

# Car-Parrinello 計算向け三次元 FFT ロジックの開発

佐々木 徹<sup>†</sup> 溝口 大介<sup>†</sup> 長嶋 雲 兵<sup>††</sup>

計算物理の分野において、第 1 原理計算としてよく使用されている Car-Parrinello 計算向けに、三次元 FFT ロジックを作成した。これを 300 万ゲート相当の FPGA 上に実装し、性能を実測してみたところ、100MHz で動作し、500 MFLOPS の性能が得られた。ボードあたり 4 チップ搭載しているため、ボードあたりの性能は 2 GFLOPS である。これは Pentium III @ 1GHz の Dual CPU 8 ノードの PC クラスタと比較して、FPGA ボード 1 枚でこの 1.5 倍に相当している。

## Development of a 3D-FFT Logic for Car-Parrinello Calculation

TOHRU SASAKI,<sup>†</sup> DAISUKE MIZOGUCHI<sup>†</sup> and UMPEI NAGASHIMA<sup>††</sup>

We developed 3D-FFT logic for Car-Parrinello calculation. In this paper, we report the performance of the 3D-FFT logic on the 3,000,000 system gate equivalent FPGA device. The logic performs 500 MFlops at 100 MHz per FPGA device, so that 2 GFlops per board. The result shows the FPGA board with the four 3D-FFT logics is about 1.5 times faster than the 8 node PC cluster with dual Pentium III at 1GHz.

### 1. はじめに

Car-Parrinello 法 (CP 法) は、1985 年に Car によって提案された手法である<sup>1)</sup>。この手法はバンド計算のような従来の固体の電子状態計算で用いていた、行列要素の対角化手法を用いることなく固有値および固有ベクトルを求めるため、計算量のオーダが  $N^3$  から最大  $N \log N$  程度まで減少し、さらには必要とする記憶容量も大体  $N^2$  から、 $N \times M$  のオーダに削減することができた。ここで、 $N$  は基底関数、 $M$  はバンド数であり、よく用いられる平面波基底では、基底関数の展開数  $N$  はバンド数  $M$  より 1 桁以上大きい。CP 法の開発によって、計算時間が大幅に短縮化され、系の電子状態とともに分子動力学計算により、構造最適化計算をも可能となった。CP 法の計算フローを図 1 に示す。計算時間のコストの大きい部分は、波動関数の直交化と電荷と勾配の計算部分の三次元 FFT 計算である。それぞれ原子数  $N$  に対して、 $N^3$  および  $N \log N$  に比例する演算量を必要とするが、我々の予備調査<sup>2)</sup> では、原子数が 1000 程度までの系

の場合、ほぼ 80%以上が三次元 FFT 計算に費やされ、圧倒的に三次元 FFT 計算のコストが高いことが明らかになった。

よく知られているように、FFT (高速フーリエ変換) のアルゴリズムは、DFT (離散フーリエ変換) と比較して、演算量が  $N^2$  から  $N \log N$  に減少するため、科学技術計算ばかりでなく、信号処理や画像処理においても非常に広く用いられており、その重要性ゆえに FFT の高速アルゴリズムに関する研究が数多くなされている<sup>3),4)</sup>。しかしながら、信号処理や画像処理ではサイズの大きな FFT を実行する必要がないため、DSP や FFT 専用回路を使用した FFT 専用ハードウェアも多く開発されてきたが、数百チップから数十 K チップ程度の小規模の FFT を、固定小数ないしは単精度浮動小数演算で高速に実行することを目的としている<sup>5),6)</sup>。

一方、科学技術計算の場合には物理量を三次元空間で波数表示されることも多く、そのような場合には三次元データに対してフーリエ変換を施す必要が出てくる。たとえば、我々がターゲットとしている CP 法のように電子の波動関数を平面波基底で展開した場合だけでなく、スペクトラル法による乱流解析<sup>7)</sup>、X 線回折像を用いた結晶構造解析<sup>8)</sup>、などをはじめとして適用範囲が非常に広く、三次元 FFT を効率良く実行することの重要度はきわめて大きい。

<sup>†</sup> 株式会社アプリアリ・マイクロシステムズ  
A Priori Microsystems, Inc.

