High Performance Systolic Array Core Architecture Design for DNA Sequencer

Dayana Saiful Nurdin[1], *Mohd. Nazrin* Md. Isa [1,*] *Rizalafande* Che Ismail[1] and *Muhammad* Imran Ahmad[2]

[1]The Integrated Circuits and Systems Design Group (ICASe),School of Microelectronic Engineering, Universiti Malaysia Perlis, Pauh Putra Campus, 02600, Arau, Perlis, Malaysia.
[2]School of Computer and Communication Engineering, Universiti Malaysia Perlis, Pauh Putra Campus, 02600, Arau, Perlis, Malaysia.

Abstract. This paper presents a high performance systolic array (SA) core architecture design for Deoxyribonucleic Acid (DNA) sequencer. The core implements the affine gap penalty score Smith-Waterman (SW) algorithm. This time-consuming local alignment algorithm guarantees optimal alignment between DNA sequences, but it requires quadratic computation time when performed on standard desktop computers. The use of linear SA decreases the time complexity from quadratic to linear. In addition, with the exponential growth of DNA databases, the SA architecture is used to overcome the timing issue. In this work, the SW algorithm has been captured using Verilog Hardware Description Language (HDL) and simulated using Xilinx ISIM simulator. The proposed design has been implemented in Xilinx Virtex -6 Field Programmable Gate Array (FPGA) and improved in the core area by 90% reduction.

1 Introduction

DNA is made up of thymine (T), guanine (G), cytosine (C) and adenine (A) nucleotides (NTs)[1]. Biological processes such as mutation produce unknown DNA sequences as it alters the natural DNA sequences [2]. Therefore, sequence alignment is important as it facilitates the mutated DNA sequences detection by finding the similarities NT region.

Aligning DNA sequence is an operation that computationally intensive. The task time execution cannot be achieved realistically by desktop computer systems as sequence databases growing exponentially as shown in Fig. 1. Hence, a faster computation platform is needed to overcome the aforementioned problem. Currently, reconfigurable hardware FPGAs has been suggested to enhance the DNA sequence alignment performance [3]. Certainly, FPGAs are attractive platforms to speed up DNA sequence alignment computation compared to General Purpose Processor (GPP).

SA architecture has been introduced by researchers to further improve the DNA sequence alignment process. SA is a pipeline network arrangement where the process task is divided among several processors. It is a build-up of row data processing units called cell or Processing Element (PE). One of the advantages of using SA architecture is that the data is streamed across the array due to the presence of local connection between the cells [4]. The sequence alignment algorithms can take advantage of SA to realize the parallelism on FPGAs.

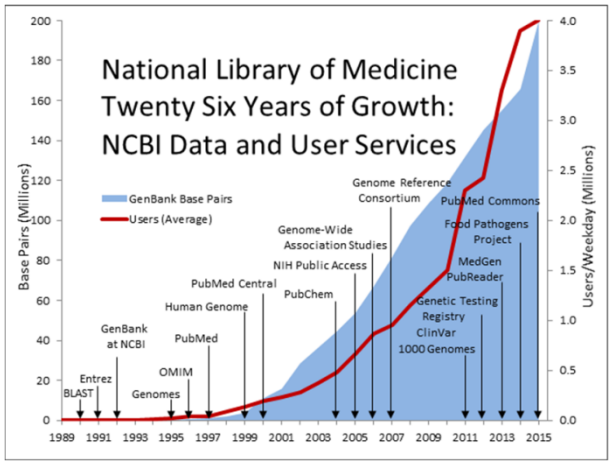

Fig. 1. Database growth from 1989 to 2015 [5].

SW algorithm, a type of Dynamic Programming (DP) algorithm is used in aligning DNA sequence where the common regions between two or more subsequence of DNA sequences represent the optimal sequence alignment [6-7]. DP-based algorithms usually exhaustive due to the accurate analysis. Unlike heuristic sequence alignment such as FASTA, give sub-optimal alignments that help to reduce the computational burden in DP algorithm analysis with uncertain accuracy of the result [8]. Thus, the sequence alignment algorithm can be implemented in SA to speed up its computation time and uses FPGA to improve the DNA sequencer computation process.

* Corresponding author: nazrin@unimap.edu.my

The implementation of FPGA related to DNA sequence alignment architectures are extensively reported in [9], [10], [11], [12], [13] and [14]. All of these architectures are designed based on SA with realization of the SW algorithm with linear gap penalty. The architectures are differentiated based on penalty gap used and also additional features that are included in their designs such additional algorithm for trace back (TB) step. For further information of the related works, please refer to [13].

