# EN25QE80A (2P)
# 8 Megabit 3V Serial Flash Memory with 4Kbyte Uniform Sector

## FEATURES

- Single power supply operation
  - Full voltage range: 2.3-3.6 volt
- Serial Interface Architecture
  - SPI Compatible: Mode 0 and Mode 3
- 8 M-bit Serial Flash
  - 8 M-bit / 1024 KByte / 4096 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO, WP#, HOLD#
  - Dual SPI: CLK, CS#, $DQ_0$, $DQ_1$, WP#, HOLD#
  - Quad SPI: CLK, CS#, $DQ_0$, $DQ_1$, $DQ_2$, $DQ_3$
  - Configurable dummy cycle number
- High performance
  - 104MHz clock rate for Standard SPI
  - 104MHz clock rate for two data bits
  - 104MHz clock rate for four data bits
- Low power consumption
  - 10 mA typical active current
  - 1 μA typical power down current
- Uniform Sector Architecture
  - 256 sectors of 4-Kbyte
  - 32 blocks of 32-Kbyte
  - 16 blocks of 64-Kbyte
  - Any sector or block can be erased individually
- Software and Hardware Write Protection
  - Write Protect all or portion of memory via software
  - Enable/Disable protection with WP# pin
- High performance program/erase speed
  - Page program time: 1ms typical
  - Sector erase time: 100ms typical
  - Half Block erase time: 300ms typical
  - Block erase time: 500ms typical
  - Chip erase time: 8 seconds typical
- Write Suspend and Write Resume
- Volatile Status Register Bits
- Lockable 3x1024 byte OTP security sector
- Support Serial Flash Discoverable Parameters (SFDP) signature
- Read Unique ID Number
- Minimum 100K endurance cycle
- Data retention time 20 years
- Package Options
  - 8 pins SOP 150mil body width
  - 8 pins SOP 200mil body width
  - 8 contact USON (3x2x0.45mm)
  - All Pb-free packages are compliant RoHS, Halogen-Free and REACH.
- Industrial temperature Range

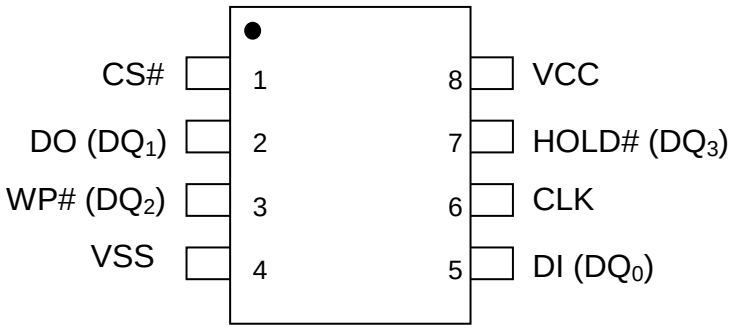
## GENERAL DESCRIPTION

The device is a 8 Megabit (1024K-byte) Serial Flash memory, with advanced write protection mechanisms. The device supports the single bit and four bits serial input and output commands via standard Serial Peripheral Interface (SPI) pins: Serial Clock, Chip Select, Serial $DQ_0$ (DI) and $DQ_1$ (DO), $DQ_2$ (WP#) and $DQ_3$ (HOLD#). The memory can be programmed 1 to 256 bytes at a time, using the Page Program instruction.

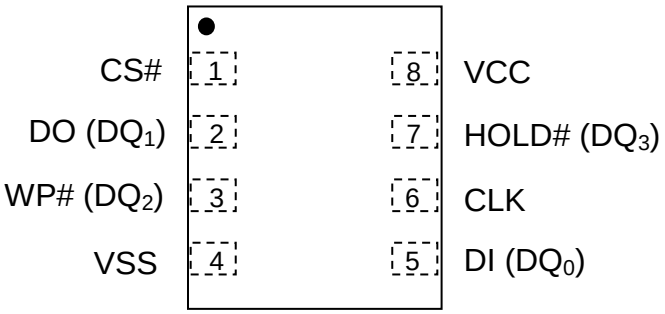
The device also offers a sophisticated method for protecting individual blocks against erroneous or malicious program and erase operations. By providing the ability to individually protect and unprotect blocks, a system can unprotect a specific block to modify its contents while keeping the remaining blocks of the memory array securely protected. This is useful in applications where program code is patched or updated on a subroutine or module basis or in applications where data storage segments need to be modified without running the risk of errant modifications to the program code segments.

The device is designed to allow either single Sector/Block at a time or full chip erase operation. The device can be configured to protect part of the memory as the software protected mode. The device can sustain a minimum of 100K program/erase cycles on each sector or block.

## CONNECTION DIAGRAMS (TOP VIEW)




8 - LEAD SOP




8 – LEAD USON

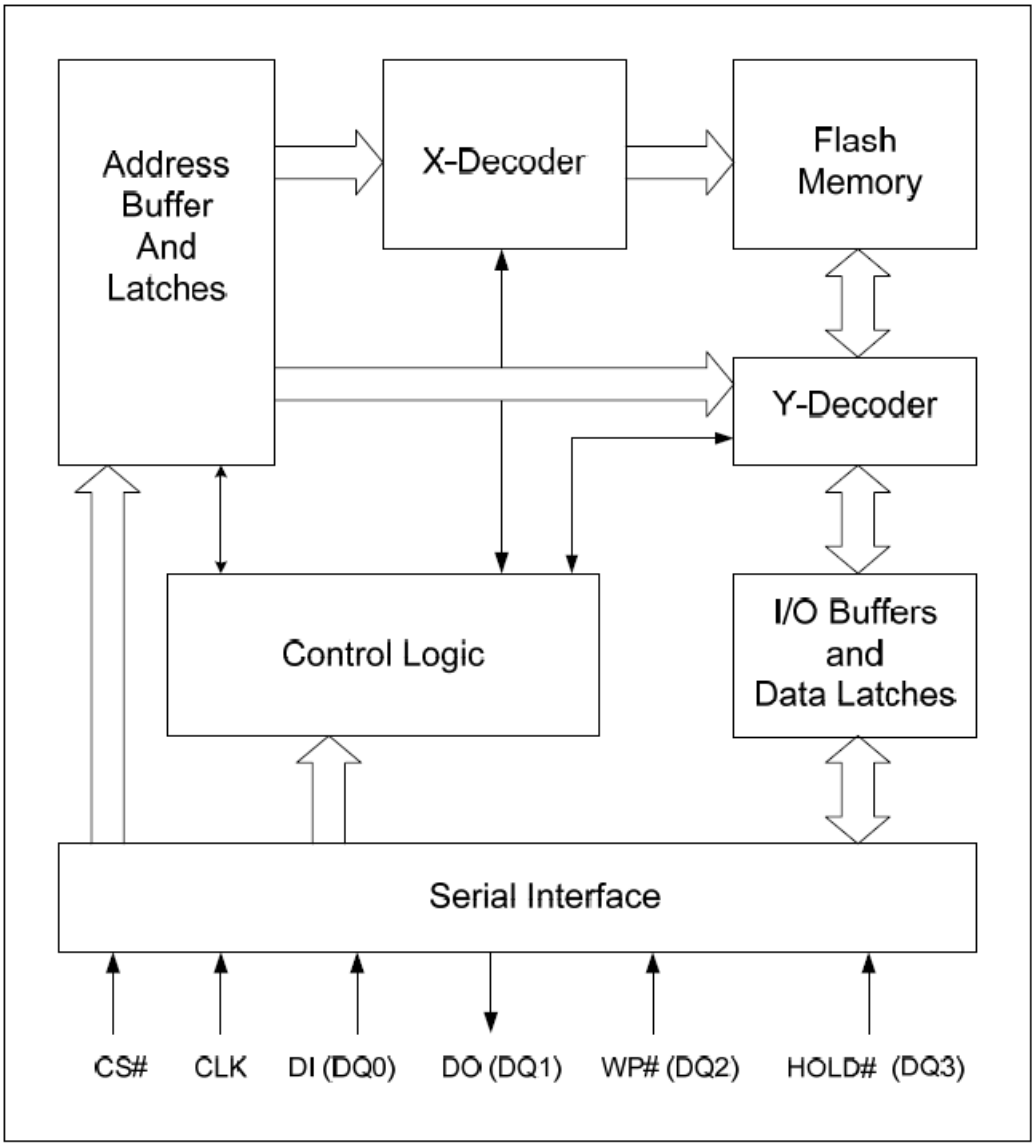
### Pin Names

| Symbol | Pin Name |
|---|---|
| CLK | Serial Clock Input |
| DI ($DQ_0$) | Serial Data Input (Data Input Output 0) *1 |
| DO ($DQ_1$) | Serial Data Output (Data Input Output 1) *1 |
| CS# | Chip Enable |
| WP# ($DQ_2$) | Write Protect (Data Input Output 2) *2 |
| HOLD# ($DQ_3$) | HOLD# pin (Data Input Output 3) *2 |
| $V_{CC}$ | Supply Voltage (2.3-3.6V) |
| $V_{SS}$ | Ground |
| NC | No Connect |

**Note:**

1. $DQ_0$ and $DQ_1$ are used for Dual and Quad instructions.
2. $DQ_0$ ~ $DQ_3$ are used for Quad instructions.
   WP# & HOLD# functions are only available for Standard/Dual SPI.

## BLOCK DIAGRAM




**Note:**

1. $DQ_0$ and $DQ_1$ are used for Dual instructions.
2. $DQ_0$ ~ $DQ_3$ are used for Quad instructions.

## SIGNAL DESCRIPTION

### Serial Data Input, Output and IOs (DI, DO and $DQ_0$, $DQ_1$, $DQ_2$, $DQ_3$)

The device support standard SPI, Dual SPI and Quad SPI operation. Standard SPI instructions use the unidirectional DI (input) pin to serially write instructions, addresses or data to the device on the rising edge of the Serial Clock (CLK) input pin. Standard SPI also uses the unidirectional DO (output) to read data or status from the device on the falling edge CLK.

Dual and Quad SPI instruction use the bidirectional IO pins to serially write instruction, addresses or data to the device on the rising edge of CLK and read data or status from the device on the falling edge of CLK.

### Serial Clock (CLK)

The SPI Serial Clock Input (CLK) pin provides the timing for serial input and output operations. ("See SPI Mode")

### Chip Select (CS#)

The SPI Chip Select (CS#) pin enables and disables device operation. When CS# is high the device is deselected and the Serial Data Output (DO, or $DQ_0$, $DQ_1$, $DQ_2$ and $DQ_3$) pins are at high impedance. When deselected, the devices power consumption will be at standby levels unless an internal erase, program or status register cycle is in progress. When CS# is brought low the device will be selected, power consumption will increase to active levels and instructions can be written to and data read from the device. After power-up, CS# must transition from high to low before a new instruction will be accepted.

### Write Protect (WP#)

The Write Protect (WP#) pin can be used to prevent the Status Register from being written. Used in conjunction with the Status Register's Block Protect (4KBL, TB, BP2, BP1 and BP0) bits and Status Register Protect (SRP and SRP1) bits, a portion or the entire memory array can be hardware protected. The WP# function is only available for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO ($DQ_2$) for Quad I/O operation.

### HOLD (HOLD#)

The HOLD# pin allows the device to be paused while it is actively selected. When QE bit is "0" (factory default), the HOLD# pin is enabled. When HOLD# is brought low, while CS# is low, the DO pin will be at high impedance and signals on the DI and CLK pins will be ignored (don't care). The hold function can be useful when multiple devices are sharing the same SPI signals. The HOLD# function is only available for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO ($DQ_3$) for Quad I/O operation.

## MEMORY ORGANIZATION

The memory is organized as:

- 1,048,576 bytes
- Uniform Sector Architecture
  - 16 blocks of 64-Kbyte
  - 32 sectors of 32-Kbyte
  - 256 sectors of 4-Kbyte
  - 4096 pages (256 bytes each)

Each page can be individually programmed (bits are programmed from 1 to 0). The device is Sector, Block or Chip Erasable but not Page Erasable.

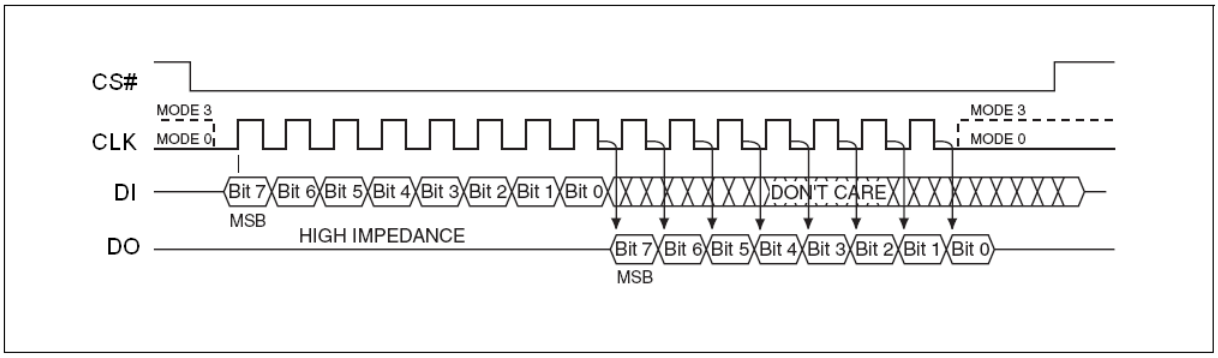
## Uniform Block Sector Architecture

| 64KB Block | 32KB Block | Sector | Address range | |
|---|---|---|---|---|
| 15 | 31 | 255 | 00FF000h | 00FFFFFh |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | 30 | 240 | 00F0000h | 00F0FFFh |
| 14 | 29 | 239 | 00EF000h | 00EFFFFh |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | 28 | 224 | 00E0000h | 00E0FFFh |
| 13 | 27 | 223 | 00DF000h | 00DFFFFh |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | 26 | 208 | 00D0000h | 00D0FFFh |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 2 | 5 | 47 | 002F000h | 002FFFFh |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | 4 | 32 | 0020000h | 0020FFFh |
| 1 | 3 | 31 | 001F000h | 001FFFFh |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | 2 | 16 | 0010000h | 0010FFFh |
| 0 | 1 | 15 | 000F000h | 000FFFFh |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | 0 | 0 | 0000000h | 0000FFFh |

## OPERATING FEATURES

### Standard SPI Modes

The device is accessed through an SPI compatible bus consisting of four signals: Serial Clock (CLK), Chip Select (CS#), Serial Data Input (DI) and Serial Data Output (DO). Both SPI bus operation Modes 0 (0, 0) and 3 (1, 1) are supported. The primary difference between Mode 0 and Mode 3, as shown in SPI Modes figure, concerns the normal state of the CLK signal when the SPI bus master is in standby and data is not being transferred to the Serial Flash. For Mode 0 the CLK signal is normally low. For Mode 3 the CLK signal is normally high. In either case data input on the DI pin is sampled on the rising edge of the CLK. Data output on the DO pin is clocked out on the falling edge of CLK.



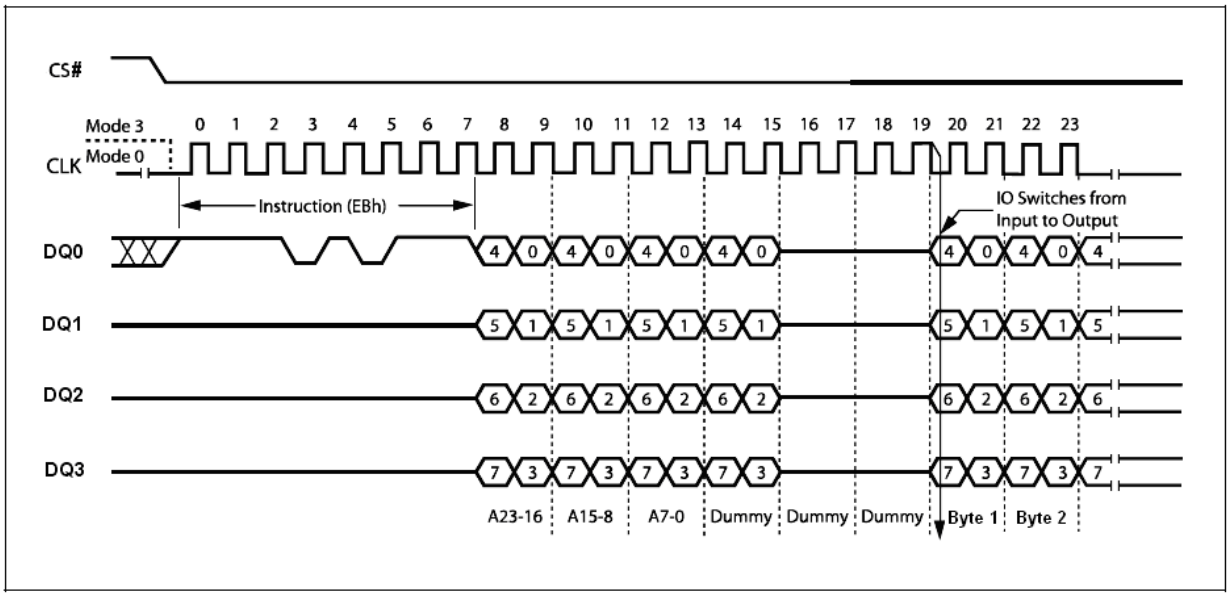

**SPI Modes**

### Dual SPI Instruction

The device supports Dual SPI operation when using the “Dual Output Fast Read and Dual I/ O FAST_READ” (3Bh and BBh) instructions. These instructions allow data to be transferred to or from the Serial Flash memory at two to three times the rate possible with the standard SPI. The Dual Read instructions are ideal for quickly downloading code from Flash to RAM upon power-up (code-shadowing) or for application that cache code-segments to RAM for execution. The Dual output feature simply allows the SPI input pin to also serve as an output during this instruction. When using Dual SPI instructions the DI and DO pins become bidirectional I/O pins; $DQ_0$ and $DQ_1$. All other operations use the standard SPI interface with single output signal.

### Quad I/O SPI Modes

The device supports Quad output operation when using the Quad I/O Fast Read (EBh).This instruction allows data to be transferred to or from the Serial Flash memory at four to six times the rate possible with the standard SPI. The Quad Read instruction offer a significant improvement in continuous and random access transfer rates allowing fast code-shadowing to RAM or for application that cache code-segments to RAM for execution.




**Quad I/O SPI Modes**

### Page Programming

To program one data byte, two instructions are required: Write Enable (WREN), which is one byte, and a Page Program (PP) or Quad Input Page Program (QPP) sequence, which consists of four bytes plus data. This is followed by the internal Program cycle (of duration $t_{PP}$).

To spread this overhead, the Page Program (PP) or Quad Input Page Program (QPP) instruction allows up to 256 bytes to be programmed at a time (changing bits from 1 to 0) provided that they lie in consecutive addresses on the same page of memory.

### Sector Erase, Half Block Erase, Block Erase and Chip Erase

The Page Program (PP) or Quad Input Page Program (QPP) instruction allows bits to be reset from 1 to 0. Before this can be applied, the bytes of memory need to have been erased to all 1s (FFh). This can be achieved a sector at a time, using the Sector Erase (SE) instruction, half a block at a time using the Half Block Erase (HBE) instruction, a block at a time using the Block Erase (BE) instruction or throughout the entire memory, using the Chip Erase (CE) instruction. This starts an internal Erase cycle (of duration $t_{SE}$, $t_{HBE}$, $t_{BE}$ or $t_{CE}$). The Erase instruction must be preceded by a Write Enable (WREN) instruction.

### Polling During a Write, Program or Erase Cycle

A further improvement in the time to Write Status Register (WRSR, WRSR2, WRSR3), Program (PP, QPP) or Erase (SE, HBE, BE or CE) can be achieved by not waiting for the worst case delay ($t_W$, $t_{PP}$, $t_{SE}$, $t_{HBE}$, $t_{BE}$ or $t_{CE}$). The Write In Progress (WIP) bit is provided in the Status Register so that the application program can monitor its value, polling it to establish when the previous Write cycle, Program cycle or Erase cycle is complete.

### Active Power, Stand-by Power and Deep Power-Down Modes

When Chip Select (CS#) is Low, the device is enabled, and in the Active Power mode. When Chip Select (CS#) is High, the device is disabled, but could remain in the Active Power mode until all internal cycles have completed (Program, Erase, Write Status Register). The device then goes into the Stand-by Power mode. The device consumption drops to $I_{CC1}$.

The Deep Power-down mode is entered when the specific instruction (the Enter Deep Power-down Mode (DP) instruction) is executed. The device consumption drops further to $I_{CC2}$. The device remains in this mode until another specific instruction (the Release from Deep Power-down Mode and Read Device ID (RDI) instruction) is executed.

All other instructions are ignored while the device is in the Deep Power-down mode. This can be used as an extra software protection mechanism, when the device is not in active use, to protect the device from inadvertent Write, Program or Erase instructions.

## Write Protection

Applications that use non-volatile memory must take into consideration the possibility of noise and other adverse system conditions that may compromise data integrity. To address this concern, the device provides the following data protection mechanisms:

- Power-On Reset and an internal timer ($t_{PUW}$) can provide protection against inadvertent changes while the power supply is outside the operating specification.
- Program, Erase and Write Status Register instructions are checked that they consist of a number of clock pulses that is a multiple of eight, before they are accepted for execution.
- All instructions that modify data must be preceded by a Write Enable (WREN) instruction to set the Write Enable Latch (WEL) bit. This bit is returned to its reset state by the following events:
  – Power-up
  – Write Disable (WRDI) instruction completion or Write Status Register (WRSR) instruction completion, Write Status Register 2(WRSR2) instruction completion or Write Status Register 3 (WRSR3) instruction completion or Page Program (PP) or Quad Input Page Program (QPP) instruction completion or Sector Erase (SE) instruction completion or Half Block Erase (HBE) / Block Erase (BE) instruction completion or Chip Erase (CE) instruction completion
- The Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits allow part of the memory to be configured as read-only. This is the Software Protected Mode (SPM).
- The Write Protect (WP#) signal allows the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits and Status Register Protect (SRP and SRP1) bit to be protected. This is the Hardware Protected Mode (HPM).
- In addition to the low power consumption feature, the Deep Power-down mode offers extra software protection from inadvertent Write, Program and Erase instructions, as all instructions are ignored except one particular instruction (the Release from Deep Power-down instruction).

## Protected Area Sizes Sector Organization

| Status Register Content [1] | | | | | | Memory Content [2] | | | |
|---|---|---|---|---|---|---|---|---|---|
| CMP | 4KBL | TB | BP2 | BP1 | BP0 | Protect Areas | Addresses | Density | Portion |
| 0 | X | X | 0 | 0 | 0 | None | None | None | None |
| 0 | 0 | 0 | 0 | 0 | 1 | Block 15 | 0F0000h-0FFFFFh | 64KB | Upper 1/16 |
| 0 | 0 | 0 | 0 | 1 | 0 | Block 14 to 15 | 0E0000h-0FFFFFh | 128KB | Upper 1/8 |
| 0 | 0 | 0 | 0 | 1 | 1 | Block 12 to 15 | 0C0000h-0FFFFFh | 256KB | Upper 1/4 |
| 0 | 0 | 0 | 1 | 0 | 0 | Block 8 to 15 | 080000h-0FFFFFh | 512KB | Upper 1/2 |
| 0 | 0 | 1 | 0 | 0 | 1 | Block 0 | 000000h-00FFFFh | 64KB | Lower 1/16 |
| 0 | 0 | 1 | 0 | 1 | 0 | Block 0 to 1 | 000000h-01FFFFh | 128KB | Lower 1/8 |
| 0 | 0 | 1 | 0 | 1 | 1 | Block 0 to 3 | 000000h-03FFFFh | 256KB | Lower 1/4 |
| 0 | 0 | 1 | 1 | 0 | 0 | Block 0 to 7 | 000000h-07FFFFh | 512KB | Lower 1/2 |
| 0 | 0 | x | 1 | 0 | 1 | Block 0 to 15 | 000000h-0FFFFFh | 1MB | All |
| 0 | 0 | x | 1 | 1 | x | Block 0 to 15 | 000000h-0FFFFFh | 1MB | All |
| 0 | 1 | 0 | 0 | 0 | 1 | Block 15 | 0FF000h-0FFFFFh | 4KB | Upper 1/256 |
| 0 | 1 | 0 | 0 | 1 | 0 | Block 15 | 0FE000h-0FFFFFh | 8KB | Upper 1/128 |
| 0 | 1 | 0 | 0 | 1 | 1 | Block 15 | 0FC000h-0FFFFFh | 16KB | Upper 1/64 |
| 0 | 1 | 0 | 1 | 0 | X | Block 15 | 0F8000h-0FFFFFh | 32KB | Upper 1/32 |
| 0 | 1 | 1 | 0 | 0 | 1 | Block 0 | 000000h-000FFFh | 4KB | Lower 1/256 |
| 0 | 1 | 1 | 0 | 1 | 0 | Block 0 | 000000h-001FFFh | 8KB | Lower 1/128 |
| 0 | 1 | 1 | 0 | 1 | 1 | Block 0 | 000000h-003FFFh | 16KB | Lower 1/64 |
| 0 | 1 | 1 | 1 | 0 | X | Block 0 | 000000h-007FFFh | 32KB | Lower 1/32 |
| 0 | 1 | X | 1 | 1 | X | Block 0 to 15 | 000000h-0FFFFFh | 1MB | All |
| 1 | X | X | 0 | 0 | 0 | Block 0 to 15 | 000000h-0FFFFFh | 1MB | All |
| 1 | 0 | 0 | 0 | 0 | 1 | Block 0 to 14 | 000000h-0EFFFFh | 960KB | Lower 15/16 |
| 1 | 0 | 0 | 0 | 1 | 0 | Block 0 to 13 | 000000h-0DFFFFh | 896KB | Lower 7/8 |
| 1 | 0 | 0 | 0 | 1 | 1 | Block 0 to 11 | 000000h-0BFFFFh | 768KB | Lower 3/4 |
| 1 | 0 | 0 | 1 | 0 | 0 | Block 0 to 7 | 000000h-07FFFFh | 512KB | Lower 1/2 |
| 1 | 0 | 1 | 0 | 0 | 1 | Block 1 to 15 | 010000h-0FFFFFh | 960KB | Upper 15/16 |
| 1 | 0 | 1 | 0 | 1 | 0 | Block 2 to 15 | 020000h-0FFFFFh | 896KB | Upper 7/8 |
| 1 | 0 | 1 | 0 | 1 | 1 | Block 4 to 15 | 040000h-0FFFFFh | 768KB | Upper 3/4 |
| 1 | 0 | 1 | 1 | 0 | 0 | Block 8 to 15 | 080000h-0FFFFFh | 512KB | Upper 1/2 |
| 1 | 0 | x | 1 | 0 | 1 | None | None | None | None |
| 1 | 0 | x | 1 | 1 | x | None | None | None | None |
| 1 | 1 | 0 | 0 | 0 | 1 | Block 0 to 15 | 000000h-0FEFFFh | 1020KB | Lower 255/256 |
| 1 | 1 | 0 | 0 | 1 | 0 | Block 0 to 15 | 000000h-0FDFFFh | 1016KB | Lower 127/128 |
| 1 | 1 | 0 | 0 | 1 | 1 | Block 0 to 15 | 000000h-0FBFFFh | 1008KB | Lower 63/64 |
| 1 | 1 | 0 | 1 | 0 | X | Block 0 to 15 | 000000h-0F7FFFh | 992KB | Lower 31/32 |
| 1 | 1 | 1 | 0 | 0 | 1 | Block 0 to 15 | 001000h-0FFFFFh | 1020KB | Upper 255/256 |
| 1 | 1 | 1 | 0 | 1 | 0 | Block 0 to 15 | 002000h-0FFFFFh | 1016KB | Upper 127/128 |
| 1 | 1 | 1 | 0 | 1 | 1 | Block 0 to 15 | 004000h-0FFFFFh | 1008KB | Upper 63/64 |
| 1 | 1 | 1 | 1 | 0 | X | Block 0 to 15 | 008000h-0FFFFFh | 992KB | Upper 31/32 |
| 1 | 1 | X | 1 | 1 | X | None | None | None | None |

**Note:**

1. X = don't care
2. If any Erase or Program command specifies a memory region that contains protected data portion, this command will be ignored.

## INSTRUCTIONS

All instructions, addresses and data are shifted in and out of the device, most significant bit first. Serial Data Input (DI) is sampled on the first rising edge of Serial Clock (CLK) after Chip Select (CS#) is driven Low. Then, the one-byte instruction code must be shifted in to the device, most significant bit first, on Serial Data Input (DI), each bit being latched on the rising edges of Serial Clock (CLK).

