DESIGN AND ANALYSIS OF HIGH PERFORMANCE MATRIX FILLING FOR DNA SEQUENCE ALIGNMENT ACCELERATOR USING ASIC DESIGN FLOW

Nurzaima Bt Mahmod
Department of Electronic
Faculty of Electrical Engineering
Universiti Teknologi MARA
40450 Shah Alam
Selangor,Malaysia
Email: nurzaima_mahmod@yahoo.com.my

Abstract - **This paper presents the design and analysis high performance matrix filling for DNA sequence alignment accelerator using ASIC design flow. The objective of this paper is to design and analysis matrix module of DNA sequence alignment accelerator using clock cycle to get high performance. The scope of this paper is to optimize the DNA sequences alignment on the matrix filling module by implementing a parallel method of the Smith-Waterman algorithm. This method provides magnificent speed up over than traditional sequential implementation methods while it sensitivity detection is still remained. To optimize the performance of the algorithm by exploiting parallelism in the design several techniques have been developed. In the advanced engineering technology, the massive parallelism can be implemented by using the Field Programmable Logic Array (FPGA) techniques. The design was developed in Verilog HDL coding and synthesis by using LINUX tools. From the LINUX tools, the optimum combination of parameters is manipulated to produce the most energy efficient IC. The design produces an ASIC that can work at 5ns until 10ns clock period and range of ICC time between 0.63ns until 1.67ns. The area of this design is 10304.358mm^2.**

Keywords- Smith-Waterman algorithm, DNA Sequencing, FPGA, ASIC.

I. INTRODUCTION

Biology is in the middle of a major paradigm shift driven by computing technology. A new hybrid field (partly molecular biology and partly computer science) began to emerge was called computational molecular biology [1]. In most common terms sequence alignment may be defined as an arrangement of two or more DNA sequences to highlight the regions of their similarity. This in turn indicates the genetic relatedness between the organisms.

The problem of sequence similarity is recurrent in the fields of Genetics and Bioinformatics. When a new gene is discovered its role and function may be inferred by its similarity to known sequences. In genetics the focus is usually on sequences of nucleic acids drawn from the set of four possibilities Adenine ('A'), Guanine ('G'), Cytosine ('C') and Thymine ('T') for DNA [2]. A typical problem will search for the best match of a query string inside a much larger database string. Methods for solving this problem with various additional boundary conditions are well known and drawn from the class of dynamic programming methods. Smith-Waterman is a dynamic programming method which finds the best local alignment, typically between a shorter query strings inside a longer database string.

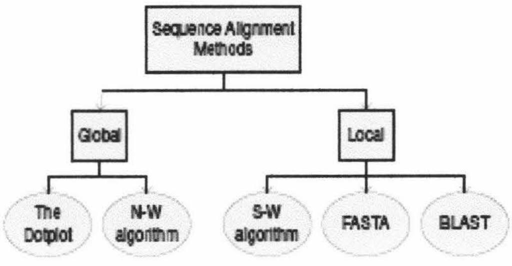


Figure 1: Various methods for sequence alignment.

Traditionally, the methods of pair wise sequence alignment are classified as either global or local. Global methods attempt to match as many characters as possible, from end to end, whereas local methods aim to identify short stretches of similarity between two sequences [3]. Apart from the S-W algorithm, there are other local search methods such as FASTA (Fast Alignment Search Tools - All) and BLAST (Basic Local Alignment Search Tool). Based on heuristics, they are faster, although much less sensitive than the S-W algorithm. The Smith-Waterman algorithm is a complex and sensitive algorithm for the DNA sequencing [4]. High level of accuracy and sensitivity compared to other algorithms make the algorithm applicable until now. However, the implementation of the algorithm has become more challenging as the memory requirements and speed, due to the high sensitivity of the large sequences long.

One of the biological sequencing solutions uses software analysis approach on general purpose computer [5].The software driven biological sequencing analysis based on Field Programmable Logic Array (FPGA) and LINUX implementation is found to be more suitable and offers a lot rooms for optimization since it suitable for high performance applications, low development and production cost and fast time to develop.

In this paper, the conventional Smith-Waterman algorithm is introduced in Section II. Furthermore, the methodology is explained in Section III. The result of the experiment is discussed in Section IV.

