Elias S. Manolakos, Assoc. Professor
Director Graduate Program "Information Technology in Medicine and Biology"
University of Athens
Dept. of Informatics and Telecommunications
eliasm@di.uoa.gr

Research Interests:

Statistical signal processing, Predictive modeling
High Performance and Embedded Computing

Multidisciplinary Applications:
- Computational & Systems Biology
- Environmental modeling and monitoring

Recent Projects

- Computational Biology
  - Machine learning methods for high throughput proteomics image analysis [*Proteomics 2009*]
  - 3D Visualization methods for integrative proteomics in biomarkers discovery *[J. Biomed. Inform. 2009]*
  - Improving sequence similarity search for protein homology inference using structural information and Bayesian methods
- Systems Biology
  - Efficient architectures for the stochastic simulation of large biochemical reaction networks

Efficient algorithms and architectures for protein 3-D structure comparison

Collaborators

Elias Manolakos (PI), Assoc. Professor, Dept. of Informatics and Telecommunications, University of Athens

Ioannis Emiris, Professor, Dept. of Informatics and Telecommunications, University of Athens

Georgios Panagiotou, Director, Institute of Molecular Oncology, B.S.R.C. "Alexander Fleming"

PhD Candidate: Anuj Sharma

Supported by:
An "ΗΡΑΚΛΕΙΤΟΣ ΙΙ" GSRT grant, 2011-14

Introduction - Motivation

In Protein Structure Comparison (PSC) methods, complexity is introduced along three orthogonal dimensions

- size of structure databases,
- need to support multiple comparison algorithms
- complexity of the pair-wise comparison (PWC) algorithms used.

In order to develop fast and efficient PSC methods, all these factors must be addressed.

Introduction - Motivation

- The modern trend, in the field, is to harness computing capabilities of distributed computing systems (clusters and grids) for the task.
  - This tackles only the first two of the three dimensions.
- Many-core CPU architectures provide a means for exploiting high level of parallelism
  - Intel Single-Cloud Computer (SCC) is an example of a many-core processor with 48 cores organized as a Network on Chip (NoC)
  - Can be used to tackle the first two of the three dimensions in a single CPU
- Field Programmable Gate Arrays (FPAGs) provide the means for exploiting the low level parallelism (PWC)
  - Recently methods using GPUs have also emerged for this task

Research Tasks – Intel SCC Related

- Port popular PSC methods to the SCC NoC architecture.
  - A limited number of third party libraries are available on the SCC
  - RCCE, a Message Passing library, must be used to exploit parallelism on the SCC
- Investigate the impact of using multiple cores on speed up of PSC methods using the Master-Slaves paradigm
  - Data-parallel strategy requires load balancing to hide latency and efficiently utilize computing resources

Research Tasks – Intel SCC Related

- Implement hybrid strategies, distributed/parallel computing, using the SCC for the PSC problem
  - Many-core architectures provide a flexible means for combining distributed and parallel computing
- Implement multi-criteria consensus based PSC using the SCC and factor out common operations
  - PSC methods share common aspects such as extracting protein structure fragments, fragment alignment etc.
  - The cores on the SCC are connected with a high speed network

Research Tasks – FPGA Related

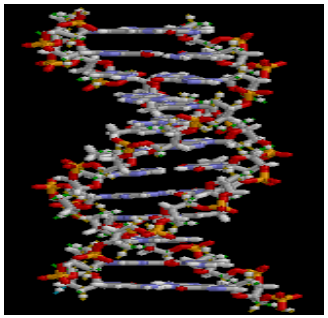
- Develop co-processors based implementation for the TM-align PSC method to provide fine-grained parallelism.
  - The initial alignments performed are independent of each other and could be performed in parallel.
  - Alignment of protein structures, based on pair-wise residue alignment, performed in order to fill a score matrix could be parallelized.
- Develop fast and accurate data compression hardware for Universal Similarity Metric PSC method.
  - USM relies on the calculation of complexities of protein structures by compressing them

Research Tasks – FPGA Related

- Develop parallel Dynamic Programming (DP) implementation for the Combinatorial Extension PSC method.
  - Dynamic Programming alignment of the protein structures is the most time consuming step
- Develop fragment alignment co-processor for multi-criteria PSC
  - Fragment alignment with different metrics, used by different methods, can be performed in parallel
  - Fast hardware implementation of fragment alignment can provide significant speedup to PSC methods

