

(51) International Patent Classification:
G06F 9/30 (2006.01)(21) International Application Number:
PCT/US2016/041779(22) International Filing Date:
11 July 2016 (11.07.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/803,728 20 July 2015 (20.07.2015) US(71) Applicant: **QUALCOMM INCORPORATED** [US/US];
ATTN: International IP Administration, 5775 Morehouse
Drive, San Diego, California 92121-1714 (US).(72) Inventors: **MAHURIN, Eric**; 5775 Morehouse Drive, San
Diego, California 92121-1714 (US). **GOLAB, Jakub
Pawel**; 5775 Morehouse Drive, San Diego, California
92121-1714 (US).(74) Agent: **TOLER, Jeffrey G.**; 8500 Bluffstone Cove, Suite
A201, Austin, Texas 78759 (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, DK, DM,DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
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, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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).**Declarations under Rule 4.17:**

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- with international search report (Art. 21(3))

(54) Title: SIMD SLIDING WINDOW OPERATION

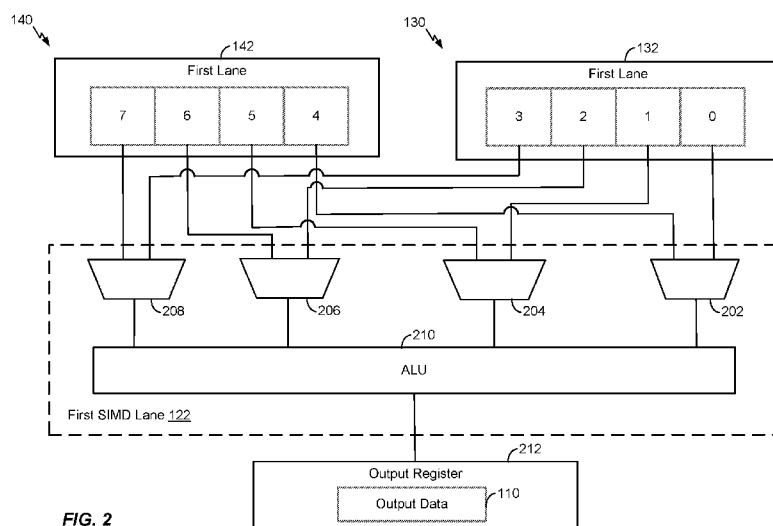


FIG. 2

(57) **Abstract:** A first register has a lane storing first input data and a second register has a lane storing second input data elements. A width of the lane of the second register is equal to a width of the lane of the first register. A single-instruction-multiple-data (SIMD) lane has a lane width equal to the width of the lane of the first register. The SIMD lane is configured to perform a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register. Performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register. The second position is different from the first position.



- 1 -

SIMD SLIDING WINDOW OPERATION

CLAIM OF PRIORITY

[0001] The present application claims priority from commonly owned U.S. Non-Provisional Patent Application No. 14/803,728 filed on July 20, 2015, the contents of which are expressly incorporated herein by reference in their entirety.

FIELD

[0002] The present disclosure is generally related to sliding window operations for a single-instruction-multiple-data (SIMD) architecture. More specifically, the present disclosure is related to reducing an amount of shifts and registers associated with performing the sliding window operations.

DESCRIPTION OF RELATED ART

[0003] Advances in technology have resulted in smaller and more powerful computing devices. For example, there currently exist a variety of portable personal computing devices, including wireless telephones such as mobile and smart phones, tablets and laptop computers that are small, lightweight, and easily carried by users. These devices can communicate voice and data packets over wireless networks. Further, many such devices incorporate additional functionality such as a digital still camera, a digital video camera, a digital recorder, and an audio file player. Also, such devices can process executable instructions, including software applications, such as a web browser application, that can be used to access the Internet. As such, these devices can include significant computing capabilities.

[0004] Advances in technology have also resulted in wireless computing devices that include processors that perform signal processing (e.g., image processing, video processing, audio processing, etc.). The processors apply a convolution function to input signals to perform signal processing. A conventional convolution function may use a “sliding window” to process an input data stream. A sliding window of a portion of data elements from the input data stream may be used to generate a first output that is a function (e.g., a sum of products) of each data element in the sliding window. To determine a next output, the sliding window is shifted and the next output is determined as a function of each data element in the shifted sliding window.

- 2 -

[0005] A relatively large number of registers and shifts may be used to perform convolution functions using a conventional sliding window instruction. As a non-limiting example, a single-instruction-multiple-data (SIMD) architecture may execute a conventional sliding window instruction using a four data input “lane.” To illustrate, the lane may perform a convolution function on four data inputs (e.g., a sliding window width of “4”) from a register to produce an output. To produce an additional output, the data elements may be shifted by one location in a second register and the lane may perform a convolution function on the data elements in the second register. Thus, for each output of the convolution function, multiple registers and shifts may be used.

SUMMARY

[0006] Systems and methods for reducing shifts and registers for performing sliding window operations are disclosed. For example, techniques for performing sliding window operations within a single SIMD lane by referencing multiple registers that include data for operations are disclosed. To illustrate, a sliding window operation may reference a first register and a second register. For example, a SIMD lane may reference the first register and the second register to perform sliding window operations on data in the first and second registers. As an example, a portion of the first register in the SIMD lane may store four data elements (e.g., data element “0”, data element “1”, data element “2”, and data element “3”) of an input data stream, and a portion of the second register in the SIMD lane may store the next four data elements (e.g., data element “4”, data element “5”, data element “6”, and data element “7”) of the input data stream. With respect to the portion of the first register, the corresponding portion of the second register may include shifted data of the input data stream.

[0007] The SIMD lane may reference the two registers to perform up to five sliding window operations. For example, a SIMD architecture may multiplex data in the registers to perform the sliding window operations. To illustrate, the first SIMD lane may perform a first sliding operation using the first four data elements in the first register (e.g., data element “0”, data element “1”, data element “2”, and data element “3”), a second sliding window operation using data elements “1, 2, and 3” of the first register and data element “4” of the second register, a third sliding window operation using data elements “2 and 3” of the first register and data elements “4 and 5” of the

- 3 -

second register, a fourth sliding window operation using data element “3” of the first register and data elements “4, 5, and 6” of the second register, etc.

[0008] In a particular aspect, an apparatus includes a first register having a lane storing first input data and a second register having a lane storing second input data elements. A width of the lane of the second register is equal to a width of the lane of the first register. The apparatus further includes a single-instruction-multiple-data (SIMD) lane having a lane width equal to the width of the lane of the first register. The SIMD lane is configured to perform a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register. Performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register. The second position is different from the first position.

[0009] In another particular aspect, a method includes storing first input data elements in a lane of a first register and storing second input data elements in a lane of a second register, where a width of the lane of the second register is equal to a width of the lane of the first register. The method further includes performing a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register using a single-instruction-multiple-data (SIMD) lane having a lane width equal to the width of the lane of the first register. Performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register. The second position is different from the first position.

[0010] In another particular aspect, an apparatus includes first means for storing data and second means for storing data. The first means for storing data has a lane storing first input data elements, and the second means for storing data has a lane storing second input data elements. A width of the lane of the second means for storing data is equal to a width of the lane of the first means for storing data. The apparatus further includes means for performing a sliding window operation on the first input data elements in the lane of the first means for storing data and the second input data elements in the lane of the second means for storing data. The means for performing the

- 4 -

sliding window operation has a lane width equal to the width of the lane of the first means for storing data. Performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first means for storing data and a second input data element stored in a second position of the second means for storing data. The second position is different from the first position.

[0011] In another particular aspect, a non-transitory computer-readable medium includes instructions that, when executed by a processor, cause the processor to store first input data elements in a lane of a first register and store second input data elements in a lane of a second register, where a width of the lane of the second register is equal to a width of the lane of the first register. The instructions are further executable to cause the processor to perform a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register using a single-instruction-multiple-data (SIMD) lane having a lane width equal to the width of the lane of the first register. Performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register. The second position is different from the first position.