II. SMITH-WATERMAN ALGORITHM

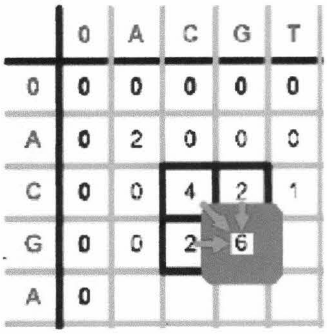
The algorithm was first proposed by Temple F. Smith and Michael S. Waterman in 1981. Smith-Waterman algorithm incorporates the evolutionary insertions and deletions techniques with the scoring function. The Smith-Waterman algorithm, based on the dynamic programming technique, was used to compute the optimal local alignment of two sequences. The procedure consists of three steps:

1) Initialization step
2) Fill in the dynamic programming matrix.
3) Trace back the path that leads to the maximal score to find the optimal local alignment.

For this paper, the design is focused on the matrix filling module of the Smith-Waterman algorithm. A matrix fill step is carried out using Equation 1, which fills out all entries in the matrix.

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) + gap\_penalty \\ F(i,j-1) + gap\_penalty \\ 0 \end{cases} \quad (1)$$

Equation 1



	0	A	C	G	T
0	0	0	0	0	0
A	0	2	0	0	0
C	0	0	4	2	1
G	0	0	2	6	
A	0				

Figure 2: Matrix filling using Smith-Waterman algorithm.

Figure 2 shows a sample "ACGT...C" (top) and a target "ACGAAC...G" (side). The first row and column of the Smith Waterman table are initialized to zero. Scores are then calculated starting in the upper left corner and moving outward. It illustrates how a score of '6' is calculated from its adjacent neighbor scores above and to the left, as well as from the fitness of the match between the 'G' subject character and the 'G' target character found at the head of its row and column. More important than cell calculation mechanics is how the algorithm is broken into smaller sub problems and solved in parallel [6-8].

III. Methodology

To design a matrix filling module for Smith-Waterman's algorithm with reasonable speed, this design have to be able to process data two input of 8bit numbers which represent the sample and target of 4 base pair DNA sequence and produce 16 output representing the score of the alignment according the Smith-Waterman algorithm.

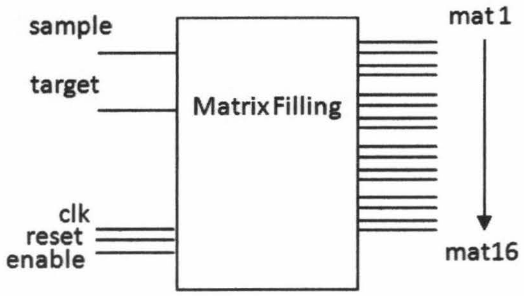


Figure 3: Block diagram of matrix filling DNA sequences alignment.

In this section, the technique or step for this project was introduced. There are 2 difference methods in implementing the design which are FPGA flow and ASIC flow.

A. FPGA Flow

The algorithm was designed using Verilog language and targeted to FPGA (Field-Programmable Gate Array) Spartan 3E XC3S100E board. The design is synthesized and simulated in search of the optimal speed Using ISE11.1 ISim to verify the outputs of the design and the search for minimal clock cycle is also done in this stage. For this design has only one main module and executed in parallel, which can produce all the answer for 16 matrixes cells in 1 clock cycles. The coding style is kept at a minimum to insure that the RTL schematic circuit is also to a minimum because larger schematic circuit consumes greater amount of time to operate. Propose of architectural design is a flatten design, where all the verilog coding is executed in parallel order. From this design, it will optimize/reduce the number of clock cycle.