The remainder of this paper will present the general information of SW algorithm with affine gap, SA and FPGA. Next, comparative timing performance evaluation of the proposed design against other FPGA platforms. Finally, conclusion of this work.

2 Sequence Alignment Algorithms

Pairwise Sequence Alignment (PSA) is one of the alignment analyses to align DNA sequence. It investigates the relationship between a newly discover query sequence and subject sequences that are taken from databases. T. F. Smith and M. S. Waterman have introduce an algorithm in 1981 known as SW algorithm to find the best local alignment between the aforementioned sequences [15]. There are two ways in penalizing insertions and deletions gaps; linear and affine.

A more complex SW algorithm was introduced by Gotoh which is suitable for the affine gaps due to mutation in DNA sequence [16] as shown in (1). Implementation of this algorithm in hardware requires more logic resources and computation time. The time complexity of this algorithm is *O(mxn)* where *m* and *n* are the length of subject and query sequences respectively. Based on (1), *d* is penalty for open gap, *e* is penalty for extended gap and $\gamma(r_i,s_j)$ is a score of substitution matrix for subject sequence r_i ($r_1,r_2,r_3 \ldots r_m$) and query sequence s_j. ($s_1,s_2,s_3 \ldots s_n$).

$$M(i,j) = \max\begin{Bmatrix} I_x(i-1,j-1)+\gamma(r_i,s_j) \\ I_y(i-1,j-1)+\gamma(r_i,s_j) \\ M(i-1,j-1)+\gamma(r_i,s_j) \end{Bmatrix}$$

$$I_x(i,j) = \max\begin{Bmatrix} I_x(i-1,j)-e \\ M(i-1,j)-d-e \end{Bmatrix} \quad (1)$$

$$I_y(i,j) = \max\begin{Bmatrix} I_y(i,j-1)-e \\ M(i,j-1)-d-e \end{Bmatrix}$$

$$F(i,j) = \max\{M(i,j), I_x(i,j), I_y(i,j), 0\}$$

The alignment between each NT pair of *r*, indexed by *i* and *s*, indexed by *j* is computed in the matrix form *F(i,j)*. This matrix computes the three neighbouring cells; the diagonal cell, *M(i,j),* the top cell $I_x(i-1,j)$, and the left cell, $I_y(i,j)$, for the maximum score as shown in Fig. 2.

The boundary values *F(0,0), F(i,0), F(0,j)* are set to zero as there is no alignment in either *r* or *s* as shown in Fig. 4. TB starts after the scores were completely filled the matrix *F(i,j)* by locating the maximum score in as the starting point. The next highest score will be the highest value among its neighbouring cells (top, left, diagonal). This process continues until it reached the origin as shown in Fig 4. TB is used to obtain the optimal alignment between *r* and *s* as shown in Fig. 3.

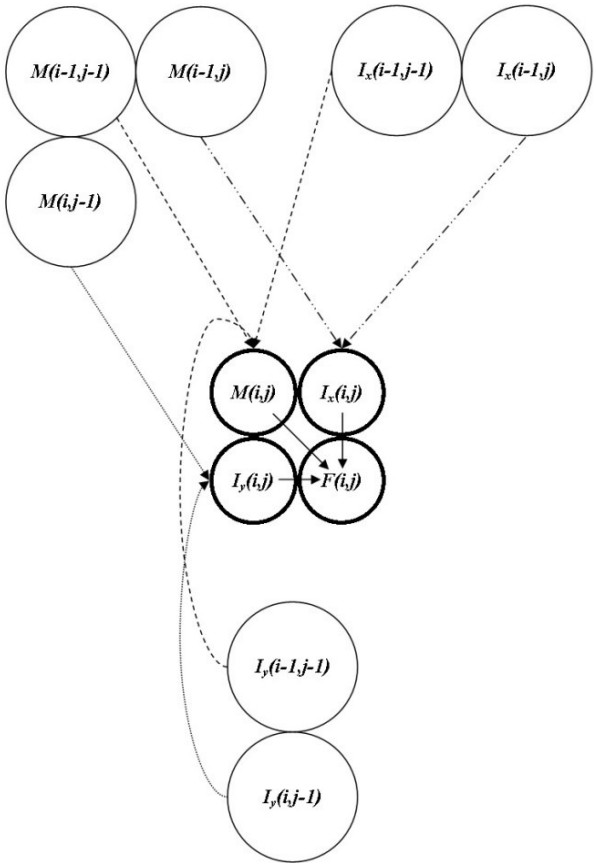

Fig. 2. Affine gap penalty SW algorithm data dependency.

