



**THE DATASHEET OF
M25PE40-VMW6TG TR**



M25PE40 4Mb 3V NOR Serial Flash Memory

Serial Flash Memory with Byte Alterability, 75 MHz SPI bus, Standard Pinout

Features

- 4Mb of page-erasable Flash memory
- 2.7V to 3.6V single supply voltage
- SPI bus-compatible serial interface
- 75 MHz clock rate (maximum)
- Page size: 256 bytes
 - Page write in 11ms (TYP)
 - Page program in 0.8ms (TYP)
 - Page erase in 10ms (TYP)
- Subsector erase: 4KB
 - Sector erase: 64KB
 - Bulk erase: 4Mb
- Deep power-down mode: 1 μ A (TYP)
- Electronic signature
 - JEDEC standard 2-byte signature (8013h)
- Software write-protection on a 64KB sector basis
- Hardware write protection of the memory area selected using the BP0, BP1, and BP2 bits
- More than 100,000 write cycles
- More than 20 years of data retention
- Packages (RoHS compliant)
 - VFQFPN8 (MP) 6mm x 5mm (MLP8)
 - SO8W (MW) 208 mils
 - SO8N (MN) 150 mils
- Automotive grade parts available



Contents

| | |
|--|----|
| Important Notes and Warnings | 6 |
| Functional Description | 7 |
| Signal Descriptions | 9 |
| SPI Modes | 10 |
| Operating Features | 12 |
| Sharing the overhead of modifying data | 12 |
| An easy way to modify data | 12 |
| A fast way to modify data | 12 |
| Polling during a Write, Program, or Erase Cycle | 13 |
| Reset | 13 |
| Active Power, Standby Power, and Deep Power-Down | 13 |
| Status Register | 13 |
| Protection Modes | 13 |
| Specific Hardware and Software Protection | 14 |
| Memory Organization | 16 |
| Command Set Overview | 19 |
| WRITE ENABLE | 21 |
| WRITE DISABLE | 22 |
| READ IDENTIFICATION | 23 |
| READ STATUS REGISTER | 24 |
| WIP Bit | 24 |
| WEL Bit | 24 |
| Block Protect Bits | 25 |
| Block Protect Bits | 25 |
| Block Protect Bits | 25 |
| SRWD Bit | 25 |
| SRWD Bit | 26 |
| SRWD Bit | 27 |
| WRITE STATUS REGISTER | 28 |
| READ DATA BYTES | 30 |
| READ DATA BYTES at HIGHER SPEED | 31 |
| READ LOCK REGISTER | 32 |
| PAGE WRITE | 34 |
| PAGE PROGRAM | 36 |
| WRITE to LOCK REGISTER | 38 |
| PAGE ERASE | 41 |
| SUBSECTOR ERASE | 42 |
| SECTOR ERASE | 43 |
| BULK ERASE | 44 |
| DEEP POWER-DOWN | 45 |
| RELEASE from DEEP POWER-DOWN | 46 |
| Power-Up and Power-Down | 47 |
| RESET | 49 |
| Initial Delivery State | 49 |
| Maximum Ratings and Operating Conditions | 50 |
| DC Parameters | 51 |
| AC Characteristics | 52 |
| Package Information | 58 |
| Device Ordering Information | 60 |
| Revision History | 62 |



M25PE40 Serial Flash Memory Features

| | |
|------------------------|----|
| Rev. D – 1/2018 | 62 |
| Rev. C – 3/2013 | 62 |
| Rev. B – 11/2012 | 62 |
| Rev. A – 09/2012 | 62 |

List of Figures

| | |
|---|----|
| Figure 1: Logic Diagram | 7 |
| Figure 2: Pin Connections: VFQFPN and SO | 8 |
| Figure 3: SPI Modes Supported | 10 |
| Figure 4: Bus Master and Memory Devices on the SPI Bus | 11 |
| Figure 5: Block Diagram | 18 |
| Figure 6: WRITE ENABLE Command Sequence | 21 |
| Figure 7: WRITE DISABLE Command Sequence | 22 |
| Figure 8: READ IDENTIFICATION Command Sequence | 24 |
| Figure 9: READ STATUS REGISTER Command Sequence | 24 |
| Figure 10: Status Register Format | 26 |
| Figure 11: Status Register Format | 26 |
| Figure 12: Status Register Format | 27 |
| Figure 13: WRITE STATUS REGISTER Command Sequence | 28 |
| Figure 14: READ DATA BYTES Command Sequence | 30 |
| Figure 15: READ DATA BYTES at HIGHER SPEED Command Sequence | 31 |
| Figure 16: READ LOCK REGISTER Command Sequence | 32 |
| Figure 17: PAGE WRITE Command Sequence | 35 |
| Figure 18: PAGE PROGRAM Command Sequence | 37 |
| Figure 19: WRITE to LOCK REGISTER Instruction Sequence | 38 |
| Figure 20: PAGE ERASE Command Sequence | 41 |
| Figure 21: SUBSECTOR ERASE Command Sequence | 42 |
| Figure 22: SECTOR ERASE Command Sequence | 43 |
| Figure 23: BULK ERASE Command Sequence | 44 |
| Figure 24: DEEP POWER-DOWN Command Sequence | 45 |
| Figure 25: RELEASE from DEEP POWER-DOWN Command Sequence | 46 |
| Figure 26: Power-Up Timing | 48 |
| Figure 27: AC Measurement I/O Waveform | 52 |
| Figure 28: Serial Input Timing | 55 |
| Figure 29: Write Protect Setup and Hold Timing | 55 |
| Figure 30: Output Timing | 56 |
| Figure 31: Reset AC Waveforms | 57 |
| Figure 32: VFQFPN8 (MLP8) 6mm x 5mm | 58 |
| Figure 33: SO8N – 8 lead plastic small outline, 150 mils body width | 59 |
| Figure 34: SO8W – 8 lead plastic small outline, 208 mils body width | 59 |

List of Tables

| | |
|---|----|
| Table 1: Signal Names | 8 |
| Table 2: Signal Descriptions | 9 |
| Table 3: Software Protection Truth Table, 64KB granularity (sectors 0-7) | 15 |
| Table 4: Protected Area Sizes | 15 |
| Table 5: Memory Organization | 16 |
| Table 6: Command Set Codes | 20 |
| Table 7: READ IDENTIFICATION Data Out Sequence | 23 |
| Table 8: READ IDENTIFICATION Data Out Sequence | 23 |
| Table 9: READ IDENTIFICATION Data Out Sequence | 23 |
| Table 10: Status Register Protection Modes | 29 |
| Table 11: Lock Register Out | 32 |
| Table 12: Not for new design: Lock Registers for the M25PE80 in T7Y Process | 33 |
| Table 13: Lock Register In | 38 |
| Table 14: Lock Register In | 38 |
| Table 15: Lock Register In | 39 |
| Table 16: Not for new design: lock registers for the M25PE80 in T7Y process | 39 |
| Table 17: Power-up Timing and V_{WI} Threshold | 48 |
| Table 18: Device Status After a RESET# LOW Pulse | 49 |
| Table 19: Absolute Maximum Ratings | 50 |
| Table 20: Operating Conditions | 50 |
| Table 21: DC Characteristics 75 MHz Operation | 51 |
| Table 22: AC Measurement Conditions | 52 |
| Table 23: Capacitance | 52 |
| Table 24: AC Specifications (50 MHz operation) | 53 |
| Table 25: AC Specifications (75MHz operation) | 54 |
| Table 26: Reset Conditions | 56 |
| Table 27: Timings After a RESET# LOW Pulse | 56 |
| Table 28: Standard Part Number Information Scheme | 60 |
| Table 29: Automotive Part Number Information Scheme | 60 |

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 M25PE40 is a 4Mb (512Kb x 8 bit) serial-paged Flash memory device accessed by a high-speed SPI-compatible bus.

The memory can be written or programmed 1 to 256 bytes at a time using the PAGE WRITE or PAGE PROGRAM command. The PAGE WRITE command consists of an integrated PAGE ERASE cycle followed by a PAGE PROGRAM cycle.

The memory is organized as 8 sectors, divided into 16 subsectors each (128 subsectors total). Each sector contains 256 pages and each subsector contains 16 pages. Each page is 256 bytes wide. The entire memory can be viewed as consisting of 2048 pages, or 524,288 bytes.

The memory can be erased one page at a time using the PAGE ERASE command, one sector at a time using the SECTOR ERASE command, one subsector at a time using the SUBSECTOR ERASE command, or as a whole using the BULK ERASE command.

The memory can be write-protected by either hardware or software using a mix of volatile and non-volatile protection features, depending on application needs. The protection granularity is 64Kb (sector granularity). The entire memory array is partitioned into 4KB subsectors.

Figure 1: Logic Diagram

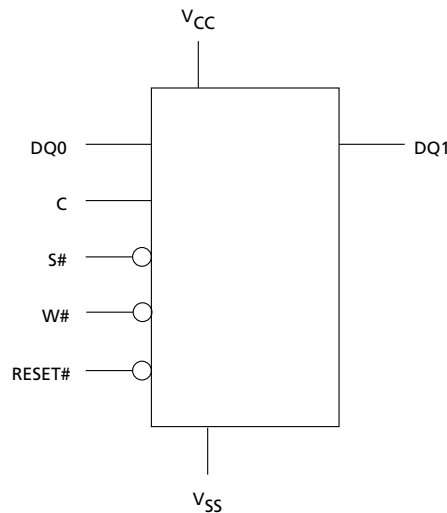
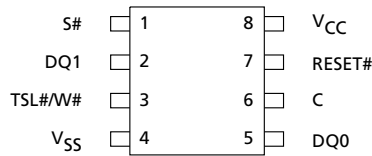


Table 1: Signal Names

| Signal Name | Function | Direction |
|-----------------|----------------|-----------|
| C | Serial clock | Input |
| DQ0 | Serial data | Input |
| DQ1 | Serial data | Output |
| S# | Chip select | Input |
| W# | Write Protect | Input |
| RESET# | Reset | Input |
| V _{CC} | Supply voltage | – |
| V _{SS} | Ground | – |

Figure 2: Pin Connections: VFQFPN and SO



There is an exposed central pad on the underside of the VFQFPN package that 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.

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). |
| 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). |
| 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 READ, PROGRAM, ERASE, or WRITE 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. |
| RESET# | Input | Reset: The RESET# input provides a hardware reset for the memory. When RESET# is driven HIGH, the memory is in the normal operating mode. When RESET# is driven LOW, the memory will enter the Reset mode. In this mode, the output is at HIGH impedance. Driving RESET# LOW while an internal operation is in progress affects the WRITE, PROGRAM, or ERASE cycle, and data may be lost. |
| W# | Input | Write protect: The W# input signal is used to freeze the size of the area of memory that is protected against WRITE, PROGRAM, and ERASE commands as specified by the values in the block protect bits in the status register. |
| V _{CC} | Input | Supply voltage |
| V _{SS} | Input | Ground: Reference for the VCC supply voltage. |

SPI Modes

These devices can be driven by a microcontroller with its serial peripheral interface (SPI) running in either of the following two SPI modes:

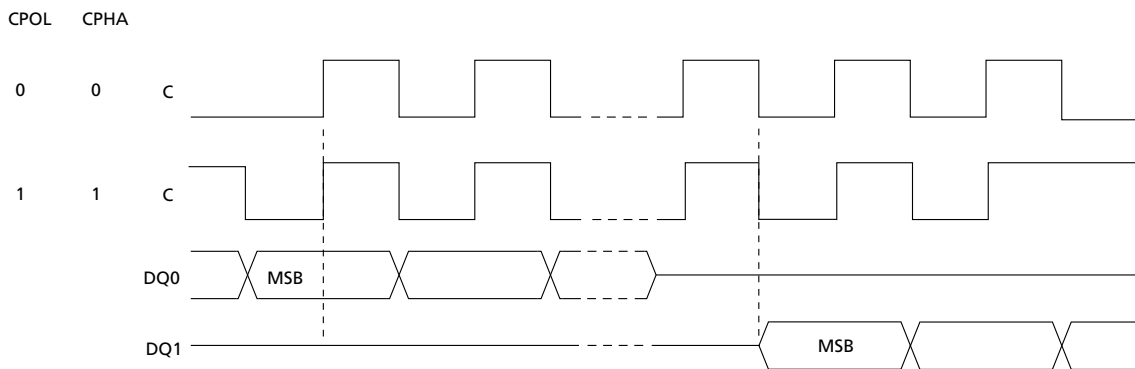
- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

For these two modes, input data is latched in on the rising edge of serial clock (C), and output data is available from the falling edge of C.

