

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 February 2010 (11.02.2010)

PCT

(10) International Publication Number
WO 2010/017263 A1

(51) International Patent Classification:
G06F 13/28 (2006.01)

(21) International Application Number:
PCT/US2009/052794

(22) International Filing Date:
5 August 2009 (05.08.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/086,631 6 August 2008 (06.08.2008) US

(71) Applicant (for all designated States except US): **SAND-BRIDGE TECHNOLOGIES, INC.** [US/US]; 120 White Plains Road, 4th Floor, Tarrytown, NY 10591 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **MOUDGILL, Mayan** [IN/US]; 143 Juniper Hill Road, White Plains, NY 10607 (US). **WANG, Shenghong** [US/US]; 2386 Claire Ct., Yorktown Hts, NY 10598 (US).

(74) Agent: **KARCESKI, Jeffrey, D.**; Barnes & Thornburg LLP, 750 17th Street, N.W., Suite 900, Washington, DC 20006 (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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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



WO 2010/017263 A1

(54) Title: HALTABLE AND RESTARTABLE DMA ENGINE

(57) Abstract: A method is described for operation of a DMA engine. Copying is initiated for transfer of a first number of bytes from first source memory locations to first destination memory locations. Then, a halt instruction is issued before the first number of bytes are copied. After copying is stopped, a second number of bytes is established, encompassing those bytes remaining to be copied. After the transfer is halted, a quantity of the second number of bytes is identified. Quantity information is then generated and stored. Second source memory locations are identified to indicate where the second number of bytes are stored. Second source memory location information is then generated and stored. Second destination memory locations are then identified to indicate where the second number of bytes are to be transferred. Second destination memory location information is then generated and stored.

Halttable and Restartable DMA Engine

Cross-Reference to Related Applications

[0001] This PCT Patent Application relies for priority on United States Provisional Patent Application Serial No. 61/086,631, filed on August 6, 2008, the contents of which are incorporated herein by reference.

Field of the Invention

[0002] This invention relates to Direct Memory Access (“DMA”) engines. More particularly, the invention concerns an architected DMA that is part of the process state of a processor.

Description of the Related Art

[0003] Traditionally, DMAs have been treated as peripheral devices. As peripheral devices, DMAs have not been considered or treated as a part of the architected state for a processor, such as a personal computer, personal data assistant (“PDA”), cell phone, or other device that is processor-based and typically would include one or more DMAs.

[0004] Since DMAs traditionally have been excluded from the architected state of a processor, in a multi-programmed environment, the DMA engine is not considered a part of the process state. Accordingly, the DMA must be accessed via a monitor, possibly within the operating system.

[0005] Accessing a DMA separately from other components included in the architected state increases the latency to access the DMA. As should be appreciated by those skilled in the art, an increase in latency for a device like a DMA makes the DMA impractical for use in a process, especially when the process incorporates small moves or operations. Simply, for small operations, the latency time associated with traditional DMA architecture may so greatly increase the overall processing time for an operation, that reliance on the DMA is impractical.

[0006] As should be appreciated by those skilled in the art, there is always a need in the computer processing art to increase processing efficiency.

Summary of the Invention

[0007] The invention offers at least one approach to increasing processing efficiency.

[0008] Specifically, the invention offers an architected DMA that is part of the process state. The DMA may be stopped and restarted, permitting a copy operation to be halted and resumed without a significant detrimental processing effect on the DMA.

[0009] The invention provides a method of operating a direct memory access engine. In the method, copying is initiated for a first number of bytes. The first number of bytes are to be copied from first source memory locations to first destination memory locations. After initiating the copying, a halt instruction is issued before the first number of bytes are copied. In response to the halt instruction, the copying is stopped. As a result, a second number of bytes is established. The second number of bytes are those bytes remaining to be copied from the first number of bytes. After the transfer is halted, a quantity of the second number of bytes is identified. Quantity information, which provides the quantity of the second number of bytes, is then generated and stored. Second source memory locations also are identified. The second source memory locations identify where the second number of bytes are stored. Second source memory location information is then generated and stored. Second destination memory locations are then identified. The second destination memory locations identify where the second number of bytes are to be transferred. Second destination memory location information is then generated and stored.

[0010] In one contemplated variation of this method, the quantity information, the second source memory location information, and the second destination memory location information are retrieved from at least one register in the direct memory access engine. One register or several registers may be used. Where there are several registers, one may be a source register, another may be a destination register. A third register may be a quantity register.

[0011] The method may also include resuming the halted transfer by initiating copying of the second number of bytes from the second source memory locations to

second destination memory locations using the quantity information, the second source memory location information, and the second destination memory location information.

[0012] With respect to this method, there are a number of variations contemplated for re-initiating the copy operation after issuance of the halt instruction.

[0013] In one contemplated embodiment, the method provides for determining a next address to read from the second source memory location information and determining a next address to write from the second destination memory location information. Then, the method flushes all pending reads and writes. The method proceeds by determining, from the next address to read and the next address to write, a decrement value. The decrement value identifies a number of bytes by which the next address to read is in advance of the next address to write. The method then decrements the next address to read by the decrement value to generate a decremented read address. Copying the second number of bytes from the second memory locations to the second memory then destinations then proceeds based at least upon the decremented read address, the next address to write, and the quantity information.

[0014] In a second contemplated variation, the method includes determining a next address to read from the second source memory location information and determining a next address to write from the second destination memory location information. The method tracks the number of bytes by which the next address to read is in advance of the next address to write and establishes a run-ahead value. The run-ahead value is stored so that the next address to read may be adjusted by the run-ahead value. This permits generation of an adjusted next address to read. Copying then proceeds based at least upon the adjusted next address to read, the next address to write, and the quantity information.

[0015] In a third contemplated variation, the method contemplates determining a next address to read from the second source memory location information and determining a next address to write from the second destination memory location information. Copying of the second number of bytes from the second memory locations to the second memory destinations proceeds using the next address to read, the next address to write, and the quantity information.

[0016] In a fourth contemplated variation of the method, an identification of the first source memory locations, an identification of the first destination memory locations, and a count of the first number of bytes are retained. After retrieving the quantity information, a next address to read is established, at least based upon the identification of the first source memory locations, the count of the first number of bytes, and the quantity information. Then, a next address to write is established at least based upon the identification of the first destination memory locations, the count of the first number of bytes, and the quantity information. Copying of the second number of bytes from the second memory locations to the second memory destinations relies at least upon the next address to read, the next address to write, the count of the first number of bytes, and the quantity information.

[0017] In a fifth contemplated variation, an indication of the next address to write is retained. After issuance of the halt instruction, reading of the first number of bytes from the first source memory locations is stopped. In addition, writing continues for any of the bytes that remain from the bytes read before stopping the reading. Then, a next address to read from the second source memory location information and a next address to write from the second destination memory location information are determined. Copying of the second number of bytes from the second memory locations to the second memory destinations relies at least upon the next address to read, the next address to write, and the quantity information.

[0018] Other aspects of the invention will be made apparent from the discussion that follows and from the drawings appended hereto.

Brief Description of the Drawings

[0019] The invention is described in connection with drawings that illustrate one or more aspects, in which:

[0020] Fig. 1 is a first portion of a flow diagram illustrating one embodiment of the method contemplated by the invention;

[0021] Fig. 2 is a second portion of a flow diagram illustrating the method begun in Fig. 1, this figure being a continuation of the portion of the method illustrated in Fig. 1;

[0022] Fig. 3 is a flow a third portion of a flow diagram illustrating the method begun in Fig. 1, this figure being a continuation of the portion of the method illustrated in Fig. 2;

[0023] Fig. 4 is a flow diagram illustrating additional operations that may be included in the method illustrated in Figs. 1-3;

[0024] Fig. 5 is a flow diagram illustrating a first contemplated variation of the method illustrated in Figs. 1-3;

[0025] Fig. 6 is a flow diagram illustrating a second contemplated variation of the method illustrated in Figs. 1-3;

[0026] Fig. 7 is a flow diagram illustrating a third contemplated variation of the method illustrated in Figs. 1-3;

[0027] Fig. 8 is a flow diagram illustrating a fourth contemplated variation of the method illustrated in Figs. 1-3; and

[0028] Fig. 9 is a flow diagram illustrating a fifth contemplated variation of the method illustrated in Figs. 1-3.

Description of Embodiment(s) of the Invention

[0029] The invention is described in connection with specific embodiments and examples detailed below. The invention, however, is not intended to be limited solely to the embodiments and examples discussed. To the contrary, the embodiments and examples are intended to define the broad scope of the invention. As should be appreciated by those skilled in the art, there are numerous equivalents and variations of the embodiments and examples that may be employed without departing from the scope of the invention. Those embodiments and variations are intended to be encompassed by the invention.

[0030] As a preliminary matter, and as should be appreciated by those skilled in the art, DMA engines are used to copy values from one location in memory to another. At their simplest, DMA engines copy values from some contiguous block of memory to some other block. One example of this basic copy function of a DMA engine is provided by Code Segment #1, below.

Code Segment #1

```
for(i=0;i<CNT;i++) {  
    DST[i] = SRC[i];  
}
```

[0031] To use a simple DMA from a processor, typically the processor writes the source address, destination address, and the transfer size to selected registers. The processor then initiates the transfer by writing to a control register. When all the values have been copied, the DMA signals the completion of the transfer by writing to a control and/or a status register and/or triggering an interrupt in the processor.

