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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 August 2016 (04.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/122655 A1

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

(21) International Application Number:
PCT/US2015/013954

(22) International Filing Date:
30 January 2015 (30.01.2015)

(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) Inventors: **BENEDICT, Melvin K.**; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US). **WARNES, Lidia**; 8000 Foothills Blvd., Roseville, California 95747 (US). **SCHOELLER, Patrick M.**; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US).

(74) Agents: **SHOOKMAN, Jeb A.** 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))*

[Continued on next page]

(54) Title: VALIDATING DATA IN A STORAGE ARRAY

(57) Abstract: Validating data in a storage array includes receiving, from a memory controller, a write command to write data to a storage array of a storage element, transferring the data with a cyclic redundancy check (CRC) character to the storage element to create a transfer, writing, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character, and validating, via a sense amplifier, the written data.

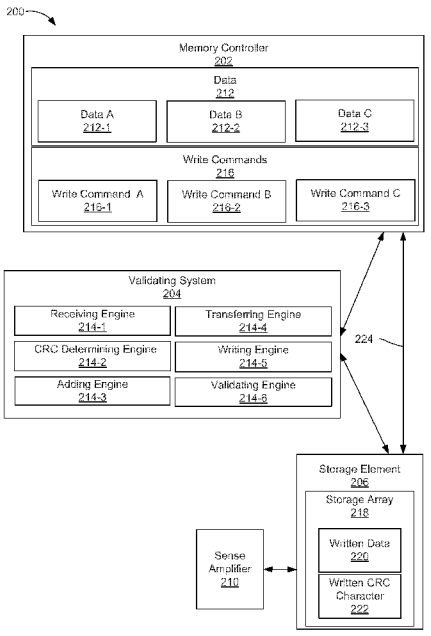

Fig. 2

WO 2016/122655 A1

— *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))*

84112767

VALIDATING DATA IN A STORAGE ARRAY

BACKGROUND

[0001] A memory controller is a digital circuit that manages the flow of data to and from storage elements of a system. The flow may be executed by a memory command cycle. The memory command cycle may include a number of commands such as a data payload command, an activate command, and a read command or a write command. The write command may write the data to the storage elements. The storage elements may be dynamic random-access memory (DRAM), synchronous dynamic random-access memory (SDRAM), non-volatile memory such as flash memory, other memory devices, or combinations thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various examples of the principles described herein and are a part of the specification. The examples do not limit the scope of the claims.

[0003] Fig. 1 is a diagram of a system for validating data in a storage array, according to one example of principles described herein.

[0004] Fig. 2 is a diagram of a system for validating data in a storage array, according to one example of principles described herein.

[0005] Fig. 3 is a diagram of a system for validating data in a storage array, according to one example of principles described herein.

[0006] Fig. 4 is a diagram of a system for validating data in a storage array via comparing cyclic redundancy check (CRC) characters, according to one example of principles described herein.

[0007] Fig. 5 is a diagram of a system for validating data in a storage array via comparing data, according to one example of principles described herein.

[0008] Fig. 6 is a diagram of a system for validating data in a storage array, according to one example of principles described herein.

[0009] Fig. 7 is a flowchart of a method for validating data in a storage array, according to one example of principles described herein.

[0010] Fig. 8 is a flowchart of a method for validating data in a storage array, according to one example of principles described herein.

[0011] Fig. 9 is a diagram of a validating system, according to one example of principles described herein.

[0012] Fig. 10 is a diagram of a validating system, according to one example of principles described herein.

[0013] Throughout the drawings, identical reference numbers designate similar, but not necessarily identical, elements.

DETAILED DESCRIPTION

[0014] As mentioned above, a memory controller is a digital circuit that manages the flow of data to and from storage elements. Further, the memory controller writes the data to a storage array of the storage element via a write command.

[0015] During a write command, the data may be subjected to timing errors, power supply noise, or internal storage array nose. This may alter the data. As a result, errors may be introduced into the data while executing the write command. If errors are introduced into the data, incorrect data is stored in the storage elements. Further, when an error is detected in the data, the entire data is retransmitted to correct the error. If the data is large in size, this can

negatively impact the performance of the memory controller and the storage elements.

[0016] The principles described herein include a method for validating data in a storage array. Such a method includes receiving, from a memory controller, a write command to write data to a storage array of a storage element, transferring the data with a cyclic redundancy check (CRC) character to the storage element to create a transfer, writing, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character, validating, via a sense amplifier, the written data. Such a method allows the data to be validated without transferring the data back to the memory controller. As a result, the use of the memory controller's bandwidth is reduced.

[0017] In the present specification and in the appended claims, the term "memory controller" means is a digital circuit that manages the flow of data to and from at least one storage element. The flow may be executed by a memory command cycle. The memory command cycle may include a number of commands such as a data payload command, an activate command, and a read command or a write command. The read command may read data from the storage element. The write command may write the data to the storage element.

[0018] In the present specification and in the appended claims, the term "storage element" means a mechanism to store data. The data may be stored in a storage array of a storage array. The storage element may include dynamic random-access memory (DRAM), synchronous dynamic random-access memory (SDRAM), non-volatile memory such as flash memory, other memory devices or combinations thereof.

[0019] In the present specification and in the appended claims, the term "write command" means executable instructions to write data to a storage array of a storage element. The write command may be issued via a memory controller.

[0020] In the present specification and in the appended claims, the term "CRC character" means a mechanism used to detect accidental changes

made to data while transferring the data from a memory controller to a storage element. A CRC character may be a short, fixed-length binary sequence that is concatenated to the data before the data is transferred to the storage element.

[0021] In the present specification and in the appended claims, the term "written CRC character" means a mechanism used to detect accidental changes made to data while transferring the data from a storage element to a storage array of the storage element. A written CRC character may be used for validation purposes to determine if the written data in the storage array is valid or invalid.

[0022] In the present specification and in the appended claims, the term "data" means information that is to be transferred between a memory controller and a storage array of a storage element. The data may be associated with a size.

[0023] In the present specification and in the appended claims, the term "written data" means data that is stored in a storage array of a storage element. The written data may be valid or invalid as determined by a validating system.

[0024] Further, as used in the present specification and in the appended claims, the term "a number of" or similar language is meant to be understood broadly as any positive number comprising 1 to infinity; zero not being a number, but the absence of a number.

[0025] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present systems and methods. It will be apparent, however, to one skilled in the art that the present apparatus, systems, and methods may be practiced without these specific details. Reference in the specification to "an example" or similar language means that a particular feature, structure, or characteristic described in connection with that example is included as described, but may not be included in other examples.

[0026] Referring now to the figures, Fig. 1 is a diagram of a system for validating data in a storage array, according to one example of principles described herein. As will be described below, a validating system is in

communication with a memory controller and a storage element to receive, from the memory controller, a write command to write data to a storage array of a storage element. The validating system transfers the data with a CRC character to the storage element to create a transfer. Further, the validating system writes, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character. The validating system validates, via a sense amplifier, the written data.

[0027] As illustrated, the system (100) includes a memory controller (102). The memory controller (102) may be a digital circuit that manages the flow of a data to and from a storage element (106) of the system (100). The flow may be executed by a number of commands. The data may be transferred via a bus (108). More information about the memory controller (102) will be described in other parts of this specification.

[0028] The system (100) further includes the storage element (106). The storage element (106) may include a number of types of memory technologies as long as the types of memory technologies use the same protocol. The storage element (106) may be in communication with the memory controller (102) via the bus (108). More information about the storage elements (106) will be described in other parts of this specification

[0029] The system (100) further includes a validating system (104). The validating system (104) is in communication with the memory controller (102) and the storage element (106).

[0030] The validating system (104) receives, from the memory controller (102), a write command to write data to a storage array of the storage element (106). The write command may include a starting address associated with the storage array of storage element (106). The starting address determines a location in the storage array of storage element (106) to start writing the data.

[0031] The validating system (104) further transfers the data with a CRC character to the storage element (106) to create a transfer. The CRC character may be used to detect accidental changes made to data while

transferring the data from the memory controller (102) to the storage element (106).

[0032] Further, the validating system (104) writes, based on the transfer, the data with the CRC character to the storage array of the storage element (106) to create written data and a written CRC character. As a result, the data is written to the storage array of the storage element (106).

[0033] A validating engine (114) of the validating system (104) further validates, via a sense amplifier, the written data. The validating system (104) may validate the written data at an array storage level. Such a method allows the data to be validated without transferring the data back to the memory controller (102). As a result, the use of the memory controller's bandwidth is reduced. More information about the validating system (104) will be described later on in this specification.

[0034] While this example has been described with reference to the validating system being located between the memory controller and the storage element, the validating system may be located in any appropriate location according to the principles described herein. The validating system may be located on the memory controller, the storage element, other locations, or combinations thereof.

