計算機アーキテクチャ　103－14
設　計　自　動　化　69－14
(1993. 12. 17)

VHDLによる相互結合網のモデル化と性能評価

奥村 勝†　　末吉 敏則†

† 九州工業大学 情報工学部 知能情報工学科
福岡県飯塚市川津680-4

okkun@mickey.ai.kyutech.ac.jp
sueyoshi@ai.kyutech.ac.jp

あらまし　ハードウェア記述言語VHDLを用いて相互結合網をモデル化することで，回路の動作に即したデータ転送を行うことができ，理論解析やプログラム言語によるソフトウェア・シミュレーションよりも意図する相互結合網に近いシミュレーション結果を得ることができる．また，並列アプリケーションから取得した通信パターンを基にシミュレーションが行えるため，実際の通信状況における相互結合網の詳細な性能評価が可能である．本稿では，VHDLによる相互結合網のモデル化について述べた後，オメガ網と2次元トーラス網においてフロー制御方式の違いによる通信性能への影響を評価した．

和文キーワード　相互結合網，フロー制御，性能評価，ハードウェア記述言語，オメガ網，トーラス網

Modeling and Performance Evaluation of Interconnection Networks using VHDL

Masaru Okumura †　Toshinori Sueyoshi †

† Department of Artificial Intelligence
Kyushu Institute of Technology
680-4 Kawazu, Iizuka-shi, Fukuoka, 820 Japan

Abstract　In this paper, we describe the modeling and performance evaluation of interconnection networks using VHDL. The modeling of interconnection networks with VHDL provide the simulation equivalent to actual circuits. The omega and 2-dimensional torus networks were modeled using VHDL, and utilized these models to evaluate the influence of different flow controls on the communication performance. Furthermore, we used the communication patterns derived from actual parallel applications to evaluate the communication performance in detail.

英文 key words　interconnection network, flow control, performance evaluation, hardware discripton language, omega network, torus network

1 はじめに

並列計算機システムにおいて多数のプロセッサおよびメモリを相互に結合する相互結合網は，システムの性能を大きく左右する重要な構成要素である．相互結合網を特徴づける要素技術として，プロセッサ要素間のトポロジ，プロセッサ要素間のデータ転送経路を決定するルーティング方式および転送データに対するトラフィックの制御や通信資源の割当てを行うフロー制御方式の3つが挙げられる．並列計算機システムでは，これらの要素技術を適切に選択し，高速な通信を実現しなければ，高い並列処理効率を得ることができない．また，並列計算機システムの並列処理効率に影響を与える要素の一つに，上記の3つの要素に加え，実行される並列アプリケーションの通信パターンがある．実行されるプログラムに依存した空間的局所性および時間的局所性を含む通信パターンは，相互結合網における通信遅延時間に影響を与えるため，相互結合網の通信性能を把握する上で考慮する必要がある．

これらの要素から成る様々な相互結合網の性能を評価するために，我々はハードウェア記述言語による相互結合網のモデル化を行い，EDA環境を利用したシミュレーションによる通信性能の評価を行っている[1]．相互結合網のモデル化にハードウェア記述言語を利用することで，回路動作に即したプロセッサ要素間のデータ転送をモデル化することが可能となり，理論解析やプログラム言語によるソフトウェア・シミュレーションよりも，意図する相互結合網に近いシミュレーション結果を得ることができる．さらに，並列アプリケーションの通信パターンを利用することで，実際の通信状況における相互結合網の詳細な性能評価が可能である．

本研究では，フロー制御方式に着目し，代表的なフロー制御方式であるstore and forward，wormhole，virtual cut-through[2]の3種類についてモデル化およびシミュレーションを行い，フロー制御方式が相互結合網の通信性能に与える影響について調査した．

本稿では，ハードウェア記述言語VHDLによる相互結合網のモデル化，このモデルを利用したシミュレーションの概要，およびシミュレーションから得られた相互結合網におけるフロー制御方式に対する性能評価について述べる．以下，第2章では相互結合網のモデル化の概要について述べ，第3章では，シミュレーションの概要について述べる．第4章では，規則的な通信パターンを用いたシミュレーションによる性能評価について述べる．さらに，第5章では実際の並列アプリケーションの通信パターンを用いたシミュレーションによる性能評価について述べる．

2 VHDLによる相互結合網のモデル化

相互結合網は大きく動的網（間接網）と静的網（直接網）の2種類に分類される．フロー制御方式の違いによる性能評価を行う場合，この両方についてモデル化およびシミュレーションを行うことが望ましい．フロー制御方式の違いによる通信性能への影響を評価する場合，プロセッサ要素間の転送経路が複数存在する相互結合網もあり，ルーティング方式も考慮する必要がある．

