

# Car-Parrinello 法のための リコンフィギャラブル三次元 FFT 専用プロセッサ

佐々木 徹<sup>a</sup>, 別役 潔<sup>b</sup>, 樋口 高年<sup>b</sup>, 長嶋 雲兵<sup>c\*</sup>

<sup>a</sup> (株) アプリオリ・マイクロシステムズ,

〒 212-0054 神奈川県川崎市幸区小倉 308-10 かわさき新産業創造センター 236

<sup>b</sup> みずほ情報総研(株), 〒 101-8443 東京都千代田区神田錦町 2-3

<sup>c</sup> 産業技術総合研究所計算科学技術センター, 〒 305-8568 茨城県つくば市梅園 1-1-1

\*e-mail: u.nagashima@aist.go.jp

(Received: July 4, 2005; Accepted for publication: September 13, 2005; Published on Web: December 3, 2005)

Car-Parrinello (CP) 法による第一原理計算における計算時間の大部分は電子状態ベクトルの三次元 FFT 演算に費やされる。そこで、FFT を専用ロジック化し、FPGA を 4 個用いたボード上に実装した。FFT 単体性能を実測したところ、2.4GHz で動作する Xeon を搭載した PC と比較して、ボードあたり 10 倍程度の性能が実測された。消費電力まで考慮すると 50 倍の性能比となる。このボードを CP 計算に利用することで、ボード 2 枚 (FPGA8 個相当) あたり、前述の Xeon 1CPU における CP 計算と比較して 10 倍の高速化が実現される。CP コードをさらに高速化するため、FFT と Gram-Schmidt 直交化を動的再構成可能な単一の FPGA において実行する手法を提案する。

キーワード: Car-Parrinello 法, 第一原理計算, 平面波展開, FFT, Gram-Schmidt 直交化, 専用計算機, リコンフィギャラブル, FPGA

## 1 はじめに

固体や表面の第一原理計算手法として、最も良く利用される手法に、結晶の周期性を考慮した平面波展開をもちいる Car-Parrinello 法 (CP 法) [1] がある。CP 法が登場する前の従来法、すなわち、ハミルトニアン行列を直接対角化する手法は、平面波基底の数を  $N$  とし、 $N$  の 3 乗の計算負荷を要していた。CP 法は固有値問題を繰り返し法による最小値問題に置き換えると同時に FFT を用いることで、固有状態の数を  $M$  とし、計算負荷を  $MN \ln N$  まで削減することに成功した。これにより、第一原理分子動力学 (MD) への道が開かれた。

しかし、単体の PC や WS では、メモリや計算時間の制約により、第一原理 MD によって扱える原子数は 100 原子程度に限られる。したがって、最近のナノテ

クノロジー研究や材料開発現場においてしばしば必要とされる 1,000 原子を超える大規模計算への要望に応えられる状況にはない。

一方、科学技術計算分野において、汎用 CPU に代わり、FPGA などのリコンフィギャラブルデバイスが注目されてきており、信号処理や画像処理、流体解析 [2]、X 線結晶構造解析などにも使用され始めている。そのため、ソフトウェアによる高速化の研究 [4-7] も盛んに行われており、リコンフィギャラブルデバイス上における FFT の高速化手法の研究 [8] も行われている。

そこで、われわれは、CP 法において計算負荷の高い箇所を FPGA へオフロードすることにより、大規模な第一原理計算を、個人で占有可能なハードウェアにおいて実現するための研究に着手した。ここに、個人で占有可能とは、低消費電力、高コストパフォーマンス

ス、デスクサイドに収まるサイズを想定している。なお、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シミュレータに関する研究」において為されたものである。以下では、左記プロジェクトを EHPC プロジェクトと記す。

## 2 CP法

Figure 1 に本稿が高速化の対象としている CP 法の処理フローの概略を示す [2]。非局所擬ポテンシャルの演算を除くと、CP 法の計算処理時間の大部分は、固有状態の三次元 FFT と固有状態間の Gram-Schmidt(GS) 直交化によって占められている。

われわれはオブジェクト指向設計と C++ 言語を利用し、CP 法プログラムを新規に開発した。簡単のため、CP の原論文と同様に擬ポテンシャルを局所的とした [1, 2]。

