

ROC: A Reconfigurable Optical Computer for Simulating Physical Processes

JEFF ANDERSON, ENGIN KAYRAKLIOGLU, SHUAI SUN, JOSEPH CRANDALL, and
YOUSRA ALKABANI, The George Washington University
VIKRAM NARAYANA, Intel Corporation
VOLKER SORGER and TAREK EL-GHAZAWI, The George Washington University

Due to the end of Moore's law and Dennard scaling, we are entering a new era of processors. Computing systems are increasingly facing power and performance challenges due to both device- and circuit-related challenges with resistive and capacitive charging. Non-von Neumann architectures are needed to support future computations through innovative post-Moore's law architectures. To enable these emerging architectures with high-performance and at ultra-low power, both parallel computation and inter-node communication on-the-chip can be supported using photons. To this end, we introduce ROC, a reconfigurable optical computer that can solve partial differential equations (PDEs). PDE solvers form the basis for many traditional simulation problems in science and engineering that are currently performed on supercomputers. Instead of solving problems iteratively, the proposed engine uses a resistive mesh architecture to solve a PDE in a single iteration (one-shot). Instead of using actual electrical circuits, the physical underlying hardware emulates such structures using a silicon-photonics mesh that splits light into separate pathways, allowing it to add or subtract optical power analogous to programmable resistors. The time to obtain the PDE solution then only depends on the time-of-flight of a photon through the programmed mesh, which can be on the order of 10's of picoseconds given the millimeter-compact integrated photonic circuit. Numerically validated experimental results show that, over multiple configurations, ROC can achieve several orders of magnitude improvement over state-of-the-art GPUs when speed, power, and size are taken into account. Further, it comes within approximately 90% precision of current numerical solvers. As such, ROC can be a viable reconfigurable, approximate computer with the potential for more precise results when replacing silicon-photonics building blocks with nanoscale photonic lumped-elements.

CCS Concepts: • **Computer systems organization** → **Reconfigurable computing**; *Interconnection architectures*; • **Hardware** → **Emerging architectures**; **Emerging optical and photonic technologies**; **Plasmonics**;

Additional Key Words and Phrases: PDE solver accelerator, approximate computing, nanophotonic computing, analog computing, optical computing, post-Moore's law processors

This work is funded by the NSF RAISE program as Award No. 1748294 under the NSF EPMD-ElectroPhotonic Mag Devices, CSR-Computer Systems Research, Networking Technology and Systems.

Authors' addresses: J. Anderson, E. Kayraklioglu, S. Sun, and J. Crandall, The George Washington University; emails: {jeffa, engin, sunshuai, jwcrandall}@gwu.edu; Y. Alkabani, Halmstad University; email: yousra.alkabani@hh.se; V. Narayana, Intel Corp; email: vikramkn@ieee.org; V. Sorger and T. El-Ghazawi, The George Washington University, 800 22nd Street NW, 5000 Science and Engineering Hall, Washington, DC 20052; emails: {sorger, tarek}@gwu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2329-4949/2020/03-ART8 \$15.00

<https://doi.org/10.1145/3380944>

ACM Reference format:

Jeff Anderson, Engin Kayraklioglu, Shuai Sun, Joseph Crandall, Yousra Alkabani, Vikram Narayana, Volker Sorger, and Tarek El-ghezawi. 2020. ROC: A Reconfigurable Optical Computer for Simulating Physical Processes. *ACM Trans. Parallel Comput.* 7, 1, Article 8 (March 2020), 29 pages.

<https://doi.org/10.1145/3380944>

1 INTRODUCTION

The recent push for post-Moore computer architectures [1] has introduced a wide variety of application-specific accelerators [2, 25, 37, 41]. Generally, these accelerators are designed to improve the performance of computationally intensive algorithms by limiting unnecessary calculations or data movements. To maximize an application-specific computer's utility, it must be capable of accelerating widely used algorithms of some critical applications.

Partial Differential Equations (PDEs) are a core mathematical tool in scientific computing and engineering, and are used to model physical phenomena such as fluid dynamics [29], electricity [30], magnetism [31], mechanics [32], optics [33], and heat flow [10]. Due to the difficulty in solving such equations, multiple techniques have been invented to simplify their calculation, such as transformations into ordinary differential equations (ODEs) and numerical methods executed by computers. However, solving PDEs includes multiple iterative stages that are computationally intensive, and much attention has been paid to creating efficient implementations of PDE solvers aimed at reducing the number of iterations [11, 12]. As PDEs form the basis for many applications in scientific computing, efficiencies gained in this domain would be of great benefit to the scientific community [42].

One specific implementation of a PDE solver, the resistance network analogue, or more generally an analog mesh computer, uses a network of resistors to solve PDEs. This architecture was originally developed to provide efficient computation of heat transfer [13] and oscillatory flow problems in aeronautical engineering [14], and has been shown to reduce the time to solution through its elimination of the iterative processing steps that plague numerical methods (Figure 1). However, due to complexities surrounding the effective integration of a static analog mesh computer in a very-large-scale-integration (VLSI) architecture, the resistance network analogue has been relegated to an academic curiosity [15]. This shortcoming was improved upon by Ramirez-Angulo and DeYong [15] with a VLSI-friendly implementation of an analog mesh computer using Complementary Metal Oxide Semiconductor (CMOS) transistors operated in the subthreshold regime. However, modern digital VLSI designs prefer the use of minimum-size devices, which is at odds with subthreshold CMOS designs, which require larger devices to ensure proper matching [16]. Recent programs advocating for new and innovative computer architectures [1], and the recent introduction of innovative, programmable VLSI devices, such as the nanophotonic modulator [43], have created opportunities for innovative architectures that can take advantage of these new devices [44, 63], leaving the analog mesh computer well positioned for inclusion in future computer architectures.

It is with this in mind that we introduce the Reconfigurable Optical Computer (ROC); the first reconfigurable nano-optical class of computing processors [8]. Shown in Figure 2, ROC is capable of performing scaling operations and summations using diffusion currents in its metatronic incarnation and signal intensity in its photonic and plasmonic implementations. The metatronic implementation refers to utilizing metamaterial inspired optical nano-circuit elements. These elements are analogous to electronic lumped circuit elements such as nanocapacitors, nanoinductances, and nanoresistors [3]. Increases in the resistive-capacitive (RC) time constant as an electric mesh scales serves as a motivation for an optics-based solution. Shown in Table 1, a ROC implemented with

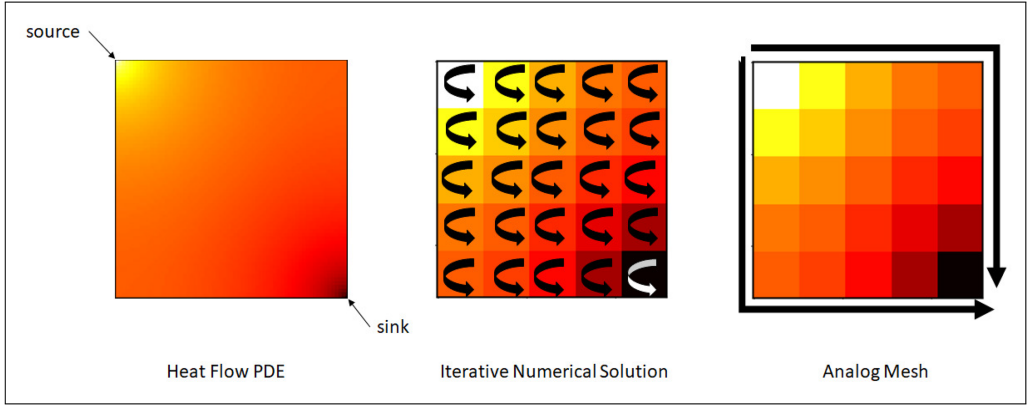


Fig. 1. PDE solution of heat flow over a homogeneous surface (left). Numerical solutions decompose the surface into an array of subsurfaces, or computational mesh, and iteratively solve for each subsurface. Subsurfaces with smaller areas yield higher-precision results, but require more computations to arrive at a solution. Analog mesh computers (right) use an array of analog components (e.g., resistors) to directly compute the numerical solution of the PDE, eliminating the iterative component required by traditional numerical solutions. A denser array of analog components increases precision.

metatronics provides a multiple order of magnitude benefit in terms of footprint, operating frequency and power when compared to all other mesh solutions. However, silicon-photonics and plasmonic implementations of ROC only yield a performance advantage against electric meshes.

Fully programmable, ROC, is capable of being reconfigured for different PDE problems, sizes and efficiencies by modifying optical connections, making it an appropriate architecture for inclusion in a static VLSI architecture, which places an emphasis on reprogrammability of components. Additionally, the ability to change the resolution of the PDE through reconfiguration makes this appropriate for future Energy-Quality (EQ) scalable systems, which require the ability to explicitly trade off energy and quality at different levels of abstraction [40], and are fundamental to green computing initiatives.

1.1 Scope

In general, the analog mesh computer can be used to solve a wide variety of problems found in the physical sciences [15]. In particular, the resistance network analogue has been used to solve a variety of physical problems based on steady-state computation [13, 14, 45]. Other variations of this architecture are capable of solving time-dependent variations, such as the Poisson Equation, Diffusion Equation and the Wave Equation. This flexibility makes the analog mesh computer important for modeling of density functional theory, atmospheric diffusion, seismic waves, and fluid flow, among other things. In this article, however, we will focus solely on steady-state computation by ROC.

While a metatronic ROC is introduced here, this article focuses on the design of a photonic implementation of ROC. Photonic components will allow us to approximate the ideal metamaterial-based implementation of ROC using materials and fabrication processes readily available. Upper bounds are placed on the accuracy by λ/L , where L is the feature size of the network components, due to the physical nature of light. However, these inaccuracies can be overcome by reconfiguration of the waveguide attenuation.

This article is organized as follows: Section 2 covers background material relevant to analytical and numerical solutions of PDEs, to include analog mesh computers. This is followed by Section 3,

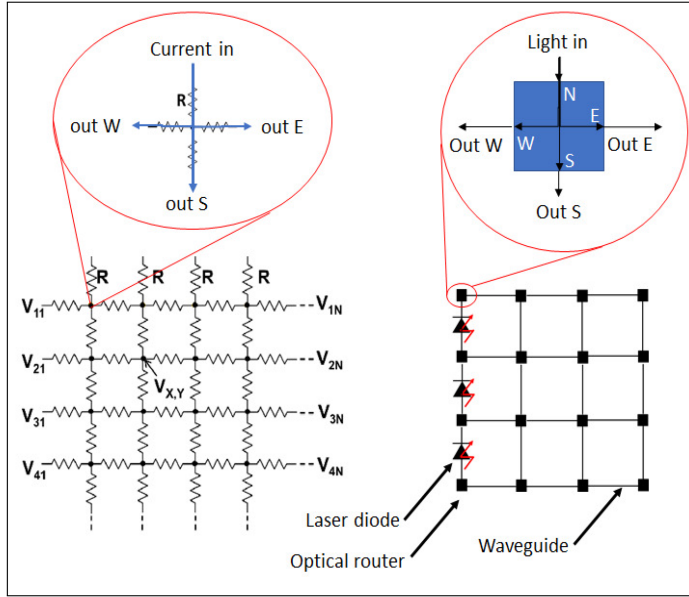


Fig. 2. Analogous to the resistive mesh (left), ROC (right) sums diffusion currents at the input port of each optical router, then attenuates at the router output ports. Metatronics enable the ideal ROC implementation, in terms of size and accuracy. However, photonics and plasmonics can yield approximate computing versions of ROC and are on a similar order to a resistive mesh in terms of power and size (right).

Table 1. Estimated ROC Performance vs. Resistive Mesh

	Si-Phot	Plasmonics	Metatronics	Resistive
Mesh Size	$4 \times 4 = 16$ ROEs			
Speed = $1/\Sigma(\text{delays})$	4.7 GHz	Prop+setup = 1 ps + 0.1 ns ≈ 10 GHz	Prop+setup = 1 ps + 0.1 ns ≈ 10 GHz	1.3 kHz
Relative Tuning Range (\tilde{E} R)	10 dB	6–10 dB	3–6 dB	n/a
Total Loss	–49 dB	–49 dB (ILEOM=5 dB)	–29 dB (ILEOM = 3 dB)	n/a
Total Power = Laser + EOM	$10 + 19 = 29$ mW	$10 + 0 = 10$ mW	$0.1 \text{ uW} + 0 = 0.1$ uW	25 mW
Power @ Rx for Laser = 10 dB	76 nA	76 nA	76 nA	n/a
Footprint	12 mm ²	0.12 mm ²	0.003 mm ²	180 mm ²

which covers related work. Section 4 describes ROC in detail. Sections 5 and 6 describe the evaluation methodology and provide a discussion of results. This is followed by future work and concluding remarks in Sections 7 and 8, respectively.

