Accelerating DNA Sequence Analysis using content-addressable memory in FPGAs

[1]Muhammad Irfan*, [2]Kizheppatt Vipin, and [3]Rizwan Qureshi
[1]Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan
[2]Birla Institute of Technology and Science (BITS), Pilani, India
[3]MD Anderson Cancer Center, The University of Texas, Texas, USA

2023 IEEE 8th International Conference on Smart Cloud (SmartCloud) | 979-8-3503-1350-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SMARTCLOUD58862.2023.00020

***Abstract*—Biological sequence alignment is a widely used technique where the sequence databases are searched to find similar sequence to the input query. In this work we focus on the most popular local sequence alignment algorithm; Basic Local Alignment Search Tool (BLAST). It is a computationally intensive operation, and with exponential growing databases, makes it further complex to execute in real time. Field-programmable gate arrays (FPGAs) provides hardware-like performance and software-like programmability which makes them the ideal candidate for computationally complex tasks. This paper presents a content-addressable memory (CAM)-based implementation of BLAST on FPGA that accelerates the alignment process using concurrent computations. The searching of the input query is performed in parallel across the database sequence to produce the result in one clock cycle. The proposed design is implemented on Xilinx Virtex-7 FPGA device XC7VX690TFFG1761. Results indicate better feasibility and accelerated performance (149-180 MHz speed) compared to the available searching algorithms.**

***Index Terms*—Field-programmable gate array, DNA Sequencing. Bioinformatics, content-addressable memory.**

I. Introduction

Advances in next-generation sequencing technology (NGS), combined with lower wet lab costs, have enabled to capture massive genomic datasets. NGS offes orders of magnitude more data at a much lower cost than conventional Sanger sequencing [1]. Improvement in genome sequencing has lead to doubling of human genome every seven months, a rate faster than *the Moore's law*. Genomics data can be used to find cancer specific mutations, genetic disorders, and other diseases [2]. This data can also be used to identify differences between humans, forensic applications, and to create personalized medicine for different diseases [3].

The method of recognising the nucleotides in a DNA sequence is known as sequencing. The storage requirement for DNA sequencing datasets are enormous. For example, the human genome has around 3.1647 billion DNA base pairs [4]. This large number of DNA data makes it difficult to do its pattern matching efficiently. Innovative computer tools for rapid and effective processing and analysis are necessary to turn the promise of these data into novel biological discoveries [5]. In computational biology and bioinformatics, aligning sequences to identify similarity is a critical and commonly used computational process for biological sequence analysis [6], [7].

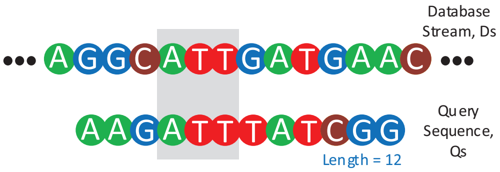

Fig. 1. Matching between a Query and Database sequence.

Sequence alignment is a method of organizing data in such a way that related sequence features are aligned together. In DNA sequence alignment, a sequence can be a database of DNA nucleotides from known sources and another smaller fragment of DNA sequence (called the query sequence) from an known source. The challenge is to determine the portions of the database which are most similar to the query sequence. The sequence alignment problem is a computationally intensive task [7], [8], which takes several minutes to few hours, even on super computing machines. One way to tackle this problem is to model sequence alignment as a string matching problem. Both the database and the query are represented as strings composed of nucleotides and the smaller string is matched against the larger string as shown in Fig. 1.

DNA sequencing introduces unique challenges, such as the small alphabet size and the need for approximate string matching. Special-purpose processors like GPUs and FPGAs can be used to accelerate computationally intensive tasks like string matching, as they offer high levels of parallelism and large memory bandwidths. In this paper we propose to use an FPGA-based solution to accelerate one of the most popular algorithms used for nucleotide sequence alignment called BLAST (basic local alignment search tool).

Content-addressable memory (CAM) works by comparing the input key to all the stored data and generating match lines indicating the presence or absence of the input [9]. In this proposed technique, a CAM-based sequence alignment system stores sequence data and compares the input sequence to all the data at once, generating match lines indicating the presence of the input sequence. The system uses a CAM that is emulated using RAM blocks on an FPGA device. While CAM can be expensive in terms of hardware due to the need for many parallel comparators, it is effective in accelerating searches

979-8-3503-1350-5/23/$31.00 ©2023 IEEE
DOI 10.1109/SmartCloud58862.2023.00020

[10].