The difference between the two modes is the clock polarity when the bus master is in STANDBY mode and not transferring data:

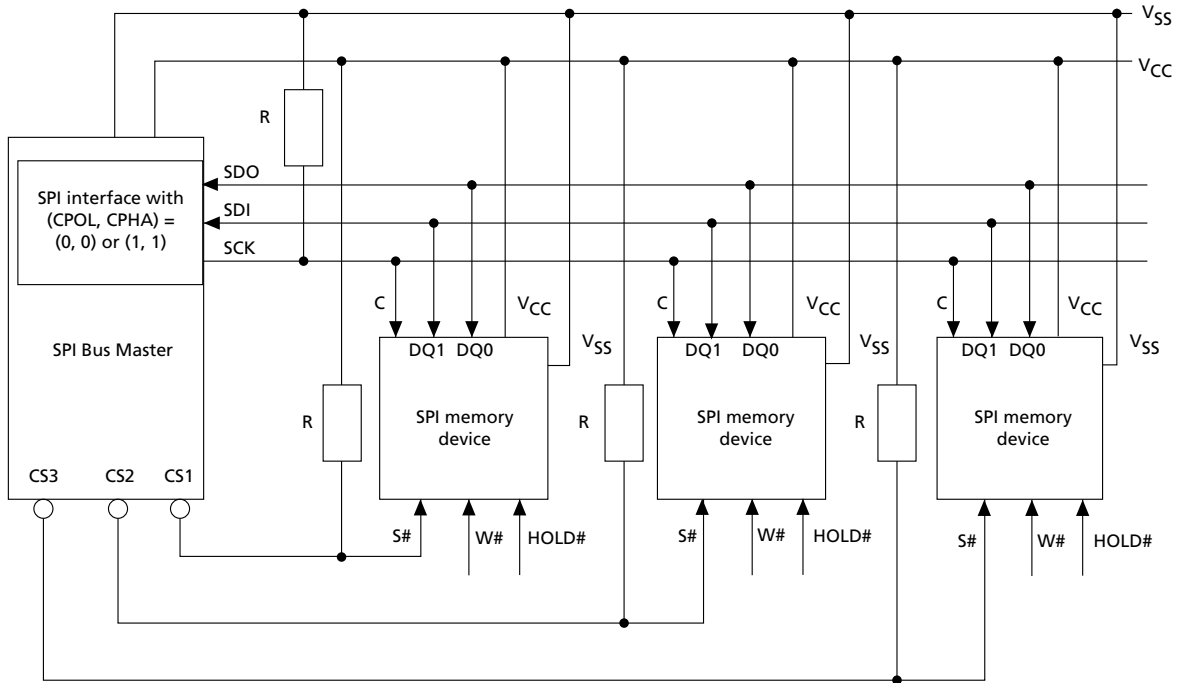
- C remains at 0 for (CPOL=0, CPHA=0)
- C remains at 1 for (CPOL=1, CPHA=1)

Figure 3: SPI Modes Supported



Because only one device is selected at a time, only one device drives the serial data output (DQ1) line at a time, while the other devices are HIGH-Z. An example of three devices connected to an MCU on an SPI bus is shown here.

Figure 4: Bus Master and Memory Devices on the SPI Bus



- Notes:
1. WRITE PROTECT (W#) and HOLD# should be driven HIGH or LOW as appropriate.
 2. Resistors (R) ensure that the memory device is not selected if the bus master leaves the S# line HIGH-Z.
 3. The bus master may enter a state where all I/O are HIGH-Z at the same time; for example, when the bus master is reset. Therefore, the C must be connected to an external pull-down resistor so that when all I/O are HIGH-Z, S# is pulled HIGH while C is pulled LOW. This ensures that S# and C do not go HIGH at the same time and that the ^tSHCH requirement is met.
 4. The typical value of R is 100 kΩ, assuming that the time constant $R \times C_p$ (C_p = parasitic capacitance of the bus line) is shorter than the time during which the bus master leaves the SPI bus HIGH-Z.
 5. Example: Given that $C_p = 50$ pF ($R \times C_p = 5\mu$ s), the application must ensure that the bus master never leaves the SPI bus HIGH-Z for a time period shorter than 5μs.

Operating Features

Sharing the overhead of modifying data

To write or program one or more data bytes, two commands are required: WRITE ENABLE (WREN), which is one byte, and a PAGE WRITE (PW) or PAGE PROGRAM (PP) sequence, which consists of four bytes plus data. This is followed by the internal cycle of duration t_{PW} or t_{PP} .

To share this overhead, the PW or PP command allows up to 256 bytes to be programmed (changing bits from 1 to 0) or written (changing bits to 0 or 1) at a time, provided that they lie in consecutive addresses on the same page of memory.

An easy way to modify data

The Page Write (PW) instruction provides a convenient way of modifying data (up to 256 contiguous bytes at a time), and simply requires the start address, and the new data in the instruction sequence.

The Page Write (PW) instruction is entered by driving Chip Select (S#) LOW, and then transmitting the instruction byte, three address bytes (A23-A0) and at least one data byte, and then driving S# HIGH. While S# is being held LOW, the data bytes are written to the data buffer, starting at the address given in the third address byte (A7-A0). When Chip S# is driven HIGH, the Write cycle starts. The remaining unchanged bytes of the data buffer are automatically loaded with the values of the corresponding bytes of the addressed memory page. The addressed memory page is then automatically put into an Erase cycle. Finally, the addressed memory page is programmed with the contents of the data buffer.

All of this buffer management is handled internally, and is transparent to the user. The user is given the facility of being able to alter the contents of the memory on a byte-by-byte basis. For optimized timings, it is recommended to use the PAGE WRITE (PW) instruction to write all consecutive targeted bytes in a single sequence versus using several PAGE WRITE (PW) sequences with each containing only a few bytes.

A fast way to modify data

The PAGE PROGRAM (PP) command provides a fast way of modifying data (up to 256 contiguous bytes at a time), provided that it only involves resetting bits to 0 that had previously been set to 1.

This might be:

- When the designer is programming the device for the first time.
- When the designer knows that the page has already been erased by an earlier PAGE ERASE (PE), SUBSECTOR ERASE (SSE), SECTOR ERASE (SE), or BULK ERASE (BE) command. This is useful, for example, when storing a fast stream of data, having first performed the erase cycle when time was available.
- When the designer knows that the only changes involve resetting bits to 0 that are still set to 1. When this method is possible, it has the additional advantage of minimizing the number of unnecessary erase operations, and the extra stress incurred by each page.

For optimized timings, it is recommended to use the PAGE PROGRAM (PP) instruction to program all consecutive targeted bytes in a single sequence versus using several PAGE PROGRAM (PP) sequences with each containing only a few bytes.

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_{PW} , t_{PP} , t_{PE} , t_{BE} , t_{WOR} or t_{SE}).

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.

Reset

An internal power-on reset circuit helps protect against inadvertent data writes. Additional protection is provided by driving RESET# LOW during the power-on process, and only driving it HIGH when V_{CC} has reached the correct voltage level, $V_{CC}(\min)$.

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). 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.

Status Register

The status register contains a number of status bits that can be read by the READ STATUS REGISTER (RDSR) command.

Protection Modes

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_{PIW}) can provide protection against inadvertent changes while the power supply is outside the operating specification.

WRITE, PROGRAM, and ERASE 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. This bit is returned to its reset state by the following events.

- Power-up
- Reset (RESET#) driven LOW

- WRITE DISABLE (WRDI) command completion
- PAGE WRITE (PW) command completion
- PAGE PROGRAM (PP) command completion
- WRITE TO LOCK REGISTER (WRLR) command completion
- PAGE ERASE (PE) command completion
- SUBSECTOR ERASE (SSE) command completion
- SECTOR ERASE (SE) command completion
- BULK ERASE (BE) command completion

The Reset (RESET#) signal can be driven LOW to freeze and reset the internal logic.

In addition to the low power consumption feature, DEEP POWER-DOWN mode offers extra software protection from inadvertent WRITE, PROGRAM, and ERASE commands while the device is not in active use.

Specific Hardware and Software Protection

The device features a hardware protected mode (HPM) and two software protected modes (SPM1 and SPM2). SPM1 and SPM2 can be combined to protect the memory array as required.

Hardware Protected Mode (HPM)

The Hardware Protected Mode (HPM) is used to write-protect the non-volatile bits of the Status Register (that is, the Block Protect Bits and the Status Register bit). HPM is entered by driving the Write Protect (W#) signal LOW with the SRWD bit set to HIGH. This additional protection allows the Status Register to be hardware-protected.

SPM1 and SPM2

The first Software Protected Mode (SPM1) is managed by specific Lock Registers assigned to each 64KB sector.

The Lock Registers can be read and written using the Read Lock Register (RDLR) and Write to Lock Register (WRLR) commands.

In each Lock Register, two bits control the protection of each sector: the Write Lock bit and the Lock Down bit.

- 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 Write Lock bit and the Lock Down bit are volatile and their value is reset to 0 after a power-down or reset.

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

Table 3: Software Protection Truth Table, 64KB granularity (sectors 0-7)

| Sector Lock Register: Lock Down Bit | Sector Lock Register: Write Lock Bit | 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 reset or power-up. |
| 1 | 1 | Sector protected from PROGRAM / ERASE / WRITE operations; protection status cannot be changed except by a reset or power-up. |

The second Software Protected Mode (SPM2) uses the block protect (BP2, BP1, BP0) bits to allow part of the memory to be configured as read-only.

Table 4: Protected Area Sizes

| Status Register Content | | | Memory Content | | Notes |
|-------------------------|----------|----------|---------------------------------|---------------------------------------|-------|
| BP Bit 2 | BP Bit 1 | BP Bit 0 | Protected Area | Unprotected Area | |
| 0 | 0 | 0 | none | All sectors (sectors 0 to 7) | 1 |
| 0 | 0 | 1 | Upper eighth (sector 7) | Lower 7/8ths (sectors 0 to 6) | |
| 0 | 1 | 0 | Upper quarter (sectors 6 and 7) | Lower three-quarters (sectors 0 to 5) | |
| 0 | 1 | 1 | Upper half (sectors 4 to 7) | Lower half (sectors 0 to 3) | |
| 1 | 0 | 0 | All sectors (sectors 0 to 7) | none | |
| 1 | 0 | 1 | All sectors (sectors 0 to 7) | none | |
| 1 | 1 | 0 | All sectors (sectors 0 to 7) | none | |
| 1 | 1 | 1 | All sectors (sectors 0 to 7) | none | |

Note: 1. The device is ready to accept a BULK ERASE command only if all block protect bits (BP2, BP1, BP0) are 0.

Memory Organization

The M25PE40 memory is organized as follows:

- 2048 pages (256 bytes each)
- 524,288 bytes (8 bits each)
- 128 subsectors (32Kb, 4096 bytes each)
- 8 sectors (512Kb, 65,536 bytes each)

Each page can be individually:

- programmed (bits are programmed from 1 to 0)
- erased (bits are erased from 0 to 1)
- written (bits are changed to either 0 or 1)

The device is page- or sector-erasable (bits are erased from 0 to 1).

