書き換え可能ハードウェアを用いた高速ホモロジー検索システム

山口佳樹[†] 宮島洋介[†]
丸山勉[†] 小長谷明彦[††,†††]

本論文では，市販の Field Programmable Gate Array（FPGA）ボード用いた高速ホモロジー検索システムについて述べる．このシステムは，使用するコンピュータの PCI バスに市販の FPGA ボードを 1 枚挿入するだけで，高速ホモロジー検索を実現するものである．本論文では高性能を得るために，プログラム実行中にそのときどきに応じた最適な回路を FPGA 上に実現している．このシステムで，約 6400 万要素のデータベース配列と 2,000 要素の問合せ配列のホモロジー検索を実行した結果，約 75 秒で検索を終了させることができた．これは Pentium III 1 GHz に対し約 150 倍の処理速度である．

High Speed Homology Search System with Reconfigurable Hardware

Yoshiki Yamaguchi,[†] Yosuke Miyajima,[†] Tsutomu Maruyama[†]
and Akihiko Konagaya[††,†††]

In this paper, we show that we can achieve high speed homology search by only adding one off-the-shelf PCI board with one Field Programmable Gate Array (FPGA) to a Pentium based computer system in use. FPGA is a reconfigurable device, and any kind of circuits, such as pattern matching program, can be realized in a moment. The performance is almost proportional to the size of FPGA and we can easily obtain latest/larger FPGAs by using off-the-shelf PCI boards with FPGAs, at low costs. The time for comparing a query sequence of 2,048 elements with a database sequence of 64 million elements by the Smith-Waterman algorithm is about 75 sec, which is about 150 times faster than a desktop computer with a 1 GHz Pentium III.

1. はじめに

近年，遺伝子情報配列の解析が進み，遺伝子情報（配列の要素）の特定が終了する遺伝子の個数が飛躍的に増加している．このため現在，遺伝子情報配列のデータベースは指数的に増加している．

従来，数百万要素を持つデータベース配列と数百要素を持つ問合せ配列のホモロジー検索においては，ソフトウェアによる高速な検索（FASTA [1]，BLAST [2),3)] など）が用いられている．しかし，高速化のために省略している処理（ギャップの挿入など）があるためにこれらの手法を用いてホモロジーの高い部分の全可能性を検索することは難しい．

† 筑波大学機能工学系
Institute of Engineering Mechanics and Systems, University of Tsukuba
†† 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
††† 理化学研究所ゲノム科学情報センター
Japan Riken Genomic Sciences Center

そこで連続ギャップの挿入などを考慮し，動的計画法を基にしたアルゴリズムである Smith-Waterman 法[4)~6)]などを用いることが考えられる．しかし，その検索時間は遺伝子データベースの要素数（M）と問合せ配列の要素数（N）の積（$O(MN)$）となるため，ソフトウェアでその計算を行うことは時間的に現実的でない[7)]．

そこで，このような連続ギャップを考慮した検索手法を高速に処理するために，ASIC や書き換え可能なハードウェアである Field Programmable Gate Array（FPGA）などを用いたハードウェアシステムの研究が行われ[8)~10)]，いくつかのシステムは実際に商用化されている[11)~13)]．しかし，これらのシステムは専用 LSI または FPGA が多数用いられ，また，それらが搭載されるボードも一般に専用設計となっているため，システム単体が非常に高価になる傾向がある．さらに，デバイス技術の進歩にともなう専用 LSI や専用ボードの再設計，およびシステム全体の再構築，などの問題からユーザが最新のシステムを導入し続ける

ことは非常に難しい．

近年，FPGA などの書き換え可能ハードウェアの回路規模の進歩は著しく，FPGA 1 チップで様々な規模の回路のラピッドプロトタイピングを行うことが可能となってきた[14]．このため，各メーカより最新かつ大規模な FPGA 1 チップを搭載した様々な PCI ボードが比較的安価にかつ継続的に提供されている[15]．FPGA を搭載したこれらの PCI ボードをデスクトップ PC に組み込むことで高速検索システムを構築することができれば，上述した専用ハードウェアシステムの問題点を解決することができ，幅広いユーザがより簡単に高速検索システムを利用することが可能となる[16]．

そこで本論文では，市販の FPGA ボード 1 枚をデスクトップ PC に組み込むことで高速ホモロジー検索を実現することを目指した．しかし，市販の PCI ボードではメモリバンド幅などが十分ではないため，FPGA の回路規模に比例した速度向上を得ることができない．この問題を解決する手段として本論文では 2 段階検索を提案した．この手法によりメモリバンド幅などの入出力のボトルネックを回避し，FPGA の回路規模に比例した速度向上を簡単に得ることができる．

本論文では，最新の FPGA（回路規模は 250 万ゲート相当）が組み込まれている PCI ボード 1 枚で，約 6400 万要素のデータベース配列と 2,000 要素の問合せ配列とのホモロジー検索を約 75 秒で実行できることを確認した．これは PentiumIII 1 GHz に対し，約 150 倍の高速化である．

以下，2 章で本論文で提案する高速ホモロジー検索システムの概要を示し，3 章ではシステムに実装する動的計画法のハードウェア化について，4 章ではシステムの詳細について述べる．5 章ではシステムの評価を行い，最後に，6 章でそのまとめと今後の課題について述べる．

