AN5747

Application note

Page EEPROM memory architecture

Introduction

Page EEPROM is a memory device with a page architecture. This means that pages are the smallest erasable blocks of memory. Each page is made up of words and each word is composed of 16 bytes.

To increase reliability, Page EEPROM devices embed ECC (error correction code). The ECC is a specific logic designed to correct single bit or a double bit read errors and flag a triple bit error. This embedded ECC algorithm is transparent for the master bus and increases the data read robustness.

The purpose of this application note is to describe the STMicroelectronics Page EEPROM architecture in order to understand the possible differences occurring in specific use-cases.

Table 1. **Applicable products**

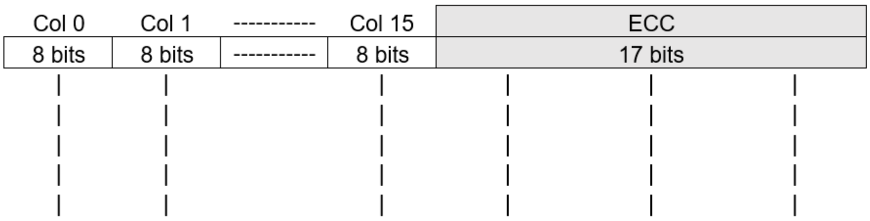
Series	Products
Standard Serial Page EEPROM	M95P32-E M95P32-I
	M95P16-E M95P16-I
	M95P08-E M95P08-I

1 Page EEPROM architecture

ST Page EEPROM devices are organized in 512 byte pages. Inside the page, the ECC bits are implemented on each group of 16 bytes. A group of 16 bytes is located at [16*N] addresses, where N is an integer. Each group being coupled with 17 ECC bits. These ECC bits are transparent for the user.

The 16 bytes plus the 17 ECC bits composing a word is illustrated in the figure below.

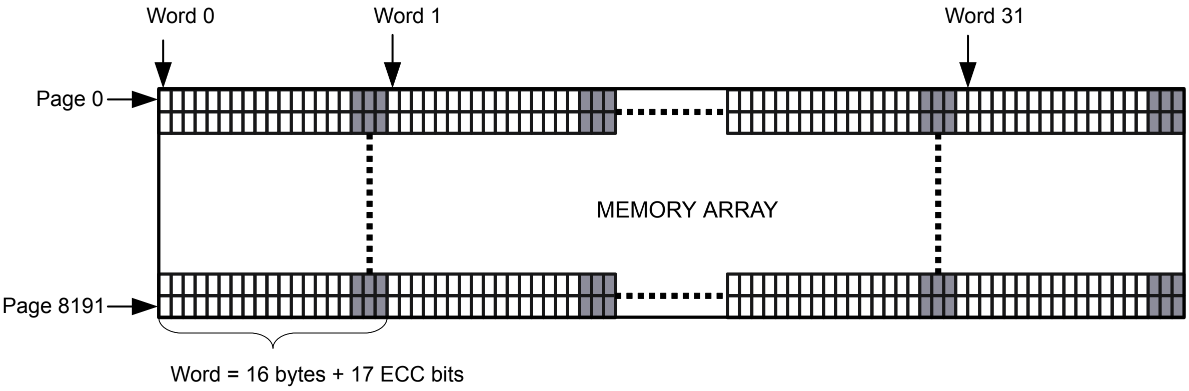
Figure 1. Word of 16 bytes plus 17 ECC bits



Col 0	Col 1	-----------	Col 15	ECC
8 bits	8 bits	-----------	8 bits	17 bits

The memory array represented in the figure below shows the memory architecture. ECC bits are standard non-volatile Page EEPROM bits. The ECC bits are linked to the value of the 16 data bytes in the word. Each time a data byte is erased, written (PGWR instruction) or programmed (PGPR instruction) the corresponding ECC bits are computed and updated. If a data byte is read, the corresponding ECC bits are also computed for error correction or detection. With the 17 ECC bits capability, up to two-bit error in the 16 data bytes can be corrected and three-bit error can be detected.