2 BACKGROUND

Many important physical systems can be modeled using PDEs [10]. However, the complexities associated with solving PDEs are at odds with the productivity. To continue the expected pace of

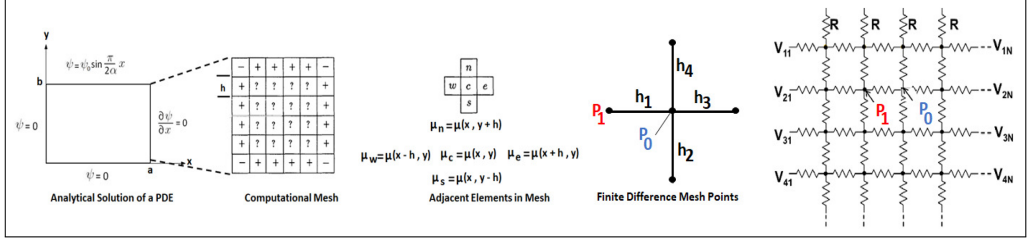


Fig. 3. Analytical solution of a PDE (left) solves over a surface using derivatives to represent continuous change in position. The numerical solution (center) decomposes the surface into a computational grid of elements, each of step size h . Difference equations are used to solve for adjacent elements. Finite difference mesh points (right) used to solve a PDE, and associated resistive mesh (far right). Note that adjacent elements, P_0 and P_1 , in the computational mesh are separated by a single resistor, of value h . To increase resolution, a larger grid of resistors can be fabricated.

innovation, techniques must be developed to accelerate physical system modeling and simulations. However, prior to elaborating on specific techniques, it is beneficial to understand which facets of PDE computation can benefit most from optimization.

2.1 Numerical Solutions for Partial Differential Equations

PDEs are equations that involve rates of change with respect to continuous variables, and they usually take the form

$$\nabla^2 f = g. \quad (1)$$

The difficulties in solving PDEs, when compared to ordinary differential equations, arise from a PDE's use of an infinite-dimensional configuration space. Infinite-dimensionality, while difficult to solve for, is necessary to adequately model complex systems, such as those commonly found in fluid dynamics. Additionally, it is important to note that a solution of a PDE is generally not unique; additional conditions must generally be specified on the boundary of the region where the solution is defined [38].

Due to the breadth of physical phenomena that PDEs can model, much attention has been invested in the development of efficient PDE solvers [13, 18–21]. However, none have found mainstream acceptance much like numerical methods.

As they can be tuned for predetermined levels of accuracy and executed on general-purpose computers, numerical methods are a common technique for evaluating PDEs. A numerical solution can only be found for the discrete form of a PDE, forcing a numerical solver to generate a computational grid of elements with uniform spacing h [46], as shown in Figure 3.

The computational grid is representative of the model being solved, with each node in the computational grid requiring an individual solution. For traditional computer architectures, an iterative algorithm is required to solve over the entire computational grid, with the number of iterations required to produce a solution being equal to n/k , where n is the number of nodes in the grid and k is the number of parallel operations that a computer can execute.

One particular mechanization of numerical methods, the difference equation, is encountered frequently when solving PDEs [22]. A difference equation equates 0 to a polynomial that is linear in the various iterates of a variable [39]. Usually the context is the evolution of some variable over time, with the current time period or discrete moment in time denoted as t , one period earlier denoted as $t - 1$, one period later as $t + 1$, and so on. For instance, an n th order linear difference

equation is one that can be written in terms of parameters a_i and b as

$$y_{t+n} = a_1 y_{t-1} + \dots + a_n y_{t-n} + b. \quad (2)$$

To convert the differential equation, shown as Equation (3), to a difference equation, the computational grid must be decomposed into clusters consisting of a central node and its four adjacent elements, shown in Figure 3:

$$\nabla \cdot \epsilon \nabla \vec{f} = 0. \quad (3)$$

By performing linear interpolation on Figure 3 and disregarding the higher-order terms we can simplify, which results in Equation (4):

$$\nabla^2 \vec{f} \approx \frac{1}{h^2} \left[\vec{f}(\vec{P}_1) + \vec{f}(\vec{P}_2) + \vec{f}(\vec{P}_3) + \vec{f}(\vec{P}_4) - 4(\vec{f}(\vec{P}_0)) \right] = 0. \quad (4)$$

If, in a system of time-dependent PDEs, all spatial derivatives are replaced by suitable difference approximations, then we obtain a system of ODEs in time [22]. This system of ODEs can, in turn, be solved to accuracy e with a complexity no worse than exponential in $\ln(e)$ for each ODE [23].

However, it should be noted that a reduced value of h results in a larger number of nodes, increasing the number of iterations required to compute a solution. For an $N \times N$ matrix, reducing the value of h by one half results in a computational mesh element increase of $O(N^2)$.

2.2 Analog Mesh Computers in Solving Difference Equations

Redshaw and Liebmann designed an apparatus that uses the relaxation technique to solve PDEs describing oscillatory flow [14] and heat transfer problems [13] using a resistive mesh. This device, called a resistance network analogue, is composed of resistors connected in a two-dimensional mesh configuration. Finite difference mesh points, as shown in Figure 3, characterize the stencil for a specific PDE, and are mapped to a resistive mesh for calculation. Solutions are read at the intersection of resistor terminals, or nodes.

The resistance network analogue takes advantage of distribution of h^2 throughout Equation (4), and replacement of $f(P_i)/h^2$ with current in Ohm's Law Equation (5):

$$I_n = \frac{V_n - V_0}{R_n}. \quad (5)$$

The right-hand side of Equation (4) now matches Kirchoff's law:

$$\sum_{n=1}^4 I_n = 0. \quad (6)$$

Setting all R_n to be equal, the current-based Equations (4) and (6) result in

$$\frac{1}{R} [(V_1 - V_0) + (V_2 - V_0) + (V_3 - V_0) + (V_4 - V_0)] = 0. \quad (7)$$

This best illustrates the resistance network analogue's ability to solve for a PDE using a voltage derived from current summation at each node. Monitoring of the voltage at each node yields a solution for each element in the computational mesh.

Liebmann showed the error introduced by voltage and current measurements to be negligible, thus reducing the factors that limit the accuracy of the resistance network analogue to simply the mesh size and tolerance of components comprising the mesh [13]. Principles that govern the relaxation technique state that the mesh size must be made so small that the replacement of the PDE by the finite difference equation is permissible, and that any error introduced by a mismatch in mesh size and resolution requirement can be corrected with a correction function [13]. Liebmann

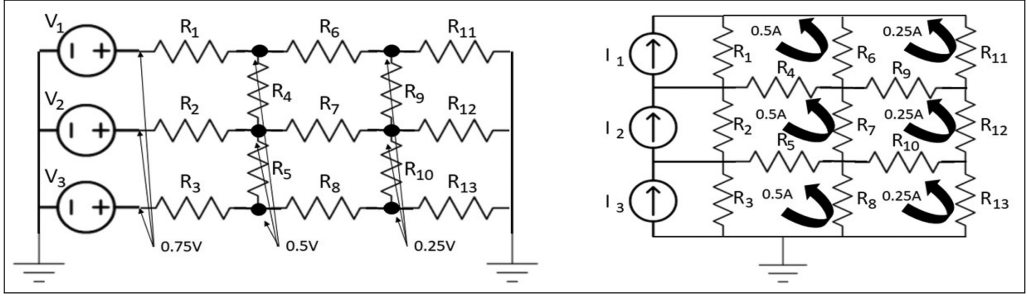


Fig. 4. Using Norton equivalence, a current-based dual can be constructed that is equivalent to a voltage-based analog mesh computer (or resistance network analogue). Readings can be taken in the dual by monitoring current in each loop, as opposed to voltages on specific nodes. This way, each current loop becomes equivalent to its corresponding node in the voltage-based mesh.

also showed that such a network of resistors contains averaging properties that minimize the error introduced by tolerances in individual resistor values [13].

Due to its structure, the time required for a resistive mesh to execute a calculation is equal to the time required for the mesh to reach a steady state, or Equation (8),

$$time = diameter * RC, \quad (8)$$

where *diameter* is the longest path through the mesh, and *RC* is the product of the lumped resistance and capacitance of each section of the mesh in between two nodes. This effectively eliminates the iterative component of solving for a computational mesh, allowing the resistive mesh, and other analog mesh computers like it, to outperform their traditional counterparts. Another advantage of this architecture is that the complexity of such a mesh is linear with respect to growth in mesh size.

2.3 Analog Mesh Computers Generalized

While the resistive network analogue is a classic example of an analog mesh computer, it is not the only implementation. Ohm's Law shows a linear relationship between voltage and current and Kirchoff's Law states that the input and output of current into a loop will always be equal [47]. Using these relationships, the architecture of the resistive network analogue can be generalized to any physical embodiment that supports multiplication and summation operations. In fact, using Equation (5), Equation (7) can be rewritten as Equation (9) by using branch currents in adjacent loops to the main loop, I_0 , where G is conductance:

$$\frac{1}{G}[(I_1 - I_0) + (I_2 - I_0) + (I_3 - I_0) + (I_4 - I_0)] = 0. \quad (9)$$

Using Norton equivalence, one can construct an analog mesh circuit dual that allows the summation of each branch current in the loop to replace a node voltage, shown in Figure 4. This enables the development of current-based analog mesh computers that are equivalent to voltage-based analog mesh computers such as the resistive network analogue. Current loops serve as replacements for their corresponding nodes in the voltage-based mesh, and can be monitored with appropriate circuitry.

One technology that supports weighted summation is photonics. Nanophotonic modulators have been shown to linearly attenuate light passing through them [43], and optical routers have the ability to combine the inputs from different channels and output their summation via a higher intensity output [44]. These two functions can be combined to achieve the photonic equivalent of

a current-based weighted summation [41]. This enables the development of photonic-based analog mesh computers, which rely on the ability to monitor the intensity of light passing through a detector.

3 RELATED WORK

Due to their importance in the modeling of physical systems, much attention has been paid to the efficient computation of PDEs. Numerical solutions are, by far, the most common computer-based solution, and have historically been made more efficient by adjusting initial values [48] and selectively adjusting numerical mesh resolution [34, 35]. These adjustments serve to reduce the number of iterations required in forming a solution.

Recent advancements in computer architectures have introduced parallel implementations of numerical solutions that decrease time-to-solution and energy consumed. Using parallel Arithmetic Logic Units (ALU) and ALUs with single-instruction-multiple-data (SIMD) support, memory accesses are reduced, and multiple execution streams are carried out simultaneously, reducing the iterations required for a solution [11, 12]. Similar results have been shown with other parallel computer architectures, such as graphics processing units (GPUs) [49, 50] and coarse grained reconfigurable arrays (CGRAs) [51].

Other advancements in traditional computer architectures and programming paradigms allow for a combination of mesh refinement and data layout techniques to increase the performance of a computational mesh solver. In Reference [57], the authors introduce the concept of an unstructured mesh, thereby enabling flexibility of data layout within memory. While the unstructured mesh can be viewed analogously to adaptive mesh refinement (AMR), by adhering to the principles of data locality during assignment of nearest-neighbor clusters, a computational mesh can be constructed that solves an approximation of the PDE to a suitable resolution while reducing access time to memory.

In Reference [58], a library of solvers is presented that uses LU factorization to efficiently solve linear matrices. The library contains kernels supporting both serial and multithreaded computation of sparse direct solvers, including support for distributed systems. While undeniably efficient, LU factorization is still an iterative process, which constrains the upper bound on performance improvement one can expect.

Narayana introduced an analytical method of solving PDEs with an electrolytic tank [18]. While indeed reducing the computational intensity of a PDE solution, the accuracy of analytical solutions has historically been called into question [13]. Additionally, the electrolytic tank suffers from an inability to solve for irregularly shaped surfaces.

Liebmann introduced a resistor-based analog mesh computer [13], called a resistance network analogue, capable of efficiently computing the solution to PDEs associated with describing heat transfer within a solid. He later improved upon this [17] to support the creation of submeshes, enabling the computation of heat transfer through irregularly shaped surfaces. However, the static nature of VLSI devices, coupled with a hardware-centric mapping of PDEs, introduces problems when programmability is desired (e.g., when changing either the resolution requirement of the solution or changing the resistance of the surface). In general, the dynamic nature of scientific programming is fundamentally at odds with any static mapping of a problem to specific hardware.

Redshaw introduced an analog mesh computer capable of efficiently solving oscillatory flow problems in aeronautical engineering [14]. Similar to the resistance network analogue proposed by Liebmann, Redshaw's computer is based on a resistive mesh and includes all of the drawbacks associated with Liebmann's device.