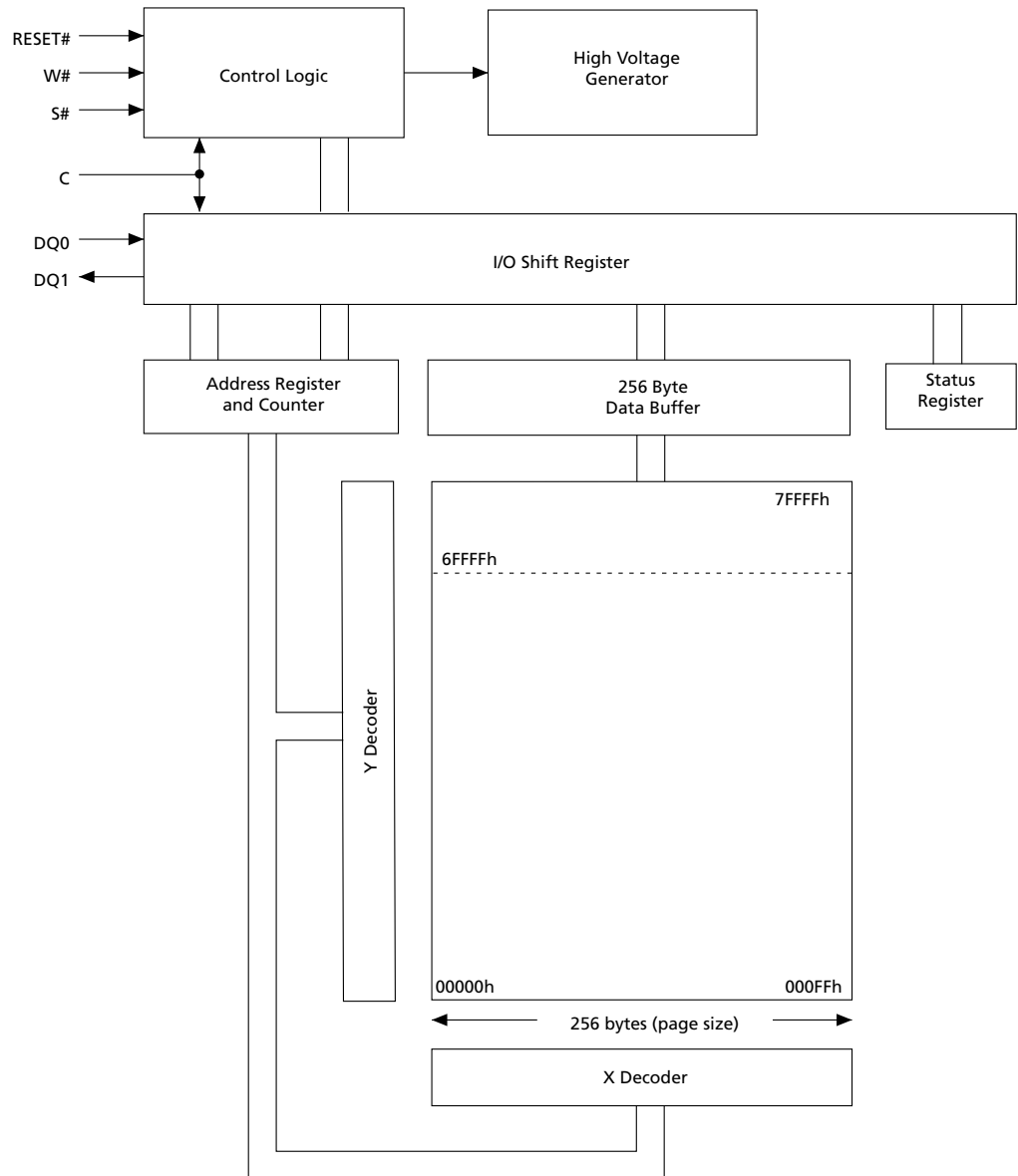
Table 5: Memory Organization

| Sector | Subsector | Address Range | |
|--------|-----------|---------------|--------|
| | | Start | End |
| 7 | 127 | 7F000h | 7FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 112 | 70000h | 70FFFh |
| 6 | 111 | 6F00h | 6FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 96 | 60000h | 60FFFh |
| 5 | 95 | 5F000h | 5FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 80 | 50000h | 50FFFh |
| 4 | 79 | 4F000h | 4FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 64 | 40000h | 40FFFh |
| 3 | 63 | 3F000h | 3FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 48 | 30000h | 30FFFh |
| 2 | 47 | 2F000h | 2FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 32 | 20000h | 20FFFh |
| 1 | 31 | 1F000h | 1FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 16 | 10000h | 10FFFh |

Table 5: Memory Organization (Continued)

| Sector | Subsector | Address Range | |
|--------|-----------|---------------|--------|
| | | Start | End |
| 0 | 15 | 0F000h | 0FFFFh |
| | ⋮ | ⋮ | ⋮ |
| | 4 | 04000h | 04FFFh |
| | 3 | 03000h | 03FFFh |
| | 2 | 02000h | 02FFFh |
| | 1 | 01000h | 01FFFh |
| | 0 | 00000h | 00FFFh |

Figure 5: Block Diagram



Note: 1. Entire memory array can be made read-only on a 64KB basis through the lock registers.

Command Set Overview

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

Serial data input (DQ1) is 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 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
- READ STATUS REGISTER
- READ TO LOCK REGISTER

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 WRITE
- PAGE PROGRAM
- PAGE ERASE
- SUBSECTOR ERASE
- SECTOR ERASE
- BULK ERASE
- WRITE ENABLE
- WRITE DISABLE
- WRITE STATUS REGISTER
- WRITE TO LOCK REGISTER
- DEEP POWER-DOWN
- RELEASE FROM DEEP POWER-DOWN

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



Table 6: 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 |
| 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 ∞ |
| PAGE WRITE | 0000 1010 | 0Ah | 3 | 0 | 1 to 256 |
| PAGE PROGRAM | 0000 0010 | 02h | 3 | 0 | 1 to 256 |
| PAGE ERASE | 1101 1011 | DBh | 3 | 0 | 0 |
| 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.

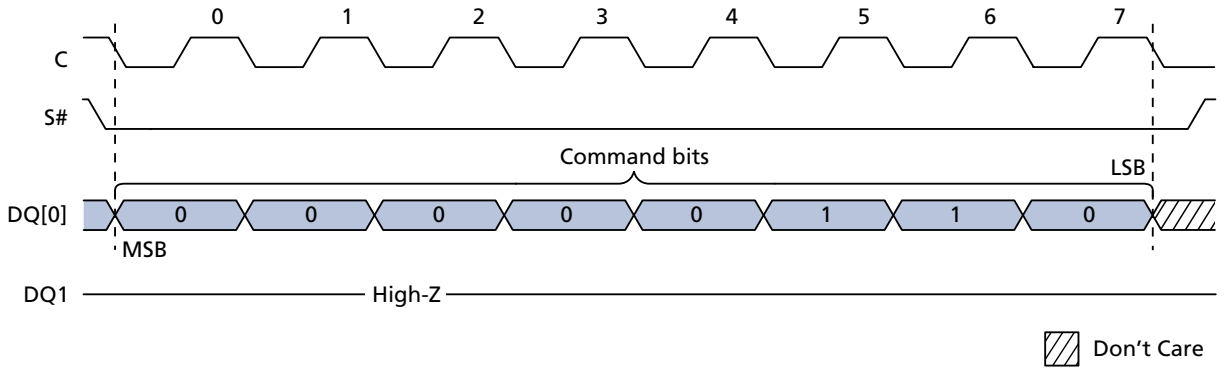
The WEL bit must be set before execution of every PAGE WRITE, PAGE PROGRAM, PAGE ERASE, SECTOR ERASE, SUBSECTOR ERASE, BULK ERASE, WRITE STATUS REGISTER, and WRITE to LOCK REGISTER command.

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

The WEL bit must be set before execution of every PAGE WRITE, PAGE PROGRAM, PAGE ERASE, SECTOR ERASE, BULK ERASE, and WRITE to LOCK REGISTER command.

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

Figure 6: WRITE ENABLE Command Sequence



WRITE DISABLE

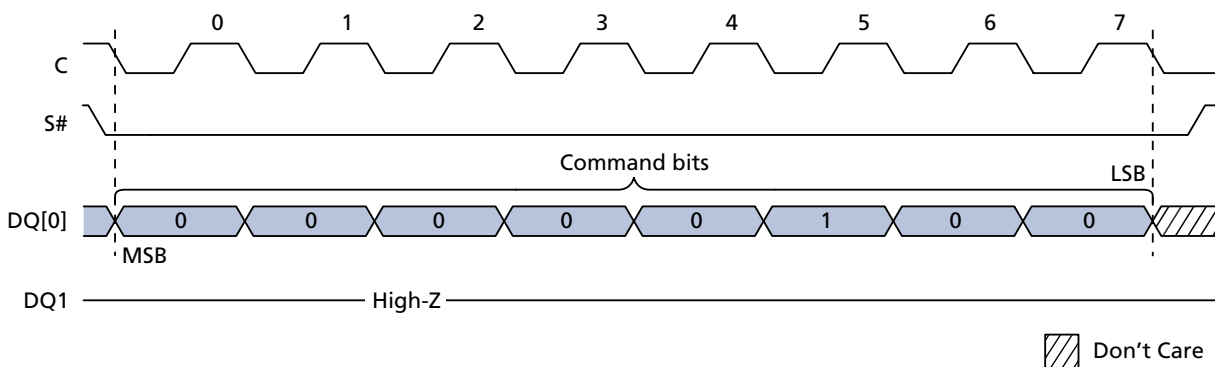
The WRITE DISABLE command resets the write enable latch (WEL) bit.

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 WRITE DISABLE operation
- Completion of PAGE WRITE operation
- Completion of PAGE PROGRAM operation
- Completion of WRITE STATUS REGISTER operation
- Completion of WRITE TO LOCK REGISTER operation
- Completion of PAGE ERASE operation
- Completion of SUBSECTOR ERASE operation
- Completion of SECTOR ERASE operation
- Completion of BULK ERASE operation

Figure 7: 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 7: READ IDENTIFICATION Data Out Sequence

| Manufacturer Identification | Device Identification | | UID | |
|-----------------------------|-----------------------|-----------------|------------|-------------|
| | Memory Type | Memory Capacity | CFD Length | CFD Content |
| 20h | 80h | 15h | 10h | 16 bytes |

Table 8: READ IDENTIFICATION Data Out Sequence

| Manufacturer Identification | Device Identification | |
|-----------------------------|-----------------------|-----------------|
| | Memory Type | Memory Capacity |
| 20h | 80h | 13h |

Table 9: READ IDENTIFICATION Data Out Sequence

| Manufacturer Identification | Device Identification | | UID | |
|-----------------------------|-----------------------|-----------------|------------|-------------|
| | Memory Type | Memory Capacity | CFD Length | CFD Content |
| 20h | 80h | 14h | 10h | 16 bytes |

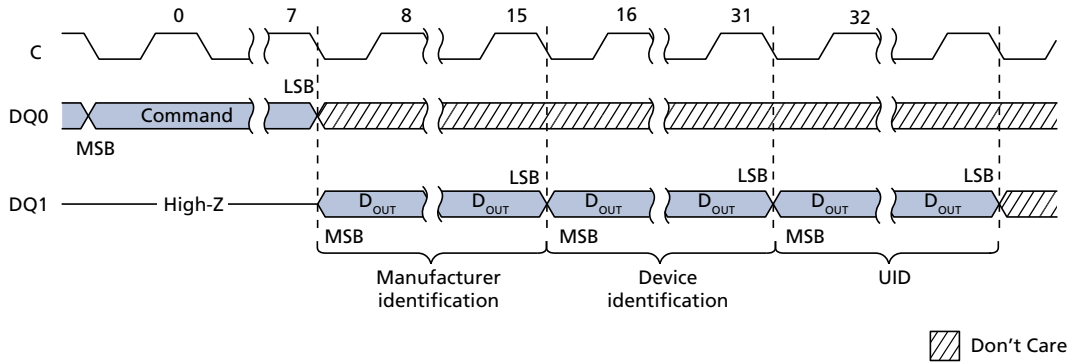
Note: 1. The CFD bytes are read-only and can be programmed with customer data upon demand. If customers do not make requests, the devices are shipped with all the CFD bytes programmed to zero (00h).

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 device is first selected by driving chip select (S#) LOW. Then, the 8-bit command code is shifted in and content is shifted out on serial data output (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.

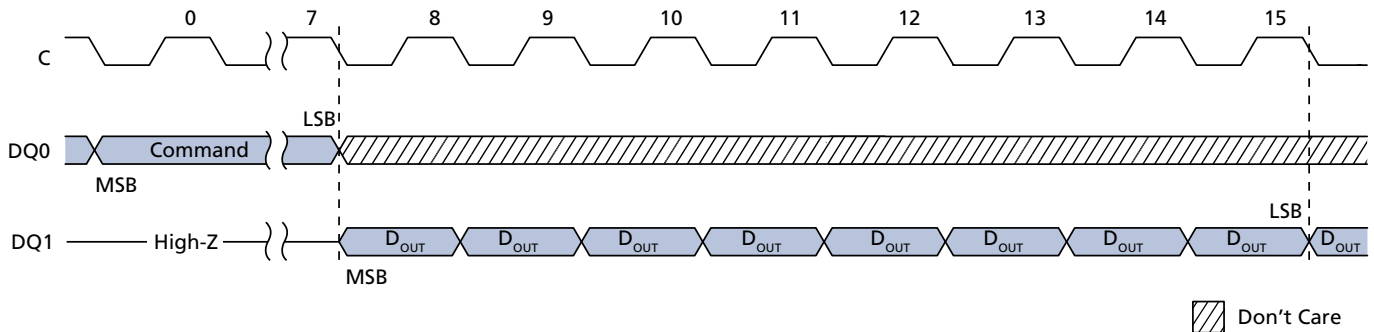
Figure 8: READ IDENTIFICATION Command Sequence



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 9: READ STATUS REGISTER Command Sequence



WIP Bit

The write in progress (WIP) bit indicates whether the memory is busy with a WRITE 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, PROGRAM, or ERASE command is accepted.

Block Protect Bits

The block protect (BP2, BP1, BP0) bits are non-volatile. 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 (BP2, BP1, BP0) bits is set to 1, the relevant memory area, as defined in the Protected Area Sizes table, becomes protected against PAGE PROGRAM, SECTOR ERASE, and SUBSECTOR ERASE commands. The block protect BP2, BP1, BP0) 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 (BP2, BP1, BP0) bits are 0 and the Lock Register protection bits are not all set to 1.

Block Protect Bits

The block protect (BP2, BP1, BP0) bits are non-volatile. 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 (BP2, BP1, BP0) bits is set to 1, the relevant memory area, as defined in the Protected Area Sizes table, becomes protected against PAGE PROGRAM, SECTOR ERASE, and SUBSECTOR ERASE commands. The block protect BP2, BP1, BP0) 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 (BP2, BP1, BP0) bits are 0 and the Lock Register protection bits are not all set to 1.

