

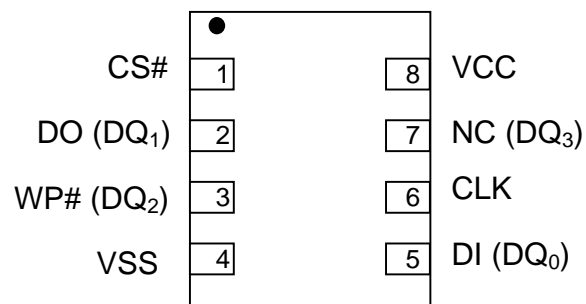
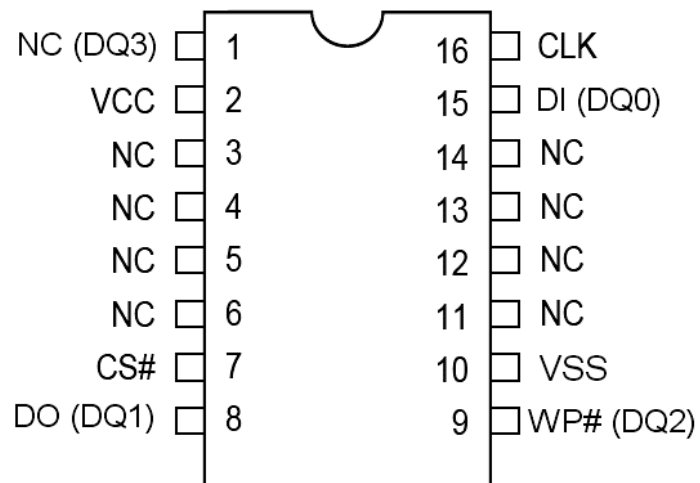
**EN25Q128****128 Megabit Serial Flash Memory with 4Kbyte Uniform Sector****FEATURES**

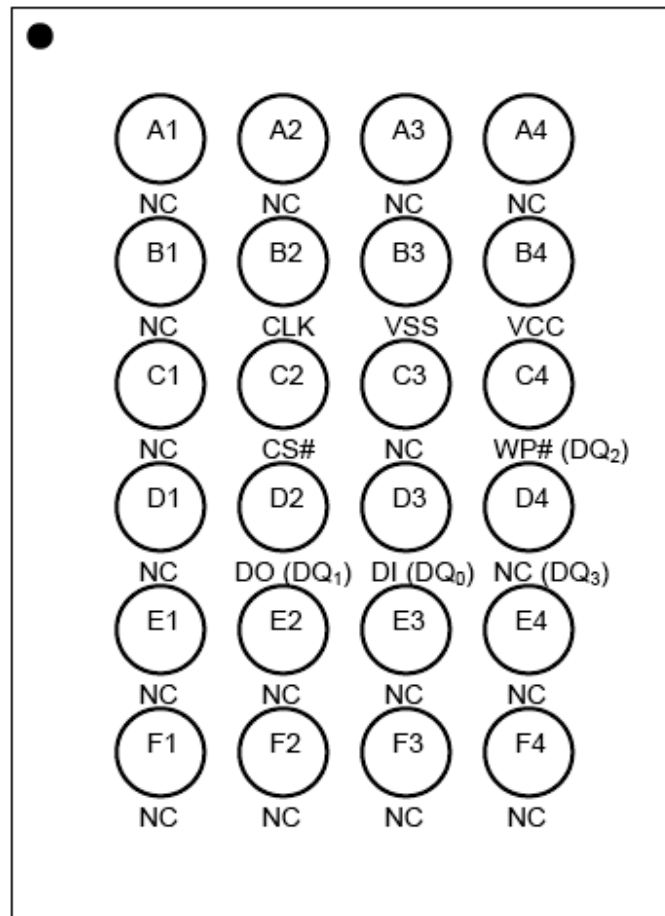
- Single power supply operation
  - Full voltage range: 2.7-3.6 volt
- Serial Interface Architecture
  - SPI Compatible: Mode 0 and Mode 3
- 128 M-bit Serial Flash
  - 128 M-bit/16,384 K-byte/65,536 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO, WP#
  - Dual SPI: CLK, CS#, DQ<sub>0</sub>, DQ<sub>1</sub>, WP#
  - Quad SPI: CLK, CS#, DQ<sub>0</sub>, DQ<sub>1</sub>, DQ<sub>2</sub>, DQ<sub>3</sub>
- High performance
  - 104MHz clock rate for Standard SPI
  - 80MHz clock rate for two data bits
  - 50MHz clock rate for four data bits
- Low power consumption
  - 12 mA typical active current
  - 1  $\mu$ A typical power down current
- Uniform Sector Architecture:
  - 4096 sectors of 4-Kbyte
  - 256 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: 0.8ms typical
  - Sector erase time: 50ms typical
  - Block erase time 200ms typical
  - Chip erase time: 45 seconds typical
- Lockable 512 byte OTP security sector
- Minimum 100K endurance cycle
- Package Options
  - 8 contact VDFN (5x6mm)
  - 8 contact VDFN (6x8mm)
  - 16 pins SOP 300mil body width
  - 24 balls TFBGA (6x8mm)
  - All Pb-free packages are RoHS compliant
- Industrial and Automotive temperature Range

**GENERAL DESCRIPTION**

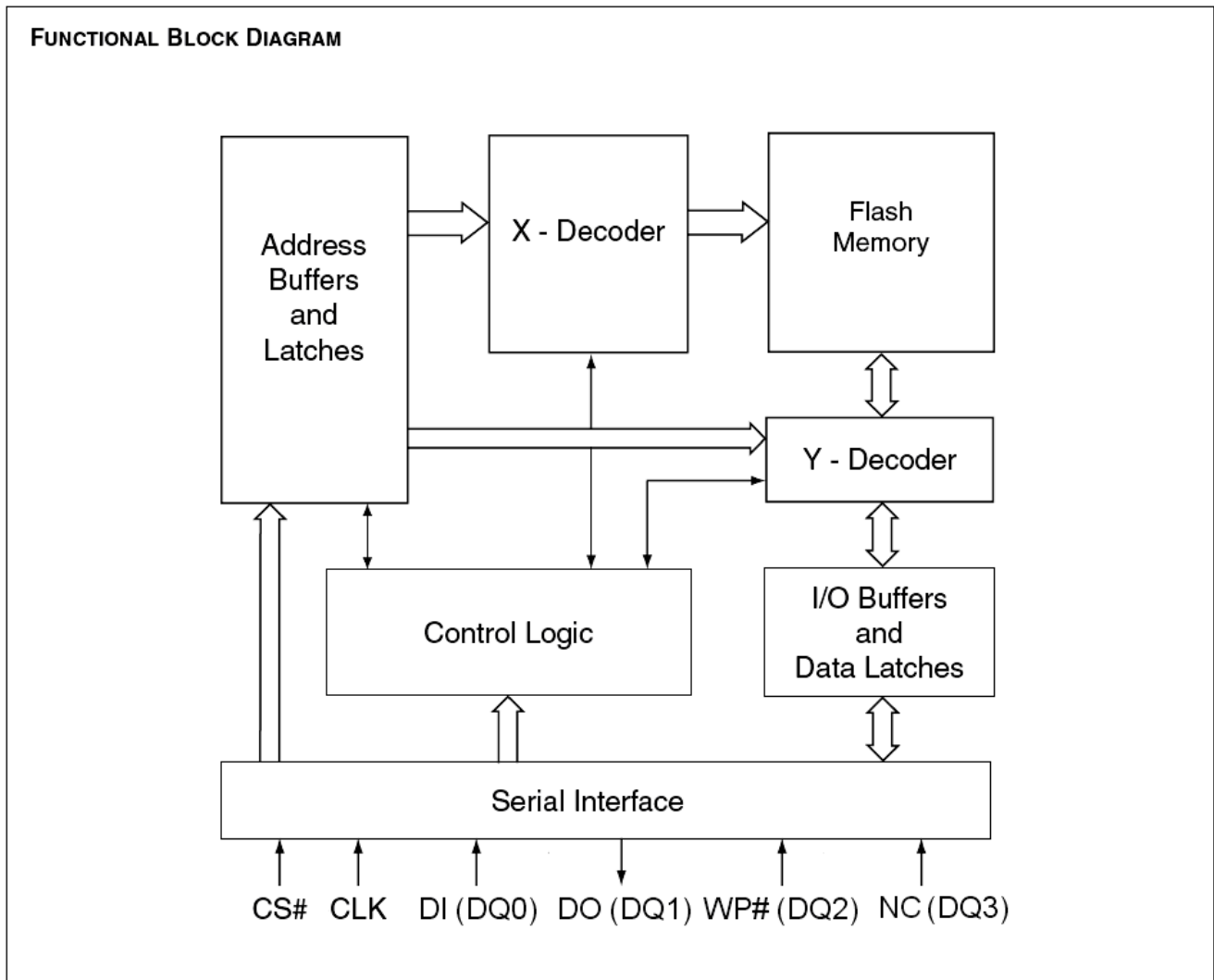
The EN25Q128 is a 128 Megabit (16,384K-byte) Serial Flash memory, with advanced write protection mechanisms. The EN25Q128 supports the standard Serial Peripheral Interface (SPI), and a high performance Dual output as well as Dual/Quad I/O using SPI pins: Serial Clock, Chip Select, Serial DQ<sub>0</sub>(DI), DQ<sub>1</sub>(DO), DQ<sub>2</sub>(WP#) and DQ<sub>3</sub>(NC). SPI clock frequencies of up to 80MHz are supported allowing equivalent clock rates of 160MHz (80MHz x 2) for Dual Output when using the Dual Output Fast Read instructions, and SPI clock frequencies of up to 50MHz are supported allowing equivalent clock rates of 200MHz (50MHz x 4) for Quad Output when using the Quad Output Fast Read instructions. The memory can be programmed 1 to 256 bytes at a time, using the Page Program instruction.

The EN25Q128 is designed to allow either single Sector/Block at a time or full chip erase operation. The EN25Q128 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 VDFN**

**16 - LEAD SOP**

**Top View, Balls Facing Down**


24 - Ball TFBGA

**Figure 2. BLOCK DIAGRAM**

**Note:**

1. DQ<sub>0</sub> and DQ<sub>1</sub> are used for Dual and Quad instructions.
2. DQ<sub>0</sub> ~ DQ<sub>3</sub> are used for Quad instructions.



Table 1. Pin Names

Symbol	Pin Name
CLK	Serial Clock Input
DI (DQ <sub>0</sub> )	Serial Data Input (Data Input Output 0) <sup>*1</sup>
DO (DQ <sub>1</sub> )	Serial Data Output (Data Input Output 1) <sup>*1</sup>
CS#	Chip Enable
WP# (DQ <sub>2</sub> )	Write Protect (Data Input Output 2) <sup>*2</sup>
NC(DQ <sub>3</sub> )	Not Connect (Data Input Output 3) <sup>*2</sup>
Vcc	Supply Voltage (2.7-3.6V)
Vss	Ground
NC	No Connect

Note:

1. DQ<sub>0</sub> and DQ<sub>1</sub> are used for Dual and Quad instructions.
2. DQ<sub>0</sub> ~ DQ<sub>3</sub> are used for Quad instructions.

## SIGNAL DESCRIPTION

### Serial Data Input, Output and IOs (DI, DO and DQ<sub>0</sub>, DQ<sub>1</sub>, DQ<sub>2</sub>, DQ<sub>3</sub>)

The EN25Q128 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<sub>0</sub>, DQ<sub>1</sub>, DQ<sub>2</sub> and DQ<sub>3</sub>) pins are at high impedance. When deselected, the devices power consumption will be at standby levels unless an internal erase, program or status register cycle is in progress. When CS# is brought low the device will be selected, power consumption will increase to active levels and instructions can be written to and data read from the device. After power-up, CS# must transition from high to low before a new instruction will be accepted.

### Write Protect (WP#)

The Write Protect (WP#) pin can be used to prevent the Status Register from being written. Used in conjunction with the Status Register's Block Protect (BP0, BP1, BP2 and BP3) bits and Status Register Protect (SRP) bits, a portion or the entire memory array can be hardware protected. The WP# function is only available for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO (DQ<sub>2</sub>) for Quad I/O operation.

**MEMORY ORGANIZATION**

The memory is organized as:

- 16,777,216 bytes
- Uniform Sector Architecture
  - 256 blocks of 64-Kbyte
  - 4,096 sectors of 4-Kbyte
  - 65,536 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 2. Uniform Block Sector Architecture ( 1/4 )

Block	Sector	Address range	
255	4095	FFF000h	FFFFFFh
	⋮	⋮	⋮
254	4080	FF0000h	FF0FFFh
	4079	FEF000h	FEFFFFh
253	4064	FE0000h	FE0FFFh
	4063	FDF000h	FDFFFFh
252	4048	FD0000h	FD0FFFh
	⋮	⋮	⋮
242	3887	F2F000h	F2FFFFh
	⋮	⋮	⋮
241	3872	F20000h	F20FFFh
	3871	F1F000h	F1FFFFh
240	3856	F10000h	F10FFFh
	3855	F0F000h	F0FFFFh
239	3840	F00000h	F00FFFh
	⋮	⋮	⋮

Block	Sector	Address range	
239	3839	EFF000h	EFFFFFh
	⋮	⋮	⋮
238	3824	EF0000h	EF0FFFh
	3823	EEF000h	EEFFFFh
237	3808	EE0000h	EE0FFFh
	3807	EDF000h	EDFFFFh
236	3792	ED0000h	ED0FFFh
	⋮	⋮	⋮
226	3631	E2F000h	E2FFFFh
	⋮	⋮	⋮
225	3616	E20000h	E20FFFh
	3615	E1F000h	E1FFFFh
224	3600	E10000h	E10FFFh
	3599	E0F000h	E0FFFFh
223	3584	E00000h	E00FFFh
	⋮	⋮	⋮

Block	Sector	Address range	
223	3583	DFF000h	DFFFFFFh
	⋮	⋮	⋮
222	3568	DF0000h	DF0FFFh
	3567	DEF000h	DEFFFFh
221	3552	DE0000h	DE0FFFh
	3551	DDF000h	DDFFFFh
210	3536	DD0000h	DD0FFFh
	⋮	⋮	⋮
210	3375	D2F000h	D2FFFFh
	⋮	⋮	⋮
209	3360	D20000h	D20FFFh
	3359	D1F000h	D1FFFFh
208	3344	D10000h	D10FFFh
	3343	D0F000h	D0FFFFh
207	3328	D00000h	D00FFFh
	⋮	⋮	⋮

Block	Sector	Address range	
207	3327	CFF000h	CFFFFFFh
	⋮	⋮	⋮
206	3312	CF0000h	CF0FFFh
	3311	CEF000h	CEFFFFh
205	3296	CE0000h	CE0FFFh
	3295	CDF000h	CDFFFFh
194	3280	CD0000h	CD0FFFh
	⋮	⋮	⋮
194	3119	C2F000h	C2FFFFh
	⋮	⋮	⋮
193	3014	C20000h	C20FFFh
	3103	C1F000h	C1FFFFh
192	3088	C10000h	C10FFFh
	3087	C0F000h	C0FFFFh
191	3072	C00000h	C00FFFh
	⋮	⋮	⋮

**Table 2. Uniform Block Sector Architecture ( 2/4 )**

Block	Sector	Address range	
191	3071	BFF000h	BFFFFFh
	⋮	⋮	⋮
190	3056	BF0000h	BF0FFFh
	3055	BEF000h	BEFFFFh
189	3040	BE0000h	BE0FFFh
	3039	BDF000h	BDFFFFh
⋮	⋮	⋮	⋮
	3024	BD0000h	BD0FFFh
178	2863	B2F000h	B2FFFFh
	⋮	⋮	⋮
177	2848	B20000h	B20FFFh
	2847	B1F000h	B1FFFFh
176	2832	B10000h	B10FFFh
	2831	B0F000h	B0FFFFh
⋮	⋮	⋮	⋮
	2816	B00000h	B00FFFh

Block	Sector	Address range	
175	2815	AFF000h	AFFFFFh
	⋮	⋮	⋮
174	2800	AF0000h	AF0FFFh
	2799	AEF000h	AEFFFFh
173	2784	AE0000h	AE0FFFh
	2783	ADF000h	ADFFFFh
⋮	⋮	⋮	⋮
	2768	AD0000h	AD0FFFh
162	2607	A2F000h	A2FFFFh
	⋮	⋮	⋮
161	2592	A20000h	A20FFFh
	2591	A1F000h	A1FFFFh
160	2576	A10000h	A10FFFh
	2575	A0F000h	A0FFFFh
⋮	⋮	⋮	⋮
	2560	A00000h	A00FFFh

Block	Sector	Address range	
159	2559	9FF000h	9FFFFFFh
	⋮	⋮	⋮
158	2544	9F0000h	9F0FFFh
	2543	9EF000h	9EFFFFh
157	2528	9E0000h	9E0FFFh
	2527	9DF000h	9DFFFFh
⋮	⋮	⋮	⋮
	2512	9D0000h	9D0FFFh
146	2351	92F000h	92FFFFh
	⋮	⋮	⋮
145	2336	920000h	920FFFh
	2335	91F000h	91FFFFh
144	2320	910000h	910FFFh
	2319	90F000h	90FFFFh
⋮	⋮	⋮	⋮
	2304	900000h	900FFFh

Block	Sector	Address range	
143	2303	8FF000h	8FFFFFFh
	⋮	⋮	⋮
142	2288	8F0000h	8F0FFFh
	2287	8EF000h	8EFFFFh
141	2272	8E0000h	8E0FFFh
	2271	8DF000h	8DFFFFh
⋮	⋮	⋮	⋮
	2256	8D0000h	8D0FFFh
130	2095	82F000h	82FFFFh
	⋮	⋮	⋮
129	2080	820000h	820FFFh
	2079	81F000h	81FFFFh
128	2064	810000h	810FFFh
	2063	80F000h	80FFFFh
⋮	⋮	⋮	⋮
	2048	800000h	800FFFh





Table 2. Uniform Block Sector Architecture ( 3/4 )

Block	Sector	Address range	
127	2047	7FF000h	7FFFFFFh
	⋮	⋮	⋮
126	2032	7F0000h	7F0FFFh
	2031	7EF000h	7EFFFFh
125	2016	7E0000h	7E0FFFh
	2015	7DF000h	7DFFFFh
125	2000	7D0000h	7D0FFFh
	⋮	⋮	⋮
114	1839	72F000h	72FFFFFFh
	1824	720000h	720FFFh
113	1823	71F000h	71FFFFFFh
	1808	710000h	710FFFh
112	1807	70F000h	70FFFFFFh
	1792	700000h	700FFFh

Block	Sector	Address range	
111	1791	6FF000h	6FFFFFFh
	1776	6F0000h	6F0FFFh
110	1775	6EF000h	6EFFFFh
	1760	6E0000h	6E0FFFh
109	1759	6DF000h	6DFFFFh
	1744	6D0000h	6D0FFFh
98	1583	62F000h	62FFFFFFh
	1568	620000h	620FFFh
97	1567	61F000h	61FFFFFFh
	1552	610000h	610FFFh
96	1551	60F000h	60FFFFFFh
	1536	600000h	600FFFh

Block	Sector	Address range	
95	1535	5FF000h	5FFFFFFh
	1520	5F0000h	5F0FFFh
94	1519	5EF000h	5EFFFFh
	1504	5E0000h	5E0FFFh
93	1503	5DF000h	5DFFFFh
	1488	5D0000h	5D0FFFh
82	1327	52F000h	52FFFFFFh
	1312	520000h	520FFFh
81	1311	51F000h	51FFFFFFh
	1296	510000h	510FFFh
80	1295	50F000h	50FFFFFFh
	1280	500000h	500FFFh

Block	Sector	Address range	
79	1279	4FF000h	4FFFFFFh
	1264	4F0000h	4F0FFFh
78	1263	4EF000h	4EFFFFh
	1248	4E0000h	4E0FFFh
77	1247	4DF000h	4DFFFFh
	1232	4D0000h	4D0FFFh
66	1071	42F000h	42FFFFFFh
	1056	420000h	420FFFh
65	1055	41F000h	41FFFFFFh
	1040	410000h	410FFFh
64	1039	40F000h	40FFFFFFh
	1024	400000h	400FFFh



Table 2. Uniform Block Sector Architecture ( 4/4 )