Yin [45] proposed a much larger resistor-based analog mesh computer for resistivity-log interpretation for formation evaluation in the matching of the reconstructed deep-reading resistivity

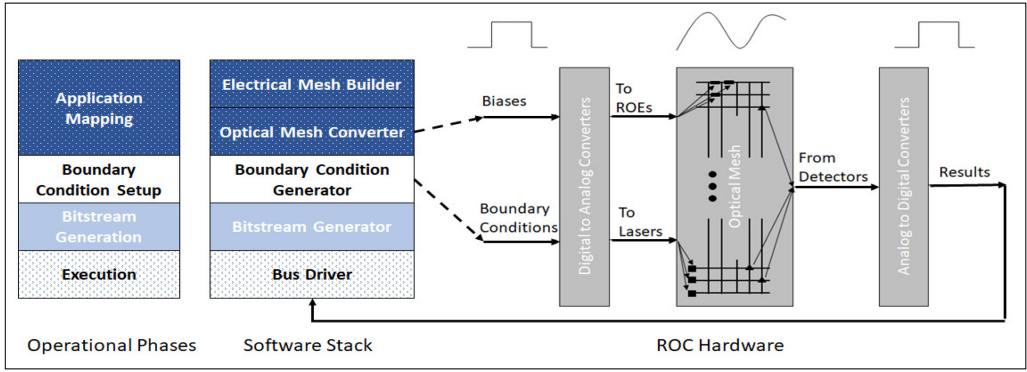


Fig. 5. Hardware and software stack required for ROC, along with simulation flow. ROC is configured by mapping a physical model to a hardware configuration. ROE biases are calculated using a simulation of the analog mesh and downloaded to the ROC hardware. ROC executions are processed by biasing perimeter laser diodes that represent boundary conditions.

logs with the field log curves. However, this analog mesh computer suffers from the same drawbacks as the aforementioned devices.

Ramirez-Angulo and DeYong recognized the problem associated with static mapping introduced by the Resistive Network Analogue, and improved upon the device with the introduction of resistor and capacitor-based analog mesh computers fabricated with CMOS devices in Reference [15]. The biasing of CMOS transistors enables the programming of the computer to solve a variety of PDEs. However, transistors being operated in the subthreshold region require large devices, as they are more easily matched. This is at odds with modern digital designs, which are developed with minimum-size devices.

Thus, it is apparent that a PDE computation engine is needed that retains the programmability of the Ramirez-Angulo architecture, but can be easily integrated into future computer systems. Doing so optically is the objective of this work, which will also enable harnessing the many benefits from the optical world such as ultra-low power.

4 THE RECONFIGURABLE OPTICAL COMPUTER

As shown in Figure 5, ROC is an analog mesh computation engine comprising both the analog mesh computer hardware and a software stack responsible for hardware configuration. A classical computer is interfaced to ROC enabling its configuration and solution readout. Configuration is done via a bit-stream, and ROC output buffers are connected directly to the system bus, where they can be directly accessed.

Unlike other analog mesh computers, to achieve weighted summation, ROC uses the transmission of the optical power associated with the electromagnetic radiation that travels through the photonic circuit as described by Equation (9). To this end, the portion of the analog mesh of the ROC comprises reconfigurable optical elements (ROE) connected into a two-dimensional grid with laser diodes for setting boundary conditions, and support electronics for laser-diode/ROE biasing and readout of results. A metamaterial-based ROE is biased when presented with a voltage, which changes its attenuation of the optical power flowing through the device.

Weighting of the input power of the electromagnetic radiation is achieved through signal attenuation, shown in Figure 6, and summation is done through the superpositions of the electromagnetic waves from multiple waveguides by means of optical interference. This is synonymous, up to a certain extent and degree of accuracy, to other analog computation architectures [13, 24].

Table 2. Comparison of Pertinent Phenomena in Electronics and Photonics

Parameters	Electronics	Eq/Units	Photonics	Eq/Units
Signal “flow”	Current	$I = V/R_{eff}$	Diffusion Current	dD/dt
Splitting Ratio	R_{eff} ratio	$R_N : R_S : R_E : R_W$	S-Parameters	dB
Node Potential	Potential	$V = I_{net}R_{eff} (V)$	$\delta Power$	$\delta \int E (W)$
Signal at node	Net current	ΣI_i	Net power	ΣP_i

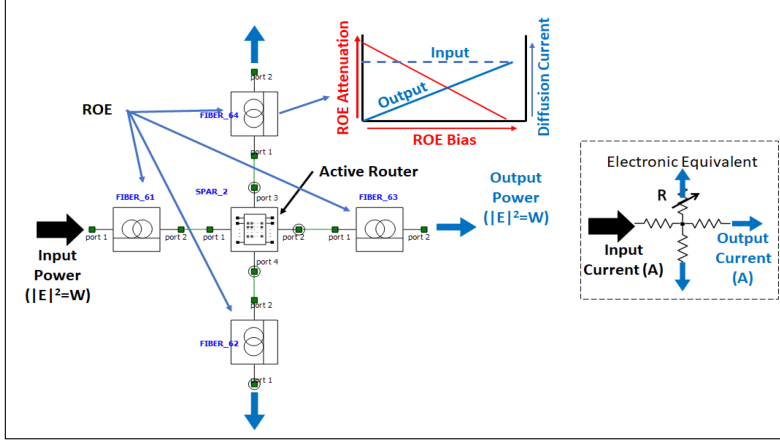


Fig. 6. The active router uses diffusion current splitting to enable the same current splitting seen at the intersection of resistors. ROEs are biased such that input current (dashed line) is attenuated as it passes through the ROE. This allows the ROC to effectively mimic the resistor networks introduced by Liebmann [13] and Redshaw [14].

In this paradigm, the laser sources are additive at each node like current sources. The main phenomenological discrepancy between an optical counterpart of a resistive network relies in its dimension with respect to the operating wavelength of the electromagnetic radiation. A purely resistive network operating a 3 GHz can be modeled as a lumped circuit if its dimension is within a decimeter. However a photonic circuit is a distributed network where the physical length of circuit is comparable or larger than the wavelength. In this case, many phenomena such as reflections, propagations losses, and impedance mismatches need to be accounted for, which could eventually hinder the accuracy of our solution. Nevertheless, having a solver able to approach the actual solution of a PDE would be of fundamental use, enabling the initial solver based on current technology. Additionally, using readily available technological approaches one could enable the signal to propagate through the network in a similar fashion to electrons flowing through an electrical network [65]. Biases for the ROEs and lasers are calculated by generating the Norton equivalent circuit for voltage biases on the perimeter of the resistive mesh. This circuit conversion replaces voltage sources with current sources, which are a better representation to the ROC perimeter laser diodes that focus out light at a specific intensity. The conversion from electric current to optical intensity enables the creation of another dual circuit, described by Equation (10):

$$\sum_{n=1}^4 P_n = 0. \quad (10)$$

Diffusion currents can be read by a photodetector after the mesh reaches its equilibrium state. Behavior described by Equation (10) enables the optical mesh to operate in the same way as the electrical mesh proposed by Liebmann [13] and Redshaw [14]. This same idea can be extended to other implementations of optical circuits where optical power, thus light intensity, is attenuated and summed in the same manner.

The ability to bias an ROE using a digital-to-analog converter (DAC) enables reprogramming of the analog mesh for low power operation or to create irregularly shaped surfaces. This programmability supported by ROC allows its static hardware to execute a variety of PDE-based problems, making it ideal for integration into a VLSI architecture. Moreover, metamaterial-based ROEs are expected to be much smaller in area compared to conventional optical devices [61, 62].

4.1 Arithmetic Operation of Optical ROC

The arithmetic operation of optical ROC is built around the constructive and destructive interference of light at crossing nodes that have four inputs/outputs. The optical splitter directs a third of the input light into the three waveguides attached to the node. However, due to the directionality of light, we can now observe the first difference between the electrical mesh design of ROC and the Optical Design of ROC. In our electrical cross points of four inputs and four outputs, we have Kirchhoff law addition of current. Where the direction of current can change at different times in the simulation as the mesh settles to an equilibrium state. In our optical ROC, the waveguide geometry is engineered so that one input feeds equally to three outputs and with a percentage of the light backscattered into the original waveguide that the light originated from. This results in the network taking on a tree structure with preset boundary conditions where the top of the mesh has input light and the three surrounding edges of the mesh have optical absorption in the form of photodetectors. This architecture is our first iteration and is not equivalent to an electrical mesh point that is not geometry dependent in the way that optics waveguides are.

4.2 Concept of Operation

As ROC is a loosely coupled coprocessor, it simply accepts configuration and boundary-condition data from a classical computer as input, computes a solution, and then presents the solution to the classical computer. A single ROC transaction comprises three stages, shown in Figure 5: application mapping, boundary-condition setup, and execution.

Each PDE, representing a physical model and resolution requirement, is mapped to a specific hardware configuration. A simulation of the system is then run on a classical computer and used to determine the bias required for each ROE. A bitstream, containing the ROE biases, is then generated and sent to the ROC hardware. Thereafter, each ROC execution, representing changing boundary conditions, only requires the changing of perimeter biases through a subsequent bitstream. This flow minimizes the number of time-consuming system simulations and only requires minimal changes at the mesh boundary for each simulation of a physical model, enabling support of batch operations, as shown in Figure 7.

4.2.1 Application Mapping. In the application mapping stage, a physical model of the system, along with resolution requirements, are mapped to a specific analog mesh configuration. The shape and resistance of the physical model's surface can be changed through individual ROE settings. ROE attenuation settings are linear in nature; at its lowest setting, a ROE exhibits minimal attenuation of light and acts as a pass-through coupler, and at its highest setting, the ROE attenuates all light that attempts to pass through. At a system-scale, this allows the analog mesh's shape to be changed by effectively disconnecting portions of the mesh using the high-attenuation setting

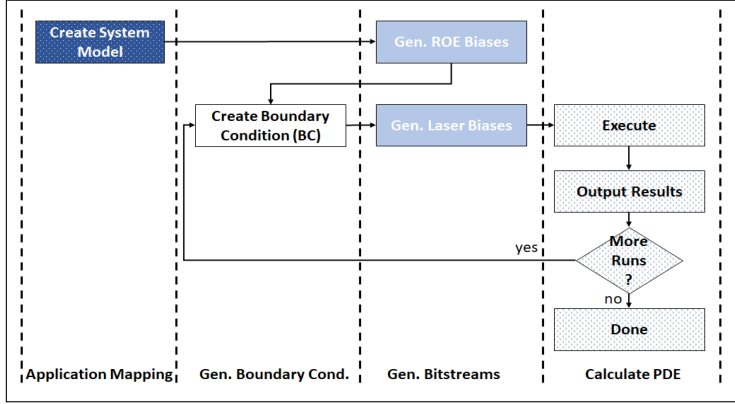


Fig. 7. Operational flow of ROC. A single, physical model is created, followed by generation of a bitstream that configures the ROEs in the mesh. Then, a boundary condition is created, followed by generation of a bitstream used to configure the lasers at the mesh perimeter. Last, ROC executes the computation of the PDE. This supports a pipelined execution, where ROC can compute and send results to memory while the next boundary condition is being created.

of the ROEs. An $N \times N$ analog mesh contains N^2 ROEs, which must be set during the application mapping stage.

This configuration can then be bundled with other pre-built configurations and ROC can be automatically partitioned to efficiently execute all calculations and provide a virtualized environment for multiple users. Alternatively, a single user's large application can use virtualization to execute over the ROC in multiple passes.

4.2.2 Boundary Condition Setup. Setup of boundary conditions includes the programming of analog mesh perimeter biases, or current sources. This is done by setting the intensity of lasers located along the analog mesh perimeter. Much like an ROE, each laser's intensity setting is linear, with the lowest setting being off. An $N \times N$ analog mesh contains $4N$ lasers, which must be set during the setup of boundary conditions with each node then being routed to its own output port with minimal reflection [62]. Compared to the standard optical power splitting mechanism, the photonic crystal design could further reduce the on-chip footprint from a few hundreds of μm^2 down to only $25 \mu\text{m}^2$ (compared to 230 lookup tables and 150 registers for an equivalent-function digital circuit synthesized for an FPGA). And in terms of data readouts, Y-junction splitters with grating couplers are considered as the initial step of prototyping, but could be replaced by integrated photodetectors for better signal-to-noise ratio (SNR) and denser packaging.

4.2.3 Creation of Bitstreams. Both application mapping and boundary-condition setup require configuration of analog mesh hardware. Much like current traditional reconfigurable logic, configuration of ROC is done via a bitstream [52]. Configuration words for each ROE and laser are generated with a traditional computer and concatenated into a single bitstream, which is then sent to ROC. Bitstreams for initial runs, which contain both application mapping and boundary conditions, of a $N \times N$ analog mesh have a complexity of $O(N^2)$, while bitstreams for subsequent runs are of complexity $O(N)$.

4.2.4 ROC Execution. Program execution of ROC follows the same model as many application-specific computing systems. After ROC has been configured, its inputs and biases, which represent the variables of the PDE, must be set and ROC immediately executes its calculation and results are

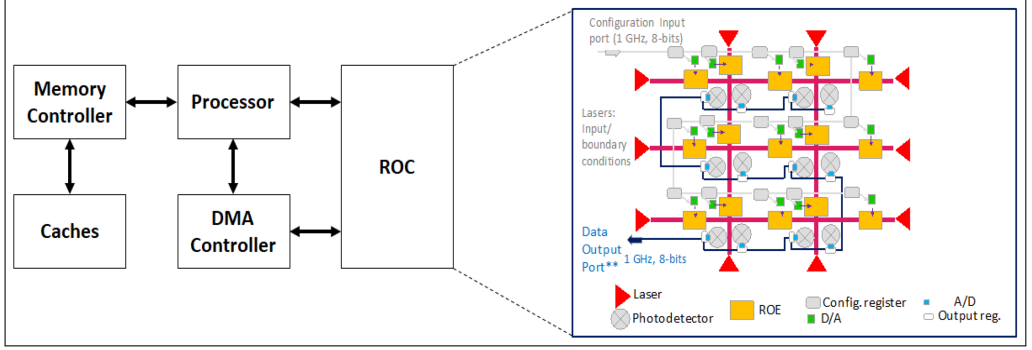


Fig. 8. Block diagram of the Reconfigurable Optical Computer loosely coupled into a classical computer system. The ROC is accessible to the microprocessor, as well as other components that support direct memory access (DMA). The configuration port is used to configure the ROC to solve a specific partial differential equation, and input lasers set the boundary conditions. The solution to the PDE is read out from each node by converting the power level seen at the node to a digital value.

made available to an output buffer, which uses a high-speed serial bus to communicate with the rest of the system.

