EFFICIENT STORAGE OF ERROR CORRECTING CODE INFORMATION

Abstract: Multiple independent point-to-point memory channels are operated, by at least one controller, in parallel to form a wider memory channel. The memory components on these point-to-point channels include the ability to connect to multiple (e.g., 2) instances of these independent memory channels. The controller operates multiple instances of the wider channels with the memory components configured in a clamshell mode. A single memory component is also operated in clamshell mode to provide error correction code information, independently of the other wider channels, to multiple instances of the wider memory channel.
Declarations under Rule 4.17:
— as to applicant’s entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:
— with international search report (Art. 21(3))
EFFICIENT STORAGE OF ERROR CORRECTING CODE INFORMATION

BRIEF DESCRIPTION OF THE DRAWINGS

[0001] Figure 1A is a block diagram illustrating a memory system.

[0002] Figure 1B is a block diagram illustrating memory data channel connections and a memory system.

[0003] Figure 1C is a block diagram illustrating memory data channel connections and a memory system with multiple controllers.

[0004] Figure 2 is an illustration of a 64-bit wide data channel burst with error correction information.

[0005] Figure 3 is a block diagram illustrating memory data channel connections and a memory system.

[0006] Figure 4 is an illustration of a 32-bit wide data channel burst with error correction information.

[0007] Figure 5 is a block diagram illustrating a low-power double-data rate (LPDDR) memory system with error correction information.

[0008] Figure 6 a flowchart illustrating a method of operating a memory controller.

[0009] Figure 7 is a block diagram illustrating a memory system with data devices operating in non-clamshell mode.

[0010] Figure 8 is a block diagram of a processing system.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0011] In an embodiment, multiple independent point-to-point memory channels (e.g., 16-bit memory channels) are operated, by a controller, in parallel to form a wider memory channel (e.g., a 64-bit data channel.) The memory components on these point-to-point channels include the ability to connect to multiple (e.g., 2) instances of these independent
memory channels. In addition, pairs of these memory components may be affixed to opposite sides of a substrate (e.g., printed circuit board) and receive the same command/address signal lines from each of the memory channels. The data signal lines of these pairs are routed to the controller separately such that each memory component of the pair, when so configured, supplies one-half (½) of the data width of the memory channel (e.g., for a 16-bit memory channel, each component of the pair supplies 8 bits.) Configuring the pair to operate in this manner is known as ‘clamshell’ mode.

[0012] A single controller operates multiple instances of the wider channels (e.g., for increased memory bandwidth) with the memory components configured in the clamshell mode. A single memory component is also operated in clamshell mode to provide error correction code information, independently of the other wider channels, to multiple instances of the wider memory channel.

[0013] The single memory component uses the ability to connect to multiple (e.g., 2) instances of these independent memory channels to independently provide the error correction information to each of the wider channels. Thus, for example, two independent 72-bit channels (i.e., 64-bits of data and 8-bits of error correction code) may be provided using just nine (9) memory components—four (4) components on each of the two channels are for providing a total of 64-bits of data (in parallel) with the single memory component being ‘shared’ by the channels such that the single memory component provides 8 bits of error correction information to each of the channels.

[0014] Figure 1A is a block diagram illustrating a memory system. In Figure 1A, memory system 100 comprises memory controller 110, memory components 121-129, memory access channels 151-154, 155-1, 155-2, and 156-159. Controller 110 includes wide memory channel #1 interface 131, wide memory channel #2 interface 132, error correcting code (ECC) circuitry for channel #1 141 and ECC circuitry for channel #2 142. Wide
memory channel #1 interface 131 includes memory access interfaces 111a, 112b, 113a, 114b, and 115a. Wide memory channel #2 interface 132 includes memory access interface 115b, 116a, 117b, H8a, and H9b.

[0015] Memory components 121-129 each have at least two independent memory access interfaces 121a-129a and 121b-129b, respectively. In other words, memory component 121 has at least two independent memory access interfaces 121a and 121b. Likewise, memory component 122 has at least two independent memory access interfaces 122a and 122b, and so on for the rest of memory components 123-129. Memory access interfaces 121a-129a and 121b-129b for a given memory component 121-129 are independent in that they can operate independently of each other to provide command, address, and data transfer functions to the respective memory access channels 151-154, 155-1, 155-2, and 156-159 that they are operatively coupled to. However, the memory access interfaces 121a-129a and 121b-129b on a respective memory component 121-129 may share at least one common signal, such as a clock and/or strobe signal(s).

[0016] Controller 110 and the memory components 121-129 may be integrated circuit type devices, such as are commonly referred to as a “chip”. Memory components 121-129 may be standalone devices, or may be one or more components of a memory module. A memory controller, such as controller 110, manages the flow of data going to and from memory devices and/or memory modules. A memory controller can be a separate, standalone chip, or integrated into another chip. For example, a memory controller may be included on a single die with a microprocessor, or included as part of a more complex integrated circuit system such as a block of a system on a chip (SOC). Memory components 121-129 can include a dynamic random access memory (DRAM) core or other type of memory cores, for example, static random access memory (SRAM) cores, or non-volatile memory cores such as
flash. In an embodiment, memory components 121-129 conform to the double data rate type six synchronous graphics dynamic random-access memory (GDDR6) specification.

