# Micron Serial NOR Flash Memory

## 3V, Multiple I/O, 4KB, 32KB, 64KB Sector Erase

## MT25QL01GBBB

## Features

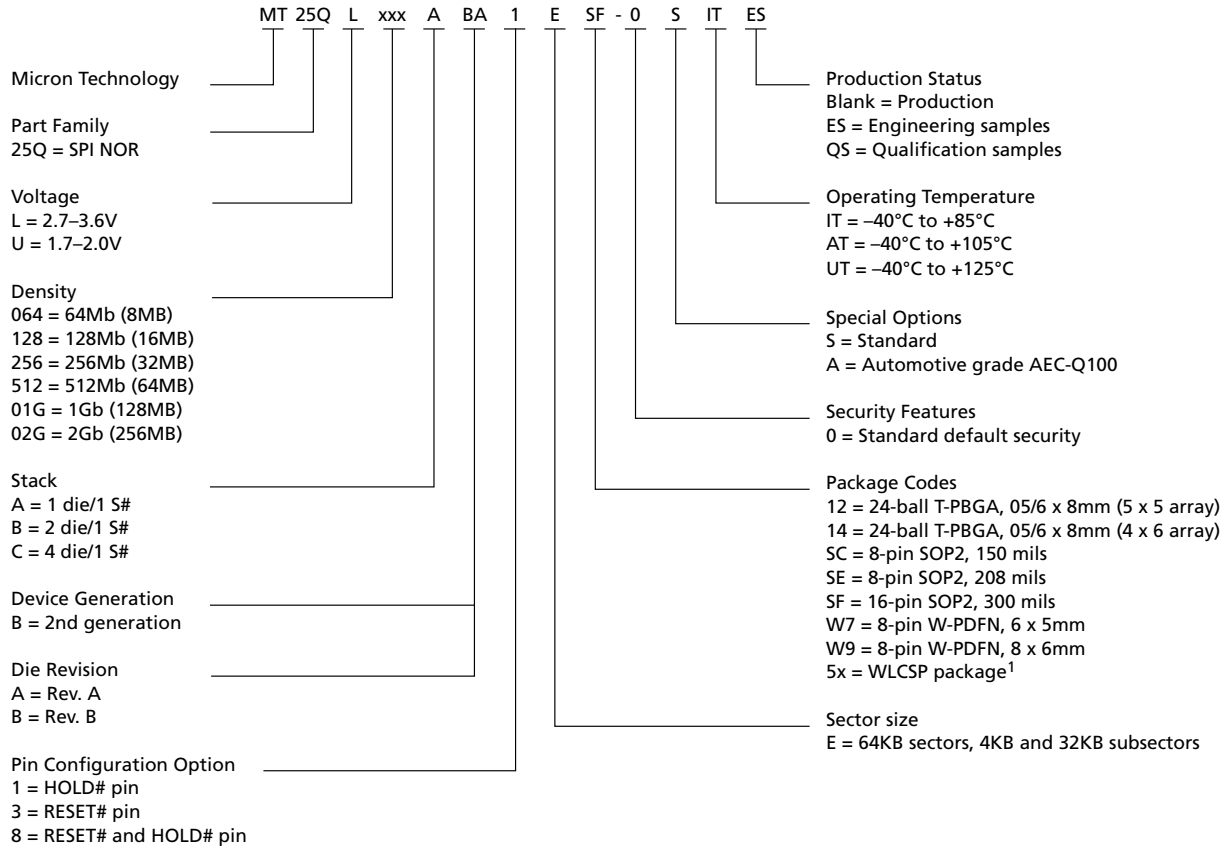
- Stacked device (two 512Mb die)
- SPI-compatible serial bus interface
- Single and double transfer rate (STR/DTR)
- Clock frequency
  - 133 MHz (MAX) for all protocols in STR
  - 90 MHz (MAX) for all protocols in DTR
- Dual/quad I/O commands for increased throughput up to 90 MB/s
- Supported protocols: Extended, Dual and Quad I/O both STR and DTR
- Execute-in-place (XIP)
- PROGRAM/ERASE SUSPEND operations
- Volatile and nonvolatile configuration settings
- Software reset
- Additional reset pin for selected part numbers
- 3-byte and 4-byte address modes – enable memory access beyond 128Mb
- Dedicated 64-byte OTP area outside main memory
  - Readable and user-lockable
- Erase capability
  - Die Erase
  - Sector erase 64KB uniform granularity
  - Subsector erase 4KB, 32KB granularity
- Erase performance: 400KB/sec (64KB sector)
- Erase performance: 80KB/sec (4KB sub-sector)
- Program performance: 2MB/sec
- Security and write protection
  - Volatile and nonvolatile locking and software write protection for each 64KB sector
  - Nonvolatile configuration locking
  - Password protection
  - Hardware write protection: nonvolatile bits (BP[3:0] and TB) define protected area size
  - Program/erase protection during power-up
  - CRC detects accidental changes to raw data
- Electronic signature
  - JEDEC-standard 3-byte signature (BA21h)
  - Extended device ID: two additional bytes identify device factory options
- JESD47H-compliant
  - Minimum 100,000 ERASE cycles per sector
  - Data retention: 20 years (TYP)

| Options | Marking |
|---|---|
| • Voltage | |
| – 2.7–3.6V | L |
| • Density | |
| – 1Gb | 01G |
| • Device stacking | |
| – 2 die stacked | B |
| • Device generation | B |
| • Die revision | B |
| • Pin configuration | |
| – HOLD# | 1 |
| – RESET and HOLD# | 8 |
| • Sector Size | |
| – 64KB | E |
| • Packages – JEDEC-standard, RoHS-compliant | |
| – 24-ball T-PBGA 05/6mm x 8mm (TBGA24) | 12 |
| – 16-pin SOP2, 300 mils (SO16W, SO16-Wide, SOIC-16) | SF |
| – W-PDFN-8 8mm x 6mm (MLP8 8mm x 6mm) | W9 |
| • Security features | |
| – Standard security | 0 |
| • Special options | |
| – Standard | S |
| – Automotive | A |
| • Operating temperature range | |
| – From –40°C to +85°C | IT |
| – From –40°C to +105°C | AT |
| – From –40°C to +125°C | UT |

## Part Number Ordering

Micron Serial NOR Flash devices are available in different configurations and densities. Verify valid part numbers by using Micron's part catalog search at www.micron.com. To compare features and specifications by device type, visit www.micron.com/products. Contact the factory for devices not found.

### Figure 1: Part Number Ordering Information




Note: 1. WLCSP package codes, package size, and availability are density-specific. Contact the factory for availability.

# Contents

# List of Figures

# List of Tables

# Important Notes and Warnings

Micron Technology, Inc. ("Micron") reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions. This document supersedes and replaces all information supplied prior to the publication hereof. You may not rely on any information set forth in this document if you obtain the product described herein from any unauthorized distributor or other source not authorized by Micron.

**Automotive Applications.** Products are not designed or intended for use in automotive applications unless specifically designated by Micron as automotive-grade by their respective data sheets. Distributor and customer/distributor shall assume the sole risk and liability for and shall indemnify and hold Micron harmless against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, death, or property damage resulting directly or indirectly from any use of non-automotive-grade products in automotive applications. Customer/distributor shall ensure that the terms and conditions of sale between customer/distributor and any customer of distributor/customer (1) state that Micron products are not designed or intended for use in automotive applications unless specifically designated by Micron as automotive-grade by their respective data sheets and (2) require such customer of distributor/customer to indemnify and hold Micron harmless against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, death, or property damage resulting from any use of non-automotive-grade products in automotive applications.

**Critical Applications.** Products are not authorized for use in applications in which failure of the Micron component could result, directly or indirectly in death, personal injury, or severe property or environmental damage ("Critical Applications"). Customer must protect against death, personal injury, and severe property and environmental damage by incorporating safety design measures into customer's applications to ensure that failure of the Micron component will not result in such harms. Should customer or distributor purchase, use, or sell any Micron component for any critical application, customer and distributor shall indemnify and hold harmless Micron and its subsidiaries, subcontractors, and affiliates and the directors, officers, and employees of each against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such critical application, whether or not Micron or its subsidiaries, subcontractors, or affiliates were negligent in the design, manufacture, or warning of the Micron product.

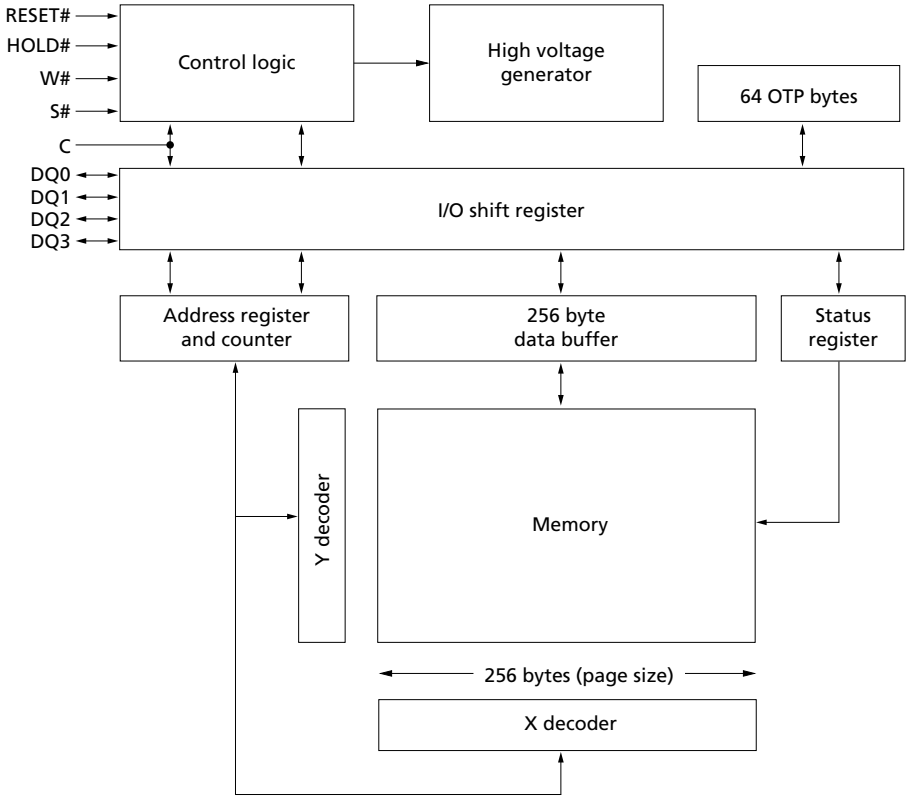
**Customer Responsibility.** Customers are responsible for the design, manufacture, and operation of their systems, applications, and products using Micron products. ALL SEMICONDUCTOR PRODUCTS HAVE INHERENT FAILURE RATES AND LIMITED USEFUL LIVES. IT IS THE CUSTOMER'S SOLE RESPONSIBILITY TO DETERMINE WHETHER THE MICRON PRODUCT IS SUITABLE AND FIT FOR THE CUSTOMER'S SYSTEM, APPLICATION, OR PRODUCT. Customers must ensure that adequate design, manufacturing, and operating safeguards are included in customer's applications and products to eliminate the risk that personal injury, death, or severe property or environmental damages will result from failure of any semiconductor component.

**Limited Warranty.** In no event shall Micron be liable for any indirect, incidental, punitive, special or consequential damages (including without limitation lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort, warranty, breach of contract or other legal theory, unless explicitly stated in a written agreement executed by Micron's duly authorized representative.

# Device Description

The MT25Q is a high-performance multiple input/output serial Flash memory device. It features a high-speed SPI-compatible bus interface, execute-in-place (XIP) functionality, advanced write protection mechanisms, and extended address access. Innovative, high-performance, dual and quad input/output commands enable double or quadruple the transfer bandwidth for READ and PROGRAM operations.

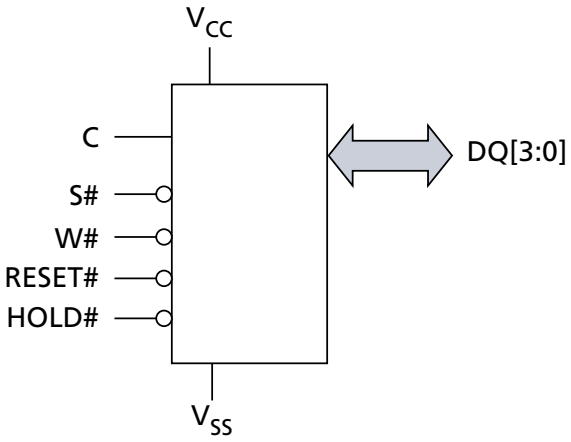
**Figure 2: Block Diagram**




Note: 1. Each page of memory can be individually programmed, but the device is not page-erasable.

## Device Logic Diagram

**Figure 3: Logic Diagram**



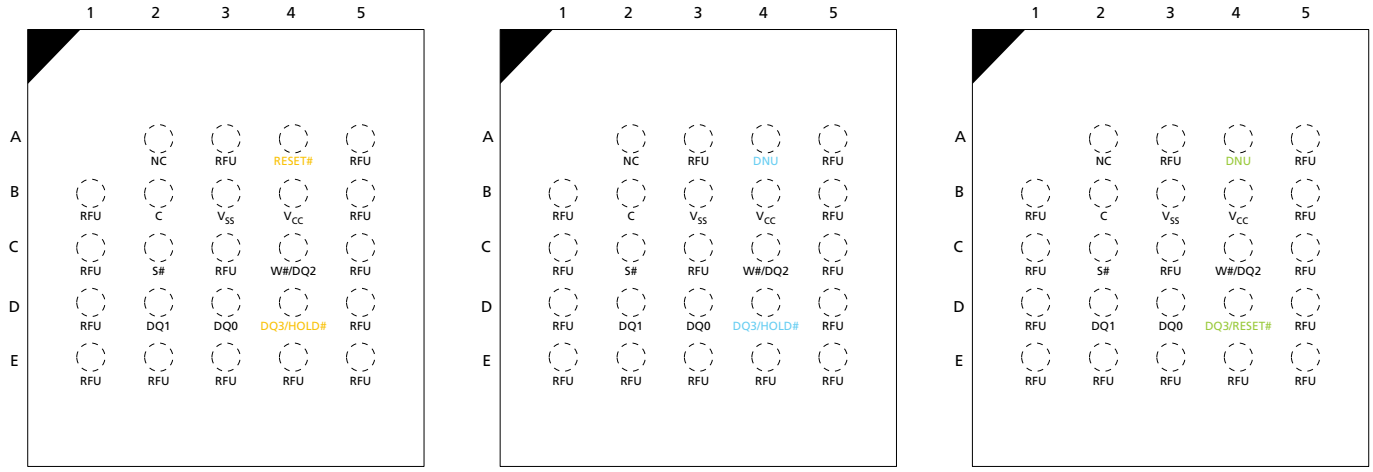

Notes:
1. Depending on the selected device (see Part Numbering Ordering Information), DQ3 = DQ3/RESET# or DQ3/HOLD#.
2. A separate RESET pin is available on dedicated part numbers (see Part Numbering Ordering Information).

## Advanced Security Protection

The device offers an advanced security protection scheme where each sector can be independently locked, by either volatile or nonvolatile locking features. The nonvolatile locking configuration can also be locked, as well password-protected. See Block Protection Settings and Sector and Password Protection for more details.

# Signal Assignments – Package Code: 12

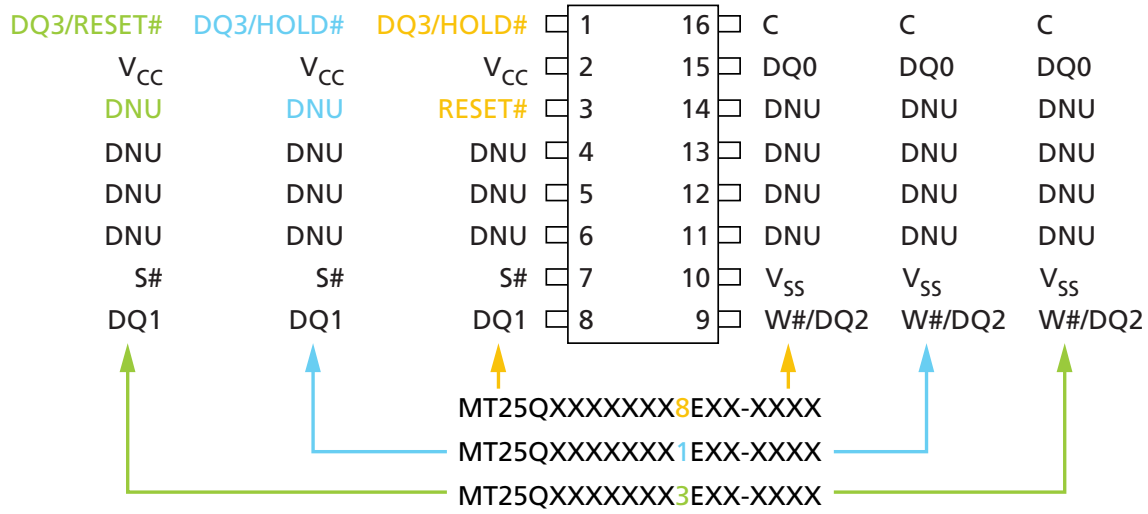
**Figure 4: 24-Ball T-BGA, 5 x 5 (Balls Down)**




Notes:
1. RESET# or HOLD# signals can share ball D4 with DQ3, depending on the selected device (see Part Numbering Ordering Information). When using single and dual I/O commands on these parts, DQ3 must be driven HIGH by the host, or an external pull-up resistor must be placed on the PCB, in order to avoid allowing the HOLD# or RESET# input to float.
2. Ball A4 = RESET# or DNU, depending on the part number. This signal has an internal pull-up resistor and may be left unconnected if not used.

# Signal Assignments – Package Code: SF

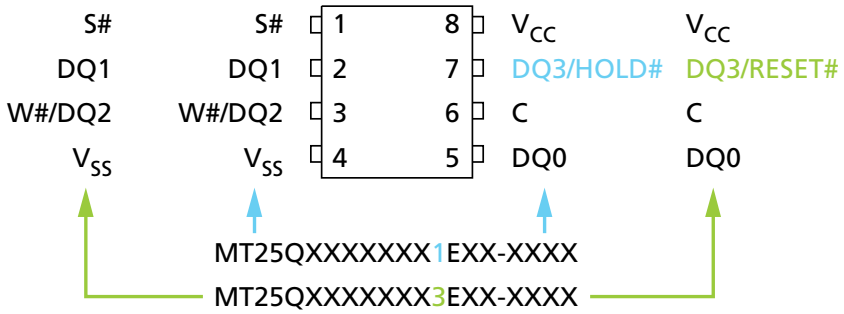
**Figure 5: 16-Pin, Plastic Small Outline – SO16 (Top View)**




Notes:
1. RESET# or HOLD# signals can share pin 1 with DQ3, depending on the selected device (see Part Numbering Ordering Information). When using single and dual I/O commands on these parts, DQ3 must be driven HIGH by the host, or an external pull-up resistor must be placed on the PCB, in order to avoid allowing the HOLD# or RESET# input to float.
2. Pin 3 = RESET# or DNU, depending on the part number. This signal has an internal pull-up resistor and may be left unconnected if not used.

# Signal Assignments – Package Code: W9

**Figure 6: 8-Pin, W-PDFN (Top View)**




Notes:
1. RESET# or HOLD# signals can share Pin 7 with DQ3, depending on the selected device (see Part Numbering Ordering Information). When using single and dual I/O commands on these parts, DQ3 must be driven high by the host, or an external pull-up resistor must be placed on the PCB, in order to avoid allowing the HOLD# or RESET# input to float.
2. On the underside of the W-PDFN package, there is an exposed central pad that is pulled internally to $V_{SS}$. It can be left floating or can be connected to $V_{SS}$. It must not be connected to any other voltage or signal line on the PCB.
3. MT25QXXXXXXX8EXX-XXXX not available on 8 pin package

# Signal Descriptions

The signal description table below is a comprehensive list of signals for the MT25Q family devices. All signals listed may not be supported on this device. See Signal Assignments for information specific to this device.

**Table 1: Signal Descriptions**

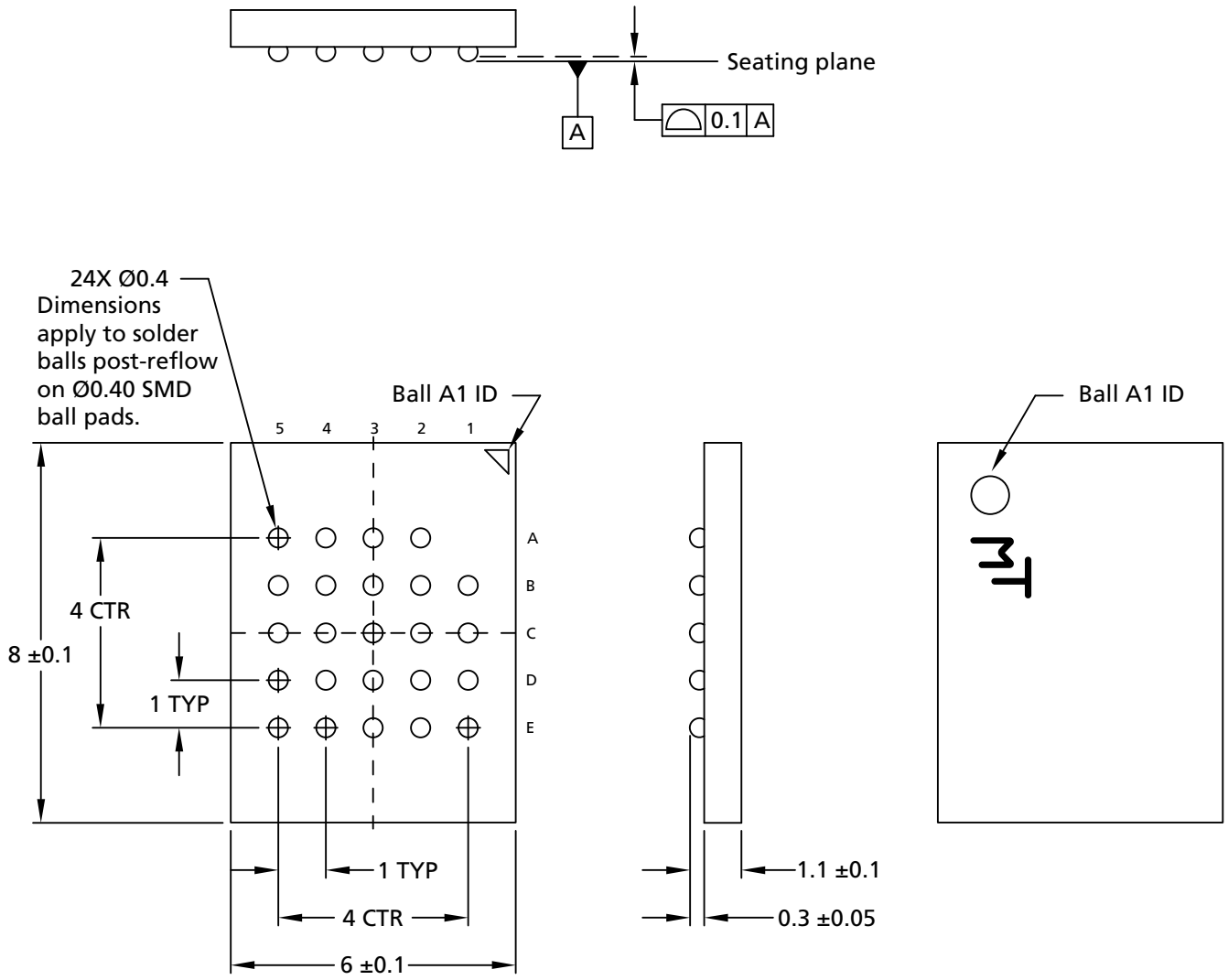
| Symbol | Type | Description |
|---|---|---|
| S# | Input | **Chip select:** When S# is driven HIGH, the device will enter standby mode, unless an internal PROGRAM, ERASE, or WRITE STATUS REGISTER cycle is in progress. All other input pins are ignored and the output pins are tri-stated. On parts with the pin configuration offering a dedicated RESET# pin, however, the RESET# input pin remains active even when S# is HIGH.<br><br>Driving S# LOW enables the device, placing it in the active mode.<br><br>After power-up, a falling edge on S# is required prior to the start of any command. |
| C | Input | **Clock:** Provides the timing of the serial interface. Command inputs are latched on the rising edge of the clock. In STR commands or protocol, address and data inputs are latched on the rising edge of the clock, while data is output on the falling edge of the clock. In DTR commands or protocol, address and data inputs are latched on both edges of the clock, and data is output on both edges of the clock. |
| RESET# | Input | **RESET#:** When RESET# is driven LOW, the device is reset and the outputs are tri-stated. If RESET# is driven LOW while an internal WRITE, PROGRAM, or ERASE operation is in progress, data may be lost. The RESET# functionality can be disabled using bit 4 of the nonvolatile configuration register or bit 4 of the enhanced volatile configuration register.<br><br>For pin configurations that share the DQ3 pin with RESET#, the RESET# functionality is disabled in QIO-SPI mode. |
| HOLD# | Input | **HOLD:** Pauses serial communications with the device without deselecting or resetting the device. Outputs are tri-stated and inputs are ignored. The HOLD# functionality can be disabled using bit 4 of the nonvolatile configuration register or bit 4 of the enhanced volatile configuration register.<br><br>For pin configurations that share the DQ3 pin with HOLD#, the HOLD# functionality is disabled in QIO-SPI mode or when DTR operation is enabled. |
| W# | Input | **Write protect:** Freezes the status register in conjunction with the enable/disable bit of the status register. When the enable/disable bit of the status register is set to 1 and the W# signal is driven LOW, the status register nonvolatile bits become read-only and the WRITE STATUS REGISTER operation will not execute. During the extended-SPI protocol with QOFR and QIOFR instructions, and with QIO-SPI protocol, this pin function is an input/output as DQ2 functionality. This signal does not have internal pull-ups, it cannot be left floating and must be driven, even if none of W#/DQ2 function is used. |
| DQ[3:0] | I/O | **Serial I/O:** The bidirectional DQ signals transfer address, data, and command information.<br><br>When using legacy (x1) SPI commands in extended I/O protocol (XIO-SPI), DQ0 is an input and DQ1 is an output. DQ[3:2] are not used.<br><br>When using dual commands in XIO-SPI or when using DIO-SPI, DQ[1:0] are I/O. DQ[3:2] are not used.<br><br>When using quad commands in XIO-SPI or when using QIO-SPI, DQ[3:0] are I/O. |
| $V_{CC}$ | Supply | Core and I/O power supply. |

**Table 1: Signal Descriptions (Continued)**

| Symbol | Type | Description |
|---|---|---|
| $V_{SS}$ | Supply | Core and I/O ground connection. |
| DNU | – | **Do not use:** Do not connect to any other signal, or power supply; must be left floating. |
| RFU | – | **Reserved for future use:** Reserved by Micron for future device functionality and enhancement. Recommend that these be left floating. May be connected internally, but external connections will not affect operation. |
| NC | – | **No connect:** No internal connection; can be driven or floated. |

# Package Dimensions – Package Code: 12

**Figure 7: 24-Ball T-PBGA (5 x 5 ball grid array) – 6mm x 8mm**

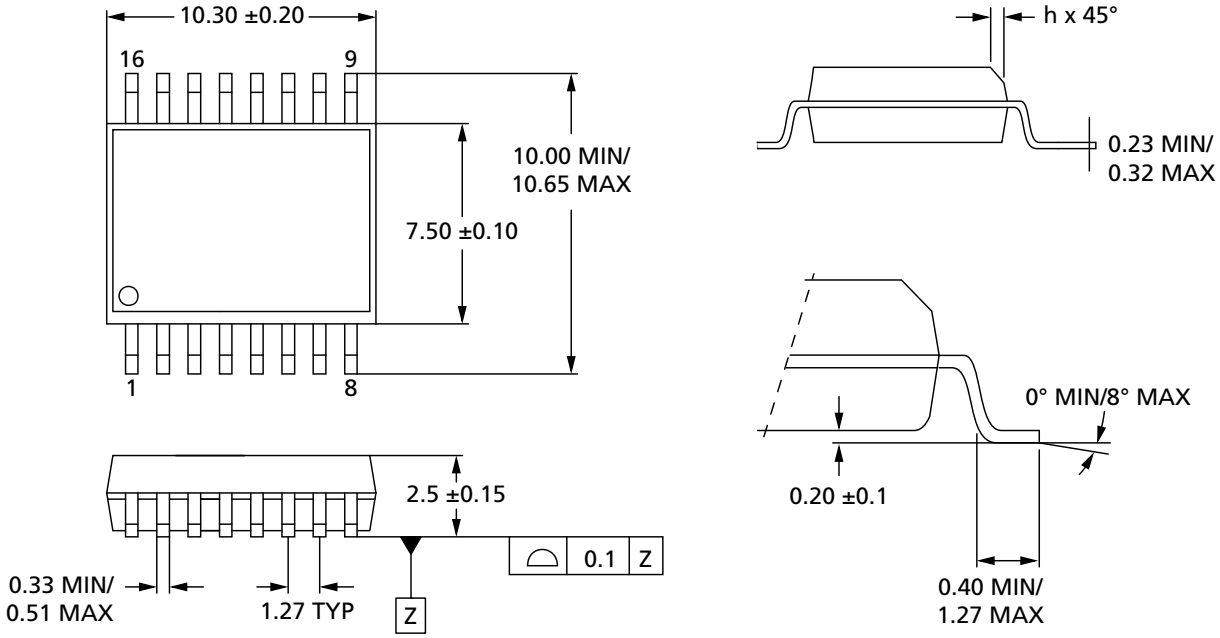



Notes: 1. All dimensions are in millimeters.

