ML-3 Partial Page Programming

Author: Mohammad Nada
Associated Part Family: S34ML0xG3/S35ML0xG3

This application note, AN223238, details the partial page programming used in the design of SkyHigh 1X-nm S34ML0xG3/ S35ML0xG3 and all the applicable rules and restrictions for both ONFI and SPI interfaces

1 Error Correcting Code (ECC)

The NAND flash architecture inherently requires ECC because data can potentially get random bit errors. Depending on the technology node and whether it is single-level cell or multi-level cell (MLC), the number of ECC bits required for correction is different.

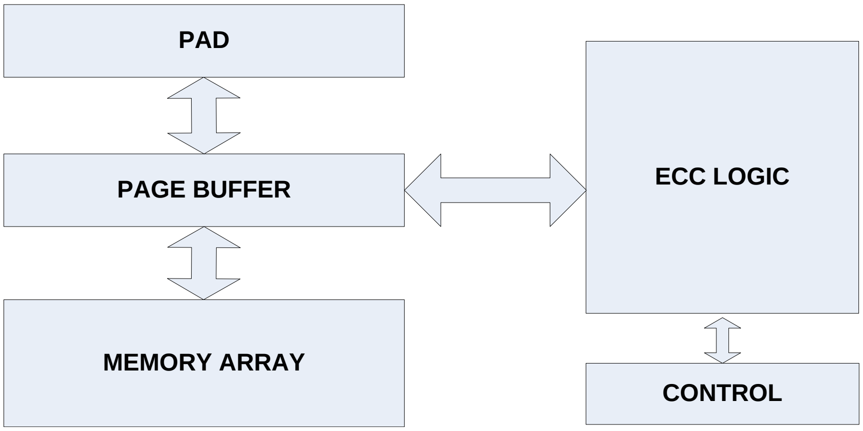
The SkyHigh 1X-nm S34ML0xG3/ S35ML0xG3 SLC NAND Flash memory is designed with a powerful internal ECC engine.

The basic unit of program operation is 32bytes per 1 ECC chunk and Program input shall be made if and only if 32bytes or more input data detected.

2 SkyHigh 1X-nm SLC On-chip ECC Architecture

During a READ operation, the page data is read from the array to the page buffer, where the ECC code is calculated and compared with the value read from the array.

Figure 1. Internal ECC Architecture



3 Partial Page Programming

A Page Program operation sequence enables the host to input 1 byte to 2176 bytes (2048 bytes + 128 bytes) for 4Gb/2Gb devices, or 2112 (2048 + 64 bytes) for 1Gb devices, of data within a page to a cache register, and moves the data from the cache register to the specified block and page address in the array. If more bytes are loaded, then those additional bytes are ignored by the cache register.

The number of consecutive partial page programming operations (NOP) within the same page must not exceed 4. Main data shall be stored in main data region, and then use Random Data Input to change column to the address of user spare for programming of corresponding user spare of input data. Both main data and user spare must be entered and program executed for NOP operation. if only user spare is entered and program executed without main data , then the main data can no longer be program executed separately, doing so will result in data corruption.

During NOP program, data that is designated to user spare area must use random data input to change column address. Because NOP is allowed in maximum of 4 times, user spare area is also divided into 4 regions to match up to each NOP area. Main data is divided using NOP, and randomized while programming the data. Although user spare that is included in each NOP is also randomized, randomization done between main data and user spare are different as shown in Figure 2 for ONFI and Figure 4 for SPI. Therefore, for each NOP,column address change should match up correct NOP with NOP user spare.

Figures 2 and 4 demonstrate programming the page and the spare area by using the Random data input to change the column address between the main area and the user spare area for ONFI and SPI interfaces. First, input 512 bytes of data for NOP0; then, input user spare data for NOP0 after changing of column address using Random DataInput. After programming data, use Random Data Input to return back to starting address of NOP1 main data. Secondly, input 512bytes of NOP1 data, using Random Data Input to change column address to user spare data for NOP1 input. After programming user spare data for NOP1, return to startingaddress of NOP2 main data. NOP2 and NOP3 must follow the same process. Both main data and user spare shall be entered andprogram executed simultaneously for NOP operation. if only the user spare is entered and program executed without main data ,then the main data can no longer be allowed to be program executed separately, doing so will result in data corruption.

