

化学計算専用計算機のプラットフォームシステム

佐々木 徹^{a*}, 村上 和彰^b

^a (株) アプリアリ・マイクロシステムズ, 〒 212-0054 川崎市幸区小倉 308-10 かわさき新産業創造センター 236

^b 九州大学大学院システム情報科学研究院, 〒 812-8581 福岡市東区箱崎 6-10-1

*e-mail: sasaki@a-priori.co.jp

(Received: July 4, 2005; Accepted for publication: November 7, 2005; Published on Web: December 15, 2005)

専用計算機を開発する際には、システム開発のターンアラウンドが大きな問題となる。一般にハードウェアの開発がプロジェクトの進捗を律速するが、特に専用 LSI を作成する場合 LSI が完成してから専用ボードを作成し、その後デバイスドライバ、通信ミドルウェア等のシステムソフトウェアの作成、アプリケーションプログラムの移植を行うため、システムが完成した時には性能・機能などの面で既に陳腐化していることも少なくない。

そこで、専用計算機の開発ターンアラウンドを改善するために、システム開発を二段階に分けて行い、専用 LSI とソフトウェアを並行して開発した。第一段階では専用 LSI の開発と並行して汎用 CPU に専用 LSI の機能をエミュレーションさせるプロトタイプシステムを用いてソフトウェア開発を行い、第二段階では専用 LSI を実際にシステムに組み込むという方式を採った。また、複数のアプリケーションに対応するため、プラットフォームシステムはアクセラレータボード上に同一の CPU を搭載し、システムソフトウェアも共通化した。その結果、GAMESS (MO)、ABINIT-MP (MO)、Car-Parrinello (DFT)、DV-X (DFT)、GSMAC-FEM (CFD) など多くのアプリケーションに対する専用機を比較的短期間に開発することができた。

キーワード：並列処理, 専用計算機

1 はじめに

MO などの大規模計算を行える低コストかつパーソナルユース可能なシステムを実現するには、近似法やアルゴリズムの開発に加え、計算機の大幅な性能向上が必要である。計算機の性能を飛躍的に向上させる方法のひとつとして、MOE[1] や GRAPE[2] のように処理に特化したハードウェアをシステムに組み込む専用計算機を開発する方法がある。専用計算機を開発する場合、ハードウェアの作成を伴うため、ハードウェア開発が開発のターンアラウンドを律速してしまうことも多く、システムが完成したときには既に陳腐化していることも珍しくない。この傾向は専用 LSI の作成を伴う場合、特に顕著である。

EHPC プロジェクトでは、専用計算機の開発ターン

アラウンドを改善するため、ベースとなるアーキテクチャを規定し、単に専用 LSI を載せ替えるだけで専用計算機を作成できるフレームを用意した。これにより、デバイスドライバ、通信ライブラリといったシステムソフトウェアやハードウェアの基本部分が共通化されるため、著しく開発効率を向上させることができた。

また、専用計算機の開発ではハードウェアの開発に重点が置かれるが、実際にはハードウェアの開発だけでなく、ソフトウェアを移植するコストも決して小さいものではない。ソフトウェアを移植する際には、まずシステムプログラムを立ち上げてから、アプリケーションプログラムを移植するという手順を踏むことになるが、ハードウェアの開発（特に専用 LSI の開発）とソフトウェアの開発が並行して行えるよう、専用 LSI の機能を汎用 CPU によってエミュレーション

するプロトタイプ機を製作し、開発のターンアラウンドの改善を試みた。これらの方策を講ずることにより、GAMESS (MO) [3]、ABINIT-MP (Fragment MO) [4]、Car-Parrinello (DFT) [5]、DV-X (DFT) [6]、GSMAC-FEM (CFD) [7] などのアプリケーションを比較的短期間で稼働させることができた。

