



- (51) International Patent Classification:
G06F 13/00 (2006.01)
- (21) International Application Number:
PCT/US2012/046236
- (22) International Filing Date:
11 July 2012 (11.07.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/184,647 18 July 2011 (18.07.2011) US
- (71) Applicant (for all designated States except US): **APPLE INC.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **WAKRAT, Nir, J.** [IL/US]; 1 Infinite Loop, Cupertino, CA 95014 (US). **POST, Daniel, J.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).
- (74) Agents: **VAN COURT, Andrew** et al.; Kramer Levin Naf-talis & Frankel LLP, 1177 Avenue of The Americas, New York, NY 10036 (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, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

(54) Title: NON-VOLATILE TEMPORARY DATA HANDLING

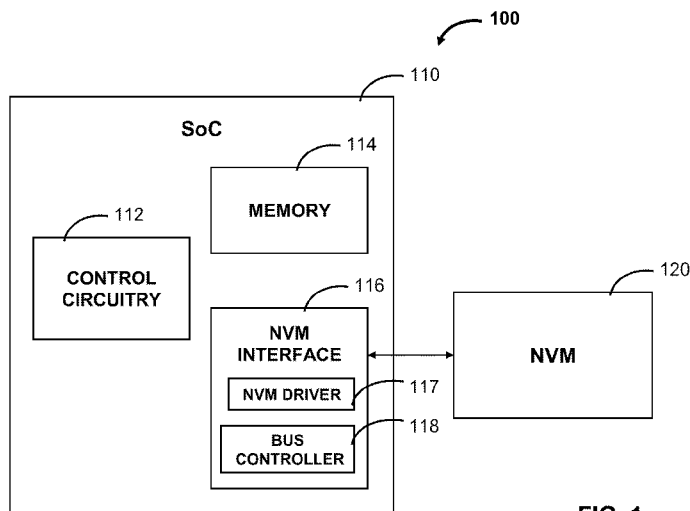


FIG. 1

(57) Abstract: Systems and methods are provided for handling temporary data that is stored in a non-volatile memory, such as NAND flash memory. The temporary data may include hibernation data or any other data needed for only one boot cycle of an electronic device. When storing the temporary data in one or more pages of the non-volatile memory, the electronic device can store a temporary marker as part of the metadata in at least one of the pages. This way, on the next bootup of the electronic device, the electronic device can use the temporary marker to determine that the associated page contains unneeded data. The electronic device can therefore invalidate the page and omit the page from its metadata tables.

WO 2013/012632 A1

- 1 -

NON-VOLATILE TEMPORARY DATA HANDLING

Background of the Disclosure

[0001] NAND flash memory, as well as other types of non-volatile memories ("NVMs"), are commonly used in electronic device for mass storage. For example, consumer electronics such as portable media players, often include NAND flash memory to store music, videos, and other media programs. Such data is typically intended to be stored for long periods of time, such as on the order of days, months, or even years.

[0002] NVMs may also be used in electronic devices for purposes other than long-term storage. For example, the electronic device may temporarily store data from a volatile memory into a NVM to enable the volatile memory to power down during a reduced power, hibernation state without losing the data. As another example, a NVM can be used in virtual memory schemes to increase the apparent size of the electronic device's volatile memory.

Summary of the Disclosure

[0003] Systems and methods are provided for processing data intended for short-term storage in a non-volatile memory, such as a flash memory. By

- 2 -

allowing short-term storage to be handled differently from long-term data, the memory management efficiency and effectiveness of a NVM can be increased.

[0004] Short-term storage may include the storage of
5 data in a NVM that is needed only during a current instance of an operating system, and is no longer useful once the electronic device is shut down or rebooted (or even earlier). This data may be referred to as "non-volatile temporary data" or "NV temporary
10 data," since the data is needed temporarily, but is stored on a persistent non-volatile medium. NV temporary data can include, for example, hardware-based hibernation data, application-generated data, application-based hibernation data, swap files, and
15 other temporary files.

[0005] In some embodiments, when storing NV temporary data, the electronic device can decorate the NV temporary data with a temporary marker. For example, for each page of the NVM in which the
20 electronic device stores NV temporary data, the electronic device can include a temporary marker in the metadata of the page. Alternatively, the electronic device can provide a temporary marker in at least one page of a block storing NV temporary data. This allows
25 the electronic device to distinguish the NV temporary data from information needed across multiple device bootups.

[0006] During bootup of the electronic device, the electronic device may be configured to perform various
30 startup procedures, such as constructing metadata tables. The metadata tables can include, for example, a table indicating which logical addresses issued by the file system (e.g., logical block addresses

- 3 -

("LBAs")) are currently allocated and/or a logical-to-physical address map. The electronic device may create the metadata tables by scanning through the pages of the NVM and determining whether each page includes a temporary marker (or whether each page is in a block marked for storing NV temporary data). If a page includes a temporary marker, the data included in the page was useful only in the previous boot of the electronic device and is not needed in the current boot. Thus, the electronic device can mark the page as having invalid data and omit the page (and its stored metadata) from the metadata tables. By performing such cleanup operations at bootup, the electronic device can ensure efficient use of the NVM memory locations from the outset. That is, at no point during the current boot cycle will the electronic device perform needless operations during garbage collection or wear leveling to maintain the obsolete NV temporary data.

[0007] In some embodiments, the electronic device can select memory locations and/or programming techniques for storing data based on whether the data is NV temporary data. For example, the electronic device can select lower-performance memory locations, such as less reliable pages or blocks (e.g., high cycled blocks), to store NV temporary data. In some embodiments, the electronic device can select higher-speed memory locations or programming techniques for storing NV temporary data, such as single-level cell ("SLC") blocks.

