

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

(19) World Intellectual Property  
Organization  
International Bureau



(10) International Publication Number

WO 2017/222798 A1

(43) International Publication Date  
28 December 2017 (28.12.2017)

(51) International Patent Classification:  
*G06F 9/30* (2006.01)      *G06F 9/345* (2006.01)  
*G06F 9/38* (2006.01)      *G06F 12/00* (2006.01)

(72) Inventors: **MAHURIN, Eric Wayne**; QUALCOMM INCORPORATED, 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **GOLAB, Jakub Pawal**; QUALCOMM INCORPORATED, 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **CODRESCU, Lucian**; QUALCOMM INCORPORATED, 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(21) International Application Number:  
PCT/US2017/036041

(22) International Filing Date:  
06 June 2017 (06.06.2017)

(25) Filing Language:  
English

(26) Publication Language:  
English

(30) Priority Data:  
15/192,992      24 June 2016 (24.06.2016)      US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US];  
ATTN: International IP Administration, 5775 Morehouse  
Drive, San Diego, California 92121-1714 (US).

(74) Agent: **CICCOZZI, John L.** et al.; Muncy, Geissler, Olds & Lowe, P.C., 4000 Legato Road, Suite 310, Fairfax, Virginia 22033 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, 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,

(54) Title: METHOD AND APPARATUS FOR PERFORMING SIMD GATHER AND COPY OPERATIONS

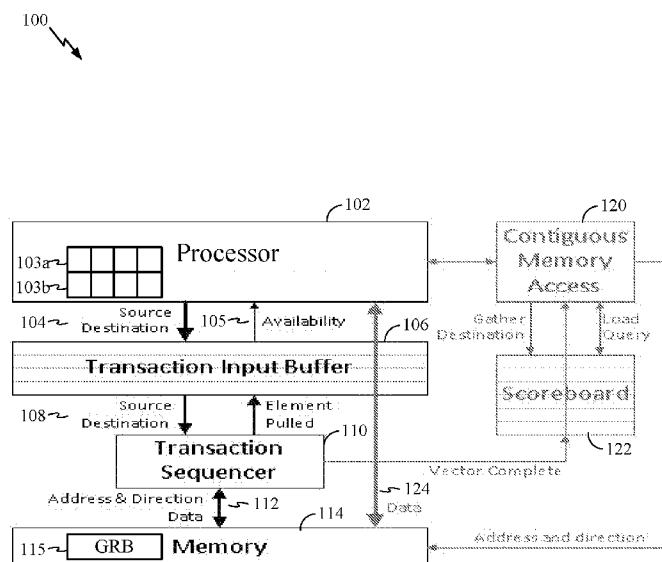


FIG. 1

(57) Abstract: Systems and methods relate to efficient memory operations. A single instruction multiple data (SIMD) gather operation is implemented with a gather result buffer located within or in close proximity to memory, to receive or gather multiple data elements from multiple orthogonal locations in a memory, and once the gather result buffer is complete, the gathered data is transferred to a processor register. A SIMD copy operation is performed by executing two or more instructions for copying multiple data elements from multiple orthogonal source addresses to corresponding multiple destination addresses within the memory, without an intermediate copy to a processor register. Thus, the memory operations are performed in a background mode without direction by the processor.

WO 2017/222798 A1



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, 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))*

## METHOD AND APPARATUS FOR PERFORMING SIMD GATHER AND COPY OPERATIONS

**Field of Disclosure**

[0001] Disclosed aspects are directed to processor instructions and efficient implementations thereof. More specifically, exemplary aspects pertain to efficient memory instructions involving multiple data elements, such as instructions related to memory copy, scatter, gather, and combinations thereof.

**Background**

[0002] Single instruction multiple data (SIMD) instructions may be used in processing systems for exploiting data parallelism. Data parallelism exists when a same or common task is to be performed on two or more data elements of a data vector, for example. Rather than use multiple instructions, the common task may be performed on the two or more data elements in parallel by using a single SIMD instruction which defines the same instruction to be performed on multiple data elements in corresponding multiple SIMD lanes. SIMD instructions may be used for variety of operations such as arithmetic operations, data movement operations, memory operations, etc. With regard to memory operations, “scatter” and “gather” are well-known operations for copying data elements from one location to another. The data elements may be located in a memory (e.g., a main memory or hard drive) and registers specified in the operations may be located on a processor or system on chip (SoC).

[0003] While a conventional load instruction may be used to read a data element from a memory location into a scalar destination register, e.g., located in the processor, a “gather” instruction on the other hand is used to load multiple data elements into a vector destination register, e.g., located in the processor. Each one of the multiple data elements may have independent or orthogonal source addresses (which may be non-contiguous in the memory), which makes SIMD implementations of a gather instruction challenging. Some implementations may execute a gather instruction through multiple load instructions to serially load each data element into its respective location in the vector destination register until the vector destination register is complete. However, serialization in this manner leads to poor performance and each component load instruction may have a variable latency depending on where each data element is sourced from (e.g., some source addresses may hit in a cache while others may not;

different source addresses may have different data dependencies, etc.). If the component load instructions are implemented to update the vector destination register in-order, then it may not be possible to pipeline the updates in software or hide the bulk of this variable latency using out-of-order processing mechanisms. For implementations where out-of-order updates of the vector destination registers are possible, additional registers (e.g., for temporary storage), tracking mechanisms per data element for individual updates, and other related software and/or hardware support may be incurred. Thus, conventional implementations of gather operations may be inefficient and involve large latencies and additional hardware.

