Optimizing Optimal Path Trace Back System for Smith-Waterman Algorithm using Structural Modelling Technique

Nur Farah Ain Saliman
Faculty of Electrical Engineering
Universiti Teknologi MARA,
40450, Shah Alam, Selangor, Malaysia.
E-mail: ain_saliman@yahoo.com

***Abstract* - This technical paper was about optimizing optimal path trace back system for Smith-Waterman Algorithm using Structural Modelling Technique. The objectives for this paper are to optimize the best trace back scanning performance and also to design the simple architecture in order to reduce the runtime. It focuses on the complexity in the trace back scanning architecture design based on structural modelling technique. Due to the increasing number of the population, it also affected the increases number of probability in Deoxyribonucleic Acid (DNA) sequences alignment. This will cause disadvantages in the performance and speed level of the Smith-Waterman Algorithm. Thus, based on other researchers there was various kind of method being use in order to overcome this problem. Then, the structural modelling is being use in this project in order to trace back by scanning the Smith-Waterman matrix calculation and find the optimal path. The paper was to study on theoretical and a simulation technique to complete the result. At the end of the analysis, simple design architecture output runtime is reducing up to 50%. This technique code is written in Verilog HDL syntax using Quartus 2 (version 9.1) and the simulation is verified using Quartus 2 (version 9.1) simulator tools.**

***Keywords* — DNA sequence alignment, Smith-Waterman Algorithm (SWA), Structural Modelling and Trace Back.**

I. INTRODUCTION

In simple terms, DNA sequences alignment is comparison between two DNA sequences layer [4]. Smith-Waterman Algorithm (SWA) being used based on the fast implementation [5]. Nowadays, the size of population was slightly increasing therefore it also drags along the increasing number of probability in DNA sequences alignment [2]. Due on this matter it comes out with two problems which is in runtime and memory space [3]. Even though Smith-Waterman algorithm (SWA) already stable, but since size of population increasing it makes this algorithm and trace back speed slow [3-6]. The objective for this paper is to optimize the best trace back scanning performance based on the simple design architecture. This design will help to reduce the runtime and increases the speed level of the trace back scanning performance.

First project team which are lead by Zubair Nawaz, Mudassir Shabbir, Zaid Al-Ars and Koen Bertels was proposed their project on finding an optimal alignment between two sequences. This optimal alignment will be compare using two different length of DNA sequences alignment. Present new method on parallelize the algorithm. This method helps to speed up the system up to 1.55 times [8].

Second researcher group member Nuno Sebasti˜ao, Tiago Dia, Nuno Roma and Paulo Flores is proposed to find the optimal sequences alignment and enhanced in trace back phase. Smith-Waterman Algorithm (SWA) is widely used in this paper. The technique being use is innovative technique. New hardware architecture and Leon3 Processor is being use to speed up into more 6000 times than original Smith-Waterman Software [7].

The section was organized as following: Section 1: Introduction, Section 2: Smith-Waterman Algorithm, Section 3: Modelling structural for trace back method, Section 4: Methodology, Section 5: Discussion and Result and Section 6: Conclusion.

II. SMITH-WATERMAN ALGORITHM

In DNA researcher area, Dynamic Programming Algorithm (DPA) was widely being used [5]. It was widely used since it can find the most similar pair of DNA sequence alignment [5]. Under Dynamic Programming Algorithm (DPA) there were another two best algorithms which are known as Needleman-Wunsch Algorithm (NWA) and Smith-Waterman Algorithm (SWA) [5]. In this paper Smith-Waterman Algorithm (SWA) is being chosen. This is because Smith-Waterman Algorithm (SWA) can break the DNA sequences alignment into short steps by picking only the best DNA sequences alignment in order to find the optimal alignment, while for Needleman-Wunsch Algorithm (NWA), it is still best in finding optimal path but it will search this through the original DNA sequences alignment [6]. Thus, Smith-Waterman Algorithm (SWA) step can be elaborate based Figure 1:

Initialization
Fill Matrix
Trace Back

Figure 1: Smith-Waterman Algorithm flow process

A. *Initialization*

First step will be on initialization module. This module is to carrying out and to optimize the DNA sequences alignment data. In order to save memory space, reducing the number of bit binary is the best way. Therefore, data minimization method is the applicable method in this stage. This method will reduce the standard number of bit for one DNA character. The DNA characters is assigned as in Table 1 , where A will reduce into "00" while C, G and T as "01", "10" and "11".

Name	Actual Data	Reduce Data
Adenine (A)	01000001	00
Cytosine (C)	01000011	01
Guanine (G)	01000111	10
Thymine (T)	01010100	11

Table 1: Proposed DNA sequence characters reduction data assignment.

B. *Fill Matrix*

Second step will be on fill matrix module. Fill matrix module was conduct on score calculation and fill in the matrix. For an example the Smith-Waterman Algorithm (SWA) consist of two sequences as in Figure 2. While Table 2 shown the complete result for matrix fill.

Search Sequence (A_x) A T C T C G T A T
Target Sequence (B_y) G T C T A T C A C

Figure 2: Two DNA sequences alignment for Search and Target Sequences

Search Sequence

Target Sequence

	∅	A	T	C	T	C	G	T	A	T
∅	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	2	1	0	0
T	0	0	2	1	2	1	0	4	3	2
C	0	0	1	4	3	4	3	3	3	2
T	0	0	2	1	6	5	4	5	4	5
A	0	2	2	2	5	5	4	4	7	6
T	0	1	4	3	4	4	4	6	6	9
C	0	0	3	6	5	6	5	5	5	8
A	0	2	2	5	5	5	5	4	7	7
C	0	1	1	4	4	7	6	5	6	6

Table 2: Smith-Waterman Algorithm output matrix table

Target Sequence	**Search Sequence**	
	NW	N
	W	score

Table 3: Score output calculation

Figure 3 above shown draft matrix calculation. Firstly, imagine that Search Sequence character same as in Target Sequence character. At that point, score will calculate as NW value plus with match value [1].

If Search Sequence and Target Sequence character is mismatch, the score will use the mismatch value plus with highest value from one of these three NW, N or W.

C. *Trace back*

Last module is on trace back scanning module. This module is to find the optimal path trace back depends on the score in the matrix table. The optimal path trace back scanning will trace from the optimal score until the lowest score. This trace back scanning process can be seen based on the different colour of tone (dark tone colour into light tone colour) as in Table 4.

