(19) **United States**

(12) **Patent Application Publication** (10) Pub. No.: **US 2015/0220386 A1**

Asnaashari et al. (43) **Pub. Date:** **Aug. 6, 2015**

(54) **DATA INTEGRITY IN MEMORY CONTROLLERS AND METHODS**

(71) Applicant: **Round Rock Research, LLC,** Parsippany, NJ (US)

(72) Inventors: **Mehdi Asnaashari**, Danville, CA (US); **Ronald Yamada**, Santa Clara, CA (US); **Siamack Nemazie**, Los Altos Hills, CA (US); **Jui-Yao Yang**, San Jose, CA (US)

(73) Assignee: **Micron Technology, Inc.**, Boise, ID (US)

(21) Appl. No.: **14/688,323**

(22) Filed: **Apr. 16, 2015**

Related U.S. Application Data

(60) Continuation of application No. 13/920,451, filed on Jun. 18, 2013, now Pat. No. 9,015,553, which is a division of application No. 12/388,305, filed on Feb. 18, 2009, now Pat. No. 8,468,417.

Publication Classification

(51) **Int. Cl.**
   G06F 11/10 (2006.01)

(52) **U.S. Cl.**
   CPC *G06F 11/1048* (2013.01); *G06F 11/1004* (2013.01)

