Micron Serial NOR Flash Memory

3V, Multiple I/O, 4KB Sector Erase

N25Q064A13Exx4My

Features

- SPI-compatible serial bus interface
- 108 MHz (MAX) clock frequency
- 2.7–3.6V single supply voltage
- Dual/quad I/O instruction provides increased throughput up to 432 MHz
- Commands
  - Fast read
  - Quad or dual output fast read
  - Quad or dual I/O fast read
- 64-byte, user-lockable, one-time programmable (OTP) dedicated area
- Erase capability
  - Subsector erase 4KB uniform granularity blocks
  - Sector erase 64KB uniform granularity blocks
  - Subsector erase 32KB uniform granularity blocks
  - Full-chip erase
  - Program/erase suspend operations
- This device is compliant with Intel Serial Flash Hardening Product External Architecture Specification
- Write protection
  - Software write protection applicable to every 64KB sector via volatile lock bit
  - Hardware write protection: protected area size defined by five nonvolatile bits (BP0, BP1, BP2, BP3, and TB)
- Electronic signature
  - JEDEC-standard 2-byte signature (BA17h)
  - Unique ID code (UID): 17 read-only bytes, including:
    - Two additional extended device ID (EDID) bytes to identify device factory options
    - Customized factory data (14 bytes)
- Minimum 100,000 ERASE cycles per sector
- More than 20 years data retention
- Packages JEDEC standard, all RoHS compliant
  - W7 = W-PDFN-8 6mm x 5mm (MLP8 6mm x 5mm)
  - SF = SOP2-16 300 mils body width (SO16W)
  - SE = SOP2-8 208 mils body width (SO8W)

Contents

List of Figures

List of Tables

Device Description

The N25Q family is the first high-performance multiple input/output serial Flash memory manufactured on 65nm NOR technology. It features advanced write protection mechanisms, and a high-speed SPI-compatible bus interface. The innovative, high-performance, dual and quad input/output instructions enable double or quadruple the transfer bandwidth for READ and PROGRAM operations. This N25Q device is specifically optimized for PC application usage models and is compliant to Intel SPI Flash Hardening Product SAS Rev. 0.63+ specifications.

Features

The device is organized as 128 (64KB) main sectors. These are divided into 256 (32KB) subsectors, which are further divided into 8 subsectors each, for a total of 2048 subsectors. An ERASE operation can be done one 4KB subsector at a time, one 32KB subsector at a time, one 64KB sector at a time, or as an entire device.

The device can be write protected by software through volatile and nonvolatile protection features, depending on the application needs. The protection granularity is of 64KB (sector granularity) for volatile protections

The device has 64 one-time programmable (OTP) bytes that can be read and programmed with the READ OTP and PROGRAM OTP commands. These 64 bytes can also be permanently locked with a PROGRAM OTP command.

The device can also pause and resume PROGRAM and ERASE cycles using PROGRAM/ERASE SUSPEND and RESUME commands.

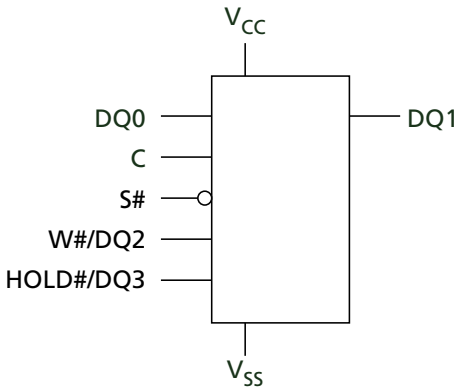
Operating Protocols

The device can be operated with extended SPI protocol, enabling the usage of the following:

- Single I/O, up to 108 MHz
- Dual I/O, up to 108 MHz
- Quad I/O, up to 108 MHz

Dual and quad operations improve the data access time and throughput of a single I/O device by transmitting commands, addresses, and data across two or four data lines.

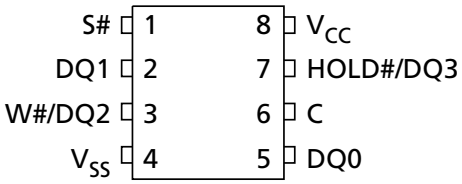
Figure 1: Logic Diagram



Note: 1. Reset functionality is available in devices with a dedicated part number. See Part Number Ordering Information for more details.

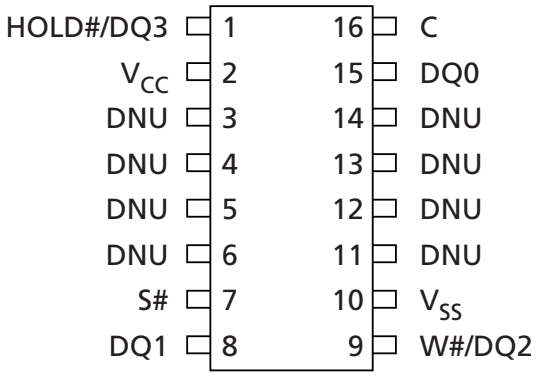
Signal Assignments

Figure 2: 8-Pin, VDFPN8 – MLP8 and SOP2 – SO8W (Top View)



Note: 1. Reset functionality is available in devices with a dedicated part number. See Part Number Ordering Information for complete package names and details.

Figure 3: 16-Pin, Plastic Small Outline – SO16 (Top View)



Note: 1. Reset functionality is available in devices with a dedicated part number. See Part Number Ordering Information for complete package names and details.

Signal Descriptions

Table 1: Signal Descriptions

Symbol	Type	Description
C	Input	**Clock:** Provides the timing of the serial interface. Commands, addresses, or data present at serial data inputs are latched on the rising edge of the clock. Data is shifted out on the falling edge of the clock.
S#	Input	**Chip select:** When S# is HIGH, the device is deselected and DQ1 is at High-Z. When in extended SPI mode, with the device deselected, DQ1 is tri-stated. Unless an internal PROGRAM, ERASE, or WRITE STATUS REGISTER cycle is in progress, the device enters 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.
DQ0	Input and I/O	**Serial data:** Transfers data serially into the device. It receives command codes, addresses, and the data to be programmed. Values are latched on the rising edge of the clock. DQ0 is used for input/output during the following operations: DUAL OUTPUT FAST READ, QUAD OUTPUT FAST READ, DUAL INPUT/OUTPUT FAST READ, and QUAD INPUT/OUTPUT FAST READ. When used for output, data is shifted out on the falling edge of the clock.
DQ1	Output and I/O	**Serial data:**Transfers data serially out of the device. Data is shifted out on the falling edge of the clock. DQ1 is used for input/output during the following operations: DUAL INPUT FAST PROGRAM, QUAD INPUT FAST PROGRAM, DUAL INPUT EXTENDED FAST PROGRAM, and QUAD INPUT EXTENDED FAST PROGRAM. When used for input, data is latched on the rising edge of the clock.
DQ2	Input and I/O	**DQ2:**When using QUAD FAST READ commands, the signal functions as DQ2, providing input/output. All data input drivers are always enabled except when used as an output. Micron recommends customers drive the data signals normally (to avoid unnecessary switching current) and float the signals before the memory device drives data on them.
DQ3	Input and I/O	**DQ3:** When using quad FAST READ commands, the signal functions as DQ3, providing input/output. HOLD# is disabled if the device is selected.
HOLD#	Control Input	**HOLD:** Pauses any serial communications with the device without deselecting the device. DQ1 (output) is High-Z. DQ0 (input) and the clock are "Don't Care." To enable HOLD, the device must be selected with S# driven LOW. HOLD# is used for input/output during the following operations: QUAD OUTPUT FAST READ, QUAD INPUT/OUTPUT FAST READ, QUAD INPUT FAST PROGRAM, and QUAD INPUT EXTENDED FAST PROGRAM. In QUAD FAST READ operation, HOLD# acts as an I/O (DQ3 functionality), and the HOLD# functionality is disabled when the device is selected and the command code has been sent. During address and data reading, HOLD# acts as an I/O (DQ3 functionality). When the device is deselected (S# is HIGH), HOLD# functionality is restored.
W#	Control Input	**Write protect:** W# can be used as a protection control input or in quad operations. In single or dual commands, the WRITE PROTECT function is selectable by the voltage range applied to the signal. If voltage range is low (0V to V_{CC}), the signal acts as a write protection control input. The memory size protected against PROGRAM or ERASE operations is locked as specified in the status register block protect bits 3:0. W# is used as an input/output (DQ2 functionality) during QUAD INPUT FAST READ and QUAD INPUT/OUTPUT FAST READ operations.
V_{CC}	Power	**Device core power supply:** Source voltage.

Table 1: Signal Descriptions (Continued)

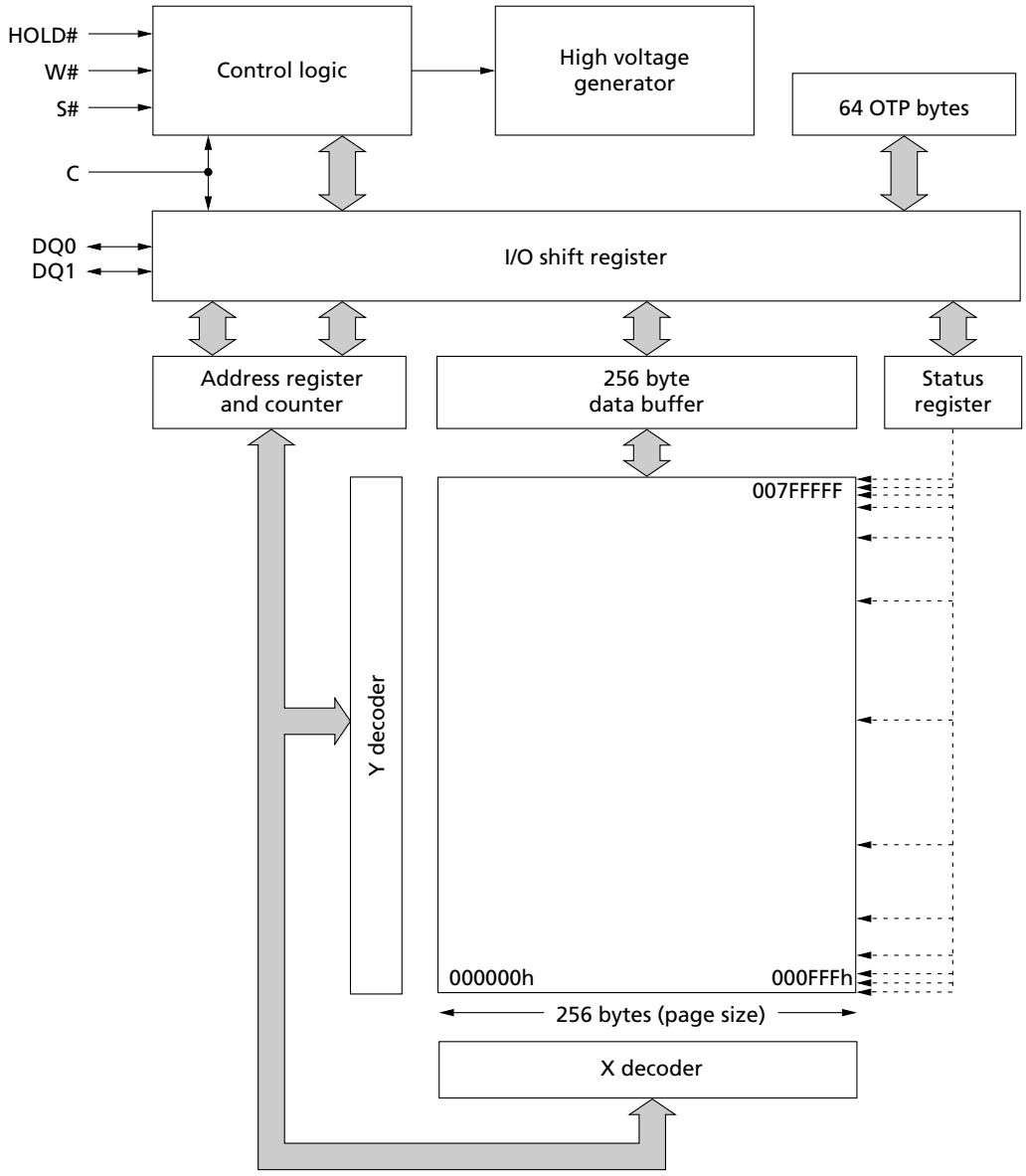
Symbol	Type	Description
V_{SS}	Ground	**Ground:** Reference for the V_{CC} supply voltage.
DNU	–	Do not use.
NC	–	No connect.

Memory Organization

Memory Configuration and Block Diagram

Each page of memory can be individually programmed. Bits are programmed from one through zero. The device is subsector, sector, or bulk-erasable, but not page-erasable. Bits are erased from zero through one. The memory is configured as 8,388,608 bytes (8 bits each); 128 sectors (64KB each); 256 subsectors (32KB each); 2048 subsectors (4KB each);; and 37,768 pages (256 bytes each); and 64 OTP bytes are located outside the main memory array.

Figure 4: Block Diagram



Memory Map – 64Mb Density

Table 2: Sectors[127:0]

64KB Sector	32KB Subsector	4KB Subsector	Address Range	
			Start	End
127	255	2047	007F F000h	007F FFFFh
		⋮	⋮	⋮
		2032	007F 0000h	007F 0FFFh
⋮	⋮	⋮	⋮	⋮
63	125	1023	003F F000h	003F FFFFh
		⋮	⋮	⋮
		1008	003F 0000h	003F 0FFFh
⋮	⋮	⋮	⋮	⋮
0	0	15	0000 F000h	0000 FFFFh
		⋮	⋮	⋮
		0	0000 0000h	0000 0FFFh

Device Protection

Table 3: Data Protection using Device Protocols

Note 1 applies to the entire table

Protection by:	Description
Power-on reset and internal timer	Protects the device against inadvertent data changes while the power supply is outside the operating specification.
Command execution check	Ensures that the number of clock pulses is a multiple of one byte before executing a PROGRAM or ERASE command, or any command that writes to the device registers.
WRITE ENABLE operation	Ensures that commands modifying device data must be preceded by a WRITE ENABLE command, which sets the write enable latch bit in the status register.

