

マルチコアシステムにおけるルータの実装と評価

姜 軒[†] 高前田 伸也[†] 渡邊 伸平[†] 三好 健文^{†‡} 吉瀬 謙二[†]
 東京工業大学大学院 情報理工学研究科[†] 科学技術振興機構 CREST[‡]

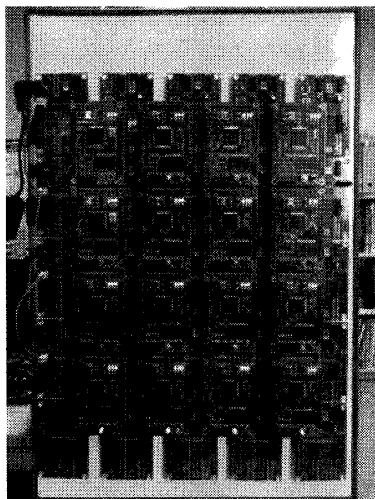


図 1: 4 × 4 ScalableCore

1 はじめに

近年プロセッサに複数のコアが搭載されるようになってきている。今後、コア数がさらに増加していく中で、コア間の通信に利用するネットワークは、プロセッサの性能に大きく影響する。

このため、コア間の通信のに利用するルータの研究が重要視されている。ここでは、ソフトシミュレータによる評価が行われることがある。しかし、実装および検証の正確さが不十分であり、またハードウェアソース量の測定精度が低いなどの問題がある。そこで、本研究では多数の FPGA ボードにより構成される ScalableCore システムを用いて、ルータの実装および評価を行う。

2 ScalableCore システム

近年メニーコアプロセッサに関する研究が進んでいる。現在ソフトウェアシミュレータを使用し、研究を検証することが多い。しかし、コア数が増加するとシミュレーション速度が大幅に低下する。そこで、本研究室はコア数ごとに複数の FPGA ボードを接続し、実装することで、スケラブルにシミュレーションを高速化できるプロトタイピングシステム ScalableCore[1] を独自に開発している。ScalableCore の利用より RTL でソースコードを記述することで、マルチコア環境であるハードウェアの構成を自由に変更し、実装と検証ができる。また、現在多くのベンチマークが実装されていて、構築したハードウェアの性能評価が容易に行える。図 1 に 4 × 4 の ScalableCore の実物を示す。

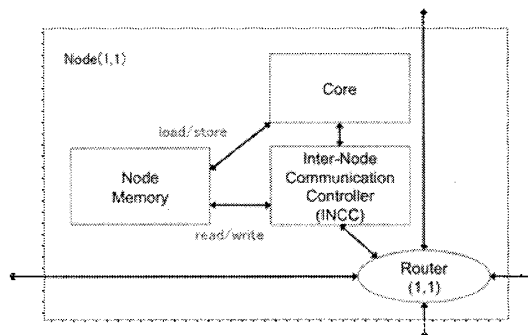


図 2: M-core ノードの構成

2.1 M-Core アーキテクチャ

現在 ScalableCore は M-Core アーキテクチャ[2] を基に構成されている。ネットワークポロジは 2D メッシュであり、ネットワークがノードと呼ばれる要素に接続されている。図 2 にノードの構成を示す。各ノードはノードメモリと演算ユニット PE (Processing Element) とネットワークコミュニケーション用の DMAC 機能を持つ INCC (Inter Node Communication Controller) とルータで構成される。ルータが近隣の四つのルータとローカル INCC と接続されている。あるノードがほかのノードとデータ通信を行う場合はまず演算ユニット PE が INCC に命令を発行する。INCC は届いた命令をデコードし、ノードメモリから指定されたアドレスのデータを読み込み、パケット (次章に説明する) を生成する。パケットはルータを介し、ほかのノードに転送される。

2.2 パケットの構成

パケットはいくつのフリットで構成される。フリットは転送される順番によりヘッドとボディとテールとして分類できる。ヘッドは一番目に転送される 1 フリットであり、中に 12 ビットの宛先ノードの ID を保持する。テールが最後の 1 フリットである。その間のフリットはボディである。フリットの長さは 15 bit であり、その内の 3 bit がフリットの種類を表す制御ビットである。フリットが有効データであることを示す valid ビットが 1 ビット、フリットがヘッドであることを示すヘッドビットが 1 ビット、フリットがテールであることを示すテールビットが 1 ビットである。ヘッドビット、テールビットが共に 0 で、valid ビットが 1 であるフリットが有効なボディとなる。