2. システム概要

2.1 システム構成

本論文が提案するホモロジー検索システムは以下の要素より構成されている（**図 1**）．

- FPGA が組み込まれている市販の PCI ボード
- PCI バスを持つ市販のコンピュータ（Host-PC）（基本ソフトは Windows もしくは LINUX）

また使用する FPGA ボードは Host-PC と FPGA ボードの間のデータ通信を高速に実行する必要があるため以下の機能を必要とする．

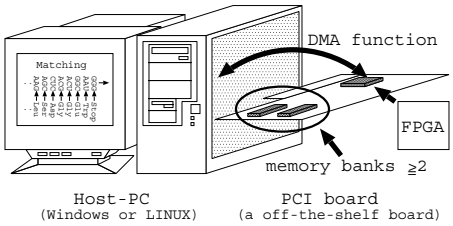

図 1 システムの構成
Fig. 1 Hardware platform.

- DMA 転送機能
- 2 個以上のメモリバンク

FPGA ボード上にメモリバンクが 1 個しかない場合，Host-PC がメモリとの通信を終了しない限り，FPGA はメモリ上の必要なデータにアクセスすることができない．そこで本論文の提案するシステムでは 2 個以上のメモリバンクを用いることで，FPGA と Host-PC がメモリバンクに交互にアクセスすることを可能にし，データ転送と FPGA によるホモロジー検索の並列化を実現している．

2.2 システムの特徴

本節では，ホモロジー検索システムに市販品を使用することで得られる利点とそれによる問題点について述べる．

初めに，市販品を使用することによる主な利点は以下の 4 つである．

- 数多くの企業が FPGA ボードを供給しているので予算に応じたシステムを構築しやすい．
- 最新の FPGA を組み込んだ PCI ボードが継続的に市場に供給されるのでユーザはそのときどきの最高性能のボードを手に入れやすい．
- FPGA ボード /Host-PC（のパーツ）を必要に応じて簡単に交換できる．
- 作成したプログラム/回路を公開することで計算機を不得手とするユーザでも簡単にシステムを更新（FPGA を再構成）できる．

遺伝子データベースの規模が指数的に増大している現在においてこれらの特徴は非常に重要なものである．特に，遺伝子のホモロジー検索では FPGA の回路規模に比例した高速化を得ることができるため，最新かつ大規模な FPGA を市場から入手しやすい，という特徴は非常に重要である．

一方で，市販品のみで構成するために避けられない問題点を以下に示す．

- FPGA ボードのハードウェア資源（メモリバンド

幅，メモリサイズ，FPGA の大きさ，など）がホモロジー検索をするに十分でないため，長い 2 本の配列を 1 度に比較することができない．
- ユーザが使いやすい十分なグラフィックユーザインタフェース（GUI）が用意されていない．

以上の問題点の中で特にメモリバンド幅とメモリサイズはホモロジー検索の高速化において非常に厳しい制限となっている．本論文ではこれらのハードウェア資源による制限を克服するために以下の手法を用いた．

（1）2 段階検索

高いホモロジーが得られる場所を高速に検索する第 1 フェーズとその詳細を出力する第 2 フェーズの 2 つのフェーズを用いて検索を行う．その際，使用する FPGA に実装する回路をプログラム実行中に動的に書き変える（Runtime Reconfiguration）ことにより限られたメモリバンド幅において高速な検索を実現する．

（2）データベース配列の分割

長いデータベース配列を 1 度に計算するためには，オーバ（アンダ）フローを避けるため，FPGA 内に実装する各演算器の演算データ幅を広くする必要がある．しかし，演算データ幅を広くすると演算器の動作周波数が低下し，また FPGA 内に実装可能な演算器数（並列度）も少なくなる．

そこでデータベース配列を適度な大きさの領域に分割し必要な演算データ幅を減らすことで，より高い動作周波数と，より高い並列度を実現することができる．このとき，分割された領域にまたがる部分に関しても正しい検索結果が得られるように，各領域の両端のデータが前後の領域と十分な長さだけ重複するように分割されなければならない．

2.3 システムの拡張性

本論文で提案するシステムでは以下にあげるハードウェア資源に比例した性能向上を得ることができる．

- FPGA の大きさ（回路規模）
- Host-PC に組み込む FPGA ボードの数
- Host-PC の台数

ただし，FPGA ボードの枚数および Host-PC の台数により性能向上を得る場合は以下のいずれかの条件を満たす必要がある．

- 異なったハードディスクにある複数のデータベース配列を問合せ配列と比較する

```
Database Seq.: D[M];
Query Seq.   : Q[N];
Score Matrix : S[nucleotide/amino acid][同左];
Optimal Score: O[column][row];

for( i=0; i<M ; i++ ){
  for( j=0; j<N ; j++ ){
    O[i][j]= max( O[i-1][j-1] + S[d[i]][q[j]],
                  O[i-1][j]   + gapcost(),
                  O[i][j-1]   + gapcost() );
  }
}
```

図 2 動的計画法
Fig. 2 Dynamic programming.

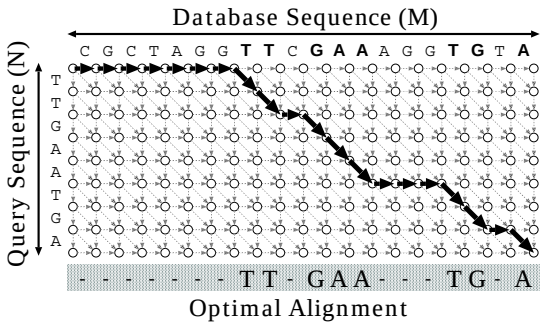

図 3 最適パスの検索
Fig. 3 Finding the optimal alignment.