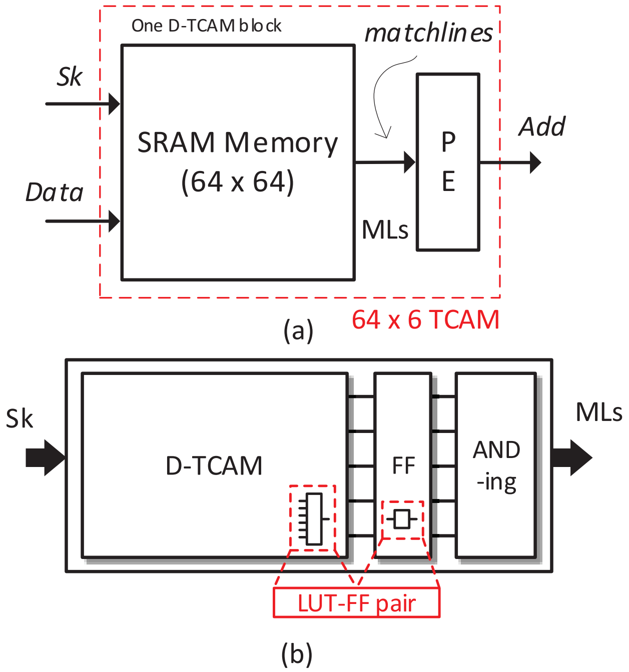

Fig. 2. A building block of D-TCAM. (Sk: Search key, MLs: Match lines, PE: Priority encoder). (a) 64×6 TCAM emulated using 64×64 SRAM memory made of 64 LUTRAMs and a priority encoder. (b) LUT-FF pair used for pipelining to improve the overall throughput.

In CAM-based system that perform the sequence alignment, preprocessing of the input key is performed. The corresponding sequence is retrieved if a successful match is resulted from the input key. The results of this method can then be used to study things like important parts within the sequence or guessing the shape of a protein. This method helps quickly find and pull out sequences from big sets of data, making it a good tool for aligning biological sequences. The value of this method is underscored by its ability to expedite the search and extraction of sequences from large pools of data. As such, it stands out as a particularly effective tool for aligning biological sequences, a task that requires dealing with extensive datasets and demands speed and accuracy.

Emulated FPGA-based CAMs are constructed from different types of FPGA memories, such as Distributed RAM, block RAM, and flip-flops [11]. Our proposed design is based on the D-TCAM which is a distributed RAM-based TCAM of 64x6 basic block as shown in Fig. 2. It provides the searching in a deterministic time with considerable hardware utilization. Key contributions of the proposed work are:

- We proposed an accelerated BLAST algorithm that is based on CAM memory to compute the searching and improve DNA search alignment through parallel processing.
- A novel content-addressable memory (CAM)-based hardware accelerator for BLAST algorithm is proposed.
- The proposed design is scalable, based on the availability of hardware resources in FPGA devices.
- We achieved a high-speed search of 200 bases sequence (12-base DNA) in one clock cycle with 149-180 MHz.
- The hardware utilization for the 200 bases sequence of 12-base. DNA is only 190 and 6, LUTRAMs and BRAMs, respectively.

II. Background and Related work

The available sequencing machines can sequence 50 humans genome a day. However, it takes 1300 hours of CPU time to align and assemble the read sequences with the reference genome [12]. The critical step in the genome analysis is the alignment against the reference stream which is available in the form of a database. The database containing genome sequences itself is growing with an outpacing speed. Accelerating of the alignment process is performed by several works where the FPGA-based systems have shown better performance because of its parallel architecture [13], [14].

The first assembling of the human genome was made possible in 1990 with a cost of 3 million USD that relied mostly on the Sanger sequencing method [15]. It takes longer time because of the few thousand base pairs long reads and is expensive. Later the NGS replaces the long and slow reads with short and parallel reads. A simple sequencing device finishes a single human genome in days while a high-speed Illuminas device can read about 45 genomes a day [16]. The cost, as well as the time, has significantly reduced with time but with the growing size of the genome database, there is a lot to improve in terms of speed to sequence a single genome [17].

