

Technical Disclosure Commons

Defensive Publications Series

October 2021

Product Error-Correcting Codes That Span NAND-Flash Dies

N/A

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

N/A, "Product Error-Correcting Codes That Span NAND-Flash Dies", Technical Disclosure Commons, (October 29, 2021)

https://www.tdcommons.org/dpubs_series/4682



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Product Error-Correcting Codes That Span NAND-Flash Dies

ABSTRACT

NAND-flash memories include a matrix of pages. Each column of the matrix is located in one physical semiconductor die. Because errors are correlated such that they occur in groups within a die, error-correcting codes (ECC) are optimally constructed across dies (matrix rows), a principle known as cross-die design. A product ECC is a type of code that encodes rows to one parity and columns to another. Although the row-constituents of product codes are cross-die, the column-constituents are not so. This disclosure describes product ECCs where both constituent codes span semiconductor dies. The described cross-die product codes provide better performance for random errors while maintaining performance comparable to traditional codes for correlated errors, at nearly the same coding overhead. A page in error can be repaired in two unique ways, both of which are cross-die, thereby improving data reliability and speed of repair.

KEYWORDS

- Solid state storage
- NAND flash
- Flash memory
- Error-correcting code (ECC)
- Product code
- Cross-die code
- Redundant array of independent disks (RAID)
- Correlated errors
- Repair group

BACKGROUND

Error-correcting codes (ECC) add redundancy to user data to provide reliable data storage. The redundant data, which is a function of user data, enables recovery of user data if part of the user data is lost, e.g., due to device failures, software errors, etc.

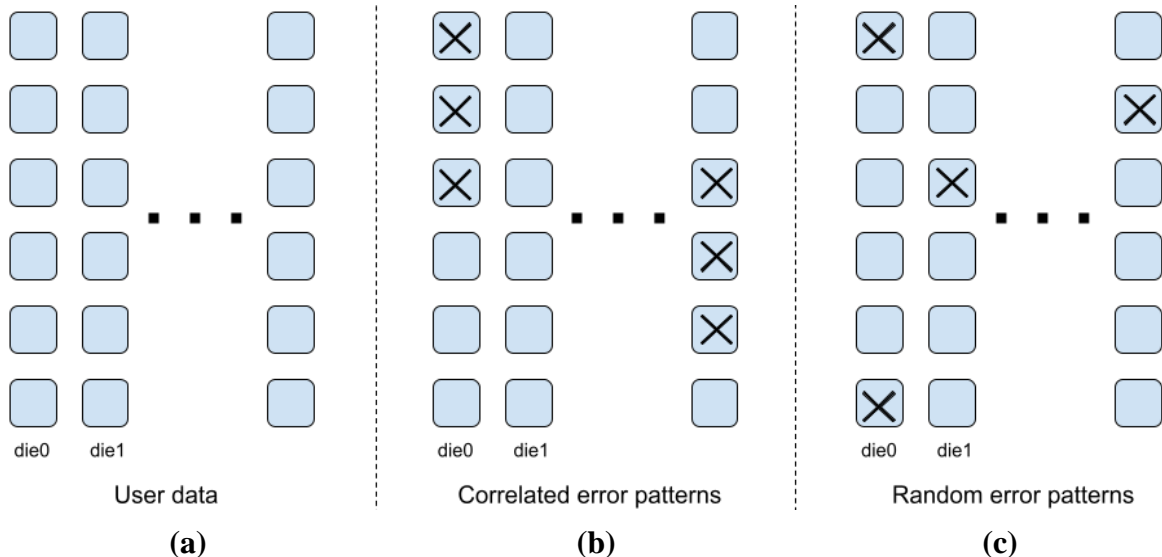


Fig. 1: Organization of a NAND flash

Fig. 1(a) illustrates the organization of a NAND flash, a type of memory. Each blue square represents user data of a certain size, e.g., four kilobytes, and can be referred to as a page of memory. A column of blue squares is physically located in one semiconductor die. Errors in NAND flashes can occur in various ways. In some cases, the errors can be correlated while in other cases they can be random. In a correlated error pattern, illustrated in Fig. 1(b), errors occur within one die. For example, the occurrence of an erroneous page predisposes nearby pages in the same die (column) to also be in error. In a random error pattern, illustrated in Fig. 1(c), errors occur at random (uncorrelated) locations. In this situation, the location of a particular error has no bearing on the location of other errors.



Fig. 2: Codewords in NAND flashes

Due to the somewhat higher occurrence of correlated errors (as opposed to random errors), pages are typically collected across dies, e.g., across columns, to form a single error-correcting codeword. For example, as illustrated in Fig. 2(a), each row of sixty-two pages of user data (blue) is protected by two pages of parity bits (pink), using, e.g., a RAID6 type (Reed-Solomon) error-correcting code (ECC), which can correct up to two errors. Error-correcting (redundant) data is known as parity data because one way to compute it is based on the parity (even or odd) of the number of ones in user data. If correlated errors occur, e.g., even a full column is erased, then, as illustrated in Fig. 2(b), the errors are distributed across rows, e.g., codewords, such that each codeword only faces only a single error, which is correctable. In contrast, if codewords were formed column-wise, e.g., on a per die basis, a correlated error pattern can introduce enough errors within a single die that the ECC of that die is not sufficient to recover from the errors.

