Example implementations relate to using a spare memory on a memory module. In example implementations, a memory module may have a plurality of memories, including default memories and a spare memory. A plurality of data buffers on the memory module may select data nibbles from the plurality of memories such that when a default memory is identified as defective, a data nibble is selected from the spare memory and not from the defective default memory. A data nibble selected from the default memory may be in a first position in an output of the memory module when the default memory is functional. A data nibble selected from the spare memory may be in a second position in an output of the memory module.
MEMORY SPARING ON MEMORY MODULES

BACKGROUND

[0001] Thousands of memory dies may reside on memory modules in a server. Error correction logic may be used to correct errors detected in the memory dies. Memory modules may include extra memory dies so that when a memory die on a memory module malfunctions, an extra memory die may be used instead of the entire memory module having to be replaced.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The following detailed description references the drawings, wherein:
[0003] FIG. 1 is a block diagram of an example data buffer that enables selections of various combinations of data nibbles;
[0004] FIG. 2 is a block diagram of an example data buffer that includes registers to store outputs of data selectors in the data buffer;
[0005] FIG. 3 is a block diagram of an example memory module that includes data buffers to select various combinations of data nibbles stored in memories on the memory module;
[0006] FIG. 4 is a block diagram of an example system controller to enable use of a spare memory on a memory module;
[0007] FIG. 5 is a block diagram of an example memory module that includes multiple spare memories;
[0008] FIG. 6 is a block diagram of an example device that includes a machine-readable storage medium encoded with instructions to enable use of a spare memory on a memory module;
[0009] FIG. 7 is a block diagram of an example device that includes a machine-readable storage medium encoded with instructions to copy and correct data stored on a memory module;
[0010] FIG. 8 is a block diagram of an example device that includes a machine-readable storage medium encoded with instructions to notify a memory controller of use of a spare memory on a memory module and to correct erroneous data in the spare memory;
FIG. 9 is a flowchart of an example method for using a spare memory on a memory module; and
FIG. 10 is a flowchart of an example method for correcting erroneous data stored on a memory module.

DETAILED DESCRIPTION

As memory die density on memory modules increases, the difficulty of routing data to and from an extra memory on a memory module may increase. The present disclosure provides for data buffers, on memory modules, having data selectors that may select data from various memories, including a spare memory. The data selectors may be configured in various ways depending on the location of a malfunctioning memory. The configurability of the data selectors may simplify the routing challenge of using extra memories, reducing the rate at which entire memory modules need to be replaced.

Referring now to the drawings, FIG. 1 is a block diagram of an example data buffer 100 that enables selections of various combinations of data nibbles. Data buffer 100 may select data nibbles from memories on a memory module. The memories may include volatile memories, such as dynamic random-access memory (DRAM), and/or non-volatile memories, such as flash memory.

In FIG. 1, data buffer 100 includes data selector 102 and data selector 104. In some implementations, data selectors 102 and 104 may include multiplexers (e.g., 2-to-1 multiplexers). Data selector 102 may select between a data nibble (e.g., Data nibble 1) stored in a first memory, and a data nibble (e.g., Data nibble 2) stored in a second memory. The term “data nibble” as used herein refers to a unit of data that is more than one bit but less than a byte. For example, a data nibble may be four bits. Data selector 104 may select between a data nibble (e.g., Data nibble 3) stored in a third memory, and the data nibble stored in the second memory (e.g., Data nibble 2). The first, second, and third memories may be on integrated circuits (ICs) on a memory module. Data selector 102 may be communicatively coupled to the first and second memories, and data selector 104 may be communicatively coupled to the second and third memories.

Selections of data selectors 102 and 104 may be determined based on whether the first and second memories on the memory module are functional.
The term "functional" as used herein refers to a memory that accurately stores data that is written to the memory, and that accurately outputs stored data when a read operation is performed on the memory. When the first memory and the second memory are functional, data selector 102 may select a data nibble (e.g., Data nibble 1) stored in the first memory, and data selector 104 may select a data nibble (e.g., Data nibble 2) stored in the second memory. When the first memory is not functional, data selector 102 may select the data nibble (e.g., Data nibble 2) stored in the second memory, and data selector 104 may select a data nibble (e.g., data nibble 3) stored in the third memory. When the second memory is not functional, data selector 102 may select the data nibble (e.g., Data nibble 1) stored in the first memory, and data selector 104 may select the data nibble (e.g., data nibble 3) stored in the third memory. Data selectors 102 and 104 may receive selection control signals from a system controller on the memory module.

[0017] An output of data buffer 100 may include data nibbles selected by data selectors 102 and 104. In some implementations, data buffer 100 may provide an 8-bit (i.e., 1-byte) output. The first four bits of the output may be the data nibble (e.g., Selected data nibble 1) selected by data selector 102, and the last four bits of the output may be the data nibble (e.g., Selected data nibble 2) selected by data selector 104.

