

Accelerating a charged single α -helix search algorithm in protein sequences using FPGA

Zoltán Nagy*[†] Zoltán Gáspári*, Ákos Kovács*

*Faculty of Information Technology, Pázmány Péter Catholic University, Budapest, Práter u. 50/A., H-1083

[†]Cellular Sensory and Optical Wave Computing Laboratory, Hungarian Academy of Sciences, Budapest, Kende u. 13-17., H-1111

Abstract—Processing and analyzing the immense amount of biological data generated each day requires fast and accurate bioinformatics algorithms. The main characteristic of these algorithms is the low precision input data which can be stored on 2-5bits. Therefore reconfigurable architectures can be used more efficiently compared to conventional 32 or 64 bit microprocessors, because the accuracy of the functional units can be customized and a relatively small amount of configurable logic is required. In the paper a new architecture is introduced and implemented on FPGA to speed-up a structural motif search algorithm. The architecture can be implemented on a versatile medium range FPGA providing about two orders of magnitude speedup. The system can be efficiently scaled to process several sequences in parallel on a larger FPGA resulting in even higher speedup.

I. INTRODUCTION

The charged single α -helix (CSAH) is a relatively recently identified structural motif occurring in about 0.2% of human proteins [1]. Its main characteristics are high charge density and regularly alternating blocks of positively and negatively charged residues. CSAH segments identified so far can act as elastic 'lever arms' in myosins IV and X and linkers or rulers in the inner centromere protein (INCENP) and possibly in paraspeckle proteins. It is important to note that there are very few proteins for which the presence of a CSAH segment was demonstrated experimentally, thus, our consensus-based prediction scheme aims at minimizing the number of false positive predictions.

FPGA devices are widely used to accelerate different sequence search and sequence matching bioinformatics algorithms. One of the well-known algorithms accelerated on FPGA is the Smith-Waterman algorithm [3]. The main advantage of the reconfigurable architectures is that low precision functional units can be used to implement the algorithms. FPGA implementation can also benefit from the low level parallelism of the bioinformatics algorithms because large amount of simple and small functional units can be implemented.

II. THE FT_CHARGE ALGORITHM

For charged single α -helix motif detection two conceptually different computational methods capable of analyzing protein sequences was presented[1]. The SCAN4CSAH algorithm is based on sequence features characteristic, while FT_CHARGE algorithm applies Fourier transformation to charges along sequences (a concept similar to that of Spectral Repeat Finder[2]). Both algorithms were implemented in Perl and targeted to examine the full Swiss-Prot protein database. The input format for both algorithms is the standard FASTA

format, which use one character in a text file to represent one protein in a sequence. The native regular expression matching features of Perl makes SCAN4CSAH very efficient searching the contents of the FASTA files. For the FT_CHARGE algorithm however long time is required (for the full SwissProt database release 201508 containing 594008 sequences, approximately 4 days on an Intel Core i5 2.4 GHz processor) to scan the entire database.

Fourier transformation is regularly used during analysis of biopolymer sequences. In the FT_CHARGE algorithm charges are assigned as follows: -1 for Asp and Glu, +0.5 for His, +1 for Arg, and Lys and zero for any other amino acid residue. During the analysis a 32 or 64 element long window from the full sequence is examined. The window is slid through the entire sequence in one element steps. In the first step a charge correlation function is computed on the window and the results are feed into the second step performing Fast Fourier Transformation (FFT).

The charge correlation function R defined for the sequence as:

$$R(k, n) = \sum_{i=k}^{k+m-n} c(i)c(i+n) \quad (1)$$

where $c(i)$ is the charge assigned to the i^{th} amino acid, m is the length of the window, $1 \leq k \leq l-m$ is the starting position of the current window and l is the length of the sequence.

From each spectra the largest amplitude and its frequency is selected which is used to fit an Extreme Value Distribution (EVD). The fitted distribution is used to adaptively determine the threshold for signaling a charged single α -helix motif.

III. FPGA IMPLEMENTATION CONSIDERATIONS

FPGAs are widely used in signal processing algorithms. Manufacturers of the FPGA devices usually provide an optimized library of signal processing blocks which can be used very efficiently. In our architecture the standard FFT IP core from the Xilinx CoreGenerator was used. The main focus of the paper is the optimization of the computation of the charge correlation function to efficiently feed the FFT cores.

Protein sequences are build from 20 different amino acids which can be encoded in 5 bits in an FPGA implementation. In the case of the charge correlation function, where only four different charge values ($-1, 0, +0.5, +1$) are allowed, the number of bits can be further decreased to 2 bits. The reduction can be executed on the host computer and the FPGA - host

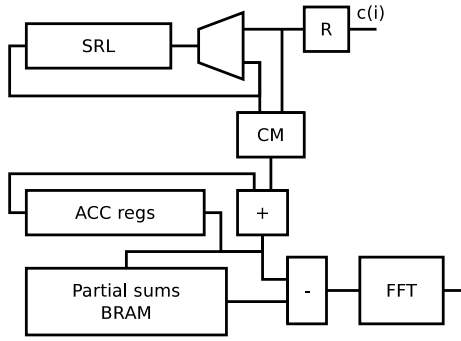


Fig. 1. Sequential charge correlation architecture (R - Register, CM - Charge multiplier LUT, ACC - Accumulator, SRLn - n-length shift register)

communication can be decreased to 25%. Multiplication of the 2bit charges $c(i)$ can be performed by using simple 4-input Look-Up-Tables (LUT). The main goal during the design of the charge correlation function unit was to match its output data-flow to the input data-flow of the FFT IP block. The resulting architecture is shown in Figure 1. where the charge correlation values are computed serially.

