Smith-Waterman Algorithm Traceback Optimization using Structural Modelling Technique

Nur Farah Ain Saliman*, Nur Dalilah Ahmad Sabri, Syed Abdul Mutalib Al Junid, Nooritawati Md Tahir, Zulkifli Abd Majid
Faculty of Electrical Engineering,
Universiti Teknologi MARA,
Shah Alam, Selangor, MALAYSIA
e-mail: ain_saliman@yahoo.com

***Abstract* — The optimal traceback procedure for the Smith-Waterman algorithm using structural modelling techniques presents in this paper. The proposed techniques been developed, compiled and simulated using Altera Quartus II Version 9.0 EDA Tools targeted to Cyclone II EP2C35 at 50MHz clock speed. Two designs evaluated. The first design use four Finite State Machines (FSM) connected to the Smith Waterman Algorithm (SWA). While, the second design use one FSM connected to the SWA in parallel. Analysis demonstrated that parallel design reduce the traceback runtime up to 50% from the original traceback system.**

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

I. INTRODUCTION

The collection of DNA data is growing every year due to the increase of population size. Moreover, it also has increased the number of probability in DNA sequence alignment. [1-2] has come out with two solutions on the runtime and the memory space based on the Smith-Waterman algorithm (SWA), due to this problem. Even though, this algorithm proves to work, the increasing size of population degrades the runtime and the traceback speed as highlighted in [3-6].The objective of this paper is to optimize the best trace back scanning performance based on the basic design architecture. This design will help to reduce the runtime and increases the speed of the trace back scanning performance.

DNA sequence alignment algorithm has been investigates by several researchers for over than 40 years. In 2009, the optimal alignment between two DNA sequences based on FPGA web server has been discovered by [5]. This optimal alignment compares using two different lengths of DNA sequences. This method helps to speed up the system up to 1.55 times using parallelization algorithm. While [3], finds the optimal sequences alignment and enhanced the traceback phase using hardware accelerator with 6000 times faster as compared to the implementation on Leon3 processor only.

This paper organizes as follows: Section I: Introduction, Section II: Smith-Waterman Algorithm, Section III: Structural Modelling, Section IV: Design Constructions, Section V: Result and Discussion, and Section VI: Conclusion.

II. SMITH-WATERMAN ALGORITHM

The Dynamic Programming Algorithm (DPA) has been used for determine the cell score for it dynamic matrix in DNA sequences alignment [5]. The two most highlighted algorithms using DPA known as Needleman-Wunsch Algorithm (NWA) and Smith-Waterman Algorithm (SWA). The NWA can break the DNA sequences alignment by picking only the best DNA sequences alignment in order to find the optimal alignment. In the other hand, the NWA finds the optimal path by searching through the original DNA sequences alignment [6]. But, the SWA provide the most sensitive optimal path. Furthermore, the traceback has reducing it advantage with large time consumed. Therefore, this paper enhances SWA to improve the overall system performance. The SWA can be explained based Figure 1:

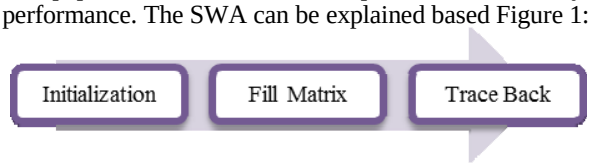

Figure 1. Smith-Waterman Algorithm flow process

A. Initialization

The Initialization module optimizes the data for the DNA sequences alignment process. In order to save memory space, data minimization technique has been used as proposed before in [1].This method reduces the number of bit per DNA character. The DNA characters assign as in TABLE I, where A will reduce into “00” while C, G and T as “01”, “10” and “11”.

TABLE I. REDUCTION OF DATA ASSIGNMENT FOR DNA SEQUENCE CHARACTERS

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

B. *Fill Matrix*

The fill matrix module calculates the score and saves the score in matrix cells. For an example, the SWA consist of two sequences A_x and B_y as shown in Figure 2.

