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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 October 2017 (05.10.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/172060 A1

(51) International Patent Classification:
*G06F 11/10* (2006.01)

(21) International Application Number:
PCT/US2017/017236

(22) International Filing Date:
9 February 2017 (09.02.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/089,316 1 April 2016 (01.04.2016) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **HALBERT, John B.**; 8784 SW Marseilles Drive, Beaverton, Oregon 97007 (US). **BAINS, Kuljit S.**; 9146 52nd Lane NE, Olympia, Washington 98516 (US).

(74) Agents: **MALLIE, Michael J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, California 94085 (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, DJ, 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, KH, KN, KP, KR, KW, 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).

**Published**:

— *with international search report (Art. 21(3))*

(54) Title: VALIDATION OF MEMORY ON-DIE ERROR CORRECTION CODE

(57) Abstract: Embodiments are generally directed to validation of memory on-die error correction code. An embodiment of a memory device includes one or more memory arrays for the storage of data; control logic to control operation of the memory device; and ECC (error correction code) logic, including ECC correction logic to correct data and ECC generation logic to generate ECC code bits and store the ECC bits in the one or more memory arrays. In a validation mode to validate operation of the ECC logic, the control logic is to allow generation of ECC code bits for a first test value and disable generation of ECC code bits for a second test value.

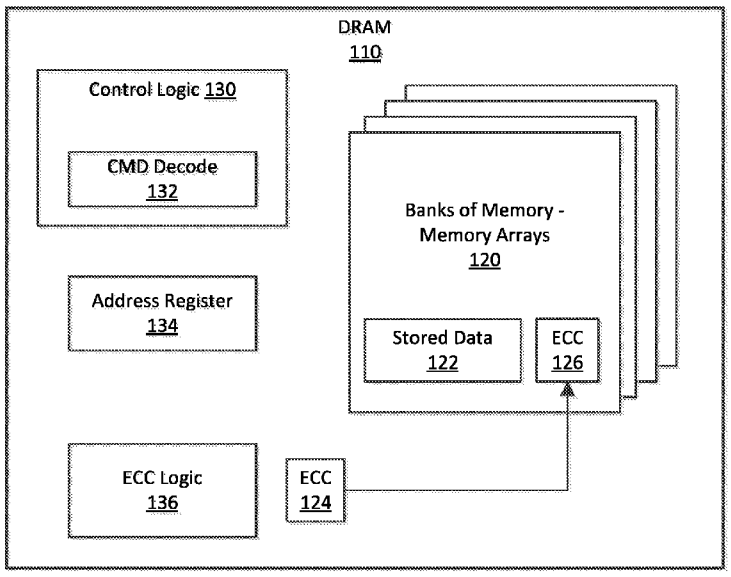


Fig. 1

WO 2017/172060 A1

# VALIDATION OF MEMORY ON-DIE ERROR CORRECTION CODE

## TECHNICAL FIELD

Embodiments described herein generally relate to the field of electronic devices and, more particularly, to validation of memory on-die error correction code.

## BACKGROUND

For computer memory, error correction is an essential component as certain error will occur in operation. As the speed and density of memory increases, the need for effective error correction also increases.

In modern computer memory, the error correction includes on-die ECC (error correction code) logic to correct certain errors, such as single bit errors.

## BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments described here are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

**Figure 1** is an illustration of a memory apparatus including validation of memory on-die error correction code operation according to an embodiment;

**Figure 2A** is an illustration of a write process for error correction validation according to an embodiment;

**Figure 2B** is an illustration of an example of validation of on-die ECC logic of a memory device according to an embodiment;

**Figure 3** is a flow chart to illustrate a process for validation of memory on-die ECC operation according to an embodiment; and

**Figure 4** is an illustration of an embodiment of an apparatus or system according to an embodiment.

## DETAILED DESCRIPTION

Embodiments described herein are generally directed to validation of memory on-die error correction code.

For the purposes of this description, the following apply:

"Error-correcting code" or "ECC" refers to the process of adding additional data to certain computer data such that the original data can be recovered when a certain number of

errors are introduced into the data. In certain implementations the additional data can be stored separately from the computer data in different memory locations.

"ECC memory" refers to computer memory including on-die ECC logic. ECC memory may include, but is not limited to, SDRAM (Synchronous Dynamic Random Access Memory) such as DDR4 (Double Data Rate, Fourth Generation) memory, referring to JEDEC DDR4 Standard JESD79-4, November 2013, and DDR4E (DDR version 4, extended) memory, referring to the extension of DDR4.

The handling of certain errors by ECC memory, such as double bit errors, may not be defined for a memory technology, and thus there is a need for validation of the ECC operation. The use of on-die ECC prevents certain validation operations. In particular, the direct modification of data without affecting ECC bits is prevented because the ECC bits are not generated until the underlying data has been stored.

In some embodiments, an apparatus, system, or method provides for validation of on-die ECC logic of ECC memory. A computer may include on-die ECC, referring to error correction logic that is located on or inside the computer die. In some embodiments, an apparatus, system, or method provides for blocking generation of ECC code values to allow for modification of stored data without recalculating the ECC code values.

ECC memory includes on-die ECC to correct errors, which may be limited to correction of single bit errors. For such memory, there may be a requirement regarding how the ECC code should handle double bit errors without requiring a particular process. For example, a particular H-Matrix ECC code may not be mandated, with memory suppliers having the responsibility to select a code that behaves as required.

In particular, it may be required that an ECC H-Matrix be selected such that, upon the ECC code attempting to correct a double bit error because of aliasing and generating a triple bit error, the ECC code is to then map the aliased triple bit error in a specific manner. However, the memory would not be aware that the triple bit error has occurred. Specifically, in the case of an aliased triple bit error, if the double bit error occurs in a top half of the code word, the code is constructed such that the miscorrection will occur in the bottom half of the code word. Similarly, if a double bit error occurs in the bottom half of the code word, then the miscorrection will occur in the top half of the code word. The code then provides that the code words are read out such that triple bit errors are not read out to the data pins.

Specifically, for a x16 (16-bit data width of the DRAM chips) device, two code words are read from two banks, with half of the code word from one code word being mapped to DQ[0:7] and the same code word half of the other code word being mapped to DQ[8:15]. By

mapping two code word halves for x16, any double bit errors that might be miscorrected into triple bit errors will not propagate to the DQ pins.

For a x8 device, the top code word half is mapped to one column (N) access and the bottom code word half is mapped to the N+1 column access. With this mapping, any double bit error that occurs in N or N+1 that might be aliased into a triple bit error, the aliased third error should always show up in the other column access (N+1, N). Further, for x4 device has the Code Word into divided into four column accesses (N, N+1, N+2, N+3). The top code word half is mapped to N and N+1, the bottom code word half is mapped to N+2 and N+3.

In this manner, an aliased three-bit error is read out in data as a double bit error, and the third bit of the triple-bit error is not included in such data. Stated in another manner, two bits of the triple bit error are mapped to the data that is to be reported (including a first portion of the data word), and the remaining third bit of the error is mapped to data that is not included is such data (a second portion of the data word), and thus the miscorrected third bit of the error is dropped.

As each memory supplier may develop the supplier's own code and the supplier's own physical implementation, an apparatus, system, or method is needed to validate that, after fabrication (post silicon), the ECC implementation of ECC memory behaves as required.

However, the conventional method of seeding errors from the central processing unit (CPU) or other processor into memory for ECC testing will not work to validate the on-die ECC logic within the DRAM. The reason for this failure is that, with on-die ECC logic in the DRAM, any error that injected into the memory via the conventional method from the CPU will be ignored by the DRAM. With on-die ECC, the check code is created after data has been received by the DRAM data pins, and the injection of an error will thus result in a corrected check code that takes into account the injected error.