Block	Sector	Address range	
63	1023	3FF000h	3FFFFFFh
	⋮	⋮	⋮
62	1008	3F0000h	3F0FFFh
	1007	3EF000h	3EFFFFh
61	992	3E0000h	3E0FFFh
	991	3DF000h	3DFFFFh
⋮	⋮	⋮	⋮
	976	3D0000h	3D0FFFh
50	815	32F000h	32FFFFFFh
	⋮	⋮	⋮
49	800	320000h	320FFFh
	799	31F000h	31FFFFFFh
48	784	310000h	310FFFh
	783	30F000h	30FFFFFFh
⋮	⋮	⋮	⋮
	768	300000h	300FFFh

Block	Sector	Address range	
47	767	2FF000h	2FFFFFFh
	⋮	⋮	⋮
46	752	2F0000h	2F0FFFh
	751	2EF000h	2EFFFFh
45	736	2E0000h	2E0FFFh
	735	2DF000h	2DFFFFh
⋮	⋮	⋮	⋮
	720	2D0000h	2D0FFFh
34	559	22F000h	22FFFFFFh
	⋮	⋮	⋮
33	544	220000h	220FFFh
	543	21F000h	21FFFFFFh
32	528	210000h	210FFFh
	527	20F000h	20FFFFFFh
⋮	⋮	⋮	⋮
	512	200000h	200FFFh

Block	Sector	Address range	
31	511	1FF000h	1FFFFFFh
	⋮	⋮	⋮
30	496	1F0000h	1F0FFFh
	495	1EF000h	1EFFFFh
29	480	1E0000h	1E0FFFh
	479	1DF000h	1DFFFFh
⋮	⋮	⋮	⋮
	464	1D0000h	1D0FFFh
18	303	12F000h	12FFFFFFh
	⋮	⋮	⋮
17	288	120000h	120FFFh
	287	11F000h	11FFFFFFh
16	272	110000h	110FFFh
	271	10F000h	10FFFFFFh
⋮	⋮	⋮	⋮
	256	100000h	100FFFh

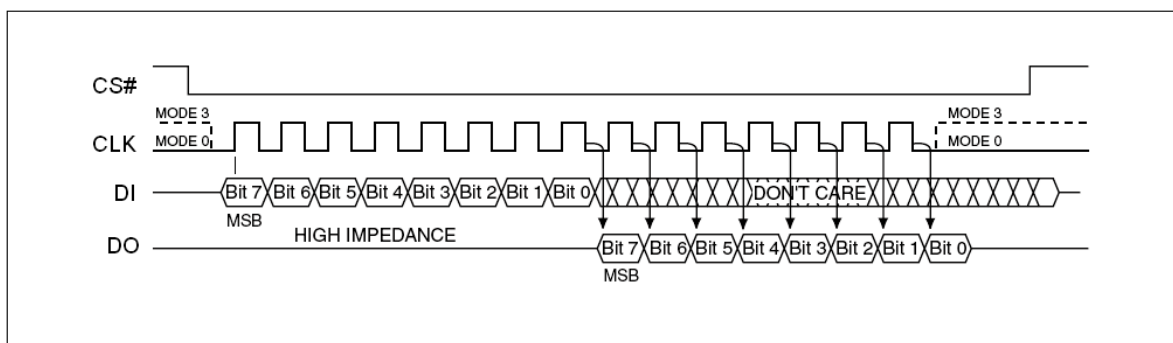
Block	Sector	Address range	
15	255	0FF000h	0FFFFFFh
	⋮	⋮	⋮
14	240	0F0000h	0F0FFFh
	239	0EF000h	0EFFFFh
13	224	0E0000h	0E0FFFh
	223	0DF000h	0DFFFFh
⋮	⋮	⋮	⋮
	208	0D0000h	0D0FFFh
2	47	02F000h	02FFFFFFh
	⋮	⋮	⋮
1	32	020000h	020FFFh
	31	01F000h	01FFFFFFh
0	16	010000h	010FFFh
	15	00F000h	00FFFFFFh
⋮	⋮	⋮	⋮
	4	004000h	004FFFh
⋮	⋮	⋮	⋮
	3	003000h	003FFFh
⋮	⋮	⋮	⋮
	2	002000h	002FFFh
⋮	⋮	⋮	⋮
	1	001000h	001FFFh
⋮	⋮	⋮	⋮
	0	000000h	000FFFh

## OPERATING FEATURES

### Standard SPI Modes

The EN25Q128 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**

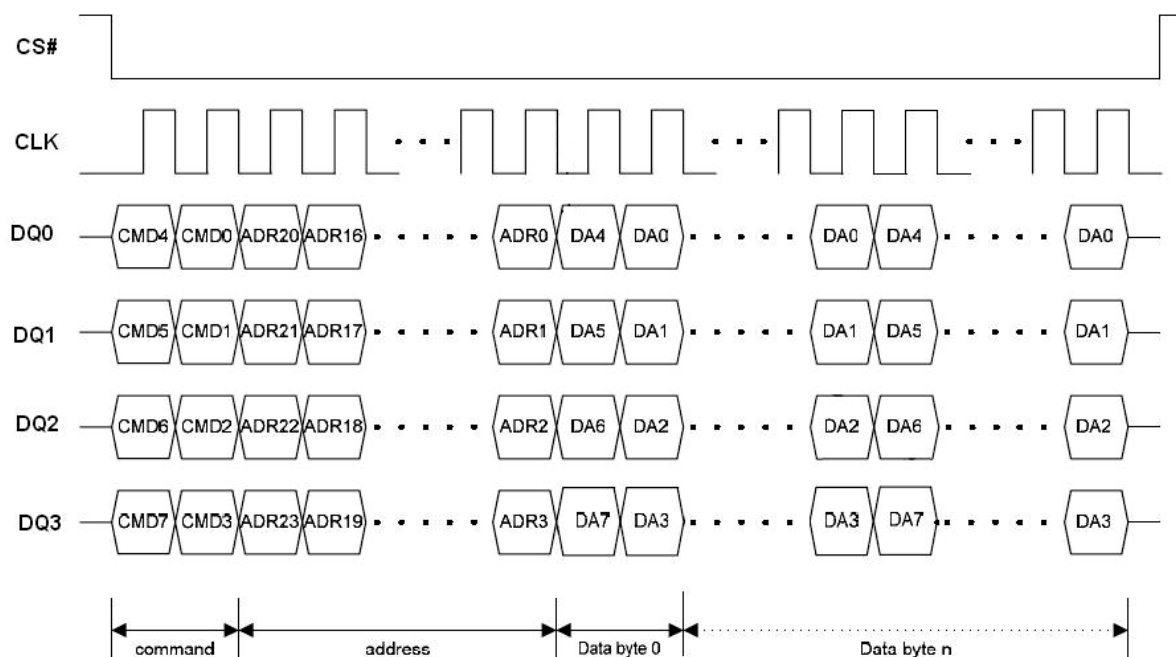


### Dual SPI Instruction

The EN25Q128 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<sub>0</sub> and DQ<sub>1</sub>. All other operations use the standard SPI interface with single output signal.

### Quad SPI Instruction

The EN25Q128 supports Quad output operation when using the Quad I/O Fast Read (EBh). This instruction allows data to be transferred to or from the Serial Flash memory at four to six times the rate possible with the standard SPI. The Quad Read instruction offer a significant improvement in continuous and random access transfer rates allowing fast code-shadowing to RAM or for application that cache code-segments to RAM for execution. The EN25Q128 also supports full Quad Mode 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<sub>0</sub> and DQ<sub>1</sub>, and the WP# and NC pins become DQ<sub>2</sub> and DQ<sub>3</sub> respectively.

**Figure 4. 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) 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) 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, Block Erase and Chip Erase

The Page Program (PP) 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, 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  $t_{SE}$ ,  $t_{BE}$  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) or Erase (SE, BE or CE) can be achieved by not waiting for the worst case delay ( $t_W$ ,  $t_{PP}$ ,  $t_{SE}$ ,  $t_{BE}$  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.

**Status Register**

The Status Register contain a number of status and control bits that can be read or set (as appropriate) by specific instructions.

**WIP bit.** The Write In Progress (WIP) bit indicates whether the memory is busy with a Write Status Register, Program or Erase cycle.

**WEL bit.** The Write Enable Latch (WEL) bit indicates the status of the internal Write Enable Latch.

**BP3, BP2, BP1, BP0 bits.** The Block Protect (BP3, BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against Program and Erase instructions.

**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 (can be floating during SPI mode.)

**SRP bit / OTP\_LOCK bit** The Status Register Protect (SRP) bit operates in conjunction 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. In this mode, the non-volatile bits of the Status Register (SRP, BP3, BP2, BP1, BP0) become read-only bits.

In OTP mode, this bit serves as OTP\_LOCK bit, user can read/program/erase OTP sector as normal sector while OTP\_LOCK bit value is equal 0, after OTP\_LOCK bit is programmed with 1 by WRSR command, the OTP sector is protected from program and erase operation. The OTP\_LOCK bit can only be programmed once.

**Note :** In OTP mode, the WRSR command will ignore any input data and program OTP\_LOCK bit to 1, user must clear the protect bits before entering OTP mode and program the OTP code, then execute WRSR command to lock the OTP sector before leaving OTP mode.

**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 EN25Q128 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) instruction completion or Sector Erase (SE) instruction completion or Block Erase (BE) instruction completion or Chip Erase (CE) instruction completion
- The Block Protect (BP3, 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 (BP3, BP2, BP1, BP0) bits and Status Register Protect (SRP) bit to be protected. This is the Hardware Protected Mode (HPM).
- In addition to the low power consumption feature, the Deep Power-down mode offers extra software protection from inadvertent Write, Program and Erase instructions, as all instructions are ignored except one particular instruction (the Release from Deep Power-down instruction).

**Table 3. Protected Area Sizes Sector Organization**

Status Register Content				Memory Content			
BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protect Areas	Addresses	Density(KB)	Portion
0	0	0	0	None	None	None	None
0	0	0	1	Block 0 to 254	000000h-FEFFFFh	16320KB	Lower 255/256
0	0	1	0	Block 0 to 253	000000h-FDFFFFh	16256KB	Lower 254/256
0	0	1	1	Block 0 to 251	000000h-FBFFFFh	16128KB	Lower 252/256
0	1	0	0	Block 0 to 247	000000h-F7FFFFh	15872KB	Lower 248/256
0	1	0	1	Block 0 to 239	000000h-EFFFFFFh	15360KB	Lower 240/256
0	1	1	0	Block 0 to 223	000000h-DFFFFFFh	14336KB	Lower 224/256
0	1	1	1	All	000000h-FFFFFFh	16384KB	All
1	0	0	0	None	None	None	None
1	0	0	1	Block 255 to 1	FFFFFFh-010000h	16320KB	Upper 255/256
1	0	1	0	Block 255 to 2	FFFFFFh-020000h	16256KB	Upper 254/256
1	0	1	1	Block 255 to 4	FFFFFFh-040000h	16128KB	Upper 252/256
1	1	0	0	Block 255 to 8	FFFFFFh-080000h	15872KB	Upper 248/256
1	1	0	1	Block 255 to 16	FFFFFFh-100000h	15360KB	Upper 240/256
1	1	1	0	Block 255 to 32	FFFFFFh-200000h	14336KB	Upper 224/256
1	1	1	1	All	FFFFFFh-000000h	16384KB	All

## 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 4. 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 Input/Output FAST\_READ (EBh), Read Status Register (RDSR) 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), Sector Erase (SE), Block Erase (BE), Chip Erase (CE), Write Status Register (WRSR), 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), and Release from Deep Power Down (RES ) minimum number of bytes specified has to be given, without which, the command will be ignored.**

**In the case of Page Program, if the number of byte after the command is less than 4 (at least 1 data byte), it will be ignored too. In the case of SE and 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 4A. Instruction Set

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
EQPI	38h						
RSTQIO <sup>(2)</sup>	FFh						
RSTEN	66h						
RST <sup>(1)</sup>	99h						
Write Enable	06h						
Write Disable / Exit OTP mode	04h						
Read Status Register	05h	(S7-S0) <sup>(3)</sup>					continuous <sup>(4)</sup>
Write Status Register	01h	S7-S0					
Page Program	02h	A23-A16	A15-A8	A7-A0	D7-D0	Next byte	continuous
Sector Erase / OTP erase	20h	A23-A16	A15-A8	A7-A0			
Block Erase	D8h	A23-A16	A15-A8	A7-A0			
Chip Erase	C7h/ 60h						
Deep Power-down	B9h						
Release from Deep Power-down, and read Device ID	ABh	dummy	dummy	dummy	(ID7-ID0)		(5)
Release from Deep Power-down							
Manufacturer/ Device ID	90h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(6)
				01h	(ID7-ID0)	(M7-M0)	
Read Identification	9Fh	(M7-M0)	(ID15-ID8)	(ID7-ID0)	(7)		
Enter OTP mode	3Ah						

**Notes:**

1. RST command only executed if RSTEN command is executed first. Any intervening command will disable Reset.
2. Device accepts eight-clocks command in Standard SPI mode, or two-clocks command in Quad SPI mode
3. Data bytes are shifted with Most Significant Bit first. Byte fields with data in parenthesis "( )" indicate data being read from the device on the DO pin
4. The Status Register contents will repeat continuously until CS# terminate the instruction
5. The Device ID will repeat continuously until CS# terminates the instruction
6. The Manufacturer ID and Device ID bytes will repeat continuously until CS# terminates the instruction.  
00h on Byte 4 starts with MID and alternate with DID, 01h on Byte 4 starts with DID and alternate with MID
7. (M7-M0) : Manufacturer, (ID15-ID8) : Memory Type, (ID7-ID0) : Memory Capacity



Table 4B. Instruction Set (Read Instruction)

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	(Next byte)	continuous
Fast Read	0Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(Next Byte) continuous
Dual Output Fast Read	3Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, ...) (1)	(one byte per 4 clocks, continuous)
Dual I/O Fast Read	BBh	A23-A8 <sup>(2)</sup>	A7-A0, dummy (2)	(D7-D0, ...) (1)			(one byte per 4 clocks, continuous)
Quad I/O Fast Read	EBh	A23-A0, dummy (4)	(dummy, D7-D0) (5)	(D7-D0, ...) (3)			(one byte per 2 clocks, continuous)

**Notes:**

## 1. Dual Output data

$$DQ_0 = (D6, D4, D2, D0)$$

$$DQ_1 = (D7, D5, D3, D1)$$

## 2. Dual Input Address

$$DQ_0 = A22, A20, A18, A16, A14, A12, A10, A8 ; A6, A4, A2, A0, \text{dummy } 6, \text{dummy } 4, \text{dummy } 2, \text{dummy } 0$$

$$DQ_1 = A23, A21, A19, A17, A15, A13, A11, A9 ; A7, A5, A3, A1, \text{dummy } 7, \text{dummy } 5, \text{dummy } 3, \text{dummy } 1$$

## 3. Quad Data

$$DQ_0 = (D4, D0, \dots)$$

$$DQ_1 = (D5, D1, \dots)$$

$$DQ_2 = (D6, D2, \dots)$$

$$DQ_3 = (D7, D3, \dots)$$

## 4. Quad Input Address

$$DQ_0 = A20, A16, A12, A8, A4, A0, \text{dummy } 4, \text{dummy } 0$$

$$DQ_1 = A21, A17, A13, A9, A5, A1, \text{dummy } 5, \text{dummy } 1$$

$$DQ_2 = A22, A18, A14, A10, A6, A2, \text{dummy } 6, \text{dummy } 2$$

$$DQ_3 = A23, A19, A15, A11, A7, A3, \text{dummy } 7, \text{dummy } 3$$

## 5. Quad I/O Fast Read Data

$$DQ_0 = (\text{dummy } 12, \text{dummy } 8, \text{dummy } 4, \text{dummy } 0, D4, D0)$$

$$DQ_1 = (\text{dummy } 13, \text{dummy } 9, \text{dummy } 5, \text{dummy } 1, D5, D1)$$

$$DQ_2 = (\text{dummy } 14, \text{dummy } 10, \text{dummy } 6, \text{dummy } 2, D6, D2)$$

$$DQ_3 = (\text{dummy } 15, \text{dummy } 11, \text{dummy } 7, \text{dummy } 3, D7, D3)$$

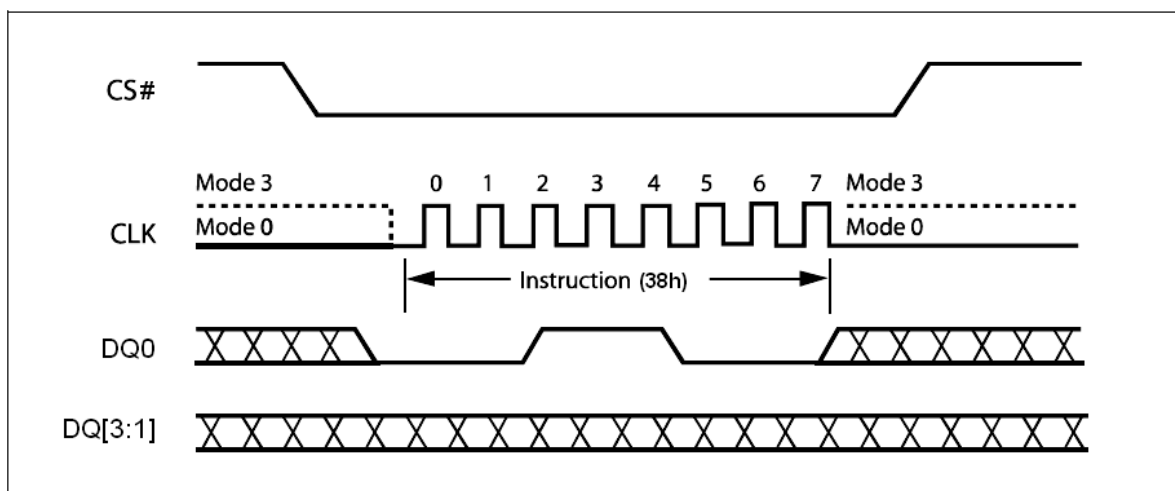


**Table 5. Manufacturer and Device Identification**

OP Code	(M7-M0)	(ID15-ID0)	(ID7-ID0)
ABh			17h
90h	1Ch		17h
9Fh	1Ch	3018h	

**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 5. The device did not support the Read Data Bytes (READ) (03h), Dual Output Fast Read (3Bh) and Dual Input/Output FAST\_READ (BBh) modes while the Enable Quad Peripheral Interface mode (EQPI) (38h) turns on.


**Figure 5. Enable Quad Peripheral Interface mode Sequence Diagram**
**Reset Quad I/O (RSTQIO) (FFh)**

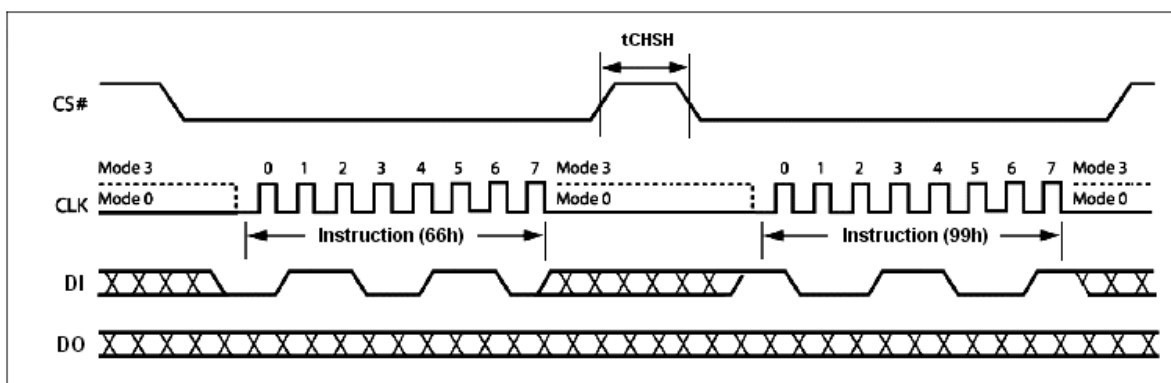
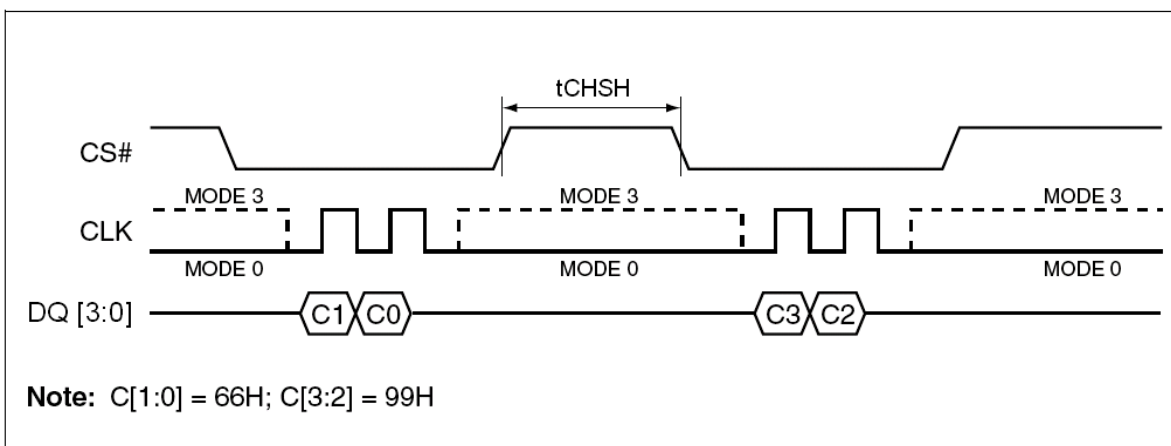
The Reset Quad I/O instruction resets the device to 1-bit Standard SPI operation. To execute a Reset Quad I/O operation, the host drives CS# low, sends the Reset Quad I/O command cycle (FFh) then, drives CS# high. This command can't be used in Standard SPI mode.