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

The matrix calculation by aligning Search Sequence character A_x and Target Sequence character B_y shows in Table II. While, the score calculated by summing NW value with match value as illustrated in Table III. If A_x and B_y character are mismatch, the mismatch value will be used as a score and plus with the highest value from one of these three NW, N or W.

TABLE II. SMITH-WATERMAN ALGORITHM OUTPUT MATRIX TABLE

		Search Sequence								
	Ø	A	T	C	T	C	G	T	A	T
Ø	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

Target Sequence

TABLE III. SCORE OUTPUT FOR BASIC TABLE CALCULATION

Target Sequence	Search Sequence	
	NW	N
	W	Score

C. *Trace Back*

The traceback function is to find the optimal path using the score in the matrix table. The optimal path traceback scanning will trace from the optimal score until the lowest score. This example of the traceback scanning is shown Table IV with different color highlighted from maximum score until initial cell.

III. STRUCTURAL MODELLING

Structural modelling is very important in order to design a smooth traceback system with minimum design complexity. This system is design using Finite State Machines (FSM) module for enhance the performance of the design. FSM can be classified into two ways; Moore machine and Mealy machine. Moore machine and Mealy machine module architecture are outlined in Figure 3 and Figure 4.

A. *Moore Machine*

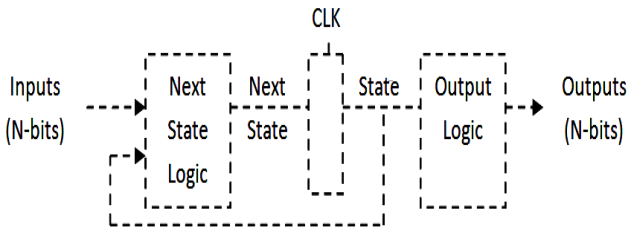

Figure 3. Moore Machine

The difference between Moore and Mealy machine is the way to achieve the result. Moore machine output relies on a current state.

B. *Mealy Machine*

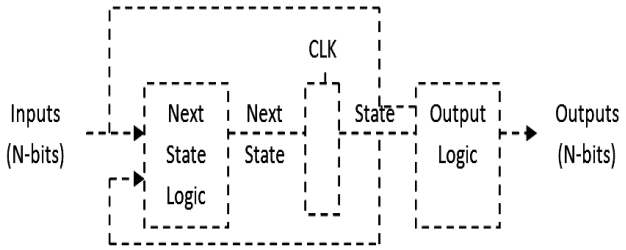

Figure 4. Mealy Machine

The output for Mealy machine depends on the current state and input. In this work, both architecture designs use the Mealy machine since the output depends on the input. The basic state diagram for the traceback is shown Figure 5.

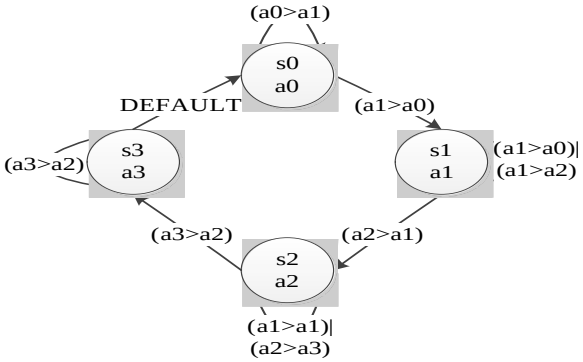

Figure 5. State Diagram

IV. DESIGN CONSTRUCTIONS

The method in this section covers two simple designs. These designs produce two different traceback. Three examples 4 x 4 matrix table use to examine both designs. The main design connecting in parallel as shown in Figure 6 below:

Figure 6. Parallel architecture for trace back

Based on the original SW, the Fill Matrix module calculates matrix scores for the traceback. Design 1 and Design 2 use the 4 x 4 Matrix size of table as in Table V below:

TABLE V. STATE AND SCORE VALUE FOR 4X4 MATRIX TABLE

	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

A. *Design 1*

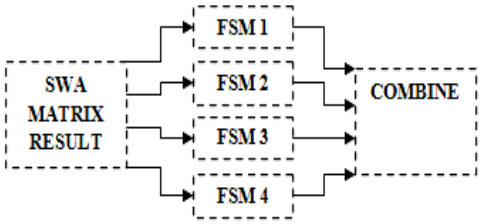

Figure 7. Design 1 Architecture

In Design 1, the FSM of 4 x 4 matrix table combined the output into 1 single line using FSM. The main architecture of Design 1is connected in parallel as shown Figure 7.

