Optimization of Initialization Module of DNA Sequence Alignment Accelerator

Zahiruddin Maksud
Faculty of Electrical Engineering
Universiti Teknologi MARA Malaysia
40450 Shah Alam
Selangor, Malaysia.
email: zahirmd@gmail.com

Abstract— **This paper presents the development and implementation of the initialization module of the DNA sequence alignment accelerator. The scopes of this paper focus on the memory reduction and speed optimization on the initialization module using parallelism and divide and conquer methodologies. The development of the optimization using data compression technique is presented in this paper. In this paper, the development target is Xilinx Spartan 3E FPGA using 50 MHz oscillator clock. The code is written in Verilog HDL using Xilinx ISE 10.1 and simulation testbench developed using Xilinx ISE 10.1 simulation tool. Theoritical analysis and simulation results have been compared in this paper.**

Keywords— **Optimized DNA Sequence Alignment, Data Reduction, Data Compression**

I. INTRODUCTION

The demand of fast and less memory usage of DNA sequence alignment accelerator tools for DNA sequences alignment investigation is increasing from time to time. The genomic database has increase double in every 16 months. The statements supported and proved by the statement in [1]. The development of initialization module of DNA sequence alignment accelerator can optimize the speed and reduce the memory consumption using parallel architectures and data compression technique. This will fulfill the market needs for device with optimized speed and less memory.

II. INITIALIZATION MODULE

Initialization module objective is to prepare the data that enter the next module of the DNA sequence alignment accelerator such as matrix filling module. The problem of the existing methods preparing the data entering the DNA Sequence Alignment Accelerator is reported in [2] and [3]. The existing system problems is it's require a large memory to store the data before the alignment process can be performed.

The memory space requirement to store the data is proportional with the length of DNA sequence.

The memory space requirement for search sequences, M_S is proportional to the length of the search sequences, S_L

$$M_S \propto S_L \qquad (1)$$

While the memory space requirement for search sequences, M_T is proportional to the length of the target sequences, T_L.

$$M_T \propto T_L \qquad (2)$$

From the equation of (1) and (2) ,it can be concluded that the requirements of the memory to store the DNA sequences is double to the length of the DNA sequences since for DNA sequences alignment it required two set of DNA sequences data which is target and search sequences data.

The maximum DNA sequences reported today in [4] is three billion. Benchmarking this statement, the number of memory space requirement is increased nowadays. And it will increase day by day. The standard format of the DNA data is written in ASCII code format with eight bit data width. ASCII code representation for DNA data characters each as shown in Table 1, where each characters is been represent with it specific ASCII code. This leads to large memory space requirements problem. And will require more space.

TABLE 1
ASCII CODE REPRESENTATION FOR DNA DATA CHARACTERS

Name	Characters	ASCII	Data bit
Adenine	A	65	0110 0101
Cytosine	C	65	0110 0111
Guanine	G	71	0111 0001
Thymine	T	84	1000 0100

To overcome this problem in this initialization module, two techniques will be introduced to reduce the memory space requirements of the sequence alignment accelerator. These two techniques are named as Data Reduction and Data Compression technique. Both of these techniques will be implemented on the initialization module of the DNA Sequence Alignment Accelerator. Those modules are able to reduce the memory requirements and increase performance using parallel architectures.

A. DATA REDUCTION TECHNIQUE

The objective of this technique is to reduce the data width of 8-bit data format ASCII code into the 2-bit data format. As the data width reduce the probability of comparison, memory requirement and the size of FPGA architecture design is also can be reduced. In this reduction data technique, every characters of A, C, G and T will be represents in new 2-bit data format. Either the streams or the patterns are composed of four basic nucleotides A, T, C and G. The tabulation of the DNA data representation is shown is Table 2.

TABLE 2
NEW REPRESENTATIONS FOR DNA DATA CHARACTERS

Name	Characters	New Representations
Adenine	A	00
Cytosine	C	01
Guanine	*G*	*10*
Thymine	T	11

As in the table, the DNA characters are mapped with the two bit data representations. Where A represents by 00, while C, G and T is represents with 01, 10 and 11 respectively. Now the DNA data width has been reduces to only 2-bit data format. Thus the memory usage can be reduced from 8-bit data to 2-bit data. This reduces 75% of the memory usage using 2-bit data rather than using 8-bit data.