- [0004] Scatter operations may be viewed as a counterpart of the above-described gather operations, wherein data elements from a source vector register, e.g., located in a processor, may be stored in multiple destination memory locations which may be non-contiguous. Some code sequences or programs may involve operations where multiple data elements are to be read from independent or orthogonal source locations (which may be non-contiguous in the memory) and copied or written to independent or orthogonal destination locations (which may also be non-contiguous in the memory). Such operations may be viewed as multiple copy operations on multiple data elements. Thus, it is desirable to use SIMD processing on such operations to implement a SIMD copying behavior of multiple data elements from orthogonal source locations to orthogonal destination locations in the memory.
- [0005] While in theory, such functionality may be achieved through a SIMD gather of the multiple data elements from the multiple source locations in the memory into a gather destination vector register located in the processor and then performing a SIMD scatter of the data elements from the gather destination vector register to the multiple destination locations in the memory, implementations of such functionality may not be practical or feasible. This is because waiting for the gather destination vector register to be complete introduces the above-described inefficiencies of the conventional implementations of the SIMD gather operations. Synchronization between the component loads of the SIMD gather and the component stores of the SIMD scatter operation is also challenging if the SIMD copy were to be implemented without waiting for the gather destination vector register to be completed first, before allowing the SIMD scatter to proceed. Furthermore, implementing a SIMD gather following a SIMD scatter to execute a SIMD copy may involve transfer of a large number of data elements

from the source locations in the memory using the gather destination vector register in the processor as an intermediate landing spot, and then back to destination locations in the memory. As can be appreciated, such large data transfers back and forth between the memory and the processor increase power consumption and latency of the SIMD copy.

[0006] Accordingly, there is a need for improved implementations of the above-described memory operations to exploit the benefits of SIMD processing, while avoiding the aforementioned drawbacks of conventional implementations.

## SUMMARY

[0007] Exemplary embodiments of the invention are directed to systems and method for efficient memory operations. A single instruction multiple data (SIMD) gather operation is implemented with a gather result buffer located within or in close proximity to memory, to receive or gather multiple data elements from multiple orthogonal locations in a memory, and once the gather result buffer is complete, the gathered data is transferred to a processor register. A SIMD copy operation is performed by executing two or more instructions for copying multiple data elements from multiple orthogonal source addresses to corresponding multiple destination addresses within the memory, without an intermediate copy to a processor register. Thus, the memory operations are performed in a background mode without direction by the processor.

[0008] For example, an exemplary aspect is directed to a method of performing a memory operation, the method comprising: providing, by a processor, two or more source addresses of a memory, copying two or more data elements from the two or more source addresses in the memory to a gather result buffer; and loading the two or more data elements from the gather result buffer to a vector register in the processor using a single instruction multiple data (SIMD) load operation.

[0009] Another exemplary aspect is directed to a method of performing a memory operation, the method comprising: providing, by a processor, two or more source addresses and corresponding two or more destination addresses of a memory, and executing two or more instructions for copying two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory, without an intermediate copy to a register in a processor.

[0010] Another exemplary aspect is directed to an apparatus comprising a processor configured to provide two or more source addresses of a memory, a gather result buffer configured to receive two or more data elements copied from the two or more source addresses in the memory, and logic configured to load the two or more data elements from the gather result buffer to a vector register in the processor based on a single instruction multiple data (SIMD) load operation executed by the processor.

[0011] Yet another exemplary aspect is directed to an apparatus comprising: a processor configured to provide two or more source addresses and corresponding two or more destination addresses of a memory, and logic configured to copy two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory, without an intermediate copy to a register in a processor.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0012] The accompanying drawings are presented to aid in the description of embodiments of the invention and are provided solely for illustration of the embodiments and not limitation thereof.

[0013] FIG. 1 illustrates a processing system configured according to exemplary aspects of this disclosure.

[0014] FIGS. 2-3 illustrate processes relating to exemplary memory operations according to exemplary aspects of this disclosure

[0015] FIG. 4 illustrates an exemplary computing device 400 in which an aspect of the disclosure may be advantageously employed.

#### **DETAILED DESCRIPTION**

[0016] Aspects of the invention are disclosed in the following description and related drawings directed to specific embodiments of the invention. Alternate embodiments may be devised without departing from the scope of the invention. Additionally, well-known elements of the invention will not be described in detail or will be omitted so as not to obscure the relevant details of the invention.

[0017] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any embodiment described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments. Likewise, the term

“embodiments of the invention” does not require that all embodiments of the invention include the discussed feature, advantage or mode of operation.

[0018] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of embodiments of the invention. As used herein, the singular forms "a," "an," and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises", "comprising," "includes," and/or "including," when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0019] Further, many embodiments are described in terms of sequences of actions to be performed by, for example, elements of a computing device. It will be recognized that various actions described herein can be performed by specific circuits (e.g., application specific integrated circuits (ASICs)), by program instructions being executed by one or more processors, or by a combination of both. Additionally, these sequence of actions described herein can be considered to be embodied entirely within any form of computer readable storage medium having stored therein a corresponding set of computer instructions that upon execution would cause an associated processor to perform the functionality described herein. Thus, the various aspects of the invention may be embodied in a number of different forms, all of which have been contemplated to be within the scope of the claimed subject matter. In addition, for each of the embodiments described herein, the corresponding form of any such embodiments may be described herein as, for example, “logic configured to” perform the described action.