Figure 2. Memory array architecture of page EEPROMs with ECC



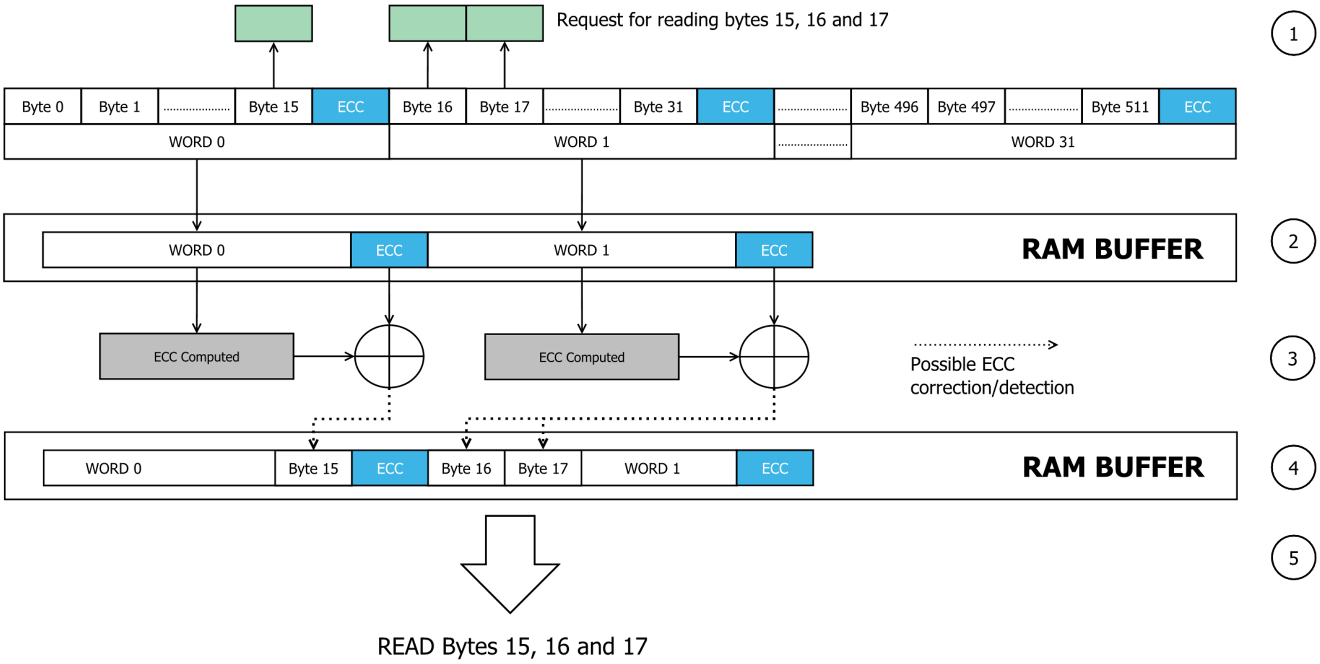
2 Page EEPROM operations

2.1 Read operation with word management

Besides the additional ECC bits in the memory array, the Page EEPROM has an additional logic block able to compute the ECC signature, to detect and correct possible bit errors in the data word. As the ECC signature (17 ECC bits) is a function of the 16 data bytes in the word, the word is always considered as the smallest entity in the read process of the Page EEPROM.

The figure below explains how the read process is managed.

Figure 3. Read process of a Page EEPROM with ECC



The steps of the read process illustrated in Figure 3 are described below:

1. The user requests a reading of three bytes: byte 15, byte 16, and byte 17.
2. Word 0 and word 1 (data bytes and ECC bits) are stored in the RAM buffer.
3. A new ECC signature is computed from the 16 data bytes read from the memory. Then, it is compared to the ECC bits signature read from the memory. Differences indicate errors:
   a. One difference indicates one error. The one-bit error is detected and corrected after further calculations.
   b. Two differences indicate two errors. The two-bits errors are detected and corrected after further calculations.
   c. Three differences indicate three errors. The bit errors are detected but not corrected.
4. The RAM buffer now contains byte 15, byte 16, byte 17 and the read ECC bits. If one or two bits failed to be read correctly, they are corrected.
5. Requested bytes are shifted out of the memory with the most significant byte first. Byte 15 is followed by byte 16 and by byte 17).