2. See Part Number Ordering Information for complete package names and details.

# Package Dimensions – Package Code: SF

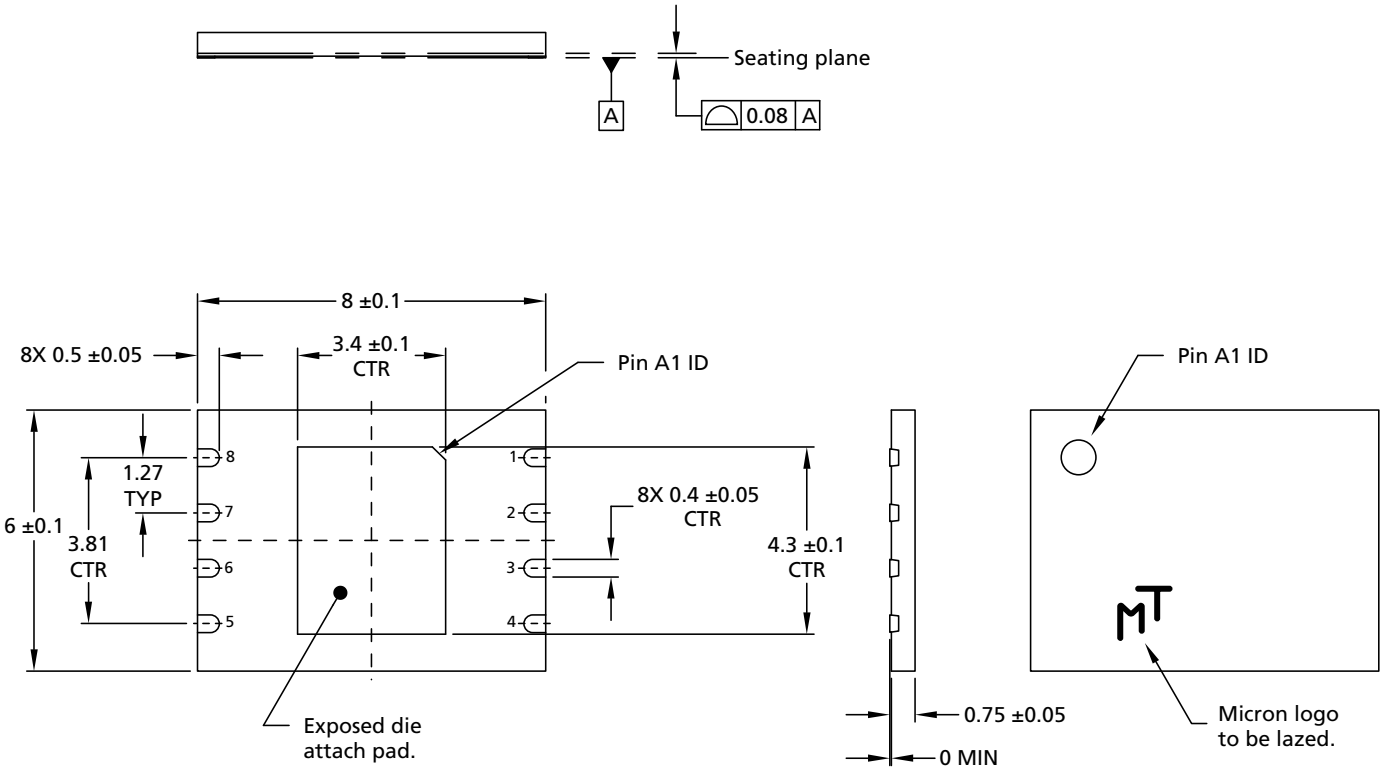
**Figure 8: 16-Pin SOP2 – 300 Mils Body Width**




Notes:
1. All dimensions are in millimeters.
2. See Part Number Ordering Information for complete package names and details.

# Package Dimensions – Package Code: W9

**Figure 9: W-PDFN-8 (MLP8) – 8mm x 6mm**




Notes:
1. All dimensions are in millimeters.
2. See Part Number Ordering Information for complete package names and details.

# Memory Map – 1Gb Density

**Table 2: Memory Map**

| Sector | Subsector (32KB) | Subsector (4KB) | Address Range | |
|---|---|---|---|---|
| | | | Start | End |
| 2047 | 4095 | 32767 | 07FF F000h | 07FF FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 32760 | 07FF 8000h | 07FF 8FFFh |
| | 4094 | 32759 | 07FF 7000h | 07FF 7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 32752 | 07FF 0000h | 07FF 0FFFh |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1023 | 2047 | 16383 | 03FF F000h | 03FF FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 16376 | 03FF 8000h | 03FF 8FFFh |
| | 2046 | 16375 | 03FF 7000h | 03FF 7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 16368 | 03FF 0000h | 03FF 0FFFh |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 511 | 1023 | 8191 | 01FF F000h | 01FF FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 8184 | 01FF 8000h | 01FF 8FFFh |
| | 1022 | 8183 | 01FF 7000h | 01FF 7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 8176 | 01FF 0000h | 01FF 0FFFh |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 0 | 1 | 15 | 0000 F000h | 0000 FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 8 | 0000 8000h | 0000 8FFFh |
| | 0 | 7 | 0000 7000h | 0000 7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 0 | 0000 0000h | 0000 0FFFh |

Note: 1. See Part Number Ordering Information, Sector Size – Part Numbers table for options.

## Status Register

Status register bits can be read from or written to using READ STATUS REGISTER or WRITE STATUS REGISTER commands, respectively. When the status register enable/disable bit (bit 7) is set to 1 and W# is driven LOW, the status register nonvolatile bits become read-only and the WRITE STATUS REGISTER operation will not execute. The only way to exit this hardware-protected mode is to drive W# HIGH.

**Table 3: Status Register**

| Bit | Name | Settings | Description | Notes |
|---|---|---|---|---|
| 7 | Status register write enable/disable | 0 = Enabled (Default)<br>1 = Disabled | **Nonvolatile control bit:** Used with W# to enable or disable writing to the status register. | |
| 5 | Top/bottom | 0 = Top (Default)<br>1 = Bottom | **Nonvolatile control bit:** Determines whether the protected memory area defined by the block protect bits starts from the top or bottom of the memory array. | |
| 6, 4:2 | BP[3:0] | See Protected Area tables | **Nonvolatile control bit:** Defines memory to be software protected against PROGRAM or ERASE operations. When one or more block protect bits is set to 1, a designated memory area is protected from PROGRAM and ERASE operations. | 1 |
| 1 | Write enable latch | 0 = Clear (Default)<br>1 = Set | **Volatile control bit:** The device always powers up with this bit cleared to prevent inadvertent WRITE, PROGRAM, or ERASE operations. To enable these operations, the WRITE ENABLE operation must be executed first to set this bit. | |
| 0 | Write in progress | 0 = Ready<br>1 = Busy | **Status bit:** Indicates if one of the following command cycles is in progress:<br>WRITE STATUS REGISTER<br>WRITE NONVOLATILE CONFIGURATION REGISTER<br>PROGRAM<br>ERASE | 2 |

Notes:
1. The DIE ERASE command is executed only if all bits = 0.
2. Status register bit 0 is the inverse of flag status register bit 7.

## Block Protection Settings

### Table 4: Protected Area

| Status Register Content | | | | | Protected Area (Sectors) | Unprotected Area (Sectors) |
|---|---|---|---|---|---|---|
| Top/ Bottom | BP3 | BP2 | BP1 | BP0 | | |
| 0 | 0 | 0 | 0 | 0 | None | All |
| 0 | 0 | 0 | 0 | 1 | 2047 | 2046:0 |
| 0 | 0 | 0 | 1 | 0 | 2047:2046 | 2045:0 |
| 0 | 0 | 0 | 1 | 1 | 2047:2044 | 2043:0 |
| 0 | 0 | 1 | 0 | 0 | 2047:2040 | 2039:0 |
| 0 | 0 | 1 | 0 | 1 | 2047:2032 | 2031:0 |
| 0 | 0 | 1 | 1 | 0 | 2047:2016 | 2015:0 |
| 0 | 0 | 1 | 1 | 1 | 2047:1984 | 1983:0 |
| 0 | 1 | 0 | 0 | 0 | 2047:1920 | 1919:0 |
| 0 | 1 | 0 | 0 | 1 | 2047:1792 | 1791:0 |
| 0 | 1 | 0 | 1 | 0 | 2047:1536 | 1535:0 |
| 0 | 1 | 0 | 1 | 1 | 2047:1024 | 1023:0 |
| 0 | 1 | 1 | 0 | 0 | All | None |
| 0 | 1 | 1 | 0 | 1 | All | None |
| 0 | 1 | 1 | 1 | 0 | All | None |
| 0 | 1 | 1 | 1 | 1 | All | None |
| 1 | 0 | 0 | 0 | 0 | None | All |
| 1 | 0 | 0 | 0 | 1 | Sector 0 | 2047:1 |
| 1 | 0 | 0 | 1 | 0 | 1:0 | 2047:2 |
| 1 | 0 | 0 | 1 | 1 | 3:0 | 2047:4 |
| 1 | 0 | 1 | 0 | 0 | 7:0 | 2047:8 |
| 1 | 0 | 1 | 0 | 1 | 15:0 | 2047:16 |
| 1 | 0 | 1 | 1 | 0 | 31:0 | 2047:32 |
| 1 | 0 | 1 | 1 | 1 | 63:0 | 2047:64 |
| 1 | 1 | 0 | 0 | 0 | 127:0 | 2047:128 |
| 1 | 1 | 0 | 0 | 1 | 255:0 | 2047:256 |
| 1 | 1 | 0 | 1 | 0 | 511:0 | 2047:512 |
| 1 | 1 | 0 | 1 | 1 | 1023:0 | 2047:1024 |
| 1 | 1 | 1 | 0 | 0 | All | None |
| 1 | 1 | 1 | 0 | 1 | All | None |
| 1 | 1 | 1 | 1 | 0 | All | None |
| 1 | 1 | 1 | 1 | 1 | All | None |

# Flag Status Register

Flag status register bits are read by using READ FLAG STATUS REGISTER command. All bits are volatile and are reset to zero on power-up.

Status bits are set and reset automatically by the internal controller. Error bits must be cleared through the CLEAR STATUS REGISTER command.

**Table 5: Flag Status Register**

| Bit | Name | Settings | Description |
|---|---|---|---|
| 7 | Program or erase controller | 0 = Busy<br>1 = Ready | **Status bit:** Indicates whether one of the following command cycles is in progress: WRITE STATUS REGISTER, WRITE NONVOLATILE CONFIGURATION REGISTER, PROGRAM, or ERASE. |
| 6 | Erase suspend | 0 = Clear<br>1 = Suspend | **Status bit:** Indicates whether an ERASE operation has been or is going to be suspended. |
| 5 | Erase | 0 = Clear<br>1 = Failure or protection error | **Error bit:** Indicates whether an ERASE operation has succeeded or failed. |
| 4 | Program | 0 = Clear<br>1 = Failure or protection error | **Error bit:** Indicates whether a PROGRAM operation has succeeded or failed. It indicates, also, whether a CRC check has succeeded or failed. |
| 3 | Reserved | 0 | Reserved |
| 2 | Program suspend | 0 = Clear<br>1 = Suspend | **Status bit:** Indicates whether a PROGRAM operation has been or is going to be suspended. |
| 1 | Protection | 0 = Clear<br>1 = Failure or protection error | **Error bit:** Indicates whether an ERASE or PROGRAM operation has attempted to modify the protected array sector, or whether a PROGRAM operation has attempted to access the locked OTP space. |
| 0 | Addressing | 0 = 3-byte addressing<br>1 = 4-byte addressing | **Status bit:** Indicates whether 3-byte or 4-byte address mode is enabled. |

# Extended Address Register

The 3-byte address mode can only access 128Mb of memory. To access the full device in 3-byte address mode, the device includes an extended address register that indirectly provides a fourth address byte A[31:24]. The extended address register bits [2:0] operate as memory address bit A[26:24] to select one of the eight 128Mb segments of the memory array.

If 4-byte addressing is enabled, the extended address register settings are ignored.

**Table 6: Extended Address Register**

| Bit | Name | Settings | Description |
|---|---|---|---|
| 7:3 | A[31:27] | 00000 | Reserved |
| 2:0 | A[26:24] | 111 = Highest 128Mb segment<br>110 = Seventh 128Mb segment<br>101 = Sixth 128Mb segment<br>100 = Fifth 128Mb segment<br>011 = Fourth 128Mb segment<br>010 = Third 128Mb segment<br>001 = Second 128Mb segment<br>000 = Lowest 128Mb segment | Enables specified 128Mb memory segment. The default setting (lowest) can be changed to the highest 128Mb segment using bit 1 of the nonvolatile configuration register. |

The PROGRAM and ERASE operations act upon the 128Mb segment selected in the extended address register. The DIE ERASE operation erases the selected die.

The READ operation begins reading in the selected 128Mb segment, but is not bound by it.

In a continuous READ, when the last byte of the segment is read, the next byte output is the first byte of the next segment. The operation wraps to 0000000h; therefore, a download of the whole array is possible with one READ operation.

The value of the extended address register does not change when a READ operation crosses the selected 128Mb boundary.

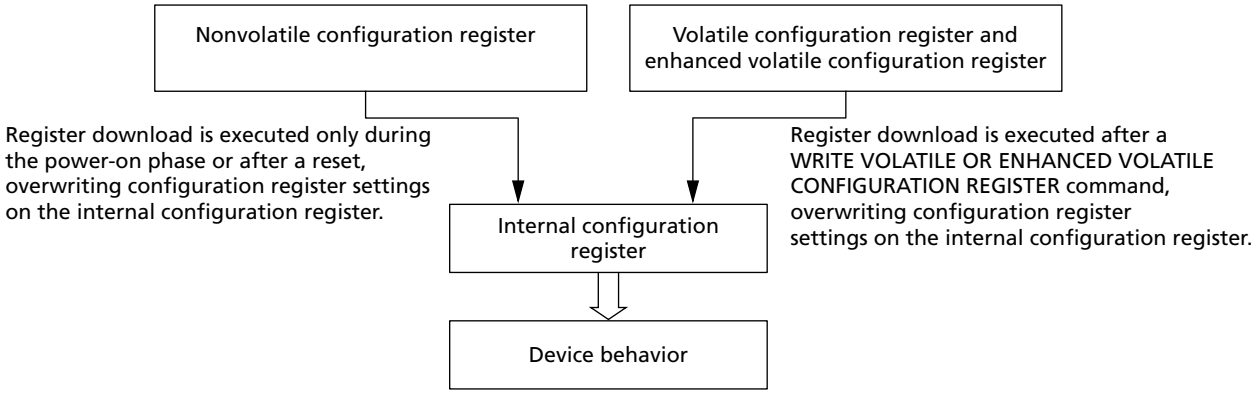
# Internal Configuration Register

The memory configuration is set by an internal configuration register that is not directly accessible to users.

The user can change the default configuration at power up by using the WRITE NONVOLATILE CONFIGURATION REGISTER. Information from the nonvolatile configuration register overwrites the internal configuration register during power-on or after a reset.

The user can change the configuration during operation by using the WRITE VOLATILE CONFIGURATION REGISTER or the WRITE ENHANCED VOLATILE CONFIGURATION REGISTER commands. Information from the volatile configuration registers overwrite the internal configuration register immediately after the WRITE command completes.

**Figure 10: Internal Configuration Register**



# Nonvolatile Configuration Register

This register is read from and written to using the READ NONVOLATILE CONFIGURATION REGISTER and the WRITE NONVOLATILE CONFIGURATION REGISTER commands, respectively. A register download is executed during power-on or after reset, overwriting the internal configuration register settings that determine device behavior.

**Table 7: Nonvolatile Configuration Register**

| Bit | Name | Settings | Description | Notes |
|---|---|---|---|---|
| 15:12 | Number of dummy clock cycles | 0000 = Identical to 1111<br>0001 = 1<br>0010 = 2<br>⋮<br>1101 = 13<br>1110 = 14<br>1111 = Default | Sets the number of dummy clock cycles subsequent to all FAST READ commands.<br>(See the Command Set Table for default setting values.) | 1 |
| 11:9 | XIP mode at power-on reset | 000 = XIP: Fast read<br>001 = XIP: Dual output fast read<br>010 = XIP: Dual I/O fast read<br>011 = XIP: Quad output fast read<br>100 = XIP: Quad I/O fast read<br>101 = Reserved<br>110 = Reserved<br>111 = Disabled (Default) | Enables the device to operate in the selected XIP mode immediately after power-on reset. | |
| 8:6 | Output driver strength | 000 = Reserved<br>001 = 90 Ohms<br>010 = Reserved<br>011 = 45 Ohms<br>100 = Reserved<br>101 = 20 Ohms<br>110 = Reserved<br>111 = 30 Ohms (Default) | Optimizes the impedance at $V_{CC}/2$ output voltage. | |
| 5 | Double transfer rate protocol | 0 = Enabled<br>1 = Disabled (Default) | Set DTR protocol as current one. Once enabled, all commands will work in DTR. | |
| 4 | Reset/hold | 0 = Disabled<br>1 = Enabled (Default) | Enables or disables HOLD# or RESET# on DQ3. | |
| 3 | Quad I/O protocol | 0 = Enabled<br>1 = Disabled (Default) | Enables or disables quad I/O command input (4-4-4 mode). | 2 |
| 2 | Dual I/O protocol | 0 = Enabled<br>1 = Disabled (Default) | Enables or disables dual I/O command input (2-2-2 mode). | 2 |
| 1 | 128Mb segment select | 0 = Highest 128Mb segment<br>1 = Lowest 128Mb segment (Default) | Selects the power-on default 128Mb segment for 3-byte address operations. See also the extended address register. | |

**Table 7: Nonvolatile Configuration Register (Continued)**

| Bit | Name | Settings | Description | Notes |
|---|---|---|---|---|
| 0 | Number of address bytes during command entry | 0 = Enable 4-byte address mode<br>1 = Enable 3-byte address mode (Default) | Defines the number of address bytes for a command. | |

Notes:
1. The number of cycles must be set to accord with the clock frequency, which varies by the type of FAST READ command (See Supported Clock Frequencies table). Insufficient dummy clock cycles for the operating frequency causes the memory to read incorrect data.
2. When bits 2 and 3 are both set to 0, the device operates in quad I/O protocol.

# Volatile Configuration Register

This register is read from and written to by the READ VOLATILE CONFIGURATION REGISTER and the WRITE VOLATILE CONFIGURATION REGISTER commands, respectively. A register download is executed after these commands, overwriting the internal configuration register settings that determine device memory behavior.

**Table 8: Volatile Configuration Register**

| Bit | Name | Settings | Description | Notes |
|---|---|---|---|---|
| 7:4 | Number of dummy clock cycles | 0000 = Identical to 1111<br>0001 = 1<br>0010 = 2<br>⋮<br>1101 = 13<br>1110 = 14<br>1111 = Default | Sets the number of dummy clock cycles subsequent to all FAST READ commands.<br>(See the Command Set Table for default setting values.) | 1 |
| 3 | XIP | 0 = Enable<br>1 = Disable (Default) | Enables or disables XIP. | |
| 2 | Reserved | 0 | 0b = Fixed value. | |
| 1:0 | Wrap | 00 = 16-byte boundary aligned | 16-byte wrap: Output data wraps within an aligned 16-byte boundary starting from the 3-byte address issued after the command code. | 2 |
| | | 01 = 32-byte boundary aligned | 32-byte wrap: Output data wraps within an aligned 32-byte boundary starting from the 3-byte address issued after the command code. | |
| | | 10 = 64-byte boundary aligned | 64-byte wrap: Output data wraps within an aligned 64-byte boundary starting from the 3-byte address issued after the command code. | |
| | | 11 = Continuous (Default) | Continuously sequences addresses through the entire array. | |

Notes:
1. The number of cycles must be set according to and sufficient for the clock frequency, which varies by the type of FAST READ command, as shown in the Supported Clock Frequencies table. An insufficient number of dummy clock cycles for the operating frequency causes the memory to read incorrect data.
2. See the Sequence of Bytes During Wrap table.

**Table 9: Sequence of Bytes During Wrap**

| Starting Address | 16-Byte Wrap | 32-Byte Wrap | 64-Byte Wrap |
|---|---|---|---|
| 0 | 0-1-2- . . . -15-0-1- . . | 0-1-2- . . . -31-0-1- . . | 0-1-2- . . . -63-0-1- . . |
| 1 | 1-2- . . . -15-0-1-2- . . | 1-2- . . . -31-0-1-2- . . | 1-2- . . . -63-0-1-2- . . |
| .... | .... | .... | .... |
| 15 | 15-0-1-2-3- . . . -15-0-1- . . | 15-16-17- . . . -31-0-1- . . | 15-16-17- . . . -63-0-1- . . |
| .... | .... | .... | .... |
| 31 | – | 31-0-1-2-3- . . . -31-0-1- . . | 31-32-33- . . . -63-0-1- . . |
| .... | .... | .... | .... |
| 63 | – | – | 63-0-1- . . . -63-0-1- . . |

## Supported Clock Frequencies

### Table 10: Clock Frequencies – STR (in MHz) for IT and AT parts

Notes apply to entire table

| Number of Dummy Clock Cycles | FAST READ | DUAL OUTPUT FAST READ | DUAL I/O FAST READ | QUAD OUTPUT FAST READ | QUAD I/O FAST READ |
|---|---|---|---|---|---|
| 1 | 94 | 79 | 60 | 44 | 39 |
| 2 | 112 | 97 | 77 | 61 | 48 |
| 3 | 129 | 106 | 86 | 78 | 58 |
| 4 | 133 | 115 | 97 | 97 | 69 |
| 5 | 133 | 125 | 106 | 106 | 78 |
| 6 | 133 | 133 | 115 | 115 | 86 |
| 7 | 133 | 133 | 125 | 125 | 97 |
| 8 | 133 | 133 | 133 | 133 | 106 |
| 9 | 133 | 133 | 133 | 133 | 115 |
| 10 | 133 | 133 | 133 | 133 | 125 |
| 11 : 14 | 133 | 133 | 133 | 133 | 133 |

Notes:
1. Values are guaranteed by characterization and not 100% tested in production.
2. A tuning data pattern (TDP) capability provides applications with data patterns for adjusting the data latching point at the host end when the clock frequency is set higher than 133 MHz in STR mode and higher than 66 MHz in double transfer rate (DTR) mode. For additional details, refer to TN-25-07: Tuning Data Pattern for MT25Q and MT25T Devices.

### Table 11: Clock Frequencies – STR (in MHz) for UT parts

Notes apply to entire table

| Number of Dummy Clock Cycles | FAST READ | DUAL OUTPUT FAST READ | DUAL I/O FAST READ | QUAD OUTPUT FAST READ | QUAD I/O FAST READ |
|---|---|---|---|---|---|
| 1 | 94 | 79 | 60 | 44 | 39 |
| 2 | 112 | 97 | 77 | 61 | 48 |
| 3 | 129 | 106 | 86 | 78 | 58 |
| 4 | 133 | 115 | 97 | 97 | 69 |
| 5 | 133 | 125 | 106 | 106 | 78 |
| 6 | 133 | 133 | 115 | 108 | 86 |
| 7 | 133 | 133 | 125 | 108 | 97 |
| 8 | 133 | 133 | 133 | 108 | 106 |
| 9 : 14 | 133 | 133 | 133 | 108 | 108 |

Notes:
1. Values are guaranteed by characterization and not 100% tested in production.
2. A tuning data pattern (TDP) capability provides applications with data patterns for adjusting the data latching point at the host end when the clock frequency is set higher than 133 MHz in STR mode and higher than 66 MHz in double transfer rate (DTR) mode.

For additional details, refer to TN-25-07: Tuning Data Pattern for MT25Q and MT25T Devices.

### Table 12: Clock Frequencies – DTR (in MHz) for IT and AT parts

Notes apply to entire table

| Number of Dummy Clock Cycles | FAST READ | DUAL OUTPUT FAST READ | DUAL I/O FAST READ | QUAD OUTPUT FAST READ | QUAD I/O FAST READ |
|---|---|---|---|---|---|
| 1 | 59 | 45 | 40 | 26 | 20 |
| 2 | 73 | 59 | 49 | 40 | 30 |
| 3 | 82 | 68 | 59 | 59 | 39 |
| 4 | 90 | 76 | 65 | 65 | 49 |
| 5 | 90 | 83 | 75 | 75 | 58 |
| 6 | 90 | 90 | 83 | 83 | 68 |
| 7 | 90 | 90 | 90 | 90 | 78 |
| 8 | 90 | 90 | 90 | 90 | 85 |
| 9 | 90 | 90 | 90 | 90 | 90 |
| 10 : 14 | 90 | 90 | 90 | 90 | 90 |

Notes:
1. Values are guaranteed by characterization and not 100% tested in production.
2. A tuning data pattern (TDP) capability provides applications with data patterns for adjusting the data latching point at the host end when the clock frequency is set higher than 133 MHz in STR mode and higher than 66 MHz in double transfer rate (DTR) mode. For additional details, refer to TN-25-07: Tuning Data Pattern for MT25Q and MT25T Devices.

### Table 13: Clock Frequencies – DTR (in MHz) for UT parts

Notes apply to entire table

| Number of Dummy Clock Cycles | FAST READ | DUAL OUTPUT FAST READ | DUAL I/O FAST READ | QUAD OUTPUT FAST READ | QUAD I/O FAST READ |
|---|---|---|---|---|---|
| 1 | 59 | 45 | 40 | 26 | 20 |
| 2 | 73 | 59 | 49 | 40 | 30 |
| 3 | 82 | 68 | 59 | 59 | 39 |
| 4 | 90 | 76 | 65 | 65 | 49 |
| 5 | 90 | 83 | 75 | 75 | 58 |
| 6 | 90 | 90 | 83 | 80 | 68 |
| 7 | 90 | 90 | 90 | 80 | 78 |
| 8 : 14 | 90 | 90 | 90 | 80 | 80 |

Notes:
1. Values are guaranteed by characterization and not 100% tested in production.
2. A tuning data pattern (TDP) capability provides applications with data patterns for adjusting the data latching point at the host end when the clock frequency is set higher than 133 MHz in STR mode and higher than 66 MHz in double transfer rate (DTR) mode.

For additional details, refer to TN-25-07: Tuning Data Pattern for MT25Q and MT25T Devices.

# Enhanced Volatile Configuration Register

This register is read from and written to using the READ ENHANCED VOLATILE CONFIGURATION REGISTER and the WRITE ENHANCED VOLATILE CONFIGURATION REGISTER commands, respectively. A register download is executed after these commands, overwriting the internal configuration register settings that determine device memory behavior.