Note: 1. Extended, dual, and quad SPI protocol functionality ensures that device data is protected from excessive noise.

Table 4: Memory Sector Protection Truth Table

Note 1 applies to the entire table

Sector Lock Register		
Sector Lock Down Bit	Sector Write Lock Bit	Memory Sector Protection Status
0	0	Sector unprotected from PROGRAM and ERASE operations. Protection status reversible.
0	1	Sector protected from PROGRAM and ERASE operations. Protection status reversible.
1	0	Sector unprotected from PROGRAM and ERASE operations. Protection status not reversible except by power cycle or reset.
1	1	Sector protected from PROGRAM and ERASE operations. Protection status not reversible except by power cycle or reset.

Note: 1. Sector lock register bits are written to when the WRITE LOCK REGISTER command is executed. The command will not execute unless the sector lock down bit is cleared (see the WRITE LOCK REGISTER command).

Table 5: Protected Area Sizes – Upper Area

Note 1 applies to the entire table

Status Register Content					Memory Content	
Top/ Bottom Bit	BP3	BP2	BP1	BP0	Protected Area	Unprotected Area
0	0	0	0	0	None	All sectors
0	0	0	0	1	Upper 128th	Sectors (0 to 126)
0	0	0	1	0	Upper 64th	Sectors (0 to 125)
0	0	0	1	1	Upper 32nd	Sectors (0 to 123)
0	0	1	0	0	Upper 16th	Sectors (0 to119)
0	0	1	0	1	Upper 8th	Sectors (0 to 111)

Table 5: Protected Area Sizes – Upper Area (Continued)

Note 1 applies to the entire table

Status Register Content					Memory Content	
Top/ Bottom Bit	BP3	BP2	BP1	BP0	Protected Area	Unprotected Area
0	0	1	1	0	Upper quarter	Sectors (0 to 95)
0	0	1	1	1	Upper half	Sectors (0 to 63)
0	1	0	0	0	All sectors	None
0	1	0	0	1	All sectors	None
0	1	0	1	0	All sectors	None
0	1	0	1	1	All sectors	None
0	1	1	0	0	All sectors	None
0	1	1	0	1	All sectors	None
0	1	1	1	0	All sectors	None
0	1	1	1	1	All sectors	None

Note: 1. See the Status Register for details on the top/bottom bit and the BP 3:0 bits.

Table 6: Protected Area Sizes – Lower Area

Note 1 applies to the entire table

Status Register Content					Memory Content	
Top/ Bottom Bit	BP3	BP2	BP1	BP0	Protected Area	Unprotected Area
1	0	0	0	0	None	All sectors
1	0	0	0	1	Lower 128th	Sectors (1 to 127)
1	0	0	1	0	Lower 64th	Sectors (2 to 127)
1	0	0	1	1	Lower 32nd	Sectors (4 to 127)
1	0	1	0	0	Lower 16th	Sectors (8 to 127)
1	0	1	0	1	Lower 8th	Sectors (16 to 127)
1	0	1	1	0	Lower quarter	Sectors (32 to 127)
1	0	1	1	1	Lower half	Sectors (64 to 127)
1	1	0	0	0	All sectors	None
1	1	0	0	1	All sectors	None
1	1	0	1	0	All sectors	None
1	1	0	1	1	All sectors	None
1	1	1	0	0	All sectors	None
1	1	1	0	1	All sectors	None
1	1	1	1	0	All sectors	None
1	1	1	1	1	All sectors	None

Note: 1. See the Status Register for details on the top/bottom bit and the BP 3:0 bits.

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 here. The difference between the two modes is the clock polarity when the bus master is in standby mode and not transferring data. Input data is latched in on the rising edge of the clock, and output data is available from the falling edge of the clock.

Table 7: 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 SPI protocols.

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

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

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

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

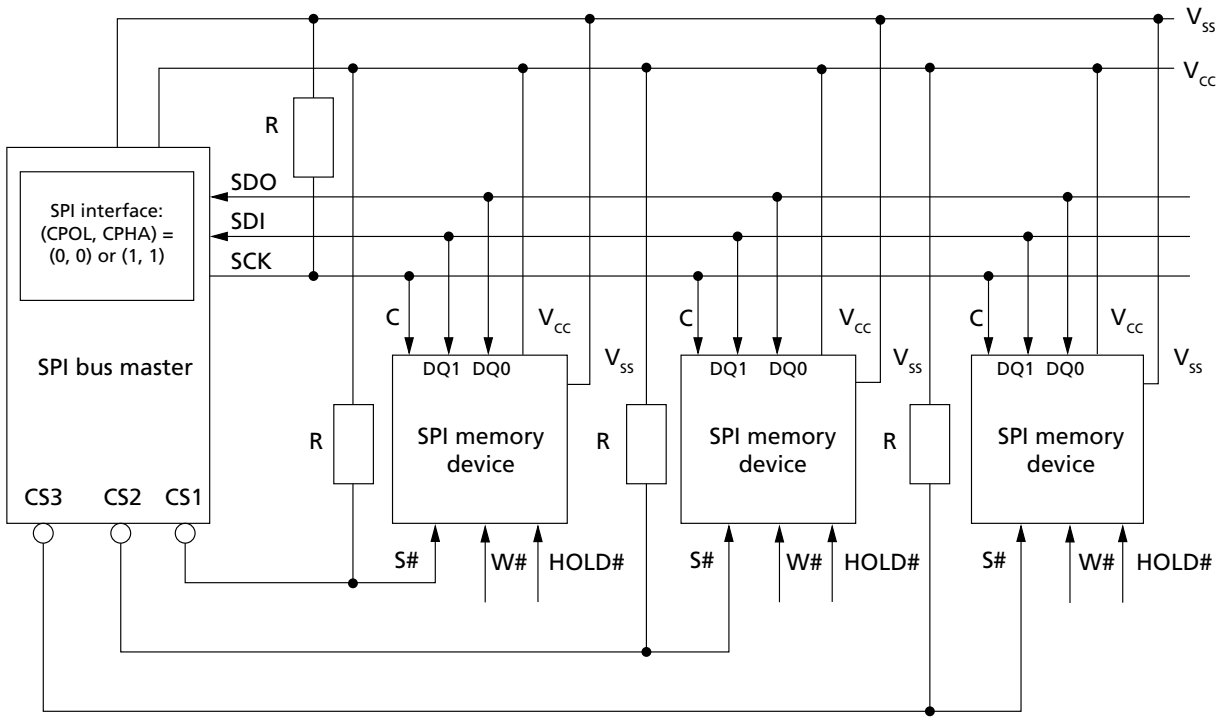
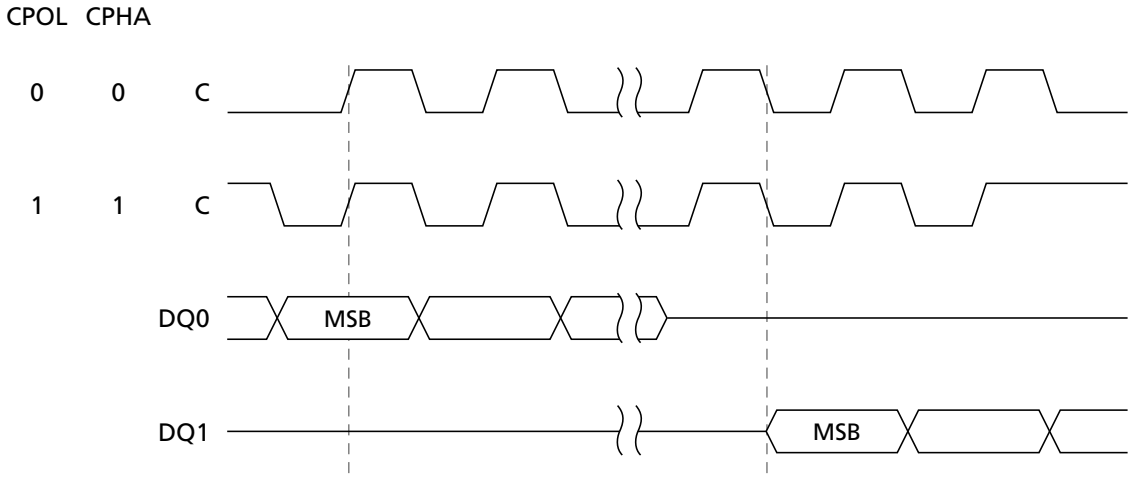


Figure 6: SPI Modes



SPI Protocol

Table 8: SPI Protocols

Protocol Name	Command Input	Address Input	Data Input/Output	Description
Extended	DQ0	Multiple DQ*n* lines, depending on the command	Multiple DQ*n* lines, depending on the command	Device default protocol from the factory. Additional commands extend the standard SPI protocol and enable address or data transmission on multiple DQ*n* lines.

Registers

The device features the following volatile and nonvolatile registers that users can access to store device parameters and operating configurations:

- Status register
- Flag status register
- Lock register

Note: The lock register is defined in READ LOCK REGISTER Command.

Status Register

Table 9: Status Register Bit Definitions

Note 1 applies to entire table

Bit	Name	Settings	Description	Notes
7	Status register write enable/disable	0 = Enabled 1 = Disabled	**Nonvolatile bit:** Used with W# to enable or disable writing to the status register.	3
5	Top/bottom	0 = Top 1 = Bottom	**Nonvolatile bit:** Determines whether the protected memory area defined by the block protect bits starts from the top or bottom of the memory array.	4
6, 4:2	Block protect 3–0	See Protected Area Sizes – Upper Area and Lower Area tables in Device Protection	**Nonvolatile bit:** Defines memory to be software protected against PROGRAM or ERASE operations. When one or more block protect bits is set to 1, a designated memory area is protected from PROGRAM and ERASE operations.	4
1	Write enable latch	0 = Cleared (Default) 1 = Set	**Volatile bit:** The device always powers up with this bit cleared to prevent inadvertent WRITE STATUS REGISTER, PROGRAM, or ERASE operations. To enable these operations, the WRITE ENABLE operation must be executed first to set this bit.	2

Table 9: Status Register Bit Definitions (Continued)

Note 1 applies to entire table

Bit	Name	Settings	Description	Notes
0	Write in progress	0 = Ready 1 = Busy	**Volatile bit:** Indicates if one of the following command cycles is in progress: WRITE STATUS REGISTER PROGRAM ERASE	2

Notes:
1. Bits can be read from or written to using READ STATUS REGISTER or WRITE STATUS REGISTER commands, respectively.
2. Volatile bits are cleared to 0 by a power cycle or reset.
3. The status register write enable/disable bit, combined with the W# signal as described in the Signal Descriptions, provides hardware data protection for the device as follows: When the enable/disable bit is set to 1, and the W# signal is driven LOW, the status register nonvolatile bits become read-only and the WRITE STATUS REGISTER operation will not execute. The only way to exit this hardware-protected mode is to drive W# HIGH.
4. See Protected Area Sizes tables in Device Protection. The BULK ERASE command is executed only if all bits are 0.

Flag Status Register

Table 10: Flag Status Register Bit Definitions

Note 1 applies to entire table

Bit	Name	Settings	Description	Notes
7	Program or erase controller	0 = Busy 1 = Ready	**Status bit:** Indicates whether a PROGRAM, ERASE, WRITE STATUS REGISTER, or WRITE NONVOLATILE CONFIGURATION command cycle is in progress.	2, 3
6	Erase suspend	0 = Not in effect 1 = In effect	**Status bit:** Indicates whether an ERASE operation has been or is going to be suspended.	3
5	Erase	0 = Clear 1 = Failure or protection error	**Error bit:** Indicates whether an ERASE operation has succeeded or failed.	4, 5
4	Program	0 = Clear 1 = Failure or protection error	**Error bit:** Indicates whether a PROGRAM operation has succeeded or failed. Moreover, indicates an attempt to program a 0 to a 1 when $V_{PP} = V_{PPH}$ and the data pattern is a multiple of 64 bits.	4, 5
3	Reserved	Reserved	Reserved	
2	Program suspend	0 = Not in effect 1 = In effect	**Status bit:** Indicates whether a PROGRAM operation has been or is going to be suspended.	3
1	Protection	0 = Clear 1 = Failure or protection error	**Error bit:** Indicates whether an ERASE or a PROGRAM operation has attempted to modify the protected array sector. Moreover, indicates whether a PROGRAM operation has attempted to access the locked OTP space.	4, 5
0	Reserved	Reserved	Reserved	

Notes:
1. Register bits are read by READ FLAG STATUS REGISTER command. All bits are volatile.
2. These program/erase controller settings apply only to PROGRAM or ERASE command cycles in progress; they do not apply to a WRITE command cycle in progress.

3. Status bits are reset automatically.
4. Error bits must be reset by CLEAR FLAG STATUS REGISTER command.
5. Typical errors include operation failures and protection errors caused by issuing a command before the error bit has been reset to 0.