本稿では、EHPCプラットフォームシステムの概要とアプリケーションプログラムの移植例を簡単に紹介する。

2 プラットフォームシステム

2.1 アーキテクチャの概要

EHPCプロジェクトにおいて、プラットフォームシステムはすべての専用システムの骨格となり、専用LSIを組み込むためのハードウェアの枠組み、および専用LSIを開発と並行してシステムプログラム開発、アプリケーションプログラム移植のため環境を提供する。Figure 1 にプラットフォームシステムの外観を示す。写真はプロトタイプシステムのものであるが、CompactPCIの筐体を実装しているため、専用LSIを用いたシステムの場合もほぼ外観は同様である。

プラットフォームシステムの内部構成はFigure 2に示すように階層構造となっている。第一層はPCクラスタで、これは既存のソフトウェアがMPIなどを用いて既に並列化されていることも多く、その場合には粒度の大きい並列化が為されている部分をそのまま活用することを意図したものである。また、新規にソフトウェアを作成する場合においても、MPI等の汎用並列化ライブラリが使用できると開発効率が高い。第二層

は、PCクラスタを構成するPCをホストとし、CompactPCIバスで接続されたEHPCボードのコントローラCPUである。第三層は専用LSIを用いたリーフの演算ノードである。階層化されたヘテロジニアスなマルチプロセッサシステムとなっている。

このような階層構造のアーキテクチャの場合、階層を下る毎に処理が細分化され、並列粒度も小さくなる場合には有効であると言われているが、リーフの演算ノード間で頻繁にデータを交換する必要があるようなアプリケーションには不利であることが予想される。しかし、MOやDV-XのようにFock行列を生成するようなアプリケーションではリーフの演算ノードは独立に計算が実行できるため、ツリー構造が適している。また、Car-Parrinelloのように三次元FFTにより平面波基底を実空間データに変換する場合には、FFTそのものを分割するのではなく、バンド並列、すなわち個々演算ノードがバンドひとつ分のFFTをすべて実行してしまえば、やはり演算ノード間のデータ転送は必要ない。

一方、GSMAC-FEMのような有限要素法では領域分割して並列化するのが一般的に行われているが、その場合には領域境界上でのデータ交換が避けられないため、演算ノード間でのデータ転送が性能向上に寄与すると考えられる。しかしながら、開発効率の観点では、専用LSIは構造を単純化し極力演算に特化して実装すべきであり、演算ノード間の通信機能まで搭載するのは好ましくない。そこで、ボード上の汎用CPUに通信ハードウェアを付加してOSの管理化でファームウェアが通信制御を行うという方法が考えられるが、本研究ではそこまでの検討は時間の関係で行えなかった。



Figure 1. EHPC Platform System

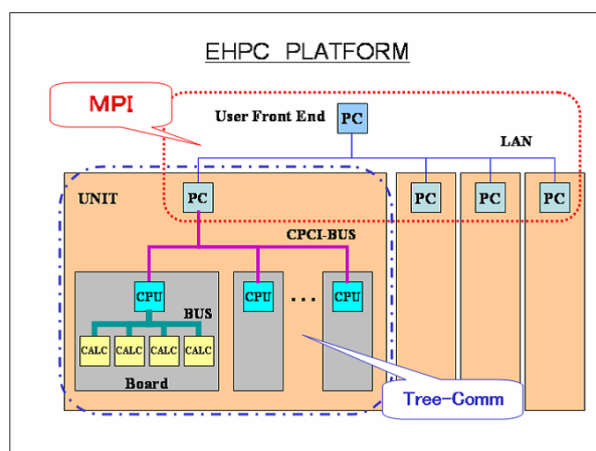


Figure 2. Block Diagram of the Platform

Figure 3 に EHPC ボードの一般的な構成を示す。図中の斜線部に所望の演算ノードを置くことにより、

1. 汎用 CPU を用いた半専用システム
2. FPGA のようなリコンフィギャラブルデバイスを用いた専用システム
3. カスタムチップを用いた専用システム

などが構成できる。

図中左下に書かれた CPU はボード全体のコントローラとして機能する。一般に、割込み源が複数存在する場合にはリアルタイム OS を搭載したほうが、プログラミングが容易になることが多いが、コントローラとなる CPU はホスト PC との通信、個々の演算ノードの制御、さらには DMA コントローラ、デバグ用のシリアル通信、タイマ等の多くの割込み源を抱えている。そこで CPU 上にリアルタイム OS として μ ITRON を搭載し、プログラミング環境を改善した。それぞれのボードにおいてコントローラ CPU に同一のものを使用すれば、CPU の周辺回路は一度開発したものがほとんどそのまま再利用可能であり、専用 LSI を駆動するドライバプログラムを差し替えるだけでシステムソフトウェアの全体的な整合性も維持される。