<sup>††</sup> 独立行政法人産業技術総合研究所グリッド研究センター  
Grid Technology Research Center, National Institute of  
Advanced Industrial Science and Technology

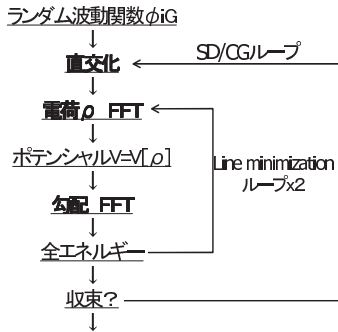


図1 Car-Parrinello 計算のフロー  
Fig.1 Flow of Car-Parrinello calculus.

FFT は演算量を削減した代償としてバタフライ演算にもなうメモリアクセスが連続アドレスにならないという問題点をかかえている．多次元 FFT の場合には，各次元に対応する座標軸に沿ったスキャンラインに対して個々に FFT を施す必要があり，メモリの並びと一致する座標軸以外のスキャン方向では隣接データ間のストライドが大きいため，この問題が非常に深刻になる．また，たとえば汎用コンピュータを使用する場合には，小規模な FFT でキャッシュメモリにすべてのデータが収まってしまえば，このようなメモリアクセスの問題はほとんど考慮する必要がないが，データサイズが巨大になり，キャッシュメモリのサイズをはるかに超えるようになると，キャッシュメモリのミス率の増加を抑える特殊な工夫が必要となってくる．たとえば，FFT のスキャン方向とメモリ上の並びに合わせるためにデータを転置するといった方法がとられている<sup>4)</sup>．すなわち，大規模な多次元 FFT の場合には，演算アルゴリズムだけでなく，データアクセスも高速化の重要なポイントとなっているのである．

三次元空間を表現する場合，データサイズが空間分解能の 3 乗に比例するため，扱うべきデータのサイズがきわめて大きくなってしまふ．実際，CP 法では Si 原子 1000 原子程度の系を計算する場合， $256^3$  (倍精度複素数で 256 MB) サイズの計算となり，CP 法を PC クラスタ上で実行する際の大きな障壁となっている．また，Si1000 原子程度の系の場合には，バンド数を 2,000 程度とる必要があり，個々のバンドに対して個々に三次元フーリエ変換を施す．これは 256 MB からなる FFT 空間 2000 個に対して独立に FFT を施すことを意味している．そこで，我々は並列化するにあたり，並列化の粒度が大きくとれるので，FFT 自体を分割して実行するのではなく，並列化手法として Band-By-Band による並列化を行っている<sup>2)</sup>．

本論文では，CP 法の 1 つのボトルネックとなってい

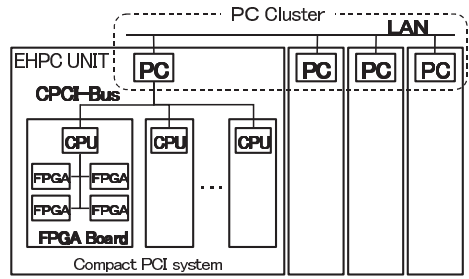


図2 評価システムの構成  
Fig.2 Block diagram of evaluation system.

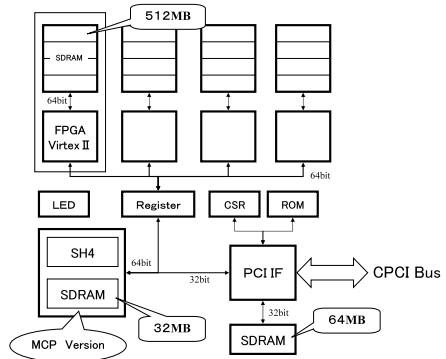


図3 FPGA ボードの構成  
Fig.3 Block diagram of FPGA board.

る三次元 FFT を高速化するにあたり，従来行われてきたソフトウェアによる高速化ではなく，演算ユニットとメモリシステムを一体化することにより効率良く三次元 FFT を行う専用ロジックを開発し，そのロジックを FPGA デバイス上に実装して性能評価を行ったので，その報告を行う．

## 2. ロジック評価に用いたシステムの構成

### 2.1 全体構成

まず，ロジックの評価に使用したシステムの構成を図 2 に示す．図 3 は FPGA ボードの構成を示す．図 4 は FPGA ボードの写真である．これらは EHPC プロジェクト<sup>9)</sup>において，種々の専用ロジックを評価する目的で作成したものである<sup>10)</sup>．このシステム上では，これ以外にも流体解析専用ロジック<sup>11)</sup>すでに稼働している．

