

# Extended Analog Computers: A Unifying Paradigm for VLSI, Plastic and Colloidal Computing Systems

Jonathan W. Mills<sup>1</sup>, Bryce Himebaugh, Andrew Allred,  
Daniel Bulwinkle, Nathan Deckard, Natarajan Gopalakrishnan,  
Joel Miller, Tess Miller, Kota Nagai, Jay Nakamura, Bayo Olowoye,  
Radu Vlas, Philip Whitener, Myint Ye, and Chen Zhang

Computer Science Department  
Indiana University  
Bloomington, Indiana 47405

<sup>1</sup> Contact primary author at this address: Professor Jonathan W. Mills, Computer Science Department, Indiana University, Bloomington, Indiana, 47405, USA, 812-855-6486, jwmills@cs.indiana.edu

## *Abstract*

*Rubel's extended analog computer defines a new paradigm for VLSI, plastic and colloid computers. For some applications they offer performance that is on a par with digital supercomputers. Extended analog computers have a canonical structure composed of partial differential equation solvers and continuous-valued logic units. They are reconfigured digitally so that the same machine can solve a wide variety of applications. The prototypes are inexpensive, and became operational over the Internet in 2004. These are real systems.*

## **1. Introduction**

At the keynote address for the 2002 International Symposium on Computer Architecture and Systems, Bob Colwell predicted that the failure of Moore's Law would lead to a new epoch in computer architecture. He attempted to predict what the new epoch might bring, but also admitted that any predictions would be unlikely to match the future. He was right.

He did not expect that the new chips and systems would be inherently and implicitly massively parallel supercomputers. He did not expect that they would have only two instructions that would be incompatible with all previous digital computer instruction sets. Nor did he expect that the first completely operational Internet-accessible prototypes would be built for only \$250 apiece. His list of

possible new technologies did not include analog (although he did hedge his bets by including non-CMOS devices). Although these new machines have been fabricated with traditional planar VLSI technology, he did not expect that they could also be implemented by literally cutting their processing units out of plastic foam with scissors or injection-molding three-dimensional blocks of conductive plastics. He certainly did not anticipate that one would be built out of strawberry Jell-O® brand gelatin.

These machines are *extended analog computers*. They implement a new computing paradigm. In September 2004, six were built, debugged, became operational over the Internet and were used to prototype a wide variety of applications by students at Indiana University. In the past few months researchers at several universities around the world have expressed interest in using these publicly-available computers.

## **2. Extended Analog Computers**

Extended analog computers (EACs) are defined by a recent theoretical model of analog computation [Rubel 1993]. They are not general purpose analog computers (GPACs) [Shannon 1941], nor are they one of the many analog neural network models implemented in silicon VLSI [Mead 1989] and studied theoretically [Siegelman/Sontag 1992]. EACs represent a new paradigm for computer architecture that until recently was believed to have no physical realization [Campagnolo/Moore/Costa 2000].

The *general purpose analog computer (GPAC)* is the mathematical model that is usually thought of when an “analog computer” is mentioned. The archetypal analog computer in this class is the differential analyzer, a mechanical, and later electronic, computer invented by Vannevar Bush of MIT. Using operational amplifiers as integrators, constants, multipliers and adders, specific configurations of GPACs were used to solve ordinary differential equations. However, GPACs could not directly solve systems of partial differential equations. These problems, known as field problems, were solved with various conductive solids, liquids, or resistive meshes but were not reconfigurable for widely different applications [Karplus 1958]. After the advent of digital computers, hybrid digital-analog computers solved field problems by iterating “across” the field, varying the boundary conditions with each trial.

The *extended analog computer (EAC)* avoids these difficulties and limitations. An EAC is composed of a general purpose analog computer to which Rubel added a “quintessential boundary-value problem solver”. In practice this is a partial differential equation solver. Those partial differential equations that are strictly boundary-value problems are Laplacian; those that have internal conditions are Poisson. Many problems described by these equations, such as heat diffusion and internal mechanical stress, may be solved electronically [Karplus 1958]. Observations that vertebrate brains solve field problems to generate spatio-temporal maps led Rubel to extend the GPAC with functional elements that performed field computation directly [Rubel 1985, Rubel 1993]. Rubel’s assertion that “...*the problem of implementation is to describe physical, chemical, and/or biological devices that do in the real world what the black boxes are supposed to do. Since the EAC is so broad, this can never be done in practice...*”, suggested that no practical extended analog computer architecture would be possible. However, research into the continuous-valued Lukasiewicz logic as a computational paradigm led to an electronic implementation of the extended analog computer.

