# EN25Q80C (2A)
## 8 Megabit Serial Flash Memory with 4Kbyte Uniform Sector

## FEATURES

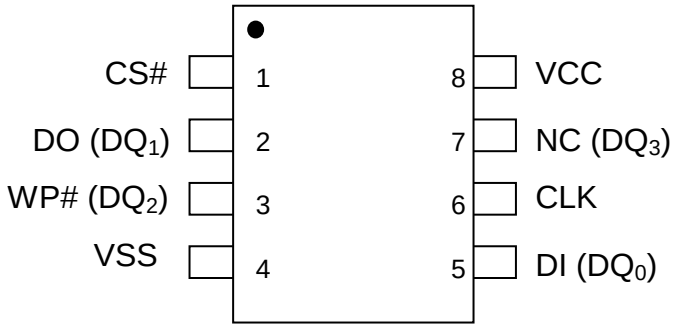
- Single power supply operation
  - Full voltage range: 2.3-3.6 volt
- Serial Interface Architecture
  - SPI Compatible: Mode 0 and Mode 3
- 8 M-bit Serial Flash
  - 8 M-bit/1,024 K-byte/4,096 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO, WP#
  - Dual SPI: CLK, CS#, $DQ_0$, $DQ_1$, WP#
  - Quad SPI: CLK, CS#, $DQ_0$, $DQ_1$, $DQ_2$, $DQ_3$
- High performance
  - 104MHz clock rate for Standard SPI
  - 104MHz clock rate for two data bits
  - 104MHz clock rate for four data bits
- Write Suspend and Write Resume
- Low power consumption
  - 5mA typical active current
  - 1μA 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: 0.5ms typical
  - 4KB Block erase time 40ms typical
  - 32KB Block erase time 120ms typical
  - 64KB Block erase time 150ms typical
  - Chip erase time: 4 seconds typical
- 3 sets of OTP lockable 512, 512, and 512 byte security sectors
- Support Serial Flash Discoverable Parameters (SFDP) signature
- Read Unique ID Number
- Write Suspend/Resume
- Minimum 100K endurance cycle
- Data retention time 20 years
- Package Options
  - 8 pins SOP 150mil body width
  - 8 pins SOP 200mil body width
  - 8 pins VSOP 150mil body width
  - 8 contact USON (2x3x0.55mm)
  - 8 contact USON (2x3x0.45mm)
  - 8 contact VDFN (5x6mm)
  - 8 pins PDIP
  - All Pb-free packages are compliant RoHS, Halogen-Free and REACH.
- Industrial temperature Range
- Volatile Status Register Bits

## GENERAL DESCRIPTION

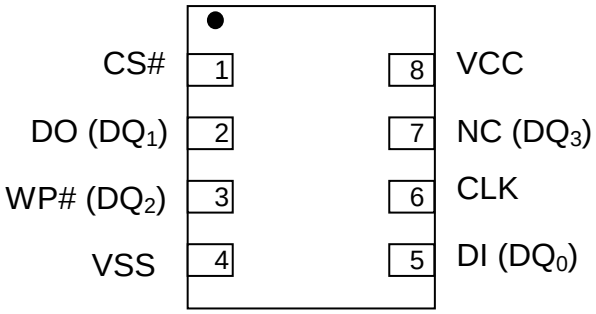
The EN25Q80C (2A) is a 8 Megabit (1024 K-byte) Serial Flash memory, with enhanced write protection mechanisms. The EN25Q80C (2A) supports the standard Serial Peripheral Interface (SPI), and a high performance Dual/Quad output as well as Dual/Quad I/O using SPI pins: Serial Clock, Chip Select, Serial $DQ_0$(DI), $DQ_1$(DO), $DQ_2$(WP#) and $DQ_3$(NC). SPI clock frequencies of up to 104MHz are supported allowing equivalent clock rates of 208MHz (104MHz x 2) for Dual Output and 416MHz (104MHz x 4) for Quad Output when using the Dual/Quad I/O Fast Read instructions. The memory can be programmed 1 to 256 bytes at a time, using the Page Program instruction.

The EN25Q80C (2A) is designed to allow either single Sector/Block at a time or full chip erase operation. The EN25Q80C (2A) 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.

**Figure.1 CONNECTION DIAGRAMS**




8 - LEAD SOP / PDIP



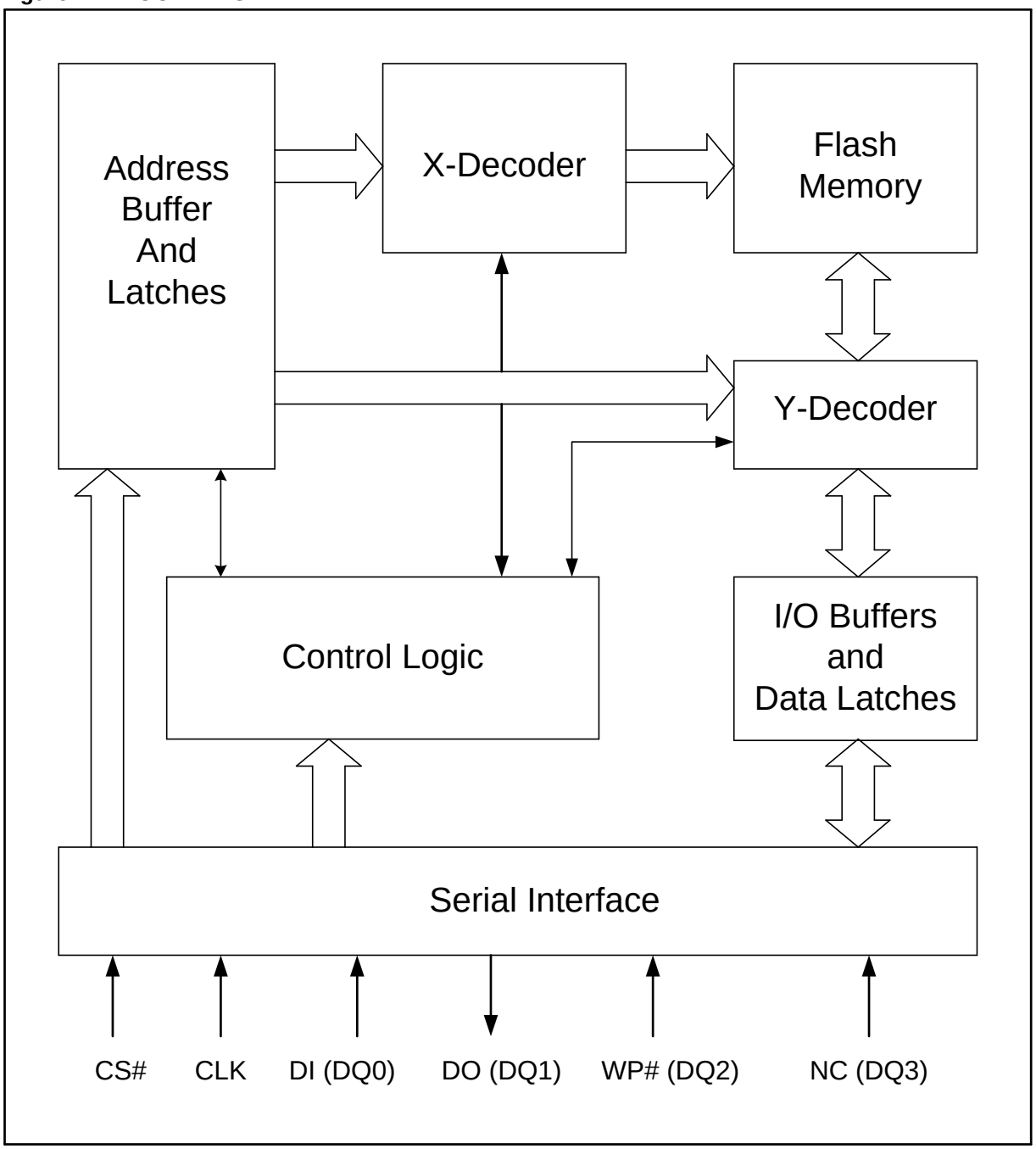

8 - LEAD VDFN

**Table 1. 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 Select |
| WP# ($DQ_2$) | Write Protect (Data Input Output 2) *2 |
| NC ($DQ_3$) | NC pin (Data Input Output 3) *2 |
| Vcc | Supply Voltage (2.3-3.6V) |
| Vss | Ground |
| NC | No Connect |

Note:
1. $DQ_0$ and $DQ_1$ are used for Dual and Quad instructions.
2. $DQ_2$ ~ $DQ_3$ are used for Quad instructions.

**Figure 2. BLOCK DIAGRAM**




**Note:**
1. $DQ_0$ and $DQ_1$ are used for Dual and Quad 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 EN25Q80C (2A) 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.

### Hold (HOLD#)

The HOLD# pin allows the device to be paused while it is actively selected. 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.
For EN25Q80C (2A), there is no HOLD# pin while shipping out from factory. User can enable it by setting bit2(HDEN) in WRSR4.

### Write Protect (WP#)

The Write Protect (WP#) pin enables the lock-down function of the Status Register Protect (SRP) bits in the Status Register. When WP# is driven low, the execution of the Write Status Register (WRSR) and Write Status Register4 (WRSR4) instructions are determined by the value of the SRP bit (see Table 2). When WP# is high, the lock-down function of the SRP bit is disabled.

**Table 2: Conditions to Execute Write-Status- Register (WRSR/ WRSR4) Instruction**

| WP# | SRP | Execute WRSR/ WRSR4 Instruction |
|---|---|---|
| L | 1 | Not Allowed |
| L | 0 | Allowed |
| H | X | Allowed |

## MEMORY ORGANIZATION

The memory is organized as:

- 1,048,576 bytes
- Uniform Sector Architecture
  - 16 blocks of 64-Kbyte
  - 32 blocks 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.

**Table 3. Uniform Block Sector Architecture**

| 64KB Block | 32KB Block | Sector | Address range | |
|---|---|---|---|---|
| 15 | 31 | 255 | 0FF000h | 0FFFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 248 | 0F8000h | 0F8FFFh |
| | 30 | 247 | 0F7000h | 0F7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 240 | 0F0000h | 0F0FFFh |
| 14 | 29 | 239 | 0EF000h | 0EFFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 232 | 0E8000h | 0E8FFFh |
| | 28 | 231 | 0E7000h | 0E7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 224 | 0E0000h | 0E0FFFh |
| 13 | 27 | 223 | 0DF000h | 0DFFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 216 | 0D8000h | 0D8FFFh |
| | 26 | 215 | 0D7000h | 0D7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 208 | 0D0000h | 0D0FFFh |
| 12 | 25 | 207 | 0CF000h | 0CFFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 200 | 0C8000h | 0C8FFFh |
| | 24 | 199 | 0C7000h | 0C7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 192 | 0C0000h | 0C0FFFh |
| 11 | 23 | 191 | 0BF000h | 0BFFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 184 | 0B8000h | 0B8FFFh |
| | 22 | 183 | 0B7000h | 0B7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 176 | 0B0000h | 0B0FFFh |
| 10 | 21 | 175 | 0AF000h | 0AFFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 168 | 0A8000h | 0A8FFFh |
| | 20 | 167 | 0A7000h | 0A7FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 160 | 0A0000h | 0A0FFFh |
| 9 | 19 | 159 | 09F000h | 09FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 152 | 098000h | 098FFFh |
| | 18 | 151 | 097000h | 097FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 144 | 090000h | 090FFFh |
| 8 | 17 | 143 | 08F000h | 08FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 136 | 088000h | 088FFFh |
| | 16 | 135 | 087000h | 087FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 128 | 080000h | 080FFFh |
| 7 | 15 | 127 | 07F000h | 07FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 120 | 078000h | 078FFFh |
| | 14 | 119 | 077000h | 077FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 112 | 070000h | 070FFFh |

**Table 3. Uniform Block Sector Architecture - Continued**

| 64KB Block | 32KB Block | Sector | Address range | |
|---|---|---|---|---|
| 6 | 13 | 111 | 06F000h | 06FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 104 | 068000h | 068FFFh |
| | 12 | 103 | 067000h | 067FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 96 | 060000h | 060FFFh |
| 5 | 11 | 95 | 05F000h | 05FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 88 | 058000h | 058FFFh |
| | 10 | 87 | 057000h | 057FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 80 | 050000h | 050FFFh |
| 4 | 9 | 79 | 04F000h | 04FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 72 | 048000h | 048FFFh |
| | 8 | 71 | 047000h | 047FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 64 | 040000h | 040FFFh |
| 3 | 7 | 63 | 03F000h | 03FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 56 | 038000h | 038FFFh |
| | 6 | 55 | 037000h | 037FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 48 | 030000h | 030FFFh |
| 2 | 5 | 47 | 02F000h | 02FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 40 | 028000h | 028FFFh |
| | 4 | 39 | 027000h | 027FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 32 | 020000h | 020FFFh |
| 1 | 3 | 31 | 01F000h | 01FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 24 | 018000h | 018FFFh |
| | 2 | 23 | 017000h | 017FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 16 | 010000h | 010FFFh |
| 0 | 1 | 15 | 00F000h | 00FFFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 8 | 008000h | 008FFFh |
| | 0 | 7 | 007000h | 007FFFh |
| | | ⋮ | ⋮ | ⋮ |
| | | 0 | 000000h | 000FFFh |

## OPERATING FEATURES

### Standard SPI Modes

The EN25Q80C (2A) is accessed through a 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 Figure 3, 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.

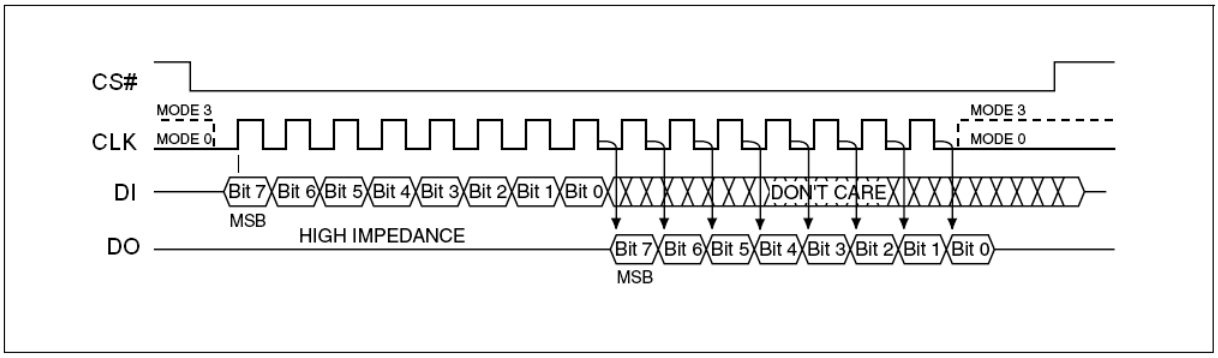



**Figure 3. SPI Modes**

### Dual SPI Instruction

The EN25Q80C (2A) 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 EN25Q80C (2A) supports Quad input / 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. When using Quad SPI instruction the DI and DO pins become bidirectional I/O pins; $DQ_0$ and $DQ_1$, and the WP# and NC pins become $DQ_2$ and $DQ_3$ respectively.

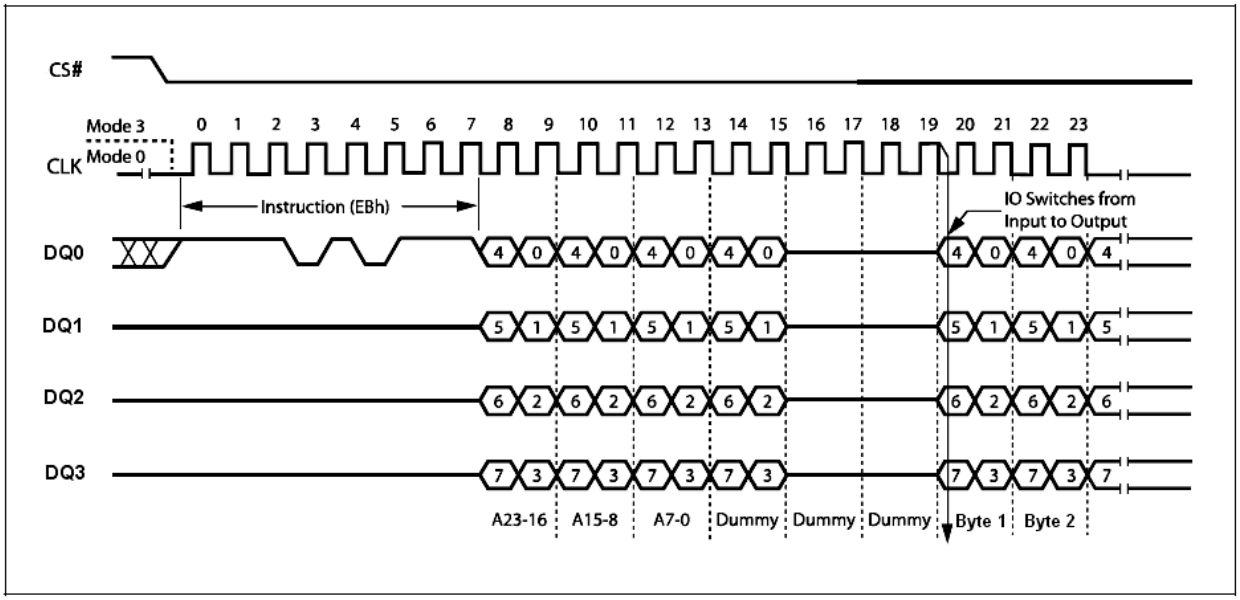



**Figure 4. Quad I/O SPI Modes**

### Full Quad SPI Modes (QPI)

The EN25Q80C (2A) also supports Full Quad SPI Mode (QPI) function while using the Enable Quad Peripheral Interface mode (EQPI) (38h). When using Quad SPI instruction the DI and DO pins become bidirectional I/O pins; $DQ_0$ and $DQ_1$, and the WP# and HOLD# pins become $DQ_2$ and $DQ_3$ respectively.

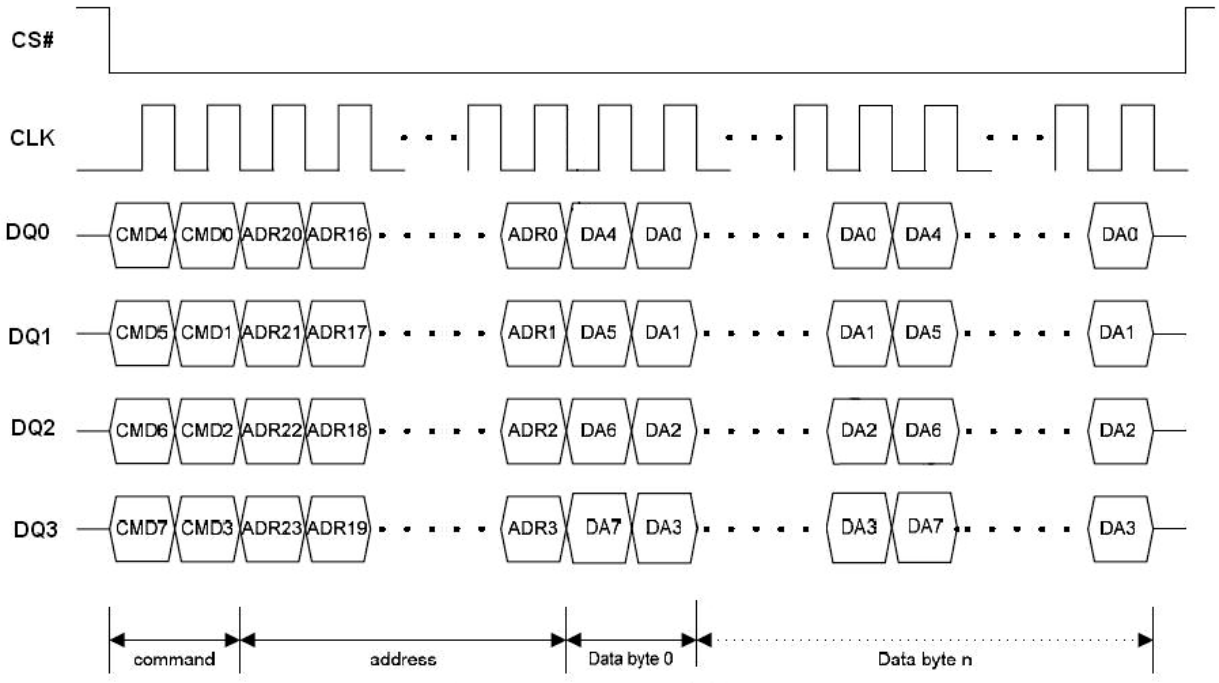



**Figure 5. Full Quad SPI Modes**

**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), 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, and 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

EN25Q80C (2A) 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 or Page Program (PP), 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 (4KBL, TB, BP2, BP1, BP0) bits, Status Register Protect (SRP) bit , HDEN, WPDIS, CMP 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).

**Table 4. Protected Area Sizes Sector Organization**

| Status Register Content | | | | | | Memory Content | | | |
|---|---|---|---|---|---|---|---|---|---|
| CMP | 4KBL | TB | BP2 | BP1 | BP0 | Protect Areas | Address | Density(KB) | portion |
| 0 | 0 | 0 | 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 and 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 | 0 | 1 | 0 | 1 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 0 | 0 | 1 | 1 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 0 | 0 | 1 | 1 | 1 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 0 | 1 | 0 | 0 | 0 | None | None | None | None |
| 0 | 0 | 1 | 0 | 0 | 1 | Block 0 | 000000h - 00FFFFh | 64KB | Lower 1/16 |
| 0 | 0 | 1 | 0 | 1 | 0 | Block 0 and 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 | 1 | 1 | 0 | 1 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 0 | 1 | 1 | 1 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 0 | 1 | 1 | 1 | 1 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 1 | 0 | 0 | 0 | 0 | None | None | None | None |
| 0 | 1 | 0 | 0 | 0 | 1 | Sector 255 | 0FF000h - 0FFFFFh | 4KB | Upper 1/256 |
| 0 | 1 | 0 | 0 | 1 | 0 | Sector 254 to 255 | 0FE000h - 0FFFFFh | 8KB | Upper 1/128 |
| 0 | 1 | 0 | 0 | 1 | 1 | Sector 252 to 255 | 0FC000h - 0FFFFFh | 16KB | Upper 1/64 |
| 0 | 1 | 0 | 1 | 0 | 0 | Sector 248 to 255 | 0F8000h - 0FFFFFh | 32KB | Upper 1/32 |
| 0 | 1 | 0 | 1 | 0 | 1 | Sector 248 to 255 | 0F8000h - 0FFFFFh | 32KB | Upper1/32 |
| 0 | 1 | 0 | 1 | 1 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 1 | 0 | 1 | 1 | 1 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 1 | 1 | 0 | 0 | 0 | None | None | None | None |
| 0 | 1 | 1 | 0 | 0 | 1 | Sector 0 | 000000h - 000FFFh | 4KB | Lower 1/256 |
| 0 | 1 | 1 | 0 | 1 | 0 | Sector 0 to 1 | 000000h - 001FFFh | 8KB | Lower 1/128 |
| 0 | 1 | 1 | 0 | 1 | 1 | Sector 0 to 3 | 000000h - 003FFFh | 16KB | Lower 1/64 |
| 0 | 1 | 1 | 1 | 0 | 0 | Sector 0 to 7 | 000000h- 007FFFh | 32KB | Lower 1/32 |
| 0 | 1 | 1 | 1 | 0 | 1 | Sector 0 to 7 | 000000h- 007FFFh | 32KB | Lower1/32 |
| 0 | 1 | 1 | 1 | 1 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |
| 0 | 1 | 1 | 1 | 1 | 1 | All | 000000h - 0FFFFFh | 1024KB | all |
| 1 | 0 | 0 | 0 | 0 | 0 | All | 000000h - 0FFFFFh | 1024KB | 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 | 0 | 1 | 0 | 1 | None | None | None | None |
| 1 | 0 | 0 | 1 | 1 | 0 | None | None | None | None |
| 1 | 0 | 0 | 1 | 1 | 1 | None | None | None | None |
| 1 | 0 | 1 | 0 | 0 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |
| 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 | 1 | 1 | 0 | 1 | None | None | None | None |
| 1 | 0 | 1 | 1 | 1 | 0 | None | None | None | None |
| 1 | 0 | 1 | 1 | 1 | 1 | None | None | None | None |
| 1 | 1 | 0 | 0 | 0 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |

| Status Register Content | | | | | | Memory Content | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | Sector 0 to 254 | 000000h - 0FEFFFh | 1020KB | Lower 255/256 |
| 1 | 1 | 0 | 0 | 1 | 0 | Sector 0 to 253 | 000000h - 0FDFFFh | 1016KB | Lower 127/128 |
| 1 | 1 | 0 | 0 | 1 | 1 | Sector 0 to 251 | 000000h - 0FBFFFh | 1008KB | Lower 63/64 |
| 1 | 1 | 0 | 1 | 0 | 0 | Sector 0 to 247 | 000000h - 0F7FFFh | 992KB | Lower 31/32 |
| 1 | 1 | 0 | 1 | 0 | 1 | Sector 0 to 247 | 000000h - 0F7FFFh | 992KB | Lower 31/32 |
| 1 | 1 | 0 | 1 | 1 | 0 | None | None | None | None |
| 1 | 1 | 0 | 1 | 1 | 1 | None | None | None | None |
| 1 | 1 | 1 | 0 | 0 | 0 | All | 000000h - 0FFFFFh | 1024KB | all |
| 1 | 1 | 1 | 0 | 0 | 1 | Sector 1 to 255 | 001000h - 0FFFFFh | 1020KB | Upper 255/256 |
| 1 | 1 | 1 | 0 | 1 | 0 | Sector 2 to 255 | 002000h - 0FFFFFh | 1016KB | Upper 127/128 |
| 1 | 1 | 1 | 0 | 1 | 1 | Sector 4 to 255 | 004000h - 0FFFFFh | 1008KB | Upper 63/64 |
| 1 | 1 | 1 | 1 | 0 | 0 | Sector 8 to 255 | 008000h - 0FFFFFh | 992KB | Upper 31/32 |
| 1 | 1 | 1 | 1 | 0 | 1 | Sector 8 to 255 | 008000h - 0FFFFFh | 992KB | Upper 31/32 |
| 1 | 1 | 1 | 1 | 1 | 0 | None | None | None | None |
| 1 | 1 | 1 | 1 | 1 | 1 | None | None | None | None |

**Enable Boot Lock**

The Enable Boot Lock feature enables user to lock the 64KB block/sector on the top/bottom of the device for protection.

The bits' definitions are described in the following table.

**Table 5. The Enable Boot Lock feature**

| Register Bit | Type | Description | Function |
|---|---|---|---|
| **Normal Mode** | | | |
| S6 | non-volatile / volatile | 4KBL bit | 0 : 64KB-Block (default)<br>1 : Sector |
| S5 | non-volatile / volatile | TB(top/bottom) bit | 0 : Top (default)<br>1 : Bottom |
| **OTP Mode** | | | |
| S3 | OTP / volatile bit | EBL(Enable Boot Lock) bit | 0 (default)<br>1 : 64KB-block/Sector lock selected |

## 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 Table 6. Every instruction sequence starts with a one-byte instruction code. Depending on the instruction, this might be followed by address bytes, or by data bytes, or by 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 4 (RDSR4) 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 Page Program (PP), Quad Input Page Program (QPP) , Sector Erase (SE), Half Block Erase (HBE), Block Erase (BE), Chip Erase (CE), Write Status Register (WRSR) , Write Status Register4 (WRSR4), Volatile Status Register Write Enable, Write Enable (WREN), Write Disable (WRDI) or Deep Power-down (DP) 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, HBE and 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.