単位ユニットは CompactPCI サブラックに搭載され，システムスロットに挿入した PC 互換カードと IO スロットに挿入した FPGA ボードからなる．サブラックにはホストとなる PC 互換カードに接続するハードディスクなどの IO 装置も具備しており，ホスト PC の OS として Linux を実装しているので，機能的に PC とまったく互換である．また，ホスト PC 間をネッ

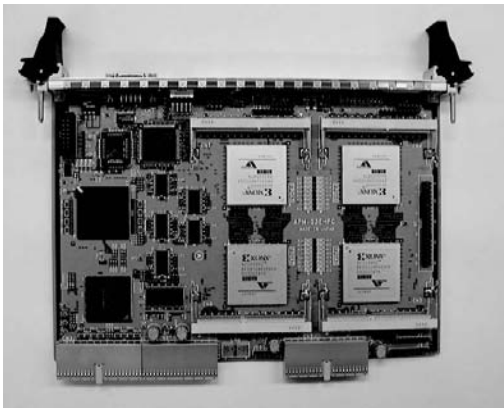


図 4 FPGA ボード

Fig. 4 Photograph of FPGA board.

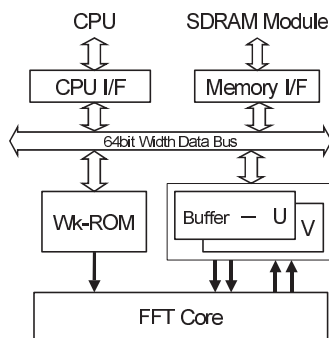


図 5 三次元 FFT ロジックの構成

Fig. 5 Block diagram of 3-dimensional FFT logic.

トワーク接続してクラスタ化することもできる。

## 2.2 FPGA ボード

FPGA ボードはコントローラとして機能する汎用 CPU と演算アクセラレータとして機能する FPGA デバイス 4 個からなり、ボードのサイズは CompactPCI 規格の 6U サイズである。

コントローラ CPU としてルネサステクノロジ製 SH4 (メモリ 32 MB 内蔵型) @ 200 MHz を、PCI ブリッジとして SH4 と直結できるファミリーデバイスのブリッジチップをそれぞれ搭載している。

FPGA デバイスには 300 万ゲート相当のゲート容量を持つ Virtex-II を使用し、個々の FPGA デバイスには 512 MB の SDRAM モジュールが接続されている。

## 3. 3 次元 FFT ロジック

### 3.1 FPGA 上に実現したロジックの全体構成

図 5 に今回作成した三次元 FFT ロジックの構成を示す。この三次元 FFT ロジックが FPGA ボード上の個々の FPGA 上にローディングされて FFT を実

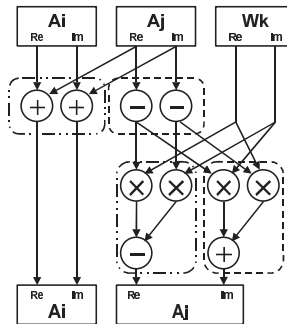


図 6 FFT 演算ユニットの構成

Fig. 6 Block diagram of FFT processing unit.

行する。

1. 一次元 FFT を行う演算ユニット (FFT Core, Buffer-U/V, および Wk-ROM)
2. コントローラ CPU との通信を行う CPU I/F
3. SDRAM モジュールを制御する Memory I/F
4. ループ制御を行うシーケンサ

などのブロックから構成される。Wk-ROM は回転因子を格納する定数テーブルである。なお、シーケンサなどの制御論理は図中から省いている。

### 3.2 FFT 演算ユニット

FFT 演算ユニットは、SDRAM からフェッチしたデータに対して一次元 FFT を行う演算ユニットである。ハードウェアによる実装であり、ロジック開発の主眼をメモリアクセスの改善に置いたので、複雑なアルゴリズムは避けて、基数 2 の平易な Culey-Tukey アルゴリズムを実装した。パイプラインの段数は 27 段で、その内訳は

- RAM アドレス生成 1 段
- RAM 読み出し 2 段
- 加算 7 段
- データ転送 1 段
- 乗算 6 段
- データ転送 1 段
- 加算 7 段
- データ転送 1 段
- RAM 書き込み 1 段

