(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
03 September 2020 (03.09.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/176538 A1

(51) International Patent Classification:
G06F 17/10 (2006.01) *G06F 7/52* (2006.01)
G06F 17/16 (2006.01) *G06F 7/523* (2006.01)
G06F 5/01 (2006.01) *G06N 3/02* (2006.01)
G06F 7/32 (2006.01) *G06N 20/00* (2019.01)
G06F 7/44 (2006.01)

(21) International Application Number:
PCT/US2020/019747

(22) International Filing Date:
25 February 2020 (25.02.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/810,851 26 February 2019 (26.02.2019) US

(71) Applicant: **LIGHTMATTER, INC.** [US/US]; 60 State Street, 10th Floor, Boston, MA 02109 (US).

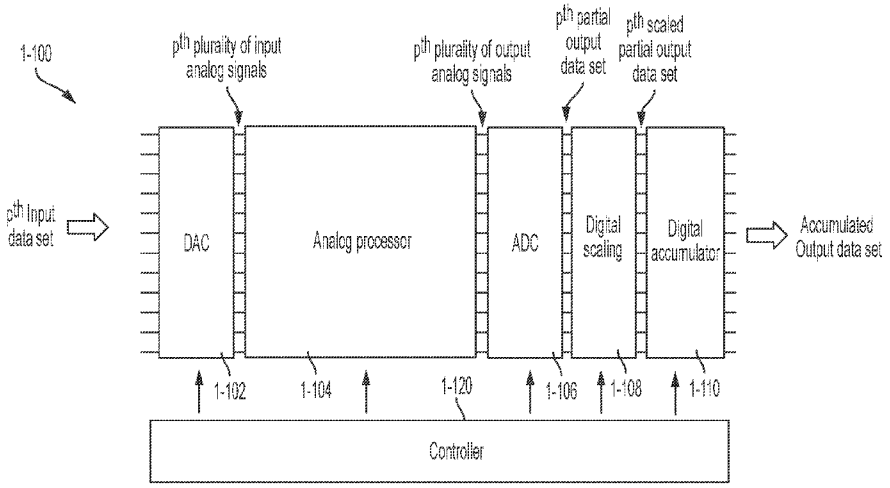
(72) Inventors: **KENNEY, Tyler, J.**; 25 Morrissey Boulevard, Unit 1537, Boston, MA 02125 (US). **FORSYTHE, Martin, B.z.**; 14 Dunster Road, Apt. 2, Jamaica Plain, MA 02130 (US). **LAZOVICH, Tomo**; 130 Cambridgepark Drive, Unit 254, Cambridge, MA 02140 (US). **BUNANDAR, Darius**; 130 Bowdoin St., Apt. 706, Boston, MA 02108 (US).

(74) Agent: **MORESCO, Michele** et al.; WOLF, GREENFIELD & SACKS, P.C., 600 Atlantic Avenue, Boston, MA 02210-2206 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(54) Title: HYBRID ANALOG-DIGITAL MATRIX PROCESSORS

FIG. 1-2A



(57) Abstract: Techniques for computing matrix operations for arbitrarily large matrices on a finite-sized hybrid analog-digital matrix processor are described. Techniques for gain adjustment in a finite- sized hybrid analog-digital matrix processor are described which enable the system to obtain higher energy efficiencies, greater physical density and improved numerical accuracy. In some embodiments, these techniques enable maximization of the predictive accuracy of a GEMM-based convolutional neural network using low-precision data representations.

WO 2020/176538 A1

[Continued on next page]

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

HYBRID ANALOG-DIGITAL MATRIX PROCESSORS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Patent Application Serial No. 62/810,851, entitled "GENERAL MATRIX MULTIPLICATION WITH SUB-ARRAY TILING AND MULTIPLE SCALING FOR HYBRID ANALOG-DIGITAL MATRIX PROCESSORS," filed on February 26, 2019 under Attorney Docket No. L0858.70011US00, which is hereby incorporated herein by reference in its entirety.

FIELD OF INVENTION

[0002] This disclosure relates hybrid analog-digital processors configured to perform mathematical matrix operations.

BACKGROUND

[0003] Deep learning, machine learning, latent-variable models, neural networks and other matrix-based differentiable programs are used to solve a variety of problems, including natural language processing and object recognition in images. Solving these problems with deep neural networks typically requires long processing times to perform the required computation. The most computationally intensive operations in solving these problems are often mathematical matrix operations, such as general matrix multiplication or multi-channel convolution. The conventional approach to speed up deep learning algorithms has been to develop specialized hardware architectures. This is because conventional computer processors, e.g., central processing units (CPUs), which are composed of circuits including hundreds of millions of transistors to implement logical gates on bits of information represented by electrical signals, are designed for general purpose computing and are therefore not optimized for the particular patterns of data movement and computation required by the algorithms that are used in deep learning and other matrix-based differentiable programs. One conventional example of specialized hardware for use in deep learning are graphics processing units (GPUs) having a highly parallel architecture that makes them more efficient than CPUs for performing image processing and graphical manipulations. After their development for graphics processing, GPUs were found to be more efficient than CPUs for other parallelizable algorithms, such as those used in neural networks and deep learning.

[0004] Deep learning using neural networks conventionally requires two stages: a training stage and an evaluation stage (sometimes referred to as “inference”). Before a deep learning

algorithm can be meaningfully executed on a processor, e.g., to classify an image or speech sample, during the evaluation stage, the neural network must first be trained. The training stage can be time consuming and requires intensive computation.

SUMMARY

[0005] Some embodiments relate to a hybrid analog-digital processor comprising: circuitry comprising an analog processor, wherein the circuitry is configured to perform a mathematical operation using a plurality of passes, wherein for each of the plurality of passes, the circuitry is configured to: determine one or more scaling factors for the pass based on a set of parameters representing a portion of a matrix; scale at least some parameters of the set of parameters based on the one or more scaling factors to produce a scaled set of parameters; program the analog processor based on the scaled set of parameters; generate a plurality of input analog signals based on an input data set; generate a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters; generate a partial output data set based on the plurality of output analog signals; and scale the partial output data set based on the one or more scaling factors to produce a scaled partial output data set, wherein the circuitry is further configured to generate an accumulated output data set by accumulating the scaled partial output data sets generated by at least two of the plurality of passes, wherein the accumulated output data set represents a result of the mathematical operation.

[0006] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the scaled set of parameters.

[0007] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a convolution based on the plurality of input analog signals and the scaled set of parameters.

[0008] In some embodiments, programming the analog processor based on the scaled set of parameters comprises setting respective gains or attenuations for a plurality of analog amplifiers or attenuators of the analog processor based on the scaled set of parameters.

[0009] In some embodiments, the analog processor comprises a photonic processor comprising a plurality of programmable photonic devices, and wherein programming the analog processor

based on the scaled set of parameters comprises setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters.

[0010] In some embodiments, the programmable photonic devices comprise Mach-Zehnder interferometers, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises: setting respective optical characteristics for the plurality of Mach-Zehnder interferometers based on the scaled set of parameters.

[0011] In some embodiments, the programmable photonic devices comprise optical resonators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises: setting respective optical characteristics for the plurality of optical resonators based on the scaled set of parameters.

[0012] In some embodiments, the circuitry comprises a plurality of digital-to-analog converters (DACs), and the plurality of DACs are configured to generate the plurality of input analog signals based on the input data set.

[0013] In some embodiments, the circuitry comprises a plurality of analog-to-digital converters (ADCs), and the plurality of ADCs are configured to generate the partial output data set based on the plurality of output analog signals.

[0014] In some embodiments, the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

[0015] In some embodiments, accumulating the scaled partial output data sets generated by at least two of the plurality of passes comprises: for at least some of the passes, summing the scaled partial output data set generated by the pass with the scaled partial output data set generated by a previous pass.

[0016] In some embodiments, programming the analog processor comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0017] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

[0018] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a

plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

[0019] In some embodiments, scaling at least some of the set of parameters based on the one or more scaling factors comprises dividing at least some of the set of parameters by the one or more scaling factors, and scaling the partial output data set based on the one or more scaling factors comprises multiplying the partial output data set by the one or more scaling factors.

[0020] In some embodiments, for each of the plurality of passes, the circuitry is further configured to determine the one or more scaling factors based on a tile of the matrix that is associated with the pass.

[0021] In some embodiments, for each of the plurality of passes, the circuitry is further configured to determine the one or more scaling factors based on one or more rows of the matrix.

[0022] In some embodiments, for each of the plurality of passes, the circuitry is further configured to determine the one or more scaling factors based on one or more singular values of the matrix.

[0023] In some embodiments, the one or more scaling factors comprise a first scaling factor and a second scaling factor, and scaling at least some parameters of the set of parameters based on the one or more scaling factors comprises: scaling a first subset of the set of parameters based on the first scaling factor, and scaling a second subset of the set of parameters based on the second scaling factor.

[0024] Some embodiments relate to a method for performing a mathematical operation using a plurality of passes through an analog processor, the method comprising: for each of the plurality of passes: determining one or more scaling factors for the pass based on a set of parameters representing a portion of a matrix; scaling at least some parameters of the set of parameters based on the one or more scaling factors to produce a scaled set of parameters; programming the analog processor based on the scaled set of parameters; generating a plurality of input analog signals based on an input data set; generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters; generating a partial output data set based on the plurality of output analog signals; and scaling the partial output data set based on the one or more scaling factors to produce a scaled partial output data set; and generating an accumulated output data set by accumulating the scaled partial output data sets generated by at least two of the plurality of passes, wherein the accumulated output data set represents a result of the mathematical operation.

[0025] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the scaled set of parameters.

[0026] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a convolution based on the plurality of input analog signals and the scaled set of parameters.

[0027] In some embodiments, programming the analog processor based on the scaled set of parameters comprises setting respective gains or attenuations for a plurality of analog amplifiers or attenuators of the analog processor based on the scaled set of parameters.

[0028] In some embodiments, the analog processor comprises a photonic processor comprising a plurality of programmable photonic devices, and wherein programming the analog processor based on the scaled set of parameters comprises setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters.

[0029] In some embodiments, the programmable photonic devices comprise Mach-Zehnder interferometers, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises: setting respective optical characteristics for the plurality of Mach-Zehnder interferometers based on the scaled set of parameters.

[0030] In some embodiments, the programmable photonic devices comprise optical resonators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises: setting respective optical characteristics for the plurality of optical resonators based on the scaled set of parameters.

[0031] In some embodiments, generating the plurality of input analog signals based on the input data set comprises providing the input data set as input to a plurality of digital-to-analog converters (DACs).

[0032] In some embodiments, generating the partial output data set based on the plurality of output analog signals the circuitry comprises providing the plurality of output analog signals as inputs to a plurality of analog-to-digital converters (ADCs).

[0033] In some embodiments, the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

[0034] In some embodiments, accumulating the scaled partial output data sets generated by at least two of the plurality of passes comprises: for at least some of the passes, summing the

scaled partial output data set generated by the pass with the scaled partial output data set generated by a previous pass.

[0035] In some embodiments, programming the analog processor comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0036] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

[0037] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

[0038] In some embodiments, scaling at least some of the set of parameters based on the one or more scaling factors comprises dividing at least some of the set of parameters by the one or more scaling factors, and scaling the partial output data set based on the one or more scaling factors comprises multiplying the partial output data set by the one or more scaling factors.

[0039] In some embodiments, determining the one or more scaling factors comprises determining the one or more scaling factors based on a tile of the matrix that is associated with the pass.

[0040] In some embodiments, determining the one or more scaling factors comprises determining the one or more scaling factors based on one or more rows of the matrix.

[0041] In some embodiments, determining the one or more scaling factors comprises determining the one or more scaling factor based on one or more singular values of the matrix.

[0042] In some embodiments, the one or more scaling factors comprise a first scaling factor and a second scaling factor, and scaling at least some parameters of the set of parameters based on the one or more scaling factors comprises: scaling a first subset of the set of parameters based on the first scaling factor, and scaling a second subset of the set of parameters based on the second scaling factor.

[0043] Some embodiments relate to a hybrid analog-digital processor comprising: a digital-to-analog converter (DAC) unit comprising a plurality of DACs; an analog processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit; an analog-to-digital (ADC) unit comprising a plurality of ADCs and coupled to outputs of the analog

processor; a digital scaling unit coupled to outputs of the ADC unit; and a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder.

[0044] In some embodiments, the hybrid analog-digital processor further comprises an amplification or attenuation unit coupled between the DAC unit and the ADC unit.

[0045] In some embodiments, the analog processor comprises a photonic processor comprising a plurality of programmable photonic devices.

[0046] In some embodiments, the programmable photonic devices comprise Mach-Zehnder interferometers.

[0047] In some embodiments, the programmable photonic devices comprise optical resonators.

[0048] In some embodiments, the analog processor is programmed with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0049] In some embodiments, the plurality of matrices collectively represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

[0050] In some embodiments, the plurality of matrices collectively represent the arbitrary matrix based on a dilation of the arbitrary matrix.

[0051] Some embodiments relate to a hybrid analog-digital processor configured to perform a mathematical operation, comprising: circuitry comprising an analog processor and an analog scaling unit, wherein the circuitry is configured to: generate a plurality of input analog signals based on an input data set; set a gain of the analog scaling unit based on a scaling factor; program the analog processor with a set of parameters representing a matrix; generate a plurality of output analog signals based on the plurality of input analog signals and the set of parameters; generate a plurality of amplified output analog signals by amplifying, using the analog scaling unit, the plurality of input analog signals and/or the plurality of output analog signals; and generate an output data set based on the plurality of amplified output analog signals.

[0052] In some embodiments, the analog scaling unit comprises an analog amplifier or attenuator.

[0053] In some embodiments, the hybrid analog-digital processor is further configured to perform a multi-pass computation based on the mathematical operation, wherein the circuitry is further configured to: set the gain of the analog scaling unit to a first value during a first pass of the multi-pass computation; and set the gain of the analog scaling unit to a second value, different from the first value, during a second pass of the multi-pass computation.

[0054] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the set of parameters.

[0055] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a convolution based on the plurality of input analog signals and the set of parameters.

[0056] In some embodiments, the circuitry comprises a plurality of analog-to-digital converters (ADCs), and the plurality of ADCs are configured to generate the output data set based on the plurality of output analog signals.

[0057] In some embodiments, the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

[0058] In some embodiments, programming the analog processor comprises: programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0059] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

[0060] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

[0061] In some embodiments, the circuitry is further configured to determine the scaling factor based on the set of parameters and the input data set.

[0062] In some embodiments, determining the scaling factor comprises determining the scaling factor based on statistics bounds on the set of parameters and statistical bounds on the input data set.

[0063] Some embodiments relate to a method for performing a mathematical operation, the method comprising: generating a plurality of input analog signals based on an input data set; setting a gain of an analog scaling unit based on a scaling factor; programming an analog processor with a set of parameters representing a matrix; generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters; generating a plurality of amplified output analog signals by amplifying, using the analog scaling unit, the

plurality of input analog signals and/or the plurality of output analog signals; and generating an output data set based on the plurality of amplified output analog signals.

[0064] In some embodiments, the analog scaling unit comprises an analog amplifier or attenuator.

[0065] In some embodiments, the hybrid analog-digital processor is further configured to perform a multi-pass computation based on the mathematical operation, wherein the circuitry is further configured to: set the gain of the analog scaling unit to a first value during a first pass of the multi-pass computation; and set the gain of the analog scaling unit to a second value, different from the first value, during a second pass of the multi-pass computation.

[0066] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the set of parameters.

[0067] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a convolution based on the plurality of input analog signals and the set of parameters.

[0068] In some embodiments, the circuitry comprises a plurality of analog-to-digital converters (ADCs), and the plurality of ADCs are configured to generate the output data set based on the plurality of output analog signals.

[0069] In some embodiments, the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

[0070] In some embodiments, programming the analog processor comprises: programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0071] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

[0072] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

[0073] In some embodiments, the method further comprises determining the scaling factor based on the set of parameters and the input data set.

[0074] In some embodiments, determining the scaling factor comprises determining the scaling factor based on statistics bounds on the set of parameters and statistical bounds on the input data set.

[0075] Some embodiments relate to a hybrid analog-digital processor comprising: circuitry comprising a photonic processor, wherein the circuitry is configured to perform a mathematical operation using a plurality of passes, wherein for each of the plurality of passes, the circuitry is configured to: determine one or more scaling factors for the pass based on a set of parameters representing a portion of a matrix; scale at least some parameters of the set of parameters based on the one or more scaling factors to produce a scaled set of parameters; program the photonic processor based on the scaled set of parameters; generate a plurality of input optical signals based on an input data set; generate a plurality of output optical signals based on the plurality of input optical signals and the scaled set of parameters; generate a partial output data set based on the plurality of output optical signals; and scale the partial output data set based on the one or more scaling factors to produce a scaled partial output data set, wherein the circuitry is further configured to generate an accumulated output data set by accumulating the scaled partial output data sets generated by at least two of the plurality of passes, wherein the accumulated output data set represents a result of the mathematical operation.

[0076] In some embodiments, the photonic processor comprises a plurality of programmable photonic devices, and wherein programming the analog processor based on the scaled set of parameters comprises setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters.

[0077] In some embodiments, the programmable photonic devices comprise optical resonators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises: setting respective optical characteristics for the plurality of optical resonators based on the scaled set of parameters.

[0078] In some embodiments, the optical resonators comprise optical ring or disk resonators, and wherein setting the respective optical characteristics for the plurality of optical resonators comprises: setting respective resonant frequencies for the plurality of optical ring or disk resonators based on the scaled set of parameters.

[0079] In some embodiments, the programmable photonic devices comprise Franz-Keldysh modulators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises: setting respective absorptions for the plurality of Franz-Keldysh modulators based on the scaled set of parameters.

[0080] In some embodiments, the photonic processor comprises: a first array of interconnected variable beam splitters (VBSs) comprising a first plurality of optical inputs and a first plurality of optical outputs; a second array of interconnected VBSs comprising a second plurality of optical inputs and a second plurality of optical outputs; and a plurality of controllable optical elements, each of the plurality of these controllable optical elements coupling a single one of the first plurality of optical outputs of the first array to a respective single one of the second plurality of optical inputs of the second array.

[0081] In some embodiments, programming the photonic processor based on the scaled set of parameters comprises programing the plurality of controllable optical elements based on the scaled set of parameters.

[0082] In some embodiments, each of the VBSs of the first and second array comprises a Mach-Zehnder interferometer comprising: a first beam splitter;

[0083] a second beam splitter; and at least one phase modulator configured to modulate a phase of light of an optical mode that couples the first beam splitter and the second beam splitter.

[0084] In some embodiments, each of the controllable optical elements comprises a variable attenuator and at least one phase modulator.

[0085] In some embodiments, generating a plurality of output optical signals based on the plurality of input optical signals and the scaled set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input optical signals and the scaled set of parameters.

[0086] In some embodiments, programming the photonic processor comprises: programming, based on the scaled set of parameters, the photonic processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0087] Some embodiments relate to a hybrid analog-digital processor comprising: a digital-to-analog converter (DAC) unit comprising a plurality of DACs; a photonic processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit; an analog-to-digital (ADC) unit comprising a plurality of ADCs and coupled to outputs of the photonic processor; a digital scaling unit coupled to outputs of the ADC unit; and a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder.

[0088] In some embodiments, the hybrid analog-digital processor further comprises an optical amplification or attenuation unit coupled between the DAC unit and the ADC unit.

[0089] In some embodiments, the photonic processor comprises a plurality of optical resonators.

[0090] In some embodiments, the photonic processor comprises a first array of interconnected variable beam splitters (VBSs) comprising a first plurality of optical inputs and a first plurality of optical outputs; a second array of interconnected VBSs comprising a second plurality of optical inputs and a second plurality of optical outputs; and a plurality of controllable optical elements, each of the plurality of these controllable optical elements coupling a single one of the first plurality of optical outputs of the first array to a respective single one of the second plurality of optical inputs of the second array.

[0091] In some embodiments, the photonic processor is programmed with a plurality of matrices that, collectively, represent an arbitrary matrix.

[0092] Some embodiments relate to a hybrid analog-digital processor configured to perform a mathematical operation, comprising circuitry comprising a photonic processor and at least one amplifier, wherein the circuitry is configured to: generate a plurality of input analog signals based on an input data set; set respective a gain of the at least one amplifier based on a scaling factor; program the photonic processor with a set of parameters representing a matrix; generate a plurality of output optical signals based on the plurality of input optical signals and the set of parameters; generate a plurality of output analog signals based on the plurality of output optical signals; generate a plurality of amplified output signals by amplifying, using the at least one amplifier, at least one among the plurality of input optical signals, the plurality of output optical signals and the plurality of output analog signals; and generate an output data set based on the plurality of amplified output signals.

[0093] In some embodiments, the at least one amplifier comprises an optical amplifier and an electronic amplifier, wherein amplifying, using the at least one amplifier, at least one among the plurality of input optical signals, the plurality of output optical signals and the plurality of output analog signals comprises: amplifying the plurality of input optical signals with the optical amplifier; and amplifying the plurality of output analog signals with the electronic amplifier.

[0094] In some embodiments, the at least one amplifier comprises a laser, and wherein setting the gain of the amplifier comprises setting a gain of the laser.

[0095] In some embodiments, the hybrid analog-digital processor is further configured to perform a multi-pass computation based on the mathematical operation, wherein the circuitry is further configured to: set the gain of the at least one amplifier to a first value during a first pass

of the multi-pass computation; and set the gain of the at least one amplifier to a second value, different from the first value, during a second pass of the multi-pass computation.

[0096] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the set of parameters.

[0097] In some embodiments, generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a convolution based on the plurality of input analog signals and the set of parameters.

[0098] In some embodiments, the photonic processor comprises a plurality of optical resonators.

[0099] In some embodiments, the photonic processor comprises: a first array of interconnected variable beam splitters (VBSs) comprising a first plurality of optical inputs and a first plurality of optical outputs; a second array of interconnected VBSs comprising a second plurality of optical inputs and a second plurality of optical outputs; and a plurality of controllable optical elements, each of the plurality of these controllable optical elements coupling a single one of the first plurality of optical outputs of the first array to a respective single one of the second plurality of optical inputs of the second array.

[00100] In some embodiments, programming the analog processor comprises: programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

[00101] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

[00102] In some embodiments, programming the analog processor with a plurality of matrices comprises: programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

[00103] In some embodiments, the circuitry is further configured to determine the scaling factor based on the set of parameters and the input data set.

[00104] In some embodiments, determining the scaling factor comprises determining the scaling factor based on statistics bounds on the set of parameters and statistical bounds on the input data set.

[00105] Some embodiments relate to a method of manufacturing a hybrid analog-digital processor comprising: forming a digital-to-analog converter (DAC) unit comprising a plurality of DACs; forming a photonic processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit; forming an analog-to-digital (ADC) unit comprising a plurality of ADCs and coupled to outputs of the photonic processor; forming a digital scaling unit coupled to outputs of the ADC unit; and forming a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder.

[00106] In some embodiments, forming the photonic processor and forming the digital accumulator comprises forming the photonic processor and the digital accumulator on a common chip.

[00107] In some embodiments, forming the photonic processor and forming the digital accumulator comprises: forming the photonic processor on a first chip; forming the digital accumulator on a second chip; and bonding the first chip to the second chip.

[00108] In some embodiments, forming the ADC unit comprises forming a plurality of n-bit ADCs, where n is less than or equal to 12.

BRIEF DESCRIPTION OF THE DRAWINGS

[00109] Various aspects and embodiments of the application will be described with reference to the following figures. It should be appreciated that the figures are not necessarily drawn to scale. Items appearing in multiple figures are indicated by the same reference number in all the figures in which they appear.

[00110] FIG. 1-1A is a representation of a matrix multiplication operation computed based on tiles, in accordance with some embodiments.

[00111] FIG. 1-1B depicts how a sub-portion of the matrix of FIG. 1-1A can be computed, in accordance with some embodiments.

[00112] FIG. 1-1C is a heat map illustrating a distribution of numerical values of an input matrix, in accordance with some embodiments.

[00113] FIG. 1-2A is a block diagram illustrating a hybrid analog-digital matrix processor, in accordance with some embodiments.

[00114] FIG. 1-2B is a diagram illustrating a matrix multiplication, in accordance with some embodiments.

[00115] FIG. 1-2C is a diagram illustrating a matrix multiplication performed using tiling, in accordance with some embodiments.

[00116] FIG. 1-3 is a flowchart illustrating a method for processing information using a hybrid analog-digital matrix processor, in accordance with some embodiments.

[00117] FIG. 1-4A is a diagram illustrating a matrix multiplication performed using tiling and scaling based on tile factors, in accordance with some embodiments.

[00118] FIG. 1-4B is a diagram illustrating a matrix multiplication performed using tiling and scaling based on tile row factors, in accordance with some embodiments.

[00119] FIG. 1-4C is a diagram illustrating another matrix multiplication performed using tiling and scaling based on tile factors, in accordance with some embodiments.

[00120] FIG. 1-4D is a diagram illustrating another matrix multiplication performed using tiling and scaling based on tile row factors, in accordance with some embodiments.

[00121] FIG. 1-5A is a diagram illustrating a matrix factorized based on singular value decomposition, in accordance with some embodiments.

[00122] FIG. 1-5B is a block diagram illustrating an example of a digital accumulator receiving n vector elements, in accordance with some embodiments.

[00123] FIG. 1-6A is a plot illustrating an output signal distribution in relation to the dynamic range of an analog processor, in accordance with some embodiments.

[00124] FIG. 1-6B is a diagram illustrating the effect of varying gain on the digital representation of data that is digitized from an analog signal, in accordance with some embodiments.

[00125] FIG. 1-7A is a block diagram illustrating another hybrid analog-digital matrix processor, in accordance with some embodiments.

[00126] FIG. 1-7B is a block diagram illustrating a plurality of analog scaling units, in accordance with some embodiments.

[00127] FIG. 1-8 is a flowchart illustrating another method for processing information using a hybrid analog-digital matrix processor, in accordance with some embodiments.

[00128] FIG. 1-9A is a plot illustrating the accuracy of a neural network computation as a function of a gain, in accordance with some embodiments.

[00129] FIG. 1-9B is a plot illustrating a mean square error in component portions of a neural network computation as a function of gain, in accordance with some embodiments.

[00130] FIG. 1-10 is a block diagram illustrating another hybrid analog-digital matrix processor, in accordance with some embodiments.

[00131] FIG. 1-11 is a flowchart illustrating another method for processing information using a hybrid analog-digital matrix processor, in accordance with some embodiments.

[00132] FIG. 1-12 is a block diagram illustrating a photonic hybrid analog-digital matrix processor, in accordance with some embodiments.

[00133] FIG. 2-1 is a schematic diagram of a photonic processing system, in accordance with some embodiments.

[00134] FIG. 2-2 is a schematic diagram of an optical encoder, in accordance with some embodiments.

[00135] FIG. 2-3 is a schematic diagram of a photonic processor, in accordance with some embodiments.

[00136] FIG. 2-4 is a schematic diagram of an interconnected variable beam splitter array, in accordance with some embodiments.

[00137] FIG. 2-5 is a schematic diagram of a variable beam splitter, in accordance with some embodiments.

[00138] FIG. 2-6 is a schematic diagram of an attenuation and phase shifting implementation that may be used to represent a diagonal matrix acting on n optical modes, in accordance with some embodiments.

[00139] FIG. 2-7 is a schematic diagram of an optical attenuator, in accordance with some embodiments.

[00140] FIG. 2-8 is a schematic diagram of an optical power tree, in accordance with some embodiments.

[00141] FIG. 2-9 is a schematic diagram of an optical receiver, in accordance with some non-limiting embodiments, in accordance with some embodiments.

[00142] FIG. 2-10 is a schematic diagram of a homodyne detector, in accordance with some non-limiting embodiments, in accordance with some embodiments.

[00143] FIG. 2-11 is a schematic diagram of a folded photonic processing system, in accordance with some embodiments.

DETAILED DESCRIPTION

I. Overview

[00144] With the recent surge in deep learning applications, there is an increasing demand for digital accelerators that specialize in high performance computation of tensor algebra operations, such as a convolutions or general matrix multiply (GEMM). Conventionally, digital processor architectures including Graphical Processing Units (GPUs), Field Programmable Gate Arrays (FPGAs), and Application-Specific Integrated Circuits (ASICs) have been used as accelerators in such applications.

[00145] The inventors have appreciated that conventional digital accelerators have a major drawback—they are power-hungry. Consider for example digital accelerators programmed to perform large GEMM operations. Such digital accelerators rely on large numbers of transistors clocked at very high frequencies to compute the required multiplications. For decades, transistors have scaled exponentially smaller in size, higher in performance, and lower in power consumption—this is what experts generally refer to as Moore's law. In recent years, however, this trend has begun to slow down. Primarily due to leakage current between extremely small-sized transistors, it has become increasingly more difficult to continue to scale performance without sacrificing power consumption.

[00146] Recognizing these limitations, the inventors have developed new accelerator architectures that improve scalability both in terms of performance and energy efficiency. The accelerators developed by the inventors combine the flexibility and programmability of digital controllers with the energy efficiency and inherent speed of analog circuits. These accelerators are referred to herein as hybrid analog-digital processors.

[00147] Hybrid analog-digital processors of the types described herein are particularly suitable for computing mathematical operations. One such mathematical operation is general matrix multiply (GEMM), and another such operation is convolution, though embodiments of the present application are not limited to any specific operation.

[00148] A GEMM operation can be expressed as follows:

$$D = \alpha(A \times B) + \beta C$$

where α and β are scalars (individual numbers), and A, B, C and D are matrices. The inner dimensions of A and B agree such that $(A \times B)$ is a matrix multiplication. Several mathematical operations can be decomposed in terms of GEMM operations, thus rendering the accelerators developed by the inventors applicable to a variety of contexts.

[00149] Some particular applications, such as deep artificial neural networks, require the computation of vast amounts of data, typically represented as matrices or tensors. One approach to perform operations over vast amounts of data is to use large hybrid analog-digital processors. For example, to perform a linear operation involving matrices or tensors having tens of thousands of rows and columns, hybrid analog-digital processors having tens of thousands of analog elements may be used.

[00150] The inventors have appreciated, however, that this approach can be costly because designing a processor with so many analog components may require more chip real estate than is available or more power than is practical for such a chip. Additionally, the inventors have appreciated that some applications involve heterogeneous workloads where some matrix operations have very large dimensions and other operations have smaller dimensions such that the overall utilization of a processor with enough elements for the largest computation would be poor. Accordingly, some embodiments relate to hybrid analog-digital processors that can process vast amounts of data with relatively small numbers of analog components. For example, a matrix operation involving matrices having thousands of rows and columns may be performed in some embodiments using a hybrid analog-digital processor having only a few hundreds of analog elements (it should be appreciated that these numbers are provided only by way of example, as different number of analog components may be used).

[00151] In some embodiments, matrix operations may be performed by making multiple passes through a hybrid analog-digital processor. At each pass through the processor, an operation is performed over a sub-portion of a matrix (referred to herein as a “tile”). FIGs. 1-1A and 1-1B illustrate how matrix multiplication can be performed using tiles. Referring first to FIG. 1-1A, matrix C is obtained by multiplying matrix A by matrix B. Matrix A has I rows and K columns, and matrix B has K rows and J columns. As a result, matrix C has I rows and J columns. A1 and A2 represent tiles of matrix A. For the sake of clarity, matrix A has been segmented in four tiles in this example, though in some embodiments matrix A may be segmented in tens, hundreds, or thousands of tiles, or more. Similarly, matrix B has been segmented into four non-square tiles in this example, though in some embodiments matrix B may also be segmented in tens, hundreds, or thousands of tiles, or more. Tiles A1 and B1 may each include N rows and M columns (though tiles of the types described herein need not be square), where $N \leq I$, $N \leq K$, $N \leq J$, $M \leq I$, $M \leq K$, and $M \leq J$. In this example, matrix B is divided into tile rows B1 and B2. Each tile row includes N rows and J columns.

[00152] Each tile row of matrix C is obtained by multiplying respective tiles of matrix A with respective tile rows of matrix B, and by accumulating the results. Each multiplication is performed as a pass through a hybrid analog-digital processor. As illustrated in FIG. 1-1B, for example, tile A1 is multiplied by tile row B1 at a first pass through a hybrid analog-digital processor and tile A2 is multiplied by tile row B2 at a second pass. The results are then accumulated, such that:

$$C1=A1\times B1+A2\times B2$$

[00153] This example involves accumulation of two partial results. In other embodiments, however, there may be more partial results, depending on the dimensions of the tile relative to the dimensions of the matrices to be multiplied. For GEMM applications such as those designed to make predictions, where the same "weight" matrix (A) is re-used with many different batches of data (B), this tile ordering may have benefits for spatial or temporal locality and reduced movement of weight-data.

[00154] The inventors have further developed a particular class of analog processors that uses light signals. The inventors have appreciated that, because light signals travel at the speed of light in the medium in which the light is traveling, latency is far less of a limitation than in electrical systems. Additionally, virtually no power is dissipated by increasing the distance traveled by the light signals, opening up new topologies and processor layouts that would not be feasible using electrical signals. Thus, light-based processors, referred to herein as "photonic processors," may have better speed and efficiency performance than conventional electrical processors. It should be appreciated that photonic processors are only one possible implementation of the accelerators described herein, as other types of hybrid analog-digital processors may also be used. Examples of such hybrid analog-digital processors are provided in detail further below.