[0020] In an exemplary aspect of this disclosure, a SIMD gather operation may be implemented by splitting the operation into two sub-operations: a first sub-operation to gather multiple data elements (e.g., from independent or orthogonal locations in a memory, which may be non-contiguous) to a gather result buffer; and a second sub-operation to load from the gather result buffer to a SIMD register, e.g., located in a processor. The exemplary SIMD gather operation may be separated by software implementations (e.g., a compiler) into the two sub-operations, and they may be pipelined to minimize latencies (e.g., using software pipelining mechanisms for the first sub-operation, to gather the multiple data elements into the gather result buffer in an out-of-order manner). The gather result buffer may be located within the memory or in proximity to

the memory, and is distinguished from a conventional gather destination vector register located in a processor. Thus, per-element tracking mechanisms are not needed for the gather result buffer. Furthermore, the second sub-operation may load multiple data elements from the gather result buffer into a destination register (e.g., located in the processor) which can accommodate the multiple data elements. The data elements may be individually accessible from the destination register and may be ordered based on the order in the gather result buffer, which simplifies the load operation of the multiple data elements from the gather result buffer to the destination register (e.g., the load operation may resemble a scalar load of the multiple data elements, rather than a vector load which specifies the location of each one of the multiple data elements). Accordingly, in an exemplary aspect, multiple data elements from orthogonal source locations can be effectively gathered into the destination register in the processor by use of the gather result buffer located in the memory.

[0021] In another exemplary aspect of this disclosure, data elements from orthogonal source locations in the memory can be efficiently copied on to orthogonal destination locations in the memory. For example, a SIMD copy operation may be implemented using a combination of gather operations and scatter operations, wherein the combination may be effectively executed within the memory. In this regard, executing the SIMD copy within the memory is meant to convey that the operation is performed without using registers located in a processor (such as a conventional gather destination vector register located in the processor) for intermediate storage. For example, executing the combination of gather and scatter operations within the memory can involve the use of a network or a sequencer located in close proximity to the memory, while avoiding the transfer of the data elements between the memory and the processor. An exemplary SIMD copy instruction with per-element addressing for multiple data elements may specify a list of the gather or source addresses from which to copy the multiple data elements and a corresponding list of scatter or destination addresses to which the multiple data elements are to be written to. From these lists, multiple copy operations may be performed in an independent or orthogonal manner to copy each one of the multiple data elements from its respective source address to its respective destination address. In exemplary aspects, each one of the multiple copy operations can be allowed to complete without requiring an intermediate vector (e.g., a gather vector) to ever be

completed, thus allowing for a relaxed memory ordering and out-of-order completion of the multiple copy operations.

[0022] With reference now to FIG. 1, an exemplary processing system 100, configured according to the above-described exemplary aspects, will be described. As shown, processing system 100 may include processor 102 which may be configured to implement an execution pipeline. In some aspects, the execution pipeline of processor 102 may support vector instructions and more specifically, SIMD processing. Two registers 103a and 103b have been illustrated in processor 102 to facilitate the description of exemplary aspects. These registers 103a-b may belong to a register file (not shown), and in some aspects, may be vector registers. Accordingly, register 103a may be a source register and register 103b may be a vector register for example cases discussed below. For example, data elements of source vector register 103a may be specified in a conventional scatter operation. Destination vector register 103b may be used in exemplary SIMD gather operations as described below.

[0023] For exemplary SIMD operations, transaction input buffer 106 may receive instructions from processor 102, with addresses for source and destination operands on bus 104. Source and destination addresses on bus 104 may correspond to the exemplary SIMD gather operation (e.g., to destination vector register 103b) or the exemplary SIMD copy operation described previously, and explained further with reference to FIGS. 2 and 3 below. Transaction input buffer 106 may implement a queueing mechanism to queue and convey feedback in terms of asserting the signal shown as availability 105, to convey whether more instructions (or related operands) can be received from processor 102 or by de-asserting availability 105 if the queue is full.

[0024] The instructions which are queued in transaction input buffer 106 may be transferred on bus 108 to transaction sequencer 110. In exemplary aspects, transaction sequencer 110 may be configured to serialize or parallelize the instructions from bus 108 based on the operations and adjustable settings. For memory operations, the source and/or destination addresses may be provided to memory 114 on bus 112 (along with respective controls). Bus 112 is shown as a two-way bus, on which data can be returned from memory 114 (a control for direction of data may indicate whether data transfer is from memory 114 or to memory 114). In various alternative implementations, separate wires may be used for the addresses, control, and data buses collectively shown as bus 112.

[0025] Processing system 100 can also include processing elements such as the blocks shown as contiguous memory access 120 and scoreboard 122. In an example, if a SIMD instruction pertains to gathering data elements from contiguous memory locations, the SIMD instruction can be executed as a conventional vector operation to load data from contiguous memory locations into a vector register (e.g., register 103b) in processor 102, for which the exemplary transaction sequencer 110 may be avoided. Scoreboard 122 may function similarly as transaction input buffer 106, and as such may implement queueing mechanisms. In one aspect, where scoreboard 122 receives data from memory 114 for a conventional vector operation such as a SIMD load or a SIMD gather from contiguous memory locations, the multiple data elements may be provided through transaction sequencer 110 to scoreboard 122, and once the destination vector is complete, the destination vector may be provided to processor 102 to be updated in vector register 103b of processor 102, for example. The operations of conventional elements such as contiguous memory access 120 and scoreboard 122 have been illustrated to convey their ability to interoperate with the exemplary blocks, transaction input buffer 106 and transaction sequencer 110 for memory operations.

[0026] With combined reference to FIGS. 1-2, process 200 related to an exemplary SIMD gather operation will now be explained. As shown in block 202, processor 102 can provide two or more source addresses, for example based on a gather instruction or two or more load instructions. A compiler or other software may recognize a SIMD gather operation and decompose it into component load instructions for an exemplary SIMD gather operation in some aspects. The two or more source addresses may be orthogonal or independent, and may pertain to non-contiguous locations in memory 114. The component load instructions may specify contiguous registers or a destination vector register (e.g., register 103b) of processor 102 to which two or more data elements from the two or more source addresses are to be gathered into.

