Journal of VLSI Design Tools & Technology
ISSN: 2249-474X (Online), ISSN: 2321-6492 (Print)
Volume 9, Issue 2
www.stmjournals.com

Performance Analysis of DNA Sequencing Using Smith-Waterman Algorithm on FPGA

Anna Hakim**, Anam Kashtwari, Rajinder Tiwari, Jamini Sharma
Department of Electronics and Communication Engineering, Model Institute of Engineering and Technology, Jammu University, Jammu, Jammu and Kashmir, India

Abstract

The Smith-Waterman algorithm is the most accurate and optimal alignment algorithm for DNA sequencing. It is utilized to locate the ideal local alignment between two sequences. While looking sequence databases that may contain billions of sequences, this algorithm turns out to be computationally costly and tedious. Smith-Waterman algorithm needs additional memory space and this constrains the extent of a sequence to be aligned. In this study we will implement the S-W algorithm on a FPGA board which will speed up the performance execution of the program with less utilization and more efficiency.

Keywords: *Bioinformatics, sequence alignment, dynamic programming, Smith-Waterman algorithm, FPGA, Pairwise and multiple sequence alignment algorithm, Systolic array*

****Author for Correspondence*** E-mail: anna.112ece15@mietjammu.in

INTRODUCTION

Bioinformatics is a promising field which concentrates on expanding computational methods for compiling, storing and abstracting biological data which leads to the discovery and integral understanding of the genetic constitution in organisms. One of the major research areas in the field of bioinformatics is the sequence alignment. The similarity between DNA sequence of different organisms or species is analysed by the sequence alignment techniques. The first genomic sequencing took 13years with a funding of around 2.1 billion dollars. From 2007–2009 the price decreased from 10 million to 100k and around 14days. Now it costs between $1500 and $1000 and can be done in 26h [1].

Aim of sequence alignment algorihm is to find similarities or dissimilarities between sequences and spot them as soon as possible. Pairwise and multiple sequence alignment algorithm are the two types of sequence alignment algorithm. In pairwise sequence alignment, we have dynamic programming, heuristic programming and hidden Markov model. Multiple sequence alignment algorithm uses hybrid method in which two or more pairwise sequence aignment algorithms are combined.

Dynamic programming is an approach which figures out problematic problems by breaking them down into simpler subproblems. It provides precise sequence alignment. However, it is time consuming due to large computational load. Therefore to boost the performance of sequencing, we use FPGA board. Smith-Waterman algorithm is a type of dynamic programming [2].

SMITH-WATERMAN ALGORITHM

The S-W algorithm is a well known database search algorithm proposed by Temple F. Smith and M S. Waterman in 1981. It is based on the Needleman-Wunsch algorithm. S-Walgorithm finds the most appropriate alignment between the two DNA sequences.

This algorithm takes alignments of any length, at any location, in any sequence, and determines whether an optimal alignment can be accquired. On these calculations, scores or weights are assigned to each nucleobase (A, T, C, G) comparison. Here we assign positive for exact match or substitution, negative for insertion or deletion. In this score matrices, scores are added together and the highest scoring alignment is reported [3–5].

For two sequences *T* and *S*, the length of *S* is *n*, |*S*|=*n*; the length of *T* is *m*, |*T*|=*m*; *V(i, j)* is the ideal alignment score of two sub-sequence *S[1]…S[i]* and *T[1]…T[j]*, calculation of *V(i, j)* is (1) and (2).

$$\text{Initialization: } \begin{cases} V(i.0) = 0, 0 \le i \le n \\ V(0,j) = 0, 0 \le j \le m \end{cases} \quad (1)$$

Recursion relation: $V(i,j) =$

$$max \begin{cases} 0 \\ V(i-1,j-1) + \sigma(S[i],T[j]) & Match/Mismatch \\ V(i-1,j) + \sigma(S[i],-) & Deletion \\ V(i,j-1) + \sigma(-,T[j]) & Insertion \end{cases} \quad (2)$$

Here, a “-” stands for gap or null character; *V(i, 0)* is the result of comparing each character in *S* with a gap in *T*; the definition of *V(0, j)* is the counterpart of comparing each character in *T* with a gap in *S*; and *(S[i], T[j])* is the value of substitution matrix.

The function $\sigma(x,y)$ determines the relative score of match, mismatch, insertion or deletion between the characters x and y.