Pairwise Sequence Alignment (PSA) is a famous technique for sequence alignment which aligns two sequences using dynamic algorithm [18]. It compares the two sequences and assigns scores based on the match and mismatch. A positive score for the match while a negative for mismatch accumulates to find the highest similarity among the possible alignment sequences. The two classical alignment algorithms are Needleman-Wunsch (NW) which works on the global alignment and Smith-Waterman (SW) which finds the local alignment within the given sequences. They are used to find the optimal alignment between the Database stream (Ds) and Query stream (Qs) as shown in Fig. 3. A resultant H matrix is calculated using the algorithm containing the scores between the two sequences, but these methods are computationally expensive and takes longer time. FPGA-based implementations are developed for the SW algorithm to reduce its complexity from O(mn) to O(m+n), utilizing the parallel architecture. In some cases a VLSI [19] model is developed for the sequence alignment using systolic arrays on FPGA [20] which brings a higher performance compared to the sequential computations on the traditional processors.

Hash tables are an alternative to the PSA which uses a heuristic approach to reduce the computations and get a considerable alignment result. Hash tables have been widely employed for short-read mapping and a variety of other sequence

TABLE I
BINARY REPRESENTATION OF BASES

Base	Notation	Binary
Adenine	A	00
Thymine	T	01
Guanine	G	10
Cytosine	C	11

alignment tasks [21]. It may appear that we have exhausted all potential applications for hash tables in sequence mapping.The BLAST algorithm is utilized to identify analogous segments within biological sequences in both a database and a query [7]. Both the database and the query sequences are denoted using strings of the English alphabet, where each character symbolizes a nucleobase such as Adenine or Thymine. A High Score Pair (HSP) is the term given to each matched pair between the database and the query, and it holds significant value for subsequent biological computations. The BLAST algorithm is composed of three main stages:

- The query is segmented into several smaller parts or words.
- The small words obtained from the query are compared with the database data to identify an exact match.
- At points where the sequences align, the comparison is extended to both sides, and the HSP is computed [22].

III. SYSTEM ARCHITECTURE

A small chunk of the database is stored in the FPGA on-chip memory to speedup the searching operation. Block RAM (BRAM) and distributed RAM (DistRAM) are the two major memory components in modern FPGAs in the form of coarse-grained and fine-grained structure. These memory components are emulated to form a CAM that searches the input typically in one clock cycle, which would otherwise take many clock cycles in random-access memory. 200 bases are stored in BRAM and LUTRAM. 11 base sequence can be detected in one clock cycle as shown in Algorithm 1

For our sequence detector implementation, any of the FPGA RAM blocks, i.e., BRAM and distRAM, can be used. We have implemented our design on both memories and is kept as a design choice on which memory would be available from the other part of the whole design.

The TCAM, we are using in this work is a distributed RAM based TCAM which achieves higher performance by using the LUT-FF pairs of the modern FPGAs. The FFs in the LUT-FF pairs (inside a slice of Xilinx FPGA) are utilized for pipelining the whole design to improve the throughput. We store a portion of the sequence alignment database inside the TCAM which is compared with the input Query stream to perform the match and mismatch.

This scoring function simply assigns a score of 1 for each pair of matching characters at the same position in Qs and s, and a score of 0 for each pair of mismatched characters. The threshold t can be used to filter out alignments with a score that is too low to be considered significant.

Algorithm 1 DNA Sequence Alignment Scoring

```
Input: DNA query string $Qs$, DNA database $Ds$
Output: List of DNA sequences from $Ds$ that align with $Qs$
Initialize empty list $L$
for each DNA sequence $s$ in $Ds$ do
    Compute the alignment score between $Qs$ and $s$ using
    the following scoring function:
        $score(Qs, s) = \sum_{i=1}^{|Qs|} [Qs_i = s_i]$
    where $[\cdot]$ is the indicator function.
    if alignment score meets a chosen threshold $t$ then
        Add $s$ to $L$
    end if
end for
return $L$
```

