



(51) International Patent Classification:
G11C 11/406 (2006.01)

(21) International Application Number:
PCT/US2015/014794

(22) International Filing Date:
6 February 2015 (06.02.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventor: **BENEDICT, Melvin**; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(74) Agents: **PINKSTON, Caroline** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

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

HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

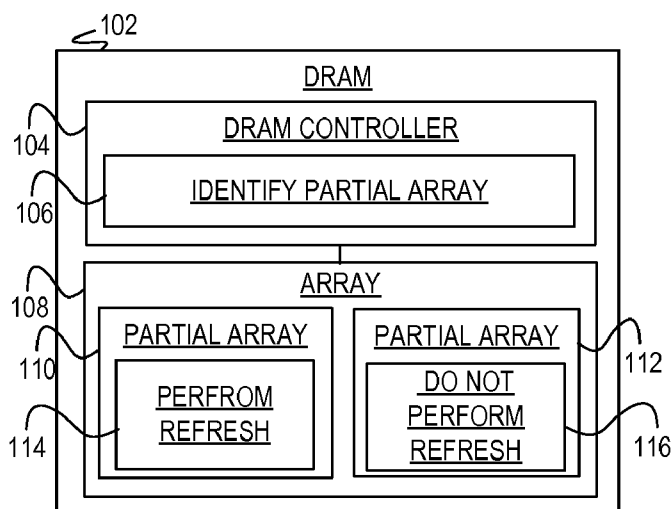
- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

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

(54) Title: REFRESHING AN IDENTIFIED PARTIAL ARRAY

FIG. 1



(57) Abstract: Examples herein disclose a dynamic random access memory (DRAM) including a controller, exclusive to the DRAM, to identify a partial array for refresh. The partial array performs the refresh. The examples maintain accessibility to the non-identified partial arrays during the refresh of the identified partial array. Refreshing the partial array rather than a full array in the DRAM, reduces the memory inaccessibility size. Additionally, refreshing a smaller region of memory reduces the period of time in which the DRAM may be inaccessible.

REFRESHING AN IDENTIFIED PARTIAL ARRAY

BACKGROUND

[0001] Memory refresh is a process of periodically reading information from an area of a computer memory and rewriting this read information back to the computer memory. Memory refresh preserves data in the computing memory.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] In the accompanying drawings, like numerals refer to like components or blocks. The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram of an example dynamic random access memory (DRAM) to identify a partial array for performing a refresh;

[0004] FIG. 2A is a block diagram of an example memory system architecture including a DRAM divided into multiple data bank groups;

[0005] FIG. 2B is a block diagram of an example dual in-line memory module (DIMM) consisting of multiple DRAM circuits to interface with a memory controller;

[0006] FIG. 3 is a flowchart of an example method to maintain activity to non-identified portions of an array while performing a refresh on an identified partial array;

[0007] FIG. 4 is a flowchart of an example method to identify a data bank group among multiple data bank groups in a DRAM for performing a refresh;

[0008] FIG. 5 is a block diagram of an example computing device with a processor to execute instructions in a machine-readable storage medium for maintaining activity to a non-identified portion of an array during a refresh of an identified partial array; and

[0009] FIG. 6 is a block diagram of an example computing device with a processor to execute instructions in a machine-readable storage medium for incrementing an address counter to identify a different partial array for a subsequent refresh.

DETAILED DESCRIPTION

[0010] A memory refresh may be a defining characteristic during operation of a dynamic random access memory (DRAM) circuit. In the DRAM circuit, each data bit value is stored in a presence or absence of an electrical charge on a capacitor in the circuit. As time passes, charges on these capacitors may leak away, which may cause a loss of the data bit value. To prevent the loss of the data bit value, a memory array internal to the DRAM is refreshed. During the refresh to the memory array, the data values within the cells of the array are read and rewritten to the data cells, thus restoring the charge on each capacitor. The process for refreshing the DRAM to a non-volatile device may take the DRAM out of operation during the refresh as the complete array may be refreshed at a given time. Taking the DRAM out of operation means the array may be inaccessible for a period of time. This may be inefficient and take much time to fully refresh the array at the given time.

[0011] To address these issues, examples disclose identifying a partial array in a DRAM and performing a refresh on the identified partial array. The examples maintain accessibility to the non-identified partial arrays during the refresh of the identified partial array. Refreshing the partial array rather than a full array in the DRAM, reduces the memory inaccessibility size. Additionally, refreshing a smaller region of memory reduces the period of time in which the DRAM may be inaccessible. For example, refreshing smaller regions of data within the DRAM means those smaller regions of data are taken out of operation during the refresh while other regions of data within the DRAM remain accessible or operational. Further, refreshing the partial array in the DRAM provides a finer granularity of control over which portions of memory in the DRAM to refresh.

[0012] In another example, a memory controller identifies the partial array and transmits the refresh request to a controller exclusive to the DRAM to perform the refresh. In this example, the memory controller transfers control to the DRAM controller for a period of time. The transfer of control to the DRAM controller, includes the refresh control, timing, address, etc. This transfer of control allows the DRAM controller to granularly control a location in which to refresh the partial array. This granular control includes performing the refresh per-bank group or per-bank, thus allowing a fine tuning control over those smaller regions of the array to refresh.

[0013] Referring now to the figures, FIG. 1 is a block diagram of an example dynamic random access memory (DRAM) 102 including a dynamic random access memory (DRAM) controller 104. The DRAM controller 104 identifies a partial array 110 at module 106. Upon identifying the partial

array at module 106, the identified partial array 110 performs a refresh at module 114. The DRAM 102 further includes a memory array 108. The memory array 108 includes the identified partial array 110 in which to perform the refresh at module 114 and a non-identified partial array 112 which does not perform refresh at module 116. FIG. 1 represents a memory system within a computing device used to retain digital data or digital content. Although FIG. 1 represents the memory system with a single DRAM 102, implementations should not be limited as the memory system may include multiple DRAM, a dual in-line memory module (DIMM) with the multiple DRAM, etc.