Research Tasks – Large Scale Validation

- Datasets
  - Chew-Kedem Dataset
  - Rost-Sander Dataset
  - SCOP Astral
- Alignments generated by ported algorithms will be compared with those of the baseline algorithms
- Performance testing will be performed comparing the time for all-vs-all comparison using the ported and baseline algorithms



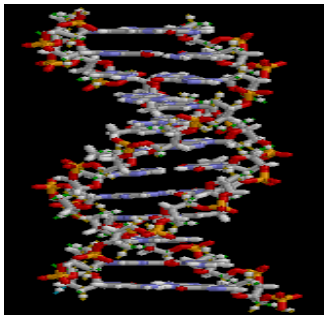
Scalable Reconfigurable SoCs for the Stochastic Simulation of large-scale Biochemical Reaction Networks

Research Team:

Prof. Elias S. Manolakos, PI

PhD Candidate: Evangelos Logaras
MSc Students: Orsalia G. Hazapis, Eleftherios Ouzounoglou

Introduction



Objective:

Perform State prediction of biochemical reaction networks in reasonable amount of time.

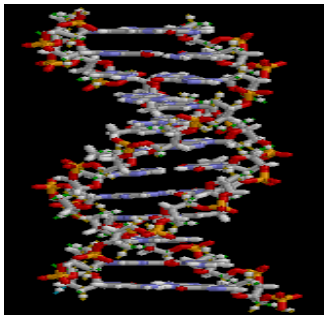
Approach:

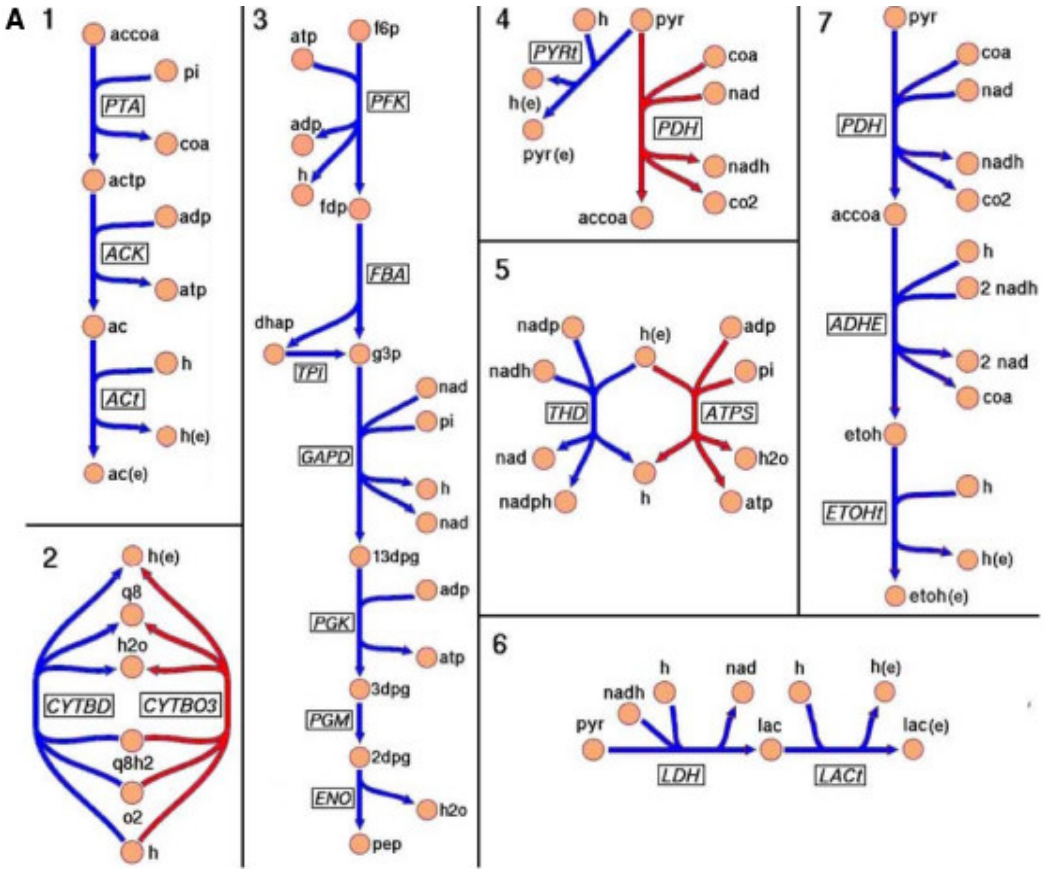
Using Gillespie's Stochastic Simulation Algorithms which attempt to simulate the process of solving the General Chemical Master equation