4.3 Hardware

Shown in Figure 8, ROC hardware comprises a nano-optical network responsible for computation, and electronic support circuitry responsible for biasing the ROEs, laser diodes and photodetectors, and readout of the solution. This architecture allows for scaling of the low-energy optical components, while minimizing the higher-energy electrical components.

The time required to compute a solution is bounded by

$$time_{meshcalc} = time_{setup} + time_{settle} + time_{readout}, \quad (11)$$

where

$$time_{settle} = diameter * c, \quad (12)$$

where c denotes the speed of light through the node.

4.3.1 Optical Network. To prove the viability of ROC, we designed a non-metamaterial version, based on passive components (such as routers and waveguides), which uses light intensity as the enabler for analog computation. While this deviation was taken initially, in part, due to historical success with passive components [44, 59], it has proven to be more helpful than originally anticipated during the prototyping effort through its direct support of a wide variety of test cases. It differs from the metamaterial-based ROC design described in Figure 6, which requires active photonic components and uses diffusion currents for computation. Shown in Figure 10, this design is based on a static optical network with a reconfigurable boundary, supporting multiple PDE configurations over a static surface.

Shown in Figure 9, the optical mesh comprises a two-dimensional array of optical four-way equal-splitting routers connected with waveguides that provide signal attenuation, much like resistors in the electrical mesh. Moreover, each router is connected to four individual laser diodes with Y-branch splitters, which provide source input to the node at each direction. On the other side of the splitter, a photodetector is used to monitor the light intensity passing each node at each direction. Note that since Y-branch splitters are used, the light intensity loss that is caused by the coupling efficiency must be considered and compensated for after the solution is generated.

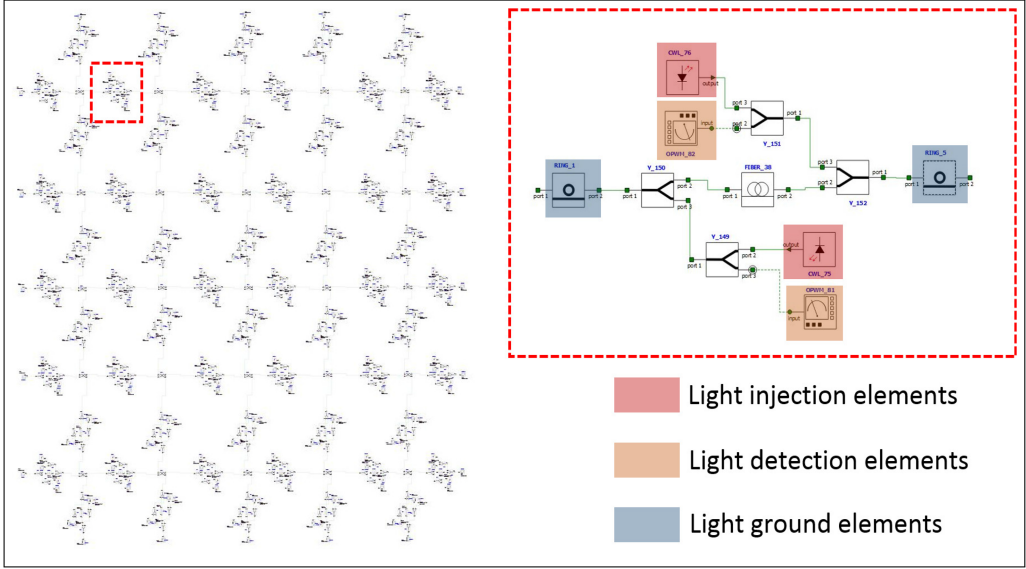


Fig. 9. Schematic of the optical network of the ROC. The insets show the light injection and the light ground details.

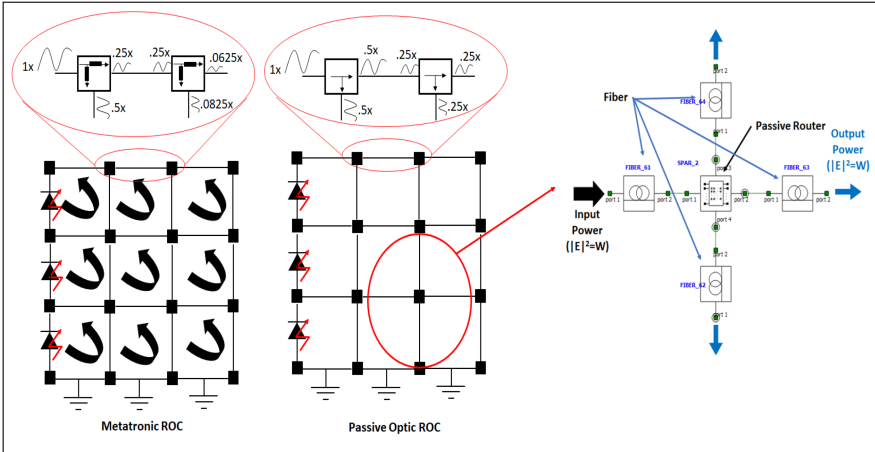


Fig. 10. Passive optics-based ROC design compared with an metamaterial-based ROC design. The metamaterial-based design can be reconfigured using ROEs such that the diffusion currents precisely model the currents in the Norton-equivalent of the resistor-based mesh. The passive optics-based design is limited by preset currents through the router output ports and attenuation through optical fibers, which sets an upper bound on its accuracy. However, it is capable of modeling the resistor-based mesh directly, with no need to convert between Thevenin and Norton equivalent circuits.

In an electrical circuit, an electrical ground gives a reference point of potential that absorbs an infinite amount of current. To represent it in optics, all four micro ring resonators (MRRs) around the 2×2 router must be tuned to the ON state (i.e., the same resonance as the laser). Thus, instead of passing through the router, the optical signal will be coupled into the ring cavities, creating a node with large optical loss that effectively acts like an electrical ground. This greatly simplifies

the fabrication process of our ROC model, since it only requires one type of 2×2 router, which can be made completely passive with equal splitting ratios (i.e., $1/3$ for each output port) and is capable of switching between a regular node and an optical ground.

The potential advantage of reconfigurable attenuation and splitting, brought about by the use of active photonic components as ROEs, is effectively nullified with a ROC based on passive optical components. Consequently, the ROC use-model from Figure 5 is simplified, as the initial ROC simulation is no longer required in the calculation of branch splitting ratios. Additionally, the need for DACs is eliminated in the mesh, resulting in savings in both power and hardware footprint.

Laser power for ROC is programmable, and the power for individual laser diodes is set using a bitstream with the maximum operating frequency limited by the settling time of the entire ROC mesh. Laser configuration registers are connected in a serial chain and accessed in a Joint Test Access Group (JTAG)-like fashion [64], supporting serial communications and simultaneous configuration. Note, that as the network scales up, the required laser power rises exponentially; however, the laser power of a 16×16 ROC network is still far below the nonlinearity region (75 mW) based on results from Reference [60]. For networks larger than 16×16 , the nonlinearity effect could be mitigated by activating other laser diodes along the source-sink path.

It is worth mentioning that the actual splitting ratios in the equivalent electrical mesh are based on the potential as seen from each node; thus, the final result generated by the ROC optical mesh does not match the electrical mesh, but can be corrected by tuning the optical losses between each node. The amount of loss needed for ideal compensation should be equal to the power difference caused by the difference between the actual splitting ratio and the equal splitting ratio. Ideally, the approach taken for router development uses reconfigurable routers, as demonstrated in Reference [44]. Each channel within the router is a separate light path, reconfigurable through the changing of bias voltages, which, in turn, changes the coupling efficiency of each switch. Attenuation settings for each channel are used to set splitting ratios for north, south, east and west channels. This enables the optical mesh to be reconfigured to a multitude of shapes and resistances.

As in other implementations of analog mesh computers, component matching influences the results computed by the ROC. In Reference [13], Liebmman showed that since resistors are shot noise driven, and that shot noise is Gaussian distributed, noise does not scale with dimension, since the average is still zero. The same is true for ROC, because its photodetectors are also shot noise limited. For the system considered here, with power threshold limits setting up the minimum power level of the photodetector, Gaussian noises are expected. Hence, noise is averaged out upon scaling.

4.3.2 Alternative Optical Networks. In the ROC design, as presented, the structure of an optical router can take any of three forms, shown in Figure 11. The first router could be developed with active hybrid photonic-plasmonic (HPP) switches, as has been demonstrated previously [44]. The coupling efficiency of each switch can be electrically tuned from 0% to 100% by actively tuning the electrical bias voltage of each switch; thus, the final signal splitting ratios can be made equal through setting the bias voltages at the four switches. The second form uses active MRRs to achieve an equal splitting with similar coupling efficiency using a heat-tuning approach [55]. The third form is a passive router that uses photonic crystals in the waveguide crossing. Defects created in the center of the crossing enable the spreading of light mode evenly into thirds, with each mode then being routed to its own output port with minimal reflection [56].

Note that the alternative active-component router designs described above are tuned to create a passive router that matches the static network-based ROC, as presented. Conversely, these active components can be used to create active routers, which are then incorporated into a fully

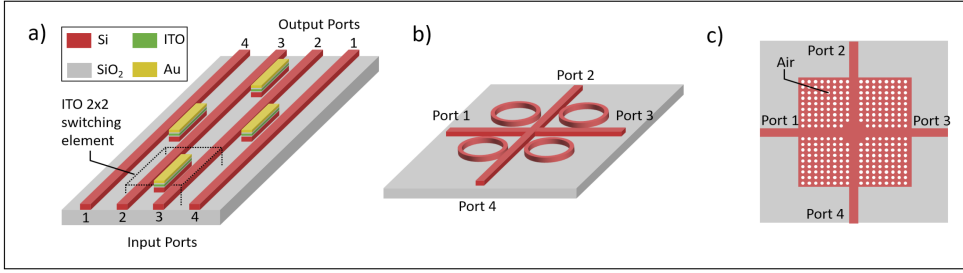


Fig. 11. Schematic of the optical passive equal splitting router models. (a) A hybrid photonic-plasmonic router design with four ITO-based MOS 2×2 switches. (b) A micro-ring-based photonic router. (c) A photonic crystal waveguide crossing with defects in the center.

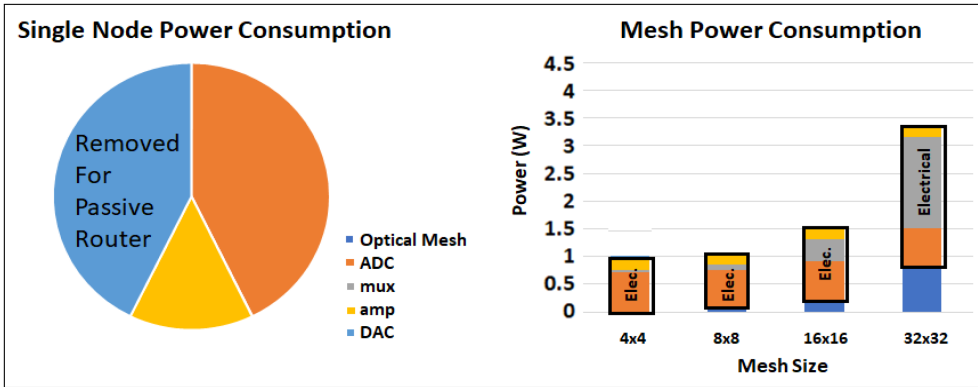


Fig. 12. ROC hardware power draw. Percentage of energy consumption for components comprising a single node (left) indicates that amplification and analog-to-digital conversion consumes the majority of energy and should be minimized. Energy consumption of resistive mesh computers of different dimensions with minimized readout circuitry (right) shows that the consumption by low-power multiplexors overtakes that of the transimpedance amplifiers by the third doubling of mesh dimensions. Note that the readout configurations chosen are more serial in nature and influence the readout time, hence performance, of the mesh.

reconfigurable optical network fabric, enabling direct calculation of PDEs over heterogeneous surfaces by ROC.

4.3.3 Electronic Circuitry. As ROC is an analog computer, it requires specific support circuitry to interface with a traditional digital computer, as shown in Figure 12. The traditional computer sets digital biases for the ROEs and reads out digital data from the analog mesh.