われわれの CP コードを、原子数を変えて逐次実行した際のベンチマーク結果を Table 1 に示す。バルクシリコンを例題として、最急降下法の 1 ステップについて計測した。ここで計測した 200 原子以下の計算規模では、処理時間の 90 % 弱は波動関数の 3 次元 FFT に費やされる。したがって、CP 法を専用ロジック化により高速化するには、三次元 FFT の高速化が第一の課題となる。

また、一般に科学技術計算で行う FFT は多次元であるが故にデータサイズが極めて大きいという特徴を持っており、CP 法の場合でも実用規模として、例えば 1000 原子程度の系を計算するには、 $256^3$  タップ (浮動小数点倍精度複素数では 256MB) の計算規模とな

る [9]。

## 3 三次元 FFT 専用ロジック

### 3.1 三次元 FFT

Figure 2 に三次元 FFT の実行方法を示す。素直に計算するには、スキャンする方向を順次替えて一次元 FFT を行えば良いが、メモリ上の連続並びと異なる方向にスキャンする場合には、隣接データ間のストライドが大きくなってしまいうため、ソフトウェアで高速に実行するには、スキャン方向を替える毎に配列を転置するなどの工夫が必要となる [7]。

本稿で提案する三次元 FFT ロジックは、三次元配列データを転置せずに高速にメモリアクセスを行うことができる。FFT のバタフライ演算を実行する Butterfly コアはあまり工夫しても、それに見合うだけのメモリバンド幅が実現できなければあまり大きな意味を持たない。そこで我々は専用ロジックが解決すべき問題点をメモリアクセスに絞り、FFT の計算アルゴリズムについてはできるだけ平易に実装できるものを選び、Figure 3 に示す Cooley-Tukey アルゴリズムを基に Butterfly コアを作成した。解決すべき最大の問題としたメモリアクセスについては、Figure 4 のように擬似的なバースト転送をそれぞれのスキャン方向について行うことによって、XYZ 各軸に関して等方的なアクセスを行い、これを Figure 5 のようにダブルバッファリングして、データ転送時間を隠蔽することにより解決した。これらをひとつの FPGA に実装すると、ロジックの全体構成は Figure 6 のようになる。三次元 FFT ロジックについての詳細は文献 [11] を参照されたい。

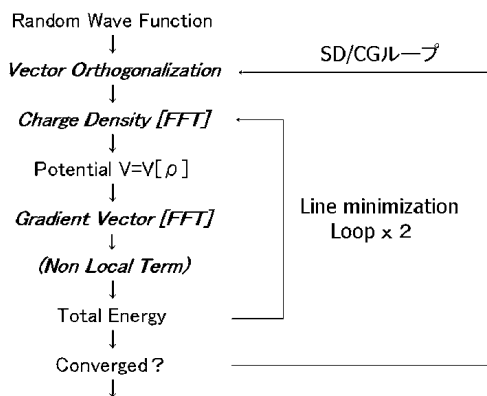


Figure 1. Flow diagram of self-consistent field calculation in the CP method

Table 1. Percentage of 3D FFT in the CP calculation

Count of Atoms	8	64	216
Count of Eigenstates	16	128	432
Count of Plane-Waves	123	739	2517
Count of FFT Taps	$16^3$	$32^3$	$64^3$
Wall-clock time for one SD iteration[sec]	0.37	73	2724
3D FFT Percentage	60.6	84.8	88.0
Orthogonalization Percentage	1.7	5.6	5.9

