Vol.2011-OS-118 No.2
2011/7/27

OS 開発のための メニーコアハードウェアシミュレータの 設計と実装

澤田武男[†1]　辻田祐一[†3]　並木美太郎[†4]
堀敦史[†5]　石川裕[†1,†2,†5]

今後の高性能計算機においては，数十から数百以上のコアを集積したメニーコア環境が重要な役割を負う．メニーコア環境では，これまでのマルチコア環境とは違った OS カーネルやシステムソフトウェアが要求されるが，それらの開発の際に必要な実験向けメニーコア環境は，現在はまだ一般に入手できない．

このような状況でも OS の開発を進めるために，本研究ではアクセラレータタイプのメニーコア環境を FPGA を用いてシミュレーションするシステムを設計する．また，HDL による実装の前段階として，Gem5 フルシステムシミュレータ上にメニーコア環境をモデリングする．

Design and Implementation of a Many Core Hardware Simulator for OS Development

Takeo Sawada,[†1] Yuichi Tsujita,[†3] Mitaro Namiki,[†4]
Atsushi Hori[†5] and Yutaka Ishikawa [†1,†2,†5]

Manycore processors, which have more than dozens of cores, will play large role in high-performance computing (HPC) in near future. Manycore environments require kernels and other system software to be designed differently from multicore counterpart. However, manycore environments that are necessary to develop those system softwares is not generally available currently.

Our study aims designing and implementing a simulator of manycore environment in FPGA for those system software development. Prior to implementing the system in FPGA, we model the manycore processor on Gem5 full-system simulator.

1. はじめに

1.1 背　景

高性能計算機においては，単一コアあたりの性能をさらに向上させることよりもより多くのコア数を搭載することを重視した設計が行なわれている．今後の高性能計算機においては，メニーコア (Manycore) と呼ばれる，シンプルな設計のコアを数十から数百以上並べたプロセッサが使われていくものと予想される．[2)]

メニーコアの利用形態としては，メニーコアがメインの CPU であるホモジニアスな形態と，メイン CPU にはマルチコアプロセッサを利用し，それとは別にメニーコア環境を PCI Express バスなどを用いてホスト CPU と接続しアクセラレータとして使用するヘテロジニアスな形態がある．前者としては Single Chip Cloud (SCC)[6)] などがあり，後者としては Intel のアクセラレータ型のメニーコアボードである Larrabee[14)] や Knights Corner[7)] などの Many Integrated Core (MIC) がある．Texas Advanced Computing Center (TACC) はこれらの Intel のメニーコアボードを使用した高性能計算機を導入すると発表しており[16)]，アクセラレータ型のメニーコアボードを利用した高性能計算機は今後数年以内に実現されると考えられる．

アクセラレータ型の計算コアとしては GPGPU が現在広く使われている．GPGPU にはコア間の同期や IO を行なう機構が無く，ホスト CPU に同期処理などを任せる必要があり，スケールさせる際にボトルネックになり得ることが挙げられている．[9)] アクセラレータ型メニーコア環境の利点を効率的に活用するには，メニーコアボードをこれまでの GPGPU のような単なる計算のみを行なうアクセラレータとして使用するのではなく，これまでホスト CPU が行っていた通信や IO などの動作を単体で行える必要がある．例えばメニーコアの各

†1 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
†2 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
†3 近畿大学工学部
Faculty of Engineering, Kinki University
†4 東京農工大学
Tokyo University of Agriculture and Technology
†5 理化学研究所 計算科学研究機構
RIKEN AICS

コア間やホスト CPU，別のホストなどと通信を行なったり，ファイルからの入出力を行なったり，プロセッサを時分割して使用したりといった要求が考えられる．これらの要求を実現するには，各コア上で OS カーネルが動作していることが必要である．しかし，メニーコアプロセッサでは現在のマルチコアプロセッサと比べてコアあたりの演算性能やキャッシュ容量が低く，これまでの SMP 環境で使われてきた Linux などの大きなカーネルを使用すると演算性能が大きく低下してしまう．また，問題サイズを変えずにコア数を増やすことで計算速度を向上させる Strong scaling を実現するには，計算と通信のオーバーラップを実現すると共に低レイテンシの高速な通信を実現する必要がある．よって，メニーコア環境向けの軽量カーネルの設計が必要である．

