Micron Serial NOR Flash Memory

3V, 16Mb, Page Erasable with Byte Alterability

M45PE16

Features

- SPI bus-compatible serial interface
- 75 MHz clock frequency (MAX)
- 2.7–3.6V single supply voltage
- 16Mb of page-erasable Flash memory
- Page size: 256 bytes
  - Page write: 11ms (TYP)
  - Page program: 0.8ms (TYP)
  - Page erase: 10ms (TYP)
- Sector erase: 512Kb
- Hardware write protection of the bottom memory area 64KB
- Electronic signature
  - JEDEC-standard, 2-byte signature (4015h)
- Deep power-down mode: 1µA (TYP)
- WRITE cycles per sector: >100,000
- Years of data retention: >20
- Packages (RoHS-compliant)
  - SO8W (MW) 208 mil
  - VFQFPN8 (MP) 6;;mm x 5mm

Contents

List of Figures

List of Tables

Functional Description

The M45PE16 is a 16Mb (2Mb x 8) serial 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 32 sectors, each containing 256 pages. Each page is 256 bytes wide. The entire memory can be viewed as consisting of 8192 pages, or 2,097,152 bytes.

The memory can be erased one page at a time using the PAGE ERASE command or one sector at a time using the SECTOR ERASE command.

To meet environmental requirements, Micron offers the M45PE16 in RoHS-compliant packages, which are also lead-free.

Delivery of parts operating with a maximum clock rate of 75 MHz begins week 8 of 2008.

Figure 1: Logic Diagram

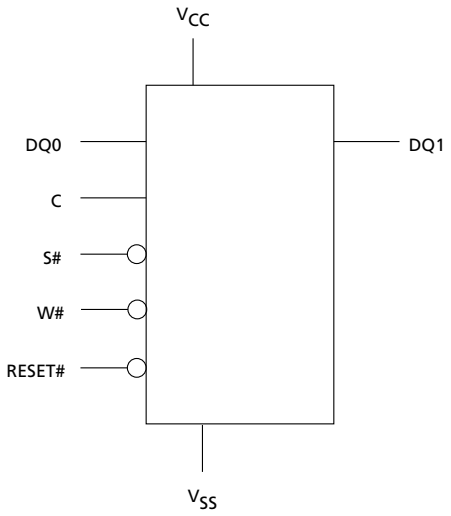
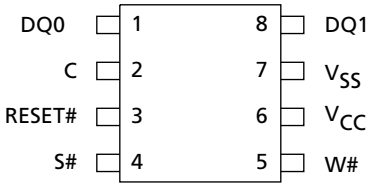


Table 1: Signal Names

Signal Name	Function	Direction
C	Serial clock	Input
DQ0	Serial data input	Input
DQ1	Serial data output	Output
S#	Chip select	Input
W#	Write protect	Input
RESET#	Reset	Input
V_{CC}	Supply voltage	–
V_{SS}	Ground	–

Figure 2: Pin Connections: MLP and SO8



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 Information section provides details about package dimensions and how to identify pin 1.

Signal Descriptions

Table 2: Signal Descriptions

Signal	Type	Description
DQ0	Input	**Serial data:** Transfers data serially into the device. DQ0 receives commands, addresses, and data to be programmed. Values are latched on the rising edge of serial clock (C).
C	Input	**Clock:** Provides timing for the serial interface. Commands, addresses, or data present at serial data input (DQ0) is latched on the rising edge of serial clock (C). Data on DQ1 changes after the falling edge of C.
S#	Input	**Chip select:** When S# is HIGH, the device is deselected and DQ1 is High-Z. Unless an internal READ, PROGRAM, ERASE, or WRITE cycle is in progress, the device will be in the standby power mode (not 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:** Provides a hardware reset for the memory. When RESET# is driven HIGH, the device is in the normal operating mode. When RESET# is driven LOW, the device enters the reset mode. In reset mode, the output is High-Z. 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:** Places the device in hardware protected mode when connected to V_{SS}, causing the first 256 pages of memory to become read-only, protected from WRITE, PROGRAM, and ERASE operations. When W# is connected to V_{CC}, the first 256 pages of memory behave like the other pages.
DQ1	Output	**Serial data:** Transfers data serially out of the device. Data is shifted out on the falling edge of the serial clock (C).
V_{CC}	Supply	**Supply voltage:** 2.7–3.6V
V_{SS}	Supply	**Ground:** Reference for V_{CC}.

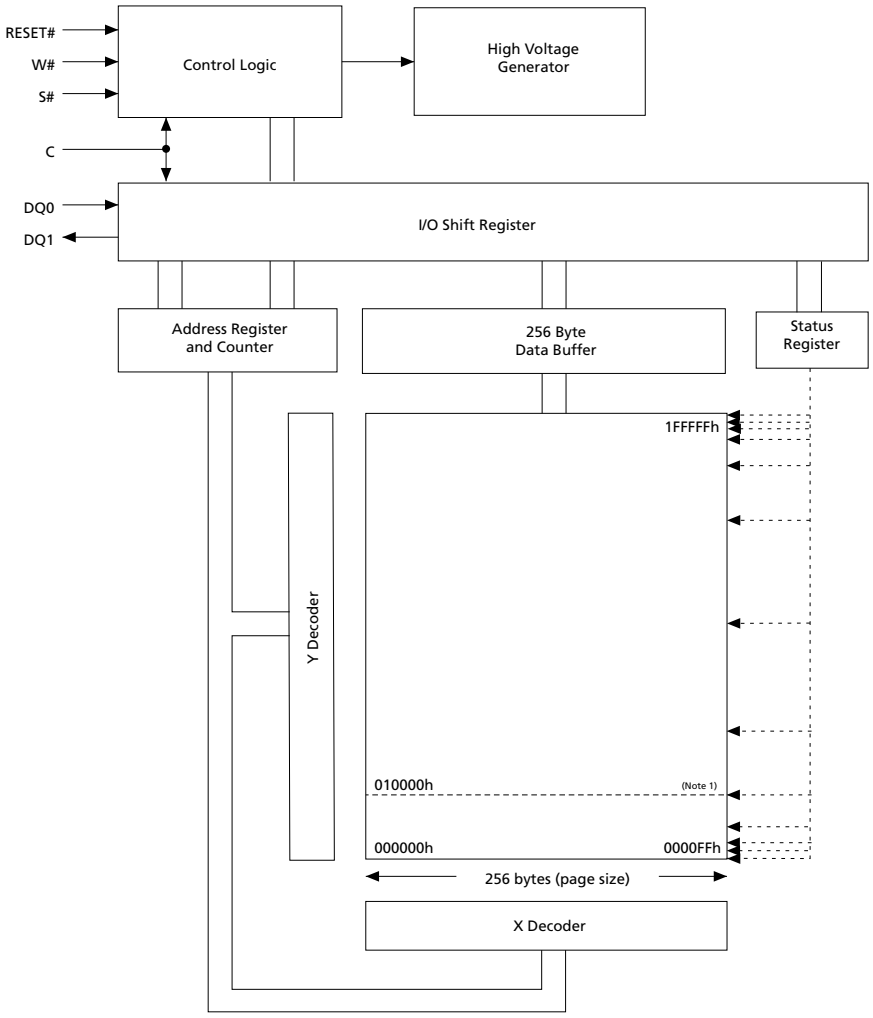
Configuration and Memory Map

Memory Configuration and Block Diagram

The device is delivered with the memory array erased; all bits are set to 1 (FFh). All usable status register bits are 0. Each page of memory can be individually programmed; bits are programmed from 1 to 0 and when written to are changed to either 0 or 1. The device is sector- and page-erasable; bits are erased from 0 to 1. The memory is configured as follows:

- 2,097,152 bytes (8 bits each)
- 32 sectors (512Kb, 65KB each)
- 8,192 pages (256 bytes each)

Figure 3: Block Diagram



Note: 1. The first 256 pages can be read-only.