**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).

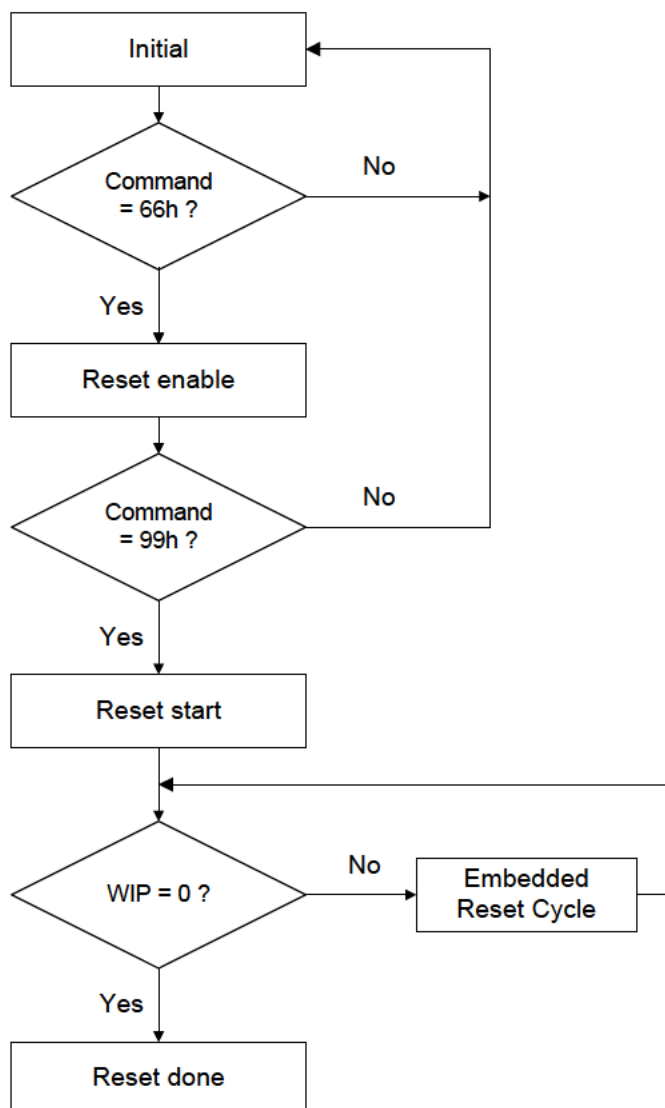
To reset the EN25Q128 the host drives CS# low, sends the Reset-Enable command (66h), and drives CS# high. Next, the host drives CS# low again, sends the Reset command (99h), and drives CS# high. The Reset operation requires the Reset-Enable command followed by the Reset command. Any command other than 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 6 for SPI Mode and Figure 6.1 for EQPI 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 6. Reset-Enable and Reset Sequence Diagram**


**Note:** C[1:0] = 66H; C[3:2] = 99H

**Figure 6.1 Reset-Enable and Reset Sequence Diagram under EQPI Mode**

**Software Reset Flow**

**Note:**

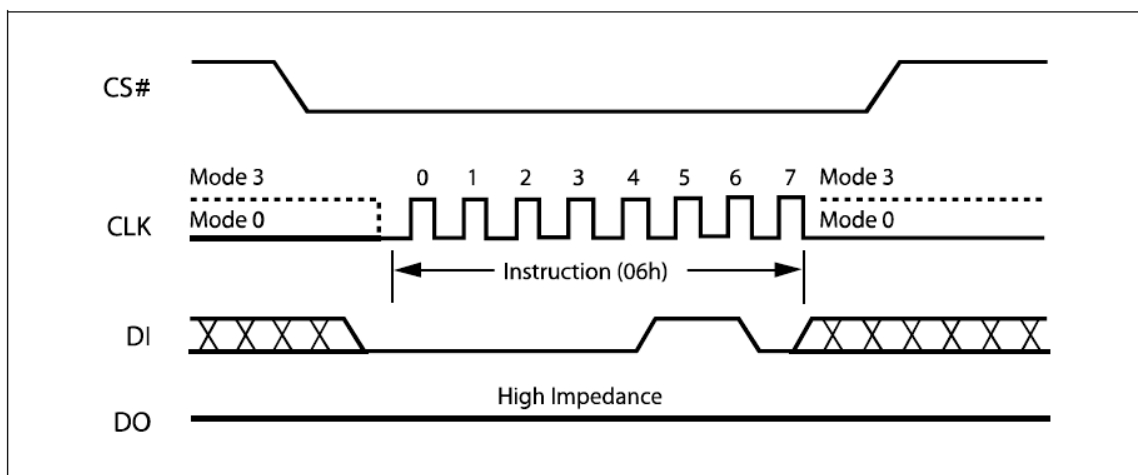
1. Reset-Enable (RSTEN) (66h) and Reset (RST) (99h) commands need to match standard SPI or EQPI (Quad) mode.
2. Continue (Enhance) EB mode need to use quad Reset-Enable (RSTEN) (66h) and quad Reset (RST) (99h) commands.
3. If user is not sure it is in SPI or Quad mode, we suggest to execute sequence 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, EQPI mode and Continue EB mode to back to SPI mode.
5. This flow cannot release the device from Deep power down mode.
6. The Status Register 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.

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

The Write Enable (WREN) instruction (Figure 7) sets the Write Enable Latch (WEL) bit. The Write Enable Latch (WEL) bit must be set prior to every Page Program (PP), Sector Erase (SE), Block Erase (BE), Chip Erase (CE) and Write Status Register (WRSR) 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 8.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

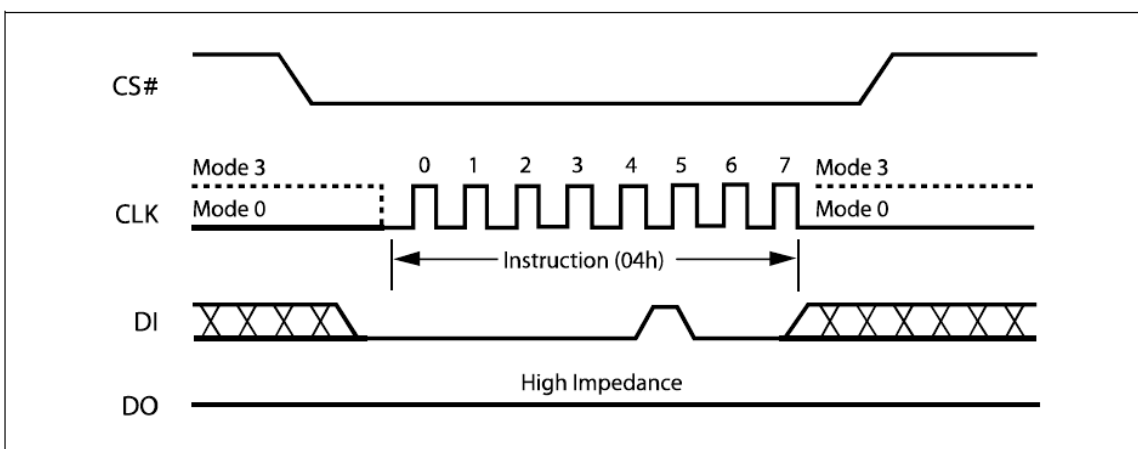


**Figure 7. Write Enable Instruction Sequence Diagram**

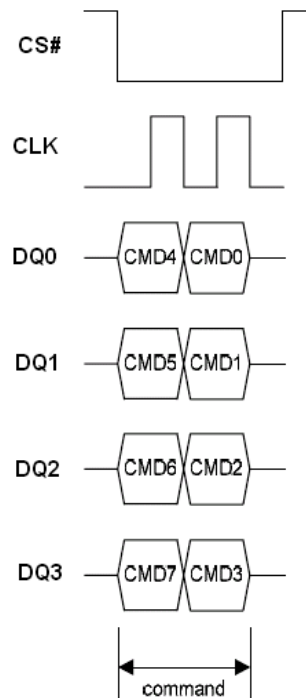
**Write Disable (WRDI) (04h)**

The Write Disable instruction (Figure 8) 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, Block Erase (BE) and Chip Erase instructions.

The instruction sequence is shown in Figure 8.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.



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

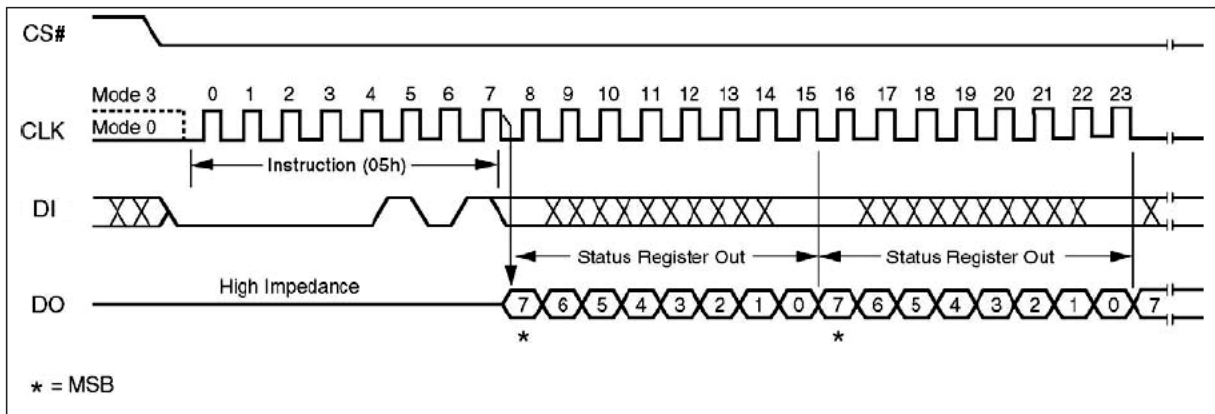


**Figure 8.1 Write Enable/Disable Instruction Sequence under EQPI Mode**

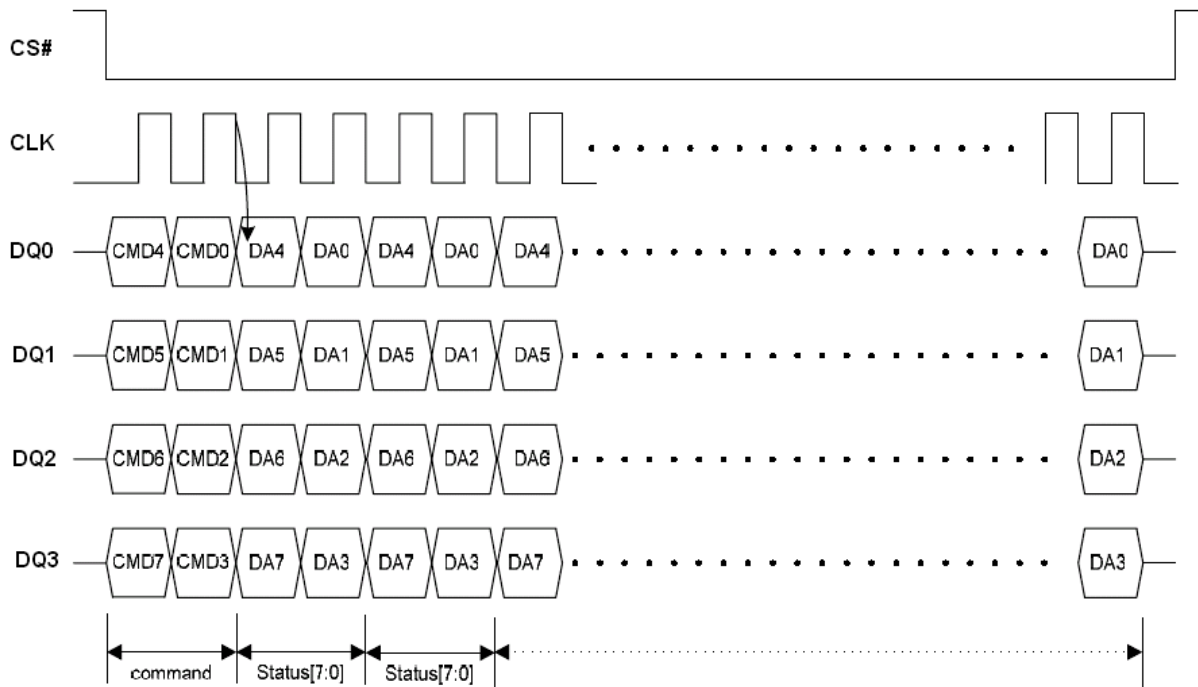
**Read Status Register (RDSR) (05h)**

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 9.

The instruction sequence is shown in Figure 9.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.



**Figure 9. Read Status Register Instruction Sequence Diagram**


**Figure 9.1 Read Status Register Instruction Sequence under EQPI Mode**
**Table 6. Status Register Bit Locations**

S7		S6	S5	S4	S3	S2	S1	S0
SRP Status Register Protect	OTP_LOCK bit (note 1)	WPDIS (WP# disable)	BP3 (Block Protected bits)	BP2 (Block Protected bits)	BP1 (Block Protected bits)	BP0 (Block Protected bits)	WEL (Write Enable Latch)	WIP (Write In Progress bit) (Note 3)
1 = status register write disable	1 = OTP sector is protected	1 = WP# disable 0 = WP# enable	(note 2)	(note 2)	(note 2)	(note 2)	1 = write enable 0 = not write enable	1 = write operation 0 = not in write operation
Non-volatile bit		Non-volatile bit	Non-volatile bit.	Non-volatile bit	Non-volatile bit	Non-volatile bit	volatile bit	volatile bit

**Note**

1. In OTP mode, SRP bit is served as OTP\_LOCK bit.
2. See the table "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.

**BP3, BP2, BP1, BP0 bits.** The Block Protect (BP3, 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 (BP3, BP2, BP1, BP0) bits is set to 1, the relevant memory area (as defined in Table 3.) becomes protected against Page Program (PP) Sector Erase (SE) and , Block Erase (BE), instructions. The Block Protect



(BP3, 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, and only if, all Block Protect (BP3, BP2, BP1, BP0) bits are 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 (can be floating during SPI mode.)

**SRP bit / OTP\_LOCK bit.** The Status Register Protect (SRP) bit operates 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, BP3, BP2, BP1, BP0) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

In OTP mode, this bit serves as OTP\_LOCK bit, user can read/program/erase OTP sector as normal sector while OTP\_LOCK value is equal 0, after OTP\_LOCK is programmed with 1 by WRSR command, the OTP sector is protected from program and erase operation. The OTP\_LOCK bit can only be programmed once.

**Note :** In OTP mode, the WRSR command will ignore any input data and program OTP\_LOCK bit to 1, user must clear the protect bits before enter OTP mode and program the OTP code, then execute WRSR command to lock the OTP sector before leaving OTP mode.

#### 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).

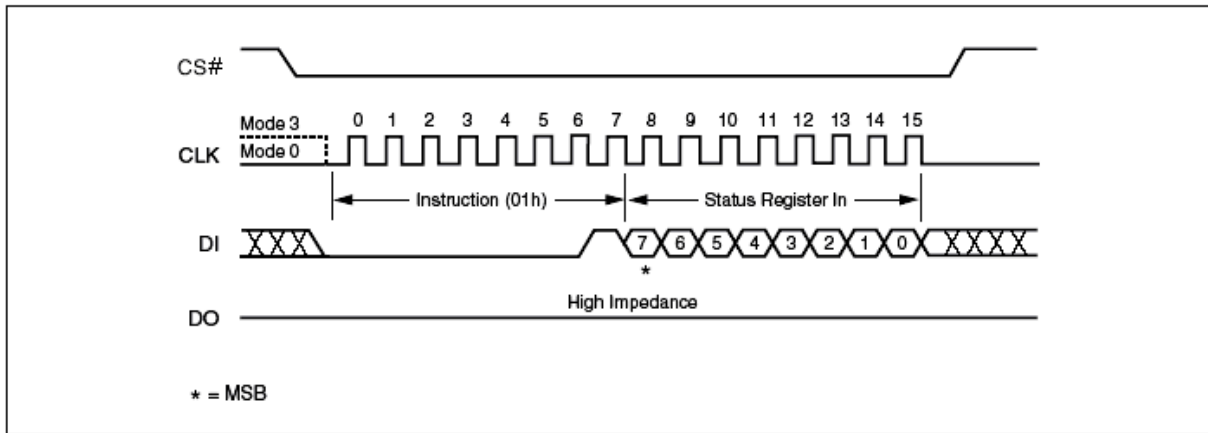
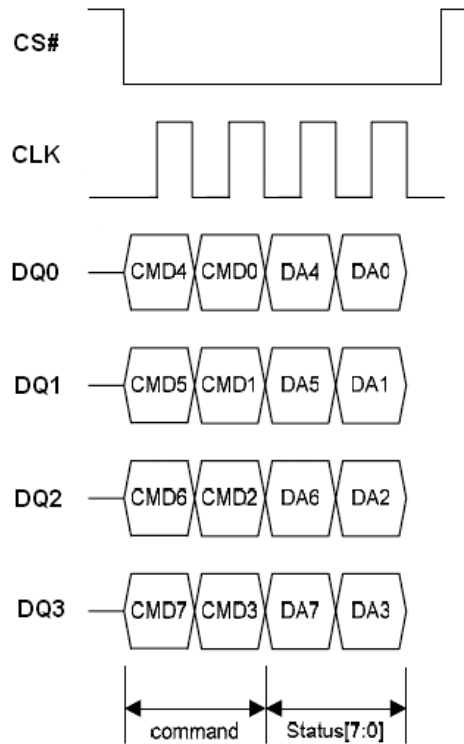
The Write Status Register (WRSR) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code and the data byte on Serial Data Input (DI).

The instruction sequence is shown in Figure 10. The Write Status Register (WRSR) instruction has no effect on S1 and S0 of the Status Register. Chip Select (CS#) must be driven High after the 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 (BP3, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in Table 3. 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 10.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

**NOTE :** In the OTP mode, WRSR command will ignore input data and program OTP\_LOCK bit to 1.


**Figure 10. Write Status Register Instruction Sequence Diagram**

**Figure 10.1 Write Status Register Instruction Sequence under EQPI Mode**

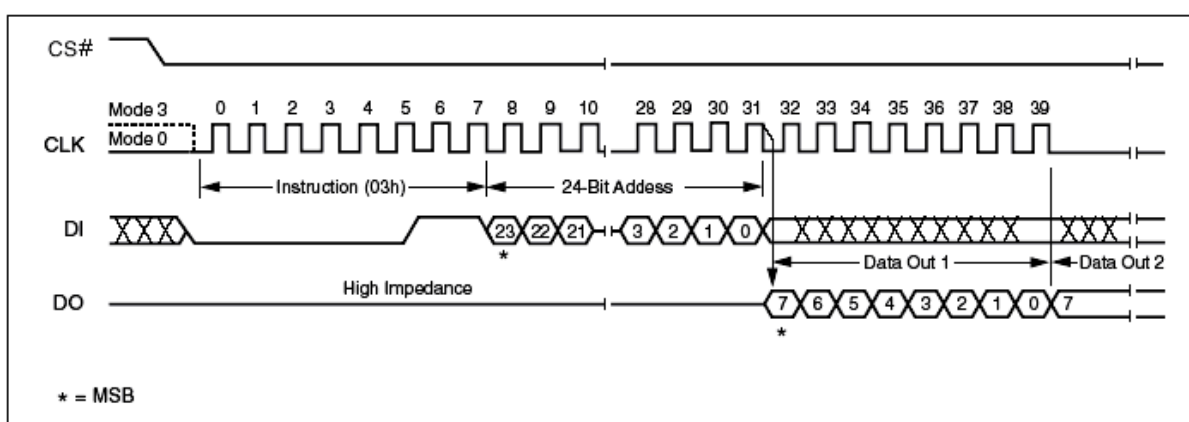


**Read Data Bytes (READ) (03h)**

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 11. The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single Read Data Bytes (READ) instruction. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

The Read Data Bytes (READ) instruction is terminated by driving Chip Select (CS#) High. Chip Select (CS#) can be driven High at any time during data output. Any Read Data Bytes (READ) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.