Search Sequence

Target Sequence

	∅	A	T	C	T	C	G	T	A	T
∅	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	2	1	0	0
T	0	0	2	1	2	1	0	4	3	2
C	0	0	1	4	3	4	3	3	3	2
T	0	0	2	1	6	5	4	5	4	5
A	0	2	2	2	5	5	4	4	7	6
T	0	1	4	3	4	4	4	6	6	9
C	0	0	3	6	5	6	5	5	5	8
A	0	2	2	5	5	5	5	4	7	7
C	0	1	1	4	4	7	6	5	6	6

Table 4: Trace back scanning table

III. STRUCTURAL MODELLING

Structural modelling is very important in order to design smooth trace back system. This system is design using Finite State Machines (FSM) module. Finite State Machines (FSM) module consists of inputs, outputs and state. The way Finite State Machines (FSM) work is depending on clock edges (positive or negative edge). Finite State Machines (FSM) module can be classified into two ways. It was Moore machine and Mealy machine. Both machines were depending on architecture design specification. Moore machine and Mealy machine module can be determined as in Figure 3 and Figure 4.

A. *Moore Machine*

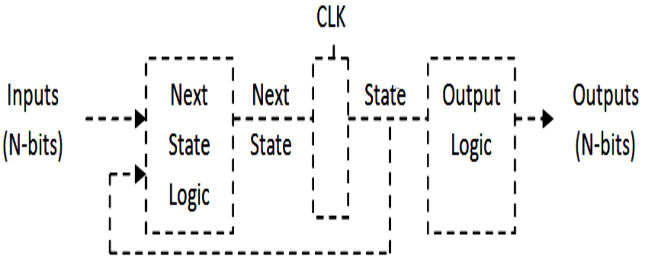


Figure 3: Moore Machine

The difference between Moore and Mealy machine is the way to perform the output result. Moore machine output was relying on current state. Thus, to go to the next state logic it will rely on current state.

B. *Mealy Machine*

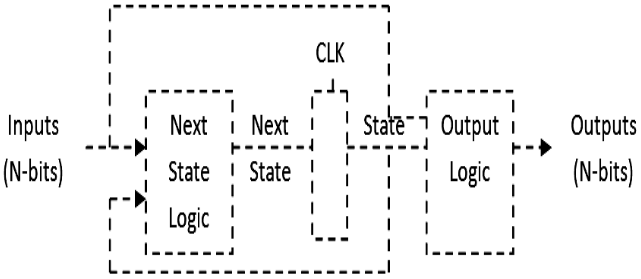


Figure 4: Mealy Machine

For second Finite State Machine (FSM) module is Mealy machine. Its output was depending on the current state and also current input. In this paper, both architecture designs are using Mealy machine since the output is depending on current state and current input.

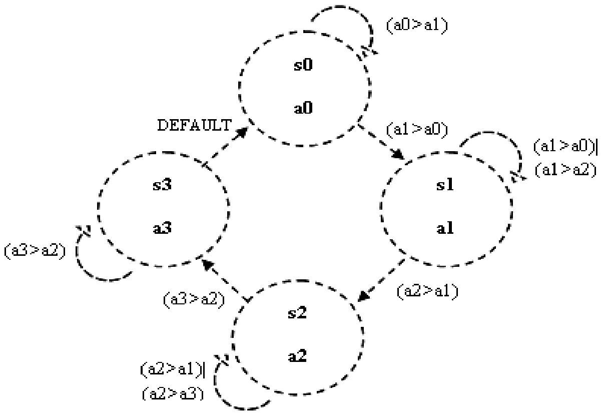


Figure 5: State Diagram

To implement using Finite State Machine (FSM) module, first need to produce state diagram based on design needed as in Figure 5. In order to perform the structural modelling, state diagram is very important. Since state diagram can be guide to come out with the design architecture code.

IV. METHODOLOGY

The method in this section will cover two simple architecture designs. This architecture design will conduct and produce two different output trace back scanning. Three example 4 x 4 matrix table will be use to examine through both architecture designs. Both were design based on Smith-Waterman Algorithm (SWA) to support calculation fill matrix value. The main architecture design will be connected in parallel as in Figure 6 below:

Fill Matrix
Trace Back

Figure 6: Parallel architecture for trace back

Based on basic Smith Waterman block diagram above, the fill matrix module is to calculate all matrix score. This calculation is to perform trace back scanning. Those score actually already being calculated by previous researcher. Design 1 and Design 2 will cover using 4 x 4 Matrix size of table as in Table 5 below:

	Search Sequence			
Target Sequence	s15 a15	s14 a14	s13 a13	s12 a12
	s11 a11	s10 a10	s9 a9	s8 a8
	s7 a7	s6 a6	s5 a5	s4 a4
	s3 a3	s2 a2	s1 a1	s0 a0

Table 5: State and score value for 4x4 matrix table

A. *Design 1*

In Design 1, the Finite State Machine (FSM) method for 4 x 4 matrix table will be break into 1 single line. Thus, four Finite State Machine (FSM) modules will be use. The main architecture Design 1 will be connected in parallel as in Figure 7 below. The design block diagram consists of four FSM stage and combine module.

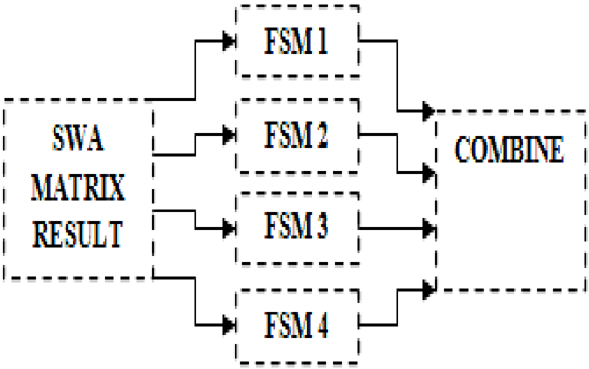


Figure 7: Design 1 architecture

Trace back scanning process will determine the highest score from each row. Firstly, Finite State Machine 1 (FSM1) will check first input score whether it meet the right condition or not. If this input meet the right condition it will automatically form as output. But if first input did not meet the right condition, then it will check next stage until it find right condition. Then, each Finite State Machine (FSM) stage will carry out with all highest score for the output. While for combine module it will organize the score from highest into lowest score. FSM 1, FSM 2, FSM 3 and FSM 4 will design as in Figure 7. The FSM code need to carry out depends on state diagram as in Figure 8.