Memory Map – 16Mb Density

Table 3: Sectors 31:0

Sector	Address Range	
	Start	End
31	001F 0000	001F FFFF
30	001E 0000	001E FFFF
29	001D 0000	001D FFFF
28	001C 0000	001C FFFF
27	001B 0000	001B FFFF
26	001A 0000	001A FFFF
25	0019 0000	0019 FFFF
24	0018 0000	0018 FFFF
23	0017 0000	0017 FFFF
22	0016 0000	0016 FFFF
21	0015 0000	0015 FFFF
20	0014 0000	0014 FFFF
19	0013 0000	0013 FFFF
18	0012 0000	0012 FFFF
17	0011 0000	0011 FFFF
16	0010 0000	0010 FFFF
15	000F 0000	000F FFFF
14	000E 0000	000E FFFF
13	000D 0000	000D FFFF
12	000C 0000	000C FFFF
11	000B 0000	000B FFFF
10	000A 0000	000A FFFF
9	0009 0000	0009 FFFF
8	0008 0000	0008 FFFF
7	0007 0000	0007 FFFF
6	0006 0000	0006 FFFF
5	0005 0000	0005 FFFF
4	0004 0000	0004 FFFF
3	0003 0000	0003 FFFF
2	0002 0000	0002 FFFF
1	0001 0000	0001 FFFF
0	0000 0000	0000 FFFF

Operating Features Overview

Sharing the Overhead of Modifying Data

To write or program 1 or more data bytes, two commands are required: WRITE ENABLE which is 1 byte, and a PAGE WRITE or PAGE PROGRAM command sequence, which consists of 4 bytes plus data. This is followed by the internal cycle of duration ^{t}PW or ^{t}PP.

To share this overhead, the PAGE WRITE or PAGE PROGRAM 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.

Easy Method to Modify Data

The PAGE WRITE command provides a convenient way of modifying data (up to 256 contiguous bytes at a time) and requires the start address and the new data in the instruction sequence.

The PAGE WRITE command is entered by driving chip select (S#) LOW, and then transmitting the instruction byte, 3 address bytes A[23:0] and at least 1 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 A[7:0]. When 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 may alter the contents of the memory on a byte-by-byte basis. For optimized timings, it is recommended to use the PAGE WRITE command to write all consecutive targeted bytes in a single sequence versus using several PAGE WRITE sequences with each containing only a few bytes.

Fast Method to Modify Data

The PAGE PROGRAM 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 or SECTOR ERASE 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 command to program all consecutive targeted bytes in a single sequence versus using several PAGE PROGRAM sequences with each containing only a few bytes.

Polling During a WRITE, PROGRAM, or ERASE Cycle

The following commands can be completed faster by not waiting for the worst-case delay (^{t}W, ^{t}PP, ^{t}PE, ^{t}BE, 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, PROGRAM, 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 driving it HIGH only 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, and power consumption drops to I_{CC1}.

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

Status Register

The status register contains a number of status bits that can be read by the READ STATUS REGISTER (RDSR) command. For a detailed description of the status register bits, see the READ STATUS REGISTER section.

Protection Modes

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

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

WRITE, PROGRAM, 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 command completion
- PAGE WRITE command completion
- PAGE PROGRAM command completion

- PAGE ERASE command completion
- SECTOR EASE command completion

The hardware-protected mode is entered when W# is driven LOW, causing the first 256 pages of memory to become read-only. When W# is driven HIGH, the first 256 pages of memory behave like the other pages of memory. The 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.

Serial Peripheral Interface Modes

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

Table 4: SPI Modes

Note 1 applies to the entire table

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

Note: 1. The listed SPI modes are supported in extended, dual, and quad SPI protocols.

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

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

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

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

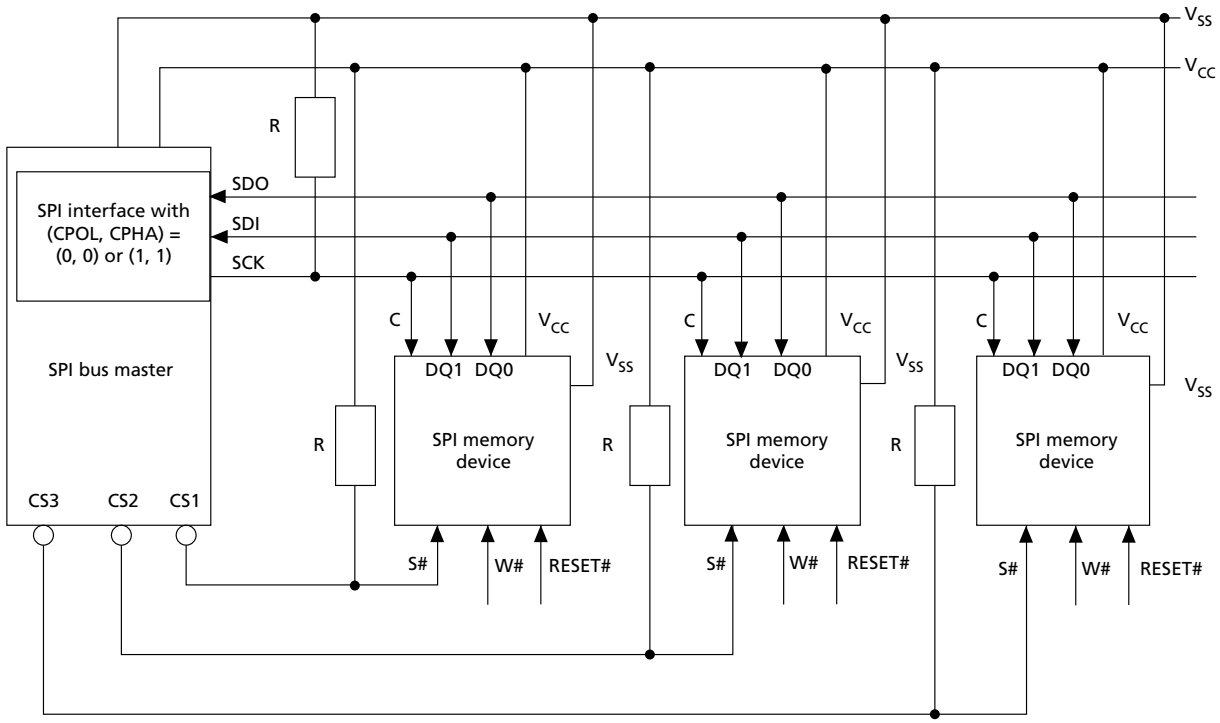
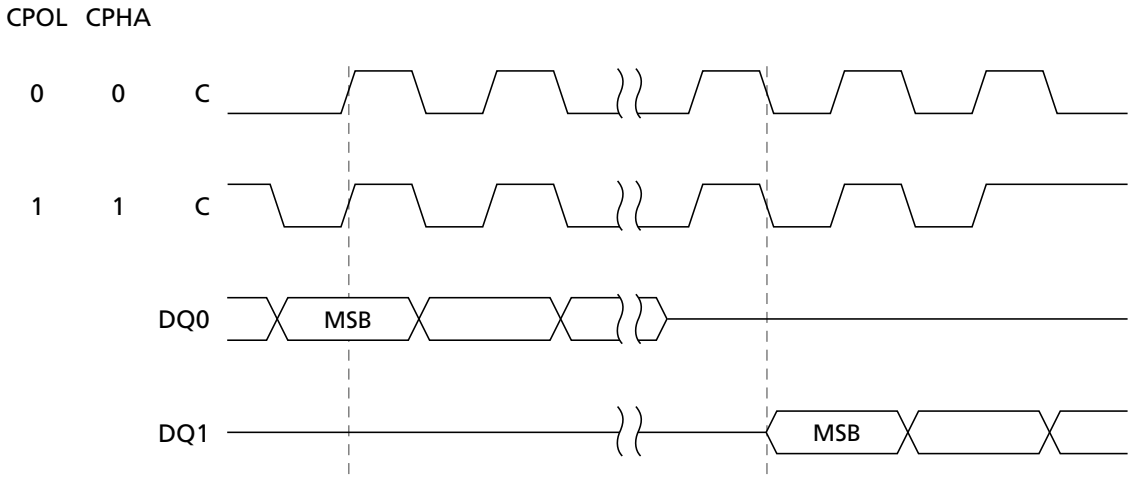


Figure 5: SPI Modes



Command Set Overview

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

Serial data inputs DQ0 and DQ1 are sampled on the first rising edge of serial clock (C) after chip select (S#) is driven LOW. Then, the 1-byte command code must be shifted into the device, most significant bit first, on DQ0 and DQ1, with each bit latched on the rising edges of C.