3 ルータの実装

図 3 にルータのアーキテクチャを以示す。ルータは五つの入力 buffer と Cross-bar と arbiter で構成される。フ

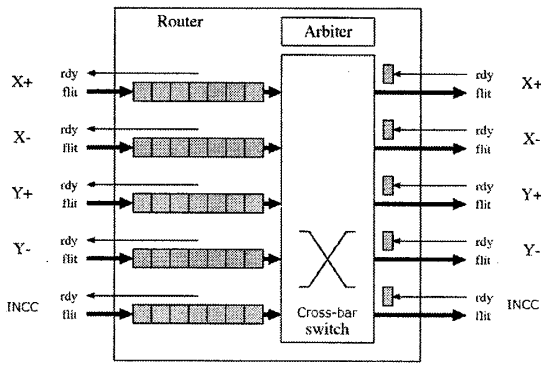


図 3: ルータアーキテクチャ



図 4: ルータ処理ステージ

フリットは入力ポートから入力され、入力 buffer に格納され、Cross-bar を経由し、出力ピンに出力される。arbiter は 5-in-5-out の Cross-bar を制御するユニットである。本稿では単一サイクルのルータを実装対象とする。これは、毎サイクルごとに 1 フリットを処理することになっている。図 4 に処理フローチャートを示す。IF(Inbuffering) は入力ピンからデータを受け、バッファに格納するステージである。RC(Ruter Computation) はバッファから出力されたフリットがヘッドであるならこのフリットに保存されている宛先 ID と自身の ID 番号とルーティングアルゴリズムよりヘッドフリットが属するパケットの出力方向を計算する。ルーティングアルゴリズムは X、Y 次元順ルーティングを採用する。SA(Switch Allocation) は arbiter の制御信号を生成するステージである。ヘッドフリットなら RC ステージの計算結果を用い、出力ポート競合が生じないなら arbiter に制御信号を用意し、次のステージでルートが決まる。テールフリットなら arbiter に次のクロックのフリットが属するパケットが通過させる信号を生成する。よって、ヘッドフリットが Cross-bar を通るとテールフリットが出力されるまでその出力ポートからほかのパケットのフリットは出力されない。これは 1 パケット中にほかのパケットのフリットが混じりこまないことを保証する。そして、スイッチングテクニックが wormhole を採用し、小容量のルータバッファを用いる。ST(Switch Traversal) は SA の信号より cross-bar を制御するステージである。OP(Output Data) は cross-bar からのフリットを出力ポートに出力する。

表 1: ISE 論理合成結果

エントリ数	スライス数	最大周波数
2	747	65.863MHz
4	991	63.432MHz
8	1108	58.278MHz

表 2: n-queen の実行結果

エントリ数	時間
2	9.3sec
4	7.7sec
8	1.9sec

4 ルータの評価

Verilog で記述し、ルータのバッファサイズを変えたときの Xilinx の ISE によるハードウェア量と最大動作周波数の結果を表 1 に示す。そして、ScalableCore を使用し、ベンチマークとして n-queen を使用し、ルータの評価を行う。このベンチマークではマスタコアがほかのコアにタスクを割り当て、実行させ、各コアが最後に実行結果をマスタコアに転送する。また、実行時間を計測することにより性能を評価する。表 2 に n-queen の実行結果を示す。ルータのバッファを大きくするとハードウェアリソースが増加し、クリティカルパスが長くなる一方で、ネットワークのオーバーヘッドを緩和でき、マルチコアプロセッサ全体の性能向上ができる。

5 まとめ

M-core アーキテクチャのルータを HDL で記述し、ハードウェアソース量を測定した。また、本研究室が独自に開発しているメニーコアプロセッサの高速プロトタイプシステム ScalableCore を用い、ルータの実装と検証を行った。そして、ルータバッファサイズを調整し、一部の性能評価を行った。今後はルータのパイプライン化や仮想チャンネルやヘッド予測などの機能を持つルータを開発し、実装と性能評価を行う。

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。

参考文献

- [1] 高前田 伸也 渡邊 伸平 姜軒 藤枝 直輝 植原 昂 三好 健文 吉瀬 謙二
メニーコアアーキテクチャ研究のためのスケーラブルな HW 評価環境 ScalableCore システム
IJPSJ SIGARC 2009.8.5
- [2] Koh Uehara, Shimpei Sato, Takefumi Miyoshi, and Kenji Kise A Study of an Infrastructure for Research and Development of Many-Core Processors
UPDAS 2009.12.11