Trace Back Optimization for DNA Sequence Alignment Using Viterbi Algorithm

Nur Farah Ain Saliman*[1], Nur Dalilah Ahmad Sabri[1], Syed Abdul Mutalib Al Junid[1,2], Abdul Karimi Halim[1], Zulkifli Abd Majid[1], Nooritawati Md Tahir[1]

[1]*Faculty of Electrical Engineering*
Universiti Teknologi Mara
Shah Alam, Malaysia
ain_saliman@yahoo.com

[2] *Community of Research (CoRE)*
Advanced Computing and Communication (ACC)
Universiti Teknologi Mara
Shah Alam, Selangor, Malaysia

***Abstract* — This paper presents two trace back design for DNA sequences alignment using Viterbi algorithm for Pair Hidden Markov Models (PairHMMs) and design optimization for reducing the utilizing design resources. The replacement of the dynamic matrix trace back determination with the Viterbi algorithm for PairHMMs and optimization has reduced the size of the design with the correct result when determine the optimal path for the sequences alignment at the optimal resource utilization. Altera Quartus II version 10.0 was used for compiling the design and targeting to EP4CE115F29C7 devices from Cyclone V E family. The trace back design simulation and verification were conducted using ModelSim Altera with the 30.15% and 323.53% utilizing design resource has been recorded with the implementation of Viterbi for PairHMMs with optimization produced the optimal result.**

Keywords — Viterbi Algorithm, Smith-Waterman Algorithm, DNA sequence alignment

I. Introduction

In bioinformatics studies, the sequence alignment has been widely used to test the similarity regions between pair of DNA sequences. There are two types of sequence alignment that extensively used; global alignment and local alignment. Three technique comprises under the global alignment method; Dot Plot [1], Needleman-Wunsch (NW) [2] and Viterbi algorithm [3].

Local alignment is more sensitive in determining the region of similarity and defines the area of higher similarity as the optimal path for the alignment. Two methods are using the approach named exact method (Smith-Waterman [2]) and heuristic method (FASTA [3] and BLAST [4]). Despite, there are two different types of DNA sequences alignment, but the aims are to identify the structural, functional and evolutionary relationship between the sequences.

In addition, Needleman-Wunsch [2] and Smith-Waterman [4] algorithms are known as the best algorithm for DNA sequences alignment. Both algorithm shares the same method for defined the cell score of the matrix cells except for the determination of optimal path trace back. Diagonal end-to-end was used for Needleman-Wunsh, and most similarity region based on maximum cell score are used for Smith-Waterman.

Several attempts have been made to improve and extend the sequences alignment process for providing the best algorithm in solving DNA sequence information. The Needleman-Wunsch algorithm [2] based on global alignment has been proposed in 1970 for the start the DNA sequences alignment based on dynamic matrix. Again, Smith-Waterman algorithm [4] being introduced after some modification on the optimal alignment based on local alignment method with the maximum cell score trace back was introduced. Then, the Gotoh [5] introduced another improvement by considered linear gap penalty for Smith-Waterman scoring-based. Followed with Myres and Miller [6] with quadratic time and linear space optimization. The comparison algorithm based on equal and non-equal was introduced by Aho, Hirschberg and Ulman [7]. The implementation of the optimization technique over the module in DNA sequences alignment has been intensively focused by [8]-[12] since it can improve the sensitivity, performance, design size and reducing complexity of the algorithm with various of platform involved including the Field Programmable Gate Array (FPGA) [13]-[16] and Application Specific Integrated Circuit (ASIC) [17]. Viterbi algorithm are among the best optimization algorithm for determining the best path, and it can be adapted for improving the optimal path trace back and it has been widely used before in communication [18] and bioinformatics.

L.R Rabiner et. al [3] applied the theory of Hidden Markov Models (HMMs) in solving speech recognition problem while the application of Hidden Markov Models in bioinformatics was proposed by Valeria De Fonzo et. al [19] in Hidden Markov Models in Bioinformatic. Meanwhile, Byung-Jun Yoon [20] paper is presented on the used of Hidden Markov Models (HMMs) in biological sequence analysis and also focus on three types of Hidden Markov Models (HMMs) modification to meet the needs of various Bioinformatics application such as Profile HMMs, PairHMMs and context-sensitive HMMs.

Alternatively, the implementation is to introduced a new efficient trace back module of Smith-Waterman algorithm by utilizing the trellis diagram and PairHMMs. The new trace back module with replacement of dynamic matrix cell calculation was used to solve the existing trace back system. In addition, the design was implemented using scheduling strategy for reducing the design complexity and optimizing the design at the same time. On the other hand, the method can improve the size of design area with small design utilizing resources.