30 Brief Description of the Drawings

[0008] The above and other aspects and advantages of the invention will be apparent upon consideration of

- 4 -

the following detailed description, taken in conjunction with accompanying drawings, in which like reference characters refer to like parts throughout, and in which:

5 [0009] FIG. 1 is a block diagram of an illustrative electronic device having a non-volatile memory configured in accordance with various embodiments of the invention;

10 [0010] FIGS. 2 and 3 are graphical representations of illustrative non-volatile memory locations storing non-volatile temporary data in accordance with various embodiments of the invention;

15 [0011] FIG. 4 is a flowchart of an illustrative process for decorating non-volatile temporary data in a non-volatile memory with a temporary marker in accordance with various embodiments of the invention; and

20 [0012] FIG. 5 is a flowchart of an illustrative process for processing non-volatile temporary data at bootup of an electronic device in accordance with various embodiments of the invention.

Detailed Description

25 [0013] FIG. 1 is a block diagram of illustrative electronic device 100. In some embodiments, electronic device 100 can be or can include a portable media player (e.g., an iPod™ made available by Apple Inc. of Cupertino, CA), a cellular telephone (e.g., an iPhone™ made available by Apple Inc.), a pocket-sized personal computer, a personal digital assistance ("PDA"), a 30 desktop computer, a laptop computer, and any other suitable type of electronic device or system.

- 5 -

[0014] Electronic device 100 can include system-on-a-chip ("SoC") 110 and non-volatile memory ("NVM") 120. NVM 120 can include a NAND flash memory based on floating gate or charge trapping technology, NOR flash
5 memory, erasable programmable read only memory ("EPROM"), electrically erasable programmable read only memory ("EEPROM"), Ferroelectric RAM ("FRAM"), magnetoresistive RAM ("MRAM"), or any combination thereof.

10 [0015] NVM 120 can be organized into "blocks", which is the smallest erasable unit, and further organized into "pages," which may be the smallest unit that can be programmed or read. In some embodiments, NVM 120 can include multiple integrated circuits, where each
15 integrated circuit may have multiple blocks. The blocks from corresponding integrated circuits (e.g., blocks having the same position or block number) may form logical units referred to as "super blocks." Each memory location (e.g., page or block) of NVM 120 can be
20 addressed using a physical address (e.g., a physical page address or physical block address). While only one NVM is shown in FIG. 1, electronic device 100 can alternatively include multiple NVM packages.

[0016] System-on-a-chip 110 can include control
25 circuitry 112, memory 114, and NVM interface 116. Control circuitry 112 can control the general operations and functions of SoC 110 and electronic device 100 in general. Control circuitry 112 can include any suitable components, circuitry, or logic,
30 such as one or more processors. Control circuitry 112 may operate under the control of a program, such as an application, operating system, or a NVM driver (e.g., NVM driver 117) loaded in memory 114.

- 6 -

[0017] Memory 114 can include any suitable type of volatile memory, such as random access memory ("RAM") (e.g., static RAM ("SRAM"), dynamic random access memory ("DRAM"), synchronous dynamic random access
5 memory ("SDRAM"), double-data-rate ("DDR") RAM), cache memory, or any combination thereof. In some embodiments, memory 114 may act as the main memory for any processors implemented as part of control circuitry 112. In these and other embodiments,
10 memory 114 can store data that may also be stored, at one point or another (i.e. concurrently or at different times), in NVM 120 as NV temporary data. As described above, NV temporary data includes data stored in a NVM that is needed only during a current instance of the
15 operating system, and is no longer useful once electronic device 100 is shut down or rebooted (or earlier).

[0018] For example, memory 114 and/or NVM 120 can store NV temporary data in the form of hardware-based
20 hibernation data. Hardware-based hibernation data may include data transferred from memory 114 to NVM 120 responsive to entering a hardware-based hibernation state (e.g., a reduced power state in which memory 114 is powered off). Thus, hardware-based hibernation data
25 can include data that enables electronic device 100 to return to its pre-hibernation operational state (e.g., the state it was in prior to entering into the reduced power state) when it wakes up from the hibernation state. The hardware-based hibernation data may
30 represent the current operational states of the device, for both hardware and software. For example, device state data may specify which programs are actively running, or more specifically, which media asset (e.g.,

- 7 -

song) is being played back, or the position within the graphical user interface the user is currently accessing. Because a user would not expect electronic device 100 to return to its pre-hibernation operational state should the user shut down electronic device 100 during hardware hibernation, such hardware-based hibernation data is an example of NV temporary data.

5
[0019] In some embodiments, memory 114 and/or NVM 120 can store application-generated data as NV temporary data. The application-generated data may include any temporarily useful data associated with or generated by an application, such as data pertaining to a current game in a gaming application. In some embodiments, the application can provide an indication to the operating system and file system of the temporary nature of the application-generated data so that this data can ultimately be stored as NV temporary data in NVM 120.

10
[0020] In some scenarios, memory 114 and/or NVM 120 can be used to store NV temporary data in the form of application-based hibernation data. Application-based hibernation data may include data, such as application-generated data, transferred from memory 114 to NVM 120 responsive to a particular application or program entering into a hibernation state. Electronic device 100 can select the application for hibernation to free up space in memory 114 when space is running low and the application is not being actively used. Thus, the application-based hibernation data can include data that enables electronic device 100 to return a hibernated application back to its pre-hibernation operational state. Because a user would not expect hibernating application to return its exact

15
20
25
30

- 8 -

former state if electronic device 100 were rebooted, application-based hibernation data is another example of NV temporary data.

[0021] In these and other scenarios, memory 114
5 and/or NVM 120 can be used for storing NV temporary data in connection with virtual memory schemes. For example, the application-generated data, application-based hibernation data, or any other data used by one or more applications can be transferred from memory 114
10 to a "swap file" or "page file" in NVM 120. Here, data can be paged out of memory 114 into a file stored in NVM 120, thereby freeing up space in memory 114 when space is needed. Other file types used in virtual memory schemes to convey NV temporary data between
15 memory 114 and NVM 120 may also be contemplated. Because the data used in a virtual memory scheme is generally not needed once electronic device 100 is shut down, a swap file is another example of NV temporary data.

