Title: CIRCUITS, APPARATUSES, AND METHODS FOR CORRECTING DATA ERRORS

Abstract: Circuits, apparatuses, and methods are disclosed for correcting data errors in integrated circuits. One example apparatus includes a first circuit configured to receive first and second data words, to correct one or more errors in the first data word, to merge the corrected first data word and the second data word responsive to a control signal to generate a final merged data word, and to provide the final merged data word to a write circuit. The apparatus also includes a second circuit configured to receive the first and second data words, to preliminarily merge the first and second data words responsive to the control signal to generate an initial merged data word, to generate an initial parity code for the initial merged data word, to correct the initial parity code, and to provide the corrected parity code to the write circuit.
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG). Published: — with international search report (Art. 21(3))
CIRCUITS, APPARATUSES, AND METHODS FOR CORRECTING DATA ERRORS

TECHNICAL FIELD

[001] Embodiments of the invention relate generally to integrated circuits, and more particularly, in one or more of the illustrated embodiments, to correcting data errors in integrated circuits.

BACKGROUND OF THE INVENTION

[002] As semiconductor processing technology improvements continue to allow for smaller and smaller device sizes, a number of issues arise. For example, while smaller storage elements in volatile or non-volatile memory allow for greater storage density and lower costs per unit of data storage, the small size of the individual storage elements can lead to an increase in data errors, such as various refresh time errors (also known as various retention time errors), soft errors, single event upset errors, and so forth. In some examples, physical damage to the storage elements (e.g., due to improper manufacturing of the element) can also lead to data errors, but the errors may be distributed such that utilizing repair regions may be ineffective. These soft or hard data errors corrupt data and are typically randomly distributed.

[003] One method for mitigating the effects of these data errors is to store error correction code ("ECC") parity information (hereinafter referred to as "parity code") together with the data, and to decode the parity code when the data is read out in order to detect and/or correct any errors in the original data or the added parity code. The parity code requires some overhead both in the amount of data stored (due to the extra bits for the parity code) and in the amount of processing time required for writing to and reading from the data storage device (due to the processing time to encode and decode the ECC parity code). Nonetheless, using ECC is still generally considered to be effective and efficient in detecting and correcting randomly distributed data errors.

[004] Some data storage devices support a data mask option whereby during a write operation, one or more portions of a stored data word retain previous data values while one or more other portions of the data word are updated with new data values. As used herein, a data word refers to a plurality of bits of information, which are normally
handled or stored together as a unit, although a data mask option may cause one or more portions of the data word to be handled or stored differently from the other portion(s) of the data word. For example, in a memory, a data word stored at a particular address may include 128 bits of data, and the data mask option during a memory write operation may prevent one or more 8 bit portions of the data word from being updated from the write operation. As an even more specific example, the last 8 bits of the 128 bit data word may be masked in that only the first 120 bits of data will be written to the memory array, with the last 8 bits retaining whatever value they had prior to the write operation.

[005] Implementing ECC in a storage device without a data mask option during a write operation is fairly straightforward because the data word and parity code to be written are not dependent on prior stored data, but are entirely a function of the new data to be written to the device. However, in a data storage device with a data mask option available, the masked portions of the data word may contain one or more errors that need to be corrected before that old data can be properly merged with the new data as required by the data mask. Also, in order to properly encode the parity code for the merged data word during the write with data mask operation, the masked portion(s) should be taken into consideration by the ECC encoder, and those masked portions should be corrected before the ECC encoder can generate the new parity code. In other words, the parity code for the merged data word is dependent not only on the new portions of the data word but also on the portions of the corrected old data word that are being masked.

[006] In order to address this dependency, one solution may be to read out the old data word and old parity code, decode the old parity code, correct any errors in the old data word based on the decode, and then, subsequently, merge the new portion of the data word with the corrected old portions of the data word, generate a new parity code, and write the merged data word with the new parity code to the storage device. This sequential order of operations, however, may be unacceptable in some instances because the ECC decode and ECC encode operations can be relatively long and the ECC encode operation waits for the ECC decode and correction operation to complete before the new parity code can be generated and stored.
Another approach to address the data mask option in a device with ECC may be to have separate parity codes for each portion of the data word amenable to being masked. For example, continuing with the illustration of a 128 bit data word in which each 8 bit portion may be masked, each 8 bit portion of the data word may be assigned its own parity code, independent from the parity codes of each of the other 8 bit portions. When a write with data mask operation is performed, the masked portions can undergo ECC decode and correction independently of the ECC encode of the new, non-masked portions. In this approach, while sequential ECC operations are avoided, the amount of overhead in both storage space and processing power is increased. Using the 128 bit example, each 8 bit portion may have a parity code that is 4 bits long, thus requiring a total of 64 parity bits (50% overhead), and 16 independent ECC encode/decode circuits may be needed to separately encode or decode the parity codes for each 8 bit portion. In comparison, for the sequential approach described above, only 8 parity bits may be required for the 128 bit data word (6.25% overhead), and a single ECC circuit may both decode/correct and encode the parity codes, because the ECC encode operation cannot proceed until the ECC decode operation is complete.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an apparatus for correcting data errors according to an embodiment of the present invention.

