



(51) International Patent Classification:
G11C 29/42 (2006.01)

(21) International Application Number:
PCT/US2013/033679

(22) International Filing Date:
25 March 2013 (25.03.2013)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(72) Inventor: **LESARTRE, Gregg B.**; 3404 E. Harmony Road, Ft. Collins, Colorado 80528-9544 (US).

(74) Agents: **MCDOWELL, Robert David** et al.; Hewlett-Packard Company, Intellectual Property Administration, Mail Stop 35 3404 E. Harmony Road, Fort Collins, Colorado 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, 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, PA, 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).

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: MEMORY DEVICE HAVING ERROR CORRECTION LOGIC

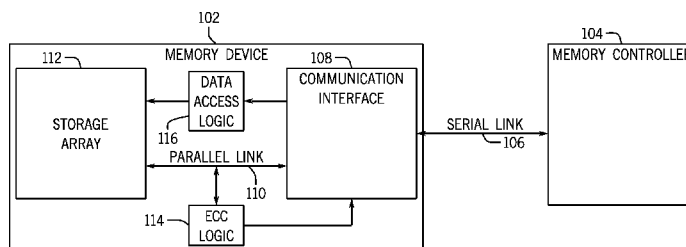


FIG. 1

(57) Abstract: Data is read from memory cells in the memory device. The read data is transferred over a link to a memory controller that is external of the memory device. While the transferring of the read data is ongoing, error detection of the read data is performed inside the memory device using an error correction code.

MEMORY DEVICE HAVING ERROR CORRECTION LOGIC

Background

[0001] A memory device can be used in a computing system to store data. In some cases, the memory device can be provided between a processor and a persistent storage subsystem, which is often implemented with lower speed storage device(s), such as disk-based storage device(s). The processor is able to perform data accesses at the memory device more quickly than at the persistent storage subsystem.

[0002] To address data errors that may occur in a memory device, an error detection and correction technique can be employed for protecting data stored in a memory device. Error correction codes can be stored with data in the memory device, and the error correction codes can be used for detecting and correcting errors in data read from a memory device. Error detection and correction can cause an increase in memory access latency, which can reduce the speed at which data of the memory device can be accessed.

Brief Description Of The Drawings

[0003] Some embodiments are described with respect to the following figures:

Fig. 1 is a block diagram of an example system that includes a memory controller and a memory device according to some implementations;

Fig. 2 is a flow diagram of an inline error detection and correction process according to some implementations; and

Fig. 3 is a block diagram of an example system according to further implementations.

Detailed Description

[0004] As dimensions of memory devices have continued to shrink, the memory devices may be more prone to suffer data errors, which have led to increased data error rates in the memory devices. There are various different types of memory devices that can be used in computing systems (e.g. desktop computers, notebook computers, tablet computers, smartphones, server computers, storage systems, communication nodes, etc.).

[0005] Examples of memory devices include dynamic random access memory (DRAM) devices, flash memory devices, phase change memory devices, memristor memory devices, and so forth. A memristor memory device can store data using memristors that store data based the resistance level of each cell. When a sufficiently large electrical current flows in a first direction through a memristor, the electrical resistance of the memristor increases. When a sufficiently large electrical current flows in the opposite direction through the memristor, the resistance of the memristor decreases. When the current is stopped, the memristor retains the last resistance that it had, which provides a representation of a data bit stored by the memristor.

[0006] A memory cell of a phase change memory device is also effectively a variable resistor. In some cases, the variable resistor can be formed of chalcogenide glass. To write to a phase change memory cell, the cell is heated, which melts the material, which is then cooled. The rate of cooling controls the crystal size of the resulting solid material, which controls the resistance of the memory cell. Different resistance levels can represent different data states.

[0007] In some cases, data error rates can be affected by how writes are performed to write data to a memory device. With a memristor memory device, for example, an electrical current flow is used for writing a data state to a particular memristor. A relatively low electrical current level can be used for writing memristors to extend the life of the memristors. However, using relatively low electrical current levels to write to memristors can result in higher rates of data bit errors.

