



(51) International Patent Classification:

G06N 3/04 (2006.01) G06N 3/063 (2006.01)
G06F 15/76 (2006.01)

(21) International Application Number:

PCT/IB2020/057724

(22) International Filing Date:

17 August 2020 (17.08.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/887,925 16 August 2019 (16.08.2019) US
62/904,142 23 September 2019 (23.09.2019) US
62/929,233 01 November 2019 (01.11.2019) US
62/983,076 28 February 2020 (28.02.2020) US

(71) Applicant: **UNTETHER AI CORPORATION** [CA/CA];
300 - 134 Peter St, Toronto, Ontario M5V 2H2 (CA).

(72) Inventors: **SNELGROVE, William Martin**; 300 - 134 Peter St, Toronto, Ontario M5V 2H2 (CA). **KITAMURA, John**; 300 - 134 Peter St, Toronto, Ontario M5V 2H2 (CA).

(74) Agent: **SMITH, Ryan T** et al.; Perry + Currier, 1300 Yonge Street, Suite 500, Toronto, Ontario M4T 1X3 (CA).

(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, IT, 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.

(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).

(54) Title: COMPUTATIONAL MEMORY WITH PROCESSING ELEMENT ROW AND BANK COMMUNICATIONS

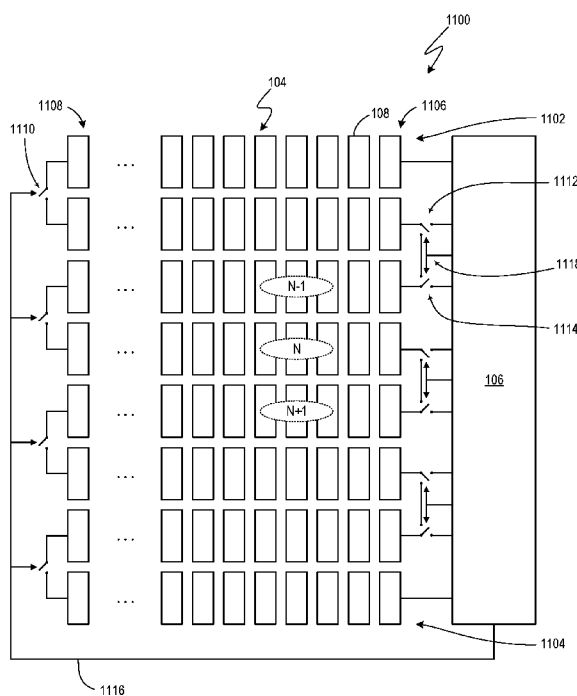


FIG. 11

(57) Abstract: An example device includes an array of rows of processing elements. Each row includes a plurality of connected processing elements. Each row terminates at a first-end processing element and a second-end processing element opposite the first-end processing element. The device further includes a controller connected to a first-end processing element of a row of the array of rows to provide data and instructions to the row to operate the processing elements of the row according to a single instruction, multiple data scheme. The device further includes a second-end switch to selectively connect second-end processing elements of adjacent rows of the array of rows to make a second-end interrow connection that allows operation of the adjacent rows as a double row of processing elements.

WO 2021/033125 A1

Published:

- *with international search report (Art. 21(3))*
- *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

COMPUTATIONAL MEMORY WITH PROCESSING ELEMENT ROW AND BANK COMMUNICATIONS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to US provisional patent application serial nos. 62/887,925 (filed Aug. 16, 2019), 62/904,142 (filed Sep. 23, 2019), 62/929,233 (filed Nov. 1, 2019), and 62/983,076 (filed Feb. 28, 2020), all of which are incorporated herein by reference.

[0002] Deep learning has proven to be a powerful technique for performing functions that have long resisted other artificial intelligence approaches. For example, deep learning may be applied to recognition of objects in cluttered images, speech understanding and translation, medical diagnosis, gaming, and robotics. Deep learning techniques typically apply many layers (hence “deep”) of neural networks that are trained (hence “learning”) on the tasks of interest. Once trained, a neural network may perform “inference”, that is, inferring from new input data an output consistent with what it has learned.

[0003] Neural networks, which may also be called neural nets, perform computations analogous to the operations of biological neurons, typically computing weighted sums (or dot products) and modifying the results with a memoryless nonlinearity. However, it is often the case that more general functionality, such as memory, multiplicative nonlinearities, and “pooling”, are also required.

[0004] In many types of computer architecture, power consumption due to physically moving data between memory and processing elements is non-trivial and is frequently the dominant use of power. This power consumption is typically due to the energy required to charge and discharge the capacitance of wiring, which is roughly proportional to the length of the wiring and hence to distance between memory and processing elements. As

such, processing a large number of computations in such architectures, as generally required for deep learning and neural networks, often requires a relatively large amount of power. In architectures that are better suited to handle deep learning and neural networks, other inefficiencies may arise, such as increased complexity, increased processing time, and larger chip area requirements.

SUMMARY

[0005] According to an aspect of this disclosure, a device includes an array of rows of processing elements. Each row includes a plurality of connected processing elements. Each row terminates at a first-end processing element and a second-end processing element opposite the first-end processing element. The device further includes a controller connected to a first-end processing element of a row of the array of rows to provide data and instructions to the row to operate the processing elements of the row according to a single instruction, multiple data scheme. The device further includes a second-end switch to selectively connect second-end processing elements of adjacent rows of the array of rows to make a second-end interrow connection that allows operation of the adjacent rows as a double row of processing elements.

[0006] The device may further include a first-end switch to selectively connect first-end processing elements of adjacent rows of the array of rows to make a first-end interrow connection that allows operation of the adjacent rows as a double row of processing elements.

[0007] The first-end switch may selectively connect a first-end processing element of a row to either a first-end processing element of an adjacent row or the controller.

[0008] The first-end switch may connect a row to a previous adjacent row of the array and the second-end switch may connect the row to a next adjacent row of the array.

[0009] Each row of the array may be connected to an adjacent row by a respective second-end switch. Outermost rows of the array may be connected to the controller at

respective first-ends. Each row of the array except the outermost rows may be connected to an adjacent row by a respective first-end switch.

[0010] The respective first-end and second-end switches may be controllable to connect rows of the array into double rows, quadruple rows, sextuple rows, octuple rows, or a combination of such.

[0011] The second-end switch may be configured to preserve a byte order of data that is shifted between the adjacent rows that form the double row.

[0012] According to another aspect of this disclosure, a device includes a two-dimensional array of processing banks. Each processing bank of the array includes a plurality of rows of processing elements operable according to a single instruction, multiple data scheme. The device further includes a bus including two legs. A first column of processing banks is positioned adjacent one of the legs of the bus and each processing bank in the first column connects to the one of the legs of the bus. A second column of processing banks is positioned between the two legs of the bus and each processing bank in the second column connects to both legs of the bus. The bus includes a reversing segment that joins the two legs. The reversing segment is to reverse an ordering of content of a message provided to the bus to preserve an orientation of the content to remain the same relative to the processing elements of the processing banks irrespective of which leg of the bus carries the message.

[0013] The reversing segment may include two buffers arranged to preserve an order of packets of the message when the message is copied from one of the buffers to another of the buffers. Each of the two buffers is connected to a respective one of the legs of the bus to reverse the order of the packets with respect to directions of the legs.

[0014] The device may further include an interface connected to the bus. The interface may be connected to a main processor of a device that contains the two-dimensional array of processing banks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] FIG. 1 is a block diagram of an example computing device that includes banks of processing elements.

[0016] FIG. 2 is a block diagram of an example array of processing elements.

[0017] FIG. 3 is a block diagram of an example array of processing elements with a controller.

[0018] FIG. 4 is a block diagram of an example array of processing elements with a controller and memory.

[0019] FIG. 5 is a schematic diagram of example processing elements and related memory cells.

[0020] FIG. 6 is an equation for an example matrix multiplication carried out by the processing elements and memory cells of FIG. 5.

[0021] FIG. 7A is a schematic diagram of an example state sequence of the processing elements and memory cells of FIG. 5.

[0022] FIG. 7B is a schematic diagram of an example state sequence of the processing elements and memory cells of FIG. 5.

[0023] FIG. 7C is a schematic diagram of an example generalized solution to movement of input vector components among a set of processing elements.

[0024] FIG. 8 is a flowchart of an example method of performing operations using processing elements and memory cells.

[0025] FIG. 9 is a block diagram of an example processing element and related memory cells.

[0026] FIG. 10 is a block diagram of an example of the neighbor processing element interconnect control of FIG. 9.

[0027] FIG. 11 is a block diagram of an example bank of processing elements with an array of processing rows and a switching circuit to couple rows together.

[0028] FIGs. 12A-E are schematic diagrams showing examples of various interrow connections that may be made with reference the example bank of FIG. 11.

[0029] FIGs. 13A-B are block diagrams showing examples of switches connecting second ends of two rows of processing elements.

[0030] FIG. 14 is a block diagram of processing elements that may be selectively paired to operate at increased bit widths.

[0031] FIG. 15 is a block diagram of an example processing element and associated memory arrangement.

[0032] FIG. 16 is a block diagram of an example two-dimensional array of processing banks connected to an interface.

[0033] FIG. 17 is a block diagram of an example of the reversing segment of FIG. 16.

[0034] FIG. 18 is a block diagram of another example of the reversing segment of FIG. 16.

DETAILED DESCRIPTION

[0035] The techniques described herein aim to improve computational memory to handle large numbers of dot-product and neural-network computations with flexible low-precision arithmetic, provide power-efficient communications, and provide local storage and decoding of instructions and coefficients. The parallel processing described herein is suitable for neural networks, particularly where power consumption is a concern, such as

in battery-powered devices, portable computers, smartphones, wearable computers, smart watches, and the like.

[0036] FIG. 1 shows a computing device 100. The computing device 100 includes a plurality of banks 102 of processing elements. The banks 102 may be operated in a cooperative manner to implement a parallel processing scheme, such as a single instruction, multiple data (SIMD) scheme.

[0037] The banks 102 may be arranged in a regular rectangular grid-like pattern, as illustrated. For sake of explanation, relative directions mentioned herein will be referred to as up, down, vertical, left, right, horizontal, and so on. However, it is understood that such directions are approximations, are not based on any particular reference direction, and are not to be considered limiting.

[0038] Any practical number of banks 102 may be used. Limitations in semiconductor fabrication techniques may govern. In some examples, 512 banks 102 are arranged in a 32-by-16 grid.

[0039] A bank 102 may include a plurality of rows 104 of processing elements (PEs) 108 and a controller 106. A bank 102 may include any practical number of PE rows 104. For example, eight rows 104 may be provided for each controller 106. In some examples, all banks 102 may be provided with the same or similar arrangement of rows. In other examples, substantially all banks 102 are substantially identical. In still other examples, a bank 102 may be assigned a special purpose in the computing device and may have a different architecture, which may omit PE rows 104 and/or a controller 106.

[0040] Any practical number of PEs 108 may be provided to a row 104. For example, 256 PEs may be provided to each row 104. Continuing the numerical example above, 256 PEs provided to each of eight rows 104 of 512 banks 102 means the computing device 100 includes about 1.05 million PEs 108, less any losses due to imperfect semiconductor manufacturing yield.

[0041] A PE 108 may be configured to operate at any practical bit size, such as one, two, four, or eight bits. PEs may be operated in pairs to accommodate operations requiring wider bit sizes.

[0042] Instructions and/or data may be communicated to/from the banks 102 via an input/output (I/O) bus 110. The I/O bus 110 may include a plurality of segments.

[0043] A bank 102 may be connected to the I/O bus 110 by a vertical bus 112. Additionally or alternatively, a vertical bus 112 may allow communication among banks 102 in a vertical direction. Such communication may be restricted to immediately vertically adjacent banks 102 or may extend to further banks 102.

[0044] A bank 102 may be connected to a horizontally neighboring bank 102 by a horizontal bus 114 to allow communication among banks 102 in a horizontal direction. Such communication may be restricted to immediately horizontally adjacent banks 102 or may extend to further banks 102.

[0045] Communications through any or all of the busses 110, 112, 114 may include direct memory access (DMA) to memory of the rows 104 of the PEs 108. Additionally or alternatively, such communications may include memory access performed through the processing functionality of the PEs 108.

[0046] The computing device 100 may include a main processor (not shown) to communicate instructions and/or data with the banks 102 via the I/O bus 110, manage operations of the banks 102, and/or provide an I/O interface for a user, network, or other device. The I/O bus 110 may include a Peripheral Component Interconnect Express (PCIe) interface or similar.

[0047] FIG. 2 shows an example row 104 including an array of processing elements 108, which may be physically arranged in a linear pattern (*e.g.*, a physical row). Each PE 108 includes an arithmetic logic unit (ALU) to perform an operation, such as addition, multiplication, and so on.

[0048] The PEs 108 are mutually connected to share or communicate data. For example, interconnections 200 may be provided among the array of PEs 108 to provide direct communication among neighboring PEs 108.

[0049] A PE 108 (*e.g.*, indicated at “n”) is connected to a first neighbor PE 108 (*i.e.*, n+1) that is immediately adjacent the PE 108. Likewise, the PE 108 (n) is further connected to a second neighbor PE 108 (n+2) that is immediately adjacent the first neighbor PE 108 (n+1). A plurality of PEs 108 may be connected to neighboring processing elements in the same relative manner, where n merely indicates an example PE 108 for explanatory purposes. That is, the first neighbor PE 108 (n+1) may be connected to its respective first and second neighbors (n+2 and n+3).

[0050] A given PE 108 (*e.g.*, n+5) may also be connected to an opposite first neighbor PE 108 (n+4) that is immediately adjacent the PE 108 (n+5) on a side opposite the first neighbor PE 108 (n+6). Similarly, the PE 108 (n+5) may further be connected to an opposite second neighbor PE 108 (n+3) that is immediately adjacent the opposite first neighbor PE 108 (n+4).

[0051] Further, a PE 108 may be connected to a fourth neighbor PE 108 that is immediately adjacent a third neighbor PE 108 that is immediately adjacent the second neighbor PE 108. For example, the PE 108 designated at n may be connected to the PE designated at n+4. A connection of the PE 108 (n) to its third neighbor PE 108 (n+3) may be omitted. The fourth-neighbor connection may also be provided in the opposite direction, so that the PE 108 (n) connects to its fourth neighbor PE 108 at n-4 (not shown).

[0052] Still further, a PE 108 may be connected to a sixth neighbor PE 108 that is immediately adjacent a fifth neighbor PE 108 that is immediately adjacent the fourth neighbor PE 108. For example, the PE 108 designated at n may be connected to the PE designated at n+6. A connection of the PE 108 (n) to its fifth neighbor PE 108 (n+5) may

be omitted. The sixth-neighbor connection may also be provided in the opposite direction, so that the PE 108 (n) connects to its sixth neighbor PE 108 at n-6 (not shown).

[0053] Again, a plurality of PEs 108 may be connected to neighboring processing elements in the above relative manner. The designation of a PE 108 as n may be considered arbitrary for non-endmost PEs 108. PEs 108 at the ends of the array may omit certain connections by virtue of the array terminating. In the example of each PE 108 being connected to its first, second, fourth, and sixth neighbor PEs 108 in both directions, the six endmost PEs 108 have differing connections.

[0054] With reference to FIG. 3, endmost PEs 108 at one end of a row 104 may have connections 300 to a controller 106. Further, endmost PEs 108 at the opposite end of the row 104 may have a reduced number of connections 302. Additionally or alternatively, end-most PEs 108 of one bank 102 may connect in the same relative manner through the controller 106 and to PEs 108 of an adjacent bank 102. That is, the controller 106 may be connected between two rows 104 of PEs 108 in adjacent banks 102, where the two rows 104 of PEs 108 are connected in the same manner as shown in FIG. 2

[0055] With reference to FIG. 4, a row 104 of PEs 108 may include memory 400 to store data for the row 104. A PE 108 may have a dedicated space in the memory 400. For example, each PE 108 may be connected to a different range of memory cells 402. Any practical number of memory cells 402 may be used. In one example, 144 memory cells 402 are provided to each PE 108. Note that in FIG. 4 the interconnections 200 among the PEs 108 and with the controller 106 are shown schematically for sake of explanation.

[0056] The controller 106 may control the array of PEs 108 to perform a SIMD operation with data in the memory 400. For example, the controller 106 may trigger the PEs 108 to simultaneously add two numbers stored in respective cells 402.

[0057] The controller 106 may communicate data to and from the memory 400 through the PEs 108. For example, the controller 106 may load data into the memory 400 by directly loading data into connected PEs 108 and controlling PEs 108 to shift the data to

PEs 108 further in the array. PEs 108 may load such data into their respective memory cells 402. For example, data destined for rightmost PEs 108 may first be loaded into leftmost PEs and then communicated rightwards by interconnections 200 before being stored in rightmost memory cells 402. Other methods of I/O with the memory, such as direct memory access by the controller 106, are also contemplated. The memory cells 402 of different PEs 108 may have the same addresses, so that address decoding may be avoided to the extent possible.

[0058] Data stored in memory cells 402 may be any suitable data, such as operands, operators, coefficients, vector components, mask data, selection data, and similar. Mask data may be used to select portions of a vector. Selection data may be used to make/break connections among neighboring PEs 108.