Ex. Array Size:  $256 \times 256 \times 256$ , Device: 256MB SO-DIMM

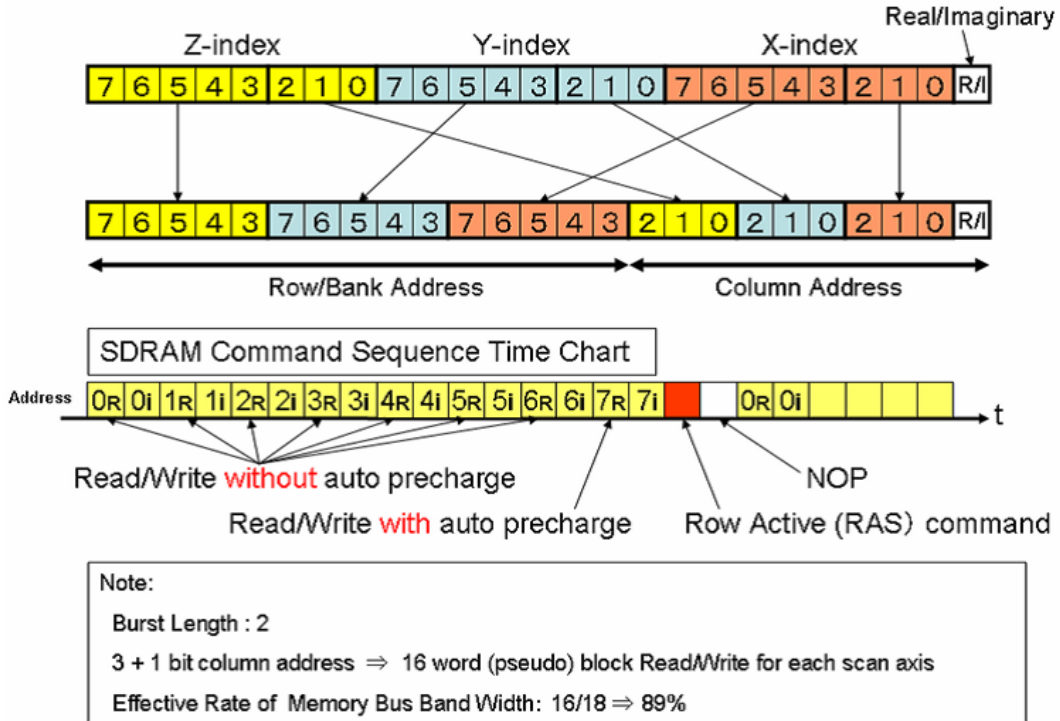


Figure 2. Pseudo Burst Transaction for 3D Array

Cooley-Tukey Algorithm

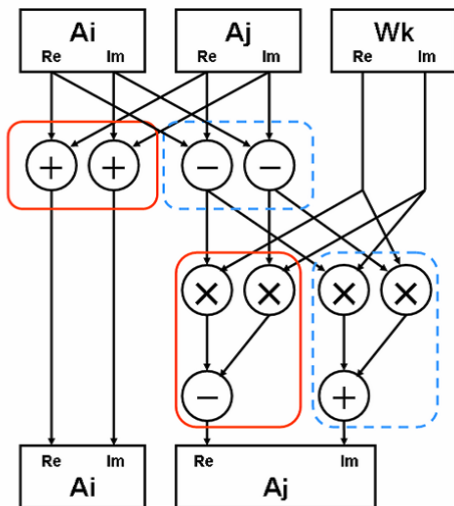


Figure 3. Butterfly Core

X-Scan 1D-FFT

$\Rightarrow$  Y-Scan 1D-FFT

$\Rightarrow$  Z-Scan 1D-FFT

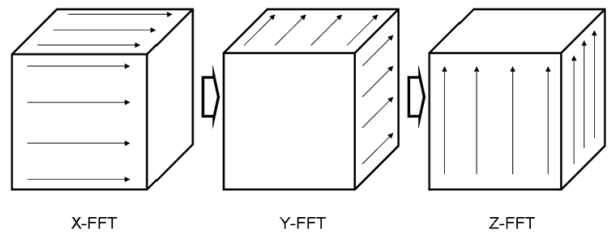


Figure 4. Scan directions of 3D FFT

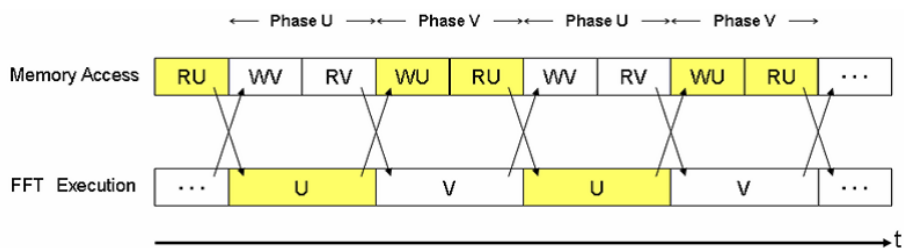


Figure 5. Double buffering

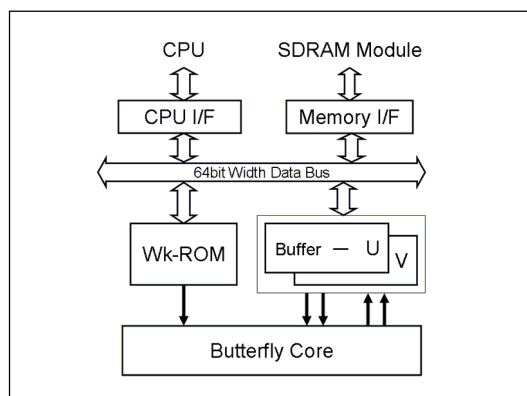


Figure 6. FFT logic

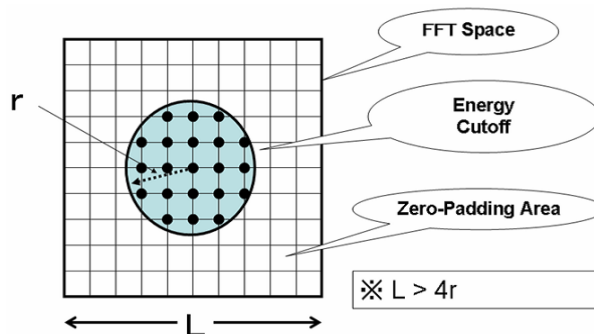


Figure 7. FFT area for the CP methods

### 3.2 ゼロパディング領域の取り扱い

CP法に限らず逆変換時の精度保証のため、元のデータよりもかなり大きなデータ領域を用いてFFTを施すことも多い。本来データの定義されない領域はゼロパディング等により、計算結果に与える影響の少ないデータが埋められる。