**Table 14: Enhanced Volatile Configuration Register**

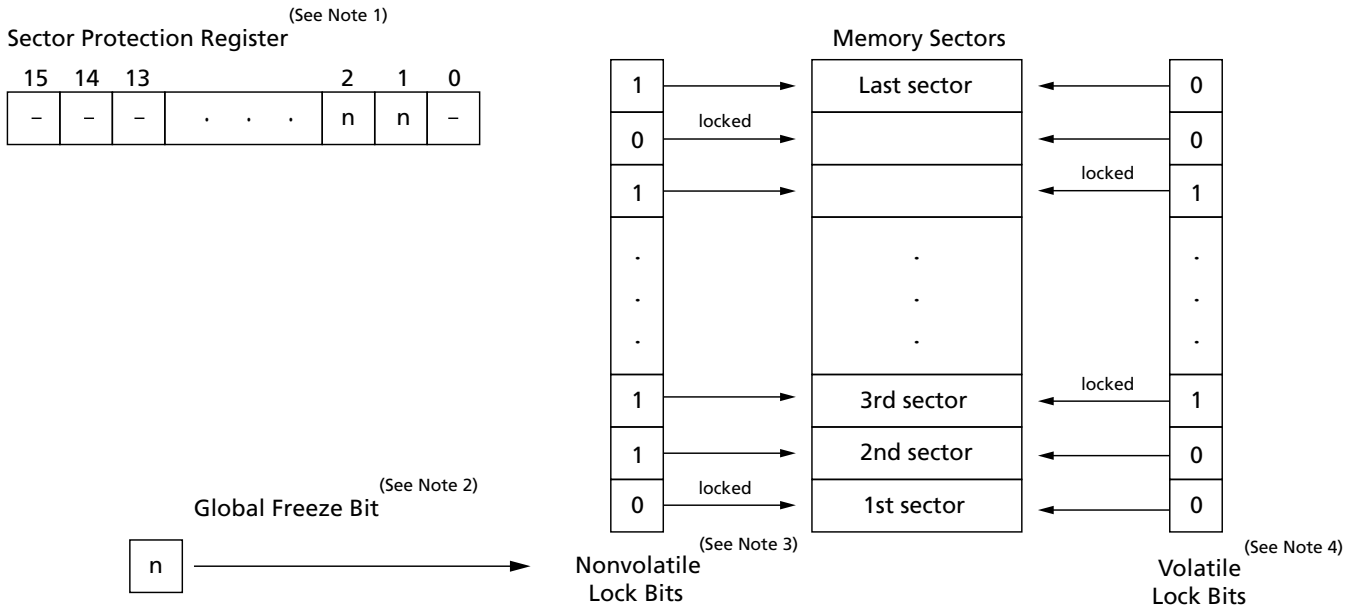
| Bit | Name | Settings | Description | Notes |
|---|---|---|---|---|
| 7 | Quad I/O protocol | 0 = Enabled<br>1 = Disabled (Default) | Enables or disables quad I/O command input (4-4-4 mode). | 1 |
| 6 | Dual I/O protocol | 0 = Enabled<br>1 = Disabled (Default) | Enables or disables dual I/O command input (2-2-2 mode). | 1 |
| 5 | Double transfer rate protocol | 0 = Enabled<br>1 = Disabled (Default, single transfer rate) | Set DTR protocol as current one. Once enabled, all commands will work in DTR. | |
| 4 | Reset/hold | 0 = Disabled<br>1 = Enabled (Default) | Enables or disables HOLD# or RESET# on DQ3. (Available only on specified part numbers.) | |
| 3 | Reserved | 1 | | |
| 2:0 | Output driver strength | 000 = Reserved<br>001 = 90 ohms<br>010 = Reserved<br>011 = 45 ohms<br>100 = Reserved<br>101 = 20 ohms<br>110 = Reserved<br>111 = 30 ohms (Default) | Optimizes the impedance at $V_{CC}$/2 output voltage. | |

Note: 1. When bits 6 and 7 are both set to 0, the device operates in quad I/O protocol. When either bit 6 or 7 is set to 0, the device operates in dual I/O or quad I/O respectively. When a bit is set, the device enters the selected protocol immediately after the WRITE ENHANCED VOLATILE CONFIGURATION REGISTER command. The device returns to the default protocol after the next power-on or reset. Also, the rescue sequence or another WRITE ENHANCED VOLATILE CONFIGURATION REGISTER command will return the device to the default protocol.

# Security Registers

Security registers enable sector and password protection on multiple levels using non-volatile and volatile register and bit settings (shown below). The applicable register tables follow.

**Figure 11: Sector and Password Protection**




Notes:
1. **Sector protection register.** This 16-bit nonvolatile register includes two active bits[2:1] to enable sector and password protection.
2. **Global freeze bit.** This volatile bit protects the settings in all nonvolatile lock bits.
3. **Nonvolatile lock bits.** Each nonvolatile bit corresponds to and provides nonvolatile protection for an individual memory sector, which remains locked (protection enabled) until its corresponding bit is cleared to 1.
4. **Volatile lock bits.** Each volatile bit corresponds to and provides volatile protection for an individual memory sector, which is locked temporarily (protection is cleared when the device is reset or powered down).
5. The first and last sectors will have volatile protections at the 4KB subsector level. Each 4KB subsector in these sectors can be individually locked by volatile lock bits setting; nonvolatile protections granularity remain at the sector level.

# Sector Protection Security Register

### Table 15: Sector Protection Register

| Bits | Name | Settings | Description | Notes |
|---|---|---|---|---|
| 15:3 | Reserved | 1 = Default | – | |
| 2 | Password protection lock | 1 = Disabled (Default)<br>0 = Enabled | **Nonvolatile bit:** When set to 1, password protection is disabled. When set to 0, password protection is enabled permanently; the 64-bit password cannot be retrieved or reset. | 1, 2 |
| 1 | Sector protection lock | 1 = Enabled, with password protection (Default)<br>0 = Enabled, without password protection | **Nonvolatile bit:** When set to 1, nonvolatile lock bits can be set to lock/unlock their corresponding memory sectors; bit 2 can be set to 0, enabling password protection permanently.<br>When set to 0, nonvolatile lock bits can be set to lock/unlock their corresponding memory sectors; bit 2 must remain set to 1, disabling password protection permanently. | 1, 3, 4 |
| 0 | Reserved | 1 = Default | – | |

Notes:
1. Bits 2 and 1 are user-configurable, one-time-programmable, and mutually exclusive in that only one of them can be set to 0. It is recommended that one of the bits be set to 0 when first programming the device.
2. The 64-bit password must be programmed and verified before this bit is set to 0 because after it is set, password changes are not allowed, thus providing protection from malicious software. When this bit is set to 0, a 64-bit password is required to reset the global freeze bit from 0 to 1. In addition, if the password is incorrect or lost, the global freeze bit can no longer be set and nonvolatile lock bits cannot be changed. (See the Sector and Password Protection figure and the Global Freeze Bit Definition table).
3. Whether this bit is set to 1 or 0, it enables programming or erasing nonvolatile lock bits (which provide memory sector protection). The password protection bit must be set beforehand because setting this bit will either enable password protection permanently (bit 2 = 0) or disable password protection permanently (bit 1 = 0).
4. By default, all sectors are unlocked when the device is shipped from the factory. Sectors are locked, unlocked, read, or locked down as explained in the Nonvolatile and Volatile Lock Bits table and the Volatile Lock Bit Register Bit Definitions table.

### Table 16: Global Freeze Bit

| Bits | Name | Settings | Description |
|---|---|---|---|
| 7:1 | Reserved | 0 | Bit values are 0 |
| 0 | Global freeze bit | 1 = Disabled (Default)<br>0 = Enabled | **Volatile bit:** When set to 1, all nonvolatile lock bits can be set to enable or disable locking their corresponding memory sectors.<br>When set to 0, nonvolatile lock bits are protected from PROGRAM or ERASE commands. This bit should not be set to 0 until the nonvolatile lock bits are set. |

Note:
1. The READ GLOBAL FREEZE BIT command enables reading this bit. When password protection is enabled, this bit is locked upon device power-up or reset. It cannot be changed without the password. After the password is entered, the UNLOCK PASSWORD command resets this bit to 1, enabling programing or erasing the nonvolatile lock bits. After the bits are changed, the WRITE GLOBAL FREEZE BIT command sets this bit to 0, protecting the nonvolatile lock bits from PROGRAM or ERASE operations.

# Nonvolatile and Volatile Sector Lock Bits Security

**Table 17: Nonvolatile and Volatile Lock Bits**

| Bit Details | Nonvolatile Lock Bit | Volatile Lock Bit |
|---|---|---|
| Description | Each sector of memory has one corresponding nonvolatile lock bit | Each sector of memory has one corresponding volatile lock bit; this bit is the sector write lock bit described in the Volatile Lock Bit Register table. |
| Function | When set to 0, locks and protects its corresponding memory sector from PROGRAM or ERASE operations. Because this bit is nonvolatile, the sector remains locked, protection enabled, until the bit is cleared to 1. | When set to 1, locks and protects its corresponding memory sector from PROGRAM or ERASE operations. Because this bit is volatile, protection is temporary. The sector is unlocked, protection disabled, upon device reset or power-down. |
| Settings | 1 = Lock disabled<br>0 = Lock enabled | 0 = Lock disabled<br>1 = Lock enabled |
| Enabling protection | The bit is set to 0 by the WRITE NONVOLATILE LOCK BITS command, enabling protection for designated locked sectors. Programming a sector lock bit requires the typical byte programming time. | The bit is set to 1 by the WRITE VOLATILE LOCK BITS command, enabling protection for designated locked sectors. |
| Disabling protection | All bits are cleared to 1 by the ERASE NONVOLATILE LOCK BITS command, unlocking and disabling protection for all sectors simultaneously. Erasing all sector lock bits requires typical sector erase time. | All bits are set to 0 upon reset or power-down, unlocking and disabling protection for all sectors. |
| Reading the bit | Bits are read by the READ NONVOLATILE LOCK BITS command. | Bits are read by the READ VOLATILE LOCK BITS command. |

# Volatile Lock Bit Security Register

One volatile lock bit register is associated with each sector of memory. It enables the sector to be locked, unlocked, or locked-down with the WRITE VOLATILE LOCK BITS command, which executes only when sector lock down (bit 1) is set to 0. Each register can be read with the READ VOLATILE LOCK BITS command. This register is compatible with and provides the same locking capability as the lock register in the Micron N25Q SPI NOR family.

**Table 18: Volatile Lock Bit Register**

| Bit | Name | Settings | Description |
|---|---|---|---|
| 7:2 | Reserved | 0 | Bit values are 0. |
| 1 | Sector lock down | 0 = Lock-down disabled (Default)<br>1 = Lock-down enabled | **Volatile bit:** Device always powers up with this bit set to 0 so that sector lock down and sector write lock bits can be set to 1. When this bit set to 1, neither of the two volatile lock bits can be written to until the next power cycle, hardware, or software reset. |
| 0 | Sector write lock | 0 = Write lock disabled (Default)<br>1 = Write lock enabled | **Volatile bit:** Device always powers up with this bit set to 0 so that PROGRAM and ERASE operations in this sector can be executed and sector content modified. When this bit is set to 1, PROGRAM and ERASE operations in this sector are not executed. |

# Device ID Data

The device ID data shown in the tables here is read by the READ ID and MULTIPLE I/O READ ID operations.

**Table 19: Device ID Data**

| Byte# | Name | Content Value | Assigned By |
|---|---|---|---|
| **Manufacturer ID** (1 byte total) | | | |
| 1 | Manufacturer ID (1 byte) | 20h | JEDEC |
| **Device ID** (2 bytes total) | | | |
| 2 | Memory type (1 byte) | BAh = 3V | Manufacturer |
| | | BBh = 1.8V | |
| 3 | Memory capacity (1 byte) | 22h = 2Gb | |
| | | 21h = 1Gb | |
| | | 20h = 512Mb | |
| | | 19h = 256Mb | |
| | | 18h = 128Mb | |
| | | 17h = 64Mb | |
| **Unique ID** (17 bytes total) | | | |
| 4 | Indicates the number of remaining ID bytes (1 byte) | 10h | Factory |
| 5 | Extended device ID (1 byte) | See Extended Device ID table | |
| 6 | Device configuration information (1 byte) | 00h = Standard | |
| 7:20 | Customized factory data (14 bytes) | Unique ID code (UID) | |

**Table 20: Extended Device ID Data, First Byte**

| Bit 7 | Bit 6 | Bit 5[1] | Bit 4 | Bit 3 | Bit 2[2] | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Reserved | Device Generation<br>1 = 2nd generation | 1 = Alternate BP scheme<br>0 = Standard BP scheme | Reserved | HOLD#/RESET#:<br>0 = HOLD<br>1 = RESET | Additional HW RESET#:<br>1 = Available<br>0 = Not available | Sector size:<br>00 = Uniform 64KB | |

Notes:
1. For alternate BP scheme information, contact the factory.
2. Available for specific part numbers. See Part Number Ordering Information for details.

# Serial Flash Discovery Parameter Data

The serial Flash discovery parameter (SFDP) provides a standard, consistent method to describe serial Flash device functions and features using internal parameter tables. The parameter tables can be interrogated by host system software, enabling adjustments to accommodate divergent features from multiple vendors. The SFDP standard defines a common parameter table that describes important device characteristics and serial access methods used to read the parameter table data.

Micron's SFDP table information aligns with JEDEC-standard JESD216 for serial Flash discoverable parameters. The latest JEDEC standard includes revision 1.6. Beginning week 42 (2014), Micron's MT25Q production parts will include SFDP data that aligns with revision 1.6.

Refer to JEDEC-standard JESD216B for a complete overview of the SFDP table definition.

Data in the SFDP tables is read by the READ SERIAL FLASH DISCOVERY PARAMETER operation.

See Micron TN-25-06: Serial Flash Discovery Parameters for MT25Q Family for serial Flash discovery parameter data.

# Command Definitions

### Table 21: Command Set

Notes 1 and 2 apply to the entire table

| Command | Code | Command-Address-Data | | | Address Bytes | Dummy Clock Cycles | | |
|---|---|---|---|---|---|---|---|---|
| | | Extended SPI | Dual SPI | Quad SPI | | Extended SPI | Dual SPI | Quad SPI |
| **Software RESET Operations** | | | | | | | | |
| RESET ENABLE | 66h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| RESET MEMORY | 99h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **READ ID Operations** | | | | | | | | |
| READ ID | 9E/9Fh | 1-0-1 | | | 0 | 0 | | |
| MULTIPLE I/O READ ID | AFh | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ SERIAL FLASH DISCOVERY PARAMETER | 5Ah | 1-1-1 | 2-2-2 | 4-4-4 | 3 | 8 | 8 | 8 |
| **READ MEMORY Operations** | | | | | | | | |
| READ | 03h | 1-1-1 | | | 3(4) | 0 | | |
| FAST READ | 0Bh | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 8 | 8 | 10 |
| DUAL OUTPUT FAST READ | 3Bh | 1-1-2 | 2-2-2 | | 3(4) | 8 | 8 | |
| DUAL INPUT/OUTPUT FAST READ | BBh | 1-2-2 | 2-2-2 | | 3(4) | 8 | 8 | |
| QUAD OUTPUT FAST READ | 6Bh | 1-1-4 | | 4-4-4 | 3(4) | 8 | | 10 |
| QUAD INPUT/OUTPUT FAST READ | EBh | 1-4-4 | | 4-4-4 | 3(4) | 10 | | 10 |
| DTR FAST READ | 0Dh | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 6 | 6 | 8 |
| DTR DUAL OUTPUT FAST READ | 3Dh | 1-1-2 | 2-2-2 | | 3(4) | 6 | 6 | |
| DTR DUAL INPUT/OUTPUT FAST READ | BDh | 1-2-2 | 2-2-2 | | 3(4) | 6 | 6 | |
| DTR QUAD OUTPUT FAST READ | 6Dh | 1-1-4 | | 4-4-4 | 3(4) | 6 | | 8 |
| DTR QUAD INPUT/OUTPUT FAST READ | EDh | 1-4-4 | | 4-4-4 | 3(4) | 8 | | 8 |
| QUAD INPUT/OUTPUT WORD READ | E7h | 1-4-4 | | 4-4-4 | 3(4) | 4 | | 4 |
| **READ MEMORY Operations with 4-Byte Address** | | | | | | | | |
| 4-BYTE READ | 13h | 1-1-1 | | | 4 | 0 | | |
| 4-BYTE FAST READ | 0Ch | 1-1-1 | 2-2-2 | 4-4-4 | 4 | 8 | 8 | 10 |
| 4-BYTE DUAL OUTPUT FAST READ | 3Ch | 1-1-2 | 2-2-2 | | 4 | 8 | 8 | |

**Table 21: Command Set (Continued)**

Notes 1 and 2 apply to the entire table

| Command | Code | Command-Address-Data | | | Address Bytes | Dummy Clock Cycles | | |
|---|---|---|---|---|---|---|---|---|
| | | Extended SPI | Dual SPI | Quad SPI | | Extended SPI | Dual SPI | Quad SPI |
| 4-BYTE DUAL INPUT/OUTPUT FAST READ | BCh | 1-2-2 | 2-2-2 | | 4 | 8 | 8 | |
| 4-BYTE QUAD OUTPUT FAST READ | 6Ch | 1-1-4 | | 4-4-4 | 4 | 8 | | 10 |
| 4-BYTE QUAD INPUT/OUTPUT FAST READ | ECh | 1-4-4 | | 4-4-4 | 4 | 10 | | 10 |
| 4-BYTE DTR FAST READ | 0Eh | 1-1-1 | 2-2-2 | 4-4-4 | 4 | 6 | 6 | 8 |
| 4-BYTE DTR DUAL INPUT/OUTPUT FAST READ | BEh | 1-2-2 | 2-2-2 | | 4 | 6 | 6 | |
| 4-BYTE DTR QUAD INPUT/ OUTPUT FAST READ | EEh | 1-4-4 | | 4-4-4 | 4 | 8 | | 8 |
| **WRITE Operations** | | | | | | | | |
| WRITE ENABLE | 06h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| WRITE DISABLE | 04h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **READ REGISTER Operations** | | | | | | | | |
| READ STATUS REGISTER | 05h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ FLAG STATUS REGISTER | 70h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ NONVOLATILE CONFIGURATION REGISTER | B5h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ VOLATILE CONFIGURATION REGISTER | 85h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ ENHANCED VOLATILE CONFIGURATION REGISTER | 65h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ EXTENDED ADDRESS REGISTER | C8h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ GENERAL PURPOSE READ REGISTER | 96h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 8 | 8 | 8 |
| **WRITE REGISTER Operations** | | | | | | | | |
| WRITE STATUS REGISTER | 01h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| WRITE NONVOLATILE CONFIGURATION REGISTER | B1h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |

## Table 21: Command Set (Continued)

Notes 1 and 2 apply to the entire table

| Command | Code | Command-Address-Data | | | Address Bytes | Dummy Clock Cycles | | |
|---|---|---|---|---|---|---|---|---|
| | | Extended SPI | Dual SPI | Quad SPI | | Extended SPI | Dual SPI | Quad SPI |
| WRITE VOLATILE CONFIGURATION REGISTER | 81h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| WRITE ENHANCED VOLATILE CONFIGURATION REGISTER | 61h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| WRITE EXTENDED ADDRESS REGISTER | C5h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| **CLEAR FLAG STATUS REGISTER Operation** | | | | | | | | |
| CLEAR FLAG STATUS REGISTER | 50h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **PROGRAM Operations** | | | | | | | | |
| PAGE PROGRAM | 02h | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 0 | 0 | 0 |
| DUAL INPUT FAST PROGRAM | A2h | 1-1-2 | 2-2-2 | | 3(4) | 0 | 0 | |
| EXTENDED DUAL INPUT FAST PROGRAM | D2h | 1-2-2 | 2-2-2 | | 3(4) | 0 | 0 | |
| QUAD INPUT FAST PROGRAM | 32h | 1-1-4 | | 4-4-4 | 3(4) | 0 | | 0 |
| EXTENDED QUAD INPUT FAST PROGRAM | 38h | 1-4-4 | | 4-4-4 | 3(4) | 0 | | 0 |
| **PROGRAM Operations with 4-Byte Address** | | | | | | | | |
| 4-BYTE PAGE PROGRAM | 12h | 1-1-1 | 2-2-2 | 4-4-4 | 4 | 0 | 0 | 0 |
| 4-BYTE QUAD INPUT FAST PROGRAM | 34h | 1-1-4 | | 4-4-4 | 4 | 0 | | 0 |
| 4-BYTE QUAD INPUT EXTENDED FAST PROGRAM | 3Eh | 1-4-4 | | 4-4-4 | 4 | 0 | | 0 |
| **ERASE Operations** | | | | | | | | |
| 32KB SUBSECTOR ERASE | 52h | 1-1-0 | 2-2-0 | 4-4-0 | 3(4) | 0 | 0 | 0 |
| 4KB SUBSECTOR ERASE | 20h | 1-1-0 | 2-2-0 | 4-4-0 | 3(4) | 0 | 0 | 0 |
| SECTOR ERASE | D8h | 1-1-0 | 2-2-0 | 4-4-0 | 3(4) | 0 | 0 | 0 |
| DIE ERASE | C4h | 1-1-0 | 2-2-0 | 4-4-0 | 3(4) | 0 | 0 | 0 |
| **ERASE Operations with 4-Byte Address** | | | | | | | | |
| 4-BYTE SECTOR ERASE | DCh | 1-1-0 | 2-2-0 | 4-4-0 | 4 | 0 | 0 | 0 |
| 4-BYTE 4KB SUBSECTOR ERASE | 21h | 1-1-0 | 2-2-0 | 4-4-0 | 4 | 0 | 0 | 0 |
| 4-BYTE 32KB SUBSECTOR ERASE | 5Ch | 1-1-0 | 2-2-0 | 4-4-0 | 4 | 0 | 0 | 0 |

## Table 21: Command Set (Continued)

Notes 1 and 2 apply to the entire table

| Command | Code | Command-Address-Data | | | Address Bytes | Dummy Clock Cycles | | |
|---|---|---|---|---|---|---|---|---|
| | | Extended SPI | Dual SPI | Quad SPI | | Extended SPI | Dual SPI | Quad SPI |
| **SUSPEND/RESUME Operations** | | | | | | | | |
| PROGRAM/ERASE SUSPEND | 75h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| PROGRAM/ERASE RESUME | 7Ah | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **ONE-TIME PROGRAMMABLE (OTP) Operations** | | | | | | | | |
| READ OTP ARRAY | 4Bh | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 8 | 8 | 10 |
| PROGRAM OTP ARRAY | 42h | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 0 | 0 | 0 |
| **4-BYTE ADDRESS MODE Operations** | | | | | | | | |
| ENTER 4-BYTE ADDRESS MODE | B7h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| EXIT 4-BYTE ADDRESS MODE | E9h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **Deep Power-Down Operations** | | | | | | | | |
| ENTER DEEP POWER DOWN | B9h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| RELEASE FROM DEEP POWER-DOWN | ABh | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **QUAD PROTOCOL Operations** | | | | | | | | |
| ENTER QUAD INPUT/OUTPUT MODE | 35h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| RESET QUAD INPUT/OUTPUT MODE | F5h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| **ADVANCED SECTOR PROTECTION Operations** | | | | | | | | |
| READ SECTOR PROTECTION | 2Dh | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| PROGRAM SECTOR PROTECTION | 2Ch | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| READ VOLATILE LOCK BITS | E8h | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 0 | 0 | 0 |
| WRITE VOLATILE LOCK BITS | E5h | 1-1-1 | 2-2-2 | 4-4-4 | 3(4) | 0 | 0 | 0 |
| READ NONVOLATILE LOCK BITS | E2h | 1-1-1 | 2-2-2 | 4-4-4 | 4 | 0 | 0 | 0 |
| WRITE NONVOLATILE LOCK BITS | E3h | 1-1-0 | 2-2-0 | 4-4-0 | 4 | 0 | 0 | 0 |
| ERASE NONVOLATILE LOCK BITS | E4h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| READ GLOBAL FREEZE BIT | A7h | 1-0-1 | | | 0 | 0 | 0 | 0 |
| WRITE GLOBAL FREEZE BIT | A6h | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| READ PASSWORD | 27h | 1-0-1 | | | 0 | 0 | 0 | 0 |
| WRITE PASSWORD | 28h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |

**Table 21: Command Set (Continued)**

Notes 1 and 2 apply to the entire table

| Command | Code | Command-Address-Data | | | Address Bytes | Dummy Clock Cycles | | |
|---|---|---|---|---|---|---|---|---|
| | | Extended SPI | Dual SPI | Quad SPI | | Extended SPI | Dual SPI | Quad SPI |
| UNLOCK PASSWORD | 29h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |
| **ADVANCED SECTOR PROTECTION Operations with 4-Byte Address** | | | | | | | | |
| 4-BYTE READ VOLATILE LOCK BITS | E0h | 1-1-1 | 2-2-2 | 4-4-4 | 4 | 0 | 0 | 0 |
| 4-BYTE WRITE VOLATILE LOCK BITS | E1h | 1-1-1 | 2-2-2 | 4-4-4 | 4 | 0 | 0 | 0 |
| **ADVANCED FUNCTION INTERFACE Operations** | | | | | | | | |
| INTERFACE ACTIVATION | 9Bh | 1-0-0 | 2-0-0 | 4-0-0 | 0 | 0 | 0 | 0 |
| CYCLIC REDUNDANCY CHECK | 9Bh/27h | 1-0-1 | 2-0-2 | 4-0-4 | 0 | 0 | 0 | 0 |

Notes:
1. Micron extended SPI protocol is the standard SPI protocol with additional commands that extend functionality and enable address or data transmission on multiple DQn lines.
2. The command code is always transmitted on DQn = 1, 2, or 4 lines according to the standard, dual, or quad protocol respectively. However, a command may be able to transmit address and data on multiple DQn lines regardless of protocol. The protocol columns show the number of DQn lines a command uses to transmit command, address, and data information as shown in these examples: command-address-data = 1-1-1, or 1-2-2, or 2-4-4, and so on.
3. The READ SERIAL FLASH DISCOVERY PARAMETER operation accepts only 3-byte address even if the device is configured to 4-byte address mode.
4. Requires 4 bytes of address if the device is configured to 4-byte address mode.
5. The number of dummy clock cycles required when shipped from Micron factories. The user can modify the dummy clock cycle number via the nonvolatile configuration register and the volatile configuration register.
6. The number of dummy cycles for the READ GENERAL PURPOSE READ REGISTER command is fixed (8 dummy cycles) and is not affected by dummy cycle settings in the nonvolatile configuration register and volatile configuration register.
7. The general purpose read register is 64 bytes. After the first 64 bytes, the device outputs 00h and does not wrap.
8. The WRITE ENABLE command must be issued first before this operation can be executed.
9. Formerly referred to as the READ LOCK REGISTER operation.
10. Formerly referred to as the WRITE LOCK REGISTER operation.

# Software RESET Operations

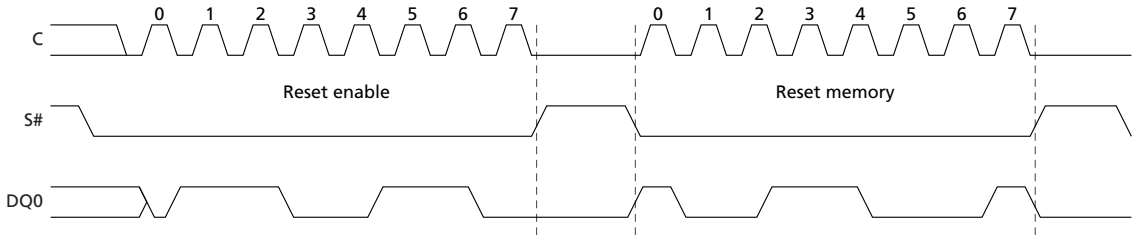
## RESET ENABLE and RESET MEMORY Commands

To initiate these commands, S# is driven LOW and the command code is input on DQn. A minimum de-selection time of $^{t}$SHSL2 must come between RESET ENABLE and RESET MEMORY or reset is not guaranteed. Then, S# must be driven HIGH for the device to enter power-on reset. A time of $^{t}$SHSL3 is required before the device can be re-selected by driving S# LOW.

**Table 22: RESET ENABLE and RESET MEMORY Operations**

| Operation Name | Description/Conditions |
|---|---|
| RESET ENABLE (66h) | To reset the device, the RESET ENABLE command must be followed by the RESET MEMORY command. When the two commands are executed, the device enters a power-on reset condition. It is recommended to exit XIP mode before executing these two commands.<br>All volatile lock bits, the volatile configuration register, the enhanced volatile configuration register, and the extended address register are reset to the power-on reset default condition according to nonvolatile configuration register settings.<br>If a reset is initiated while a WRITE, PROGRAM, or ERASE operation is in progress or suspended, the operation is aborted and data may be corrupted.<br>Reset is effective after the flag status register bit 7 outputs 1 with at least one byte output.<br>A RESET ENABLE command is not accepted during WRITE STATUS REGISTER and WRITE NONVOLATILE CONFIGURATION REGISTER operations. |
| RESET MEMORY (99h) | |

**Figure 12: RESET ENABLE and RESET MEMORY Command**




Note: 1. Above timing diagram is showed for Extended-SPI Protocol case, however these commands are available in all protocols. In DIO-SPI protocol, the instruction bits are transmitted on both DQ0 and DQ1 pins. In QIO-SPI protocol the instruction bits are transmitted on all four data pins. In Extended-DTR-SPI protocol, the instruction bits are transmitted on DQ0 pin in double transfer rate mode. In DIO-DTR-SPI protocol, the instruction bits are transmitted on both DQ0 and DQ1 pins in double transfer rate mode. In QIO-DTR-SPI protocol, the instruction bits are transmitted on all four data pins in double transfer rate mode.

