

MirrorBit Write Buffer Programming and Page Buffers



Application Note

Overview

The write buffer in Spansion MirrorBit Flash memory devices is designed to reduce the overall system programming time when writing to the Flash device. The host system first fills the write buffer with data, and then issues the write buffer programming command. The flash device programs the buffered data in parallel, and thus reduces the programming time compared to programming bytes or words one at a time. Depending on the device design, a write buffer can store up to 32 words (64 bytes) of data that can then be written using a single programming operation.

MirrorBit devices use a specific, additional set of commands to accomplish write buffer programming. Refer to the appropriate Spansion data sheet for the complete list of device commands.

The page buffer read function accelerates read operations for addresses within a specific range.

Write Buffers

Benefits of Using the Write Buffers

The main benefit is pure programming speed. Data can be written into the write buffers at the same rate that data can be read from the part (a 90 ns part can have new data written into the buffer every 90 ns). In addition, the write buffers program the Flash memory array in parallel (four words at a time), greatly decreasing the time needed to write to the Flash.

Figure 1 shows that using the write buffers is much more efficient than standard byte or word programming.

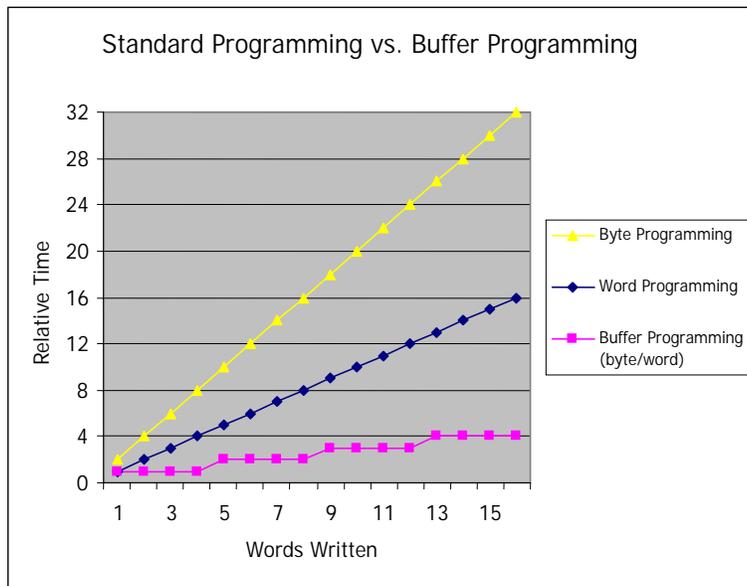


Figure I. Standard Programming vs. Buffer Programming

Write Buffer Operation

Write buffer programming is only available through the *Write to Buffer* and *Program Buffer to Flash* command sequences. The *Write-to-Buffer Abort Reset* command sequence is used to exit out of the Write-Buffer-Abort state. Table 1 lists all software program sequences associated with the write buffer.

Table 1. Write Buffer Operation

Command Sequence	Interface	Bus Cycles											
		First		Second		Third		Fourth		Fifth		Sixth	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Write to Buffer	Word	555	AA	2AA	55	SA	25	SA	WC (Note)	PA	PD	WBL (Note)	PD
	Byte	AAA		555									
Program Buffer to Flash (Confirm)	Both	SA	29										
Write-to-BufferAbort Reset	Word	555	AA	2AA	55	XXX	FO						
	Byte	AAA		555									

Note: The sixth cycle must be repeated to complete the number of buffer writes specified by the WC in cycle four.

Legend

PA = Program Address of the memory location to be programmed. This can be any address within the target write buffer page.

PD = Program Data to be programmed at location PA.

SA = Sector Address containing locations to be programmed. This can be any valid address within the sector.

WC = Write Count is the number of write buffer locations to load minus one (to load 7 locations the WC would be 6).

WBL = Write Buffer Location. The address must be within the same write buffer page (32-byte range located on a 32-byte boundary) as PA.

In Table 1, the host system first loads data at any location in the target write buffer page. Subsequent write buffer locations do not need to be loaded in any particular order as long as they reside in the same write buffer page and sector (the S29WS-P family is the only exception to this rule; it requires that addresses be loaded sequentially). A write buffer page, for a device with a 32-word write buffer, is defined as the address from xxx0h to xx1Fh (any sequence of address where $A_{MAX} - A_5$ do not change).

Note that the internal write counter decrements for every data load operation, not for each unique write buffer address location. If the same write buffer location is loaded multiple times, the internal write counter will decrement after each load operation. The last data loaded into a given write-buffer location will be programmed into the device after the Program Buffer to Flash command. The host system must therefore account for the effects of loading a write-buffer location more than once.