Note:

- *ECC correction or detection flag bits from safety register are updated according to bit correction or detection during the read command.*
- *If three bits failed are detected the requested bytes are shifted out of the memory without correction.*

2.2 Erase operations with page architecture

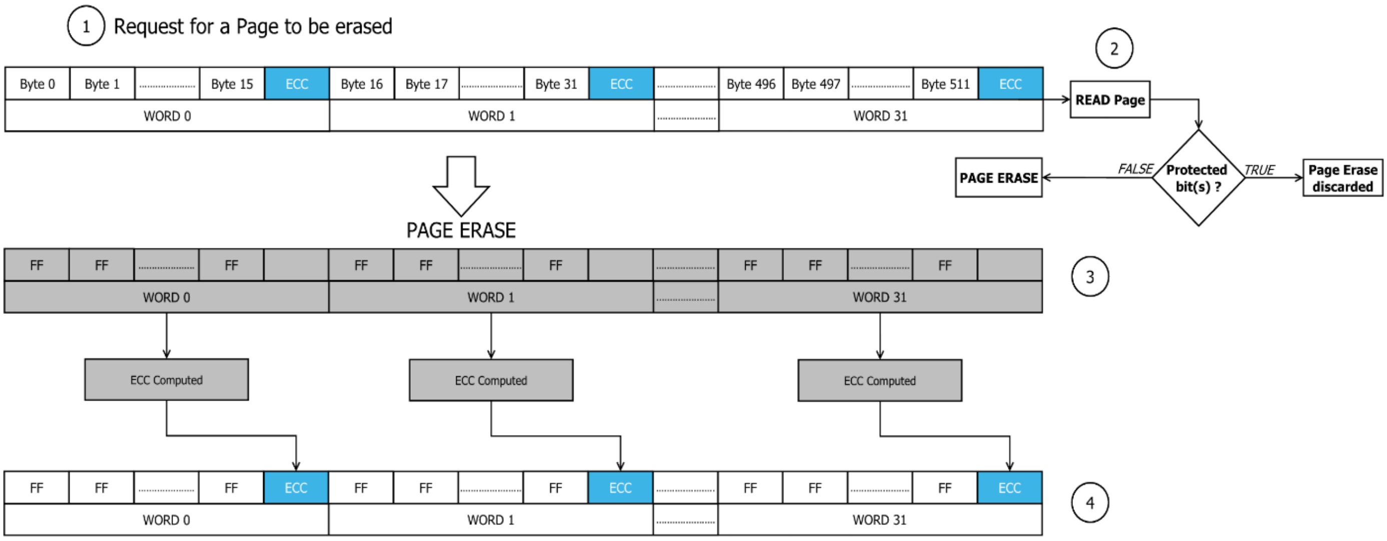
An erase operation set all requested bytes to 0xFF value. Page EEPROM erase operations are:

- page erase (512 bytes)
- sector erase (4K bytes)
- block erase (64k bytes)
- chip erase (32, 16 or 8 Mbit, according to the product datasheet)

Any page can be erased except protected ones. If one or more bytes of the targeted area are protected, the erase operation is not performed. After any erase operation, a new ECC signature is performed for each erased word.

The figure below illustrates how a page erase operation is performed.

Figure 4. Page erase process of a Page EEPROM with ECC



Note: *In a single instruction, sector erase sets eight pages to 0xFF and block erase sets 128 pages to 0xFF.*

The steps of the page erase process are illustrated in Figure 4 are described below:

1. The user sends a request to erase one page in the memory array.
2. The page (or requested area for sector, block, or chip erase operation) is read to check for any protected bit. If any bits are protected the erase operation is canceled.
3. The page is erased. All data bytes are set to 0xFF.
4. New ECC bits signatures are computed and updated.