The instruction set is listed in Instruction Set table. Every instruction sequence starts with a one-byte instruction code. Depending on the instruction, it might be followed by address bytes, or data bytes, or both or none. Chip Select (CS#) must be driven High after the last bit of the instruction sequence has been shifted in. In the case of a Read Data Bytes (READ), Read Data Bytes at Higher Speed (Fast_Read), Dual Output Fast Read (3Bh), Dual I/O Fast Read (BBh), Quad Output Fast Read (6Bh), Quad Input/Output FAST_READ (EBh), Read Status Register (RDSR), Read Status Register 2 (RDSR2), Read Status Register 3 (RDSR3) or Release from Deep Power-down, and Read Device ID (RDI) instruction, the shifted-in instruction sequence is followed by a data-out sequence. Chip Select (CS#) can be driven High after any bit of the data-out sequence is being shifted out.

In the case of a write instruction, Chip Select (CS#) must be driven High exactly at a byte boundary, otherwise the instruction is rejected, and is not executed. That is, Chip Select (CS#) must driven High when the number of clock pulses after Chip Select (CS#) being driven Low is an exact multiple of eight. For Page Program, if at any time the input byte is not a full byte, nothing will happen and WEL will not be reset.

**In the case of multi-byte commands of Page Program (PP), Quad Input Page Program (QPP) and Release from Deep Power Down (RES) minimum number of bytes specified has to be given, without which, the command will be ignored.**

**In the case of Page Program, if the number of byte after the command is less than 4 (at least 1 data byte), it will be ignored too. In the case of SE and HBE/BE, exact 24-bit address is a must, any less or more will cause the command to be ignored.**

All attempts to access the memory array during a Write Status Register cycle, Program cycle or Erase cycle are ignored, and the internal Write Status Register cycle, Program cycle or Erase cycle continues unaffected.

**Instruction Set**

| Instruction Name | Byte 1 Code | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | n-Bytes |
|---|---|---|---|---|---|---|---|
| RSTEN | 66h | | | | | | |
| RST (1) | 99h | | | | | | |
| Write Enable (WREN) | 06h | | | | | | |
| Volatile Status Register Write Enable (3) | 50h | | | | | | |
| Write Disable (WRDI) | 04h | | | | | | |
| Read Status Register (RDSR) | 05h | (S7-S0) (4) | | | | | continuous (5) |
| Read Status Register 2 (RDSR2) | 09h /35h | (S2.7-S2.0) (4) | | | | | continuous (5) |
| Read Status Register 3 (RDSR3) | 95h /15h | (S3.7-S3.0) (4) | | | | | |
| Write Status Register (WRSR) | 01h | S7-S0 | (S2.7-S2.0) | (S3.7-S3.0) | | | |
| Write Status Register 2 (WRSR2) | 31h | S2.7-S2.0 | | | | | |
| Write Status Register 3 (WRSR3) | C0h /11h | S3.7-S3.0 | | | | | |
| Write Suspend | 75h/B0h | | | | | | |
| Write Resume | 7Ah/30h | | | | | | |
| Deep Power-down | B9h | | | | | | |
| Release from Deep Power-down, and read Device ID (RES) | ABh | dummy | dummy | dummy | (ID7-ID0) | | (6) |
| Release from Deep Power-down (RDP) | | | | | | | |
| Manufacturer/ Device ID | 90h | dummy | dummy | 00h | (M7-M0) | (ID7-ID0) | (7) |
| | | | | 01h | (ID7-ID0) | (M7-M0) | |
| Read Identification (RDID) | 9Fh | (M7-M0) | (ID15-ID8) | (ID7-ID0) | (8) | | |
| Read SFDP mode | 5Ah | A23-A16 | A15-A8 | A7-A0 | dummy | (D7-D0) | (Next Byte) continuous |
| Read Unique ID | 4Bh | 00h | 00h | 00h | dummy | (UID7-UID0) | continuous |
| Erase Security Registers | 44h | A23-A16 | A15-A8 | A7-A0 | | | |
| Program Security Registers | 42h | A23-A16 | A15-A8 | A7-A0 | (D7-D0) | (Next Byte) | continuous |
| Read Security Registers | 48h | A23-A16 | A15-A8 | A7-A0 | dummy | (D7-D0) | (Next Byte) continuous |
| Set Burst with Wrap | 77h | dummy | dummy | dummy | W7-W0 | | |

**Note:**

1. RST command only executed if RSTEN command is executed first. Any intervening command will disable Reset.
2. Volatile Status Register Write Enable command must precede WRSR command without any intervening commands to write data to Volatile Status Register.
3. Data bytes are shifted with Most Significant Bit first. Byte fields with data in parenthesis "( )" indicate data being read from the device on the DO pin.
4. The Status Register contents will repeat continuously until CS# terminates the instruction.
5. The Device ID will repeat continuously until CS# terminates the instruction.
6. The Manufacturer ID and Device ID bytes will repeat continuously until CS# terminates the instruction. 00h on Byte 4 starts with MID and alternate with DID, 01h on Byte 4 starts with DID and alternate with MID.
7. (M7-M0): Manufacturer, (ID15-ID8): Memory Type, (ID7-ID0): Memory Capacity.

### Instruction Set (Read Instruction)

| Instruction Name | OP Code | Address bits | Dummy bits / Clocks (Default) | Data Out | Remark |
|---|---|---|---|---|---|
| Read Data | 03h | 24 bits | 0 | (D7-D0, …) | (Next Byte) continuous |
| Fast Read | 0Bh | 24 bits | 8 bits / 8 clocks | (D7-D0, …) | (Next Byte) continuous |
| Dual Output Fast Read | 3Bh | 24 bits | 8 bits / 8 clocks | (D7-D0, …) | (one byte Per 4 clocks, continuous) |
| Dual I/O Fast Read | BBh | 24 bits | 8 bits / 4 clocks | (D7-D0, …) | (one byte Per 4 clocks, continuous) |
| Quad Output Fast Read | 6Bh | 24 bits | 8 bits / 8 clocks | (D7-D0, …) | (one byte per 2 clocks, continuous) |
| Quad I/O Fast Read | EBh | 24 bits | 24 bits / 6 clocks | (D7-D0, …) | (one byte per 2 clocks, continuous) |

### Instruction Set (Program Instruction)

| Instruction Name | OP Code | Address bits | Dummy bits / Clocks (Default) | Data Out | Remark |
|---|---|---|---|---|---|
| Page Program (PP) | 02h | 24 bits | 0 | (D7-D0, …) | (Next Byte) continuous |
| Quad Input Page Program (QPP) | 32h | 24 bits | 0 | (D7-D0, …) | (one byte per 2 clocks, continuous) |

### Instruction Set (Erase Instruction)

| Instruction Name | OP Code | Address bits | Dummy bits / Clocks (Default) | Data Out | Remark |
|---|---|---|---|---|---|
| Sector Erase (SE) | 20h | 24 bits | 0 | - | |
| 32K Half Block Erase (HBE) | 52h | 24 bits | 0 | - | |
| 64K Block Erase (BE) | D8h | 24 bits | 0 | - | |
| Chip Erase (CE) | C7h/ 60h | - | 0 | - | |

### Manufacturer and Device Identification

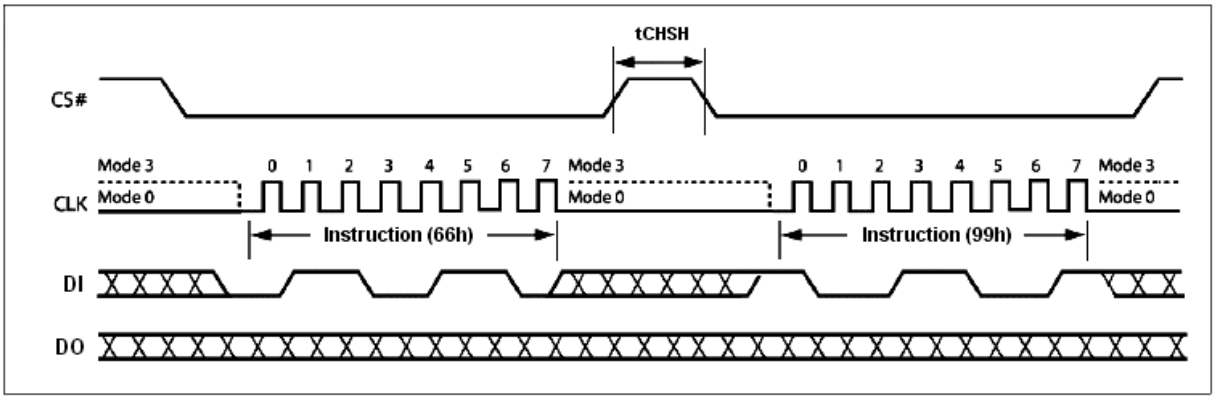
| OP Code | (M7-M0) | (ID15-ID0) | (ID7-ID0) |
|---|---|---|---|
| ABh | | | 13h |
| 90h | 1Ch | | 13h |
| 9Fh | 1Ch | 4114h | |

### Reset-Enable (RSTEN) (66h) and Reset (RST) (99h)

The Reset operation is used as a system (software) reset that puts the device in normal operating Ready mode and lost all the current volatile setting, such as Volatile Status Register bits, Write Enable Latch status (WEL) Program/Erase Suspend status, Read Parameter (P7-0) Deep Power Down Mode, Continuous Read Mode bit setting (M7-0) and Wrap Bit Setting (W6-4). This operation consists of two commands: Reset-Enable (RSTEN) and Reset (RST).

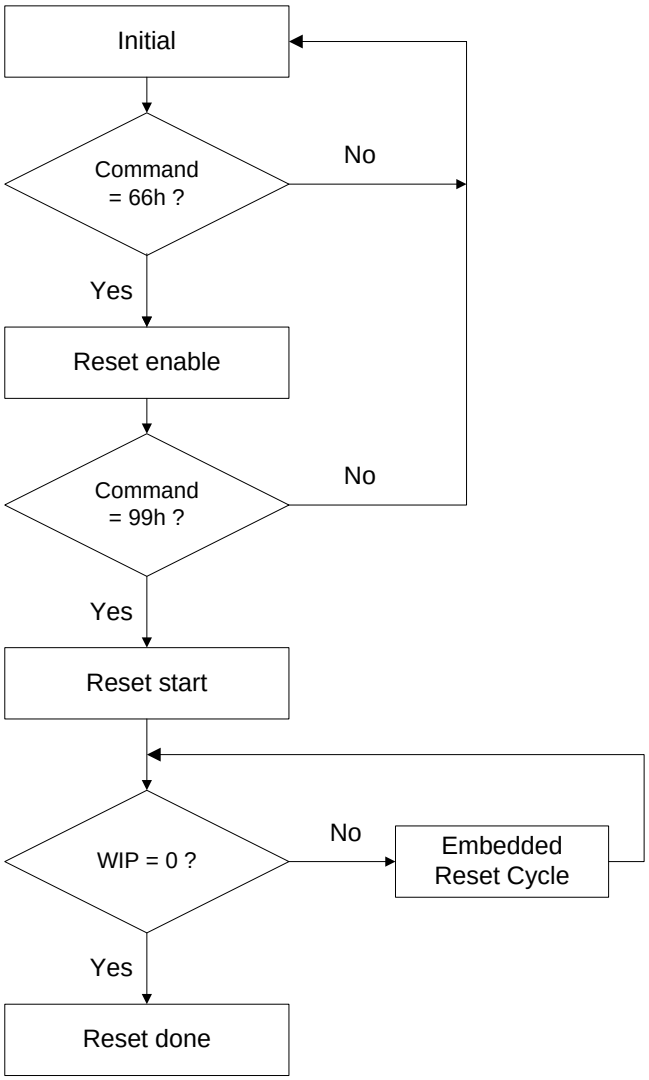
To reset the device the host drives CS# low, sends the Reset-Enable command (66h), and drives CS# high. Next, the host drives CS# low again, sends the Reset command (99h), and drives CS# high.
The Reset operation requires the Reset-Enable command followed by the Reset command. Any command other than the Reset command after the Reset-Enable command will disable the Reset-Enable.

A successful command execution will reset the status register, see Reset-Enable and Reset Sequence Diagram figure for SPI Mode. A device reset during an active Program or Erase operation aborts the operation, which can cause the data of the targeted address range to be corrupted or lost. Depending on the prior operation, the reset timing may vary. Recovery from a Write operation requires more software latency time ($t_{SR}$) than recovery from other operations.




**Reset-Enable and Reset Sequence Diagram**

**Software Reset Flow**



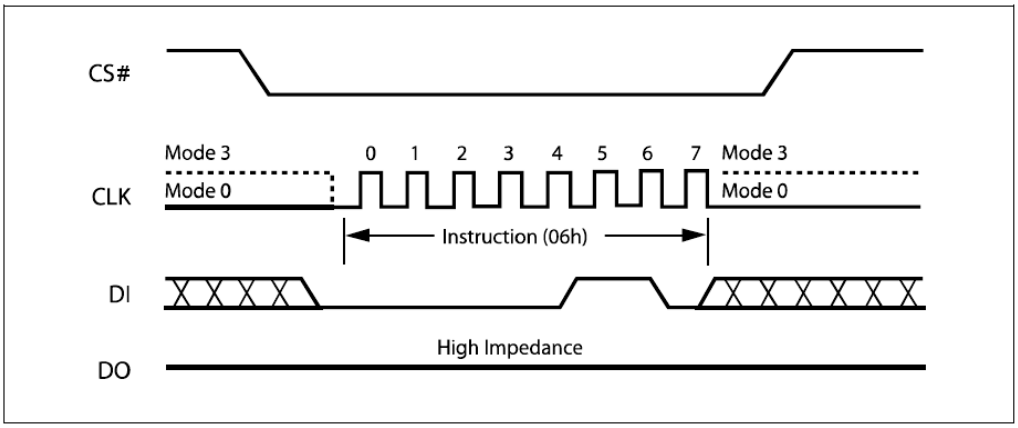

**Note:**

1. Reset-Enable (RSTEN) (66h) and Reset (RST) (99h) commands need to match standard SPI.
2. The reset command could be executed during embedded program and erase process.
3. This flow can release the device from Deep power down mode.
4. The Status Register Bit and Status Register 2 Bit will reset to default value after reset done.
5. If user reset device during erase, the embedded reset cycle software reset latency will take about 28us in worst case.

### Write Enable (WREN) (06h)

The Write Enable (WREN) instruction (Write Enable Instruction Sequence Diagram figure) sets the Write Enable Latch (WEL) bit. The Write Enable Latch (WEL) bit must be set prior to every Page Program (PP), Quad Input Page Program (QPP), Sector Erase (SE), Block Erase (HBE/BE), Chip Erase (CE) and Write Status Register (WRSR/ WRSR3) instruction.

The Write Enable (WREN) instruction is entered by driving Chip Select (CS#) Low, sending the instruction code, and then driving Chip Select (CS#) High.

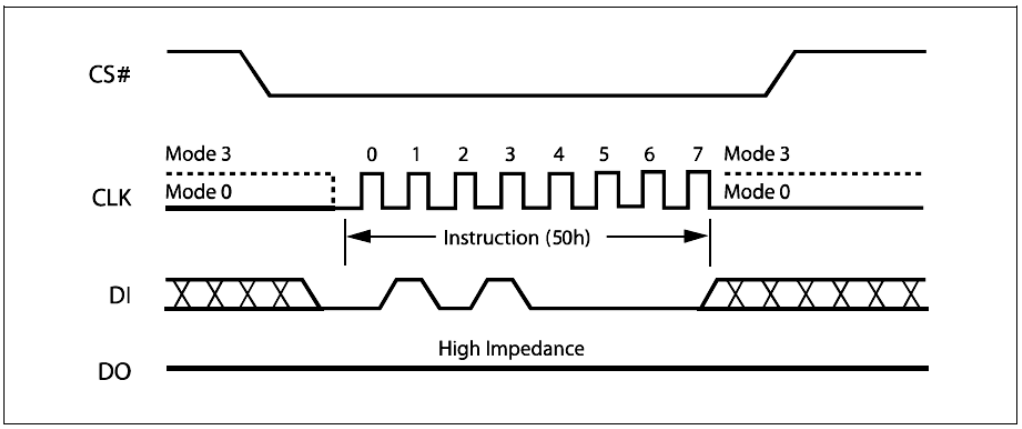



**Write Enable Instruction Sequence Diagram**

### Volatile Status Register Write Enable (50h)

This feature enable user to change memory protection schemes quickly without waiting for the typical non-volatile bit write cycles or affecting the endurance of the Status Register non-volatile bits. The Volatile Status Register Write Enable (50h) command won't set the Write Enable Latch (WEL) bit, it is only valid for 'Write Status Register' command to change the Volatile Status Register bit values.

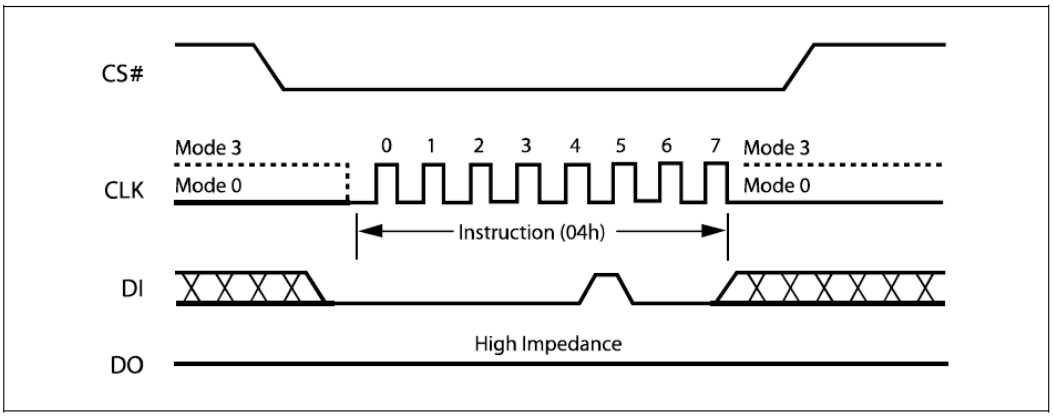
To write to Volatile Status Register, issue the Volatile Status Register Write Enable (50h) command prior issuing WRSR (01h). The Status Register bits will be refresh to Volatile Status Register (SR[7:2]) within $t_{SHSL2}$ (50ns). Upon power off or the execution of a Software/Hardware Reset, the volatile Status Register bit values will be lost, and the non-volatile Status Register bit values will be restored. The instruction sequence is shown in Volatile Status Register Write Enable Instruction Sequence Diagram figure.




**Volatile Status Register Write Enable Instruction Sequence Diagram**

### Write Disable (WRDI) (04h)

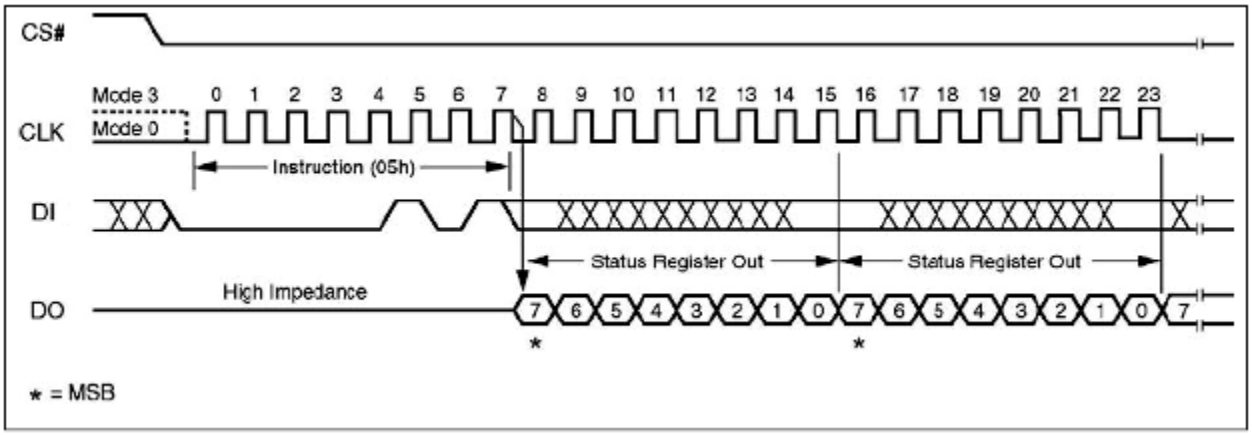
The Write Disable instruction (Write Disable Instruction Sequence Diagram figure) resets the Write Enable Latch (WEL) bit in the Status Register to a 0. The Write Disable instruction is entered by driving Chip Select (CS#) low, shifting the instruction code “04h” into the DI pin and then driving Chip Select (CS#) high. Note that the WEL bit is automatically reset after Power-up and upon completion of the Write Status Register, Page Program, Sector Erase, Block Erase (HBE/BE) and Chip Erase instructions.




**Write Disable Instruction Sequence Diagram**

### Read Status Register (RDSR) (05h)

The Read Status Register (RDSR) instruction 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 instruction to the device. It is also possible to read the Status Register continuously, as shown in Read Status Register Instruction Sequence Diagram figure.




**Read Status Register Instruction Sequence Diagram**

**Status Register Bit Locations**

| SR7 | SR6 | SR5 | SR4 | SR3 | SR2 | SR1 | SR0 |
|---|---|---|---|---|---|---|---|
| **SRP** (Status Register Protect) | **4KBL** (4KB boot lock) | **TB** (Top / Bottom Protect) | **BP2** (Block Protected) | **BP1** (Block Protected) | **BP0** (Block Protected) | **WEL** (Write Enable Latch) | **WIP** (Write In Progress) |
| 1 = status register write disable | 1 = Sector 0 = 64KB Block (default 0) | 1 = Bottom 0 = Top (default 0) | (note 2) | (note 2) | (note 2) | 1 = write enable 0 = not write enable | 1 = write operation 0 = not in write operation |
| Volatile bit / Non-volatile bit | Volatile bit / Non-volatile bit | Volatile bit / Non-volatile bit | Volatile bit / Non-volatile bit | Volatile bit / Non-volatile bit | Volatile bit / Non-volatile bit | Read only bit | Read only bit |

**Note:**
1. See the “Protected Area Sizes Sector Organization” table.

The status and control bits of the Status Register are as follows:

**WIP bit.** The Write In Progress (WIP) bit indicates whether the memory is busy with a Write Status Register, Program or Erase cycle. When set to 1, such a cycle is in progress, when reset to 0 no such cycle is in progress.

**WEL bit.** The Write Enable Latch (WEL) bit indicates the status of the internal Write Enable Latch. When set to 1 the internal Write Enable Latch is set, when set to 0 the internal Write Enable Latch is reset and no Write Status Register, Program or Erase instruction is accepted.

**BP2, BP1, BP0 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 instructions. These bits are written with the Write Status Register (WRSR) instruction. When one or both of the Block Protect (BP2, BP1, BP0) bits is set to 1, the relevant memory area (as defined in Protected Area Sizes Sector Organization table.) becomes protected against Page Program (PP), Quad Input Page Program (QPP), Sector Erase (SE) and Block Erase (HBE/BE) instructions. The Block Protect (BP2, BP1, BP0) bits can be written and provided that the Hardware Protected mode has not been set. The Chip Erase (CE) instruction is executed if all memory regions aren’t protected by the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits and EBL bit is 0.

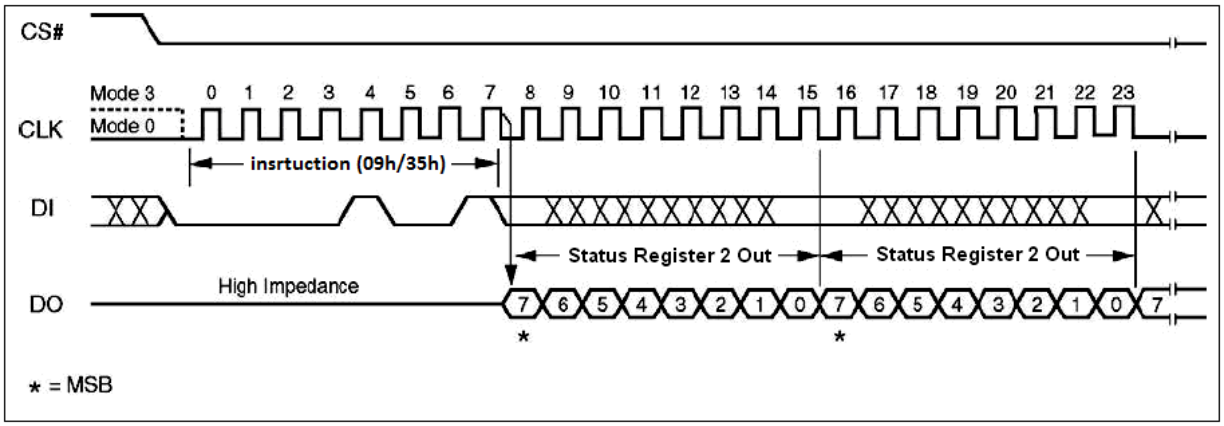
**TB bit.** The Top/Bottom Protect Bit (TB) controls if the Block Protect Bits (BP2, BP1, BP0) protect from the Top (TB = 0) or the Bottom (TB = 1) of the array as shown in the Protected Area Sizes Sector Organization table. The factory default setting is TB = 0. The TB bit can be set with the Write Status Register instruction.

**4KBL bit.** The 4KB Boot Lock bit (4KBL) is set by WRSR command. 4KBL also controls Block Protect table, please refer to Protected Area Sizes Sector Organization table.

**SRP bit.** The Status Register Protect (SRP) bit is operated in conjunction with the Write Protect (WP#) signal. The Status Register Write Protect (SRP) bit and Write Protect (WP#) signal allow the device to be put in the Hardware Protected mode (when the Status Register Protect (SRP) bit is set to 1, and Write Protect (WP#) is driven Low). In this mode, the bits of the Status Register (4KBL, TB, BP2, BP1, BP0) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

### Read Status Register 2 (RDSR2) (09h/35h)

The Read Status Register 2 (RDSR2) instruction allows the Status Register 2 to be read. The Status Register 2 may be read at any time, even while a Write Suspend or Write Resume cycle is in progress, OTP array lock bits, QE bits enable or not and BP protection table related bit: CMP. When one of these cycles is in progress, it is recommended to check the Write In Progress (WIP) bit before sending a new instruction to the device. It is also possible to read the Status Register 2 continuously, as shown in Read Status Register 2 Instruction Sequence Diagram figure.




**Read Status Register 2 Instruction Sequence Diagram**

**Status Register 2 Bit Locations**

| SR2.7 | SR2.6 | SR2.5 | SR2.4 | SR2.3 | SR2.2 | SR2.1 | SR2.0 |
|---|---|---|---|---|---|---|---|
| WSE (Write Suspend Erase status bit) | CMP bit | SPL0 bit | SPL1 bit | SPL2 bit | WSP (Write Suspend Program bits) | QE | reserved |
| 1 = Erase suspended 0 = Erase is not suspended | (note 2) | 1 = OTP1 sector is protected | 1 = OTP2 sector is protected | 1 = OTP3 sector is protected | 1 = Program suspended 0 = Program is not suspended | 1 = WP# and HOLD#/RESET# disable 0 = WP# and HOLD#/RESET# enable (default 0) | |
| Indicator bit | Non-volatile / Volatile bit | OTP bit | OTP bit | OTP bit | Indicator bit | Non-volatile / Volatile bit | |

**Note:**
1. The default of each volatile bit is “0” at Power-up or after reset.
2. See the “Protected Area Sizes Sector Organization” table.

The status and control bits of the Suspend Status Register 2 are as follows:

**WSE bit.** The Write Suspend Erase Status (WSE) bit indicates when an Erase operation has been suspended. The WSE bit is “1” after the host issues a suspend command during an Erase operation. Once the suspended Erase resumes, the WSE bit is reset to “0”.

**WSP bit.** The Write Suspend Program Status (WSP) bit indicates when a Program operation has been suspended. The WSP is “1” after the host issues a suspend command during the Program operation.
Once the suspended Program resumes, the WSP bit is reset to “0”.

**SPL2 bit.** The SPL2 bit is non-volatile One Time Program (OTP) bit in status register that provide the write protect control and status to the security sector 2. User can read/program/erase security sector 2 as normal sector while SPL2 value is equal 0, after SPL2 is programmed with 1 by WRSR command, the security sector 2 is protected from program and erase operation. The SPL2 bit can only be programmed once.