Every command sequence starts with a 1-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 nor 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

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
- SECTOR ERASE
- WRITE ENABLE
- WRITE DISABLE
- DEEP POWER-DOWN
- RELEASE FROM DEEP POWER-DOWN

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

Table 5: Command Set Codes

Command Name	1-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 ∞
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
SECTOR ERASE	1101 1000	D8h	3	0	0
DEEP POWER-DOWN	1011 1001	B9h	0	0	0
RELEASE from DEEP POWER-DOWN	1010 1011	ABh	0	0	0

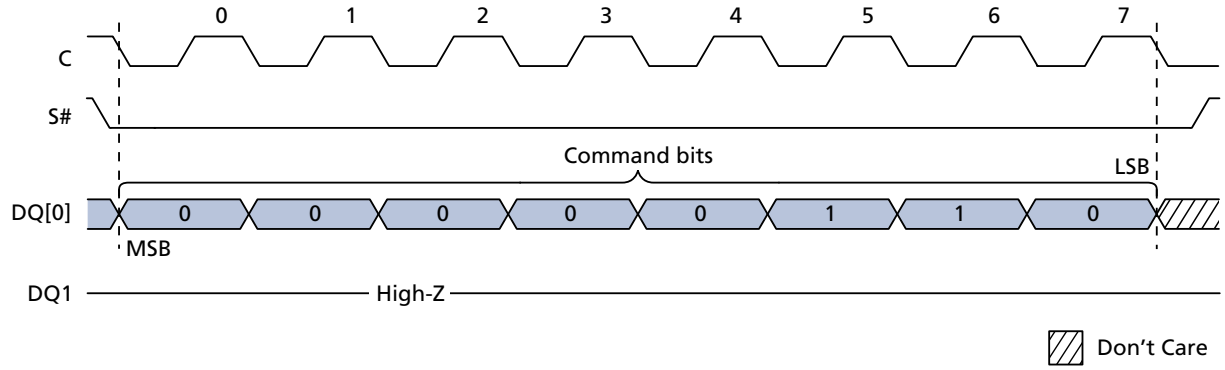
WRITE ENABLE

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, and SECTOR ERASE 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

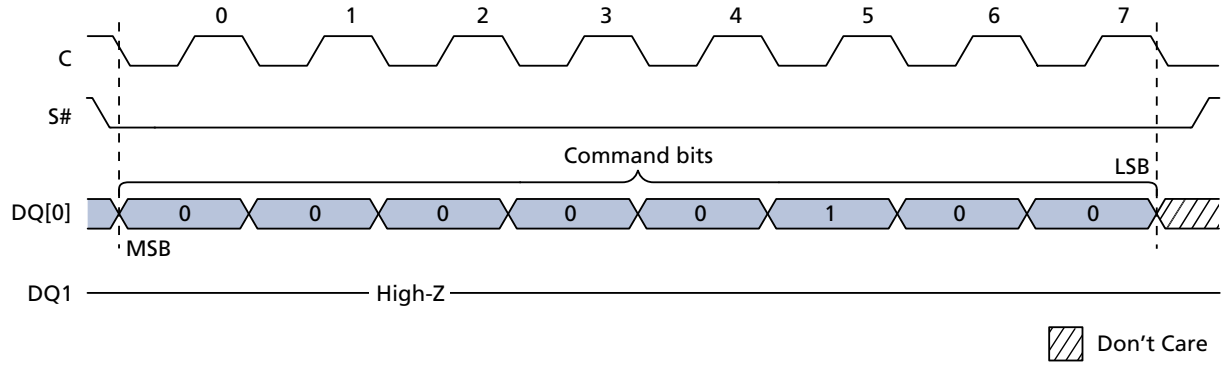
The WRITE DISABLE command resets the write enable latch (WEL) bit.

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 PAGE ERASE operation
- Completion of SECTOR 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 the length of the data to follow; the remaining 16 bytes contain optional customized factory data (CFD) content.

Table 6: READ IDENTIFICATION Data-Out Sequence

Manufacturer Identification	Device Identification		UID	
	Memory Type	Memory Capacity	CFD Length	CFD Content
20h	40h	15h	10h	16 bytes

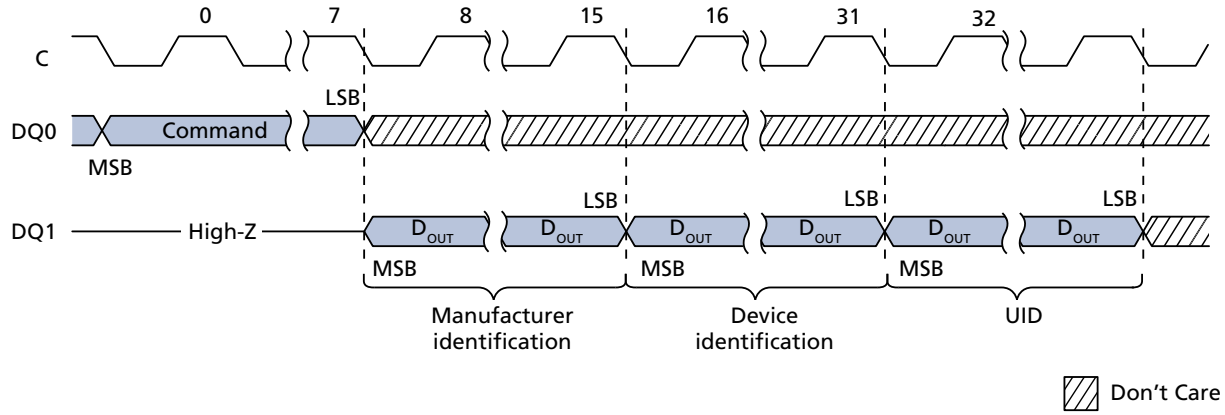
Note: 1. The CFD bytes are read-only and can be programmed with customer data upon demand. If customers do not make requests, the devices are shipped with all the CFD bytes programmed to 0.

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 stored in memory, then the 8-bit CFD length, followed by 16 bytes of CFD content. Each bit is shifted out during the falling edge of the 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

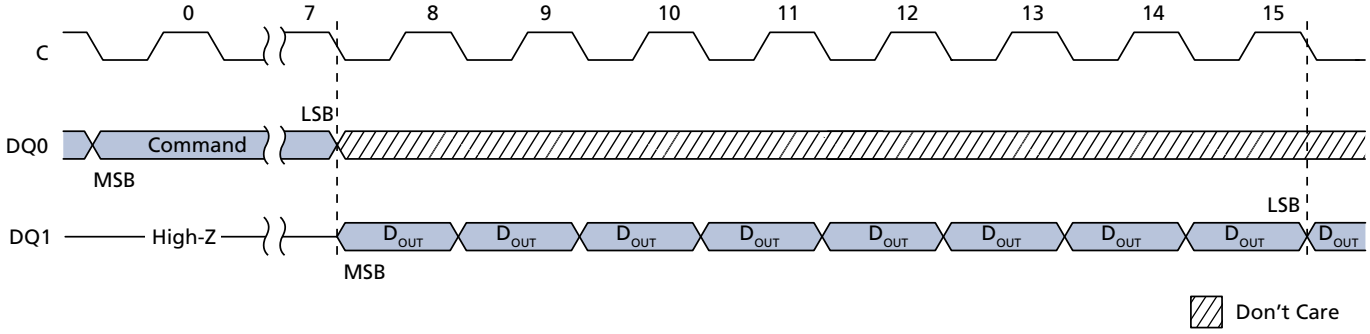
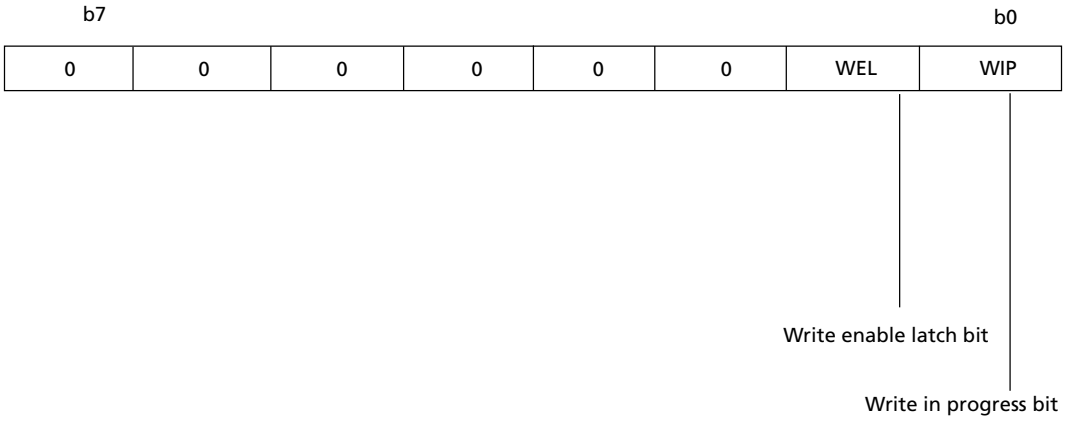


Figure 10: Status Register Format



WIP Bit

The write in progress (WIP) bit is a volatile read-only bit that indicates whether the memory is busy with a WRITE, a PROGRAM, or 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. WIP is set and reset automatically by the internal logic of the device.