[0059] Further, the controller 106 may perform a rearrangement of data within the array of PEs 108 by controlling communication of data through the interconnections 200 among the array of PEs 108. A rearrangement of data may include a rotation or cycling that reduces or minimizes a number of memory accesses while increasing or maximizing operational throughput. Other examples of rearrangements of data include reversing, interleaving, and duplicating.

[0060] In other examples, a set of interconnections 200 may be provided to connect PEs 108 in up-down (column-based) connections, so that information may be shared directly between PEs 108 that are in adjacent rows. In this description, interconnections 200 and related components that are discussed with regard to left-right (row-based) connections among PEs apply in principle to up-down (column-based) connections among PEs.

[0061] FIG. 5 shows an array of PEs 108 and related memory cells 402. Each PE 108 may include local registers 500, 502 to hold data undergoing an operation. Memory cells 402 may also hold data contributing to the operation. For example, the PEs 108 may carry out a matrix multiplication, as shown in FIG. 6.

[0062] A matrix multiplication may be a generalized matrix-vector multiply (GEMV). A matrix multiplication may use a coefficient matrix and an input vector to obtain a resultant vector. In this example, the coefficient matrix is a four-by-four matrix and the vectors are of length four. In other examples, matrices and vectors of any practical size may be used. In other examples, a matrix multiplication may be a generalized matrix-matrix multiply (GEMM).

[0063] As matrix multiplication involves sums of products, the PEs 108 may additively accumulate resultant vector components d_0 to d_3 in respective registers 500, while input vector components a_0 to a_3 are multiplied by respective coefficients c_{00} to c_{33} . That is, one PE 108 may accumulate a resultant vector component d_0 , a neighbor PE 108 may accumulate another resultant vector component d_1 , and so on. Resultant vector components d_0 to d_3 may be considered dot products. Generally, a GEMV may be considered a collection of dot products of a vector with a set of vectors represented by the rows of a matrix.

[0064] To facilitate matrix multiplication, the contents of registers 500 and/or registers 502 may be rearranged among the PEs 108. A rearrangement of resultant vector components d_0 to d_3 and/or input vector components a_0 to a_3 may use the direct interconnections among neighbor PEs 108, as discussed above. In this example, resultant vector components d_0 to d_3 remain fixed and input vector components a_0 to a_3 are moved. Further, coefficients c_{00} to c_{33} may be loaded into memory cells to optimize memory accesses.

[0065] In the example illustrated in FIG. 5, the input vector components a_0 to a_3 are loaded into a sequence of PEs 108 that are to accumulate resultant vector components d_0 to d_3 in the same sequence. The relevant coefficients c_{00} , c_{11} , c_{22} , c_{33} are accessed and multiplied by the respective input vector components a_0 to a_3 . That is, a_0 and c_{00} are multiplied and then accumulated as d_0 , a_1 and c_{11} are multiplied and then accumulated as d_1 , and so on.

[0066] The input vector components a_0 to a_3 are then rearranged, as shown in the PE state sequence of FIG. 7A, so that a remaining contribution of each input vector components a_0 to a_3 to a respective resultant vector components d_0 to d_3 may be accumulated. In this example, input vector components a_0 to a_2 are moved one PE 108 to the right and input vector components a_3 is moved three PEs 108 to the left. With reference to the first and second neighbor connections shown in FIG. 2, this rearrangement of input vector components a_0 to a_3 may be accomplished by swapping a_0 with a_1 and simultaneously swapping a_2 with a_3 , using first neighbor connections, and then by swapping a_1 with a_3 using second neighbor connections. The result is that a next arrangement of input vector components a_3, a_0, a_1, a_2 at the PEs 108 is achieved, where each input vector component is located at a PE 108 that it has not yet occupied during the present matrix multiplication.

[0067] Appropriate coefficients $c_{03}, c_{10}, c_{21}, c_{32}$ in memory cells 402 are then accessed and multiplied by the respective input vector components a_3, a_0, a_1, a_2 . That is, a_3 and c_{03} are multiplied and then accumulated as d_0 , a_0 and c_{10} are multiplied and then accumulated as d_1 , and so on.

[0068] The input vector components a_0 to a_3 are then rearranged twice more, with multiplying accumulation being performed with the input vector components and appropriate coefficients at each new arrangement. At the conclusion of four sets of multiplying accumulation and three intervening rearrangements, the accumulated resultant vector components d_0 to d_3 represent the final result of the matrix multiplication.

[0069] Rearrangement of the input vector components a_0 to a_3 allows each input vector component to be used to the extent needed when it is located at a particular PE 108. This is different from traditional matrix multiplication where each resultant vector component is computed to finality prior to moving to the next. The present technique simultaneously accumulates all resultant vector components using sequenced arrangements of input vector components.

[0070] Further, such rearrangements of data at the PEs 108 using the PE neighbor interconnections (FIG. 2) may be optimized to reduce or minimize processing cost. The example given above of two simultaneous first neighbor swaps followed by a second neighbor swap is merely one example. Additional examples are contemplated for matrices and vectors of various dimensions.

[0071] Further, the arrangements of coefficients c_{00} to c_{33} in the memory cells 402 may be predetermined, so that each PE 108 may access the next coefficient needed without requiring coefficients to be moved among memory cells 402. The coefficients c_{00} to c_{33} may be arranged in the memory cells 402 in a diagonalized manner, such that a first row of coefficients is used for a first arrangement of input vector components, a second row of coefficients is used for a second arrangement of input vector components, and so on. Hence, the respective memory addresses referenced by the PEs 108 after a rearrangement of input vector components may be incremented or decremented identically. For example, with a first arrangement of input vector components, each PE 108 may reference its respective memory cell at address 0 for the appropriate coefficient. Likewise, with a second arrangement of input vector components, each PE 108 may reference its respective memory cell at address 1 for the appropriate coefficient, and so on.

[0072] FIG. 7B shows another example sequence. Four states of a set of PEs 108 are shown with four sets of selected coefficients. Input vector components a_0 to a_3 are rotated so that each component a_0 to a_3 is used exactly once to contribute to the accumulation at each resultant vector component d_0 to d_3 . The coefficients c_{00} to c_{33} are arranged so that the appropriate coefficient c_{00} to c_{33} is selected for each combination of input vector component a_0 to a_3 and resultant vector component d_0 to d_3 . In this example, the input vector components a_0 to a_3 are subject to the same rearrangement three times to complete a full rotation. Specifically, the input vector component of an n^{th} PE 108 is moved right to the second neighbor PE 108 (*i.e.*, $n+2$), the input vector component of the PE 108 $n+1$ is moved left (opposite) to its first neighbor PE 108 (*i.e.*, n) in that direction, the input

vector component of the PE 108 $n+2$ is moved right to the first neighbor PE 108 (*i.e.*, $n+3$), and the input vector component of the PE 108 $n+3$ is moved left to the second neighbor PE 108 (*i.e.*, $n+1$).

[0073] FIG. 7C shows a generalized solution, which is implicit from the examples discussed herein, to movement of input vector components among a set of PEs 108. As shown by the row-like arrangement 700 of input vector components a_0 to a_i , which may be held by a row 104 of PEs 108, rotating information may require many short paths 702, between adjacent components a_0 to a_i , and a long path 704 between end-most components a_i and a_0 . The short paths are not a concern. However, the long path 704 may increase latency and consume additional electrical power because charging and discharging a conductive trace takes time and is not lossless. The longer the trace, the greater the time/loss. The efficiency of a row 104 of PEs 108 is limited by its long path 704, in that power is lost and other PEs 108 may need to wait while data is communicated over the long path 704.

[0074] As shown at 710, a circular arrangement of PEs 108 may avoid a long path 704. All paths 712 may be segments of a circle and may be made the same length. A circular arrangement 710 of PEs 108 may be considered an ideal case. However, a circular arrangement 710 is impractical for manufacturing purposes.

[0075] Accordingly, the circular arrangement 720 may be rotated slightly and flattened (or squashed), while preserving the connections afforded by circular segment paths 712 and the relative horizontal (X) positions of the PEs, to provide for an efficient arrangement 720, in which paths 722, 724 connect adjacent PEs or skip one intermediate PE. As such, PEs 108 may be connected by a set of first-neighbor paths 722 (*e.g.*, two end-arriving paths) and a set of second neighbor paths 724 (*e.g.*, four intermediate and two end-leaving paths) that are analogous to circular segment paths 712 of a circular arrangement 710. The paths 722, 724 have much lower variance than the short and long paths 702, 704, so power may be saved and latency reduced. Hence, the arrangement 720

represents a readily manufacturable implementation of an ideal circular arrangement of PEs 108.

[0076] FIG. 8 shows a method 900 that generalizes the above example. The method 900 may be performed with the computing device 100 or a similar device.

[0077] At block 902, operands (*e.g.*, matrix coefficients) are loaded into PE memory cells. The arrangement of operands may be predetermined with the constraint that moving operands is to be avoided where practical. An operand may be duplicated at several cells to avoid moving an operand between such cells.

[0078] At block 904, operands (*e.g.*, input vector components) are loaded into PE registers. The operands to be loaded into PE registers may be distinguished from the operands to be loaded into PE memory cells, in that there may be fewer PE registers than PE memory cells. Hence, in the example of a matrix multiplication, it may be more efficient to load the smaller matrix/vector to the into PE registers and load the larger matrix into the PE memory cells. In other applications, other preferences may apply.

[0079] At block 906, a set of memory cells may be selected for use in an operation. The set may be a row of memory cells. For example, a subset of coefficients of a matrix to be multiplied may be selected, one coefficient per PE.

[0080] At block 908, the same operation is performed by the PEs on the contents of the selected memory cells and respective PE registers. The operation may be performed substantially simultaneously with all relevant PEs. All relevant PEs may be all PEs of a device or a subset of PEs assigned to perform the operation. An example operation is a multiplication (*e.g.*, multiplying PE register content with memory cell content) and accumulation (*e.g.*, accumulating the resulting product with a running total from a previous operation).

[0081] Then, if a subsequent operation is to be performed, via block 910, operands in the PE registers may be rearranged, at block 912, to obtain a next arrangement. A next set of

memory cells is then selected at block 906, and a next operation is performed at block 908. For example, a sequence of memory cells may be selected during each cycle and operands in the PE registers may be rearranged to correspond to the sequence of memory cells, so as to perform a matrix multiplication. In other examples, other operations may be performed.

[0082] Hence, a sequence or cycle or operations may be performed on the content of selected memory cells using the content of PE registers that may be rearranged as needed. The method 900 ends after the last operation, via block 910.

[0083] The method 900 may be varied. In various examples, selection of the memory cells need not be made by selection of a contiguous row. Arranging data in the memory cells according to rows may simplify the selection process. For example, a single PE-relative memory address may be referenced (*e.g.*, all PEs refer to their local memory cell with the same given address). That said, it is not strictly necessary to arrange the data in rows. In addition or alternatively, a new set of memory cells need not be selected for each operation. The same set may be used in two or more consecutive cycles. Further, overlapping sets may be used, in that a memory cell used in a former operation may be deselected and a previously unselected memory cell may be selected for a next operation, while another memory cell may remain selected for both operations. In addition or alternatively, the operands in the PE registers need not be rearranged each cycle. Operands may remain in the same arrangement for two or more consecutive cycles. Further, operand rearrangement does not require each operand to change location, in that a given operand may be moved while another operand may remain in place.

[0084] FIG. 9 shows an example PE 108 schematically. The PE 108 includes an ALU 1000, registers 1002, a memory interface 1004, and neighbor PE interconnect control 1006.

[0085] The ALU 1000 performs the operational function of the PE. The ALU 1000 may include an adder, multiplier, accumulator, or similar. In various examples, the ALU 1000

is a multiplying accumulator. The ALU 1000 may be connected to the memory interface 1004, directly or indirectly, through the registers 1002 to share information with the memory cells 402. In this example, the ALU 1000 is connected to the memory interface 1004 through the registers 1002 and a bus interface 1008.

[0086] The registers 1002 are connected to the ALU 1000 and store data used by the PE 108. The registers 1002 may store operands, results, or other data related to operation of the ALU 1000, where such data may be obtained from or provided to the memory cells 402 or other PEs 108 via the neighbor PE interconnect control 1006.

[0087] The memory interface 1004 is connected to the memory cells 402 and allows for reading/writing at the memory cells 402 to communicate data with the registers 1002, ALU 1000, and/or other components of the PE 108.

[0088] The neighbor PE interconnect control 1006 connects to the registers 1002 and controls communication of data between the registers 1002 and like registers of neighboring PEs 108, for example via interconnections 200 (FIG. 2), and/or between a controller (see 106 in FIG. 3). The neighbor PE interconnect control 1006 may include a logic/switch array to selectively communicate the registers 1002 to the registers 1002 of neighboring PEs 108, such as first, second, fourth, or sixth neighbor PEs. The neighbor PE interconnect control 1006 may designate a single neighbor PE 108 from which to obtain data. That is, the interconnections 200 may be restricted so that a PE 108 only at most listens to one selected neighbor PE 108. The neighbor PE interconnect control 1006 may connect PEs 108 that neighbor each other in the same row. Additionally or alternatively, a neighbor PE interconnect control 1006 may be provided to connect PEs 108 that neighbor each other in the same column.

[0089] The PE may further include a bus interface 1008 to connect the PE 108 to a bus 1010, such as a direct memory access bus. The bus interface 1008 may be positioned between the memory interface 1004 and registers 1002 and may selectively communicate data between the memory interface 1004 and either a component outside the PE 108

connected to the bus 1010 (*e.g.*, a main processor via direct memory access) or the registers 1002. The bus interface 1008 may control whether the memory 402 is connected to the registers 1002 or the bus 1010.

[0090] The PE may further include a shifter circuit 1012 connected to the ALU 1000 and a wide-add bus 1014 to perform shifts to facilitate performing operations in conjunction with one or more neighbor PEs 108.

[0091] FIG. 10 shows an example of the neighbor PE interconnect control 1006. The neighbor PE interconnect control 1006 includes a multiplexer 1100 or similar switch/logic array and a listen register 1102.

[0092] The multiplexer 1100 selectively communicates one interconnection 200 to a neighbor PE 108 to a register 1002 used for operations of the PE 108 to which the neighbor PE interconnect control 1006 belongs. Hence, a PE 108 listens to one neighbor PE 108.

[0093] The listen register 1102 controls the output of the multiplexer 1100, that is, the listen register 1102 selects a neighbor PE 108 as source of input to the PE 108. The listen register 1102 may be set by an external component, such as a controller 106 (FIG. 3), or by the PE 108 itself.

[0094] FIG. 11 shows a bank 1100 that may be used in a computing device, such as the computing device 100 of FIG. 1. Features and aspects of the other devices and methods described herein may be used with the bank 1100. Like reference numerals and like terminology denote like components, and redundant description will not be repeated here.

[0095] The bank 1100 includes an array of processing rows 104 and a switching circuit.

[0096] The array of processing rows 104 may include a first-end row 1102 and a second-end row 1104 positioned at opposite ends of the array of processing rows 104. The processing rows 104 may be arranged in a linear or stacked arrangement.

[0097] Each processing row 104 includes an array PEs 108, which may be arranged in a row or other linear pattern. Each processing row 104 includes a first-end PE 1106 and a second-end PE 1108 that are positioned at opposite ends of the processing row 104. The first-end PE 1106 of each row 104 may be positioned adjacent a controller 106. The second-end PE 1108 may be the PE 108 that is furthest from the controller 106.

[0098] The switching circuit selectively connects first-end PEs 1106 and second-end PEs 1108 of different processing rows 104. The switching circuit may include switches 1110, 1112, 1114. Each row 104 may be connected to a switch 1110 at its second-end PE 1108, so that the row 104 may be selectively connected to an adjacent row 104. That is, the second-end PEs 1108 of pairs of adjacent rows 104 may be selectively connectable via a switch 1110. Pairs of adjacent rows 104 may be selectively connectable either to each other or to the controller 106 via switches 1112, 1114. The adjacent pairs of rows 104 connectable by a switch 1110 may be offset by one row from the adjacent pairs of rows 104 connectable by switches 1112, 1114.

[0099] Switches 1110, 1112, 1114 may be provided as switches on interconnections 200 of adjacent PEs 108, as shown in FIG. 2. Byte order may be preserved when moving data between rows 104 by proper routing of interconnections. FIG. 13A shows an example of a switch 1110 connecting second ends of two rows 104. FIG. 13B shows another example of a switch 1110 connecting second ends of two rows 104 and preserving byte order. The switch 1110 includes a plurality of individual switches, one for each interconnection among PEs 108 of the rows 104. Switches 1112, 1114 may be designed using the same principle.

[0100] Given an array with an arbitrary number of processing rows 104, a first-end PE 1106 of an N processing row 104 may be selectively connectable to first-end PE 1106 of an N+1 processing row 104 by switches 1112, 1114. When not connected to each other, the N and N+1 rows 104 may be connected to the controller 106. A second-end PE 1108 of the N processing row 104 may be selectively connectable to second-end PE 1108 of an N-1 processing row 104 via a switch 1110. When not connected to each other, the N and

N-1 processing rows 104 may have their second ends unconnected to a row 104 in the device 1100.

[0101] At the extents of the array of rows 104, the first-end row 1102 and the second-end row 1104 may be non-selectively connected to the controller 106. That is, the first-end row 1102 and the second-end row 1104 may be permanently connected to the controller 106, in that the first-end row 1102 and the second-end row 1104 are not selectively connectable to another row 104.