[0018] FIG. 2 is a block diagram of an example data buffer 200 that includes registers to store outputs of data selectors in the data buffer. Data buffer 200 may include data selectors 202 and 204, which may be analogous to data selectors 102 and 104, respectively, of FIG. 1. Data buffer 200 may also include registers 206 and 208. Register 208 may store an output of data selector 202. The output of data selector 202 may include a data nibble (e.g., Data nibble 1 or 2) selected by data selector 202. Register 208 may store an output of data selector 204. The output of data selector 204 may include a data nibble (e.g., Data nibble 2 or 3) selected by data selector 204.

[0019] Registers 208 and 208 may output their respective stored data nibbles, and an output of data buffer 200 may include the data nibbles output by registers 206 and 208. For example, register 206 may output Selected data nibble 1, and register 208 may output Selected data nibble 2. Each selected data
nibble may be four bits long. Data buffer 200 may provide an 8-bit (i.e., 1-byte) output; the first four bits of the output may be Selected data nibble 1, and the last four bits of the output may be Selected data nibble 2.

[0020] FIG. 3 is a block diagram of an example memory module 300 that includes data buffers to select various combinations of data nibbles stored in memories on the memory module. Memory module 300 may be an in-line memory module, such as a single in-line memory module (SIMM) or a dual in-line memory module (DIMM), or any memory module suitable for mounting memory integrated circuits (ICs). In FIG. 3, memory module 300 includes default memories 302, 304, 308, and 308, spare memory 310, data buffers 320 and 322, and system controller 330.

[0021] Memories 302, 304, 308, 308, and 310 may include volatile (e.g., DRAM) or non-volatile (e.g., flash, read-only memory) memories, and may be on ICs on memory module 300. Memories 302, 304, 306, and 308 may be default memories on memory module 300. The term "default memory" as used herein refers to a memory on which read and write operations are regularly performed. Memory 310 may be a spare memory on memory module 300. The term "spare memory" as used herein refers to a memory used to replace a default memory when a default memory does not accurately store or output data. The term "defective" may be used in the present disclosure to describe a memory or portion of a memory that does not accurately store data that is written to the memory, and/or that does not accurately output stored data when a read operation is performed on the memory. In some implementations, memories 302, 304, 308, 308, and 310 may be volatile memories having error-correcting code (ECC) functionality.

[0022] Data buffers 320 and 322 on memory module 300 may be analogous to data buffer 100 or data buffer 200. The data buffers may perform read and write operations on the memories on memory module 300. Data buffer 320 may be communicatively coupled to default memories 302, 304, and 306. Data buffer 322 may be communicatively coupled to default memories 308 and 308, and to spare memory 310. Each data buffer on memory module 300 may select two data nibbles (i.e., a data buffer may select data nibbles from two of the three memories to which the data buffer is coupled). A data strobe signal may be
associated with each data nibble, and each data strobe signal may be associated
with a pin on memory module 300. It should be understood that memory
module 300 may have more or less memories (e.g., memory module 300 may
have more than one spare memory) and/or data buffers than shown in FIG. 3,
and that discussions herein regarding the memories and/or data buffers shown in
FIG. 3 may be applicable to additional memories and/or data buffers.

[0023] When all default memories on memory module 300 are functional, data
buffer 320 may select a data nibble stored in default memory 302 and a data
nibble stored in default memory 304, and data buffer 322 may select a data
nibble stored in default memory 308 and a data nibble stored in default
memory 308. When a default memory, or a portion of a default memory, is
determined to be defective, data buffers 320 and 322 may select data nibbles
from functional default memories and from spare memory 310, and may not
select a data nibble from the default memory that is defective or that has a
defective portion. For example, if default memory 302 is defective, data
buffer 320 may select a data nibble from default memory 304 and a data nibble
from default memory 306, and data buffer 322 may select a data nibble from
default memory 308 and a data nibble from spare memory 310.

[0024] Data buffer 320 may provide an 8-bit output, for example Byte 0.
Similarly, data buffer 322 may provide an 8-bit output, for example Byte 1. An
output of memory module 300 may include outputs of the data buffers in a
specified order. For example, memory module 300 may have a 16-bit output; the
first eight bits of the output may be the two data nibbles (e.g., Byte 0) outputted
by data buffer 320, and the last eight bits of the output may be the two data
nibbles (e.g., Byte 1) outputted by data buffer 322. Each bit outputted by a data
buffer may be associated with a pin on memory module 300. The outputs of the
data buffers may include the data nibbles selected by the data buffers from
memories on memory module 300. An output of memory module 300 may be a
cache line made up of Byte 0 and Byte 1, each byte being made up of the data
nibbles selected by the respective data buffer.

[0025] When all default memories on memory module 300 are functional, a
data nibble selected from default memory 302 may be the first four bits of Byte 0,
a data nibble selected from default memory 304 may be the last four bits of
Byte 0, a data nibble selected from default memory 308 may be the first four bits of Byte 1, and a data nibble selected from default memory 308 may be the last four bits of Byte 1. When a default memory on memory module 300 is defective, a data nibble may be selected from spare memory 310, and a data nibble may not be selected from the defective default memory. The data nibble selected from spare memory 310 may be the last four bits of Byte 1 (e.g., the last four bits of an output of memory module 300), and data nibbles selected from functional default memories may be shifted in the output of memory module 300, relative to an output of memory module 300 produced when all data nibbles are selected from default memories (i.e., when all default memories are functional).

