

Iterative photonic processor for fast complex-valued matrix inversion

MINJIA CHEN,¹ QIXIANG CHENG,^{1,*} MASAFUMI AYATA,² MARK HOLM,² AND RICHARD PENTY¹

¹Department of Engineering, Centre for Photonic Systems, Electrical Engineering Division, University of Cambridge, Cambridge CB3 0FA, UK

²Huawei Technologies (Sweden) AB, 164 40 Kista, Sweden

*Corresponding author: qc223@cam.ac.uk

Received 16 June 2022; revised 9 August 2022; accepted 25 August 2022; posted 29 August 2022 (Doc. ID 468097); published 17 October 2022

An $N \times N$ iterative photonic processor is proposed for the first time, we believe, for fast computation of complex-valued matrix inversion, a fundamental but computationally expensive linear algebra operation. Compared to traditional digital electronic processing, optical signal processing has a few unparalleled features that could enable higher representational efficiency and faster computing speed. The proposed processor is based on photonic integration platforms—the inclusion of III-V gain blocks offers net neutral loss in the phase-sensitive loops. This is essential for the Richardson iteration method that is adopted in this paper for complex linear systems. Wavelength multiplexing can be used to significantly improve the processing efficiency, allowing the computation of multiple columns of the inverse matrix using a single processor core. Performances of the key building blocks are modeled and simulated, followed by a system-level analysis, which serves as a guideline for designing an $N \times N$ Richardson iteration processor. An inversion accuracy of >98% can be predicted for a 64×64 photonic processor with a >80 times faster inversion rate than electronic processors. Including the power consumed by both active components and electronic circuits, the power efficiency of the proposed processor is estimated to be over an order of magnitude more energy-efficient than electronic processors. The proposed iterative photonic integrated processor provides a promising solution for future optical signal processing systems. © 2022 Chinese Laser Press

<https://doi.org/10.1364/PRJ.468097>

1. INTRODUCTION

Complex-valued matrix inversion is a fundamental but computationally expensive linear algebra operation, which is widely used in wireless communication systems [1–5], control systems [6], image processing [7,8], for solving partial differential equations [9], and in many other applications. Complex-valued arithmetic is preferred in many applications due to its better performance compared to real-valued calculations in terms of representational capacity, convergence speed, generalization, and noise robustness [10,11].

Two main advantages of optical matrix computations over traditional digital electronic computations include higher representational efficiency and faster processing speed [12]. First, the real and imaginary parts of one complex-valued number are stored in two registers on traditional electronic computing platforms, with calculations being carried out for real and imaginary parts, respectively. In contrast, an optical signal inherently has two dimensions to encode the real and imaginary parts, which are its amplitude and phase. Second, optical matrix computation allows faster processing speeds through inherent high-speed optical processing. The speeds of digital electronic processors are limited by the transistors' switching time and time for storing, fetching, and moving intermediate computing

results, while the speed of optical matrix computation is solely determined by the signal propagation time in the processor. Photonic integration platforms guarantee ultrafast computing speed due to their ultracompact millimeter-level optical path length. In addition, intermediate computing results can be directly sent to the succeeding components without storing and fetching. In other words, the computation is completed during light propagation.

Optical computing has a long history that dates back to the 1980s [13]. Optical systems for real-valued matrix inversion have been implemented using both free-space optics [14] and fiber-optic networks [15]. Though free-space optical systems have good scalability, the bulky system and high alignment requirements pose big obstacles for practical usage and lead to higher latency than more compact integrated solutions. Fiber-optic systems have large phase fluctuations, making them unsuitable for complex-valued computations. In contrast, photonic integrated platforms provide ultracompact designs and accurate phase control [16–21], which enable the implementation of scalable optical networks for complex-valued matrix inversion.

In this paper, we propose an $N \times N$ photonic integrated Richardson iteration processor (RIP) for fast computation of

complex-valued matrix inversion and a “Min-Max” algorithm for choosing the optimal parameter for the Richardson (RICH) method for the highest convergence rate, the first time this has been proposed to our knowledge. We also build mathematical models for key building blocks and predict both block-level and system-level performances, which can be used as guidelines for designing an $N \times N$ RIP.

The paper is organized as follows. Section 2 reviews the major matrix inversion methods and presents the Min-Max algorithm. Section 3 shows the system architecture of the proposed RIP and the detailed mathematical models of key blocks. Implementations on integrated photonic platforms are also compared. Section 4 presents system-level performance analyses, including speed, power efficiency, accuracy, and scalability based on the models built in Section 3, which can be used as guidelines for choosing input signal powers and bandwidths (BW's) of optical bandpass filters (BPFs) when designing an $N \times N$ processor. Section 5 concludes the paper.

2. INVERSION METHODS

Choosing an appropriate computing method is the first concern when we start to build up an optical matrix inverter. In this section, we review two major types of inversion methods, and introduce the Min-Max algorithm for implementing the RICH method in our design.

A. Review of Inversion Methods

In a broad sense, every matrix has its inverse matrix, which is called a generalized inverse or a pseudoinverse [22]. In this paper, we focus on computing the inverses of nonsingular (invertible) square matrices. For a nonsingular matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ (\mathbf{A} is a matrix with N rows and N columns of complex-valued elements), the unique inverse matrix \mathbf{A}^{-1} is defined by $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_N$, where \mathbf{I}_N is the $N \times N$ identity matrix with ones on the main diagonal and zeros elsewhere.

Calculating the inverse matrix \mathbf{A}^{-1} is equivalent to solving N linear equations, $\mathbf{A}\mathbf{x}_i = \mathbf{e}_i$ ($i = 1, 2, \dots, N$), where \mathbf{x}_i is the i th column of the inverse matrix \mathbf{A}^{-1} , and \mathbf{e}_i is the i th column of the identity matrix \mathbf{I}_N . Furthermore, matrix inversion methods can be classified into direct inversion methods and iterative inversion methods [9].

Direct inversion methods are based on the decomposition of the original matrix \mathbf{A} . Gaussian elimination (GE), lower-upper decomposition (LUD), Cholesky decomposition (CD), and QR decomposition (QRD) all convert \mathbf{A} into an upper or lower triangular form, \mathbf{U} or \mathbf{L} , and use back or forward substitution to serially compute the inversion results. Singular value decomposition (SVD) decomposes \mathbf{A} into two unitary matrices \mathbf{P} , \mathbf{Q} , and one diagonal matrix with nonnegative real elements $\mathbf{\Sigma}$, taking advantage of the facts that the inverse of a unitary matrix is its conjugate transpose, and the inverse of a diagonal matrix consists of reciprocals of its elements. Table 1 summarizes main direct inversion methods mentioned above for nonsingular matrices [9,23]. The time complexity shown in Table 1 refers to the complexity for solving all N equations. Time complexity depicts how the computing time scales with the matrix size, which is usually measured by floating-point operations per second (FLOP/s), since traditionally matrix inversion is computed

Table 1. Summary of Main Direct Inversion Methods

Method	Constraints on \mathbf{A}	Key Steps	Complexity
GE	None	1) $\mathbf{Ax} = \mathbf{e} \rightarrow \mathbf{Ux} = \mathbf{y}$ 2) Back substitution	$\sim O(N^3)$
LUD	None	1) $\mathbf{A} = \mathbf{LU}$ 2) $\mathbf{Ux} = \mathbf{y}$, $\mathbf{Ly} = \mathbf{b}$ 3) Forward substitution: \mathbf{y} 4) Back substitution: \mathbf{x}	$\sim O(N^3)$
CD	Positive definite	1) $\mathbf{A} = \mathbf{U}^*\mathbf{U}$ 2) $\mathbf{Ux} = \mathbf{y}$, $\mathbf{U}^*\mathbf{y} = \mathbf{b}$ 3) Forward substitution: \mathbf{y} 4) Back substitution: \mathbf{x}	$\sim O(N^3)$
QRD	None	1) $\mathbf{A} = \mathbf{QR}$ 2) $\mathbf{Rx} = \mathbf{y}$, $\mathbf{Qy} = \mathbf{b}$ 3) $\mathbf{y} = \mathbf{Q}^*\mathbf{b}$ 4) Back substitution	$\sim O(N^3)$
SVD	None	1) $\mathbf{A} = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^*$ 2) $\mathbf{A}^{-1} = \mathbf{Q}\mathbf{\Sigma}^{-1}\mathbf{P}^*$	$\sim O(N^3)$

on digital electronic computers. Though special matrices (such as sparse matrices) require fewer FLOP/s to compute, time complexity without any acceleration is shown here for generality. As shown in Table 1, direct methods all have a time complexity of $\sim O(N^3)$.

Iterative inversion methods compute the inverse matrix by first choosing an approximate initial guess and then updating it in each iteration until it converges. The uniqueness of inverse matrix guarantees the process always converges to the correct solution if the convergence condition is satisfied. Specifically, a solution sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}$ is generated in order to solve $\mathbf{Ax} = \mathbf{e}$. The iterative relationship is expressed as $\mathbf{x}^{(k+1)} = \mathbf{M}\mathbf{x}^{(k)} + \mathbf{f}$, which converges to $\mathbf{x}^* = \mathbf{M}\mathbf{x}^* + \mathbf{f}$ if $\|\mathbf{M}\| < 1$, where \mathbf{x}^* is the ideal solution, and $\|\cdot\|$ denotes the 2-norm of the matrix. In principle, accurate solutions can be reached after infinite numbers of iterations, while in practice, the iterative process stops after the deviation from the ideal solution is smaller than pre-set values. Iterative inversion methods include classical methods such as Jacobi (JC), Gauss–Seidel (GS), successive overrelaxation (SOR) and RICH, and gradient descent methods such as steepest descent (SD) and conjugate gradient (CG). Table 2 summarizes main iterative methods [9,24–26]. The total computing time depends on both the complexity and the convergence rate. The $\sim O(N^3)$ time complexity shown in Table 2 denotes computing time for each iteration without any acceleration, which is essentially the computing time of a matrix–matrix multiplication. The convergence rate, representing the number of iterations for convergence, depends on the inversion method, the initial guess, $\mathbf{x}^{(0)}$, and the required accuracy.

In general, direct inversion methods provide more robust solutions in linear systems than iterative inversion methods. Nevertheless, a good initial guess and the use of preconditioning (a means of transforming the original linear equations into one with the same solutions but that is easier to solve) in iterative methods allow for faster computing speed than direct methods, which is especially advantageous for large sparse systems.

