Hybrid Optimal Path Traceback System Development for Optimizing DNA Sequences Alignment

Faculty of Electrical Engineering
Universiti Teknologi MARA,
40450 Shah Alam, Selangor. Malaysia.
Email:dalilah_sabri@yahoo.com

***Abstract* - This paper present the new hybrid optimal path trace back system development for solving optimal path trace back complexity issue. The objective of this paper is to study several optimization techniques and determine the best optimization technique for optimizing the optimal path trace back system. Therefore, two designs are proposed and analysed for in this study. The project is divided in two stages which are theoretical and experimental design. In theoretical design, the proved of concept for the proposed design is calculated based on mathematical equation which defined by Smith-Waterman algorithm. Design construction which covers code development, compilation and simulation is carried out under experimental design for both of the proposed designs using Altera Quartus II version 9.0 EDA tools and targeted to Cyclone II EP2C35 FPGA at 100MHz clock cycle. As a result, the second design required three times design area as compare to the first design for determine the optimal path for the same matrix size. Therefore, the first design is the best hybrid approach for the optimal path trace back size since theoretical result has shown both of the design has the same output but the second design suffers in terms of large design area.**

Keywords— **DNA sequence alignment, dynamic programming, Smith-Waterman algorithm, optimal path trace back.**

I. INTRODUCTION

Lately, the size of DNA sequence is increasing due to the increases of the number of population. In 2010, [2] has discussed that the consequences of the increment drags the DNA sequences alignment system to become so slow. Sequence alignment is the main point that needs to be highlighted since it was the important role in DNA data sequence alignment field has been proposed by [1] in 2008. Although S-W is stable however since the population increase, it causes the algorithm and trace back slow. Hence the optimal traceback method came up because of that. This paper actually focussed on optimal path traceback using Smith-Waterman Algorithm due to optimize as well as find the best similarity technique in optimal traceback value for DNA sequence alignment. In addition, in 2009 [4] says that for the traceback in the database, it takes from the optimal point to the lowest point by using comparison method. The design architecture will scan the best optimize performance. The performances are in terms of runcycle time and the speed.

Based on [7] in 2007 paper, it is focused on Smith-Waterman algorithm due to accelerate runtime using FPGA. This project is to implement score cell in SW matrix. In addition, this module is to perform speed in propagation through the FPGA circuit. Finally the project is accelerated the algorithm time up to 160 folds.

Other researcher in 2005, [10] has proposed on statistical to be used for selective database scanning. Even though dynamic programming algorithms are efficient but they still have a problem. The problem can cause the scanning time process run in a very long time. It happens since there is increasing number of population. In this paper, it present the architectures design can be used to derive an efficient dynamic programming calculation. The technique helps save runtime for HMM trace back scanning.

II. SEQUENCE ALIGNMENT

As we know sequence alignment is actually work by comparing two or more of DNA sequences or based on the protein sequences that was implement from same DNA [1-2]. In fact [6], DNA consists of four characters and they are A, G, C and T or as known as Adenine, Cytosine, Guanine and Thymine in 1970. The uniqueness of these DNA's cells is that every individual has the same alignment in each part of the body but somehow different from other person. Therefore to overcome this matter, many researchers' focus on dynamic programming which cover in two types of sequences alignment.

Under sequence alignment methods can only be consider as either local or global. The local alignment normally aiming on how to identify of the similarity between two sequences, while global alignment was trying to match and mismatch as many DNA sequence or all of the possibilities [2]. Based on Figure 1, it shows that the sequence alignment methods can be divided into various kinds of methods.

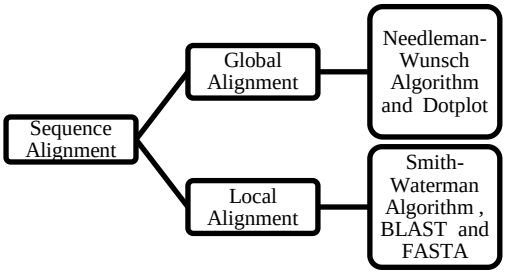

Figure 1: Sequences Alignment Methods.

In other sub modules will discuss for all sequences alignment which is follow by [8] researcher in year 1985 that proposed with global alignment (Needleman-Wunsch Algorithm and Dot plot) and local alignment (Smith-Waterman Algorithm, BLAST and FASTA).