Command Definitions

Table 11: Command Set

Command	Code	Extended	Data Bytes	Notes
IDENTIFICATION Operations				
READ ID	9E/9Fh	Yes	1 to 20	1
READ SERIAL FLASH DISCOVERY PARAMETER	5Ah	Yes	1 to ∞	3
READ Operations				
READ	03h	Yes	1 to ∞	2
FAST READ	0Bh	Yes		3
DUAL OUTPUT FAST READ	3Bh	Yes	1 to ∞	3
DUAL INPUT/OUTPUT FAST READ	BBh	Yes		3
QUAD OUTPUT FAST READ	6Bh	Yes	1 to ∞	3
QUAD INPUT/OUTPUT FAST READ	EBh	Yes		4
WRITE Operations				
WRITE ENABLE	06h	Yes	0	1
WRITE DISABLE	04h			
REGISTER Operations				
READ STATUS REGISTER	05h	Yes	1 to ∞	1
WRITE STATUS REGISTER	01h		1	1, 5
READ FLAG STATUS REGISTER	70h	Yes	1 to ∞	1
CLEAR FLAG STATUS REGISTER	50h		0	
READ LOCK REGISTER	E8h	Yes	1 to ∞	2
WRITE LOCK REGISTER	E5h		1	2, 5
PROGRAM Operations				
PAGE PROGRAM	02h	Yes	1 to 256	2, 5
DUAL INPUT FAST PROGRAM	A2h	Yes	1 to 256	2, 5
EXTENDED DUAL INPUT FAST PROGRAM	D2h	Yes		2, 5
QUAD INPUT FAST PROGRAM	32h	Yes	1 to 256	2, 5
EXTENDED QUAD INPUT FAST PROGRAM	12h	Yes		2, 5
PROGRAM/ERASE SUSPEND	7Ah	Yes	0	1, 5
PROGRAM/ERASE RESUME	75h	Yes	0	1, 5
ERASE Operations				
4KB SUBSECTOR ERASE	20h	Yes	0	2, 5
32KB SUBSECTOR ERASE	52h			2, 5
64KB SECTOR ERASE	D8h	Yes	0	2, 5
BULK ERASE	C7h			1, 5

Table 11: Command Set (Continued)

Command	Code	Extended	Data Bytes	Notes
ONE-TIME PROGRAMMABLE (OTP) Operations				
READ OTP ARRAY	4Bh	Yes	1 to 64	3
PROGRAM OTP ARRAY	42h			2

Notes:
1. Address bytes = 0. Dummy clock cycles = 0.
2. Address bytes = 3. Dummy clock cycles = 0.
3. Address bytes = 3. Dummy clock cycles = 8.
4. Address bytes = 3. Dummy clock cycles = 10.
5. The WRITE ENABLE command must be issued first before this command can be executed.

Table 12: Authenticated Command List

Command	Code	Subcode	Data Bytes	Comments	Notes
WRITE ROOT KEY REGISTER	9Bh	00h	Counter address = 8 bits Reserved = 8 bits Root key = 256 bits Truncated signature = 224 bits	Root key register can be written only once.	2
UPDATE HMAC KEY		01h	Counter address = 8 bits Reserved = 8 bits Key data = 32 bits Truncated signature = 256 bits	Command is executed by host at power-up to initialize HMAC key registers.	2
INCREMENT MONOTONIC COUNTER		02h	Counter address = 8 bits Reserved = 8 bits Counter data = 32 bits Truncated signature = 256 bits	Counter is incremented by 1.	2
REQUEST MONOTONIC COUNTER VALUE		03h	Counter address = 8 bits Reserved = 8 bits Tag = 96 bits Truncated signature = 256 bits	Identical tag must be returned by the READ MONOTONIC COUNTER command.	2
READ MONOTONIC COUNTER	96h	–	Extended status register = 8 bits Tag = 96 bits Counter data = 32 bits Truncated signature = 256 bits	Reads monotonic counter value (from previous REQUEST MONOTONIC COUNTER VALUE operation) and extended status register. Requires internal authentication before being executed.	3

Notes:
1. These commands work within a dedicated interface to manage authenticated operations for RPMC secure feature. The READ MONOTONIC COUNTER command can be called anytime to read operation status and counter values.
2. Payload provided by host (input to memory device).
3. Payload provided by memory (output from memory device). Eight dummy clock cycles are required between the command and the read-out. No address is required

READ REGISTER and WRITE REGISTER Operations

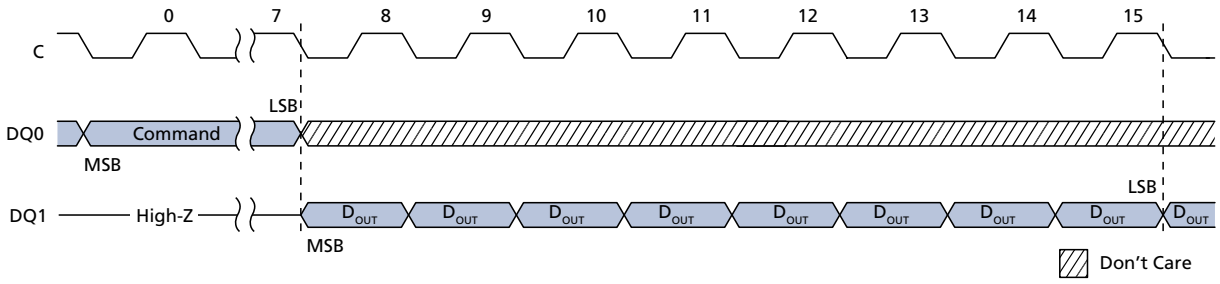
READ STATUS REGISTER or FLAG STATUS REGISTER Command

To initiate a READ STATUS REGISTER command, S# is driven LOW. For extended SPI protocol, the command code is input on DQ0, and output on DQ1. The operation is terminated by driving S# HIGH at any time during data output.

The status register can be read continuously and at any time, including during a PROGRAM, ERASE, or WRITE operation.

If one of these operations is in progress, checking the write in progress bit is recommended before executing the command.

Figure 7: READ REGISTER Command



Notes:
1. Supports all READ REGISTER commands except READ LOCK REGISTER.
2. A READ NONVOLATILE CONFIGURATION REGISTER operation will output data starting from the least significant byte.

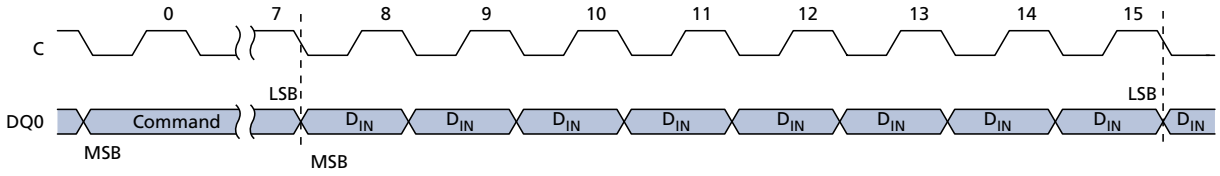
WRITE STATUS REGISTER Command

To issue a WRITE STATUS REGISTER command, the WRITE ENABLE command must be executed to set the write enable latch bit to 1. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. For extended SPI protocol, the command code is input on DQ0, followed by the data bytes. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is ^{t}W.

This command is used to write new values to status register bits 7:2, enabling software data protection. The status register can also be combined with the W# signal to provide hardware data protection. The WRITE STATUS REGISTER command has no effect on status register bits 1:0.

When the operation is in progress, the write in progress bit is set to 1. The write enable latch bit is cleared to 0, whether the operation is successful or not. The status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0, whether the operation is successful or not.

Figure 8: WRITE REGISTER Command



Notes:
1. Supports all WRITE REGISTER commands except WRITE LOCK REGISTER.
2. Waveform must be extended for each protocol, to 23 for extended, 11 for dual, and 5 for quad.
3. A WRITE NONVOLATILE CONFIGURATION REGISTER operation requires data being sent starting from least significant byte.

READ LOCK REGISTER Command

To execute the READ LOCK REGISTER command, S# is driven LOW. For extended SPI protocol, the command code is input on DQ0, followed by three address bytes that point to a location in the sector. Each address bit is latched in during the rising edge of the clock. For extended SPI protocol, data is shifted out on DQ1 at a maximum frequency $^{f}C during the falling edge of the clock. The operation is terminated by driving S# HIGH at any time during data output.

When the register is read continuously, the same byte is output repeatedly. Any READ LOCK REGISTER command that is executed while an ERASE, PROGRAM, or WRITE cycle is in progress is rejected with no affect on the cycle in progress.

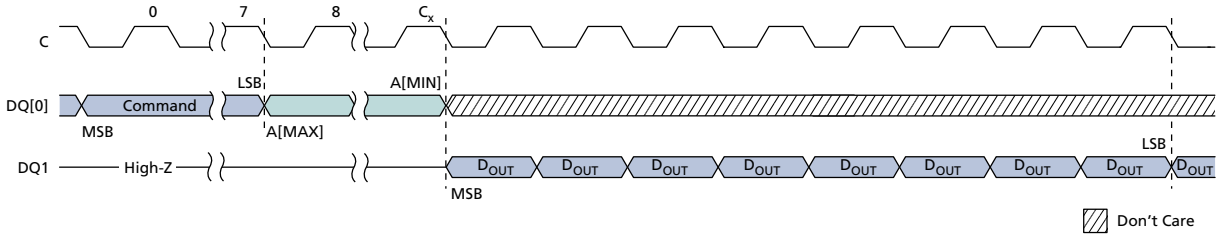
Table 13: Lock Register

Note 1 applies to entire table

Bit	Name	Settings	Description
7:2	Reserved	0	Bit values are 0.
1	Sector lock down	0 = Cleared (Default) 1 = Set	Volatile bit: the device always powers-up with this bit cleared, which means sector lock down and sector write lock bits can be set. When this bit set, neither of the lock register bits can be written to until the next power cycle.
0	Sector write lock	0 = Cleared (Default) 1 = Set	Volatile bit: the device always powers-up with this bit cleared, which means that PROGRAM and ERASE operations in this sector can be executed and sector content modified. When this bit is set, PROGRAM and ERASE operations in this sector will not be executed.

Note:
1. Sector lock register bits 1:0 are written to by the WRITE LOCK REGISTER command. The command will not execute unless the sector lock down bit is cleared.

Figure 9: READ LOCK REGISTER Command



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

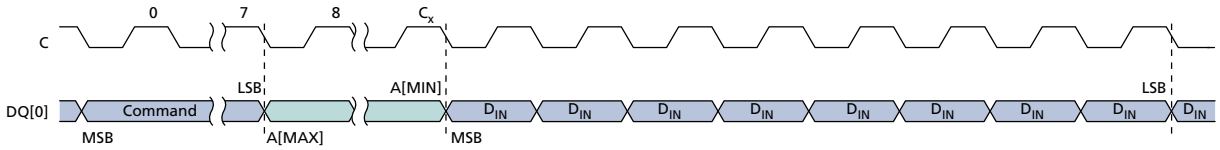
WRITE LOCK REGISTER Command

To initiate the WRITE LOCK REGISTER command, the WRITE ENABLE command must be executed to set the write enable latch bit to 1. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQn, followed by three address bytes that point to a location in the sector, and then one data byte that contains the desired settings for lock register bits 0 and 1. Each address bit is latched in during the rising edge of the clock.

When execution is complete, the write enable latch bit is cleared within ^{t}SHSL2 and no error bits are set. Because lock register bits are volatile, change to the bits is immediate.

WRITE LOCK REGISTER can be executed when an ERASE SUSPEND operation is in effect. If S# is not driven HIGH, the command is not executed, flag status register error bits are not set, and the write enable latch remains set to 1.

Figure 10: WRITE LOCK REGISTER Command



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

CLEAR FLAG STATUS REGISTER Command

To execute the CLEAR FLAG STATUS REGISTER command and reset the error bits (erase, program, and protection), S# is driven LOW. For extended SPI protocol, the command code is input on DQ0. For dual SPI protocol, the command code is input on DQ[1:0]. For quad SPI protocol, the command code is input on DQ[3:0]. The operation is terminated by driving S# HIGH at any time.

READ IDENTIFICATION Operations

READ ID Command

To execute the READ ID command, S# is driven LOW and the command code is input on DQ*n*. The device outputs the information shown in the tables below. If an ERASE or PROGRAM cycle is in progress when the command is executed, the command is not decoded and the command cycle in progress is not affected. When S# is driven HIGH, the device goes to standby. The operation is terminated by driving S# HIGH at any time during data output.

Table 14: Data/Address Lines for READ ID Command

Command Name	Data In	Data Out	Unique ID is Output
READ ID	DQ0	DQ0	Yes

Table 15: Read ID Data Out

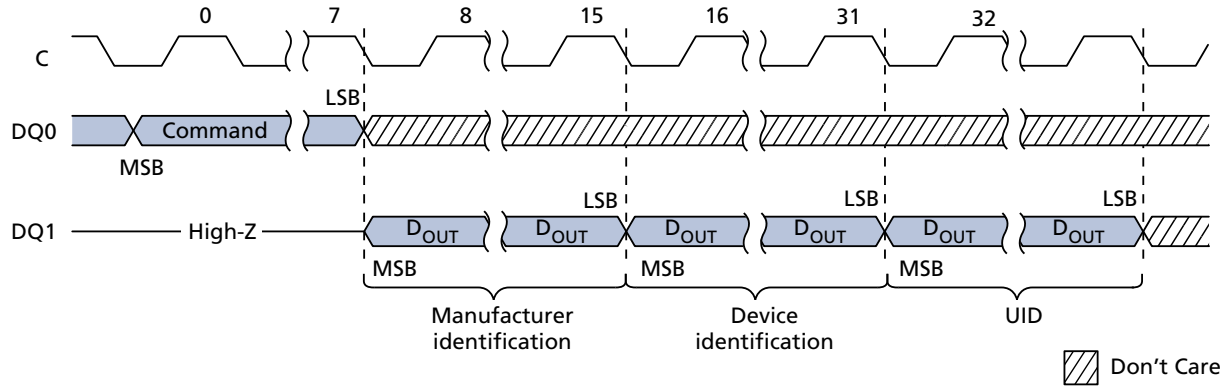
Size (Bytes)	Name	Content Value	Assigned by
1	**Manufacturer ID**	20h	JEDEC
2	**Device ID**		
	Memory Type	BAh	Manufacturer
	Memory Capacity	17h (64Mb)	
17	**Unique ID**		
	1 Byte: Length of data to follow	10h	Factory
	2 Bytes: Extended device ID and device configuration information	ID and information such as uniform architecture, and HOLD or RESET functionality	
	14 Bytes: Customized factory data	Unique ID code (*n* read-only bytes)	

Note: 1. The 17 bits of information in the unique ID is read by the READ ID command.