This paper had been divided into four sections, whereby in section II will discuss the basic of algorithm and method of implementation. Section III will highlight the results and discussion of this improvement. Finally, the conclusion for the finding will be summarized in Section IV.

II. Methodology

The methodology for the proposed system has been divided into two sub-section named trace back system design and design setup.

A. Trace Back Technique and Optimization

Three trace back technique for DNA sequences alignment has been highlighted in this sub-section. The original trace back based on the Smith-Waterman algorithm has been discussed under the first topic and followed by the Viterbi algorithm for pairHMMs and the last topic is Trace back Viterbi algorithm optimization.

1) Smith-Waterman Algorithm (Design1)

In 1981, the Smith-Waterman algorithm [2] is proposed by Smith and Waterman for local sequence alignment and identified as most sensitive algorithm for identifying similarity regions. This regions is very important to find the DNA relationship as described in section I. Thus, Smith and Waterman introduced a matrix, H where is to keep track the degree of similarity between the two sequences; query sequence, N_q and subject sequence, N_s. The H, N_s and N_q used in the algorithm are shown in Fig. 1.

	Query Sequence (Nq)					
			A	**C**	**G**	**T**
Subject Sequence (Ns)		**0**	**0**	**0**	**0**	**0**
	A	**0**	**1**	**0**	**0**	**0**
	C	**0**	**0**	**2**	**1**	**0**
	G	**0**	**0**	**1**	**3**	**2**
	T	**0**	**0**	**0**	**2**	**4**

Fig.1. Smith-Waterman H matrix with trace back path

Generally, consider two sequences of DNA as query sequence, N_q and subject sequence, N_s with a length of n and m. In this example, if N_q = {ACGT} represents a query sequence and N_s = {ACGT} represent a subject sequence. So that, length of query sequence, N_q and subject sequence, N_s are $n = 4$ and $m = 4$. Assume that, a simple scoring scheme of Smith-Waterman algorithm as followed:

$$S_{i,j} = \begin{Bmatrix} +1 \text{ if } N_q = N_s \\ -1 \text{ if } N_q \neq N_s \end{Bmatrix}$$

$$\text{and } d = 1 \text{ (gap penalty)} \tag{1}$$

The Smith-Waterman algorithm involved several steps as followed:

a) Initialization

The matrix H was constructed by placing the query sequence, N_q on the column of the matrix and the subject sequence, N_s on the row of H matrix as shown in Fig. 2. In addition, the first row and first column of the matrix need to be initializing with zeroes during the initialization process.

	Query Sequence (Nq)					
			Nq1	**Nq2**	**Nq3**	**Nq4**
Subject Sequence (Ns)		**1**	**0**	**0**	**0**	**0**
	Ns1	**0**				
	Ns2	**0**				
	Ns3	**0**				
	Ns4	**0**				

Fig.2. Smith-Waterman initialization step

b) Fill Matrix

The calculations for the cell value were conducted in Fill Matrix step. It starts from the upper corner left cell of the matrix H at the cell number $H_{1,1}$. The cell value was calculated based on the equation (2) with the similar sequences will be added to score and mismatch sequences will be penalized by a penalty. Determination of the cell value required the neighbor cell value from the left cell, upper cell and upper left diagonal cell position.

$$H_{i,j} = \max\begin{Bmatrix} 0 \\ H_{i-1,j-1} + S_{i,j} \\ H_{i-1,j} - d \\ H_{i,j-1} - d \end{Bmatrix} \qquad (2)$$

Furthermore, to calculate each $H_{i,j}$ cell will depend on three neighbor cells which are in terms of position matrix $H_{i-1,j-1}$, $H_{i-1,j}$ and $H_{i,j-1}$ are known as upper left (diagonal) cell, left cell and upper cell as describe in Fig. 3.