Our Solution:

Application Specific architectures implemented with FPGAs for high performance, as the problem scales to networks with thousands of reactions

Reactions Networks





MAP-Kinase Reaction Network

State of the art

ReCSiP2 is, a dedicated FPGA board level design that can be used as a PC accelerator.

ReCSiP2 was mapped on the Xilinx Virtex XC2VP70-5 FPGA, achieves operating frequency of 107 MHz and has been shown to outperform a Xeon 2.80 GHz CPU by approximately 80 times when simulating up to *M=1024* reactions.

University of Washington researchers designed circuits that implement a reaction model's operation directly in model-specific circuitry which consists of a large system of distributed logic and random number generator blocks.

State of the art - GPUs

Researchers from the University of Pisa Italy have implemented the FRM SSA on GPUs and accomplished a ~2x in accelerating a single FRM-SSA simulation run, using Nvidia's GEForce 8600M GS GPU to simulate an E-coli bacterial biomodel with only 28 species and 61 reactions.

STOCHSIMGPU is a software tool using Nvidia GPUs to accelerate the SSA implementation in Matlab's System Biology Toolbox 2. The average speedup factor achieved is 85x for simulating a biomodel of 30 reactions.

Our solution – SoC for the FRM-SSA

Designed a SoC for the FRM-SSA using up to *N* = 8 cores (PEs)

The SoC can simulate networks with up to 4096 reactions using a moderate size FPGA

It is synthesized based on a fully parameterized VHDL IP core

Parameters: number of processors *(N)*, number of species *(S)* in the model, number of reactions *(M)*, maximal reactions order *(q)*.

Supports two modes of operation (using *N* PEs):

one simulation running in parallel using N PEs

(each PE runs M/N reactions)

N different simulation runs executed in parallel

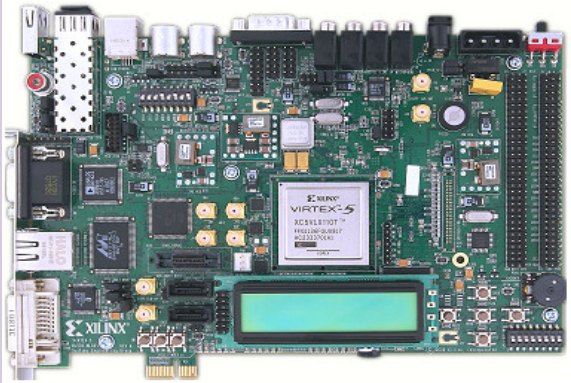
(each PE runs M reactions)

Achieves the highest throughput reported in the literature

(2.6 Mega Reactions cycles/sec)

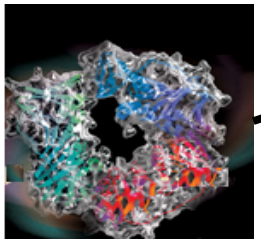
Demonstrated speedup factor 342x compared to i7 Intel CPU (6 GB Ram, 2.66 GHz) for a benchmark biomodel with 512 Reactions