WEL Bit

The write enable latch (WEL) bit is a volatile read-only bit that 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. The WEL bit is set and reset by specific commands.

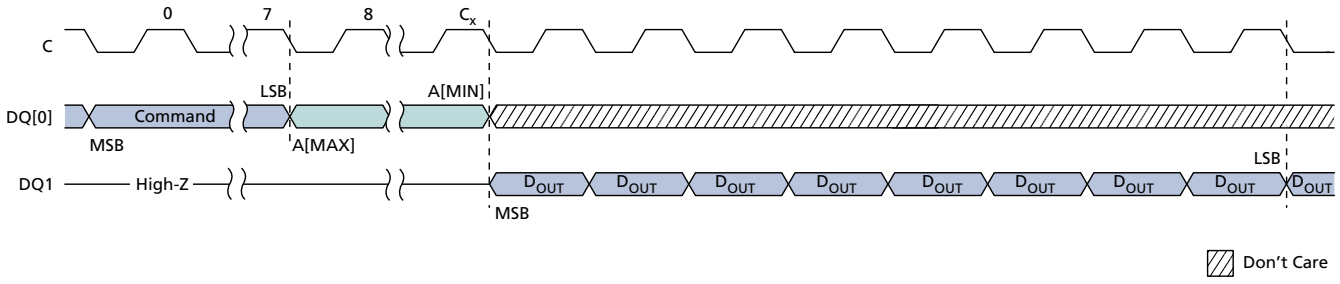
READ DATA BYTES

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 A[23:0], with each bit latched in during the rising edge of the serial clock (C). The memory contents at that address are then shifted out on a serial data output (DQ1), with each bit 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 11: READ DATA BYTES Command Sequence



Notes: 1. C_x= 7 + (A[MAX] + 1).
2. Address bits A[23:21] are "Don't Care" in the M25PE16.

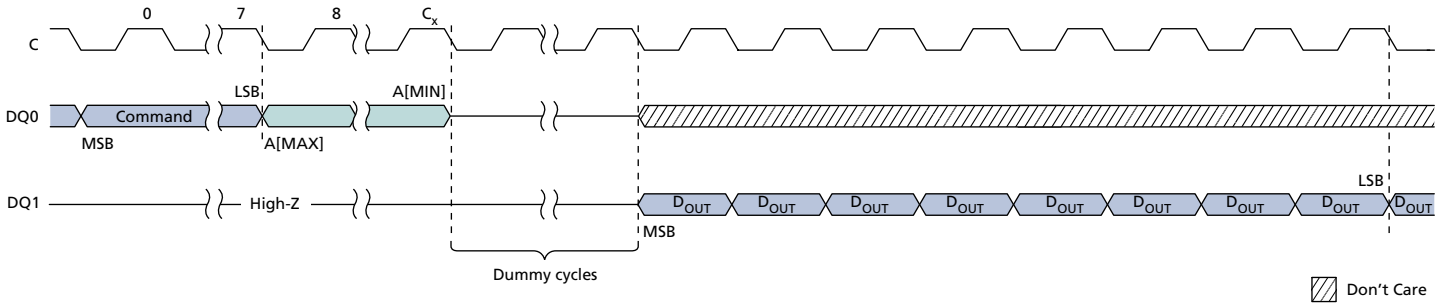
READ DATA BYTES at HIGHER SPEED

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 A[23:0] and a dummy byte, with each bit latched in during the rising edge of the serial clock (C). The memory contents at that address are then shifted out on a 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 12: READ DATA BYTES at HIGHER SPEED Command Sequence



Notes: 1. C_x= 7 + (A[MAX] + 1).
2. Address bits A[23:21] are "Don't Care" in the M25PE16.

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, 3 address bytes, and at least 1 data byte on a 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 1 byte is updated.

If the eight least-significant address bits A[7:0] are not all 0, all transmitted data that goes beyond the end of the current page is programmed from the start address of the same page; that is, from the address whose eight least-significant bits A[7:0] are all 0. 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 fewer than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without any effect on the other bytes of the same page.

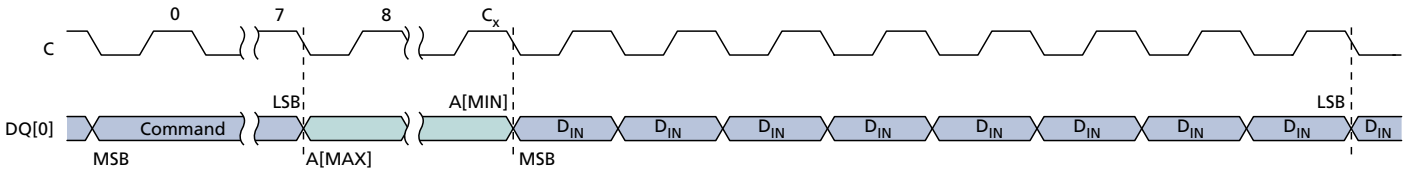
For optimized timings, it is recommended to use the PAGE WRITE command to program all consecutive targeted bytes in a single sequence rather than to use several PAGE WRITE command sequences, each containing only a few bytes.

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 is not executed if it applies to a page that is hardware-protected. Any PAGE WRITE command while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected without having any effect on the cycle that is in progress.

Figure 13: PAGE WRITE Command Sequence



Notes:
1. C_x= 7 + (A[MAX] + 1).
2. Address bits A[23:21] are "Don't Care" in the M25PE16.
3. 1 ≤n≤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, 3 address bytes, and at least 1 data byte on a serial data input (DQ0).

If the eight least-significant address bits A[7:0] are not all 0, all transmitted data that goes beyond the end of the current page is programmed from the start address of the same page, that is, from the address whose eight least-significant bits A[7:0] are all 0. 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 fewer than 256 data bytes are sent to device, they are correctly programmed at the requested addresses without any effect on the other bytes of the same page.

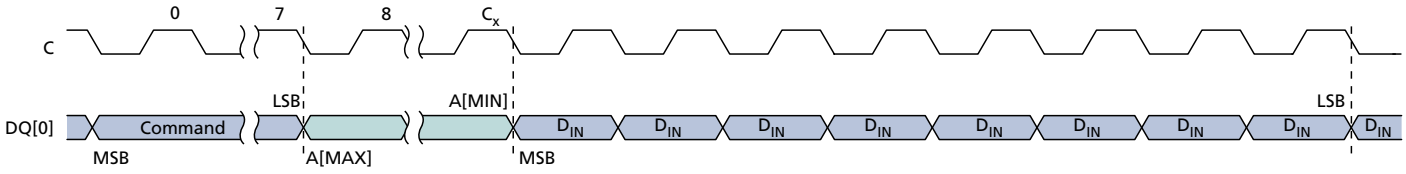
For optimized timings, it is recommended to use the PAGE PROGRAM command to program all consecutive targeted bytes in a single sequence rather than to use several PAGE PROGRAM sequences, each containing only a few bytes.

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

As soon as S# is driven HIGH, the self-timed PAGE PROGRAM cycle is initiated; the cycles's duration is ^{t}PP. While the PAGE PROGRAM cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed PAGE PROGRAM cycle and 0 when the cycle is completed. At some unspecified time before the cycle is completed, the write enable latch (WEL) bit is reset.

A PAGE PROGRAM command is not executed if it applies to a page protected by all the block-protect bits.

Figure 14: PAGE PROGRAM Command Sequence



Notes:
1. Cx = 7 + (A[MAX] + 1).
2. Address bits A[23:21] are "Don't Care" in the M25PE16.

PAGE ERASE

The PAGE ERASE command sets to 1 (FFh) all bits inside the designated 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.

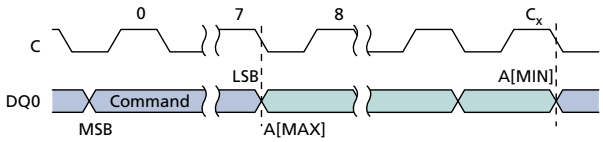
The PAGE ERASE command is entered by driving chip select (S#) LOW, followed by the command code and 3 address bytes on a serial data input (DQ0). Any address inside the sector is a valid address for the PAGE ERASE command. S# must be driven LOW for the entire duration of the sequence.