[0014] The DRAM 102 is a type of random access memory which stores data bits or content in multiple capacitors (not illustrated) in the array 108. These multiple capacitors include a charged state representing a bit value of “0” or “1,” accordingly. The DRAM 102 is a volatile memory as the capacitors may lose a charged state, thus losing data when power is removed. As such, the DRAM 102 may include, by way of example, memory component, dynamic memory, random access memory (RAM), or combination of such memory components that may use power to retain stored data.

[0015] The DRAM controller 104 identifies one of the partial arrays 110 and 112 within the array 108 to refresh. In implementations, the DRAM controller 104 performs a standard refresh or self-refresh to one of these partial arrays 110 and 112. For example in a standard refresh, the DRAM controller 104 communicates over a bus to a memory controller (not illustrated). The memory controller transmits a request to refresh one of the partial arrays 110 and 112 and as such includes the address pointing to the particular location of the partial array 110 or 112 for refreshing. In another example in a self-refresh, the DRAM controller 104 uses a timer and address counter to determine which partial array 110 or 112 may undergo refreshing at a given time. In this example, the DRAM controller 104 determines when and which partial array 110 or 112 to refresh. The DRAM controller 104 is a hardware component exclusive to the particular DRAM 102 such that the DRAM controller 104 manages the array 108 within the DRAM 102. As such, the DRAM controller 104 may include, by way of example, an integrated circuit, processing device, semiconductor, circuit, or other type of hardware component for managing the DRAM 102.

[0016] At module 106, the DRAM controller 104 identifies one of the partial arrays 110 and 112 in which to refresh. The partial array 110 is identified based on the address provided by the memory controller or from the DRAM controller 104 which may use an internal register to determine the identified partial array 110. Using the internal register, the DRAM controller 104 identifies the address to the location of the identified partial array 110 based upon the timer indicating to refresh

the particular partial array 110. The module 106 may include, by way of example, instructions (e.g., stored on a machine-readable medium) that, when executed (e.g., by the DRAM controller 104) implement the functionality of module 106. Alternatively, or in addition, the module 106 may include electronic circuitry (i.e., hardware) that implements the functionality of module 106.

[0017] The array 108 is an ordered arrangement of content in the DRAM 102. The ordered arrangement may be in rows, columns, and/or a matrix. As such, the array 108 includes multiple capacitors which include the charged state to represent the data value. These multiple capacitors are in an ordered arrangement of content to store the digital content. Although FIG. 1 illustrates the array 108 as including two partial arrays 110 and 112, this was done for illustration purposes as the array 108 may include multiple partial arrays.

[0018] The identified partial array 110 is considered a region of the array 108 for performing the refresh at module 114. In implementations the identified partial array 110 may include a data bank group or data bank. In another implementation, the identified partial array 110 size does not exceed the memory size of one of the data bank groups. This ensures smaller portions (e.g., the identified partial array 110) of the array 108 within the DRAM 102 are refreshed at a given time. In a further implementation, the identified partial array 110 becomes inaccessible for storing and retrieving digital content during performance of the refresh at module 114.

[0019] At module 114, the identified partial array 110 performs the refresh. During performance of the refresh at module 114, the DRAM controller 104 reads the data values from the multiple capacitors within the identified partial array 110 and writes these values back into the multiple capacitors. This ensures data values are not lost due to power leakage. During performance of the refresh within the identified partial array 110, the DRAM controller 104 and/or the memory controller quiets activity to that identified partial array 110. In this implementation, the identified partial array 110 becomes inaccessible until completion of the refresh. The module 114 may include, by way of example, instructions (e.g., stored on a machine-readable medium) that, when executed (e.g., by the DRAM controller 104) implement the functionality of module 114. Alternatively, or in addition, the module 114 may include electronic circuitry (i.e., hardware) that implements the functionality of module 114.

[0020] The non-identified partial array 112 is the portion of the array 108 which the DRAM controller 104 does not identify for performance of the refresh. As such, the DRAM controller 104 may continue activity for accessing the non-identified partial array 112.

[0021] At module 116, the non-identified partial array 112 does not perform the refresh. During this non-performance, the non-identified partial array 112 may be accessible to the DRAM controller 104 and/or memory controller. The module 116 may include, by way of example, instructions (e.g., stored on a machine-readable medium) that, when executed (e.g., by the DRAM controller 104) implement the functionality of module 116. Alternatively, or in addition, the module 116 may include electronic circuitry (i.e., hardware) that implements the functionality of module 116.

[0022] FIG. 2A is an example memory system architecture of a DRAM 202 divided into multiple data bank groups 210. Each of the multiple data bank groups 210 may be further divided into multiple data banks 212. As such, the DRAM controller 204 may perform a per-bank group 210 refresh or per-bank 212 refresh. In these implementations, a memory size of the identified partial array does not exceed one of the multiple data bank groups 210. The multiple data bank groups 210 each represent an equal size of memory which are considered semi-autonomous. These multiple data bank groups 210 are semi-autonomous in that each group 210 is separate from one another, but may share the data input/output selection 214 and thus the data I/O bus 216. Each of the multiple data banks 212 represent an equal size of memory within each of multiple data bank groups 210. The multiple data bank groups 210 and the multiple data banks 212 represent the division of an array within the DRAM 102 which may be identified for performing the refresh at these divided regions.

