# POSTER: BioSEAL: In-Memory Biological Sequence Alignment Accelerator for Large-Scale Genomic Data

Roman Kaplan, Leonid Yavits and Ran Ginosar
Faculty of Electrical Engineering
Technion, Israel Institute of Technology
romankap@gmail.com, leonid.yavits@nububbles.com, ran@ee.technion.ac.il

## I. Introduction

Improvement in genome sequencing technology has led to a reduction in the cost of sequencing and an increase in genomic database sizes, far outpacing Moore's law. An estimated 100 million to 2 billion human genomes could be sequenced by 2025, surpassing other big data aggregators such as YouTube and Twitter [1]. Moreover, it is soon expected that the cost of sequencing a genome will drop below a hundred dollars [2], enabling population-scale genomic datasets. Large genomic datasets can be analyzed in different ways; each provides different insights and most require sequence alignment [1]. Sequence alignment is a fundamental problem in genomics. It aims to find how one sequence could transform to the other by using scores for each of the possible character transformations: insertion, deletion and gap. The resulting score from sequence alignment indicates the functionality of a protein, the distance between organisms in a phylogenetic tree, or the role of a gene [3].

Whole genome alignment is a form of population genomic data analysis, used for genome annotation and phylogeny reconstruction [3]. A single whole genome alignment between human and mouse consumes ~100 CPU hours [4]. By 2025, ~2.5 million species genomes are expected to be available, requiring roughly 50-100 trillion such whole genome alignments, six orders of magnitude more than is possible today in reasonable time.

Searching for similarities in pairs of protein and DNA sequences (also called Pairwise Alignment) has become a routine procedure in computational biology and it is a crucial operation in many bioinformatics tasks. The Smith-Waterman (S-W) [5] local sequence alignment algorithm provides an optimal solution for comparing two biological sequences (protein or DNA). However, it has a high computational complexity of score calculations, $O(n \cdot m)$, where $n$ and $m$ are the lengths of the sequences being compared. Therefore, this algorithm is not commonly used for where multiple sequences must be aligned. Common methods to for sequence alignment, instead of the Smith-Waterman algorithm, are based on heuristic search to reduce the computational complexity to $O(\max(n, m))$. However, these methods are suboptimal in the sense that they may miss the highest similarity sequence [6].

This work presents BioSEAL, a massively parallel processing-in-memory Biological SEquence ALignment accelerator. BioSEAL facilitates associative processing and consists of multiple Resistive Content Addressable (ReCAM) dies, conceptually serving as a large-scale associative processing array. Its main applications include (1) pairwise alignment of long, whole-genome, DNA sequences and (2) alignment of a query sequence with an entire database of sequences, protein or DNA. The paper presents several novelties, as follows:

## II. Novelty 1: Resistive NAND CAM Array

The first is novelty is at the circuit-level. A previously-proposed NOR-based ReCAM [7][8] requires to discharge CAM array rows upon a mismatch. Since ReCAM performs computation with associative processing, the majority of CAM rows discharge on each cycle, which leads to power inefficiency. This work proposes a NAND-based Resistive Content Addressable Memory. The basic bitcell is modified so that charge remains within each array bitcell upon a mismatch, while it discharges upon a match. Figure 1 shows the difference between NAND and NOR ReCAM bitcells, in case of a match or a mismatch. The NAND-based ReCAM leads to improved energy efficiency compared to the NOR-based, with an overall of 27-42% energy reduction for the most common associative processing operations, 2- and 3-bit compares. The lower energy per operation leads to significantly better energy efficiency than other previously published works, when performing large-scale sequence alignment (Section V).

## III. Novelty 2: Batch-Write Associative Processing

Second novelty proposed in the paper is at the functional level of the system. Arithmetic with associative processing is bit-serial, with a constant number of cycles required to execute a single-bit operation. The number of cycles equals twice the number of input bit combinations (first cycle for compare, second cycle for write in the matching array rows). For example, a naïve implementation of 1-bit full addition (three input bits) requires sixteen cycles [9]. In many arithmetic operations the same output value repeats for different inputs. Table 1 shows a reordered full addition truth table where the entries having the same output are placed consecutively. The output combinations '01' and '10' are each repeated three times. By batching multiple writes of the same value, it is possible to reduce the number of execution cycles of

978-1-7281-3613-4/19/$31.00 ©2019 IEEE
DOI 10.1109/PACT.2019.00044

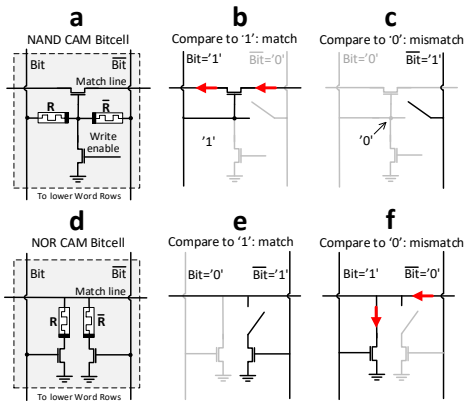


Figure 1: (a,b,c) NAND ReCAM 2T2R bitcell, with the appropriate behavior in case of a match and mismatch. (d,e,f) NOR CAM 2T2R bitcell showing opposite behavior than NAND bitcell.

full 1-bit addition performed with associative processing. In the example of Table 1, four cycles, equal to 25% of the total number of cycles, can be spared.