本研究では，モデル化する相互結合網として，動的網については代表的な多段結合網であるオメガ網を採用した．静的網については2次元トーラス網を採用した．オメガ網は，プロセッサ要素間に複数段の交換スイッチ群を配置し，それらの間をシャッフル結合によって結合したものである．このオメガ網は，送信側と受信側のプロセッサ要素が決定するとその間の経路が一意に決定するという特徴を持つ．このためフロー制御方式の違いのみによる通信性能への影響を見ることができる．また，2次元トーラス網では複数の転送経路が存在することに加え，wormhole等ではデッドロックが発生する可能性がある．そこで，2次元トーラス網においては転送データの順序保証とデッドロック回避を実現するためにe-cubeルーティングと仮想チャネル[3]を採用する．

2.1 ハードウェア記述言語VHDL

相互結合網のモデル化には，現在ハードウェア設計において普及してきているハードウェア記述言語VHDL(VHSIC Hardware Description Language)を用いた[4]．VHDLでは回路の動作をゲートレベルで記述を行う構造記述と回路の振舞いを記述する機能記述の2通りの方法によって記述することができる．さらに，これら2通りの記述方法を組み合わせて記述できるため，制御回路や演算回路のように構成の複雑な回路も容易にモデル化することができる．また，回路動作に伴う内部遅延や信号線の幅等を容易に設定・変更できるため，これらの値を詳細に設定することで意図する相互結合網の動作に即したシミュレーション結果を得ることができる．さらに，VHDLはIEEEにより仕様が標準化されており，この仕様に準拠したVHDLの処理系では動作の互換性が保たれる．従って，様々なCADシステム上のシミュレーション環境を利用して相互結合網の性能評価を行うことが可能である．

2.2 相互結合網のモデル化

今回相互結合網をモデル化するにあたり，通信方式としてはパケット交換方式，制御方式としては分散制御方式を仮定した．プロセッサ要素間のメッセージの転送は，図1に示すようにstore and forwardではパケット単位，wormhole，virtual cut-throughではフリット単位で転送を行う．

2.2.1 オメガ網のモデル化

モデル化では，まず，図2に示すように交換スイッチをモデル化しVHDLで記述を行った．プロセッサ要素間のパケットの転送は，パケットをチャネル幅毎（フリット）に分割して転送する．交換スイッチに入力されたフリットは，チャネル毎に用意されたバッファに保管される．制御回路はパケットの先頭フリットから転送先のアドレスを取り出し，適切なクロスバ・スイッチの状態を要求する．要求はアービタにより調停され，要求に応じてクロスバ・スイッチの状態が変更される．制御回路はクロスバ・スイッチの状態が自分の要求した状態と同じになると，次の交換スイッチに対してフリットを転送する．

store and forwardでは交換スイッチ内に，物理チャネル毎に1パケット分のバッファを用意し，バッファへのデータの入出力はパケット単位で行う．また，wormholeでは1フリット分のバッファを用意し，バッファへのデータの入出力はフリット単位で行う．さらに，virtual cut-throughではstore and forwardと同様に物理チャネル毎に1パケット分のバッファを持つが，バッファへのデータの入出力はフリット単位で行う．

図2に示す交換スイッチ，プロセッサ要素(PE)および入出力インタフェース等の構成要素についてもVHDLで記述し，図3に示すように接続した．

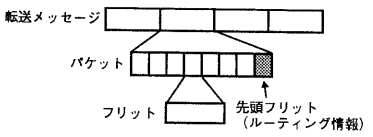


図1 プロセッサ要素間のデータ転送

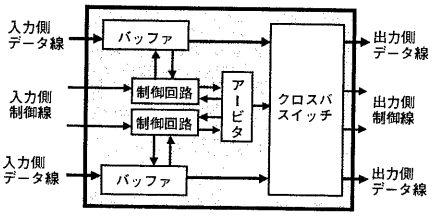


図2 2x2交換スイッチの構成

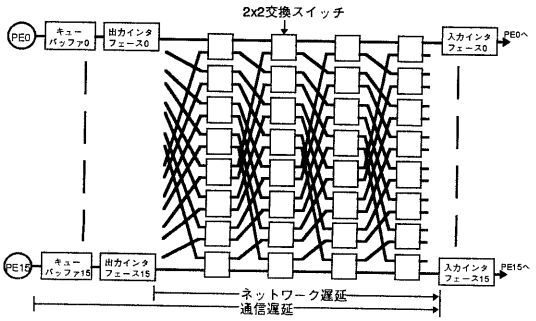


図3 オメガ網のモデル化（プロセッサ要素数:16）

2.2.2 2次元トーラス網のモデル化

2次元トーラス網ではデッドロックを回避するために，図4に示すように2本の仮想チャネルにより物理チャネルを仮想化している．オメガ網と同様に，まず図5に示すノードをVHDLで記述を行い，それらを図4のように接続している．ノード内では，X，Y軸方向の

