

DRAM リソースに着目した メニーコアソフトプロセッサの性能チューニング

森 悠[†] 吉瀬 謙二[†]

[†] 東京工業大学 大学院情報理工学研究科

1 はじめに

プロセッサに搭載されるコア数は日々増加しており、数十から数千のコアを搭載するメニーコアプロセッサが主流になりつつある。今後のメニーコアプロセッサの研究に対応していくため、FPGA 上で実現されるソフトプロセッサとしてメニーコアソフトプロセッサを設計した¹⁾。設計したプロセッサは大規模 FPGA を搭載する FPGA ボードをターゲットとしており、ボード上に搭載される DRAM リソースを利用した構成となっている。本稿では、この DRAM リソースに着目し DRAM 性能を引き出すようなチューニングをプロセッサに与える。

2 メニーコアソフトプロセッサ

2.1 アーキテクチャ

メニーコアプロセッサの研究基盤とするべく、大規模 FPGA 上で動作するフレキシブルなメニーコアプロセッサを設計した。設計したメニーコアプロセッサはハードウェア記述言語 (HDL) による 2,400 行程度のコードで記述され、ユーザは要求性能や研究要素、実装するターゲットデバイスに応じてプロセッサ数やキャッシュ容量などの構成を変更することが可能である。

設計したメニーコアプロセッサは内部接続として 2 次元メッシュ状のネットワークで構成される Network on Chip²⁾³⁾ を採用した。プロセッサやキャッシュとしての機能を持つノードはメッシュネットワークで相互に接続され、ノード同士はネットワークを介したパケット交換によりノード間通信を行う。

設計したメニーコアプロセッサは数種のノードによって構成され、32bit の MIPS アーキテクチャを採用するパイプラインプロセッサ⁴⁾⁵⁾ と L1 キャッシュから成るプロセッサノード、L2 キャッシュとして機能するキャッシュノード、プロセッサのスケジューリングを行うためのスケジューラノード、メインメモリへのアクセス権を持つメモリノード、ノード間通信を仲介するフロートノードが存在する。

図 1 に 36 ノード構成のメニーコアプロセッサのアーキテクチャを示す。P, C, S, M, F および数字が付された円はノードを表し、それぞれプロセッサノード (P)、キャッシュノード (C)、スケジューラノード (S)、メモリノード (M)、フロートノード (F) を意味する。図 1 の実線は二次元メッシュネットワークにおけるノード間の物理的配線を、色掛けされた領域はプロセッサノードとキャッシュノードが構成するペアを表している。キャッシュノードはプロセッサノードとペアを組むことで、プロセッサノードのプライベートキャッシュとして機能する。

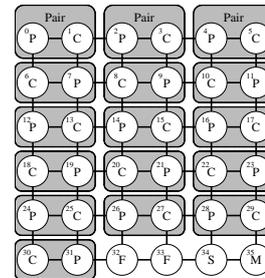


図 1: 36 ノード構成のメニーコアプロセッサ

図 1 ではプロセッサノードとキャッシュノードが交互に配置されており、隣接するプロセッサノードとキャッシュノードがペアを構成している。たとえば、図 1 中のノード番号 18 と 19 がペアに相当する。

メモリ・キャッシュメモリは階層化されており、メモリシステムはプロセッサごとに独立した分散メモリの形態を取る。すなわち、プロセッサとキャッシュの組は自身のメモリ領域へのアクセス要求をメモリノードに対して行い、メモリノードはその要求を逐次処理している。

2.2 実装と制約

大規模 FPGA である Virtex-7 XC7VX485T を搭載する VC707 Evaluation Board を用い、設計したメニーコアプロセッサを FPGA に実装した。この FPGA ボードは RAM リソースとして、FPGA 内部に BRAM (Block RAM) と呼ばれる On-Chip RAM と、ボード上に取り付けられた Off-Chip DRAM モジュールの両方を有する。

これまでの実装では、On-Chip RAM の移植性の高さとアクセスの手軽さから、メインメモリリソースとして BRAM を利用してきた。しかし、メモリ容量が大きく制限される BRAM だけでは複雑で大規模なアプリケーションをメニーコアプロセッサ上で動作させたいという要求を満たすことができないため、VC707 をターゲットとして Off-Chip DRAM を利用した実装を行った。

FPGA およびボードの開発元である Xilinx の提供する DRAM 制御のための IP コアを利用し、メモリノードのアクセス対象を BRAM から DRAM に置き換えたところ、メモリアクセスが頻発するアプリケーションにおいて実行サイクル数の増加および並列処理性能の低下がみられた。

図 2 に並列行列積計算において、処理コア数を変化させたときの相対性能を示す。相対性能は実行に要したサイクル数に基いて算出され、系列は BRAM を利用した版と DRAM を利用した版、および相対性能が処理台数に比例する理想値である。メモリリソースを BRAM から DRAM に変更したところ、16 コア実行時の相対性能が 13 倍から 3 倍へと低下している。これはメインメモリへのアクセスレイテンシが増加し、メモリノードの処理がボトルネックとなっていることに起因する。本稿では DRAM リソースの性質に着目した性能チューニングを行い、メニーコアプロセッサの性能向上を目指す。

Performance tuning of a manycore soft processor focusing on DRAM resources

Haruka MORI[†], and Kenji KISE[†]

[†]Graduate School of Information Science and Engineering, Tokyo Institute of Technology