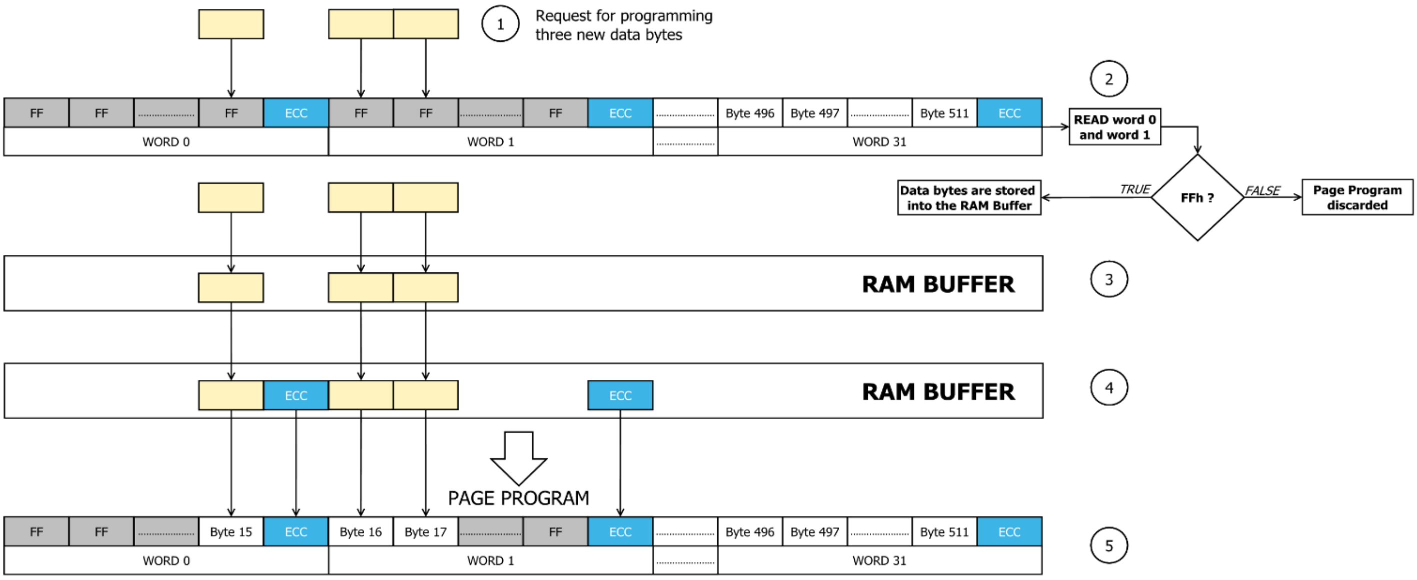
2.3 Page program and page write operations with page architecture

2.3.1 Page program operation

A page program operation allows one to 512 bytes of data initially in the erased state (0xFF) to be programmed. To program any byte inside a word, the whole word must be erased. If a page program operation includes two bytes in two successive words, the two words must be in the erased state. After a page program operation, the new ECC signatures are computed and stored in the associated words of the updated bytes.

The figure below explains how the page program process is managed.

Figure 5. **Page program process of a Page EEPROM with ECC**



The steps of the page program process illustrated in Figure 5 are described below:

1. The user sends a request to program three bytes into the memory array.
2. Word 0 and word 1 where the new data bytes must be programmed are read to check the erase state. If one or more bytes are not erased the operation is discarded.

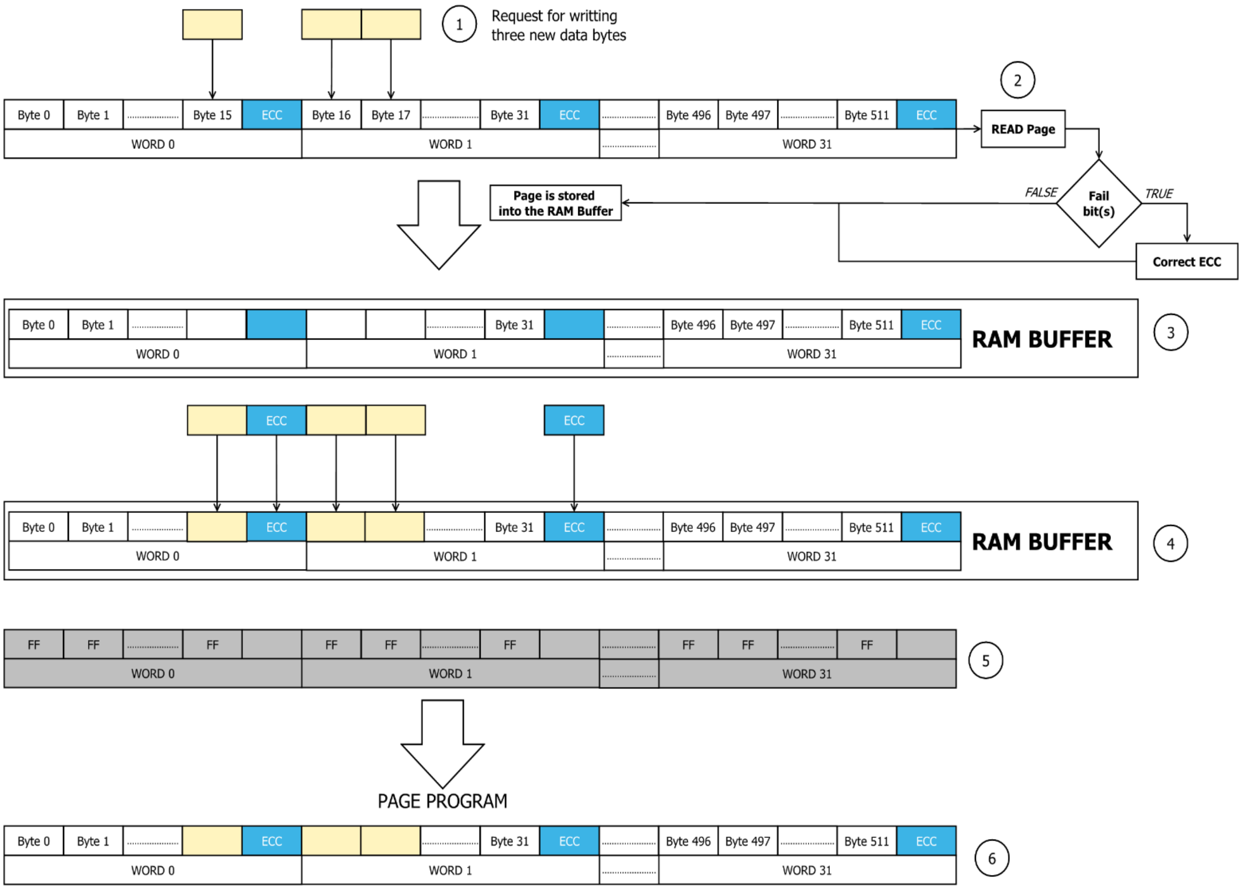
Note: During the erased state check, if one or two data bit are incorrect, the bit is corrected by ECC in the same way as a normal read operation.