[0026] For example, when all default memories shown in FIG. 3 are functional, data nibbles selected from default memories 306 and 308 may be the first four bits of Byte 1 and the last four bits of Byte 1, respectively, but when default memory 304 is defective, data nibbles selected from default memories 306 and 308 may be the second four bits of Byte 0 and the first four bits of Byte 1, respectively, since a data nibble may not be selected from default memory 304 (a data nibble selected from default memory 302 may be the first four bits of Byte 0 in both cases). As illustrated by this example, when a default memory in a memory module is defective, data bits selected from default memories between the defective default memory and a spare memory may shift by four bits in an output of the memory module, relative to an output of the memory module when all default memories are functional. Data nibbles selected from default memories on the other side of the defective default memory may appear in the same position in the output of the memory module in both cases.

[0027] In some implementations, a first portion of a default memory may be defective, and a second portion of the same default memory may be functional. In such implementations, when the address of a data nibble to be read from the default memory refers to a location in the defective portion, data buffers 320 and 322 may read and select a data nibble stored in a spare memory (e.g., spare memory 310), and may not select a data nibble from the default memory. When the address of a data nibble to be read from the default memory refers to a location in the functional portion, data buffers 320 and 322 may read and select
the data nibble stored in the default memory, and may not select a data nibble from the spare memory.

[0028] Machine-readable instructions (e.g., software) on memory module 300 may determine which memories data nibbles should be selected from, and may specify selection control signals accordingly. System controller 330 on memory module 300 may be notified of the selection control signals. For example, machine-readable instructions may write the selection control signals into internal registers within system controller 330. System controller 330 may transmit the selection control signals to data buffers 320 and 322. The selection control signals may be used to configure data selectors in the data buffers.

[0029] System controller 330 may be communicatively coupled to data buffers 320 and 322 via a data bus, which may carry selection control signals as well as control signals for directional timing and/or die termination (ODT). System controller 330 may control routing of data on memory module 300. In some implementations, system controller 330 may buffer control signals from a memory controller external to memory module 300. The external memory controller may receive data (e.g., Byte 0, Byte 1) that is output by the data buffers on memory module 300.

[0030] A memory controller that receives a data output from memory module 300 may detect an error in the output and/or detect a defect in a default memory on memory module 300. The memory controller may determine the severity of the error and whether a defective default memory should continue to be used. The memory controller may determine which byte or data nibble of a memory controller output has an error, and identify a memory from which erroneous data was read. Machine-readable instructions (e.g., software and/or firmware) on the memory controller and/or on memory module 300 may determine a range of memory addresses that exhibits errors.

[0031] In some implementations, an external memory controller communicatively coupled to memory module 300 may not be informed of the presence of a spare memory on memory module 300. When an error is detected in the memory module output, machine-readable instructions (e.g., software) on the memory controller may correct the error (e.g., using ECC logic) and store a copy of the corrected data in a temporary location (e.g., a buffer, virtual memory).
The machine-readable instructions may also determine selection control signals for data buffers on memory module 300 such that a data nibble is not selected from the defective default memory and a data nibble is selected from a spare memory. The corrected data may then be copied from the temporary location to the memories from which data nibbles will be selected, according to the selection control signals. Such copying may result in default memories between the defective default memories and the spare memory storing different data than that stored when all default memories are functional.

[0032] For example, if default memory 308 in memory module 300 is defective, selection control signals may configure data buffer 320 to select data nibbles from default memories 302 and 304, and data buffer 322 to select data nibbles from default memory 308 and spare memory 310. After the corrected data is copied into the appropriate memories, default memory 308 may store a data nibble formerly stored in default memory 308, and spare memory 310 may store a data nibble formerly stored in default memory 308 (default memories 302 and 304 may store the same respective data nibbles as before). Thus, the order of data nibbles in the output of memory module 300 may be the same when spare memory 310 is used as when all default memories are functional (and spare memory 310 is not used).

[0033] In some implementations, an external memory controller communicatively coupled to memory module 300 may be informed, for example by machine-readable instructions (e.g., software and/or firmware) on memory module 300, of the presence of a spare memory on memory module 300. In such implementations, the spare memory may store a copy of data stored in a defective default memory, and data buffers on memory module 300 may be configured such that a data nibble is selected from the spare memory instead of the defective default memory. For example, if default memory 304 is defective, spare memory 310 may store a copy of data stored in default memory 304. Data buffer 320 may select a data nibble stored in default memory 302 and a data nibble stored in default memory 308, and data buffer 322 may select a data nibble stored in default memory 308 and spare memory 310. Thus, the data in the memory module output may be "scrambled" when the spare memory is used; when all default memories are functional, a data nibble selected from default
memory 304 makes up the last four bits of Byte 0, but when default memory 304 is defective, the information in that data nibble is in the last four bits of Byte 1, and the bits in between are shifted relative to where they would be if all default memories were functional. That is, when all default memories shown in FIG. 3 are functional, the data nibbles selected from default memories 306 and 308 are the first four bits and the last four bits, respectively, of Byte 1, but when default memory 304 is defective, the data nibbles selected from default memories 308 and 308 are the last four bits of Byte 0 and the first four bits of Byte 1, respectively.