To enable a digital computer to set ROE biases, DACs are placed in the addressable space of the computer. Due to the low power draw of ROEs, a single DAC can be shared among multiple ROEs to reduce the overall system power. Additional reductions can be gained through the replacement of commercial low-power DACs with custom, low-power DACs with a reduced dynamic range. The sensitivity of the ROE allows an 8-bit DAC to be used without creating additional errors due to its differential nonlinearity (DNL).

Readout of the optical array is accomplished by converting the laser intensity to a digital value. This is a multi-stage process, in which a photodiode is used to convert light intensity to a current, followed by a current-to-voltage conversion using an operational transimpedance amplifier.

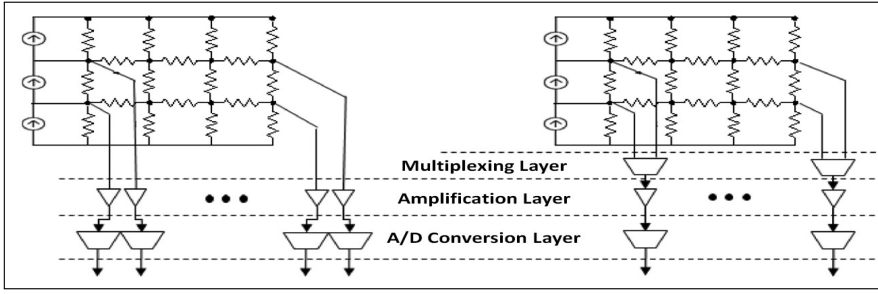


Fig. 13. ROC readout architectures. Fully parallel readout architecture (left) allows all nodes values to be read out in parallel, giving highest theoretical throughput. Low-power readout architecture (right) serializes the readout of nodes in the mesh, which increases the time required to access the solution.

This voltage can then be quantized using an analog to digital converter (ADC) and the individual outputs registered.

Energy consumption of the mesh is the sum of energy required for mesh control circuitry, comprising input bias circuitry and readout circuitry, and the mesh components, as shown in Figure 12. As the solution of the PDE is the sum of currents taken at each node, a brute-force approach requires an operational transimpedance amplifier at the output of each node, followed by an analog-to-digital converter (ADC). However, when considering the energy consumption of the individual readout circuitry components [26, 28], it becomes obvious that this approach will not support scaling, and that other architectures should be considered. As the mesh computer reaches a steady state upon completion of a calculation, the mesh can be considered to have a memory, thus node currents are not required to be read simultaneously. Relief of the simultaneous readout requirement allows for serial readout, reducing the number of ADCs required by the mesh, with an expected reduction in performance (Figure 13), described in Equation (11). The study of readout circuitry and its effect on mesh performance was studied in detail by Liebmann in Reference [36].

Additional energy savings can be gained by relaxing the requirement for an amplifier at every current-summing node, as seen in Figure 13. The sharing of operational transimpedance amplifiers by multiple nodes was proposed in Reference [24], and reduces energy consumption further through a reduction in amplifiers and their replacement with comparatively low-power analog multiplexers [26, 27]. This architecture has the advantage of slowing the growth of the high-power components, allowing the low-power components of the mesh to grow fast and eventually dominate energy consumption, as seen in Figure 14.

4.4 Software

Due to its reconfigurable nature, the ROC architecture has a great deal of similarity to that of a Field Programmable Gate Array (FPGA). Shown in Figure 15, the ROC software stack runs on a classical computer and is therefore analogous to the tools developed in support of the FPGA design flow. The ROC software stack enables a user to specify the PDE, boundary conditions and resolution requirements. This design flow results in a bitstream that is then downloaded to the ROC hardware, which uses the bitstream to configure biases for the ROEs and perimeter laser diodes. Results from ROC processing are then uploaded to the classical computer.

The software stack was designed to enable the natural progression of a traditional physics simulation, where a physical model is described and then used for multiple simulations under different boundary conditions.

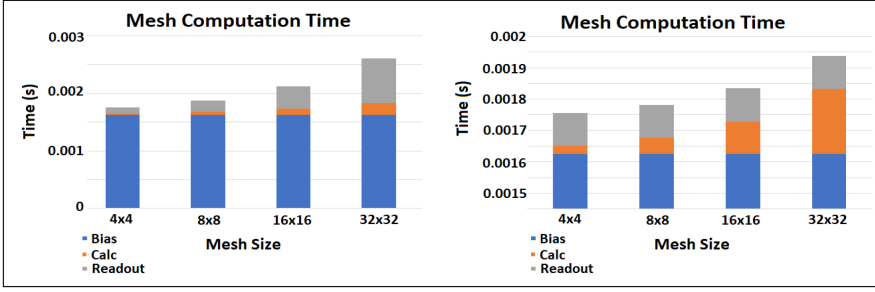


Fig. 14. ROC readout times. Low-power readout architecture (left) results in increased execution time with increasing mesh size, due to the serial nature of readout, which always dominates calculation time. High-power readout architecture (right) results in constant readout time, allowing the calculation time to dominate for large meshes.

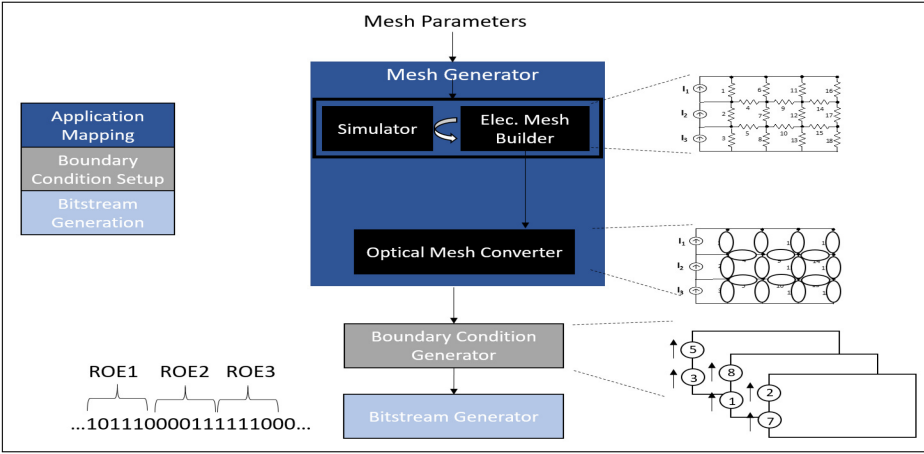


Fig. 15. Software stack and development flow required for ROC. An analog mesh is generated to solve a specific PDE. The mesh is simulated and current splits are calculated for each node. These splits are used to set the attenuations of the optical router ports.

Laser and MRR biases are calculated for different simulation cases for the physical model to be simulated. These biases are then sent to a bitstream generator, which is responsible for converting the biases into bitstreams for each laser and MRR within the ROC optical mesh fabric. The bitstream is then downloaded to the ROC hardware for mesh configuration over a JTAG-like bus.

Innovative architectures like ROC require fundamental changes in the software stack as described above. In our studies thus far, we have built a software simulation interface that mimics the full software stack to the extent possible. Our simulation library requires the user to create a PDE object that represents the problem to be solved. This is done by specifying the size of the domain and the boundary conditions. Then, this PDE is translated into an abstract mesh that consists of nodes and links between them. The size of the mesh created in this step is equal to the size of the simulated ROC mesh. Finally, this abstract mesh is translated into an input script for the optical network simulation software Lumerical Interconnect. We believe that our current software simulation modules can eventually be extended into a fully functional software stack that can address the user interface and execution environment for such new optoelectronic computing systems. Given

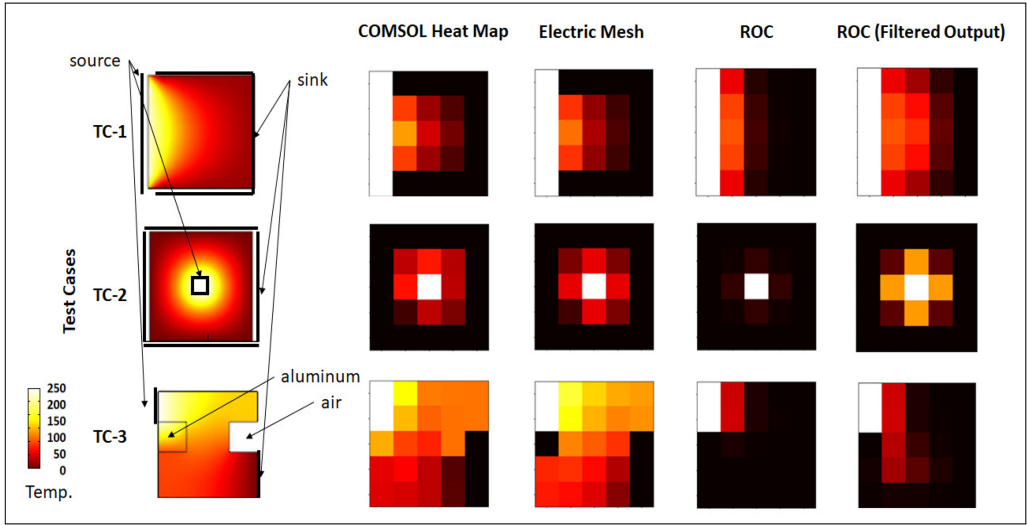


Fig. 16. Heat transfer problem configurations used in experiments. The nonlinear case (left) has source on the entirety of left side, and sink on the remainder of the perimeter. Symmetric case (right) has a center source with a perimeter sink. Numerical solutions for 5×5 meshes computed using COMSOL were compared with SPICE simulations of the resistance network analogue and Interconnect simulations of ROC. Filtered output of ROC applies a scale to the output for readability. Note that the calculations from ROC trend similarly to other analog mesh computers.

the analog nature of ROC there is an opportunity to develop productive programming interfaces that are more meaningful to the domain scientists.

5 EVALUATION METHODOLOGY

Two facets of ROC were evaluated: accuracy of results and time-to-solution. This was done using three different heat transfer configurations, as shown in Figure 16, each selected to highlight ROC performance in relevant scenarios. The heat transfer configurations selected include: (TC-1) the nonlinear case, a single-source/single-sink on the perimeter of a homogeneous plane, (TC-2) the symmetric case, a single-source in the center of a homogeneous plane with a sink around the entirety of the surface, and (TC-3) the asymmetric case consisting of a heterogeneous plane with an irregular shape. We used COMSOL, industry-standard software for scientific simulations, as a reference point for accuracy, and also included a resistive mesh for comparison. Time-to-solution evaluations compare ROC to both uniprocessor and GPU-based architectures.

We have implemented a set of scripts that takes, as input, the continuous heat transfer problem with a set of boundary conditions and translates this problem into a discrete problem of appropriate size based on the size of the simulated network (optical or electrical). Then, the script builds an abstract mesh network to represent the discretized problem and translates the abstract network represented to different flavors of SPICE: (1) a version that can be imported to Lumerical Interconnect, optical circuit simulation software, and (2) a version that can be imported to ngspice, an electric circuit simulation program. The script runs said applications and parses their output to populate values in the abstract mesh, which, in turn, can be queried for different characteristics or visualized for inspection. The results we obtained through this workflow were compared to the results from COMSOL, which we regarded as the ground truth.

For a fair comparison of accuracy results, equal size meshes were generated for COMSOL, SPICE, and Interconnect. Meshes of different sizes (4×4 up to 32×32) were compared to show effects of mesh size on the resolution of the solution generated.

For time-to-solution evaluation, the execution time of ROC was compared with COMSOL running on both an Intel Xeon E5603 at 1.6 GHz and an Nvidia V100 GPU for equivalently sized computational meshes with 32-bit precision. The time-to-solution for ROC was estimated as

$$t_{\text{solution}} = t_{\text{bias}} + t_{\text{settling}_{\text{DAC}}} + t_{\text{settling}_{\text{mesh}}} + t_{\text{settling}_{\text{ADC}}}, \quad (13)$$

where DAC settling time and ADC settling time were estimated from current literature [26, 28]. The size of bias data required for each $N \times N$ mesh is dependent on the size of the mesh and calculated as the sum of mesh router biases and laser diode biases (shown in Equation (14)):

$$\text{size}_{\text{data}} = N^2 \times \text{size}_{\text{word}_{\text{DAC}}} + 4N \times \text{size}_{\text{word}_{\text{DAC}}}. \quad (14)$$

The ROC biases were written in a serial chain running at 100 MHz and ROC optical mesh settling times were estimated for large meshes by simulating small meshes (3×3 , 4×4 , and 5×5) using Interconnect, and then extrapolating for larger meshes. The small-mesh simulation runs in Interconnect confirmed that mesh settling time was, indeed, a function of the diameter of the mesh and time-of-flight of the optical signal, thereby supporting our estimates for meshes too large to efficiently simulate. Large mesh dimensions were bounded by current research in the domain [54].

6 RESULTS

Simulation results of the resistive mesh and ROC are shown in Figure 16. Note that the solutions computed by ROC trend with both COMSOL and the resistive mesh, an expected behavior for approximate-computers of this type. While the heat maps computed by ROC do not match the heat maps produced by the resistive mesh, it is worth noting that a physical instantiation of ROC can be reprogrammed to take the form of any surface, while a new resistive mesh would have to be fabricated for each surface under evaluation.