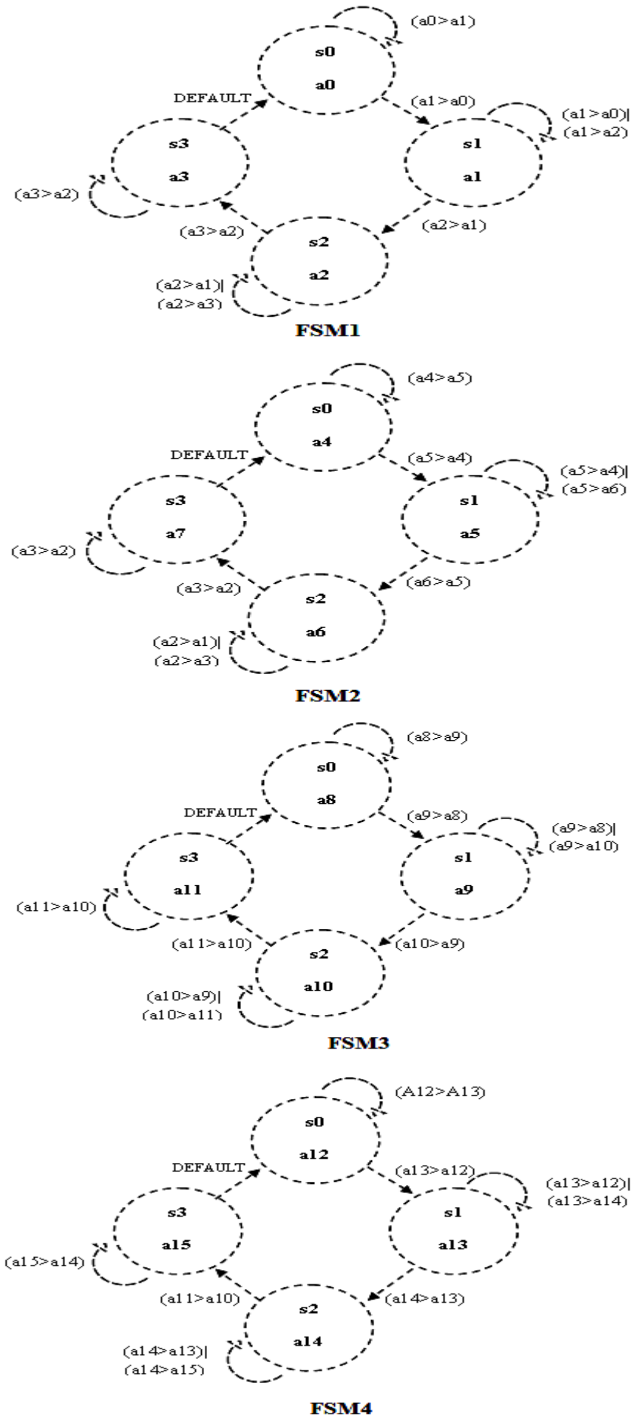


Figure 8: State diagram for FSM in Design 1

B. Design 2

Design 2 will be connected in parallel as in Figure 9. Based on design block diagram, Design 2 consist only one Finite State Machine (FSM) module. The input signals will form using Smith-Waterman score. The process stills same as in Design 1, but it slightly different in Finite State Machine (FSM) design and score output.

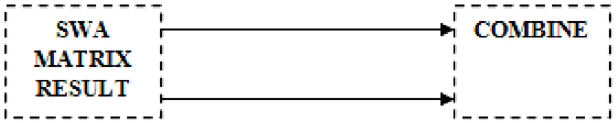


Figure 9: Design 2 architecture

For Design 2 process, it will work start from first input score stage. This stage will check whether it meet the right condition or not. It will automatically form as output if this stage meets the right condition. Each stage will depend on nearest stage condition in order to perform the output. Finally, Finite State Machine (FSM) output will carry out with highest score. State diagram in Figure10 show the state diagram for Design 2. Figure 10 also attach in appendix.

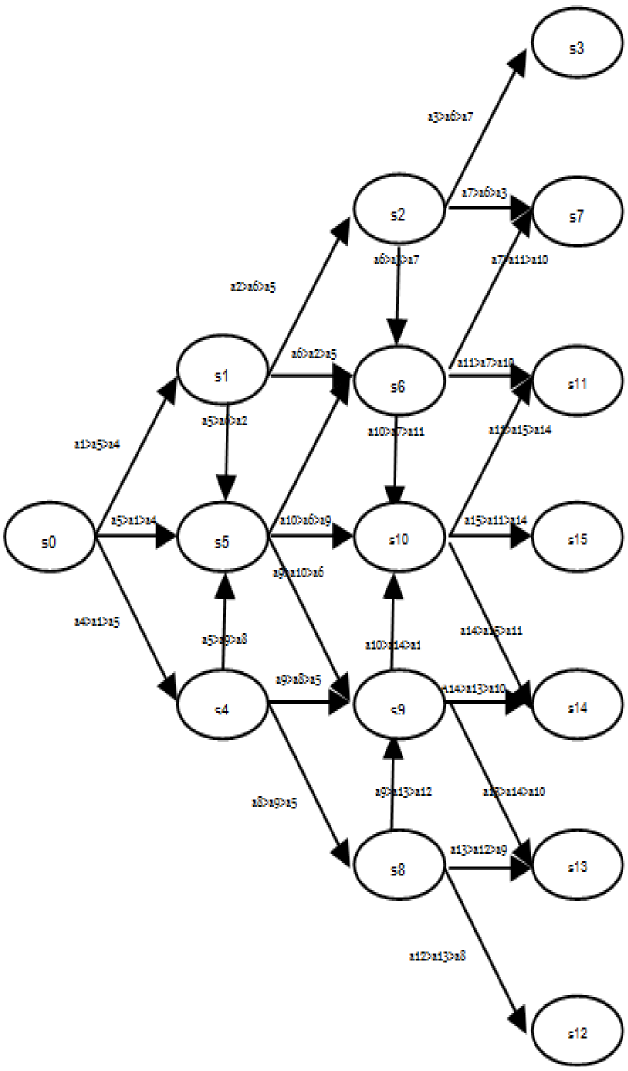


Figure 10: State diagram for FSM in Design 2

V. DISCUSSION AND RESULT

A. Theoretical

This section will be discuses the output. Finally this project comes out with two outputs as shown in Waveform 1 and Waveform 2. The Design 1 and Design 2 were synthesized in Verilog code forms that are using Altera software tools. In this paper, the systems were writing and simulate in form of Verilog code. After synthesized process is completed, the RTL schematic diagram was produced as in figure below. Figure 11 is RTL schematic for the Design 1 (using 4 x 4 matrix size which is break the FSM into 4 module and the Design 1 is attach in appendix) and Figure 12 is RTL schematic for Design 2 (using 4 x4 FSM module and the Design 2 is attach in appendix). The output process for Design 1 is formed as in Waveform 1. It means the output come out with sequences value from optimal score into lowest score. Meanwhile for the Design 2 output it comes out directly with the optimal score as shown in Waveform 2.