Codewords that span dies are said to have a cross-die RAID (CDR) design. Simply put, CDR is the practice of not putting all redundancy data in the same die as the user data that it is

computed from. CDR designs result in NAND flashes that are more reliable than non-CDR designs, since non-CDR designs are less likely to lose redundancy data along with user data.

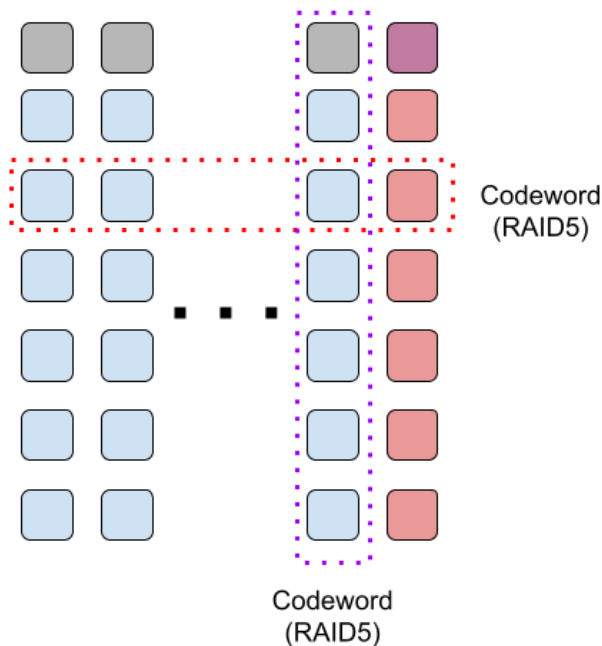


Fig. 3: Product codes

Fig. 3 illustrates a product code, another type of error-correcting arrangement that can be used in NAND flash memories. In a product code, each row is protected by a parity page (pink) and each column is protected by a parity page (grey). The row and column parities are themselves protected by a parity-of-parities (purple). Due to error protection in both X- and Y-directions, the product code can localize random (uncorrelated) errors and is resilient to row-wise correlated errors. It can also correct any pattern of three-page failures, but, unlike the CDR RAID6 code, cannot correct two-die failures, which, however, are relatively infrequent. To maintain an equivalent ECC overhead, e.g., ratio of redundant data to user data, the row and column parity checks each use a single parity page, e.g., based on RAID5 (XOR codes), which can correct up to one error.

Although the row codeword of the product code is of cross-die design, the column codeword is not. Although product codes have been proposed for NAND flash memories, there are thus far no product codes that are also fully cross-die, e.g., whose constituent codes are both of cross-die type.

DESCRIPTION

This disclosure describes product codes that are of cross-die design in both their constituent row codes and column codes. The described cross-die product codes provide a more efficient and viable alternative to traditional CDR RAID6 codes for page-level error correction in NAND flash memories.

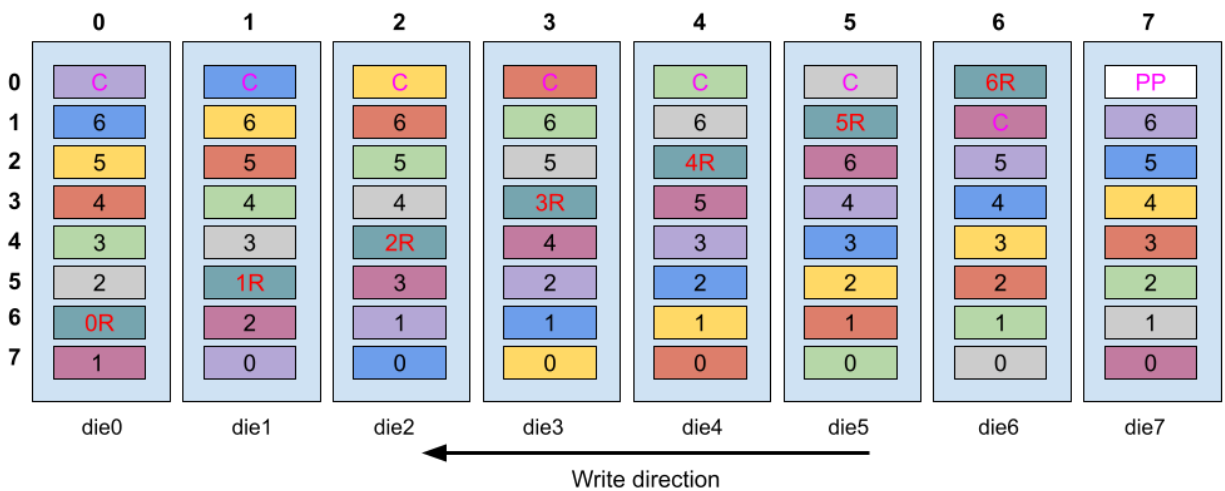


Fig. 4: A cross-die product code

Fig. 4 illustrates an example of a cross-die product code for memory pages in an 8x8 matrix. Pages of the same color correspond to one codeword. Pages comprising user data are marked with numeric identifiers 0 thru’ 6. A page with a ‘C’ in it indicates the column parity page for a codeword. For example, yellow-C is the column parity for user data in yellow-0 through yellow-6. Row parities are computed of the same numeric identifier over all colors. For example, the row parity 0R is formed by computing parity over yellow-0, peach-0, green-0, grey-