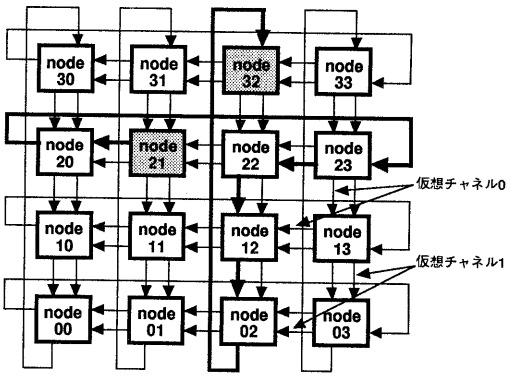


図4 トーラス網のモデル化（プロセッサ要素数:16）

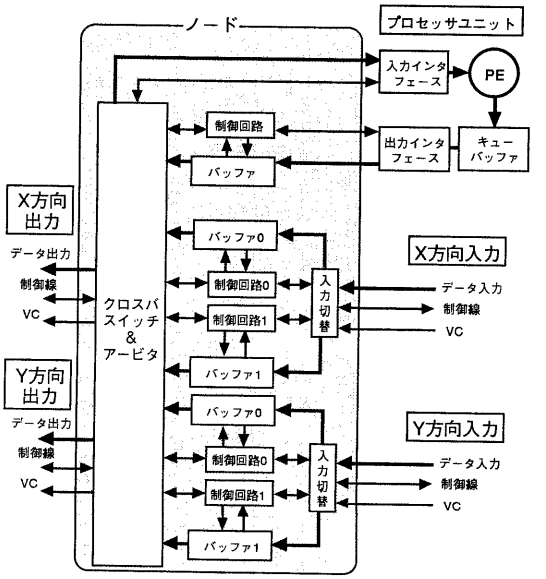


図5 ノードの構成

入力物理チャネルにつき2組のバッファと制御回路を用意し，プロセッサ要素からの入力に対してもバッファと制御回路を用意している．

プロセッサ要素間のパケットの転送は，オメガ網の場合と同様にパケットをフリットに分割して転送を行う．ノードに入力されたフリットは，仮想チャネル毎に用意されたバッファに保管される．制御回路はパケットの先頭フリットから転送先のアドレスを取り出し，適切なクロスバ・スイッチの状態を要求する．要求はアービタにより調停され，要求に応じてクロスバ・スイッチの状態が変更される．制御回路はクロスバ・スイッチの状態が自分の要求した状態になり，さらに転送可能な仮想チャネルを示す信号VCが有効であれば次の交換スイッチに対してフリットの転送を行う．この際，次のノードへの転送要求が1つの仮想チャネルのみであれば連続してフリットは転送されるが，両方の仮想チャネルが転送を要求する場合には，時分割で仮想チャネルを切り替えてフリットの転送を行う．

ノード内に用意するバッファの大きさは，オメガ網と同様にstore and forwardでは1パケット分，wormholeでは1フリット分，virtual cut-throughではstore and forwardと同様に1パケット分とする．

3 シミュレーションの概要

3.1 オメガ網，2次元トーラス網の仕様

プロセッサ要素数が16のオメガ網および2次元トーラス網についてシミュレーションを行う．これらの相互結合網の仕様を表1に示す．

3.2 通信パターン

シミュレーションでは，モデル化を行ったオメガ網および2次元トーラス網に対して様々な通信パターンのパケットを転送し，store and forward，wormholeおよびvirtual cut-throughの3つのフロー制御方式について，それぞれの転送遅延の測定結果をもとに性能評価を行う．通信パターンとしては，規則的な通信パターンと実際の並列アプリケーションから取得した通信パターンの2つを用意した．

表1 オメガ網および2次元トーラス網の仕様

プロセッサ要素数	16	
チャネルサイズ	16bit（1フリット）	
パケットサイズ	128bit（8フリット）固定長	
交換スイッチ内のバッファ数	store and forward virtual cut-through	1パケット（8フリット）
	wormhole	1フリット
データ転送周波数	20 MHz	
ルーティング処理時間	2クロック	