A *Lukasiewicz logic array (LLA)* is an H-tree array that electronically implements a schema

for sentences of Lukasiewicz logic [Mills/Beavers/Daffinger 1990]. Lukasiewicz logic arrays approximate GPACs sufficiently well for practical purposes. A little-known theorem validates this substitution by proving that algebraic ordinary differential equations—the class of equations solved by GPACs—are approximated arbitrarily closely by Lukasiewicz logic [McNaughton 1950].

*Lukasiewicz logic* has a denumerably infinite number of truth values. It describes a class of ideal analog circuits that have an infinite maximum precision. Real analog circuits, which have a finite maximum precision, are classified by the number of data values encoded on individual wires in the circuit. Boolean logic is the most familiar subset of Lukasiewicz logic: it describes binary digital circuits. Continuous-valued analog circuits are described by subsets of Lukasiewicz logic with  $2 \leq n \leq 2^p$ , where  $p$  is the maximum measurable precision of the circuit in bits. Externally  $p$  falls in the range  $6 \leq p \leq 12$ . Internally, the precision is finite but sufficiently “large” enough to not impede the computation. Our experience over the past five years has shown that for many applications, especially those where massive data sets are reduced, or where biological systems are modeled, or when non-symbolic values are desired, the precision of the output need never be measured explicitly. The “collapse” of internal precision at the end of the computation does not detract from the value of the result (see examples in 6. Applications).

### 3. Architecture

The architecture an extended analog computer is composed of a finite number of “levels” of functional units with numerous interconnections, some providing feedback within and between each layer (Figure 1).

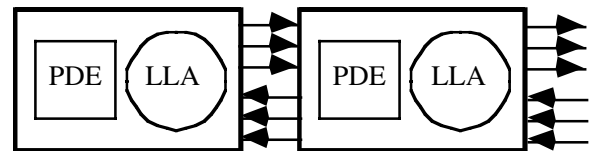


Figure 1. Extended analog computer architecture

In practice two or three layers of partial differential equation solvers operating in parallel are sufficient for many applications. Each level is composed of one conductive sheet, and from two to six Lukasiewicz logic arrays in the various prototypes (Figure 2).

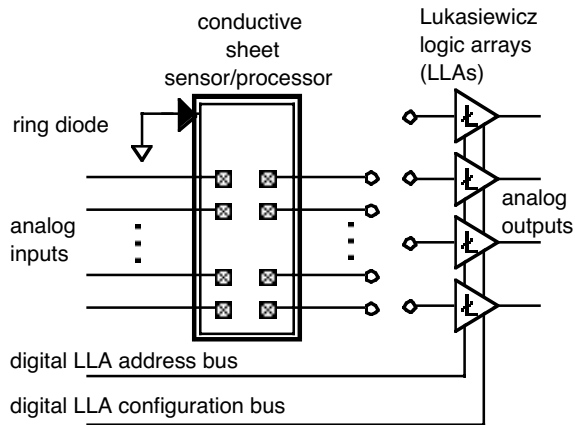


Figure 2. Extended analog computer schematic

Few interconnections are needed between levels, a benefit for VLSI designs. In a digital computer, wire is necessary to move data and instructions between memory and the central processor. This creates the well-known von Neumann bottleneck. In an extended analog computer, computation is pervasive, and often even the wires compute. And in an interesting twist, the computational elements often are massive interconnects. So, while some parts of the machine may not contribute directly to the computation, it is unusual for any part to restrict it. For example, the pattern-recognizing sensor merges detection and recognition, performing the computation continuously over the entire image (Section 6, Applications).