Accuracy of solutions computed by ROC, shown in Figure 17, was calculated for a single row by taking the difference between corresponding nodes in ROC computations and COMSOL computations, which were regarded as the ground truth. Results show that the output of ROC follows the same trend as other approximate computers. For testcase 1, the average accuracy of ROC was calculated as 91.7%, which is well within the thresholds allowable (50%) for approximate PDE solutions in the signal processing, fluid dynamics and numerical electromagnetics [7, 33, 34]. The strongest deviation of ROC output from COMSOL, and the resistive mesh, occurs when highly linear results are expected, as illustrated by the symmetric test case, which produced an average accuracy of 72%. This is to be expected with an optical approximate-computing system, as the coupling of Y-branch splitters has a known -3 dB efficiency. Differences in accuracy for multiple problem configurations is acceptable in the approximate computing domain, as approximable variables and operations are identified in the system, and thresholds assigned based on impact to the overall calculation [7, 9, 35]. Moreover, ROC accuracy can be further improved by adjusting the optical loss between each pair of nodes. For certain heatsink conditions, the equal splitting ratio of the optical router yields higher discrepancies than the actual splittings generated by COMSOL, and these differences, in terms of power, result in a faster or slower drop of the optical signals. However, after an optimized optical loss of the wire is found, the error between the COMSOL results and ROC results can be minimized.

Time-to-solution for ROC is shown in Figure 18. Simulation of small meshes confirm that the settling time of the ROC mesh is governed by the time-of-flight of photons through the waveguide. ROC consistently outperforms the Xeon E5603 in time-to-solution by multiple orders of magnitude

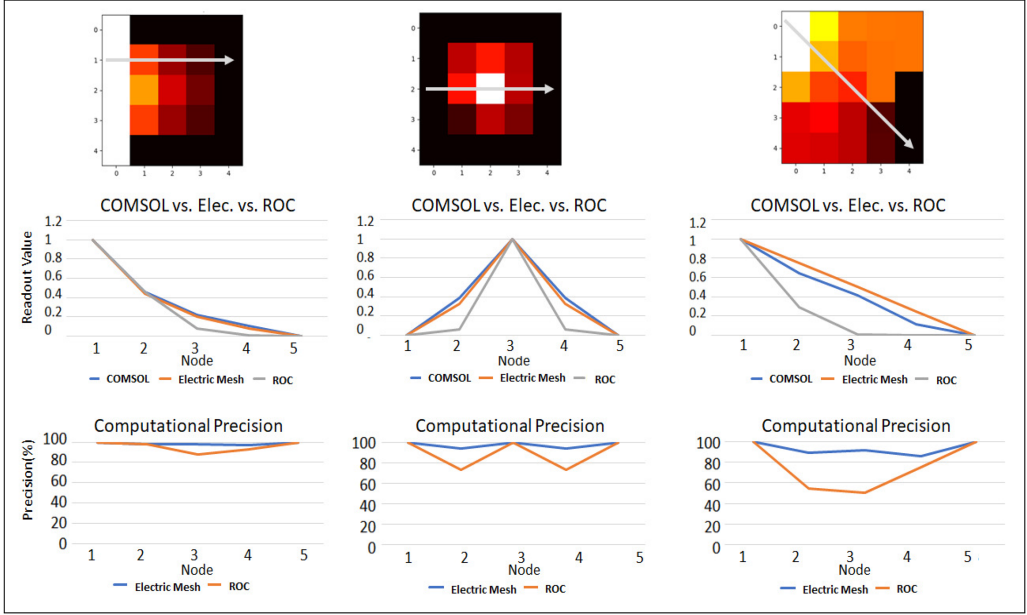


Fig. 17. Readouts for specific nodes in analog mesh computers (top). Error was calculated by comparing corresponding nodes of each mesh (middle). Accuracy is shown as a function of the difference between analog mesh computers and COMSOL (bottom).

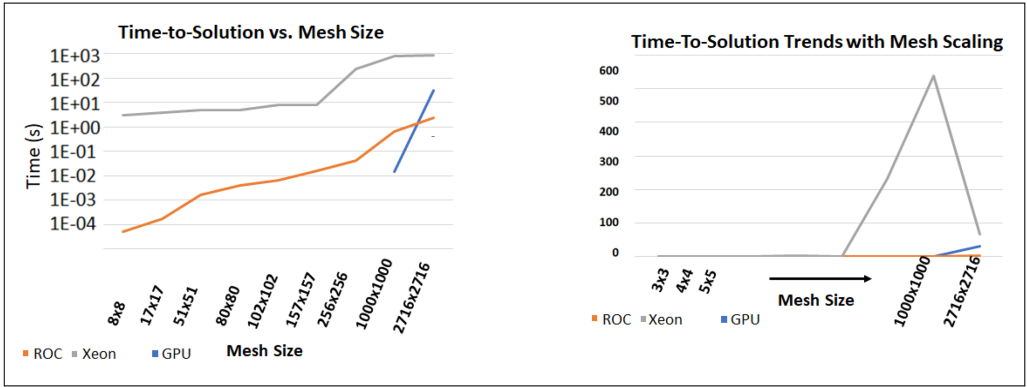


Fig. 18. Run times scale with mesh size for ROC, V100 and Xeon E5603 (left). ROC consistently outperforms the Xeon E5603 and outperforms the V100, designed for massive parallelization, for meshes larger than 1,000 \times 1,000. Trends in time-to-solution as the mesh scales in size (right). Note the point in inflection at the 256 \times 256 mesh for Xeon E5603 and 1,000 \times 1,000 for the V100, as ROC stays constant.

due to its elimination of the iterative component of the computation. The V100 outperforms ROC for a 1,000 \times 1,000 mesh, however, ROC regains its edge as the mesh becomes larger. It should be noted that while reduced-precision datatypes are typically used to enhance the performance of GPUs by enabling higher parallelism, this technique cannot be used to surpass ROC performance, since individual submesh results are influenced by results from preceeding submeshes in

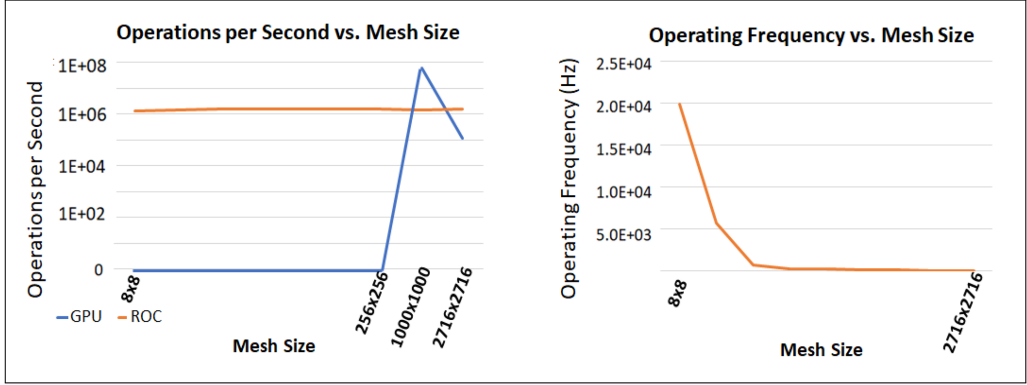


Fig. 19. For large mesh sizes, ROC outperforms a V100 GPU in terms of operations per second (left). Note that as the V100 performance dips, the ROC performance stays constant. This is true, even as the ROC operating frequency decreases for large meshes (right).

the expected direction of current flow. This imposes a sequential constraint on the system, where a maximum of $2N$ nodes in an $N \times N$ mesh can be operated on simultaneously.

A more telling statistic is seen in the slope of time-to-solution associated with growth in mesh size, shown in Figure 18. For meshes larger than 256×256 , the Xeon E5603 reaches a point of inflection, but ROC stays constant. The same observation holds for the V100 with a $1,000 \times 1,000$ mesh. This ensures that for large meshes, ROC will always outperform the Xeon and V100. It is worth noting that ROC biases were written naively, in a large serial chain clocked at 100 MHz. The ROC time-to-solution can be further reduced by parallelizing writes to biasing circuitry configured as multiple sub-chains, enabled by the mismatch in the bit-width of the biasing DACs (8 bit), and common bus widths available for high-performance computers (>128 bit) [53].

It should be noted that ROC times-to-solution are reflective of a device with components biased in a daisy-chain formation. Simulation results show that for computational meshes larger than 17×17 , time-to-solution is greatly dominated by the time required to bias devices. Therefore, the use of parallel chains for biasing an $N \times N$ mesh can reduce the time-to-solution by N^2/C , where C is the number of parallel chains used for configuration. For a $1,000 \times 1,000$ mesh, a configuration bitstream of 8 MB would require 0.6 s to complete, as compared to 1,000 15 MB bitstreams for a Xilinx XC5VFX200T, which would require 1,400 s to configure in a similar configuration using reported configuration rates [4, 5, 6].

GPU performance is typically measured in operations per second (OPS), as it takes into account the massive parallelism in operations enabled by such devices. While the steady-state heat transfer problem is not embarrassingly parallel, it benefits greatly from a parallel architecture, as seen in Figure 18. To create a proper OPS comparison between ROC, which provides a one-shot computation, and the V100, which provides an iterative computation, one must define an *operation* in the context of a computational mesh as the set of operations required to calculate a single node in the mesh. Figure 19 shows a reduction in the performance of the V100 as the mesh size increases, due to the serial features of the iterative computation process. Contrast this with the constant performance of ROC over meshes of any size.

Shown in Figure 20, a scaling study was performed where multiple mesh sizes of ROC was simulated and compared with its electrical counterpart for testcases TC-1 and TC-2. As seen in Table 3, accuracy increases with the size of the optical mesh, as is expected for other mesh-based architectures, where resolution is directly proportional to mesh size.

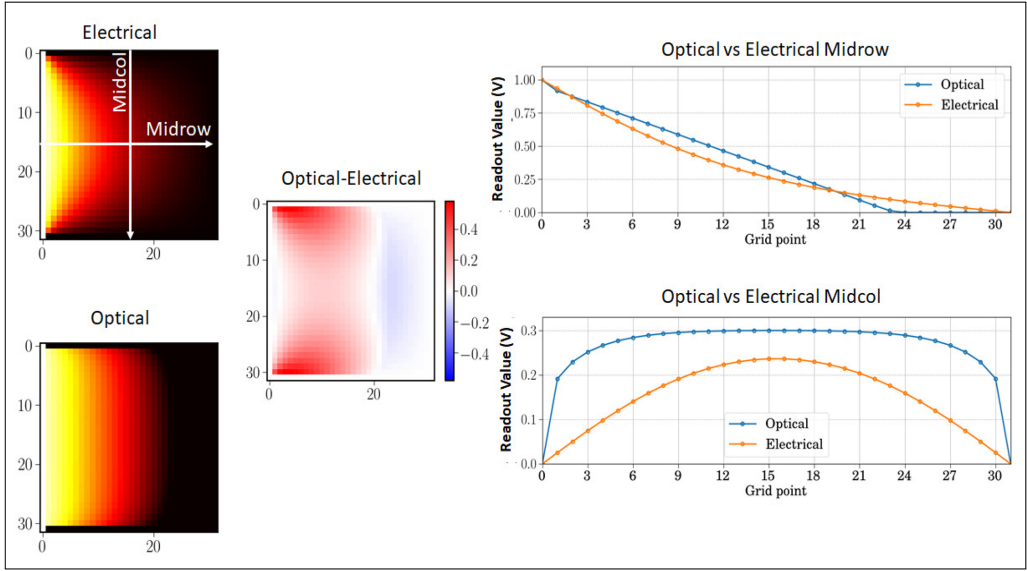


Fig. 20. Heat maps show the difference between solutions calculated by ROC and the resistive mesh. Midrow and midcolumn results are also contrasted. These results are extended to a full scaling study of different-sized meshes, which show performance of larger implementations of ROC as compared to the resistive mesh. As the size of the mesh increases, it more closely matches its counterpart.

Table 3. Accuracy of ROC Implementations with Different Mesh Sizes

Mesh Size	TC1 Mean	TC1 Max (%)	TC2 Mean	TC2 Max
4×4	85	31.8	100	100
8×8	69.4	36.2	72	32.8
16×16	71.6	39.9	66.4	49.3
32×32	87.6	69.2	87.6	70.2

7 ACCURACY AT SCALE

Ostensibly, an increased number of nodes within the ROC mesh results in higher accuracy due to increased resolution. However, as ROC nodes have equal-splitting routers, the number of hops between the optical signal source and sink is higher than those in a comparable electrical mesh. Moreover, the number of hops is dependent on the size of the mesh and the problem configuration (i.e., size and location of boundary conditions). Figure 21 contrasts the signal splitting behavior in electrical and optical meshes in a simple use case where the left side is the source and the right side is the sink. In the electrical case (Figure 21(a)), the currents move toward the sink without any splitting at the node. In the optical case (Figure 21(b)), due to the relative-scale of feature size to wavelength, light splits evenly at every intersection and does not mirror electrons in following the shortest path to ground. This not only reduces the amplitude of the signal but also adds noise due to additional signals created at each intersection.