Table 2. Summary of Main Iterative Inversion Methods

Method	Convergence Condition	Iterative Relationship	Complexity	Convergence Rate
JC	Positive definite	$\mathbf{x}^{(k+1)} = (\mathbf{I}_N - \mathbf{D}^{-1}\mathbf{A})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{e}$	$\sim O(N^3)$	Slow
GS	Positive definite	$\mathbf{x}^{(k+1)} = (\mathbf{I}_N - (\mathbf{D} - \mathbf{E})^{-1}\mathbf{A})\mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{E})^{-1}\mathbf{e}$	$\sim O(N^3)$	Faster than JC
SOR	Positive definite, $0 < \omega < 2$	$\mathbf{x}^{(k+1)} = (\mathbf{I}_N - \omega(\mathbf{D} - \omega\mathbf{E})^{-1}\mathbf{A})\mathbf{x}^{(k)} + \omega(\mathbf{D} - \omega\mathbf{E})^{-1}\mathbf{e}$	$\sim O(N^3)$	$\omega > 1$: accelerate; $\omega = 1$: GS; $\omega < 1$: slow down
RICH	Eigenvalues lie in a half-complex plane	$\mathbf{x}^{(k+1)} = (\mathbf{I}_N - \omega\mathbf{A})\mathbf{x}^{(k)} + \omega\mathbf{e}$	$\sim O(N^3)$	Depend on the choice of ω
SD	Positive definite	1) $\mathbf{r}^{(0)} = \mathbf{e} - \mathbf{A}\mathbf{x}^{(0)}$ 2) $\mathbf{p}^{(k)} = \mathbf{r}^{(k)}$ 3) $\alpha_k = \frac{\mathbf{p}^{(k)T}\mathbf{r}^{(k)}}{\mathbf{p}^{(k)T}\mathbf{A}\mathbf{p}^{(k)}}$ 4) $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{p}^{(k)}$ 5) $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k\mathbf{A}\mathbf{p}^{(k)}$	$\sim O(N^3)$	As slow as JC Be accelerated with preconditioning
CG	Positive definite	1) $\mathbf{r}^{(0)} = \mathbf{e} - \mathbf{A}\mathbf{x}^{(0)}$ 2) $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$ 3) $\alpha_k = \frac{\mathbf{r}^{(k)T}\mathbf{r}^{(k)}}{\mathbf{p}^{(k)T}\mathbf{A}\mathbf{p}^{(k)}}$ 4) $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{p}^{(k)}$ 5) $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k\mathbf{A}\mathbf{p}^{(k)}$ 6) $\beta_k = \frac{\mathbf{r}^{(k+1)T}\mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)T}\mathbf{r}^{(k)}}$ 7) $\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta_k\mathbf{p}^{(k)}$	$\sim O(N^3)$	Slightly faster than steepest descent Faster than SOR with preconditioning

B. Richardson's Method and Min-Max Algorithm

The matrix-form Richardson's method (the matrix-form RICH method) is shown in Eq. (1),

$$\mathbf{X}^{(k+1)} = (\mathbf{I}_N - \omega_{\text{opt}}\mathbf{A})\mathbf{X}^{(k)} + \omega_{\text{opt}}\mathbf{I}_N. \quad (1)$$

It is chosen for optical matrix inversion mainly for its easy implementation on integrated photonic platforms. A lack of mature on-chip optical memory [27] restricts the inversion methods to classical iterative methods, since both direct methods and gradient descent methods require storage of intermediate results, while for classical iterative methods, outputs can be easily sent back to inputs through waveguides for iterative computations. Among all the classical iterative methods, the RICH method has the simplest form and the fewest constraints, justifying its usage in computing matrix inverses. To implement the weight matrix $\mathbf{M} = \mathbf{I}_N - \omega_{\text{opt}}\mathbf{A}$ in the RICH method optically, we can either use Mach-Zehnder interferometer (MZI) arrays to encode the matrix elements directly or use cascaded MZI meshes to encode the SVD of the weight matrix. Here we choose to use MZI arrays to encode the elements directly to avoid the heavy overhead introduced by SVD.

In the general iterative relationship, $\mathbf{x}^{(k+1)} = \mathbf{M}\mathbf{x}^{(k)} + \mathbf{f}$, the convergence is guaranteed when the spectral radius $\rho(\mathbf{M})$ satisfies $\rho(\mathbf{M}) < 1$ [$\rho(\mathbf{M}) = \max_{\lambda \in \sigma(\mathbf{M})} |\lambda|$, $\sigma(\mathbf{M})$ is the collection of all the eigenvalues of matrix \mathbf{M}]. The convergence

rate, $\tau = -\ln(\rho(\mathbf{M}))$, is also determined by the spectral radius. As in the case of the RICH method, the convergence condition is $\rho(\mathbf{I}_N - \omega\mathbf{A}) < 1$, and the convergence rate is $\tau = -\ln(\rho(\mathbf{I}_N - \omega\mathbf{A}))$. As long as all the eigenvalues of \mathbf{A} lie in a half-complex plane, there exists a damping parameter ω such that the RICH method converges. We propose a Min-Max algorithm to compute the optimal ω corresponding to the fastest convergence rate. The basic idea is to scale the eigenvalues of matrix \mathbf{A} up or down into a unit circle centred at (1,0) on the complex plane and minimize the maximal distance of an eigenvalue to the point (1,0). A graphical explanation is shown in Fig. 1.

3. PHOTONIC DESIGN

This section consists of three parts. In part A, the system architecture of the proposed RIP is introduced. In part B, the mathematical modeling of key building blocks in the iterative photonic processor is presented. In part C, photonic integration schemes of the iterative photonic processor are discussed and compared.

A. System Architecture

Figure 2(a) illustrates the workflow of the proposed iterative processing system. The computation includes four main steps:

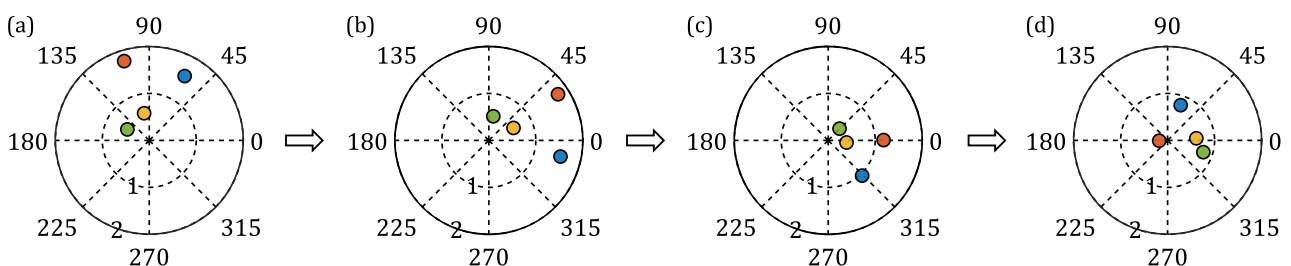


Fig. 1. Graphical explanation of the Min-Max algorithm for a 4×4 processor. Eigenvalues represented by circles in (a)–(d) correspond to colors. (a) Eigenvalues of \mathbf{A} shown in the polar coordinate system (all the eigenvalues lie in a half-complex plane, indicating the RICH method converges for a certain ω); (b) rotate all the eigenvalues of \mathbf{A} into the right half-plane; (c) eigenvalues of $\omega_{\text{opt}}\mathbf{A}$; (d) eigenvalues of $\mathbf{I}_N - \omega_{\text{opt}}\mathbf{A}$. Now the convergence condition $\rho(\mathbf{I}_N - \omega_{\text{opt}}\mathbf{A}) < 1$ is satisfied and the fastest convergence rate is also reached.

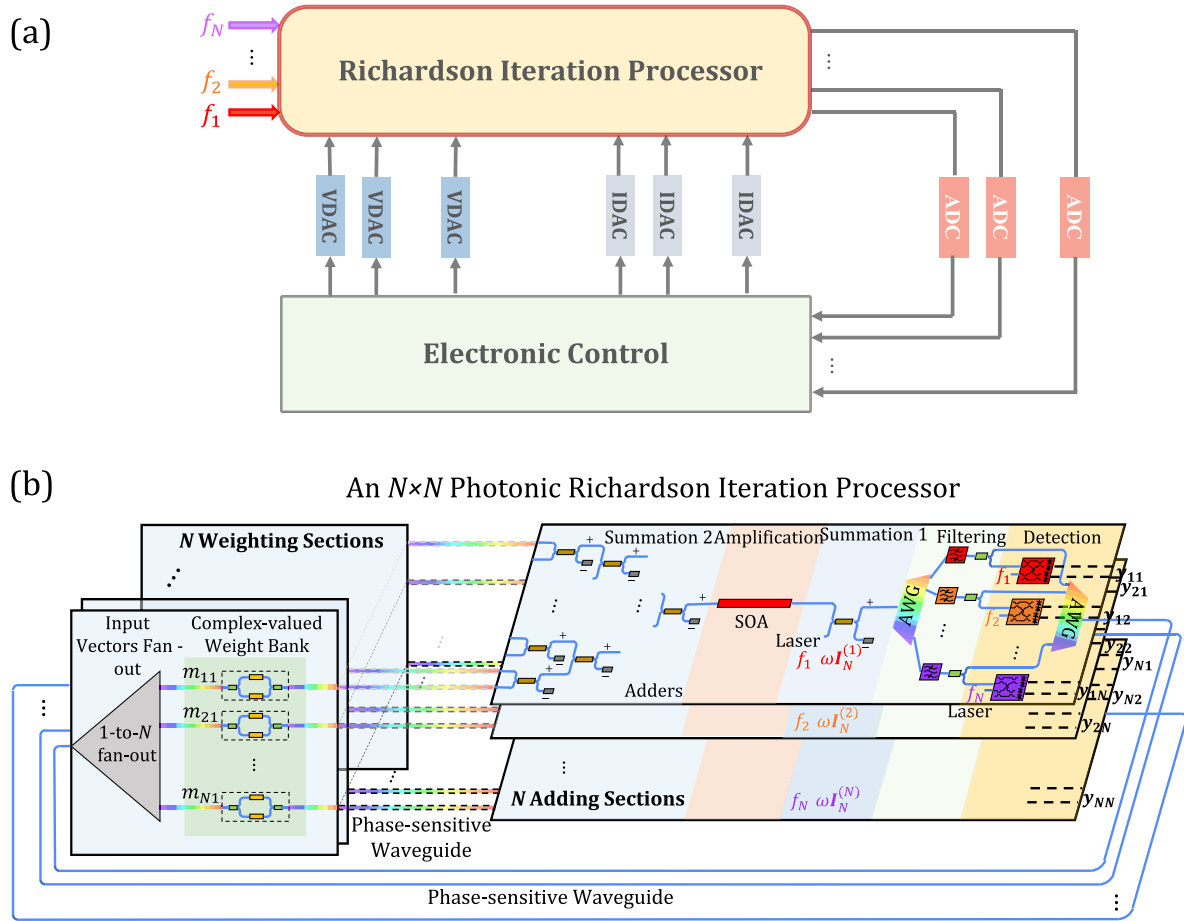


Fig. 2. System architecture of the proposed $N \times N$ iterative photonic processor for complex-valued matrix inversion. (a) Workflow of the iterative photonic processing system. The computation includes four main steps: 1) weights loading; 2) gain setting; 3) computation activation; and 4) results readout. VDAC, voltage digital-to-analog converter; IDAC, current digital-to-analog converter. (b) Architecture of an $N \times N$ iterative photonic processor. It consists of nine key photonic blocks, including *Laser*, *Summation 1*, *Input Vectors Fan-Out*, *Weight Bank*, *Summation 2*, *Amplification*, *Filtering*, *Detection*, and *Recirculating Loop*. AWG, arrayed waveguide gratings.