r	–	*T*	*C*	*G*	*T*	*A*
	|	|	|	⋮		|
s	*T*	*T*	*C*	*A*	*T*	*A*
score :	–2	+5	+5	–2	+5	+5 = 16

Fig. 3. Optimal sequence alignment [17].

		T		G		C		T		C		G		T		A	
		0		0		0		0		0		0		0		0	
T	0	5	I_x:-2	0	I_x:-2	0	I_x:-2	5	I_x:-2	0	I_x:-2	0	I_x:-2	5	I_x:-2	0	I_x:-2
		I_y:-2	*M*:5	I_y:-4	*M*:-2	I_y:-6	*M*:-2	I_y:-8	*H*:5	I_y:-9	*H*:-2	I_y:-11	*H*:-2	I_y:-13	*H*:5	I_y:-9	*H*:-2
T	0	5	I_x:-4	3	I_x:-4	0	I_x:-4	3	I_x:-4	3	I_x:-4	0	I_x:-4	3	I_x:-4	3	I_x:-4
		I_y:-2	*M*:5	I_y:-4	*M*:3	I_y:-6	*M*:-4	I_y:-8	*H*:3	I_y:-10	*H*:3	I_y:-11	*H*:-4	I_y:-13	*H*:3	I_y:-11	*H*:3
C	0	0	I_x:-6	3	I_x:-6	8	I_x:-6	0	I_x:-6	8	I_x:-6	1	I_x:-6	0	I_x:-6	1	I_x:-6
		I_y:-2	*M*:-2	I_y:-4	*M*:3	I_y:-6	*M*:8	I_y:-6	*H*:-6	I_y:-8	*H*:8	I_y:-6	*H*:1	I_y:-8	*H*:-6	I_y:-10	*H*:1
A	0	0	I_x:-8	0	I_x:-8	1	I_x:-6	6	I_x:-8	0	I_x:-6	6	I_x:-8	0	I_x:-8	0	I_x:-8
		I_y:-2	*M*:-2	I_y:-4	*M*:-4	I_y:-6	*M*:1	I_y:-8	*H*:6	I_y:-8	*H*:-8	I_y:-10	*H*:6	I_y:-8	*H*:-1	I_y:-10	*H*:-1
T	0	5	I_x:-10	0	I_x:-10	0	I_x:-8	6	I_x:-8	4	I_x:-8	0	I_x:-8	11	I_x:-10	0	I_x:-10
		I_y:-2	*M*:5	I_y:-4	*M*:-4	I_y:-6	*M*:-6	I_y:-8	*M*:6	I_y:-8	*H*:4	I_y:-10	*H*:-10	I_y:-12	*H*:11	I_y:-3	*H*:-3
A	0	0	I_x:-9	3	I_x:-12	0	I_x:-10	0	I_x:-8	4	I_x:-10	2	I_x:-10	0	I_x:-3	16	I_x:-12
		I_y:-2	*M*:-2	I_y:-4	*M*:3	I_y:-6	*M*:-6	I_y:-8	*M*:-8	I_y:-10	*H*:4	I_y:-10	*H*:2	I_y:-12	*H*:-12	I_y:-14	*H*:16

Fig. 4. Alignment matrix of local alignment with affine gap penalty with its optimal sequence alignment, match=+5, mismatch=-2, *d*=-12, *e*=-2 [17]

3 Systolic Arrays

Parallel architectures such as SA have been used in computing technologies to accelerate the algorithm in FPGA [18]. A linear SA consists of an array of PEs with each PE holds one DNA character of a query DNA sequence as shown in Fig. 5. The highest score will be stored in SA after the subject sequence is streamed in. If *r* is related to *s*, the score will be high.

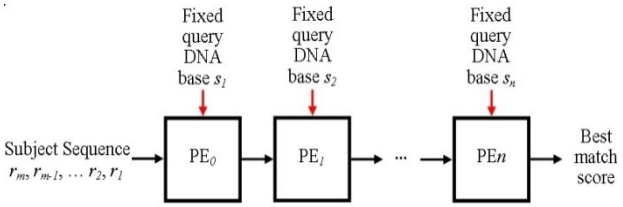

Fig. 5. A linear SA architecture [19].