3.2.1 規則的な通信パターン

規則的な通信パターンでは，ある時間間隔でパケットを生成して宛先に転送する．あるパケットが生成されてから次のパケットが生成されるまでの間隔は，VHDLで乱数を生成させ決定している．パケットの宛先として，オメガ網には(A)送信するプロセッサ要素を宛先として交換スイッチ内でのパケットの衝突が起きないもの(contention free)，(B)ランダムな宛先(random)の2種類を用意し，2次元トーラス網には(B)ランダムな宛先(random)を用意した．転送するパケット数は，十分なトラフィックを生成するため各プロセッサ要素につき100パケットとした．各プロセッサ要素で生成されたパケットは図3および図5のキューバッファにキューイングされ，それぞれの宛先に転送される．

3.2.2 並列アプリケーションの通信パターン

実際の並列アプリケーションの通信パターンは，我々の研究室で構築した並列処理環境(DSE：Distributed Supercomputing Environment)を利用して取得した[5][6]．DSEは，通信ネットワークで接続された複数のワークステーションを利用して，分散共有メモリモデルに基づく並列処理環境を実現したものである．DSEは予め用意された接続情報ファイルをもとに各ワークステーションにプロセッサ要素を割り当て，通信することで意図する結合形態の相互結合網を仮想的に実現する．また，DSEは各プロセッサ要素における入出力メッセージの履歴情報を保存できるため，並列計算機上で実際の並列アプリケーションを実行した場合の通信パターンを取得することができる．

本研究では，Sun microsystems社のSPARC Station 2を16台を使用し，DSE上で並列アプリケーションを実行させて，通信パターンを取得した．

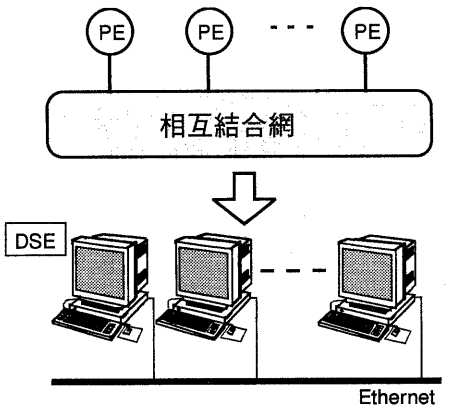
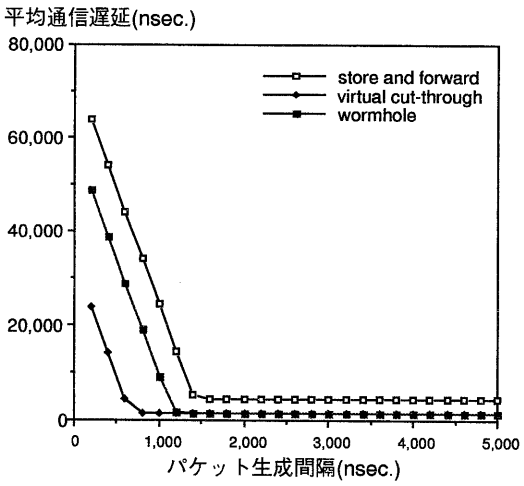


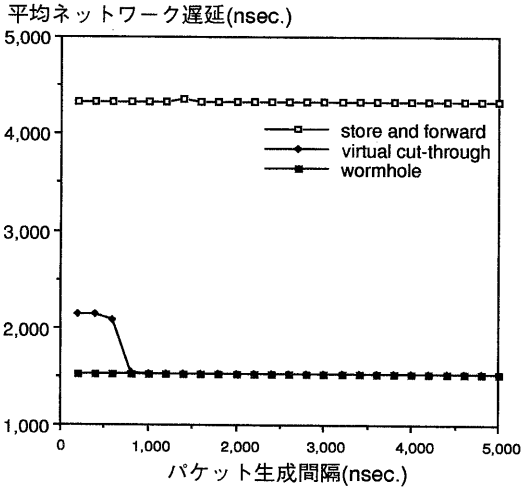
図6 分散スーパーコンピューティング環境（DSE）

3.3 評価指標

シミュレーションによる性能の評価指標としては図3に示すように，通信遅延（プロセッサ要素がパケットを生成してからそのパケットが宛先に到着するまでの時間），およびネットワーク遅延（相互結合網に対して出力インターフェースがパケットの送信を開始してからそのパケットが宛先に到着するまでの時間）の2種類を用いる．