1) **Weights loading:** a preprocessed matrix $\mathbf{M} = \mathbf{I}_N - \omega_{\text{opt}}\mathbf{A}$ is loaded into the iterative processor by an array of voltage digital-to-analog converters (VDACs). Optimal ω is precalculated in the electronic peripherals according to the Min-Max algorithm. 2) **Gain setting:** the gains of semiconductor optical amplifiers (SOAs) are set by an array of current DACs (IDACs). SOAs are used to compensate for on-chip optical losses, ensuring lossless computations. 3) **Computation activation:** laser signals with frequencies ranging from f_1 to f_N are sent to N inputs of the iterative processor, corresponding to an initial guess, $\mathbf{X}^{(0)} = \mathbf{0}$, and an initial input to the processor, $\omega_{\text{opt}}\mathbf{I}_N$. The calculated ω_{opt} can be loaded into the processor by adjusting the laser power. The gain variations among different wavelength channels are equalized through adjusting laser powers. During the computation, each frequency corresponds to one column of the computation results. 4) **Results readout:** the optical inversion results are converted to electrical signals by coherent detection when the computation reaches its convergence. The electrical signals are then amplified by N arrays of transimpedance amplifiers (TIAs) and sampled by N arrays of

analog-to-digital converters (ADCs). Each ADC array contains $2N$ ADCs for the sampling of one row of the inversion results, which are then postprocessed to obtain the final inversion results. Figure 2(b) shows the detailed architecture of an $N \times N$ iterative photonic processor, which is the core part of the processing system in Fig. 2(a). The correspondence between key photonic blocks and their computational functionalities is summarized in Table 3.

B. Modeling

This part introduces the modeling of the iterative system in detail, serving as a preliminary for the system performance analyses in Section 4. Figure 3 shows the models of key building blocks, whose operational principles are characterized by mathematical formulas shown in the following [transverse-electric (TE) mode is assumed in the modeling].

1. Input Vectors Fan-Out Block

Figure 3(a) shows the model of a 1-to- N *Input Vectors Fan-out* block which consists of $\lceil \log_2 N \rceil$ stages of cascaded 50:50 1×2 MMI (multimode interference) couplers. The insertion loss

Table 3. Correspondence between Key Photonic Blocks and Computational Functionalities

Photonic Blocks	Components	Functionality
Laser	CW LDs	Input signal
Summation 1	Single-stage 50:50 $2 \times 2/1 \times 2$ MMI coupler	1) Couple initial input; 2) add ωI_N in each iteration
Input Vectors Fan-out	Cascaded 50:50 1×2 MMI couplers	Split looped-back signals
Weight Bank	Push-pull MZIs	Encode elements of complex-valued matrix \mathbf{M}
Summation 2	Cascaded 50:50 $2 \times 2/1 \times 2$ MMI couplers	Add signals up during matrix multiplication $\mathbf{M}\mathbf{X}^{(k)}$
Amplification	Cascaded SOAs	Compensate for on-chip losses
Filtering	AWGs and BPFs	Reduce the ASE noise from SOAs
Detection	Coherent detectors	Inversion results readout
Recirculating Loop	Phase-sensitive waveguides	Provide connections for iterative computation

(IL) of each MMI coupler is assumed to be α (in dB). E_i and $E_{o1} - E_{oN}$ represent the electric fields of the input signal and output signals, respectively. The output electric fields can be expressed by Eq. (2),

$$E_{oq} = 10^{-\frac{\alpha \cdot \lceil \log_2 N \rceil}{20}} \cdot \left(\frac{1}{\sqrt{2}} \right)^{\lceil \log_2 N \rceil} E_i, \quad q = 1, 2, \dots, N. \quad (2)$$

The absolute phase shift introduced during propagation is neglected in this model, since it can always be compensated by phase shifters. Only relative phase changes affect the encoded phase information. For a wavelength range of around 20 nm, which is enough to support 200 channels assuming a frequency spacing of 0.1 nm (according to the narrowest spectral grid of the ITU standard for dense wavelength division multiplexing technology [28]), <1% non-ideal coupling ratio and <1° phase deviation are reported over the entire BW [29]. The small

deviations can be compensated by adjusting the amplitude and phase of the MZI unit on the corresponding path. Thus, non-ideal coupling is neglected in the model.

2. Summation Block

Figure 3(b) shows the model of the *Summation* block, which consists of cascaded 50:50 2×1 MMI combiners. *Summation 1* block in Fig. 2(b) is the simplest form, with only one stage, while *Summation 2* block is a general form with $\log_2 N$ stages that adds up N optical signals according to Eq. (3), where α (dB) is the IL of an MMI coupler. Non-ideal coupling is neglected in this model, since it can be compensated in the *Weight Bank* block,

$$E_o = 10^{-\frac{\alpha \cdot \lceil \log_2 N \rceil}{20}} \cdot \left(\frac{j}{\sqrt{2}} \right)^{\lceil \log_2 N \rceil} \sum_{q=1}^N E_{iq}, \quad q = 1, 2, \dots, N. \quad (3)$$

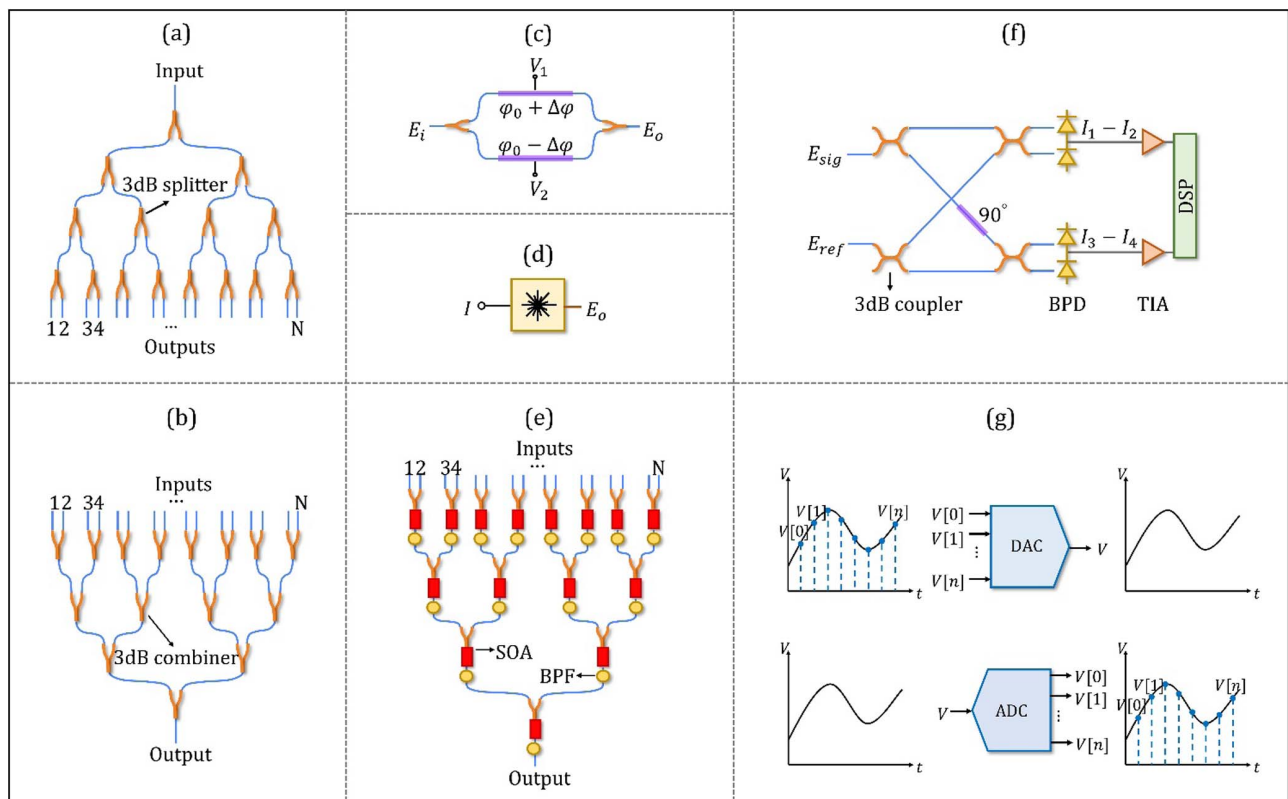


Fig. 3. Models of (a) *I-to-N Fan-Out* block, (b) *Summation* block, (c) *Weight Bank* block, (d) *Laser* block, (e) *Amplification and Filtering* blocks, (f) *Detection* block, and (g) electronic peripherals.

The 2×1 MMI combiners can be replaced with 2×2 MMI couplers with one output port configured as the monitor port for calibrating input phases, as shown in Fig. 2(b). Phase matching is required for both 2×1 and 2×2 MMI couplers when configured as adders. For 2×1 MMI couplers, the phase of the two input signals should be equal, while for 2×2 MMI couplers, the phase difference between the two inputs should equal 90° . The phase relationship between stage p ($1 \leq p \leq \lceil \log_2 N \rceil - 1$) and $p + 1$ of 2×2 MMI couplers is expressed in Eq. (4),

$$\varphi_q^p = \begin{cases} \varphi_q^{p+1}, & q = 1, 2, \dots, 2^{\lceil \log_2 N \rceil - 1}, \\ j\varphi_q^{p+1}, & q = 2^{\lceil \log_2 N \rceil}, 2^{\lceil \log_2 N \rceil} + 1, \dots, N, \\ 1 \text{ (} q = 1 \text{) or } j \text{ (} q = 2 \text{)}, & p = \lceil \log_2 N \rceil, \end{cases} \quad (4)$$

where φ_q^p represents the signal phase of the q^{th} input at stage p . Note that for *Summation 1* block in each of the “ N Adding Sections,” only the amplified signal whose frequency matches the laser frequency is required to satisfy the phase-matching condition, while amplified signals at other frequencies simply go through a power-splitting process since in each iteration, the addition of $\omega_{\text{opt}} \mathbf{I}_N$ in Eq. (1) is effective for only one element in each column. The phase-matching condition can be satisfied by applying a phase tuner between the laser and the input of the MMI coupler in *Summation 1*.