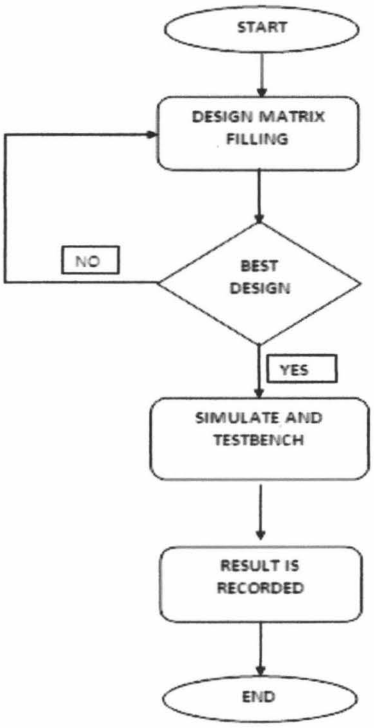


Figure 4: Flow chart for design matrix filling by using XILINX ISE

B. ASIC Flow

Then the design is synthesized by using Linux tools. There are several steps in ASIC flow to design a matrix filling of DNA sequences alignment. At every step the idea conceived keeps changing forms. Then, Synopsys Verilog Compiler Simulator (VCS) is used to simulate and debug the design. By using DVE, the simulation is represented visually where the waveform is generated. Then it is synthesized by using Design Compiler (DC tool), this is where the design libraries and design constraint are applied and the netlist is generated. Reports are also acquired during the synthesis process; these reports consist of timing reports, and design area report. Timing analysis is developed in order to get the timing reports. This is where we can see how fast the chip is going to interact with other chips, how fast the signal go from the input to the output and to check if there is any timing violations. Then Integrated Circuit Compiler (ICC) tool is used to generate the layout (VLSI) of the chip. Using the ICC tools, the floor planning, placement and the routing is done. Finally the GDSII file for the design is generated through the ICC tools.

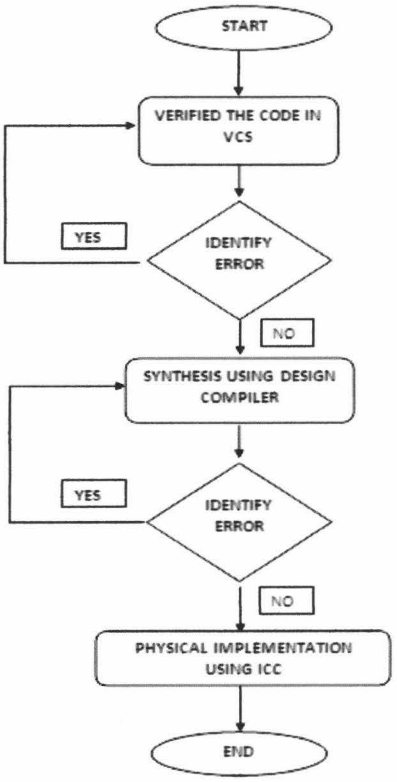


Figure 5: Flow chart for design matrix filling by using LINUX tools

IV. Result and Discussion

A. Architectural Design

The character been assigned in 2bit word in order to optimize the speed of alignment. All of the initial characters from ASCII been converted into 2 bits size to make it smaller and faster during alignment process [6].The schematic diagram of the design is generated by using the RTL schematic function in ISE11.1. From the RTL schematic diagram, it will shows the difference between 2x2 matrixes, 3x3 matrixes and 4x4 matrixes. After the design is synthesized, we can obtain the value of logic gate used and flip-flop depends on the number of matrixes cell.

TABLE1: Effect of Slice and Flip-Flop Depending on the Number of Matrixes Cell

Matrix	Matrixes Cell	Number of Logic	Number of Flip-flop
2x2	4	29	8
3x3	9	297	21
4x4	16	832	38

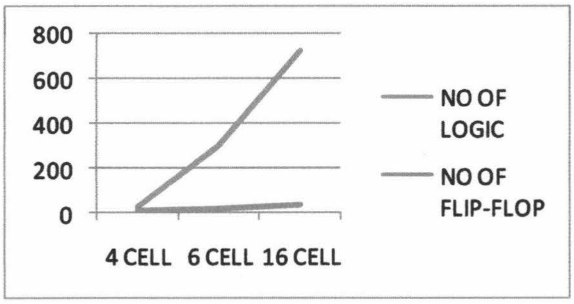


Figure 6: Graph of logic gate and flip-flop depending on number of matrixes cell.

Graph 6 shows the number of logic gate and flip-flop versus number of matrix cell. From this graph, it shows that number of logic gate and flip-flop will increase when number of matrix cell is increase. On behalf on that, the estimation of logic gates and number of flip-flops for 8x8 matrix design will be around 1307 and 65.

B. Design Verification

The output waveform gained from the Isim and DVE tools (Synopsis) can be compared with the expected results from Smith-Waterman algorithm. The expected result of S-W is obtained by theoretical calculation.

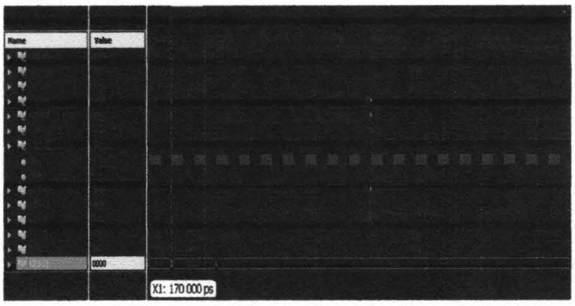


Figure 7: Simulation waveform (matrix 4x4) from ISE11.1 ISim.