The traceback scanning process determines the highest score from each row. FSM1 checks the first input score for verification. If this input meets the right condition, it will automatically form as output, but if the first input did not meet the right condition, FSM1 checks the next stage until it finds right condition. Each FSM carries out the same procedure. The function of Combine Module is to organize the score from highest to lowest score. Figure 7 shows the design for FSM1, FSM2, FSM3 and FSM4. A FSM state diagram for Design 1 is shown in Figure 8.

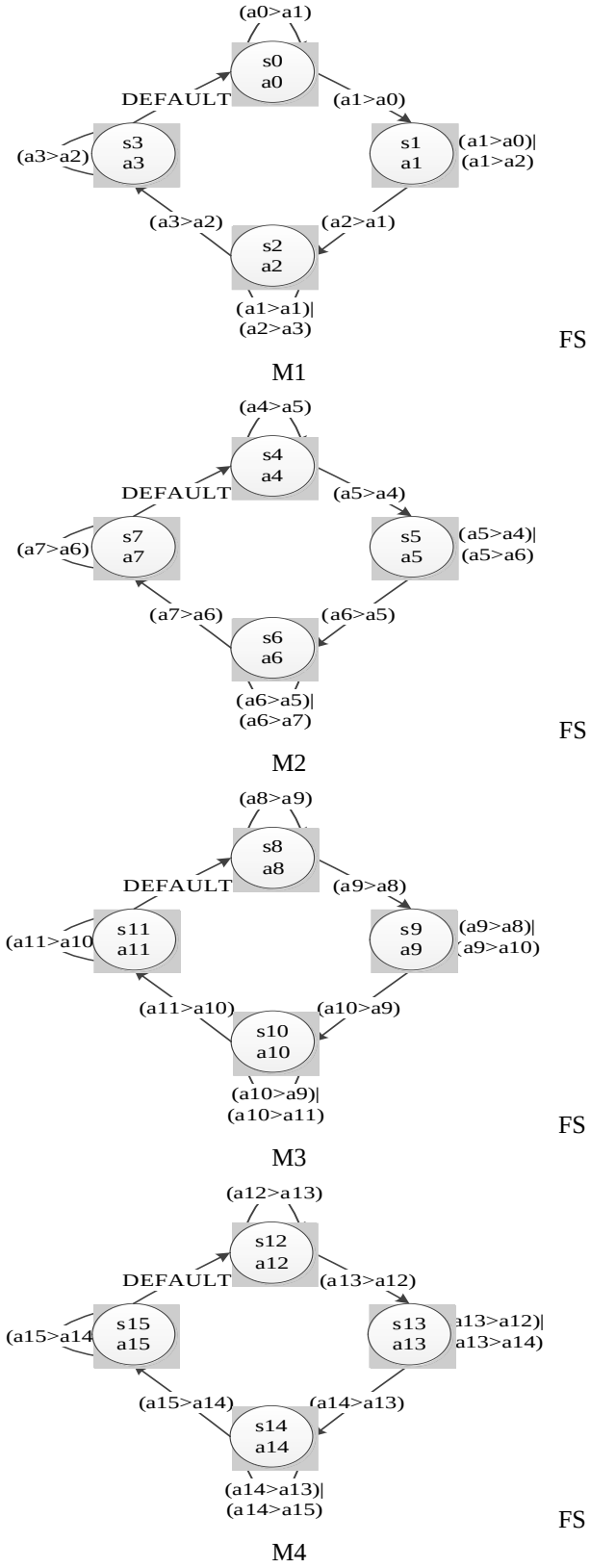

Figure 8. State diagram for FSM in Design 1

B. *Design 2*

The Design 2 connection is in cascading or parallel as shown in Figure 9. Design 2 only consist one Finite State Machine (FSM) module. The input signals are forming by Smith-Waterman score. The different between the Design 1 is on the Finite State Machine (FSM) design and score output.