Block Protect Bits

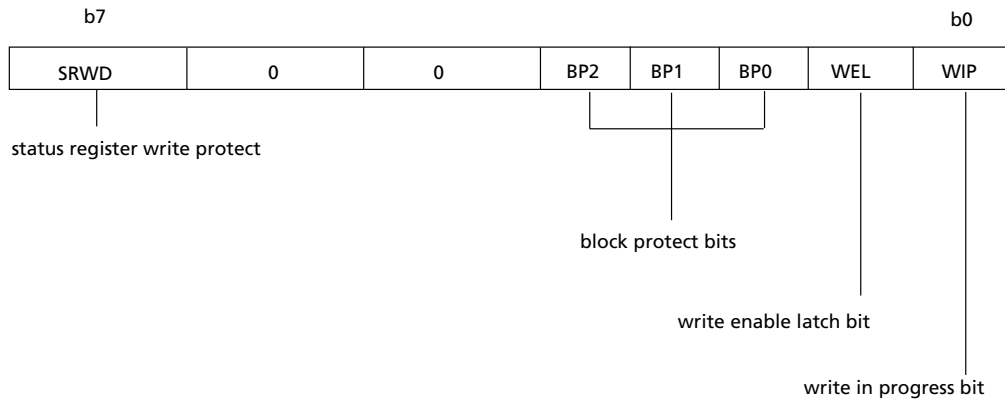
The block protect (BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against PROGRAM and ERASE commands. The block protect bits are written with the WRITE STATUS REGISTER command.

When one or more of the block protect (BP2, BP1, BP0) bits is set to 1, the relevant memory area, as defined in the Protected Area Sizes table, becomes protected against PAGE PROGRAM, SECTOR ERASE, and SUBSECTOR ERASE commands. The block protect BP2, BP1, BP0) 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 (BP2, BP1, BP0) bits are 0 and the Lock Register protection bits are not all set to 1.

SRWD Bit

The status register write disable (SRWD) bit is operated in conjunction with the write protect (W#) signal. When the SRWD bit is set to 1 and W# is driven LOW, the device is put in the hardware protected mode. In the hardware protected mode, the non-volatile bits of the status register (SRWD, BP2, BP1, BP0) become read-only bits and the WRITE STATUS REGISTER command is no longer accepted for execution.

Figure 10: Status Register Format

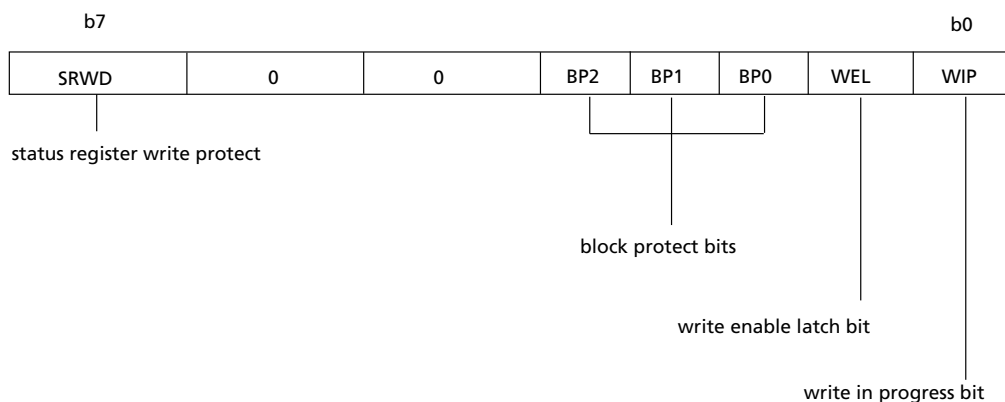


Note: WEL and WIP are volatile read-only bits (WEL is set and reset by specific instructions; WIP is automatically set and reset by the internal logic of the device). SRWD = status register write protect bit; BP0, BP1, BP2 = block protect bits.

SRWD Bit

The status register write disable (SRWD) bit is operated in conjunction with the write protect ($W\#$) signal. When the SRWD bit is set to 1 and $W\#$ is driven LOW, the device is put in the hardware protected mode. In the hardware protected mode, the non-volatile bits of the status register (SRWD, BP2, BP1, BP0) become read-only bits and the WRITE STATUS REGISTER command is no longer accepted for execution.

Figure 11: Status Register Format

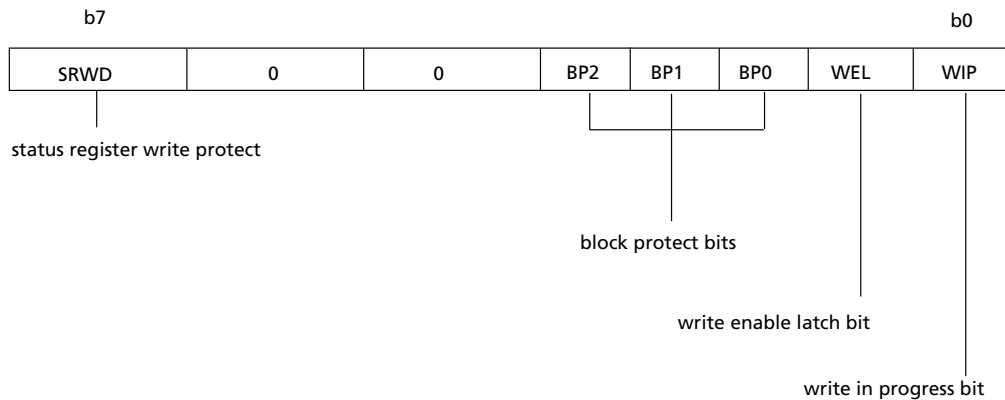


Note: WEL and WIP are volatile read-only bits (WEL is set and reset by specific instructions; WIP is automatically set and reset by the internal logic of the device). SRWD = status register write protect bit; BP0, BP1, BP2 = block protect bits.

SRWD Bit

The status register write disable (SRWD) bit is operated in conjunction with the write protect (W#) signal. When the SRWD bit is set to 1 and W# is driven LOW, the device is put in the hardware protected mode. In the hardware protected mode, the non-volatile bits of the status register (SRWD, BP2, BP1, BP0) become read-only bits and the WRITE STATUS REGISTER command is no longer accepted for execution.

Figure 12: Status Register Format



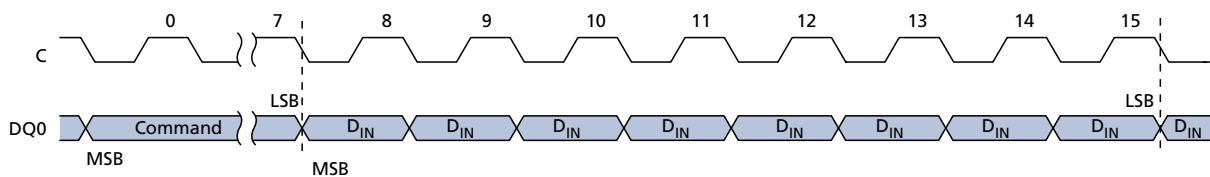
Note: WEL and WIP are volatile read-only bits (WEL is set and reset by specific instructions; WIP is automatically set and rest by the internal logic of the device). SRWD = status register write protect bit; BP0, BP1, BP2 = block protect bits.

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, 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 13: 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. 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#) signal. The SRWD bit and the W# 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 10: Status Register Protection Modes

| W# Signal | SRWD Bit | Protection Mode (PM) | Status Register Write Protection | Memory Content | |
|--------------|-------------|---|-------------------------------------|--------------------------------|----------------------------------|
| | | | | Protected Area ¹ | Unprotected Area ¹ |
| 1 | 0 | SECOND SOFTWARE PROTECTED mode (SPM2) | Software protection | Commands not accepted | Commands accepted |
| 0 | 0 | | | | |
| 1 | 1 | | | | |
| 0 | 1 | HARDWARE PROTECTED mode (HPM) | Hardware protection | Commands not accepted | Commands accepted |

Note: 1. As defined by the values in the Block Protect bits of the status register.

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# 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# signal:

- If the W# 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# 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 SPM2 by the status register block protect bits (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# signal LOW
- Driving the W# signal LOW after setting the status register SRWD bit.

The only way to exit the HPM is to pull the W# signal HIGH. If the W# signal is permanently tied HIGH, the HPM can never be activated. In this case, only the SPM2 is available, using the status register block protect bits.

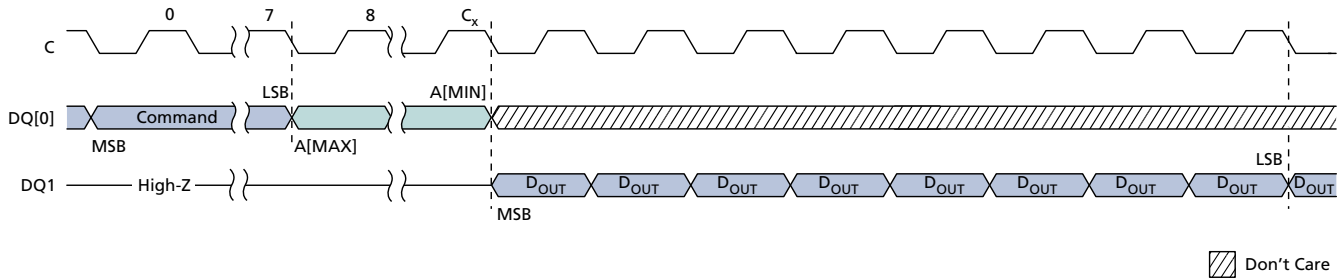
READ DATA BYTES

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 14: READ DATA BYTES Command Sequence



- Notes:
1. Address bits A₂₃-A₂₁ are *don't care*.
 2. Address bits A₂₃-A₁₉ are *don't care*.
 3. Address bits A₂₃-A₂₀ are *don't care*.

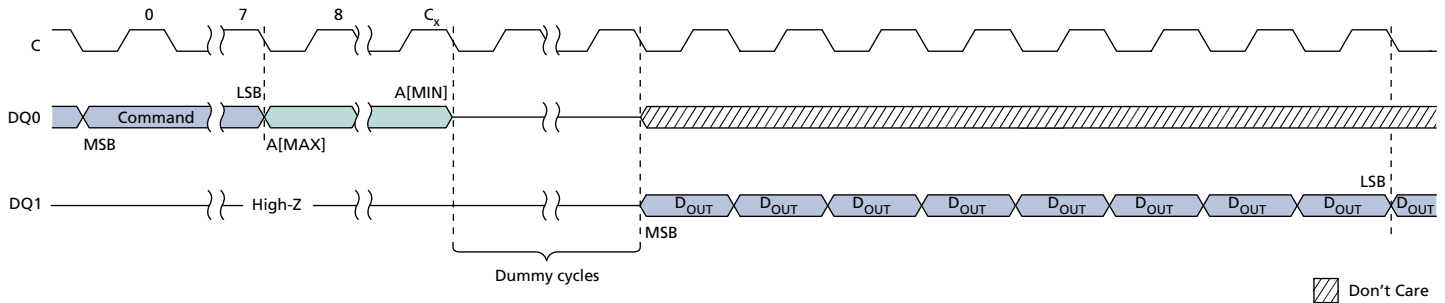
READ DATA BYTES at HIGHER SPEED

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 15: READ DATA BYTES at HIGHER SPEED Command Sequence



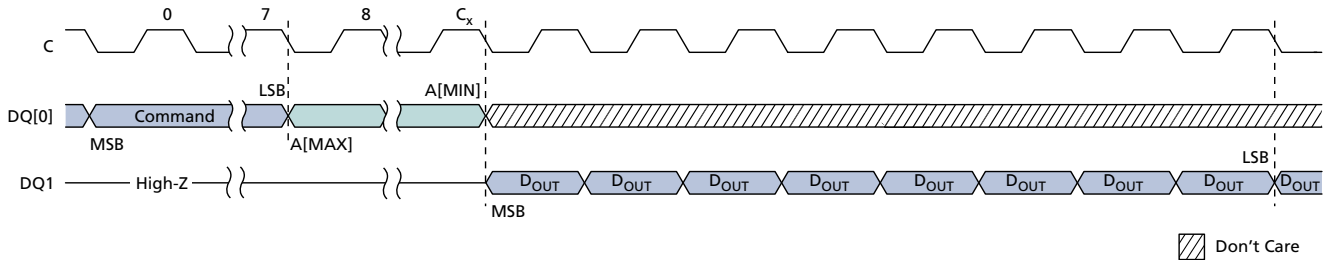
- Notes:
1. Address bits A23-A21 are *don't care* .
 2. Address bits A23-A19 are *don't care* .
 3. Address bits A23-A20 are *don't care* .

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 (or subsector). 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 16: READ LOCK REGISTER Command Sequence



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 the table below.

Table 11: Lock Register Out

| Bit | Bit name | Value | Function |
|-------|-------------------|-------|--|
| b7-b2 | | | Reserved |
| b7-b4 | | | Reserved |
| b7-b4 | | | 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. |

Table 12: Not for new design: Lock Registers for the M25PE80 in T7Y Process

| Bit | Bit name | Value | Function |
|-------|----------------------|-------|---|
| b7-b4 | Reserved | | |
| b3 | Subsector lock down | 1 | This functionality must not be used for new designs, as the M25PE80 delivered from Feb 2007 will not offer this functionality. The write lock and lock down bits cannot be changed. Once a 1 is written to the lock down bit it cannot be cleared to 0 except by a reset or power-up. |
| | | 0 | This functionality must not be used for new designs, as the M25PE80 delivered from Feb 2007 will not offer this functionality. The write lock and lock down bits can be changed by writing new values to them (default value). |
| b2 | Subsector write lock | 1 | This functionality must not be used for new designs, as the M25PE80 delivered from Feb 2007 will not offer this functionality. WRITE, PROGRAM, and ERASE operations in this subsector will not be executed. The memory contents will not be changed. |
| | | 0 | This functionality must not be used for new designs, as the M25PE80 delivered from Feb 2007 will not offer this functionality. WRITE, PROGRAM, and ERASE operations in this subsector are executed and will modify the subsector contents (default value). |
| 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 reset or power-up. |
| | | 0 | The write lock and lock-down bits can be changed by writing new values to them (default value) . |
| 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 (default value). |

Note: 1. Subsector write lock is valid for sector 0 and 15 only (the value 0 is returned for other sectors).