			Query Sequence (Nq)	
Subject Sequence (Ns)			Nq1	Nq2
		0	0	0
	Ns1	0	Hi,j-1	Hi,j-1
	Ns2	0	Hi-1,j	Hi,j

Fig.3. Smith-Waterman data dependencies

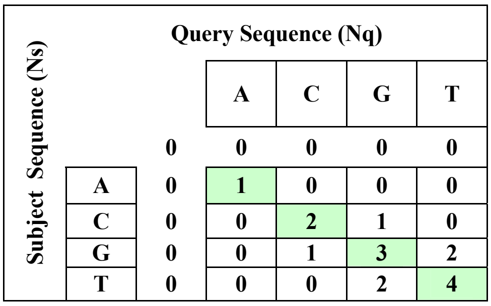
c) Trace Back

After completion of the matrix fill step for the entire cell, the maximum cell value will be selected as the start location for trace back process as stated in Equation (3) and (4).

$$H_{(opt)} = \max(H_{i,j}) \qquad (3)$$

$$\text{traceback } (H_{(opt)}) \qquad (4)$$

Finally, trace back step begins in cell with the highest score, and it tracing until the last cell with the lowest score or initial value as shown in Fig. 4.



			Query Sequence (Nq)			
Subject Sequence (Ns)			A	C	G	T
		0	0	0	0	0
	A	0	1	0	0	0
	C	0	0	2	1	0
	G	0	0	1	3	2
	T	0	0	0	2	4

Fig.4. Smith-Waterman algorithm trace back

2) Viterbi Algorithm for PairHMMs (Design2)

The Viterbi algorithm [12] has been proposed by A.J Viterbi and the Viterbi algorithm in 1967. Furthermore, it has been defined as the most efficient computational technique since it can determine the most probable path through searching all the possible path by using the trellis diagram.

a) Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) is enhancement from simple Markov chain where it represents a random variable sequence. The flow processes to determine the current state of Markov chain always depend on the previous state. This Markov chain also known as hidden state, Z in Hidden Markov Models (HMMs) structure. However, Hidden Markov Models (HMMs) cannot observe directly through hidden state. Thus, observed state, X require to lean the hidden state, Z or the originally known as Markov chain.

Fig. 5 shows that all the unobservable state as hidden state, Z is inferences through the observable state, X. Hidden Markov Models (HMMs) can be defined as 5-tuple HMMs = $\{A,B,\pi,Z,X\}$. The hidden state sequence as $Z_i = \{Z_1, Z_2, ...Z_L\}$, meanwhile the observed state sequence will be $X_i = \{X_1, X_2,....X_L\}$, where L represents the length of the state sequence. Despite that, each hidden state, Z is takes one of the set of state values $S = \{1, 2 \ldots M\}$, with M is denoted as number of distinct model states. In addition, Hidden Markov Model (HMMs) is assuming that the hidden state, Z as homogeneous first order Markov chain. The hidden states, Z connection is known connected in some way, and there is a possibility that the state move from one state to another. This process involved the transition probability and has a movement between the states through a matrix of state transition probability as shown in Equation (5).

$$A = \{a_{i,j}\} = P\,(Z_{n+1} = S_j \,|Z_n = S_i);\ 1 \leq i,j \leq M \qquad (5)$$

For the transition probability i need to be greater or equal to 1 and j need to be smaller or equal to distinct model states values, M. Other than that, the initial hidden state, $Z_1$is known as initial state and the initial state probability is denoting as

$$\pi = \pi_i = P\,(Z_1 = i);\ 1\leq i \leq M \qquad (6)$$

For the initial state probability i denote as greater or equal to 1 and smaller or equal to distinct model states, M. In addition, inside each observed state there it will have a set of observation $O_i = \{O_1, O_2, ...O_N\}$, with N denote as number of distinct observations. The observed output is referring to the symbols in every state and the symbols can be mention as the discrete alphabet. This observation symbols can be defined as a distribution probability, $B = \{b_j (k)\}$ as shown in equation (7)

$$B = \{b_j\,(k)\} = P\,(X_n = k|\, Z_n = S_j);\ 1 \leq i \leq M,\ 1 \leq k \leq N \qquad (7)$$

For the distribution probability *i* denote as greater or equal to 1 and smaller or equal to distinct model states values, *M*. Meanwhile, *k* need to be greater or equal to 1 and smaller or equal to length distinct observations, *N*. However, distribution probability sometimes known as emission probability that is the probability of S_j is generated the observation of *k*. Finally, the complete set of Hidden Markov Models (HMMs) parameter can be denoted as 3-tuple parameter where consisting three probability measures as HMMs comprise = $\{A, B, \pi\}$.

b) Viterbi Algorithm for PairHMMs

This process comprises of four steps towards finding the similarity sequences between two DNA sequences; initialization, iteration and trace back.

i. Initialization

The initialization with the initial value or zero for the first row and column are required in the first step as shown in Equation (8).

$$V^M_{0,0} = 1;\ V^M_{i,0} = 0;\ V^M_{0,j} = 0$$

$$V^X_{i,0} = V^X_{0,j} = 0$$

$$V^Y_{i,0} = V^Y_{0,j} = 0 \qquad (8)$$

The placement of the initial value to the first row and column is shown in Fig. 6.