FIG. 2 is a block diagram of an apparatus for correcting data errors according to an embodiment of the present invention.

FIG. 3 is a block diagram of an apparatus for correcting data errors according to an embodiment of the present invention.

FIG. 4 is a flow diagram for correcting data errors according to an embodiment of the invention.

FIG. 5 is a block diagram of a memory according to an embodiment of the invention.
DETAILED DESCRIPTION

[013] Certain details are set forth below to provide a sufficient understanding of embodiments of the invention. However, it will be clear to one skilled in the art that embodiments of the invention may be practiced without these particular details. Moreover, the particular embodiments of the present invention described herein are provided by way of example and should not be used to limit the scope of the invention to these particular embodiments. In other instances, well-known circuits, control signals, timing protocols, and software operations have not been shown in detail in order to avoid unnecessarily obscuring the invention.

[014] FIG. 1 illustrates an apparatus 100 for correcting data errors, according to an embodiment of the invention. As used herein, apparatus may refer to, for example, an integrated circuit, a memory device, a memory system, an electronic device or system, a smart phone, a tablet, a computer, a server, etc. In one example, the apparatus 100 includes a data storage device, such as an array 106 of volatile or non-volatile memory. In other embodiments, the data storage device may be an intermediate storage device, such as a buffer or a cache. The apparatus 100 is configured to store a plurality of data words at respective addresses within the array 106, and to read stored data words from the array 106. In one example, each data word may include 128 bits of data, whereas in other examples, each data word may include 32 data bits, 64 data bits, 256 data bits, 512 data bits, and so forth. The data words stored in the array 106 may be subject to soft or hard data errors - for example, one or more data bits within the data words may "flip" and change data values in between the time that the data bits are stored to the array 106 and the time when they are read. In other examples, one or more storage devices in the array 106 may be damaged, thereby preventing the correct storage and reading of one or more data bits. In general, any mechanism may cause one or more bits in data words read from the array 106 to differ from how those respective bits were supposed to have been written into the array 106.

[015] In order to mitigate the effects of such data errors, the apparatus 100 includes an error correction and/or detection code ("ECC") system 102, whereby a parity code (which may include one or more parity bits) are generated for each data word and stored in the array 106. ECC, as used herein, includes any type of error detection and/or error correction methodology - for example, the ECC system 102 of the
apparatus 100 may in some instances detect one or more data errors and may additionally correct one or more errors in some embodiments. The ECC system 102 may be based on a Hamming code (including an extended Hamming code), a cyclic redundancy check, or any other error detection and/or error correction methodology. The parity code generated by the ECC system 102 and stored in the array 106 with the associated data words represent information about some portion(s) or the entirety of the associated data word, and may subsequently be used to detect and/or correct errors in the data word when the data word is read out from the array 106. For brevity, an example ECC system 102 and code that can detect and correct a single bit data error in a data word will be described herein, but it will be appreciated that the principles described herein can be extended to detecting and/or correcting any number of data errors.

[016] Referring again to FIG. 1, the ECC system 102 includes a first ECC circuit 120 and a second ECC circuit 140 that may generally operate in parallel. The first ECC circuit 102 receives a first data word DATA-OLD and a parity code PARITY-OLD associated with the first data word DATA-OLD. The first data word DATA-OLD may be an old data word previously stored and read out from the array 106. While it was stored in the array 106, or as it was being stored into the array 106, one or more data bits of the first data word DATA-OLD may have changed values from their intended values, and thus, the first data word DATA-OLD provided to the first ECC circuit 120 may have one or more bits of data errors. The first ECC circuit 120 also receives a second data word DATA-NEW, which may be a new data word, of which at least some portions are to be stored in the array 106 in place of respective portions of the first data word DATA-OLD (e.g., at least some portions of the second data word DATA-NEW are to overwrite respective portions of the first data word DATA-OLD).