である。図 6 に FFT 演算ユニットの構成を示す。なお、演算器は倍精度浮動小数点の演算を行う。

図 6 において、 $A_i, A_j$  はスキャンライン上の  $i$  番目と  $j$  番目の要素で Buffer-U/V に格納されている。Wk は Wk-ROM に格納された回転因子である。Culey-Tukey のアルゴリズムはコアループにおいて、6 個加算と 4 個の乗算 (i.e. 2 個の複素加算と 1 個の複素乗算) を行うが、今回使用したデバイスではゲート容量

の制限から、この半分の3個の加算器と2個の乗算器しか搭載できなかった。そこで1クロックごとにモードを切り替えて、2サイクルを1ピッチとして処理を行っている。複素加算器1個と複素乗算器の半分が実装されているので、これらを1クロックごとにモジュールに動作させている。図中、点線が奇数サイクル、2点鎖線が偶数サイクルの動作を表している。

演算ユニットには、SDRAMからフェッチしたデータを保持するためのバッファ (Buffer-U/V) を備えており、1スキャンライン保持して演算器に供給する。また、データ転送時間を隠蔽するためにダブルバッファとしているが、ダブルバッファの効果については後述する。このバッファは単なる2ポートRAMであり、2サイクルを1ピッチとしているので、奇数サイクルに2オペランドリードし、偶数サイクルに2オペランドライトしている。

回転因子は、FPGAの内部RAMに定数テーブルとして保持している。計算開始時に外部からロードして使用するので、タップ数を変更するときなど必要に応じて書き換えることができる。

### 3.3 CPU I/F

コントローラCPUであるSH4はDMAコントローラを内蔵しており、64ビット幅でバースト転送を行うことができる。バースト転送される倍精度浮動小数点データの入出力用にそれぞれ64ビット幅のFIFOを置いている。また、タップ数などの演算パラメータの設定やコントローラCPUとの同期制御はCPU I/F内に置かれたCSR (Control and Status Register) を介して行われる。

### 3.4 メモリアドレスの生成

一般に多次元配列をラスタ展開する場合には、各軸成分のIndexを連結して展開されることが多い。しかし、単純に連結すると多次元配列データにアクセスする場合に、ストライドが大きくなってしまい、バースト転送を行っても転送効率が上がらない。

そこで、図7のようにメモリアドレスを生成する際に、XYZの各INDEXにスクランブルを施してメモリ (SDRAM) のアドレスを生成している。複素数の実数部と虚数部が1組になるように並べれば、SDRAMのバースト長を2に設定できて、配列インデックスがインクリメントされても、Rowアドレスが切り替わらない限り、Columnアドレスのセットだけで、切れ目なくメモリアクセスが行える。このようにしてXYZ各スキャン軸に対して、すべての軸方向に対して16ワードの擬似的なバースト転送を行うことが可能となる。

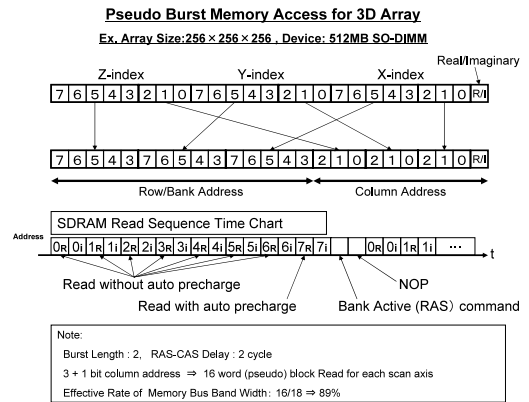


図7 メモリアドレスの生成

Fig. 7 Memory address generation.