III. GLOBAL ALIGNMENT

a. Dot plot

The beginning of history for sequences alignment is dotplot. Dotplot is the most basic method when the time of comparing two sequences [7]. A dot is placed wherever the search sequence (column) letter and target sequence (row) letter intersects with each other and when compare, they are the same. When it matches a diagonal stretch of dots will be placed to show its region.

For example, assume that the globally alignment consist of two DNA sequences alignment such as search sequence (vertical) = gtacc and target sequence (horizontal) = gtacc and form it using dot plot (star) matrix table [2] as in Table 1.

SEARCH SEQUENCE

TARGET SEQUENCE

	g	t	a	c	c
g	*				
t		*			
a			*		
c				*	*
c				*	*

Table 1: Dot Plot

b. Needleman-Wunsch Algorithm

Meanwhile for the Needleman-Wunsch Algorithm it uses dynamic programming method which is the most popular method in determining the sequence alignment by comparison. With this algorithm, can find out the best alignment between two DNA of sequences and it is suitable for large sequences alignment. Due to the structure, it relates to each other from the beginning to the end. The disadvantage of this algorithm is that it has too much gap penalty noises that reduce the accuracy of the alignment.

IV. LOCAL ALIGNMENT

a. Smith-Waterman Algorithm

In this project S-W algorithm is a promising algorithm. This algorithm being promising algorithm since it is good in finding the optimal path as well as improvement of N-W algorithm [4]. S-W algorithm is the second algorithm using dynamic programming method. Unfortunately, S-W also has disadvantage which is it slows down when computing the larger system. The way to get the data is by initialization, fill matrix and trace back.

There are three neighbouring cells which each above cell (n), left cell (w), and diagonally upper-left cell (nw) that must be compared with each other such in Figure 2 [7]. After compare, then calculate the maximum or highest score that has been select in the cell. Figure 2 is the basic structure of S-W matrix.

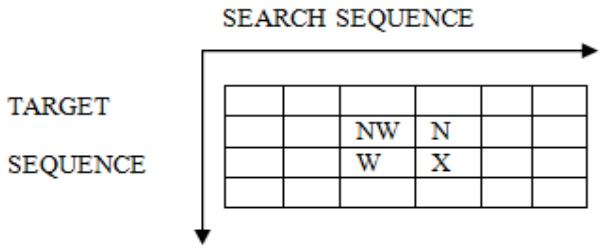

Figure 2: Basic structure of SW matrix

Figure 2 shows the basic structure of S-W score that will be place. Based on Figure 2 it can be elaborate based on Figure 3 flow process.

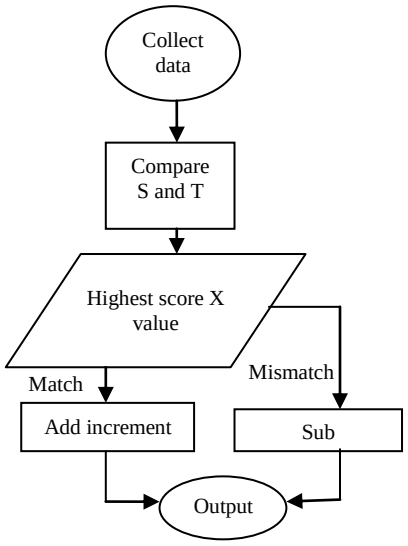

Figure 3: Flow chart for S-W algorithm

Figure 3, shows the flow process for S-W algorithm score calculation. Firstly in 2010 based on [3] need to collect data for Search Sequence and Target Sequence character. Then, compare two characters from each side. At that point, score X will calculate as NW value plus with match value if both character is same. If Search Sequence and Target Sequence character is mismatch the score will calculate with any of these large values between NW, N and W will minus with mismatch value. To make things thing clear, Figure 3 above show the flowcharts for the S-W steps.

b. FASTA and BLAST

Apart from S-W algorithm, FASTA (Fast Alignment Search Tools – All) and BLAST (Basic Local Alignment Search Tools) is also under local alignment. Both are tend to be faster, however this method will reduce the sensitivity of accuracy. FASTA is discovered by [8] in 1985. FASTA can roughly get optimal alignment by matching and searching the sequences with k tuples. K tuples is the length for the two proteins. In 1990, [9] has discussed BLAST is similar to FASTA but instead of using k tuples, it uses w where w is the word size.

