https://ijrps.com

ISSN: 0975-7538
Review Article

FPGA based parallel computation techniques for bioinformatics applications

Surendar A

Vignan's University, Guntur, Andhra Pradesh, India

ABSTRACT

NEXT Generation sequencing technology is largely improved the development of molecular biology and genomic research. A huge volume of gene data or protein data can be generated with lesser cost, which leads to the exponential growth of existing gene banks or databases. Thus, it becomes a exciting task for conventional algorithms or tools to extract information with genetic significance among these ever increasing databases. There is an urgent need for advanced methods, algorithms, or tools to accomplish these complicated data analysis tasks on a more computationally powerful platform. After decades of development, the FPGA has proved itself in the field of high performance reconfigurable computation. For each generation, one can expect an immediate performance boost with the help of newer manufacturing technologies and a huge amount of volume resources on a single chip, both of which make it a competitive candidate for application acceleration.

Keywords: Bioinformatics; FPGA; Parallel computation techniques; Sequence analysis; Protein structure.

INTRODUCTION

Computational Biology and Bioinformatics are fields involving analysis of a large amount of information of biological origins. Many research areas are involved in this interdisciplinary field, such as, sequence analysis, genome annotation, computational evolutionary biology, and so on. Taking sequence analysis as an example, many aspects of significant biological importance (such as RNA genes, genes that encode proteins, structural motifs, regulatory sequences, and repetitive sequences) can be studied by analyzing the sequence information.

A genome comparison within a species can disclose similarity between protein functions, while the genome comparison among different species can be used to construct the phylogenetic tree. Programs are used daily to analyze sequences greater than 260,000 organisms, which contains more 190 billion nucleotides (Alachiotis et al., 2009) and the databases containing the biological information keep growing rapidly (they double their sizes approximately every 12-16 months). Figure 1.1 shows the growth of the GenBank DNA database and the UniProtKB/TrEMBL databases in terms of the number of base pairs (bps) (Dong et al., 2008). The rapid growth of these databases poses an urgent need to accelerate the computational analyses in a corresponding manner. Another challenge in this field is that the analysis or computation process is very time consuming and sometimes it also requires intensive memory accesses. For example, DNA sequence analysis attempts to identify similar regions between two sequences from a biological point of view, i.e. maximize the number of identical base pairs while keeping the number of differences to a minimum, which requires complex computations. Another example with intensive computation is shotgun sequencing. Thousands of small overlapping DNA fragments (ranging from 35bp to 900bp) are generated which then need to be aligned to form a complete genome

In general, three approaches exist to accelerate the computations involving this large volume of data. The first approach is to design parallel algorithms runni ng on general-purpose multi-core processor based PCs. Multi-core central processing units (CPU) can achieve a speedup factor proportion to the number of cores. Amdahl's law (surendar et al., 2013) gives a speedup estimation based on the number of cores and the fraction of computation that can be parallelized. If the cache within each core is optimally utilized rather than using the much slower main memory, an even better performance can be expected. The difficulty with this approach lies in how to parallelize the algorithm or the data flow to the different cores efficiently. The second approach is to utilize specialized parallel computing platforms. For example, graphic processing units (GPUs), IBM's Cell processors, and field-programmable gate arrays (FPGAs) are all popular platforms suitable for parallel data processing. These parallel platforms can be roughly divided into four groups based on the level of parallelism supported, bit-level parallelism, instruction-level parallelism, data parallelism, and task

* Corresponding Author
Email: phdresearch001@gmail.com
Contact: +91-9962547733
Received on: 13-01-2017
Revised on: 07-04-2017
Accepted on: 13-04-2017

parallelism. The third approach is hybrid computation, a combination of the previous two approaches.

FPGA technology

For conventional computation methods, almost all can be divided into two categories, computation based on hardware circuits and computation based on software programs. A popular example of the first category is application specific integrated circuits (ASICs). ASICs are integrated circuits customized for a particular application rather than intended for general -purpose usage. They can achieve very fast processing speed with high computation efficiency and low power consumption. Despite their obvious advantages, ASICs also have several drawbacks that hinder their utilization in this area. The other category, software program-based computation, is far more flexible. A computation operation is accomplished by executing a set of instructions. By altering the composition of the instructions, the functionality of the system is changed without any modification to the physical structure of the proces sor.