3. The new data bytes are loaded in the correct RAM buffer position.
4. The new ECC bits corresponding to word 0 and word 1 are calculated and replace the previous ECC bits in the RAM buffer.
5. The memory words are programmed with the RAM buffer content.

2.3.2 Page write operation

Page write operation allows up to one to 512 bytes of data to be written in a single instruction. The device internally manages the page erase operation followed by the program operation. If one or more bytes are needed to be written, the whole page is erased, then rewritten, and the word ECC signatures of the written bytes are updated. The figure below explains how the page write process is managed.

Figure 6. **Page write process of a Page EEPROM with ECC**



The steps of the page write process illustrated in Figure 6 are described below:

1. The user sends a request to write three bytes into the memory array.
2. The whole page is read, if one or two data bits are incorrect, the corresponding words are corrected.
3. The whole page (512 bytes) except the requested bytes is stored in the RAM buffer.
4. The new data bytes are loaded at the correct position into the RAM buffer and the new ECC bits corresponding to word 0 and word 1 are calculated and also loaded into the RAM buffer.
5. The page is erased (0xFF) with a page erase operation.
6. The page is programmed with the RAM buffer content: requested bytes and their new ECC signatures plus unchanged words and their initial ECC signatures.

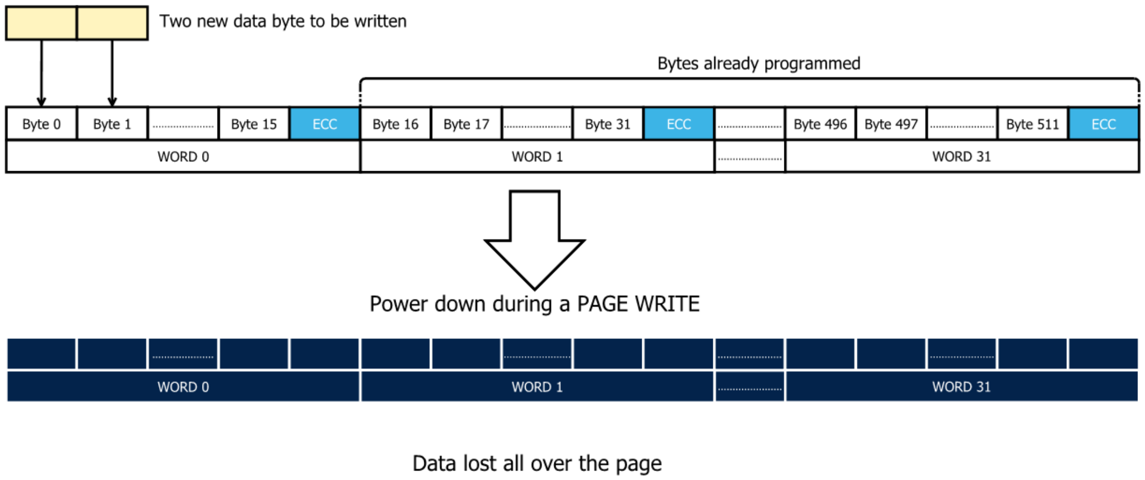
3 Power loss and data corruption

3.1 Power loss with page write operation

In Page EEPROMs, when writing one or more bytes inside a word with the page write operation, the whole page is written once the internal write cycle (t_{PW}) is triggered. If the supply voltage (V_{CC}) drops during the internal write cycle the whole page might be lost.