3. Weight Bank Block

Figure 3(c) shows the model of a single MZI unit of the *Weight Bank* block, since all the MZI units are functionally equivalent during computation. The MZI consists of two 50:50 1×2 MMI couplers and two phase shifters. Thermo-optic phase shifters (TOPSs) are used as an example in the modeling. A faster reconfiguration rate can be achieved using electro-optic phase shifters. The induced phase changes in two arms are determined by the applied voltage V_1 and V_2 , according to Eq. (5),

$$\Delta\phi_q(f) = \pi \frac{V_q^2}{V_\pi^2(f)} = \frac{2\pi f}{c} \cdot \frac{dn}{dT} \cdot \Delta T \cdot L, \quad q = 1, 2, \quad (5)$$

where $\frac{dn}{dT}$ is the thermo-optic (TO) coefficient of the waveguide material, ΔT is the temperature change due to the applied power, L is the length of phase shifters, and c is the speed of light in vacuum. The propagation loss in the waveguide of an MZI is mainly the propagation loss of two MMI couplers, which is 2α (in dB) according to the definition in *Summation 2* block. The amplitude and phase of a complex-valued weight element, m_{ij} , are encoded by the amplitude and phase differences between output and input electric fields as shown in Eqs. (6a), (6b),

$$|m_{ij}| = \frac{|E_o|}{|E_i|} = 10^{-\frac{2\alpha}{20}} |\cos(\Delta\phi(f))|, \quad (6a)$$

$$\arg(m_{ij}) = \arg(E_o) - \arg(E_i) = \varphi_o(f). \quad (6b)$$

The compensation of the non-ideal coupling mentioned in Sections 1 and 2 can be realized by tuning the amplitudes and phases of the MZI weights on corresponding paths.

4. Laser Block

Figure 3(d) shows the model of a continuous-wave (CW) laser source. N lasers are used to generate signals with frequencies ranging from f_1 to f_N . A higher wall-plug efficiency helps to reduce the power consumption of laser sources. The finite laser linewidth, $\Delta\nu$, is reflected in a randomly fluctuated phase term, $\theta(t)$, in the generated laser signal, as shown in Eq. (7),

$$\mathbf{E}_o = A_0 e^{j(2\pi f t + \varphi_0 + \theta(t))}. \quad (7)$$

The input signal power, $P_{\text{in}} = A_0^2$, serves as a normalization power (real-valued unit “1”) for the matrix computation. Laser phase noise normally causes unstable interference when two signals are combined in a summation block. This can be resolved by using a single laser block at each frequency point and splitting the laser signals for homodyne detection as explained in Section 6.

5. Amplification and Filtering Blocks

Figure 3(e) shows the model of the *Amplification* block together with the *Filtering* block for one wavelength channel. A cascaded SOA structure is used to reduce the signal power variations and avoid unwanted noise caused by nonlinear effects such as four-wave mixing, cross-gain modulation, and cross-phase modulation. Filters are placed after each SOA stage to reduce the ASE (amplified spontaneous emission) noise. Though the amplitude of the signal is dynamic among different iterations, they can be treated as DC signals by the optical BPFs within each iteration. The optical filter BW is thus not a limiting factor to the high data updating rate during multiple iterations. Figure 4(a) shows the ideal amplitude changes during computation without filtering, which can be expressed by Eq. (8), where T_0 is the amplitude updating period (time for signals to propagate in one iteration), and a_0, a_1, \dots, a_k are the amplitudes in the 1st, 2nd, ... and k^{th} iteration, respectively,

$$f(t) = a_0 u(t) + \sum_{i=0}^{k-1} (a_{i+1} - a_i) u(t - iT_0). \quad (8)$$

By removing the optical carrier frequency, the spectrum analysis can be shifted to baseband, which simplifies the Fourier analysis.

The baseband spectrum of the signal is shown in Eq. (9),

$$\mathcal{F}(\omega) = a_0 \left(\pi\delta(\omega) + \frac{1}{j\omega} \right) + \sum_{i=0}^{k-1} (a_{i+1} - a_i) \left(\pi\delta(\omega) + \frac{1}{j\omega} \right) e^{-j\omega iT_0}. \quad (9)$$

An optical BW of $2W$ (Hz) is equivalent to a baseband BW of W (Hz). The filter filters out part of the signal frequency components introduced by the abrupt amplitude changes among different iterations while keeping the DC components intact. The m^{th} term in Eq. (9) corresponds to a DC term with an amplitude of $a_m/2$ and a small variation term from $-\frac{\text{Si}(2\pi W T_0) (a_m - a_{m-1})}{\pi}$ at mT_0 to $\frac{\text{Si}(2\pi W T_0) (a_m - a_{m-1})}{\pi}$ at $(m+1)T_0$, where $\text{Si}(x)$ is the sine integral function, as shown in Fig. 4(b).

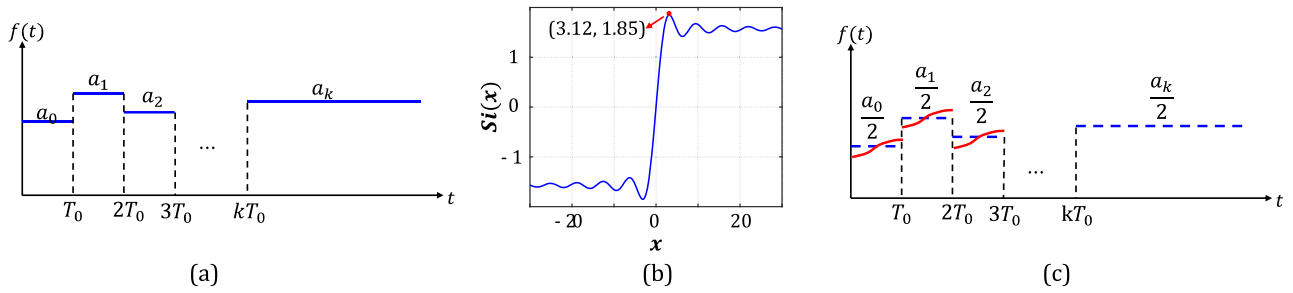


Fig. 4. (a) Typical signal amplitude changes during computation without filtering. (b) Plot of the sine integral function; (c) typical signal amplitude changes during computation after filtering.

When the filter BW W is very narrow (\sim MHz) compared to the updating period T_0 (\sim ns), the variation term is close to 0, leaving the DC term only. Otherwise, the filtered signal amplitude increases gradually in each iteration, which looks similar to what is shown in red in Fig. 4(c). The red line crosses half of the DC values in each iteration, which can be used to extract the inversion results by multiplying a scaling factor. In this sense, the optical filter treats the signal as a DC signal within each iteration and solely suppresses out-of-band ASE noises.

Note that propagation through the BPF in multiple iterations is equivalent to cascading multiple BPFs, leading to the BW narrow-down effect. Assuming a Lorentzian spectral shape and a single-stage 3 dB filter BW of B_0 (Hz), the effective 3 dB filter BW after k ($k \geq 1$) stages is $B_k = \sqrt{2^{1/k} - 1} \cdot B_0$. The arrayed waveguide gratings (AWGs) are modeled as ILs, which are included in the IL of BPFs. The single-stage ASE noise after a BPF and the total ASE noise power after k stages of SOAs and BPFs are shown in Eqs. (10a), (10b). β (dB) is the IL of a BPF. G (dB) and NF (dB) are the gain and noise figures of a single-stage SOA. h is the Planck constant. f is the signal frequency. The ASE noise is expressed as a complex-valued signal $n_{\text{ASE}} = n_r + jn_i$. The real and imaginary parts n_r and n_i independently obey Gaussian distribution as $\sim \mathcal{N}(0, \sigma^2)$, where $\sigma^2 = \frac{1}{2}\sigma_{\text{ASE}}^2 = \frac{1}{2}P_{\text{ASE}}$. The output signal considering ASE noise is shown in Eq. (11). φ_{SOA} and φ_{BPF} are extra phase shift introduced by each SOA and BPF stage,

$$P_{\text{ASE1}} = 10^{\frac{\text{NF}}{10}} h f (10^{\frac{G}{10}} - 1), \quad (10a)$$

$$P_{\text{ASEk}} = 10^{\frac{\text{NF}}{10}} h f (10^{\frac{G}{10}} - 1) \cdot \left(\sum_{s=0}^{k-1} 10^{\frac{sG}{10}} \cdot \sqrt{2^{\frac{1}{2^{s+1}}} - 1} \right) B_1, \quad (10b)$$

$$\mathbf{E}_o = 10^{\frac{k(G-\beta)}{20}} e^{jk(\varphi_{\text{SOA}} + \varphi_{\text{BPF}})} \mathbf{E}_i + n_{\text{ASE}}. \quad (11)$$

6. Detection Block

Figure 3(f) shows the model of one unit of the *Detection* block, which extracts both real and imaginary parts of the computing results by homodyne detection. The computed signal and the reference split from the laser in the homodyne detection are assumed to be $\mathbf{E}_{\text{sig}} = A_1 e^{j(2\pi f_1 t + \varphi_1 + \theta(t))}$, and $\mathbf{E}_{\text{ref}} = A_2 e^{j(2\pi f_1 t + \varphi_2 + \theta(t))}$, respectively [$\theta(t)$ is the laser phase noise]. The differential photocurrents containing real and imaginary parts of the computation results are shown in Eqs. (12a), (12b).