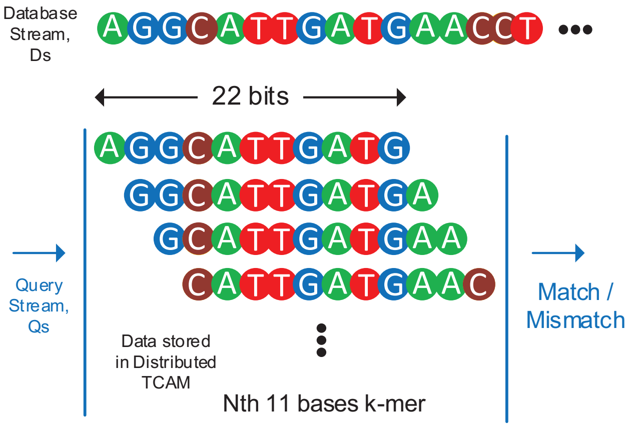

Fig. 3. Matching between a Query and Database sequence.

1) FPGA Implementation: Figure 3 shows how the data is stored in the TCAM where the k-mer data is arranged in such a way that only one base is unique in each location of the TCAM. First 11 bases from 0 to 10 are stored in location 0, second 11 bases from 1 to 11 are stored in location 2, third 11 bases from 2 to 12 are stored in location 3, and so on. We are storing a total of 200 bases in the TCAM memory, so a total of 200 locations are needed and each of 22 bits.

2) Experimental setup: The small chuck of 200 bases is stored in the FPGA-based CAM which took 6 BRAMs of 36k and 190 LUTRAMs (DistRAM). Each base is represented by two bits as show in Table I. The DNA sequence consists of 11 bases making it an input pattern of 22 bits for each search operation. The design is implemented successfully on Xilinx Virtex-7 FPGA device xc7vx690tffg1761 using Xilinx Vivado Design Suite 2019.1 as the developmental tool with speed grade -2. The speed obtained is 180 MHz.

The current setup stores the DNA sequencing data in the local memory, i.e., distributed memory, and serves as a proof

TABLE II
FPGA IMPLEMENTATIONS

Memory Designs	LUTs	LUTRAMs	BRAMs	FFs	Speed (MHz)	Features
Design-I	676	190	12	8	149	4 BRAMs and 1 DistRAM
Design-II	676	190	6	198	180	2 BRAMs and 1 DistRAM
Design-III	865	380	5.5	8	151	1 BRAM and 2 DistRAMs

of concept that the acceleration is possible through the CAM-based searching. We plan to implement the off-chip memory with large amount of data from the DNA database which will be continuously fetching by the CAM-based memory controller to perform the searching operation. The results in Table II shows the resource utilization and corresponding speed. Our design is scalable as the searching can be performed in parallel. Thus multiple memory elements and controllers can combine together to perform the searching of large data simultaneously.

IV. CONCLUSION AND FUTURE WORK

Sequence alignment is a method of arranging DNA, RNA, or protein primary sequences to find regions of similarity that may be the result of functional, structural, or evolutionary links between the sequences, which is a very computational task. In this work, the DNA sequencing acceleration is performed via the FPGA-based CAM. In the future, more data will be stored in the off-chip memory to continuously fetch the DNA sequence and accelerate the search process further.

REFERENCES

[1] O. Harismendy, P. C. Ng, R. L. Strausberg, X. Wang, T. B. Stockwell, K. Y. Beeson, N. J. Schork, S. S. Murray, E. J. Topol, S. Levy *et al.*, "Evaluation of next generation sequencing platforms for population targeted sequencing studies," *Genome biology*, vol. 10, no. 3, pp. 1–13, 2009.

[2] E. D. Pleasance, R. K. Cheetham, P. J. Stephens, D. J. McBride, S. J. Humphray, C. D. Greenman, I. Varela, M.-L. Lin, G. R. Ordóñez, G. R. Bignell *et al.*, "A comprehensive catalogue of somatic mutations from a human cancer genome," *Nature*, vol. 463, no. 7278, pp. 191–196, 2010.

[3] S. Salamat and T. Rosing, "FPGA Acceleration of Sequence Alignment: A Survey," *arXiv preprint arXiv:2002.02394*, 2020.