# READ ID Operations

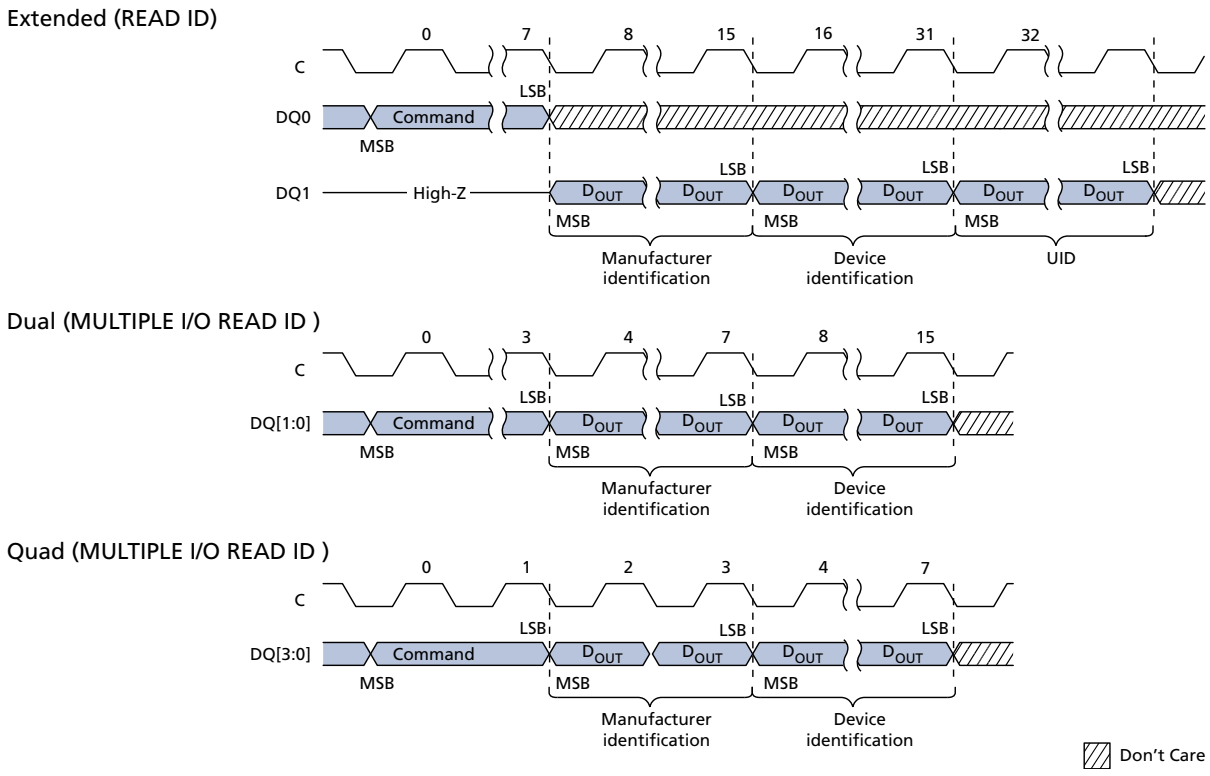
## READ ID and MULTIPLE I/O READ ID Commands

To initiate these commands, S# is driven LOW and the command code is input on DQ*n*. When S# is driven HIGH, the device goes to standby. The operation is terminated by driving S# HIGH at any time during data output.

### Table 23: READ ID and MULTIPLE I/O READ ID Operations

| Operation Name | Description/Conditions |
|---|---|
| READ ID (9Eh/9Fh) | Outputs information shown in the Device ID Data tables. If an ERASE or PROGRAM cycle is in progress when the command is initiated, the command is not decoded and the command cycle in progress is not affected. |
| MULTIPLE I/O READ ID (AFh) | |

### Figure 13: READ ID and MULTIPLE I/O READ ID Commands




Note: 1. S# not shown.

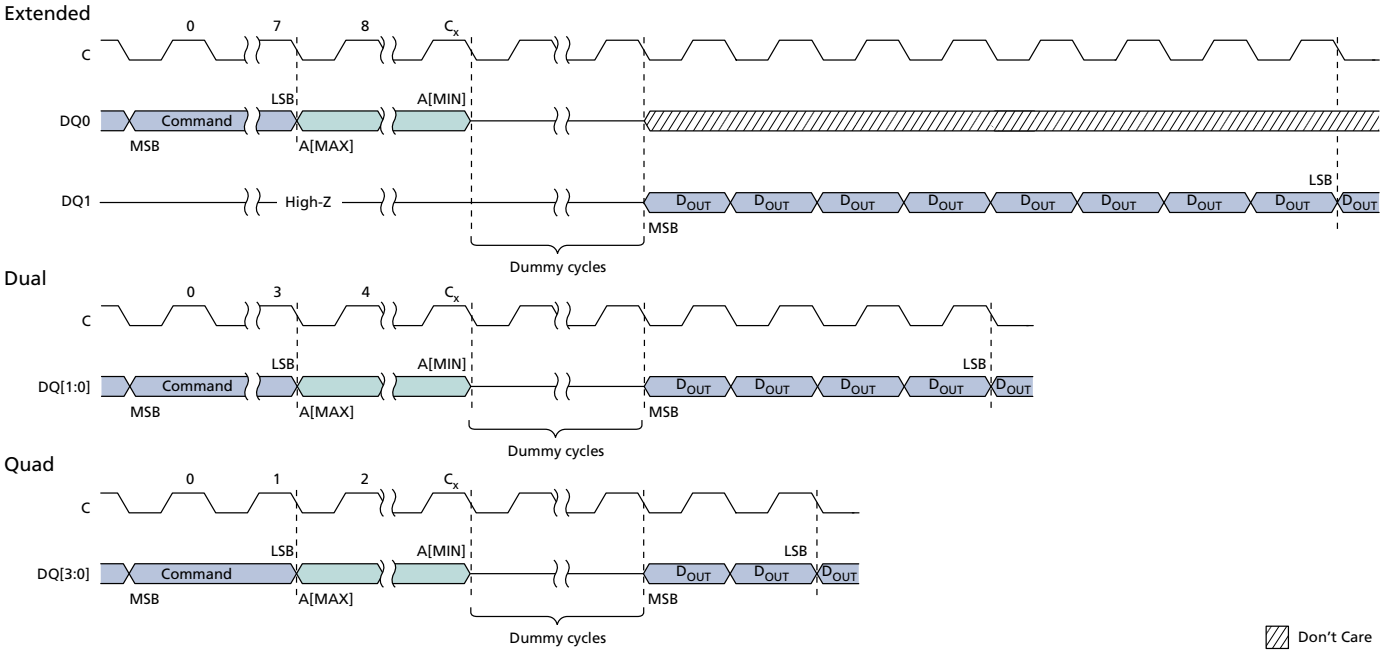
# READ SERIAL FLASH DISCOVERY PARAMETER Operation

## READ SERIAL FLASH DISCOVERY PARAMETER Command

To execute READ SERIAL FLASH DISCOVERY PARAMETER command, S# is driven LOW. The command code is input on DQ0, followed by three address bytes and eight dummy clock cycles (address is always 3 bytes, even if the device is configured to work in 4-byte address mode). The device outputs the information starting from the specified address. When the 2048-byte boundary is reached, the data output wraps to address 0 of the serial Flash discovery parameter table. The operation is terminated by driving S# HIGH at any time during data output.

**Note:** The operation always executes in continuous mode so the read burst wrap setting in the volatile configuration register does not apply.

**Figure 14: READ SERIAL FLASH DISCOVERY PARAMETER Command – 5Ah**




Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1); For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.

# READ MEMORY Operations

To initiate a command, S# is driven LOW and the command code is input on DQ*n*, followed by input of the address bytes on DQ*n*. The operation is terminated by driving S# HIGH at any time during data output.

**Table 24: READ MEMORY Operations**

| Operation Name | Description/Conditions |
|---|---|
| READ (03h) | The device supports 3-byte addressing (default), with A[23:0] input during address cycle. After any READ command is executed, the device will output data from the selected address. After the boundary is reached, the device will start reading again from the beginning.<br>Each address bit is latched in during the rising edge of the clock. The addressed byte can be at any location, and the address automatically increments to the next address after each byte of data is shifted out; therefore, a die can be read with a single command.<br>FAST READ can operate at a higher frequency ($^{f}C$).<br>DTR commands function in DTR protocol regardless of settings in the nonvolatile configuration register or enhanced volatile configuration register; other commands function in DTR protocol only after DTR protocol is enabled by the register settings.<br>E7h is similar to the QUAD I/O FAST READ command except that the lowest address bit (A0) must equal 0 and only four dummy clocks are required prior to the data output. This command is supported in extended-SPI and quad-SPI protocols, but not in the DTR protocol; it is ignored it in dual-SPI protocol. |
| FAST READ (0Bh) | |
| DUAL OUTPUT FAST READ (3Bh) | |
| DUAL INPUT/OUTPUT FAST READ (BBh) | |
| QUAD OUTPUT FAST READ (6Bh) | |
| QUAD INPUT/OUTPUT FAST READ (EBh) | |
| DTR FAST READ (0Dh) | |
| DTR DUAL OUTPUT FAST READ (3Dh) | |
| DTR DUAL INPUT/OUTPUT FAST READ (BDh) | |
| DTR QUAD OUTPUT FAST READ (6Dh) | |
| DTR QUAD INPUT/OUTPUT FAST READ (EDh) | |
| QUAD INPUT/OUTPUT WORD READ (E7h) | |

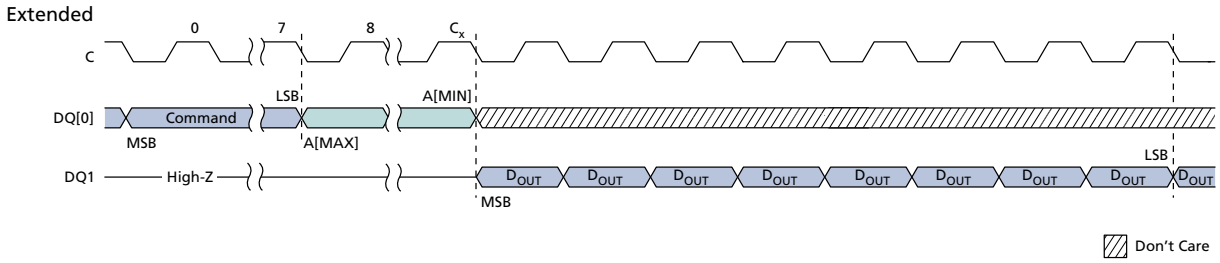
# 4-BYTE READ MEMORY Operations

**Table 25: 4-BYTE READ MEMORY Operations**

| Operation Name | Description/Conditions |
|---|---|
| 4-BYTE READ (13h) | READ MEMORY operations can be extended to a 4-byte address range, with [A31:0] input during address cycle.<br>Selection of the 3-byte or 4-byte address range can be enabled in two ways: through the nonvolatile configuration register or through the ENABLE 4-BYTE ADDRESS MODE/EXIT 4-BYTE ADDRESS MODE commands.<br>Each address bit is latched in during the rising edge of the clock. The addressed byte can be at any location, and the address automatically increments to the next address after each byte of data is shifted out; therefore, a die can be read with a single command.<br>FAST READ can operate at a higher frequency ($^{f}C$).<br>4-BYTE commands and DTR 4-BYTE commands function in 4-BYTE and DTR 4-BYTE protocols regardless of settings in the nonvolatile configuration register or enhanced volatile configuration register; other commands function in 4-BYTE and DTR protocols only after the specific protocol is enabled by the register settings. |
| 4-BYTE FAST READ (0Ch) | |
| 4-BYTE DUAL OUTPUT FAST READ (3Ch) | |
| 4-BYTE DUAL INPUT/OUTPUT FAST READ (BCh) | |
| 4-BYTE QUAD OUTPUT FAST READ (6Ch) | |
| 4-BYTE QUAD INPUT/OUTPUT FAST READ (ECh) | |
| DTR 4-BYTE FAST READ (0Eh) | |
| DTR 4-BYTE DUAL INPUT/OUTPUT FAST READ (BEh) | |
| DTR 4-BYTE QUAD INPUT/OUTPUT FAST READ (EEh) | |

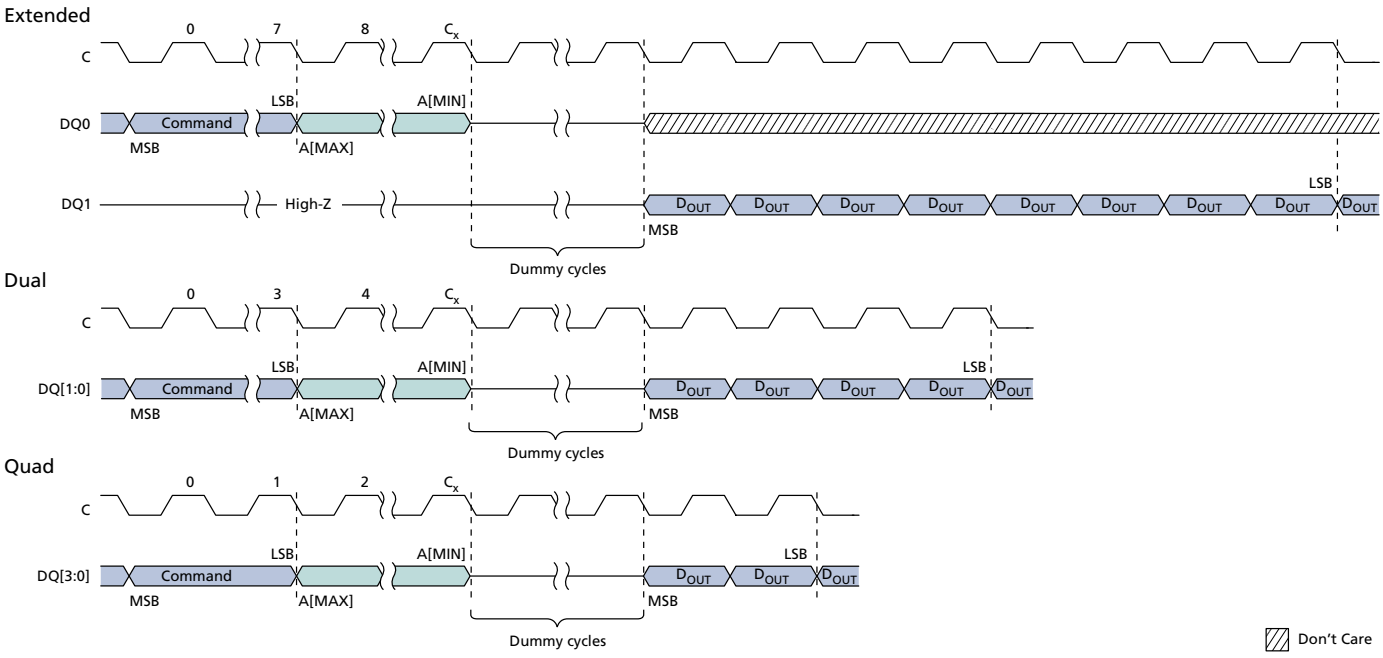
# READ MEMORY Operations Timings

### Figure 15: READ – 03h/13h[3]



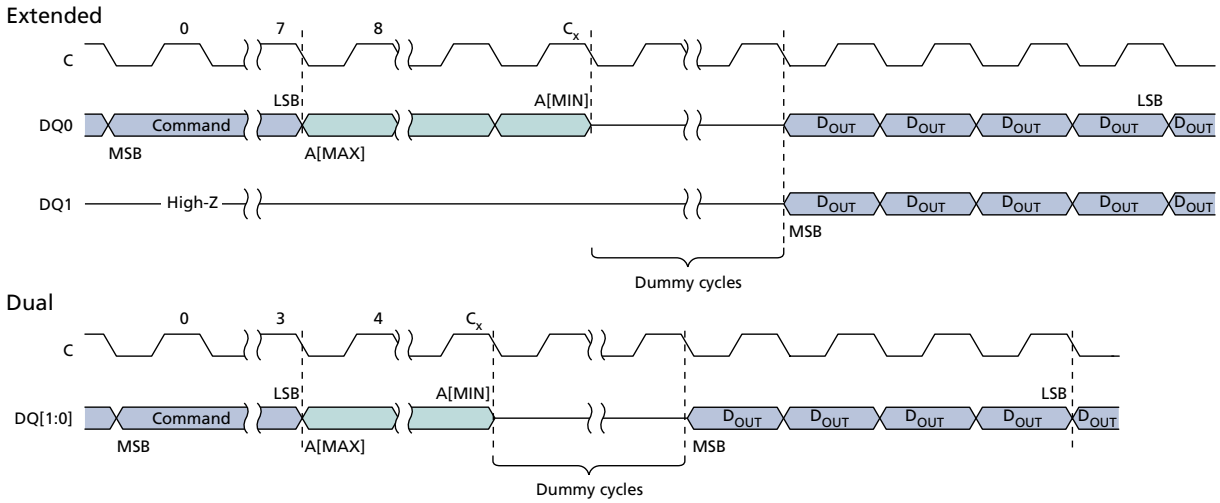

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1); For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.
3. READ and 4-BYTE READ commands.

### Figure 16: FAST READ – 0Bh/0Ch[3]



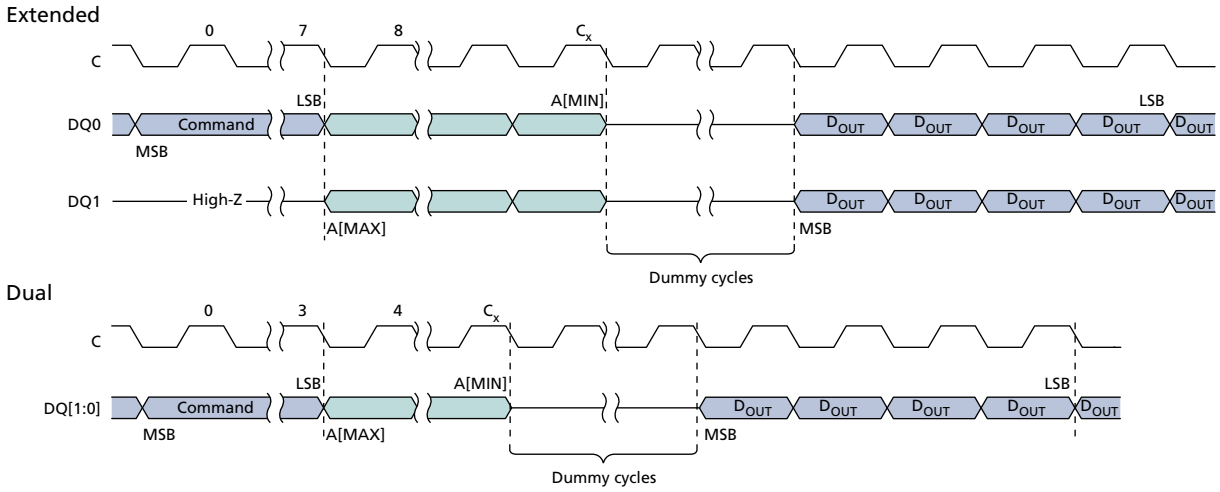

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1); For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.
3. FAST READ and 4-BYTE FAST READ commands.

**Figure 17: DUAL OUTPUT FAST READ – 3Bh/3Ch[3]**



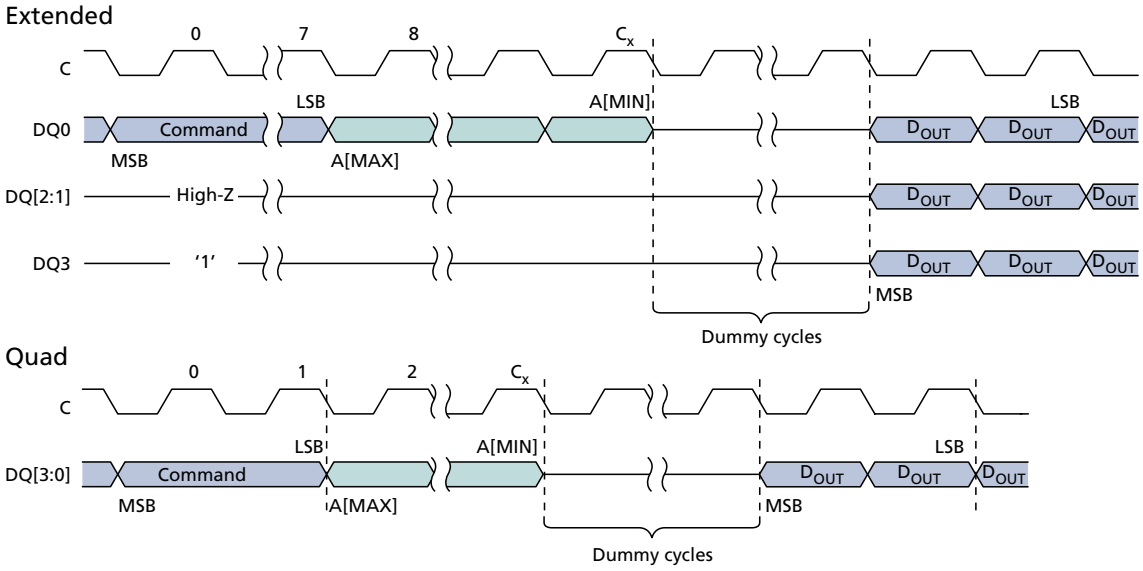

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1); For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/2.
2. S# not shown.
3. DUAL OUTPUT FAST READ and 4-BYTE DUAL OUTPUT FAST READ commands.

**Figure 18: DUAL INPUT/OUTPUT FAST READ – BBh/BCh[3]**



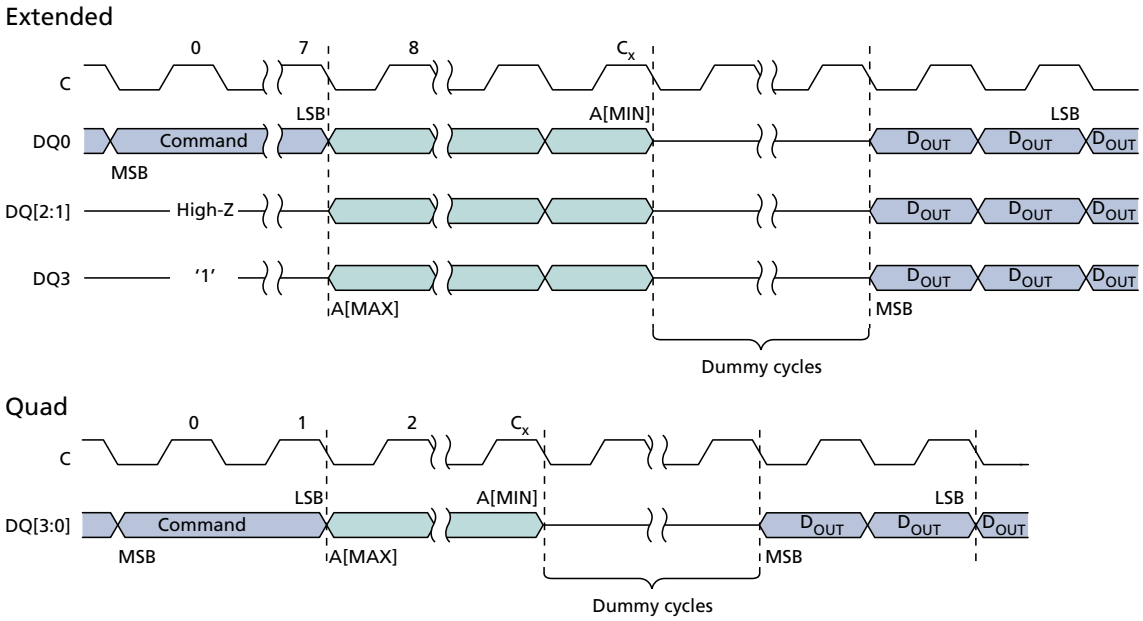

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/2; For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/2.
2. S# not shown.
3. DUAL INPUT/OUTPUT FAST READ and 4-BYTE DUAL INPUT/OUTPUT FAST READ commands.

**Figure 19: QUAD OUTPUT FAST READ – 6Bh/6Ch[3]**




Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1); For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.
3. QUAD OUTPUT FAST READ and 4-BYTE QUAD OUTPUT FAST READ commands.

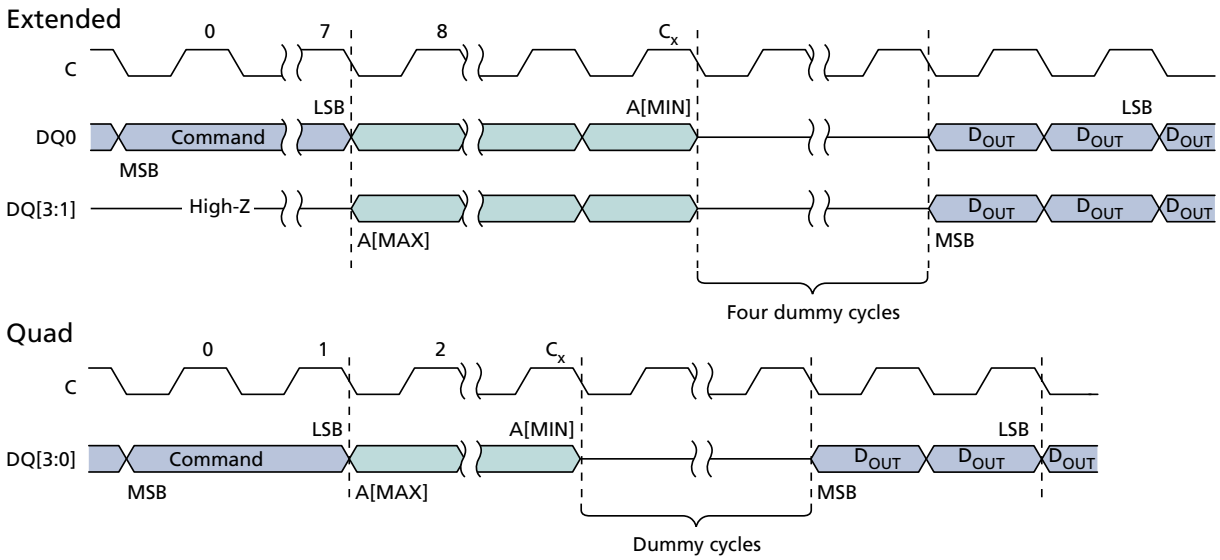
**Figure 20: QUAD INPUT/OUTPUT FAST READ – EBh/ECh[3]**




Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/4; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.

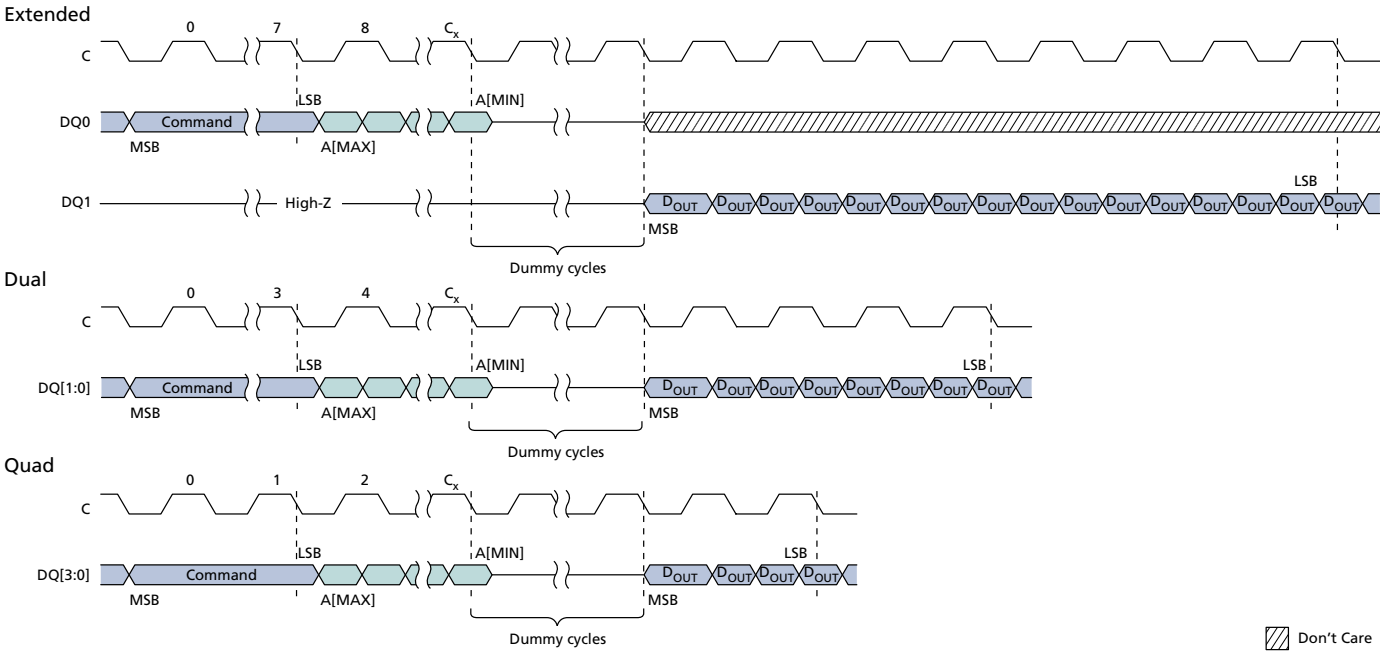
3. QUAD INPUT/OUTPUT FAST READ and 4-BYTE QUAD INPUT/OUTPUT FAST READ commands.