**Figure 11. 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 12. 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 12.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

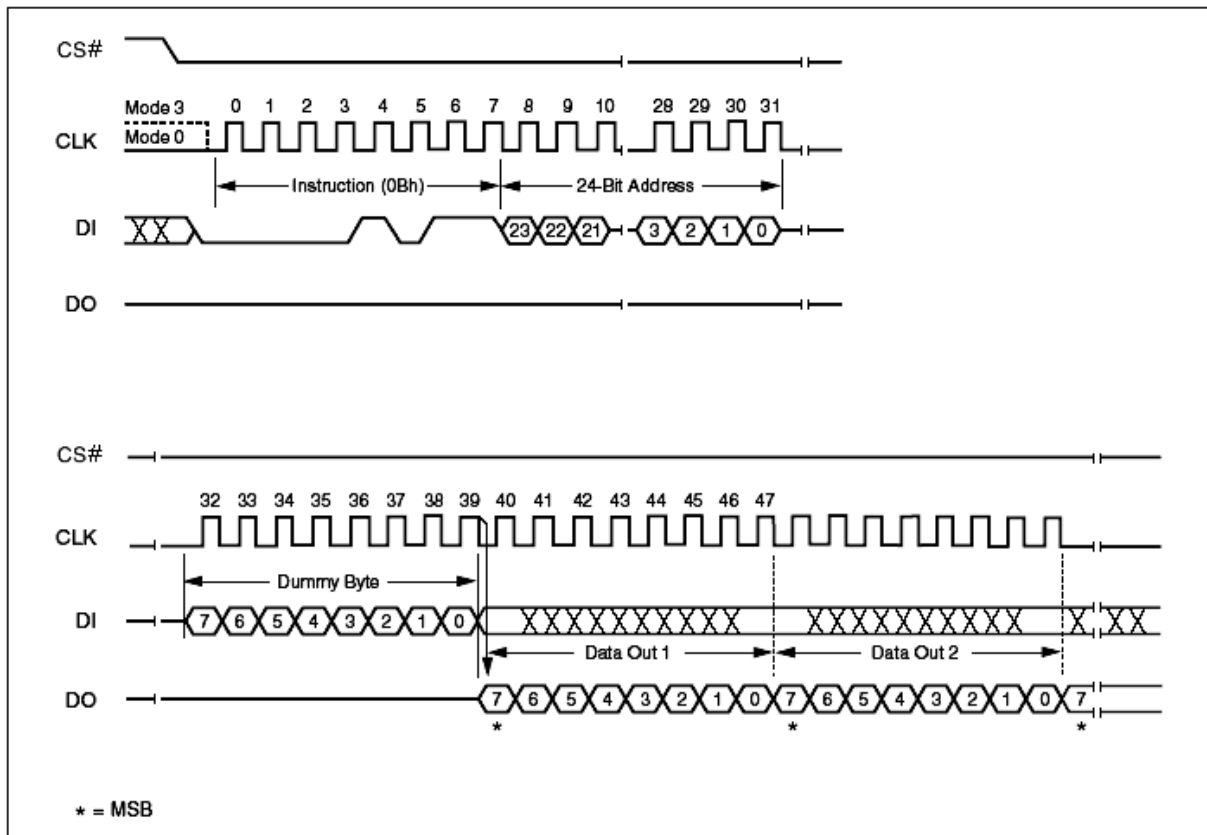


Figure 12. Fast Read Instruction Sequence Diagram

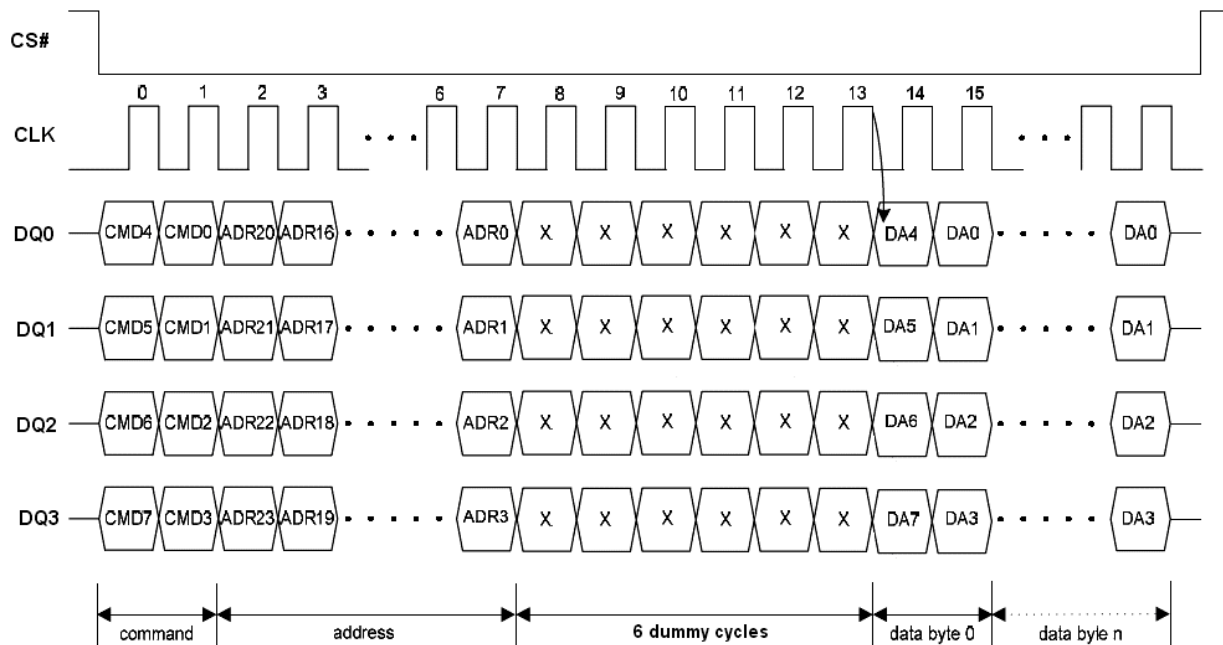
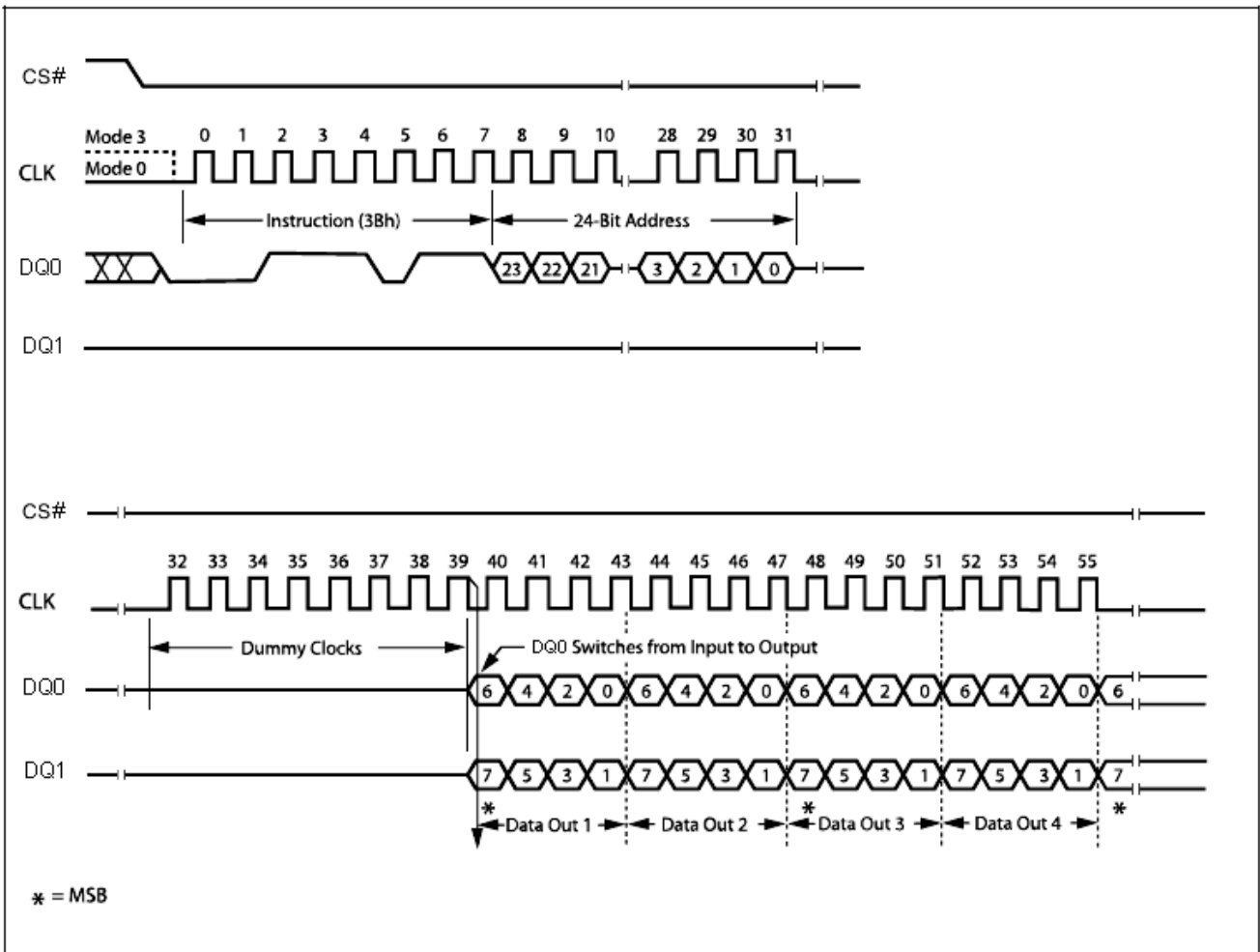


Figure 12.1 Fast Read Instruction Sequence under EQPI Mode

**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<sub>0</sub> and DQ<sub>1</sub>, instead of just DQ<sub>0</sub>. This allows data to be transferred from the EN25Q128 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 FR (see AC Electrical Characteristics). This is accomplished by adding eight “dummy clocks after the 24-bit address as shown in Figure 13. 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.

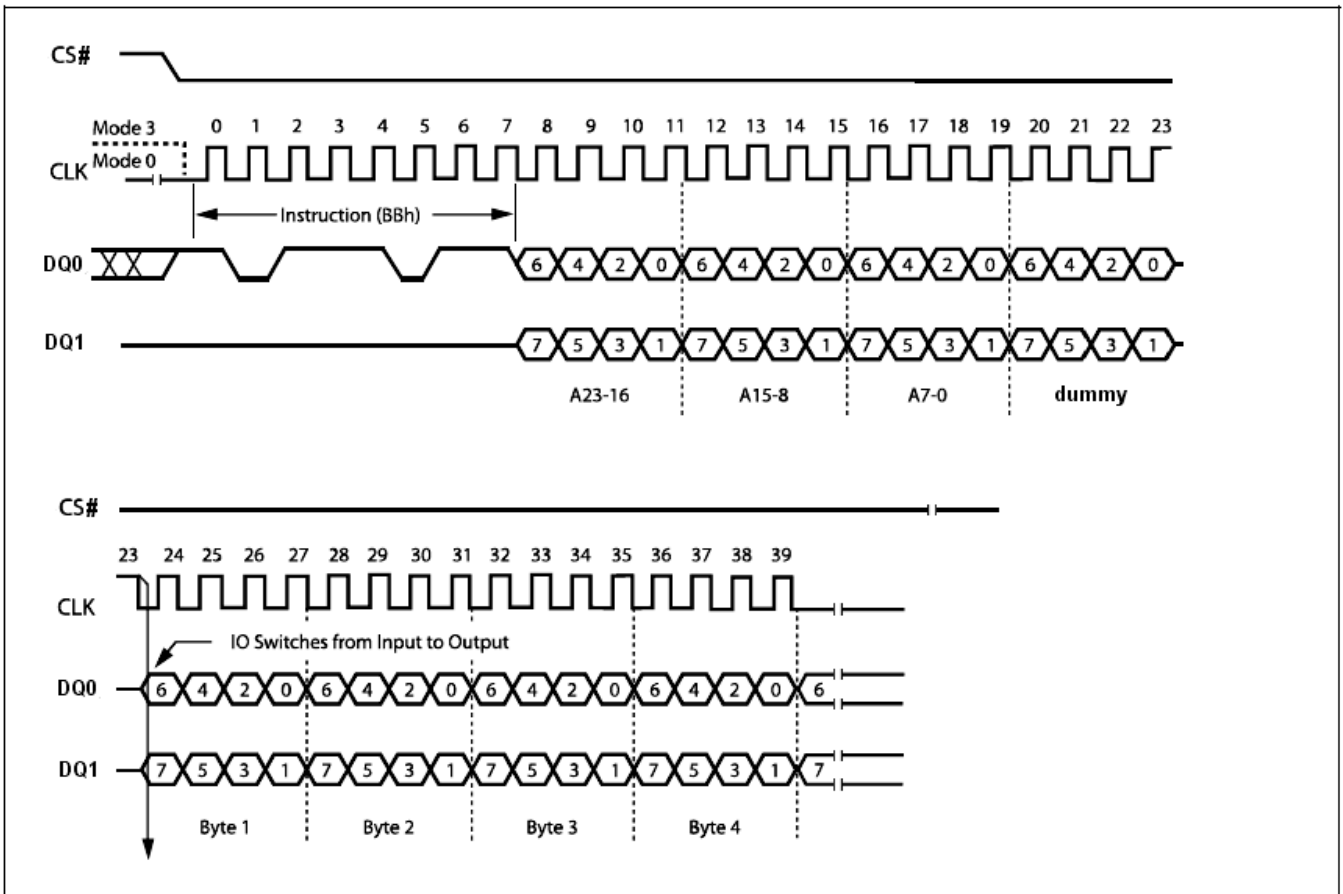


**Figure 13. Dual Output Fast Read Instruction Sequence Diagram**

**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<sub>0</sub> and DQ<sub>1</sub>. It is similar to the Dual Output Fast Read (3Bh) instruction but with the capability to input the Address bits (A23-0) two bits per clock. This reduced instruction overhead may allow for code execution (XIP) directly from the Dual SPI in some applications.

The Dual I/O Fast Read instruction enable double throughput of Serial Flash in read mode. The address is latched on rising edge of CLK, and data of every two bits (interleave 2 I/O pins) shift out on the falling edge of CLK at a maximum frequency. The first address can be at any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Dual I/O Fast Read instruction. The address counter rolls over to 0 when the highest address has been reached. Once writing Dual I/O Fast Read instruction, the following address/dummy/data out will perform as 2-bit instead of previous 1-bit, as shown in Figure 14.



**Figure 14. Dual Input / Output Fast Read Instruction Sequence Diagram**

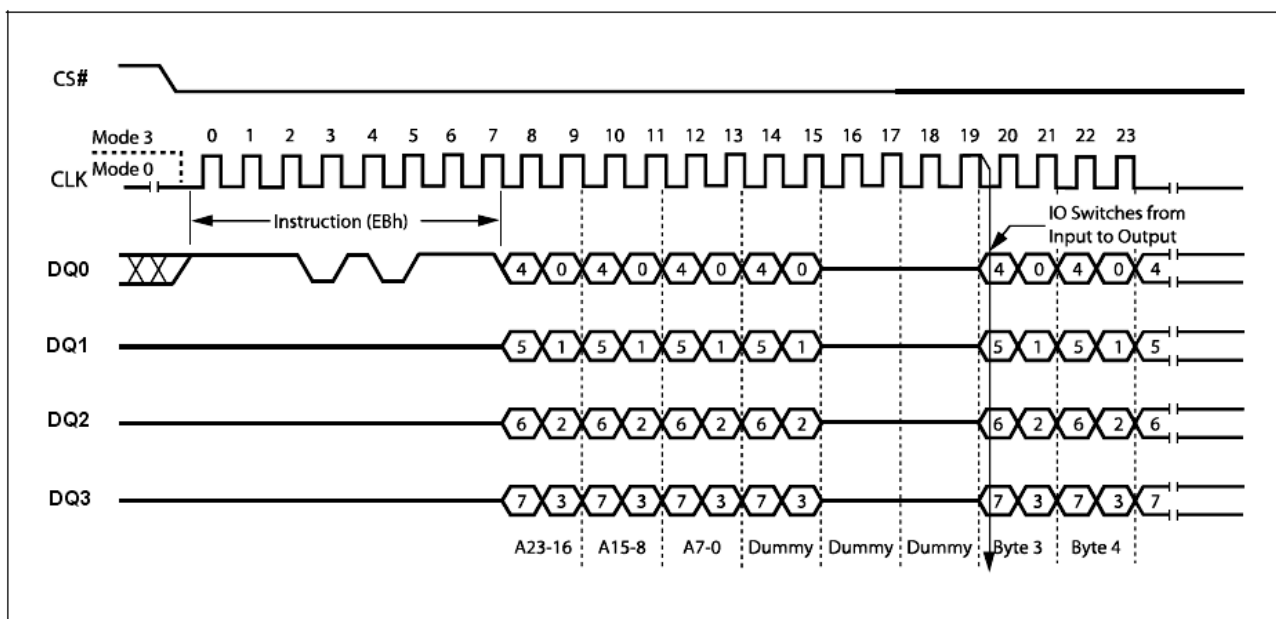
**Quad Input / Output FAST\_READ (EBh)**

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<sub>0</sub>, DQ<sub>1</sub>, DQ<sub>2</sub> and DQ<sub>3</sub> and six dummy clocks are required prior to the data output. The Quad I/O dramatically reduces instruction overhead allowing faster random access for code execution (XIP) directly from the Quad SPI.

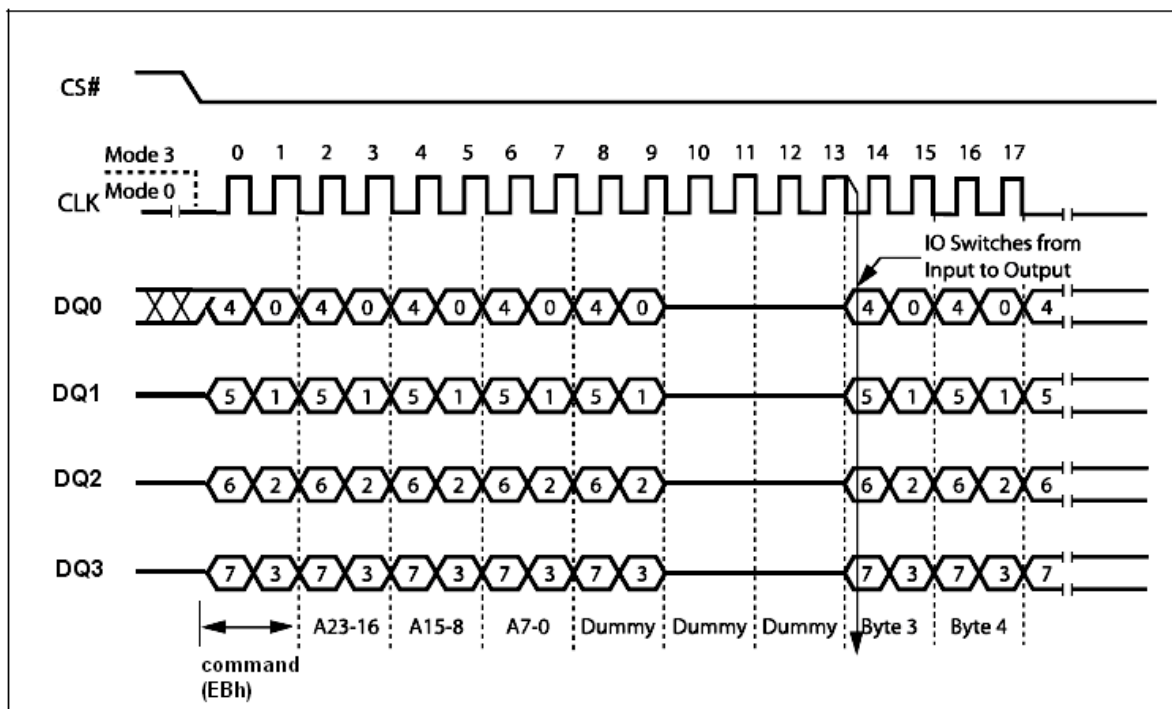
The Quad Input/Output FAST\_READ (EBh) instruction enable quad throughput of Serial Flash in read mode. The address is latching on rising edge of CLK, and data of every four bits (interleave on 4 I/O pins) shift out on the falling edge of CLK at a maximum frequency F<sub>R</sub>. The first address can 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<sub>3</sub>, DQ<sub>2</sub>, DQ<sub>1</sub> and DQ<sub>0</sub> -> 6 dummy clocks -> data out interleave on DQ<sub>3</sub>, DQ<sub>2</sub>, DQ<sub>1</sub> and DQ<sub>0</sub> -> to end Quad Input/Output FAST\_READ (EBh) operation can use CS# to high at any time during data out, as shown in Figure 15.