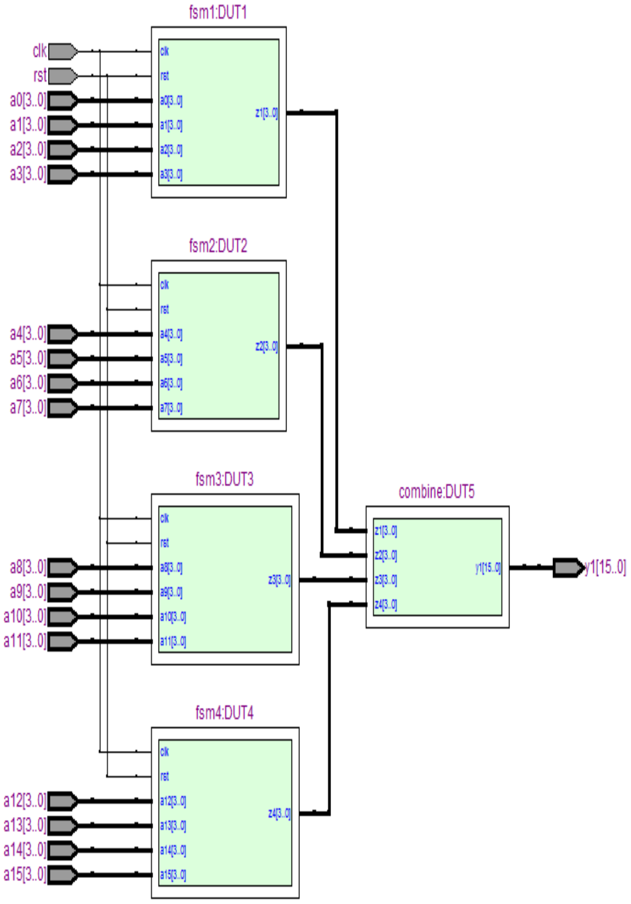


Figure 11: RTL for Design 1 (4 x 4 matrix size)

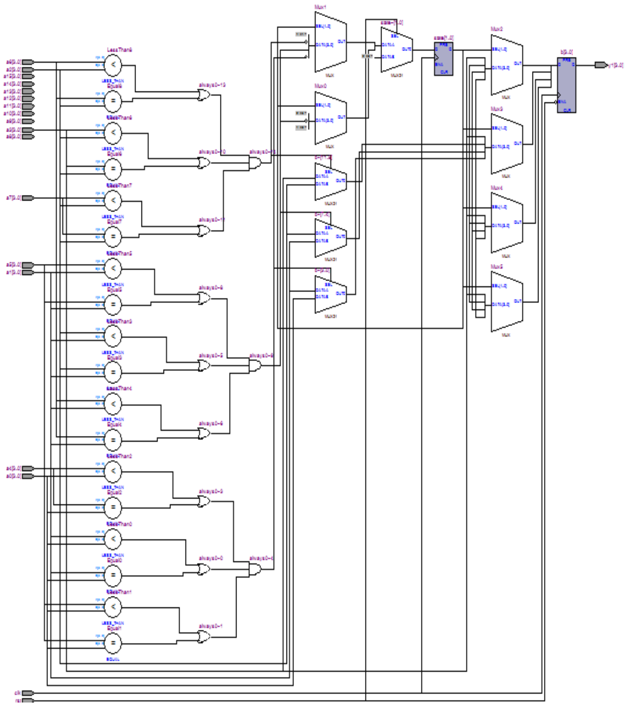


Figure 12: RTL for Design 2 (4 x4 matrix size)

After the RTL schematic was constructed correctly, both designs will be verified using tool simulator. The inputs will take randomly using Table 6, Table 7 and Table 8 score matrix. Then, for the expected result for the outcome supposedly need to get as in Table 9 for Design 1 while for Design 2 as in Table 10. This three different inputs table for Search Sequence and Target Sequence DNA will form the output as in Waveform1 (Design 1) and Waveform 2 (Design 2).

		Search Sequence			
		G	T	A	T
Target Sequence	C	3	3	3	2
	T	4	5	4	5
	A	4	4	7	6
	T	4	6	6	9

Table 6: 4 x 4 matrix table score 1

		Search Sequence			
		G	T	A	T
Target Sequence	C	1	0	4	3
	T	4	3	3	3
	A	5	4	5	4
	T	5	4	4	7

Table 7: 4 x 4 matrix table score 2

		Search Sequence			
		G	T	A	T
Target Sequence	C	0	0	2	1
	T	2	1	0	4
	A	3	4	3	3
	T	6	5	4	5

Table 8: 4 x 4 matrix table score 3

The waveform should get the output same as the expected result. As expected result table 9 shows that the trace back scanning score from highest to lowest scores same as in Waveform 1. Then, Table 10 show the expected result for each optimal point that need to get is 9 from Table 6, 7 from Table 7 and finally 5 from Table 8.

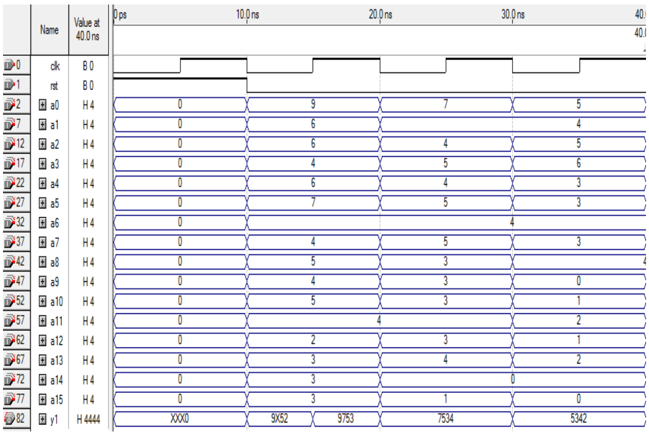
INPUT		OUTPUT
Search	Target	
GTAT	CTAT	9753/ 9X53
CGTA	TCTA	7534
TCGT	GTCT	5342