[0034] The memory controller, having been informed of the presence of a spare memory, may be notified (e.g., when a default memory is determined to be defective) that a data nibble is selected from the spare memory instead of from a defective default memory. The memory controller may "unscramble" the order of data nibbles in the memory module output (e.g., put the data nibbles in the order they would have been in had they all been selected from default memories), then apply error correction logic. In some implementations, erroneous data in the spare memory may be corrected. For example, data stored in the spare memory may be scanned, and ECC logic and/or other suitable error correction logic may be used to detect and correct errors, in some implementations, the entire spare memory may be scanned, in some implementations, a portion of the spare memory, having a copy of data that was stored in a defective portion of a default memory, may be scanned.

[0035] FIG. 4 is a block diagram of an example system controller 400 to enable use of a spare memory on a memory module. System controller 400 may be implemented, for example, as system controller 330. System controller 400 may include modules 402 and 404. A module may include a set of instructions encoded on a machine-readable storage medium and executable by a processor of system controller 400. In addition or as an alternative, a module may include a hardware device comprising electronic circuitry for implementing the functionality described below.

[0036] Data buffer interface module 402 may transmit information to data buffers communicatively coupled to system controller 400 on a memory module. Data buffer interface module 402 may transmit selection control signals to data buffers. In
some implementations, data buffer interface module 402 may transmit, to data buffers, addresses from which data nibbles should be read. The addresses may be specified by a memory controller external to the memory module.

[0037] Records module 404 may maintain records of defective default memories and respective spare memories from which data nibbles are selected instead. The term "maintain", as used herein with respect to records, refers to generating and/or storing new records, updating existing records, and deleting existing records. For example, a memory module may have a first spare memory and a second spare memory. When a first data buffer on the memory module receives, from system controller 400, a first address corresponding to a location in a first defective default memory on the memory module (or a location in a defective portion of a first default memory, as discussed above with respect to FIG. 3), data buffers on the memory module may select a data nibble from the first spare memory instead of the first defective default memory. When a second data buffer on the memory module receives, from system controller 400, a second address corresponding to a location in a second defective default memory (or a location in a defective portion of a second default memory) on the memory module, data buffers on the memory module may select a data nibble from the second spare memory instead of the second defective default memory. Records module 404 may store a record identifying the first default memory (or identifying a defective portion of the first default memory), and identifying the first spare memory as being used to replace the (defective portion of the) first default memory. Records module 404 may also store a record identifying the second default memory (or identifying a defective portion of the second default memory), and identifying the second spare memory as being used to replace the (defective portion of the) second default memory. The records may be maintained in, for example, a read-only memory (ROM) in system controller 400.

[0038] FIG. 5 is a block diagram of an example memory module 500 that includes multiple spare memories. Memory module 500 may be an in-line memory module, such as a SIMM or DIMM, or any memory module suitable for mounting memory ICs. In FIG. 5, memory module 500 includes default memories 502, 504, 506, 508, 510, 512, 514, 518, 518, 520, 522, 524, 526, 528, 530, 532, 534, 536, 542, 544, 546, 548, 550, 552, 554, 556, 558, 560, 562, 564,
588, 568, 570, 572, 574, and 578, spare memories 538 and 540, data buffers 580, 582, 584, 586, 588, 590, 592, 594, and 596, and system controller 578. System controller 578, which may be implemented as system controller 400, may be communicatively coupled to memory controller 598, which may be external to memory module 500. Memory controller 598 may perform similar functions to an external memory controller discussed above with respect to FIG. 3.

[0039] Default memories 502, 504, 506, 508, 510, 512, 514, 516, 518, 520, 522, 524, 526, 528, 530, 532, 534, and 536 may be in a first rank, and default memories 542, 544, 546, 548, 550, 552, 554, 556, 558, 560, 562, 564, 568, 568, 570, 572, 574, and 578 may be in a second rank. A default memory in FIG. 5 may be analogous to a default memory in FIG. 3. Each group of four default memories directly above a data buffer may be housed in a quad die package (QDP). For example, default memories 502, 504, 542, and 544 may be housed in a first QDP, default memories 506, 508, 546, and 548 may be housed in a second QDP, etc.

[0040] Spare memories 538 and 540 in FIG. 5 may be analogous to spare memory 310 in FIG. 3. In some implementations, spare memories 538 and 540 may both be in the first rank or may both be in the second rank. In some implementations, spare memory 538 may be in the first rank and spare memory 540 may be in the second rank, or vice-versa. An IC in which spare memories are housed may have a different number of die than the ICs in which default memories are housed.

[0041] A data buffer in FIG. 5 may be analogous to a data buffer in FIG. 3. Each data buffer in FIG. 5 may select data nibbles from two memories communicatively coupled to the respective data buffer, in a similar manner as that discussed above with respect to FIG. 3. Data buffers 580, 582, 584, 586, 588, 590, 592, 594, and 596 may receive selection control signals from system controller 578. The memories from which data buffers 580, 582, 584, 588, 588, 590, 592, 594, and 596 select data nibbles may all be in the same rank. For example, if spare memories 538 and 540 are both in the first (or second) rank, up to two defective default memories in the first (or second) rank may be replaced. If spare memory 538 is in the first rank and spare memory 540 is in the second
rank, spare memory 538 may be used to replace a defective default memory in the first rank, and spare memory 540 may be used to replace a defective default memory in the second rank.