[017] The first ECC circuit 120 is configured to selectively correct one or more data errors in the first data word DATA-OLD based on the first data word DATA-OLD and its associated parity code PARITY-OLD when one or more data errors are detected in the first data word DATA-OLD. The corrected data word may correspond to the data word that was supposed to have been stored in the array 106. During operation of the ECC system 102, the first data word DATA-OLD provided to the first ECC circuit 120 may not always have one or more data errors. In fact, in some examples, relatively few
of the first data words DATA-OLD read out from the array 106 and provided to the first ECC circuit 120 will have any data errors at all. It will be understood that, although the first ECC circuit 120 is configured to correct one or more data errors in the data words DATA-OLD provided to it when such errors are detected by the first ECC circuit 120, when a data word DATA-OLD provided to the first ECC circuit 120 does not have any data errors, the first ECC circuit 120 will not "correct" the data bits in the data word, as there are no errors to correct.

[018] The first ECC circuit 120 is also configured to selectively merge the corrected data word (i.e., the first data word as corrected by the first ECC circuit or the uncorrected first data word if no data errors were detected in the same) with the second data word DATA-NEW responsive to a control signal, such as a data mask signal DATAMASK. The first ECC circuit 120 may generate a final merged data word DATA-MERGED based on one or more portions of the corrected data word and one or more portions of the second data word DATA-NEW, with the control signal DATAMASK determining which portions of which data word are used to generate the final merged data word DATA-MERGED. In other words, the control signal DATAMASK may cause the first ECC circuit 120 to retain one or more portions of the corrected data word from being replaced by one or more corresponding portions of the second data word DATA-NEW during a data write operation.

[019] In some cases, responsive to the control signal DATAMASK, the entirety of one of the corrected data word or the new data word DATA-NEW may be provided as the final merged data word DATA-MERGED - the control signal DATAMASK may entirely mask the corrected data word such that the final merged data word DATA-MERGED is identical to the corrected data word, or, on the other hand, the control signal DATAMASK may not mask any of the corrected data word such that the final merged data word DATA-MERGED is identical to the second data word DATA-NEW. However, in other examples, the control signal DATAMASK may mask one or more portions of the corrected data word such that those masked portions are provided in their respective places in the final merged data word DATA-MERGED, with the other portions of the final merged data word DATA-MERGED coming from respective portions of the second data word DATA-NEW.
The portions of the data words that are masked by the control signal DATAMASK may in some examples be 8 bit (1 byte) portions, but may generally be any size. If the data words are 128 bits long and the maskable portions are 8 bits long, for example, there may be 16 separate portions that may or may not be masked, based on the control signal DATAMASK.

Referring again to FIG. 1, the first ECC circuit 120 may provide the final merged data word DATA-MERGED to a write circuit 160, which may subsequently write the final merged data word DATA-MERGED to the array 106 together with an associated parity code PARITY-CORRECTED, as described in more detail below.

The second ECC circuit 140 may be separate and distinct from the first ECC circuit 120 in some embodiments, and may operate at least partially contemporaneously with the first ECC circuit 120. In other words, the time during which the second ECC circuit 140 operates as described herein may at least partially overlap the time during which the first ECC circuit 120 operates as described herein - e.g., such that one of the circuits 120, 140 begins its respective operation(s) before the other of the circuits 120, 140 has finished its respective operation(s). This parallel operation of the first and second ECC circuits 120, 140 may in some examples improve the timing of reading from and/or writing to the array 106 as compared to an apparatus in which the operations of similar first and second ECC circuits operate in series in that the second ECC circuit does not begin operation until the first ECC circuit has completed operations (of course, in a sequential operation, only a single ECC circuit may be needed).

The second ECC circuit 140 receives the first data word DATA-OLD and the second data word DATA-NEW, but does not receive the corrected first data word from the first ECC circuit 120. The second ECC circuit 140 preliminarily merges the first data word DATA-OLD and the second data word DATA-NEW responsive to the control signal DATAMASK and thereby generates an initial merged data word. The control signal DATAMASK provided to the first ECC circuit 120 is generally the same as the control signal DATAMASK provided to the second ECC circuit 140. The result of the preliminary merge is an 'initial' merged data word because any portions of the first data word DATA-OLD that are merged into the initial merged data word may include one or more data errors because the input to the second ECC circuit 140 is
"raw" data words DATA-OLD from the array 106 that may include one or more data errors.

[024] The second ECC circuit 140 also generates an initial parity code associated with the initial merged data word, and corrects the initial parity in a corrector circuit 146. The corrected parity code PARITY-CORRECTED is provided by the second ECC circuit 140 to the write circuit 160, which, as mentioned above, writes the corrected parity code PARITY-CORRECTED to the array 106 together with the final merged data word DATA-MERGED. The corrected parity code may be representative of the entire final merged data word DATA-MERGED in some examples, or in other examples may be representative of two or more portions (e.g., with each portion corresponding to a section of the data word that may be masked) of the final merged data word DATA-MERGED.