			Query Sequence (Nq)			
			Nq1	**Nq2**	**Nq3**	**Nq4**
		1	**0**	**0**	**0**	**0**
Subject Sequence (Ns)	**Ns1**	**0**	**V(1,1)**	**V(1,2)**	**V(1,3)**	**V(1,4)**
	Ns2	**0**	**V(2,1)**	**V(2,2)**	**V(2,3)**	**V(2,4)**
	Ns3	**0**	**V(3,1)**	**V(3,2)**	**V(3,3)**	**V(3,4)**
	Ns4	**0**	**V(4,1)**	**V(4,2)**	**V(4,3)**	**V(4,4)**

Fig.6. Match matrix table

ii. Iteration

In this algorithm, the Pair Hidden Markov Models (PairHMMs) was used to find the probability hidden state by searching through all possible observed states. Viterbi algorithm for PairHMMs iteration step was illustrated as in Fig. 7. The arrow is denoted as a transition arrow with the value of the proposed work was described in TABLE I as followed

TABLE I. TRANSITION TABLE

	M	**X**	**Y**
M	1-2δ	Δ	δ
X	1-ε	E	0
Y	1-ε	0	ε

iii.Trace Back

In trace back step, this algorithm will compare and select the highest score as defined by the equation (9). Finally, it will trace the most probable path by searching through all possible paths of the matrix using the trellis diagram.

$$Z_{m,n*} = \arg\max_K \{V^M_{i,j}, V^X_{i,j}, V^Y_{i,j}\} \qquad (9)$$

3) Implementation of trace back Viterbi algorithm for optimization (Design3)

The implementation is combining both of the algorithms, and we named as an extension of Smith-Waterman. It has been divided into three steps.

a) Initialization Step

We filled all the first and column cell with initial value or zero as defined by the equation (10)

$$H_{j,0} = H_{0,j} = 0 \qquad (10)$$

b) Fill Matrix Step

After the initialization step, matrix fill equation defined in equation (2) has been used to calculate all cell score. The calculation will consider the entire neighbor cell, match and mismatch for calculating the new cell score.

c) Trace Back Steps

An overview of the proposed trace-back is shown in the Fig.8, where the trace back steps involved two modules named termination and trace back.

i. Termination

Termination step was used to find the maximum score value for $H_{i,j}$ cell from the entire matrix, *H* cell score value. The entire cell has been searched for identified the maximum score bounded by *n* rows and *m* columns of the query sequence $N_q = \{N_{q1}, N_{q2}, ...,N_{qm}\}$ and subject sequence $N_s = \{N_{s1}, N_{s2}, ..., N_{sn}\}$. However, the hidden state cannot be determined directly. Thus, hidden state will underlie to the state *k*. The maximum score $H_{i,j}$ cell can be expressed as in equation (11).

$$H_{i,0} = \max_K(H_{n,m}) \qquad (11)$$

ii. Trace Back

We used the equation (12) to search the hidden state which is to trace the likely sequence between length of *n* rows and *m* columns. Then, trace back will be stopped when it reaches or identified the initial probability value.

$$Z_{m,n*} = \arg\max_K (H_{*n,m}) \qquad (12)$$

The score value is underlie to hidden state was highlighted in TABLE V and it present the score with the probable sequence between a pair of sequence. The highlighted cells are the trace back path output of the matrix *H*.

B. Simulation setup for implementation

The code for the proposed trace back optimization was constructed using Verilog Hardware Description Language (HDL) and targeted to the Cyclone II EP4CE115F29C7 devices from Cyclone V E family running at 50MHz clock frequency. The code was developed and compiled using Altera Quartus II version 10.0 tools and simulation was running on ModelSim Altera SE version 10.0c. Furthermore, the simulation was running at the Intel Core i5 processor with 1025MB RAM.

Four different data have been used as input in this study as shown in TABLE II. It has been used for both theoretical and simulation phase. The obtained result produced by the proposed system will be used for analyzing the performance of the system and comparing with the theoretical result.

