THE DATASHEET OF

M25PX32SOVZM6F

M25PX32 NOR Serial Flash Embedded Memory

32Mb, Dual I/O, 4KB Subsector Erase, 3V Serial Flash Memory with 75 MHz SPI Bus Interface

Features

- SPI bus compatible serial interface
- 75 MHz (maximum) clock frequency
- 2.7V to 3.6V single supply voltage
- Dual input/output commands resulting in an equivalent clock frequency of 150 MHz
  - DUAL OUTPUT FAST READ command
  - DUAL INPUT FAST PROGRAM command
- Continuous READ of entire memory using FAST READ or DUAL OUTPUT FAST READ command
- 32Mb Flash memory
  - Uniform 4KB subsectors
  - Uniform 64KB sectors
- Additional 64-byte user-lockable, one-time programmable (OTP) area
- Erase capability
  - Subsector (4KB granularity)
  - Sector (64KB granularity)
  - Bulk erase (32Mb) in 34 s typical
- Write protections
  - Software write protection: applicable to every 64KB sector (volatile lock bit)
  - Hardware write protection: non-volatile bits BP0, BP1, BP2 define protected area size
- Deep power down: 5µA typical
- Electronic signature
  - JEDEC standard 2-byte signature (7116h)
  - Unique ID code (UID) with 16-byte read-only space, available upon request
- More than 100,000 write cycles per sector
- More than 20 years data retention
- Packages (RoHS compliant)
  - VFQFPN8 (MP) 6mm x 5mm (MLP8)
  - SO8W (MW) 208 mils width
  - SO16 (MF) 300 mils width
  - TBGA24 (ZM) 6mm x 8mm
- Automotive certified parts available
- This latest generation of this product line is available in the N25Q device (see TN 12-13 for migration guide)

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.

Functional Description

The M25PX32 is a 32Mb (4Mb x 8) serial Flash memory, with advanced write protection mechanisms, accessed by a high speed SPI-compatible bus. The device supports two high-performance dual input/output instructions that double the transfer bandwidth for read and program operations:

- DUAL OUTPUT FAST READ instruction reads data at up to 75 MHz by using both pin DQ1 and pin DQ0 as outputs.
- DUAL INPUT FAST PROGRAM instruction programs data at up to 75 MHz by using both pin DQ1 and pin DQ0 as inputs.

The memory can be programmed 1 to 256 bytes at a time, using the PAGE PROGRAM instruction. It is organized as 64 sectors that are further divided into 16 subsectors each (1024 total subsectors).

The memory can be erased a 4KB subsector at a time, a 64KB sector at a time, or as a whole. It can be write protected by software using a mix of volatile and non-volatile protection features, depending on the application needs. Protection granularity is 64KB sectors.

The device has 64 one-time-programmable bytes (OTP bytes) that can be read and programmed using two dedicated instructions, READ OTP and PROGRAM OTP, respectively. These 64 bytes can be locked permanently using a PROGRAM OTP command sequence.

Further features are available as additional security options. More information on these security features is available, upon completion of an NDA (nondisclosure agreement), and are, therefore, not described in this datasheet. For more details of this option contact your nearest Micron sales office.

Figure 1: Logic Diagram

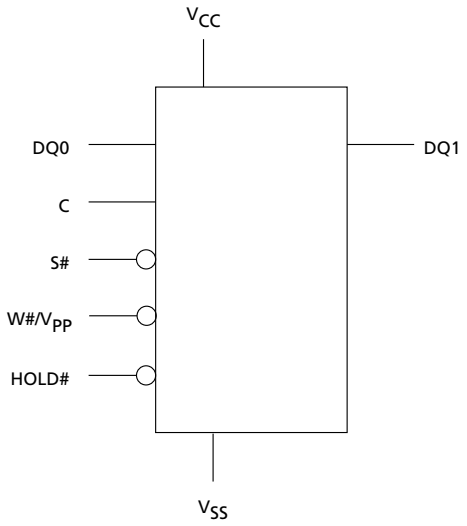
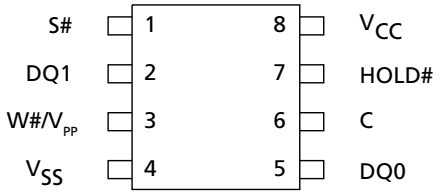


Table 1: Signal Names

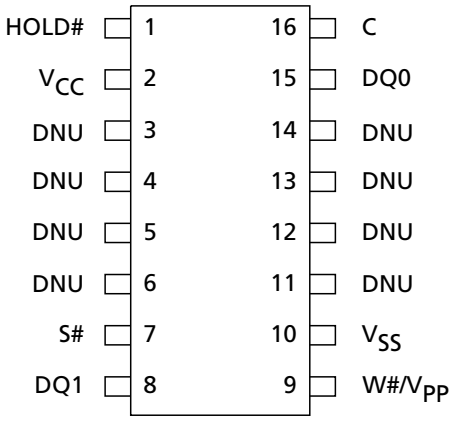
Signal Name	Function	Direction
C	Serial clock	Input
DQ0	Serial data input (Serves as output during DUAL OUTPUT FAST READ operation)	I/O
DQ1	Serial data output (Serves as input during DUAL INPUT FAST PROGRAM operation)	I/O
S#	Chip select	Input
W#/V_{PP}	Write protect or enhanced program supply voltage	Input
HOLD#	Hold	Input
V_{CC}	Supply voltage	
V_{SS}	Ground	

Figure 2: Pin Connections: VFQFPN and SO8



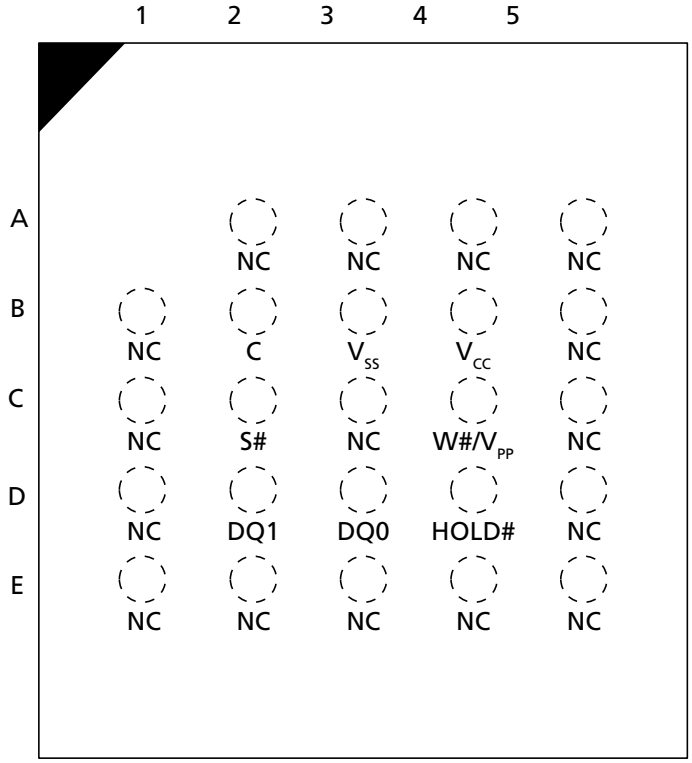
There is an exposed central pad on the underside of the VFQFPN package. This is pulled internally to V_{SS}, and must not be connected to any other voltage or signal line on the PCB. The Package Mechanical section provides information on package dimensions and how to identify pin 1.

Figure 3: Pin Connections: SO16



Note: 1. DNU = do not use.

Figure 4: Pinout: 24-Ball BGA, 6x8mm



Note: 1. DNU = do not use. NC = no connect.

Signal Descriptions

Table 2: Signal Descriptions

Signal	Type	Description
DQ1	Output	**Serial data:** The DQ1 output signal is used to transfer data serially out of the device. Data is shifted out on the falling edge of the serial clock (C). During the DUAL INPUT FAST PROGRAM command, pin DQ1 is used as an input. It is latched on the rising edge of C.
DQ0	Input	**Serial data:** The DQ0 input signal is used to transfer data serially into the device. It receives commands, addresses, and the data to be programmed. Values are latched on the rising edge of the serial clock (C). During the DUAL OUTPUT FAST READ command, pin DQ0 is used as an output. Data is shifted out on the falling edge of C.
C	Input	**Clock:** The C input signal provides the timing of the serial interface. Commands, addresses, or data present at serial data input (DQ0) is latched on the rising edge of the serial clock (C). Data on DQ1 changes after the falling edge of C.
S#	Input	**Chip select:** When the S# input signal is HIGH, the device is deselected and DQ1 is at high impedance. Unless an internal PROGRAM, ERASE, or WRITE STATUS REGISTER cycle is in progress, the device will be in the standby power mode (not the deep power-down mode). Driving S# LOW enables the device, placing it in the active power mode. After power-up, a falling edge on S# is required prior to the start of any command.
HOLD#	Input	**Hold:** The HOLD# signal is used to pause any serial communications with the device without deselecting the device. During the hold condition, DQ1 is High-Z. DQ0 and C are "Don't Care." To start the hold condition, the device must be selected, with S# driven LOW.
W#/V_{PP}	Input	**Write protect/enhanced program supply voltage:** The W#/V_{PP} signal is both a control input and a power supply pin. The two functions are selected by the voltage range applied to the pin. If the W#/V_{PP} input is kept in a low voltage range (0 V to V_{CC}) the pin is seen as a control input. The W# input signal is used to freeze the size of the area of memory that is protected against program or erase commands as specified by the values in BP2, BP1, and BP0 bits of the Status Register. V_{PP} acts as an additional power supply if it is in the range of V_{PPH}, as defined in the AC Measurement Conditions table. Avoid applying V_{PPH} to the W#/V_{PP} pin during a BULK ERASE operation.
V_{CC}	Power	**Device core power supply:** Source voltage.
V_{SS}	Ground	**Ground:** Reference for the V_{CC} supply voltage.
DNU	–	Do not use.

Serial Peripheral Interface Modes

The device can be driven by a microcontroller while its serial peripheral interface (SPI) is in either of the two modes shown here. The difference between the two modes is the clock polarity when the bus master is in standby mode and not transferring data. Input data is latched in on the rising edge of the clock, and output data is available from the falling edge of the clock.

Table 3: SPI Modes

SPI Modes	Clock Polarity
CPOL = 0, CPHA = 0	C remains at 0 for (CPOL = 0, CPHA = 0)
CPOL = 1, CPHA = 1	C remains at 1 for (CPOL = 1, CPHA = 1)

The following figure is an example of three memory devices in a simple connection to an MCU on an SPI bus. Because only one device is selected at a time, that one device drives DQ1, while the other devices are HIGH-Z.

Resistors ensure the device is not selected if the bus master leaves S# HIGH-Z. The bus master might enter a state in which all input/output is HIGH-Z simultaneously, such as when the bus master is reset. Therefore, the serial clock must be connected to an external pull-down resistor so that S# is pulled HIGH while the serial clock is pulled LOW. This ensures that S# and the serial clock are not HIGH simultaneously and that t_{SHCH} is met. The typical resistor value of 100kΩ, assuming that the time constant R × Cp (Cp = parasitic capacitance of the bus line), is shorter than the time the bus master leaves the SPI bus in HIGH-Z.

Example: Cp = 50 pF, that is R × Cp = 5µs. The application must ensure that the bus master never leaves the SPI bus HIGH-Z for a time period shorter than 5µs. W# and HOLD# should be driven either HIGH or LOW, as appropriate.

Figure 5: Bus Master and Memory Devices on the SPI Bus

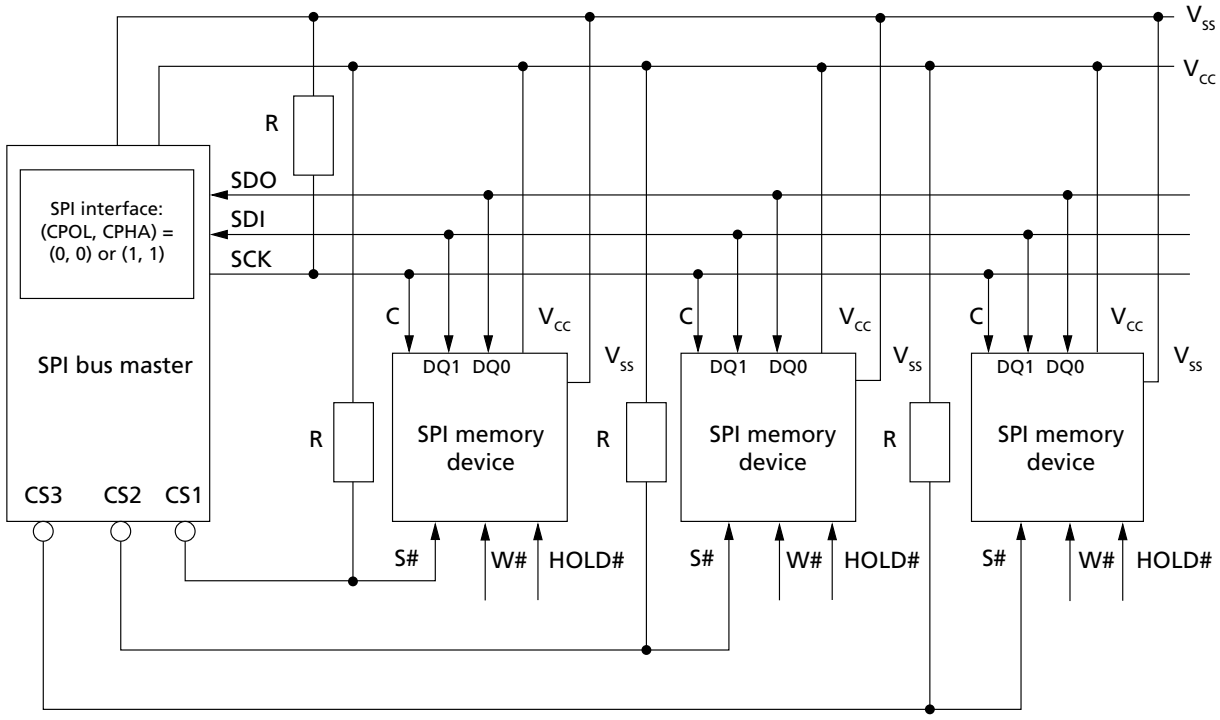
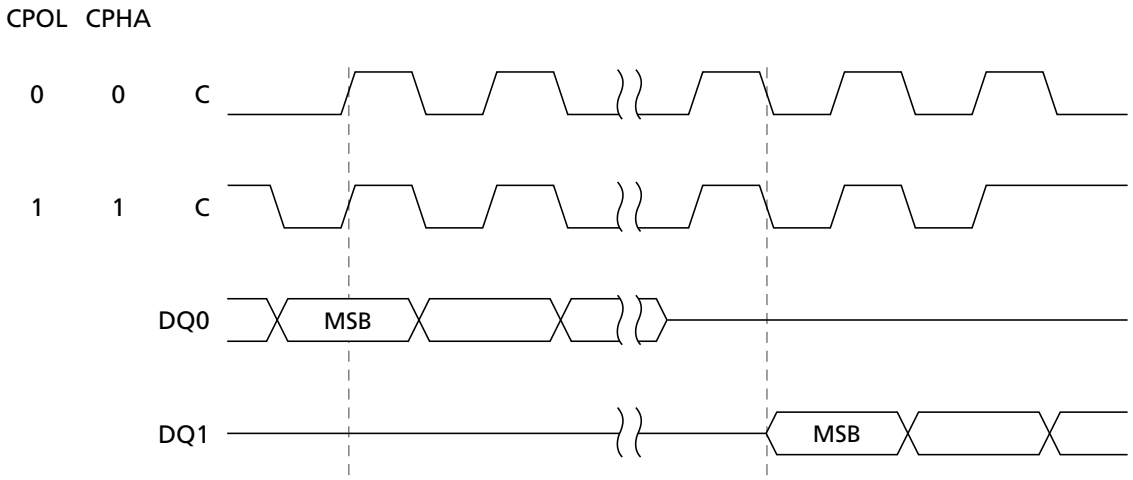


Figure 6: SPI Modes



Operating Features

Page Programming

To program one data byte, two commands are required: WRITE ENABLE, which is one byte, and a PAGE PROGRAM sequence, which is four bytes plus data. This is followed by the internal PROGRAM cycle of duration t_{PP}. To spread this overhead, the PAGE PROGRAM command allows up to 256 bytes to be programmed at a time (changing bits from 1 to 0), provided they lie in consecutive addresses on the same page of memory. To optimize timings, it is recommended to use the PAGE PROGRAM command to program all consecutive targeted bytes in a single sequence than to use several PAGE PROGRAM sequences with each containing only a few bytes.