**Figure 21: QUAD INPUT/OUTPUT WORD READ – E7h[3]**



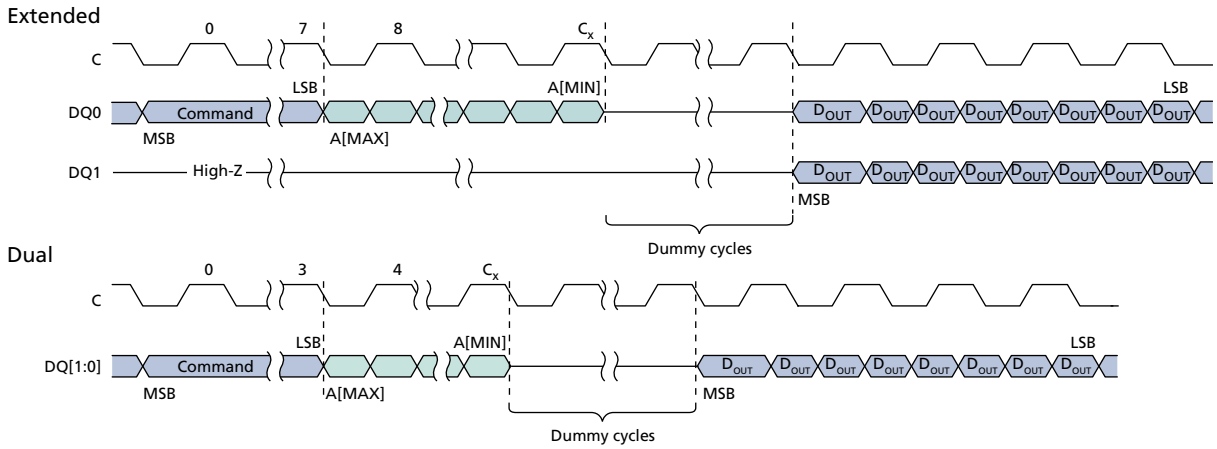

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/4; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.
3. QUAD INPUT/OUTPUT WORD READ and 4-BYTE QUAD INPUT/OUTPUT WORD READ commands.

## Figure 22: DTR FAST READ – 0Dh/0Eh[3]



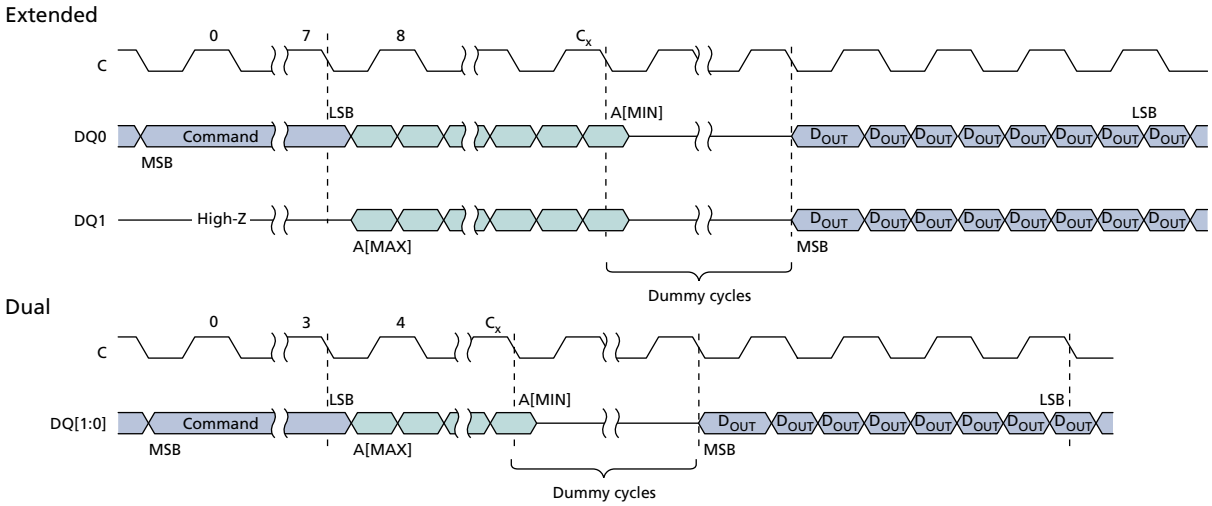

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/2; For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/4; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/8.
2. S# not shown.
3. DTR FAST READ and 4-BYTE DTR FAST READ commands.

## Figure 23: DTR DUAL OUTPUT FAST READ – 3Dh[3]



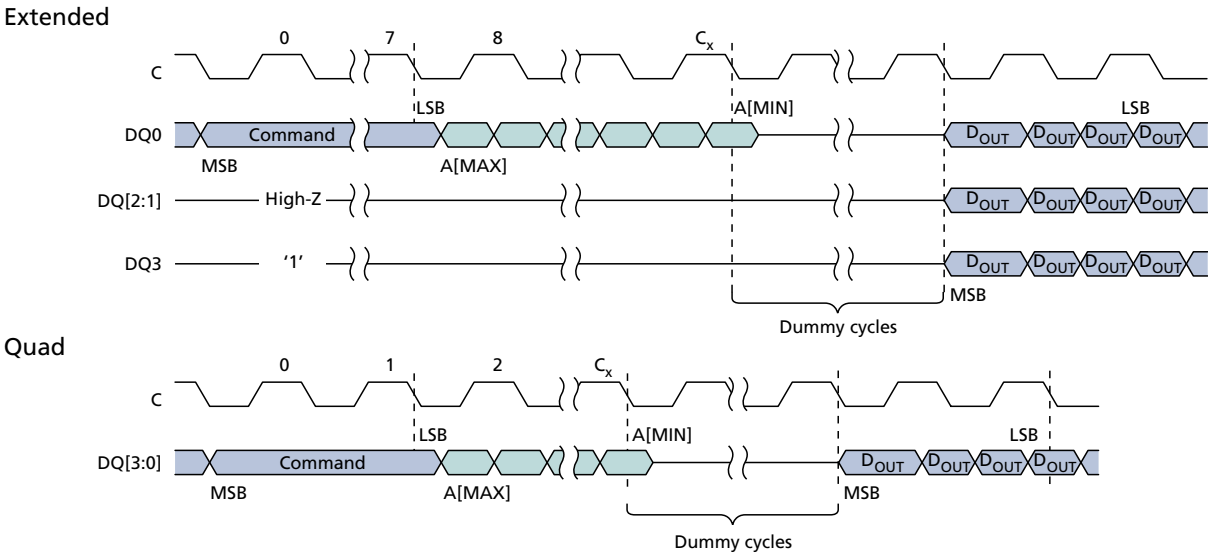

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/2; For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/4.
2. S# not shown.
3. DTR DUAL OUTPUT FAST READ and 4-BYTE DTR DUAL OUTPUT FAST READ commands.

**Figure 24: DTR DUAL INPUT/OUTPUT FAST READ – BDh[3]**



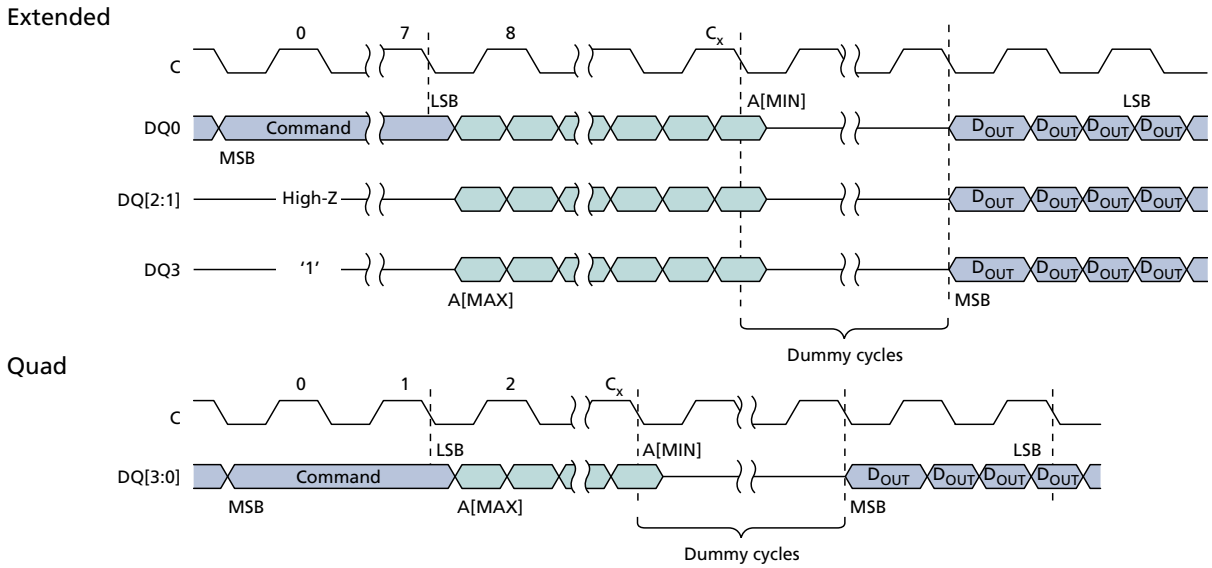

Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/4; For dual protocol, $C_x$ = 3 + (A[MAX] + 1)/8.
2. S# not shown.
3. DTR DUAL INPUT/OUTPUT FAST READ and 4-BYTE DTR DUAL INPUT/OUTPUT FAST READ commands.

**Figure 25: DTR QUAD OUTPUT FAST READ – 6Dh[3]**




Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/2; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/8.
2. S# not shown.
3. DTR QUAD OUTPUT FAST READ and 4-BYTE DTR QUAD OUTPUT FAST READ commands.

**Figure 26: DTR QUAD INPUT/OUTPUT FAST READ – EDh[3]**




Notes:
1. For extended protocol, $C_x$ = 7 + (A[MAX] + 1)/8; For quad protocol, $C_x$ = 1 + (A[MAX] + 1)/8.
2. S# not shown.
3. DTR QUAD INPUT/OUTPUT FAST READ and 4-BYTE DTR QUAD INPUT/OUTPUT FAST READ commands.

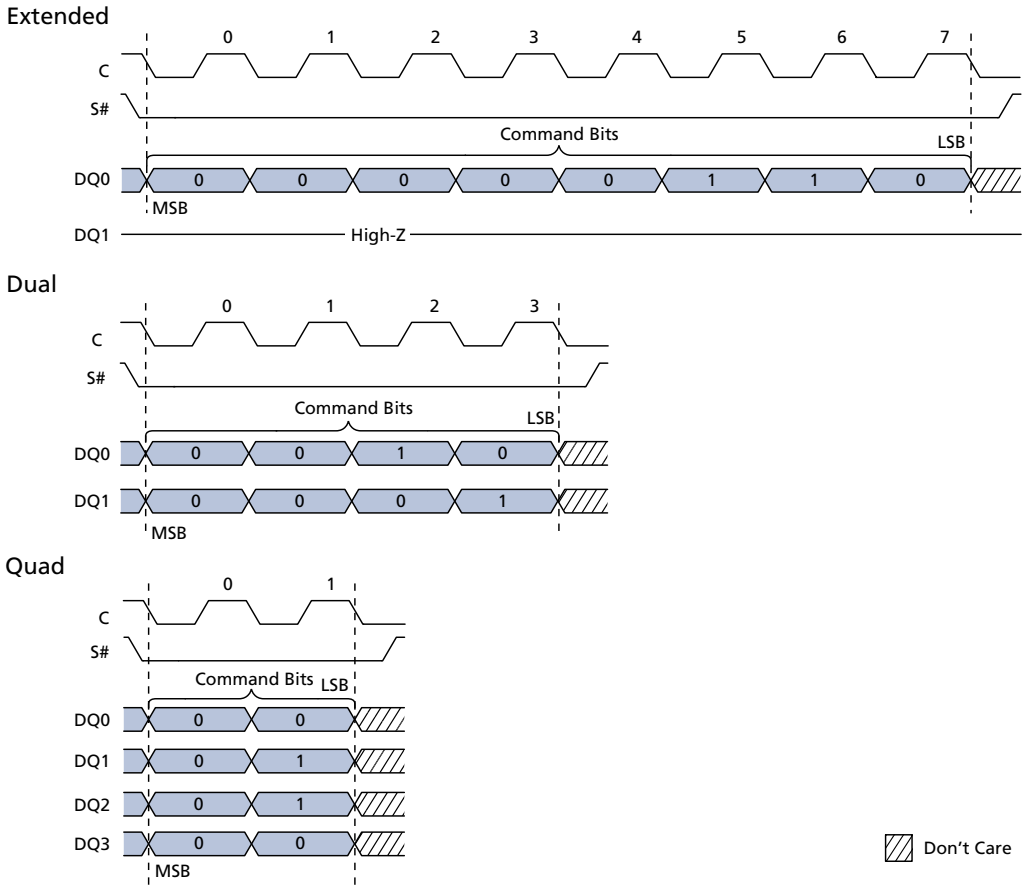
# WRITE ENABLE/DISABLE Operations

To initiate a command, S# is driven LOW and held LOW until the eighth bit of the command code has been latched in, after which it must be driven HIGH. For extended-, dual-, and quad-SPI protocols respectively, the command code is input on DQ0, DQ[1:0], and DQ[3:0]. If S# is not driven HIGH after the command code has been latched in, the command is not executed, flag status register error bits are not set, and the write enable latch remains cleared to its default setting of 0, providing protection against errant data modification.

**Table 26: WRITE ENABLE/DISABLE Operations**

| Operation Name | Description/Conditions |
|---|---|
| WRITE ENABLE (06h) | Sets the write enable latch bit before each PROGRAM, ERASE, and WRITE command. |
| WRITE DISABLE (04h) | Clears the write enable latch bit. In case of a protection error, WRITE DISABLE will not clear the bit. Instead, a CLEAR FLAG STATUS REGISTER command must be issued to clear both flags. |

**Figure 27: WRITE ENABLE and WRITE DISABLE Timing**




Note: 1. WRITE ENABLE command sequence and code, shown here, is 06h (0000 0110 binary); WRITE DISABLE is identical, but its command code is 04h (0000 0100 binary).

# READ REGISTER Operations

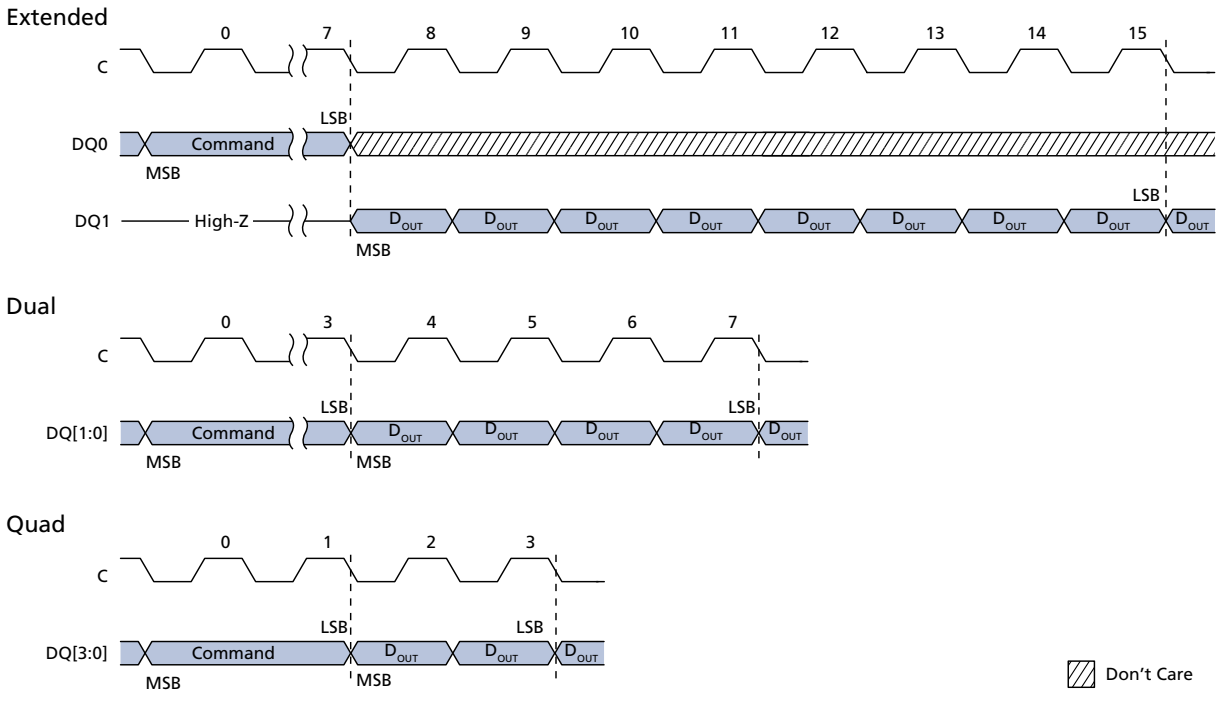
To initiate a command, S# is driven LOW. For extended SPI protocol, input is on DQ0, output on DQ1. For dual SPI protocol, input/output is on DQ[1:0] and for quad SPI protocol, input/output is on DQ[3:0]. The operation is terminated by driving S# HIGH at any time during data output.

**Table 27: READ REGISTER Operations**

| Operation Name | Description/Conditions | Note |
|---|---|---|
| READ STATUS REGISTER (05h) | Can be read continuously and at any time, including during a PROGRAM, ERASE, or WRITE operation. If one of these operations is in progress, checking the write in progress bit or P/E controller bit is recommended before executing the command. | |
| READ FLAG STATUS REGISTER (70h) | | |
| READ NONVOLATILE CONFIGURATION REGISTER (B5h) | Can be read continuously. After all 16 bits of the register have been read, a 0 is output. All reserved fields output a value of 1. | 1 |
| READ VOLATILE CONFIGURATION REGISTER (85h) | When the register is read continuously, the same byte is output repeatedly. | |
| READ ENHANCED VOLATILE CONFIGURATION REGISTER (65h) | | |
| READ EXTENDED ADDRESS REGISTER (C8h) | | |

Note: 1. The operation will have output data starting from the least significant byte.

**Figure 28: READ REGISTER Timing**




Notes: 1. Supports all READ REGISTER commands except DYNAMIC PROTECTION BITS READ.

2. A READ NONVOLATILE CONFIGURATION REGISTER operation will output data starting from the least significant byte.
3. S# not shown.

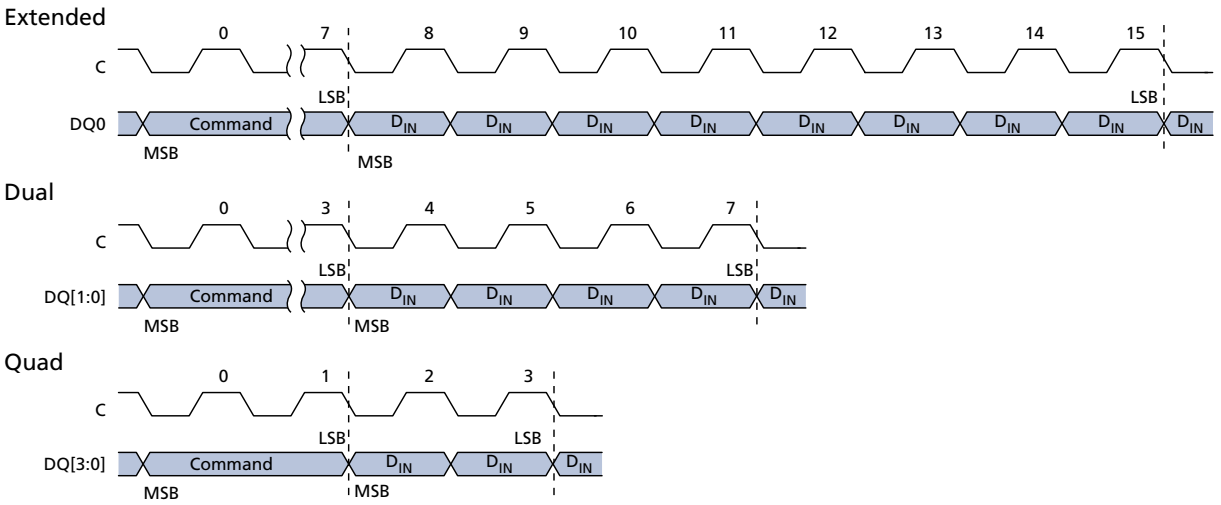
# WRITE REGISTER Operations

Before a WRITE REGISTER command is initiated, the WRITE ENABLE command must be executed to set the write enable latch bit to 1. To initiate a command, S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH; for the WRITE NONVOLATILE CONFIGURATION REGISTER command, S# is held LOW until the 16th bit of the last data byte has been latched in. For the extended, dual, and quad SPI protocols respectively, input is on DQ0, DQ[1:0], and DQ[3:0], followed by the data bytes. If S# is not driven HIGH, the command is not executed, flag status register error bits are not set, and the write enable latch remains set to 1. The operation is self-timed and its duration is $^{t}$W for WRITE STATUS REGISTER and $^{t}$NVCR for WRITE NONVOLATILE CONFIGURATION REGISTER.

**Table 28: WRITE REGISTER Operations**

| Operation Name | Description/Conditions |
|---|---|
| WRITE STATUS REGISTER (01h) | The WRITE STATUS REGISTER command writes new values to status register bits 7:2, enabling software data protection. The status register can also be combined with the W# signal to provide hardware data protection. This command has no effect on status register bits 1:0. |
| WRITE NONVOLATILE CONFIGURATION REGISTER (B1h) | For the WRITE STATUS REGISTER and WRITE NONVOLATILE CONFIGURATION REGISTER commands, when the operation is in progress, the write in progress bit is set to 1. The write enable latch bit is cleared to 0, whether the operation is successful or not. The status register and flag status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0, whether the operation is successful or not .It is possible to obtain the operation status by reading the flag status register a number of times corresponding to the die stacked, with S# toggled in between the READ FLAG STATUS REGISTER commands. When the operation completes, the program or erase controller bit of the flag status register is cleared to 1. The end of operation can be detected when the program or erase controller bit of the flag status register outputs 1 for all the die of the stack. **Note: The The WRITE NONVOLATILE CONFIGURATION REGISTER operation must have input data starting from the least significant byte** |
| WRITE VOLATILE CONFIGURATION REGISTER (81h) | Because register bits are volatile, change to the bits is immediate. Reserved bits are not affected by this command. |
| WRITE ENHANCED VOLATILE CONFIGURATION REGISTER (61h) | |
| WRITE EXTENDED ADDRESS REGISTER (C5h) | |

**Figure 29: WRITE REGISTER Timing**




Notes:
1. Supports all WRITE REGISTER commands except WRITE LOCK REGISTER.
2. Data is two bytes for a WRITE NONVOLATILE CONFIGURATION REGISTER operation, input starting from the least significant byte.
3. S# not shown.

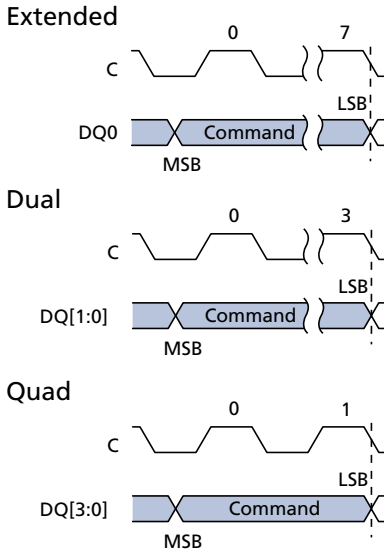
# CLEAR FLAG STATUS REGISTER Operation

To initiate a command, S# is driven LOW. For the extended-, dual-, and quad-SPI protocols respectively, input is on DQ0, DQ[1:0], and DQ[3:0]. The operation is terminated by driving S# HIGH at any time.

**Table 29: CLEAR FLAG STATUS REGISTER Operation**

| Operation Name | Description/Conditions |
|---|---|
| CLEAR FLAG STATUS REGISTER (50h) | Resets the error bits (erase, program, and protection) |

**Figure 30: CLEAR FLAG STATUS REGISTER Timing**




Note: 1. S# not shown.

# PROGRAM Operations

Before a PROGRAM command is initiated, the WRITE ENABLE command must be executed to set the write enable latch bit to 1. To initiate a command, S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. If S# is not driven HIGH, the command is not executed, flag status register error bits are not set, and the write enable latch remains set to 1. Each address bit is latched in during the rising edge of the clock. When a command is applied to a protected sector, the command is not executed, the write enable latch bit remains set to 1, and flag status register bits 1 and 4 are set. If the operation times out, the write enable latch bit is reset and the program fail bit is set to 1.

**Note:** The manner of latching data shown and explained in the timing diagrams ensures that the number of clock pulses is a multiple of one byte before command execution, helping reduce the effects of noisy or undesirable signals and enhancing device data protection.