It is possible alternatively to provide certain other operational modifications to generate internal errors in ECC memory, but such operations generally will not allow accurate testing of the ECC logic. For example, changing the refresh time of a DRAM may operate to create internal errors. However, such internal errors are essentially random, and thus the process lacks control of which bit or bits in the data will be flipped, which prevents accurate testing of the ECC logic.

In some embodiments, in order to provide accurate testing of operation of on-die ECC, an apparatus, system, or process provides bit by bit control to flip check bits in the array, and thereby generate data with a known error (which is referred to as poisoned data), without the DRAM generating ECC check bits in response to the flipped bit values. In this manner, precise control of how errors are created and corrected or aliased into triple bits can be maintained. In

some embodiments, a controller is applied to affect only regeneration of check bits, thereby resulting in the controlled injection of errors into the memory array. In some embodiments, an apparatus, system, or process allows for validation of on-die ECC logic operation without requiring knowledge of the particular error correction code matrix utilized in the on-die ECC logic.

In memory operation, the ECC check bits for an ECC memory may include sufficient data to correct any single bit error. However, the ECC check bits do not include enough data to correct a two-bit error, and thus two-bit errors should pass without correction. In operation, the CPU is capable of detecting a two-bit error and thus can take appropriate measures to address the error, but the CPU generally cannot detect errors with greater than two errors.

Because of aliasing in which a double bit error will be mistaken for a single-bit error, the ECC logic of a memory will occasionally attempt to correct a double bit (rather than pass through the error), and will then generate a triple bit error. In operation, to ensure that the CPU is presented with a two-bit error that can be detected, two bits of the triple bit error are to be presented in the data value for the CPU, and the third bit of the triple bit error is mapped to a second half of the code word that is not reported in the data value for the CPU.

In some embodiments, in a read cycle for a certain data value, if a single bit has been changed during the previous write cycle (utilizing the blocked check bit write for a poisoned data value with a single flipped bit), the on-die ECC logic (if operating properly) corrects the data before returning the data to the CPU. The ECC check bits include sufficient data to correct any single bit error. In some embodiments, a validating unit (which may be, for example, the CPU, another processor, or a separate testing unit) may determine proper operation of the on-die ECC logic by inspection of the data that is written to and read from the ECC memory.

In some embodiments, if two bits are changed (utilizing the blocked check bit write for a data value with two flipped bits), the on-die ECC logic can (in certain cases) detect and report the two-bit error. The on-die ECC can (in certain cases) mistakenly miscorrect the double bits into triple bits, the on-die ECC then reporting two of the error bits of the three-bit error as a two-bit error in a data value that is reported, with a remaining single bit error not being included in the data value. Without validation of such operation, a system is at risk of silent data corruption that is not detected by the CPU.

In some embodiments, an apparatus, system, or method operates to modify the DRAM array values without modifying the ECC data stored in the array during a write cycle, and to utilize this operation to change bits from the previous data value in a known way. In some embodiments, the DRAM array values are modified by allowing the generation of check

data values for a first data value and blocking the generation of a new check data value for a second data value, resulting in a controlled modification of the data (as the new second data value may be stored with specific bits flipped from the previous stored first data value) without resulting in new check data values (because the generation of the ECC data is disabled, and thus the ECC data for the previous stored data remains stored).

In an example, it may be assumed that a first data value (Data-1) is initially stored in a memory, and second data value (Data-2) is next stored in the memory. In some embodiments, a validation process may include modifying a certain number of bits of Data-1 to generate Data-2. For, example a particular bit (bit[j]) may be flipped to generate a one-bit error for testing, and two particular bits (bit[j] and bit [k]) may be flipped to generate a two-bit error for testing, where bit[j] and bit[k] may be any bits of the stored data.

In some embodiments, if a single bit is changed in the stored data during a write cycle with a blocked check bit write (generation of ECC code disabled), then in a subsequent read cycle the on-die ECC logic should correct the data before returning the data to the CPU, and thus the validating unit can determine proper data correction operation by the on-die ECC. If two bits are changed during the blocked check bit write, the on-die ECC logic should (in some cases) mistakenly correct the double bits into triple bits, and on a subsequent read cycle the on-die ECC should pass a double-bit error in the reported data, with the third error bit not being included in the data value, the third error bit being mapped to a second half of a data word that is not reported in such data value. It is known that certain patterns will generate triple bit errors, but, if the on-die ECC is operating correctly, a controller should not see any triple bit errors in the reported data resulting from the test data. In this manner, in a validation mode in which a known double bit error is injected into a memory, this case of miscorrected triple bit errors can then be validated by the validating unit (such as the CPU, another processor, or a separate testing unit). If the on-die ECC operation is validated in the validation mode, then the H-Matrix ECC code for the on-die ECC may be determined to be acceptable.

In some embodiments, an apparatus, system, or process is to:

(1) Write a first data (Data-1) to a DRAM, the DRAM including on-die ECC logic.

(2) Calculate an ECC code for Data-1 using the on-die ECC logic and store the value in memory. The calculated ECC code (ECC-1 being the ECC code generated for Data-1) will be utilized in verification of the on-die ECC operation.

(3) Turn off (disable) calculation of ECC check bits to block the calculation of an ECC code for a following data.

(4) Write a second data (Data-2) to the DRAM, wherein Data-1 has been modified in a known way to generate Data-2, such as to flip a single bit of Data-1 (e.g., a bit[j]) to create a single bit error or to flip two bits of Data-1 (e.g., a bit[j] and a bit[k]) to create a double bit error. In this manner, a known error is injected into the memory.

The disabling of the calculation of ECC check bits for Data-2 results in leaving the existing check bits of ECC-1 (which were generated for Data-1) in place in memory, and controlling the flipping of certain bits into memory to inject errors for testing. Stated in another way, ECC-1 becomes the error code for Data-2 (or, if ECC-2 is the error code for Data-2, then ECC-2 = ECC-1).

In some embodiments, in this manner, a known error is introduced into a memory, the error being the bits of Data-1 that are flipped to generate Data-2, with the flipped bits being established as an error by blocking the generation of the ECC code for Data-2 such that the prior ECC code (ECC-1 for Data-1) is the error code for Data-2. In some embodiments, an apparatus, system, or process utilizes the known error to evaluate the operation of the on-die ECC of the memory, and specifically the handling of single bit errors and double bit errors.

**Figure 1** is an illustration of a memory apparatus including validation of memory on-die error correction code operation according to an embodiment. In some embodiments, a DRAM memory 110, such as an SDRAM, includes banks of memory including memory arrays 120 for the storage of data values. In some embodiments, the DRAM includes control logic 130 with command decode, and an address register 134 for the receipt of memory addresses. In some embodiments, the DRAM further includes on-die ECC logic 136 to provide error correction operation. In some embodiments, the ECC logic includes ECC correction logic and ECC generation logic.

In operation, the ECC logic H-matrix will generally only correct a single bit error. For double bit error there is a certain chance (such as a 20% chance) that the system will attempt to correct the double bit error and will miscalculate to create a triple bit error. However, the DRAM 110 may be required to only present a double bit error in a read operation, and thus the memory device should report two bits of the error in a data value, with the third error bit not included in the reported data value.

However, if in accordance with a method for testing of such operation there is an attempt to poison (i.e., change) the stored value by a certain number of bits and submit the poisoned bits to DRAM, the on-die ECC will correct the received bits internally. For this reason, it is not generally possible in ECC memory to directly poison the stored data bits for validation testing by modifying the stored data.