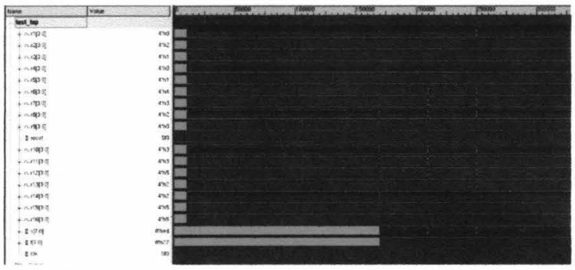


Figure 8: Simulation waveform (matrix 4x4) from LINUX tools.

TABLE 2: THE DISPLAY OUTPUT OF SMITH-WATERMAN ALGORITHM.

SAMPLE	TARGET	r1	r4	r5	r6	r9
00011011	00110100	2	0	1	1	0
00100011	00110100	2	1	1	1	0
00111001	00111000	2	0	1	4	0
00111001	00101001	2	0	1	1	0
00001001	00100100	2	0	1	1	0
00100111	00110010	2	0	1	1	0

From the figure 7 and 8, it shows simulation result is equal to theoretical values of Smith-Waterman algorithm. The simulation is completed in one clock cycle where it is proved that the design in achieved high performance.

C. Timing Analysis For High Performance

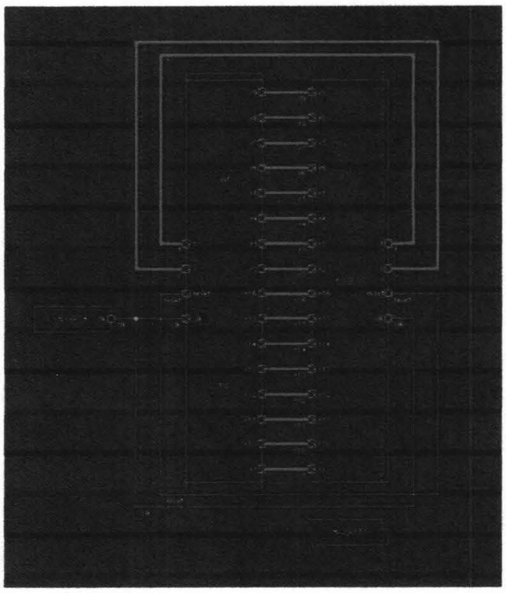


Figure 9: Schematic design of DNA sequences alignment.

Figure 9 shows the schematic design simulated from verification-RTL. From the figure, there are CLK, RESET, sample (S) and target (T) pin connections. The output of comparison between sample and target is displayed at pin r1 until r16.

For high performance by varying the clock cycle period of the design we can obtain the various value of timing and area and this technique is called frequency scaling technique.

TABLE 3: THE NUMBER OF CLOCK VERSUS TIMING CONSTRAINT FOR HIGH PERFORMANCE.

CLOCK (ns)	AREA ($mm^{2)}$	DC MAX (ns)	ICC MAX (ns)	ICC MIN (ns)	PT MAX (ns)
2.5	10251.964	0.00	-0.51	0.17	-0.52
5	10304.358	0.64	0.63	0.24	1.10
10	10351.382	3.85	1.97	0.08	5.04
15	10531.382	9.02	7.06	0.09	7.29
20	10598.118	13.81	11.83	0.08	12.53
25	10610.231	20.35	19.86	0.07	14.75
30	10615.415	23.81	21.97	0.08	17.26
35	10623.247	28.77	27.28	0.09	20.71

Table 3 shows when varying the clock cycle, it will affect the area, DC max, ICC max, ICC min and PT max. The lock cycle is proportionally to area. This happened because when there is involved a large space, it will required more time to complete the compiling process of the design. The timing analysis should be not violated which means it have to be in positive values. If there is consist negative value; (clock 2.5ns and ICC max -0.51ns), it shows the clock cycle over minimum limits. In order to have a good design, it is rather to choose the appropriate value for clock cycle and at the same time smaller design area.