All the above sequence alignment methods were more focussed on the way to calculate the value of the matrix box. Based on previous researched the best sequences alignment was Smith-Waterman Algorithm. Even though this algorithm is the best sequence alignment, it still faced with some problem which is the algorithm become slow. This happen cause this algorithm cannot support large DNA sequence alignment comparison. Thus, it needs other supporter to increase the speed otherwise it will still be slow. This problem can be settled by using either BLAST an FASTA, but both methods will cause the result not accurate. Data minimization is also one of the method or another way that can solve this problem. Not only it minimize the number of DNA bit but it also can save memory space.

V. OPTIMAL PATH TRACE BACK

This paper presents on DNA trace back sequence alignment. This method will help to find the optimal path. Traceback will trace through all DNA Sequence alignment length. Then the highest sequence will be developed by the S-W matrix. Yet, it is good to know it uses in the application such as server, GPU and CPU. There were two steps in order to use this technique as in Figure4 below:

Figure 4: Traceback flow process.

Firstly, is filling all matrix score into the matrix table. As we know when filling in the matrix, uses computation from the S-W algorithm. Next step is to trace back all score to find the optimal path from matrix table. This can be done by comparing from the highest value towards the small value.

VI. METHODOLOGY

There are two simple methods that will be used in this section. It will conduct and produce using two methods to form the output trace back based on the two different architecture designs. Both designs will use Smith-Waterman Algorithm (SWA). This algorithm is being use to support the Matrix score calculation. The output will be trace back by using Comparator technique.

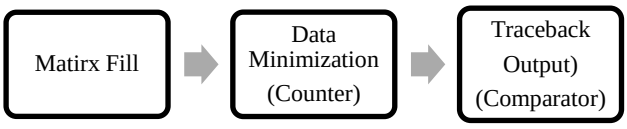

Figure 5: Flow process Block Diagram

Based on the flow process in the Figure 5, it starts with matrix filling. The score actually already calculated by the previous researcher. Next step is to create the data minimization column. In this column, minimization is needed as the data reduced from 8 bit into 2 bit. While for the trace back method will be apply using comparator design. Table 2 shows the design 1 and Design 2 that uses 4 x 4 Matrix size of table.

	SEARCH SEQUENCE			
TARGET SEQUENCE	X15	X14	X13	X12
	X11	X10	X9	X8
	X7	X6	X5	X4
	X3	X2	X1	X0

Table 2: 4 by 4 matrix score

a. Design 1

In design 1, the comparison method for 4 x 4 matrix table will be break into five comparators. The main architecture Design 1 will be connected as in Figure 6 below. The design block diagrams consist of four comparators in one stage and remain one comparator in another stage.

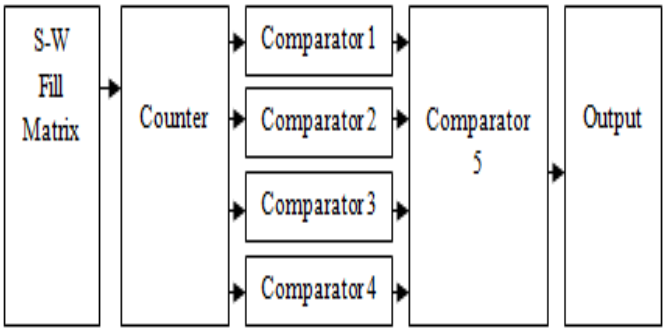

Figure 6: Architecture of Design 1.

Based on Figure 6, traceback process is to determine the highest score from each row of comparator. The comparator will check first condition. If first condition states the correct condition it will choose it as output. After each comparator design meet the requirement it will go to the next comparator 5 to compare all output based on output from comparator 1, comparator 2, comparator 3 and comparator 4 to form the output. However if it does not met the requirement, then it goes to the next condition till it finds the correct value like in Figure7, 8, 9 and 10 below.

