International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 4, Issue 10, pp 20-27, October-2020

https://doi.org/10.47001/IRJIET/2020.410004

Performance Analysis of DNA Sequence Alignment using FPGA's Tools

[1]Rajinder Tiwari, [2]Jamini Sharma, [3]Anna Hakim

[1,2]Dept. of ECE, Model Institute of Engineering & Technology, Jammu, India
[3]Student, Dept of ECE, MIET, Jammu, India

***Abstract* - The DNA data generation rate exceeds its rate of computational processing with the increase in the development of DNA sequencing. Standard sequence alignment techniques using existing computational machines cannot achieve the exponentially growing requirements. Acceleration of the algorithm on FPGA improves the performance in comparison to other platforms. This paper will define and categorize the present sequence alignment algorithms and implement it on FPGA boards. We will also present a comparison of different types of sequence alignment algorithms and surmise the current alternatives and deliver a testimony to advance accelerating sequence alignment on FPGA.**

Keywords: Bioinformatics, FPGA, Hardware Acceleration, Heterogeneous Architectures Sequence Alignment Algorithm, Systolic Array.

I. INTRODUCTION

Bioinformatics is a developing field which aims at expanding computational techniques for assembling, managing and processing biological data guiding us to the basic understanding and breakthrough of the genetic constitution in an organism. Sequence alignment is a significant research attempt in bioinformatics. Sequence alignment examines the resemblances between DNA to evaluate the interrelation between organism and species. The sequence resemblance might also be the outcome of structural or functional connections. Sequence resemblance analysis is facing an alarming issue in current times because the evaluation of data generation is speedily overtaking the standard at which it can be systematically handled.

Tools for bioinformatics research are creating enormous amounts of data that conventional results can't keep pace with in time or space. Delivering an efficacious result for sequence alignment can create an opportunity for enhancing different bioinformatics operations .Bioinformatics research has two ways of providing algorithms that are: Pair-wise and Multiple sequence alignment algorithms. The Pair-wise alignment further has three methods: Dynamic Programming, Heuristic Methods and Hidden Markov Model. These algorithms find the best matching piecewise alignments of two query sequences. The multiple sequence alignment has Hybrid method which is an expansion of pair-wise alignment to process greater than two sequences at a time [1-2].

II. DIFFERENT ALGORITHMS AVAILABLE

a) Dynamic Programming

Dynamic programming offers the exact sequence alignment. Besides, it is additionally usually utilized in the heuristic programming algorithms to channel the arrangement results. Because of the substantial computational load, dynamic programming is dependably tedious. In this way, it is important to figure out how to support their execution. The S-W algorithm is a dynamic programming technique for global pair-wise alignment of two DNA sequences.

The DNA sequences comprises of four nucleotides: adenine (A), guanine (G), thiamine (T) and cytosine (C). These find an optimal alignment between two DNA or protein sequences. In order to utilize this optimal alignment of the DNA, a scoring framework has been implemented to uncover the ideal neighbourhood arrangement between two groupings. From this system of the aligning framework, we obtain the most desired score matrix that plays quite a dominant role in getting the most optimum result of DNA sequence alignment with the use of proper algorithm [3-4].

$$H_{i,j} = \max \begin{cases} 0 \\ H_{i-1,j-1} + w(match/mismatch), \\ H_{i,j-1} - W_k \\ H_{i-1,j} - W_k \end{cases} \qquad (1)$$

The below shown figure 1 shows the proper value of the framework with the location of the DNA or the protein element under observation. Their location can be represented with the help of coordinates in terms of rows and columns i.e. (i, j) of the matrix. The w parameter of the obtained matrix provides the resulting aligning score in terms of proper match or mismatch and the term wk provides the resulting compromise in terms of the gap of the mismatch. The score obtained for each cell of the matrix is computed by using the equations 1 and 2 keeping aside the first row and column that are loaded up with zeros. The arrangement with the top sum score is the ideal alignment. Take the alignment of two DNA sequences as an example using the following scheme [6]:

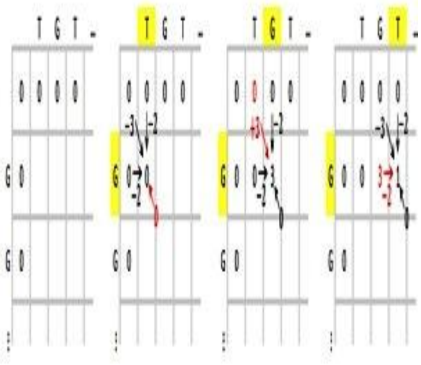


Figure 1: Scoring Framework Matrix using Algorithm [5]

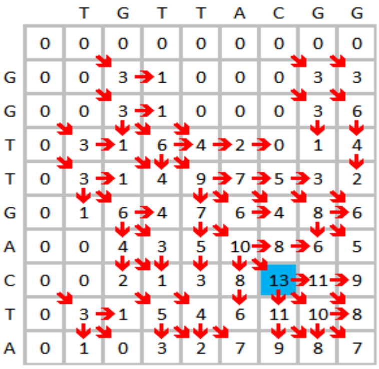
Substitution matrix:

$$w=\begin{cases}+3, & a_i = b_j\\ -3, & a_i \neq b_j\end{cases} \qquad (2)$$

Gap penalty: $W_k = 2k$(a linear gap penalty of$W_1 = 2$)

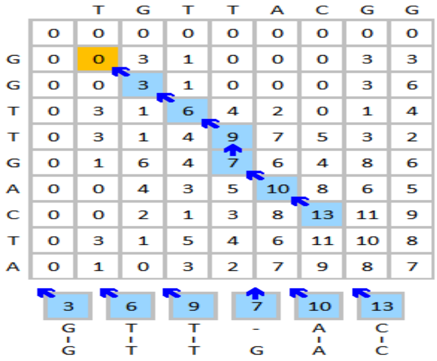
Introduce and fill the scoring framework, appearing in figure 1. This figure demonstrates the scoring strategy for the initial three components. The yellow shading demonstrates the bases being taken. The red shading shows the highest conceivable score for the cell being scored. The outcome scoring grid appears in figure 2. The blue shading demonstrates the highest score.

Note that a component can get score from in excess of one component and each will shape a differing way if this component is followed back. If there arises an occurrence of different most elevated scores, follow back is ought to be done beginning with each most elevated score. The follow back process appears in figure 3. The best nearby arrangement is prepared the regressive way as shown in the below figure 2 [7-10].



		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0	0	3	1	0	0	0	3	3
G	0	0	3	1	0	0	0	3	6
T	0	3	1	6	4	2	0	1	4
T	0	3	1	4	9	7	5	3	2
G	0	1	6	4	7	6	4	8	6
A	0	0	4	3	5	10	8	6	5
C	0	0	2	1	3	8	13	11	9
T	0	3	1	5	4	6	11	10	8
A	0	1	0	3	2	7	9	8	7

Figure 2: Finished scoring matrix (highest score is in blue) [11]



		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0	0	3	1	0	0	0	3	3
G	0	0	3	1	0	0	0	3	6
T	0	3	1	6	4	2	0	1	4
T	0	3	1	4	9	7	5	3	2
G	0	1	6	4	7	6	4	8	6
A	0	0	4	3	5	10	8	6	5
C	0	0	2	1	3	8	13	11	9
T	0	3	1	5	4	6	11	10	8
A	0	1	0	3	2	7	9	8	7

3	6	9	7	10	13
G G	T T	T T	- G	A A	C C

Figure 3: Trace back process for alignment [11]

```
G T T - A C
| | |   | |
G T T G A C
```

Figure 4: Alignment result [11]