Table 9: Expected Output Result for Design 1

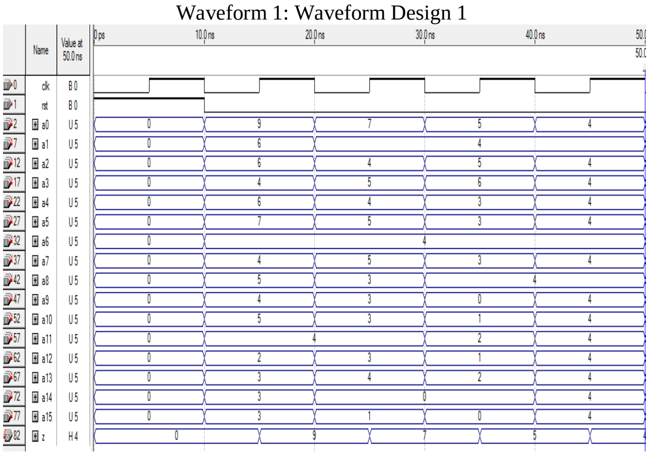
INPUT		OUTPUT
Search	Target	
GTAT	CTAT	9
CGTA	TCTA	7
TCGT	GTCT	5

Table 10: Expected Output Result for Design 2

All the waveform comes out same as the expected result. Table 9 the waveform show that the output was trace from the optimal score into lowest score. In Table 10, came out with the score of the optimal alignment.

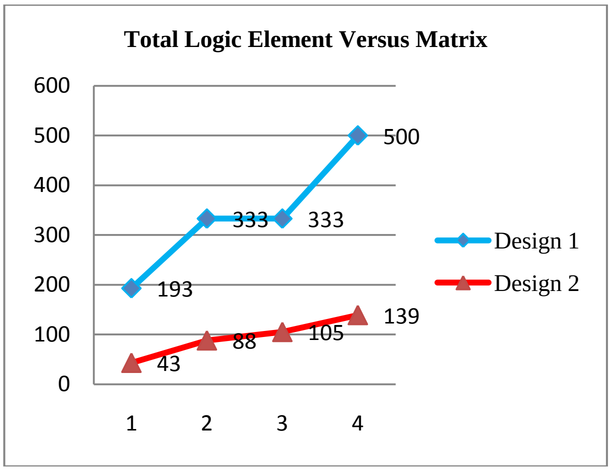


Waveform 1: Waveform Design 1



Waveform 2: Waveform Design 2

Each Waveform 1 and Waveform 2 also attach in appendix.



Graph 1: Logic element versus matrix size for Design 1 and Design 2

Matrix Size (4 X4)	Design 1	Design 2
1	193/ 33216 (1%)	43/ 33216 (1%)
2	333/ 33216 (1%)	88/ 33216 (1%)
3	333/ 33216 (1%)	105/ 33216 (1%)
4	500/ 33216 (2%)	139/ 33216 (1%)

Table 11: Total logic element for each design in different matrix size

From graph 1 find that total logic element for Design 1 is drastically increase at range between 140 until 167 logic element for each size matrix (one of 4 x 4 matrix size, two of 4 x 4 matrix size, three of 4 x 4 matrix size and four of 4 x 4 matrix size). While for Design 2, the total logic element does not increase drastically as Design 1. In addition, the theoretical results for this paper are to optimize the performance for each design. Since each design came out with different number of logic element, it also causes different performance for each design. The performance for Design 1 is slow compare to Design 2. Therefore, by minimize the number of logic element is the best way to perform the best and faster output result.

Performances parameters:	
Clock Hold Time (t_H):	Minimum time of length the data must enable after the active clock edge.

Table 12: Introduction for parameters performance

Table 12, define list the performance parameters that need to carry out in order to find the optimizing the runtime for both designs.

Parameters	**Design 1**			
	1	**2**	**3**	**4**
t_{CO} (ns)	14.179	15.887	14.751	15.138

Table 13: Timing analysis for Design 1

Parameters	**Design 2**			
	1	**2**	**3**	**4**
t_{CO} (ns)	6.915	7.031	7.520	8.033

Table 14: Timing analysis for Design 2

Based on Table 13 and Table 14, it show that the timing analysis output for Design 1 and Design 2. Design 1 and Design 2 carries out with different timing analysis value since the architecture design was totally different. In this Table 13 and Table 14, the timing analysis is run for each size matrix for Design 1 and Design 2. Find out for all matrix size in Design 1 come with different value since use different matrix size. Same goes for Design 2. This matter is influence by the different size of architecture design. The delay time is increases due the increases size of the architecture design. The average clock-to-output delay Design 1 gives 14.989ns, while the average value for Design 2 is 7.374ns. It gives reducing for runtime output up to 50%. Conclude that, Design 2 is faster than Design 1 since the increment value delay for Design 2 is smaller than Design 1.

B. Implementation

This paper value was implement using 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 finally the simulation process is verified using simulator tools in Quartus II (version 9.1).

C. Simulation

For the simulation part, this project were using simulator tool in Quartus II (version 9.1) to implement the output waveform.

D. The study of implementation and simulation

Based on the implementation and simulation that are cover in this technical paper, find that there still need a lot of improvement in the trace back design. Thus, it can be said the technique can be improved and there were no fixed technique to come out with the best result.

VI. CONCLUSION

The output is successfully achieved with trace back scanning optimal point. The input of the data was taken from previous researches using Smith-Waterman Algorithm module. During analysis, the result of optimal point has compared with expected value. Based on the result output, it can be said that both designs already achieve the target same as the expected result. At the end, it can be conclude that the Design 2 is better than Design 1 since the speed level reduce up to 50%. Therefore it proves that, simple design architecture can reduce trace back scanning runtime output.

VII. RECOMMENDATION

For further improvement, it is proposed that future study look into the different strategy. Use in the form of systolic array, grid and tree structure as well so that it can come out the output as we want. Find other best trace back design that can perform well and increasing the speed performance.