CP法では実空間波動関数を波数空間（周波数空間）上に定義され、Figure 7のようにエネルギーの上限値により高周波成分がカットオフされるが、カットオフ球の直径の倍以上、ゼロパディングして解析領域を拡大する必要がある。これは、勾配ベクタ計算において

逆フーリエ変換を施し、波動関数を実空間上の波動関数に変換した後、その点のポテンシャルを乗じて再度フーリエ変換して波数空間に戻すため、波数空間に戻したときの計算精度を確保することが理由である。

ゼロパディング領域では、メモリからバッファにフェッチした一本のスキャンラインがすべてゼロであることが多い。一本のスキャンラインのデータがすべてゼロであれば、FFTを施した結果もゼロなので、演算を行う必要がない。従って、ゼロパディング領域であることを検出してそのスキャンラインの演算を省略することにより演算時間の短縮を行った。

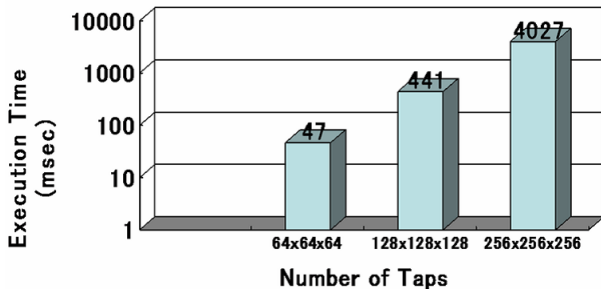


Figure 8. Performance of 3D FFT on the FPGA Board

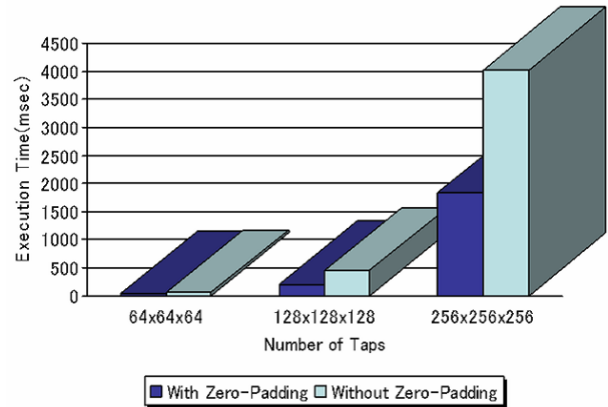


Figure 9. Comparison of 3D FFT performance with or without zero Padding

### 3.3 CP法固有の処理

CP法では、単にIFFTやFFTを専用ロジックで実行するだけでは、システム上のメリットが小さいため、電荷密度の計算配ベクタの計算も専用ロジックで行うのが好ましい。