Dual Input Fast Program

The DUAL INPUT FAST PROGRAM command makes it possible to program up to 256 bytes using two input pins at the same time (by changing bits from 1 to 0). For optimized timings, it is recommended to use the DUAL INPUT FAST PROGRAM command to program all consecutive targeted bytes in a single sequence than to use several DUAL INPUT FAST PROGRAM sequences each containing only a few bytes.

Subsector Erase, Sector Erase, Bulk Erase

The PAGE PROGRAM command allows bits to be reset from 1 to 0. Before this can be applied, the bytes of memory need to have been erased to all 1s (FFh). This can be achieved a subsector at a time using the SUBSECTOR ERASE command, a sector at a time using the SECTOR ERASE command, or throughout the entire memory using the BULK ERASE command. This starts an internal ERASE cycle of duration t_{SSE}, t_{SE} or t_{BE}. The ERASE command must be preceded by a WRITE ENABLE command.

Polling during a Write, Program, or Erase Cycle

An improvement in the time to complete the following commands can be achieved by not waiting for the worst case delay (t_W, t_{PP}, t_{SSE}, t_{SE}, or t_{BE}).

- WRITE STATUS REGISTER
- PROGRAM OTP
- PROGRAM
- DUAL INPUT FAST PROGRAM
- ERASE (SUBSECTOR ERASE, SECTOR ERASE, BULK ERASE)

The write in progress (WIP) bit is provided in the status register so that the application program can monitor this bit in the status register, polling it to establish when the previous WRITE cycle, PROGRAM cycle, or ERASE cycle is complete.

Active Power, Standby Power, and Deep Power-Down

When chip select (S#) is LOW, the device is selected, and in the ACTIVE POWER mode. When S# is HIGH, the device is deselected, but could remain in the ACTIVE POWER mode until all internal cycles have completed (PROGRAM, ERASE, WRITE STATUS REGISTER). The device then goes in to the STANDBY POWER mode. The device consumption drops to I_{CC1}.

The DEEP POWER-DOWN mode is entered when the DEEP POWER-DOWN command is executed. The device consumption drops further to I_{CC2}. The device remains in this mode until the RELEASE FROM DEEP POWER-DOWN command is executed. While in the DEEP POWER-DOWN mode, the device ignores all WRITE, PROGRAM, and ERASE commands. This provides an extra software protection mechanism when the device is not in active use, by protecting the device from inadvertent WRITE, PROGRAM, or ERASE operations. For further information, see DEEP POWER-DOWN (page 45).

Status Register

The status register contains a number of status and control bits that can be read or set (as appropriate) by specific commands. For a detailed description of the status register bits, see READ STATUS REGISTER.

Data Protection by Protocol

Non-volatile memory is used in environments that can include excessive noise. The following capabilities help protect data in these noisy environments.

Power on reset and an internal timer (t_{PUW}) can provide protection against inadvertent changes while the power supply is outside the operating specification.

WRITE, PROGRAM, ERASE, and WRITE STATUS REGISTER commands are checked before they are accepted for execution to ensure they consist of a number of clock pulses that is a multiple of eight.

All commands that modify data must be preceded by a WRITE ENABLE command to set the write enable latch (WEL) bit.

In addition to the low power consumption feature, the DEEP POWER-DOWN mode offers extra software protection since all WRITE, PROGRAM, and ERASE commands are ignored when the device is in this mode.

Software Data Protection

Memory can be configured as read-only using the top/bottom bit and the block protect bits (BP2, BP1, BP0) as shown in the Protected Area Sizes table.

Memory sectors can be protected by specific lock registers assigned to each 64KB sector. These lock registers can be read and written using the READ LOCK REGISTER and WRITE to LOCK REGISTER commands. In each lock register the following two bits control the protection of each sector:

- Write lock bit: This bit determines whether the contents of the sector can be modified using the WRITE, PROGRAM, and ERASE commands. When the bit is set to ‘1’, the sector is write protected, and any operations that attempt to change the data in the sector will fail. When the bit is reset to ‘0’, the sector is not write protected by the lock register, and may be modified.
- Lock down bit: This bit provides a mechanism for protecting software data from simple hacking and malicious attack. When the bit is set to '1’, further modification to the write lock bit and lock down bit cannot be performed. A power-up, is required before changes to these bits can be made. When the bit is reset to ‘0’, the write lock bit and lock down bit can be changed.

The software protection truth table shows the lock down bit and write lock bit settings and the sector protection status.

Table 4: Software Protection Truth Table

Sector Lock Register Bits		
Lock Down	**Write Lock**	**Protection Status**
0	0	Sector unprotected from PROGRAM / ERASE / WRITE operations; protection status reversible
0	1	Sector protected from PROGRAM / ERASE / WRITE operations; protection status reversible
1	0	Sector unprotected from PROGRAM / ERASE / WRITE operations; protection status cannot be changed except by a power-up.
1	1	Sector protected from PROGRAM / ERASE / WRITE operations; protection status cannot be changed except by a power-up.

Hardware Data Protection

Hardware data protection is implemented using the write protect signal applied on the W#/V_{PP} pin. This freezes the status register in a read-only mode, protecting the block protect (BP) bits and the status register write disable bit (SRWD). The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 5: Sectors 0 to 16, Protected Area Sizes – Upper Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	**BP2**	**BP1**	**BP0**	**Protected Area**	**Unprotected Area**
0	0	0	0	None	All sectors [1]
0	0	0	1	Upper 16th (sector 15)	Lower 15/16ths (sectors 0 to 14)
0	0	1	0	Upper 8th (sectors 14 to 15)	Lower 7/8ths (sectors 0 to 13)
0	0	1	1	Upper 4th (sectors 12 to 15)	Lower 3/4ths (sectors 0 to 11)
0	1	0	0	Upper half (sectors 8 to 15)	Lower half (sectors 0 to 7)
0	1	0	1	All sectors	None
0	1	1	0	All sectors	None
0	1	1	1	All sectors	None

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 6: Sectors 0 to 16, Protected Area Sizes – Lower Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	**BP2**	**BP1**	**BP0**	**Protected Area**	**Unprotected Area**
1	0	0	0	None	All sectors [1]
1	0	0	1	Lower 16th (sector 0)	Upper 15/16ths (sectors 1 to 15)
1	0	1	0	Lower 8th (sectors 0 to 1)	Upper 7/8ths (sectors 2 to 15)
1	0	1	1	Lower 4th (sectors 0 to 3)	Upper 3/4ths (sectors 4 to 15)
1	1	0	0	Lower half (sectors 3 to 7)	Upper half (sectors 8 to 15)
1	1	0	1	All sectors	None
1	1	1	0	All sectors	None

Table 6: Sectors 0 to 16, Protected Area Sizes – Lower Area Protection (Continued)

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
1	1	1	1	All sectors	None

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 7: Sectors 0 to 32, Protected Area Sizes – Upper Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	BP 2	BP 1	BP 0	Protected Area	Unprotected Area
0	0	0	0	none	All sectors [1]
0	0	0	1	Upper 32nd (sector 31)	Lower 31/32nds (sectors 0 to 30)
0	0	1	0	Upper 16th (sectors 30 to 31)	Lower 15/16ths (sectors 0 to 29)
0	0	1	1	Upper 8th (sectors 28 to 31)	Lower 7/8ths (sectors 0 to 27)
0	1	0	0	Upper 4th (sectors 24 to 31)	Lower 3/4ths (sectors 0 to 23)
0	1	0	1	Upper half (sectors 16 to 31)	Lower half (sectors 0 to 15)
0	1	1	0	All sectors	none
0	1	1	1	All sectors	none

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 8: Sectors 0 to 32, Protected Area Sizes – Lower Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	BP 2	BP 1	BP 0	Protected Area	Unprotected Area
1	0	0	0	none	All sectors [1]
1	0	0	1	Lower 32nd (sector 0)	Upper 31/32nds (sectors 1 to 31)
1	0	1	0	Lower 16th (sectors 0 to 1)	Upper 15/16ths (sectors 2 to 31)
1	0	1	1	Lower 8th (sectors 0 to 3)	Upper 7/8ths (sectors 4 to 31)
1	1	0	0	Lower 4th (sectors 0 to 7)	Upper 3/4ths (sectors 8 to 31)
1	1	0	1	Lower half (sectors 0 to 15)	Upper half (sectors 16 to 31)
1	1	1	0	All sectors	none
1	1	1	1	All sectors	none

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 9: Sectors 0 to 63, Protected Area Sizes – Upper Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
0	0	0	0	None	All sectors [1]
0	0	0	1	Upper 64th (sector 63)	Lower 63/64ths (sectors 0 to 62)

Table 9: Sectors 0 to 63, Protected Area Sizes – Upper Area Protection (Continued)

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
0	0	1	0	Upper 32th (sectors 62 to 63)	Lower 31/32nds (sectors 0 to 61)
0	0	1	1	Upper 16th (sectors 60 to 63)	Lower 15/16ths (sectors 0 to 59)
0	1	0	0	Upper 8th (sectors 56 to 63)	Lower 7/8ths (sectors 0 to 55)
0	1	0	1	Upper 4th (sectors 48 to 63)	Lower 3/4ths (sectors 0 to 47)
0	1	1	0	Upper half (sectors 32 to 63)	Lower half (sectors 0 to 31)
0	1	1	1	All sectors	None

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 10: Sectors 0 to 63, Protected Area Sizes – Lower Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
1	0	0	0	None	All sectors [1]
1	0	0	1	Lower 64th (sector 0)	Upper 63/64ths (sectors 1 to 63)
1	0	1	0	Lower 32th (sectors 0 to 1)	Upper 31/32nds (sectors 2 to 63)
1	0	1	1	Lower 16th (sectors 0 to 3)	Upper 15/16ths (sectors 4 to 63)
1	1	0	0	Lower 8th (sectors 0 to 7)	Upper 7/8ths (sectors 8 to 63)
1	1	0	1	Lower 4th (sectors 0 to 15)	Upper 3/4ths (sectors 16 to 63)
1	1	1	0	Lower half (sectors 0 to 31)	Upper half (sectors 32 to 63)
1	1	1	1	All sectors	None

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 11: Sectors 0 to 127, Protected Area Sizes – Upper Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
0	0	0	0	None	All sectors [1]
0	0	0	1	Upper 64th (sectors 126 to 127)	Lower 63/64ths (sectors 0 to 125)
0	0	1	0	Upper 32nd (sectors 124 to 127)	Lower 31/32nds (sectors 0 to 123)
0	0	1	1	Upper 16th (sectors 120 to 127)	Lower 15/16ths (sectors 0 to119)
0	1	0	0	Upper 8th (sectors 112 to 127)	Lower 7/8ths (sectors 0 to 111)
0	1	0	1	Upper 4th (sectors 96 to 127)	Lower 3/4ths (sectors 0 to 95)
0	1	1	0	Upper half (sectors 64 to 127)	Lower half (sectors 0 to 63)
0	1	1	1	All sectors	None

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

Table 12: Sectors 0 to 127, Protected Area Sizes – Lower Area Protection

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
1	0	0	0	None	All sectors [1]
1	0	0	1	Lower 64th (sectors 0 to 1)	Upper (sectors 2 to 127)
1	0	1	0	Lower 32nd (sectors 0 to 3)	Upper (sectors 4 to 127)
1	0	1	1	Lower 16th (sectors 0 to 7)	Upper (sectors 8 to 127)
1	1	0	0	Lower 8th (sectors 0 to 15)	Upper (sectors 16 to 127)
1	1	0	1	Lower 4th (sectors 0 to 31)	Upper (sectors 32 to 127)
1	1	1	0	Lower half (sectors 0 to 63)	Upper (sectors 64 to 127)

Table 12: Sectors 0 to 127, Protected Area Sizes – Lower Area Protection (Continued)

Status Register Content				Memory Content	
Top/Bottom Bit	BP2	BP1	BP0	Protected Area	Unprotected Area
1	1	1	1	None	All sectors

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits are 0.

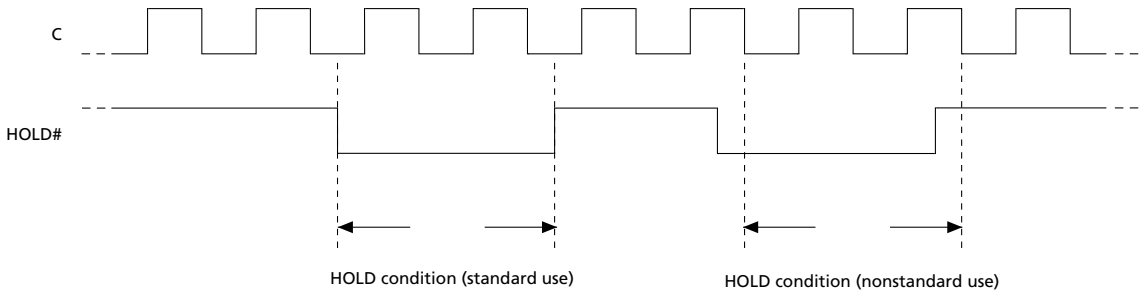
Hold Condition

The HOLD# signal is used to pause any serial communications with the device without resetting the clocking sequence. However, taking this signal LOW does not terminate any WRITE STATUS REGISTER, PROGRAM, or ERASE cycle that is currently in progress.

To enter the hold condition, the device must be selected, with S# LOW. The hold condition starts on the falling edge of the HOLD# signal, if this coincides with serial clock (C) being LOW. The hold condition ends on the rising edge of the HOLD# signal, if this coincides with C being LOW. If the falling edge does not coincide with C being LOW, the hold condition starts after C next goes LOW. Similarly, if the rising edge does not coincide with C being LOW, the hold condition ends after C next goes LOW.

During the hold condition, DQ1 is HIGH impedance while DQ0 and C are Don't Care. Typically, the device remains selected with S# driven LOW for the duration of the hold condition. This ensures that the state of the internal logic remains unchanged from the moment of entering the hold condition. If S# goes HIGH while the device is in the hold condition, the internal logic of the device is reset. To restart communication with the device, it is necessary to drive HOLD# HIGH, and then to drive S# LOW. This prevents the device from going back to the hold condition.

Figure 7: Hold Condition Activation

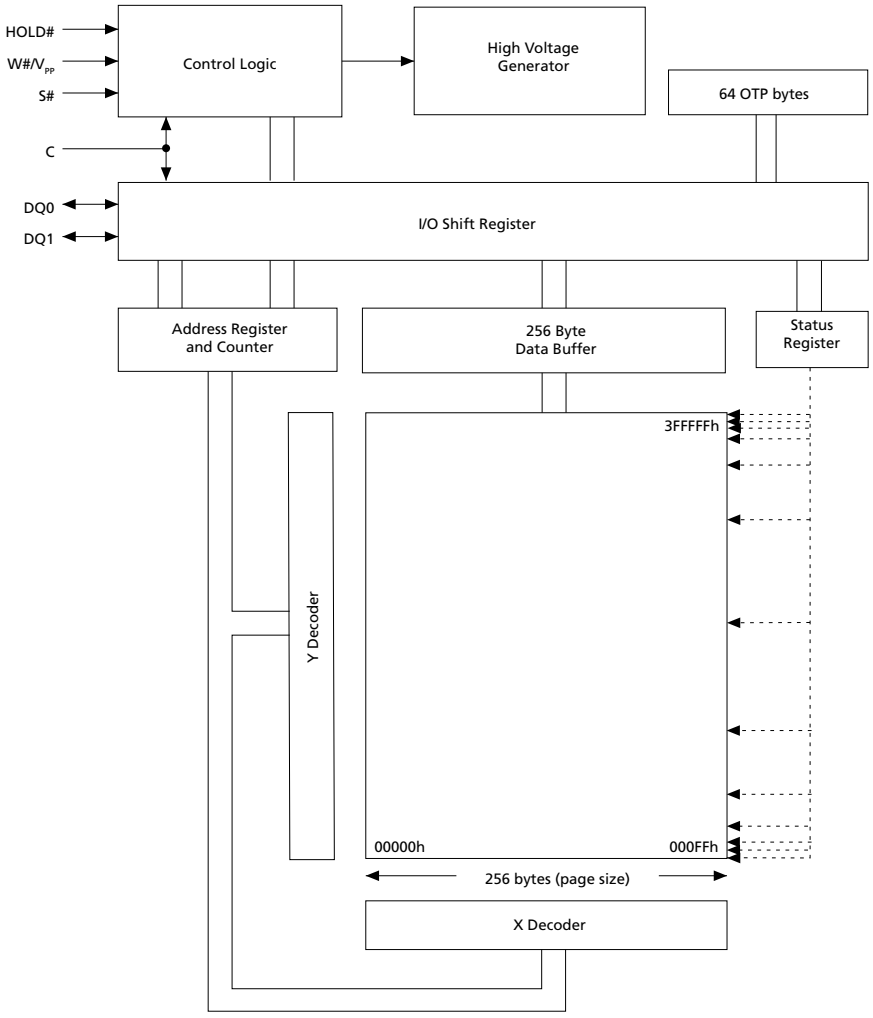


Memory Configuration and Block Diagram

Each page of memory can be individually programmed; bits are programmed from 1 to 0. The device is sector or bulk-erasable, but not page-erasable; bits are erased from 0 to 1. The memory is configured as follows:

- 4,194,304 bytes (8 bits each)
- 1024 subsectors (4KB each)
- 64 sectors (64KB each)
- 16,384 pages (256 bytes each)
- 64 OTP bytes located outside main memory

Figure 8: Block Diagram



Memory Map – 32Mb Density

Table 13: Sectors[127:0]

Sector	Subsector	Address Range	
		Start	End
63	1023	003F F000h	003F FFFFh
	⋮	⋮	⋮
	1008	003F 0000h	003F 0FFFh
⋮	⋮	⋮	⋮
0	15	0000 F000h	0000 FFFFh
	⋮	⋮	⋮
	0	0000 0000h	0000 0FFFh

Command Set Overview

All commands, addresses, and data are shifted in and out of the device, most significant bit first.