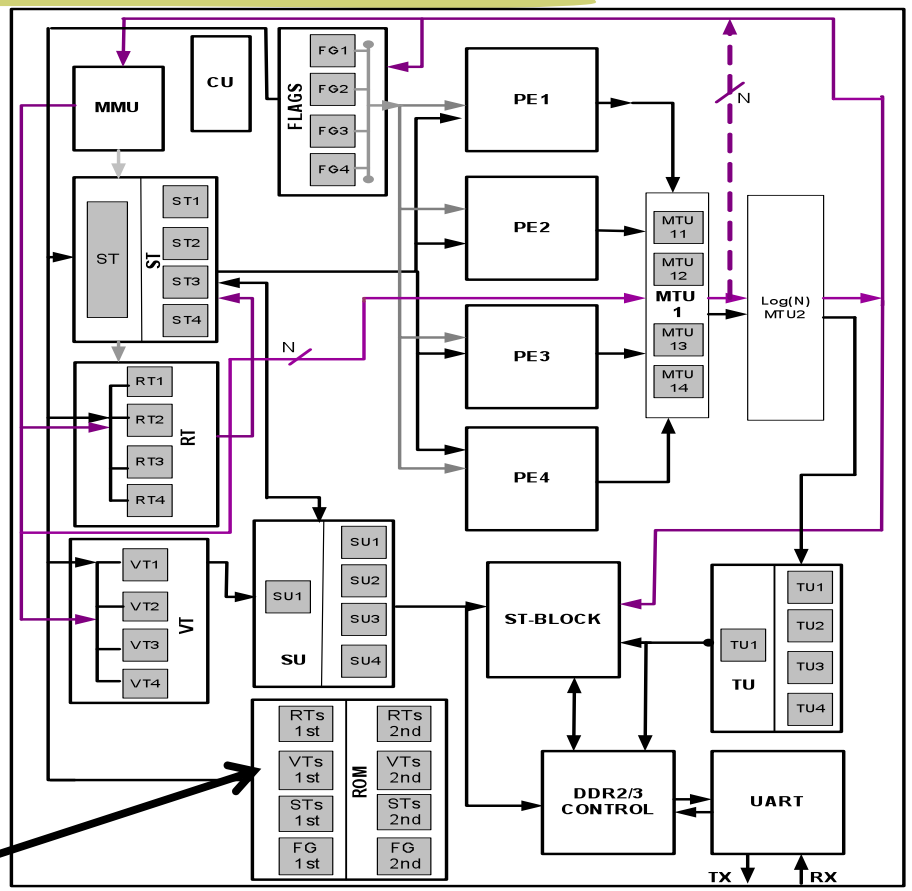
SoC Architecture



Virtex 5 FPGA

Biomodel Information





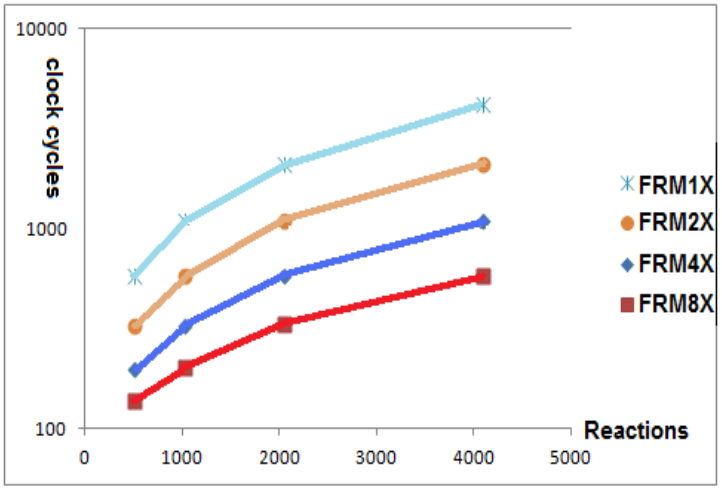
FRM SoC implementation with 4 PE cores.

Simulation and Performance Results

Xilinx (Virtex 5 XC5VLX110T-1ff1136 – Virtex 7 XC7V855T-3ffg1157)				
	FRM1X	**FRM2X**	**FRM4X**	**FRM8X**
Reactions	512	1024	2048	4096
Frequency (MHz)	200-320	200-320	200-320	200-320
Latency (cycles)	**Q (44-70)+9**	**Q (44-70)+11**	**Q (44-70)+13**	**Q (44-70)+15**
Performance (Mreactions/sec)	173- 181*	303-526*	485-776 *	687-1099*
	292- 302**	329-548**	553-922**	832-1332**
LUTs	24779	36919	54557	71641
DSPs	12	24	58	64
REGS	15720	30650	57323	72258
RAM-BLOCKS	53	118	122	141

** Worst case of 3rd reaction order biomodels where the reactants are identical. Performance results respectively for Virtex5 -7*
***Biomodels of first or second order (most usual case). Performance results respectively for Virtex5 -7*

P.P.R synthesis results for the scalable FRMNX SoC implementations (N=1, 2, 4, 8 PEs and up to 3rd order reactions) for Xilinx Virtex 5 and 7 FPGAs



Clock cycles for simulating a reactions cycle with *m=512,1K,2K, 4K* reactions using FRM SoCs with increasing number of PEs.