実効的なメモリバンド幅を見積もってみると、SDRAMモジュールを100MHzで動作させた場合、RAS/CAS delayを2サイクルに設定できるので、RefreshやWrite Recover Timeを無視すれば、Rowアドレスの切り替わりの際にRowアドレスを設定するためのペナルティは2サイクルである。したがって、メモリバス稼働率は

$$16/18 = 89\%$$

であり、実効メモリバンド幅は

$$8 \text{ バイト} \times 100 \text{ MHz} \times 89\% = 711 \text{ MB/sec}$$

となる。

### 3.5 ビットリバーサルアドレッシング

FFTはアルゴリズムの特性上、計算の前後でデータ順が入れ替わってしまうため、ソフトウェアで実行する場合にはFFTを施す前か後にビットリバーサルアドレスアクセスによって、データを入れ替える必要がある。ハードウェアで実現するには所望のデータにアクセスする際に単純にアドレス線のビット順の上位下位をつなぎかえるだけでよい。この処理はSDRAMと内部バッファとの間でデータを転送する際に行うこととし、サイズの小さいFPGAの内部SRAMで実現されているバッファに対してビットリバーサルアドレスアクセスを行う。

### 3.6 データ転送時間の隠蔽

FPGA内部のスキャンラインのバッファをダブルバッファ構成にしている。これは画像処理などでよく採用されている方法であるが、この場合にも有効に機能し、データ転送時間を効果的に隠蔽することが可能である。ダブルバッファリングすることにより、メモリとのデータ転送とFFTの演算を、図8のようにパイプライン動作させることが可能となり、データ転送時間を隠蔽することが可能となる。

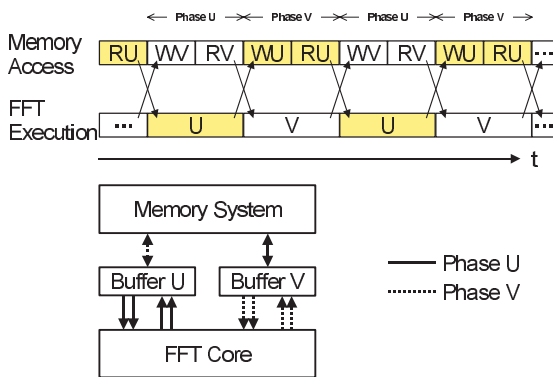


図 8 ダブルバッファの動作  
Fig. 8 Double buffer behavior.

### 3.7 ゼロパディング領域の処理

CP 法では、演算精度を保証する場合には、1 つのスキャンライン上に少なくとも有効データ領域と同程度以上の領域にゼロパディングする必要がある<sup>2)</sup>。一次元 FFT の場合には、ゼロパディングの領域の処理方法は性能上大きな問題にはならないが、多次元 FFT の場合には、データがすべてゼロとなるスキャンラインが多数存在するので、不必要な計算が生じてしまい、これを FFT の演算から除去できると大きな効果がある。三次元 FFT の場合で、その効果を試算してみると、演算量は、X 軸 Y 軸 Z 軸の順にスキャンするとして

X 軸 Scan FFT : 1/4

Y 軸 Scan FFT : 1/2

Z 軸 Scan FFT : 1 (効果なし)

Total = (1/4 + 1/2 + 1)/3 = 7/12

となる。したがって、最低でも約 40% 演算量を削減できる。ゼロパディング領域に対する無駄な演算を除去する機構を備えた三次元 FFT ロジックは現在実装を終えてデバッグを行っている。この効果については次の機会に報告する。

## 4. 評価結果

### 4.1 チップ単体の演算時間測定

タップ数  $64^3$ ,  $128^3$ ,  $256^3$  の場合についての実行速度の測定結果を表 1 に示す。なお、理想値は演算律速の場合の計算時間

タップ数/2 ×  $\log_2$  タップ数 × 演算ピッチ

から算出した。演算ピッチはコアループの半分しか演算器を搭載していないので、クロック周期の倍の 20 nsec である。また、今回の評価はメモリバンド幅と演算速度の関係に焦点を絞っているため、FFT を施すデータをあらかじめ SDRAM モジュールにロードしてか

表 1 三次元 FFT 実行時間

Table 1 Execution time of 3D FFT.

タップ数	理想値 (msec)	実測値 (msec)
$64^3$	47	47
$128^3$	440	440
$256^3$	4027	4027

ら測定した。

実測値は理想値と非常によく一致しており、これらのケースではメモリアクセスがネックではなく、演算器が計算時間を律速していることが分かる。実際、1 スキャンラインのメモリアクセス時間と FFT の演算時間を、256 タップの場合で見積もってみると、メモリアクセス時間は倍精度複素数を 64 ビット幅 RAS/CAS delay2 の SDRAM @ 100 MHz に対してリード・ライトを行うので、

$$256 \times 2 \times 2 \times 18/16 \times 10 \text{ nsec} = 11,520 \text{ nsec}$$