2.2 通信ライブラリ

Figure 4 に EHPC プラットフォームの通信ライブラリである Tree-Comm の概要を示す。Tree-Comm の I/F は

```
tcom_initialize();
tcom_send();
tcom_recv();
tcom_broadcast();
etc.
```

のように MPI ライクな I/F となっているが、ハードウェアがヘテロジーニヤスな階層構造であるから、ハードウェアの構造に依存して通信方式も階層構造を採り、通信方向がひとつ上の階層とひとつ下の階層に限定されている。前述したように各演算ノード間が独立に演算可能であるようなアプリケーションに対してはこれで充分である。また、PCI バスはバス構造ではあっても、ホスト対 IO のデータ転送を念頭に置いて設計されたものであるため、ホスト PC と EHPC ボードとの関係を考えても無理のない実装となっている。

Figure 5 は EHPC ボード上の PCI ブリッジチップに接続された 64MB の共有メモリを利用した非同期通信路 BrdShm の概要を示す。Tree-Comm は基本的に同期通信路であり、ホストとのデータ交換は一旦ボード上の主記憶を介して行われる。そのため、大規模なデータを頻繁に転送する場合には CPU の主記憶を介さずに直接専用 LSI と共有メモリとの間でデータの受け渡しが行える BrdShm を使用すると有効である。状況にもよるが、ホストから専用 LSI へのデータ転送は BrdShm のほうが Tree-Comm より概ね 40 ~ 60 % 程度高速である。タイトに同期を取る場合には Tree-Comm を使い、大規模なデータのやり取りには BrdShm を使う、など適時両者を使いわけて用いている。