When the Write to Buffer command programming sequence has been completed, the Program Buffer to Flash command must be issued to move the data from the write-buffer into the flash memory array.

Write Buffer Programming Abort

As stated earlier, write buffer programming cannot be performed across multiple write buffer pages or across multiple sectors. If this is attempted, the write buffer programming operation will be automatically aborted. The abort condition is detected by performing a status read operation, in which the data shows DQ1 = 1, DQ7 = DATA# (for the "Last Loaded Address"), DQ6 = TOGGLE, DQ5=0.

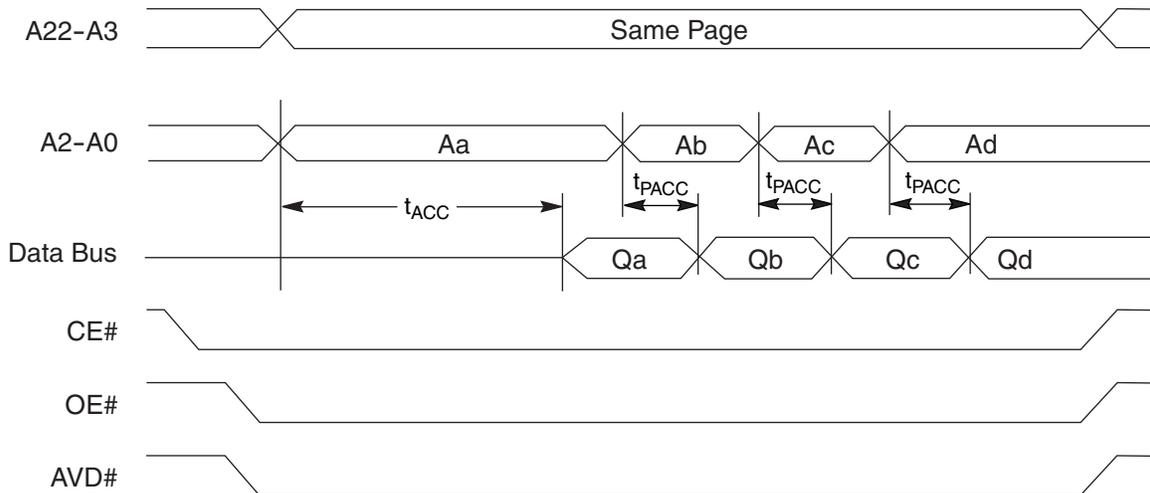
The write buffer programming sequence aborts in the following cases:

- Loading a value that is greater than the write buffer size (write-buffer-page) during the “fourth cycle (WC) Numbers of Locations to Program” step.
- Writing to an address in a sector that is different than the one specified during the Write-Buffer-Load command.
- Writing an Address/Data pair to a different write buffer page than the one selected by the starting address during the “write buffer data loading” stage of the operation.
- Writing data other than the Program Buffer to Flash command after loading the specified number of write buffer locations.

Note that the *Write-to-Buffer Abort Reset* command sequence must be written to the device to return to READ mode.

Page Buffer Read Introduction

Whenever the host system changes a “page address” (or toggles CE# during a read), the device performs a “random access”. During this “random access” the read page buffer is loaded in parallel with data within the selected read-page boundaries. Subsequent intra-page accesses are 3 to 4 times faster than random accesses because the data are already available in the buffer. Therefore, read performance is significantly improved.



Note: Figure shows device in word mode. Addresses A1–A-1 are used during byte mode.

Figure 2. Page Buffer Read Timing Diagram

Read Buffer Operation with a 4-word Page Buffer

Most Spansion page mode flash devices use a 4-word (8-byte) page buffer. For page buffer read operation, the user must issue a read address, or “RA”, for any memory location. During the initial access time (t_{CE}/t_{ACC}) a page of 4-words (8-bytes), starting from an 8-byte boundary, is read into the page buffer. If the device is in word mode, address bits A1 and A0 can then be used to access any of the four words within the page with a reduced page access time (t_{PACC}). If the device is in byte mode in a x8/x16 device, A1 through A-1 can be used to access any of the eight bytes in the page. If the device is a x8-only device, A2 through A0 can be used to access any of the eight bytes within the page.

The appropriate page is selected by the higher address bits: $A_{(MAX)}-A_2$ for x16-only and x8/x16 devices, and $A_{(MAX)}-A_3$ for x8-only devices. Fast page mode accesses are obtained by keeping the high-order “read page address” bits constant and changing the “intra-read page” address bits: A0 to A1 for x16-only and x8/x16 in word mode; A-1 to A1 for x8/x16 in byte mode; and

A0 to A2 for x8-only. This is an asynchronous operation with the host system supplying the specific byte or word location.