[0027] In block 204, processor 102 can implement the exemplary SIMD gather operation by sending the two or more source addresses to transaction input buffer 106, and from there on to transaction sequencer 110 on buses 104 and 108. Transaction sequencer 110 may provide, either in parallel, or in series, two or more instructions to copy the two or more data elements from the two or more source addresses to a gather result buffer (e.g., GRB 115) exemplarily shown in memory 114. Gather result buffer 115 may be a circular buffer implemented within memory 114. In some aspects, gather result buffer

115 may be located outside memory 114 (e.g., in closer proximity to memory 114 than to processor 102) and in communication with memory 114. In some aspects gather result buffer 115 may be any other appropriate storage structure, and not necessarily a circular buffer. The two or more copy operations of the two or more data elements may involve two or more different latencies. Further, the two or more copy operations of the two or more data elements to gather result buffer 115 may be performed in the background, e.g., under the direction of transaction sequencer 110 without direction by processor 102. Thus, processor 102 may perform other operations (e.g., utilizing one or more execution units which are not explicitly shown) while the multiple copy operations are being executed in the background.

[0028] Once gather result buffer 115 is complete, as shown in block 206, a load instruction may be issued to load the data elements from gather result buffer 115 to a vector register such as register 103b, in processor 102. The load may correspond to a SIMD load to load two or more data elements from contiguous memory locations within gather result buffer 115 into vector register 103b. Scoreboard 122 may also be utilized to keep track of how many copy operations have been performed to determine whether gather result buffer 115 is complete before the load instruction is issued. In some approaches, one or more synchronization instructions may be executed (e.g., by software control) to ensure that gather result buffer 115 is complete before loading the data elements from gather result buffer 115 into vector register 103b in processor 102. In this way, the latency of the copy operations to gather result buffer 115 can be hidden from processor 102 and the load instruction may be executed with precise timing to avoid delays.

[0029] With combined reference to FIGS. 1 and 3, process 300 related to an exemplary SIMD copy operation will be explained. The SIMD copy operation of process 300 can achieve equivalent results as a conventional SIMD gather operation followed by a conventional SIMD scatter operation. However, the exemplary SIMD copy operation can be implemented in exemplary aspects with less complexity and latency than implementing a SIMD gather operation followed by a SIMD scatter operation in a conventional manner.

[0030] For example, with reference to block 302, processor 102 may provide two or more source addresses and corresponding two or more destination addresses of memory 114. The two or more source addresses and/or the two or more destination addresses may be orthogonal or independent and non-contiguous. For example, a compiler may

decompose a conventional gather-to-scatter sequence of instructions or code into component instructions for supplying the source and destination addresses to processor 102. Once again, processor 102 may provide the two or more source addresses and corresponding two or more destination addresses to transaction input buffer 106. Transaction input buffer 106 may supply the two or more source addresses and corresponding two or more destination addresses to transaction sequencer 110 (as explained with reference to process 200 of FIG. 2 above). Transaction sequencer 110 may supply instructions to memory 114 for performing the following operations in block 304.

- [0031] In block 304, the two or more instructions may be executed for copying two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory, without an intermediate copy to a processor register in processor 102. For example, network elements such as transaction sequencer 110 may be utilized without transferring data to processor 102 during execution of the two or more instructions for copying. Accordingly, copying the two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory (e.g., memory-to-memory copy operations) may comprise executing a SIMD copy instruction, in a background mode without direction by processor 102. In this manner, forming an intermediate gather vector result may be avoided, and in some cases, a complete gather vector may never be fully formed in the execution of the two or more instructions for copying. Once the execution of the two or more instructions for copying is completed, transaction sequencer 110 may inform scoreboard 122, and/or processor 102 of the status of the two or more memory-to-memory copy operations as complete.
- [0032] Referring to FIG. 4, a block diagram of a particular illustrative aspect of computing device 400 according to exemplary aspects. Computing device 400 includes processor 102 which may be configured to support and implement the execution of exemplary memory operations according to processes 200 and 300 of FIGS. 2-3, respectively. In FIG. 4, processor 102 (comprising registers 103a-b), transaction input buffer 106, transaction sequencer 110, and memory 114 (comprising gather result buffer 115) of FIG. 1 have been specifically identified, while remaining details of FIG. 1 have been omitted in this depiction for the sake of clarity. Although not shown, one or more caches or other memory structures may also be included in computing device 400.

[0033] FIG. 4 shows display controller 426 coupled to processor 102 and to display 428. FIG. 4 also shows several components which may be optional blocks based on particular implementations of computing device 400, e.g., for wireless communication. Accordingly, coder/decoder (CODEC) 434 (e.g., an audio and/or voice CODEC) can be optional and where present, coupled to processor 102, and optional blocks speaker 436 and microphone 438 can be coupled to CODEC 434. Wireless controller 440 (which may include a modem) may also be optional and coupled to wireless antenna 442. In a particular aspect, processor 402, display controller 426, memory 432, CODEC 434, and wireless controller 440 are included in a system-in-package or system-on-chip device 422.

[0034] In a particular aspect, input device 430 and power supply 444 are coupled to the system-on-chip device 422. Moreover, in a particular aspect, as illustrated in FIG. 4, display 428, input device 430, speaker 436, microphone 438, wireless antenna 442, and power supply 444 are external to the system-on-chip device 422. However, each of display 428, input device 430, speaker 436, microphone 438, wireless antenna 442, and power supply 444 can be coupled to a component of the system-on-chip device 422, such as an interface or a controller.