TABLE II. SAMPLE DATA FOR TESTING

Input	
Query	***Subject***
AAAA	AAAA
ACGT	ACGT
AATT	AATT
ATAT	TATA

III. RESULTS AND DISCUSSION

A. Theoretical

TABLE III shows the result of the optimal path through a Hidden Markov Models using Viterbi algorithm termination and trace back steps. It was obtained by selected by the most optimal path with high similarity of DNA sequence alignment.

TABLE III. THEORETICAL RESULT

Input		**Output**
Query	***Subject***	
AAAA	AAAA	1234
ACGT	ACGT	1234
AATT	AATT	1234
ATAT	TATA	1233

B. Simulation

Fig.9, Fig.10, Fig.11 and Fig.12 shown the final trace back output decision based on Viterbi algorithm using PairHMMs termination and trace back module. In the result the “00” represents the A, “01” represents C, “10” represents G and “11” represents T. All the data were comprised in one register with a length of 16 bits data length. The results produced by the simulation are similar to the theoretical result in TABLE III obtained from the calculation.

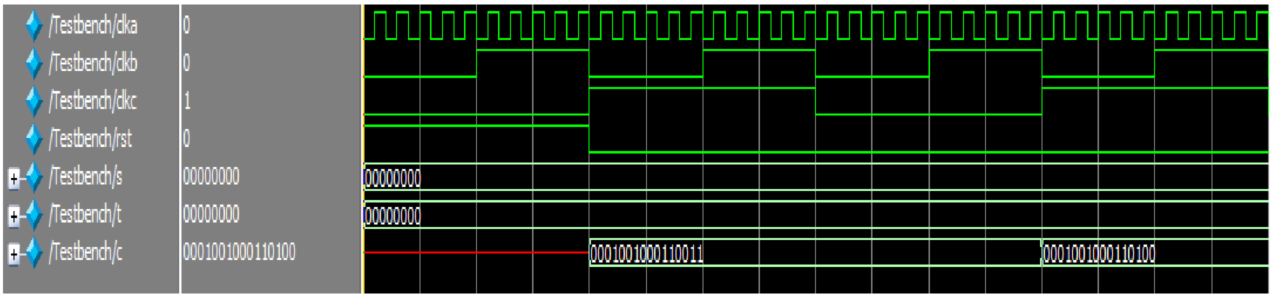


Fig.9. Data 1 simulation result

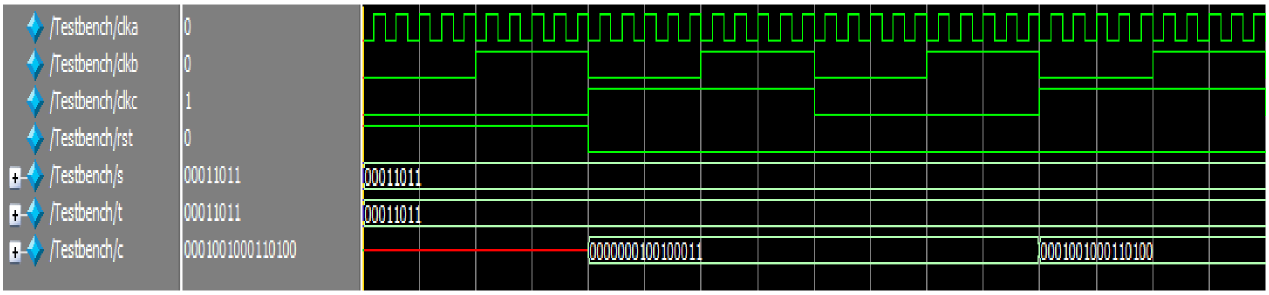
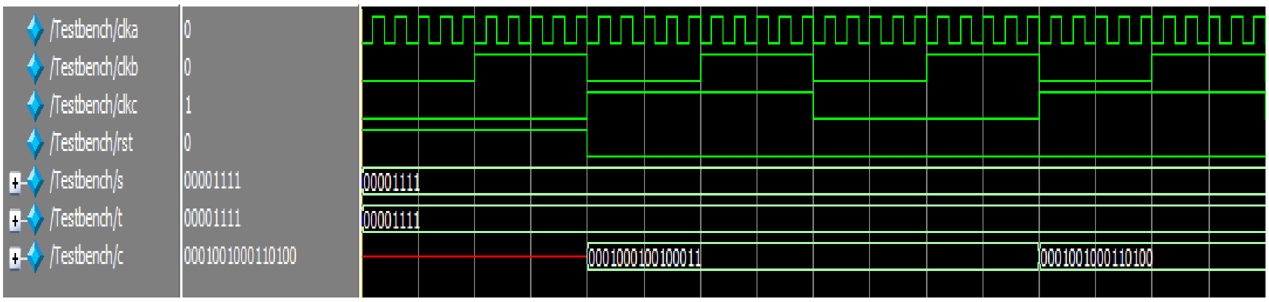