The internal architecture of PE implement the affine gap penalty SW algorithm as shown in Fig. 6. The computation of the alignment algorithm ise main task of a PE comparing the score in the top, left and diagonal elements which have been described in previous section based on (1).

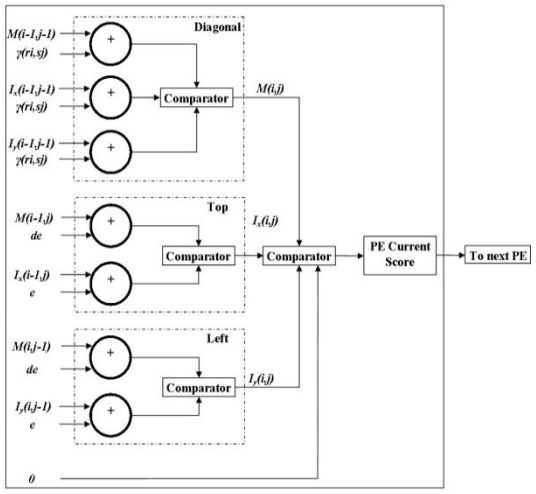

Fig. 6. PE core architecture.

Each PE performs one calculation due to the comparison of a DNA character between the query and subject sequence respectively. In each clock cycle, the PE generates one alignment matrix element. After one clock cycle, a column is generated in each PE where the column size is based on *m*. The execution of full alignment between the two sequences using SW algorithm in *O(m+n-1)* time complexity is illustrated in Fig. 7.

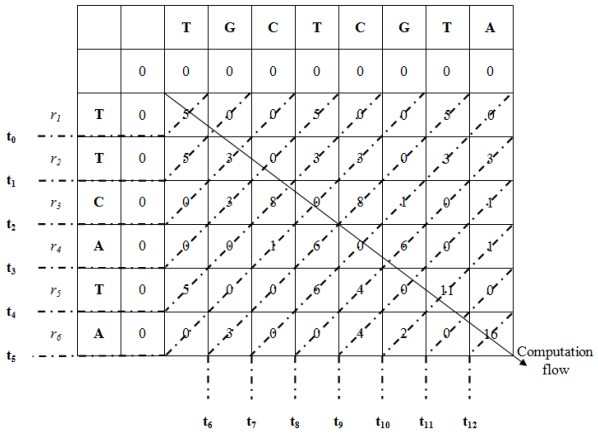

Fig. 7. Diagonal wavefront.

4 FPGA, the accelerator

The affine gap penalty SW algorithm is used as the backbone SA to speed up the computation process in aligning DNA sequence where the FPGA acts as an accelerator. Different FPGA vendors have different FPGA architecture. Altera uses ALMs (Adaptive Logic Module) while Xilinx uses Slices to represent their logic resources [20]. Slices are the combination of Flip Flops (FFs) and Look Up Tables (LUTs). The LUTs can either be 4-input or 6-input depending on devices. Virtex – 4 uses 4-input LUTs whereas Virtex- 6 uses 6-input LUTs. Altera's FPGA devices also uses 4-input LUTs or 6-input LUTs as the building block of ALM. Thus, normalization is required to analyze speed performance

* Corresponding author: nazrin@unimap.edu.my

equally between different logic resources which will be discussed in the next section.

The results were normalized in terms of Slices as the proposed core architecture is implemented in Virtex-6 FPGA. The performance of the core architecture will be evaluated in terms of (Slice / PE), total number of PEs that can be occupied by the device (#PE), peak frequency (Peak Freq), and Peak Cell Update Per Second (Peak CUPS). Slice / PE are used to find the total number of slices that can reside in a PE. #PE can be calculated by dividing the total slices of the device with Slice/PE. Peak Freq is the time in Hertz (Hz) of the longest path between 2 FFs in a PE which can be taken in the timing report generated by Xilinx software. Peak CUPS gives the total time based on #PE. It can be calculated by the multiplying the peak frequency with the #PE.