**SPL1 bit.** The SPL1 bit is non-volatile One Time Program (OTP) bit in status register that provide the write protect control and status to the security sector 1. User can read/program/erase security sector 1 as normal sector while SPL1 value is equal 0, after SPL1 is programmed with 1 by WRSR command, the security sector 1 is protected from program and erase operation. The SPL1 bit can only be programmed once.

**SPL0 bit.** The SPL0 bit is non-volatile One Time Program (OTP) bit in status register that provide the write protect control and status to the security sector 0. User can read/program/erase security sector 0 as normal sector while SPL0 value is equal 0, after SPL0 is programmed with 1 by WRSR command, the security sector 0 is protected from program and erase operation. The SPL0 bit can only be programmed once.

**CMP bit.** The Complement Protect bit (CMP) is a non-volatile bit in Status Register 2. It is used in conjunction with 4KBL, TB, BP2, BP1, BP0 bits to provide more flexibility for the array protection. The default setting is CMP=0.

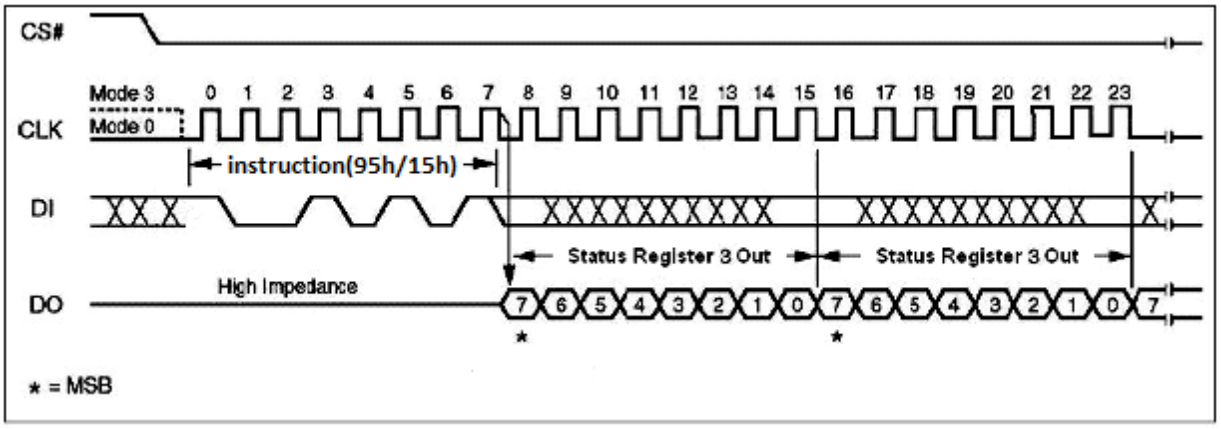
**QE bit.** The Quad Enable (QE) bit is a non-volatile Read/Write bit in the Status Register2 to disable WP# and Hold# before Quad operation. When it is “0” (factory default), the WP# and HOLD# are enabled.
On the other hand, while QE bit is “1”, the WP# and HOLD# are disabled.

User can use Flash Programmer to set QE bit as “1” and then the host system can let WP# and HOLD# keep floating in SPI mode.

| SRP | WP# | status | Description |
|---|---|---|---|
| 0 | X | Software protect | Status register can be changed |
| 1 | 0 | Hardware protect | status register is locked |
| 1 | 1 | Hardware unprotect | Status register can be changed |

### Read Status Register 3 (RDSR3) (95h/15h)

The Read Status Register 3 (RDSR3) instruction allows the Status Register 3 to be read. The Status Register 3 may be read at any time, even while output drive strength setting, wrap read length, blank check bit and dummy byte bit. When one of these bytes is in progress, it is recommended to check the Write In Progress (WIP) bit before sending a new instruction to the device. It is also possible to read the Read Status Register 3 continuously, as shown in Read Status Register 3 Instruction Sequence Diagram figure.




**Read Status Register 3 Instruction Sequence Diagram**

| SR3.7 | SR3.6 | SR3.5 | SR3.4 | SR3.3 | SR3.2 | SR3.1 | SR3.0 |
|---|---|---|---|---|---|---|---|
| Dummy configuration | Output Drive Strength | | Burst Length | | Blank check | **WEL** (Write Enable Latch) | **WIP** (Write In Progress) |
| 0= default dummy byte(default)<br>1 = more dummy byte at BBh/EBh for higher speed | 00 = 67% (default)<br>01 = 100%<br>10 = 50% Drive<br>11 = 83% Drive | | 00 = 8 Bytes (default)<br>01 = 16 Bytes<br>10 = 32 Bytes<br>11 = 64 Bytes | | 1 = flash is blank after ship out (default)<br>0 = flash had been programmed | 1 = write enable<br>0 = not write enable | 1 = write operation<br>0 = not in write operation |
| Non-volatile/volatile bit | Non-volatile/volatile bit | | Indicator bit | | Indicator bit | Read only bit | Read only bit |

**Status Register 3 Bit Locations**

The status and control bits of the Status Register 3 are as follows:

**Output Drive Strength bit.** The Output Drive Strength (SR3.6 and SR3.5) bits indicate the status of output Drive Strength in I/O pins.

**Burst Length.** The Burst Length (SR3.4 and SR3.3) bits indicate the status of wrap burst read length. The Burst Length feature is set by “Set Burst with Wrap (77h)” operation. These two bits only show the Burst Length.

**Blank check bit.** This bit is related with whole chip blank as factory default. Once any byte is programmed, this bit turns to 0 and will not be restored by further erase operation.

**Dummy configuration bit.** While set DC (dummy configuration) to 1, it provides more latency time while flash preparing to output data.

| Read Command | Dummy configuration | Number of dummy clk |
|---|---|---|
| BBh | 0(default) | 4 |
| | 1 | 8 |
| EBh | 0(default) | 6 |
| | 1 | 10 |

**WIP bit.** The Write In Progress (WIP) bit indicates whether the memory is busy with a Write Status Register, Program or Erase cycle. When set to 1, such a cycle is in progress, when reset to 0 no such cycle is in progress.

**WEL bit.** The Write Enable Latch (WEL) bit indicates the status of the internal Write Enable Latch. When set to 1 the internal Write Enable Latch is set, when set to 0 the internal Write Enable Latch is reset and no Write Status Register, Program or Erase instruction is accepted.

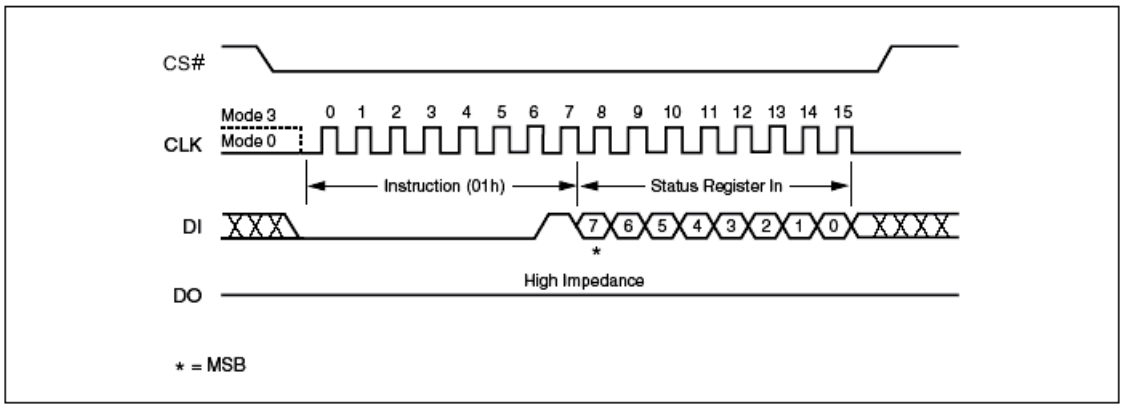
### Write Status Register (WRSR) (01h)

The Write Status Register (WRSR) instruction allows new values to be written to the Status Registers. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded and executed, the device sets the Write Enable Latch (WEL).

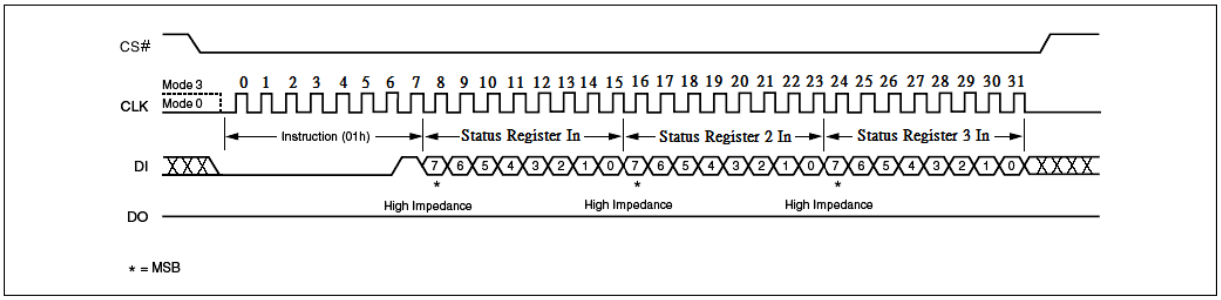
The Write Status Register (WRSR) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code and the data byte or data bytes on Serial Data Input (DI). The WRSR instruction also support multi bytes data input to set other status registers.

The instruction sequence is shown in Write Status Register Instruction Sequence Diagram figure and Write Status Register Instruction Sequence Diagram (multi byte) (3byte) figure. The Write Status Register (WRSR) instruction has no effect on S1 and S0 of the Status Register. Chip Select (CS#) must be driven High after the $8^{th}$ or $16^{th}$ or $24^{th}$ bits of the data byte has been latched in. If not, the Write Status Register (WRSR) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Write Status Register cycle (whose duration is $t_W$) is initiated. 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 Write In Progress (WIP) bit is 1 during the self-timed Write Status Register cycle, and is 0 when it is completed. When the cycle is completed, the Write Enable Latch (WEL) is reset.

The Write Status Register (WRSR) instruction allows the user to change the values of the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in Protected Area Sizes Sector Organization table. The Write Status Register (WRSR) instruction also allows the user to set or reset the Status Register Protect (SRP) bit in accordance with the Write Protect (WP#) signal. The Status Register Protect (SRP) bit and Write Protect (WP#) signal allow the device to be put in the Hardware Protected Mode (HPM). The Write Status Register (WRSR) instruction is not executed once the Hardware Protected Mode (HPM) is entered.



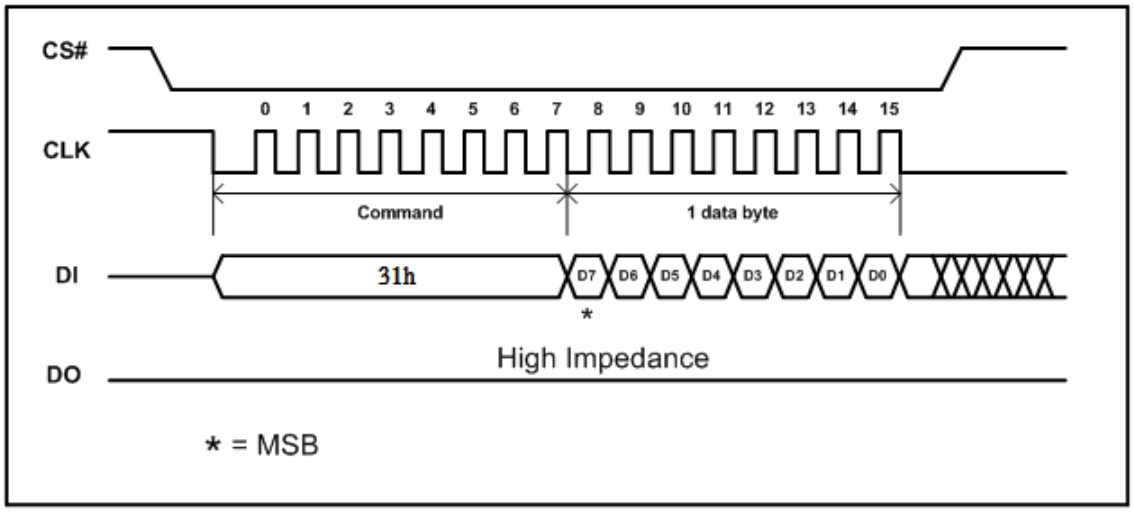

**Write Status Register Instruction Sequence Diagram**




**Write Status Register Instruction Sequence Diagram (multi byte) (3byte)**

### Write Status Register 2 (31h/01h)

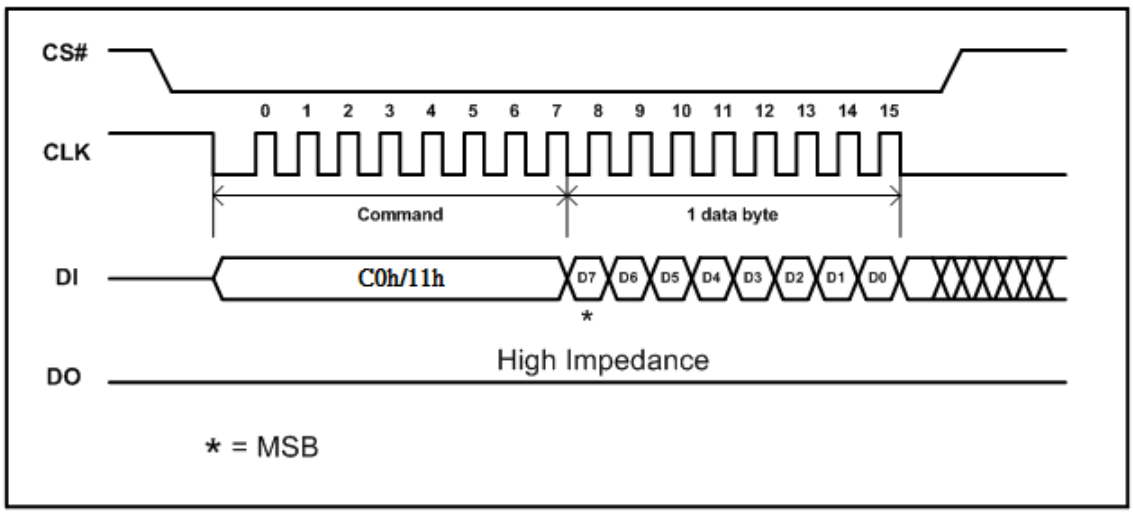
The Write Status Register 2 (31h) command can be used to set SPL0/SPL1/SPL2 OTP bits , QE bit, CMP bit To set these bits to the host driver CS# low, sends the Write Status Register 2 (31h) and one data byte, then drivers CS# high.
01h (WRSR) command also can set status register 2.




**Write Status Register 2 Instruction Sequence Diagram**

### Write Status Register 3 (C0h/11h/01h)

The Write Status Register 3 (C0h/11h) command can be used to set output drive strength in I/O pins and dummy configuration for read command bit. To set these bits to the host driver CS# low, sends the Write Status Register 3 (C0h or 11h) and one data byte, then drivers CS# high.
01h (WRSR) command also can set status register 3.



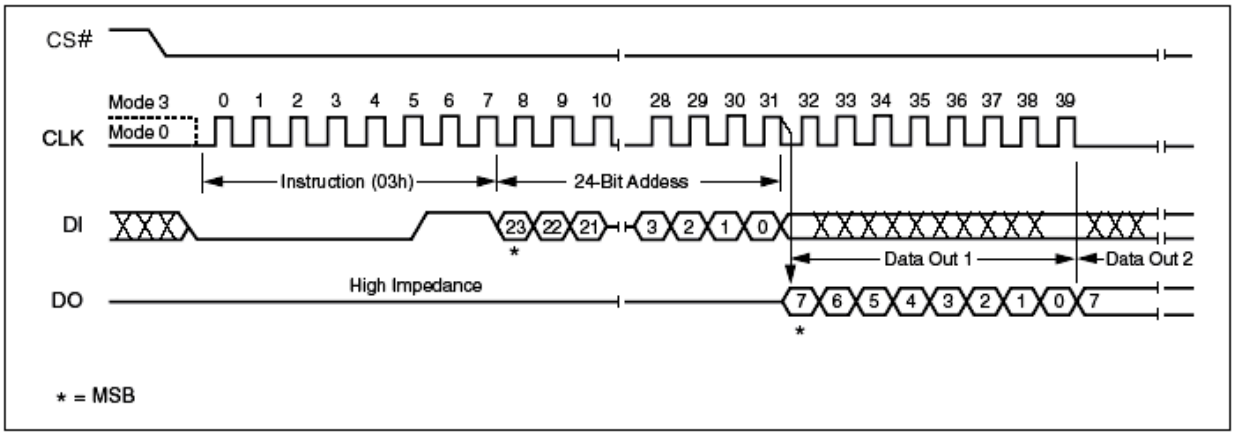

**Write Status Register 3 Instruction Sequence Diagram**

### Read Data Bytes (READ) (03h)

The device is first selected by driving Chip Select (CS#) Low. The instruction code for the Read Data Bytes (READ) instruction is followed by a 3-byte address (A23-A0), each bit being latched-in during the rising edge of Serial Clock (CLK). Then the memory contents, at that address, is shifted out on Serial Data Output (DO), each bit being shifted out, at a maximum frequency $f_R$, during the falling edge of Serial Clock (CLK).

The instruction sequence is shown in Read Data Instruction Sequence Diagram figure. 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. The whole memory can, therefore, be read with a single Read Data Bytes (READ) instruction. 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 (READ) instruction is terminated by driving Chip Select (CS#) High. Chip Select (CS#) can be driven High at any time during data output. Any Read Data Bytes (READ) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.



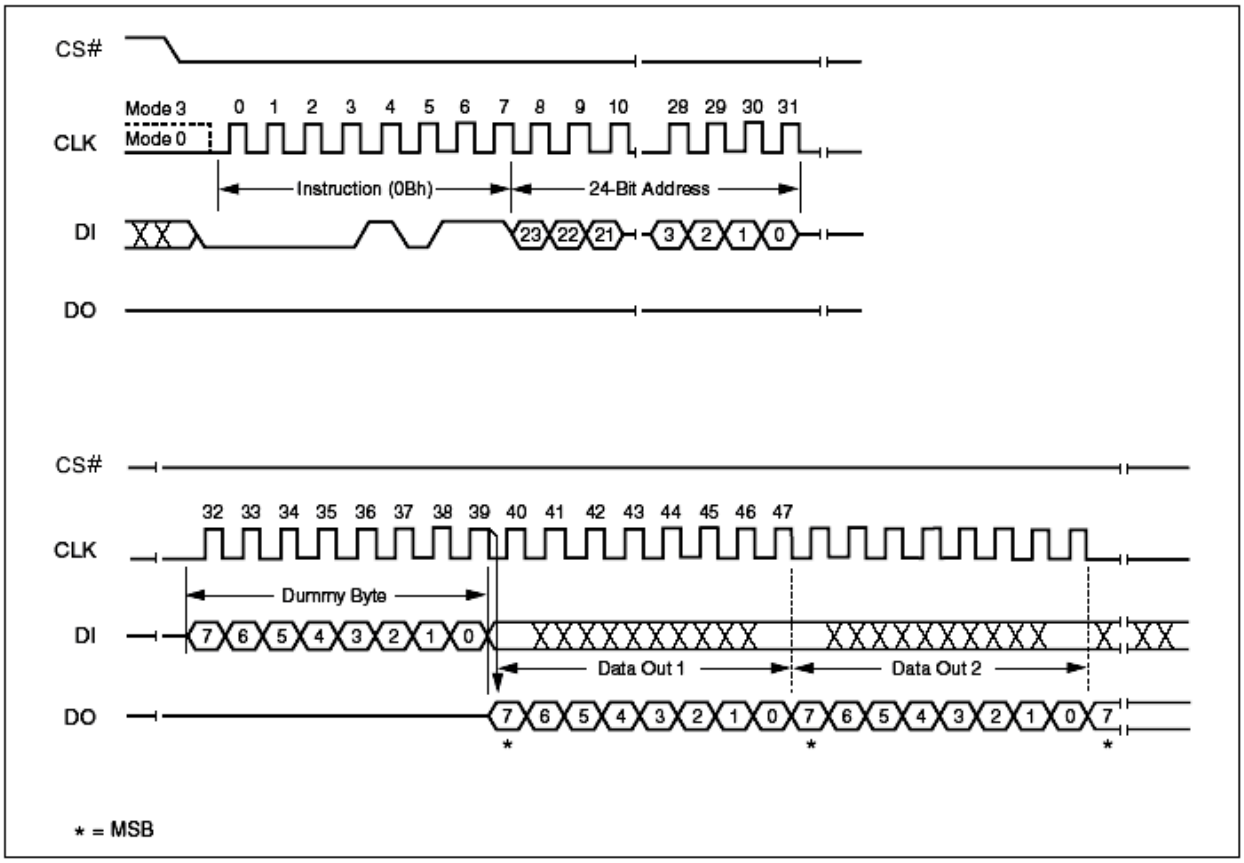

**Read Data Instruction Sequence Diagram**

### Read Data Bytes at Higher Speed (FAST_READ) (0Bh)

The device is first selected by driving Chip Select (CS#) Low. The instruction code for the Read Data Bytes at Higher Speed (FAST_READ) instruction 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 (CLK). Then the memory contents, at that address, is shifted out on Serial Data Output (DO), each bit being shifted out, at a maximum frequency $F_R$, during the falling edge of Serial Clock (CLK).

The instruction sequence is shown in Fast Read Instruction Sequence Diagram figure. 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. The whole memory can, therefore, be read with a single Read Data Bytes at Higher Speed (FAST_READ) instruction. 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 (FAST_READ) instruction is terminated by driving Chip Select (CS#) High. Chip Select (CS#) can be driven High at any time during data output. Any Read Data Bytes at Higher Speed (FAST_READ) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

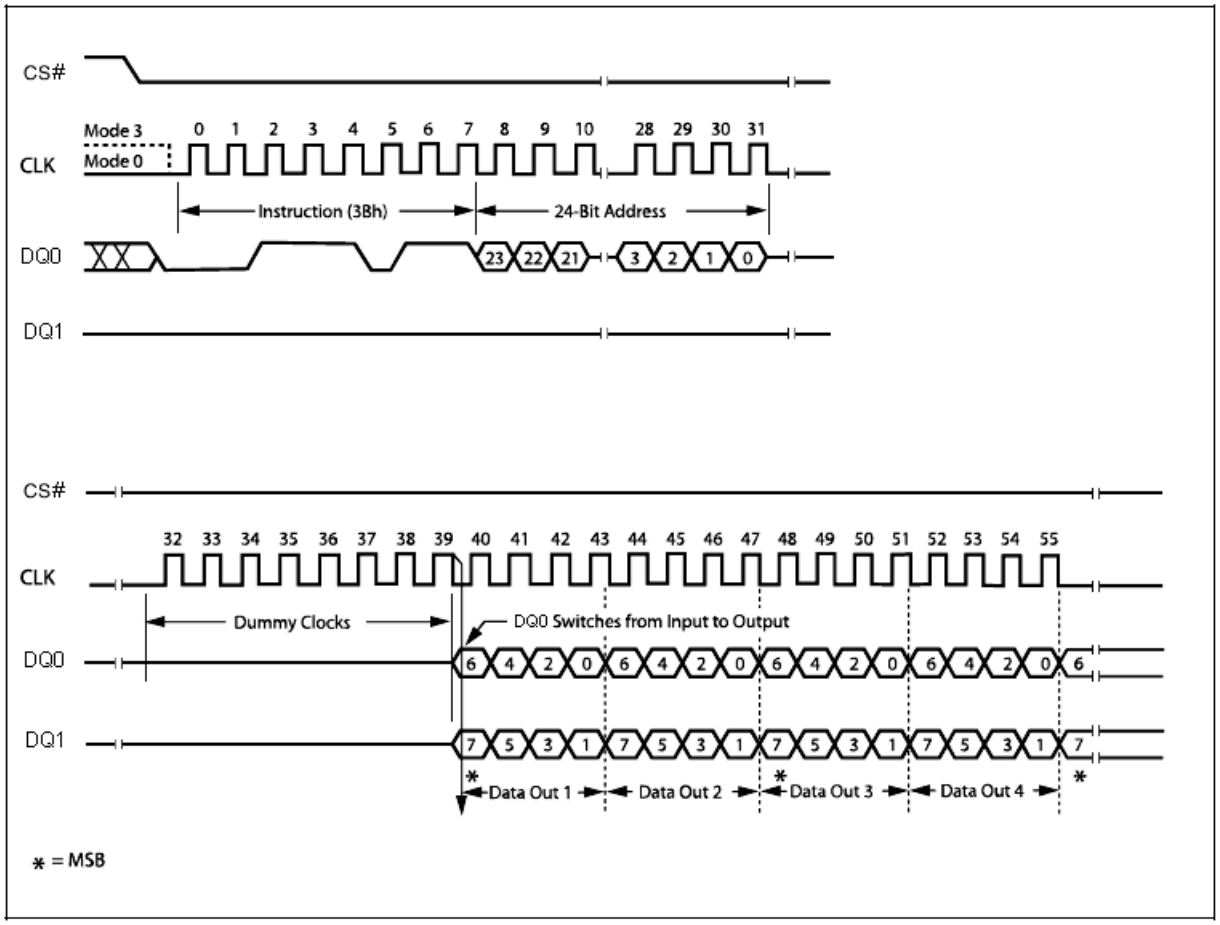



**Fast Read Instruction Sequence Diagram**

### Dual Output Fast Read (3Bh)

The Dual Output Fast Read (3Bh) is similar to the standard Fast Read (0Bh) instruction except that data is output on two pins, $DQ_0$ and $DQ_1$, instead of just $DQ_1$. This allows data to be transferred from the device at twice the rate of standard SPI devices. The Dual Output Fast Read instruction is ideal for quickly downloading code from to RAM upon power-up or for applications that cache code-segments to RAM for execution.

Similar to the Fast Read instruction, the Dual Output Fast Read instructions can operation at the highest possible frequency of $F_R$ (see AC Electrical Characteristics). This is accomplished by adding eight “dummy clocks after the 24-bit address as shown in Dual Output Fast Read Instruction Sequence Diagram figure. The dummy clocks allow the device’s internal circuits additional time for setting up the initial address. The input data during the dummy clock is “don’t care”. However, the DI pin should be high-impedance prior to the falling edge of the first data out clock.

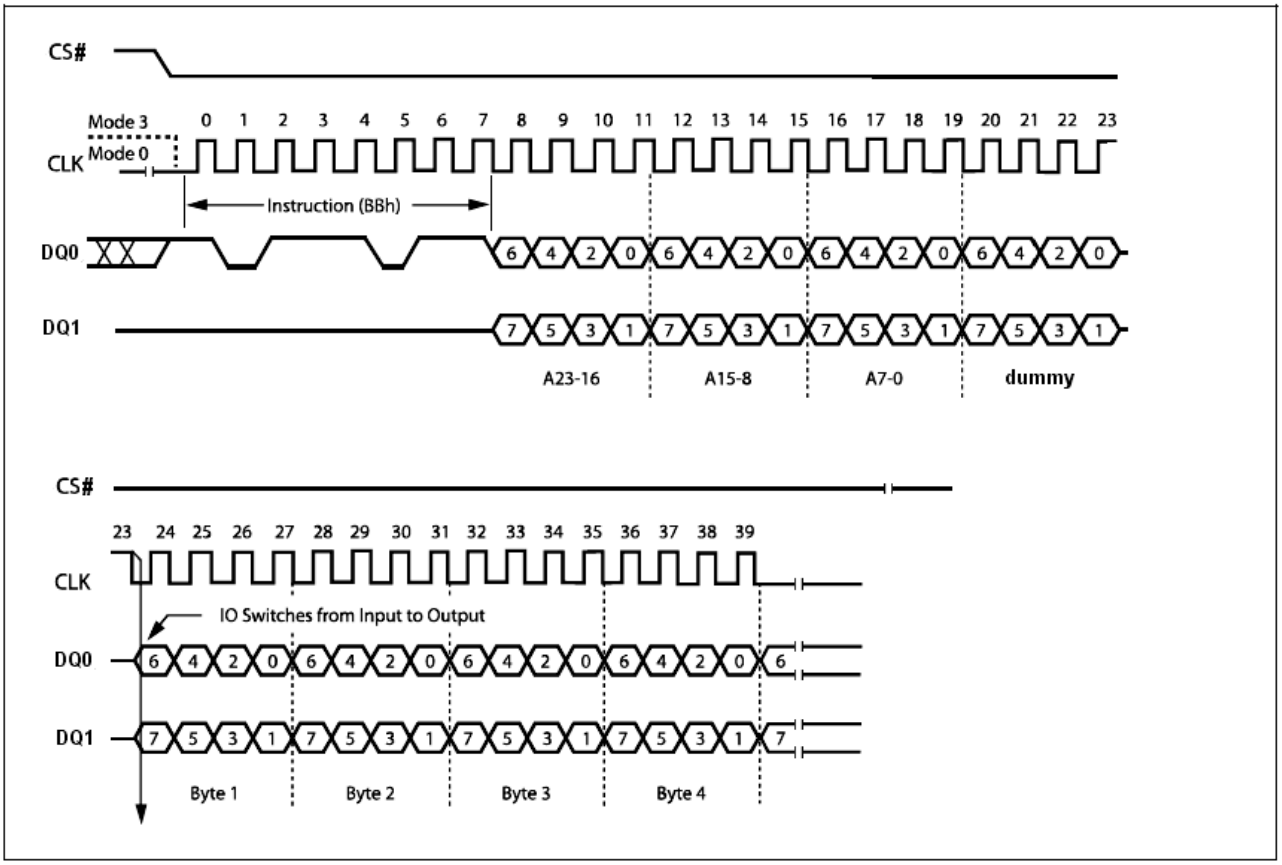



**Dual Output Fast Read Instruction Sequence Diagram**

### Dual Input / Output FAST_READ (BBh)

The Dual I/O Fast Read (BBh) instruction allows for improved random access while maintaining two IO pins, $DQ_0$ and $DQ_1$. It is similar to the Dual Output Fast Read (3Bh) instruction but with the capability to input the Address bits (A23-0) two bits per clock. This reduced instruction overhead may allow for code execution (XIP) directly from the Dual SPI in some applications.