D. Design Compiler Analysis

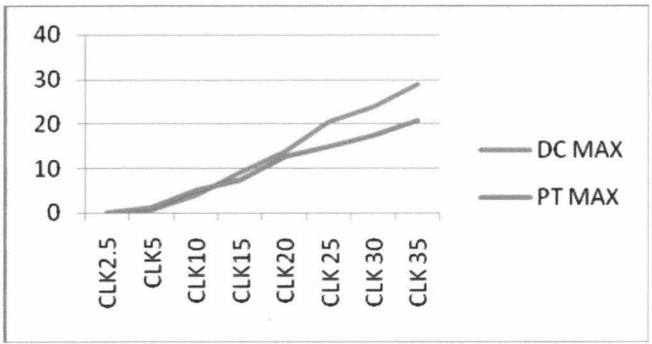


Figure 10: Graph of number of clock versus DC MAX and PT MAX.

Figure 10 indicates the timing for DC and PT is proportionally to clock period. Design compiler is overall surface timing analysis. Maximum design compiler will execute the long path of time taken depends on the clock cycle. Increasingly in clock cycle will make the path timing increased. Same goes to maximum prime time but PT analyze the timing delays in the design and flags violations that must be corrected.

E. Integrated Circuit Compiler Analysis

For the real time analysis it can be view in IC Compiler (ICC). IC Compiler timing analysis is the overall time taken to implement the design.

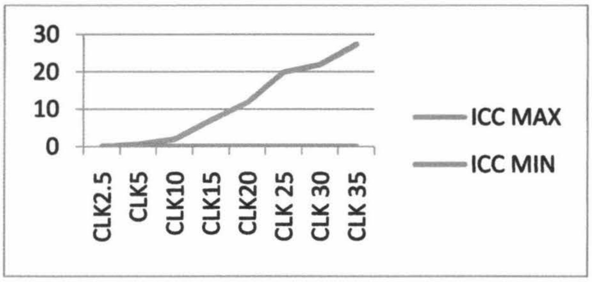


Figure 11: Graph number of clock versus ICC MAX and ICC MIN.

From figure 11, it shows that clock comparatively to ICC max and ICC min. ICC max will increased due to additional of clock cycle. For ICC min it shows that there are small significant changes when clock cycle changes. This is because, ICC timing analysis is indicates the real timing for a design.

DC and ICC timing analysis are almost similar but ICC indicates the real time analysis compared to DC. This is due to IC compiler is done during physical implementation process. On top of that, the physical implementation is the last step before generate GDSII file. So, it is the most important part to be viewed.

V. Conclusion

In conclusion, the new approach in this paper introduced to improve the design to be high performance by using Smith Waterman algorithm. The simulation result shown that the design is fully implemented in 1 clock cycle compared to other design. This technique proved to give the best timing analysis without affecting the sensitivity. It is referred to simulation result in Xilinx ISE. For ASIC design it shows that the overall timing is optimize to be high performance. Thus, the objective of this paper is successfully achieved due to design simulated in 1 clock cycle and obtained the best timing analysis.

Acknowledgement

I would like to express my sincere thanks and appreciation to Mdm. Norhazlin Khairudin, my project supervisor, my co-supervisor Mr. Abdul Karimi Halim and other contributors for their guidance, support and helpful advices to develop the Design and Analysis of High Performance Matrix Filling for DNA Sequence Alignment Accelerator Using ASIC Design Flow.

My family provides the strength and motivation and has been so tolerance and being supportive. Very special thanks for their love, support and encouragement they had given to me.

Last but not least, I would like to express my sincerest gratitude to my friends for their help, constant support and encouragement throughout the entire duration of this project.

REFERENCES

[1] http://www.roseindia.net/bioinformatics/

[2] Smith, Temple F.; and Waterman, Michael S. (1981). "Identification of Common Molecular Subsequences". *Journal of Molecular Biology* 147: 195–197. doi:10.1016/0022-2836(81)90087-5

[3]http://www.dbmi.columbia.edu/bioinformatics/.

[4]T.F. Smith, M.S. Waterman, "Identification of common molecular subsequences" J. of Molecular Biology, 147(1): 195-197, 1981

[5]F. Zhang, X-Z. Qiao, Z-Y. Liu, "A parallel smith-waterman algorithm based on divide and conquer," ICA3PP '02, 2002

[6] Sternberg, M. (Ed.), Protein Structure Prediction: A Practical Approach, Chapter by Barton: Protein Sequence Alignment and Database Scanning, *Oxford University Press* ISBN 0199634963.