[00155] The inventors have further appreciated that, in some circumstances, the matrix representing the input data (B) and/or the "weight" matrix (A) may exhibit large variations across the magnitude of its values. Consider for example the matrix of FIG. 1-1C, where darker regions represent clusters of larger magnitude values and lighter regions represent clusters of smaller magnitude values. Having entries with large variations across the magnitude of values poses a significant challenge, which arises due to the finite dynamic range of digitizing an

analog signal, as is done for number representations in a hybrid analog-digital processor. Small magnitude values get rounded to the smallest non-zero representable, meaning that information in the lower bits may be limited by the input quantization noise. Simply amplifying the signal to read the lower bits may produce noisy results.

[00156]

[00157] The inventors have developed techniques for overcoming the aforementioned challenges. The techniques developed by the inventors involve normalization of the entries of the weight matrix, or normalization of the entries of the input data set, to match the dynamic range of the system. The normalization may involve scaling different portions of a matrix by different amounts. The normalization may be made on a tile basis, on a row basis, on a tile row basis, or in any other suitable manner.

[00158] When different portions of a matrix are scaled by different amounts, however, an issue arises— the values calculated from different matrix portions may no longer be directly accumulated. To address this issue, an inverse scaling is performed in some embodiments. Once back-scaled, the results of different matrix portions are restored to the original data range, thus allowing for direct accumulation.

[00159] The inventors have further appreciated that representing digital values with large numbers of bits, while providing high degrees of representational accuracy, can lead to a significant increase in power consumption. Consider for example analog-to-digital converters (ADCs). The power consumption of an ADC depends, among other factors, upon the number of bits with which values are represented. The energy required for some ADCs grows exponentially with the number of output bits. In essence, there is a trade-off between representational accuracy and power consumption.

[00160] Conventional digital processors utilize binary number representations such as floating-point, integer, fixed-point, flex-point etc. To achieve higher performance for certain applications, non-standard binary number representations including nonlinear significand maps (such as logarithmic number representations), binary stochastic numbers, and entropy codings (such as posits, unums, and tapered floating point) have been developed. Relative to some true matrix multiplication, rounding and overflow/underflow with such representations may cause the result of a digital GEMM to disagree with the true arithmetic result. In floating-point (FP) representations such as the standard IEEE 754 “binary64,” also known as “double” or “FP64,” this loss of accuracy is for the most part negligible. The partitioning of mantissa, exponent and exponent bias may also be modified relative to the IEEE 754 standard. For instance, the 16-bit

Brain Floating Point or "bfloat16" representation devotes 8-bits to the exponent rather than the 5-bits that the IEEE standard "half" precision (FP16). To achieve higher performance and improve energy efficiency, many accelerators perform computations with lower precision representations, such as 16-bit floating-point (FP16), and may use mixed-precision representations. For instance, NVIDIA's Volta generation Tensor Cores compute a GEMM with 4x4 matrices A and B in "half" precision (FP16) and C in either half or "single" precision (FP32).

[00161] The inventors have appreciated that low-precision (e.g., 16-bit or less, 12-bit or less, 10-bit or less) fixed-point binary number representations are far less demanding than floating-point binary number representations in terms of chip area and power requirements. Further, not all applications require the results of mathematical operations to be represented with full precision. For example, deep learning applications can often tolerate significantly lower precision than other applications of numerical linear algebra. However, using low-precision representations poses a significant challenge in that it reduces the accuracy with which numbers can be represented. To make things even worse, reducing the precision of an ADC can also reduce its dynamic range. As a result, there is a trade-off between representing the full dynamic range of a signal, and representing smaller output values precisely. In converting signals from the analog to the digital domain, a low-precision ADC may attempt to set the uppermost bits of the output. However, this may result in the ADC setting the uppermost bits to zero—that is, the ADC is attempting to read a signal range having no meaningful information. This problem is further exacerbated by the properties of computations in a neural network model—conditions vary over time, for example from one layer of the network to another layer of the network.

[00162] The inventors have developed accelerator architectures designed to operate with low-precision fixed-point binary number representations, thereby limiting chip area and power consumption, that provide sufficient accuracy for a variety of practical applications, including for deep learning applications. These accelerator architectures can be reconfigured dynamically, thus accounting for the varying dynamic conditions of a neural network model. The accelerator architectures developed by the inventors are designed to shift the signal range or "window" that an ADC is able to read. Shifting the readout window allows the ADC to focus the conversion on a portion of a signal that is likely to have the most amount of meaningful information, or at least to have some meaningful information.

[00163] In some embodiments, shifting the readout window in this manner may involve the use of analog scaling units (e.g., amplifiers or attenuators). Adjusting the gain of an analog scaling

unit causes the readout window to shift, which in turn allows for a more accurate reading. Considering for example an 8-bit ADC, setting the gain to 2 may cause the readout window to shift 1-bit towards the lowermost bit, while setting the gain to 1/2 may cause the readout window to shift 1-bit towards the uppermost bit. In some embodiments, the position of a readout window is adjusted dynamically, thus allowing an accelerator to reconfigure itself based on the scale of the problem at hand. For example, it may be desirable to vary the position of the readout window dynamically between different layers of a neural network model to account for changes in output data distribution that may occur from layer to layer, as described in detail in connection with FIG. 1-9B.

II. Matrix tiling

[00164] In some embodiments, matrix operations may be performed by making multiple passes through a hybrid analog-digital processor. At each pass, the processor performs an operation over a tile of a matrix. FIG. 1-2A is a block diagram illustrating an example of an hybrid analog-digital matrix processor implementing tiling techniques in accordance with some embodiments. Hybrid analog-digital matrix processor 1-100 (also referred to herein as accelerator 1-100) comprises a digital-to-analog converter (DAC) unit 1-102, an analog processor 1-104, an analog-to-digital converter (ADC) unit 1-106, a digital scaling unit 1-108, digital accumulator 1-110 and a controller 1-120. Controller 1-120 may control the operations of accelerator 1-100. For example, controller 1-120 may comprise software instructions that, when executed, cause the accelerator to perform a mathematical operation. The components of FIG. 1-2A, and optionally other components, are collectively referred to herein as "circuitry." The components of FIG. 1-2A may be formed on a common chip, or on different chips bonded together.

[00165] As described in detail further below, accelerator 1-100 is configured to perform mathematical operations by performing several passes through analog processor 1-104. At each pass through analog processor 1-104, accelerator 1-100 computes a partial result using analog processor 1-104. The partial results are added together using digital accumulator 1-110.

[00166] Analog processor 1-104 can be implemented in any of numerous ways using analog components. Analog processor 1-104 is configured, in some embodiments, to perform linear operations such as matrix-vector multiplication or matrix-matrix multiplication. Accordingly, processor 1-104 may include multiple analog components designed to perform analog multiplications or other mathematical operations.

[00167] DAC unit 1-102 comprises multiple DACs. Any suitable type of DAC may be used, including for example resistive ladder DACs, switched-capacitor DACs, switched resistor DACs, binary-weighted DACs, thermometer-coded DACs, successive approximation DACs, oversampling DACs, interpolating DACs, hybrid DACs, etc. Similarly, ADC unit 1-106 comprises multiple ADCs. Any suitable type of ADC may be used, including for example parallel comparator ADCs, flash ADCs, successive-approximation ADCs, Wilkinson ADCs, integrating ADCs, sigma-delta ADCs, , pipelined ADCs, integrating ADCs, cyclic ADCs, time-interleaved ADCs, etc.

[00168] FIG. 1-2B is a representation of a matrix-matrix multiplication that may be performed using an accelerator 1-100. In some embodiments, matrix A represents a weight matrix and matrix B represents an input matrix that is encoded in terms of input data to the analog processor. In other embodiments, matrix A represents the input matrix and matrix B represents a weight matrix. Either way, matrix C represents output data. The entries of matrix A (also referred to herein as "parameters") may be stored in a memory of the accelerator, and the entries of matrix B may be provided as input to the accelerator. C is computed by applying weight matrix A to input matrix B. More specifically, matrix C is computed by multiplying matrix A by matrix B. For the sake of clarity, matrix A is illustrated as a 4x4 matrix, matrix B as a 4x3 matrix and matrix C as a 4x3 matrix. Of course, the matrices may have any suitable number of rows and columns. In some embodiments, the matrices may have rows and columns in the hundreds, thousands, tens of thousands, or more. In some embodiments, A and B may be N-dimensional tensors such that the matrix operations are to be performed on multiple instances of the same-sized matrices. In some embodiments, the number of columns of A equals the number of rows of B, thus allowing for the matrices to be multiplied to one another. For example, matrix A may have I rows and K columns, matrix B may have K rows and J columns. As a result, matrix C has I rows and J columns. This relation can be expressed as follows:

$$dim(A) = (I, K) \ \& \ dim(B) = (K, J) \ \& \ dim(C) = (I, J)$$

where the operator dim(), when applied to a matrix, returns the number of rows and columns of the matrix. The entries of matrix A are a_{ik}, where i=1, 2,…,I, and k=1, 2,…,K. The entries of matrix B are b_{kj}, where j=1, 2,…,J. The entries of matrix C are c_{ij}.

[00169] Multiplying the entirety of matrix A with the entirety of matric B with a single pass requires large analog processors, especially when I, K and J are large. Designing large processors may be impractical because it may require more chip real estate than is available and may result in significant power dissipation. In some embodiments, it may be more practical to perform the operation of FIG. 1-2B using a relatively small analog processor. In these embodiments, instead of directly multiplying the matrices in their entireties, the multiplication may be broken down by tile. In the depiction of FIG. 1-2C, for example, matrix A has been segmented in four tiles—A1, A2, A3 and A4. In this specific example, N=2—each tile of A has two rows and two columns (though other values of N are also possible, so long as $N \leq I$, $N \leq J$, and $N \leq K$). Matrix B is segmented in tile rows B1 and B2, and matrix C is segmented in tile rows C1 and C2. The tile rows of matrix C are given by the following expressions:

$$C1=A1\times B1+A2\times B2$$

$$C2=A3\times B1+A4\times B2$$

[00170] The multiplication $A1\times B1$ may be performed with a first pass through the analog processor, the multiplication $A2\times B2$ may be performed with a second pass through the analog processor, and the results may subsequently be accumulated to one another to obtain C1. Similarly, the multiplication $A3\times B1$ may be performed with a third pass through the analog processor, the multiplication $A4\times B2$ may be performed with a fourth pass through the analog processor, and the results may subsequently be accumulated to one another to obtain C2.

[00171] FIG. 1-3 is a flowchart illustrating an example of a method for performing a mathematical operation, in accordance with some embodiments. Method 1-300 may be performed using accelerator 1-100 (FIG. 1-2A), though any other suitable accelerator may be used. It should be appreciated that the steps of method 1-300 may be performed in the order depicted in FIG. 1-3, or in any other suitable order. In particular, the order of operation may be chosen to exploit spatial and/or temporal locality. Method 1-300 loops multiple times, where each loop corresponds to a pass through an analog processor. FIG. 1-2A illustrates the pth iteration, with $p=1,2\ldots P$.

[00172] At step 1-301, controller 1-120 determines one or more scaling factors for the current pass. As described above in connection with FIG. 1-1C, in some embodiments, a matrix may exhibit significant variations in magnitude across its entries, such that it may be desirable to

normalize the entries of matrix A, or matrix B, to match the dynamic range of the analog processor. The determination of step 1-301 may involve determining the value of the scaling factor(s) that scales the input data set to match the dynamic range of the ADC unit. This determination may be performed in any of numerous ways. In some embodiments, the determination may be made on the basis of the parameters of matrix A. In some embodiments, the determination may be made on the basis of the parameters of the current tile of matrix A, or other tiles. In this case, it may be advantageous to reorder the matrix A such that entries with similar dynamic range fall in the same tile. This may also be the case for matrix B. In some embodiments, the dimension of the tiles of matrix A may be chosen depending on the degree of granularity with which one desires to scale the matrix. For example, if it is expected that the matrix will exhibit sharp variations across its values, the dimension of the tiles may be chosen to be small, thus providing a fine resolution. The drawback is that more passes will be needed. By contrast, if it is expected that matrix B will exhibit relatively slow variations across its values, the dimension of the tiles may be chosen to be larger, thus limiting the number of passes.

[00173] The scaling factors may be determined in any suitable ways. In one example, a scaling factor is set to the average value of the entries of the current tile. In another example, a scaling factor is set to the highest value of the entries of the current tile. In yet another example, a scaling factor is set to the lowest value of the entries of the current tile. In yet another example, the scaling factors may be set to the maximum absolute values of the rows of the current tile.

[00174] In some embodiments, matrix A can be factorized or decomposed in multiplying matrices. The entries of these matrices may be such that combining (e.g., multiplying) the matrices together produces the original matrix A. Example of matrix decompositions that may be applied to matrix A include singular value decomposition (SVD), lower-upper decomposition, dilation (e.g., unitary dilation), rank decomposition, Cholesky decomposition, QR decomposition, interpolative decomposition, spectral decomposition, polar decomposition, etc. In some embodiments, a scaling factor is determined based on a parameter of one of the matrices resulting from the decomposition of matrix A.

[00175] In one example, matrix A is decomposed based on SVD, and a scaling factor is determined based on one or more of the singular values of matrix A. For example, a scaling factor may be set to the highest singular value of the current tile, such as to the highest among the absolute values of the singular values of the current tile. In some embodiments:

$$t_p = max\{|\sigma_i|\} \ such\ that\ i = 1,2,\ldots,N\ and\ \Sigma_p = diag[\sigma_i]$$

where t_p represents the scaling factor for tile p, Σ_p represents the diagonal matrix associated with tile p, and σ_i represent the diagonal values of matrix Σ_p (that is, the singular values of the pth tile of A). The quantity $|\sigma_i|$ indicates the absolute value of σ_i. In yet another example, a scaling factor is set to the average of the singular values of the current tile. In yet another example, a scaling factor is set to the lowest of the singular values of the current tile.

[00176] Matrix A need not be segmented by tile for purposes of determining a scaling factor. In some embodiments, a scaling factor may be determined based on a row of a tile (tile row) of matrix A, based on multiple tiles of matrix A, based on a row of matrix A, based on one or more columns of matrix A, or based on any other suitable portion of matrix A. In some embodiments, a scaling factor may be set to the same value for more than one tile of matrix A.

[00177] At step 1-302, controller 1-120 scales at least some of the parameters of matrix A based on the scaling factor(s) determined at step 1-301. In some embodiments, controller 1-120 scales the parameters (or at least some of the parameters) of the current tile of matrix A based on the scaling factors. For example, controller 1-120 may scales a first subset (e.g., one or more rows of matrix A and/or one or more columns of matrix) of the parameters based on a first scaling factor and may scales a second subset of the parameters based on a second scaling factor. In the example of FIG. 1-4A, the parameters of a tile are scaled based on the scaling factor determined for that particular tile. Scaling factor t1 indicates the scaling factor determined for tile A1, scaling factor t2 indicates the scaling factor determined for tile A2, scaling factor t3 indicates the scaling factor determined for tile A3 and scaling factor t4 indicates the scaling factor determined for tile A4. In this example, the entries of tile A1 are divided by t1, the entries of tile A2 are divided by t2, the entries of tile A3 are divided by t3 and the entries of tile A4 are divided by t4 (though operations other than a division are also possible, including a multiplication).

[00178] In the example of FIG. 1-4B, the parameters of a tile row of matrix A are scaled based on the scaling factor(s) determined for that particular tile row. Scaling factor tr1 indicates the scaling factor determined for tiles A1 and A2, and scaling factor tr2 indicates the scaling factor determined for tiles A3 and A4. In this example, the entries of the tile row consisting of tiles A1 and A2 are divided by tr1 and the entries of the tile row consisting of tiles A3 and A4 are divided by tr2.

[00179] In some embodiments, instead of scaling the parameters of matrix A, controller 1-120 may scale the entries of matrix B, the input data set. The entries may be scaled in the digital domain or in the analog domain (after having passed through DAC unit 1-102). When done in the analog domain, the scaling may involve attenuating or amplifying the input analog signal based on the scaling factor.

[00180] In the example of FIG. 1-4C, the parameters of a tile row of matrix B are scaled based on the scaling factor(s) determined for a corresponding tile of matrix A. Here, scaling factor t1 indicates the scaling factor determined for tile A1 and scaling factor t2 indicates the scaling factor determined for tile A2. In this example, the entries of tile row B1 are divided by t1 and the entries of tile row B2 are divided by t2.

[00181] At step 1-304, controller 1-120 programs analog processor 1-104 based on the parameters of the current tile of matrix A. In at least some of the embodiments in which the parameters of matrix A have been scaled at step 1-302, controller 1-120 programs analog processor 1-104 based on the scaled parameters of the current tile of matrix A. Referring back to FIG. 1-2A, for example, controller 1-120 programs analog processor 1-104 based on a11/t1, a12/t1, a21/t1 and a22/t1. Programming the analog processor may involve setting certain characteristics of the analog processor based on the parameters. In one example, analog processor 1-104 includes multiple electronic amplifiers (e.g., voltage amplifiers, current amplifiers, power amplifiers, transimpedance amplifiers, transconductance amplifiers, operational amplifiers, or transistor amplifiers, among others), and programming the analog processor involves setting the gains of the electronic amplifiers based on the parameters of tile A1. In another example, analog processor 1-104 includes multiple electronic attenuators (e.g., voltage attenuators, current attenuators or power attenuators, among others), and programming the analog processor involves setting the attenuations of the electronic attenuators based on the parameters of tile A1. In yet another example, analog processor 1-104 includes multiple electronic phase shifters, and programming the analog processor involves setting the phase shifts of the electronic phase shifters based on the parameters of tile A1. In another example, the analog processor includes an array of memory devices, such as flash or ReRAM, typically laid out in a crossbar fashion. Programming the analog processor involves setting the conductances (or resistances) of each of the memory cells.

[00182] As described in detail further below, in some embodiments analog processor 1-104 is implemented with a photonic processor. In some such embodiments, analog processor 1-104 includes multiple optical amplifiers (e.g., semiconductor optical amplifiers or other solid state

amplifiers, rare earth doped optical amplifiers such as erbium doped optical amplifiers, Raman amplifiers, or optical parametric amplifiers, among others), and programming the analog processor involves setting the gains of the optical amplifiers based on the parameters of tile A1. In another example, analog processor 1-104 includes multiple optical attenuators (e.g., Mach Zehnder interferometers, leaky waveguides, loopback attenuators, liquid crystal attenuators, saturable absorbers, or Franz-Keldysh absorbers, metal-oxide-semiconductor-capacitor modulators, carrier depletion electro-optic modulators, among others), and programming the analog processor involves setting the attenuations of the optical attenuators based on the parameters of tile A1. In yet another example, analog processor 1-104 includes multiple optical phase shifters (e.g., thermo-optic phase shifters, acousto-optic phase shifters, electro-optic phase shifters, plasma dispersion effect phase shifters, Pockels effect phase shifters, nano-opto-electro-mechanical phase shifters, or non-linear phase modulators, among others), and programming the analog processor involves setting the phase shifts of the optical phase shifters based on the parameters of tile A1. In yet another example, analog processor 1-104 includes multiple optical resonators (e.g., ring resonators, and disk resonators, photonic crystals, ridge resonators, among others), and programming the analog processor involves setting the resonant frequencies of the optical resonators based on the parameters of tile A1.

[00183] At step 1-306, accelerator 1-100 receives an input data set. Referring back to FIG. 1-2C, for example, the accelerator may receive the entries of tile row B1 at one pass and the entries of tile row B2 at another pass. In some embodiments, controller 1-120 receives and stores the entries of matrix B at the same time, but provides one tile row at a time (e.g., per pass) to DAC unit 1-102. The entries may represent real numbers. In some embodiments, the entries may represent positive real numbers or signed real numbers (positive as well as negative real numbers).

[00184] At step 1-308, accelerator 1-100 generates a plurality of input analog signals based on the input data set received at step 1-302. Referring back to FIG. 1-2A, the generation of step 1-304 involves converting the input data set to analog signals using DAC unit 1-102.

[00185] At step 1-310, analog processor 1-104 generates a plurality of output analog signals based on the plurality of input analog signals and the set of parameters with which the analog processor was programmed at step 1-301. This may involve, in some embodiments, performing a matrix multiplication. The result is that the plurality of input analog signals are multiplied by respective parameters of the current tile. For the pth pass, the result is the following expression:

$$\frac{A_p}{t_p} \times B_p$$

where A_p is the pth tile of matrix A, B_p is the pth tile row of matrix B and t_p is the pth scaling factor. In other embodiments, generating the plurality of output analog signals involves a convolution based on the plurality of input analog signals and the set of parameters.

[00186] Analog processor 1-104 may include any suitable analog components designed to perform a matrix multiplication, or a convolution. As one example, analog processor 1-104 may include analog mixers. Each mixer mixes an input analog signal with an analog signal encoded (e.g., modulated) based on the corresponding parameter of matrix A. As another example, analog processor 1-104 may include amplitude modulators. As another example analog processor 1-104 may include current steering circuits. Each modulator modulates an input analog signal by a quantity obtained based on the corresponding parameter of matrix A. As another example, analog processor 1-104 may include amplifiers. Each amplifier amplifies an input analog signal by a gain obtained based on the corresponding parameter of matrix A. As another example, analog processor 1-104 may include attenuators. Each attenuator attenuates an input analog signal by an attenuation obtained based on the corresponding parameter of matrix A. Analog processor 1-104 may be implemented using any suitable technology, including for example using complementary metal-oxide-semiconductor (CMOS) components, radio-frequency (RF) components, and microwave components, though other technologies may be used. In some embodiments, analog processor 1-104 includes a photonic processor, and the multiplication is performed in the optical domain. In some embodiments analog processor 1-04 includes a combination of photonic and analog electronic components and some multiplications may be performed in the optical domain while other multiplications or additions are performed in the electronic domain. Examples of photonic processors are described in detail further below.

[00187] In some embodiments, instead of performing an operation (e.g., matrix multiplication or convolution) based on the input analog signals and the scaled set of parameters, step 1-310 may involve performing an operation based on scaled input analog signals and unscaled parameters (as described for example in connection with FIGs. 1-4C and 1-4D). Either way, the result is $A_p \times \frac{B_p}{t_p}$. In some embodiments, scaling of matrix B may be performed on the basis of column tiles. That is, distinct column tiles of matrix B are scaled by different factors.

[00188] In some embodiments, the analog multiplication may involve a matrix decomposition. The entries of these matrices may be such that combining (e.g., multiplying) the matrices

together produces the original matrix A. In some such embodiments, analog processor 1-104 may be designed to perform a transformation on the input data set based on each of the matrices in which matrix A has been decomposed. Specific examples of matrix decompositions are provided above. One such example is SVD. As illustrated in FIG. 1-5A, the controller 1-120 may decompose matrix A to be represented by three constituent matrices: $A = U\Sigma V^{\dagger}$, where U and V are unitary I x I and K x K matrices, respectively ($U^{\dagger}U = UU^{\dagger} = I$ and $V^{\dagger}V = VV^{\dagger} = I$), and Σ is an I x K diagonal matrix with real or complex-valued entries. In some embodiments, the values of the diagonal singular values may also be further normalized such that the maximum absolute value of the singular values is 1. The values σ_1, σ_2... σ_K represent the singular values. Analog processor 1-104 may be include analog hardware designed to transform the input analog signals according to each of matrices U, Σ and V.

[00189] Referring back to FIG. 1-3, at step 1-312, accelerator 1-100 generates a partial output data set based on the plurality of output analog signals obtained at step 1-310. In some embodiments, the generation of step 1-312 involves passing the output analog signals through ADC unit 1-106.

[00190] At step 1-314, accelerator 1-100 scales the partial output data set based on the scaling factor to produce a scaled partial output data set. The scaling of step 1-314 may involve passing the partial output data set though digital scaling unit 1-108. In some embodiments, digital scaling unit 1-108 may be configured to apply the inverse of the operation performed at step 1-308. For example, in at least some of the embodiments in which step 1-308 involves a division, step 1-314 may involve a multiplication. Similarly, in at least some of the embodiments in which step 1-308 involves a multiplication, step 1-314 may involve a division. This restores the entries of the output matrix to the original range.

[00191] In some embodiments, the scaling of step 1-314 may be based on the same scaling factor used for the scaling of step 1-302 (though it may be applied using the inverse operation). Thus, if the scaling of step 1-302 involves a division by t_p, the scaling of step 1-314 involves a multiplication by t_p. In other embodiments, the scaling of step 1-314 may involve an additional factor. For example, if the scaling of step 1-308 involves a division t_p, the scaling of step 1-314 may involve a multiplication by t_p/r, where factor r may be computed based on all the scaling factors t_p. In one example, factor r is computed as follows:

$$r = f(\{t_p : p = 1,2,\ldots,K//N\})$$

where the function f(•) may represent any suitable function, such as max(•), min(•), or mean(•). Scaling by the factor t_p/r has the effect of unifying the data range of the scaling factors.

[00192] At step 1-316, the scaled partial output data set generated at step 1-314 is accumulated. At each pass, the accumulation may involve summing the scaled partial output data set generated by the current pass with the result of a previous accumulation. Thus, each pass adds the current result to the cumulative result. The accumulation of step 1-320 may be performed using digital accumulator 1-110. An example of accumulator 1-110 is illustrated in FIG. 1-5B, in accordance with some embodiments. In this example, accumulator 1-110 includes a memory unit 1-520 and a digital adder 1-522. Memory unit 1-520 stores the result of the previous accumulation. Digital adder 1-522 adds the scaled partial output data set generated by the current pass to the cumulative result.

[00193] In some embodiments, instead of performing an accumulation step at each pass, the partial results may be temporarily stored in a memory, and a single accumulation may be performed after completion of the passes. Either way, process 1-100 may involve at least one accumulation step in which the scaled partial output data sets generated by at least two of the plurality of passes are accumulated.

[00194] At step 1-318, accelerator 1-100 determines whether an additional pass should be performed. This may involve, for example, determining whether there are additional tiles of matrix A that have not been processed. That is, in some embodiments, method 1-300 iterates for a number of cycles equal to the number of tiles of matrix A.

[00195] If accelerator 1-100 determines that an additional iteration is needed, method 1-300 performs another pass through the accelerator. In the next pass, another tile of matrix A is processed. By contrast, if accelerator 1-100 determines that no additional iterations are needed, method 1-300 ends.

III. <u>Partial products</u>

[00196] In some embodiments, the analog processor performs a matrix-vector product: $\vec{c} = A\vec{b}$, where $\vec{b}$ is the input vector, A is an $n \times n$ matrix, and $\vec{c}$ is the output vector. In index notation, this multiplication is written as $c_i = \sum_{j=1}^{n} A_{ij}\, b_j$ (For all i in [0,N-1]) which is the multiplication between n elements of A_{ij}(iterate over j) and n elements of b_j (iterate over j) and then summation of these multiplication results altogether. In some embodiments the elements A_{ij}

and b_j may be represented with an unsigned fixed-point number representation (e.g., INT8). Within this representation, if $A_{ij} \in \{0,1\}^{m_1}$ is an m_1-bit number and $b_j \in \{0,1\}^{m_2}$ is an m_2-bit number, then a total of $m_1 + m_2 + log_2(n)$ bits are used to fully represent the resulting vector element c_i. In general, the number of bits required to represent the result of a matrix-vector product without loss of accuracy is larger than the number of bits required to represent the inputs of the operation. If ADC unit 1-106 is unable to read out the output vector at full precision, then the output vector elements may be rounded to the precision of the ADC.

[00197] The inventors have recognized and appreciated that constructing an ADC with a high bit-precision at bandwidths that correspond to the rate at which input vectors in the form of analog signals are sent through analog processor 1-104 can be difficult to achieve. Therefore, in some embodiments, the bit precision of DAC 1-102 may limit the bit precision at which the matrix elements A_{ij} and the vector element b_j are represented. Accordingly, the inventors have devised a method of obtaining an output vector at its full precision, which can be arbitrarily high, by computing partial products and sums. For the sake of clarity, it will be assumed that the number of bits needed to represent either A_{ij} or b_j is the same, i.e. $m_1 = m_2 = m$. However, this assumption can be removed and does not limit the scope of embodiments of the present disclosure.

[00198] The method, according to some embodiments, as a first act, includes dividing the bit-string representation of the matrix element A_{ij} and the vector element b_j into d divisions with each division containing $k = m/d$ bits. (If k is not an integer, zeros may be appended until m is divisible by d.) As a result, the matrix element

$A_{ij} = A_{ij}^{[0]}2^{k(d-1)} + A_{ij}^{[1]}2^{k(d-2)} + \cdots + A_{ij}^{[d-1]}2^{0}$, where $A_{ij}^{[f]}$is the k-bit value of the f-th most significant k-bit string of A_{ij}. In terms of bit strings, one can write

$A_{ij} = A_{ij}^{[0]}A_{ij}^{[1]}\cdots A_{ij}^{[d-1]}$. Similarly, one can also obtain

$b_j = b_j^{[0]}2^{k(d-1)} + b_j^{[1]}2^{k(d-2)} + \cdots + b_j^{[d-1]}2^{0}$, where the vector element

$b_j = b_j^{[0]}b_j^{[1]}\cdots b_j^{[d-1]}$ in terms of its bit string. The multiplication $c_i = \Sigma_j A_{ij}b_j$can be broken down in terms of these divisions as: $c_i = \sum_{p=0}^{2(d-1)}((\Sigma_{f,g\in S_p}\Sigma_j A_{ij}^{[f]}b_j^{[g]})2^{2k(d-1)-pk})$, where the set $S_p$is the set of all integer values of f and g, where $f + g = p$.

[00199] The method, as a second act, includes controlling analog processor 1-104 to implement the matrix $A_{ij}^{[f]}$ and propagating the input vector $b_j^{[g]}$, each of which is only k-bit precise, through analog processor 1-104 in the form of encoded analog signals. Using any of the tiling approaches described above, the matrix-vector product operation performs $c_i^{[f,g]} = \Sigma_j A_{ij}^{[f]} b_j^{[g]}$. The method includes, storing the output vector $c_i^{[f,g]}$which is precise up to $2k + log_2(n)$bits.

[00200] The method further includes iterating over the different values of f, g within the set S_p and repeating the second act for each of the different values of *f, g* and storing the intermediate results $c_i^{[f,g]}$.

[00201] As a third act, the method includes computing the final result $\Sigma_{f,g \in S_p} \Sigma_j A_{ij}^{[f]} b_j^{[g]} = \Sigma_{f,g \in S_p} c_i^{[f,g]}$ by summing over the different iterations of f and g with digital electronics, such as a processor.

[00202] The precision of the ADC used to capture a fully precise computation according to some embodiments of this method is only $2k + log_2(n)$ bits, which is fewer than the $2m + log_2(n)$ bits of precision that would otherwise be needed.

[00203] The inventors have further recognized and appreciated that embodiments of the foregoing method can be generalized to operate on tensors. As previously described, the hybrid analog-digital processing system can perform matrix-matrix multiplications using tiles. The method described above can be applied to the matrix tiles and input data set tile rows to obtain the output data set tile row at full precision.

[00204] As described in detail further below, in some of the embodiments in which the analog processor comprises a photonic processor, the matrix may be represented in terms of the phases of the optical variable beam splitters (VBSs). In some such embodiments, the divisions may be performed on the bit strings representing the phases, instead of the matrix elements directly. In some embodiments, when the map between the phases to the matrix elements is a linear map, then the relationship between the input parameters—the phases of the VBSs and the input vector elements in this case—and the output vector may be linear. When this relationship is linear, the method described above is still applicable. However, in general, a nonlinear map from the elementary representation of the matrix to the photonic representation may be considered, according to some embodiments. For example, the bit-string division of the Euclidean space

matrix elements from their most-significant k-bit string to the least-significant k-bit string may be used to produce a series of different matrices that are decomposed to a phase representation and implemented using a photonic processor.

The divisions need not be performed on both the weight matrix entries and the input data set entries simultaneously. In some embodiments, a photonic processor may propagate many input data sets for the same matrices. It may be efficient to only perform the divisions on the input data sets and keep the VBS controls at a set precision (e.g., full precision) because the digital-to-analog converters (DACs) for the vector preparations may operate at a high bandwidth while the DACs for the VBSs may be quasi-static for multiple vectors. In general, including a DAC with a high bit precision at higher bandwidth is more difficult than designing one at a lower bandwidth. Thus, in some embodiments, the output vector elements may be more precise than what is allowed by the ADC, but the ADC will automatically perform some rounding to the output vector value up to the bit precision allowed by the ADC.