- 異なったハードディスクの中に長いデータベース配列を分配して格納し比較する

3. 動的計画法のハードウェア化

3.1 動的計画法

2 つの配列を比較する最も有効な手法に動的計画法（Dynamic Programming 法，以下 DP 法）がある．本論文で使用する Smith-Waterman 法（以下 SW 法）[4)~6)] もこの手法が元になっている．

図 2 に DP 法の処理を，**図 3** に DP 法で得られる最適経路グラフについてそれぞれ示す．このとき使用される得点とギャップによる損失についてはあらかじめ決定されたものを使用する[17),18)]．

DP 法の計算時間について考えると，比較する 2 本の配列の長さがそれぞれ M と N のとき計算時間は $O(MN)$ となる．ホモロジー検索では複数個（k）のデータベース配列を問合せ配列と比較するので，計算時間は最終的に $O(kMN)$ となりその計算時間は非常に膨大なものとなる．そこで計算時間短縮のためにハードウェアを用いて並列処理することが考えられる．

このとき，ソフトウェアによる逐次処理とハードウェ

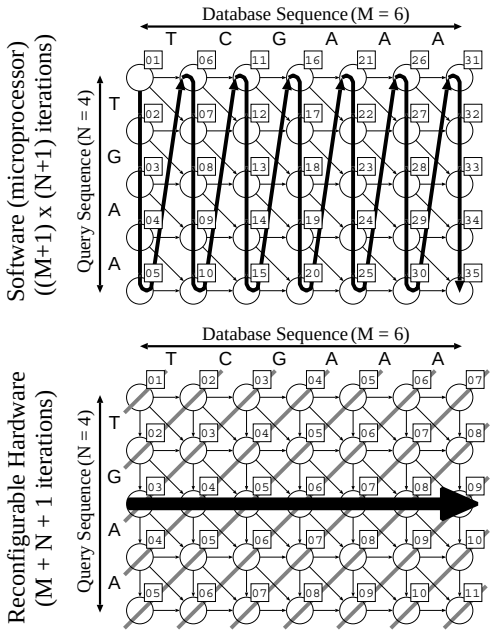

図 4 逐次処理と並列処理
Fig. 4 Serial processing and parallel processing.

アによる並列処理の違いを**図 4** に示す．図 4 上はソフトウェアによる逐次処理を，図 4 下はハードウェアによる並列処理をそれぞれ示している．また図 4 において格子状に配置された ‘O’ は比較処理（する場所）を表している．

ソフトウェアによる逐次処理では（図 4 上），‘O’ の右肩にある数字に比例した計算時間（図 4 中では 35）が必要となる．一方でハードウェアによる並列処理では（図 4 下），斜線方向に同時に計算することができるため計算時間を大幅に短縮することができる．これにより計算時間を $O(mn)$ から $O(m+n)$ まで減らすことが可能となる．

3.2 比較回路

図 4 において格子状に配置された ‘O’ はハードウェアでは比較回路に相当している．ハードウェアで処理する際，この比較回路の個数が多いほど並列度が上がりシステムはより高速に動作する．**図 5** にこの比較回路の構成を示す．

この比較回路は 4 段のパイプラインステージから構成され，各々のステージでは以下に示す処理を行う．

（1 段目）比較する配列の 2 つの要素に対する得点をメモリ（得点行列）から読み込む．
（2 段目）1 段目で得た得点を斜め方向の得点に加える．
（3 段目）縦/横方向の得点にギャップコストを加える．

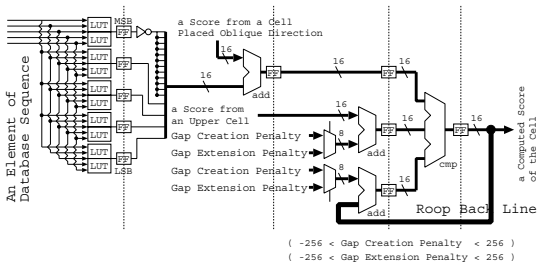

図 5 比較回路
Fig. 5 Processing unit.

（4 段目）縦・横・斜めの中で最大の得点を選択する．

本論文で提案する手法ではアフィンギャップを採用しており，ギャップ挿入時のコスト（Gap Creation Penalty）とギャップ継続時のコスト（Gap Extension Penalty）は 3 段目の選択器により適当なものが加算される（図 5）．これらのギャップコストは使用する得点行列やユーザの要求に応じ，各々“−256〜256” の範囲内で自由に設定することができる．

また FPGA に実装するこの比較回路はデータ通信を考えると PCI バスの動作周波数（33 MHz）以上で動作することが望ましい．一方で動作周波数を高くするためにパイプラインを深くすると比較回路の回路規模が増加するため並列度が低くなる．このトレードオフを考慮した結果，本論文では比較回路のパイプライン段数を 4 段とした．

3.3 マルチスレッド処理の採用

本論文の比較回路では，動作周波数の高速化のために 4 段のパイプライン処理を採用した．この回路において 4 段目の出力を 3 段目の入力へループバックさせる必要があるため（図 5），2 clock に 1 clock の割合で回路が idle となる．そこでこの無駄を避けるために本論文ではマルチスレッド処理を採用した．本論文で採用したマルチスレッド処理について**図 6** に示す．

