Determining a chip select from physical rank information

Figure 3
DETERMINING A CHIP SELECT FROM PHYSICAL RANK INFORMATION

Background

[0001] Server size and memory capacity continue to grow at a phenomenal rate; however, correctable and uncorrectable memory errors also continue to increase. These errors can cause applications and operating systems to crash and can cause data to be lost.

Brief Description of the Drawings

[0002] Figure 1 illustrates a diagram of an example system according to the present disclosure.
[0003] Figure 2 illustrates a diagram of an example computing device according to the present disclosure.
[0004] Figure 3 illustrates a flow diagram of an example of a process for determining a chip select from physical rank information according to the present disclosure.
[0005] Figure 4 illustrates a flow diagram of an example of a method according to the present disclosure.
[0006] Figure 5 illustrates a diagram of an example system including a processor and non-transitory computer readable medium according to the present disclosure.
Detailed Description

[0007] As overall server memory capacity grows and as the number of bits per memory chip increase, the number and frequency of memory errors can increase significantly. Correctable and uncorrectable memory errors can cause applications and/or operating systems to crash, and can therefore be costly in terms of downtime and/or repairs. As used herein, correctable errors are memory errors that can be corrected or repaired, while uncorrectable errors are memory errors that cannot be corrected or repaired. Further, memory errors refer to the incorrect recall and/or complete loss of information in a memory system.

[0008] Solutions for data reliability and protection can include a pre-failure alert notification system that can monitor and predict potential problems with critical components such as system memory modules, and Resiliency, Availability, and Serviceability (RAS) systems. With a pre-failure alert notification system, a notification can be sent to a system administrator when a memory module exceeds a predefined threshold value for correctable memory errors so that the system administrator can schedule server maintenance to replace a memory module that may fail. A pre-failure notification system can help avoid unexpected interruption of applications and/or operating systems, but can also involve unnecessary memory module replacement and/or downtime while maintenance is carried out.

[0009] In RAS systems, hardware can be utilized to help determine the logical rank identification (ID) of a Dual In-Line Memory Module (DIMM) (e.g., a series of dynamic random-access memory (DRAM) integrated circuits) requiring service. DRAM is a type of random-access memory that stores each bit of data in a separate capacitor within an integrated circuit. Determining the logical rank ID of a DIMM can be accomplished by scanning hardware error counters to determine which counter(s) exceed a predetermined error count threshold value. A scan index (e.g., an index containing information corresponding to a scan of data on various pages) can contain the information required by the RAS code to compute the logical rank ID of the DIMM exhibiting correctable errors in excess of the predetermined threshold value. Such RAS code systems can
help avoid unexpected interruption of applications and/or operating systems; however, they can include scanning hardware rank counters to determine the logical rank ID of a DIMM exhibiting correctable errors in excess of the predetermined threshold value. This process can be time consuming and can include using different techniques for different operating systems.

[0010] In contrast, examples of the present disclosure can allow for location and logging of memory errors without directly accessing the hardware in the system. For example, a physical chip select to logical chip select can be mapped to emulate the DIMM rank scanning process independent of hardware. As used herein, emulation is the reproduction of a function or action of a different computer and/or software system, etc. In some examples, knowledge of how physical chip select lines can be mapped to the DIMM chip select lines can be used to locate and log memory errors. For example, physical chip select lines can be mapped to DIMM chip select lines to locate and log memory errors.

[0011] In various examples, the point at which a DIMM rank has exceeded a threshold value can be determined independently of the hardware. A DIMM rank includes a set of DRAM chips connected to the same chip select. For instance, there can be no correlation between the DIMM rank error counters in the silicon and the emulated counters. That is, in some examples, the original architecture can be emulated where rank counter position can be determined independently of the silicon in the hardware. In some examples, determining rank counter position independently of the silicon can allow for locating and logging of memory errors on different systems and/or different operating systems.

[0012] Figure 1 illustrates a diagram of an example system 100 according to the present disclosure. As shown in Figure 1, the system 100 can include a database 102 accessible to and in communication with a plurality of engines 104. The plurality of engines 104 can include a physical rank engine 106, an array index engine 108, a decoding engine 110, and a chip select engine 112. The system 100 can include additional or fewer engines than illustrated to perform the various functions described herein and examples are not limited to the example shown in Figure 1. The system 100 can include hardware, for
example, in the form of transistor logic and/or application specific integrated
circuitry (ASICs), firmware, and software, for example, in the form of machine
readable and executable instructions (e.g., program instructions stored in a
machine readable medium), which, in cooperation can form the computing
device as discussed in connection with Figure 2.