(57) **ABSTRACT**

The present disclosure includes methods, devices, and systems for data integrity in memory controllers. One memory controller embodiment includes a host interface and first error detection circuitry coupled to the host interface. The memory controller can include a memory interface and second error detection circuitry coupled to the memory interface. The first error detection circuitry can be configured to calculate error detection data for data received from the host interface and to check the integrity of data transmitted to the host interface. The second error detection circuitry can be configured to calculate error correction data for data and first error correction data transmitted to the memory interface and to check integrity of data and first error correction data received from the memory interface.
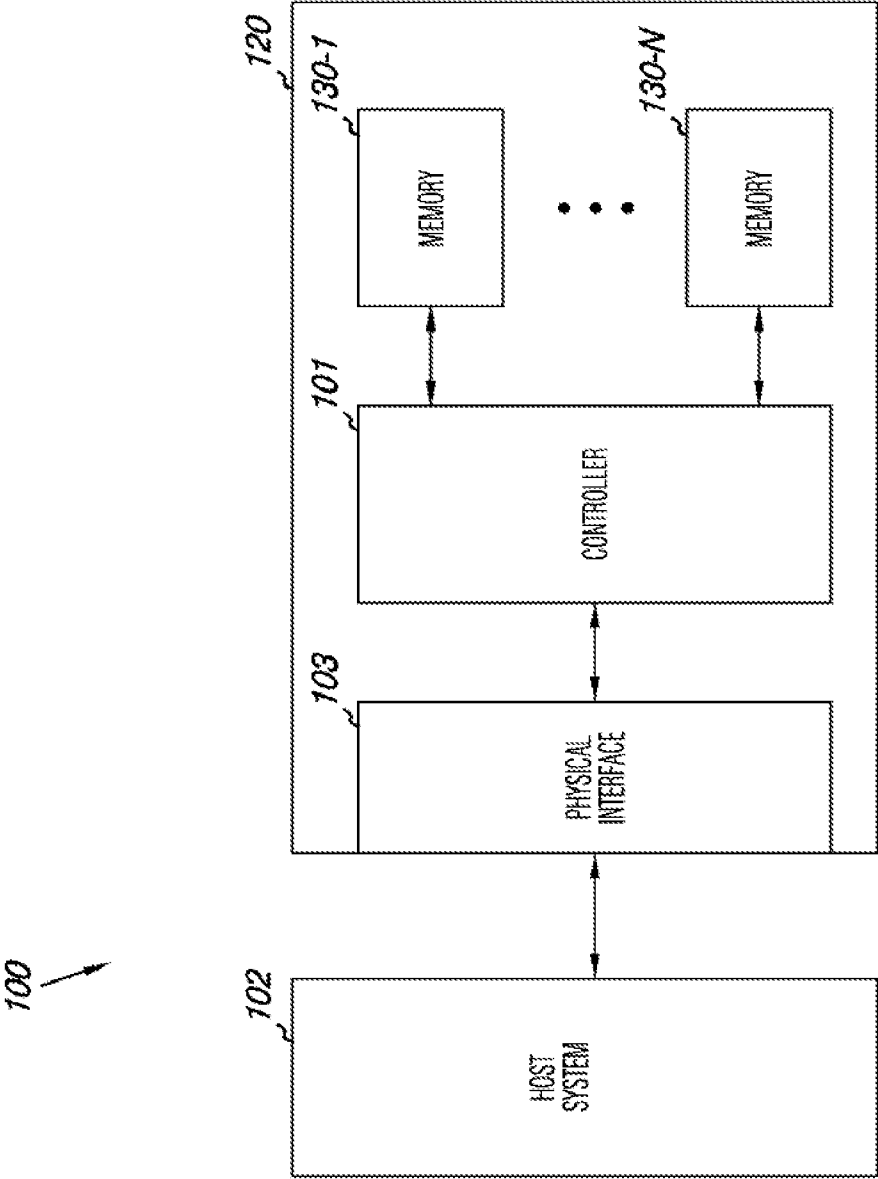
100
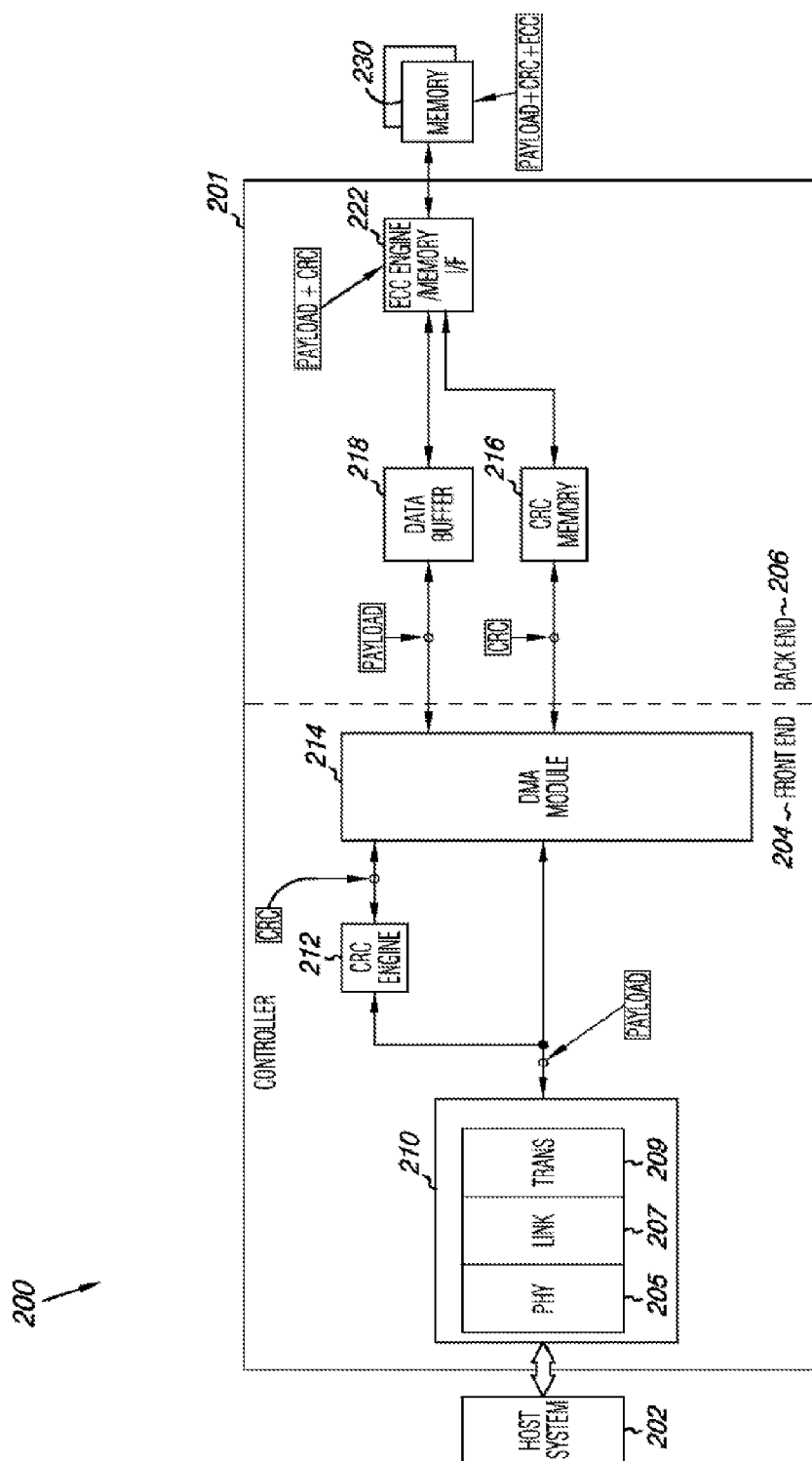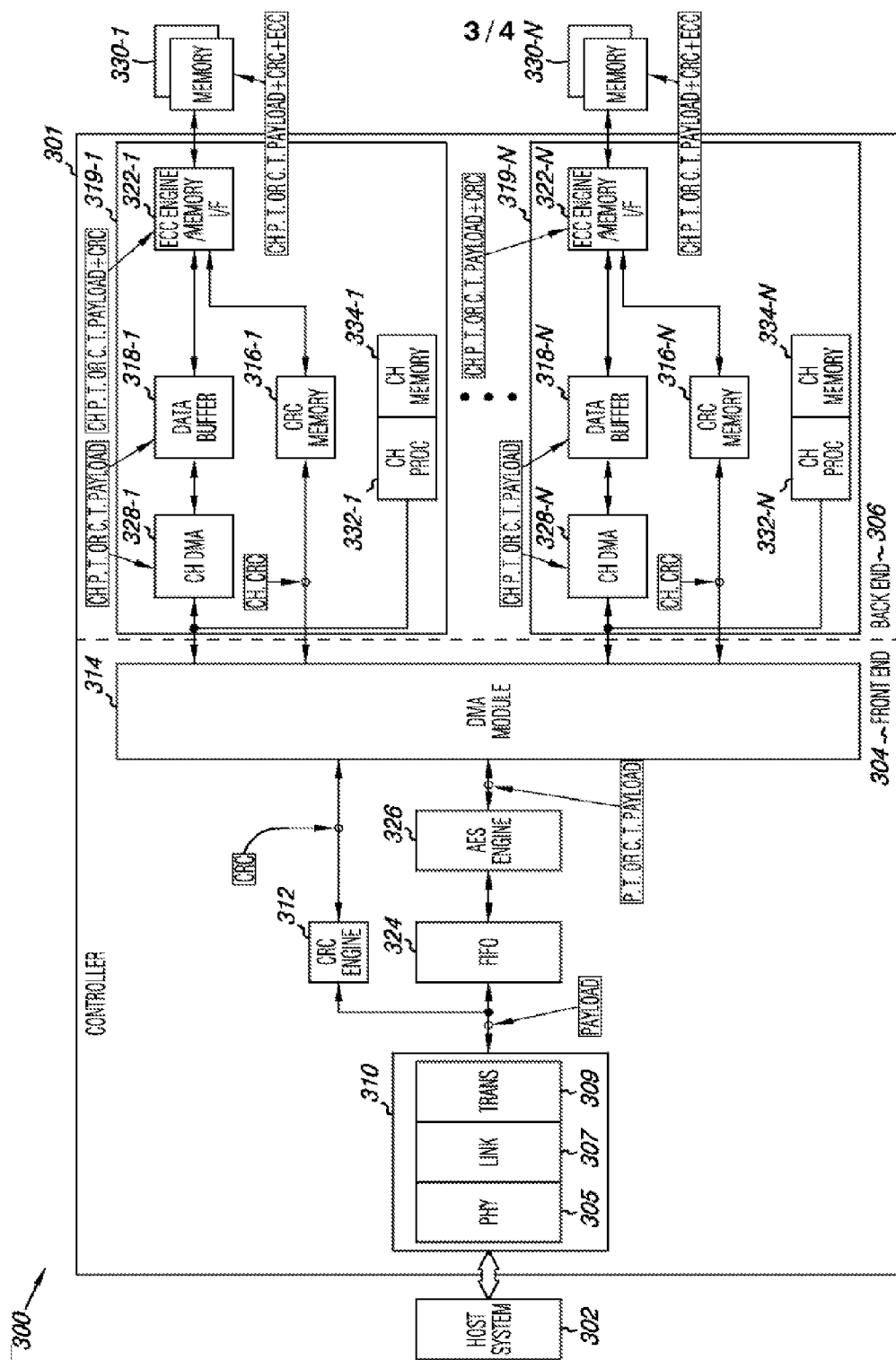
Fig. 1