1.2 本研究の目的

1.1 節で述べたように，メニーコア向けのシステムソフトウェアの研究はまだ充分であるとは言えない状況である．しかし，現在メニーコアボードは一般に入手が困難で，実機を用いてメニーコア上での研究を行なうのが難しい．そこで本研究では，このような状況でもメニーコア向けシステムソフトウェアの研究を進めるために，安価に入手可能である FPGA を用いた実用的な速度で動作するアクセラレータ型のメニーコアボードを開発する．

本稿では，まず開発用メニーコアボードに求められる要件を定義し，コア部分の仕様を定義した．また，Hardware Description Language (HDL) による実装の前段階として，HDL 実装のデバッグやホストカーネルのデバッグのために Gem5 フルシステムシミュレータ[5] にメニーコアボードをモデリングし，ホスト CPU 上での Linux カーネルの動作を含めたフルシステムシミュレーションを行なう．

2 章では，メニーコア環境やメニーコアにおけるシステムソフトウェアについての関連研究について述べる．3 章では，FPGA におけるメニーコアプロセッサの設計について述べる．4 章では，ソフトウェアによるメニーコアシミュレータの設計について述べ，5 章がまとめになっている．

2. 関 連 研 究

2.1 IBM Blue Gene/L

Blue Gene は PowerPC 440 ベースのコアで構成されたスーパーコンピュータである．[1] Blue Gene/L のノードは Compute Node と I/O Node からなり，I/O Node では Linux カーネルが動作しているが，Compute Node では Compute Node Kernel (CNK) と呼ばれる POSIX のサブセットのみをサポートする機能が制限された軽量なカーネルが動作している．[12] CNK はシングルユーザ，シングルプロセスのみをサポートしている．Linux のような巨大なカーネルの代わりに CNK を利用することにより，Blue Gene は大幅な性能向上を達成している．[15]

2.2 Larrabee

Larrabee[14] は Intel が 2008 年に発表したアクセラレータ型のメニーコアボードである．P54C コアにいくつかの命令を追加したインオーダ実行のコアが 8 から 48 個程度接続してある．ボード上にあるメモリのメモリ空間は全部のコアで共有されており，各コアはリングバスで接続されている．キャッシュコヒーレンシーはハードウェアで保障される．

2.3 RAMP

Research Accelerator for Multiple Processors (RAMP)[18] は，RAMP Red，RAMP Blue[8]，RAMP Gold[17] などのサブプロジェクトからなる．メニーコア環境を FPGA でシミュレーションする基盤を実装することで，メニーコアにおけるハードウェアやシステムソフトウェア，プログラミングモデルの研究の推進を目的としている．ソフトウェアシミュレータのみでメニーコアをシミュレーションしようとすると，シミュレーションに時間がかかりすぎることが問題となる．RAMP では，部分的に FPGA を用いてシミュレーションを加速することができ，100MHz 程度の速度でメニーコア環境をシミュレーションできるとしている．また，関連する研究に FAST[4] や HASim[13] がある．

2.4 M-Core

M-Core プロジェクトは東京工業大学で行なわれている，メニーコアプロセッサの研究と教育を支援する基盤である．[21] シンプルなメニーコアアーキテクチャである M-Core アーキテクチャが提案されており，そのサイクルレベルシミュレータである SimMc が開発されている．また，SimMC のシミュレーションを FPGA を用いて高速化する ScalableCore システム[22] も提案されている．

しかし M-Core はメニーコア部分のみのシミュレーションにフォーカスしており，ホスト CPU を含めたヘテロジニアス環境のフルシステムのシミュレーションが行なえない．また，OS 開発の為には，ホスト CPU との通信やファイル IO，ネットワーク通信が再現できる必要があが，M-Core は単純な計算ノードとして使われることが想定されており，OS の動作に必要な特権モードや割り込みといった機能を供えていない．以上のことから，M-Core および SimMc は本研究には適さない．