**Table 6A. Instruction Set**

| Instruction Name | Byte 1 Code | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | n-Bytes |
|---|---|---|---|---|---|---|---|
| **EQPI** | **38h** | | | | | | |
| **RSTQIO(1) Release Quad I/O or Fast Read Enhanced Mode** | **FFh** | | | | | | |
| **RSTEN** | **66h** | | | | | | |
| **RST(2)** | **99h** | | | | | | |
| **Write Enable(WREN)** | **06h** | | | | | | |
| **Volatile Status Register Write Enable (3)** | **50h** | | | | | | |
| **Write Disable(WRDI) / Exit OTP mode** | **04h** | | | | | | |
| **Read Status Register (RDSR)** | **05h** | (SR7-SR0) (4) | | | | | continuous(5) |
| **Read Status Register 2 (RDSR2)** | **09h** | (SR2.7-SR2.0)(4) | | | | | continuous(5) |
| **Read Status Register 4 (RDSR4)** | **85h** | (SR4.7-SR4.0)(4) | | | | | continuous(5) |
| **Write Status Register (WRSR)** | **01h** | SR7-SR0 | | | | | |
| **Write Status Register 4 (WRSR4)** | **C1h** | SR4.7-SR4.0 | | | | | |
| **Write Suspend** | **B0h** | | | | | | |
| **Write Resume** | **30h** | | | | | | |
| **Deep Power-down** | **B9h** | | | | | | |
| **Release from Deep Power-down, and read Device ID(RDS)** | **ABh** | dummy | dummy | dummy | (ID7-ID0) | | (6) |
| **Release from Deep Power-down(RDP)** | | | | | | | |
| **Manufacturer/ Device ID** | **90h** | dummy | dummy | 00h<br>01h | (M7-M0)<br>(ID7-ID0) | (ID7-ID0)<br>(M7-M0) | (7) |
| **Read Identification** | **9Fh** | (M7-M0) | (ID15-ID8) | (ID7-ID0) | (8) | | |
| **Enter OTP mode** | **3Ah** | | | | | | |
| **Read SFDP mode and Unique ID Number** | **5Ah** | A23-A16 | A15-A8 | A7-A0 | dummy | (D7-D0) | (Next Byte) continuous |

**Notes:**

1. Device accepts eight-clocks command in Standard SPI mode, or two-clocks command in Quad SPI mode
2. RST command only executed if RSTEN command is executed first. Any intervening command will disable Reset.
3. Volatile Status Register Write Enable command must precede WRSR command without any intervening commands to write data to Volatile Status Register
4. 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
5. The Status Register contents will repeat continuously until CS# terminate the instruction
6. The Device ID will repeat continuously until CS# terminates the instruction
7. 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
8. (M7-M0) : Manufacturer, (ID15-ID8) : Memory Type, (ID7-ID0) : Memory Capacity

**Table 6B. 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) |

**Table 6C. 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) |

**Table 6D. Instruction Set (Erase Instruction)**

| Instruction Name | OP Code | Address bits | Dummy bits / Clocks (Default) | Data Out | Remark |
|---|---|---|---|---|---|
| Sector Erase 4K (SE) | 20h | 24 bits | 0 | (D7-D0, …) | |
| 32K Half Block Erase (HBE) | 52h | 24 bits | 0 | (D7-D0, …) | |
| 64K Block Erase (BE) | D8h | 24 bits | 0 | (D7-D0, …) | |
| Chip Erase (CE) | C7h/ 60h | 24 bits | 0 | (D7-D0, …) | |

**Table 6E. Instruction Set (Read Instruction support mode and dummy cycle setting)**

| Instruction Name | OP Code | Start From SPI/QPI (1) | | Dummy Cycle | |
|---|---|---|---|---|---|
| | | SPI | QPI | Start From SPI | Start From QPI |
| Read Data | 03h | Yes | No | N/A | N/A |
| Fast Read | 0Bh | Yes | Yes | 8 clocks | 6 clocks |
| Dual Output Fast Read | 3Bh | Yes | No | 8 clocks | N/A |
| Dual I/O Fast Read | BBh | Yes | No | 4 clocks | N/A |
| Quad Output Fast Read | 6Bh | Yes | No | 8 clocks | N/A |
| Quad I/O Fast Read | EBh | Yes | Yes | 6 clocks | 6 clocks |

**Note:**
1. 'Start From SPI/QPI' means if this command is initiated from SPI or QPI mode.
.

**Table 7. Manufacturer and Device Identification**

| OP Code | (M7-M0) | (ID15-ID0) | (ID7-ID0) |
|---|---|---|---|
| ABh | | | 13h |
| 90h | 1Ch | | 13h |
| 9Fh | 1Ch | 3014h | |

**Enable Quad Peripheral Interface mode (EQPI) (38h)**

The Enable Quad Peripheral Interface mode (EQPI) instruction will enable the flash device for Quad SPI bus operation. Upon completion of the instruction, all instructions thereafter will be 4-bit multiplexed input/output until a power cycle or " Reset Quad I/O instruction " instruction, as shown in Figure 6. The device did not support the Read Data Bytes (READ) (03h)**,** Dual Output Fast Read (3Bh) and Dual Input/Output FAST_READ (BBh), Quad Output Fast Read(6Bh) and Quad Input Page Program (32h) modes while the Enable Quad Peripheral Interface mode (EQPI) (38h) turns on.

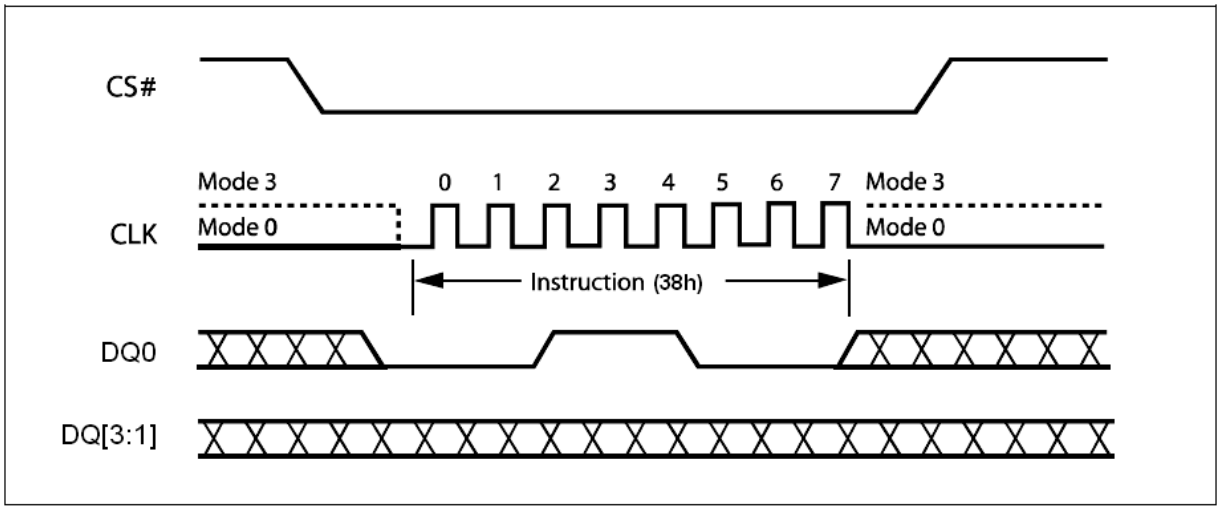



**Figure 6. Enable Quad Peripheral Interface mode Sequence Diagram**

**Reset Quad I/O (RSTQIO) or Release Quad I/O Fast Read Enhancement Mode (FFh)**

The Reset Quad I/O instruction resets the device to 1-bit Standard SPI operation. To execute a Reset Quad I/O operation, the host drives CS# low, sends the Reset Quad I/O command cycle (FFh) then, drives CS# high. This command can't be used in Standard SPI mode.

User also can use the 0xFFh command to release the Quad I/O Fast Read Enhancement Mode. The detail description, please see the Quad I/O Fast Read Enhancement Mode section.

**Note:**

If the system is in the Quad I/O Fast Read Enhance Mode in QPI Mode, it is necessary to execute 0xFFh command by two times. The first 0xFFh command is to release Quad I/O Fast Read Enhance Mode, and the second 0xFFh command is to release QPI Mode.

**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. This operation consists of two commands: Reset-Enable (RSTEN) and Reset (RST).

To reset the EN25Q80C (2A) 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 to data = 00h, see Figure 7 for SPI Mode and Figure 7.1 for QPI 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.

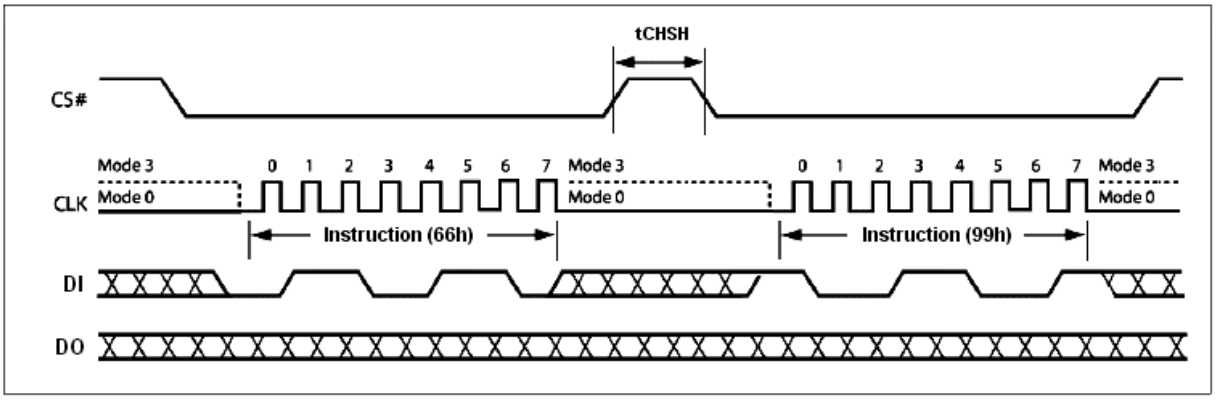



**Figure 7. Reset-Enable and Reset Sequence Diagram**

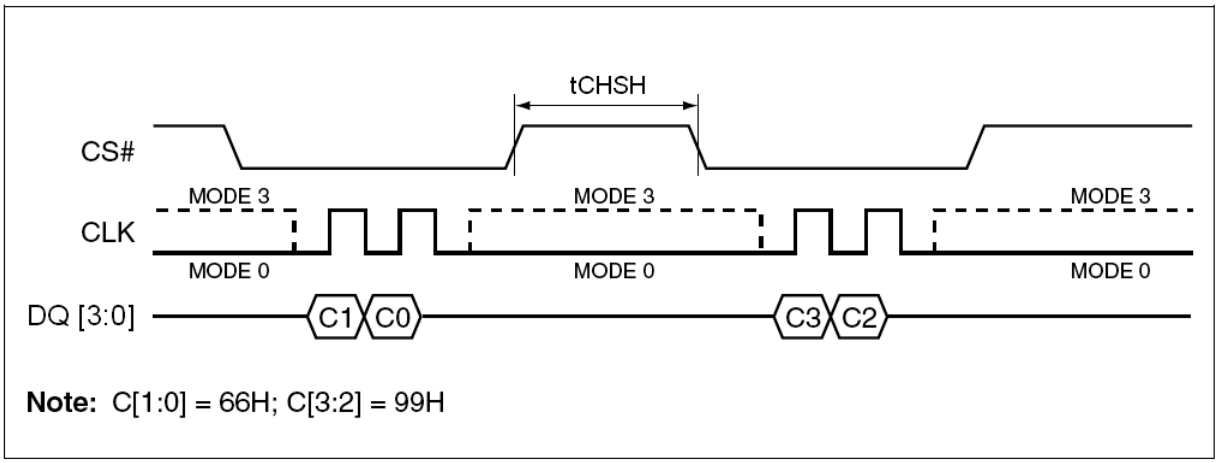
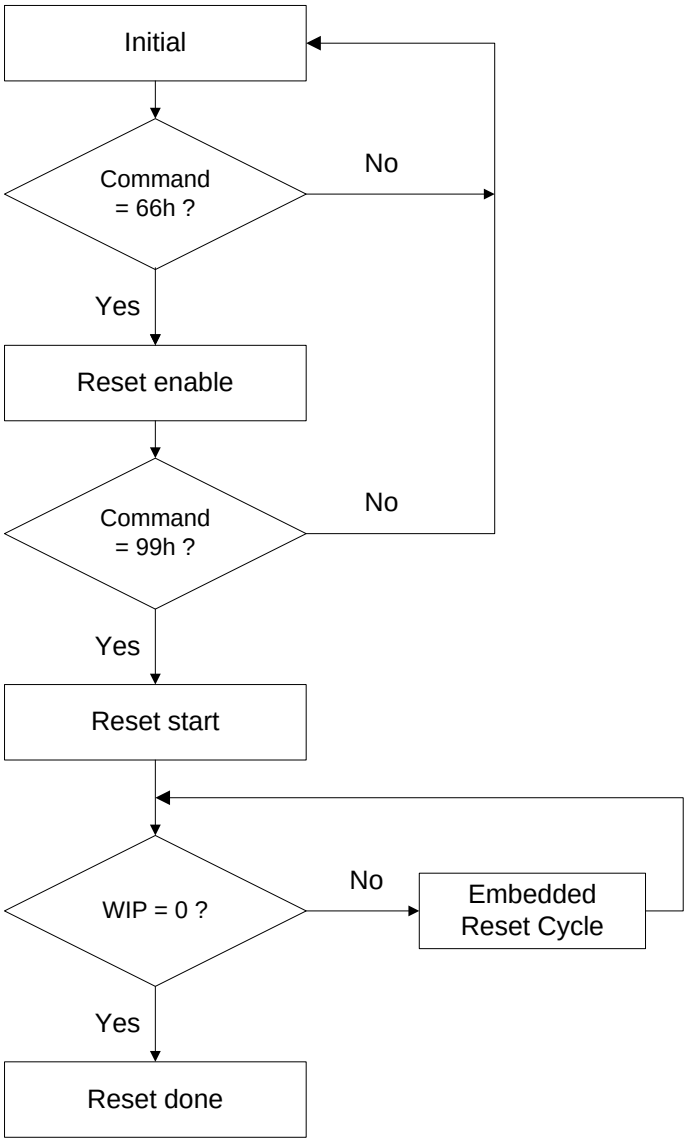



**Figure 7.1 Reset-Enable and Reset Sequence Diagram in QPI Mode**

**Software Reset Flow**




**Note:**

1. Reset-Enable (RSTEN) (66h) and Reset (RST) (99h) commands need to match standard SPI or QPI (Full Quad) mode.
2. Continue (Enhance) EB mode need to use quad Reset-Enable (RSTEN) (66h) and quad Reset (RST) (99h) commands.
3. If user is not sure it is in SPI or Quad mode, we suggest to execute sequence as follows:
   Quad Reset-Enable (RSTEN) (66h) -> Quad Reset (RST) (99h) -> SPI Reset-Enable (RSTEN) (66h)
   -> SPI Reset (RST) (99h) to reset.
4. The reset command could be executed during embedded program and erase process, QPI mode and Continue EB mode and suspend mode to back to SPI mode.
5. This flow cannot release the device from Deep power down mode.
6. The Status Register Bit and Status Register 2 Bit will reset to default value after reset done.
7. If user reset device during erase, the embedded reset cycle software reset latency will take about 28us in worst case.
8. User can't do software reset command while doing 4K/32K erase operation.

**Write Enable (WREN) (06h)**

The Write Enable (WREN) instruction (Figure 8) 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), Half Block Erase (HBE), Block Erase (BE), Chip Erase (CE) , Write Status Register (WRSR) and Write Status Register4 (WRSR4) 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.

The instruction sequence is shown in Figure 10.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

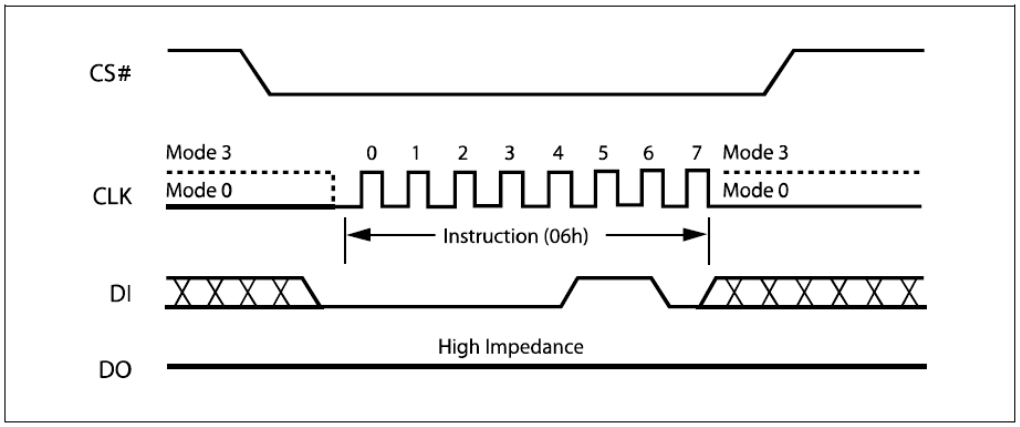



**Figure 8. Write Enable Instruction Sequence Diagram**

**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' (01h) 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) and WRSR4 (C1h). The Status Register bits will be refresh to Volatile Status Register (SR[7:2]) or Volatile Status Register4 (SR[7:1]) within tSHSL2 (50ns). Upon power off or the execution of a Software/Hardware Reset, the volatile Status Register/ Volatile Status Register4 bit values will be lost, and the non-volatile Status Register/ non-volatile Status Register4 bit values will be restored. The instruction sequence is shown in Figure 9.

The instruction sequence is shown in Figure 10.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

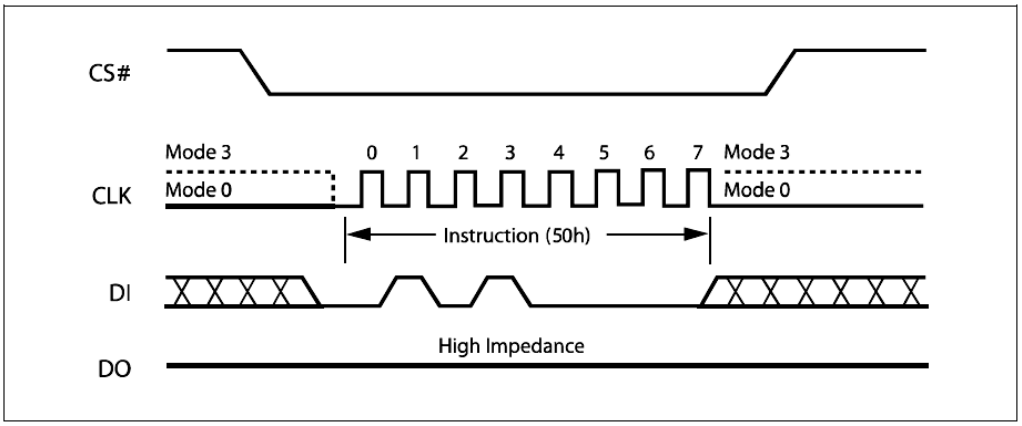



**Figure 9. Volatile Status Register Write Enable Instruction Sequence Diagram**

**Write Disable (WRDI) (04h)**

The Write Disable instruction (Figure 10) resets the Write Enable Latch (WEL) bit in the Status Register to a 0 or exit from OTP mode to normal mode. 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, Half Block Erase (HBE), Block Erase (BE) and Chip Erase instructions.

The instruction sequence is shown in Figure 10.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

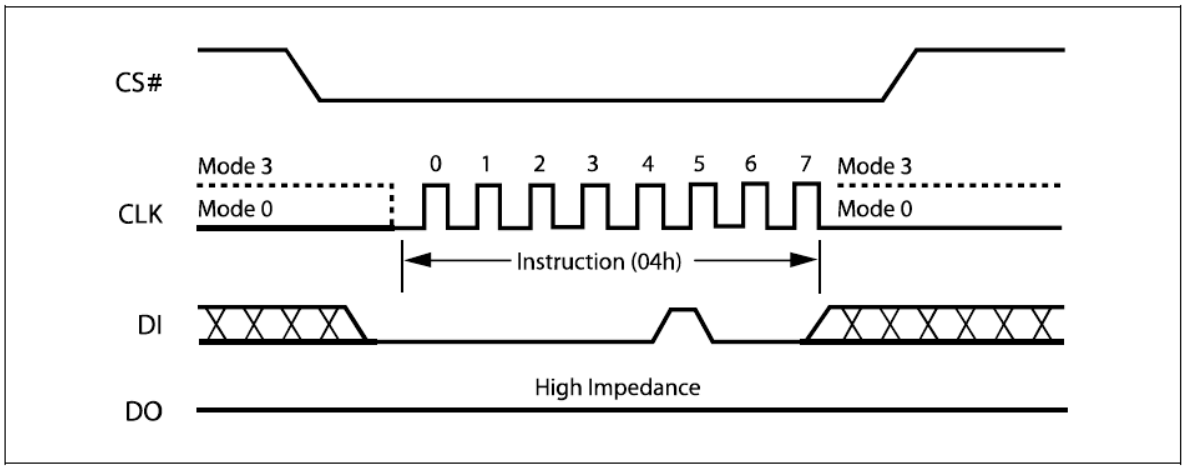



**Figure 10. Write Disable Instruction Sequence Diagram**

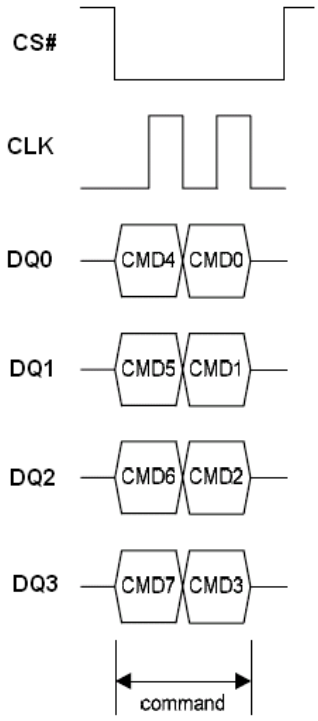



**Figure 10.1 Write Enable/Disable Instruction Sequence in QPI Mode**

**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 Figure 11.

The instruction sequence is shown in Figure 11.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

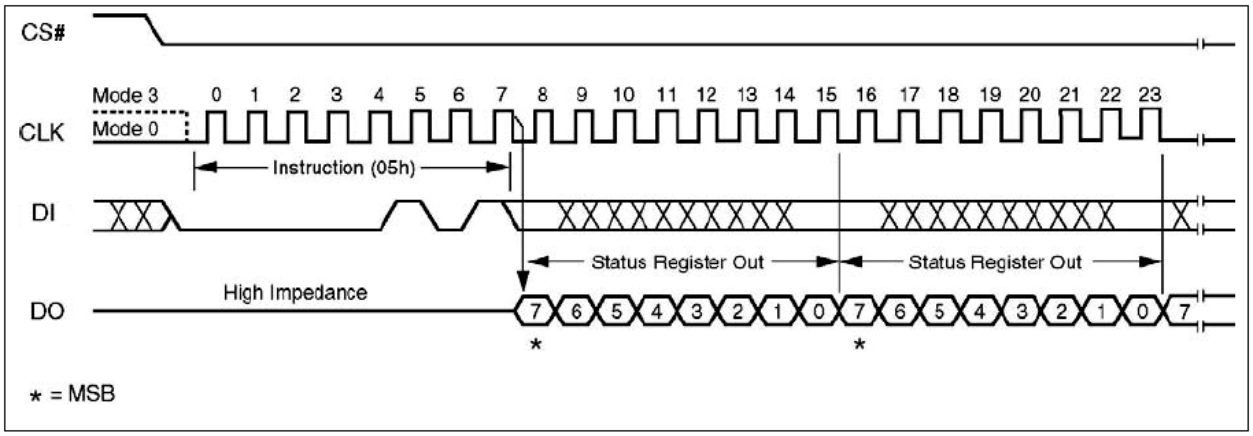



**Figure 11. Read Status Register Instruction Sequence Diagram**

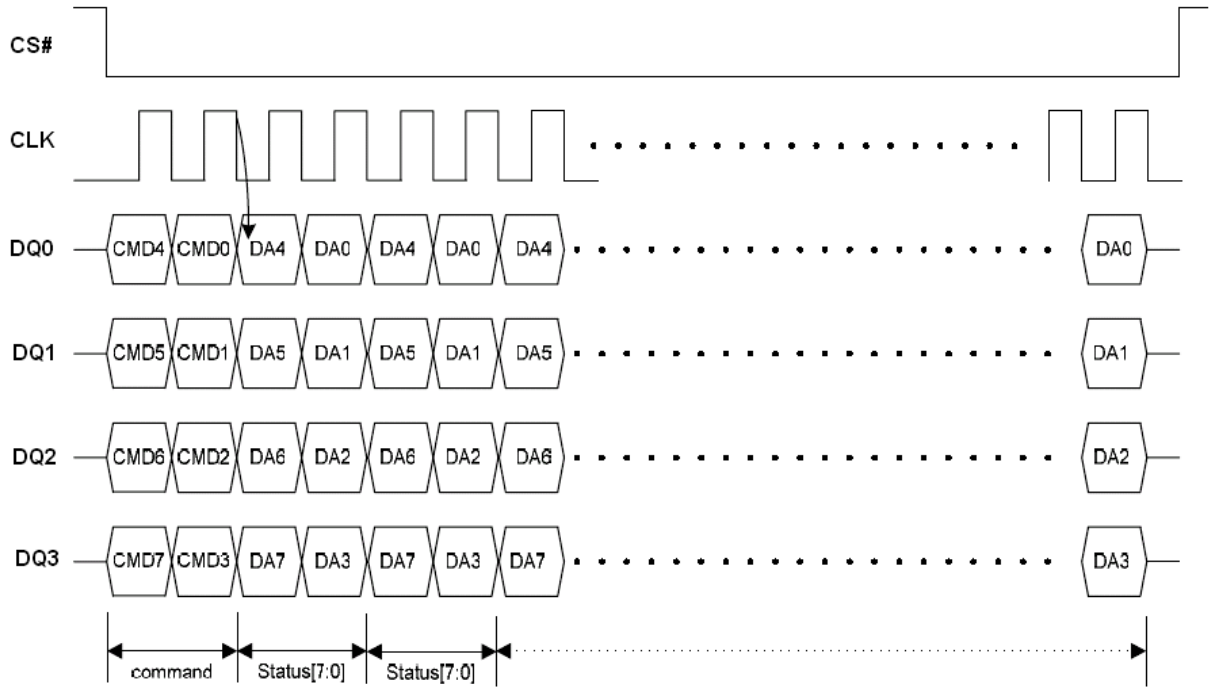



**Figure 11.1 Read Status Register Instruction Sequence in QPI Mode**

**Table 8. Status Register Bit Locations**

| SR7 | SR6 | SR5 | SR4 | SR3 | SR2 | SR1 | SR0 |
|---|---|---|---|---|---|---|---|
| SRP bit | 4KBL | TB | BP2 bit | BP1 bit | BP0 bit | WEL bit | WIP bit |
| SPL0 bit | reversed | reversed | reversed | EBL | SPL1 bit | SPL2 bit | WIP bit |

**Table 8.1 Status Register Bit Locations (In Normal mode)**

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

**Table 8.2 Status Register Bit Locations (In OTP mode)**

| SR7 | SR6 | SR5 | SR4 | SR3 | SR2 | SR1 | SR0 |
|---|---|---|---|---|---|---|---|
| SPL0 bit | Reserved bit | Reserved bit | Reserved bit | **EBL**<br>(Enable Boot Lock) | SPL1 bit | SPL2 bit | WIP bit<br>(Write In Progress bit) |
| 1 = security sector 0 is protected | | | | 1 = lock selected 64KB-Block/Sector | 1 = security sector 1 is protected | 1 = security sector 2 is protected | 1 = write operation<br>0 = not in write operation |
| OTP bit / Volatile bit | | | | OTP bit / Volatile bit | OTP bit / Volatile bit | OTP bit / Volatile bit | Read only bit |

**Note :**
1. In OTP mode, SR7 bit is served as SPL0 bit; SR3 bit is served as EBL bit; SR2 bit is served as SPL1 bit ; SR1 bit is served as SPL2 bit and SR0 bit is served as WIP bit.
2. See the table 4 “Protected Area Sizes Sector Organization”.

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 Table 4.) 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 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. It also controls if the Top (TB=0) or the Bottom (TB=1) 64KB-block/sector is protected when Boot Lock feature is enabled. 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. It is used to set the protection area size as block (64KB) or sector (4KB) while EBL bit is set to 1. 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 non-volatile bits of the Status Register (SRP, 4KBL, TB, BP2, BP1, BP0), Status Register4(HDEN, WHDIS, CMP) become read-only bits and the Write Status Register (WRSR)/Write Status Register4(WRSR4) instruction is no longer accepted for execution.

In OTP mode, SR7~SR0 bits are served as SPL0 bit, reserved bit, reserved bit, reserved bit and EBL bit, SPL1 bit, SPL2 and WIP bit.

**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.

**EBL bit.** The Enable Boot Lock (EBL) bit is used to enable the Boot Lock feature. When this bit is programmed to '1', the sector/block selected by the TB bit and 4KBL bit will be locked.

**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.

**Read Status Register 2 (RDSR 2) (09h)**

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. 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 2 continuously, as shown in Figure 12.