本論文では遺伝子情報のホモロジー検索を目的にしているため，図 6 中の M は一般に遺伝子データベース配列の大きさを，N は問合せ配列の大きさを，p はハードウェア化により得られる並列度を，w はスレッド数（N/p）を各々表している．また遺伝子情報処理においては一般に $M \gg N$，$N > p$ であるため以下これを仮定する．

まず，図 6 上においては，マルチスレッド処理が行われていないため，各列の処理時間は $O(2M)$ となり，また，w 列の処理が必要であるため，処理時間は $O(2wM)$ となる．

次に，図 6 下を用いてマルチスレッド処理について

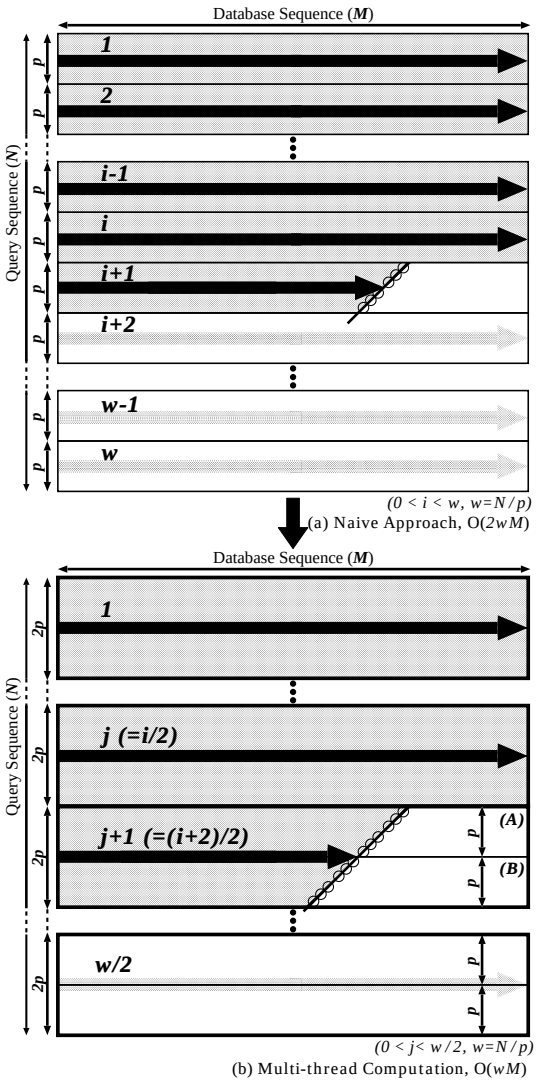

図 6 マルチスレッドによる計算（$M \gg N, N > p$）
Fig. 6 Multi-thread execution ($M \gg N, N > p$).

説明する．図中 (A)，(B) の 2 列のマルチスレッド処理において，まず最初の k clock 間（$k < p$）は (A) 列中の k 個の要素のみが比較され，この間はマルチスレッド処理は行われない．しかし，$p+1$ clock 以降になるとマルチスレッド処理が行われ (A) 列の p 個の要素と (B) 列の p 個の要素が交互に比較されていく．これは，(A) 列の最下段にある p 番目の要素の比較結果を (B) 列の最上段にある $p+1$ 番目の要素に与えることができるようになるためである．このように，2 列のマルチスレッド処理を行うことにより，ハードウェアが idle となることを避けることができる．ただし，図 6 に示したように斜線上の要素に対してマルチスレッド処理を実現しているため，最初の p 個と同じように，最後の p 個の要素に関してもマルチスレッド処理を行うことはできない．

このようなマルチスレッド処理により，計算時間を $O(2wM)$ から $O(wM)$ に短縮することができる．

4. システム詳細

4.1 2 段階検索

遺伝子情報のホモロジー検索において，データベース配列と問合せ配列のホモロジーが高い部分はごく少数である．そのためつねにすべての検索結果が必要なわけではなく，ホモロジーが高い部分の結果を高速に出力することが重要となる．

そこで本論文では，ホモロジーの高い部分を見つける第 1 フェーズ（高速ホモロジー検索）と第 1 フェーズで見つけたホモロジーの高い部分の詳細を出力する第 2 フェーズ（詳細ホモロジー検索）を動的に切り替えることで高速化を実現している．この動的な切替えはユーザの設定する閾値により自動的に実行される．また，ハードウェア（回路）部分の動的な切替えは FPGA によるプログラム実行中動的回路再構成（Runtime Reconfiguration）を用いて実現している[19)~22)]．これら両フェーズの特徴を以下に示す．

- 第 1 フェーズ（高速ホモロジー検索）
  - 閾値を超える得点とその位置のみを出力することで入出力のオーバヘッドを最小化し高速に検索．
  - 性能は FPGA の回路規模に比例．
- 第 2 フェーズ（詳細ホモロジー検索）．
  - 第 1 フェーズの出力でリストアップされた部分のみ検索を実行し最適経路情報を出力．
  - 性能はメモリバンド幅に比例．