Fig.10. Data 2 simulation result

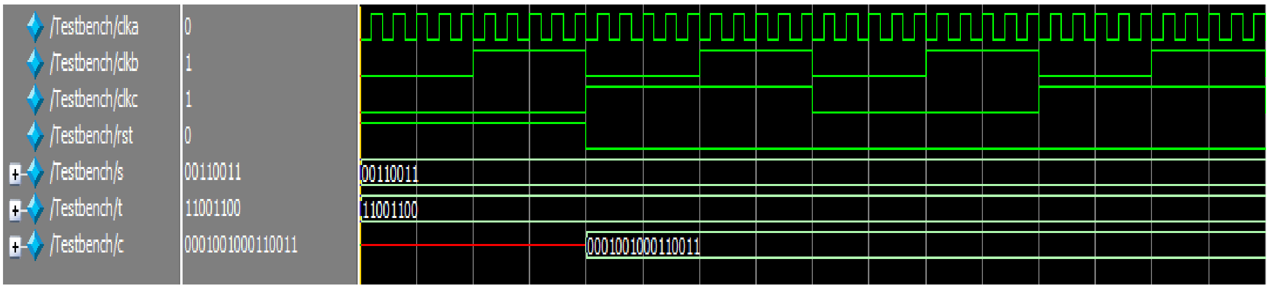
DOI 10.5013/IJSSST.a.14.06.06

ISSN: 1473-804x online, 1473-8031 print



```
/Testbench/clka   0
/Testbench/clkb   0
/Testbench/clkc   1
/Testbench/rst    0
/Testbench/s      00001111            00001111
/Testbench/t      00001111            00001111
/Testbench/c      0001001000110100    0001000100100011    0001001000110100
```

Fig.11. Data 3 simulation result



```
/Testbench/clka   0
/Testbench/clkb   1
/Testbench/clkc   1
/Testbench/rst    0
/Testbench/s      00110011            00110011
/Testbench/t      11001100            11001100
/Testbench/c      0001001000110011    0001001000110011
```

Fig.12. Data 4 simulation result

There are three sub-module in this design which are controlled by three clock source named *Testbench/clka*, *Testbench/clkb* and *Testbench/clkc* with 10MHz, 50MHz and 100MHz interval as shown in Fig. 9, Fig. 10, Fig. 11 and Fig. 12. The *Testbench/rst* is set as active high where it will reset all the input and output when it was at logic high and allow the computational when it was at logic low. Furthermore, the subject sequence and query sequence were denoted as *Testbench/s* and *Testbench/t*. The character in each of the sequence was represented in the form of 2 bit data as shown in TABLE IV.

TABLE IV. BASES REPRESENT

Bases	Number of Bits
Adenine (A)	00
Cytosine (C)	01
Guanine (G)	10
Thymine (T)	11

The result for the optimal path trace back was represented by the *Testbench/clkc* signal and the optimal path trace back result for all four samples data are shown Fig. 9, Fig. 10, Fig.11 and Fig.12. The last obtained result was the final result since the optimization technique trying to search for the optimal path during the trace back. The final result for Fig. 9, Fig. 10 and Fig.11 are obtained after the initial result or before the optimization while the final result obtained for the Fig.12 are the optimized result since the system did not found any optimized after the first or initial result. The result are written in sixteen bit data format to represent the numerical value of the optimal path cell score with four bit representing one cell value or score. TABLE V summarized the result obtained from the simulation in Fig.9, Fig.10, Fig.11 and Fig.12 and compared with the expected result from the TABLE III.

TABLE V. SIMULATION RESULT REPRESENT

Expected Output	Simulation Output
1234	0001 0010 0011 0100
1234	0001 0010 0011 0100
1234	0001 0010 0011 0100
1233	0001 0010 0011 0011

Furthermore, the utilization summary shows that the design only used 246 logic elements from the total of 114480 total logic elements consisting in the EP4CE115F29C7 devices for the Design3 . The comparative the devices utilization size of all three designs are shown in TABLE VI. Design3 are only utilized 30.15% from the original Design1 while Design2 required more logic element are recorded at 323.53% from the Design1.