REFFERENCES

[1] Isaac TS Li1, Warren Shum2 and Kevin Truong*(160-fold acceleration of the Smith-Waterman algorithm using a field programmable gate array (FPGA))

[2] Al Junid, S. A. M.; Haron, M.A.; Abd Majid, Z.; Osman, F.N.; Hashim, H.; Idros, M.F.M.; Dohad, M.R. (Optimization of DNA Sequences Data for Accelerate DNA Sequences Alignment on FPGA)

[3] Nuno Sebasti˜ao, Tiago Dias, Nuno Roma and Paulo Flores (Integrated Accelerator Architecture for DNA Sequences Alignment with Enhanced Trace back Phase)

[4] Scott Lloyd and Quinn O.Snell (Hardware Accelerated Sequence Alignment with Trace back)

[5] Ying Liu, Khaled Benkid, AbdSamad Benkrid and Server Kasap (An FPGA Web Server for High Performance Biological Sequence Alignment)

[6] Liaq Hasan, Zaid AlArs, Zubair Nawaz and Koen Bertel (Hardware Implemantation of Smith Waterman Algorith Usinf Recursive Variable Expension)

[7] Nuno Sebasti˜ao Tiago Dias Nuno Roma Paulo Flores (Integrated Accelerator Architecture for DNA Sequences Alignment with Enhanced Traceback Phase)

[8] Nawaz, Mudassir Shabbir, Zaid Al-Ars and Koen Bertel (Acceleration of Biological Sequence Alignmnet using Recursive variable Expansion)

[9] Xianyang Jiang, Xinchun Liu, Lin Xu, Peiheng Zhang, and Ninghui Sun (A Reconfigurable Accelerator for Smith–Waterman Algorithm)

[10] Syed Abdul Mutalib Al Junid, Zulkifli Abd Majid, Abdul Karimi Halim (Development of DNA Sequencing Accelerator Based on Smith Waterman Algorithm with Heuristic Divide and Conquer Technique for FPGA Implementation)

APPENDIXES

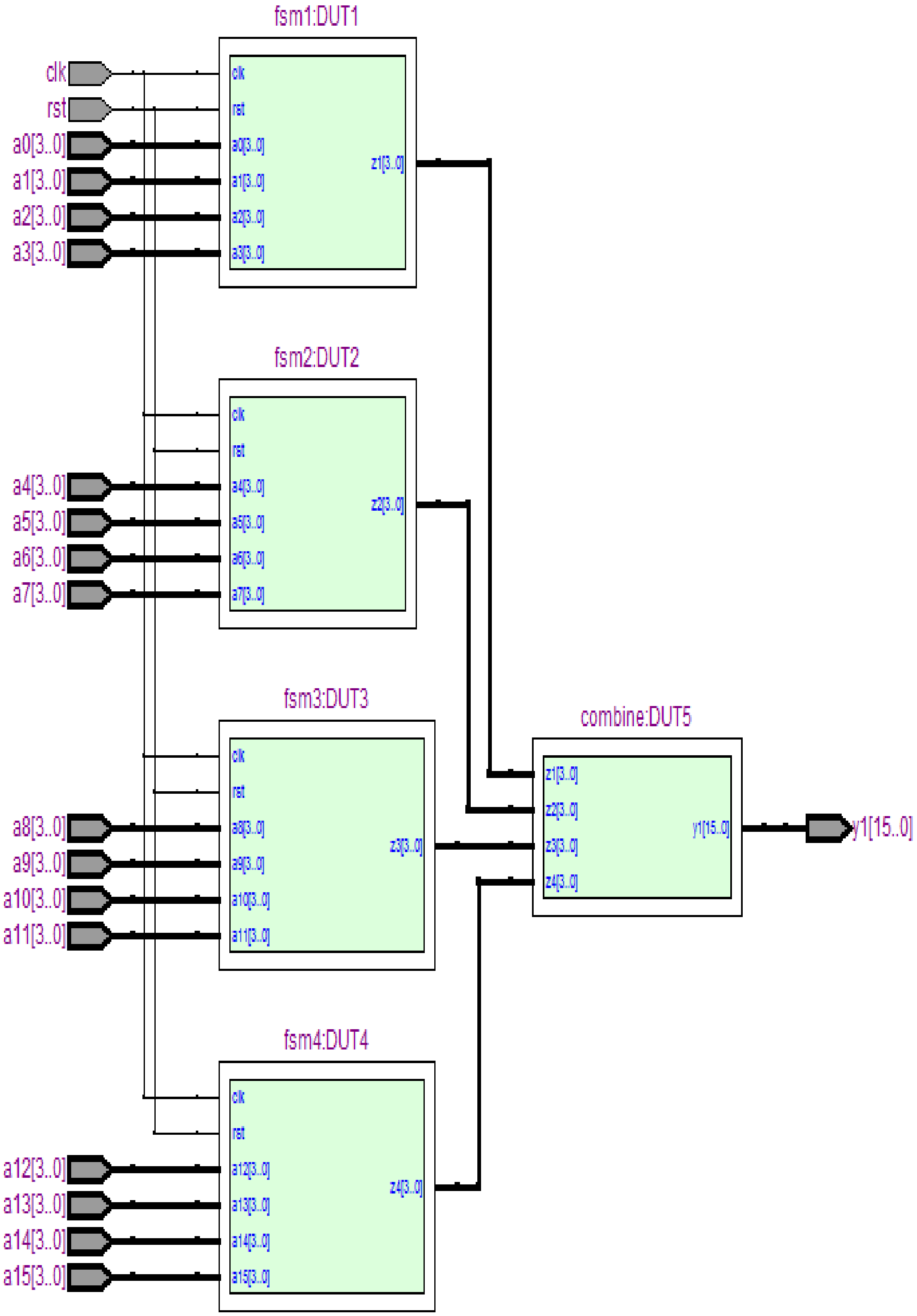


Figure 12: RTL for Design 1 (4 x4 matrix size)