である。一方、FFT 処理時間は

$$256/2 \times \log_2 256 \times 20 \text{ nsec} = 20,480 \text{ nsec}$$

となり、机上の見積りからも演算律速であることが分かる。ダブルバッファリングの効果により、演算器とデータ転送がパイプライン動作してデータ転送時間が隠蔽できている。なお、ループの前後で 1 スキャンラインのリード・ライトのオーバーヘッドがあるが、ただか、数十  $\mu\text{sec}$  でありほとんど無視できる。したがって、5 個の浮動小数点演算器が動作周波数 100 MHz、稼働率ほぼ 100% であるから、チップあたりの性能は約 500 MFlops である。

### 4.2 ボードの演算性能

本システムでは CP 法を各バンド (1000 原子なら 2000 バンド程度) ごとの独立に FFT を実行する Band-By-Band による並列化を行っており、1 バンドの FFT を FPGA1 個に割り当てて、各 FPGA が独立に FFT を実行できるので、ホストからのデータ転送がボトルネックにならなければ、4 個の FPGA を搭載したボードでは、ほぼリアに 4 倍の性能向上が期待でき、ボードあたり 500 MFLOPS × 4 個 = 2 GFLOPS の性能となる。

### 4.3 他の手法との比較

今までに DSP や専用回路を使用した FFT 専用ハードウェアが開発されてきた<sup>5),6)</sup> が、従来の専用ハードウェアは固定小数ないしは単精度浮動小数による数十 K タップの小規模な FFT を高速に実行することを目的としており、本論文のように倍精度複素数による  $256^3$  タップを超えるような三次元 FFT を実行することのできる専用ハードウェアはほとんど前例がなく、スーパーコンピュータないしは PC クラスタを用いてソ

ソフトウェアによって実現されたものしか確認できていない。そこで PC との比較を試みる。

まず、PentiumIII との比較してみる。高橋らが PentiumIII @ 1GHz を使用した Dual CPU 8 ノードの PC クラスタで達成した FFT の演算速度は約 1.3 GFlops である<sup>4)</sup>。したがって、本論文の FFT ロジックを FPGA ボード 1 枚に搭載したものは、これの約 1.5 倍の性能に相当する。

次に、Pentium4 などのさらに高速な CPU を搭載した PC との比較を 256<sup>3</sup> タップの FFT の場合で行ってみる。Xeon @ 2.8GHz Dual CPU (FSB: 400 MHz, L2Cache: 512 MB) を使用した場合、高橋らの FFTE3.0<sup>12)</sup> は 2 CPU 用いて 561 MFLOPS、世界的に広く使われている FFTW<sup>13)</sup> Ver.2.1.5 の単一 CPU 版 (MPI 版もあり) は 384.1 MFLOPS を達成している。Pentium4 (Prescott) @ 3.2GHz (FSB: 800 MHz, L2Cache: 1 MB) を使用した場合で、単一 CPU 版のみ公開されている FFTW 最新版 Ver.3.0.1 は、834.6 MFLOPS である (これらのベンチマークデータは高橋大介氏の私信による)。今回、三次元 FFT ロジック評価に使用したデバイスの能力 (演算器 100 MHz, SDRAM 100 MHz) よりもはるかに能力の高いハードウェア性能を持った PC と比較して、チップ単体でも遜色のない性能が得られた。筆者らはこれをメモリアクセスの効率化が大きく寄与した結果であり、最新のメモリデバイスをフォローできれば、PC に対する優位性を保つことも十分に可能であると考えている。

また、現段階で対電力効率を云々するのははなはだ早計ではあるが、FPGA ボードの消費電力が FPGA1 個につき 2.5 W、ボード全体でも 15 W 程度であることを付け加えておく。

## 5. 今後の課題

今回の実装は原理試作であり、より効果的にシステムに組み込むためには、高速化や機能の拡充を施す必要がある。また、実用化に向けた検討として、ロジックの単体の性能ではなく、システムに組み込んだ場合のデータ転送を含めた性能が重要である。