$I_1 - I_4$ are photocurrents of the four photodiodes (PDs), $i_{\text{shot1}} - i_{\text{shot4}}$ are shot noise currents of the photodiodes, and $\mathcal{R} = e/(hf)$ is the responsivity of the PDs (e is the electron charge; the quantum efficiency is assumed to be 1). The effects of shot noise are modeled in Eqs. (13a), (13b). B_e is the BW of the low-pass electrical filter in the sampling circuits placed after balanced PDs, which is used to filter out the unwanted high-frequency components. In practice, the electronic filters are only used for output readout at convergence (corresponding to DC signals), instead of capturing its entire temporal trace, allowing the use of a narrow BPF. B_e is generally matched to be half of the BW of optical filters,

$$I_1 - I_2 = \mathcal{R} A_1 A_2 \cos(\varphi_1 - \varphi_2) + i_{\text{shot1}} - i_{\text{shot2}}, \quad (12a)$$

$$I_3 - I_4 = \mathcal{R} A_1 A_2 \sin(\varphi_1 - \varphi_2) + i_{\text{shot3}} - i_{\text{shot4}}, \quad (12b)$$

$$\sigma_r^2 = \sigma_1^2 + \sigma_2^2 = e\mathcal{R}(|\mathbf{E}_{\text{sig}}|^2 + |\mathbf{E}_{\text{ref}}|^2) B_e, \quad (13a)$$

$$\sigma_i^2 = \sigma_3^2 + \sigma_4^2 = e\mathcal{R}(|\mathbf{E}_{\text{sig}}|^2 + |\mathbf{E}_{\text{ref}}|^2) B_e. \quad (13b)$$

Another major noise source during detection is the thermal noise of TIAs after the balanced PDs. Thermal noise power can be expressed in the current form, as shown in Eq. (14),

$$i_{\text{thermal}}^2 = 4kTB_e R^{-1}, \quad (14)$$

where k is the Boltzmann constant. T is room temperature. B_e is the BW of the electrical sampling circuits. R is the resistance of the TIA. The signal-to-noise-ratio (SNR) is expressed in Eq. (15). $P_{\text{sig}} = |\mathbf{E}_{\text{sig}}|^2$ and $P_{\text{ref}} = |\mathbf{E}_{\text{ref}}|^2$ are optical powers of the computed signal and the reference signal,

$$\text{SNR} = \frac{\mathcal{R}^2 P_{\text{sig}} P_{\text{ref}}}{e\mathcal{R}(P_{\text{sig}} + P_{\text{ref}}) B_e + 4kTB_e R^{-1}}. \quad (15)$$

7. Electronic Peripherals

Figure 3(g) shows the model of the electronic peripherals, including DACs and ADCs. The bit resolutions and the power consumption of DACs and ADCs are two characteristics that influence the performances of the photonic RIP. The models show the conversion of an n -bit digital signal to an analog voltage or current signal and the conversion of an analog signal to an n -bit digital signal. The finite resolutions of DACs and ADCs lead to the quantization errors in the process of loading matrix elements to the *Weight Bank* block and sampling of the

computation results from the *Detection* block, which degrade the inversion accuracy.

C. Implementations on Photonic Integration Platforms

Major photonic integration platforms include indium phosphide (InP) [19], silicon photonics (SiPh) or silicon-on-insulator (SOI) [30], and silicon nitride (Si_3N_4) [31]. InP is capable of integrating both active and passive components. SiPh is compatible with CMOS fabrication, which saves cost. The high refractive index contrast in SiPh allows the design to be compact. The Si_3N_4 platform provides low propagation losses and a wide transparency window. Since the proposed design includes both passive and III-V active components, hybrid integration or heteroepitaxy integration is needed for silicon-based platforms to incorporate active blocks. Hybrid integration methods include flip-chip bonding, wafer/die bonding, and microtransfer printing (μTP). A comparison of various III-V-on-Si methods is summarized in Table 4 [18,20,32,33]. Albeit current InP platforms are capable of cointegration of passive and active components, the relatively low refractive index contrast, high propagation losses, and the high cost are the limiting factors to the use of this platform. Recently, the newly developed InP-based platform, InP membrane on silicon (IMOS), has demonstrated a small footprint by increasing index contrast through the insertion of a low-index buffer layer [34]. It is promising for the implementation of a compact monolithic integrated photonic processor. Figure 5 shows an example of a 4×4 monolithically integrated RIP, where the laser diodes (LDs), SOAs, and balanced photodetectors (BPDs) are all integrated on-chip.

4. PERFORMANCE ANALYSIS

In this section, numerical methods are used to characterize the system-level performance of the proposed photonic RIP based on the models built in Section 3. Four major performance metrics, including speed, power efficiency, accuracy, and scalability are investigated.

A. Speed

Processing speed is determined by the processing time of the photonic RIP, which consists of three parts: (1) matrix weights loading time, (2) net computation time, and (3) post-processing time.

1. Matrix Weights Loading Time

An $N \times N$ photonic RIP requires loading N^2 weights to the MZI units, which can be realized by using multichannel

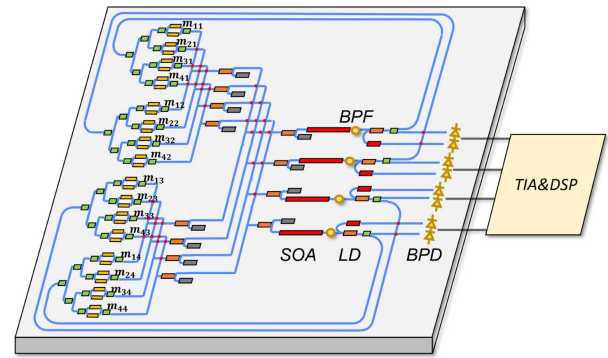


Fig. 5. Conceptual figure of an integrated 4×4 inverter (without wavelength multiplexing) where the LDs, BPFs, SOAs, and BPDs are monolithically integrated on-chip. TIAs and digital signal processing (DSP) are used for results readout. One column of the inverse matrix can be computed at a time by turning on one of the LDs, while the complete computation results can be obtained by turning on each of the LDs, respectively, or using multiple copies of the unit shown here.

DACs. Though TOPSs are used in the previous modeling, high-speed weights loading (~ 50 GHz [35]) can be realized by using electro-optic phase shifters. Once loaded, matrix weights are kept constant for multiple iterations during the entire computation process.

2. Net Computation Time

The net computation time of the photonic RIP is solely determined by the signal propagation time in the recirculating loop. A compact design reduces the propagation time in one iteration, while matrix M , with a smaller spectral radius, guarantees fewer iterations for convergence. Table 5 summarizes the estimated length of each block on mainstream photonic integration platforms for an $N \times N$ photonic RIP. The length of the *Laser* block is not included, since it can be placed near the input port of *Summation 1* block, which makes the input propagation time negligible. The length of the *Detection* block is not included in the simulation, since it is only used at the final iteration and is negligible compared to other parts. The length of the looped-back line is estimated to be the same as forward propagation length.

The estimated net inversion rate and processing rate are shown in Figs. 6(a) and 6(b). The inversion rate is measured by the number of matrix inversions implemented in a second,

Table 4. Comparison of III-V-on-Si Integration Methods

Method	Flip-Chip Bonding	Wafer/Die Bonding	μTP	Hetero-epitaxy
Integration density	Low	Medium	High	High
Efficiency of III-V material use	Medium	Medium	High	Very High
Alignment accuracy	High	High	Medium	High
Throughput	Medium	High	High	High
Cost	High	Medium	Low	Low
Maturity	Mature	Mature	R&D	R&D

Table 5. Length Estimation of an $N \times N$ Iterative Photonic Processor on Photonic Integration Platforms

Component	SOI (μm)	Si_3N_4 (μm)	IMOS (μm)
Summation 1	20	240	47
Input Vectors	$(2N-1) \cdot 72$	$(2N-1) \cdot 180$	$(2N-1) \cdot 80$
Fan-out			
Weight Bank	100	1100	200
Summation 2	$(2N-1) \cdot 90$	$(2N-1) \cdot 300$	$(2N-1) \cdot 120$
Amplification	$2.2 \log_2 N \cdot 176$ [36]	$2.2 \log_2 N \cdot 246$ [37]	$2.2 \log_2 N \cdot 63$ [38]
Filtering	128	130	200

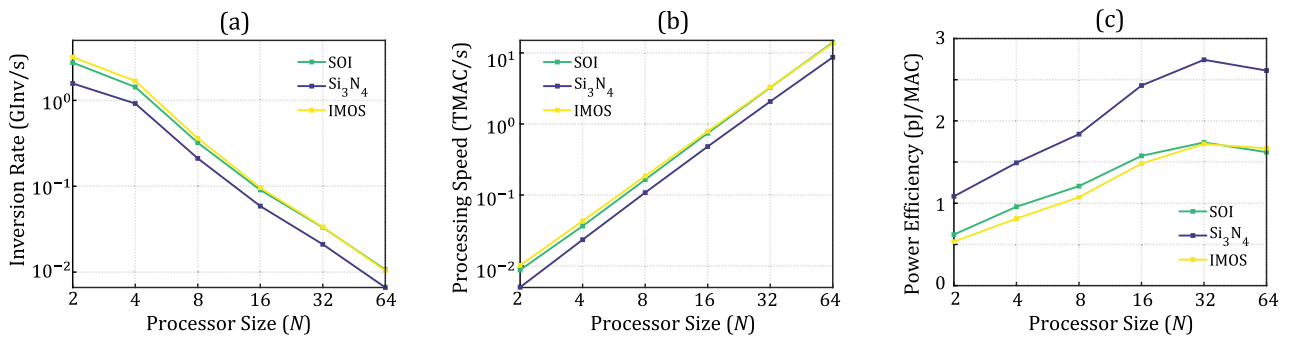


Fig. 6. (a), (b) Net computing speed of different-sized $N \times N$ photonic RIPs on SOI, Si₃N₄, and IMOS platforms. The light propagation speed is estimated through the effective indices of the waveguides, while the computing speed is estimated considering light propagation speed, loop length, and number of iterations simultaneously. (a) Inversion rate in terms of GInv/s and (b) processing speed in terms of TMAC/s are shown. (c) Power efficiency of different-sized $N \times N$ photonic RIPs.

which decreases with the matrix size because a larger-sized processor has a longer loop length and requires more iterations for convergence. An inversion rate of 3.2 GInv/s (10^9 inversions/s) can be reached for a 2×2 processor, whereas for a 64×64 processor, the inversion rate becomes 10.7 MInv/s (10^6 inversions/s). Platforms with a higher refractive index contrast and effective index enable a higher inversion rate due to more compact designs.

The photonic processing rate is also shown here as a reference. It is measured by the number of multiply-accumulated operations per second [MAC/s, ($a' \leftarrow a + (b \times c)$, one real-valued multiplication and one real-valued addition in digital electronic computers)]. For an $N \times N$ processor, the proposed photonic RIP computes $MX^{(k)} + \omega I_N$ in each iteration, which is equivalent to around N^3 MAC (extra N additions for ωI_N can be neglected compared to operations in matrix multiplication). The photonic processing rates, N^3 MAC/s (time required for a half iteration), are shown in Fig. 6(b). Though larger-sized processors have a slower inversion rate, they implement many more operations than smaller-sized matrices. The processing speed increases with the matrix size, with 1.1 TMAC/s (10^{12} MAC/s) for 2×2 processors and 1.4 PMAC/s (10^{15} MAC/s) for 64×64 processors. One complex MAC/s on photonic iterative photonic processor is equivalent to four real MAC/s on electronic processors, providing >180 times faster rate in terms of FLOP/s (1 real-valued MAC/s = 2 FLOP/s) than the 4×4 NVIDIA GPU accelerator reported recently [39].

The processing rate of electronic processors is limited by the internal clock rate. The peak clock frequency for CPU, GPU, and FPGA are 8.7 [40], 3.2 [41], and 1.6 GHz [42], respectively. Inverting an $N \times N$ matrix requires $\sim O(N^3)$ operations, corresponding to an inversion rate of only around 0.13 MInv/s for a 64×64 electronic processor using the RICH method, indicating the proposed photonic RIP is >80 times faster than electronic processors.