Serial data inputs DQ0 and DQ1 are sampled on the first rising edge of serial clock (C) after chip select (S#) is driven LOW. Then, the one-byte command code must be shifted in to the device, most significant bit first, on DQ0 and DQ1, each bit being latched on the rising edges of C.

Every command sequence starts with a one-byte command code. Depending on the command, this command code might be followed by address or data bytes, by address and data bytes, or by neither address or data bytes. For the following commands, the shifted-in command sequence is followed by a data-out sequence. S# can be driven HIGH after any bit of the data-out sequence is being shifted out.

- READ DATA BYTES (READ)
- READ DATA BYTES at HIGHER SPEED
- DUAL OUTPUT FAST READ
- READ OTP
- READ LOCK REGISTERS
- READ STATUS REGISTER
- READ IDENTIFICATION
- RELEASE from DEEP POWER-DOWN

For the following commands, S# must be driven HIGH exactly at a byte boundary. That is, after an exact multiple of eight clock pulses following S# being driven LOW, S# must be driven HIGH. Otherwise, the command is rejected and not executed.

- PAGE PROGRAM
- PROGRAM OTP
- DUAL INPUT FAST PROGRAM
- SUBSECTOR ERASE
- SECTOR ERASE
- BULK ERASE
- WRITE STATUS REGISTER
- WRITE to LOCK REGISTER
- WRITE ENABLE
- WRITE DISABLE
- DEEP POWER-DOWN

All attempts to access the memory array are ignored during a WRITE STATUS REGISTER command cycle, a PROGRAM command cycle, or an ERASE command cycle. In addition, the internal cycle for each of these commands continues unaffected.

Table 14: Command Set Codes

Command Name	One-Byte Command Code		Bytes		
			Address	Dummy	Data
WRITE ENABLE	0000 0110	06h	0	0	0
WRITE DISABLE	0000 0100	04h	0	0	0
READ IDENTIFICATION	1001 1111	9Fh	0	0	1 to 20
	1001 1110	9Eh			1 to 20
READ STATUS REGISTER	0000 0101	05h	0	0	1 to ∞
WRITE STATUS REGISTER	0000 0001	01h	0	0	1
WRITE to LOCK REGISTER	1110 0101	E5h	3	0	1
READ LOCK REGISTER	1110 1000	E8h	3	0	1
READ DATA BYTES	0000 0011	03h	3	0	1 to ∞
READ DATA BYTES at HIGHER SPEED	0000 1011	0Bh	3	1	1 to ∞
DUAL OUTPUT FAST READ	0011 1011	3Bh	3	1	1 to ∞
READ OTP (Read 64 bytes of OTP area)	0100 1011	4Bh	3	1	1 to 65
PROGRAM OTP (Program 64 bytes of OTP area)	0100 0010	42h	3	0	1 to 65
PAGE PROGRAM	0000 0010	02h	3	0	1 to 256
DUAL INPUT FAST PROGRAM	1010 0010	A2h	3	0	1 to 256
SUBSECTOR ERASE	0010 0000	20h	3	0	0
SECTOR ERASE	1101 1000	D8h	3	0	0
BULK ERASE	1100 0111	C7h	0	0	0
DEEP POWER-DOWN	1011 1001	B9h	0	0	0
RELEASE from DEEP POWER-DOWN	1010 1011	ABh	0	0	0

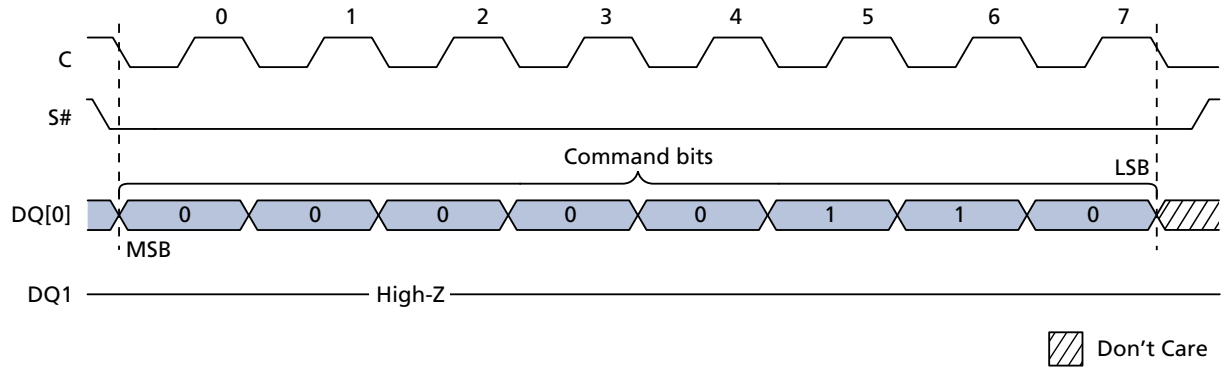
WRITE ENABLE

The WRITE ENABLE command sets the write enable latch (WEL) bit.

The WEL bit must be set before execution of every PROGRAM, ERASE, and WRITE command.

The WRITE ENABLE command is entered by driving chip select (S#) LOW, sending the command code, and then driving S# HIGH.

Figure 9: WRITE ENABLE Command Sequence



WRITE DISABLE

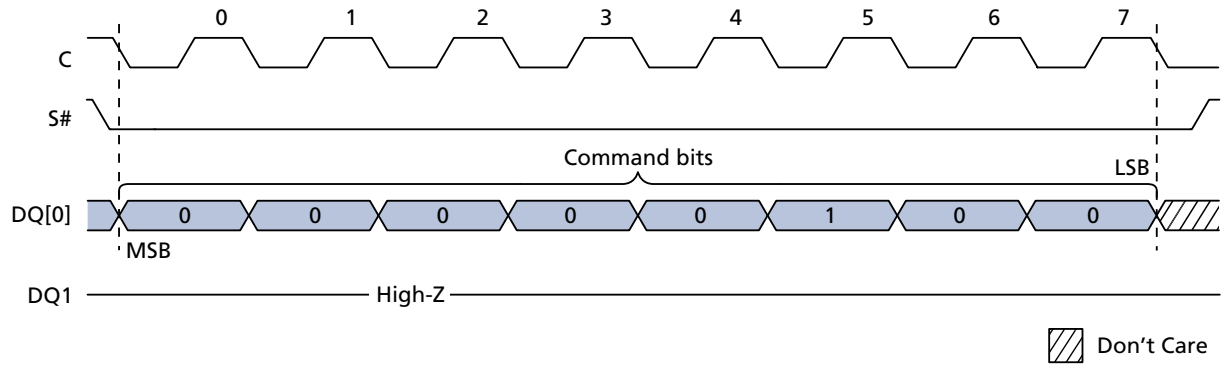
The WRITE DISABLE command resets the write enable latch (WEL) bit.

The WRITE DISABLE command is entered by driving chip select (S#) LOW, sending the command code, and then driving S# HIGH.

The WEL bit is reset under the following conditions:

- Power-up
- Completion of any ERASE operation
- Completion of any PROGRAM operation
- Completion of any WRITE STATUS REGISTER operation
- Completion of WRITE DISABLE operation

Figure 10: WRITE DISABLE Command Sequence



READ IDENTIFICATION

The READ IDENTIFICATION command reads the following device identification data:

- Manufacturer identification (1 byte): This is assigned by JEDEC.
- Device identification (2 bytes): This is assigned by device manufacturer; the first byte indicates memory type and the second byte indicates device memory capacity.
- A Unique ID code (UID) (17 bytes,16 available upon customer request): The first byte contains length of data to follow; the remaining 16 bytes contain optional Customized Factory Data (CFD) content.

Table 15: READ IDENTIFICATION Data Out Sequence

Manufacturer Identification	Device Identification		UID	
	Memory Type	Memory Capacity	CFD Length	CFD Content
20h	20h	11h	10h	16 bytes
	71h	12h		
		13h		
		14h		
		15h		
		16h		
		17h		
		18h		

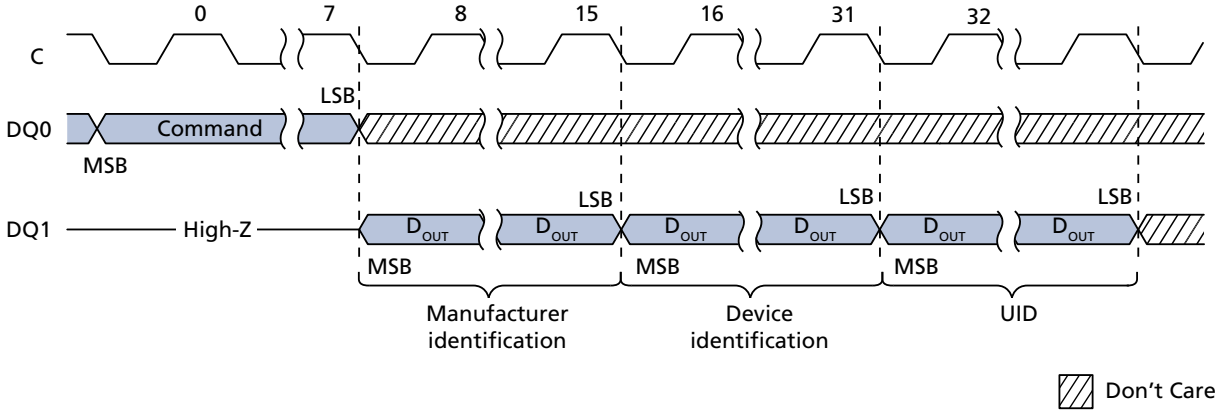
Note: 1. The CFD bytes are read-only and can be programmed with customer data upon demand. If customers do not make requests, the devices are shipped with all the CFD bytes programmed to zero.

A READ IDENTIFICATION command is not decoded while an ERASE or PROGRAM cycle is in progress and has no effect on a cycle in progress. The READ IDENTIFICATION command must not be issued while the device is in DEEP POWER-DOWN mode.

The device is first selected by driving S# LOW. Then the 8-bit command code is shifted in and content is shifted out on DQ1 as follows: the 24-bit device identification that is stored in the memory, the 8-bit CFD length, followed by 16 bytes of CFD content. Each bit is shifted out during the falling edge of serial clock (C).

The READ IDENTIFICATION command is terminated by driving S# HIGH at any time during data output. When S# is driven HIGH, the device is put in the STANDBY POWER mode and waits to be selected so that it can receive, decode, and execute commands.

Figure 11: READ IDENTIFICATION Command Sequence



READ STATUS REGISTER

The READ STATUS REGISTER command allows the status register to be read. The status register may be read at any time, even while a PROGRAM, ERASE, or WRITE STATUS REGISTER cycle is in progress. When one of these cycles is in progress, it is recommended to check the write in progress (WIP) bit before sending a new command to the device. It is also possible to read the status register continuously.

Figure 12: READ STATUS REGISTER Command Sequence

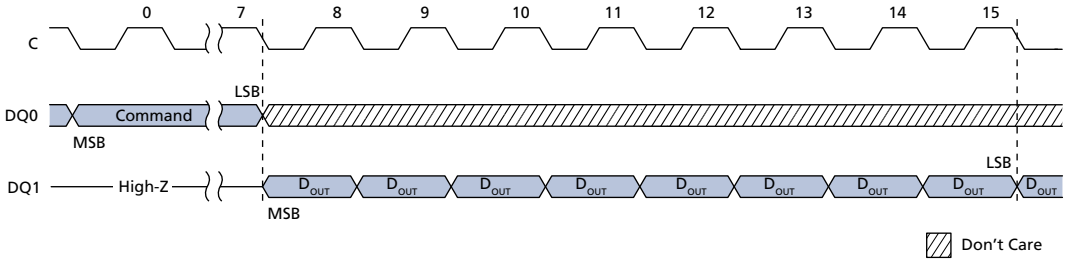
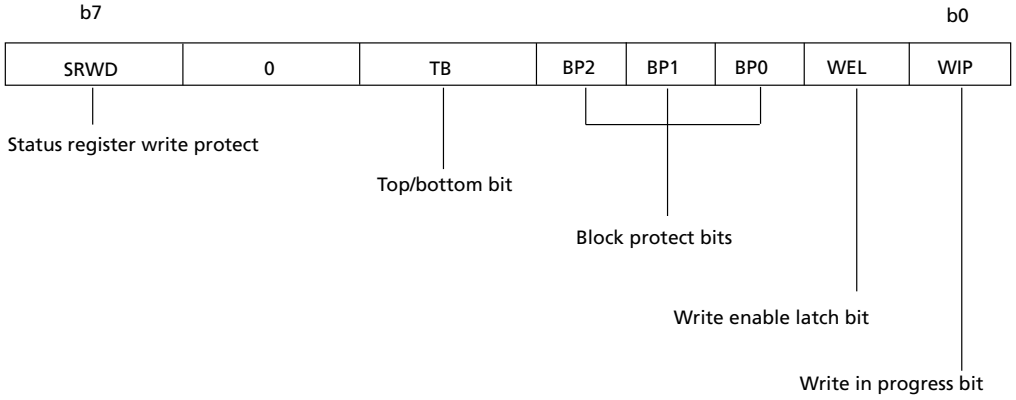


Figure 13: Status Register Format



WIP Bit

The write in progress (WIP) bit indicates whether the memory is busy with a WRITE STATUS REGISTER cycle, a PROGRAM cycle, or an ERASE cycle. When the WIP bit is set to 1, a cycle is in progress; when the WIP bit is set to 0, a cycle is not in progress.

WEL Bit

The write enable latch (WEL) bit indicates the status of the internal write enable latch. When the WEL bit is set to 1, the internal write enable latch is set; when the WEL bit is set to 0, the internal write enable latch is reset and no WRITE STATUS REGISTER, PROGRAM, or ERASE command is accepted.

Block Protect Bits

The block protect bits are nonvolatile. They define the size of the area to be software protected against PROGRAM and ERASE commands. The block protect bits are written with the WRITE STATUS REGISTER command.

When one or more of the block protect bits is set to 1, the relevant memory area, as defined in the Protected Area Sizes table, becomes protected against PAGE PROGRAM and SECTOR ERASE commands. The block protect bits can be written provided that the HARDWARE PROTECTED mode has not been set. The BULK ERASE command is executed only if all block protect bits are 0.

Top/Bottom Bit

The top/bottom (TB) bit is nonvolatile. It can be set and reset with the WRITE STATUS REGISTER command provided that the WRITE ENABLE command has been issued. The TB bit is used in conjunction with the block protect bits to determine if the protected area defined by the block protect bits starts from the top or the bottom of the memory array:

- When TB is reset to 0 (default value), the area protected by the block protect bits starts from the top of the memory array
- When TB is set to 1, the area protected by the block protect bits starts from the bottom of the memory array

The TB bit cannot be written when the status register write disable (SRWD) bit is set to 1 and the W# pin is driven LOW.