[0008] Other types of memory devices are also associated with respective factors that can result in increased data error rates.

[0009] In some systems, error detection and correction can be performed in a memory controller that is used for managing access of a memory device (or multiple memory devices). However, performing error detection and correction at the memory controller can increase memory access latency. When error detection and correction is applied at the memory controller, the memory controller has to first receive read data from the memory device. After all of the read data (along with corresponding error correction codes) is received, the memory controller can then apply memory error detection and correction, which involves computations that may add to the overall time involved in providing data from the memory device to a requestor of the data. The requestor can be a processor or other device in the computing system.

[0010] In accordance with some implementations, an inline error detection and correction technique or mechanism is provided to perform inline error detection and correction of data within the memory device, as the data read from the memory device is delivered over a data link between the memory device and the memory controller. The data path inside the memory device may be a parallel path, where a collection (e.g. row) of data bits (along with bits corresponding to an error correction code) can be read out in parallel. Once read in parallel, the memory device can proceed to apply error detection and correction on the read data, while portions of the read data are communicated over a link between the memory device and the memory controller. The portions of the read data can be provided from the memory device to the memory controller before the error detection and correction computations at the memory device complete. As a result, the error detection and correction does not add to the data access latency in the normal case that an error is not detected.

[0011] Fig. 1 shows an example arrangement that includes a memory device 102 and a memory controller 104. The memory device 102 and the memory controller 104 are connected over a data link 106. In some examples, the data link 106 is a

serial link, such as a Serializer/Deserializer (SerDes) link. In other examples, the data link 106 can be a different type of data link.

[0012] The memory device 102 includes a communication interface 108, which allows the memory device 102 to communicate over the serial link 106. In some implementations, the communication interface 108 can include a parallel-to-serial converter and a serial-to-parallel converter to convert between a serial format (used on the serial link 106) and a parallel format (used on a parallel link 110 within the memory device 102). A parallel link 110 has multiple data paths over which data bits can be communicated in parallel. In contrast, the serial link 106 has one data path over which data bits are communicated serially.

[0013] The parallel link 110 is provided between a storage array 112 and the communication interface 108. The storage array 112 includes an array of memory cells, which can be DRAM cells, flash memory cells, memristor cells, phase change memory cells, and so forth, depending on the type of the memory device 102. More generally, a memory cell refers to circuitry used to store a data bit (or data bits).

[0014] The parallel link 110 allows a collection of data bits (e.g. a row of data bits or some other segment of data bits) to be retrieved at the same time from the storage array 112. For example, the collection of data bits can include read data as well as error correction code (ECC) bits. Once read in parallel from the storage array 112, the read data and ECC is available on the parallel link 110 to error detection and correction logic 114 (referred to herein as "ECC logic 114").

[0015] Also, the read data on the parallel link 110 can be provided to the communication interface 108, where the read data is converted to a format suitable for communication over the link 106. For example, as noted above, the communication interface 108 can apply a parallel-to-serial conversion.

[0016] In accordance with some implementations, during the time that the read data is being provided from the parallel link 110 through the communication interface 108 to the serial link 106, the ECC logic 114 can perform computations relating to error detection and correction on the read data. Examples of error detection and

correction techniques that can be applied include symbol-based ECC techniques, such as a Reed-Solomon ECC technique, or bit-based ECC techniques, such as a BCH technique. Although reference is made to example ECC techniques, it is noted that in other implementations, the ECC logic 114 can apply other types error detection and correction techniques.

[0017] The error detection and correction technique performed by the ECC logic 114 is considered to be an inline error detection and correction technique, in that the error detection and correction can be performed while the communication interface 108 is transferring portions of data read from the storage array 112 over the serial link 106 to the memory controller 104. Stated differently, the communication interface 108 is able to transfer read data from the storage array 112 of the memory device 102 over the serial link 106 to the memory controller 104, prior to completion of error detection and correction of the read data by the ECC logic 114 inside the memory device 102.