In view of the Smith-Waterman algorithm, as shown over, usage on FPGA stage is isolated into four sections: Initialization, Matrix Calculation, Backtrack and Result Generation. Initialization refers to setting up the information sequence into an organization which can be prepared by FPGA. Matrix calculation aims to ascertain the matrix dependent on the algorithm with the end goal to discover the most extreme score. Backtrack intends to discover the best arrangement by following again from the cell with the greatest score. Result generation yields the result in an immediate view.

b) Heuristic programming

The algorithms, for example, BLAST [2] and FASTA [3] are the heuristics in nature that provide quite compromising sequence aligning results with the use of contrast with dynamic programming. On the other hand, this programming technique based algorithm provides a substantially quicker sequence arrangement as compared with the results obtained by using the dynamic programming algorithms. Now, we compare the operating or functional behavior of these two different approaches of programming based algorithms used for DNA sequence alignment. To be sure, rather than breaking down each build-up against each different as in case of dynamic approach, the heuristic algorithms in turns, breaks all the elements of the matrix into small pieces and then provides the results based on comparative analysis.

When arranged in line with the query and scored with a substitution matrix, heuristic method tries to obtain words with length W that score at least T. In the database, the words are expanded which acquire a score T or greater in both directions to discover a ideal un-gapped alignment or HSP (high scoring pair) with a score no less than S or an E value lower than the predetermined threshold. If they do not surpass the cut off merit determined for number of representations or alignments to be addressed; HSPs that fulfil these standards will be reported by BLAST. In initial stage, BLAST uses words with length w instead of k-tuples. These words contain conservative substitutions. The words used in BLAST consist of all w-tuples that get a score T, over a specific level, when looked at utilizing the amino acids substitution matrix.

BLAST utilizes w = 3 and T = 11 by default. A prescribed triplet in the inquiry sequence will then complement the triplets in the database sequence that have a score of at least 11 when the 3 sets of amino acids are compared. [12-15]. For the 2nd stage, using amino acid substitution, the initial words into HSPs are extended by BLAST. It is performed in both directions and is stopped when score falls level below currently found max. score of HSP. [4]. It was observed that 90% of the time was taken to extend word most of which wouldn’t lead to HSP. Most of HSPs had multiple hits, so we would extend a word only if there were multiple hits rather than extending on a single hit. By decreasing threshold T, we sustain probability of finding similarity constant known as Two-Hit alignment method. There are various diverse forms of the BLAST algorithm which are executed on FPGA board which provide quite feasible results, so to differentiate every single conceivable blend of inquiry and database sequences. For each sort of pursuit activity, the designs are indistinguishable. It can be executed on an FPGA board with a mechanism as reflected in the below-given figure 5.

As shown in this figure, this algorithm can be implemented with the help of three inter modular tools i.e. Find Hits, Ungapped Extension, and Gapped Extension. The Find Hits square instates the information sequences, makes words, distinguishes sub-strings in the database which totally coordinate a substring of the question sequence and records the places of those having a correct match. The Ungapped Extension square stretches out, in the two bearings, the exact match to distinguish a couple of longer sequences between the inquiry and the database. The Gapped Extension hinders, the last stage of the technique and utilizes Smith-Waterman to broaden the past result into a gapped arrangement [16].

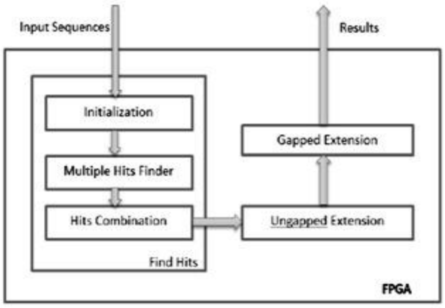


Figure 5: Block Diagram of BLAST on the FPGA platform [17]

c) Hidden Markov Model (HMM)

Although it has been expanded for speech recognition since the 1970s, HMM algorithm is a statistical prototype suitable for various ventures in molecular biology’s like database search and multiple sequence alignment of protein families & their dominions. HMM is used to replicate protein families like globins and kinases. The widely received application of the HMM is as a probabilistic profile of a protein family, called profile HMM. Profile HMM can be created for probing a database for other constituents of the family. The HMMER creates a protein progression by radiating amino acids as when advances past a movement of states. Every state has a table of amino acids outflow with probabilities indistinguishable to those predefined within a profile depict. There are transition probabilities for going from state to state. Figure 6 indicates topology for a Hidden Markov Model.

