# Technical Note

## Using the Block Lock Feature on MT29F NAND Flash Memory Devices

## Introduction

This technical note describes the block lock feature on Micron's MT29F NAND Flash memory devices.

The block lock feature protects the entire device or ranges of blocks from being programmed and erased. Using this feature is preferable to using the write protect signal (WP#) to prevent PROGRAM and ERASE operations.

For complete product information, see the component data sheet.

# Block Lock Feature

Block lock is enabled and disabled at power-on through the LOCK pin. At power-on, if LOCK is LOW, all BLOCK LOCK commands are disabled. However if LOCK is HIGH at power-on, the BLOCK LOCK commands are enabled and, by default, all the blocks on the device are protected, or locked, from PROGRAM and ERASE operations, even if WP# is HIGH.

Before the contents of the device can be modified, the device must first be unlocked. Either a range of blocks or the entire device may be unlocked. PROGRAM and ERASE operations complete successfully only in the block ranges that have been unlocked. Blocks, once unlocked, can be locked again to protect them from further PROGRAM and ERASE operations.

Blocks that are locked can be protected further, or locked tight. When locked tight, the device's blocks can no longer be locked or unlocked.

## WP# and Block Lock

The following is true when the block lock feature is enabled:

- Holding WP# LOW locks all blocks, provided the blocks are not locked tight.
- If WP# is held LOW to lock blocks, then returned to HIGH, a new UNLOCK command must be issued to unlock blocks.

## UNLOCK (23h-24h)

By default at power-on, if LOCK is HIGH, all the blocks are locked and protected from PROGRAM and ERASE operations. The UNLOCK (23h) command is used to unlock a range of blocks. Unlocked blocks have no protection and can be programmed or erased.

The UNLOCK command uses two registers, a lower boundary block address register and an upper boundary block address register, and the invert area bit to determine what range of blocks are unlocked. When the invert area bit = 0, the range of blocks within the lower and upper boundary address registers are unlocked. When the invert area bit = 1, the range of blocks outside the boundaries of the lower and upper boundary address registers are unlocked. The lower boundary block address must be less than the upper boundary block address. The figures below show examples of how the lower and upper boundary address registers work with the invert area bit.

To unlock a range of blocks, issue the UNLOCK (23h) command followed by the appropriate address cycles that indicate the lower boundary block address. Then issue the 24h command followed by the appropriate address cycles that indicate the upper boundary block address. The least-significant page address bit, PA0, should be set to 1 if setting the invert area bit; otherwise, it should be 0. The other page address bits should be 0.

Only one range of blocks can be specified in the lower and upper boundary block address registers. If after unlocking a range of blocks the UNLOCK command is again issued, the new block address range determines which blocks are unlocked. The previous unlocked block address range is not retained.

