Paper ID: E&TC05

BEHAVIORAL MODELING OF SMITH-WATERMAN ALGORITHM FOR DNA COMPARISONUSING FPGA

Swaroopa S. Kulkarni
PG student, Dr. D. Y. Patil college of Engineering and Technology,Kolhapur

Dr.T.B.Mohite-Patil
Principal, D. Y. PatilPratishthan's D. Y. Patil Engineering college, SalokheNagar, Kolhapur

Digamber B.Gote
PG student, Dr. D. Y. Patil college of Engineering and Technology, Kolhapur

K.M.Patil
Jnanasangam,V.T.U, Belgavi

Abstract: Smith Waterman algorithm uses divide and conquer approach. For finding the difference between two DNA strands. For implementation of this algorithm previously structural modeling was started and continued till the configuration of the basic cell the matrix structure. But it was found after creating its schematic symbol that the behavioral modeling would be more simple and convenient for this work. In this paper the implementation is discussed using behavioral modeling for Spartan-6 family of FPGA in order to implement the Smith –Waterman algorithm.

Keywords- Behavioral modeling, Smith-Waterman algorithm, FPGA, Mutation distance, DNA strand

I. INTRODUCTION

There are different algorithms for comparing DNA [1] strands. One of them is Smith-Waterman [2] algorithm finds differences in strands of different DNAs by adopting the divide and conquers approach. For implementing the algorithm on FPGA, options are available like structural modeling, behavioral modeling etc. Also two languages are available, VHDL and Verilog in Xilinx [3] software. For implementation of Smith –Waterman algorithm on FPGA behavioral modeling is adopted with selected as Verilog.

II. SYSTEM DESIGN

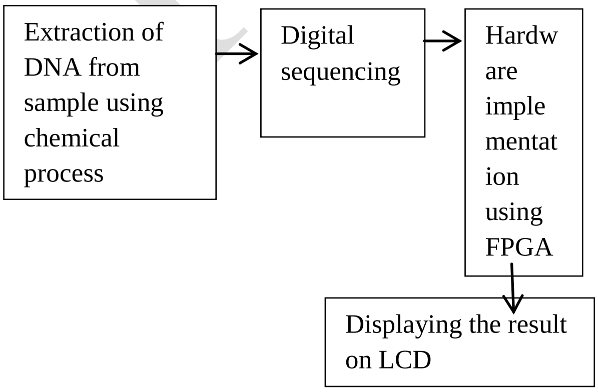
Extraction of DNA from sample using chemical process
Digital sequencing
Hardware implementation using FPGA
Displaying the result on LCD

Fig. 1 The system block diagram

The system shown above is divided into a number of stages of implementation. These stages are described as follows:

1. Extraction of DNA from sample- DNA is extracted by chemical processes such as mixing the grounded sample of material (for example, green peas) with liquid soap, then adding alcohol etc.
2. Digital sequencing - The DNA sequence of the extracted sample is then generated by the digital sequencer. Sequencer is available in the Shivaji University, Biotechnology department. Its usage is permitted theby the concerned authorities.
3. Hardware implementation using FPGA- Smith-Waterman algorithm is implemented on Xilinx platform by running the Verilog code onISE suite and further, the code is tested on Spartan -6[3] FPGA family. The result is displayed on LEDs giving the output that is, mutation distance between two strands.
4. Displaying the final result- The result of comparisons (whether the DNAs are matching or not) will be displayed on LCD display. For LCD interfacing microcontroller [5] is used.

The DNA sample is synthesized and the sequence is found with the help of digital sequencer. This target sequence is compared with the source DNA sequence. For hardware implementation of Smith –Waterman algorithm FPGA programmed. Finally we get the mutation distance between the two DNAs on LEDs available on the FPGA development board. Depending on this mutation distance DNAs matching result is displayed on LCD display which is interfaced to our system using a simple microcontroller.

III. METHODOLOGY

Two sequences represented as two strings are compared. This algorithm compares segments of small possible lengths (here we have considered only three pairs in this application) of these two strings and measures the mutation distance between each pair of segments. The Smith-Waterman algorithm is implemented as follows:

The mutation distance between two sequences represents the mutations needed togo from one sequence (called Source or just S) to the other (called Target or just T).Three different mutations are considered: insertion, deletion and substitution, each of these mutations have a distance assigned. Typically the mutation distance for the insertion and deletion operations is 1, and for substitution is two. The reason is that a substitution produces the same result as that in a combination of an insertion and a deletion. The entire sequence is compared with another by ‘divide and conquer’ approach. To compare the two strands of different DNA’s a matrix structure is used. Each cell in this matrix includes a combinational circuit. Basically each cell receives the distances a, b and c from the previous cells:

a: distance to go from S0-Si-1 to T0-Tj-1
b: distance to go from S0-Si to T0- Tj-1
c: distance to go from S0-Si-1 to T0- Tj

Then each cell updates these distances carrying out a mutation, if needed, to go from S0-Si to T0- Tj:

b + 1 represent the insertion of Tj
c + 1 represent the deletion of Si
a + 2 represents the substitution of Si for Tj.