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



**Figure 15. Quad Input / Output Fast Read Instruction Sequence Diagram**



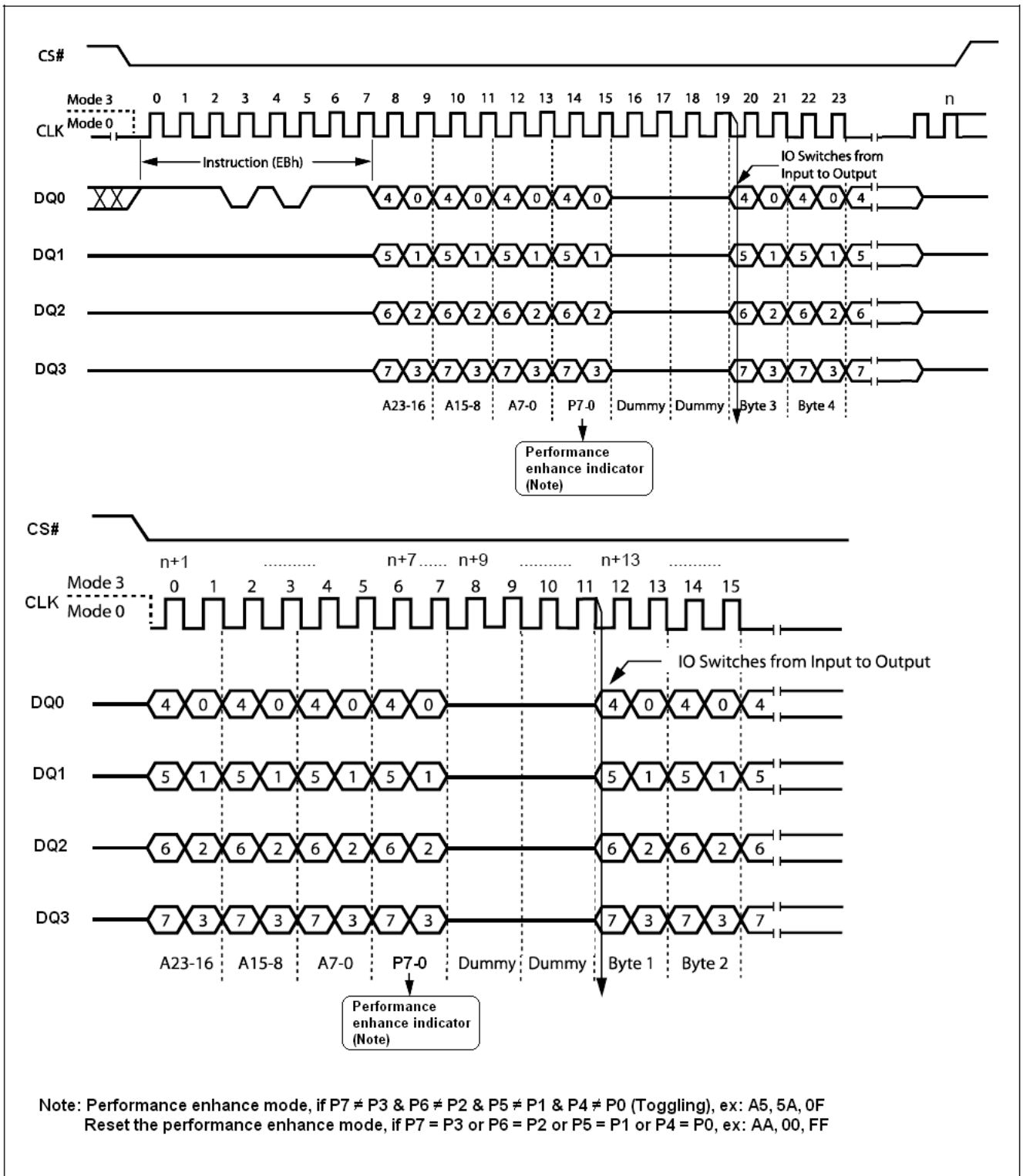
**Figure 15.1. Quad Input / Output Fast Read Instruction Sequence under EQPI Mode**

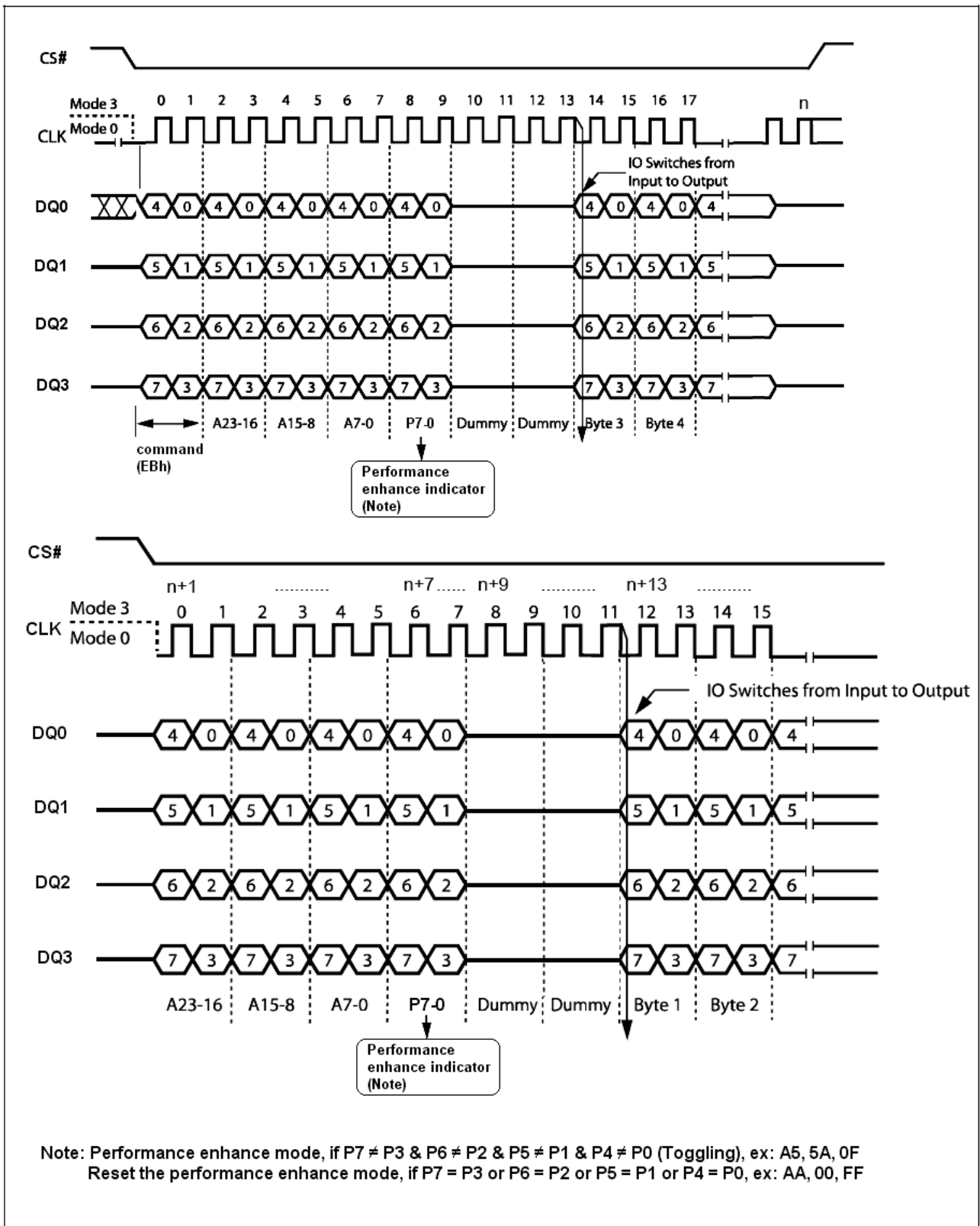
Another sequence of issuing Quad Input/Output FAST\_READ (EBh) instruction especially useful in random access is : CS# goes low -> sending Quad Input/Output FAST\_READ (EBh) instruction -> 24-bit address interleave on DQ<sub>3</sub>, DQ<sub>2</sub>, DQ<sub>1</sub> and DQ<sub>0</sub> -> performance enhance toggling bit P[7:0] -> 4 dummy clocks -> data out interleave on DQ<sub>3</sub>, DQ<sub>2</sub>, DQ<sub>1</sub> and DQ<sub>0</sub> till CS# goes high -> CS# goes low (reduce Quad Input/Output FAST\_READ (EBh) instruction) -> 24-bit random access address, as shown in Figure 16.

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 16.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.


**Figure 16. Quad Input/Output Fast Read Enhance Performance Mode Sequence Diagram**


**Figure 16.1 Quad Input/Output Fast Read Enhance Performance Mode Sequence under EQPI 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.

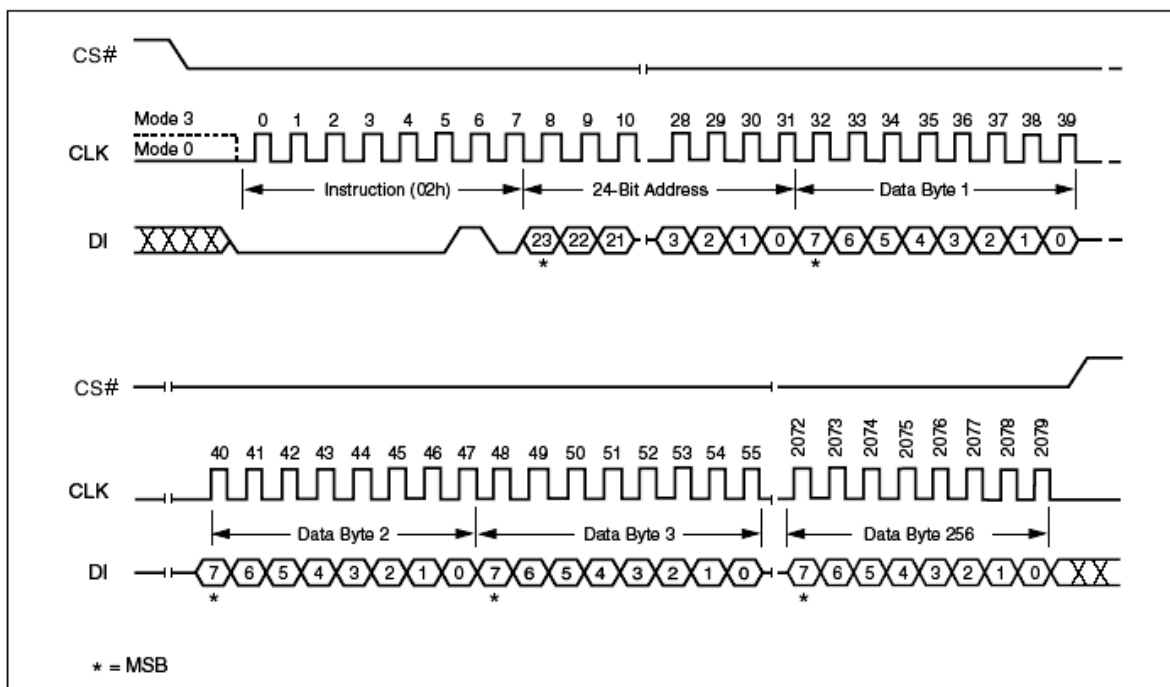
The instruction sequence is shown in Figure 17. If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 Data bytes are sent to device, they are correctly programmed 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.

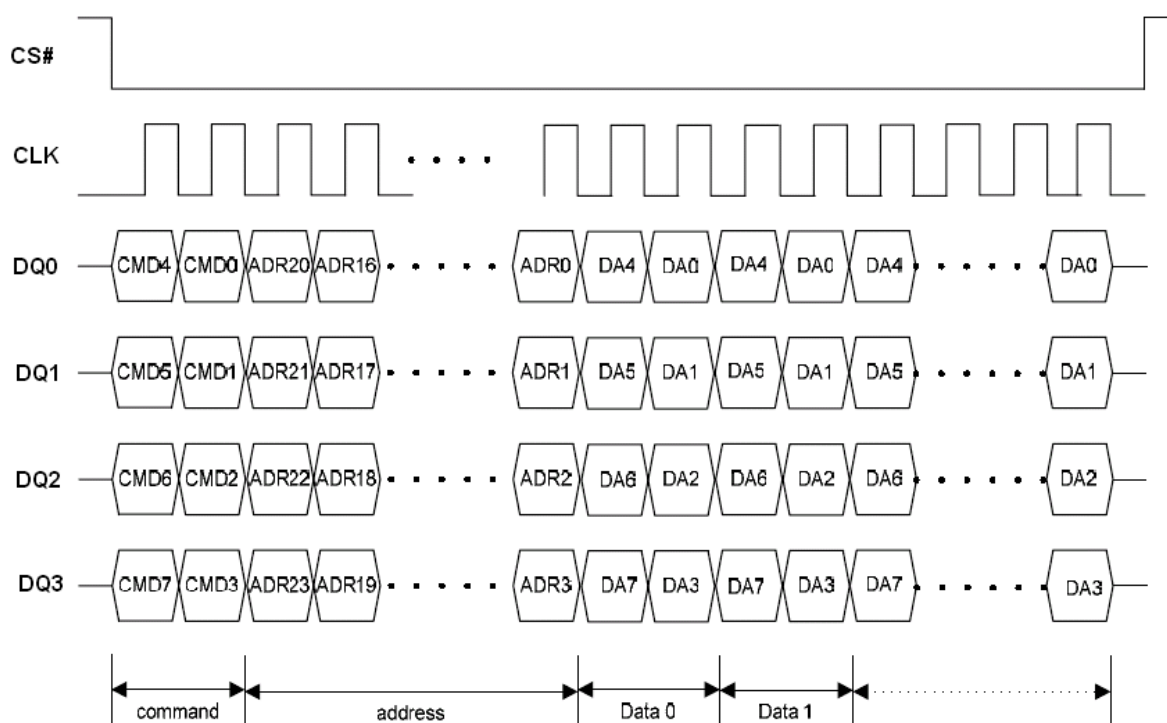
As soon as Chip Select (CS#) is driven High, the self-timed Page Program cycle (whose duration is  $t_{pp}$ ) is initiated. While the Page Program cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the 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 (BP3, BP2, BP1, BP0) bits (see Table 3) is not executed.

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



**Figure 17. Page Program Instruction Sequence Diagram**



**Figure 17.1 Program Instruction Sequence under EQPI Mode**

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

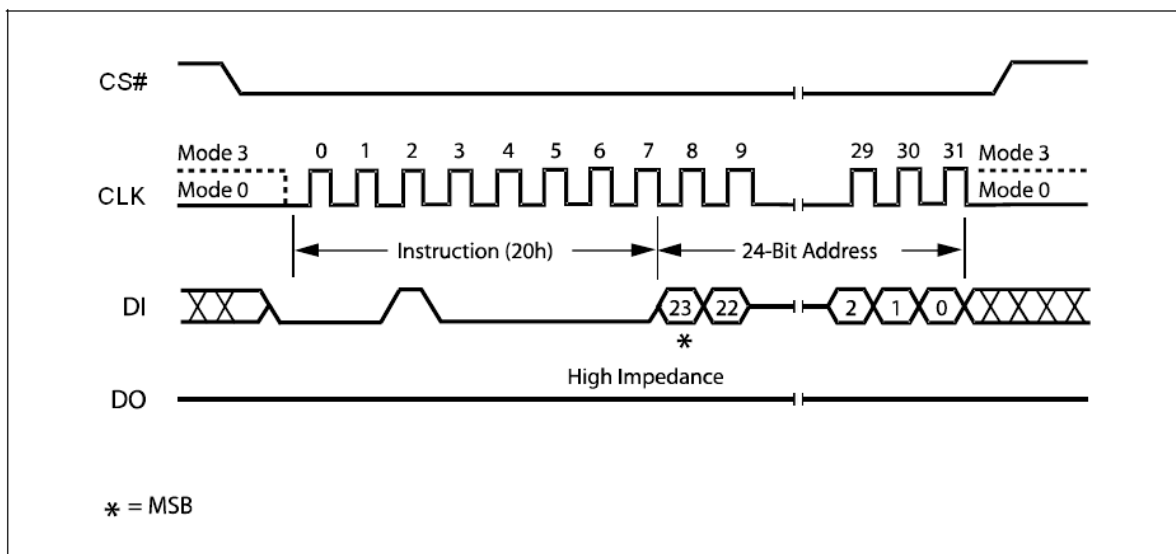
The Sector Erase (SE) instruction sets to 1 (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

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 2) 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 18. 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 (BP3, BP2, BP1, BP0) bits (see Table 3) is not executed.

The instruction sequence is shown in Figure 19.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.



**Figure 18. Sector Erase Instruction Sequence Diagram**

### Block Erase (BE) (D8h)

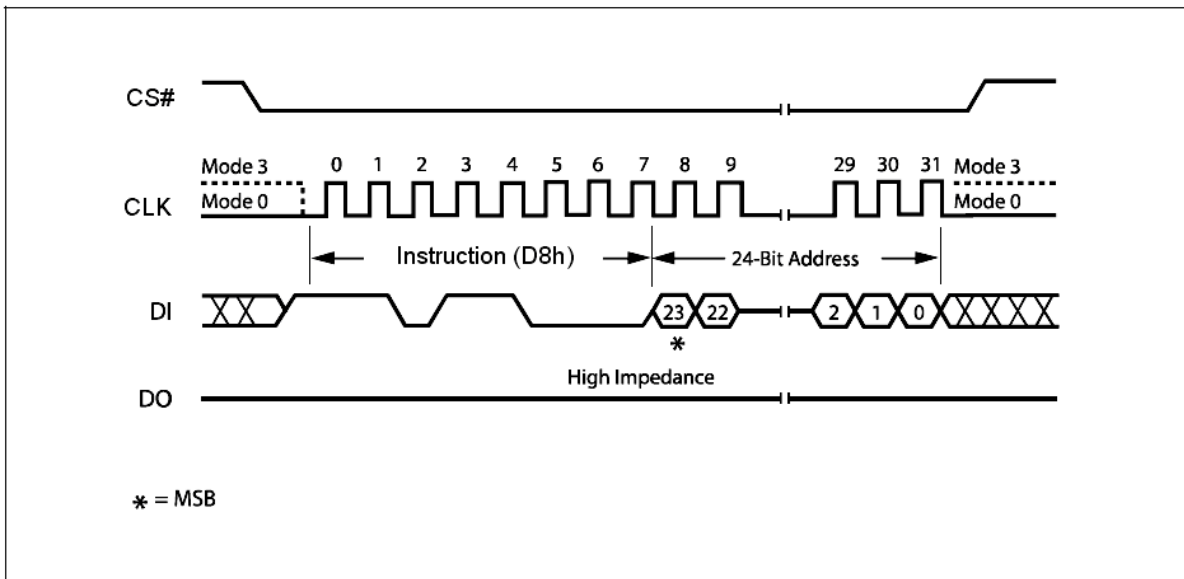
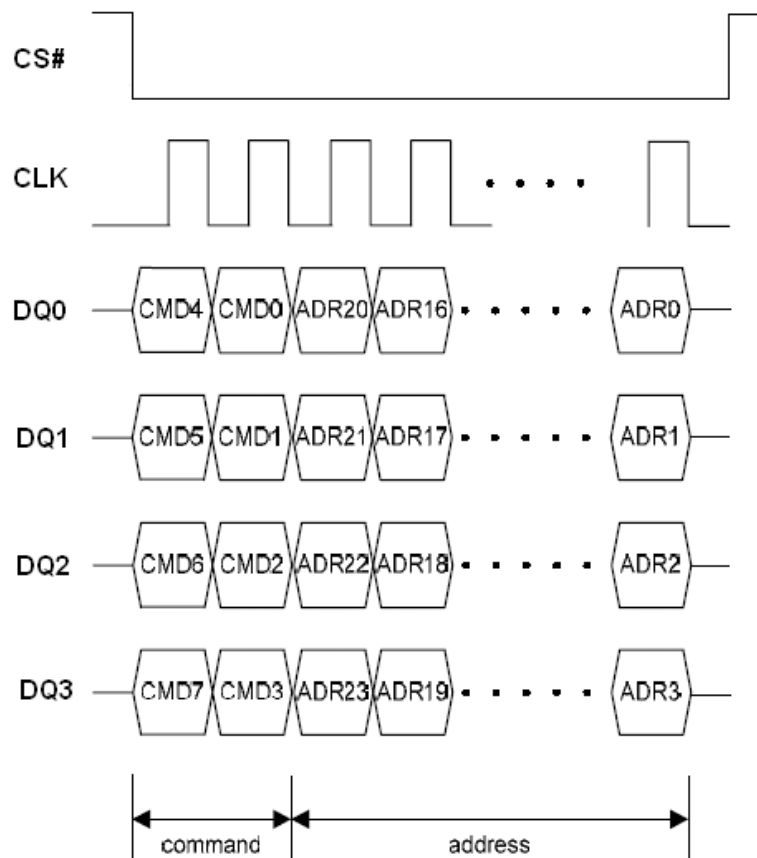
The Block Erase (BE) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

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 2) 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 19. 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 (BP3, BP2, BP1, BP0) bits (see Table 3) is not executed.