第 1 フェーズと第 2 フェーズにおいて，FPGA 上に実装する比較回路はまったく同じものを使用しており，両フェーズの回路の違いは出力部分のみである．しかし，両回路を別の回路として実現することで，各フェーズ（特に第 1 フェーズ）において回路規模，配線量の最適化を図ることができ，より高速な処理が実現可能となる．

第 1 フェーズでは，データベース配列と問合せ配列の比較において，高いホモロジーを持つ部分を「高速に」特定することが重要である．そこで出力を得点情報とその経路の終端部に限ることで，FPGA ボード上のメモリバンド幅のボトルネックを回避し高速に検索できるようにしている．このため FPGA に実装する回路の並列度，すなわち FPGA の回路規模に比例

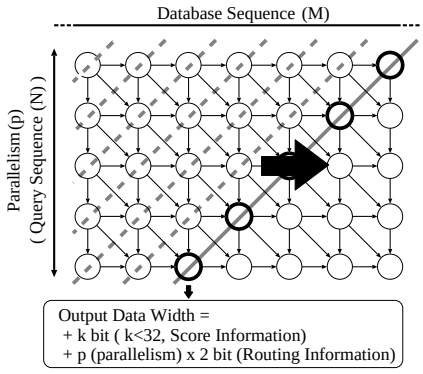

図 7 並列度と出力データ幅
Fig. 7 Relationship between parallelism and data width.

した高速化を得ることができる．

第 2 フェーズでは最適経路情報も含めて出力するため，FPGA ボード上のメモリバンド幅がボトルネックとなる．最適経路情報を出力するために必要とするメモリバンド幅を**図 7** に示す．

最適経路情報を出力すると 1 要素あたり 2 bit（縦，横，斜め）の経路データを必要とするため，得点情報（k bit）に加えて，以下のメモリバンド幅が必要となる．

$$memorybandwidth = k(bit) + 2(bit) * p(parallelism)$$

このとき，回路規模の大きい FPGA を使用して並列度（p）を上げてもメモリバンド幅が使用する FPGA ボードにより決定されているためメモリバンド幅以上の高速化を得ることはできない．

本論文で評価した環境（RC1000-PP [23] 使用時）では，並列度 p を 16 とし出力メモリバンド幅を 96 bit/clock に制限することでパイプラインを止めずに演算を実行できるようにした．

ここから第 1 フェーズと第 2 フェーズの切替え方式について述べる．第 1 フェーズと第 2 フェーズの切替えには次の 2 つの方式がある．

（1）第 1 フェーズを実行中に経路の得点が閾値を超えた場合，第 2 フェーズの回路を FPGA にダウンロードし詳細情報を出力する．その後，再び第 1 フェーズをダウンロードしデータベース配列の最後までこの操作を繰り返す．
（2）第 1 フェーズを最初に実行し，その後リストアップされたデータに基づいて第 2 フェーズを実行する．

このときの概要を**図 8** に示す．

（1）は，より素早くホモロジーの高い部分をユーザに出力することができるが，書き換え時間がオーバヘッドとなるため全体の処理時間は長くなる（図 8 上）．

（2）は，第 1 フェーズの処理を中断しないでデータベース配列を最後まで検索するので，書き換えによるオーバヘッドは最小となり総計算時間は短くなる．しかし，第 1 フェーズが終了するまで検索結果を表示することができないため，ユーザはそれまで待つ必要がある（図 8 下）．

ユーザは目的に応じてどちらの方式で実行するかをあらかじめ選択することができる．

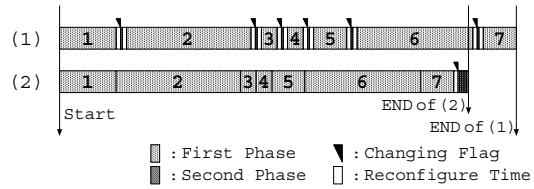

図 8 各フェーズの切替え
Fig. 8 Reconfiguration for swapping circuits.

4.2 データベース配列の分割と高速化

本論文の手法では，第 2 フェーズより第 1 フェーズを高速化することが重要である．これは第 2 フェーズを実際に実行する回数，および計算時間が第 1 フェーズのものと比較すると無視できるほどに小さいからである．

そこで本論文では，FPGA 上でより高い性能を得るために動作周波数を高くし，また，多くの比較回路を実装し並列度を上げるために，演算データ幅を狭くし比較回路を小さくすることで，動作周波数を高くすると同時に並列度を上げることを試みた．

現在のデータベース配列の大きさを考えると演算データ幅は 32 bit であることが望ましい．これは演算データ幅が 32 bit であるならば，最適経路の得点の計算において十分長いデータベース配列に対しても，オーバ（アンダ）フローを起こすことがなくなるからである．しかし，比較回路の回路規模が大きくなるため高い並列度を得ることができないこと，また，動作周波数も低下することを考えると現在の FPGA で実装する方法としては現実的ではない．

一方で，演算データ幅をより狭くすることで比較回路を小さくし，また動作周波数をより高くすることができる．演算データ幅を狭くした場合，データベース配列を分割し，その分割した範囲ごとにホモロジー検索を行うことによりオーバ（アンダ）フローの発生を防ぐことができる．しかしこの場合は，分割された範囲にまたがる部分のホモロジーを正しく保つために，問合せ配列に対して十分な長さ（一般に，問合せ配列