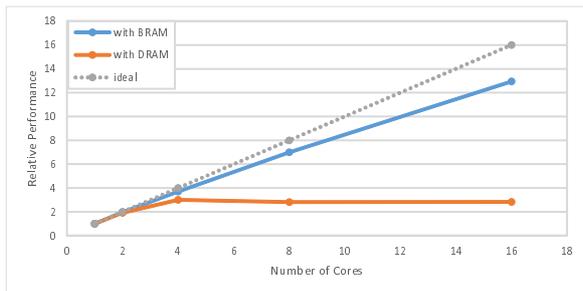


図 2: メモリリソース変更に伴う相対性能の変化

表 1: 1GB バーストアクセス時の DRAM 帯域

	変更前	変更後
write cycle	20,132,690	10,066,635
read cycle	18,705,013	9,352,508
write bandwidth	5.0 [GB/sec]	9.9 [GB/sec]
read bandwidth	5.3 [GB/sec]	10.6 [GB/sec]

3 チューニング手法

3.1 DRAM の高速動作

DRAM のアクセスレイテンシを短縮する手法の一つとして、DRAM の動作周波数の引き上げが挙げられる。提供されている IP コアは DRAM コントローラと DRAM のクロック比率を 1:4 としており、メニーコアプロセッサおよび DRAM コントローラの動作周波数を 200MHz とすることで、DRAM を最大動作周波数である 800MHz で駆動することができる。しかし、現在の版のメニーコアプロセッサの安定動作周波数は 100MHz 程度であるため、DRAM が持つ本来の帯域性能を十分に活かすことができていない。

この問題を解消するため、DRAM コントローラにバッファリング機能を付し、メニーコアプロセッサと DRAM コントローラの動作クロックを分離した。メニーコアプロセッサと DRAM コントローラ、DRAM のクロック比率を 1:2:8 とすることで、メニーコアプロセッサと DRAM をそれぞれの最大動作周波数で駆動させる。

表 1 に 1GB のデータを DRAM に対しバースト書き込み/読み込みしたとき、動作に要したサイクル数と平均スループットを示す。検証に用いた回路の動作周波数は 100MHz であり、変更前と変更後の DRAM の動作周波数はそれぞれ 400MHz と 800MHz である。変更により検証に要したサイクル数はほぼ半分となり、約 2 倍のスループットが得られた。また、DRAM が毎サイクルのアクセスが可能であると仮定すると、スループットの最大理論値は 12.8GB/sec であり、検証におけるアクセスパターンでは理論値に対し 8 割程度のスループットを達成することができた。

3.2 ブロックサイズの調整

DRAM のアクセスレイテンシを短縮するもう一つの手法として、DRAM 内部の Row buffer を活用する方法が挙げられる。DRAM が自身のメモリセルアレイにアクセスするとき、まずメモリセルへの Row アクセスにより Row がまとめて Row buffer に格納される。その後、Column アクセスにより Row buffer の中から所望の Column が選択される。次の DRAM アクセス時、もし前のアクセスと Row が同じであれば Row buffer hit となり、Row アクセス分のレイテンシを削減することができる。Row buffer

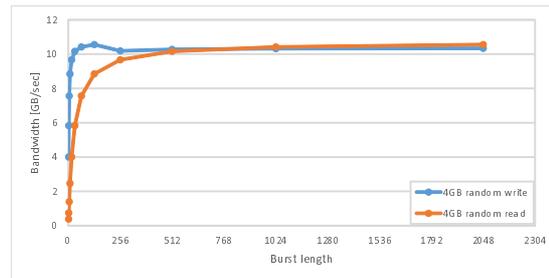


図 3: バーストアクセス時のスループット
に格納されているデータには空間的局所性が存在するため、バーストアクセスすることが一般に望ましい。

図 3 にバースト長を変えて計 4GB のランダムアクセスを行ったときのスループットを示す。block は DRAM コントローラにアクセスする最小の粒度であり、1block あたり 64byte である。128block(8KB) 単位で書き込みのスループットがピークに、256block(16KB) 以上の単位で読み込みスループットの上昇が概ね飽和することがわかる。現在の仕様ではキャッシュのブロックサイズが 64byte 以下であるためバーストアクセスは行っておらず、スループットは 1block ごとのアクセスに近い値となっている。この手法を適用するためにキャッシュのブロックサイズをより大きくし、積極的にバーストアクセスを行うことが求められる。

4 まとめと今後の課題

設計したメニーコアプロセッサがより大きなメモリ領域を扱えるよう、Off-Chip DRAM を用いた実装を行った。また、DRAM の構造や特性に着目し、設計したメニーコアプロセッサが DRAM 性能を引き出せるようなチューニング方法を考案した。考案した手法を設計したメニーコアプロセッサに適用し、並列アプリケーションによる詳細な評価を行うことが今後の課題である。

謝辞

本研究は MEXT 科研費 25330056 の助成を受けたものです。

参考文献

- 1) Haruka Mori and Kenji Kise. "Design and Performance Evaluation of a Manycore Processor for Large FPGA", 2014 IEEE 8th International Symposium on Embedded Multicore/Manycore SoCs, 207-214.
- 2) Shashi Kumar, Axel Jantsch, Juha-Pekka Soinen, Martti Forsell, Mikael Millberg, Johny berg, Kari Tiensyrj, and Ahmed Hemani. "A network on chip architecture and design methodology", ISVLSI '02 Proceedings of the IEEE Computer Society Annual Symposium on VLSI, 105-112.
- 3) Benini, L and Bertozzi, D. "Network-on-chip architectures and design methods", IEE Proceedings - Computers and Digital Techniques, 2005, 261-272.
- 4) David A. Patterson and John L. Hennessy. "Computer Organization and Design, Fourth Edition: The Hardware/Software Interface", 2011, Morgan Kaufmann.
- 5) John L. Hennessy and David A. Patterson. "Computer Architecture, Fifth Edition: A Quantitative Approach", 2011, Morgan Kaufmann.