[0017] As used herein, the terms related to error correction (e.g., ECC, ECC code, check bits, etc.) refer broadly to the generation and/or addition of redundancy (i.e., some extra data) to stored data, which can be used (e.g., by ECC circuitry 141 and/or ECC circuitry 142) to check the consistency of the retrieved data, and/or to recover data that has been determined to be corrupted. Thus, at least the terms ECC information and ECC code should be understood to include redundant information that merely checks for corruption (e.g., parity so that retrieval may be retried) as well as more complex error detection/correction schemes such as checksums, cyclic redundancy checks (CRC—e.g., CRC-8), and forward error correction schemes.

[0018] Controller 110 is operatively coupled to the memory access interfaces l21a-l29a and l21b-l29b on a respective memory component 121-129 via respective memory access channels 151-154, 155-1, 155-2, and 156-159 to communicate commands, addresses, and data with memory devices 121-129. Memory access channels 151-154 and 156-159 respectively comprise first bidirectional data signals (e.g., DQ[0:7]) l51a-l54a and l56a-l59a, second bidirectional data signals (e.g., DQ[8:15]) l51b-l54b and l56b-l59b, and command/address (CA or C/A) signals l51c-154c and 156c-159c. Memory access channels 155-1 and 155-2 respectively comprise bidirectional data signals (e.g., DQ[0:7]) 155-1a and 155-2a, and command/address (CA or C/A) signals 155-1c and 155-2c.

[0019] Controller 110 can operate, via memory access channels 151-154, 155-1, 155-2, and 156-159, each respective independent portion (e.g., memory access interfaces 121a-129a and 121b-129b) of memory components 121-129 independently. Each of memory access interfaces 111a, 112b, 113a, 114b, 115a, 115b, 116a, 117b, 118a, and 119b, and thus each of
memory access channels 151-154, 155-1, 155-2, and 156-159 include command, address, and
data transfer functions.

[0020] Each of memory access interfaces 111a, H2b, H3a, H4b, H5a, H5b, H6a,
117b, 118a, and 119b may operate the command, address, and data transfer functions of their
respective memory access channel 151-154, 155-1, 155-2, and 156-159 independently of the
other memory access channels 151-154, 155-1, 155-2, and 156-159. Memory access
interfaces 111a, 112b, 113a, 114b, 115a, 115b, 116a, 117b, 118a, and 119b may be operated
independently of each other because the memory access interfaces 12la-129a and 12lb-129b
can operate independently of the other memory access interfaces 12la-129a and 12lb-129b
and other memory components 121-129.

[0021] In an embodiment, memory access interfaces 111a, 112b, 113a, 114b, and 115a
are effectively operated in ‘lockstep’ (i.e., receiving the same command and address
information for what is effectively the same access cycles) to form wide memory channel #1
interface 131. Memory access interfaces 115b, 116a, 117b, 118a, and 119b are effectively
operated in lockstep to form wide memory channel #2 interface 132.

[0022] Wide memory channel #1 interface 131 is operatively coupled to ECC circuitry
141. Wide memory channel #2 interface 132 is operatively coupled to ECC circuitry 142.
The data communicated via wide memory channel #1 interface 131 is provided to ECC
circuitry 141. The data communicated via wide memory channel #1 interface 131 may be
corrected by ECC circuitry 141. The check bits communicated via wide memory channel #1
interface 131 are either generated by, or checked by, ECC circuitry 141. In an embodiment,
ECC circuitry 141 can correct a single bit error in the data and check bits. ECC circuitry 141
may also be able to detect two or more bit errors in the data and check bits. Data, corrected
data, and/or check bits processed by wide memory channel #1 interface 131 and/or ECC
circuitry 141 may be output or otherwise used by controller 110.
The data communicated via wide memory channel #2 interface 132 is provided to ECC circuitry 142. The data communicated via wide memory channel #2 interface 132 may be corrected by ECC circuitry 142. The check bits communicated via wide memory channel #2 interface 132 are either generated by, or checked by, ECC circuitry 142. In an embodiment, ECC circuitry 142 can correct a single bit error in the data and check bits. ECC circuitry 142 may also be able to detect two or more bit errors in the data and check bits. Data, corrected data, and/or check bits processed by wide memory channel #2 interface 132 and/or ECC circuitry 142 may be output or otherwise used by controller 110. In an embodiment, the ECC scheme (code) implemented by ECC circuitry 141 and ECC circuitry 142 are the same. In an embodiment, the ECC scheme (code) implemented by ECC circuitry 141 and ECC circuitry 142 are not the same.