[0012] One particular advantage provided by at least one of the disclosed implementations is that a SIMD lane may perform sliding window operations using fewer registers and shifts than a conventional SIMD architecture. Other aspects, advantages, and features of the present disclosure will become apparent after review of the entire application, including the following sections: Brief Description of the Drawings, Detailed Description, and the Claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 illustrates operations of a system to perform sliding window operations;

[0014] FIG. 2 illustrates circuitry within a single-instruction-multiple-data (SIMD) lane;

[0015] FIG. 3 is a diagram of a system that is operable to load input data into a first register and to load a shifted version of the input data into a second register;

[0016] FIG. 4 is a flow chart of a method for performing sliding window operations;
and

- 5 -

[0017] FIG. 5 is a block diagram of a device that includes components that are operable to perform sliding window operations.

DETAILED DESCRIPTION

[0018] Referring to FIG. 1, operations of a system to perform sliding window operations is shown. The system may perform the sliding window operations within a single-instruction-multiple-data (SIMD) lane by referencing multiple registers that include data for the operations. The system may include a SIMD architecture 102 (e.g., convolutional function hardware and/or sliding window hardware).

[0019] Input data 104 may be provided to the SIMD architecture 102. As described below, particular portions of the input data 104 may be shifted and stored in certain register positions to enable sliding window operations to be performed according to the techniques described herein. The input data 104 may include sixty-four data elements (e.g., data element “0” to data element “63”). In alternative implementations, the input data 104 may include additional (or fewer data elements). In the illustrative implementation, the input data 104 may include first input data elements 106 and second input data elements 108. The first input data elements 106 may include data element “0”, data element “1”, data element “2”, and data element “3”. The second input data elements 108 may include data element “4”, data element “5”, data element “6”, and data element “7”. Although not explicitly shown with respect to the input data 104, the input data 104 may also include third input data elements (e.g., data element “8” through data element “11”), fourth input data elements (e.g., data element “12” through data element “15”), fifth input data elements (e.g., data element “16” through data element “19”), etc.

[0020] The SIMD architecture 102 may be configured to apply a convolution function to the input data 104 to generate output data 110. For example, the SIMD architecture 102 may apply a “sliding window” function to data elements in the input data 104 to generate the output data 110. Each output data element of the output data 110 may be a function (e.g., a sum of products) of the input data elements in the sliding window. To illustrate, a first output data element of the output data 110 may be a sum of products of the first input data elements 106. For example, the first output data element of the output data 110 may be a function of data element “0”, data element “1”, data element “2”, and data element “3”. The second output data element of the output data 110 may

- 6 -

be a function of data element “1”, data element “2”, data element “3”, and data element “4”. Techniques for performing the sliding window operations are explained in greater detail below.

[0021] The SIMD architecture 102 may include a first SIMD lane 122, a second SIMD lane 124, a third SIMD lane 126, and a fourth SIMD lane 128. Although four SIMD lanes are depicted in FIG. 1, in alternative implementations, additional (or fewer) SIMD lanes may be included in the SIMD architecture 102. As described below, each SIMD lane 122-128 may be configured to perform sliding window operations on input data elements within the corresponding SIMD lane 122-128.

[0022] The system may also include a first register 130 and a second register 140. Each register 130, 140 may store input data elements of the input data 104. The first register 130 may include a first lane 132, a second lane 134, a third lane 136, and a fourth lane 138. In the illustrated implementation, each lane 132-138 of the first register 130 may have a width of four. For example, each lane 132-138 of the first register 130 may be allocated to store four data elements. For example, each lane 132-138 of the first register 130 may be operable to store four input data elements of the input data 104. The first lane 132 of the first register 130 may store the first input data elements 106 (e.g., data element “0” through data element “3”) of the input data 104, the second lane 134 of the first register 130 may store the second input data elements 108 (e.g., data element “4” through data element “7”) of the input data 104, the third lane 136 of the first register 130 may store the third input data elements (e.g., data element “8” through data element “11”) of the input data 104, and the fourth lane 138 of the first register 130 may store the fourth input data elements (e.g. data element “12” through data element “15”) of the input data 104.

[0023] Additionally, the width of each lane 132-138 of the first register 130 may be equal to a lane width of a corresponding SIMD lane 122-128. For example, the lane width of the first SIMD lane 122 is equal to the width of the first lane 132 of the first register 130, the lane width of the second SIMD lane 124 is equal to the width of the second lane 134 of the first register 130, the lane width of the third SIMD lane 126 is equal to the width of the third lane 136 of the first register 130, and the lane width of the fourth SIMD lane 128 is equal to the width of the fourth lane 138 of the first register 130.

- 7 -

[0024] The second register 140 may include a first lane 142, a second lane 144, a third lane 146, and a fourth lane 148. In the illustrated implementation, each lane 142-148 of the second register 140 may also have a width of four. For example, each lane 142-148 of the second register 140 may be operable to store four input data elements of the input data 104. With respect to the input data elements stored in lanes 132-138 of the first register 130, the input data elements stored in the corresponding lanes 142-148 of the second register 140 may be shifted by the lane width (e.g., shifted by four). For example, the first lane 142 of the second register 140 may store the second input data elements 108 (e.g., data element “4” through data element “7”) of the input data 104, the second lane 144 of the second register 140 may store the third input data elements (e.g., data element “8” through data element “11”) of the input data 104, the third lane 146 of the second register 140 may store the fourth input data elements (e.g., data element “12” through data element “15”) of the input data 104, and the fourth lane 148 of the second register 140 may store the fifth input data elements (e.g. data element “16” through data element “19”) of the input data 104.

[0025] Each data element may be stored in a particular position of a corresponding register. As used herein, a “position” of a register corresponds to a relative address (or location) in the register with respect to the starting address. For example, a data element stored in a first position may be stored at a location at the beginning of the register, and a data element stored in a last position may be stored at a location at the end of the register. To illustrate, with respect to the registers 130, 140, data element “0” may be stored at a first position of the first register 130 and data element “4” may be stored at a first position of the second register 140. Data element “1” and data element “5” may be stored at a second position of the first register 130 and a second position of the second register 140, respectively. Data element “2” and data element “6” may be stored at a third position of the first register 130 and a third position of the second register 140, respectively. Data element “3” and data element “7” may be stored at a fourth position of the first register 130 and a fourth position of the second register 140, respectively.

[0026] The first SIMD lane 122 may reference the first register 130 and the second register 140 to perform sliding window operations on data that is equivalent to the lane size. As described in greater detail with respect to FIG. 2, circuitry within the first

- 8 -

SIMD lane 122 may multiplex data in the registers 130, 140 to perform the sliding window operations. The first SIMD lane 122 may reference the first lanes 132, 142 of the first and second registers 130, 140, respectively, to perform up to five sliding window operations. To illustrate, the first SIMD lane 122 may perform a first sliding window operation using the first four data elements in the first register 130 (e.g., data element “0”, data element “1”, data element “2”, and data element “3”). After performing the first sliding window operation, the first SIMD lane 122 may perform a second sliding window operation using data element “1” through data element “3” in the first lane 132 of the first register 130 and using data element “4” in the first lane 142 of the second register 140.

[0027] The first SIMD lane 122 may also be configured to perform a third sliding window operation using data element “2” and data element “3” of the first lane 132 of the first register 130 and using data element “4” and data element “5” of the first lane 142 of the second register 140. After performing the third sliding window operation, the first SIMD lane 122 may perform a fourth sliding window operation using data element “3” of the first lane 132 of the first register 130 and using data element “4” through data element “6” of the first lane 142 of the second register 140. The first SIMD lane 122 may further be configured to perform a fifth sliding window operation using the first four data elements in the second register 140 (e.g., data element “4” through data element “7”).

[0028] Thus, the first SIMD lane 122 may perform five sliding window operations using two registers 130, 140. Data elements in the first lane 132 of the first register 130 and data elements in the first lane 142 of the second register 142 may be “offset” by a single shift (e.g., a four element shift) in the input data 104. As a result, the first SIMD lane 122 may have “access” to eight data elements using two registers 130, 140. Thus, two registers 130, 140 may be used to perform five sliding window operations compared to a conventional SIMD architecture that may require a single SIMD lane to use four registers and four shifts (e.g., four single-elements shifts). As shown in FIG. 1, because the data in the second register 140 is shifted by four instead of one, all of the data elements necessary to calculate up to five outputs are accessible to the first SIMD lane 122 using two registers 130, 140. Thus, a reduced number of registers and shifts may be utilized compared to other sliding window operations and SIMD architectures.