[0032] More sophisticated DMAs are capable of operations such as “scatter” operations. “Scatter” refers to an operation whereby a single procedure call sequentially reads data from a single data stream to multiple buffers. A scatter operation may be written in code as set forth in Code Segment #2, below.

Code Segment #2

```
for(i=0;i<CNT;i++) {  
    DST [OFF[i]] = SRC[i];  
}
```

“SRC” refers to the address of the source. “DST” refers to the address of the destination. “OFF” refers to the address of the offset array.

[0033] In one contemplated embodiment, a scatter operation takes data from a plurality of source memory locations and “scatters” the data to a number of destination memory locations. As indicated above, a scatter operation also may operate from a single data stream, which streams data from a plurality of source memory locations. If so, the scatter operation may include a number of data manipulations. Specifically, such a scatter operation may: (1) read a sequential data stream of data to be copied, (2) read a sequential address stream or an index stream, and (3) if using an index stream, create an address stream by adding each index to a single base address. Each element in the data may then be copied to a corresponding address in the stream. It is noted that these operations are not required for all scatter operations but are provided merely as guidance

for those skilled in the art and to assist with an understanding of one aspect of the invention.

[0034] More sophisticated DMAs also are capable of specific operations, referred to as “gather” operations. “Gather” operations are those operations where a single procedure call sequentially writes data from a multiple buffers to a single data stream. A gather operation may be written in code as set forth in Code Segment #3, below.

Code Segment #3

```
for(i=0;i<CNT;i++) {  
    DST[i] = SRC [OFF[i]];  
}
```

As with Code Segment #2, “SRC” refers to the address of the source. Similarly, “DST” refers to the address of the destination. Additionally, “OFF” refers to the address of the offset array.

[0035] With respect to “gather” operations, the information may be gathered from a plurality of source memory locations. The data may then be provided to a plurality of destination memory locations, perhaps via a scatter algorithm. Alternatively, the data may be gathered and funneled to a single data stream for further processing. As should be appreciated by those skilled in the art, a gather operation does the opposite of a scatter operation, at least in some ways. For example, a gather operation may: (1) read either an address stream or an index stream and generate an address stream by adding it to a base, or (2) read the data from the locations in the address stream, creating a sequential data stream. The sequential data stream may then be copied sequentially to the destination(s). This discussion is provided to clarify at least this one aspect of the invention. It is not intended to be limiting of the invention.

[0036] In addition, more sophisticated DMAs are capable of “multi-level” operations. Multi-level operations are those where data from multiple sources are read from and/or written to multiple destinations. A multi-level operation may be written in code as set forth in Code Segment #4, below.

Code Segment #4

```
for(i=0;i<CNT0;i++) {  
    for (j=0; j<CNT1; j++)  
        DST[i][j] = SRC[i][j];  
    }  
}
```

As with Code Segments ## 2 and 3, “SRC” refers to the address of the source. “DST” refers to the address of the destination. “OFF” refers to the address of the offset array.

[0037] Scatter, gather, and multi-level operations may be combined to generate still further functions, as should be appreciated by those skilled in the art. Accordingly, further details concerning these functions are not provided here.

Context Switch

[0038] Once again with reference to the typical DMA engine, after the engine initiates a transfer, the engine runs to completion. The DMA engine then signals the processor that the transfer has been completed. Following this, the DMA engine executes a second transfer. The process is repeated as needed. As should be appreciated by those skilled in the art, this is but one way to describe the operation of a DMA engine.

[0039] There are other ways to setup DMAs and series of DMAs. For example, some DMAs support the use of “shadow registers”. A shadow register permits a DMA to be programmed with the next transfer while the current DMA transfer is “in-flight” or in process. The shadow register, therefore, facilitates the next transfer because the DMA may start the next transfer as soon as the current transfer is completed. Other DMAs support “chaining”. This approach differs from the shadow register approach. Instead of programming the DMA directly, a control block is written with the details of the transfer. The next transfer is written to a control block, and a location in the first control block is set to point to this next transfer. In this manner, a chain of transfers may be established. As soon as the DMA completes the transfer determined by a control block, it proceeds to execute the transfer determined by the next block in the chain.

[0040] As should be apparent to those skilled in the art, these alternative set-ups also suffer from the same problems noted with respect to DMAs generally. In particular,

DMAs are not well suited to execute all types of operations because of several deficiencies, including the aforementioned latency issues. For example, the existing architecture for DMAs provides no mechanism to suspend an ongoing transfer, to program and execute a new transfer, and then to resume the suspended transfer. In addition, if multiple sources try to program a DMA engine, these sources must coordinate efforts to the DMA engine to prevent simultaneous access or to prevent multiple instructions from overwriting one another.

[0041] To facilitate understanding of the deficiencies in the prior art with respect to DMAs, the following example is provided. Specifically, the example encompasses an instance where the DMA is to be used to execute data transfers on a general purpose processor. To simplify the example, it is assumed that all memory accesses are in a real-mode. Real-mode operation is assumed to avoid the complications that arise when programming for virtual memory, as should be understood by those skilled in the art. Specifically, by ignoring virtual memory in this example, the complications introduced by virtual memory translation are also avoided. For this example, several problems arise, as detailed below.

[0042] The first problem may arise in the context of a transfer process that is interrupted and the context is switched to another transfer operation. Specifically, if the transfer process is interrupted during the interval when the DMA is being programmed but before the DMA writes to the control register, the DMA registers must be saved as part of the process state. As a result, before the transfer operation may be resumed, the DMA registers must be recreated (or “written back” to the appropriate addresses).

[0043] The second problem that may arise concerns processes that program long-running DMAs. As may be immediately apparent, when a process programs a long-running DMA and the process is then context-switched, the new process must await completion of the first transfer process before being executed. As a result, the execution of the first transfer operation stalls the execution of the second transfer operation.

[0044] A third problem that may arise with respect to DMA processing arises in the context of a multi-threaded or multi-processor environment. Specifically, in a multi-threaded or multi-processor machine, multiple processes simultaneously may try to reprogram the DMA. Since only one process may proceed at any given time, the

multiple processes require execution of a mutual exclusion algorithm. As should be appreciated by those skilled in the art, mutual exclusion algorithms increase the latency period associated with programming the DMA.

[0045] The fourth problem that may arise concerns the generation of an interrupt signal by the DMA, which may be issued at the conclusion of a transfer operation. Specifically, when a DMA signals completion of a transfer operation by means of an interrupt, there is no guarantee that the process that programmed the DMA is the currently-running process. Consequently, when issued, the completion interrupt must be captured by a shared interrupt handler, which decides if the currently-running process is the process to which the interrupt must be delivered.

[0046] If the processor uses virtual memory, additional problems arise, because the DMA engine must translate between virtual and real addresses. It is contemplated that the operating system may be employed to translate the desired transfer into real addresses, and pin the source and target pages. However, reliance on the operating system for this translation dramatically increases the latency to initiate a DMA. Lighter-weight schemes (i.e., less latency-burdened schemes) typically rely on virtual addresses in the DMA. These schemes run the DMA transfer addresses through the processor's translation mechanism for operation. A problem with this approach is that, after a context switch, the processor's translation mechanism may not be valid for the previous process. As a result, there must be a method for stopping the DMA to avoid conflicts in processing.

[0047] Each of these problems present challenges in the execution of transfer operations by a DMA. The invention offers solutions for these problems.

Restartable Stop

[0048] The present invention offers a restartable stop of a block-transfer DMA. A restartable stop of a block-transfer DMA is one where it is possible to stop the DMA in mid-transfer. Once the transfer has been stopped, it is then possible to determine: (1) how many bytes are left to be transferred, (2) the address of the next byte to be read, and (3) the address of the next byte to be written. This approach may be implanted simply by

reading the control registers of the DMA. Other possible implementations are also contemplated.

Simple

[0049] The first variation presented by the invention is referred to as the “simple” implementation. In the simple implementation, copies of the initial values of the source, destination, and byte count are retained, along with a count of the number of the written bytes. Since this information is retained, a restart state of the DMA transfer operation may be determined by adding and/or subtracting the initial values to or from the transferred byte count. As should be appreciated by those skilled in the art, this simple approach is inefficient. One reason for this inefficiency is that the implementation of the DMA requires the system to keep track of the next byte to read and/or to write.

Drain

[0050] The second variation presented by the invention is referred to as the “drain” implementation. In this implementation, the DMA keeps a running count of the next address to read and/or to write. The DMA also keeps track of the number of bytes remaining to be read. When stopped mid-transfer, the DMA stops reading and waits for the bytes that have already been read to be written to memory. Once the bytes have been written to memory, the DMA is placed into a restartable state, from which the DMA may complete the transfer operation. As should be apparent, this implementation suffers from at least one disadvantage. Specifically, all pending reads must be written before the DMA is in a restartable state. This arrangement is likely to result in latencies during execution of the transfer operation.

Early Stop

[0051] The third variation presented by the invention is referred to as the “early stop” implementation. In this implementation, the DMA keeps a running count of the next address to read and/or to write. The DMA also retains the number of bytes left to write. When the DMA is stopped mid-transfer, the DMA stops writing and flushes all pending reads and/or writes. Generally, in this embodiment, the read address runs ahead

of the write address (i.e., more bytes have been read than written). Consequently, the read address must be decremented by the amount of bytes that the read address is ahead of the write address. This approach does not suffer from the kinds of disadvantages discussed in connection with the first two approaches and, therefore, is proffered as one attractive approach for implementation of the present invention.