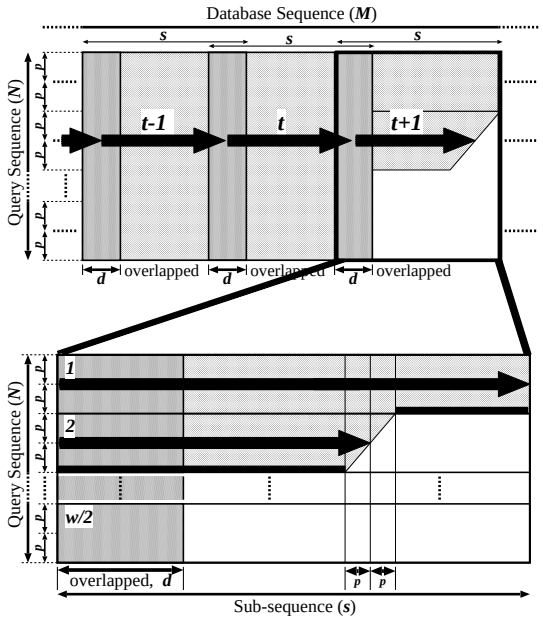

図 9 データベース配列の分割と高速化
Fig. 9 Division of database sequence for first phase.

に対して数倍程度）の領域を重複させる必要がある．このため分割する間隔をあまり小さくすると分割された部分に対し重複する部分に対する処理のオーバヘッド（これらの重複された部分に対しては検索が 2 回実行される）が大きくなり処理速度が低下する．

そこで本論文では演算データ幅を 14，15，16，17，18，19，20 bit としそれぞれの場合において評価を行った．その結果 16，17，18 bit のときが現在の FPGA では最適であることを確認した．最終的に本論文では，Host-PC とのデータ通信，およびメモリバンド幅を考慮して比較回路の演算データ幅を 16 bit とすることにした．

図 9 にデータベース配列（問合せ配列）の分割の方法とその処理についてに示す．図 9 において，データベース配列の長さを M，FPGA の演算データ幅により制限される連続に処理できる長さを s とする．また問合せ配列の長さを N，FPGA で同時に処理できる長さ（並列度）を p とする．このとき一般に $M \gg s$ であり，また，現在最大の回路規模を持つ FPGA を用いても一般に $N > p$ である．

本論文の提案する手法では，初めにデータベース配列を大きさ s の部分配列に分割し，次に先頭の部分配列から順番（..,$t-1, t, t+1$,..）に問合せ配列と比較していく（図 9 上）．

図 9 下は部分配列と問合せ配列の比較を拡大して示したものである．この比較は高速化のために 3.3 節のマルチスレッドを採用している．初めに，最上段の列（図 9 の下，1 列目）が計算され，その計算結果の最下段のデータのみが一時メモリに保存される．これは，次の列の最上段を計算する際に保存したデータが必要になるからである．次に，1 列分の計算が終了した後，比較した問合せ配列につながる次の部分を FPGA に読み込み，再度同じ部分配列と比較を行う（図 9 下，2 列目）．2 列目の計算が進むと一時メモリに保存された 1 列目の計算結果は必要なくなるので，一時メモリは適宜内容が上書きされる．このため，実際にメモリに保存されているデータは図 9 下の 1，2 列目の最下段に示される太い黒線部分となる．この処理を最下段まで繰り返すことで部分配列と問合せ配列の比較を終了する．

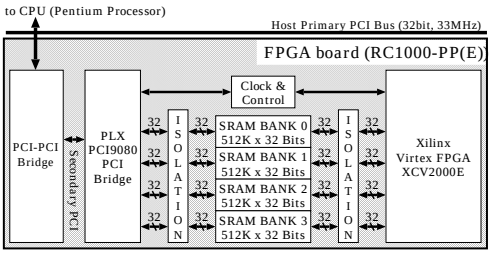

図 10 FPGA ボード（RC1000-PP）の概要
Fig. 10 Block diagram of RC1000-PP.

本論文の提案する手法では，d を重複して検索する区間を示すものとすると，データベース配列あたり $\frac{dMN}{s-d}$ の要素を重複して比較する必要があり（図 9 上），高速検索における速度低下の大きな要因となっている．この重複部分（$\frac{dMN}{s-d}$）は，データベース配列の長さ（M）だけでなく問合せ配列の長さ（N）にも比例して増大する．

5. 性能評価

5.1 Desktop-PC 環境

Desktop-PC 環境において，本論文では RC1000-PP（Celoxica 社製[23]）という FPGA ボードを用いて性能評価を行った．RC1000-PP の概要を**図 10** に示す．この FPGA ボードは 2 MBytes のメモリバンクを 4 個持ち，Host-PC は PCI バスを介して各々のメモリバンクに直接アクセスすることができる．

このとき使用した FPGA は XCV2000E（XILINX 社製[24]）で回路規模は 250 万ゲート相当のものである．そしてこの FPGA は，現在 PCI ボードに組み込まれて市販化されている FPGA の中で最も大きい回路規模を持つ FPGA の 1 つである．また，この FPGA の内部メモリの大きさは 80 KBytes である．