Fig. 2

Fig. 3

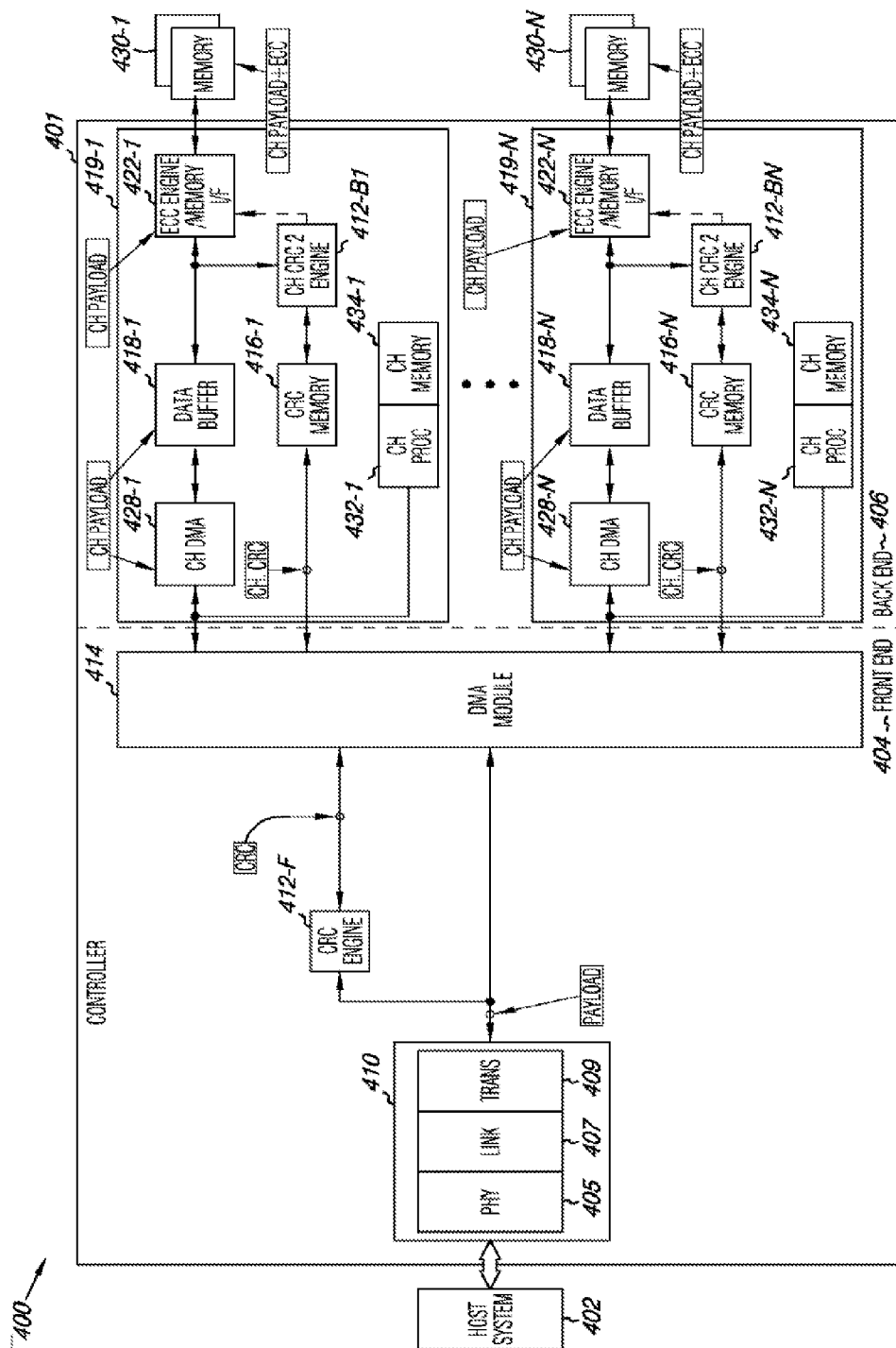*Fig. 4*

DATA INTEGRITY IN MEMORY CONTROLLERS AND METHODS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. patent application Ser. No. 13/920,451, filed on Jun. 18, 2013, and is scheduled to issue as U.S. Pat. No. 9,015,553 on Apr. 21, 2015, which is a divisional of U.S. patent application Ser. No. 12/388,305, filed on Feb. 18, 2009, issued as U.S. Pat. No. 8,468,417, on Jun. 18, 2013, all of which are incorporated herein by reference.

TECHNICAL FIELD

[0002] The present disclosure relates generally to semiconductor memory devices, methods, and systems, and more particularly, to data integrity.

BACKGROUND OF THE INVENTION

[0003] Memory devices are typically provided as internal, semiconductor, integrated circuits in computers or other electronic devices. There are many different types of memory including volatile and non-volatile memory. Volatile memory can require power to maintain its data and includes random-access memory (RAM), dynamic random access memory (DRAM), and synchronous dynamic random access memory (SDRAM), among others. Non-volatile memory can provide persistent data by retaining stored information when not powered and can include NAND flash memory, NOR flash memory, read only memory (ROM), Electrically Erasable Programmable ROM (EEPROM), Erasable Programmable ROM (EPROM), and phase change random access memory (PCRAM), among others.

[0004] Memory devices can be combined together to form a solid state drive (SSD). An SSD can include non-volatile memory, e.g., NAND flash memory and NOR flash memory, and/or can include volatile memory, e.g., DRAM and SRAM, among various other types of non-volatile and volatile memory.

[0005] An SSD can be used to replace hard disk drives as the main storage device for a computer, as the SSD can have advantages over hard drives in terms of performance, size, weight, ruggedness, operating temperature range, and power consumption. For example, SSDs can have superior performance when compared to magnetic disk drives due to their lack of moving parts, which may avoid seek time, latency, and other electro-mechanical delays associated with magnetic disk drives. SSD manufacturers can use non-volatile flash memory to create flash SSDs that may not use an internal battery supply, thus allowing the drive to be more versatile and compact.

[0006] An SSD can include one or more memory devices, e.g., one or more memory chips. As one of ordinary skill in the art will appreciate, a memory chip can include one or more dies. Each die can include one or more memory arrays and peripheral circuitry thereon. A memory array can include one or more planes, with each plane including one or more physical blocks of memory cells. Each physical block can include one or more pages of memory cells that can store one or more sectors of data.

[0007] SSDs can interface with a host system with a serial interface such as universal serial bus (USB), serial advanced technology attachment (SATA), or peripheral component interconnect express (PCIe), among others. Serial interfaces such as USB, SATA, and PCIe may have error detection mechanisms such as cyclic redundancy check (CRC) built into the link layer or transport layer of the architecture of the respective interfaces. CRC may include the ability to detect the presence of errors caused by noise or other impairments during transmission of data from a transmitter to a receiver. CRC data generation may be based on a frame structure supported by a particular interface.

[0008] For example, a SATA frame structure may include a start-of-frame (SOF), followed by a frame information structure (FIS), followed by CRC data, and followed by an end-of-frame (EOF). SATA may use 32-bits of CRC data calculated over the contents of the FIS. An example of a 32-bit CRC polynomial used in the calculation of CRC data is: $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$. In the event that the FIS contains an odd number of words, e.g., 16 bit segments, the last word of the FIS may be padded with zeros to form a full double word (dword), e.g., 32 bits. A SATA frame may include 2064 dwords including the FIS and CRC data, wherein the FIS payload may include 2048 dwords (8192 bytes). However, the payload may include less data.

[0009] As one of ordinary skill in the art will appreciate, logical block addressing is a scheme that can be used by a host for identifying a sector of information, e.g., each sector can correspond to a unique logical block address (LBA). A sector may be the smallest addressable portion of a storage volume. As an example, a sector of data can be a number of bytes of data, e.g., 512 bytes. Because each payload at a serial host interface, e.g., a SATA interface, does not necessarily include the same number of bytes, and because payloads at a serial host interface of a mass storage device may be of different units, the CRC data may not propagate past the host interface, e.g., the link or transport layer of the host interface.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a functional block diagram of a computing system including a memory system in accordance with one or more embodiments of the present disclosure.

[0011] FIG. 2 is a functional block diagram of a system including a memory controller in accordance with one or more embodiments of the present disclosure.

[0012] FIG. 3 is a functional block diagram of a system including a memory controller in accordance with one or more embodiments of the present disclosure.

[0013] FIG. 4 is a functional block diagram of a system including a memory controller in accordance with one or more embodiments of the present disclosure.

DETAILED DESCRIPTION

[0014] The present disclosure includes methods, devices, and systems for data integrity in memory controllers. One memory controller embodiment includes a host interface and first error detection circuitry coupled to the host interface. The memory controller can include a memory interface and second error detection circuitry coupled to the memory interface. The first error detection circuitry can be configured to calculate error detection data for data received from the host interface and to check the integrity of data transmitted to the host interface. The second error detection circuitry can be configured to calculate error correction data for data and first error correction data transmitted to the memory interface and

to check integrity of data and first error correction data received from the memory interface.