Run-Ahead

[0052] The fourth variation is referred to as the “run-ahead” implementation. This implementation is similar to the early stop implementation, except that a separate register (i.e., an extra register) is used to keep track of how far the read address is ahead of the write address (i.e., the difference between the number of bytes read and written). This value is referred to as the “run-ahead” value. Since the run-ahead value is generally small, the extra register may be implemented fairly easily and cheaply. In other words, although the extra register is required for the run-ahead approach, the additional register does not add significantly to the overall “cost” associated with operation of the DMA. In this implementation, to recover the restart state, the processor subtracts the run-ahead value from the read address to obtain the correct restart read address.

Offset State

[0053] As discussed above, DMAs may implement both scatter and gather operations. When a DMA is engaged in a scatter and/or a gather, the reading of the offsets tends to run ahead of the reads and/or writes of the data. To recover the restart state of the offset, then, it is necessary to use some variation of the early stop or run-ahead techniques described above.

Architected Registers

[0054] Traditionally, DMAs have been treated as peripherals, and consequently, the DMA registers have been accessed either through load and/or store operations or through special input and/or output instructions.

[0055] Since one goal of the invention is to make the DMA a part of the processor context, the DMA registers part are made a part of the architected register state

of the processor. In the invention, the DMA registers are special-purpose registers accessible through the same instructions as other special purpose registers.

[0056] With respect to the various embodiments of the invention discussed above, reference is now made to the figures appended hereto. With respect to the figures, it is possible that any of the operations identified may encompass one or more steps. Moreover, different operations may be combined into a single step in some instances. These possibilities are intended to be encompassed by the invention.

[0057] Fig. 1 is a flow diagram outlining the fundamental method of the invention. For reference, the method is provided with the reference identifier 10. The method 10 is contemplated to be executed on a processor, such as a DMA engine. The method begins at 12. After the start 12, the method 10 proceeds to 14, where copying is initiated for a first number of bytes from first source memory locations to first destination memory locations. After initiating the copying at 14, a halt instruction is issued at 16 before the first number of bytes are copied from the first source memory locations to the first destination memory locations. At 18, in response to issuance of the halt instruction at 16, the copying is halted, thereby establishing a second number of bytes, which comprise those bytes remaining to be copied from the first number of bytes. Then, at 20, a quantity of the second number of bytes is identified. Fig. 1 includes a transition block 22, which indicates that the method continues in Fig. 2.

[0058] Fig. 2 provides the continuation of the method illustrated in Fig. 1. Specifically, once the quantity is identified, quantity information is generated concerning the quantity of the second number of bytes at 24. This quantity information is then stored at 26. From 26, the method proceeds to 28 where the second source memory locations are identified. The second source memory locations encompass the locations where the second number of bytes are stored. From 28, the method 10 proceeds to 30 where second source memory location information is generated. As should be apparent, the second source memory location information encompasses the second source memory locations in which the second number of bytes are stored. Then, at 32, the second memory source location information is stored. Fig. 2 includes a transition block 34, which indicates that the method 10 continues to Fig. 3.

[0059] Fig. 3 illustrates the remainder of the method 10. In Fig. 3, the method 10 continues from 32 where second destination memory locations are identified at 36. The second destination memory locations encompass the destinations into which the second number of bytes are to be transferred. From 36, the method 10 continues to 38. At 38, second destination memory location information is generated. The second destination memory location information encompasses the second destination memory locations into which the second number of bytes are to be stored. The method 10 then proceeds to 40 where the second destination memory location information is stored. The method 10 ends at 42.

[0060] As may be appreciated from Figs. 1-3, method 10 encompasses operations that form a basic halt operation of a DMA engine. In order to resume the transfer of the bytes from the source locations to the destination locations, further processing is required. Figs. 4-9 detail different embodiments and aspect of the invention that are directed to the restart operation.

[0061] As a preliminary matter, the various embodiments of the restart operation encompass a continuation of the halted copying process. In this regard, the copying process continues as before. In the resumed copying operation, the second bytes are copied from the source locations to the destination locations. How the copying is resumed underlies aspects of the embodiments that are described in connection with Figs. 4-9.

[0062] Fig. 4 provides for retrieval of specific information for operation of the DMA engine when the DMA engine resumes the copying operation after issuing the halt instruction at 16. Specifically, Fig. 4 illustrates a flow chart for a method 44 that starts at 46. The method 44 is contemplated to be a continuation of the method 10, as should be apparent to those skilled in the art. In the method 44, the quantity information is retrieved at 48. Then, at 50, the second source memory location information is retrieved. Following this, the second destination memory location information is retrieved at 52. As should be appreciated by those skilled in the art, once the DMA engine retrieves these three pieces of information, the DMA engine may proceed to resume the halted copying operation detailed in Figs. 1-3.

[0063] For the method 44, it is contemplated that the quantity information, the second source memory location information, and the second destination memory location information are read from at least one register in the direct memory access engine. Since the three types of information are being retrieved from a register, the DMA engine is not required to access a memory location. As a result, the DMA engine may proceed more rapidly to resume the halted copying operation than it would if memory locations were to be accessed. As should be apparent to those skilled in the art, efficient processors are designed to avoid memory accesses, where prudent. Access to memory typically accounts for the longest delays when executing instructions.

[0064] As noted above, the quantity information, the second source memory location information, and the second destination memory location information may be read from at least one register. In one contemplated embodiment, the three types of information may be read from a single register. However, other variations are also contemplated. For example, several registers may be employed. This includes two or more registers. The two or more registers may include at least one source register in which the first source memory locations are retained and at least one destination register in which the first destination memory locations are retained. In addition, the two or more registers may encompass a quantity register in which the quantity information is retained.

[0065] Returning to Fig. 4, once the quantity information, the second source memory location information, and the second destination memory location information are retrieved from the register or registers, the method proceeds to 54 where the halted transfer is resumed by initiating copying of the second number of bytes from the second source memory locations to second destination memory locations using the quantity information, the second source memory location information, and the second destination memory location information. The method 44 then ends at 56.

[0066] As noted in the discussion above, there are several different embodiments by which copying of the bytes from the source locations to the destination locations may be made once the copying operation is resumed. The first embodiment refers the “early stop” method 58. A flow chart for the early stop method 58 is provided in Fig. 5. It is noted that the name “early stop method” is provided merely to distinguish the first

embodiment from other embodiments contemplated to fall within the scope of the invention. The name is not intended to be limiting of the method 58. Moreover, while reference is made to other embodiments also by a specific name, the names are not intended to be limiting of the invention. The names are being used merely to facilitate understanding of the invention.

[0067] With reference to Fig. 5, the early stop method 58 is intended to continue after the retrieval of the quantity information, the second source memory location information, and the second destination memory location information. These three operations are discussed at 48, 50, and 52 in connection with the discussion of Fig. 4.

[0068] The method 58 begins at 60. From 60, the method 58 proceeds to 62, where a next address to read from the second source memory location information is determined. The method 58 then proceeds to 64 where a next address to write from the second destination memory location information is determined. All pending reads and writes are then flushed at 66. This means that all of the pending reads and writes are deleted. From 66, the method 58 proceeds to 68, where a decrement value is determined from the next address to read and the next address to write. The decrement value identifies a number of bytes by which the next address to read is in advance of the next address to write. At 70, the next address to read is decremented by the decrement value, thereby generating a decremented read address. Once this decrement read address is determined, the copying operation may resume. Specifically, the second number of bytes may be copied from the second memory locations to the second memory destinations based at least upon the decremented read address, the next address to write, and the quantity information. The method 58 ends at 72.

[0069] Fig 6 illustrates a method 74, which is referred to as the “run-ahead” method”. The run-ahead method 74 also is intended to initiate after the operation 52 that is detailed in Fig. 4. The run-ahead method 74 begins at 76. The method 74 then proceeds to 78 where a next address to read from the second source memory location information is determined. Then, at 80, a next address to write from the second destination memory location information is determined.

[0070] In the method 74, tracking is provided for a run-ahead value. The run ahead value is the number of bytes by which the next address to read is in advance of the

next address to write. By tracking this information, a run-ahead value may be established at 82. The run-ahead value is stored at 84. So that the DMA engine may resume the copying operation, the run-ahead value is retrieved at 86. Obviously this operation occurs at a subsequent time, when it is appropriate to resume the copying operation. Finally, at 88, the next address to read is adjusted by the run-ahead value, thereby generating an adjusted next address to read. Once the next address to read has been determined the method 74 may proceed to resume the halted copying operation. Copying of the second number of bytes from the second memory locations to the second memory destinations then proceeds based at least upon the adjusted next address to read, the next address to write, and the quantity information. The method 74 ends at 90.

[0071] Fig. 7 provides a flow diagram for a third embodiment of the method of the invention. This method is referred to as the “restartable stop” method 92. As indicated in Fig. 7, the method 92 begins at 94. The method 92 then proceeds to 96 where a next address to read from the second source memory location information is determined. Then, at 98, a next address to write from the second destination memory location information is determined. With these two bits of information, copying of the second number of bytes from the second memory locations to the second memory destinations may be resumed using the next address to read, the next address to write, and the quantity information. The method 92 ends at 100.

