Micron M25PE80 Serial Flash Memory

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

Features

- 8Mb 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: 8Mb
- Deep power-down mode: 1µA (TYP)
- Electronic signature
  - JEDEC standard 2-byte signature (8014h)
  - Unique ID code (UID) with 16 bytes read-only
- 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)
  - QFN8L (MS) 6mm x 5mm (MLP8)
  - SO8W (MW) 208 mils
  - SO8N (MN) 150 mils
- Automotive grade parts available

Contents

List of Figures

List of Tables

Functional Description

The M25PE80 is an 8Mb (1Mb 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 16 sectors, divided into 16 subsectors each (256 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 4096 pages, or 1,048,576 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 whoe 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.

Note: The write protection (defined by the WL and LD lock bits) of the 4KB subsectors in the top and bottom sectors is no longer offered. For more details, see PCNMPG062148.

Figure 1: Logic Diagram

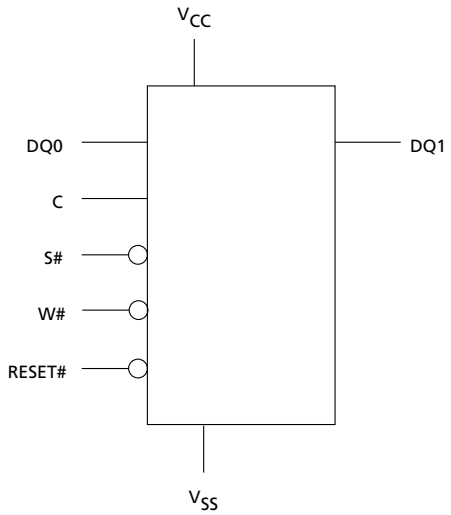
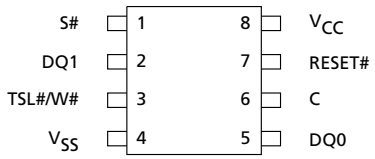


Table 1: Signal Names

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	Input
V_{SS}	Ground	–

Figure 2: Pin Connections: VFQFPN, QFN8L, and SO



There is an exposed central pad on the underside of the MLP8 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:

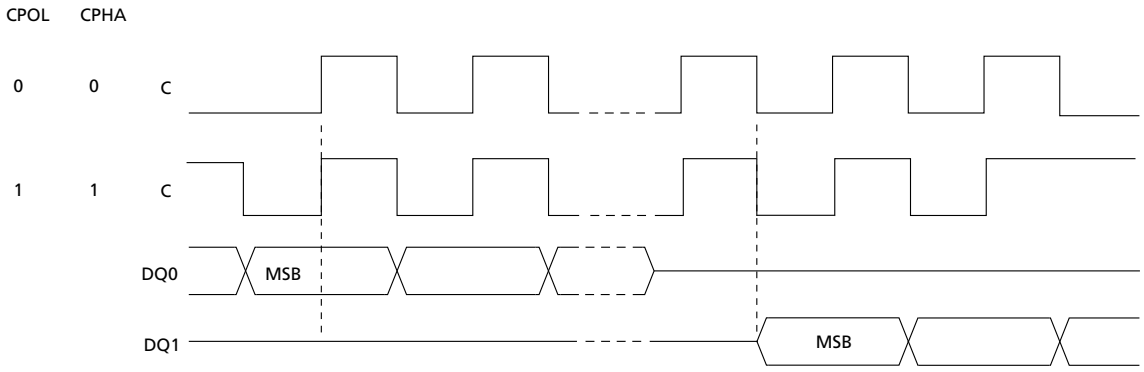
- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

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:

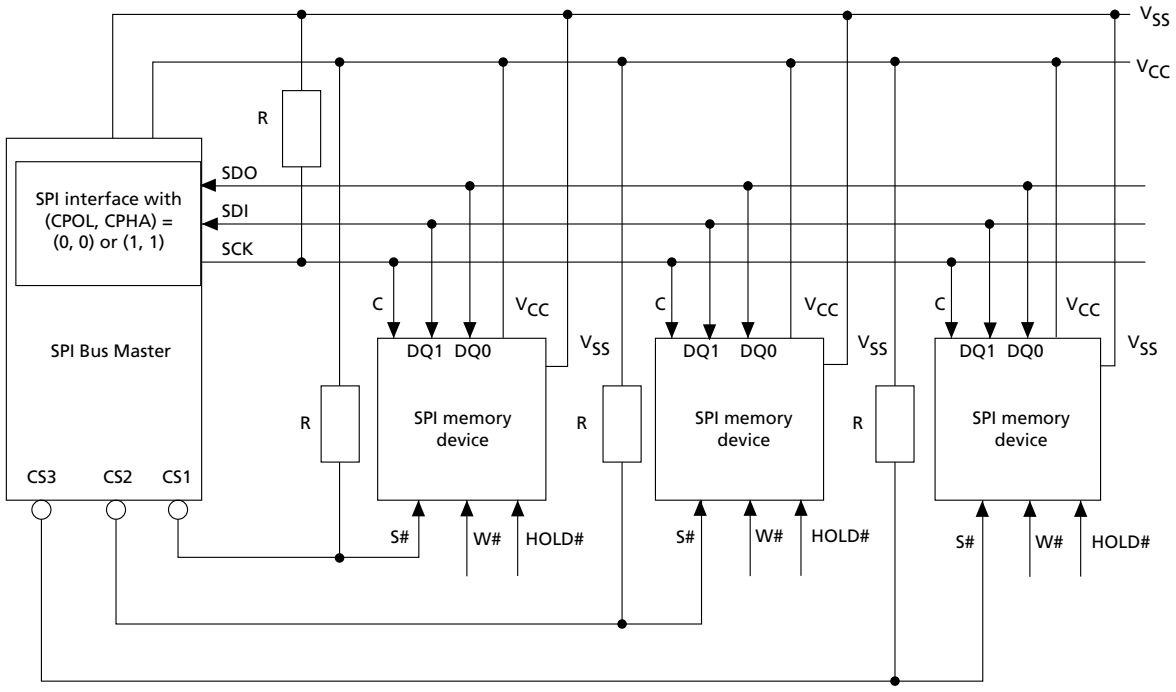
- C remains at 0 for (CPOL=0, CPHA=0)
- C remains at 1 for (CPOL=1, CPHA=1)

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 $^{t}SHCH$ requirement is met.
4. The typical value of R is 100 kΩ, assuming that the time constant R × 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 × C_p = 5µs), 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 providesa 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_{W}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_{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 (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 EASE (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 inadvertant WRITE, PROGRAM, and ERASE commands while the device is not in active use.

Specific Hardware and Software Protection

The M25PE80 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 0-15)

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.

Table 4: Not for new design: TY7 process only, software protection scheme truth table (sectors 0 and 15)

Sector Lock Register		Subsector Lock Register		Protection Status
Lock Down Bit	Write Lock Bit	Lock Down Bit	Write Lock Bit	
0	0	0	0	Current subsector unprotected from PROGRAM / ERASE / WRITE operations; current subsector protection status reversible.
		0	1	Current subsector protected from PROGRAM / ERASE / WRITE operations; current subsector protection status reversible.
		1	0	Current subsector unprotected from PROGRAM / ERASE / WRITE operations; current subsector protection status cannot be changed except by a reset or power-up.
		1	1	Current subsector protected from PROGRAM / ERASE / WRITE operations; protection status cannot be changed except by a reset or power-up.
	1	0	1	All subsectors protected from PROGRAM / ERASE / WRITE operations, current subsector protection status reversible.
		1	1	All subsectors protected from PROGRAM / ERASE / WRITE operations, current subsector protection cannot be changed except by a reset or power-up.
1	0	1	0	Current subsector unprotected from PROGRAM / ERASE / WRITE operations, all subsectors protection status cannot be changed except by a reset or power-up.
		1	1	Current subsector protected from PROGRAM / ERASE / WRITE operations, all subsectors protection status cannot be changed except by a reset or power-up.
	1	1	1	All subsectors protected with their protection status cannot be changed except by a reset or power-up.

All other bit combinations are not-applicable. For more details, see the WRLR instruction.