[0015] In the following detailed description of the present disclosure, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration how one or more embodiments of the disclosure may be practiced. These embodiments are described in sufficient detail to enable those of ordinary skill in the art to practice the embodiments of this disclosure, and it is to be understood that other embodiments may be utilized and that process, electrical, and/or structural changes may be made without departing from the scope of the present disclosure. As used herein, the designator "N," particularly with respect to reference numerals in the drawings, indicates that a number of the particular feature so designated can be included with one or more embodiments of the present disclosure. The designators can represent the same or different numbers of the particular features.

[0016] The figures herein follow a numbering convention in which the first digit or digits correspond to the drawing figure number and the remaining digits identify an element or component in the drawing. Similar elements or components between different figures may be identified by the use of similar digits. For example, 110 may reference element "10" in FIG. 1, and a similar element may be referenced as 210 in FIG. 2. As will be appreciated, elements shown in the various embodiments herein can be added, exchanged, and/or eliminated so as to provide a number of additional embodiments of the present disclosure. In addition, as will be appreciated, the proportion and the relative scale of the elements provided in the figures are intended to illustrate the embodiments of the present disclosure, and should not be taken in a limiting sense.

[0017] FIG. 1 is a functional block diagram of a computing system 100 including a memory system 120 in accordance with one or more embodiments of the present disclosure. In the embodiment illustrated in FIG. 1, the memory system 120, e.g., a solid state drive (SSD), can include a memory controller 101, a physical interface 103, and one or more solid state memory devices 130-1, . . . , 130-N, e.g., NAND flash devices. As illustrated in FIG. 1, the memory controller 101 can be coupled to the physical interface 103 and to the solid state memory devices 130-1, . . . , 130-N.

[0018] The physical interface 103 can be used to communicate information between the memory system 120 and another device such as a host system 102. Host system 102 can include a memory access device, e.g., a processor. One of ordinary skill in the art will appreciate that "a processor" can intend one or more processors, such as a parallel processing system, a number of coprocessors, etc. Examples of host systems include laptop computers, personal computers, digital cameras, digital recording and playback devices, mobile telephones, PDAs, memory card readers, interface hubs, and the like. For one or more embodiments, the physical interface 103 can be in the form of a standardized physical interface. For example, when the memory system 120 is used for data storage in a computing system 100, the physical interface 103 can be a serial advanced technology attachment (SATA), peripheral component interconnect express (PCIe), or a universal serial bus (USB), among other physical interfaces. In general, however, physical interface 103 can provide a physical connection for passing control, address, data, and other signals between a host interface, e.g., host interface 210 in

FIG. 2, of the controller 101 of the memory system 120 and a host system 102 having compatible receptors for the physical interface 103.

[0019] The memory controller 101 can communicate with the solid state memory devices 130-1, . . . , 130-N to read, write, and erase data, among other operations. Memory controller 101 can have circuitry that may be one or more integrated circuits and/or discrete components. For one or more embodiments, the circuitry in memory controller 101 may include control circuitry for controlling access across the solid state memory devices 130-1, . . . , 130-N and circuitry for providing a translation layer between a host system 102 and the memory system 120. Thus, a memory controller could selectively couple an I/O connection (not shown in FIG. 1) of a solid state memory device 130-1, . . . , 130-N to receive the appropriate signal at the appropriate I/O connection at the appropriate time. Similarly, the communication protocol between a host system 102 and the memory system 120 may be different than what is required for access of a solid state memory device 130-1, . . . , 130-N. Memory controller 101 could then translate the commands received from a host into the appropriate commands to achieve the desired access to a solid state memory device 130-1, . . . , 130-N.

[0020] The embodiment of FIG. 1 can include additional circuitry that is not illustrated so as not to obscure embodiments of the present disclosure. For example, the memory system 120 can include address circuitry to latch address signals provided over I/O connections through I/O circuitry. Address signals can be received and decoded by a row decoder and a column decoder to access the solid state memory devices 130-1, . . . , 130-N. It will be appreciated by those skilled in the art that the number of address input connections can depend on the density and architecture of the solid state memory devices 130-1, . . . , 130-N.

[0021] FIG. 2 is a functional block diagram of a system 200 including a memory controller 201 in accordance with one or more embodiments of the present disclosure. Controller 201 can be analogous to controller 101 illustrated in FIG. 1. In one or more embodiments, controller 201 can be a component of a memory system, such as an SSD. It will be appreciated by one of ordinary skill in the art that additional circuitry and components can be provided beyond those illustrated in FIG. 2, and that the controller detail of FIG. 2 has been reduced to facilitate ease of illustration.

[0022] As shown in FIG. 2, memory controller 201 can be coupled to one or more solid state memory devices 230. The solid state memory devices 230 can be analogous to solid state memory devices 130-1, . . . , 130-N, illustrated in FIG. 1. Controller 201 can include a front end portion 204 and a back end portion 206. The memory controller can include a number of front end components coupled between a host interface 210 and data transfer circuitry, e.g., a direct memory access (DMA) module 214. The memory controller can also include a number of back end components coupled between the DMA module 214 and a memory interface, e.g., a memory interface component of error detection circuitry/memory interface (I/F) 222. In one or more embodiments, the error detection circuitry component of the error detection circuitry/memory I/F 222 can be an error correction code (ECC) engine. In general, the memory controller 201 can process commands and data received from or transferred to a host system 202, e.g., host system 102 in FIG. 1, with the front end 204. In general, the memory controller 201 can manage communications with the solid state memory devices 230 to read, write,

and erase data, among other operations, on the solid state memory devices **230** with the back end **206**. However, certain aspects of command processing and memory communication management can be handled by the controller **201** on either or both of the front end **204** and back end **206**. As illustrated in FIG. **2**, the host interface **210** and DMA module **214** can be front end components.

[0023] As shown in FIG. **2**, the front end portion **204** can include a host interface **210** that can be coupled to a host system **202**, e.g., host system **102** in FIG. **1**. The host interface **210** can be configured to receive data from the host system. The data received from the host interface can be a data payload including a number of sectors of data. In one or more embodiments, the data received from the host interface can be streaming data. The host interface **210** can interface with the host system **202** through a number of layers including a physical layer **205**, a link layer **207**, and a transport layer **209**. One of ordinary skill in the art will appreciate that the number of layers for a particular host interface can be defined by an interface standard such as SATA, PCIe, USB, etc. As used herein, a transport layer can indicate at least a transport layer as part of a SATA standard or a transaction layer as part of a PCIe standard. One of ordinary skill in the art will appreciate that a transport layer according to a SATA standard can be analogous to a transaction layer according to a PCIe standard. Host interface **210** can be coupled to a physical interface, e.g., physical interface **103** illustrated in FIG. **1**, on a memory system, e.g., an SSD, to communicate with a host system **202**. Such detail is not shown in FIG. **2**, for ease of illustration.

[0024] The host interface **210** can be coupled with a DMA module **214** and front end error detection circuitry, e.g., cyclic redundancy check (CRC) engine **212**. Although illustrated as "CRC ENGINE" **212**, error detection circuitry can provide functionality other than cyclic redundancy checks. For example, error detection circuitry can include repetition schemes, parity schemes, checksums, Hamming distance based checks, hash functions, horizontal and vertical redundancy checks, polarity schemes, and/or error correction schemes such as ECC, among others. In one or more embodiments, the host interface **210** can be directly coupled with the front end CRC engine **212** and directly coupled with the DMA module **214**.