[0072] Fig. 8 provides a flow diagram for a method referred to herein as the “simple” method 102. The simple method 102 is the fourth embodiment of a method for resuming the halted copying instruction. The method 102 also is intended to begin after the operation 40, detailed in Fig. 3.

[0073] With reference to Fig. 8, the simple method 102 begins at 104. From 104, the method 102 proceeds to 106 where an identification of the first source memory locations is retained. Then, at 108, an identification of the first destination memory locations is retained. Following this, at 110, a count of the first number of bytes is retained. At a subsequent moment in time, prior to resuming the halted copying operation, the method 102 retrieves the quantity information at 112. At 114, the method 102 establishes a next address to read at least based upon the identification of the first source memory locations, the count of the first number of bytes, and the quantity

information. At 116, the method 102 establishes a next address to write at least based upon the identification of the first destination memory locations, the count of the first number of bytes, and the quantity information. From this, the method 102 may resume the halted copying. The resumed copying in this method relies at least upon the next address to read, the next address to write, the count of the first number of bytes, and the quantity information. The method 102 ends at 118.

[0074] Fig. 9 illustrates a method 120, which is referred to herein as the “drain” method. This is a fifth embodiment contemplated to fall within the scope of the invention. Like the method 102, the drain method 120 is intended to follow the method 10 illustrated in Figs. 1-3.

[0075] As illustrated in Fig. 9, the drain method 120 begins at 122. At 124, the method 120 retains an indication of the next address to read. At 126, the method 120 retains an indication of the next address to write. At 128, after the halt instruction, the method 120 stops reading of the first number of bytes from the first source memory locations. At 130, the method 120 stops writing any bytes to the first destination memory locations that remain from the bytes read before stopping the reading. Then, at 132, the method 120 determines a next address to read from the second source memory location information. At 134, a next address to write from the second destination memory location information is determined. As a result, copying of the second number of bytes from the second memory locations to the second memory destinations may proceed at least based upon the next address to read, the next address to write, and the quantity information. The method 120 ends at 136.

[0076] As noted above, the method of the invention may be applied in the context of scatter and/or gather operations. In the gather context, the method may include an operation where the first number of bytes are retrieved from the first source memory locations, which are distributed in a plurality of buffers. Then, the first number of bytes may be buffered into a single data stream. A scatter operation may be employed after this. If so, the method may include the additional operation of providing the first number of bytes to the first destination memory locations from the single data stream. In this context, the first destination memory locations may be distributed in a plurality of buffers. Alternatively, a gather operation may pull information from a plurality of first

destination memory locations directly. Distribution may then be directed to a plurality of buffers.

[0077] As noted above, the invention has been described in connection with several specific embodiments. It is not intended for the invention to be limited solely to the embodiments described. To the contrary, the invention is intended to encompass any equivalents and variations, as should be apparent to those skilled in the art.

What is claimed is:

1. A method of operating a direct memory access engine, comprising:
 - initiating copying of a first number of bytes from first source memory locations to first destination memory locations;
 - after initiating the copying, issuing a halt instruction before the first number of bytes are copied from the first source memory locations to the first destination memory locations;
 - in response to issuance of the halt instruction, halting the copying, thereby establishing a second number of bytes, which comprise those bytes remaining to be copied from the first number of bytes;
 - identifying a quantity of the second number of bytes;
 - generating quantity information concerning the quantity of the second number of bytes;
 - storing the quantity information;
 - identifying second source memory locations in which the second number of bytes are stored;
 - generating second source memory location information encompassing the second source memory locations in which the second number of bytes are stored;
 - storing the second memory source location information;
 - identifying second destination memory locations into which the second number of bytes are to be transferred;
 - generating second destination memory location information encompassing the second destination memory locations into which the second number of bytes are to be stored; and
 - storing the second destination memory location information.
2. The method of claim 1, further comprising:
 - retrieving the quantity information;
 - retrieving the second source memory location information; and
 - retrieving the second destination memory location information,

wherein the quantity information, the second source memory location information, and the second destination memory location information are read from at least one register in the direct memory access engine.

3. The method of claim 2, wherein the at least one register comprises a plurality of registers.
4. The method of claim 3, wherein the plurality of registers comprise:
 - at least one source register in which the first source memory locations are retained; and
 - at least one destination register in which the first destination memory locations are retained.
5. The method of claim 4, wherein the plurality of registers further comprise:
 - a quantity register in which the quantity information is retained.
6. The method of claim 2, further comprising:
 - resuming the halted transfer by initiating copying of the second number of bytes from the second source memory locations to second destination memory locations using the quantity information, the second source memory location information, and the second destination memory location information .
7. The method of claim 6, further comprising:
 - determining a next address to read from the second source memory location information;
 - determining a next address to write from the second destination memory location information;
 - flushing all pending reads and writes;
 - determining, from the next address to read and the next address to write, a decrement value, wherein the decrement value identifies a number of bytes by which the next address to read is in advance of the next address to write; and

decrementing the next address to read by the decrement value, thereby generating a decremented read address,

wherein copying the second number of bytes from the second memory locations to the second memory destinations proceeds based at least upon the decremented read address, the next address to write, and the quantity information.

8. The method of claim 6, wherein further comprising:

determining a next address to read from the second source memory location information;

determining a next address to write from the second destination memory location information;

tracking a number of bytes by which the next address to read is in advance of the next address to write, thereby establishing a run-ahead value;

storing the run-ahead value;

retrieving the run-ahead value; and

adjusting the next address to read by the run-ahead value, thereby generating an adjusted next address to read,

wherein copying the second number of bytes from the second memory locations to the second memory destinations proceeds based at least upon the adjusted next address to read, the next address to write, and the quantity information.

9. The method of claim 6, further comprising:

determining a next address to read from the second source memory location information; and

determining a next address to write from the second destination memory location information,

wherein copying the second number of bytes from the second memory locations to the second memory destinations proceeds using the next address to read, the next address to write, and the quantity information.

10. The method of claim 1, further comprising:

retaining an identification of the first source memory locations;
retaining an identification of the first destination memory locations;
retaining a count of the first number of bytes;
retrieving the quantity information;

establishing a next address to read at least based upon the identification of the first source memory locations, the count of the first number of bytes, and the quantity information; and

establishing a next address to write at least based upon the identification of the first destination memory locations, the count of the first number of bytes, and the quantity information,

wherein copying the second number of bytes from the second memory locations to the second memory destinations relies at least upon the next address to read, the next address to write, the count of the first number of bytes, and the quantity information.

11. The method of claim 1, further comprising:

retaining an indication of the next address to read;
retaining an indication of the next address to write;
after issuance of the halt instruction,

stopping reading of the first number of bytes from the first source memory locations, and

writing any bytes to the first destination memory locations that remain from the bytes read before stopping the reading,

determining a next address to read from the second source memory location information; and

determining a next address to write from the second destination memory location information,

wherein copying the second number of bytes from the second memory locations to the second memory destinations relies at least upon the next address to read, the next address to write, and the quantity information.

12. The method of claim 1, further comprising:

retrieving the first number of bytes from the first source memory locations, which are distributed in a plurality of buffers; and

funneling the first number of bytes into a single data stream.

13. The method of claim 12, further comprising:

providing the first number of bytes to the first destination memory locations from the single data stream,

wherein the first destination memory locations are distributed in a plurality of buffers.