In the first step of the operation the new charge is loaded into the input and into the shift register. In the next $m-1$ steps the previously loaded charges $c(i-1) \dots c(i-m+1)$ are read out from the shift register, multiplied by the current charge $c(i)$ and feed back to the shift register. The oldest charge value $c(i-m+1)$ is not required during the following steps and discarded. The length of the ACC register array is also m elements and each store the state of the cumulative sum accumulator for each element in the window. On the FPGA it can be implemented as a shift register because each element should be incremented in every m^{th} cycle. The running sum values are also saved into the Partial sums BRAM which must store $m \times (m-1)/2$ elements and can be implemented using one or two 36kb BRAM instances when the window size is 32 or 64 elements respectively. The difference of the current and old cumulative sum can be directly feed into the input of the FFT IP block.

IV. RESULTS, PERFORMANCE COMPARISON

The proposed system will be implemented on the Zedboard from Digilent Inc. which contains a Z-7020 FPGA, 512Mbyte DDR3 memory running on 1066MHz data rate [4][5]. Communication with the board and loading of the sequences can be carried out via a Gigabit Ethernet interface.

During implementation the Xilinx Vivado HLS was used to speed up the development process. The proposed architecture is divided into four modules forming a pipeline. The first stage is the correlation computing unit described in Section III. The resulting stream is connected to a Xilinx FFT IP core. The next module compute the amplitude and search for the largest component. During the final stage an Extreme Value Distribution is fitted and P-value is computed. If the P-value is under a threshold the parameters of the window is written into the output stream indicating the presence of a CSAH segment. The correlation computing unit and the amplitude computation is pipelined with Initiation Interval (II) = 1, while the EVD fit module can be optimized for area because it is started once for every 32 or 64 clock cycles according to the window size.

TABLE I. AREA REQUIREMENTS OF THE SYSTEM

	LUT	FF	Slice	BRAM	DSP
Correlation	1428	943	518	0	0
FFT	681	1287	320	0.5	6
Amplitude and max	657	640	267	0	2
EVD fit	5117	4130	1693	1	30
Other (AXI-IC, DMAs, etc.)	3781	4621	1381	9.5	0
Complete system	11664	11621	4179	11	38

TABLE II. PERFORMANCE COMPARISON

	Compute time (μ s/window)	Speedup	Clock	Power
Perl	518.012	0.07482	2.5GHz	35W
C++	38.75	1	2.5GHz	35W
FPGA	0.33	117.42	100MHz	1.887W

Area requirements of the complete system in case of 32 element window size is summarized on Table I. We find that increasing the size of the window to 64 elements does not change area requirements significantly. The complete system occupies 32% of the slice resources on the ZedBoard. A conservative approximation of the number of implementable processing units is 3.

Performance of the proposed single processing element system is compared to the original Perl and a C++ implementation, where the FFTW library is used to speed up FFT computations. For testing purposes a small set of sequences used where 1263 windows were computed and the results are summarized on Table II. The Perl and C++ measurements were carried out on dual-core Intel i5-2520M microprocessor running on 2.5GHz clock frequency. The results show that one order speedup can be achieved by simply using C++ to implement the algorithm. Using the proposed processing system with just one computing unit on a cheap FPGA additional two orders speedup can be achieved, while its power dissipation is considerably smaller.

V. CONCLUSION

In the paper we proposed an FPGA based system to speed up charged single α -helix detection algorithm. For efficient computation of the charge correlation function a sequential architecture is proposed and its implementation area and computing performance was compared to a microprocessor. In this case we find that implementing several small sequential units is more efficient. The system was implemented on a relatively cheap FPGA platform and two orders of magnitude speedup was achieved, which makes it possible to examine the SwissProt database regularly. In the future the system will be extended to use several processing units in parallel to enable fast search on larger protein databases.

REFERENCES

- [1] D. Süveges, Z. Gáspári, G. Tóth, L. Nyitray L (2009) *Charged single alpha-helix: a versatile protein structural motif*. Proteins 74:905-916.
- [2] D. Sharma, B. Issac, G.P. Raghava, R. Ramaswamy (2004) *Spectral repeat finder (SRF): identification of repetitive sequences using Fourier transformation*. Bioinformatics 20:1405-1412
- [3] Y. Yamaguchi, H.K. Tsoi, W. Luk, *FPGA-Based Smith-Waterman Algorithm: Analysis and Novel Design*, Reconfigurable Computing: Architectures, Tools and Applications, Lecture Notes in Computer Science volume (6578), 181-192, (2011)
- [4] *Xilinx official webpage (2016)*, <http://www.xilinx.com>
- [5] *Digilent Zedboard official webpage (2016)*, <http://www.zedboard.org>