Another platform which provides some of the benefits of both hardware circuits and software programmability is the FPGA. Generally speaking, FPGAs can provide a better performance than the software program, while also achieving a higher degree of flexibility than ASICs, which makes it one of the most widely used reconfigurable devices nowadays. A typical FPGA architecture consists of different types of programmable functional blocks. These include configurable logic blocks (CLBs), embedded multipliers, on-chip memory blocks and programmable I/O blocks. The CLBs can be used to implement various logic operations. The embedded multipliers optimize the multiply operation to improve design performance. The on-chip memory blocks can provide fast memory access. The I/O blocks connect the chip to the peripheral components. All these functional blocks are connected through the programmable routing fabric. In practice, the design tools (e.g. ISE for Xilinx or Quartus for Altera) analyze the design described using HDLs, and decompose it into different basic functional blocks targeted to a specific FPGA device. The design tool will also determine the optimal interconnect path for all these blocks. The same FPGA device can be programmed repetitively, which provides a great flexibility on design modification on FPGAs.

FPGA technology has a number of advantages in terms of performance, time to market, and cost (vimalkumar et al., 2017).

Performance: as the only limitation on the number of "processing cores"on an FPGA is the physical size, designers can improve a design's performance by simply duplicating the circuits with the same functionality and then execute them in parallel. Fine-grained pipelining and massive parallelism take full utilization of the "inherent parallel" attribute of circuits. Meanwhile, current high speed I/O blocks attached to the FPGA chip can further improve a design's throughput.

Time to market: the flexibility that FPGAs provide shortens the prototype time. New designs and architectures can be implemented and tested on the same FPGA chip without going through the long fabrication stage of an ASIC design. Furthermore, a large number of commercial IP cores can save a great portion of the development time and provide guaranteed performance at the same time.

Cost: the non-recurring engineering (NRE) cost for FPGA design is much smaller than that of an ASIC design. The requirement of a digital system might change over time and the customer would like to make incremental changes to the functionality. For ASIC solutions, this means a completely new design, as it is impossible to change the inner structure of an ASIC; however, for FPGA solutions, it is possible to make minor changes and download new programming data (a bitstream) to the FPGA chip. While an individual FPGA is relatively costly, for small to medium volumes, this cost is negligible compared to that of ASIC.

Xilinx and Altera are the two major vendors in the FPGA industry. Together, their products have over 80% market share. Like other computing devices, FPGA products have evolved over time. For example, the Xilinx Virtex FPGA family (from Virtex-2 to Virtex-7) is shown in Table 1.

Table 1 shows that, from Virtex-II to Virtex-7, the volume of slices and on-chip memory has increased dramatically. Note that the Virtex-5 FPGA reports fewer slices than the Virtex-4 device. This is because a different slice organization has applied since Virtex-5 (i.e. each slice in a Virtex-5 FPGA contains 4 6-input lookup tables (LUTs) and 4 flip-flops; in contrast, the earlier generation slice contains 2 4-input LUTs and 2 flip-flops). A LUT is one of the basic components in an FPGA, used to implement combinational logic as truth tables. As more resources are available on a single FPGA chip, we can design and implement more complicated applications on a single (vimalkumar et al.,2017) FPGA device, which leads to greater improvements in the FPGA's computation capability.

Bioinformatics is an interdisciplinary research field that uses computational methodologies or algorithms in computer science to analyze information gained from biological origins or to solve particular biology problems. Since it is closely related to computer science, tools developed for different bioinformatics applications are usually built on general-purpose processors. Along with the rapid growth of biological datasets, these software-based tools are facing greater pressure to process this large volume of data. FPGAs can be a competitive platform to solve some of the biological problems. In fact, we have already seen many FPGA applications in various research areas in Bioinformat-

Table 1: Xilinx Virtex-series specifications*

	Virtex-II	Virtex-II pro	Virtex-4	Virtex-5	Virtex-6	Virtex-7
Slices	46,592	44,096	89,088	51,840	118,560	178,000
BRAM(Kbits)	3,024	7,992	6,048	10,368	25,920	67,680
Max I/O	1,108	1,164	960	1,200	1,200	1,100
DSPslices	N.A	N.A	96	192	864	3,360
Freq(MHz)	650	1050	1028	1098	1098	1818

* The Virtex-II series specifications refer to the XC2V8000, XC2VP100, XC4VLX200, XC5VLX330, XC6VLX760, and XC7VX1140T, respectively. The devices chosen here are the largest designs in their family. The frequency value listed is the toggle frequency for CLB switching of each device at the lowest speed grade.