[0023] A command address bus 206 is a computer bus which is used to specify a physical address as identifying one of the multiple bank groups 210 communication that may occur between a memory controller and the DRAM 202 to identify which bank group 210 or data bank 212 in which to perform the refresh. A memory controller as illustrated in FIG. 2B communicates over the command address bus 206 to the DRAM 202. In this implementation, the memory controller specifies the memory address of one of the data bank groups 210 or at least one of the data banks 212. A command and address decode 208 is used to decode the address to identified data bank group 210 and/or data bank 212 to identify which portion of the memory should be refreshed. Upon decoding the address, this may be transmitted to the DRAM 202 for performing the refresh on the identified bank group 210 and/or identified data bank 212.

[0024] A data input/output selection 214 serves as a multiplexer to receive digital signals from the DRAM 202 and forwarding out on a data I/O bus 216. The data I/O bus 216 may then carry the digital signals to the appropriate location.

[0025] FIG. 2B is a block diagram of an example dual in-line memory module (DIMM) 218 consisting of multiple DRAM circuits 202. The DIMM 218 interfaces to a memory controller 220 over a double data rate interface 222. Each DRAM circuit 202 includes a DRAM controller 204 exclusive to that DRAM circuit and multiple data bank groups 210. Although FIG. 2B illustrates each DRAM circuit 202 as including two data bank groups 210 implementations should not be limited as the DRAM circuit 202 may include at least four data bank groups 210 which may further include multiple data banks 212 as in FIG. 2A.

[0026] The DIMM 218 represents the module which comprise a series of the DRAM circuits 202. The DIMM 218 may include a printed circuit for use in a computing device. Additionally, the DIMM 218 may communicate between each DRAM circuit 202 to the memory controller 220. As such, the DIMM 218 may further include electrical contacts on each side of the DRAM circuits 202 to provide these communications.

[0027] The memory controller 220 is a hardware component which manages multiple memory components, such as the DIMM 210 and multiple DRAM circuits 202. The memory controller 220 interfaces to the DIMM 218 to transmit a refresh request to one of the DRAM circuits 202. The memory controller 220 may include, by way of example, an integrated circuit, processing device, semiconductor, circuit, or other type of hardware component for managing multiple memory components, such as the DRAM circuits 202.

[0028] The DDR 222 is a type of interface between the memory controller 220 and the DIMM 218. In one implementation, the DDR 222 includes a double data rate fourth generation (DDR4) or subsequent type of interface generation between the DIMM 218 and the memory controller 220.

[0029] FIG. 3 is a flowchart of an example method, executable by a computing device, to maintain activity to non-identified portions of an array while performing a refresh on an identified partial array. The computing device identifies the partial array among multiple portions of the array for performing the refresh. Upon identifying the partial array within the DRAM, the computing device proceeds to perform the refresh on that identified partial array. During the performance of the refresh on the identified partial array, the computing device maintains activity and/or communications from a bus to the other portions of the array which were not identified for the refresh. Maintaining the activity to those non-identified portions of the array, reduces an amount of time the DRAM is out of communication with a memory controller. Refreshing the partial portion of the array within the DRAM, provides a finely controlled method for refreshing smaller portions of memory. In discussing

FIG. 3, references may be made to the components in FIGS. 1-2B to provide contextual examples. In one implementation of FIG. 3, the DRAM controller 104 as in FIG. 1 operates on the computing device to perform operations 302-306. In this implementation, the operations 302-306 may operate in the background of the computing device to perform the memory refresh on the partial array. In another implementation, the memory controller 220 as in FIG. 2B operates in conjunction with the DRAM controller 104 as in FIG. 1 to perform operations 302-306. Further, although FIG. 3 is described as implemented by the computing device, it may be executed on other suitable components. For example, FIG. 3 may be implemented by processor 602 to execute instructions on a machine-readable storage medium 604 as in FIG. 6.

[0030] At operation 302, the computing device identifies the partial array in which to perform the refresh. The memory controller transmits a request to the DRAM controller to refresh a particular partial array. The request includes an address of the particular partial array which should be refreshed. The memory controller includes a timer which may track which partial array should be refreshed at various points in time. This allows the memory controller to identify which partial array among multiple partial arrays should be refreshed at a given time. In this implementation, the memory controller tracks which partial array is in a refresh mode, thus the memory controller may cease operations to that partial array until completion of the refresh. In another implementation, the DRAM controller includes a register such as an address counter which determines which partial array should be refreshed. This allows the DRAM controller to determine which partial array will not be accessible during a period of time. In a further implementation, the DRAM includes multiple data bank groups and within each bank group includes multiple data banks. The partial array may include one of the multiple data bank groups or one of the multiple data banks. Thus, the computing device may perform a per-bank data refresh or a per-bank group data refresh. Yet, in another implementation, the partial array is no greater in memory size than one of the data bank groups. This ensures smaller portions of the array within the DRAM are refreshed at a given time.

[0031] At operation 304, the computing device performs the memory refresh on the partial array which was identified at operation 302. The memory refresh operation includes reading the bits per each identified partial array from sense amps and writing these read bit values back into the identified partial array. During the refresh operation, the memory controller and/or the DRAM controller quiets activity to the identified partial array. Quieting the activity to the identified partial array, means that identified partial array will not be accessible until completion of the refresh operation. The refresh

operation may include a self-refresh or a standard refresh. During the standardized refresh, the memory controller decides when and a potential location of which partial array to refresh. During the self-refresh operation, the DRAM controller decides when and a potential location of which partial array component to refresh.

[0032] At operation 306, the computing device maintains activity from a bus to the non-identified portions of the array. Operation 306 may occur during the performance of the refresh at operation 304. The non-identified portions of the array or other partial arrays within the DRAM remain accessible to store content and/or access the stored content. Additionally, the identified partial array is inaccessible during the performance of the refresh as at operation 304. In another implementation upon completion of the refresh to the identified partial array, the computing device increments an address counter which identifies a different partial array for another refresh. In this implementation, the originally identified partial array becomes accessible again, while the identified different partial array becomes inaccessible.