[0022] Memory 114 and/or NVM 120 may store NV
20 temporary data of any other suitable type, such as any data typically stored in a "temp file" on UNIX-based systems. For example, NV temporary data can include data stored in temporary files on NVM 120 that one
25 program or application creates to pass the data to another program or application. As another example, NV temporary data can include short-term data stored in temporary files on NVM 120 that is too large to fit into memory 114 or would take up more than a
30 predetermined amount of space in memory 114.

[0023] To enable the components of SoC 110 (such as memory 114) to pass information to and from NVM 120, SoC 110 can include NVM interface 116. NVM

- 9 -

interface 116 may include any suitable combination of hardware, software, and firmware configured to act as an interface or driver between NVM 120 and the non-NVM-specific components of SoC 110. For example, NVM

5 interface 116 can include NVM driver 117 to provide a software/firmware interface that gives the operating system and file system indirect access to NVM 120, thereby allowing the operating system and file system to issue read or write requests to store or retrieve NV

10 temporary data in or from NVM 120. NVM interface 116 can further include bus controller 118, which may include any suitable hardware components that enable NVM driver 117 to access NVM 120 using the bus specifications (e.g., data rate) of NVM 120.

15 **[0024]** NVM interface 116 may perform various tasks to manage the memory locations of NVM 120, such as garbage collection, wear leveling, and bad block management. NVM driver 117 may be configured to maintain or generate "metadata," which can be any

20 memory management data used by NVM driver 117 to manage NVM 120 and the memory locations therein. NVM driver 117 may be configured to maintain metadata tables or mappings, such as a table indicating which

25 logical addresses issued by the file system (e.g., logical block addresses ("LBAs")) are currently allocated for use and/or a mapping between logical addresses and physical addresses of NVM 120. Logical addresses are issued by the file system on read or write requests, which NVM driver 170 may map to

30 different physical addresses that actually correspond to physical memory locations of NVM 120. NVM driver 117 can therefore maintain the metadata tables

- 10 -

in order to properly handle the read and write requests from the file system.

[0025] NVM interface 116 is depicted in FIG. 1 as being implemented completely on SoC 110. In some
5 embodiments, some of the components of NVM interface 116, as well as some of the functionality of NVM driver 117, may be implemented and performed by a separate memory controller (e.g., flash controller) included, for example, in NVM 120. Thus, it should be
10 understood that any descriptions of NVM interface 116 or NVM-related functionality are not limited to components or actions performed on SoC 110.

[0026] NVM interface 116 may receive write requests from the file system to store any of a variety of types
15 of data. For example, the file system may instruct NVM interface 116 to store media assets, such as songs and videos, for an indefinite period of time. In other scenarios, the file system may instruct NVM interface 116 to store NV temporary data, such as
20 hardware-based hibernation data, application-generated data, application-based hibernation data, swap files, or other temporary files, in NVM 120. Because NV temporary data has such a different expected lifespan (i.e. one boot cycle of electronic device 100) than
25 other forms of data (i.e. more than one boot cycle of electronic device 100), NVM interface 116 may be configured to handle NV temporary data in a different manner that can increase the effectiveness and efficiency of memory use in NVM 120.

[0027] In some embodiments, NVM interface 116 may
30 select memory locations (i.e. pages, blocks, or super blocks) in which to store data based on whether the data is temporary data. For example, NVM interface 116

- 11 -

can select lower-performance memory locations for storing NV temporary data. The lower-performance memory locations may be less reliable pages or blocks, such as high cycled blocks (i.e. blocks that have been through more erase cycles, and may therefore have experienced more deterioration). NVM interface 116 may select higher-performance memory locations for storing non-temporary data, since non-temporary data is expected to be maintained in NVM 120 for a longer period of time.

[0028] In some embodiments, NVM interface 116 can select higher-speed memory locations or programming techniques for storing NV temporary data. This way, the delay between transferring data between memory 114 and NVM 120 is reduced, which may be beneficial for virtual memory schemes, when entering into a reduced-power hibernation mode, or in a variety of other situations. To improve speed, NVM interface 116 can select single-level cell ("SLC") blocks or can decrease the tuning resolution when programming the blocks with NV temporary data. NVM interface 116 can choose SLC blocks to increase programming speed while also improving reliability.

[0029] In some embodiments, NVM interface 116 can handle NV temporary data differently from other data by decorating NV temporary data with a "temporary marker" when storing the NV temporary data in NVM 120. This way, when data is read back out of NVM 120, NVM interface 116 can quickly and efficiently differentiate the NV temporary data with other information intended to be stored for a longer period of time.

[0030] FIG. 2 illustrates one way in which NVM interface 116 can decorate the NV temporary data with a

- 12 -

temporary marker. FIG. 2 is a graphical representation of the pages in a block of a non-volatile memory, such as NVM 120 of FIG. 1. As illustrated, each page of the block may be used to store a data field and an
5 associated metadata field. Each data field can store any suitable type of data, such as NV temporary data or any data that the file system requested to be stored (e.g., media assets). The metadata field of the same page can store metadata associated with the data in the
10 data field, such as any memory management data received or generated by NVM driver 117, including error correction code ("ECC") data, any information used for managing the mapping between logical and physical addresses (e.g., an LBA), and the like.

15 **[0031]** The metadata field can include space for storing a marker indicating whether the data in the data field is NV temporary data. For example, the "T" (or "temporary") markers in the metadata fields of pages 1 and 2 in FIG. 2 indicate that the associated
20 data fields include NV temporary data, while the "P" (or "persistent") markers in the metadata field of page 0 and the last page of the block indicate that the data in the associated data fields include data intended for persistent, longer-term storage. The "T"
25 temporary marker and the "P" persistent marker may be implemented using any suitable approach. That is, these markers may include any suitable number of bits (e.g., one or more bits) and the actual bit or word values of the "T" and "P" markers can each take on any
30 suitable predetermined bit pattern.

