The New Computer Science and Its Unifying Principle

Complementarity and Unconventional Computing

Jonathan W. Mills
Computer Science Department
Indiana University
Bloomington, Indiana 47405

1. Introduction

Unconventional computing has been looming on the horizon of computer science for more than a decade. Its shadow is cast by Moore's Law, the physical limits of computation (such as *c,* the speed of light), non-procedural languages based on logic and dataflow, and the complexity of all aspects of computation, ranging from networks to programs. But without a unifying principle to bridge the gap between the old and new paradigms, and some compelling architecture *and* its implementation, the new computer science is dead computer science.

The principle proposed here, *complementarity,* is broad enough to explore the links between all models of unconventional computing, from analog to quantum. It is based on exposing dualities, then unifying them. Complementarity was not identified easily—new paradigms sound better when they are dramatic replacements for the old—but there is too much of value to discard in conventional computing, just as its weaknesses have grown too obvious to ignore.

In the rest of this document the relationship between conventional digital computers and Rubel's extended analog computer, or EAC, will be briefly described, and a series of three vignettes for the practical application of EACs will be offered. These little visions have become increasingly realistic. EACs are operational and in use at Indiana University–Bloomington in the U.S.A. At the March 2005 Workshop on Unique Chips and Systems, demonstrating an EAC over the web raised interest in silicon VLSI versions of this novel kind of computer. Such machines would augment conventional digital computers by adding single-cycle machine instructions to solve partial differential equations. These have extremely broad applications, ranging from artificial intelligence, to computational chemistry and pharmaceutical design, to video games. IBM, Toshiba and Centaur, an Intel Pentium-clone manufacturer who is using analog technology to support network security and cryptography, expressed interest in learning more about EACs for the first time, and also recognized their economic viability. What might Microsoft do with Intel processors with 100-nanosecond machine instructions that solve arbitrary, easily reconfigurable partial differential equations?

Some suggestions are presented in this paper.

2. The Extended Analog Computer and Complementarity

Historically, analog computing has been used to simulate specific systems. It has not been a paradigm for universal computation, but it is this paradigm we will examine here, fictionally and playfully near the end of this paper. However, the theory and practice of universal analog computation is real. It is not as implausible as the typical reader might believe.

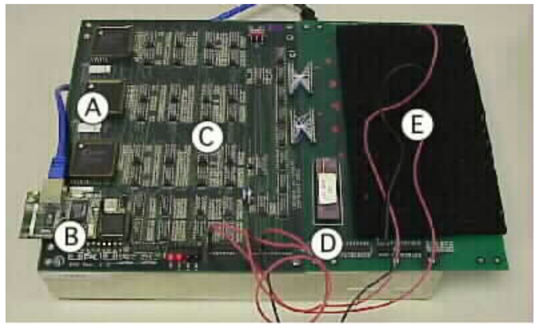

Figure 1. Extended Analog Computer

A number of increasingly sophisticated prototypes of universal analog computers have been built since the mid 1990's at Indiana University (Figure 1). The most recent version is composed of Lukasiewicz logic functions (A), an Ethernet interface to the world wide web (B), analog-to-digital converters for the Ethernet interface (C), and both silicon VLSI (D) and conductive plastic (E) partial differential equation solver units. The prototypes have been used to study applications that include cognitive models of intelligent agents, solvers for general instances NP-complete problems, image classifiers, feedback controllers, and numerical accelerators for protein folding computations. In theory the device is universal; in practice it supplements digital computers exactly where they "run out of steam," that is, in the difficult-to-program and costly-to-run massively-parallel Grand Challenge problems that use matrix manipulation to solve partial differential equations. EACs solve partial differential equations directly and blindingly fast.

Analog computation balances digital computation. The "Tao of Computing" can be envisioned as digital *yin* and analog *yang*. Both are necessary. Each contains a tiny core of the other. Either alone meets limitations in practice. Together they are unified by complementarity. Here are a few examples of this duality:

Digital	Analog
Inherent bottleneck	Inherent "waterfall"
(computation restricted by CPU and memory)	(computation occurs everywhere, always)
Sequential processor	Parallel processor
Internally fixed precision	Externally fixed precision
Precision increases temporally	Precision increases spatially
Explicit error-correction	Implicit error-correction
(coding)	(structure)
Algorithms	Analogies
Many simple instructions	A few complex instructions
Small problems are easy	Small problems are difficult
Programming model is	Programming model is
modular and temporal	holistic and spatial
Lexical (tokens)	Visual (whole systems)

Vannevar Bush championed computing by analogy with his Memex, believing it to be the most natural way for humans to interact with computers. However, just as Ted Nelson's Xanadu changed as technology enabled it in the form of the World-Wide Web, so future analog computers will almost certainly be radically different from the prototypes in use today.