[0042] FIG. 6 is a block diagram of an example device 600 that includes a machine-readable storage medium encoded with instructions to enable use of a spare memory on a memory module. Device 600 may be a memory controller, such as memory controller 598. In FIG. 6, device 600 includes processor 602 and machine-readable storage medium 604.

[0043] Processor 602 may include a central processing unit (CPU), microprocessor (e.g., semiconductor-based microprocessor), and/or other hardware device suitable for retrieval and/or execution of instructions stored in machine-readable storage medium 604. Processor 602 may fetch, decode, and/or execute instructions 606, 608, and 610 to enable use of a spare memory on a memory module, as described below. As an alternative or in addition to retrieving and/or executing instructions, processor 602 may include an electronic circuit comprising a number of electronic components for performing the functionality of instructions 606, 608, and/or 610.

[0044] Machine-readable storage medium 604 may be any suitable electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, machine-readable storage medium 604 may include, for example, a random-access memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage device, an optical disc, and the like. In some implementations, machine-readable storage medium 604 may include a non-transitory storage medium, where the term "non-transitory" does not encompass transitory propagating signals. As described in detail below, machine-readable storage medium 604 may be encoded with a set of executable instructions 606, 608, and 610.

[0045] Instructions 606 may identify a defective portion of a default memory on a memory module. In some implementations, a portion of a default memory may be identified as defective if data stored in/read from the portion of the default memory is repeatedly corrected using ECC logic or other error correction logic. For example, a counter may be incremented each time an error from the portion of the default memory is corrected, and when the value in the counter exceeds a threshold value,
the portion of the default memory may be identified as defective. Records of
defective DRAMs may be updated (e.g., by records module 404 of FIG. 4) when the
portion of the default memory is identified as defective.

[0046] Instructions 808 may determine selection control signals for data
selectors on a memory module. Selection control signals may be determined
such that a data nibble is selected from a spare memory on the memory module
and not from a defective default memory. A data nibble selected from the default
memory may be in a first position in an output of the memory module when the
defective portion is functional. A data nibble selected from the spare memory may
be in a second position in the output of the memory module.

[0047] Instructions 810 may notify a system controller, on a memory module,
of selection control signals. In some implementations, instructions 610 may write
selection control signals into internal registers within the system controller. The
system controller may be communicatively coupled to data buffers having data
selectors.

[0048] FIG. 7 is a block diagram of an example device 700 that includes a
machine-readable storage medium encoded with instructions to copy and correct
data stored on a memory module. Device 700 may be a memory controller, such
as memory controller 598. In FIG. 7, device 700 includes processor 702 and
machine-readable storage medium 704.

[0049] As with processor 602 of FIG. 6, processor 702 may include a CPU,
microprocessor (e.g., semiconductor-based microprocessor), and/or other
hardware device suitable for retrieval and/or execution of instructions stored in
machine-readable storage medium 704. Processor 702 may fetch, decode, and/
or execute instructions 706, 708, 710, 712, 714, and 716 to enable copying and
correction of data on a memory module, as described below. As an alternative or
in addition to retrieving and/or executing instructions, processor 702 may include
an electronic circuit comprising a number of electronic components for performing
the functionality of instructions 706, 708, 710, 712, 714, and/or 716.

[0050] As with machine-readable storage medium 604 of FIG. 6, machine-
readable storage medium 704 may be any suitable physical storage device that
stores executable instructions. Instructions 706, 708, and 712 on machine-readable
storage medium 704 may be analogous to instructions 608, 608, and 810 on
machine-readable storage medium 804. Instructions 710 may correct erroneous data read from a defective portion of a default memory on a memory module. For example, the erroneous data may be corrected using ECC logic and/or other suitable error correction logic.

[0051] Instructions 714 may write corrected data into a non-defective default memory on the memory module. The corrected data may be stored in a temporary location, such as a virtual memory or disk drive, with data copied from other non-defective default memories. The corrected data may be copied from the temporary location and written into a non-defective default memory from which a data nibble will be selected, in accordance with determined selection control signals. The non-defective default memory to which the corrected data is copied may store different data when all default memories are functional, as discussed above with respect to FIG. 3.

[0052] Instructions 718 may transfer data from a non-defective default memory to a spare memory on the memory module. A first data nibble may be copied from a first non-defective default memory and stored in a temporary location, such as a virtual memory or disk drive. The temporary location may also store a second data nibble copied from a second non-defective default memory, and corrected data from a defective default memory. The first data nibble may be copied from the temporary location and written into a spare memory. When the first data nibble is written into the spare memory, the second data nibble may be written into the first non-defective default memory.

[0053] FIG. 8 is a block diagram of an example device 800 that includes a machine-readable storage medium encoded with instructions to notify a memory controller of use of a spare memory on a memory module and to correct erroneous data in the spare memory. Device 800 may be a memory controller, such as memory controller 598. In FIG. 8, device 800 includes processor 802 and machine-readable storage medium 804.