(a)通信遅延



(b)ネットワーク遅延

図7　オメガ網におけるcontention freeの結果

4　規則的な通信パターンを用いた性能評価

4.1　オメガ網におけるcontention free

図7(a)にオメガ網におけるcontention freeのパケット生成間隔に対する平均通信遅延を示す．いずれのフロー制御方式の場合もパケットの生成間隔が大きい場合には通信遅延，ネットワーク遅延ともに一定値を示している．これはパケットの生成間隔がオメガ網のネットワーク遅延よりも十分大きいため，転送中のフリットが交換スイッチ内で待たされることなく次の交換スイッチに転送されるためである．この状態においてwormholeとvirtual cut-throughは同じ通信遅延である．これはどちらのフロー制御方式もクロスバ・スイッチにおける衝突がなく，パイプライン的にフリットを転送するためである．さらに，パケットの生成間隔を小さくして行き，生成間隔が約1,500 nsecよりも小さくなると通信遅延はいずれのフロー制御方式についても上昇を始める．これは，この上昇点付近でパケットの生成間隔がオメガ網のネットワーク遅延よりも小さくなり，生成されたパケットがキューバッファに待たされるためである．

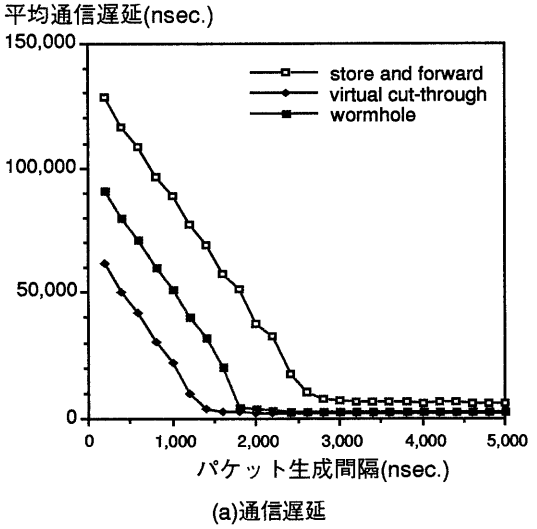
図7(b)にオメガ網におけるcontention freeのパケット生成間隔に対する平均ネットワーク遅延を示す．ネットワーク遅延についてみると生成間隔が大きい場合，いずれのフロー制御方式も一定値である．さらに生成間隔を小さくしていくと，virtual cut-throughのみネットワーク遅延が上昇を始め，再び一定値をとる．store and forwardとwormholeにおいて，出力インタフェースからオメガ網へのパケットあるいはフリットの注入は，1段目の交換スイッチ内のバッファが空くまで行われない．そのため生成間隔が小さくなりキューバッファにパケットが待たされている状況においても，オメガ網に注入されるパケット数あるいはフリット数は一定であるためネットワーク遅延も一定値となる．これに対し，virtual cut-throughでは交換スイッチ内のバッファにフリットをバッファリングするためネットワーク遅延が上昇し，バッファが一杯になると再び一定値を示すと推測できる．

4.2　オメガ網におけるrandom

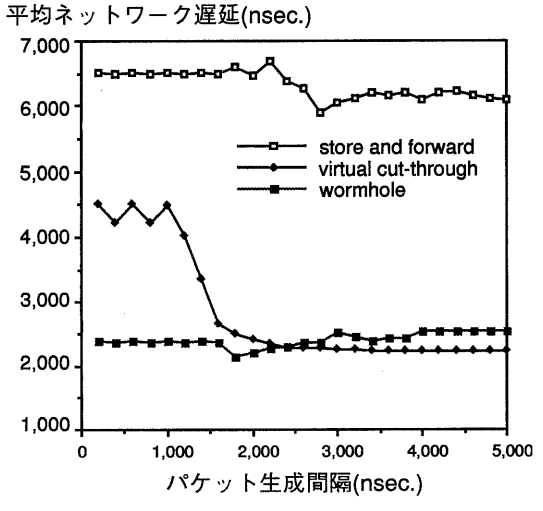
図8(a)にオメガ網におけるrandomのパケット生成間隔に対する平均通信遅延を示す．パケットの生成間隔を小さくして行き，生成間隔が3,000 nsec付近を越えるとcontention freeの場合と同様に通信遅延は上昇を始める．

図8(b)にオメガ網におけるrandomのパケット生成間隔に対する平均ネットワーク遅延を示す．ネットワーク遅延についてみるとパケットの生成間隔が大きいときは，contention freeの場合とは異なり完全な一定値にはなっていない．これは宛先がランダムであるため，