[0025] In one or more embodiments, the front end CRC engine **212** can be coupled to the host interface **210**, and thus to the host system, by a link layer **207** and/or a transport layer **209**. The front end error detection circuitry, e.g., CRC engine **212**, can be configured to detect errors in data, e.g., while the data is in a memory system such as memory system **120** in FIG. **1**. For example, the front end CRC engine **212** can calculate error correction data, e.g., CRC data, corresponding to data, e.g., a sector of a data payload such as a DMA payload, received from the link layer **207** and/or the transport later **209**, e.g., as part of a write operation. In one or more embodiments, the front end CRC engine **212** can calculate CRC data on a sector-by-sector basis. In one or more embodiments, the front end CRC engine **212** can be coupled to the host interface **210** by a link layer **207** and/or a transport layer **209** and receive a number of sectors of data, e.g., a DMA payload, from the link layer **207** and/or the transport layer **209**. In one or more embodiments, the front end CRC engine **212** can receive the number of sectors of data from either the link layer **207** exclusively, or from the transport layer **209** exclusively. However, embodiments are so limited. CRC data that may have been generated by a host system **202** for data

transferred to a memory controller **201** may be stripped from the data at a transport layer **209** for PCIe type interfaces or at a link layer **207** for SATA type interfaces. However, according to one or more embodiments of the present disclosure, the front end CRC engine **212** can calculate and check CRC data per sector of data received from a host interface **210**, to which it can be coupled.

[0026] The DMA module **214** can be coupled to a back end data buffer **218** and a back end error detection memory, e.g., CRC memory **216**. The back end data buffer **218** can be coupled to the ECC engine/memory I/F **222**. The back end data buffer **218** can be configured to buffer at least a portion, e.g., a number of sectors of data, of a DMA payload received from DMA module **214** for the ECC engine/memory I/F **222** during a write operation. The back end data buffer **218** can be configured to buffer a number of sectors of data received from the ECC engine/memory IN **222** during a read operation. The back end data buffer **218** and back end CRC memory **216** can be coupled to error detection circuitry, e.g., an ECC engine and memory interface. In one or more embodiments, and as illustrated in the embodiment of FIG. **2**, the ECC engine and memory interface can be coupled together as one component, e.g., "ECC ENGINE/MEMORY I/F" **222**. Reference herein collectively to an ECC engine/memory I/F does not exclude embodiments having error detection circuitry and memory interfaces as separate components. Likewise, reference herein to either component individually does not exclude embodiments where error detection circuitry and a memory interface are included as one component.

[0027] The back end CRC memory **216** can be configured to store CRC data calculated by the front end CRC engine **212**. The back end CRC memory **216** can be coupled to an ECC engine/memory OF **222** and be configured to receive and store CRC data from the memory interface portion of ECC engine/memory I/F **222**, e.g., CRC data that had previously been stored in the solid state memory devices **230**. The back end data buffer **218** can be configured to transfer a number of sectors of data received from an ECC engine/memory I/F **222** across the DMA module **214** to the front end CRC engine **212**. The front end CRC engine **212** can be coupled to the error detection circuitry portion of the ECC engine/memory I/F **222**, e.g., via the DMA module **214**. The back end CRC memory **216** can be configured to transfer CRC data corresponding to the number of sectors of data received from the ECC engine/memory I/F **222** across the DMA module **214** to the front end CRC engine **212**. In such an instance, the front end CRC engine **212** can be configured to calculate CRC data for a, e.g., each, sector of data received from the DMA module **214**. The front end CRC engine can then compare the calculated CRC data for the sector of data received from the DMA module **214** to the CRC data received from the back end CRC memory **216** via the DMA module **214**, e.g., to verify the integrity of the sector of data received from the DMA module **214** before the sector of data is transferred across the host interface **210** to a host system. Additional detail on the operation of a DMA module can be found in commonly assigned U.S. patent application Ser. No. 12/421,093, filed Apr. 9, 2009, now U.S. Pat. No. 8,055,816, issued Nov. 8, 2011 entitled "Memory Controllers Memory Systems, Solid State Drives and Methods for Processing a Number of Commands", which is incorporated by reference herein.

[0028] The ECC engine/memory I/F **222** can be coupled to the solid state memory devices **230**. In one or more embodi-

4

ments, the ECC engine/memory I/F **222** can be configured to append CRC data, e.g., CRC data received from a back end CRC memory **216**, to a corresponding sector of data. The ECC engine portion of the ECC engine/memory I/F **222** can be configured to detect errors in data, e.g., in a sector of data. In one or more embodiments, the ECC engine portion of the ECC engine/memory I/F **222** can be configured to detect and/or correct errors in data while the data is in a memory system such as memory system **120** in FIG. **1**. For example, the ECC engine portion of the ECC engine/memory I/F **222** can calculate error correction data, e.g., ECC data, for data alone and/or for data and appended error detection, e.g., CRC, data. In one or more embodiments, the ECC engine portion of the ECC engine/memory I/F **222** can be configured to calculate ECC data on a sector-by-sector basis. The ECC engine/memory I/F **222** can be configured to append ECC data to a corresponding sector of data. In one or more embodiments, the ECC engine portion of the ECC engine/memory I/F **222** can be configured to calculate ECC data for a corresponding sector of data along with appended CRC data, e.g., CRC data calculated by the front end CRC engine **212**. During read operations, the ECC engine portion of the ECC engine/ memory I/F **222** can be configured to correct errors in a number of sectors of data while the number of sectors of data are in the back end data buffer **218**. The ECC engine portion of the ECC engine/memory I/F **222** can be configured to correct errors in the CRC data while the CRC data is in the back end CRC memory **216**. Embodiments are not limited to correcting errors in these particular locations.

[0029] In one or more embodiments, the memory controller **201** can be configured to transfer a number of sectors of data, corresponding calculated CRC data, and corresponding ECC data, or a different combination thereof, across the ECC engine/memory I/F **222**, e.g., to the solid state memory devices **230** for a write operation. That is, the controller can be configured to store a sector of data, corresponding CRC data, and corresponding ECC data in a location in the solid state memory devices **230**.

[0030] One or more embodiments can include the control- ler **201** receiving a number of sectors of data from a host system **202** with a link layer **207** and/or a transport layer **209** of a host interface **210**, e.g., in conjunction with a write operation. The number of sectors of data can be transferred through a host interface **210** to a front end CRC engine **212** and a DMA module **214**. In one or more embodiments, the number of sectors of data can be received in parallel with the front end CRC engine **212** and the DMA module **214**. The front end CRC engine **212** can calculate CRC data corre- sponding to each of the number of sectors of data, e.g., each sector of data can correspond to unique CRC data. DMA module **214** can transfer the CRC data from CRC engine **212** to back end CRC memory **216**. The front end CRC engine **212** can be coupled to the back end CRC memory **216**, e.g., via the DMA module **214**. In one or more embodiments, the back end CRC memory **216** can store the CRC data. The back end CRC memory **216** can store more than one unique CRC data where each unique CRC data can correspond to a particular sector of data. The DMA module **214** can transfer the number of sec- tors of data to the back end data buffer **218**. The ECC engine/ memory I/F **222** can transfer the number of sectors of data from the back end data buffer **218** and the corresponding CRC data from back end CRC memory **216**, calculate unique ECC data for each of the number of sectors of data and the corre- sponding CRC data, append the ECC data to the correspond-

ing sector of data, and store the sector of data, CRC data, and ECC data in the one or more solid state memory devices **230**.