- 9 -

[0029] It will also be appreciated that the second SIMD lane 124, the third SIMD lane 126, and the fourth SIMD lane 128 may each perform up to five sliding window operations using the two registers 130, 140. The SIMD lanes 124-128 may operate in a substantially similar manner with respect to the corresponding lanes 134-138, 144-148 of the registers 130, 140, respectively, as the first SIMD lane 122 operates with respect to the first lanes 132, 142 of the registers 130, 140. For example, the second SIMD lane 124 may perform up to five sliding window operations using the second input data elements 108 in the second lane 134 of the first register 130 and using the third input data elements in the second lane 144 of the second register 140. The third SIMD lane 126 may perform up to five sliding window operations using the third input data elements in the third lane 136 of the first register 130 and using the fourth input data elements in the third lane 146 of the second register 140. Additionally, the fourth SIMD lane 128 may perform up to five sliding window operations using the fourth input data elements in the fourth lane 138 of the first register 130 and using the fifth input data elements in the fourth lane 148 of the second register 140. Additional SIMD lanes (not shown) may operate in a substantially similar manner as the illustrated SIMD lanes 122-128 to perform sliding window operations on the remaining input data 104.

[0030] Referring to FIG. 2, circuitry within the first SIMD lane 122 is shown. The first SIMD lane 122 includes a first multiplexer 202, a second multiplexer 204, a third multiplexer 206, a fourth multiplexer 208, and an arithmetic and logic unit (ALU) 210. In a particular implementation, the ALU 210 may include sum of products circuitry. The circuitry within the first SIMD lane 122 is for illustrative purposes and is not intended to be limiting. For example, additional (or fewer) circuit components may be included in the first SIMD lane 122. Although circuitry within the first SIMD lane 122 is depicted in FIG. 2, the other SIMD lanes 124-128 of the system of FIG. 1 may include similar circuit components.

[0031] A first input of the first multiplexer 202 may be coupled to data element "0" stored in the first lane 132 of the first register 130, and a second input of the first multiplexer 202 may be coupled to data element "4" stored in the first lane 142 of the second register 140. Based on a selection signal (not shown) provided to the first multiplexer 202, the first multiplexer 202 may provide data element "0" or data element "4" to the ALU 210. In a particular aspect, an immediate field in an operational code of

- 10 -

an instruction may determine whether the multiplexer 202 provides data element “0” (e.g., a data element in a first window in the first lane 142) or data element “4” (e.g., a data element in a first window in the first lane 142) to the ALU 210.

[0032] A first input of the second multiplexer 204 may be coupled to data element “1” stored in the first lane 132 of the first register 130, and a second input of the second multiplexer 204 may be coupled to data element “5” stored in the first lane 142 of the second register 140. Based on a selection signal (not shown) provided to the second multiplexer 204, the second multiplexer 204 may provide data element “1” or data element “5” to the ALU 210. In a particular aspect, the immediate field in the operational code of the instruction may determine whether the multiplexer 204 provides data element “1” (e.g., a data element in a second window in the first lane 142) or data element “5” (e.g., a data element in a second window in the first lane 142) to the ALU 210.

[0033] A first input of the third multiplexer 206 may be coupled to data element “2” stored in the first lane 132 of the first register 130, and a second input of the third multiplexer 206 may be coupled to data element “6” stored in the first lane 142 of the second register 140. Based on a selection signal (not shown) provided to the third multiplexer 206, the third multiplexer 206 may provide data element “2” or data element “6” to the ALU 210. In a particular aspect, the immediate field in the operational code of the instruction may determine whether the multiplexer 206 provides data element “2” (e.g., a data element in a third window in the first lane 142) or data element “6” (e.g., a data element in a third window in the first lane 142) to the ALU 210.

[0034] Additionally, a first input of the fourth multiplexer 208 may be coupled to data element “3” stored in the first lane 132 of the first register 130, and a second input of the fourth multiplexer 208 may be coupled to data element “7” stored in the first lane 142 of the second register 140. Based on a selection signal (not shown) provided to the fourth multiplexer 208, the fourth multiplexer 208 may provide data element “3” or data element “7” to the ALU 210. In a particular aspect, the immediate field in the operational code of the instruction may determine whether the multiplexer 208 provides data element “3” (e.g., a data element in a fourth window in the first lane 142) or data element “7” (e.g., a data element in a fourth window in the first lane 142) to the ALU

- 11 -

210. Thus, the first SIMD lane 122 may simultaneously perform sliding window operations on a plurality of windows in the first lane 132 of the first register 130, a plurality of windows in the first lane 142 of the second register 140, or any combination thereof, based on the instruction.

[0035] During operation, the first SIMD lane 122 may perform up to five sliding window operations based on data in the first lanes 132, 142 of the registers 130, 140. The following example corresponds to the third sliding window operation (e.g., a sum of products operations using data element “2”, data element “3”, data element “4”, and data element “5”); however, similar techniques may be applied to perform the first sliding window operation, the second sliding window operation, the fourth sliding window operation, and the fifth sliding window operation.

[0036] To perform the third sliding window operation, the first multiplexer 202 may provide data element “4” to the ALU 210 based on the selection signal provided to the first multiplexer 202, and the second multiplexer 204 may provide data element “5” to the ALU 210 based on the selection signal provided to the second multiplexer 204. Additionally, the third multiplexer 206 may provide data element “2” to the ALU 210 based on the selection signal provided to the third multiplexer 206, and the fourth multiplexer 208 may provide data element “3” to the ALU 210 based on the selection signal provided to the fourth multiplexer 208.

[0037] The ALU 210 may be configured to generate output data 110 based on the sum of products of data element “2”, data element “3”, data element “4”, and data element “5”. For example, the ALU 210 may perform a vertical operation (e.g., a multiplication operation) on the data elements provided by the multiplexers 202-208. As a non-limiting example of the multiplication operation, the ALU 210 may multiply data element “2” with data element “3” to generate a first product, multiply data element “3” with data element “4” to generate a second product, multiply data element “4” with data element “5” to generate a third product, and multiply data element “5” with data element “2” to generate a fourth product. After the vertical operation, the ALU 210 may perform a horizontal operation on the products resulting from the vertical operation. Non-limiting examples of a horizontal operation may include a summation operation, a bitwise OR operation, or a multiplication operation. As a non-limiting example of the summation operation, the ALU 210 may sum the first product, the second product, the

- 12 -

third product, and the fourth product. Thus, the horizontal operation may be performed after the vertical operation. In the above example, the horizontal operation and the vertical operation correspond to a sum-of-products.

[0038] Although FIG. 2 depicts the ALU 210 as sum of products circuitry, in alternative implementations, ALUs may be implemented with the first SIMD lane 122 to perform sliding window operations. As non-limiting examples, an ALU that performs a product of sums operation, a logical AND operation, a logical OR operation, a logical NAND operation, or a logical NOR operation, may be implemented with the first SIMD lane 122.

[0039] Thus, the circuitry within the first SIMD lane 122 may perform up to five sliding window operations using two registers 130, 140 and a single shift (e.g., a four element shift between the registers 130, 140) compared to a SIMD architecture that may require a single SIMD lane to use four registers and four shifts (e.g., four single-elements shifts). As shown in FIG. 2, because the data in the second register 140 is shifted by four instead of one, all of the data elements necessary to calculate up to five outputs are accessible to the first SIMD lane 122 using two registers 130, 140. Thus, a reduced number of registers and shifts may be utilized compared to other sliding window operations and SIMD architectures.

[0040] Referring to FIG. 3, a system 300 that is operable to load the input data 104 into the first register 130 and to load a shifted version of the input data 104 into the second register 140 is shown. The system 300 includes a processor 301, a memory 302, the first register 130, and the second register 140. The processor 301 includes a logic shifter circuit 304.