The instruction sequence is shown in Figure 12.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

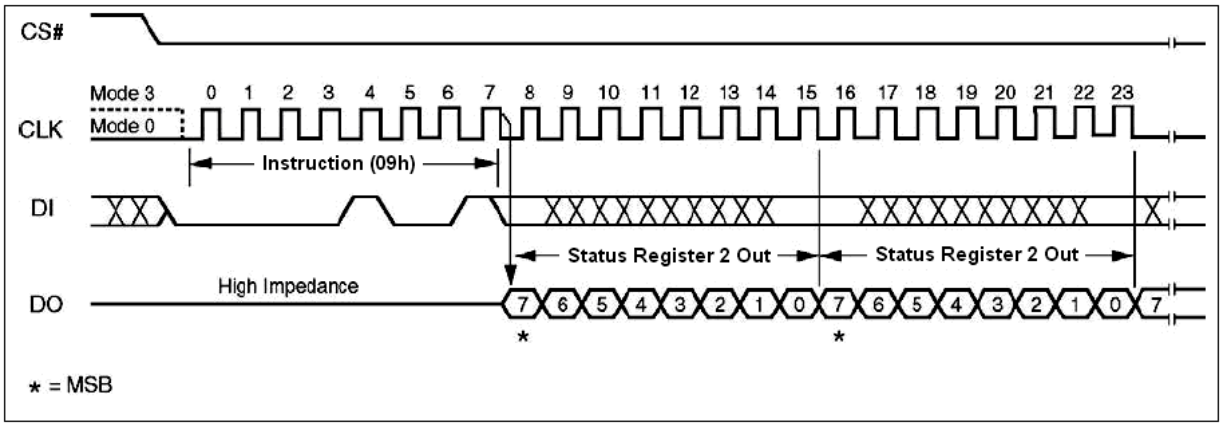



**Figure 12. Read Status Register 2 Instruction Sequence Diagram**

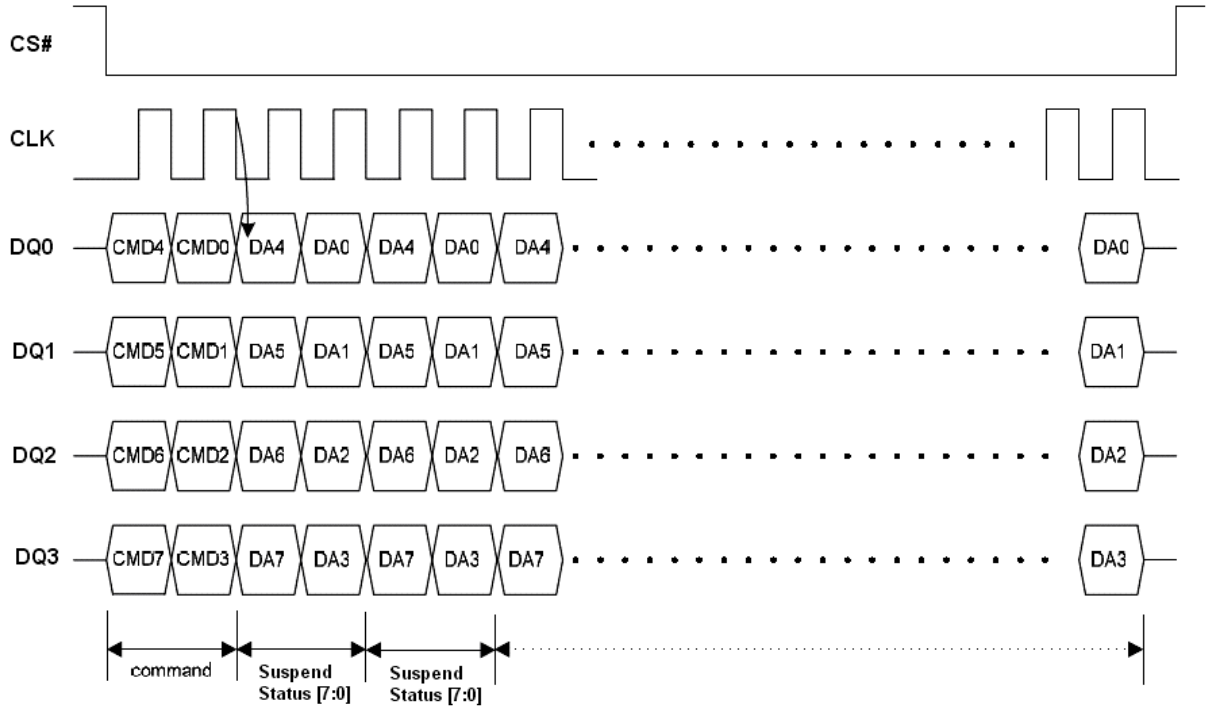



**Figure 12.1 Read Status Register 2 Instruction Sequence in QPI Mode**

**Table 9. Status Register 2 Bit Locations**

| SR2.7 | SR2.6 | SR2.5 | SR2.4 | SR2.3 | SR2.2 | SR2.1 | SR2.0 |
|---|---|---|---|---|---|---|---|
| Reserved bit | Reserved bit | Reserved bit | Reserved bit | WSP (Write Suspend Program bits) | WSE (Write Suspend Erase status bit) | Reserved bit | WIP (Write In Progress bit) (Note 1) |
| | | | | 1 = Program suspended 0 = Program is not suspended | 1 = Erase suspended 0 = Erase is not suspended | | 1 = write operation 0 = not in write operation |
| | | | | volatile bit | volatile bit | | volatile bit |
| | | | | Read Only | Read Only | | Read Only |

**Note:**

1. The default of each volatile bit is “0” at Power-up or after reset.
2. When executed the (RDSR 2) (09h) command, the WIP (SR2.0) value is the same as WIP (SR0) in table 8.

The status and control bits of the Status Register 2 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.

**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”.

**Reserved bit.** In Status Register 2, SR2.1, SR2.4 and SR2.7 are reserved for future use. Current devices will read 0 for these bit locations. It is recommended to mask out the reserved bit when testing the Status Register 2. Doing this will ensure compatibility with future devices.

**Read Status Register 4 (RDSR 4) (85h)**

The Read Status Register 4 (RDSR4) instruction allows the Status Register 4 to be read. The Status Register 4 may be read at any time, even while a Write Suspend or Write Resume cycle is in progress. It is possible to read the Read Status Register 4 continuously, as shown in Figure 13.

The instruction sequence is shown in Figure 13.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

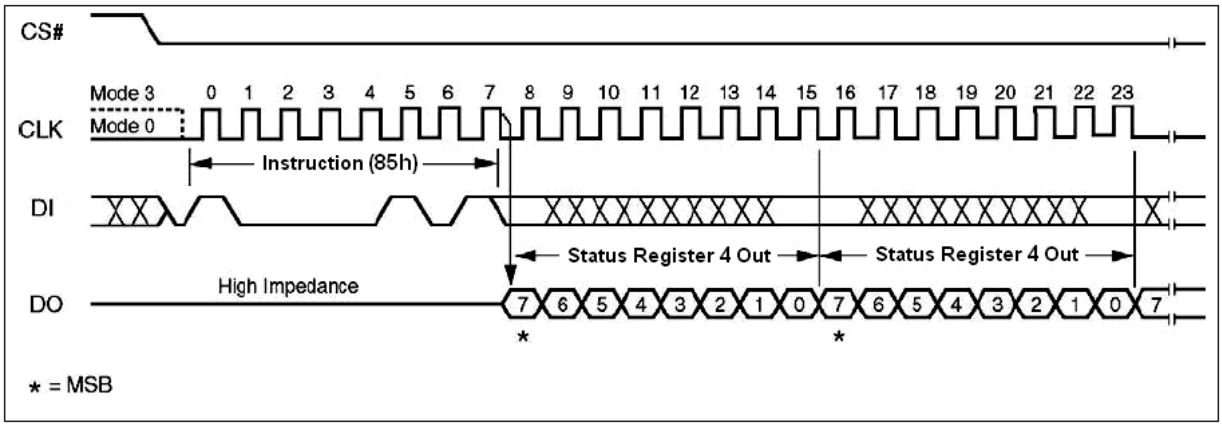



**Figure 13. Read Status Register 4 Instruction Sequence Diagram**

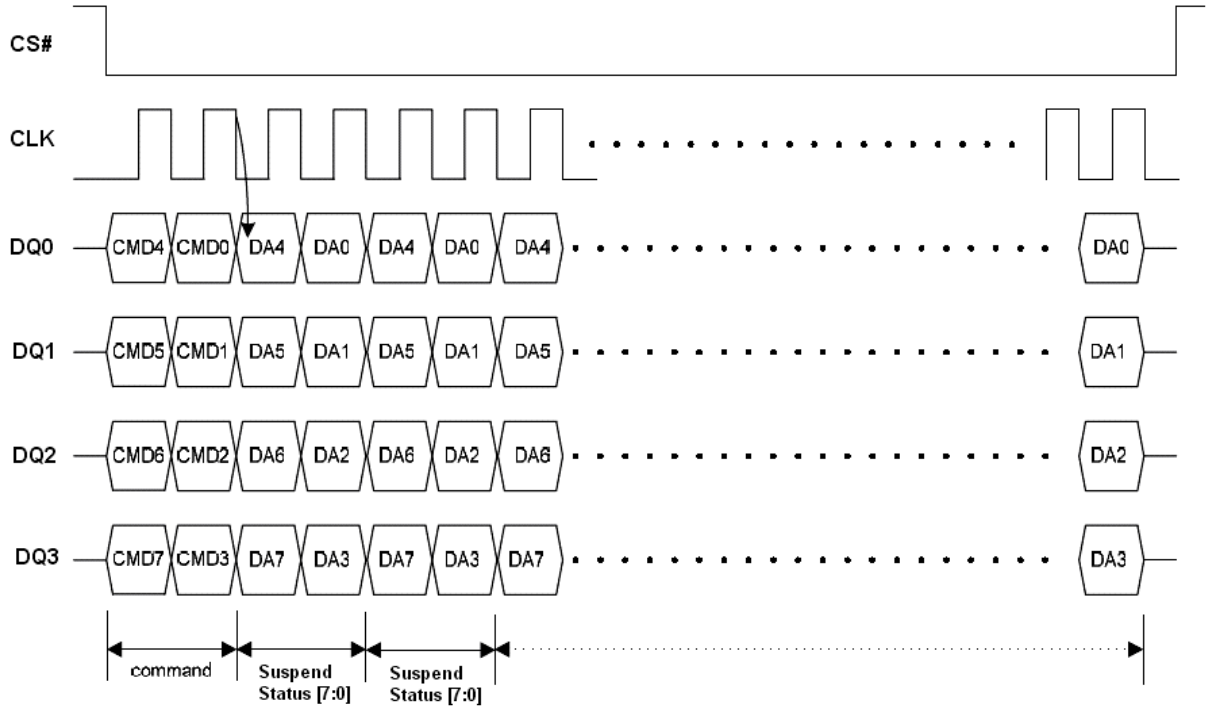



**Figure 13.1 Read Status Register 4 Instruction Sequence in QPI Mode**

**Table 10. Status Register 4 Bit Locations**

| SR4.7 | SR4.6 | SR4.5 | SR4.4 | SR4.3 | SR4.2 | SR4.1 | SR4.0 |
|---|---|---|---|---|---|---|---|
| Reserved bit | CMP (Complement Protect) | Reserved bit | Reserved bit | Reserved bit | WPDIS bit (WP# disabled) | HDEN (HOLD# enabled) | WIP bit |
| | (note 1) | | | | 1= WP# disable 0=WP# enable (default 0) | 1 = HOLD# enable 0=HOLD# disable (default 0) | |
| | Volatile/ Non-volatile bit | | | | Volatile/ Non-volatile bit | Volatile/ Non-volatile bit | |

**Note:**

1. See the table 4 “Protected Area Sizes Sector Organization”.

The status and control bits of the Status Register 4 are as follows:

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

**WPDIS bit.** The Write Protect disable (WPDIS) bit, non-volatile bit, when it is reset to “0” (factory default) to enable WP# function or is set to “1” to disable WP# function. No matter WPDIS is “0” or “1”, the system can executes Quad I/O Fast Read (EBh) or EQPI (38h) command directly. User can use Flash Programmer to set WPDIS bit as “1” and then the host system can let WP# keep floating in SPI mode.

**HDEN bit.** The HOLD# enable bit (HDEN bit), non-volatile bit, it indicates the HOLD# is enabled or not. When it is “0” (factory default), the HOLD# is disabled. On the other hand, while HDEN bit is “1”, the HOLD# is enabled. If the system executes Quad mode commands, this HDEN bit becomes no affection since HOLD# function will be disabled by Quad mode commands.

**Reserved bit.** In Status Register 4, SR4.0, SR4.3, SR4.4, SR4.5 and SR4.7 are reserved for future use.. It is recommended to mask out the reserved bit when testing the Status Register 4. Doing this will ensure compatibility with future devices.

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

The Write Status Register (WRSR) instruction allows new values to be written to the Status Register. 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 on Serial Data Input (DI).

The instruction sequence is shown in Figure 14. The Write Status Register (WRSR) instruction has no effect on SR1 and SR0 of the Status Register. Chip Select (CS#) must be driven High after the eighth bit 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 (SRP, 4KBL, TB, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in Table 4. 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.

The instruction sequence is shown in Figure 14.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

**NOTE :**

**In the OTP mode without enabling Volatile Status Register function (50h), WRSR command is used to program SPL0 bit, SPL1 bit, SPL2 bit, EBL bit, reversed, switch bit to '1', but these bits can only be programmed once.**

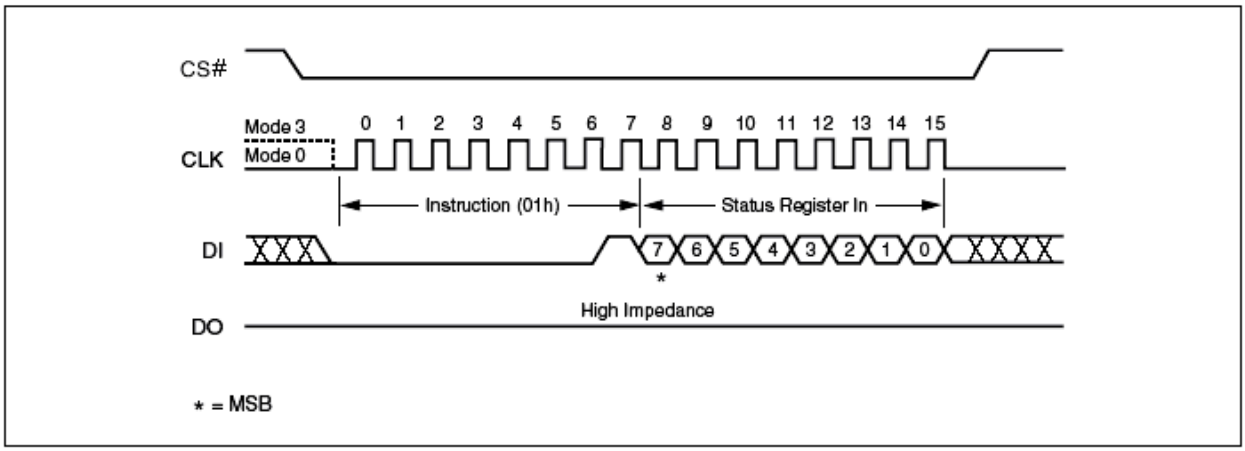



**Figure 14. Write Status Register Instruction Sequence Diagram**

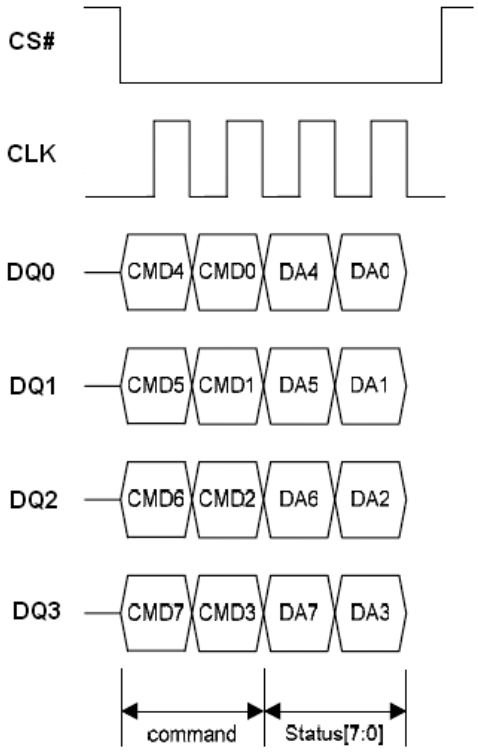



**Figure 14.1 Write Status Register Instruction Sequence in QPI Mode**

**Write Status Register 4 (C1h)**

The Write Status Register 4 (C1h) command can be used to enable or disable HOLD# and WP# pin by using HDEN and WPDIS bit and set CMP bit to protect array data. This register also can driver CS# low, sends the Write Status Register 4 (C1h) and one data byte, then drivers CS# high, please refer to Table 9 for Status Register 4 data and Figure 15 for the sequence. In QPI mode, a cycle is two nibbles, or two clocks, long, most significant nibble first.

The Write Status Register4 (WRSR4) 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 Register4 (WRSR4) instruction is not executed once the Hardware Protected Mode (HPM) is entered.

The instruction sequence is shown in Figure 15.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

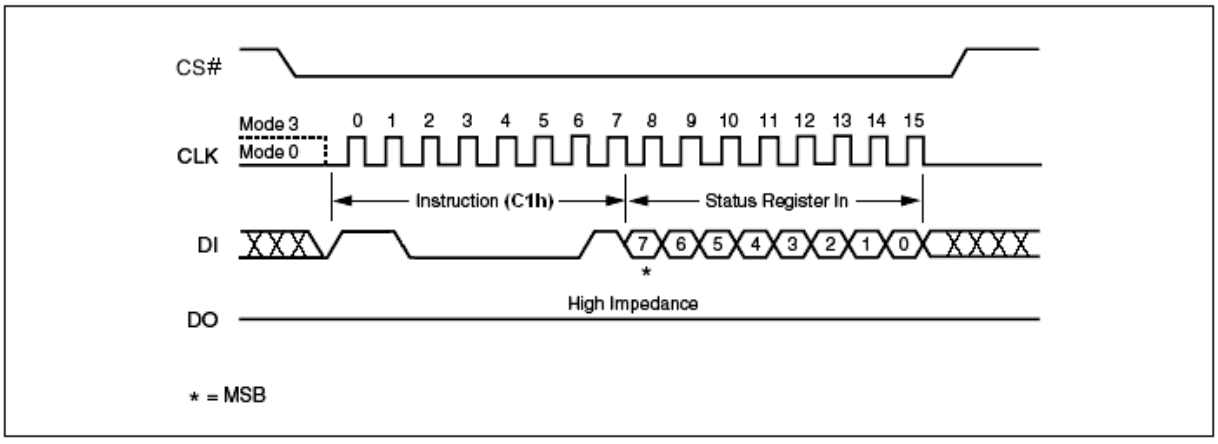



**Figure 15. Write Status Register 4 Instruction Sequence Diagram**

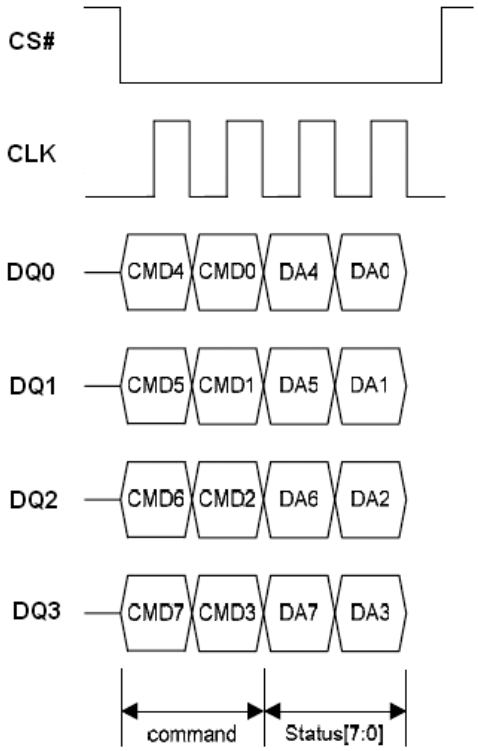



**Figure 15.1 Write Status Register 4 Instruction Sequence in QPI Mode**

**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 Figure 16. The first byte addresses 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.

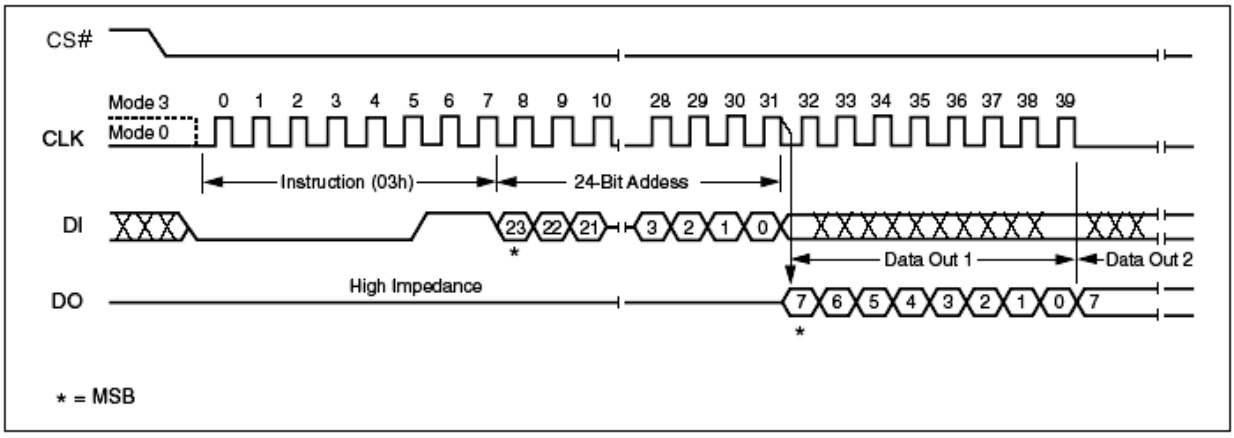



**Figure 16. Read Data Instruction Sequence Diagram**

**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 Figure 17. 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.

The instruction sequence is shown in Figure 17.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

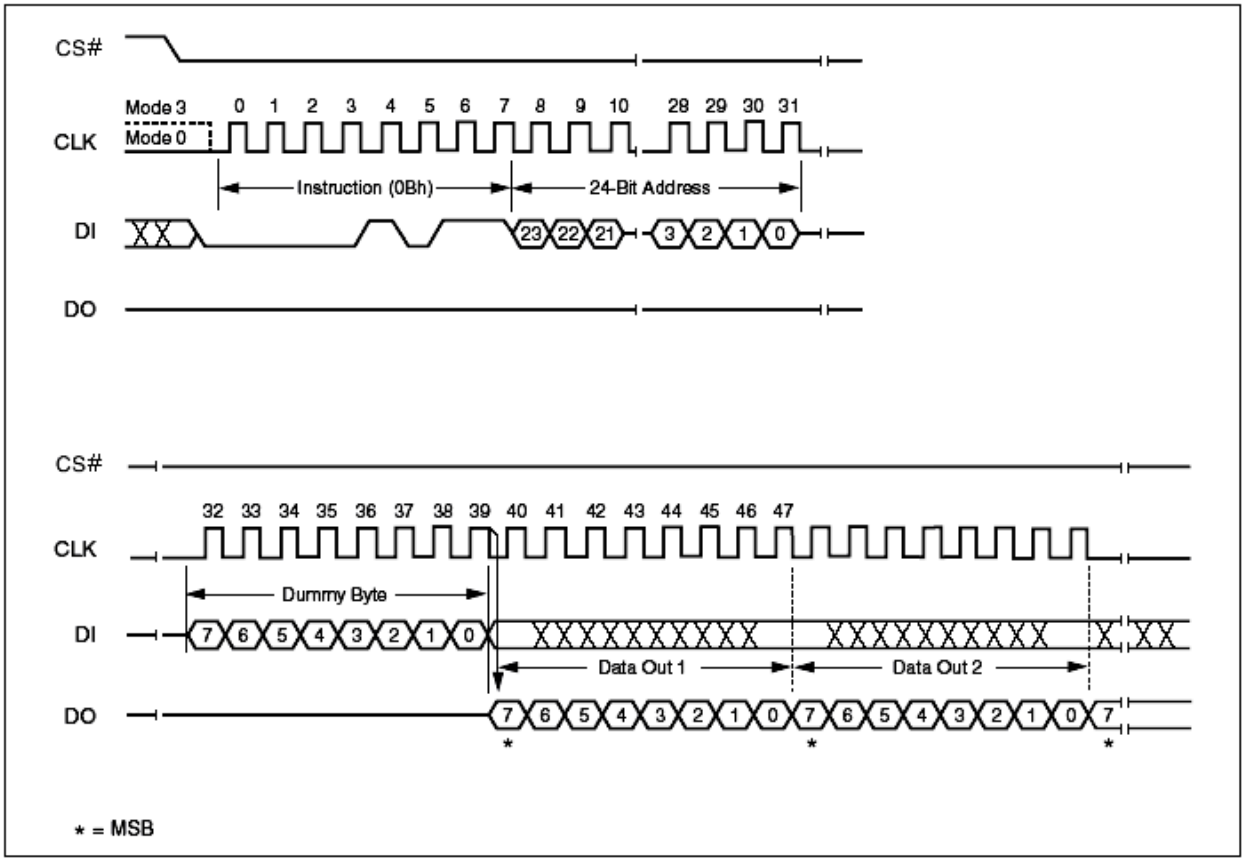



**Figure 17. Fast Read Instruction Sequence Diagram**

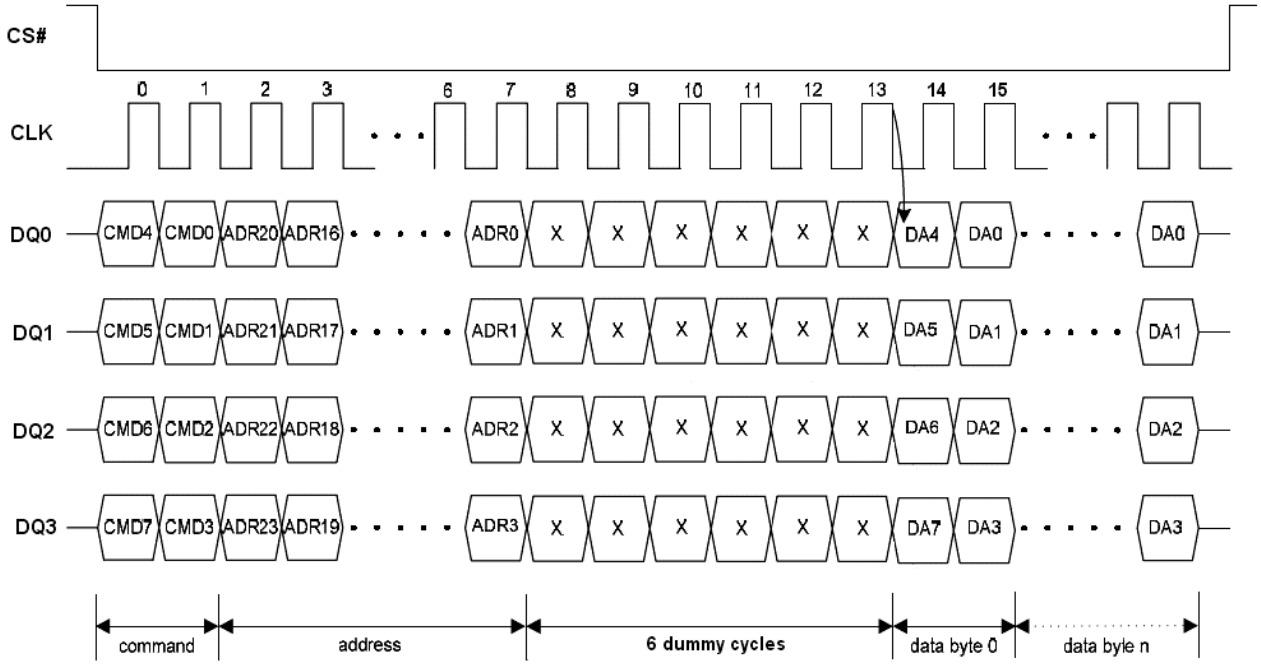



**Figure 17.1 Fast Read Instruction Sequence in QPI Mode**

**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 EN25Q80C (2A) 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 instruction 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 Figure 18. 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.