IV. Analog gain

[00205] The inventors have appreciated that low- precision (16-bit or less, 12-bit or less, 10-bit or less) fixed-point binary number representations are far less demanding than high- precision floating-point binary number representations in terms of chip area and power requirements. However, compared to high- precision floating-point representations (e.g., FP32) conventionally used in modern processors, low- precision fixed-point representations lead to significantly lower representational accuracy. Further, reducing the precision can reduce the dynamic range of an ADC. This results in a trade-off between representing the full dynamic range of a data vector, and representing smaller output values precisely. Described herein are techniques for performing matrix operations using hybrid analog-digital processors with low- precision ADCs. More specifically, some accelerator architectures according to some embodiments use techniques for shifting the readout window of an ADC. Shifting the readout window allows the ADC to focus the conversion on a portion of a signal that is likely to have the most amount of meaningful information, or at least to have some meaningful information. Leveraging these techniques, some accelerator architectures are designed to operate with low- precision fixed-point binary number representations, thus significantly reducing the power consumption compared to conventional high- precision floating-point representations.

[00206] FIG. 1-6A is a plot illustrating one such technique. The plot represents an example of a signal distribution that may be presented as input to an ADC (see curved labeled "input distribution"). (In this example, distributions are expressed as a function of voltage, but expressing distributions as a function of other quantities is also possible.) The input distribution depends upon the data that needs to be processed, and may vary dramatically from data to data. The dynamic range of the ADC (labeled "dynamic range") indicates the largest magnitude range the ADC can read. At one end of the dynamic range is the smallest signal (e.g., voltage) that the ADC can read. At the opposite end of the dynamic range is the largest signal that the ADC can read. In this particular example, the input distribution extends over only a fraction of the dynamic range of the ADC. In other circumstances, however, the input data may be such that the signal extends predominantly outside the dynamic range.

[00207] The techniques described herein allow for adjustments of the input distribution to match more closely the dynamic range of an ADC. In some embodiments, this may be accomplished by scaling (e.g., amplifying or attenuating) an analog signal prior to be provided as input to an ADC. In some embodiments, analog signals may be scaled so that the signal distribution extends from one end of the dynamic range to the other end of the dynamic range. In some embodiments, as illustrated in FIG. 1-6A, analog signals may be scaled so that a small portion of the tail of the distribution is outside the dynamic range. In the example of FIG. 1-6A, an amplification is applied that stretches the input distribution (the resulting distribution being labelled "amplified distribution") across the entire dynamic range, with portions of the tails being outside the dynamic range. Those inputs that are not within the dynamic range may be clipped into the minimum and maximum values of the ADC range. In FIG 1-6A this is illustrated with the vertical spikes in the "amplified distribution" at the endpoints of the ADC range. The inventors have recognized that, although cutting off portions of the distribution as shown in FIG. 1-6A may result in loss of information, the lost information may not be as meaningful as the information inside the dynamic range.

[00208] This concept is illustrated in FIG. 1-6B. Consider for example a matrix-vector multiplication $\vec{c} = A\vec{b}$ with integer (or fixed-point) arithmetic, where A is an n x n matrix. The entries of A are unsigned numbers represented with m_1 bits, and the entries of b are unsigned numbers represented with m_2 bits. The full accuracy representation of the entries of the output vector c requires b_c bits, where $b_c = \log_2(n) + m_1 + m_2$ (see bit string 1). However, if the output of such a computation is read with an ADC having $b_{adc} < b_c$ bits, there is a tradeoff between representing the full dynamic range of c, and representing smaller output values more precisely.

Considering by way of example n=16, $m_1 = 2$ and $m_2 = 4$, the full precision would be $b_c = 10$. As further shown in the figure, a 4-bit ADC configured to readout the full dynamic range of c will read the uppermost 4-bits of the output (see bit string 2). For some distributions of values, there may be useful information in this range. However, there is generally a high probability that the uppermost bits will not be set. As a result, the conversion does not produce any meaningful information.

[00209] In some embodiments, the ADC readout window can be shifted by amplifying the signal. Bit string 3 is obtained by amplifying the signal with a gain equal to 4, which shifts the readout window 2 bits towards the lowermost bit. As a result, useful information begins to appear inside the window. The resulting 4-bit output captures most of the meaningful information of the output having full precision. Bit string 4 is obtained by amplifying the signal with a gain equal to 64 (which in this case happens to saturate the ADC), which further shifts the readout window 3 bits towards the lowermost bit. The saturated result indicates that the true value is larger than the largest data value that is representable at the scale of these lowest bits, and is thus the most accurate representation of this input bit string on this data scale.

[00210] The inventors have appreciated the fact that useful information in the lower bits may be limited by the input quantization noise. Consider an example where the analog device programming a normalized matrix A (with values between 0 and 1) is precise to m1 bits and has an encoding noise at the level of $2^{-(m1+1)}$, and the analog device programming the normalized entries of b (with values between 0 and 1) is precise to m_2 bits and has an encoding noise at the level of $2^{-(m2+1)}$. For n=16, m_1=2 and m_2=4, the full precision would be b_c=10. But, consider the case where the maximum value of A (all values equal to 1) is programmed in, then the ($\log_2 n + m_2 + 1$)-th bit may be noisy. Similarly, in the case where the maximum value of b (all values equal to 1) is programmed in, then the ($\log_2 n + m_1 + 1$)-th bit may be noisy. Amplifying the signal to read this bit and the lesser significant bits may produce noisy results. However, it may be possible to recover the value of these noisy bits by taking multiple measurements and averaging. This encoding quantization noise essentially provides a limit to how much amplification one can apply to the signal before one encounters noisy bits. In some embodiments, the encoding quantization noise is not the only source of noise. The noise floor of the TIA and the optical receiver (in the case of a photonic processor) can limit the amount of amplification before noisy bits are read out.

[00211] In general, it may be difficult to predict how much gain is needed to maximize the collection of meaningful information. In some embodiments, the gain may be determined on the

basis of the statistics of a weight matrix (e.g., matrix A). This determination may be further based, in some embodiments, on either known statistics of the input vector data (matrix B) or bounds on the statistics of the input vector data. In one example, the gain may be determined as part of a training procedure. That is, the gain is a parameter that can be learned. In other embodiments, the gain parameter may be learned without using machine learning techniques. For example, it may be determined whether a particular gain level is sufficiently accurate for the training data, and if not, it may be tuned until it is. Further, it may be desirable to regularize the gain during a training procedure towards lower values, for example by adding a regularization loss term to the overall training loss function, to reduce the amount of energy needed when evaluating the computation in the analog processor or to enable the analog processor to be operated at a higher bandwidth while conserving the gain-bandwidth product of amplifiers.

[00212] In some embodiments, the gain settings may differ for each layer of a neural network. In some such embodiments, it may be desirable to learn the gain settings for each layer, and the gain may be varied when transitioning from one layer to the next. In some embodiments, when an accelerator receives new data, the controller may set the gain to the value used in training the new data.

[00213] An example of an accelerator based on the techniques described above is illustrated in FIG. 1-7A, in accordance with some embodiments. Similar to accelerator 1-100, accelerator 1-700 includes DAC unit 1-102, analog processor 1-104 and ADC unit 1-106. Accelerator 1-700 further includes analog scaling unit 1-103, which is placed between DAC unit 1-102 and analog processor 1-104, and analog scaling unit 1-105, which is placed between analog processor 1-104 and ADC unit 1-106. The components of FIG. 1-7A, and optionally other components, are collectively referred to herein as "circuitry."

[00214] Unlike accelerator 1-100, accelerator 1-700 is configured to perform mathematical operations in a single pass. Therefore, digital accumulator 1-110 is omitted. However, as described in detail further above, techniques for improving the accuracy of low- precision fixed-point representations may be used with accelerators arranged to perform multiple passes. In particular, multiple passes can be performed between different levels of precision of the input data and the values encoded in the analog processor.

[00215] The scaling described in connection with FIG. 1-6A may be performed anywhere along the data path from the DAC unit to the ADC unit. Analog scaling unit 1-103 may be configured to scale (e.g., amplify or attenuate) the analog signals before they are provided to the analog

processor. Analog scaling unit 1-105 may be configured to scale the analog signals after the analog processor 1-104. The analog scaling units may comprise any suitable type of controllable amplifier or attenuator, including voltage amplifiers, current amplifiers, power amplifiers, transimpedance amplifiers, transconductance amplifiers, voltage attenuators, current attenuators, and power attenuators, among others. In a photonic analog processor, as described in detail further below, the analog scaling unit can be one of the aforementioned analog scaling units, an optical amplifier, or a variable gain laser, among others. In the example of FIG. 1-7B, analog scaling unit 1-103 comprises a plurality of amplifiers or attenuators 1-702. Controller 1-120 sets the gains of these devices. (Gain is referred to herein to indicate amplification or attenuation, depending on whether the gain is greater or less than 1).

[00216] FIG. 1-8 is a flowchart illustrating an example of a method for performing a mathematical operation, in accordance with some embodiments. In some embodiments, accelerator 1-700 (FIG. 1-7A) may perform method 1-800, though any other suitable accelerator may be used. It should be appreciated that the steps of method 1-800 may be performed in the order depicted in FIG. 1-8, or in any other suitable order.

[00217] At step 1-801, accelerator 1-700 receives an input data set. Referring back to FIG. 1-2B, for example, the accelerator may receive the entries of matrix B.

[00218] At step 1-802, accelerator 1-700 generates a plurality of input analog signals based on the input data set received at step 1-801. Referring back to FIG. 1-7A, in some embodiments, the generation of step 1-804 involves converting the input data set to analog signals using DAC unit 1-102.

[00219] At step 1-804, controller 1-120 determines one or more scaling factors. As described above, the scaling factor(s) may be determined in some embodiments based on the set of parameters and the input data set. For example, the scaling factor may be determined based on statistics of the set of parameters and statistics of the input data set. Alternatively, the scaling factor may be determined based on statistics of the set of parameters and assumptions about the statistics of the input data set. In some embodiments, two scaling factors may be determined—one for analog scaling unit 1-103 and one for analog scaling unit 1-105.

[00220] As described above, there may be an optimum gain value that maximizes the accuracy of the computation being performed with the numeric representation. Though the optimum gain may not be known a priori, in some embodiments, the gain may be learned as part of a training procedure. FIG. 1-9A is a plot illustrating the accuracy, expressed in percentage, of a neural network performing an image classification task, as a function of the gain. As shown in the

figure, there is an optimum gain that maximizes the accuracy, which may be learned as part of a training procedure. In some embodiments, the optimum may be computed prior to performing method 1-800, and information indicative of the optimum gain may be stored in a memory or look-up table to be retrieved during methods 1-800.

[00221] At step 1-806, controller 1-120 sets the gain of analog scaling unit 1-103 and/or the gain of analog scaling unit 1-105 based on the scaling factor(s) determined at step 1-806.

[00222] At step 1-808, analog scaling unit 1-103 amplifies or attenuates the input analog signal produced by DAC unit 1-102 based on the scaling factor(s). As discussed above, this may cause the signal distribution to match the dynamic range of the ADCs more closely.

[00223] At step 1-810, controller 1-120 programs analog processor 1-104. More specifically, controller 1-120 programs analog processor 1-104 based on the set of parameters (e.g., based on the parameters of matrix A). Programming the analog processor may involve setting certain characteristics of the analog processor based on the parameters, as described above in connection with FIG. 1-2A and FIG. 1-3.

[00224] At step 1-812, analog processor 1-104 generates a plurality of output analog signals based on the plurality of input analog signals and the set of parameters with which the analog processor was programmed at step 1-810. This may involve, in some embodiments, performing a matrix multiplication or a convolution. As in the example of FIG. 1-3, the operation may be performed using a matrix decomposition.

[00225] At step 1-814, accelerator 1-700 amplifies or attenuates the analog signals output by DAC unit 1-102 with the gain or attenuation factor determined at step 1-806. The amplification or attenuation may be performed using analog scaling unit 1-105. In some embodiments, only one between steps 1-808 and 1-814 is performed. That is, signals are amplified or attenuated only before or after passing through analog processor 1-104. In other embodiments, both steps are performed.

[00226] At step 1-816, accelerator 1-700 generates an output data set based on the plurality of output analog signals obtained at step 1-814. In some embodiments, the generation of step 1-816 involves passing the output analog signals through ADC unit 1-106.

[00227] The inventors have appreciated that, in some embodiments, it may be beneficial to vary the gain or attenuation factor in a multi-layer (e.g., multi-pass) neural network model from layer to layer. In fact, the inventors have appreciated that the error of a low- precision fixed-point representation may be layer-dependent. FIG. 1-9B plots how the mean square error associated with a low- precision fixed-point representation varies as a function of the gain over three

different layers of a neural network. The mean square error is computed relative to a FP32 representation. In this particular example, setting the gain to 2 minimizes the mean square error of the second layer, setting the gain to 4 minimizes the mean square error of the third layer and setting the gain to 6 minimizes the mean square error of the fourth layer. This occurs primarily due to changing data distributions at different layers of the neural network model. Accordingly, in some embodiments, the gain may be varied from layer to layer. The gain may additionally be varied between any suitable sub-portion of the network that is assigned to the processor, including but not limited to tiles of matrices within a layer.

[00228] The accelerator of FIG. 1-10 combines aspects described in connection with the accelerator of FIG. 1-2A with aspects described in connection with the accelerator of FIG. 1-7A. Accelerator 1-1000 includes DAC unit 1-102, analog scaling unit 1-103, analog processor 1-104, analog scaling unit 1-105, ADC unit 1-106, digital scaling unit 1-108, digital accumulation unit 1-110 and controller 1-120. Accelerator 1-1000 is configured to perform multiple passes through analog processor 1-104 as described in connection with FIG. 1-2A, and is also configured to perform amplification or attenuation as described in connection with FIG. 1-7A. The components of FIG. 1-10, and optionally other components, are collectively referred to herein as "circuitry."

[00229] FIG. 1-11 is a flowchart illustrating an example of a method for performing a mathematical operation, in accordance with some embodiments. Method 1-1100 may be performed using accelerator 1-1000 (FIG. 1-10), though any other suitable accelerator may be used. Method 1-1100 is similar to method 1-300, though it further includes some of the steps of method 1-800.

V. Examples of optical analog-digital matrix processors

[00230] The inventors have developed a particular class of analog processors that uses light signals. The inventors have appreciated that, because light signals travel at the speed of light in the medium in which the light is traveling, latency is far less of a limitation than in electrical systems. Additionally, virtually no power is dissipated by increasing the distance traveled by the light signals, opening up new topologies and processor layouts that would not be feasible using electrical signals. Thus, photonic processors may be used in some embodiments as part of an accelerator.

[00231] Any of the techniques described above may be applied to accelerators that include photonic processors. One example of such an accelerator is illustrated in FIG. 1-12, in accordance with some embodiments. Accelerator 1-1200 includes a DAC unit 1-102, one or more lasers 1-1250, an optical encoder 1-1201, an optical scaling unit 1-1203, a photonic processor 1-1204, an optical scaling unit 1-1205, an optical receiver 1-1207, an electronic analog scaling unit 1-105, an ADC unit 1-106, a digital scaling unit 1-108, a digital accumulation unit 1-110 and a controller 1-120. Though using optical signals, accelerator 1-1200 may be configured to operate in a manner similar to that of accelerator 1-100 (FIG. 1-2A), in a manner similar to that of accelerator 1-700 (FIG. 1-7A), in a manner similar to that of accelerator 1-1000 (FIG. 1-10), or any suitable combination of these accelerators. Photonic processor 1-1204 may be designed to perform mathematical operations similar to those performed by analog processor 1-104 (FIG. 1-2A and FIG. 1-7A), but in the optical domain. Examples of photonic processors are described in detail further below. The components of FIG. 1-12 (whether optical or electronic), and optionally other components, are collectively referred to herein as "circuitry."

[00232] DAC unit 1-102, electronic analog scaling unit 1-105, ADC unit 1-106, digital scaling unit 1-108 and digital accumulation unit 1-110 have characteristics similar to those described in connection with FIG. 1-2A and FIG. 1-7A.

[00233] Laser 1-1250 provides a reference optical signal to optical encoder 1-1201. In some embodiments, controller 1-120 may control the output power of the laser. Optical encoder 1-1201 is configured to convert analog signal received from DAC unit 1-102 to optically encoded signals to be processed by photonic processor 1-1204. In some embodiments, the optical signals represent the value and sign of the associated bit string as an amplitude and a phase of an optical pulse. In some embodiments, the phase may be limited to a binary choice of either a zero phase shift or a π phase shift, representing a positive and negative value, respectively. Embodiments are not limited to real input vector values. Complex vector components may be represented by, for example, using more than two phase values when encoding the optical signal.

[00234] Similar to analog scaling unit 1-103 (FIG. 1-7A), optical scaling unit 1-1203 may be configured to scale the optical input signals produced by optical encoder 1-1201 to match more closely the dynamic range of the ADCs. Further, optical scaling unit 1-1205 may be configured to scale the optical outputs of processor 1-1204. As in FIG. 1-7A, analog scaling unit 1-105 may be configured to scale the received electrical outputs to match more closely the dynamic range of the ADCs. In some embodiments, analog scaling unit 1-105 is implemented with a transimpedance amplifier. In addition or in alternative, the signal distribution may be adjusted to

match more closely the dynamic range of the ADCs based on the output power of the laser. Effectively, the laser, which includes an optical amplification medium, may be utilized an amplifier. It should be appreciated that, while the example of FIG. 1-12 includes optical scaling unit 1-1203, optical scaling unit 1-1205, and electrical analog scaling unit 1-105, other embodiments may include only a subset of these units. In some embodiments, accelerator 1-1200 generates a plurality of amplified output optical signals using any combination of the following optical amplifiers: laser 1-1250, optical scaling unit 1-1203 and optical scaling 1-1205, as amplification or attenuation may be performed anywhere along the path to the ADCs.

[00235] Optical receiver 1-1207 is configured to convert optical signals into electrical signals. For example, optical receiver 1-1207 may include a plurality of photodetectors, such pn-junction photodiodes, pin-junction photodiodes, avalanche photodiodes or phototransistors. In some embodiments, the photodetectors include germanium photodiodes.

[00236] As in FIG. 1-2A, digital scaling unit 1-108 may be configured to scale partial output data sets, and digital accumulation unit 1-110 may be configured to accumulate scaled partial output data sets.

[00237] Photonic processor 1-1204 may be implemented in any of numerous ways. As described in connection with analog processor 1-104, photonic processor 1-1204 may be designed to perform matrix operations based on a matrix decomposition, examples of which are provided below. Different processor architectures may be used to encode a weight matrix. In some embodiments, this may be accomplished programmable photonic devices, such as optical modulators. Examples of optical modulators are optical resonant modulators, including ring and disk modulators. Another example is a Franz-Keldysh modulator. In some embodiments, programming a photonic processor based on the parameters of a matrix involves setting optical characteristics for the programmable photonic devices. For example, programming an optical resonator may involve programming the resonant frequency of the resonator, and programming a Franz-Keldysh modulator may involve programming the absorption of the modulator.

[00238] Other implementations of the photonic processor are now discussed.

[00239] FIG. 2-1 illustrates a photonic processing system 2-100 that may be used to implement some of the components of accelerator 1-1200, in accordance with some embodiments. Photonic processing system 2-100 includes an optical encoder 2-101, a photonic processor 2-103, an optical receiver 2-105, and a controller 2-107. Optical encoder 2-100 serves as optical encoder 1-1201 (FIG. 1-12), photonic processor 2-103 serves as photonic processor 1-1204, optical receiver 2-105 serves as optical receiver 1-1207 and controller 2-107 serves as controller 1-120.

Though not illustrated in FIG. 2-1, in some embodiments an optical analog scaling unit may be inserted between optical encoder 2-101 and a photonic processor 2-103. Additionally, or alternatively, an optical analog scaling unit may be inserted between photonic processor 2-103 and optical receiver 2-105.

[00240] The photonic processing system 2-100 receives, for example from an external processor (e.g., a CPU), an input vector represented by a group of input bit strings and produces an output vector represented by a group of output bit strings. For example, if the input vector is an *n*-dimensional vector, the input vector may be represented by *n* separate bit strings, each bit string representing a respective component of the vector. The input bit string may be received as an electrical or optical signal from the external processor and the output bit string may be transmitted as an electrical or optical signal to the external processor. In some embodiments, the controller 2-107 does not necessarily output an output bit string after every process iteration. Instead, the controller 2-107 may use one or more output bit strings to determine a new input bit stream to feed through the components of the photonic processing system 2-100. In some embodiments, the output bit string itself may be used as the input bit string for a subsequent iteration of the process implemented by the photonic processing system 2-100. In other embodiments, multiple output bit streams are combined in various ways to determine a subsequent input bit string. For example, one or more output bit strings may be summed together as part of the determination of the subsequent input bit string.

[00241] The optical encoder 2-101 outputs *n* separate optical pulses that are transmitted to the photonic processor 2-103. Each output of the optical encoder 2-101 is coupled one-to-one to a single input of the photonic processor 2-103. In some embodiments, the optical encoder 2-101 may be disposed on the same substrate as the photonic processor 2-103 (e.g., the optical encoder 2-101 and the photonic processor 2-103 are on the same chip). In such embodiments, the optical signals may be transmitted from the optical encoder 2-101 to the photonic processor 2-103 in waveguides, such as silicon photonic waveguides. In other embodiments, the optical encoder 2-101 may be disposed on a separate substrate from the photonic processor 2-103. In such embodiments, the optical signals may be transmitted from the optical encoder 2-101 to the photonic processor 103 in optical fiber.

[00242] The photonic processor 2-103 performs the multiplication of the input vector by a matrix *M*. As described in detail below, the matrix *M* may be decomposed into three matrices using a combination of a singular value decomposition (SVD) and a unitary matrix

decomposition. In some embodiments, the unitary matrix decomposition is performed with operations similar to Givens rotations in QR decomposition. For example, an SVD in combination with a Householder decomposition may be used. The decomposition of the matrix *M* into three constituent parts may be performed by the controller 2-107 and each of the constituent parts may be implemented by a portion of the photonic processor 2-103. In some embodiments, the photonic processor 2-103 includes three parts: a first array of variable beam splitters (VBSs) configured to implement a transformation on the array of input optical pulses that is equivalent to a first matrix multiplication (see, e.g., the first matrix module 2-301 of FIG. 2-3); a group of controllable optical elements configured to adjust the intensity and/or phase of each of the optical pulses received from the first array, the adjustment being equivalent to a second matrix multiplication by a diagonal matrix (see, e.g., the second matrix module 2-303 of FIG. 2-3); and a second array of VBSs configured to implement a transformation on the optical pulses received from the group of controllable electro-optical element, the transformation being equivalent to a third matrix multiplication (see, e.g., the third matrix module 2-305 of FIG. 3).

[00243] The photonic processor 2-103 outputs *n* separate optical pulses that are transmitted to the optical receiver 2-105. Each output of the photonic processor 2-103 is coupled one-to-one to a single input of the optical receiver 2-105. In some embodiments, the photonic processor 2-103 may be disposed on the same substrate as the optical receiver 2-105 (e.g., the photonic processor 2-103 and the optical receiver 2-105 are on the same chip). In such embodiments, the optical signals may be transmitted from the photonic processor 2-103 to the optical receiver 2-105 in silicon photonic waveguides. In other embodiments, the photonic processor 2-103 may be disposed on a separate substrate from the optical receiver 2-105. In such embodiments, the optical signals may be transmitted from the photonic processor 103 to the optical receiver 2-105 in optical fibers.

[00244] The optical receiver 2-105 receives the *n* optical pulses from the photonic processor 2-103. Each of the optical pulses is then converted to electrical signals. In some embodiments, the intensity and phase of each of the optical pulses is measured by optical detectors within the optical receiver. The electrical signals representing those measured values are then output to the controller 2-107.

[00245] The controller 2-107 includes a memory 2-109 and a processor 2-111 for controlling the optical encoder 2-101, the photonic processor 2-103 and the optical receiver 2-105. The memory 2-109 may be used to store input and output bit strings and measurement results from the optical receiver 2-105. The memory 2-109 also stores executable instructions that, when executed by

the processor 2-111, control the optical encoder 2-101, perform the matrix decomposition algorithm, control the VBSs of the photonic processor 103, and control the optical receivers 2-105. The memory 2-109 may also include executable instructions that cause the processor 2-111 to determine a new input vector to send to the optical encoder based on a collection of one or more output vectors determined by the measurement performed by the optical receiver 2-105. In this way, the controller 2-107 can control an iterative process by which an input vector is multiplied by multiple matrices by adjusting the settings of the photonic processor 2-103 and feeding detection information from the optical receiver 2-105 back to the optical encoder 2-101. Thus, the output vector transmitted by the photonic processing system 2-100 to the external processor may be the result of multiple matrix multiplications, not simply a single matrix multiplication.

[00246] In some embodiments, a matrix may be too large to be encoded in the photonic processor using a single pass. In such situations, one portion of the large matrix may be encoded in the photonic processor and the multiplication process may be performed for that single portion of the large matrix. The results of that first operation may be stored in memory 2-109. Subsequently, a second portion of the large matrix may be encoded in the photonic processor and a second multiplication process may be performed. This "chunking" of the large matrix may continue until the multiplication process has been performed on all portions of the large matrix. The results of the multiple multiplication processes, which may be stored in memory 2-109, may then be combined to form the final result of the multiplication of the input vector by the large matrix.

[00247] In other embodiments, only collective behavior of the output vectors is used by the external processor. In such embodiments, only the collective result, such as the average or the maximum/minimum of multiple output vectors, is transmitted to the external processor.

A. Optical Encoders

[00248] Referring to FIG. 2-2, the optical encoder includes at least one light source 2-201, a power tree 2-203, an amplitude modulator 2-205, a phase modulator 2-207, a digital to analog converter (DAC) 2-209 associated with the amplitude modulator 2-205, and a DAC 2-211 associated with the phase modulator 2-207, according to some embodiments. DACs 2-209 and 2-211 may be part of DAC unit 1-102 (FIG. 1-12). While the amplitude modulator 2-205 and phase modulator 2-207 are illustrated in FIG. 2-2 as single blocks with n inputs and n outputs (each of the inputs and outputs being, for example, a waveguide), in some embodiments each

waveguide may include a respective amplitude modulator and a respective phase modulator such that the optical encoder includes *n* amplitude modulators and *n* phase modulators. Moreover, there may be an individual DAC for each amplitude and phase modulator. In some embodiments, rather than having an amplitude modulator and a separate phase modulator associated with each waveguide, a single modulator may be used to encode both amplitude and phase information. While using a single modulator to perform such an encoding limits the ability to precisely tune both the amplitude and phase of each optical pulse, there are some encoding schemes that do not require precise tuning of both the amplitude and phase of the optical pulses. Such a scheme is described later herein.

[00249] The light source 2-201 may be any suitable source of coherent light. In some embodiments, the light source 2-201 may be a diode laser or a vertical-cavity surface emitting lasers (VCSEL). In some embodiments, the light source 2-201 is configured to have an output power greater than 10 mW, greater than 25 mW, greater than 50 mW, or greater than 75mW. In some embodiments, the light source 2-201 is configured to have an output power less than 100 mW. The light source 2-201 may be configured to emit a continuous wave of light or pulses of light ("optical pulses") at one or more wavelengths (e.g., the C-band or O-band). The temporal duration of the optical pulses may be, for example, about 100 ps.

[00250] While light source 2-201 is illustrated in FIG. 2-2 as being on the same semiconductor substrate as the other components of the optical encoder, embodiments are not so limited. For example, the light source 2-201 may be a separate laser packaging that is edge-bonded or surface-bonded to the optical encoder chip. Alternatively, the light source 2-201 may be completely off-chip and the optical pulses may be coupled to a waveguide 2-202 of the optical encoder 2-101 via an optical fiber and/or a grating coupler.

[00251] The light source 2-201 is illustrated as two light sources 2-201a and 2-201b, but embodiments are not so limited. Some embodiments may include a single light source. Including multiple light sources 201a-b, which may include more than two light sources, can provide redundancy in case one of the light sources fails. Including multiple light sources may extend the useful lifetime of the photonic processing system 2-100. The multiple light sources 2-201a-b may each be coupled to a waveguide of the optical encoder 2-101 and then combined at a waveguide combiner that is configured to direct optical pulses from each light source to the power tree 2-203. In such embodiments, only one light source is used at any given time.

[00252] Some embodiments may use two or more phase-locked light sources of the same wavelength at the same time to increase the optical power entering the optical encoder system. A

small portion of light from each of the two or more light sources (e.g., acquired via a waveguide tap) may be directed to a homodyne detector, where a beat error signal may be measured. The bear error signal may be used to determine possible phase drifts between the two light sources. The beat error signal may, for example, be fed into a feedback circuit that controls a phase modulator that phase locks the output of one light source to the phase of the other light source. The phase-locking can be generalized in a master-slave scheme, where $N \geq 1$ slave light sources are phase-locked to a single master light source. The result is a total of $N + 1$ phase-locked light sources available to the optical encoder system.

[00253] In other embodiments, each separate light source may be associated with light of different wavelengths. Using multiple wavelengths of light allows some embodiments to be multiplexed such that multiple calculations may be performed simultaneously using the same optical hardware.

[00254] The power tree 2-203 is configured to divide a single optical pulse from the light source 2-201 into an array of spatially separated optical pulses. Thus, the power tree 2-203 has one optical input and n optical outputs. In some embodiments, the optical power from the light source 2-201 is split evenly across n optical modes associated with n waveguides. In some embodiments, the power tree 2-203 is an array of 50:50 beam splitters 2-801, as illustrated in FIG. 2-8. The number "depth" of the power tree 2-203 depends on the number of waveguides at the output. For a power tree with n output modes, the depth of the power tree 2-203 is $\text{ceil}(log_2(n))$. The power tree 2-203 of FIG. 2-8 only illustrates a tree depth of three (each layer of the tree is labeled across the bottom of the power tree 2-203). Each layer includes 2^{m-1} beam splitters, where m is the layer number. Consequently, the first layer has a single beam splitter 2-801a, the second layer has two beam splitters 2-801b – 2-801c, and the third layer has four beam splitters 2-801d – 2-801g.

[00255] While the power tree 2-203 is illustrated as an array of cascading beam splitters, which may be implemented as evanescent waveguide couplers, embodiments are not so limited as any optical device that converts one optical pulse into a plurality of spatially separated optical pulses may be used. For example, the power tree 2-203 may be implemented using one or more multimode interferometers (MMI), in which case the equations governing layer width and depth would be modified appropriately.

[00256] No matter what type of power tree 2-203 is used, it is likely that manufacturing a power tree 2-203 such that the splitting ratios are precisely even between the n output modes will be difficult, if not impossible. Accordingly, adjustments can be made to the setting of the amplitude

modulators to correct for the unequal intensities of the n optical pulses output by the power tree. For example, the waveguide with the lowest optical power can be set as the maximum power for any given pulse transmitted to the photonic processor 2-103. Thus, any optical pulse with a power higher than the maximum power may be modulated to have a lower power by the amplitude modulator 2-205, in addition to the modulation to the amplitude being made to encode information into the optical pulse. A phase modulator may also be placed at each of the n output modes, which may be used to adjust the phase of each output mode of the power tree 2-203 such that all of the output signals have the same phase.

[00257] Alternatively or additionally, the power tree 2-203 may be implemented using one or more Mach-Zehnder Interferometers (MZI) that may be tuned such that the splitting ratios of each beam splitter in the power tree results in substantially equal intensity pulses at the output of the power tree 2-203.

[00258] The amplitude modulator 2-205 is configured to modify, based on a respective input bit string, the amplitude of each optical pulse received from the power tree 2-203. The amplitude modulator 2-205 may be a variable attenuator or any other suitable amplitude modulator controlled by the DAC 2-209, which may further be controlled by the controller 2-107. Some amplitude modulators are known for telecommunication applications and may be used in some embodiments. In some embodiments, a variable beam splitter may be used as an amplitude modulator 2-205, where only one output of the variable beam splitter is kept and the other output is discarded or ignored. Other examples of amplitude modulators that may be used in some embodiments include traveling wave modulators, cavity-based modulators, Franz-Keldysh modulators, plasmon-based modulators, 2-D material-based modulators and nano-opto-electro-mechanical switches (NOEMS).

[00259] The phase modulator 2-207 is configured to modify, based on the respective input bit string, the phase of each optical pulse received from the power tree 2-203. The phase modulator may be a thermo-optic phase shifter or any other suitable phase shifter that may be electrically controlled by the 2-211, which may further be controlled by the controller 2-107.

[00260] While FIG. 2-2 illustrates the amplitude modulator 2-205 and phase modulator 2-207 as two separate components, they may be combined into a single element that controls both the amplitudes and phases of the optical pulses. However, there are advantages to separately controlling the amplitude and phase of the optical pulse. Namely, due to the connection between amplitude shifts and phase shifts via the Kramers-Kronenig relations, there is a phase shift associated with any amplitude shift. To precisely control the phase of an optical pulse, the phase

shift created by the amplitude modulator 2-205 should be compensated for using the phase modulator 2-207. By way of example, the total amplitude of an optical pulse exiting the optical encoder 2-101 is $A = a_0 a_1 a_2$ and the total phase of the optical pulse exiting the optical encoder is $\theta = \Delta\theta + \Delta\varphi + \varphi$, where a_0 is the input intensity of the input optical pulse (with an assumption of zero phase at the input of the modulators), $a_1$is the amplitude attenuation of the amplitude modulator 2-205, $\Delta\theta$ is the phase shift imparted by the amplitude modulator 2-205 while modulating the amplitude, $\Delta\varphi$ is the phase shift imparted by the phase modulator 2-207, a_2 is the attenuation associated with the optical pulse passing through the phase modulator 2-209, and φ is the phase imparted on the optical signal due to propagation of the light signal. Thus, setting the amplitude and the phase of an optical pulse is not two independent determinations. Rather, to accurately encode a particular amplitude and phase into an optical pulse output from the optical encoder 2-101, the settings of both the amplitude modulator 2-205 and the phase modulator 2-207 should be taken into account for both settings.