[0035] It should be noted that although FIG. 4 depicts a wireless communications device, processor 102 and memory 114 may also be integrated into a set top box, a music player, a video player, an entertainment unit, a navigation device, a personal digital assistant (PDA), a fixed location data unit, a communications device, a server, or a computer. Further, at least one or more exemplary aspects of wireless device 400 may be integrated in at least one semiconductor die.

[0036] Those of skill in the art will appreciate that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[0037] Further, those of skill in the art will appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of

hardware and software, various illustrative components, blocks, 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 invention.

- [0038] The methods, sequences and/or algorithms described in connection with the embodiments 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 RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary 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.
- [0039] Accordingly, an embodiment of the invention can include a computer readable media embodying a method for efficient memory copy operations such as scatter and gather. Accordingly, the invention is not limited to illustrated examples and any means for performing the functionality described herein are included in embodiments of the invention.
- [0040] While the foregoing disclosure shows illustrative embodiments of the invention, it should be noted that various changes and modifications could be made herein without departing from the scope of the invention as defined by the appended claims. The functions, steps and/or actions of the method claims in accordance with the embodiments of the invention described herein need not be performed in any particular order. Furthermore, although elements of the invention may be described or claimed in the singular, the plural is contemplated unless limitation to the singular is explicitly stated.

**CLAIMS****WHAT IS CLAIMED IS:**

1. A method of performing a memory operation, the method comprising:  
providing, by a processor, two or more source addresses of a memory;  
copying two or more data elements from the two or more source addresses in the memory to a gather result buffer; and  
loading the two or more data elements from the gather result buffer to a vector register in the processor using a single instruction multiple data (SIMD) load operation.
2. The method of claim 1, wherein the gather result buffer is located in the memory or in close proximity to the memory.
3. The method of claim 1, wherein the gather result buffer is a circular buffer.
4. The method of claim 1, wherein the two or more source addresses are orthogonal or independent and non-contiguous in the memory.
5. The method of claim 1, comprising copying the two or more data elements to the gather result buffer out-of-order.
6. The method of claim 5, wherein copying the two or more data elements to the gather result buffer out-of-order involves two or more different latencies.
7. The method of claim 5, comprising copying the two or more data elements to the gather result buffer out-of-order in a background mode without direction by the processor.
8. The method of claim 5, comprising tracking the gather result buffer and loading the two or more data elements from the gather result buffer after the gather result buffer is complete.
9. A method of performing a memory operation, the method comprising:

providing, by a processor, two or more source addresses and corresponding two or more destination addresses of a memory; and

executing two or more instructions for copying two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory, without an intermediate copy to a register in a processor.

10. The method of claim 9, wherein the two or more source addresses are orthogonal or independent and non-contiguous.

11. The method of claim 9, wherein the two or more destination addresses are orthogonal or independent and non-contiguous in the memory.

12. The method of claim 9, wherein copying two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory comprises executing a single instruction multiple data (SIMD) copy instruction.

13. The method of claim 12, comprising executing the SIMD copy instruction in a background mode without direction by the processor.

14. An apparatus comprising:

a processor configured to provide two or more source addresses of a memory;

a gather result buffer configured to receive two or more data elements copied from the two or more source addresses in the memory; and

logic configured to load the two or more data elements from the gather result buffer to a vector register in the processor based on a single instruction multiple data (SIMD) load operation executed by the processor.

15. The apparatus of claim 14, wherein the gather result buffer is located in the memory or in close proximity to the memory.

16. The apparatus of claim 14, wherein the gather result buffer is a circular buffer or storage structure configured to receive the two or more data elements out-of-order.

17. The apparatus of claim 14, wherein the two or more source addresses are orthogonal or independent and non-contiguous in the memory.
18. The apparatus of claim 14, wherein the two or more data elements are copied to the gather result buffer out-of-order in a background mode without direction by the processor.
19. The apparatus of claim 14, wherein the logic comprises a transaction sequencer configured to track the gather result buffer and generate a vector complete signal when the gather result buffer is complete.
20. An apparatus comprising:
  - a processor configured to provide two or more source addresses and corresponding two or more destination addresses of a memory; and
  - logic configured to copy two or more data elements from the two or more source addresses to corresponding two or more destination addresses within the memory, without an intermediate copy to a register in a processor.
21. The apparatus of claim 20, wherein the two or more source addresses are orthogonal or independent and non-contiguous.
22. The apparatus of claim 20, wherein the two or more destination addresses are orthogonal or independent and non-contiguous in the memory.
23. The apparatus of claim 20, comprising logic configured to copy the two or more data elements from the two or more source addresses to corresponding two or more destination addresses in a background mode without direction by the processor.