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 0 when the cycle is completed. At some unspecified time before the cycle is completed, the WEL bit is reset.

A PAGE ERASE command is not executed if it applies to a page that is protected by the block-protect bits BP1 and BP0.

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

Figure 15: SECTOR ERASE Command Sequence



Notes: 1. C_x= 7 + (A[MAX] + 1).
2. Address bits A[23:21] are "Don't Care" in the M25PE16.

SECTOR ERASE

The SECTOR ERASE command sets all bits inside the chosen sector to 1 (FFh). 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.

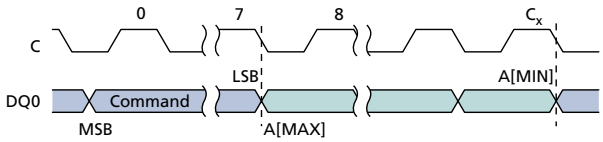
The SECTOR ERASE command is entered by driving chip select (S#) LOW, followed by the command code and 3 address bytes on a serial data input (DQ0). Any address inside the sector is a valid address for the SECTOR ERASE command. S# must be driven LOW for the entire duration of the sequence.

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

Figure 16: SECTOR ERASE Command Sequence



Notes: 1. C_x= 7 + (A[MAX] + 1).
2. Address bits A[23:21] are "Don't Care" in the M25PE16.

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 standby power mode if there is no internal cycle currently in progress. After entering 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}.

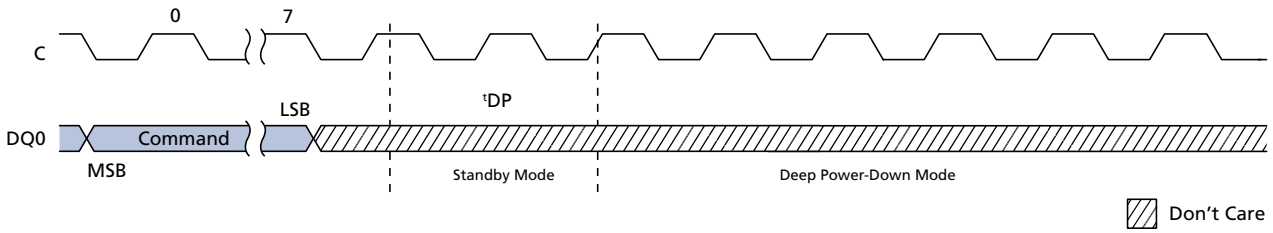
To take the device out of deep power-down mode, the RELEASE from DEEP POWER-DOWN command must be issued. Other commands must not be issued while the device is in deep power-down mode. The deep power-down mode stops automatically at power-down. The device always powers up in standby power mode.

The DEEP POWER-DOWN command is entered by driving S# LOW, followed by the command code on a 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, a delay of ^{t}DP is required before the supply current is reduced to I_{CC2}, and 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 17: DEEP POWER-DOWN Command Sequence



RELEASE from DEEP POWER-DOWN

After 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 deep power-down mode.

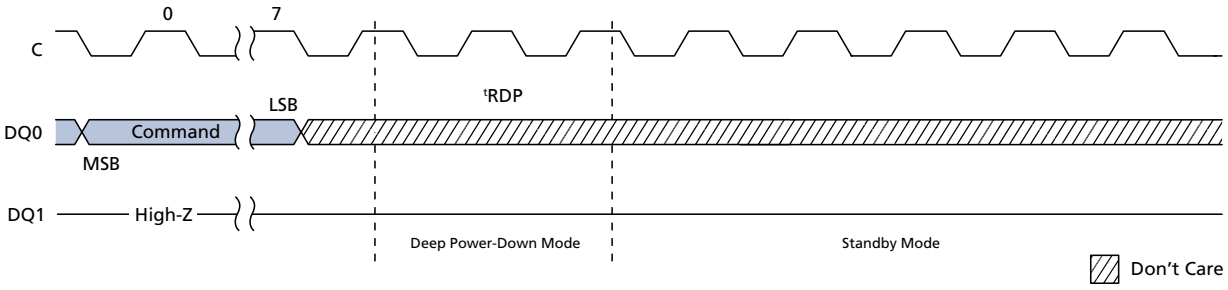
The RELEASE from DEEP POWER-DOWN command is entered by driving chip select (S#) LOW, followed by the command code on a serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

The RELEASE from DEEP POWER-DOWN command is terminated by driving S# HIGH. Sending additional clock cycles on the 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 18: 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, S# must follow the voltage applied on $V_{CC,min}$ until V_{CC} reaches the correct value: $V_{CC,min}$ at power-up, and then for a further delay of ^{t}VSL; V_{SS} at power-down. A safe configuration is provided in the Serial Peripheral Interface Modes section.

To avoid data corruption and inadvertent WRITE operations during power-up, a power-on-reset (POR) circuit is included. The logic inside the device is held at 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 command. In addition, the device ignores the following commands 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

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

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

If the time, ^{t}VSL, has elapsed, after V_{CC} rises above $V_{CC,min}$, the device can be selected for READ commands 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 19: Power-Up Timing

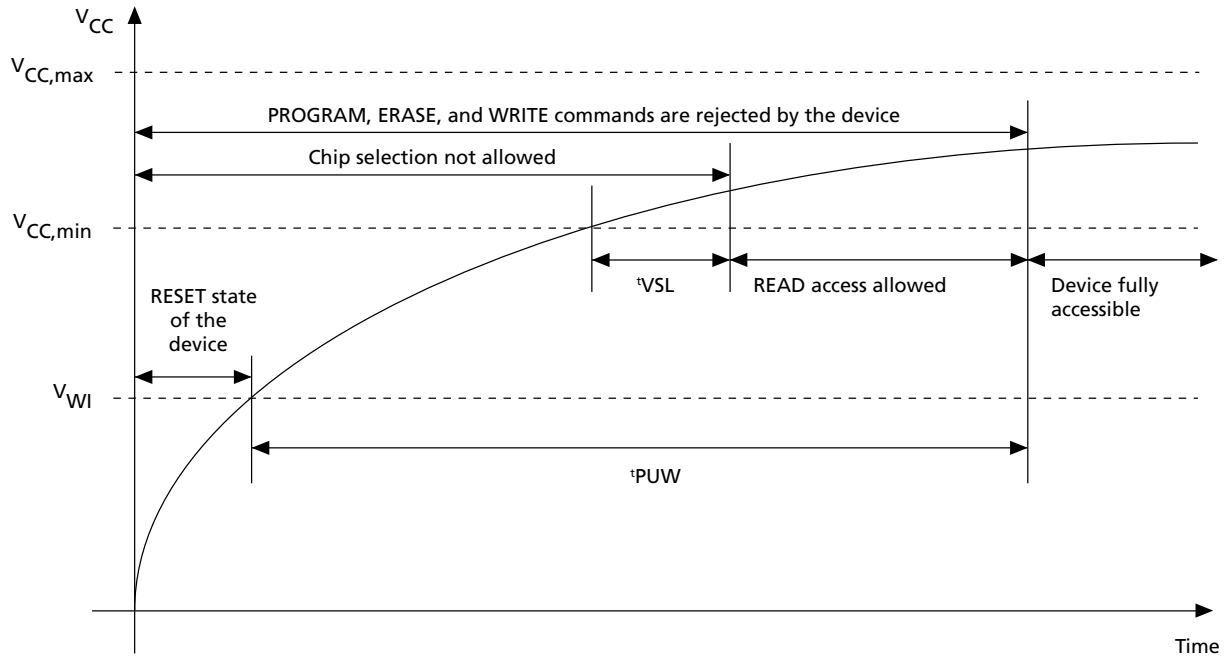


Table 7: Power-up Timing and V_{WI} Threshold

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 command	1	10	ms
V_{WI}	Write-inhibit voltage	1.5	2.5	V

Note: 1. These parameters are characterized only, over the temperature range -40°C to +85°C.

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.

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 100nF.

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 command.

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.

Maximum Ratings and Operating Conditions

Stressing the device above the rating listed in the Absolute Maximum Ratings table may cause permanent damage to the device. These are stress ratings only, and operation of the device at these or any other conditions beyond those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.