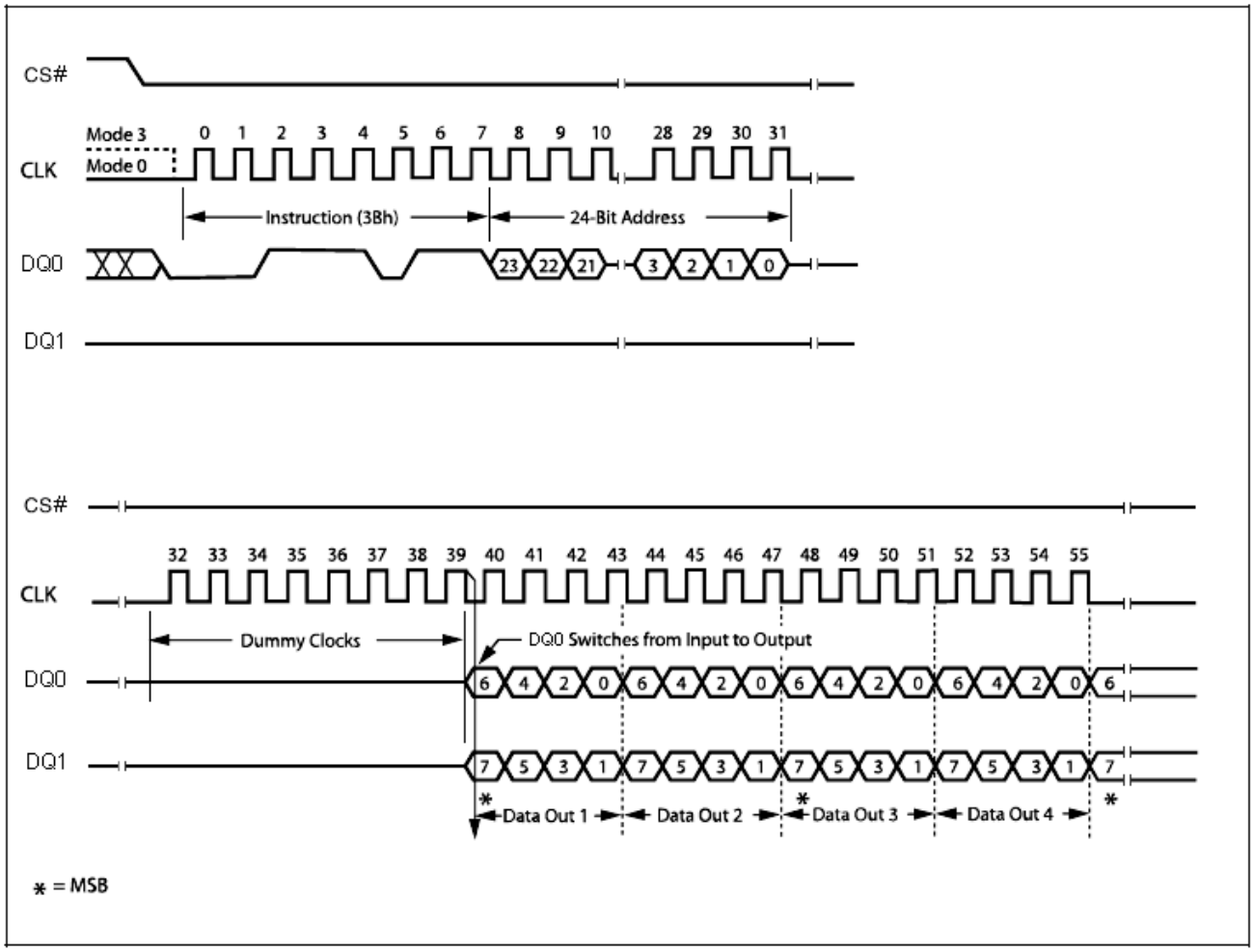



**Figure 18. Dual Output Fast Read Instruction Sequence Diagram**

**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-A0) 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 Figure 19.

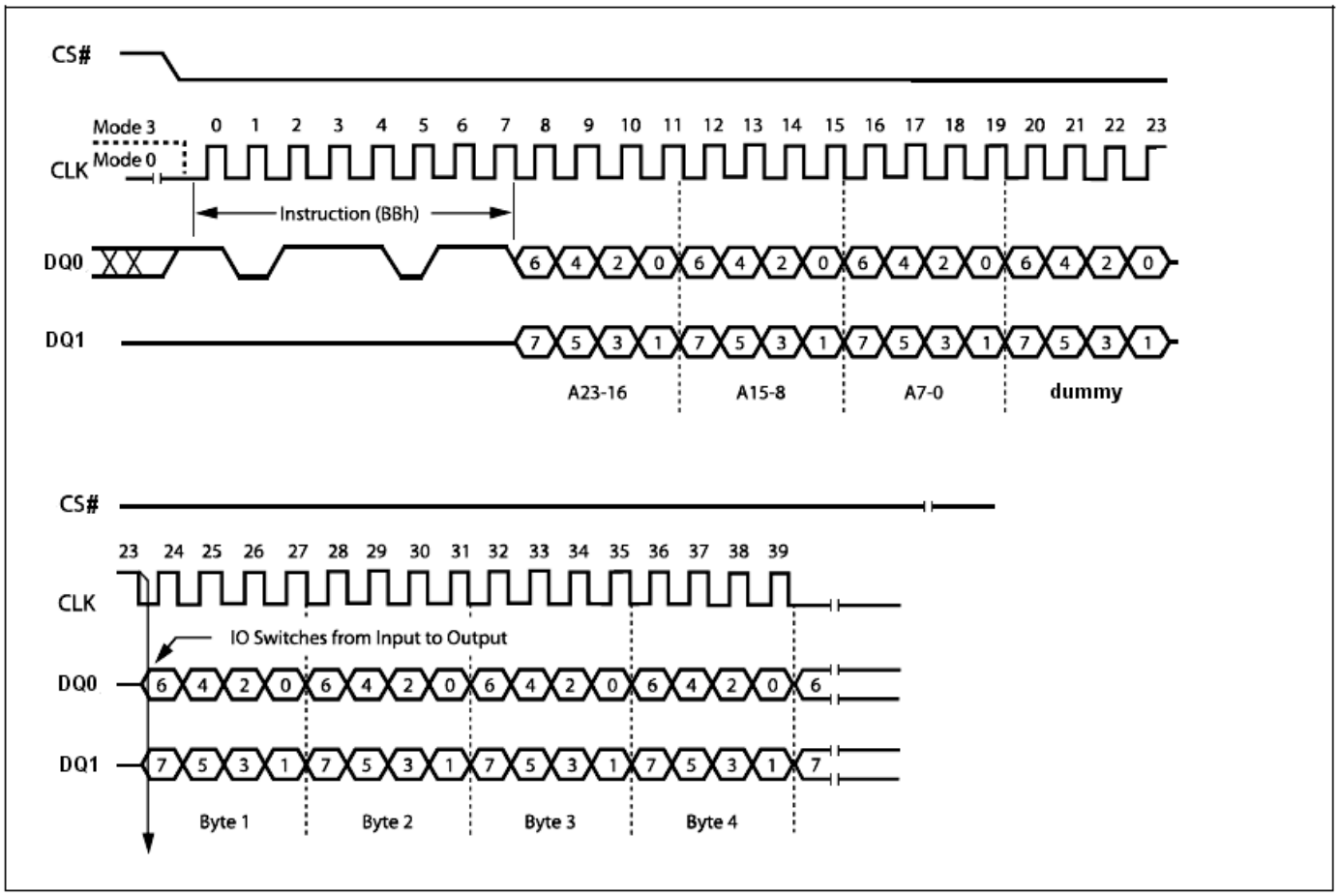



**Figure 19. Dual Input / Output Fast Read Instruction Sequence Diagram**

### 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, DQ0, DQ1, DQ2 and DQ3 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 FR. 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 DQ0 -> 8 dummy clocks -> data out interleave on DQ3, DQ2, DQ1 and DQ0 -> to end Quad Output Fast Read (6Bh) operation can use CS# to high at any time during data out, as shown in Figure 20. The WP# (DQ2) and NC(DQ3) need to drive high before address input if HDEN bit in Status Register4 is 1 and WPDIS bit is 0.

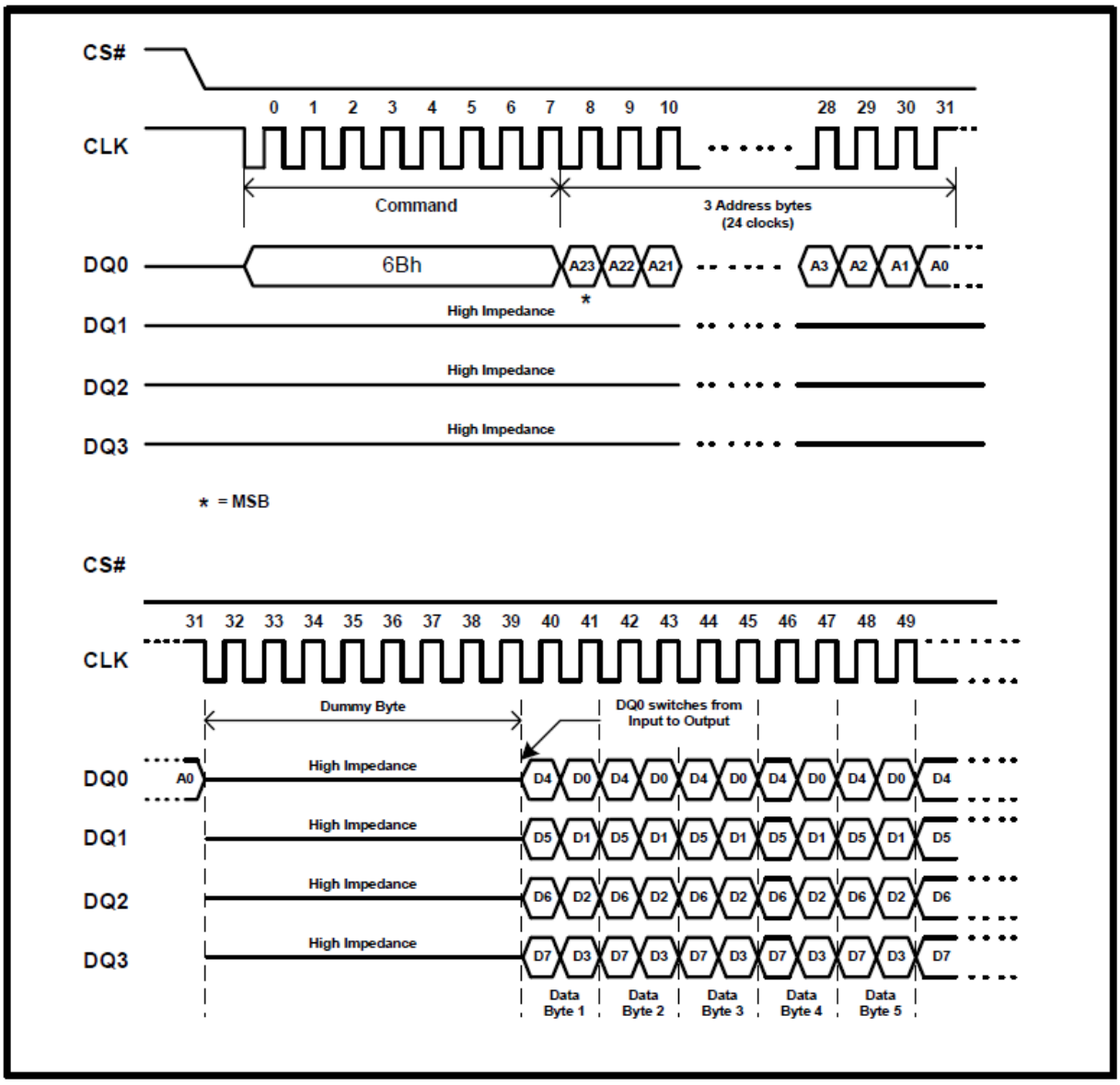



**Figure 20. Quad Output Fast Read Instruction Sequence Diagram**

**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 our 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 cycles -> 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 Figure 21.

The instruction sequence is shown in Figure 21.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

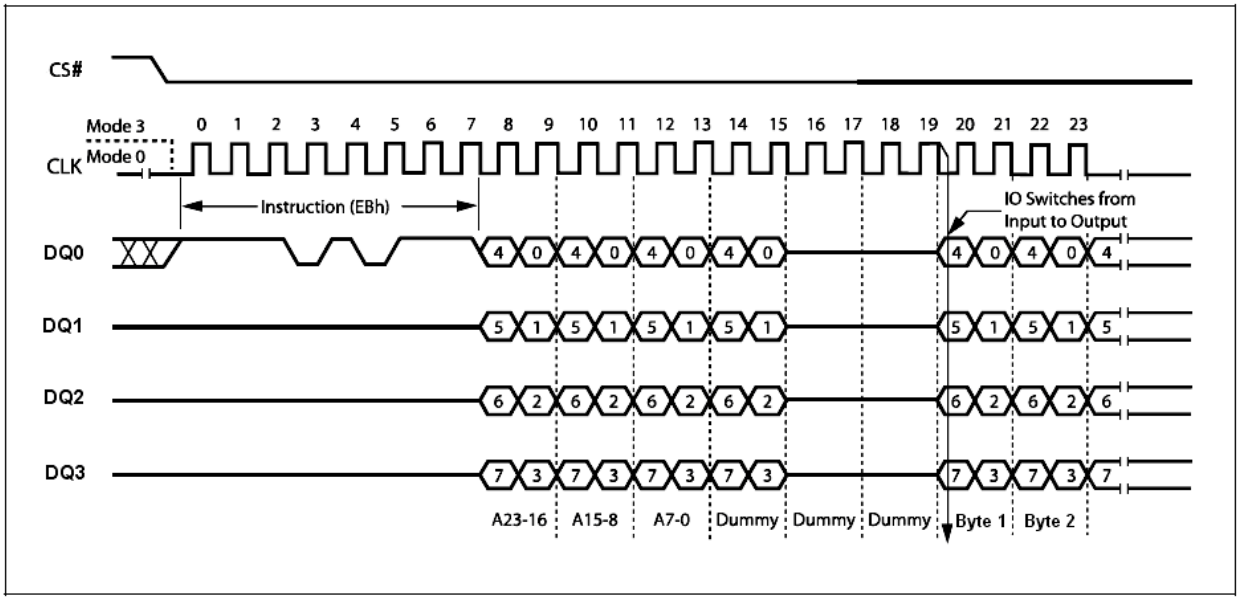



**Figure 21. Quad Input / Output Fast Read Instruction Sequence Diagram**

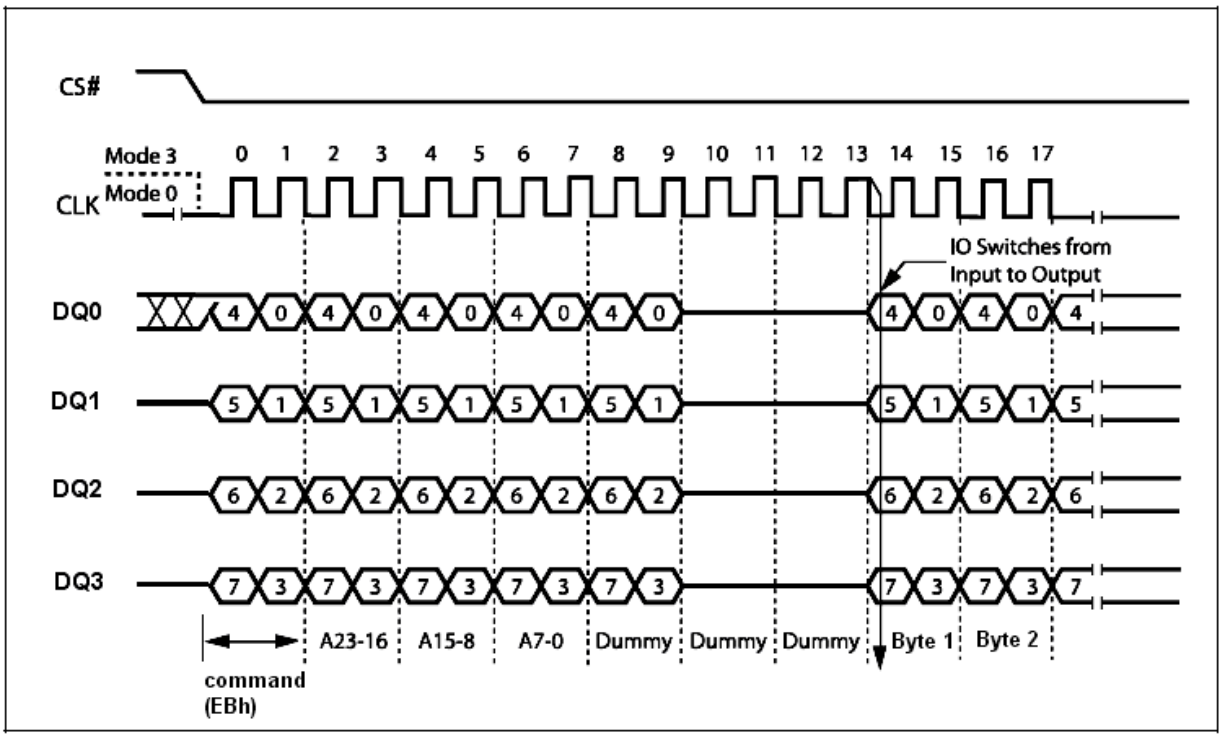



**Figure 21.1 Quad Input / Output Fast Read Instruction Sequence in QPI Mode**

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 $DQ_0$ -> performance enhance toggling bit P[7:0] -> 4 dummy cycles -> 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 access address, as shown in Figure 22.

In the performance – enhancing mode, P[7:4] must be toggling with P[3:0] ; likewise P[7:0] = A5h, 5Ah, F0h or 0Fh can make this mode continue and reduce the next Quad Input/Output FAST_READ (EBh) instruction. Once P[7:4] is no longer toggling with P[3:0] ; likewise P[7:0] = FFh, 00h, AAh or 55h. And afterwards CS# is raised, the system then will escape from performance enhance mode and return to normal operation.

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.

The instruction sequence is shown in Figure 22.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

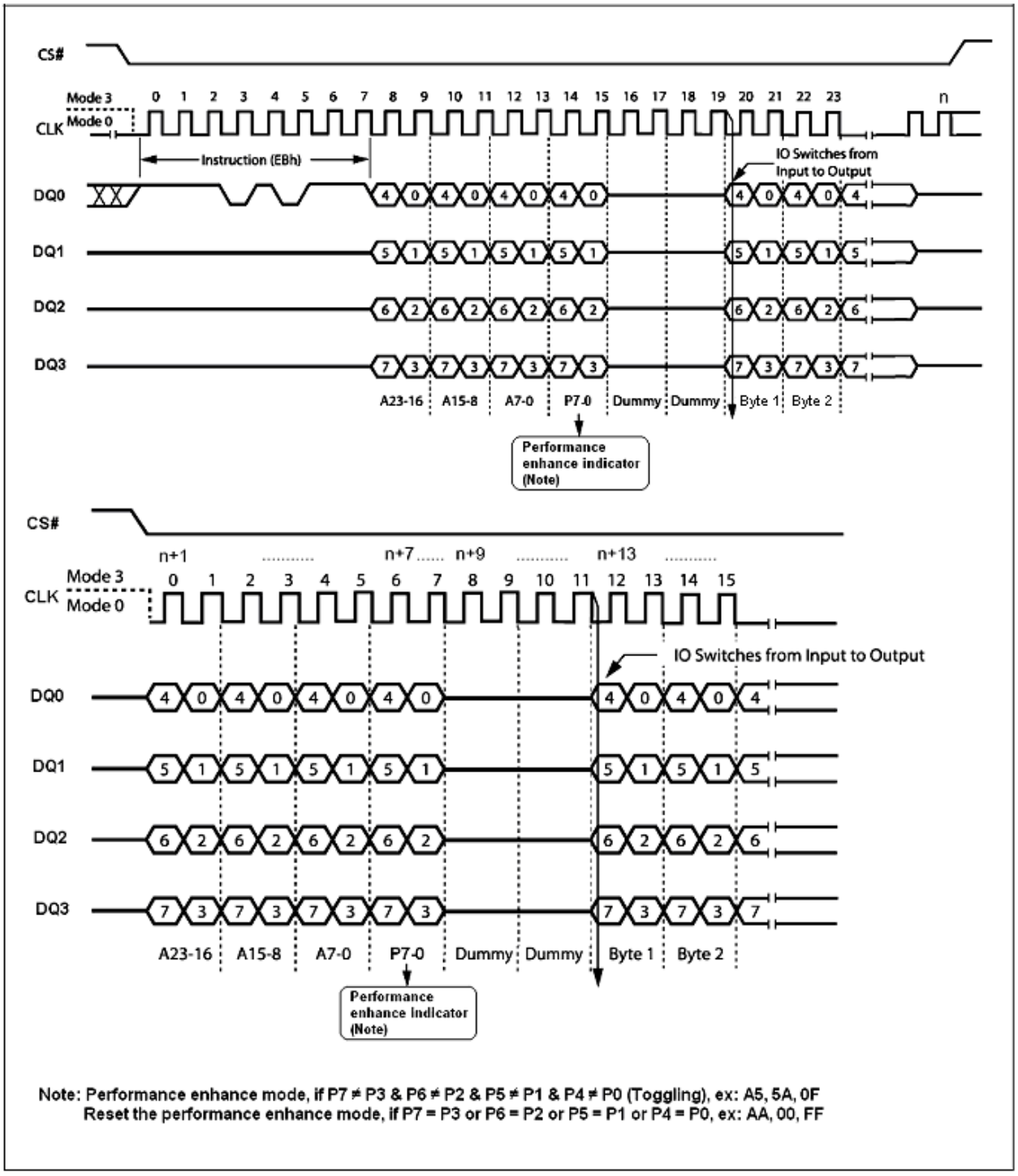
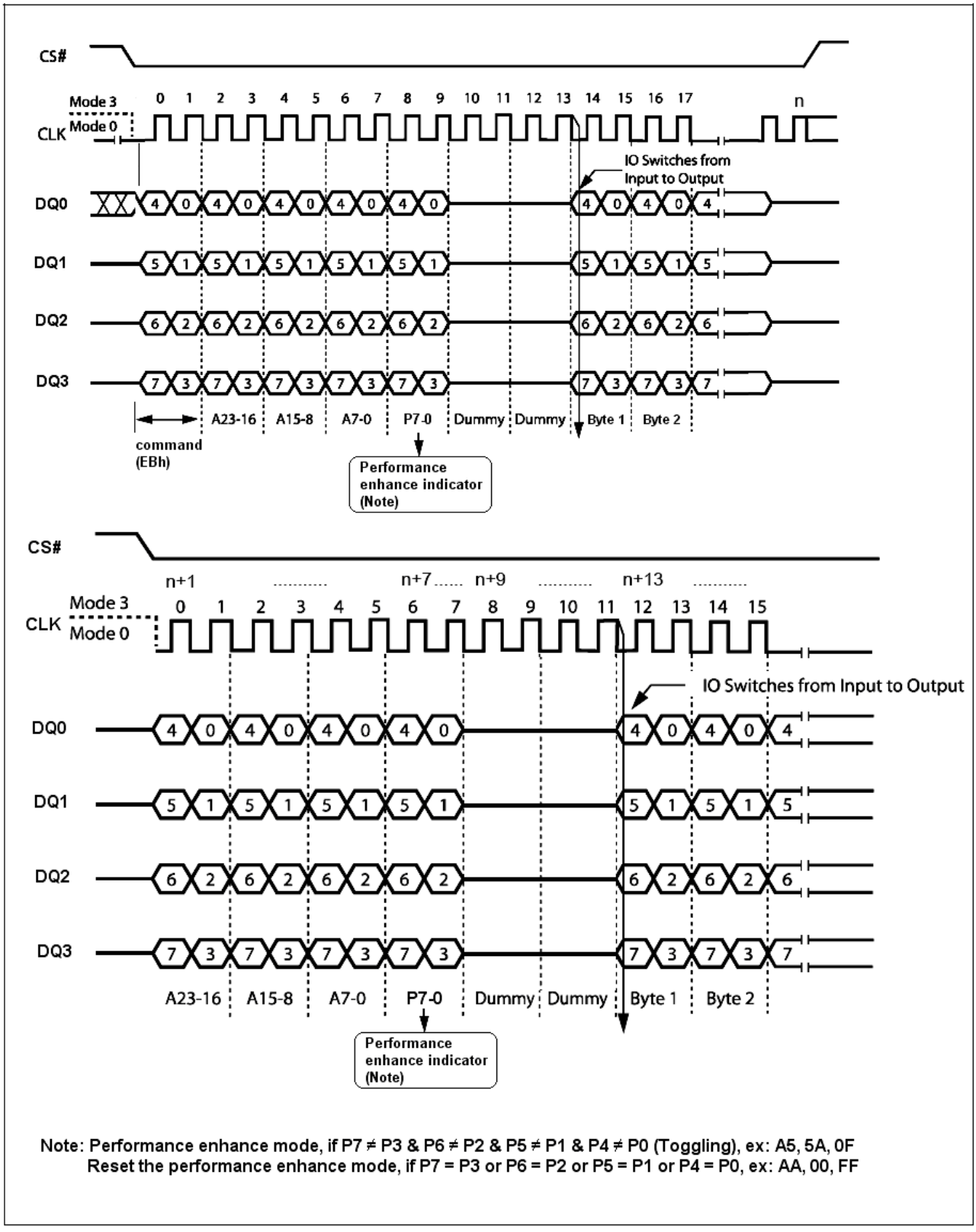



**Figure 22. Quad Input/Output Fast Read Enhance Performance Mode Sequence Diagram**




Note: Performance enhance mode, if P7 ≠ P3 & P6 ≠ P2 & P5 ≠ P1 & P4 ≠ P0 (Toggling), ex: A5, 5A, 0F
Reset the performance enhance mode, if P7 = P3 or P6 = P2 or P5 = P1 or P4 = P0, ex: AA, 00, FF

**Figure 22.1 Quad Input/Output Fast Read Enhance Performance Mode Sequence in QPI Mode**

**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 Figure 23. 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 Table 4) is not executed.