SRWD Bit

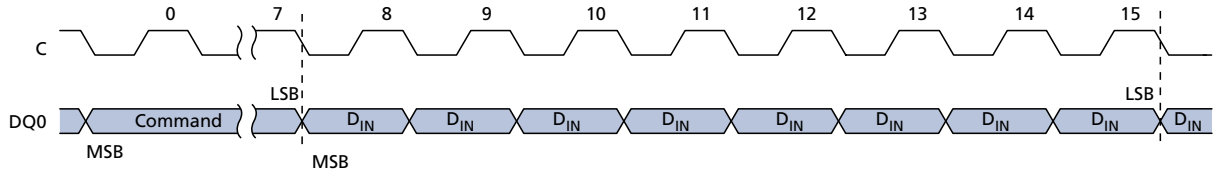
The status register write disable (SRWD) bit is operated in conjunction with the write protect (W#/V_{PP}) signal. When the SRWD bit is set to 1 and W#/V_{PP} is driven LOW, the device is put in the hardware protected mode. In the hardware protected mode, the nonvolatile bits of the status register (SRWD, and the block protect bits) become read-only bits and the WRITE STATUS REGISTER command is no longer accepted for execution.

WRITE STATUS REGISTER

The WRITE STATUS REGISTER command allows new values to be written to the status register. Before the WRITE STATUS REGISTER command can be accepted, a WRITE ENABLE command must have been executed previously. After the WRITE ENABLE command has been decoded and executed, the device sets the write enable latch (WEL) bit.

The WRITE STATUS REGISTER command is entered by driving chip select (S#) LOW, followed by the command code and the data byte on serial data input (DQ0). The WRITE STATUS REGISTER command has no effect on b6, b5, b4, b1, and b0 of the status register. The status register b6 is, b5, and b4 are always read as "0". S# must be driven HIGH after the eighth bit of the data byte has been latched in. If not, the WRITE STATUS REGISTER command is not executed.

Figure 14: WRITE STATUS REGISTER Command Sequence



As soon as S# is driven HIGH, the self-timed WRITE STATUS REGISTER cycle is initiated; its duration is ^{t}W. While the WRITE STATUS REGISTER cycle is in progress, the status register may still be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed WRITE STATUS REGISTER cycle, and is 0 when the cycle is completed. Also, when the cycle is completed, the WEL bit is reset.

The WRITE STATUS REGISTER command allows the user to change the values of the block protect bits (BP2, BP1, BP0). Setting these bit values defines the size of the area that is to be treated as read-only, as defined in the Protected Area Sizes table.

The WRITE STATUS REGISTER command also allows the user to set and reset the status register write disable (SRWD) bit in accordance with the write protect (W#/V_{PP}) signal. The SRWD bit and the W#/V_{PP} signal allow the device to be put in the hardware protected (HPM) mode. The WRITE STATUS REGISTER command is not executed once the HPM is entered. The options for enabling the status register protection modes are summarized here.

Table 16: Status Register Protection Modes

W#/V_{PP} Signal	SRWD Bit	Protection Mode (PM)	Status Register Write Protection	Memory Content		Notes
				Protected Area	Unprotected Area	
1	0	Software protected mode (SPM)	Software protection	Commands not accepted	Commands accepted	1, 2, 3
0	0					
1	1					
0	1	Hardware protected mode (HPM)	Hardware protection	Commands not accepted	Commands accepted	3, 4, 5,

Notes:
1. Software protection: status register is writable (SRWD, BP2, BP1, and BP0 bit values can be changed) if the WRITE ENABLE command has set the WEL bit.
2. PAGE PROGRAM, SECTOR ERASE, and BULK ERASE commands are not accepted.
3. PAGE PROGRAM and SECTOR ERASE commands can be accepted.
4. Hardware protection: status register is not writable (SRWD, BP2, BP1, and BP0 bit values cannot be changed).
5. PAGE PROGRAM, SECTOR ERASE, and BULK ERASE commands are not accepted.

When the SRWD bit of the status register is 0 (its initial delivery state), it is possible to write to the status register provided that the WEL bit has been set previously by a WRITE ENABLE command, regardless of whether the W#/V_{PP} signal is driven HIGH or LOW. When the status register SRWD bit is set to 1, two cases need to be considered depending on the state of the W#/V_{PP} signal:

- If the W#/V_{PP} signal is driven HIGH, it is possible to write to the status register provided that the WEL bit has been set previously by a WRITE ENABLE command.
- If the W#/V_{PP} signal is driven LOW, it is not possible to write to the status register even if the WEL bit has been set previously by a WRITE ENABLE command. Therefore, attempts to write to the status register are rejected, and are not accepted for execution. The result is that all the data bytes in the memory area that have been put in SPM by the status register block protect bits (BP2, BP1, BP0) are also hardware protected against data modification.

Regardless of the order of the two events, the HPM can be entered in either of the following ways:

- Setting the status register SRWD bit after driving the W#/V_{PP} signal LOW
- Driving the W#/V_{PP} signal LOW after setting the status register SRWD bit.

The only way to exit the HPM is to pull the W#/V_{PP} signal HIGH. If the W#/V_{PP} signal is permanently tied HIGH, the HPM can never be activated. In this case, only the SPM is available, using the status register block protect bits (BP2, BP1, BP0).

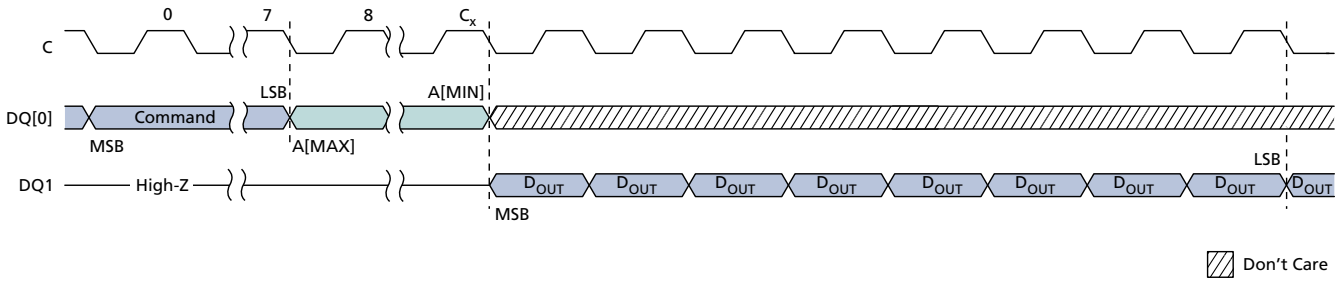
READ DATA BYTES

The device is first selected by driving chip select (S#) LOW. The command code for READ DATA BYTES is followed by a 3-byte address (A23-A0), each bit being latched-in during the rising edge of serial clock (C). Then the memory contents at that address is shifted out on serial data output (DQ1), each bit being shifted out at a maximum frequency ^{f}R during the falling edge of C.

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. Therefore, the entire memory can be read with a single READ DATA BYTES command. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

The READ DATA BYTES command is terminated by driving S# HIGH. S# can be driven HIGH at any time during data output. Any READ DATA BYTES command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 15: READ DATA BYTES Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

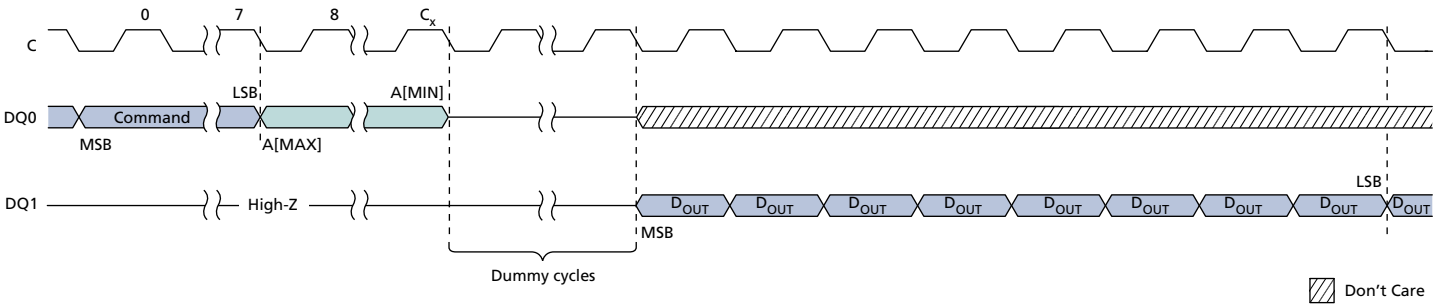
READ DATA BYTES at HIGHER SPEED

The device is first selected by driving chip select (S#) LOW. The command code for the READ DATA BYTES at HIGHER SPEED command is followed by a 3-byte address (A23-A0) and a dummy byte, each bit being latched-in during the rising edge of serial clock (C). Then the memory contents at that address are shifted out on serial data output (DQ1) at a maximum frequency ^{f}C, during the falling edge of C.

The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. Therefore, the entire memory can be read with a single READ DATA BYTES at HIGHER SPEED command. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

The READ DATA BYTES at HIGHER SPEED command is terminated by driving S# HIGH. S# can be driven HIGH at any time during data output. Any READ DATA BYTES at HIGHER SPEED command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 16: READ DATA BYTES at HIGHER SPEED Command Sequence



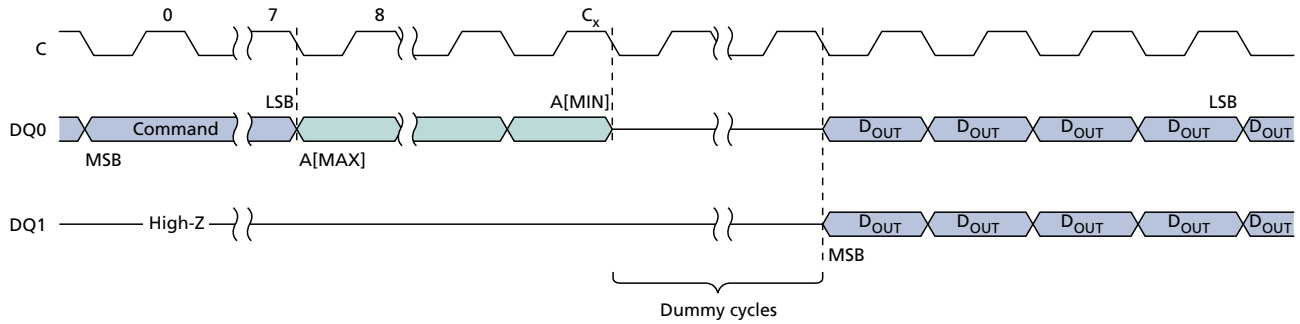
Note: 1. Cx = 7 + (A[MAX] + 1).

DUAL OUTPUT FAST READ

The DUAL OUTPUT FAST READ command is similar to the READ DATA BYTES at HIGHER SPEED command, except that data is shifted out on two pins (DQ0 and DQ1) instead of one. Outputting the data on two pins doubles the data transfer bandwidth compared to the READ DATA BYTES at HIGHER SPEED command.

The device is first selected by driving chip select S# LOW. The command code for the DUAL OUTPUT FAST READ command is followed by a 3-byte address (A23-A0) and a dummy byte, each bit being latched-in during the rising edge of serial clock (C). Then the memory contents at that address are shifted out on DQ0 and DQ1 at a maximum frequency f_C, during the falling edge of C.

Figure 17: DUAL OUTPUT FAST READ Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

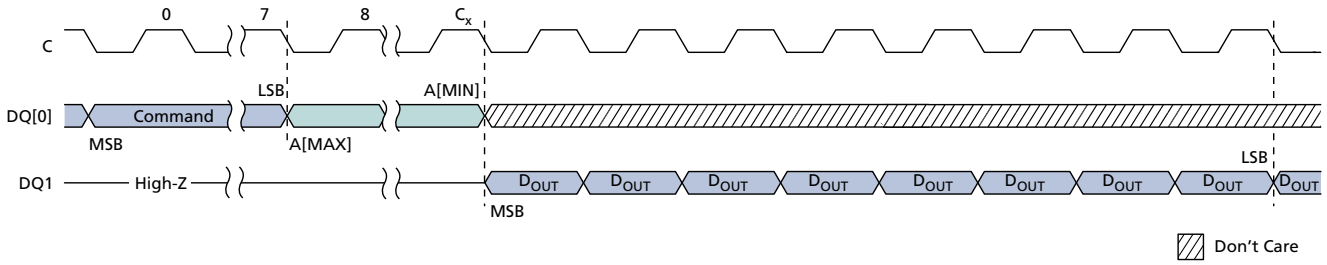
The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out on DQ0 and DQ1. Therefore, the entire memory can be read with a single DUAL OUTPUT FAST READ command. When the highest address is reached, the address counter rolls over to 00 0000h so that the read sequence can be continued indefinitely.

READ LOCK REGISTER

The device is first selected by driving chip select (S#) LOW. The command code for the READ LOCK REGISTER command is followed by a 3-byte address (A23-A0) pointing to any location inside the concerned sector. Each address bit is latched-in during the rising edge of serial clock (C). Then the value of the lock register is shifted out on serial data output (DQ1), each bit being shifted out at a maximum frequency f_C during the falling edge of C.

The READ LOCK REGISTER command is terminated by driving S# HIGH at any time during data output.

Figure 18: READ LOCK REGISTER Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

Any READ LOCK REGISTER command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Values of b1 and b0 after power-up are defined in Power-Up/Down and Supply Line Decoupling (page 47).

Table 17: Lock Register Out

Bit	Bit name	Value	Function
b7-b2	Reserved		
b1	Sector lock down	1	The write lock and lock-down bits cannot be changed. Once a value of 1 is written to the lock-down bit, it cannot be cleared to a value of 0 except by a power-up.
		0	The write lock and lock-down bits can be changed by writing new values to them.
b0	Sector write lock	1	WRITE, PROGRAM, and ERASE operations in this sector will not be executed. The memory contents will not be changed.
		0	WRITE, PROGRAM, or ERASE operations in this sector are executed and will modify the sector contents.

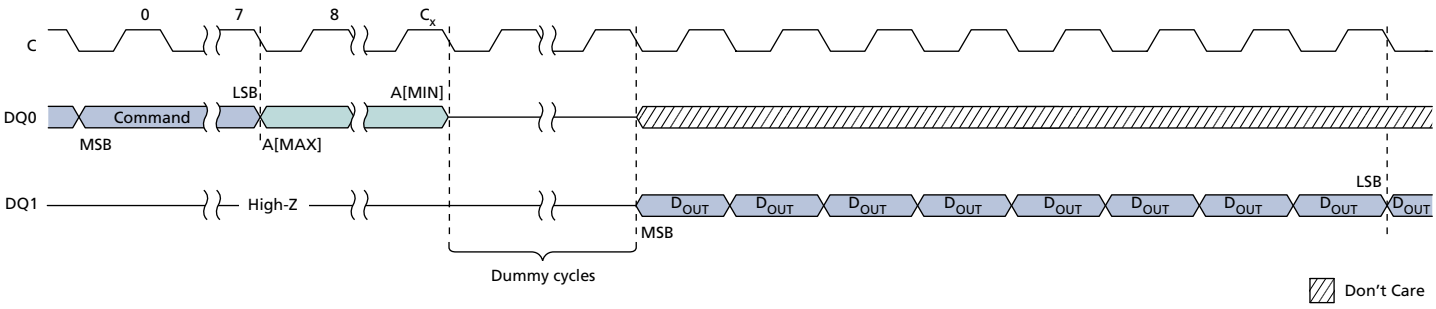
READ OTP

The device is first selected by driving chip select (S#) LOW. The command code for the READ OTP (one-time programmable) command is followed by a 3-byte address (A23-A0) and a dummy byte. Each bit is latched in on the rising edge of serial clock (C). Then the memory contents at that address are shifted out on serial data output (DQ1). Each bit is shifted out at the maximum frequency $f_C$max on the falling edge of C. The address is automatically incremented to the next higher address after each byte of data is shifted out.

There is no rollover mechanism with the READ OTP command. This means that the READ OTP command must be sent with a maximum of 65 bytes to read because once the 65th byte has been read, the same 65th byte continues to be read on the DQ1 pin.

The READ OTP command is terminated by driving S# HIGH. S# can be driven HIGH at any time during data output. Any READ OTP command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effect on the cycle that is in progress.

Figure 19: READ OTP Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

PAGE PROGRAM

The PAGE PROGRAM command allows bytes in the memory to be programmed, which means the bits are changed from 1 to 0. Before a PAGE PROGRAM command can be accepted a WRITE ENABLE command must be executed. After the WRITE ENABLE command has been decoded, the device sets the write enable latch (WEL) bit.

The PAGE PROGRAM command is entered by driving chip select (S#) LOW, followed by the command code, three address bytes, and at least one data byte on serial data input (DQ0).

If the eight least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same page; that is, from the address whose eight least significant bits (A7-A0) are all zero. S# must be driven LOW for the entire duration of the sequence.

If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without any effects on the other bytes of the same page.