5 Result and discussion

The Verilog HDL which has been used is targetable for a variety of FPGAs platforms. The score obtained from the matrix elementary operation can be compared with the waveform from the proposed design simulated in Xilinx ISIM simulator as shown in Fig. 8. This is to ensure the best-matched score from the core architecture is the same as the matrix alignment in Fig. 4. The parameters used for the core architecture were the same as the parameters for matrix alignment operation.

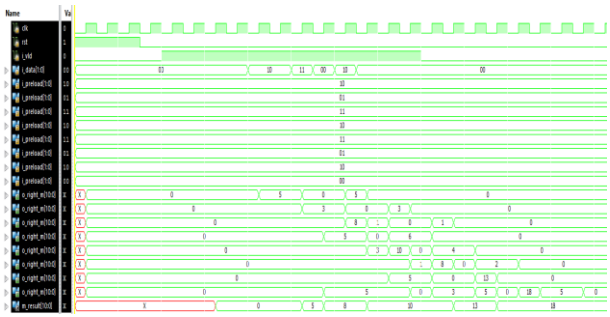

Fig. 8. Affine gap penalty SW algorithm computation output waveform.

In this work, the proposed core architecture is implemented in Virtex-6 FPGA whereby 6-input LUT is the building block of the device. For a fair performance comparison, the equivalent 6-input LUT is determined based on (2) for Stratix IV [20] and Virtex-4 and Spartan III [21] (3) first before determining the equivalent Slices using (4) [22]. The Logic Cell value (LC), Slices and ALM value can be retrieve from User Guide and Device Handbook respectively. Then, the equivalent slices are compared to reference slices as shown in Table 1 for normalized Slice/PE calculation.

$$6-\text{Input LUT (Equivalent)} = \frac{ALM}{1.2} \quad (2)$$

$$6-\text{Input LUT (Equivalent)} = \frac{LC}{0.5625} \quad (3)$$

$$\text{Slices (Equivalent)} = \frac{6-\text{Input LUT (Equivalent)}}{4} \quad (4)$$

Table 1. Logic blocks normalization.

FPGA Devices	Slice equivalent ratio with respect to Virtex - 6 XC6VLX75T
Virtex - 6 XC6VLX75T (Reference)	1
Virtex – 7 XC7VX485T	0.1533
Virtex – 5 XC5VLX330T	0.2245
Virtex – 4 XC4VLX100	0.2368
Spartan III XC3S250 XC3S1600E	4.754 0.7890
Stratix IV EP4SGX230KF40C2	0.1531

Virtex-6 FPGA has 11,640 slices where 776 PEs can be implemented in the aforementioned device with a 15 slices/PE. The proposed design was operated 600 MHz for the sequence alignment computation that is shown in Fig. 4. The comparison against other FPGAs platforms is reviewed in Table 2. Usually, the bottleneck for FPGA designs for area is the LUTs number [23] and timing is Peak CUPS [24]. Thus, based on Table 2, the proposed designed is the second slowest. However, the proposed design has the smallest slice as compared to other previous work. Since smaller core area can implement more # PEs, the speed performance is flopped to second place.

6 Conclusions

This paper reviewed on the current FPGA platform on implementing SA architecture for DNA sequence alignment. The proposed design is implemented on Xilinx Virtex - 6 FPGA using the affine gap penalty SW algorithm. The result shows that the proposed design has the smallest area core architecture and gained 465 GCUPS as compared to other architecture. Although it has the second highest in terms of speed, the core area is the smallest. This architecture design is suitable for high speed algorithm computation specifically for early disease detection such as cancer using its genetic profile.

Table 2. Performance comparison among devices.

Ref	Year	FPGA Device	Slice/PE	#PE	Peak Freq (MHz)	Peak GCups
[9]	2011	XC5VLX330T	75	1536	500	768
[10]	2012	XC4VLX100	90	128	60	7.61
[11]	2013	XC3S250E	332	35	50	1.75
[12]	2014	EP4SGX230KF40C2	1187	64	125	8
[13]	2016	XC3S1600E	96	241	98.7	23.79
[14]	2017	XC7VX485T	22	512	200	102.4
		XC6VLX75T	**15**	**776**	**600**	**465**

This work was supported by the FRGS under Grant 9003-00480. Modelling of an Efficient DNA/Protein Sequencer using Hardware Description Language (HDL) for Biomedical Applications.