In some embodiments, the ECC logic 136 is to calculate an ECC code 124 for stored data 126, and the ECC logic 136 writes the ECC code 124 to the memory as ECC 126 in a normal operation mode. In some embodiments, in a validation mode, the control logic 130 is operable to change one or more bits of the stored data in a new write operation, and block the generation and writing of new ECC bits for the new write of data. Accordingly, the value of certain bits of the stored data may be flipped without changing the ECC code because the generation and writing a new ECC or replacement value 126 is blocked, and the previous value of such ECC bits is retained. Stated in another way, the blocking of the writing of ECC bits allows for indirectly poisoning the stored data without modification of the previously calculated ECC bits in a validation operation.

**Figure 2A** is an illustration of a write process for error correction validation according to an embodiment. As illustrated in Figure 2A, a validation operation utilizes the operation of a DRAM array 210, on-die ECC correction logic 220, on-die ECC generation logic 230, and control logic 240. (In this example, an x4 memory device, 4-bit data width of the DRAM chips, is provided, but the memory may also be, for example, an x8 memory device in a similar process, or an x16 memory without requiring the prefetch of data.)

In an example write cycle, the on-die ECC correction logic reads 128 data bits from the DRAM array 210 along with 8 check bits. In parallel the CPU 250 sends 32 data bits (write data) to the DRAM to be written. The 128 data bits are first checked for ECC errors, with any single bit errors being corrected in the 128 data bits. The 32 data bits from the CPU 250 are merged into the 128 corrected data bits, and the new ECC check bits (ECC 8b) are generated and the 32 data bits from the CPU along with the newly generated check bits are written to the DRAM array 210.

On a read cycle, the 128 data bits and the 8 check bits are read from the DRAM array 210, the ECC will correct any single bit errors in the 128 data bits, and the 32 bits for the read cycle are the sent to the DQ (data) pins (which are not illustrated in Figure 2).

In some embodiments, in order to indirectly inject errors into the DRAM array during a write cycle, the generation of error correction code bits by the ECC generation logic 230 is blocked. In some embodiments, error correction code bits are generated but are not stored in DRAM array 210, thus retaining the previous ECC value. In this way, new data bits are written to the DRAM array without changing the corresponding check bits, thus allowing for errors to be indirectly injected into data stored in the DRAM array 210. In some embodiments, the generation of new ECC bits by the ECC Generation Logic 230 is blocked by command of control logic 240, and as a result previously calculated ECC bits remain in the DRAM array, even though corresponding stored data may change.

**Figure 2B** is an illustration of an example of validation by on-die ECC logic of a memory device according to an embodiment. In a validation mode of a memory 260, a first test data (Data-1) is stored in a memory array of the memory 262. In some embodiments, an on-die ECC logic is to calculate and store a new ECC code (ECC-1) for the Data-1 value 264. Thus, an error cannot be injected directly into the first data value because the ECC codes will be calculated based on the data value stored. In some embodiments, a first data value is instead used to generate a second data value with a known error while retaining the prior ECC value, and thus allow for indirect injection of the error.

In some embodiments, Data-1 is utilized to generate a second data value, Data-2, by flipping a certain number *n* of bits 265, such as changing one bit of Data-1 to generate a single bit error or changing two bits of Data-1 to generate a double bit error.

In some embodiments, the on-die ECC code calculation is disabled 266 to maintain ECC code values. In the illustrated example, Data-2 (which differs from Data-1 by a certain *n* bits) is stored in the memory array 268. In some embodiments, with the ECC code calculation by the on-die logic being disabled, the prior ECC code value (ECC-1) remains in storage 270.

In some embodiments, the on-die ECC performs error checking of Data-2 using ECC-1 272 because the ECC value has not changed. In some embodiments, validation of the on-die ECC logic is performed by a validating unit (such as the CPU, another processor, or a separate testing unit) 274, wherein the validation of the on-die ECC logic is based at least in part on the error-checking of Data-2 using ECC-1.

**Figure 3** is a flow chart to illustrate a process for validation of memory on-die ECC operation according to an embodiment. In some embodiments, a validation process (where validation may be performed by a validating unit, such as a system CPU or other processor or a separate testing unit) 300 includes entering a validation mode 302, wherein the validation mode refers to any mode or operation to provide for validation of operation of the on-die ECC logic of a memory device.

In some embodiments, the process includes writing a first test value 304 into a memory array and generating first ECC code bits 306. In some embodiments, generation of ECC code values is then blocked 308. In some embodiments, the process proceeds with writing a second test value to memory, wherein one or more bits of the first test value are flipped in the second test value 310. In a particular validation operation the number of bits changed are either one or two bits, but embodiments are not limited to the change of any particular number of bits. In some embodiments, because the generation of a second ECC code for the write of the second test value is blocked, thereby preventing the change of the ECC bits in response to the write of

the modified data, the storing of the second test value results in creating a particular known error, such as a single bit error or a double bit error representing the flipping of the one or two bits of the first test value to create the second test value.

In some embodiments, the validation process then proceeds based on whether the introduced error is a single bit error or a double bit error 312. For a single bit error 312, the process proceeds with a validation that the single bit error (i.e., the one bit flipped from the first test value) is corrected by the on-die ECC logic 314. Thus, the memory device should correct the second test value data to match the first test value data.

For a double bit error 312, in normal operation, a double bit error may be recognized as a double bit error or, in certain cases, the double bit error may be miscorrected to generate a triple bit error. In some embodiments, a validation process may proceed with validating that either 316:

The error is properly reported out as a double bit error in a data value 318; or

The error is miscorrected to generate a triple bit error, and the triple bit error is properly reported out as a double bit error in a data value, with a remaining third error bit not included in the reported data value 320. If the on-die ECC is operating correctly, the reported data should not contain triple bit errors resulting from the test data.

The validation process then may continue as appropriate 322. If the on-die ECC operation is validated in the validation process, then the H-Matrix ECC code for the on-die ECC may be determined to be acceptable.

**Figure 4** is an illustration of an embodiment of an apparatus or system according to an embodiment. In this illustration, certain standard and well-known components that are not germane to the present description are not shown. Elements shown as separate elements may be combined, including, for example, an SoC (System on Chip) combining multiple elements on a single chip.

In some embodiments, an apparatus or system 400, such as a computing system, includes processing means such as one or more processors 410 coupled to one or more buses or interconnects, shown in general as bus 465. The processors 410 may comprise one or more physical processors and one or more logical processors. In some embodiments, the processors 410 may include one or more general-purpose processors or special-processor processors. The bus 465 is a communication means for transmission of data. The bus 465 is illustrated as a single bus for simplicity, but may represent multiple different interconnects or buses and the component connections to such interconnects or buses may vary. The bus 465 shown in Figure 4 is an abstraction that represents any one or more separate physical buses, point-to-point connections, or both connected by appropriate bridges, adapters, or controllers.

In some embodiments, the apparatus or system 400 comprises a random access memory (RAM) or other dynamic storage device or element as a main memory 405 for storing information and instructions to be executed by the processors 410. Main memory 410 may include, but is not limited to, dynamic random access memory (DRAM).

In some embodiments, the DRAM 405 includes on-die ECC logic 470, which may include the ECC logic 136 illustrated in Figure 1, and may include the ECC correction logic 220 and ECC generation logic 230 illustrated in Figure 2A. In some embodiments, the DRAM 405 further includes one or more memory arrays 475, wherein the DRAM is to block generation of ECC code bits for the purpose of changing stored data values in the memory array 475 without modifying ECC values, thus allowing for validation of on-die ECC operation.

The apparatus or system 400 also may comprise a non-volatile memory (NVM) 420; a storage device such as a solid state drive (SSD) 425; and a read only memory (ROM) 430 or other static storage device for storing static information and instructions for the processors 410.