[00261] In some embodiments, the amplitude of an optical pulse is directly related to the bit string value. For example, a high amplitude pulse corresponds to a high bit string value and a low amplitude pulse corresponds to a low bit string value. The phase of an optical pulse encodes whether the bit string value is positive or negative. In some embodiments, the phase of an optical pulse output by the optical encoder 2-101 may be selected from two phases that are 180 degrees (π radians) apart. For example, positive bit string values may be encoded with a zero degree phase shift and negative bit string values may be encoded with a 180 degree (π radians) phase shift. In some embodiments, the vector is intended to be complex-valued and thus the phase of the optical pulse is chosen from more than just two values between 0 and 2π.

[00262] In some embodiments, the controller 2-107 determines the amplitude and phase to be applied by both the amplitude modulator 2-205 and the phase modulator 2-207 based on the input bit string and the equations above linking the output amplitude and output phase to the amplitudes and phases imparted by the amplitude modulator 2-204 and the phase modulator 2-207. In some embodiments, the controller 2-107 may store in memory 2-109 a table of digital values for driving the amplitude modulator 2-205 and the phase modulator 2-207. In some embodiments, the memory may be placed in close proximity to the modulators to reduce the communication temporal latency and power consumption.

[00263] DAC 2-209, associated with and communicatively coupled to the amplitude modulator 2-205, receives the digital driving value from the controller 2-107 and converts the digital driving value to an analog voltage that drives the amplitude modulator 2-205. Similarly, the

DAC 2-211, associated with and communicatively coupled to the phase modulator 2-207, receives the digital driving value from the controller 2-107 and converts the digital driving value to an analog voltage that drives the phase modulator 2-207. In some embodiments, the DAC may include an amplifier that amplifies the analog voltages to sufficiently high levels to achieve the desired extinction ratio within the amplitude modulators (e.g., the highest extinction ratio physically possible to implement using the particular phase modulator) and the desired phase shift range within the phase modulators (e.g., a phase shift range that covers the full range between 0 and 2π). While the DAC 2-209 and the DAC 2-211 are illustrated in FIG. 2-2 as being located in and/or on the chip of the optical encoder 2-101, in some embodiments, the DACs 2-209 and 2-211 may be located off-chip while still being communicatively coupled to the amplitude modulator 2-205 and the phase modulator 2-207, respectively, with electrically conductive traces and/or wires.

[00264] After modulation by the amplitude modulator 2-205 and the phase modulator 2-207, the *n* optical pulses are transmitted from the optical encoder 2-101 to the photonic processor 2-103.

B. Photonic Processors

[00265] Referring to FIG. 2-3, the photonic processor 2-103 implements matrix multiplication on an input vector represented by the *n* input optical pulse and includes three main components: a first matrix module 2-301, a second matrix module 2-303, and a third matrix module 2-305. In some embodiments, as discussed in more detail below, the first matrix module 2-301 and the third matrix module 2-305 include an interconnected array of programmable, reconfigurable, variable beam splitters (VBSs) configured to transform the *n* input optical pulses from an input vector to an output vector, the components of the vectors being represented by the amplitude and phase of each of the optical pulses. In some embodiments, the second matrix module 2-303 includes a group of electro-optic elements.

[00266] The matrix by which the input vector is multiplied, by passing the input optical pulses through the photonic processor 2-103, is referred to as A. The matrix A is a general *m* x *n* known to the controller 2-107 as the matrix that should be implemented by the photonic processor 2-103. As such, the controller 2-107 decomposes the matrix A using a singular value decomposition (SVD) such that the matrix A is represented by three constituent matrices: $A = V^T \Sigma U$, where *U* and *V* are real orthogonal *n* x *n* and *m* x *m* matrices, respectively ($U^T U = UU^T = I$ and $V^T V = VV^T = I$), and Σ is an *m* x *n* diagonal matrix with real entries. The superscript "T" in all equations represents the transpose of the associated matrix.

Determining the SVD of a matrix is known and the controller 2-107 may use any suitable technique to determine the SVD of the matrix A. In some embodiments, the matrix A is a complex matrix, in which case the matrix A can be decomposed into $A = V^{T}\Sigma U$, where V and U are complex unitary $n \times n$ and $m \times m$ matrices, respectively $U^{\dagger}U = UU^{\dagger} = I$ and $V^{\dagger}V = VV^{\dagger} = I$), and Σ is an $m \times n$ diagonal matrix with real or complex entries. The values of the diagonal singular values may also be further normalized such that the maximum absolute value of the singular values is 1.

[00267] Once the controller 2-107 has determined the matrices U, Σ and V for the matrix A, in the case where the matrices U and V are orthogonal real matrices, the control may further decompose the two orthogonal matrices U and V into a series of real-valued Givens rotation matrices. A Givens rotation matrix $G(i, j, \theta)$ is defined component-wise by the following equations:

$$g_{kk} = 1 \text{ for } k \neq i, j$$

$$g_{kk} = cos(\theta) \text{ for } k = i, j$$

$$g_{ij} = -g_{ji} = -sin(\theta),$$

$$g_{kl} = 0 \text{ otherwise,}$$

where g_{ij} represents the element in the i-th row and j-th column of the matrix G and θ is the angle of rotation associated with the matrix. Generally, the matrix G is an arbitrary 2×2 unitary matrix with determinant 1 (SU(2) group) and it is parameterized by two parameters. In some embodiments, those two parameters are the rotation angle θ and another phase value ϕ. Nevertheless, the matrix G can be parameterized by other values other than angles or phases, e.g. by reflectivities/transmissivities or by separation distances (in the case of NOEMS).

[00268] Algorithms for expressing an arbitrary real orthogonal matrix in terms of a product of sets of Givens rotations in the complex space are provided in M. Reck, *et al.*, "Experimental realization of any discrete unitary operator," Physical Review Letters **73**, 58 (1994) ("Reck"), and W. R. Clements, *et al.*, "Optimal design for universal multiport interferometers," Optica **3**, 12 (2016) ("Clements"), both of which are incorporated herein by reference in their entirety and at least for their discussions of techniques for decomposing a real orthogonal matrix in terms of Givens rotations. (In the case that any terminology used herein conflicts with the usage of that

terminology in Reck and/or Clements, the terminology should be afforded a meaning most consistent with how a person of ordinary skill would understand its usage herein.). The resulting decomposition is given by the following equation:

$$U = D\prod_{k=1}^{n}\prod_{(i,j)\in S_k} G\left(i,j,\theta_{ij}^{(k)}\right),$$

[00269] where U is an n x n orthogonal matrix, S_k is the set of indices relevant to the k-th set of Givens rotations applied (as defined by the decomposition algorithm), $\theta_{ij}^{(k)}$ represents the angle applied for the Givens rotation between components i and j in the k-th set of Givens rotations, and D is a diagonal matrix of either +1 or -1 entries representing global signs on each component. The set of indices S_k is dependent on whether n is even or odd. For example, when n is even:

$$S_k = \{(1,2),(3,4),\ldots,(n-1,n)\} \text{ for odd k}$$

$$S_k = \{(2,3),(4,5),\ldots,(n-2,n-1)\} \text{ for even k}$$

[00270] When n is odd:

$$S_k = \{(1,2),(3,4),\ldots,(n-2,n-1)\} \text{ for odd k}$$

$$S_k = \{(2,3),(4,5),\ldots,\ (n-1,n)\} \text{ for even k}$$

[00271] By way of example and not limitation, the decomposition of a 4 x 4 orthogonal matrix can be represented as:

$$U = DG\left(1,2,\theta_{12}^{(1)}\right)G\left(3,4,\theta_{34}^{(1)}\right)G\left(2,3,\theta_{23}^{(2)}\right)G\left(1,2,\theta_{12}^{(3)}\right)G\left(3,4,\theta_{34}^{(3)}\right)G\left(2,3,\theta_{23}^{(4)}\right)$$

[00272] A brief overview of one embodiment of an algorithm for decomposing an n x n matrix U in terms of n sets of real-valued Givens rotations, which may be implemented using the controller 2-107, is as follows:

```
U' ← U
For i from 1 to n−1:
    If i is odd:
        For j = 0 to i−1:
            Find G^T_{i−j,i−j+1}(θ) that nullifies element U'_{n−j,i−j} i.e. θ = tan^{−1}(U'_{n−j,i−j}/U'_{n−j,i−j+1})
            U' ← U' G^T_{i−j,i−j+1}(θ)
    Else if i is even:
        For j = 1 to i:
            Find G_{n−j−1,n−j−i}(θ) that nullifies element U'_{n−j−1,j} i.e. θ = tan^{−1}(−U'_{n−j−1,j}/U'_{n−j−1,j})
            U' ← G_{n−j−1,n−j−i}(θ) U'
```

[00273] The resultant matrix U' of the above algorithm is lower triangular and is related to the original matrix U by the equation:

$$U = \prod_{(j,k)\in S_L} G^T{}_{jk}\, U' \prod_{(j,k)\in S_R} G_{jk} = D_U \prod_{(j,k)\in S} G_{jk} \quad ,$$

where the label S_L labels the set of two modes connected by the VBS to the left of U' and the label S_R labels the set of two modes connected by the VBS to the right of U'. Because U is an orthogonal matrix, U' is a diagonal matrix with {-1,1} entries along the diagonal. This matrix, $U' = D_U$, is referred to as a "phase screen."

[00274] The next step of the algorithm, is to repeatedly find $G^T{}_{jk}(\theta_1)D_U = D_U G_{jk}(\theta_2)$ which is accomplished using the following algorithm, which may be implemented using the controller 2-107:

```
For every (j,k) in S_L:
    If U'_{j,j} and U'_{k,k} have different signs:
        θ2 = −θ1
    Else:
        θ2 = θ1
```

[00275] The above algorithm may also be used to decompose V and/or V^T to determine the m layers of VBS values and the associated phase screen.

[00276] The above concept of decomposing an orthogonal matrix into real-valued Givens rotation matrices can be expanded to complex matrices, e.g., unitary matrices rather than orthogonal matrices. In some embodiments, this may be accomplished by including an

additional phase in the parameterization of the Givens rotation matrices. Thus, a general form of the Givens matrices with the addition of the additional phase term is T(i,j,θ,ϕ), where

$$t_{kk} = 1 \text{for } k \neq i, j,$$

$$t_{ii} = e^{i\phi} cos(\theta),$$

$$t_{jj} = cos(\theta),$$

$$t_{ij} = -sin(\theta),$$

$$t_{ji} = e^{i\phi} sin(\theta),$$

$$t_{kl} = 0 \text{ otherwise},$$

where t_{ij} represents the i-th row and j-th column of the matrix T, θ is the angle of rotation associated with the matrix, and ϕ is the additional phase. Any unitary matrix can be decomposed into matrices of the type T(i,j,θ,ϕ). By making the choice to set the phase $\phi = 0$, the conventional real-valued Givens rotation matrices described above are obtained. If, instead, the phase $\phi = \pi$, then a set of matrices known as Householder matrices are obtained. A Householder matrix, H, has the form $H = I - (v \otimes v)$, where I is the $n \times n$ identity matrix, v is a unit vector, and $\otimes$ is the outer product. Householder matrices represent reflections about a hyperplane orthogonal to the unit vector v. In this parameterization the hyperplane is a two-dimensional subspace, rather than an $n - 1$ dimensional subspace as is common in defining Householder matrices for the QR decomposition. Thus, a decomposition of a matrix into Givens rotations is equivalent to a decomposition of the matrix into Householder matrices.

[00277] Based on the aforementioned decomposition of an arbitrary unitary matrix into a restricted set of Givens rotations, any unitary matrix can be implemented by a particular sequence of rotations and phase shifts. And in photonics, rotations may be represented by variable beam splitters (VBS) and phase shifts are readily implemented using phase modulators. Accordingly, for the n optical inputs of the photonic processor 2-103, the first matrix module 2-301 and the third matrix module 2-305, representing the unitary matrices of the SVD of the matrix A may be implemented by an interconnected array of VBSs and phase shifters. Because of the parallel nature of simultaneously passing n optical pulses through a VBS array, matrix multiplication of dimension (I,K) x (K,J) can be performed in O(IJK/n^2) time. The second matrix

module 2-303 is a diagonal matrix of the SVD of the matrix *A* combined with the diagonal matrices *D* associated with each of the orthogonal matrices of the SVD. As mentioned above, each matrix D is referred to as a "phase screen" and can be labeled with a subscript to denote whether it is the phase screen associated with the matrix *U* or the matrix *V*. Thus, the second matrix module 303 is the matrix $\Sigma' = D_V \Sigma D_U$. Factoring these phase screens out of the matrix U and the matrix V, we obtain the matrix U and Vin Fig 2-3.

[00278] In some embodiments, the VBS unit cell of the photonic processor 2-103 associated with the first matrix module 2-301 and the third matrix module 2-305 may be a Mach-Zehnder interferometer (MZI) with an internal phase shifter. In other embodiments, the VBS unit cell may be a microelectromechanical systems (MEMS) actuator. An external phase shifter may be used in some embodiments to implement the additional phase needed for the Givens rotations.

[00279] The second matrix module 2-303, representing the diagonal matrix $D_V \Sigma D_U$ may be implemented using an amplitude modulator and a phase shifter. In some embodiments, a VBS may be used to split off a portion of light that can be dumped to variably attenuate an optical pulse. Additionally or alternatively, a controllable gain medium may be used to amplify an optical signal. For example, GaAs, InGaAs, GaN , or InP may be used as an active gain medium for amplifying an optical signal. Other active gain processes such as the second harmonic generation in materials with crystal inversion symmetric, e.g. KTP and lithium niobate, and the four-wave mixing processes in materials that lack inversion symmetry, e.g. silicon, can also be used. A phase shifter in each optical mode may be used to apply either a zero or a π phase shift, depending on the phase screen being implemented. In some embodiments, only a single phase shifter for each optical mode is used rather than one phase shifter for each phase screen. This is possible because each of the matrices D_V, Σ, *and* D_U are diagonal and therefore commute. Thus, the value of each phase shifter of the second matrix module 2-303 of the photonic processor 2-103 is the result of the product of the two phase screens: $D_V D_U$.

[00280] Referring to FIG. 2-4, the first and third matrix module 2-301 and 2-305 are implemented as an array of VBSs 2-401, according to some embodiments. For the sake of simplicity, only *n=6* input optical pulses (the number of rows) are illustrated resulting in a "circuit depth" (e.g., the number of columns) equal to the number of input optical pulses (e.g., six). For the sake of clarity, only a single VBS 2-401 is labeled with a reference numeral. The VBS are labeled, however, with subscripts that identify which optical modes are being mixed by a particular VBS and a super script labeling the associated column. Each VBS 2-401 implements

a complex Givens rotation, T(i,j,θ,ϕ), as discussed above, where i and j are equivalent to the subscript labels of the VBSs 2-401, θ is the rotation angle of the Givens rotation, and ϕ is the additional phase associated with the generalized rotation.

[00281] Referring to FIG. 2-5, each VBS 2-401 may be implemented using a MZI 2-510 and at least one external phase shifter 2-507. In some embodiments, a second external phase shifter 2-509 may also be included. The MZI 2-510 includes a first evanescent coupler 2-501 and a second evanescent coupler 2-503 for mixing the two input modes of the MZI 2-510. An internal phase shifter 2-505 modulates the phase θ in one arm of the MZI 2-510 to create a phase difference between the two arms. Adjusting the phase θ causes the intensity of light output by the VBS 2-401 to vary from one output mode of the MZI 2-510 to the other thereby creating a beam splitter that is controllable and variable. In some embodiments, a second internal phase shifter can be applied on the second arm. In this case, it is the difference between the two internal phase shifters that cause the output light intensity to vary. The average between the two internal phases will impart a global phase to the light that enter mode i and mode j. Thus the two parameters θ and ϕ may each be controlled by a phase shifter. In some embodiments, the second external phase shifter 2-509 may be used to correct for an unwanted differential phase across the output modes of the VBS due to static phase disorder.

[00282] In some embodiments, the phase shifters 2-505, 2-507 and 2-509 may include a thermo-optic, electro-optic, or optomechanic phase modulator. In other embodiments, rather than including an internal phase modulator 505 within an MZI 510, a NOEMS modulator may be used.

[00283] In some embodiments, the number of VBSs grows with the size of the matrix. The inventors have appreciated that controlling a large number of VBSs can be challenging and there is a benefit to sharing a single control circuit among multiple VBSs. An example of a parallel control circuit that may be used to control multiple VBSs is a digital-to-analog converter that receives as an input a digital string that encodes the analog signal to be imparted on a specific VBS. In some embodiments, the circuit also receives a second input the address of the VBS that is to be controlled. The circuit may then impart analog signals on the addressed VBS. In other embodiments, the control circuit may automatically scan through a number of VBSs and impart analog signals on the multiple VBSs without being actively given an address. In this case, the addressing sequence is predefined such that it traverses the VBS array in known order.

[00284] Referring to FIG. 2-6, the second matrix module 2-303 implements multiplication by the diagonal matrix $\Sigma' = D_V \Sigma D_U$. This may be accomplished using two phase shifters 2-601 and

2-605 to implement the two phase screens and an amplitude modulator 2-603 to adjust the intensity of an associate optical pulse by an amount η. As mentioned above, in some embodiments only a single phase modulator 2-601 may be used, as the two phase screens can be combined together since the three constituent matrices that form Σ' are diagonal and therefore commute.

[00285] In some embodiments, the amplitude modulators 2-603 may be implemented using an attenuator and/or an amplifier. If the value of the amplitude modulation η is greater than one, the optical pulse is amplified. If the value of the amplitude modulation η is less than one, the optical pulse is attenuated. In some embodiments, only attenuation is used. In some embodiments, the attenuation may be implemented by a column of integrated attenuators. In other embodiments, as illustrated in FIG. 2-7, the attenuation 2-603 may be implemented using a MZI that includes two evanescent couplers 2-701 and 2-703 and a controllable internal phase shifter 2-705 to adjust how much of the input light is transmitted from the input of the MZI to a first output port 2-709 of the MZI. A second output port 2-707 of the MZI may be ignored, blocked or dumped.

[00286] In some embodiments, the controller 2-107 controls the value of each phase shifter in the photonic processor 2-103. Each phase shifter discussed above may include a DAC similar to the DACs discussed in connection with the phase modulator 2-207 of the optical encoder 2-101.

[00287] The photonic processor 2-103 can include any number of input modes, but the size and complexity of the interconnected VBS arrays 2-301 and 2-305 will increase as the number of input modes increases. For example, if there are n input optical modes, then the photonic processor 2-103 will have a circuit depth of $2n+1$, where the first matrix module 2-301 and the second matrix module 2-305 each has a circuit depth n and the second matrix module 2-303 has a circuit depth of one. Importantly, the complexity in time of performing a single matrix multiplication linear with the number of columns of input data . If the dimensions of the matrix multiplication being processed are (I, K) x (K x J), the time complexity of performing this multiplication on an optical processor with n-modes is $O(IKJ/n^2)$. In some embodiments, this low order complexity afforded by the parallelization results in energy and time efficiencies that cannot be obtained using conventional electrical processors.

[00288] It is noted that, while embodiments described herein illustrate the photonic processor 2-103 as having n inputs and n outputs, in some embodiments, the matrix A implemented by the photonic processor 2-103 may not be a square matrix. In such embodiments, the photonic processor 2-103 may have a different number of outputs and inputs.

[00289] It is also noted that, due to the topology of the interconnections of the VBSs within the first and second matrix modules 2-301 and 2-305, it is possible to subdivide the photonic processor 2-103 into non-interacting subsets of rows such that more than one matrix multiplication can be performed at the same time. For example, in the VBS array illustrated in FIG. 2-4, if each VBS 2-401 that couples optical modes 3 and 4 is set such that optical modes 3 and 4 do not couple at all (e.g., as if the VBSs 2-401 with subscript "34" were absent from FIG. 2-4) then the top three optical modes would operate completely independently from the bottom three optical modes. Such a subdivision may be done at a much larger scale with a photonic processor with a larger number of input optical modes. For example, an $n=64$ photonic processor may multiply eight eight-component input vectors by a respective 8 x 8 matrix simultaneously (each of the 8 x 8 matrices being separately programmable and controllable). Moreover, the photonic processor 2-103 need not be subdivided evenly. For example, an $n=64$ photonic processor may subdivide into seven different input vectors with 20, 13, 11, 8, 6, 4, and 2 components, respectively, each multiplied by a respective matrix simultaneously. It should be understood that the above numerical examples are for illustration purposes only and any number of subdivisions is possible.

[00290] Additionally, while the photonic processor 2-103 performs matrix-vector multiplication, where a vector is multiplied by a matrix by passing the optical signals through the array of VBSs, the photonic processor 2-103 may also be used to perform matrix-matrix multiplication. For example, multiple input vectors may be passed through the photonic processor 2-103, one after the other, one input vector at a time, where each input vector represents a column of an input matrix. After optically computing each of the individual matrix-vector multiplications (each multiplication resulting in an output vector that corresponds to a column of an output column of the resulting matrix), the results may be combined digitally to form the output matrix resulting from the matrix-matrix multiplication.

C. Optical Receivers

[00291] The photonic processor 2-103 outputs n optical pulses that are transmitted to the optical receiver 2-105. The optical receiver 2-105 receives the optical pulses and generates an electrical signal based on the received optical signals. In some embodiments, the amplitude and phase of each optical pulse is determined. In some embodiments, this is achieved using homodyne or heterodyne detection schemes. In other embodiments, simple phase-insensitive photodetection may be performed using conventional photodiodes.

[00292] Referring to FIG. 2-9, the optical receiver 2-105 includes a homodyne detector 2-901, a transimpedance amplifier 2-903 and an ADC 2-905, according to some embodiments. Transimpedance amplifier 2-903 may serve as electronic analog scaling unit 1-105 (FIG. 1-12) and ADC 2-905 may be part of ADC unit 1-106. While the components are shown as one element for all optical modes in FIG. 2-9, this is for the sake of simplicity. Each optical mode may have a dedicated homodyne detector 2-901, a dedicated transimpedance amplifier 2-903 and a dedicated ADC 2-905. In some embodiments, a transimpedance amplifier 2-903 may not be used. Instead, any other suitable electronic circuit that converts a current to a voltage may be used.

[00293] Referring to FIG. 2-10, the homodyne detector 2-903 includes a local oscillator (LO) 2-1001, a quadrature controller 2-1003, a beam splitter 2-1005 and two detectors 2-1007 and 2-1009, according to some embodiments. The homodyne detector 2-903 outputs an electrical current that is based on the difference between the current output by the first detector 2-1007 and the second detector 2-1009.

[00294] The local oscillator 2-1001 is combined with the input optical pulse at the beam splitter 2-1005. In some embodiments, a portion of the light source 2-201 is transmitted via an optical waveguide and/or an optical fiber to the homodyne detector 2-901. The light from the light source 2-201 may itself be used as the local oscillator 2-1001 or, in other embodiments, the local oscillator 2-1001 may be a separate light source that uses the light from the light source 2-201 to generate a phase matched optical pulse. In some embodiments, an MZI may replace the beam splitter 2-1005 such that adjustments can be made between the signal and the local oscillator.

[00295] The quadrature controller 2-1003 controls the cross-section angle in phase space in which the measurement is made. In some embodiments, the quadrature controller 2-1003 may be a phase shifter that controls the relative phase between the input optical pulse and the local oscillator. The quadrature controller 2-1003 is shown as a phase shifter in the input optical mode. But in some embodiments, the quadrature controller 2-1003 may be in the local oscillator mode.

[00296] The first detector 2-1007 detects light output by a first output of the beam splitter 2-1005 and the second detector 2-1009 detects light output by a second output of the beam splitter 2-1005. The detectors 2-1007 and 2-1009 may be photodiodes operated with zero bias.

[00297] A subtraction circuit 2-1011 subtracts the electrical current from the first detector 2-1007 from the electrical current from the second detector 2-1009. The resulting current therefore has an amplitude and a sign (plus or minus). The transimpedance amplifier 2-903 converts this

difference in current into a voltage, which may be positive or negative. Finally, an ADC 2-905 converts the analog signal to a digital bit string. This output bit string represents the output vector result of the matrix multiplication and is an electrical, digital version of the optical output representation of the output vector that is output by the photonic processor 2-103. In some embodiments, the output bit string may be sent to the controller 2-107 for additional processing, which may include determining a next input bit string based on one or more output bit strings and/or transmitting the output bit string to an external processor, as described above.

[00298] The inventors have further appreciated that the components of the above-described photonic processing system 2-100 need not be chained together back-to-back such that there is a first matrix module 2-301 connected to a second matrix module 2-303 connected to a third matrix module 2-305. In some embodiments, the photonic processing system 2-103 may include only a single unitary circuit for performing one or more multiplications. The output of the single unitary circuit may be connected directly to the optical receiver 2-105, where the results of the multiplication are determined by detecting the output optical signals. In such embodiments, the single unitary circuit may, for example, implement the first matrix module 2-301. The results detected by the optical receiver 2-105 may then be transmitted digitally to a conventional processor (e.g., processor 2-111) where the diagonal second matrix module 2-303 is performed in the digital domain using a conventional processor (e.g., 2-111). The controller 2-107 may then reprogram the single unitary circuit to perform the third matrix module 2-305, determine an input bit string based on the result of the digital implementation of the second matrix module, and control the optical encoder to transmit optical signals, encoded based on the new input bit string, through the single unitary circuit with the reprogrammed settings. The resulting output optical signals, which are detected by the optical receiver 105, are then used to determine the results of the matrix multiplication.

[00299] The inventors have also appreciated that there can be advantages to chaining multiple photonic processors 2-103 back-to-back, in series. For example, to implement a matrix multiplication $M = M_1 M_2$, where M_1 and M_2 are arbitrary matrices but M_2 changes more frequently than M_1 based on a changing input workload, the first photonic processor can be controlled to implement M_2 and the second photonic processor coupled optically to the first photonic processor can implement M_1 which is kept static. In this way, only the first photonic processing system needs to be frequently updated based on the changing input workload. Not only does such an arrangement speed up the computation, but it also reduces the number of data bits that travel between the controller 2-107 and the photonic processors.

D. Folded Photonic Processing System

[00300] In FIG. 2-1, in such an arrangement, the optical encoder 2-101 and the optical receiver 2-105 are positioned on opposite sides of the photonic processing system 2-100. In applications where feedback from the optical receiver 2-105 is used to determine the input for the optical encoder 2-101 for a future iteration of the process, the data is transferred electronically from the optical receiver 2-105 to the controller 2-107 and then to the optical encoder 2-101. The inventors have appreciated that reducing the distance that these electrical signals need to travel (e.g., by reducing the length of electrical traces and/or wires) results in power savings and lower latency. Moreover, there is no need for the optical encoder 2-101 and optical receiver 2-105 to be placed on opposite ends of the photonic processing system.

[00301] Accordingly, in some embodiments, the optical encoder 2-101 and the optical receiver 2-105 are positioned near one another (e.g., on the same side of the photonics processor 2-103) such that the distance electrical signals have to travel between the optical encoder 2-101 and the optical receiver 2-105 is less than the width of the photonics processor 2-103. This may be accomplished by physically interleaving components of the first matrix module 2-301 and the third matrix module 2-305 such that they are physically in the same portion of the chip. This arrangement is referred to as a "folded" photonic processing system because the light first propagates in a first direction through the first matrix module 2-301 until it reaches a physical portion of the chip that is far from the optical encoder 2-101 and the optical receiver 2-105, then folds over such that the waveguides turn the light to be propagating in a direction opposite to the first direction when implementing the third matrix module 2-305. In some embodiments, the second matrix module 2-303 is physically located adjacent to the fold in the waveguides. Such an arrangement reduces the complexity of the electrical traces connecting the optical encoder 2-101, the optical receiver 2-105, and the controller 2-107 and reduces the total chip area used to implement the photonic processing system 2-100. For example, some embodiments using the folded arrangement only use 65% of the total chip area that would be needed if the back-to-back photonic arrangement of FIG. 2-1 was used. This may reduce the cost and complexity of the photonic processing system.

[00302] The inventors have appreciated that there are not only electrical advantages to a folded arrangement, but also optical advantages. For example, by reducing the distance that the light signal has to travel from the light source to be used as a local oscillator for the homodyne detection, the time-dependent phase fluctuations of the optical signal may be reduced, resulting

in higher quality detection results. In particular, by locating the light source and the homodyne on the same side of the photonics processor, the distance traveled by the light signal used for the local oscillator is no longer dependent on the size of the matrix. For example, in the back-to-back arrangement of FIG. 2-1, the distance traveled by the light signal for the local oscillator scales linearly with the size of the matrix, whereas the distance traveled in the folded arrangement is constant, irrespective of the matrix size.

[00303] FIG. 2-11 is a schematic drawing of a folded photonic processing system 2-1100, according to some embodiments. The folded photonic processing system 2-1100 includes a power tree 2-1101, a plurality of optical encoders 2-1103a – 2-1103d, a plurality of homodyne detectors 2-1105a – 2-1105d, a plurality of selector switches 2-1107a – 2-1107d, a plurality of U-matrix components 2-1109a – 2-1109j, a plurality of diagonal-matrix components 2-1111a – 2-1111d, and a plurality of V-matrix components 2-1113a – 2-1113j. For the sake of clarity, not all components of the folded photonic processing system are shown in the figure. It should be understood that the folded photonic processing system 2-1100 may include similar components as the back-to-back photonic processing system 2-100.

[00304] The power tree 2-1101 is similar to the power tree 2-203 of FIG. 2 and is configured to deliver light from a light source (not shown) to the optical encoders 2-1103. However, a difference in the power tree 2-1101 and the power tree 2-203 is that the power tree delivers optical signals to the homodyne detectors 2-1105a directly. In FIG. 2, the light source 201 delivers a local oscillator signal to the homodyne detectors on the other side of the photonic processor by tapping off a portion of the optical signal from the light source and guiding the optical signal using a waveguide. In FIG. 2-11, the power tree 2-1101 includes a number of outputs that is equal to twice the number of spatial modes. For example, FIG. 2-11 illustrates only four spatial modes of a photonic processor, which results in eight output modes from the power tree 2-1101 – one output directing light to each optical encoder 2-1103 and one output directing light to each homodyne detector 2-1105. The power tree may be implemented, for example, using cascading beam splitters or a multimode interferometer (MMI).

[00305] The optical encoders 2-1103 are similar to the power tree optical encoder 2-101 of FIG. 1 and are configured to encode information into the amplitude and/or phase of the optical signals received from the power tree 2-1101. This may be achieved, for example as described in connection with the optical encoder 2-101 of FIG. 2.

[00306] The homodyne detectors 2-1105 are located between the power tree 2-1101 and the U-matrix components 2-1109. In some embodiments, the homodyne detectors 2-1105 are

physically positioned in a column with the optical encoder 2-1103. In some embodiments, the optical encoders 2-1103 and the homodyne detectors 2-1105 may be interleaved in a single column. In this way, the optical encoders 2-1103 and the homodyne detectors 2-1105 are in close proximity to one another, reducing the distance of electrical traces (not shown) used to connect the optical encoders 2-1103 and the homodyne detectors 2-1105 and a controller (not shown) which may be physically located adjacent to the column of the optical encoders 2-1103 and the homodyne detectors 2-1105.

[00307] Each of the optical encoders 2-1103 is associated with a respective homodyne detector 2-1105. Both the optical encoders 2-1103 and the homodyne detectors 2-1105 receive optical signals from the power tree 2-1101. The optical encoders 2-1103 use the optical signals to encode an input vector, as described above. The homodyne detectors 2-1105 use the received optical signals received from the power tree as the local oscillator, as described above.

[00308] Each pair of the optical encoders 2-1103 and the homodyne detectors 2-1105 is associated with and connected to a selector switch 2-1107 by a waveguide. The selector switches 2-1107a – 2-1107d may be implemented using, for example, a conventional 2x2 optical switch. In some embodiments, the 2×2 optical switch is a MZI with an internal phase shifter to control the MZI's behavior from a crossing to a bar. The switch 2-1107 is connected to a controller (not shown) to control whether an optical signal received from the optical encoder 2-1103 will be guided towards the U-matrix components 2-1109 or the V-matrix components 2-1113. The optical switch is also controlled to guide light received from the U-matrix components 2-1109 and/or the V-matrix components 2-1113 toward the homodyne detectors 2-1105 for detection.