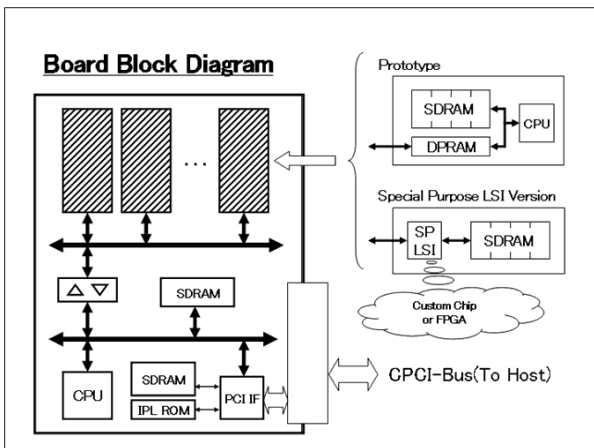


Figure 3. Block Diagram of EHPC Board

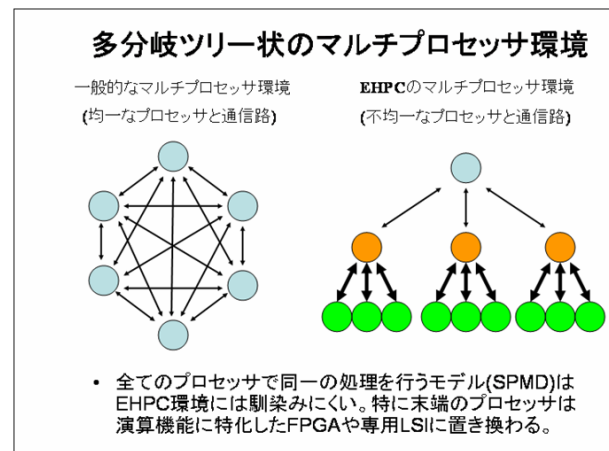


Figure 4. Tree-Comm

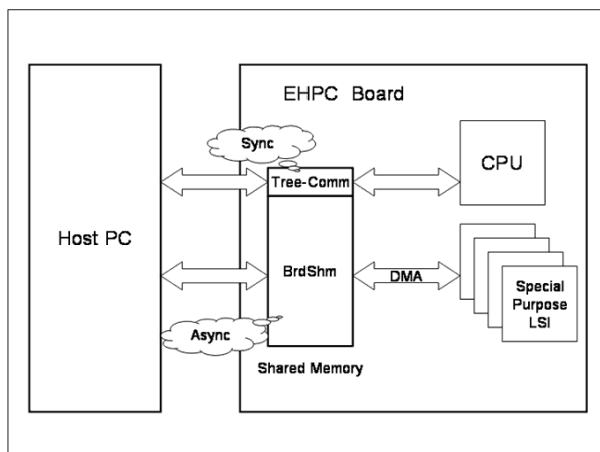


Figure 5. Asynchronous Communication

2.3 プロトタイプシステム

Figure 6 にプロトタイプボードを示す。ボード上には 4 個の汎用 CPU を搭載している。

諸元は、CPU (ルネッサステクノロジ製 SH4@200MHz) × 4、メモリは各 CPU に SDRAM64MB 搭載、CPU 間の通信バッファ DPRAM128KB、ルネッサステクノロジ製の SH4 ファミリーチップの PCI ブリッジに共有メモリとして SDRAM64MB を接続している。PCI バスは 32bit33MHz で動作し、ホスト PC と EHPC ボード間のデータ転送レートは約 70MB/sec である。なお、PCIブリッジと直接接続している CPU 周辺回路は各専用ボードにも再利用している。プロトタイプシステムではリーフの演算ノードにも汎用 CPU を使用しているため、演算ノードにもリアルタイム OS (μITRON)

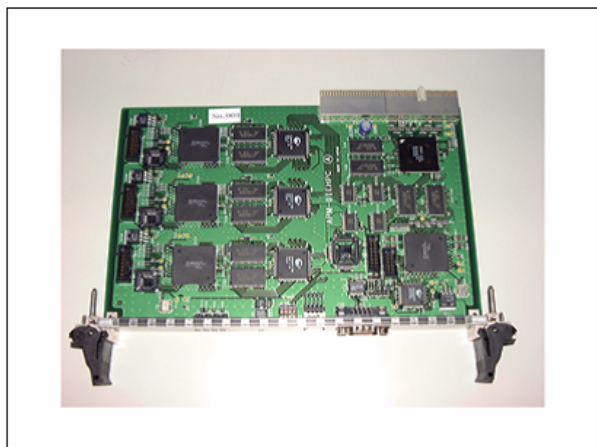


Figure 6. Prototype Board

を搭載し、Tree-Comm を実装した。

プロトタイプシステムは主にシステムプログラムの開発、および専用 LSI を組み込むための準備、例えば API や並列化方法の実証的な検討などに使用したが、DV-X の場合はプロトタイプシステムを専用機として位置付けた。DV-X では Fock 行列を生成する部分が処理時間の 90% 以上を占めており、MO と同様に Fock 行列生成部がアクセラレーションの対象となり得るが、MO とは異なり電子間の相互作用を電子密度分布から求めるため、明確なホットスポットを持たず、専用 LSI ではなく汎用 CPU 上のプログラムのチューニングによる専用システム化が妥当であると判断されるためである。解法としての精度を考慮すると、行列要素生成部は単精度演算でも十分な精度が得られるため、SH4 のような組み込み用途の CPU でも十分な性能が得ることができる [8]。

2.4 専用システム

四中心二電子積分計算専用 LSI (ERIC チップ) [10] を組み込んだボード (ERIC ボード) を Figure 7 に、FPGA デバイスを組み込んだボード (EHPC-FPGA ボード) を Figure 8 にそれぞれ示す。FPGA は内部の論理を確定させない限り機能が固定されないため厳密な意味での専用機とは言えず、半専用機という表現のほうが適切であるようにも思われるが、ここでは各ボードを簡単に紹介する。ERIC チップおよび三次元 FFT ロジックについてはそれぞれ紹介されるので、そちらを参照して頂きたい。