クロスバ・スイッチでパケットの衝突が生じ，ネットワークのトラフィックが一定にならないためである．生成間隔が2,500 nsec付近よりも大きい場合，wormholeのネットワーク遅延はvirtual cut-throughより若干大きくなっている．virtual cut-throughは交換スイッチ内に1パケット分のバッファを持つため，クロスバ・スイッチでパケットの衝突が生じた場合でもパケットをバッファリングすることができる．これに対しwormholeでは1フリット分のバッファしか持たないため，衝突が生じた場合に経路上のバッファを全て占有してしまい，他のパケットの転送を妨げる．このためwormholeのネットワーク遅延がvirtual cut-throughよりも大きくなっている．また，store and forwardのネットワーク遅延は，生成間隔を小さくして行くと経路上のクロスバ・スイッチでのパケットの衝突に加え，バッファが一杯になるため上昇する．さらに生成間隔が小さくなり，1,500 nsec付近になるとネットワークにパケットを注入できなくなるため，store and forwardのネットワーク遅延はほぼ一定値となる．一方，virtual cut-throughでは生成間隔が1,500 nsec付近よりも小さくなると，ネットワーク遅延はcontention freeの場合と同様に急激に上昇を始める．ここでcontention freeの場合よりも上昇時間が大きくなるのは，パケットの宛先がランダムであり，クロスバ・スイッチでの衝突による待ち時間が加わるためである．



(a)通信遅延



(b)ネットワーク遅延

図8　オメガ網におけるrandomの結果

いずれのフロー制御方式についても，通信遅延およびネットワーク遅延はcontention freeの結果に比べその差が大きくなっている．これは経路上のバッファが一杯になることに加え，クロスバ・スイッチでのパケットの衝突によりフリットがバッファで待たされていることが原因である．

4.3　2次元トーラス網におけるrandom

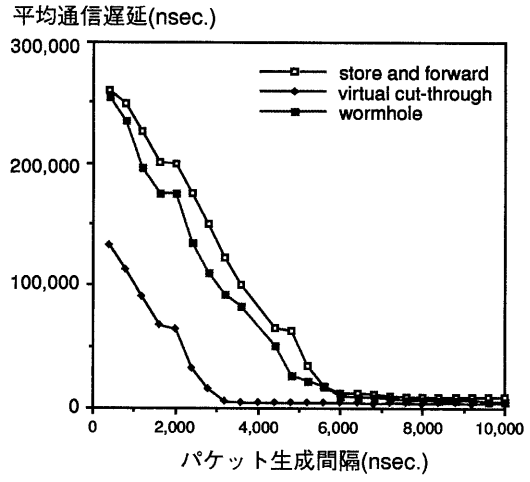
図9(a)にトーラス網におけるrandomのパケット生成間隔に対する平均通信遅延を示す．パケットの生成間隔を小さくして行き，生成間隔が6,000 nsec付近を越えると通信遅延は上昇を始める．通信遅延が上昇を開始した後の，store and forwardとwormhole間の通信遅延の差はオメガ網におけるrandomの場合に比べて縮まっている．これは，ノード内でパケットの衝突が起こった際に経路上のバッファを占有するwormholeは，トーラス網では経路上で他のパケットをブロッキングする確率が高い．従って，パケットの衝突時の待ち時間によりwormholeの通信遅延が増加するためと考えられる．

図9(b)にトーラス網におけるrandomのパケット生成間隔に対する平均ネットワーク通信遅延を示す．ネットワーク遅延についてみると，生成間隔が4,000 nsec付近よりも大きい場合はwormholeのネットワーク遅延はvirtual cut-throughよりも大きくなっている．これはオメガ網におけるrandomの場合と同様の結果であるが，前述の理由によりその差が拡大している．パケットの生成間隔を小さくしていくと，store and forwardとvirtual cut-throughについては，ネットワーク遅延は上昇し，オメガ網のrandomの場合と同様の結果を示す．これに対し，store and forwardはオメガ網の場合ほど一定したネットワーク遅延にはならない．これはトーラス網の場合，宛先までのパケットの転送距離が一定でないためといえる．