[0102] In various examples, only the first-end row 1102 and the second-end row 1104 lack selective connections to the controller 106. Each processing row 104 except the first-end row 1102 and the second-end row 1104 is selectively connectable to either the first-end PE 1106 of an adjacent row 104 or the controller 106.

[0103] The switching circuit, as may be embodied by the switches 1110, 1112, 1114, allows different mutual connections of the rows 104. Rows 104 may be combined to form super-rows of double, quadruple, or other factor of individual row size. This may allow for the processing of longer segments of information. For example, if each row 104 contains 1024 8-bit PEs 108, then the rows 104 may perform parallel operations on 1 kB of data, two rows 104 may be connected to perform parallel operations on 2 kB of data, four rows 104 may be connected to perform parallel operations on 4 kB of data, and so on.

[0104] The switching circuit may be controlled by the controller 106. The second-end switches 1110 may be controlled via a common signal line 1116 that provides a second-end control signal from the controller 106. Pairs of adjacent rows 104 may therefore all be controlled together to have second-end PEs 1108 connected or disconnected. Each set of first-end switches 1112, 1114 may be controlled by a first-end signal line 1118 that provides a first-end control signal from the controller 106. As such, pairs of adjacent rows 104 may be individually controlled to have their first-end PEs 1106 either mutually connected or connected to the controller 106. Accordingly, the common signal line 1116

may carry an enable or disable signal depending on whether or not the connecting of rows 104 in general is to be enabled or disabled. Further, each individual first-end signal lines 1118 may carry an enable or disable signal to facilitate the specific super-rows desired.

[0105] FIGs. 12A-E show examples of various interrow connections that may be made with reference the device 1100 of FIG. 11. Eight contiguous rows 104 are shown as an example and it should be appreciated that more or fewer may be provided.

[0106] In FIG. 12A, no interrow connections are enabled, and the result is eight individual rows of processing elements. A connection 1200 of a row 104 to the controller 106 may be provided through a switch 1112 (FIG. 11). A connection 1202 of an adjacent row 104 to the controller 106 may be provided through a switch 1114 (FIG. 11).

[0107] In FIG. 12B, interrow connections provide four double rows. Connections 1204 at second-ends of adjacent rows may be provided through switches 1110 (FIG. 11). Meanwhile, all switches 1112, 1114 (FIG. 11) at the controller ends of the rows 104 may provide connections 1206 to the controller 106.

[0108] In FIG. 12C, interrow connections provide two quadruple rows of processing elements. Pairs of switches 1112, 1114 (FIG. 11) at the controller ends of the rows 104 may, in an alternating manner, provide row-to-row connections 1208 and connections 1206 to the controller 106.

[0109] In FIG. 12D, interrow connections provide one octuple row. Switches 1112, 1114 (FIG. 11) at the controller ends of the rows 104 may be used to provide row-to-row connections 1208, except for end-most rows 104 that may be permanently or non-switchably connected to the controller 106.

[0110] In FIG. 12E, interrow connections provide combined rows of double and six-times size of a single row 106. Various other combinations of combined rows are contemplated.

[0111] FIG. 14 shows example, PEs 1402, 1412 that may be selectively paired to operate at increased bit widths. For example, each PE 1402, 1412 may individually operate on 4 bits and pairing them may allow for 8-bit operations to be performed.

[0112] Each PE 1402, 1412 may include a respective accumulator 1404, 1414 and respective registers 1408, 1418. The accumulators 1404, 1414 may be configured to have four times the bit size of a single PE 1402, 1412 (*e.g.*, 16 bits) to allow for a large number (*e.g.*, 256) of multiplication and accumulation operations on 4-bit values without overflow.

[0113] In an example operation, a 4-bit PE 1412 performs a sequence of multiplications and accumulations 1430. The PE 1412 may multiply two 4-bit numbers to obtain an 8-bit number and then perform addition of 8-bit numbers. This may obtain an intermediate result of 16-bits at the accumulator 1414. The PE 1412 may store the 16-bit intermediate result by storing 8 bits in its registers 1418 and storing the remaining 8 bits in the registers 1408 of the adjacent PE 1404, as indicated at 1432 and 1434 respectively. That is, the 16-bit intermediate value may be split into two separate 8-bit components for storage in the separate sets of registers 1418, 1408 of the neighboring PEs 1414, 1402. The registers 1418 associated with the PE 1412 may be used to store the least-significant bits, while the neighboring registers 1408 may be used to store the most-significant bits.

[0114] Subsequently, the PE 1412 is to add a current value, which may be of 16 bits, with the 16-bit value stored in registers 1408, 1418. The accumulator 1414 of the PE 1412 obtains 1436 the 8-bit component from the associated registers 1418 and also obtains 1438 the other 8-bit component from the registers 1408 of the neighboring PE 1404. The two 8-bit components are concatenated into a 16-bit value that may then be added to a current 16-bit value, already present in the accumulator 1414. By linking 1440 the accumulators 1404, 1414, the addition may be carried over from the accumulator 1414 to the neighboring accumulator 1404 to allow for a 32-bit result, which ultimately may be scaled or truncated as needed. As such, the PEs 1402, 1412 may cooperate on data of bit sizes larger than their individual sizes.

[0115] For the paired, wide addition operation, such as described above, the accumulators 1404, 1414 may be temporarily linked 1440 such that they act as a single double-width accumulator (*e.g.*, of 32 bits). The linkage 1440 allows carry over from the most-significant bit (MSB) of the accumulator 1414 to the least-significant bit (LSB) of the neighboring accumulator 1404. If the value is signed, then a sign bit may be carried over from the accumulator 1414 to the neighboring accumulator 1404.

[0116] The linkage 1440 is selectable and may be deactivated to operate the PEs 1402, 1412 independently and without the capacity to handle wider operations.

[0117] FIG. 15 shows a diagram of an example PE 1500 and its associated memory 1502. The memory 1502 may be arranged into blocks, so that the PE 1500 may access one block at the same time that an external process, such as direct memory access, accesses another block. Such simultaneous access may allow for faster overall performance of a row, bank, or other device containing the PE, as the PE and external process can perform operations with different blocks of memory at the same time and there will be fewer occasions of the PE or external process having to wait for the other to complete its memory access. In general, PE access to memory is faster than outside access, so it is expected that the PE 1500 will be able to perform N memory operations to one block per one outside operation to the other block.

[0118] The memory 1502 includes two blocks 1504, 1506, each containing an array of memory cells 1508. Each block 1504, 1506 may also include a local I/O circuit 1510 to handle reads/writes to the cells of the block 1504, 1506. In other examples, more than two blocks may be used.

[0119] The memory 1502 further includes a global I/O circuit 1512 to coordinate access by the PE and external process to the blocks 1504, 1506.

[0120] The PE 1500 may include memory access circuits 1520-1526, such as a most-significant nibble (MSN) read circuit 1520, a least-significant nibble (LSN) read circuit

1522, an MSN write circuit 1524, and an LSN write circuit 1526. The memory access circuits 1520-1526 are connected to the global I/O circuit 1512 of the memory 1502.

[0121] The memory address schema of the blocks 1504, 1506 of memory 1502 may be configured to reduce latency. In this example, block 1504 contains cells 1508 with even addresses and the block 1506 contains cells 1508 with odd addresses. As such, when the PE 1500 is to write to a series of addresses, the global I/O circuit 1512 connects the PE 1500 in an alternating fashion to the blocks 1504, 1506. That is, the PE 1500 switches between accessing the blocks 1504, 1506 for a sequence of memory addresses. This reduces the chance that the PE 1500 will have to wait for a typically slower external memory access. Timing between block access can overlap. For example, one block can still be finishing latching data into an external buffer while the other block is concurrently providing data to the PE 1500.

[0122] FIG. 16 shows an example two-dimensional array 1600 of processing banks 102 connected to an interface 1602 via I/O busses 1604. The array 1600 may be grid-like with rows and columns of banks 102. Rows need not have the same number of banks 102, and columns need not have the same number of banks 102.

[0123] The interface 1602 may connect the I/O busses 1604 to a main processor, such as a CPU of a device that contains the array 1600. The interface 1602 may be a PCIe interface.

[0124] The interface 1602 and buses 1604 may be configured to communicate data messages 1606 with the banks 102. The interface 1602 may pump messages through the busses 1604 with messages becoming accessible to banks 102 via bus connections 1608. A bank 102 may read/write data from/to a message 1606 via a bus connection 1608.

[0125] Each bus 1604 includes two legs 1610, 1612. Each leg 1610, 1612 may run between two adjacent columns of banks 102. Depending on its column, a given bank 102 may have bus connections 1608 to both legs 1610, 1612 of the same bus 1604 or may have bus connections 1608 to opposite legs 1610, 1612 of adjacent busses. In this

example, even columns (*e.g.*, 0th, 2nd, 4th) are connected to the legs 1610, 1612 of the same bus 1604 and odd columns (*e.g.*, 1st, 3rd) are connected to different legs 1610, 1612 of adjacent busses 1604.

[0126] In each bus 1604, one end of each leg 1610, 1612 is connected to the interface 1602, and the opposite end of each leg 1610, 1612 is connected to a reversing segment 1620. Further, concerning the direction of movement of messages on the bus 1604, one leg 1610 may be designated as outgoing from the interface 1602 and the other leg 1612 may be designated as incoming to the interface 1602. As such, a message 1606 put onto the bus 1604 by the interface 1602 may be pumped along the leg 1610, through the reversing segment 1620, and back towards the interface 1602 along the other leg 1612.

[0127] The reversing segment 1620 reverses an ordering of content for each message 1606, such that the orientation of the content of each message 1606 remains the same relative to the PEs of the banks 102, regardless of which side of the bank 102 the message 1606 is on. This is shown schematically as message packets “A,” “B,” and “C,” which are discrete elements of content of a message 1606. As can be seen, the orientation of the packets of the message 1606 whether on the leg 1610 or the leg 1612 is the same due to the reversing segment 1620. This is shown in detail in FIG. 17. Without the reversing segment, *i.e.*, with a simple loop bus, the orientation of the message 1606 on the return leg 1612 would be opposite.

[0128] As shown in FIG. 17, the reversing segment 1620 provides that the packets 1702 “A,” “B,” and “C,” of a message 1606 have the same orientation relative to the PEs 108 and respective memory cells 402, as arranged in rows 104 of various banks 102, irrespective of the position of the bank 102 relative to bus legs 1610, 1612. That is, the grid like arrangement of PEs 108 and respective memory cells 402 may be made the same for each bank 102 without needing to accommodate the flipping of message content. Rather, message content is flipped once at the appropriate time via the reversing segment 1620.

[0129] A message 1606 may include any number of packets 1702. Each packet 1702 may have a length that corresponds to a particular number of PEs 108 and respective memory cells 402. For example, a packet 1702 may be 128 bits wide which may correspond to 16 PEs 108 that each have 8-bit wide memory cells 402. A row 104 of 256 PEs 108 therefore aligns with 16 packets 1702. A message 1606 may therefore be formed of 16 packets 1702 with one header packet that includes source/destination information, and which may be read by the controller 106. The packets 1702 are reordered by the reversing segment 1620. However, their content is not reordered. For example, the 16 8-bit elements of a 128-bit packet “A” remain in the same order as the packet “A” is moved from first to last of a stream of packets “A,” “B,” and “C.”

[0130] FIG. 18 shows an example reversing segment 1620. The reversing segment 1620 includes two buffers 1802, 1804. A message 1606 (with packets shown as H for header packet and 0-15 for data packets) is clocked into the first buffer 1802, packet-wise in the direction of the bus leg 1610, until the entire message 1606 is present in the buffer 1802. The content of the buffer 1802 is then copied to the second buffer 1804, preserving the order of the packets with respect to the buffers 1802, 1804 but reversing the order of the packets with respect to the directions of the legs 1610, 1612. The copy of the message 1806 in the second buffer 1804 is then clocked out, packet-wise in the direction of the bus leg 1612, of the reversing segment 1620. It should be noted that the header packet moves to the end of the message 1806 with respect to the direction of the bus leg 1612, and this accommodates the same relative positions of the controllers 106 with respect to the PE rows 104 (see FIG. 17).

[0131] As should be apparent from the above discussion, the techniques discussed herein are suitable for low-power neural-network computations and applications. Further, the techniques are capable of handling a large number of computations with flexibility and configurability.

[0132] It should be recognized that features and aspects of the various examples provided above can be combined into further examples that also fall within the scope of the present

disclosure. In addition, the figures are not to scale and may have size and shape exaggerated for illustrative purposes.

CLAIMS

1. A device comprising:

an array of rows of processing elements, each row including a plurality of connected processing elements, each row terminating at a first-end processing element and a second-end processing element opposite the first-end processing element;

a controller connected to a first-end processing element of a row of the array of rows to provide data and instructions to the row to operate the processing elements of the row according to a single instruction, multiple data scheme;

a second-end switch to selectively connect second-end processing elements of adjacent rows of the array of rows to make a second-end interrow connection that allows operation of the adjacent rows as a double row of processing elements.

2. The device of claim 1, further comprising a first-end switch to selectively connect first-end processing elements of adjacent rows of the array of rows to make a first-end interrow connection that allows operation of the adjacent rows as a double row of processing elements.

3. The device of claim 2, wherein the first-end switch selectively connects a first-end processing element of a row to either a first-end processing element of an adjacent row or the controller.

4. The device of claim 2, wherein the first-end switch connects a row to a previous adjacent row of the array and wherein the second-end switch connects the row to a next adjacent row of the array.

5. The device of claim 2, wherein each row of the array is connected to an adjacent row by a respective second-end switch, wherein outermost rows of the array are connected to the controller at respective first-ends, and wherein each row of the array except the outermost rows are connected to an adjacent row by a respective first-end switch.

6. The device of claim 5, wherein the respective first-end and second-end switches are controllable to connect rows of the array into double rows, quadruple rows, sextuple rows, octuple rows, or a combination of such.

7. The device of claim 1, wherein the second-end switch is configured to preserve a byte order of data that is shifted between the adjacent rows that form the double row.

8. A device comprising:

a two-dimensional array of processing banks, each processing bank of the array including a plurality of rows of processing elements operable according to a single instruction, multiple data scheme;

a bus including two legs, wherein a first column of processing banks is positioned adjacent one of the legs of the bus and each processing bank in the first column connects to the one of the legs of the bus, and wherein a second column of processing banks is positioned between the two legs of the bus and each processing bank in the second column connects to both legs of the bus; and

wherein the bus includes a reversing segment that joins the two legs, wherein the reversing segment is to reverse an ordering of content of a message provided to the bus to preserve an orientation of the content to remain the same relative to the processing elements of the processing banks irrespective of which leg of the bus carries the message.

9. The device of claim 8, wherein the reversing segment includes two buffers arranged to preserve an order of packets of the message when the message is copied from one of the buffers to another of the buffers, wherein each of the two buffers is connected to a respective one of the legs of the bus to reverse the order of the packets with respect to directions of the legs.

10. The device of claim 10, further comprising an interface connected to the bus, wherein the interface is connected to a main processor of a device that contains the two-dimensional array of processing banks.