Memory components 121-124 and 126-129 (i.e., not memory component 125) may be paired. When paired, memory components 121-124 and 126-129 may be affixed to opposite sides of a substrate (e.g., printed circuit board). These pairs are operatively coupled (e.g., connected) to receive the same command/address signal lines from a respective pair of memory access channels 151-154, and 156-159. In other words, memory component 121 (via memory access interface 121a) and memory component 122 (via memory access interface 122a) are both connected to, command/address signal lines 151c of memory access channel 151 (and thus connected to memory access interface 111a.) Memory component 121 (via memory access interface 121a) and memory component 122 (via memory access interface 122a) are also both connected to, command/address signal lines 152c of memory access channel 152 (and thus connected to memory access interface 112b.) Memory components 123 and 124 are paired and connected in the same manner to each other and memory access interfaces 113a and 114b - and so on such that memory components 126 and 127 are paired and connected, as are memory components 128 and 129.
The data signal lines of these pairs are routed to the controller separately such that each memory component of the pair, when so configured, supplies a fraction (e.g., 1/2) of the data width of the memory channel (e.g., for a 16-bit memory channel, each component of the pair supplies 8 bits.) Thus, memory component 121 (via memory access interface 121a) is connected to memory access interface 111a using first data signal lines 151a of memory access channel 151. Memory component 122 (via memory access interface 122a) is connected to memory access interface 111a using second data signal lines 151b of memory access channel 151. Memory component 121 (via memory access interface 121b) is also connected to memory access interface 112b using second data signal lines 152b of memory access channel 152. Memory component 122 (via memory access interface 122b) is connected to memory access interface 112b using first data signal lines 152a of memory access channel 151. The first and second data signals of memory components 123 and 124 are connected in the same manner to memory access interfaces 113a and 114b - and so on such that the first and second data signals of memory components 126 and 127 are connected to memory access interfaces 116a and 117b, and the first and second data signals of memory components 128 and 129 are connected to memory access interfaces 118a and 119b. Configuring a pair to operate in this manner is known as 'clamshell' mode.

The command/address signal lines 155-la of memory access channel 155-1 are connected to memory access interface 115a and memory access interface 125a of memory component 125. The data signal lines 155-la of memory access channel 155-1 are connected to memory access interface 115a and memory access interface 125a of memory component 125. The command/address signal lines 155-2c of memory access channel 155-2 are connected to memory access interface 115b and memory access interface 115b of memory component 125. The data signal lines 155-2a of memory access channel 155-2 are connected
to memory access interface 115b and memory access interface 125b of memory component 125.

[0027] The clamshell configuration of memory components 121-129 is further illustrated in Figure 1B. For clarity, in Figure 1B, the command/address portions 15lc-154c, 155-lc, 155-2c, and 156c-l59c of memory access channels 151-154, 155-1, 155-2, and 156-159 have been omitted. Figure 1B illustrates that: memory access interface 121a of memory component 121 communicates data B0 (e.g., a byte) with memory access interface 111a; memory access interface 122a of memory component 122 communicates data B1 with memory access interface 111a; memory access interface 121b of memory component 121 communicates data B2 with memory access interface 112b; memory access interface 122b of memory component 122 communicates data B3 with memory access interface 112b. Paired memory components 123-124, 126-127, and 128-129 are connected in the same manner. Thus, for wide memory channel #1 interface 131: memory access interfaces 123a and 124a communicate data B4 and B5, respectively with memory access interface 113a; and, memory access interfaces 123b and 124b communicate data B6 and B7, respectively with memory access interface 114b. For wide memory channel #2 interface 132: memory access interfaces 126a and 127a communicate data B0 and B1, respectively with memory access interface 116a; memory access interfaces 126b and 127b communicate data B2 and B3, respectively with memory access interface 117b; memory access interfaces 128a and 129a communicate data B4 and B5, respectively with memory access interface 118a; memory access interfaces 128b and 129b communicate data B6 and B7, respectively with memory access interface 119b.

[0028] Memory component 125 is also operated in clamshell mode to provide respective error correction code information to wide memory channel #1 interface 131 and wide memory channel #2 interface 132. The ability of memory component 125 to independently
be operated by multiple (i.e., 2) instances of wide memory channel interfaces 131-132 (i.e.,
via memory access interface 125a which is independent of memory access interface 125b)
allows memory component 125 to independently provide the error correcting information
(EC1 and EC2, respectively) to each of these wider channels 131-132. Thus, two independent
72-bit channels (i.e., 64-bits of data and 8-bits of error correction code) may be provided
using just nine (9) memory components. Four memory components 121-124 on wide
memory channel #1 interface 131 provide 64-bits of data and one-half (½) of memory
component 125 provides 8 bits of error correction code data for a total of 72 bits. Likewise,
four memory components 126-129 on wide memory channel #2 interface 132 provide 64-bits
of data and the other one-half (½) of memory component 125 provides 8 bits of error
correction code data for a second total of 72 bits.

[0029] In Figures 1A and 1B, each memory component 121-129 has two (2) independent
memory access interfaces. However, it should be understood that the principles and
techniques described herein can be applied to memory components and/or controllers that
have more than two independent memory access interfaces.

[0030] In an embodiment, controller 100 includes a plurality of memory channel
interfaces (e.g., memory channel #1 interface 131 and memory channel #2 interface 132.)
These memory channel interfaces each include a respective plurality of memory access
interfaces (e.g., memory channel #1 interface 131 includes memory access interfaces 111a,
112b, 113a, 114b, and 115a, memory channel #2 interface 132 includes memory access
interface H5b, H6a, H7b, H8a, and H9b.) These memory access interfaces operate
independently of the other memory access interfaces.

[0031] The memory channel interfaces include at least a first group of interfaces (i.e.,
memory access interfaces 111a, 112b, 113a, and 114b) to access a first data word group, and
a second group (i.e., memory access interfaces 116a, 117b, 118a, and 119b) to access a
second data word group. The memory channel interfaces also include at least a first error
correction memory access interface (i.e., memory access interface 15a) and a second error
correction memory access interface (i.e., memory access interface 15b.) The first error
correction memory access interface and the second error correction memory access interface
access the same memory component (i.e., memory component 125). The first error
correction memory access interface (i.e., memory access interface 15a) accesses a first group
of check bits associated with the first data word group. The second error correction memory
access interface (i.e., memory access interface 15b) accesses a second group of check bits
associated with the second data word group.