PAGE WRITE

The PAGE WRITE command allows bytes in the memory to be programmed. Before a PAGE WRITE 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 WRITE 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). The reset of the page remains unchanged if no power failure occurs during this write cycle.

The PAGE WRITE command performs a page erase cycle even if only one byte is updated.

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 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 effects on the other bytes of the same page.

For optimized timings, it is recommended to use the PAGE WRITE command to program all consecutive targeted bytes in a single sequence rather than to use several PAGE WRITE 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 WRITE command is not executed.

As soon as S# is driven HIGH, the self-timed PAGE WRITE cycle is initiated. While the PAGE WRITE 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 WRITE 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.

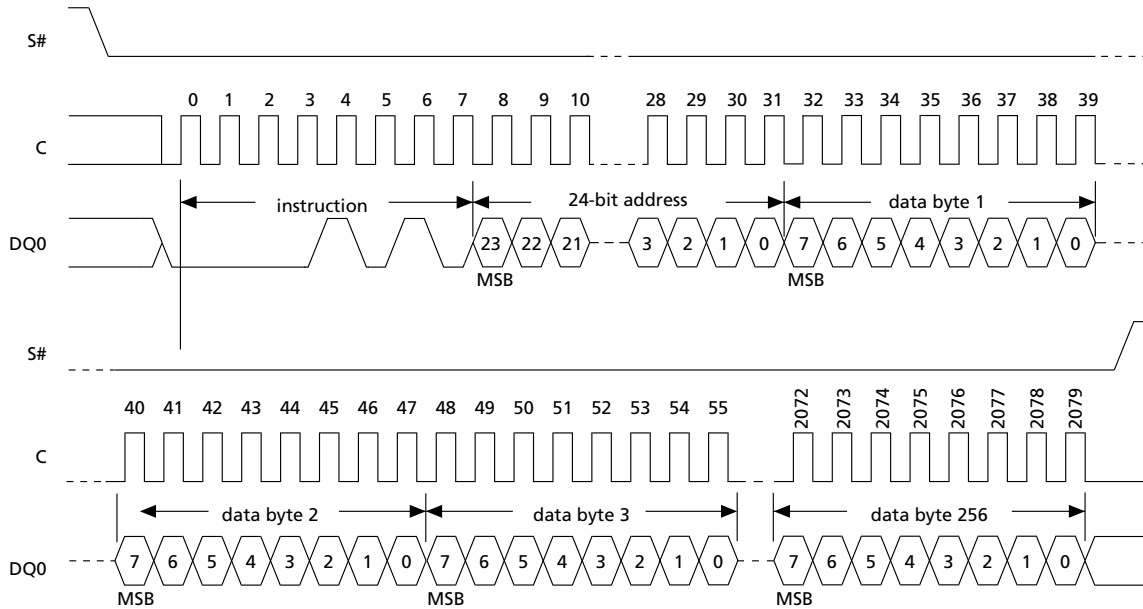
A PAGE WRITE command applied to a page that is hardware or software protected is not executed.

Any PAGE WRITE command while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effects on the cycle that is in progress.

If RESET is drive LOW while a PAGE WRITE cycle is in progress, the PAGE WRITE cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be rescheduled by selecting Chip Select (S#) LOW.

If RESET is drive LOW while a PAGE WRITE cycle is in progress, the PAGE WRITE cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be rescheduled by selecting Chip Select (S#) LOW.

Figure 17: PAGE WRITE Command Sequence



Note:

Address bits A23-A21 are don't care. $1 \leq n \leq 256$.

Note:

Address bits A23-A19 are don't care. $1 \leq n \leq 256$.

Note:

Address bits A23-A20 are don't care. $1 \leq n \leq 256$.

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 cycle'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 applied to a page that is hardware protected is not executed.

A PAGE PROGRAM command applied to a page that is hardware protected is not executed.

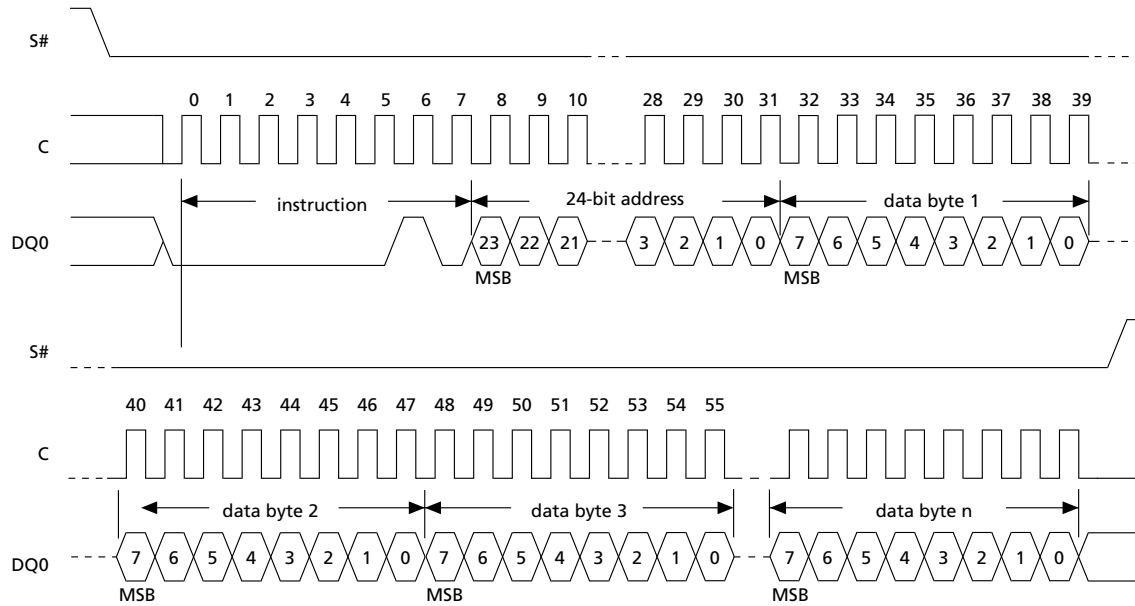
A PAGE PROGRAM command applied to a page that is hardware protected is not executed.

Any PAGE PROGRAM command while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effects on the cycle that is in progress.

If RESET# is driven LOW while a page program cycle is in progress, the page program cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be reselected by driving Chip Select (S#) LOW.

If RESET# is driven LOW while a page program cycle is in progress, the page program cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be reselected by driving Chip Select (S#) LOW.

Figure 18: PAGE PROGRAM Command Sequence



- Notes:
1. Address bits A23-A21 are don't care.
 2. Address bits A23-A19 are don't care. $1 \leq n \leq 256$.
 3. Address bits A23-A20 are don't care. $1 \leq n \leq 256$.

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 19: WRITE to LOCK REGISTER Instruction Sequence

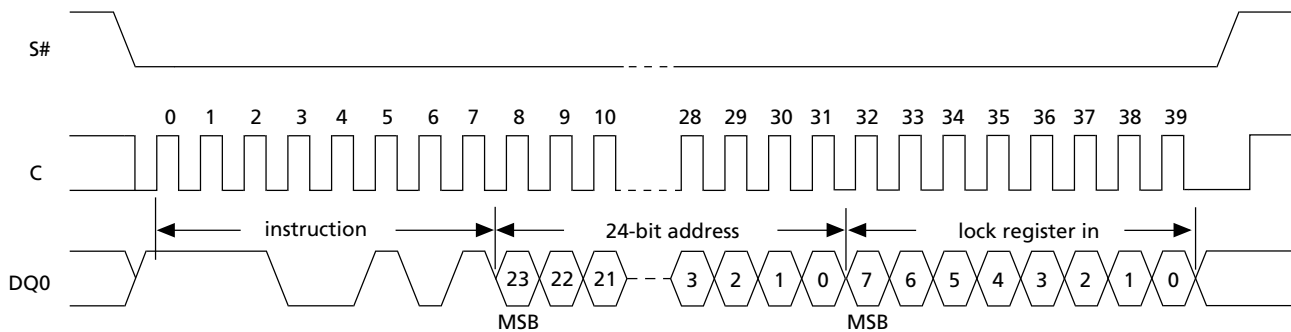


Table 13: Lock Register In

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

Table 14: Lock Register In

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

Table 15: Lock Register In

| Sector | Bit | Value |
|--|-------|-----------------------------|
| All sectors except for sector 0 and sector 15 in T7Y process | b7–b2 | 0 |
| | b1 | Sector lock-down bit value |
| | b0 | Sector write lock bit value |

Table 16: Not for new design: lock registers for the M25PE80 in T7Y process

| Sector | Bit | Value | |
|--|-------|--------------------------------|--|
| All sectors except for sector 0 and sector 15 in T7Y process | b7–b2 | 0 | |
| | b1 | Sector lock-down bit value | |
| | b0 | Sector write lock bit value | |
| Sector 0, sector 15 in T7Y process | b7 | 1 | Only b3 and b2 are taken into account to modify the subsector write lock and lock down bits ¹ |
| | | 0 | Only b1 and b0 are taken into account to modify the sector write lock and lock down bits ² |
| | b3 | Subsector lock down bit value | |
| | b2 | Subsector write lock bit value | |
| | b1 | Sector lock down bit value | |
| | b0 | Sector write lock bit value | |

- Notes:
1. b6-b4 and b1-b0 must be reset to 0.
 2. b6-b2 must be reset to 0.

For products processed in the T7Y process only:

Protection always prevails:

- When the lock down bit of sector 0 or sector 15 is set to 1.
 - If the lock down bit of sector 0 is 1, all the lock down bits of the subsectors in sector 0 are forced to 1.
 - If the lock down bit of sector 15 is 1, all the lock down bits of the subsectors in sector 15 are forced to 1.
- When the write lock bit of sector 0 or sector 15 is set to 1.
 - If the write lock bit of sector 0 is 1, the write lock bits of all the subsectors in sector 0 are forced to 1 (even if their lock down bits are set to 1).
 - If the write lock bit of sector 15 is 1, the write lock bits of all the subsectors in sector 15 are forced to 1 (even if their lock down bits are set to 1).
- When the write lock bit of sector 0 or sector 15 is reset to 0.
 - If the write lock bit of sector 0 is 0, all the subsectors in sector 0 whose lock down bits are 0 have their write lock bits forced to 0.

- If the write lock bit of sector 15 is 0, all the subsectors in sector 15 whose lock down bit is 0 have their write lock bits forced to 0.
- When the write lock bit of any sector or subsector is set to 1, any instruction that may modify the contents of this sector or subsector will be rejected (including SECTOR ERASE and BULK ERASE).

Note that when the WRLR instruction acts both on WRITE LOCK (WL) and LOCK DOWN (LD) bits, it programs the WL bit first, and then the LD bit. As an example, if a subsector lock register settings are xxxx0101b and a WRLR instruction is issued with a lock register in data set to 00000010b:

1. The sector WL bit is first set to 0 (and all subsectors that are not locked down will have their WL bit reset to 0).
2. The sector LD bit and all subsectors LD bits are set to 1. In this case, the final value of the above subsector lock register is xxxx1010b.

PAGE ERASE

The PAGE ERASE command sets to 1 (FFh) all bits inside the chosen page. Before the PAGE 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 PAGE 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 PAGE 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 PAGE ERASE command is not executed. As soon as S# is driven HIGH, the self-timed PAGE ERASE cycle is initiated; the cycle's duration is t_{PE} . While the PAGE 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 PAGE ERASE cycle, and is 0 when the cycle is completed. At some unspecified time before the cycle is completed, the WEL bit is reset.

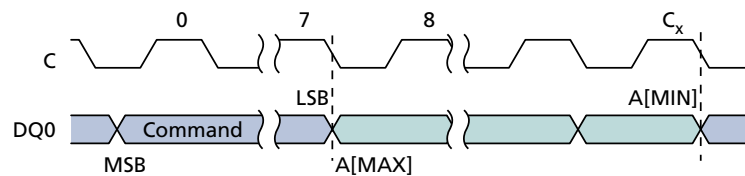
A PAGE ERASE command applied to a page that is hardware or software protected is not executed.

A PAGE ERASE command while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effects on the cycle that is in progress.

If RESET# is driven LOW while a PAGE ERASE cycle is in progress, the PAGE ERASE cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be reselected by driving Chip Select (S#) LOW.

If RESET# is driven LOW while a PAGE ERASE cycle is in progress, the PAGE ERASE cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be reselected by driving Chip Select (S#) LOW.