[0031] As noted above, CRC data can be calculated on the contents of a frame information structure (FIS) in dword (32 bit) quantities, among other CRC computational methods. However, a data payload received from a host system **202** may include a number of sectors, e.g., 512 byte portions, of a data payload. According to some previous approaches, CRC data may be calculated for the data payload as a whole rather than for each sector of data included in the data payload. That is, even if CRC data propagates past the link layer **207** or the transport layer **209**, the CRC data may not be useful for a particular sector of data at least partially because data may be transferred to and/or from a memory system, e.g., memory system **120** in FIG. **1**, in quantities different from the quanti- ties in which it is written to and/or read from a memory device, e.g., memory device **130-1** in FIG. **1**. Thus, using the same CRC data for a data payload as for a number of sectors of data associated with a read and/or write operation, even for the same data, may not provide data integrity for the number of sectors of data.

[0032] In contrast, according to one or more embodiments of the present disclosure CRC data is calculated for each sector of data on a sector by sector basis, e.g., unique CRC data can be calculated for each sector of data. For example, if a data payload comprises 2048 bytes, CRC data can be cal- culated for each of four 512 byte sectors of data. Data integ- rity can be provided for the number of sectors from the point it is received by the controller **201** from the host system **202** at the front end **204**, e.g., at the host interface **210**, to the point where it is transferred from the back end **206** to the solid state memory devices **230**. Providing data integrity can include maintaining the same CRC data for a sector of data from the host interface **210** to the ECC engine/memory I/F **222** and/or to the solid state memory devices **230**.

[0033] In one or more embodiments, ECC engine/memory I/F **222** can receive a number of sectors of data, their corre- sponding first CRC data, and ECC data from one or more solid state memory devices **230**, e.g., in conjunction with a read operation. The ECC engine/memory I/F **222** can store the number of sectors of data in the back end data buffer **218** and store their corresponding CRC data in back end CRC memory **216**. The number of sectors of data and their corre- sponding first CRC data and ECC data can be error checked with an ECC engine portion of an ECC engine/memory I/F **222**. One or more errors identified by the ECC engine portion of the ECC engine/memory I/F **222** in the CRC data can be corrected while the CRC data is stored in the back end CRC memory **216**. One or more errors identified by the ECC engine portion of the ECC engine/memory I/F **222** in the number of sectors of data can be corrected while the number of sectors of data are buffered in back end data buffer **218**. After error correction, the DMA module **214** can receive and transfer the number of sectors of data and their corresponding CRC data to the front end CRC engine **212**. The front end CRC engine **212** can calculate second CRC data for the num- ber of transferred sectors of data and compare the first CRC data with the corresponding second CRC data to verify the integrity of the number of sectors of data. After comparing the first CRC data with the second CRC data, the number of sectors of data can be transferred across a transport layer **209** of host interface **210**, e.g., to a host system **202** that requested data, e.g., the number of read sectors of data. In one or more embodiments, the number of sectors of data can be trans-

ferred across the host interface **210** without either the first CRC data or the second CRC data. In one or more embodiments, the ECC engine/memory I/F **222** can remove ECC data from the number of sectors of data and transfer the number of sectors of data and first CRC data in parallel to a front end CRC engine **212**.

[0034] According to one or more embodiments of the present disclosure, such as the read operation described above, data integrity can be provided for the number of sectors of data. That is, data integrity can be provided for the number of sectors of data from the point is the number of sectors are received by the back end **206** of controller **201** from the solid state memory devices **230**, e.g., at ECC engine/memory I/F **222**, to the point where the number of sectors are transferred across the host interface **210**, e.g., from the front end **204** to a host system **202**. Providing data integrity can include maintaining the same CRC data from the solid state memory devices **230** to the host interface **210**.

[0035] FIG. 3 is a functional block diagram of a system **300** including a memory controller **301** in accordance with one or more embodiments of the present disclosure. Controller **301** can be analogous to controller **101** illustrated in FIG. **1**. In one or more embodiments, controller **301** can be a component of a memory system, such as an SSD. It will be appreciated by those skilled in the art that additional circuitry and components can be provided beyond those illustrated in FIG. **3**, and that the controller detail of FIG. **3** has been reduced to facilitate ease of illustration.

[0036] The embodiment illustrated in FIG. **3** includes components that can be analogous to those illustrated in FIG. **2** including host interface **310** with physical layer **305**, link layer **307**, and transport layer **309**, as well as front end error detection circuitry, e.g., CRC engine **312**, and data transfer circuitry, e.g., DMA module **314**. However, in one or more embodiments, the host interface **310** can be coupled to the DMA module via a front end host buffer **324**, e.g., FIFO **324** as shown, and encryption circuitry, e.g., advanced encryption standard (AES) engine **326**. Although illustrated as "AES ENGINE" **326**, other forms of encryption may be used with one or more embodiments of the present disclosure. FIFO **324** can be configured to buffer a DMA payload between a host interface **310** and an AES engine **326**. Although not illustrated in FIG. **3**, the AES engine **326** can include or be coupled to an AES buffer separate from the FIFO **324**.

[0037] As illustrated in FIG. **3**, the AES engine **326** may be configured to receive a number of sectors of data, e.g., a data payload (DMA payload) such as a data stream derived from a number of data packets received by the controller **301**. Prior to encryption, the number of sectors of data can be plaintext (P.T.). The AES engine **326** may be arranged and configured to process the number of sectors of data optionally to provide an encrypted output, e.g., ciphertext (C.T.) to the DMA module **314**. In one or more embodiments the AES engine **326** can process the number of sectors of data optionally, meaning that the AES engine can encrypt the number of sectors of data from plaintext to ciphertext, or the AES engine **326** can transfer the number of sectors of data to the DMA module **314** without encryption, e.g., in plaintext.

[0038] In one or more embodiments, a number of sectors of data received from a host system via host interface **310** can be received in parallel with the front end CRC engine **312** and the front end host buffer **324**, e.g., FIFO **324** as shown. The number of sectors of data can be transferred from the host buffer **324** to the AES engine **326**, where the number of

sectors of data can be encrypted from plaintext to ciphertext. After encryption, the number of sectors of data can be transferred from the AES engine **326** to a DMA module **314** for further processing. Additional examples of the operation of encryption engines are described in more detail in commonly assigned U.S. patent application Ser. No. 12/333,822, entitled "Parallel Encryption/Decryption", having at least one common inventor, which issued as U.S. Pat. No. 8,355,499 on Jan. 15, 2013 and is incorporated by reference herein.