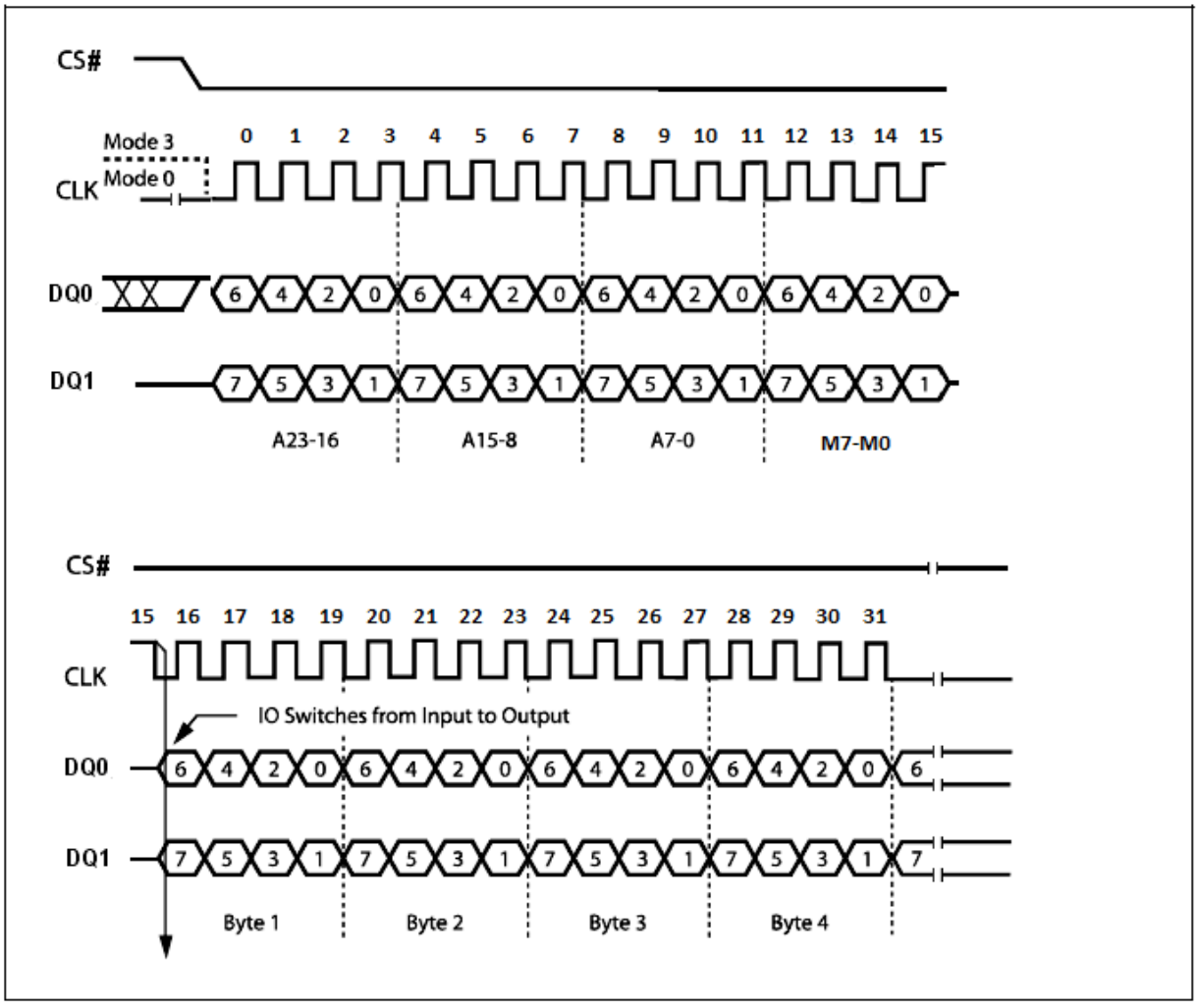
The Dual I/O Fast Read instruction enable double throughput of Serial Flash in read mode. The address is latched on rising edge of CLK, and data of every two bits (interleave 2 I/O pins) shift out on the falling edge of CLK at a maximum frequency. The first address can be at any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Dual I/O Fast Read instruction. The address counter rolls over to 0 when the highest address has been reached. Once writing Dual I/O Fast Read instruction, the following address/dummy/data out will perform as 2-bit instead of previous 1-bit, as shown in Dual Input / Output Fast Read Instruction Sequence Diagram ( M5-4 ≠ (1,0) ) figure.




**Dual Input / Output Fast Read Instruction Sequence Diagram ( M5-4 ≠ (1,0) )**

### Dual I/O Fast Read with "Enhance Read Mode"

The Dual I/O Fast Read command can further reduce command overhead through setting the "Enhance Read Mode" bits (M7-0) after the input 3-byte address (A23-A0). If the "Enhance Read Mode" bits (M5-4) = (1, 0), then the next Dual I/O Fast Read command does not require the BBh command code while next CS# goes low. If the "Enhance Read Mode" bits (M5-4) ≠ (1, 0), the next command needs the command code. A "Enhance Read Mode" Reset command can be used to reset (M5-4) before issuing normal command.




**Dual Input / Output Fast Read Instruction Sequence Diagram with "Enhance Read Mode" ( M5-4 = (1,0) )**

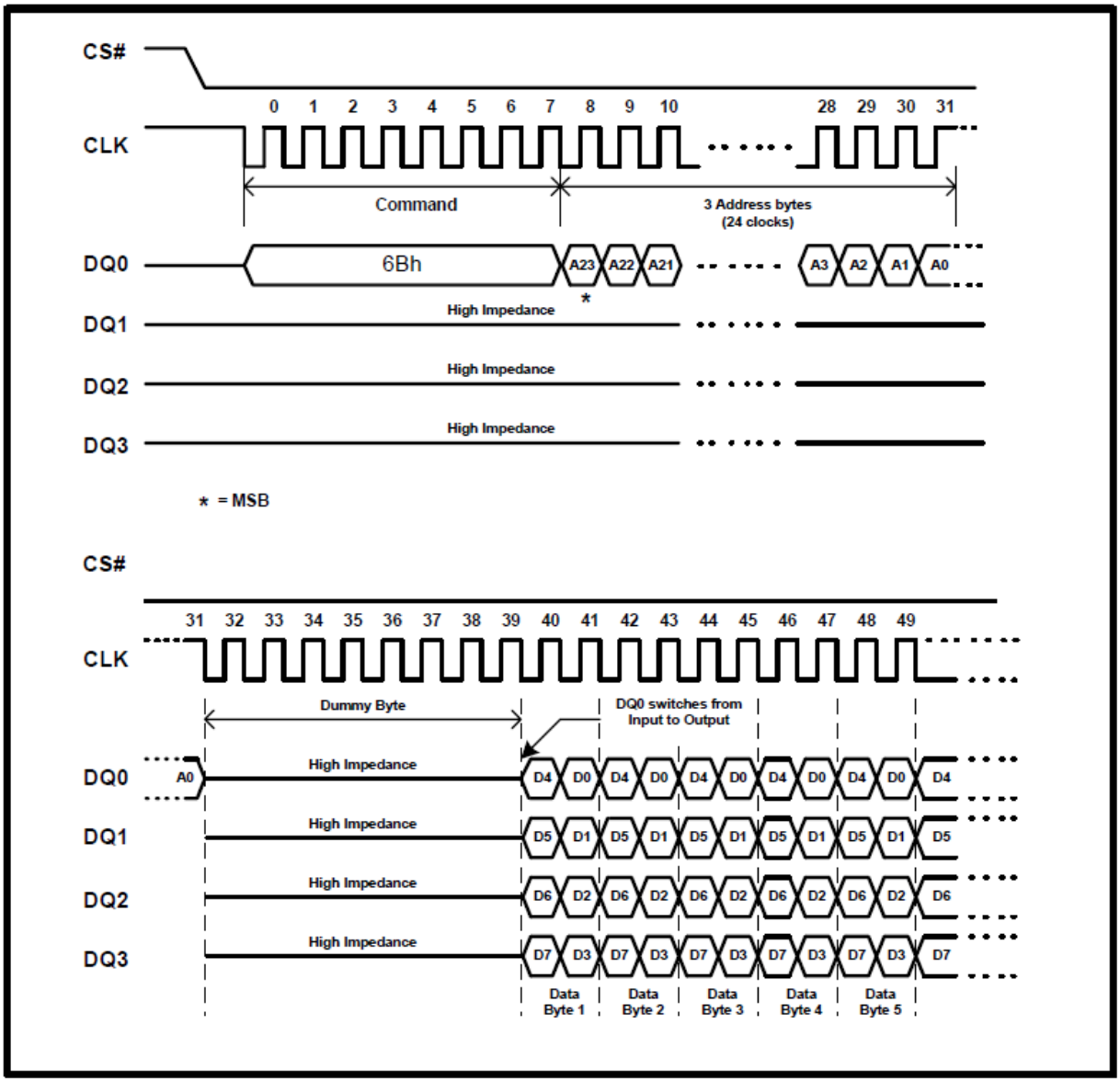
### Quad Output Fast Read (6Bh)

The Quad Output Fast Read (6Bh) instruction is similar to the Dual Output Fast Read (3Bh) instruction except that data is output through four pins, $DQ_0$, $DQ_1$, $DQ_2$ and $DQ_3$ and eight dummy clocks are required prior to the data output. The Quad Output dramatically reduces instruction overhead allowing faster random access for code execution (XIP) directly from the Quad SPI.

The Quad Output Fast Read (6Bh) address is latching on rising edge of CLK, and data of every four bits (interleave on 4 I/O pins) shift out on the falling edge of CLK at a maximum frequency $F_R$. The first address can be any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Quad Output Fast Read instruction. The address counter rolls over to 0 when the highest address has been reached.

The sequence of issuing Quad Output Fast Read (6Bh) instruction is: CS# goes low -> sending Quad Output Fast Read (6Bh) instruction -> 24-bit address on $DQ_0$ -> 8 dummy clocks -> data out interleave on $DQ_3$, $DQ_2$, $DQ_1$ and $DQ_0$ -> to end Quad Output Fast Read (6Bh) operation can use CS# to high at any time during data out, as shown in Quad Output Fast Read Instruction Sequence Diagram figure.

The Quad Enable bit (QE) of Status Register (S9) must be set to enable for the Quad Output Fast Read command.




**Quad Output Fast Read Instruction Sequence Diagram**

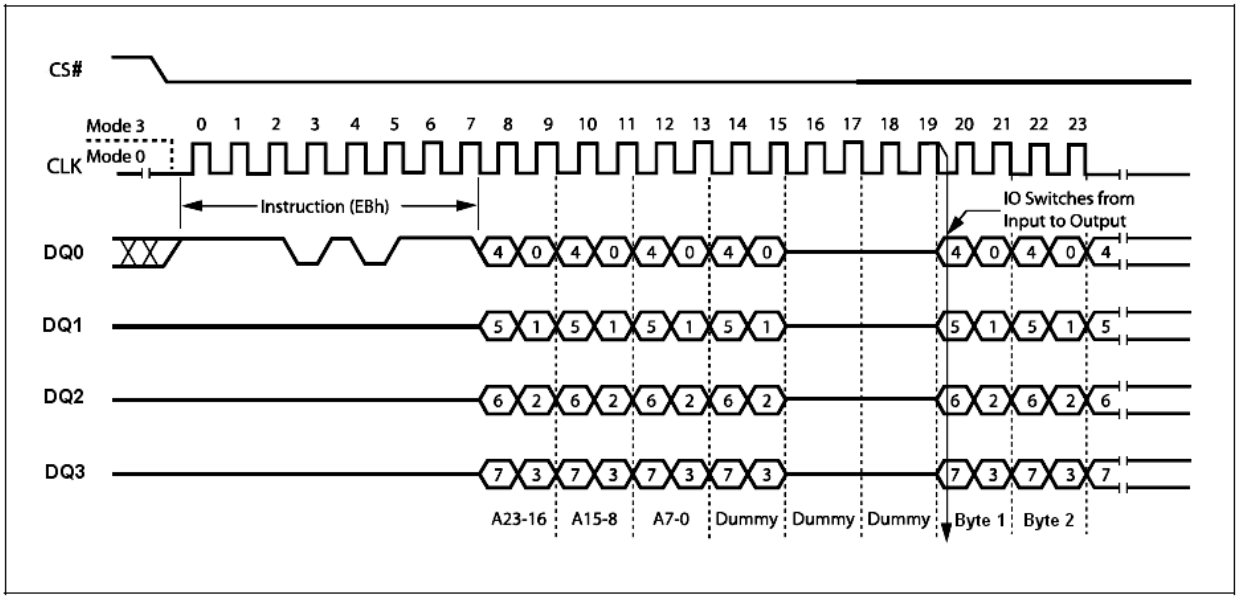
### Quad Input / Output FAST_READ (EBh)

The Quad Input/Output FAST_READ (EBh) instruction is similar to the Dual I/O Fast Read (BBh) instruction except that address and data bits are input and output through four pins, $DQ_0$, $DQ_1$, $DQ_2$ and $DQ_3$ and six dummy clocks are required prior to the data output. The Quad I/O dramatically reduces instruction overhead allowing faster random access for code execution (XIP) directly from the Quad SPI.

The Quad Input/Output FAST_READ (EBh) instruction enable quad throughput of Serial Flash in read mode. The address is latching on rising edge of CLK, and data of every four bits (interleave on 4 I/O pins) shift out on the falling edge of CLK at a maximum frequency $F_R$. The first address can be any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Quad Input/Output FAST_READ instruction. The address counter rolls over to 0 when the highest address has been reached. Once writing Quad Input/Output FAST_READ instruction, the following address/dummy/data out will perform as 4-bit instead of previous 1-bit.

The sequence of issuing Quad Input/Output FAST_READ (EBh) instruction is: CS# goes low -> sending Quad Input/Output FAST_READ (EBh) instruction -> 24-bit address interleave on $DQ_3$, $DQ_2$, $DQ_1$ and $DQ_0$ -> 6 dummy clocks -> data out interleave on $DQ_3$, $DQ_2$, $DQ_1$ and $DQ_0$ -> to end Quad Input/Output FAST_READ (EBh) operation can use CS# to high at any time during data out, as shown in Quad Input / Output Fast Read Instruction Sequence Diagram figure.

The Quad Enable bit (QE) of Status Register (S9) must be set to enable for the Quad I/O Fast read command.



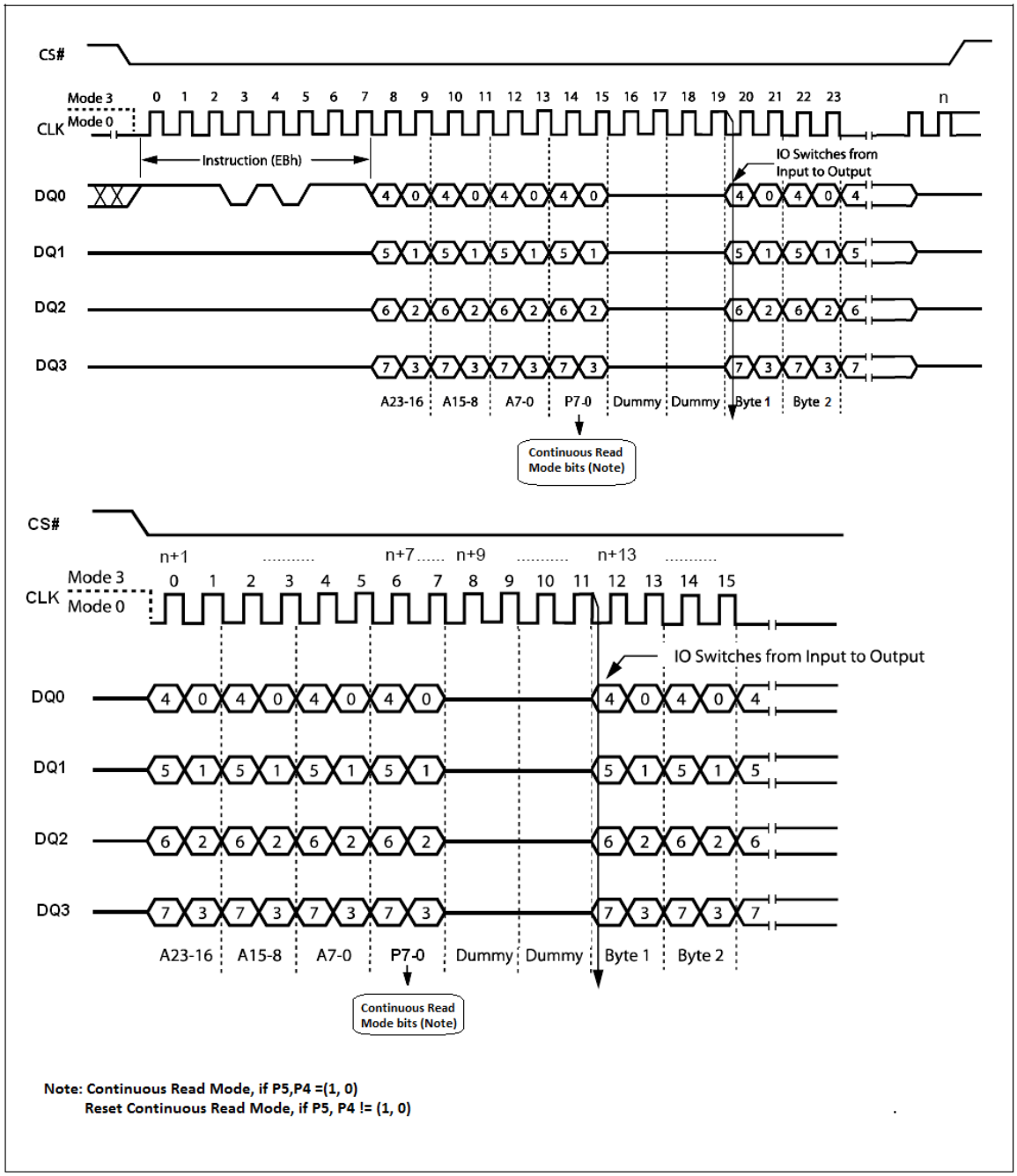

**Quad Input / Output Fast Read Instruction Sequence Diagram**

### Quad I/O Fast Read with "Enhance Read Mod"

Another sequence of issuing Quad Input/Output FAST_READ (EBh) instruction especially useful in random access is : CS# goes low -> sending Quad Input/Output FAST_READ (EBh) instruction -> 24-bit address interleave on $DQ_3$, $DQ_2$, $DQ_1$ and $DQ_0$ -> Enhance Read Mode bit P[5,4]=(1,0) -> 4 dummy clocks -> data out interleave on $DQ_3$, $DQ_2$, $DQ_1$ and $DQ_0$ till CS# goes high -> CS# goes low (reduce Quad Input/Output FAST_READ (EBh) instruction) -> 24-bit random access address, as shown in Quad Input/Output Fast Read Enhance Read Mode Sequence Diagram figure.

In the Enhance Read Mode, P[5:4] must be equal to (1, 0) ; likewise P[7:0] = Axh, Exh, can make this mode continue and reduce the next Quad Input/Output FAST_READ (EBh) instruction. Once P[5:4] is no longer equal to (1,0); likewise P[7:0] = FFh, 00h. These commands will reset the enhance read mode.

While Program/ Erase/ Write Status Register is in progress, Quad Input/Output FAST_READ (EBh) instruction is rejected without impact on the Program/ Erase/ Write Status Register current cycle.




**Quad Input/Output Fast Read Enhance Read Mode Sequence Diagram**

### Quad Intput/Output Fast Read with "8/16/32/64 byte Wrap Around"

The Quad I/O Fast Read command can be used to access a specific portion within a page by issuing "Set Burst with Wrap" (77h) commands prior to EBh. The "Set Burst with Wrap" (77h) command can either enable or disable the "Wrap Around" feature for the following EBh commands. When "Wrap Around" is enabled, the data being accessed can be limited to either an 8/16/32/64-byte section of a 256-byte page. The output data starts at the initial address specified in the command, once it reaches the ending boundary of the 8/16/32/64-byte section, the output will wrap around the beginning boundary automatically until CS# is pulled high to terminate the command.

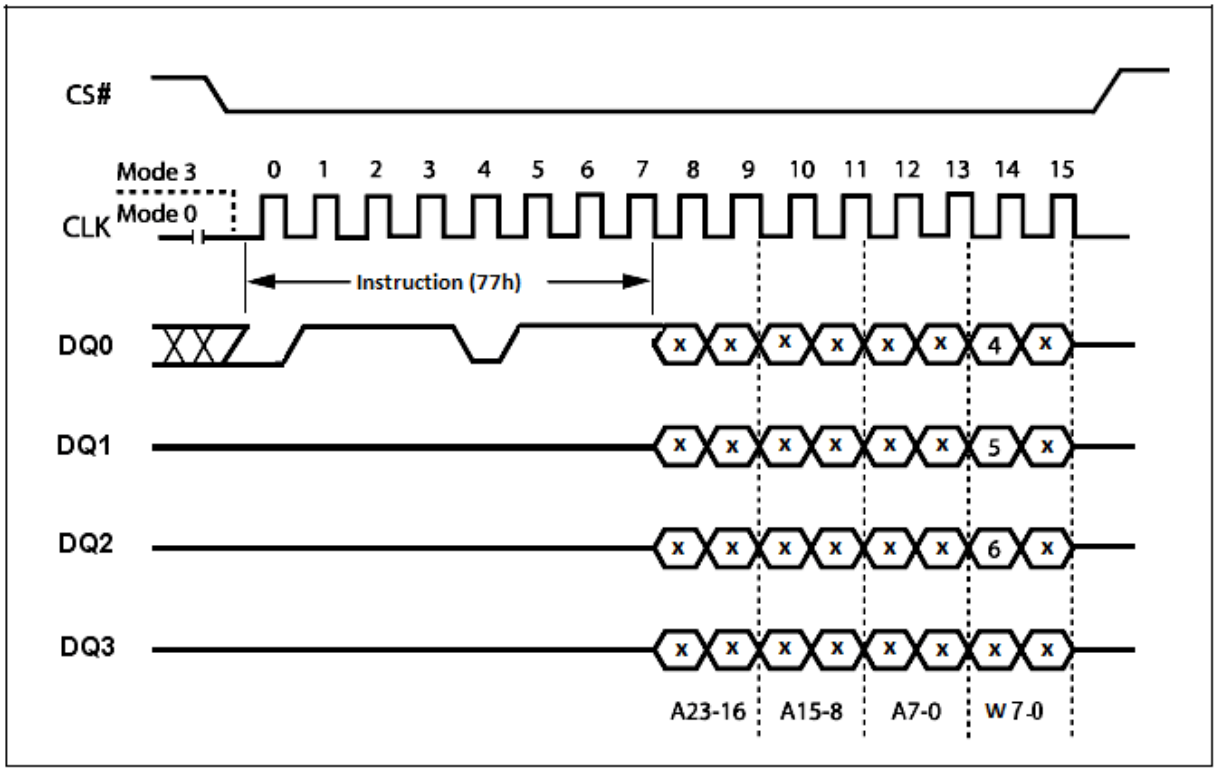
The Burst with Wrap feature allows applications that use cache to quickly fetch a critical address and then fill the cache afterwards within a fixed length (8/16/32/64-byte) of data without issuing multiple read commands. The "Set Burst with Wrap" command allows three "Wrap Bits" W6-W4 to be set. The W4 bit is used to enable or disable the "Wrap Around" operation while W6-W5 is used to specify the length of the wrap around section within a page.

### Set Burst with Wrap (77h)

The Set Burst with Wrap command is used in conjunction with "Quad I/O Fast Read (EBh)" command to access a fixed length of 8/16/32/64-byte section within a 256-byte page, in standard SPI mode.
The Set Burst with Wrap command sequence: CS# goes low -> Send Set Burst with Wrap command (77h)-> 24 dummy bits -> 8 bits "Wrap bits" ->CS# goes high. (Set Burst with Wrap Sequence Diagram figure.)

| W6, W5 | W4=0 | | W4=1 (default) | |
|---|---|---|---|---|
| | Wrap Around | Wrap Length | Wrap Around | Wrap Length |
| 0, 0 | Yes | 8 byte | No | - |
| 0, 1 | Yes | 16 byte | No | - |
| 1, 0 | Yes | 32 byte | No | - |
| 1, 1 | Yes | 64 byte | No | - |

If the W6-W4 bits are set by the Set Burst with Wrap (77h) command, all the following "Quad I/O Fast Read (EBh)" command will use the W6-W4 setting to access the 8/16/32/64-byte section within any page. To exit the "Wrap Around" function and return to normal read operation, another Set Burst with Wrap command should be issued to set W4=1.




**Set Burst with Wrap Sequence Diagram**

### Page Program (PP) (02h)

The Page Program (PP) instruction allows bytes to be programmed in the memory. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Page Program (PP) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, three address bytes and at least one data byte on Serial Data Input (DI). If the 8 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 (from the address whose 8 least significant bits (A7-A0) are all zero). Chip Select (CS#) must be driven Low for the entire duration of the sequence.

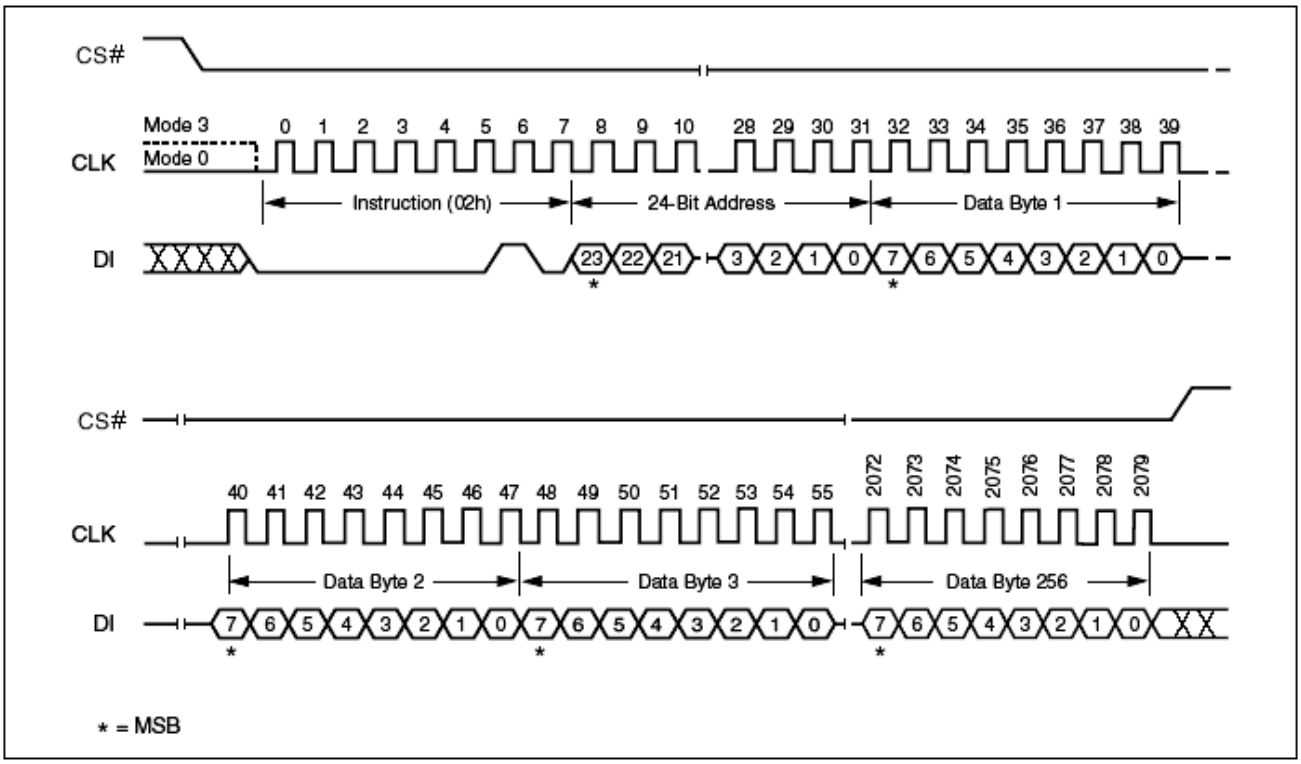
The instruction sequence is shown in Page Program Instruction Sequence Diagram figure. If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 Data bytes are sent to device, they are correctly programmed at the requested addresses without having any effects on the other bytes of the same page.