B. DATA COMPRESSION TECHNIQUE

This techniques objective is to compress the data from a group data into a set of data. For example, let's say that the data have sixteen DNA data in which search sequence,

S = {a g g c c t a c}

And target sequence,

T = {c c a g g t a g}

Both of the sequences have eight data or eight lengths of data. In data compression, this data will be compressed to become set of data with data width, P. For example, choose P = 4. The data will be compressed into the memory.
The memory will be divided into 4 memories and each memory will have 4 data inside them.

Using this Data Compression technique, the memory of 32-bit for example will be fully utilized. Instead of putting one DNA data into a single register, Data Compression put 8 DNA data into a single register. Thus this technique can optimize the speed of the DNA at later stages because the DNA data is stored in 32-bit memory can be recalled at anytime as 32-bit length of data.

The total number of data is for both of the sequence is sixteen DNA data but after the data compression technique the total data has been merges to only four data.

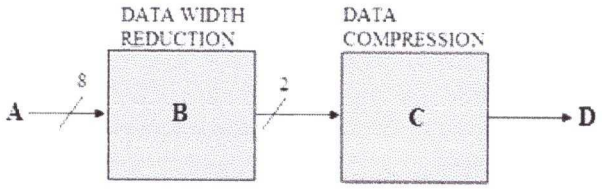

Fig. 1 Initialization module

The process of data compression is shown in Figure 1, starting with the Data Reduction Technique and end with Data Compression.

At point A, 16 based-pair of DNA sequences with 32 data in ASCII code each will enter the module.

For example,
S= { a c g t a c g t}
T= { a c g t a c g t}

At point B, Data Reduction module will reduce the data size from 8-bit to 2-bit.

At point C, Data Compression module will compress the data.

At point D, One based-pair of DNA data sequences set with 2 data compressed data.

S={*00 01 10 11 00 01 10 11*}
T={00 01 10 11 00 01 10 11}

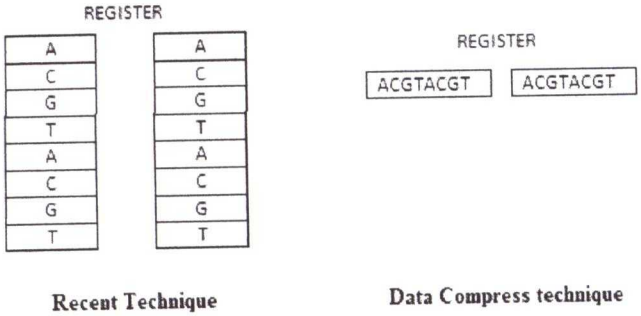

Fig. 2 Comparison between recent technique and Data Compress technique

Figure 2 shows that the comparison between the recent technique and Data Compress technique. It shows that Data Compress use simplify the data into only two register and reduce the memory requirements for data storage and space in the chip. The point is, instead of using 16 registers to store 16 DNA data, it is better to store 16 DNA data into 2 registers and save the memory space.

III. RESULTS AND DISCUSSION

The theoretical results of the Data Compression technique are analyzed according to the performance and the memory usage. The performance of the Data Compression technique is proportional with the data size, Pother effect of the data size P with the performance is shown in figure 3.

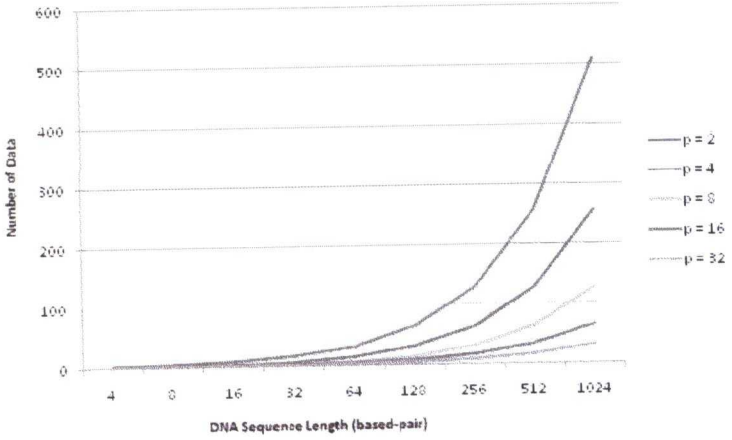

Fig. 3 Effect of Data Width in Data Compression technique for DNA sequence alignment.