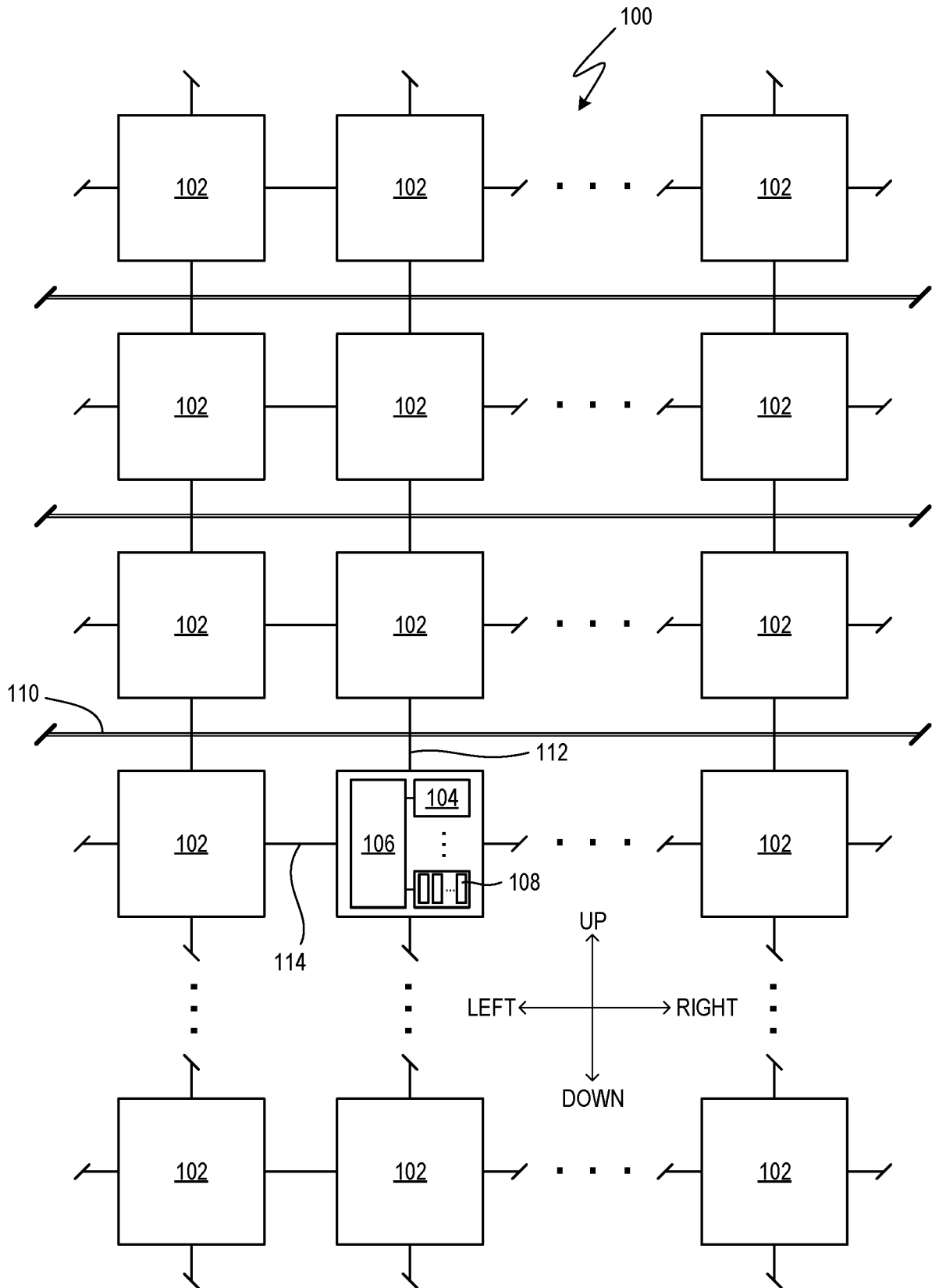
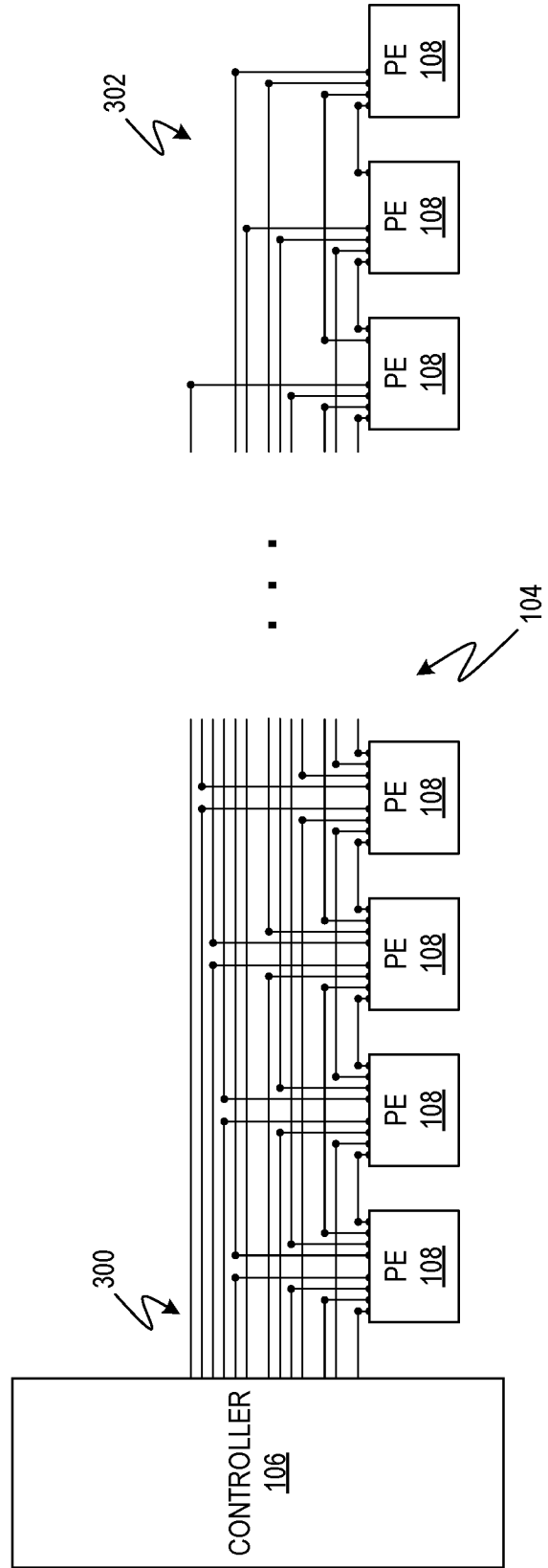
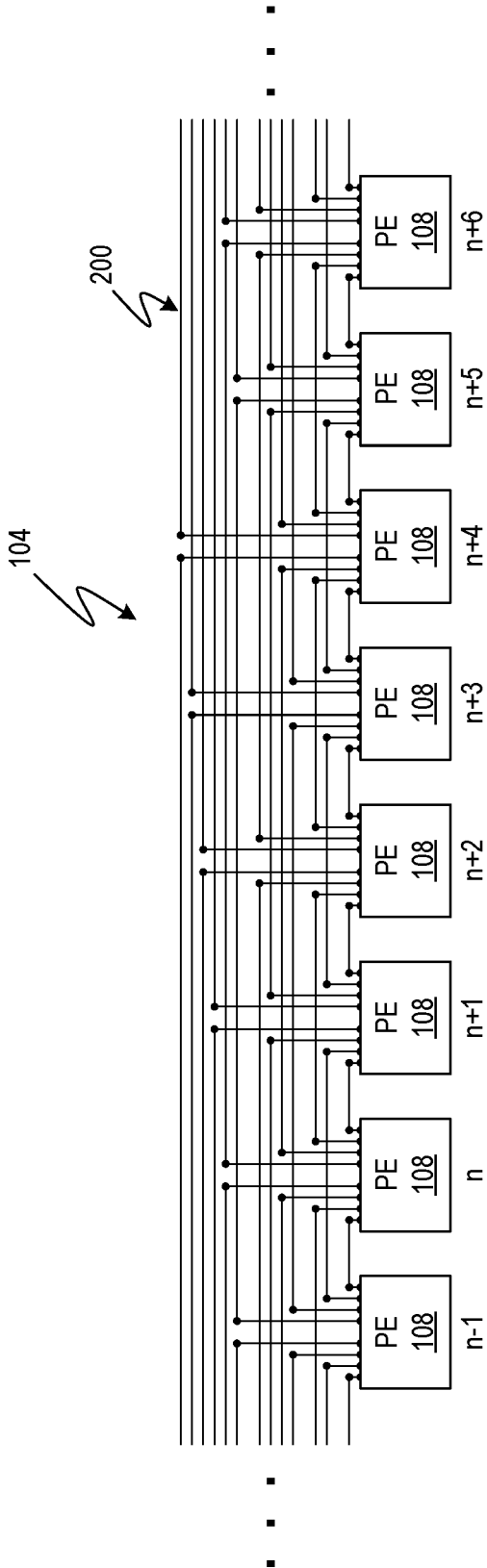