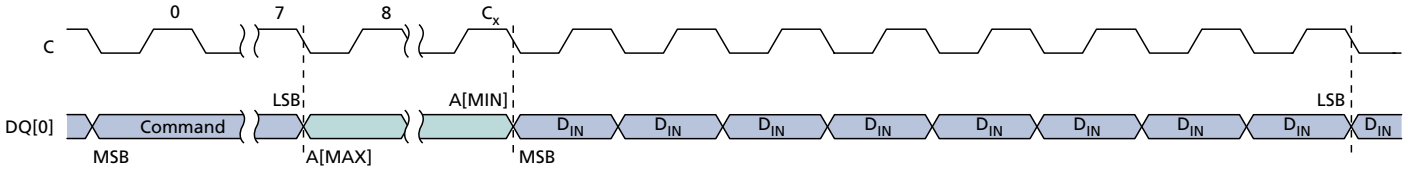
For optimized timings, it is recommended to use the PAGE PROGRAM command to program all consecutive targeted bytes in a single sequence rather than to use several PAGE PROGRAM sequences, each containing only a few bytes.

S# must be driven HIGH after the eighth bit of the last data byte has been latched in. Otherwise the PAGE PROGRAM command is not executed.

As soon as S# is driven HIGH, the self-timed PAGE PROGRAM cycle is initiated; the cycles's duration is t_{PP}. While the PAGE PROGRAM cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed PAGE PROGRAM cycle, and 0 when the cycle is completed. At some unspecified time before the cycle is completed, the write enable latch (WEL) bit is reset.

A PAGE PROGRAM command is not executed if it applies to a page protected by the block protect bits BP2, BP1, and BP0.

Figure 20: PAGE PROGRAM Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

DUAL INPUT FAST PROGRAM

The DUAL INPUT FAST PROGRAM command is similar to the PAGE PROGRAM command, except that data is entered on two pins (DQ0 and DQ1) instead of one, doubling the data transfer bandwidth.

The DUAL INPUT FAST PROGRAM command is entered by driving Chip Select S# LOW, followed by the command code, three address bytes, and at least one data byte on serial data input (DQ0).

If the eight least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page is programmed from the start address of the same page; that is, from the address whose eight least significant bits (A7-A0) are all zero. S# must be driven LOW for the entire duration of the sequence.

If more than 256 bytes are sent to the device, previously latched data is discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without any effect on other bytes in the same page.

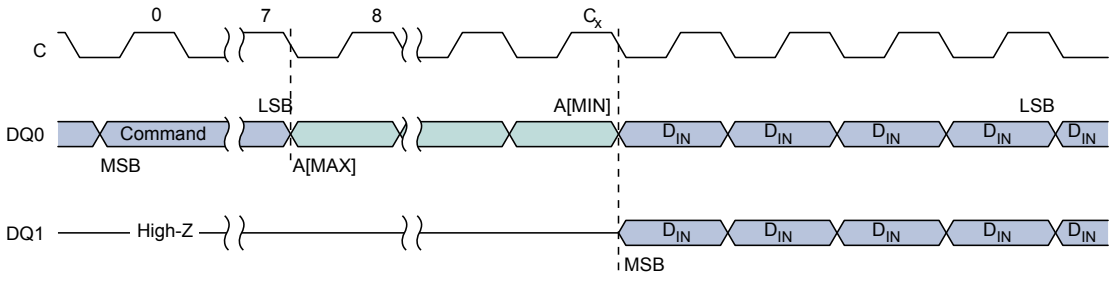
For optimized timings, it is recommended to use the DUAL INPUT FAST PROGRAM command to program all consecutive targeted bytes in a single sequence than to use several DUAL INPUT FAST PROGRAM sequences, each containing only a few bytes.

S# must be driven HIGH after the eighth bit of the last data byte has been latched in. Otherwise the DUAL INPUT FAST PROGRAM command is not executed.

As soon as S# is driven HIGH, the self-timed PAGE PROGRAM cycle is initiated; the cycle's duration is t_{PP}. While the DUAL INPUT FAST PROGRAM cycle is in progress, the status register may be read to check the value of the write In progress (WIP) bit. The WIP bit is 1 during the self-timed PAGE PROGRAM cycle, and 0 when the cycle is completed. At some unspecified time before the cycle is completed, the write enable latch (WEL) bit is reset.

A DUAL INPUT FAST PROGRAM command is not executed if it applies to a page protected by the block protect bits BP2, BP1, and BP0.

Figure 21: DUAL INPUT FAST PROGRAM Command Sequence



Notes:
1. For the M25PX16, the DUAL INPUT FAST PROGRAM command is available only in VCC range 2.7 V - 3.6 V.
2. Cx = 7 + (A[MAX] + 1).

PROGRAM OTP

The PROGRAM OTP command allows a maximum of 64 bytes in the OTP memory area to be programmed, which means the bits are changed from 1 to 0. Before a PROGRAM OTP command can be accepted, a WRITE ENABLE command must have been executed previously. After the WRITE ENABLE command has been decoded, the device sets the write enable latch (WEL) bit.

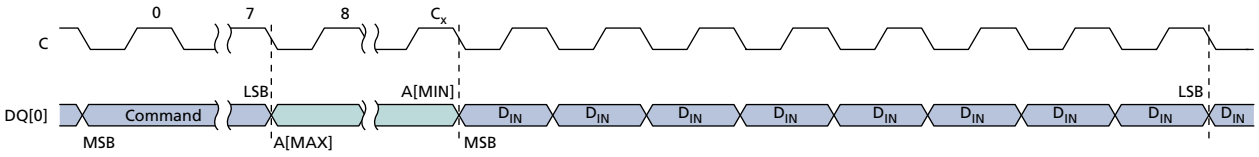
The PROGRAM OTP command is entered by driving chip select (S#) LOW, followed by the command opcode, three address bytes, and at least one data byte on serial data input (DQ0).

S# must be driven HIGH after the eighth bit of the last data byte has been latched in. Otherwise the PROGRAM OTP command is not executed.

There is no rollover mechanism with the PROGRAM OTP command. This means that the PROGRAM OTP command must be sent with a maximum of 65 bytes to program. When all 65 bytes have been latched in, any following byte will be discarded.

As soon as S# is driven HIGH, the self-timed PAGE PROGRAM cycle is initiated; the cycle's duration is t_{PP}. While the PROGRAM OTP cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed PROGRAM OTP cycle, and 0 when when the cycle is completed. At some unspecified time before the cycle is complete, the WEL bit is reset.

Figure 22: PROGRAM OTP Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

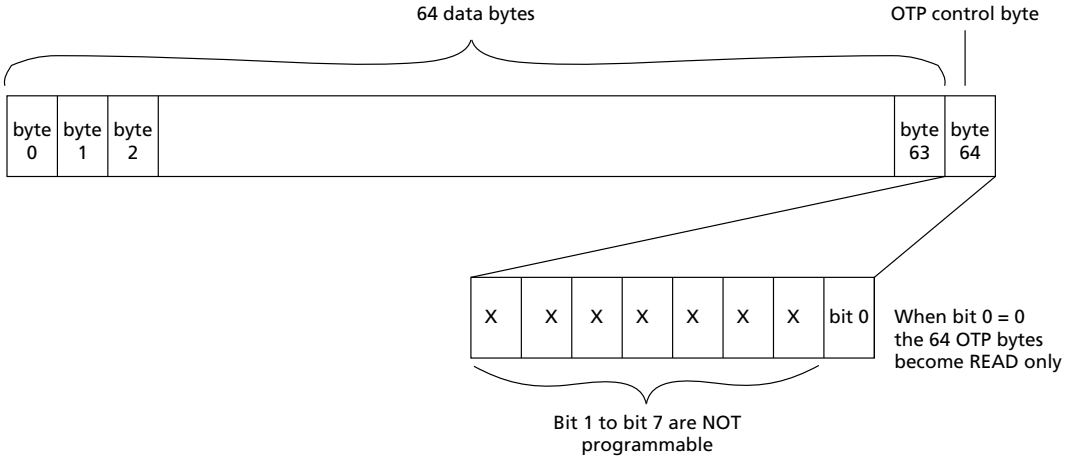
The OTP control byte is byte 64. Bit 0 of this OTP control byte is used to permanently lock the OTP memory array.

- When bit 0 of the OTP control byte = 1, the 64 bytes of the OTP memory array can be programmed.
- When bit 0 of the OTP control byte = 0, the 64 bytes of the OTP memory array are read-only and cannot be programmed anymore.

Once a bit of the OTP memory has been programmed to 0, it can no longer be set to 1. Therefore, as soon as bit 0 of the control byte is set to 0, the 64 bytes of the OTP memory array is set permanently as read-only.

Any PROGRAM OTP command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 23: How to Permanently Lock the OTP Bytes



WRITE to LOCK REGISTER

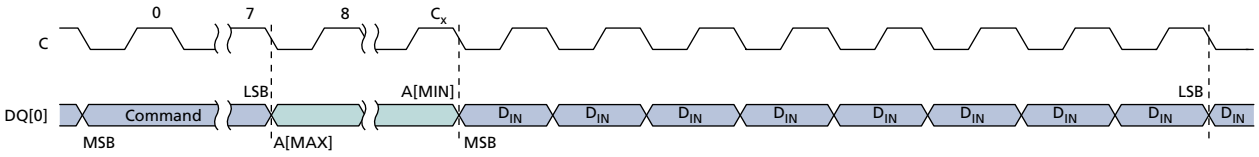
The WRITE to LOCK REGISTER instruction allows the lock register bits to be changed. Before the WRITE to LOCK REGISTER instruction can be accepted, a WRITE ENABLE instruction must have been executed previously. After the WRITE ENABLE instruction has been decoded, the device sets the write enable latch (WEL) bit.

The WRITE to LOCK REGISTER instruction is entered by driving chip select (S#) LOW, followed by the instruction code, three address bytes, and one data byte on serial data input (DQ0). The address bytes must point to any address in the targeted sector. S# must be driven HIGH after the eighth bit of the data byte has been latched in. Otherwise the WRITE to LOCK REGISTER instruction is not executed.

Lock register bits are volatile, and therefore do not require time to be written. When the WRITE to LOCK REGISTER instruction has been successfully executed, the WEL bit is reset after a delay time of less than t_{SHSL} minimum value.

Any WRITE to LOCK REGISTER instruction issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 24: WRITE to LOCK REGISTER Instruction Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

Table 18: Lock Register In

Sector	Bit	Value
All sectors	b7–b2	0
All sectors	b1	Sector lock-down bit value
All sectors	b0	Sector write lock bit value

Note: Values of b1 and b0 after power-up are defined in Power-Up/Down and Supply Line Decoupling (page 47). For the sector lock down and sector write lock values, see the Lock Register Out table.

SUBSECTOR ERASE

The SUBSECTOR ERASE command sets to 1 (FFh) all bits inside the chosen subsector. Before the SUBSECTOR ERASE command can be accepted, a WRITE ENABLE command must have been executed previously. After the WRITE ENABLE command has been decoded, the device sets the write enable latch (WEL) bit.

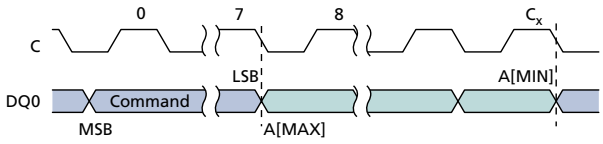
The SUBSECTOR ERASE command is entered by driving chip select (S#) LOW, followed by the command code, and three address bytes on serial data input (DQ0). Any address inside the subsector is a valid address for the SUBSECTOR ERASE command. S# must be driven LOW for the entire duration of the sequence.

S# must be driven HIGH after the eighth bit of the last address byte has been latched in. Otherwise the SUBSECTOR ERASE command is not executed. As soon as S# is driven HIGH, the self-timed SUBSECTOR ERASE cycle is initiated; the cycle's duration is t_{SSE}. While the SUBSECTOR ERASE cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed SUBSECTOR ERASE cycle, and is 0 when the cycle is completed. At some unspecified time before the cycle is complete, the WEL bit is reset.

A SUBSECTOR ERASE command issued to a sector that is hardware or software protected is not executed.

Any SUBSECTOR ERASE command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 25: SUBSECTOR ERASE Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

SECTOR ERASE

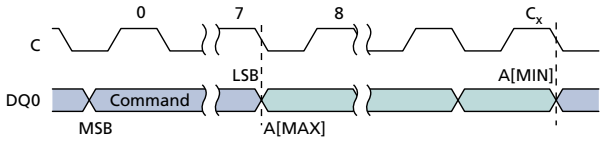
The SECTOR ERASE command sets to 1 (FFh) all bits inside the chosen sector. Before the SECTOR ERASE command can be accepted, a WRITE ENABLE command must have been executed previously. After the WRITE ENABLE command has been decoded, the device sets the write enable latch (WEL) bit.

The SECTOR ERASE command is entered by driving chip select (S#) LOW, followed by the command code, and three address bytes on serial data input (DQ0). Any address inside the sector is a valid address for the SECTOR ERASE command. S# must be driven LOW for the entire duration of the sequence.

S# must be driven HIGH after the eighth bit of the last address byte has been latched in. Otherwise the SECTOR ERASE command is not executed. As soon as S# is driven HIGH, the self-timed SECTOR ERASE cycle is initiated; the cycle's duration is t_{SE}. While the SECTOR ERASE cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed SECTOR ERASE cycle, and is 0 when the cycle is completed. At some unspecified time before the cycle is completed, the WEL bit is reset.

A SECTOR ERASE command is not executed if it applies to a sector that is hardware or software protected.

Figure 26: SECTOR ERASE Command Sequence



Note: 1. Cx = 7 + (A[MAX] + 1).

BULK ERASE

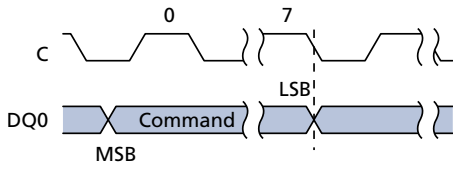
The BULK ERASE command sets all bits to 1 (FFh). Before the BULK ERASE command can be accepted, a WRITE ENABLE command must have been executed previously. After the WRITE ENABLE command has been decoded, the device sets the write enable latch (WEL) bit.

The BULK ERASE command is entered by driving chip select (S#) LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

S# must be driven HIGH after the eighth bit of the command code has been latched in. Otherwise the BULK ERASE command is not executed. As soon as S# is driven HIGH, the self-timed BULK ERASE cycle is initiated; the cycle's duration is t_{BE}. While the BULK ERASE cycle is in progress, the status register may be read to check the value of the write In progress (WIP) bit. The WIP bit is 1 during the self-timed BULK ERASE cycle, and is 0 when the cycle is completed. At some unspecified time before the cycle is completed, the WEL bit is reset.

The BULK ERASE command is executed only if all block protect (BP2, BP1, BP0) bits are 0. The BULK ERASE command is ignored if one or more sectors are protected.

Figure 27: BULK ERASE Command Sequence



DEEP POWER-DOWN

Executing the DEEP POWER-DOWN command is the only way to put the device in the lowest power consumption mode, the DEEP POWER-DOWN mode. The DEEP POWER-DOWN command can also be used as a software protection mechanism while the device is not in active use because in the DEEP POWER-DOWN mode the device ignores all WRITE, PROGRAM, and ERASE commands.

Driving chip select (S#) HIGH deselects the device, and puts it in the STANDBY POWER mode if there is no internal cycle currently in progress. Once in STANDBY POWER mode, the DEEP POWER-DOWN mode can be entered by executing the DEEP POWER-DOWN command, subsequently reducing the standby current from I_{CC1} to I_{CC2}.

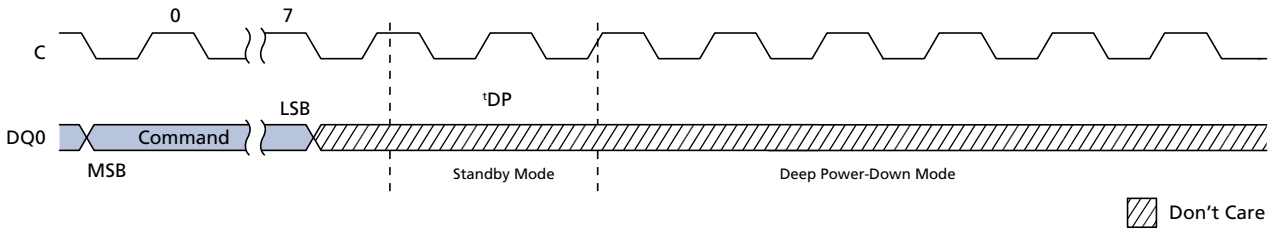
To take the device out of DEEP POWER-DOWN mode, the RELEASE from DEEP POWER-DOWN command must be issued. Other commands must not be issued while the device is in DEEP POWER-DOWN mode. The DEEP POWER-DOWN mode stops automatically at power-down. The device always powers up in STANDBY POWER mode.