Table 16: Extended Device ID, First Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	0 = 65nm technology	1 = Reserved 0 = Standard BP scheme	Reserved	0 = HOLD	Addressing: 0 = by byte	Architecture: 00 = Uniform	

Figure 11: READ ID Command



Note: 1. The READ ID command is represented by the extended SPI protocol timing shown first.

READ SERIAL FLASH DISCOVERY PARAMETER Command

To execute READ SERIAL FLASH DISCOVERY PARAMETER command, S# is driven LOW. The command code is input on DQ0, followed by three address bytes and 8 dummy clock cycles. The device outputs the information starting from the specified address. When the 2048-byte boundary is reached, the data output wraps to address 0 of the serial Flash discovery parameter table. The operation is terminated by driving S# HIGH at any time during data output.

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

Note: Data to be stored in the serial Flash discovery parameter area is still in the definition phase.

Table 17: Serial Flash Discovery Parameter – Header Structure

DW	Description		Byte Address	Bits	Data
1	SFDP signature		00h	7:0	53h
			01h	7:0	46h
			02h	7:0	44h
			03h	7:0	50h
2	SFDP revision	Minor	04h	7:0	00h
		Major	05h	7:0	01h
	Number of parameter headers		06h	7:0	00h
	Unused		07h	7:0	FFh
3	Parameter ID (1)	LSB	08h	7:0	00h
	Parameter revision	Minor	09h	7:0	00h
		Major	0Ah	7:0	01h
	Parameter length (in DW)		0Bh	7:0	09h

Table 17: Serial Flash Discovery Parameter – Header Structure (Continued)

DW	Description		Byte Address	Bits	Data
4	Parameter table pointer		0Ch	7:0	30h
			0Dh	7:0	00h
			0Eh	7:0	00h
	Parameter 1 ID	MSB	0Fh	7:0	FFh
5	Parameter 2 ID	LSB	10h	7:0	03h
	Parameter revision	Minor	11h	7:0	00h
		Major	12h	7:0	01h
	Parameter length (in DW)		13h	7:0	02h
6	Parameter table pointer		14h	7:0	00h
			15h	7:0	01h
			16h	7:0	00h
	Parameter 2 ID	MSB	17h	7:0	FFh

Note: 1. Locations 10h to 2Fh contain FFh.

Table 18: Parameter ID

DW	Description	Byte Address	Bits	Data
1	Minimum block/sector erase sizes	30h	1:0	01b
	Write granularity		2	1
	WRITE ENABLE command required for writing to volatile status registers		3	0
	WRITE ENABLE command code select for writing to volatile status register		4	0
	Unused		7:5	111b
	4KB ERASE command code	31h	7:0	20h
	Supports 1-1-2 fast read	32h	0	1
	Address bytes		2:1	00b
	Supports double transfer rate clocking		3	0
	Supports 1-2-2 fast read		4	1
	Supports 1-4-4 fast read		5	1
	Supports 1-1-4 fast read		6	1
	Unused		7	1
	Reserved	33h	7:0	FFh
2	Flash size (in bits)	34h	7:0	FFh
		35h	7:0	FFh
		36h	7:0	FFh
		37h	7:0	03h

Table 18: Parameter ID (Continued)

DW	Description	Byte Address	Bits	Data
3	1-4-4 FAST READ DUMMY cycle count	38h	4:0	01010b
	1-4-4 fast read number of mode bits		7:5	000b
	1-4-4 FAST READ command code	39h	7:0	EBh
	1-1-4 FAST READ DUMMY cycle count	3Ah	4:0	01000b
	1-1-4 fast read number of mode bits		7:5	000b
	1-1-4 FAST READ command code	3Bh	7:0	6Bh
4	1-1-2 FAST READ DUMMY cycle count	3Ch	4:0	01000b
	1-1-2 fast read number of mode bits		7:5	000b
	1-1-2 FAST READ command code	3Dh	7:0	3Bh
	1-2-2 FAST READ DUMMY cycle count	3Eh	4:0	01000b
	1-2-2 fast read number of mode bits		7:5	000b
	1-2-2 Instruction opcode	3Fh	7:0	BBh
5	Supports 2-2-2 fast read	40h	0	0
	Reserved		3:1	111b
	Supports 4-4-4 fast read		4	0
	Reserved		7:5	111b
	Reserved	43:41h	FFFFFFh	FFFFFFh
6	Reserved	45:44h	FFFFh	FFFFh
	2-2-2 FAST READ DUMMY cycle count	46h	4:0	00000b
	2-2-2 fast read number of mode bits		7:5	000b
	2-2-2 FAST READ command code	47h	7:0	FFh
7	Reserved	49:48h	FFFFh	FFFFh
	4-4-4 FAST READ DUMMY cycle count	4Ah	4:0	00000b
	4-4-4 fast read number of mode bits		7:5	000b
	4-4-4 FAST READ command code	4Bh	7:0	FFh
8	Sector type 1 size	4Ch	7:0	0Ch
	Sector type 1 command code	4Dh	7:0	20h
	Sector type 2 size	4Eh	7:0	10h
	Sector type 2 command code	4Fh	7:0	D8h
9	Sector type 3 size	50h	7:0	00h
	Sector type 3 command code	51h	7:0	00h
	Sector type 4 size	52h	7:0	00h
	Sector type 4 command code	53h	7:0	00h

Table 18: Parameter ID (Continued)

DW	Description	Byte Address	Bits	Data
10	Multiplier from typical to maximum (ERASE time)	57h:54h	3:0	0100b
	Sector type 1 ERASE typical time		8:4	01101b
			10:9	01b
	Sector type 2 ERASE typical time		15:11	00100b
			17:16	10b
	Sector type 3 ERASE typical time		22:18	00010b
			24:23	10b
	Sector type 4 ERASE typical time		29:25	00000b
			31:30	00b
11	Multiplier from typical to maximum (Page/Byte PROGRAM time)	5Bh:58h	3:0	0010b
	Page size		7:4	1000b
	Page PROGRAM (typical time)		12:8	01010b
			13	1b
	Byte PROGRAM (typical time)		17:14	1110b
			18	0b
	Byte PROGRAM (typical time, additional byte)		22:19	0000b
			23	0b
	Chip ERASE (typical time)		28:24	01011b
			30:29	10b
	Reserved		31	1b
12	Prohibited operations during PROGRAM SUSPEND	5F:5Ch	3:0	1100b
	Prohibited operations during ERASE SUSPEND		7:4	0110b
	Reserved		8	1b
	Program resume to suspend interval		12:9	0000b
	Suspend in progress PROGRAM maximum latency		17:13	11000b
			19:18	01b
	Erase resume to suspend interval		23:20	0010b
	Suspend in progress ERASE maximum latency		28:24	11000b
			30:29	01b
	Suspend/Resume supported		31	0b
13	PROGRAM RESUME command	60h	7:0	7Ah
	PROGRAM SUSPEND command	61h	7:0	75h
	RESUME command	62h	7:0	7Ah
	SUSPEND command	63h	7:0	75h

Table 18: Parameter ID (Continued)

DW	Description	Byte Address	Bits	Data
14	Reserved	67h:64h	1:0	11b
	Status register polling device busy		2	0b
			3	1b
			7:4	1111b
	Exit deep-power-down to next operation delay		12:8	00000b
			14:13	00b
	EXIT DEEP-POWER-DOWN command		22:15	0h
	ENTER DEEP-POWER-DOWN command		30:23	0h
	Deep-power-down supported		31	1b
15	4-4-4 mode disable sequences	6Bh:68h	3:0	1010b
	4-4-4 mode enable sequences		8:4	1_0100b
	0-4-4 mode supported		9	1b
	0-4-4 mode exit method		15:10	00_0011b
	0-4-4 mode enter method		19:16	0010b
	Quad enable requirements		22:20	000b
	HOLD and WP disable		23	1b
	Reserved		31:24	FFh
16	Volatile or nonvolatile register and write enable	6Fh:6Ch	6:0	0000001b
	Reserved		7	1b
	Soft reset adn rescue sequence support		13:8	111101b
	Exit 4-byte addressing		23:14	00_1111_0110b
	Enter 4-byte addressing		31:24	0011_0110b

Table 19: RPMC Features

Description		Byte Address	Bits	Data
RPMC Features	Flash hardening	100h	0	0b
	RPMC_Size		1	0b
			2	1b
	Reserved		3	1b
	Num_Counter		7:4	0011b
	OP1	101h	7:0	9Bh
	OP2	102h	7:0	96h
	Update Rate	103h	3:0	0000b
	Reserved		7:4	1111b

Table 19: RPMC Features (Continued)

Description		Byte Address	Bits	Data
RPMC Delay Parameters	Read counter polling delay	104h	4:0	00110b
			6:5	11b
			7	1b
	Write counter polling short delay (no erase)	105h	4:0	00011b
			6:5	11b
			7	1b
	Write counter polling long delay (typical sector)	106h	4:0	00010b
			6:5	10b
			7	1b
	Reserved	107h	7:0	FFh

READ MEMORY Operations

The device supports default reading and writing to an A[MAX:MIN] of A[23:0].

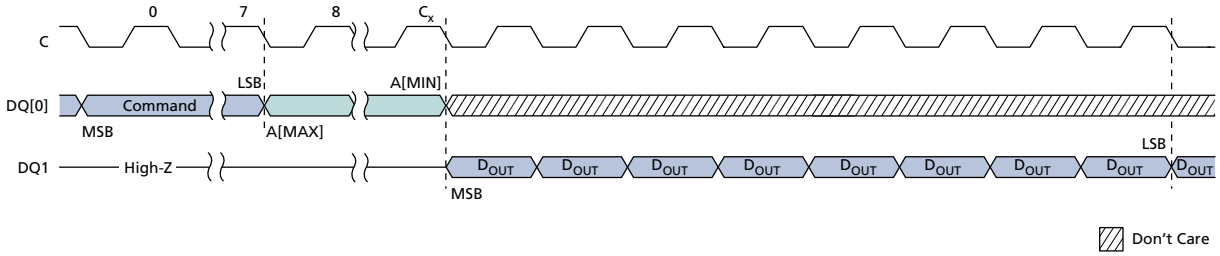
To execute READ MEMORY commands, S# is driven LOW. The command code is input on DQ*n*, followed by input on DQ*n* of three address bytes. Each address bit is latched in during the rising edge of the clock. The addressed byte can be at any location, and the address automatically increments to the next address after each byte of data is shifted out; therefore, the entire memory can be read with a single command. The operation is terminated by driving S# HIGH at any time during data output.

Table 20: Command/Address/Data Lines for READ MEMORY Commands

	Command Name					
	READ	FAST READ	DUAL OUTPUT FAST READ	DUAL INPUT/OUTPUT FAST READ	QUAD OUTPUT FAST READ	QUAD INPUT/OUTPUT FAST READ
	03	0B	3B	BB	6B	EB
Extended SPI Protocol						
Command Input	DQ0	DQ0	DQ0	DQ0	DQ0	DQ0
Address Input	DQ0	DQ0	DQ0	DQ[1:0]	DQ0	DQ[3:0]
Data Output	DQ1	DQ1	DQ[1:0]	DQ[1:0]	DQ[3:0]	DQ[3:0]

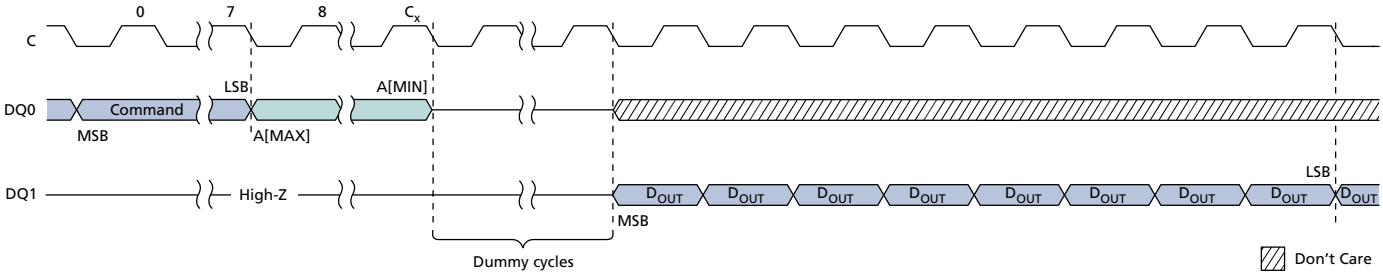
Note: 1. FAST READ is similar to READ, but requires dummy clock cycles following the address bytes and can operate at a higher frequency (f_C).

Figure 12: READ Command



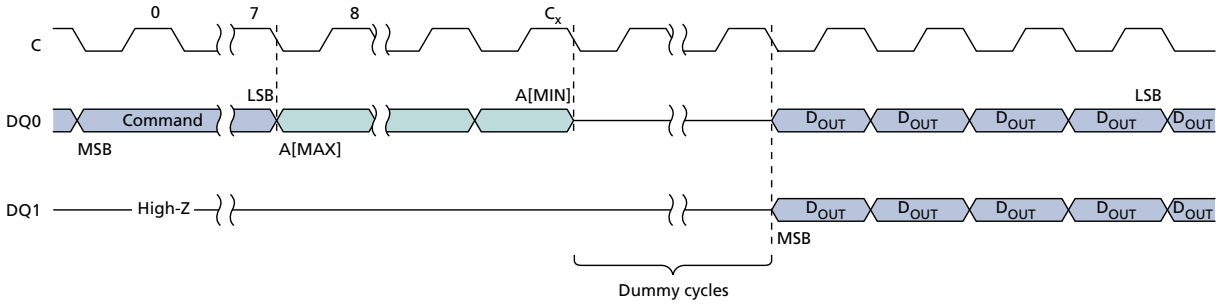
Note: 1. $C_x = 7 + (A[MAX] + 1)$.