[00309] The techniques for implementing matrix multiplication is similar in the photonic folded photonic processing system 2-1100 as was described above in connection with the back-to-back system, described in FIG. 2-3. A difference between the two systems is in the physical placement of the matrix components and the implementation of a fold 2-1120, where the optical signals change from propagating approximately left to right in FIG. 2-11 to propagating approximately right to left. In FIG. 2-11, the connections between components may represent waveguides. The solid-lined connections represent portions of waveguide where the optical signals are propagating from left to right, in some embodiments, and the dashed-lined connections represent portions of waveguide where the optical signals are propagating from right to left, in some embodiments. In particular, given this nomenclature, the embodiment illustrated in FIG. 2-11 is an embodiment where the selector switches 2-1107 guide the optical signals to the U-matrix components 2-1109 first. In other embodiments, the selector switches 2-

1107 may guide the optical signals to the V-matrix components 2-1113 first, in which case the dashed lines would represent portions of waveguide where the optical signals are propagating from left to right, and the solid-lined connections would represent portions of waveguide where the optical signals are propagating from right to left.

[00310] The U-matrix of the SVD of a matrix A is implemented in photonic processing system 2-1100 using U-matrix components 2-1109 that are interleaved with the V-matrix components 2-1113. Thus, unlike the embodiment of the back-to-back arrangement illustrated in FIG. 2-3, all of the U-matrix components 2-1109 and the V-matrix components 2-1113 are not physically located in a respective self-contained array within a single physical area. Accordingly, in some embodiments, the photonic processing system 2-1100 includes a plurality of columns of matrix components and at least one of the columns contains both U-matrix components 2-1109 and V-matrix components 2-1113. In some embodiments, the first column may only have U-matrix components 2-1109, as illustrated in FIG. 2-11. U-matrix components 2-1109 are implemented similarly to the first matrix module 2-301 of FIG. 3.

[00311] Due to the interleaving structure of the U-matrix components 2-1109 and the V-matrix components 2-1113, the folded photonic processing system 2-1100 includes waveguide crossovers 2-1110 at various locations between the columns of matrix elements. In some embodiments, the waveguide crossovers can be constructed using adiabatic evanescent elevators between two or more layers in an integrated photonics chip. In other embodiments, the U-matrix and the V-matrix may be positioned on different layers of the same chip and the waveguide crossovers are not used.

[00312] After optical signals propagate through all of the U-matrix components 2-1109, the optical signals propagate to the diagonal-matrix components 2-1111, which are implemented similarly to the second matrix module 2-303 of FIG. 2-3.

[00313] After optical signals propagate through all of the diagonal-matrix components 2-1111, the optical signals propagate to the V-matrix components 2-1113, which are implemented similarly to the third matrix module 2-305 of FIG. 2-3. The V-matrix of the SVD of a matrix A is implemented in photonic processing system 2-1100 using V-matrix components 2-1113 that are interleaved with the U-matrix components 2-1109. Thus, all of the V-matrix components 2-1113 are not physically located in a single self-contained array.

[00314] After the optical signals propagate through all of the V-matrix components 2-1113, the optical signals return to the selector switch 2-1107, which guides the optical signals to the homodyne detectors 2-1105 for detection.

[00315] The inventors have further appreciated that by including selector switches after the optical encoders and before the matrix components, the folded photonic processing system 2-1100 allows efficient bi-directionality of the circuit. Thus, in some embodiments, a controller, such as the controller 2-107 described in connection with FIG. 2-1, may control whether the optical signals are multiplied by the U matrix first or the V^T matrix first. For an array of VBSs set to implement a unitary matrix U when propagating the optical signals from left to right, propagating the optical signals from right to left implements a multiplication by a unitary matrix U^T. Thus, the same settings for an array of VBSs can implement both U and U^T depending which way the optical signals are propagated through the array, which may be controlled using the selector switch in 2-1107. In some applications, such as back-propagation used to train a machine learning algorithm, it may be desirable to run optical signals through one or more matrices backwards. In other applications, the bi-directionality can be used to compute the operation of an inverted matrix on an input vector. For example, for an invertible $n \times n$ matrix A an SVD results in $A = V^T \Sigma U$. The inverse of this matrix is $A^{-1} = U^T \Sigma^{-1} V$ where Σ^{-1} is the inverse of a diagonal matrix which can be computed efficiently by inverting each diagonal element. To multiply a vector by the matrix A, the switches are configured to direct the optical signals through the matrix U, then Σ, then V^T in a first direction. To multiply a vector by the inverse A^{-1}, the singular values are first set to program the implementation of the Σ^{-1} matrix. This constitutes changing the settings of only one column of VBSs instead of all $2n + 1$ columns of the photonic processor, which is the case for a single-directional photonic processing system such as the one illustrated in FIG. 2-3. The optical signals representing the input vector are then propagated through the matrix V^T, then Σ^{-1}, and then U in a second direction that is opposite the first direction. Using the selector switches 2-1107, the folded photonic processing system 2-1100 may be easily changed from implementing the U matrix (or its transpose) first and implementing the V^T matrix (or its transpose) first.

VI. Conclusion

[00316] Having thus described several aspects and embodiments of the technology of this application, it is to be appreciated that various alterations, modifications, and improvements will readily occur to those of ordinary skill in the art. Such alterations, modifications, and improvements are intended to be within the spirit and scope of the technology described in the application. It is, therefore, to be understood that the foregoing embodiments are presented by way of example only and that, within the scope of the appended claims and equivalents thereto,

inventive embodiments may be practiced otherwise than as specifically described. In addition, any combination of two or more features, systems, articles, materials, and/or methods described herein, if such features, systems, articles, materials, and/or methods are not mutually inconsistent, is included within the scope of the present disclosure.

[00317] Also, as described, some aspects may be embodied as one or more methods. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[00318] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[00319] The indefinite articles "a" and "an," as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean "at least one."

[00320] The phrase "and/or," as used herein in the specification and in the claims, should be understood to mean "either or both" of the elements so conjoined, *i.e.*, elements that are conjunctively present in some cases and disjunctively present in other cases.

[00321] The term "circuitry," as used herein in the specification and in the claims, should be understood to mean any type of circuit, whether considered alone or in combination with other circuits, including for example analog electrical circuits, digital electronic circuits and optical circuits. Optical circuits include, among others, solid state optical components and combinations thereof and free space optical components and combinations thereof.

[00322] As used herein in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified.

[00323] The terms "approximately" and "about" may be used to mean within ±20% of a target value in some embodiments, within ±10% of a target value in some embodiments, within ±5%

of a target value in some embodiments, and yet within ±2% of a target value in some embodiments. The terms “approximately” and “about” may include the target value.

CLAIMS

What is claimed is:

1. A hybrid analog-digital processor comprising:

circuitry comprising an analog processor, wherein the circuitry is configured to perform a mathematical operation using a plurality of passes, wherein for each of the plurality of passes, the circuitry is configured to:

determine one or more scaling factors for the pass based on a set of parameters representing a portion of a matrix;

scale at least some parameters of the set of parameters based on the one or more scaling factors to produce a scaled set of parameters;

program the analog processor based on the scaled set of parameters;

generate a plurality of input analog signals based on an input data set;

generate a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters;

generate a partial output data set based on the plurality of output analog signals; and

scale the partial output data set based on the one or more scaling factors to produce a scaled partial output data set,

wherein the circuitry is further configured to generate an accumulated output data set by accumulating the scaled partial output data sets generated by at least two of the plurality of passes, wherein the accumulated output data set represents a result of the mathematical operation.

2. The hybrid analog-digital processor of claim 1, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the scaled set of parameters.

3. The hybrid analog-digital processor of claim 1, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters

comprises performing a convolution based on the plurality of input analog signals and the scaled set of parameters.

4. The hybrid analog-digital processor of claim 1, wherein programming the analog processor based on the scaled set of parameters comprises setting respective gains or attenuations for a plurality of analog amplifiers or attenuators of the analog processor based on the scaled set of parameters.

5. The hybrid analog-digital processor of claim 1, wherein the analog processor comprises a photonic processor comprising a plurality of programmable photonic devices, and wherein programming the analog processor based on the scaled set of parameters comprises setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters.

6. The hybrid analog-digital processor of claim 5, wherein the programmable photonic devices comprise Mach-Zehnder interferometers, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises:

setting respective optical characteristics for the plurality of Mach-Zehnder interferometers based on the scaled set of parameters.

7. The hybrid analog-digital processor of claim 5, wherein the programmable photonic devices comprise optical resonators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises:

setting respective optical characteristics for the plurality of optical resonators based on the scaled set of parameters.

8. The hybrid analog-digital processor of claim 1, wherein the circuitry comprises a plurality of digital-to-analog converters (DACs), and the plurality of DACs are configured to generate the plurality of input analog signals based on the input data set.

9. The hybrid analog-digital processor of claim 1, wherein the circuitry comprises a plurality of analog-to-digital converters (ADCs), and the plurality of ADCs are configured to generate the partial output data set based on the plurality of output analog signals.

10. The hybrid analog-digital processor of claim 9, wherein the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

11. The hybrid analog-digital processor of claim 1, wherein accumulating the scaled partial output data sets generated by at least two of the plurality of passes comprises:

for at least some of the passes, summing the scaled partial output data set generated by the pass with the scaled partial output data set generated by a previous pass.

12. The hybrid analog-digital processor of claim 1, wherein programming the analog processor comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

13. The hybrid analog-digital processor of claim 12, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

14. The hybrid analog-digital processor of claim 12, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

15. The hybrid analog-digital processor of claim 1, wherein:

scaling at least some of the set of parameters based on the one or more scaling factors comprises dividing at least some of the set of parameters by the one or more scaling factors, and

scaling the partial output data set based on the one or more scaling factors comprises multiplying the partial output data set by the one or more scaling factors.

16. The hybrid analog-digital processor of claim 1, wherein, for each of the plurality of passes, the circuitry is further configured to determine the one or more scaling factors based on a tile of the matrix that is associated with the pass.

17. The hybrid analog-digital processor of claim 1, wherein, for each of the plurality of passes, the circuitry is further configured to determine the one or more scaling factors based on one or more rows of the matrix.

18. The hybrid analog-digital processor of claim 1, wherein, for each of the plurality of passes, the circuitry is further configured to determine the one or more scaling factors based on one or more singular values of the matrix.

19. The hybrid analog-digital processor of claim 1, wherein:

the one or more scaling factors comprise a first scaling factor and a second scaling factor, and

scaling at least some parameters of the set of parameters based on the one or more scaling factors comprises:

scaling a first subset of the set of parameters based on the first scaling factor, and

scaling a second subset of the set of parameters based on the second scaling factor.

20. A method for performing a mathematical operation using a plurality of passes through an analog processor, the method comprising:

for each of the plurality of passes:

determining one or more scaling factors for the pass based on a set of parameters representing a portion of a matrix;

scaling at least some parameters of the set of parameters based on the one or more scaling factors to produce a scaled set of parameters;

programming the analog processor based on the scaled set of parameters;

generating a plurality of input analog signals based on an input data set;

generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters;

generating a partial output data set based on the plurality of output analog signals; and

scaling the partial output data set based on the one or more scaling factors to produce a scaled partial output data set; and

generating an accumulated output data set by accumulating the scaled partial output data sets generated by at least two of the plurality of passes, wherein the accumulated output data set represents a result of the mathematical operation.

21. The method of claim 20, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the scaled set of parameters.

22. The method of claim 20, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the scaled set of parameters comprises performing a convolution based on the plurality of input analog signals and the scaled set of parameters.

23. The method of claim 20, wherein programming the analog processor based on the scaled set of parameters comprises setting respective gains or attenuations for a plurality of analog amplifiers or attenuators of the analog processor based on the scaled set of parameters.

24. The method of claim 20, wherein the analog processor comprises a photonic processor comprising a plurality of programmable photonic devices, and wherein programming the analog processor based on the scaled set of parameters comprises setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters.

25. The method of claim 24, wherein the programmable photonic devices comprise Mach-Zehnder interferometers, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises:

setting respective optical characteristics for the plurality of Mach-Zehnder interferometers based on the scaled set of parameters.

26. The method of claim 24, wherein the programmable photonic devices comprise optical resonators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises:

setting respective optical characteristics for the plurality of optical resonators based on the scaled set of parameters.

27. The method of claim 20, wherein generating the plurality of input analog signals based on the input data set comprises providing the input data set as input to a plurality of digital-to-analog converters (DACs).

28. The method of claim 20, wherein generating the partial output data set based on the plurality of output analog signals the circuitry comprises providing the plurality of output analog signals as inputs to a plurality of analog-to-digital converters (ADCs).

29. The method of claim 28, wherein the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

30. The method of claim 20, wherein accumulating the scaled partial output data sets generated by at least two of the plurality of passes comprises:

for at least some of the passes, summing the scaled partial output data set generated by the pass with the scaled partial output data set generated by a previous pass.

31. The method of claim 20, wherein programming the analog processor comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

32. method of claim 31, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

33. The method of claim 31, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

34. The method of claim 20, wherein:

scaling at least some of the set of parameters based on the one or more scaling factors comprises dividing at least some of the set of parameters by the one or more scaling factors, and

scaling the partial output data set based on the one or more scaling factors comprises multiplying the partial output data set by the one or more scaling factors.

35. The method of claim 20, wherein determining the one or more scaling factors comprises determining the one or more scaling factors based on a tile of the matrix that is associated with the pass.

36. The method of claim 20, wherein determining the one or more scaling factors comprises determining the one or more scaling factors based on one or more rows of the matrix.

37. The method of claim 20, wherein determining the one or more scaling factors comprises determining the one or more scaling factor based on one or more singular values of the matrix.

38. The method of claim 20, wherein:

the one or more scaling factors comprise a first scaling factor and a second scaling factor, and

scaling at least some parameters of the set of parameters based on the one or more scaling factors comprises:

scaling a first subset of the set of parameters based on the first scaling factor, and

scaling a second subset of the set of parameters based on the second scaling factor.

39. A hybrid analog-digital processor comprising:

a digital-to-analog converter (DAC) unit comprising a plurality of DACs;

an analog processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit;

an analog-to-digital (ADC) unit comprising a plurality of ADCs and coupled to outputs of the analog processor;

a digital scaling unit coupled to outputs of the ADC unit; and

a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder.

40. The hybrid analog-digital processor of claim 39, further comprising an amplification or attenuation unit coupled between the DAC unit and the ADC unit.

41. The hybrid analog-digital processor of claim 39, wherein the analog processor comprises a photonic processor comprising a plurality of programmable photonic devices.

42. The hybrid analog-digital processor of claim 41, wherein the programmable photonic devices comprise Mach-Zehnder interferometers.

43. The hybrid analog-digital processor of claim 41, wherein the programmable photonic devices comprise optical resonators.

44. The hybrid analog-digital processor of claim 39, wherein the analog processor is programmed with a plurality of matrices that, collectively, represent an arbitrary matrix.

45. The hybrid analog-digital processor of claim 44, wherein the plurality of matrices collectively represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

46. The hybrid analog-digital processor of claim 44, wherein the plurality of matrices collectively represent the arbitrary matrix based on a dilation of the arbitrary matrix.

47. A hybrid analog-digital processor configured to perform a mathematical operation, comprising:

circuitry comprising an analog processor and an analog scaling unit, wherein the circuitry is configured to:

generate a plurality of input analog signals based on an input data set;

set a gain of the analog scaling unit based on one or more scaling factors;

program the analog processor with a set of parameters representing a matrix;

generate a plurality of output analog signals based on the plurality of input analog signals and the set of parameters;

generate a plurality of amplified or attenuated output analog signals by amplifying or attenuating, using the analog scaling unit, the plurality of input analog signals and/or the plurality of output analog signals; and

generate an output data set based on the plurality of amplified or attenuated output analog signals.

48. The hybrid analog-digital processor of claim 47, wherein the analog scaling unit comprises an analog amplifier or attenuator.

49. The hybrid analog-digital processor of claim 47, wherein the hybrid analog-digital processor is further configured to perform a multi-pass computation based on the mathematical operation, wherein the circuitry is further configured to:

set the gain of the analog scaling unit to a first value during a first pass of the multi-pass computation; and

set the gain of the analog scaling unit to a second value, different from the first value, during a second pass of the multi-pass computation.

50. The hybrid analog-digital processor of claim 47, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the set of parameters.

51. The hybrid analog-digital processor of claim 47, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a convolution based on the plurality of input analog signals and the set of parameters.

52. The hybrid analog-digital processor of claim 47, wherein the circuitry comprises a plurality of analog-to-digital converters (ADCs), and the plurality of ADCs are configured to generate the output data set based on the plurality of output analog signals.

53. The hybrid analog-digital processor of claim 52, wherein the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

54. The hybrid analog-digital processor of claim 47, wherein programming the analog processor comprises:

programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

55. The hybrid analog-digital processor of claim 54, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

56. The hybrid analog-digital processor of claim 54, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

57. The hybrid analog-digital processor of claim 47, wherein the circuitry is further configured to determine the one or more scaling factors based on the set of parameters and the input data set.

58. The hybrid analog-digital processor of claim 57, wherein determining the one or more scaling factors comprises determining the one or more scaling factors based on statistical bounds on the set of parameters and statistical bounds on the input data set.

59. A method for performing a mathematical operation, the method comprising:

generating a plurality of input analog signals based on an input data set;

setting a gain of an analog scaling unit based on one or more scaling factors;

programming an analog processor with a set of parameters representing a matrix;

generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters;

generating a plurality of amplified or attenuated output analog signals by amplifying or attenuating, using the analog scaling unit, the plurality of input analog signals and/or the plurality of output analog signals; and

generating an output data set based on the plurality of amplified or attenuated output analog signals.

60. The method of claim 59, wherein the analog scaling unit comprises an analog amplifier or attenuator.

61. The method of claim 59, wherein the hybrid analog-digital processor is further configured to perform a multi-pass computation based on the mathematical operation, wherein the circuitry is further configured to:

set the gain of the analog scaling unit to a first value during a first pass of the multi-pass computation; and

set the gain of the analog scaling unit to a second value, different from the first value, during a second pass of the multi-pass computation.

62. The method of claim 59, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the set of parameters.

63. The method of claim 59, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a convolution based on the plurality of input analog signals and the set of parameters.

64. The method of claim 59, wherein the circuitry comprises a plurality of analog-to-digital converters (ADCs), and the plurality of ADCs are configured to generate the output data set based on the plurality of output analog signals.

65. The method of claim 64, wherein the plurality of ADCs comprise n-bit ADCs, with n equal to or less than 12.

66. The method of claim 59, wherein programming the analog processor comprises:

programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

67. The method of claim 66, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

68. The method of claim 66, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

69. The method of claim 59, further comprising determining the one or more scaling factors based on the set of parameters and the input data set.

70. The method of claim 69, wherein determining the one or more scaling factors comprises determining the one or more scaling factors based on statistical bounds on the set of parameters and statistical bounds on the input data set.

71. A hybrid analog-digital processor comprising:

circuitry comprising a photonic processor, wherein the circuitry is configured to perform a mathematical operation using a plurality of passes, wherein for each of the plurality of passes, the circuitry is configured to:

determine one or more scaling factors for the pass based on a set of parameters representing a portion of a matrix;

scale at least some parameters of the set of parameters based on the one or more scaling factors to produce a scaled set of parameters;

program the photonic processor based on the scaled set of parameters;

generate a plurality of input optical signals based on an input data set;

generate a plurality of output optical signals based on the plurality of input optical signals and the scaled set of parameters;

generate a partial output data set based on the plurality of output optical signals; and

scale the partial output data set based on the one or more scaling factors to produce a scaled partial output data set,

wherein the circuitry is further configured to generate an accumulated output data set by accumulating the scaled partial output data sets generated by at least two of the plurality of passes, wherein the accumulated output data set represents a result of the mathematical operation.

72. The hybrid analog-digital processor of claim 71, wherein the photonic processor comprises a plurality of programmable photonic devices, and wherein programming the analog processor based on the scaled set of parameters comprises setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters.

73. The hybrid analog-digital processor of claim 72, wherein the programmable photonic devices comprise optical resonators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises:

setting respective optical characteristics for the plurality of optical resonators based on the scaled set of parameters.

74. The hybrid analog-digital processor of claim 73, wherein the optical resonators comprise optical ring or disk resonators, and wherein setting the respective optical characteristics for the plurality of optical resonators comprises:

setting respective resonant frequencies for the plurality of optical ring or disk resonators based on the scaled set of parameters.

75. The hybrid analog-digital processor of claim 72, wherein the programmable photonic devices comprise Franz-Keldysh modulators, and wherein setting respective characteristics for the plurality of programmable photonic devices based on the scaled set of parameters comprises:

setting respective absorptions for the plurality of Franz-Keldysh modulators based on the scaled set of parameters.

76. The hybrid analog-digital processor of claim 71, wherein the photonic processor comprises:

a first array of interconnected variable beam splitters (VBSs) comprising a first plurality of optical inputs and a first plurality of optical outputs;

a second array of interconnected VBSs comprising a second plurality of optical inputs and a second plurality of optical outputs; and

a plurality of controllable optical elements, each of the plurality of these controllable optical elements coupling a single one of the first plurality of optical outputs of the first array to a respective single one of the second plurality of optical inputs of the second array.

77. The hybrid analog-digital processor of claim 76, programming the photonic processor based on the scaled set of parameters comprises programing the plurality of controllable optical elements based on the scaled set of parameters.

78. The hybrid analog-digital processor of claim 76, wherein each of the VBSs of the first and second array comprises a Mach-Zehnder interferometer comprising:

a first beam splitter;

a second beam splitter; and

at least one phase modulator configured to modulate a phase of light of an optical mode that couples the first beam splitter and the second beam splitter.

79. The hybrid analog-digital processor of claim 76, wherein each of the controllable optical elements comprises a variable attenuator and at least one phase modulator.

80. The hybrid analog-digital processor of claim 71, wherein generating a plurality of output optical signals based on the plurality of input optical signals and the scaled set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input optical signals and the scaled set of parameters.

81. The hybrid analog-digital processor of claim 71, wherein programming the photonic processor comprises:

programming, based on the scaled set of parameters, the photonic processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

82. A hybrid analog-digital processor comprising:

a digital-to-analog converter (DAC) unit comprising a plurality of DACs;

a photonic processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit;

an analog-to-digital (ADC) unit comprising a plurality of ADCs and coupled to outputs of the photonic processor;

a digital scaling unit coupled to outputs of the ADC unit; and

a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder.

83. The hybrid analog-digital processor of claim 82, further comprising an optical amplification or attenuation unit coupled between the DAC unit and the ADC unit.

84. The hybrid analog-digital processor of claim 82, wherein the photonic processor comprises a plurality of optical resonators.

85. The hybrid analog-digital processor of claim 82, wherein the photonic processor comprises:

a first array of interconnected variable beam splitters (VBSs) comprising a first plurality of optical inputs and a first plurality of optical outputs;

a second array of interconnected VBSs comprising a second plurality of optical inputs and a second plurality of optical outputs; and

a plurality of controllable optical elements, each of the plurality of these controllable optical elements coupling a single one of the first plurality of optical outputs of the first array to a respective single one of the second plurality of optical inputs of the second array.

86. The hybrid analog-digital processor of claim 82, wherein the photonic processor is programmed with a plurality of matrices that, collectively, represent an arbitrary matrix.

87. A hybrid analog-digital processor configured to perform a mathematical operation, comprising:

circuitry comprising a photonic processor and at least one amplifier, wherein the circuitry is configured to:

generate a plurality of input analog signals based on an input data set;

set respective a gain of the at least one amplifier based on one or more scaling factors;

program the photonic processor with a set of parameters representing a matrix;

generate a plurality of output optical signals based on the plurality of input optical signals and the set of parameters;

generate a plurality of output analog signals based on the plurality of output optical signals;

generate a plurality of amplified output signals by amplifying, using the at least one amplifier, at least one among the plurality of input optical signals, the plurality of output optical signals and the plurality of output analog signals; and

generate an output data set based on the plurality of amplified output signals.

88. The hybrid analog-digital processor of claim 87, wherein the at least one amplifier comprises an optical amplifier and an electronic amplifier, wherein amplifying, using the at least one amplifier, at least one among the plurality of input optical signals, the plurality of output optical signals and the plurality of output analog signals comprises:

amplifying the plurality of input optical signals with the optical amplifier; and

amplifying the plurality of output analog signals with the electronic amplifier.

89. The hybrid analog-digital processor of claim 87, wherein the at least one amplifier comprises a laser, and wherein setting the gain of the amplifier comprises setting a gain of the laser.

90. The hybrid analog-digital processor of claim 87, wherein the hybrid analog-digital processor is further configured to perform a multi-pass computation based on the mathematical operation, wherein the circuitry is further configured to:

set the gain of the at least one amplifier to a first value during a first pass of the multi-pass computation; and

set the gain of the at least one amplifier to a second value, different from the first value, during a second pass of the multi-pass computation.

91. The hybrid analog-digital processor of claim 87, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a matrix-matrix multiplication based on the plurality of input analog signals and the set of parameters.

92. The hybrid analog-digital processor of claim 87, wherein generating a plurality of output analog signals based on the plurality of input analog signals and the set of parameters comprises performing a convolution based on the plurality of input analog signals and the set of parameters.

93. The hybrid analog-digital processor of claim 87, wherein the photonic processor comprises a plurality of optical resonators.

94. The hybrid analog-digital processor of claim 87, wherein the photonic processor comprises:

a first array of interconnected variable beam splitters (VBSs) comprising a first plurality of optical inputs and a first plurality of optical outputs;

a second array of interconnected VBSs comprising a second plurality of optical inputs and a second plurality of optical outputs; and

a plurality of controllable optical elements, each of the plurality of these controllable optical elements coupling a single one of the first plurality of optical outputs of the first array to a respective single one of the second plurality of optical inputs of the second array.

95. The hybrid analog-digital processor of claim 87, wherein programming the analog processor comprises:

programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent an arbitrary matrix.

96. The hybrid analog-digital processor of claim 95, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a singular value decomposition (SVD) of the arbitrary matrix.

97. The hybrid analog-digital processor of claim 96, wherein programming the analog processor with a plurality of matrices comprises:

programming, based on the scaled set of parameters, the analog processor with a plurality of matrices that, collectively, represent the arbitrary matrix based on a dilation of the arbitrary matrix.

98. The hybrid analog-digital processor of claim 87, wherein the circuitry is further configured to determine the one or more scaling factors based on the set of parameters and the input data set.

99. The hybrid analog-digital processor of claim 98, wherein determining the one or more scaling factors comprises determining the one or more scaling factors based on statistical bounds on the set of parameters and statistical bounds on the input data set.

100. A method of manufacturing a hybrid analog-digital processor comprising:

forming a digital-to-analog converter (DAC) unit comprising a plurality of DACs;

forming a photonic processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit;

forming an analog-to-digital (ADC) unit comprising a plurality of ADCs and coupled to outputs of the photonic processor;

forming a digital scaling unit coupled to outputs of the ADC unit; and

forming a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder.

101. The method of claim 100, wherein forming the photonic processor and forming the digital accumulator comprises forming the photonic processor and the digital accumulator on a common chip.

102. The method of claim 100, wherein forming the photonic processor and forming the digital accumulator comprises:

forming the photonic processor on a first chip;

forming the digital accumulator on a second chip; and

bonding the first chip to the second chip.

103. The method of claim 100, wherein forming the ADC unit comprises forming a plurality of n-bit ADCs, where n is less than or equal to 12.

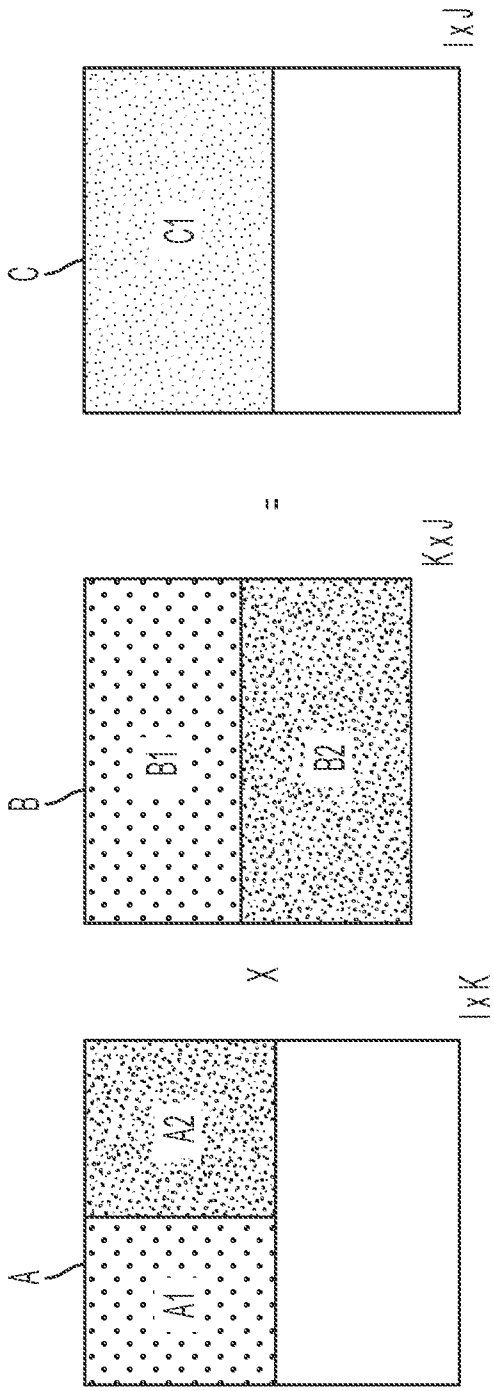


FIG. 1-1A

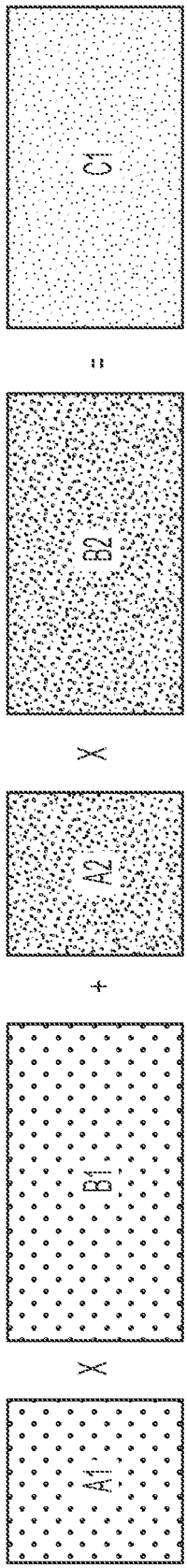


FIG. 1-1B

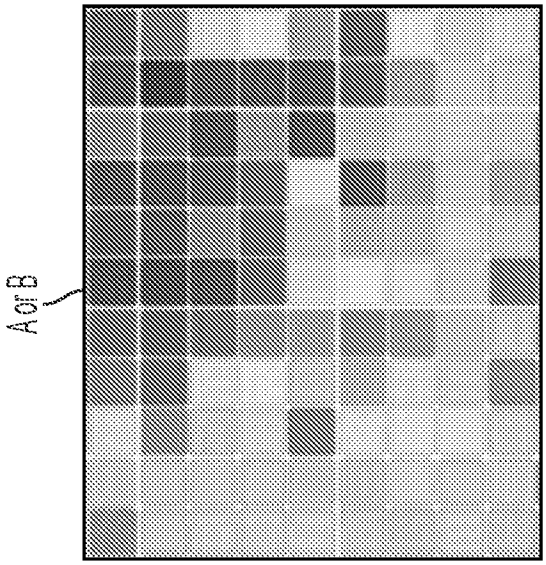


FIG. 1-1C

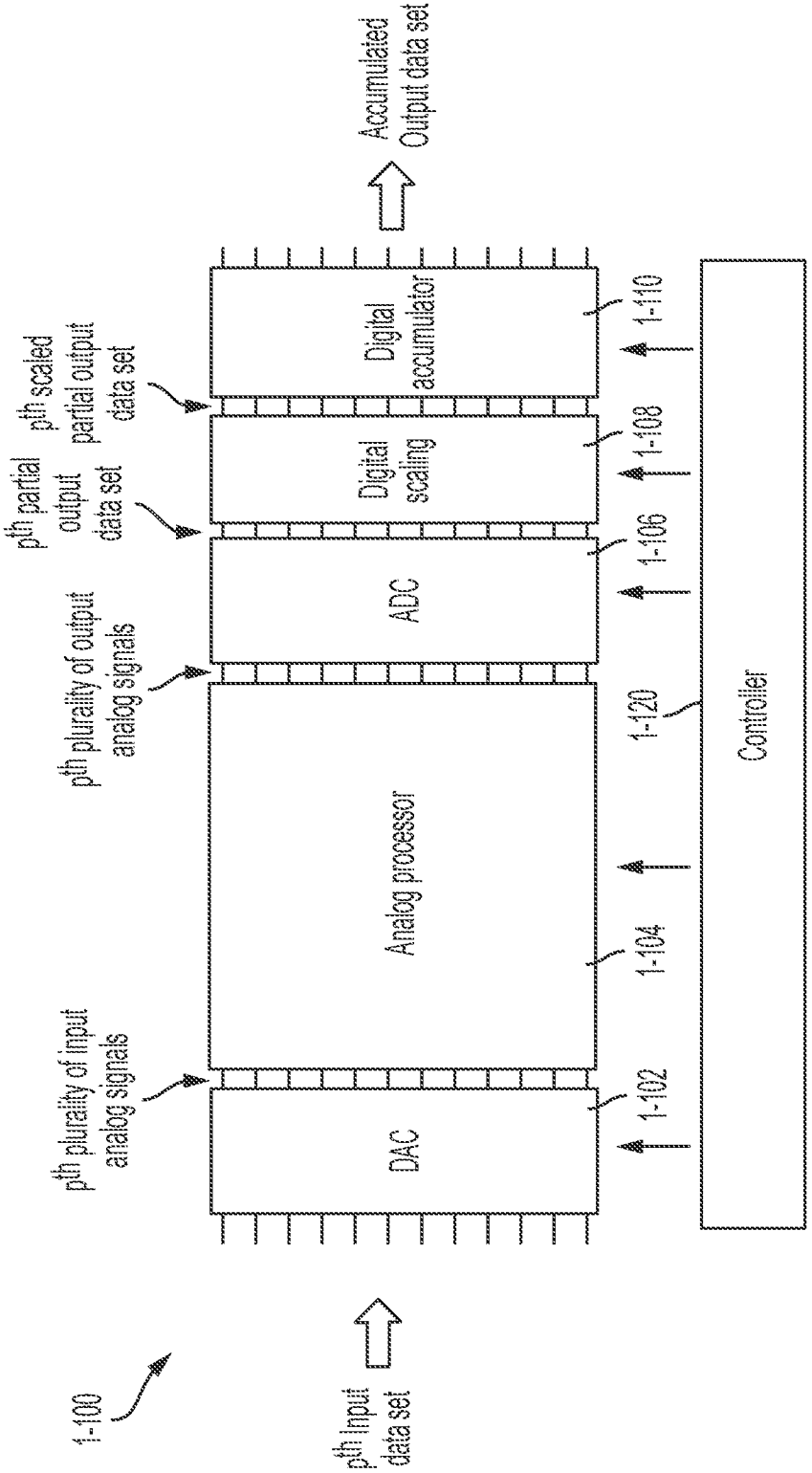

1-100
pth Input data set
pth plurality of input analog signals
DAC
1-102
Analog processor
1-104
pth plurality of output analog signals
1-120
ADC
1-106
pth partial output data set
Digital scaling
1-108
pth scaled partial output data set
Digital accumulator
1-110
Accumulated Output data set
Controller