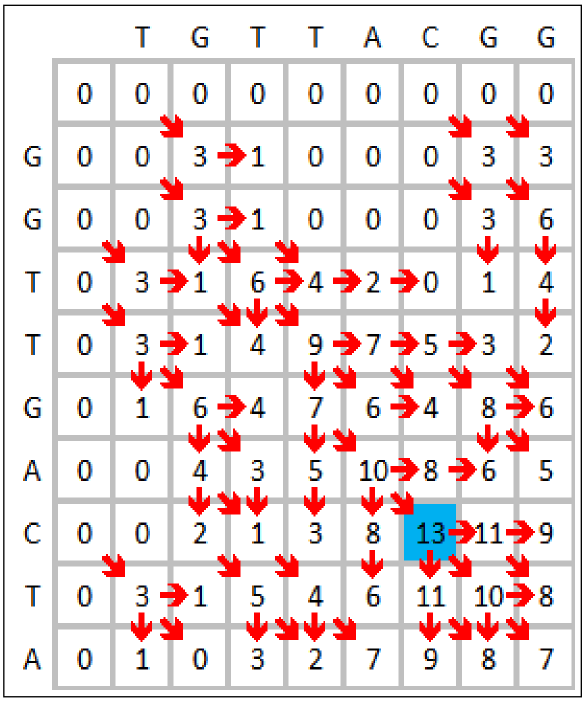
The weighting can be accustomed according to the sequence alignment demands. Once the matrix *V* is complete, we then search the entire matrix *V(i, j)* for the highest value. It marks the start of optimal alignment.

At this point, we back track through the matrix established until we find a zero score which marks the end of the optimal sequence. Take the alignment of two DNA sequences as an example using the following schemes [6, 7].

$$\text{Substitution matrix: } S[i], T[j] = \begin{cases} +3, a_i = b_j \\ -3, a_i \ne b_j \end{cases}$$

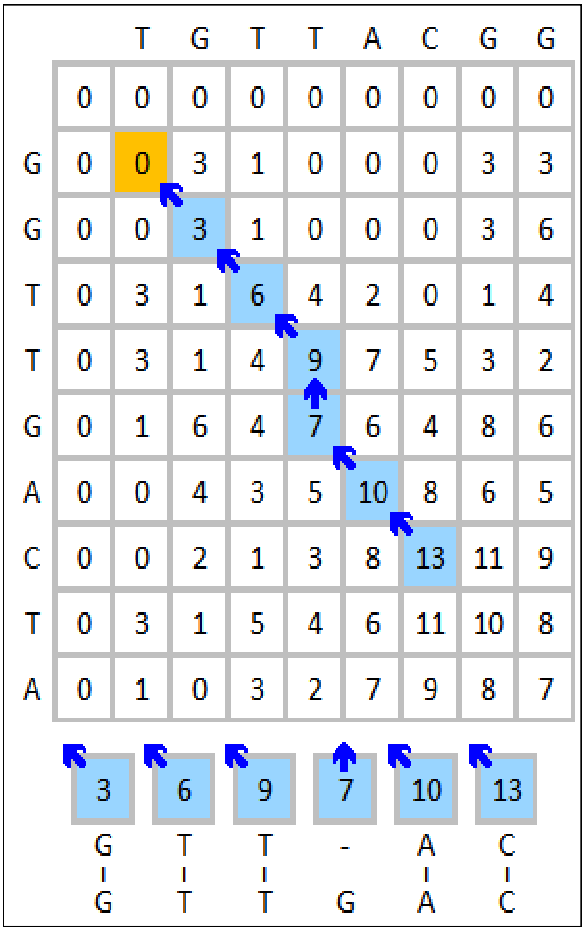
Gap penalty: 2k (a linear gap penalty of 2).

Initialize and fill the scoring matrix according to the algorithm (Figure1). The red color points out highest probable scores for the cell being scored. Blue color shows highest score. An element can receive score from more than one element and each forms a different path if traced back (Figure2).



	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

Fig. 1. *Finished Scoring Matrix (Highest Score is in Blue).*



	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

Fig. 2. *Trace Back Process.*

We start the trace back with the highest score by traversing the entire matrix and continue till the score falls down to the minimum or here 0. Based on the Smith-Waterman algorithm, FPGA platform implementation can be divided into four parts:

- Initialization: prepare input sequence in a format that can be processed by FPGA.
- Matrix calculation: calculate matrix based on Smith-Waterman algorithm to find out maximum score.
- Backtrack: trace back the cell with maximum score to find out the best alignment.
- Result generation: output.