[0032] The memory access interfaces (i.e., memory access interfaces 11a, 112b, 113a,
114b, H5a, H5b, H6a, H7b, H8a, and H9b) each include command, address, and data
transfer functions that operate independently of the command, address, and data transfer
functions of the other memory access interfaces. In an embodiment, all of the memory
components connected to the memory access interfaces are operated in clamshell mode.

[0033] Figure 2 is an illustration of a 64-bit wide data channel burst with error correction
information. The data channel burst illustrated in Figure 2 may be received or transmitted by
wide memory channel #1 interface 131 and/or wide memory channel #2 interface 132.

[0034] In Figure 2, data channel burst 200 has a burst length of 16 groups of data. Each
group of data has 64 data bits and 8 error correction code bits (i.e., check bits.) In an
embodiment, each set of check bits is associated (i.e., provides error detection and correction
for) with the corresponding 64 data bits in the same group of data. In another embodiment,
the check bits for each half (i.e., ECC span #0 and ECC span #1) of the data burst are
aggregated to provide error correction information for the corresponding half of the
respective data burst. In another embodiment, the check bits for the entire data burst are
aggregated to provide error correction information for the entire data burst.
In Figures 1A and 1B, a single controller 110 is illustrated as controlling both wide memory channel #1 interface 131 and wide memory channel #2 interface 132. However, it is contemplated that multiple memory controllers 110a and 110b may independently control wide memory channel #1 interface 131 and wide memory channel #2 interface 132. This is illustrated in Figure 1C by memory system 101 having controller 110a controlling wide memory channel #1 interface 131 and controller 110b controlling wide memory channel #2 interface 132. Controllers 110a and 110b may be on the same integrated circuit substrate.

Figure 3 is a block diagram illustrating memory data channel connections and a memory system. In Figure 3, memory system 300 comprises memory controller 310, memory components 321-325. Controller 310 includes wide memory channel #1 interface, and wide memory channel #2 interface. Wide memory channel #1 interface includes memory access interfaces 311a, 312b, and 315a. Wide memory channel #2 interface includes memory access interface 315b, 313a, and 314b.

Memory components 321-325 each have at least two independent memory access interfaces 321a-325a and 321b-325b, respectively. In other words, memory component 321 has at least two independent memory access interfaces 321a and 321b. Likewise, memory component 322 has at least two independent memory access interfaces 322a and 322b, and so on for the rest of memory components 323-325. Memory access interfaces 321a-325a and 321b-325b for a given memory component 321-325 are independent in that they can operate independently of each other to provide command, address, and data transfer functions to the respective memory access channel that they are operatively coupled to. However, the memory access interfaces 321a-325a and 321b-325b on a respective memory component 321-325 may share at least one common signal, such as a clock and/or strobe signal(s).

Memory components 321-325 can include a dynamic random access memory (DRAM) core.
or other type of memory cores, for example, static random access memory (SRAM) cores, or non-volatile memory cores such as flash.

[0038] Memory access interfaces 311a, 312b, and 315a are effectively operated in ‘lockstep’ (i.e., receiving the same command and address information for what is effectively the same access cycles) to form wide memory channel #1 interface. Memory access interfaces 315b, 313a, and 314b are effectively operated in lockstep to form wide memory channel #2 interface.

[0039] Memory components 321, 322, 323, and 324 (i.e., not memory component 325) may be paired. When paired, memory components 321, 322, 323, and 324 may be affixed to opposite sides of a substrate (e.g., printed circuit board). These pairs are operatively coupled (e.g., connected) to receive the same command/address signal lines from controller 310. In other words, memory component 321 is connected to, via memory access interface 321a, the same command/address signal lines of controller 310 (i.e., connected to memory access interface 311a) as memory component 322 is connected to, via memory access interface 322a. Memory component 321 is also connected to, via memory access interface 321b, to the same command/address signal lines of controller 310 (and thus connected to memory access interface 312b) as memory component 322 is connected to, via memory access interface 322b. Memory components 323 and 324 are paired and connected in the same manner to each other and memory access interfaces 313a and 314b.

[0040] The clamshell configuration of memory components 321-324 is illustrated in Figure 3. For clarity, in Figure 3, the command/address portions of the connections between controller 310 and memory components 321-325 have been omitted. Figure 3 illustrates that: memory access interface 321a of memory component 321 communicates data B0 (e.g., a byte) with memory access interface 311a; memory access interface 322a of memory component 322 communicates data B1 with memory access interface 311a; memory access
interface 32lb of memory component 321 communicates data B2 with memory access
interface 3l2b; memory access interface 322b of memory component 322 communicates data B3 with memory access interface 3l2b. Paired memory components 323 and 324 are connected in the same manner. Thus, for wide memory channel #2 interface 323: memory access interfaces 323a and 324a communicate data B0 and B1, respectively with memory access interface 313a; memory access interfaces 323b and 324b communicate data B2 and B3, respectively with memory access interface 3l4b.