Table 8: Absolute Maximum Ratings

Symbol	Parameter	Min.	Max.	Unit	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. Compliant with JEDEC Std J-STD-020C (for small body, Sn-Pb or Pb assembly) and the European directive on Restrictions on Hazardous Substances (RoHS) 2002/95/EU
2. JEDEC Std JESD22-A114A (C1=100 pF, R1=1500 Ω, R2=500 Ω)

Table 9: Operating Conditions

Symbol	Parameter	Min.	Max.	Unit
V_{CC}	Supply voltage	2.7	3.6	V
T_A	Ambient operating temperature	–40	85	°C

Electrical Characteristics

Table 10: DC Current Specifications

Parameter	Symbol	Test Conditions	Min	Max	Units
Input leakage current	I_{LI}	–	–	±2	µA
Output leakage current	I_{LO}	–	–	±2	µA
Standby current (standby and reset modes)	I_{CC1}	S# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	–	50	µA
Deep power-down current	I_{CC2}	S# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	–	10	µA
Operating current (READ)	I_{CC3}	C = 0.1 × V_{CC} / 0.9 × V_{CC} at 75 MHz, DQ1 = open	–	12	mA
		C = 0.1 × V_{CC} / 0.9 × V_{CC} at 33 MHz, DQ1 = open	–	4	mA
Operating current (PAGE PROGRAM)	I_{CC4}	S# = V_{CC}	–	15	mA
Operating current (SECTOR ERASE)	I_{CC5}	S# = V_{CC}	–	15	mA

Table 11: DC Voltage Specifications

Parameter	Symbol	Test Conditions	Min	Max	Units
Input LOW voltage	V_{IL}	–	–0.5	0.3 × V_{CC}	V
Input HIGH voltage	V_{IH}	–	0.7 × V_{CC}	V_{CC} + 0.4	V
Output LOW voltage	V_{OL}	I_{OL} = 1.6mA	–	0.4	V
Output HIGH voltage	V_{OH}	I_{OH} = –100µA	V_{CC} - 0.2	–	V

AC Characteristics

In the following AC specifications, output High-Z is defined as the point where Data Out is no longer driven.

Table 12: AC Measurement Conditions

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

Figure 20: AC Measurement I/O Waveform

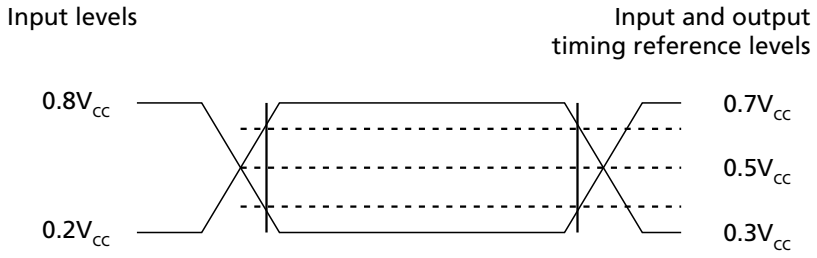


Table 13: Capacitance

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

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

Table 14: AC Specifications (50 MHz)

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

Parameter	Symbol	Alt	Min	Typ	Max	Unit	Notes
Clock frequency for the following commands: FAST_READ, RDLR, PW, PP, WRLR, PE, SE, SSE, DP, RDP, WREN, WRDI, RDSR, WRSR	f_C	f_C	DC	–	50	MHz	
Clock frequency for READ command	f_R	–	DC	–	33	MHz	
Clock HIGH time	^tCH	^tCLH	9	–	–	ns	1
Clock LOW time	^tCL	^tCLL	9	–	–	ns	1
Clock slew rate (peak-to-peak)			0.1	–	–	V/ns	2
S# active setup time (relative to C)	^tSLCH	^tCSS	5	–	–	ns	
S# not active hold time (relative to C)	^tCHSL		5	–	–	ns	
Data In setup time	^tDVCH	^tDSU	2	–	–	ns	
Data In hold time	^tCHDX	^tDH	5	–	–	ns	
S# active hold time (relative to C)	^tCHSH	–	5	–	–	ns	
S# not active setup time (relative to C)	^tSHCH	–	5	–	–	ns	
S# deselect time	^tSHSL	^tCSH	100	–	–	ns	
Output disable time	^tSHQZ	^tDIS	–	–	8	ns	2
Clock LOW to output valid	^tCLQV	^tV	–	–	8	ns	
Output hold time	^tCLQX	^tHO	0	–	–	ns	
WRITE PROTECT setup time	^tWHSL	–	50	–	–	ns	
WRITE PROTECT hold time	^tSHWL	–	100	–	–	ns	
S# to deep power-down mode	^tDP	–	–	–	3	µs	2
S# HIGH to standby mode	^tRDP	–	–	–	30	µs	2
Reset pulse width	^tRLRH	–	–	–	10	ns	2
Reset recovery time	^tRHSL	–	–	–	3	ns	
Chip deselected before RESET# is asserted	^tSHRH	–	–	–	10	ns	
PAGE WRITE cycle time (256 bytes)	^tPW	–	–	11	23	ms	3
PAGE PROGRAM cycle time (256 bytes)	^tPP	–	–	0.8	3	ms	3
PAGE PROGRAM cycle time (n bytes)	^tPP	–	–	int(n/8) × 0.025	3	ms	
PAGE ERASE cycle time	^tPE	–	–	10	20	ms	
SECTOR ERASE cycle time	^tSE	–	–	1	5	s	

Notes:
1. The ^tCH and ^tCL 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. n = number of bytes to program; int(A) corresponds to the upper integer part of A; for example, int(12/8) = 2, int(32/8) = 4, int(15.3) = 16.

Table 15: AC Specifications (75 MHz)

Parameter	Symbol	Alt	Min	Typ	Max	Unit	Notes
Clock frequency for all commands (except READ)	f_C	f_C	DC	–	75	MHz	
Clock frequency for READ command	f_R	–	DC	–	33	MHz	
Clock HIGH time	^tCH	^tCLH	6	–	–	ns	1
Clock LOW time	^tCL	^tCLL	6	–	–	ns	1
Clock rise time (peak-to-peak)	^tCLCH	–	0.1	–	–	V/ns	2
S# active setup time (relative to C)	^tSLCH	^tCSS	5	–	–	ns	
S# not active hold time (relative to C)	^tCHSL		5	–	–	ns	
Data In setup time	^tDVCH	^tDSU	2	–	–	ns	
Data In hold time	^tCHDX	^tDH	5	–	–	ns	
S# active hold time (relative to C)	^tCHSH	–	5	–	–	ns	
S# not active setup time (relative to C)	^tSHCH	–	5	–	–	ns	
S# deselect time	^tSHSL	^tCSH	100	–	–	ns	
Output disable time	^tSHQZ	^tDIS	–	–	8	ns	2
Clock LOW to output valid under 30pF	^tCLQV	^tV	–	–	8	ns	
Clock LOW to output valid under 10pF			–	–	6	ns	
Output hold time	^tCLQX	^tHO	0	–	–	ns	
WRITE PROTECT setup time	^tWHSL	–	20	–	–	ns	3
WRITE PROTECT hold time	^tSHWL	–	100	–	–	ns	3
S# HIGH to deep power-down mode	^tDP	–	–	–	3	µs	2
S# HIGH to standby mode	^tRDP	–	–	–	30	µs	2
WRITE STATUS REGISTER cycle time	^tW	–	–	3	15	µs	4
PAGE WRITE cycle time (256 bytes)	^tPW	–	–	11	23	ms	4
PAGE PROGRAM cycle time (256 bytes)	^tPP	–	–	0.8	3	ms	4
PAGE PROGRAM cycle time (*n* bytes)	^tPP	–	–	int (n/8) × 0.025	3	ms	4, 5
PAGE ERASE cycle time	^tPE	–	–	10	20	s	
SECTOR ERASE cycle time	^tSE	–	–	1	5	s	

Notes:
1. The ^tCH and ^tCL 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 the WRITE STATUS REGISTER command when SRWD is set to 1.
4. When using the PAGE PROGRAM command to program consecutive bytes, optimized timings are obtained in one sequence that includes all the bytes rather than in several sequences of only a few bytes (1 ≤ n ≤ 256).
5. int(A) corresponds to the upper integer part of A; for example, int(12/8) = 2, int(32/8) = 4, int(15.3) = 16.