While different topologies are utilized, the one demonstrated is perceived in the protein arrangement examination. There are 3 types of states communicated by 3 shapes. In this figure, the 3 squares represent the match states of the elements, the amino acids discharged from the desired shape and essential characteristics of the protein element. These amino acids are equivalent to those in the regular precursor. Diamond stone structures represent inserted states of the matrix and produce the desired amino acids elements due to the phenomenon of insertions. Similarly, the structures represented by the circles are specially designed indicators for representing the silent states of the technique [18-19].

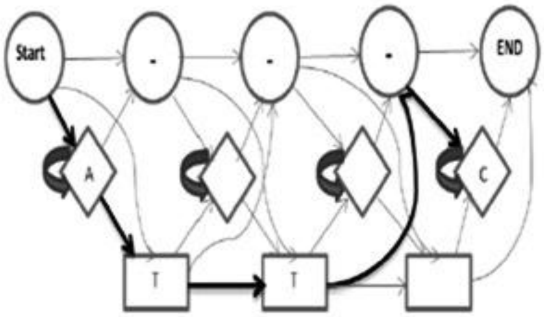


Figure 6: Probabilistic Finite State Machine [19]

III. PROFILING OF SEQUENCE ALIGNMENT ALGORITHMS

In this section, we have put forward the most optimized algorithm so as to obtain the basic requirement of the problem i.e. DNA sequence alignment of the protein element under test for any probable disease or the disorder in the body of the human. This target has to be obtained simply by undergoing the profiling of all the optimum algorithms on FPGA board and the implementation can be carried out with the use of

VHDL or Verilog tool on a PC with desired configurations. From the below figure 7, it has been observed that the time and level of implementation of the algorithms discussed and put forward quite result in oriented ones. Figure 7(a) demonstrates the time usage points of interest of the Smith-Waterman algorithm with their inter-modular arrangements.

In this manner, the significant research exertion can quicken matrix calculation. The contrasts between various executions are additionally centered on the Matrix Calculation part. Figure 7(b) describes the maximum execution time of the BLAST algorithm that has been spent in the initial two blocks more than 99%. Particularly, in the Find Hits block, it assumes the control of about 80% time required by the algorithm. Figure 7(c), shows the level of execution time required by each channel: about 2% of all sequences which will pass first followed by the following channel and about 0.1% sequences that will achieve 3rd channel. Also, the profile obtained from the most optimized tool demonstrates that the first channel possesses 75% of aggregate execution time and the 2nd channel involves 22% [20].

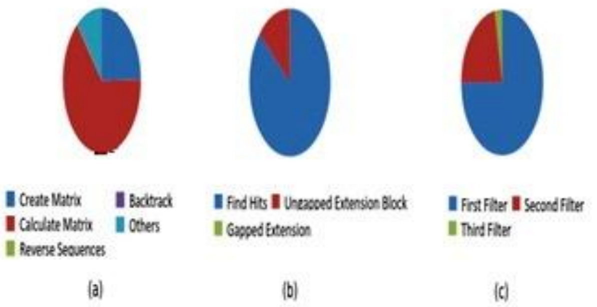


Fig. 7: Time-Consumption by each algorithm in S-W, BLAST and HMM [20]

IV. SPECIALIZED HARDWARE (FPGA) FOR DNA SEQUENCE ALIGNMENT

In the course of recent years, basic computational algorithms, for example, the S-W have been administered on FPGAs and thus have empowered numerous computational investigations considered unserviceable. Numerous issues in computational biology are inherently parallel and aid from coexisting computing models.

a) Parallelism

Due to their adaptive and conventional nature, FPGA logic resources are often much slower than the dedicated logic used to create modern microprocessors. To compete with faster microprocessors, FPGAs must perform operations in parallel. For S-W algorithm, alignment scores can be calculated in parallel. Some computations of single alignment scores can also be performed in parallel. The first level is to align one info sequence against various sequences from a database, by copying the algorithm preparing unit.