**Table 30: PROGRAM Operations**

| Operation Name | Description/Conditions |
|---|---|
| PAGE PROGRAM (02h) | A PROGRAM operation changes a bit from 1 to 0.<br>When the operation is in progress, the write in progress bit is set to 1. The write enable latch bit is cleared to 0, whether the operation is successful or not. The status register and flag status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0. An operation can be paused or resumed by the PROGRAM/ERASE SUSPEND or PROGRAM/ERASE RESUME command, respectively.<br>If the bits of the least significant address, which is the starting address, are not all zero, all data transmitted beyond the end of the current page is programmed from the starting address of the same page. If the number of bytes sent to the device exceed the maximum page size, previously latched data is discarded and only the last maximum page-size number of data bytes are guaranteed to be programmed correctly within the same page. If the number of bytes sent to the device is less than the maximum page size, they are correctly programmed at the specified addresses without any effect on the other bytes of the same page. |
| DUAL INPUT FAST PROGRAM (A2h) | |
| EXTENDED DUAL INPUT FAST PROGRAM (D2h) | |
| QUAD INPUT FAST PROGRAM (32h) | |
| EXTENDED QUAD INPUT FAST PROGRAM (38h) | |

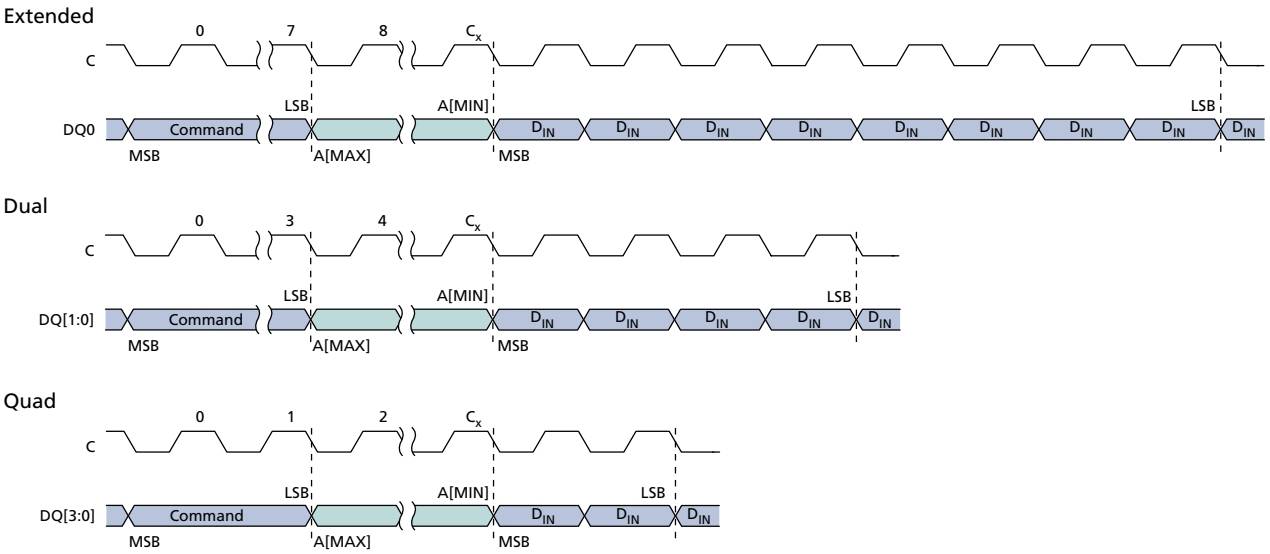
# 4-BYTE PROGRAM Operations

### Table 31: 4-BYTE PROGRAM Operations

| Operation Name | Description/Conditions |
|---|---|
| 4-BYTE PAGE PROGRAM (12h) | PROGRAM operations can be extended to a 4-byte address range, with [A31:0] input during address cycle.<br>Selection of the 3-byte or 4-byte address range can be enabled in two ways: through the nonvolatile configuration register or through the ENABLE 4-BYTE ADDRESS MODE/EXIT 4-BYTE ADDRESS MODE commands.<br>4-BYTE commands and DTR 4-BYTE commands function in 4-BYTE and DTR 4-BYTE protocol regardless of settings in the nonvolatile configuration register or enhanced volatile configuration register; other commands function in 4-BYTE and DTR protocols only after the specific protocol is enabled by the register settings. |
| 4-BYTE QUAD INPUT FAST PROGRAM (34h) | |
| 4-BYTE EXTENDED QUAD INPUT FAST PROGRAM (3Eh) | |

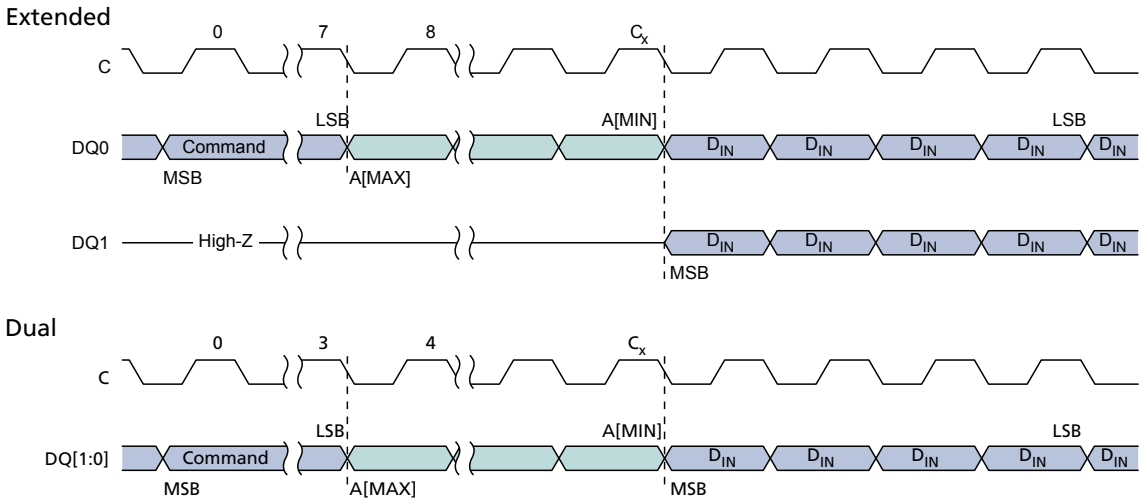
# PROGRAM Operations Timings

### Figure 31: PAGE PROGRAM Command



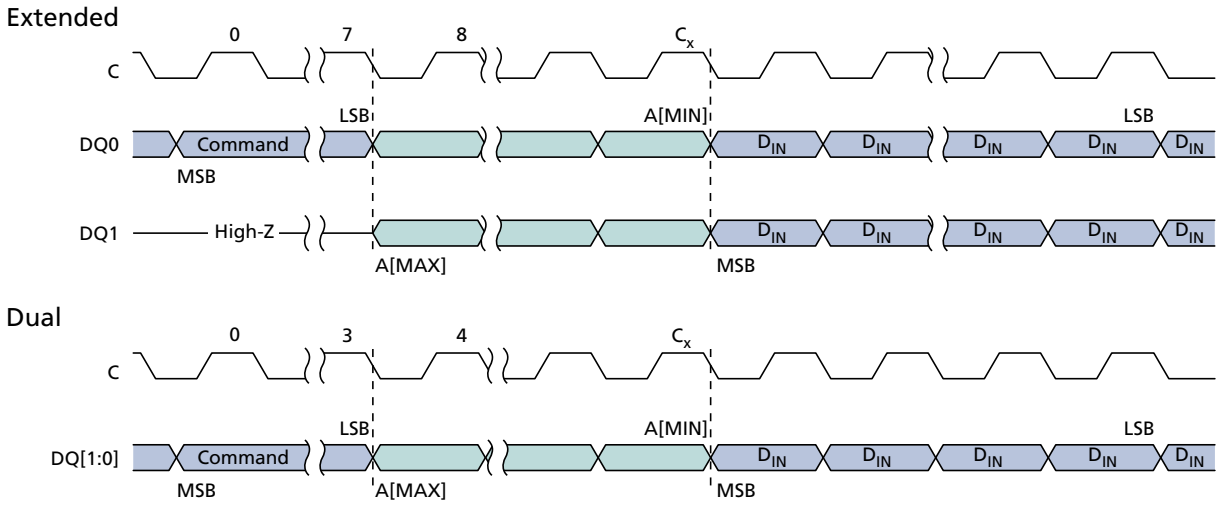

Notes:
1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1); For dual-SPI protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad-SPI protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown. The operation is self-timed, and its duration is $^{t}$PP.

**Figure 32: DUAL INPUT FAST PROGRAM Command**



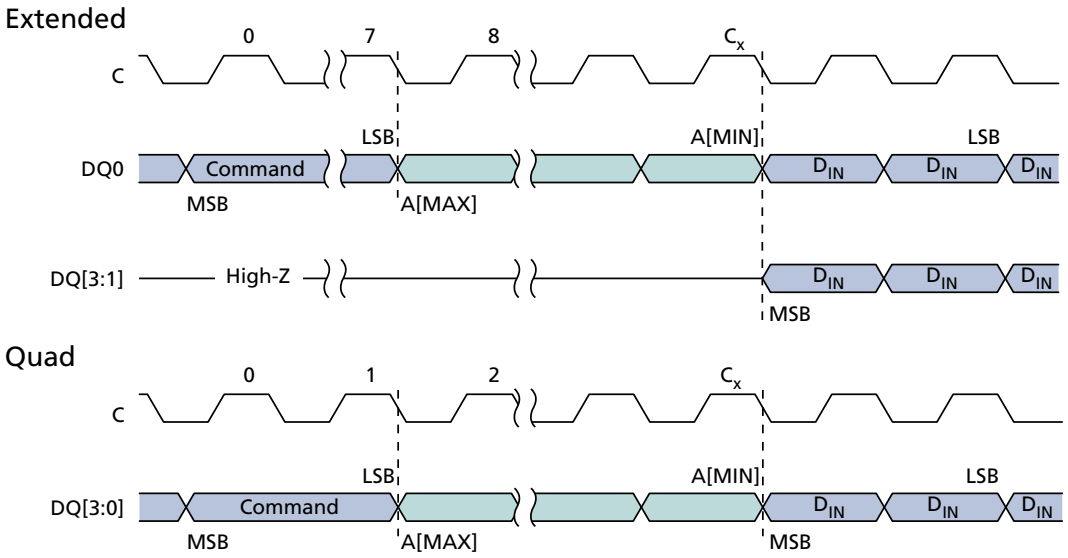

Notes:
1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1); For dual-SPI protocol, $C_x$ = 3 + (A[MAX] + 1)/2.
2. S# not shown.

**Figure 33: EXTENDED DUAL INPUT FAST PROGRAM Command**



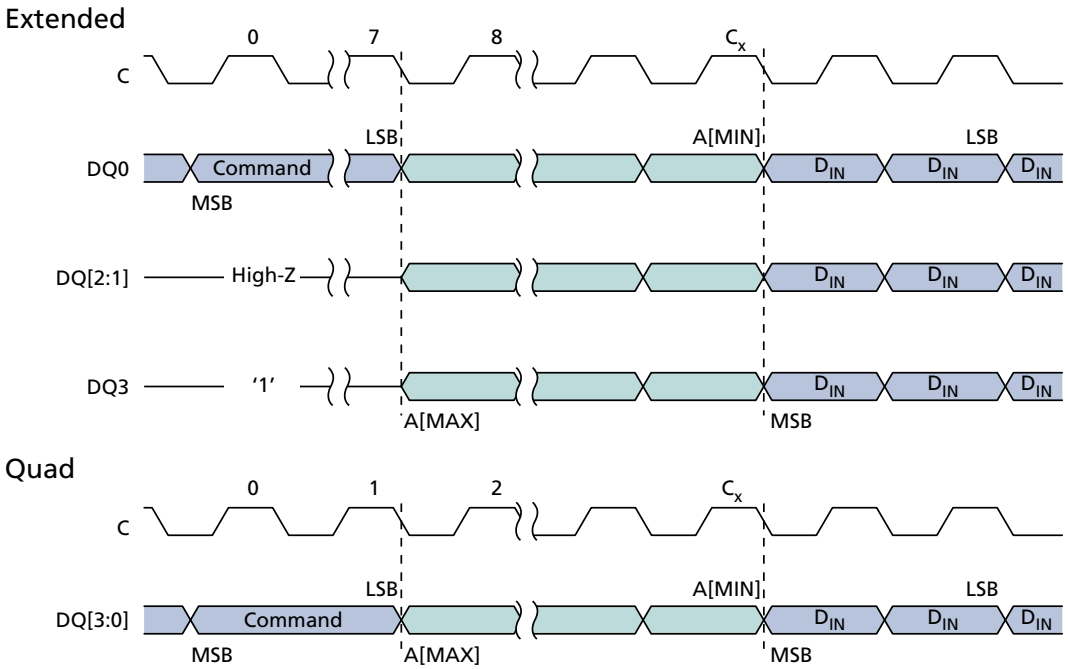

Notes:
1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1)/2; For dual-SPI protocol, $C_x$ = 3 + (A[MAX] + 1)/2.
2. S# not shown.

**Figure 34: QUAD INPUT FAST PROGRAM Command**




Notes:
1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1); For quad-SPI protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.

**Figure 35: EXTENDED QUAD INPUT FAST PROGRAM Command**




Notes:
1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1)/4; For quad-SPI protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.

# ERASE Operations

An ERASE operation changes a bit from 0 to 1. Before any ERASE command is initiated, the WRITE ENABLE command must be executed to set the write enable latch bit to 1; if not, the device ignores the command and no error bits are set to indicate operation failure. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The operations are self-timed, and duration is $^{t}$SSE, $^{t}$SE, or $^{t}$BE according to command.

If S# is not driven HIGH, the command is not executed, flag status register error bits are not set, and the write enable latch remains set to 1. A command applied to a protected subsector is not executed. Instead, the write enable latch bit remains set to 1, and flag status register bits 1 and 5 are set.

When the operation is in progress, the program or erase controller bit of the flag status register is set to 0. In addition, the write in progress bit is set to 1. When the operation completes, the write in progress bit is cleared to 0. The write enable latch bit is cleared to 0, whether the operation is successful or not. If the operation times out, the write enable latch bit is reset and the erase error bit is set to 1.

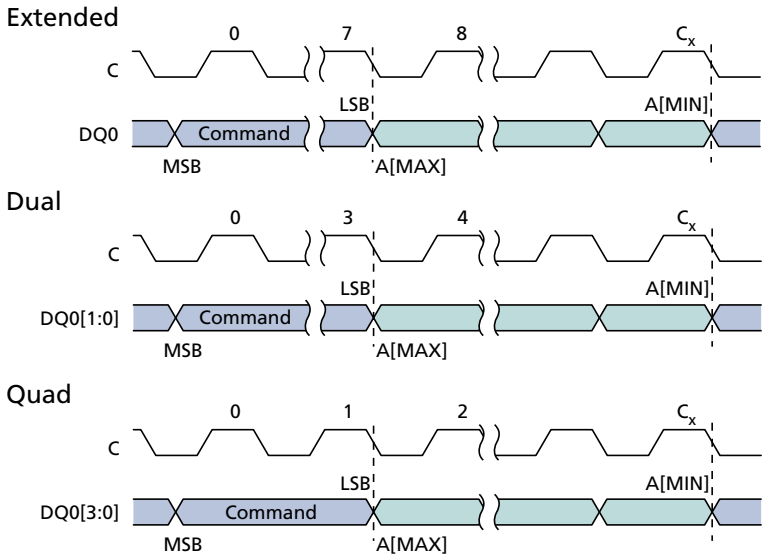
The status register and flag status register can be polled for the operation status. When the operation completes, these register bits are cleared to 1.

**Note:** For all ERASE operations, noisy or undesirable signal effects can be reduced and device data protection enhanced by holding S# LOW until the eighth bit of the last data byte has been latched in; this ensures that the number of clock pulses is a multiple of one byte before command execution.

**Table 32: ERASE Operations**

| Operation Name | Description/Conditions |
|---|---|
| SUBSECTOR ERASE | Sets the selected subsector or sector bits to FFh. Any address within the subsector is valid for entry. Each address bit is latched in during the rising edge of the clock. The operation can be suspended and resumed by the PROGRAM/ERASE SUSPEND and PROGRAM/ERASE RESUME commands, respectively. |
| SECTOR ERASE | |
| DIE ERASE | Sets the device bits to FFh.<br>The command is not executed if any sector is locked. Instead, the write enable latch bit remains set to 1, and flag status register bits 1 and 5 are set. |

**Figure 36: SUBSECTOR , SECTOR ERASE and DIE ERASE Timing**




Notes:
1. For extended SPI protocol, $C_x$ = 7 + (A[MAX] + 1); For dual SPI protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad SPI protocol, $C_x$ = 1 + (A[MAX] + 1)/4.
2. S# not shown.

# SUSPEND/RESUME Operations

## PROGRAM/ERASE SUSPEND Operations

A PROGRAM/ERASE SUSPEND command enables the memory controller to interrupt and suspend an array PROGRAM or ERASE operation within the program/erase latency. To initiate the command, S# is driven LOW, and the command code is input on DQn. The operation is terminated by the PROGRAM/ERASE RESUME command.

For a PROGRAM SUSPEND, the flag status register bit 2 is set to 1. For an ERASE SUSPEND, the flag status register bit 6 is set to 1.

After an erase/program latency time, the flag status register bit 7 is also set to 1, but the device is considered in suspended state once bit 7 of the flag status register outputs 1 with at least one byte output. In the suspended state, the device is waiting for any operation.

If the time remaining to complete the operation is less than the suspend latency, the device completes the operation and clears the flag status register bits 2 or 6, as applicable. Because the suspend state is volatile, if there is a power cycle, the suspend state information is lost and the flag status register powers up as 80h.

It is possible to nest a PROGRAM/ERASE SUSPEND operation inside a PROGRAM/ERASE SUSPEND operation just once. Issue an ERASE command and suspend it. Then issue a PROGRAM command and suspend it also. With the two operations suspended, the next PROGRAM/ERASE RESUME command resumes the latter operation, and a second PROGRAM/ERASE RESUME command resumes the former (or first) operation.

## PROGRAM/ERASE RESUME Operations

A PROGRAM/ERASE RESUME operation terminates the PROGRAM/ERASE RESUME command. To initiate the command, S# is driven LOW, and the command code is input on DQn. The operation is terminated by driving S# HIGH.

**Table 33: SUSPEND/RESUME Operations**

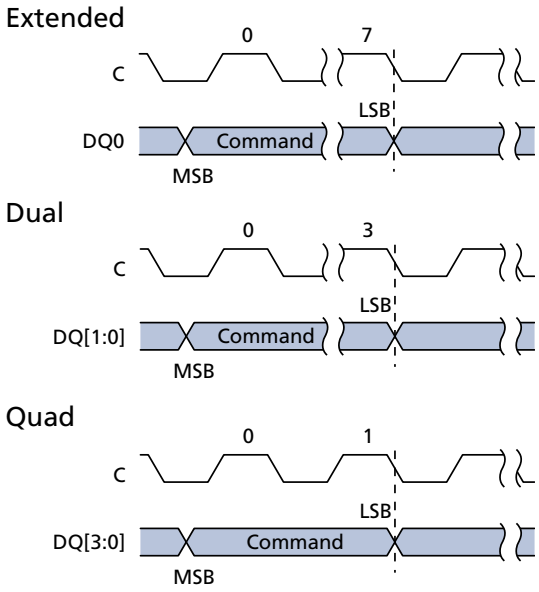
| Operation Name | Description/Conditions |
|---|---|
| PROGRAM SUSPEND (75h) | A READ operation is possible in any page except the one in a suspended state. Reading from a sector that is in a suspended state will output indeterminate data. |
| ERASE SUSPEND (75h) | A PROGRAM or READ operation is possible in any sector except the one in a suspended state. Reading from a sector that is in a suspended state will output indeterminate data.<br>During a SUSPEND SUBSECTOR ERASE operation, reading an address in the sector that contains the suspended subsector could output indeterminate data.<br>The device ignores a PROGRAM command to a sector that is in an erase suspend state; it also sets the flag status register bit 4 to 1 (program failure/protection error) and leaves the write enable latch bit unchanged.<br>When the ERASE resumes, it does not check the new lock status of the WRITE VOLATILE LOCK BITS command. |

**Table 33: SUSPEND/RESUME Operations (Continued)**

| Operation Name | Description/Conditions |
|---|---|
| PROGRAM RESUME (7Ah) | The status register write in progress bit is set to 1 and the flag status register program erase controller bit is set to 0. The command is ignored if the device is not in a suspended state.<br>When the operation is in progress, the program or erase controller bit of the flag status register is set to 0. The flag status register can be polled for the operation status. When the operation completes, that bit is cleared to 1. |
| ERASE RESUME (7Ah) | |

Note: 1. See the Operations Allowed/Disallowed During Device States table.

**Figure 37: PROGRAM/ERASE SUSPEND and RESUME Timing**



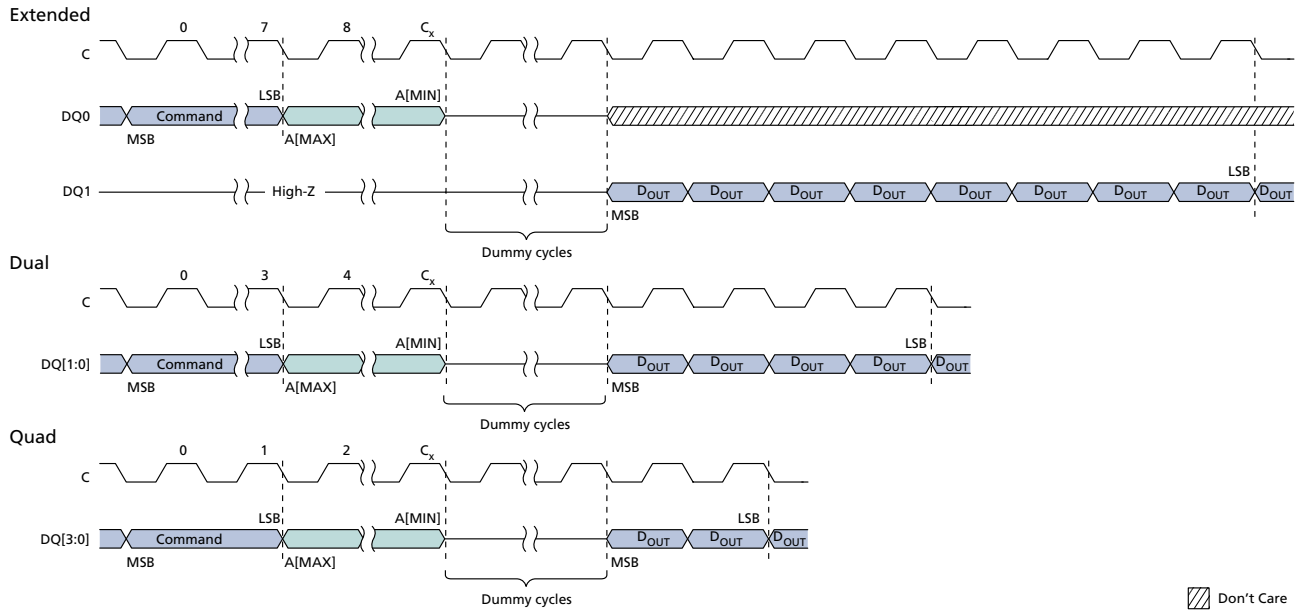

Note: 1. S# not shown.

# ONE-TIME PROGRAMMABLE Operations

## READ OTP ARRAY Command

To initiate a READ OTP ARRAY command, S# is driven LOW. The command code is input on DQ0, followed by address bytes and dummy clock cycles. Each address bit is latched in during the rising edge of C. Data is shifted out on DQ1, beginning from the specified address and at a maximum frequency of $^{f}C$ (MAX) on the falling edge of the clock. The address increments automatically to the next address after each byte of data is shifted out. There is no rollover mechanism; therefore, if read continuously, after location 0x40, the device continues to output data at location 0x40. The operation is terminated by driving S# HIGH at any time during data output.

**Figure 38: READ OTP ARRAY Command Timing**




Note: 1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1); For dual-SPI protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad-SPI protocol, $C_x$ = 1 + (A[MAX] + 1)/4.

## PROGRAM OTP ARRAY Command

To initiate the PROGRAM OTP ARRAY command, the WRITE ENABLE command must be issued to set the write enable latch bit to 1; otherwise, the PROGRAM OTP ARRAY command is ignored and flag status register bits are not set. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQ0, followed by address bytes and at least one data byte. Each address bit is latched in during the rising edge of the clock. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is $^{t}$POTP. There is no rollover mechanism; therefore, after a maximum of 65 bytes are latched in the subsequent bytes are discarded.

PROGRAM OTP ARRAY programs, at most, 64 bytes to the OTP memory area and one OTP control byte. When the operation is in progress, the write in progress bit is set to 1.

The write enable latch bit is cleared to 0, whether the operation is successful or not, and the status register and flag status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0.

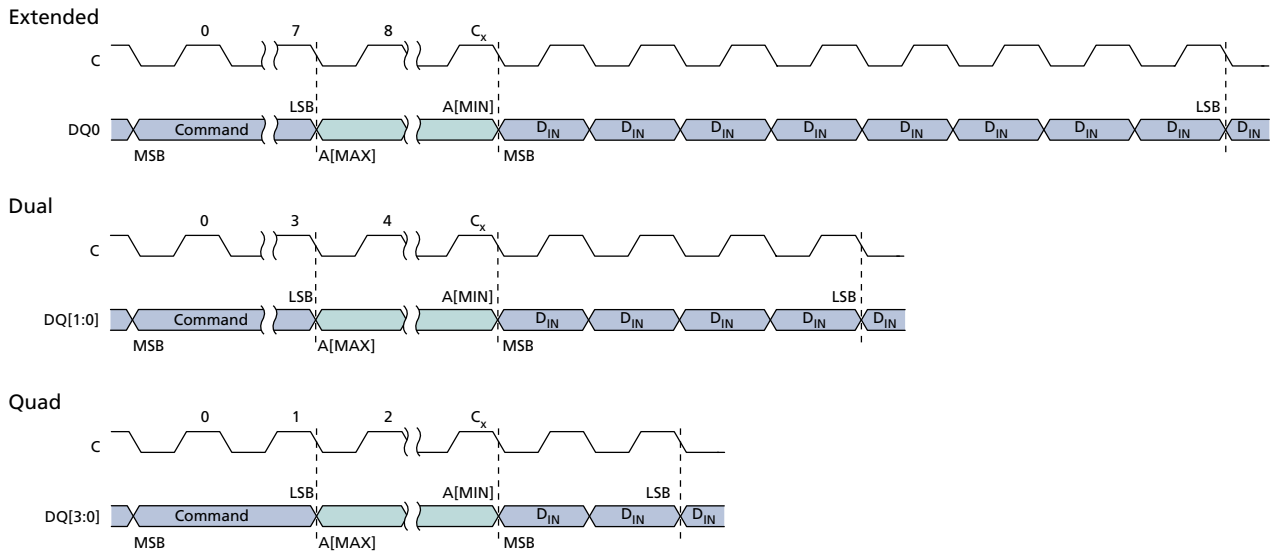
If the operation times out, the write enable latch bit is reset and the program fail bit is set to 1. If S# is not driven HIGH, the command is not executed, flag status register error bits are not set, and the write enable latch remains set to 1. The operation is considered complete once bit 7 of the flag status register outputs 1 with at least one byte output.

The OTP control byte (byte 64) is used to permanently lock the OTP memory array.

**Table 34: OTP Control Byte (Byte 64)**

| Bit | Name | Settings | Description |
|---|---|---|---|
| 0 | OTP control byte | 0 = Locked<br>1 = Unlocked (default) | Used to permanently lock the 64-byte OTP array. When bit 0 = 1, the 64-byte OTP array can be programmed. When bit 0 = 0, the 64-byte OTP array is read only.<br><br>Once bit 0 has been programmed to 0, it can no longer be changed to 1. Program OTP array is ignored, the write enable latch bit remains set, and flag status register bits 1 and 4 are set. |

**Figure 39: PROGRAM OTP Command Timing**




Note: 1. For extended-SPI protocol, $C_x$ = 7 + (A[MAX] + 1); For dual-SPI protocol, $C_x$ = 3 + (A[MAX] + 1)/2; For quad-SPI protocol, $C_x$ = 1 + (A[MAX] + 1)/4.

# ADDRESS MODE Operations

## ENTER and EXIT 4-BYTE ADDRESS MODE Command

To initiate these commands, S# is driven LOW, and the command is input on DQ*n*.

**Table 35: ENTER and EXIT 4-BYTE ADDRESS MODE Operations**

| Operation Name | Description/Conditions |
|---|---|
| ENTER 4-BYTE ADDRESS MODE (B7h) | The effect of the command is immediate. The default address mode is three bytes, and the device returns to the default upon exiting the 4-byte address mode. |
| EXIT 4-BYTE ADDRESS MODE (E9h) | |

# DEEP POWER-DOWN Operations

## ENTER DEEP POWER-DOWN Command

To execute ENTER DEEP POWER-DOWN, S# must be driven HIGH after the eighth bit of the command code is latched in, after which, $^{t}DP$ time must elapse before the supply current is reduced to $I_{CC2.}$. Any attempt to execute ENTER DEEP POWER-DOWN during a WRITE operation is rejected without affecting the operation.

In deep power-down mode, no device error bits are set, the WEL state is unchanged, and the device ignores all commands except RELEASE FROM DEEP POWER-DOWN, RESET ENABLE, RESET, hardware reset, and power-loss rescue sequence commands.

## RELEASE FROM DEEP POWER-DOWN Command

To execute the RELEASE FROM DEEP POWER-DOWN command, S# is driven LOW, followed by the command code. Sending additional clock cycles on C while S# is driven LOW voids the command.

RELEASE FROM DEEP POWER-DOWN is terminated by driving S# HIGH. The device enters standby mode after S# is driven HIGH followed by a delay of $^{t}RDP$. S# must remain HIGH during this time.