[0032] In some embodiments, NVM interface 116 may not use a specific "P" marker, and may instead use only "T" temporary markers to indicate where NV temporary

- 13 -

data is stored in NVM 120. In these embodiments, on readback of a page, NVM interface 116 can determine that the page's data field does not contain NV temporary data based on the lack of a temporary marker in the metadata field.

[0033] FIG. 2 provides an example where the same block can include both NV temporary data as well as persistent data. In other embodiments, such as the embodiment illustrated in FIG. 3, an entire block or superblock may be used to store either NV temporary data or persistent data.

[0034] FIG. 3 is a graphical representation of multiple blocks across several integrated circuits of a non-volatile memory, such as NVM 120 of FIG. 1. Each column represents a different integrated circuit and each row represents a different superblock. Thus, as illustrated in the first integrated circuit ("IC 0"), an entire block (i.e., block 2) may be used to store NV temporary data. In some embodiments, NVM interface 116 can provide a temporary marker in every used page of the block, which can provide extra insurance that NVM interface 116 can correctly interpret this block as storing NV temporary data. In other embodiments, NVM interface 116 may store temporary markers in a subset of the pages in the block. For example, NVM interface 116 can store a temporary marker in only the first page of the block so that NVM interface 116 can later determine whether the block contains NV temporary data when reading the block from the first page to the last page. Alternatively, NVM interface 116 can store a temporary marker in only the last page (so NVM interface 116 can later read the block from last page to first page), or NVM interface 116 can store

- 14 -

temporary markers in both the first and last pages of the block.

[0035] In some embodiments, NVM interface 116 may select an entire superblock for use in storing NV temporary data. This is illustrated in the fifth superblock ("SUPERBLOCK 4") in FIG. 3, where all of the blocks in the superblock can be used for storing NV temporary data. As with single-block case, NVM interface 116 may or may not store temporary markers in all of the used pages of the superblock. Instead, NVM interface 116 can store temporary markers in only the first page of each block in the superblock, only the last page of each block in the superblock, in both the first and last pages of each block, or in one or more pages of a subset of the blocks in the superblock.

[0036] While not depicted in the figure to preserve clarity, the blocks that do not include "T" temporary markers may (or may not) include "P" persistent markers in some or all of the pages. Moreover, while five integrated circuits and six superblocks (for a total of 30 blocks) are illustrated, it should be understood that NVM 120 can include any other suitable number of integrated circuits and superblocks.

[0037] NVM interface 116 can use the temporary markers decorating the NV temporary data to process the NV temporary data differently from other types of data. For example, the temporary markers can affect the bootup process of electronic device 100. Upon bootup of electronic device 100, NVM driver 117 may perform a series of initialization procedures so that NVM driver 117 can properly determine the current state of NVM 120. Included in the initialization procedures may be the reconstruction of one or more metadata tables or

- 15 -

maps, such as the table indicating which logical addresses are being used and the logical-to-physical address mapping.

[0038] To reconstruct the tables, NVM driver 117 may
5 read each page of NVM 120 and, for each page, may determine whether to include the page in the tables. If NVM driver 117 determines that the page is designated with a temporary marker (or is in a block or superblock currently designated with temporary
10 markers), NVM driver 117 can leave the page out of the tables. This is because the page has been used to store NV temporary data, which was only valid during the previous boot of electronic device 100. Thus, NVM driver 117 can mark the page as invalid so that, at
15 some appropriate time, the page can be freed up (e.g., during garbage collection or wear leveling) and used to store other information. Since NVM driver 117 invalidates the page and omits it from the tables right at bootup time, memory use of NVM 120 may be made
20 immediately efficient. That is, at no point during the current boot cycle will NVM driver 117 perform needless operations during garbage collection or wear leveling to maintain the now-obsolete NV temporary data.

[0039] Referring now to FIGS. 4 and 5, flowcharts of
25 illustrative processes are shown in accordance with various embodiments of the invention. The steps of these illustrative processes may be performed by any suitable component or combination of components in an electronic device, such as by control circuitry 112
30 (FIG. 1) operating under the control of NVM driver 117 (FIG. 1).

[0040] Looking first at FIG. 4, process 400 is shown for decorating NV temporary data with a temporary

- 16 -

marker when storing the NV temporary data in a NVM, such as NVM 120 (FIG. 1). Process 400 may begin at step 402, where the file system may decide to store data in the NVM. For NV temporary data, this can occur, for example, when the volatile memory is running out of space or when the volatile memory needs to be powered down in a reduced power state (i.e. hibernation state).

[0041] At step 404, data may be provided from the file system to the NVM interface (e.g., a NVM driver) for storage in the NVM. Then, at step 406, the NVM interface can determine whether this data is temporary data that should be stored as NV temporary data. In some embodiments, the NVM interface can interpret whether the data is temporary data based on the write request received from the file system. The file system may have issued a write request signaling that the data is temporary data in response to an indication from an application.

[0042] If, at step 406, the NVM interface determines that the data is NV temporary data, the NVM interface can prepare metadata for the data at step 408, where the metadata includes a temporary marker. Preparing the metadata at step 408 (or step 412, discussed below) can include determining and maintaining a mapping from a logical address (received with the write request) to a physical address associated with the NVM. The NVM interface can also select, at step 410, lower-performance and/or higher-speed memory locations (i.e. pages, blocks, or super blocks) or programming techniques for storing the data, since the data is expected to have a short lifespan in non-volatile memory. For example, the NVM interface can select less

- 17 -

reliable blocks (e.g., high cycled blocks) or SLC blocks in which to store the data.

[0043] Otherwise, if the NVM interface determines at step 406 that the data is not NV temporary data, the NVM interface can prepare metadata that does not include a temporary marker (and may instead include a persistent marker) at step 412. In this case, because the data may be stored long-term in non-volatile memory, the NVM interface can select higher-performance and/or lower-speed memory locations or programming techniques for storing the data at step 414. For example, the NVM interface can select more reliable blocks or MLC blocks in which to store the data.

