Microelectronics Journal 44 (2013) 670–682

Contents lists available at SciVerse ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

Parallel architecture for DNA sequence inexact matching with Burrows-Wheeler Transform

Yao Xin [a], Benben Liu [a], Biao Min [a], Will X.Y. Li [a], Ray C.C. Cheung [a,*], Anthony S. Fong [a], Ting Fung Chan [b]

[a] Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China
[b] School of Life Sciences and Hong Kong Bioinformatics Centre, The Chinese University of Hong Kong, Hong Kong SAR, China

ARTICLE INFO

Article history:
Received 28 September 2012
Received in revised form
20 May 2013
Accepted 22 May 2013
Available online 14 June 2013

Keywords:
DNA sequence alignment
Inexact matching
Burrows-Wheeler Transform
FPGA

ABSTRACT

The Burrows-Wheeler Transform (BWT) based methodology seems ideally suited for DNA sequence alignment due to its high speed and low space complexity. Despite being efficient in exact matching, the application of BWT in inexact matching still has problems due to the excessive backtracking process. This paper presents a hardware architecture for the BWT-based inexact sequence mapping algorithm using the Field Programmable Gate Array (FPGA). The proposed design can handle up to two errors, including mismatches and gaps. The original recursive algorithm implementation is dealt with using hierarchical tables, and is then parallelized to a large extension through a dual-base extension method. Extensive performance evaluations for the proposed architecture have been conducted using both Virtex 6 and Virtex 7 FPGAs. This design is considerably faster than a direct implementation. When compared with the popular software evaluation tool BWA, our architecture can achieve the same match quality tolerating up to two errors. In an execution speed comparison with the BWA *aln* process, our design outperforms a range of CPU platforms with multiple threads under the same configuration conditions.

1. Introduction

DNA sequence alignment is a fundamental task in modern bioinformatics [1]. It has been extensively utilized in both basic and applied biological research. Complete sequencing of an organism makes it possible to explore and understand the valuable information encoded in genomes. In recent years, there has been a shift from the traditional Sanger sequencing approach to Next Generation Sequencing (NGS) technology. NGS, introduced in 2004, has revolutionized the landscape of genomic research [2].

NGS is also called high-throughput or massively parallel sequencing. It is capable of generating hundreds of millions of short sequences in each run. These short sequences vary in length between 20 and 200 DNA bases, which are also called short reads. This recent NGS technology presents a significant challenge for computing the huge volume of data produced by sequencing machines. Traditional alignment tools like the Smith–Waterman [3] and BLAST [4] are not suitable to process huge databases. The extensive adoption of NGS techniques transforms the sequence alignment process into a short-read mapping problem: short reads are mapped onto a reference genome before all subsequent steps in the analysis pipeline take place.

These types of alignment may be exact—tolerating no errors—or inexact—tolerating mismatches or gaps. Given the large number of reads to handle, the last three years have witnessed the rapid development of numerous sophisticated algorithms to tackle the read-mapping problem [5]. These algorithms can be categorized into two major types according to the indexing methods [6]: algorithms based on the hash table (hashing the reads or the reference genome) such as MAQ [7], SSAHA [8], BFAST [9] and algorithms based on Burrows-Wheeler Transform (BWT) such as Bowtie [10], BWA [11] and SOAP2 [12].

In some applications, the BWT-based algorithms require a significantly smaller memory footprint than the hash-indexing methods. Meanwhile, it is efficient and fast for exact matching because of being independent of reference length. Thus, BWT-based methods have been increasingly used in areas of pattern matching and DNA sequence alignment. One representative application is the BWA tool [11], which extends the exact matching to the inexact one.

Due to the fact that the ever-growing amount of sequencing data results in a great computational demand, the general-purpose CPU is not a suitable platform. The attention now has been shifted to parallel platforms, such as multi-core multi-thread processors, graphics processing units (GPUs), and Field Programmable Gate Arrays (FPGAs). Among these parallel platforms, FPGAs can bring more benefits. They provide optimal reconfigurability and

* Corresponding author. Tel.: +852 34429849.
E-mail addresses: yaoxin2@student.cityu.edu.hk (Y. Xin), benbenliu2@student.cityu.edu.hk (B. Liu), biaominhk@gmail.com (B. Min), xyli@ee.cityu.edu.hk (W.X.Y. Li), r.cheung@cityu.edu.hk, xinyaocityu@gmail.com (R.C.C. Cheung), anthony.fong@cityu.edu.hk (A.S. Fong), tf.chan@cuhk.edu.hk (T.F. Chan).

0026-2692/$ - see front matter
http://dx.doi.org/10.1016/j.mejo.2013.05.004

maximum design space to satisfy different levels of parallel granularity. FPGAs are also competitive for future embedded and portable systems because of their low power consumption. The product of MinION—a USB device DNA sequencer—from Oxford Nanopore Inc. [13] indicates the future demand of potable bio-sequence analysis tools.