[0018] By performing inline error detection and correction using the ECC logic 114, data read from the storage array 112 is protected against errors without adding to the memory access time. For example, if the memory device 102 were to wait until the error detection and correction were completed by the ECC logic 114 before transferring the read data from the memory device 102 to the memory controller 104, then memory access latency would be increased.

[0019] The memory device 102 also includes data access logic 116, which is responsive to a command received from the memory controller 104 through the communication interface 108 to issue respective commands to the storage array 112. The request received from the memory controller 104 can be a read request or a write request.

[0020] If the ECC logic 114 determines that there is no error in the read data that is transferred from the memory device 102 to the memory controller 104, then the ECC logic 114 does not have to take any further action, since the read data has been transferred to the memory controller 104 is without error.

[0021] However, if the ECC logic 114 determines that an error is present in the read data that is or has been transferred to the memory controller 104, then the ECC logic 114 can provide an indication to the memory controller 104 regarding the presence of an error in the transferred data. The indication can be an indication that the read transaction has failed. The indication that the transaction has failed can be an explicit failed indication (e.g. a signal asserted to a specified state). Alternatively, the indication can be implicitly communicated to the memory controller 104 by “poisoning” a check code that is used to protect communication over the serial link 106. An example of a check code is a cyclic redundancy check (CRC) code. To protect data transferred over the serial link 106, the communication interface 108 can produce a CRC code, which is calculated based on the content of the data to be transferred over the serial link 106. The CRC code can then be provided with the actual data over the serial link 106. At the memory controller 104, the memory controller re-generates the CRC code based on the received data, and compares the re-generated CRC code with the received CRC code. If the CRC codes match, then that means that the data transfer over the serial link 106 was successful. However, if the CRC codes do not match, then that is an indication that a data communication error has occurred over the serial link 106. Poisoning the CRC code refers to deliberately making the CRC code wrong or invalid such that the CRC code comparison performed at the memory controller 104 in response to the received data would result in a determination of a data communication error.

[0022] Once the ECC logic 114 detects an error in the data, in some implementations, the ECC logic 114 can correct the error, assuming that the data error is of a type that is correctable using the associated error correction code. For example, the error correction code associated with a specific collection of data bits can correct up to n bits of error, where $n \geq 1$.

[0023] When the memory controller 104 receives the indication of failed transaction (either explicit indication or implicit indication as discussed above), in some implementations, the memory controller 104 can retry a read of the erroneous data. In response to the retried read, the communication interface 108 can provide the corrected data (corrected by the ECC logic 114) to the memory controller 104.

[0024] In alternative implementations, instead of the memory controller 104 performing a read retry in response to a failed read transaction indication, the memory device 102 can send the ECC to the memory controller 104 if the ECC logic 114 detects an error in the data that can be corrected. In such implementations, the ECC logic 114 does not perform the error correction—instead, the error correction is performed by the memory controller 104.

[0025] The memory controller 104 can stall the return of data to a requestor (e.g. processor or other requestor) until the memory controller 104 has completed the error correction of the data. Furthermore, in such implementations, the memory controller 104 can proceed to correct the data stored in the memory device 102 by writing the corrected data back to the memory device 102.

[0026] Fig. 2 is a flow diagram of an inline error detection and correction process, according to some implementations. The process of Fig. 2 can be performed by the memory device 102, for example. The process reads (at 202), in the memory device 102, data from the memory cells of the storage array 112 in the memory device 102. The read data is transferred (at 204) over the serial link 106 to the memory controller 104. While the transferring of the read data is ongoing, error detection of the data read from the memory cells is performed (at 206) by the ECC logic 114 inside the memory device 102. In other words, the error detection of the read data is performed concurrently with the transfer of the read data over the serial link 106.

[0027] Fig. 3 is a block diagram of an example system 300 according to alternative implementations. In addition to the memory device 102 and the memory controller 104, a requestor 301 is also present in the system 300. The requestor 301 is able to issue data requests (read requests or write requests) for accessing data in the memory device 102. In response to a data request from requestor 301, the memory controller 104 issues a respective data request to the memory device 102.