[0035] Fig. 2 is a diagram of a system for validating data in a storage array, according to one example of principles described herein. As mentioned above, a validating system is in communication with a memory controller and a storage element to receive, from the memory controller, a write command to write data to a storage array of a storage element. The validating system transfers the data with a CRC character to the storage element to create a transfer. Further, the validating system writes, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character. The validating system validates, via a sense amplifier, the written data.

[0036] As illustrated, the system (200) includes a memory controller (202). As mentioned above, the memory controller (202) may be a digital circuit that manages the flow of a data to and from a storage element (206) of the

system (200). The flow may be executed by a number of commands. As mentioned above, a write command may write the data to the storage element (206). The data may be written to the storage element (206) by the memory controller (202) via a bus (224). As a result, the data may be transferred via the bus (224).

[0037] As illustrated, the memory controller (202) may include data (212). The data (212) may be information that is to be transferred between the memory controller (202) and the storage element (206). In Fig. 2, the data (212) includes data A (212-1), data B (212-2), and data C (212-3). The size of the data (212) may vary based on the amount of information. The more information the larger the size of the data (212). In an example, Data A (212-1) may be one-kilobyte. Data B (212-2) may be sixty-four bytes. Data C (212-3) may be thirty-two bytes.

[0038] As illustrated, the memory controller (202) may include a number of write commands (216). The write commands (216) may be mechanisms to write the data (212) to a storage array (218) of the storage element (206). As illustrated, the write commands (216) include write command A (216-1), write command B (216-2), and write command C (216-3). Write command A (216-1) may be a write operation to write data A (212-1) to the storage array (218) of the storage element (206). Write command B (216-2) may be a write operation to write data B (212-2) to the storage array (218) of the storage element (206). Write command C (216-3) may be a write operation to write data C (212-3) to the storage array (218) of the storage element (206).

[0039] Further, the write commands (216) may include a starting address. The starting address determines a location in the storage array (218) of storage element (206) to start writing the data. For example, write command A (216-1) may indicate to write data A (212-1) to a start address of 0x002F for the storage array (218). As a result, data A (212-1) is written to the start address of 0x002F of the storage array (218).

[0040] The system (200) further includes the storage element (206). The storage element (206) may include a number of types of memory technologies. The types of storage element (206) may include DRAM, SRAM,

non-volatile memory such as flash memory, other storage element, or combinations thereof. As illustrated, the storage element (206) includes the storage array (218). As will be described below, the storage array (218) stores written data (220) and written CRC characters (222).

[0041] The system (200) further includes a validating system (204). In one example, the validating system (204) includes a processor and computer program code. The computer program code is communicatively coupled to the processor. The computer program code includes a number of engines (214). The engines (214) refer to program instructions designed to perform a designated function. The computer program code causes the processor to execute the designated function of the engines (214). In other examples, the engines (214) refer to a combination of hardware and program instructions to perform a designated function. Each of the engines (214) may include a processor and memory. The program instructions are stored in the memory and cause the processor to execute the designated function of the engine. As illustrated, the validating system (204) includes a receiving engine (214-1), a CRC determining engine (214-2), an adding engine (214-3), a transferring engine (214-4), a writing engine (214-5), and a validating engine (214-6).

[0042] As mentioned above, the validating system (204) includes a receiving engine (214-1). The receiving engine (214-1) receives, from the memory controller (202), a write command (216) to write data (212) to the storage array (218) of the storage element (206). The receiving engine (214-1) may receive write command A (216-1), write command B (216-2), write command C (216-2), or combinations thereof. As will be described in other parts of this specification each of the write commands (216) may be stored by the receiving engine (214-1) in a first in first out (FIFO) queue. As a result, if write command A (216-1) is received, by the receiving engine (214-1), before write command B (216-2), write command A (216-1) is executed before write command B (216-2).

[0043] As mentioned above, the validating system (204) includes a CRC determining engine (214-2). The CRC determining engine (214-2) determines a CRC character for the data. The CRC character may be a short,

fixed-length binary sequence that is concatenated to the data. The CRC character may be eight bits.

[0044] As mentioned above, the validating system (204) includes an adding engine (214-3). The adding engine (214-3) adds the CRC character to the data. In some examples, the adding engine (214-3) adds the CRC character to the front of the data via concatenation. In other examples, the adding engine (214-3) adds the CRC character to the end of the data via concatenation.

[0045] As mentioned above, the validating system (204) includes a transferring engine (214-4). The transferring engine (214-4) transfers the data with a CRC character to the storage element (206) to create a transfer. For example, if write command A (216-1) is executing, data A (212-1) is transferred with a CRC character to the storage element (206).

[0046] The transferring engine (214-4) transfers the data with the CRC character to the storage element (206) to create the transfer by obtaining the CRC character associated with the data before the transfer before the transfer. Further, the transferring engine (214-4) recalculates the CRC character associated with the data after the transfer to create a recalculated CRC character.

[0047] Further, the transferring engine (214-4) compares the CRC character with the recalculated CRC character to determine if an error has occurred while transferring the data. If the CRC character and the recalculated CRC character match, no errors have occurred while transferring the data from the memory controller (202) to the storage element (206). As a result, a writing engine (214-5) writes the data with the CRC character to the storage array (218) of the storage element (206) to create written data (220) and a written CRC character (222).

[0048] If the CRC character and the recalculated CRC character do not match, an error has occurred while transferring the data from the memory controller (202) to the storage element (206). As a result, a writing engine (214-5) does not writes the data with the CRC character to the storage array (206) of the storage element (218) to create written data (220) and a written CRC

character (222). Further, the validating system (204) may retransmit this data to the storage element (206).

[0049] As mentioned above, the validating system (204) includes a validating engine (214-6). The validating engine (214-6) validates, via a sense amplifier (210), the written data (220). As will be described in other parts of this specification, the validating engine (214-6) validates the written data (220) via CRC characters. The validating engine (214-6) validates the written data (220) by obtaining the recalculated CRC character and obtaining the written CRC character (222). The written CRC character (222) may be obtained via the sense amplifier (210).

[0050] Further, the validating engine (214-6) validates the written data (220) by comparing the recalculated CRC character with the written CRC character (222) to create a CRC comparison. The validating engine (214-6) determines, based on the CRC comparison, if the written data (220) is valid or invalid. If the CRC comparison illustrates that the recalculated CRC character and the written CRC character match, the written data (220) is valid. However, if the CRC comparison illustrates that the recalculated CRC character and the written CRC character (222) do not match, the written data (220) is invalid.

[0051] As will be described in other parts of this specification, the validating engine (214-6) may validate the written data (220) via the data itself. The validating engine (214-6) validates the written data (220) by obtaining the data and obtaining the written data (220). The data may be the data (212) stored in the memory controller (212). The written data (220) may be obtained via the sense amplifier (210).

[0052] Further, the validating engine (214-6) validates the written data (220) by comparing the data (212) with the written data (220) to create a data comparison. The validating engine (214-6) determines, based on the data comparison, if the written data (220) is valid or invalid.

[0053] If the data comparison illustrates the data (212) and the written data (220) match, the written data (220) is valid. However, if the data comparison illustrates the data (212) and the written data (220) do not match, the written data (220) is invalid.

[0054] An overall example of the system of Fig. 2 will now be described. The receiving engine (214-1) receives, from the memory controller (202), write command A (216-1) to write data A (212-1) to the storage array (218) of the storage element (206). The CRC determining engine (214-2) determines an eight bit CRC character for data A (212-1). The adding engine adds the CRC character to data A (212-1) via concatenation. The transferring engine (214-4) transfers data A (212-1) with the CRC character to the storage element (206) to create a transfer. The transferring engine (214-4) validates data A (212-1) via the CRC character and a recalculated CRC character. If data A (212-1) is valid, the writing engine (214-5) writes data A (212-1) with the CRC character to the storage array (218) of the storage element (206) to create written data (220) and a written CRC character (222). The validating engine (214-6) validates, via the sense amplifier (210), the written data (220) as described above.

[0055] Fig. 3 is a diagram of a system for validating data in a storage array, according to one example of principles described herein. As mentioned above, a validating system is in communication with a memory controller and a storage element. As will be described below, elements of the memory controller and the storage element may aid the validating system in validating written data in the storage array.

[0056] As illustrated, the system (300) includes a memory controller (302). The memory controller (302) may include system interfaces (310). The system interfaces (310) may allow the memory controller (302) to communicate with other elements in the system (300). Further, the memory controller (302) may include data path logic (312). The data path logic (312) may allow the memory controller (312) to execute various functions. Further, the memory controller (302) may include a CRC character generator (314). The CRC character generator (314) may include processors, memory, and instructions to aid the CRC determining engine of the validating system (304) in determining a CRC character for the data.