今回は暫定的にFPGAロジックの内部バスにEXTRAロジックとして接続してシステム評価を行っているが、本来Butterflyコアの演算器の接続を変形して実行させるか、あるいはリコンフィギュラブルなスキームを用いるべきであると考えている。これについては今後の検討課題である。

## 4 評価に使用したシステム

### 4.1 システム構成

FPGAボードを用いたシステム全体はPCクラスタから構成され、CompactPCIバスを介して個々のPCにFPGAボードが接続されている。EHPCプロジェクトにおいて、ロジックの評価用にFPGAボードを作成した。本ボード上では、FFT以外にも、GSMAC-FEMに基づく流体解析専用ロジックなども稼働している [14]。

コントロール・ノードとして組込み制御用のCPU (SH4) と、演算ノードとして300万ゲート相当のFPGAを4個搭載している。個々FPGAには512MB

のSDRAMモジュールが接続できる [13]。

### 4.2 API

「FFTエンジン」をタップ数等のパラメータを与えてオブジェクトとして生成し、FFTエンジンに対して波動関数のリストを渡せば電荷密度や勾配ベクタが算出される。「FFTエンジン」は完全に抽象化されており、アプリケーションプログラムはFPGAボードの枚数や搭載されるFPGAデバイスの数などといったハードウェアの情報は一切知る必要なく、アプリケーションプログラムと接続することができる。

## 5 結果

### 5.1 FFT単体性能

Figure 8に三次元FFTの単体性能の実測値を示す。ソフトウェアで実行する場合と異なり、タップ数によらず演算性能は500MFLOPSである。これは、搭載した5個の浮動小数点演算器が動作周波数100MHzで、ほぼ稼働率100%で稼働していることを示している。従って、EHPC-FPGAボードでの測定ではメモリバンド幅がボトルネックになっている、言い換えると懸案となっているメモリアクセス時間が完全に隠蔽された、ということができる。

## 5.2 FFTゼロパディング領域の演算省略

ゼロパディング領域を検出して、FFTを省略した場合の結果を Figure 9 に示す。

なお、ここでのカットオフ半径はメッシュ数に換算すると

64<sup>3</sup>TAP : 13  
 128<sup>3</sup>TAP : 19  
 256<sup>3</sup>TAP : 36

である。カットオフ半径と解析空間の大きさの比率にも依存するが、各辺のタップ数をすべて  $N$ 、有効範囲を  $M$  とすれば

$$\text{演算削減率} = ((M/N)^2 + (M/N) + 1) / 3$$

と見積もられるので、演算量は 1/2 程度には削減される。

## 5.3 汎用 CPU との比較

FPGA ボードの性能を評価するため、世界的に広く使われており、性能にも定評もある FFTW を C P コードに実装した。以下に単独計算、電荷密度計算、勾配計算の比較を示す。FFTW の結果は Intel Xeon 2.4GHz で測定した。FPGA ボードでの計算ではゼロパディング削除機能を使用しており、FFT のデータパディング領域はそれぞれ 27\*3, 39\*3, 73\*3 としている。SCF 計算部の処理時間は式 (i) で評価できる。Si 原子 512 個の系の計算における性能予測をおこなった。Si 原子 512 個 (1024 バンド)、カットオフエネルギー 16Ha の系では、FFTW の場合には式 (ii-a)、FPGA チップ 1 つの場合には (ii-b) のようにそれぞれ評価される。Table

2 に 1 チップ、1 ボード (FPGA4 個搭載) の場合の処理時間を示す。256×256×256 の場合、FFTW よりも個別の処理で、ボードあたり 10 倍程度高速である。

$$\begin{aligned} \text{(SCF 計算)} &= (\text{直交化計算} + \text{その他}) + \\ &(\text{電荷密度計算}) \times 3 + (\text{勾配計算}) \times 3 \dots \text{(i)} \\ &2512 + 6.043 \times 1024 \times 3 + 9.900 \times 1024 \times 3 = \\ &51489 \text{ sec (14h18m)} \dots \text{(ii-a)} \\ &2512 + 2.002 \times 1024 \times 3 + 3.836 \times 1024 \times 3 = \\ &20446 \text{ sec (5h41m)} \dots \text{(ii-b)} \end{aligned}$$