Chip Select (CS#) must be driven High after the eighth bit of the last data byte has been latched in, otherwise the Page Program (PP) instruction is not executed.

As soon as Chip Select (CS#) is driven high, the self-timed Page Program cycle (whose duration is $t_{PP}$) is initiated.

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 Write In Progress (WIP) bit is 1 during the self-timed Page Program cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Page Program (PP) instruction applied to a page which is protected by the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) is not executed.

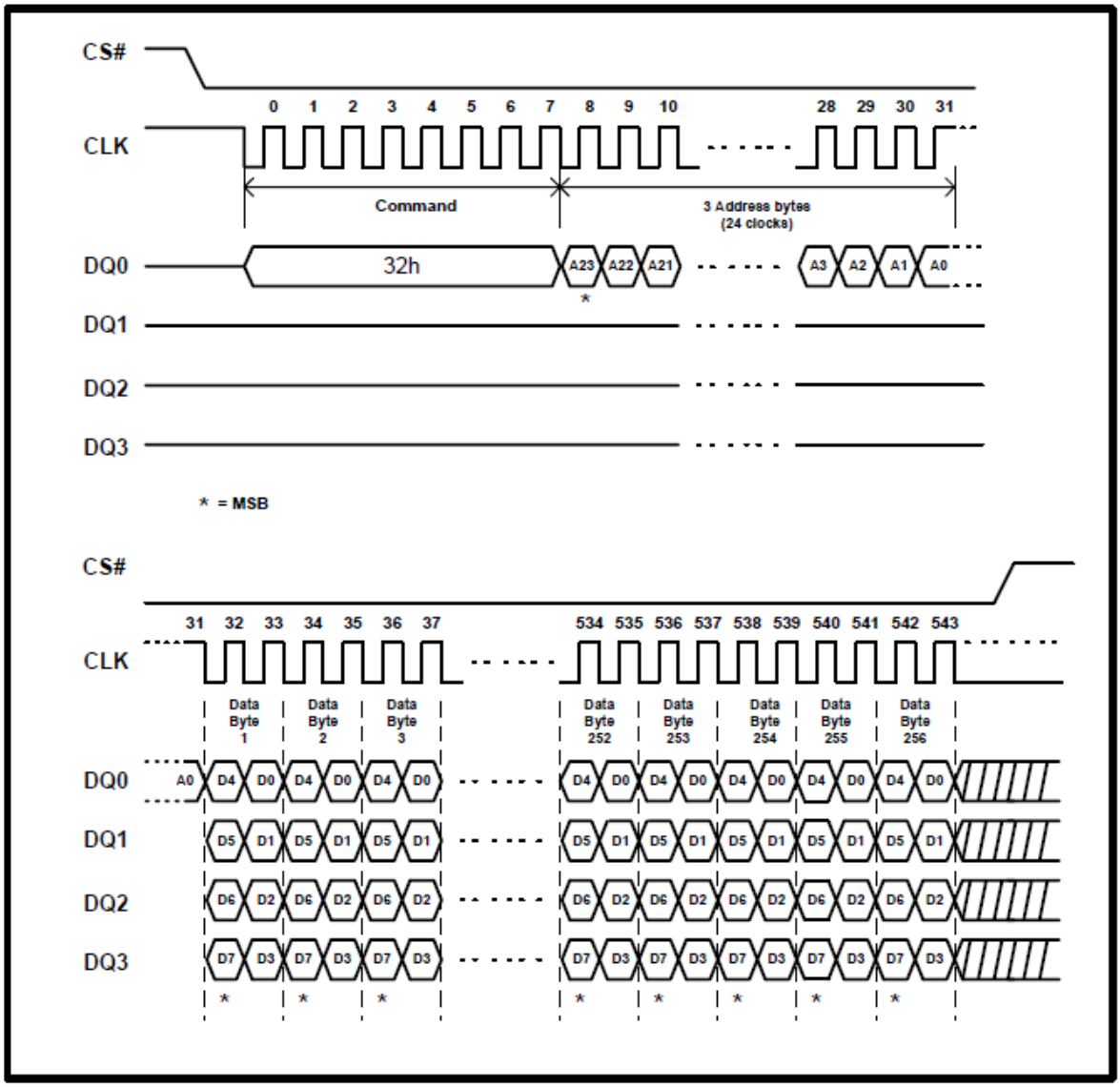



**Page Program Instruction Sequence Diagram**

### Quad Input Page Program (QPP) (32h)

The Quad Page Program (QPP) instruction allows up to 256 bytes of data to be programmed at previously erased (FFh) memory locations using four pins: $DQ_0$, $DQ_1$, $DQ_2$ and $DQ_3$. The Quad Page Program can improve performance for PROM Programmer and applications that have slow clock speeds < 5MHz. Systems with faster clock speed will not realize much benefit for the Quad Page Program instruction since the inherent page program time is much greater than the time it take to clock- in the data.

To use Quad Page Program (QPP) the WP# and HOLD# Disable (QE) bit in Status Register must be set to 1. A Write Enable instruction must be executed before the device will accept the Quad Page Program (QPP) instruction (SR.1, WEL=1). The instruction is initiated by driving the CS# pin low then shifting the instruction code “32h” followed by a 24-bit address (A23-A0) and at least one data byte, into the IO pins. The CS# pin must be held low for the entire length of the instruction while data is being sent to the device. All other functions of Quad Page Program (QPP) are identical to standard Page Program. The Quad Page Program (QPP) instruction sequence is shown in Quad Input Page Program Instruction Sequence Diagram figure.




**Quad Input Page Program Instruction Sequence Diagram**

### Write Suspend (75h/B0h)

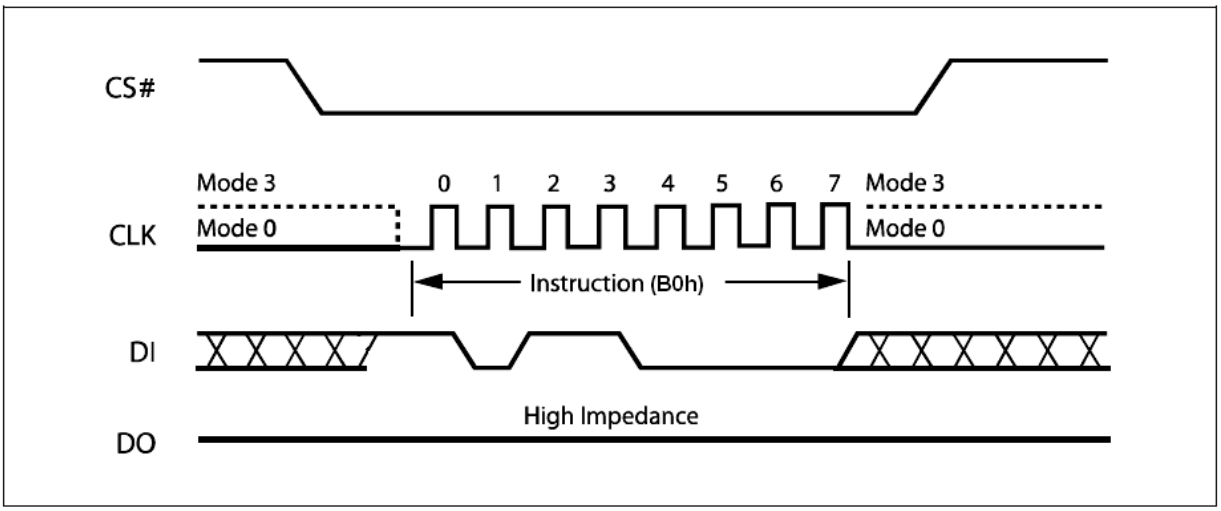
Write Suspend allows the interruption of Sector Erase, Block Erase or Page Program operations in order to read data in another sector or block of memory. The original operation can be continued with Write Resume command. The instruction sequence is shown in Suspend Instruction Sequence Diagram figure.

Only one write operation can be suspended at a time; if an operation is already suspended, the device will ignore the Write Suspend command. Write Suspend during Chip Erase is ignored; Chip Erase is not a valid command while a write is suspended.

Suspend to suspend ready timing: 28us.

Resume to another suspend timing: min 0.3us typ 400us

**Note:**
User can use resume to another suspend minimum timing for issue next suspend after resume, but the device needs equal or longer typical time to make other progress after resume command.




**Write Suspend Instruction Sequence Diagram**

### Write Suspend During Sector Erase or Block Erase

The Write Status Register command (01h, 31h, C0h, 11h) and Erase OTP array command (44h) and Erase commands (20h, 52h, D8h, C7h/60h) are not allowed during Erase suspend.

Issuing a Write Suspend instruction during Erasing allows the host to read any block which is not being erased or to program any block which is not being erased. Program commands pointing to the suspended sector will be ignored. Any attempt to read from the suspended block will output unknown data because the Sector or Block Erase will be incomplete.

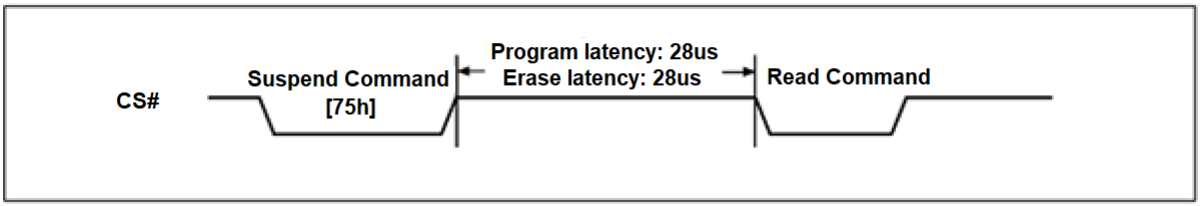
To execute a Write Suspend operation, the host drives CS# low, sends the Write Suspend command cycle (B0h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. The Suspend Status register indicates that the erase has been suspended by changing the WSE bit from “0” to “1”, but the device will not accept another command until it is ready. To determine when the device will accept a new command, poll the WIP bit in the Suspend Status register or after issue erase suspend command, latency time 28us is needed before issue another command. For “Suspend to Read”, “Resume to Read”, “Resume to Suspend” timing specification please note Suspend to Read Latency, Resume to Read Latency and Resume to Suspend Latency figure.

### Write Suspend During Page Programming

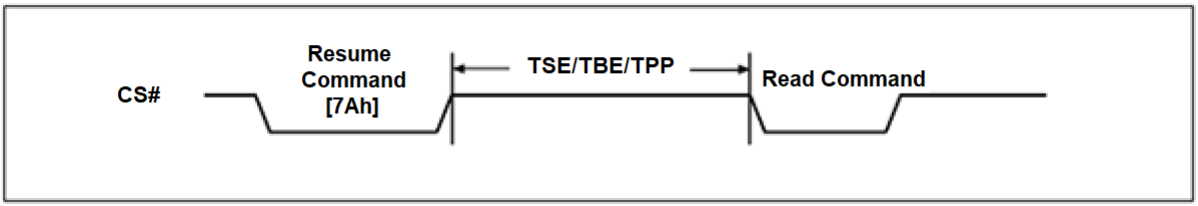
The Write Status Register command (01h, 31h, C0h, 11h) and Erase/Program OTP array command (44h, 42h) and Erase commands (20h, 52h, D8h, C7h/60h) and Page Program command (02h, 32h) are not allowed during Program suspend.

Issuing a Write Suspend instruction during Page Programming allows the host to read any page which is not being programmed. Erase commands pointing to the suspended sector will be ignored. Any attempt to read from the suspended page will output unknown data because the program will be incomplete.

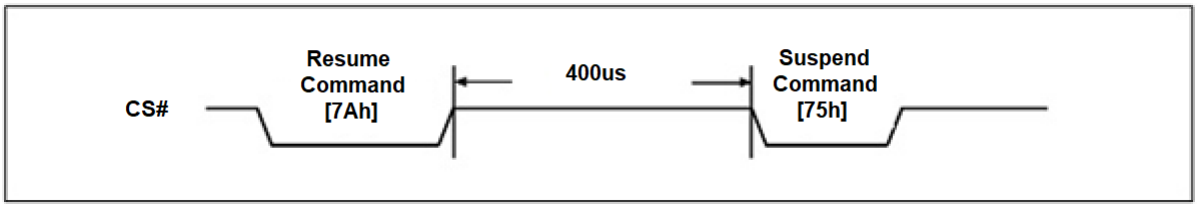
To execute a Write Suspend operation, the host drives CS# low, sends the Write Suspend command cycle (B0h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. The Suspend Status register indicates that the programming has been suspended by changing the WSP bit from “0” to “1”, but the device will not accept another command until it is ready. To determine when the device will accept a new command, poll the WIP bit in the Suspend Status register or after issue program suspend command, latency time 28us is needed before issue another command. For “Suspend to Read”, “Resume to Read”, “Resume to Suspend” timing specification please note Suspend to Read Latency, Resume to Read Latency and Resume to Suspend Latency figure.




**Suspend to Read Latency**




**Resume to Read Latency**

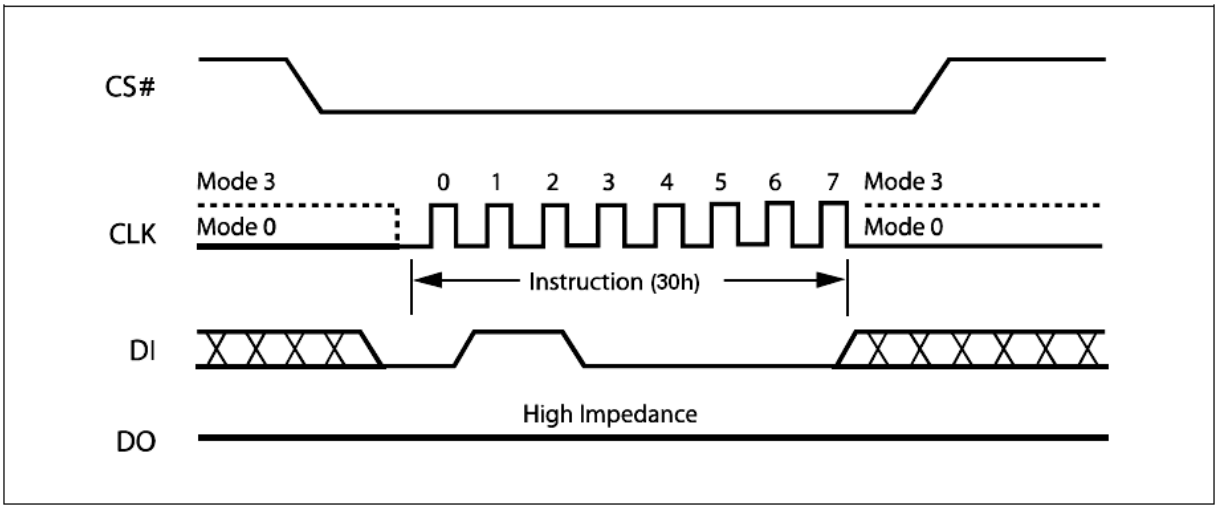



**Resume to Suspend Latency**

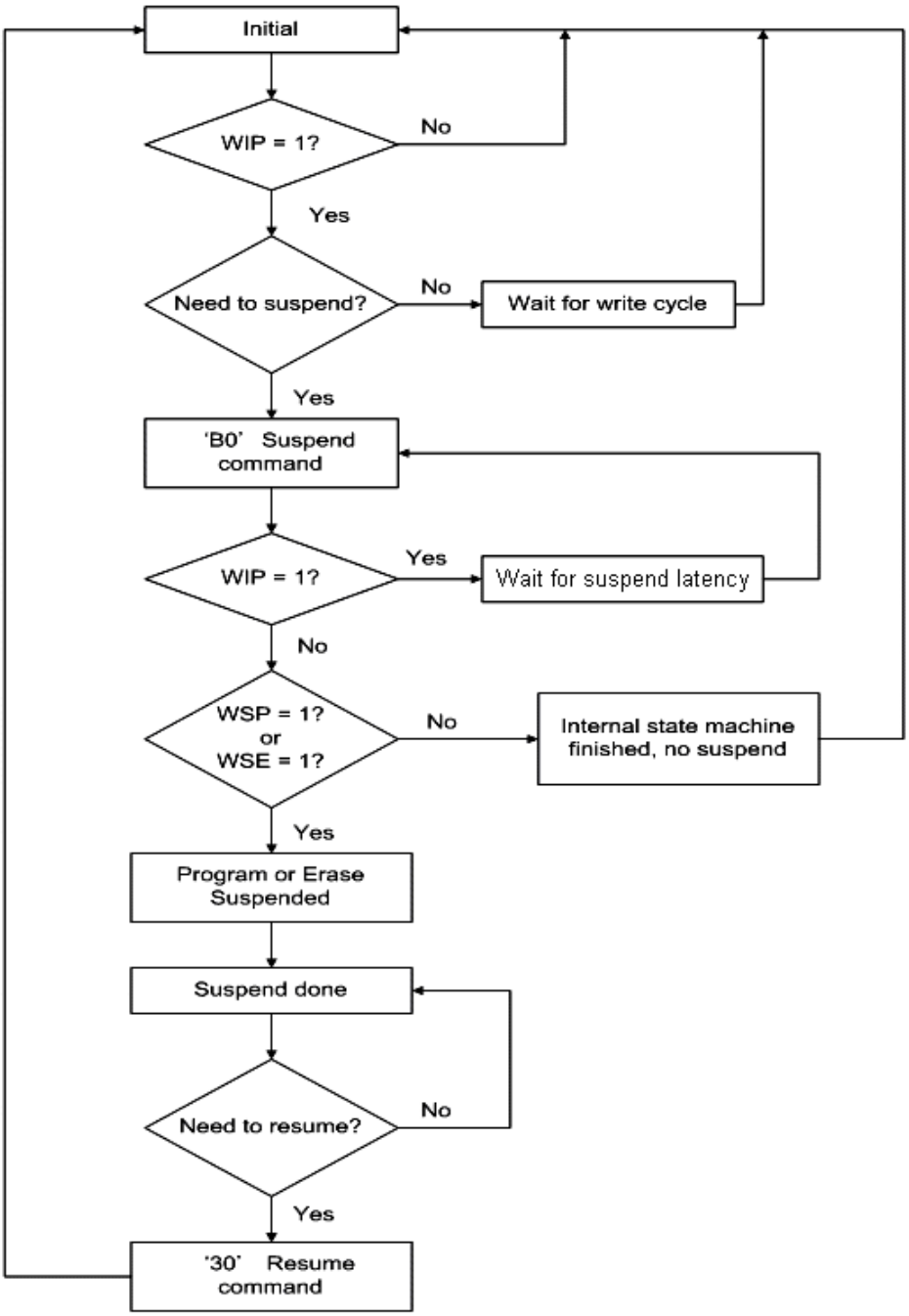
### Write Resume (7Ah/30h)

Write Resume restarts a Write command that was suspended, and changes the suspend status bit in the Status Register 2 (WSE or WSP) back to “0”.

The instruction sequence is shown in Resume Instruction Sequence Diagram figure. To execute a Write Resume operation, the host drives CS# low, sends the Write Resume command cycle (7Ah/30h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. To determine if the internal, self-timed Write operation completed, poll the WIP bit in the Status Register 2, or wait the specified time $t_{SE}$, $t_{HBE}$, $t_{BE}$ or $t_{PP}$ for Sector Erase, Half Block Erase, Block Erase or Page Programming, respectively. The total write time before suspend and after resume will not exceed the uninterrupted write times $t_{SE}$, $t_{HBE}$, $t_{BE}$ or $t_{PP}$. Resume to another suspend operation requires latency time of 400us.




**Write Resume Instruction Sequence Diagram**




**Write Suspend/Resume Flow**

**Note:**

1. The 'WIP' can be either checked by command '05h' polling.
2. 'Wait for write cycle' can be referring to maximum write cycle time or polling the WIP.
3. 'Wait for suspend latency', after issue program suspend command, latency time 28us is needed before issue another command or polling the WIP.
4. The 'WSP' and 'WSE' can be checked by command '09h/35h' polling.
5. 'Suspend done' means the chip can do further operations allowed by suspend spec.

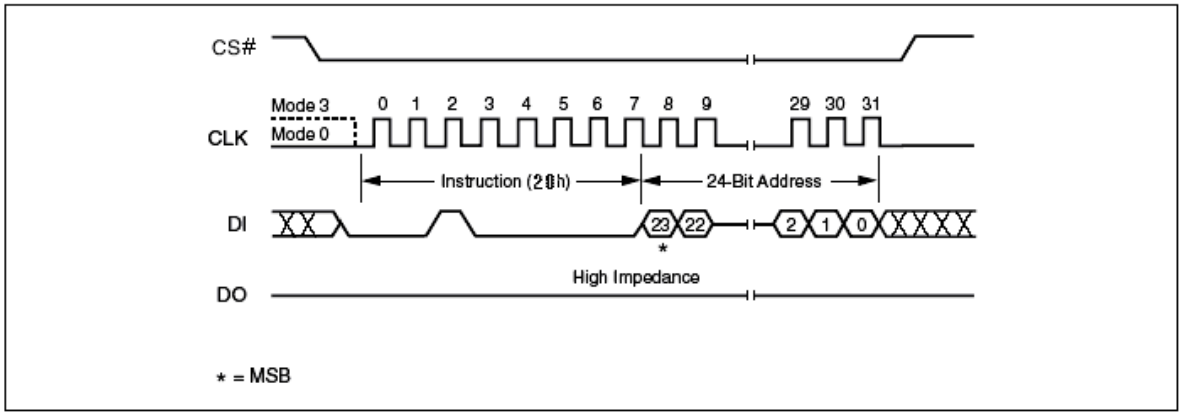
### Sector Erase (SE) (20h)

The Sector Erase (SE) instruction sets to 1 (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Sector Erase (SE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Sector (see Uniform Block Sector Architecture table) is a valid address for the Sector Erase (SE) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in Sector Erase Instruction Sequence Diagram figure. Chip Select (CS#) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Sector Erase (SE) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Sector Erase cycle (whose duration is $t_{SE}$) is initiated. 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 Write In Progress (WIP) bit is 1 during the self-timed Sector Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Sector Erase (SE) instruction applied to a sector which is protected by the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) or Boot Lock feature will be ignored.




**Sector Erase Instruction Sequence Diagram**

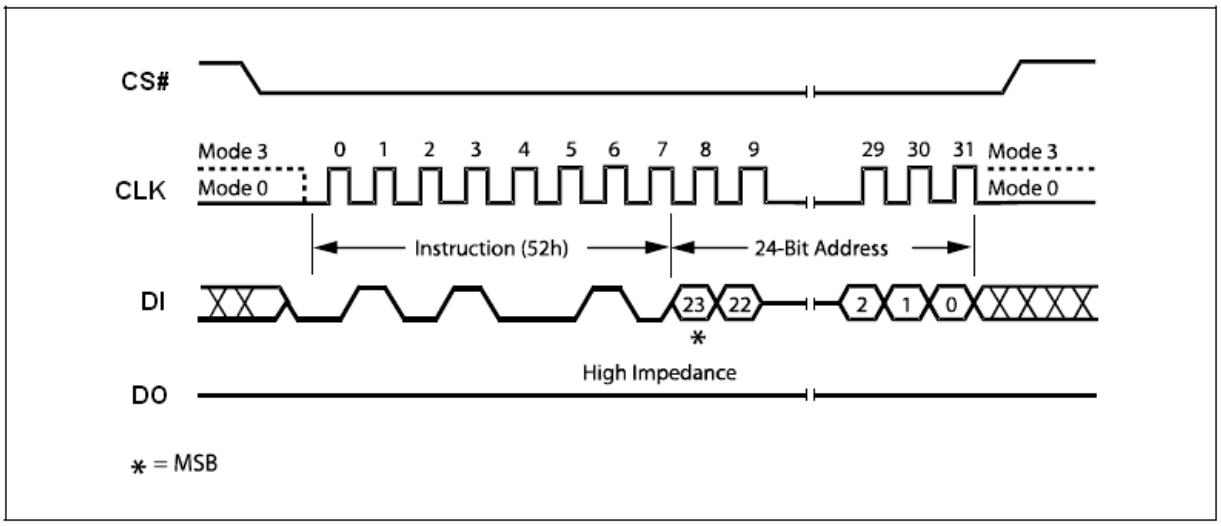
### 32KB Half Block Erase (HBE) (52h)

The Half Block Erase (HBE) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Half Block Erase (HBE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Block (see Uniform Block Sector Architecture table) is a valid address for the Half Block Erase (HBE) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in 32KB Half Block Erase Instruction Sequence Diagram figure. Chip Select (CS#) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Half Block Erase (HBE) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Block Erase cycle (whose duration is $t_{HBE}$) is initiated. While the Half Block Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Half Block Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Half Block Erase (HBE) instruction applied to a block which is protected by the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table).




**32KB Half Block Erase Instruction Sequence Diagram**

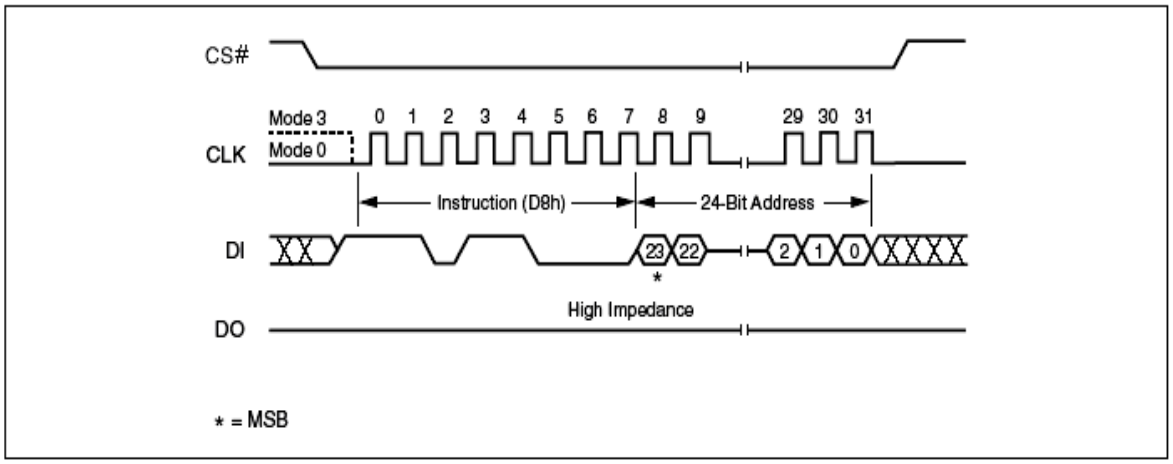
### 64KB Block Erase (BE) (D8h)

The Block Erase (BE) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Block Erase (BE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Block (see Uniform Block Sector Architecture table) is a valid address for the Block Erase (BE) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in 64KB Block Erase Instruction Sequence Diagram figure. Chip Select (CS#) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Block Erase (BE) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Block Erase cycle (whose duration is $t_{BE}$) is initiated. While the Block Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Block Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

A Block Erase (BE) instruction applied to a block which is protected by the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table).