[0057] As illustrated, the system (300) includes a storage element (306). As mentioned above, the storage element (306) includes a storage array

(316) to store written data. Further, the storage element (306) includes a CRC character checker (318). The CRC character checker (318) may include processors, memory, and instructions to aid the validating engine of the validating system (304) in validating the written data. If the written data is invalid, an alert signal (320) may be triggered. The alert signal (320) may alert the memory controller (302) that the written data is invalid. As will be described in Figs. 4 and 5, the system (300) may include other components to aid the validating system in determining if the written data in the storage array (316) is valid.

[0058] Fig. 4 is a diagram of a system for validating data in a storage array via comparing CRC characters, according to one example of principles described herein. As mentioned above, a validating system is in communication with a memory controller and a storage element. As will be described below, elements of the memory controller and the storage element may aid the validating system in validating written data in the storage array via CRC characters.

[0059] As illustrated in Fig. 4, the storage element (406) includes a CRC character checker (422). The CRC character checker (422) may include processors, memory, and instructions to aid the transferring engine of the validating system (404) to determine if the data transferred to the storage element (406) is valid as described above.

[0060] Further, the storage element (406) includes a CRC character comparer (418). The CRC character comparer (418) may include processors, memory, and instructions to aid the validating engine of the validating system (404) to validate the written data. The CRC character comparer (418) may validate the written data by obtaining a recalculated CRC character from the CRC character checker (422) and obtain the written CRC character via the sense amplifier of the storage array (416). Further, the CRC character comparer (418) may validate the written data by comparing the recalculated CRC character with the written CRC character to create a CRC comparison and determining, based on the CRC comparison, if the written data is valid as

described above. If the written data is invalid, an alert signal (420) may be triggered.

[0061] Fig. 5 is a diagram of a system for validating data in a storage array via comparing data, according to one example of principles described herein. As mentioned above, a validating system is in communication with a memory controller and a storage element. As will be described below, elements of the memory controller and the storage element may aid the validating system in validating written data in the storage array via the data.

[0062] As illustrated in Fig. 5, the storage element (506) includes a CRC character checker (522). The CRC character checker (522) may include processors, memory, and instructions to aid the transferring engine of the validating system (504) to determine if the data transferred to the storage element (506) is valid.

[0063] Further, the storage element (506) includes a data comparer (518). The data comparer (518) may include processors, memory, and instructions to aid the validating engine of the validating system (504) to validate the written data. The data comparer (518) may validate the written data by obtaining the data from the CRC character checker (522) and obtain the written data via the sense amplifier of the storage array (516). Further, the data comparer (518) may validate the written data by comparing the data with the written data to create a data comparison and determining, based on the data comparison, if the written data is valid as described above. If the written data is invalid, an alert signal (520) may be triggered.

[0064] Fig. 6 is a diagram of a system for validating data in a storage array, according to one example of principles described herein. As will be described below, a sense amplifier may be used to validate data written to a storage array after a write command has executed.

[0065] As illustrated, the system (600) includes a bus (602). The bus (602) may be a bidirectional bus. As a result, the bus (602) allows data to be transferred between the memory controller and a storage array (612).

[0066] The system (600) further includes a FIFO queue (604). The FIFO queue (604) receives write commands, data, and a CRC character from a CRC character generator (606) as described above.

[0067] As illustrated, the system (600) includes a CRC character comparer (608) and a storage array (612) that function as described above. The system (600) further includes a multiplexer (608). The multiplexer (608) is used to allow a validating system to select information from the CRC character comparer (608) or select information from the FIFO queue (604).

[0068] The system (600) further includes a sense amplifier (610). The sense amplifier (610) may be utilized to read written data from the storage array (612). As mentioned above, the sense amplifier (610) is used to validate the written data in the storage array (612).

[0069] Fig. 7 is a flowchart of a method for validating data in a storage array, according to one example of principles described herein. The method (700) may be executed by the system (100) of Fig. 1. The method (700) may be executed by other systems such as system 200, system 300, system 400, system 500, system 600, system 900, or system 1000. In this example, the method (700) includes receiving (701), from a memory controller, a write command to write data to a storage array of a storage element, transferring (702) the data with a CRC character to the storage element to create a transfer, writing (703), based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character, and validating (704), via a sense amplifier, the written data.

[0070] As mentioned above, the method (700) includes receiving (701), from a memory controller, a write command to write data to a storage array of a storage element. The write command may be sent to a FIFO queue. As a result, each of the write commands is executed in the order that they are received. Further, the write command may specify a storage array to write the data to.

[0071] As mentioned above, the method (700) includes transferring (702) the data with a CRC character to the storage element to create a transfer. The method (700) may validate the data in the transfer to determine if the data

if valid. The method (700) may validate the data via CRC characters or the method (700) may validate the data the data directly.

[0072] As mentioned above, the method (700) includes writing (703), based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character. If the data is valid when transferred from the memory controller to the storage element, the data is written to a storage array of the storage element. If the data is invalid when transferred from the memory controller to the storage element, the data is not written to the storage array of the storage element. If the data is invalid, the data may be retransmitted by the memory controller to the storage element.

[0073] As mentioned above, the method (700) includes validating (704), via a sense amplifier, the written data. As mentioned above, the data may be validated via CRC characters or by the data directly.

[0074] Fig. 8 is a flowchart of a method for validating data in a storage array, according to one example of principles described herein. The method (800) may be executed by the system (100) of Fig. 1. The method (800) may be executed by other systems such as system 200, system 300, system 400, system 500, system 600, system 900, or system 1000. In this example, the method (800) includes receiving (801), from a memory controller, a write command to write data to a storage array of a storage element, determining (802) a CRC character for the data, adding (803) the CRC character to the data, transferring (804) the data with a CRC character to the storage element to create a transfer, writing (805), based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character, and validating (806), via a sense amplifier, the written data.

[0075] As mentioned above, the method (800) includes determining (802) a CRC character for the data. In some examples, the method (800) may determine the size of the CRC character. The size of the CRC character may be eight bits. However, the size of the CRC character may vary based on the write command, the size of the data, or combinations thereof.

[0076] As mentioned above, the method (800) includes adding (803) the CRC character to the data. The CRC character may be added, via concatenation, to the data before the data and the CRC character are transferred from the memory controller to the storage element. As a result, the CRC character before the transfer and the CRC character after the transfer may be compared to determine if any errors were introduced to the data during the transfer.

[0077] Fig. 9 is a diagram of a validating system, according to one example of principles described herein. The validating system (900) includes an adding engine (914-1), a transferring engine (914-2), a writing engine (914-3), and a validating engine (914-4). The engines (914) refer to a combination of hardware and program instructions to perform a designated function. Alternatively, the engines (914) may be implemented in the form of electronic circuitry (e.g., hardware). Each of the engines (914) may include a processor and memory. Alternatively, one processor may execute the designated function of each of the engines (914). The program instructions are stored in the memory and cause the processor to execute the designated function of the engine.

[0078] The adding engine (914-1) adds the CRC character to the data. The adding engine (914-1) adds one CRC character to the data. The adding engine (914-1) may add several CRC characters to the data.

[0079] The transferring engine (914-2) transfers the data with the CRC character to the storage element to create a transfer. The transferring engine (914-2) may validate the transfer via the data or the CRC character as described above.

[0080] The writing engine (914-3) writes, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character. If the transfer is valid, the writing engine (914-3) writes the data. If the data is invalid, the writing engine (914-3) does not write the data.

[0081] The validating engine (914-4) validates, via a sense amplifier, the written data. The validating engine (914-4) validates the written data based on CRC characters or the data.

[0082] Fig. 10 is a diagram of a validating system, according to one example of principles described herein. In this example, validating system (1000) includes resource(s) (1002) that are in communication with a machine-readable storage medium (1004). Resource(s) (602) may include one processor. In another example, the resource(s) (602) may further include at least one processor and other resources used to process instructions. The machine-readable storage medium (1004) represent generally any memory capable of storing data such as instructions or data structures used by the validating system (1000). The instructions shown stored in the machine-readable storage medium (1004) include determining instructions (1006), transferring instructions (1008), writing instructions (1010), and validating instructions (1012).

[0083] The machine-readable storage medium (1004) contains computer readable program code to cause tasks to be executed by the resource(s) (1002). The machine-readable storage medium (1004) may be tangible and/or physical storage medium. The machine-readable storage medium (1004) may be any appropriate storage medium that is not a transmission storage medium. A non-exhaustive list of machine-readable storage medium types includes non-volatile memory, volatile memory, random access memory, write only memory, flash memory, electrically erasable program read only memory, or types of memory, or combinations thereof.