次にシステム性能として、Figure 1 中の Steepest Descent(SD)/Conjugate Gradient(CG) ループ一回の速度を見積ってみると、1 チップによるスピードアップは 5.34、非並列化部分は 0.123 であるので、アムダールの式で評価される加速率は Figure 10 のようになる。したがって、FPGA ボード 2 枚、すなわち、FPGA チップ 8 個を用いた場合に 10 倍の高速化が可能である。

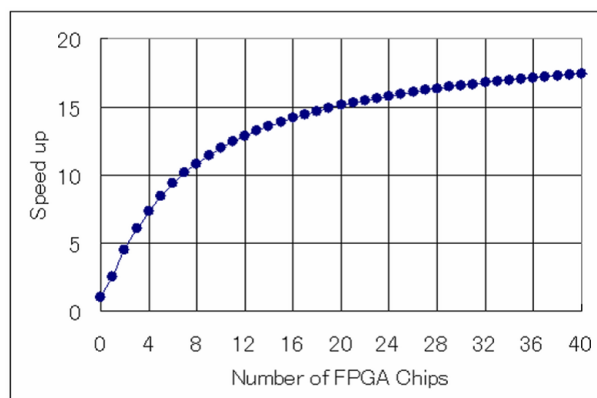


Figure 10. Speed up of the CP code by the FPGA boards

Table 2. 3D FFT performance of general purpose CPU and FPGA chip in msec.

Number of Taps		64 × 64 × 64	128 × 128 × 128	256 × 256 × 256
FFTW on Xeon @2.4GHz	FFT only	24	291	4542
	Charge Density	60	546	6043
	Gradient Vector	88	848	9900
3D FFT Logic on FPGA Chip	FFT only	25	205	1834
	Charge Density	28	228	2002
	Gradient Vector	52	432	3836
3D FFT Logic on FPGA Board	FFT only	6	51	459
	Charge Density	7	57	500
	Gradient Vector	13	108	959

## 6 考察

### 6.1 演算性能

三次元 FFT の持つメモリアクセスの特殊性を考慮し、演算器だけでなく、メモリシステムを三次元 FFT に特化し、さらにゼロパディング領域の演算を省略することにより、Xeon@2.4GHz を搭載した PC 上のソフトウェアで実行した場合と比較してボードあたり 10 倍程度の性能が達成できた。これは、専用ロジックを開発する場合に、単に実装する演算器や演算アルゴリズムだけでなく、メモリコントローラを含めて考えると有効な場合があることを示している、と考えられる。

### 6.2 低消費電力

EHPC-FPGA ボード全体の消費電力が約 15W であるから PC の消費電力を 100W 程度に見積もると 1/7 程度の電力である。消費電力まで考慮すれば、PC に対して数 10 倍のコストパフォーマンス比となる。

### 6.3 ダイナミック・リコンフィギャラブル・システム

CP 法では計算規模が 1000 原子を超えたあたりから三次元 FFT だけでなく、ベクトル直交化も大きな負荷となる [9]。例えばベクタ直交化として GS 直交化を用いた場合、 $N$  を原子数として計算規模のオーダーは

$$3D\text{-FFT} : O(N^2 \log N)$$

$$GS \text{ 直交化} : O(N^3)$$

である。このふたつの処理は同時に実行されることが無いため、同一の FPGA デバイス上で両者を行うことができれば HW の使用効率という点で有利である。従って、3D-FFT と GS 直交化の専用ロジックを開発し、それをリコンフィギャラブルシステム上で、適時ロジックをデバイスにロードして実行するという方法が有力であると考えている。そのため、ボード上のコントローラ CPU ( SH4 ) のメモリ上にロジック・レジスタというオブジェクトを生成し、複数のロジックを予めレジスタに登録しておけば任意の箇所でのロジックを動的に切り替えられる機構を設けている。

この機構を活用することによって、三次元 FFT と GS 直交化の両者を効率よく実行できるシステムを構築していくことも今後の課題である。

## 7 まとめ

FFT 固有のバタフライ演算に伴うメモリアクセスを工夫し、さらにゼロパディング領域に対する演算の省略を行うことにより、低消費電力の高速三次元 FFT 専用プロセッサを開発することができた。