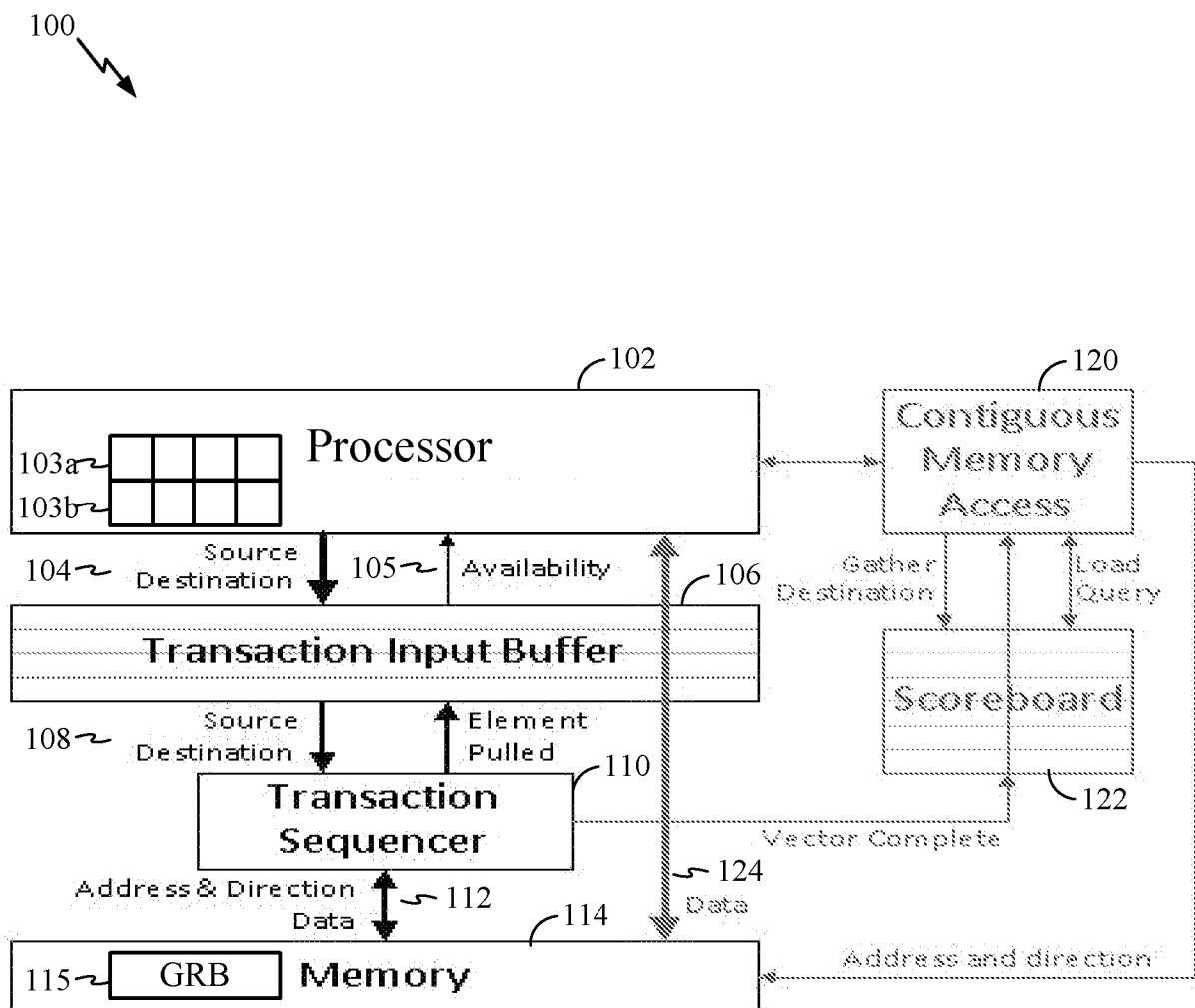


FIG. 1

2/4

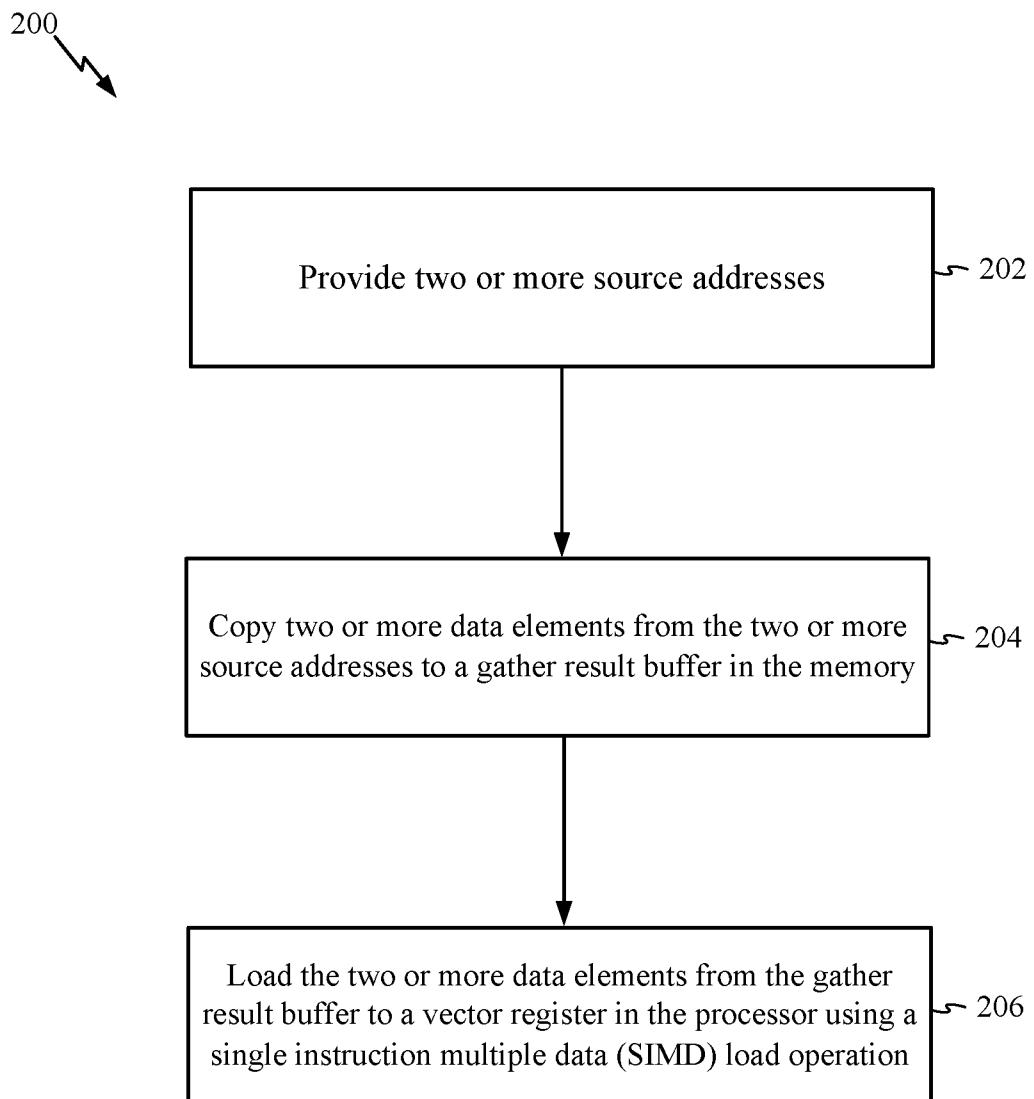


FIG. 2

3/4

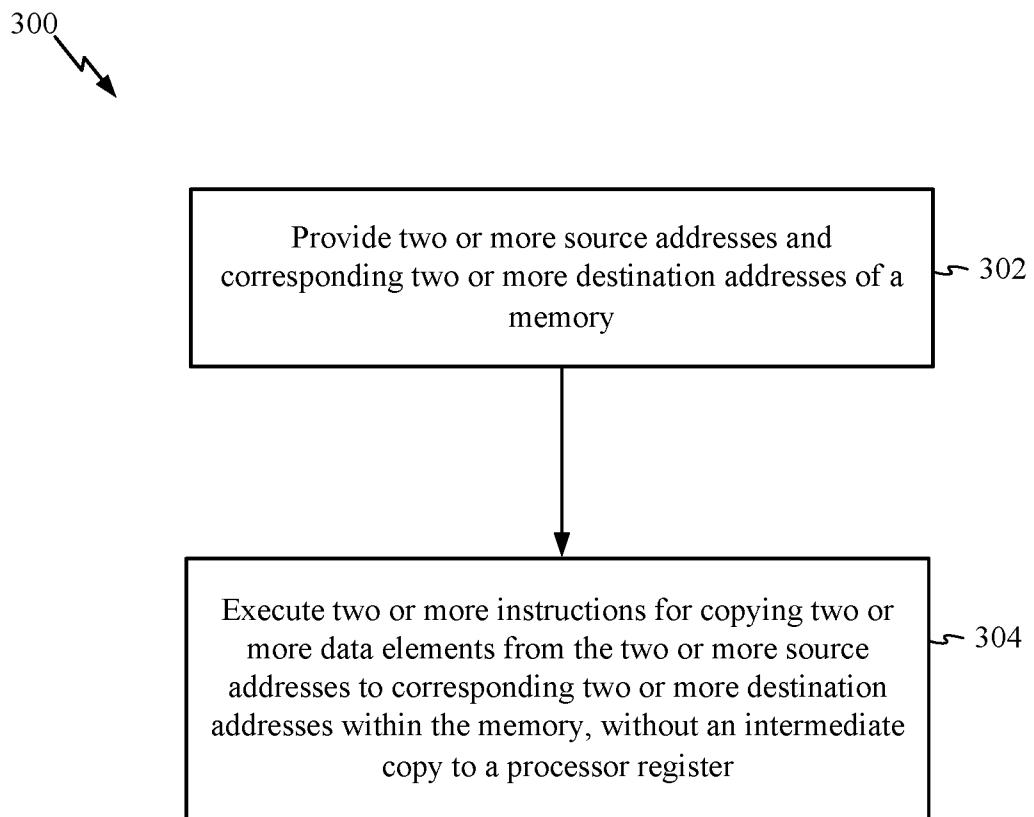


FIG. 3

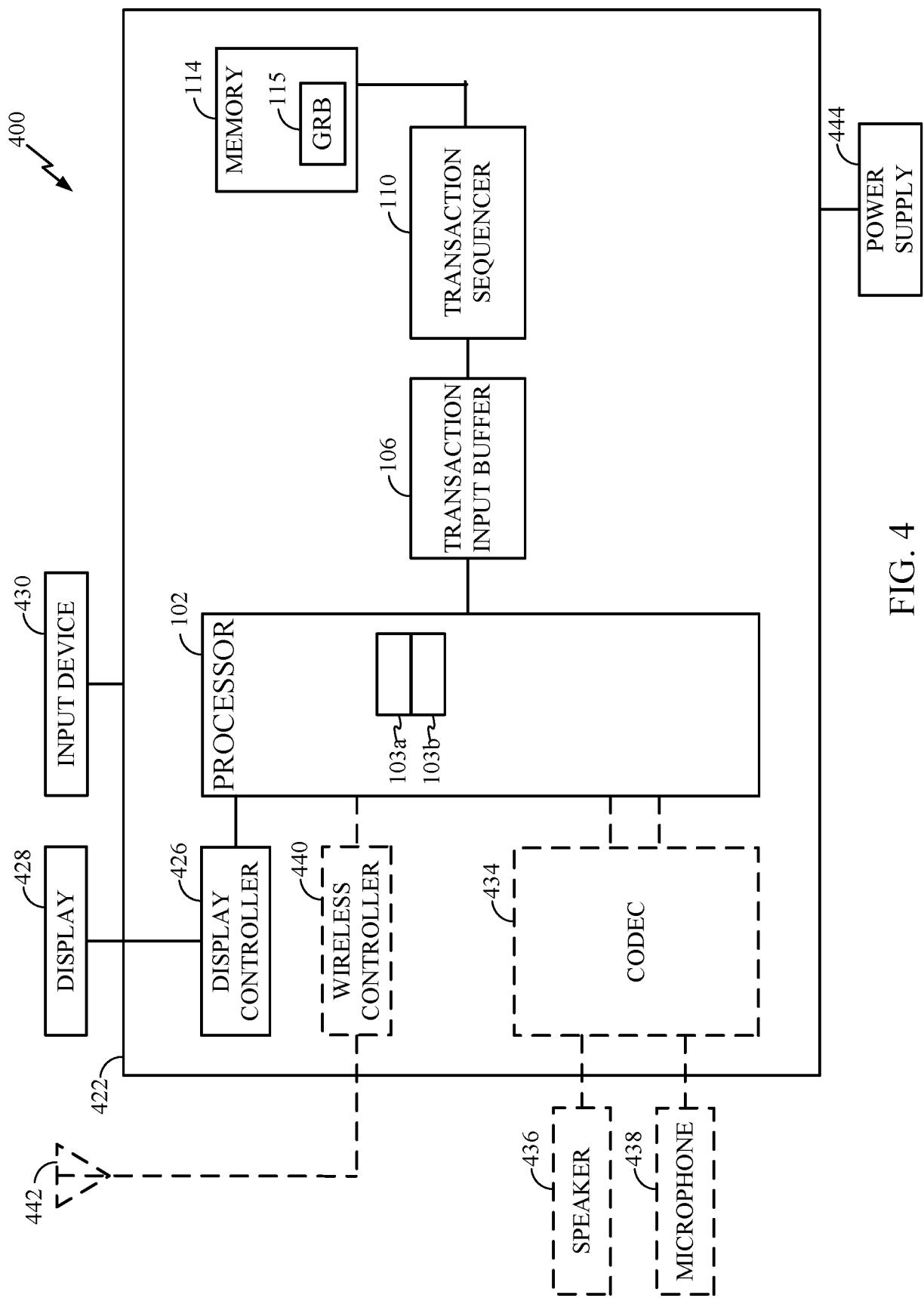


FIG. 4

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US2017/036041

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

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

1.  Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
2.  Claims Nos.:  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
  
3.  Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

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

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

see additional sheet

1.  As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
  
2.  As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
  
3.  As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
  
4.  No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

#### Remark on Protest

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

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

No protest accompanied the payment of additional search fees.

# INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2017/036041

A. CLASSIFICATION OF SUBJECT MATTER			
INV.	G06F9/30	G06F9/38	G06F9/345
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 6 513 107 B1 (ANSARI AHMAD R [US]) 28 January 2003 (2003-01-28) the whole document -----	1-8, 14-19	
X	US 2012/060016 A1 (EICHENBERGER ALEXANDRE E [US] ET AL) 8 March 2012 (2012-03-08) the whole document -----	1-8, 14-19	