Due to the challenges imposed by the irregular data access mode, there is only limited literature on the parallel architecture of FPGA [14] or GPU [15–19] based on BWT. The designs in [14,15] are targeting exact matching, while the work in [15–19] supports inexact matching in the GPU. However, there is no FPGA-based hardware architecture for BWT-based inexact matching tolerating both mismatches and gaps. In practice, inexact matching supporting mismatches or gaps has more application value compared with mere exact matching. There may, however, be mismatches or gaps for possible alignments, due to sequencing errors or inherent genetic variations. Variations of human DNA sequences can affect the ways human beings develop diseases and respond to pathogens, drugs and vaccines. Furthermore, the most common genetic variation, the single-nucleotide polymorphism (SNP), will likely be a key enabler in personalized medicine [20].

To explore the possibility of maximal parallelization of inexact matching algorithms, we have designed and implemented a hardware architecture for DNA sequence inexact alignment with FPGA, based on the basic algorithm from [11]. Our architecture is capable of finding all possible suffix array (SA) intervals (introduced in Section 3.2) for each short read, tolerating two errors including mismatches and gaps. The search of SA intervals is the most time-consuming part for BWT-based methods, and we think this process has more potential for parallelization. Thus, our architecture focuses only on this essential task, instead of the complete alignment process. As for the conversion of SA interval to original positions for each occurrence in the reference genome, this task can be completed by software and is not addressed here.

In this paper, we utilize hierarchical tables to enumerate possible conditions when performing an inexact search. The hierarchical table makes the recursive algorithm more suitable for hardware. Instead of a single-base extension (i.e., processing one DNA base in each step), a dual-base consecutive extension can parallelize the search process and reduce the computation burden. Therefore, the hardware parallel architecture in this paper is based on the dual-base extension method and is implemented on Xilinx Virtex 6 (XC6VLX365T) and Virtex 7 (XC7VX980T) FPGA, respectively. Finally, we compare our design with BWA [11]—a state-of-the-art gap-tolerant software tool—executed on different CPU platforms. With the same search options, experimental results show that our architecture is superior in execution speed and can achieve the same search quality compared with the essential *aln* process in BWA. The major contributions of this paper are as follows:

(1) The hardware parallel architecture is presented for BWT-based inexact matching which can handle both mismatches and gaps. This design can be extended to implement algorithms with irregular data accessing mode.
(2) The algorithm has been modified to be suitable for hardware, while maximal parallelism has been exploited to enhance performance. To achieve this, we have utilized a dual-base proceeding method based on BWT. To our knowledge, this procedure has not been used in previous research.
(3) Instead of a search trie, we employ a hierarchical table to enumerate all possible inexact combinations. A search of the literature indicated that this procedure is used for the first time in the present research.

The rest of the paper is organized as follows: Section 2 describes the work relevant to hardware design for DNA sequence alignment. Section 3 introduces the inexact search algorithm with BWT. Section 4 suggests different approaches to implement the recursive algorithm. Section 5 describes our hardware architecture and its components in detail. Section 6 analyzes the resource utilization with different implementation options. Section 7 presents performance evaluation and comparison results. Section 8 summarizes this paper.

2. Related work

The most frequently used techniques for traditional DNA sequence alignment are the Smith–Waterman [3] and BLAST [4]. The FPGA-based hardware architectures for Smith-Waterman usually employ systolic arrays [21,22]. In the case of emerging short-read mapping algorithms, only several hardware architectures have been published. This might be due to the complexity of algorithms and irregular data access mode.

The hardware designs in [23,24] are based on the hash-table method. Relying on a powerful hybrid reconfigurable platform, a complex architecture is constructed in [23]. This architecture is based on the indexing solution of hashing reference genomes in order to obtain more sensitive mapping results. The design finds all the seeds for a single read, locates in the reference where the seeds occur, then performs the Smith–Waterman alignment at these locations. This design is able to considerably accelerate mapping reads to the human genome, as compared with the BFAST [9] or the Bowtie [10] software. Although the procedure can perform inexact matching at very high speed, the hardware cost is relatively high, and the design implementation is quite complex.

BWT-based algorithms are memory footprint efficient compared with hash table based ones without affecting the speed. Even so, the research literature on hardware architecture is limited. In [14], the first FPGA-based hardware architecture with BWT for exact pattern matching is presented. It is an efficient architecture and has a relative good performance for exact matching. However, mere exact matching in DNA sequence alignment plays a limited role. Moreover—because of the larger search space—inexact matching requires exponentially increasing computation time compared with exact matching.

Apart from algorithms using indexing, there is also a kind of architecture based on the brute-force method [25,26]. By using direct comparison, it allows a freely adjustable character-mismatch threshold for alignment between reads and the reference database. This kind of architecture is an efficient example of coarse parallelism, which can scale up easily with FPGA clusters. It is also simple to implement without the need of storing reference genomes in the hardware. However, the speed of the brute-force method depends on the length of the reference, and it does not support gap alignment. Table 1 summarizes the differences among the aforementioned hardware architectures. Our proposed architecture is also included in this table.

Table 1
Comparison of current hardware architectures.

Existing work	[23]	[14]	[25,26]	Proposed
Index method	Hash Table	BWT	No index	BWT
Inexact matching	Supported	No	Supported	Supported
Gap alignment	Supported	No	No	Supported
Complexity	High	Low	Low	Medium
Memory efficiency	Low	High	High	High