

Workshop on  
**Computation in Nanoscale Dynamical Systems**

January 19-20, 2006  
 Bishop's Lodge Resort and Spa, Santa Fe, New Mexico

<b>Thursday, January 19</b>			
8:00 - 8:30a	Breakfast		
8:30 - 9:30	Richard Kiehl	University of Minnesota	"Computation in Nanoscale Dynamical Systems: Motivation and Goals"
9:30 - 10:30	Stephen Wolfram	Wolfram Research	TBA
10:30 - 11:00	Coffee Break		
11:00 - 12:00	Lawrence Gray	University of Minnesota	"Computing with Cellular Automata: What Can I Do with a Very Large Cellular Automata "
12:00 - 1:30p	Lunch		
1:30 - 2:30	Tommaso Toffoli	Boston University	"The Question They All Ask: How do You Make Up a Rule to Give a Desired Behavior?"
2:30 - 3:30	Melanie Mitchell	Portland State University	"Evolving Cellular Arrays to Perform Computations"
3:30 - 4:00	Coffee Break		
4:00 - 5:30	Panel Discussion		TBA
6:00	Dinner		

<b>Friday, January 20</b>			
8:00 - 8:30a	Breakfast		
8:30 - 9:30	Steven Levitan	University of Pittsburg	"Why Smart Pixels Never Graduated, and Other Lessons from the Birth of the VLSI Era"
9:30 - 10:30	Tamas Roska	Hungarian Academy of Science	"Cellular Sensory Wave Computing - CNN Technology in Nanoscale"
10:30 - 11:00	Coffee Break		
11:00 - 12:00	Supriyo Bandyopadhyay	Virginia Commonwealth University	"Toward Realization of a Self-Assembled Nanostructure-Based CNN"
12:00 - 1:30p	Lunch		
1:30 - 2:30	Erik Winfree	California Institute of Technology	"Algorithmic Self-Assembly for Nano-scale Fabrication"
2:30 - 3:30	Christof Teuscher	Los Alamos National Laboratory	"Computation in Irregular and Uncertain Nano-scale Architectures"
3:30 - 4:30	Christian Reidys	Los Alamos National Laboratory	"Sequential Dynamical Systems"
4:30 -5:30	Panel Discussion		TBA
5:30	Adjourn		

## **Supriyo Bandyopadhyay**

Virginia Commonwealth University

“Toward Realization of a Self Assembled Nanostructure Based Cellular Nonlinear Network”

### **ABSTRACT**

We will describe our progress towards the realization of a cellular neural network implemented via a 2-D array of non-linear resistors, each exhibiting a negative differential resistance. The network is self assembled by electrodepositing a semiconductor within the pores of a porous anodic alumina film containing a well ordered hexagonal array of 10-nm sized pores. The semiconductors self assemble into a network of nanowires isolated from each other by an insulating (alumina) barrier.

The nanowires are capped with a metal and behave as MIS diodes exhibiting an S-type negative differential resistance with a peak to valley ratio of about 19:1 at room temperature. The capacitance across each nanowire has been measured to be about 0.5 aF and the inter-wire resistance is about 640 mega-ohms.

Using these measured parameters, we have simulated the performance of this network for various image processing tasks such as edge-enhancement detection. We find that this simple self assembled structure can process images containing 640,000 x 640,000 pixels in less than 1 nanosecond. The same structure can also propagate trigger waves and behave as a biologically inspired network mimicing the Fitzhugh-Nagumo model of impulse propagation along nerve cell membranes.

## **Lawrence Gray**

University of Minnesota

“Computing with Cellular Automata: What Can I Do with a Very Large CA”

### **ABSTRACT**

I will focus generally on what can be said, from a mathematical point of view, about the following topic:

If I could build a very large 2-dimensional CA, and if I had some flexibility with the transition rule for the CA, what could I do with it?

There is one very general answer to this question, arising from the work of Peter Gacs, which is that I would be able to reliably perform any computation of any size, even if the cells in the CA are subject to random errors, and I could do it in a way that is easily scalable, in the sense that I wouldn't have to change the error-correction scheme no matter how big the computation gets. Furthermore, the time and space requirements would essentially be only constant multiples of what is possible with completely reliable components.