Figure 13: FAST READ Command



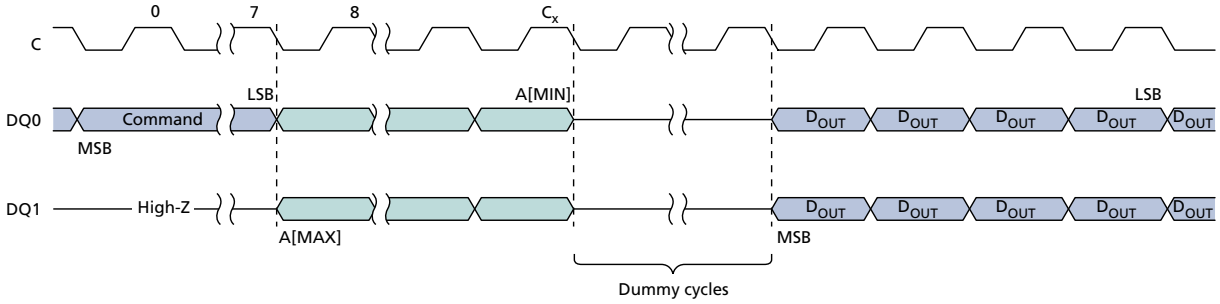
Note: 1. $C_x = 7 + (A[MAX] + 1)$.

Figure 14: DUAL OUTPUT FAST READ



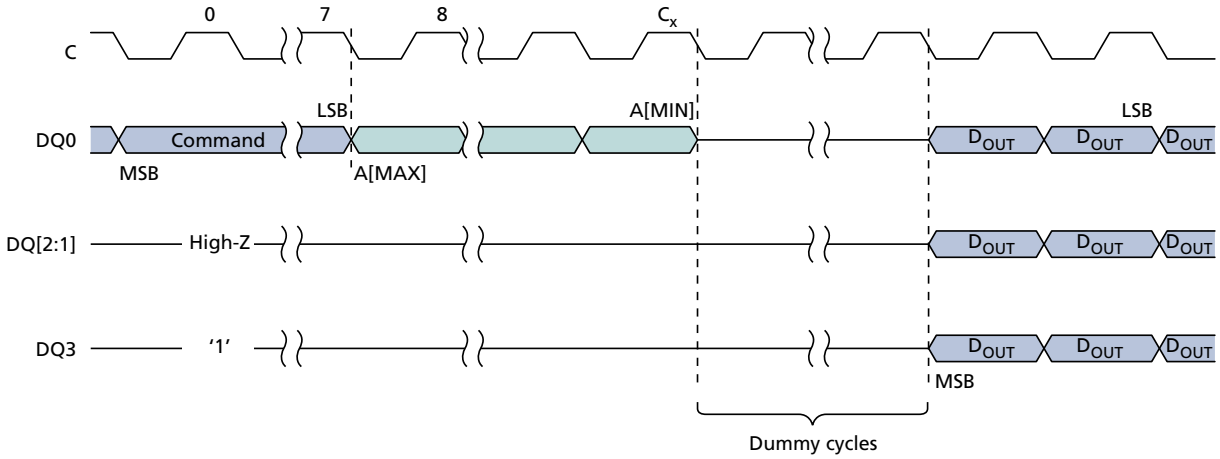
Note: 1. $C_x = 7 + (A[MAX] + 1)$.

Figure 15: DUAL INPUT/OUTPUT FAST READ Command



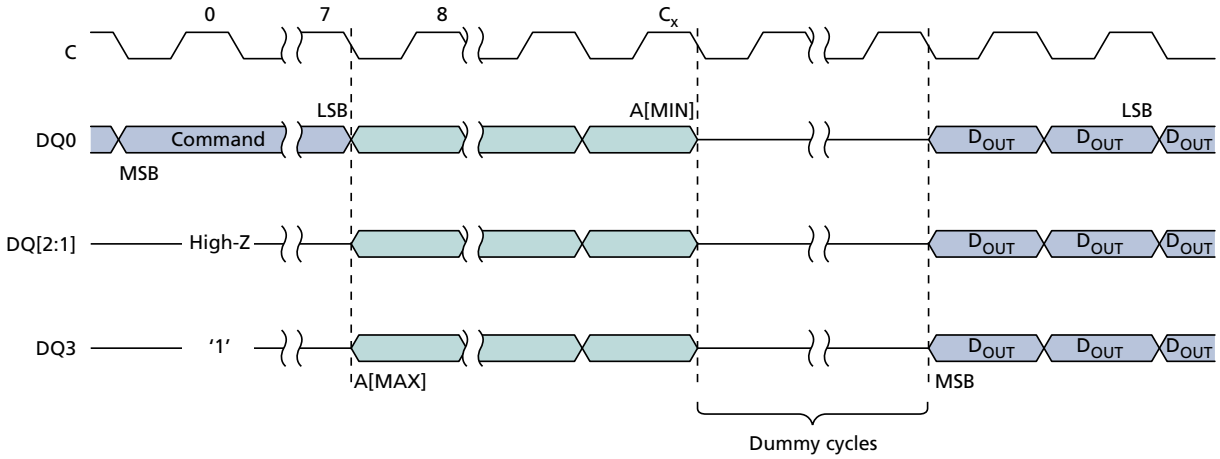
Note: 1. $C_x = 7 + (A[MAX] + 1)/2$.

Figure 16: QUAD OUTPUT FAST READ Command



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

Figure 17: QUAD INPUT/OUTPUT FAST READ Command



Note: 1. C_x = 7 + (A[MAX] + 1)/4.

PROGRAM Operations

PROGRAM commands are initiated by first executing the WRITE ENABLE command to set the write enable latch bit to 1. S# is then driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQ0, followed by input on DQ[n] of address bytes and at least one data byte. Each address bit is latched in during the rising edge of the clock. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is ^{t}PP.

If the bits of the least significant address, which is the starting address, are not all zero, all data transmitted beyond the end of the current page is programmed from the starting address of the same page. If the number of bytes sent to the device exceed the maximum page size, previously latched data is discarded and only the last maximum page-size number of data bytes are guaranteed to be programmed correctly within the same page. If the number of bytes sent to the device is less than the maximum page size, they are correctly programmed at the specified addresses without any effect on the other bytes of the same page.

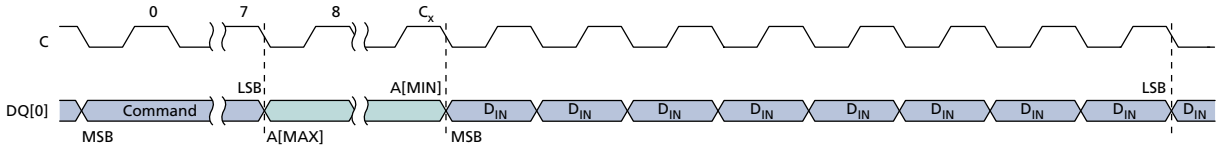
When the operation is in progress, the write in progress bit is set to 1. The write enable latch bit is cleared to 0, whether the operation is successful or not. The status register and flag status register can be polled for the operation status. An operation can be paused or resumed by the PROGRAM/ERASE SUSPEND or PROGRAM/ERASE RESUME command, respectively. When the operation completes, the write in progress bit is cleared to 0.

If the operation times out, the write enable latch bit is reset and the program fail bit is set to 1. If S# is not driven HIGH, the command is not executed and the write enable latch remains set to 1. When a command is applied to a protected sector, the command is not executed, and the write enable latch bit remains set to 1.

Table 21: Data/Address Lines for PROGRAM Commands

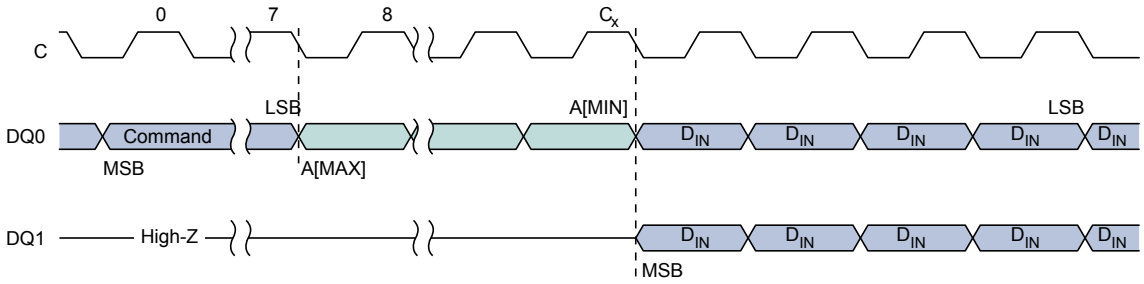
Command Name	Data In	Address In
PAGE PROGRAM	DQ0	DQ0
DUAL INPUT FAST PROGRAM	DQ[1:0]	DQ0
EXTENDED DUAL INPUT FAST PROGRAM	DQ[1:0]	DQ[1:0]
QUAD INPUT FAST PROGRAM	DQ[3:0]	DQ0
EXTENDED QUAD INPUT FAST PROGRAM	DQ[3:0]	DQ[3:0]

Figure 18: PAGE PROGRAM Command



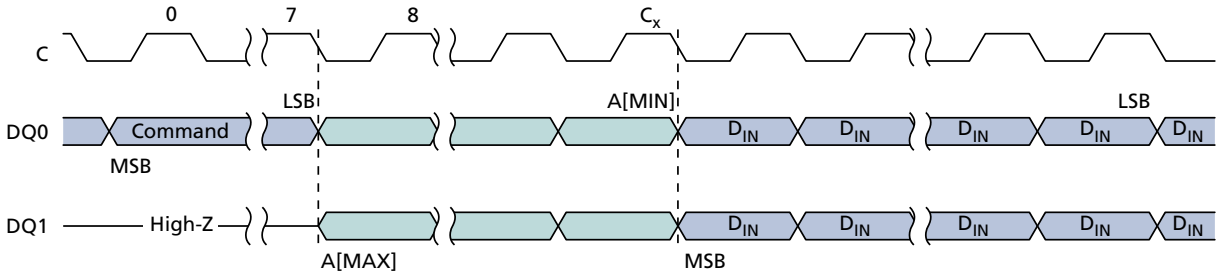
Note: 1. C_x = 7 + (A[MAX] + 1).

Figure 19: DUAL INPUT FAST PROGRAM Command



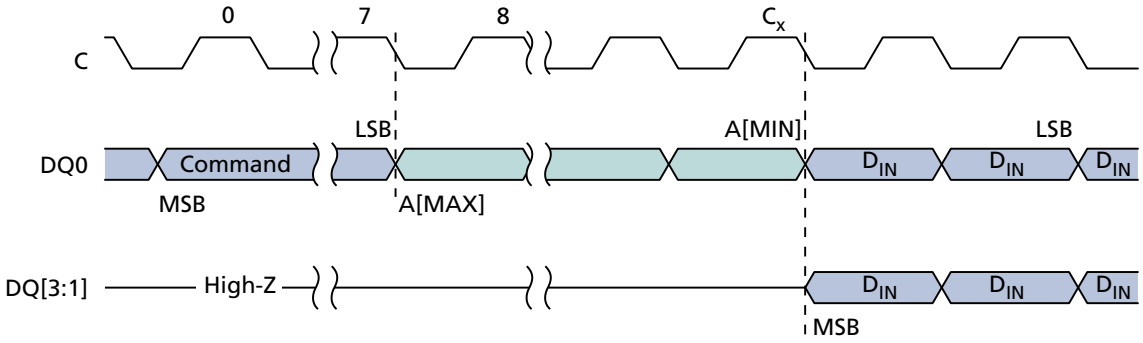
Note: 1. C_x = 7 + (A[MAX] + 1).

Figure 20: EXTENDED DUAL INPUT FAST PROGRAM Command



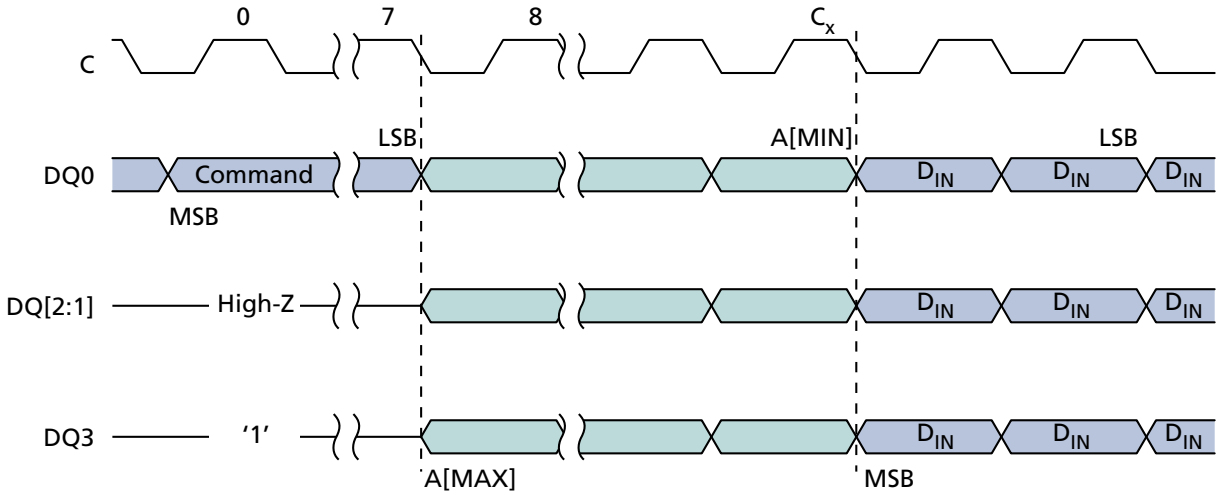
Note: 1. $C_x = 7 + (A[MAX] + 1)/2$.