[0039] The DMA module **314** effectively couples the front end **304** circuitry to the back end **306** circuitry. In one or more embodiments, the back end portion **306** of controller **301** can include more than one back end channel. In the embodiment illustrated in FIG. **3**, the controller **301** includes a number of back end channels **319-1**, . . . , **319-N**. Each back end channel **319-1**, . . . , **319-N** can include a channel processor, **332-1**, . . . , **332-N**, and channel memory, **334-1**, . . . , **334-N**. The back end channel processors **332-1**, . . . , **332-N** and memories **334-1**, . . . , **334-N**, can be coupled to the DMA module **314** and to the back end channel DMA modules **328-1**, . . . , **328-N**. Each back end channel **319-1**, . . . , **319-N** can include back end channel data transfer circuitry, e.g., channel DMA modules **328-1**, . . . , **328-N**, which can be coupled to the DMA module **314**. The DMA module **314** can be configured to direct commands associated with a DMA payload to a respective back end channel processor, e.g., back end channel processor **332-1**, and to direct data associated with a DMA payload to a respective back end channel DMA module, e.g., back end channel DMA module **328-1**. Example operations of channel processors are described in more detail in commonly assigned U.S. patent application Ser. No. 12/351,206, entitled "Modifying Commands", having at least one common inventor, which issued as U.S. Pat. No. 8,078,848 on Dec. 13, 2011 and is incorporated by reference herein.

[0040] The back end portion **306** of the controller **301** can include components analogous to those illustrated in FIG. **2**. For example, each back end channel **319-1**, . . . , **319-N** can include a back end data buffer **318-1**, . . . , **318-N**, back end error detection memory, e.g., CRC memory **316-1**, . . . , **316-N**, and ECC engines/memory interfaces **322-1**, . . . , **322-N**. In one or more embodiments, and as illustrated in the embodiment of FIG. **3**, the ECC engine and memory interface can be coupled together as one component, e.g., "ECC ENGINE/MEMORY I/F" **322**. In one or more embodiments, the ECC engine and memory interface can be separate components. As illustrated in FIG. **3**, the back end data buffers **318-1**, . . . , **318-N** can be coupled between the back end channel DMA modules **328-1**, . . . , **328-N** and the ECC engines/memory interfaces **322-1**, . . . , **322-N**. Furthermore, the back end CRC memory **316-1**, . . . , **316-N** can be coupled between the DMA module **314** and the ECC engines/memory interfaces **322-1**, . . . , **322-N**. As shown in FIG. **3**, a particular memory device, chip, array, etc., can correspond to a particular channel. For example, the solid state memory device(s) **330-1** can correspond to channel **319-1**.

[0041] FIG. 4 is a functional block diagram of a system **400** including a memory controller **401** in accordance with one or more embodiments of the present disclosure. Controller **401** can be analogous to controller **101** illustrated in FIG. **1**. In one or more embodiments, controller **401** can be a component of a memory system, such as an SSD. It will be appreciated by those skilled in the art that additional circuitry and compo-

nents can be provided beyond those illustrated in FIG. 4, and that the detail of controller 401 in FIG. 4 has been reduced to facilitate ease of illustration.

[0042] The embodiment illustrated in FIG. 4 includes components that can be analogous to those illustrated in FIG. 3 including front end 404 components of controller 401 such as host interface 410 including physical layer 405, link layer 407, and transport layer 409, as well as front end error detection circuitry, e.g., CRC engine 412-F, and data transfer circuitry, e.g., DMA module 414. The controller 401 also includes a number of back end 406 components that can be analogous to those illustrated in FIG. 3 such as back end channels 419-1, . . . , 419-N, including back end channel processors 432-1, . . . , 432-N and memory 434-1, . . . , 434-N, back end channel data transfer circuitry, e.g., channel DMA modules 428-1, . . . , 428-N, back end data buffers 418-1, . . . , 418-N, error detection circuitry, e.g., ECC engines/memory interfaces 422-1, . . . , 422-N, and back end error detection memory, e.g., CRC memory 416-1, . . . , 416-N. In one or more embodiments, and as illustrated in FIG. 4, the ECC engine and memory interface can be coupled together as one component, e.g., "ECC ENGINE/MEMORY I/F" 422. In one or more embodiments, the ECC engine and memory interface can be separate components. As illustrated in FIG. 4, each channel 419-1, . . . , 419-N can be coupled to one or more solid state memory devices 430-1, . . . , 430-N by the ECC engines/ memory interfaces 422-1, . . . , 422-N.

[0043] However, in one or more embodiments, back end channels 419-1, . . . , 419-N can include a back end error detection circuitry, e.g., back end CRC engine 412-B1, . . . , 412-BN coupled to the back end CRC memory 416-1, . . . , 416-N. A back end CRC engine 412-B1, . . . , 412-BN can be configured to calculate error detection data, e.g., "second" CRC data, for corresponding data, e.g., a corresponding sector of data. In one or more embodiments, the corresponding data can be at least a portion of a DMA payload. The front end CRC engine 412-F can calculate "first" CRC data for a sector of data. A back end CRC engine 412-B1, . . . , 412-BN can be configured to compare the first CRC data with the second CRC data to check and/or verify the integrity of the sector of data. For example, the DMA module 414 can be coupled to the front end CRC engine 412-F and to the back end CRC memory 416-1, . . . , 416-N. The DMA module 414 can be configured to transfer the first CRC data to a back end CRC memory 416-1, . . . , 416-N. Then a back end CRC engine 412-B1, . . . , 412-BN can compare the first CRC data received from a back end CRC memory 416-1, . . . , 416-N with the calculated second CRC data. Although a back end CRC engine 412-B1, . . . , 412-BN is disclosed with respect to FIG. 4, which illustrates a controller 401 including multiple back end channels 419-1, . . . , 419-N, embodiments are not so limited. For example, one or more embodiments of the present disclosure can include a memory controller with a single back end channel, e.g., as illustrated with respect to the back end 206 in FIG. 2, that includes a back end CRC engine.

[0044] A back end CRC engine 412-B1, . . . , 412-BN can be coupled to an ECC engine/memory interface 422-1, . . . , 422-N. In one or more embodiments, the back end CRC engine 412-B1, . . . , 412-BN can be directly coupled to the ECC engine/memory I/F 422-1, . . . , 422-N. The ECC engine portion of an ECC engine/memory I/F 422-1, . . . , 422-N can be configured to calculate ECC data for a sector of data, e.g., a portion of a DMA payload received from a back end data buffer 418-1, . . . , 418-N. In one or more embodiments, the

controller 401 can be configured to transfer the sector of data and corresponding ECC data across the ECC engine/memory I/F 422-1, . . . , 422-N without the first or the second corresponding CRC data. That is, in one or more embodiments, the sector of data and corresponding ECC data can be stored in the solid state memory devices 430-1, . . . , 430-N without storing the corresponding CRC data in the solid state memory devices 430-1, . . . , 430-N. In one or more embodiments, the controller 401 can be configured to store the sector of data and the corresponding ECC data without the corresponding CRC data in a location in the solid state memory devices 430-1, . . . , 430-N when a back end CRC engine 412-B1, . . . , 412-BN verifies the integrity of the sector of data. That is, if a back end CRC engine 412-B 1, . . . , 412-BN does not verify the integrity of the sector of data, then the controller 401, in one or more embodiments, may not store the sector of data in the solid state memory devices 430-1, . . . , 430-N.

[0045] One or more embodiments can include receiving a number of sectors of data with the front end CRC engine 412-F from a link layer 407 and/or a transport layer 409 of a host interface 410, e.g., in conjunction with a write operation. The front end CRC engine 412-F can be configured to calculate first CRC data for each sector of data. Accordingly, for the write operation, data integrity of the number of sectors of data can be maintained on the front end 404 of the controller 401 at least in part due to CRC data being calculated for the number of sectors of data before, or separate from, the performance of other operations on the number of sectors of data. The first CRC data can be transferred to the DMA module 414.