Any memory device described herein may include, for example, block addressable memory device, such as NAND or NOR technologies, phase change memory (PCM), three dimensional cross point memory, or other byte addressable nonvolatile memory devices, memory devices that use chalcogenide phase change material (e.g., chalcogenide glass), resistive memory, nanowire memory, ferro-electric transistor random access memory (FeTRAM), flash memory such as solid state disk (SSD) NAND or NOR, multi-threshold level NAND flash memory, NOR flash memory, magnetoresistive random access memory (MRAM) memory that incorporates memristor technology, spin transfer torque (STT)-MRAM, or a combination of any of the above, or other memory.

In some embodiments, the apparatus or system 400 includes one or more transmitters or receivers 440 coupled to the bus 465 to provide wired or wireless communications. In some embodiments, the apparatus or system 400 may include one or more antennae 444, such as dipole or monopole antennae, for the transmission and reception of data via wireless communication using a wireless transmitter, receiver, or both, and one or more ports 442 for the transmission and reception of data via wired communications. Wireless communication includes, but is not limited to, Wi-Fi, Bluetooth™, near field communication, and other wireless communication standards. .

In some embodiments, the apparatus or system 400 may include one or more input devices 450 for the input of data, including hard and soft buttons, a joy stick, a mouse or other pointing device, a keyboard, voice command system, or gesture recognition system.

In some embodiments, the apparatus or system 400 may include an output display 455, where the display 455 may include a liquid crystal display (LCD) or any other display technology, for displaying information or content to a user. In some environments, the display 455 may include a touch-screen that is also utilized as at least a part of an input device 450. The apparatus or system 400 may further include audio output, including one or more speakers, audio output jacks, or other audio, and other output to the user.

The apparatus or system 400 may also comprise a battery or other power source 460, which may include a solar cell, a fuel cell, a charged capacitor, near field inductive coupling, or other system or device for providing or generating power in the apparatus or system 400. The power provided by the power source 460 may be distributed as required to elements of the apparatus or system 400.

In the description above, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the described embodiments. It will be apparent, however, to one skilled in the art that embodiments may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form. There may be intermediate structure between illustrated components. The components described or illustrated herein may have additional inputs or outputs that are not illustrated or described.

Various embodiments may include various processes. These processes may be performed by hardware components or may be embodied in computer program or machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the processes. Alternatively, the processes may be performed by a combination of hardware and software.

Portions of various embodiments may be provided as a computer program product, which may include a computer-readable medium having stored thereon computer program instructions, which may be used to program a computer (or other electronic devices) for execution by one or more processors to perform a process according to certain embodiments. The computer-readable medium may include, but is not limited to, magnetic disks, optical disks, compact disk read-only memory (CD-ROM), and magneto-optical disks, read-only memory (ROM), random access memory (RAM), erasable programmable read-only memory (EPROM), electrically-erasable programmable read-only memory (EEPROM), magnet or optical cards, flash memory, or other type of computer-readable medium suitable for storing electronic instructions. Moreover, embodiments may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting computer.

Many of the methods are described in their most basic form, but processes can be added to or deleted from any of the methods and information can be added or subtracted from any of the described messages without departing from the basic scope of the present embodiments. It will be apparent to those skilled in the art that many further modifications and adaptations can be made. The particular embodiments are not provided to limit the concept but to illustrate it. The scope of the embodiments is not to be determined by the specific examples provided above but only by the claims below.

If it is said that an element "A" is coupled to or with element "B," element A may be directly coupled to element B or be indirectly coupled through, for example, element C. When the specification or claims state that a component, feature, structure, process, or characteristic A "causes" a component, feature, structure, process, or characteristic B, it means that "A" is at least a partial cause of "B" but that there may also be at least one other component, feature, structure, process, or characteristic that assists in causing "B." If the specification indicates that a component, feature, structure, process, or characteristic "may", "might", or "could" be included, that particular component, feature, structure, process, or characteristic is not required to be included. If the specification or claim refers to "a" or "an" element, this does not mean there is only one of the described elements.

An embodiment is an implementation or example. Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments. The various appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments. It should be appreciated that in the foregoing description of exemplary embodiments, various features are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various novel aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed embodiments requires more features than are expressly recited in each claim. Rather, as the following claims reflect, novel aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims are hereby expressly incorporated into this description, with each claim standing on its own as a separate embodiment.

In some embodiments, a memory device includes one or more memory arrays for the storage of data; control logic to control operation of the memory device; and ECC (error correction code) logic, including ECC correction logic to correct data and ECC generation logic

to generate ECC code bits and store the ECC bits in the one or more memory arrays. In some embodiments, in a validation mode to validate operation of the ECC logic, the control logic is to allow generation of ECC code bits for a first test value and disable generation of ECC code bits for a second test value.

In some embodiments, the memory device is to write the first test value in a memory array, generate a first set of ECC bits for the first test value, write the second test value in the memory array, and maintain the first set of ECC bits for the second test value.

In some embodiments, the validation mode includes the ECC logic to perform an error check of the second test value using the first set of ECC bits.

In some embodiments, the second test value differs from the first test by a single flipped bit value, and wherein the validation mode is to validate whether a single bit error represented by the single flipped bit value is co.

In some embodiments, the correction of the single bit error is to correct the second test value to be the first test value.

In some embodiments, the second test value differs from the first test by two flipped bit values, wherein the validation mode is to validate whether either: a double bit error represented by the two flipped bit values is correctly read out; or the double bit error represented by the two flipped bit values is miscorrected into a triple bit error, and a double bit error represented by two bit values of the triple bit error is correctly read out in a data value with a third bit value of the triple bit error not being included in the reported data value.

In some embodiments, the validation of operation of the ECC logic is performed by one of a central processing unit (CPU) of a system, another processor of the system, or a separate testing unit.

In some embodiments, one or more non-transitory computer-readable storage mediums having stored thereon data representing sequences of instructions that, when executed by one or more processors, cause the one or more processors to perform operations comprising: writing a first test value in a computer memory; generating a first set of ECC (error correction code) code bits by an on-die ECC logic of the computer memory in response to the writing of the first test value; blocking generation of ECC code bits by a memory controller of the computer memory; and writing a second test value in the computer memory, the second test value differing from the first test value by one or more bits.

In some embodiments, the one or more mediums further include instructions for performing an error check of the second test value based on the first set of ECC code bits.

In some embodiments, the one or more mediums further include instructions for validating operation of the on-die ECC logic based at least in part on the error check of the second test value based on the first set of ECC code bits.

In some embodiments, the second test value differs from the first test value by a single bit value; and the one or more mediums further include instructions for validating whether a single bit error represented by the single bit value is corrected.

In some embodiments, validating whether the single bit error is corrected includes determining whether the second test value is corrected to be the first test value.

In some embodiments, the second test value differs from the first test value by a double bit value; and the one or more mediums further include instructions for validating whether: a double bit error represented by the double bit value is correctly read out, or the double bit error represented by the two flipped bit values is miscorrected into a triple bit error, and a double bit error represented by two bit values of the triple bit error is correctly read out in a data value, with a third bit value of the triple bit error not included in the data value.

In some embodiments, validation of operation of the ECC logic is performed by one of a central processing unit (CPU) of a system, another processor of the system, or a separate testing unit.

In some embodiments, a computing system includes one or more processors; and synchronous dynamic random access memory (SDRAM) for the storage of data, the SDRAM including a memory device having: one or more memory arrays for the storage of data, control logic to control operation of the memory device, and ECC (error correction code) logic, including ECC correction logic to correct data and ECC generation logic to generate ECC code bits and store the ECC bits in the one or more memory arrays. In some embodiments, in a validation mode to validate operation of the ECC logic, the control logic is to allow generation of ECC code bits for a first test value and disable generation of ECC code bits for a second test value.