**64KB Block Erase Instruction Sequence Diagram**

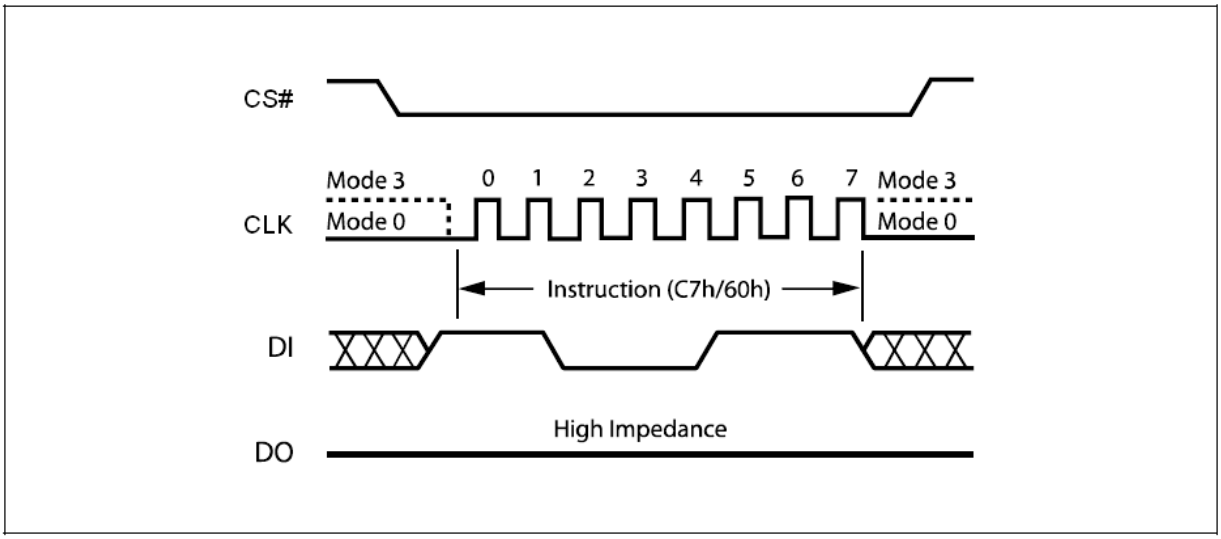
### Chip Erase (CE) (C7h/60h)

The Chip Erase (CE) instruction sets all bits to 1 (FFh). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Chip Erase (CE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code on Serial Data Input (DI). Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in Chip Erase Instruction Sequence Diagram figure. Chip Select (CS#) must be driven High after the eighth bit of the instruction code has been latched in, otherwise the Chip Erase instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Chip Erase cycle (whose duration is $t_{CE}$) is initiated. While the Chip Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Chip Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

The Chip Erase (CE) instruction is ignored if one or more blocks are protected.




**Chip Erase Instruction Sequence Diagram**

### Deep Power-down (DP) (B9h)

Executing the Deep Power-down (DP) instruction is the only way to put the device in the lowest consumption mode (the Deep Power-down mode). It can also be used as an extra software protection mechanism, while the device is not in active use, since in this mode, the device ignores all Write, Program and Erase instructions.

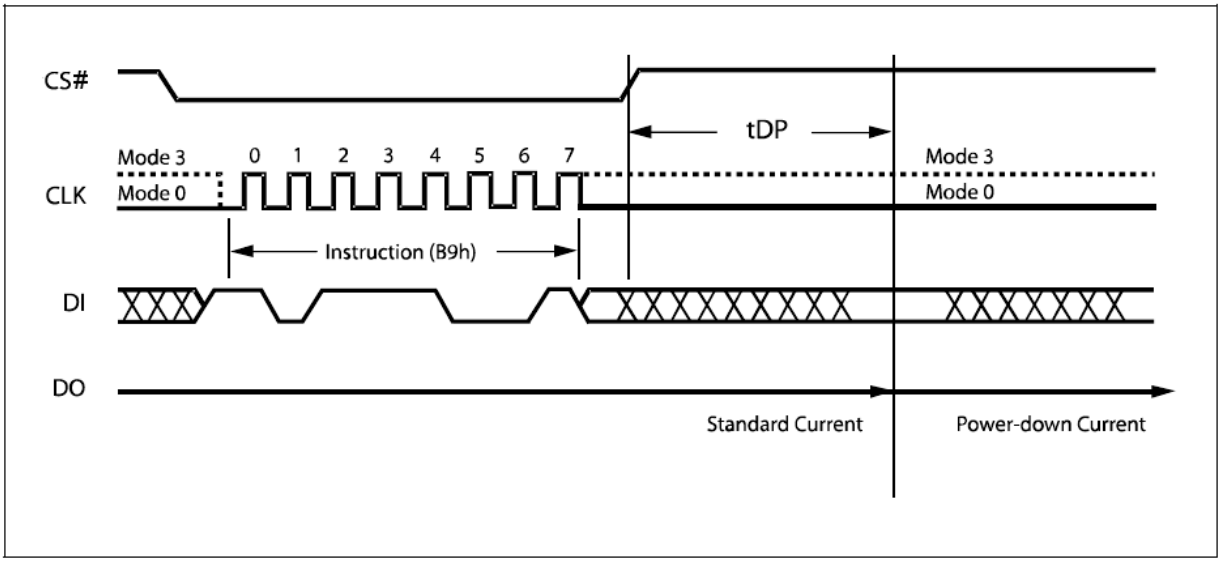
Driving Chip Select (CS#) High deselects the device, and puts the device in the Standby mode (if there is no internal cycle currently in progress). But this mode is not the Deep Power-down mode. The Deep Power-down mode can only be entered by executing the Deep Power-down (DP) instruction, to reduce the standby current (from $I_{CC1}$ to $I_{CC2}$, as specified in DC Characteristics table.)

Once the device has entered the Deep Power-down mode, all instructions are ignored except the Release from Deep Power-down, Read Device ID (RDI) or Software Reset instruction which release the device from this mode. The Release from Deep Power-down and Read Device ID (RDI) instruction also allows the Device ID of the device to be output on Serial Data Output (DO).

The Deep Power-down mode automatically stops at Power-down, and the device always Powers-up in the Standby mode. The Deep Power-down (DP) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code on Serial Data Input (DI). Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in Deep Power-down Instruction Sequence Diagram figure. Chip Select (CS#) must be driven High after the eighth bit of the instruction code has been latched in, otherwise the Deep Power-down (DP) instruction is not executed. As soon as Chip Select (CS#) 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 (DP) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.




**Deep Power-down Instruction Sequence Diagram**

### Release from Deep Power-down and Read Device ID (RDI)

Once the device has entered the Deep Power-down mode, all instructions are ignored except the Release from Deep Power-down and Read Device ID (RDI) instruction. Executing this instruction takes the device out of the Deep Power-down mode.

Please note that this is not the same as, or even a subset of, the JEDEC 16-bit Electronic Signature that is read by the Read Identifier (RDID) instruction. The old-style Electronic Signature is supported for reasons of backward compatibility, only, and should not be used for new designs. New designs should, instead, make use of the JEDEC 16-bit Electronic Signature, and the Read Identifier (RDID) instruction.

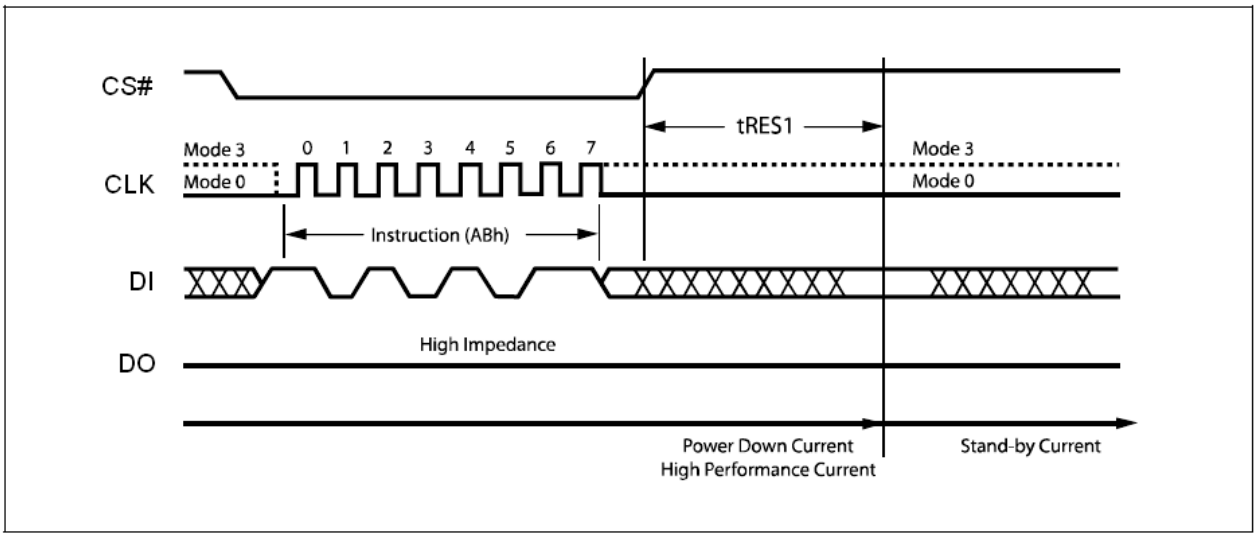
When used only to release the device from the power-down state, the instruction is issued by driving the CS# pin low, shifting the instruction code “ABh” and driving CS# high as shown in Release Power-down Instruction Sequence Diagram figure. After the time duration of $t_{RES1}$ (See AC Characteristics) the device will resume normal operation and other instructions will be accepted. The CS# pin must remain high during the $t_{RES1}$ time duration.

When used only to obtain the Device ID while not in the power-down state, the instruction is initiated by driving the CS# pin low and shifting the instruction code “ABh” followed by 3-dummy bytes. The Device ID bits are then shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Release Power-down / Device ID Instruction Sequence Diagram figure. The Device ID value for the device is listed in Manufacturer and Device Identification table. The Device ID can be read continuously. The instruction is completed by driving CS# high.

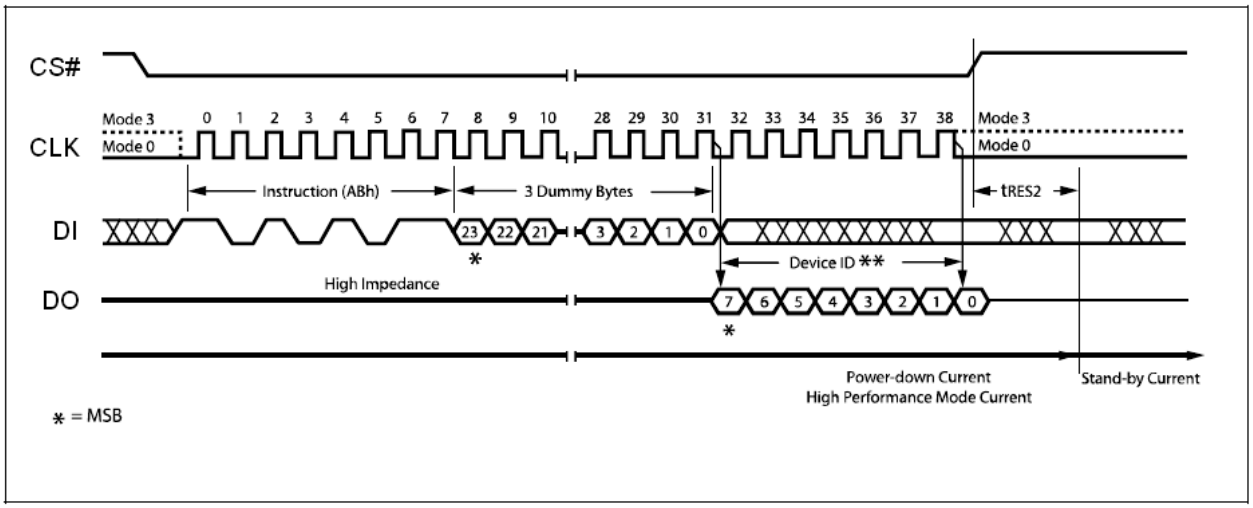
When Chip Select (CS#) is driven High, the device is put in the Stand-by Power mode. If the device was not previously in the Deep Power-down mode, the transition to the Stand-by Power mode is immediate. If the device was previously in the Deep Power-down mode, though, the transition to the Standby Power mode is delayed by $t_{RES2}$, and Chip Select (CS#) must remain High for at least $t_{RES2}$ (max), as specified in AC Characteristics table. Once in the Stand-by Power mode, the device waits to be selected, so that it can receive, decode and execute instructions.

Except while an Erase, Program or Write Status Register cycle is in progress, the Release from Deep Power-down and Read Device ID (RDI) instruction always provides access to the 8bit Device ID of the device, and can be applied even if the Deep Power-down mode has not been entered.

Any Release from Deep Power-down and Read Device ID (RDI) instruction while an Erase, Program or Write Status Register cycle is in progress, is not decoded, and has no effect on the cycle that is in progress.




**Release Power-down Instruction Sequence Diagram**

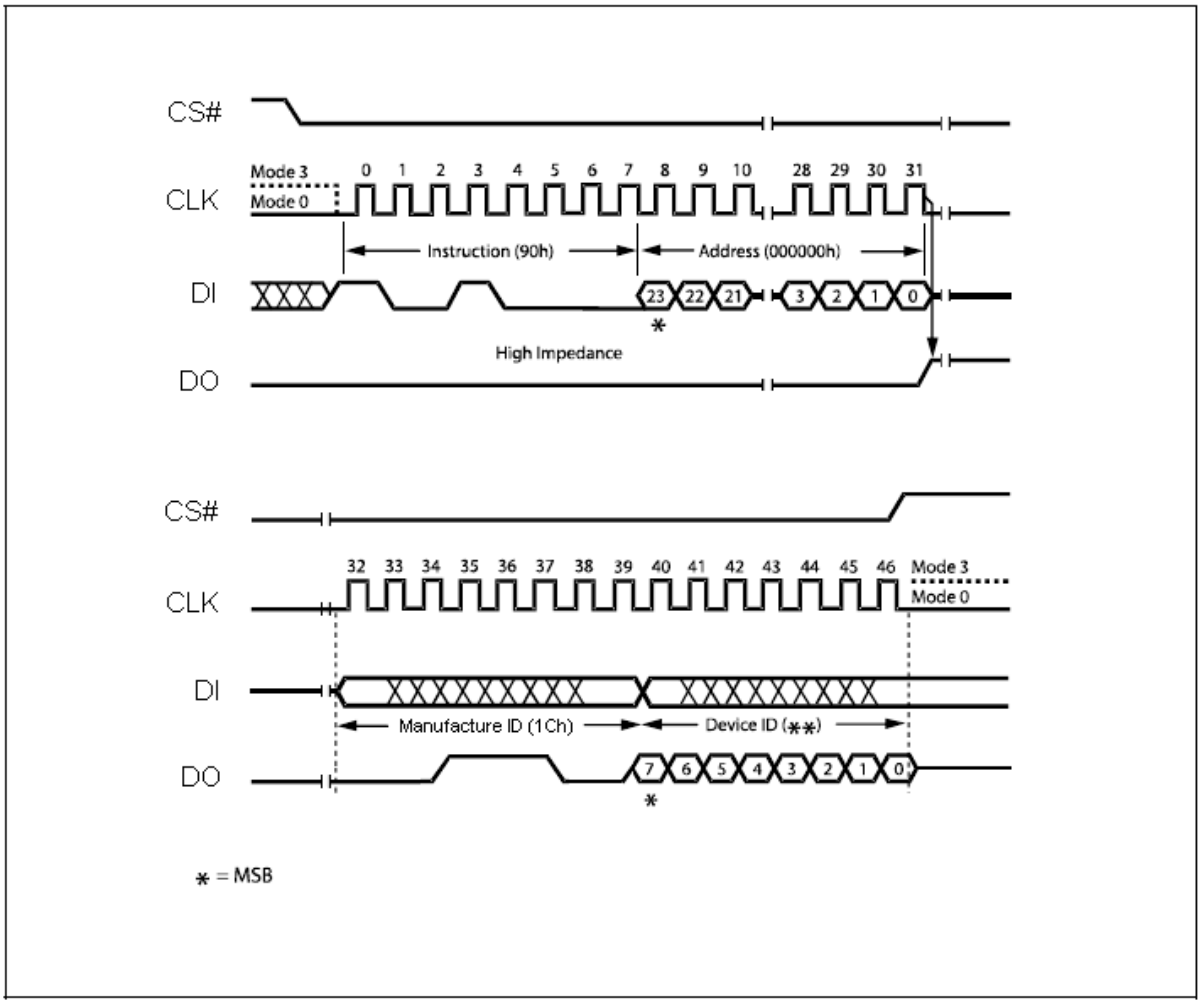



**Release Power-down / Device ID Instruction Sequence Diagram**

### Read Manufacturer / Device ID (90h)

The Read Manufacturer/Device ID instruction is an alternative to the Release from Power-down / Device ID instruction that provides both the JEDEC assigned manufacturer ID and the specific device ID.

The Read Manufacturer/Device ID instruction is very similar to the Release from Power-down / Device ID instruction. The instruction is initiated by driving the CS# pin low and shifting the instruction code “90h” followed by a 24-bit address (A23-A0) of 000000h. After which, the Manufacturer ID for Eon (1Ch) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Read Manufacturer / Device ID Diagram figure. The Device ID values for the device are listed in Manufacturer and Device Identification table. If the 24-bit address is initially set to 000001h the Device ID will be read first




**Read Manufacturer / Device ID Diagram**

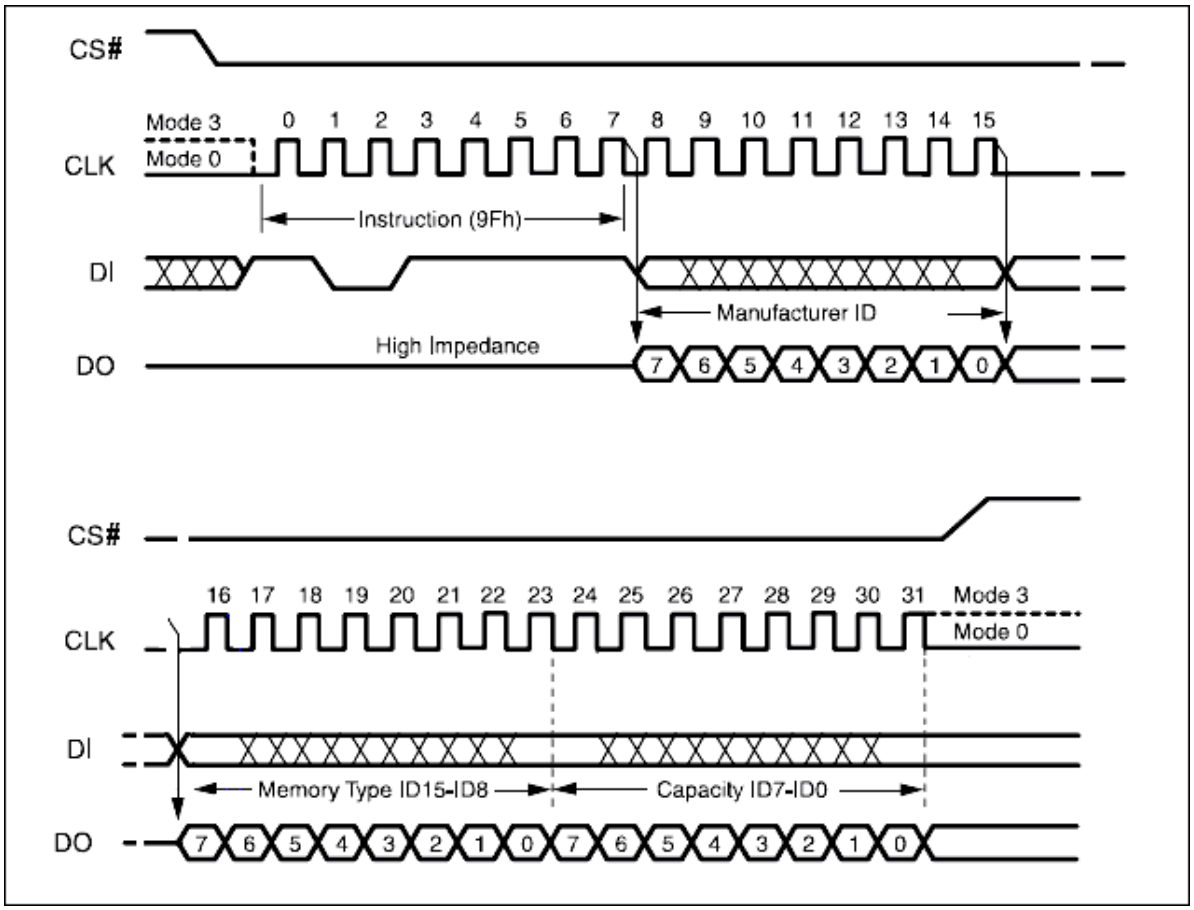
### Read Identification (RDID) (9Fh)

The Read Identification (RDID) instruction allows the 8-bit manufacturer identification to be read, followed by two bytes of device identification. The device identification indicates the memory type in the first byte, and the memory capacity of the device in the second byte.

Any Read Identification (RDID) instruction while an Erase or Program cycle is in progress, is not decoded, and has no effect on the cycle that is in progress. The Read Identification (RDID) instruction should not be issued while the device is in Deep Power down mode.

The device is first selected by driving Chip Select Low. Then, the 8-bit instruction code for the instruction is shifted in. This is followed by the 24-bit device identification, stored in the memory, being shifted out on Serial Data Output, each bit being shifted out during the falling edge of Serial Clock. The instruction sequence is shown in Read Identification (RDID) figure. The Read Identification (RDID) instruction is terminated by driving Chip Select High at any time during data output.

When Chip Select is driven High, the device is put in the Standby Power mode. Once in the Standby Power mode, the device waits to be selected, so that it can receive, decode and execute instructions.




**Read Identification (RDID)**

### Program OTP array (42h)

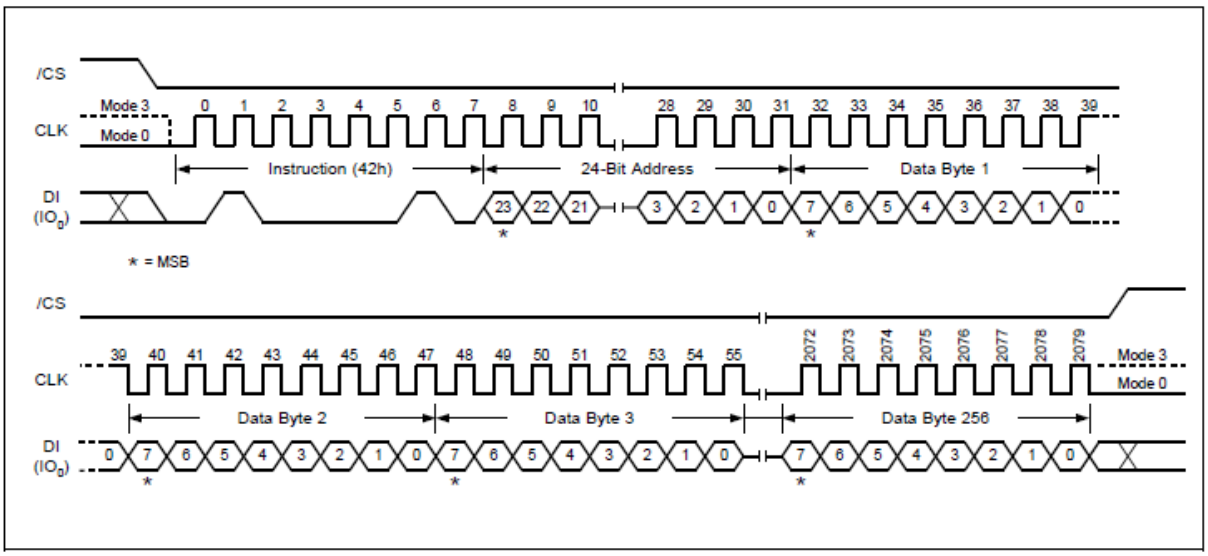
The Program OTP array operation is similar to the Page Program instruction. It allows from one byte to 256 bytes of security register data to be programmed at previously erased (FFh) memory locations. A Write Enable instruction must be executed before the device will accept the Program OTP array Instruction. The instruction is initiated by driving the CS# pin low then shifting the instruction code “42h” followed by a 24-bit address (A23-A0) and at least one data byte, into the DI pin. The CS# pin must be held low for the entire length of the instruction while data is being sent to the device.

The Program OTP array instruction sequence is shown in Program OTP array figure. The OTP array Lock Bits (SPL0-SPL3) in Status Register 2 can be used to OTP protect the OTP array data. Once a lock bit is set to 1, the corresponding OTP array will be permanently locked, Program OTP array instruction to that register will be ignored.

### OTP Sector Address

| Lock bit | Sector | Sector Size | Address Range |
|---|---|---|---|
| SPL0 | 255 | 1024 byte | 0FF000h – 0FF3FFh |
| SPL1 | 254 | 1024 byte | 0FE000h – 0FE3FFh |
| SPL2 | 253 | 1024 byte | 0FD000h – 0FD3FFh |

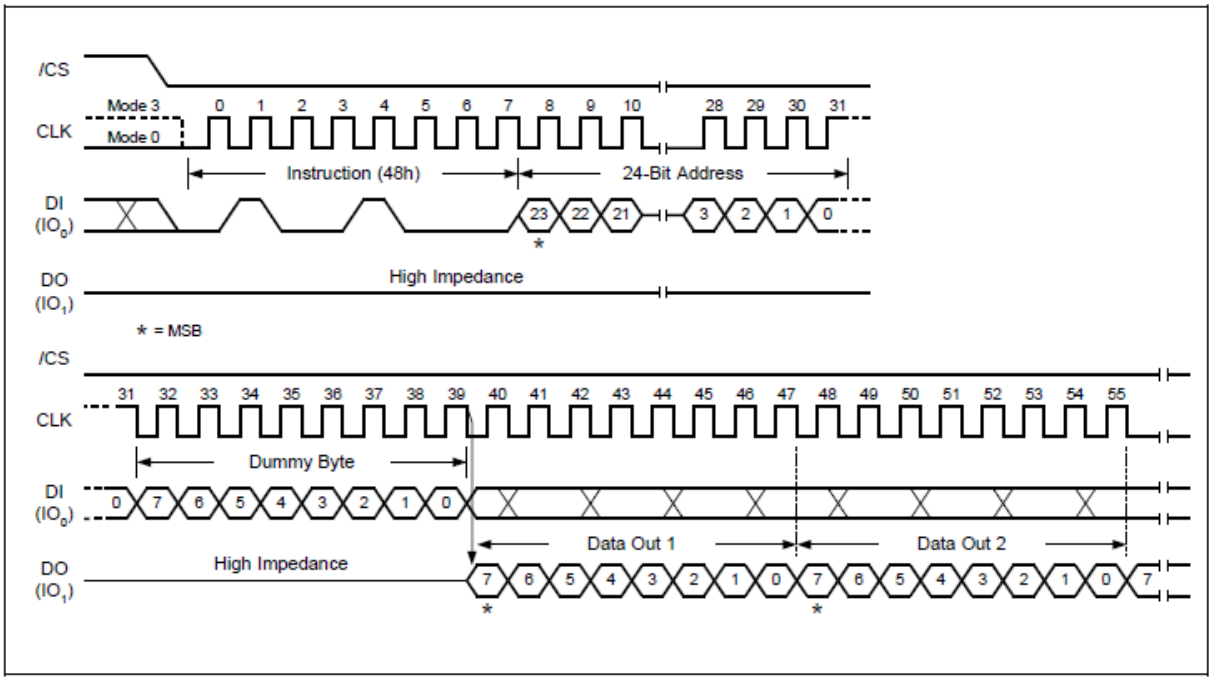
**Note:** The OTP sector is mapping to sector 255, 254 and 253




**Program OTP array**

### Read OTP array (48h)

The Read OTP array instruction is similar to the Fast Read instruction and allows one or more data bytes to be sequentially read from one of the three OTP array. The instruction is initiated by driving the CS# pin low and then shifting the instruction code “48h” followed by a 24-bit address (A23-A0) and eight “dummy” clocks into the DI pin. The code and address bits are latched on the rising edge of the CLK pin. After the address is received, the data byte of the addressed memory location will be shifted out on the DO pin at the falling edge of CLK with most significant bit (MSB) first. The byte address is automatically incremented to the next byte address after each byte of data is shifted out. Once the byte address reaches the last byte of the register (byte address FFh), it will reset to address 00h, the first byte of the register, and continue to increment. The instruction is completed by driving CS# high. The Read OTP array instruction sequence is shown in Read OTP array figure. If a Read OTP array instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle. The Read OTP array instruction allows clock rates from D.C. to a maximum of $F_R$ (see AC Electrical Characteristics).