3. Extended Analog Computers of the (Increasingly Near) Future

There are supercomputers in my toaster!

You wake up, go into the kitchen, and put four supercomputers to work. Inside the toaster an array of ceramic heating squares begins to glow red hot. A fiber optic sensor array with binary optic lenses, only approximately focused on the muffin but capable of precise measurement with their overlapping "eyes," transmit the infrared reflection of the surface of your muffin back to four extended analog computers.

Each one registers the changes in infrared intensity to develop a toasting map for the surface of your muffin it monitors, and having been configured by the settings "light brown" and "whole wheat" raises or lowers the temperature of each ceramic tile to produce a heat map that will toast your muffin perfectly, with no burnt edges, and that colder spot in the middle perfectly done. When the muffin halves are a crusty, even, light brown, they pop up, piping hot. You did not set a timer, but simply sipped a bit of coffee while your ToasterKing checked the progress of your muffin. After the muffin popped up, the sensors inside were cleaned automatically, ready to transmit the raw data for the next computation. You leave the four supercomputers idle until tomorrow morning, ready to make perfect toast—if you want it. You might choose to have a bowl of cereal instead.

Crayon Inside! (an analog supercomputer in your laptop)

"Crayon" became the popular name that applied to the extended analog computers that are now found everywhere, from embedded controllers in toasters to massively-parallel workstations. Your laptop, built in 2011 A.D., has a "Crayon" inside that accelerates the intensive matrix operations needed to compute the animations, visual displays of data, gesture processing, and neural field interactions that turn your laptop into a quasi-intelligent and devoted assistant.

Your accelerator looks unimpressive from the outside, being a steel-gray flat square embedded in optical memory slabs. But on the inside gigabytes of data are used to digitally configure its functions in a nanosecond-by-nanosecond response to the analog computation, and binary optic lenses—nothing more than fancy Fresnel lenses—are used to translate the binary data into optical inputs, and split the analog outputs back to binary to perform the equivalent of trillions of computations each second. Because the machine is an analog computer, it doesn't actually perform each operation explicitly, as a digital computer would, so this number cannot be measured exactly. Instead, analog computers rely on the laws of physics to perform the mathematics implicitly. This is one of the characteristics of analog computing: the precision of the computation is "buried" inside the analogy.

The "Crayon" processor looks like a tiny "sandwich", but is a marvel of micromachining and VLSI fabrication. The two pieces of "bread" that make up the sandwich are the input and output layers, an array of infrared laser dots and a binary optic lens to focus data onto the "meat" of the sandwich: a carefully implanted thin layer of an organic semiconductor that solves partial differential equations. Output is obtained from another slice of "bread": an array of tunneling electron microscope probes that measure the picosecond changes in voltage in the conductive sheet, which feed the Lukasiewicz logic arrays that are configured to the specific function needed for each 20 picosecond computing cycle.

This tiny "sandwich" is the equivalent of dozens of parallel digital processors that can be individually configured to perform complex steps in an even more complex computation, and cycle results back to optical memory quickly enough to perform thousands of trillions of operations each second. What is amazing is that these operations are not the ones that your laptop uses when you write a letter in your word processor, or balance your checkbook. Most of the tasks you do are the same things a digital computer did 60 years ago—and still does today.

What differs is the operating system you use, and the need for extremely rapid recognition of the gestures you make above the flat keyslab, and the keyspots you touch, and the mistakes your laptop has learned that you make, so that you can look at what appears to be a desk with paper and tools on it, and use those tools just by gesturing. A lot of the Crayon's power goes into its artificial intelligence and agent routines, using the field computing paradigm pioneered by MacLennan and applied by others to create the powerful recognition and behavior systems that lie at the heart of the agents that you interact with. They run your programs in a DWIM operating system: "Do What I Mean". It does, too—or it asks!