[7] Margerm, Steve, and Maltby, Jim; Accelerating the Smith- Waterman Algorithm on the Cray XD1, *Cray WP-0060406* 2006.

[8] Storaasli, Olaf, Yu, Weikuan, Strenski, Dave, & Malby,Jim; Perfomance Evaluation of FPGA-Based Biological Applications, *Cray Users Group Proceedings*, Seattle WA, May 2007.

[9] Syed Abdul Mutalib Al Junid, Muhammad AdibHaron, Zulkifli Abd Majid, Abdul Karimi Halim, Fairul Nazmie, Hadzli Hashim, "Development of novel data compression technique for accelerate DNA sequence alignment based on Smith-Waterman algorithm".

APPENDIX

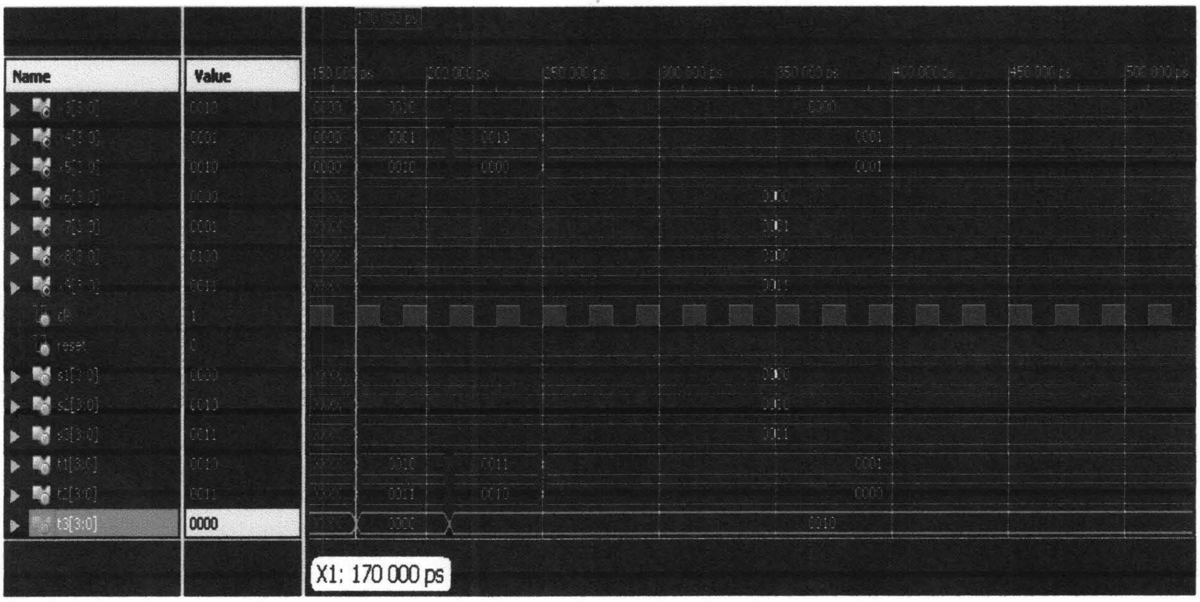


Figure 7: Simulation waveform (matrix 4x4) from ISE11.1 ISim.