[0084] The determining instructions (1006) represents instructions that, when executed, cause the resource(s) (1002) to determine a CRC character for the data. The transferring instructions (1008) represents instructions that, when executed, cause the resource(s) (1002) to transfer the data with the CRC character to the storage element to create a transfer.

[0085] The writing instructions (1010) represents instructions that, when executed, cause the resource(s) (1002) to write, based on the transfer, the data with the CRC character to the storage array of the storage element to

create written data and a written CRC character. The validating instructions (1012) represents instructions that, when executed, cause the resource(s) (1002) to validate, via a sense amplifier, the written data.

[0086] Further, the machine-readable storage medium (1004) may be part of an installation package. In response to installing the installation package, the instructions of the machine-readable storage medium (1004) may be downloaded from the installation package's source, such as a portable medium, a server, a remote network location, another location, or combinations thereof. Portable memory media that are compatible with the principles described herein include DVDs, CDs, flash memory, portable disks, magnetic disks, optical disks, other forms of portable memory, or combinations thereof. In other examples, the program instructions are already installed. Here, the memory resources can include integrated memory such as a hard drive, a solid state hard drive, or the like.

[0087] In some examples, the resource(s) (1002) and the machine-readable storage medium (1004) are located within the same physical component, such as a server, or a network component. The machine-readable storage medium (1004) may be part of the physical component's main memory, caches, registers, non-volatile memory, or elsewhere in the physical component's memory hierarchy. Alternatively, the machine-readable storage medium (1004) may be in communication with the resource(s) (1002) over a network. Further, the data structures, such as the libraries, may be accessed from a remote location over a network connection while the programmed instructions are located locally. Thus, the validating system (1000) may be implemented on a user device, on a server, on a collection of servers, or combinations thereof.

[0088] The validating system (1000) of Fig. 10 may be part of a general purpose computer. However, in alternative examples, the validating system (1000) is part of an application specific integrated circuit.

[0089] The preceding description has been presented to illustrate and describe examples of the principles described. This description is not intended to be exhaustive or to limit these principles to any precise form disclosed. Many modifications and variations are possible in light of the above teaching.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for validating data in a storage array, the method comprising:

receiving, from a memory controller, a write command to write data to a storage array of a storage element;

transferring the data with a cyclic redundancy check (CRC) character to the storage element to create a transfer;

writing, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character; and

validating, via a sense amplifier, the written data.

2. The method of claim 1, wherein transferring the data with the CRC character to the storage element to create the transfer comprises:

obtaining the CRC character associated with the data before the transfer;

recalculating the CRC character associated with the data after the transfer to create a recalculated CRC character; and

comparing the CRC character with the recalculated CRC character to determine if an error has occurred while transferring the data.

3. The method of claim 1, wherein validating, via the sense amplifier, the written data comprises:

obtaining a recalculated CRC character;

obtaining the written CRC character via the sense amplifier;

comparing the recalculated CRC character with the written CRC character to create a CRC comparison; and

determining, based on the CRC comparison, if the written data is valid.

4. The method of claim 1, wherein validating, via the sense amplifier, the written data comprises:

obtaining the data;

obtaining the written data via the sense amplifier;

comparing the data with the written data to create a data comparison; and

determining, based on the data comparison, if the data is valid.

5. The method of claim 1, further comprising determining the CRC character for the data.

6. The method of claim 1, further comprising adding the CRC character to the data.

7. A system for validating data in a storage array, the system comprising:

an adding engine to add a CRC character to data;

a transferring engine to transfer the data with the CRC character to a storage element to create a transfer;

a writing engine to write, based on the transfer, the data with the CRC character to a storage array of the storage element to create written data and a written CRC character; and

a validating engine to validate, via a sense amplifier, the written data.

8. The system of claim 7, wherein the transferring engine transfers the data with the CRC character to the storage element to create the transfer by:

obtaining the CRC character associated with the data before the transfer;

recalculating the CRC character associated with the data after the transfer to create a recalculated CRC character; and

comparing the CRC character with the recalculated CRC character to determine if an error has occurred while transferring the data.

9. The system of claim 7, wherein the validating engine validates, via the sense amplifier, the written data by:

obtaining a recalculated CRC character;

obtaining the written CRC character via the sense amplifier;

comparing the recalculated CRC character with the written CRC character to create a CRC comparison; and

determining, based on the CRC comparison, if the written data is valid.

10. The system of claim 7, wherein the validating engine validates, via the sense amplifier, the written data by:

obtaining the data;

obtaining the written data via the sense amplifier;

comparing the data with the written data to create a data comparison; and

determining, based on the data comparison, if the data is valid.

11. A machine-readable storage medium encoded with instructions for validating data in a storage array, the instructions executable by a processor of a system to cause the system to:

determine a cyclic redundancy check (CRC) character for data;

transfer the data with the CRC character to a storage element to create a transfer;

write, based on the transfer, the data with the CRC character to a storage array of the storage element to create written data and a written CRC character; and

validate, via a sense amplifier, the written data.

12. The product of claim 11, further comprising instructions that, when executed, cause the processor to:

receive, from a memory controller, a write command to write the data to the storage array of the storage element; and

add the CRC character to the data.

13. The product of claim 11, further comprising instructions that, when executed, cause the processor to:

obtain the CRC character associated with the data before the transfer;

recalculate the CRC character associated with the data after the transfer to create a recalculated CRC character; and

compare the CRC character with the recalculated CRC character to determine if an error has occurred while transferring the data.

14. The product of claim 11, further comprising instructions that, when executed, cause the processor to:

obtain a recalculated CRC character;

obtain the written CRC character via the sense amplifier;

compare the recalculated CRC character with the written CRC character to create a CRC comparison; and

determine, based on the CRC comparison, if the written data is valid.

15. The product of claim 11, further comprising instructions that, when executed, cause the processor to:

obtain the data;

obtain the written data via the sense amplifier;

compare the data with the written data to create a data comparison; and

determine, based on the data comparison, if the data is valid.

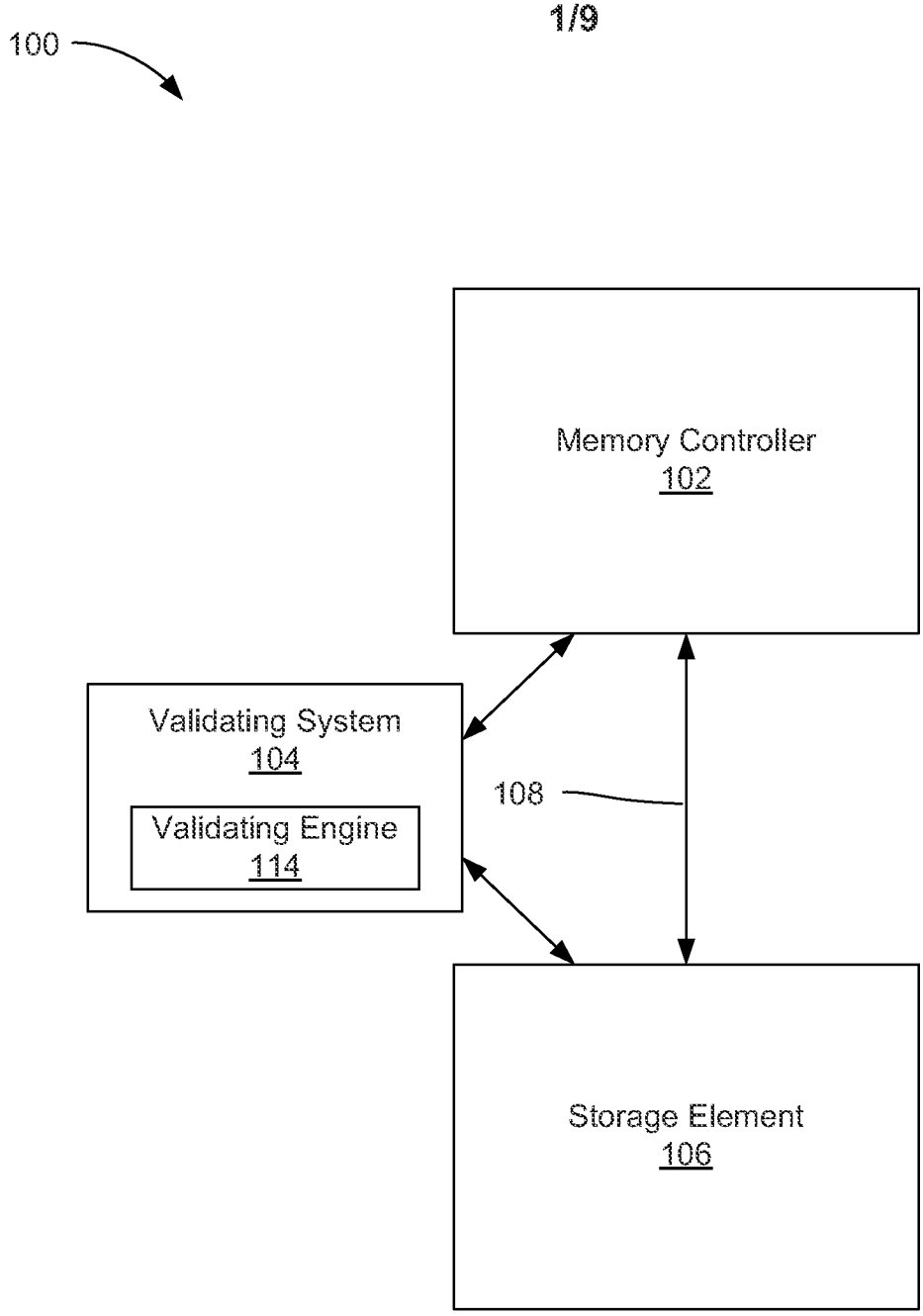
100
Memory Controller
102
Validating System
104
Validating Engine
114
108
Storage Element
106