Four Partial page programming with main data and user spare area can be programmed independently as shown in Figure 3 for ONFI and Figure 5 for SPI. The main requirement is that the user area and the spare area must be programmed simultaneously for each of the NOP operation.

If user decide to follow the same order as programming order during data out, read out NOP0 area first, then use Random Data Output to change column address to read out NOP0 user spare area.

3.1 Partial Page Programming - ONFI

Figure 2. Random Data Input Timings - ONFI

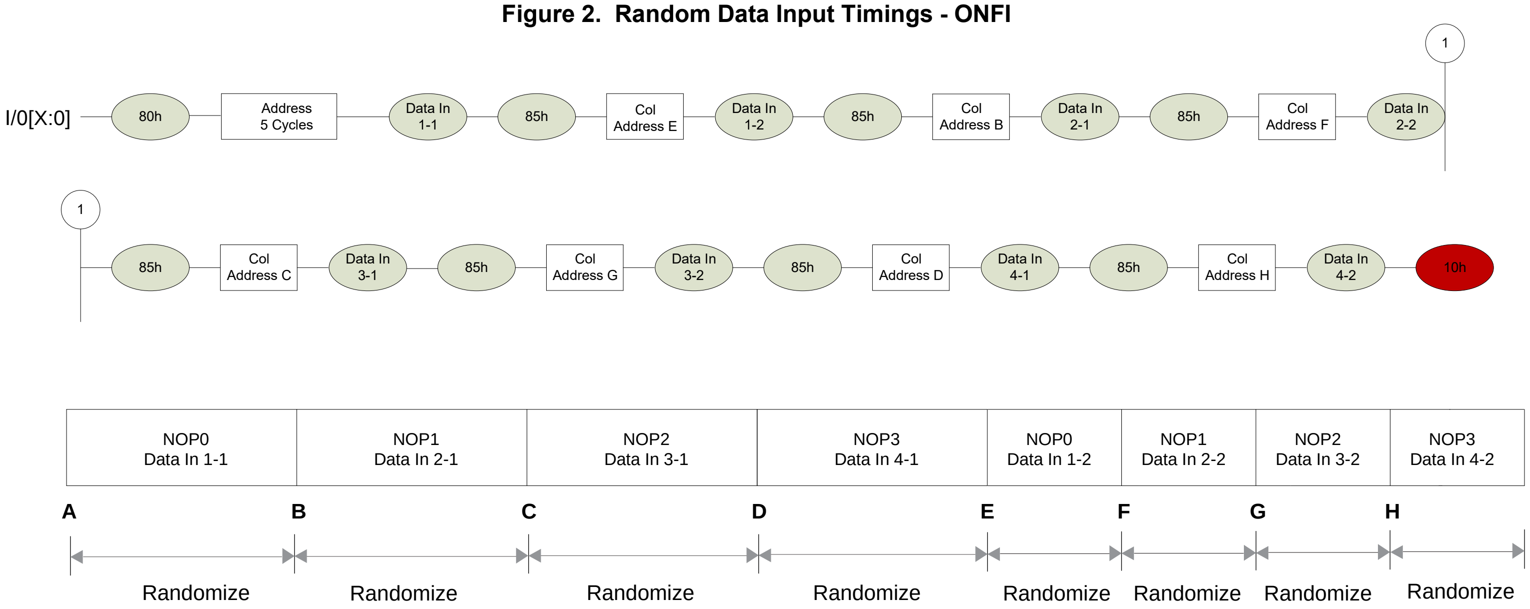
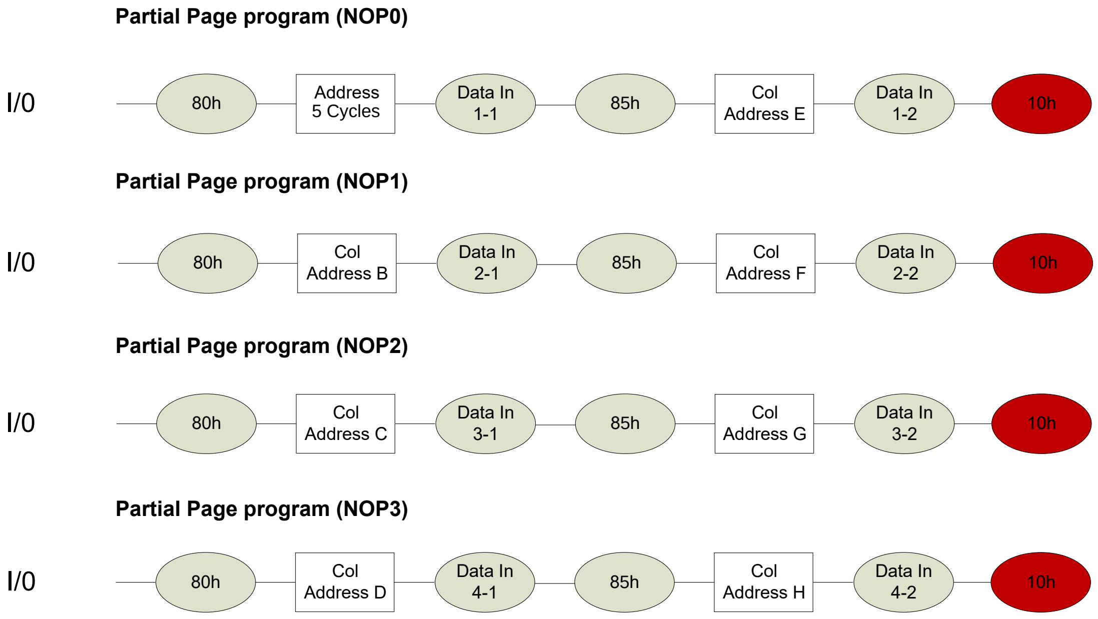