The instruction sequence is shown in Figure 23.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

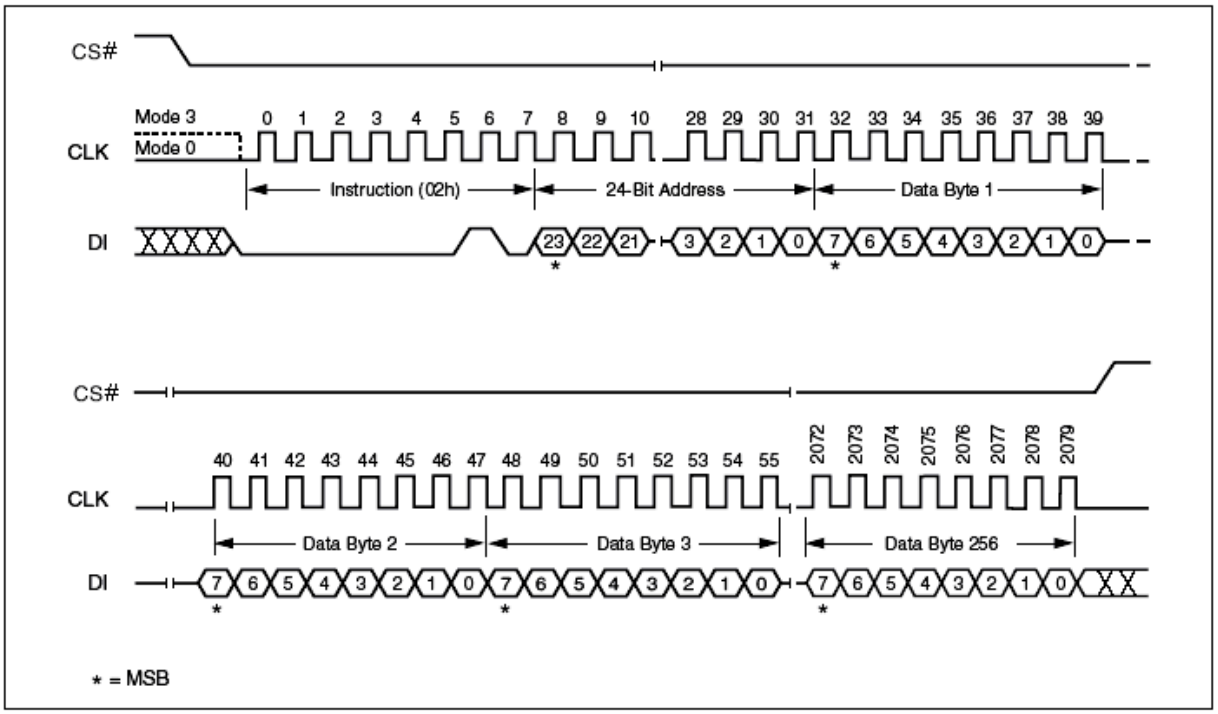



**Figure 23. Page Program Instruction Sequence Diagram**

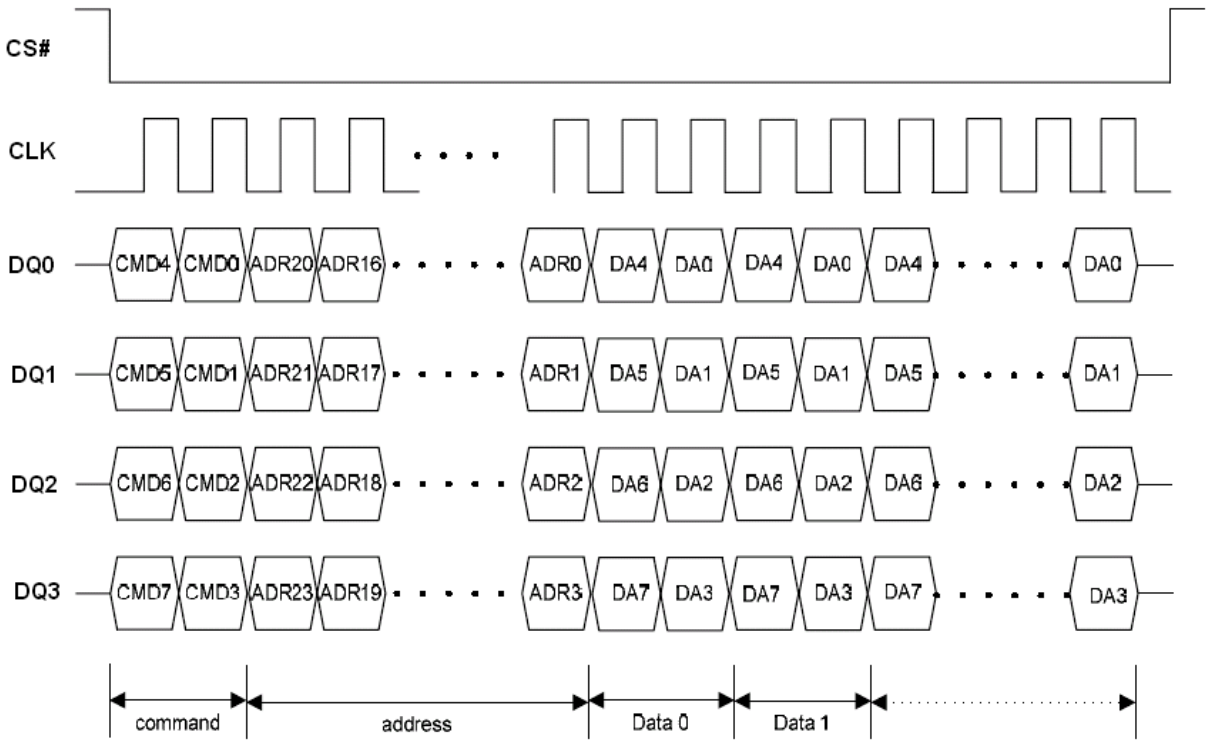



**Figure 23.1 Program Instruction Sequence in QPI Mode**

**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 Hold# Enable(HDEN) bit in Status Register must be set to 0. A Write Enable instruction must be executed before the device will accept the Quad Page Program (QPP) instruction (SR1, 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 Figure 24.

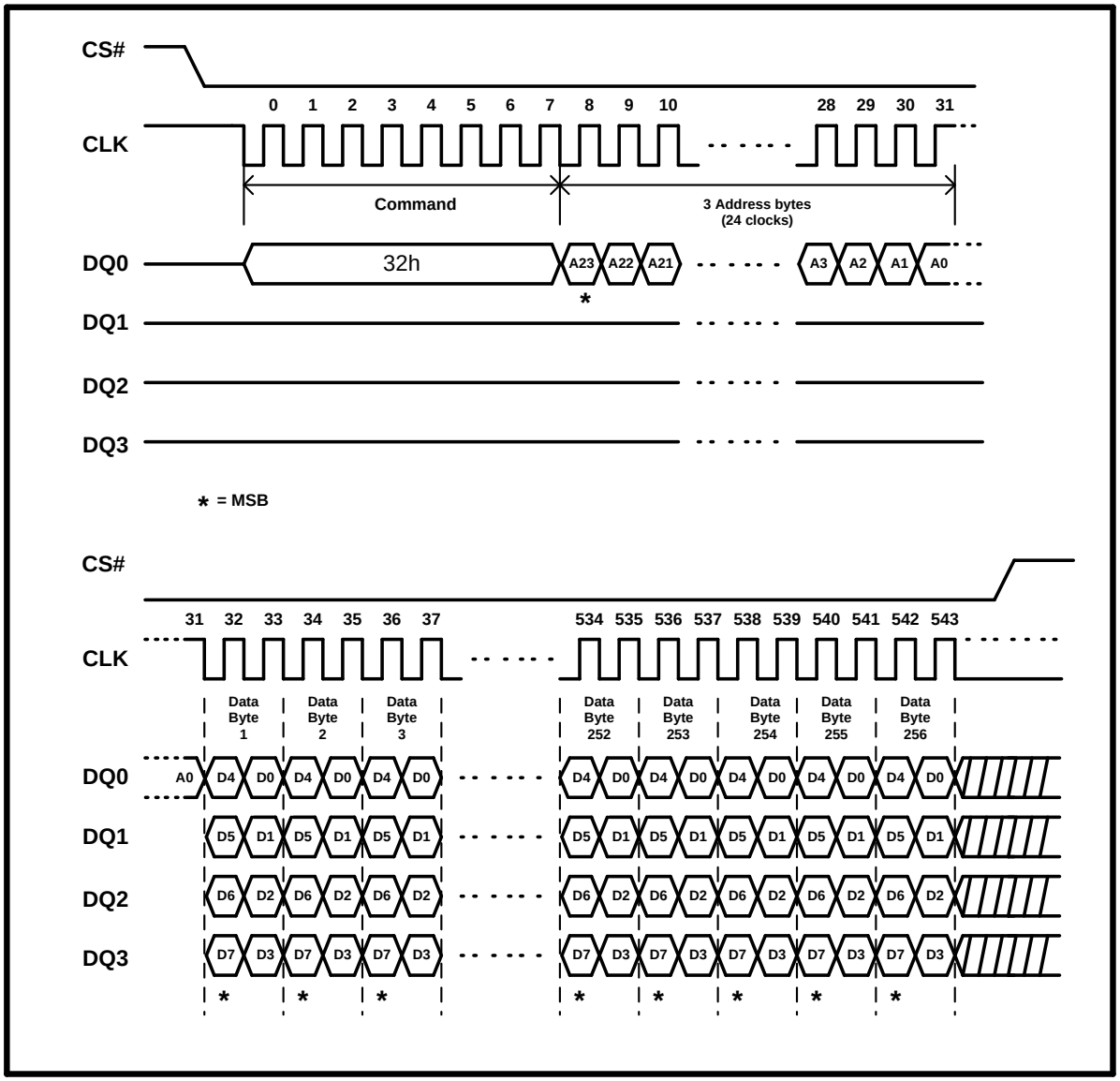



**Figure 24. Quad Input Page Program Instruction Sequence Diagram (SPI Mode only)**

**Write Suspend (B0h)**

Write Suspend allows the interruption of Sector Erase, Block Erase or Page Program operations in order to read data in another portion of memory. The original operation can be continued with Write Resume command. The instruction sequence is shown in Figure 25.

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: 20us.

Resume to another suspend timing: 5ms.

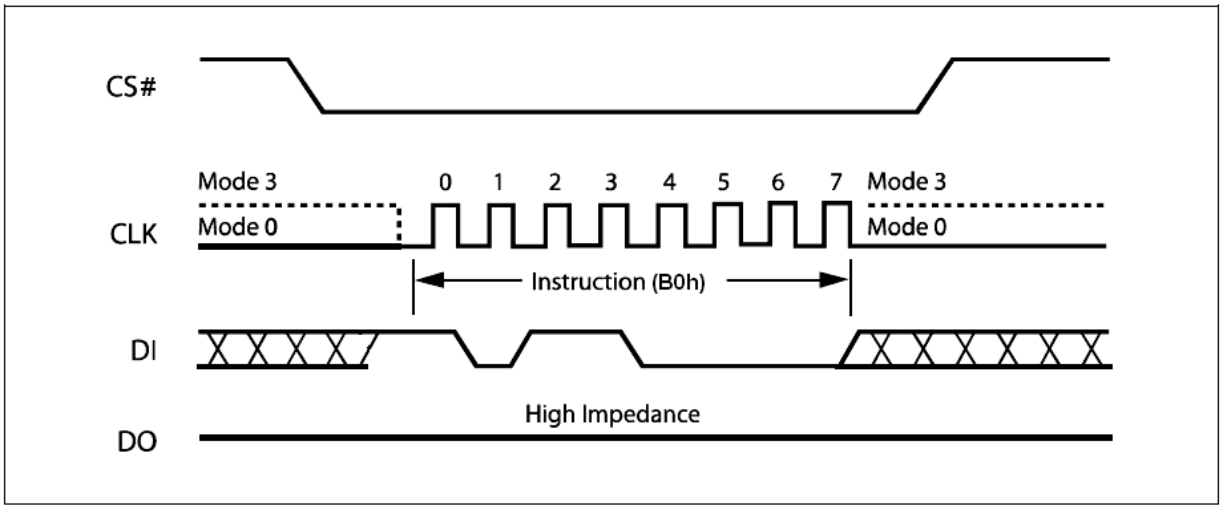



**Figure 25. Write Suspend Instruction Sequence Diagram**

**Write Suspend During Sector Erase or Block Erase**

Issuing a Write Suspend instruction during Sector Erase or Block Erase allows the host to read any sector except erase suspended block. The device can only accept Read and Read related instructions (EQPI, RSTQIO, RDSR, RDSR2, RDSR4, Write Resume, RDI, Read Manufacturer / Device ID, RDID, Enter OTP mode, and Read SFDP and Unique ID Number) after Write Suspend instruction during Sector Erase or Block Erase. Any attempt to read from the suspended block will out put 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 Status Register 2 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 Status Register 2 or after issue program suspend command, latency time 20us is needed before issue another command. For “Suspend to Read”, “Resume to Read”, “Resume to Suspend” timing specification please note Figure 26.1, 26.2 and 26.3.

**Write Suspend During Page Programming**

Issuing a Write Suspend instruction during Page Programming allows the host to read any sector except program suspended block. The device can only accept Read and Read related instructions (EQPI, RSTQIO, RDSR, RDSR2, RDSR4, Write Resume, RDI, Read Manufacturer / Device ID, RDID, Enter OTP mode, and Read SFDP and Unique ID Number) after Write Suspend instruction during Page Programming. Any attempt to read from the suspended block 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 Status Register 2 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 Status Register 2 or after issue program suspend command, latency time 20us is needed before issue another command. For "Suspend to Read", "Resume to Read", "Resume to Suspend" timing specification please note Figure 26.1, 26.2 and 26.3.

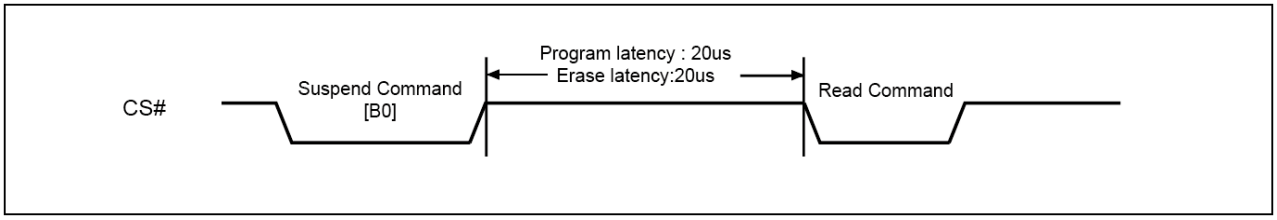



**Figure 26.1 Suspend to Read Latency**

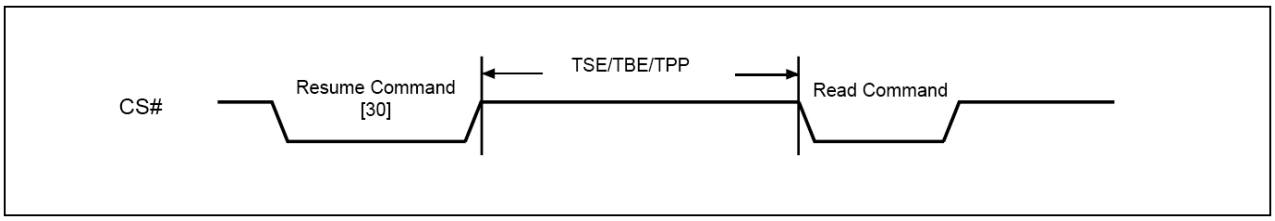



**Figure 26.2 Resume to Read Latency**

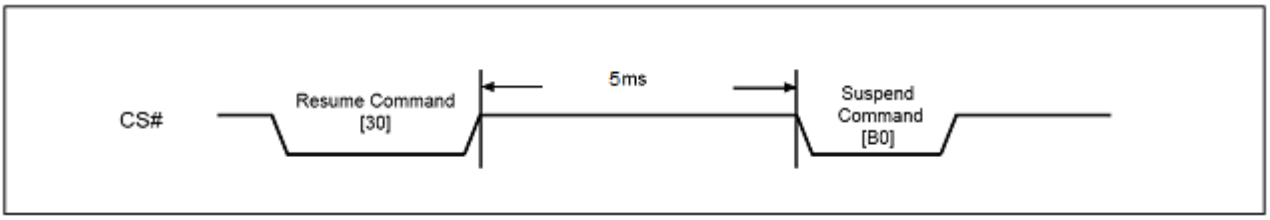



**Figure 26.3 Resume to Suspend Latency**

The instruction sequence is shown in Figure 27.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

**Write Resume (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 Figure 27. To execute a Write Resume operation, the host drives CS# low, sends the Write Resume command cycle (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 5ms.

The instruction sequence is shown in Figure 27.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

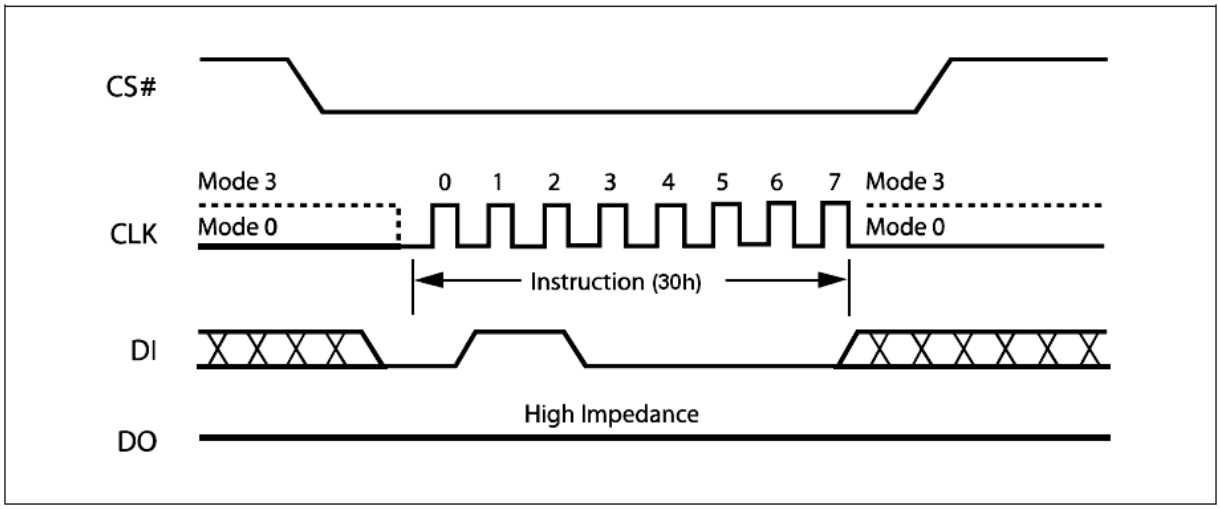



**Figure 27. Write Resume Instruction Sequence Diagram**

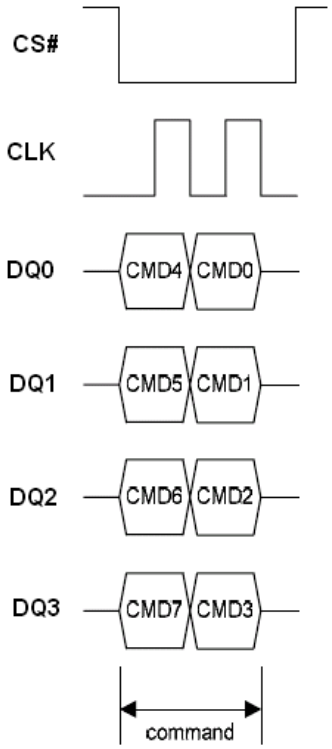



**Figure 27.1. Write Suspend/Resume Instruction Sequence in QPI Mode**

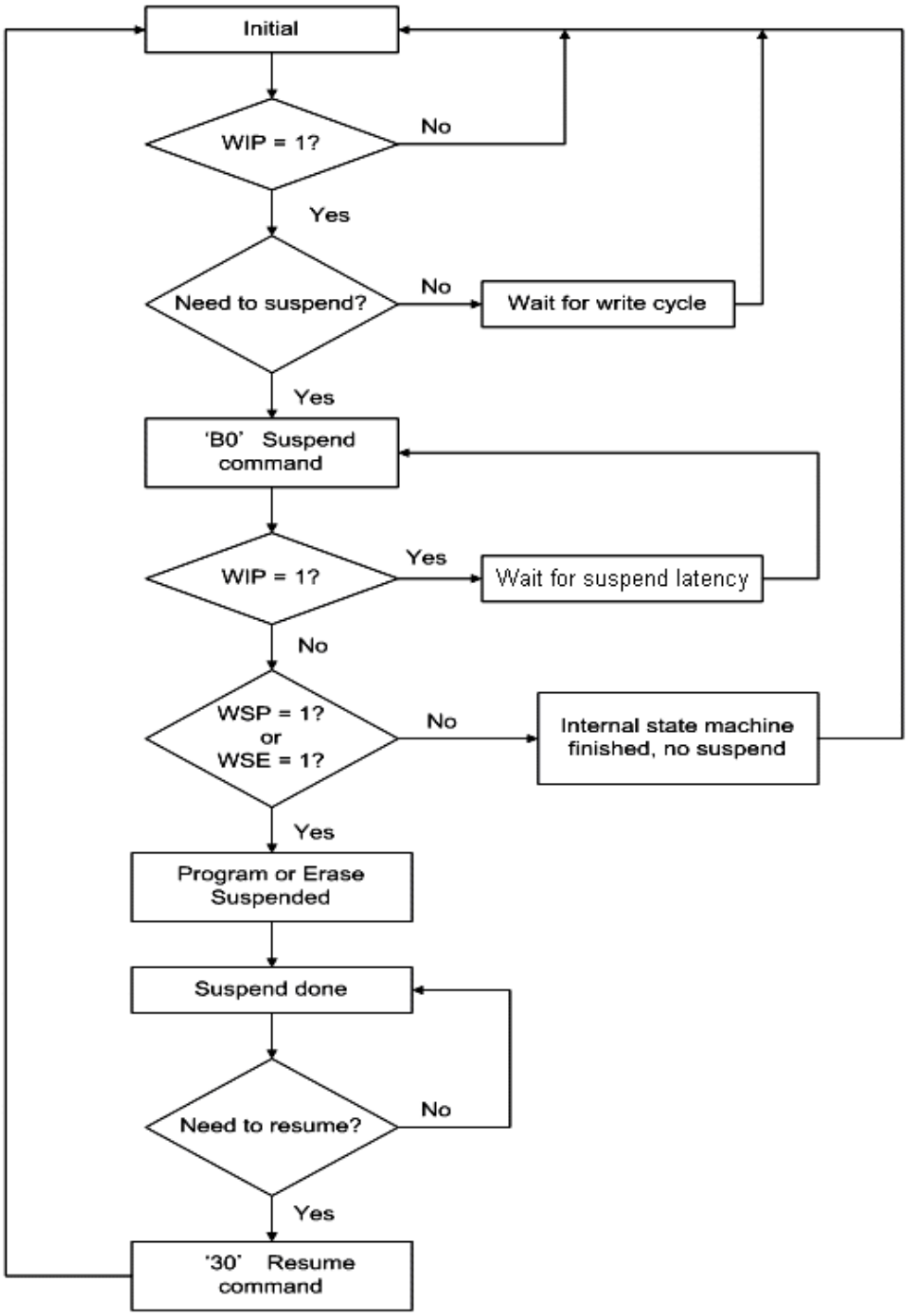



**Figure 27.2. Write Suspend/Resume Flow**

**Note:**

1. The 'WIP' can be either checked by command '09'or '05' 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 20us is needed before issue another command or polling the WIP.
4. The 'WSP' and 'WSE' can be checked by command '09' 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 Table 3) 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 Figure 28. 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 Table 4) or Boot Lock feature will be ignored.

The instruction sequence is shown in Figure 30.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

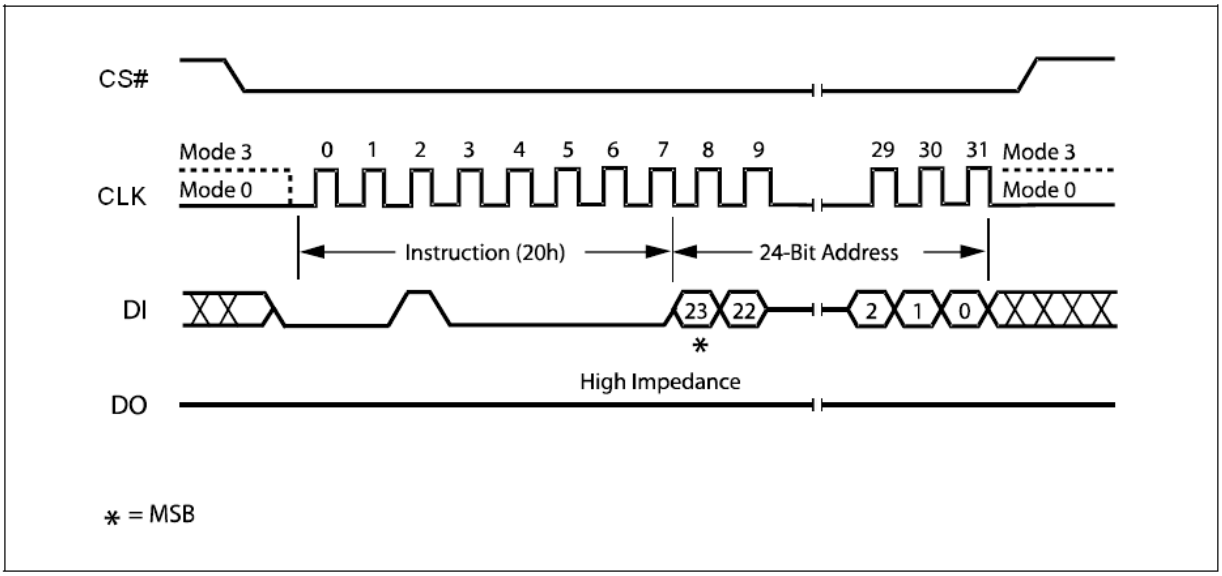



**Figure 28. Sector Erase Instruction Sequence Diagram**

**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 Table 3) 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 Figure 29. 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 Half Block Erase cycle (whose duration is $t_{HBE}$) 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 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 Table 4) or Boot Lock feature will be ignored.

The instruction sequence is shown in Figure 30.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

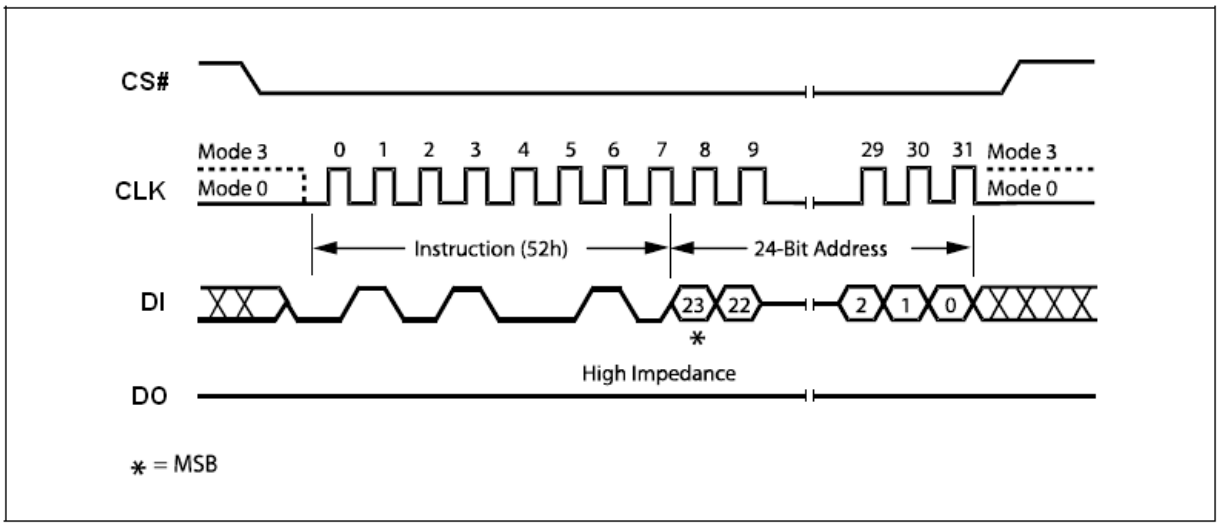



**Figure 29. 32KB Half Block Erase Instruction Sequence Diagram**

**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 Table 3) 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 Figure 30. 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 Table 4) or Boot Lock feature will be ignored.