**Table 36: DEEP POWER-DOWN Operations**

| Operation Name | Description/Conditions |
|---|---|
| ENTER DEEP POWER-DOWN (B9h) | The command is used to place the device in deep power-down mode for the lowest device power consumption, with device current reduced to $I_{CC2.}$ This command can also be used as a software protection mechanism while the device is not in active use. |
| RELEASE FROM DEEP POWER-DOWN (ABh) | The command is used to exit from deep power-down mode. The device also exits deep power-down mode upon:<br>A power-down, entering standby mode with the next power-up.<br>A hardware or software reset operation, entering standby mode with a recovery time as specified in the AC Reset Specifications. |

## DEEP POWER-DOWN Timings

**Figure 40: ENTER DEEP POWER-DOWN Timing**

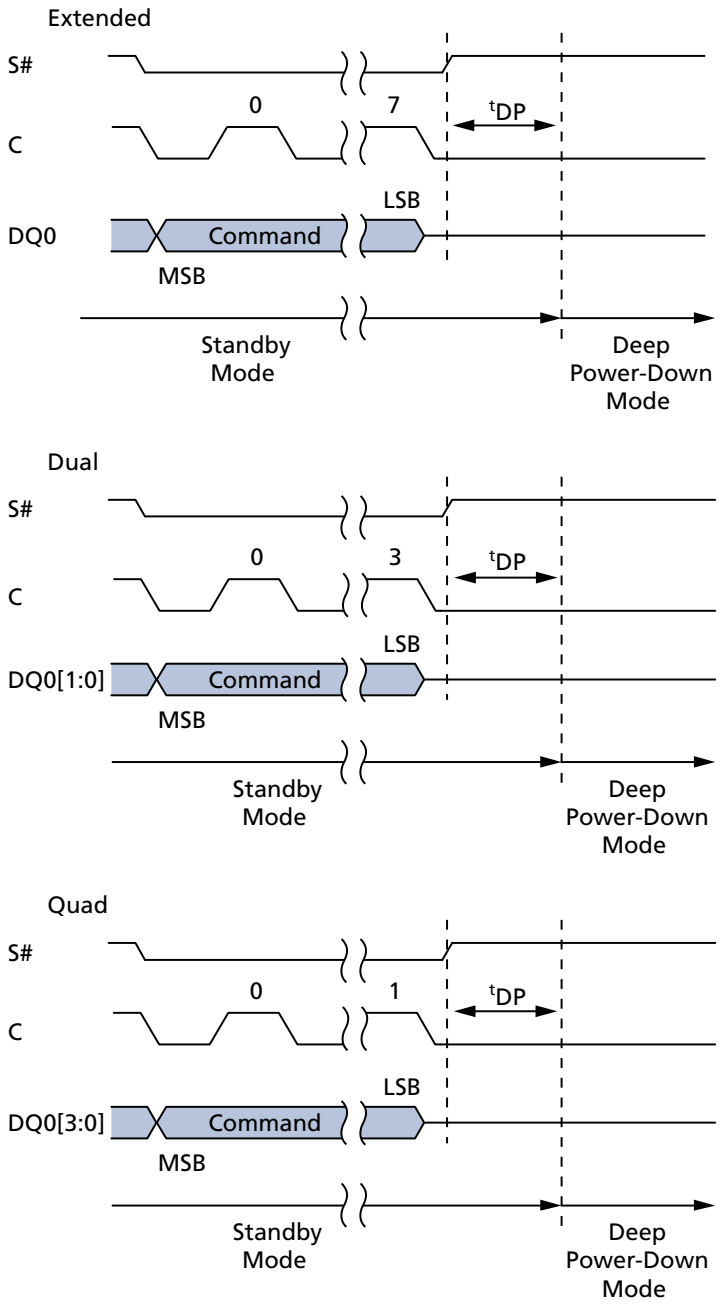
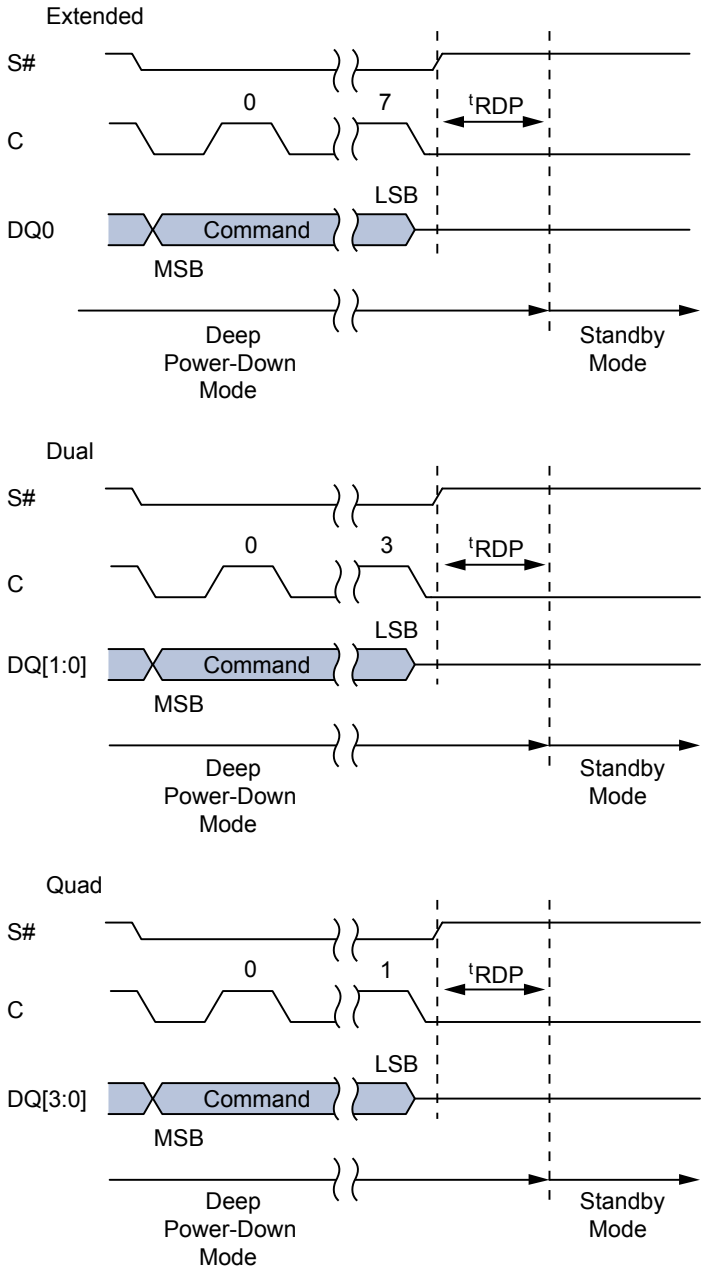


**Figure 41: RELEASE FROM DEEP POWER-DOWN Timing**



# QUAD PROTOCOL Operations

## ENTER or RESET QUAD INPUT/OUTPUT MODE Command

To initiate these commands, the WRITE ENABLE command must not be executed. S# must be driven LOW, and the command must be input on DQ*n*.

**Table 37: ENTER and RESET QUAD PROTOCOL Operations**

| Operation Name | Description/Conditions |
|---|---|
| ENTER QUAD INPUT/OUTPUT MODE (35h) | The effect of the command is immediate. |
| RESET QUAD INPUT/OUTPUT MODE (F5h) | |

# CYCLIC REDUNDANCY CHECK Operations

## Cyclic Redundancy Check

A CYCLIC REDUNDANCY CHECK (CRC) operation is a hash function designed to detect accidental changes to raw data and is used commonly in digital networks and storage devices such as hard disk drives. A CRC-enabled device calculates a short, fixed-length binary sequence, known as the CRC code or just CRC, for each block of data. CRC can be a higher performance alternative to reading data directly in order to verify recently programmed data. Or, it can be used to check periodically the data integrity of a large block of data against a stored CRC reference over the life of the product. CRC helps improve test efficiency for programmer or burn-in stress tests. No system hardware changes are required to enable CRC.

The device CRC operation generates the CRC result of an address range specified by the operation. Then the CRC result is compared with the expected CRC data provided in the sequence. Finally the device indicates a pass or fail through the status register. If the CRC fails, it is possible to take corrective action such as verifying with a normal read mode or by rewriting the array data.

The CRC-64 operation follows the ECMA standard. The generating polynomial is:

$$G(x) = x^{64} + x^{62} + x^{57} + x^{55} + x^{54} + x^{53} + x^{52} + x^{47} + x^{46} + x^{45} + x^{40} + x^{39} + x^{38} + x^{37} + x^{35} + x^{33} + x^{32} + x^{31} + x^{29} + x^{27} + x^{24} + x^{23} + x^{22} + x^{21} + x^{19} + x^{17} + x^{13} + x^{12} + x^{10} + x^{9} + x^{7} + x^{4} + x + 1$$

**Note:** The data stream sequence is from LSB to MSB and the default initial CRC value is all zero.

The state of the write enable latch bit is "Don't Care" for the CYCLIC REDUNDANCY CHECK operation. The stop address must be equal to or greater than the start address; otherwise, the device will abort the operation with flag status register bit 1 set. The minimum granularity of the range is one byte. For stacked devices the start and the end byte address must belong to the same die.

If the CRC value generated by device does not match value provided by the user, error is indicated by setting flag status register bit 4.The CRC operation cannot be suspended. The user interface accepts a suspend request (immediately sets flag status register bit 2); however, the device ignores suspend request and completes the operation (at the end of the operation, flag status register bit 2 is cleared). The operation is aborted with hardware or software reset.

CRC operation supports CRC data read back when CRC check fails; the CRC data generated from the target address range or entire device will be stored in a dedicated register general purpose read register (GPRR) only when CRC check fails, and it can be read out through the GPRR read sequence with command 96h, least significant byte first. GPRR is reset to default all 0 at the beginning of the CRC operation, and so customer will read all 0 if CRC operation pass.

Note that the GPRR is a volatile register. It is cleared to all 0s on power-up and hardware/software reset. Read GPRR starts from the first location, when clocked continuously, will output 00h after location 64.

**Table 38: CRC Command Sequence on a Range**

| Command Sequence | | Description |
|---|---|---|
| Byte# | Data | |
| 1 | 9Bh | Command code for interface activation |
| 2 | 27h | Sub-command code for CRC operation |
| 3 | FEh | CRC operation option selection (CRC operation on a range) |
| 4 | CRC[7:0] | 1st byte of expected CRC value |
| 5 to 10 | CRC[55:8] | 2nd to 7th byte of expected CRC value |
| 11 | CRC[63:56] | 8th byte of expected CRC value |
| 12 | Start address [7:0] | Specifies the starting byte address for CRC operation |
| 13 to 14 | Start address [23:8] | |
| 15 | Start address [31:24] | |
| 16 | Start address [7:0] | Specifies the ending byte address for CRC operation |
| 17 to 18 | Start address [23:8] | |
| 19 | Start address [31:24] | |
| Drive S# HIGH | | Operation sequence confirmed; CRC operation starts |

# State Table

The device can be in only one state at a time. Depending on the state of the device, some operations as shown in the table below are allowed (Yes) and others are not (No). For example, when the device is in the standby state, all operations except SUSPEND are allowed in any sector. For all device states except the erase suspend state, if an operation is allowed or disallowed in one sector, it is allowed or disallowed in all other sectors. In the erase suspend state, a PROGRAM operation is allowed in any sector except the one in which an ERASE operation has been suspended.

**Table 39: Operations Allowed/Disallowed During Device States**

| Operation | Standby State | Program or Erase State | Subsector Erase Suspend or Program Suspend State | Erase Suspend State | Notes |
|---|---|---|---|---|---|
| READ (memory) | Yes | No | Yes | Yes | 1 |
| READ (status/flag status registers) | Yes | Yes | Yes | Yes | 6 |
| PROGRAM | Yes | No | No | Yes/No | 2 |
| ERASE (sector/subsector) | Yes | No | No | No | 3 |
| WRITE | Yes | No | No | No | 4 |
| WRITE | Yes | No | Yes | Yes | 5 |
| SUSPEND | No | Yes | No | No | 7 |

Notes:
1. All READ operations except READ STATUS REGISTER and READ FLAG REGISTER. When issued to a sector or subsector that is simultaneously in an erase suspend state, the READ operation is accepted, but the data output is not guaranteed until the erase has completed.
2. All PROGRAM operations except PROGRAM OTP. In the erase suspend state, a PROGRAM operation is allowed in any sector (Yes) except the sector (No) in which an ERASE operation has been suspended.
3. Applies to the SECTOR ERASE or SUBSECTOR ERASE operation.
4. Applies to the following operations: WRITE STATUS REGISTER, WRITE NONVOLATILE CONFIGURATION REGISTER, PROGRAM OTP, and DIE ERASE.
5. Applies to the WRITE VOLATILE CONFIGURATION REGISTER, WRITE ENHANCED VOLATILE CONFIGURATION REGISTER, WRITE ENABLE, WRITE DISABLE, CLEAR FLAG STATUS REGISTER, WRITE EXTENDED ADDRESS REGISTER, or WRITE LOCK REGISTER operation.
6. Applies to the READ STATUS REGISTER or READ FLAG STATUS REGISTER operation.
7. Applies to the PROGRAM SUSPEND or ERASE SUSPEND operation.

# XIP Mode

Execute-in-place (XIP) mode allows the memory to be read by sending an address to the device and then receiving the data on one, two, or four pins in parallel, depending on the customer requirements. XIP mode offers maximum flexibility to the application, saves instruction overhead, and reduces random access time.

## Activate and Terminate XIP Using Volatile Configuration Register

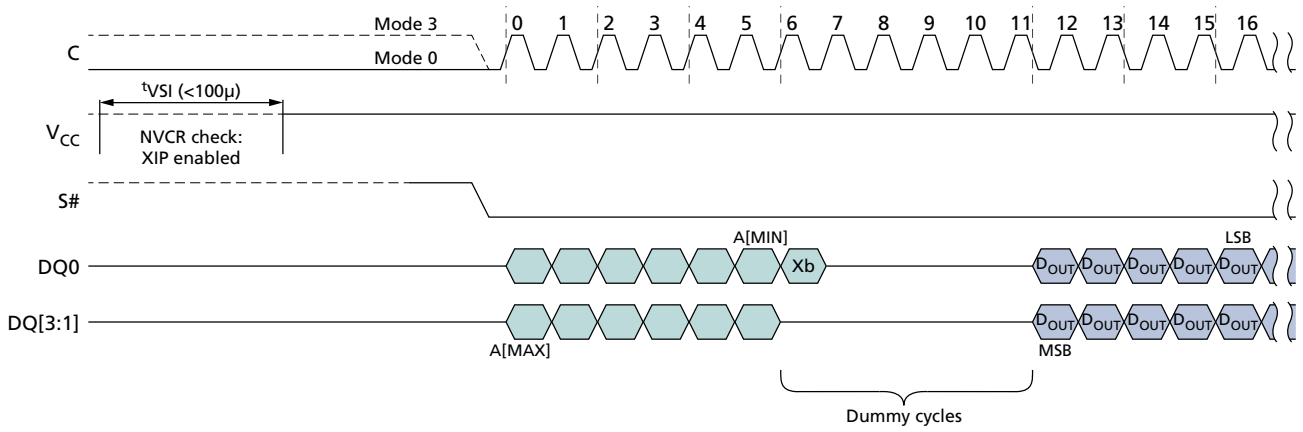
Applications that boot in SPI and must switch to XIP use the volatile configuration register. XIP provides faster memory READ operations by requiring only an address to execute, rather than a command code and an address.

To activate XIP requires two steps. First, enable XIP by setting volatile configuration register bit 3 to 0. Next, drive the XIP confirmation bit to 0 during the next FAST READ operation. XIP is then active. Once in XIP, any command that occurs after S# is toggled requires only address bits to execute; a command code is not necessary, and device operations use the SPI protocol that is enabled. XIP is terminated by driving the XIP confirmation bit to 1. The device automatically resets volatile configuration register bit 3 to 1.

## Activate and Terminate XIP Using Nonvolatile Configuration Register

Applications that must boot directly in XIP use the nonvolatile configuration register. To enable a device to power-up in XIP using this register, set nonvolatile configuration register bits [11:9]. Settings vary according to protocol, as explained in the Nonvolatile Configuration Register section. Because the device boots directly in XIP, after the power cycle, no command code is necessary. XIP is terminated by driving the XIP confirmation bit to 1.

**Figure 42: XIP Mode Directly After Power-On**




Note: 1. Xb is the XIP confirmation bit and should be set as follows: 0 to keep XIP state; 1 to exit XIP mode and return to standard read mode.

## Confirmation Bit Settings Required to Activate or Terminate XIP

The XIP confirmation bit setting activates or terminates XIP after it has been enabled or disabled. This bit is the value on DQ0 during the first dummy clock cycle in the FAST READ operation. In dual I/O XIP mode, the value of DQ1 during the first dummy clock cycle after the addresses is always "Don't Care." In quad I/O XIP mode, the values of DQ3, DQ2, and DQ1 during the first dummy clock cycle after the addresses are always "Don't Care."

**Table 40: XIP Confirmation Bit**

| Bit Value | Description |
|---|---|
| 0 | Activates XIP: While this bit is 0, XIP remains activated. |
| 1 | Terminates XIP: When this bit is set to 1, XIP is terminated and the device returns to SPI. |

**Table 41: Effects of Running XIP in Different Protocols**

| Protocol | Effect |
|---|---|
| Extended I/O and Dual I/O | In a device with a dedicated part number where RESET# is enabled, a LOW pulse on that pin resets XIP and the device to the state it was in previous to the last power-up, as defined by the nonvolatile configuration register. |
| Dual I/O | Values of DQ1 during the first dummy clock cycle are "Don't Care." |
| Quad I/O[1] | Values of DQ[3:1] during the first dummy clock cycle are "Don't Care." In a device with a dedicated part number, it is only possible to reset memory when the device is deselected. |

Note: 1. In a device with a dedicated part number where RESET# is enabled, a LOW pulse on that pin resets XIP and the device to the state it was in previous to the last power-up, as defined by the nonvolatile configuration register only when the device is deselected.

## Terminating XIP After a Controller and Memory Reset

The system controller and the device can become out of synchronization if, during the life of the application, the system controller is reset without the device being reset. In such a case, the controller can reset the memory to power-on reset if the memory has reset functionality. (Reset is available in devices with a dedicated part number.)

- 7 clock cycles within S# LOW (S# becomes HIGH before 8th clock cycle)
- + 9 clock cycles within S# LOW (S# becomes HIGH before 10th clock cycle)
- + 13 clock cycles within S# LOW (S# becomes HIGH before 14th clock cycle)
- + 17 clock cycles within S# LOW (S# becomes HIGH before 18th clock cycle)
- + 25 clock cycles within S# LOW (S# becomes HIGH before 26th clock cycle)
- + 33 clock cycles within S# LOW (S# becomes HIGH before 34th clock cycle)

These sequences cause the controller to set the XIP confirmation bit to 1, thereby terminating XIP. However, it does not reset the device or interrupt PROGRAM/ERASE operations that may be in progress. After terminating XIP, the controller must execute RESET ENABLE and RESET MEMORY to implement a software reset and reset the device.

# Power-Up and Power-Down

## Power-Up and Power-Down Requirements

At power-up and power-down, the device must not be selected; that is, S# must follow the voltage applied on $V_{CC}$ until $V_{CC}$ reaches the correct values: $V_{CC,min}$ at power-up and $V_{SS}$ at power-down.

To provide device protection and prevent data corruption and inadvertent WRITE operations during power-up, a power-on reset circuit is included. The logic inside the device is held to RESET while $V_{CC}$ is less than the power-on reset threshold voltage shown here; all operations are disabled, and the device does not respond to any instruction. During a standard power-up phase, the device ignores all commands except READ STATUS REGISTER and READ FLAG STATUS REGISTER. These operations can be used to check the memory internal state. After power-up, the device is in standby power mode; the write enable latch bit is reset; the write in progress bit is reset; and the dynamic protection register is configured as: (write lock bit, lock down bit) = (0,0).

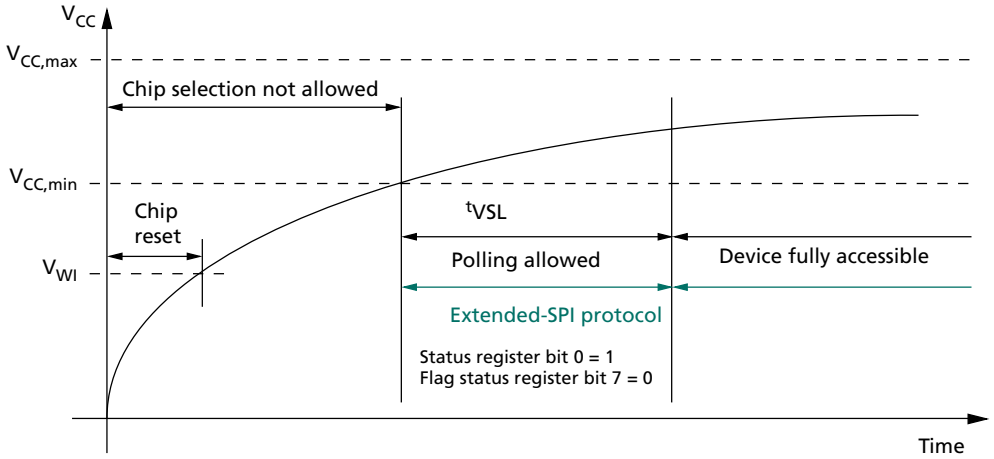
Normal precautions must be taken for supply line decoupling to stabilize the $V_{CC}$ supply. Each device in a system should have the $V_{CC}$ line decoupled by a suitable capacitor (typically 100nF) close to the package pins. At power-down, when $V_{CC}$ drops from the operating voltage to below the power-on-reset threshold voltage shown here, all operations are disabled and the device does not respond to any command.

When the operation is in progress, the program or erase controller bit of the flag status register is set to 0. It's possible to obtain the operation status by reading the flag status register a number of times corresponding to the die stacked, with S# toggled in between the READ FLAG STATUS REGISTER commands. When the operation completes, the program or erase controller bit of the flag status register is cleared to 1. The end of operation can be detected when the program or erase controller bit of the flag status register outputs 1 for all the die of the stack. Alternatively, it's possible to wait tVSL and in that case polling the flag status register is not required.

**Note:** If power-down occurs while a WRITE, PROGRAM, or ERASE cycle is in progress, data corruption may result.

**Note:** For additional details about how to properly apply and remove the power supply to the device, refer to TN-25-38: Power-Up, Power-Down, and Brownout Considerations on MT25Q, MT25T, and MT35X NOR Flash Memory

**Figure 43: Power-Up Timing**




Notes:
1. $^{t}VSL$ polling has to be in Extended-SPI protocol and STR mode.
2. During $^{t}VSL$ period, HOLD# is enabled, RESET# disabled, and output strength is in default setting.
3. In a system that uses a fast $V_{CC}$ ramp rate, current design requires a minimum 100µs after $V_{CC}$ reaches $^{t}VWI$, and before the polling is allowed, even though $V_{CC,min}$ is achieved.
4. In extended SPI protocol, the 1Gb/2Gb device must wait 100us after $V_{CC}$ reaches $V_{CC,min}$ before polling the status register or flag status register.

**Table 42: Power-Up Timing and $V_{WI}$ Threshold**

Note 1 applies to entire table

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $^{t}VSL$ | $V_{CC,min}$ to device fully accessible | – | 300 | µs | 2, 3 |
| $V_{WI}$ | Write inhibit voltage | 1.5 | 2.5 | V | 2 |

Notes:
1. When $V_{CC}$ reaches $V_{CC,min}$, to determine whether power-up initialization is complete, the host can poll status register bit 0 or flag status register bit 7 only in extended SPI protocol because the device will accept commands only on DQ0 and output data only on DQ1. When the device is ready, the host has full access using the protocol configured in the nonvolatile configuration register. If the host cannot poll the status register in x1 SPI mode, it is recommended to wait $^{t}VSL$ before accessing the device.
2. Parameters listed are characterized only.
3. On the first power up after an event causing a sub-sector erase operation interrupt (e.g. due to power-loss), the maximum time for tVSL will be up to 4.5ms in case of 4KB sub-sector erase interrupt and up to 36ms in case of 32KB sub-sector erase interrupt; this accounts for erase recovery embedded operation.

# Active, Standby, and Deep Power-Down Modes

When S# is LOW, the device is selected and in active power mode. When S# is HIGH, the device is deselected but could remain in active power mode until ongoing internal operations are completed. Then the device goes into standby power mode and device current consumption drops to $I_{CC1}$.

Deep power-down mode enbles users to place the device in the lowest power consumption mode, $I_{CC2}$. The ENTER DEEP POWER-DOWN command is used to put the device in deep power-down mode, and the RELEASE FROM DEEP POWER-DOWN command is used to bring the device out of deep power-down mode. Command details are in the Command Set table and the DEEP POWER-DOWN Operations section of this data sheet.

# Power Loss and Interface Rescue

If a power loss occurs during a WRITE NONVOLATILE CONFIGURATION REGISTER command, after the next power-on, the device might begin in an undetermined state (XIP mode or an unnecessary protocol). If this occurs, a power loss recovery sequence must reset the device to a fixed state (extended-SPI protocol without XIP) until the next power-up.

If the controller and memory device get out of synchronization, the controller can follow an interface rescue sequence to reset the memory device interface to power-up to the last reset state (as defined by latest nonvolatile configuration register). This resets only the interface, not the entire memory device, and any ongoing operations are not interrupted.

After each sequence, the issue should be resolved definitively by running the WRITE NONVOLATILE CONFIGURATION REGISTER command again.

**Note:** The two steps in each sequence must be in the correct order, and $^{t}$SHSL2 must be at least 50ns for the duration of each sequence.

The first step for both the power loss recovery and interface rescue sequences is described under "Recovery." The second step in the power loss recovery sequence is under "Power Loss Recovery" and the second step in the interface rescue sequence is under "Interface Rescue."

## Recovery

Step one of both the power loss recovery and interface rescue sequences is DQ0 (PAD DATA) and DQ3 (PAD HOLD) equal to 1 for the situations listed here:

- 7 clock cycles within S# LOW (S# becomes HIGH before 8th clock cycle)
- + 9 clock cycles within S# LOW (S# becomes HIGH before 10th clock cycle)
- + 13 clock cycles within S# LOW (S# becomes HIGH before 14th clock cycle)
- + 17 clock cycles within S# LOW (S# becomes HIGH before 18th clock cycle)
- + 25 clock cycles within S# LOW (S# becomes HIGH before 26th clock cycle)
- + 33 clock cycles within S# LOW (S# becomes HIGH before 34th clock cycle)

## Power Loss Recovery

For power loss recovery, the second part of the sequence is exiting from dual- or quad-SPI protocol by using the following FFh sequence: DQ0 and DQ3 equal to 1 for 8 clock cycles within S# LOW; S# becomes HIGH before 9th clock cycle. After this two-part sequence the extended-SPI protocol is active.

## Interface Rescue

For interface rescue, the second part of the sequence is for exiting from dual or quad-SPI protocol by using the following FFh sequence: DQ0 and DQ3 equal to 1 for 16 clock cycles within S# LOW; S# becomes HIGH before 17th clock cycle. For DTR protocol, 1 should be driven on both edges of clock for 16 cycles with S# LOW. After this two-part sequence, the extended-SPI protocol is active.

# Absolute Ratings and Operating Conditions

Stresses greater than those listed may cause permanent damage to the device. This is a stress rating only. Exposure to absolute maximum rating for extended periods may adversely affect reliability. Stressing the device beyond the absolute maximum ratings may cause permanent damage.

**Table 43: Absolute Ratings**

| Symbol | Parameter | Min | Max | Units | Notes |
|---|---|---|---|---|---|
| $T_{STG}$ | Storage temperature | –65 | 150 | °C | |
| $T_{LEAD}$ | Lead temperature during soldering | – | See note 1 | °C | |
| $V_{CC}$ | Supply voltage | –0.6 | 4.0 | V | 2 |
| $V_{IO}$ | Input/output voltage with respect to ground | -0.6 | $V_{CC}$ + 0.6 | V | 2 |
| $V_{ESD}$ | Electrostatic discharge voltage (human body model) | –2000 | 2000 | V | 2, 3 |

Notes:
1. Compliant with JEDEC Standard J-STD-020C (for small-body, Sn-Pb or Pb assembly), RoHS, and the European directive on Restrictions on Hazardous Substances (RoHS) 2002/95/EU.
2. All specified voltages are with respect to $V_{SS}$. During infrequent, nonperiodic transitions, the voltage potential between $V_{SS}$ and the $V_{CC}$ may undershoot to –2.0V for periods less than 20ns, or overshoot to $V_{CC,max}$ + 2.0V for periods less than 20ns.
3. JEDEC Standard JESD22-A114A (C1 = 100pF, R1 = 1500Ω, R2 = 500Ω).

**Table 44: Operating Conditions**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | 2.7 | 3.6 | V |
| $T_A$ | Ambient operating temperature (IT range) | –40 | 85 | °C |
| $T_A$ | Ambient operating temperature (AT range) | –40 | 105 | °C |
| $T_A$ | Ambient operating temperature (UT range) | –40 | 125 | °C |

**Table 45: Input/Output Capacitance**

Note 1 applies to entire table

| Symbol | Description | Min | Max | Units |
|---|---|---|---|---|
| $C_{IN/OUT}$ | Input/output capacitance (DQ0/DQ1/DQ2/DQ3) | – | 18 | pF |
| $C_{IN}$ | Input capacitance (other pins) | – | 8 | pF |
| $C_{IN/S\#}$ | Input/Chip select | – | 14 | pF |

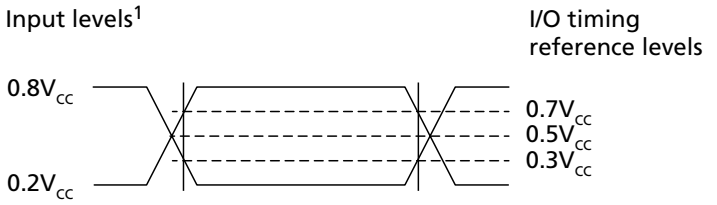
Note:
1. Verified in device characterization; not 100% tested. These parameters are not subject to a production test. They are verified by design and characterization. The capacitance is measured according to JEP147 ("PROCEDURE FOR MEASURING INPUT CAPACITANCE USING A VECTOR NETWORK ANALYZER (VNA)") with $V_{CC}$ and $V_{SS}$ applied and all other pins floating (except the pin under test), $V_{BIAS}$ = $V_{CC}$/2, $T_A$ = 25°C, Frequency = 54 MHz

### Table 46: AC Timing Input/Output Conditions

| Symbol | Description | Min | Max | Units | Notes |
|---|---|---|---|---|---|
| $C_L$ | Load capacitance | - | 30 | pF | 1 |
| – | Input rise and fall times | – | 1.5 | ns | |
| | Input pulse voltages | $0.2V_{CC}$ to $0.8V_{CC}$ | | V | 2 |
| | Input timing reference voltages | $0.3V_{CC}$ to $0.7V_{CC}$ | | V | |
| | Output timing reference voltages | $V_{CC}/2$ | | V | |

Notes: 1. Output buffers are configurable by user.
2. For quad/dual operations: 0V to $V_{CC}$.