Figure 3. FourPartialPageProgramming - ONFI



3.2 Partial Page Programming - SPI

Figure 4. Random Data Input Timings - SPI (1X)

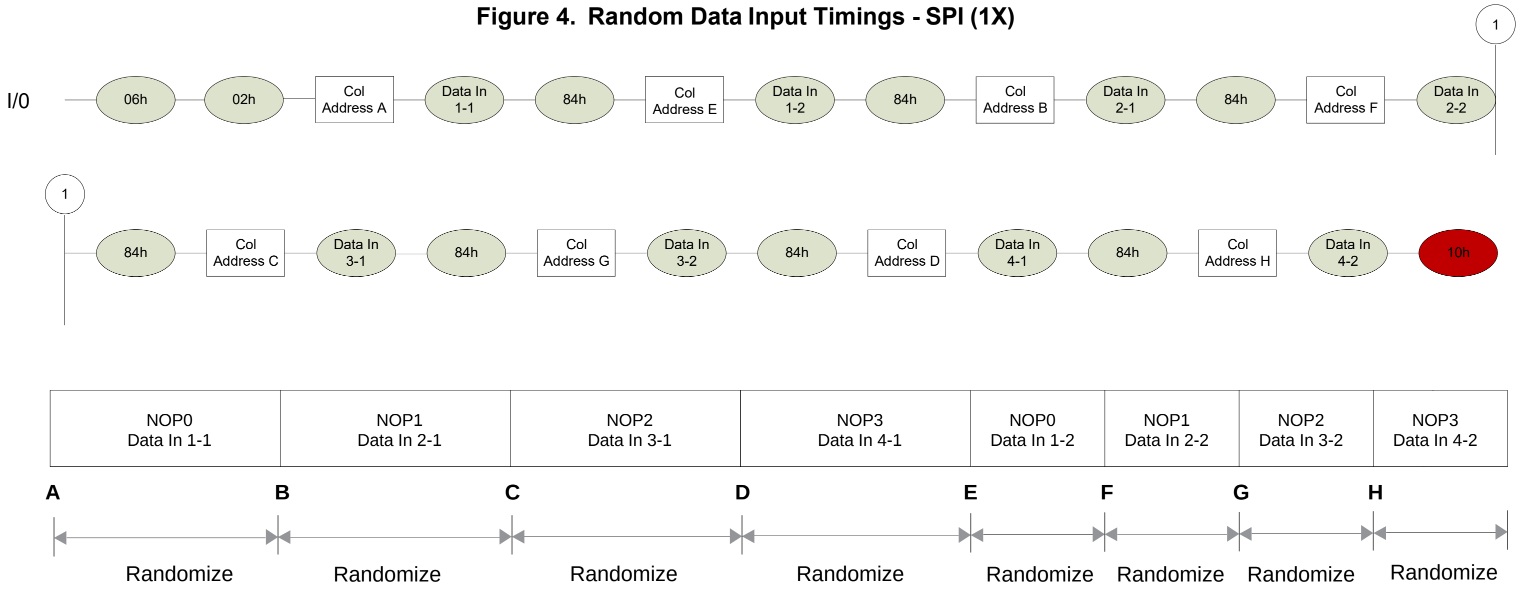
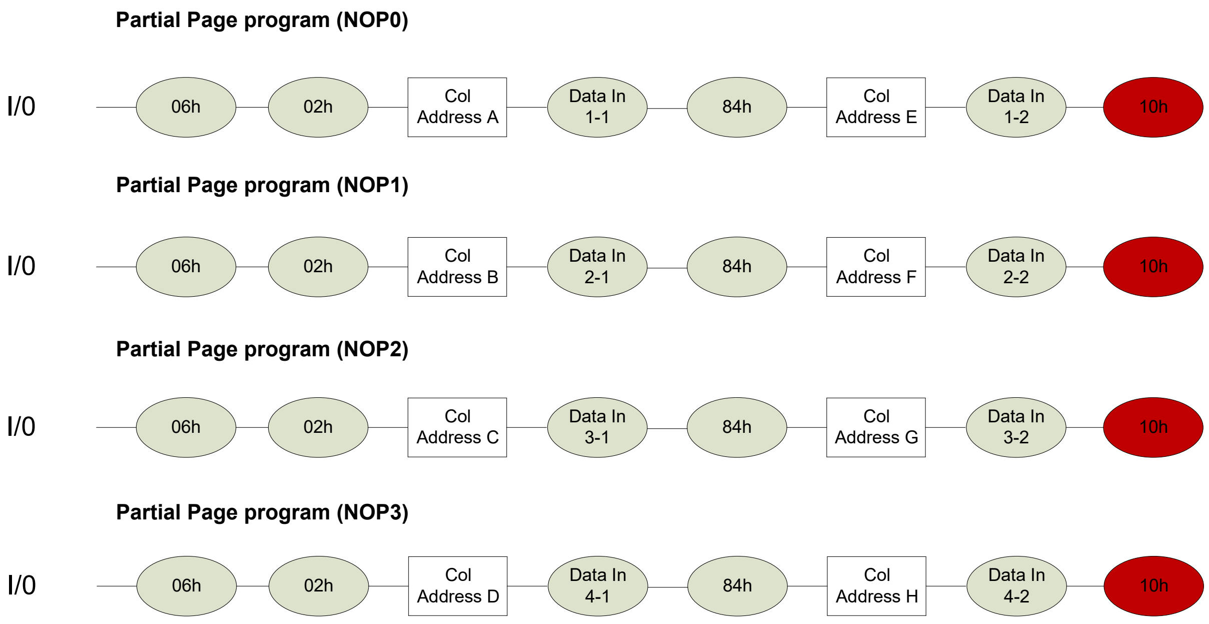


Figure 5. FourPartialPageProgramming - SPI (1X)



Partial Page program (NOP0)

Partial Page program (NOP1)

Partial Page program (NOP2)

Partial Page program (NOP3)

5 References

- ONFI- S34ML04G3 4 Gb, 3 V, 2K Page Size, x8 I/O SLC NAND Flash Memory for Embedded, Datasheet, Specification Number 002-19204
- ONFI- S34ML01/02G3 1/2 Gb, 3 V, 2K Page Size, x8 I/O SLC NAND Flash Memory for Embedded, Datasheet, Specification Number 002-19206
- SPI- S35ML01/02/04 G3 1/2 Gb, 3 V, 2K Page Size, x8 I/O SLC NAND Flash Memory for Embedded, Datasheet, Specification Number 002-19205

6 Document History

Document Title: AN223238 - ML-3 Partial Page Programming

Document Number: 002-23238

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**		MNAD	07/25/2022	New Application Note