Substitution is needed only when Si and Tj are different.

Mutation distance ‘d’ is calculated as,
d=min (a’,b+1,c+1)…..where a’= a if $S_i=T_{j,}$ else a+2
The basic cell is shown in following figure:

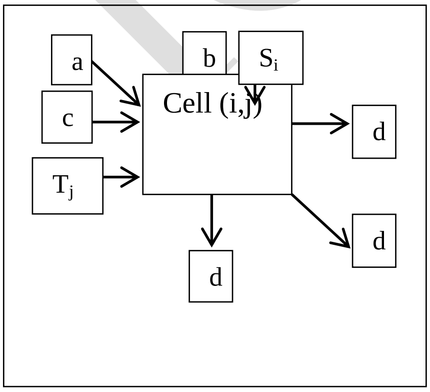

Fig. 2 Basic cell

Finally the option with the smallest penalty is selected, and displayed on LEDs. If this distance is non-zero the result on LCD is ‘DNAs are not matched’ otherwise it is, ‘DNAs are matched’

As LCD display is not available on the Spartan-6 development board, it is externally interfaced with the help of microcontroller. The LEDs display the actual mutation distance count in binary format.

The aim is to check whether the DNAs are having any non-zero mutation distance, or they are having exactly same pairs of proteins (the mutation distance will be zero).

IV. IMPLEMENTATION

For representing the A, T, C, G proteins we have assigned the binary numbers to them. For example consider 00 for A, 01 for T, and so on. We have compared the three pairs in strands of two different DNAs in this work and implemented the algorithm on Xilinx Spartan 6 Family development board device XC6SLx45.

Once the code has been developed and written in Verilog, the file is selected with ‘Add source’ option and adding this file with ‘.v’ extension in the project. It can be written directly in the project also by selecting ‘new source’ option. It gets automatically added in the project with .v extension.

Then the user constraints file is added in the project either by adding ‘.ucf’files as source or writing the one in the project as new source.

Both these files define thesystemand its working along with the output and input.

Fig. 3 FPGA editor window showing the design

After adding these source files, the code is synthesized. We can simply select option ‘Run’ in the software. When the process is finished’. bit’ file is created at the same location as that of the project file.

Further this file is downloaded in the FPGA on board using software provided. In this work Diligent Adept[4] software is used.

V. RESULTS

The mutation distance is obtained as follows:
Consider one example in which source and target sequences are:
Source sequence: T T G
Target sequence: A T G
As only the first pair is different, the mutation distance calculated using algorithm is binary 10 that is, 2.
In the following table results obtained for some different sequences are shown:

Table1. Results for different inputs

Sr. No.	Source sequence	Target sequence	Mutation distance	LED output
1.	TTG	ATG	2	10
2.	CGA	CTA	2	10
3.	AAT	CGA	3	11

VI. CONCLUSION

For comparing the DNA strands in this application Smith-Waterman algorithm is suitable as it reduces the computing, unlike in Neddleman and Wuncsh [6] algorithm, which calculates similarity between two strands. Also for implementing the algorithm on FPGA behavioral modeling is better suited than structural modeling. For displaying final result LCD is used indicating whether DNAs are matched or not. The number of compared protein pairs can also be increased in future scope. In this design it was tested for maximum three pairs. For increasing the count of comparisons more cells of the matrix will be tested for differences in the same way.

REFERENCES

[1]S. L. Wolfe (Ed.) “Molecular and cellular biology.” Wadsworth Publishing Company, 1993.
[2]T. F. Smith and M. S. Waterman, “Identification of common molecular sequences”,J. Mol. Biol.,147:195-197,1981
[3]www.xilinx.com
[4]www.digilentinc.com/adept
[5]Mohmad-ali-mazidi,Janice-Gelispe-mazidi Roline D. Mckinlay, “The 8051 microcontroller & embedded system” Pearson / Prentice hall
[6]S. B. Needleman and C. D.Wunsch “A general method applicable to the search for similarities in the amino acid sequences of two proteins”,J. Mol. Biol.,48:443-453,1970