[0054] As with processor 602 of FIG. 6, processor 802 may include a CPU, microprocessor (e.g., semiconductor-based microprocessor), and/or other hardware device suitable for retrieval and/or execution of instructions stored in machine-readable storage medium 804. Processor 802 may fetch, decode, and/or execute instructions 806, 808, 810, 812, 814, and 818 to enable notifying a
memory controller of use of a spare memory and correcting data in the spare memory, as described below. As an alternative or in addition to retrieving and/or executing instructions, processor 802 may include an electronic circuit comprising a number of electronic components for performing the functionality of instructions 806, 808, 810, 812, 814, and/or 816.

[0055] As with machine-readable storage medium 804 of FIG. 8, machine-readable storage medium 804 may be any suitable physical storage device that stores executable instructions. Instructions 806, 808, and 810 on machine-readable storage medium 804 may be analogous to instructions 608, 608, and 810 on machine-readable storage medium 604. Instructions 812 may notify a memory controller, external to a memory module, of selection control signals determined by instructions 808. Instructions 812 may thus notify the memory controller that a data nibble is selected from a spare memory instead of a defective default memory. The spare memory may store a copy of data stored in a defective portion of the default memory.

[0056] Instructions 814 may transmit selection control signals from a system controller on a memory module to data selectors on the memory module. The selection control signals may be used to configure the data selectors, which may be in data buffers on the memory module. The selection control signals may be transmitted from the system controller to the data selectors via a data bus.

[0057] Instructions 818 may correct erroneous data in a spare memory on a memory module. For example, data stored in the spare memory may be scanned, and ECC logic and/or other suitable error correction logic may be used to detect and correct errors. In some implementations, the entire spare memory may be scanned. In some implementations, a portion of the spare memory, having a copy of data that was stored in a defective portion of a default memory, may be scanned.

[0058] FIG. 9 is a flowchart of an example method 900 for using a spare memory on a memory module. Although execution of method 900 is described below with reference to processor 602 of FIG. 6, it should be understood that execution of method 900 may be performed by other suitable devices, such as processors 702 and 802 of FIGS. 7 and 8, respectively. Method 900 may be
implemented in the form of executable instructions stored on a machine-readable storage medium and/or in the form of electronic circuitry.

[0059] Method 900 may start in block 902, where processor 602 may identify a defective portion of a default memory. In some implementations, a portion of a default memory may be identified as defective if data stored in/read from the portion of the default memory is repeatedly corrected using ECC logic or other error correction logic. For example, a counter may be incremented each time an error from the portion of the default memory is corrected, and processor 602 may monitor the value in the counter. When the value in the counter exceeds a threshold value, processor 602 may identify the portion of the default memory as defective.

[0060] Next, in block 904, processor 602 may determine selection control signals for data selectors on the memory module. Selection control signals may be determined such that a data nibble is selected from a spare memory on the memory module and not from a defective default memory. A data nibble selected from the default memory may be in a first position in an output of the memory module when the defective portion is functional. A data nibble selected from the spare memory may be in a second position in the output of the memory module.

[0061] Finally, in block 906, processor 602 may notify a system controller, on the memory module, of the selection control signals. In some implementations, processor 602 may write selection control signals into internal registers within the system controller. The system controller may be communicatively coupled to data buffers having data selectors.

[0062] FIG. 10 is a flowchart of an example method 1000 for correcting erroneous data stored on a memory module. Although execution of method 1000 is described below with reference to processor 602 of FIG. 6, it should be understood that execution of method 1000 may be performed by other suitable devices, such as processors 702 and 802 of FIGS. 7 and 8, respectively. Method 1000 may be implemented in the form of executable instructions stored on a machine-readable storage medium and/or in the form of electronic circuitry.

[0063] Method 1000 may start in block 1002, where processor 602 may determine the location of an error. For example, processor 602 may determine which byte or data nibble in an output of a memory module contains an error. In some implementations, processor 602 may determine an address range in a
memory from which erroneous data was read. For example, processor 802 may
determine a rank, bank group, bank, and/or row of the memory from which
erroneous data was read.

[0064] In block 1004, processor 602 may determine whether a memory
controller has knowledge of a spare memory on the memory module. If the memory
controller does not have knowledge of a spare memory, method 1000 may proceed
to block 1006, in which processor 602 may correct erroneous data read from a
defective portion of a default memory. For example, processor 602 may correct
erroneous data using ECC logic and/or other suitable error correction logic.

[0065] In block 1008, processor 602 may copy the corrected data to a temporary
location. The temporary location may include a disk drive, virtual memory, and/or
buffer. In some implementations, processor 602 may copy data from non-defective
default memories to the temporary location. The temporary location may be on the
memory controller and/or on the memory module.

[0066] In block 1010, processor 602 may notify a system controller on the
memory module of selection control signals. In some implementations,
processor 602 may write selection control signals into internal registers within the
system controller. The system controller may be communicatively coupled to data
buffers having data selectors.

[0067] In block 1012, processor 602 may transfer data from a non-defective
default memory to a spare memory on the memory module. For example, a first
data nibble may be copied from a first non-defective default memory and stored in
the temporary location of block 1008. Processor 602 may copy the first data nibble
from the temporary location and write the first data nibble into a spare memory.
When processor 602 writes the first data nibble into the spare memory,
processor 602 may also write the corrected data of block 1008 to the first non-
defective default memory or another non-defective default memory.