Table 16: Reset Specifications

Parameter	Symbol	Alt	Conditions	Min	Typ	Max	Unit	Notes
Reset pulse width	^{t}RLRH	^{t}RST		10	–	–	µs	1
Chip select HIGH to Reset HIGH	^{t}SHRH	–	Chip should have been deselected before RESET# de-asserted	10	–	–	ns	
Reset recovery time	^{t}RHSL	^{t}REC	Clock frequency for commands (see note)	–	–	30	µs	2, 1, 3
			Under completion of ERASE or PROGRAM cycle for commands (see note)	–	–	300	µs	1
			Device deselected (S# HIGH) and in standby	–	–	0	µs	1

Notes:
1. Value guaranteed by characterization; not 100% tested in production.
2. WRITE ENABLE/DISABLE, READ DATA BYTES, READ DATA BYTES at HIGHER SPEED, PAGE WRITE, PAGE PROGRAM, PAGE ERASE, SECTOR ERASE, DEEP POWER-DOWN, RELEASE from DEEP POWER-DOWN, READ STATUS REGISTER, READ IDENTIFICATION.
3. S# remains LOW while RESET# is LOW.
4. PAGE WRITE, PAGE PROGRAM, PAGE ERASE, SECTOR ERASE.

Figure 21: Serial Input Timing

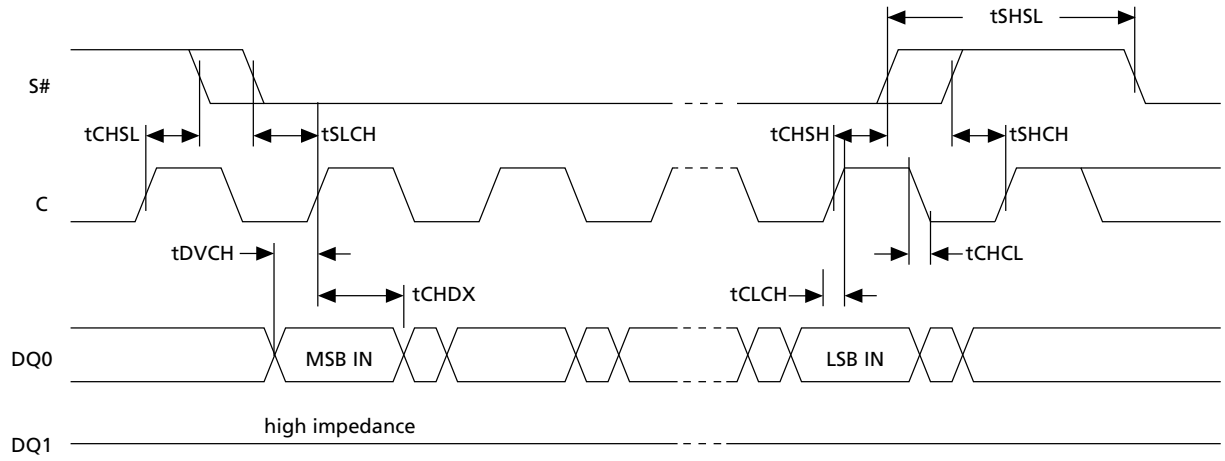


Figure 22: Write Protect Setup and Hold during WRSR when SRWD=1 Timing

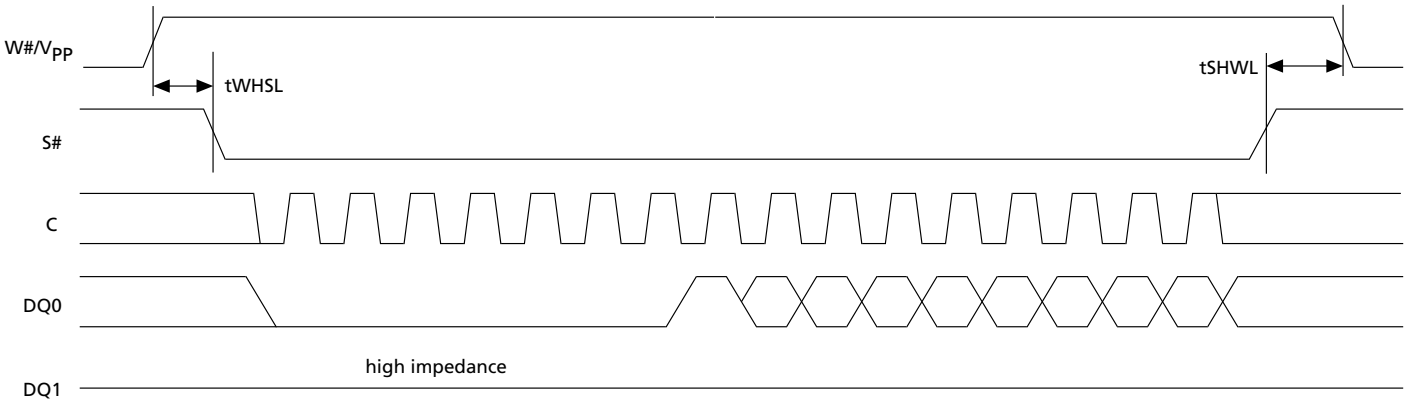


Figure 23: Hold Timing

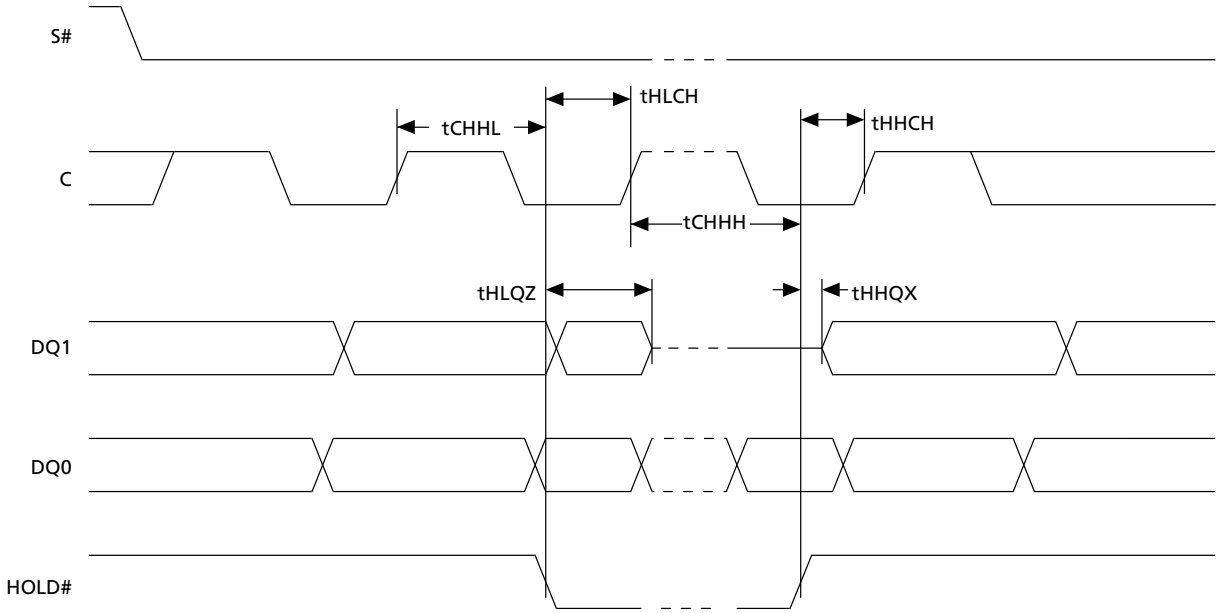


Figure 24: Output Timing

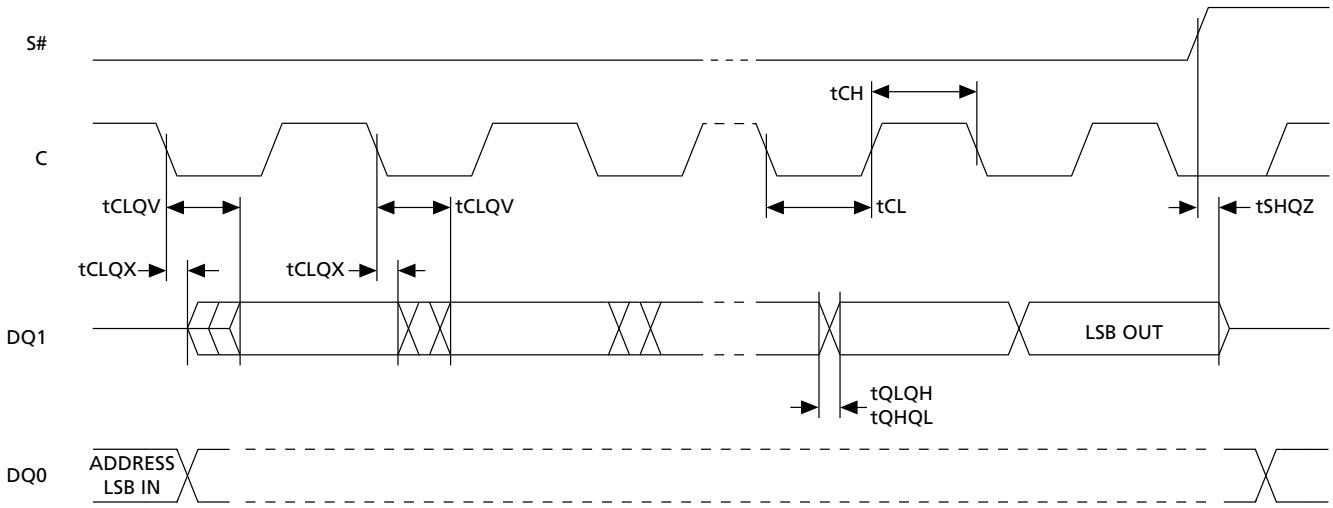
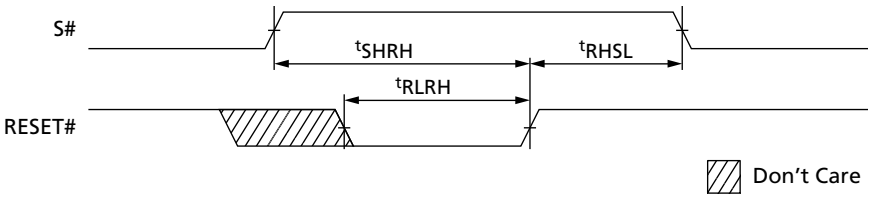
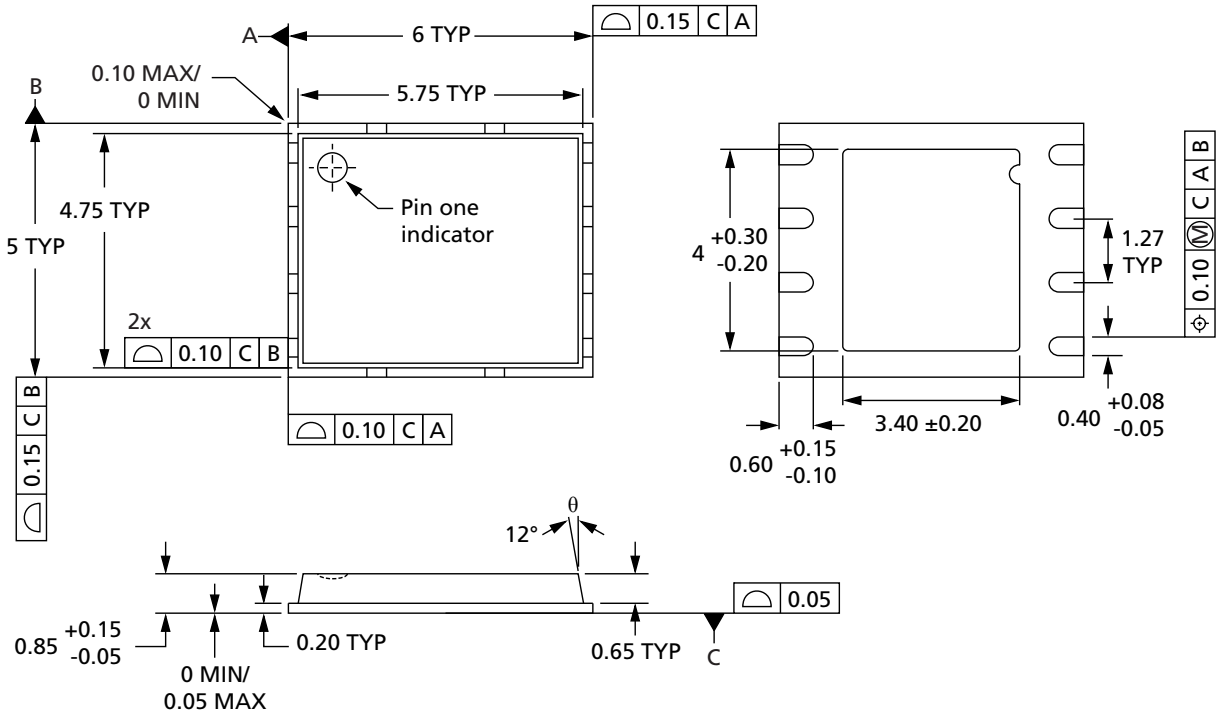


Figure 25: Reset AC Timing During PROGRAM or ERASE Cycle