[0044] From step 410 or step 414, process 400 can continue to step 416, where the NVM interface can store the data (whether it be NV temporary data or other data) and at least some of its associated metadata in the selected one or more pages of the NVM. Process 400 may then end at step 418.

[0045] Turning now to FIG. 5, process 500 is shown for processing NV temporary data at bootup of an electronic device. Process 500 may begin at step 502 at bootup of the electronic device. At step 504, an NVM interface can perform various bootup procedures so that the NVM interface can determine the initial state of the NVM and its memory locations. The initial bootup procedures can therefore include constructing metadata tables, such as a table indicating which logical addresses are allocated for use and/or a mapping of logical-to-physical addresses for each page of the NVM that contains valid data.

[0046] From step 504, the NVM interface can prepare the tables by scanning through and processing each page

- 18 -

of the NVM. In particular, at step 506, the NVM interface can read a first page from the NVM, and at step 508, the NVM interface can determine whether this first page includes a temporary marker in the page's metadata field. If not, the first page does not include NV temporary data, so at step 510, the NVM interface can add the page to the tables. For example, the NVM interface may determine (e.g., from the metadata field) the logical address associated with the page and may add the page's logical-to-physical address mapping to the appropriate table, or the NVM interface may indicate in a table that the logical address is currently allocated by the file system.

[0047] If, at step 508, the NVM interface determines instead that the page includes a temporary marker in the page's metadata field, process 500 can continue to step 512. Because the temporary marker indicates that the page stores NV temporary data, which is no longer valid now that the electronic device has been rebooted, process 500 can branch to step 512. At step 512, the NVM interface can ignore the metadata in the page for purposes of metadata table reconstruction. For example, while a logical address stored in the page would otherwise have indicated that the logical address is being used by the file system, the NVM interface can ignore the presence of the logical address in the page. Then, at step 514, the NVM interface can mark the page as containing invalid data. This may allow the NVM interface to, at the appropriate time, perform garbage collection on the page to free up space for storing other information.

[0048] From step 510 or step 514, process 500 can continue to step 516, where the NVM interface can

- 19 -

determine whether the metadata tables are done being constructed. This determination can involve determining whether there are additional pages that have not yet been read and processed. If there are
5 additional pages that need to be considered for inclusion in the metadata tables, process 500 may return to step 506 so that NVM interface can read another page of the NVM. Otherwise, if there are no additional pages, process 500 can end at step 518.

10 **[0049]** It should be understood that the steps of processes 400 and 500 of FIGS. 4 and 5, respectively, are merely illustrative. Any of the steps may be modified, removed, or combined, and additional steps may be included, without departing from the scope of
15 the invention.

[0050] The described embodiments of the invention are presented for the purpose of illustration and not limitation.

- 20 -

What is Claimed is:

1. An electronic device comprising:
a volatile memory for storing temporary data;
a non-volatile ("NV") memory comprising
5 a plurality of pages;
control circuitry operative to store the temporary data in the non-volatile memory as NV temporary data, wherein the control circuitry stores a temporary marker in at least one of the pages in which
10 the NV temporary data is stored.
2. The electronic device of claim 1,
wherein the electronic device is further operative to:
prepare metadata for the NV temporary data; and
5 store the metadata and the temporary marker in a metadata field of the at least one of the pages in which the NV temporary data is stored.
3. The electronic device of claim 1,
wherein the temporary marker comprises a predetermined bit pattern.
4. The electronic device of claim 1,
wherein the plurality of pages are organized into blocks, and wherein the control circuitry is further operative to:
5 select one of the blocks for storing the NV temporary data; and
store a temporary marker in at least one page of the selected block.

- 21 -

5. The electronic device of claim 4, wherein the control circuitry is further operative to store a temporary marker in all of the pages of the selected block.

6. The electronic device of claim 1, and wherein the control circuitry is further operative to:
identify that the volatile memory is in
a low memory state; and

5 store the temporary data in the non-volatile memory responsive to identifying the low memory state.

7. The electronic device of claim 1, wherein the control circuitry is further operative to select lower-performance pages for storing the temporary data.

8. The electronic device of claim 1, wherein the control circuitry is further operative to select a higher-speed programming technique for storing the temporary data.

9. A method of storing data in a non-volatile memory, wherein the non-volatile memory comprises a plurality of pages, the method comprising:
identifying the data as temporary data;
5 preparing metadata for the data, wherein the metadata comprises a temporary marker; and
storing the data and the metadata in at least one page of the non-volatile memory.

10. The method of claim 9, further comprising:

- 22 -

providing the data from a file system to
a non-volatile memory interface with a write request,
5 wherein the identifying is performed by
the non-volatile memory interface in response to the
write request.

11. The method of claim 9, wherein the
temporary marker comprises at least one bit of a
predetermined value.

12. The method of claim 9, wherein the
preparing further comprises determining a logical-to-
physical address mapping for the data.

13. The method of claim 9, further
comprising selecting the at least one page of the non-
volatile memory based on the identifying.

14. The method of claim 9, further
comprising:

identifying additional data for storage
in the non-volatile memory, wherein the additional data
5 is identified as not being temporary data;

preparing additional metadata for the
additional data, wherein the additional metadata does
not include a temporary marker; and

10 storing the additional data and the
additional metadata in at least one other page of the
non-volatile memory.

15. The method of claim 9, wherein the
identifying comprises identifying that the data
comprises at least one of hardware-based hibernation
data, application-generated data, application-based
5 hibernation data, and data for a virtual memory scheme.

- 23 -

16. An electronic device comprising:
a non-volatile memory;
control circuitry operative to create a
metadata table for the non-volatile memory at bootup of
5 the electronic device, wherein creating the metadata
table comprises:

reading a plurality of pages of the
non-volatile memory, wherein at least one of the pages
includes a temporary marker; and
10 omitting the at least one page from
the metadata table.