FIG. 1



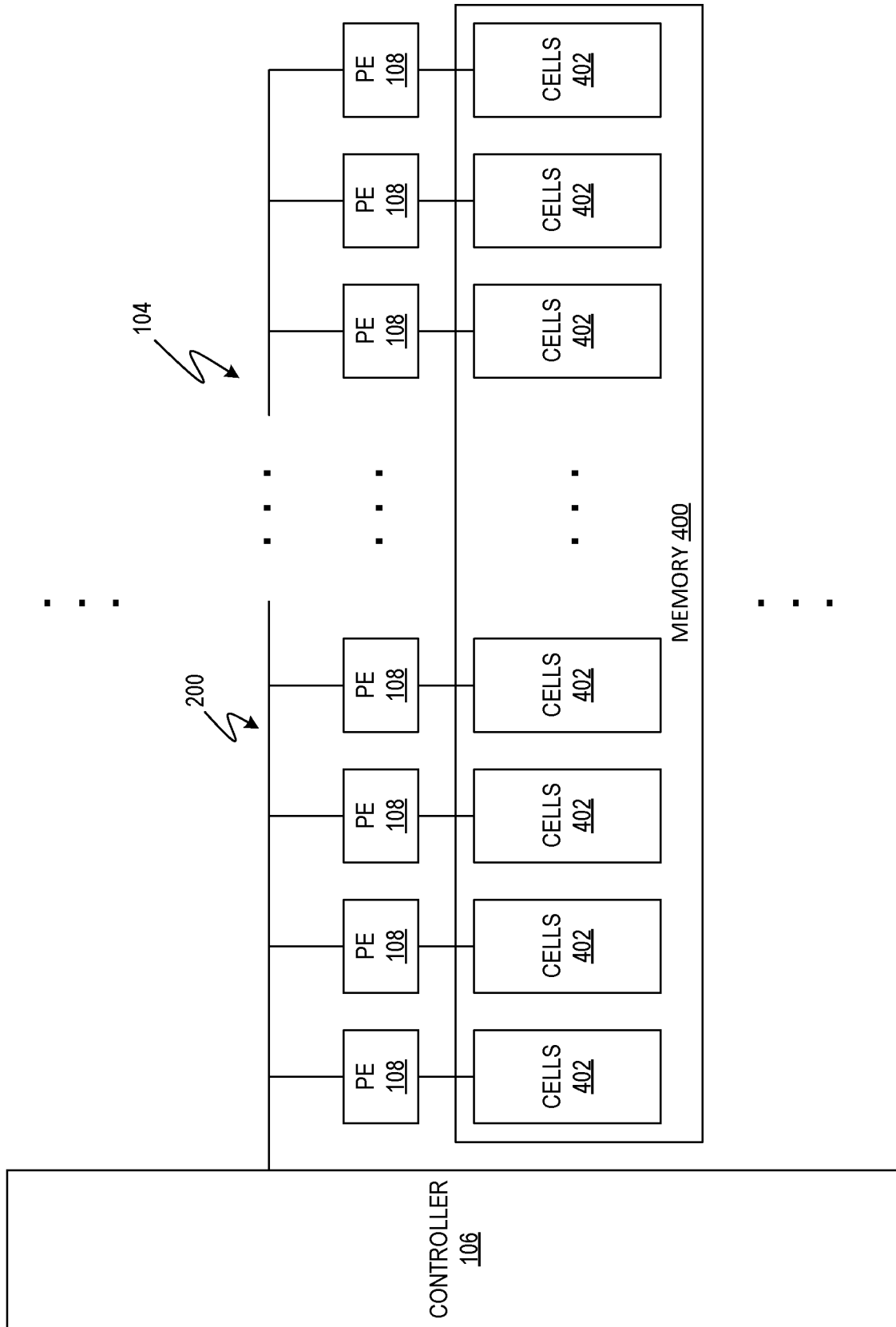


FIG. 4

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

FIG. 6

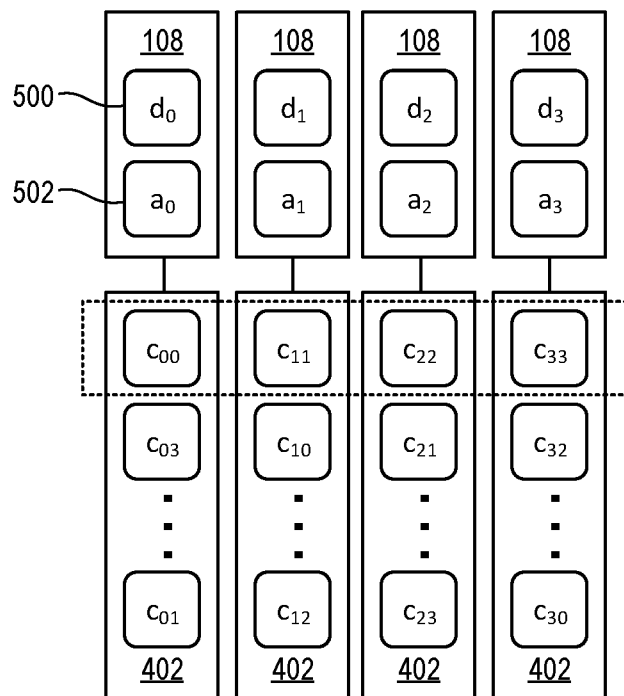


FIG. 5

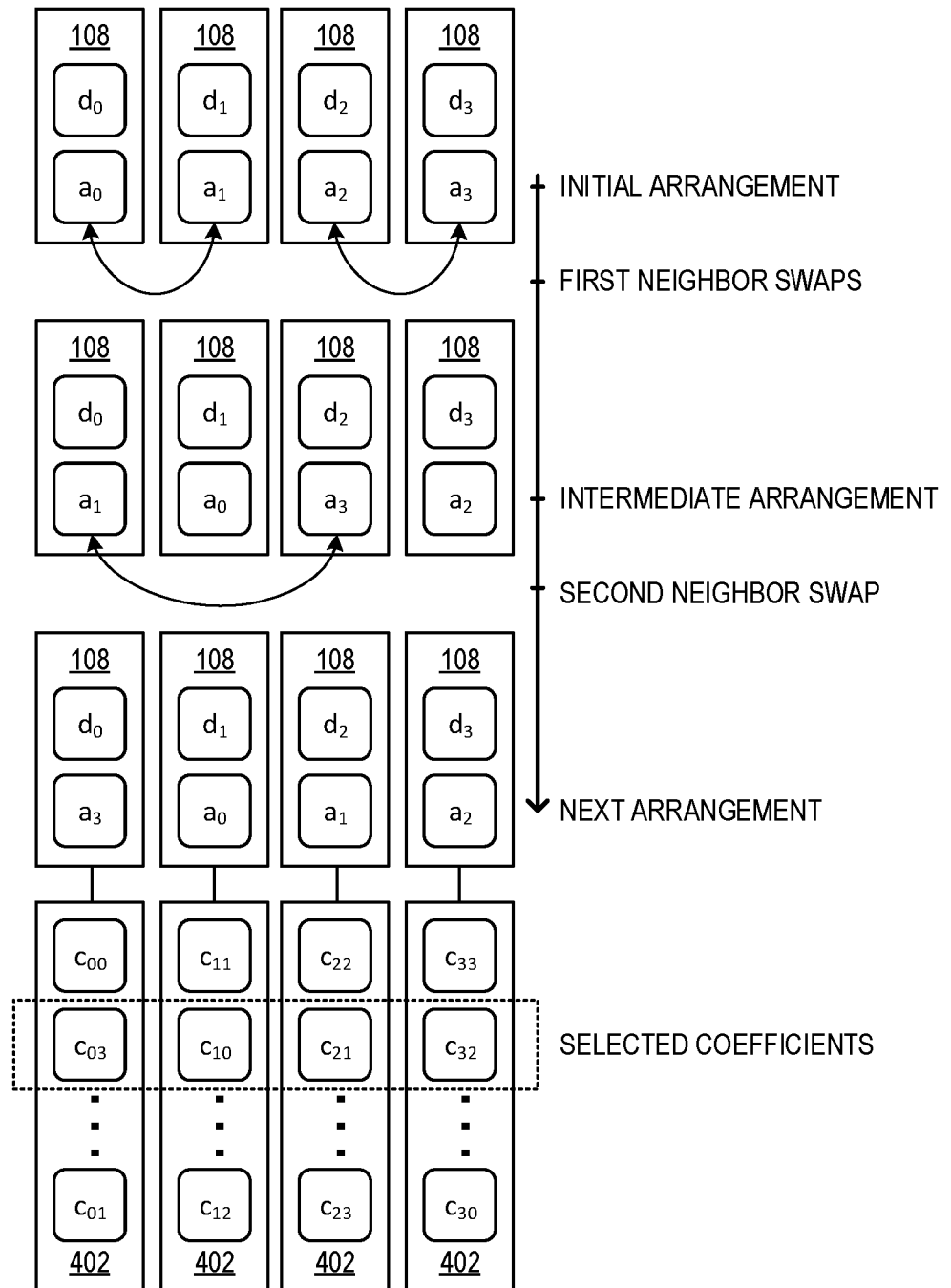


FIG. 7A

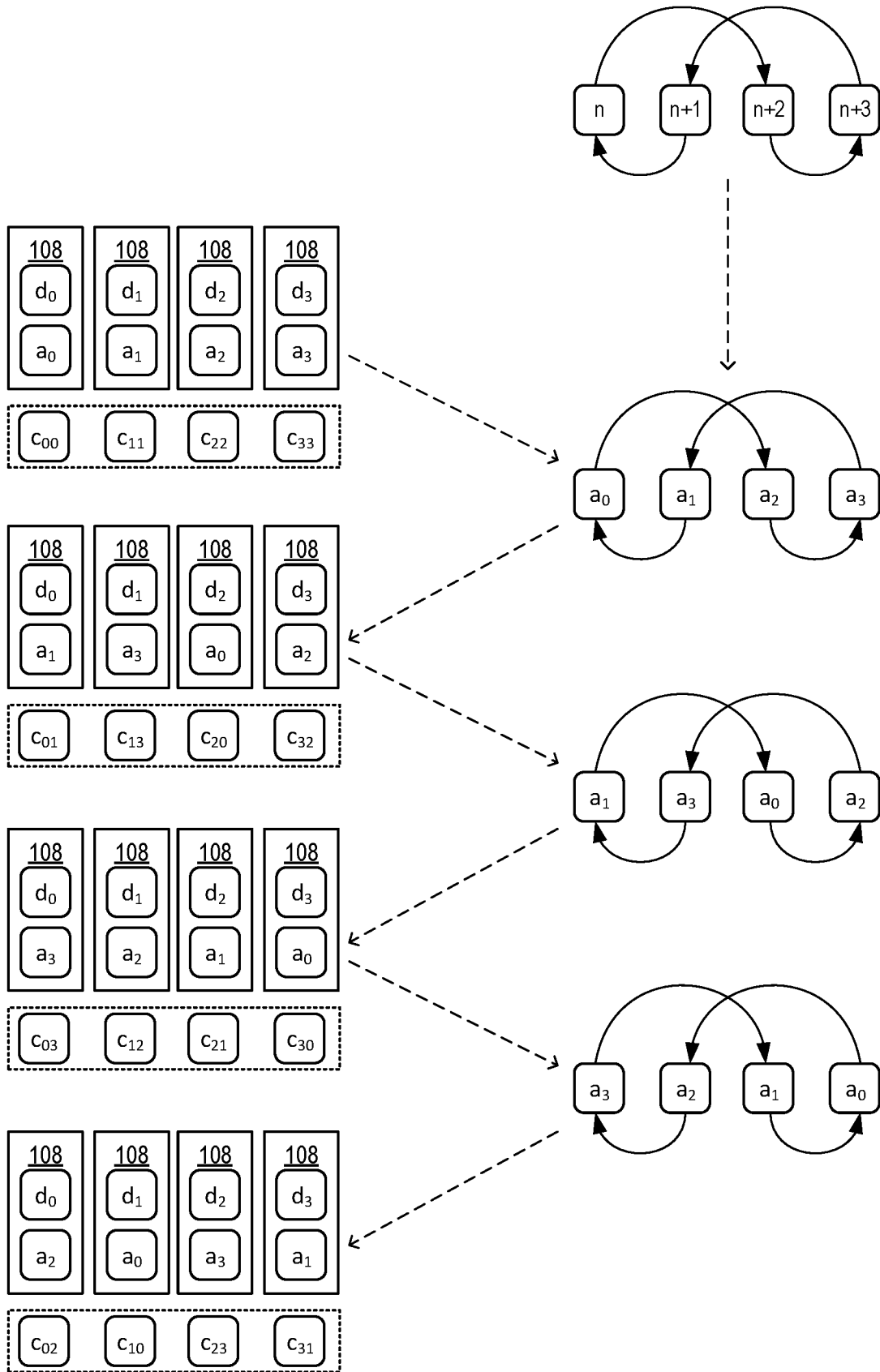


FIG. 7B

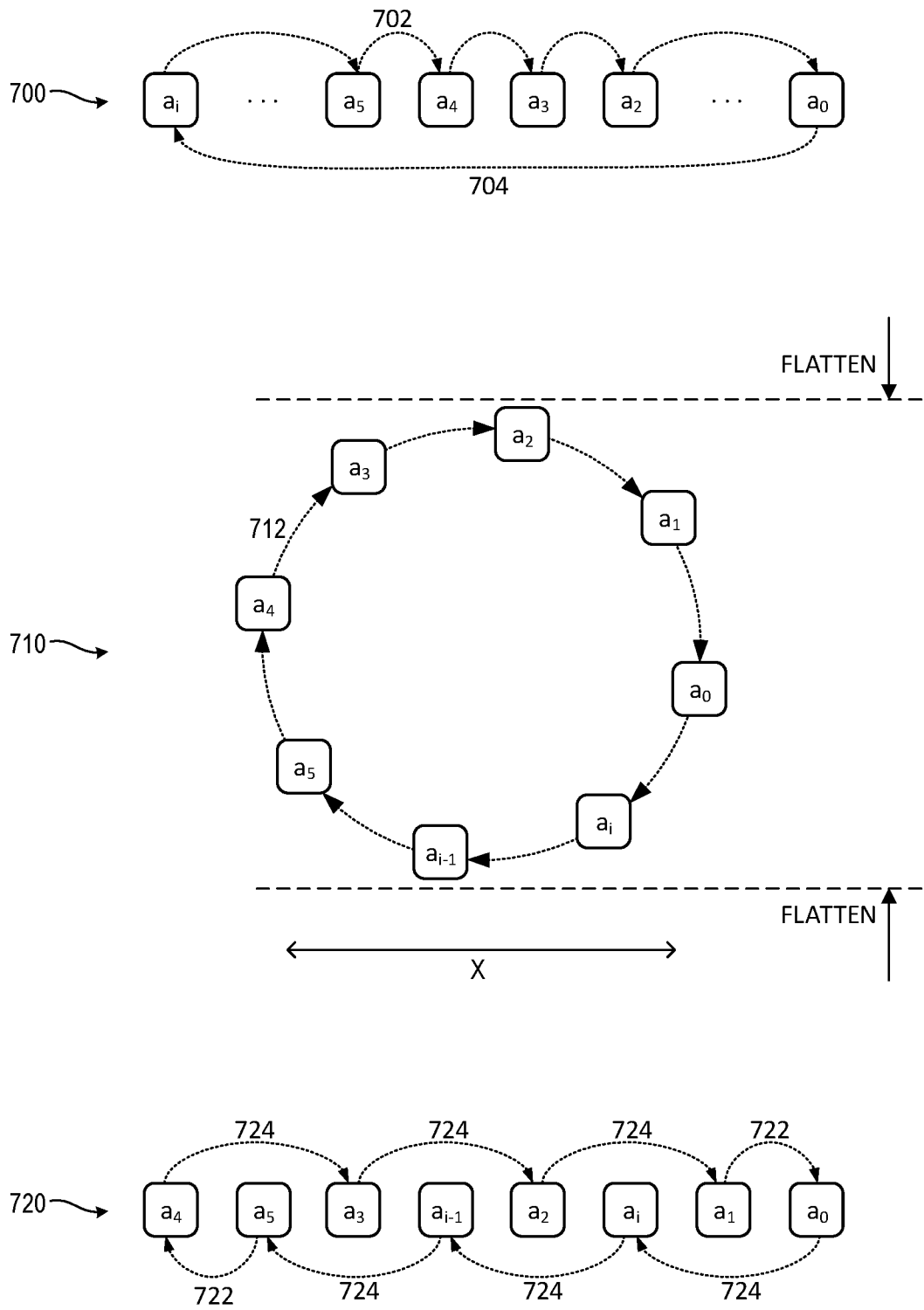


FIG. 7C

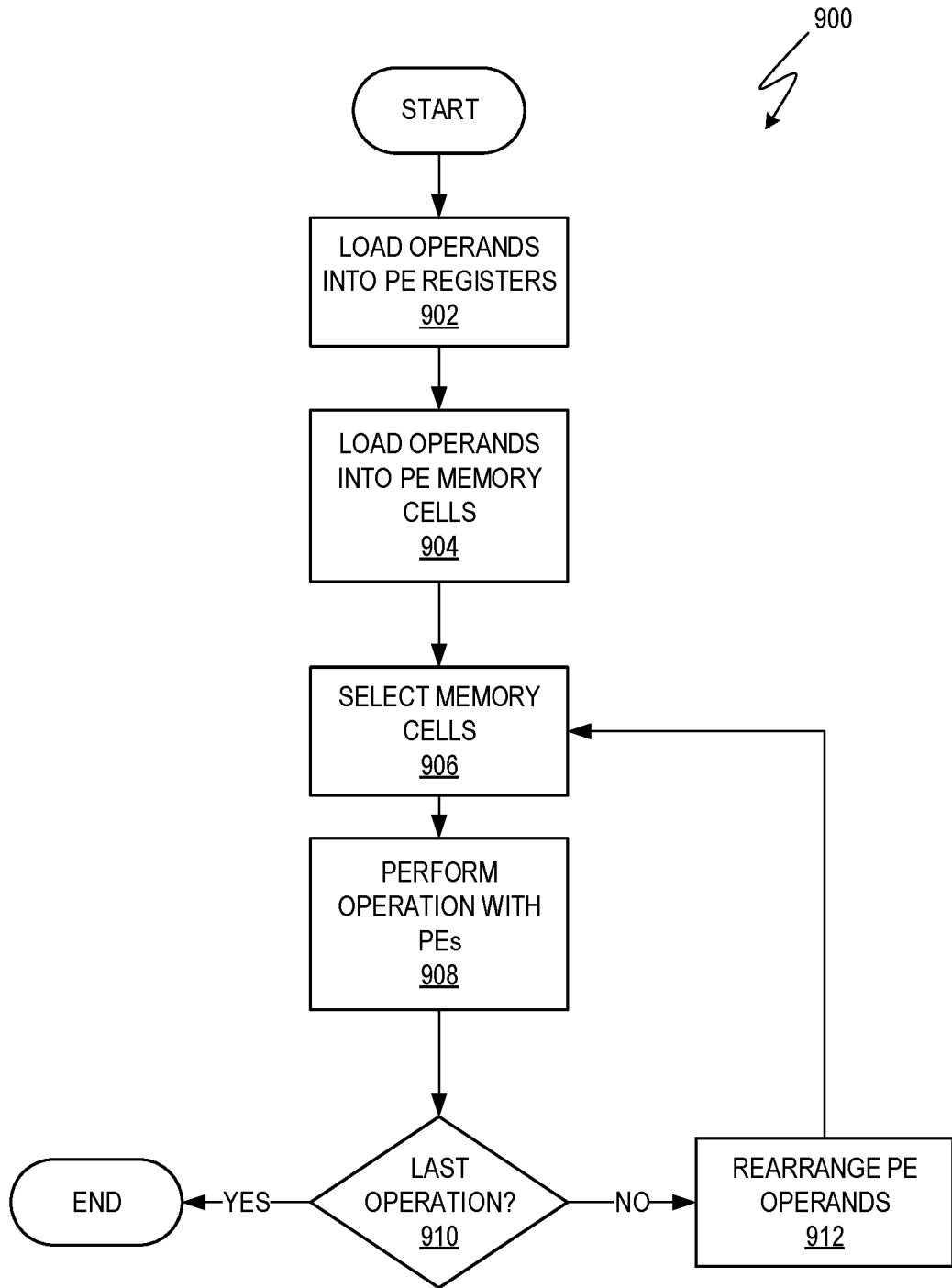


FIG. 8

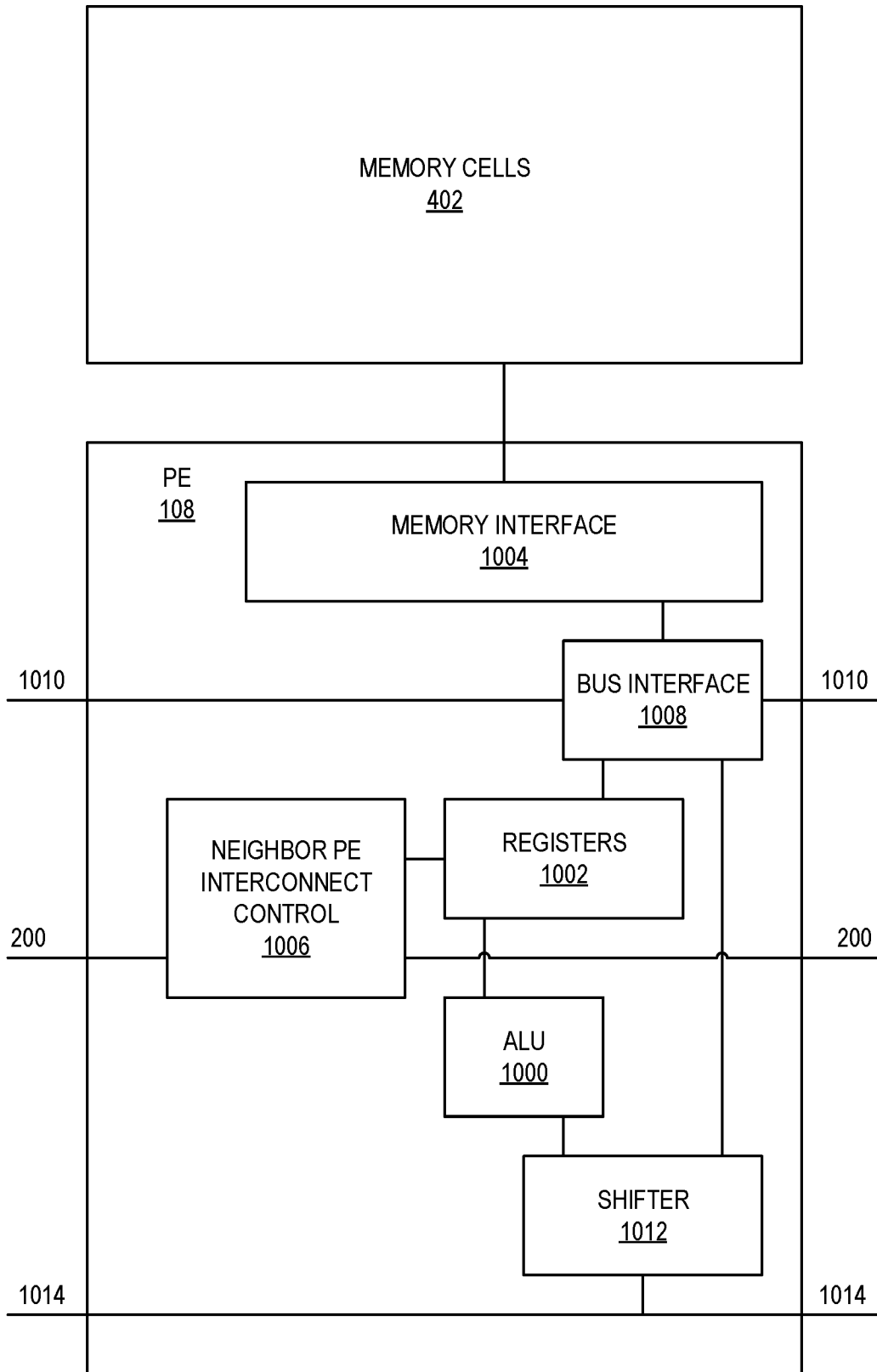


FIG. 9

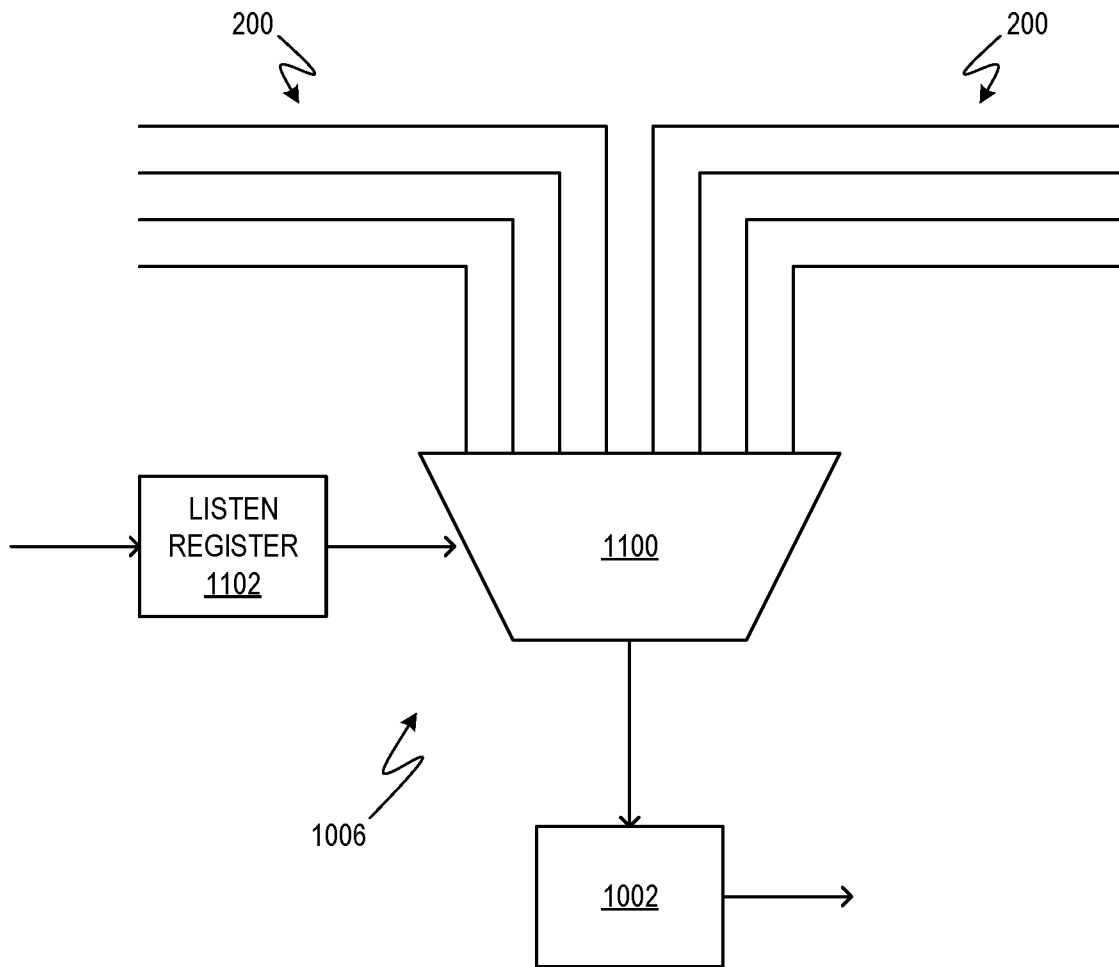