The Figure 3 shows that then data size, P increasing, the more DNA Sequence Length can be stored. This should increase the performance of the DNA Sequence Alignment Accelerator.

From Figure 3, the data compression technique give superior reduction on memory requirement with 87.5% reduction compare to existing technique in [2-3],[4],[5] when data width, $P = 16$. The reduction show 97.5% when the data width, $P = 32$. These shows that the data compression technique can reduce the memory space requirement to analyze DNA sequences alignment sixteen times better than recent technique. This optimized Initialization Module can provide a very good set of data for the next stage of DNA Sequence Alignment Accelerator.

A. IMPLEMENTATION

Data Reduction and Data Compression techniques are written in Verilog HDL code using the Xilinx ISE 10.1. The code is implemented and burned on SPARTAN 3E FPGA starter kit board. Figure 4 shows the RTL schematic of the technique that extracted from the Verilog HDL code.

Both tables below list the necessary information and timing constraints for Data Reduction, Data Compression and Combination of both when targeted to the SPARTAN 3E board.

TABLE 3
LOGIC UTILIZATION AND TIMING CONSTRAINT FOR BOTH MODULES

	Data Reduction	Data Compression
Register	2	64
Slices	6	56
Slice Flip Flop	2	64
Minimum clock period	1.804ns	8.688ns
Maximum frequency	554.339MHz	115.102MHz
Minimum input arrival time before clock	5.097ns (Logic=3.488ns Route=1.609ns)	2.253ns (Logic=1.374ns Route=0.879ns)
Maximum output required time after clock:	4.063ns (Logic=3.683ns Route=0.380ns)	4.040ns (Logic=3.683ns Route=0.357ns)

TABLE 4
LOGIC UTILIZATION AND TIMING CONSTRAINT FOR COMBINATION OF BOTH MODULES

	Combinational
Register	132
Slices	120
Slice Flip Flop	132
Minimum clock period	8.746ns
Maximum frequency	114.341MHz
Minimum input arrival time before clock	4.621ns (Logic = 3.125ns Route = 1.496ns)
Maximum output required time after clock:	4.040ns (Logic = 3.683ns Route = 0.357ns)

B. SIMULATION

The Initialization module of the DNA sequence alignment accelerator simulation has been done with Xilinx ISE 10.1 simulation tool with the clock is set to 50MHz. This includes the Data Reduction and Data Compression module.
Figure 5 shows the simulation waveform of Data Reduction. While Figure 6 shows the simulation waveform of Data Compression.
Simulation of both module are using 5ns duty cycle clock and the input injected in 10ns delay.

IV. CONCLUSION

This paper presents the Data Reduction and Data Compression techniques that implemented in the Initialization Module of DNA Sequence Alignment Accelerator. DNA data sequence has been reduced in size and compressed in memory register using both of the techniques. When the DNA sequence is reduced and compressed, the number of clock process is also reduced, thus increasing the performance such as speed of the device. Implementation on the targeted SPARTAN 3E FPGA board confirms that the technique used in the Initialization module can be accelerated with the parallel architectures on the hardware. Other than that, the complexity of the code is reduced but the high performance and less memory usage is achieved.

ACKNOWLEDGEMENT

I wish to acknowledge to my project supervisor, Mdm. Norhazlin Khairudin and my co-supervisor Mr. Syed Abdul Mutalib Al-Junid and other contributors for their guidance, support and helpful advices to develop the Initialization Module of DNA Sequence Alignment Accelerator.

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] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, D.L. Wheeler, "GenBank, Nucleic Acids Res.", Jan 1;33(Database issue):D34-8, 2005.

[2] Syed Abdul Mutalib Al Junid, Zulkifli Abd Majid, Abdul Karimi Halim, "Development of DNA Sequensing Accelerator based on Smith-Waterman Algorithm with Heuristic Divide and Conquer Technique for FPGA Implementation" ICCCE 08, 2008.

[3] Syed Abdul Mutalib Al Junid, Zulkifli Abd Majid, Abdul Karimi Halim "High Performance DNA Sequences Alignment" ICED 08, 2008

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

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

[6] Daniel P. Lopresti. Rapid implementation of a genetic sequence comparator using field programmable logic arrays. Conference on Advanced research in VLSI pages 138-152, 1991.