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 5: 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 15)	1
0	0	1	Upper sixteenth (sector 15)	Lower 15/16ths (sectors 0 to 14)	
0	1	0	Upper eighth (sectors 14 and 15)	Lower 7/8ths (sectors 0 to 13)	
0	1	1	Upper quarter (sectors 12 to 15)	Lower three quarters (sectors 0 to 11)	
1	0	0	Upper half (sectors 8 to 15)	Lower half (sectors 0 to 7)	
1	0	1	All sectors (sectors 0 to 15)	none	
1	1	0	All sectors (sectors 0 to 15)	none	
1	1	1	All sectors (sectors 0 to 15)	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 M25PE80 memory is organized as follows:

- 4096 pages (256 bytes each)
- 1,048,576 bytes (8 bits each)
- 16 sectors (512Kb, 65,536 bytes each)
- 256 sectors (32Kb, 4096 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- or bulk-erasable (bits are erased from 0 to 1).

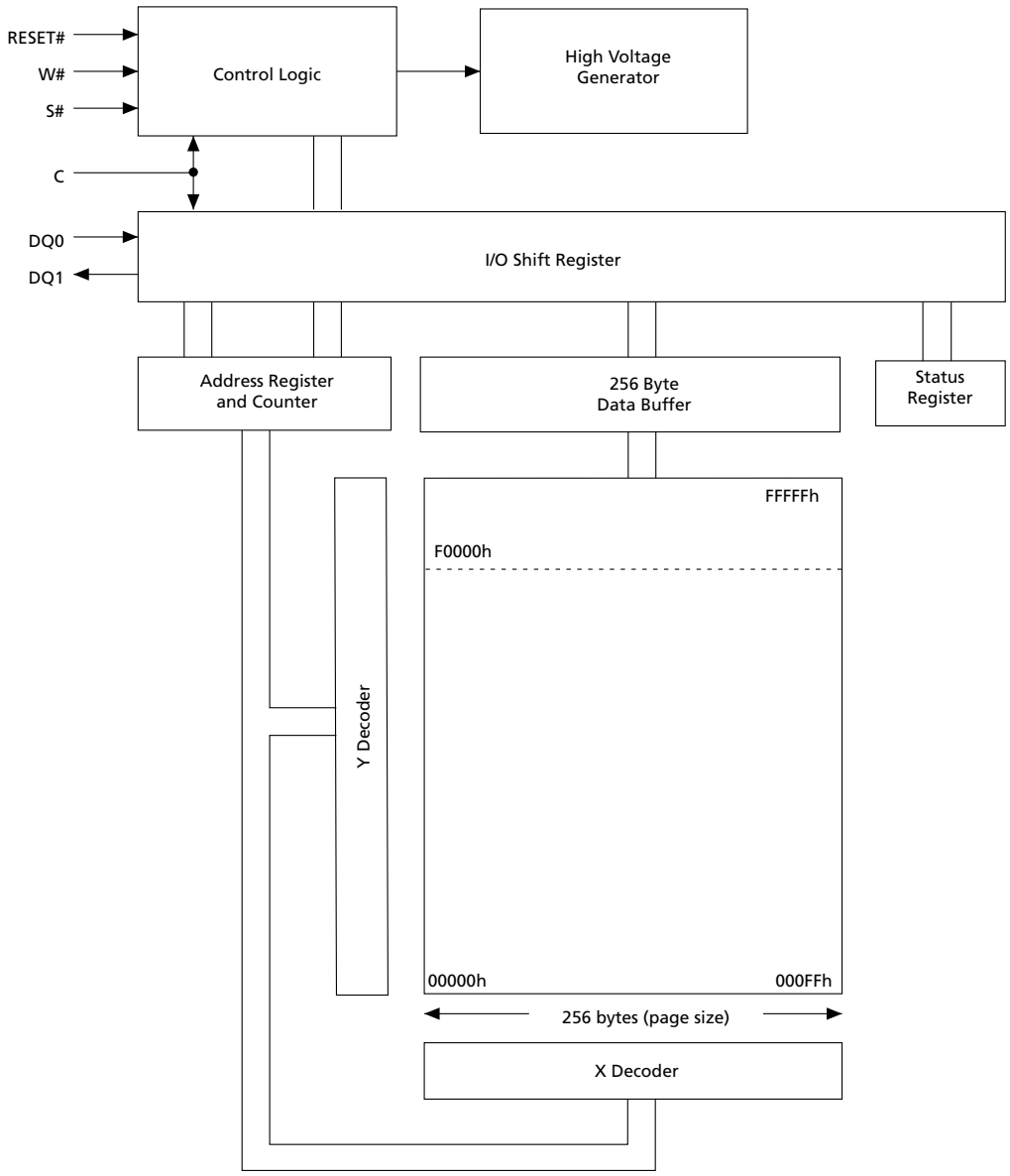
Table 6: Memory Organization

Sector	Subsector	Address Range	
		Start	End
15	255	FF000h	FFFFFh
	⋮	⋮	⋮
	240	F0000h	F0FFFh
14	239	EF000h	EFFFFh
	⋮	⋮	⋮
	224	E0000h	E0FFFh
13	223	DF000h	DFFFFh
	⋮	⋮	⋮
	208	D0000h	D0FFFh
12	207	CF000h	CFFFFh
	⋮	⋮	⋮
	192	C0000h	C0FFFh
11	191	BF000h	BFFFFh
	⋮	⋮	⋮
	176	B0000h	B0FFFh
10	175	AF000h	AFFFFh
	⋮	⋮	⋮
	160	A0000h	A0FFFh
9	159	9F000h	9FFFFh
	⋮	⋮	⋮
	144	90000h	90FFFh
8	143	8F000h	8FFFFh
	⋮	⋮	⋮
	128	80000h	80FFFh

Table 6: Memory Organization (Continued)

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
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 7: 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 3
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

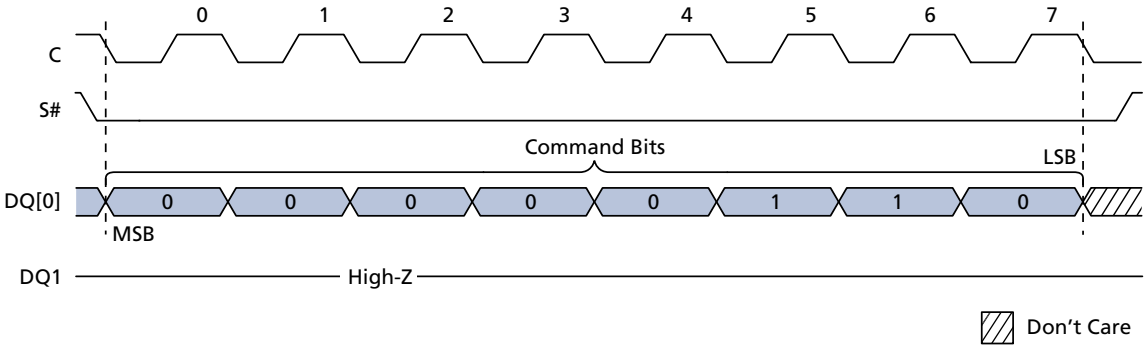
WRITE ENABLE

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, 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

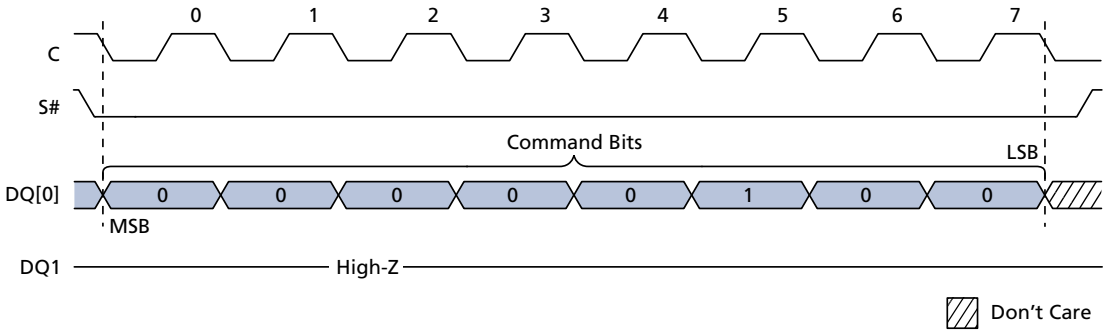
The WRITE DISABLE command resets the write enable latch (WEL) bit.