FIG. 10

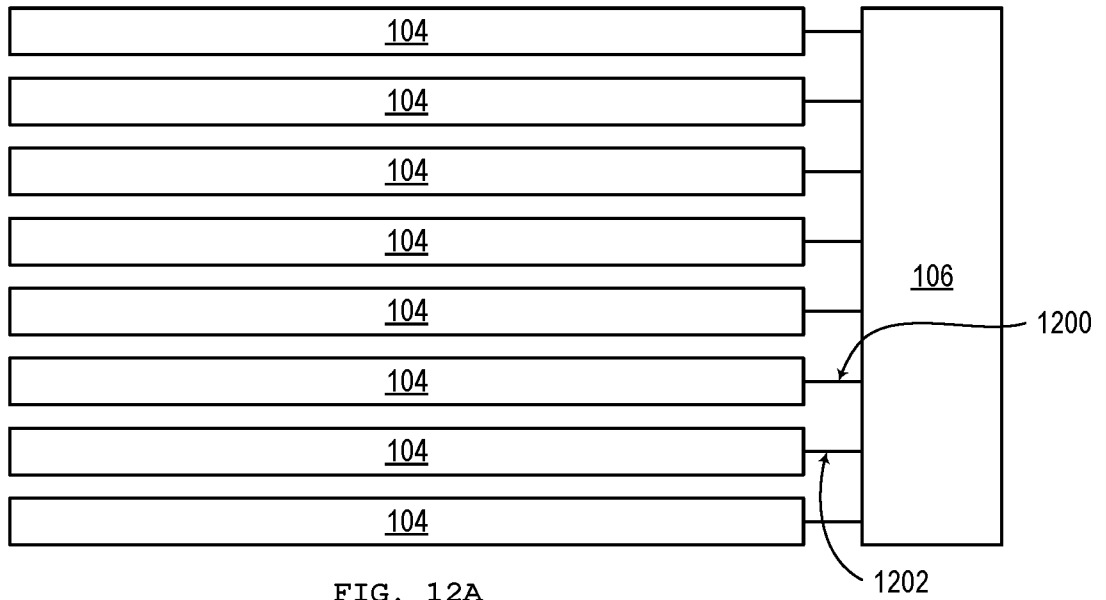


FIG. 12A

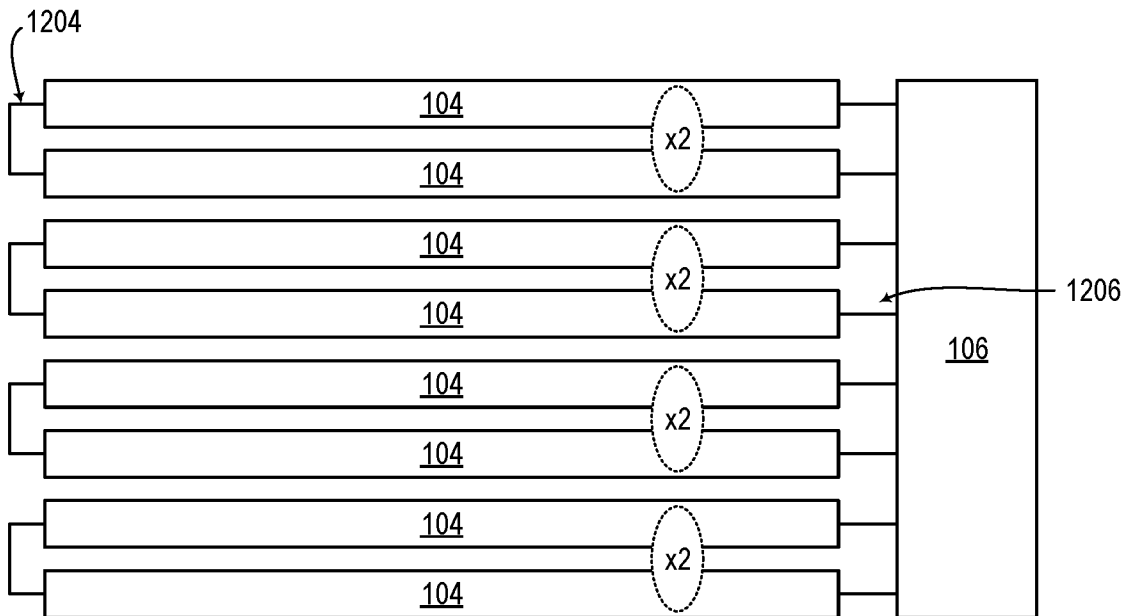


FIG. 12B

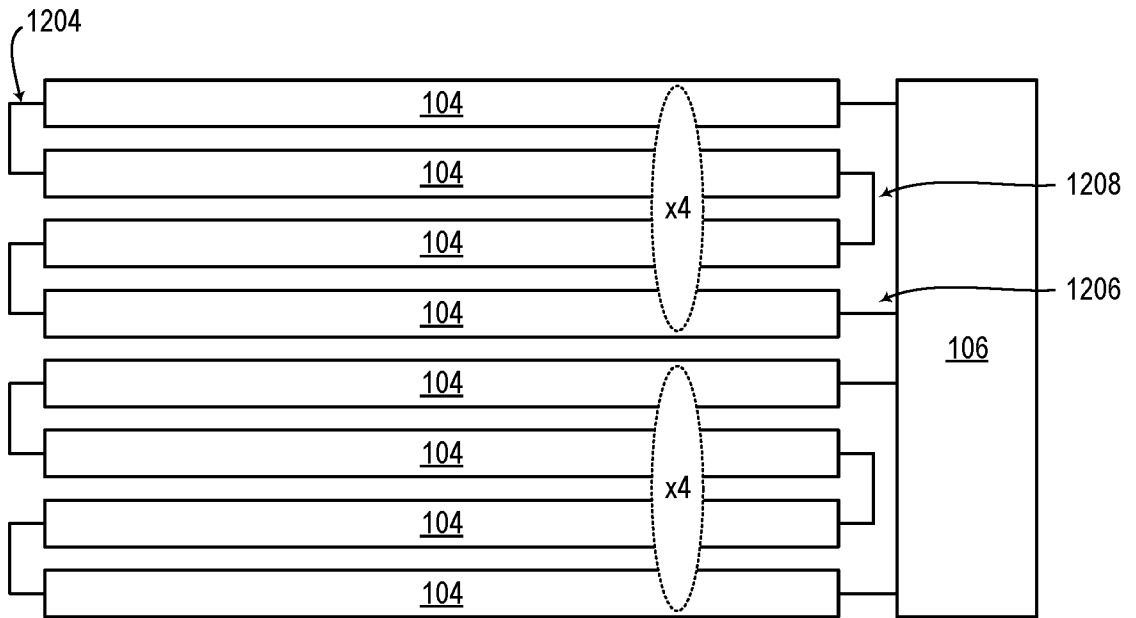


FIG. 12C

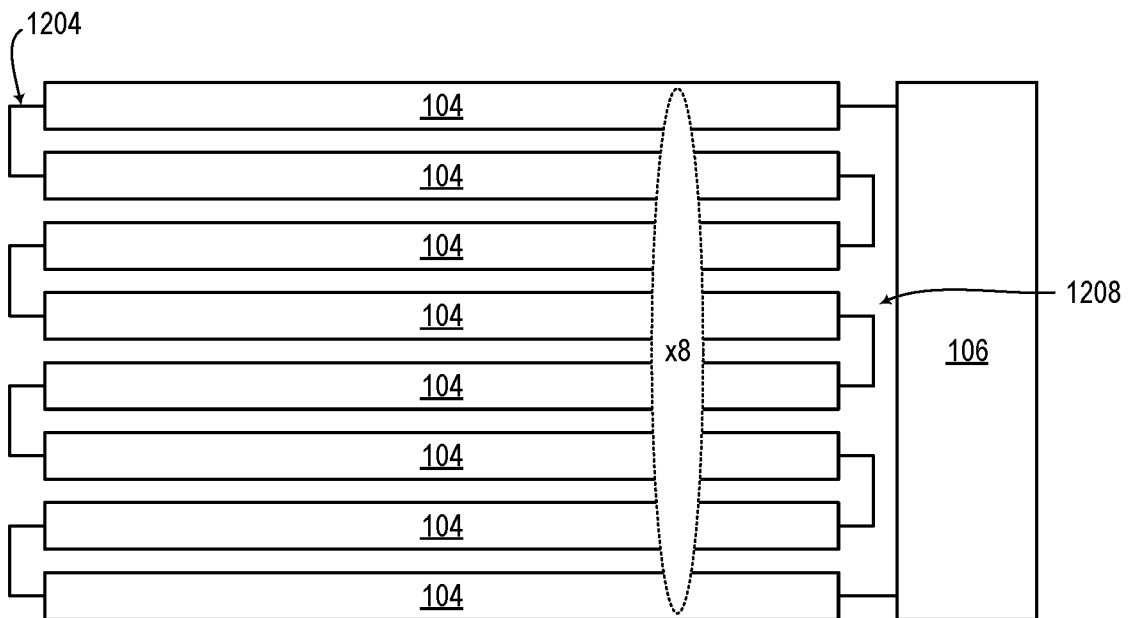


FIG. 12D

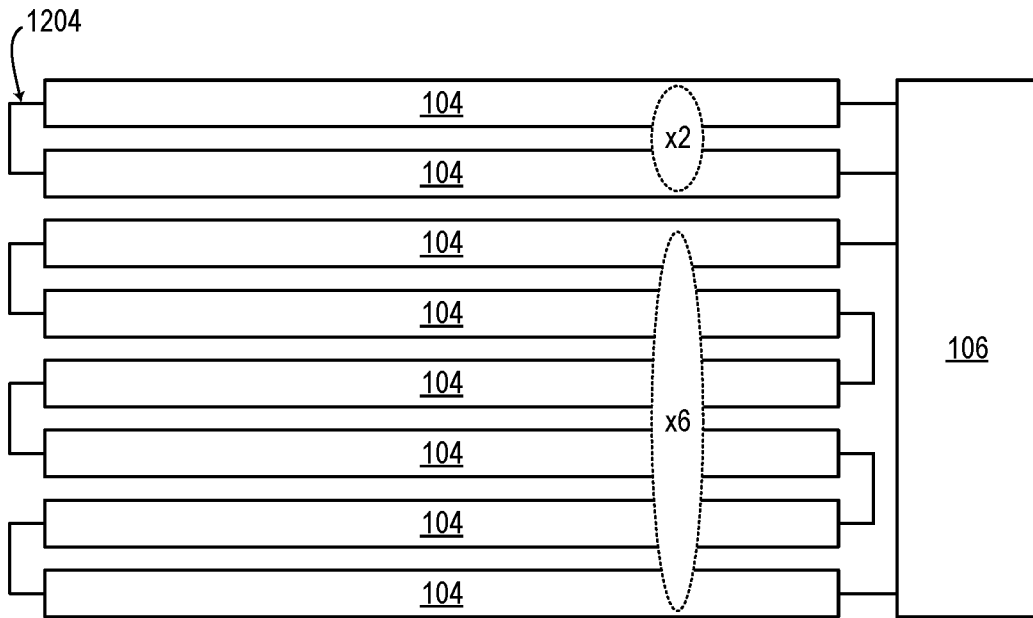


FIG. 12E

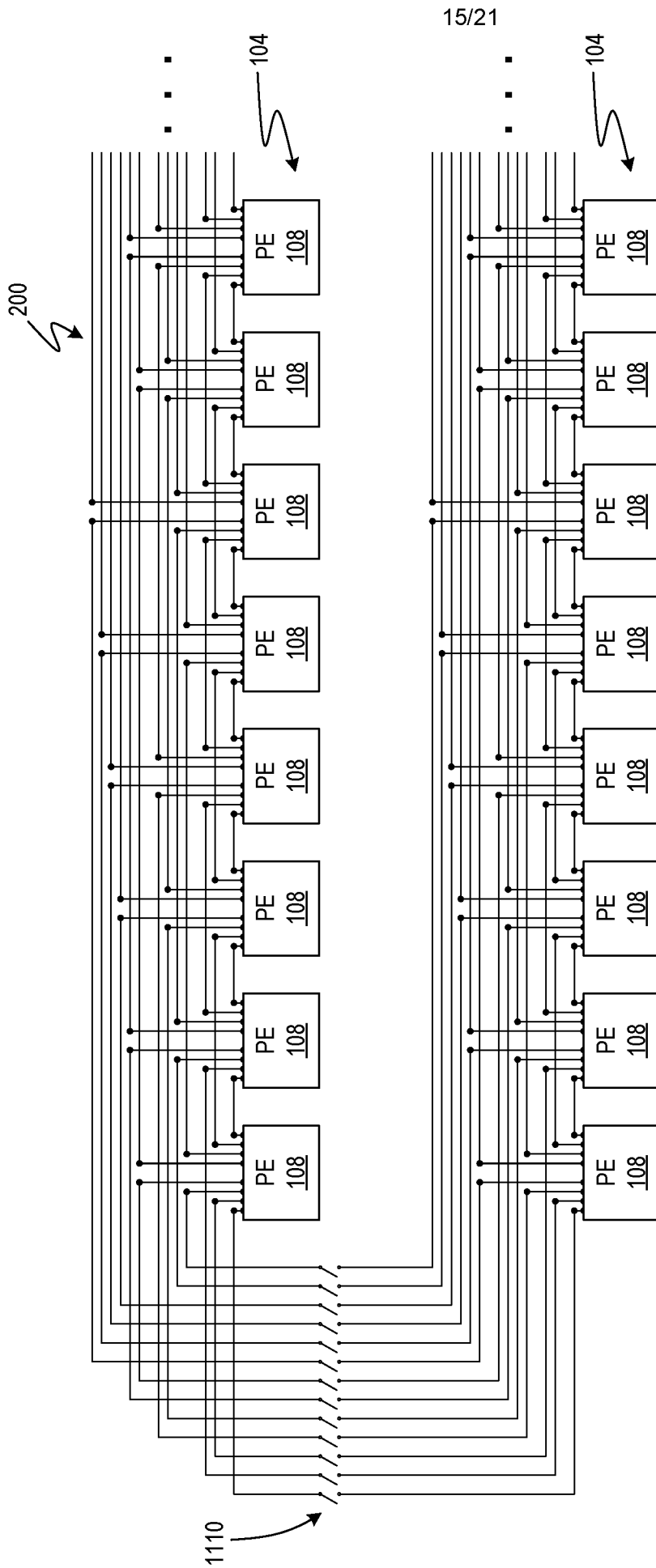


FIG. 13A

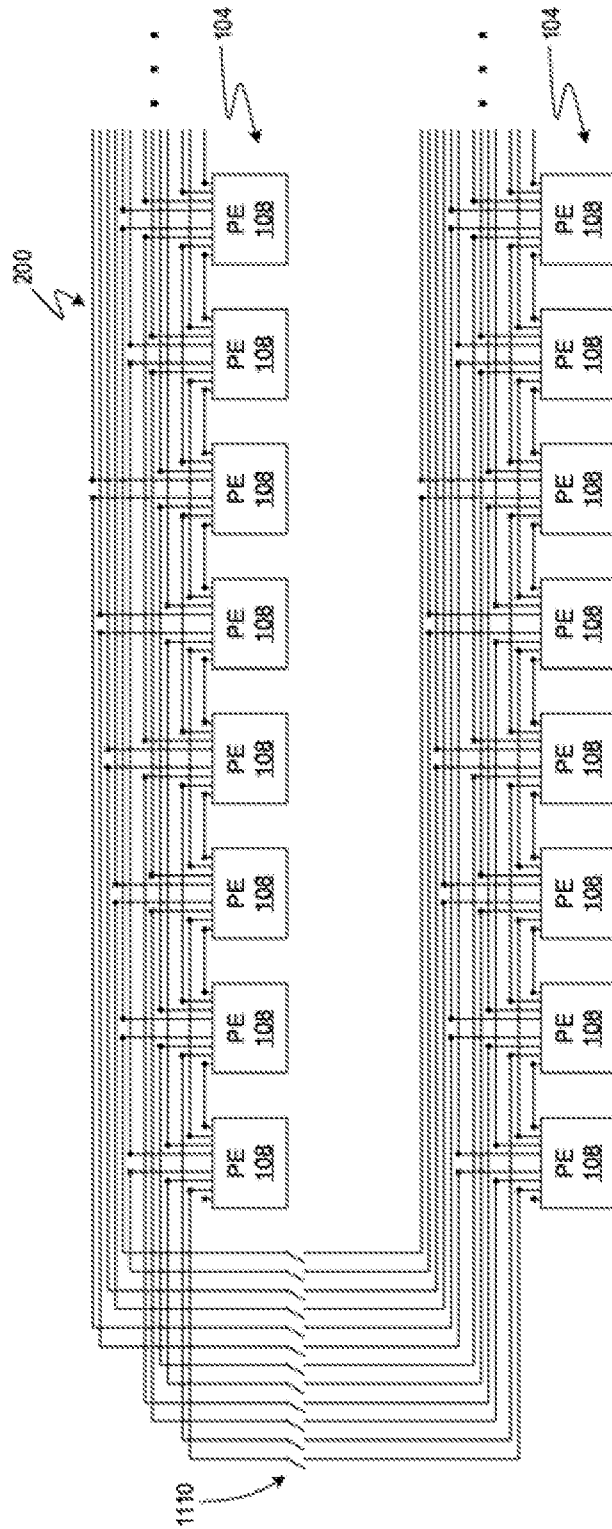


FIG. 13B

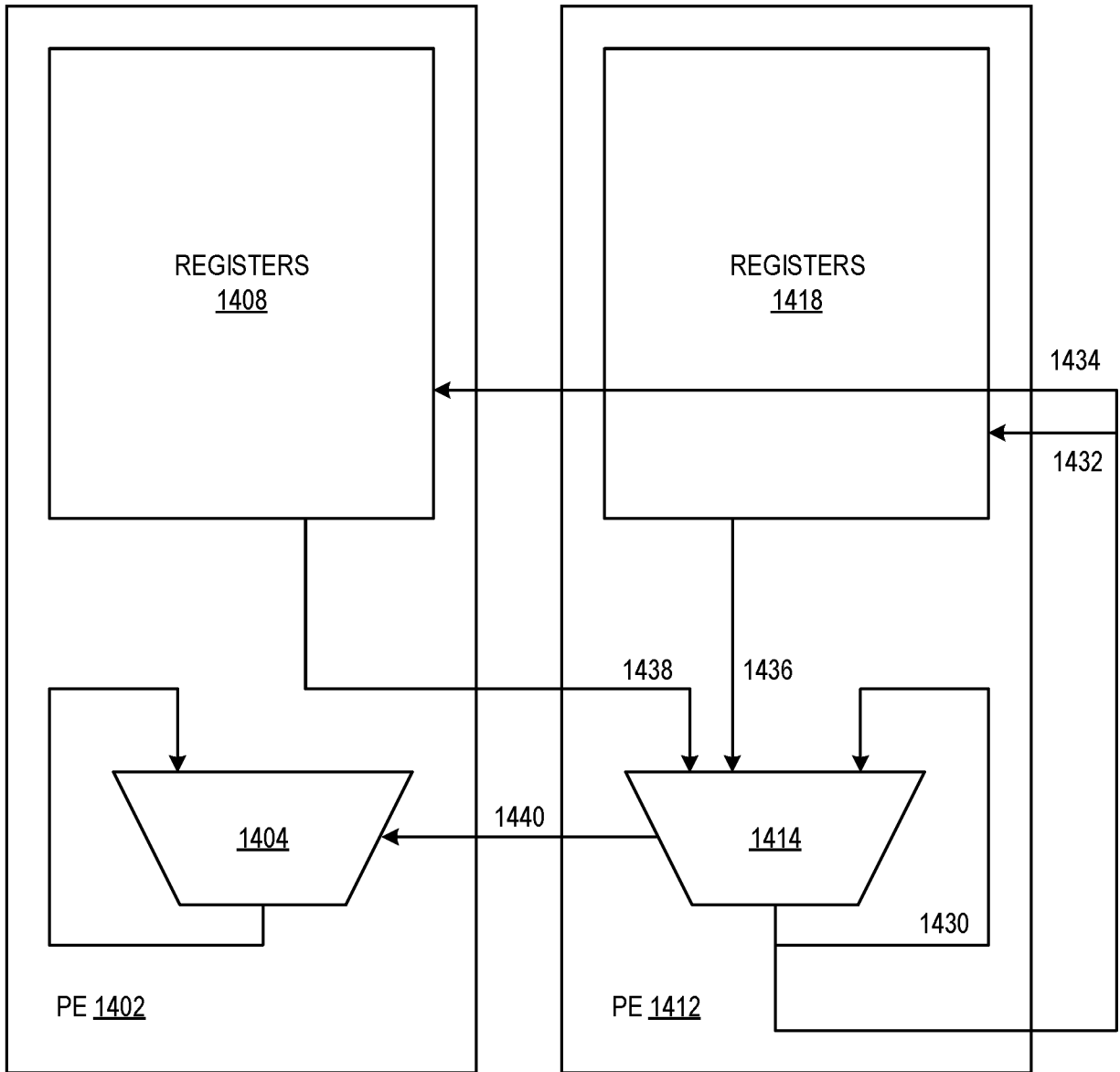


FIG. 14

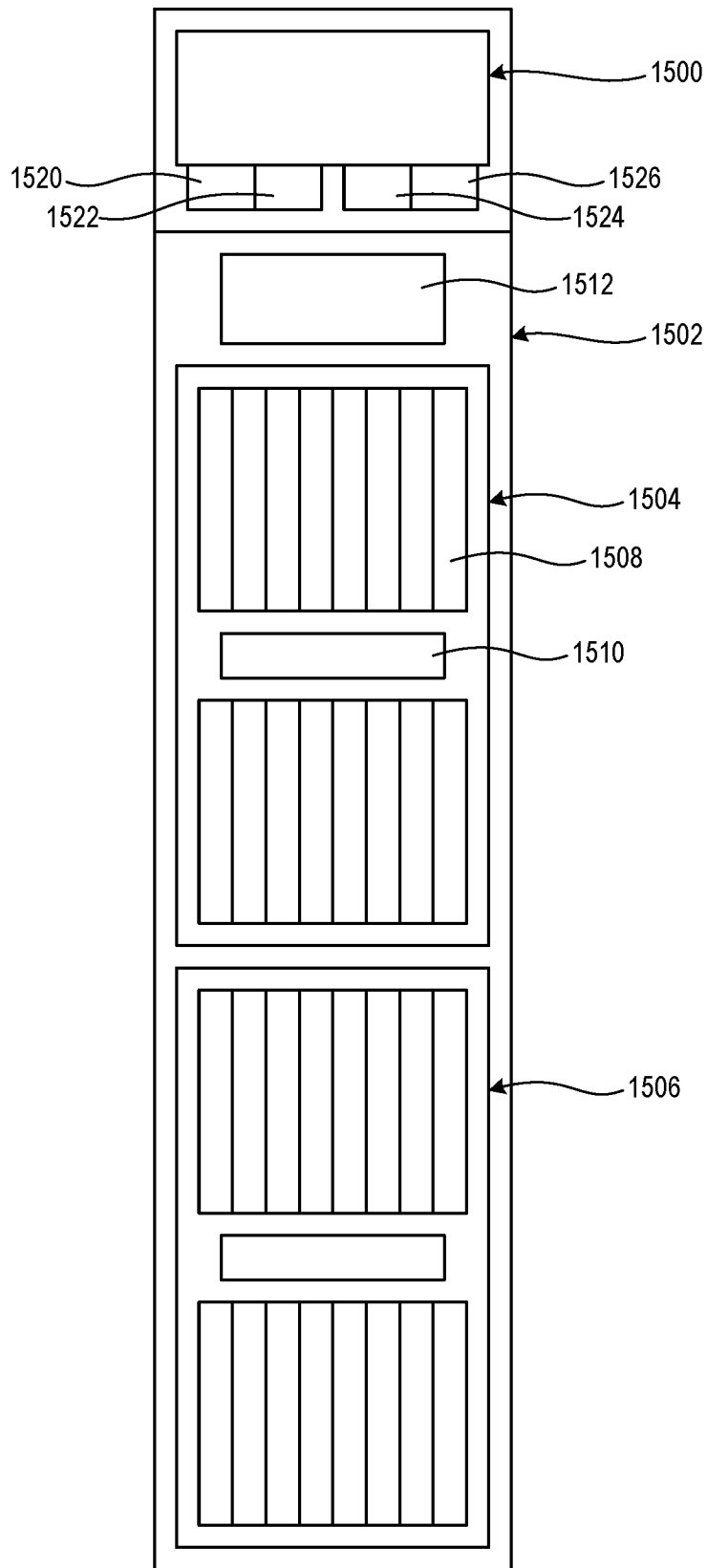