[0041] Upon receiving a load instruction, the processor 301 may retrieve the input data 104 from a memory location in the memory 302 and may load the input data 104 into the first register 130. The processor 301 may load the input data 104 into the first register 130 according to an alignment depicted in FIG. 1. For example, data element “0” through data element “3” may be loaded into the first lane 132 of the first register 130, data element “4” through data element “7” may be loaded into the second lane 134 of the first register 130, etc.

- 13 -

[0042] The processor 301 may also retrieve the input data 104 from the memory location in the memory 302 and provide the input data 104 to the logic shifter circuit 304 upon receiving the load instruction. The logic shifter circuit 304 may be configured to shift the input data 104 by the lane width of the SIMD lanes 122-128. For example, the logic shifter circuit 304 may shift the input data 104 by four to generate shifted input data 314. The processor 301 may load the shifted input data 314 into the second register 140 according to the alignment depicted in FIG. 1. For example, data element “4” through data element “7” may be loaded into the first lane 142 of the second register 140, data element “8” through data element “11” may be loaded into the second lane 144 of the second register 140, etc.

[0043] Although the logic shifter circuit 304 is depicted to be included in the processor 301, in other implementations, the logic shifter circuit 304 may be external to the processor 301. In a particular implementation, the processor 301 may be a central processing unit (CPU) of a device (e.g., a mobile phone). In alternate implementations, the processor 301 may be external to the CPU (e.g., the processor 301 may be an application-specific integrated circuit).

[0044] The system 300 of FIG. 3 may provide a shift during a load operation to align the data elements in the registers 130, 140 as depicted in FIG. 1. For example, the logic shifter circuit 304 may shift the input data 104 such that the first lane 142 of the second register 140 stores data element “4” through data element “7” (as opposed to data element “0” through data element “3”). The shift may enable the first SIMD lane 122 to perform up to five sliding window operations using the two registers 130, 140, as described with respect to FIGS. 1-2.

[0045] Referring to FIG. 4, a flowchart of a method 400 for performing a sliding window operation is shown. The method 400 may be performed by the SIMD architecture 102 of FIG. 1 (e.g., the SIMD lanes 122-128) and/or the processor 301 of FIG. 3 using the first register 130 of FIGS. 1-3, the second register 140 of FIGS. 1-3, the output register 212 of FIG. 2, the memory 302 of FIG. 3, or any combination thereof.

[0046] The method 400 includes storing first input data elements in a lane of a first register, at 402. For example, referring to FIGS. 1-3, the processor 301 may store the first input data elements 106 (e.g., data element “0” through data element “3”) of the

- 14 -

input data 104 into the first lane 132 of the first register 130. To illustrate, the processor 301 may load the first input data elements 106 into the first lane 132 of the first register 130 in response to receiving the load instruction. Additionally, the processor 301 may store the second input data elements 108 (e.g., data element “4” through data element “7”) of the input data 104 into the second lane 124 of the first register 130. To illustrate, the processor 301 may load the second input data elements 108 into the second lane 134 of the first register 130 in response to receiving the load instruction.

[0047] Second input data elements may be stored in a lane of a second register, at 404. For example, referring to FIGS. 1-3, the processor 301 may store the second input data elements 108 of the input data into the first lane 142 of the second register 140. To illustrate, the processor 301 may retrieve the input data 104 from the memory location in the memory 302 and provide the input data 104 to the logic shifter circuit 304 upon receiving the load instruction. The logic shifter circuit 304 may shift the input data 104 by the lane width of the SIMD lanes 122-128. For example, the logic shifter circuit 304 may shift the input data 104 by four to generate shifted input data 314. The processor may load the shifted input data 314 into the second register 140 according to the alignment depicted in FIG. 1. For example, data element “4” through data element “7” may be loaded into the first lane 142 of the second register 140. The width of the first lane 142 of the second register 140 may be equal to the width of the first lane 132 of the first register 130.

[0048] A sliding window operation may be performed on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register using a SIMD lane having a lane width equal to the width of the lane of the first register, at 406. As used herein, a “sliding window operation” may include any operation in which multiple neighboring data elements (e.g., the first input data elements and the second input data elements) in a data stream are taken as data inputs and an output is produced. Although a sum of products is described above as a sliding window operation, it should be understood that a sum of products is merely one example. Other examples of sliding window operations may include a product of sums, summation operations, multiplication operations, etc. Performing the sliding window operation may include determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second

- 15 -

position of the second register that is different from the first position. As a non-limiting example, referring to FIGS. 1-2, the result of one sliding window operation may be based on data elements “1-4”. Data element “1” may be in one position of the first register 130 (e.g., the second position) and data element “4” may be in another position of the second register 140 (e.g., the first position) that is different from the position of data element “1”. As described above, the first SIMD lane 122 may perform up to five sliding window operations (e.g., five sliding window operations) using the first input data elements 106 stored in the first lane 132 of the first register 130 and using the second input data elements 108 stored in the first lane 142 of the second register 140. Thus, the number of sliding window operations (e.g., five) performed by the first SIMD lane 122 may be greater than or equal to the width (e.g., four) of the first SIMD lane 122.

[0049] The method 400 of FIG. 4 may enable the first SIMD lane 122 to perform up to five sliding window operations using two registers 130, 140 and a single shift (e.g., a four data element shift between the registers 130, 140 by the logic shifter circuit 304 as depicted in FIG. 3) compared to another SIMD architecture that may require a single SIMD lane to use four registers and four shifts (e.g., four single-elements shifts). Because the data in the second register 140 is shifted by four instead of one, all of the data elements necessary to calculate up to five outputs are accessible to the first SIMD lane 122 using two registers 130, 140. Thus, a reduced number of registers and shifts may be utilized compared to other sliding window operations and SIMD architectures.

[0050] In a particular implementation, the method 400 may include performing similar operations at additional SIMD lanes. For example, the SIMD lanes 124-128 may operate in a substantially similar manner with respect to the corresponding lanes 134-138, 144-148 of the registers 130, 140, respectively, as the first SIMD lane 122 operates with respect to the first lanes 132, 142 of the registers 130, 140. For example, the second SIMD lane 124 may perform up to five sliding window operations using the second input data elements 108 in the second lane 134 of the first register 130 and using the third input data elements in the second lane 144 of the second register 140. The third SIMD lane 126 may perform up to five sliding window operations using the third input data elements in the third lane 136 of the first register 130 and using the fourth input data elements in the third lane 146 of the second register 140. Additionally, the

- 16 -

fourth SIMD lane 128 may perform up to five sliding window operations using the fourth input data elements in the fourth lane 138 of the first register 130 and using the fifth input data elements in the fourth lane 148 of the second register 140. Additional SIMD lanes (not shown) may operate in a substantially similar manner as the illustrated SIMD lanes 122-128 to perform sliding window operations on the remaining input data 104.

[0051] Referring to FIG. 5, a wireless communication device is depicted and generally designated 500. The device 500 includes a processor 510, such as a digital signal processor, coupled to a memory 532.

[0052] The processor 510 may be configured to execute software (e.g., a program of one or more instructions 568) stored in the memory 532. The processor 510 may include the SIMD architecture 102 of FIG. 1, the first register 130 of FIG. 1, and the second register 140 of FIG. 1. For example, the SIMD architecture 102 may be within an execution unit of the processor 510 and may reference the registers 130, 140 to perform convolution functions, as described above. In a particular implementation, the processor 510 may be operable to perform the method 400 of FIG. 4. For example, the processor 510 (e.g., the SIMD architecture 102) may store the first input data elements 106 in the first lane 132 of the first register 130 and may store the second input data elements 108 in the first lane 142 of the second register 140. The SIMD architecture 102 may perform up to five sliding window operations using the data elements stored in the first lanes 132, 142 of the registers 130, 140, respectively, as described above.

[0053] In a particular implementation, the processor 510 may be configured to execute one or more instructions 568 stored in the memory 532 to perform the method 400 of FIG. 4. For example, the memory 532 may be a non-transitory computer-readable medium that includes instructions 568 that, when executed by the processor 510, cause the processor 510 to perform the method 400 of FIG. 4.