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 8: READ IDENTIFICATION Data Out Sequence

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

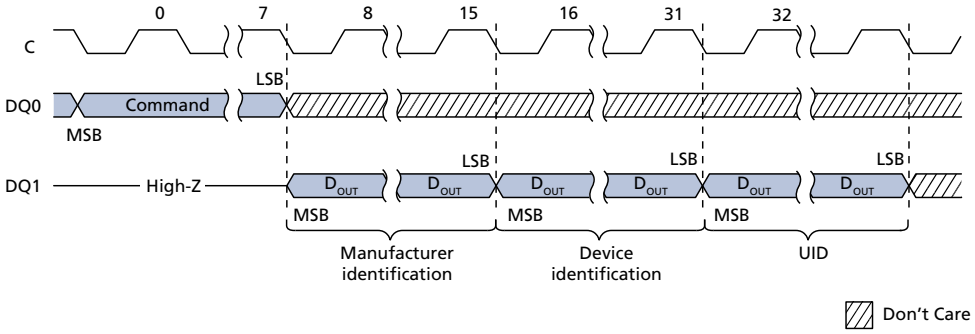
Note: 1. The CFD bytes are read-only and can be programmed with customer data upon demand. If customers do not make requests, the devices are shipped with all the CFD bytes programmed to zero (00h).

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.

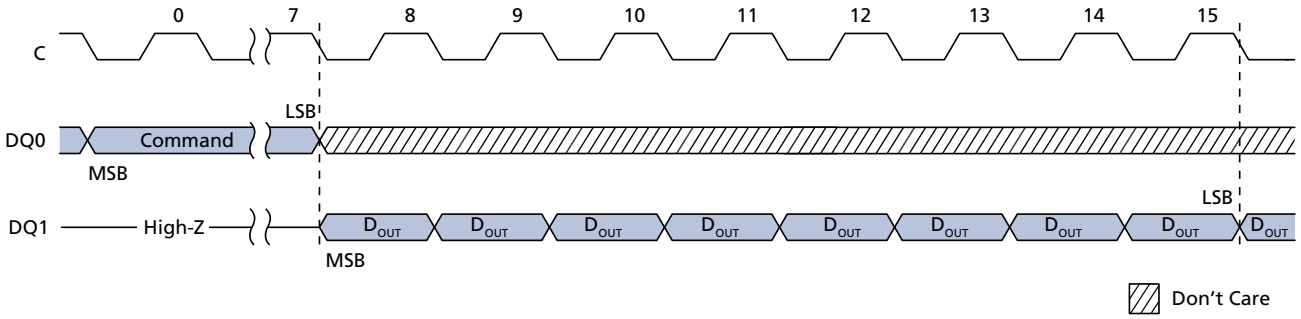
Figure 8: READ IDENTIFICATION Command Sequence



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

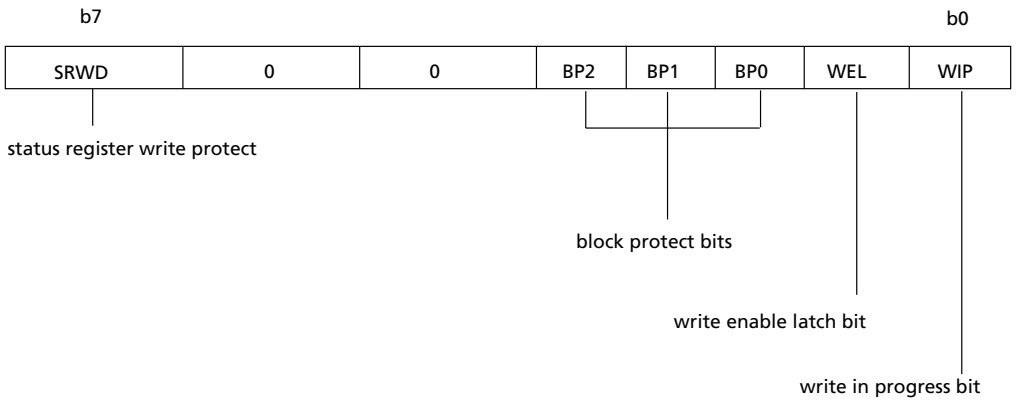
The block protect (BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against PROGRAM and ERASE commands. The block protect bits are written with the WRITE STATUS REGISTER command.

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.

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



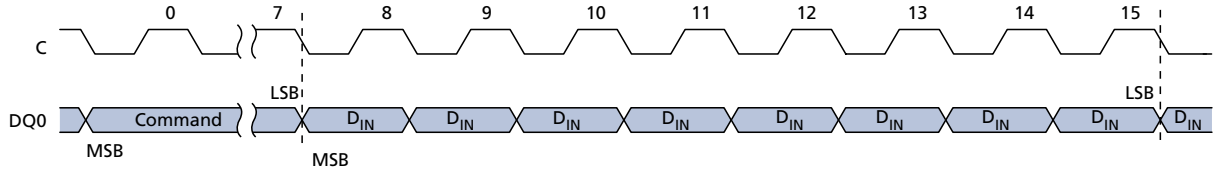
Note: WEL and WIP are volatile read-only bits (WEL is set and reset by specific instructions; WIP is automatically set and rest by the internal logic of the device). SRWD = status register write protect bit; BP0, BP1, BP2 = block protect bits.

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 11: 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 9: 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.

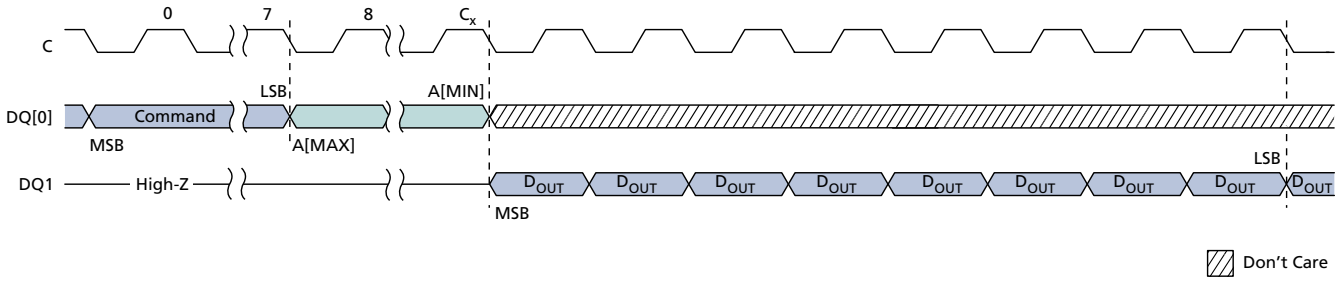
READ DATA BYTES

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 12: READ DATA BYTES Command Sequence



Note: 1. Address bits A23-A20 are *don't care*.

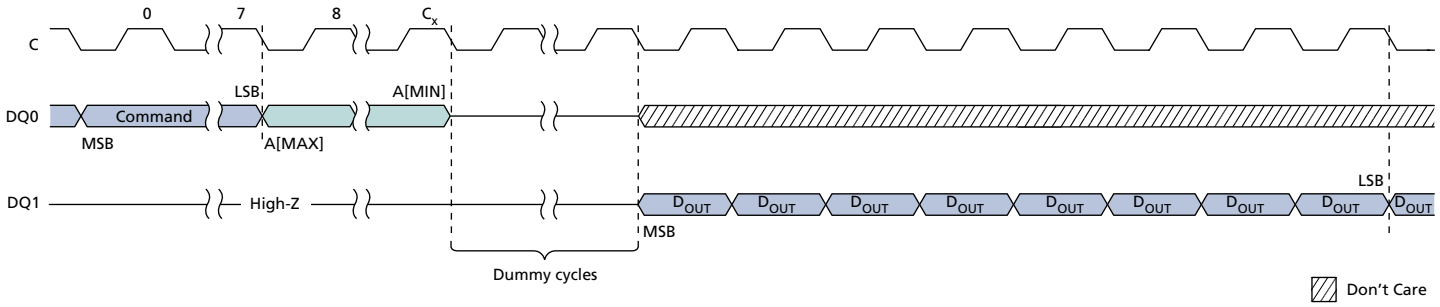
READ DATA BYTES at HIGHER SPEED

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 13: READ DATA BYTES at HIGHER SPEED Command Sequence



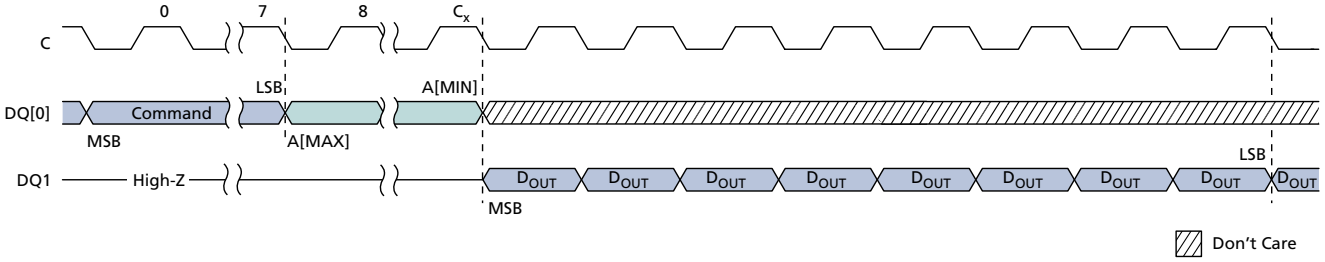
Note: 1. Address bits A23-A20 are *don't care* .