PARALLELISM EXPLORATION BY SYSTOLIC ARRAY

In view of hardware architecture, we use parallel exploration design which results in significant decrease of FPGA resource utilization, which thus allows more parallelism to be exploited from the FPGA. The parallelism of sequence alignment algorithm can be accomplished at two levels: first is to align one input sequence against different sequences from the database at the same time and second depends on the internal parallelism of algorithm itself.

Systolic array is processed to achieve this parallelism of algorithms on FPGA (Figure 3). Systolic array is the layout of processors in an array where data flows synchronously across the array neighbors. Each processing element can store the data of each other. The sequence alignment algorithms take help of systolic array to realize parallelism. The number of PEs adds to the performance [8–10].

In the Figure 3, there are two vector array inputs a and b. The processing cells have a value c_{ij} which is a result due to the algorithm within the cells. Due to less hardware resources, we can only implement limited number of PEs on the FPGA. Thus for calculation of a similarity matrix, we divide the matrix into sub-matrices. In single repetition, the PE array will calculate one sub-matrix, and store intermediate results in memory for the next repetition to use.

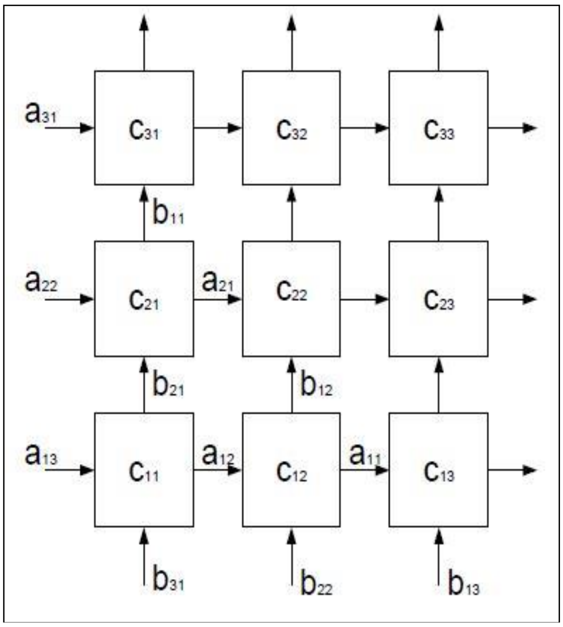


Fig. 3. *Simple Example of Systolic Array Architecture.*

FPGA RESOURCE UTILIZATION

Field Programmable Gate Arrays (FPGAs) is a semiconductor device established over a matrix of configurable logic blocks (CLBs) and is connected via programmable interconnects. FPGAs can be easily adjusted to any operation or functional requirements after the production. The FPGAs comprises of an array of programmable logic blocks, a ranking of reconfigurable interconnects that grant the blocks to be "wired together", that can be inter-wired in different configurations. LUTs can be arranged to execute complex combinational functions, or simple logic gates like AND and XOR. Logic blocks also comprise of memory elements, which may be simple flip-flops or more complete blocks of memory [11].

It is obvious that more the advanced FPGA more resource on chip can accomplish higher performance. Systolic array as well as the control unit takes a portion of FPGA resources, for example systolic array takes 70% of the FPGA resources while control unit takes 26% of used FPGA resources, which means determination of systolic array size is also related to the control unit and block RAMs [1]. Researches of accelerating sequences alignment related to FPGA resources utilization is shown in Table 1.

Table 1: *Resource Utilization.*

Type	Platform	PE	Slices	BRAM	Speedup	Reference Design	Ref
HMM	XC3S1500	10	34%	-	30x	AMD Opteron	[7]
HMM	XC6VLX760	24	61%	-	60x	Xeon E5520	[8]
HMM	XC5VLX110T	25	90%	-	67x	Pentium 4	[9]
HMM	EP2S180	85	84%	-	184x	-	[10]
HMM	XS2VP100	90	98%	-	-	Pentium 4	[11]
HMM	XC6VLX760	128	97%	-	250x	Xeon E5520	[8]
DP	EP2S180-3	384	78%	57%	185x	AMD Opteron	[12]
DP	XC4VLX160	500	81.3%	-	172x	AMD Opteron	[13]
HR	XC4VLX160	1024	78%	88%	7x	Pentium 4	[14]
HR	XC5VLX110T	2048	55%	20.3%	45x	Intel Core i7	[15]
HR	XC2VP70	2048	87%	42%	45x	Intel Core i7	[15]
HR	XC4VLX160	2048	59.2%	28%	45x	Intel Core i7	[15]
HR	EP2S130C5	3072	87%	11%	-	Pentium 4	[16]
HR	XC4VLX160	-	71%	38%	71x	Pentium 4	[16]

Table 2: *Comparison between Our Cell Design and Already Reported Similar Designs.*

Cell Design	Resource Utilization- LUTs	Frequency (MHz)	Performance in Terms of CUPS
Our paper	9	52	78 GCUPS
M Gok *et al*. 2006	13	112.8	54.3 GCUPS
Oliver T. *et al*. 2005	13	55	13.9 GCUPS

EXPERIMENTAL RESULTS

The Smith-Waterman algorithm has been implemented on Zybo board series Zync-7000. The program is written in VHDL code. The PE array working frequency was 52MHz and the peak performance was 78 GCUPS. Comparison is shown in Table 2.