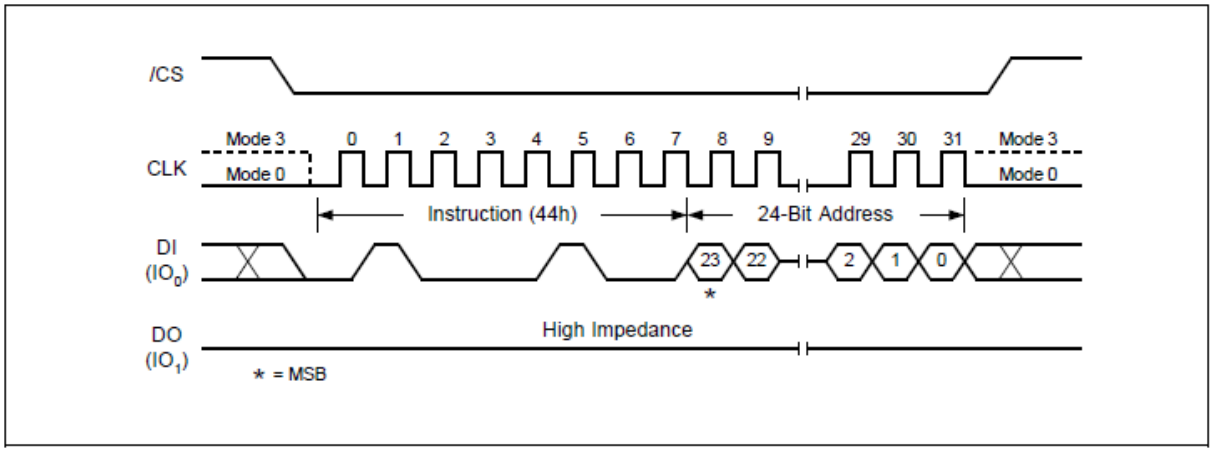

**Read OTP array**

## Erase OTP array (44h)

The device offers three set of 3x1024byte OTP array which can be erased and programmed individually. These registers may be used by the system manufacturers to store security and other important information separately from the main memory array.

The Erase OTP array instruction is similar to the Sector Erase instruction. A Write Enable instruction must be executed before the device will accept the Erase OTP array Instruction (Status Register bit WEL must equal 1). The instruction is initiated by driving the CS# pin low and shifting the instruction code “44h” followed by a 24-bit address (A23-A0) to erase one of the three security registers.

The Erase OTP array instruction sequence is shown in Erase OTP array figure. The CS# pin must be driven high after the eighth bit of the last byte has been latched. If this is not done the instruction will not be executed. After CS# is driven high, the self-timed Erase OTP array operation will commence for a time duration of $t_{SE}$ (See AC Characteristics). While the Erase OTP array cycle is in progress, the Read Status Register instruction may still be accessed for checking the status of the WIP bit. The WIP bit is a 1 during the erase cycle and becomes a 0 when the cycle is finished and the device is ready to accept other instructions again. After the Erase OTP array cycle has finished the Write Enable Latch (WEL) bit in the Status Register is cleared to 0. The Security Register Lock Bits (SPL0-3) in the Status Register 2 can be used to OTP protect the security registers. Once a lock bit is set to 1, the corresponding security register will be permanently locked, Erase OTP array instruction to that register will be ignored.



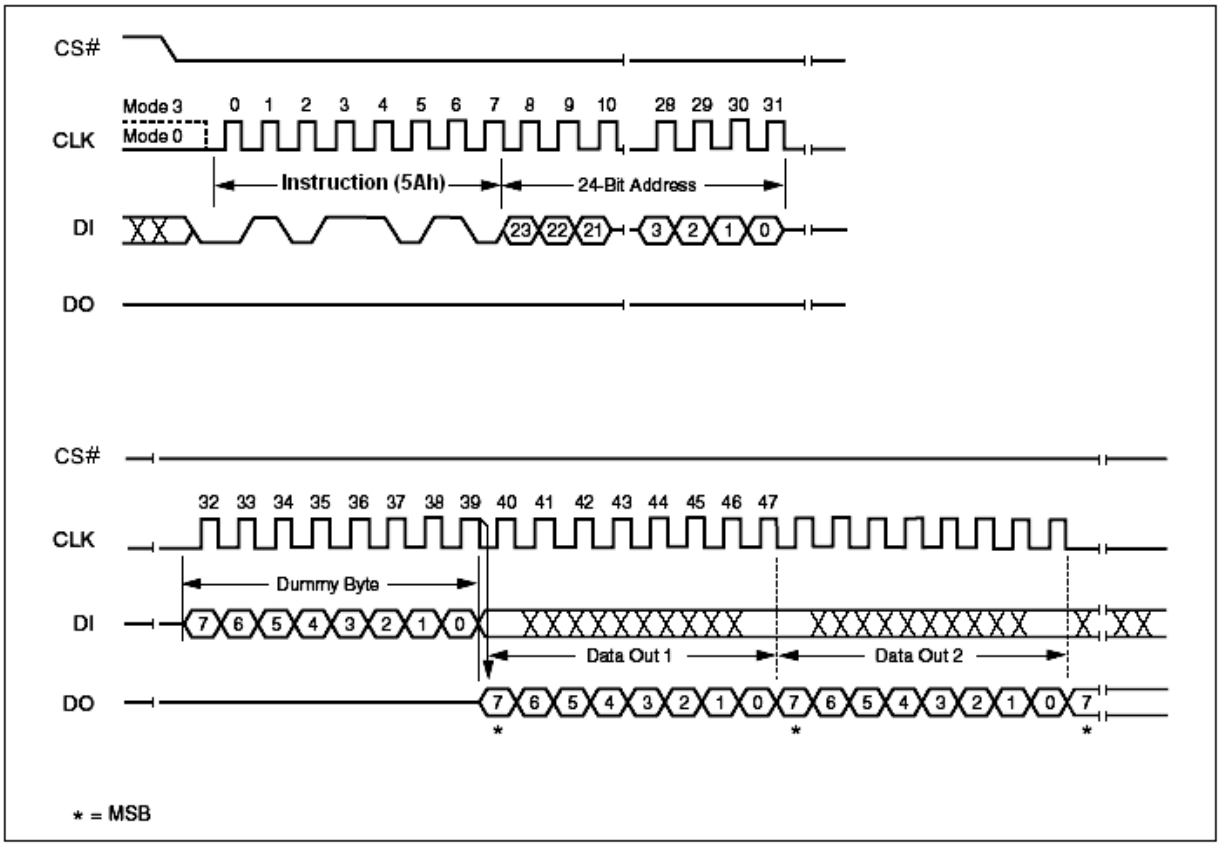

**Erase OTP array**

### Read SFDP Mode (5Ah)

Device features Serial Flash Discoverable Parameters (SFDP) mode. Host system can retrieve the operating characteristics, structure and vendor specified information such as identifying information, memory size, operating voltage and timing information of this device by SFDP mode.

The device is first selected by driving Chip Select (CS#) Low. The instruction code for the Read SFDP Mode 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 (CLK). Then the memory contents, at that address, is shifted out on Serial Data Output (DO), each bit being shifted out, at a maximum frequency $F_R$, during the falling edge of Serial Clock (CLK).

The instruction sequence is shown in Read SFDP Mode and Unique ID Number Instruction Sequence Diagram figure. 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. The whole memory can, therefore, be read with a single Serial Flash Discoverable Parameters (SFDP) instruction. When the highest address is reached, the address counter rolls over to 0x00h, allowing the read sequence to be continued indefinitely. The Serial Flash Discoverable Parameters (SFDP) instruction is terminated by driving Chip Select (CS#) High. Chip Select (CS#) can be driven High at any time during data output. Any Read Data Bytes at Serial Flash Discoverable Parameters (SFDP) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.




**Read SFDP Mode and Unique ID Number Instruction Sequence Diagram**

**Serial Flash Discoverable Parameters (SFDP) Signature and Parameter Identification Data Value (Advanced Information)**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (h) | Comment |
|---|---|---|---|---|
| **SFDP Signature** | 00h | 07 : 00 | 53h | Signature [31:0]: Hex: 50444653 |
| | 01h | 15 : 08 | 46h | |
| | 02h | 23 : 16 | 44h | |
| | 03h | 31 : 24 | 50h | |
| **SFDP Minor Revision Number** | 04h | 07 : 00 | 00h | Star from 0x00 |
| **SFDP Major Revision Number** | 05h | 15 : 08 | 01h | Star from 0x01 |
| **Number of Parameter Headers (NPH)** | 06h | 23 : 16 | 00h | 1 parameter header |
| **Unused** | 07h | 31 : 24 | FFh | Reserved |
| **ID Number** | 08h | 07 : 00 | 00h | JEDEC ID |
| **Parameter Table Minor Revision Number** | 09h | 15 : 08 | 00h | Star from 0x00 |
| **Parameter Table Major Revision Number** | 0Ah | 23 : 16 | 01h | Star from 0x01 |
| **Parameter Table Length (in DW)** | 0Bh | 31 : 24 | 09h | 9 DWORDs |
| **Parameter Table Pointer (PTP)** | 0Ch | 07 : 00 | 30h | 000030h |
| | 0Dh | 15 : 08 | 00h | |
| | 0Eh | 23 : 16 | 00h | |
| **Unused** | 0Fh | 31 : 24 | FFh | Reserved |

**Parameter ID (0) (Advanced Information) 1/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (b/h) | Data (h) | Comment |
|---|---|---|---|---|---|
| **Block / Sector Erase sizes** Identifies the erase granularity for all Flash Components | 30h | 00 | 01b | EDh | 00 = reserved<br>01 = 4KB erase<br>10 = reserved<br>11 = 64KB erase |
| | | 01 | | | |
| **Write Granularity** | | 02 | 1b | | 0 = No, 1 = Yes |
| **Write Enable Instruction Required for Writing to Volatile Status Register** | | 03 | 01b | | 00 = N/A<br>01 = use 50h opcode<br>11 = use 06h opcode |
| **Write Enable Opcode Select for Writing to Volatile Status Register** | | 04 | | | |
| **Unused** | | 05 | 111b | | Reserved |
| | | 06 | | | |
| | | 07 | | | |
| **4 Kilo-Byte Erase Opcode** | 31h | 08 | 20h | 20h | 4 KB Erase Support<br>(FFh = not supported) |
| | | 09 | | | |
| | | 10 | | | |
| | | 11 | | | |
| | | 12 | | | |
| | | 13 | | | |
| | | 14 | | | |
| | | 15 | | | |
| **Supports (1-1-2) Fast Read** Device supports single input opcode & address and dual output data Fast Read | 32h | 16 | 1b | F1h | 0 = not supported<br>1 = supported |
| **Address Byte** Number of bytes used in addressing for flash array read, write and erase. | | 17 | 00b | | 00 = 3-Byte<br>01 = 3- or 4-Byte (e.g. defaults to 3-Byte mode; enters 4-Byte mode on command)<br>10 = 4-Byte<br>11 = reserved |
| | | 18 | | | |
| **Supports Double Data Rate (DDR) Clocking** Indicates the device supports some type of double transfer rate clocking. | | 19 | 0b | | 0 = not supported<br>1 = supported |
| **Supports (1-2-2) Fast Read** Device supports single input opcode, dual input address, and dual output data Fast Read | | 20 | 1b | | 0 = not supported<br>1 = supported |
| **Supports (1-4-4) Fast Read** Device supports single input opcode, quad input address, and quad output data Fast Read | | 21 | 1b | | 0 = not supported<br>1 = supported |
| **Supports (1-1-4) Fast Read** Device supports single input opcode & address and quad output data Fast Read | | 22 | 1b | | 0 = not supported<br>1 = supported |
| **Unused** | | 23 | 1b | | Reserved |
| **Unused** | 33h | 24 | FFh | FFh | Reserved |
| | | 25 | | | |
| | | 26 | | | |
| | | 27 | | | |
| | | 28 | | | |
| | | 29 | | | |
| | | 30 | | | |
| | | 31 | | | |

**Parameter ID (0) (Advanced Information) 2/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (h) | Comment |
|---|---|---|---|---|
| **Flash Memory Density** | 37h : 34h | 31 : 00 | 007FFFFFh | 8 Mbits |

**Parameter ID (0) (Advanced Information) 3/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (b/h) | Data (h) | Comment |
|---|---|---|---|---|---|
| **(1-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 38h | 00 | 00100b | 44h | 4 dummy clocks |
| | | 01 | | | |
| | | 02 | | | |
| | | 03 | | | |
| | | 04 | | | |
| **Quad Input Address Quad Output (1-4-4) Fast Read Number of Mode Bits** | | 05 | 010b | | 8 mode bits |
| | | 06 | | | |
| | | 07 | | | |
| **(1-4-4) Fast Read Opcode** Opcode for single input opcode, quad input address, and quad output data Fast Read. | 39h | 08 | EBh | EBh | |
| | | 09 | | | |
| | | 10 | | | |
| | | 11 | | | |
| | | 12 | | | |
| | | 13 | | | |
| | | 14 | | | |
| | | 15 | | | |
| **(1-1-4) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 3Ah | 16 | 01000b | 08h | |
| | | 17 | | | |
| | | 18 | | | |
| | | 19 | | | |
| | | 20 | | | |
| **(1-1-4) Fast Read Number of Mode Bits** | | 21 | 000b | | |
| | | 22 | | | |
| | | 23 | | | |
| **(1-1-4) Fast Read Opcode** Opcode for single input opcode & address and quad output data Fast Read. | 3Bh | 31 : 24 | 6Bh | 6Bh | |

**Parameter ID (0) (Advanced Information) 4/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (b/h) | Data (h) | Comment |
|---|---|---|---|---|---|
| **(1-1-2) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 3Ch | 00<br>01<br>02<br>03<br>04 | 01000b | 08h | 8 dummy clocks |
| **(1-1-2) Fast Read Number of Mode Bits** | | 05<br>06<br>07 | 000b | | Not Supported |
| **(1-1-2) Fast Read Opcode**<br>Opcode for single input opcode & address and dual output data Fast Read. | 3Dh | 15 : 08 | 3Bh | 3Bh | |
| **(1-2-2) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 3Eh | 16<br>17<br>18<br>19<br>20 | 00100b | 04h | 4 dummy clocks |
| **(1-2-2) Fast Read Number of Mode Bits** | | 21<br>22<br>23 | 000b | | Not Supported |
| **(1-2-2) Fast Read Opcode**<br>Opcode for single input opcode, dual input address, and dual output data Fast Read. | 3Fh | 31 : 24 | BBh | BBh | |

**Parameter ID (0) (Advanced Information) 5/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (b/h) | Data (h) | Comment |
|---|---|---|---|---|---|
| **Supports (2-2-2) Fast Read**<br>Device supports dual input opcode & address and dual output data Fast Read. | 40h | 00 | 0b | FEh | 0 = not supported<br>1 = supported |
| Reserved. These bits default to all 1's | | 01<br>02<br>03 | 111b | | Reserved |
| **Supports (4-4-4) Fast Read**<br>Device supports Quad input opcode & address and quad output data Fast Read. | | 04 | 1b | | 0 = not supported<br>1 = supported<br>(EQPI Mode) |
| Reserved. These bits default to all 1's | | 05<br>06<br>07 | 111b | | Reserved |
| Reserved. These bits default to all 1's | 43h : 41h | 31 : 08 | FFh | FFh | Reserved |

**Parameter ID (0) (Advanced Information) 6/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (b/h) | Data (h) | Comment |
|---|---|---|---|---|---|
| Reserved. These bits default to all 1's | 45h : 44h | 15 : 00 | FFh | FFh | Reserved |
| **(2-2-2) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 46h | 16 | 00000b | 00h | Not Supported |
| | | 17 | | | |
| | | 18 | | | |
| | | 19 | | | |
| | | 20 | | | |
| **(2-2-2) Fast Read Number of Mode Bits** | | 21 | 000b | | Not Supported |
| | | 22 | | | |
| | | 23 | | | |
| **(2-2-2) Fast Read Opcode** Opcode for dual input opcode & address and dual output data Fast Read. | 47h | 31 : 24 | FFh | FFh | Not Supported |

**Parameter ID (0) (Advanced Information) 7/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (b/h) | Data (h) | Comment |
|---|---|---|---|---|---|
| Reserved. These bits default to all 1's | 49h : 48h | 15 : 00 | FFh | FFh | Reserved |
| **(4-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 4Ah | 16 | 00h | 00h | Reserved |
| | | 17 | | | |
| | | 18 | | | |
| | | 19 | | | |
| | | 20 | | | |
| **(4-4-4) Fast Read Number of Mode Bits** | | 21 | 000b | | Reserved |
| | | 22 | | | |
| | | 23 | | | |
| **(4-4-4) Fast Read Opcode** Opcode for quad input opcode/address, quad output data Fast Read. | 4Bh | 31 : 24 | FFh | FFh | Reserved |

**Parameter ID (0) (Advanced Information) 8/9**

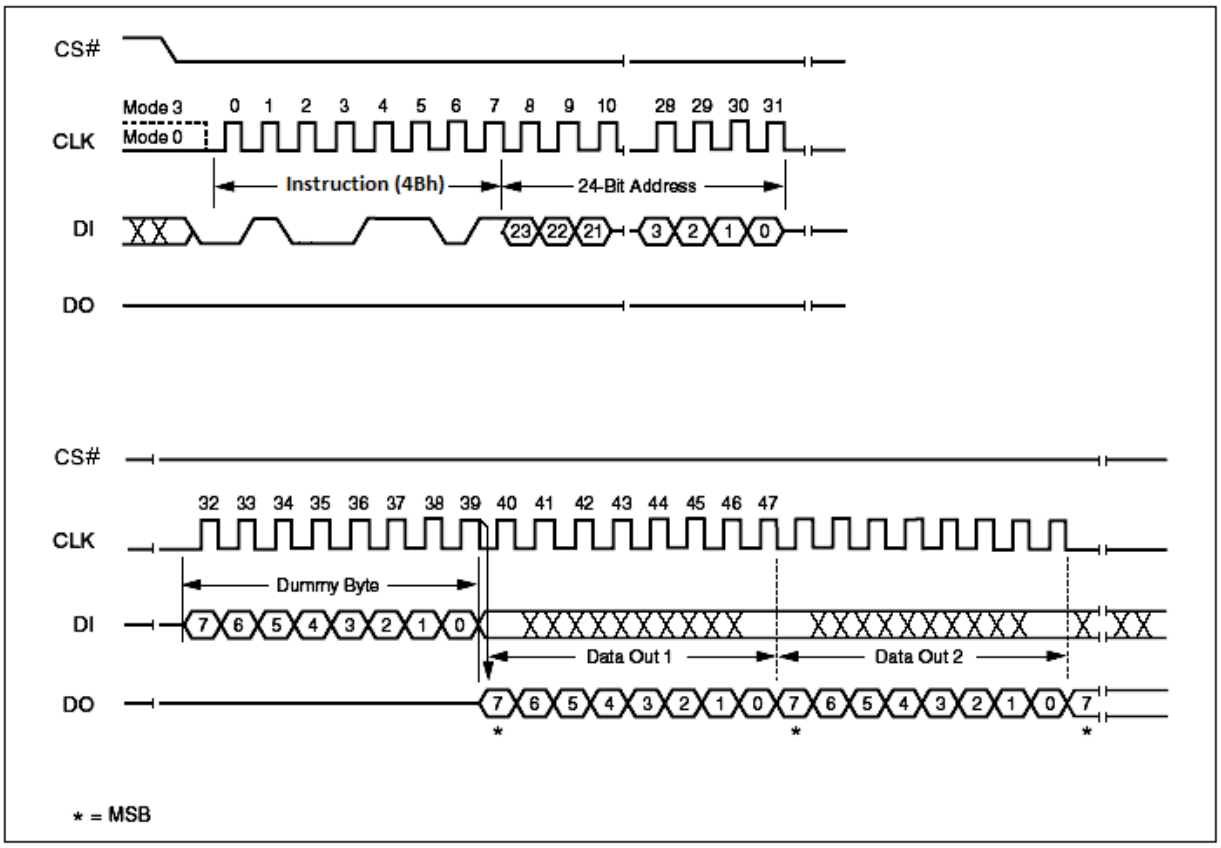
| Description | Address (h) (Byte Mode) | Address (Bit) | Data (h) | Comment |
|---|---|---|---|---|
| **Sector Type 1 Size** | 4Ch | 07 : 00 | 0Ch | 4 KB |
| **Sector Type 1 Opcode** | 4Dh | 15 : 08 | 20h | |
| **Sector Type 2 Size** | 4Eh | 23 : 16 | 0Fh | 32KB |
| **Sector Type 2 Opcode** | 4Fh | 31 : 24 | 52h | |

**Parameter ID (0) (Advanced Information) 9/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data (h) | Comment |
|---|---|---|---|---|
| **Sector Type 3 Size** | 50h | 07 : 00 | 10h | 64 KB |
| **Sector Type 3 Opcode** | 51h | 15 : 08 | D8h | |
| **Sector Type 4 Size** | 52h | 23 : 16 | 00h | Not Supported |
| **Sector Type 4 Opcode** | 53h | 31 : 24 | FFh | Not Supported |

### Read Unique ID Number (4Bh)

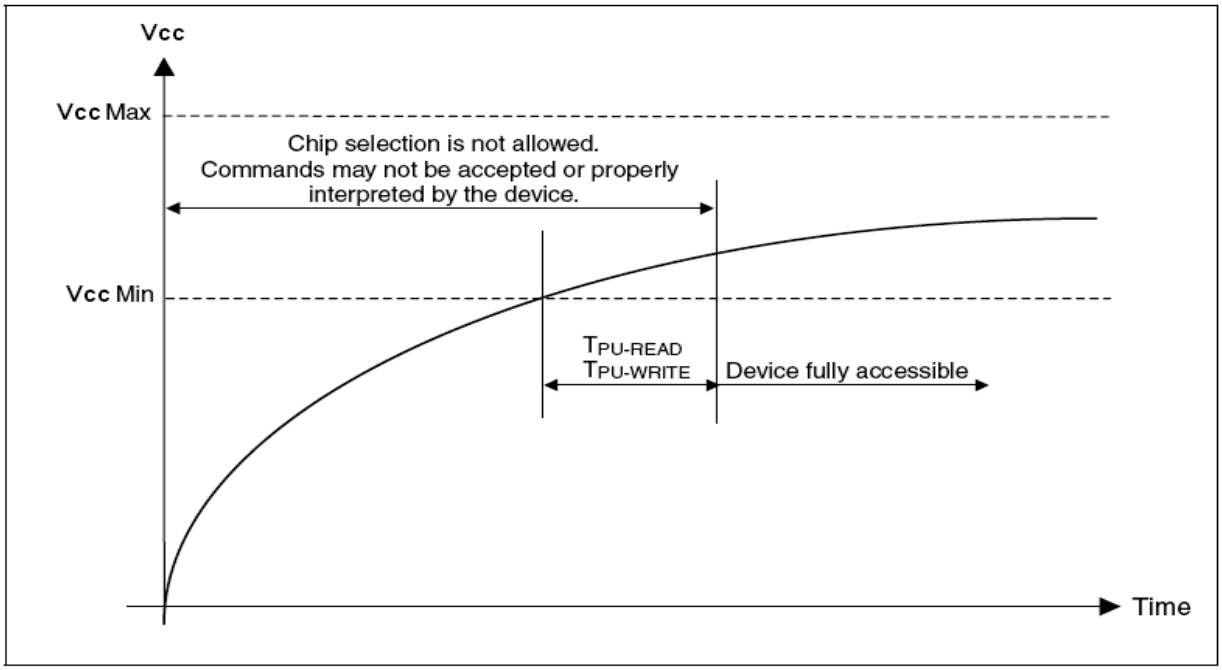
The Read Unique ID Number instruction accesses a factory-set read-only 128-bit number that is unique to each device. The ID number can be used in conjunction with user software methods to help prevent copying or cloning of a system. The Read Unique ID instruction is initiated by driving the CS# pin low and shifting the instruction code "4Bh" followed by a three bytes of addresses, 0x000000h, and one byte of dummy clocks. After which, the 128-bit ID is shifted out on the falling edge of CLK. (Read Unique ID Instruction Sequence Diagram figure)




**Read Unique ID Instruction Sequence Diagram**

## Power-up Timing

All functionalities and DC specifications are specified for a $V_{CC}$ ramp rate of greater than 1V per 100 ms (0V to 2.3V in less than 230 ms). See Power-Up Timing table and Power-up Timing figure for more information.




**Power-up Timing**

**Power-Up Timing**

| Symbol | Parameter | Min. | Unit |
|---|---|---|---|
| $T_{PU-READ}$ (1) | $V_{CC}$ Min to Read Operation | 100 | µs |
| $T_{PU-WRITE}$ (1) | $V_{CC}$ Min to Write Operation | 100 | µs |

**Note:**

1. This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

## INITIAL DELIVERY STATE

The device is delivered with the memory array erased: all bits are set to 1 (each byte contains FFh). The Status Register contains 00h (all Status Register bits are 0) except SR3.2 (blank check bit) equals to 1.

## DC Characteristics

($T_A$ = -40°C to 85°C; $V_{CC}$ = 2.3-3.6V)

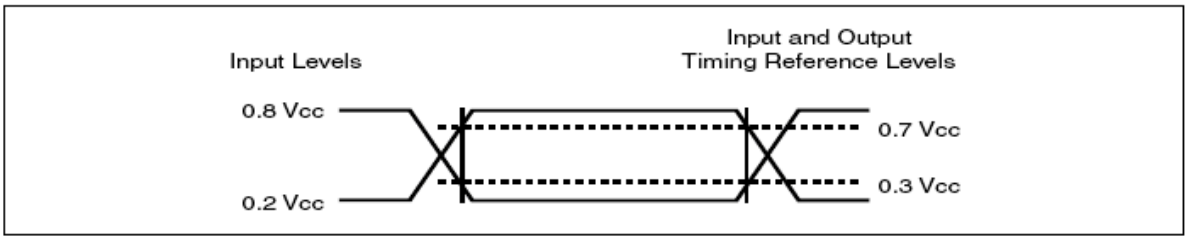
| Symbol | Parameter | Test Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Leakage Current | | - | 1 | ± 2 | µA |
| $I_{LO}$ | Output Leakage Current | | - | 1 | ± 2 | µA |
| $I_{CC1}$ | Standby Current | CS# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | - | 1 | 10 | µA |
| $I_{CC2}$ | Deep Power-down Current | CS# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | - | 1 | 10 | µA |
| $I_{CC3}$ | Operating Current (READ) | CLK = 0.1 $V_{CC}$ / 0.9 $V_{CC}$ at 104MHz, DQ = open | - | 10 | 16 | mA |
| | | CLK = 0.1 $V_{CC}$ / 0.9 $V_{CC}$ at 104MHz for Quad Output Read, DQ = open | - | 12 | 18 | mA |
| $I_{CC4}$ | Operating Current (PP) | CS# = $V_{CC}$ | - | - | 25 | mA |
| $I_{CC5}$ | Operating Current (WRSR/WRSR3) | CS# = $V_{CC}$ | - | - | 25 | mA |
| $I_{CC6}$ [1] | Operating Current (SE) | CS# = $V_{CC}$ | - | 13 | 25 | mA |
| $I_{CC7}$ [1] | Operating Current (HBE/BE) | CS# = $V_{CC}$ | - | 13 | 25 | mA |
| $V_{IL}$ | Input Low Voltage | | -0.5 | - | 0.2 $V_{CC}$ | V |
| $V_{IH}$ | Input High Voltage | | 0.7$V_{CC}$ | - | Vcc+0.4 | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 100 µA, $V_{CC}$=$V_{CC}$ Min. | - | - | 0.3 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -100 µA , $V_{CC}$=$V_{CC}$ Min. | $V_{CC}$-0.2 | - | - | V |

**Note:**

1. Erase current measure on all cells = '0' state.
2. Typical value of $I_{CC3}$@104MHz at T = 25℃, $V_{CC}$ = 3.3V.
3. Value guaranteed by design and/or characterization, not 100% tested in production.