Fig. 1

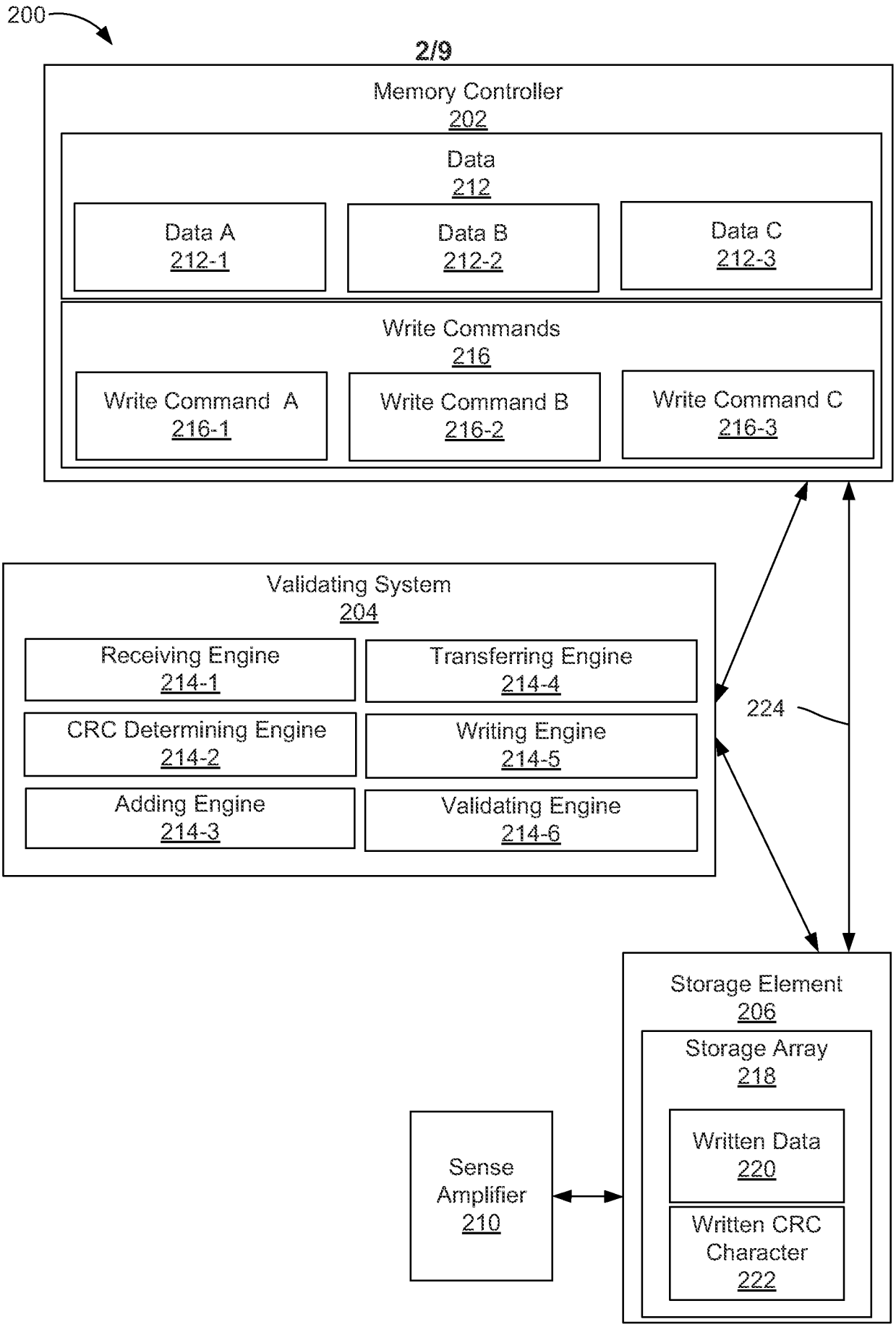
200
Memory Controller
202
Data
212
Data A
212-1
Data B
212-2
Data C
212-3
Write Commands
216
Write Command A
216-1
Write Command B
216-2
Write Command C
216-3
Validating System
204
Receiving Engine
214-1
Transferring Engine
214-4
CRC Determining Engine
214-2
Writing Engine
214-5
Adding Engine
214-3
Validating Engine
214-6
224
Storage Element
206
Storage Array
218
Written Data
220
Written CRC Character
222
Sense Amplifier
210

Fig. 2

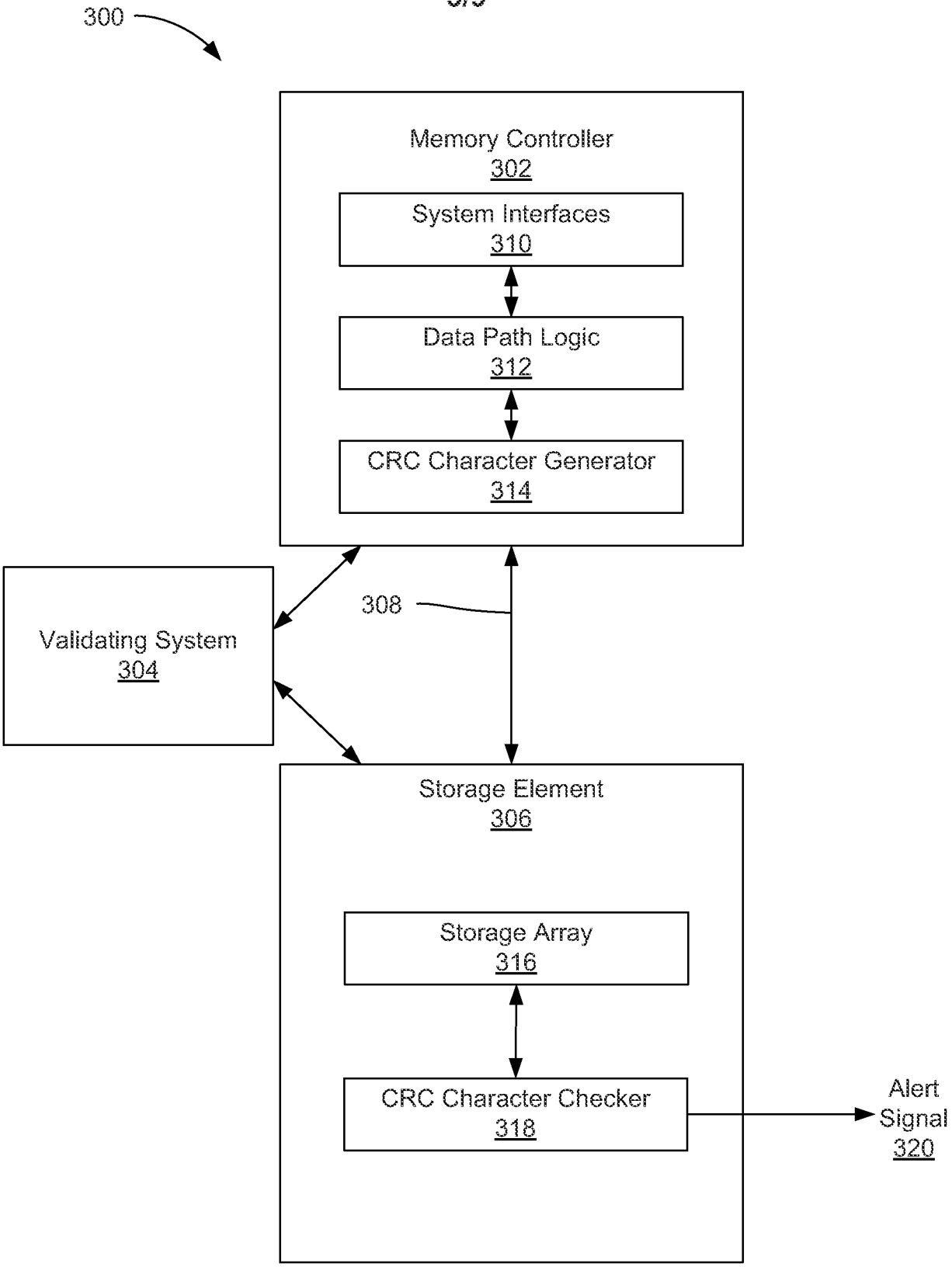
300
Memory Controller
302
System Interfaces
310
Data Path Logic
312
CRC Character Generator
314
308
Validating System
304
Storage Element
306
Storage Array
316
CRC Character Checker
318
Alert
Signal
320