[0028] As further depicted in Fig. 3, the communication interface 108 in the memory device 102 includes a buffer 302 and a parallel-serial converter 304. The buffer 302 is used to temporarily store data that is read from the storage array 112 that is to be delivered over the serial link 106 to the memory controller 104. The parallel-serial

converter 304 converts data between the parallel format on the parallel link 110 and the serial format on the serial link 106.

[0029] The memory controller 104 includes error handling logic 306, which is responsive to a failed read transaction indication 308 from the memory device to perform an error handling procedure. The error handling procedure can include sending a read retry 310 to the memory device 102, to perform a retry of the data read. The error handling procedure can also alternatively wait for an error correction code to be received from the memory device 102 for use in correcting the received data. Also, the error handling procedure can include performing a write of the corrected data back to the memory device 102.

[0030] By using techniques or mechanisms according to some implementations, data correctness can be checked without adding to the access time latency, except in situations where an error occurs, in which case extra time would be added to perform the data handling procedure to correct the data error. When there is no data error, an error correction code does not have to be provided from the memory device 102 to the memory controller 104, in some implementations. This can reduce consumption of the data bandwidth over the serial link 106, since error correction codes do not have to be provided in non-error situations.

[0031] In the foregoing description, numerous details are set forth to provide an understanding of the subject disclosed herein. However, implementations may be practiced without some or all of these details. Other implementations may include modifications and variations from the details discussed above. It is intended that the appended claims cover such modifications and variations.

What is claimed is:

- 1 1. A memory device comprising:
2 a communication interface to communicate over a link to a memory controller
3 that is external of the memory device;
4 data access logic to read data from memory cells, wherein the communication
5 interface is to output the read data through the communication interface over the link;
6 and
7 error correction logic to perform error detection of the read data using an error
8 correction code associated with the read data, wherein the error correction logic is to
9 perform the error detection concurrently with portions of the read data being
10 communicated over the link.
- 1 2. The memory device of claim 1, wherein the link is a serial link, and wherein
2 the communication interface includes a parallel-serial converter to convert between
3 data of a parallel format used internally in the memory device and data of a serial
4 format used on the serial link.
- 1 3. The memory device of claim 2, further comprising a parallel link in the
2 memory device, wherein the error correction logic is to receive the read data over the
3 parallel link from the memory cells.
- 1 4. The memory device of claim 1, wherein the error correction logic is to output
2 an indication of failure to the memory controller in response to the error correction
3 logic detecting an error in the read data.
- 1 5. The memory device of claim 4, wherein the failure indication includes a signal
2 asserted to a specific state.
- 1 6. The memory device of claim 4, wherein the failure indication is provided by
2 poisoning a check code associated with transfer of the read data over the link.

1 7. The memory device of claim 4, wherein the memory device is to receive a
2 read retry from the memory controller, where the read retry is responsive to the
3 failure indication.

1 8. The memory device of claim 4, wherein the error correction logic is to output
2 the error correction code to the memory controller in response to detecting the error,
3 and to decline to output the error correction code if no error is detected in the read
4 data.

1 9. The memory device of claim 8, wherein provision of the error correction code
2 to the memory controller allows the memory controller to correct the read data using
3 the error correction code.

1 10. A method comprising:
2 reading, in a memory device, data from memory cells in the memory device;
3 transferring the read data over a link to a memory controller that is external of
4 the memory device; and
5 while the transferring of the read data is ongoing, performing, inside the
6 memory device, error detection of the read data using an error correction code.

1 11. The method of claim 10, wherein the transferring of the read data over the link
2 to the memory controller occurs prior to completion of the error detection of the read
3 data inside the memory device.

1 12. The method of claim 10, further comprising:
2 determining, by the memory device, whether error exists in the read data;
3 in response to determining that no error exists in the read data, declining to
4 send the error correction code to the memory controller.

1 13. The method of claim 12, further comprising:
2 in response to determining that error exists in the read data, sending a failure
3 indication to the memory controller.

1 14. The method of claim 13, wherein the failure indication is an explicit failure
2 indication or an implicit failure indication.