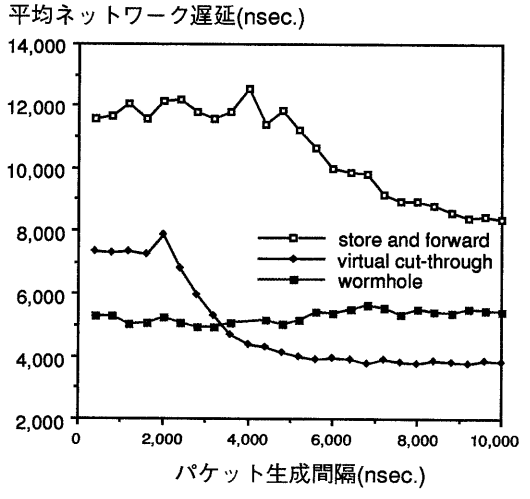
また，トーラス網における通信遅延およびネットワーク遅延はオメガ網に比べ増大している．これはオメガ網ではパケットの転送に要する距離が均一であるのに

対しトーラス網では不均一であり，さらにe-cubeルーティングと仮想チャネルの利用のため，最適なルーティングが行われていないことによる影響と考えられる．

これらの規則的な通信パターンによるシミュレーション結果からパケットの生成間隔が大きい場合，virtual cut-throughが他の2つのフロー制御方式に比べ良い性能を示す．一方，生成間隔が小さくトラフィックが増加する場合ネットワーク遅延が悪化することが分かる．また，wormholeについてはパケットの生成間隔に関わらずほぼ安定した性能であるが，ルーティング方式の差異により性能の低下が起こる．さらに，規則的な通信パターンとしてパケットが特定の宛先に集中する場合およびプロセッサ要素数の増加による影響も調査する必要がある．



(a)通信遅延



(b)ネットワーク遅延

図9　トーラス網におけるrandomの結果

5　並列アプリケーションを用いた性能評価

本研究では，並列アプリケーションによる通信パターンとして，前述のDSEを利用することで16×16の行列乗算とバイトニック・ソートの2種類の通信パターンを用意した．それぞれの通信パターンにおけるプロセッサ要素0のアクセスパターンを図10および図11に示す．横軸はDSEにおける実行時間であり，縦軸はアクセスの対象となるプロセッサ要素番号を示している．

(a)16×16行列乗算

プロセッサ要素数よりも大きな行列の演算を行う場合，行列要素の割当て方法等による影響も考慮しなければならない．この割当て問題を自明のものとするために行列サイズをプロセッサ数と同数の16とし，16×16の行列乗算を1回行った．各プロセッサ要素の共有メモリには対応する被乗数の行と乗数の列を割り当てる．このため，全対全通信を行う通信パターンになる．シミュレーションの実行に要したパケット数は20,640個であった．

(b)バイトニック・ソート

64個の整数型データを各プロセッサ要素に4個割り当て，バイトニック・ソートを行った．図11からも分かるようにプロセッサ要素によりアクセスの対象となるプロセッサ要素が限定される通信パターンである．シミュレーションの実行に要したパケット数は2,710個であった．

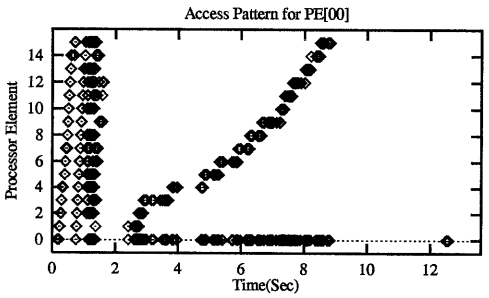


図10　行列乗算のアクセスパターン

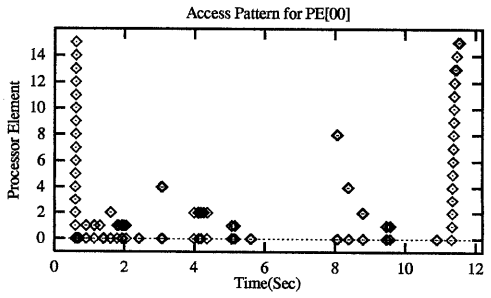


図11　バイトニック・ソートのアクセスパターン

DSEから取得した履歴情報に基づき直接シミュレーションを行った場合，プロセッサ要素における処理時間がパケットの転送遅延に比べ非常に大きいため，相互結合網におけるトラフィックが非常に小さくなる．そこで，意図的にトラフィックを増加させるために，プロセッサ要素の処理能力を1,000倍に想定し，シミュレーションを行った．

これら2種類の通信パターンによるシミュレーション結果を表2および表3に示す．表中の通信遅延およびネットワーク遅延は，通信遅延およびネットワーク遅延の算術平均値である．

表2　オメガ網におけるシミュレーション結果

フロー制御方式	プログラム	実行時間(nsec)	通信遅延(nsec)	ネットワーク遅延(nsec)
store and forward	行列乗算	23,910,737	103,426	4,576
wormhole	行列乗算	17,783,687	76,675	1,646
virtual cut-through	行列乗算	14,649,912	27,605	2,007
store and forward	ソート	10,895,137	8,415	4,325
wormhole	ソート	10,577,087	4,953	1,525
virtual cut-through	ソート	10,451,587	3,520	1,704