3. Postprocessing Time

Coherent detection together with postprocessing is implemented once at the final iteration. The speeds of PDs and ADCs need to satisfy the highest inversion rate shown in Fig. 6(a) (3.2 GHz), which is not a limiting factor since

high-speed PDs (32 Gb/s [43]) and ADCs (32 GSPS [44]) have been demonstrated.

B. Power Efficiency

Power consumption in the proposed photonic processor mainly comes from the active components, including lasers, TOPSs and SOAs, and electronic peripheral circuits. Table 6 summarizes the required numbers of power-consuming components for an $N \times N$ processor and their typical power consumption. While TOPSs are used in the analysis, their power efficiency can be further improved by using nonvolatile materials [49]. The number of SOA stages, x , is further explored in Fig. 7(g). The simulated power efficiency of an $N \times N$ iterative photonic processor is shown in Fig. 6(c), which is defined by power (W)/(MAC/s). It is worth noting that, while the total power consumed by larger-sized processors is higher, their larger number of operations evidently lowers the overall power efficiency. The power efficiency of 2×2 to 64×64 processors is estimated to be 0.53, 0.81, 1.07, 1.48, 1.72, and 1.62 pJ/MAC, respectively. Compared with the state-of-the-art electronic processor that features 31.9 pJ/MAC [50], the proposed photonic solution can easily be over 10 times more energy-efficient.

C. Accuracy

Inversion accuracy measures how accurate the computed matrix inverse is for the iterative photonic processor, which is defined as $1 - \varepsilon$, where ε is the relative error given as $\varepsilon = (||Y_{\text{sim}} - Y_{\text{ideal}}|| / ||Y_{\text{ideal}}||) \times 100\%$ (Y_{sim} is the simulated inversion results, and Y_{ideal} is the theoretical inversion results). Three key error sources include: 1) quantization error, 2) ASE noise introduced during amplification, and 3) thermal/shot

Table 6. Power Estimation of an $N \times N$ Iterative Photonic Processor

Component	Number	Unit Power (mW)	Total Power (mW)
Laser	N	69 [45]	$69N$
TOPS	$2N^2$	0.49 [46]	$0.98N^2$
SOA	xN^{2a}	50 [36]	$50xN^2$
DAC	$2N^2$	0.045 [47]	$0.09N^2$
ADC	$2N^2$	0.46 [48]	$0.92N^2$

^a $x = 2, 4, 6, 9, 11, 11$ according to Fig. 7(g).

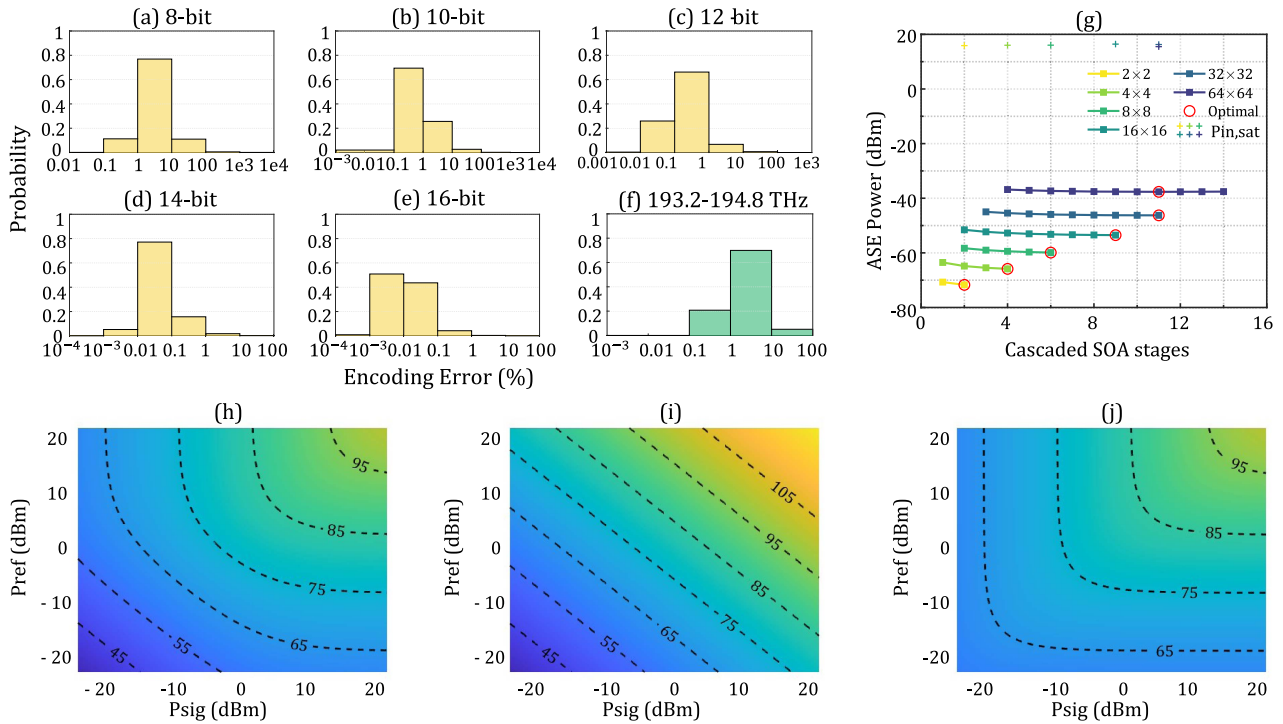


Fig. 7. Matrix weights encoding error for (a)–(e) different DAC bit resolutions and (f) 20 nm wavelength span. Using a 16-bit DAC is enough to guarantee $<0.1\%$ relative weight encoding error. The encoding error due to wavelength multiplexing is around 3%. (g) ASE noise powers of different-sized processors when cascading different numbers of SOA stages. Red circles highlight the minimal achievable ASE powers for different-sized processors. $P_{in,sat}$ at optimal stages of different-sized processors are indicated by the “+” sign. SNR of coherent detection when (h) both thermal noise and shot noise are considered, (i) only thermal noise is considered, and (j) only shot noise is considered. Thermal noise is dominant when signal power is low, while shot noise is dominant when signal power is high.

noise introduced during detection. The three error sources are first analyzed respectively, followed by a system-level analysis.

1. Quantization Error

The digital and analog conversions during computation introduce quantization errors. The error of DAC is introduced in the *Weight Bank* block, while the error of ADC is introduced in the readout stage.

The *Weight Bank* block used to encode matrix weights suffers from deviations from ideal values due to two reasons. 1) The finite bit resolution of DACs limits the accuracy of the phase shift applied to two arms of the MZI unit, which introduces quantization noises in the encoded matrix weights. This can be mitigated by using high-resolution DACs. 2) Weight encoding deviations due to deviations in phase encoding among different wavelength channels [$\Delta\phi_q(f) = \frac{f}{f_c} \Delta\phi_q(f_c)$, f_c is the center wavelength, and $\Delta\phi_q$ ($q = 1, 2$) represents the phase shifts of upper and lower arms]. Independent impacts of 1) and 2) on the accuracy of weights encoding (defined as $1 - \frac{|m_{encoded} - m_{ideal}|}{|m_{ideal}|} \times 100\%$) are numerically simulated and shown in Figs. 7(a)–7(f). 10,000 randomly generated matrix elements are simulated. According to Figs. 7(a)–7(f), a DAC with a resolution of 16 bits is enough to guarantee $<0.1\%$ relative weight-encoding error. The weight-encoding error caused by wavelength multiplexing spanning 20 nm is around 3%.

There exists an output resolution limitation in analog computation, since the resolution power has to be higher than the shot noise limit to avoid accuracy loss in the readout stage [51].

The allowed output bit resolution is $B_{out} = \frac{1}{2} \log_2 \left(\frac{PT}{hc/\lambda} \right)$, where P (W) is the full-scale optical power (corresponds to the real-valued unit “1”), and T (s) is the data updating period (time taken for one iteration). The permitted range of full-scale optical power is determined by the processor size and the SOA saturation power, while the data updating rate is decided by the loop length. These effects are discussed in the following sections and included in the accuracy analysis.

2. Noise Introduced during Amplification

SOAs are used to compensate for on-chip losses to ensure a lossless operation. On-chip losses for 2×2 to 64×64 processors are estimated to be 7.4, 14.3, 21.3, 28.5, 36.5, and 45.1 dB, respectively. Main loss sources include the 3 dB power loss of MMI couplers, IL of MMI couplers, BPFs, waveguide crossings, and waveguide propagation.

The ASE noise power is calculated as a function of the cascaded stage number, k , according to Eq. (10b) and is shown in Fig. 7(g). Each SOA is assumed to have 3.8 dB noise figure, 15 dB maximal gain, 19.6 dBm output saturation power, and 120 nm 3 dB gain BW in the numerical simulations [52]. It should be noted that there exists an optimal number of SOA stages that generates the least ASE power for each processor size,

as indicated by the red circle in Fig. 7(g). The smallest ASE powers are estimated to be -71.7 , -65.8 , -59.9 , -53.5 , -46.2 , and -37.6 dBm for 2×2 to 64×64 processors, respectively. The corresponding single stage gains are 3.7, 3.6, 3.6, 3.2, 3.3, and 4.1 dB, respectively.

The ASE power is normalized to input signal power and then added to the signal in accuracy analysis. The maximum input signal power is limited by nonlinear effects in the SOA when operating in the saturation region. The minimal SOA input signal power is limited by the optical SNR and will influence the inversion accuracy. An input signal power of at least 20 dB higher than the ASE power is chosen in the performance analyses.

3. Noise Introduced during Detection

A homodyne coherent detection scheme is chosen to extract the computation results in the photonic RIP. As mentioned before, shot noise from PDs and thermal noise from TIAs are two main noise sources of coherent detection. Based on Eq. (15), the SNRs (dB) for different signal and reference powers are

Table 7. Parameters Used in Accuracy Analyses of the Iterative Photonic Processor

Parameter	Value
Processor size	$2 \times 2 - 64 \times 64$
Number of random matrix instances	500/processor size
Half-wave voltage of MZI	4.36 V [53]
DAC resolution	16 bits
SOA NF	3.8 dB [52]
BW of the optical BPF	64.5 MHz [54]
IL of the optical BPF	0.2 dB [54]
IL of an MMI coupler	0.2 dB [29]
IL of a waveguide crossing	0.019 dB [55]
Center frequency	193.6 THz
WDM channel spacing	0.1 nm [28]
Electron charge	1.6×10^{-19} C
Planck's constant	6.626×10^{-34} J · s
BW of the electronic filter	32.25 MHz
Boltzmann constant	1.38×10^{-23} J/K
Temperature	300 K
Electronic resistance	50 Ω

simulated and shown in Figs. 7(h)–7(j). The lower limit of signal power is determined by the output signal resolution, while the upper limit of the signal power is determined by the SOA output saturation power. When both signal power and reference power are low, thermal noise is the dominant noise source, while shot noise is dominant when signal power and reference power are high.