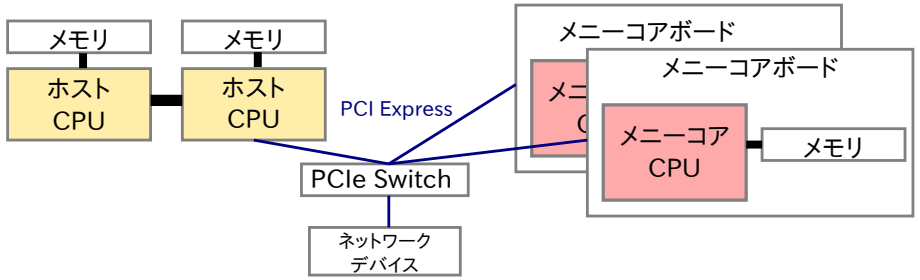

図 1 全体構成図
Fig. 1 Overall structure

3. 設　　計

図 1 は，本研究で作成するシステムの全体構成を示す．1 台のノードは，ホストとなる PC アーキテクチャのマシンと，それに PCI Express で接続されたメニーコアボードとネットワークインターフェースカードなどからなる．ホストマシンには一般的な Xeon や Core i などのマルチプロセッサ CPU が搭載されていて，カーネルとして Linux が動作する．ノード同士は InfiniBand を使用して接続される．

3.1 実 装 環 境

実装に使用する FPGA 環境は，価格と性能，入手し易さなどを考慮し，Xilinx 社の ML605 評価キット[19] を選んだ．ML605 は，XC6VLX240T-1FFG1156 FPGA と，512MB DDR3 SO-DIMM，Gen2 で x4 まで設定可能な PCI Express エッジコネクタがなどが 1 枚のボードに実装された評価ボードである．

3.2 アクセラレータの設計

図 2 は，アクセラレータ部分の設計を表している．

目標とするアクセラレータが満たすべき要件として，以下のようなものが考えられる．(i) タイミングシミュレーションが出来ること．これはパフォーマンスの計測を行なうのに必要である．(ii) コンパイラなどは出来る限り既存の開発リソースが使用できること．コンパイラの研究ではないので，独自コンパイラを開発するのではなく既存の開発ツールと互換性のあるコア設計にする必要がある．(iii) ハードウェアでの実装が容易であること．これは目標である FPGA を利用した高速なシミュレーションに必要である．(iv) 1 枚の FPGA ボードで実現できる規模と複雑さであること．これもこのシステムの利用が容易であるようにするために必要である．(v) OS 開発に必要なデバッグ機能やタイミングデータを取得できること．

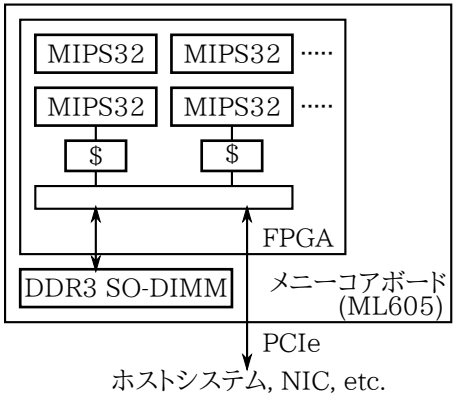

図 2 アクセラレータ部分の設計
Fig. 2 Design of the Accelerator Part

以上に挙げた 5 つの制約条件を満たすように，メニーコアボードの仕様を決定した．各コアは，インオーダー動作の MIPS32 アーキテクチャを採用した．スーパースカラ動作や SMT は行なわない．OS カーネルを動作させるため，全てのコアは特権命令と割り込みをサポートする．FPU は搭載せず，整数演算命令のみをサポートする．全てのコアが同一のメモリーアドレス空間を共有する．キャッシュコヒーレンシはハードウェアでは保障しない．

3.3 アクセラレータとホストのインターフェース

アクセラレータボードとホストとの接続は，ML605 に搭載されている PCI Express バスで行なう．これにより，ホスト CPU の持つメモリや NIC と高速な DMA を通してデータ送受信が行なえる．

