

## メニーコアプロセッサの高速プロトタイプシステム ScalableCore の提案

高前田 伸也<sup>†</sup> 渡邊 伸平<sup>‡</sup> 吉瀬 謙二<sup>‡</sup>東京工業大学 工学部情報工学科<sup>†</sup> 東京工業大学大学院 情報理工学研究科<sup>‡</sup>

## 1 はじめに

近年, 1チップ上に複数のコアを集積するマルチコアプロセッサが主流となっており, 今後は数十から数千のコアを集積するメニーコアプロセッサが主流になると考えられる.

効率的なプロセッサの研究開発を行うには, 迅速にシステムを検証する必要がある. 検証にはソフトウェアによるシミュレータが用いられることが多いが, メニーコアの場合, シミュレーション時間が急激に増加してしまう. その為, 高速な検証が行える環境が求められている. そこで本稿では, FPGA を用いた高速プロトタイプシステム ScalableCore を提案する. 独自開発のボードを用いて安価な FPGA を複数接続することで, コストを抑えつつ高速なシミュレーションを可能とする.

## 2 M-Core アーキテクチャ

本稿では M-Core アーキテクチャ [1] をプロトタイプ化の対象とする. 図 1 にその構成を示す. M-Core アーキテクチャは, シンプルで均一なコアを多数並べる構成を採用し, システムソフトウェアとの協調によりチップの高性能化を目指している. 複雑な割込み機構は持たず, コアの効率化を図っている.

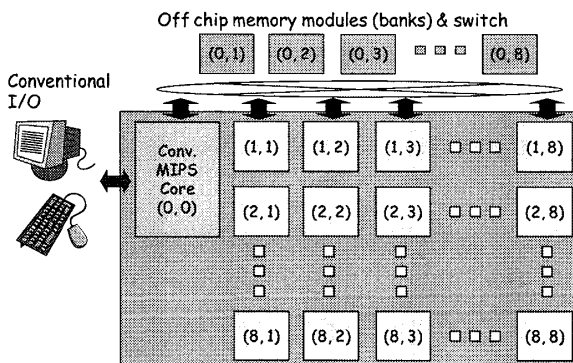


図 1: メニーコアプロセッサアーキテクチャ M-Core. 汎用プロセッサコアである Module(0,0) とメッシュ状に配置された多数のノードを搭載し, オフチップメモリと接続する.

タイル状に並べられているそれぞれをノードと呼ぶ. 各ノード内部にはコアとルータがあり, コアの中に PE(Processing Element), DMAC(Direct Memory Access Controller), ノードメモリがある. DMAC は他ノ

ScalableCore: High-Speed Prototyping System for Many-Core Processors

Shinya TAKAMAEDA<sup>†</sup>, Shimpei WATANABE<sup>‡</sup>, Kenji KISE<sup>‡</sup>

<sup>†</sup>Depart. of Computer Science, Tokyo Institute of Technology

<sup>‡</sup>Graduate School of Information Science and Engineering, Tokyo Institute of Technology

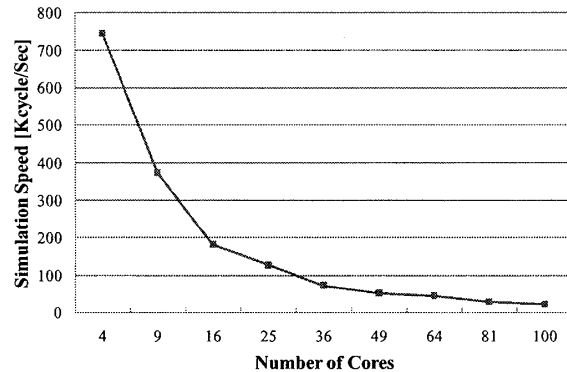


図 2: SimMc によるシミュレーション速度.

ドのメモリもしくはオフチップメモリに DMA を発行する. ルータは近隣 4 つのルータと 1 つの DMAC に接続されており, データ転送を制御する機能を持つ.

M-Core モデルのシミュレータに SimMc [1] がある. SimMc にて並列アプリケーション Equation Solver Kernel を実行した際のコア数とシミュレーション速度の関係を図 2 に示す. 実行環境は CPU: Intel Xeon X5365 3.00GHz, Memory: 16GB である. シミュレーション対象のコア数が増えるにつれて, 通信の数が増加するため速度が低下する. 100 コアのシミュレーションを行うと, 1 秒あたりの実行サイクル数は約 22K サイクルと低速である. 本モデルを FPGA でハードウェア実装し, 2.2MHz で動作させれば, 約 100 倍の高速化となる.

## 3 ScalableCore の提案

本稿で提案する ScalableCore は FPGA によるプロセッサプロトタイプシステムである. ScalableCore では, プロセッサ自体に重点を置いており, プロセッサの構成の変更などにも柔軟に対応できる. また独自開発したボード (図 4) は必要に応じて任意に追加することができるので, コア数の変更が容易である.

FPGA を用いたプロトタイプシステムとして RAMP [2] があるが, マルチプロセッサを対象としている点や計算機全体をシミュレーションの対象としている点などで異なる.