The components of a extended analog computer fall into two categories. The first category is computational and implements the two primary instructions: SPDE (solve partial differential equation) and CPLF (compute piecewise linear function), as well as simple addition and subtraction performed by wire junctions. The partial differential equation solvers are implemented as conductive sheets. Piecewise linear function generators are implemented as reconfigurable Lukasiewicz logic arrays, which are usually nothing more than binary trees of

diode-connected field-effect transistors. Addition and subtraction is performed with wire junctions, and is governed by Kirchoff's current laws. Components in the second category manipulate data. Current mirrors duplicate values for use in multiple locations. Wires explicitly pass parameters between spatial locations in the machine. Conductive sheets serve to communicate values and simultaneously scale their sum.

#### 4. Implementation

Extended analog computers can be implemented with silicon VLSI and many other conducting and semiconducting materials. For example, some of the most interesting materials (because they permit hybrid analog-digital systems on a uniform substrate) are semiconducting organic polymers. They were discovered by physicists Heeger, MacDiarmid, and Shirakawa. Garnier developed a polymer field-effect transistor using their discovery. Extended analog computers could use these technologies in novel ways, for example, to print disposable application-specific supercomputers.

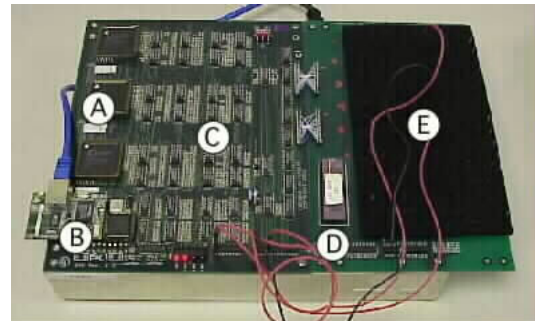


Figure 3. Extended analog computer

One of the six Internet-accessible extended analog computers (EACs) built at Indiana University is shown with its components labeled (Figure 3). The analog (continuous-valued) Lukasiewicz logic arrays are digitally reconfigurable (A). An Ethernet interface connects the EAC to the worldwide web (B). Analog-to-digital converters convert the continuous output to digital values for transmission and external measurement (C). A socket permits different components to be used (D). A VLSI chip is shown here, but Jell-O® brand gelatin was also connected to the EAC to

study three-dimensional colloid computers (Figure 4). Cultured neural tissue, organic semiconductors, etc. can be attached instead. The prototype is always configured with a sheet of conductive foam that solves partial differential equations in microseconds (E). The VLSI chip at (D) solves the same equations in nanoseconds.



Figure 4. Jell-O® brand gelatin processor

The layout (Figure 5) and a photomicrograph of an EAC fabricated by MOSIS in 1995 (Figure 6) are shown. The actual chip was limited to only two LLA units due to pin limitations of the 40-pin package. This is the same VLSI chip used in the Internet-accessible EAC. It has been in continuous use in various prototypes since 1995, often with the cover removed. It is so robust that it still works after being cleaned of dust and dirt with a soft paintbrush!

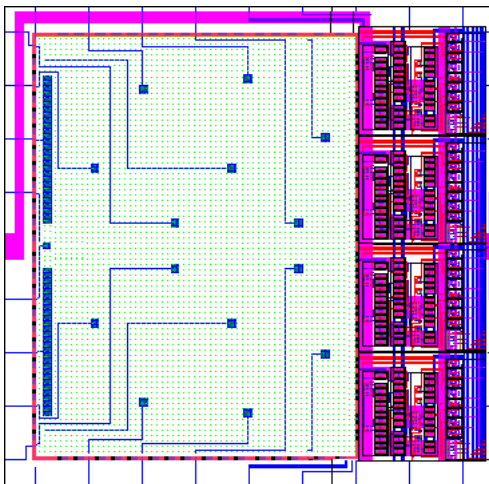


Figure 5. Extended analog computer layout

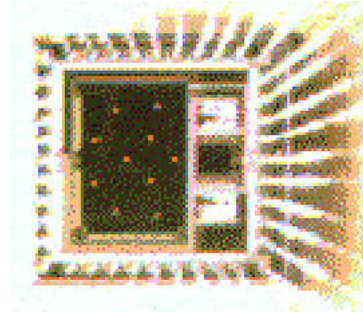


Figure 6. VLSI extended analog computer chip

## 5. Programming

In a GPAC, the hardware is the software, and vice versa. The “programs” were separate pieces of the hardware. To change the program, one changed the machine’s structure. The user carried a part of the GPAC, called a “patch board” because it was a tangle of wires and components interconnected on a circuit board, from the office to the computer where the “program” was run. The wiring on the patch board was changed if an error was detected. After each use the patch board was disconnected and returned to storage. GPAC programs were cumbersome, at best.