[0013] The plurality of engines 104 can include a combination of
hardware and software (e.g., program instructions), but at least include
hardware configured to perform particular functions, tasks, and/or actions. For
example, the plurality of engines 104 shown in Figure 1 can be used to
determine physical rank information corresponding to a memory module,
convert the physical rank information into an array index, decode the physical
rank information using the array index to identify logical rank identification
information of the memory module, and determine a chip select in the memory
module using the logical rank identification information. As used herein, an
array index is a family or set of elements that contains at least one pointer to
identify an element of an array.

[0014] The physical rank engine 106 can determine physical rank
information corresponding to a memory module. As used herein, physical rank
information includes information corresponding to a plurality of dynamic random
access memory circuits connected to a chip select, as further discussed herein.
As used herein, a memory module includes a dynamic random access memory
circuit.

[0015] In some examples, the physical rank engine 106 can determine a
physical position of a rank counter in a string of rank counters. As used herein,
a rank counter is a counter that can locate and/or keep track of the position of a
specific location in a memory module. In some examples, the physical position
of a rank counter can correspond to a memory address on an address bus in
order to enable access to a particular storage cell on a memory module.

[0016] As used herein, a memory address is a fixed-length sequence of
digits that can be used to access a memory module. An address bus, for
example, is a computer bus that can be used to specify a physical address
(e.g., physical rank information). Further, a storage cell stores one bit of binary information.

[0017] In some examples, physical rank information can be determined from an error sourcing code. As used herein, error sourcing code is a set of program instructions that can locate single-bit (e.g., an error in one bit of data) and/or multi-bit failures (e.g., errors in more than one bit of data) in a memory module.

[0018] The array index engine 108 can convert the physical rank information into an array index. For example, translation data from a memory module, double data rate (DDR) channel, and/or DIMM rank can be input into the array index. The array index can replace the positional aspect of the rank counters. That is, in some examples, the array index can emulate where the rank counters can exist in silicon in a memory module, Remote Access Service (RAS) feature, and/or interleaving.

[0019] The decoding engine 110 can decode the array index to identify logical rank identification (ID) information corresponding to the memory module. Logical rank ID information can include an address where a particular item (e.g., a memory module location) can appear to reside from the perspective of application software.

[0020] In some examples, the decoding engine 110 can decode the array index through system address decode (SAD), target address decode (TAD), and/or a memory controller to convert the physical rank information down to the DIMM level. As used herein, SAD and TAD are address decoders that can have two or more bits of an address bus as inputs and one or more device selection lines as outputs. A memory controller manages data flow going to and from a memory module.

[0021] Decoding can take place at the system address level (e.g., 32 bit, 64 bit). For example, information encoded within addresses at the system address level can determine a chip select for a particular memory module. As used herein, a chip select is a control line used to select one chip out of several connected to the same computer bus. For example, a chip select can be used to select a particular chip out of a plurality of chips connected to the same bus.
In some examples, the memory node to be targeted can be determined from the chip select, as discussed further herein.

[0022] The chip select engine 112 can use the logical rank ID information to determine a chip select in the memory module. In some examples, the chip select can be used to locate error counts in a memory module independently of the silicon in a memory module. Further, the chip select engine 112 can count a number of error counts in the memory module and/or track and/or tabulate the number of error counts in the memory module. As used herein, error counts are the number of bit failures in the memory module.

[0023] For example, the chip select engine 112 can determine whether the number of error counts has exceeded a predefined threshold value. For example, the chip select engine can determine whether the number of error counts has exceed a predetermined number of errors over a predetermined time period (e.g., 3000 error counts in 24 hours).