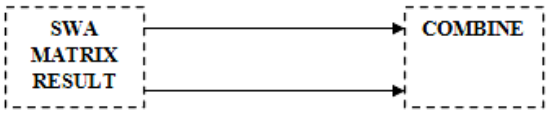

Figure 9. Design 2 Architecture

The Design 2 works from the first input score stage. In this stage, it checks whether it meets the right condition or not. It automatically forms as output if this stage meets the right condition. Each stage depends on the current stage condition in order to perform the output. The state diagram in for Design 2 is shown in Figure10.

V. Result and Discussion

Both designs simulate using the Altera Quartus II software. Figure 11 and 12 are showing the RTL schematic for the Design 1 and Design 2 respectively. Both of the figures compute input in the form of 4x4 matrixes.

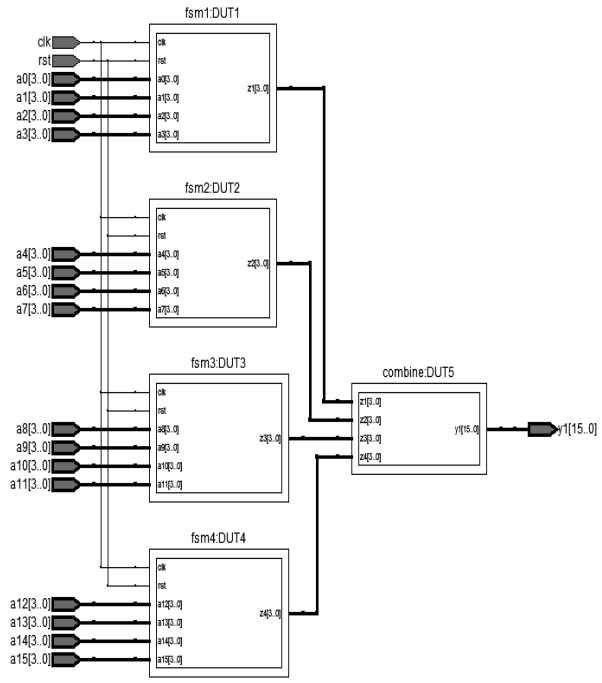

Figure 11. RTL for Design 1 (4 x 4 matrix size)

The input is randomly selects using the scoring matrix in the Table VI, VII and VIII. These tables constructed based on the Table II. The expected results for both designs tabulates out in Table IX and Table X respectively. This three different inputs table for Search Sequence and Target Sequence DNA forms the output as shown in Figure 13 (for the Design 1) and Figure 14 (for the Design 2).

TABLE VI. 4 X 4 Matrix Table Score 1

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

TABLE VII. 4 X 4 Matrix Table Score 2

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

TABLE VIII. 4 X 4 Matrix Table Score 3

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

The results in the Table IX show the optimal path traceback scores from the highest to lowest scores. The Table X shows the Design 1 expected result for example in Table VI, VII and VIII. While, Table XI shows the Design 2 expected result for the same example.

TABLE IX. Expected Output Result for Design 1

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

TABLE X. Expected Output Result for Design 2

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

All of the waveforms produced were equal to the expected result in the Table IX. The output tracing from the optimal score to lowest score was obtained.