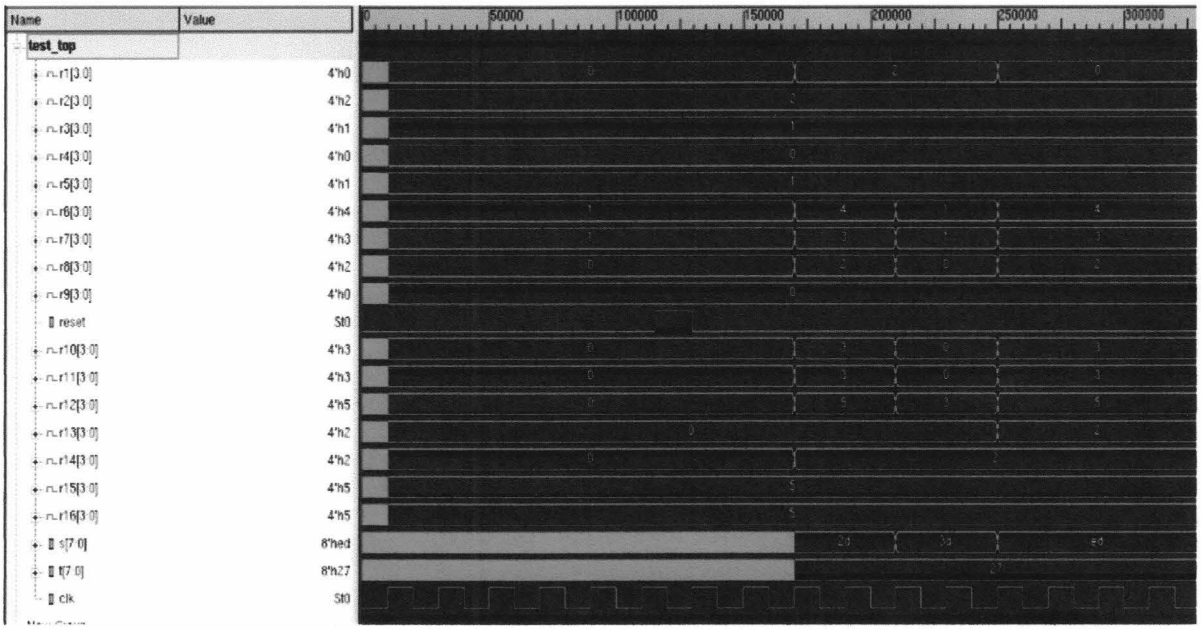


Figure 8: Simulation waveform (matrix 4x4) from LINUX tools.

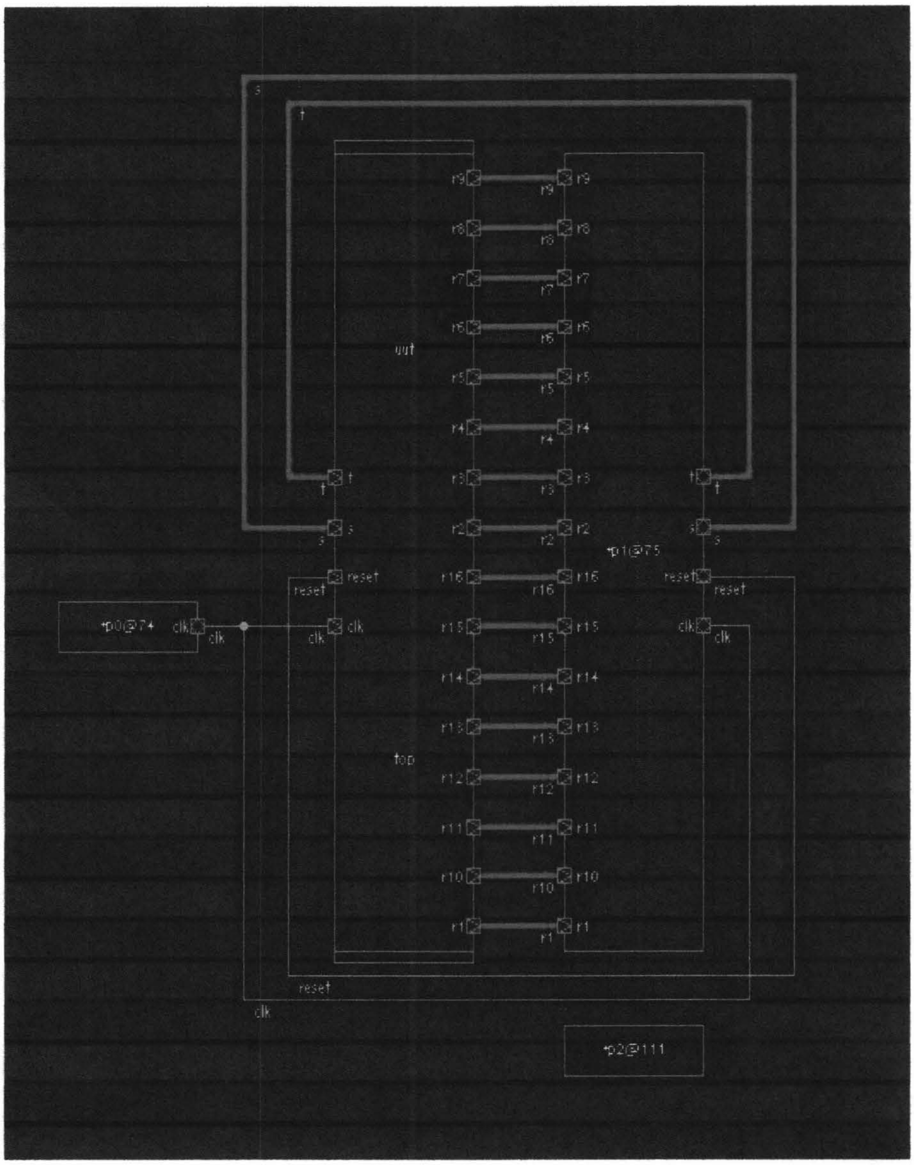


Figure 9: Schematic design of DNA sequences alignment.