The figure below shows one example of data corruption in the Page EEPROM with the page write process.

Figure 7. **Data corruption due to power-down during a page write process**

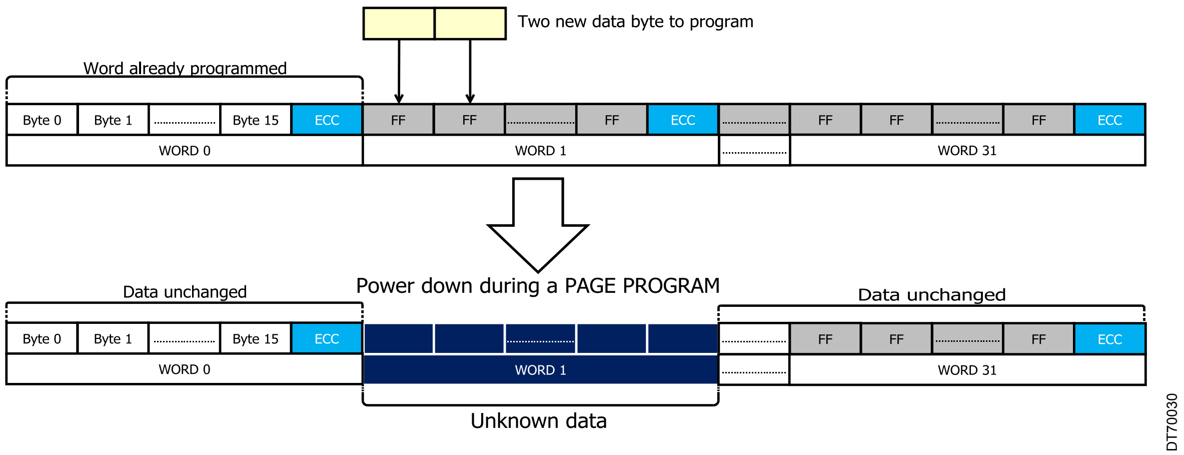


3.2 Power loss with page program operation

If the page program operation is used to program new data bytes, the drop of supply voltage only affects the word(s) associated to the programmed bytes. Page data not concerned by page program operation stay unchanged.

The figure below explains the power-down during a page program process.

Figure 8. **Power-down during a page program process**



4 Read-only parameters

When using the page write operation with a Page EEPROM, care must be taken to the data mapping of read-only bytes and read/write bytes. Read-only data handle key values for the end application. Theses values must never be changed, even if some uncontrolled event occurs (for example, power loss during Page Write operation).

As a power-down during a page write cycle can modify the whole addressed page, a page must not store both read-only bytes and read/write bytes. For a robust application design, define two separate data sets:

- read-only data, which is only read during the application life
- read/write data, in which the bytes can be updated during the application life

Read-only data must be stored in a different page from the page in which the read or write data is stored.
An unexpected voltage supply drop during a write cycle does not corrupt the read-only data. It limits the data corruption to only the addressed page.

5 Conclusion

Page EEPROM is page architecture based

- Pages are the smallest erasable block.
- Each page contains words, and each word is composed of 16 bytes + 17 ECC bits.

Reliability improvement by ECC

ECC is computed and updated during page program, page erase, and page write operation. During read operations, ECC is computed to correct single bit or double bit error and flag triple bit errors.

Power loss during page write operations

If the supply voltage (V_{CC}) drops during the internal page write cycle, data over the whole page can be considered as lost. As a consequence, read-only parameter bytes must not share the same page as the read/ write parameter bytes.

Revision history

Table 2. **Document revision history**

Date	Version	Changes
10-Jun-2022	1	Initial release.
16-Feb-2023	2	Addition of the Applicable products table in the Section Introduction. Minor changes applied to the whole document.
21-Apr-2023	3	Updated the following elements: • Section 2.1 Read operation with word management to update the steps of the read process. • Section 2.2 Erase operations with page architecture to update the list of erase operations. • Section 2.3.1 Page program operation to update the steps of the page program process. • Section 3.1 Power loss with page write operation to add t_{pw}. • Section 3.1 Power loss with page write operation to update the Figure 7. Data corruption due to power-down during a page write process

Contents

List of tables

List of figures