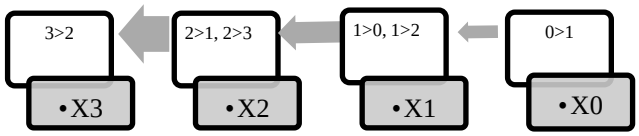

Figure 7: State Diagram Comparator 1

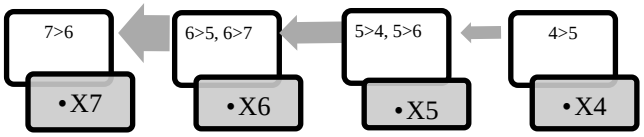

Figure 8: State Diagram Comparator 2

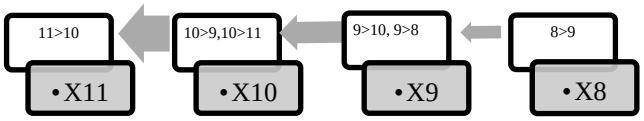

Figure 9: State Diagram Comparator 3

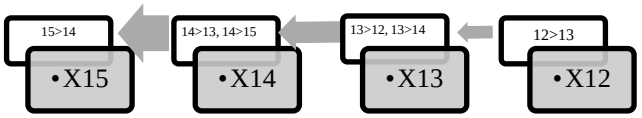

Figure 10: State Diagram Comparator 4

b. Design 2

In Design 2, the design is using only one Comparator. The design 2 is design as in Figure 10. The design block diagram for Design 2 consist only one Comparator.

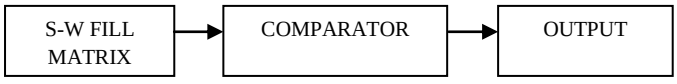

Figure 9: Architecture of design 2

.

Trace back process in design 2, will check from first input condition. If this first condition is correct, it will routinely form as output. Actually the step in design 2 is the same as design 1 but in for design 2 it will combine all condition under comparator. Then, the comparator will carry out with highest score from the output. Figure11 is an example for one input to compare with all other input and same goes to else input.

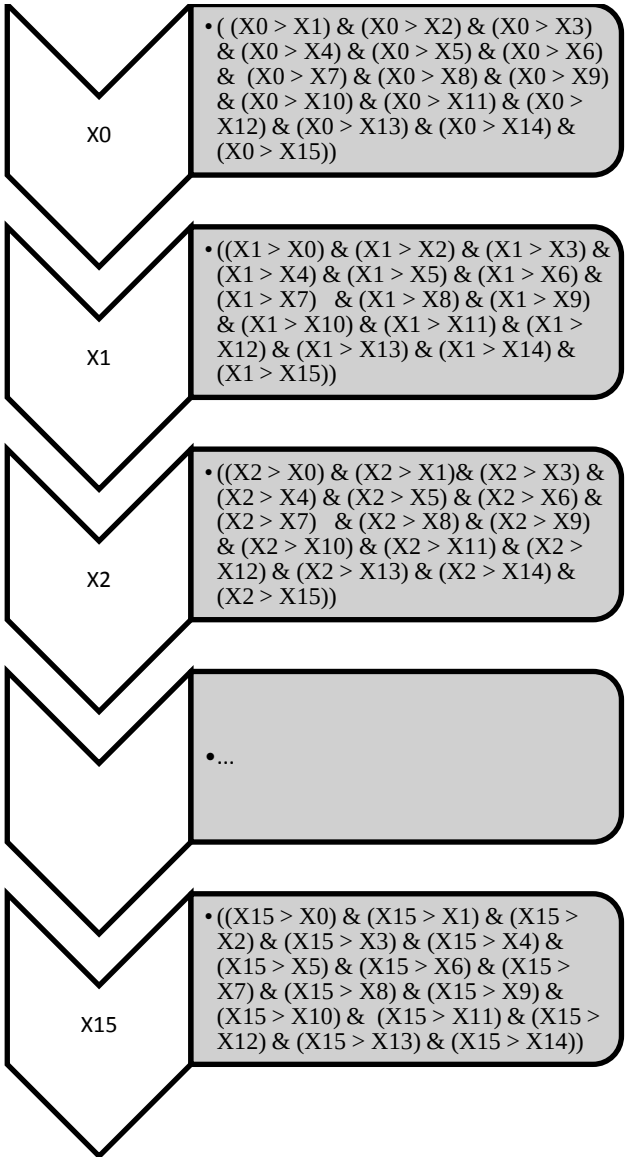

Figure 11: State diagram for design 2

VII. DISCUSSION AND RESULT

Theoretically, based on the research of this project there are two outputs. This part will discuss about the output that perform as in the final result. Both results will perform as in waveform 1 and waveform 2.