Fig. 3

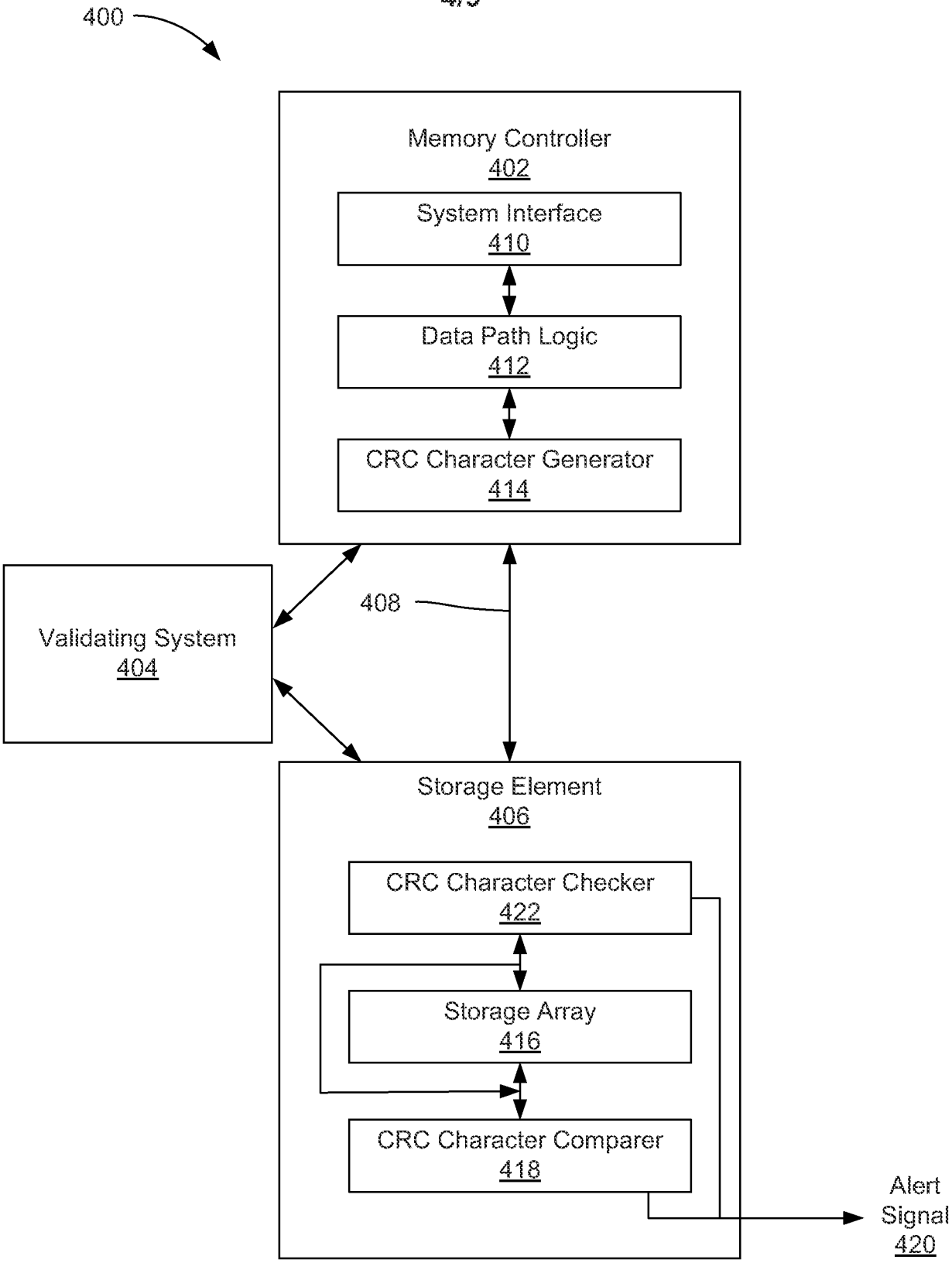
400
Memory Controller
402
System Interface
410
Data Path Logic
412
CRC Character Generator
414
408
Validating System
404
Storage Element
406
CRC Character Checker
422
Storage Array
416
CRC Character Comparer
418
Alert
Signal
420

Fig. 4

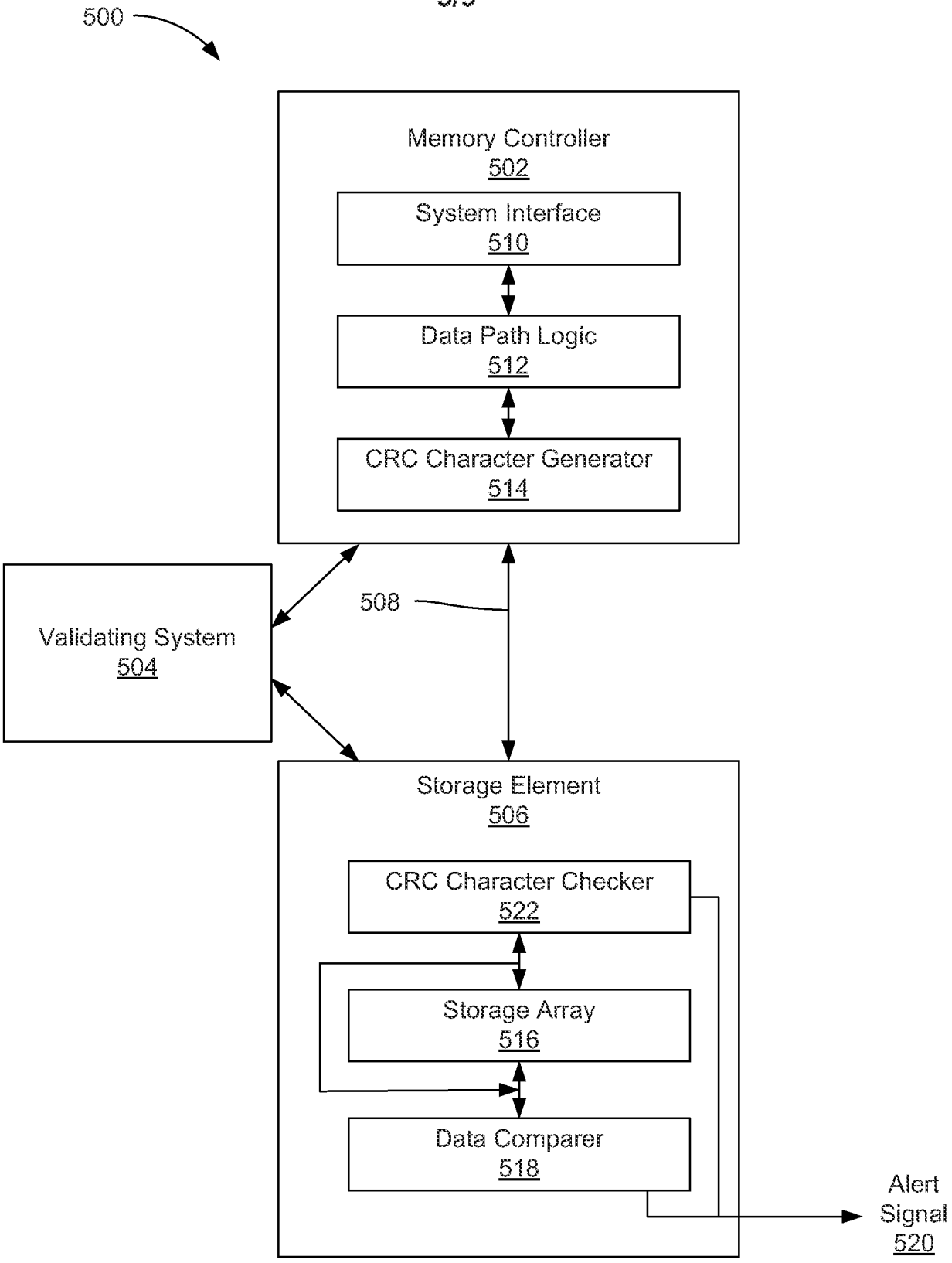
500
Memory Controller
502
System Interface
510
Data Path Logic
512
CRC Character Generator
514
508
Validating System
504
Storage Element
506
CRC Character Checker
522
Storage Array
516
Data Comparer
518
Alert
Signal
520