14. The method of claim 1, wherein the first destination memory locations are distributed in a plurality of buffers.

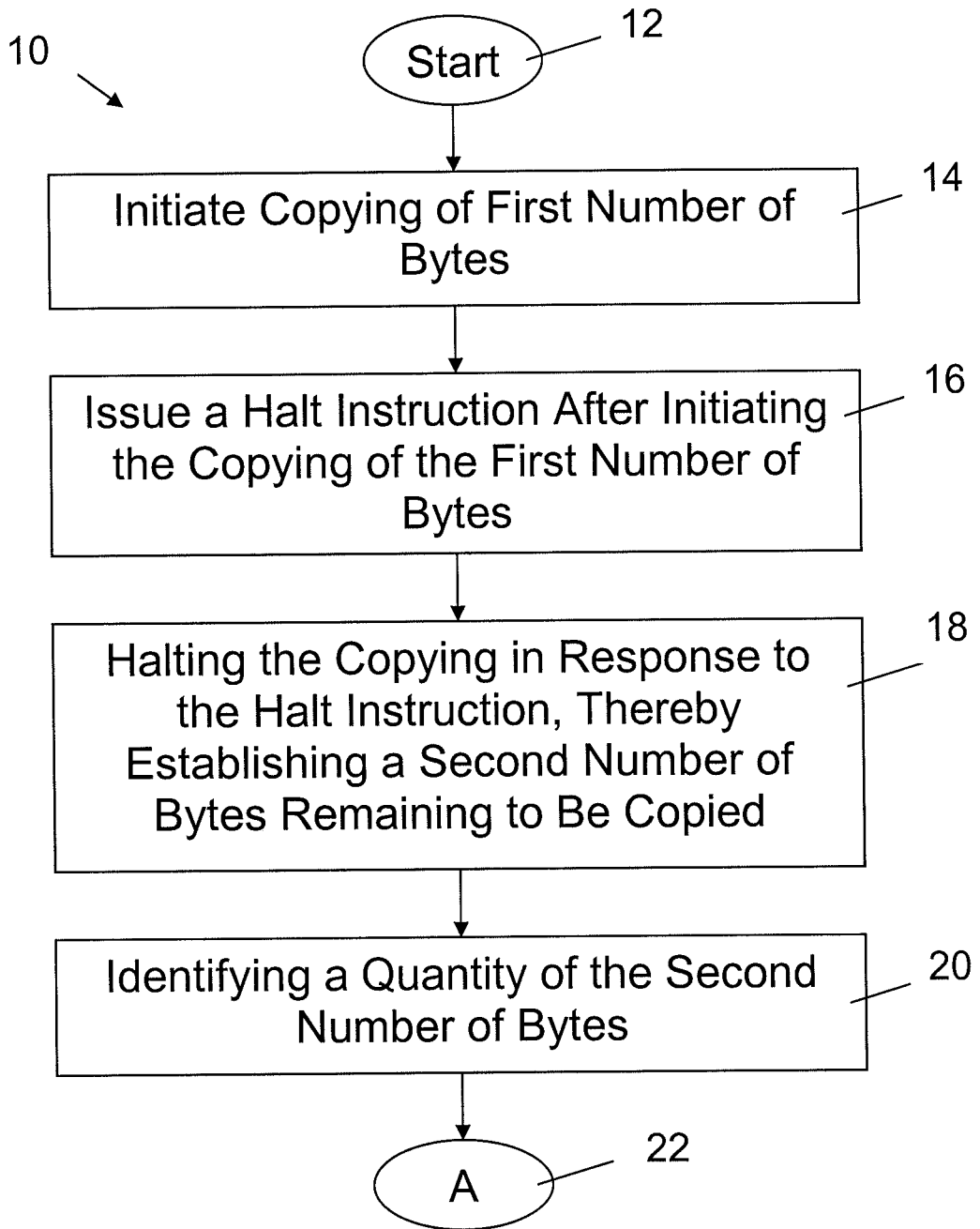