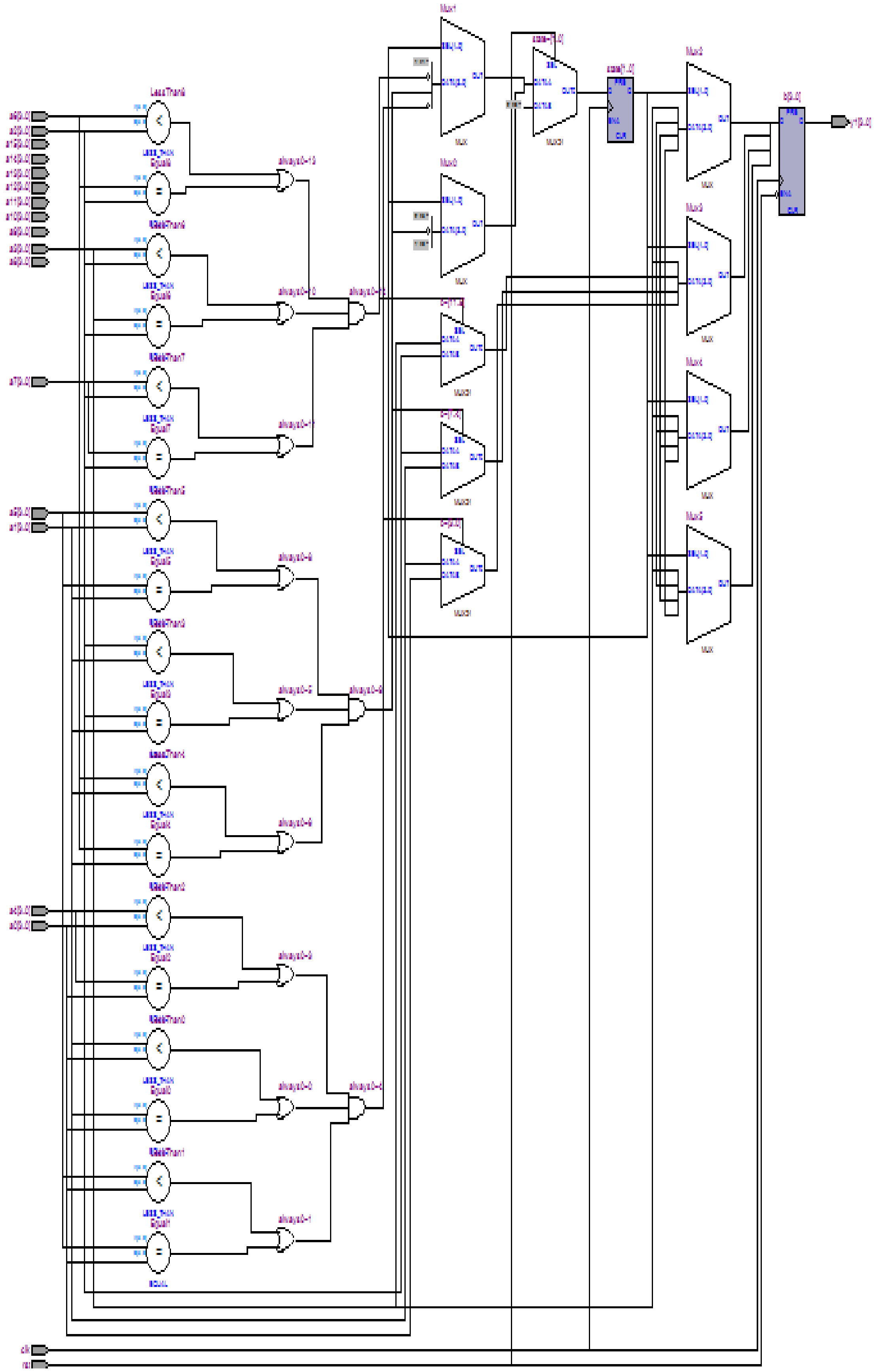


Figure 12: RTL for Design 2 (4 x4 matrix size)

.