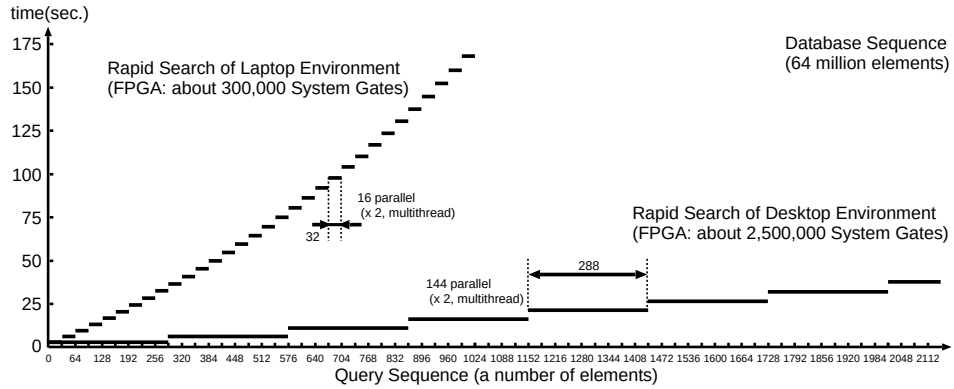

図 11 性能比較（第 1 フェーズ）
Fig. 11 Comparison of each environment in rapid search.

この FPGA には比較回路を 144 個実装することが可能であり，最大で 40 MHz で動作することを確認した（ただし，評価は 33 MHz で行っている）．

第 1 フェーズ（高速検索）における問合せ配列の長さと FPGA のみの計算時間の関係を**図 11** に示す．このときデータベース配列は要素数が約 6,400 万個のものを使用した．性能が 288（＝ 2 * 144）刻みのステップ関数になっているのはシステムがマルチスレッドを採用しているためである．

また通信時間も含めた実測時間は，データベース配列の要素が約 6,400 万個（約 130 MBytes），配列の長さが 2,000 要素のときには約 75 秒である．これは同じデータベース配列を，PentiumIII 1 GHz，物理メモリ 1 GByte，基本ソフトに LINUX（kernel version 2.2.5，gcc-2.91.66）を用いた計算時間（約 11,000 秒）と比較して約 145 倍の速度向上率であった．

第 2 フェーズ（詳細検索）においては，本論文では 16 並列で処理を行っている．これは RC1000-PP において書き込めるデータ幅の最大が 128 bit であるからである．また実行時間は，2,048 要素の問合せ配列と 8,192 要素のデータベースの部分配列の比較を 100 msec 以内に終了するというものであった．

5.2 Laptop-PC 環境

Laptop-PC 環境において，本論文では WILDCARD（Annapolis Micro Systems, Inc. 製[25]）という PC カードを用いた性能評価を行った．この PC カードの概要を**図 12** に示す．この PC カードは 2 MBytes のメモリバンクを 2 つ持っており，これらのメモリバンクをデータの送受信に使用する．

このとき使用した FPGA は XCV300（XILINX 社製[26]）であり回路規模は Desktop-PC 環境で使用した XCV2000E の約 $\frac{1}{6}$ である．またこの FPGA の内部メモリの大きさは 8 KBytes である．

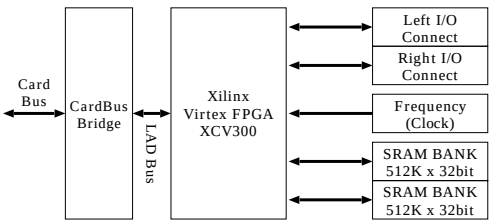

図 12 PC カード（WILDCARD）の概要
Fig. 12 Block diagram of WILDCARD.

本論文ではこの FPGA に比較回路を 16 個実装し，最大 40 MHz で動作することを確認した（ただし，評価は 33 MHz で行っている）．

第 1 フェーズ（高速検索）における問合せ配列の長さと FPGA の計算時間の関係を図 11 に示す．

この Laptop-PC 環境では FPGA の回路規模，およびメモリバンクの容量がホモロジー検索を実行するには小さいため，データベース配列の分割によるオーバラップ区間が大きくなる．このため，問合せ配列の長さにより計算時間が飛躍的に増加してしまう．

問合せ配列の長さが 1,024 の場合，オーバラップ区間の検索に要する処理が全体の 40%以上になるため性能に非常に大きな影響を与える．このため，現在 PC カードに実装されている FPGA やメモリの規模を考えると．問合せ配列の大きさは 1,024 程度が上限であるといえる．

通信時間を含めた実測時間での比較は Laptop 環境においてはまだ実現していない．これはこの PC カードのドライバが Windows98 のものしかサポートされていないからである．

6. おわりに

本論文では遺伝子情報処理におけるホモロジー検索において市販品を用いた高速化システムの有用性について議論した．

その性能は，250万ゲート相当のFPGAをDesktop-PC環境で使用したとき，約6,400万個の要素を持つデータベース配列と2,000要素を持つ問合せ配列の比較を約75秒で終了させるというものであり，これはPentiumIII 1 GHzのDesktop-PCに対して約150倍の性能向上を実現している．

本論文の残された課題として，Host-PCを複数にしたときの評価，および，ユーザが使用しやすい環境作り（GUIの作成）などが残されている．今後はこれらについて研究を進めていく予定である．