There are also some specific answers to the question. For example, CA's seem to be particularly good at generating random numbers. They also seem to be good at simulating certain situations that involve emergent behavior, such as traffic jams.

## **Richard Kiehl**

University of Minnesota

“Computation in Nanoscale Dynamical Systems: Promise and Challenges”

Can nanoscale electronic hardware based on massive arrays of ultrasmall components enable the development of powerful computing systems and subcomponents?

Proposals for nanoscale dynamical systems (NDS) include locally connected arrays of simple nonlinear components (e.g., negative differential resistance elements) and of dynamical components (e.g., integrate-and-fire elements). Such arrays are capable of performing simple image processing tasks and memory.

But, can arrays of simple locally connected nonlinear components provide functions beyond basic image processing tasks? Can the richness of nonlinear dynamics compensate for the simplicity of nanoscale

components? Can the local communication and limited connectivity of cellular automata, and certain random Boolean networks, alleviate the interconnect bottleneck and reduce power consumption? Do studies of nonlinear dynamics in the neocortex provide useful models for implementing NDS? And finally, can these systems be made to work within limiting constraints imposed by lithographic resolutions, fine-grained regularity of self-assembly technologies, sustainable power-dissipation densities, low manufacturing yields and high error rates?

This talk will introduce these and other questions on the promise and challenges of computation in nanoscale dynamical systems.

## **Steven Levitan**

University of Pittsburg

"Why Smart Pixels Never Graduated, and Other Lessons from the Birth of the VLSI Era"

### **ABSTRACT**

In this talk I will characterize some of the successes and failures from the explosion in computational paradigms brought on by the birth of the VLSI era, and summarize the lessons we can learn about the relationship of technology, devices, components, and systems for computation. In the 1970s and 1980s VLSI technology promised integration levels which would support thousands of cellular processors for image processing, database and simulation tasks; however, today most computing is still done on uniprocessors or small clusters.

The question arises, "what is different now?" How might the application of nanotechnology to the realization of cellular automata be more successfully adopted? In particular, I will focus on how decisions at the physical, logical and architectural levels affect the hard problems of interconnect, power, memory, I/O and ultimately, scalability.

## **Melanie Mitchell**

Portland State University

"Evolving Cellular Arrays to Perform Computations"

### **ABSTRACT**

In this talk I will review research on using evolutionary algorithms to design cellular automata and related cellular architectures to perform computations. I will describe these evolutionary algorithms, including a particularly successful one called "coevolutionary learning", and will sketch some methods for understanding the "collective computation" that such algorithms design. I will also discuss some effects of asynchronous updating and non-local communication on computation in cellular arrays. Finally, I will start a discussion on the prospects for these methods in designing useful nanoscale electronic hardware, in the hopes that the audience will continue this discussion at the end of the talk.

## **Christian Reidys**

Los Alamos National Laboratory

"Sequential Dynamical Systems"

### **ABSTRACT**

Sequential dynamical systems (SDS) are a class of dynamical systems that generalize classical CA or RBN concepts in several ways. An SDS consists of the following (a) a graph  $Y$  whose vertices have a state, (b) a mapping  $L: v[Y] \rightarrow \{X \subset Y\}$  from  $Y$ -vertices into subgraphs of  $Y$  and a vertex labeled local map which updates all vertices contained in  $L(v)$  in parallel and finally (c) some word according to which the local updates are scheduled. Accordingly, an SDS has sequential and parallel features and can in the extreme case update synchronously all vertices at once or only neighborhoods of distance 1 in a sequential way. Furthermore, an SDS allows for the immediate state-changes of all vertices contained in  $L(v)$ , generalizing the concepts of CA and RBN where only the state of the updated vertex changes. Structural results on SDS include the characterization of equivalent systems, the combinatorial classification of update schedules,

morphisms of SDS which allow to relate the original system to simpler systems, analysis of periodic points and in particular fixed points. SDS concepts have been applied in the context of gene-regulatory networks and packet/information flow in networks, where we developed SDS-protocols for load balanced routing.