Figure 21: QUAD INPUT FAST PROGRAM Command



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

Figure 22: EXTENDED QUAD INPUT FAST PROGRAM Command



Note: 1. $C_x = 7 + (A[MAX] + 1)/4$.

WRITE Operations

WRITE ENABLE Command

The WRITE ENABLE operation sets the write enable latch bit. To execute a WRITE ENABLE command, S# is driven LOW and held LOW until the eighth bit of the command code has been latched in, after which it must be driven HIGH. The command code is input on DQ0.

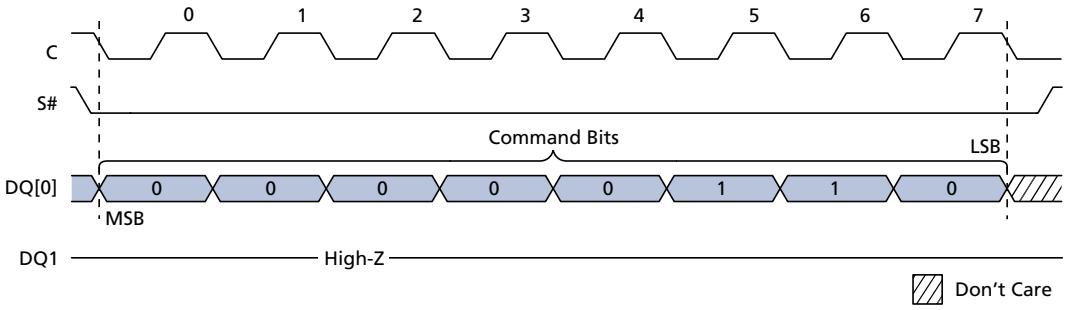
The write enable latch bit must be set before every PROGRAM, ERASE, and WRITE command. If S# is not driven HIGH after the command code has been latched in, the command is not executed and the write enable latch remains cleared to its default setting of 0.

WRITE DISABLE Command

The WRITE DISABLE operation clears the write enable latch bit. To execute a WRITE DISABLE command, S# is driven LOW and held LOW until the eighth bit of the command code has been latched in, after which it must be driven HIGH. The command code is input on DQ0.

If S# is not driven HIGH after the command code has been latched in, the command is not executed and the write enable latch remains set to 1.

Figure 23: WRITE ENABLE and WRITE DISABLE Command Sequence



Note: 1. Shown here is the WRITE ENABLE command code, which is 06h or 0000 0110 binary. The WRITE DISABLE command sequence is identical, except the WRITE DISABLE command code is 04h or 0000 0100 binary.

ERASE Operations

SUBSECTOR ERASE Command

To execute the SUBSECTOR ERASE command (and set the selected subsector bits set to FFh), the WRITE ENABLE command must be issued to set the write enable latch bit to 1. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQ0, followed by three address bytes; any address within the subsector is valid. Each address bit is latched in during the rising edge of the clock. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is ^{t}SSE.

The operation can be suspended and resumed by the PROGRAM/ ERASE SUSPEND and PROGRAM/ERASE RESUME commands, respectively. If the write enable latch bit is not set, the device ignores the SUBSECTOR ERASE command and no error bits are set to indicate operation failure.

When the operation is in progress, the write in progress bit is set to 1. The write enable latch bit is cleared to 0, whether the operation is successful or not. The status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0.

If the operation times out, the write enable latch bit is reset and the erase error bit is set to 1. If S# is not driven HIGH, the command is not executed and the write enable latch remains set to 1. When a command is applied to a protected subsector, the command is not executed. Instead, the write enable latch bit remains set to 1.

SECTOR ERASE Command

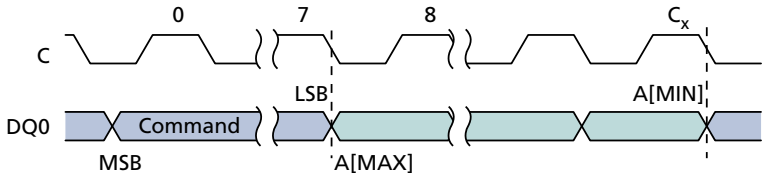
To execute the SECTOR ERASE command (and set selected sector bits to FFh), the WRITE ENABLE command must be issued to set the write enable latch bit to 1. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQ0, followed by three address bytes; any address within the sector is valid. Each address bit is latched in during the rising edge of the clock. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is ^{t}SE.

The operation can be suspended and resumed by the PROGRAM/ ERASE SUSPEND and PROGRAM/ERASE RESUME commands, respectively. If the write enable latch bit is not set, the device ignores the SECTOR ERASE command and no error bits are set to indicate operation failure.

When the operation is in progress, the write in progress bit is set to 1 and the write enable latch bit is cleared to 0, whether the operation is successful or not. The status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0.

If the operation times out, the write enable latch bit is reset and erase error bit is set to 1. If S# is not driven HIGH, the command is not executed and the write enable latch remains set to 1. When a command is applied to a protected sector, the command is not executed. Instead, the write enable latch bit remains set to 1.

Figure 24: SUBSECTOR and SECTOR ERASE Command



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

BULK ERASE Command

To initiate the BULK ERASE command, the WRITE ENABLE command must be issued to set the write enable latch bit to 1. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQ0. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is ^{t}BE.

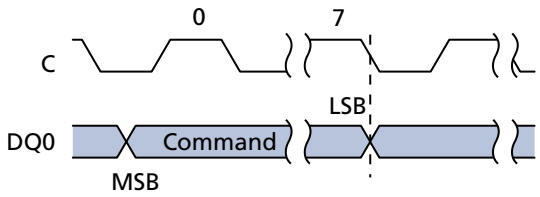
If the write enable latch bit is not set, the device ignores the BULK ERASE command and no error bits are set to indicate operation failure.

When the operation is in progress, the write in progress bit is set to 1 and the write enable latch bit is cleared to 0, whether the operation is successful or not. The status register can be polled for the operation status. When the operation completes, the write in progress bit is cleared to 0.

If the operation times out, the write enable latch bit is reset and erase error bit is set to 1. If S# is not driven HIGH, the command is not executed and the write enable latch remains set to 1.

The command is not executed if any sector is locked. Instead, the write enable latch bit remains set to 1.

Figure 25: BULK ERASE Command



PROGRAM/ERASE SUSPEND Command

To initiate the PROGRAM/ERASE SUSPEND command, S# is driven LOW. The command code is input on DQ0. The operation is terminated by the PROGRAM/ERASE RESUME command.

PROGRAM/ERASE SUSPEND command enables the memory controller to interrupt and suspend an array PROGRAM or ERASE operation within the program/erase latency.

If a SUSPEND command is issued during a PROGRAM operation, then the flag status register bit 2 is set to 1. After erase/program latency time, the flag status register bit 7 is also set to 1, showing the device to be in a suspended state, waiting for any operation (see the Operations Allowed/Disallowed During Device States table).

If a SUSPEND command is issued during an ERASE operation, then the flag status register bit 6 is set to 1. After erase/program latency time, the flag status register bit 7 is also set to 1, showing that device to be in a suspended state, waiting for any operation (see the Operations Allowed/Disallowed During Device States table).

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

During an ERASE SUSPEND operation, a PROGRAM or READ operation is possible in any sector except the one in a suspended state. Reading from a sector that is in a suspended state will output indeterminate data. The device ignores a PROGRAM command to a sector that is in an ERASE SUSPEND state; it also sets the flag status register bit 4 to 1: program failure/protection error, and leaves the write enable latch bit unchanged. The commands allowed during an erase suspend state include the WRITE LOCK REGISTER command. When the ERASE operation resumes, it does not check the new lock status of the WRITE LOCK REGISTER command.

During a PROGRAM SUSPEND operation, a READ operation is possible in any page except the one in a suspended state. Reading from a page that is in a suspended state will output indeterminate data.

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

Table 22: Suspend Parameters

Parameter	Condition	Typ	Max	Units	Notes
Erase to suspend	Sector erase or erase resume to erase suspend	100	–	µs	1
Program to suspend	Program resume to program suspend	5	–	µs	1
Subsector erase to suspend	Subsector erase or subsector erase resume to erase suspend	50	–	µs	1
Suspend latency	Program	7	–	µs	2
Suspend latency	Subsector erase	15	–	µs	2
Suspend latency	Erase	15	–	µs	3

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

Table 23: Operations Allowed/Disallowed During Device States

Note 1 applies to entire table

Operation	Standby State	Program or Erase State	Subsector Erase Suspend or Program Suspend State	Erase Suspend State	Notes
READ	Yes	No	Yes	Yes	2
PROGRAM	Yes	No	No	Yes/No	3
ERASE	Yes	No	No	No	4
WRITE	Yes	No	No	No	5
WRITE	Yes	No	Yes	Yes	6
READ	Yes	Yes	Yes	Yes	7
SUSPEND	No	Yes	No	No	8

Notes:
1. The device can be in only one state at a time. Depending on the state of the device, some operations are allowed (Yes) and others are not (No). For example, when the device is in the standby state, all operations except SUSPEND are allowed in any sector. For all device states except the erase suspend state, if an operation is allowed or disallowed in one sector, it is allowed or disallowed in all other sectors. In the erase suspend state, a PROGRAM operation is allowed in any sector except the one in which an ERASE operation has been suspended.
2. All READ operations except READ STATUS REGISTER and READ FLAG REGISTER. When issued to a sector or subsector that is simultaneously in an erase suspend state, the READ operation is accepted, but the data output is not guaranteed until the erase has completed.
3. All PROGRAM operations except PROGRAM OTP. In the erase suspend state, a PROGRAM operation is allowed in any sector (Yes) except the sector (No) in which an ERASE operation has been suspended.
4. Applies to the SECTOR ERASE or SUBSECTOR ERASE operation.
5. Applies to the following operations: WRITE STATUS REGISTER, PROGRAM OTP, and BULK ERASE.
6. Applies to the following operations: WRITE ENABLE, WRITE DISABLE, CLEAR FLAG STATUS REGISTER, WRITE LOCK REGISTER.
7. Applies to the READ STATUS REGISTER or READ FLAG STATUS REGISTER operation.
8. Applies to the PROGRAM SUSPEND or ERASE SUSPEND operation.

PROGRAM/ERASE RESUME Command

To initiate the PROGRAM/ERASE RESUME command, S# is driven LOW. The command code is input on DQ0. The operation is terminated by driving S# HIGH.

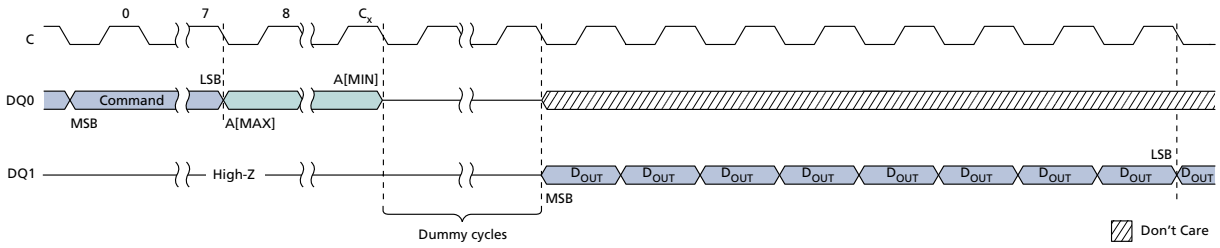
When this command is executed, the status register write in progress bit is set to 1, and the flag status register program erase controller bit is set to 0. This command is ignored if the device is not in a suspended state.

ONE TIME PROGRAMMABLE Operations

READ OTP ARRAY Command

To initiate a READ OTP ARRAY command, S# is driven LOW. The command code is input on DQ0, followed by three bytes and dummy clock cycles. Each address bit is latched in during the rising edge of C. Data is shifted out on DQ1, beginning from the specified address and at a maximum frequency of ^{f}C (MAX) on the falling edge of the clock. The address increments automatically to the next address after each byte of data is shifted out. There is no rollover mechanism; therefore, if read continuously, after location 40h, the device continues to output data at location 40h. The operation is terminated by driving S# HIGH at any time during data output.

Figure 26: READ OTP Command



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

PROGRAM OTP ARRAY Command

To initiate the PROGRAM OTP ARRAY command, the WRITE ENABLE command must be issued to set the write enable latch bit to 1; otherwise, the PROGRAM OTP ARRAY command is ignored. S# is driven LOW and held LOW until the eighth bit of the last data byte has been latched in, after which it must be driven HIGH. The command code is input on DQ0, followed by three bytes and at least one data byte. Each address bit is latched in during the rising edge of the clock. When S# is driven HIGH, the operation, which is self-timed, is initiated; its duration is ^{t}POTP. There is no rollover mechanism; therefore, after a maximum of 65 bytes are latched in and subsequent bytes are discarded.

PROGRAM OTP ARRAY programs, at most, 64 bytes to the OTP memory area and one OTP control byte. When the operation is in progress, the write in progress bit is set to 1. The write enable latch bit is cleared to 0, whether the operation is successful or not. When the operation completes, the write in progress bit is cleared to 0.

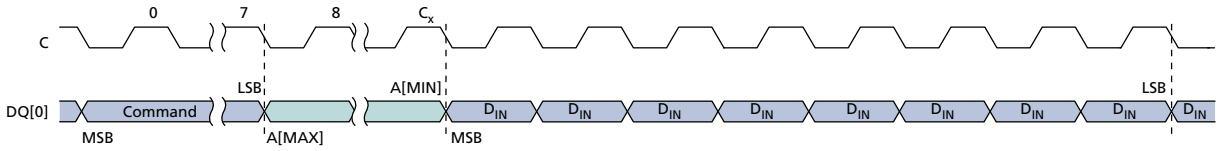
If the operation times out, the write enable latch bit is reset and the program fail bit is set to 1. If S# is not driven HIGH, the command is not executed and the write enable latch remains set to 1.