The paper shows how the repeat of written values can be exploited with a new logic circuit to write a repeating value once, thus saving processing cycles in many arithmetic and logic operations. We call this new approach Batch-Write. Batch-Write leads to higher performance than existing associative processing architecture, with performance benefit between 25-37.5% for most arithmetic and logic operations, and can reach 48.5% for bioinformatics-related operations (e.g., amino acid match score for S-W).

## IV. Novelty 3: Entire Database Associative-Processing Based Sequence Alignment

Previous work [7] has shown how pairwise sequence alignment can be performed with associative processing. This work presents a novel algorithm for aligning an entire database of sequences, DNA or protein, against a query sequence in parallel for all database sequences, instead of executing only pairwise sequence alignment at any iteration. The proposed algorithm uses bit-sized flags, which mark ReCAM rows that are part of the wavefront of S-W at any iteration. Additional flags mark the beginning and end of each of the database sequences, so that the similarity score is attached to the each sequence and thus the highest score is always attached to the highest-similarity sequence.

## V. Performance and Energy Efficiency Comparisons

The paper shows performance and energy efficiency comparison of BioSEAL with other large-scale solutions, which include a CPU-GPU heterogenous system, 128-FPGA system [10], 384-GPU cluster [11] and a previously proposed NOR-based associative processing array [7].

We show that for large-scale DNA sequence alignment, BioSEAL outperforms existing solutions by up to 57x and is more energy efficient by more than two orders of magnitude. For protein database search, BioSEAL achieves up to 6x performance improvement with 10x better energy efficiency.

A preprint of the full paper, containing the full details of compared systems with a complete description of the performance and energy efficiency comparisons, is available at [12].

Table 1: Full-addition truth table reordered for batch-write ('Cmp'=Compare, 'Wr'=Write). Consecutive inputs with the same output have the same background color. On a typical Associative Processor, the operation requires 16 cycles. On BioSEAL, the operation requires only 12 cycles

| Input | | | Output | | Batch-Write |
|---|---|---|---|---|---|
| A | B | $C_{in}$ | $C_{out}$ | S | Cycle Types |
| 0 | 0 | 0 | 0 | 0 | Cmp & Wr |
| 0 | 0 | 1 | 1 | 0 | Cmp |
| 0 | 1 | 0 | 1 | 0 | Cmp |
| 1 | 0 | 0 | 1 | 0 | Cmp & Wr |
| 0 | 1 | 1 | 0 | 1 | Cmp |
| 1 | 0 | 1 | 0 | 1 | Cmp |
| 1 | 1 | 0 | 0 | 1 | Cmp & Wr |
| 1 | 1 | 1 | 1 | 1 | Cmp & Wr |

## VI. References


[1] Z. D. Stephens *et al.*, "Big Data: Astronomical or Genomical?," *PLoS Biol.*, vol. 13, no. 7, 2015.

[2] Illumina, "Illumina Promises To Sequence Human Genome For $100 -- But Not Quite Yet." [Online]. Available: https://www.forbes.com/sites/matthewherper/2017/01/09/illumina-promises-to-sequence-human-genome-for-100-but-not-quite-yet.

[3] J. Harrow *et al.*, "GENCODE: the reference human genome annotation for The ENCODE Project.," *Genome Res.*, vol. 22, no. 9, pp. 1760–74, Sep. 2012.

[4] S. Kurtz *et al.*, "Versatile and open software for comparing large genomes," *Genome Biol.*, vol. 5, no. 2, p. R12, Jan. 2004.

[5] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Mol. Biol.*, vol. 147, no. 1, pp. 195–197, 1981.

[6] A. Backurs and P. Indyk, "Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false)," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing - STOC '15*, 2015, pp. 51–58.

[7] R. Kaplan, L. Yavits, R. Ginosar, and U. Weiser, "A Resistive CAM Processing-in-Storage Architecture for DNA Sequence Alignment," *IEEE Micro*, vol. 37, no. 4, pp. 20–28, 2017.

[8] R. Kaplan, L. Yavits, and R. Ginosar, "PRINS: Processing-in-Storage Acceleration of Machine Learning," *IEEE Trans. Nanotechnol.*, pp. 1–1, 2018.

[9] L. Yavits, S. Kvatinsky, A. Morad, and R. Ginosar, "Resistive Associative Processor," *IEEE Comput. Archit. Lett.*, vol. 14, no. 2, pp. 148–151, Jul. 2015.

[10] L. Wienbrandt, "The FPGA-Based High-Performance Computer RIVYERA for Applications in Bioinformatics," in *Conference on Computability in Europe*, 2014, pp. 383–392.

[11] E. F. de O. Sandes *et al.*, "CUDAlign 4.0: Incremental Speculative Traceback for Exact Chromosome-Wide Alignment in GPU Clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2838–2850, Oct. 2016.

[12] R. Kaplan, L. Yavits, and R. Ginosar, "BioSEAL: In-Memory Biological Sequence Alignment Accelerator for Large-Scale Genomic Data," *arXiv Prepr. arXiv1901.05959*, 2019.