長距離リンクを張れるため，その優位性が更に大きくなったと考えられる．

7.3 規則トポロジーとランダムコアリンクを用いたトポロジーとの総配線長及び遅延性能の比較

本章では従来の規則的なトポロジーとランダムコアリンクを用いたトポロジーとの性能比較を行う．

表 3 に 64 コアにおける，規則的なトポロジーとランダムコアリンクトポロジーとの総配線長及び Zero-load 遅延の比較を示す．ここでは，規則的なトポロジーに TORUS を採用し，ランダムコアリンクを適用するトポロジーに MESH を採用している．また，ランダムコアリンクの接続半径を 2 とし，コア当りの追加本数

を 1 本としている．Zero-load 遅延の解析手法は **3.3** などと同様とし，総配線長の測定手法は **7.2** と同様とした．

この表より，ランダムコアリンクを用いることで，コア間の平均ホップ数を 11.5%削減しつつ，総配線長を 4.5%減少させるという結果が得られた．このことから，ランダムコアリンクは，規則的な追加リンクに比べて，総配線長に対する性能が優れていることが示せた．

7.4 規則トポロジーとランダムコアリンクを用いたトポロジーとの配線密度の偏りの比較

ランダムコアリンクを用いたトポロジーは，チップ上で配線密度に偏りを生じさせるおそれがある．そこで，本章では，従来の規則的なトポロジーとランダムコアリンクを用いたトポロジーとの配線密度の偏りの比較を行う．ここでの配線密度とは，一つのコアを通過する x 方向若しくは y 方向のリンク数を示す．そして本章では，このリンク数が各コア間でどの程度偏るかを解析する．

表 4 に 64 コアにおける，規則的なトポロジーとランダムコアリンクトポロジーとの配線密度の比較を示す．ここでは，規則的なトポロジーに TORUS を採用し，ランダムコアリンクを適用するトポロジーに MESH を採用している．また，ランダムコアリンクの接続半径を 4 とし，コア当りの追加本数を 1 本としている．また，評価結果の値は，x 方向，y 方向のそれぞれで最大，平均，標準偏差，相対標準偏差を集計した後，2 方向の平均値をとっている．

ランダムコアリンクのレイアウトにおいて，コアと接続先ルータとの位置関係が斜めである場合は，1 回のみ配線をターンさせることとして計算した．また，ターンしている場所のコア上には，x 方向と y 方向に 1 本ずつリンクがあるものとして本数を数えた．更に，各コア上の通過リンク数の標準偏差が少なくなるよう局所的に最適化したレイアウトを評価対象として用いた．なお，ランダムコアリンクを用いた場合の評価結果の値は，乱数を変えて生成した 10 個のトポロジーの平均値をとっている．

表 4　8x8 TORUS と 8x8 MESH-1-4（64 コア）におけるコア上の配線密度
Table 4　Wire density on each core on 8x8 TORUS and 8x8 MESH-1-4 (64 cores).

	8x8 TORUS	8x8 MESH-1-4
Max	1 link	3.05 links
Average	0.75 links	1.13 links
Standard deviation	0.43 links	0.80 links
Relative standard deviation	0.58	0.71

この表より，ランダムコアリンクを用いた場合，コア上の配線密度の相対標準偏差が 22%の増加に留まることが分かった．

8. むすび

本研究では，end-to-end 通信遅延の削減を目指して，単一コアとランダムに選択した近隣のルータとの間にリンクを追加し，コアからのリンクを複数とする方法を提案した．得られた知見を以下にまとめる．