The second level depends upon the interior parallelism of algorithms itself. The systolic array is normally connected to achieve this parallelism of algorithms on FPGAs. The systolic array is made of a pipe to organize a course of action of handling component. Each handling component can register and store the information of one another. The sequence alignment algorithms can exploit the systolic array to understand the parallelism. We can see from the Table 1, that the algorithm dependent on dynamic programming and HMM is less demanding to accomplish the better enhancement; even the quantity of PEs isn't as much as the heuristic and HMM method [21].

Table I: Comparative Analysis

S. No	Comparison of State-of-art Implementations of Sequence Alignment on FPGA [17-31]					
	Type	*Platform*	*Frequency*	*PE Number*	*Speed-up*	*Reference design*
1	DP	XC4VLX 160	150MHz	500	172x	AMD Opteron
2	DP	XC5VLX 330	133MHz	512	500x	Intel Q9400
3	DP	XC6VLX 760	133MHz	1024	100x	Intel Q9400 CPU
4	DP	XC2V60 00-4	47.7MHz	252	241x	1.6GHz Pentium 4
5	DP	XC2V60 00-4	47.7 MHz	168	150x	1.6GHz Pentium 4
6	DP	Spartan - 3E	50 MHz	92	92x	1.6GHz Pentium 4
7	DP	EP2S180 -3	66.7 MHz	384	185x	AMD Opteron
8	HR	XC4VLX 160	15 MHz	8	22-40x	2.2GHz Intel Centrino

9	HR	XC5VLX110T	136 MHz	2048	45x	2.8GHz Intel Core i7
10	HR	XC2VP70	145 MHz	2048	45x	2.8GHz Intel Core i7
11	HR	XC4VLX160	-	2048	45x	2.8GHz Intel Core i7
12	HR	EP2S130C5	113 MHz	3072	-	2.6GHz Pentium 4
13	HR	XC4VLX160	189 MHz	-	71x	2.6GHz Pentium 4
14	HR	XC4VLX160	178 MHz	1024	7x	2.6GHz Pentium 4
15	HMM	XC6VLX760	100 MHz	128	250x	Intel Xeon E5520
16	HMM	XC6VLX760	100 MHz	24	60x	Intel Xeon E5520
17	HMM	XC5VLX110T	130 MHz	25	67x	3.2GHz Pentium 4
18	HMM	XC2VP100	100 MHz	90	-	2.8GHz Pentium 4
19	HMM	XC3VP1500	70 MHz	10	30x	AMD Opteron
20	HMM	EP2S180	67 MHZ	85	184x	-
21	HMM	EP1S25F102	76.9 MHz	44	267x	-

b) Sufficient memory

The above analysis is accurate only if the FPGA can estimate and store the entire table of scores in one go. It is unlikely it requires the scores in entire row to be calculated and stored in parallel which in turn limits size of the query. It must hold intermediate data in local memory for FPGA to calculate table of scores [14].

For breaking scores into vertical segments, it must store the last column of a segment in calculating the first column of the next segment. Since only one column must be stored, memory bandwidth necessary will not likely be the limiting factor. Size of memory available, however, will restrict the maximum length of the query and database sequences.

c) FPGA Resource Utilization

These days, FPGAs comprises of huge resources of rationale entryways and RAM blocks. For the bioinformatics applications, for example, sequence alignment, they are both computational and information concentrated applications .Not just the sensible elements require consideration, the Block RAMs on-chip additionally require mulling over. Table 2 demonstrates the research on quickening sequences alignment identified with the FPGA resources utilization.