FIG. 1

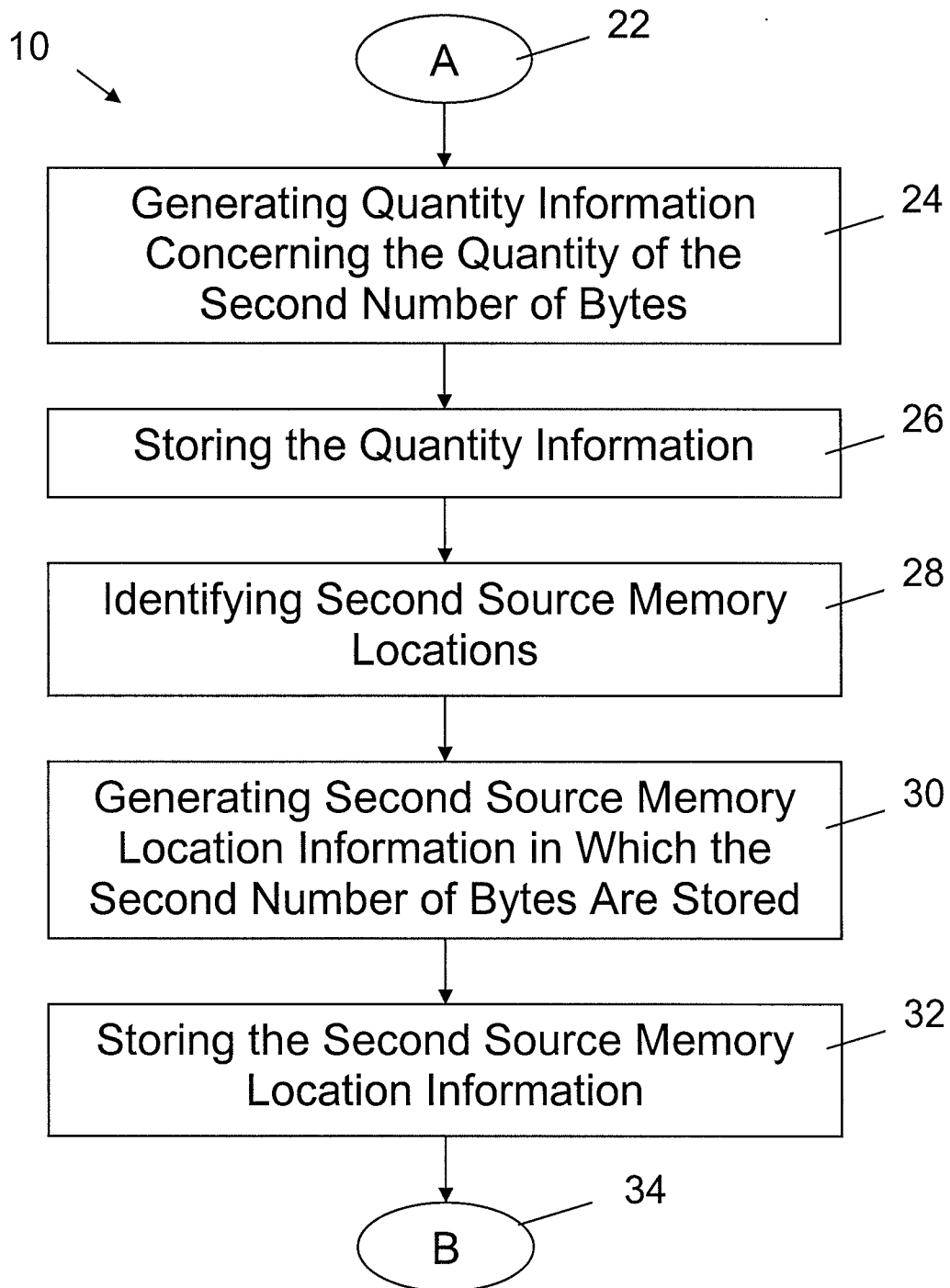
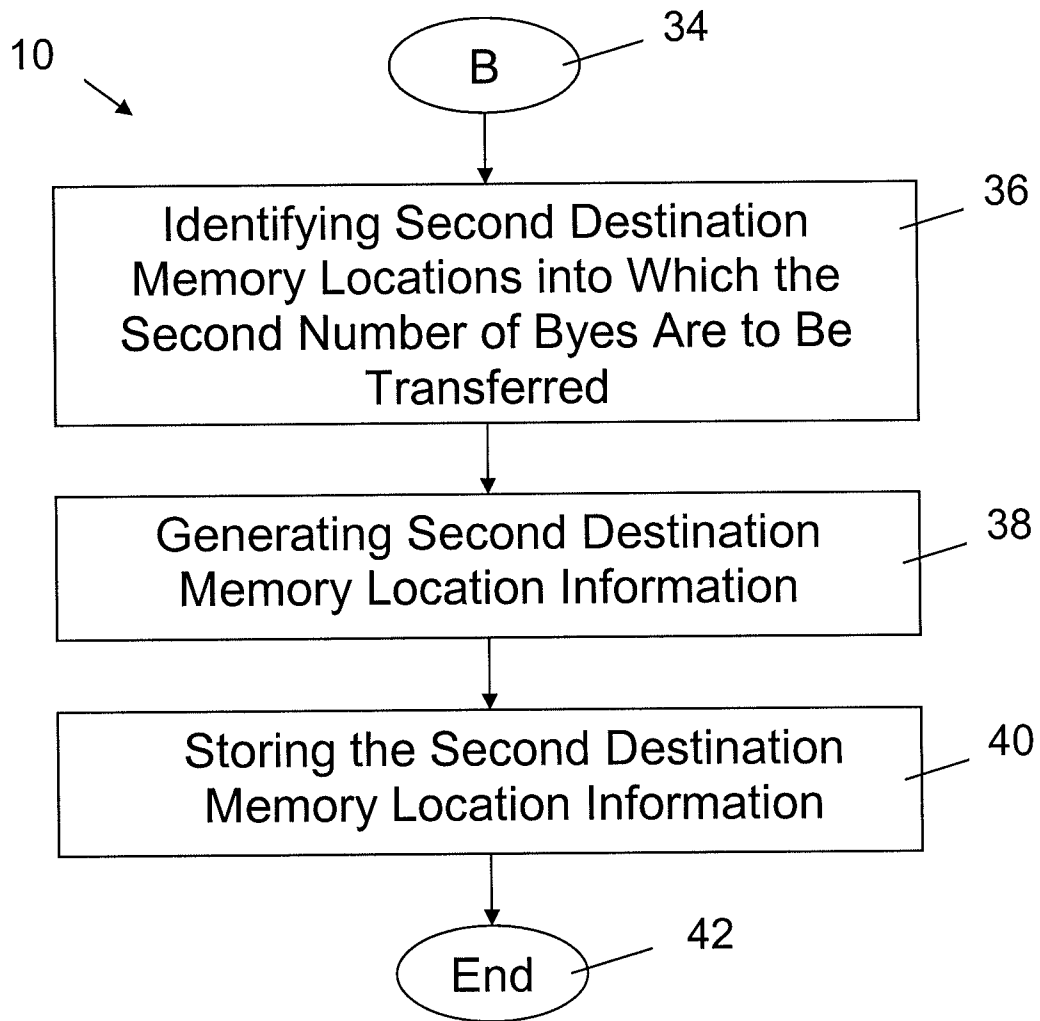
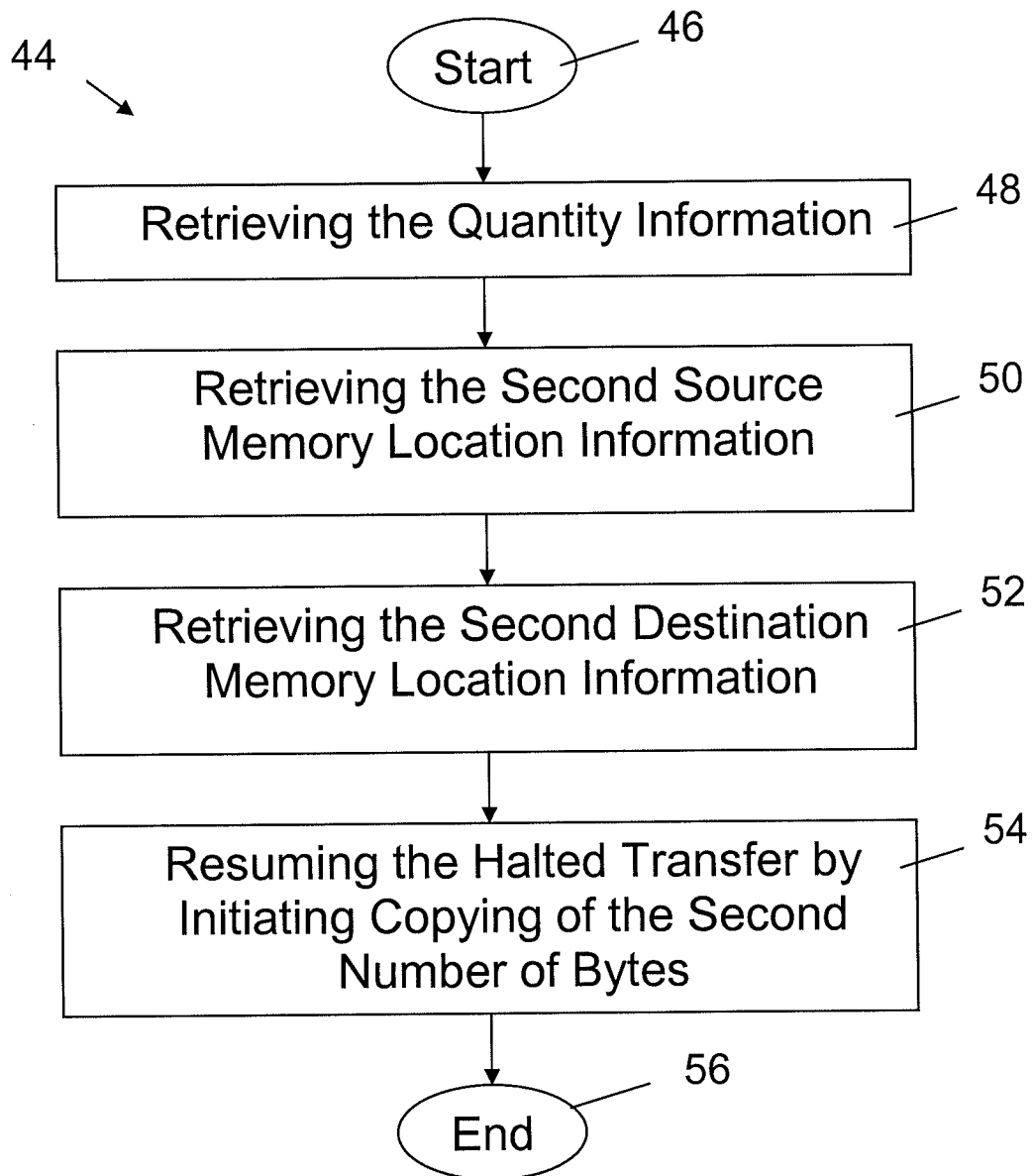