[0033] FIG. 4 is a flowchart of an example method, executable by a computing device, to identify a data bank group among multiple data bank groups in a DRAM. Upon the identification of the data bank group, the computing device performs a refresh on the identified data bank group. The computing device receives a command from a memory controller to perform a refresh on a partial array within the DRAM. In turn, the computing device identifies which partial array should undergo the refresh. The computing device proceeds to identify a particular data bank group among multiple data bank groups as the partial array for performing the refresh. As such, the computing device identifies the address which corresponds to that partial array for performing the refresh. The computing device may proceed to perform the refresh on the identified partial array. Upon the performance of the refresh, the computing device increments an address counter to identify a different region within the identified partial array for the refresh (e.g., a different data bank) and maintains activity to those portions of the array in the DRAM which were not identified (e.g., non-identified portions of the array). The computing device completes the refresh of each of the rows within the identified partial array and upon the complete refresh of each of the rows increments the address counter to identify a different partial array (e.g., a different data bank group) for a subsequent refresh. In discussing FIG. 4, references may be made to the components in FIGS. 1-2B to provide contextual examples. In one implementation of FIG. 4, the DRAM controller 104 as in FIG. 1 operates on the computing device to perform operations 402-418. In this implementation, the

operations 402-418 may operate in the background of the computing device to perform the memory refresh on the partial array. In another implementation, the memory controller 220 as in FIG. 2B operates in conjunction with the DRAM controller 104 as in FIG. 1 to perform operations 402-418. Further, although FIG. 4 is described as implemented by the computing device, it may be executed on other suitable components. For example, FIG. 4 may be implemented by processor 602 to execute instructions on a machine-readable storage medium 604 as in FIG. 6.

[0034] At operation 402, the computing device receives the command from the memory controller to perform the refresh. This command may include an address of the partial array in which to perform the refresh. The memory controller includes a timer which may time at which point particular partial arrays should be refreshed. In another implementation, the memory controller tracks which arrays should be refreshed in a sequential manner. In this implementation, the computing device performs a standardized refresh of the identified partial array as at operation 410.

[0035] At operation 404, the computing device identifies the partial array in which to perform the refresh. In one implementation, the partial array includes one of the data bank groups or one of the data banks within the data bank groups. Upon receiving the refresh command, the computing device may retrieve the address corresponding to the identified partial array or may identify which partial array to perform the refresh. Operation 404 may be similar in functionality to operation 302 as in FIG. 3.

[0036] At operation 406, the computing device identifies the specific of particular data bank group to perform the refresh. The specific or particular data bank group is identified among multiple data bank groups. In this implementation, a complete array within the DRAM may include at least two data bank groups. These data bank groups are broken up into equal semi-autonomous regions. For example, assume there is a 4 gigabyte DRAM memory, then if there are four data bank groups, each data bank group may represents 1 gigabyte of memory. Each of the data bank groups is associated with a different address. Thus, upon receiving the command by the memory controller at operation 402 or by using a tracking mechanism internal to the DRAM, the computing device the particular data bank group for performing the refresh at operation 408.

[0037] At operation 408, the computing device identifies the address corresponding to the partial array identified at operation 404. The computing device includes a timer which may track at which given time which partial array should be refreshed. In this implementation, the computing device may include an internal register indicating which partial array should be refreshed.

[0038] At operation 410, the computing device performs the refresh on the identified partial array. In an implementation, the refresh may include the normalized refresh in which the memory controller identifies which partial array should be refreshed. In another implementation, the refresh may include a self-refresh in which the DRAM controller identifies which of the partial arrays should be refreshed. Operation 410 may be similar in functionality to operation 304 as in FIG. 3.

[0039] At operation 412, the computing device increments the address counter to identify the different region within the identified partial array for refreshing. At operation 412, the identified partial array may include one of the data banks within the data bank group. Thus, the computing device identifies one of the other data banks in which to perform the refreshing.

[0040] At operation 414, the computing device maintains activity to the non-identified portions of the array. Maintaining the activity to these non-identified portions of the array includes those portions of the array remaining accessible to the memory controller to store content and/or retrieve content. Operation 414 may be similar in functionality to operation 306 as in FIG. 3.

[0041] At operation 416, the computing device checks to verify if each row or if each data bank within the partial array has been refreshed. Determining each row of data has been refreshed, the computing device increments the address counter to identify another partial array to perform the subsequent refresh.

[0042] At operation 418, the computing device increments the address counter. Incrementing the address counter, the computing device identifies the different partial array for a subsequent refresh. The computing device identifies the different partial array once determining each of the rows within the identified partial array has been refreshed. As such, the address may include the location of the different partial array for the computing device to perform the subsequent refresh. In this implementation, the partial array identified at operation 404 may become accessible while the different partial array may become inaccessible during performance of the subsequent refresh.

[0043] FIG. 5 is a block diagram of computing device 500 with a processor 502 to execute instructions 506-510 within a machine-readable storage medium 504. Specifically, the computing device 500 with the processor 502 is to identify a partial array within a DRAM for refreshing. Although the computing device 500 includes processor 502 and machine-readable storage medium 504, it may also include other components that would be suitable to one skilled in the art. For example, the computing device 500 may include the controller 104 as in FIG. 1. The computing device 500 is an electronic device with the processor 502 capable of executing instructions 506-510,

and as such embodiments of the computing device 500 include a mobile device, client device, personal computer, desktop computer, laptop, tablet, video game console, or other type of electronic device capable of executing instructions 506-510. The instructions 506-510 may be implemented as methods, functions, operations, and other processes implemented as machine-readable instructions stored on the storage medium 504, which may be non-transitory, such as hardware storage devices (e.g., random access memory (RAM), read only memory (ROM), erasable programmable ROM, electrically erasable ROM, hard drives, and flash memory).