Fig. 5

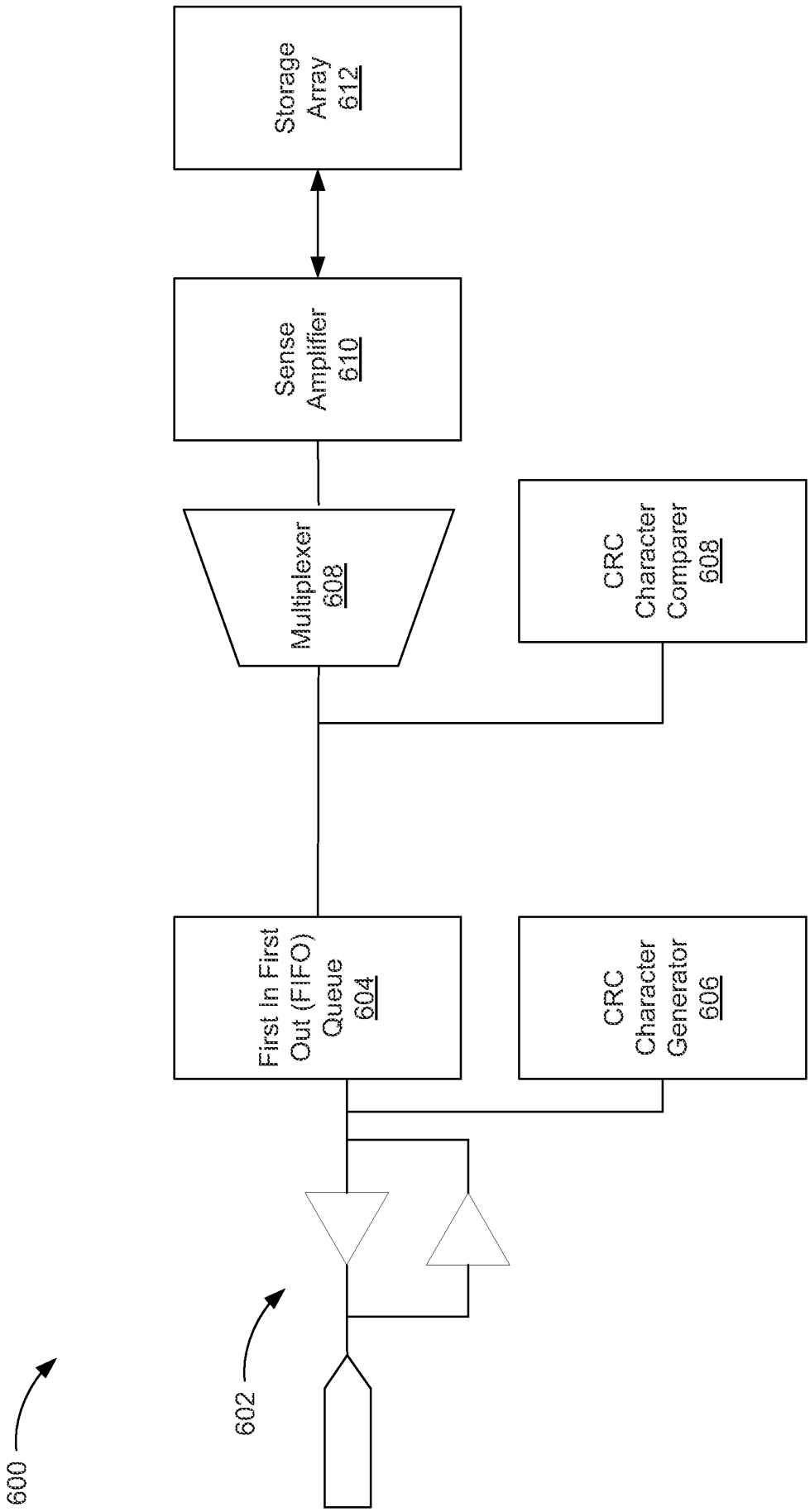
600
602
First In First Out (FIFO) Queue 604
CRC Character Generator 606
Multiplexer 608
CRC Character Comparer 608
Sense Amplifier 610
Storage Array 612

Fig. 6

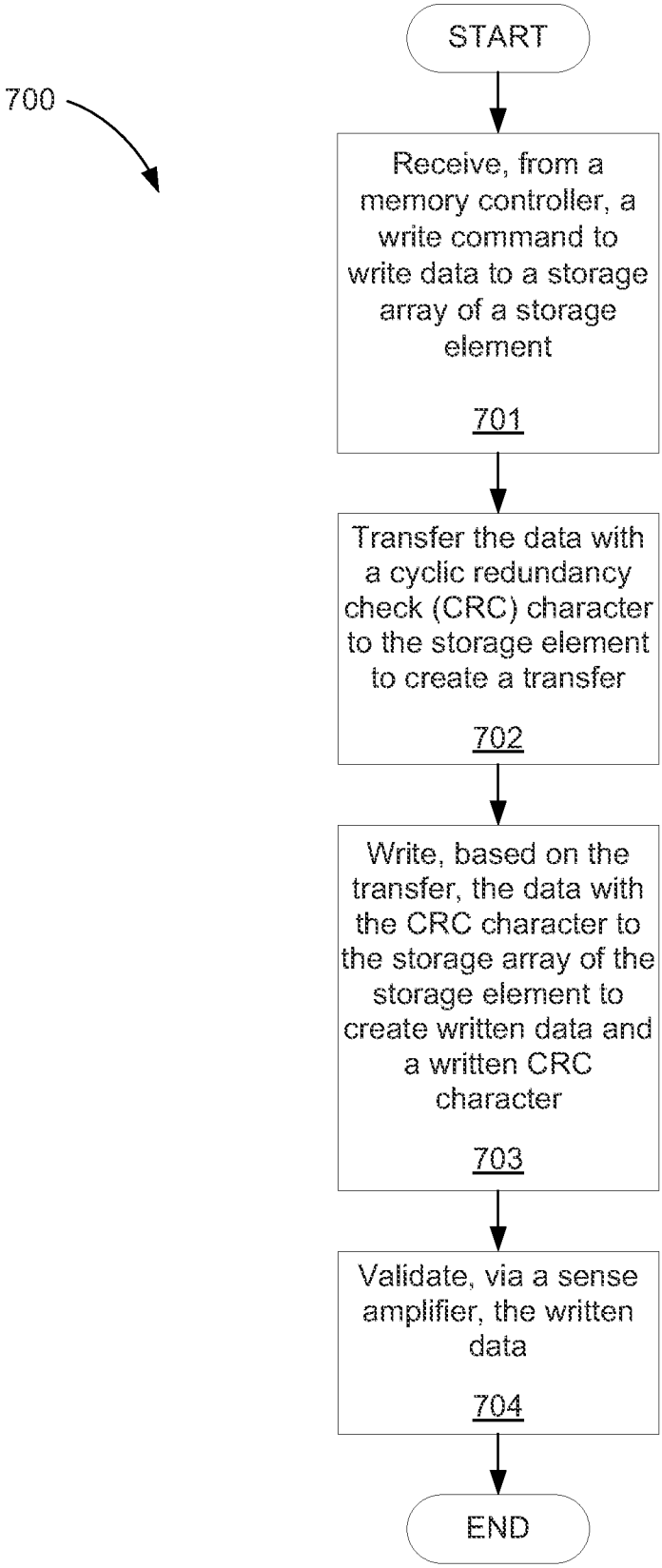
700
START
Receive, from a memory controller, a write command to write data to a storage array of a storage element
701
Transfer the data with a cyclic redundancy check (CRC) character to the storage element to create a transfer
702
Write, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character
703
Validate, via a sense amplifier, the written data
704
END