[0054] A wireless interface 540 may be coupled to the processor 510 and to an antenna 542. A coder/decoder (CODEC) 534 can also be coupled to the processor 510. A speaker 536 and a microphone 538 can be coupled to the CODEC 534. A display controller 526 can be coupled to the processor 510 and to a display device 528. In a particular implementation, the processor 510, the display controller 526, the memory 532, the CODEC 534, and the wireless interface 540 are included in a system-in-

- 17 -

package or system-on-chip device 522. In a particular implementation, an input device 530 and a power supply 544 are coupled to the system-on-chip device 522. Moreover, in a particular implementation, as illustrated in FIG. 5, the display device 528, the input device 530, the speaker 536, the microphone 538, the antenna 542, and the power supply 544 are external to the system-on-chip device 522. However, each of the display device 528, the input device 530, the speaker 536, the microphone 538, the antenna 542, and the power supply 544 can be coupled to one or more components of the system-on-chip device 522, such as one or more interfaces or controllers.

[0055] The convolution functions with respect to FIGS. 1-4 may be used to perform image processing at the device 500. For example, the convolution functions may be applied to a kernel (e.g., a convolution matrix or a mask) to blur pixels of an image, to sharpen pixels of an image, to detect edges within an image, etc. Thus, the device 500 may perform image processing using fewer registers and shifts compared to devices using other SIMD architectures.

[0056] In conjunction with the described implementations, an apparatus includes first means for storing data. The first means for storing data has a lane storing first input data element. For example, the first means for storing data may include the first register 130 of FIGS. 1-3 and 5, one or more other devices, circuits, modules, or any combination thereof.

[0057] The apparatus may also include second means for storing data. The second means for storing data has a lane storing second input data elements. For example, the second means for storing data may include the second register 140 of FIGS. 1-3 and 5, one or more other devices, circuits, modules, or any combination thereof. A width of the first lane of the second means for storing data is equal to a width of the first lane of the first means for storing data. For example, the width (e.g., four) of the first lane 132 of the first register 130 may be equal to the width (e.g., four) of the first lane 142 of the second register 140.

[0058] The apparatus may also include means for a performing sliding window operation on the first input data elements in the lane of the first means for storing data and the second input data elements in the lane of the second means for storing data. For example, the means for performing the sliding window operations may include the SIMD architecture 102 of FIGS. 1 and 5, the multiplexers 202-208 of FIG. 2, the

- 18 -

processor 301 of FIG. 3, the instructions 568 executable by the processor 510 of FIG. 5, one or more other devices, circuits, modules, or any combination thereof. The means for performing the sliding window operations has a lane width equal to the width of the lane of the first means for storing data. Performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first means for storing data and a second input data element stored in a second position of the second means for storing data. The second position is different from the first position.

[0059] Those of skill would further appreciate that the various illustrative logical blocks, configurations, modules, circuits, and algorithm steps described in connection with the implementations disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. Various illustrative components, blocks, configurations, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0060] The steps of a method or algorithm described in connection with the implementations disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in random access memory (RAM), flash memory, read-only memory (ROM), programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), registers, hard disk, a removable disk, a compact disc read-only memory (CD-ROM), or any other form of storage medium known in the art. An exemplary non-transitory (e.g. tangible) storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an application-specific integrated circuit (ASIC). The ASIC may reside in a computing device or a user terminal. In the alternative, the processor and the

- 19 -

storage medium may reside as discrete components in a computing device or user terminal.

[0061] The previous description of the disclosed implementations is provided to enable a person skilled in the art to make or use the disclosed implementations. Various modifications to these implementations will be readily apparent to those skilled in the art, and the principles defined herein may be applied to other implementations without departing from the scope of the disclosure. Thus, the present disclosure is not intended to be limited to the implementations shown herein but is to be accorded the widest scope possible consistent with the principles and novel features as defined by the following claims.

WHAT IS CLAIMED IS:

1. An apparatus comprising:
 - a first register having a lane storing first input data elements;
 - a second register having a lane storing second input data elements, wherein a width of the lane of the second register is equal to a width of the lane of the first register; and
 - a single-instruction-multiple-data (SIMD) lane having a lane width equal to the width of the lane of the first register, wherein the SIMD lane is configured to perform a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register, wherein performing the sliding window operation includes determining a result based a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register, the second position different from the first position.
2. The apparatus of claim 1, wherein the SIMD lane simultaneously performs sliding window operations on a plurality of windows in the lane of the first register, a plurality of windows in the lane of the second register, or any combination thereof, based on an instruction.
3. The apparatus of claim 2, wherein the plurality of windows in the lane of the first register and the plurality of windows in the lane of the second register are selected based on an immediate field in an operational code of the instruction.
4. The apparatus of claim 1, wherein the SIMD lane comprises:
 - an arithmetic and logic unit (ALU);
 - a first multiplexer configured to provide one of a first input data element in the lane of the first register to the ALU or a first input data element in the lane of the second register to the ALU; and
 - a second multiplexer configured to provide one of a second input data element in the lane of the first register to the ALU or a second input data element in the lane of the second register to the ALU,

- 21 -

wherein the ALU is operable to perform a horizontal operation on a data element provided by the first multiplexer and a data element provided by the second multiplexer.

5. The apparatus of claim 4, wherein the horizontal operation comprises a summation operation, a bitwise OR operation, or a multiplication operation.

6. The apparatus of claim 4, wherein the ALU is operable to perform a vertical operation on the data element provided by the first multiplexer and on the data element provided by the second multiplexer, and wherein the horizontal operation is performed after the vertical operation.

7. The apparatus of claim 6, wherein the vertical operation comprises a multiplication operation.

8. The apparatus of claim 1, further comprising a plurality of lanes including the SIMD lane and a second SIMD lane, the second SIMD lane configured to perform a sliding window operation on input data elements stored in a second lane of the first register and input data elements stored in a second lane of the second register.

9. The apparatus of claim 8, wherein the sliding window operation performed by the SIMD lane and the sliding window operation performed by the second SIMD lane are performed in parallel.

10. The apparatus of claim 1, further comprising a logic shifter circuit configured to shift input data and to provide shifted data to the second register.

11. A method comprising:

storing first input data elements in a lane of a first register;

storing second input data elements in a lane of a second register, wherein a width of the lane of the second register is equal to a width of the lane of the first register; and

performing a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of

- 22 -

the second register using a single-instruction-multiple-data (SIMD) lane having a lane width equal to the width of the lane of the first register, wherein performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register, the second position different from the first position.

12. The method of claim 11, wherein the first input data elements include N data elements starting at a first data element and ending at an N^{th} data element, wherein the second input data elements include N data elements starting at an $(N+1)^{\text{th}}$ data element and ending at a $2*N^{\text{th}}$ data element, and wherein N corresponds to the lane width of the SIMD lane.

13. The method of claim 11, wherein the SIMD lane simultaneously performs sliding window operations on a plurality of windows in the lane of the first register, a plurality of windows in the lane of the second register, or any combination thereof, based on an instruction.

14. The method of claim 13, wherein the plurality of windows in the lane of the first register and the plurality of windows in the lane of the second register are selected based on an immediate field in an operational code of the instruction.

15. The method of claim 11, further comprising:

providing, at a first multiplexer, one of a first data element in the lane of the first register to an arithmetic and logic unit (ALU) or a first input data element in the lane of the second register to the ALU; and
providing, at a second multiplexer, one of a second data element in the lane of the first register to the ALU or a second input data element in the lane of the second register to the ALU.

- 23 -

16. The method of claim 15, performing, at the ALU, a horizontal operation on a data element provided by the first multiplexer and a data element provided by the second multiplexer.

17. The method of claim 16, wherein the horizontal operation comprises a summation operation, a bitwise OR operation, or a multiplication operation.

18. The method of claim 16, further comprising performing a vertical operation on the data element provided by the first multiplexer and on the data element provided by the second multiplexer, and wherein the horizontal operation is performed after the vertical operation.

19. The method of claim 18, wherein the vertical operation comprises a multiplication operation.

20. The method of claim 11, further comprising performing a sliding window operation on second input data elements stored in a second lane of the first register and third input data elements stored in a second lane of the second register using a second SIMD lane of a plurality of SIMD lanes, wherein the SIMD lane is included in the plurality of SIMD lanes.