In some embodiments, the memory device is to write the first test value in a memory array, generate a first set of ECC bits for the first test value, write the second test value in the memory array, and maintain the first set of ECC bits for the second test value.

In some embodiments, the validation mode includes the ECC logic to perform an error check of the second test value using the first set of ECC bits.

In some embodiments, the second test value differs from the first test by a single flipped bit value, and wherein the memory device is to validate whether a single bit error represented by the single flipped bit value is corrected.

In some embodiments, the correction of the single bit error is to correct the second test value to be the first test value.

In some embodiments, the second test value differs from the first test by two flipped bit values, wherein the memory device is to validate whether either: a double bit error represented by the two flipped bit values is correctly read out; or the double bit error represented by the two flipped bit values is miscorrected into a triple bit error, and a double bit error represented by two bit values of the triple bit error is correctly read out in a data value with a third bit value of the triple bit error not being included in the data value.

In some embodiments, validation of operation of the ECC logic is performed by one of a processor of the computing system or a separate testing unit.

In some embodiments, the computing system includes one or more of: a display communicatively coupled to the one or more processors; a network interface communicatively coupled to the one or more processors; or a battery coupled to the one or more processors.

In some embodiments, an apparatus includes means for writing a first test value in a computer memory; means for generating a first set of ECC (error correction code) code bits by an on-die ECC logic of the computer memory in response to the writing of the first test value; means for blocking generation of ECC code bits by a memory controller of the computer memory; and means for writing a second test value in the computer memory, the second test value differing from the first test value by one or more bits.

In some embodiments, the apparatus further includes means for performing an error check of the second test value based on the first set of ECC code bits.

In some embodiments, the apparatus further includes means for validating operation of the on-die ECC logic based at least in part on the error check of the second test value based on the first set of ECC code bits.

In some embodiments, the second test value differs from the first test value by a single bit value; and the apparatus further includes means for validating whether a single bit error represented by the single bit value is corrected.

In some embodiments, the means for validating whether the single bit error is corrected includes means for determining whether the second test value is corrected to be the first test value.

In some embodiments, the second test value differs from the first test value by a double bit value; and the apparatus further includes means for validating whether: a double bit error represented by the double bit value is correctly read out, or the double bit error represented by the two flipped bit values is miscorrected into a triple bit error, and a double bit error

represented by two bit values of the triple bit error is correctly read out in a data value with a third bit value of the triple bit error not being included in the data value.

In some embodiments, validation of operation of the on-die ECC logic is performed by one of a central processing unit (CPU) of a system, another processor, or a separate testing unit.

In some embodiments, a method includes writing a first test value in a computer memory; generating a first set of ECC (error correction code) code bits by an on-die ECC logic of the computer memory in response to the writing of the first test value; blocking generation of ECC code bits by a memory controller of the computer memory; and writing a second test value in the computer memory, the second test value differing from the first test value by one or more bits.

In some embodiments, the method further includes performing an error check of the second test value based on the first set of ECC code bits.

In some embodiments, the method further includes validating operation of the on-die ECC logic based at least in part on the error check of the second test value based on the first set of ECC code bits.

In some embodiments, the second test value differs from the first test value by a single bit value; and the method further includes validating whether a single bit error represented by the single bit value is corrected.

In some embodiments, validating whether the single bit error is corrected includes determining whether the second test value is corrected to be the first test value.

In some embodiments, the second test value differs from the first test value by a double bit value; and the method further includes validating whether: a double bit error represented by the double bit value is correctly read out, or the double bit error represented by the two flipped bit values is miscorrected into a triple bit error, and a double bit error represented by two bit values of the triple bit error is correctly read out in a data value with a third bit value of the triple bit error not being included in the data value.

In some embodiments, validation of operation of the on-die ECC logic is performed by one of a central processing unit (CPU) of a system, another processor, or a separate testing unit.

## CLAIMS

What is claimed is:

1. A memory device comprising:

one or more memory arrays for the storage of data;

control logic to control operation of the memory device; and

ECC (error correction code) logic, including ECC correction logic to correct data and ECC generation logic to generate ECC code bits and store the ECC bits in the one or more memory arrays;

wherein the control logic is to block generation an ECC code for a test value in a validation mode for the memory device, the blocking of the generation of the generation of the ECC code resulting in ECC bits in memory remaining unchanged from prior ECC bit values.

2. The memory device of claim 1, wherein the memory device is to write a first value in a memory array, generate a first set ECC bits for the first value, write a second value in the memory array, and block generation of ECC bits for the second value.

3. The memory device of claim 2, wherein the ECC logic is to perform an error check of the second value using the first set of ECC bits.

4. The memory device of claim 3, wherein the memory device is to determine if a single bit error is corrected.

5. The memory device of claim 3, wherein the memory device is to determine if a two-bit error is miscorrected into a three-bit error.

6. The memory device of claim 5, wherein the memory device is to determine whether the error is properly read from the one or more memory arrays.

7. One or more non-transitory computer-readable storage mediums having stored thereon data representing sequences of instructions that, when executed by one or more processors, cause the one or more processors to perform operations comprising:

writing a first test value in a computer memory;

generating a first set of ECC (error correction code) code bits by an on-die ECC logic of the computer memory in response to the writing of the first test value;

writing a second test value in the computer memory, the second test value differing from the first test value by one or more bits; and

blocking generation of a second set of ECC code bits by a memory controller of the computer memory.

8. The one or more mediums of claim 7, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:

performing an error check of the second test value based on the first set of ECC code bits.

9. The one or more mediums of claim 8, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:

performing validation of the ECC logic based on the performance of the error check.

10. The one or more mediums of claim 8, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:

determining if a single bit error is corrected.

11. The one or more mediums of claim 8, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:

determining if a two-bit error is miscorrected into a three-bit error.

12. The one or more mediums of claim 11, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:

determining whether the error is properly read from the memory.

13. A computing system comprising:

one or more processors;

synchronous dynamic random access memory (SDRAM) for the storage of data, the SDRAM including a memory device having:

one or more memory arrays for the storage of data;

control logic to control operation of the memory device; and

ECC (error correction code) logic, including ECC correction logic to correct data and ECC generation logic to generate ECC code bits and store the ECC bits in the one or more memory arrays;

wherein the control logic of the memory device is to block generation an ECC code for a test value in a validation mode for the memory device, the blocking of the generation of the generation of the ECC code resulting in ECC bits in memory remaining unchanged from prior ECC bit values.

14. The computing system of claim 13, wherein the memory device is to write a first value in a memory array, generate a first set ECC bits for the first value, write a second value in the memory array, and block generation of ECC bits for the second value.

15. The computing system of claim 14, wherein the ECC logic is to perform an error check of the second value using the first set of ECC bits.

16. The computing system of claim 15, wherein the memory device is to determine if a single bit error is corrected.

17. The computer system of claim 15, wherein the memory device is to determine if a two-bit error is miscorrected into a three-bit error.

18. The computing system of claim 17, wherein the memory device is to determine whether the error is properly read from the one or more memory arrays.