Design 1 and design 2 were synthesized using verilog syntax code. It formed using the Altera software tools. Once the synthesized process is completed, the RTL schematic diagrams would formed as in figure 12 and 13 below.

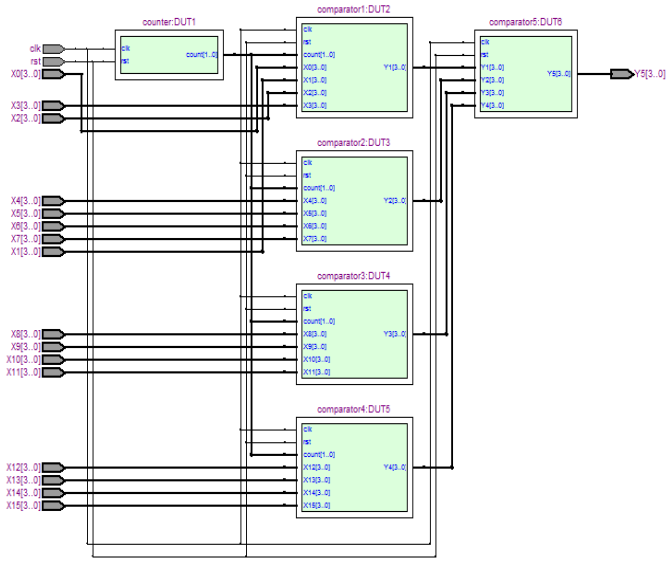

Figure 12: RTL schematic design 1

RTL schematic for the Design 1 (using 4 x 4 matrix size of comparator) is shown in Figure 12.

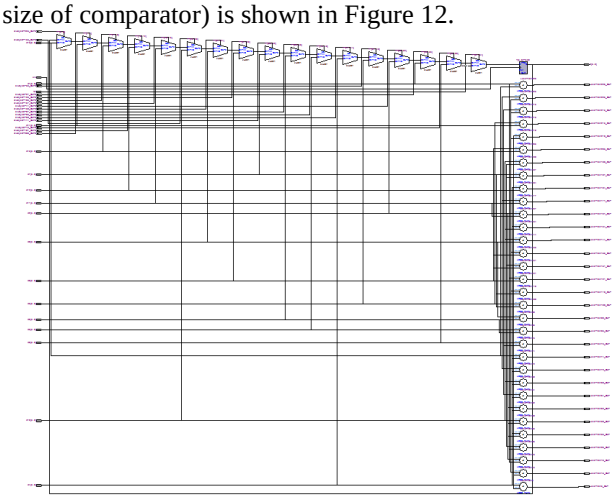

Figure 13: RTL schematic design 2

Next RTL schematic is for Design 2 (using 4 x4 combining all comparator) figure 13 shows the RTL schematic.

SEARCH SEQUENCE

TARGET SEQUENCE	-	A	C	A	C	A	C	T	A
-	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Table 3: Output of fill scoring matrix

Table 3 shows the fill matrix that has been added manually when needs to generate the waveform.

SEARCH SEQUENCE

TARGET SEQUENCE	A	C	T	A
C	6	7	6	5
A	9	8	7	8
C	8	11	10	9
A	10	10	10	12

Table 4: score 1 for 4x4 matrix

SEARCH SEQUENCE

TARGET SEQUENCE	C	A	C	T
A	4	5	4	3
C	7	6	7	6
A	6	9	8	7
C	8	8	11	10

Table 5: score 2 for 4x4 matrix

SEARCH SEQUENCE

TARGET SEQUENCE	A	C	A	C
C	2	3	2	3
A	5	4	5	4
C	4	7	6	7
A	6	6	9	8

Table 6: score 3 for 4x4 matrix

SEARCH SEQUENCE

TARGET SEQUENCE	C	A	C	A
G	1	1	1	1
C	3	2	3	2
A	2	5	4	5
C	4	4	7	6

Table 7: score 4 for 4x4 matrix

SEARCH SEQUENCE

TARGET SEQUENCE	A	C	A	C
A	2	1	2	1
G	1	1	1	1
C	0	3	2	3
A	2	2	5	4

Table 8: score 5 for 4x4 matrix

The output for both designs is the same, the only thing that are different about both designs is their processes to get the output. The optimal path for search sequence and target sequence is by using from table 4, 5, 6, 7 and 8.