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 14: 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 10: Lock Register Out

Bit	Bit name	Value	Function
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 11: Not for new design: Lock Registers for the M25PE80 in T7Y Process

Bit	Bit name	Value	Function
b7-b4	Reserved		

Table 11: Not for new design: Lock Registers for the M25PE80 in T7Y Process (Continued)

Bit	Bit name	Value	Function
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.

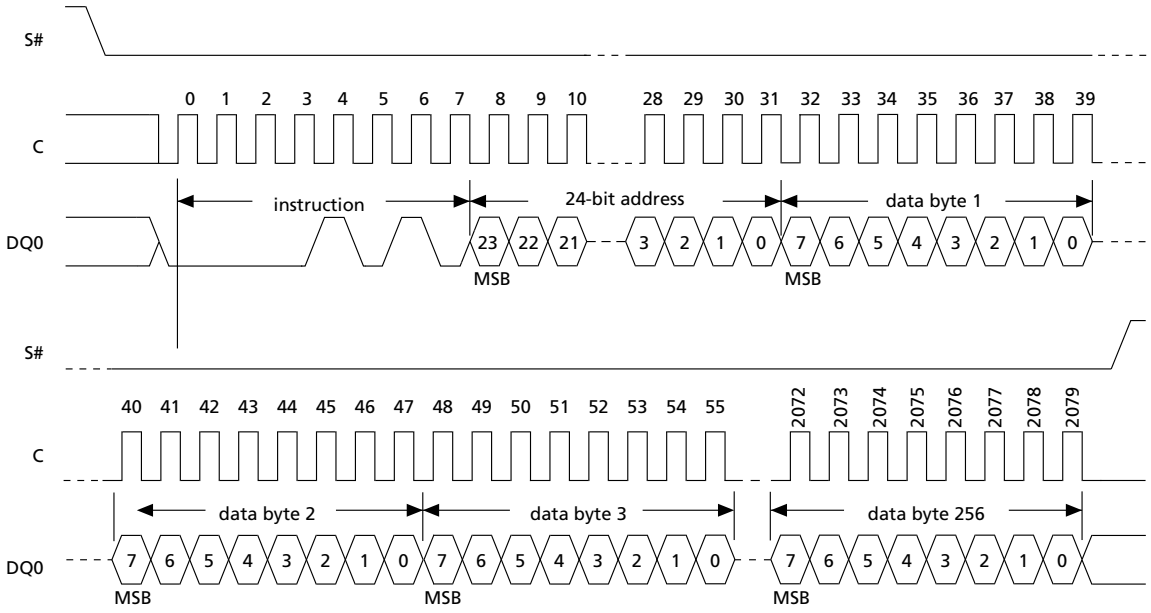
As soon as S# is driven HIGH, the self-timed PAGE WRITE cycle is initiated. While the PAGE WRITE cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed PAGE WRITE cycle, and 0 when the cycle is completed. At 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.

Figure 15: PAGE WRITE Command Sequence



Note:

Address bits A23-A20 are don't care. 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, three address bytes, and at least one data byte on serial data input (DQ0).

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

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

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

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

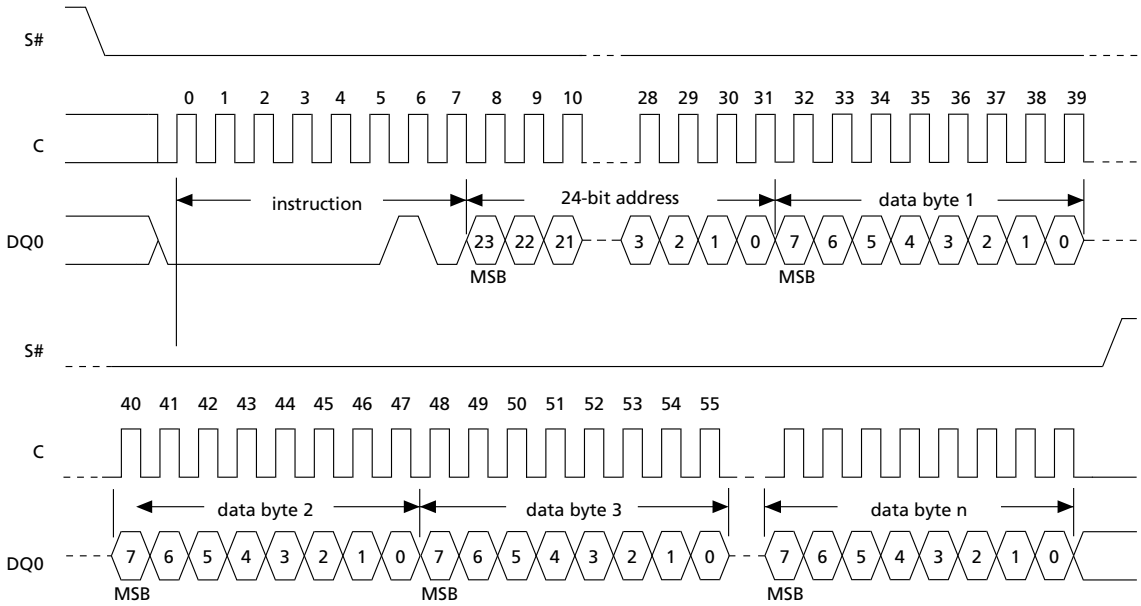
As soon as S# is driven HIGH, the self-timed PAGE PROGRAM cycle is initiated; the cycles's duration is t_{PP}. While the PAGE PROGRAM cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The WIP bit is 1 during the self-timed PAGE PROGRAM cycle,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

A PAGE PROGRAM command 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.

Figure 16: PAGE PROGRAM Command Sequence



Note: 1. Address bits A23-A20 are don't care. 1≤n≤ 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 17: WRITE to LOCK REGISTER Instruction Sequence

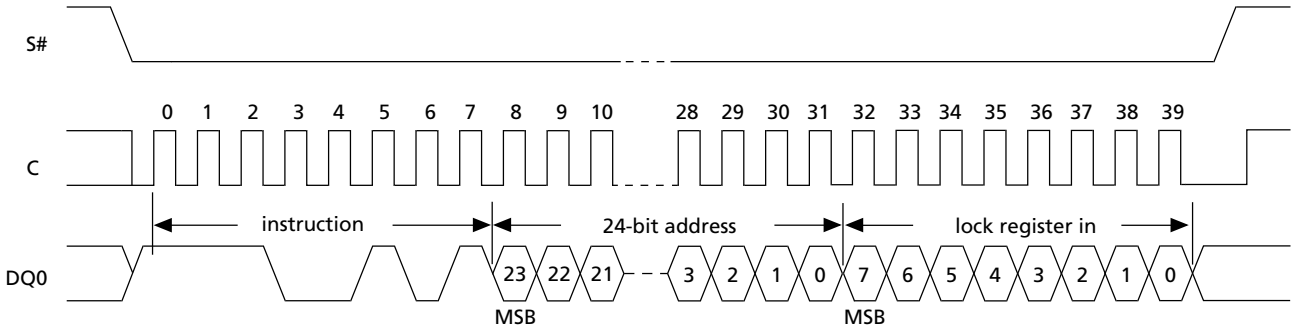


Table 12: Lock Register In

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 13: 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

Table 13: Not for new design: lock registers for the M25PE80 in T7Y process (Continued)

Sector	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.