X	US 2013/326160 A1 (SPERBER ZEEV [IL] ET AL) 5 December 2013 (2013-12-05) the whole document -----	1-8, 14-19	
X	US 5 887 183 A (AGARWAL RAMESH CHANDRA [US] ET AL) 23 March 1999 (1999-03-23) the whole document ----- -/-	1-3,5,6, 8,14-18	
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C.		<input checked="" type="checkbox"/> See patent family annex.	
<small>* Special categories of cited documents :</small>			
<small>"A" document defining the general state of the art which is not considered to be of particular relevance</small>		<small>"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</small>	
<small>"E" earlier application or patent but published on or after the international filing date</small>		<small>"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</small>	
<small>"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)</small>		<small>"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</small>	
<small>"O" document referring to an oral disclosure, use, exhibition or other means</small>		<small>"&amp;" document member of the same patent family</small>	
<small>"P" document published prior to the international filing date but later than the priority date claimed</small>			
Date of the actual completion of the international search		Date of mailing of the international search report	
29 August 2017		05/09/2017	
Name and mailing address of the ISA/		Authorized officer	
<small>European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016</small>		Klocke, Lynn	

**INTERNATIONAL SEARCH REPORT**

International application No PCT/US2017/036041
---

**C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2007/071606 A2 (IBM [US]; IBM UK [GB]; ARIMILLI RAVI [US]; GOVINDARAJU RAMA [US]; HOCH) 28 June 2007 (2007-06-28) the whole document -----	9-13, 20-23
X	WO 2010/088129 A1 (MIPS TECH INC [US]; KISHORE KARAGADA R [US]; KISSELL KEVIN D [US]; BEL) 5 August 2010 (2010-08-05) the whole document -----	9-13, 20-23

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2017/036041

Patent document cited in search report	Publication date	Patent family member(s)			Publication date
US 6513107	B1	28-01-2003	US 6513107 B1		28-01-2003
			US 2003167387 A1		04-09-2003
			US 2004186980 A1		23-09-2004
US 2012060016	A1	08-03-2012	NONE		
US 2013326160	A1	05-12-2013	US 2013326160 A1		05-12-2013
			US 2016103785 A1		14-04-2016
			US 2017192934 A1		06-07-2017
US 5887183	A	23-03-1999	NONE		
WO 2007071606	A2	28-06-2007	US 2007150659 A1		28-06-2007
			US 2009138664 A1		28-05-2009
			WO 2007071606 A2		28-06-2007
WO 2010088129	A1	05-08-2010	CN 102301326 A		28-12-2011
			US 2010199054 A1		05-08-2010
			US 2016098206 A1		07-04-2016
			WO 2010088129 A1		05-08-2010

**FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210**

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-23

Method of performing a memory operation

1.1. claims: 1-8, 14-19

Method of performing a memory gather operation

1.2. claims: 9-13, 20-23

Method of performing a memory copy operation

---