[025] The initial parity code may need to be corrected because it was generated based on the initial merged data word, which may contain one or more data errors. If the initial merged data word contains one or more errors, as compared to the final merged data word DATA-MERGED generated by the first ECC circuit 120, the initial parity code generated from the initial merged data word will have one or more errors as compared to what a parity code for the final merged data word DATA-MERGED should have. In other words, the second ECC circuit 140 corrects the initial parity code such that the corrected parity code PARITY-CORRECTED is the parity code for the final merged data word DATA-MERGED. As described in more detail below, in those instances where there are no data errors in the portions of the first data word DATA-OLD that are merged into the final merged data word DATA-MERGED, the initial parity code will not need to be corrected, and will simply be passed through the corrector circuit 146 as the corrected parity code PARITY-CORRECTED.

[026] Operating the first and second ECC circuits 120, 140 in parallel allows the first parity circuit 120 to correct a data error(s) in the first data word DATA-OLD while at least partially contemporaneously allowing the second ECC circuit 140 to generate the initial parity code as if there were no errors in the portions of the first data word DATA-OLD that are merged into the final data word DATA-MERGED. Because these two operations are relatively lengthy, and the correction of the initial parity code is relatively quick, the read-modify-write of the write with data mask operation can be
more quickly and efficiently processed than in the sequential ECC decode and subsequent ECC encode procedure described above.

FIG. 2 illustrates an apparatus 200 for correcting data errors, according to an embodiment of the invention. Like the apparatus 100 illustrated in FIG. 1, the apparatus 200 illustrated in FIG. 2 includes an ECC system 102 with first and second ECC circuits 120, 140, and a write circuit 160. Furthermore, the first ECC circuit 120 in FIG. 2 includes an ECC decode and correction circuit 122 and a first selection circuit 134. The ECC decode and correction circuit 122 receives the first data word DATA-OLD and its associated parity code PARITY-OLD, decodes the parity code PARITY-OLD and the first data word DATA-OLD to determine if there are one or more data errors in the first data word DATA-OLD, and, if so, corrects those one or more errors to provide the corrected data word DATA-OLD-CORRECTED.

The first selection circuit 134 in FIG. 2 receives the corrected data word DATA-OLD-CORRECTED and the second data word DATA-NEW, along with a control signal DATAMASK. The first selection circuit 134 may include a plurality of multiplexers which, depending on the control signal DATAMASK, select between portions of each of the corrected data word DATA-OLD-CORRECTED and the second data word DATA-NEW in order to generate the final merged data word DATA-MERGED.

The second ECC circuit 140 in FIG. 2 includes a second selection circuit 142, an ECC encode circuit 144, and a parity correction circuit 146. The second selection circuit 142 receives the first data word DATA-OLD and the second data word DATA-NEW, along with the control signal DATAMASK. The second selection circuit may include a plurality of multiplexers which, depending on the control signal DATAMASK, select between portions of each of the first data word DATA0OLD and the second data word DATA-MASK in order to preliminarily generate the initial merged data word DATA-MERGED-INITIAL. The initial merged data word DATA-MKRGED-iNITIAL may include one or more data errors because the first data word DATA-OLD was provided without first passing through the ECC decode and correction circuit 122 of the first ECC circuit 120.

The initial merged data word DATA-MERGED-INITIAL is then provided to the ECC encode circuit 144, which generates an initial parity code PARITY-INITIAL
based on the initial merged data word DATA-MERGED-INITIAL. The initial parity code PARITY-INITIAL is in turn provided to the parity correction circuit 146, which corrects the initial parity code in the event that there are one or more data errors in the masked portions of the first data word DATA-OLD, and provides the corrected parity code PARITY-CORRECTED to the write circuit 160, which in turn writes the corrected parity code PARITY-CORRECTED along with the final merged data word DATA-MERGED from the first ECC circuit 120 to the array 106.

[031] Thus the second ECC circuit 140 merges the first and second data words DATA-OLD and DATA-NEW and generates an associated initial parity code as if there were no errors in the initial merged data word DATA-MERGED-INITIAL, or as if the initial merged data word DATA-MERGED-INITIAL was going to be written to the array 106 with the initial parity code PARITY-INITIAL. However, as described above, the parity correction circuit 146 corrects that initial parity code PARITY-INITIAL to bring it into alignment with what the parity code should be for the final merged (and corrected) data word DATA-MERGED.

[032] FIG. 3 illustrates an apparatus 300 for correcting data errors, according to an embodiment of the invention. Like the apparatus 100 illustrated in FIG. 1 and the apparatus 200 illustrated in FIG. 2, the apparatus 300 illustrated in FIG. 3 includes an ECC system 102 with first and second ECC circuits 120, 140, and a write circuit 160. Furthermore, like the first ECC circuit 120 in FIG. 2, the first ECC circuit 120 in FIG. 3 includes an ECC decode and correction circuit 122, and a first selection circuit 134. The ECC decode and correction circuit 122 in FIG. 3 includes a syndrome generation circuit 124 that receives the first data word DATA-OLD and its associated parity code PARITY-OLD, and generates a syndrome SYNDROME representative of one or more data errors in the first data word DATA-OLD as received. The syndrome SYNDROME is provided to the error locator circuit 126, and as described in more detail below, is also provided to the parity correction circuit 146.