In an extended analog computer, much of the operating system and many non-symbolic applications are implemented with parameter templates for the hardware, not algorithms and programs. In effect, an analog computer is "programmed" by vast look-up tables, a very simple concept (and one that has had an equally vast effect on computer scientists, who suddenly saw much of their "science" disappear as it was replaced by physics). With template-based configuration for virtual realities and AI systems, programming the operating system and the specialized virtual worlds became elevated to an art form similar to sculpting, painting, or playing a musical instrument. New doctorates now apprentice themselves to "masters" to learn the "feel" of the computer, an art that they refine over time with the help of an army of intelligent agents who have watched their predecessors. Let's meet an artist at work.

Massively parallel analog biocomputation evolves new pharmaceuticals

Where your laptop has a single Crayon inside, a modern workstation has over one million. Where workstations in the late 20th century had four, eight, 16 or possibly 32 processors, a workstation today must have a massively parallel array of "Crayons," supported by digital processors to perform symbolic and discrete computations. The Evolution Engine has one square meter of "Crayons" inside, and an array of 1024 digital Neo processors from Intel®, each Neo roughly equivalent to ten 1990's-era Pentium processors. The overall computing power of the single Evolution Engine is not definable using any of the metrics used by computer scientists at the turn of the millennium. There is no way to measure the operation of a Crayon in terms of MIPS, MOPS, or MFLOPS. It is, after all, a non-algorithmic non-Von Neumann machine.

Let us talk with Dr. Sanjay Patel, Senior Design Artist of Artificial Pharmaceuticals, Inc., Madras, as he works with a massively-parallel extended analog computer, the Evolution Engine he has nicknamed "Eve".

"Do you see those little green ants over there? They are part of Eve. That's the name of my Evolution Engine. You can think of the way she operates as splitting herself up to assemble a random collection of bushes from the ones that met my criteria. Those bushes are pharmaceutical molecules. When the bushes are planted then Eve goes to work to evolve them. That's where the Crayon array is used. Each bush goes through a series of tests against membrane patch identification, neurotransmitter activity, bond energy levels (which are critical by the way, we have to build in a way to flush these undesirable pharmaceuticals out of the system after they have finished their work), and reaction path sequences. She can solve one trillion ERIs—electron repulsion integrals—per second.

“All of these tasks would have taken a digital processor centuries to perform, if the mathematics had to be done in binary. In fact, the first, simple artificial pharmaceutical designs took several years to complete on a network of digital supercomputers. I have been told that the biggest problem was getting the machines to talk to each other using their different primitive programming languages over a clumsy network that spanned the continent. Eve fits in my office!

“Eve is fast. She does the computations we need to evaluate a molecular structure in seconds. She can evaluate a million molecules at a time, or just ten. Why would I do that? Well, I use Eve at different resolutions to increase or decrease the precision with which I manipulate the molecule. Eve provides the scaling I need to solve really interesting problems. Yes, my old professor thought that analog processors had a limited precision, but the poor fellow had never seen an analog computer like a Crayon before. And the analog computers his advisor used before that were built out of discrete components—op amps, resistors, etc. It’s a pity, but it was actually good for India. The third world took advantage of the first world’s stubborn obsession with quantum to concentrate on these obsolete “analog” machines, and now the United States sends its students here for training—if they can afford it, poor people. The cost of living here is getting absolutely outrageous for foreigners.”

4. Conclusions

1950s-era analog computers are obsolete. Current digital computers are reaching their limits, or have reached them. Useful quantum computers have yet to be built.

Extended analog computers are here now. A new generation of students at Indiana University is learning to use them, *and apply them to real-world problems.* And, perhaps foreshadowing more changes than computing paradigms alone, interested students come from India, China and Africa.

And why shouldn’t they be interested? Powerful EACs can be built out of even the oldest silicon VLSI process technology, injection molded plastics, or even Jell-O® brand gelatin. EACs are simple, cheap, powerful, and too old-fashioned for the First World to study—the new computing is well-suited for the Third World, and that is where the author expects it to be adopted first.

There is a precedent. Edward O. Deming was unable to interest the automotive industry of Detroit, Michigan, in the U.S.A. in statistical process control. In the 1950’s, Japanese automakers brought him to their country, and adopted his techniques. By 1990, in part to curb import restrictions, Japan was building new automotive manufacturing plants in the United States, while Detroit was closing plants and laying off its workers.