FIG. 2

**FIG. 3**

**FIG. 4**

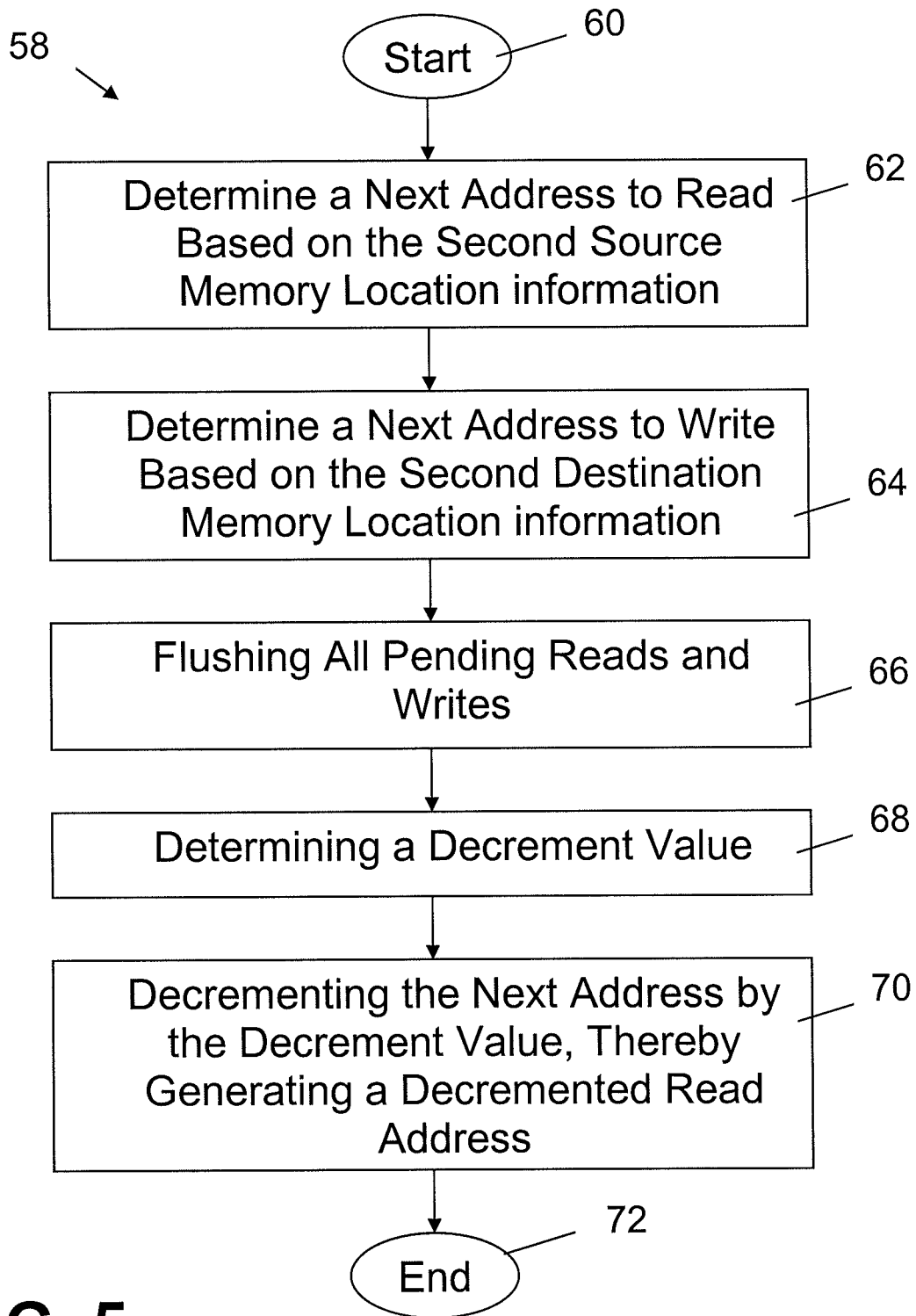


FIG. 5

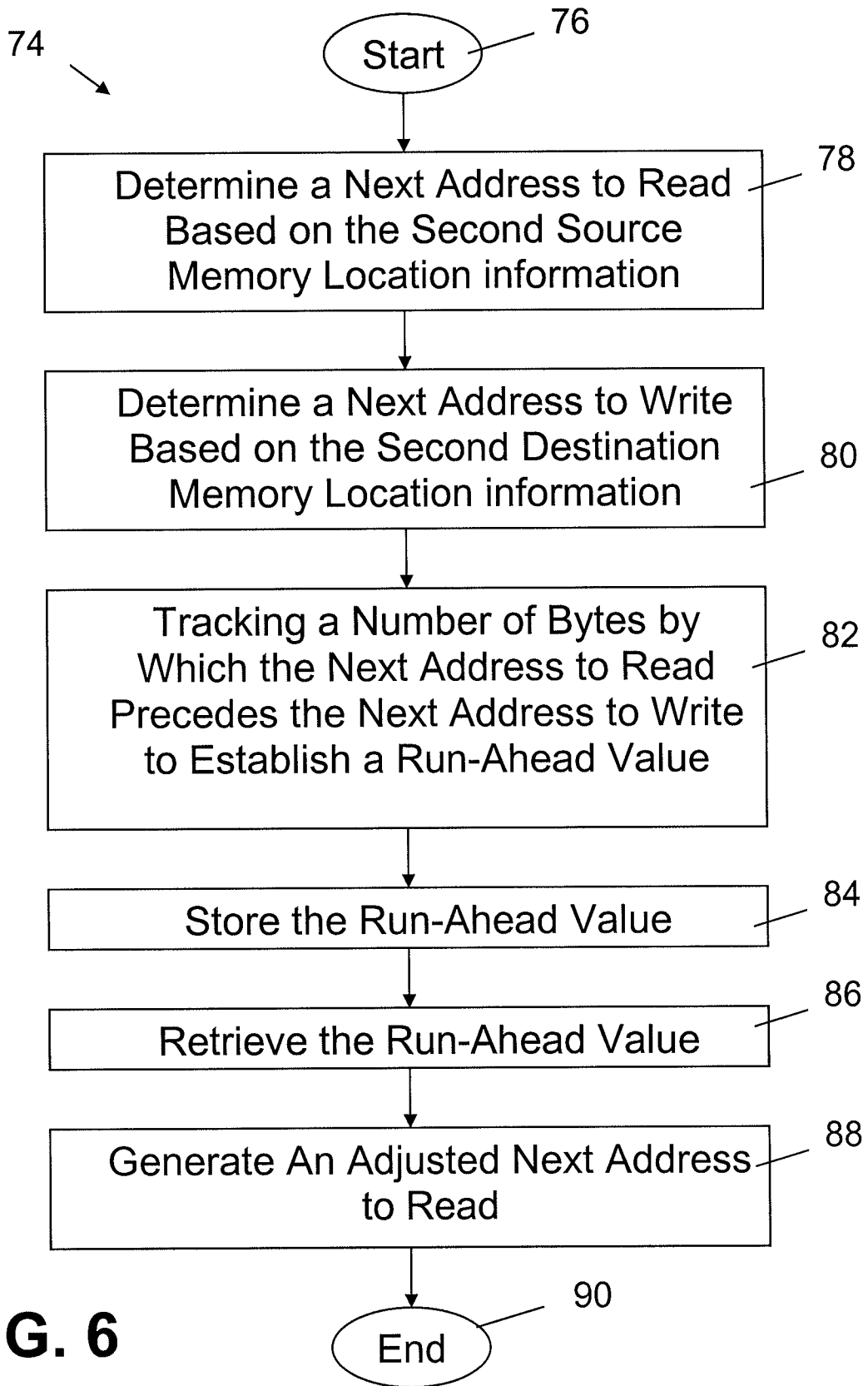


FIG. 6

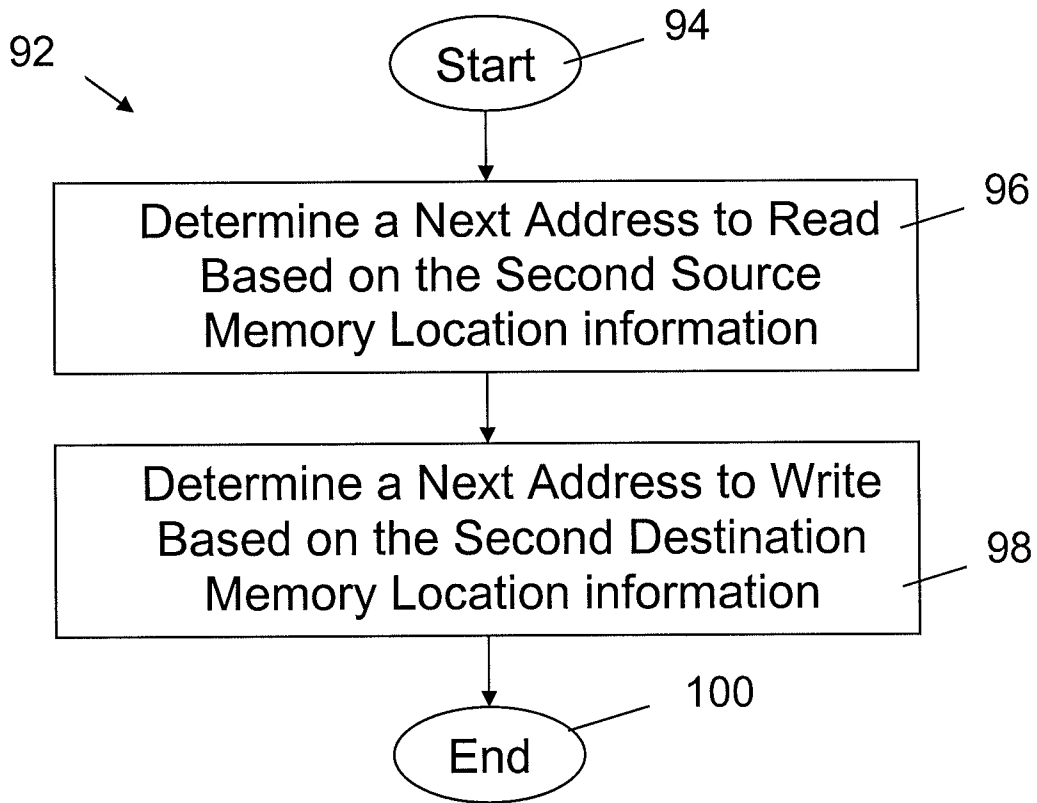


FIG. 7

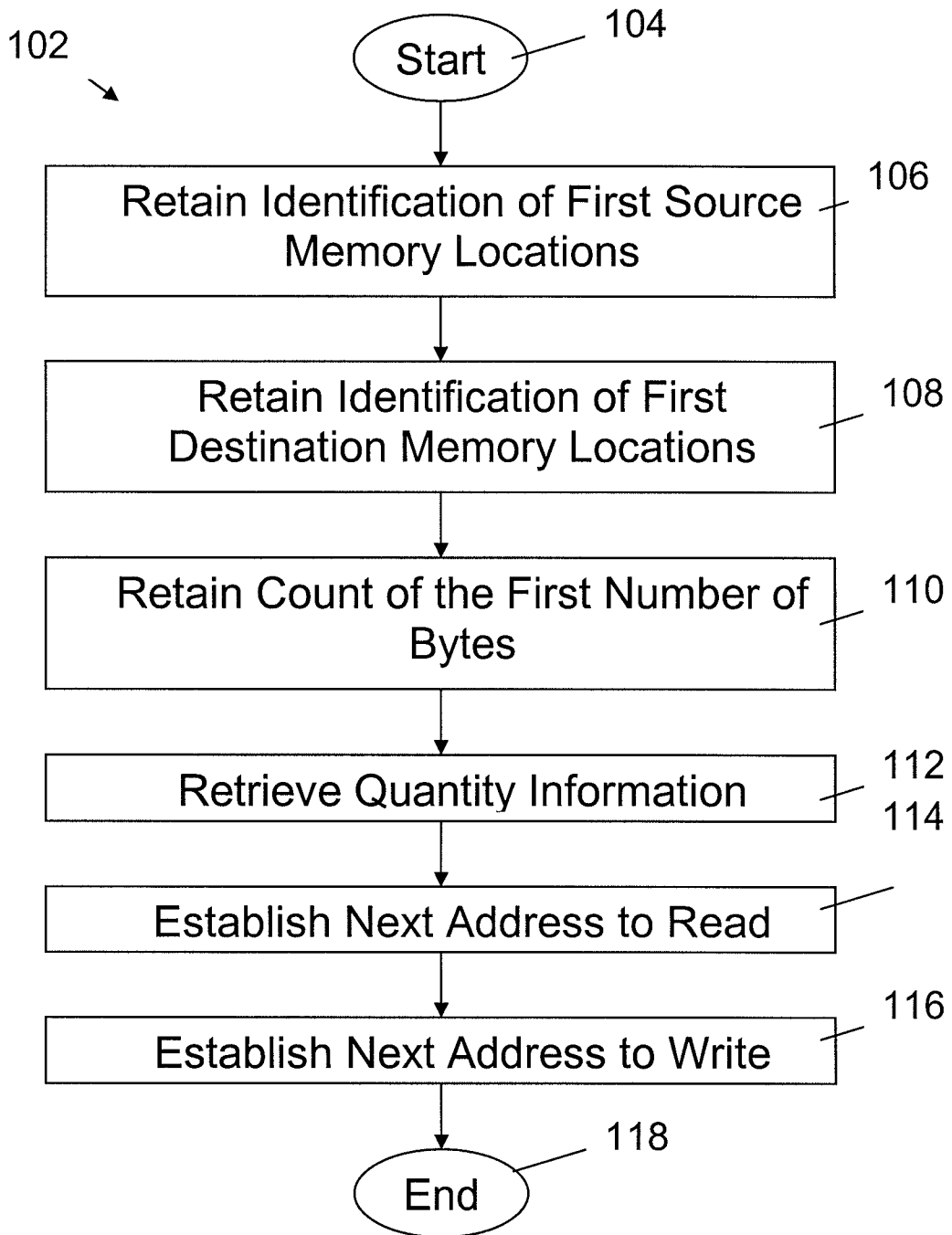


FIG. 8

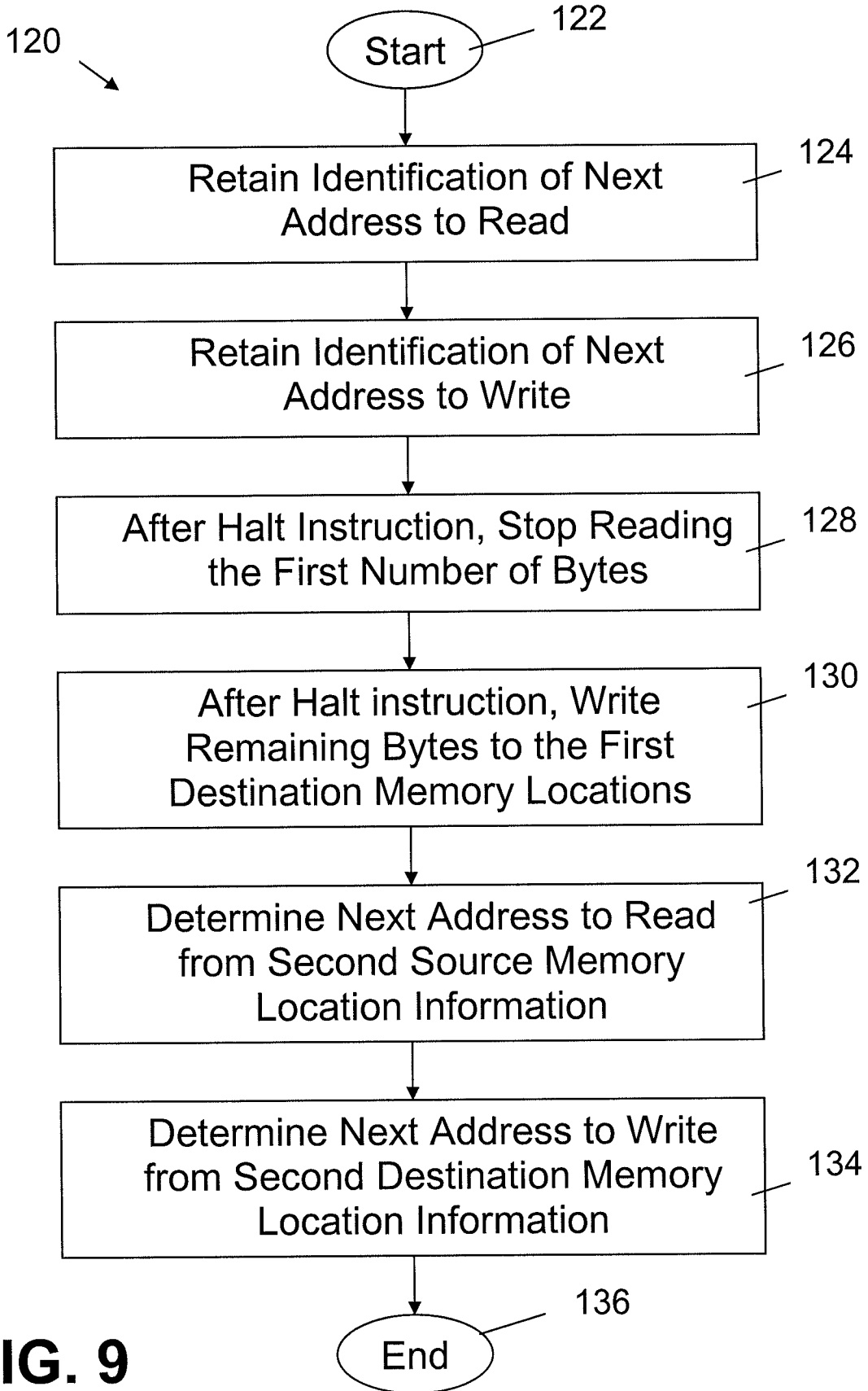


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 09/52794

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 13/28 (2009.01) USPC - 710/22 According to International Patent Classification (IPC) or to both national classification and IPC</p>																																																
<p>B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC(8) - G06F 13/28 (2009.01) USPC - 710/22</p> <p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched USPC - 710/22,28,36,40-41,1; 709/230 - search terms below.</p> <p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) Google Scholar; WEST (PGPB,USPT,EPAB, JPAB) - direct memory access, DMA, process state, location, address, source, destination, halt, stop, instruction, quantity, byte, number, count, register, copy, transfer, resume, continue, restart, flush, decrement, read, write, run-ahead, value, adjust, buffer, data stream, funnel, interrupt, pause, command.</p>																																																
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1"> <thead> <tr> <th>Category*</th> <th>Citation of document, with indication, where appropriate, of the relevant passages</th> <th>Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>US 5,325,489 A (MITSUHIRA et al.) 28 June 1994 (28.06.1994) ? col 1, ln 30-44; col 1, ln 62 to col 2, ln 18; col 2, ln 30-43; col 3, ln 18 to col 4, ln 5; col 4, ln 10-24; col 5, ln 34-52; col 6, ln 1-17; col 6, ln 21-55.</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 2007/0073925 A1 (LIM et al.) 29 March 2007 (29.03.2007).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 2004/0243739 A1 (SPENCER) 02 December 2004 (02.12.2004).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 7,380,114 B2 (SANE et al.) 27 May 2008 (27.05.2008).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 6,065,071 A (PRIEM et al.) 16 May 2000 (16.05.2000).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 5,623,622 A (YUKI et al.) 22 April 1997 (22.04.1997).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 5,497,501 A (KOHZONO et al.) 05 March 1996 (05.03.1996).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 5,481,756 A (KANNO) 02 January 1996 (02.01.1996).</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 5,251,312 A (SODOS) 05 October 1993 (05.10.1993).</td> <td>1-14</td> </tr> </tbody> </table> <p><input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/></p> <table border="1"> <tr> <td>* Special categories of cited documents:</td> <td>"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</td> </tr> <tr> <td>"A" document defining the general state of the art which is not considered to be of particular relevance</td> <td>"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</td> </tr> <tr> <td>"E" earlier application or patent but published on or after the international filing date</td> <td>"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</td> </tr> <tr> <td>"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)</td> <td>"&" document member of the same patent family</td> </tr> <tr> <td>"O" document referring to an oral disclosure, use, exhibition or other means</td> <td></td> </tr> <tr> <td>"P" document published prior to the international filing date but later than the priority date claimed</td> <td></td> </tr> </table> <table border="1"> <tr> <td>Date of the actual completion of the international search 07 September 2009 (07.09.2009)</td> <td>Date of mailing of the international search report 23 SEP 2009</td> </tr> <tr> <td>Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201</td> <td>Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774</td> </tr> </table>			Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	X	US 5,325,489 A (MITSUHIRA et al.) 28 June 1994 (28.06.1994) ? col 1, ln 30-44; col 1, ln 62 to col 2, ln 18; col 2, ln 30-43; col 3, ln 18 to col 4, ln 5; col 4, ln 10-24; col 5, ln 34-52; col 6, ln 1-17; col 6, ln 21-55.	