[0068] If, in block 1004, processor 602 determines that the memory controller
does have knowledge of a spare memory, method 1000 may proceed to
block 1014, at which processor 602 may notify the system controller and memory
controller of selection control signals. Processor 602 may write selection control
signals into internal registers within the system controller as well as into registers
within the memory controller. After being notified of the spare memory, the
memory controller may be able to "unscramble" the order of data nibbles in the memory module output (e.g., put the data nibbles in the order they would have been in had they all been selected from default memories), then apply error correction logic, as discussed above with respect to FIG. 3.

[0069] Next, in block 1016, processor 602 may correct erroneous data in the spare memory. For example, processor 602 may scan data stored in the spare memory, and ECC logic and/or other suitable error correction logic may be used to detect and correct errors. In some implementations, the entire spare memory may be scanned. In some implementations, a portion of the spare memory, having a copy of data that was stored in a defective portion of a default memory, may be scanned.

[0070] The foregoing disclosure describes memory modules having data buffers with data selectors. When all default memories on a memory module are functional, data selectors on the memory module may select data nibbles from the default memories. When a default memory on the memory module is defective, the data selectors may select a data nibble from a spare memory and may not select a data nibble from the defective default memory.
We claim:

1. A data buffer comprising:
   a first data selector to select between a first data nibble stored in a first memory and a second data nibble stored in a second memory; and
   a second data selector to select between a third data nibble stored in a third memory and the second data nibble; wherein:
   when the first memory and the second memory are functional, the first and second data selectors are to select the first and second data nibbles, respectively;
   when the first memory is not functional, the first and second data selectors are to select the second and third data nibbles, respectively;
   when the second memory is not functional, the first and second data selectors are to select the first and third data nibbles, respectively; and
   an output of the data buffer comprises data nibbles selected by the first and second data selectors.

2. The data buffer of claim 1, further comprising:
   a first register to store an output of the first data selector, wherein the output of the first data selector comprises a data nibble selected by the first data selector; and
   a second register to store an output of the second data selector, wherein the output of the second data selector comprises a data nibble selected by the second data selector; wherein:
   the first and second registers output their respective stored data nibbles; and
   the output of the data buffer comprises the data nibbles output by the first and second registers.

3. The data buffer of claim 1, wherein:
   the data buffer, first memory, second memory, and third memory are on integrated circuits on a memory module; and
selection control signals for the first and second data selectors are received from a system controller on the memory module.

4. A machine-readable storage medium encoded with instructions executable by a processor of a memory module for using a spare memory on the memory module, wherein a plurality of memories on the memory module comprise default memories and the spare memory, the machine-readable storage medium comprising:

   instructions to identify a defective portion of a first memory of the default memories, wherein a data nibble selected from the first memory is in a first position in an output of the memory module when the defective portion is functional;

   instructions to determine selection control signals for data selectors on the memory module such that a data nibble is selected from the spare memory and not from the first memory, wherein the data nibble selected from the spare memory is in a second position in the output of the memory module; and

   instructions to notify a system controller, on the memory module, of the selection control signals.

5. The machine-readable storage medium of claim 4, further comprising instructions to transmit the selection control signals from the system controller to the data selectors, wherein in response to receiving the selection control signals, the data selectors select data nibbles from default memories other than the first memory such that data nibbles between the first position and the second position in the output of the memory module are shifted relative to an output of the memory module produced when all data nibbles are selected from default memories.

8. The machine-readable storage medium of claim 4, further comprising:

   instructions to correct erroneous data read from the defective portion;

   instructions to write the corrected data into a second memory of the default memories; and
instructions to transfer data from a non-defective one of the default memories to the spare memory.

7. The machine-readable storage medium of claim 4, further comprising:
   instructions to notify a memory controller external to the memory module that a data nibble is selected from the spare memory instead of the first memory, wherein the spare memory stores a copy of data stored in the defective portion of the first memory; and
   instructions to correct erroneous data in the spare memory.

8. A system for memory sparing, the system comprising:
   a plurality of memories comprising default memories and a first spare memory;
   a plurality of data buffers to select data nibbles from the plurality of memories such that when a first default memory of the default memories is identified as defective, a data nibble is selected from the first spare memory and not from the first default memory, wherein:
   a data nibble selected from the first default memory is in a first position in an output of the system when the first default memory is functional; and
   the data nibble selected from the first spare memory is in a second position in the output of the system; and
   a system controller to transmit selection control signals to the plurality of data buffers.

9. The system of claim 8, wherein:
   the output of the system comprises outputs of the plurality of data buffers in a specified order;
   the outputs of the plurality of data buffers comprise data nibbles selected from the plurality of memories; and
   the plurality of data buffers are further to select data nibbles from default memories other than the first default memory such that data nibbles between the first position and the second position in the output of the system are
shifted relative to an output of the system produced when all data nibbles are selected from default memories.

10. The system of claim 8, wherein each of the plurality of data buffers is communicatively coupled to three of the plurality of memories, and wherein each of the plurality of data buffers is further to select data nibbles from two memories.