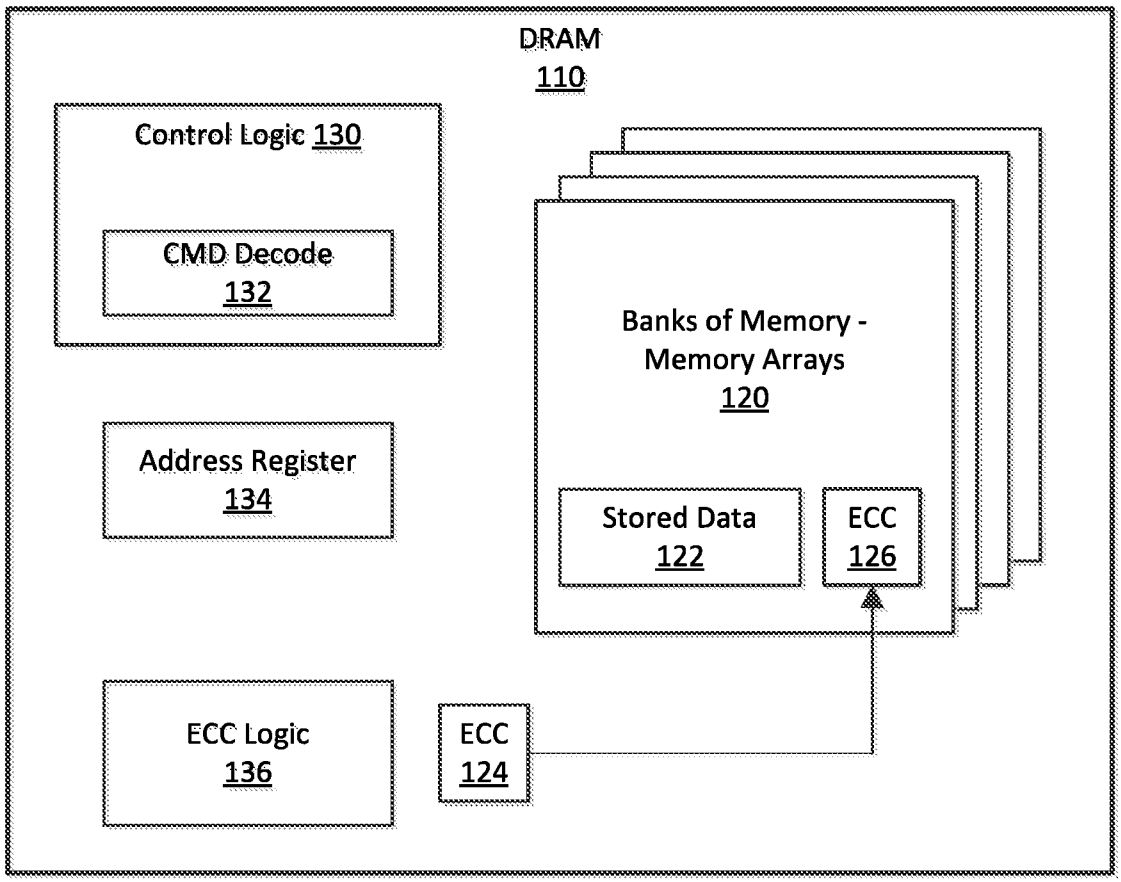
DRAM
110
Control Logic 130
CMD Decode
132
Address Register
134
ECC Logic
136
ECC
124
Banks of Memory -
Memory Arrays
120
Stored Data
122
ECC
126

*Fig. 1*

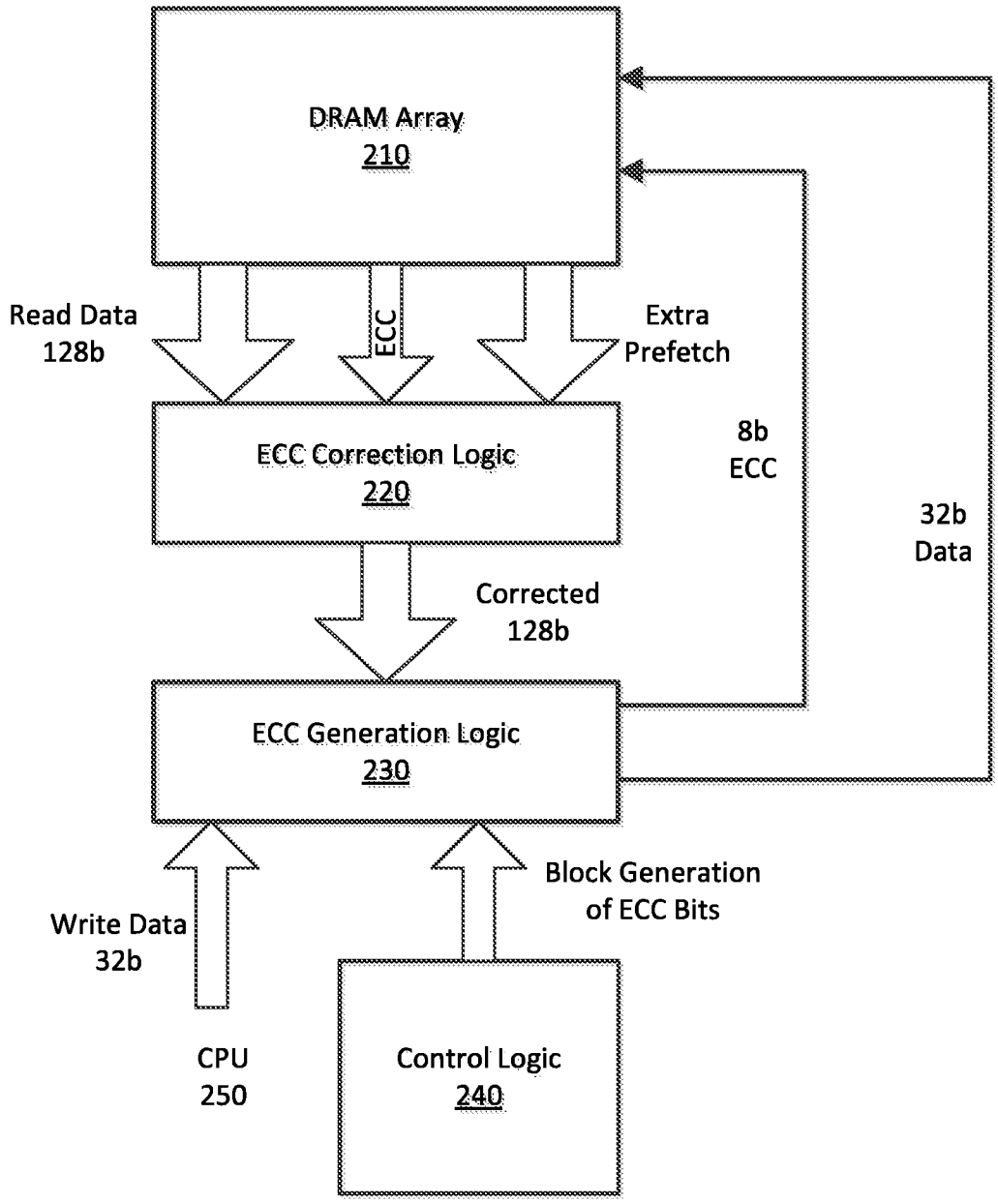
DRAM Array
210
Read Data
128b
ECC
Extra
Prefetch
ECC Correction Logic
220
8b
ECC
32b
Data
Corrected
128b
ECC Generation Logic
230
Block Generation
of ECC Bits
Write Data
32b
CPU
250
Control Logic
240