The instruction sequence is shown in Figure 19.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.


**Figure 19. Block Erase Instruction Sequence Diagram**

**Figure 19.1 Block/Sector Erase Instruction Sequence under EQPI Mode**

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

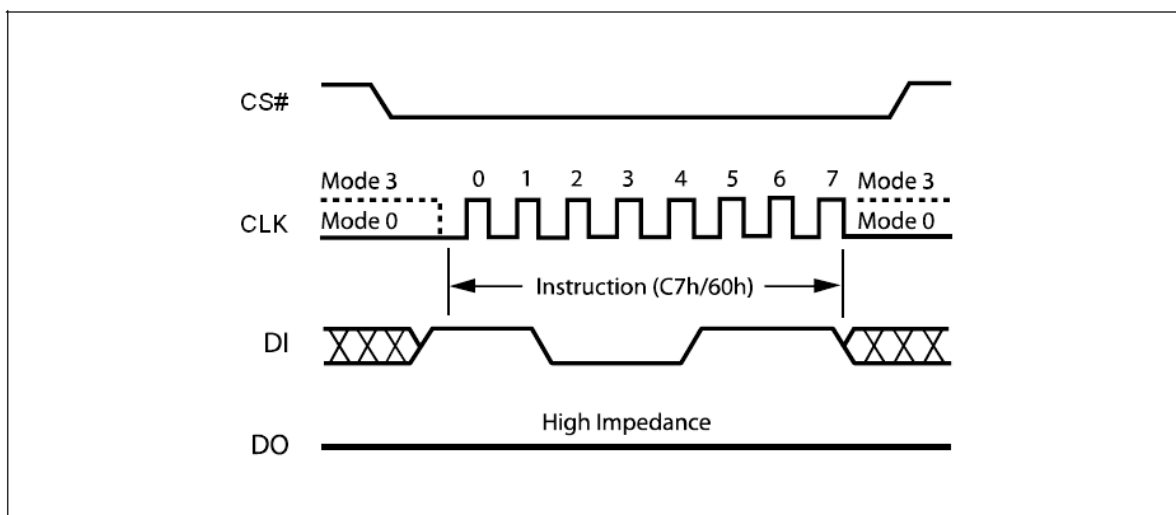
The Chip Erase (CE) instruction sets all bits to 1 (FFh). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

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.

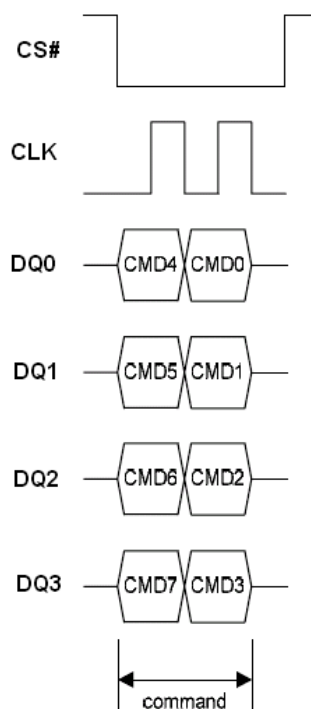
The instruction sequence is shown in Figure 20. Chip Select (CS#) must be driven High after the eighth bit of the instruction code has been latched in, otherwise the Chip Erase instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Chip Erase cycle (whose duration is  $t_{CE}$ ) 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 executed only if all Block Protect (BP3, BP2, BP1, BP0) bits are 0. The Chip Erase (CE) instruction is ignored if one, or more blocks are protected.

The instruction sequence is shown in Figure 20.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.



**Figure 20. Chip Erase Instruction Sequence Diagram**



**Figure 20.1 Chip Erase Sequence under EQPI 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.

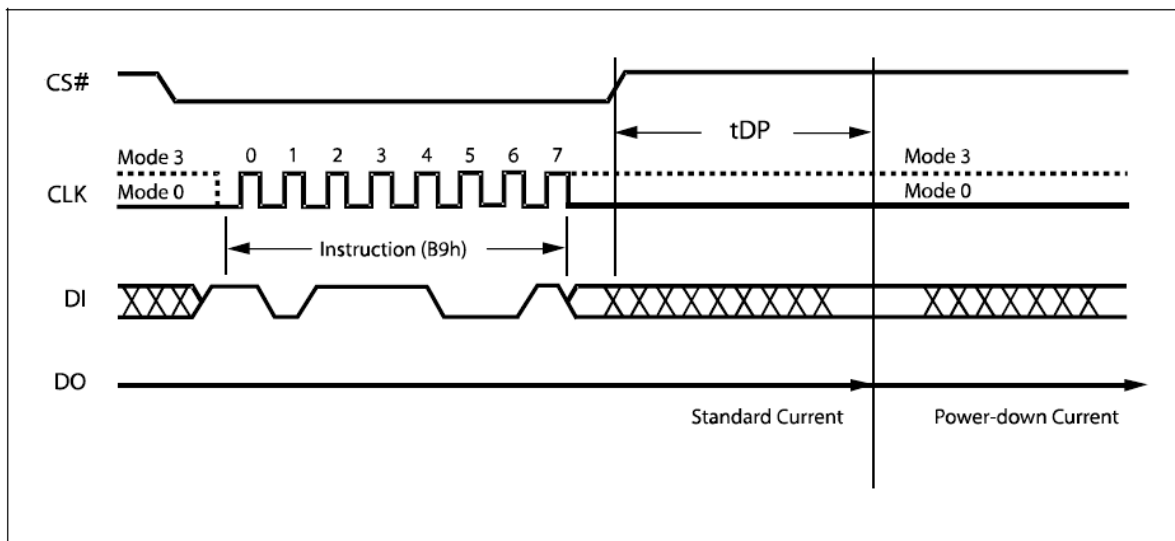
Driving Chip Select (CS#) High deselects the device, and puts the device in the Standby mode (if there is no internal cycle currently in progress). But this mode is not the Deep Power-down mode. The Deep Power-down mode can only be entered by executing the Deep Power-down (DP) instruction, to reduce the standby current (from  $I_{CC1}$  to  $I_{CC2}$ , as specified in Table 9.)

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 21. 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 21. 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.

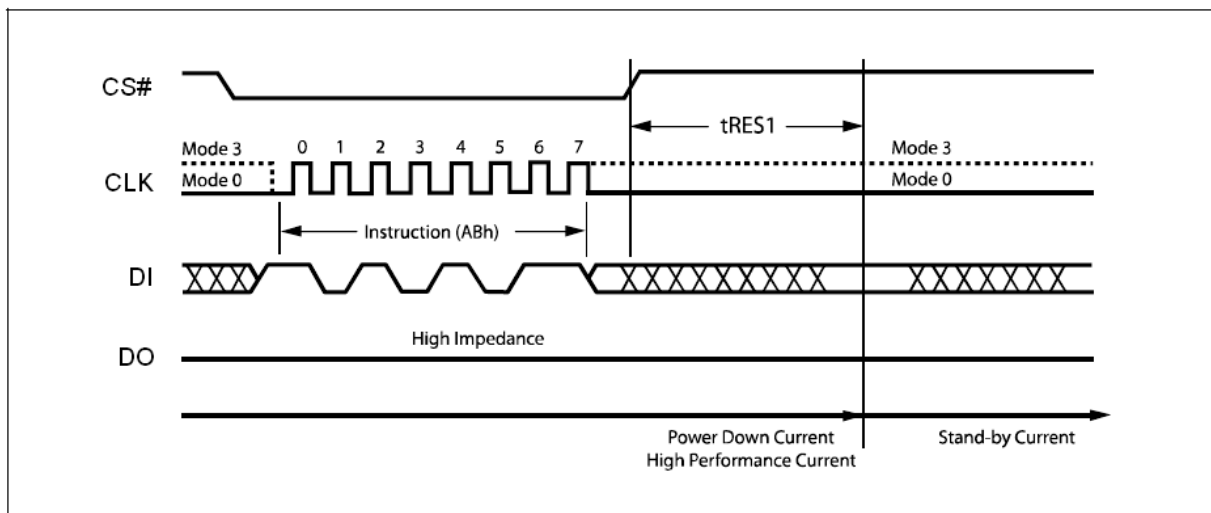
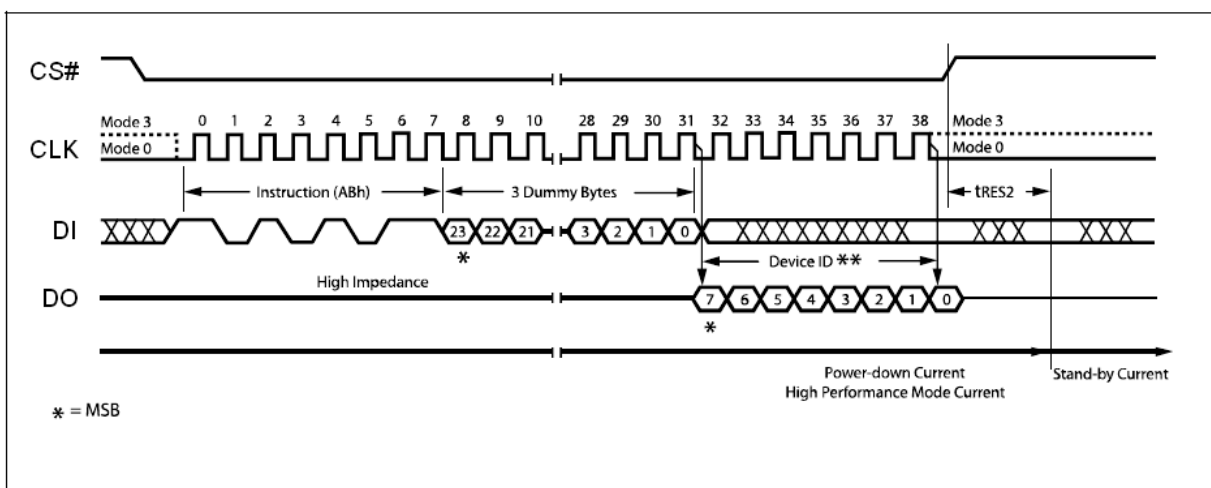
When used only to release the device from the power-down state, the instruction is issued by driving the CS# pin low, shifting the instruction code “ABh” and driving CS# high as shown in Figure 22. After the time duration of  $t_{RES1}$  (See AC Characteristics) the device will resume normal operation 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 23. The Device ID value for the EN25Q128 are listed in Table 5. 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 11. 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 22. Release Power-down Instruction Sequence Diagram**

**Figure 23. Release Power-down / Device ID Instruction Sequence Diagram**

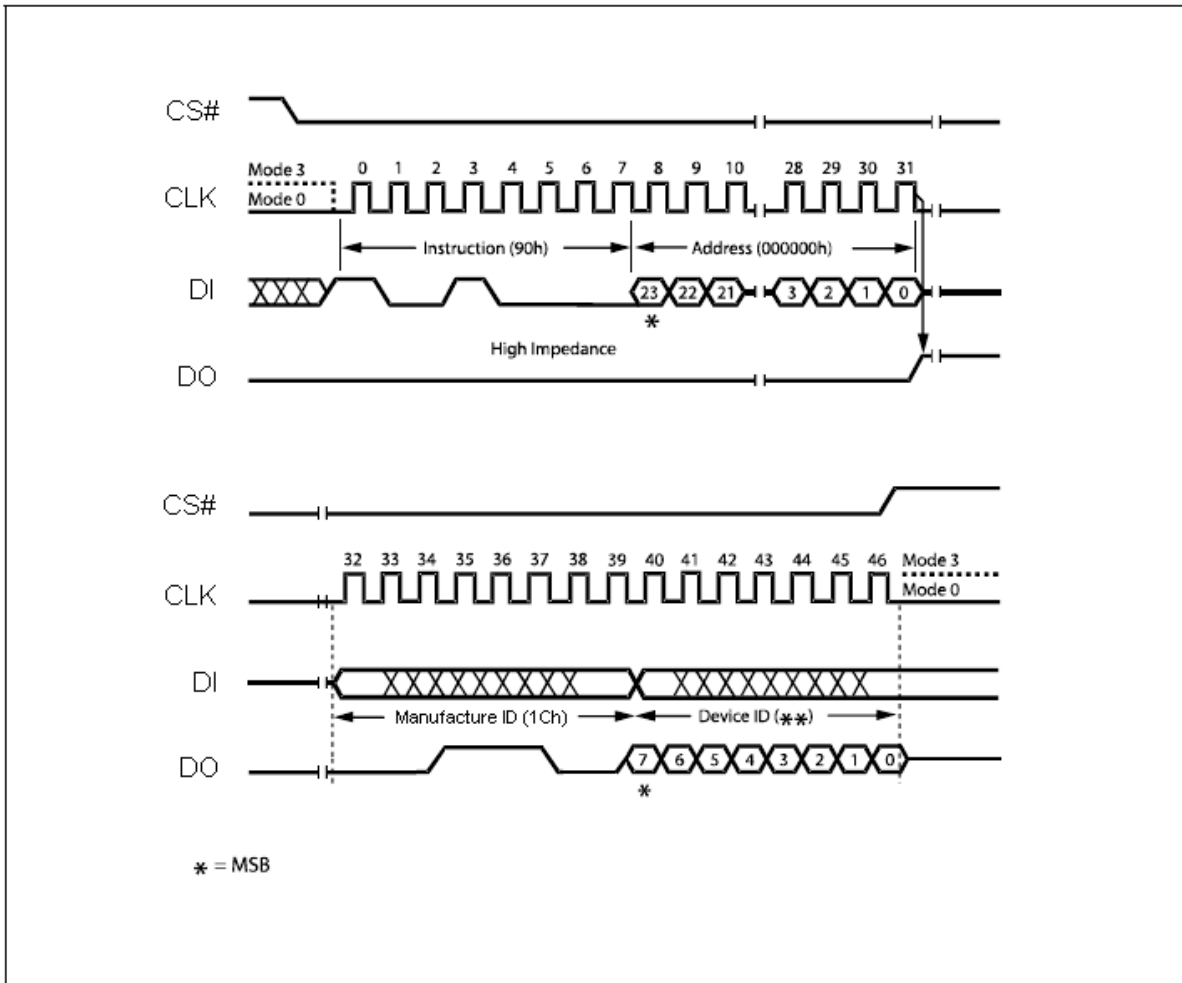
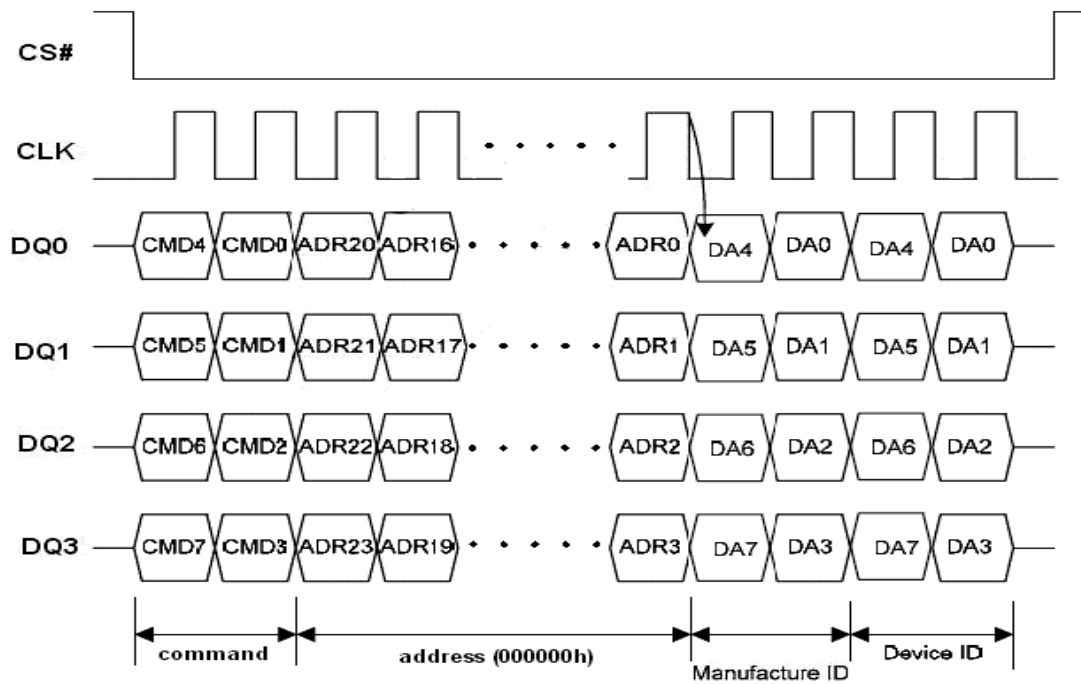
### Read Manufacturer / Device ID (90h)

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

The Read Manufacturer/Device ID instruction is very similar to the Release from Power-down / Device ID instruction. The instruction is initiated by driving the CS# pin low and shifting the instruction code "90h" followed by a 24-bit address (A23-A0) of 000000h. After which, the Manufacturer ID for Eon (1Ch) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Figure 24. The Device ID values for the EN25Q128 are listed in Table 5. If the 24-bit address is initially set to 000001h the Device ID will be read first

The instruction sequence is shown in Figure 24.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.




**Figure 24. Read Manufacturer / Device ID Diagram**

**Figure 24.1. Read Manufacturer / Device ID Diagram under EQPI Mode**

**Read Identification (RDID) (9Fh)**

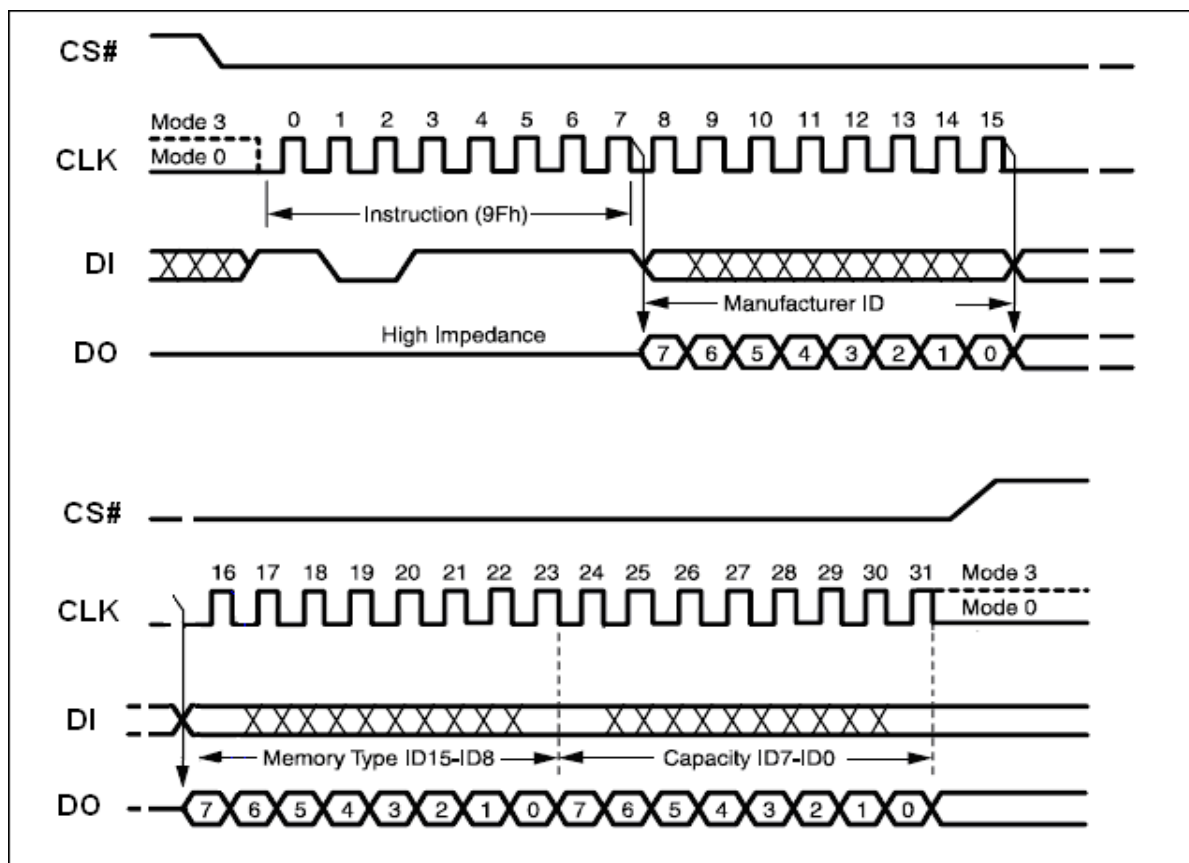
The Read Identification (RDID) instruction allows the 8-bit manufacturer identification to be read, followed by two bytes of device identification. The device identification indicates the memory type in the first byte, and the memory capacity of the device in the second byte.