3.4 ホストカーネル側ドライバ，ユーザースペース

PCI バス上に接続したメニーコアボードを制御するには，ホストカーネル側のドライバとユーザースペースとのインターフェースが必要である．本研究では，カーネル側のインターフェースに Accelerator Abstraction Layer (AAL)[23] を採用する．AAL はメニーコア環境を抽象化し，様々なアクセラレータタイプのメニーコアボードをカーネルやユーザースペースから統一的に扱えるようにする抽象化レイヤーである．AAL に対して実装したメニーコアボードへの下位デバイスドライバを実装した．

ユーザースペースからメニーコアボードを制御するには，AAL が提供する /dev/mcd*デバイスや/dev/mcos*デバイスを利用して行なう．

4. ソフトウェアによる実装

ML605 上に HDL でメニーコアアクセラレータを実装する前段階として，HDL デザインやホスト側のカーネルのドライバのデバッグや動作検証のためにメニーコアアクセラレータをソフトウェアで実装する．このソフトウェア実装は，例えばメニーコア側に何らかの機能追加などを行なう際のデバッグや，より詳細なタイミングを検証したい場合にも利用できる．ホスト側のカーネルとアクセラレータの通信をデバッグするためには，ホスト CPU を含めたシステム全体をシミュレーションできるフルシステムシミュレータが必要である．

本研究では，このシステムを再現するシミュレータに Gem5 シミュレータ[5] を用いた．Gem5 は，M5 シミュレータ[3] と GEMS シミュレータ[11] を統合したシミュレータである．M5 の持つ CPU モデルや ISA 定義と，GEMS の持つ Ruby メモリ (キャッシュコヒーレンシ) などを統合している．x86，Alpha，PowerPC，ARM などの主だった ISA の多くをサポートし，Out of Order 実行をモデリングしたサイクルレベルのシミュレーションと高速な機能レベルシミュレーションの両方をサポートしている．C++と Python をで書かれており，用意されたオブジェクトをインスタンス化しつなぎ合わせることでシステムを柔軟に表現でき，例えば複数のホストが TCP/IP を用いて通信する環境をモデリングすることも可能である．大部分が BSD-like なオープンソースライセンスで公開されており，研究をはじめとしたあらゆる目的に自由に利用可能である．

既存のフルシステムシミュレータとしては，Gem5 の他にも Simics[10] や PTLSim[20] などが著名な物として挙げられる．本研究で使用するシミュレータについて，上の 2 つのシミュレータについても適切であるかどうか検討した．Simics はライセンスの制限が厳しく，メニーコアの研究に自由に使用することを目的とした本研究には適合しにくいと考え，除外した．PTLSim は，x86_64 アーキテクチャのみをサポートしており，本研究に必要な柔軟なシステム構成を再現しにくいと考え，除外した．

5. おわりに

5.1 まとめ

プロセッサはシングルコアの性能を追い求めるかわりに，より多くのコアを集積することを重視して設計されるようになった．今後の高性能計算では数十から数百以上のコアを集積したメニーコア環境が利用されることが予想される．メニーコア環境の上にこれまでホスト CPU で行なわれていた処理を移行する為には，高性能計算で要求される通信や IO などの機能を実現する必要がある．このためにはメニーコアの各コア上でカーネルを動作させる必要があるが，メニーコアのコアあたりの性能はホスト CPU に比べて低いため，これまでホスト CPU で使われてきたような大きなカーネルでは計算性能が低下してしまう．メニーコア向けの軽量カーネルを設計する必要があるが，現在メニーコア環境は一般には入手しづらいため，OS カーネル研究のためのメニーコアシミュレーション基盤が必要である．

本研究ではメニーコアの OS カーネルの開発を進めるために，FPGA を利用したアクセラレータ型のメニーコアボードを実装することを目標にしている．本稿では OS 開発のためのメニーコアボードに必要な機能を列挙し，その仕様を定義した．また，HDL での実装の前段階として，HDL やドライバやカーネルのデバッグのために，Gem5 シミュレータ上にメニーコアプロセッサをソフトウェアで実装する．