*Fig. 2A*

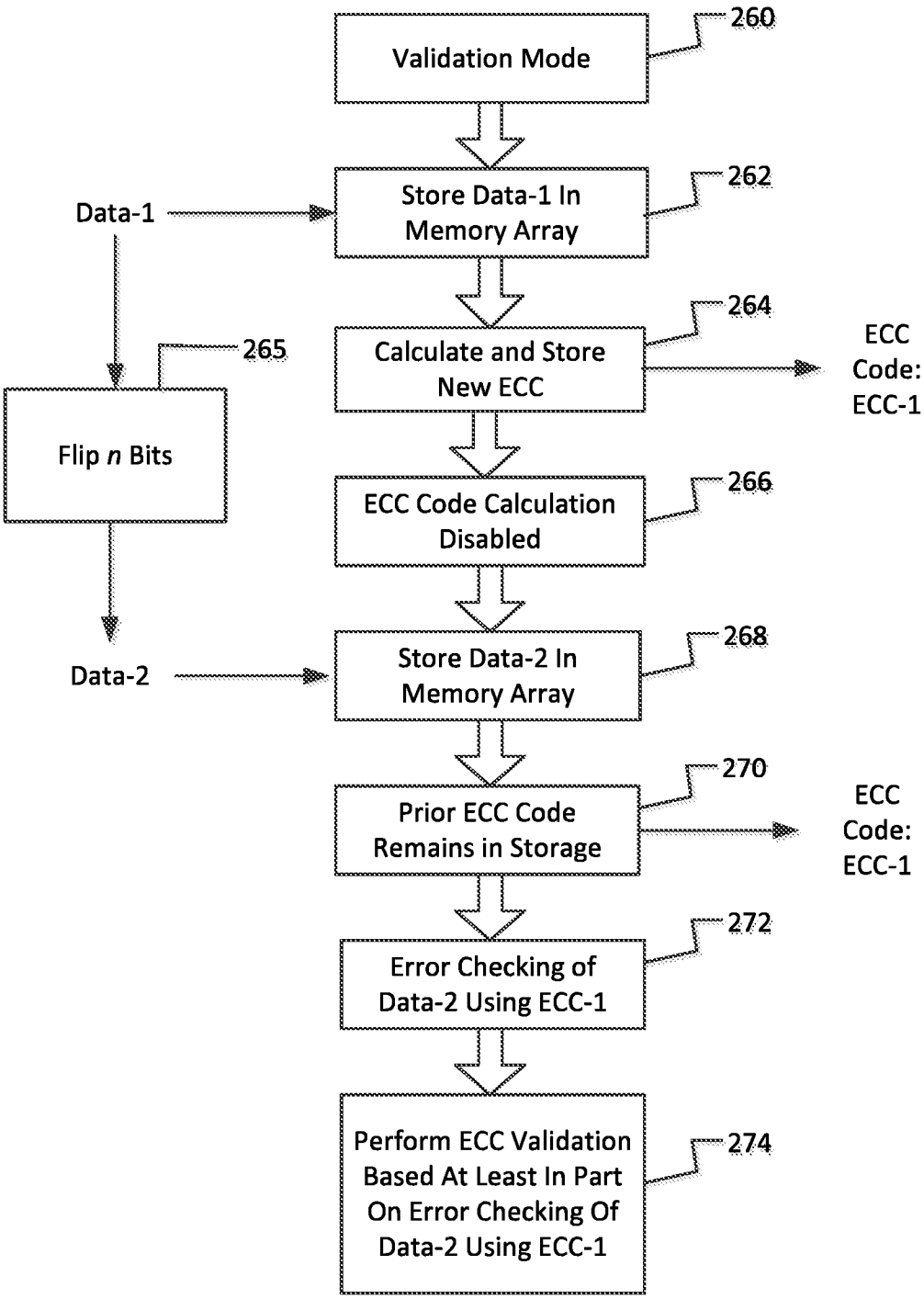
Validation Mode
260
Store Data-1 In Memory Array
262
Data-1
Calculate and Store New ECC
264
ECC Code: ECC-1
265
Flip n Bits
ECC Code Calculation Disabled
266
Data-2
Store Data-2 In Memory Array
268
Prior ECC Code Remains in Storage
270
ECC Code: ECC-1
Error Checking of Data-2 Using ECC-1
272
Perform ECC Validation Based At Least In Part On Error Checking Of Data-2 Using ECC-1
274

*Fig. 2B*

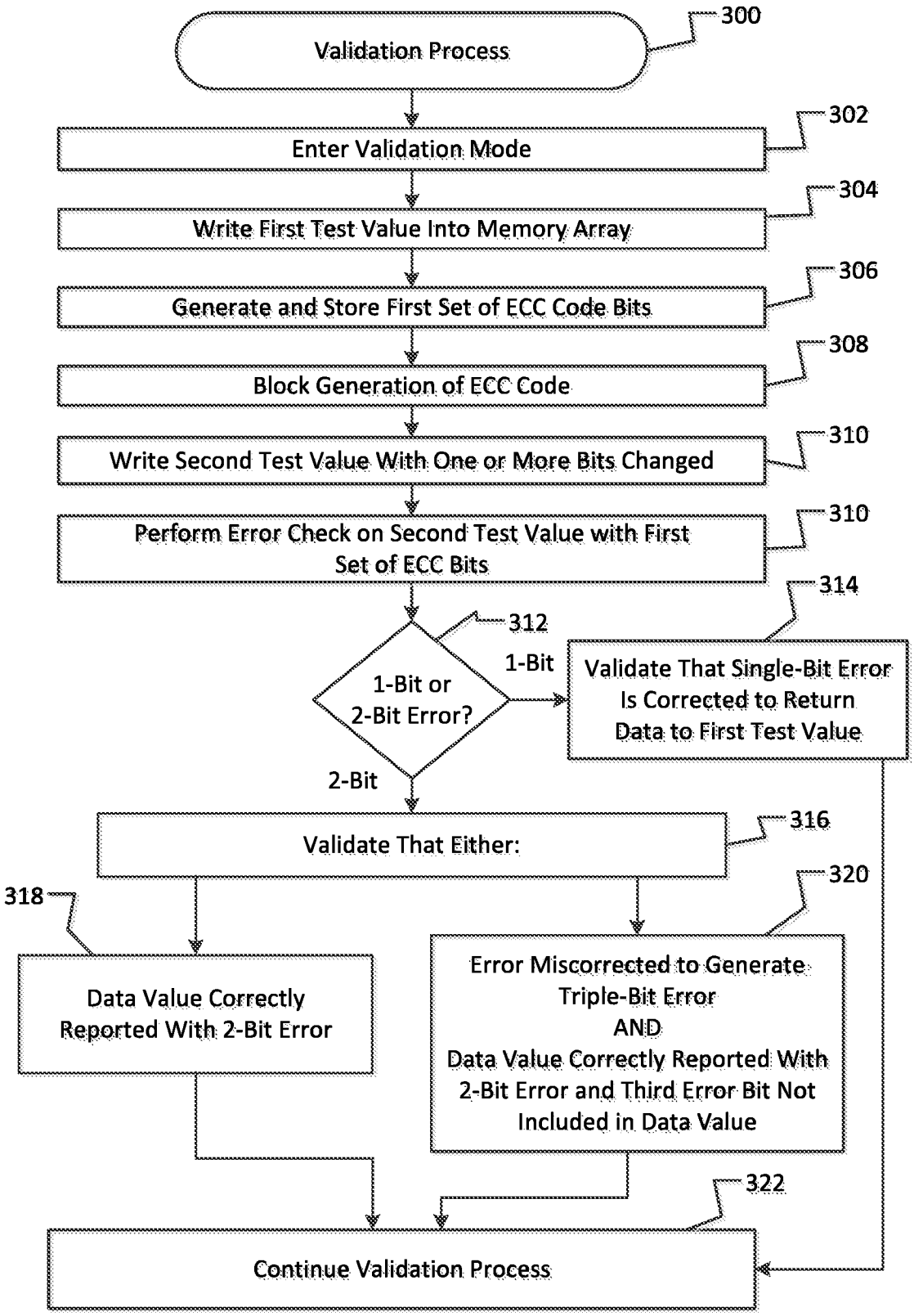
Validation Process
300
Enter Validation Mode
302
Write First Test Value Into Memory Array
304
Generate and Store First Set of ECC Code Bits
306
Block Generation of ECC Code
308
Write Second Test Value With One or More Bits Changed
310
Perform Error Check on Second Test Value with First Set of ECC Bits
310
1-Bit or 2-Bit Error?
312
1-Bit
Validate That Single-Bit Error Is Corrected to Return Data to First Test Value
314
2-Bit
Validate That Either:
316
318
Data Value Correctly Reported With 2-Bit Error
320
Error Miscorrected to Generate Triple-Bit Error AND Data Value Correctly Reported With 2-Bit Error and Third Error Bit Not Included in Data Value
322
Continue Validation Process

