

An Approach for Homology Search with Reconfigurable Hardware*

Yoshiki Yamaguchi¹

yoshiki@darwin.esys.tsukuba.ac.jp

Tsutomu Maruyama¹

maruyama@darwin.esys.tsukuba.ac.jp

Akihiko Konagaya^{2,3}

kona@jaist.ac.jp

¹ Institute of Engineering Mechanics and Systems, University of Tsukuba, 1-1-1 Ten-ou-dai, Tsukuba, Ibaraki 305-8573, Japan

² Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan

³ Japan Riken Genomic Sciences Center, 1-7-22 Suehiro, Tsurumi, Yokohama, Kanagawa 230-0045, Japan

Keywords: genome informatics, homology search, field programmable gate array

1 Introduction

Smith Waterman Algorithm[1] is an efficient and useful algorithm for homology search problems. However, it can not be processed within reasonable time on desktop computer systems, therefore, dedicated hardware systems which is very expensive are used in general.

In this paper, we propose a homology search system with reconfigurable hardware. For reducing the cost, the system is composed of off-the-shelf components. As the result, we could reduce the cost to thousands of dollars and achieved 330 times speedup compared with a desktop computer with a 1GHz PentiumIII. The performance is almost comparable with small to middle class dedicated hardware systems.

2 System Overview

Our system is composed of one off-the-shelf PCI board with FPGA (Field Programmable Gate Array) and one Pentium based computer system (Fig.1). FPGA is a reconfigurable device and any kinds of circuits can be realized on the FPGA in a moment (less than 100 msec in general) by downloading configuration data (namely circuit data) for the circuits from host computers. The configuration data for the FPGA is generated by compiling programs written in hardware description languages using CAD tools for FPGAs.

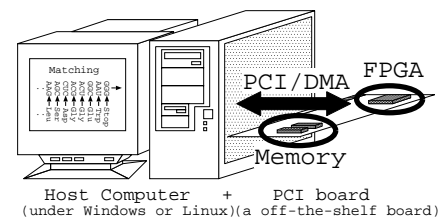


Figure 1: Hardware Platform.

3 Outline of the Homology Search

In our approach, the search consists of two phases in order to make up for the limited hardware resources of off-the-shelf components, and different configuration data are downloaded from the host computer in each phase.

*This work was supported by Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from the Ministry of Education, Culture, Sports, Science and Technology of Japan, and Japan Society for the Promotion of Science (JSPS) Research Fellowships for Young Scientists (#5304).

In the first phase, the query sequence is compared with database sequences by Smith-Waterman algorithm, but only the positions of fragments which are similar to the query sequence (thresholds can be given by the users) are output, because of the following reasons.

1. In general, memory bandwidth of off-the-shelf components is not enough to output all the results by the Smith-Waterman algorithm at the speed the results are generated by the FPGA (at least $2 \times p$ bits memory width is necessary when p elements on the comparison array can be computed at once).
2. The performance is proportional to the number of the elements (p) which can be processed at once. Therefore, it is very important to process more elements at once.
3. The improvement of the memory bandwidth is very slow compared with the improvement of the size of FPGAs.

In this phase, the database sequences are divided into subsequences with fixed size (the size is decided based on the size of internal memory of FPGA), and compared with the query sequence. Suppose that the FPGA can compare p elements at a time, and the length of the query sequence (m) is longer than p ($m > p$). Then, the first p elements of the query sequence are compared with database sequences first, and the intermediate results are stored in the internal memory of the FPGA. Then, the next p elements are compared with the database sequences using the intermediate results. However, the length of each database sequence is very long, and it is impossible to store all the intermediate results in the internal memory of the FPGA. Therefore, each database sequence is divided, and compared with the query sequence.

In this division, the first and the last parts of each subsequence (their length is a few times of the length of the query sequence) are overlapped. These overlapped areas are processed twice, which becomes the major overhead in our approach. The overhead is almost proportional to the length of the overlapped area, and becomes relatively smaller by using larger FPGAs because the internal memory size becomes larger according to the size of the FPGAs.

In the second phase, the detailed results by the Smith-Waterman algorithm for the fragments are output. The performance of this phase is fixed by the memory bandwidth of the FPGA board. However, the performance of this phase is not so important because the computation time of the second phase is much smaller than the first phase.

4 Current Status and Future Works

We have developed the circuits for homology search, and could achieve high performance using off-the-shelf FPGA boards. The performance is almost comparable with small to middle class dedicated hardware systems when we use one board with one of the latest FPGAs (Xilinx XCV2000E). The time for comparing a query sequence of 2048 elements with a database sequence of 64 million elements by the Smith-Waterman algorithm is about 34 sec, which is about 330 times faster than a desktop computer with a 1GHz PentiumIII.

We are now evaluating the performance for the translated nucleotides. When we need to translate the sequences during the comparison, the size of each unit on the FPGA becomes about 10% larger and the parallelism in the first phase will go down to 120 from 144 (about 20% performance down). We are now improving the circuits of the unit to achieve higher performance.

Some parts of the programs for the homology search are still under development, and we also need to improve other parts. We are also developing softwares for parallel processing of the homology search with more number of pairs of FPGAs and host computers connected by Ethernet.

References

- [1] Smith T. F. and Waterman M. S., Identification of common molecular subsequences, *Journal of Molecular Biology*, 147:195–197, 1981.