A depiction of the command sequence definition for read accesses is shown in [Table 2](#).

Table 2. Read Access

Command Sequence	Interface	Bus Cycles									
		First		Second		Third		Fourth		Fifth	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read	Both	RA	RD	RA	RD	RA	RD	RA	RD	Note	Note

Note: For reading bytes, eight consecutive memory locations can be read, compared to four memory locations for reading words. "Intra-read page" locations can be accessed in any order.

Legend

RA = Read Address

RD = Read Data

A depiction of the device bus operation for read accesses is shown in [Table 3](#).

Table 3. Device Bus Operation for Read Access

Operation	CE#	OE#	WE#	RESET#	WP#/ACC	Address	Data
Read	L	L	H	H	X	A _{IN}	D _{OUT}

During page buffer read operations, the CE# pin must be kept at voltage level V_{IL} during all fast page mode accesses. If the CE# pin toggles or changes state during a page buffer read operation, the current data transfer will automatically be aborted and another initial page access is started. This will result in an unnecessary delay in read timings.

Read Buffer Operation with an 8-word Page Buffer

The Spansion S29GL-N family of devices uses an 8-word (16-byte) page buffer. The feature works the same way as devices with a 4-word page buffer, except that page boundaries on these devices are on 16-byte increments. If the device is in word mode address bits A2–A0 can be used to access any of the eight words within the page with a reduced page access time (t_{PACC}). Hardware and software that was designed to work with 4-word page buffers will work just as well on devices with an 8-word page buffer without any changes.

Write Buffer Programming in S29PL-N Devices

In Spansion S29PL-N devices, write buffer programming with program and erase suspend/resume functionality allows the system to write to a maximum of 32 words in one programming operation. This provides increased programming performance up to four times faster than word programming. [Table 4](#) lists the programming steps. Please refer to the appropriate Spansion data sheet for a review of the full write buffer operation.

Table 4. Write Buffer Programming Procedure in PL-N Devices

Cycle#	Description	Address	Data	Comment
1	Unlock	Base + 555h	00AAh	
2	Unlock	Base + 2AAh	0055h	
3	Write Buffer Load Command	Program Address	0025h	A _{MAX} to A ₁₅ is latched in this command cycle and must not change in subsequent cycles
4	Write Word Count	Program Address	Word Count (N-1)h	
5	Load Buffer Word N	Program Address, Word N	Word N	A _{MAX} to A ₅ is latched in this command cycle and must not change in subsequent command cycles

Note: For maximum efficiency, it is recommended that the write buffer be loaded with the highest number of words (N words) possible.