5.2 今後の課題

ソフトウェアで実装されたメニーコアアクセラレータは，コア数が増加するに従ってシミュレーションにかかる時間が大幅に増えてしまい，実用的なシミュレーションを行なうのが困難である．まずは今回ソフトウェア上で実装したメニーコアボードをハードウェアを実装し，利用のしやすさと速度を両立させたシステムソフトウェア開発のための実用的なシミュレーション基盤を作成することが第一の目標である．次に，FPGA で書かれたメニーコア環境には，新しい同期プリミティブ命令や特権命令などを追加することが比較的容易であるので，このシミュレーションシステムを使って，どのようなハードウェア側の支援が軽量カーネルにおける計算効率の向上に有効であるか研究したい．

謝辞 本研究の一部は，戦略的国際科学技術協力推進事業（共同研究型）研究領域「情報通信技術」研究課題名「ポストペタスケールコンピューティングのためのフレームワークとプログラミング」、および科学技術振興機構戦略的創造研究推進事業 (CREST) 研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」研究課題「メニーコア混在型並列計算機用基盤ソフトウェア」による．

参考文献

1) Adiga, N., Almási, G., Almasi, G., Aridor, Y., Barik, R., Beece, D., Bellofatto, R., Bhanot, G., Bickford, R., Blumrich, M. et al.: An overview of the BlueGene/L supercomputer, *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing,*