INPUT		OUTPUT
Search	Target	
ACTA	CACA	C
CACT	ACAC	B
ACAC	CACA	9
CACA	GCAC	7
ACAC	AGCA	5

Table 9: Expected Output

By comparing without using simulation software, the expected output values for both designs are as in Table 9 above. After simulate the waveform output should get the same as the output result.

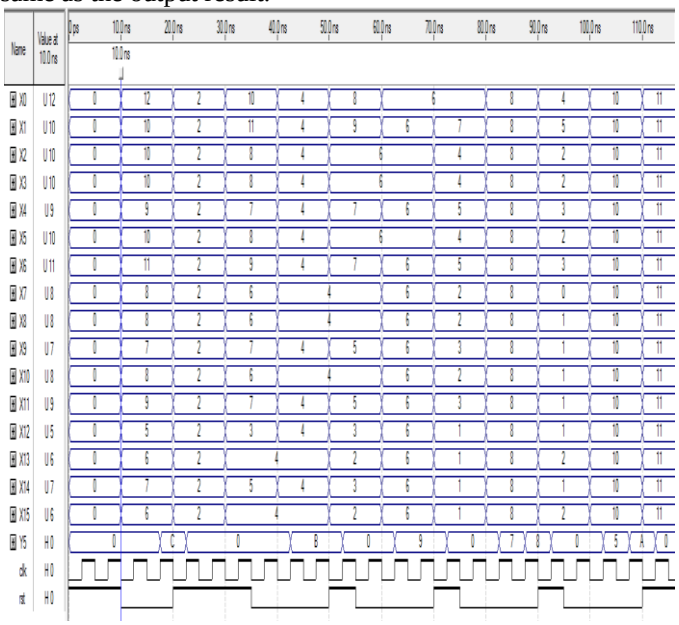

Figure 13: Output waveform design 1

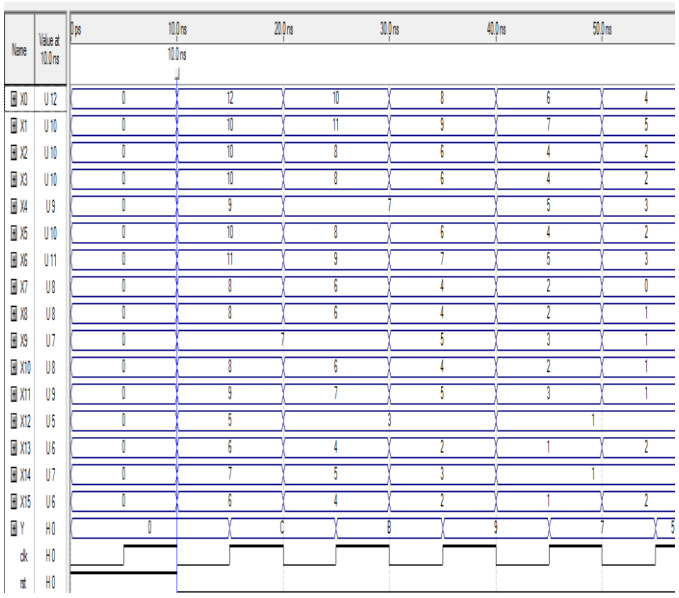

Figure14: Output waveform design 2

Table 10 and 11 shows that there are two designs that has been developed in the result. Table 10 is the first design and table 11 is the second design and all the result above was implemented by using in form of Verilog code in Quartus II (version 9.1). The target device that will be use for this implementation is Cyclone II (EP2C35F672C6) and lastly the simulation process is verified using simulator tools in Quartus II (version 9.1). Moreover to implement the output waveform, the input clock that will be use for the target Cyclone II is 100MHz.

MATRIX SIZE (4x4)	CYCLONE II (EP2C35F672C6) Total Logic Element
1	70/33216(1%)
2	144/33216(1%)
3	242/33216(1%)
4	318/33216(1%)

Table 10: Design 1

MATRIX SIZE (4x4)	CYCLONE II (EP2C35F672C6) Total Logic Element
1	835/33216(3%)
2	835/33216(3%)
3	2492/33216(8%)
4	3323/33216(10%)

Table 11: Design 2

.