[0024] Figure 2 illustrates a diagram of an example computing device according to the present disclosure. The computing device 201 can utilize hardware, software (e.g., program instructions), firmware, and/or logic to perform a number of functions described herein. The computing device 201 can be any combination of hardware and program instructions configured to share information. The hardware can, for example, include a processing resource 203 and a memory resource 205 (e.g., computer or machine readable medium (CRM/MRM), database, etc.). A processing resource 203, as used herein, can include one or more processors capable of executing instructions stored by the memory resource 205. The processing resource 203 may be implemented in a single device or distributed across multiple devices. The program instructions (e.g., computer or machine readable instructions (CRI/MRI)) can include instructions stored on the memory resource 204 and executable by the processing resource 203 to perform a particular function, task and/or action (e.g., determine physical rank information corresponding to a memory module, etc.).

[0025] The memory resource 205 can be a non-transitory machine readable medium, include one or more memory components capable of storing
instructions that can be executed by a processing resource 203, and may be integrated in a single device or distributed across multiple devices. Further, memory resource 205 may be fully or partially integrated in the same device as processing resource 203 or it may be separate but accessible to that device and processing resource 203. Thus, it is noted that the computing device 201 may be implemented on a participant device, on a server device, on a collection of server devices, and/or a combination of a participant, (e.g., user/consumer endpoint device), and one or more server devices as part of a distributed computing environment, cloud computing environment, etc.

[0026] The memory resource 205 can be in communication with the processing resource 203 via a communication link (e.g., a path) 219. The communication link 219 can provide a wired and/or wireless connection between the processing resource 203 and the memory resource 205.

[0027] In the example of Figure 2, the memory resource 205 includes physical rank instructions 216, array index instructions 218, decoding instructions 220, and chip select instructions 222. As used herein instructions include at least software that can be executed by a processing resource, for example, processing resource 203, to perform a particular task, function and/or action. The plurality of instructions may be combined or may be subroutines of other instructions. As shown in Figure 2, the physical rank instructions 216, the array index instructions 218, the decoding instructions 220, and the chip select instructions 222 can be individual instructions located on one memory resource 205. Examples are not so limited, however, and a plurality of instructions can be located at separate and distinct memory resource locations, for example, in a distributed computing environment, cloud computing environment, etc.

[0028] Each of the plurality of instructions can include instructions that when executed by the processing resource 203 can function as an engine such as the engines described in connection with Figure 1. For example, the physical rank instructions 216 can include instructions that when executed by the processing resource 203 can function as the physical rank engine 106. The array index instructions 218 can include instructions that when executed by the processing resource 203 can function as the array index engine 108. The
decoding instructions 220 can include instructions that when executed by the
processing resource 203 can function as the decoding engine 110. Additionally, the chip select instructions 222 can include instructions that when executed by the processing resource 203 can function as the chip select engine 112.

[0029] Examples are not limited to the example instructions shown in Figure 2 and in some cases a number of instructions can operate together to function as a particular engine. In addition, one or more engines described, or one or more instructions described may be combined or may be a sub-engine of another engine. Further, the engines and/or instructions of Figures 1 and 2 can be located in a single system and/or computing device or reside in separate distinct locations in a distributed network, cloud computing, enterprise service environment (e.g., Software as a Service (SaaS) environment), etc.

[0030] Figure 3 illustrates a flow diagram of an example of a process according to the present disclosure. At 321, physical rank information can be known or obtained. Physical rank information can include physical rank counter position, as described above in connection with Figures 1 and 2.

[0031] At 323, information corresponding to a memory node, DDR channel, DIMM rank, and/or translation data can be known and/or collected. As used herein, a memory node includes a particular location on a memory module. A DRR channel is a channel that can allow for transfer of data between DDR memory and a memory controller. For example, DIMM rank information can be collected (e.g., received) in response to a query or upon initiation of determining a chip select from physical rank information. Memory node information, DDR channel information, DIMM rank information and/or translation data can be output for conversion and/or indexing.

[0032] At 324, memory node information, DDR channel information, DIMM rank information, and/or translation data can be converted into an array index. In some examples, creating the array index can include converting the memory node information, DDR channel information, DIMM rank information, and/or translation data into an array index. In some examples, DIMM rank information can be collected in response to an input from a user.
At 326, a rank array containing physical to logical mapping information can be used to read logical rank ID information using the array index. For examples, the rank array can include entries that can determine logical rank ID information based on physical rank information and the array index. In some examples, the array can contain entries corresponding to DIMM rank information.

At 328, logical rank ID information can be read using the array index. In some examples, index mapping (e.g., mapping raw data from an array using an array index) can be used to read the logical rank ID information.