0, pink-0, purple-0, and blue-0. The page marked PP is the parity of the row-parities and the column-parities. Each die (column) includes exactly one page of each number and one page of each color.

Codewords are arranged in a modulo-diagonal fashion. Another way of describing the codeword arrangement is that they form stripes of color across the matrix of memory pages. For example, the grey codeword, comprising grey-0 through grey-6 followed by grey-C, starts with grey-0 at (row, column) location (7,6). Grey-1 is the next location along the diagonal modulo the dimensions of the matrix, e.g., $(7-1, 6+1)$ modulo 8 = (6, 7). Grey-2 is the next location along the diagonal, modulo the dimensions of the matrix, e.g., $(6-1, 7+1)$ modulo 8 = (5, 0).

When constructed as described above, neither the row-parity page nor the column-parity page of a given codeword fall on the same die as the user data pages of that codeword; thus, the product code is of fully cross-die design. Any seven pages of the same color can be used to recover the eighth page of that color. Any seven pages of the same number can be used to recover the eighth page of that number. Thus, as in a true product code, a given page has two *repair groups*, e.g., two sets of pages that enable the page to recover from failure. Unlike a conventional product code, *both* repair groups are of cross-die type.

Each page can be reconstructed in two unique, cross-die ways. For example, if the yellow-5 (on die0) is lost, it can be reconstructed in two cross-die ways, e.g.,

- (i) from all the yellow pages on die1 through die7 (the color repair group); and
- (ii) from all the pages labeled 5 on die1 through die7 (the number repair group).

The color repair group and the number repair group each have the cross-die property, resulting not only in higher reliability but also fast, parallelizable reads during the repair operation.

In contrast to traditional NAND ECCs, e.g., RAID5 or RAID6, a failed die can be repaired by using less than three-fourths of the remaining pages. This property, where a failed die can be recovered by reading only a subset of the remaining data, is known as locality. For example, suppose die0 has failed. Then pages 1, 0R, 2, 3 can be repaired using their color groups. But this will result in four of 4s, 5s, 6s and Cs already being read from the flash memory. Then 4, 5, 6, and C can be repaired from their respective number groups by just reading the remainder of their respective number groups. Therefore, the total reads are $7 \times 4 + 3 \times 4 = 40$ compared to $8 \times 7 = 56$ for RAID5. Thus, the described product codes are also faster than traditional ECCs in repairing errors. As mentioned earlier, an additional speed-of-repair advantage accrues from read operations during repair being fully parallel, since the pages used for the recovery of a given page reside in different dies.

Although the techniques are described for a matrix of NAND flash pages, they are applicable to any matrix of storage devices, e.g., disks, network storage, distributed (cloud) storage, etc. For example, the blue squares of Fig. 1(a) can each represent not NAND flash pages but large disks (e.g., of size 1 TB) that are known to suffer occasional patterns (e.g., row-wise, column-wise, random, etc.) of errors. A product code can be designed using the described techniques that is robust to the anticipated pattern of errors.

The described cross-die product code has better performance for random errors while maintaining performance comparable to traditional (RAID5, RAID6) codes for correlated errors, at nearly the same redundancy (coding overhead). A page in error can be repaired in two unique ways, both of which are cross-die, thereby improving data reliability and speed of repair.

CONCLUSION

This disclosure describes product ECCs where both constituent codes span semiconductor dies. The described cross-die product codes provide better performance for random errors while maintaining performance comparable to traditional codes for correlated errors, at nearly the same coding overhead. A page in error can be repaired in two unique ways, both of which are cross-die, thereby improving data reliability and speed of repair.

REFERENCES

- [1] Yang, Chengen, Yunus Emre, and Chaitali Chakrabarti. "Product code schemes for error correction in MLC NAND flash memories." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 20, no. 12 (2011): 2302-2314.
- [2] Wolf, J., and Bernard Elspas. "Error-locating codes—A new concept in error control." *IEEE transactions on information theory* 9, no. 2 (1963): 113-117.
- [3] Wolf, J. "On codes derivable from the tensor product of check matrices." *IEEE Transactions on Information Theory* 11, no. 2 (1965): 281-284.
- [4] Wolf, Jack K. "On an extended class of error-locating codes." *Information and control* 8, no. 2 (1965): 163-169.