## **Tamas Roska**

Hungarian Academy of Science and Pazmany University

"Cellular Sensory Wave Computing - CNN Technology in Nanoscale"

### **ABSTRACT**

Nowadays, cellular focal plane visual microprocessor chips can operate with 50,000 frame per sec, integrating optical sensing and stored programmable computing, and its new, low power version is dissipating only 100 mw with 10, 000 frames per second containing 25,000 cell processors , having 100,000 optical sensors (4 in each cell), implemented on a 180 nm CMOS technology.

The core of the cellular sensory wave computer architecture implemented in many different ways is the Cellular Nonlinear Network (CNN dynamics).

First, this dynamics and different versions of its input driven spatial temporal properties will be introduced, including the local activity principle.

Next, the CNN Universal Machine architecture, a universal machine on image flows, will be shown, defining the ways how the elementary instructions as cellular wave dynamics can be reprogrammed dynamically and also tuning the input sensor array locally. The classes of problems as well as the design of the different local interaction patterns, the cloning templates, will be reviewed.

A summary of various physical implementations will be followed by the description of recent CMOS embedded nano-scale implementation experiments, including the nano-antenna arrays, the CMOL architecture, the RTD arrays, and the function-in-layout principle.

Finally, the biology relevance will be motivated by the CNN models of the mammalian retina, in view of the recent discovery of its multi-channel operating principle, as well as the non-synaptic signal transmission in the brain.

## **Tommaso Toffoli**

Boston University

"The Question They All Ask: `Yes, It's Lovely to Watch the Behavior of a Given Rule -- But How do You Make Up a Rule to Give a Desired Behavior?'"

### **ABSTRACT**

This "inverse problem" question has two distinctly different kinds of answer. We shall briefly review concepts and facts for the two cases.

If we are willing to microprogram the desired behavior in all its gory details, like in a video game, then it's "just" a matter of writing the appropriate program in the cellular automaton's low-level language---the local rule itself---with suitable initial data. For this we need at the very least a rule that is computation-universal, and hopefully one in which the operations that we want done can be programmed easily and can be made to run efficiently in space and time. This is essentially the art of choosing and programming an FPGA.

A conceptually more exciting answer is to have the desired behavior EMERGE, in a statistical-mechanical sense, as the macroscopic attractor of a dynamics. There are no easy recipes for this (the challenge is akin to designing a PDE for a given behavior rather than integrating a given PDE). However, the chief task is to decide which features of the dynamics will be averaged out by the statistics, and which instead shall "percolate upwards" and be retained at the macroscopic level. Thence the stress on symmetries and conserved quantities, on the degree of irreversibility of the dynamics, and on the signs (positive and negative), the coefficients, and the spatio-temporal delays of the various feedback loops. Think of linear analysis with a small dose of nonlinear steroids.

## **Christof Teuscher**

Los Alamos National Laboratory

"Computation in Irregular and Uncertain Nano-scale Architectures"

### **ABSTRACT**

Making the nano-computing revolution happen involves a wide variety of challenges on different levels and in different fields. The path to universal, reliable, faster, and economically viable computation beyond today's transistor integrated circuits is highly uncertain and it is still unclear what computing architecture and what nano-scale devices will allow for a major breakthrough. The goal of this talk is to delineate promising candidate approaches by rethinking the way we build, (self-) organize, and program future nano-scale computer architectures. While most proposed architectures assume a homogeneous, regular, and precise arrangement of components and interconnections, it is unclear whether this can be achieved in large-scale assemblies and whether it is even necessary. Using CAs and RBNs as a showcase, I will outline ways to reliably compute in irregular, inhomogeneous, and uncertain assemblies of simple components and highlight tradeoffs and paths for genuine hardware realizations. A particular focus shall be directed on various types of interconnect architectures and their assets and drawbacks. Further, I will discuss the need and solutions for reversible computation in vast assemblies of nano-scale electronics and will also outline unconventional ways for self-adaptation, self-(re)configuration, self-repair, and new paradigms for organizing and programming systems.

## **Erik Winfree**

California Institute of Technology  
"Algorithmic Self-Assembly for Nano-scale Fabrication"

I will review recent advances in algorithmic self-assembly, including theoretical models of programmable growth, self-repair, and error correction, and the experimental demonstration of DNA Wang tiles that self-assemble to create periodic arrays, tubes, ribbons, Sierpinski triangles, and binary counting patterns.

## **Stephen Wolfram**

Wolfram Research

TBA