Further developed FPGA with more resource on-chip can accomplish higher execution. Moreover, other than the resources utilized by the systolic array, the control units additionally partake in FPGA resources. For example, in [24] the systolic array takes 70% of utilized FPGA resource while the control unit takes 26% percentage of utilized FPGA resources. It implies that the assurance of systolic array measure is additionally identified with the control unit and BRAM required.

d) Instruction Efficiency

64-bit processors today have powerful instruction sets. For many simple application calculations (e.g. utilizing 64-bit microchips is pointless, excess and inefficient). FPGAs utilize the negligible logic for given calculations, freeing silicon to exploit parallelism. The basic Smith-Waterman data types are character sequences. Each character is represented by few as two bits definitely lessening the rationale required for every computation.

e) Communication Interfaces

This method requires an adequate bandwidth so as to help the execution of the algorithms properly without any affectivity issues. We suggest utilizing the FPGA board so as to obtain the processing of the data quite fast and with the desired percentage of accuracy which in turn requires proper interfacing between the real-time data through sensors and the processors through the FPGA board. By including additional techniques of the connections, the bandwidth can achieve 1 GB/s which satisfy some usage's specifications. The weakness of the high throughput is a lot of inertness relying upon the way the data are voyaging. With the advancement of USB innovation, the bandwidth of USB 3.0 can reach to 4GB/s. It can likewise effectively fit the necessities of data exchanging for sequence alignment. It gives another choice to exchange data among FPGA and host. For example USB is connected to help the communication among host and FPGA, once in a while, the FPGA-based sequence alignment quickening is utilized as the web server to get administrated remotely. For this situation, Ethernet ends up being an option, since it can give direct access to the Internet. The below-given Table 3 demonstrates the correlation between various algorithms executed on FPGA Platform [22-26].

Table II: Resources Utilization

S.No	**Resource Utilization [17-24]**						
	Type	***Platform***	***PE***	***Slices***	***BRAM***	***Speed up***	***Reference Design***
1	HMM	XC3S1500	10	34%	-	30x	AMD Opteron
2	HMM	XC6VLX760	24	61%	-	60x	Xeon E5520
3	HMM	XC5VLX110T	25	90%	-	67x	Pentium 4
4	HMM	XC3VP1500	85	84%	-	184x	-
5	HMM	XS2VP100	90	98%	-	-	Pentium 4
6	HMM	XC6VLX760	128	97%	-	250x	Xeon E5520
7	DP	EP2S180-3	384	78%	57%	185x	AMD Opteron
8	DP	XC4VLX160	500	81.3%	-	172x	AMD Opteron
9	HR	XC4VLX160	1024	78%	88%	7x	Pentium 4
10	HR	XC5VLX110T	2048	55%	20.3%	45x	Intel Core i7
11	HR	XC2VP70	2048	87%	42%	45x	Intel Core i7
12	HR	XC4VLX160	2048	59.2%	28%	45x	Intel Core i7
13	HR	EP2S130C5	3072	87%	11%	-	Pentium 4
14	HR	XC4VLX160	-	71%	38%	71x	Pentium 4

Table III: Comparative Analysis of Algorithms

S. No	**Comparison between different algorithms**		
	Smith-Waterman	***BLAST***	***Hidden Markov Model***
1	Most accurate sequence alignment	Gives less optimal results	Approximate results.
2	Analyzes fragments of every conceivable length and enhances similarity.	Partitions the sequences into smaller pieces and compares them	Represents a sequence pattern and is well suited for multiple sequence alignments
3	Time consuming	Provides much faster sequence alignment	Time consuming
4	Critical to boost performance	The heuristic algorithm is much faster than other approaches, such as calculating an optimal alignment	HMM algorithm is complicated; however, implementation of FPGA is not too sophisticated.
5	It is useful for scanning databases or when we don't know that sequences are same over their entire lengths	BLAST can find basic qualities in two related species and can be utilized to outline starting with one creature then onto the next	Used for searching sequence databases for sequence homolog's and for making sequence alignments

V. CONCLUSIONS AND FUTURE WORK

This work proposes a state of the art DNA sequence alignment and implementations on FPGA with the comparison. In the wake of presenting the essential standards of sequence alignment algorithms, it's characterization, and profiling, it uncovers normal highlights of various usages by investigating parallelism of algorithm and FPGA utilization. We observe that an FPGA based solution is a promising candidate for sequencing and alignment. Hardware accelerated by FPGA improves the processing time. Therefore FPGAs is a promising candidate for future research in genomic sequencing.