The instruction sequence is shown in Figure 30.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

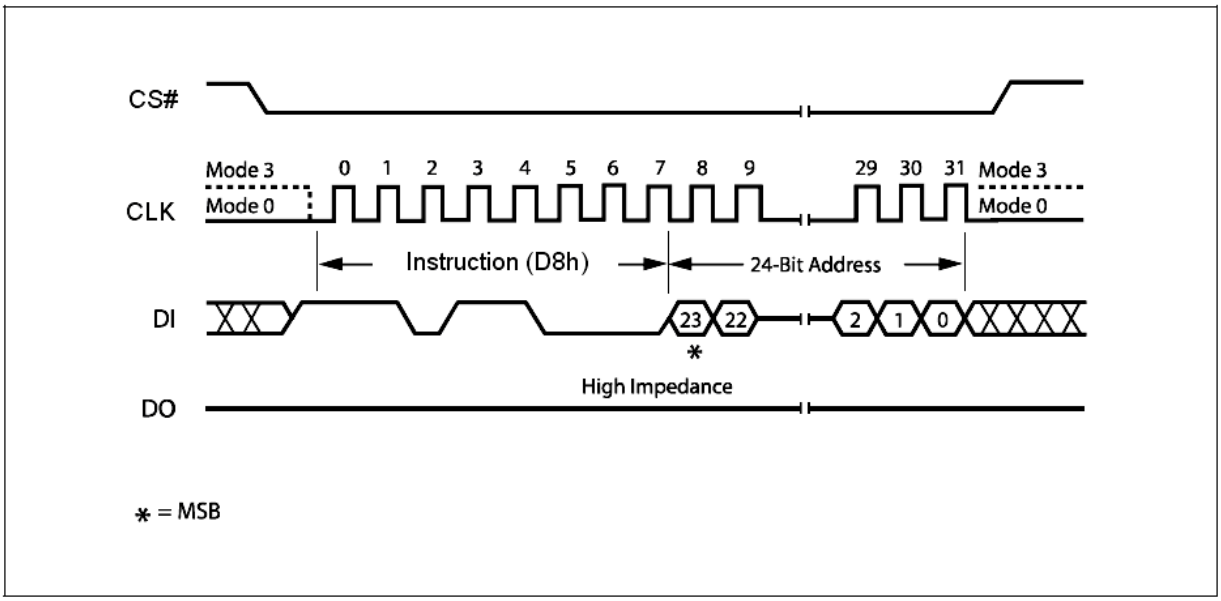



**Figure 30. 64KB Block Erase Instruction Sequence Diagram**

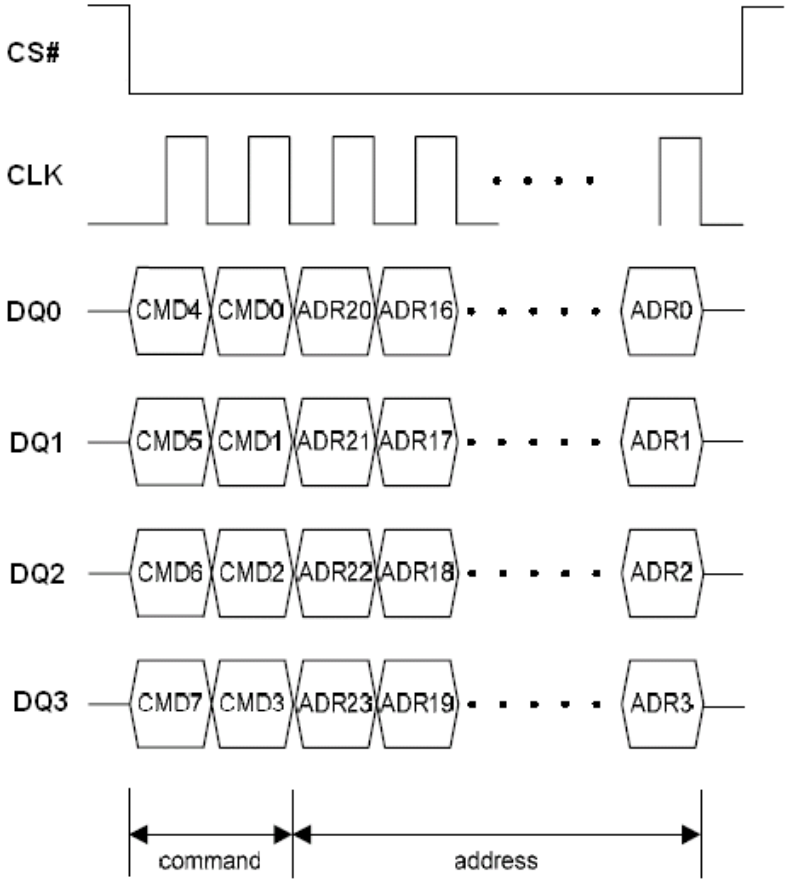



**Figure 30.1 Half Block/Block/Sector Erase Instruction Sequence in QPI Mode**

**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 Figure 31. 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.

The instruction sequence is shown in Figure 31.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

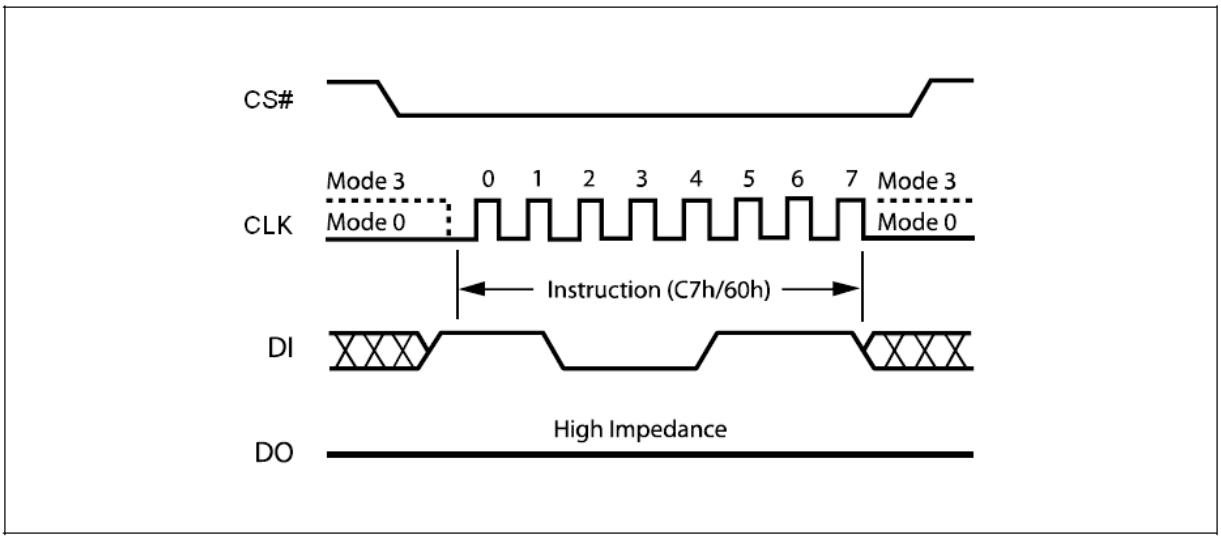



**Figure 31. Chip Erase Instruction Sequence Diagram**

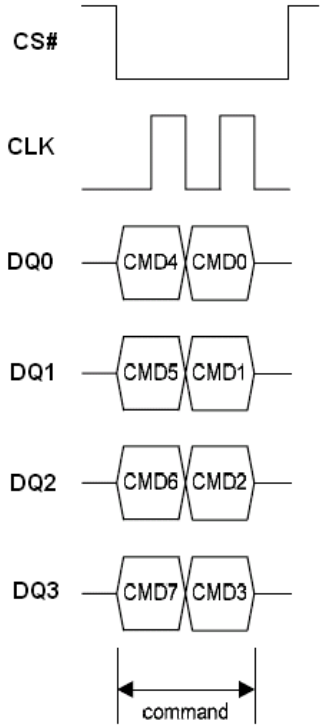



**Figure 31.1 Chip Erase Sequence in QPI Mode**

**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 Table 15.)

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. This releases 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 Figure 32. 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.

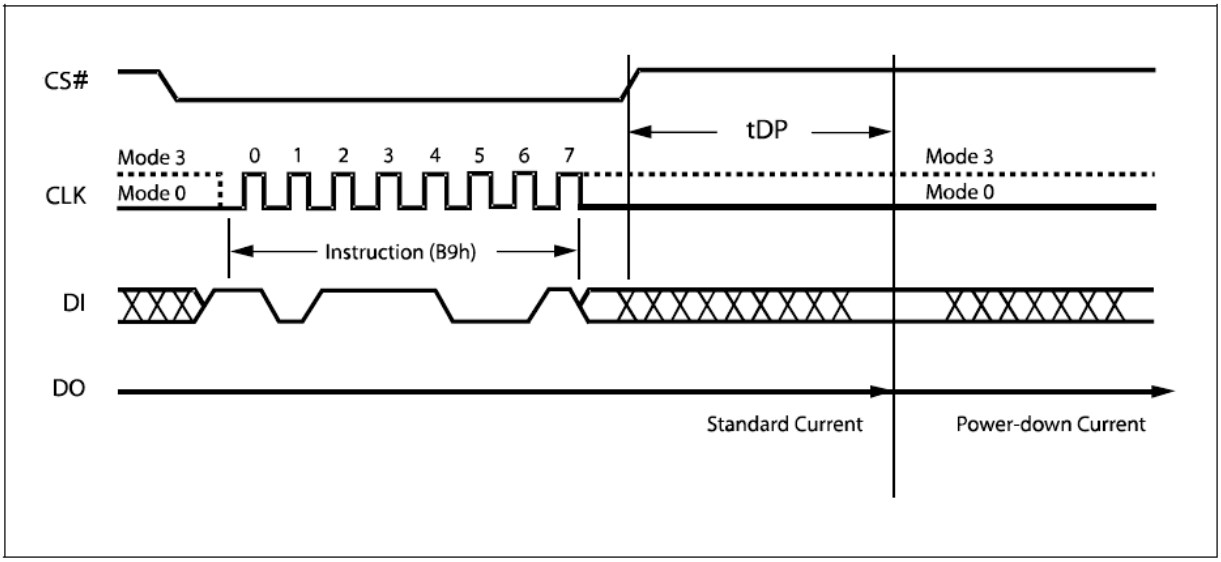



**Figure 32. Deep Power-down Instruction Sequence Diagram**

**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 Figure 33. 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 Figure 34. The Device ID value for the EN25Q80C (2A) are listed in Table 7. 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 Table 17. 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.

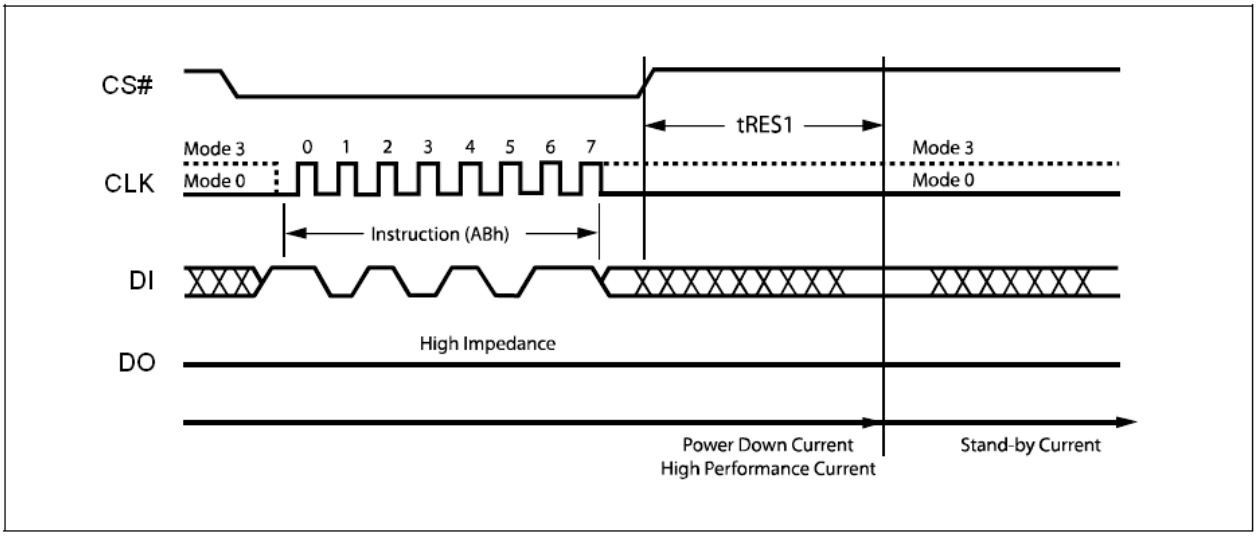



**Figure 33. Release Power-down Instruction Sequence Diagram**

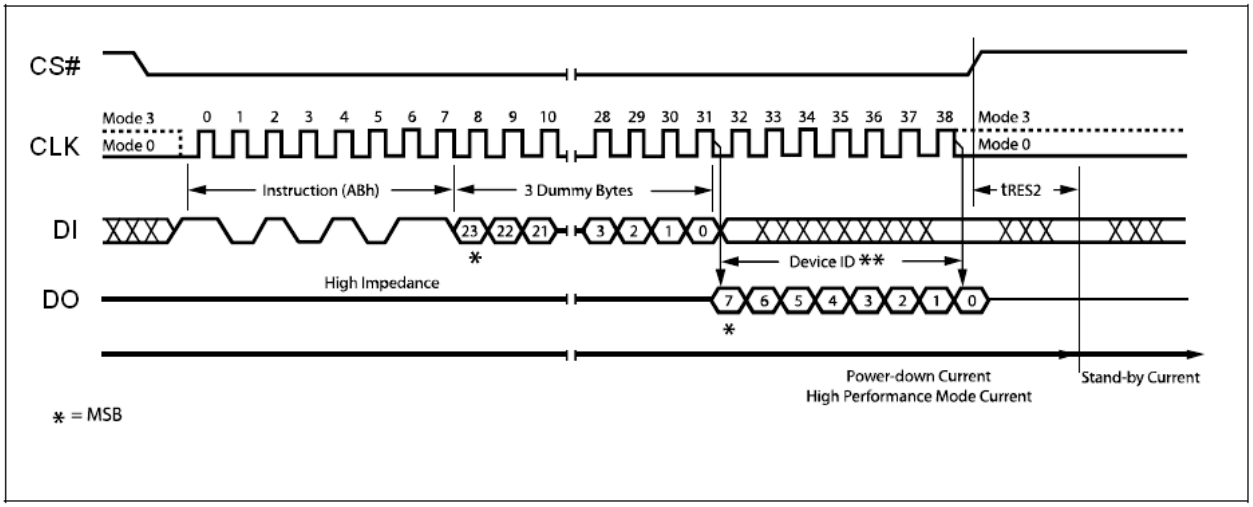



**Figure 34. Release Power-down / Device ID Instruction Sequence Diagram**

**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 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 Figure 35. The Device ID values for the EN25Q80C (2A) are listed in Table 7. If the 24-bit address is initially set to 000001h the Device ID will be read first

The instruction sequence is shown in Figure 35.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

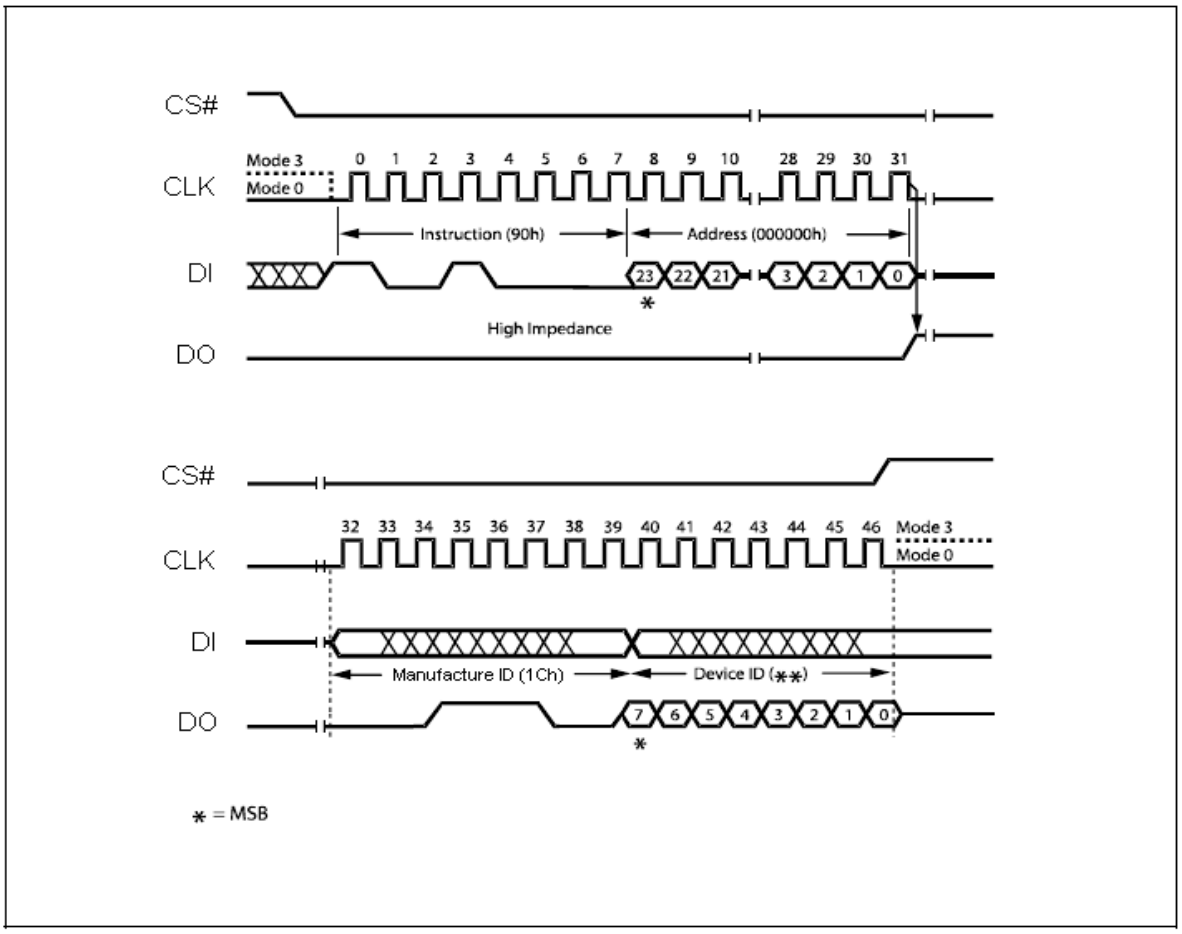



**Figure 35. Read Manufacturer / Device ID Diagram**

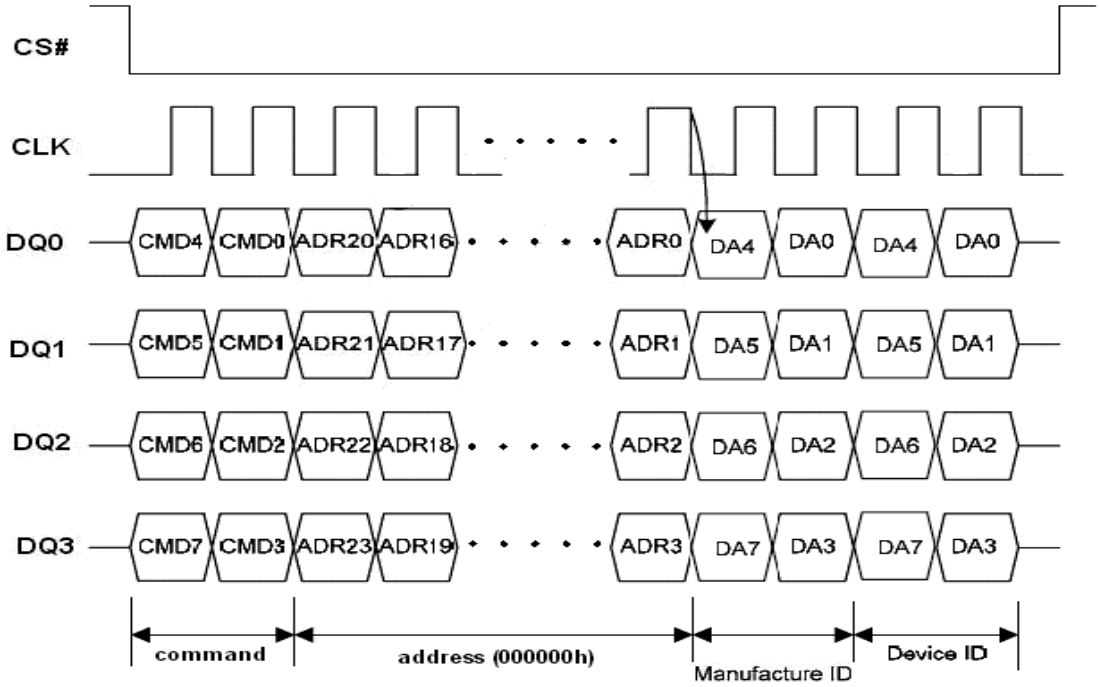



**Figure 35.1 Read Manufacturer / Device ID Diagram in QPI Mode**

**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 Figure 36. 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.

The instruction sequence is shown in Figure 36.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

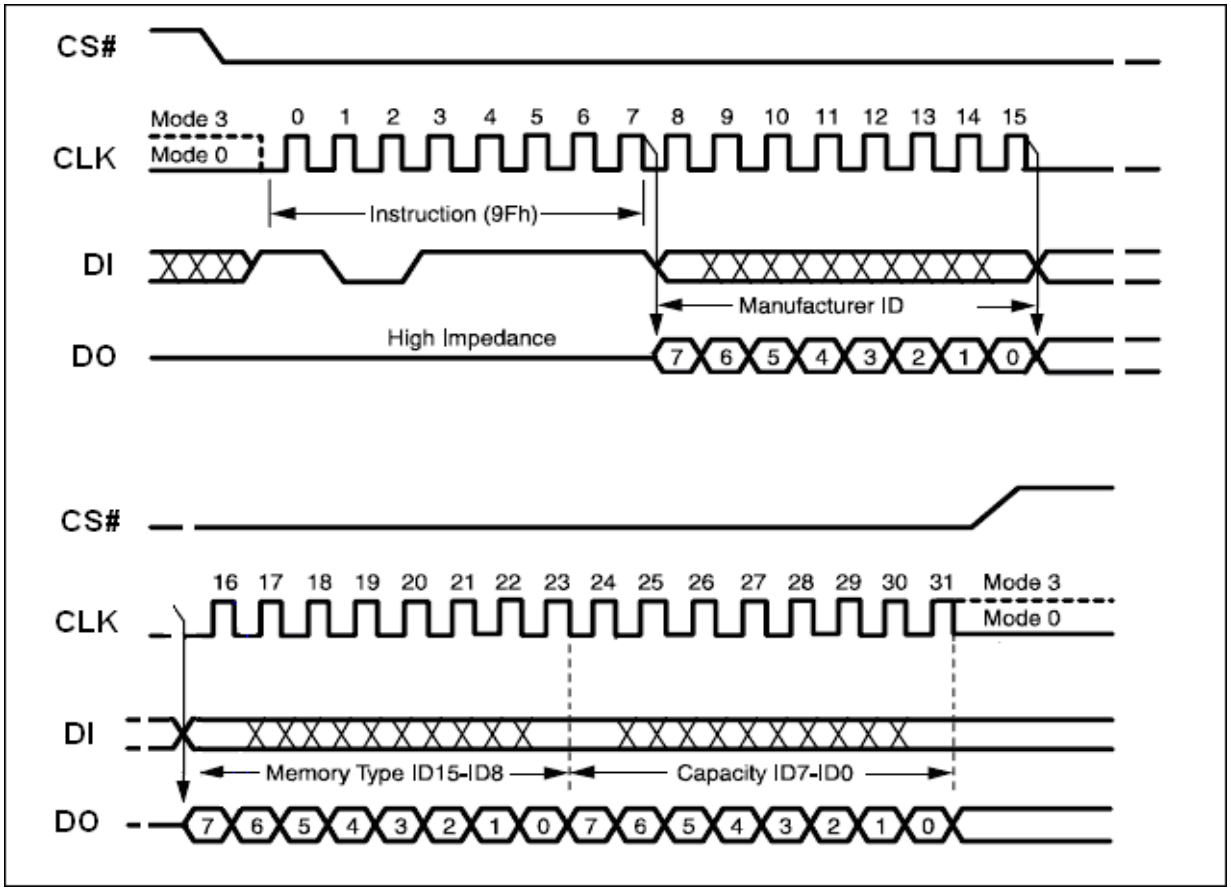



**Figure 36. Read Identification (RDID)**

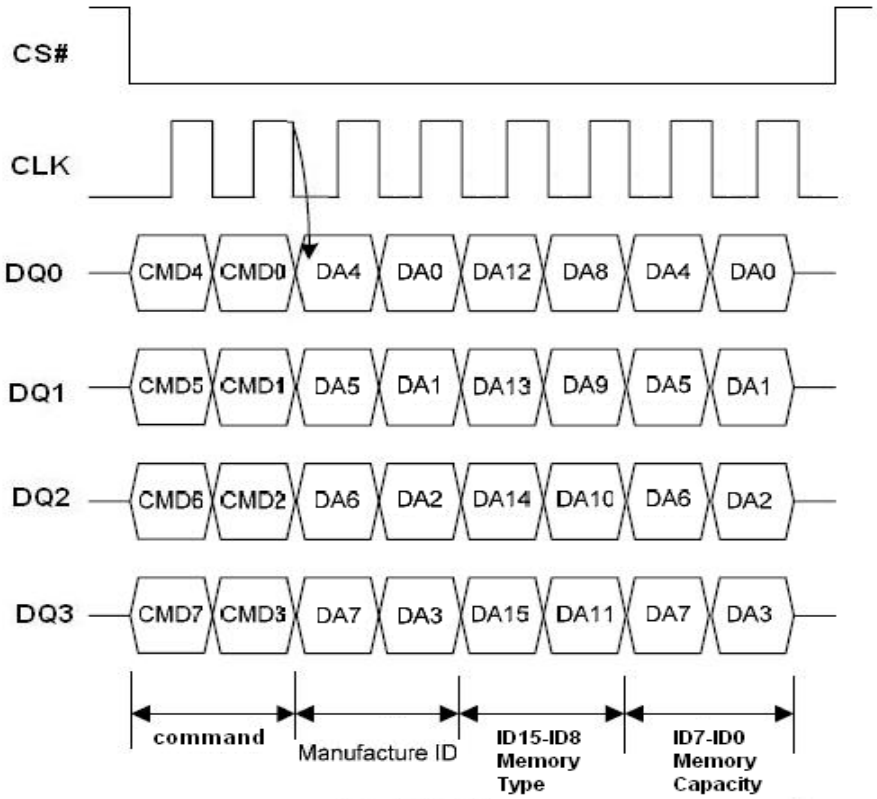



**Figure 36.1 Read Identification (RDID) in QPI Mode**

### Enter OTP Mode (3Ah)

This Flash support OTP mode to enhance the data protection, user can use the Enter OTP mode (3Ah) command for entering this mode. In OTP mode, the Status Register SR7 bit is served as SPL0 bit, SR3 bit is served as EBL bit, SR2 bit is served as SPL1 bit, SR1 bit is served as SPL2 bit and SR0 bit is served as WIP bit. They can be read by RDSR command.

This Flash has 3 OTP sectors which density are 512byte, 512byte and 512byte, user must issue ENTER OTP MODE command to read, program or erase OTP sectors. After entering OTP mode, the OTP sectors ~~is~~are mapping to sector 255, 254, and 240, and, **SRP bit** becomes SPL0 bit, BP0 bit becomes SPL1 bit, WEL bit become SPL2 bit. The Chip Erase, Block Erase and Half Block Erase commands are also disabled.

The OTP sectors can ***only*** be erased by Sector Erase (20h) commands. The Chip Erase (C7h/ 60h), 64K Block Erase (D8h) and 32K Half Block Erase (52h) commands are disable in OTP mode.

**Table 11. OTP Sector Address**

| Lock bit | Sector | Sector Size | Address Range |
|---|---|---|---|
| SPL0 | 255 | 512 byte | 0FF000h – 0FF1FFh |
| SPL1 | 254 | 512 byte | 0FE000h – 0FE1FFh |
| SPL2 | 240 | 512 byte | 0F0000h –0F01FFh |

**Note:**

**1.**The OTP sector is mapping to sector 240, 254, 255.
2.While user want to erase one of security sector, only needs to do 1 time of Sector Erase instruction(20h).

WRSR command is used to program SPL0 bit, EBL bit, SPL1 bit and SPL2 bit to '1', but these bits only can be programmed once. User can use WRDI (04h) command to exit OTP mode.

The instruction sequence is shown in Figure 37.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

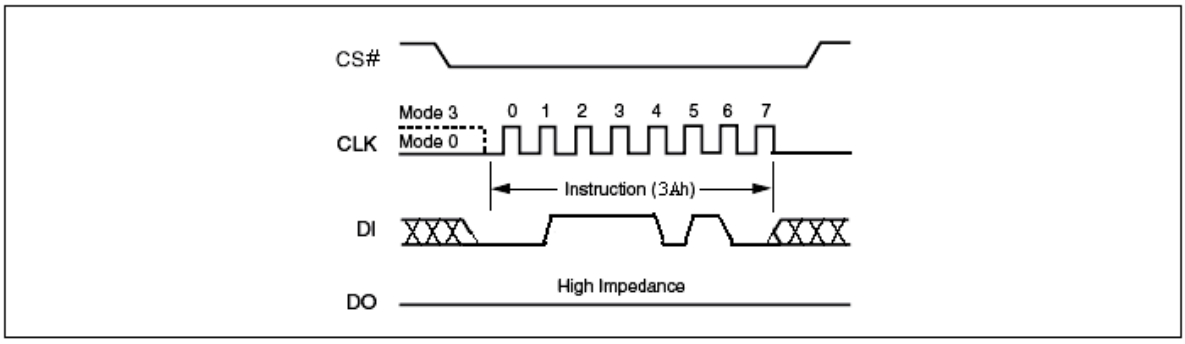
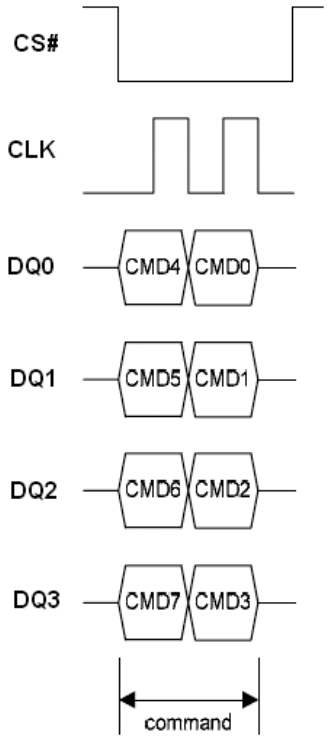



**Figure 37. Enter OTP Mode Sequence**




**Figue 37.1 Enter OTP Mode Sequence in QPI Mode**

### Read SFDP Mode and Unique ID Number (5Ah)

### Read SFDP Mode

EN25Q80C (2A) 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 Figure 38. 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.