**Figure 1: Flash Array Protected: Invert Area Bit = 0**

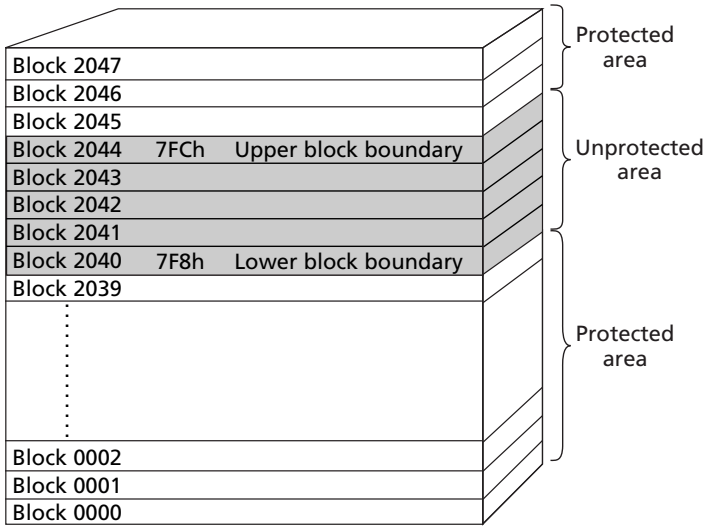



**Figure 2: Flash Array Protected: Invert Area Bit = 1**

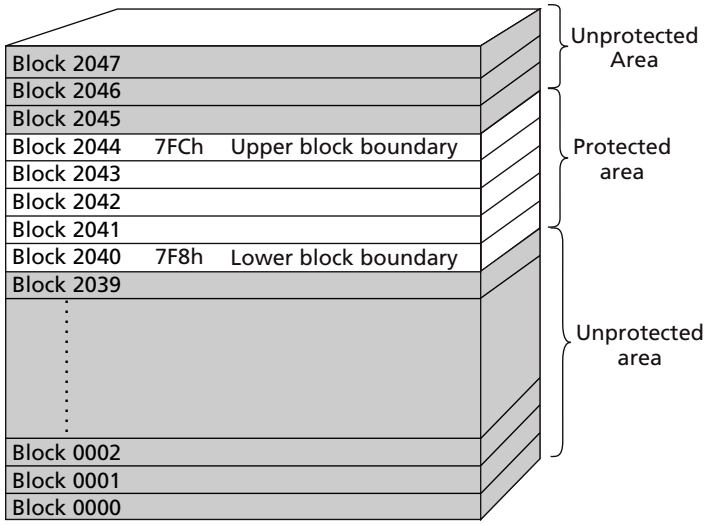
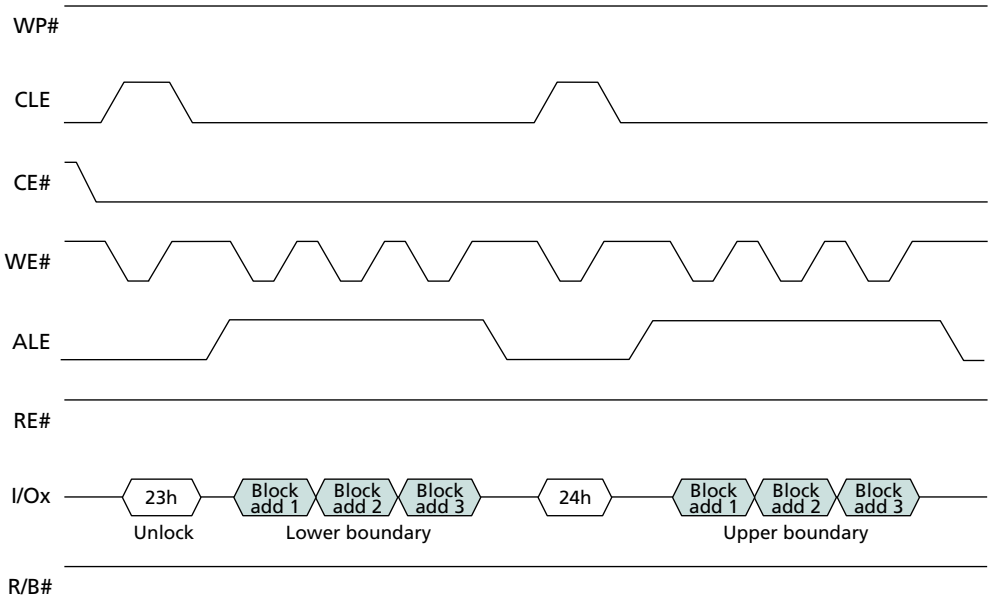


**Table 1: Block Lock Address Cycle Assignments**

| ALE Cycle[1] | I/O[15:8][2] | I/O7 | I/O6 | I/O5 | I/O4 | I/O3 | I/O2 | I/O1 | I/O0 |
|---|---|---|---|---|---|---|---|---|---|
| First | LOW | BA7 | BA6 | LOW | LOW | LOW | LOW | LOW | Invert area bit[3] |
| Second | LOW | BA15 | BA14 | BA13 | BA12 | BA11 | BA10 | BA9 | BA8 |
| Third | LOW | LOW | LOW | LOW | LOW | LOW | LOW | BA17 | BA16 |

Notes:
1. ALE = ADDRESS LATCH ENABLE.
2. I/O[15:8] is applicable only for x16 devices.
3. Invert area bit is applicable for 24h command; it may be LOW or HIGH for 23h command.