17. The electronic device of claim 16,
wherein the control circuitry is further operative to
create the metadata table by marking the at least one
page as invalid.

18. The electronic device of claim 16,
wherein the control circuitry is further operative to
create the metadata table by adding valid pages of the
non-volatile memory that do not include a temporary
5 marker to the metadata table.

19. The electronic device of claim 16,
wherein the metadata tables comprises a table
indicating which logical addresses are currently
allocated for use by a file system.

20. The electronic device of claim 16,
wherein the metadata table comprises a logical-to-
physical address mapping.

- 24 -

21. The electronic device of claim 16,
wherein the non-volatile memory comprises NAND flash
memory.

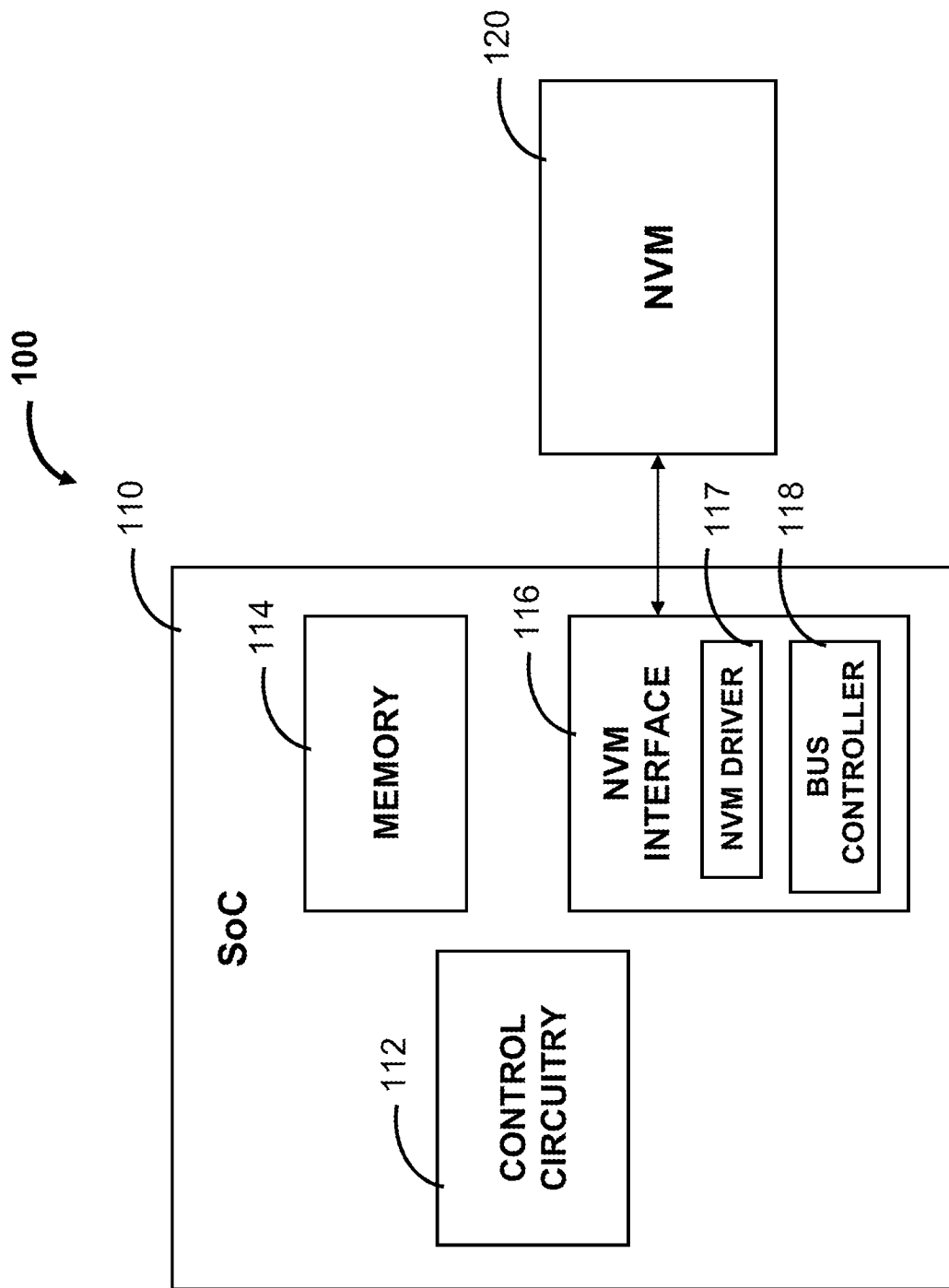


FIG. 1

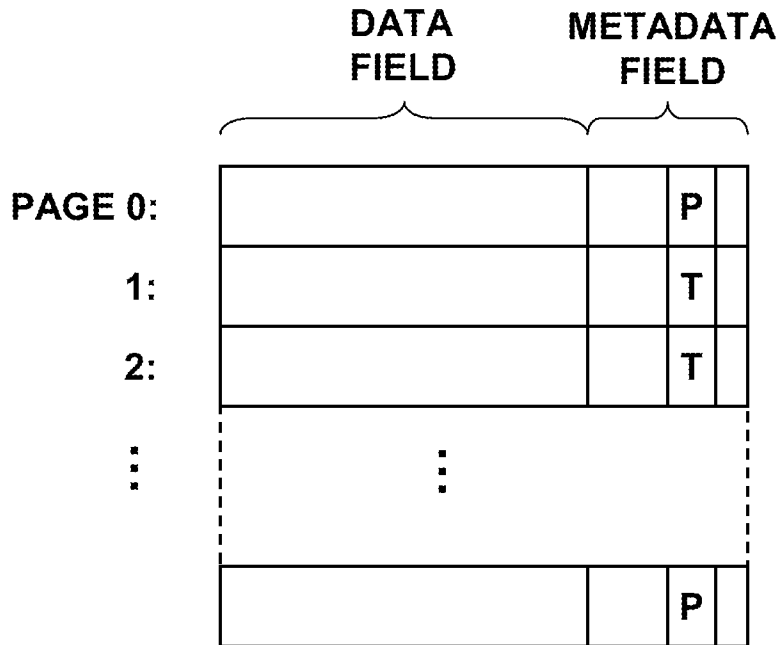