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.

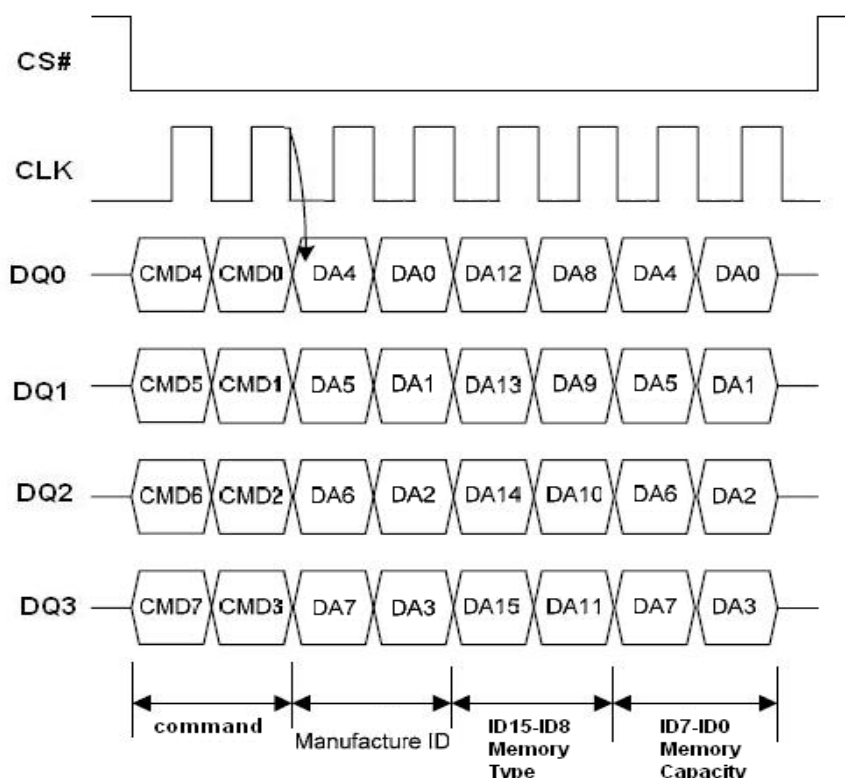
The device is first selected by driving Chip Select Low. Then, the 8-bit instruction code for the instruction is shifted in. This is followed by the 24-bit device identification, stored in the memory, being shifted out on Serial Data Output, each bit being shifted out during the falling edge of Serial Clock. The instruction sequence is shown in Figure 25. 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 25.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.



**Figure 25. Read Identification (RDID)**



**Figure 25.1. Read Identification (RDID) under EQPI Mode**

### Enter OTP Mode (3Ah)

This Flash has an extra 512 bytes OTP sector, user must issue ENTER OTP MODE command to read, program or erase OTP sector. After entering OTP mode, the OTP sector is mapping to sector 4095, **SRP bit** becomes OTP\_LOCK bit and can be read with RDSR command. Program / Erase command will be disabled when OTP\_LOCK bit is '1'

WRSR command will ignore the input data and program OTP\_LOCK bit to 1.

User must clear the protect bits before enter OTP mode.

OTP sector can only be program and erase before OTP\_LOCK bit is set to '1' and BP [3:0] = '0000'. In OTP mode, user can read other sectors, but program/erase other sectors only allowed when OTP\_LOCK bit equal to '0'.

User can use WRDI (04h) command to exit OTP mode.

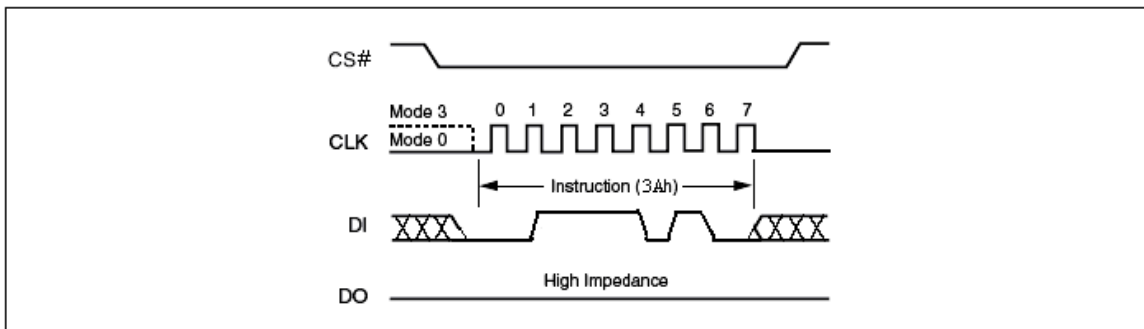
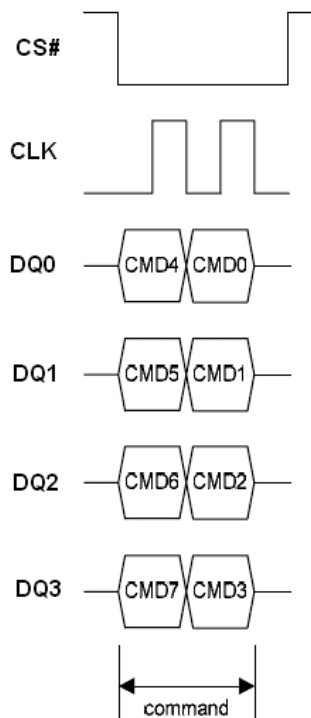
While in OTP mode, user can use Sector Erase (20h) command only to erase OTP data.

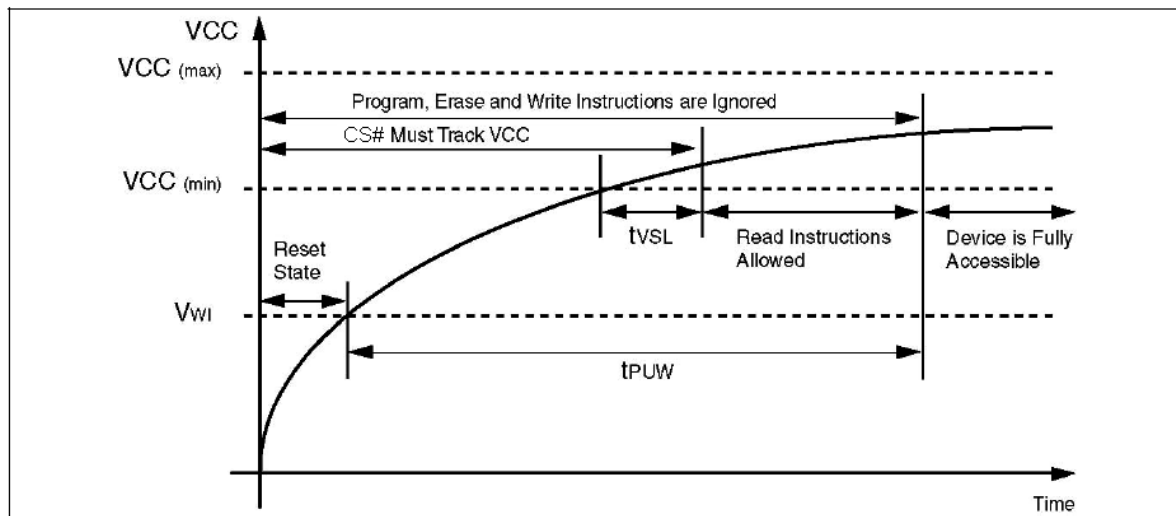
The instruction sequence is shown in Figure 26.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

**Table 7. OTP Sector Address**

Sector	Sector Size	Address Range
4095	512 byte	FFF000h – FFF1FFh

Note: The OTP sector is mapping to sector 4095


**Figure 26. Enter OTP Mode Sequence**

**Figure 26.1 Enter OTP Mode Sequence under EQPI Mode**

**Power-up Timing**

**Figure 27. Power-up Timing**
**Table 8. Power-Up Timing and Write Inhibit Threshold**

Symbol	Parameter	Min.	Max.	Unit
$t_{VSL}^{(1)}$	VCC(min) to CS# low	10		$\mu s$
$t_{PUW}^{(1)}$	Time delay to Write instruction	1	10	ms
$V_{WI}^{(1)}$	Write Inhibit Voltage	1	2.5	V

**Note:**

1. The parameters are characterized only.
2. VCC (max.) is 3.6V and VCC (min.) is 2.7V

**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 9. DC Characteristics**
 $(T_a = -40^{\circ}\text{C to } 85^{\circ}\text{C or } -40^{\circ}\text{C } \sim 125^{\circ}\text{C}; V_{CC} = 2.7\text{-}3.6\text{V})$ 

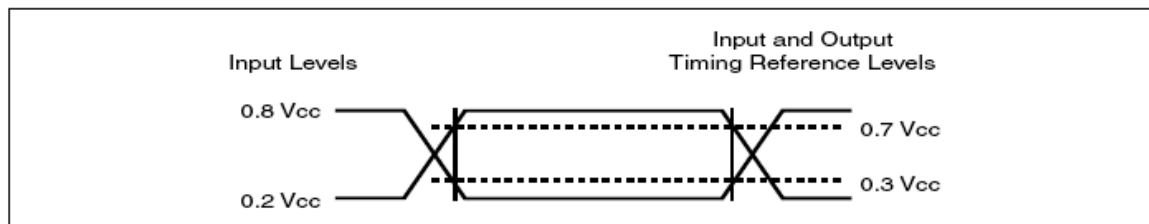
Symbol	Parameter	Test Conditions	Min.	Max.	Unit
$I_{LI}$	Input Leakage Current		-	$\pm 2$	$\mu\text{A}$
$I_{LO}$	Output Leakage Current		-	$\pm 2$	$\mu\text{A}$
$I_{CC1}$	Standby Current	$CS\# = V_{CC}, V_{IN} = V_{SS} \text{ or } V_{CC}$	-	20	$\mu\text{A}$
$I_{CC2}$	Deep Power-down Current	$CS\# = V_{CC}, V_{IN} = V_{SS} \text{ or } V_{CC}$	-	20	$\mu\text{A}$
$I_{CC3}$	Operating Current (READ)	CLK = $0.1 V_{CC} / 0.9 V_{CC}$ at 104MHz, DQ = open	-	25	mA
		CLK = $0.1 V_{CC} / 0.9 V_{CC}$ at 80MHz, DQ = open	-	20	mA
$I_{CC4}$	Operating Current (PP)	$CS\# = V_{CC}$	-	28	mA
$I_{CC5}$	Operating Current (WRSR)	$CS\# = V_{CC}$	-	18	mA
$I_{CC6}$	Operating Current (SE)	$CS\# = V_{CC}$	-	25	mA
$I_{CC7}$	Operating Current (BE)	$CS\# = V_{CC}$	-	25	mA
$V_{IL}$	Input Low Voltage		-0.5	$0.2 V_{CC}$	V
$V_{IH}$	Input High Voltage		$0.7V_{CC}$	$V_{CC}+0.4$	V
$V_{OL}$	Output Low Voltage	$I_{OL} = 1.6 \text{ mA}$	-	0.4	V
$V_{OH}$	Output High Voltage	$I_{OH} = -100 \mu\text{A}$	$V_{CC}-0.2$	-	V

**Table 10. AC Measurement Conditions**

Symbol	Parameter	Min.	Max.	Unit
$C_L$	Load Capacitance	20/30		pF
	Input Rise and Fall Times	5		ns
	Input Pulse Voltages	$0.2V_{CC}$ to $0.8V_{CC}$		V
	Input Timing Reference Voltages	$0.3V_{CC}$ to $0.7V_{CC}$		V
	Output Timing Reference Voltages	$V_{CC} / 2$		V

**Notes:**

- $C_L = 20 \text{ pF}$  when CLK=104MHz,  $C_L = 30 \text{ pF}$  when CLK = 80MHz,


**Figure 28. AC Measurement I/O Waveform**



**Table 11. AC Characteristics**

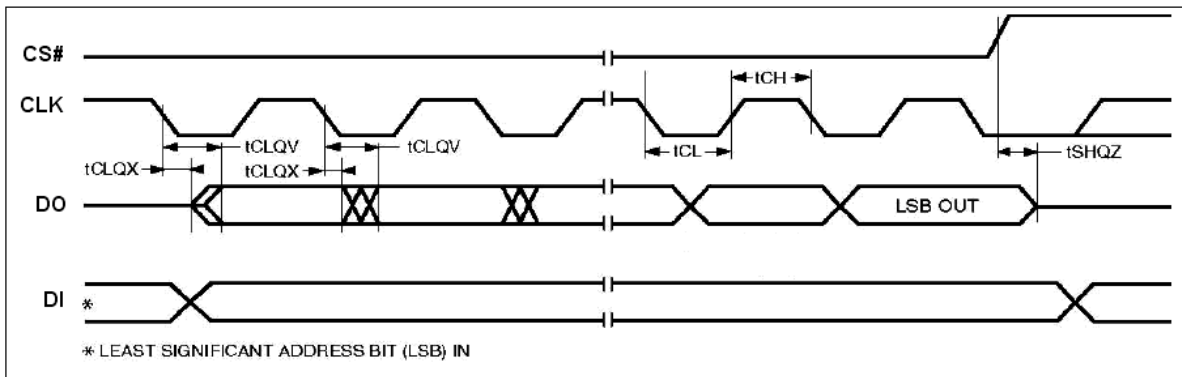
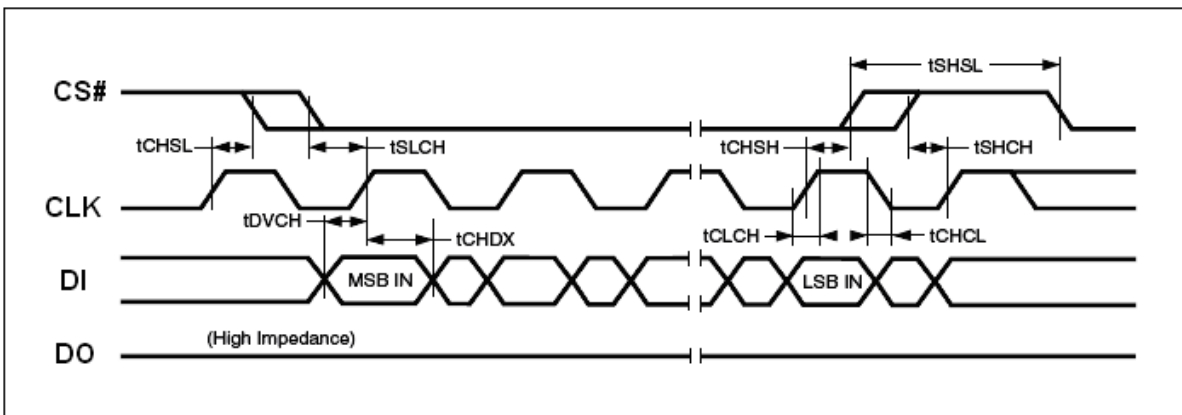
( $T_a = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$  or  $-40^{\circ}\text{C} \sim 125^{\circ}\text{C}$ ;  $V_{CC} = 2.7\text{-}3.6\text{V}$ )

Symbol	Alt	Parameter	Min	Typ	Max	Unit	
$F_R$	$f_C$	Serial Clock Frequency for: FAST_READ, PP, SE, BE, DP, RES, WREN, WRDI, WRSR	D.C.	-	104	MHz	
		Serial Clock Frequency for: RDSR, RDID, Dual Output Fast Read	D.C.	-	80	MHz	
$f_R$		Serial Clock Frequency for READ, Quad I/O Fast Read	D.C.	-	50	MHz	
$t_{CH}^1$		Serial Clock High Time	4	-	-	ns	
$t_{CL}^1$		Serial Clock Low Time	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 for read	15	-	-	ns	
		CS# High Time for program/erase	50	-	-	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	
$t_{CLQV}$	$t_V$	Output Valid from CLK	-	-	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	$\mu\text{s}$	
$t_{RES1}^2$		CS# High to Standby Mode without Electronic Signature read	-	-	3	$\mu\text{s}$	
$t_{RES2}^2$		CS# High to Standby Mode with Electronic Signature read	-	-	1.8	$\mu\text{s}$	
$t_W$		Write Status Register Cycle Time	-	15	50	ms	
$t_{PP}$		Page Programming Time	-	0.8	5	ms	
$t_{SE}$		Sector Erase Time	-	0.05	0.3	s	
$t_{BE}$		Block Erase Time	-	0.2	2	s	
$t_{CE}$		Chip Erase Time	-	45	140	s	
$t_{SR}$		Software Reset Latency	WIP = write operation	-	-	28	$\mu\text{s}$
			WIP = not in write operation	-	-	0	$\mu\text{s}$

**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 29. Serial Output Timing**

**Figure 30. Input Timing**



## ABSOLUTE MAXIMUM RATINGS

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

Parameter	Value	Unit
Storage Temperature	-65 to +150	°C
Plastic Packages	-65 to +125	°C
Output Short Circuit Current <sup>1</sup>	200	mA
Input and Output Voltage (with respect to ground) <sup>2</sup>	-0.5 to +4.0	V
V <sub>cc</sub>	-0.5 to +4.0	V

Notes:

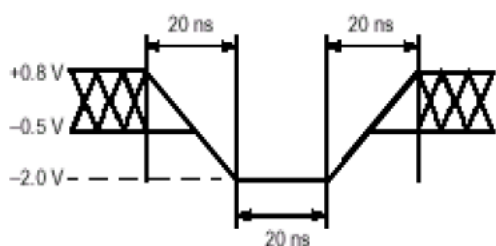
- No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.
- Minimum DC voltage on input or I/O pins is -0.5 V. During voltage transitions, inputs may undershoot V<sub>ss</sub> to -1.0V for periods of up to 50ns and to -2.0 V for periods of up to 20ns. See figure below. Maximum DC voltage on output and I/O pins is V<sub>cc</sub> + 0.5 V. During voltage transitions, outputs may overshoot to V<sub>cc</sub> + 1.5 V for periods up to 20ns. See figure below.