FIG. 1-2A

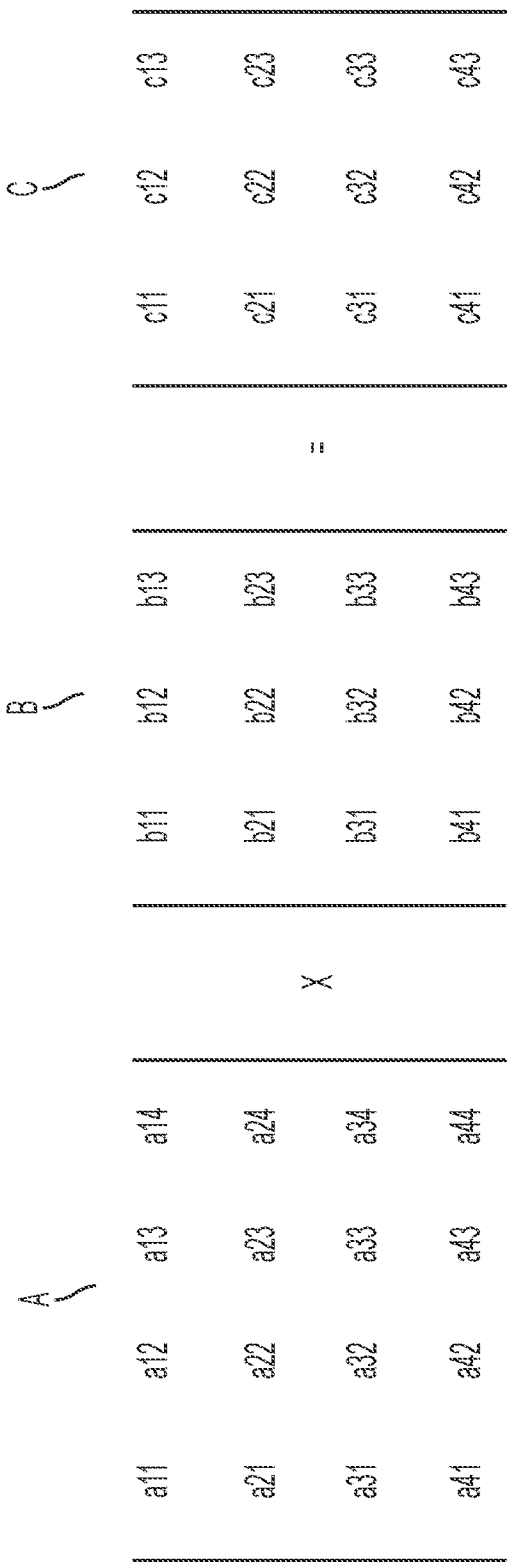

A
a11 a12 a13 a14
a21 a22 a23 a24
a31 a32 a33 a34
a41 a42 a43 a44
X
B
b11 b12 b13
b21 b22 b23
b31 b32 b33
b41 b42 b43
=
C
c11 c12 c13
c21 c22 c23
c31 c32 c33
c41 c42 c43

FIG. 1-2B

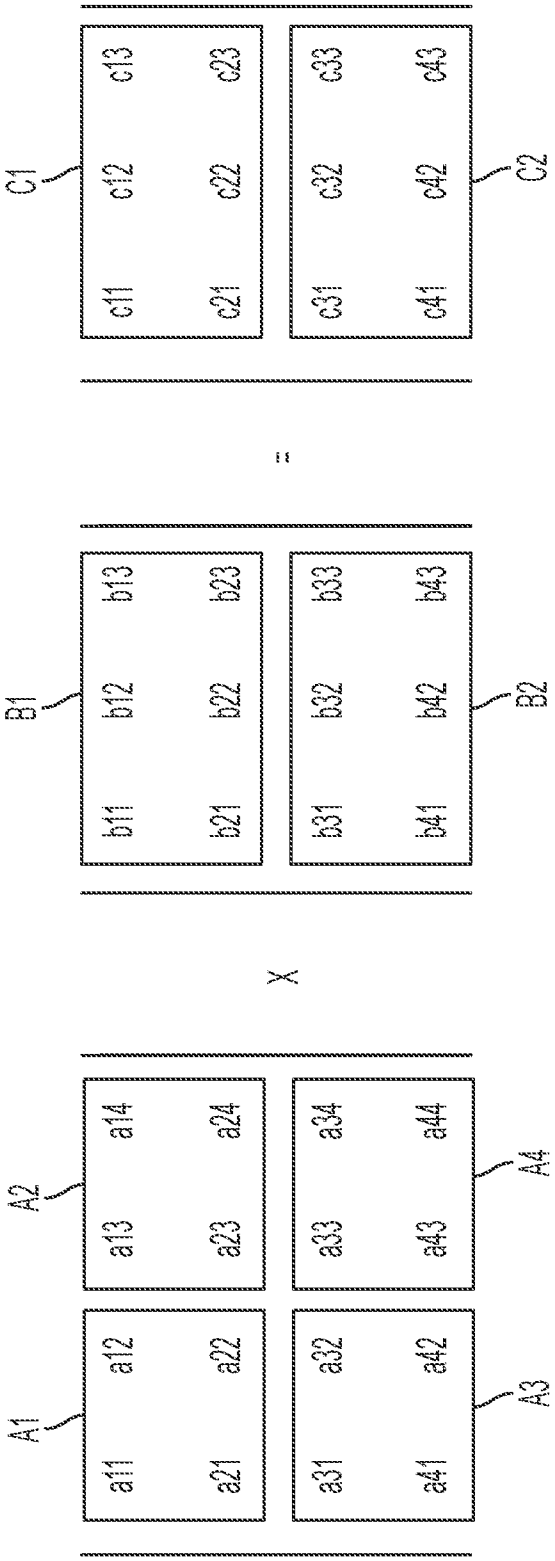

A1
a11 a12
a21 a22
A2
a13 a14
a23 a24
A3
a31 a32
a41 a42
A4
a33 a34
a43 a44
X
B1
b11 b12 b13
b21 b22 b23
B2
b31 b32 b33
b41 b42 b43
=
C1
c11 c12 c13
c21 c22 c23
C2
c31 c32 c33
c41 c42 c43

FIG. 1-2C

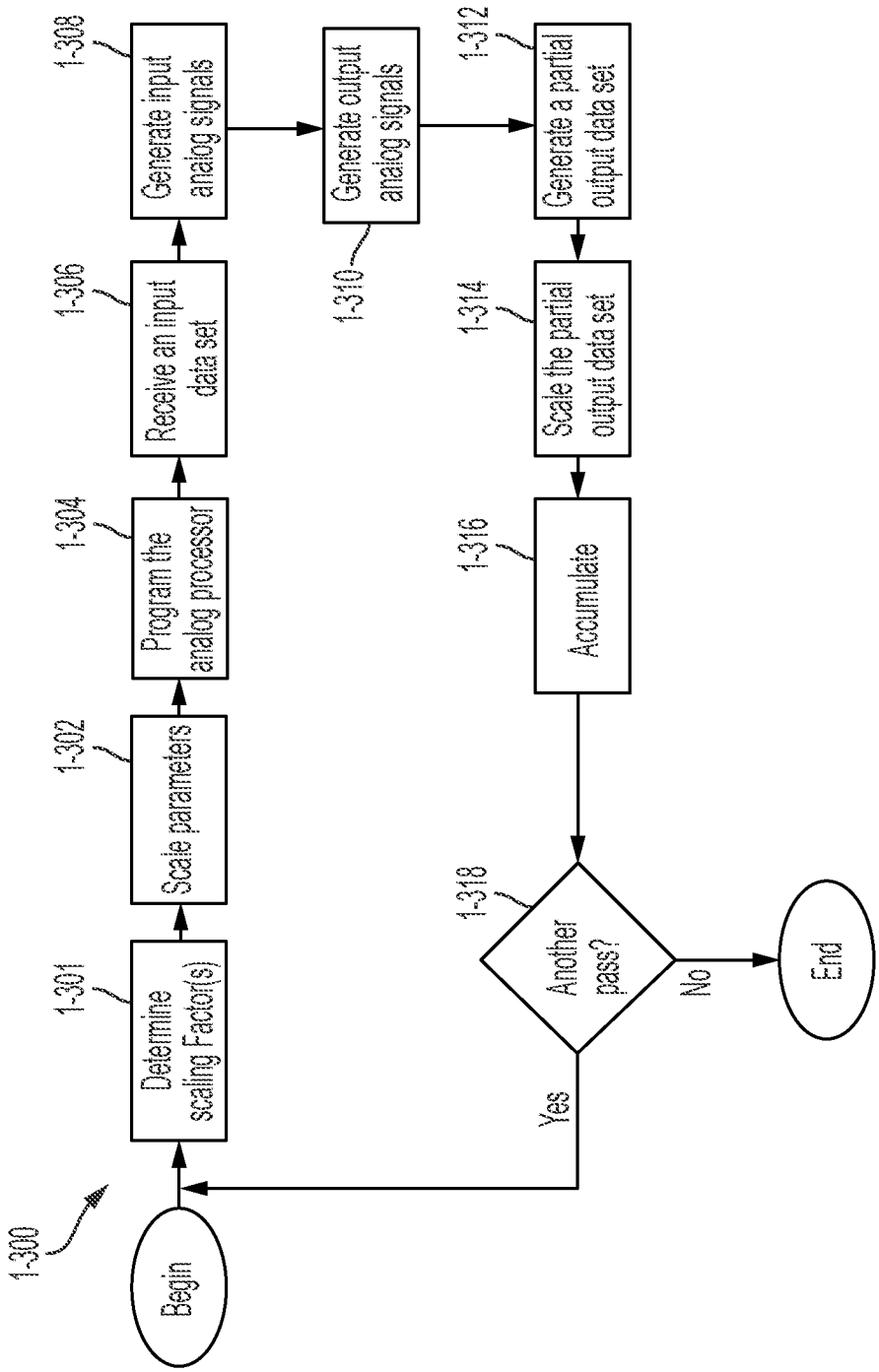

1-300
Begin
1-301
Determine scaling Factor(s)
1-302
Scale parameters
1-304
Program the analog processor
1-306
Receive an input data set
1-308
Generate input analog signals
1-310
Generate output analog signals
1-312
Generate a partial output data set
1-314
Scale the partial output data set
1-316
Accumulate
1-318
Another pass?
Yes
No
End

FIG. 1-3

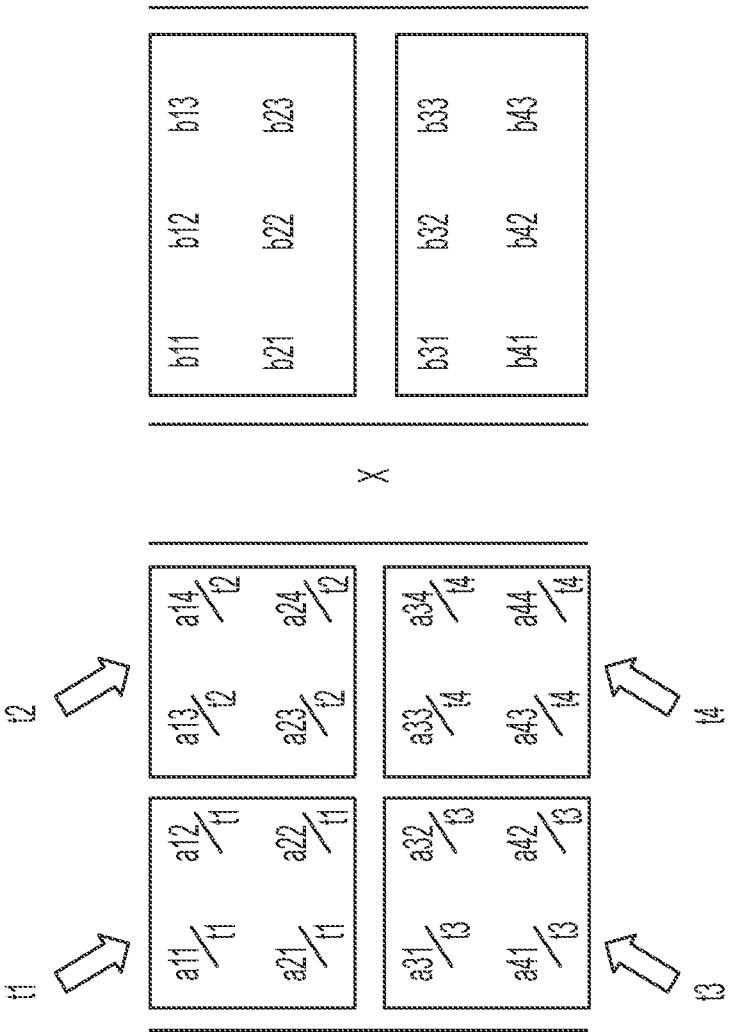

t1
t2
t3
t4
a11/t1
a12/t1
a21/t1
a22/t1
a13/t2
a14/t2
a23/t2
a24/t2
a31/t3
a32/t3
a41/t3
a42/t3
a33/t4
a34/t4
a43/t4
a44/t4
X
b11
b12
b13
b21
b22
b23
b31
b32
b33
b41
b42
b43

FIG. 1-4A

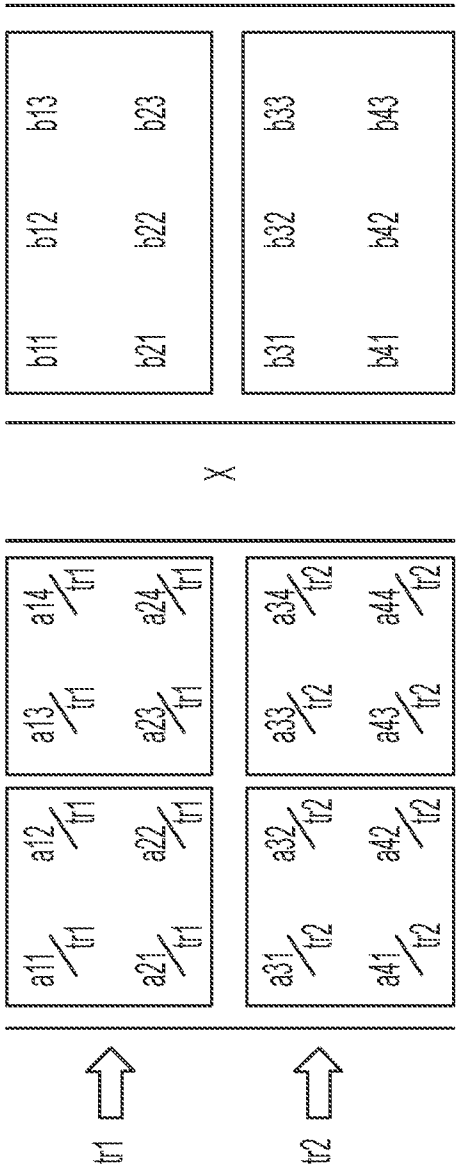

tr1
tr2
a11/tr1
a12/tr1
a13/tr1
a14/tr1
a21/tr1
a22/tr1
a23/tr1
a24/tr1
a31/tr2
a32/tr2
a33/tr2
a34/tr2
a41/tr2
a42/tr2
a43/tr2
a44/tr2
X
b11
b12
b13
b21
b22
b23
b31
b32
b33
b41
b42
b43

FIG. 1-4B

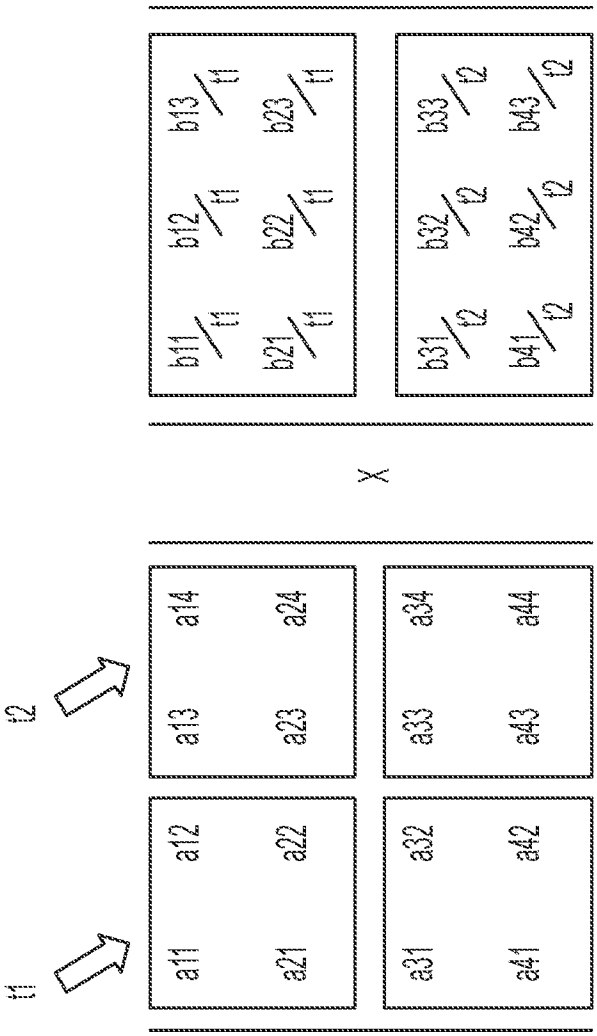

t1
t2
a11 a12 a14
a21 a22 a24
a13 a23
a31 a32 a34
a41 a42 a44
a33 a43
X
b11/t1 b12/t1 b13/t1
b21/t1 b22/t1 b23/t1
b31/t2 b32/t2 b33/t2
b41/t2 b42/t2 b43/t2

FIG. 1-4C

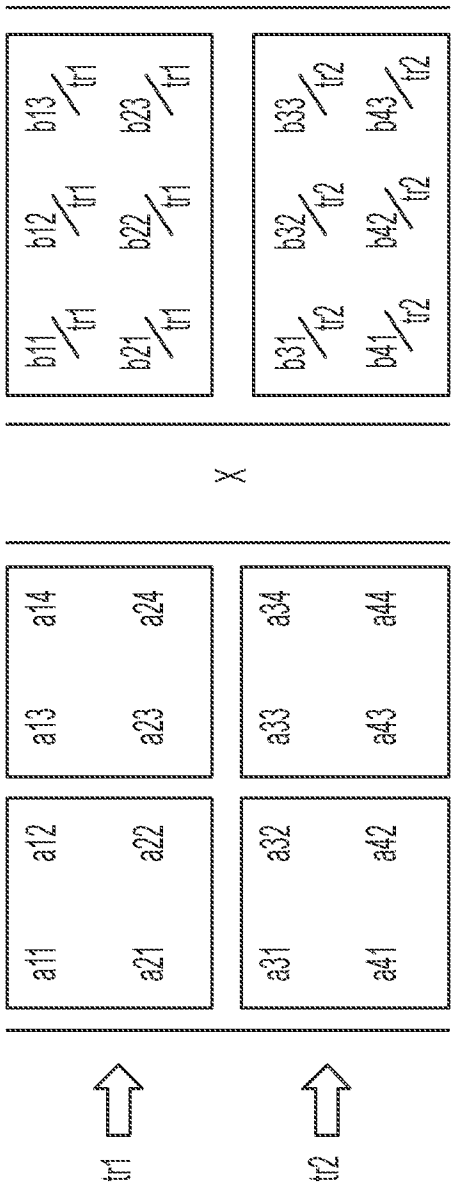

tr1
tr2
a11 a12 a13 a14
a21 a22 a23 a24
a31 a32 a33 a34
a41 a42 a43 a44
X
b11/tr1 b12/tr1 b13/tr1
b21/tr1 b22/tr1 b23/tr1
b31/tr2 b32/tr2 b33/tr2
b41/tr2 b42/tr2 b43/tr2

FIG. 1-4D

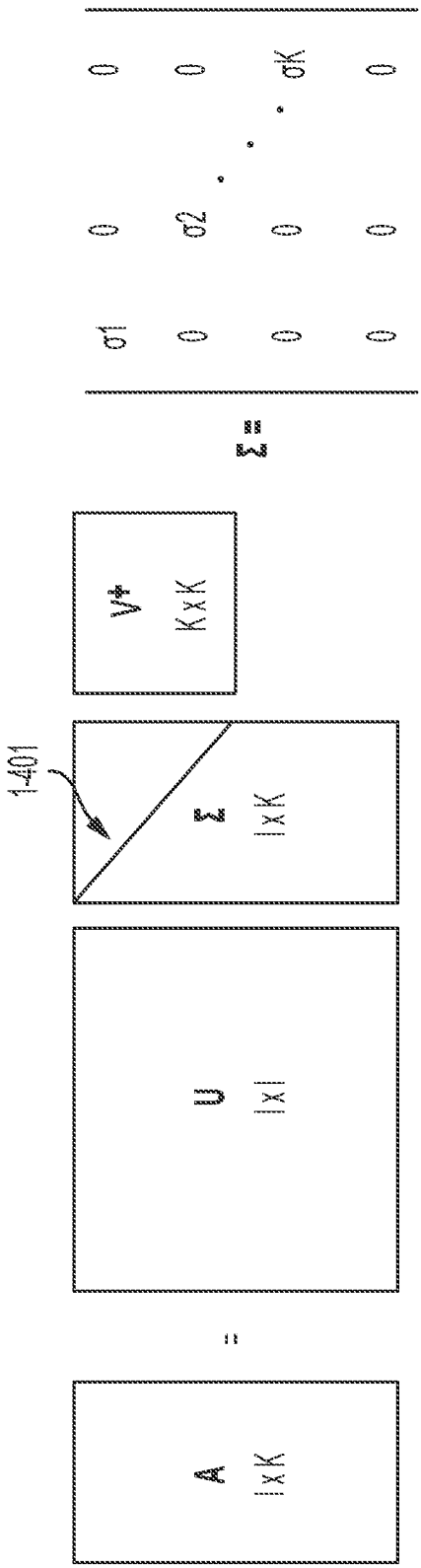


FIG. 1-5A

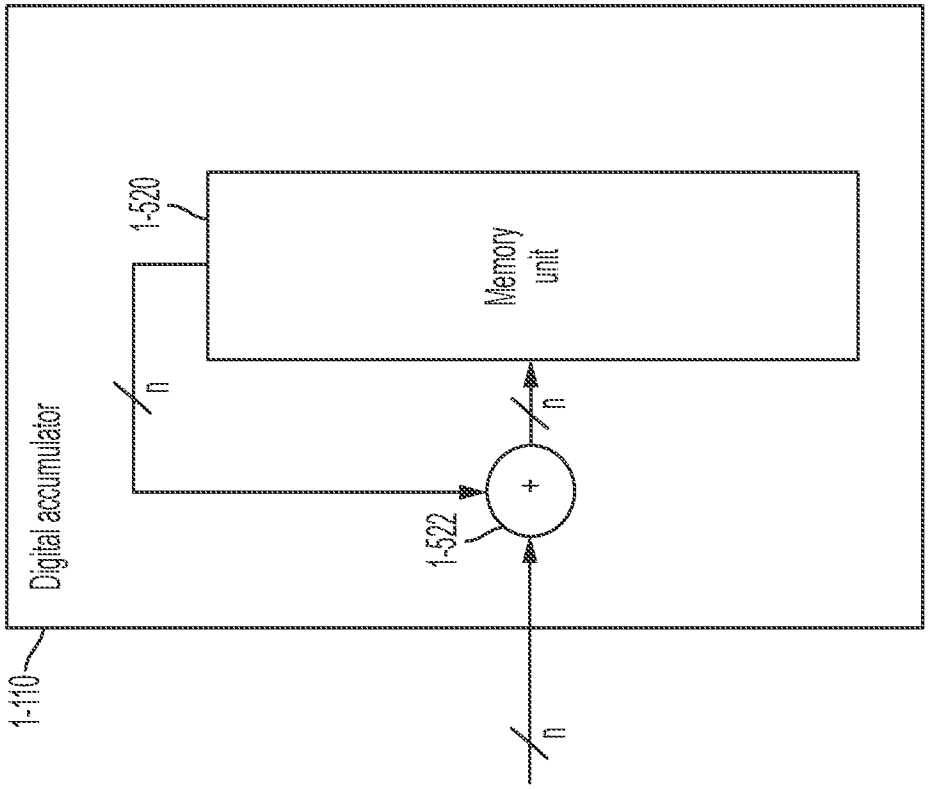


FIG. 1-5B

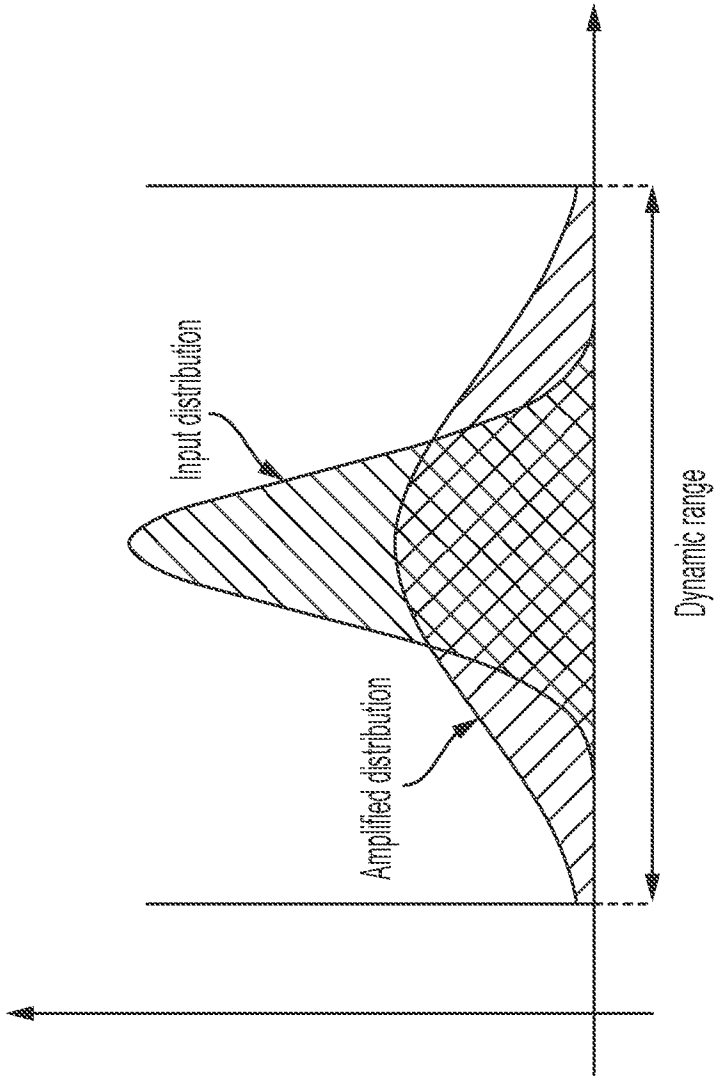

Input distribution
Amplified distribution
Dynamic range

FIG. 1-6A

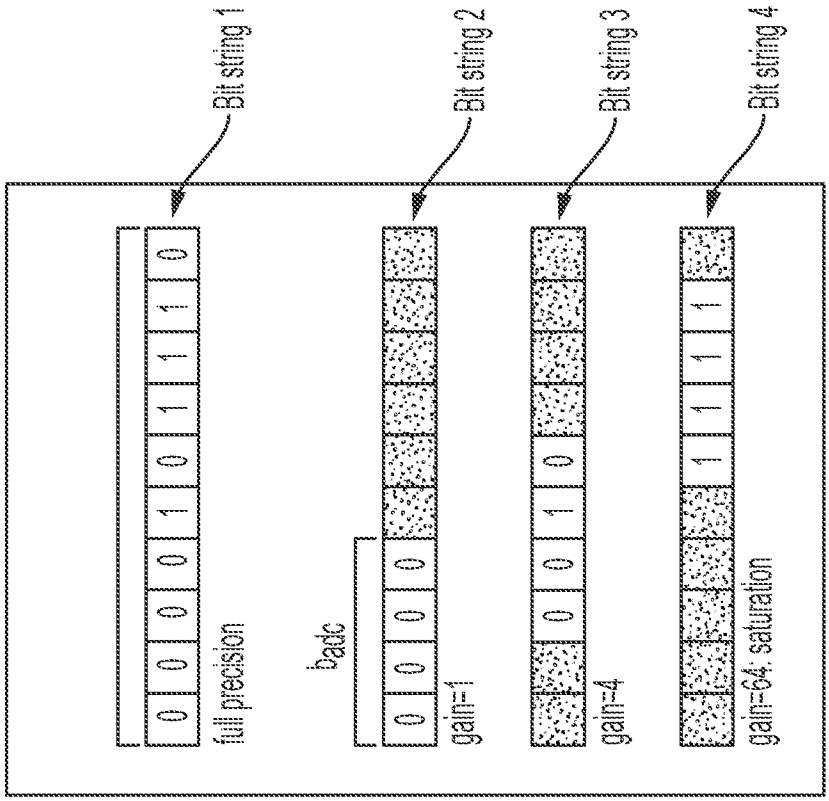

Bit string 1
Bit string 2
Bit string 3
Bit string 4
0 0 0 0 1 0 1 1 1 0
full precision
b_{adc}
0 0 0 0
gain=1
0 0 1 0
gain=4
1 1 1 1
gain=64 saturation