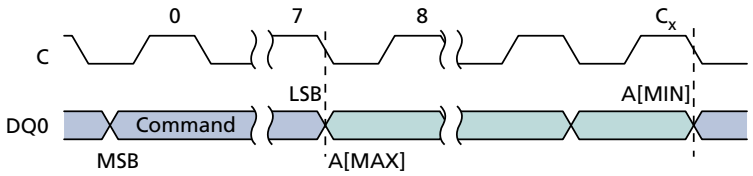
S# must be driven HIGH after the eighth bit of the last address byte has been latched in. Otherwise the PAGE ERASE command is not executed. As soon as S# is driven HIGH, the self-timed PAGE ERASE cycle is initiated; the cycle's duration is t_{PE}. While the PAGE ERASE cycle is in progress, the status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

Figure 18: PAGE ERASE Command Sequence



Note: 1. 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.

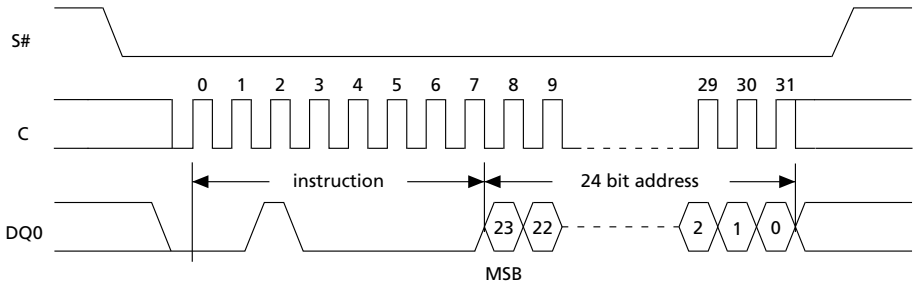
S# must be driven HIGH after the eighth bit of the last address byte has been latched in. Otherwise the SUBSECTOR ERASE command is not executed. As soon as S# is driven HIGH, the self-timed SUBSECTOR ERASE cycle is initiated; the cycle's duration is t_{SSE}. While the SUBSECTOR ERASE cycle is in progress, 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 19: SUBSECTOR ERASE Command Sequence



Note: 1. 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.

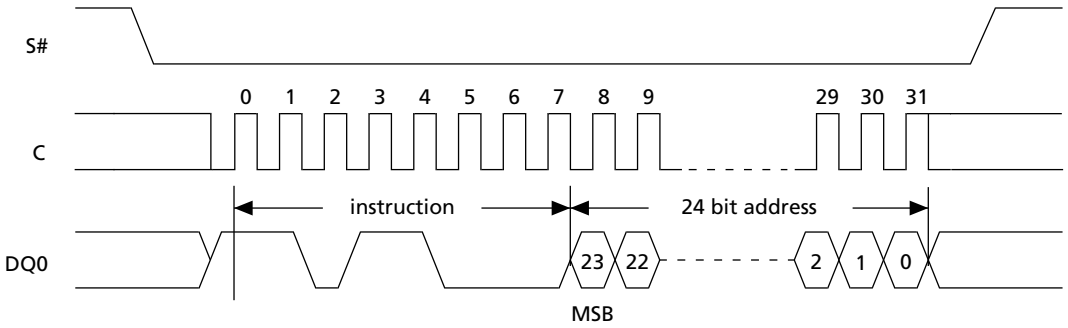
S# must be driven HIGH after the eighth bit of the last address byte has been latched in. Otherwise the SECTOR ERASE command is not executed. As soon as S# is driven HIGH, the self-timed SECTOR ERASE cycle is initiated; the cycle's duration is t_{SE}. While the SECTOR ERASE cycle is in progress, the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

Figure 20: SECTOR ERASE Command Sequence



Note: 1. 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.

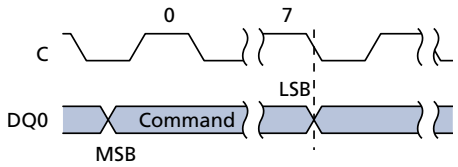
The BULK ERASE command is entered by driving chip select (S#) LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

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 21: 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}.

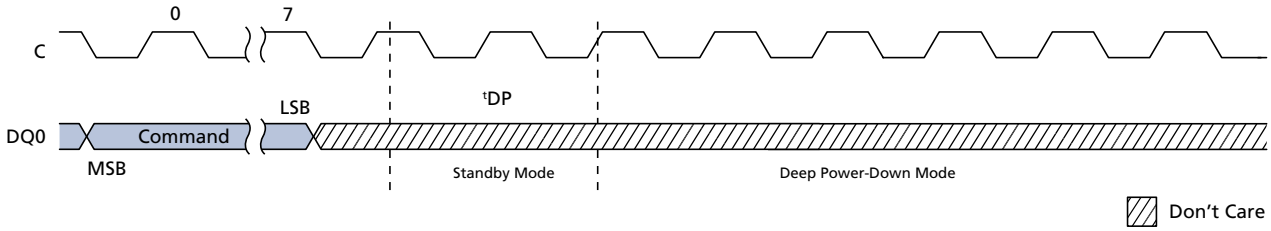
To take the device out of DEEP POWER-DOWN mode, the RELEASE from DEEP POWER-DOWN command must be issued. Other commands must not be issued while the device is in DEEP POWER-DOWN mode. The DEEP POWER-DOWN mode stops automatically at power-down. The device always powers up in STANDBY POWER mode.

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 22: DEEP POWER-DOWN Command Sequence



RELEASE from DEEP POWER-DOWN

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

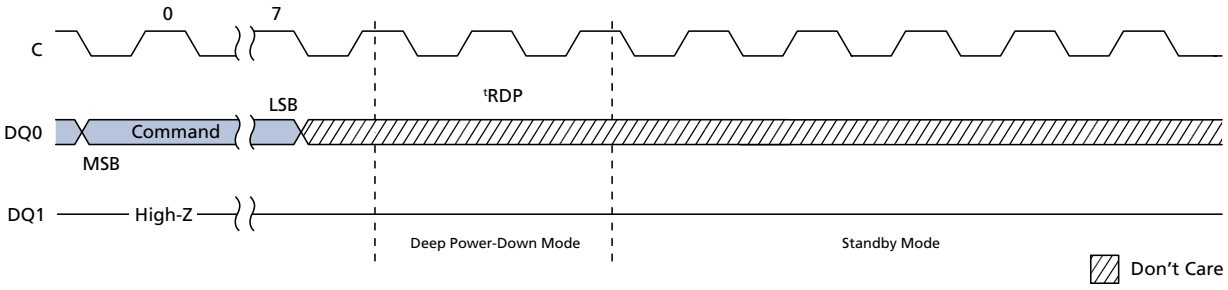
The RELEASE from DEEP POWER-DOWN command is entered by driving chip select (S#) LOW, followed by the command code on serial data input (DQ0). S# must be driven LOW for the entire duration of the sequence.