### 5.1 演算性能の向上

演算性能向上を図るには、一般論としては

- 演算器数の増強
- 動作周波数の向上
- メモリアクセス速度の向上

などの方策が考えられる。より大きなゲート容量を持ったデバイスを使用すれば、演算器を増強すること

ができると予想されるので、次のバージョンとして 600 万ゲート相当の Xilinx 製 Virtex-II Pro を搭載した FPGA ボードを作成中である。300 万ゲート相当の FPGA では、現状のロジックが全体でゲートの 60% 程度使用し、そのうち FFT 演算ユニットは 30% 程度ゲートを占有しているため、600 万ゲート規模のデバイスの場合、演算器が現在の 4 倍程度搭載できることが期待される。また、浮動小数点演算器単体であれば、論理合成 CAD が出力するレポートが、120 MHz 以上で動作可能であることを示しており、動作周波数も大きく改善できる余地がある。さらに今回評価に使用した FPGA ボードでは 512 MB SDRAM モジュールを搭載しているが、1 GB の容量を持った DDR-SDRAM モジュールを搭載することにより、メモリアクセス時間の軽減、および大規模計算への適合性を高めることができる。

### 5.2 データ転送

一般に、ロジック単体がいかに高速であっても、ホストとのデータ転送がネックになってしまうと所望の性能が得られないが、CP 法の場合には、ホストから FFT 空間すべてをホストとロジックの間で転送する必要がなく、波動関数だけ転送すればよいので転送すべきデータのサイズは演算量と比較して小さいので大きな問題とはならない。たとえば、1000 原子の系では 1 バンドの波動関数のサイズは 4 MB 強である<sup>2)</sup> から、その転送時間を見積もってみると、仮に PCI バスの転送レートを 70 MB/sec としても 0.06 sec 以下となり、256<sup>3</sup> タップの FFT の実行時間約 4 sec と比較して非常に小さい。また、ボード上の CPU から各 FPGA の転送レートは 400 MB/sec である<sup>10)</sup> から、こちらも問題にならない。ただし、データ転送量を削減するために波動関数を転送する場合には、転送されてきた波動関数をボード上でゼロパディングして FFT 空間を生成しなければならなくなる。そのためには、FPGA の内部 RAM をゼロクリアして、必要な箇所に波動関数を埋め込む、などの特殊なロジックが必要となる。

### 5.3 機能の拡充

CP 法では、三次元 FFT の前後処理として、電荷密度計算や勾配ベクトル生成も同一チップで実行できるとデータ転送量がさらに削減できて、より専用ロジックとしての効果が高い。これらの拡張機能の実装を考えた場合、演算器の接続を変更する程度で実現できることが望ましい。今回は Culey-Tukey アルゴリズムに基づいたロジックを実装したが、Danielson-Lanczos アルゴリズムを比較してみると、どちらも基数 2 の一次元 FFT なので性能上の差はないが、Danielson-

Lanczos アルゴリズムは、最初に加算ではなく乗算を行うため、こちらをベースにしたほうが拡張機能を実装するのに適している可能性もあり、今後検討を要する。

また、さらに巨大な系を計算する場合、ベクタ直交化がボトルネックになると予想されている<sup>2)</sup>。Gram-Schmidt 直交化をロジック化することも検討しているが、FFT とベクタ直交化を 1 つのロジックで実現すべきか、あるいはロジックを動的に切り替えてリコンフィギュラブルシステムとして実現すべきか、ということも今後の検討課題である。

## 6. ま と め

CP 法のボトルネックとなっている三次元 FFT を高速化するための専用ロジックを開発し、300 万ゲート相当の FPGA を 4 個搭載したボード上に実装して性能評価を行った。その結果、89% という高いメモリバス使用率を達成し、このロジックの持つ性能が FPGA 1 個あたり 500 MFLOPS であることが確認できた。ボード 1 枚では 2 GFLOPS となり、これは Pentium4 @ 1 GHz の Dual CPU 8 ノードの PC クラスタに対して、約 1.5 倍に相当する。

今回の実装は原理試作であり、5 章で述べたように実用化までの課題は決して少なくない。しかしながら、メモリシステムを特殊化することにより三次元 FFT が高速に実行できるという方向性自体の妥当性は確認することができた。このことから筆者らは、専用計算機や専用ロジックの目指す方向として、アプリケーションによっては、演算器の専用化だけでなく、メモリシステムまで含めて専用化する必要がある、と考えている。

謝辞 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」によるものである。また、研究環境を提供していただいた慶應義塾大学先端科学技術研究センター、また、ロジック化およびリコンフィギュラブルシステムに関して九州大学村上和彰教授、慶應義塾大学天野英晴教授、FFT に関して筑波大学高橋大介講師、CP 法に関して東京大学塚田捷教授、大谷泰昭氏、樋口高年氏はじめとする富士総合研究所科学技術グループのみなさんなど、多くの方々に数多くの有用なご議論をいただいたことをここに深く感謝いたします。

## 参 考 文 献

1) Car, R. and Parrinello, M.: *Phys. Rev. Lett.*, Vol.55, pp.2471-2474 (1985).