FIG. 1-6B

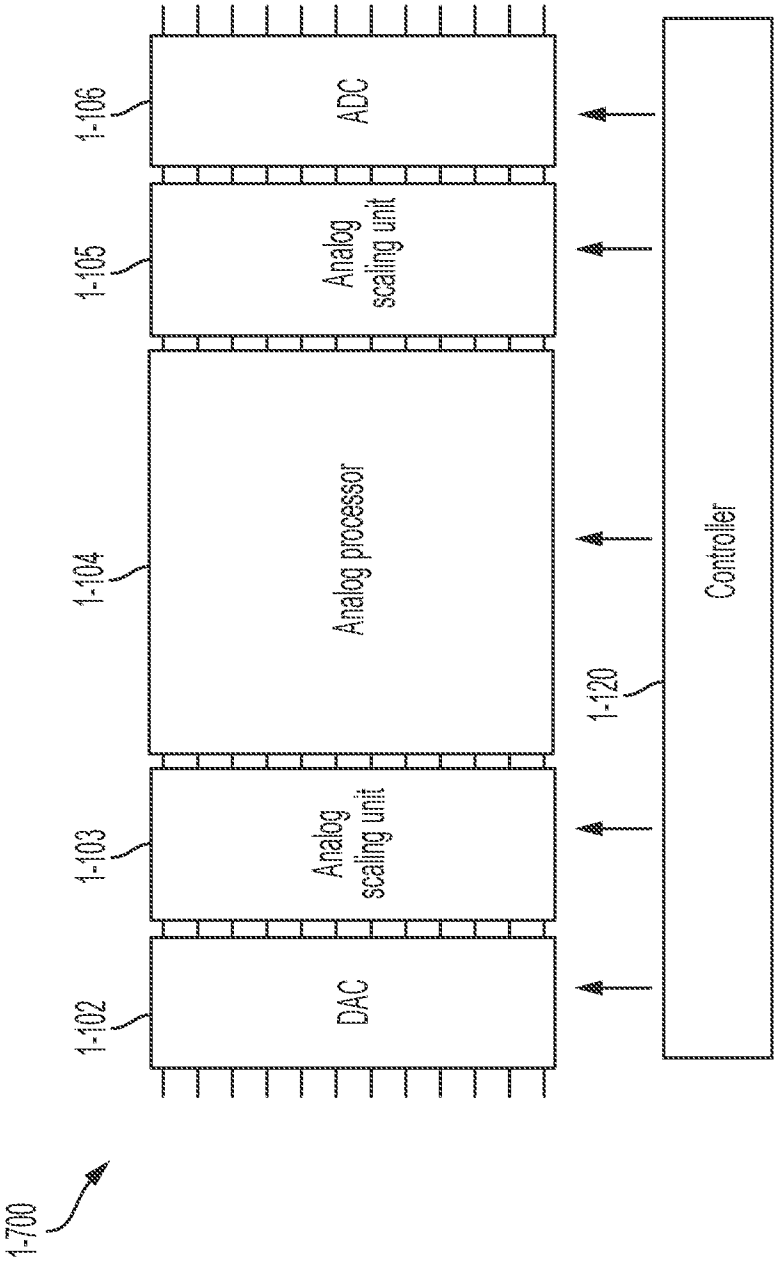

1-700
1-102
DAC
1-103
Analog scaling unit
1-104
Analog processor
1-105
Analog scaling unit
1-106
ADC
1-120
Controller

FIG. 1-7A

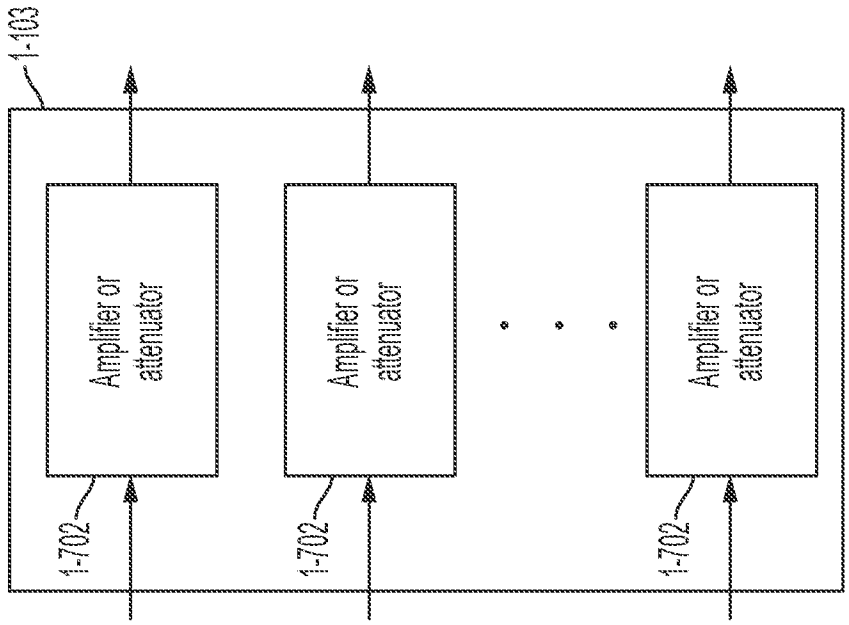

1-103
1-702
Amplifier or attenuator
1-702
Amplifier or attenuator
1-702
Amplifier or attenuator

FIG. 1-7B

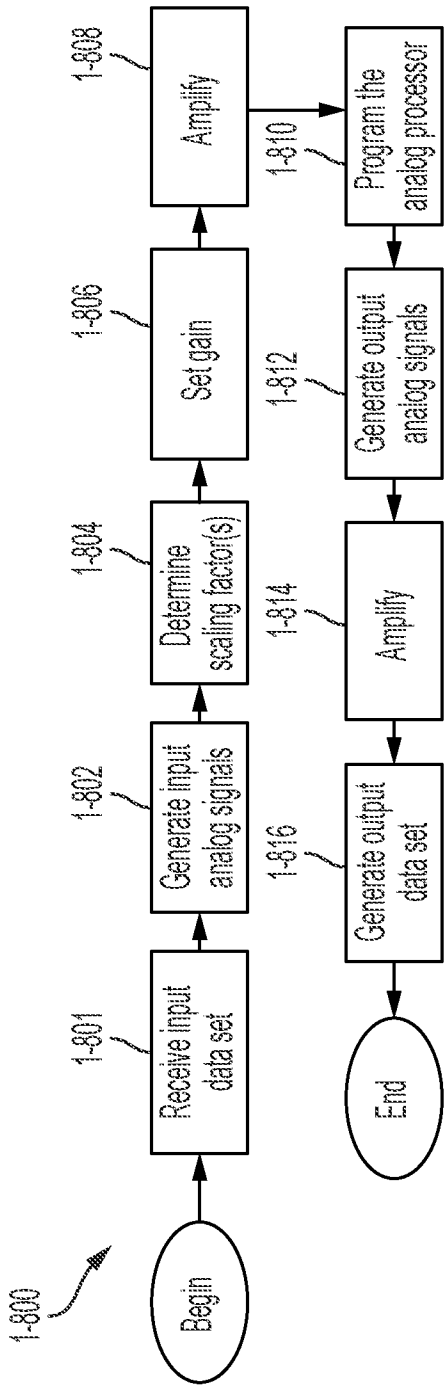

1-800
Begin
1-801
Receive input data set
1-802
Generate input analog signals
1-804
Determine scaling factor(s)
1-806
Set gain
1-808
Amplify
1-810
Program the analog processor
1-812
Generate output analog signals
1-814
Amplify
1-816
Generate output data set
End

FIG. 1-8

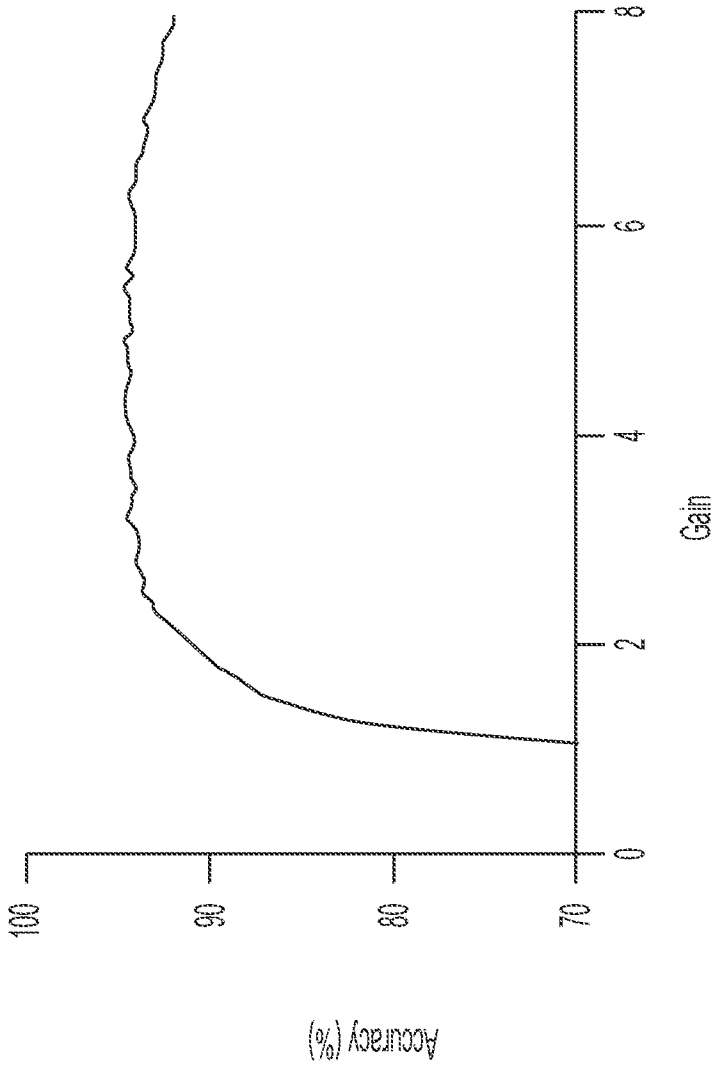


FIG. 1-9A

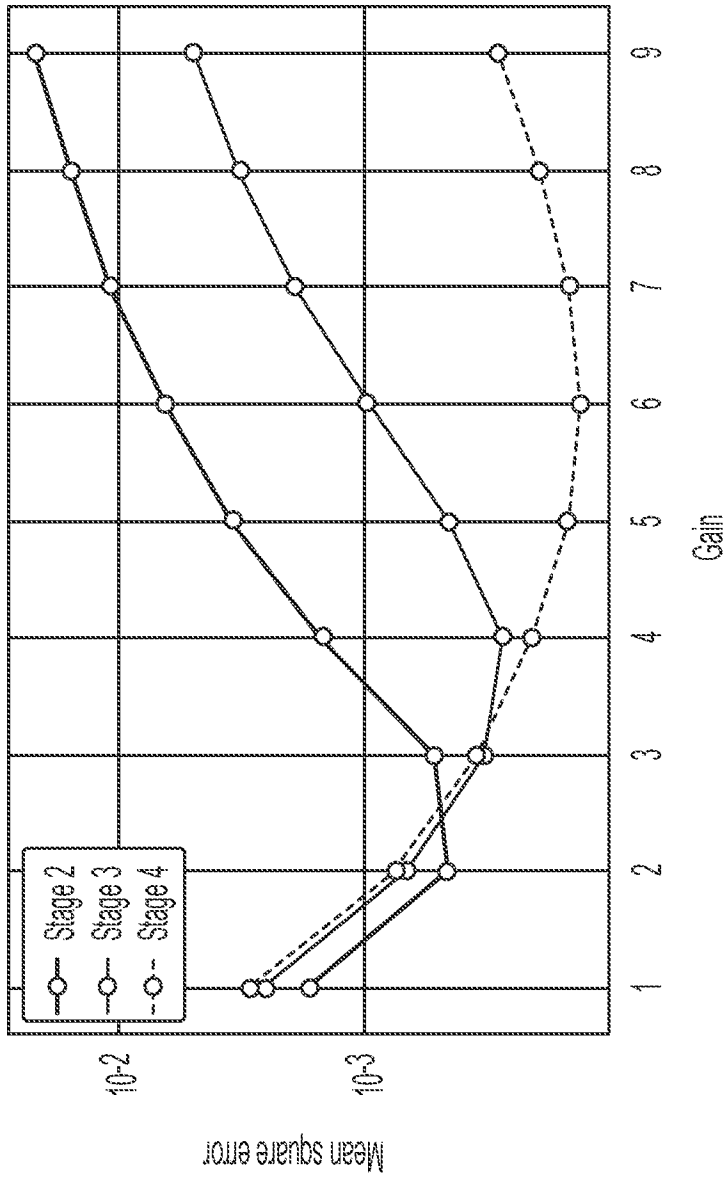

Mean square error
10-2
10-3
Stage 2
Stage 3
Stage 4
1
2
3
4
5
6
7
8
9
Gain

FIG. 1-9B

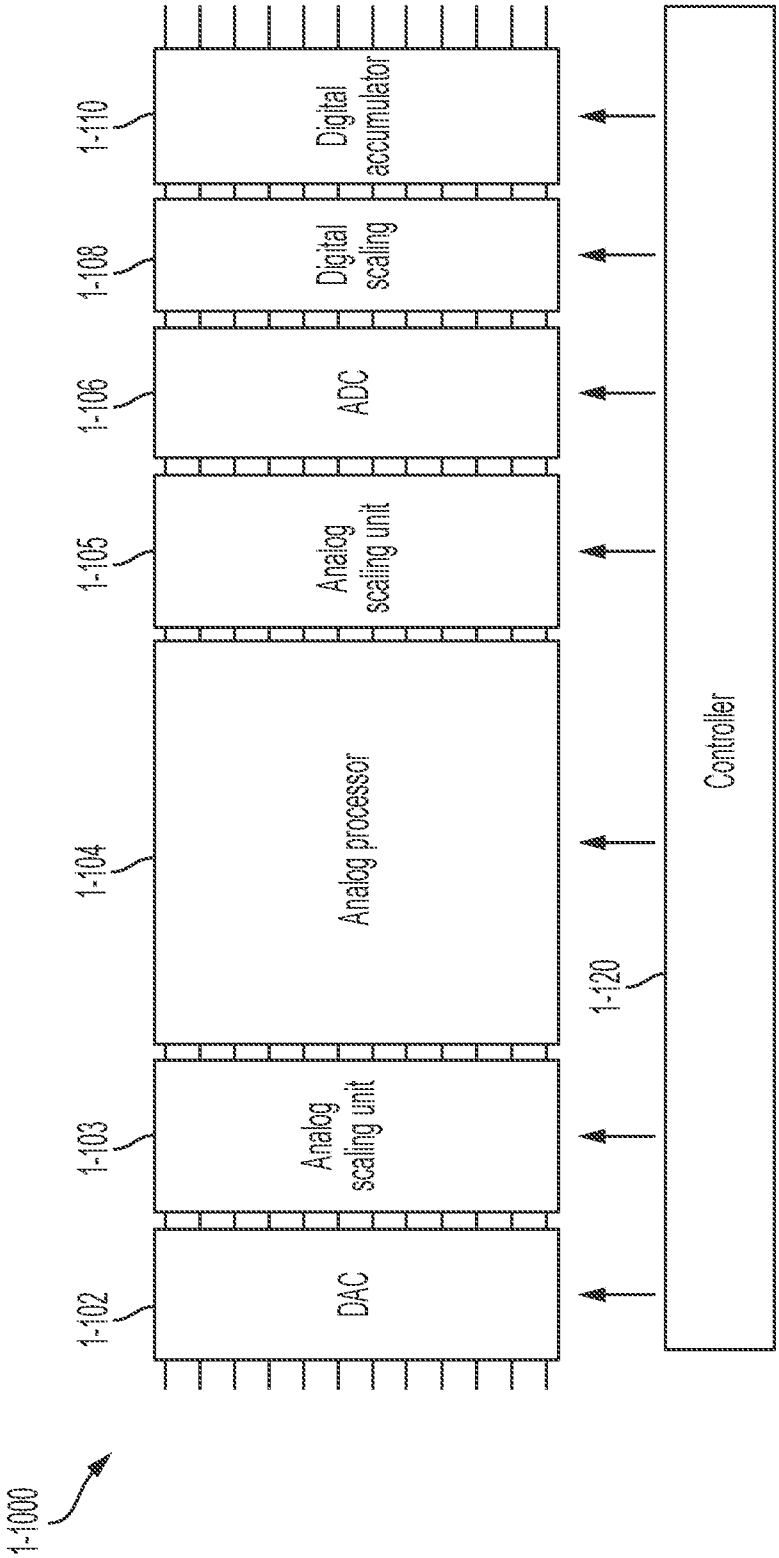

1-1000
1-102
DAC
1-103
Analog scaling unit
1-104
Analog processor
1-105
Analog scaling unit
1-106
ADC
1-108
Digital scaling
1-110
Digital accumulator
1-120
Controller

FIG. 1-10

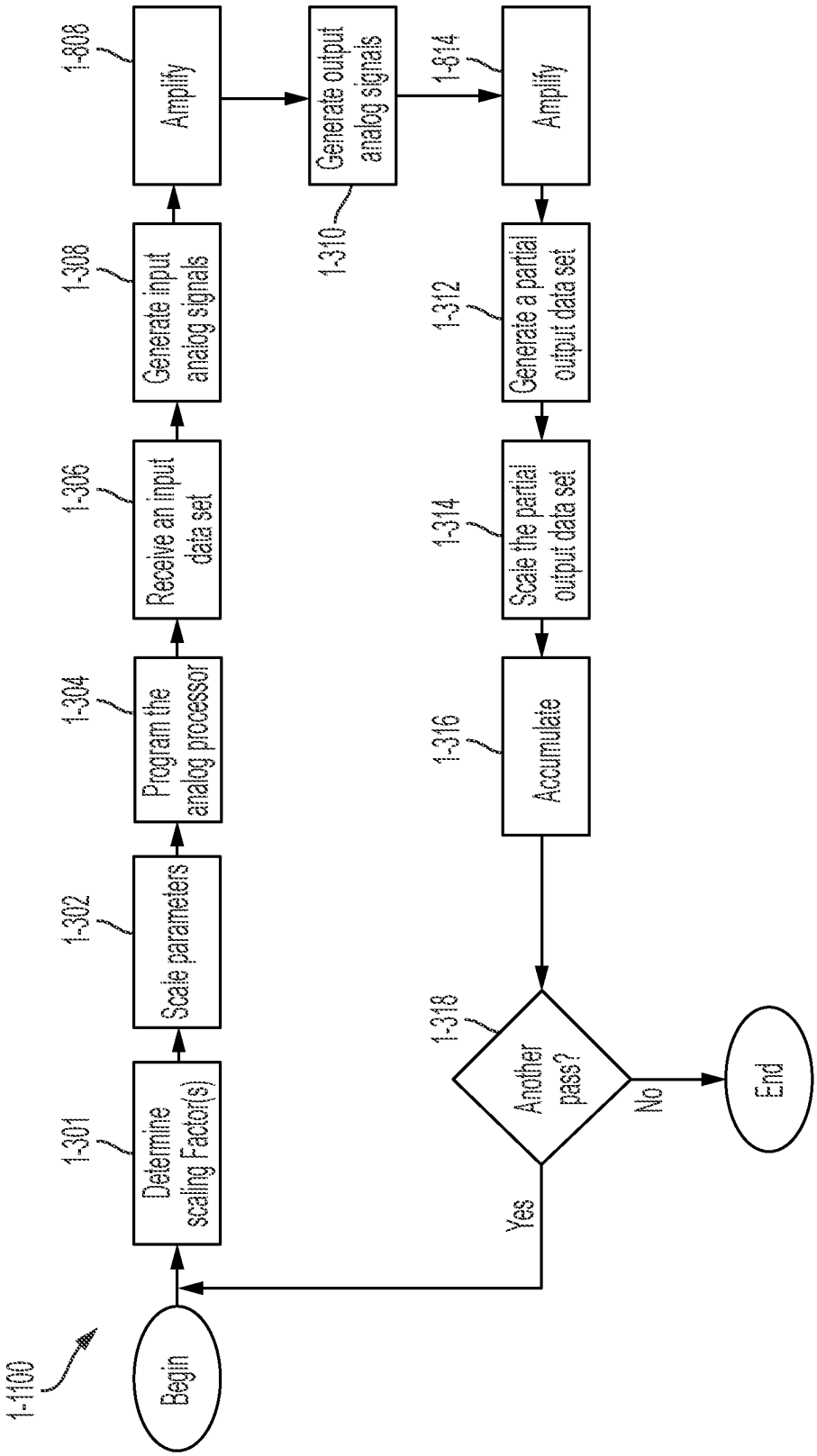

1-1100
Begin
1-301
Determine scaling Factor(s)
1-302
Scale parameters
1-304
Program the analog processor
1-306
Receive an input data set
1-308
Generate input analog signals
1-808
Amplify
1-310
Generate output analog signals
1-814
Amplify
1-312
Generate a partial output data set
1-314
Scale the partial output data set
1-316
Accumulate
1-318
Another pass?
Yes
No
End

FIG. 1-11

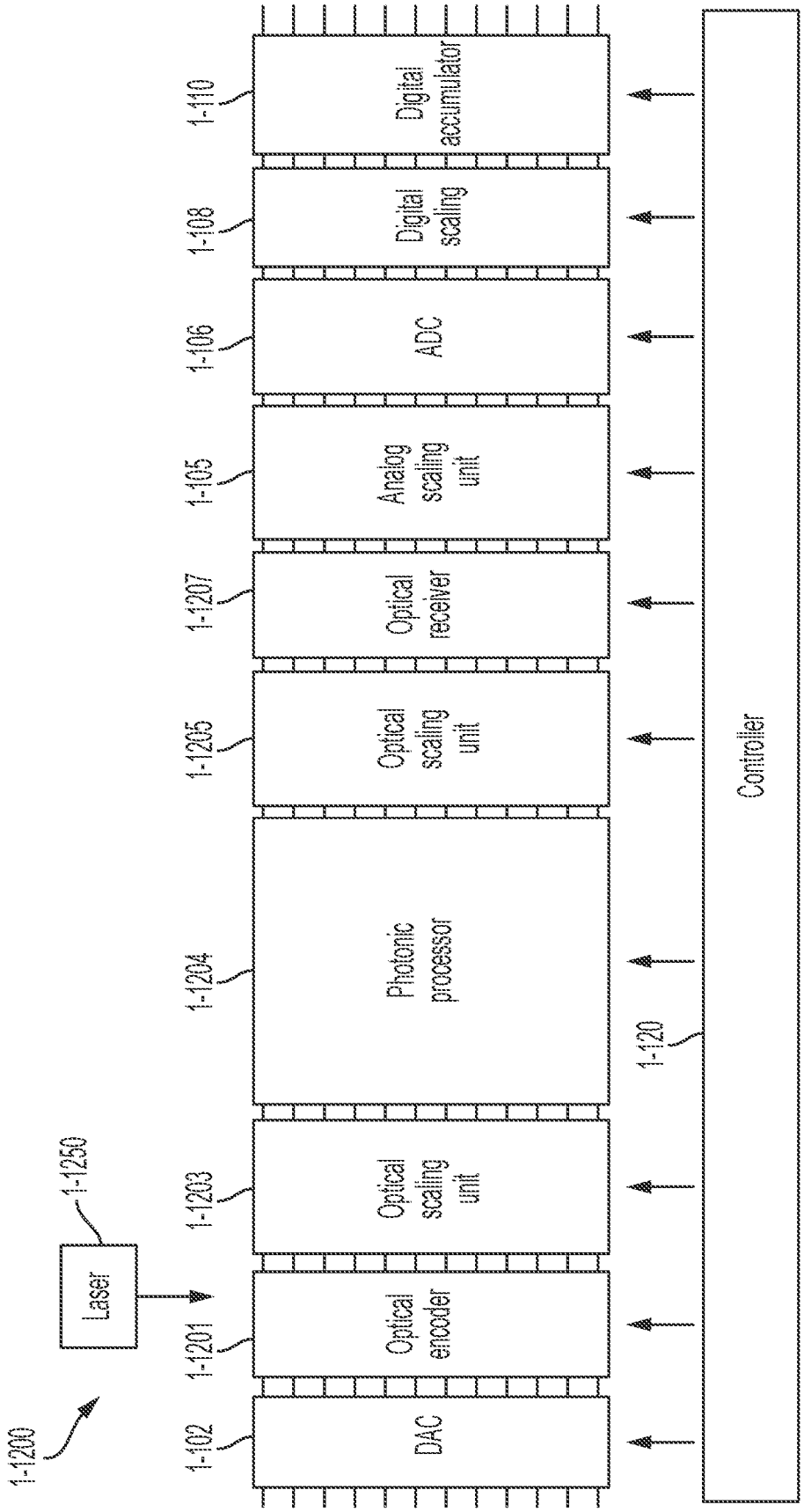

1-1200
1-1250
Laser
1-102
DAC
1-1201
Optical encoder
1-1203
Optical scaling unit
1-1204
Photonic processor
1-1205
Optical scaling unit
1-1207
Optical receiver
1-105
Analog scaling unit
1-106
ADC
1-108
Digital scaling
1-110
Digital accumulator
1-120
Controller

FIG. 1-12

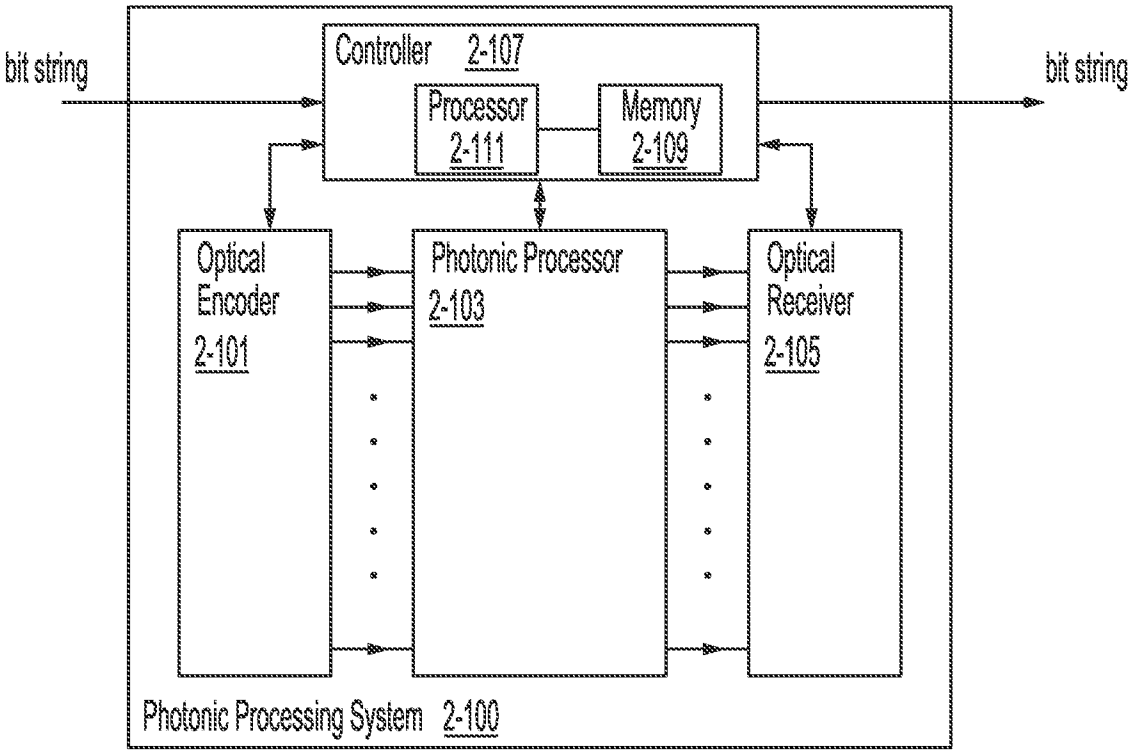

bit string
Controller 2-107
Processor 2-111
Memory 2-109
bit string
Optical Encoder 2-101
Photonic Processor 2-103
Optical Receiver 2-105
Photonic Processing System 2-100

FIG. 2-1

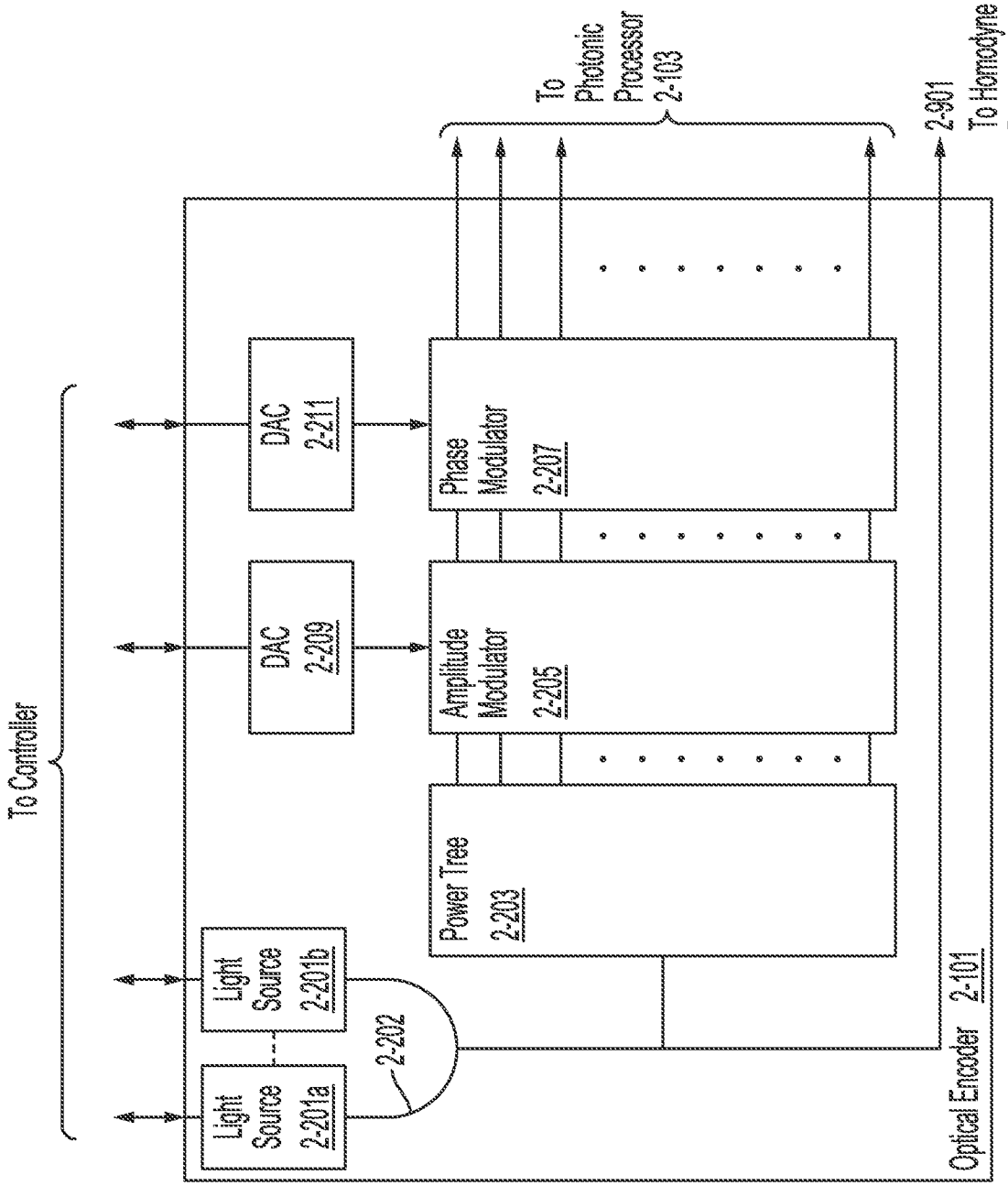

To Controller
Light Source 2-201a
Light Source 2-201b
2-202
DAC 2-211
DAC 2-209
Power Tree 2-203
Amplitude Modulator 2-205
Phase Modulator 2-207
To Photonic Processor 2-103
2-901 To Homodyne Detector
Optical Encoder 2-101

FIG. 2-2

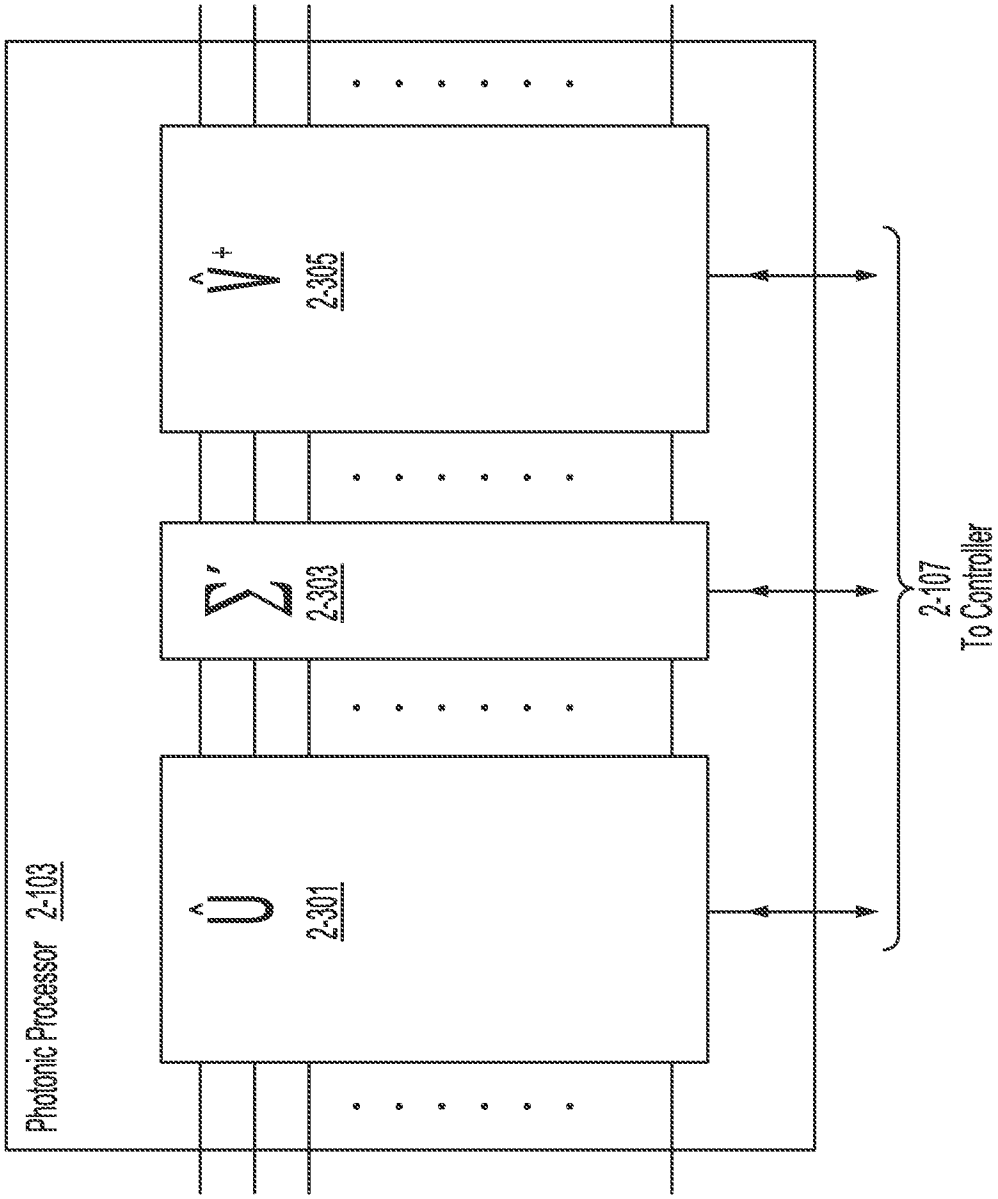

Photonic Processor 2-103
$\hat{U}$
2-301
$\hat{\Sigma}'$
2-303
$\hat{V}^{+}$
2-305
2-107
To Controller