**Figure 3: UNLOCK Operation**



## LOCK (2Ah)

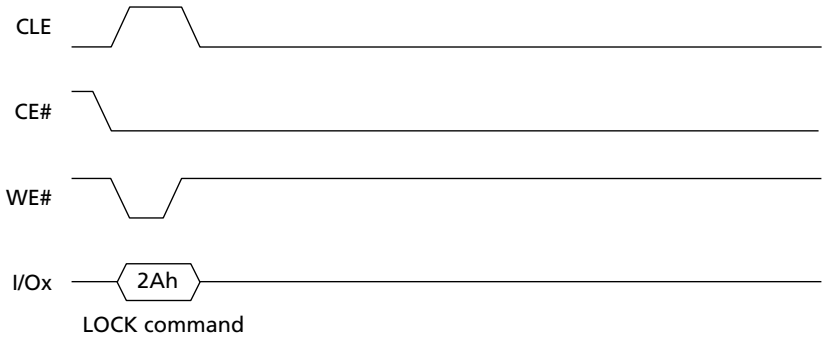
By default at power-on, if LOCK is HIGH, all the blocks are locked and protected from PROGRAM and ERASE operations. If portions of the device are unlocked using the UNLOCK (23h) command, they can be locked again using the LOCK (2Ah) command. The LOCK command locks all of the blocks in the device. Locked blocks are write-protected from PROGRAM and ERASE operations.

To lock all of the blocks in the device, issue the LOCK (2Ah) command.

When a PROGRAM or ERASE operation is issued to a locked block, R/B# goes LOW for $^{t}$LBSY. The PROGRAM or ERASE operation does not complete. Any READ STATUS command reports bit 7 as 0, indicating that the block is protected.

The LOCK (2Ah) command is disabled if LOCK is LOW at power-on or if the device is locked tight.

**Figure 4: LOCK Operation**



## LOCK TIGHT (2Ch)

The LOCK TIGHT (2Ch) command prevents locked blocks from being unlocked and also prevents unlocked blocks from being locked. When this command is issued, the UNLOCK (23h) and LOCK (2Ah) commands are disabled. This provides an additional level of protection against inadvertent PROGRAM and ERASE operations to locked blocks.

To implement LOCK TIGHT in all of the locked blocks in the device, verify that WP# is HIGH and then issue the LOCK TIGHT (2Ch) command.

When a PROGRAM or ERASE operation is issued to a locked block that has also been locked tight, R/B# goes LOW for $^{t}$LBSY. The PROGRAM or ERASE operation does not complete. The READ STATUS (70h) command reports bit 7 as 0, indicating that the block is protected. PROGRAM and ERASE operations complete successfully to blocks that were not locked at the time the LOCK TIGHT command was issued.

After the LOCK TIGHT command is issued, the command cannot be disabled via a software command. When the lock tight status is disabled, all of the blocks become locked, the same as if the LOCK (2Ah) command had been issued.

The LOCK TIGHT (2Ch) command is disabled if LOCK is LOW at power-on.