The DEEP POWER-DOWN command is entered by driving S# LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

S# must be driven HIGH after the eighth bit of the command code has been latched in. Otherwise the DEEP POWER-DOWN command is not executed. As soon as S# is driven HIGH, it requires a delay of t_{DP} before the supply current is reduced to I_{CC2} and the DEEP POWER-DOWN mode is entered.

Any DEEP POWER-DOWN command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 28: DEEP POWER-DOWN Command Sequence



RELEASE from DEEP POWER-DOWN

Once the device has entered DEEP POWER-DOWN mode, all commands are ignored except RELEASE from DEEP POWER-DOWN. Executing this command takes the device out of the DEEP POWER-DOWN mode.

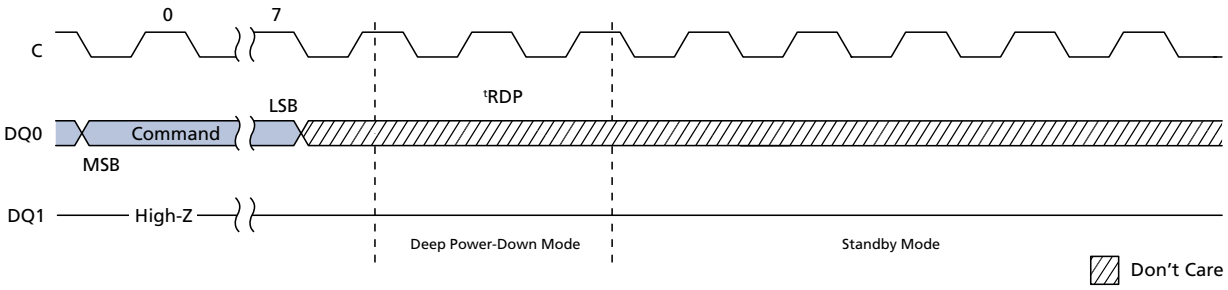
The RELEASE from DEEP POWER-DOWN command is entered by driving chip select (S#) LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

The RELEASE from DEEP POWER-DOWN command is terminated by driving S# high. Sending additional clock cycles on serial clock C while S# is driven LOW causes the command to be rejected and not executed.

After S# has been driven high, followed by a delay, t_{RDP}, the device is put in the STAND-BY mode. S# must remain HIGH at least until this period is over. The device waits to be selected so that it can receive, decode, and execute commands.

Any RELEASE from DEEP POWER-DOWN command issued while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without any effect on the cycle that is in progress.

Figure 29: RELEASE from DEEP POWER-DOWN Command Sequence



Power-Up/Down and Supply Line Decoupling

At power-up and power-down, the device must not be selected; that is, chip select (S#) must follow the voltage applied on V_{CC} until V_{CC} reaches the correct value:

- $V_{CC,min}$ at power-up, and then for a further delay of t_{VSL}
- V_{SS} at power-down

A safe configuration is provided in the SPI Modes section.

To avoid data corruption and inadvertent write operations during power-up, a power-on-reset (POR) circuit is included. The logic inside the device is held reset while V_{CC} is less than the POR threshold voltage, V_{WI} – all operations are disabled, and the device does not respond to any instruction. Moreover, the device ignores the following instructions until a time delay of ^{t}PUW has elapsed after the moment that V_{CC} rises above the V_{WI} threshold:

- WRITE ENABLE
- PAGE PROGRAM
- DUAL INPUT FAST PROGRAM
- PROGRAM OTP
- SUBSECTOR ERASE
- SECTOR ERASE
- BULK ERASE
- WRITE STATUS REGISTER
- WRITE to LOCK REGISTER

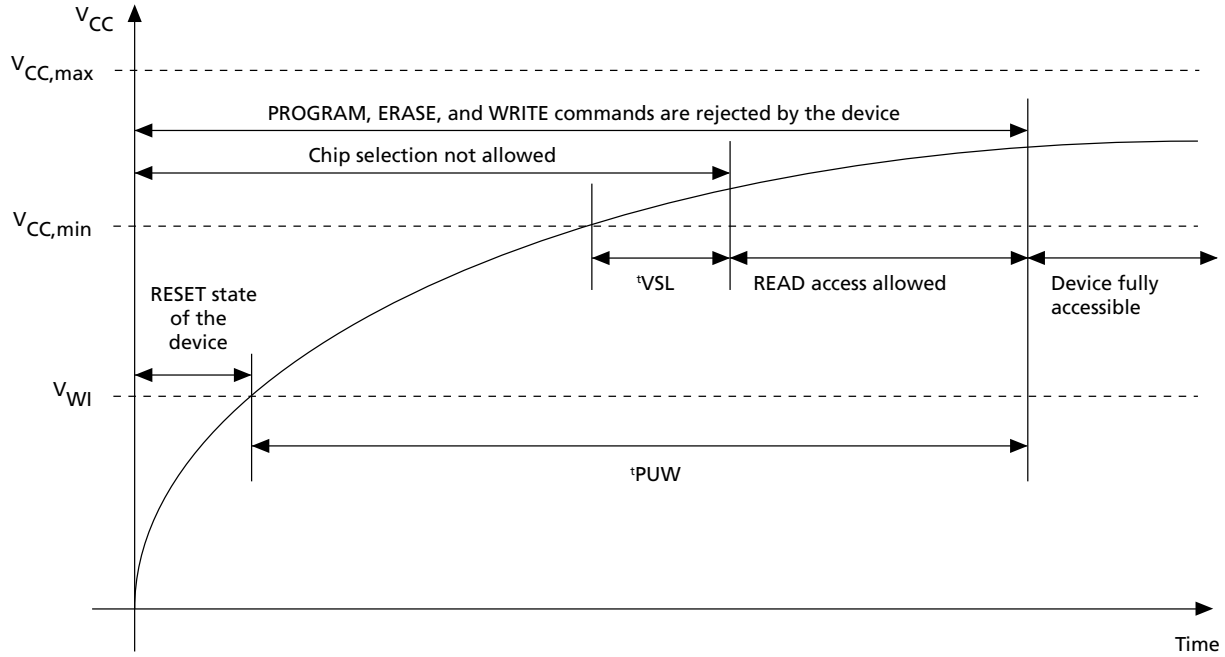
However, the correct operation of the device is not guaranteed if, by this time, V_{CC} is still below $V_{CC.min}$. No WRITE STATUS REGISTER, PROGRAM, or ERASE instruction should be sent until:

- ^{t}PUW after V_{CC} has passed the V_{WI} threshold
- ^{t}VSL after V_{CC} has passed the $V_{CC,min}$ level

If the time, ^{t}VSL, has elapsed, after V_{CC} rises above $V_{CC,min}$, the device can be selected for READ instructions even if the ^{t}PUW delay has not yet fully elapsed.

V_{PPH} must be applied only when V_{CC} is stable and in the $V_{CC,min}$ to $V_{CC,max}$ voltage range.

Figure 30: Power-Up Timing



After power-up, the device is in the following state:

- Standby power mode (not the deep power-down mode)
- Write enable latch (WEL) bit is reset
- Write in progress (WIP) bit is reset
- Write lock bit = 0
- Lock down bit = 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 close to the package pins; generally, this capacitor is of the order of 100 nF.

At power-down, when V_{CC} drops from the operating voltage to below the POR threshold voltage V_{WI}, all operations are disabled and the device does not respond to any instruction.

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

Maximum Ratings and Operating Conditions

Caution: Stressing the device beyond the absolute maximum ratings may cause permanent damage to the device. These are stress ratings only and operation of the device beyond any specification or condition in the operating sections of this datasheet is not recommended. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 19: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units	Notes
T_{STG}	Storage temperature	–65	150	°C	
T_{LEAD}	Lead temperature during soldering	—	See note	°C	1
V_{IO}	Input and output voltage (with respect to ground)	–0.6	V_{CC}+0.6	V	
V_{CC}	Supply voltage	–0.6	4.0	V	
V_{PP}	FAST PROGRAM / ERASE voltage	–0.2	10.0	V	2
V_{ESD}	Electrostatic discharge voltage (Human Body model)	–2000	2000	V	3

Notes:
1. The T_{LEAD} signal is compliant with JEDEC Std J-STD-020C (for small body, Sn-Pb or Pb assembly), the Micron RoHS compliant 7191395 specification, and the European directive on Restrictions on Hazardous Substances (RoHS) 2002/95/EU.
2. Avoid applying V_{PPH} to the W#/VPP pin during the BULK ERASE operation.
3. The V_{ESD} signal: JEDEC Std JESD22-A114A (C1 = 100 pF, R1 = 1500 Ω, R2 = 500 Ω).

Table 20: Operating Conditions

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply voltage	2.7	3.6	V
V_{PPH}	Supply voltage on V_{PP} pin	8.5	9.5	V
T_A	Ambient operating temperature (device grade 6)	–40	85	°C
T_A	Ambient operating temperature (device grade 3)	–40	125	°C

Table 21: Data Retention and Endurance

Parameter	Condition	Min	Max	Unit
PROGRAM and ERASE cycles	Grade 3; Autograde 6; Grade 6	100,000	–	Cycles per sector
Data Retention	at 55°C	20	–	years

Electrical Characteristics

Table 22: Power Up Timing Specifications

Symbol	Parameter	Min	Max	Units
^{t}VSL	V_{CC}[MIN] to S# LOW	30	–	µs
^{t}PUW	Time delay to WRITE command	1	10	ms
V_{WI}	Write Inhibit voltage	1.5	2.5	V

Table 23: DC Current Specifications

Symbol	Parameter	Test Condition	Min	Max	Units	Notes
I_{LI}	Input leakage current	–	–	±2	µA	
I_{LO}	Output leakage current	–	–	±2	µA	
I_{CC1}	Standby current	S# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	–	50/100	µA	1
I_{CC2}	Deep power-down current	S# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	–	10/100	µA	
I_{CC3}	Operating current (READ)	C = 0.1V_{CC} / 0.9V_{CC} at 75MHz, DQ1 = open	–	12	mA	
		C = 0.1V_{CC} / 0.9V_{CC} at 33MHz, DQ1 = open	–	4	mA	
	Operating current (DUAL OUTPUT FAST READ)	C = 0.1V_{CC} / 0.9V_{CC} at 75MHz, DQ1 = open	–	15	mA	
I_{CC4}	Operating current (PAGE PROGRAM)	S# = V_{CC}	–	15	mA	
	Operating current (DUAL INPUT FAST PROGRAM)	S# = V_{CC}	–	15	mA	
I_{CC5}	Operating current (WRITE STATUS REGISTER)	S# = V_{CC}	–	15	mA	
I_{CC6}	Operating current (SECTOR ERASE)	S# = V_{CC}	–	15	mA	
I_{CC7}	Operating current (BULK ERASE)	S# = V_{CC}	–	15	mA	

Note: 1. All specifications apply to both device grade 6 and device grade 3, except for the following standby and deep power-down current differences: device grade 6, I_{CC1} = 50µA and I_{CC2} = 10µA; device grade 3, I_{CC1} = 100µA and I_{CC2} = 100µA.

Table 24: DC Voltage Specifications

Symbol	Parameter	Test Condition	Min	Max	Units
V_{IL}	Input LOW voltage	–	–0.5	0.3V_{CC}	V
V_{IH}	Input HIGH voltage	–	0.7V_{CC}	V_{CC}+0.4	V
V_{OL}	Output LOW voltage	I_{OL} = 1.6mA	–	0.4	V

Table 24: DC Voltage Specifications (Continued)

Symbol	Parameter	Test Condition	Min	Max	Units
V_{OH}	Output HIGH voltage	I_{OL} = –100µA	V_{CC}–0.2	–	V

Note: 1. All specifications apply to both device grade 6 and device grade 3.

AC Characteristics

In the following AC specifications, output High-Z is defined as the point where data out is no longer driven; however, this is not applicable to the M25PX64 device.

Table 25: AC Measurement Conditions

Symbol	Parameter	Min	Max	Unit
C_L	Load capacitance	30	30	pF
	Input rise and fall times	–	5	ns
	Input pulse voltages	$0.2V_{CC}$	$0.8V_{CC}$	V
	Input timing reference voltages	$0.3V_{CC}$	$0.7V_{CC}$	V
	Output timing reference voltages	V_{CC} / 2	V_{CC} / 2	V

Figure 31: AC Measurement I/O Waveform

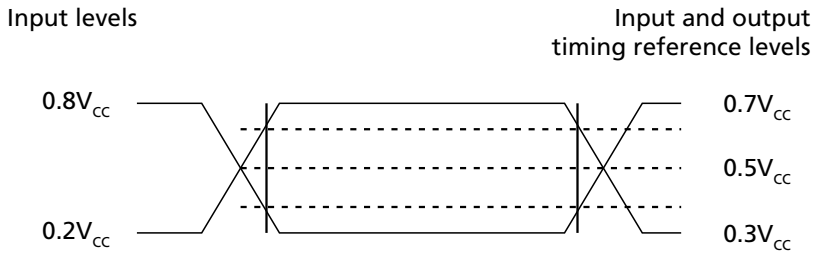


Table 26: Capacitance

Symbol	Parameter	Test condition	Min	Max	Unit	Notes
$C_{IN/OUT}$	Input/output capacitance (DQ0/DQ1)	V_{OUT} = 0 V	–	8	pF	1
C_{IN}	Input capacitance (other pins)	V_{IN} = 0 V	–	6	pF	

Note: 1. Values are sampled only, not 100% tested, at T_A=25°C and a frequency of 33MHz.

Table 27: AC Specifications (75MHz)

Note 1 applies to the entire table.

Symbol	Alt.	Parameter	Min	Typ	Max	Unit	Notes
f_C	f_C	Clock frequency for all commands (except READ)	DC	–	75	MHz	
f_R	–	Clock frequency for READ command	DC	–	33	MHz	
t_{CH}	t_{CLH}	Clock HIGH time	6	–	–	ns	2
t_{CL}	t_{CLL}	Clock LOW time	6	–	–	ns	2
t_{CLCH}	–	Clock rise time (peak to peak)	0.1	–	–	V/ns	3, 4
t_{CHCL}	–	Clock fall time (peak to peak)	0.1	–	–	V/ns	3, 4
t_{SLCH}	t_{CSS}	S# active setup time (relative to C)	5	–	–	ns	
t_{CHSL}		S# not active hold time (relative to C)	5	–	–	ns	
t_{DVCH}	t_{DSU}	Data-in setup time	2	–	–	ns	
t_{CHDX}	t_{DH}	Data-in hold time	5	–	–	ns	
t_{CHSH}	–	S# active hold time (relative to C)	5	–	–	ns	
t_{SHCH}	–	S# not active setup time (relative to C)	5	–	–	ns	
t_{SHSL}	t_{CSH}	S# deselect time	80	–	–	ns	
t_{SHQZ}	t_{DIS}	Output disable time	–	–	8	ns	3
t_{CLQV}	t_V	Clock LOW to output valid under 30pF	–	–	8	ns	
		Clock LOW to output valid under 10pF	–	–	6	ns	
t_{CLQX}	t_{HO}	Output hold time	0	–	–	ns	
t_{HLCH}	–	HOLD# setup time (relative to C)	5	–	–	ns	
t_{CHHH}	–	HOLD# hold time (relative to C)	5	–	–	ns	
t_{HHCH}	–	HOLD# setup time (relative to C)	5	–	–	ns	
t_{CHHL}	–	HOLD# hold time (relative to C)	5	–	–	ns	
t_{HHQX}	t_{LZ}	HOLD# to output Low-Z	–	–	8	ns	3
t_{HLQZ}	t_{HZ}	HOLD# to output High-Z	–	–	8	ns	3
t_{WHSL}	–	WRITE PROTECT setup time	20	–	–	ns	5
t_{SHWL}	–	WRITE PROTECT hold time	100	–	–	ns	5
t_{VPPHSL}	–	Enhanced program supply voltage HIGH (V_{PPH}) to S# LOW	200	–	–	ns	6
t_{DP}	–	S# HIGH to deep power-down mode	–	–	3	µs	3
t_{RDP}	–	S# HIGH to standby mode	–	–	30	µs	3
t_W	–	WRITE STATUS REGISTER cycle time	–	1.3	15	ms	
t_{PP}	–	PAGE PROGRAM cycle time (256 bytes)	–	0.8	5	ms	7
t_{PP}	–	PAGE PROGRAM cycle time (n bytes)	–	int(n/8) × 0.025	5	ms	7, 8
t_{PP}	–	PROGRAM OTP cycle time (64 bytes)	–	0.2	5	ms	7
t_{SSE}	–	SUBSECTOR ERASE cycle time	–	70	150	ms	
t_{SE}	–	SECTOR ERASE cycle time	–	0.7	3	s	

Table 27: AC Specifications (75MHz) (Continued)

Note 1 applies to the entire table.