TABLE VI. DESIGN COMPARISON

Cell Design	Total Logic Elements
Original SW (Design1)	816
VA for PairHMMs (Design2)	2640
Optimization SW (Design3)	246

IV. CONCLUSION

This paper presents two new design based on the Viterbi algorithm for PairHMMs and compared with the existing trace back for Smith-Waterman algorithm. The result demonstrate that, the optimized Viterbi algorithm for PairHMMs has produced the optimal resource utilization from the other design even though the original Viterbi algorithm for PairHMMMs required more resource as compare to the existing trace back in Smith-Waterman algorithm. Therefore, we can summarized that, the optimized Viterbi algorithm for PairHMMs produced the optimized solution and capable to reduce the size of design for application in DNA sequences alignment accelerator.

ACKNOWLEDGMENT

The authors would like to acknowledge the Ministry of Science, Technology and Innovation Malaysia (MOSTI) and Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM), Malaysia for providing financial support under Science Fund Grant (100-RMI/SF 16/6/2 (17/2012)) and laboratory facilities.

REFERENCES

[1] A. J. Gibbs, G. A. Mcintyre, A. Acid, and S.- Cytochromes, “The Diagram, a Method for Comparing Sequences,” vol. 16, pp. 1–11, 1970.

[2] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins.,” J. Mol. Biol., vol. 48, no. 3, pp. 443–53, Mar. 1970.

[3] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proc. IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[4] M. S. Waterman, “Identification of Common Molecular Subsequences Identification of Common Molecular Subsequences,” pp. 195–197, 1981.

[5] O. Gotoh, “An improved algorithm for matching biological sequences.,” Journal of molecular biology, vol. 162, no. 3. pp. 705–8, 15-Dec-1982.

[6] E. W. Myers and W. Miller, “Optimal alignments in linear space.,” Comput. Appl. Biosci., vol. 4, no. 1, pp. 11–7, Mar. 1988.

[7] D. S. Hirschberg, “Algorithm for Computing Maximal Common Subsequences,” vol. 18, no. 6, pp. 2–4, 1975.

[8] S. A. M. Al Junid, N. M. Tahir, Z. A. Majid, and M. F. M. Idros, “Potential of Graph Theory Algorithm Approach for DNA Sequence Alignment and Comparison,” 2012 Third Int. Conf. Intell. Syst. Model. Simul., pp. 187–190, Feb. 2012.

[9] Junid, S.A.M.A.; Tahir, N.M.; Majid, Z.A.; Halim, A.K.; Shariff, K.K.M., "Improved data minimization technique in reducing memory space complexity for DNA local alignment accelerator application," Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on , vol., no., pp.153,156, 3-4 Dec. 2012

[10] Al Junid, S.A.M.; Reffin, M.S.; Majid, Z.A.; Tahir, N.M.; Haron, M.A., "Implementation of genetic algorithm for optimizing DNA sequence alignment," Business Engineering and Industrial Applications Colloquium (BEIAC), 2012 IEEE , vol., no., pp.80,85, 7-8 April 2012

[11] S. A. M. Al Junid, Z. A. Majid, and A. K. Halim, “Development of DNA sequencing accelerator based on Smith Waterman algorithm with heuristic divide and conquer technique for FPGA implementation,” 2008 Int. Conf. Comput. Commun. Eng., pp. 994–996, May 2008.

[12] S. A. M. Al Junid, M. A. Haron, Z. A. Majid, A. K. Halim, F. N. Osman, And H. Hashim, “Development of Novel Data Compression Technique for Accelerate DNA Sequence Alignment Based on Smith–Waterman Algorithm,” 2009 Third UKSim Eur. Symp. Comput. Model. Simul., pp. 181–186, 2009.

[13] S. A. M. Al Junid, M. A. Haron, Z. A. Majid, F. N. Osman, H. Hashim, M. F. M. Idros, and M. R. Dohad, “Optimization of DNA Sequences Data to Accelerate DNA Sequence Alignment on FPGA,” 2010 Fourth Asia Int. Conf. Math. Model. Comput. Simul., pp. 231–236, 2010.

[14] S. A. M. Al Junid, N. Md Tahir, Z. Abd Majid, Z. Othman, and K. K. Mohd Shariff, “Reducing memory complexity using data minimization technique on FPGA,” in 2012 International Conference on Computer & Information Science (ICCIS), 2012, pp. 431–434.