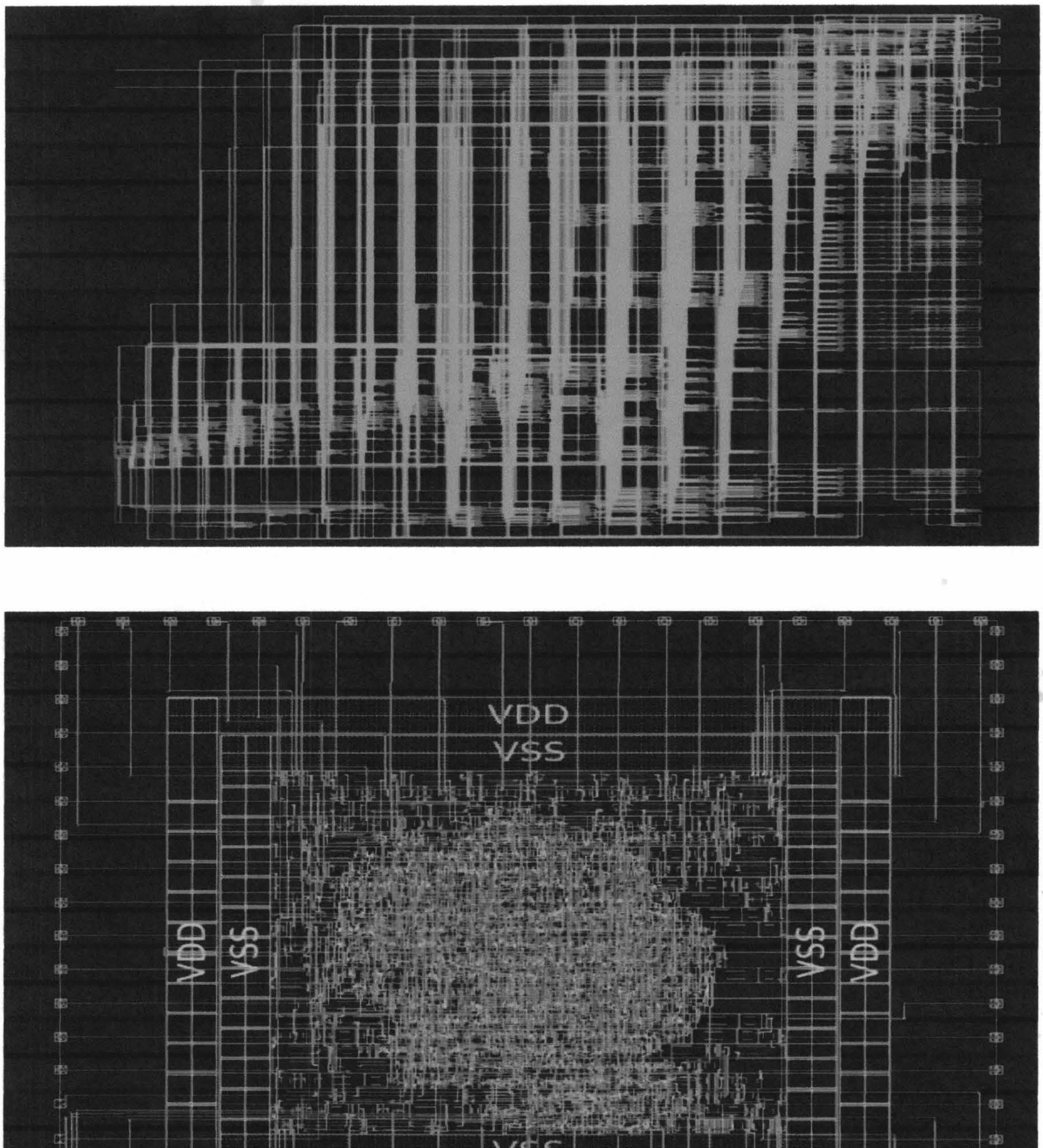


Figure 12: Physical layout of the design generated by Integrated Circuit Compiler (ICC).