An extended analog computer has a uniform, canonical structure that provides a general framework for many different computations. The hardware is standardized like a digital computer rather than a GPAC. Although the computation is carried out simultaneously at many points in the architecture, like a GPAC, these points are easily reconfigured, in the manner of a digital computer. A vector of bits, similar to a program for a digital computer controls the reconfiguration. The configuration vector is called an *overlay*, expressing the idea that the function of the machine changes over all components, rather than in a restricted location—the memory—as occurs in a stored program digital computer.

An overlay for an extended analog computer is developed in a fashion similar to an application specific field-programmable gate array. A digital computer automatically generates the overlay, most recently by using a genetic algorithms and a fluid model of the conductive sheet (similar to the fluid model of the field-effect transistor). In a research seminar conducted at Indiana

University in the fall of 2002, an overlay for the design prototype of the current extended analog computer was described using a genome. Genetic algorithms were developed by students to evolve overlays for several diverse applications. The results were encouraging. Starting with populations of 10,000 individuals, each a randomly generated bit vector, the chosen applications were evolved within 1000 generations. But recognizing the standard structure of the canonical processor, and observing that an analog computer has a strong correlation between the problem semantics and machine, the students reduced the population to 1000 individuals, the number of generations to three, and the time to find a solution to several minutes on a Pentium-class laptop. These results are still being explored. One possibility is to dynamically evolve control overlays for extended analog computers used as embedded controllers.

## 6. Applications

For those applications within their scope, extended analog computers are inexpensive high-performance scientific computers. On early concern was that EACs could not do symbolic computing at all. However, a study by Mills in the early 1990's identified the relationship between extended analog computers and reaction-diffusion systems. Turing used such systems to model morphogenesis [Turing 1952]. Later, an alphabet defined by butterfly wing patterns served to illustrate the ability of EACs to generate and manipulate text, although at a staggeringly inefficient cost (Figure 7).



Figure 7. Butterfly alphabet [Sandved 2000]

### 6.1 Exclusive-OR, a simple neural network

The next application for the extended analog computers, exclusive-OR, was difficult to design because it was so simple. Finding the correct approach to the implementation took over one year. This application is significant because the ability to compute exclusive-OR distinguishes general neural networks from perceptrons, and also illustrated the conceptual difficulty of “applied overcomputation”, that is, using the power of a partial differential equation solver as a single instruction for a simple logic function. In retrospect the solution was trivial, but there was no existing frame of reference within which the problem could be understood at that time [Hedger 1998]. A complex aggregate of simple components is encapsulated in an EAC. The conductive sheet can be viewed as a network of leaky neurons (Figure 8). The individual neurons’ functions are distributed evenly spatially. The network’s connectivity and fixed weights (due to distance) are a function of the sheet. The network’s nonlinear weights and threshold functions are computed by the Lukasiewicz logic arrays.

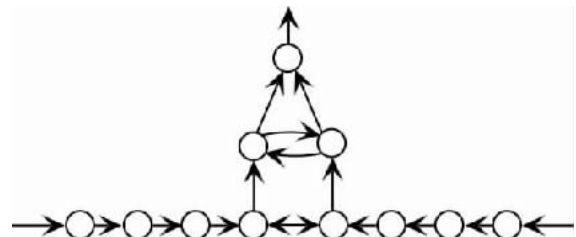


Figure 8. Network of “leaky neurons”

Exclusive-OR is a simple non-linearly-separable function. As inputs are presented at each edge of the neuronal array—the conductive sheet—the output at the center is mapped to this truth table:

(0,0):	0.0
(0,1):	0.5
(1,0):	0.5
(1,1):	1.0

The conductive sheet “forces” the (0,1) and (1,0) inputs to the same value, 0.5, but cannot produce the correct final output. The intermediate value, 0.5, is taken from the center of the array and passed through an “inverted notch” that is one of the piecewise linear functions generated by the Lukasiewicz logic array. This performs the

functionality of a pair of mutually recurrent and inhibiting neurons to produce the exclusive-OR at the output (Figure 9).