[0044] The processor 502 may fetch, decode, and execute instructions 506-510 for refreshing the identified partial array within the DRAM. Specifically, the processor 502 executes instructions 506-510 to: identify the partial array for refresh in the DRAM, the partial array is identified as a portion of the array within the DRAM; perform the refresh on the identified partial array; and maintain activity to the non-identified portion of the array while activity to the identified partial array is halted.

[0045] The machine-readable storage medium 504 includes instructions 506-510 for the processor 502 to fetch, decode, and execute. In another embodiment, the machine-readable storage medium 504 may be an electronic, magnetic, optical, memory, storage, flash-drive, or other physical device that contains or stores executable instructions. Thus, the machine-readable storage medium 504 may include, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a memory cache, network storage, a Compact Disc Read Only Memory (CDROM) and the like. As such, the machine-readable storage medium 504 may include an application and/or firmware which can be utilized independently and/or in conjunction with the processor 502 to fetch, decode, and/or execute instructions of the machine-readable storage medium 504. The application and/or firmware may be stored on the machine-readable storage medium 504 and/or stored on another location of the computing device 500.

[0046] FIG. 6 is a block diagram of computing device 600 with a processor 602 to execute instructions 606-614 within a machine-readable storage medium 604. Specifically, the computing device 600 with the processor 602 is to identify a particular bank group among multiple bank groups within a DRAM for refreshing. Upon performing the refresh on the identified particular bank group, an address counter is incremented for identifying a different partial array for a subsequent refresh. The computing device 600, the processor 602, and the machine-readable

medium 604 may be similar in structure and functionality to the computing device 500, the processor 502, and the machine-readable medium 504 as in FIG. 5.

[0047] The processor 602 may fetch, decode, and execute instructions 606-614 for incrementing the address counter to identify the different partial array for performing the refresh. In one implementation upon execution of instruction 606, the processor 602 may execute instruction 608 for identifying a data bank group to perform the refresh. In another implementation upon execution of instructions 606-612, the processor 602 may execute proceed to execute instruction 614 for identifying the different partial array based on the incremented address counter. Specifically, the processor 602 executes instructions 606-614 to: identify the partial array to perform the refresh in the DRAM; identify the particular data bank group to perform the refresh among multiple data bank groups; perform the refresh on the identified partial array within the DRAM; during refresh of the identified partial array within the DRAM, the non-identified portions of the array maintain activity and/or communication from a bus; and upon completion of the refresh, increment the address counter for identification of a different partial array to perform a subsequent refresh.

[0048] As such, examples disclose performing an efficient refresh on a partial array within DRAM. Performing the refresh on the partial array, reduces a period of performing the refresh smaller chunks of data are refreshed at different times.

CLAIMSI claim:

1. A system to perform a refresh on a partial array in a dynamic random access memory, the system comprising:
 - a dynamic random access memory (DRAM) including:
 - a controller, exclusive to the DRAM, to identify a partial array within the DRAM for a refresh; and
 - the partial array to perform the refresh.
2. The system of claim 1 wherein the DRAM comprises multiple data bank groups, each data bank group includes multiple data banks and further wherein the identified partial array is no greater in size than one of the multiple data bank groups.
3. The system of claim 1 comprising:
 - a memory controller comprising:
 - an address register to track the identified partial array for the refresh within the array, wherein the identified partial array becomes inaccessible during the refresh.
4. The system of claim 1 comprising:
 - a double data rate interface to provide activity from the memory controller to the DRAM.
5. The system of claim 1 wherein the DRAM comprises:
 - a different partial array to maintain operation during the refresh of the partial array, wherein requests to the partial array are reduced during the refresh.
6. The system of claim 1 comprising:
 - a dual in-line memory module (DIMM) including multiple DRAMs.
7. A non-transitory machine-readable storage medium comprising instructions that when executed by a processor cause the processor to:
 - identify a partial array in a dynamic random-access memory (DRAM) to perform a refresh;

perform a refresh on the identified partial array in the DRAM; and
maintain accessibility to a non-identified partial array in the DRAM.

8. The non-transitory machine-readable storage medium including the instructions of claim 7 and further comprising instructions that when executed by the processor cause the processor to:
restrict accessibility to the identified partial array during the refresh.

9. The non-transitory machine-readable storage medium including the instructions of claim 7 wherein to identify the partial array in DRAM comprises instructions that when executed by the processor causes the processor to:

identify a data bank group among the multiple bank groups, wherein the identified partial array is no greater in size than the identified data bank group.

10. The non-transitory machine-readable storage medium including the instructions of claim 7 comprising instructions that when executed by the processor cause the processor to:

upon completion of the refresh to the identified partial, increment an address counter, internal to the DRAM, for identifying a different partial array for performance of a subsequent refresh.

11. A method, executable by a computing device, the method comprising:

identifying a partial array among multiple partial arrays in a dynamic random-access memory (DRAM) to perform a refresh;

refreshing the identified partial array in the DRAM; and

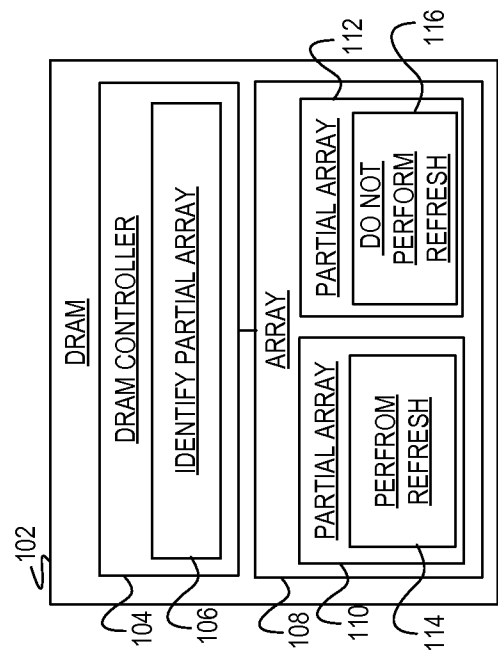
maintaining activity to a non-identified partial arrays in the DRAM during refresh of the identified partial array.