Figure 20: PAGE ERASE Command Sequence



- Notes:
1. Address bits A23-A21 are *don't care*.
 2. Address bits A23-A19 are *don't care*.
 3. Address bits A23-A20 are *don't care*.

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.

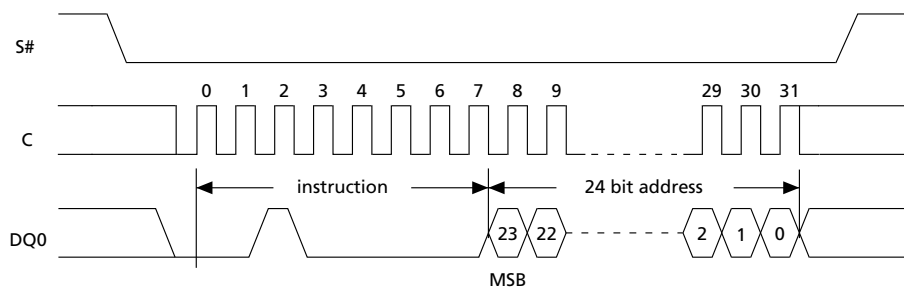
S# must be driven HIGH after the eighth bit of the last address byte has been latched in. Otherwise the SUBSECTOR ERASE command is not executed. As soon as S# is driven HIGH, the self-timed SUBSECTOR ERASE cycle is initiated; the cycle's duration is t_{SSE} . While the SUBSECTOR ERASE cycle is in progress, 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.

If RESET# is driven LOW while a SUBSECTOR ERASE cycle is in progress, the SUBSECTOR ERASE cycle is interrupted and data may not be erased correctly. On RESET# going LOW, the device enters the RESET mode and a time of t_{RHSL} is then required before the device can be reselected by driving S# LOW.

Figure 21: SUBSECTOR ERASE Command Sequence



- Notes:
1. Address bits A23-A21 are *don't care*.
 2. Address bits A23-A19 are *don't care*.
 3. Address bits A23-A20 are *don't care*.

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.

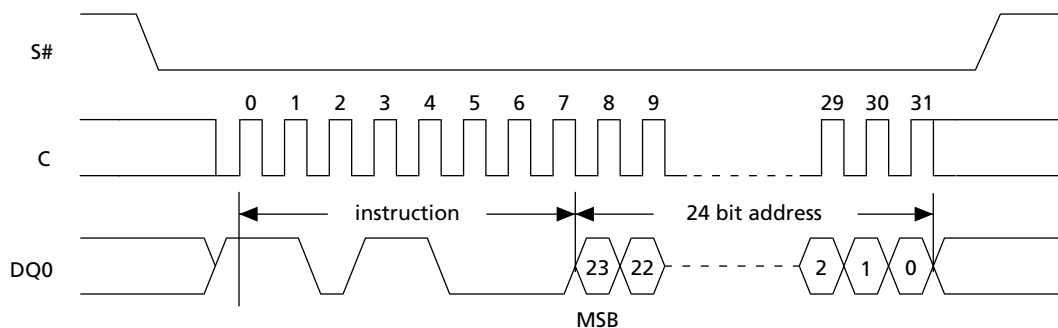
A SECTOR ERASE command applied to a sector that contains a page that is hardware protected is not executed.

Any SECTOR ERASE command while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effects on the cycle that is in progress.

If RESET# is driven LOW while a SECTOR ERASE cycle is in progress, the SECTOR ERASE cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be reselected by driving Chip Select (S#) LOW.

If RESET# is driven LOW while a SECTOR ERASE cycle is in progress, the SECTOR ERASE cycle is interrupted and the programmed data may be corrupted. On RESET going LOW, the device enters the reset mode and a time of t_{RHSL} is then required before the device can be reselected by driving Chip Select (S#) LOW.

Figure 22: SECTOR ERASE Command Sequence



- Notes:
1. Address bits A23-A21 are *don't care* .
 2. Address bits A23-A19 are *don't care* .
 3. Address bits A23-A20 are *don't care* .

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.

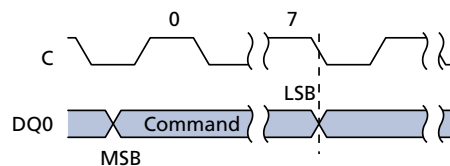
The BULK ERASE command is entered by driving chip select (S#) LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

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.

Any BULK ERASE command while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effects on the cycle that is in progress. A BULK ERASE command is ignored if at least one sector or subsector is write-protected (hardware or software protection).

If RESET# is driven LOW while a BULK ERASE is in progress, the BULK ERASE cycle is interrupted and data may not be erased correctly. On RESET# going LOW, the device enters the RESET mode and a time of t_{RHSL} is then required before the device can be reselected by driving S# LOW.

Figure 23: 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} .

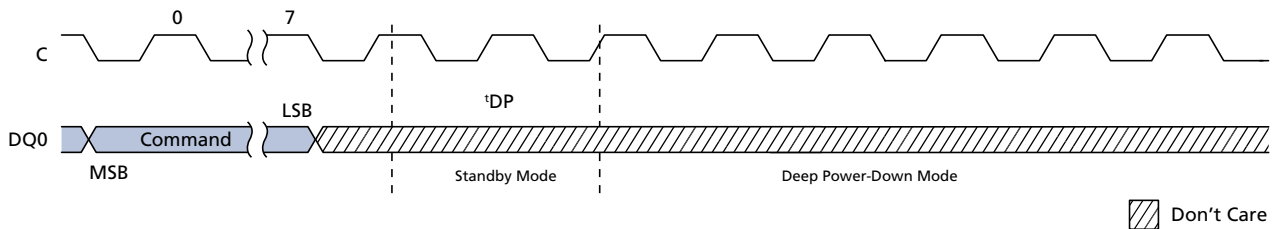
To take the device out of DEEP POWER-DOWN mode, the RELEASE from DEEP POWER-DOWN command must be issued. Other commands must not be issued while the device is in DEEP POWER-DOWN mode. The DEEP POWER-DOWN mode stops automatically at power-down. The device always powers up in STANDBY POWER mode.

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 24: 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.

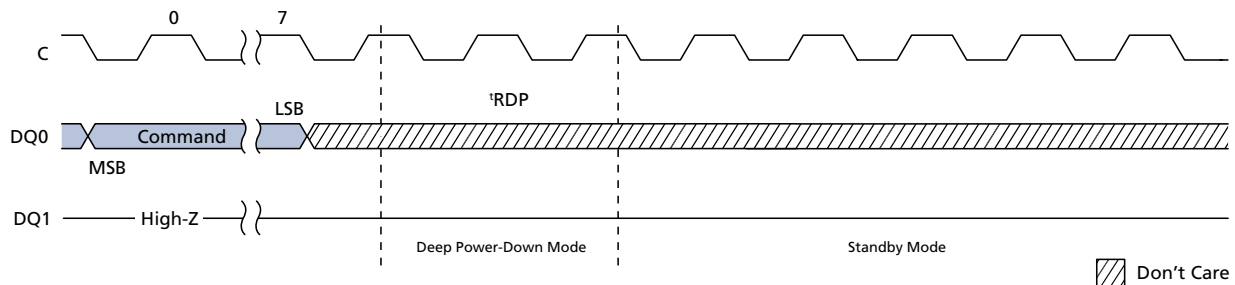
The RELEASE from DEEP POWER-DOWN command is entered by driving chip select (S#) LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

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 STANDBY 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 25: RELEASE from DEEP POWER-DOWN Command Sequence



Power-Up and Power-Down

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:

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

A safe configuration is provided under the SPI modes heading.

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 WRITE
- PAGE PROGRAM
- PAGE ERASE
- SECTOR ERASE

However, the correct operation of the device is not guaranteed if, by this time, V_{CC} is still below $V_{CC}(\text{min})$. No WRITE, 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 $V_{CC}(\text{min})$ level

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

As an extra precaution, the RESET# signal could be driven LOW for the entire duration of the power-up and power-down phases.

Figure 26: Power-Up Timing

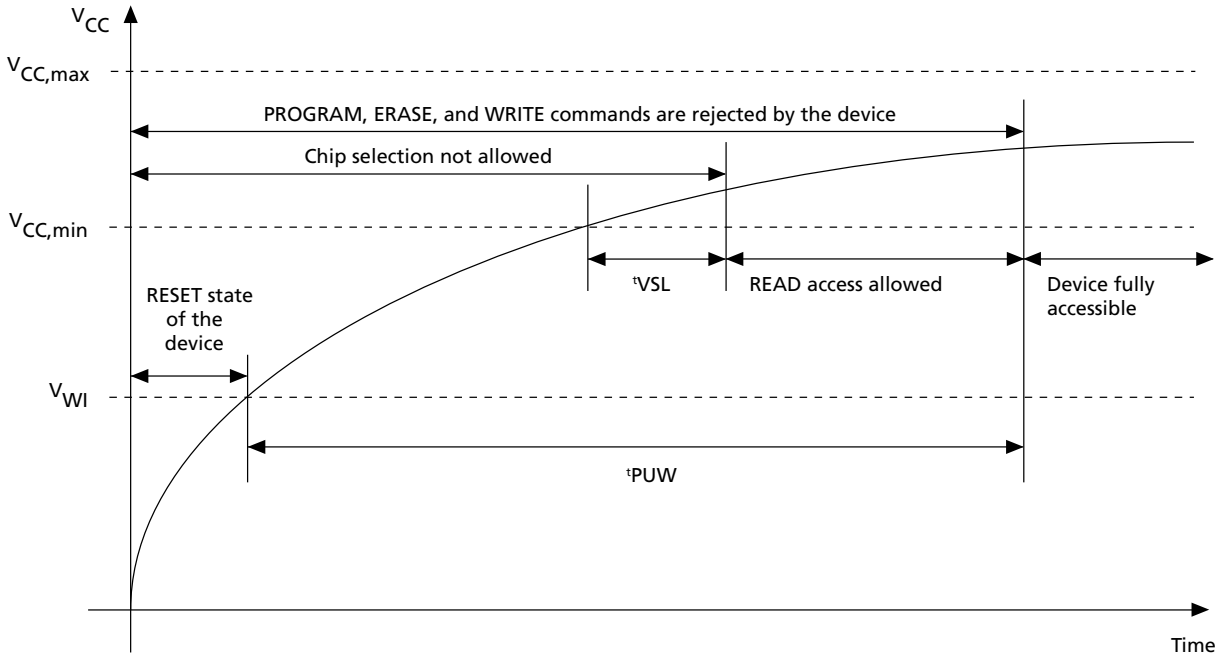


Table 17: Power-up Timing and V_{WI} Threshold

Note: These parameters are characterized only, over the temperature range $-40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$.

| Symbol | Parameter | Min | Max | Unit |
|-----------|--|-----|-----|---------------|
| t_{VSL} | $V_{CC}(min)$ to S# LOW | 30 | – | μs |
| t_{PUW} | Time delay before the first WRITE, PROGRAM, or ERASE instruction | 1 | 10 | ms |
| V_{WI} | Write inhibit voltage | 1.5 | 2.5 | V |

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.
- The Lock Registers are reset (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 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: Designers need to be aware that if power-down occurs while a WRITE, PROGRAM, or ERASE cycle is in progress, some data corruption may result.

RESET

Driving RESET# LOW while an internal operation is in progress will affect this operation (WRITE, PROGRAM, or ERASE cycle) and data may be lost. All lock bits are reset to 0 after a RESET# LOW pulse.

Table 18: Device Status After a RESET# LOW Pulse

| Conditions: RESET Pulse Occurred | Lock Bits Status | Internal Logic Status | Addressed Data |
|---|------------------|---|----------------------------------|
| While decoding the following commands: | | | |
| WRITE ENABLE, WRITE DISABLE, READ ID, READ STATUS REGISTER, READ, READ LOCK REGISTER, FAST READ, WRITE LOCK REGISTER, PAGE WRITE, PAGE PROGRAM, PAGE ERASE, SECTOR ERASE, BULK ERASE, SUBSECTOR ERASE, DEEP POWER-DOWN, RELEASE DEEP POWER-DOWN | Reset to 0 | Same as power-on/reset | Not significant |
| Under completion of ERASE or PROGRAM cycle for the following commands: | | | |
| PAGE WRITE, PAGE PROGRAM, PAGE ERASE, SUBSECTOR ERASE, SECTOR ERASE, or BULK ERASE | Reset to 0 | Equivalent to power-on/reset | Addressed data could be modified |
| Under completion of a WRITE STATUS REGISTER operation | Reset to 0 | Equivalent to power-on/reset (after t_W) | Write is correctly completed |
| Device deselected (S# HIGH) and in STANDBY mode | Reset to 0 | Same as power-on/reset | Not significant |

Note: 1. S# remains LOW while RESET# is LOW.

Initial Delivery State

The device is delivered with the memory array erased: all bits are set to 1 (each byte contains FFh). All usable status register bits are 0.