FIG. 2-3

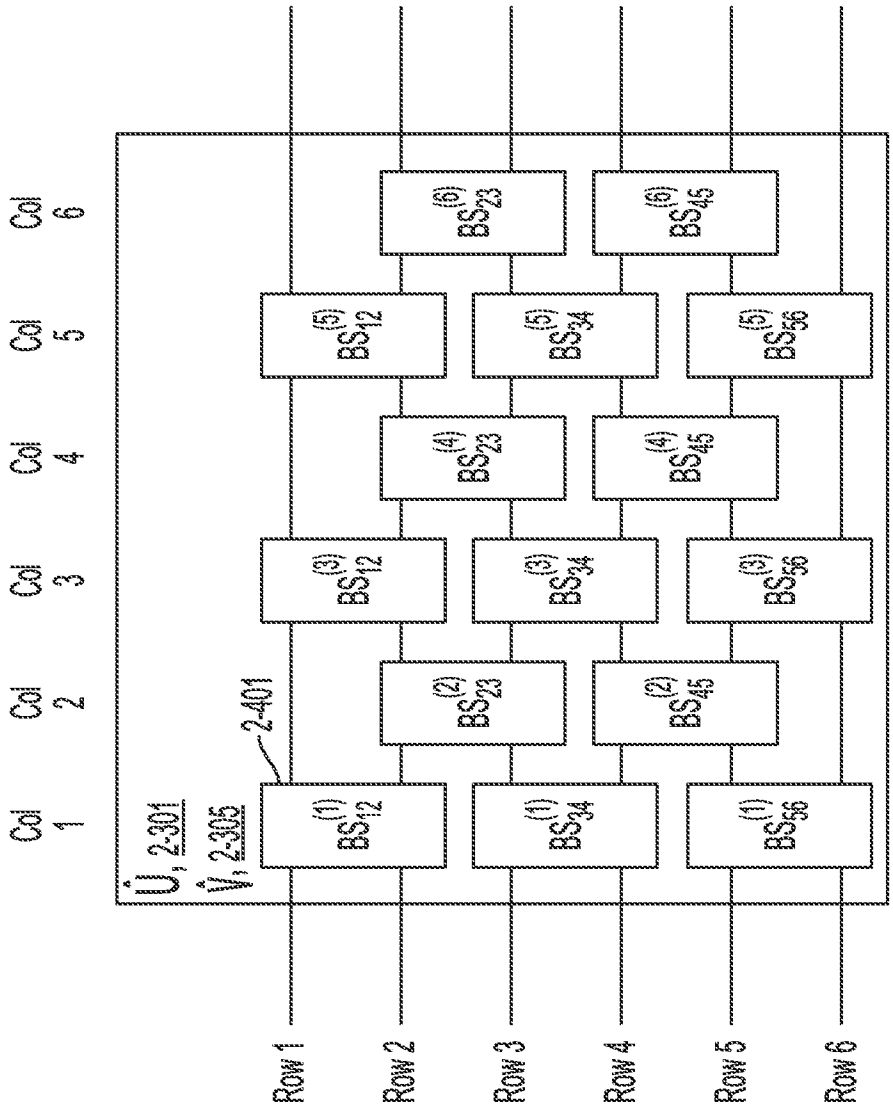

Col 1
Col 2
Col 3
Col 4
Col 5
Col 6
Row 1
Row 2
Row 3
Row 4
Row 5
Row 6
Û, 2-301
V̂, 2-305
2-401
$BS_{12}^{(1)}$
$BS_{23}^{(2)}$
$BS_{12}^{(3)}$
$BS_{23}^{(4)}$
$BS_{12}^{(5)}$
$BS_{23}^{(6)}$
$BS_{34}^{(1)}$
$BS_{45}^{(2)}$
$BS_{34}^{(3)}$
$BS_{45}^{(4)}$
$BS_{34}^{(5)}$
$BS_{45}^{(6)}$
$BS_{56}^{(1)}$
$BS_{56}^{(3)}$
$BS_{56}^{(5)}$

FIG. 2-4

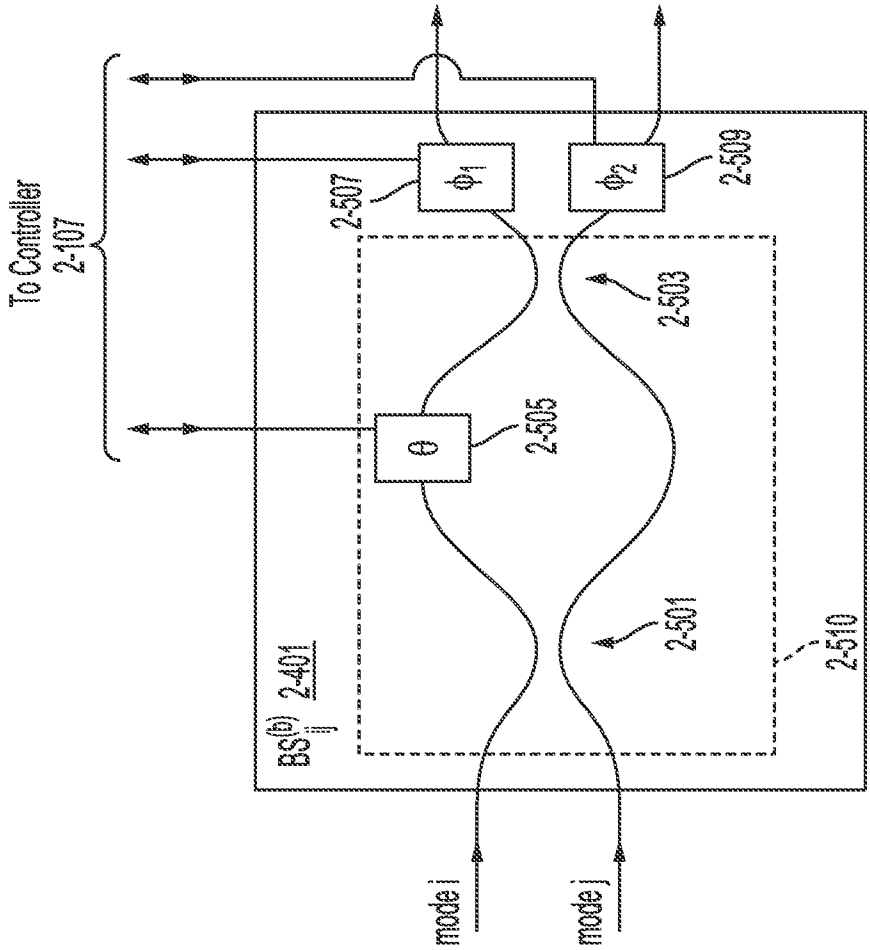

To Controller
2-107
2-507
ϕ_1
ϕ_2
2-509
2-503
2-505
θ
2-501
2-510
$BS_{ij}^{(b)}$ 2-401
mode i
mode j

FIG. 2-5

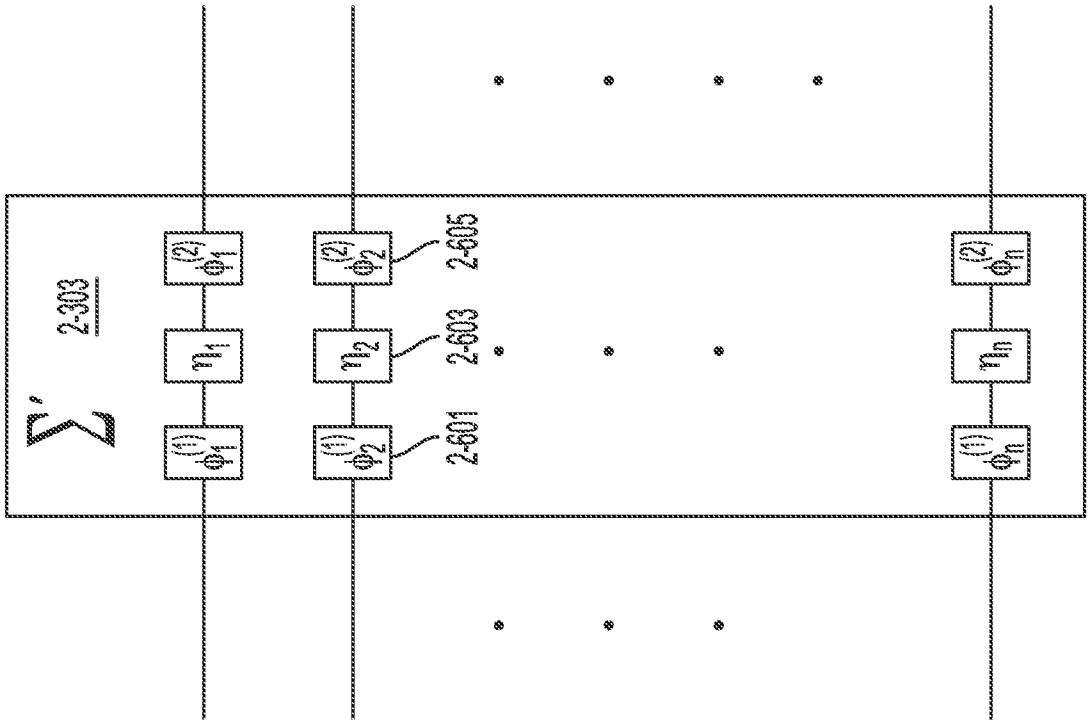

2-303
Σ'
$\phi_1^{(1)}$
η_1
$\phi_1^{(2)}$
$\phi_2^{(1)}$
η_2
$\phi_2^{(2)}$
$\phi_n^{(1)}$
η_n
$\phi_n^{(2)}$
2-601
2-603
2-605

FIG. 2-6

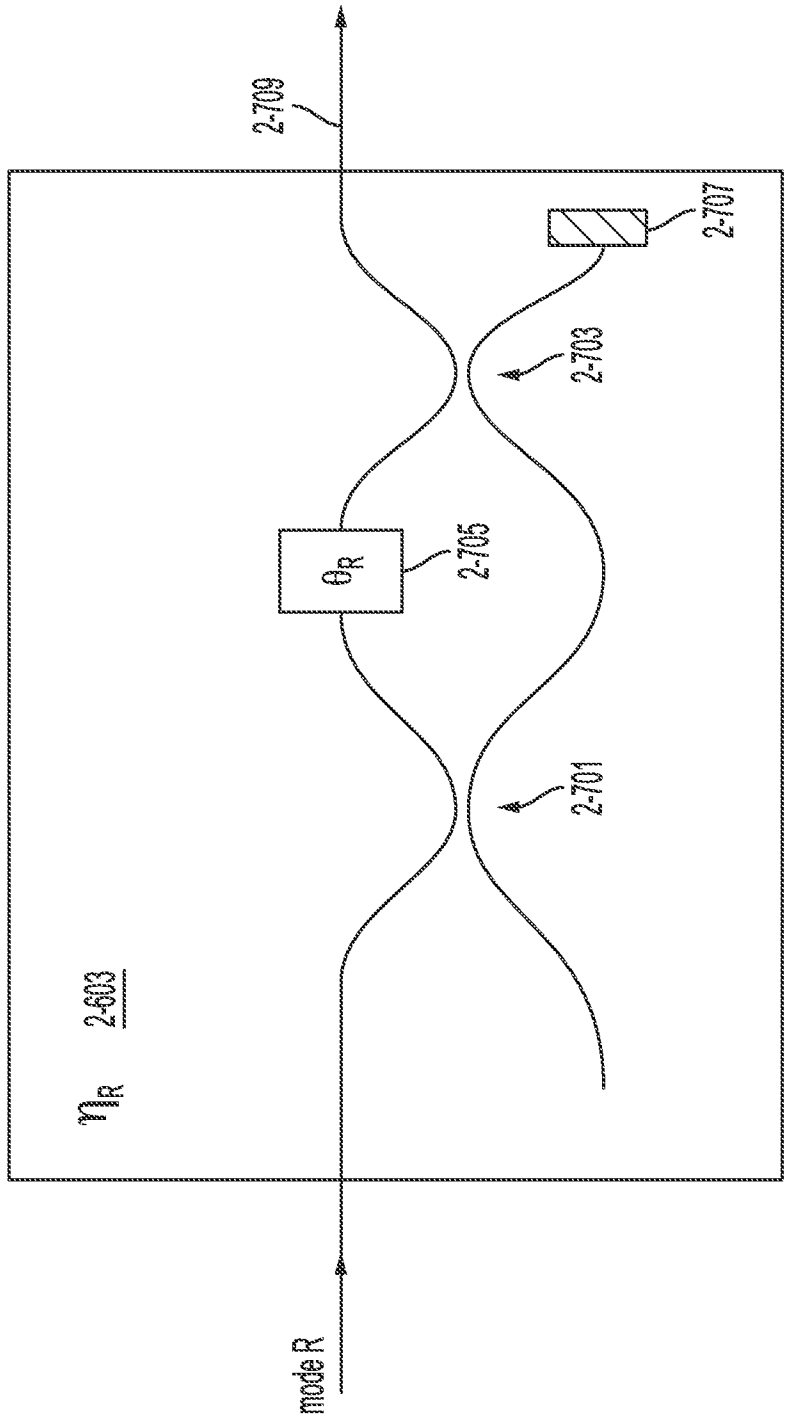


FIG. 2-7

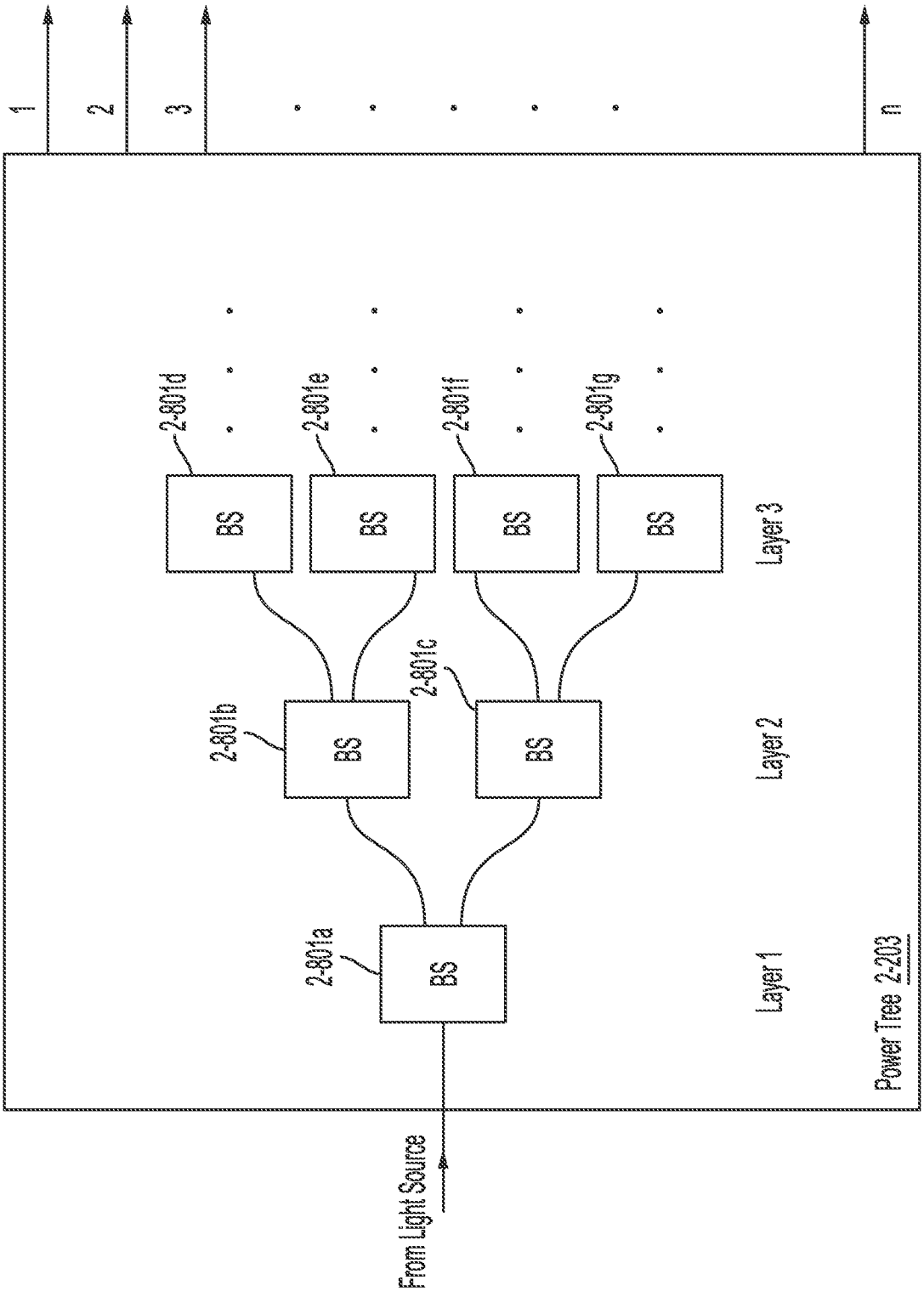

1
2
3
n
2-801d
2-801e
2-801f
2-801g
BS
BS
BS
BS
Layer 3
2-801c
2-801b
BS
BS
Layer 2
2-801a
BS
Layer 1
Power Tree 2-203
From Light Source

FIG. 2-8

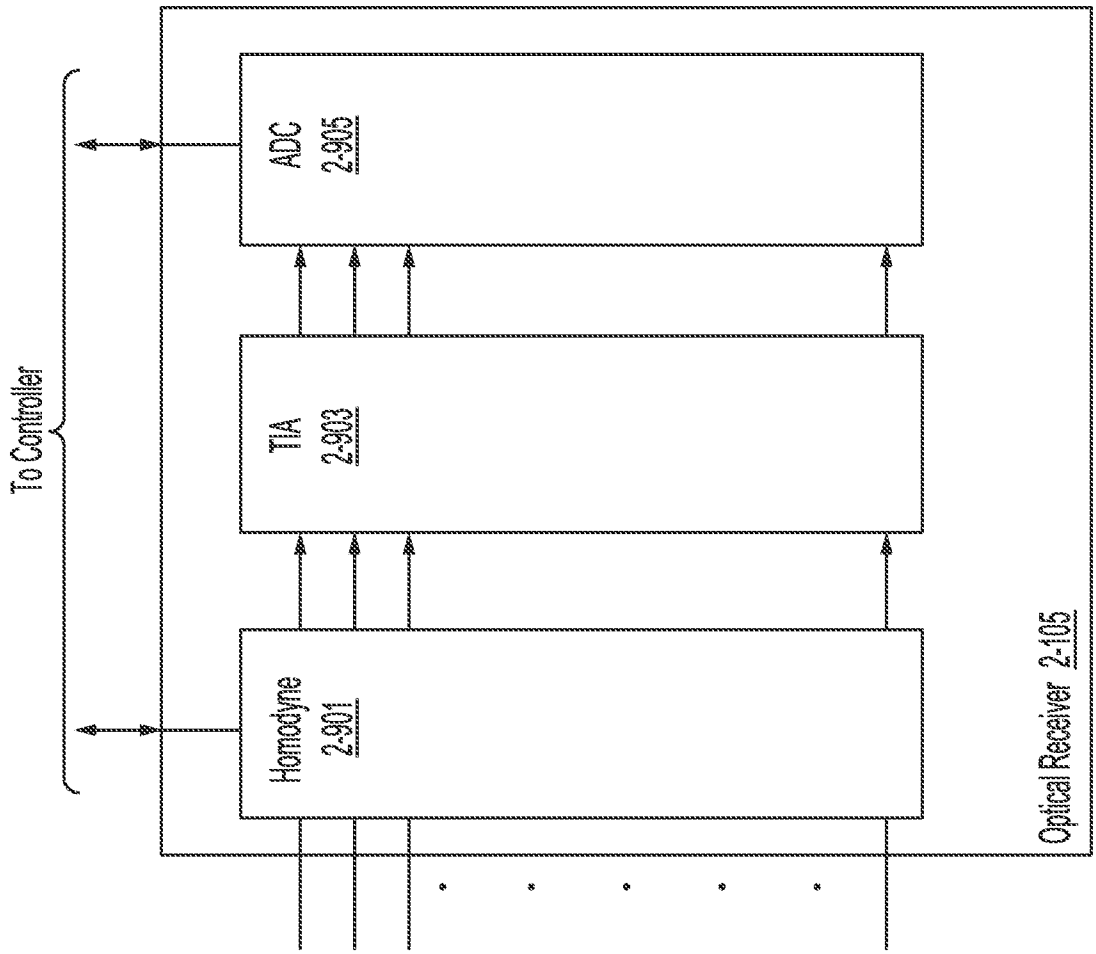

To Controller
Homodyne
2-901
TIA
2-903
ADC
2-905
Optical Receiver 2-105

FIG. 2-9

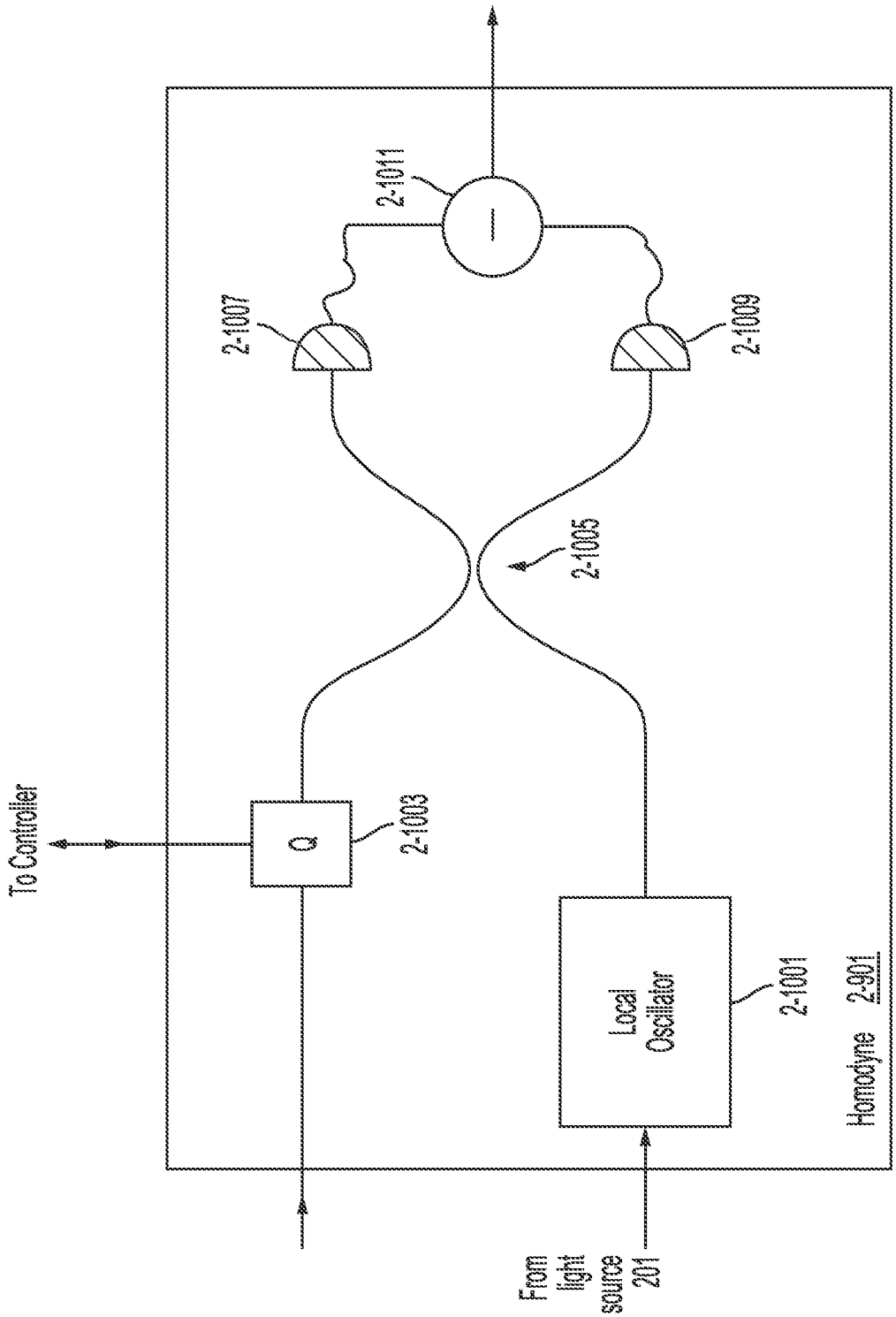

To Controller
Q
2-1003
2-1007
2-1011
2-1009
2-1005
Local Oscillator
2-1001
Homodyne 2-901
From light source 201

FIG. 2-10

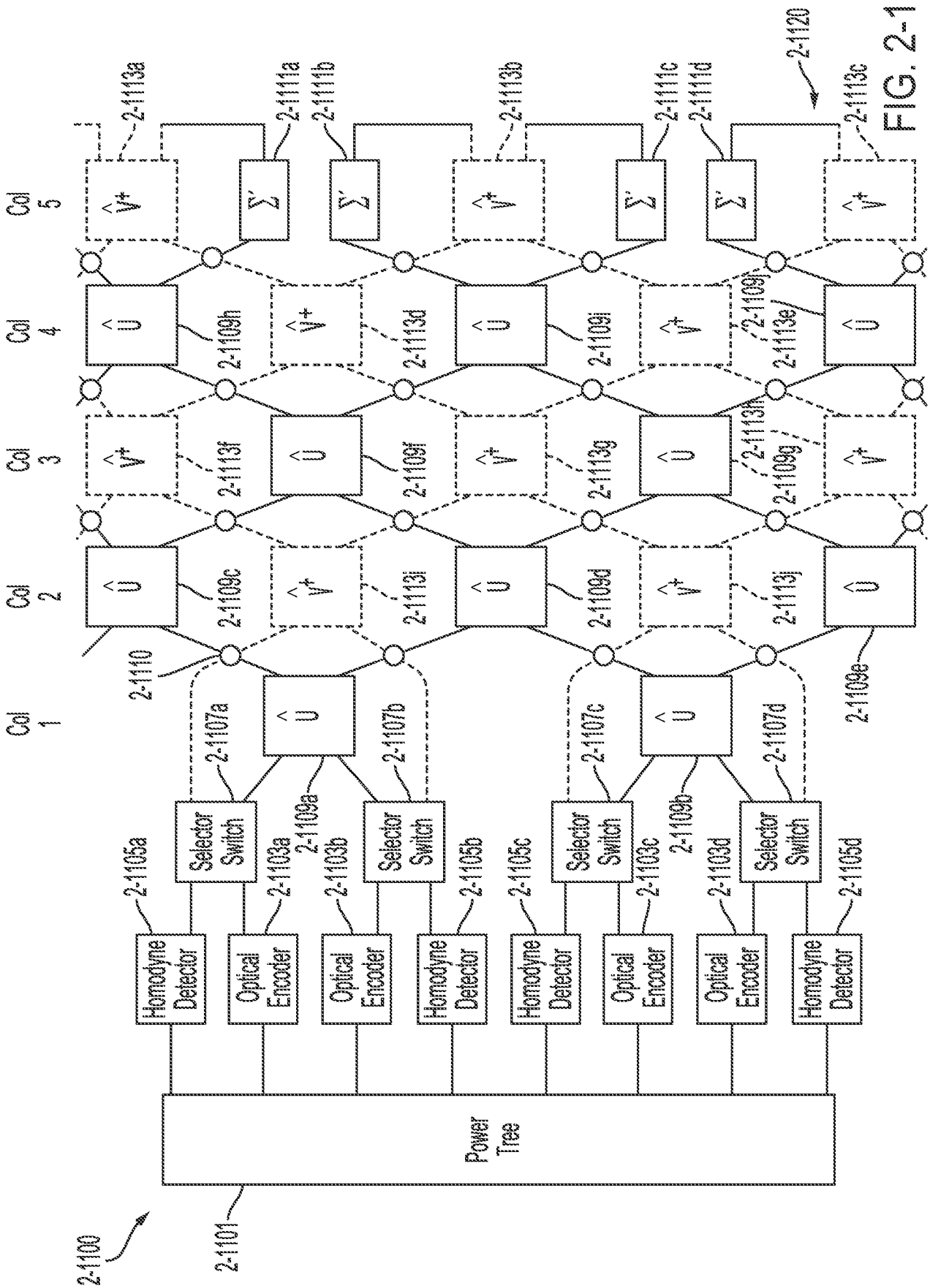

Col 1
Col 2
Col 3
Col 4
Col 5
2-1100
2-1101
Power Tree
Homodyne Detector
Optical Encoder
Selector Switch
2-1105a
2-1103a
2-1107a
2-1109a
2-1103b
2-1107b
2-1105b
2-1105c
2-1103c
2-1107c
2-1109b
2-1103d
2-1107d
2-1105d
2-1110
2-1109c
2-1113i
2-1109d
2-1113j
2-1109e
2-1113f
2-1109f
2-1113g
2-1109g
2-1113h
2-1113d
2-1109h
2-1109i
2-1113e
2-1109j
2-1113a
2-1111a
2-1111b
2-1113b
2-1111c
2-1111d
2-1113c
2-1120

FIG. 2-11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20/19747

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

See extra sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Claims 1-38, 47-81 and 87-99

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20/19747

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G06F 17/10, 17/16, 5/01, 7/32, 7/44, 7/52, 7/523; G06N 3/02, 20/00 (2020.01)

CPC - G06F 17/10, 17/16, 5/01, 7/32, 7/44, 7/52, 7/523; G06E 1/045; G06N 3/02, 20/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017/0372202 A1 (NVIDIA CORPORATION) 28 December 2017; the entire document	1-38, 47-81, 87-99
A	US 5,995,748 A (GUTTAG, K et al.) 30 November 1999; the entire document	1-38, 47-81, 87-99
A	US 2008/0304600 A1 (BOTTOMLEY, G) 11 December 2008; the entire document	1-38, 47-81, 87-99

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 April 2020 (13.04.2020)	02 JUL 2020
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Shane Thomas Telephone No. PCT Helpdesk: 571-272-4300

-***-Continued from Box No. III Observations where unity of invention is lacking-***-

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fee must be paid.

Group I: Claims 1-38, 47-81 and 87-99 are directed towards a hybrid analog-digital processor to perform mathematical operations based on scaling factors and scaled partial output data.
Group II: Claims 39-46, 82-86 and 100-103 are directed towards a hybrid analog-digital processor comprising a plurality of DACs.

The inventions listed as Groups I-II do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

The special technical features of Group I include at least wherein: circuitry comprising an analog processor and an analog scaling unit, wherein the circuitry is configured to: generate a plurality of input analog signals based on an input data set; one or more scaling factors; program the analog processor with a set of parameters; generate a plurality of output analog signals based on the plurality of input analog signals and the set of parameters; and generate an output data set, which are not present in Group II.

The special technical features of Group II include at least wherein a digital-to-analog converter (DAC) unit comprising a plurality of DACs; a processor arranged to perform matrix-matrix multiplication and coupled to outputs of the DAC unit; an analog-to-digital "ADC" unit comprising a plurality of ADCs and coupled to outputs of the analog processor; a digital scaling unit coupled to outputs of the ADC unit; and a digital accumulator coupled to outputs of the digital scaling unit, wherein the digital accumulator comprises a memory unit and a digital adder, which are not present in Group I.

The common technical features shared by Groups I-II are a hybrid analog-digital processor comprising: a processor; a scaling unit. However, these common features are previously disclosed by US 5,550,993 A to EHLIG, P et al. (hereinafter "EHLIG"). EHLIG discloses a hybrid analog-digital processor comprising (a data processing device; abstract): a processor (a processor 11; column 6, lines 5-17); a scaling unit (a processor scaling shifter; column 6, lines 5-17).

Since the common technical features are previously disclosed by the EHLIG reference, these common features are not special and so Groups I-II lack unity.