- グラフ解析結果から典型的なルータ間トポロジーである 2D-MESH, 2D-TORUS, H-TREE, HYPERCUBE に対してランダムコアリンクを付加した場合，コア間のホップ数や Zero-load 遅延が劇的に改善する．
- ランダムコアリンクの接続半径を抑えたとしても十分に Zero-load 遅延を削減することができる．64 コアの場合，4 コア長のリンクの制限を課した場合，リンク長に制限のない場合と比べて，Zero-load 遅延に関して 7 割の削減効果が得られた．
- 三次元 NoC にランダムコアリンクを追加した場合，二次元 NoC に追加する場合と比べて，より Zero-load 遅延の削減効率が大きい．
- フリットレベルシミュレーションにより，ランダムコアリンクを用いたトポロジーは，従来のトポロジーに比べ，低負荷時の平均遅延を最大 45%減少させた．
- そのネットワークを用いたフルシステム CMP シミュレーションの結果，NAS 並列ベンチマークの実行時間を最大 10.9%向上させることが分かった．

今後の課題としては，ランダムコアリンクを用いた本トポロジーの利用法と最適化が挙げられる．ルータ間トポロジーは HYPERCUBE や TORUS などの規則網である．そのため，これらの構造に最適化，マップされた並列アプリケーションを実行する場合はランダムコアリンクを shutdown し，規則網のリンクのみを用いるルーティング制御を行う．そして，トラヒックパターンが自明でない，あるいは遅延に繊細な CMP アプリケーションを実行する場合は全てのリンクを使う．こういったアプリケーション最適化及び NoC 電力最適化，更には本ランダムコアリンクの追加による NoC の耐故障性の向上の評価などを今後実施する予

定である．

謝辞　本研究の一部は国立情報学研究所公募型共同研究（一般研究公募型）の助成を受けたものである．

文　献

[1] W.J. Dally and B. Towles, “Route packets, not Wires: On-chip interconnection networks,” Proc. Design Automation Conference (DAC’01), pp.684–689, June 2001.

[2] 河野隆太，藤原一毅，松谷宏紀，天野英晴，鯉渕道紘，“ホストから複数リンクを用いた低遅延ネットワークトポロジ，” 信学技報，CPSY2012-77, Jan. 2013.

[3] D.J. Watts and S.H. Strogatz, “Collective dynamics of ’small-world’ networks,” Nature, vol.393, no.6684, pp.440–442, 1998.

[4] M. Koibuchi, H. Matsutani, H. Amano, D.F. Hsu, and H. Casanova, “A case for random shortcut topologies for HPC interconnects,” Proc. International Symposium on Computer Architecture (ISCA), pp.177–188, 2012.

[5] H. Matsutani, M. Koibuchi, H. Amano, and T. Yoshinaga, “Prediction router: Yet another low latency on-chip router architecture,” Proc. International Symposium on High-Performance Computer Architecture (HPCA’09), pp.367–378, Feb. 2009.

[6] D.Burger, S.W.Keckler, K. McKinley, M.Dahlin, L. John, C.Lin, C. Moore, J.Burrill, R. McDonald, W.Yoder, and the TRIPS Team, “Scaling to the end of silicon with EDGE architectures,” Computer, vol.37, no.7, pp.44–55, July 2004.

[7] P. Gratz, C. Kim, K. Sankaralingam, H. Hanson, P. Shivakumar, S.W. Keckler, and D. Burger, “On-chip Interconnection networks of the TRIPS chip,” IEEE Micro, vol.27, no.5, pp.41–50, 2007.

[8] Ü.Y. Ogras and R. Marculescu, ““it’s a small world after all”: Noc performance optimization via long-range link insertion,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.14, no.7, pp.693–706, July 2006.

[9] 飯尾亮介，平木　敬，“三次元トポロジ NoC の比較評価，” 先進的計算基盤システムシンポジウム論文集，pp.18–19, May 2012.

[10] Y. Nishioka, M. Iida, and T. Sueyoshi, “Small-world network to reduce delay in fpga routing structures,” Int. J. Innovative Computing, Information and Control (IJICIC), vol.6, no.2, pp.551–566, Feb. 2010.

[11] S. Deb, A. Ganguly, P.P. Pande, B. Belzer, and D. Heo, “Wireless NoC as interconnection backbone for multicore chips: Promises and challenges,” IEEE J. Emerging and Selected Topics in Circuits and Systems, vol.2, no.2, pp.228–239, June 2012.