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 _{ESD} | Electrostatic discharge voltage (Human Body model) | -2000 | 2000 | V | 2 |

- 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. 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 |
| T _A | Ambient operating temperature | -40 | 85 | °C |

DC Parameters

Table 21: DC Characteristics 75 MHz Operation

| Symbol | Parameter | Test Conditions (in addition to those listed in Operating Conditions table) | Min | Max | Units |
|-----------|---|--|--------------|--------------|---------|
| I_{LI} | Input leakage current | – | – | ± 2 | μA |
| I_{LO} | Output leakage current | – | – | ± 2 | μA |
| I_{CC1} | Standby current (Standby and Reset modes) | $S\# = V_{CC}, V_{IN} = V_{SS} \text{ or } V_{CC}$ | – | 50 | μA |
| I_{CC2} | Deep power-down current | $S\# = V_{CC}, V_{IN} = V_{SS} \text{ or } V_{CC}$ | – | 10 | μA |
| I_{CC3} | Operating current (FAST_READ) | $C = 0.1V_{CC} / 0.9V_{CC}$ at 33 MHz, DQ1 = open | – | 4 | mA |
| | | $C = 0.1V_{CC} / 0.9V_{CC}$ at 75 MHz, DQ1 = open | – | 12 | mA |
| I_{CC4} | Operating current (PAGE WRITE) | $S\# = V_{CC}$ | – | 15 | mA |
| I_{CC5} | Operating current (SECTOR ERASE) | $S\# = V_{CC}$ | – | 15 | mA |
| 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 |
| V_{OH} | Output High Voltage | $I_{OH} = -100 \mu A$ | $V_{CC}-0.2$ | | V |

AC Characteristics

In the following AC measurement conditions, output HIGH-Z is defined as the point where data out is no longer driven.

Table 22: 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 and output timing reference voltages | $0.3V_{CC}$ | $0.7V_{CC}$ | V |

Figure 27: AC Measurement I/O Waveform

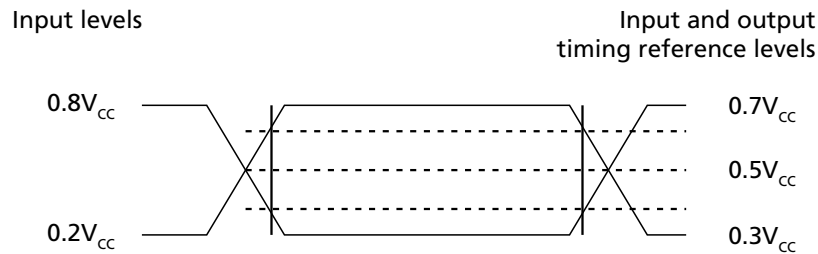


Table 23: Capacitance

| Symbol | Parameter | Test condition | Min | Max | Unit | Notes |
|-----------|--------------------------------|-----------------|-----|-----|------|-------|
| C_{OUT} | 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 $T_A=25^\circ C$ and a frequency of 25MHz.

Table 24: AC Specifications (50 MHz operation)

Test conditions are specified in the Operating Conditions and AC Measurement Conditions tables.

| Symbol | Alt. | Parameter | Min | Typ | Max | Unit | Notes |
|------------|-----------|---|------|--------------------------------|-----|---------|-------|
| f_C | f_C | Clock frequency for the following commands: FAST_READ, RDLR, PW, PP, WRLR, PE, SE, SSE, DP, RDP, WREN, WRDI, RDSR, WRSR | D.C. | – | 50 | MHz | |
| f_R | – | Clock frequency for READ command | D.C. | – | 33 | MHz | |
| t_{CH} | t_{CLH} | Clock HIGH time | 9 | – | – | ns | 1 |
| t_{CL} | t_{CLL} | Clock LOW time | 9 | – | – | ns | 1 |
| | | Clock Slew Rate (peak to peak) | 0.1 | – | – | V/ns | |
| 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 | 100 | – | – | ns | |
| t_{SHQZ} | t_{DIS} | Output disable time | – | – | 8 | ns | 2 |
| t_{CLQV} | t_V | Clock LOW to output valid under 30pF/10pF | – | – | 8/6 | ns | |
| t_{CLQX} | t_{HO} | Output hold time | 0 | – | – | ns | |
| t_{WHSL} | – | WRITE PROTECT setup time | 50 | – | – | ns | 3 |
| t_{SHWL} | – | WRITE PROTECT hold time | 100 | – | – | ns | 3 |
| t_{DP} | – | S# to DEEP POWER-DOWN mode | – | – | 3 | μ s | 2 |
| t_{RDP} | – | S# HIGH to STANDBY mode | – | – | 30 | μ s | 2 |
| t_W | – | WRITE STATUS REGISTER cycle time | – | 3 | 15 | ms | |
| t_{PW} | – | PAGE WRITE cycle time (256 bytes) | – | 11 | 23 | ms | |
| t_{PP} | – | PAGE PROGRAM cycle time (256 bytes) | – | 0.8 | 3 | ms | 4 |
| t_{PP} | – | PAGE PROGRAM cycle time (n bytes) | – | $\text{int}(n/8) \times 0.025$ | 3 | ms | 4, 5 |
| t_{PE} | – | PAGE ERASE cycle time | – | 10 | 20 | ms | |
| t_{SE} | – | SECTOR ERASE cycle time | – | 1.5 | 5 | s | |
| t_{SSE} | – | SUBSECTOR ERASE cycle time | – | 80 | 150 | ms | |
| t_{BE} | – | BULK ERASE cycle time | – | 8 | 10 | s | |

- Notes:
1. The t_{CH} and t_{CL} signal values must be greater than or equal to $1/f_C$.
 2. Signal values are guaranteed by characterization, not 100% tested in production.
 3. Only applicable as a constraint for a WRSR instruction when SRWD is set to 1.
 4. When using PP and PW commands to update consecutive bytes, optimized timings are obtained with one sequence including all the bytes versus several sequences of only a few bytes ($1 \leq n \leq 256$).
 5. $\text{int}(A)$ corresponds to the upper integer part of A. For example, $\text{int}(12/8) = 2$, $\text{int}(32/8) = 4$, $\text{int}(15.3) = 16$.

Table 25: AC Specifications (75MHz operation)

Test conditions are specified in the Operating Conditions and AC Measurement Conditions tables.

| Symbol | Alt. | Parameter | Min | Typ | Max | Unit | Notes |
|------------|-----------|---|------|--------------------------------|-------|---------|-------|
| f_C | f_C | Clock frequency for the following commands: FAST_READ, RDLR, PW, PP, WRLR, PE, SE, SSE, DP, RDP, WREN, WRDI, RDSR, WRSR | D.C. | – | 75 | MHz | |
| f_R | – | Clock frequency for READ command | D.C. | – | 33 | MHz | |
| t_{CH} | t_{CLH} | Clock HIGH time | 6 | – | – | ns | 1 |
| t_{CL} | t_{CLL} | Clock LOW time | 6 | – | – | ns | 1 |
| | | Clock Slew Rate (peak to peak) | 0.1 | – | – | V/ns | |
| 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 | 100 | – | – | ns | |
| t_{SHQZ} | t_{DIS} | Output disable time | – | – | 8 | ns | 2 |
| t_{CLQV} | t_V | Clock LOW to output valid | – | – | 8 / 6 | ns | |
| t_{CLQX} | t_{HO} | Output hold time | 0 | – | – | ns | |
| t_{WHSL} | – | WRITE PROTECT setup time | 20 | – | – | ns | 3 |
| t_{SHWL} | – | WRITE PROTECT hold time | 100 | – | – | ns | 3 |
| t_{DP} | – | S# to DEEP POWER-DOWN mode | – | – | 3 | μ s | 2 |
| t_{RDP} | – | S# HIGH to STANDBY mode | – | – | 30 | μ s | 2 |
| t_W | – | WRITE STATUS REGISTER cycle time | – | 3 | 15 | ms | |
| t_{PW} | – | PAGE WRITE cycle time (256 bytes) | – | 11 | 23 | ms | 4 |
| t_{PP} | – | PAGE PROGRAM cycle time (256 bytes) | – | 0.8 | 3 | ms | 4 |
| t_{PP} | – | PAGE PROGRAM cycle time (n bytes) | – | $\text{int}(n/8) \times 0.025$ | 3 | ms | 4, 5 |
| t_{PE} | – | PAGE ERASE cycle time | – | 10 | 20 | ms | |
| t_{SSE} | – | SUBSECTOR ERASE cycle time | – | 80 | 150 | ms | |
| t_{SE} | – | SECTOR ERASE cycle time | – | 1.5 | 5 | s | |
| t_{BE} | – | BULK ERASE cycle time | – | 8 | 10 | s | |

- Notes:
1. The t_{CH} and t_{CL} signal values must be greater than or equal to $1/f_C$.
 2. Signal values are guaranteed by characterization, not 100% tested in production.
 3. Signal values are only applicable as a constraint for a WRITE STATUS REGISTER command when SRWD bit is set at 1.
 4. When using PP and PW commands to update consecutive bytes, optimized timings are obtained with one sequence including all the bytes versus several sequences of only a few bytes ($1 \leq n < 256$).
 5. $\text{int}(A)$ corresponds to the upper integer part of A. For example, $\text{int}(12/8) = 2$, $\text{int}(32/8) = 4$, $\text{int}(15.3) = 15$.