[033] The error locator circuit 126 decodes the syndrome to provide physical location(s) DECODED within the first data word DATA-OLD of one or more data errors. The error locator circuit 126 may be a typical address decoder in some embodiments. If, for example, the syndrome is 8 bits long and is indicative of a single data bit error, the error locator circuit may decode the 8 bit syndrome and provide a 128
bit long DECODED signal, with all values of the DECODED signal being binary 0 other than the location in the first data word DATA-OLD where there is a data error, which may have a value of binary 1.

[034] The physical location(s) DECODED of the data errors are provided, together with the first data word DATA-OLD to the data correction circuit 128, which corrects the one or more data errors. In some examples, the data correction circuit 128 may include a plurality of XOR gates, each of which receives as its inputs one bit of the first data word DATA-OLD and a corresponding physical location DECODED. Continuing on with the example of the 8 bit syndrome and 128 bit long DECODED signal, if there is an error in the first data word DATA-OLD, the location of that error would be given by a binary 1 in the DECODED signal and, when that binary one is XORed with the incorrect value in the first data word DATA-OLD, the value at that location would be restored to its correct binary value.

[035] The data correction circuit 128 thus provides the corrected data word DATA-OLD-CORRECTED to the first selection circuit 134, which also receives the second data word DATA-NEW and the control signal DATAMASK, and provides the final merged data word DATA-MERGED, as described above with reference to FIG. 2.

[036] Also like the apparatus 200 illustrated in FIG. 2, the apparatus 300 illustrated in FIG. 3 includes a second ECC circuit 140, which in turn includes a second selection circuit 142, an ECC encode circuit 144, and a parity correction circuit 146. The parity correction circuit 146 may in some embodiments include a plurality of XOR logic gates 152 configured to receive as input the initial parity code PARITY-INITIAL and the syndrome SYNDROME from the syndrome generating circuit 124. By comparing (e.g., XORing) the initial parity code PARITY-INITIAL with the syndrome SYNDROME, the parity correction circuit 146 may correct the initial parity code PARITY-INITIAL so that it corresponds with the corrected and final merged data word DATA-MERGED. As mentioned above, the syndrome SYNDROME may contain information regarding the location of a data error in the first data word DATA-OLD, which can be used to correct that data error through, for example, an XOR gate in the data correction circuit 128. Because of the way in which the syndrome and the parity codes are generated, that same location information in the syndrome SYNDROME may also be used to locate and correct an incorrect value of the initial parity code PARITY-INITIAL in the
parity correction circuit 146 through, for example, an XOR gate 152, because the incorrect value in the initial parity code PARITY-INITIAL results from the same data error in the first data word DATA-OLD that is corrected in the data correction circuit 128.

[037] In some embodiments, a signal (not shown), may be provided to the parity correction circuit 146 to determine whether the initial parity code PARITY-INITIAL should be passed along as-is to the write circuit 160 as the corrected parity code PARITY-CORRECTED, or whether one or more data bits need to be changed in the initial parity code PARITY-INITIAL first. For example, if there are no data errors in the first data word DATA-OLD, particularly if there are no data errors in the masked portions of the first data word DATA-OLD, then the initial parity code PARITY-INITIAL may not need to be modified. Thus, the parity correction circuit 146 may in some embodiments make a determination of whether there is a data error in the masked portions of the first data word DATA-OLD at all, and if not, then just pass the initial parity code PARITY-INITIAL on to the write circuit 160. In other embodiments, however, the initial parity code PARITY-INITIAL may always be compared (e.g., XORed) together with the syndrome SYNDROME and, if there are no errors in the first data word DATA-OLD, then the syndrome SYNDROME will be all binary 0s - in which case an XOR function would simply pass the initial parity code PARITY-INITIAL without any changes to the write circuit 160 in any event.

[038] The operation of the apparatuses 100, 200, 300 in FIGS. 1, 2, and 3, will now be described with reference to FIG. 4, which illustrates a flow chart of operations 400 for correcting data errors during a write with data mask operation, according to an embodiment of the invention.