Figure 13. Waveform for Design 1

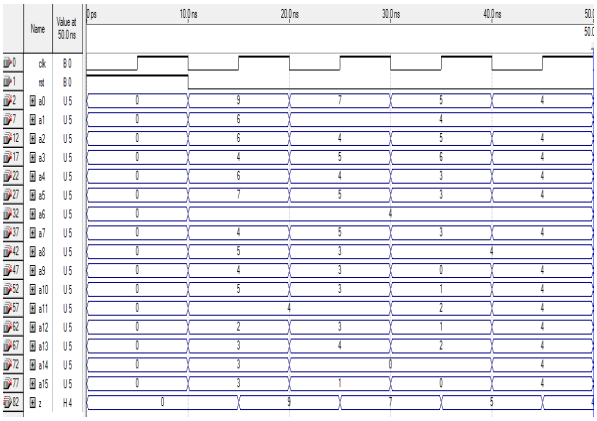

Figure 14. Waveform for Design 2

The total logic elements for Design 1 and Design 2 are shown in Figure 15 and Table XI. In Design 1, the total logic element increased drastically between 140 to 167 logic elements for each of the 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, the total logic element do not increased drastically for Design 2.

In addition, the different number of logic elements producing different performance for each of the design. The performance for Design 1 is slower compared to Design 2. Therefore, the faster output will be obtained by minimizing the number of logic elements.

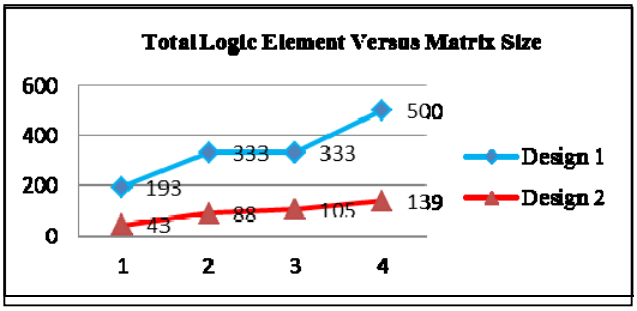

Figure 15. The effect of Logic element on matrix size for Design 1 and Design 2

TABLE XI. TOTAL LOGIC ELEMENT FOR EACH DESIGN IN DIFFERENT MATRIX SIZE

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 XII. INTRODUCTION FOR PARAMETERS PERFORMANCE

Performances parameters:	
Clock-to-Output Delay (t_{CO}):	Maximum time to obtain the output.

The list of the performance parameters that need to carry out in order to find the optimizing the runtime for both designs as defined in Table XII.

TABLE XIII. TIMING ANALYSIS FOR DESIGN 1

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

TABLE XIV. TIMING ANALYSIS FOR DESIGN 2

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

Design 1 and 2 carry out different value of timing analysis since the design was constructed using different architecture. Therefore, the different timing analysis recorded for both of the design as tabulated in Table XIII and Table XIV due to the different size of architecture design. On the other hand, the delay time is increases due to the same factor. The average clock-to-output delay for Design 1 is 14.989ns. While, the average value for Design 2 is slightly lower with 7.374ns. It gives reducing for runtime output up to 50%.

VI. CONCLUSIONS

The optimal traceback performance improvement achieves in this study two different design; Design 1 and Design 2. Both of the design has been compared for analyze the performance at different performance variable. Based on the result, both of the design achieved the design objective but Design 2 gives a tremendous performance as compared to Design 1 50% reduction in runtime. Therefore it proved that, simple design architecture can reduce the traceback runtime.

ACKNOWLEDGMENT

This project partially supported by Ministry of Science, Technology and Innovations Malaysia and Universiti Teknologi MARA under Science Fund Research Grant (03-01-01-SF0473)

REFERENCES

[1] 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. 8(1):185. (2007).

[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", AMS2010: Asia Modelling Symposium 2010 - 4th International Conference on Mathematical Modelling and Computer Simulation, pp. 231

[3] Nuno Sebastião, Tiago Dias, Nuno Roma and Paulo Flores. "Integrated Accelerator Architecture for DNA Sequences Alignment with Enhanced Traceback Phase", In International Conference on High Performance Computing & Simulation (HPCS 2010), pages 16-23, June 2010.