Memory component 325 is also operated in clamshell mode to provide respective error correction code data to wide memory channel #1 interface and wide memory channel #2 interface. The ability of memory component 325 to independently be operated by multiple (i.e., 2) instances of wide memory channel interfaces (i.e., via memory access interface 325a which is independent of memory access interface 325b) allows memory component 325 to independently provide the error correction information (EC1 and EC2, respectively) to each of these wider channels. Thus, two independent 39 or 40-bit channels (i.e., 32-bits of data and 7-bits of error correction code or 32-bits of data and 8-bits of error correction code) may be provided using just five (5) memory components. Two memory components 321 and 322 on wide memory channel #1 interface provide 32-bits of data and one-half (½) of memory component 325 provides either 7 or 8 bits (depending upon a desired configuration and/or ECC coding scheme) of error correcting code data for a total of either 39 or 40-bits, respectively. Likewise, two memory components 323 and 324 on wide memory channel #2 interface provide 32-bits of data and one-half (½) of memory component 325 provides either 7 or 8 bits (depending upon a desired configuration and/or ECC coding scheme) of error correcting code data for a total of either 39 or 40-bits, respectively.

Figure 4 is an illustration of a 32-bit wide data channel burst with error correction information. The data channel burst illustrated in Figure 4 may be received or transmitted by
wide memory channel #1 interface and/or wide memory channel #2 interface of controller 310.

[0043] In Figure 4, data channel burst 400 has a burst length of 16 groups of data. Each group of data has 32 data bits and either 7 or 8 error correction code bits (i.e., check bits.) In an embodiment, each set of check bits is associated (i.e., provides error detection and correction for) with the corresponding 32 data bits in the same group of data. In another embodiment, the check bits for half of the data burst are aggregated to provide error correction information for the corresponding half of the respective data burst. In another embodiment, the check bits for the entire data burst are aggregated to provide error correction information for the entire data burst.

[0044] Figure 5 is a block diagram illustrating a low-power double-data rate (e.g., LPDDR4) memory system with error correction information. In Figure 5, memory system 500 comprises memory controller 510 and memory components 521-529. Controller 510 includes wide memory channel #1 interface and wide memory channel #2 interface. Wide memory channel #1 interface includes memory access interfaces 511a, 512b, 513a, 514b, and 515a. Wide memory channel #2 interface includes memory access interface 515b, 516a, 517b, 518a, and 519b.

[0045] Memory components 521-529 each have at least two independent memory access interfaces 521a-529a and 521b-529b, respectively. In other words, memory component 521 has at least two independent memory access interfaces 521a and 521b. Likewise, memory component 522 has at least two independent memory access interfaces 522a and 522b, and so on for the rest of memory components 523-529. Memory access interfaces 521a-529a and 521b-529b for a given memory component 521-529 are independent in that they can operate independently of each other to provide command, address, and data transfer functions to the respective memory access channels that they are operatively coupled to. However, the
memory access interfaces 52la-529a and 52lb-529b on a respective memory component 521-529 may share at least one common signal, such as a clock and/or strobe signal(s). Memory components 521-529 can include a dynamic random access memory (DRAM) core or other type of memory cores, for example, static random access memory (SRAM) cores, or non-volatile memory cores such as flash.

[0046] Controller 510 is operatively coupled to the memory access interfaces 52la-529a and 52lb-529b on a respective memory component 521-529 via respective memory access channels to communicate commands, addresses, and data with memory devices 521-529. Controller 510 can operate each respective independent portion (e.g., memory access interfaces 521a-529a and 521b-529b) of memory components 521-529 independently. Each of memory access interfaces 511a, 512b, 513a, 514b, 515a, 515b, 516a, 517b, 518a, and 519b include command, address, and data transfer functions.

[0047] Each of memory access interfaces 511a, 512b, 513a, 514b, 515a, 515b, 516a, 517b, 518a, and 519b may operate the command, address, and data transfer functions independently of the other of memory access interfaces 511a, 512b, 513a, 514b, 515a, 515b, 516a, 517b, 518a, and 519b. Memory access interfaces 511a, 512b, 513a, 514b, 515a, 515b, 516a, 517b, 518a, and 519b may be operated independently of each other because the memory access interfaces 521a-529a and 521b-529b can operate independently of the other memory access interfaces 521a-529a and 521b-529b and other memory components 521-529.

[0048] In an embodiment, memory access interfaces 511a, 512b, 513a, 514b, and 515a are effectively operated in ‘lockstep’ (i.e., receiving the same command and address information for what is effectively the same access cycles) to form wide memory channel #1 interface. Memory access interfaces 515b, 516a, 517b, 518a, and 519b are effectively operated in lockstep to form wide memory channel #2 interface.
In an embodiment, memory components 521-524 and 526-529 (i.e., not memory component 525) may be paired. When paired, memory components 521-524 and 526-529 may be affixed to opposite sides of a substrate (e.g., printed circuit board). These pairs are operatively coupled (e.g., connected) to receive the same command/address signal lines from a respective memory access channel—except for the chip select (CS) signal which is used to determine which devices (e.g., side of board, rank) are to provide the data to controller 510. In other words, memory component 521 is connected to, via memory access interface 521a, the same (except for CS) command/address signal lines of memory access interface 511a as memory component 522 is connected to, via memory access interface 522a. Memory component 521 is also connected to, via memory access interface 521b, the same (except for CS) command/address signal lines of memory access interface 512b as memory component 522 is connected to, via memory access interface 522b. Memory components 523 and 524 are paired and connected in the same manner to each other and memory access interfaces 513a and 514b —and so on such that memory components 526 and 527 are paired and connected, as are memory components 528 and 529.