## AC Measurement Conditions

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $C_L$ | Load Capacitance | 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 |




**AC Measurement I/O Waveform**

## AC Characteristics

($T_A$ = -40°C to 85°C; $V_{CC}$ = 2.3-3.6V)

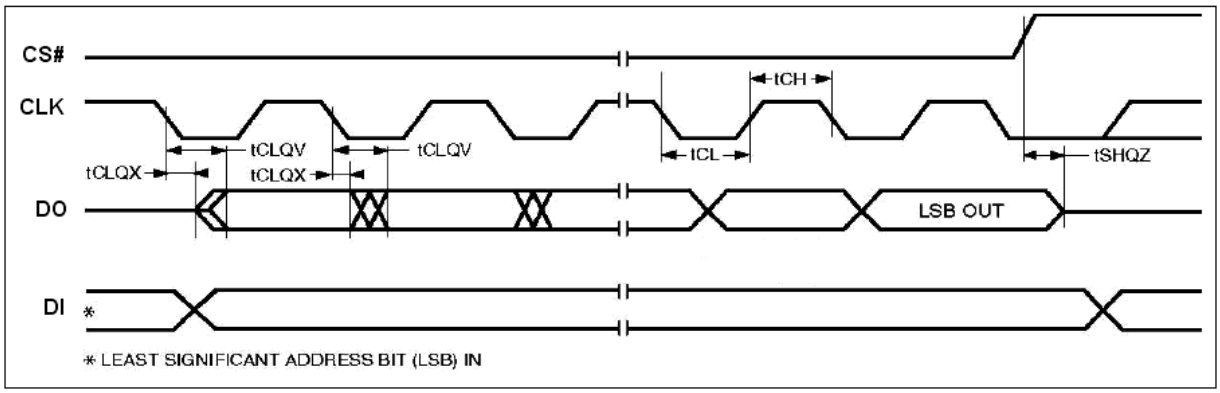
| Symbol | Alt | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_R$ | $f_C$ | Serial Clock Frequency for all commands, 0xBBh-0xEB Read at DC=1, at $V_{CC}$= 2.3-3.6V | - | - | 104 | MHz |
| | | Serial Clock Frequency for all commands, 0xBBh-0xEB Read at DC=0, at $V_{CC}$= 2.3-3.6V | - | - | 66 | MHz |
| $f_R$ | | Serial Clock Frequency for READ(0x03h Read) | D.C. | - | 50 | MHz |
| $t_{CH}$[1] | | Serial Clock High Time for SDR | 45% (1/Fc) | - | - | ns |
| $t_{CL}$[1] | | Serial Clock Low Time for SDR | 45% (1/Fc) | - | - | ns |
| $t_{CLCH}$[2] | | Serial Clock Rise Time (Slew Rate) | 0.2 | - | - | V / ns |
| $t_{CHCL}$[2] | | Serial Clock Fall Time (Slew Rate) | 0.2 | - | - | V / ns |
| $t_{SLCH}$ | $t_{CSS}$ | CS# Active Setup Time | 10 | - | - | ns |
| $t_{CHSH}$ | | CS# Active Hold Time | 10 | - | - | ns |
| $t_{SHCH}$ | | CS# Not Active Setup Time | 10 | - | - | ns |
| $t_{CHSL}$ | | CS# Not Active Hold Time | 10 | - | - | ns |
| $t_{SHSL}$ | $t_{CSH}$ | CS# High Time | 40 | - | - | ns |
| $t_{SHSL}$[2] | $t_{CSH}$ | Volatile Register Write Time | 50 | - | - | ns |
| $t_{SHQZ}$[2] | $t_{DIS}$ | Output Disable Time | - | - | 12 | ns |
| $t_{CLQX}$ | $t_{HO}$ | Output Hold Time | 1.2 | - | - | ns |
| $t_{DVCH}$ | $t_{DSU}$ | Data In Setup Time | 4 | - | - | ns |
| $t_{CHDX}$ | $t_{DH}$ | Data In Hold Time | 4 | - | - | ns |
| $t_{HLCH}$ | | HOLD# Low Setup Time ( relative to CLK ) | 5 | - | - | ns |
| $t_{HHCH}$ | | HOLD# High Setup Time ( relative to CLK ) | 5 | - | - | ns |
| $t_{CHHH}$ | | HOLD# Low Hold Time ( relative to CLK ) | 5 | - | - | ns |
| $t_{CHHL}$ | | HOLD# High Hold Time ( relative to CLK ) | 5 | - | - | ns |
| $t_{HLQZ}$[2] | $t_{HZ}$ | HOLD# to Output High-Z | - | - | 10 | ns |
| $t_{HHQX}$[2] | $t_{LZ}$ | HOLD# to Output Low-Z | - | - | 10 | ns |
| $t_{CLQV}$ | $t_V$ | Output Valid from CLK for 30pF | - | - | 8 | ns |
| | | Output Valid from CLK for 15pF | - | - | 6 | ns |
| $t_{WHSL}$[3] | | Write Protect Setup Time before CS# Low | 20 | - | - | ns |
| $t_{SHWL}$[3] | | Write Protect Hold Time after CS# High | 100 | - | - | ns |
| $t_{DP}$[2] | | CS# High to Deep Power-down Mode | - | - | 3 | µs |
| $t_{RES1}$[2] | | CS# High to Standby Mode without Electronic Signature read | - | - | 30 | µs |
| $t_{RES2}$[2] | | CS# High to Standby Mode with Electronic Signature read | - | - | 30 | µs |

## AC Characteristics-Continued

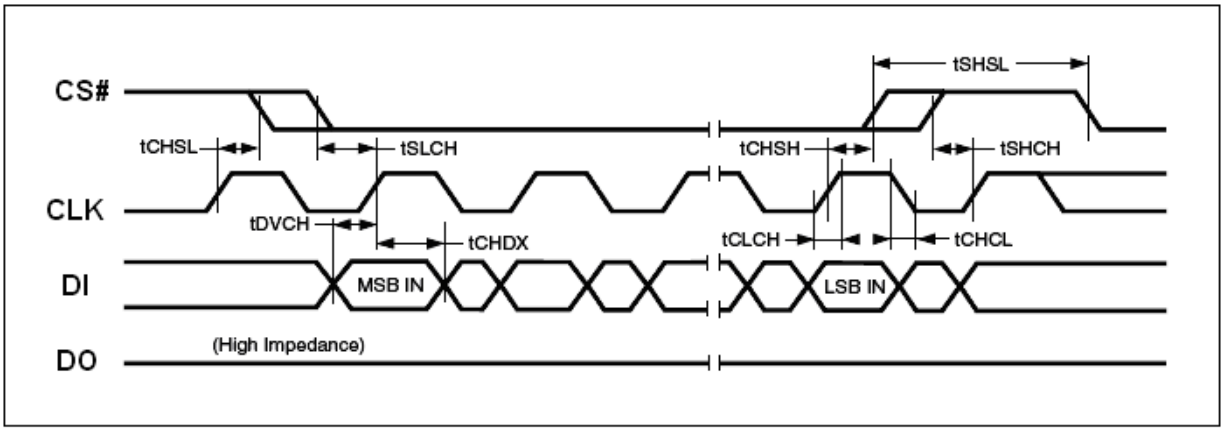
| Symbol | Alt | Parameter | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| $t_W$ | | Write Status Register Cycle Time | | - | 4 | 30 | ms |
| $t_{PP}$ | | Page Programming Time | | - | 1 | 4 | ms |
| $t_{SE}$ | | Sector Erase Time | | - | 0.1 | 0.5 | s |
| $t_{HBE}$ | | Half Block Erase Time | | - | 0.3 | 2 | s |
| $t_{BE}$ | | Block Erase Time | | - | 0.5 | 3 | s |
| $t_{CE}$ | | Chip Erase Time | | - | 8 | 18 | s |
| | $t_{SR}$ | Software Reset Latency | WIP = write operation | - | - | 28 | µs |
| | | | WIP = not in write operation | - | - | 0 | µs |

**Note:**

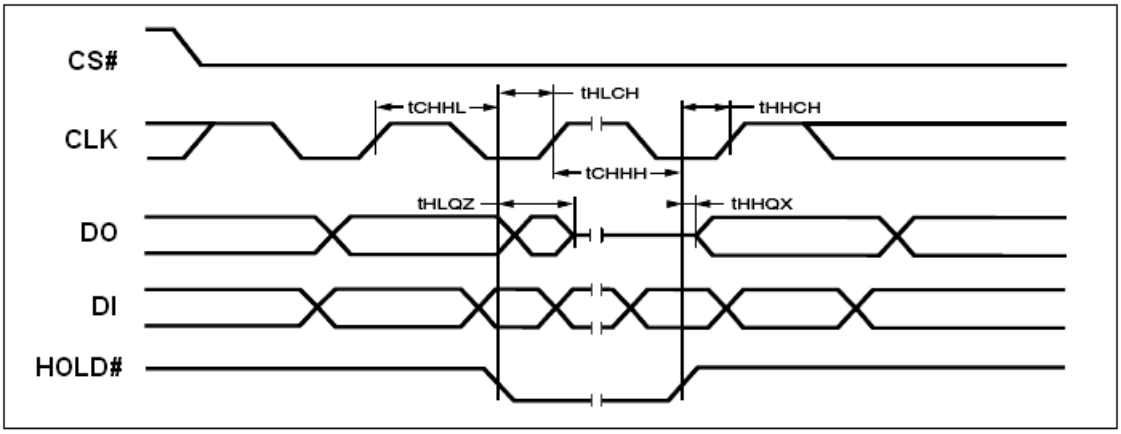
1. $t_{CH} + t_{CL}$ must be greater than or equal to $1/ f_C$
2. Value guaranteed by characterization, not 100% tested in production.
3. Only applicable as a constraint for a Write status Register instruction when Status Register Protect Bit is set at 1.
4. Typical value at T=25℃




**Serial Output Timing**




**Input Timing**




**Hold Timing**

## ABSOLUTE MAXIMUM RATINGS

Stresses above the values so mentioned above may cause permanent damage to the device. These values are for a stress rating only and do not imply that the device should be operated at conditions up to or above these values. Exposure of the device to the maximum rating values for extended periods of time may adversely affect the device reliability.

| Parameter | Value | Unit |
|---|---|---|
| Storage Temperature | -65 to +150 | °C |
| Output Short Circuit Current [1] | 200 | mA |
| Input and Output Voltage (with respect to ground) [2] | -0.5 to $V_{CC}$+0.5 | V |
| $V_{CC}$ | -0.5 to $V_{CC}$+0.5 | V |

**Note:**

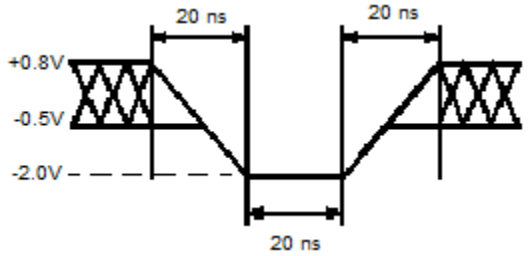
1. No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.
2. Minimum DC voltage on input or I/O pins is -0.5V. During voltage transitions, inputs may undershoot $V_{SS}$ to -1.0V for periods of up to 50ns and to -2.0V for periods of up to 20ns. See figure below. Maximum DC voltage on output and I/O pins is $V_{CC}$ + 0.5V. During voltage transitions, outputs may overshoot to $V_{CC}$ + 2.0V for periods up to 20ns. See figure below.

## RECOMMENDED OPERATING RANGES [1]

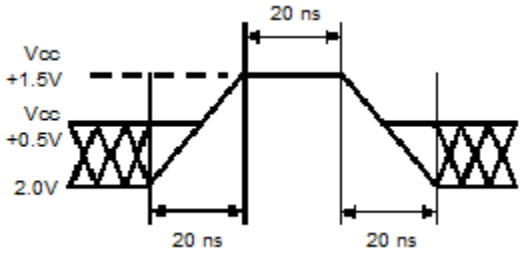
| Parameter | Value | Unit |
|---|---|---|
| Ambient Operating Temperature Industrial Devices | -40 to 85 | °C |
| Operating Supply Voltage $V_{CC}$ | Full: 2.3 to 3.6 | V |

**Note:**

1. Recommended Operating Ranges define those limits between which the functionality of the device is guaranteed.




Maximum Negative Overshoot Waveform




Maximum Positive Overshoot Waveform

## CAPACITANCE

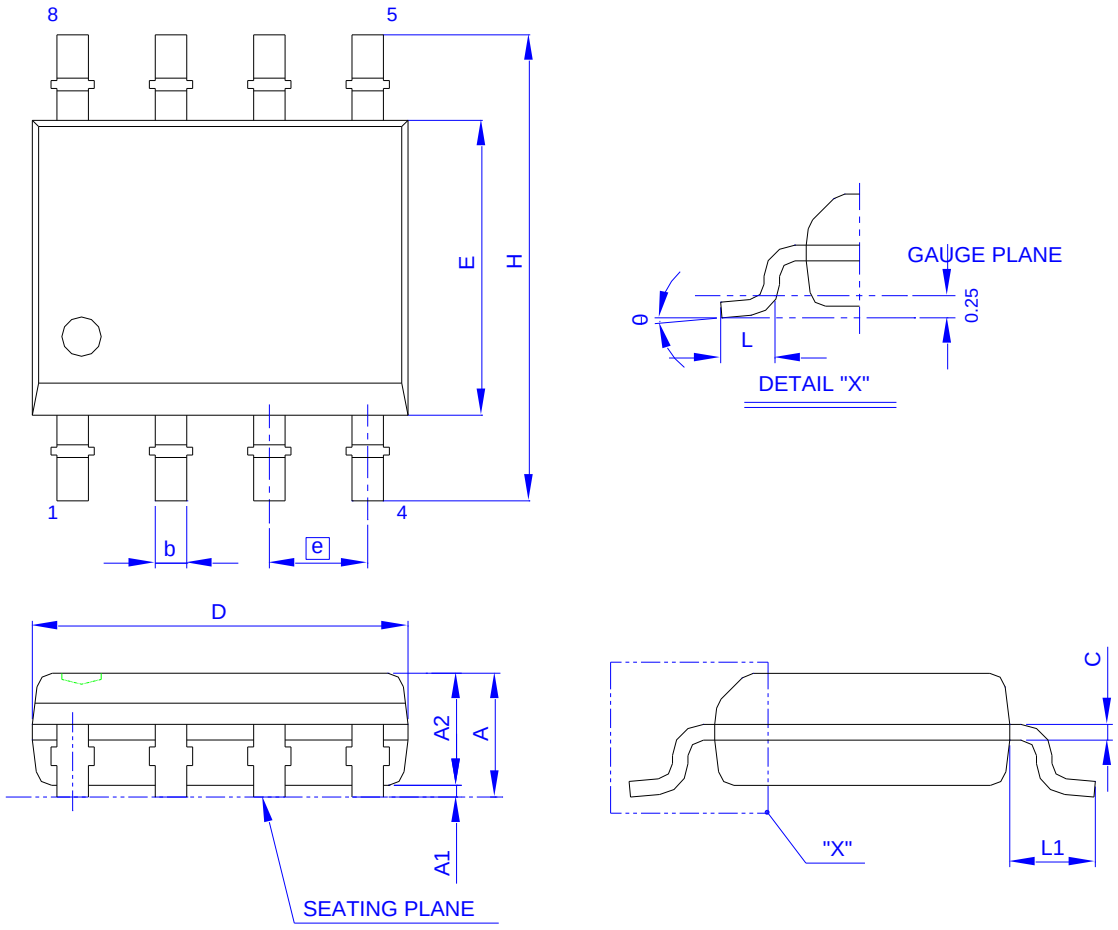
($V_{CC}$ = 2.3-3.6V)

| Parameter Symbol | Parameter Description | Test Setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0 | - | 6 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0 | - | 8 | pF |

**Note:** Sampled only, not 100% tested, at $T_A$ = 25°C and a frequency of 20MHz.

## PACKAGE MECHANICAL

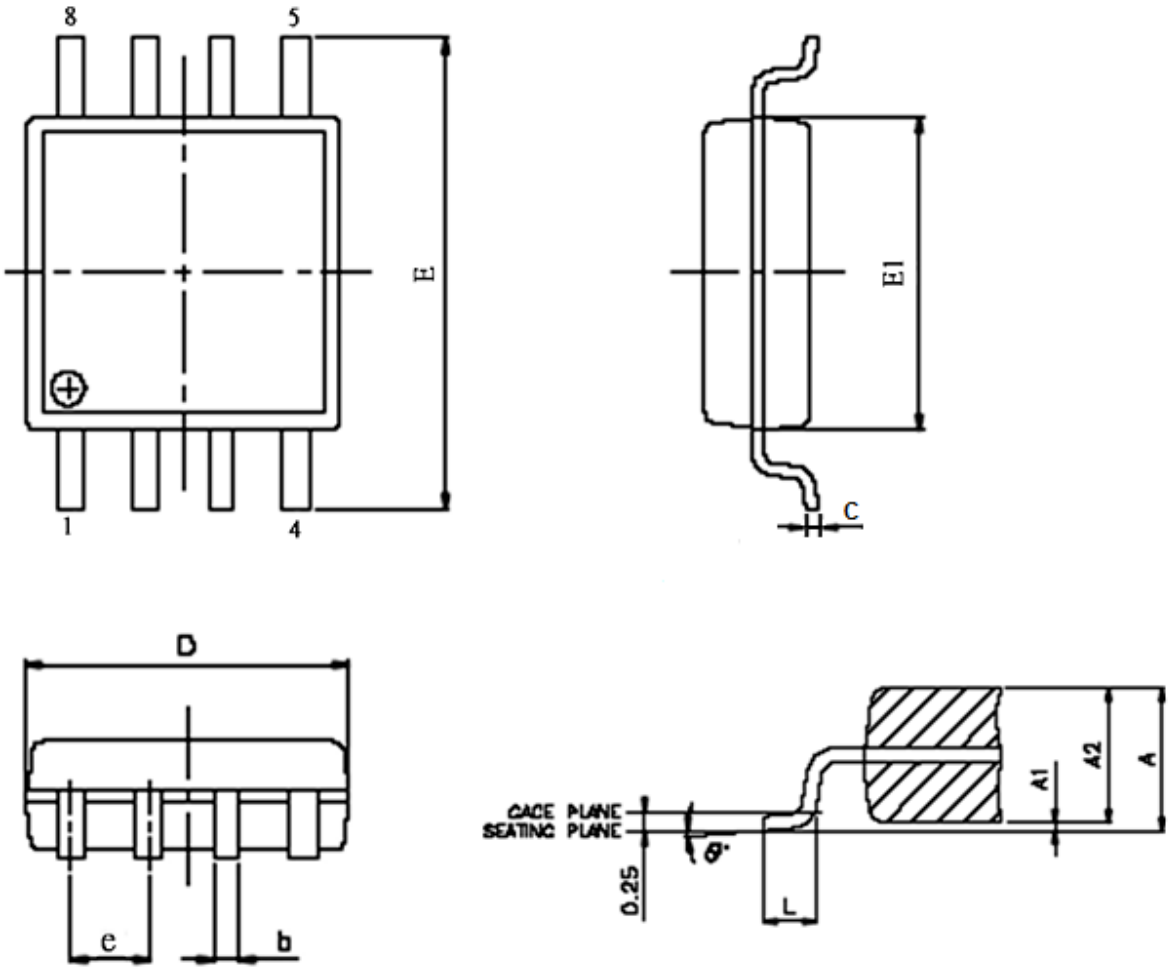
### SOP 8 (150 mil)




| Symbol | Dimension in mm | | | Dimension in inch | | | Symbol | Dimension in mm | | | Dimension in inch | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Norm | Max | Min | Norm | Max | | Min | Norm | Max | Min | Norm | Max |
| A | 1.35 | 1.60 | 1.75 | 0.053 | 0.063 | 0.069 | D | 4.80 | 4.90 | 5.00 | 0.189 | 0.193 | 0.197 |
| $A_1$ | 0.10 | 0.15 | 0.25 | 0.004 | 0.006 | 0.010 | E | 3.80 | 3.90 | 4.00 | 0.150 | 0.154 | 0.157 |
| $A_2$ | 1.25 | 1.45 | 1.55 | 0.049 | 0.057 | 0.061 | L | 0.40 | 0.66 | 0.86 | 0.016 | 0.026 | 0.034 |
| b | 0.33 | 0.406 | 0.51 | 0.013 | 0.016 | 0.020 | e | 1.27 BSC | | | 0.050 BSC | | |
| c | 0.19 | 0.203 | 0.25 | 0.0075 | 0.008 | 0.010 | $L_1$ | 1.00 | 1.05 | 1.10 | 0.039 | 0.041 | 0.043 |
| H | 5.80 | 6.00 | 6.20 | 0.228 | 0.236 | 0.244 | θ | 0° | --- | 8° | 0° | --- | 8° |

**Controlling dimension : millimenter**

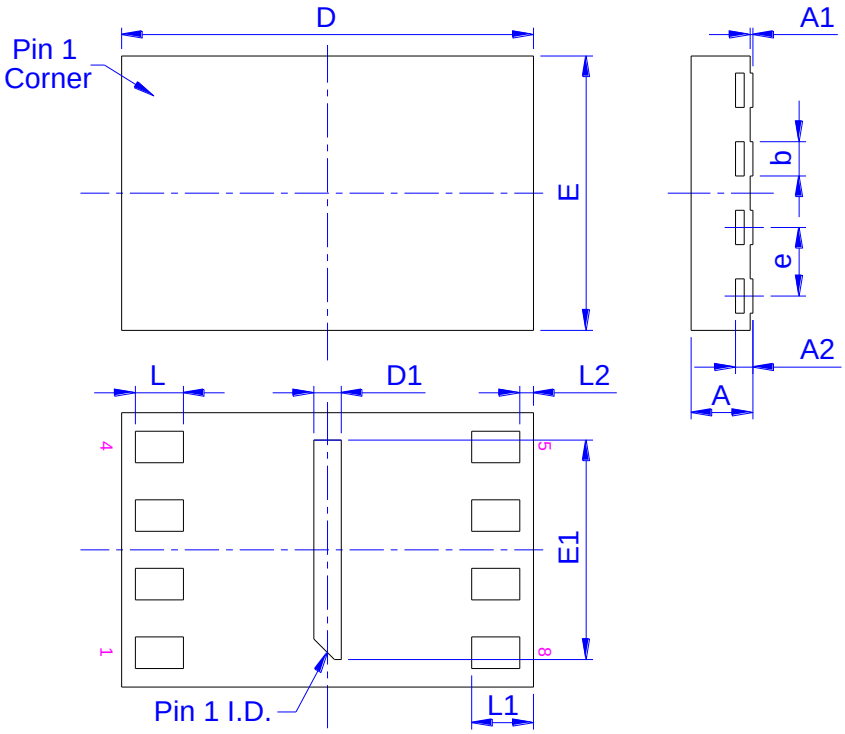
**SOP 200 mil (official name = 208 mil)**




| SYMBOL | DIMENSION IN MM | | |
|---|---|---|---|
| | MIN. | NOR | MAX |
| A | 1.75 | 1.975 | 2.20 |
| A1 | 0.05 | 0.15 | 0.25 |
| A2 | 1.70 | 1.825 | 1.95 |
| D | 5.15 | 5.275 | 5.40 |
| E | 7.70 | 7.90 | 8.10 |
| E1 | 5.15 | 5.275 | 5.40 |
| e | - - - | 1.27 | - - - |
| b | 0.35 | 0.425 | 0.50 |
| C | 0.19 | 0.200 | 0.25 |
| L | 0.5 | 0.65 | 0.80 |
| θ | $0^0$ | $4^0$ | $8^0$ |

Note : 1. Coplanarity: 0.1 mm
2. Max. allowable mold flash is 0.15 mm
at the pkg ends, 0.25 mm between leads.

**USON (8L 3x2x0.45 mm)**



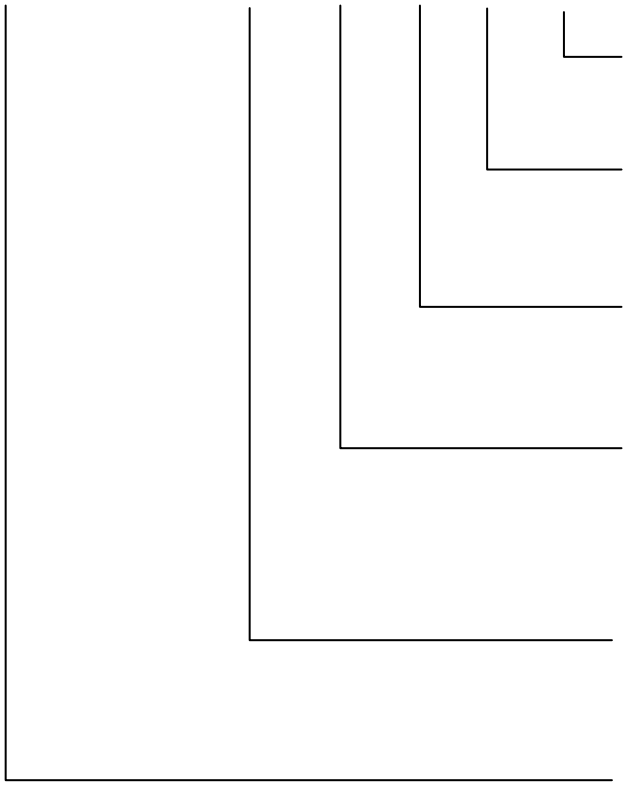

| Symbol | Dimension in mm | | | Dimension in inch | | |
|---|---|---|---|---|---|---|
| | Min. | Norm. | Max. | Min. | Norm. | Max. |
| **A** | 0.40 | 0.45 | 0.50 | 0.016 | 0.018 | 0.020 |
| **A1** | 0.00 | 0.02 | 0.05 | 0.000 | 0.001 | 0.002 |
| **A2** | | 0.152 REF | | | 0.006 REF | |
| **b** | 0.20 | 0.25 | 0.30 | 0.008 | 0.010 | 0.012 |
| **D** | 2.90 | 3.00 | 3.10 | 0.114 | 0.118 | 0.122 |
| **D1** | 0.10 | 0.20 | 0.30 | 0.004 | 0.008 | 0.012 |
| **E** | 1.90 | 2.00 | 2.10 | 0.075 | 0.079 | 0.083 |
| **E1** | 1.50 | 1.60 | 1.70 | 0.059 | 0.063 | 0.067 |
| **e** | | 0.50 BSC | | | 0.020 BSC | |
| **L** | 0.30 | - | - | 0.012 | - | - |
| **L1** | 0.40 | 0.45 | 0.50 | 0.016 | 0.018 | 0.020 |
| **L2** | - | - | 0.15 | - | - | 0.006 |

Controlling dimension : millimeter

(Revision date : Nov 20 2020)

## ORDERING INFORMATION

EN25QE80A - 104 H I P 2P



**DIFFERENTIATION CODE**

**PACKAGING CONTENT**
P = RoHS, Halogen-Free and REACH compliant

**TEMPERATURE RANGE**
I = Industrial (-40°C to +85°C)

**PACKAGE**
G = 8 pin 150mil SOP
H = 8 pin 200mil SOP
XF = 8 contact USON (3x2x0.45mm)

**SPEED**
104 = 104 MHz

**BASE PART NUMBER**
EN = Eon Silicon Solution Inc.
25QE = 2.3-3.6V Serial Flash with 4KB Uniform-Sector
80 = 8 Megabit (1024K x 8)
A = version identifier

## Revisions List

| Revision No | Description | Date |
|---|---|---|
| Preliminary 0.1 | Initial Release | 2022.07.15 |
| 1.0 | Version upgrade | 2023.12.27 |

## Important Notice



The contents contained in this document are believed to be accurate at the time of publication. ESMT assumes no responsibility for any error in this document, and reserves the right to change the products or specification in this document without notice.

The information contained herein is presented only as a guide or examples for the application of our products. No responsibility is assumed by ESMT for any infringement of patents, copyrights, or other intellectual property rights of third parties which may result from its use. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of ESMT or others.

Any semiconductor devices may have inherently a certain rate of failure. To minimize risks associated with customer's application, adequate design and operating safeguards against injury, damage, or loss from such failure, should be provided by the customer when making application designs.

ESMT's products are not authorized for use in critical applications such as, but not limited to, life support devices or system, where failure or abnormal operation may directly affect human lives or cause physical injury or property damage. If products described here are to be used for such kinds of application, purchaser must do its own quality assurance testing appropriate to such applications.