[15] Al Junid, S.A.M.; Tahir, N.M.; Majid, Z.A.; Osman, F.N.; Mohd Shariff, K.K., "Comparative study for DNA data minimization technique on FPGA," Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on , vol.2, no., pp.765,767, 12-14 June 2012

[16] Al Junid, S.A.M.; Majid, Z.A.; Halim, A.K., "High speed DNA sequencing accelerator using FPGA," Electronic Design, 2008. ICED 2008. International Conference on , vol., no., pp.1,4, 1-3 Dec. 2008

[17] N. Khairudin, M. a. Haron, S. a. M. Al Junid, a. K. A. Halim, M. F. M. Idros, and N. F. A. Razak, “Design and Analysis of High Performance and Low Power Matrix Filling for DNA Sequence Alignment Accelerator Using ASIC Design Flow,” 2011 UKSim 5th Eur. Symp. Comput. Model. Simul., pp. 123–128, Nov. 2011.

[18] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” Inf. Theory, IEEE Trans., pp. 260–269, 1967.

[19] B. Schuster-Böckler and A. Bateman, “An introduction to hidden Markov models.,” Curr. Protoc. Bioinformatics, vol. Appendix 3, no. January, p. Appendix 3A, Jun. 2007.

[20] B.-J. Yoon, “Hidden Markov Models and their Applications in Biological Sequence Analysis.,” Curr. Genomics, vol. 10, no. 6, pp. 402–15, Sep. 2009.

APPENDIX

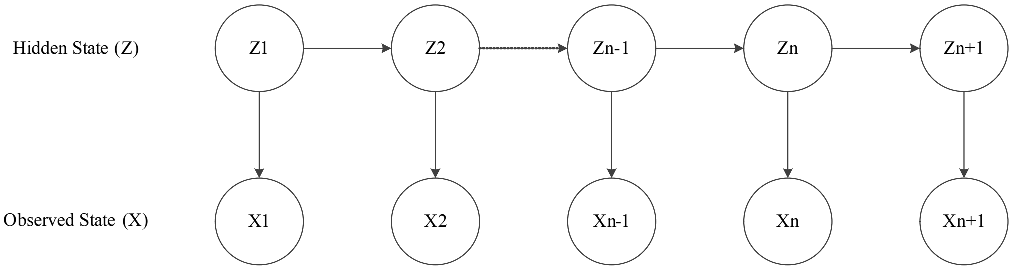


Fig.5. HMMs with hidden state and observed state.

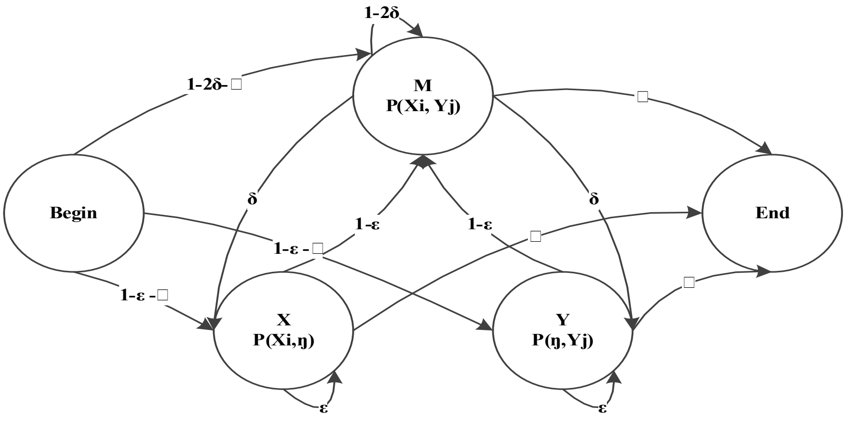


Fig.7. Iteration state flow process

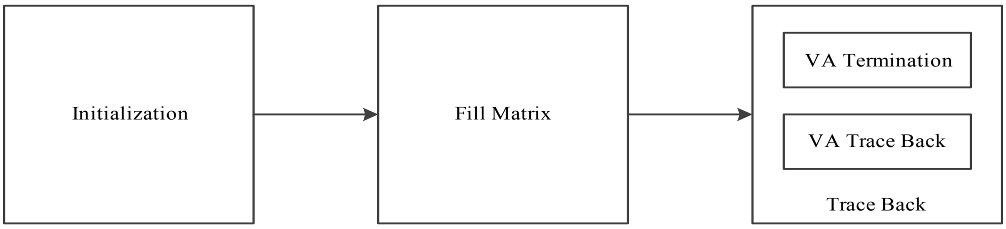


Fig.8. Smith-Waterman extension flow process