At 330, logical rank ID information can be determined and/or identified. For example, the logical rank of a chip select corresponding to the physical rank of a selected memory location can be determined and/or identified. In some examples, the logical rank ID information can be used to indict a physical DIMM slot that has accumulated correctable error counts. Indicting, as used herein, can include selecting a physical location of the DIMM and/or DIMM slot. For example, a DIMM that has accumulated correctable error counts in excess of a predetermined threshold value can be located and indicted independently of the hardware and silicon based on the logical rank identification information corresponding to the DIMM.

In some examples, the predetermined threshold value can be predetermined. For example, a predetermined threshold value of 3000 correctable errors in a 24 hour period can be selected; however, examples are not so limited, and the predetermined threshold value can be selected or assigned such that a variety of threshold values can be used (e.g., 100 errors in 10 minutes, 10,000 errors in 48 hours, etc.).

To indict (e.g., select a physical location) the correct DIMM for pre-failure warranty analysis independent of the silicon, a chip select can be used. Using a chip select can be beneficial when synchronicity between hardware is not present. For example, when visibility or synchronicity between the hardware and the position of an error (e.g., a physical rank position) is lost, the hardware can be emulated so that the silicon can be directly used.
In some examples, because an error sourcing code can execute asynchronously with respect to error signaling hardware, DIMM rank errors can be no longer discernable due to the time displacement between the actual error and when a memory code (e.g., firmware that determines how a computer's memory can be read and written) detects the error. In some examples, the memory code can detect an error through model specific registers (MSRs), for example, because the MSRs can refresh whenever a new error occurs. As used herein, MSRs include various control registers used for debugging, program execution tracing, computer performance monitoring, and toggling certain central processing unit (CPU) features.

Figure 4 illustrates a flow diagram for an example method according to the present disclosure. In various examples, the method 481 can be performed using the system 100 shown in Figure 1 and/or the computing device 201 and instructions shown in Figure 2. Examples are not, however, limited to these example systems, device, and/or instructions.

At 480, the method 481 can include receiving physical rank information corresponding to a memory module from an error sourcing code. The physical rank information can include a physical position of a rank counter in a string of rank counters.

At 482, the method 481 can include indexing the physical rank information into an array index. As previously discussed, the array index can be used in place of the positional aspect of the rank counters, thus separating the position of the rank counters in the hardware from the hardware itself.

At 484, the method 481 can include decoding the index to identify logical rank ID information corresponding to the memory module. Decoding can be performed at the system address level. For example, bits in addresses can determine what node in the CPU can be selected, targeted, and/or indicted. In some examples, bits in addresses can determine where a physical position of a rank counter is located.

At 486, the method 481 can include using the logical rank ID information to locate a chip select in the memory module. In some examples, the chip select can correspond to a location of a DIMM. For example, a DIMM
that has experienced correctable errors in excess of a predetermined threshold value can be indicted based on information corresponding to the chip select.

[0044] At 488, the method 481 can include using the chip select to locate correctable errors in the memory module. For example, once the correct DIMM to indict is determined from the chip select, correctable errors at that location can be corrected and/or fixed. In some examples, a number of correctable errors in the memory module can be determined and/or counted. If the number of counted correctable errors exceeds a predefined threshold value, the method 481 can include correcting the correctable errors in the memory module.

[0045] Figure 5 illustrates a diagram of an example system 590 including a processing resource 503 and a memory resource 505 (e.g., non-transitory computer readable medium) according to the present disclosure. For example, the system 590 may be an implementation of the example system 100 of Figure 1 and/or the example computing device 201 of Figure 2. Further, the system 590 can be used to implement the example process of Figure 3, and/or the example method 481 of Figure 4.

[0046] The processing resource 503 can be configured to execute instructions stored on the memory resource 505. For example, the memory resource 505 may be any type of volatile or non-volatile memory or storage, such as random access memory (RAM), flash memory, read-only memory (ROM), storage volumes, a hard disk, or a combination thereof.

[0047] The example memory resource 505 may store instructions 592 executable by the processing resource 503 to determine a chip select from physical rank information. For example, the processing resource 503 may execute the instructions 592 to receive physical rank information corresponding to a memory module (e.g., to perform block 480 of the example method of Figure 4, and/or in connection with the physical rank engine 106 of Figure 1, and/or the physical rank instructions 216 of Figure 2).