4. System-Level Analysis

In this part, randomly generated matrices with sizes ranging from 2×2 to 64×64 are inverted to show the overall inversion accuracy of the photonic RIP, including the three main error sources mentioned above. In our simulation, $\rho(\mathbf{I}_N - \omega\mathbf{A})$ is limited to <0.99 for a reasonable simulation time. Preconditioning can be applied to matrices with large spectral radius (0.99–1) for faster convergence rates [9].

The dependence between inversion accuracy and input signal power is assessed and shown in Fig. 8(a) (parameters used in the simulations are listed in Table 7; the input power and reference power are set to be equal). Note that the input signal power corresponds to the real-valued unit “1” in the computation. The solid lines represent the inversion accuracy when a single wavelength processor and multiple copies of the unit are used. The inversion accuracy increases with the input signal power first linearly and then at a saturated rate. The number of iterations taken before convergence is also shown in blue text next to the figure legends in Fig. 8(a). The larger numbers of eigenvalues for larger-sized processors lead to a larger $\rho(\mathbf{I}_N - \omega\mathbf{A})$, corresponding to a slower convergence rate (larger iterations for convergence). The red dashed lines represent the input signal power range when wavelength multiplexing is used and the corresponding inversion accuracy for different-sized processors. It can be seen that an accuracy of $>98\%$ can be reached for a 64×64 processor when a single wavelength is used, while the accuracy slightly drops to $>90\%$ when wavelength multiplexing is used. The use of wavelength multiplexing technique is limited when the matrix size becomes large. This is because the allowable maximal input signal power decreases when increasing the number of multiplexed wavelengths, leading to a decrease in the inversion accuracy.

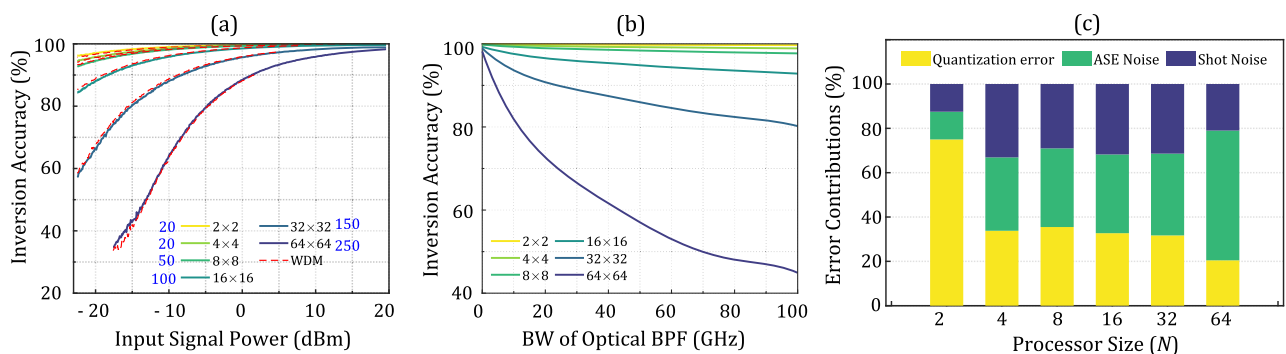


Fig. 8. (a) Inversion accuracy of different-sized photonic RIPs when input signal powers are different (optical filter BW = 64.5 MHz). Values in blue indicate the required iteration numbers for convergence. High-input signal power (>1 dBm) is necessary for ensuring an accuracy of $>90\%$ when using wavelength multiplexing technique. (b) Fitted relationship between inversion accuracy and optical filter BW (input signal power is 16.6 dBm) for processor size ranging from 2×2 to 64×64 ; (c) error breakdown of different-sized photonic RIPs (input signal power is 16.6 dBm).

The inversion accuracy for processors larger than 64×64 is estimated to drop evidently, which is due to the heavily accumulated ASE noise and the limited input signal power. We believe it is sensible to apply the block-wise inversion method for inverting matrices larger than 64×64 . Details are introduced in Section 4.D.

The BW of the optical filter determines the ASE noise power and in turn the inversion accuracy. The relationship between inversion accuracy and the filter BW is shown in Fig. 8(b) (input signal power @16.6 dBm; single wavelength). For processors $\leq 32 \times 32$, applying a filter with BW of smaller than 20 GHz guarantees an inversion accuracy of $>90\%$. The BW, however, needs to drop to below 2 GHz for a 64×64 processor to ensure an accuracy of $>90\%$. The error breakdown for processors with port counts of 2×2 to 64×64 (with input signal power at 16.6 dBm) is shown in Fig. 8(c). The quantization error dominates when matrix size is small, while ASE noise becomes dominant when matrix size becomes large.

D. Scalability

The proposed iterative photonic processor shows good scalabilities in terms of accuracy, speed, and power efficiency. A larger-sized processor requires a longer computation time, while the power efficiency remains similar to that of a smaller-sized processor. The proposed iterative photonic processor is capable of inverting a 64×64 matrix with an accuracy of $>98\%$ and an inversion rate of 10.7 MInv/s. Considering that the typical diameter of InP wafers is 100 mm, the device footprint could be a limiting factor to its scalability. However, the heterogeneous integration schemes on SOI wafers can easily break the limitation (wafer diameter is up to 300 mm) [16]. Additionally, the induced ASE noise plays a more critical role in the scale-up of processors, and thus we propose the use of block-wise inversion techniques for larger-sized processors as explained in the following.

In principle, the iterative photonic processor can invert arbitrarily large matrices by using block-wise inversion techniques. If a $(P + Q) \times (P + Q)$ complex-valued matrix \mathbf{M} is partitioned into block form $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$, where $\mathbf{A} \in \mathbb{C}^{P \times P}$, $\mathbf{B} \in \mathbb{C}^{P \times Q}$, $\mathbf{C} \in \mathbb{C}^{Q \times P}$, and $\mathbf{D} \in \mathbb{C}^{Q \times Q}$, then the inverse of \mathbf{M} can be computed in four blocks as shown in Eq. (16),

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}. \quad (16)$$

Referring back to Section 3, the core of the iterative photonic processor is a matrix-vector multiplier. By adding modulators between the input lasers and the MMI couplers in *Summation 1* block, the processor is enabled to support arbitrary complex-valued input vectors. It performs matrix multiplication by running only one iteration if an additional optical switch is added at the readout stage, while by sending in a unit input and modulating the input vector in the second iteration, the processor is configured to compute matrix addition and subtraction. Therefore, the proposed processor can be programmed to compute matrix addition, subtraction, multiplication, and inversion. It is thus capable of solving Eq. (16) by

computing $(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$ first and then implementing matrix multiplications and additions separately for each block. The inverse of the large matrix \mathbf{M} is then obtained by combining the computation results of four blocks.

5. CONCLUSION

An $N \times N$ photonic integrated RIP is proposed for the first time in this paper for fast computations of complex-valued matrix inversions. The RICH method is chosen as the inversion algorithm due to its easy implementations and fewer constraints on the matrix to be inverted. A Min-Max algorithm is proposed to choose the optimal damping parameter ω of the RICH method for the highest convergence rate, where the maximum distance between the scaled eigenvalues and point (1,0) is minimized.

System architectures and modeling are presented for the photonic RIP, followed by a discussion of implementations on mainstream photonic integration platforms. The integration of III-V gain components enables a lossless scalable design, which is necessary for the iterative computations based on the RICH method. Wavelength multiplexing can be used to significantly improve the processing efficiency of a single processor, allowing for the computation of multiple columns of the inverse matrix in a single processor. The accurate phase control provided by integrated platforms realizes the phase-sensitive design of the proposed processor.

Performance analyses are also carried out for key photonic building blocks, followed by a system-level analysis including speed, power efficiency, accuracy, and scalability of the processor. The predicted performances can be used as a guideline for designing an $N \times N$ processor. Simulation results show the feasibility of an iterative photonic processor up to size 64×64 with an inversion accuracy of $>98\%$ and >80 times faster inversion rate than electronic computers. The proposed photonic processor is estimated to be over an order of magnitude more energy-efficient than electronic processors.

The proposed photonic integrated RIP serves as a promising solution for future systems that require fast and highly parallel computations.

Funding. Huawei Technologies (Sweden) AB (G107576).

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

REFERENCES

1. H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, and F. Rusek, "Hardware efficient approximative matrix inversion for linear pre-coding in massive MIMO," in *IEEE International Symposium on Circuits and Systems (ISCAS)* (2014), pp. 1700–1703.
2. F. A. Monteiro, F. Rosário, and A. Rodrigues, "Fast matrix inversion updates for massive MIMO detection and precoding," *IEEE Signal Process. Lett.* **23**, 75–79 (2016).
3. C. Tang, C. Liu, L. Yuan, and Z. Xing, "High precision low complexity matrix inversion based on Newton iteration for data detection in the massive MIMO," *IEEE Commun. Lett.* **20**, 490–493 (2016).