21. The method of claim 20, wherein the sliding window operation performed by the SIMD lane and the sliding window operation performed by the second SIMD lane are performed in parallel.

22. The method of claim 11, further comprising:
shifting input data; and
providing shifted input data to the second register.

23. An apparatus comprising:
first means for storing data, the first means for storing data having a lane storing first input data elements;
second means for storing data, the second means for storing data having a lane storing second input data elements, wherein a width of the lane of the

- 24 -

second means for storing data is equal to a width of the lane of the first means for storing data; and
means for performing a sliding window operation on the first input data elements in the lane of the first means for storing data and the second input data elements in the lane of the second means for storing data, the means for performing the sliding window operation having a lane width equal to the width of the lane of the first means for storing data, wherein performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first means for storing data and a second input data element stored in a second position of the second means for storing data, the second position different from the first position.

24. The apparatus of claim 23, wherein the first input data elements include N data elements starting at a first data element and ending at an N^{th} data element, wherein the second input data elements include N data elements starting at an $(N+1)^{\text{th}}$ data element and ending at a $2*N^{\text{th}}$ data element, and wherein N corresponds to the lane width of the means for performing the sliding window operation.

25. The apparatus of claim 23, wherein the means for performing the sliding window operation simultaneously performs sliding window operations on a plurality of windows in the lane of the first means for storing data, a plurality of windows in the lane of the second means for storing data, or any combination thereof, based on an instruction.

26. The apparatus of claim 25, wherein the plurality of windows in the lane of the first means for storing data and the plurality of windows in the lane of the second means for storing data are selected based on an immediate field in an operational code of the instruction.

27. The apparatus of claim 23, further comprising:
means for providing a data element in the lane of the first means for storing data to an arithmetic and logic unit (ALU); and

- 25 -

means for providing a data element in the lane of the second means for storing data to the ALU.

28. The apparatus of claim 23, further comprising second means for performing a sliding window operation on second input data elements stored in a second lane of the first means for storing data and third input data elements stored in a second lane of the second means for storing data.

29. A non-transitory computer-readable medium comprising instructions that, when executed by a processor, cause the processor to:

store first input data elements in a lane of a first register;

store second input data elements in a lane of a second register, wherein a width of the lane of the second register is equal to a width of the lane of the first register; and

perform a sliding window operation on the first input data elements in the lane of the first register and the second input data elements in the lane of the second register using a single-instruction-multiple-data (SIMD) lane having a lane width equal to the width of the lane of the first register, wherein performing the sliding window operation includes determining a result based on a first input data element stored in a first position of the first register and a second input data element stored in a second position of the second register, the second position different from the first position.

30. The non-transitory computer-readable medium of claim 29, wherein the first input data elements include N data elements starting at a first data element and ending at an Nth data element, wherein the second input data elements include N data elements starting at an (N+1)th data element and ending at a 2*Nth data element, and wherein N corresponds to the lane width of the SIMD lane.

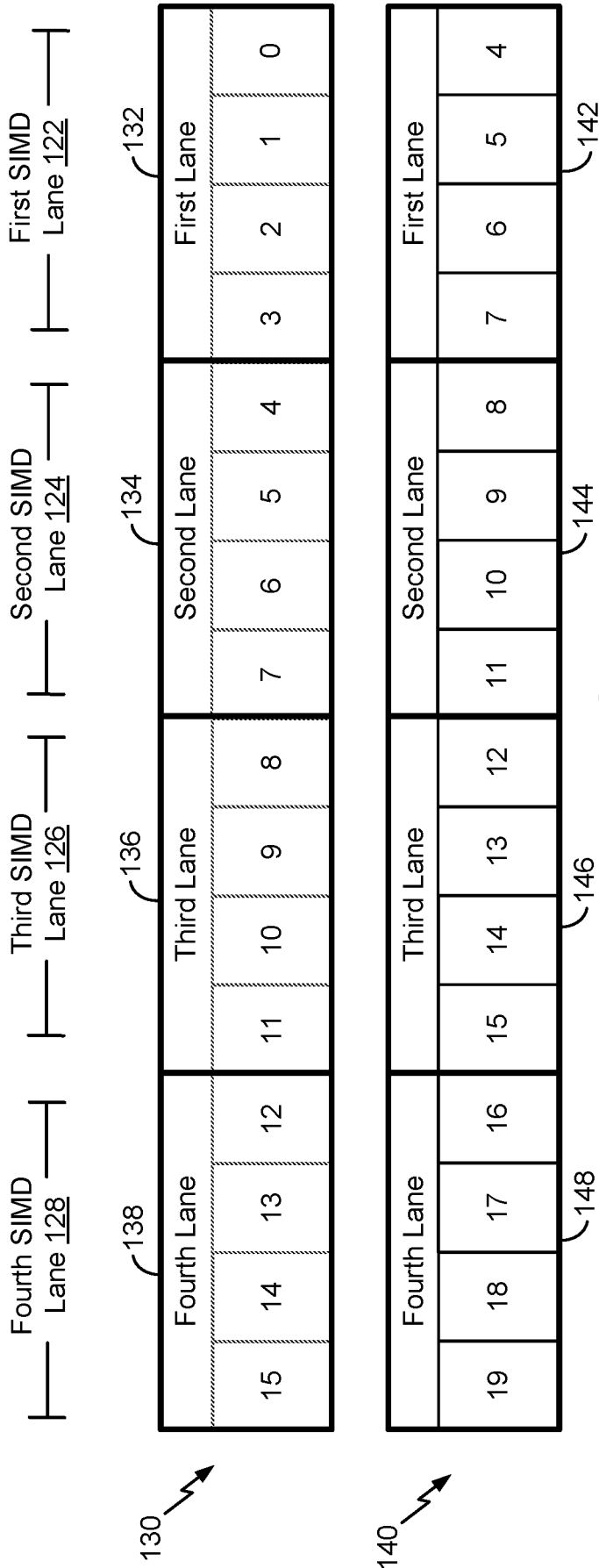
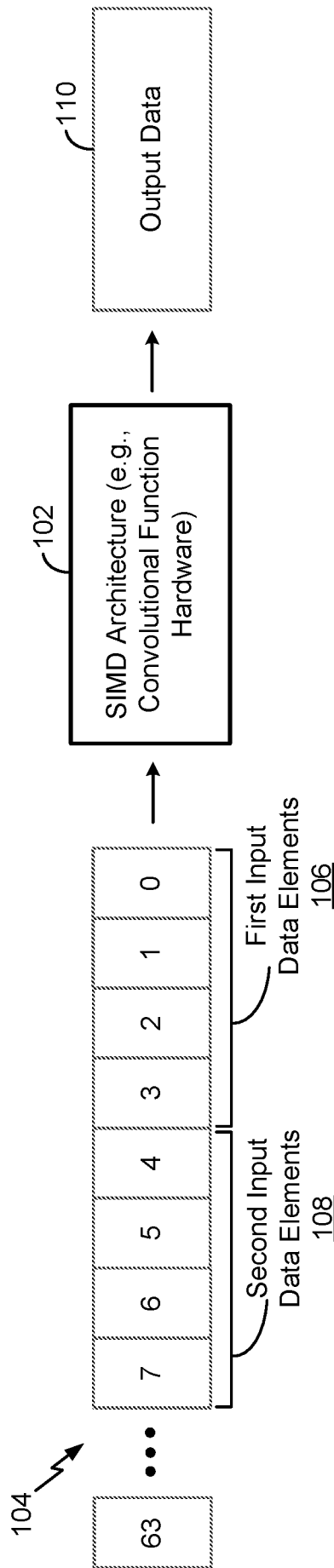