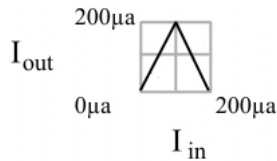


Figure 9. Inverted notch function

The output, shown as it is computed by the Jell-O® brand gelatin conductive solid, is a three-dimensional and continuous-valued exclusive-OR. It corresponds to Boolean exclusive-OR at the endpoints and midpoint along the centerline of the function (Figure 10).

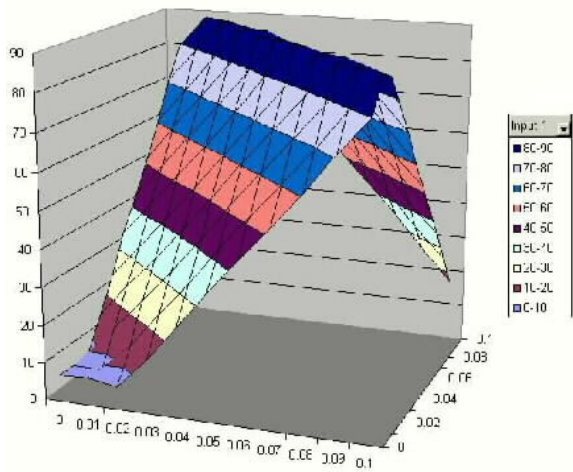


Figure 10. Output of three-dimensional XOR

By the fall of 2004, many more applications had been prototyped using silicon and plastic conductive sheets. All applications used a canonical form of the extended analog computer. Some were evolved using exclusive-OR as the starting point for genetic algorithms.

## 6.2 Pattern recognition

One application found frequent use. It is a single-pixel pattern recognizer that appears in various guises as the Hamiltonian Circuit detector, a cyclotron beam line controller, an Internet router using the Random Early Dropout algorithm, and a real-time ceramic spall detector for lubricating systems. This application will be discussed in detail.

The pattern recognizer is a one-level extended analog computer (see previous Figures 5 and 6). When implemented in silicon VLSI, the pattern is projected onto the sheet, which serves a dual purpose as a photodetector.

Used as a trainable character recognizer, a task often performed by neural networks, the recognizer is trained to distinguish between “A” and “T” to illustrate how the piecewise linear functions implement reconfigurability. A binary encoding for the characters is chosen, with “A” indicated by 111 and “T” by 000. The functions are initially set to produce a 1 for every input to the sensor. Initially presenting an “A” does not generate an error, so no adjustment is necessary (Figure 11).

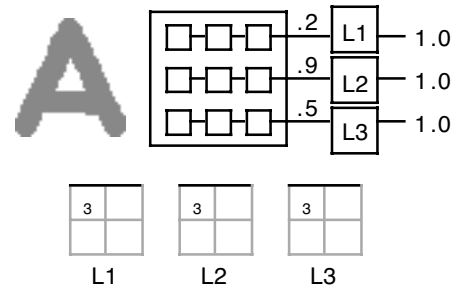


Figure 11. Initial presentation of “A”

When the “T” is presented, an error results, which can be used with the desired output and the values input to the Lukasiewicz logic arrays to determine the new functions. A simple rule for choosing the functions is to minimize the error between the current presentation and the most recent but different character presented. Following this rule, the new functions are shown in Figure 12.

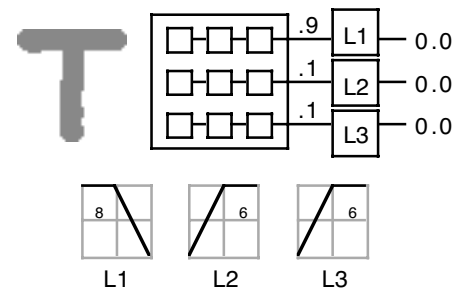


Figure 12. Presentation of “T”

Presenting “A” again indicates that this choice of function settings is acceptable. Finally, the ability of the sensor to classify characters as either “A”, “T”, or similar to one or the other is tested with four images: a scrawled “A” and a “T” with a circular patch removed (Figure 13).