[0048] The example memory resource 505 may further store instructions 593 executable by the processing resource 503 to convert the physical rank information into an array index. In addition, processing resource 503 execute the instructions 593 for indexing the physical rank information into an array
index (e.g., to perform block 482 of the example method of Figure 4 and/or in connection with the array index engine 108 of Figure 1 and/or the array index instructions 218 of Figure 2).

[0049] The example memory resource 505 may further store instructions 594 executable by the processing resource 503 to decode the array index to identify logical rank identification information corresponding to the memory module (e.g., to perform block 484 of the example method of Figure 4 and/or in connection with the decoding engine 110 of Figure 1 and/or the decoding instructions 220 of Figure 2).

[0050] The example memory resource 505 may further store instructions 595 executable by the processing resource 503 to use the logical rank identification information to determine a chip select in the memory module (e.g., to perform block 486 of the example method of Figure 4 and/or in connection with the chip select engine 112 of Figure 1 and/or the chip select instructions 222 of Figure 2).

[0051] In the foregoing detailed description of the present disclosure, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration how examples of the disclosure can be practiced. These examples are described in sufficient detail to enable those of ordinary skill in the art to practice the examples of this disclosure, and it is to be understood that other examples can be utilized and that process, electrical, and/or structural changes can be made without departing from the scope of the present disclosure.

[0052] The figures herein follow a numbering convention in which the first digit corresponds to the drawing figure number and the remaining digits identify an element or component in the drawing. For example, reference numeral 102 can refer to element "02" in Figure 1 and an analogous element can be identified by reference numeral 202 in Figure 2. Elements shown in the various figures herein can be added, exchanged, and/or eliminated so as to provide a number of additional examples of the present disclosure. In addition, the proportion and the relative scale of the elements provided in the figures are intended to illustrate the examples of the present disclosure, and should not be
taken in a limiting sense. Further, as used herein, a number of an element and/or feature can refer to one or more of such elements and/or features.

[0053] As used herein, logic is an alternative or additional processing resource to perform a particular action and/or function, etc., described herein, which includes hardware, for example, various forms of transistor logic, application specific integrated circuits (ASICs), etc., as opposed to computer executable instructions, for example, software firmware, etc., stored in memory and executable by a processor.
What is claimed:

1. A non-transitory computer readable medium storing instructions executable by a processing resource of a system to cause the system to:
   receive physical rank information corresponding to a memory module;
   convert the physical rank information into an array index;
   decode the array index to identify logical rank identification (ID) information corresponding to the memory module; and
   use the logical rank ID information to determine a chip select in the memory module.

2. The non-transitory computer readable medium of claim 1, wherein the instructions are executable by the processing resource to use the chip select to locate correctable errors in the memory module.

3. The non-transitory computer readable medium of claim 1, wherein the instructions are executable by the processing resource to use the chip select to count a number of the correctable errors in the memory module.

4. The non-transitory computer readable medium of claim 3, wherein the instructions are executable by the processing resource to determine whether the number of correctable errors in the memory module exceeds a predefined threshold value.

5. The non-transitory computer readable medium of claim 4, wherein the predefined threshold value is 3000 correctable errors in 24 hours.

6. A system, comprising:
   a physical rank engine to determine physical rank information corresponding to a memory module;
   an array index engine to convert the physical rank information into an array index;
a decoding engine to identify logical rank identification (ID) information corresponding to the physical rank information using the array index; and

a chip select engine to use the logical rank ID information to determine a chip select in the memory module.

7. The system of claim 6, wherein the physical rank engine determines the physical rank information corresponding to the memory module from an error sourcing code.

8. The system of claim 6, wherein the decoding engine decodes bits in an address at a system address level to identify the logical rank ID information corresponding to the physical rank information.

9. The system of claim 6, wherein the decoding engine decodes the array index through a system address decode, a target address decode, and a memory controller to convert the physical rank information to a dual in-line memory module level.

10. A method, comprising:
    receiving physical rank information corresponding to a memory module from an error sourcing code;
    indexing the physical rank information into an array index;
    decoding the array index to identify logical rank identification (ID) information corresponding to the memory module;
    using the logical rank ID information to determine a chip select in the memory module; and
    using the chip select to locate correctable errors in the memory module.