FIG. 15

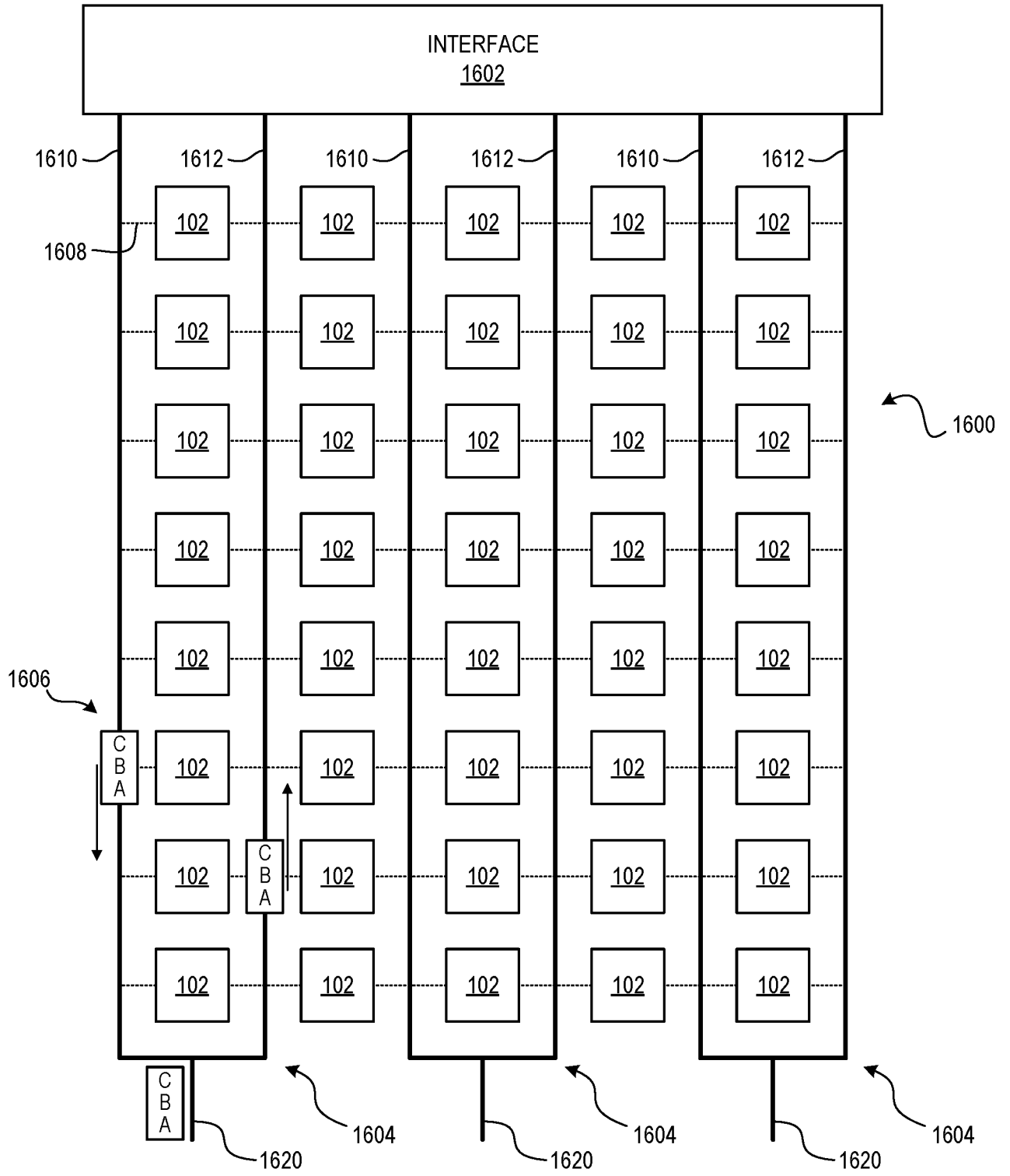


FIG. 16

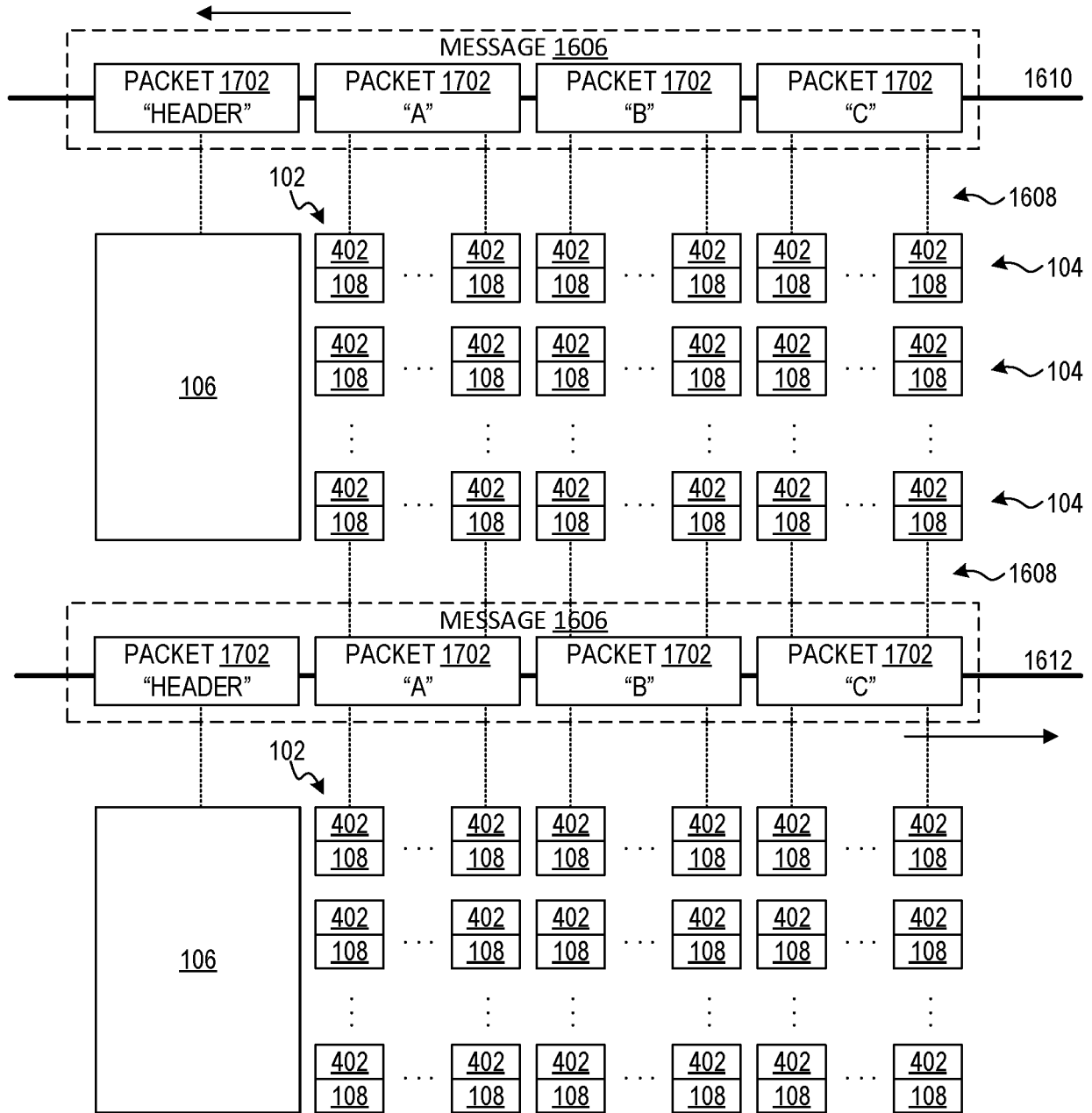


FIG. 17

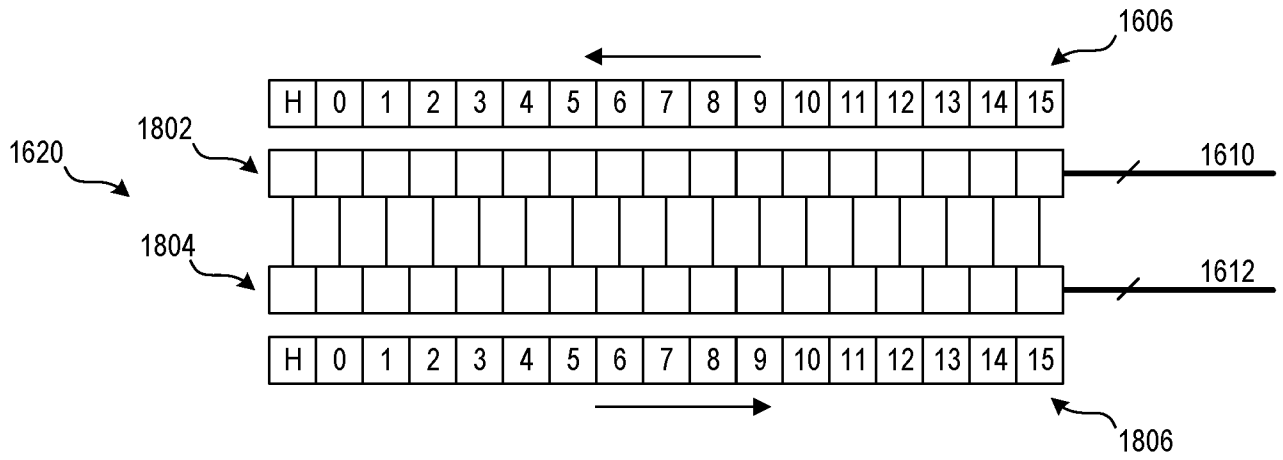


FIG. 18

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2020/057724A. CLASSIFICATION OF SUBJECT MATTER
IPC: *G06N 3/04* (2006.01), *G06F 15/76* (2006.01), *G06N 3/063* (2006.01)

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)
IPC: *G06N 3/04* (2006.01), *G06F 15/76* (2006.01), *G06N 3/063* (2006.01)