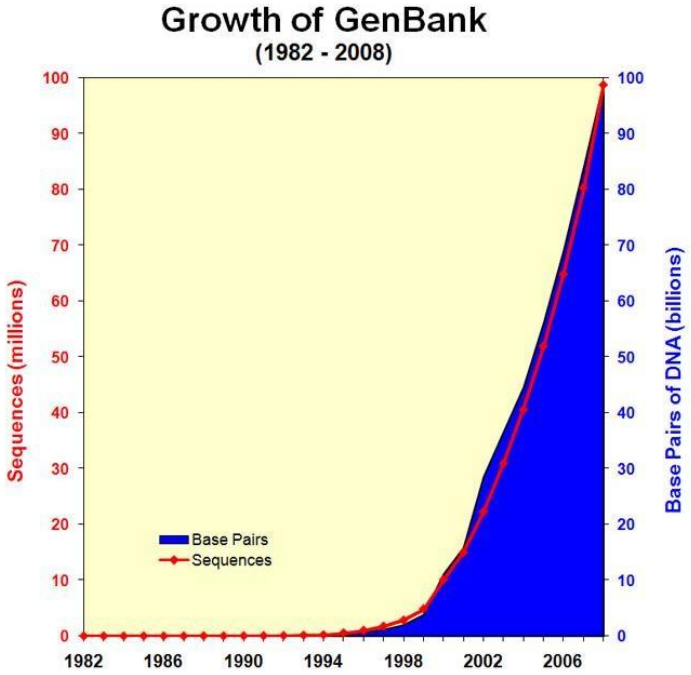

Figure 1: Growth of databases [Dong et al., 2008]

ics. Some of these are briefly described in the following sections.

FPGA Based Bioinformatics Applications

Sequence analysis using FPGA

In Biology, DNA sequences are analyzed to find out motifs, repetitive sequences, regulatory sequences, and RNA genes. As the original data is presented using characters, sequence analysis can be viewed as the fundamental method to understand the relationship between different sequences, which also has a significant influence on other research areas in Bioinformatics. For eg, a FPGA based framework for pair-wise sequence alignment (surendar et al.,2016) has been designed, which can be parameterized to support different alignment algorithms, such as the Smith-Waterman algorithm (surendar et al.,2016) and the Needleman-Wunsch algorithm (93).Over 100 processing elements (PEs) are implemented to improve the performance of pair-wise comparison. This design reports a 62× speedup compared to the equivalent software implementation running on a general purpose PC. The Convey GraphConstructor (CGC) (surendar et al., 2014) is a commercial product designed for sequence assembly using both FPGAs and general-purpose CPUs. It optimizes the memory architecture for the intensive random memory access and shortens the computation time with the help of multiple FPGAs. CGC achieves a 6.2× speedup and reduces RAM consumption by 76%. An FPGA-based coprocessor to solve the gene identification problem was designed in (surendar et al., 2013) based on the Glimmer algorithm (Xia et al., 2011). It presents a tree structure to implement the computation intensive part of the algorithm on the FPGA and achieves a 2.37× speedup compared to the CPU program. The process of finding protein motifs is accelerated in FPGA in (surendar et al., 2016) by utilizing Hidden Markov Models (HMMs) (Friedman et al., 2000). The HMMer engine implemented on the FPGA utilizes a systolic array structure to enhance the comparison operations achieving a 190× speedup over the same computation running in a general purpose CPU.

Evolutionary biology using FPGAs

Evolutionary biology studies the evolutionary relationship among different species. Research in evolutionary biology includes phenomena explanation, phylogenetics, genetic architecture, and evolutionary synthesis.

The study of evolutionary biology helps us understand the origin of species and predict possible changes based on existing population models. A co-processor structure is developed by (surendar et al., 2014) to accelerate median-based phylogenetic reconstruction, where the computationally expensive part (the break median core) is accelerated using an FPGA. This design achieves a speedup ranging from 5× to 189× under different test conditions. A di fferent parallel architecture proposed by (Hussain, et al., 2011) accelerates the computation of the phylogenetic maximum likelihood function (Garnier et al., 1978). On-chip DSP resources are utilized to implement the "Basic Cells" of the FPGA core. Compared to the software program running on a single core, this design achieves a speedup of 13.68. Another heterogeneous design (Krogh et al., 1994) that aims to accelerate the phylogenetic likelihood function is built within the Mr Bayes (surendar et al., 2013) framework. This hybrid design implements multiple float-point operations in parallel and provides a 10× speedup relative to the software program. (Jain et al., 2009) proposed a hybrid system to accelerate the computation of the discrete parsimony function. 512 processing units (PRUs) are implemented on a Xilinx Virtex 6 FPGA achieving a speedup factor of 9.65.