**Figure 5: LOCK TIGHT Operation**

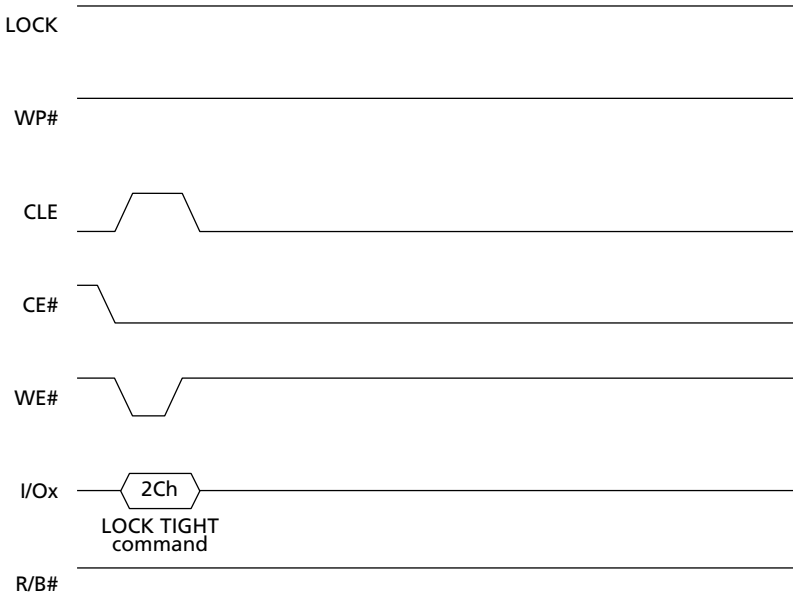
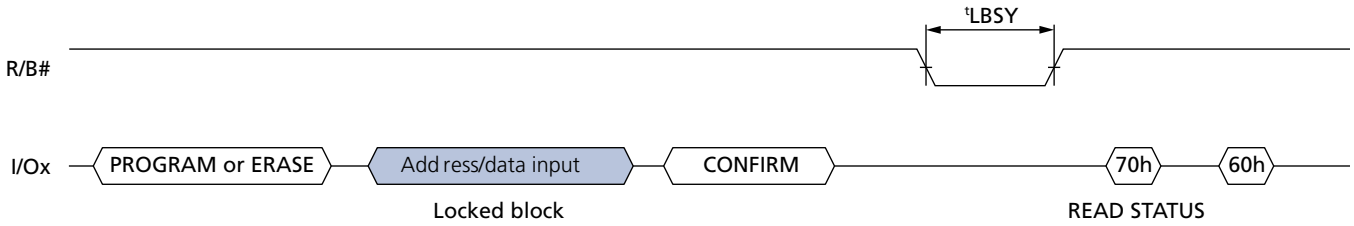


**Figure 6: PROGRAM/ERASE Command Issued to Locked Block**




## BLOCK LOCK READ STATUS (7Ah)

The BLOCK LOCK READ STATUS (7Ah) command is used to determine the protection status of individual blocks. The address cycles have the same format, as shown below, and the invert area bit should be set LOW. On the falling edge of RE# the I/O pins output the block lock status register, which contains the information on the protection status of the block.

**Table 2: Block Lock Status Register Bit Definitions**

| Block Lock Status Register Definitions | I/O[7:3] | I/O2 (Lock#) | I/O1 (LT#) | I/O0 (LT) |
|---|---|---|---|---|
| Block is locked tight | X | 0 | 0 | 1 |
| Block is locked | X | 0 | 1 | 0 |
| Block is unlocked, and device is locked tight | X | 1 | 0 | 1 |
| Block is unlocked, and device is not locked tight | X | 1 | 1 | 0 |