*Fig. 3*

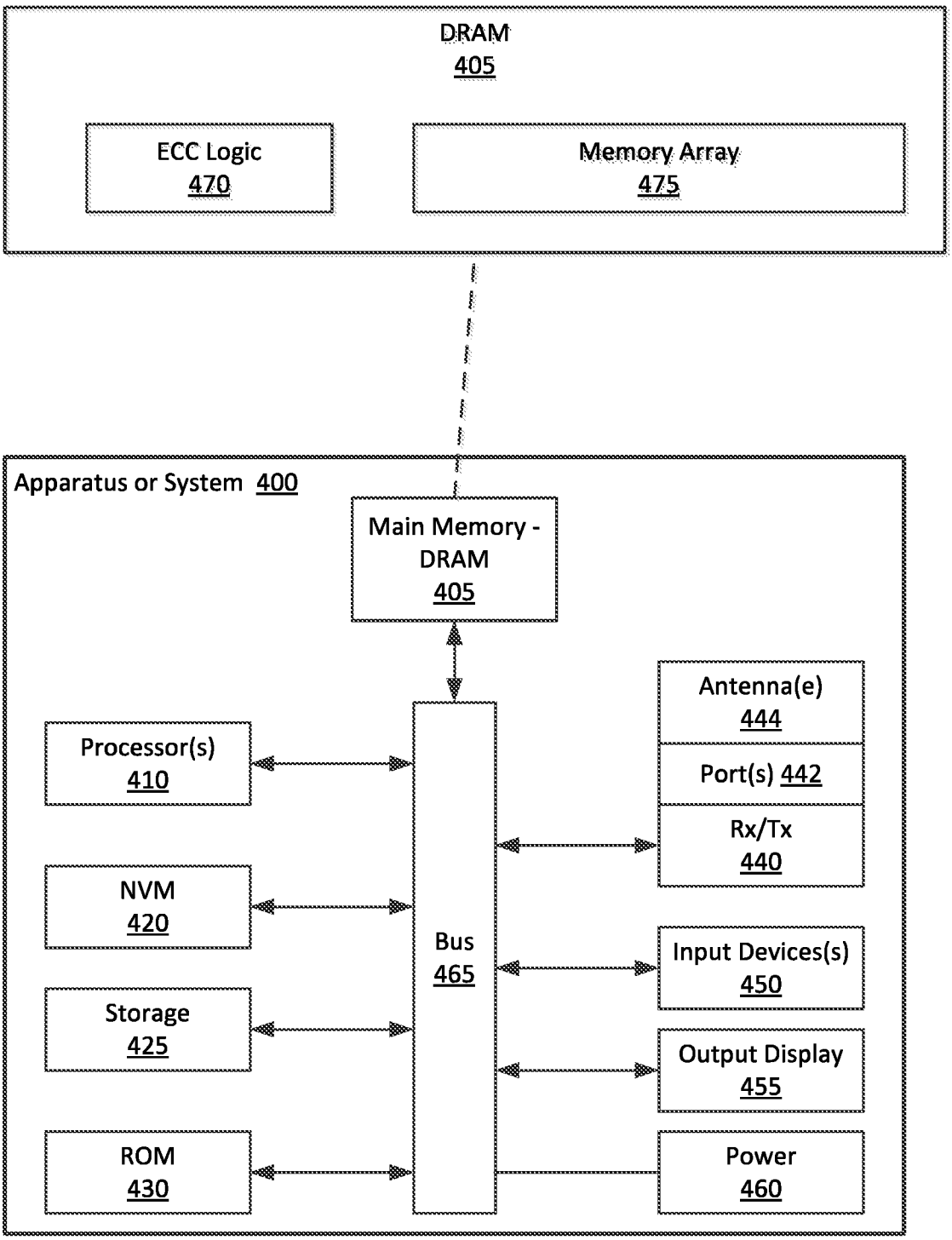
DRAM
405
ECC Logic
470
Memory Array
475
Apparatus or System 400
Main Memory - DRAM
405
Processor(s)
410
NVM
420
Storage
425
ROM
430
Bus
465
Antenna(e)
444
Port(s) 442
Rx/Tx
440
Input Devices(s)
450
Output Display
455
Power
460

*Fig. 4*

**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/US2017/017236**

**A. CLASSIFICATION OF SUBJECT MATTER**

**G06F 11/10(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)
G06F 11/10; H03M 13/00; G06F 11/08; G11B ; G11C 29/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: storage, memory, control logic, ECC (error correction code), generate, block, test value, unchange, 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. |
|---|---|---|
| A | US 2011-0307758 A1 (JEREMY FILLINGIM) 15 December 2011<br>See paragraphs [0012], [0041], [0094], [0104], and [0122]; and figure 5. | 1-18 |
| A | US 2013-0254626 A1 (MICRON TECHNOLOGY, INC) 26 September 2013<br>See paragraphs [0019] and [0035]-[0036]; and figure 1. | 1-18 |
| A | US 2004-0225943 A1 (CHRISTOPHER M. BRUEGGEN) 11 November 2004<br>See paragraphs [0028]-[0029] and figure 2. | 1-18 |
| A | WO 2005-029469 A2 (INFINEON TECHNOLOGIES FLASH LTD.) 31 March 2005<br>See paragraphs [008]-[0010] and figure 1. | 1-18 |
| A | US 2007-0113155 A1 (HIROYUKI TAKAHASHI) 17 May 2007<br>See paragraphs [0008]-[0009] and figure 1. | 1-18 |

☐ 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 |
|---|---|
| 19 May 2017 (19.05.2017) | **19 May 2017 (19.05.2017)** |

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.
**PCT/US2017/017236**

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2011-0307758 A1 | 15/12/2011 | US 2011-314354 A1 | 22/12/2011 |
| | | US 8892980 B2 | 18/11/2014 |
| | | WO 2011-159805 A2 | 22/12/2011 |
| | | WO 2011-159805 A3 | 05/04/2012 |
| | | WO 2011-159806 A2 | 22/12/2011 |
| | | WO 2011-159806 A3 | 05/04/2012 |
| US 2013-0254626 A1 | 26/09/2013 | US 2008-109705 A1 | 08/05/2008 |
| | | US 2011-138252 A1 | 09/06/2011 |
| | | US 2014-047305 A1 | 13/02/2014 |
| | | US 7900120 B2 | 01/03/2011 |
| | | US 8413007 B2 | 02/04/2013 |
| | | US 8601341 B2 | 03/12/2013 |
| | | US 8880974 B2 | 04/11/2014 |
| US 2004-0225943 A1 | 11/11/2004 | TW 200501167 A | 01/01/2005 |
| | | TW I308760 B | 11/04/2009 |
| | | US 7149945 B2 | 12/12/2006 |
| WO 2005-029469 A2 | 31/03/2005 | CN 1898657 A | 17/01/2007 |
| | | DE 112004001781 B4 | 24/12/2013 |
| | | DE 112004001781 T5 | 11/01/2007 |
| | | US 2005-0066110 A1 | 24/03/2005 |
| | | US 7197594 B2 | 27/03/2007 |
| | | WO 2005-029469 A3 | 15/12/2005 |
| US 2007-0113155 A1 | 17/05/2007 | CN 1963950 A | 16/05/2007 |
| | | JP 2007-133986 A | 31/05/2007 |
| | | US 7810016 B2 | 05/10/2010 |