[4] D. Adjeroh, Y. Zhang, A. Mukherjee, M. Powell, and T. Bell, "Dna sequence compression using the burrows-wheeler transform," in *Proceedings. IEEE Computer Society Bioinformatics Conference*. IEEE, 2002, pp. 303–313.

[5] Y. S. Lee, E.-Y. Chung, Y.-H. Gong, and S. W. Chung, "Quant-pim: An energy-efficient processing-in-memory accelerator for layerwise quantized neural networks," *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 162–165, 2021.

[6] X. Guo, H. Wang, and V. Devabhaktuni, "A systolic array-based FPGA parallel architecture for the BLAST algorithm," *International Scholarly Research Notices*, vol. 2012, 2012.

[7] S. Datta, P. Beeraka, and R. Sass, "RC-BLASTn: Implementation and evaluation of the BLASTn scan function," in *2009 17th IEEE Symposium on Field Programmable Custom Computing Machines*. IEEE, 2009, pp. 88–95.

[8] D. Zoni, L. Cremona, and W. Fornaciari, "All-digital energy-constrained controller for general-purpose accelerators and cpus," *IEEE Embedded Systems Letters*, vol. 12, no. 1, pp. 17–20, 2020.

[9] R. Karam, R. Puri, S. Ghosh, and S. Bhunia, "Emerging trends in design and applications of memory-based computing and content-addressable memories," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1311–1330, 2015.

[10] K. Sai Reddy and K. Vipin, "Opennoc: An open-source noc infrastructure for fpga-based hardware acceleration," *IEEE Embedded Systems Letters*, vol. 11, no. 4, pp. 123–126, 2019.

[11] M. Irfan, H. E. Yantır, Z. Ullah, and R. C. Cheung, "Comp-tcam: An adaptable composite ternary content-addressable memory on fpgas," *IEEE Embedded Systems Letters*, vol. 14, no. 2, pp. 63–66, 2021.

[12] H. Li, "Minimap2: pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018.

[13] A. E. E.-D. Rashed, M. Obaya, H. El, and D. Moustafa, "Accelerating dna pairwise sequence alignment using fpga and a customized convolutional neural network," *Computers & Electrical Engineering*, vol. 92, p. 107112, 2021.

[14] Z. Wu, K. Hammad, E. Ghafar-Zadeh, and S. Magierowski, "Fpga-accelerated 3rd generation dna sequencing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 1, pp. 65–74, 2019.

[15] Y. Turakhia, G. Bejerano, and W. J. Dally, "Darwin: A genomics co-processor provides up to 15,000 x acceleration on long read assembly," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 199–213, 2018.

[16] S. Kumar, K. K. Krishnani, B. Bhushan, and M. P. Brahmane, "Metagenomics: retrospect and prospects in high throughput age," *Biotechnology research international*, vol. 2015, 2015.

[17] D. Fujiki, A. Subramaniyan, T. Zhang, Y. Zeng, R. Das, D. Blaauw, and S. Narayanasamy, "Genax: A genome sequencing accelerator," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2018, pp. 69–82.

[18] K. Katoh, J. Rozewicki, and K. D. Yamada, "Mafft online service: multiple sequence alignment, interactive sequence choice and visualization," *Briefings in bioinformatics*, vol. 20, no. 4, pp. 1160–1166, 2019.

[19] V. Pathak, S. J. Nanda, A. M. Joshi, and S. S. Sahu, "Vlsi implementation of anti-notch lattice structure for identification of exon regions in eukaryotic genes," *IET Computers & Digital Techniques*, vol. 14, no. 5, pp. 217–229, 2020.

[20] A. Cinti, F. M. Bianchi, A. Martino, and A. Rizzi, "A novel algorithm for online inexact string matching and its FPGA implementation," *Cognitive Computation*, vol. 12, no. 2, pp. 369–387, 2020.

[21] W. J. Kent, "Blat—the blast-like alignment tool," *Genome research*, vol. 12, no. 4, pp. 656–664, 2002.

[22] M. Bekbolat, S. Kairatova, A. Shymyrbay, and K. Vipin, "HBLast: An Open-Source FPGA Library for DNA Sequencing Acceleration," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019, pp. 79–82.