11. The system of claim 10, wherein:
   a first data buffer of the plurality of data buffers is communicatively coupled to the first default memory and to second and third default memories of the default memories;
   a second data buffer of the plurality of data buffers is communicatively coupled to the first spare memory;
   the first data buffer is to select data nibbles from the second and third default memories; and
   the second data buffer is to select a data nibble from the first spare memory.

12. The system of claim 8, wherein the plurality of memories are volatile memories having error-correcting code (ECC) functionality.

13. The system of claim 8, wherein a memory controller is notified that a data nibble is selected from the spare memory instead of the first memory.

14. The system of claim 8, wherein:
   the first spare memory stores a copy of data stored in the first default memory; and
   erroneous data in the first spare memory is corrected.

15. The system of claim 8, wherein:
   the plurality of memories further comprise a second spare memory;
   when a first data buffer of the plurality of data buffers receives, from the system controller, a first address corresponding to a location in the first
default memory, the plurality of data buffers are to select a data nibble from the first spare memory instead of the first default memory;

    when the first data buffer receives, from the system controller, a second address corresponding to a location in a second default memory of the default memories, the plurality of data buffers are to select a data nibble from the second spare memory instead of the second default memory; and

    the system controller is further to maintain records of defective default memories and respective spare memories from which data nibbles are selected instead.
**FIG. 1**

**FIG. 2**
FIG. 7
FIG. 8

- Processor

Machine-Readable Storage Medium
- Identify defective portion of default memory
- Determine selection control signals for data selectors
- Notify system controller, on memory module, of selection control signals
- Notify memory controller
- Transmit selection control signals from system controller to data selectors
- Correct erroneous data in spare memory
Identify defective portion of default memory

Determine selection control signals for data selectors

Notify system controller, on memory module, of selection control signals

FIG. 9
Determine location of error

Memory controller has knowledge of spare memory?

Yes

Correct erroneous data in spare memory

No

Correct erroneous data read from defective portion of memory

Copy corrected data to temporary location

Notify system controller of selection control signals

Transfer data from non-defective default memory to spare memory

FIG. 10
**A. CLASSIFICATION OF SUBJECT MATTER**

G11C 7/10(2006.01)

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

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G11C 7/10; G11C 16/26; G06F 12/00; G11C 29/44; G11C 16/06; G06F 11/00; G11C 29/04

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic database consulted during the international search (name of database and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: data selector, nibble, spare memory, defective, functional, buffer

**C. 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>
<tbody>
<tr>
<td>A</td>
<td>US 2013-0028035 Al (NO [IAQI KONO]) 31 January 2013 See paragraphs [0004], [0013], [0029H0032], [0046], [0074]; claim 1; and figures 1-4.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2012-0069659 Al (JIN-MAN HAN et al.) 22 March 2012 See paragraphs [0041], [0050]-[0052], [0073], [0079H0081], [0102]; and figure 3.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2008-0126876 Al (YUN-SANG LEE) 29 May 2008 See paragraphs [0012H0015], [0021], [0053]-[0057], [0137]; claim 1; and figure 1.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2011-0164450 Al (ROBERT NORMAN) 07 July 2011 See paragraphs [0003]-[0005], [0019H0022], [0027]-[0032], [0036]-[0038]; and figures 1, 3, 5.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2005-0283581 Al (CHEN MIN CHIANG) 22 December 2005 See paragraphs [0016H0019], [0024]-[0027]; and figure 1.</td>
<td>1-15</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent but published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

* document member of the same patent family

Date of the actual completion of the international search: 23 June 2014 (23.06.2014)

Date of mailing of the international search report: 25 June 2014 (25.06.2014)

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
KTM, Seong Woo

Telephone No. +82-42-481-3348

Form PCT/ISA/210 (second sheet) (July 2009)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>us 2013-0028035 Al</td>
<td>31/01/2013</td>
<td>JP 2013-030238 A</td>
<td>07/02/2013</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8644097 B2</td>
<td>04/02/2014</td>
</tr>
<tr>
<td>us 2012-0069659 Al</td>
<td>22/03/2012</td>
<td>us 2010-020609 Al</td>
<td>28/01/2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 2011-019474 Al</td>
<td>27/01/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 7821830 B2</td>
<td>26/10/2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8081511 B2</td>
<td>20/12/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8379448 B2</td>
<td>19/02/2013</td>
</tr>
<tr>
<td>us 2008-0126876 Al</td>
<td>29/05/2008</td>
<td>JP 2008-130221 A</td>
<td>05/06/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 10-0877701 Bl</td>
<td>08/01/2009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20080046826 A</td>
<td>28/05/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TW 200823909 A</td>
<td>01/06/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TW 1344651 B</td>
<td>01/07/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 7535780 B2</td>
<td>19/05/2009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 2011-0007589 Al</td>
<td>13/01/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 7796451 B2</td>
<td>14/09/2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 7903485 B2</td>
<td>08/03/2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>us 8270238 B2</td>
<td>18/09/2012</td>
</tr>
<tr>
<td>us 2005-0283581 Al</td>
<td>22/12/2005</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>