The data signal lines of these pairs are routed to the controller together such that each memory component of the pair, supplies the whole data width of the memory channel (e.g., for a 16-bit memory channel, each component of the pair supplies 16 bits.)

The (non-clamshell) configuration of memory components 521-529 is illustrated in Figure 5. For clarity, in Figure 5, the command/address portions of the memory access channels have been omitted. Figure 5 illustrates that: memory access interface 521a of memory component 521 communicates data W0 (e.g., a 16-bit word) with memory access interface 511a; memory access interface 522a of memory component 522 also communicates data W0 with memory access interface 511a (as determined by the CS signal); memory access interface 521b of memory component 521 communicates data W1 with memory
access interface 5l2b; memory access interface 522b of memory component 522 also
communicates data W1 with memory access interface 5l2b (as determined by the CS signal).
Paired memory components 523-524, 526-527, and 528-529 are connected in the same
manner. Thus, for wide memory channel #1 interface: memory access interfaces 523a and
524a communicate data W2 with memory access interface 5l3a; and, memory access
interfaces 523b and 524b communicate data W3 with memory access interface 5l4b. For
wide memory channel #2 interface: memory access interfaces 526a and 527a communicate
data W0 with memory access interface 5l6a; memory access interfaces 526b and 527b
communicate data W1 with memory access interface 5l7a; memory access interfaces 528a
and 529a communicate data W2 with memory access interface 5l8a; memory access
interfaces 528b and 529b communicate data W3 with memory access interface 5l9b.

[0052] Memory component 525 is operated to provide respective error correction code
data to wide memory channel #1 interface and wide memory channel #2 interface. The
ability of memory component 525 to independently be operated by multiple (i.e., 2) instances
of wide memory channel interfaces (i.e., via memory access interface 525a which is
independent of memory access interface 525b) allows memory component 525 to
independently provide the error correcting information (EC1 and EC2, respectively) to each
of these wider channels. Thus, two independent 72-bit channels (i.e., 64-bits of data and 8-bits
of error correction code) may be provided using just nine (9) memory components.
However, because memory component 525 supplies word-wide (e.g., 16-bits) of data (due to
not being in, and/or having, a clamshell mode), controller 510 must manage which portion
(e.g., byte) of the ECC word is used. Controller 510 may select the portion used based on the
address being accessed.

[0053] Figure 6 is a flowchart illustrating a method of operating a memory controller.
The steps illustrated in Figure 6 may be performed by one or more of memory system 100,
memory system 300, memory system 500, and/or their components. It should be understood that the steps illustrated in Figure 6 may be performed in orders other than what is illustrated and/or performed in parallel. In addition, steps may not be dependent upon other steps in order to be performed.

[0054] A first memory component is accessed via a first memory channel interface group to receive a first data word group (602). For example, controller 100 may access, via memory access interfaces 111a, 112b, 113a, and 114b, at least memory component 121 to receive a first 64-bit data word burst. A second memory component is accessed via a second memory channel interface group to receive a second data word group (604). For example, controller 100 may access, via memory access interfaces 116a, 117b, 118a, and 119b, at least memory component 126 to receive a second 64-bit data word burst.

[0055] A third memory component is accessed, via a first error correction memory channel interface, to receive a first set of check bits associated with the first data word group (606). For example, controller 100 may access, via memory access interface 115a, memory component 125 to receive error correction code information associated with data in the first 64-bit data word burst. The third memory component is accessed, via a second error correction memory channel interface, to receive a second set of check bits associated with the second data word group (608). For example, controller 100 may access, via memory access interface 115b, memory component 125 to receive error correction code information associated with data in the second 64-bit data word burst.

[0056] Figure 7 is a block diagram illustrating a memory system with data devices operating in non-clamshell mode. The (non-clamshell) configuration of memory components 721-724 is illustrated in Figure 7. For clarity, in Figure 7, the command/address portions of the memory access channels have been omitted. Figure 7 illustrates that: memory access interface 72la of memory component 721 communicates data W0 (e.g., a 16-bit word) with
memory access interface 711a; memory access interface 721b of memory component 721
communicates data W1 with memory access interface 712b; memory access interface 722a of
memory component 722 communicates data W2 with memory access interface 713a; memory
access interface 722b of memory component 722 communicates data W3 with memory access interface 714b; memory access interface 723a of memory component 723 communicates data W0 (e.g., a 16-bit word) with memory access interface 716a; memory access interface 723b of memory component 723 communicates data W1 with memory access interface 717b; memory access interface 724a of memory component 724 communicates data W2 with memory access interface 718a; memory access interface 724b of memory component 724 communicates data W3 with memory access interface 719b. Thus, since each of memory components 721-724 is communicating 16 bit words, it should be understood that memory components 721-724 are operating in non-clamshell mode.