FIG. 2

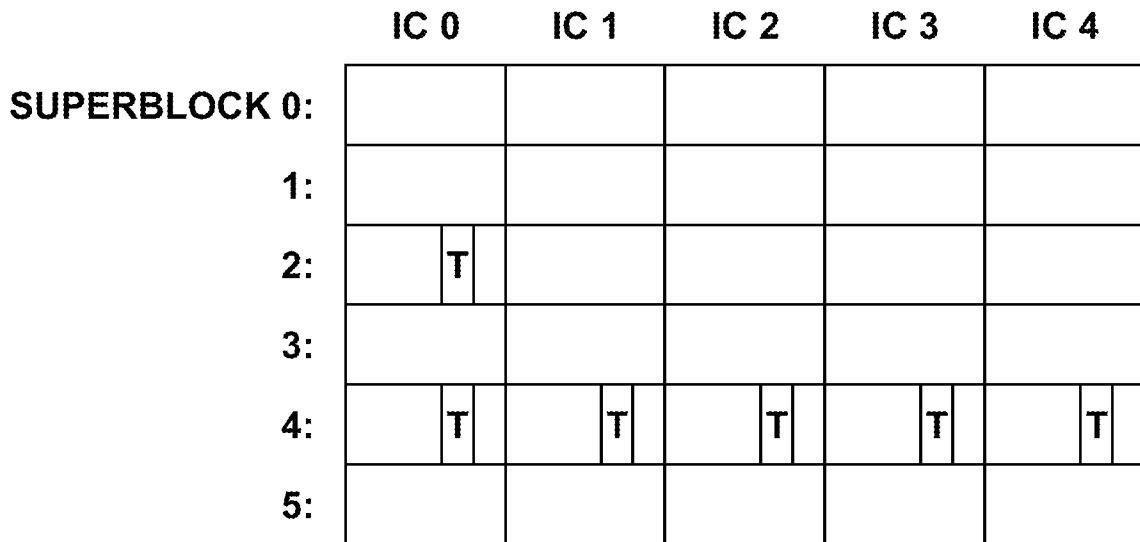


FIG. 3

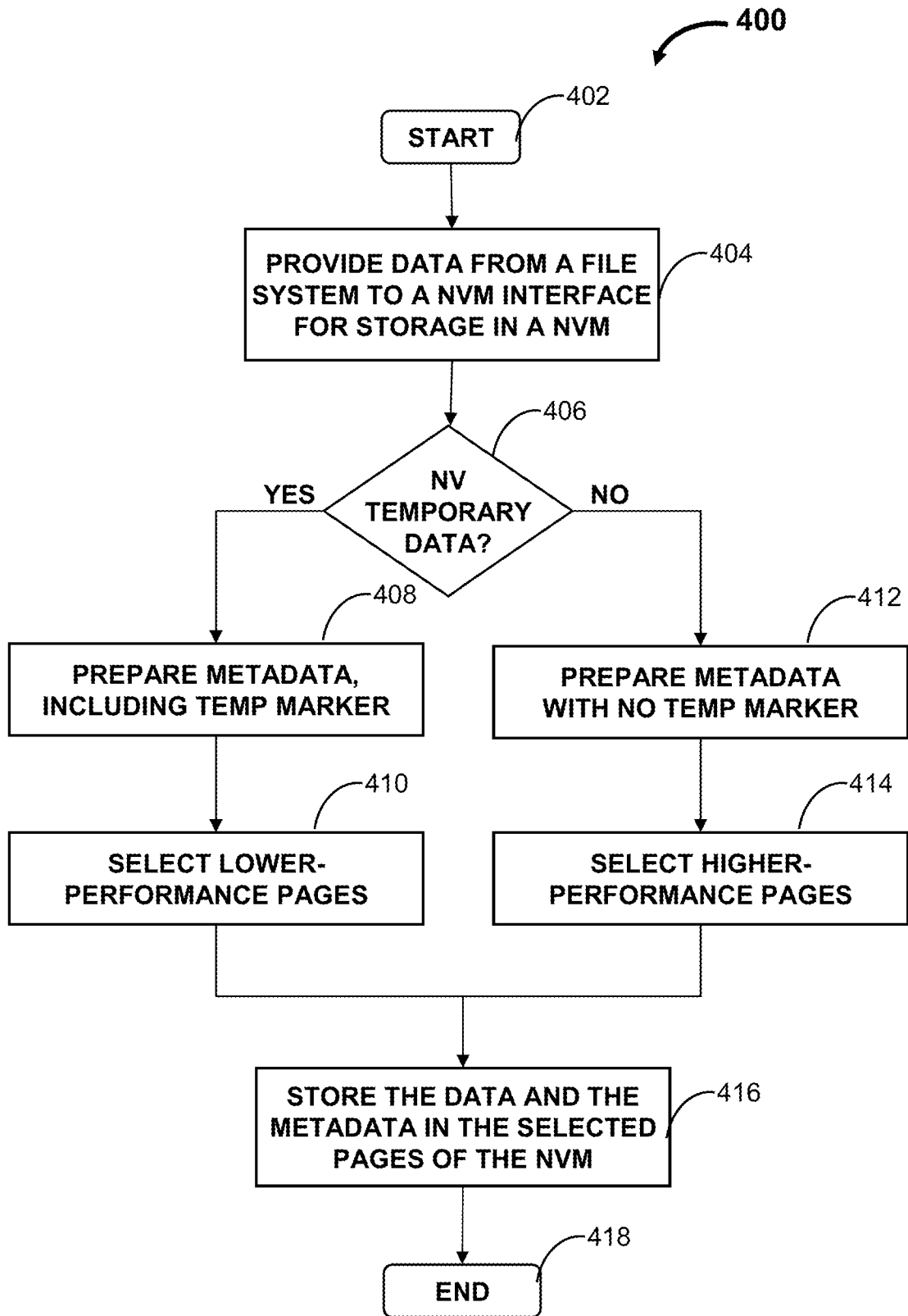


FIG. 4

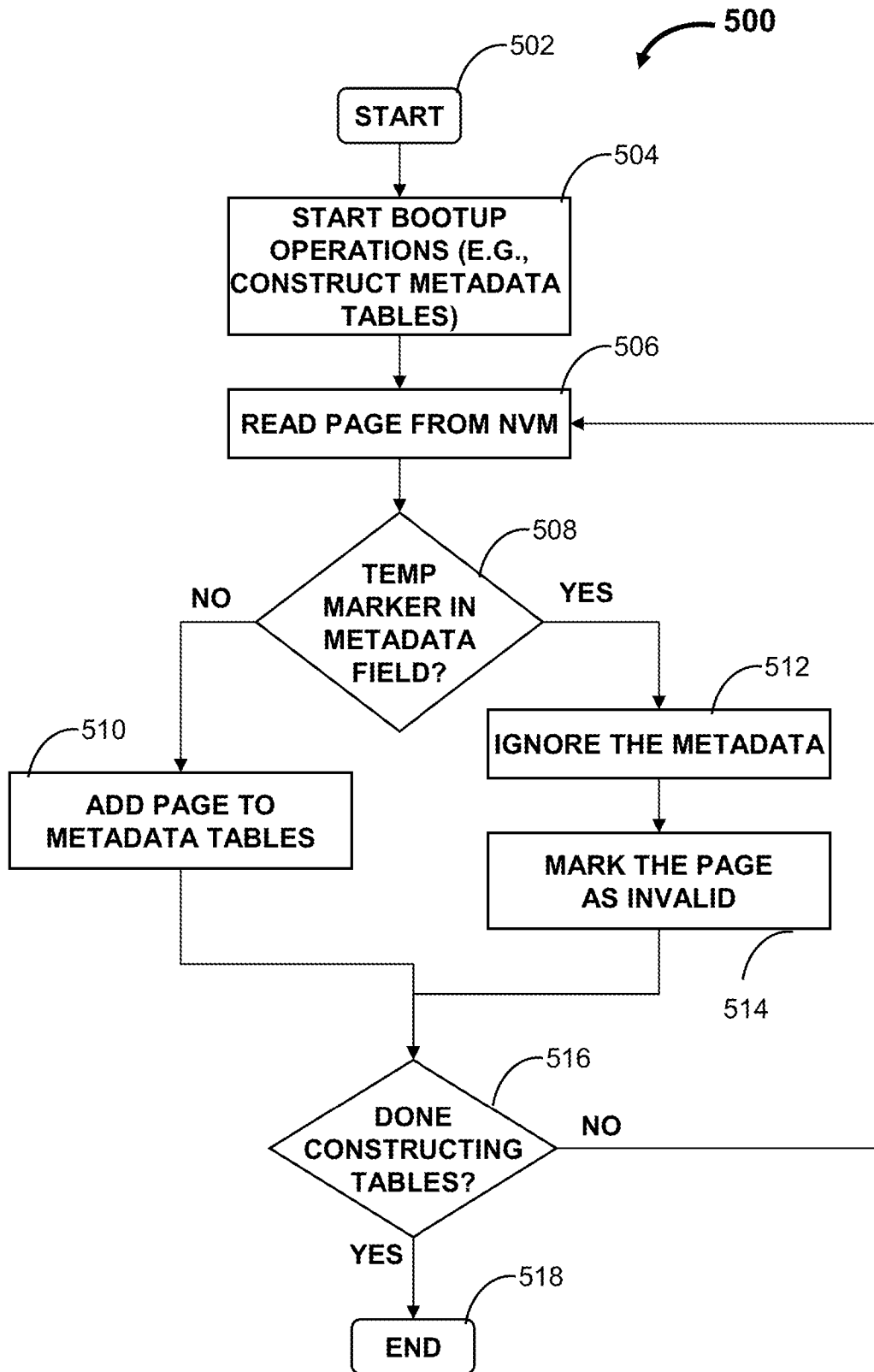


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 12/46236

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 13/00 (2012.01)

USPC - 711/111

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)

USPC: 711/111

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

USPC: 711/100, 101, 111; 714/5.111, 6.2, 6.12 (keyword limited - see search terms below)

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

PubWEST (PGPB, USPT, USOC, EPAB, JPAB); GOOGLE; Google Scholar

Terms: nonvolatile, memory, storage, cache, flash, temporary, data, address, transfer, allocate, restore, page, block, logical, physical, volatile, mark, pointer, tag, flag, metadata, map.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 2005/0117418 A1 (Jewell et al.) 02 June 2005 (02.06.2005), entire document, especially abstract, Fig. 1, para [0005], [0020], [0024], [0032], [0077], [0078], [0083], [0094], [0104], [0109].	1, 3, 7-8, 16 --- 2, 4-6, 9-15, 17-21
Y	US 2010/0325351 A1 (Bennett) 23 December 2010 (23.12.2010), entire document, especially abstract, para [0005], [0009], [0007], [0038], [0054], [0066], [0072], [0083], [0088], [0090].	2, 4-6, 9-15, 17-21
A	US 2011/0066808 A1 (Flynn et al.) 17 March 2011 (17.03.2011), entire document, especially abstract, para [0005], [0007], [0016], [0046].	1-21

 Further documents are listed in the continuation of Box C.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

11 September 2012 (11.09.2012)

Date of mailing of the international search report

25 SEP 2012

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