The following is an example of write buffer programming Command:

```

/* Example: Write Buffer Programming Command */
/* NOTES: Write buffer programming limited to 16 words. */
/* All addresses to be written to the flash in */
/* one operation must be within the same flash */
/* page. A flash page begins at addresses */
/* evenly divisible by 0x20. */
UINT16 *src = source_of_data; /* address of source data */
volatile UINT16 *dst = destination_of_data; /* flash destination address */
int wc = words_to_program-1; /* word count (minus 1) */
*( (volatile UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (volatile UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (volatile UINT16 *)sector_address ) = 0x0025; /* write write buffer load command */
*( (volatile UINT16 *)sector_address ) = wc; /* write word count (minus 1) */
while (wc >= 0) /* ALL dst MUST BE SAME PAGE */
{
    *dst = *src; /* write source data to destination */
    dst++; /* increment destination pointer */
    src++; /* increment source pointer */
    wc--; /* decrement word count */
}
*( (volatile UINT16 *)sector_address ) = 0x0029; /* write confirm command */

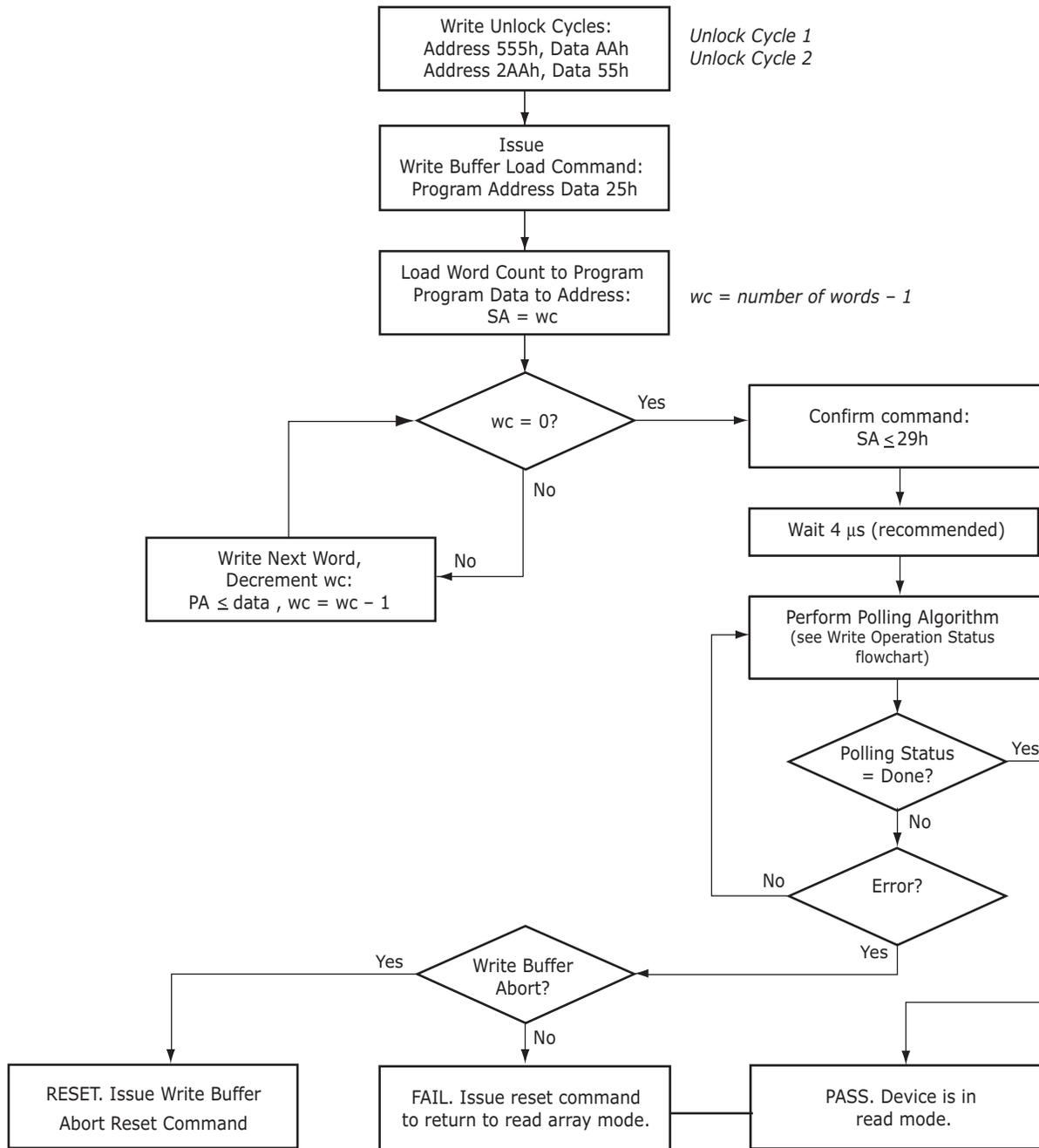
/* insert code to poll for completion */

/* Example: Write Buffer Abort Reset */
*( (volatile UINT16 *)addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (volatile UINT16 *)addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (volatile UINT16 *)addr + 0x555 ) = 0x00F0; /* write buffer abort reset */

```

The write buffer page is selected by using the addresses A_{max} - A₅. The write buffer page addresses must be the same for all address/data pairs loaded into the write buffer. Write buffer programming cannot be performed across multiple write buffer pages and sectors. If a write buffer address is loaded multiple times, the address/data pair counter decrements for every data load operation. The software takes care of the ramifications of loading a write buffer location more than once.

Write buffer programming operation can be suspended using the suspend/resume command. If the write buffer command sequence is entered incorrectly the device enters write buffer abort.



Conclusion

The write buffer programming feature of MirrorBit Flash memory devices can decrease the programming time by over 75%, when compared to single word programming. Write buffer programming is enabled via a simple addition of three commands to the standard embedded algorithm bus command set.

The read page buffer feature of MirrorBit Flash memories can increase performance significantly. Following each random (inter-page) access all locations of the referenced 4- or 8-word page are available for fast access. When read accesses can be grouped within a page the average read performance can be increased by 3 to 4 times.

Revision History

Revision I (May 2, 2006)

Initial Release.

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion LLC. Spansion LLC reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion LLC assumes no liability for any damages of any kind arising out of the use of the information in this document.

Copyright © 2006 Spansion LLC.

All rights reserved. Spansion, the Spansion logo, MirrorBit, combinations thereof, and ExpressFlash are trademarks of Spansion LLC. Other company and product names used in this publication are for identification purposes only and may be trademarks of their respective companies.