### Figure 44: AC Timing Input/Output Reference Levels




Note: 1. $0.8V_{CC} = V_{CC}$ for dual/quad operations; $0.2V_{CC} = 0V$ for dual/quad operations.

# DC Characteristics and Operating Conditions

### Table 47: DC Current Characteristics and Operating Conditions

Notes 1–5 apply to entire table

| Parameter | Symbol | Test Conditions | Typ | Max | Unit |
|---|---|---|---|---|---|
| Input leakage current | $I_{LI}$ | | – | ±2 | µA |
| Output leakage current | $I_{LO}$ | | – | ±2 | µA |
| Standby current (IT range) | $I_{CC1}$ | S# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | 45 | 160 | µA |
| Standby current (AT range) | $I_{CC1}$ | S# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | 60 | 300 | µA |
| Standby current (UT range) | $I_{CC1}$ | S# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | 60 | 600 | µA |
| Deep power-down current (IT range) | $I_{CC2}$ | S# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | 10 | 60 | µA |
| Deep power-down current (AT range) | $I_{CC2}$ | S# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | 10 | 140 | µA |
| Deep power-down current (UT range) | $I_{CC2}$ | S# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | 10 | 300 | µA |
| Operating current (fast-read extended I/O) | $I_{CC3}$ | C = 0.1$V_{CC}$/0.9$V_{CC}$ at 133 MHz, DQ1 = open | – | 35 | mA |
| | | C = 0.1$V_{CC}$/0.9$V_{CC}$ at 54 MHz, DQ1 = open | – | 20 | mA |
| Operating current (fast-read dual I/O) | | C = 0.1$V_{CC}$/0.9$V_{CC}$ at 133 MHz DQ = open | – | 35 | mA |
| Operating current (fast-read quad I/O) | | C = 0.1$V_{CC}$/0.9$V_{CC}$ at 133 MHz DQ = open | – | 45 | mA |
| | | C = 0.1$V_{CC}$/0.9$V_{CC}$ at 80 MHz DTR, DQ = open | – | 50 | mA |
| | | C = 0.1$V_{CC}$/0.9V; at 90 MHz DTR, DQ = open | – | 55 | mA |
| Operating current (PROGRAM operations) | $I_{CC4}$ | S# = $V_{CC}$ | – | 35 | mA |
| Operating current (WRITE operations) | $I_{CC5}$ | S# = $V_{CC}$ | – | 35 | mA |
| Operating current (erase) | $I_{CC6}$ | S# = $V_{CC}$ | – | 35 | mA |

Notes:
1. All currents are RMS unless noted. Typical values at typical $V_{CC}$ (3.0/1.8V); $V_{IO}$ = 0V/$V_{CC}$; $T_C$ = +25°C.
2. Standby current is the average current measured over any time interval 5µs after S de-assertion (and any internal operations are complete).
3. Deep power-down current is the average current measured 5ms over any 5ms time interval, 100µs after the ENTER DEEP POWER-DOWN operation (and any internal operations are complete).
4. All read currents are the average current measured over any 1KB continuous read. No load, checker-board pattern.
5. All program currents are the average current measured over any 256-byte typical data program.

## Table 48: DC Voltage Characteristics and Operating Conditions

Note 1 applies to entire table

| Parameter | Symbol | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| Input low voltage | $V_{IL}$ | | –0.5 | $0.3V_{CC}$ | V |
| Input high voltage | $V_{IH}$ | | $0.7V_{CC}$ | $V_{CC}$ + 0.4 | V |
| Output low voltage | $V_{OL}$ | $I_{OL}$ = 1.6mA | – | 0.4 | V |
| Output high voltage | $V_{OH}$ | $I_{OH}$ = –100µA | $V_{CC}$ - 0.2 | – | V |

Note: 1. $V_{IL}$ can undershoot to –1.0V for periods <2ns and $V_{IH}$ may overshoot to $V_{CC,max}$ + 1.0V for periods less than 2ns.

# AC Characteristics and Operating Conditions

### Table 49: Supported Maximum Frequency

| Parameter | Symbol | Parts | Single IO STR | Single IO DTR | Dual IO STR | Dual IO DTR | Quad IO STR | Quad IO DTR | Unit |
|---|---|---|---|---|---|---|---|---|---|
| Clock frequency for all commands other than READ (Extended-SPI, DIO-SPI, and QIO-SPI protocol) | $^{f}C$ | IT, AT | 133 | 90 | 133 | 90 | 133 | 90 | MHz |
| | | UT | 133 | 90 | 133 | 90 | 108 | 80 | |
| Clock frequency for READ command (03h) | $^{f}R$ | IT, AT, UT | 54 | 27 | | | | | |

### Table 50: AC Characteristics and Operating Conditions

| Parameter | Symbol | Data Transfer Rate | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Clock HIGH time | $^{t}CH$ | STR | 3.375 | – | – | ns | 2,3 |
| | | DTR | 5.0 | – | – | | |
| Clock LOW time | $^{t}CL$ | STR | 3.375 | – | – | ns | 2,4 |
| | | DTR | 5.0 | – | – | | |
| Clock rise time (peak-to-peak) | $^{t}CLCH$ | STR/DTR | 0.1 | – | – | V/ns | 5, 6 |
| Clock fall time (peak-to-peak) | $^{t}CHCL$ | STR/DTR | 0.1 | – | – | V/ns | 5, 6 |
| S# active setup time (relative to clock) | $^{t}SLCH$ | STR/DTR | 3.375 | – | – | ns | |
| S# not active hold time (relative to clock) | $^{t}CHSL$ | STR/DTR | 3.375 | – | – | ns | |
| Data in setup time | $^{t}DVCH$ | STR | 1.75 | – | – | ns | |
| | | DTR | 1.5 | – | – | ns | |
| | $^{t}DVCL$ | DTR only | 1.5 | – | – | ns | |
| Data in hold time | $^{t}CHDX$ | STR/DTR | 2.3 | – | – | ns | |
| | $^{t}CLDX$ | DTR only | 2.3 | – | – | ns | |
| S# active hold time (relative to clock) | $^{t}CHSH$ | STR | 3.375 | – | – | ns | |
| | | DTR | 5.0 | – | – | | |
| S# active hold time (relative to clock LOW) Only for writes in DTR | $^{t}CLSH$ | DTR only | 3.375 | – | – | ns | |
| S# not active setup time (relative to clock) | $^{t}SHCH$ | STR | 3.375 | – | – | ns | |
| | | DTR | 5.0 | – | – | ns | |
| S# deselect time after a READ command | $^{t}SHSL1$ | STR/DTR | 20 | – | – | ns | |
| S# deselect time after a nonREAD command | $^{t}SHSL2$ | STR/DTR | 50 | – | – | ns | 7 |
| Output disable time | $^{t}SHQZ$ | STR/DTR | – | – | 7 | ns | 5 |
| Clock LOW to output valid under 30pF | $^{t}CLQV$ | STR/DTR | – | – | 6 | ns | |
| Clock LOW to output valid under 10pF | | STR/DTR | – | – | 5 | ns | |

**Table 50: AC Characteristics and Operating Conditions (Continued)**

| Parameter | Symbol | Data Transfer Rate | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Clock HIGH to output valid under 30pF | $^{t}$CHQV | DTR only | – | – | 6 | ns | |
| Clock HIGH to output valid under 10pF | | DTR only | – | – | 5 | ns | |
| Output hold time | $^{t}$CLQX | STR/DTR | 1.5 | – | – | ns | |
| Output hold time | $^{t}$CHQX | DTR only | 1.5 | – | – | ns | |
| HOLD setup time (relative to clock) | $^{t}$HLCH | STR/DTR | 3.375 | – | – | ns | |
| HOLD hold time (relative to clock) | $^{t}$CHHH | STR/DTR | 3.375 | – | – | ns | |
| HOLD setup time (relative to clock) | $^{t}$HHCH | STR/DTR | 3.375 | – | – | ns | |
| HOLD hold time (relative to clock) | $^{t}$CHHL | STR/DTR | 3.375 | – | – | ns | |
| HOLD to output Low-Z | $^{t}$HHQX | STR/DTR | – | – | 8 | ns | 5 |
| HOLD to output High-Z | $^{t}$HLQZ | STR/DTR | – | – | 8 | ns | 5 |
| CRC check time: main block | $^{t}$CRC | STR/DTR | – | 1.3 | - | ms | |
| CRC check time: full chip (512Mb) | $^{t}$CRC | STR/DTR | – | 2 | - | s | |
| Write protect setup time | $^{t}$WHSL | STR/DTR | 20 | – | – | ns | 8 |
| Write protect hold time | $^{t}$SHWL | STR/DTR | 100 | – | – | ns | 8 |
| S# HIGH to deep power-down | $^{t}$DP | STR/DTR | 3 | – | – | us | |
| S# HIGH to standby mode (DPD exit time) | $^{t}$RDP | STR/DTR | 30 | – | – | us | |
| WRITE STATUS REGISTER cycle time | $^{t}$W | STR/DTR | – | 1.3 | 8 | ms | |
| WRITE NONVOLATILE CONFIGURATION REGISTER cycle time | $^{t}$WNVCR | STR/DTR | – | 0.2 | 1 | s | |
| Nonvolatile sector lock time | $^{t}$PPBP | STR/DTR | – | 0.1 | 2.8 | ms | |
| Program ASP register | $^{t}$ASPP | STR/DTR | – | 0.1 | 0.5 | ms | |
| Program password | $^{t}$PASSP | STR/DTR | – | 0.2 | 0.8 | ms | |
| Erase nonvolatile sector lock array | $^{t}$PPBE | STR/DTR | – | 0.2 | 1 | s | |
| Page program time (256 bytes) | $^{t}$PP | STR/DTR | – | 120 | 1800 | us | 9 |
| Page program time (n bytes) | | | – | 18+ 2.5x int(n/6) | 1800 | us | 10 |
| PROGRAM OTP cycle time (64 bytes) | $^{t}$POTP | STR/DTR | – | 0.12 | 0.8 | ms | |
| Sector erase time | $^{t}$SE | STR/DTR | – | 0.15 | 1 | s | |
| 4KB subsector erase time | $^{t}$SSE | STR/DTR | – | 0.05 | 0.4 | s | |
| 32KB subsector erase time | $^{t}$SSE | STR/DTR | – | 0.1 | 1 | s | |
| 512Mb DIE ERASE time | $^{t}$BE | STR/DTR | – | 153 | 460 | s | |

Notes:
1. Typical values given for $T_A$ = 25 °C.
2. $^{t}$CH + $^{t}$CL must add up to 1/$^{f}$C.
3. Only for UT parts in Quad I/O: $^{t}$CH in STR = 4.0ns (MIN).
4. Only for UT parts in Quad I/O: $^{t}$CL in STR = 4.0ns (MIN).
5. Value guaranteed by characterization; not 100% tested.
6. Expressed as a slew-rate.

7. The non-READ commands are WRITE, PROGRAM, and ERASE.
8. Only applicable as a constraint for a WRITE STATUS REGISTER command when STATUS REGISTER WRITE is set to 1.
9. Typical value is applied for pattern: 50% = 0, and 50% = 1.
10. int(n) correspond to the integer part of n, For example, int (12/8) = 1; int (32/8) = 4; int(15.3) = 15.

# AC Reset Specifications

### Table 51: AC RESET Conditions

Note 1 applies to entire table

| Parameter | Symbol | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Reset pulse width | $^{t}$RLRH[2] | | 50 | – | – | ns |
| Reset recovery time | $^{t}$RHSL | Device deselected (S# HIGH) and is in XIP mode | 40 | – | – | ns |
| | | Device deselected (S# HIGH) and is in standby mode | 40 | – | – | ns |
| | | Commands are being decoded, any READ operations are in progress or any WRITE operation to volatile registers are in progress | 40 | – | – | ns |
| | | Any device array PROGRAM/ERASE/SUSPEND/RESUME, PROGRAM OTP, NONVOLATILE SECTOR LOCK, and ERASE NONVOLATILE SECTOR LOCK ARRAY operations are in progress | 30 | – | – | µs |
| | | While a WRITE STATUS REGISTER operation is in progress | – | $^{t}$W | – | ms |
| | | While a WRITE NONVOLATILE CONFIGURATION REGISTER operation is in progress | – | $^{t}$WNVCR | – | ms |
| | | On completion or suspension of a SUBSECTOR ERASE operation | – | $^{t}$SSE | – | s |
| | | Device in deep power-down mode | – | $^{t}$RDP | – | ms |
| | | While ADVANCED SECTOR PROTECTION PROGRAM operation is in progress | – | $^{t}$ASPP | – | ms |
| | | While PASSWORD PROTECTION PROGRAM operation is in progress | – | $^{t}$PASSP | – | ms |
| Software reset recovery time | $^{t}$SHSL3 | Device deselected (S# HIGH) and is in standby mode | 40 | – | – | ns |
| | | Any Flash array PROGRAM/ERASE/SUSPEND/RESUME, PROGRAM OTP, NONVOLATILE SECTOR LOCK, and ERASE NONVOLATILE SECTOR LOCK ARRAY operations are in progress | 30 | – | – | µs |
| | | While WRITE STATUS REGISTER operation is in progress | – | $^{t}$W | – | ms |
| | | While a WRITE NONVOLATILE CONFIGURATION REGISTER operation is in progress | – | $^{t}$WNVCR | – | ms |
| | | On completion or suspension of a SUBSECTOR ERASE operation | – | $^{t}$SSE | – | s |
| | | Device in deep power-down mode | – | $^{t}$RDP | – | ms |
| | | While ADVANCED SECTOR PROTECTION PROGRAM operation is in progress | – | $^{t}$ASPP | – | ms |
| | | While PASSWORD PROTECTION PROGRAM operation is in progress | – | $^{t}$PASSP | – | ms |

Notes:
1. Values are guaranteed by characterization; not 100% tested.
2. The device reset is possible but not guaranteed if $^{t}$RLRH < 50ns.

**Figure 45: Reset AC Timing During PROGRAM and ERASE Cycle**

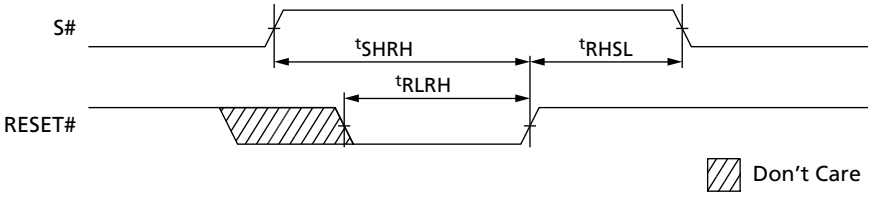



**Figure 46: Reset Enable and Reset Memory Timing**

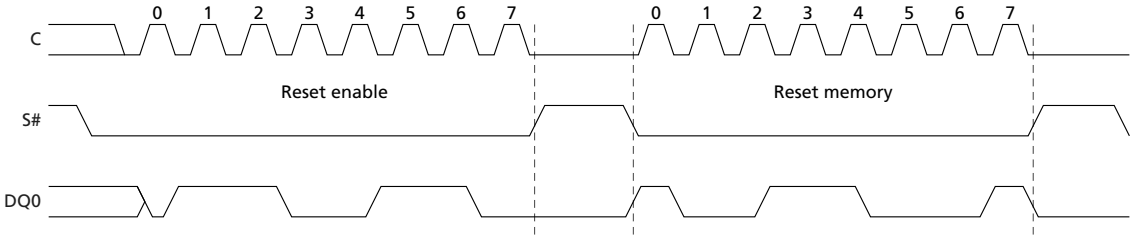



**Figure 47: Serial Input Timing STR**

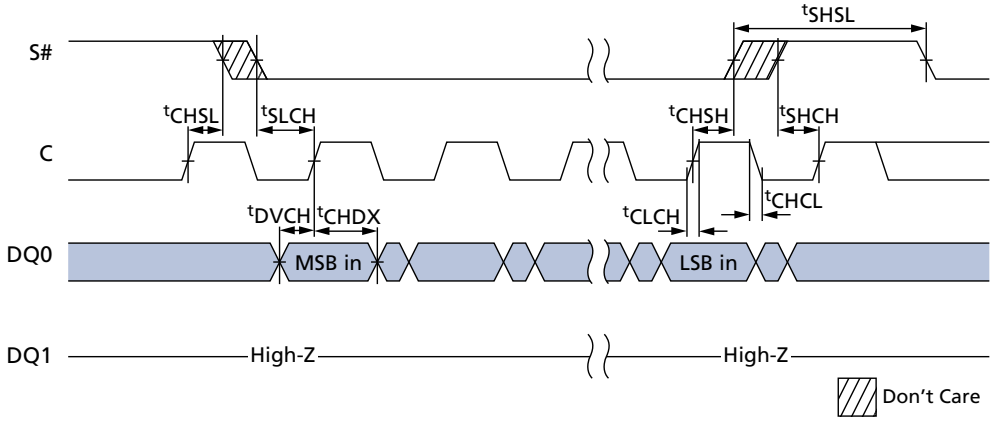


**Figure 48: Serial Input Timing DTR**

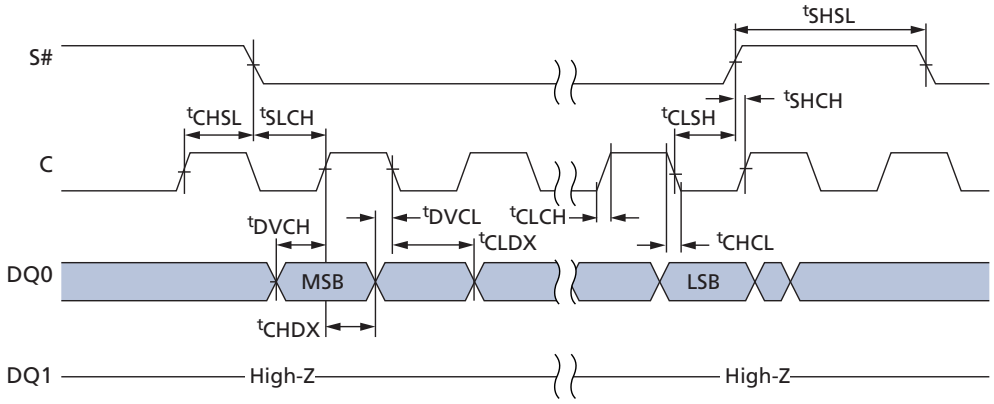



**Figure 49: Write Protect Setup and Hold During WRITE STATUS REGISTER Operation (SRWD = 1)**

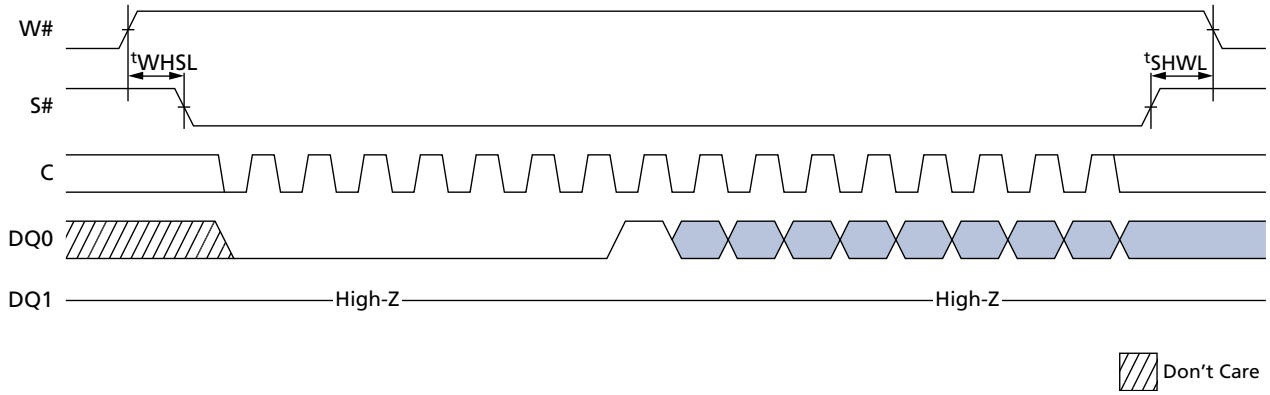



**Figure 50: Hold Timing**

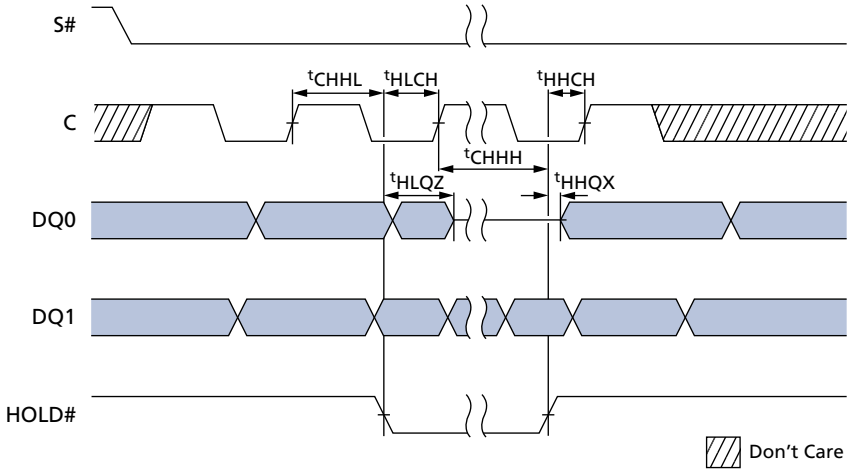


**Figure 51: Output Timing for STR**

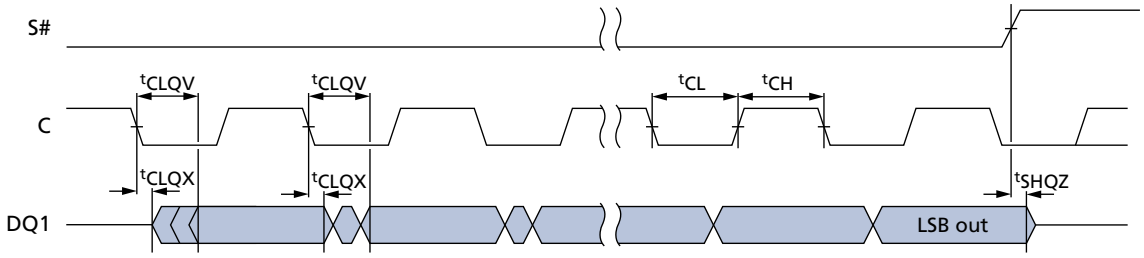
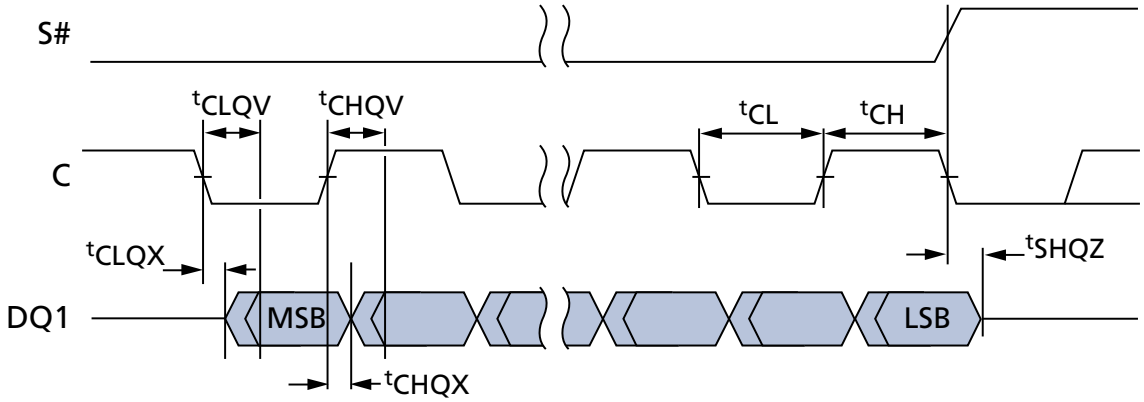



**Figure 52: Output Timing for DTR**



# Program/Erase Specifications

**Table 52: Program/Erase Specifications**

| Parameter | Condition | Typ | Max | Units | Notes |
|---|---|---|---|---|---|
| Erase to suspend | Sector erase or erase resume to erase suspend | 150 | – | µs | 1 |
| Program to suspend | Program resume to program suspend | 5 | – | µs | 1 |
| Subsector erase to suspend | Subsector erase or subsector erase resume to erase suspend | 50 | – | µs | 1 |
| Suspend latency | Program | 7 | 25 | µs | 2 |
| Suspend latency | Subsector erase | 15 | 30 | µs | 2 |
| Suspend latency | Erase | 15 | 30 | µs | 3 |

Notes:
1. Timing is not internally controlled.
2. Any READ command accepted.
3. Any command except the following are accepted: SECTOR, SUBSECTOR, or DIE ERASE; WRITE STATUS REGISTER; WRITE NONVOLATILE CONFIGURATION REGISTER; and PROGRAM OTP.

# Revision History

## Rev. F – 07/18

- Added Clock Frequencies tables for UT part (STR) in Volatile Configuration Register
- Updated DC Current Characteristics and Operating Conditions table: Added $I_{cc1}$ and $I_{cc2}$ values for UT range
- Added Supported Maximum Frequency table in AC Characteristics and Operating Conditions

## Rev. E - 12/17

- Added Important Notes and Warnings section for further clarification aligning to industry standards
- Added UT device (operating temperature range: from –40°C to +125°C)
- Added DEEP POWER-DOWN Operations
- Added Active Power, Standby Power, and Deep Power-Down modes
- Added figure for Serial Input Timing DTR

## Rev. D - 06/16

- Updated Max DTR frequency to 90MHz
- Add READ GENERAL PURPOSE READ REGISTER 96h command and associated information
- Added general purpose read register notes to Command Definitions table
- Add initial delivery status

## Rev. C - 04/16

- Datasheet version from preliminary to production

## Rev. B - 9/15

- Revised wrap table
- Revised supported clock frequencies DTR
- Change bit 3 setting of Enhanced Volatile Configuration Register from 0 to 1
- Revised AC table

## Rev. A – 08/15

- Initial release

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-4000
www.micron.com/products/support Sales inquiries: 800-932-4992

# X-ON Electronics

Largest Supplier of Electrical and Electronic Components

***Click to view similar products for*** NOR Flash ***category:***

***Click to view products by*** Micron ***manufacturer:***

Other Similar products are found below :

AM29BL162CB-80RDPE1 AM29F800BT-90DPE 2 S29AL016J70BFI012 S29AL016J70FFI022 S29GL064N90BFI032 S29GL064N90BFI042 S29GL064N90FFI012 S29GL128P10FFI022 S29GL256P11FFIV12 S29GL256P11FFIV22 S29GL512S10GHI020 S70FL01GSAGMFV011 S99-50389 P AM29F016D-120DPI 1 AT25DF011-SSHN-T AT25DF011-XMHN-T AT25DF041B-MHN-Y AT25DN256-SSHF-T AT25DN512C-MAHF-T AT45DB161E-CCUD-T S29AL008J55BFIR22 S29GL032N90TFI033 S29GL032N90TFI043 S29GL256P11FFI012 S29JL064J55TFI003 S70FL01GSAGMFI013 S99-50052 S29AL008J55BFIR23 M29W128GL70ZS3F PC28F128J3F75B S29GL128P10TFI013 S29GL064N90FFI022 S29GL032N11FFIS12 S29AL016D90DGI027 S29JL032J70TFI423 S29GL128S10TFIV13 S29AL016J70WEI029 S25FL132K0XMFB043 AT25DF041B-SSHNHR-T S25FL116K0XNFB010 S25FL132K0XBHA020 AT25SL321-MBUE-T SST39VF800A-70-4I-MAQE LE25S40MB-AH S25FL116K0XNFA013 SST39VF400A-70-4C-MAQE AT25DF512C-XMHN-T S25FL256LAGMFI001 S26KL128SDABHN020 520366231286