1 15. A system comprising:
2 a memory controller;
3 a memory device; and
4 a link interconnecting the memory controller and the memory device, wherein
5 the memory device comprises:
6 a communication interface to communicate over a link to a memory
7 controller that is external of the memory device;
8 data access logic to read data from memory cells; and
9 error correction logic to perform error detection of the read data using
10 an error correction code associated with the read data, wherein the communication
11 interface is to output portions of the read data through the communication interface
12 over the link prior to completion of the error detection of the read data.

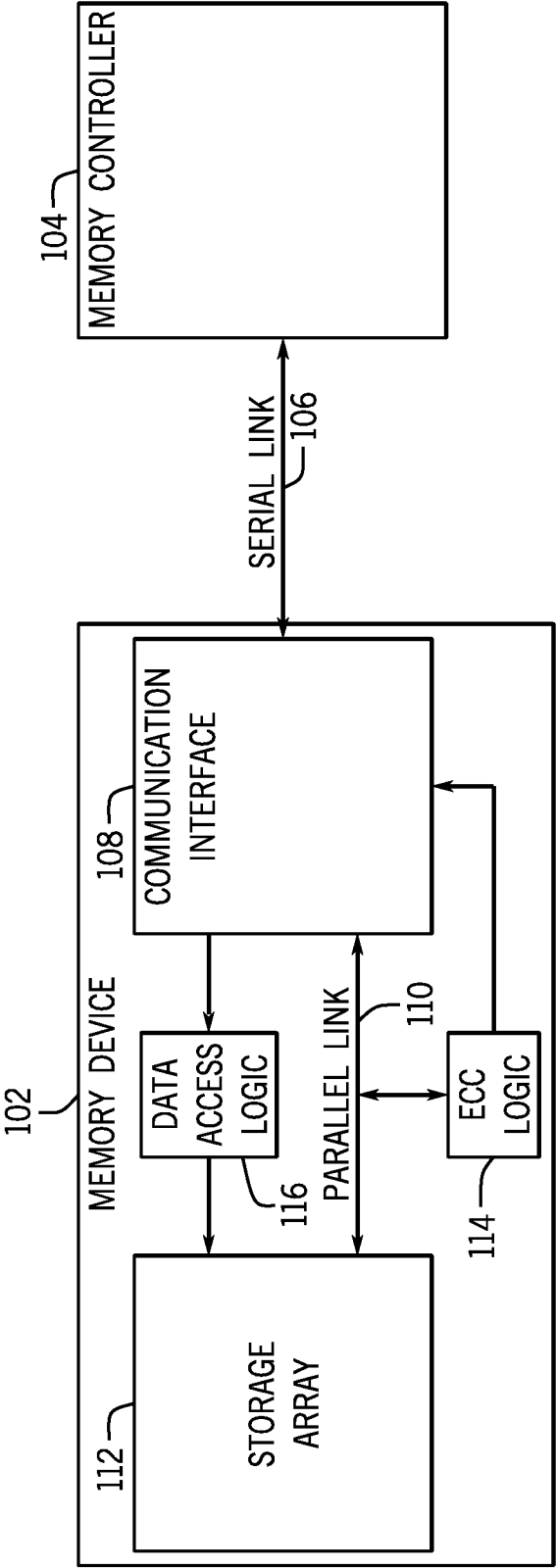


FIG. 1

2 / 3

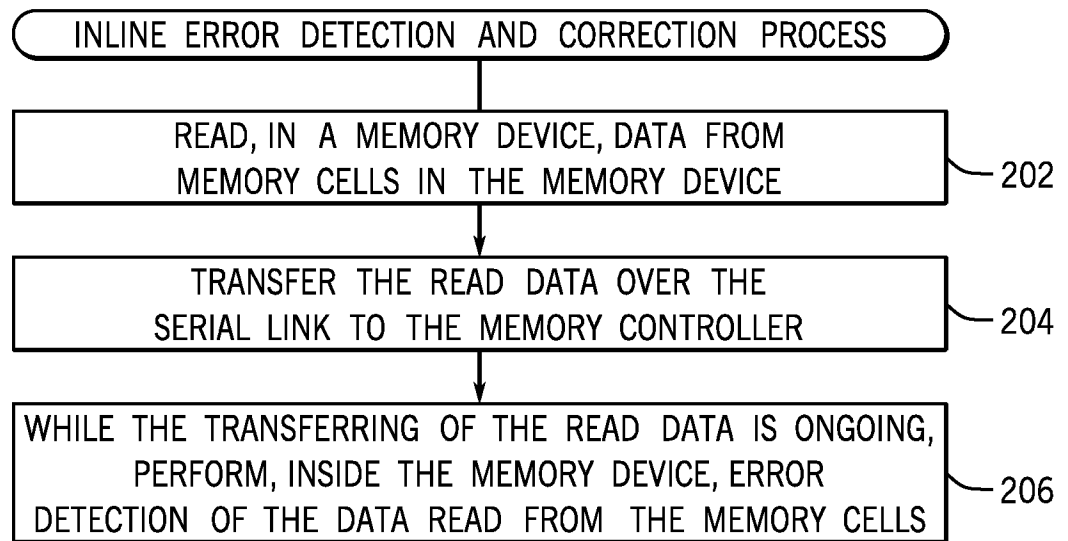


FIG. 2

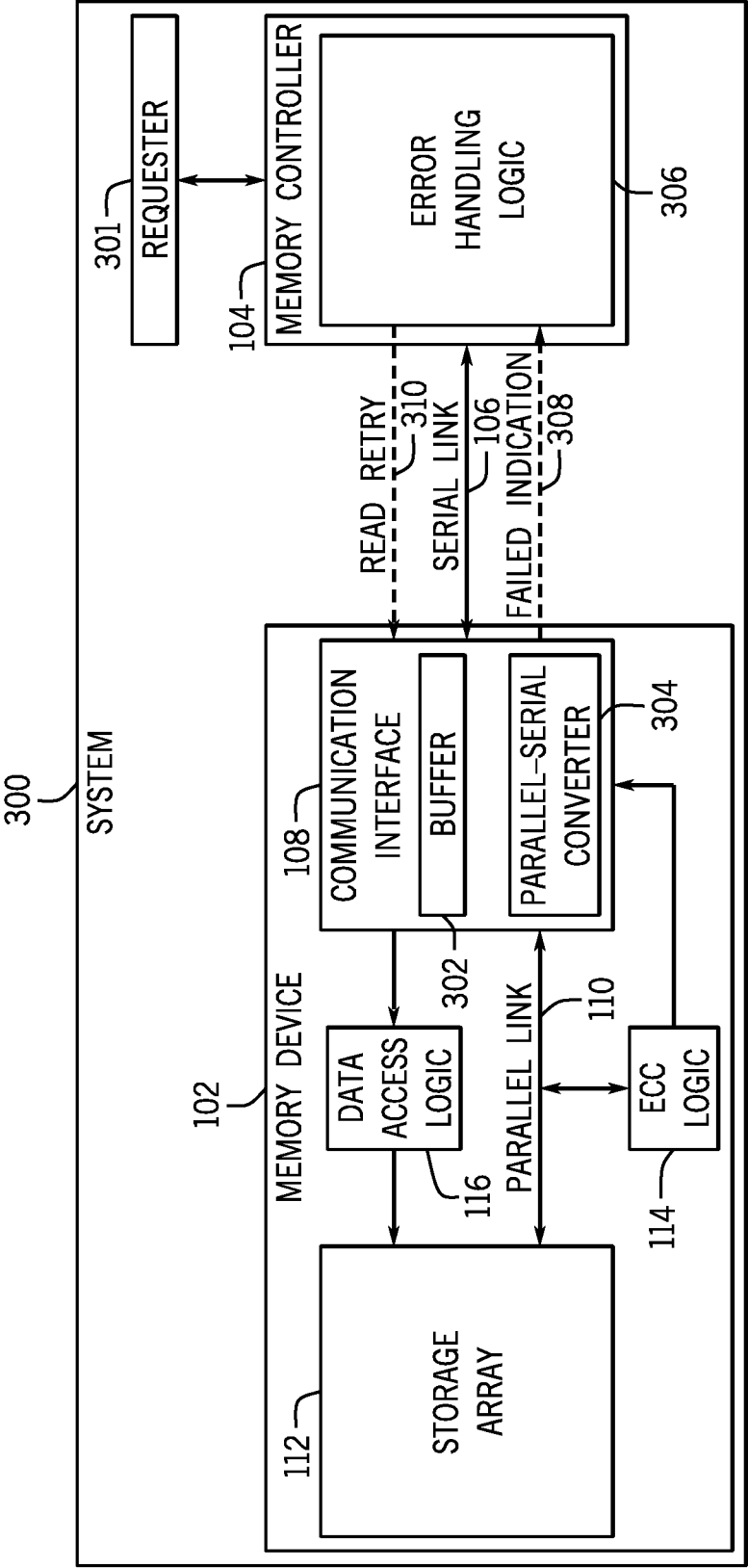


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/033679**A. CLASSIFICATION OF SUBJECT MATTER****G11C 29/42(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 29/42; G06F 11/07; H03M 13/09; G06F 11/10; H03M 13/07; H03M 13/29