Work in progress

Design a Multiple Operation SoC to implement different SSA algorithms FRM, NRM, wNRM etc.

Create an on-line platform with FPGAs allowing Biologists to directly connect and simulate any biomodel of large complexity

Extend the SSA capabilities to fully describe bimolecular processes such as diffusion processes , analysis of molecular interactions in 3D space and time.

Orsalia Hazapis, Elias S. Manolakos, "*Scalable FRM-SSA SoC Design for the simulation of Networks with thousands of biochemical reactions in real time*" IEEE FPL pp 459-463, 2011

Thank you for your attention!

Questions?

References

[1] D.T GILLESPIE," Stochastic Simulation of Chemical Kinetics" Annu. Rev. Phys. Chem. V.58 pg 35-55 2007.

[2] M.YOSIMI, Y.OSANA, Y.NISHIKAWA, N.IWANAGA, A.FUNAHASHI, N.H.KITANO, H.AMANO, Y. SHIBATA, Y.IWAOKA,T.KOJIMA "An FPGA implementation of high throughput stochastic simulator for large scale biochemical systems", IEEE FPL pp. 1-6., 2006.

[3] J. KEANE,C. BRADLEY,C. EBELING," A Compiled Accelerator for Biological Cell Signaling Simulations" 12th ACM Int'l. Symp. on FPGAs, pp. 233-241, 2004.

[4] H. LI, L. PETZOLD, "Efficient parallelization of the stochastic simulation algorithm for chemically reacting systems on the graphics processing unit". Int J. High Performance Comp Appl vol 24(2):107, SAGE, May 2010.

[5] Guido Klingbeil, Radek Erban, Mike Giles and Philip K. Maini "STOCHSIMGPU: Parallel stochastic simulation for the

Systems Biology Toolbox 2 for MATLAB" BIOINFORMATICS Vol. 00 no. 00 2010.

[6] LOGARAS E, MANOLAKOS, E.S. "SysPy: using Python for processor-centric SoC design" Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International pp 762 – 765.

[7]http://www.ebi.ac.uk/biomodels-main/

[8] E. OUZOUNOUGLOU, K. VEKRELLI, E. MANOLAKOS" In silico Modeling of Alpha-synuclein Oligomerization Effects on Neuronal Homeostasis "Abstract Book p. 291, PS 578.

[9] M. A.GIBSON and J. BRUCK, "Efficient exact stochastic simulation of chemical systems with many species and many channels," J. Phys. Chem A , vol. 104, no. 9, pp. 1876–1889, 2000.

[10] D.T GILLESPIE," A General Method for Numerically Simulating the Stochastic Time Evolution of coupled chemical reactions" Journal of Computational Physics Vol 22 pg 403-434 1976

[11] Xu and Cai: "Weighted next reaction method and parameter selection for efficient simulation of rare events in

biochemical reaction systems ". EURASIP Journal on Bioinformatics and Systems Biology 2011 2011 :4

[12]ORSALIA-GEORGIA HAZAPIS, ELIAS S MANOLAKOS, Scalable FRM-SSA SoC Design for the simulation of Networks with thousands of biochemical reactions in real time " IEEE FPL pp 459-463 2011.

BACKUP SLIDES

The FRM-SSA

Algorithm 1. First Reaction method

Initialize $X_o(t) = x_o$

While (t < tmax)

{ ***For*** *every j do % start a new reactions cycle*

{ *Evaluate propensity function* $a_j(x)$, generate a random number r_j ;

Determine the time τ_j until the next R_j reaction

$$\tau_j = \frac{1}{a_j} * \ln\left(\frac{1}{r_j}\right)$$

}**end for ;**

If $\tau_\mu = \min(\tau_j)$ **then**

{ **Let** R_μ **be the "winning" reaction resulting the smallest** τj

Determine the new state after reaction R_μ *occurs;*

$t' = t + \tau_\mu$ **and** $X(t) = x + v_\mu$;

}**end if;**

End While ;