4. C. Zhang, X. Liang, Z. Wu, F. Wang, S. Zhang, Z. Zhang, and X. You, "On the low-complexity, hardware-friendly tridiagonal matrix inversion for correlated massive MIMO systems," *IEEE Trans. Veh. Technol.* **68**, 6272–6285 (2019).
5. S. Hashima and O. Muta, "Fast matrix inversion methods based on Chebyshev and Newton iterations for zero forcing precoding in massive MIMO systems," *EURASIP J. Wireless Commun. Netw.* **2020**, 34 (2020).
6. X.-W. Zhang, L. Zuo, M. Li, and J.-X. Guo, "High-throughput FPGA implementation of matrix inversion for control systems," *IEEE Trans. Ind. Electron.* **68**, 6205–6216 (2021).
7. B. Fischer and J. Modersitzki, "Fast inversion of matrices arising in image processing," *Numerical Algorithms* **22**, 1–11 (1999).
8. D. F. G. Coelho, R. J. Cintra, A. C. Frery, and V. S. Dimitrov, "Fast matrix inversion and determinant computation for polarimetric synthetic aperture radar," *Comput. Geosci.* **119**, 109–114 (2018).
9. D. S. Watkins, *Fundamentals of Matrix Computations*, 2nd ed. (Wiley-Interscience, 2002).
10. J.-M. Muller, *Handbook of Floating-Point Arithmetic* (Springer, 2018).
11. H. Zhang, M. Gu, X. D. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. H. Yung, Y. Z. Shi, F. K. Muhammad, G. Q. Lo, X. S. Luo, B. Dong, D. L. Kwong, L. C. Kwek, and A. Q. Liu, "An optical neural chip for implementing complex-valued neural network," *Nat. Commun.* **12**, 457 (2021).
12. Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. Carloni, and K. Bergman, "Silicon photonics codesign for deep learning," *Proc. IEEE* **108**, 1261–1282 (2020).
13. R. Athale and D. Psaltis, "Optical computing: past and future," *Opt. Photon. News* **27**, 32–39 (2016).
14. H. Rajbenbach, Y. Fainman, and S. H. Lee, "Optical implementation of an iterative algorithm for matrix inversion," *Appl. Opt.* **26**, 1024–1031 (1987).
15. K. Wu, C. Soci, P. P. Shum, and N. I. Zheludev, "Computing matrix inversion with optical networks," *Opt. Express* **22**, 295–304 (2014).
16. D. Thomson, A. Zilkie, J. Bowers, T. Komljenovic, G. Reed, L. Vivien, D. Marris-Morini, E. Cassan, L. Viro, J. Fédéli, J. Hartmann, J. Schmid, D.-X. Xu, F. Boeuf, O. Peter, B. G. Mashanovich, and M. Nedeljkovic, "Roadmap on silicon photonics," *J. Opt.* **18**, 073003 (2016).
17. D. J. Blumenthal, R. Heideman, D. Geuzebroek, A. Leinse, and C. Roeloffzen, "Silicon nitride in silicon photonics," *Proc. IEEE* **106**, 2209–2231 (2018).
18. A. Y. Liu and J. Bowers, "Photonic integration with epitaxial III–V on silicon," *IEEE J. Sel. Top. Quantum Electron.* **24**, 6000412 (2018).
19. M. Smit, K. Williams, and J. van der Tol, "Past, present, and future of InP-based photonic integration," *APL Photon.* **4**, 050901 (2019).
20. J. Zhang, G. Muliuk, J. Juvert, S. Kumari, S. Kumari, J. Goyvaerts, B. Haq, C. Op de Beeck, B. Kuyken, G. Morthier, D. Van Thourhout, R. Baets, G. Lepage, P. Verheyen, J. Van Campenhout, A. Gocalinska, J. O'Callaghan, E. Pelucchi, K. Thomas, B. Corbett, A. J. Trindade, and G. Roelkens, "III-V-on-Si photonic integrated circuits realized using micro-transfer-printing," *APL Photon.* **4**, 110803 (2019).
21. J. M. Ramirez, "III-V-on-silicon integration: from hybrid devices to heterogeneous photonic integrated circuits," *IEEE J. Sel. Top. Quantum Electron.* **26**, 6100213 (2020).
22. "Barata and Hussein—2012—the Moore–Penrose pseudoinverse a tutorial review.pdf," 2021, <https://link.springer.com/content/pdf/10.1007/s13538-011-0052-z.pdf>.
23. X. Li, S. Wang, and Y. Cai, "Tutorial: complexity analysis of singular value decomposition and its variants," arXiv:1906.12085 (2019).
24. "An introduction to the conjugate gradient method without the agonizing pain," (2021), <https://www.cs.cmu.edu/quake-papers/painless-conjugate-gradient.pdf>.
25. W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations* (Springer, 1994).
26. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (Society for Industrial and Applied Mathematics, 2003).
27. T. Alexoudi, G. T. Kanellos, and N. Pleros, "Optical RAM and integrated optical memories: a survey," *Light Sci. Appl.* **9**, 91 (2020).
28. "Spectral grids for WDM applications: DWDM frequency grid," 2021, <https://www.itu.int/rec/T-REC-G.694.1-202010-l/en>.
29. A. Maese-Novo, R. Halir, S. Romero-García, D. Pérez-Galacho, L. Zavargo-Peche, A. Ortega-Moñux, I. Molina-Fernández, J. G. Wangüemert-Pérez, and P. Cheben, "Wavelength independent multi-mode interference coupler," *Opt. Express* **21**, 7033–7040 (2013).
30. D.-X. Xu, J. H. Schmid, G. T. Reed, G. Z. Mashanovich, D. J. Thomson, M. Nedeljkovic, X. Chen, D. Van Thourhout, S. Keyvaninia, and S. K. Selvaraja, "Silicon photonic integration platform—have we found the sweet spot?" *IEEE J. Sel. Top. Quantum Electron.* **20**, 198–205 (2014).
31. K. Wörhoff, R. G. Heideman, A. Leinse, and M. Hoekman, "TriPleX: a versatile dielectric photonic platform," *Adv. Opt. Technol.* **4**, 189–207 (2015).
32. "III-V/Si photonics by die-to-wafer bonding | Elsevier enhanced reader," 2021, <https://reader.elsevier.com/reader/sd/pii/S1369702107701785?token=6CD3314CE12A4A68711516FBBBE65E8C9F21FE7EA366E547060B35AE60D2975659E11B439B06C870F93EAAFE8B84899F&originRegion=eu-west-1&originCreation=20210727120307>.
33. T. Matsumoto, T. Kurahashi, R. Konoike, K. Suzuki, K. Tanizawa, A. Uetake, K. Takabayashi, K. Ikeda, H. Kawashima, S. Akiyama, and S. Sekiguchi, "Hybrid integration of SOA on silicon photonics platform based on flip-chip bonding," *J. Lightwave Technol.* **37**, 307–313 (2019).
34. Y. Jiao, J. van der Tol, V. Pogoretskii, J. van Engelen, A. A. Kashi, S. Reniers, Y. Wang, X. Zhao, W. Yao, T. Liu, F. Pagliano, and A. Fiore, "Indium phosphide membrane nanophotonic integrated circuits on silicon," *Phys. Status Solidi A* **217**, 1900606 (2020).
35. D. Patel, A. Samani, V. Veerasubramanian, S. Ghosh, and D. Plant, "Silicon photonic segmented modulator-based electro-optic DAC for 100 Gb/s PAM-4 generation," *IEEE Photon. Technol. Lett.* **27**, 2433–2436 (2015).
36. B. Haq, S. Kumari, K. Van Gasse, J. Zhang, A. Gocalinska, E. Pelucchi, B. Corbett, and G. Roelkens, "Micro-transfer-printed III-V-on-silicon C-band semiconductor optical amplifiers," *Laser Photon. Rev.* **14**, 1900364 (2020).
37. C. Op de Beeck, B. Haq, L. Elsinger, A. Gocalinska, E. Pelucchi, B. Corbett, G. Roelkens, and B. Kuyken, "Heterogeneous III-V on silicon nitride amplifiers and lasers via microtransfer printing," *Optica* **7**, 386–393 (2020).
38. Y. Jiao, D. Heiss, L. Shen, S. P. Bhat, M. K. Smit, and J. J. G. M. van der Tol, "Design and fabrication technology for a twin-guide SOA concept on InP membranes," in *Proceedings of the 17th European Conference on Integrated Optics and Technical Exhibition, 19th Microoptics Conference (ECIO-MOC)* (2014), paper P030.
39. S. Markidis, S. W. Der Chien, E. Laure, I. B. Peng, and J. S. Vetter, "NVIDIA tensor core programmability, performance amp; precision," in *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (2018), pp. 522–531.
40. HWBOT, "CPU frequency overclocking records @ HWBOT," 2021, https://hwbot.org/benchmark/cpu_frequency/halloffame.
41. VideoCardz, 2021, <https://videocardz.com/amd/radeon-rx-6000/radeon-rx-6900-xt>.
42. "Virtex-6 family overview (DS150)," 2015, <https://datasheet.octopart.com/XC6VLX130T-2FFG784C-Xilinx-datasheet-10027064.pdf>.
43. S. Mauthe, Y. Baumgartner, M. Sousa, Q. Ding, M. D. Rossell, A. Schenk, L. Czornomaz, and K. E. Moselund, "High-speed III-V nanowire photodetector monolithically integrated on Si," *Nat. Commun.* **11**, 4565 (2020).
44. T. Kojima, H. Kiuchi, K. Uemizu, Y. Uzawa, M. Kroug, A. Gonzalez, T. Dippon, and T. Kageura, "Demonstration of a wideband submillimeter-wave low-noise receiver with 4–21 GHz IF output digitized by a high-speed 32 GSps ADC," *Astron. Astrophys.* **640**, L9 (2020).
45. J. R. Vaskasi, N. Singh, J. Van Kerrebrouck, J. Bauwelinck, G. Roelkens, and G. Morthier, "High wall-plug efficiency and narrow linewidth III-V-on-silicon C-band DFB laser diodes," *Opt. Express* **30**, 27983–27992 (2022).
46. Q. Fang, J. Song, T. Liow, H. Cai, M. Yu, G. Lo, and D. Kwong, "Ultralow power silicon photonics thermo-optic switch with suspended phase arms," *IEEE Photon. Technol. Lett.* **23**, 525–527 (2011).

47. O. Aiello, P. Crovetto, and M. Bruno, "Standard cell-based ultra-compact DACs in 40-nm CMOS," *IEEE Access* **7**, 126479 (2019).
48. P. Bisiaux, "A 14-bit 250 kS/s two-step inverter-based incremental $\Sigma\Delta$ ADC for CMOS image sensor in 0.18 μm technology," *Analog Integr. Circ. Sig. Process.* **97**, 427–435 (2018).
49. M. Wuttig, H. Bhaskaran, and T. Taubner, "Phase-change materials for non-volatile photonic applications," *Nat. Photonics* **11**, 465–476 (2017).
50. TOP500, 2022, <https://www.top500.org/lists/green500/2022/06/>.
51. M. A. Nahmias, T. F. De Lima, A. N. Tait, H. T. Peng, B. J. Shastri, and P. R. Prucnal, "Photonic multiply-accumulate operations for neural networks," *IEEE J. Sel. Top. Quantum Electron.* **26**, 7701518 (2020).
52. K. Morito, S. Tanaka, S. Tomabechei, and A. Kuramata, "A broad-band MQW semiconductor optical amplifier with high saturation output power and low noise figure," *IEEE Photon. Technol. Lett.* **17**, 974–976 (2005).
53. N. C. Harris, Y. Ma, J. Mower, T. Baehr-Jones, D. Englund, M. Hochberg, and C. Galland, "Efficient, compact and low loss thermo-optic phase shifter in silicon," *Opt. Express* **22**, 10487–10493 (2014).
54. A. Gondarenko, J. S. Levy, and M. Lipson, "High confinement micron-scale silicon nitride high Q ring resonator," *Opt. Express* **17**, 11366–11370 (2009).
55. Y. Ma, Y. Zhang, S. Yang, A. Novack, R. Ding, A. E.-J. Lim, G.-Q. Lo, T. Baehr-Jones, and M. Hochberg, "Ultralow loss single layer submicron silicon waveguide crossing for SOI optical interconnect," *Opt. Express* **21**, 29374–29382 (2013).