[039] In operation 402, a first data word DATA-OLD (and its associated parity code PARITY-OLD) is read from array 106, and in operation 404, the first data word DATA-OLD and a second data word DATA-NEW are received by first and second ECC circuits 120, 140. In operation 410, the syndrome generator 124 of the first ECC circuit 120 generates a syndrome SYNDROME based on the first data word DATA-OLD and its associated parity code PARITY-OLD. In operation 412, the syndrome SYNDROME is decoded to provide locations DECODED of one or more data errors in the first data word DATA-OLD. In operation 414, the DECODED location
information is used to correct the first data word DATA-OLD, which is merged together with the second data word DATA-NEW in operation 416 based on a control signal DATAMASK. Operations 410-418 proceed at least partially contemporaneously with operations 420-424, which will now be described.

[040] In operation 420, the first data word DATA-OLD is preliminarily merged with the second data word DATA-NEW based on the control signal DATAMASK to generate the initial merged data word DATA-MERGE-INITIAL. In operation 422, an initial parity code is generated by encoding the initial merged data word DATA-MERGE-INITIAL. Then, in operation 424, the initial parity code is corrected based on the syndrome SYNDROME generated in operation 410 to generate the corrected parity code PARITY-CORRECTED. Then, in operation 430, the final merged data word DATA-MERGED is written together with the corrected parity code PARITY-CORRECTED to the array 106.

[041] With reference to operation 424, as mentioned above, before the initial parity code PARITY-INITIAL is corrected using the syndrome SYNDROME, in some examples, a check is first made to determine whether there is a data error in the first data word DATA-OLD at all (or at least those portions that are being masked), and if not, then no changes are made to the initial parity code PARITY-INITIAL, but it is instead directly passed through as the corrected parity code PARITY-CORRECTED to write circuit 160. In other examples, however, no check is made, and the initial parity code PARITY-INITIAL is always "corrected," with the understanding that if no bits are incorrect, then the operation of correcting the code does not alter the data values of it.

[042] There are instances in which the initial parity code PARITY-INITIAL may not need to be corrected, for example, when there is no data error in the first data word DATA-OLD or its associated parity code PARITY-OLD, when there is only a data error in a portion of the first data word DATA-OLD which is not masked, and when there is only a data error on the parity code PARITY-OLD that is retrieved with the first data word DATA-OLD. In these instances, the initial parity code correctly corresponds to the final merged data word DATA-MERGED, and no corrections are needed. In another instance in which there is a data error in a portion of the first data word DATA-OLD which is being masked, the initial parity code PARITY-INITIAL
will be incorrect and thus may be corrected by the parity correction circuit 146 so that it
properly corresponds with the final merged data word DATA-MERGED. In this last
situation, the syndrome SYNDROME may determine the location of not only the error
in the first data word DATA-OLD but may also determine the location of the incorrect
bit or bits in the initial parity code PARITY-INITIAL.

FIG. 5 illustrates a portion of a memory 500 according to an embodiment of the
present invention. The memory 500 includes an array 501 of memory cells, which may
be, for example, DRAM memory cells, SRAM memory cells, flash memory cells, or
some other types of memory cells, and may correspond to the array 106 shown in FIGS.
1, 2, and 3. The memory 500 includes a control circuit 506 that receives memory
commands and addresses through an ADDR/CMD bus. The control circuit 506
provides control signals, based on the commands received through the ADDR/CMD
bus. The control circuit 506 also provides row and column addresses to the memory
500 through an address bus and an address latch 510. The address latch then outputs
separate column addresses and separate row addresses.

The row and column addresses are provided by the address latch 510 to a row
address decoder 522 and a column address decoder 528, respectively. The column
address decoder 528 selects bit lines extending through the array 502 associated with
respective column addresses. The row address decoder 522 is connected to word line
driver 524 that activates respective rows of memory cells in the array 502 associated
with received row addresses. The selected data line (e.g., a bit line or bit lines)
associated with a received column address are coupled to a read/write circuit 530 to
provide read data to a data output circuit 534 via an input-output data bus 540. An
output pad 542 coupled to the data output circuit 534 is used for electrically coupling to
the memory 500. Write data are provided to the memory array 502 through a data input
circuit 544 and the memory array read/write circuit 530. An input pad 546 coupled to
the data input circuit 542 is used for electrically coupling to the memory 500. The
control circuit 506 responds to memory commands and addresses provided to the
ADDR/CMD bus to perform various operations on the memory array 502. In particular,
the control circuit 506 is used to provide internal control signals to read data from and
write data to the memory array 502.
In some embodiments, the read/write circuit 530 includes an ECC system 502, which may be for example the ECC system 102 illustrated in FIGS. 1, 2, and/or 3, and described herein. The ECC system 502 may thus receive data words from the memory array 501 with one or more errors, and, in the case of a write with data mask operation, correct those one or more errors if needed, as described above in more detail.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. For example, FIGS. 1, 2, 3, and 5 illustrate embodiments of ECC systems 102, 502, FIGS. 1, 2, and 3 illustrate embodiments of parity correction circuits 146, FIGS. 2 and 3 illustrate embodiments of selection circuits 134, 142, and so forth. However, other ECC systems, parity correction circuits, selection circuits, and so forth may be used, which are not limited to having the same design, and may be of different designs and include circuitry different from the circuitry in the embodiments illustrated in these figures.