[0057] Memory component 725 is operated to provide respective error correction code data to wide memory channel #1 interface and wide memory channel #2 interface. The ability of memory component 725 to independently be operated by multiple (i.e., 2) instances of wide memory channel interfaces (i.e., via memory access interface 725a which is independent of memory access interface 725b) allows memory component 725 to independently provide the error correction information (EC1 and EC2, respectively) to each of these wider channels. Thus, two independent 72-bit (or 80-bit) channels (i.e., 64-bits of data and 8-bits of error correction code, or 64-bits of data and 16 bits of error correction code) may be provided using just five (5) memory components. For 8-bits of error correction code information, memory component 725 may be operated either in clamshell mode or non-clamshell mode (i.e., discard 8-bits). For 16-bits of error correction code information, memory component 725 is operated in non-clamshell mode.
The methods, systems and devices described above may be implemented in computer systems, or stored by computer systems. The methods described above may also be stored on a non-transitory computer readable medium. Devices, circuits, and systems described herein may be implemented using computer-aided design tools available in the art, and embodied by computer-readable files containing software descriptions of such circuits. This includes, but is not limited, to one or more elements of memory system 100, memory system 101, memory system 300, memory system 500, memory system 700, and their components. These software descriptions may be: behavioral, register transfer, logic component, transistor, and layout geometry-level descriptions. Moreover, the software descriptions may be stored on storage media or communicated by carrier waves.

Data formats in which such descriptions may be implemented include, but are not limited to: formats supporting behavioral languages like C, formats supporting register transfer level (RTL) languages like Verilog and VHDL, formats supporting geometry description languages (such as GDSII, GDSIII, GDSIV, CIF, and MEBES), and other suitable formats and languages. Moreover, data transfers of such files on machine-readable media may be done electronically over the diverse media on the Internet or, for example, via email. Note that physical files may be implemented on machine-readable media such as: 4 mm magnetic tape, 8 mm magnetic tape, 3-1/2 inch floppy media, CDs, DVDs, and so on.

Figure 8 is a block diagram illustrating one embodiment of a processing system 800 for including, processing, or generating, a representation of a circuit component 820. Processing system 800 includes one or more processors 802, a memory 804, and one or more communications devices 806. Processors 802, memory 804, and communications devices 806 communicate using any suitable type, number, and/or configuration of wired and/or wireless connections 808.
Processors 802 execute instructions of one or more processes 812 stored in a memory 804 to process and/or generate circuit component 820 responsive to user inputs 814 and parameters 816. Processes 812 may be any suitable electronic design automation (EDA) tool or portion thereof used to design, simulate, analyze, and/or verify electronic circuitry and/or generate photomasks for electronic circuitry. Representation 820 includes data that describes all or portions of memory system 100, memory system 101, memory system 300, memory system 500, memory system 700, and their components, as shown in the Figures.

Representation 820 may include one or more of behavioral, register transfer, logic component, transistor, and layout geometry-level descriptions. Moreover, representation 820 may be stored on storage media or communicated by carrier waves.

Data formats in which representation 820 may be implemented include, but are not limited to: formats supporting behavioral languages like C, formats supporting register transfer level (RTL) languages like Verilog and VHDL, formats supporting geometry description languages (such as GDSII, GDSIII, GDSIV, CIF, and MEBES), and other suitable formats and languages. Moreover, data transfers of such files on machine-readable media may be done electronically over the diverse media on the Internet or, for example, via email.

User inputs 814 may comprise input parameters from a keyboard, mouse, voice recognition interface, microphone and speakers, graphical display, touch screen, or other type of user interface device. This user interface may be distributed among multiple interface devices. Parameters 816 may include specifications and/or characteristics that are input to help define representation 820. For example, parameters 816 may include information that defines device types (e.g., NFET, PFET, etc.), topology (e.g., block diagrams, circuit descriptions, schematics, etc.), and/or device descriptions (e.g., device properties, device dimensions, power supply voltages, simulation temperatures, simulation models, etc.).
Memory 804 includes any suitable type, number, and/or configuration of non-transitory computer-readable storage media that stores processes 812, user inputs 814, parameters 816, and circuit component 820.

Communications devices 806 include any suitable type, number, and/or configuration of wired and/or wireless devices that transmit information from processing system 800 to another processing or storage system (not shown) and/or receive information from another processing or storage system (not shown). For example, communications devices 806 may transmit circuit component 820 to another system. Communications devices 806 may receive processes 812, user inputs 814, parameters 816, and/or circuit component 820 and cause processes 812, user inputs 814, parameters 816, and/or circuit component 820 to be stored in memory 804.

The foregoing description of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and other modifications and variations may be possible in light of the above teachings. The embodiment was chosen and described in order to best explain the principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the invention in various embodiments and various modifications as are suited to the particular use contemplated. It is intended that the appended claims be construed to include other alternative embodiments of the invention except insofar as limited by the prior art.
CLAIMS

What is claimed is:

1. A controller, comprising:

   a plurality of memory channel interfaces that each include a respective plurality of memory access interfaces, the plurality of memory access interfaces to operate independently of the other of the plurality of memory access interfaces;

   the plurality of memory channel interfaces including at least a first group of memory access interfaces to access a first data word group and a second group of memory access interfaces to access a second data word group;

   the plurality of memory channel interfaces including at least a first error correction memory access interface and a second error correction memory access interface, the first error correction memory access interface and the second error correction memory access interface to access a same memory component, the first error correction memory access interface to access a first group of check bits associated with the first data word group, the second error correction memory access interface to access a second group of check bits associated with the second data word group.

2. The controller of claim 1, wherein the plurality of memory access interfaces each include command, address, and data transfer functions that operate independently of the command, address, and data transfer functions of the other of the memory access interfaces.
3. The controller of claim 1, wherein the first data word group consists of a first number of bits and the first group of check bits consists of a second number of bits, the first number of bits to be eight (8) times the second number of bits.