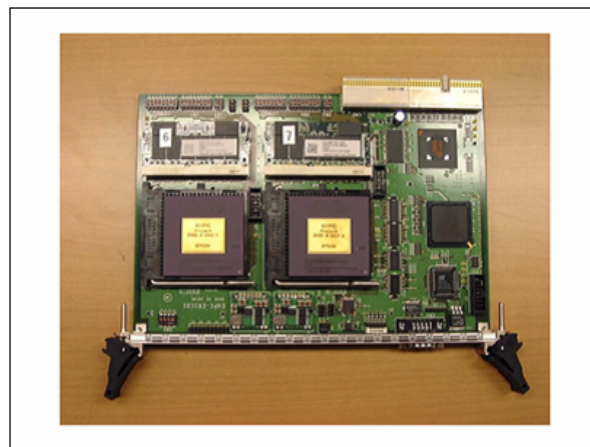


Figure 7. ERIC Board

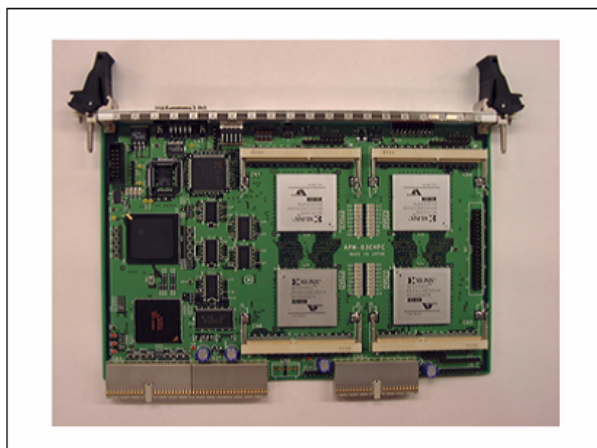


Figure 8. EHPC FPGA Board

ERIC ボードは演算ノードとして ERIC チップを 2 個搭載しており、FPGA ボードは FPGA を 4 個搭載している。どちらもコントローラ CPU として SH4 を搭載し、通信ライブラリもプロトタイプ機と同じものが使用できる。そのため、短期間でシステムとして稼働させることができた。FPGA ボードを用いたシステム上で三次元 FFT と GSMAC-FEM が稼働している。

3 アプリケーションプログラムの実装例

ここではアプリケーションプログラムの実装例としてフラグメント MO と三次元 FFT を紹介する。

3.1 フラグメント MO

フラグメント MO は、大規模な分子をフラグメントに分割することによって、問題サイズを小さくして解く効果的な手法であるが、各フラグメントが PC クラスタ上で各 PC に割り当てられて実行されている [9]。すなわち、そもそも解法として階層化されているため、EHPC システムのシステムデバグとしても有効であった。Figure 9 にプロトタイプシステム上でのソフトウェア構成を示す。

プログラムの移植に際し、予め二電子積分のコアルーチン（新小原アルゴリズムの構成要素：誤差関数テーブル管理、初期積分、漸化計算、Fock 行列生成）を ABINIT-MP に組み込んで API の妥当性を評価しておき、実際に EHPC システムに実装する際には、図の

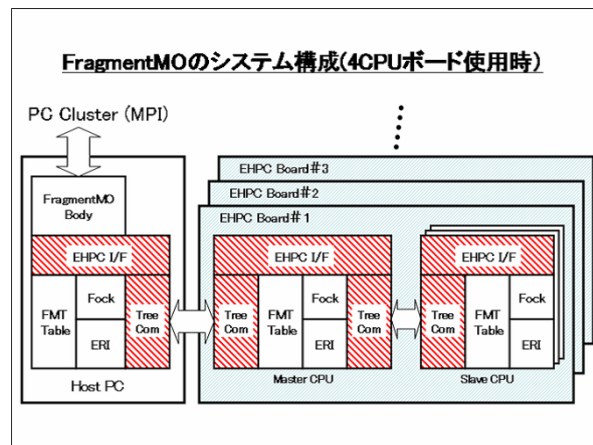


Figure 9. Implementation of Fragment MO

斜線部の通信ルーチンを挿入するだけで簡単に移植することができた。

3.2 三次元 FFT

Car-Parrinello 法に使用することを念頭においた三次元 FFT 専用機 [11] は、FPGA デバイスにハードウェアロジックを実装することにより実現した。ソフトウェアから見ると専用ロジックの実装形態が ASIC であっても FPGA 上のロジックであってもほとんど問題ではなく、やはり API の確定とその評価方法が重要なポイントとなる。