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

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

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

Figure 23: 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}(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}(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}(min) level

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

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

Figure 24: Power-Up Timing

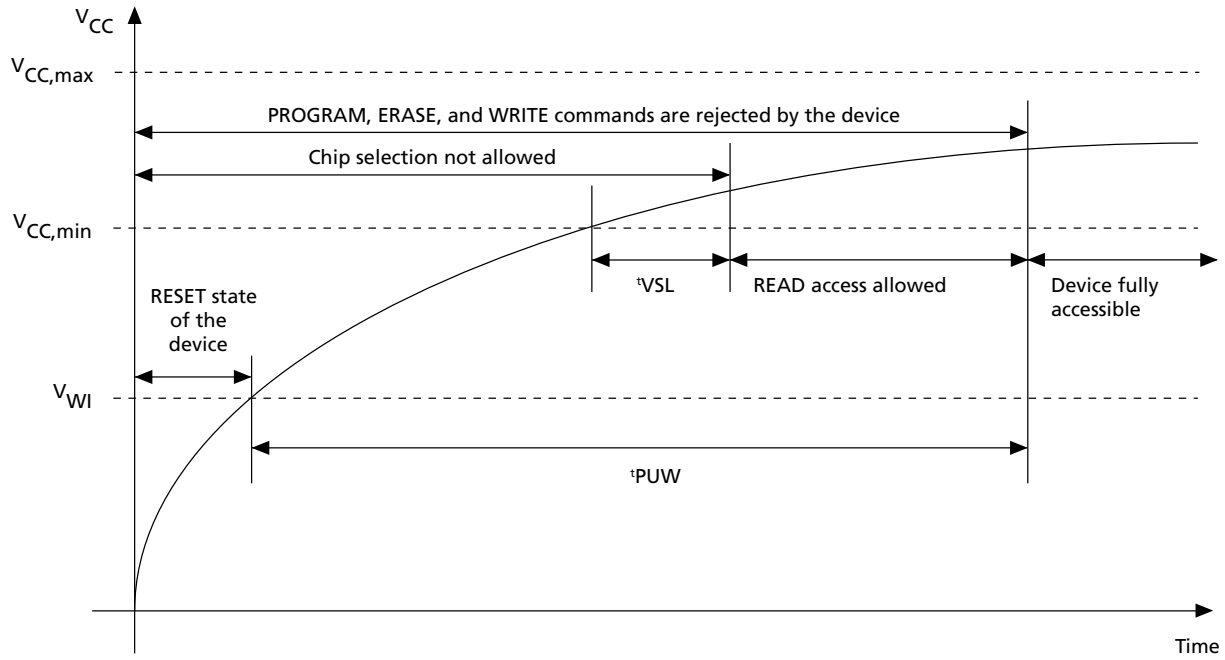


Table 14: Power-up Timing and V_{WI} Threshold

Note: These parameters are characterized only, over the temperature range -40 °C to +85 °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 15: Device Status After a RESET# LOW Pulse

Conditions: RESET Pulse Occurred	Lock Bits Status	Internal Logic Status	Addressed Data
While decoding an instruction: WREN, WRDI, RDID, RDSR, READ, RDLR, FAST_READ, WRLR, PW, PP, PE, SE, BE, SSE, DP, RDP (While decoding an instruction, S# remains LOW while RESET# is LOW.)	Reset to 0	Same as POR	Not significant
Under completion of an ERASE or PROGRAM cycle of a PW, PP, PE, SSE, SE, or BE operation	Reset to 0	Equivalent to POR	Addressed data could be modified
Under completion of a WRSR operation	Reset to 0	Equivalent to POR (after t_W)	Write is correctly completed
Device deselected (S# HIGH) and in STANDBY mode	Reset to 0	Same as POR	Not significant

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 16: 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 17: 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 18: DC Characteristics

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} or V_{CC}	–	50	µA
I_{CC2}	Deep power-down current	S# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	–	10	µA
I_{CC3}	Operating current (FAST_READ)	C = 0.1V_{CC} / 0.9V_{CC} at 50 MHz, DQ1 = open	–	8	mA
		C = 0.1V_{CC} / 0.9V_{CC} at 33 MHz, DQ1 = open	–	6	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 µA	V_{CC}-0.2		V

Table 19: DC Characteristics 75 MHz Operation

Symbol	Parameter	Test Conditons (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} or V_{CC}	–	50	µA
I_{CC2}	Deep power-down current	S# = V_{CC}, V_{IN} = V_{SS} 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

Table 19: DC Characteristics 75 MHz Operation (Continued)

Symbol	Parameter	Test Conditons (in addition to those listed in Operating Conditions table)	Min	Max	Units
V_{OL}	Output Low Voltage	I_{OL} = 1.6mA		0.4	V
V_{OH}	Output High Voltage	I_{OH} = -100 µA	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 20: 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 25: AC Measurement I/O Waveform

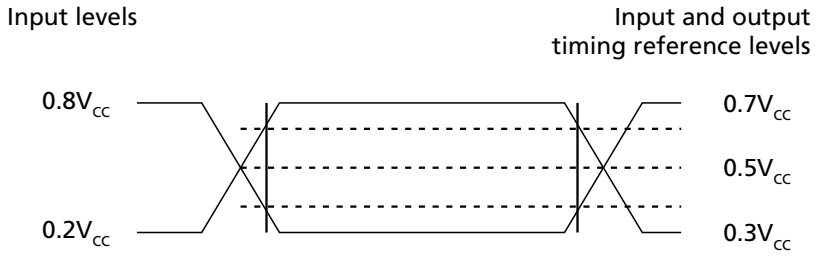


Table 21: Capacitance

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°C and a frequency of 33MHz.

Table 22: AC Characteristics

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, PW, PP, PE, SE, DP, RDP, WREN, WRDI, RDSR	D.C.	–	50	MHz	
f_R	–	Clock frequency for READ command	D.C.	–	20	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	2
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	ns	
t_{CLQX}	t_{HO}	Output hold time	0	–	–	ns	
t_{THSL}	–	TOP SECTOR LOCK setup time	50	–	–	ns	
t_{TSHTL}	–	TOP SECTOR LOCK hold time	100	–	–	ns	
t_{DP}	–	S# to DEEP POWER-DOWN mode	–	–	3	µs	2
t_{RDP}	–	S# HIGH to STANDBY mode	–	–	30	µs	2
t_{PW}	–	PAGE WRITE cycle time (256 bytes)	–	11	25	ms	3
t_{PW}	–	PAGE WRITE cycle time (n bytes)		10.1+n*0.9/256	25	ms	3
t_{PP}	–	PAGE PROGRAM cycle time (256 bytes)	–	1.35	3	ms	3
t_{PP}	–	PAGE PROGRAM cycle time (n bytes)	–	0.45+n*0.9/256	5	ms	3
t_{PE}	–	PAGE ERASE cycle time	–	10	20	ms	
t_{SE}	–	SECTOR ERASE cycle time	–	1	5	s	
t_{SE}	–	BULK ERASE cycle time	–	10	60	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. 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 ($1 \leq n \leq 256$).

Table 23: AC Characteristics (50MHz 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	1
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	ns	
t_{CLQX}	t_{HO}	Output hold time	0	–	–	ns	
t_{THSL}	–	WRITE PROTECT setup time	50	–	–	ns	3
t_{TSHTL}	–	WRITE PROTECT hold time	100	–	–	ns	3
t_{DP}	–	S# to DEEP POWER-DOWN mode	–	–	3	µs	2
t_{RDP}	–	S# HIGH to STANDBY mode	–	–	30	µs	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, 5
		PAGE PROGRAM cycle time (n bytes)		int(n/8) x 0.025			
t_{PE}	–	PAGE ERASE cycle time	–	10	20	ms	
t_{SE}	–	SECTOR ERASE cycle time	–	1	5	s	
t_{SSE}	–	SUBSECTOR ERASE cycle time	–	50	150	ms	
t_{BE}	–	BULK ERASE cycle time	–	10	20	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 ≤256).
5. int(A) corresponds to the upper integer part of A. For instance, int(12/8) = 2, int(32/8) = 4 int(15.3) = 16.

Table 24: AC Specifications (75 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.	–	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	1
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	20	–	–	ns	3
t_{SHWL}	–	WRITE PROTECT hold time	100	–	–	ns	3
t_{DP}	–	S# to DEEP POWER-DOWN mode	–	–	3	µs	2
t_{RDP}	–	S# HIGH to STANDBY mode	–	–	30	µs	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)	–	int(n/8) x 0.025	3	ms	4, 5
t_{PE}	–	PAGE ERASE cycle time	–	10	20	ms	
t_{SE}	–	SECTOR ERASE cycle time	–	1	5	s	
t_{SSE}	–	SUBSECTOR ERASE cycle time	–	50	150	ms	
t_{BE}	–	BULK ERASE cycle time	–	10	20	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. int(A) corresponds to the upper integer part of A. For example, int(12/8) = 2, int(32/8) = 4, int(15.3) = 16.