## 3.1 構成

ScalableCore は主に ScalableCoreNode (以下 Node) と ScalableCoreBoard (以下 Board) により構成されている. Node は M-Core におけるノードであり, 単一の FPGA ボードで実装されている. Node は PE, DMAC, ルータのほか, 近隣の FPGA と通信を行う Serializer/Deserializer モジュールとデバッグ用液晶のコント

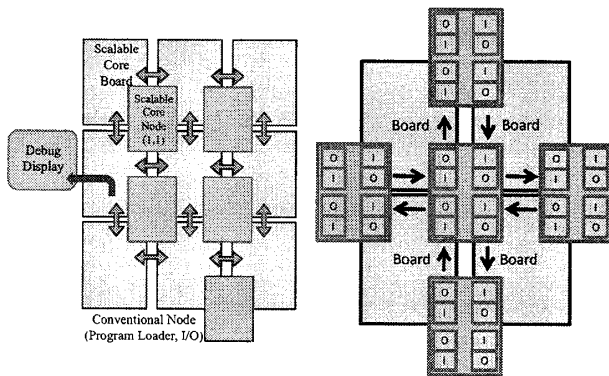


図 3: ScalableCore の構成. 左:全体の構成 右:通信の方向

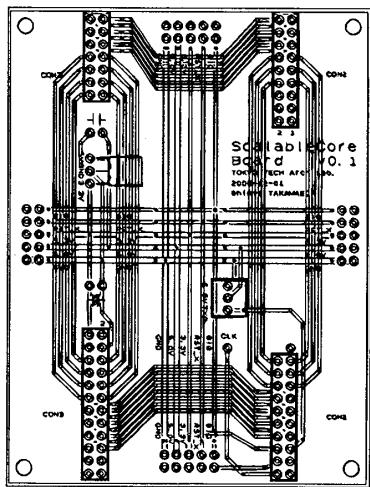


図 4: ScalableCoreBoard. 中央の十字配線がグローバルワイヤ、周りの配線がシリアル通信用。四隅に Node が載る

ローラによって構成されている。また今回の構成ではオフチップメモリは考慮していない。

市販の FPGA ボードの多くは基板の端に I/O ポートがあり、形状が長方形である。ScalableCore はその点に注目し、I/O ポートを 4 分割し各方向へ規則的に割り当てている。図 3 のように 4 枚の Board が 1 つの Node を取り囲むように配置すると、Board が Node 同士を接続するワイヤの役目を果たし複雑な配線が不要となる。この接続方法により、コア数を任意に調整することができ、スケーラビリティの確保が容易となる。各 Board 間は数本のワイヤで直接接続され、電源供給やグローバルクロック・リセットなどに用いられている。

Node と Board 以外の要素として、Debug Display と Conventional Node がある。Debug Display にはコマンドインタプリタ型液晶を用い、各 Node の通信状態やレジスタの内容などを表示する。Conventional Node は M-Core における Module(0,0) であり、プログラムのロードなどを行う。

### 3.2 通信機構の初期検討

初期検討として、図 5 に示すように、4 個の FPGA を  $2 \times 2$  のメッシュ状に接続し、それぞれの通信機構が動作していることを以下の手順で確認する。各 FPGA

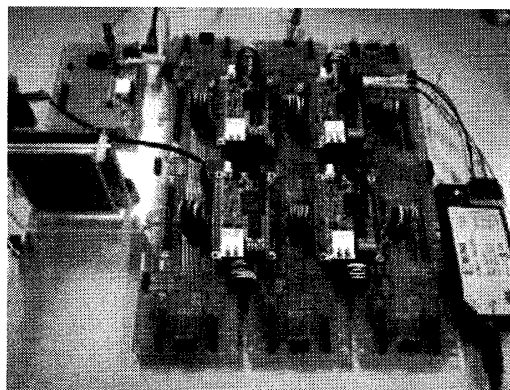


図 5:  $2 \times 2$  の ScalableCore. 左にあるのがデバッグ用コマンドインタプリタ型液晶 ITC-2432-035H, FPGA ボードはアットマークテクノ社の SUZAKU-S を使用,

には Serializer/Deserializer モジュールとデバッグ液晶用モジュール、テスト用トップモジュールが実装されている。

図 5 において右下、右上、左下、左上に位置する Node をそれぞれ Node0, Node1, Node2, Node3 とする。Node0 は自らが生成したクロックを Node1 と Node2 に供給し、Node1 は Node0 から供給されたクロックを Node3 に供給している。また、Node0 は Node1 と Node2 にそれぞれ 40bit の値を送信している。Node1 は Node0 から来た値を次のステップに Node3 に送信する。Node1, Node2, Node3 にそれぞれ接続されたコマンドインタプリタ型液晶に、各 Node が受信した受信した値を表示し、その値が正しいことを確認する。

現在、上記の手順で、1 方向に対して 100Mbps のシリアル転送が 2 系列同時にできていることを確認できた。

### 4 まとめ

メニーコアプロセッサの高速プロトタイピングシステム ScalableCore を提案した。独自開発のボードを用いて安価な FPGA を複数接続することで、低コスト・スケーラブルな高速プロトタイピングを可能とする。今後の課題はルータ、DMAC, PE を実装し M-Core 全体の動作検証を行うことである。

### 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。

### 参考文献

- [1] 植原昂, 佐藤真平, 森谷章, 藤枝直輝, 高前田伸也, 渡邊伸平, 三好健文, 小林良太郎, 吉瀬謙二.  
シンプルで効率的なメニーコアアーキテクチャの開発  
情報処理研究 2008-ARC-180 pp.39-44,2008
- [2] Wawrzynnek, J. and Patterson, D. and Oskin, M. and Lu, S.L. and Kozyrakis, C. and Hoe, J.C. and Chiou, D. and Asanovic, K. RAMP: Research Accelerator for Multiple Processors IEEE Micro pp.46-57, 2007