References

1. F. H. C. Crick, "The Structure of DNA." Dept. of Biological Syst., The Cavendish Laboratory, Cambridge, England, Tech. Rep.
2. D. T. Hoang. "FPGA Implementation of Systolic Sequence Alignment," in *International Workshop on Field Programmable Logic and Applications,* Vienna, Austria, 1992.
3. D. T. Hoang, "Searching genetic databases on Splash 2", in *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines,* 1993, pp. 185-191.
4. L. Hasan, Y. M. Khawaja, A. Bais, "A Systolic Array Architecture for The Smith-Waterman Algorithm With High Performance Cell Design," in *Proceedings of IADIS European Conference on Data Mining, Amsterdam, The Netherlands,* Jul., 2008.
5. (2015, Feb. 5). *Congressional Justification FY 2015.* Retrieved September 24, 2016 from https://www.nlm.nih.gov/about/2015CJ.html.
6. T. F. Smith and M. S. Waterman. "Identification of Common Molecular Subsequences," *Journal of Molecular Biology,* vol. 147, no. 1, pp. 195-197, Mar. 25, 1981.
7. R. Giegerich, "A Systematic Approach to Dynamic Programming in Bioinformatics.," *Bioinformatics,* vol. 16, pp. 665-677, 2000.
8. W. 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.
9. A. Pułka and A. Milik, "Considerations on Incremental Approach to Hardware Implementation of Smith-Waterman Algorithm," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2011, IEEE 18th International Conference,* pp. 283-288, June 16-18, 2011.
10. N. Sebastião, N. Roma and P. Flores, "Integrated Hardware Architecture for Efficient Computation of the n-Best Bio-Sequence Local Alignments in Embedded Platforms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 20, no. 7, pp. 1262 – 1275, July,2012.
11. H. A. Shah, L. Hasan and N. Ahmad, "An Optimized and Low-cost FPGA-based DNA Sequence Alignment – A Step towards Personal Genomics," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS),* pp. 2696 – 2699, July 3 – 7, 2013.
12. J. Marmolejo-Tejada, M. V. Trujillo-Olaya, C. P. Rentería-Mejía and J. Velasco-Medina, "Hardware Implementation of the Smith-Waterman Algorithm using a Systolic Architecture," in *Circuits and Systems (LASCAS), 2014 IEEE 5th Latin America Symposium*, pp. 1 – 4, February, 2014.
13. D. S. Nurdin, M. N. Isa, and S. H Goh, "DNA sequence alignment: A review of hardware accelerators and a new core architecture," in *2016 IEEE 3rd International Conference on Electronic Design (ICED)*, pp. 264-268, August 11 -12, 2016.
14. X. Fei, Z. Dan, L. Lina, M. Xin and Z. Chunlei, "FPGASW: Accelerating Large-Scale Smith-Waterman Sequence Alignment Application with Backtracking pn FPGA Linear Systolic Array," *Interdisciplinary Sciences Computational Life Sciences,* pp. 1-13, 2017.
15. T. F. Smith and M. S. Waterman. "Identification of Common Molecular Subsequences," *Journal of Molecular Biology,* vol. 147 no. 1, pp. 195-197, Mar. 25, 1981.

16. O. Gotoh. "An improved algorithm for matching biological sequences," *Journal of Molecular Biology,* vol. 162, no. 3, pp. 705, 1982.
17. R. Durbin, S. Eddy, A. Krogh and G. Mitchison, *Biological Sequence Analysis:Probabilistic Models for Proteins and Nucleic Acids.* United Kingdom: Cambridge University Press, 1998.
18. K. Benkrid, A. Akoglu, C. Ling, Y. Song, Y. Liu and X. Tian, "High Performance Biological Pairwise Sequence Alignment: FPGA vs. GPU vs. Cell BE vs. GPP," *International Journal of Reconfigurable Computing,* Apr., 2012.
19. C. Fenton, (Aug. 2009,). *Final Project Report.*
20. "Advantages of the Virtex-5 FPGA 6-Input LUT Architecture,"Xilinx Inc., 2007.
21. "FPGA Logic Cell Conversion Ratios," Galorath Incorporated, 2014.
22. "Virtex-6 Family Overview," Xilinx, Inc*.,* 2015.
23. "FPGA Logic Cells Comparison," Core technologies.
24. M. N. Isa, M. I. Ahmad, S. A. Z. Murad, R. C. Ismail, K. Benkrid, "Biological Sequence Alignments: A Review of Hardware Accelerators and a New PE Computing Strategy," *IEEE Region Symposium,* pp. 1-6, 2014.