ACKNOWLEDGMENT

The authors are thankful to Prof (Dr.) Ankur Gupta, Director, MIET Jammu, Prof. S K Sharma, Director Academics for their motivation, kind cooperation, and suggestions.

REFERENCES

[1] S. Sarkar, G. R. Kulkarni, P. P. Pande, and A. Kalyanaraman, “Network-on-Chip Hardware Accelerators for Biological Sequence Alignment,” *IEEE Trans. Comput.*59, 1 (January 2010), 29-41.

[2] Ian Korf, MarkYandell and Joseph Bedell, “BLAST,” *O'Reilly& Associates, Inc.* Sebastopol, CA, USA 2003, ISBN:0596002998.

[3] William R. Pearson,” Using the FASTA Program to Search Protein and DNA Sequence Databases, Computer Analysis of Sequence Data: Part I,” *Methods in Molecular Biology*, Vol 24, pp 307-331,1994

[4] Karlin, S., and Altschul, S. F., “Method for assessing the statistical significance of molecular sequence features by using general scoring schemes,” *Proceedings of the National Academy of Science*, USA 87, 2264-2268,

[5] G. D. Forney, “The Viterbi algorithm,” *Proceedings of the IEEE*, Vol. 61, No. 3, pp.268-278, March 1973.

[6] Scott Lloyd, Quinn O.Snell., “Hardware Accelerated Sequence Alignment with Traceback,” *International Journal of Reconfigurable Computing*, Volume 2009, January 2009 Article No. 9

[7] Altera Inc., “Implementation of the Smith-Waterman Algorithm on a Reconfigurable Supercomputing Platform,” *White Paper*. September 2007.

[8] Nawazn, M. Nadeem, H. van Someren, K. Bertels, “A parallel FPGA design of Smith-Waterman traceback,” *International Conference on Field-Programmable Technology (FPT)*, 8-10 Dec.2010.

[9] Kasap, S., Benkrid, K., Liu, Y., “Design and Implementation of an FPGA-based Core for Gapped BLAST Sequence Alignment with the Two-Hit Method,” *Engineering Letters*, Vol. 16, Issue: 3, pp. 443-452,2008.

[10] Jacob, A., Lancaster, J., Buhler, J., Chamberlain, R.D., “FPGA-accelerated seed generation in Mercury BLASTP,” Field-Programmable Custom Computing Machines, 2007. FCCM 2007. *15th Annual IEEE Symposium* on, vol., no., pp.95-106, 23-25 April 2007

[11] Y. Sun, P. Li, G. Gu, Y. Wen, Y. Liu, and D. Liu, “Accelerating HMMer on FPGAs Using Systolic Array-Based Architecture,” *Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, 2009.

[12] K. Benkrid, P. Velentzas, and S. Kasap, “A High-Performance Reconfigurable Core for Motif Searching Using Profile HMM,” *Conference on Adaptive Hardware and Systems, 2008.* AHS '08. NASA/ESA, 2008.

[13] Olaf O. Storaasli, Dave Strenski, “Exploring Accelerating Science Applications with FPGAs,”

[14] Adam Hall, and Short Read, “ DNA sequence Alignment with custom Designed FPGA-based Hardware Dissertation,” *University of British Columbia*, 2011-05

[15] XinyuGuo, Hong Wang, and Vijay Devabhaktuni, “A Systolic Array-Based FPGA Parallel Architecture for the BLAST Algorithm,” *ISRN Bioinformatics*, vol. 11, 2012