謝辞 本研究は，文部科学省科学研究費特定研究（C）ゲノム4領域（ゲノム情報科学），および，日本学術振興会特別研究員奨励費（#5304）の補助による．

参考文献

1) Pearson, W.R.: Searching Protein Sequence Libraries: Comparison of the Sensitivity of the Smith-Waterman and FASTA Algorithms, *Genomics*, Vol.11, No.3, pp.635–650 (1991).
2) Altschul, S.F., et al.: Basic Local Alignment Search Tool, *Journal of Molecular Biology*, No.215, pp.403–410 (1990).
3) Altschul, S.F., et al.: Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, *Nucleic Acids Research*, No.25, pp.3389–3402 (1997).
4) Smith, T.F. and Waterman, M.S.: Identification of common molecular subsequences, *Journal of Molecular Biology*, No.147, pp.195–197 (1981).
5) Gotoh, O.: An improved algorithm for matching biological sequences, *Journal of Molecular Biology*, No.162, pp.705–708 (1982).
6) Myers, E.W. and Miller, W.: Optimal Alignments in Linear Space, *Computer Application in the Biosciences*, Vol.4, No.1, pp.11–17 (1988).
7) 森下真一，久光 徹，高木利久（編）：特集：ゲノム情報科学，情報処理，Vol.43, No.1, pp.1–41 (2001).
8) Fagin, W., et al.: A Special-Processor for Gene Sequence Analysis, *Computer Application in the Biosciences*, Vol.9, No.2, pp.221–226 (1993).
9) Dahle, D., et al.: The UCSC Kestrel General Purpose Parallel Processor, *International Conference on Parallel and Distributed Processing Techniques and Applications*, pp.1243–1249 (1999).
10) Mo, Y., et al.: A Study of GeneWise with the Drosophila Adh Region, *13th Annual Genome Sequencing and Analysis Conference* (2001).
11) http://www.compugen.com
12) http://www.paracel.com/index.html
13) http://www.timelogic.com
14) 末吉敏則，稲吉宏明（編）：特集：やわらかいハードウェア，情報処理，Vol.40, No.8, pp.777–782 (1999).
15) http://www.optimagic.com/boards.html
16) Yamaguchi, Y., et al.: High Speed Homology Search with FPGAs, *Pacific Symposium on Biocomputing 2002*, pp.271–282 (2002).
17) Henikoff, S. and Henikoff, J.G.: Amino Acid Substitution Matrices from Protein Blocks, *Proc. Natl. Acad. Sci. 89*, pp.10915–10919 (1992).
18) Jones, D.T., et al.: The Rapid Generation of Mutation Data Matrices from Protein Sequences, *Computer Application in the Biosciences*, No.8, pp.275–282 (1992).
19) Xilinx, Inc.: *Vietex Series Configuration Architecture User Guide*, Ver.1.5 (2000).
20) Styles, H. and Luk, W.: Customizing Graphics Applications: Techniques and Programming Interface, *2000 IEEE Symposium on Field-Programmable Custom Computing Machines*, pp.77–87 (2000).
21) Simmler, H., et al.: Multitasking on FPGA Coprocessors, *FPL2000*, pp.121–130 (2000).
22) Yamaguchi, Y., et al.: A Co-processor System with a Virtex FPGA for Evolutionary Computation, *FPL2000*, pp.240–249 (2000).
23) Celoxica, Limited.: *RC1000 Hardware Reference Manual*, Ver.2.3 (2001).
24) Xilinx, Inc.: *Vietex^{TM}-E 1.8V Field Programmable Gate Arrays*, Ver.2.2 (2001).
25) Annapolis Micro Systems, Inc.: *WILDCARD^{TM} Reference Manual*, Rev.2.0 (2000).
26) Xilinx, Inc.: *Vietex^{TM} 2.5V Field Programmable Gate Arrays*, Ver.2.5 (2001).

(平成14年1月29日受付)
(平成14年5月10日採録)

山口 佳樹（学生会員）

1975 年生．2000 年筑波大学大学院修士課程理工学研究科理工学専攻修了．同年，筑波大学大学院博士課程工学研究科編入学，日本学術振興会特別研究員．書き換え可能ハードウェアを用いた諸問題の高速化，遺伝子情報処理，および，進化的計算に関する研究に従事．計測自動制御学会学生会員．

宮島 洋介（学生会員）

1979 年生．2002 年筑波大学第 3 学群工学システム学類卒業．同年，筑波大学大学院博士課程工学研究科入学．書き換え可能ハードウェアを用いた遺伝子情報処理システムの研究に従事．

丸山 勉（正会員）

1958 年生．1987 年東京大学大学院工学系研究科情報工学専門課程博士課程修了．工学博士．同年，日本電気（株）入社．並列オブジェクト指向言語，並列遺伝的アルゴリズム，並列マシン Cenju の開発/研究に従事．1997 年より筑波大学機能工学系助教授．書き換え可能なハードウェアを用いた計算の高速化に関する研究に従事．

小長谷明彦（正会員）

1955 年生．1980 年東京工業大学大学院理工学研究科情報科学専攻修士課程修了．同年，日本電気（株）入社．推論マシン CHI，並列マシン Cenju の開発/研究に従事．1987 年米 MIT Laboratory for Computer Scirence 客員研究員．1995 年工学博士．1996 年東京工業大学大学院知能システム科学専攻客員助教授．1997 年北陸先端科学技術大学院大学知識科学研究科教授．2000 年理化学研究所ゲノム科学総合研究センターチームリーダー兼務，2001 年より同センタープロジェクトディレクター兼務．バイオインフォマティクス，知識処理，高性能計算システムの研究に従事．人工知能学会，バイオインフォマティクス学会，ソフトウェア科学会，電子情報通信学会，生物物理学会各会員．