今後はさらにリコンフィギャラブルという特徴を活かすことにより、より高性能な CP 法専用計算機の開発を目指す。

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] Car, R. and Parrinello, M., *Phys. Rev. Lett.*, **55**, 2471-2474 (1985).
- [2] 陽電子の第一原理計算では局所擬ポテンシャルで十分である。例えば、Kawasuso, A., Chiba, T., and Higuchi, T., *Phys. Rev. B*, **71**, 193204 (2005).  
Kawasuso, A., Yoshikawa, M., Itoh, H., Chiba, T., Higuchi, T., Betsuyaku, K., Redmann, F., and Krause-Rehberg, R., *Phys. Rev. B* (2005), in press.
- [3] Yokokawa, Itakura, Uno, Ishihara and Kaneda, 16.4TFlops Direct Numerical Simulation of Turbulence by Fourier Spectral Method on Earth Simulator, *SC2002* (2002).
- [4] <http://www.fftw.org/>
- [5] <http://www.ffte.jp/>
- [6] <http://momonga.t.u-tokyo.ac.jp/~ooura/fftman/index.html>
- [7] 高橋大介, 朴泰祐, 佐藤三久, Short Vector SIMD 命令を用いた並列 FFT の実現と評価, *SACIS2004*, 277-286 (2004).
- [8] 黒滝俊輔, 鈴木紀章, 安生健一朗, 本村真人, 若林一敏, 天野英晴, FFT の DRP によるアクセラレーション手法, *FPGA/PLD Conference and Exhibition* (2004).

- [9] 樋口高年, 谷村直樹, 大谷泰昭, 佐々木徹, 長嶋雲兵, 情報処理学会シンポジウムシリーズ ハイパフォーマンスコンピューティングと計算科学シンポジウム (HPCS2003), **2003(4)**, 133 (2003).
- [10] 村上和彰, 稲垣祐一郎, 上原正光, 大谷泰昭, 小原繁, 小関史朗, 佐々木徹, 棚橋隆彦, 中馬寛, 塚田捷, 長嶋雲兵, 中野達也, 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の開発 - プロジェクト全体像 -, *SWoPP2000* (2000).
- [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**, 313-320 (2004).
- [12] Payne, M. C., Teter, M. P., Allan, D. C., Arias, T. A. and Joannopoulos, J. D., *Rev. Mod. Phys.*, **64**, 1045 (1992).
- [13] 溝口大介, 荒木健悟, 佐々木徹, 青木すみえ, 棚橋隆彦, 科学技術計算向け FPGA 基板の設計と評価, 情報処理学会研究報告 *ARC*, **NO.153-02** (2003).
- [14] 青木すみえ, 溝口大介, 荒木健悟, 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, 日本流体力学会, 第 17 回数値流体力学シンポジウム, **C11-5** (2003).

## Reconfigurable 3D-FFT Processor for the Car-Parrinello Method

Tohru SASAKI<sup>a</sup>, Kiyoshi BETSUYAKU<sup>b</sup>, Takatoshi HIGUCHI<sup>b</sup> and Umpei NAGASHIMA<sup>c\*</sup>

<sup>a</sup>A-Priori Microsystems, Inc.

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

<sup>b</sup>Science and Technology, Mizuho Information & Research Institute, Inc.

2-3 Kanda-Nishikicho, Chiyoda-ku, Tokyo 101-8443, Japan

<sup>c</sup>Grid Technology Research Center, National Institute for Advanced Industrial Science and Technology

1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

\*e-mail: [u.nagashima@aist.go.jp](mailto:u.nagashima@aist.go.jp)

The execution time of Car-Parrinello based first principles calculation is dominated by 3D FFTs of electronic-state vectors. To accelerate these parts, the authors developed 3D-FFT logic, and implemented it on an FPGA board with four FPGA devices. The single board performs FFTs 10 times faster than a Xeon 2.4GHz processor. The speed up is about 50 times under the same power supply. With these two FPGA boards, we could accelerate the CP calculations 10 times faster than those on a Xeon 2.4GHz processor. For further acceleration of the CP codes, we propose a dynamic reconfigurable FPGA where both 3D-FFT and Gram-Schmidt orthogonalizations are performed.

**Keywords:** Car-Parrinello method, First principles calculation, Plane-wave expansion, FFT, Gram-Schmidt orthogonalization, Special-purpose computer, Reconfigurable, FPGA