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: memory device, error correction logic, communication interface, parallel-serial converter and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012-0266041 A1 (DAVID WANG et al.) 18 October 2012	1,4-15
Y	See claims 1, 2, 4, 12; paragraphs 34-38; and figures 1-4.	2-3
Y	US 2011-0231737 A1 (KENSHI DACHIKU) 22 September 2011	2-3
	See claims 7, 8; and figure 6.	
A	US 2008-0163033 A1 (YONG-TAE YIM) 03 July 2008	1-15
	See claim 1; and figures 4-9.	
A	US 2013-0007563 A1 (JONG-WOOK PARK) 03 January 2013	1-15
	See abstract, claim 1; and figures 1-23.	
A	US 2011-0107191 A1 (HOE-JU CHUNG et al.) 05 May 2011	1-15
	See abstract, claim 1; and figures 2-9.	



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

30 December 2013 (30.12.2013)

Date of mailing of the international search report

30 December 2013 (30.12.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,
302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

HWANG, Yun Koo

Telephone No. +82-42-481-5715



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/033679

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0266041 A1	18/10/2012	None	
US 2011-0231737 A1	22/09/2011	JP 04940322 B2 JP 2011-192206 A KR 10-1252382 B1 KR 10-2011-0104428 A US 8495468 B2	30/05/2012 29/09/2011 08/04/2013 22/09/2011 23/07/2013
US 2008-0163033 A1	03/07/2008	CN 101211667 A CN 101211667 C0 DE 102007058828 A1 JP 2008-165808 A KR 10-0856129 B1 KR 10-2008-0062306 A US 8091010 B2	02/07/2008 02/07/2008 03/07/2008 17/07/2008 03/09/2008 03/07/2008 03/01/2012
US 2013-0007563 A1	03/01/2013	KR 10-2013-0004045 A	09/01/2013
US 2011-0107191 A1	05/05/2011	CN 101114271 A CN 101114271 B CN 101114271 C0 DE 102007008180 A1 JP 05000324 B2 JP 2007-220278 A JP 2008-034095 A KR 10-0746225 B1 KR 10-0772841 B1 KR 10-0781974 B1 KR 10-0781983 B1 KR 10-0850277 B1 TW 200813724 A US 2007-0204199 A1 US 2008-0077937 A1 US 2008-0172516 A1 US 2011-0107006 A1 US 7870326 B2 US 7882417 B2 US 7941612 B2	30/01/2008 04/04/2012 30/01/2008 13/09/2007 15/08/2012 30/08/2007 14/02/2008 30/07/2007 02/11/2007 06/12/2007 06/12/2007 04/08/2008 16/03/2008 30/08/2007 27/03/2008 17/07/2008 05/05/2011 11/01/2011 01/02/2011 10/05/2011