12. The method of claim 11 wherein identifying the partial array among the multiple partial arrays comprise:

identifying a data bank group among multiple data bank groups, wherein the identified partial array is no greater in size than the identified data bank group.

13. The method of claim 11 comprising:
incrementing an address counter, internal to the DRAM, to identify a different partial array for a subsequent refresh.
14. The method of claim 11 wherein identifying the partial array among the multiple partial arrays comprises:
identifying an address corresponding to the identified partial array.
15. The method of claim 11 comprising:
receiving a command from a memory controller to refresh the identified partial array wherein the command includes an address to the identified partial array.

FIG. 1



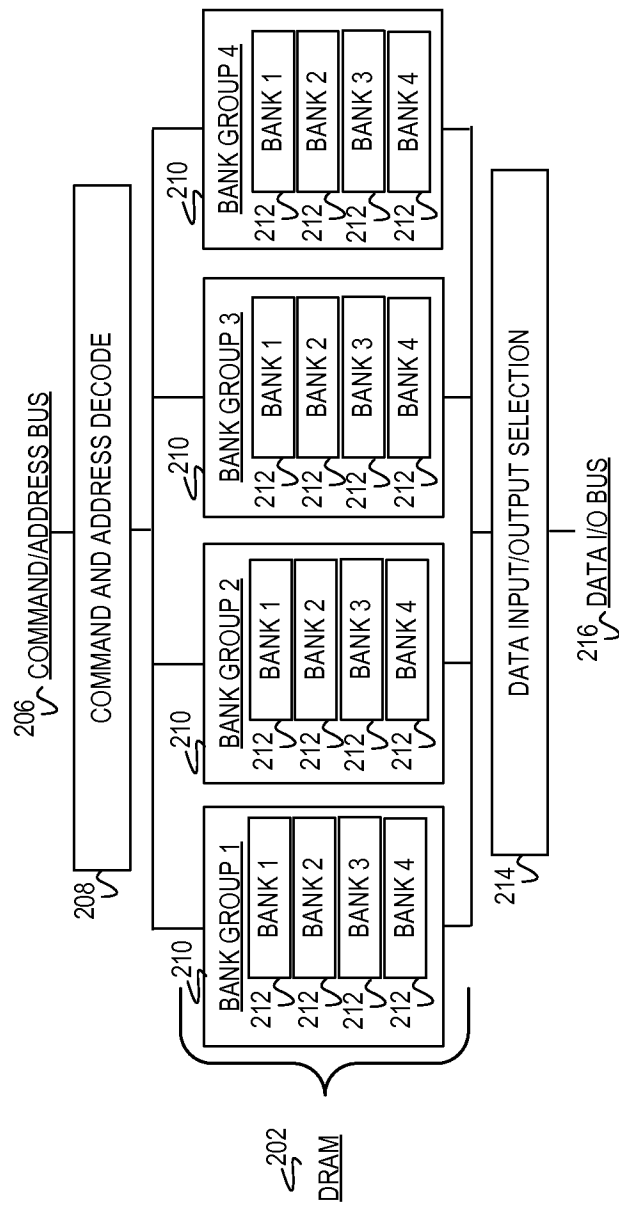


FIG. 2A

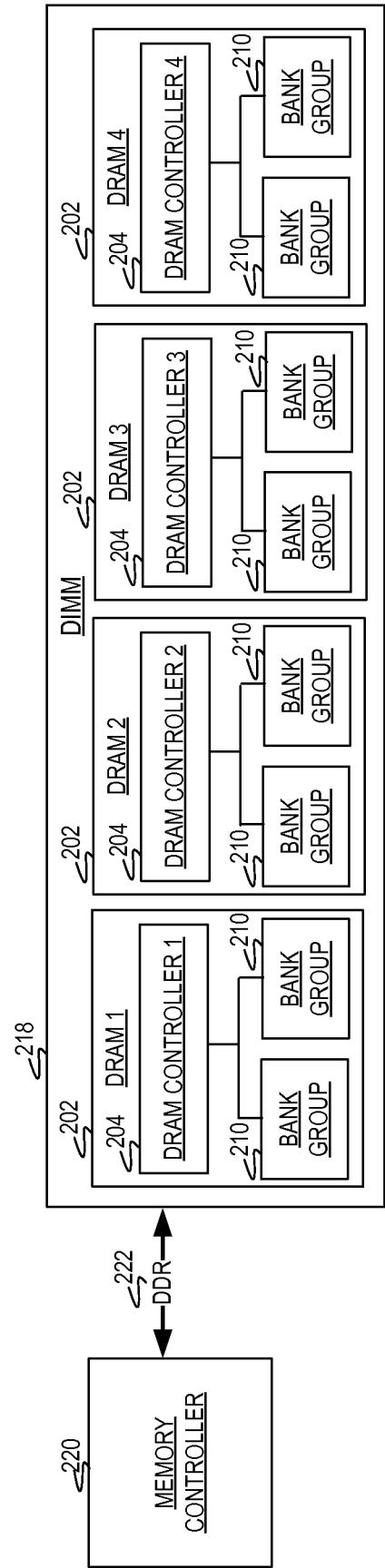
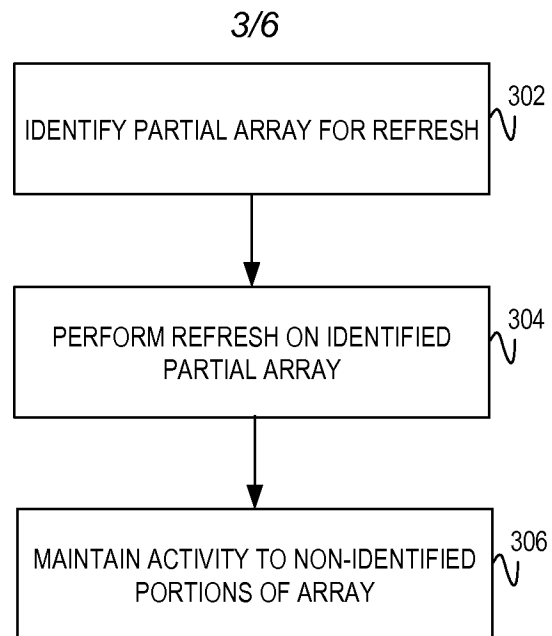
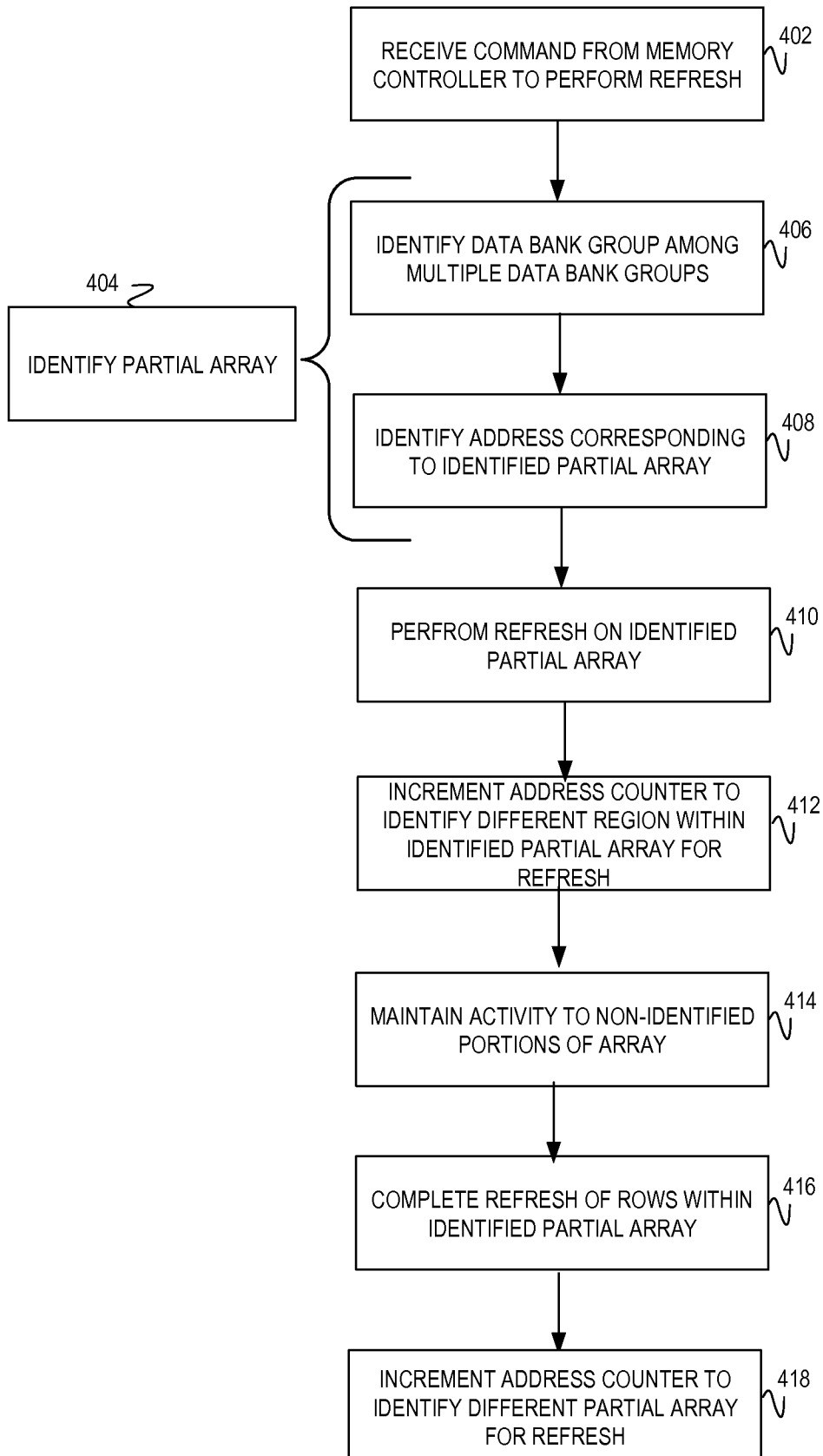