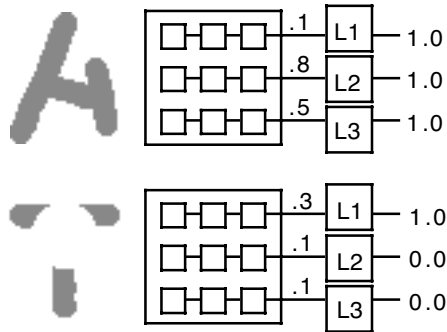


Figure 13. Categorization of similar characters

The recognizer is robust because the induced current gradient is continuous. The conductive sheet is inherently capable of smoothing discontinuities in the image.

The next three applications illustrate the range of problems computable by digitally reconfiguring the same extended analog computer.

### 6.3 An NP-C problem: Hamiltonian Circuit

NP-Complete problems are computationally difficult. EACs cannot solve these problems in polynomial time, but may offer useful speed-ups. One factor is the ability to find an analogy for the problem, such as the one students used to recognize small instances of Hamiltonian Circuit in graph images (Figure 14).

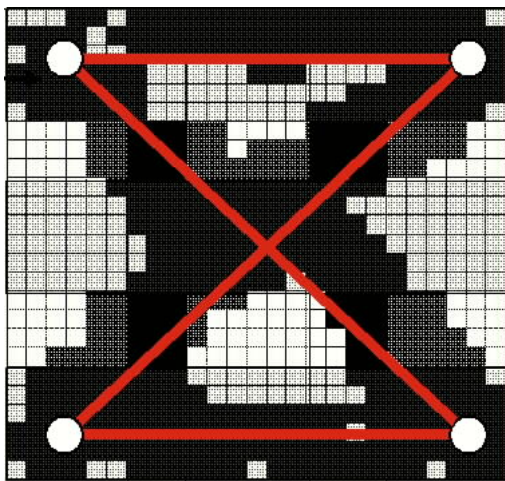


Figure 14. Hamiltonian Circuit

This approach suggests that circuits in larger images may also be detectable, even though there are theoretical difficulties [Minsky/Papert 1969]. Research in neural networks was stalled due to their observations for years before pragmatic approaches were suggested, for example in [Hopfield 1982]. EACs may open a return to similar avenues for theoretical study.

### 6.4 Image Rendering

Extended analog computers diffuse current through planar and solid conductive media. This led to the hypothesis that media with different conductances would model the behavior of light, even in diffuse media. A prototype application illustrated that this approach, which is not radiosity-based, but similar to it (one computer scientist called it “image rendering with physics!”). Multiple levels of the EAC are used to reflect, diffuse or transmit the incident light on the objects in this two-dimensional implementation. Light sources (bright spots) are current sources; objects (dark squares) are current sinks; the resultant image is shown in color (Figure 15). This application merits further study with three-dimensional EACs.

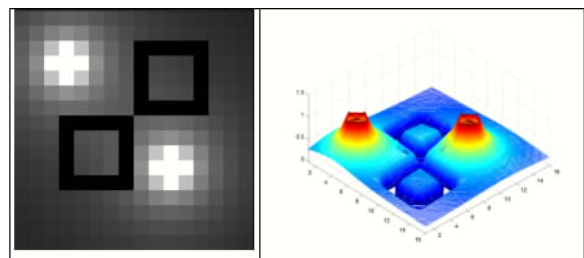


Figure 15. Radiosity-based image rendering

### 6.5 Chaotic Dynamical Systems

Finally, extended analog computers implement natural randomness, and have been shown to generate chaotic dynamical systems with multiple attractors. This research is very recent, but with real EACs instead of software simulators, trajectories similar to the Lorenz attractor have been generated (Figure 16). In this case, Lukasiewicz logic functions that are non-monotonic over their domain, such as the inverted notch function (see previous Figure 9) orbit around two stable points in a state space. Recurrent connections through the conductive

sheet serve to delay the feedback to the function, and also add noise to perturb it slightly. Only when the function escapes from a basin of attraction, that is, approaches the inflection point in the non-monotonic function, does it change its orbit to the second attractor. Designing chaotic dynamical functions for EACs is important. It is the only way that these systems can implement “memory” without relying on digital latches or analog sample-and-hold circuits. Our research is in its infancy, with a large body of external work that must be related to this new paradigm.