FIG. 1

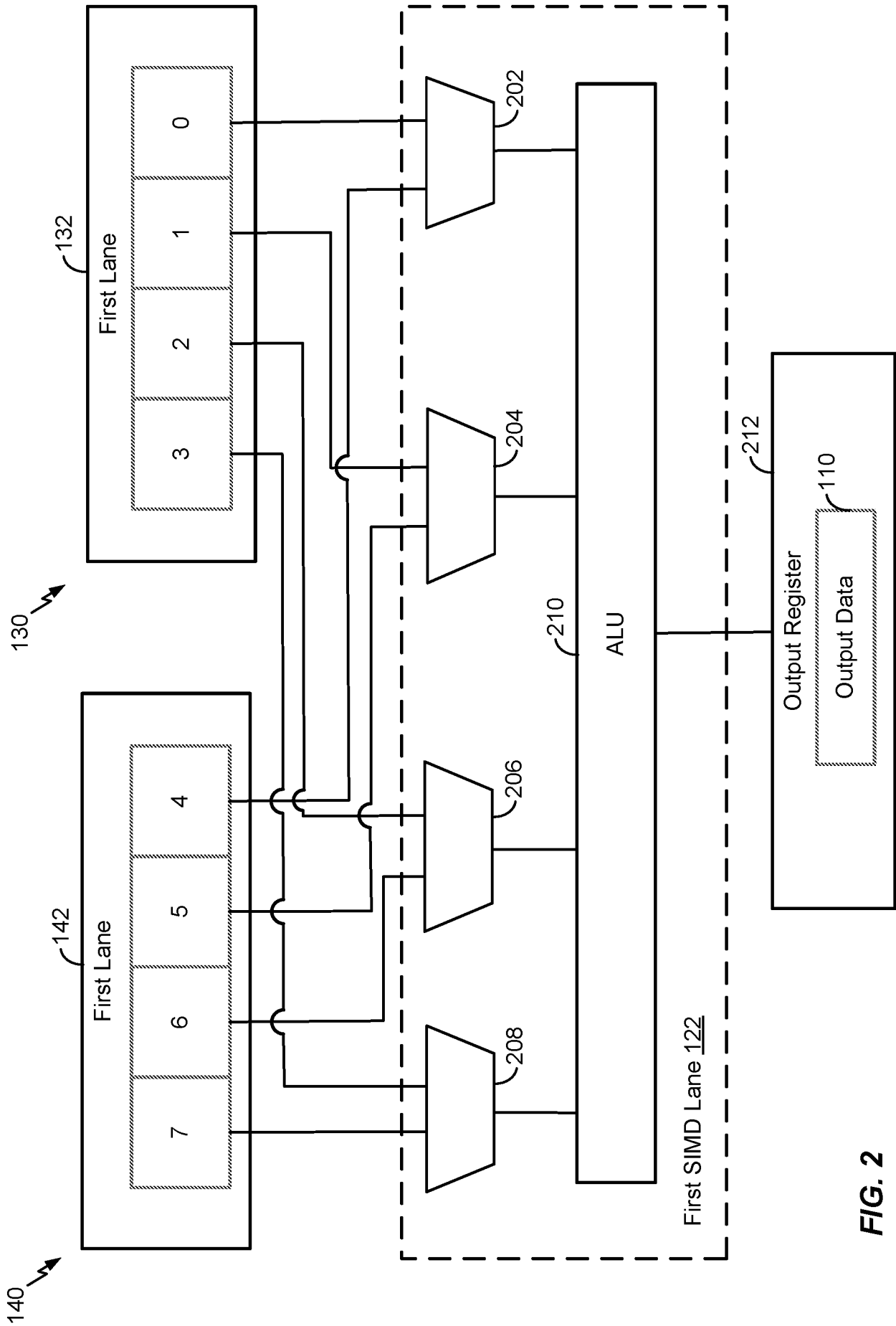


FIG. 2

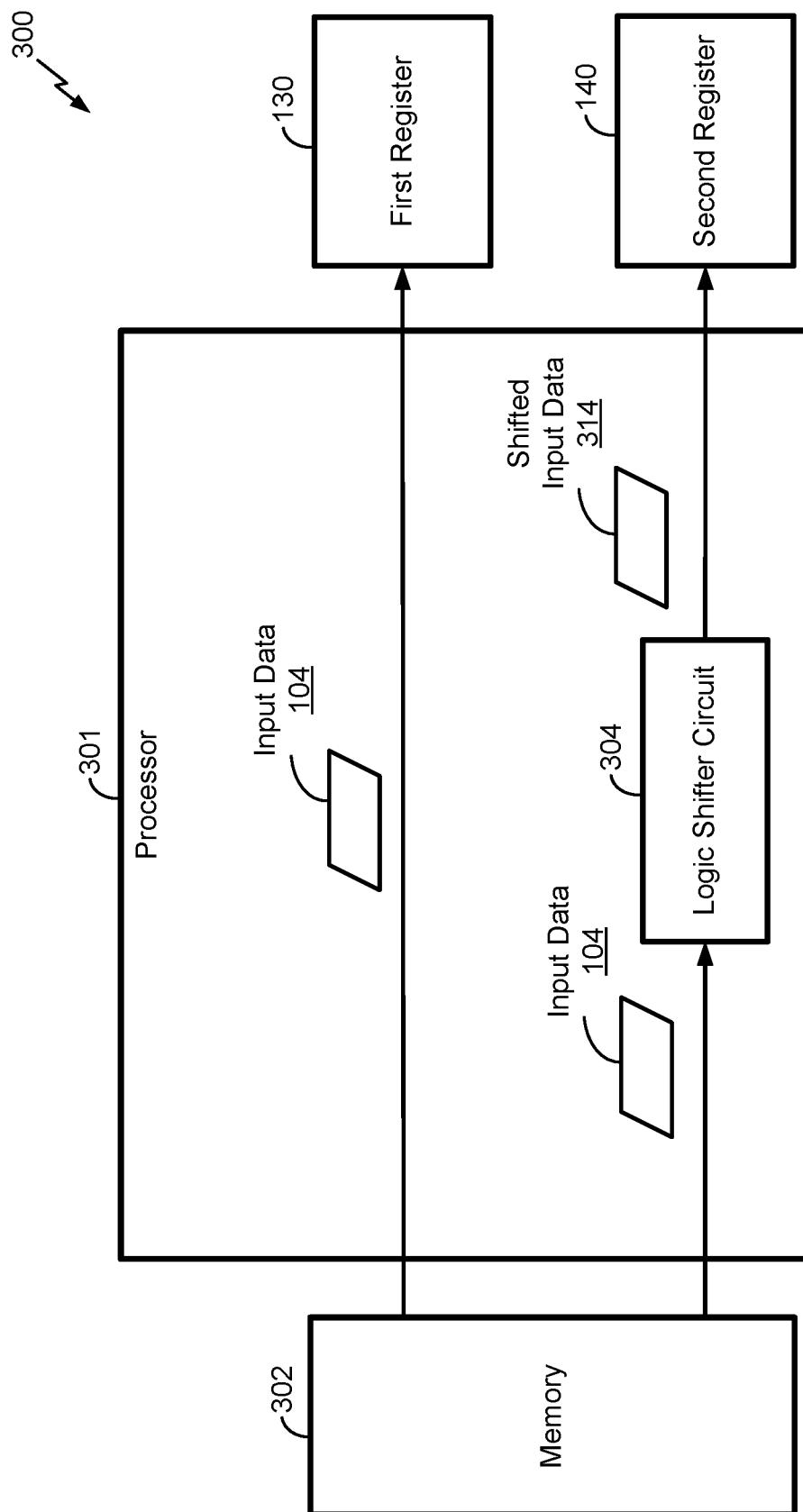
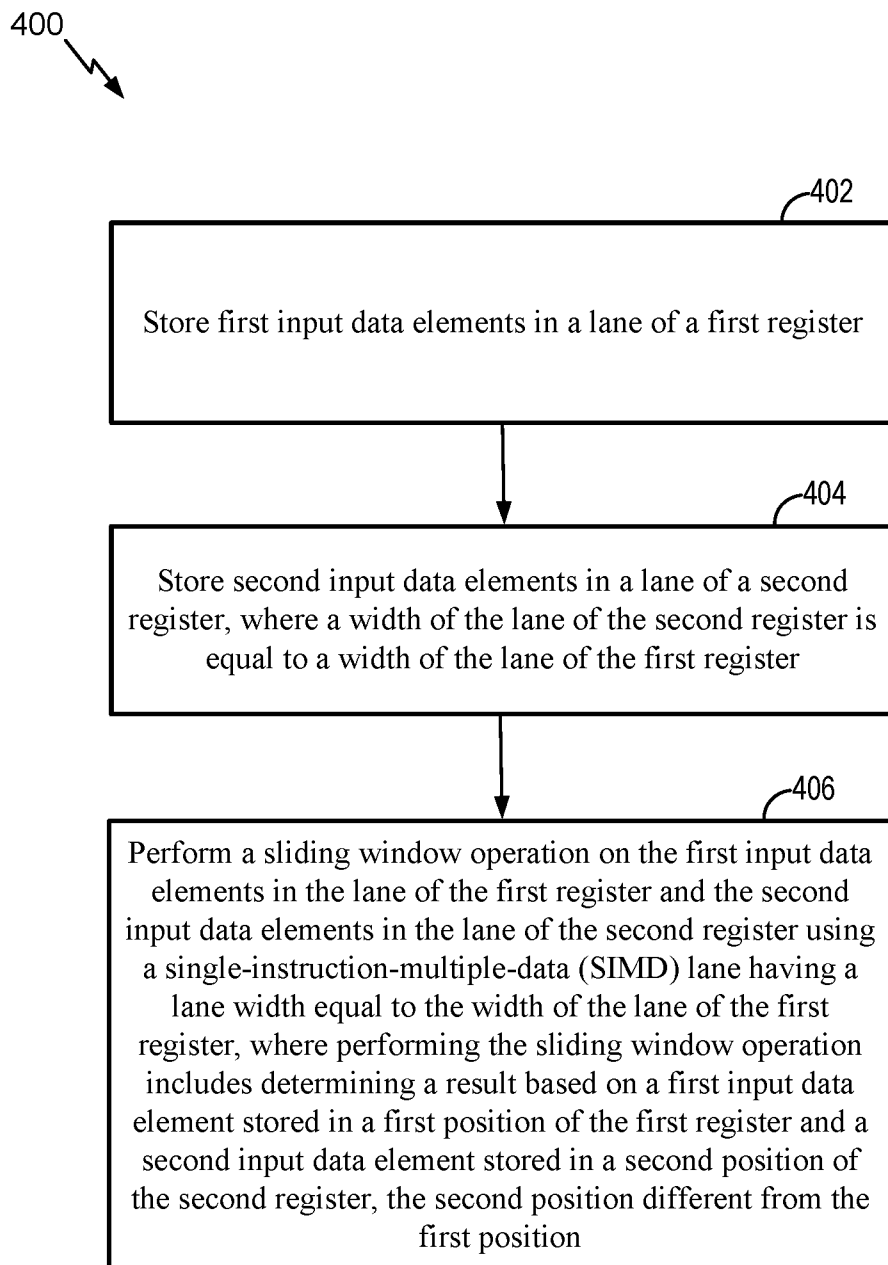