1-14	A	US 2007/0073925 A1 (LIM et al.) 29 March 2007 (29.03.2007).	1-14	A	US 2004/0243739 A1 (SPENCER) 02 December 2004 (02.12.2004).	1-14	A	US 7,380,114 B2 (SANE et al.) 27 May 2008 (27.05.2008).	1-14	A	US 6,065,071 A (PRIEM et al.) 16 May 2000 (16.05.2000).	1-14	A	US 5,623,622 A (YUKI et al.) 22 April 1997 (22.04.1997).	1-14	A	US 5,497,501 A (KOHZONO et al.) 05 March 1996 (05.03.1996).	1-14	A	US 5,481,756 A (KANNO) 02 January 1996 (02.01.1996).	1-14	A	US 5,251,312 A (SODOS) 05 October 1993 (05.10.1993).	1-14	* Special categories of cited documents:	"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	"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone	"E" earlier application or patent but published on or after the international filing date	"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	"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)	"&" document member of the same patent family	"O" document referring to an oral disclosure, use, exhibition or other means		"P" document published prior to the international filing date but later than the priority date claimed		Date of the actual completion of the international search 07 September 2009 (07.09.2009)	Date of mailing of the international search report 23 SEP 2009	Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.																																														
X	US 5,325,489 A (MITSUHIRA et al.) 28 June 1994 (28.06.1994) ? col 1, ln 30-44; col 1, ln 62 to col 2, ln 18; col 2, ln 30-43; col 3, ln 18 to col 4, ln 5; col 4, ln 10-24; col 5, ln 34-52; col 6, ln 1-17; col 6, ln 21-55.	1-14																																														
A	US 2007/0073925 A1 (LIM et al.) 29 March 2007 (29.03.2007).	1-14																																														
A	US 2004/0243739 A1 (SPENCER) 02 December 2004 (02.12.2004).	1-14																																														
A	US 7,380,114 B2 (SANE et al.) 27 May 2008 (27.05.2008).	1-14																																														
A	US 6,065,071 A (PRIEM et al.) 16 May 2000 (16.05.2000).	1-14																																														
A	US 5,623,622 A (YUKI et al.) 22 April 1997 (22.04.1997).	1-14																																														
A	US 5,497,501 A (KOHZONO et al.) 05 March 1996 (05.03.1996).	1-14																																														
A	US 5,481,756 A (KANNO) 02 January 1996 (02.01.1996).	1-14																																														
A	US 5,251,312 A (SODOS) 05 October 1993 (05.10.1993).	1-14																																														
* Special categories of cited documents:	"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																																															
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone																																															
"E" earlier application or patent but published on or after the international filing date	"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																																															
"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)	"&" document member of the same patent family																																															
"O" document referring to an oral disclosure, use, exhibition or other means																																																
"P" document published prior to the international filing date but later than the priority date claimed																																																
Date of the actual completion of the international search 07 September 2009 (07.09.2009)	Date of mailing of the international search report 23 SEP 2009																																															
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774																																															