Gene expression analysis using FPGAs

Gene expression is a process whereby a gene is synthesized to a functional gene product. Depending on the different types of gene, the functional gene product can be protein or functional RNA. The gene expression process includes several steps, such as transcription, RNA splicing, translation, and post-translational modification. By studying the gene expression, we can understand the function of a particular gene which can then be used to explain the cause of diseases, such as cancer. The inherent parallelism of the computation-intensive Bayesian learning method (surendar et al., 2013) (used for the reconstruction of gene regulatory networks) is explored by (surendar et al., 2013). The reconfiguration capability is utilized to accomplish the network node score computation and the network structure update iteratively on the same FPGA. The performance evaluation demonstrates a speedup of 76 times over the software implementation. A parallel architecture on an FPGA platform (surendar et al., 2016) is introduced to accelerate the K-means clustering algorithm (Krogh et al., 1994) which is widely used for Microarray data analysis. The 5-core FPGA design reports 51.7× speedup and 206.8× energy efficiency.

Protein structure prediction using FPGAs

As an important branch in Bioinformatics, protein structure prediction is used to predict the structure of an amino acid sequence which can then be used to understand the function of the corresponding protein. Protein structure prediction has wide application potential, such as in drug design in the medicine industry or novel enzyme design in biotechnology. For example, a fine-grained FPGA accelerator (surendar et al., 2013) is proposed to compute the Garnier-Osguthorpe-Robson (GOR) algorithm (surendar et al.,2013) for the secondary protein structure prediction. The evaluation experiments demonstrate a speedup factor of 430× over the original algorithm and 110× over the multi - thread software implementation. Meanwhile, its power consumption is only 30% of that of the CPUs. Another CPU-FPGA co-design (surendar et al., 2013) maps the most computation intensive operations (float point arithmetic operations) onto an FPGA to reduce the overall computation time for the protein energy minimization algorithm. For a single FPGA board, it achieves 5 times speedup compared to the software program.

CONCLUSION

Since the FPGA's inception in the early 1980s, its performance has increased dramatically, with this trend likely to continue into the future. The FPGA's outstanding computation capability provides opportunities to solve a range of different kinds of bioinformatics problems. In this investigation, we have presented a brief introduction to some of the existing FPGA-based solutions in the different bioinformatics fields. This brief overview shows that there are many possible avenues where FPGA based parallel computation could be applied. As sequence analysis is fundamental to bioinformatics research, it has received more attention. In fact, sequence analysis also includes many sub-problems which could be investigated.

REFERENCES

Alachiotis, N., Sotiriades, E., Dollas, A. and Stamatakis, A., 2009, May. Exploring FPGAs for accelerating the phylogenetic likelihood function. In Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on (pp. 1-8). IEEE.

Dong, W., Yang, L., Shen, K., Kim, B., Kleter, G.A., Marvin, H.J., Guo, R., Liang, W. and Zhang, D., 2008. GMDD: a database of GMO detection methods. *BMC bioinformatics*, *9*(1), p.260.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D., 2000. Using Bayesian networks to analyze expression data. Journal of computational biology, 7(3 -4), pp.601-620.

Garnier, J., Osguthorpe, D.J. and Robson, B., 1978. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, *120*(1), pp.97-120.

Hussain, H.M., Benkrid, K., Seker, H. and Erdogan, A.T., 2011, June. FPGA implementation of K-means algorithm for bioinformatics application: An accelerated approach to clustering Microarray data. In Adaptive Hardware and Systems (AHS), 2011 NASA/ESA Conference on (pp. 248-255). IEEE.

Jain, A., Gambhir, P., Jindal, P., Balakrishnan, M. and Paul, K., 2009, April. Fpga accelerator for protein structure prediction algorithm. In Programmable Logic, 2009. SPL. 5th Southern Conference on (pp. 123 - 128). IEEE.

Krogh, A., Brown, M., Mian, I.S., Sjölander, K. and Haussler, D., 1994. Hidden Markov models in computational biology: Applications to protein modeling. Journal of molecular biology, 235(5), pp.1501 -1531.

Surendar, A., Arun, M. and Bagavathi, C., 2013. Evolution of Reconfigurabl e Based Algorithms for Bioinformatics Applications: An Investigation. Int. J. Life Sci. Bt & Pharm. Res, 2(4), pp.17-27.

Surendar, A., Arun, M. and Periasamy, P.S., 2014. A parallel reconfigurable platform for efficient sequence alignment. African Journal of Biotechnology, 13(33).

Surendar, A., Arun, M. and Basha, A.M., 2016. Micro Sequence Identification of Bioinformatics Data Using Pattern Mining Techniques in FPGA Hardware Implementation. Asian Journal of Information Technology, 15(1), pp.76-81.

Vimalkumar M.N, Helenprabha K. and Surendar A.,2017, Classification of mammographic image abnormalities based on emo and LS-SVM techniques, Research Journal of Biotechnology,12(1),pp.35 -40.

Xia, F., Dou, Y., Lei, G. and Tan, Y., 2011. FPGA accelerator for protein secondary structure prediction based on the GOR algorithm. BMC bioinformatics, 12(1), p.S5.