FIG. 3

**FIG. 4**

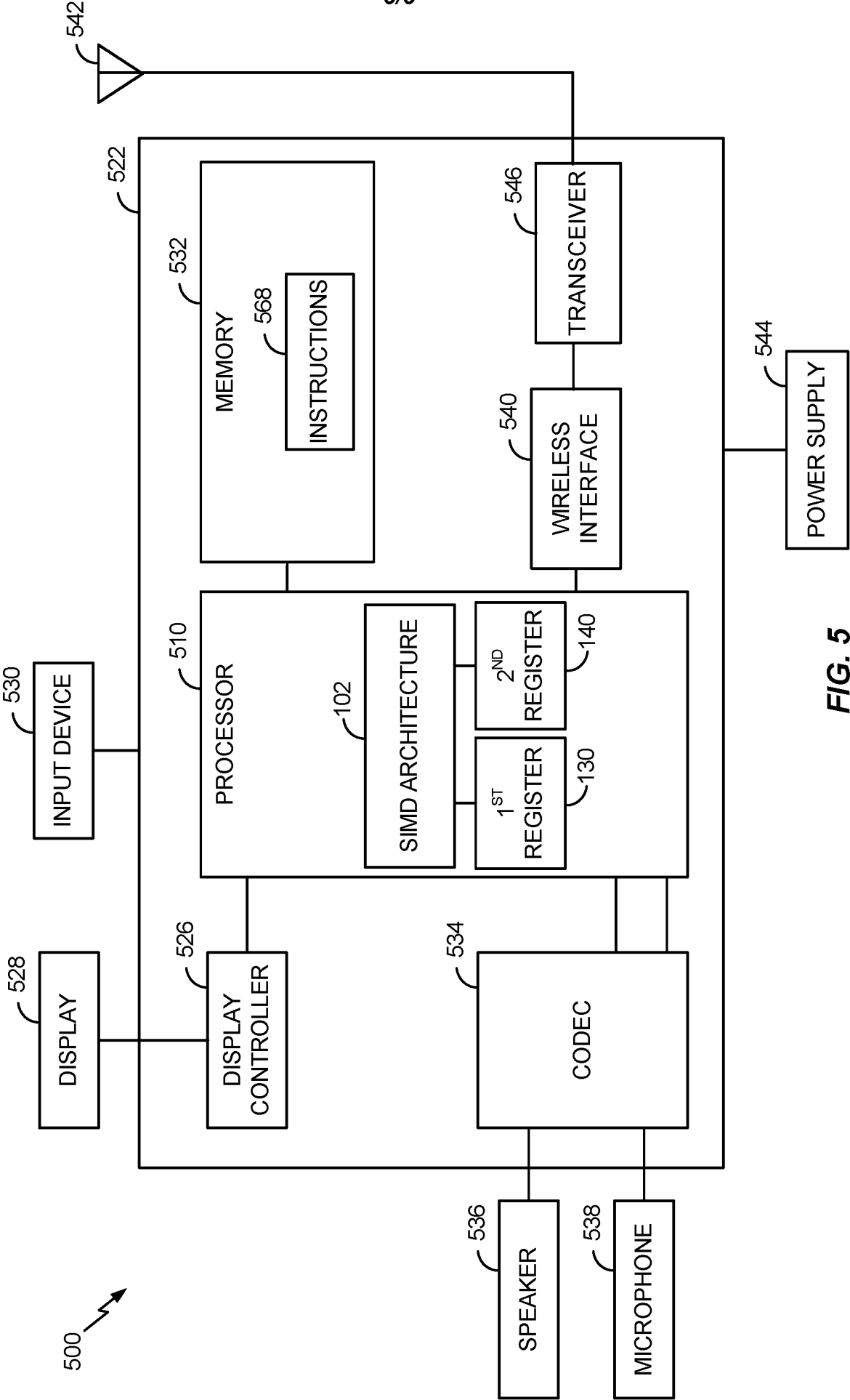


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/041779

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/30
ADD.

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)
G06F

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 009 505 A (THAYER JOHN S [US] ET AL) 28 December 1999 (1999-12-28) column 8, line 51 - column 9, line 34 column 15, line 66 - column 16, line 32 column 17, line 23 - column 18, line 65 column 24, line 17 - line 56 figures 9, 12	1-30
A	----- WO 2006/106342 A2 (ICERA INC [GB]; KNOWLES SIMON [GB]) 12 October 2006 (2006-10-12) page 18, line 6 - page 19, line 18 -----	1



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

"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

21 October 2016

Date of mailing of the international search report

31/10/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Daskalakis, T

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2016/041779

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
US 6009505	A	28-12-1999	US 6009505 A	28-12-1999
			US 6047372 A	04-04-2000

WO 2006106342	A2	12-10-2006	BR PI0609742 A2	18-10-2011
			CA 2600744 A1	12-10-2006
			CN 101208658 A	25-06-2008
			CN 103744639 A	23-04-2014
			EP 1866744 A2	19-12-2007
			JP 5047944 B2	10-10-2012
			JP 2008535115 A	28-08-2008
			KR 20070118623 A	17-12-2007
			US 2006227966 A1	12-10-2006
			WO 2006106342 A2	12-10-2006