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

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)

Databases: Questel-Orbit (FamPat); Search Engine: Google

Keywords: single instruction multiple data, SIMD, processing elements, mesh, selectable rows, configurable switch, array, grid, controller

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	SVENSSON, B., "SIMD Processor Array Architectures", pages 1-44, 16 May 1990 (16-05-1990) retrieved from https://www.cs.utexas.edu/~rossbach/cs378h/papers/simd.pdf on 13 November 2020 (13-11-2020) *whole document, in particular: - sections 1,2 *	1-7
Y	US6145072A (SHAMS, S. et al.) 7 November 2000 (07-11-2000) *whole document, in particular: - col 1, last two paragraphs, col 4, last paragraph, col 6, lines 56-65, cols 7-8; - claim 1*	1-7

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

* Special categories of cited documents:	"I" 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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"D" document cited by the applicant in the international application	"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
"E" earlier application or patent but published on or after the international filing date	"&" document member of the same patent family
"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	

Date of the actual completion of the international search
13 November 2020 (13-11-2020)Date of mailing of the international search report
23 November 2020 (23-11-2020)Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage 1, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 819-953-2476

Authorized officer

Ayube Ali (873) 353-0486

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	SERRANO, M.J. et al., "Optimal Architectures and Algorithms for Mesh-Connected Parallel Computers with Separable Row/Column Buses", <i>IEEE Transactions on Parallel and Distributed Systems</i> , Volume: 4, Issue: 10, pages 1073-1080, 31 October 1993 (31-10-1993) retrieved from https://ieeexplore.ieee.org/document/246069 on 13 November 2020 (13-11-2020) *whole document, in particular: - abstract and section I *	1-7
A	BEIVIDE, R. et al., "Optimized mesh-connected networks for SIMD and MIMD architectures", <i>ISCA '87: Proceedings of the 14th annual international symposium on Computer architecture</i> , pages 163-170, 30 June 1987 (30-06-1987), retrieved from https://dl.acm.org/doi/abs/10.1145/30350.30369 on 13 November 2020 (13-11-2020) *whole document*	
A	US8443169B2 (PECHANEK, G.G.) 14 May 2013 (14-05-2013) *whole document*	
A	US5903771A (SGRO, J.A. et al.) 11 May 1999 (11-05-1999) *whole document*	
A	US4809347A (NASH, J.G. et al.) 28 February 1989 (28-02-1989) *whole document*	
A	US5689719A (MIURA, H. et al.) 18 November 1997 (18-11-1997) *whole document*	
A	US6067609A (MEEKER, W.L. et al.) 23 May 2000 (23-05-2000) *whole document*	
A	US2013103925A1 (MEEKER, W.L.) 25 April 2013 (25-04-2013) *whole document*	
A	US2002198911A1 (BLOMGREN, J.S. et al.) 26 December 2002 (26-12-2002) *whole document*	
A	US6405185B1 (PECHANEK, G.G. et al.) 11 June 2002 (11-06-2002) *whole document*	
A	US5822608A (DIEFFENDERFER, J.W. et al.) 13 October 1998 (13-10-1998) *whole document*	

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of the 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. Claim Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. Claim 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. Claim 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:

Group A (claims 1-7) refers to a device comprising: an array of rows of processing elements (PE's), each row terminating at a first-end PE and a second-end PE opposite the first-end PE, a controller connected to a first-end PE to provide data and instructions according to a single instruction, multiple data scheme, a second-end switch to selectively connect second-end PE's of adjacent rows, and a first-end switch to selectively connect first-end PE's of adjacent rows.

Group B (claims 8-10) refers to a device comprising: a two-dimensional array of processing banks, each processing bank including a plurality of rows of processing elements (PE's) operable according to a single instruction, multiple data scheme; a bus including two legs, wherein a column of processing banks is adjacent to and connected to the one of the legs, a second column in between and connected to both legs, a reversing segment that joins the two legs, and an interface connected to the bus.

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 claim 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 claim Nos.:

1 - 7

- 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
Information on patent family members

International application No.

PCT/IB2020/057724

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US6145072A	07 November 2000 (07-11-2000)	CA2129882A1 EP0638867A2 EP0638867A3 IL110634D0 JPH07152722A	13 February 1995 (13-02-1995) 15 February 1995 (15-02-1995) 18 October 1995 (18-10-1995) 11 November 1994 (11-11-1994) 16 June 1995 (16-06-1995)
US8443169B2	14 May 2013 (14-05-2013)	US2011161625A1 US2006218376A1 US7581079B2 US2009265512A1 US7886128B2 US2011072237A1 US8156311B2 US2013198488A1 US9460048B2 US2015039855A1 US9507603B2 US2016357569A1 US10078517B2 US2018189068A1 US10503515B2	30 June 2011 (30-06-2011) 28 September 2006 (28-09-2006) 25 August 2009 (25-08-2009) 22 October 2009 (22-10-2009) 08 February 2011 (08-02-2011) 24 March 2011 (24-03-2011) 10 April 2012 (10-04-2012) 01 August 2013 (01-08-2013) 04 October 2016 (04-10-2016) 05 February 2015 (05-02-2015) 29 November 2016 (29-11-2016) 08 December 2016 (08-12-2016) 18 September 2018 (18-09-2018) 05 July 2018 (05-07-2018) 10 December 2019 (10-12-2019)
US5903771A	11 May 1999 (11-05-1999)	AU1577297A AU725592B2 CA2243547A1 EP0875027A1 IL125383D0 WO9726593A1	11 August 1997 (11-08-1997) 12 October 2000 (12-10-2000) 24 July 1997 (24-07-1997) 04 November 1998 (04-11-1998) 12 March 1999 (12-03-1999) 24 July 1997 (24-07-1997)
US4809347A	28 February 1989 (28-02-1989)	DE3780615D1 DE3780615T2 DE3782653D1 DE3782653T2 EP0276237A1 EP0276237B1 EP0301086A1 EP0301086B1 ES2004645A6 ES2006312A6 IL82826D0 IL82826A IL84933D0 IL84933A JPH01500780A JP2624733B2 JPH01502220A JPH07104948B2 US4809346A WO8800737A1 WO8806322A2 WO8806322A3	27 August 1992 (27-08-1992) 11 March 1993 (11-03-1993) 17 December 1992 (17-12-1992) 29 April 1993 (29-04-1993) 03 August 1988 (03-08-1988) 22 July 1992 (22-07-1992) 01 February 1989 (01-02-1989) 11 November 1992 (11-11-1992) 16 January 1989 (16-01-1989) 16 April 1989 (16-04-1989) 20 December 1987 (20-12-1987) 23 December 1990 (23-12-1990) 30 June 1988 (30-06-1988) 25 May 1992 (25-05-1992) 16 March 1989 (16-03-1989) 25 June 1997 (25-06-1997) 03 August 1989 (03-08-1989) 13 November 1995 (13-11-1995) 28 February 1989 (28-02-1989) 28 January 1988 (28-01-1988) 25 August 1988 (25-08-1988) 07 September 1988 (07-09-1988)
US5689719A	18 November 1997 (18-11-1997)	JPH0512230A JP2834909B2 JPH0512231A JP2975722B2 JPH05127899A JP3133426B2 US5898881A	22 January 1993 (22-01-1993) 14 December 1998 (14-12-1998) 22 January 1993 (22-01-1993) 10 November 1999 (10-11-1999) 25 May 1993 (25-05-1993) 05 February 2001 (05-02-2001) 27 April 1999 (27-04-1999)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2020/057724

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US6067609A	23 May 2000 (23-05-2000)	AU3741499A CO4880825A1 EP1070295A1 TW454125B WO9953413A1	01 November 1999 (01-11-1999) 31 January 2000 (31-01-2000) 24 January 2001 (24-01-2001) 11 September 2001 (11-09-2001) 21 October 1999 (21-10-1999)
US2013103925A1	25 April 2013 (25-04-2013)	US8898432B2	25 November 2014 (25-11-2014)
US2002198911A1	26 December 2002 (26-12-2002)	US6898691B2	24 May 2005 (24-05-2005)
US6405185B1	11 June 2002 (11-06-2002)	EP0459222A2 EP0459222A3 EP0484479A1 EP0484479A4 EP0484506A1 EP0484506A4 EP0484507A1 EP0484507A4 EP0484522A1 EP0484522A4 EP0486635A1 EP0486635A4 EP0486684A1 EP0486684A4 EP0564847A2 EP0564847A3 JPH04507027A JP2502867B2 JPH04507025A JP2575565B2 JPH0675986A JP2647327B2 JPH05500429A JP2663995B2 JPH04505824A JP2663996B2 JPH04506879A JP2666830B2 JPH04507026A JP2746350B2 JPH04232562A JPH07117948B2 US5065339A US5146420A US5146543A US5148515A US5243688A US5251287A US5325464A US5329611A US5337395A US5483620A US5509106A US5517596A US5542026A US5577262A US5612908A US5613044A US5617512A	04 December 1991 (04-12-1991) 30 March 1994 (30-03-1994) 13 May 1992 (13-05-1992) 13 April 1994 (13-04-1994) 13 May 1992 (13-05-1992) 23 March 1994 (23-03-1994) 13 May 1992 (13-05-1992) 15 July 1992 (15-07-1992) 13 May 1992 (13-05-1992) 23 March 1994 (23-03-1994) 27 May 1992 (27-05-1992) 23 March 1994 (23-03-1994) 27 May 1992 (27-05-1992) 23 March 1994 (23-03-1994) 27 May 1992 (27-05-1992) 23 March 1994 (23-03-1994) 13 October 1993 (13-10-1993) 20 July 1994 (20-07-1994) 03 December 1992 (03-12-1992) 29 May 1996 (29-05-1996) 03 December 1992 (03-12-1992) 29 January 1997 (29-01-1997) 18 March 1994 (18-03-1994) 27 August 1997 (27-08-1997) 28 January 1993 (28-01-1993) 15 October 1997 (15-10-1997) 08 October 1992 (08-10-1992) 15 October 1997 (15-10-1997) 26 November 1992 (26-11-1992) 22 October 1997 (22-10-1997) 03 December 1992 (03-12-1992) 06 May 1998 (06-05-1998) 20 August 1992 (20-08-1992) 18 December 1995 (18-12-1995) 12 November 1991 (12-11-1991) 08 September 1992 (08-09-1992) 08 September 1992 (08-09-1992) 15 September 1992 (15-09-1992) 07 September 1993 (07-09-1993) 05 October 1993 (05-10-1993) 28 June 1994 (28-06-1994) 12 July 1994 (12-07-1994) 09 August 1994 (09-08-1994) 09 January 1996 (09-01-1996) 16 April 1996 (16-04-1996) 14 May 1996 (14-05-1996) 30 July 1996 (30-07-1996) 19 November 1996 (19-11-1996) 18 March 1997 (18-03-1997) 18 March 1997 (18-03-1997) 01 April 1997 (01-04-1997)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2020/057724

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
		WO9118347A1	28 November 1991 (28-11-1991)
		WO9118348A1	28 November 1991 (28-11-1991)
		WO9118349A1	28 November 1991 (28-11-1991)
		WO9118350A1	28 November 1991 (28-11-1991)
		WO9118351A1	28 November 1991 (28-11-1991)
		WO9201257A1	23 January 1992 (23-01-1992)
US5822608A	13 October 1998 (13-10-1998)	AT177547T	15 March 1999 (15-03-1999)
		AT180586T	15 June 1999 (15-06-1999)
		BR9104603A	23 June 1992 (23-06-1992)
		CA2050166A1	14 May 1992 (14-05-1992)
		CA2064164A1	14 November 1992 (14-11-1992)
		CA2073516A1	28 May 1993 (28-05-1993)
		CN1072788A	02 June 1993 (02-06-1993)
		CN1044646C	11 August 1999 (11-08-1999)
		CN1061482A	27 May 1992 (27-05-1992)
		CN1050919C	29 March 2000 (29-03-2000)
		CZ344091A3	17 May 1995 (17-05-1995)
		CZ280210B6	13 December 1995 (13-12-1995)
		DE69131272D1	01 July 1999 (01-07-1999)
		DE69131272T2	09 December 1999 (09-12-1999)
		DE69228586D1	15 April 1999 (15-04-1999)
		DE69228586T2	14 October 1999 (14-10-1999)
		EP0485690A2	20 May 1992 (20-05-1992)
		EP0485690A3	21 September 1994 (21-09-1994)
		EP0485690B1	26 May 1999 (26-05-1999)
		EP0514043A2	19 November 1992 (19-11-1992)
		EP0514043A3	18 May 1994 (18-05-1994)
		EP0544127A2	02 June 1993 (02-06-1993)
		EP0544127A3	20 April 1994 (20-04-1994)
		EP0544127B1	10 March 1999 (10-03-1999)
		EP0570729A2	24 November 1993 (24-11-1993)
		EP0570729A3	20 July 1994 (20-07-1994)
		EP0570741A2	24 November 1993 (24-11-1993)
		EP0570741A3	28 September 1994 (28-09-1994)
		EP0570950A2	24 November 1993 (24-11-1993)
		EP0570950A3	20 July 1994 (20-07-1994)
		EP0570951A2	24 November 1993 (24-11-1993)
		EP0570951A3	19 October 1994 (19-10-1994)
		EP0570952A2	24 November 1993 (24-11-1993)
		EP0570952A3	20 April 1994 (20-04-1994)
		HU913542D0	28 February 1992 (28-02-1992)
		HUT59496A	28 May 1992 (28-05-1992)
		HU215139B	28 September 1998 (28-09-1998)
		JPH05181821A	23 July 1993 (23-07-1993)
		JP2512661B2	03 July 1996 (03-07-1996)
		JPH0635878A	10 February 1994 (10-02-1994)
		JP2521401B2	07 August 1996 (07-08-1996)
		JPH0636060A	10 February 1994 (10-02-1994)
		JP2525117B2	14 August 1996 (14-08-1996)
		JPH0628325A	04 February 1994 (04-02-1994)
		JP2533282B2	11 September 1996 (11-09-1996)
		JPH0619864A	28 January 1994 (28-01-1994)
		JP2543306B2	16 October 1996 (16-10-1996)
		JPH0635873A	10 February 1994 (10-02-1994)
		JP2549240B2	30 October 1996 (30-10-1996)
		JPH0675931A	18 March 1994 (18-03-1994)
		JP2549241B2	30 October 1996 (30-10-1996)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2020/057724

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
		JPH0635875A	10 February 1994 (10-02-1994)
		JP2552075B2	06 November 1996 (06-11-1996)
		JPH0635872A	10 February 1994 (10-02-1994)
		JP2552076B2	06 November 1996 (06-11-1996)
		JPH0652125A	25 February 1994 (25-02-1994)
		JP2557175B2	27 November 1996 (27-11-1996)
		JPH06214964A	05 August 1994 (05-08-1994)
		JP2561800B2	11 December 1996 (11-12-1996)
		JPH06139200A	20 May 1994 (20-05-1994)
		JP2579419B2	05 February 1997 (05-02-1997)
		JPH0635876A	10 February 1994 (10-02-1994)
		JP2620487B2	11 June 1997 (11-06-1997)
		JPH0668282A	11 March 1994 (11-03-1994)
		JP2625628B2	02 July 1997 (02-07-1997)
		JPH05233569A	10 September 1993 (10-09-1993)
		JP2647315B2	27 August 1997 (27-08-1997)
		JPH06231092A	19 August 1994 (19-08-1994)
		JP2710536B2	10 February 1998 (10-02-1998)
		JPH0652124A	25 February 1994 (25-02-1994)
		JPH04267466A	24 September 1992 (24-09-1992)
		JPH07287700A	31 October 1995 (31-10-1995)
		KR920010473A	26 June 1992 (26-06-1992)
		KR960016880B1	26 December 1996 (26-12-1996)
		KR930010758A	23 June 1993 (23-06-1993)
		KR970008529B1	24 May 1997 (24-05-1997)
		MX9206864A	01 May 1993 (01-05-1993)
		PL292368A1	07 September 1992 (07-09-1992)
		PL167329B1	31 August 1995 (31-08-1995)
		RU2084953C1	20 July 1997 (20-07-1997)
		SK344091A3	05 January 1995 (05-01-1995)
		TW229289B	01 September 1994 (01-09-1994)
		US5313645A	17 May 1994 (17-05-1994)
		US5475856A	12 December 1995 (12-12-1995)
		US5517642A	14 May 1996 (14-05-1996)
		US5588152A	24 December 1996 (24-12-1996)
		US5590345A	31 December 1996 (31-12-1996)
		US5594918A	14 January 1997 (14-01-1997)
		US5615309A	25 March 1997 (25-03-1997)
		US5615360A	25 March 1997 (25-03-1997)
		US5617577A	01 April 1997 (01-04-1997)
		US5625836A	29 April 1997 (29-04-1997)
		US5630162A	13 May 1997 (13-05-1997)
		US5708836A	13 January 1998 (13-01-1998)
		US5710935A	20 January 1998 (20-01-1998)
		US5713037A	27 January 1998 (27-01-1998)
		US5717943A	10 February 1998 (10-02-1998)
		US5717944A	10 February 1998 (10-02-1998)
		US5734921A	31 March 1998 (31-03-1998)
		US5752067A	12 May 1998 (12-05-1998)
		US5754871A	19 May 1998 (19-05-1998)
		US5761523A	02 June 1998 (02-06-1998)