Symbol	Alt.	Parameter	Min	Typ	Max	Unit	Notes
t_{BE}	–	BULK ERASE cycle time	–	34	80	s	

Notes:
1. AC specification values for 75MHz operations shown here are allowed only on the VCC range 2.7V - 3.6V. Typical values are given for T_A = 25°C.
2. The sum of t_{CH} + t_{CL} signal values must be greater than or equal to $1/f_C$.
3. The t_{CLCH}, t_{CHCL}, t_{SHQZ}, t_{HHQX}, t_{HLQZ}, t_{DP}, and t_{RDP} signal values are guaranteed by characterization, not 100% tested in production.
4. The t_{CLCH} and t_{CHCL} signals clock rise and fall time values are expressed as a slew-rate.
5. The t_{WHSL} and t_{SHWL} signal values are only applicable as a constraint for a WRITE STATUS REGISTER command when SRWD bit is set at 1.
6. The t_{VPPHSL} signal value for V_{PPH} should be kept at a valid level until the program or erase operation has completed and its result (success or failure) is known. Avoid applying V_{PPH} to the W/VPP pin during the BULK ERASE operation.
7. To obtain optimized timings (t_{PP}) when programming consecutive bytes with the PAGE PROGRAM command, use one sequence including all the bytes versus several sequences of only a few bytes (1 is less than or equal to n is less than or equal to 256).
8. int(A) corresponds to the upper integer part of A. For example, int(12/8) = 2, int(32/8) = 4 int(15.3) =16.
9. OE# may be delayed by up to ^{t}ELQV - ^{t}GLQV after CE#'s falling edge without impact to ^{t}ELQV.

Figure 32: Serial Input Timing

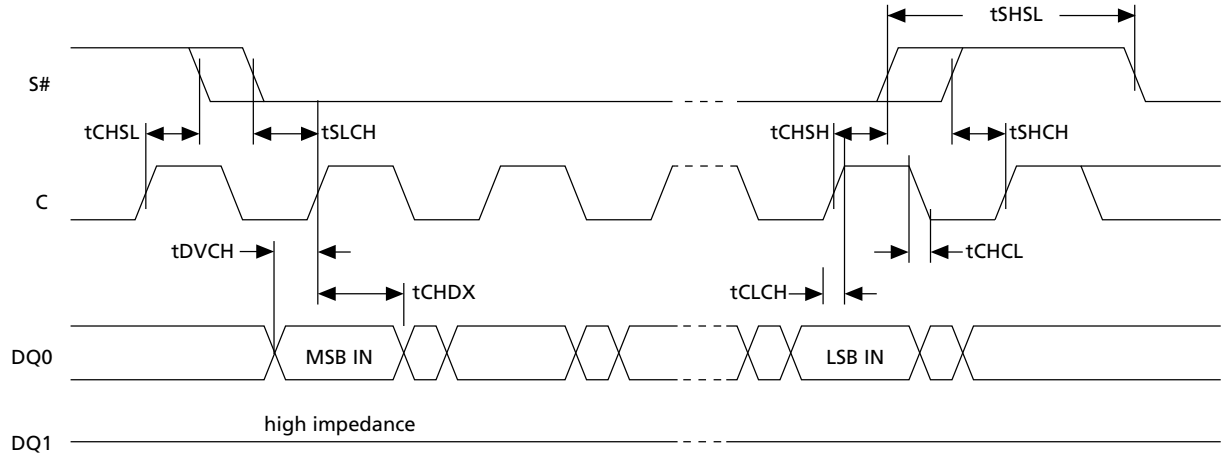


Figure 33: Write Protect Setup and Hold During WRSR when SRWD = 1 Timing

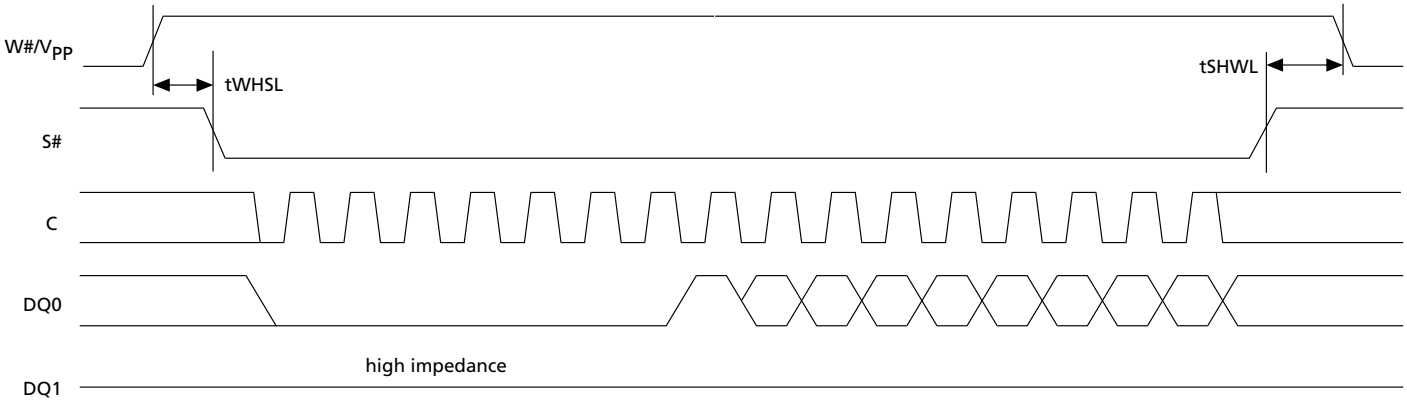


Figure 34: Hold Timing

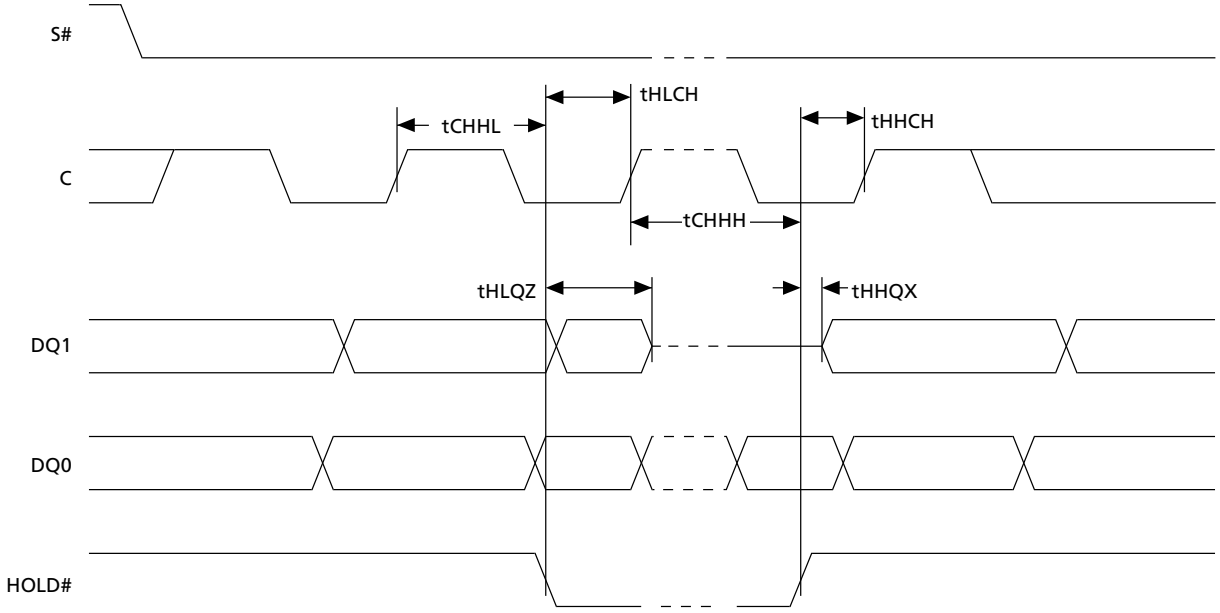


Figure 35: Output Timing

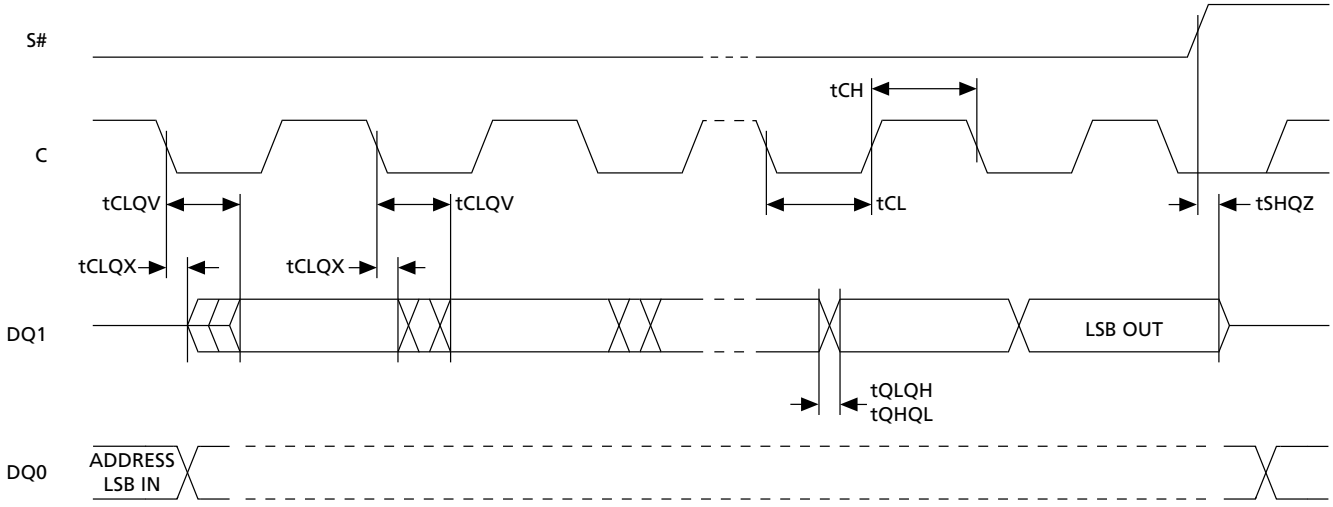
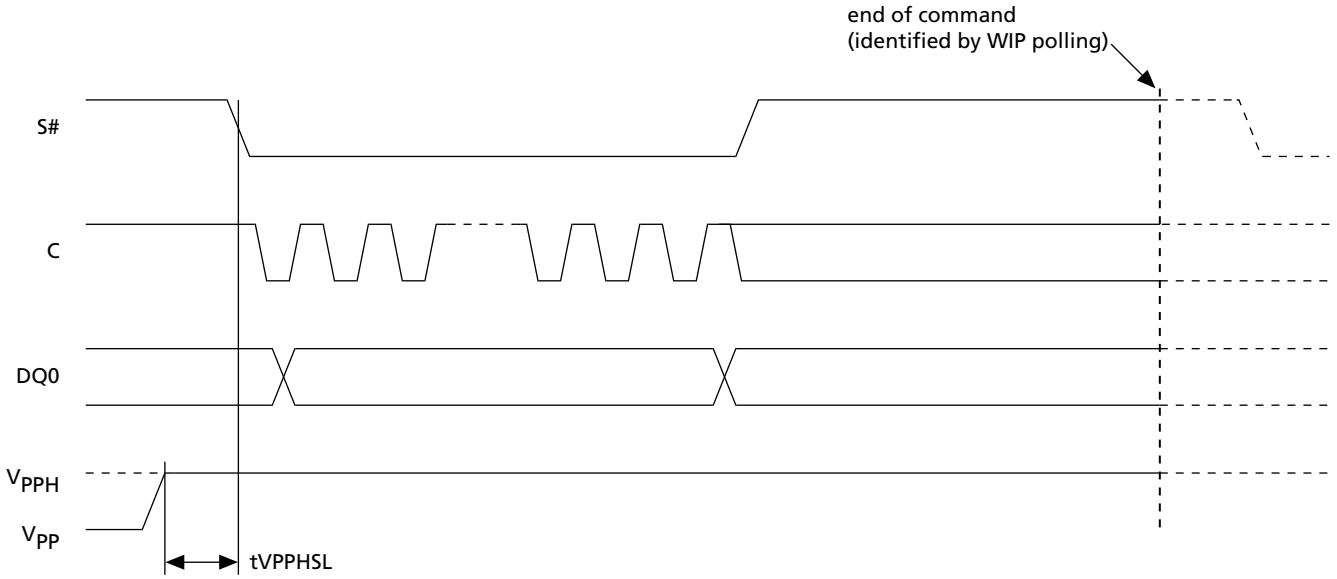
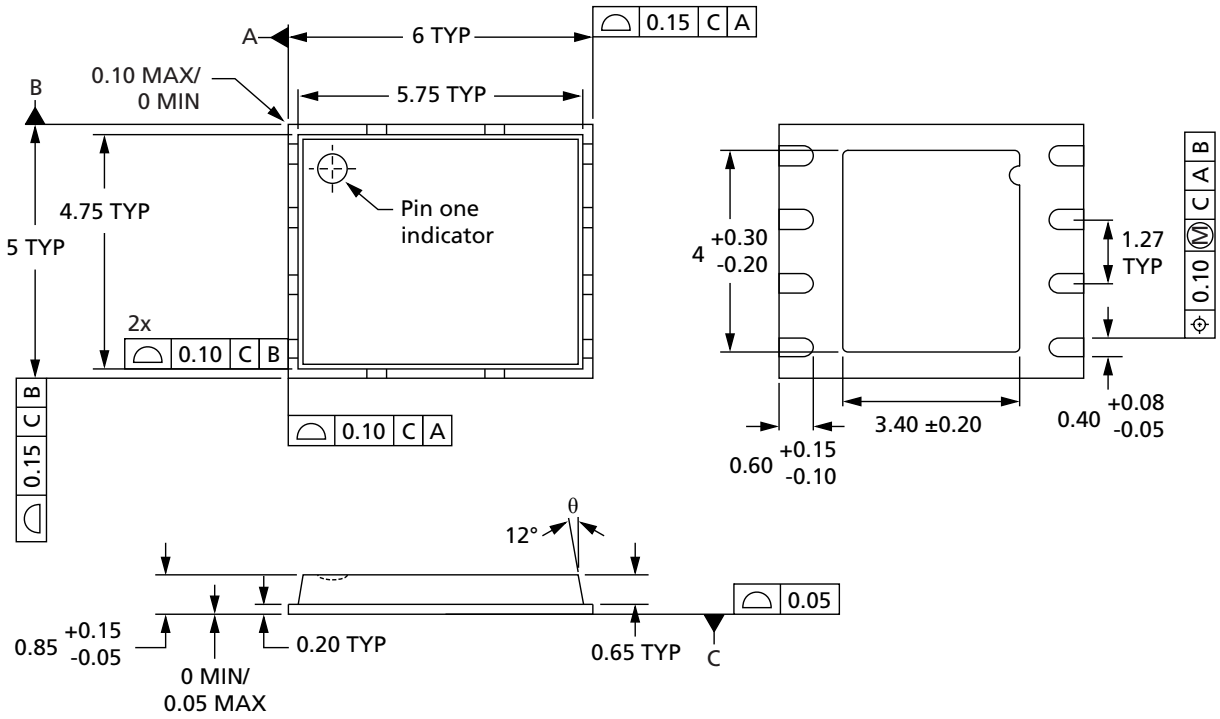