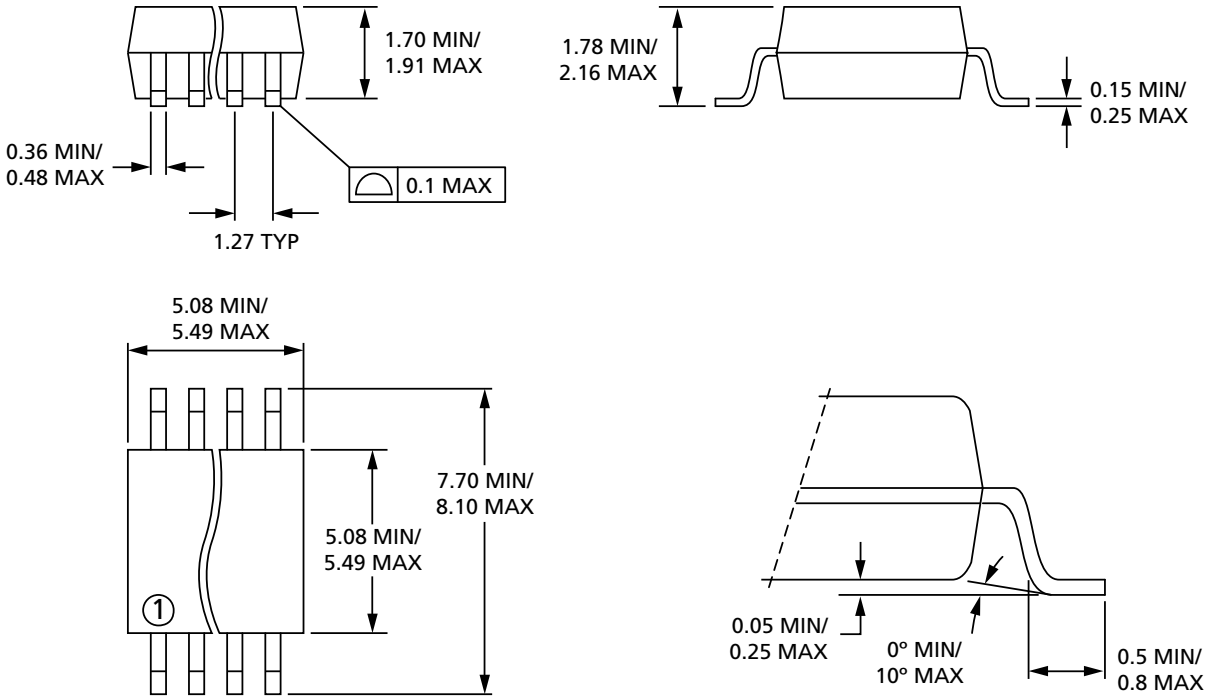
Package Information

Figure 26: VFQFPN8 (MLP8) 6mm x 5mm



Note: 1. Drawing is not to scale.

Figure 27: SO8W 208 mils Body Width



Note: 1. Drawing is not to scale.

Device Ordering Information

Standard Parts

Micron Serial NOR Flash memory is available in different configurations and densities. Verify valid part numbers using Micron's part catalog search at micron.com.

To compare features and specifications by device type, visit micron.com/products. Contact the factory for any devices not found.

For more information on how to identify products and top-side marking by 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 17: Part Number Information Scheme

Part Number Category	Category Details
Device type	M45PE = Page-erasable serial Flash memory
Density	16 = 16Mb (2Mb x 8)
Security	– = No extra security
Operating voltage	V = V_{CC} = 2.7–3.6V
Package	MP = VFQFPN8, 6mm x 5mm (MLP8)
	MW = SO8W (208 mil 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

Revision History

Rev. C – 03/14

- In PAGE ERASE, revised the first sentence to say "designated page" instead of "sector."

Rev. B – 06/13

- Added SO8W 208 mils Body Width figure

Rev. A – 05/13

- Micron rebrand

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

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.