Figure 26: Serial Input Timing

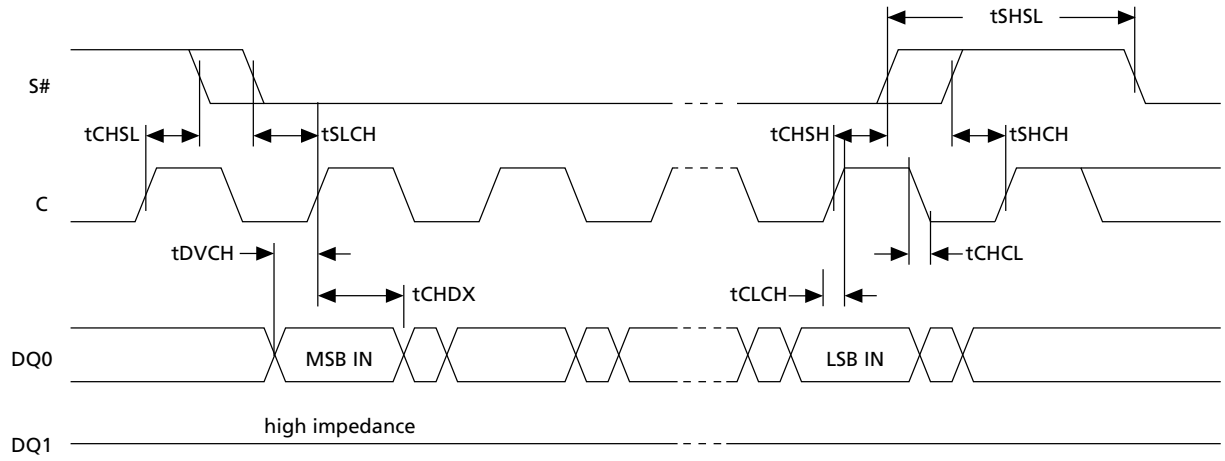


Figure 27: Write Protect Setup and Hold Timing

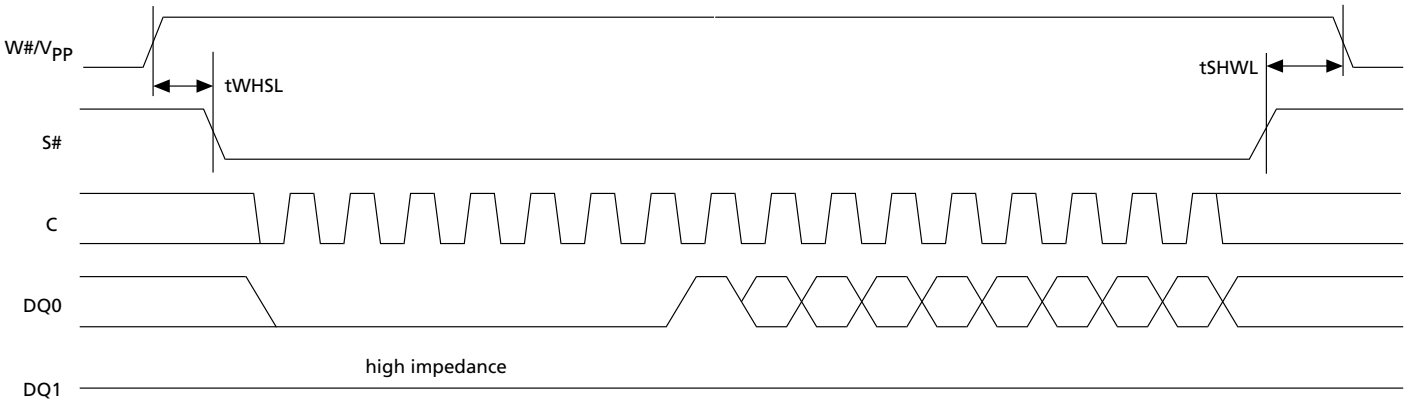


Figure 28: Output Timing

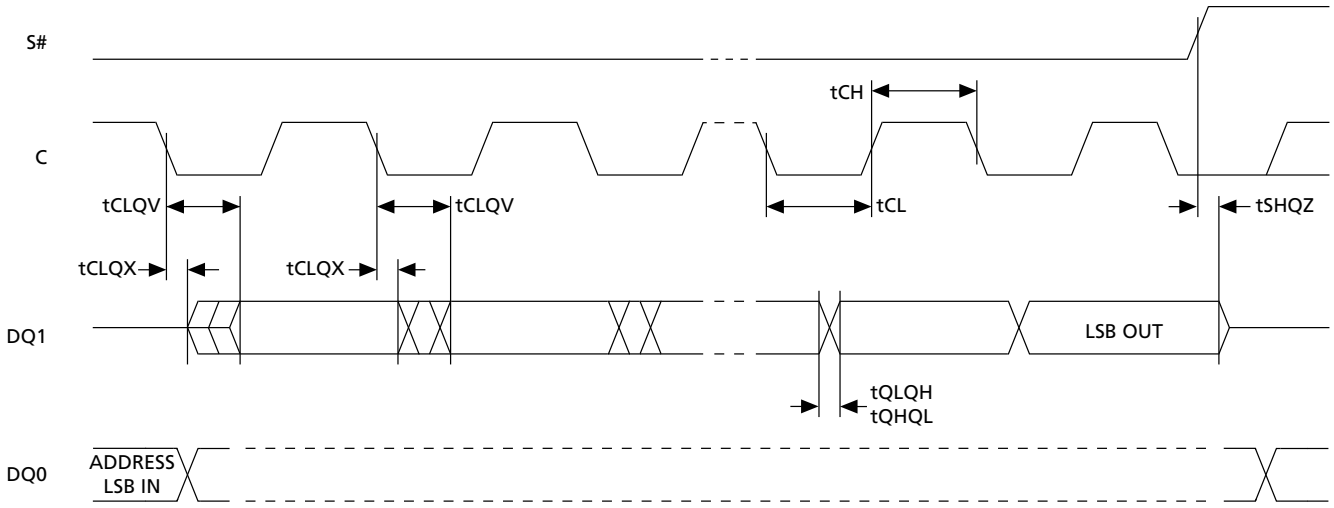


Table 25: Reset Conditions

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

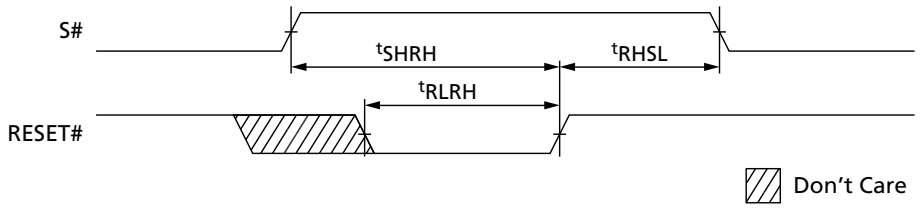
Note: 1. Test conditions specified in Operating Conditions and AC Measurement Conditions tables. t_{RLRH} value guaranteed by characterization; not 100% tested in production.

Table 26: Timings After a RESET# LOW Pulse

Symbol	Alt	Parameter	Conditions: Reset Pulse Occurred	Max	Unit	
$^{t}RHSL$	^{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	

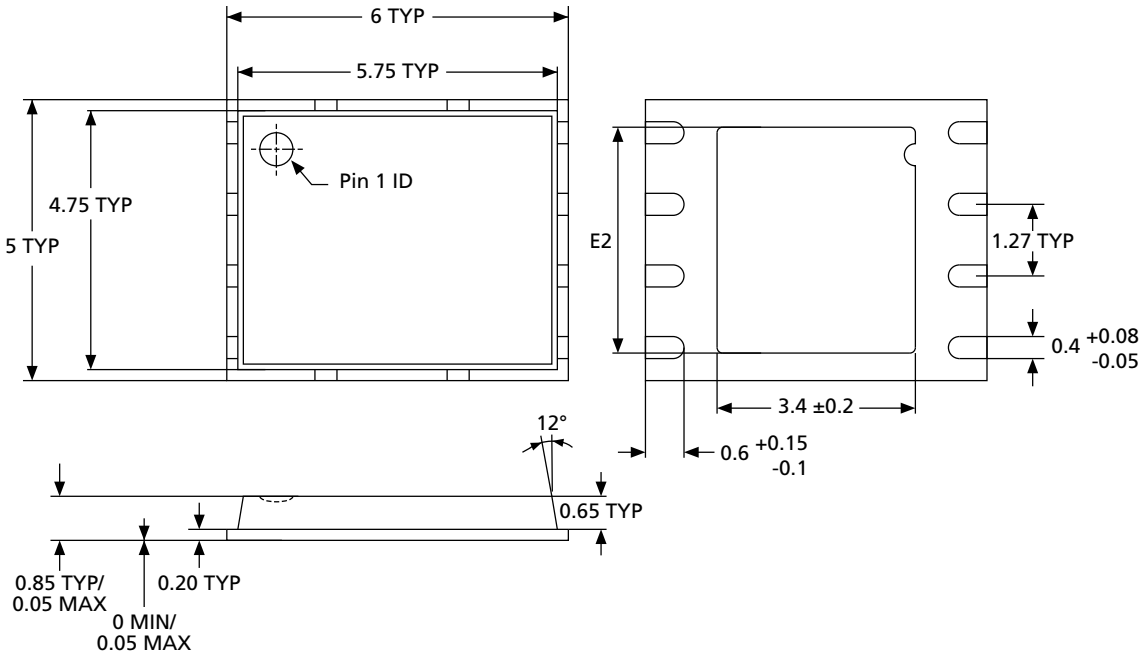
Notes: 1. All values are guaranteed by characterization; not 100% tested in production. See Device Status after a RESET# LOW Pulse table for a description of the device status after a RESET# LOW pulse. While decoding an instruction, S# remains LOW while RESET# is LOW.
2. See AC Characteristics 75 MHz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

