Macronix MX25L_G Serial Flash Programming Guide

1. Introduction

This application note is a programming guide for the Macronix MX25L_G Serial NOR Flash family with its focus on the Page Program function. It describes how to optimize the MX25L_G Page Program function to maximize data reliability. The information in this document is based on datasheets listed in Section 6. Newer datasheet versions may override the contents of this document.

2. Page Program Introduction

The MX25L_G family provides a page buffer for programming up to 256-bytes of data.
Most applications contain system code (boot code, file system, O.S., and so on) which take multiple sectors to store. Typically, systems use the full page program size to gain performance for code storage. However, metadata (parameter settings, system configuration, user data, and so on) which is variable in size (ranging from a few bytes to hundreds of bytes) frequently uses partial page programming or byte programming.
We will discuss how to best to program with these types of data in the following sections.

3. System Code Programming Guide

Most system code sizes are large and take many sectors/blocks to store. When programming large blocks of code, it is recommended to use the full page size to program the flash. This will improve overall write throughput. To show the performance benefit of using a full page size, we compare the time it takes to program 48MB of system code using 256-byte page programming and byte programming. Table 3-1 shows that programming with full 256-byte pages significantly reduces code programming time and will in turn reduce mass production line programming costs.

Table 3-1: System Code Programming Time (typ.)[*1] vs. Page Program Size

System Code Size	Page Program Size	Macronix MX25L_G
48MB (384Mb)	1 Byte	604s
	256 Bytes	**118s**

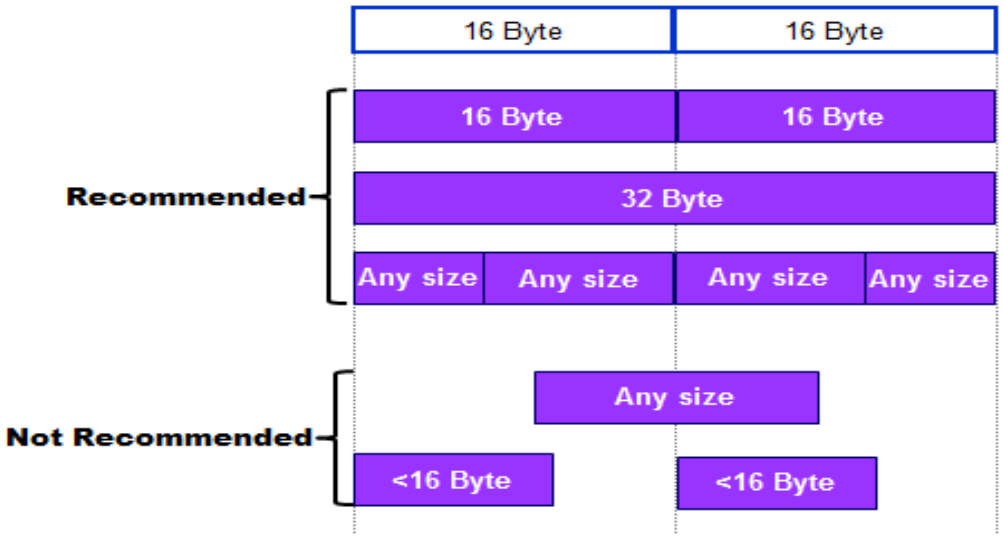
Note: 1. The timing does not include command overhead time.

4. Metadata Programming Guide

Metadata varies in length depending on value and type definition. Metadata storage is sector based (a sector is the smallest erase size) and will not generally require multiple sector programming. In order to achieve enhanced data reliability, address alignment should be used to program variable length data (such as Metadata) into the flash. Strictly enforcing a 16-byte length boundary for variable length Metadata is recommended. For simplicity, the boundary concept is the easiest way to calculate the address. For data shorter than 16 bytes, a partial program algorithm can be used to pad the remainder of the buffer with null data (FFh) within a 16-byte boundary. For

address alignment, programming using an address ordering algorithm can be easily implemented in most applications, so that the system can manage flash space, since flash sector size is typically much larger than the Metadata size. This will ensure that a minimum 16-byte page length is always used and that resulting address alignment will always be on 16-byte boundaries (Figure 4-1).

Figure 4-1: Apply a Boundary Condition to achieve Maximum Data Integrity



The system then programs the Metadata into the allocated address space. Table 4-2 shows this concept using a 16-byte alignment as an example.

Table 4-2: Metadata Programming Example

Original Non-Aligned Structure

Data Order	0	1	2	3
Data Size	5B	256B	4B	6B
Address Offset	0	5	260	264

New Preferred Structure

Data Order	0	0	1	2	3	3
Data Size	5B	11B	256B	4B	6B	6B
Address Offset	0	5	16	272	276	282

Original Data: 5B
Add 11B null data (FFh)
for 16B alignment.

Data: 256B
1 time 256B page
program for alignment.

Original Data: 4B, 6B
Add 6B null data (FFh)
for 16B alignment.

5. Summary

The MX25L_G supports Byte Program and Page Program functions and is backward compatible with the existing MX25L series. However, in order maintain the same data reliability as the existing MX25L (non G rev) series, it is strongly recommended that data be written to the MX25L_G using full 256-byte pages. Writing individual bytes or partial page programs (non-multiples of 16-bytes) is accepted, but allocating 16-byte length boundaries for address alignment is recommended for maximum data reliability.

6. Reference

Table 6-1 shows the datasheet versions used for comparison in this application note. For the most current, detailed Macronix specification, please refer to the Macronix Website at http://www.macronix.com

Table 6-1: Datasheet Version

Datasheet	Location	Date Issue	Revision
MX25L51245G	Website	Jan. 06, 2014	Rev. 0.01
MX66L1G45G	Website	Dec. 09, 2013	Rev. 0.00

Note: Macronix data sheet is subject to change without notice.

7. Revision History

Revision	Description	Date
1.0	Initial Release	May. 30, 2014

Macronix MX25L_G Serial Flash Programming Guide

Except for customized products which have been expressly identified in the applicable agreement, Macronix's products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and not for use in any applications which may, directly or indirectly, cause death, personal injury, or severe property damages. In the event Macronix products are used in contradicted to their target usage above, the buyer shall take any and all actions to ensure said Macronix's product qualified for its actual use in accordance with the applicable laws and regulations; and Macronix as well as it's suppliers and/or distributors shall be released from any and all liability arisen therefrom.

Copyright© Macronix International Co., Ltd. 2014. All rights reserved, including the trademarks and tradename thereof, such as Macronix, MXIC, MXIC Logo, MX Logo, Integrated Solutions Provider, NBit, Nbit, NBiit, Macronix NBit, eLiteFlash, HybridNVM, HybridFlash, XtraROM, Phines, KH Logo, BE-SONOS, KSMC, Kingtech, MXSMIO, Macronix vEE, Macronix MAP, Rich Au-dio, Rich Book, Rich TV, and FitCAM. The names and brands of third party referred thereto (if any) are for identification purposes only.

For the contact and order information, please visit Macronix's Web site at: http://www.macronix.com