4. The controller of claim 2, wherein the same memory component is to be operated in a clamshell mode.

5. The controller of claim 4, wherein the first group of memory access interfaces includes four (4) memory access interfaces to access 64-bit data word groups.

6. The controller of claim 4, wherein a first one of the first group of memory access interfaces is to be connected to a first memory component and a second memory component.

7. The controller of claim 6, wherein a first memory component and a second memory component are to be operated in the clamshell mode.

8. A controller, comprising:

   a first memory channel interface to access at least a first memory component to communicate a first data word group;

   a second memory channel interface to access at least a second memory component to communicate a second data word group;

   an error correction data channel interface to access a third memory component, the error correction data channel interface including a first memory access interface and a second memory access interface, the first memory access interface and the second memory access interface to be connected to the third
memory component, the first memory access interface to access the third memory component to communicate a first group of check data associated with the first data word group, the second memory access interface to access the third memory component to communicate a second group of check data associated with the second data word group.

9. The controller of claim 8, wherein the third memory component is to be operated in a clamshell mode.

10. The controller of claim 8, wherein the at least first memory component and the at least second memory component are to be operated in a clamshell mode.

11. The controller of claim 8, wherein the first memory channel interface is comprised of a first plurality of memory access interfaces and the second memory channel interface is comprised of a second plurality of memory access interfaces.

12. The controller of claim 11, wherein the error correction data channel interface comprises a third plurality of memory access interfaces, the third plurality of memory access interfaces including the first memory access interface and the second memory access interface.

13. The controller of claim 12, wherein the first plurality of memory access interfaces, the second plurality of memory access interfaces, and third plurality of memory access interfaces each include respective command, address, and data transfer functions.
14. The controller of claim 13, where the respective command, address, and data transfer functions of the first plurality of memory access interfaces, the second plurality of memory access interfaces, and third plurality of memory access interfaces operate independently of the command, address, and data transfer functions of the other of the first plurality of memory access interfaces, the second plurality of memory access interfaces, and third plurality of memory access interfaces.

15. A method of operating a controller, comprising:

   accessing at least a first memory component via a first memory channel interface group, to receive a first data word group;

   accessing at least a second memory component via a second memory channel interface group, to receive a second data word group;

   accessing a third memory component to receive a first set of check bits associated with the first data word group via a first error correction memory channel interface;

   accessing the third memory component to receive a second set of check bits associated with the second data word group via a second error correction memory channel interface.

16. The method of claim 15, wherein each of the memory channel interfaces includes a respective plurality of command, address, and data transfer functions that operate independently of the command, address, and data transfer functions of the other of the memory channel interfaces.
17. The method of claim 16, wherein the accessing, to receive the first set of check bits, is performed concurrently with the accessing by via the first memory channel interface group.

18. The method of claim 15, the wherein the first data word group consists of a first number of bits and the first group of check bits consists of a second number of bits, the first number of bits to be eight (8) times the second number of bits.

19. The method of claim 15, the wherein the first data word group consists of a first number of bits and the first group of check bits consists of a second number of bits, the first number of bits to be four (4) times the second number of bits.

20. The method of claim 15, wherein the at least first memory component includes a third memory component and the at least second memory component includes a fourth memory component.
ACCESS A FIRST MEMORY COMPONENT VIA A FIRST MEMORY CHANNEL INTERFACE GROUP TO RECEIVE A FIRST DATA WORD GROUP

ACCESS A SECOND MEMORY COMPONENT VIA A SECOND MEMORY CHANNEL INTERFACE GROUP TO RECEIVE A SECOND DATA WORD GROUP

ACCESS A THIRD MEMORY COMPONENT, VIA A FIRST ERROR CORRECTION INFORMATION MEMORY CHANNEL INTERFACE, TO RECEIVE A FIRST SET OF CHECK BITS ASSOCIATED WITH THE FIRST DATA WORD GROUP

ACCESS THE THIRD MEMORY COMPONENT, VIA A SECOND ERROR CORRECTION INFORMATION MEMORY CHANNEL INTERFACE, TO RECEIVE A SECOND SET OF CHECK BITS ASSOCIATED WITH THE SECOND DATA WORD GROUP

FIGURE 6
**INTERNATIONAL SEARCH REPORT**

**CLASSIFICATION OF SUBJECT MATTER**

**IPC** - G06F 11/07, 11/10, 11/30 (2019.01)

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

**FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
See Search History document

**DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>
| X         | WO 2016/049448 A1 (EVERSPIN TECHNOLOGIES, INC.) 31 March 2016; paragraphs [0029], [0032], [0033], [0037]; claim 7; figure 1 | 1-3, 8, 11-20 ...
|           |                                                                         | 4-7, 9, 10          |
| Y         | US 2013/0342998 A1 (CHEN, K) 26 December 2013; paragraph [0057]          | 4-7, 9, 10          |
| A         | US 9262261 B2 (MICRON TECHNOLOGY, INC.) 16 February 2016; entire document| 1-20                 |

Further documents are listed in the continuation of Box C.

See patent family annex.

Date of the actual completion of the international search  
17 June 2019 (17.06.2019)

Date of mailing of the international search report  
09 JUL 2019

Name and mailing address of the ISA/  
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents  
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-8300

Authorized officer  
Shane Thomas
PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774