[4] Scott Lloyd and Quinn O.Snell. "Hardware Accelerated Sequence Alignment with Trace back", International Journal of Reconfigurable Computing Volume 2009 (2009), Article ID 762362, 10 pages doi:10.1155/2009/762362.

[5] Ying Liu, Khaled Benkid, Abd Samad Benkrid and Server Kasap "An FPGA Web Server for High Performance Biological Sequence Alignment", ISBN: 978-0-7695-3714-6 doi>10.1109/AHS.2009.59.

[6] Liaq Hasan, Zaid Al Ars, Zubair Nawaz and Koen Bertel "Hardware Implementation of Smith Waterman Algorithm Using Recursive Variable Expansion" Design and Test Workshop, 2008. IDT 2008. 3rd International.

[7] Nawaz, Mudassir Shabbir, Zaid Al-Ars and Koen Bertel "Acceleration of Biological Sequence Alignment using Recursive variable Expansion", Conference: Euromicro Symposium on Digital Systems Design - DSD, pp. 915-922, 2008.

[8] Xianyang Jiang, Xinchun Liu, Lin Xu, Peiheng Zhang, and Ninghui Sun "A Reconfigurable Accelerator for Smith-Waterman Algorithm", IEEE Transaction on Circuits and Systems II Express Briefs, VOL. 54, and NO. 12, DECEMBER 2007.

[9] Al Junid, S.A.M., Majid, Z.A. & Halim, A.K. 2008, "Development of DNA sequencing accelerator based on Smith Waterman algorithm with heuristic divide and conquer technique for FPGA implementation", Proceedings of the International Conference on Computer and Communication Engineering 2008, ICCCE08: Global Links for Human Development, pp. 994.