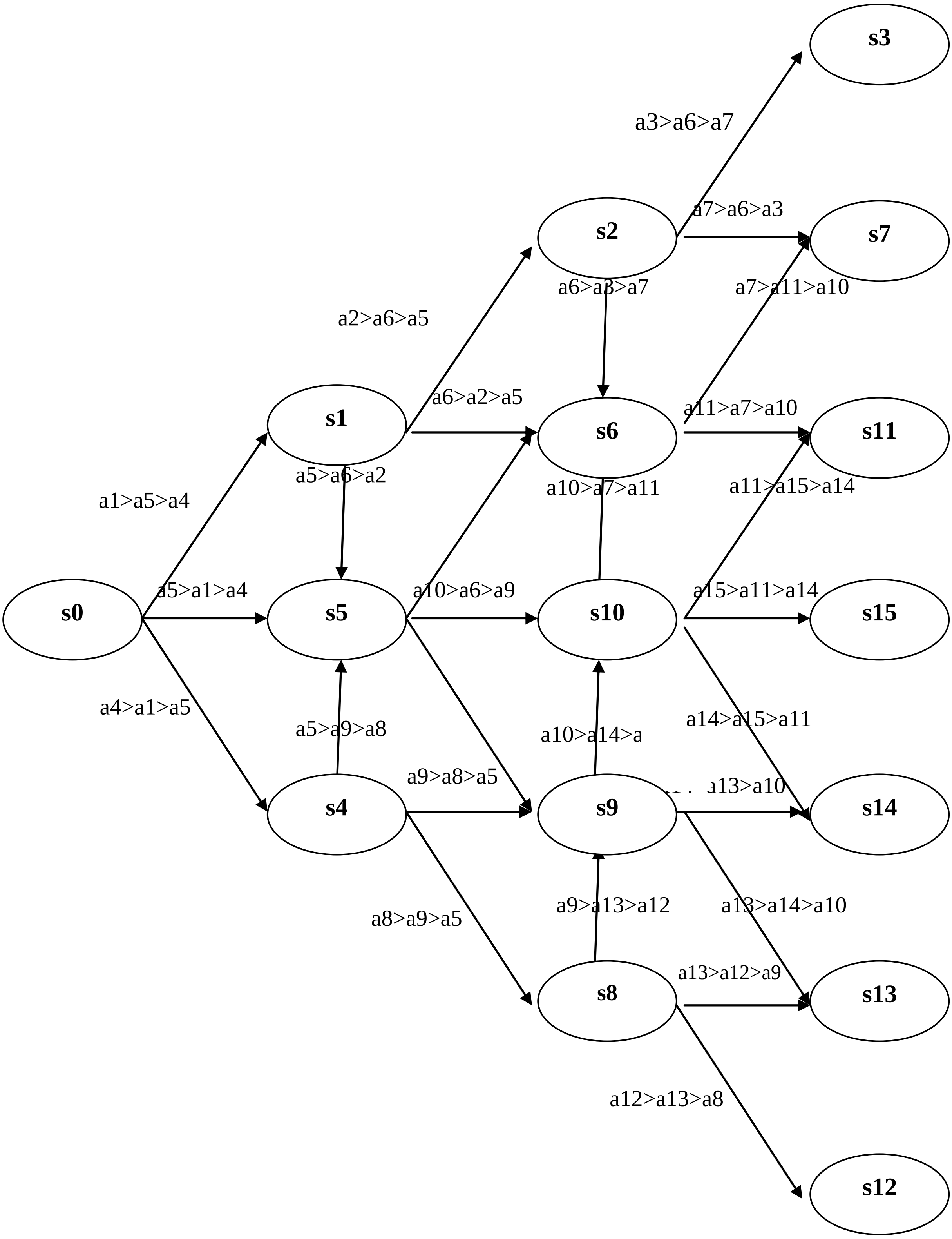
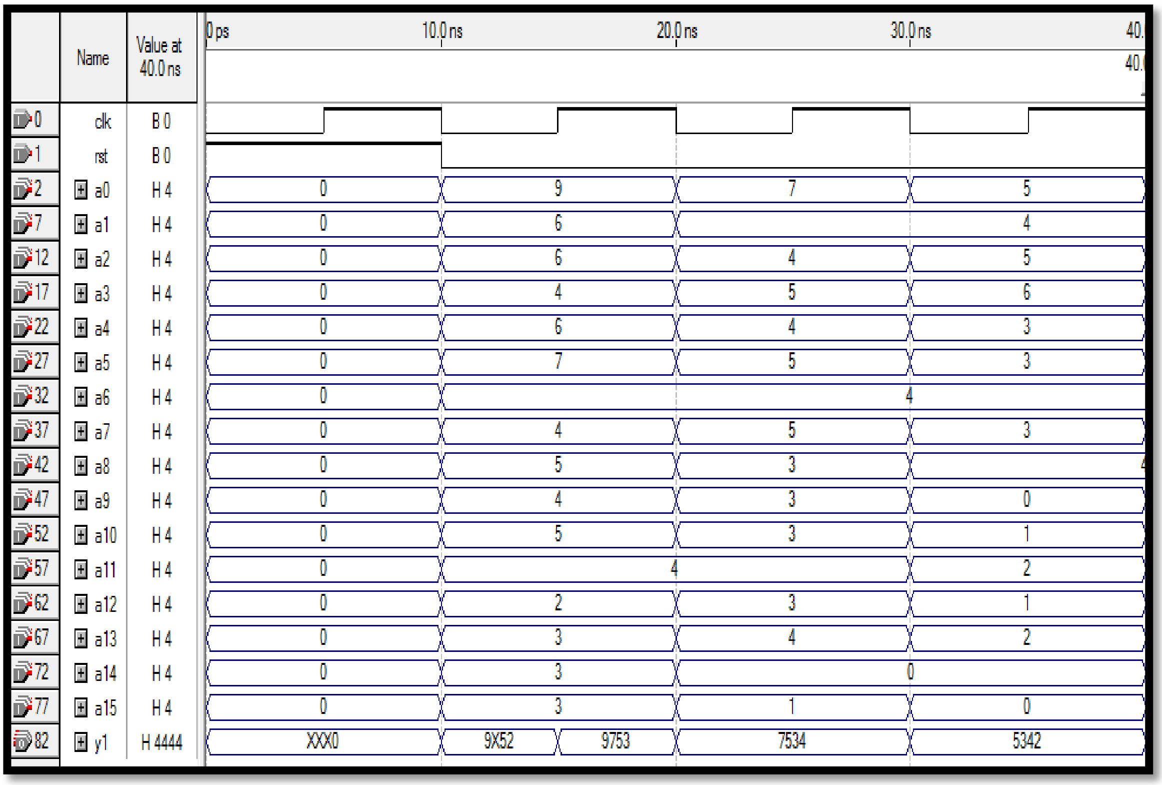
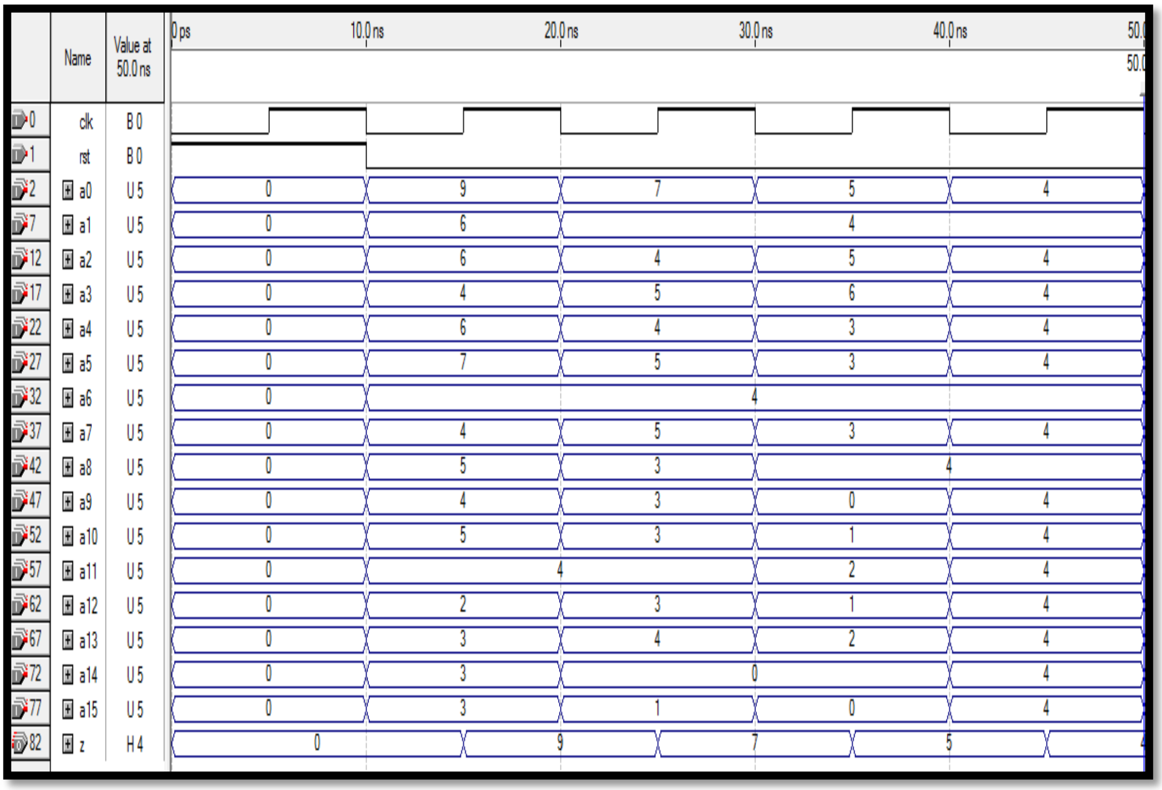


Figure 13: State diagram for FSM in Design 2.



Waveform 1: Waveform Design 1



Waveform 2; Waveform Design 2