While this disclosure has described the operation of an ECC system 102, 502 during a write with data mask operation, it will be understood that the ECC systems 102, 502 described herein generally function similar to traditional ECC systems during other operations. For example, during a read operation, the ECC systems 102, 502 read out the first data word DATA-OLD together with its associated parity code PARITY-OLD, and detect and/or correct any data errors in the first data word DATA-OLD. Similarly, for a write operation without a data mask, the ECC systems 102, 502 generate a parity code for the second data word DATA-NEW that can be stored together with the second data word DATA-NEW in the array 106.

Also, as previously mentioned, while this disclosure described an example in which an ECC code is used that is capable of detecting and correcting one bit of data error, other examples within the scope of this disclosure include ECC codes that can detect two bit errors and correct one bit error, can correct two bit errors, can merely detect a single bit error, and so forth. In general, any error detection and/or error correction methodology may be used.

Accordingly, the invention is not limited to the specific embodiments of the invention described herein.
CLAIMS

What is claimed is:

1. An apparatus, comprising:
   a first circuit configured to receive first and second data words, and configured to correct one or more errors in the first data word and merge the corrected first data word and the second data word responsive to a control signal to generate a final merged data word, the first circuit further configured to provide the final merged data word to a write circuit; and
   a second circuit configured to receive the first and second data words, and configured to preliminarily merge the first and second data words responsive to the control signal to generate an initial merged data word and generate an initial parity code for the initial merged data word, the second circuit further configured to correct the initial parity code and provide the corrected parity code to the write circuit.

2. The apparatus of claim 1, wherein the first and second circuits are configured to operate in parallel.

3. The apparatus of claim 1, wherein the control signal is a data mask signal configured to retain one or more portions of the corrected first data word from being replaced by one or more corresponding portions of the second data word during a data write operation.

4. The apparatus of claim 1, wherein the second circuit comprises a plurality of XOR logic gates that are configured to receive as input the initial parity code and a syndrome generated from the first data word and an associated parity code for the first data word.

5. The apparatus of claim 1, wherein the initial parity code is based on two or more portions of the initial merged data word.
6. The apparatus of claim 1, wherein the corrected parity code is representative of the entire final merged data word.

7. An apparatus, comprising:
   - an ECC decode and correction circuit configured to provide a corrected data word responsive to a first data word;
   - a first selection circuit configured to merge the corrected data word and a second data word responsive to a control signal and provide a final merged data word;
   - a second selection circuit configured to merge the first and second data words responsive to the control signal and provide an initial merged data word;
   - an ECC encode circuit configured to generate an initial parity code based on the initial merged data word; and
   - a parity correction circuit configured to correct the initial parity code responsive to a correction signal from the ECC decode and correction circuit.

8. The apparatus of claim 7, wherein the ECC encode circuit is configured to generate the initial parity code at least partially contemporaneously with the ECC decode and correction circuit providing the corrected data word.

9. The apparatus of claim 8, wherein the ECC decode and correction circuit comprises a syndrome generation circuit configured to generate a syndrome, an error locator circuit configured to decode the syndrome, and a data correction circuit configured to correct the first data word based on the decoded syndrome, further wherein the syndrome is provided as the correction signal to the parity correction circuit.

10. The apparatus of claim 9, wherein the ECC encode circuit is configured to generate the initial parity code at least partially contemporaneous with the syndrome generation circuit generating the syndrome.
11. The apparatus of claim 7, wherein the first and second selection circuits comprise one or more multiplexers configured to select between portions of input data words to generate their respective output data words.

12. The apparatus of claim 7, wherein the first selection circuit is configured to provide a final data word, and the final data word includes one or more portions of the first data word and one or more portions of the second data word.

13. A method, comprising:
merging a first data word with a second data word to generate an initial merged data word;
generating an initial parity code based on the initial merged data word;
generating a syndrome for the first data word; and
correcting the initial parity code to generate a final parity code.

14. The method of claim 13, further comprising:
correcting the first data word based on the syndrome in an ECC decode and correction circuit; and
merging the corrected first data word with a second data in a first selection circuit word to generate a final merged data word.

15. The method of claim 14, further comprising writing the final merged data word and the final parity code to a data storage device using a write circuit.