With the passive-optics ROC, this effect can be compensated for by adjusting the signal attenuation between routers. Configurable attenuation of the optical signal is achieved by placing an ROE at each of the router's output ports, enabling attenuations between 0 and 100 percent.

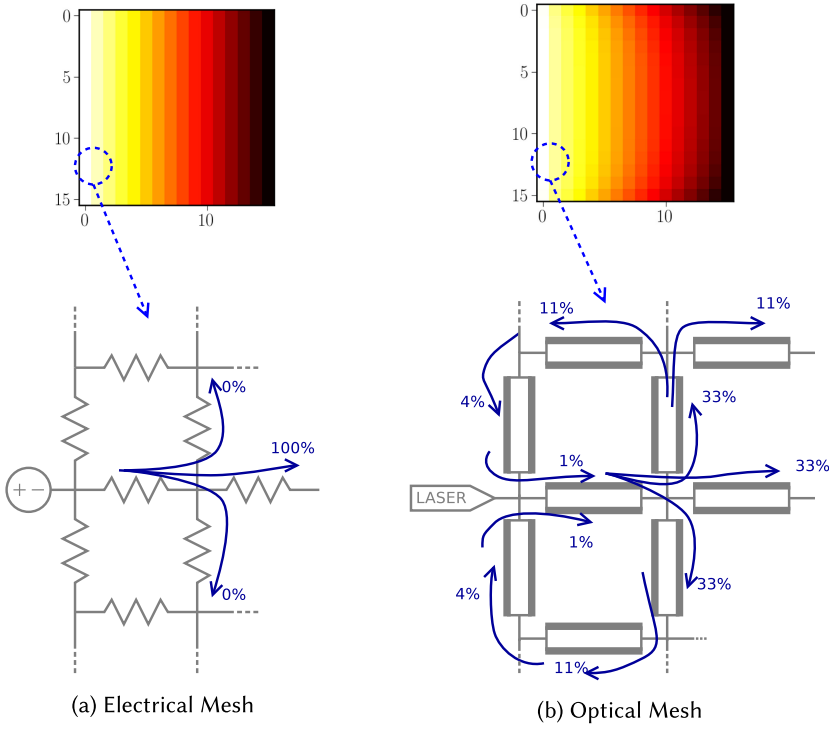


Fig. 21. Difference of signal splitting in electrical and optical meshes. Note the dispersion of the optical signal through all paths of the interconnect, while the electrical current is confined to paths predetermined by the voltages seen at each node.

To this end, we study the effect of different waveguide attenuation values on accuracy. Figure 22 shows the change of error with different loss values in Test Case 2 with meshes of size 16×16 . With first two cases, a very low attenuation leads to numerically higher results being obtained from ROC. However, with a much higher attenuation, ROC generates much lower values. For this case, 10^1 dB attenuation generates the most accurate results as shown in Figure 22(c).

We studied the effect of attenuation on accuracy with the cases describe above with ROC meshes of size 8×8 , 16×16 , and 32×32 . The results are shown in Figure 23 and Tables 4 through 6. In virtually all cases, correct attenuation adjustment results in errors close to 10%. The results also confirm that correct attenuation depends on the problem configuration. Arguably, deriving correct attenuation value based on problem configuration is a non-trivial problem, since number of hops is not a global value. However, simpler heuristics can be developed to pick an attenuation value based on problem configuration and size.

8 FUTURE WORK

Future work can be binned into three main thrusts: hardware, software, and applications. Software will consider programming and execution models as well as virtualization techniques that allow a single physical implementation of the ROC to approximate analog mesh computers of different sizes. Hardware considers alternative optical implementations using both plasmonics and metatronics. Also, a small-scale Silicon Photonic prototype is planned by using standard CMOS

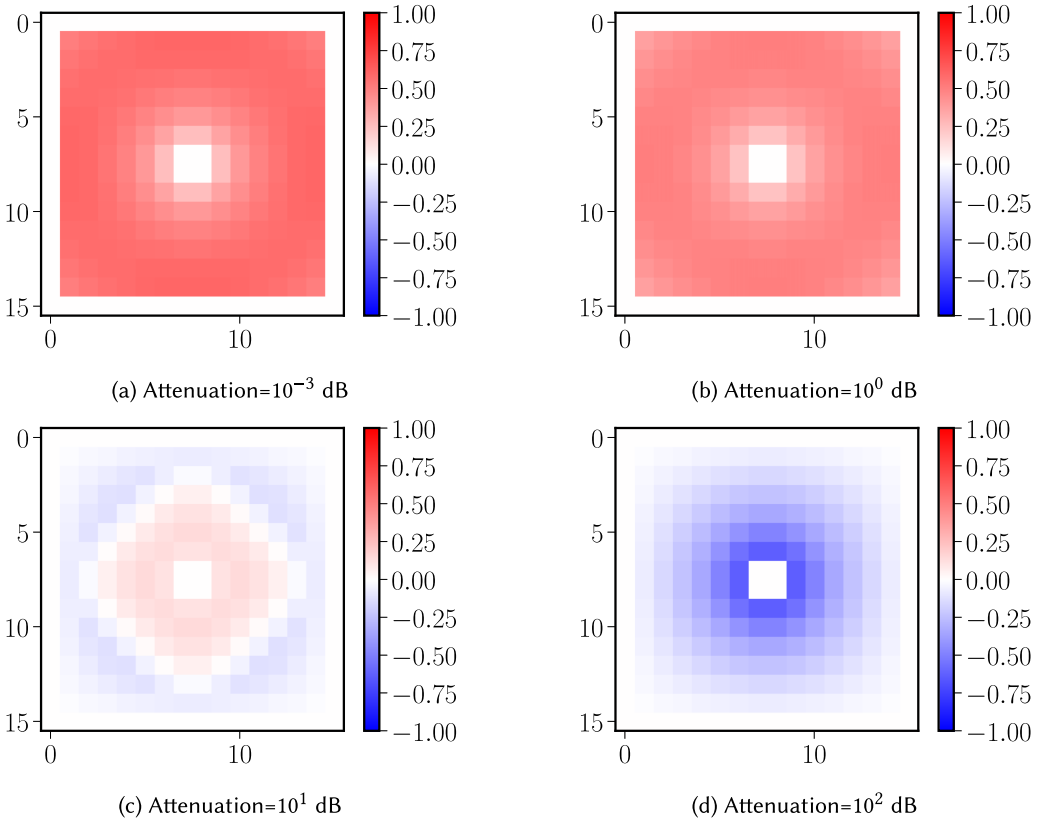


Fig. 22. Difference between ROC and corresponding electrical resistive mesh, for Test Case 2 with meshes of size 16×16 .

fabrication technology and material, as well as a large scale ROC network by using Multiple-Project-Wafer (MPW) provided by AIM Photonics [66].

Plasmonic-based implementations of the ROC require the addition of Indium-Tin-Oxide (ITO) devices, which allow the ROEs to be integrated into the optical router. The addition of ITO enables unequal splitting ratios to be programmed into the router, thus combining attenuation and routing into the same function and reducing both footprint and power. Comparing with traditional optical router that usually takes $102\text{--}104\ \mu\text{m}^2$ space on-chip, plasmonic-based metatronics could reduce the element scale down to a few 10's of nanometers, and intrinsically supports the “lumped element” feature due to its scale, which is orders of magnitude smaller than the wavelength.

Metatronic-based implementations of ROC enable the nanophotonic emulation of capacitors and inductors, in addition to resistors, through the biasing of a single metatronic device. This will enable the ROC to compute solutions for time-dependent PDEs, increasing its usefulness to the scientific community even further.

Due to the small size of metatronic components, a metatronic-based ROC will support a computational mesh orders of magnitude denser than the current optical implementation. This device density will enable ROC to offer practical support of partial reconfiguration, thus an ability to compute solutions to multiple problems or to iterate on solutions with growing limits on resolution.

Applications work will include supporting other types of PDEs and as such exploring the creation of photonic circuits to emulate meshes of arbitrary electric components. While using our

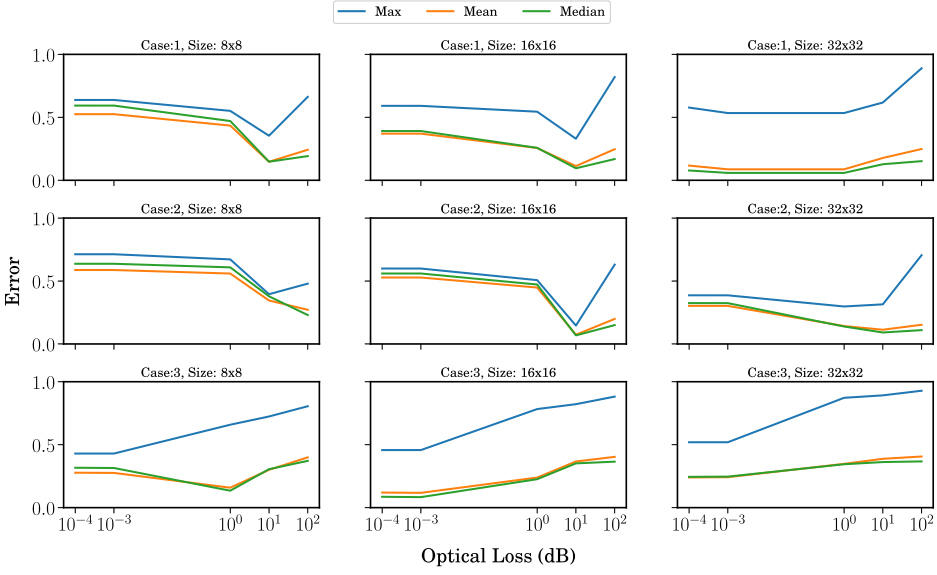


Fig. 23. Effect of waveguide attenuation on accuracy.

Table 4. Attenuation Sweep Results for Test Case 1

Loss (dB)	Size = 8×8			Size = 16×16			Size = 32×32		
	Max	Mean	Med	Max	Mean	Med	Max	Mean	Med
10^{-4}	0.64	0.52	0.59	0.59	0.37	0.39	0.53	0.12	0.08
10^{-3}	0.64	0.52	0.59	0.59	0.37	0.39	0.53	0.09	0.06
10^0	0.55	0.43	0.47	0.54	0.26	0.26	0.53	0.09	0.06
10^1	0.35	0.15	0.15	0.33	0.11	0.10	0.62	0.18	0.13
10^2	0.66	0.24	0.19	0.82	0.25	0.17	0.89	0.25	0.15

Table 5. Attenuation Sweep Results for Test Case 2

Loss (dB)	Size = 8×8			Size = 16×16			Size = 32×32		
	Max	Mean	Med	Max	Mean	Med	Max	Mean	Med
10^{-4}	0.71	0.59	0.64	0.60	0.53	0.56	0.39	0.30	0.32
10^{-3}	0.71	0.59	0.64	0.60	0.53	0.56	0.39	0.30	0.32
10^0	0.67	0.56	0.61	0.51	0.45	0.47	0.30	0.14	0.14
10^1	0.40	0.35	0.38	0.15	0.07	0.07	0.31	0.11	0.09
10^2	0.48	0.27	0.23	0.63	0.20	0.15	0.71	0.15	0.11

silicon photonics implementations 16×16 networks can be created, such networks can be scaled to much larger sizes using future technology developments such as the utilization of metatronics. Further, the use of repeaters can architecturally allow us to design larger networks. Finally, from the system perspective, virtualization can be introduced to reuse hardware allowing large problems to be mapped onto the realized hardware. These and similar investigations will be further explored.

Table 6. Attenuation Sweep Results for Case 3

Loss (dB)	Size = 8 × 8			Size = 16 × 16			Size = 32 × 32		
	Max	Mean	Med	Max	Mean	Med	Max	Mean	Med
10^{-4}	0.43	0.28	0.32	0.46	0.12	0.09	0.52	0.24	0.24
10^{-3}	0.43	0.28	0.31	0.46	0.12	0.08	0.52	0.24	0.25
10^0	0.66	0.16	0.13	0.78	0.24	0.23	0.87	0.35	0.34
10^1	0.72	0.30	0.31	0.82	0.37	0.35	0.89	0.39	0.36
10^2	0.81	0.40	0.37	0.88	0.40	0.37	0.93	0.41	0.37

9 CONCLUSION

Due to current technological limitations, traditional von Neumann architectures cannot continue to support future computations, and creative post-Moore's computer architectures are being sought out. Many of the proposed architectures target green computing and potentially exascale computing by efficiently computing solutions to domain-specific problems.

Here, we introduce ROC, the first reconfigurable nano-optical class of approximate computing processors, capable of efficiently accelerating the computation of PDEs. ROC does this by using optical components to emulate a two-dimensional resistor mesh architecture, which is capable of computing the solution of difference equations in a single shot, thereby eliminating the iterative stages required by traditional numerical solvers. This reduces the time-to-solution required for physical simulations by orders of magnitude. It can also harness the ultra-low power advantage of nano-photonics.