**Figure 7: BLOCK LOCK READ STATUS**

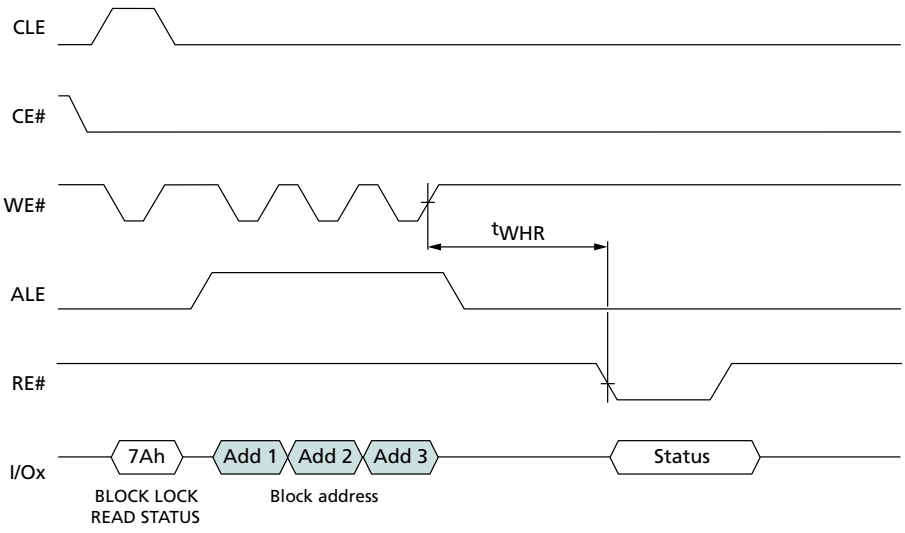
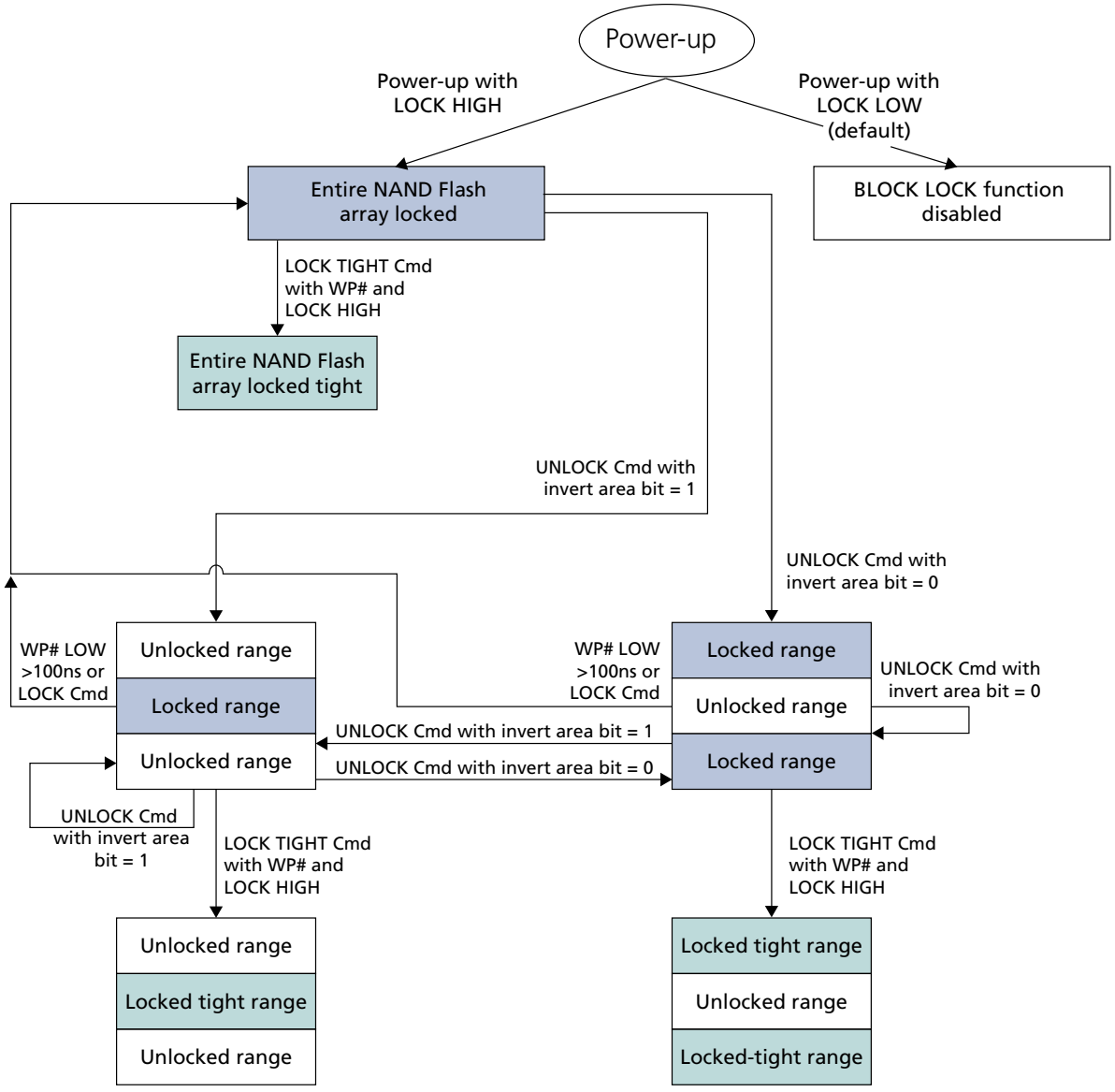


**Figure 8: Block Lock Flowchart**



# Revision History

## Rev. A – 2/2014

- Initial release

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
www.micron.com/productsupport Customer Comment Line: 800-932-4992