TABLE IV. TRACE BACK SCANNING TABLE

		Search 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

Target Sequence

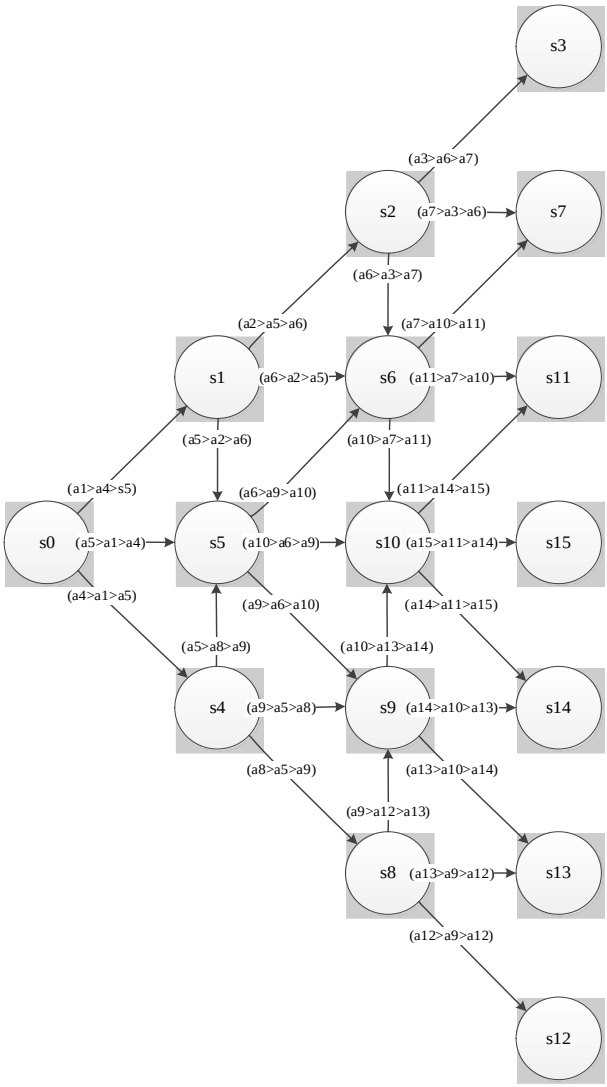

Figure 10. State diagram for FSM in Design 2

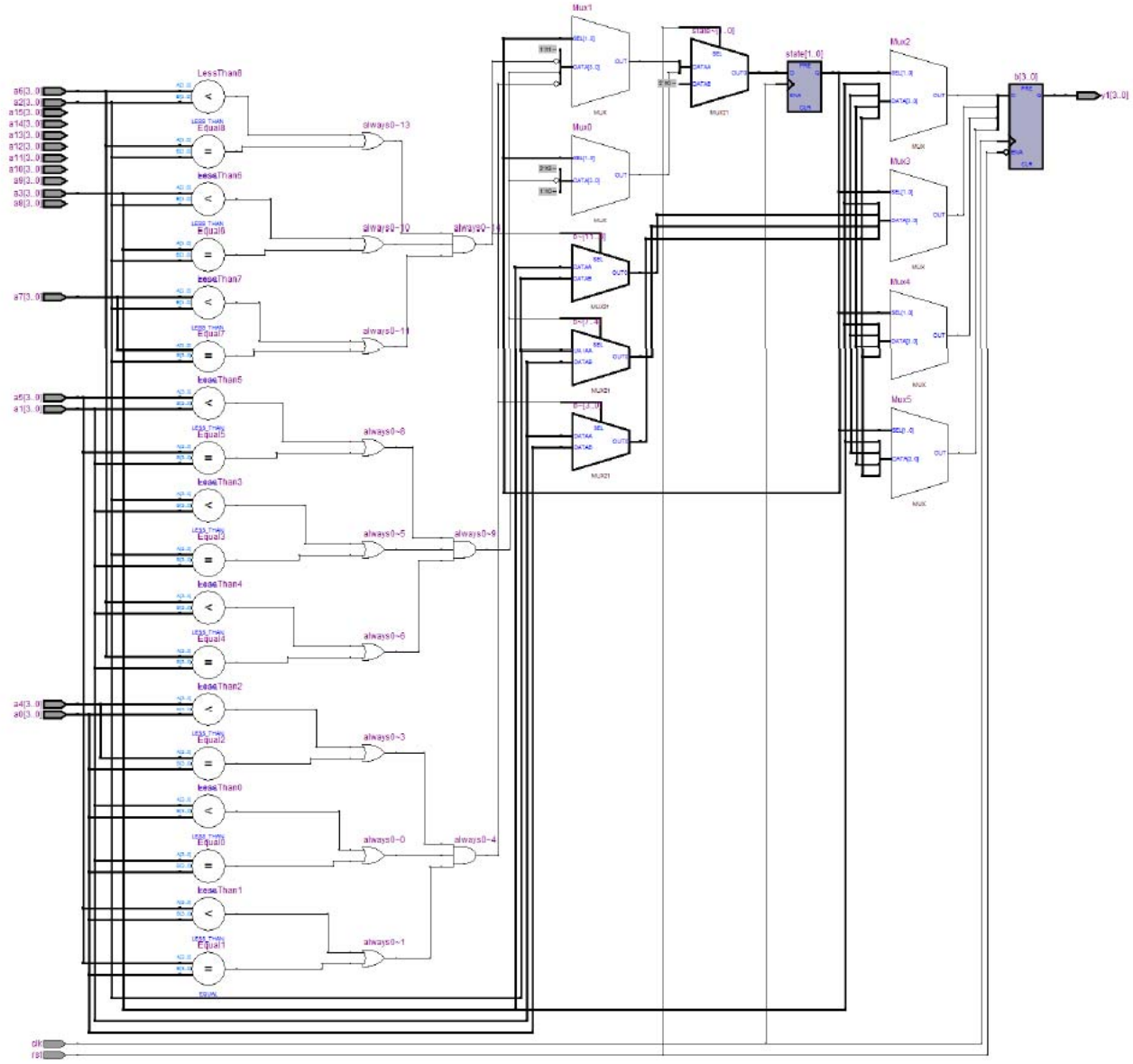

Figure 12. RTL for Design 2 (4 x4 matrix size)