当初は Car-Parrinello 法向け三次元 FFT の API は汎用のソフトウェアライブラリと同様に直接 FFT の入出力となる三次元配列データをインタフェースとして開発を進めた。しかしながら、Car-Parrinello 法の場合、三次元配列に展開する前のエネルギーカットオフされた波動関数を入出力したほうがホスト-ボード間のデータ転送量が遥かに小さいため API を変更した。三次元 FFT を施す箇所を具体的に分析し、最終的には Figure 10 のように API を

1. 電荷密度算出
 - 入力 = 波動関数
 - 出力 = 電荷密度の空間分布
2. 勾配ベクトル生成
 - 入力 = 波動関数
 - 出力 = 実空間ポテンシャルを乗じて、波数空間に戻したデータ

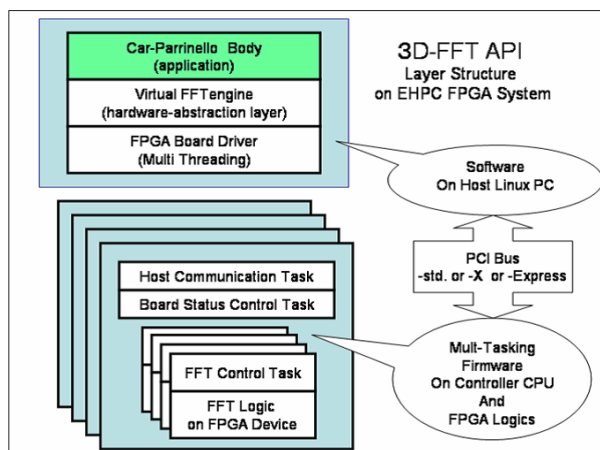


Figure 10. Implementation of 3D-FFT

と規定し直し、それに合わせて FFT ロジックにも機能追加を行った。従って、この例は開発事例としてはあまり適切ではないが、専用計算機開発のひとつの重要なポイントを示唆している。汎用のパッケージライブラリの切り口をそのまま専用ロジックのインタフェースとしても専用計算機の切り口としては必ずしも適当ではないことを示しており、興味深い事例である。

4 考察とまとめ

専用計算機を開発する場合、ハードウェア開発が全体の開発スケジュールを律速してしまうことが少なくない。そこで、我々はシステムプログラムや回路の再利用およびソフトウェアとハードウェア（特に専用LSI）の並行開発を可能とするため、システムアーキテクチャの標準化とプロトタイプシステムの作成を行った。その結果、専用LSIのリリースに先立ってアプリケーションプログラムの準備ができていたため、ハードウェアが完成してから1～3ヶ月という比較的短期間でシステムを立ち上げることができた。

また、階層構造のアーキテクチャはMOのように演算ノード間でデータ転送の必要がないものには非常に有効であり、構造が単純なため、専用LSIをシステムに組み込むのも容易であった。

システムバスとしてPCIバスではなく産業用規格であるCompactPCIバスを用いたが、CompactPCIシステムはPC互換カードを含めて8枚まで1筐体に装着できるので、1システムに専用LSIが多数搭載できて



Figure 11. System Debugging

専用計算機のシステム評価に好都合である。また、産業用であるから、(1)筐体やボードの機械的な強度が大きい、(2)ボードの挿抜回数の制限が緩い、さらに(3)ボードへの電力供給/放熱対策の点で自由度が大きい、等の理由で専用計算機の開発には適していた。