16. The method of claim 13, further comprising correcting the initial parity code based on the syndrome.

17. The method of claim 13, wherein the act of generating a syndrome for the first data word is performed at least partially contemporaneous with the act of generating the initial parity code.
18. The method of claim 13, wherein the first data word is an old data word previously stored in a data storage device, and the second data word is a new data word, of which at least some portions are to be stored in the data storage device.

19. The method of claim 18, wherein the old data word and the new data word are preliminarily merged responsive to a control signal that masks one or more portions of the old data word from being updated by the new data word.

20. The method of claim 13, wherein the initial parity code is generated using a single bit error detection, single bit error correction Hamming code in an ECC encode circuit.

21. A method, comprising:
   merging old and new data words responsive to a control signal to obtain an initial merged data word;
   encoding the merged data word to obtain an initial parity code;
   decoding the old data word responsive to an old parity code;
   correcting the old data word responsive to said decoding; and
   correcting the initial parity code responsive to said decoding.

22. The method of claim 21, wherein the acts of correcting the old data word and correcting the initial parity code are done responsive to a syndrome generated during the act of decoding the old data word responsive to the old parity code.

23. The method of claim 22, wherein one or more bits in the initial parity code are corrected in a corrector circuit based on a comparison of the initial parity code and the syndrome.

24. The method of claim 21, further comprising merging the corrected old data word and the new data word in a selection circuit responsive to the control signal.

25. The method of claim 24, wherein the control signal is a data mask signal.
26. The method of claim 24, further comprising receiving the old data word from a read circuit in a data storage device and writing the final merged data word and the corrected parity code to the data storage device using a write circuit.

27. The method of claim 21, wherein the initial parity code is configured to detect and correct at least one data bit error in the old data word.

28. The method of claim 21, wherein one or more bits in the initial parity code are corrected in a corrector circuit using a syndrome generated in an ECC decode circuit from the old data word and the old parity code.

29. The method of claim 21, wherein the acts of encoding and decoding are performed in parallel using distinct circuits.
FIG. 1

Diagram showing data flow through two ECC circuits, with connections labeled for DATA-OLD, DATA-NEW, PARITY-OLD, DATAMASK, DATA-MERGED, PARITY-CORRECTED, and WRITE CIRCUIT.

The diagram includes a box labeled 'CORRECTOR CIRCUIT' within the second ECC circuit.
FIG. 2
Fig. 4

1. Read old data
2. Receive new data and datamask
3. Generate syndrome based on old data
4. Decode syndrome
5. Correct old data
6. Merge corrected old data with new data based on datamask
7. Merge old data and new data based on datamask
8. Encode initial merged data
9. Correct initial parity code based on syndrome
10. Write merged, corrected data and corrected parity bits to array
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

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)

rPC: GHC

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>CN 101246748 A (SAMSUNG ELECTRONICS CO., LTD.) 20 August 2008 (20.08.2008) description, pages 2-6</td>
<td>1-29</td>
</tr>
<tr>
<td>A</td>
<td>CN104194363 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) 18 April 1990 (18.04.1990) the whole document</td>
<td>1-29</td>
</tr>
<tr>
<td>A</td>
<td>US2003103586 A1 (POEPPELMAN, Alan D. et al.) 05 June 2003 (05.06.2003) the whole document</td>
<td>1-29</td>
</tr>
<tr>
<td>A</td>
<td>US7954034 B1 (EMC CORPORATION) 31 May 2011 (31.05.2011) the whole document</td>
<td>1-29</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
06 March 2014 (06.03.2014)

Date of mailing of the international search report
27 Mar. 2014 (27.03.2014)

Name and mailing address of the ISA/CN
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088
facsimile No. 86-10-62019451

Authorized officer WANG Jing
Telephone No. (86-10)62413627

Form PCT/ISA/10 (second sheet) (July 2009)
## INTERNATIONAL SEARCH REPORT

### Information on patent family members

<table>
<thead>
<tr>
<th>Patent Documents referred in the Report</th>
<th>Publication Date</th>
<th>Patent Family</th>
<th>Publication Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN 101246748 A</td>
<td>20.08.2008</td>
<td>US 2008195919 A1</td>
<td>14.08.2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TW 200845021 A</td>
<td>16.11.2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20080075750 A</td>
<td>19.08.2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2008198330A</td>
<td>28.08.2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 920006998 B1</td>
<td>24.08.1992</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU 3928289 A</td>
<td>29.03.1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 68920523 T2</td>
<td>23.02.1995</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 0360424 A2</td>
<td>28.03.1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BR 8904735 A</td>
<td>01.05.1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 1309495 C</td>
<td>27.10.1992</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP H0295029 A</td>
<td>05.04.1990</td>
</tr>
<tr>
<td>US2003103586 A1</td>
<td>05.06.2003</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>US7954034B1</td>
<td>31.05.2011</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Form PCT/ISA /210 (patent family annex) (July 2009)