Figure 29: Reset AC Waveforms



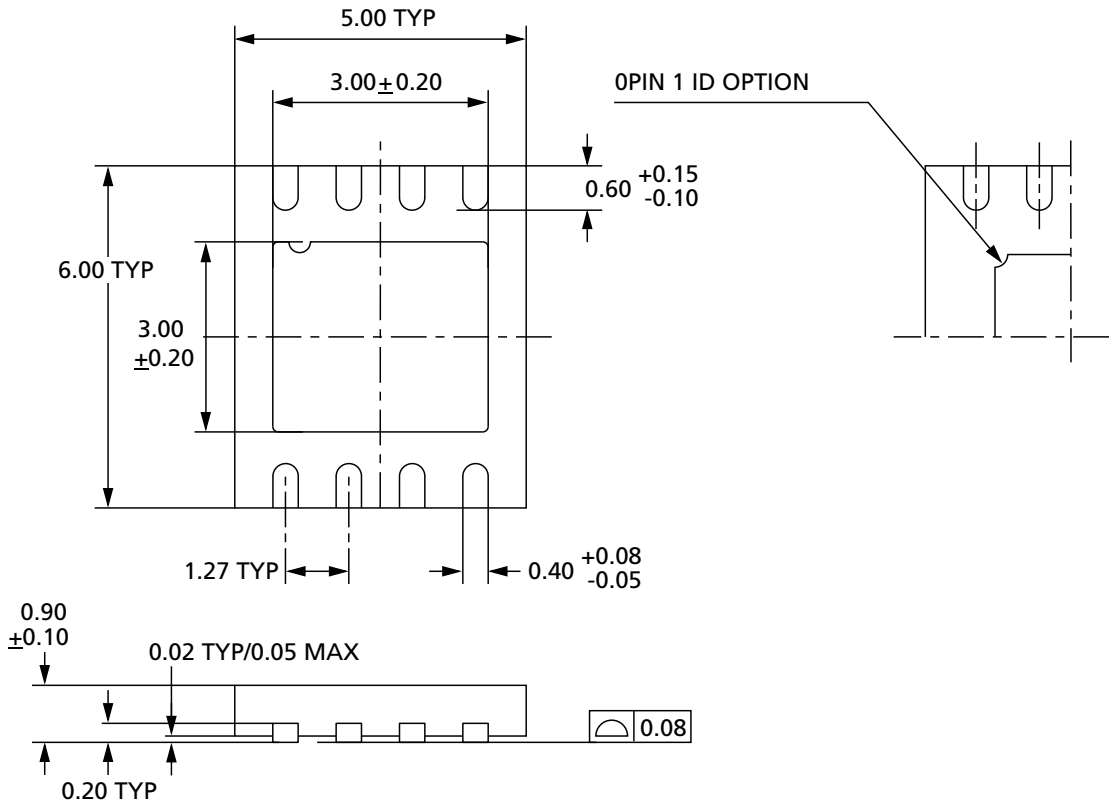
Package Information

Figure 30: VFQFPN8 (MLP8) 6mm x 5mm



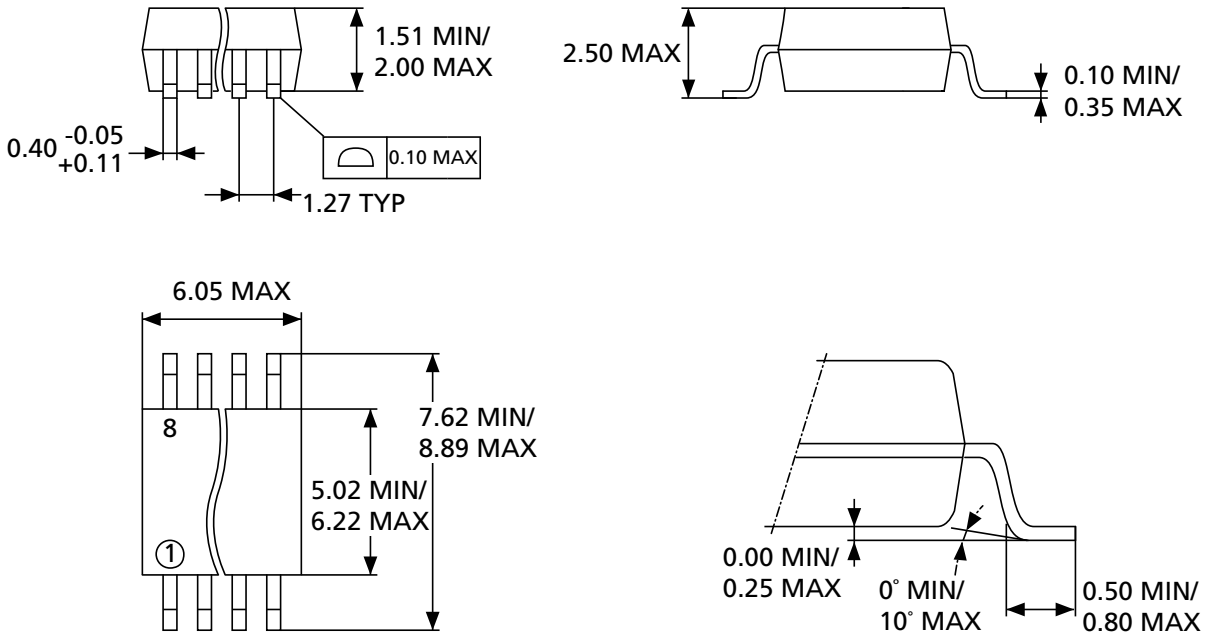
Note: 1. Drawing is not to scale.

Figure 31: QFN8L (MLP8) 8-lead, quad flat package no lead, 6mm x 5mm



Note: 1. Drawing is not to scale.

Figure 32: SO8W – 8 lead plastic small outline, 208 mils body width



Note: 1. Drawing is not to scale.

Device Ordering Information

For a list of available options (speed, packaging, and so on), for further information on any aspect of this device, or when ordering parts operating at 75 MHz, contact your nearest representative.

Table 27: Standard Part Number Information Scheme

Part Number Category	Category Details	Notes
Device type	M25PE = Page-erasable serial Flash memory	
Density	80 = 8Mb (1Mb x 8)	
Operating voltage	V = V_{CC} = 2.7V to 3.6V	
Package	MW = SO8W (208 mils width)	
	MN = SO8N (150 mils width)	
	MP = VFQFPN8 6mm x 5mm (MLP8)	
	MS = QFN8L 6mm x 5mm (MLP8)	1
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	

Note: 1. Exposed pad of 3mm x 3mm.

Table 28: Automotive Part Number Information Scheme

Part Number Category	Category Details
Device type	M25PE = Page-erasable serial Flash memory
Density	80 = 8Mb (1Mb x 8)
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
	blank = Industrial part (Refer to Standard Part Number Information Scheme)
Automotive grade	A = Automotive grade –40 to 85 °C. Device tested with high reliability certified flow
	blank = Standard –40 to 85 °C device (Refer to Standard Part Number Information Scheme)

Revision History

Rev. B – 11/2012

- Removed T9HX process information
- Added automotive grade part information

Rev. A – 09/2012

- Applied Micron branding.