CD ROM (2002).
2) Asanovic, K., Bodik, R., Catanzaro, B., Gebis, J., Husbands, P., Keutzer, K., Patterson, D., Plishker, W., Shalf, J., Williams, S. et al.: The landscape of parallel computing research: A view from berkeley, Technical report (2006).
3) Binkert, N., Dreslinski, R., Hsu, L., Lim, K., Saidi, A. and Reinhardt, S.: The M5 simulator: Modeling networked systems, *Micro, IEEE*, Vol.26, No.4, pp.52–60 (2006).
4) Chiou, D., Sunwoo, D., Kim, J., Patil, N. A., Reinhart, W., Johnson, D. E., Keefe, J. and Angepat, H.: FPGA-Accelerated Simulation Technologies (FAST): Fast, Full-System, Cycle-Accurate Simulators, *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 40, Washington, DC, USA, IEEE Computer Society, pp. 249–261 (online), DOI:http://dx.doi.org/10.1109/MICRO.2007.16 (2007).
5) Gem5: Main Page - gem5, Gem5 (online), available from ⟨`http://gem5.org/Main_Page`⟩ (accessed 2011-06-21).
6) Howard, J., Dighe, S., Hoskote, Y., Vangal, S., Finan, D., Ruhl, G., Jenkins, D., Wilson, H., Borkar, N., Schrom, G. et al.: A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, IEEE, pp.108–109 (2010).
7) Intel: Intel Unveils New Product Plans for High-Performance Computing, Intel, Corp (online), available from ⟨http://www.intel.com/pressroom/archive/releases/2010/20100531comp.htm⟩ (accessed 2011-06-21).
8) Krasnov, A., Schultz, A., Wawrzynek, J., Gibeling, G. and Droz, P.: RAMP Blue: A message-passing manycore system in FPGAs, *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, IEEE, pp.54–61 (2007).
9) Lee, V., Kim, C., Chhugani, J., Deisher, M., Kim, D., Nguyen, A., Satish, N., Smelyanskiy, M., Chennupaty, S., Hammarlund, P. et al.: Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU, *ACM SIGARCH Computer Architecture News*, Vol. 38, No. 3, ACM, pp. 451–460 (2010).
10) Magnusson, P., Christensson, M., Eskilson, J., Forsgren, D., Hållberg, G., H "ogberg, J., Larsson, F., Moestedt, A. and Werner, B.: Simics: A full system simulation platform, *COMPUTER,*, pp.50–58 (2002).
11) Martin, M. M.K., Sorin, D.J., Beckmann, B.M., Marty, M.R., Xu, M., Alameldeen, A.R., Moore, K.E., Hill, M.D. and Wood, D.A.: Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset, *SIGARCH Comput. Archit. News*, Vol.33, pp.92–99 (online), DOI:http://doi.acm.org/10.1145/1105734.1105747 (2005).
12) Moreira, J., Brutman, M., Castaños, J., Engelsiepen, T., Giampapa, M., Gooding, T., Haskin, R., Inglett, T., Lieber, D., McCarthy, P., Mundy, M., Parker, J. and Wallenfelt, B.: Designing a highly-scalable operating system: the Blue Gene/L story, *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, SC '06, New York, NY, USA, ACM, (online), DOI:http://doi.acm.org/10.1145/1188455.1188578 (2006).
13) Pellauer, M., Adler, M., Kinsy, M., Parashar, A. and Emer, J.: HAsim: FPGA-based high-detail multicore simulation using time-division multiplexing, *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, IEEE, pp.406–417 (2011).
14) Seiler, L., Carmean, D., Sprangle, E., Forsyth, T., Abrash, M., Dubey, P., Junkins, S., Lake, A., Sugerman, J., Cavin, R. et al.: Larrabee: a many-core x86 architecture for visual computing, *ACM Transactions on Graphics (TOG)*, Vol. 27, No. 3, pp. 1–15 (2008).
15) Shmueli, E., Almasi, G., Brunheroto, J., Castanos, J., Dozsa, G., Kumar, S. and Lieber, D.: Evaluating the effect of replacing CNK with linux on the compute-nodes of blue gene/l, *Proceedings of the 22nd annual international conference on Supercomputing*, ICS '08, New York, NY, USA, ACM, pp. 165–174 (online), DOI:http://doi.acm.org/10.1145/1375527.1375554 (2008).
16) Singer-Villalobos, F.: TACC Collaborates with Intel to Accelerate Open Science Research Using Intel®MIC Processor Line, Texas Advanced Computing Center (online), available from ⟨`http://www.tacc.utexas.edu/news/press-releases/intel-collaboration`⟩ (accessed 2011-06-21).
17) Tan, Z., Waterman, A., Avizienis, R., Lee, Y., Cook, H., Patterson, D. and Asanović, K.: RAMP gold: an FPGA-based architecture simulator for multiprocessors, *Proceedings of the 47th Design Automation Conference*, DAC '10, New York, NY, USA, ACM, pp. 463–468 (online), DOI:http://doi.acm.org/10.1145/1837274.1837390 (2010).
18) Wawrzynek, J., Patterson, D., Oskin, M., Lu, S., Kozyrakis, C., Hoe, J., Chiou, D. and Asanovic, K.: RAMP: Research accelerator for multiple processors, *Micro, IEEE*, Vol.27, No.2, pp.46–57 (2007).
19) Xilinx: Virtex-6 FPGA ML605 Evaluation Kit, Xilinx, Inc (online), available from ⟨http://www.xilinx.com/products/boards-and-kits/EK-V6-ML605-G.htm⟩ (accessed 2011-06-21).
20) Yourst, M.: PTLsim: A cycle accurate full system x86-64 microarchitectural simulator, *Performance Analysis of Systems & Software, 2007. ISPASS 2007. IEEE*

International Symposium on, IEEE, pp.23–34 (2007).
21) 植原　昂，佐藤真平，佐野伸太郎，吉瀬謙二：メニーコアプロセッサの研究・教育を支援する実用的な基盤環境 M-Core，技術報告 8，東京工業大学大学院情報理工学研究科, 東京工業大学大学院情報理工学研究科, 東京工業大学工学部情報工学科, 東京工業大学大学院情報理工学研究科 (2010).
22) 高前田伸也，佐藤真平，藤枝直輝，三好健文，吉瀬謙二：メニーコアアーキテクチャの HW 評価環境 ScalableCore システム，情報処理学会論文誌コンピューティングシステム（ACS），Vol.4, No.1, pp.24–42 (2011).
23) 下沢　拓，石川　裕，堀敦　史，並木美太郎，辻田祐一：メニーコア向けシステムソフトウェア開発のための実行環境の設計と実装，情報処理学会研究報告 (2011-OS-118, SWoPP2011)，No.1 (2011).