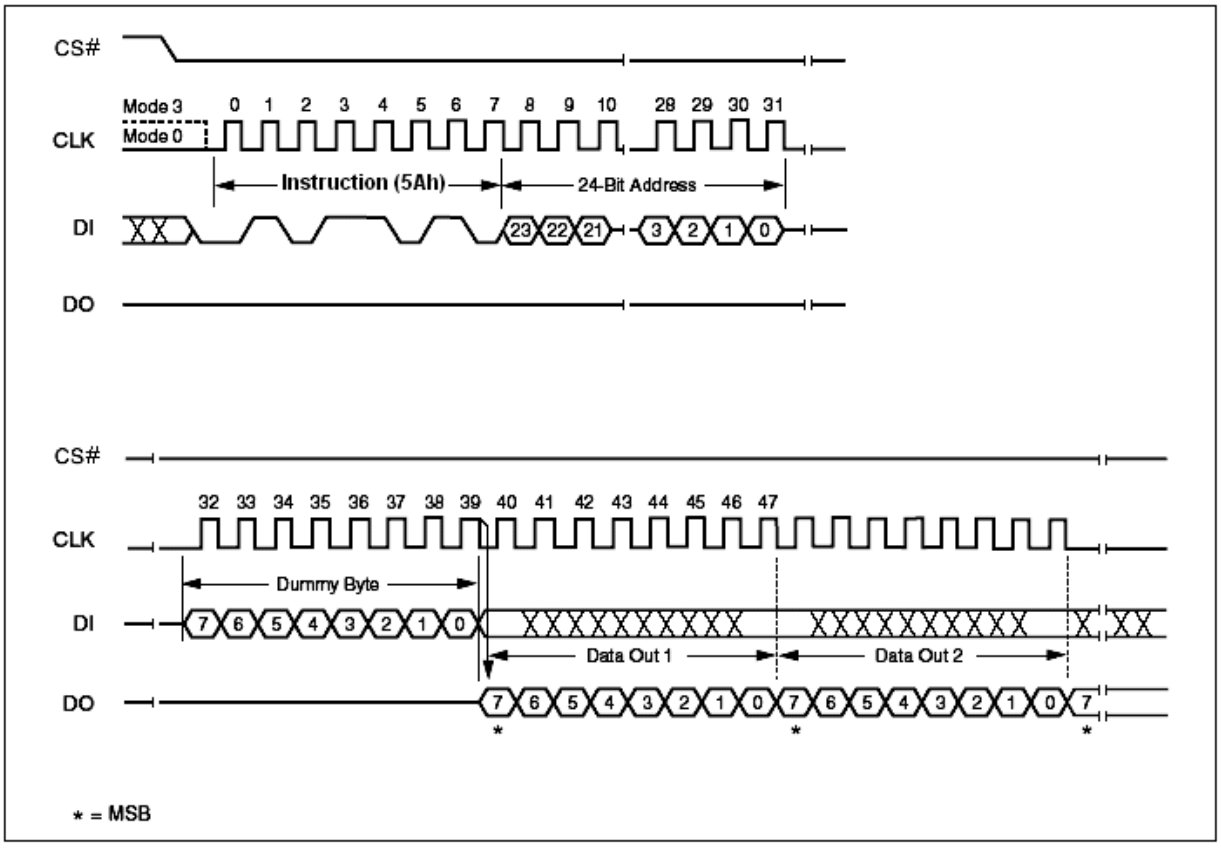



**Figure 38. Read SFDP Mode and Unique ID Number Instruction Sequence Diagram**

**Table 12. Serial Flash Discoverable Parameters (SFDP) Signature and Parameter Identification Data Value (Advanced Information)**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | 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 |

**Table 13. Parameter ID (0) (Advanced Information) 1/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | Comment |
|---|---|---|---|---|
| **Block / Sector Erase sizes**<br>Identifies the erase granularity for all Flash Components | 30h | 00<br>01 | 01b | 00 = reserved<br>01 = 4KB erase<br>10 = reserved<br>11 = 64KB erase |
| **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<br>06<br>07 | 111b | Reserved |
| **4 Kilo-Byte Erase Opcode** | 31h | 08<br>09<br>10<br>11<br>12<br>13<br>14<br>15 | 20h | 4 KB Erase Support<br>(FFh = not supported) |
| **Supports (1-1-2) Fast Read**<br>Device supports single input opcode & address and dual output data Fast Read | 32h | 16 | 1b | 0 = not supported<br>1 = supported |
| **Address Byte**<br>Number of bytes used in addressing for flash array read, write and erase. | | 17<br>18 | 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 |
| **Supports Double Data Rate (DDR) Clocking**<br>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**<br>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**<br>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**<br>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<br>25<br>26<br>27<br>28<br>29<br>30<br>31 | FFh | Reserved |

**Table 13. Parameter ID (0) (Advanced Information) 2/9**

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

**Table 13. Parameter ID (0) (Advanced Information) 3/9**

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

**Table 13. Parameter ID (0) (Advanced Information) 4/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | 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 | 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 | |
| **(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 | 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 | |

**Table 13. Parameter ID (0) (Advanced Information) 5/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | Comment |
|---|---|---|---|---|
| **Supports (2-2-2) Fast Read**<br>Device supports dual input opcode & address and dual output data Fast Read. | 40h | 00 | 0b | 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>(QPI 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 | FF FF FFh | Reserved |

**Table 13. Parameter ID (0) (Advanced Information) 6/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | Comment |
|---|---|---|---|---|
| Reserved. These bits default to all 1's | 45h : 44h | 15 : 00 | FF FFh | Reserved |
| **(2-2-2) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 46h | 16 | 00000b | 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 | Not Supported |

**Table 13. Parameter ID (0) (Advanced Information) 7/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | Comment |
|---|---|---|---|---|
| Reserved. These bits default to all 1's | 49h : 48h | 15 : 00 | FF FFh | Reserved |
| **(4-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output** | 4Ah | 16 | 00100b | 4 dummy clocks |
| | | 17 | | |
| | | 18 | | |
| | | 19 | | |
| | | 20 | | |
| **(4-4-4) Fast Read Number of Mode Bits** | | 21 | 010b | 8 mode bits |
| | | 22 | | |
| | | 23 | | |
| **(4-4-4) Fast Read Opcode** Opcode for quad input opcode/address, quad output data Fast Read. | 4Bh | 31 : 24 | EBh | Must Enter QPI Mode Firstly |

**Table 13. Parameter ID (0) (Advanced Information) 8/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | 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 | 32 KB |
| **Sector Type 2 Opcode** | 4Fh | 31 : 24 | 52h | |

**Table 13. Parameter ID (0) (Advanced Information) 9/9**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | 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**

The Read Unique ID Number instruction accesses a factory-set read-only 96-bit number that is unique to each EN25Q80C (2A) 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 "5Ah" followed by a three bytes of addresses, 0x80h, and one byte of dummy clocks. After which, the 96-bit ID is shifted out on the falling edge of CLK.

**Table 14. Unique ID Number**

| Description | Address (h) (Byte Mode) | Address (Bit) | Data | Comment |
|---|---|---|---|---|
| Unique ID Number | 80h : 8Bh | 95 : 00 | By die | |

## Power-up Timing

All functionalities and DC specifications are specified for a $V_{CC}$ ramp rate of greater than 1V per 100 ms. See Table 15 and Figure 39 for more information.

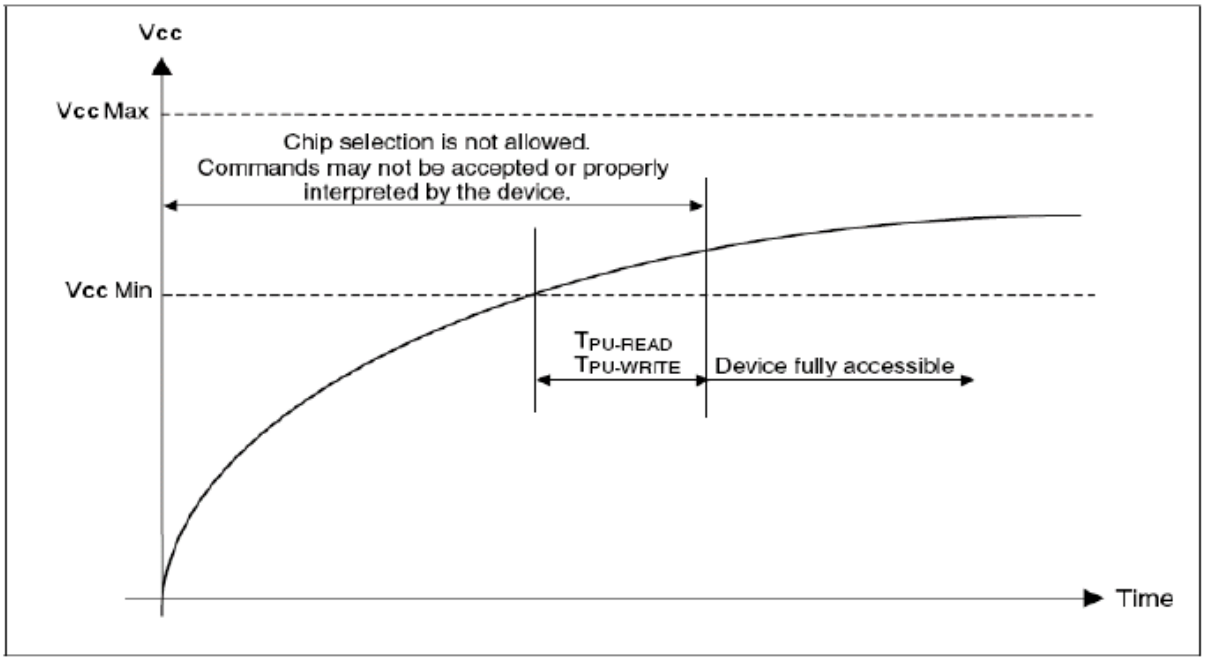



**Figure 39. Power-up Timing**

**Table 15. Power-Up Timing and Write Inhibit Threshold**

| 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).

**Table 16. 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 | 20 | µA |
| $I_{CC2}$ | Deep Power-down Current | CS# = $V_{CC}$, $V_{IN}$ = $V_{SS}$ or $V_{CC}$ | - | 1 | 20 | µA |
| $I_{CC3}$ | Operating Current (READ) | CLK = 0.1 $V_{CC}$ / 0.9 $V_{CC}$ at 104MHz, DQ = open | - | 5 | 10 | mA |
| | | CLK = 0.1 $V_{CC}$ / 0.9 $V_{CC}$ at 104MHz for Quad Output Read, DQ = open | | 14 | 18 | mA |
| $I_{CC4}$ | Operating Current (PP) | CS# = $V_{CC}$ | - | 9 | 20 | mA |
| $I_{CC5}$ | Operating Current (WRSR/ WRSR4) | CS# = $V_{CC}$ | - | | 12 | mA |
| $I_{CC6}$[1] | Operating Current (SE) | CS# = $V_{CC}$ | - | 9 | 20 | mA |
| $I_{CC7}$[1] | Operating Current (BE) | CS# = $V_{CC}$ | - | 9 | 20 | mA |
| $V_{IL}$ | Input Low Voltage | | – 0.5 | | 0.2 $V_{CC}$ | V |
| $V_{IH}$ | Input High Voltage | | 0.7$V_{CC}$ | | $V_{CC}$+0.4 | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ =100uA, $V_{CC}$= $V_{CC}$ 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.

**Table 17. AC Measurement Conditions**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $C_L$ | Load Capacitance | 30 | | pF |
| | Input Rise and Fall Times | | 5 | ns |
| | Input Pulse Voltages | 0.2$V_{CC}$ to 0.8$V_{CC}$ | | V |
| | Input Timing Reference Voltages | 0.3$V_{CC}$ to 0.7$V_{CC}$ | | V |
| | Output Timing Reference Voltages | $V_{CC}$ / 2 | | V |

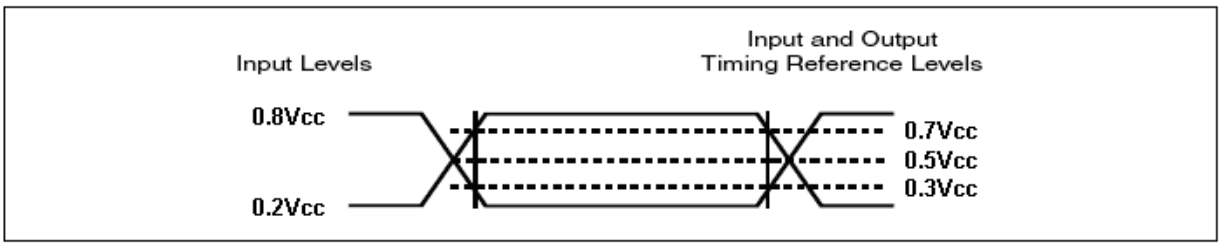



**Figure 40. AC Measurement I/O Waveform**

**Table 18. 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: FAST_READ, QPP, PP, SE, HBE, BE, DP, RES, WREN, WRDI, WRSR, RDSR, RDSR2 | 2.7V ≦ $V_{CC}$ ≦ 3.6V | D.C. | - | 104 | MHz |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | D.C | - | 86 | MHz |
| | | Serial Clock Frequency for: RDID, Dual Output Fast Read and Quad I/O Fast Read | 2.7V ≦ $V_{CC}$ ≦ 3.6V | D.C. | - | 104 | MHz |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | D.C | - | 86 | MHz |
| $f_R$ | | Serial Clock Frequency for READ | | D.C. | - | 50 | MHz |
| $t_{CH}$ [1] | | Serial Clock High Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | 3.5 | - | - | ns |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | 4 | - | - | ns |
| $t_{CL}$ [1] | | Serial Clock Low Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | 3.5 | - | - | ns |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | 4 | - | - | ns |
| $t_{CLCH}$ [2] | | Serial Clock Rise Time (Slew Rate) | | 0.1 | - | - | V / ns |
| $t_{CHCL}$ [2] | | Serial Clock Fall Time (Slew Rate) | | 0.1 | - | - | V / ns |
| $t_{SLCH}$ | $t_{CSS}$ | CS# Active Setup Time (Relative to CLK) | | 5 | - | - | ns |
| $t_{CHSH}$ | | CS# Active Hold Time (Relative to CLK) | | 5 | - | - | ns |
| $t_{SHCH}$ | | CS# Not Active Setup Time (Relative to CLK) | | 5 | - | - | ns |
| $t_{CHSL}$ | | CS# Not Active Hold Time (Relative to CLK) | | 5 | - | - | ns |
| $t_{SHSL}$ | $t_{CSH}$ | CS# High Time | | 30 | - | - | ns |
| $t_{SHQZ}$ [2] | $t_{DIS}$ | Output Disable Time | | - | - | 6 | ns |
| $t_{CLQX}$ | $t_{HO}$ | Output Hold Time | | 0 | - | - | ns |
| $t_{DVCH}$ | $t_{DSU}$ | Data In Setup Time | | 2 | - | - | ns |
| $t_{CHDX}$ | $t_{DH}$ | Data In Hold Time | | 5 | - | - | 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# Low to High-Z Output | | | | 6 | ns |
| $t_{HHQX}$ [2] | $t_{LZ}$ | HOLD# High to Low-Z Output | | | | 6 | ns |
| $t_{CLQV}$ | $t_V$ | Output Valid from CLK for 30 pF | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | - | 8 | ns |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | - | 10 | ns |
| | | Output Valid from CLK for 15 pF | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | - | 6 | ns |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | - | 8 | 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 | | - | - | 3 | µs |
| $t_{RES2}$ [2] | | CS# High to Standby Mode with Electronic Signature read | | - | - | 1.8 | µs |
| $t_W$ | | Write Status Register Cycle Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | 4 | 30 | ms |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | 10 | 50 | ms |
| $t_{PP}$ | | Page Programming Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | 0.5 | 3 | ms |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | 0.9 | 5 | ms |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_{SE}$ | | Sector Erase Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | 40 | 300 | ms |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | 150 | 1000 | ms |
| $t_{HBE}$ | | 32KB Block Erase Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | 0.12 | 1 | s |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | 0.25 | 2 | s |
| $t_{BE}$ | | 64KB Block Erase Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | 0.15 | 2 | s |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | 0.4 | 3 | s |
| $t_{CE}$ | | Chip Erase Time | 2.7V ≦ $V_{CC}$ ≦ 3.6V | - | 4 | 12 | s |
| | | | 2.3V ≦ $V_{CC}$ < 2.7V | - | 7 | 20 | 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.

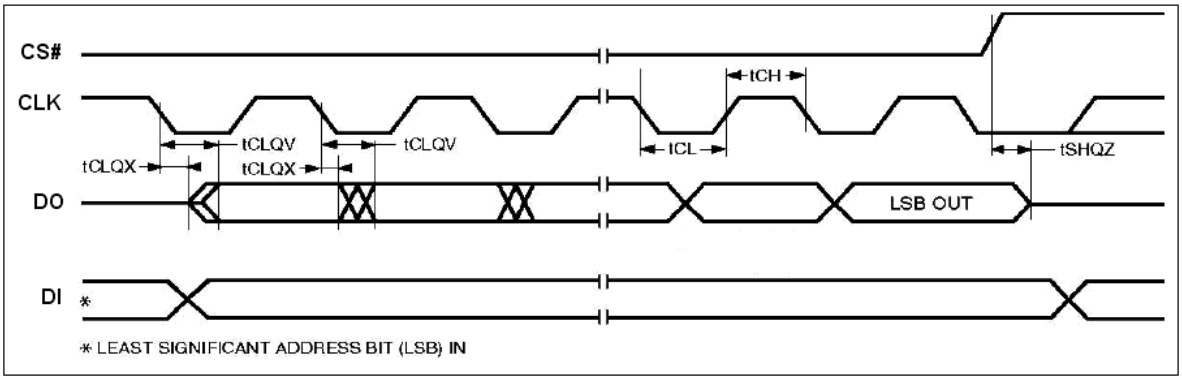



**Figure 41. Serial Output Timing**

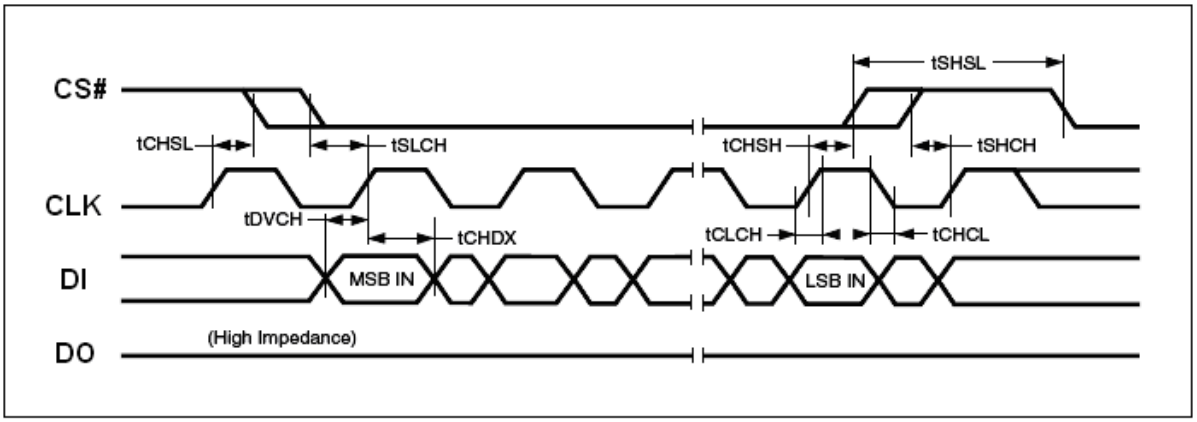



**Figure 42. Input Timing**

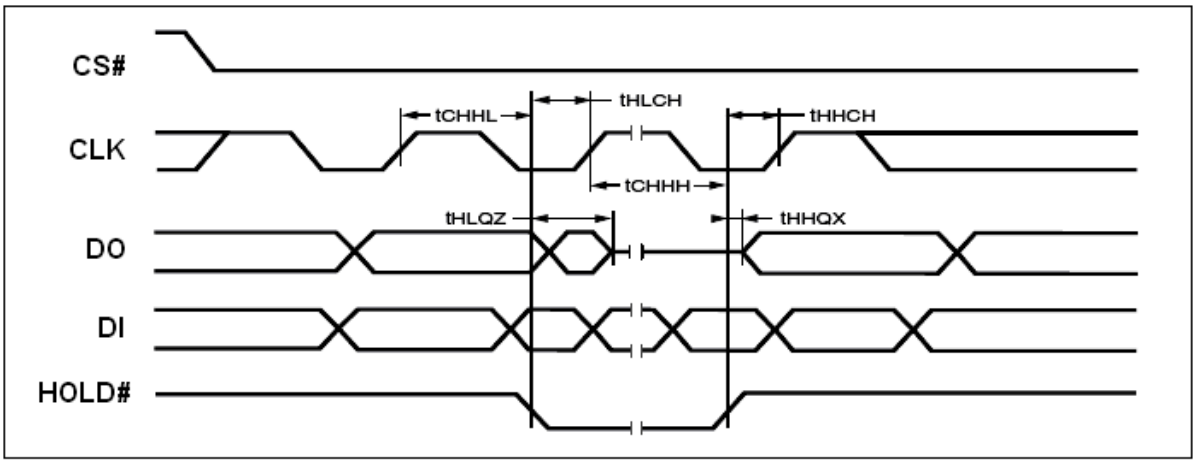



**Figure 43. Hold Timing**

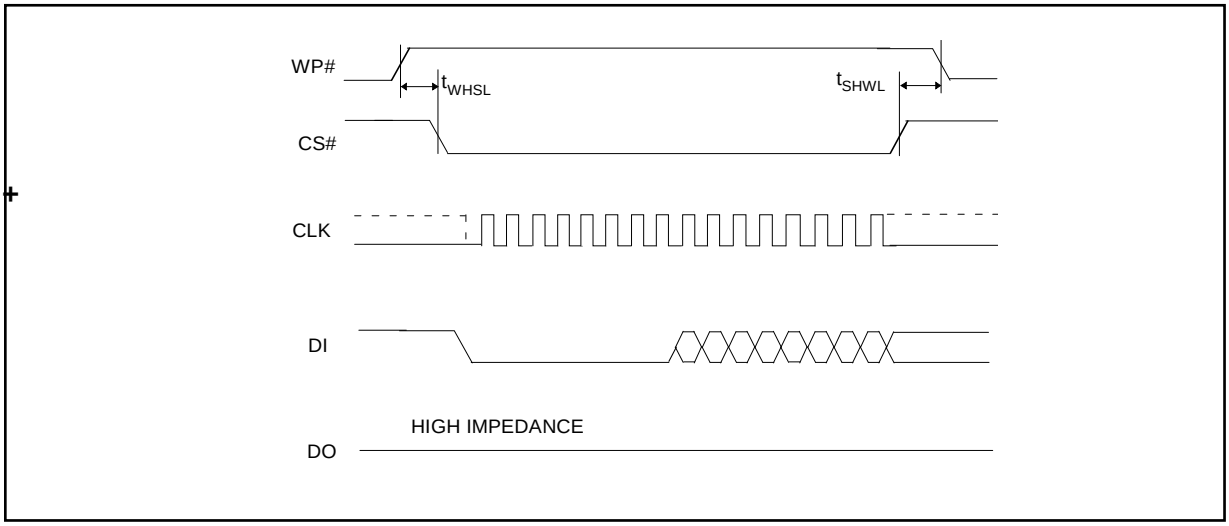



**Figure 44: Write Protect setup and hold timing during WRSR when SRP = 1**

## 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.

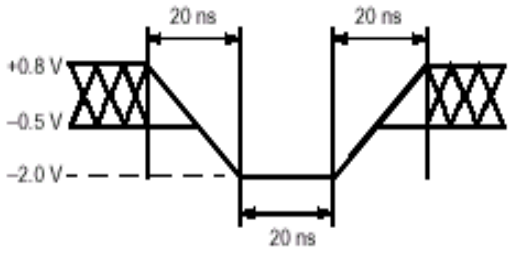
| Parameter | Value | Unit |
|---|---|---|
| Storage Temperature | -65 to +150 | °C |
| Output Short Circuit Current[1] | 200 | mA |
| Input and Output Voltage (with respect to ground) [2] | -0.5 to +4.0 | V |
| $V_{CC}$ | -0.5 to +4.0 | V |

Notes:
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 –0.5 V. During voltage transitions, inputs may undershoot $V_{SS}$ to –1.0V for periods of up to 50ns and to –2.0 V for periods of up to 20ns. See figure below. Maximum DC voltage on output and I/O pins is $V_{CC}$ + 0.5 V. During voltage transitions, outputs may overshoot to $V_{CC}$ + 1.5 V for periods up to 20ns. See figure below.