Figure 28: Serial Input Timing

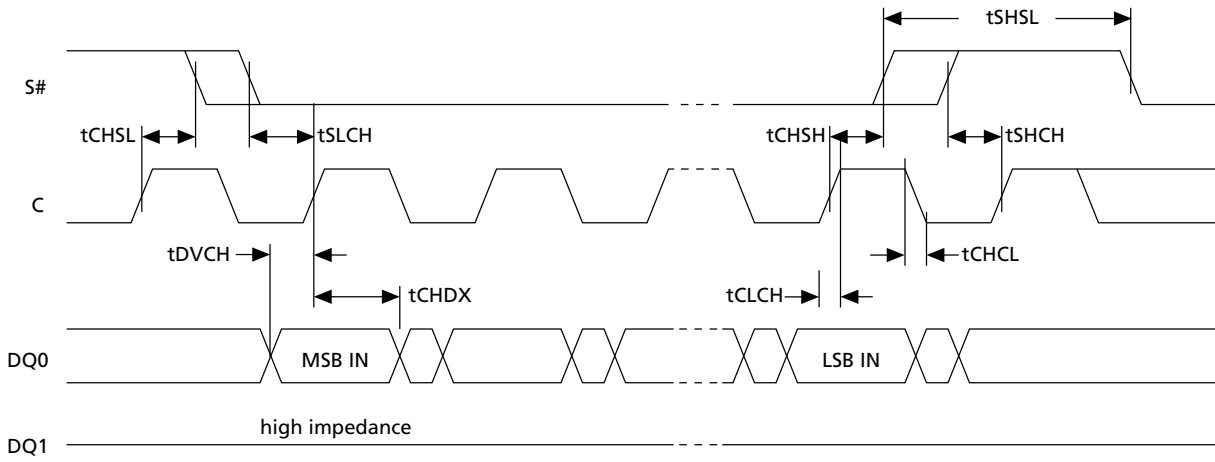


Figure 29: Write Protect Setup and Hold Timing

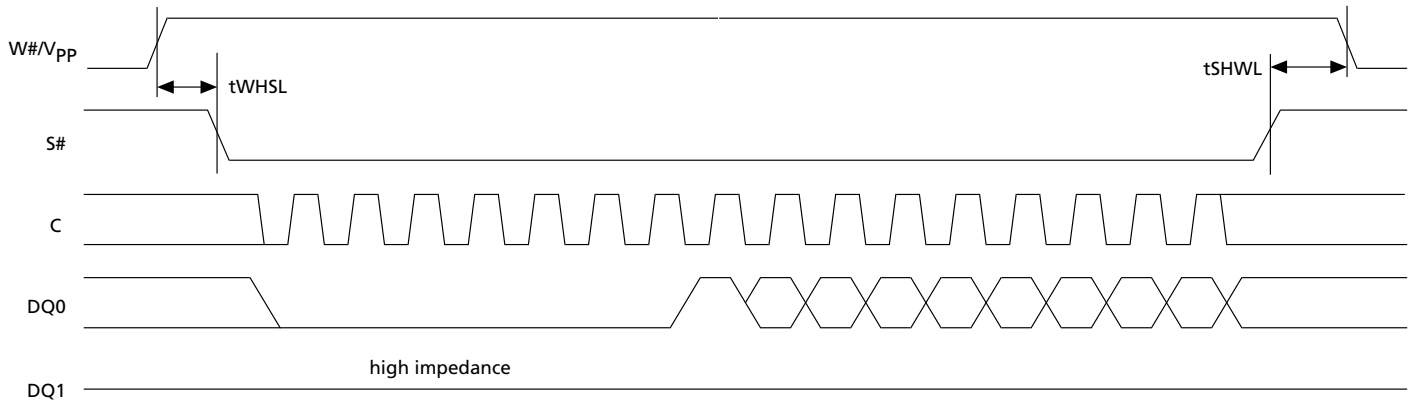


Figure 30: Output Timing

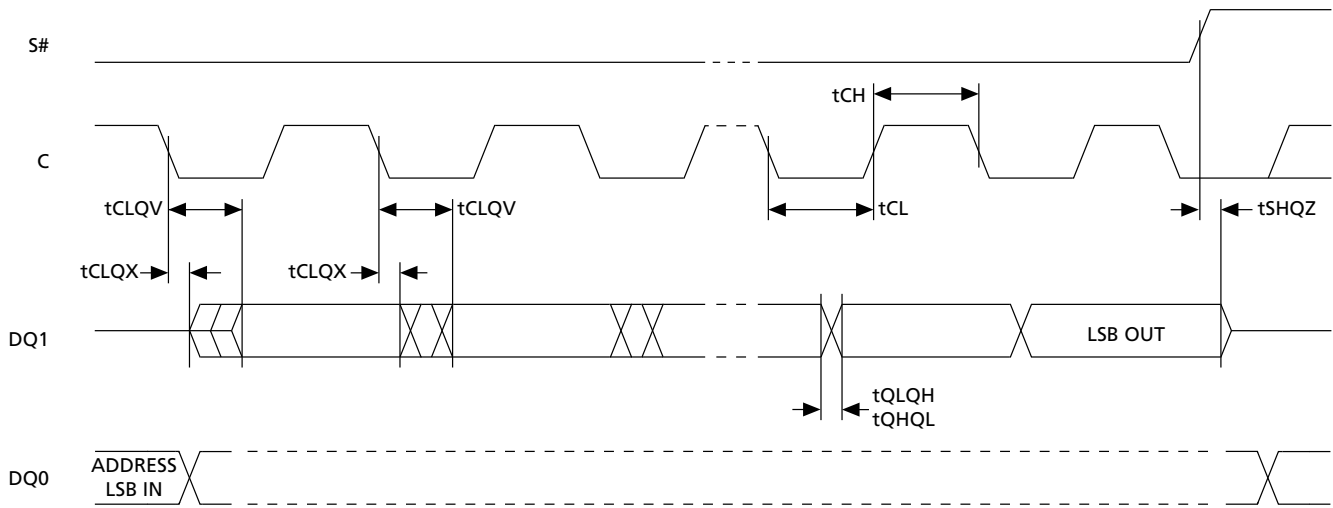


Table 26: Reset Conditions

| Symbol | Alt | Parameter | Conditions | Min | Typ | Max | Unit |
|-------------------|------------------|--------------------------------|--|-----|-----|-----|------|
| t _{RLRH} | t _{RST} | Reset Pulse width | | 10 | – | – | μs |
| t _{SHRH} | | Chip Select HIGH to RESET HIGH | Chip should have been deselected before RESET is de-asserted | 10 | – | – | μs |

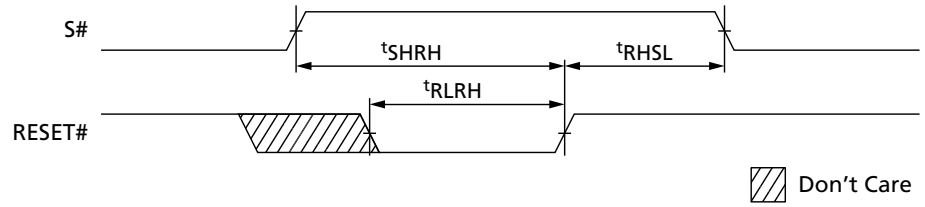
Note: 1. Test conditions specified in Operating Conditions and AC Measurement Conditions tables. t_{RLRH} value guaranteed by characterization; not 100% tested in production.

Table 27: Timings After a RESET# LOW Pulse

| Symbol | Alt | Parameter | Conditions: Reset Pulse Occurred | Max | Unit | Notes |
|-------------------|------------------|---------------------|--|----------------|------|-------|
| t _{RHSL} | t _{REC} | Reset recovery time | While decoding an instruction: WREN, WRDI, RDID, RDSR, READ, RDLR, FAST_READ, WRLR, PW, PP, PE, SE, BE, SSE, DP, RDP | 30 | μs | |
| | | | Under completion of an ERASE or PROGRAM cycle of a PW, PP, PE, SE, BE operation | 300 | μs | |
| | | | Under completion of an ERASE cycle of an SSE operation | 3 | ms | |
| | | | Under completion of a WRSR operation | t _W | ms | 2 |
| | | | Device deselected (S# HIGH) and in Standby mode | 0 | μs | |

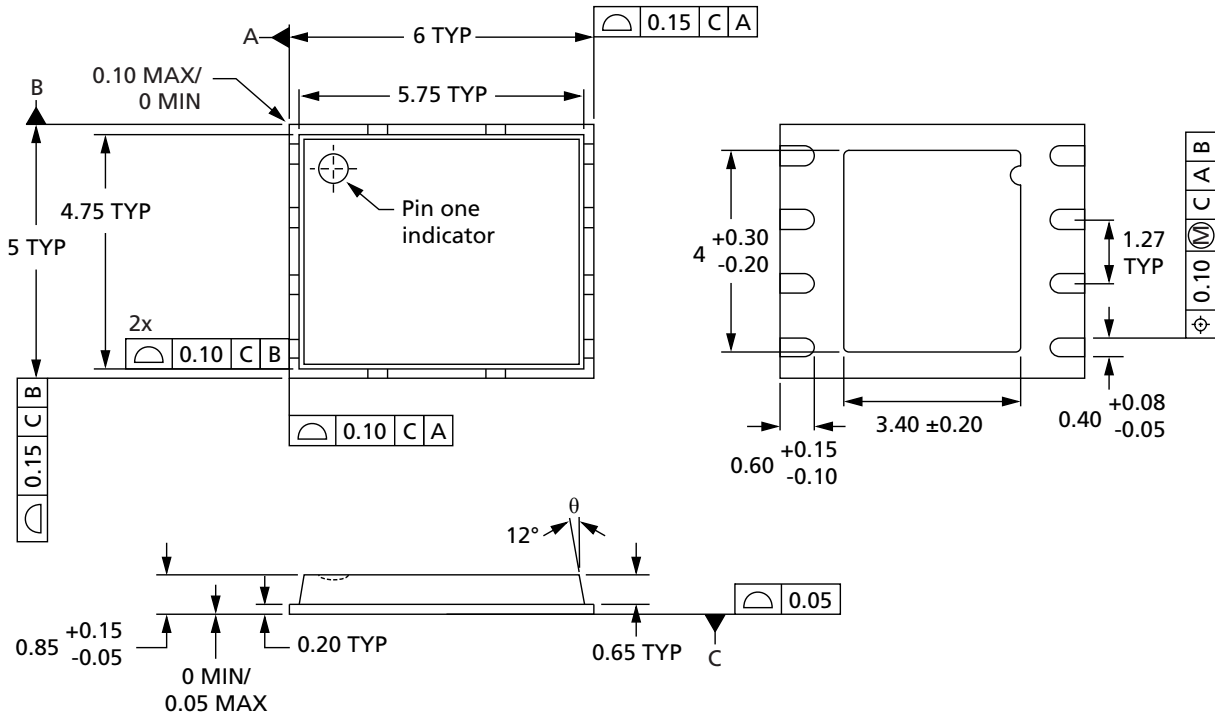
Notes: 1. All values are guaranteed by characterization; not 100% tested in production. See Device Status after a RESET# LOW Pulse table for a description of the device status after a RESET# LOW pulse. While decoding an instruction, S# remains LOW while RESET# is LOW.
2. See AC Characteristics 75 MHz operation tables.

Figure 31: Reset AC Waveforms



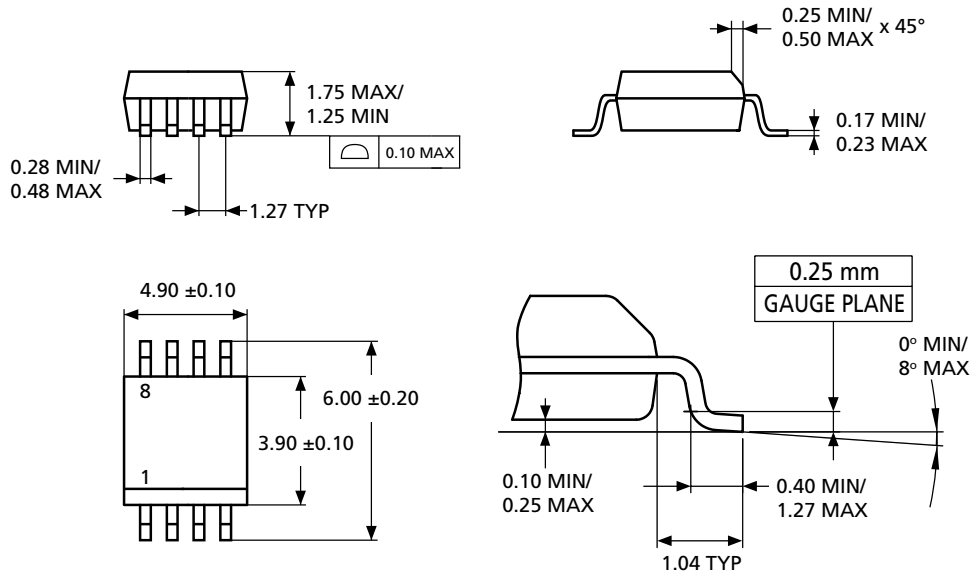
Package Information

Figure 32: VFQFPN8 (MLP8) 6mm x 5mm



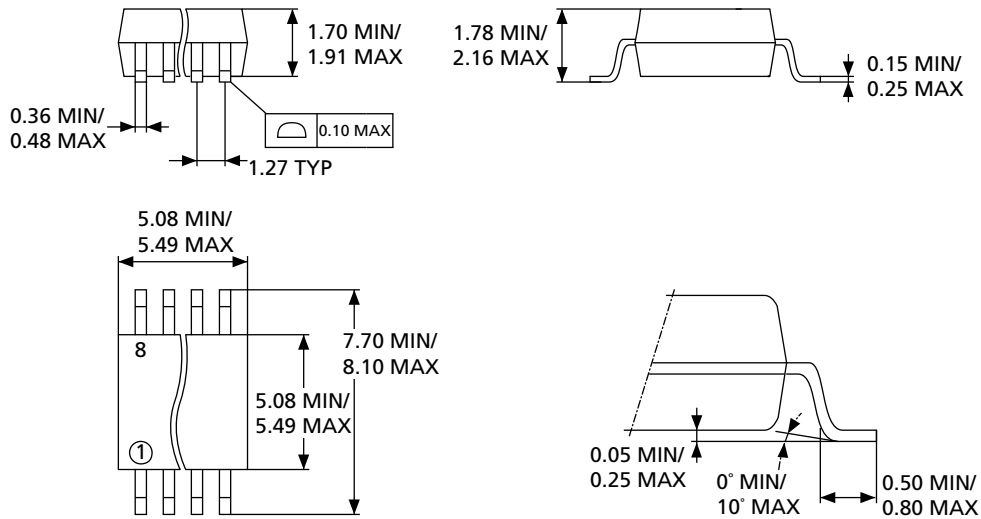
Note: 1. Drawing is not to scale.

Figure 33: S08N – 8 lead plastic small outline, 150 mils body width



Note: 1. Drawing is not to scale.

Figure 34: S08W – 8 lead plastic small outline, 208 mils body width



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: Standard Part Number Information Scheme

| Part Number Category | Category Details |
|----------------------|---|
| Device type | M25PE = Page-erasable serial Flash memory |
| Density | 40 = 4Mb (512K x 8-bit) |
| Security features | – = No extra security |
| Operating voltage | V = V _{CC} = 2.7V to 3.6V |
| Package | MP = VFQFPN8 6 x 5mm (MLP8) |
| | MW = SO8W (208 mils width) |
| | MN = SO8N (150 mils width) |
| Device Grade | 6 = Industrial temperature range: –40°C to 85°C. Device tested with standard test flow. |
| Packing Option | – = Standard packing |
| | T = Tape and reel packing |
| Plating technology | P or G = RoHS compliant |

Table 29: Automotive Part Number Information Scheme

| Part Number Category | Category Details |
|----------------------|---|
| Device type | M25PE = Page-erasable serial Flash memory |
| Density | 40 = 4Mb (512K x 8-bit) |
| Security features | – = No extra security |
| Operating voltage | V = V _{CC} = 2.7V to 3.6V |
| Package | MN = SO8N (150 mils width) |
| Device Grade | 6 = Industrial temperature range: –40°C to 85°C. Device tested with standard test flow. |
| Packing Option | – = Standard packing |
| | T = Tape and reel packing |
| Plating technology | P = RoHS compliant |
| Lithography | B = 110nm, fab 2, diffusion plant (available only with automotive) |
| | blank = Standard (industrial) |
| Automotive grade | A = Automotive –40°C to 85°C (together with device grade = 6 to order automotive product) . Device tested with high reliability certified flow. |
| | blank = Standard –40°C to 85°C device. |



M25PE40 Serial Flash Memory Device Ordering Information

Revision History

Rev. D – 1/2018

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

Rev. C – 3/2013

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

Rev. B – 11/2012

- Removed T9HX process information
- Added automotive grade part information

Rev. A – 09/2012

- Applied Micron branding

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
Micron and the Micron logo are trademarks of Micron Technology, Inc.
All other trademarks are the property of their respective owners.

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 M25PE40-VMW6TG TR 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