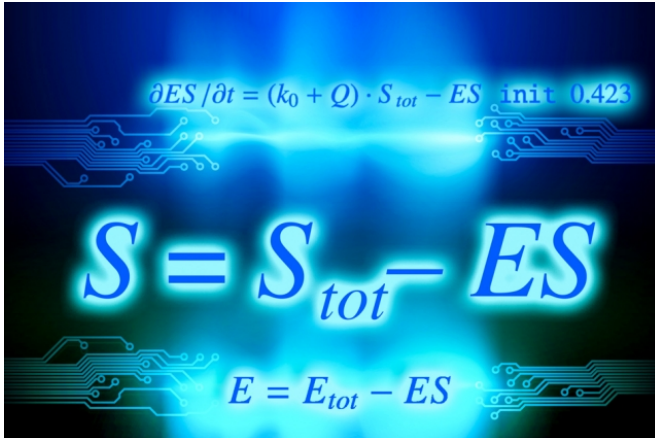


New analog compiler could help enable simulation of whole organs and even organisms

21 June 2016, by Larry Hardesty



The researchers' compiler takes as input differential equations and translates them into voltages and current flows across an analog chip. Credit: Jose-Luis Olivares/MIT

A transistor, conceived of in digital terms, has two states: on and off, which can represent the 1s and 0s of binary arithmetic.

But in [analog](#) terms, the transistor has an infinite number of states, which could, in principle, represent an infinite range of mathematical values. Digital computing, for all its advantages, leaves most of transistors' informational capacity on the table.

In recent years, analog computers have proven to be much more efficient at simulating biological systems than digital computers. But existing analog computers have to be programmed by hand, a complex process that would be prohibitively time consuming for large-scale simulations.

Last week, at the Association for Computing

Machinery's conference on Programming Language Design and Implementation, researchers at MIT's Computer Science and Artificial Intelligence Laboratory and Dartmouth College presented a new compiler for analog computers, a program that translates between high-level instructions written in a language intelligible to humans and the low-level specifications of circuit connections in an analog computer.

The work could help pave the way to highly efficient, highly accurate analog simulations of entire organs, if not organisms.

"At some point, I just got tired of the old digital hardware platform," says Martin Rinard, an MIT professor of electrical engineering and computer science and a co-author on the paper describing the new compiler. "The digital hardware platform has been very heavily optimized for the current set of applications. I want to go off and fundamentally change things and see where I can get."

The first author on the paper is Sara Achour, a graduate student in electrical engineering and computer science, advised by Rinard. They're joined by Rahul Sarpeshkar, the Thomas E. Kurtz Professor and professor of engineering, physics, and microbiology and immunology at Dartmouth.

Sarpeshkar, a former MIT professor and currently a visiting scientist at the Research Lab of Electronics, has long studied the use of [analog circuits](#) to simulate cells. "I happened to run into Rahul at a party, and he told me about this platform he had," Rinard says. "And it seemed like a very exciting new platform."

Continuities

The researchers' compiler takes as input [differential](#)

[equations](#), which biologists frequently use to describe cell dynamics, and translates them into voltages and current flows across an analog chip. In principle, it works with any programmable analog device for which it has a detailed technical specification, but in their experiments, the researchers used the specifications for an analog chip that Sarpeshkar developed.

The researchers tested their compiler on five sets of differential equations commonly used in biological research. On the simplest test set, with only four equations, the compiler took less than a minute to produce an analog implementation; with the most complicated, with 75 differential equations, it took close to an hour. But designing an implementation by hand would have taken much longer.

Differential equations are equations that include both mathematical functions and their derivatives, which describe the rate at which the function's output values change. As such, differential equations are ideally suited to describing chemical reactions in the cell, since the rate at which two chemicals react is a function of their concentrations.

According to the laws of physics, the voltages and currents across an analog circuit need to balance out. If those voltages and currents encode variables in a set of differential equations, then varying one will automatically vary the others. If the equations describe changes in chemical concentration over time, then varying the inputs over time yields a complete solution to the full set of equations.

A digital circuit, by contrast, needs to slice time into thousands or even millions of tiny intervals and solve the full set of equations for each of them. And each transistor in the circuit can represent only one of two values, instead of a continuous range of values. "With a few transistors, cytomorphic analog circuits can solve complicated differential equations—including the effects of noise—that would take millions of digital transistors and millions of digital clock cycles," Sarpeshkar says.

Mapmaking

From the specification of a circuit, the researchers' compiler determines what basic computational operations are available to it; Sarpeshkar's chip includes circuits that are already optimized for types of differential equations that recur frequently in models of cells.

The compiler includes an algebraic engine that can redescribe an input equation in terms that make it easier to compile. To take a simple example, the expressions $a(x + y)$ and $ax + ay$ are algebraically equivalent, but one might prove much more straightforward than the other to represent within a particular circuit layout.

Once it has a promising algebraic redescription of a set of differential equations, the compiler begins mapping elements of the equations onto circuit elements. Sometimes, when it's trying to construct circuits that solve multiple equations simultaneously, it will run into snags and will need to backtrack and try alternative mappings.

But in the researchers' experiments, the compiler took between 14 and 40 seconds per equation to produce workable mappings, which suggests that it's not getting hung up on fruitless hypotheses.

"Digital' is almost synonymous with 'computer' today, but that's actually kind of a shame," says Adrian Sampson, an assistant professor of computer science at Cornell University. "Everybody knows that analog hardware can be incredibly efficient—if we could use it productively. This paper is the most promising compiler work I can remember that could let mere mortals program analog computers. The clever thing they did is to target a kind of problem where analog computing is already known to be a good match—biological simulations—and build a compiler specialized for that case. I hope Sara, Rahul, and Martin keep pushing in this direction, to bring the untapped efficiency potential of analog components to more kinds of computing."

More information: Configuration synthesis for programmable analog devices with Arco. [DOI: 10.1145/2908080.2908116](https://doi.org/10.1145/2908080.2908116)

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

APA citation: New analog compiler could help enable simulation of whole organs and even organisms (2016, June 21) retrieved 27 September 2020 from <https://techxplore.com/news/2016-06-analog-enable-simulation.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.