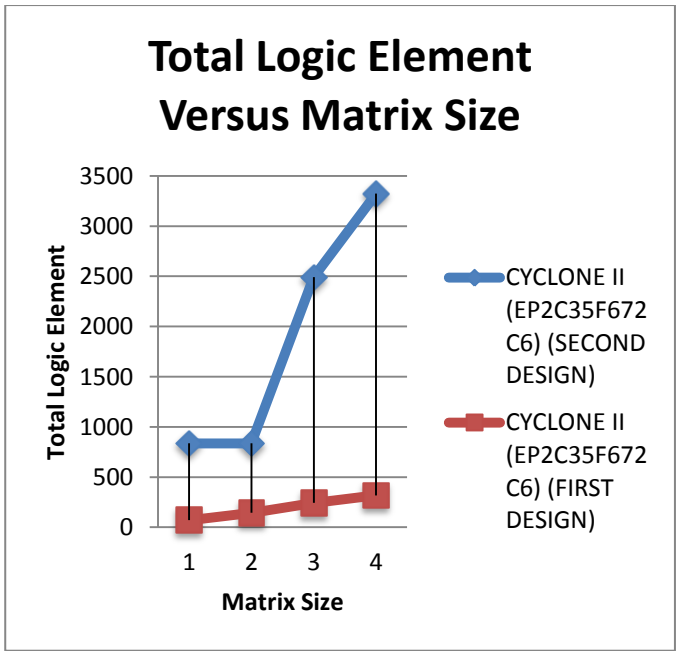

Figure 15: Total Logic Element

The graph total logic element output for both design1 and design 2 are shown in Figure 15. For the first design the total logic element increase when the matrix size increase. For the second design, at first it has the same value than it increases drastically. Both have different result due to the differences in the process of design. It also consist different performance for both designs since it depend on logic area design. Hence, Design 2 is slower than design 1 due to higher value of total logic element.

Timing Analysis Parameters and its Descriptions.
Clock Setup Time, t_{su} is when the minimum length of time that stabilized before clock rising
Clock to Output, t_{co} is the maximum length of time to come out with the output.
Clock Hold Time, t_h is the minimum length of time that stabilized after clock rising.

The timing analysis and parameter above is the description for table 12 and 13.

Type	Top1	Top2	Top3	Top4
Worst-case, tsu	9.850ns	9.508ns	10.152ns	10.937ns
Worst-case, tco	6.918ns	7.923ns	7.295ns	7.689ns
Worst-case, th	-3.317ns	0.668ns	0.471ns	0.295ns

Table 12: Worst-case timing analysis for Design 1

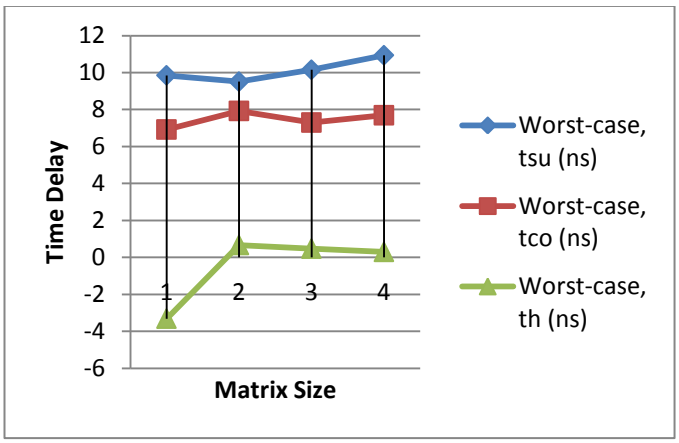

Figure 16: Time delay versus matrix size Design 1

Type	Top1	Top2	Top3	Top4
Worst-case tsu	16.276ns	17.256ns	16.570ns	17.532ns
Worst-case tco	8.747ns	7.796ns	8.568ns	8.710ns
Worst-case th	-0.760ns	-2.029ns	-0.257ns	0.399ns