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

Table 24: OTP Control Byte (Byte 64)

Bit	Name	Settings	Description
0	OTP control byte	0 = Locked 1 = Unlocked (Default)	Used to permanently lock the OTP array (byte 64). When bit 0 = 1, the OTP array can be programmed. When bit 0 = 0, the OTP array is read only. Once bit 0 has been programmed to 0, it can no longer be changed to 1. PROGRAM OTP ARRAY is ignored, write enable latch bit remains set.

Figure 27: PROGRAM OTP Command



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

Power-Up and Power-Down

Power-Up and Power-Down Requirements

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

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

Normal precautions must be taken for supply line decoupling to stabilize the V_{CC} supply. Each device in a system should have the V_{CC} line decoupled by a suitable capacitor (typically 100nF) close to the package pins. At power-down, when V_{CC} drops from the operating voltage to below the power-on-reset threshold voltage shown here, all operations are disabled and the device does not respond to any command.
Note: If power-down occurs while a WRITE, PROGRAM, or ERASE cycle is in progress, data corruption may result.

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

Figure 28: Power-Up Timing

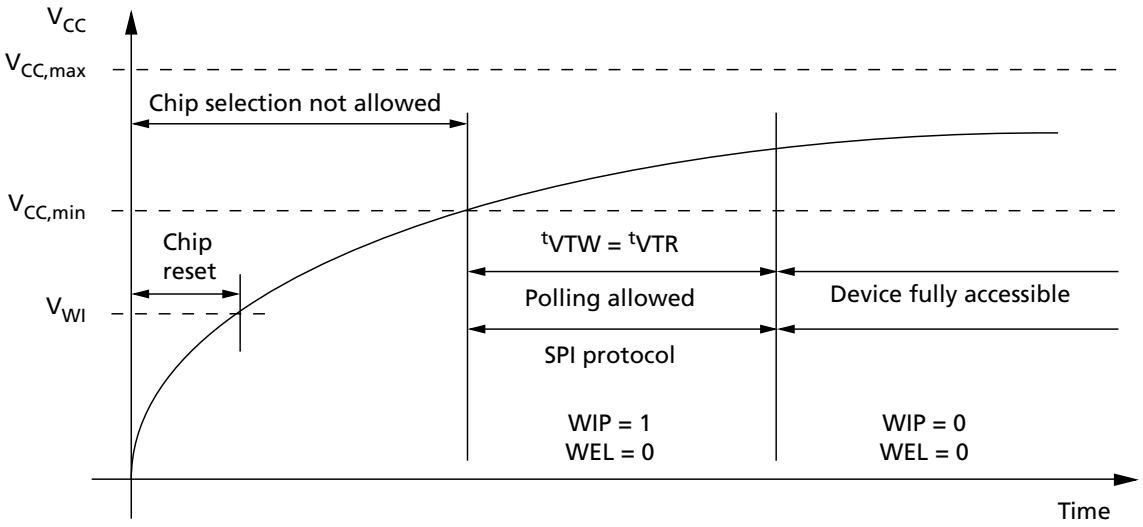


Table 25: Power-Up Timing and V_{WI} Threshold

Note 1 applies to entire table

Symbol	Parameter	Min	Max	Unit
^{t}VTR	$V_{CC,min}$ to read	–	150	µs
^{t}VTW	$V_{CC,min}$ to device fully accessible	–	150	µs
V_{WI}	Write inhibit voltage	1.5	2.5	V

Note: 1. Parameters listed are characterized only.

AC Reset Timings

Figure 29: Reset AC Timing During PROGRAM or ERASE Cycle

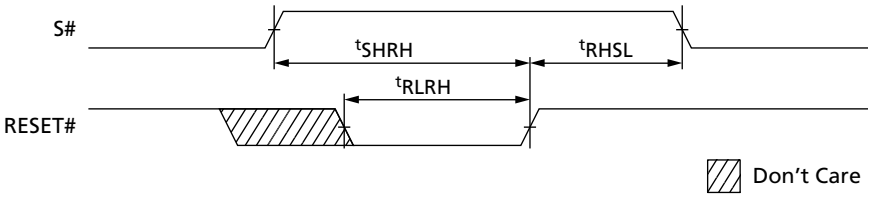


Figure 30: Serial Input Timing

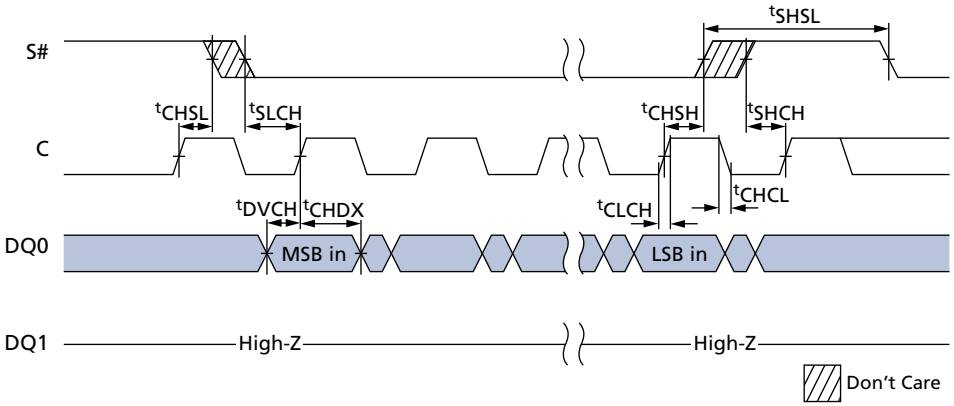


Figure 31: Write Protect Setup and Hold During WRITE STATUS REGISTER Operation (SRWD = 1)

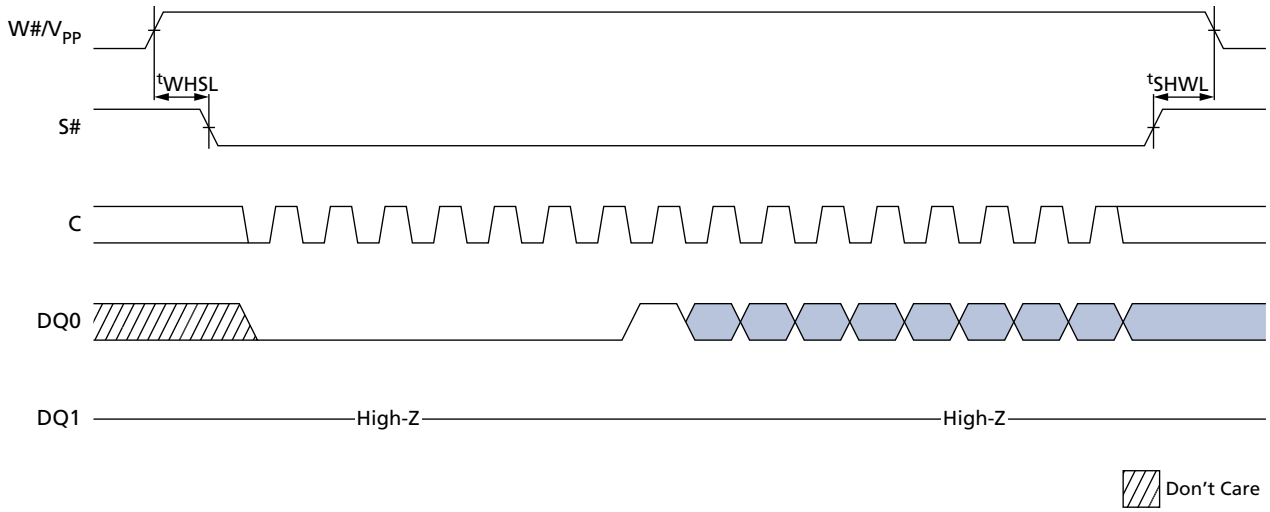


Figure 32: Hold Timing

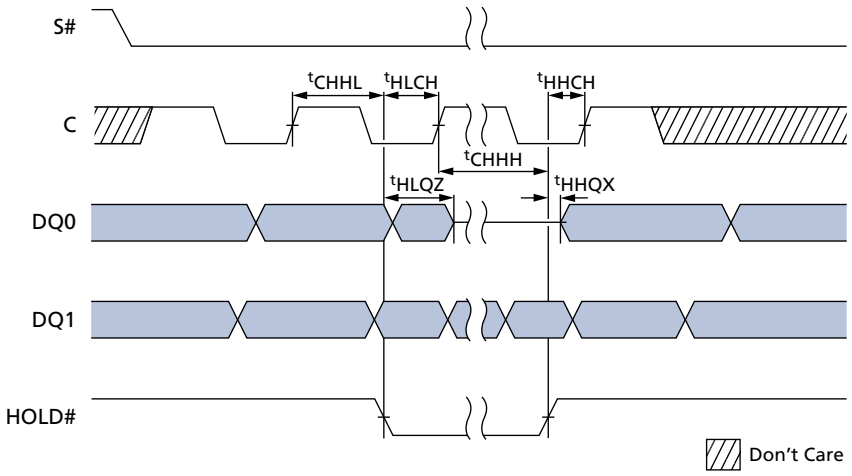
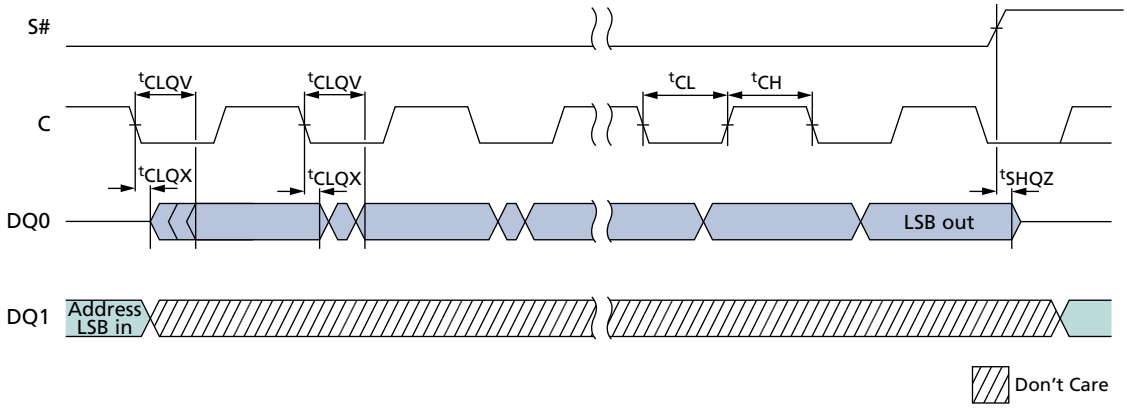


Figure 33: Output Timing



Absolute Ratings and Operating Conditions

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

Table 26: Absolute Ratings

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

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

Table 27: Operating Conditions

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

Table 28: Input/Output Capacitance

Note 1 applies to entire table

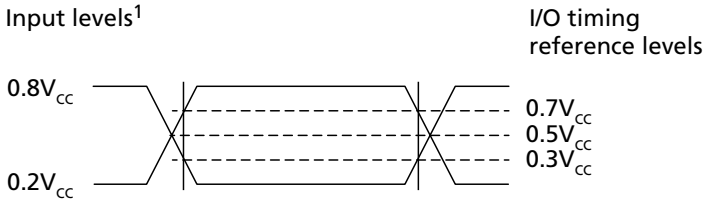
Symbol	Description	Test Condition	Min	Max	Units
$C_{IN/OUT}$	Input/output capacitance (DQ0/DQ1/DQ2/DQ3)	V_{OUT} = 0V	–	8	pF
C_{IN}	Input capacitance (other pins)	V_{IN} = 0V	–	6	pF

Note:
1. These parameters are sampled only, not 100% tested. T_A = 25°C at 54 MHz.

Table 29: AC Timing Input/Output Conditions

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

Figure 34: AC Timing Input/Output Reference Levels



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

DC Characteristics and Operating Conditions

Table 30: DC Current Characteristics and Operating Conditions

Parameter	Symbol	Test Conditions	Min	Max	Unit
Input leakage current	I_{LI}		–	±2	µA
Output leakage current	I_{LO}		–	±2	µA
Standby current	I_{CC1}	S = V_{CC}, V_{IN} = V_{SS} or V_{CC}	15	100	µA
Operating current	I_{CC3}	C = 0.1V_{CC}/0.9V_{CC} at 108 MHz, DQ1 = open	–	15	mA
		C = 0.1V_{CC}/0.9V_{CC} at 54 MHz, DQ1 = open	–	6	mA
Operating current (fast-read dual I/O)		C = 0.1V_{CC}/0.9V_{CC} at 108 MHz	–	18	mA
Operating current (fast-read quad I/O)		C = 0.1V_{CC}/0.9V_{CC} at 108 MHz	–	20	mA
Operating current (program)	I_{CC4}	S# = V_{CC}	–	20	mA
Operating current (write status register)	I_{CC5}	S# = V_{CC}	–	20	mA
Operating current (erase)	I_{CC6}	S# = V_{CC}	–	20	mA