[7] Steve Margerm Cray Inc. Reconfigurable computing in realworld applications. *FPGA and Structured ASIC Journal (www.fpgajournal.com)*, February 7, 2006.

[8] D. W. Mount.Bioinformatics Sequence and Genome Analysis. Chapter Alignment of Pairs of Sequences, Cold Spring Harbour Laboratoy, Press 2001.

[9] Euripides Sotiriades, Christos Kozanitis, Apostolos Dollas.
FPGA based Architecture for DNA Sequence Comparison and Database Search.

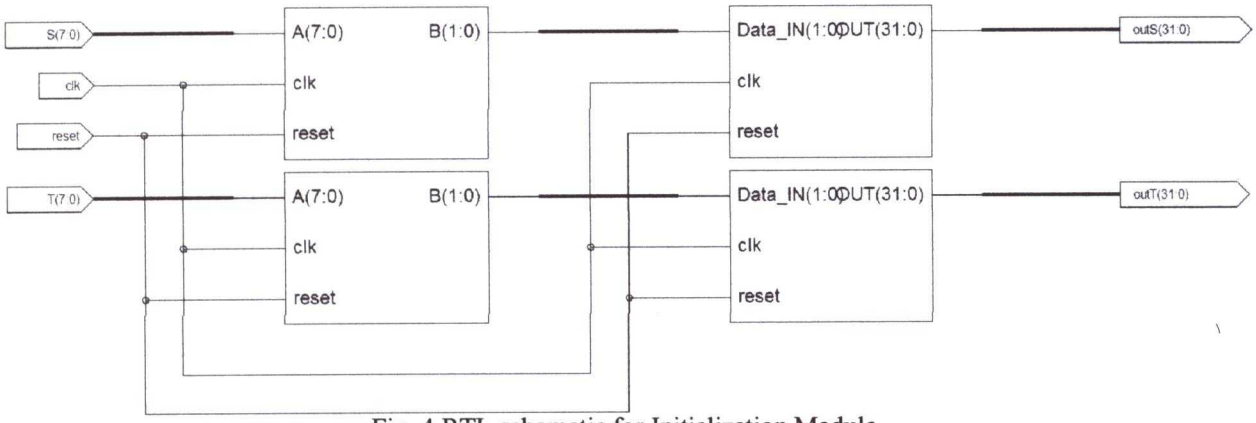

Fig. 4 RTL schematic for Initialization Module.

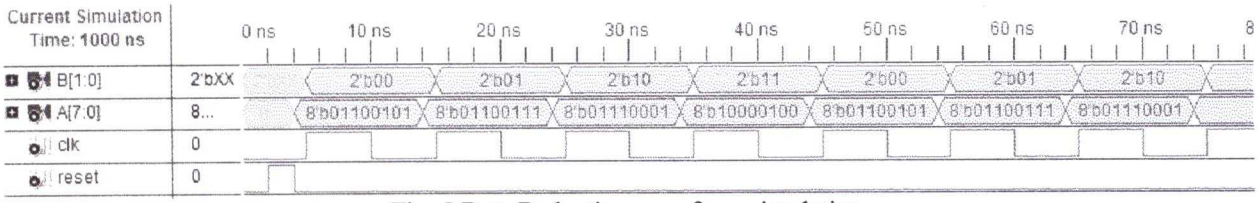

Fig. 5 Data Reduction waveform simulation

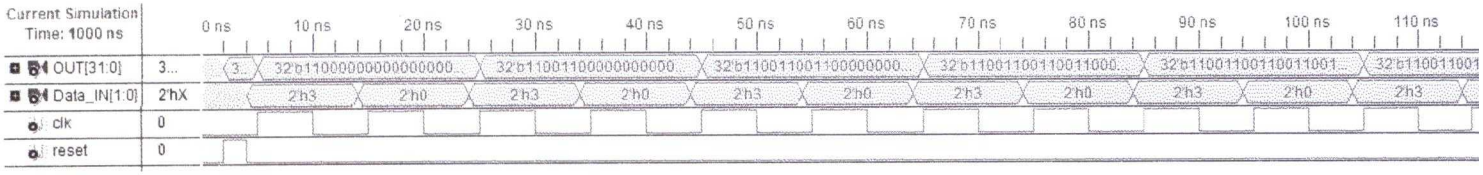

Fig. 6 Data Compress waveform simulation