Table 13: Worst-case timing analysis for Design 2

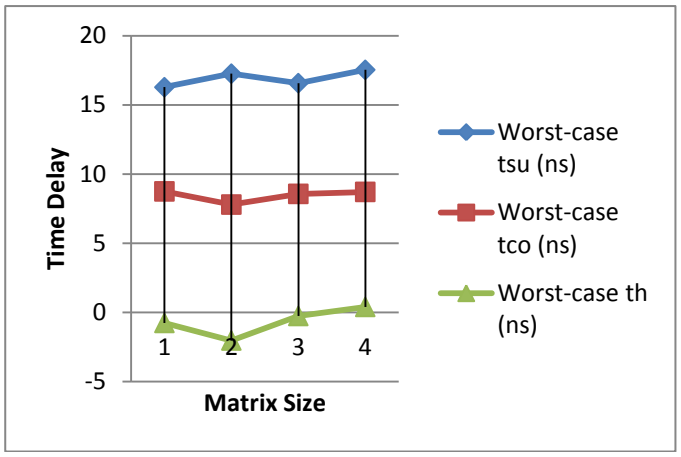

Figure 17: Time delay versus matrix size Design 2

Based on table 12 and table 13, the timing analysis for design 1 is lesser than design 2. Figure 17 and 18 is the time delay versus the matrix size for design 1 and design 2 based on the table 12 and 13.This timing happens since in Design 2 architecture consist longer critical path. This critical path causes the input drive the data in a longer way. Therefore Design 1 is way faster than design 2.

VIII. CONCLUSION

Overall, this paper is successful because the output with traceback scanning optimal point is achieved and the result is the same as the expected output. The input was taken by the previous researcher in using S-W algorithm however the optimal traceback has been developed. There are two results that have been achieved and it has been optimize by applying clock and reset. Moreover, design 1 has better performance compare to design 2. It proves that simple design architecture can reduce the traceback scanning delay time. Thus it can be concluded that design 1 has better performance in terms of speed and area.

IX. RECOMMENDATION

Based on the result, there are still many things to be improved for the traceback design. For further improvement, going to look into other or another different strategy, varies the technique of optimal path traceback and proposed it in the future by creating new system as well as to find other optimization DNA sequences trace back design with less total logic element that would increased the speed and can perform well in terms of its performance. Hence, it's recommended to improve the design by constructing a new coding system because there were no fixed techniques to come out with the best result and implement it using Quartus 2 version 9.1.

REFERENCES

[1] Liaq Hasan; Zaid Al-Ars, Zubair Nawaz, Koen Bertels "Hardware implementation of the Smith-Waterman Algorithm using recursive variable expansion" Proceedings of 3rd Inernational Design and Test Workshop IDT08, Monastir,2008

[2] Al Junid, Haron, Abd Majid, Osman, Hashim, Idros, Dohad, "Optimization of DNA Sequences Data for Accelerate DNA Sequences Alignment on FPGA" Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, 2010

[3] Zubair Nawaz, Koen Bertels, H. Ekin Sumbul, "Fast Smith-Waterman hardware implementation," ipdpsw, pp.1-4, 2010 IEEE International Symposium on Parallel&Distributed Processing, Workshops and Phd Forum, 2010

[4] Scott Lloyd and Quinn O.snell "Hardware accelerated sequence alignment with trace back", International Journal of Reconfigurable Computing Volume, 2009

[5] Jacop Yanto; Tomothy F.Oliver; Bertil Schmidt; Douglas L.Maskell "Biological sequence analysis with hidden markov model on an FPGA", Transactions on Information Technology in Biomedicine 13(5): 740-746 (2009)

[6] Saul Needleman and Christian Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." J Mol Biol. 48(3):443-53,(1970)

[7] Isaac TS Li, Warren Shum and Kevin Truong "160-fold acceleration of the Smith Waterman algorithm using a field programmable gate array (FPGA)" BMC Bioinformatics, 01/2007

[8] D.J. Lipman and W.R. Pearson."Rapid and sensitive protein simlarity searches". Science 227:1435.1441, 1985.

[9] Gish W. Miller W. Myers E. W. Altschul, S. F. and D. J. Lipman. "A basic local alignment search tool". J. Mol. Biol., 215:403.410, 1990

[10] Jacop Yanto, Timothy F. Oliver, Bertil Schmidt, Douglas L. Maskell "Biological Sequence Analysis with Hidden Markov Models on an FPGA". Asia-Pacific Computer Systems Architecture Conference : 429-439, 2005