FIG. 2B

FIG. 3

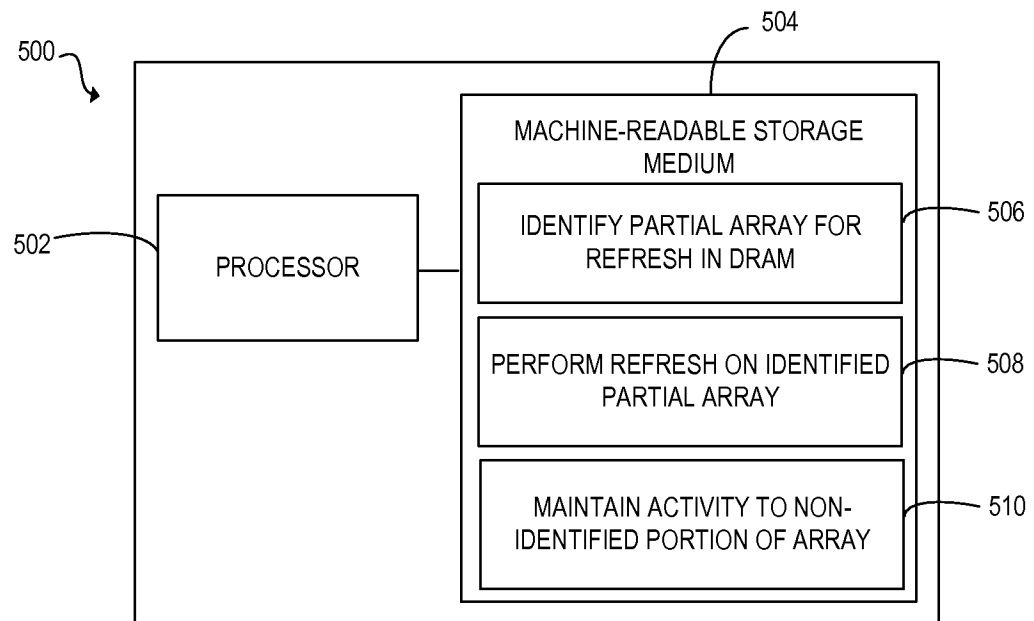


4/6



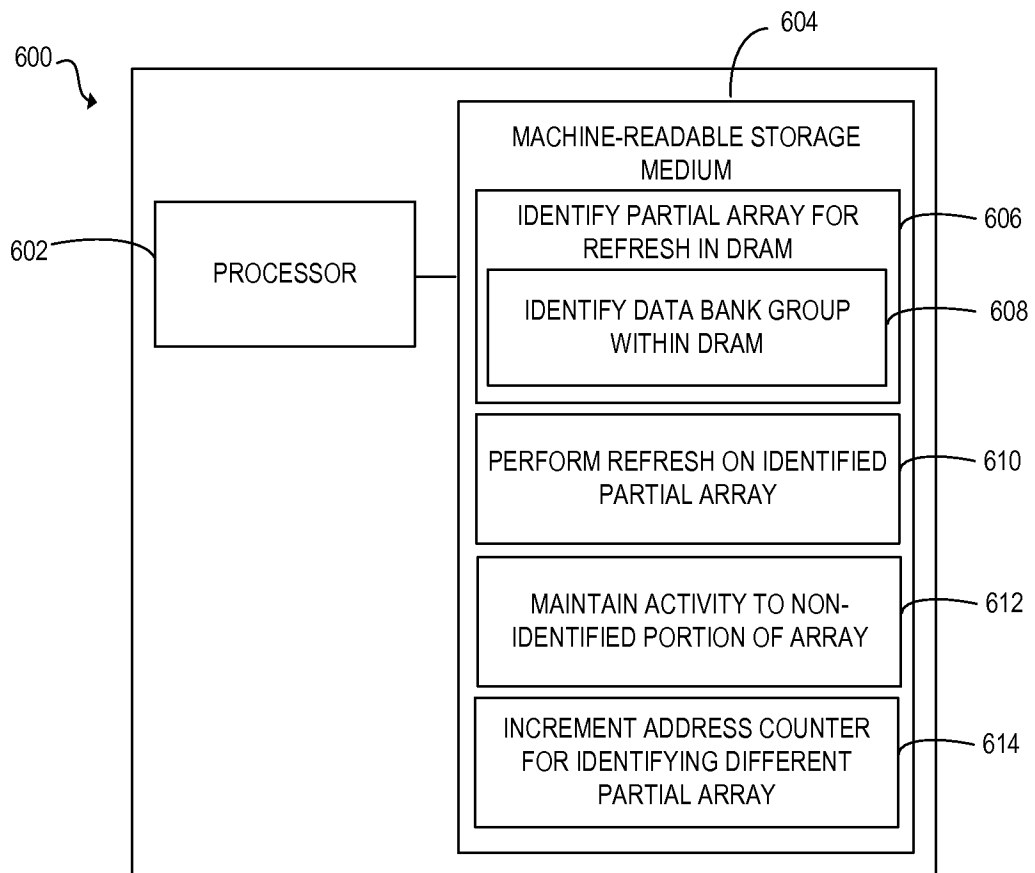
5/6

FIG. 5



6/6

FIG. 6



A. CLASSIFICATION OF SUBJECT MATTER**G11C 11/406(2006.01)i**

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)

G11C 11/406; G11C 11/34; G11C 7/00

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: refresh, dynamic random access memory, partial array, bank group, address counter

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014-0078846 A1 (SAMSUNG ELECTRONICS CO., LTD.) 20 March 2014 See paragraphs [0007]-[0008], [0027], [0066]-[0068], [0106]; claim 1; and figure 1.	1-9, 11-12, 14-15
Y		10, 13
Y	WO 02-078006 A1 (QUALCOMM INCORPORATED) 03 October 2002 See page 4, lines 6-8; and figure 1.	10, 13
A	US 7944764 B1 (SHEKOUFEH QAWAMI et al.) 17 May 2011 See column 4, line 60 - column 5, line 4; and figure 2.	1-15
A	US 2008-0094931 A1 (SUK-SOO PYO et al.) 24 April 2008 See paragraphs [0028]-[0029]; and figure 1.	1-15
A	US 2005-0105362 A1 (JONG-HYUN CHOI et al.) 19 May 2005 See paragraphs [0019]-[0020]; and figure 3.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

05 November 2015 (05.11.2015)

Date of mailing of the international search report

06 November 2015 (06.11.2015)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/014794

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0078846 A1	20/03/2014	KR 10-2014-0036620 A US 8976615 B2	26/03/2014 10/03/2015
WO 02-078006 A1	03/10/2002	US 2002-0138690 A1	26/09/2002
US 7944764 B1	17/05/2011	None	
US 2008-0094931 A1	24/04/2008	KR 10-0806341 B1 US 7755966 B2	03/03/2008 13/07/2010
US 2005-0105362 A1	19/05/2005	DE 102004053497 A1 KR 10-0621619 B1 KR 10-2005-0046427 A US 7180808 B2	23/06/2005 13/09/2006 18/05/2005 20/02/2007