Table 31: DC Voltage Characteristics and Operating Conditions

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

AC Characteristics and Operating Conditions

Table 32: AC Characteristics and Operating Conditions

Parameter	Symbol	Min	Typ[1]	Max	Unit	Notes
Clock frequency for all commands other than READ	^{f}C	DC	–	108	MHz	
Clock frequency for READ commands	^{f}R	DC	–	54	MHz	
Clock HIGH time	^{t}CH	4	–	–	ns	2
Clock LOW time	^{t}CL	4	–	–	ns	1
Clock rise time (peak-to-peak)	$^{t}CLCH$	0.1	–	–	V/ns	3, 4
Clock fall time (peak-to-peak)	$^{t}CHCL$	0.1	–	–	V/ns	3, 4
S# active setup time (relative to clock)	$^{t}SLCH$	4	–	–	ns	
S# not active hold time (relative to clock)	$^{t}CHSL$	4	–	–	ns	
Data-in setup time	$^{t}DVCH$	2	–	–	ns	
Data-in hold time	$^{t}CHDX$	3	–	–	ns	
S# active hold time (relative to clock)	$^{t}CHSH$	4	–	–	ns	
S# not active setup time (relative to clock)	$^{t}SHCH$	4	–	–	ns	
S# deselect time after a READ command	$^{t}SHSL1$	20	–	–	ns	
S# deselect time after a nonREAD command	$^{t}SHSL2$	50	–	–	ns	
Output disable time	$^{t}SHQZ$	–	–	8	ns	3
Clock LOW to output valid under 30pF	$^{t}CLQV$	–	–	7	ns	
Clock LOW to output valid under 10pF		–	–	5	ns	
Output hold time (clock LOW)	$^{t}CLQX$	1	–	–	ns	
Output hold time (clock HIGH)	$^{t}CHQX$	1	–	–	ns	
HOLD command setup time (relative to clock)	$^{t}HLCH$	4	–	–	ns	
HOLD command hold time (relative to clock)	$^{t}CHHH$	4	–	–	ns	
HOLD command setup time (relative to clock)	$^{t}HHCH$	4	–	–	ns	
HOLD command hold time (relative to clock)	$^{t}CHHL$	4	–	–	ns	
HOLD command to output Low-Z	$^{t}HHQX$	–	–	8	ns	3
HOLD command to output High-Z	$^{t}HLQZ$	–	–	8	ns	3
Write protect setup time	$^{t}WHSL$	20	–	–	ns	5
Write protect hold time	$^{t}SHWL$	100	–	–	ns	5
WRITE STATUS REGISTER cycle time	^{t}W	–	1.3	8	ms	
PAGE PROGRAM cycle time (256 bytes)	^{t}PP	–	0.5	5	ms	6
PAGE PROGRAM cycle time (*n* bytes)		–	int(n/8) × 0.015	5	ms	6, 7
PROGRAM OTP cycle time (64 bytes)		–	0.2	–	ms	6
4KB Subsector ERASE cycle time	$^{t}SSE1$	–	60	200	ms	
32KB Subsector ERASE cycle time	$^{t}SSE2$	–	0.22	3	s	
64KB Sector ERASE cycle time	^{t}SE	–	0.46	3	s	

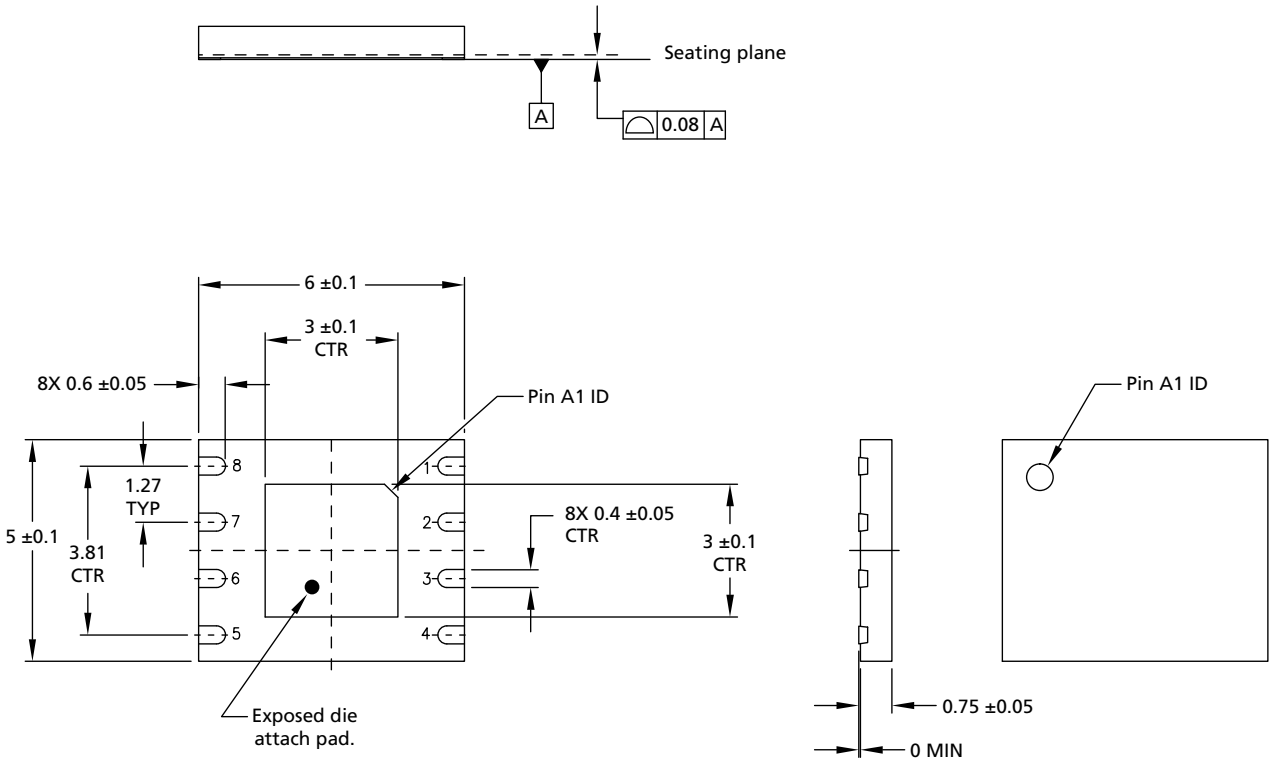
Table 32: AC Characteristics and Operating Conditions (Continued)

Parameter	Symbol	Min	Typ[1]	Max	Unit	Notes
Bulk ERASE cycle time	^{t}BE	–	45	250	s	

Notes:
1. Typical values given for T_A = 25°C.
2. ^{t}CH + ^{t}CL must add up to 1/^{f}C.
3. Value guaranteed by characterization; not 100% tested.
4. Expressed as a slew rate.
5. Only applicable as a constraint for a WRITE STATUS REGISTER command when STATUS REGISTER WRITE is set to 1.
6. When using the PAGE PROGRAM command to program consecutive bytes, optimized timings are obtained with one sequence including all the bytes versus several sequences of only a few bytes (1 < n < 256).
7. int(A) corresponds to the upper integer part of A. For example int(12/8) = 2, int(32/8) = 4 int(15.3) =16.

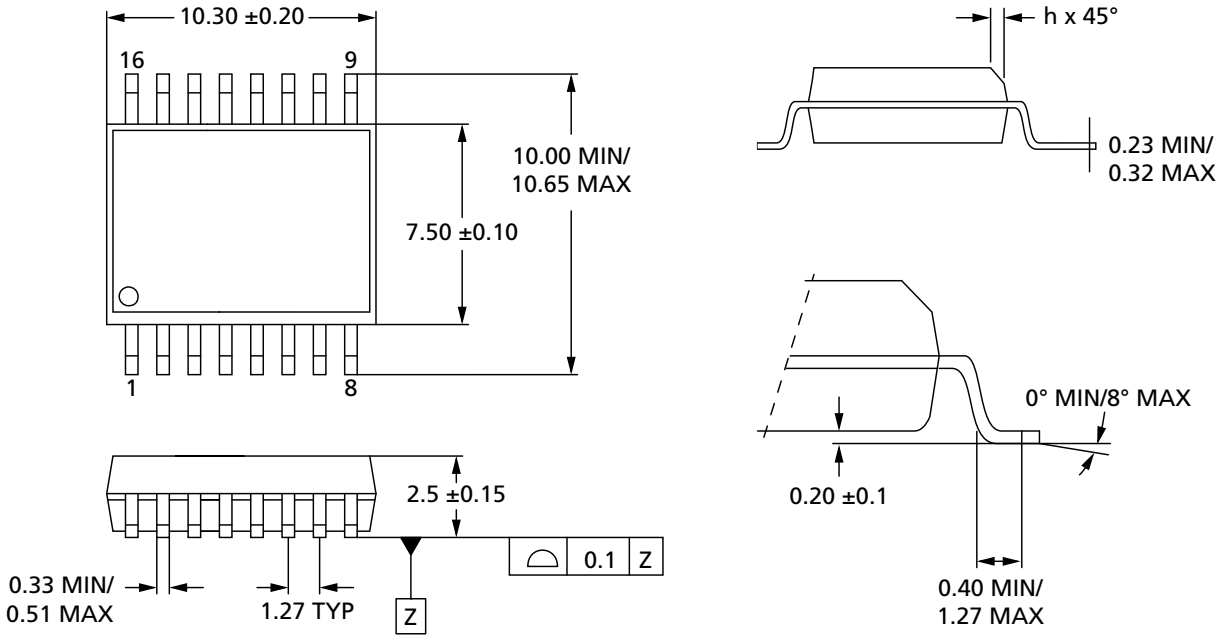
Package Dimensions

Figure 35: W-PDFN-8 6mm x 5mm (MLP8) - Package Code: W7



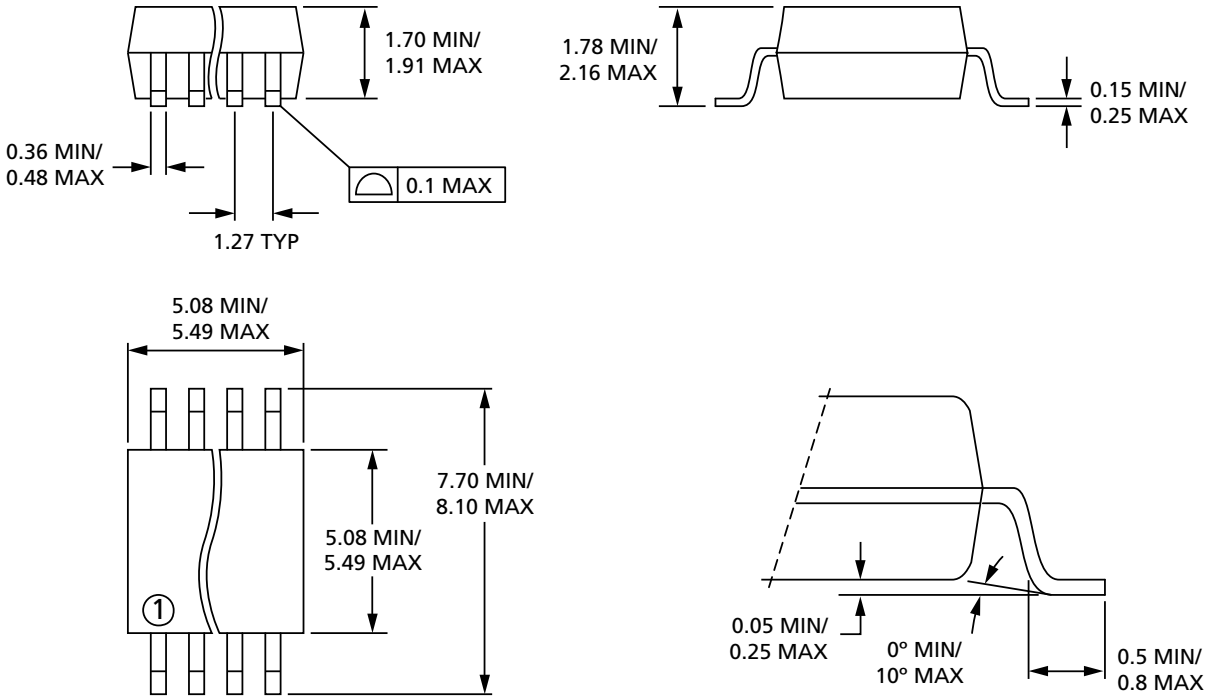
Notes:
1. All dimensions are in millimeters.
2. See Part Number Ordering Information for complete package names and details.

Figure 36: SOP2-16 (300 mils body width) – Package Code: SF



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

Figure 37: SOP2-8 (208 mils body width) – Package Code: SE



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

Part Number Ordering Information

A list of all active Serial NOR part numbers is available on Micron's Serial NOR Flash part catalog page. For further information on line items not listed here or on any aspect of this device, contact your nearest representative.

Table 33: Part Number Information

Part Number Category	Category Details
Device type	N25Q = Serial NOR Flash memory, Multiple Input/Output (Single, Dual, Quad I/O)
Density	064 = 64Mb
Technology	A = 65nm
Feature set	1 = Byte addressability; HOLD pin
Operating voltage	3 = V_{CC} = 2.7 to 3.6V
Block structure	E = Uniform (64KB and 4KB)
Package (RoHS-compliant)	W7 = W-PDFN-8 6mm x 5mm (MLP8 6mm x 5mm) SF = SOP2-16 300 mils body width (SO16W) SE = SOP2-8 208 mils body width (SO8W)
Temperature and test flow	4 = IT: –40°C to 85°C; Device tested with standard test flow
Security features	M = RPMC authenticated secure feature
Shipping material	E = Tray F = Tape and reel G = Tube

Table 34: Package Details

Micron SPI and JEDEC Package Name	Shortened Package Name	Package Description	M25P M45PE Symbol	N25Q Symbol	M25P M45PE Package Names	Alternate Package Name
W-PDFN-8 8mm x 5mm	WDFN/8mm x 5mm	Very, very thin plastic small outline, 8 terminal pads (no leads), 6mm x 5mm	–	W7	MLP8, VDFPN8	W-PSON1-8/6mm x 5mm, WSON
SOP2- 16/ 300 mil	SO16W	Small-outline integrated circuit, 16 pins, wide (300 mil)	MF	SF	SO16 wide 300 mil body width	SOIC-16/300 mil, SOP 16L 300 mil
SOP2- 8/ 208 mil	SO8W	Small-outline integrated circuit, 8-pins, wide (208 mil)	MW	SE	SO8 wide 208 mil body width	SOIC-8/208 mil, SOP 8L 200 mil

Revision History

Rev. B – 09/2013

- Added W7 packages and codes

Rev. A – 05/2013

- Initial release

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.