[0046] The DMA module 414 can transfer the number of sectors of data to a back end channel DMA module 428-1, ... 428-N. In one or more embodiments, the DMA module 414 can transfer the number of sectors of data to a back end channel 419-1, . . . , 419-N that corresponds to a memory address associated with the number of sectors of data, e.g., the channel coupled to the solid state memory device corresponding to the memory address. For example, the DMA module can transfer the number of sectors of data to back end channel DMA module 428-1 on channel 419-1 when a memory address associated with the number of sectors of data corresponds to a particular solid state memory device, e.g., 430-1. The number of sectors of data can be transferred from the back end channel DMA module 428-1, . . . , 428-N to a back end data buffer 418-1, . . . , 418-N. The back end data buffer 418-1, . . . , 418-N can buffer the sector of data received from the back end channel DMA module 428-1, . . . , 428-N during a write operation before the number of sectors of data are transferred to the ECC engine/memory interface 422-1, . . . , 422-N and/or the back end CRC engine 412-B1, . . . , 412-BN. In one or more embodiments, the number of sectors of data can be transferred in parallel from a back end data buffer 418-1, . . . , 418-N to the back end CRC engine 412-B1, . . . , 412-BN and to an ECC engine/memory I/F 422-1, . . . , 422-N.

[0047] The DMA module 414 can transfer the first CRC data to back end CRC memory 416-1, . . . , 416-N. In one or more embodiments, the DMA module 414 can transfer the first CRC data to a back end channel that corresponds to a memory address associated with the sector of data for which the first CRC data was calculated. The first CRC data can be stored in back end CRC memory 416-1, . . . , 416-N before the back end CRC engine 412-B1, . . . , 412-BN calculates second CRC data. A back end CRC engine 412-B1, . . . , 412-BN can calculate second CRC data for the sector of data and compare

the first CRC data with the second CRC data. Subsequent to the comparison, the sector of data can be stored in one or more solid state memory devices **430-1**, . . . , **430-N**. In one or more embodiments, an ECC engine portion of an ECC engine/memory I/F **422-1**, . . . , **422-N** can calculate ECC data for the sector of data after the comparison of first CRC data with second CRC data, but prior to the sector of data being stored in a solid state memory device **430-1**, . . . , **430-N**. The ECC data can be appended to the sector of data before it is stored in a solid state memory device **430-1**, . . . , **430-N**, such that the sector of data and the ECC data are stored, but neither the first nor the second CRC data are stored in a solid state memory device **430-1**, . . . , **430-N**. Such embodiments can be beneficial at least partially due to a reduction in the amount of information stored in the solid state memory devices **430-1**, . . . , **430-N**, e.g., when CRC data is not stored therein.

[0048] In one or more embodiments, the number of sectors of data are only transferred from the ECC engine/memory I/F **422-1**, . . . , **422-N** to the solid state memory device **430-1**, . . . , **430-N** if the back end CRC engine **412-B1**, . . . , **412-BN** verifies the integrity of the number of sectors of data, e.g., if the first CRC data matches the second CRC data. Accordingly, for the write operation, data integrity of the number of sectors of data can be maintained on the back end **406** of the controller **401** at least in part due to second CRC data being calculated for the number of sectors of data after processing by the DMA module **414** and back end channel DMA module **428-1**, . . . , **428-N** and compared to the first CRC data, which was calculated before, or separate from, the performance of other operations on the number of sectors of data. Thus, one or more embodiments of the present disclosure can provide data integrity for the number of sectors of data on the controller **401**.

[0049] One or more memory system operation embodiments can include a read operation including reading a number of sectors of data and corresponding ECC data from one or more solid state memory devices **430-1**, . . . , **430-N**. The number of sectors of data can be error checked with an ECC engine portion of an ECC engine/memory IN **422-1**, . . . , **422-N**. After error checking with the ECC engine portion of the ECC engine/memory I/F **422-1**, . . . , **422-N**, the ECC data can be removed from the number of sectors of data. A back end CRC engine **412-B1**, . . . , **412-BN** can calculate CRC data for each of the number of sectors of data. The number of sectors of data and the CRC data can be transferred in parallel, e.g., from the host buffer **424** and DMA module **414** respectively, to the front end CRC engine **412-F**. The front end CRC engine **412-F** can calculate CRC data for the each of the number of sectors of data and compare it to the CRC data calculated by the back end CRC engine to check the integrity of the number of sectors of data. The number of sectors of data can be transferred across a transport layer **409** of the host interface **410** after the comparison.

[0050] Accordingly, for the read operation, data integrity of a number of sectors of data can be maintained from the back end **406** of the controller **401** to the front end **404** at least in part due to calculating CRC data for each of the number of sectors of data after is the number of sectors of data are transferred from an ECC engine/memory I/F **422-1**, . . . , **422-N**. Furthermore, transferring previously calculated CRC data to the front end **404** of the controller **401** for comparison with newly calculated CRC data for a particular sector of data can help to verify that the integrity of the particular sector of data has been maintained. Thus, one or more embodiments of the present disclosure can provide data integrity for a number of sectors of data on the controller **401**.

[0051] The present disclosure includes methods, devices, and systems for data integrity in memory controllers. One memory controller embodiment includes a host interface and first error detection circuitry coupled to the host interface. The memory controller can include a memory interface and second error detection circuitry coupled to the memory interface. The first error detection circuitry can be configured to calculate error detection data for data received from the host interface and to check the integrity of data transmitted to the host interface. The second error detection circuitry can be configured to calculate error correction data for data and first error correction data transmitted to the memory interface and to check integrity of data and first error correction data received from the memory interface.

[0052] It will be understood that when an element is referred to as being "on," "connected to" or "coupled with" another element, it can be directly on, connected, or coupled with the other element or intervening elements may be present. In contrast, when an element is referred to as being "directly on," "directly connected to" or "directly coupled with" another element, there are no intervening elements or layers present. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

[0053] It will be understood that, although the terms first, second, etc. may be used herein to describe various elements and that these elements should not be limited by these terms. These terms are only used to distinguish one element from another element. Thus, a first element could be termed a second element without departing from the teachings of the present disclosure.

[0054] Although specific embodiments have been illustrated and described herein, those of ordinary skill in the art will appreciate that an arrangement calculated to achieve the same results can be substituted for the specific embodiments shown. This disclosure is intended to cover adaptations or variations of one or more embodiments of the present disclosure. It is to be understood that the above description has been made in an illustrative fashion, and not a restrictive one. Combination of the above embodiments, and other embodiments not specifically described herein will be apparent to those of skill in the art upon reviewing the above description. The scope of the one or more embodiments of the present disclosure includes other applications in which the above structures and methods are used. Therefore, the scope of one or more embodiments of the present disclosure should be determined with reference to the appended claims, along with the full range of equivalents to which such claims are entitled.

[0055] In the foregoing Detailed Description, some features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the disclosed embodiments of the present disclosure have to use more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus, the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separate embodiment.

1. A memory system, comprising: one or more solid state memory devices; and a memory controller coupled to the one or more solid state memory devices by a memory interface,

wherein the memory controller includes: a host interface; first error detection circuitry coupled to the host interface; and second error detection circuitry coupled to the memory interface; wherein the first and the second error detection circuitries are configured to detect one or more errors in data transferred across the host interface while the data is in the memory system.

* * * * *