[16] Xin Chang1, Fernando A. Escobar, Carlos Valderrama, and Vincent Robert, “Exploring Sequence Alignment Algorithms on FPGA-based Heterogeneous Architectures,” *Proceedings IWBBIO 2014*. Granada 7-9 April 2014

[17] Ying Liu; Benkrid, K.; Benkrid, A.; Kasap, S., “An FPGA-Based Web Server forHigh Performance Biological Sequence Alignment,” AHS 2009. *NASA/ESA Conference on Adaptive Hardware and Systems*, 2009, pp.361-368, July 29 2009-Aug. 1 2009

[18] Dan Zou, Yong Dou, and FeiXia, “Optimization schemes and performance evaluation of SW algorithm on CPU, GPU, and FPGA,” *Concurrency and Computation: Practice and Experience*, Vol 24, pp 1625-1644.

[19] K. Benkrid, Ying Liu, and A. Benkrid, “A Highly Parameterized and Efficient FPGA-Based Skeleton for Pairwise Biological Sequence Alignment,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.17, no.4, pp.561,570, April 2009

[20] Chris Fenton, “Sequence Alignment,” Available Online at http://www.chrisfenton.com/seqalign/

[21] Peiheng Zhang, Guangming Tan, and Guang R. Gao, “Implementation of the Smith-Waterman Algorithm on a Reconfigurable Supercomputing Platform,”

Proceeding of HPRCTA 07, *1st International Workshop on High-performance reconfigurable computing technology and applications*, pp 39-48

[22] Fei Xia, Yong Dou, and Jiaqing Xu, "Families of FPGA-Based Accelerators for BLAST Algorithm with Multi-seeds Detection and Parallel Extension," pp. 43-57, *Springer Berlin Heidelberg*, 2008

[23] Ouano, M.O.L., Jongco, G.F.V.M.G.C.D., and Escabarte, E.B., "FPGA Based AGREP for DNA Microarray Sequence Searching," *Proceedings of International Conference on Computer Engineering and Applications*, 2009, pp 217-221

[24] Xiaoqiang Li, Wenting Han, Gu Liu, Hong An, Mu Xu, Wei Zhou, and Qi Li, "speculative HMMER Search Implementation on GPU," *IEEE 26th International Parallel and Distributed Processing Symposium Workshops and Ph.D.* Forum, CHINA, 2012.

[25] Nathaniel McVicar, Walter L. Ruzzo, and Scott Hauck, "Accelerating ncRNA Homology Search with FPGAs," *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, 2013, pp 43-52.

[26] Y. Sun, P. Li, G. Gu, Y. Wen, Y. Liu, and D. Liu. Accelerating HMMer on FPGAs Using Systolic Array-Based Architecture, in *Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, 2009.

[27] K. Benkrid, P. Velentzas, and S. Kasap, "A High-Performance Reconfigurable Core for Motif Searching Using Profile HMM," *presented at Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on, 2008*.

AUTHOR'S BIOGRAPHIES

Rajinder Tiwari is working as an Assoc. Professor in Department of ECE, Model Institute of Engineering & Technology, Jammu University, Jammu with an experience of 18+ years in the domain of academics, research & industry. The area of research is Quantum Computing with Reversible Logic Gates, Smart System Design. Published papers in premier and reputed journals and presented papers in IEEE/SCOPUS indexed conference in Ireland, USA and Indonesia and associated with reputed international journals as a reviewer and editors.

Jamini Sharma has 18 years of experience in academics and administration. He graduated from Grad IETE Delhi (AMIETE) in 2001 and did his PG from Aligarh Muslim University. Currently he is Head of Department in Electronics and communication In MIET Jammu. His research areas are Compression Techniques, Image processing Techniques.

Citation of this Article:

Rajinder Tiwari, Jamini Sharma, Anna Hakim, "Performance Analysis of DNA Sequence Alignment using FPGA's Tools" Published in *International Research Journal of Innovations in Engineering and Technology* - *IRJIET*, Volume 4, Issue 10, pp 20-27, October 2020. https://doi.org/10.47001/IRJIET/2020.410004

****