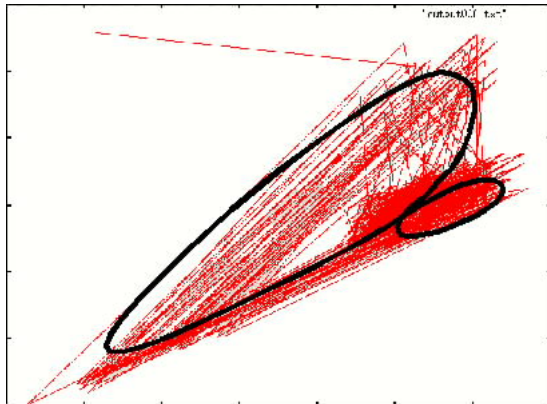


Figure 16. Two-state chaotic dynamical system similar to the Lorentz attractor

## 7. Summary

As our understanding of extended analog computers improves, applications that challenge digital computers continue to emerge, and new architectures evolve. EACs will surely roam some electronic continent in the fourth epoch of computer architecture predicted by Bob Colwell!

## 8. References

- [Campagnolo/Moore/Costa 2000] Iteration, Inequalities, and Differentiability in Analog Computers. Santa Fe Institute. <http://www.santafe.edu/~moore>.
- [Colwell 2002] Computer Architecture, The Next Epoch. Keynote address, Int. Symp. Comp. Arch. And Sys. 2002.
- [Hedger 1998] Analog Computation: Everything Old is New Again. Indiana University Research & Creative Activities, [XXI:2], pp. 24-27. <http://www.indiana.edu/~rcapub/v21n2/p24.html>
- [Hopfield 1982] Neural networks and physical systems with emergent collective computational

properties. Proc. Nat. Acad. Sci. USA, **79**, 2554-2558.

- [Karplus 1958] Analog Simulation, McGraw-Hill: New York.
- [Maass/Sontag (no date)] Analog Neural Nets with Gaussian or other Common Noise Distributions Cannot Recognize Arbitrary Regular Languages. Email: maass@igi.tu-graz.ac.at.
- [McNaughton 1951] A Theorem About Infinite-Valued Sentential Logic. Journal of Symbolic Logic, 16 (1950), pp. 1-13.
- [Mead 1989] Analog VLSI and Neural Systems, Addison-Wesley: Reading, Massachusetts.
- [Mills/Beavers/Daffinger 1990] Lukasiewicz Logic Arrays. Proceedings of 20th International Symposium on Multiple-Valued Logic. Charlotte, North Carolina: IEEE Press.
- [Mills 1992] Area-Efficient Implication Circuits for Very Dense Lukasiewicz Logic Arrays. Proceedings of 22nd International Symposium on Multiple-Valued Logic. Sendai, Japan: IEEE Press.
- [Mills 1998] U.S. Patent 5,770,966. Area-Efficient Implication Circuits for Very Dense Lukasiewicz Logic Arrays.
- [Mills/Walker/Himebaugh 2003] Lukasiewicz' Insect: Continuous-Valued Robotic Control after Ten Years. Int. Jour. Multiple-Valued Logic.
- [Minsky/Papert 1969] Perceptrons. Cambridge, MA: MIT Press.
- [Olowoyeye 2003] Image Rendering using Wave Theory of Light and Extended Analog Computers. Indiana University Computer Science Department Technical Report 581.
- [Rubel 1985] The Brain as an Analog Computer. *J. Theoretical Neurobiology* **4** (July 23).
- [Rubel 1993] The Extended Analog Computer. Adv. In Appl. Math. 14, 39-50.
- [Sandved 2000] The Butterfly Alphabet Poster.
- [Seigelmann/Sontag 1992] On the Computational Power of Neural Nets, Proc. Fifth ACM Workshop on Computational Learning Theory, Pittsburgh, July 1992, pp. 440-449.
- [Shannon 1941] Mathematical Theory of the Differential Analyzer, J. Math. Phys., MIT, 20 (1941) pp. 337-354.
- [Turing 1952] The Chemical Basis of Morphogenesis, Phil. Trans. Roy. Soc. Lond. **B237**, pp. 37-72 (1952).