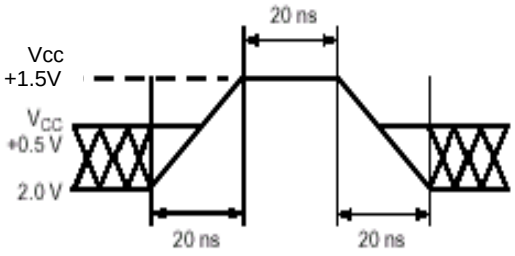
## RECOMMENDED OPERATING RANGES

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

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




Maximum Negative Overshoot Waveform




Maximum Positive Overshoot Waveform

**Table 19. CAPACITANCE**

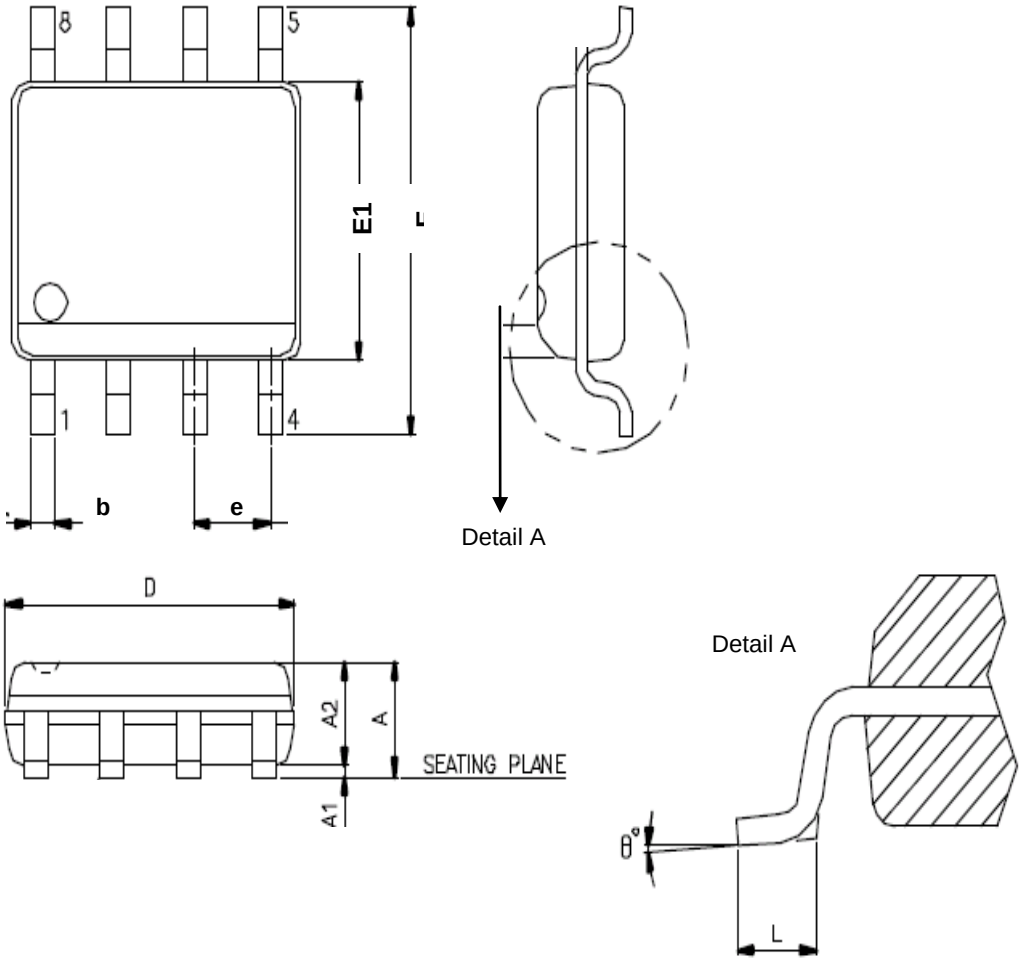
($V_{CC}$ = 2.3-3.6V)

| Parameter Symbol | Parameter Description | Test Setup | 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

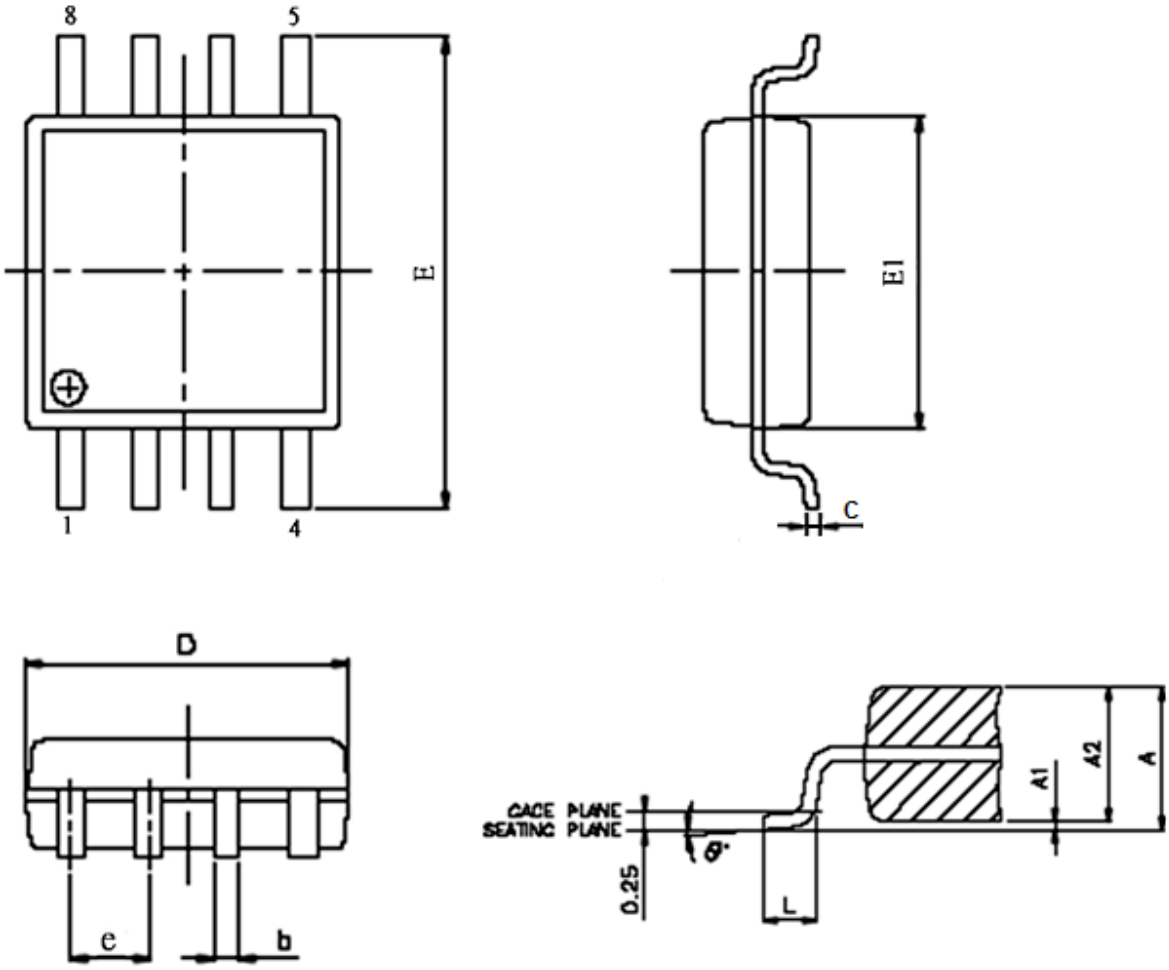
**Figure 45. SOP 150mil**




| SYMBOL | DIMENSION IN MM | | |
|---|---|---|---|
| | MIN. | NOR | MAX |
| A | 1.35 | - - - | 1.75 |
| A1 | 0.10 | - - - | 0.25 |
| A2 | - - - | - - - | 1.50 |
| D | 4.80 | - - - | 5.00 |
| E | 5.80 | - - - | 6.20 |
| E1 | 3.80 | - - - | 4.00 |
| e | - - - | 1.27 | - - - |
| b | 0.33 | - - - | 0.51 |
| L | 0.4 | - - - | 1.27 |
| θ | $0^0$ | - - - | $8^0$ |

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

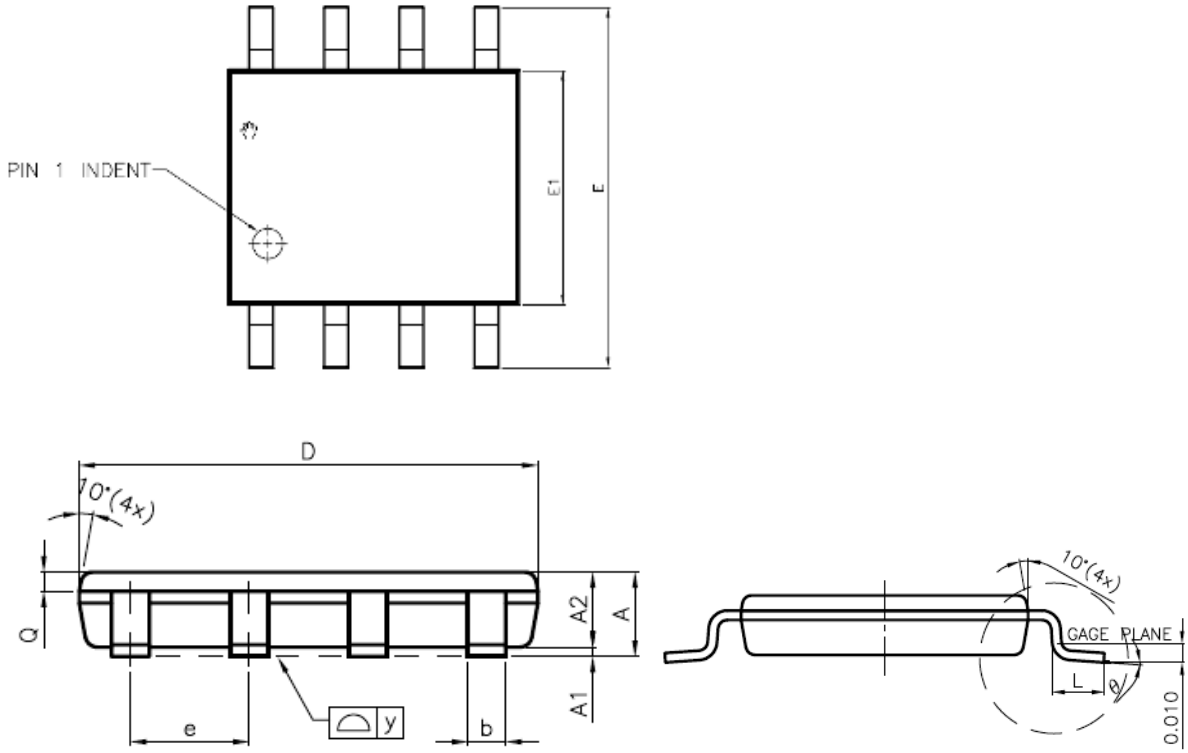
**Figure 46. SOP 200 mil ( official name = 208 mil )**




| 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.

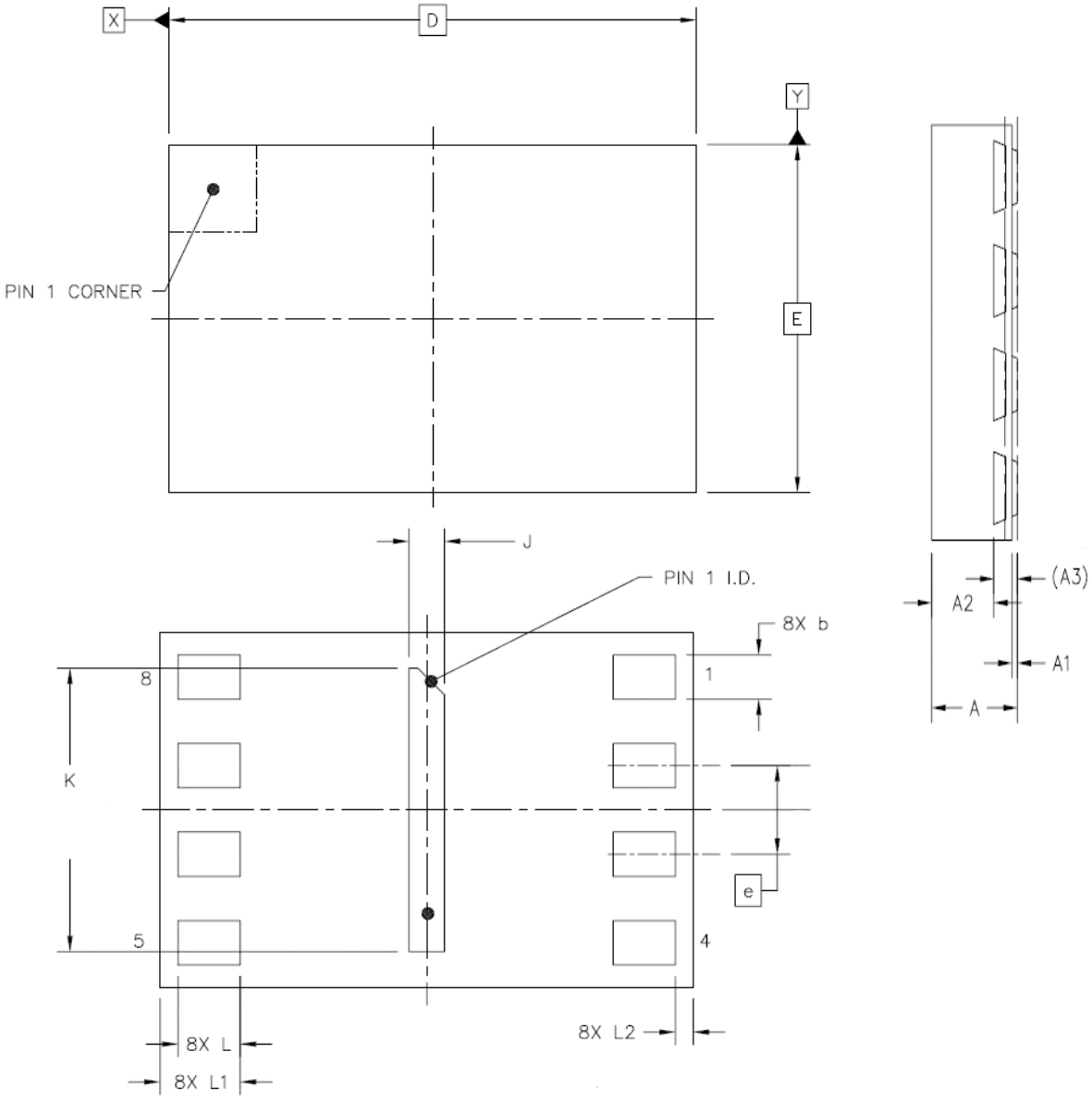
**Figure 47. VSOP 8 ( 150 mil )**




| SYMBOL | DIMENSION IN MM | | |
|---|---|---|---|
| | MIN. | NOR | MAX |
| A | -- | -- | 0.90 |
| A1 | 0.05 | 0.10 | 0.15 |
| A2 | 0.65 | 0.70 | 0.75 |
| D | 4.80 | 4.90 | 5.00 |
| E | 5.80 | 6.00 | 6.20 |
| E1 | 3.80 | 3.90 | 4.00 |
| e | - - - | 1.27 | - - - |
| b | 0.33 | 0.41 | 0.51 |
| L | 0.40 | 0.71 | 1.27 |
| θ | 0 | -- | 10 |

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

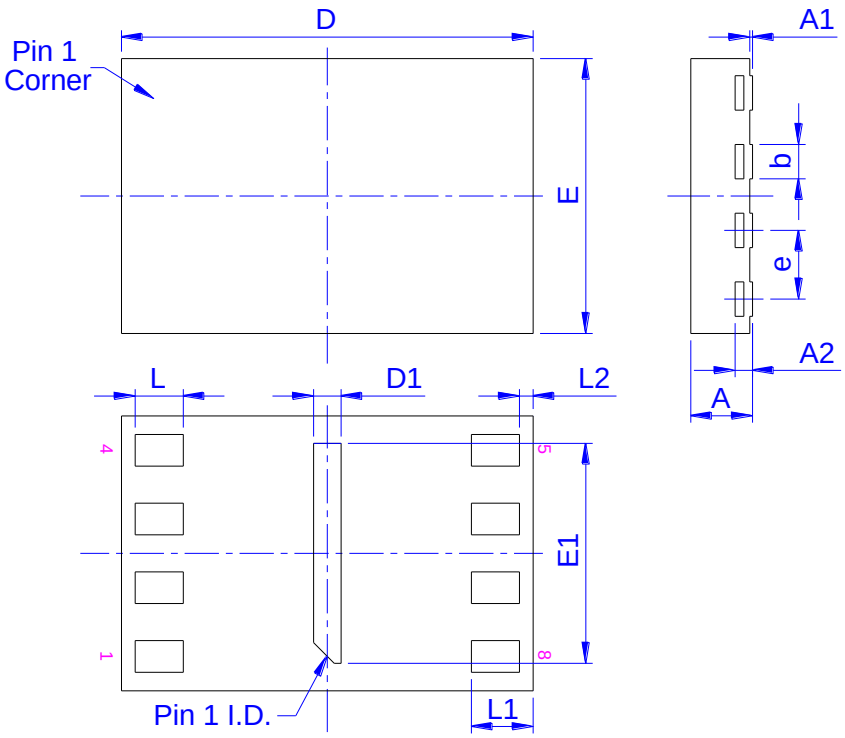
**Figure 48. USON8 (2x3x0.55mm)**




| SYMBOL | DIMENSION IN MM | | |
|---|---|---|---|
| | MIN. | NOR | MAX |
| A | 0.50 | 0.55 | 0.60 |
| A1 | 0.00 | 0.035 | 0.05 |
| A2 | - - - | 0.40 | 0.425 |
| A3 | 0.152 REF | | |
| D | 2.90 | 3.00 | 3.10 |
| E | 1.90 | 2.00 | 2.10 |
| J | 0.10 | 0.20 | 0.30 |
| K | 1.50 | 1.60 | 1.70 |
| e | 0.5 BSC | | |
| b | 0.20 | 0.25 | 0.30 |
| L | 0.30 | - - - | - - - |
| L1 | 0.40 | 0.45 | 0.50 |
| L2 | - - - | - - - | 0.15 |

Notice: This package can't contact to metal trace or pad on board due to expose metal pad underneath the package.

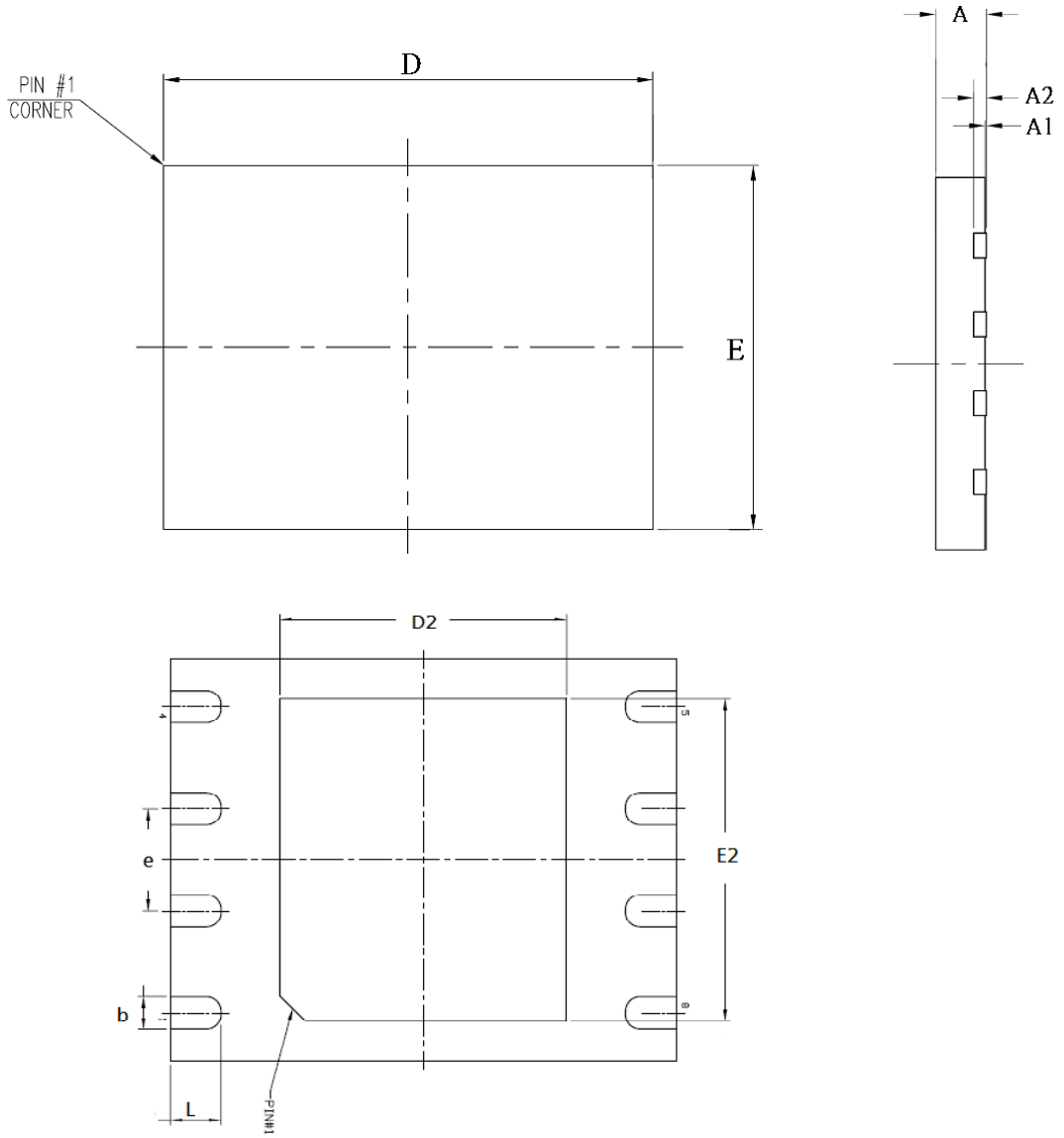
**Figure 49. USON8 (2x3x0.45mm)**




| Symbol | Dimension in mm | | |
|---|---|---|---|
| | Min. | Norm. | Max. |
| **A** | 0.40 | 0.45 | 0.50 |
| **A1** | 0.00 | - | 0.05 |
| **A2** | | 0.152 REF | |
| **b** | 0.20 | 0.25 | 0.30 |
| **D** | 2.90 | 3.00 | 3.10 |
| **D1** | 0.10 | 0.20 | 0.30 |
| **E** | 1.90 | 2.00 | 2.10 |
| **E1** | 1.50 | 1.60 | 1.70 |
| **e** | | 0.50 BSC | |
| **L** | 0.30 | - | - |
| **L1** | 0.40 | 0.45 | 0.50 |
| **L2** | - | - | 0.15 |

Controlling dimension: millimeter (Revision date : Dec 22 2016)

**Figure 50. VDFN 8 ( 5x6 mm )**

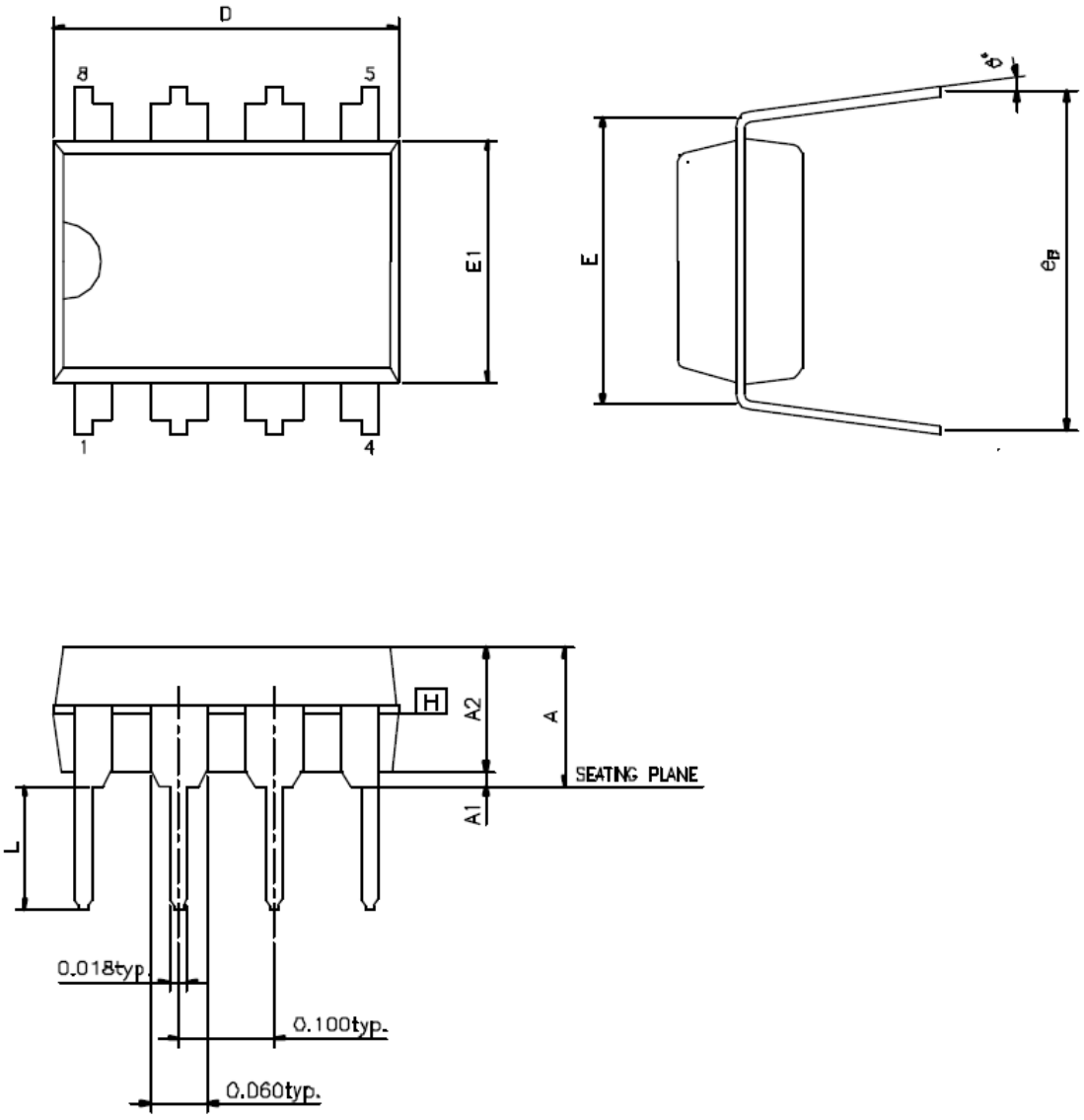



**Controlling dimensions are in millimeters (mm).**

| SYMBOL | DIMENSION IN MM | | |
|---|---|---|---|
| | MIN. | NOR | MAX |
| A | 0.70 | 0.75 | 0.80 |
| A1 | 0.00 | 0.02 | 0.04 |
| A2 | - - - | 0.20 | - - - |
| D | 5.90 | 6.00 | 6.10 |
| E | 4.90 | 5.00 | 5.10 |
| D2 | 3.30 | 3.40 | 3.50 |
| E2 | 3.90 | 4.00 | 4.10 |
| e | - - - | 1.27 | - - - |
| b | 0.35 | 0.40 | 0.45 |
| L | 0.55 | 0.60 | 0.65 |

**Note : 1. Coplanarity: 0.1 mm**

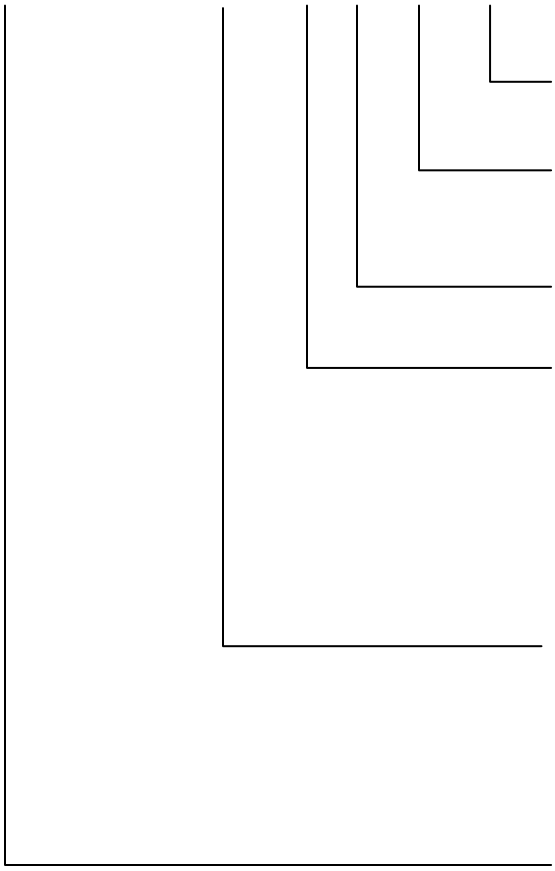
**Figure 51. PDIP8**




| SYMBOL | DIMENSION IN INCH | | |
|---|---|---|---|
| | MIN. | NOR | MAX |
| A | - - - | - - - | 0.210 |
| A1 | 0.015 | - - - | - - - |
| A2 | 0.125 | 0.130 | 0.135 |
| D | 0.355 | 0.365 | 0.400 |
| E | 0.300 | 0.310 | 0.320 |
| E1 | 0.245 | 0.250 | 0.255 |
| L | 0.115 | 0.130 | 0.150 |
| $e_B$ | 0.310 | 0.350 | 0.375 |
| $\Theta^0$ | 0 | 7 | 15 |

## ORDERING INFORMATION

EN25Q80C - 104 H I P 2A



**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
RB= 8-pin 150mil VSOP
X = 8-pin USON (2x3x0.55 mm)
XF= 8-pin USON (2x3x0.45 mm)
W = 8-pin VDFN (5x6mm)
Q = 8-pin PDIP

**SPEED**
104 = 104 MHz

**BASE PART NUMBER**
EN = Eon Silicon Solution Inc.
25Q = 3V Serial Flash with 4KB Uniform-Sector, Dual and Quad I/O
8 = 8 Megabit (1024K x 8)
A = version identifier

## Revisions List

| Revision No | Description | Date |
|---|---|---|
| Preliminary 0.1 | Initial Release | 2017/01/23 |
| Preliminary 0.2 | 1. Modify OTP size<br>2. Add USON 2x3x0.45mm Package<br>3. Delete 1KB/2KB Sector function<br>4. Delete Erase Fail Flag/Program Erase Flag<br>5. Modify Icc3 (33MHz) DC Characteristics | 2017/03/03 |
| Preliminary 0.3 | Add differentiation code into product ID | 2017/05/05 |
| Preliminary 0.4 | 1. Modify Low power consumptiont current and High performance program/erase speed<br>2. 38h= 01000b to 00100b<br>3. update Power-up Timing figure<br>4. Delete ICC3 for 33MHz and add a note for erase current<br>5. Modify the specification of tCH, tCL,tW, tPP,tSE,tHBE,tBE,tCE<br>6. Update USON2x3x0.45 figure | 2017/06/26 |
| Preliminary 0.5 | Modify the specification of tW, tHBE, tBE, tCE. | 2017/08/03 |
| 1.0 | 1. Delete "Preliminary"<br>2. Modify the specification of Parameter ID table. | 2017/11/14 |
| 1.1 | Correct the typo of Uniform Block Sector Architecture table. | 2018/09/14 |
| 1.2 | Modify the packing dimension of SOP 200 mil (official name = 208 mil) | 2018/10/05 |
| 1.3 | 1. Modify voltage from 2.7V~3.6V to 2.3V~3.6V<br>2. Add some AC specifications for VCC under 2.7V<br>3. Revise PACKAGE of USON (8L 2x3x0.55mm) | 2019/03/04 |
| 1.4 | Delete Plastic Packages Temperature | 2020/10/15 |