Figure 36: V_{PPH} Timing



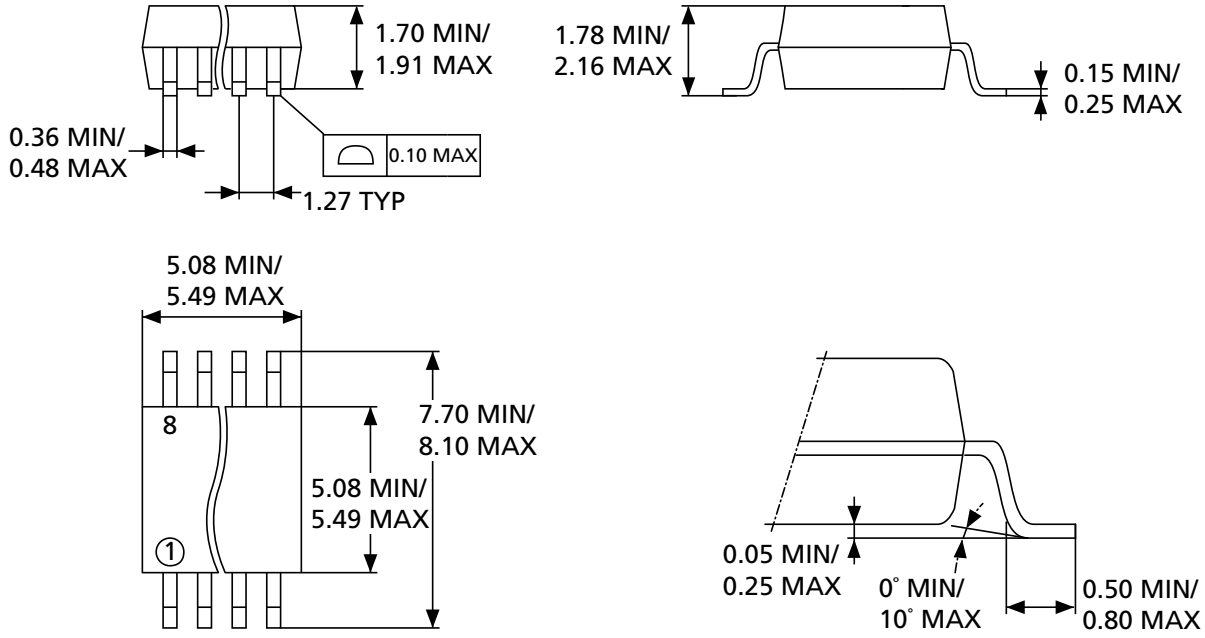
Package Information

Figure 37: VFQFPN8 (MLP8) 6mm x 5mm



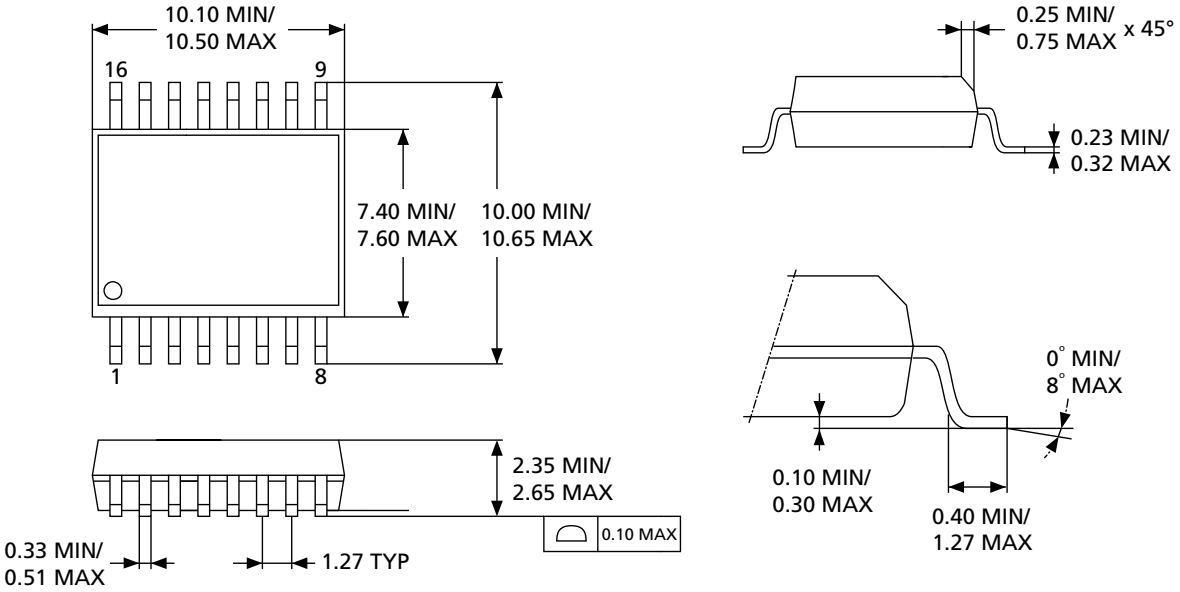
Note: 1. Drawing is not to scale.

Figure 38: SO8W 208 mils Body Width



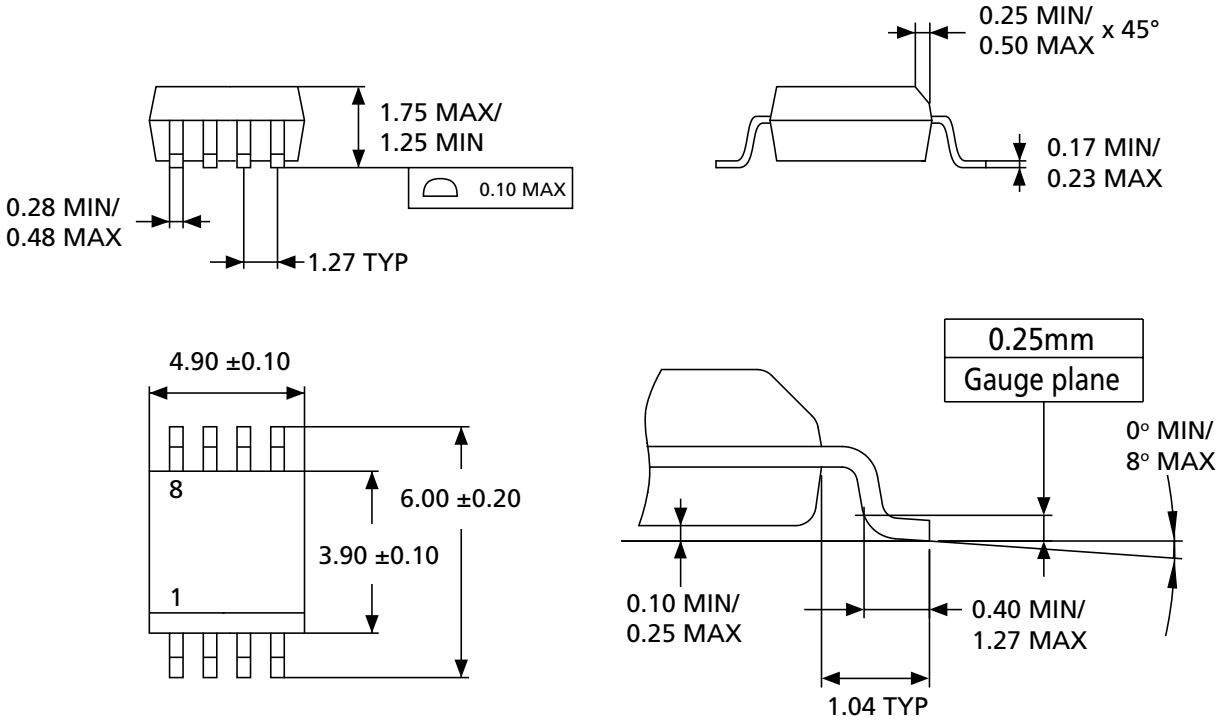
Note: 1. Drawing is not to scale.

Figure 39: SO16W 300 mils Body Width



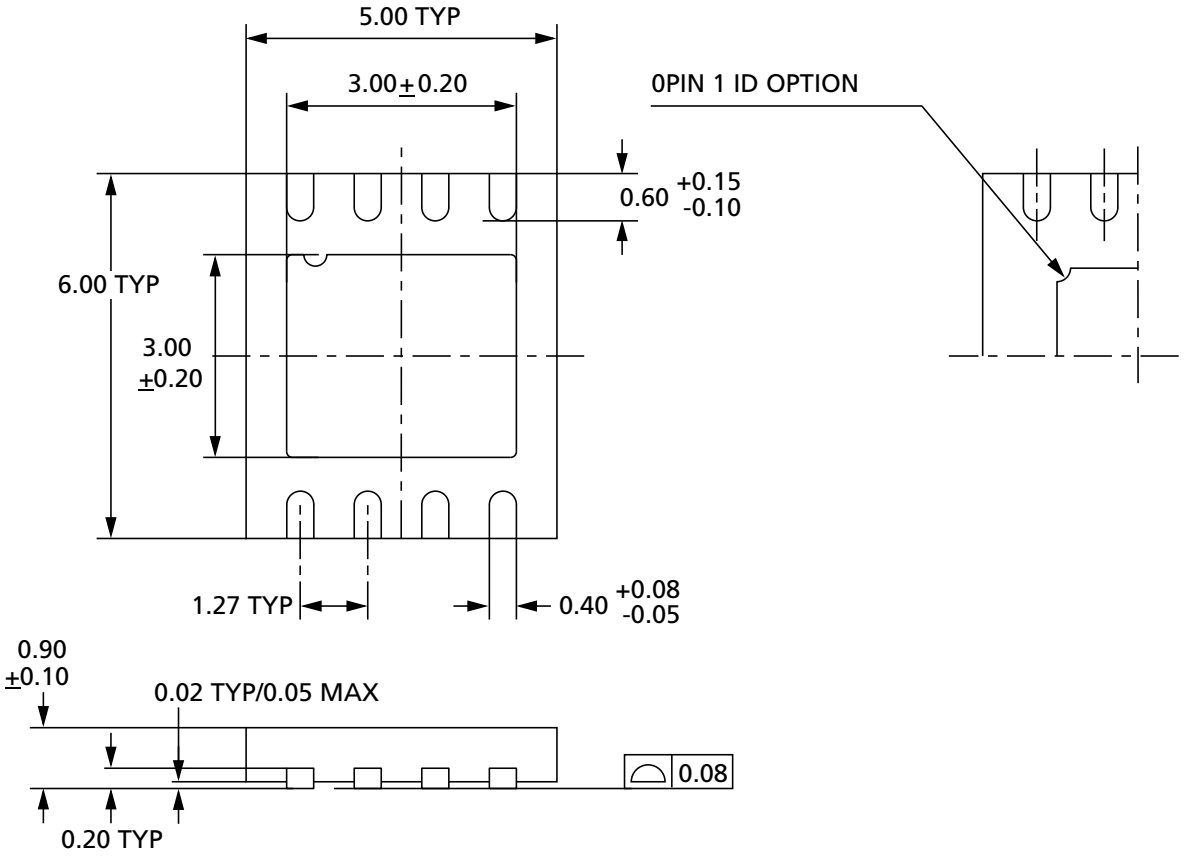
Note: 1. Drawing is not to scale.

Figure 40: SO8N 150 mils Body Width



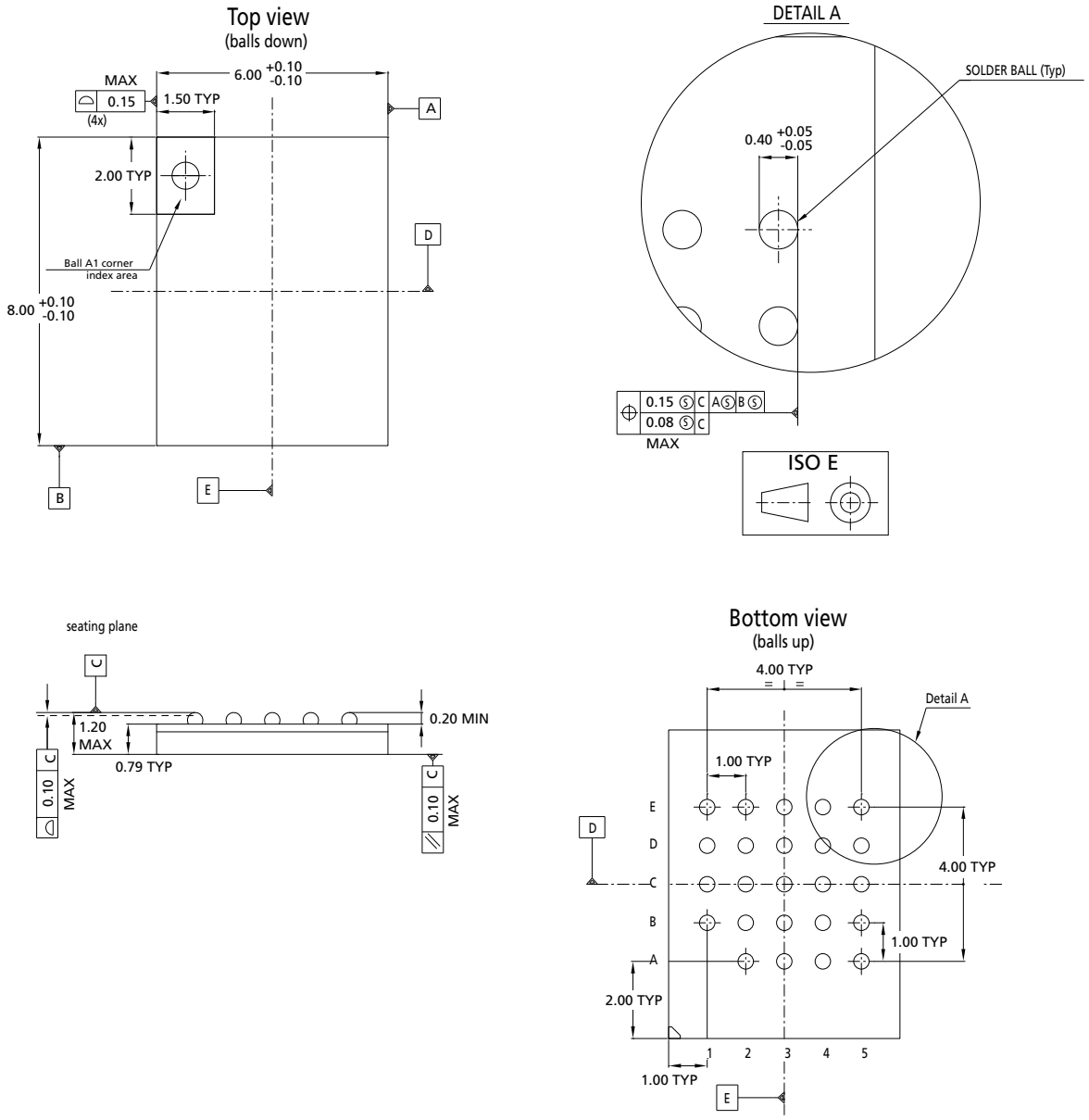
Note: 1. Drawing is not to scale.

Figure 41: QFN8L 6mm x 5mm (MLP8)



Note: 1. Drawing is not to scale.

Figure 42: TBGA 24-Ball, 6mm x 8mm



Note: 1. Drawing is not to scale.

Device Ordering Information

Micron Serial NOR Flash memory is 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. For more information on how to identify products and top-side marking by the process identification letter, refer to technical note *TN-12-24, Serial Flash Memory Device Marking for the M25P, M25PE, M25PX, and N25Q Product Families*.

Table 28: Part Number Information Scheme

Part Number Category	Category Details	Notes
Device type	M25PX = serial Flash memory, 4KB and 64KB erasable sectors, dual I/O	
Density	32 = 32Mb (4Mb x 8-bit)	
Security features	– = no extra security	1
	SO = OTP configurable	
	ST = OTP configurable plus protection at power-up	
	S = CFD programmed with UID	
Operating voltage	V = V_{CC} = 2.7 to 3.6V	
Package	MP = VFQFPN 6mm x 5mm (MLP8)	
	MW = SO8W (208 mils width)	
	MF = SO16 (300 mils width)	
	ZM = TBGA24 6mm x 8mm	
Grade	6 = Industrial temperature range: –40°C to 85°C. Device tested with standard test flow (option A is not selected). Device tested with high reliability certified test flow if automotive grade option A is selected.	
	3 = Automotive temperature range: –40°C to 125°C. Device tested with high reliability certified test flow.	2
Packing	E = Standard packing	
	F = Tape and reel packing	
Lithography	B = 110nm, Fab 2 diffusion plant	
	Blank = 110nm	
Automotive grade	A = Automotive –40°C to 85°C (only with device grade 6). Device tested with high reliability certified test flow.	2
	Blank = Automotive –40°C to 125°C	

Notes:
1. Secure options available upon customer request.
2. Micron strongly recommends the use of the Automotive Grade devices (AutoGrade 6 and Grade 3) for use in an automotive environment. The High Reliability Certified Flow (HRCF) is described in the quality note QNEE9801.

Revision History

Rev. D – 06/18

- Added Important Notes and Warnings section for further clarification aligning to industry standards

Rev. C – 10/14

- Corrected tape and reel packing information code in the Part Number Information Scheme table.

Rev. B – 03/13

- Replaced SO8W package dimension figure
- Revised text at the beginning of Ordering Information

Rev. A – 06/12

- Initial Micron release with rebrand.

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

This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein.
Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View M25PX32SOVZM6F on WIN SOURCE
- Micron Technology Inc. Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management