[12] J. Kim, J. Balfour, and W.J. Dally, “Flattened butterfly topology for on-chip networks,” Proc. International Symposium on Microarchitecture (MICRO’07), pp.172–182, Dec. 2007.

[13] B. Black, D.W. Nelson, C. Webb, and N. Samra, “3D processing technology and its impact on iA32 microprocessors,” Proc. International Conference on Computer Design (ICCD’04), pp.316–318, Oct. 2004.

[14] W.R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer, and P.D. Franzon, “Demystifying 3D ICs: The pros and cons of going vertical,” IEEE Desi. Test Comput., vol.22, no.6, pp.498–510, Nov. 2005.

[15] V.F. Pavlidis and E.G. Friedman, “3-D Topologies for Networks-on-Chip,” Proc. International System-on-Chip Conference, pp.285–288, Sept. 2006.

[16] H. Matsutani, M. Koibuchi, and H. Amano, “Tightly-coupled multi-layer topologies for 3-D NoCs,” Proc. International Conference on Parallel Processing (ICPP’07), Sept. 2007.

[17] H. Matsutani, M. Koibuchi, Y. Yamada, D.F. Hsu, and H. Amano, “Fat H-Tree: A cost-efficient tree-based on-chip network,” IEEE Trans. Parallel Distrib. Syst., vol.20, no.8, pp.1126–1141, Aug. 2009.

[18] N. Binkert, B. Beckmann, G. Black, S.K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D.R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M.D. Hill, and D.A. Wood, “The gem5 Simulator,” ACM SIGARCH Computer Architecture News, vol.39, no.2, pp.1–7, May 2011.

[19] H. Jin, M. Frumkin, and J. Yan, “The OpenMP implementation of NAS parallel benchmarks and its performance,” NAS Technical Report, NAS-99-011, Oct. 1999.

[20] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed., Addison-Wesley, Boston, 2010.

（平成 25 年 6 月 19 日受付，10 月 20 日再受付）

河野　隆太　（学生員）

2013 慶應義塾大学理工学部情報工学科卒業．現在，同大大学院理工学研究科開放環境科学専攻前期博士課程在籍．電子情報通信学会会員．

藤原　一毅　（正員）

2002 東京工業大学工学部制御システム工学科卒業．2004 同大大学院理工学研究科機械制御システム専攻修了．2004～2008 日立製作所情報制御システム事業部．2012 総合研究大学院大学複合科学研究科情報学専攻修了．博士（情報学）．現在，国立情報学研究所特任研究員．IEEE，情報処理学会，電子情報通信学会各会員．

松谷　宏紀　（正員）

2004 慶應義塾大学環境情報学部卒業．2008 同大学大学院理工学研究科後期博士課程修了．博士（工学）．2009 年度より 2010 年度まで，東京大学大学院情報理工学系研究科特別研究員，日本学術振興会特別研究員（SPD）．現在，慶應義塾大学理工学部情報工学科専任講師．コンピュータアーキテクチャとインタコネクトに関する研究に従事．IEEE，情報処理学会，電子情報通信学会各会員．

天野　英晴　（正員）

1986 慶應義塾大学理工学研究科後期博士課程修了．工学博士．現在，慶應義塾大学理工学部情報工学科教授．並列計算機と再構成可能システムに関する研究に従事．IEEE，ACM，情報処理学会各会員，電子情報通信学会シニア会員．

鯉渕　道紘　（正員）

2000 慶應義塾大学理工学部情報工学科卒業．2003 同大大学院理工学研究科開放環境科学専攻後期博士課程修了．博士（工学）．2002 年度より 2004 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（併任）．ハイパフォーマンスコンピューティングとインタコネクトに関する研究に従事．2007 年度情報処理学会論文賞，2013 年度文部科学大臣表彰若手科学者賞等受賞．IEEE，情報処理学会各会員，電子情報通信学会シニア会員．