表3　トーラス網におけるシミュレーション結果

フロー制御方式	プログラム	実行時間(nsec)	通信遅延(nsec)	ネットワーク遅延(nsec)
store and forward	行列乗算	26,393,262	119,617	5,269
wormhole	行列乗算	21,409,713	101,486	2,252
virtual cut-through	行列乗算	17,764,963	62,722	2,925
store and forward	ソート	10,967,788	8,930	4,132
wormhole	ソート	10,683,328	6,143	1,881
virtual cut-through	ソート	10,579,088	4,899	2,405

オメガ網および2次元トーラス網ともに行列乗算とバイトニック・ソートではいずれのフロー制御方式でも通信遅延に大きな差が生じている．行列乗算は全対全通信であるのに加え，転送されるパケット数がバイトニック・ソートよりも非常に多い．このため，トラフィックが大きくなりキューバッファにおけるパケットの待ち時間の増加が通信遅延に影響を与えている．

また，オメガ網におけるネットワーク遅延についてみると，オメガ網におけるcontention freeのネットワーク遅延（図7(b)）に近い結果となっている．このことから，交換スイッチ内でのパケットの衝突があまり発生していない通信パターンであると考えられる．

ここでトーラス網におけるバイトニック・ソートのネットワーク遅延についてみると，store and forwardについてはオメガ網よりも小さくなっているが，wormholeとvirtual cut-throughについてはトーラス網のネットワーク遅延が大きくなっている．このようにあまり衝突が生じていない通信パターンの場合，store and forwardについてはオメガ網との大きな差は生じず，逆にwormholeとvirtual cut-throughではネットワーク遅延が増加しているといえる．これはトーラス網においてwormholeとvirtual cut-throughがe-cubeルーティングと仮想チャネルの影響を受けやすいためと考えられる．トーラス網ついては今後，ルーティング方式の違いによる影響などを調査する必要がある．

6　おわりに

本稿では，ハードウェア記述言語VHDLによる相互結合網のモデル化とこのモデルを用いたシミュレーション結果から通信性能を評価した．2種類の相互結合網についてモデル化を行い，規則的な通信パターンおよび実際の並列アプリケーションの通信パターンを用いたシミュレーションによりフロー制御方式の違いによる通信性能への影響を評価した．この結果，virtual cut-throughがパケットの生成間隔が小さい際にはネットワーク遅延が増加するものの，store and forwardとwormholeに比べ全般的に低い通信遅延，ネットワーク遅延を示すことが分かった．現在，取得情報の拡張，シミュレーションの高速化を行い，様々な相互結合網における性能評価を進めている．

謝辞

相互結合網のモデル化，シミュレーションに関して日頃から御討論いただく本学マイクロ化総合技術センターの久我守弘講師，本学情報工学研究科の柴村英智氏に感謝の意を表す．また，シミュレーションに必要な通信パターンをDSEを利用して取得していただいた本学知能情報工学科の岡雅樹氏，ならびに情報処理機構講座の諸氏に感謝します．

参考文献

[1] 船越, 久我, 末吉 : 相互結合網におけるフロー制御方式の通信パターンに基づく性能評価, 情報処理学会研究報告, 93-ARC-99, pp141-148, March 1993.

[2] William J. Dally : Network and Processor Architecture for Message-Driven Computers, *VLSI and PARALLEL COMPUTATION*, Edited by Robert Suaya and Graham Birtwistle, pp.140-222(Chapter 3), 1990.

[3] William J. Dally : Virtual-Channel Flow Control, *IEEE Transaction on Parallel and Distributed Systems*, vol.3, no.2, pp.192-205, March 1990.

[4] IEEE : IEEE Standard VHDL Language Reference Manual, 1988.

[5] B.O. Apduhan, T. Sueyoshi, Y. Namiuchi, T. Tezuka and I. Arita : Experiments of A Reconfigurable Multiprocessor Simulation on A Distributed Environment, *Proceedings of 1992 IEEE International Phoenix Conference on Computers and Communication*, pp. 539-546, April 1992.

[6] T. Tezuka, K. Ryokai, B.O. Apduhan and T. Sueyoshi : Implementation and Evaluation of a Distributed Supercomputing Environment on a Cluster of Workstations, *Proceedings of 1992 International Conference on Parallel And Distributed Systems*, pp. 58-65, December 1992.