CONCLUSION

We present, in this study, Smith-Waterman algorithm for DNA sequencing alignment, which is based on the technique for identifying common regions in sequences that share local similarity characteristics. We also implemented S-W algorithm with systolic arrays. Since S-W algorithm is computationally expensive for sequencing in large database therefore we accelerate the runtime by implementing it on FPGA board. The results showed us that the implementation of systolic array architecture on FPGA speeds up the performance up to 625x in comparison with a software only implementation. Thus, FPGA based solution is a promising candidate in high performance computing.

ACKNOWLEDGEMENT

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

REFERENCES

1. Xin Chang, Escobar Fernando A, Carlos Valderrama, *et al.* Exploring Sequence Alignment Algorithms on FPGA-based Heterogeneous Architectures. *Proceedings IWBBIO 2014*, Granada. 7–9 Apr 2014.
2. Deepa BC, Nagaveni V. Gene Sequencing Parallelization Using Smith-Waterman Algorithm. *IJARCCE*. Aug 2015; 4(8): 20–23p.
3. Smith TF, Waterman MS. Identification of Common Molecular Subsequences. *J Mol Biol*. 1981; 147(1): 195~197p.
4. Laiq Hassan, Khawaja Yahya M. A Systolic Architecture for the Smith-Waterman Algorithm with High Performance Cell Design. *IADIS European Conference on Data Mining 2008, Amsterdam, The Netherlands, Proceedings.* July 24–26, 2008.
5. Xilinx-Adaptable Intelligent-Devices FPGA and 3D ICs. http://www.xilinx.com
6. Field Programmable Gate Array. http://en.m.wikipedia.org
7. Maddimsetty Rahul P, Jeremy Buhler, Chamberlain Roger D, *et al.* Accelerator Design for Protein Sequence HMM Search.

Proceedings of the 20th annual international conference on Supercomputing, Cairns, Queensland, Australia. Jun 28–Jul 01, 2006.

8. Sun Y, Li P, Gu G, *et al.* Accelerating HMMer on FPGAs Using Systolic Array Based Architecture. In *Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*. 2009.
9. Benkrid K, Velentzas P, Kasap S. 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.
10. Morales Snchez, Jos Luis. *Hardware Design of Algorithm for the Classification of rna Sequences.* Diss. burgerlijkingenieurcomputerwetenschappen. 2006.
11. John Paul Walters, Xiandong Meng, Vipin Chaudhary, *et al.* MPI-HMMER-Boost: Distributed FPGA Acceleration. *J VLSI Sig Proc Syst Signal, Image, and Video Technology*. 2007; 48(3): 223p.
12. Peiheng Zhang, Guangming Tan, Gao Guang R. 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*. 2007; 39–48p.
13. Ying Liu, Benkrid K, Benkrid A, *et al.* An FPGA-Based Web Server forHigh Performance Biological Sequence Alignment. *AHS 2009. NASA/ESA Conference on Adaptive Hardware and Systems*, 2009. Jul 29–Aug 1, 2009; 361–368p.
14. Nathaniel McVicar, Ruzzo Walter L, Scott Hauck. Accelerating ncRNA Homology Search with FPGAs. *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. 2013; 43–52p.
15. Ouano MOL, Jongco GFVMGCD, Escabarte EB. FPGA Based AGREP for DNA Microarray Sequence Searching. *Proceedings of International Conference on Computer Engineering and Applications*. 2009; 217–221p.
16. Xiaoqiang Li, Wenting Han, Gu Liu, *et al.* A Speculative HMMER Search Implementation on GPU. *IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum*, CHINA. 2012.

Cite this Article
Anna Hakim, Anam Kashtwari, Rajinder Tiwari, Jamini Sharma. Performance Analysis of DNA Sequencing Using Smith-Waterman Algorithm on FPGA. *Journal of VLSI Design Tools & Technology*. 2019; 9(2): 8–12p.