ROC was shown to calculate the solution to PDEs associated with heat transfer through a solid over four orders of magnitude faster than the numerical solver, COMSOL, and two orders of magnitude faster than a GPU-based solver. Additionally, the time-to-solution for a given mesh size scales at a much slower pace than COMSOL and the GPU-based solver, ensuring its performance dominance in the face of increasing resolution requirements.

As ROC's calculations trended in a similar fashion to solutions produced by COMSOL and the resistance network analogue, it behaves as expected for an approximate-computing engine and was shown to produce solutions within 10% of COMSOL. Much like other implementations of approximate-computers, solutions produced by ROC can be made more accurate through error-correction strategies, with the most efficient being through optimization of system-wide attenuation. Results are accurate, making this a viable approximate computer with potential for becoming an efficient exact computing machine by leveraging additional developments from material science such as using metatronics.

REFERENCES

- [1] Defense Advanced Research Project Agency. 2017. *Electronics Resurgence Initiative: Page 3 Investments Architectures Thrust*. Retrieved on January 20, 2018 from <https://www.darpa.mil/news-events/2017-09-13>.
- [2] J. Anderson, S. Sun, Y. Alkabani, V. Sorger, and T. El-Ghazawi. 2019. Photonic processor for fully discretized neural networks. In *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP'19)*.
- [3] Y. Li, Yue, I. Liberal, C. D. Giovampaola, and N. Engheta. 2016. Waveguide metatronics: Lumped circuitry based on structural dispersion. *Sci. Adv.* 2, 6 (2016).
- [4] K. Papadimitriou, A. Anyfantis, and A. Dollas. 2007. Methodology and experimental setup for the determination of system-level dynamic reconfiguration overhead. In *Proceedings of the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM'07)*.
- [5] K. Papadimitriou, A. Dollas, and S. Hauck. 2011. Performance of partial reconfiguration in FPGA systems: A survey and a cost model. *ACM Trans. Reconfigur. Technol. Syst.* 4, 4 (2011).

- [6] K. Papadimitriou, A. Anyfantis, and A. Dollas. 2010. An effective framework to evaluate dynamic partial reconfiguration in FPGA systems. *IEEE Trans. Instrument. Measure.* 59, 6 (2010).
- [7] A. Agrawal, J. Choi, K. Gopalakrishnan, S. Gupta, R. Nair, J. Oh, D. Prener, S. Shukla, V. Srinivasan, and Z. Sura. 2016. Approximate computing: Challenges and opportunities. In *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC'16)*.
- [8] T. El-Ghazawi, V. J. Sorger, S. Sun, A. A. Badawy, and V. Narayana. 2019. Reconfigurable optical computer (June 2019). Patent No. U.S. 10,318,680 B2. Filed December 5, 2016, issued June 11, 2019.
- [9] S. Mittal. 2016. Survey of techniques for approximate computing. In *ACM Computing Surveys*. Association of Computing Machinists.
- [10] S. Farlow. 1993. *Partial Differential Equations for Scientists and Engineers*. Dover.
- [11] J. Zhu. 1994. *Solving Partial Differential Equations on Parallel Computers*. World Scientific, River Edge, NJ.
- [12] L. Pinuel, I. Martin, and F. Tirado. 1998. A special-purpose parallel computer for solving partial differential equations. In *Proceedings of the 6th Euromicro Workshop on Parallel and Distributed Processing*. IEEE.
- [13] G. Liebmann. 1950. *Solution of Partial Differential Equations with a Resistance Network Analogue*. British Journal of Applied Physics, Institute of Physics.
- [14] P. Palmer, A. R. Copson, and S. C. Redshaw. 1959. *Investigations into the Use of an Electrical Resistance Analogue for the Solution of Certain Oscillatory-Flow Problems*, Reports and Memoranda No. 312. Aeronautical Research Council.
- [15] J. Ramirez-Angulo and Mark R. DeYong. 2000. *Digitally-configurable analog VLSI Chip and Method for Real-time Solution of Partial Differential Equations*, U.S. Patent US6141676 A.
- [16] A. Hastings. 2005. *The Art of Analog Layout*, 2nd ed. Prentice Hall.
- [17] G. Liebmann. 1954. *Resistance-network Analogues with Unequal Meshes or Subdivided Meshes*. British Journal of Applied Physics, Institute of Physics.
- [18] V. K. Narayana, O. Serres, J. Lau, S. Licht, and T. El-Ghazawi. 2014. Towards a computational model for heat transfer in electrolytic cells. *Int. J. Comput. Theory Eng.* 6, 3 (2014).
- [19] W. Schiesser. 1991. *The Numerical Method of Lines, 1st Edition, Integration of Partial Differential Equations*. Elsevier.
- [20] R. Wienands and W. Joppich. 2005. *Practical Fourier Analysis for Multigrid Methods*. CRC Press.
- [21] H. J. Lee and W. E. Schiesser. 2003. *Ordinary and Partial Differential Equation Routines in C, C++, Fortran, Java, Maple and MATLAB*. CRC Press.
- [22] R. M. M. Mattheij, S. W. Rienstra, and J. H. M. ten Thijs Boonkkamp. 2005. Partial differential equations: Modeling, analysis, computation. *Soc. Industr. Appl. Math.* (2005), 79–82.
- [23] R. M. Corless. 2002. A new view of the computational complexity of IVP for ODE. *Numer. Algor.* 31, 1 (2002), 115–124.
- [24] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie. 2016. PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In *Proceedings of ACM/IEEE 43rd Annual International Symposium on Computer Architecture*.
- [25] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams. 2016. *Dot-Product Engine for Neuromorphic Computing: Programming 1T1M Crossbar to Accelerate Matrix-Vector Multiplication*, HPE-2016-23, Hewlett-Packard Labs.
- [26] Texas Instruments. 2016. *OPA857 Ultralow-Noise, Wideband, Selectable-Feedback Resistance Transimpedance Amplifier, Revision D*.
- [27] Texas Instruments. 2005. *TS5N118 1-OF-8 FET Multiplexer/Demultiplexer High-bandWidth Bus Switch*.
- [28] Texas Instruments. 2007. *ADS1605 16-Bit, 5MSPS Analog-to-Digital Converter*.
- [29] I. Herron and M. Foster. 2008. *Partial Differential Equations in Fluid Dynamics*. Cambridge Press.
- [30] M. N. O. Sadiku and L. C. Agba. 1990. A simple introduction to the transmission-line modeling. *IEEE Trans. Circ. Syst.* 37, 8 (1990), IEEE.
- [31] H. Berestycki and Y. Pomeau (eds). 2002. *Nonlinear PDE's in Condensed Matter and Reactive Flows*. Springer.
- [32] A. P. S. Selvadurai. 2000. *Partial Differential Equations in Mechanics 1: Fundamentals, Laplace's Equation, Diffusion Equation, Wave Equation*. Springer.
- [33] C. Tang, Q. Mi, H. Yan, J. Yang, and S. Liu. 2013. PDE (ODE)-based image processing methods for optical interferometry fringe. In *Proceedings of International Conference on Optics in Precision Engineering and Nanotechnology*.
- [34] C. D. Sarris. 2006. Adaptive mesh refinement in time-domain numerical electromagnetics. *Synthesis Lectures on Computational Electromagnetics*. Morgan & Claypool.
- [35] M. J. Berger and J. Oliger. 1984. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* 53, 3 (1984), Elsevier.
- [36] G. Liebmann. 1955. *An Improved Experimental Iteration Method for Use with Resistance-network Analogues*. British Journal of Applied Physics, Institute of Physics.
- [37] M. N. Bojnordi and E. Ipek. 2016. Memristive Boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*. IEEE.

- [38] L. C. Evans. 1998. *Partial Differential Equations*. American Mathematical Society, Providence, RI.
- [39] A. Benoit, F. Chyzak, A. Darrasse, S. Gerhold, M. Mezzarobba, and B. Salvy. 2010. *The Dynamic Dictionary of Mathematical Functions*. International Congress on Mathematical Software, Springer, Berlin.
- [40] M. Alioto. 2017. Energy-quality scalable adaptive VLSI circuits and systems beyond approximate computing. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*. IEEE.
- [41] J. George, H. Nejadrahi, and V. J. Sorger. 2017. Towards On-Chip Optical FFTs for Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC'17)*. IEEE.
- [42] J. Dongarra. 2017. *Current Trends in High Performance Computing and Challenges for the Future*. ACM Learning Webinar.
- [43] V. J. Sorger, N. D. Lanzillotti-Kimura, R. M. Ma, and X. Zhang. 2012. Ultra-compact silicon nanophotonic modulator with broadband response. *Nanophotonics* 1, 1 (2012).
- [44] S. Sun, V. K. Narayana, I. Sarpkaya, J. Crandall, R. A. Soref, H. Dalir, T. El-Ghazawi, and V. J. Sorger. 2017. Hybrid photonic-plasmonic non-blocking broadband 5x5 router for optical networks. *IEEE Photon. J.* 10, 2 (2017), 1–13.
- [45] H. Yin. 2011. *Application of Resistivity-Tool-Response Modeling For Formation Evaluation: AAPG Archive Series 2*, 2. AAPG.
- [46] P. B. Hansen. 1992. *Numerical Solution of Laplace's Equation*, Electrical Engineering and Computer Science Technical Reports, 9-1992, Syracuse University.
- [47] J. D. Irwin and R. M. Nelms. 2015. *Basic Engineering Circuit Analysis*, 11th ed. Wiley.
- [48] T. Meis and U. Marcowitz. 1981. *Numerical Solution of Partial Differential Equations*. Springer-Verlag.
- [49] A. R. Sanderson, M. D. Meyer, R. M. Kirby, and C. R. Johnson. 2009. A framework for exploring numerical solutions of advection-reaction-diffusion equations using a GPU-based approach. *Comput. Visual. Sci.* 12, 4 (2009).
- [50] Y. Zhao. 2008. Lattice Boltzmann-based PDE solver on the GPU. *Visual Comput.* 24, 5 (2008).
- [51] M. Karunaratne, A. K. Mohite, T. Mitra, and L. S. Peh. 2017. HyCUBE: A CGRA with reconfigurable single-cycle multi-hop interconnect. In *Proceedings of the 54th Annual Design Automation Conference (DAC'17)*. ACM.
- [52] Xilinx Corporation. 2008. *XST User Guide*, v. 10.1 (2008).
- [53] J. Jeffers, J. Reinders, and A. Sodani. 2016. *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition*, 2nd ed. Morgan Kaufmann.
- [54] M. T. Ibn Ziad, M. Hossam, M. A. Masoud, M. Nagy, H. A. Adel, Y. Alkabani, M. W. El-Kharashi, K. Salah, and M. AbdelSalamb. 2015. *On Kernel Acceleration of Electromagnetic Solvers via Hardware Emulation*. Elsevier.
- [55] R. Ji, J. Xu, and L. Yang. 2013. Five-port optical router based on microring switches for photonic networks-on-chip. In *IEEE Photonics Technology Letters*. IEEE.
- [56] Y. Jiao, S. F. Mingaleev, M. Schillinger, D. Miller, S. Fan, and K. Busch. 2005. Wannier basis design and optimization of a photonic crystal waveguide crossing. *IEEE Photonics Technology Letters*. IEEE.
- [57] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neu Rentz-Reichert, and C. Wieners. 1997. UG—A flexible software toolbox for solving partial differential equations. *Comput. Visual. Sci.* 1, 1 (1997).
- [58] X. Li. 2005. An overview of SuperLU: Algorithms, implementation, and user interface. *ACM Trans. Math. Softw.* 31, 3 (2005).
- [59] N. Sherwood-Droz, H. Wang, L. Chen, B. G. Lee, A. Biberman, K. Bergman, and M. Lipson. 2008. Optical 4×4 hitless silicon router for optical Networks-on-Chip (NoC). *Optics Expr.* 16, 20 (2008).
- [60] H. K. Tsang and Y. Liu. 2008. Nonlinear optical properties of silicon waveguides. *Semicond. Sci. Technol.* 23, 6 (2008), 064007.
- [61] N. Engheta. 2007. Circuits with light at nanoscales: Optical nanocircuits inspired by metamaterials. *Science* 317, 5845 (2007), 1698–1702.
- [62] A. Silva, F. Monticone, G. Castaldi, V. Galdi, A. Alu, and N. Engheta. 2014. Performing mathematical operations with metamaterials. *Science* 343, 6167 (2014), 160–163.
- [63] A. Mehrabian, Y. Alkabani, V. Sorger, and T. El-Ghazawi. 2018. PCNNA: A photonic convolutional neural network accelerator. arXiv:1807.08792.
- [64] Institute of Electrical and Electronics Engineers. 2013. > 1149.1-2013—IEEE Standard for Test Access Port and Boundary-Scan Architecture. IEEE.
- [65] J. Townsend. 2012. *A Modern Approach to Quantum Mechanics*, 2nd ed. University Science Books.
- [66] AIM Photonics. 2019. Retrieved from <http://www.aimphotonics.com>.

Received December 2018; revised August 2019; accepted October 2019