11. The method of claim 10, wherein indexing the physical rank information into the array index includes mapping the logical rank ID information to an array using an index map.
12. The method of claim 10, wherein receiving the physical rank information includes receiving a physical position of a rank counter in a string of rank counters.

13. The method of claim 10, further comprising determining a plurality of correctable errors in the memory module.

14. The method of claim 13, further comprising determining whether the plurality of correctable errors in the memory module exceeds a predefined threshold value.

15. The method of claim 14, further comprising correcting the correctable errors in the memory module in response to the number of correctable errors in the memory module exceeding the predefined threshold value.
Figure 3
RECEIVING PHYSICAL RANK INFORMATION CORRESPONDING TO A MEMORY MODULE FROM AN ERROR SOURCING CODE

INDEXING THE PHYSICAL RANK INFORMATION INTO AN ARRAY INDEX

DECODING THE INDEX TO IDENTIFY LOGICAL RANK IDENTIFICATION INFORMATION CORRESPONDING TO THE MEMORY MODULE

USING THE LOGICAL RANK IDENTIFICATION INFORMATION TO DETERMINE A CHIP SELECT IN THE MEMORY MODULE

USING THE CHIP SELECT TO LOCATE CORRECTABLE ERRORS IN THE MEMORY MODULE

Figure 4
Figure 5
INTERNATIONAL SEARCH REPORT

International application No. PCT/US2014/061919

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/06; G06F 12/02; G11C 8/00; G06F 12/00; G11C 16/34; G11C 29/00; G11C 16/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 12/06; G06F 12/02; G11C 8/00; G06F 12/00; G11C 16/34; G11C 29/00; G11C 16/06

Documented 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: physical rank information, logical rank information, correctable error, threshold value, memory module, chip, select

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<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 2009-0177853 A1 (MARIO MAZZOLA et al.) 09 July 2009 See paragraphs [0027], [0055], [0066], [0070], [0077], [0084], [0091], [0102H0103], [0107], [0124]; figure 2.</td>
<td>1-3, 6-13</td>
</tr>
<tr>
<td>Y</td>
<td>US 2009-0132875 A1 (JUN KITAHARA et al.) 21 May 2009 See paragraph [0086]; and figure 5.</td>
<td>4-5, 14-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2011-0055458 A1 (REINHARD KUEHNE) 03 March 2011 See paragraphs [0053], [0073], [0077]; and figure 2.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>WO 2008-005735 A2 (SANDISK CORPORATION) 10 January 2008 See paragraphs [0049]-[0050], [0055], [0061], [0064], [0071], [0074], [0079], [0090]; and figure 4.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2002-0176310 A1 (MARK FRANCIS HILBERT) 28 November 2002 See paragraph [0017]; and figure 1.</td>
<td>1-15</td>
</tr>
</tbody>
</table>

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 18 June 2015 (18.06.2015)
Date of mailing of the international search report 19 June 2015 (19.06.2015)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongna-ro, Seo-gu, Daejeon Metropolitan City, 302-781, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262

Form PCT/ISA/210 (second sheet) (January 2015)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>us 2009--0177853 Al</td>
<td>09/07/2009</td>
<td>us 8407394 B2</td>
<td>26/03/2013</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 2014-0082457 Al</td>
<td>20/03/2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8612830 B2</td>
<td>17/12/2013</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8972823 B2</td>
<td>03/03/2015</td>
</tr>
<tr>
<td>us 2011--0055458 Al</td>
<td>03/03/2011</td>
<td>CN 102012867 A</td>
<td>13/04/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 102012867 B</td>
<td>24/12/2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104298610 A</td>
<td>21/01/2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T/2 201122812 A</td>
<td>01/07/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T/2 1474167 B</td>
<td>21/02/2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8688894 B2</td>
<td>01/04/2014</td>
</tr>
<tr>
<td>%o 2008--005735 A2</td>
<td>10/01/2008</td>
<td>us 2008-0002468 Al</td>
<td>03/01/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 7355892 B2</td>
<td>08/04/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>%o 2008-005735 A3</td>
<td>13/03/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wo 2002-095759 Al</td>
<td>28/11/2002</td>
</tr>
</tbody>
</table>