- 2) 樋口, 谷村, 大谷, 佐々木, 長嶋: Car-Parrinello コードの組み込み型高性能計算機への実装, 情報処理学会, HPCS2003 (2003).
- 3) <http://momonga.t.u-tokyo.ac.jp/%7Eooura/fftman/index.html>
- 4) 高橋, 朴, 佐藤: PC クラスタにおける並列一次元 FFT のブロックアルゴリズム, 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, Vol.43, No.SIG6(HPS 5) (2003).
- 5) 小畑: 浮動小数点 DSP による高並列アレイプロセッサシステム, 情報処理学会論文誌, Vol.32, No.9 (1991).
- 6) 黒瀧, 鈴木, 安生, 本村, 若林, 天野: FFT (Fast Fourier Transform) の DRP によるアクセラレーション手法, FPGA/PLD Conference and Exhibition (2004).
- 7) Yokokawa, Itakura, Uno, Ishihara and Kaneda: 16.4 Tflops Direct Numerical Simulation of Turbulence by Fourier Spectral Method on the Earth Simulator, *SC2002* (2002).
- 8) 池田, 坂田, 高田, 西堀, 山村: MEM による電子密度分布のイメージング—蛋白質結晶構造解析への適用を目指して, 構造生, Vol.7, No.2 (2001).
- 9) 村上, 稲垣, 上原, 大谷, 小原, 小関, 佐々木, 棚橋, 中馬, 塚田, 長嶋, 中野: 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, 情報処理学会, *SWoPP2000* (2000).
- 10) 溝口, 荒木, 石橋, 佐々木, 青木, 棚橋: 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「計算機アーキテクチャ」, No.153-02 (2003).
- 11) 青木, 溝口, 荒木, 石橋, 佐々木, 棚橋: 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 回数値流体シンポジウム, C11-5 (2003).
- 12) <http://www.ffte.jp/>
- 13) <http://www.fft.org/>

(平成 16 年 1 月 31 日受付)

(平成 16 年 5 月 9 日採録)



佐々木 徹（正会員）

昭和 33 年生。昭和 58 年東京大学教養学部基礎科学科卒業。昭和 60 年東京大学大学院工学系研究科金属材料専攻修士課程修了。同年株式会社東芝半導体技術研究所。平成 2 年株式会社ソニー木原研究所。平成 10 年株式会社アプリアリ・マイクロシステムズ取締役。平成 12 年同代表取締役。プロセッサアーキテクチャ、並列処理アーキテクチャ、各種専用プロセッサ・専用計算機の研究開発に従事。



溝口 大介

昭和 50 年生。平成 6 年九州大学電気系学科入学。平成 12 年九州大学大学院システム情報科学研究科情報工学専攻修了。同年（株）アプリアリ・マイクロシステムズ入社。平成 13 年 9 月慶應義塾大学大学院理工学研究科後期博士課程入学。



長嶋 雲兵（正会員）

昭和 30 年生。昭和 58 年北海道大学大学院理学研究科博士後期課程化学第二専攻修了。理学博士。同年岡崎国立共同研究機構分子科学研究所電子計算機センター助手。平成 4 年お茶の水女子大学理学部情報科学科助教授。平成 8 年同教授。平成 10 年通産省工業技術院物質工学工業技術研究所基礎部理論化学研究室長。平成 11 年同産業技術融合領域研究所計算科学研究グループ長。平成 13 年独立行政法人産業技術総合研究所先端情報計算センター情報基盤研究開発室長。平成 14 年同グリッド研究センター統括研究員。筑波大学連携大学院大学教授。計算化学、情報化学、大規模数値計算、広域分散並列処理の研究開発に従事。日本化学会、IEEE、応用数理学会、コンピュータ化学会各会員。