Fig. 7

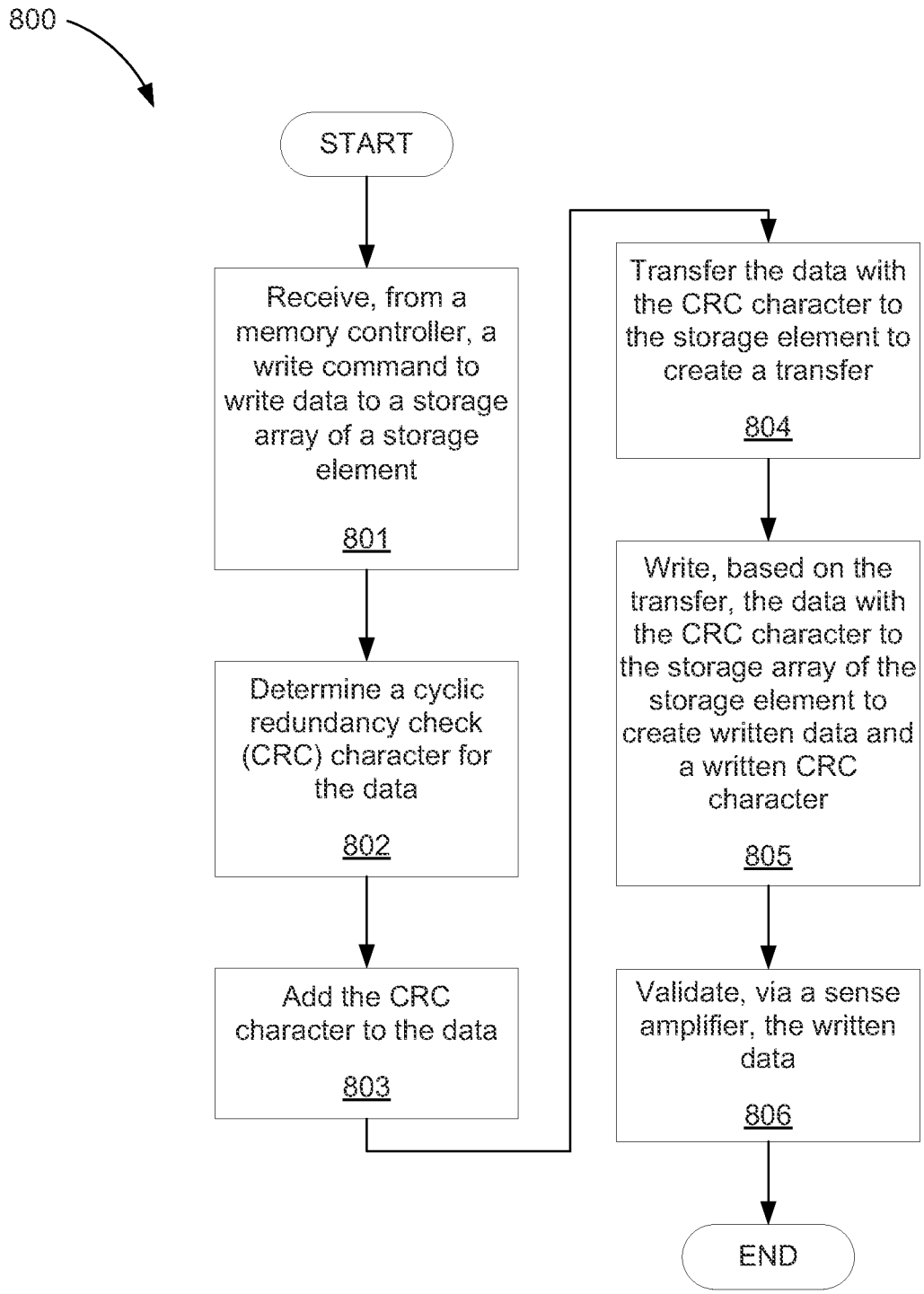
800
START
Receive, from a memory controller, a write command to write data to a storage array of a storage element
801
Determine a cyclic redundancy check (CRC) character for the data
802
Add the CRC character to the data
803
Transfer the data with the CRC character to the storage element to create a transfer
804
Write, based on the transfer, the data with the CRC character to the storage array of the storage element to create written data and a written CRC character
805
Validate, via a sense amplifier, the written data
806
END

Fig. 8

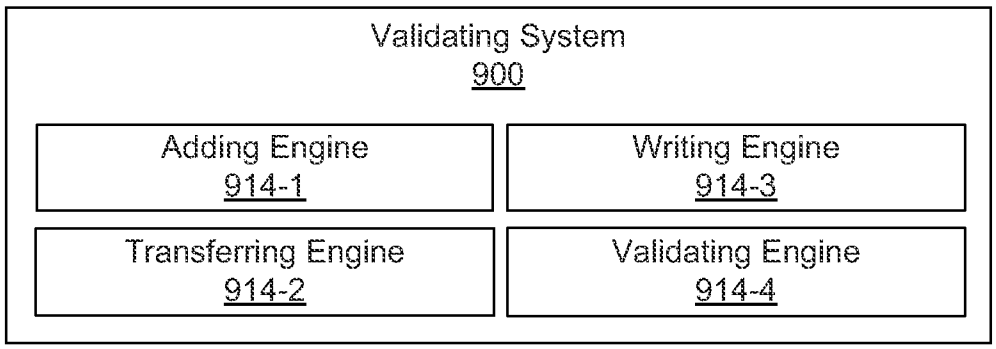
Validating System
900
Adding Engine
914-1
Writing Engine
914-3
Transferring Engine
914-2
Validating Engine
914-4

Fig. 9

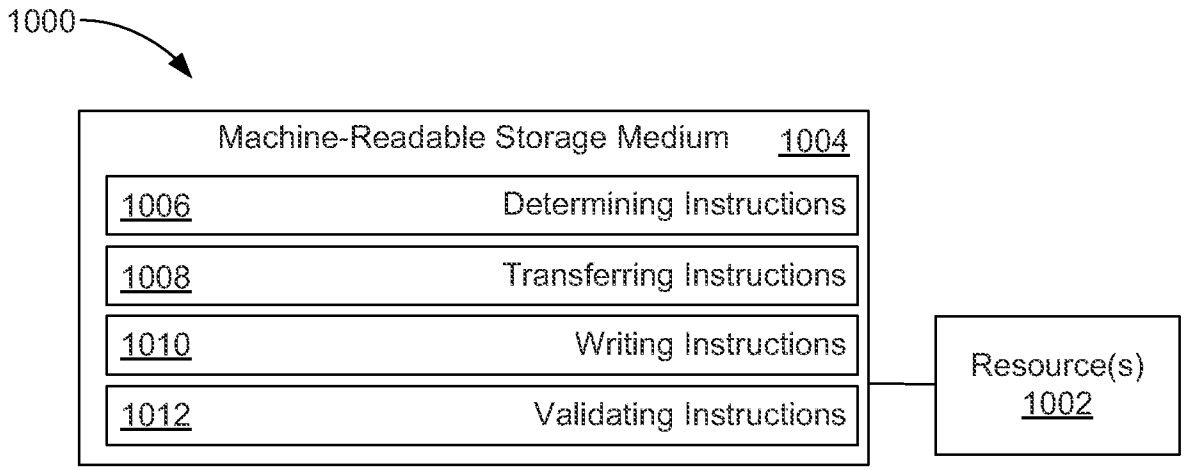
1000
Machine-Readable Storage Medium
1004
1006
Determining Instructions
1008
Transferring Instructions
1010
Writing Instructions
1012
Validating Instructions
Resource(s)
1002

Fig. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/013954

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; H03M 13/00; G11C 7/00; H03M 13/05; G06F 11/10; G11C 29/00; G06F 11/00; G11B 20/18

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: validate, data, storage, array, transfer, CRC, write, compare, calculate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6901551 B1 (BRIAN E. CORRIGAN, III) 31 May 2005 See column 3, lines 15-17; claims 1, 5; and figure 1.	1-15
Y	US 6968478 B1 (ERIC E. EDWARDS et al.) 22 November 2005 See column 1, lines 61-62; column 2, lines 9-11; claims 1, 22; and figures 1A, 4A.	1-15
A	US 6182266 B1 (SHERWIN J. CLUTTER et al.) 30 January 2001 See column 6, lines 1-16; and figure 8.	1-15
A	US 2005-0081085 A1 (ROBERT M. ELLIS et al.) 14 April 2005 See paragraphs [0013], [0019], [0026]; and figures 1-2.	1-15
A	WO 01-001581 A1 (EMC CORPORATION) 04 January 2001 See page 5, lines 12-22; page 6, lines 1-11; 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	Date of mailing of the international search report
21 October 2015 (21.10.2015)	**26 October 2015 (26.10.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-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

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2015/013954

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6901551 B1	31/05/2005	US 2003-0115417 A1	19/06/2003
		US 6760814 B2	06/07/2004
		US 6772289 B1	03/08/2004
US 6968478 B1	22/11/2005	None	
US 6182266 B1	30/01/2001	None	
US 2005-0081085 A1	14/04/2005	US 7386765 B2	10/06/2008
WO 01-01581 A1	04/01/2001	DE 60032239 T2	31/05/2007
		EP 1198891 A1	24/04/2002
		EP 1198891 B1	06/12/2006
		EP 1758253 A1	28/02/2007
		US 6915475 B1	05/07/2005
		WO 01-01581 A9	25/07/2002