科学技術計算専用計算機を開発する際には、開発の重点がハードウェアに置かれるのは当然としても、筆者らはソフトウェア開発もハードウェア開発と同程度の期間を要すると考えており、その意味ではプロトタイプ機を用いたソフトウェアの開発環境を有効に活用することができた。開発期間を短縮するには、ハードウェアおよびソフトウェアの評価環境、特に開発初期段階の評価環境の構築が現実問題として非常に重要なアイテムである。

本研究の一部は科学技術振興調整費 総合研究「科学技術計算専用ロジック組込み型プラットフォーム・アーキテクチャに関する研究」(代表 村上和彰 九州大学教授)によるものである。

参考文献

- [1] Koji Hashimoto et al., MOE: a special purpose parallel computer for high-speed, large-scale molecular orbital calculation, *Supercomputing 99*, IEEE & ACM, Seattle (1999).
- [2] <http://grape.astron.s.u-tokyo.ac.jp/grape/>

- [3] <http://www.msg.ameslab.gov/GAMESS/GAMESS.html>
- [4] Kazuo Kitaura et al., Fragment molecular orbital method: an approximate computational method for large molecules, *Chem. Phys. Lett.*, **313**, 701 (1999).
- [5] Richard Car and Mark Parrinello, Unified Approach for Molecular Dynamics and Density-Functional Theory, *Phys. Rev. Lett.*, **55**, 2471 (1985).
- [6] 足立裕彦, 量子材料化学入門 - DV-Xa 法からのアプローチ -, 三共出版 (1991).
- [7] 棚橋隆彦, 流れの有限要素法解析 I/II, 朝倉書店.
- [8] 佐々木徹, 長嶋雲兵, 第一原理 DVX 計算専用計算機 EHPC-DVX の開発, *J. Comp. Chem. Jpn.*, **2**, 111-118 (2003).
- [9] 稲富雄一ら, フラグメント分子軌道法計算プログラム ABINIT-MP における二電子積分ルーチン的高速化ならびに並列化と性能評価, *IPSI Symposium Series*, **2001(6)**, 93-94 (2001).
- [10] 原田宗幸ら, 二電子積分計算専用プロセッサ・アーキテクチャの開発, 情報処理学会論文誌: ハイパフォーマンスコンピューティングシステム, **44**, No. SIG 1 (HPS6), 1-9 (2003).
- [11] 佐々木徹, 溝口大介, 長嶋雲兵, Car-Parrinello 計算向け三次元 FFT ロジックの開発, 情報処理学会論文誌, **45** No.SIG 11(ACS 7), 313-320 (2004).
- [12] 青木すみえ, 荒木健悟, 溝口大介, 石橋政一, 佐々木徹, 棚橋隆彦, FPGA による GSMAC-FEM 専用計算機の実装と評価, 第 17 回数値流体力学シンポジウム論文集, **C11-5** (2003).

A Platform System of Special Purpose Computers for Various Kinds of Chemical Simulations

Tohru SASAKI^{a*} and Kazuaki MURAKAMI^b

^aA-Priori Microsystems, Inc.

236 KBIC 308-10 Ogura, Saiwai-ku, Kawasaki, Kanagawa 212-0054, Japan

^bGraduate school of Information Sciences and Electrical Engineering, Kyushu University

6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan

**e-mail: sasaki@a-priori.co.jp*

In the development of special purpose computers for scientific calculation with special hardware, including special purpose LSI, for some heavy load processes (hot spots), the hardware design and system debugging often become serious bottlenecks of the development process. We took a software/hardware co-design approach to improve turnaround time. At first we designed the platform architecture (EHPC platform) as the base system, and we developed a prototype system based on the platform architecture. The prototype has a large number of general purpose CPUs instead of special purpose LSI's and can emulate the functions of the LSI with the firmware on the CPU. In the next step we developed the system with special purpose LSI. Then we have developed some special purpose computers for Molecular Orbital Calculation (GAMESS, Fragment MO), Density Functional Theory (Car-Parrinello, DV-Xalpha), Computational Fluid Dynamics (GSMAC-FEM). In this paper, we introduce the EHPC architecture and show implementations of some application programs on the architecture.

Keywords: Parallel processing, Special purpose computer