## RECOMMENDED OPERATING RANGES <sup>1</sup>

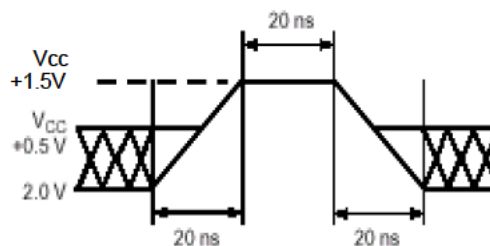
Parameter	Value	Unit
Ambient Operating Temperature Industrial Devices Automotive Devices	-40 to 85 -40 to 125	°C
Operating Supply Voltage V <sub>cc</sub>	Full: 2.7 to 3.6	V

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 12. DATA RETENTION and ENDURANCE

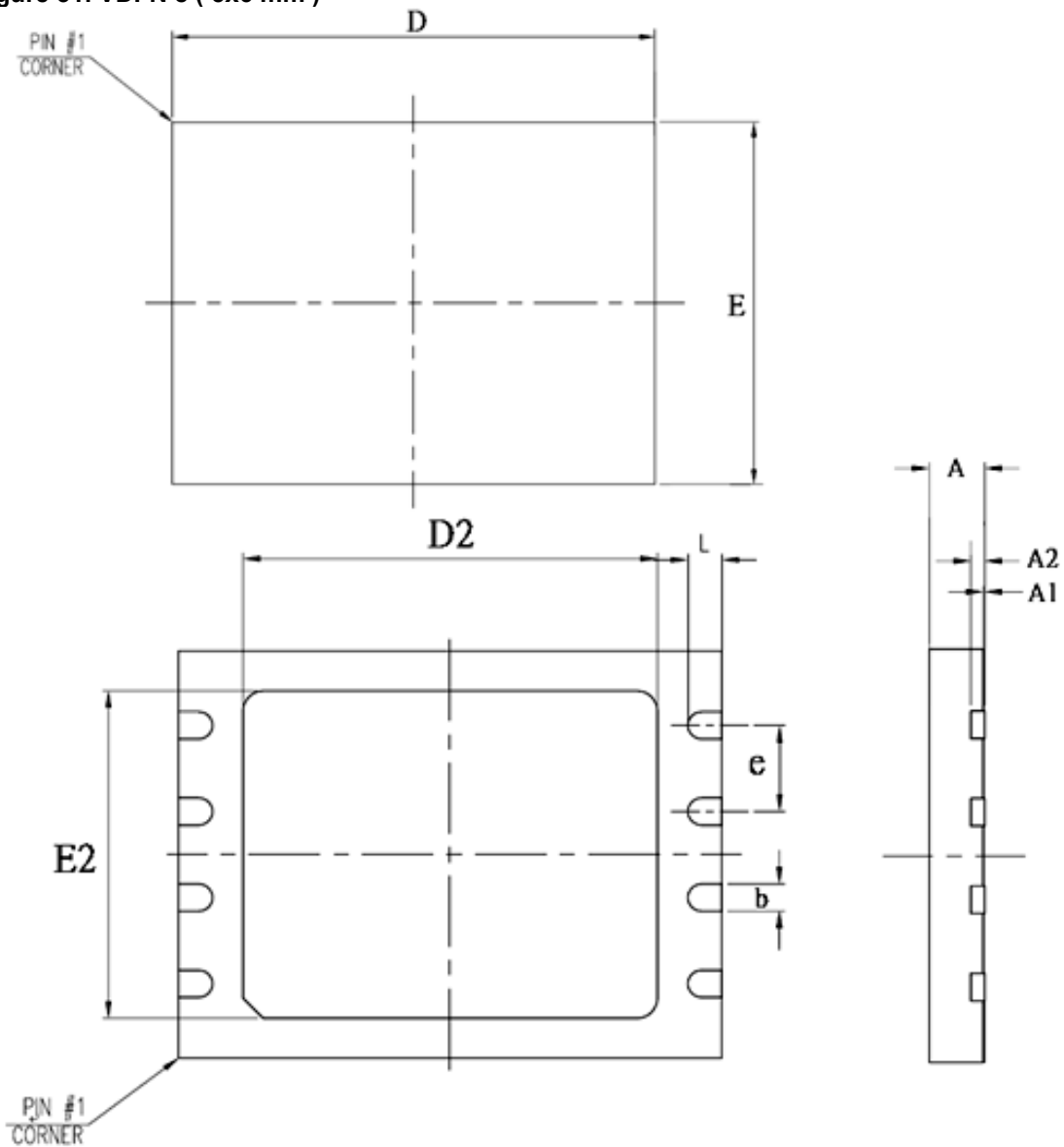
Parameter Description	Test Conditions	Min	Unit
Data Retention Time	150°C	10	Years
	125°C	20	Years
Erase/Program Endurance	-40 to 85 °C	100k	cycles

Table 13. CAPACITANCE

 $(V_{CC} = 2.7-3.6V)$ 

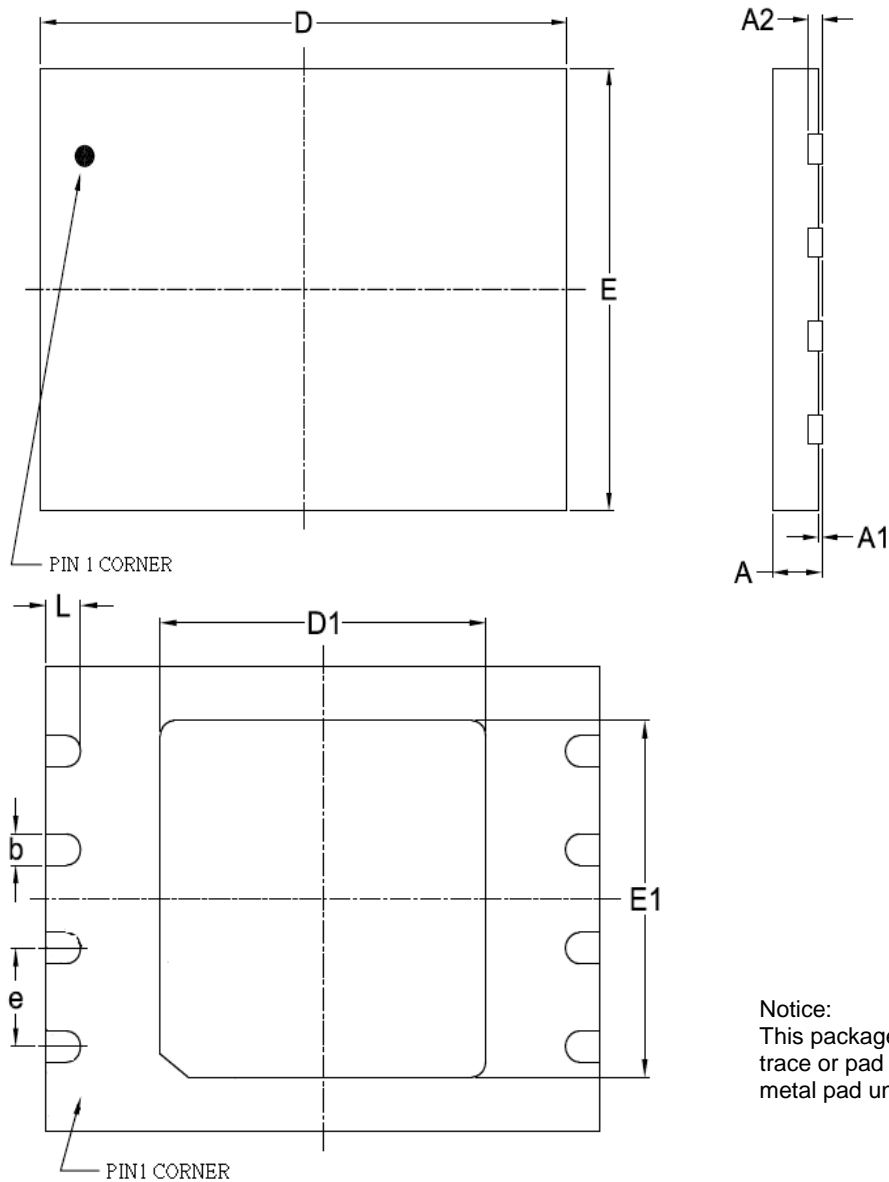
Parameter Symbol	Parameter Description	Test Setup	Max	Unit
C <sub>IN</sub>	Input Capacitance	V <sub>IN</sub> = 0	6	pF
C <sub>OUT</sub>	Output Capacitance	V <sub>OUT</sub> = 0	8	pF

**Note :** Sampled only, not 100% tested, at T<sub>A</sub> = 25°C and a frequency of 20MHz.

**PACKAGE MECHANICAL**
**Figure 31. 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

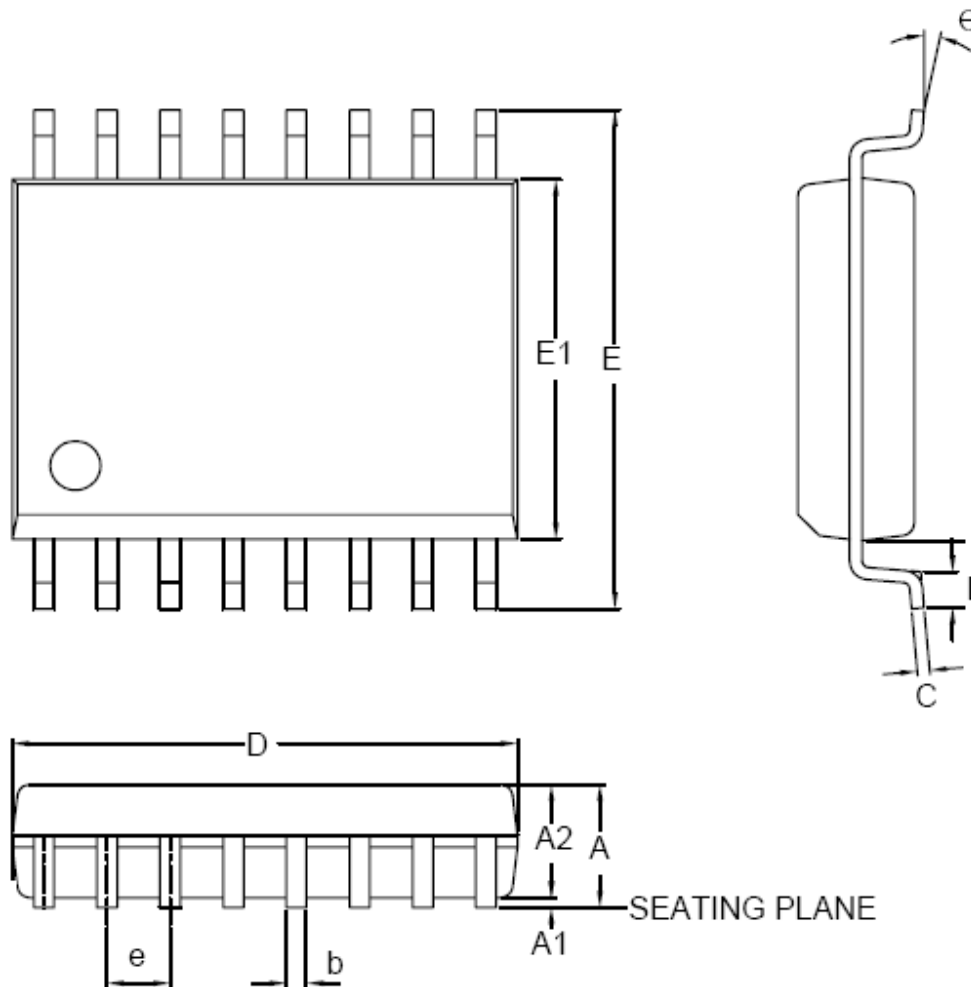
**Note : 1. Coplanarity: 0.1 mm**

**Figure 32. VDFN 8 ( 6x8 mm )**


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

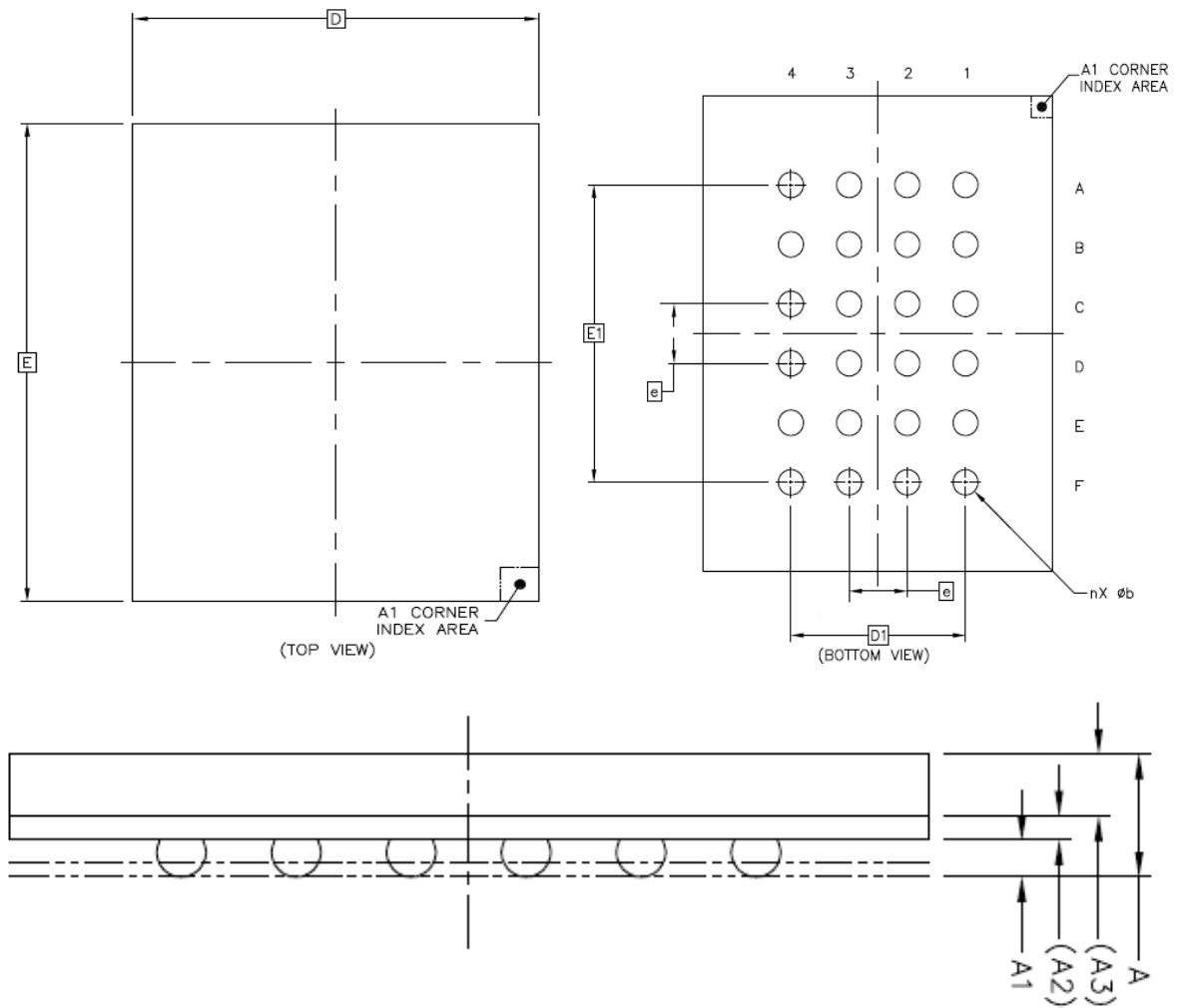
SYMBOL	DIMENSION IN MM		
	MIN.	NOR	MAX
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A2	---	0.20	---
D	7.90	8.00	8.10
E	5.90	6.00	6.10
D1	4.65	4.70	4.75
E1	4.55	4.60	4.65
e	---	1.27	---
b	0.35	0.40	0.48
L	0.4	0.50	0.60

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

**Figure 33. 16 LEAD SOP 300 mil**


SYMBOL	DIMENSION IN MM		
	MIN.	NOR	MAX
A	---	---	2.65
A1	0.10	0.20	0.30
A2	2.25	---	2.40
C	0.20	0.25	0.30
D	10.10	10.30	10.50
E	10.00	---	10.65
E1	7.40	7.50	7.60
e	---	1.27	---
b	0.31	---	0.51
L	0.4	---	1.27
θ	0°	5°	8°

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

**Figure 34. 24-ball Thin Profile Fine-Pitch Ball Grid Array (6 x 8 mm) Package**


SYMBOL	DIMENSION IN MM		
	MIN	NCR	MAX
A	---	---	1.20
A1	0.27	---	0.37
A2	0.21 REF		
A3	0.54 REF		
D	6 BSC		
E	8 BSC		
D1	---	300	---
E1	---	500	---
e	---	1.00	---
b	---	0.40	---



## Purpose

Eon Silicon Solution Inc. (hereinafter called "Eon") is going to provide its products' top marking on ICs with < cFeon > from January 1<sup>st</sup>, 2009, and without any change of the part number and the compositions of the Ics. Eon is still keeping the promise of quality for all the products with the same as that of Eon delivered before. Please be advised with the change and appreciate your kindly cooperation and fully support Eon's product family.

## Eon products' Top Marking

### cFeon Top Marking Example:

**cFeon**

**Part Number: XXXX-XXX**

**Lot Number: XXXXX**

**Date Code: XXXXX**

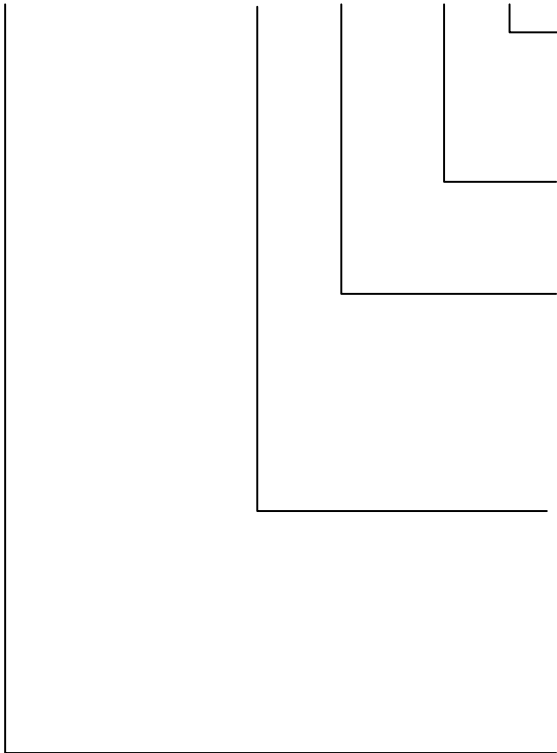
## For More Information

Please contact your local sales office for additional information about Eon memory solutions.



**ORDERING INFORMATION**

EN25Q128 - 104 F I P



**PACKAGING CONTENT**  
P = RoHS compliant

**TEMPERATURE RANGE**  
I = Industrial (-40°C to +85°C)  
A = Automotive (-40°C to +125°C)

**PACKAGE**  
W = 8-pin VDFN (5x6mm)  
Y = 8-pin VDFN (6x8mm)  
F = 16-pin 300mil SOP  
BB = 24-ball Ball Grid Array (6 x 8 mm)

**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  
128 = 128 Megabit (16,384K x 8)





## Revisions List

Revision No	Description	Date
A	Initial Release	2009/07/13
B	<ol style="list-style-type: none"><li>Update Page programming, Sector, Block and Chip erase time (typ.) and (max.) parameter on page 1 and 50.<ol style="list-style-type: none"><li>Page programming: from 1.3ms to 0.8ms (typ.) , from 5ms to 3ms (max.)</li><li>Sector erase: from 0.06s to 0.05s (typ.) , from 0.3s to 0.25s (max.)</li><li>Block erase: from 0.25s to 0.2s (typ.) , from 2s to 1.6s (max.)</li><li>Chip erase: from 50s to 45s (typ.) , from 100s to 90s (max.)</li></ol></li><li>Add Figure 23. Write Suspend/Resume Flow on page 39</li></ol>	2009/09/03
C	<ol style="list-style-type: none"><li>Add VDFN 8 ( 6 x 8 mm ) dimension on page 55.</li><li>Modify Table 10. DC Characteristics I<sub>CC1</sub> (Standby) and I<sub>CC2</sub> (Deep Power-down) Current from 5<math>\mu</math>A to 20<math>\mu</math>A on page 49.</li></ol>	2009/10/21
D	<ol style="list-style-type: none"><li>Add 24-ball Ball Grid Array (6 x 8 mm) package option.</li><li>Remove Write Suspend and Write Resume information.</li><li>Modify Software Reset Latency from 20<math>\mu</math>s to 28<math>\mu</math>s on page 47.</li></ol>	2010/02/04
E	<ol style="list-style-type: none"><li>Update AC Characteristics in Table 11 on page 47.<ol style="list-style-type: none"><li>Page Programming time from 3ms to 5ms (max.)</li><li>Sector Erase time from 0.25s to 0.3s (max.)</li><li>Block Erase time from 1.6s to 2s (max.)</li></ol></li></ol>	2010/04/19
F	<ol style="list-style-type: none"><li>Rename 38h command from Enable Quad I/O (EQIO) to Enable Quad Peripheral Interface mode (EQPI).</li><li>Revise the speed of Quad SPI from 80MHz to 50MHz.</li></ol>	2011/01/10
G	<ol style="list-style-type: none"><li>Update Write Status Register Cycle Time from 10 (typ.) /15 (max.) ms to 15 (typ.) / 50 (max.) ms on page 47.</li><li>Remove the package option of 8 contact VDFN (5x6mm).</li><li>Rename 24 Ball package from BGA to TFBGA.</li></ol>	2011/04/19
H	<ol style="list-style-type: none"><li>Add the note "5. This flow cannot release the device from Deep power down mode." on page 19.</li><li>Correct the typo of 6 dummy clocks for EBh command on page 29.</li><li>Update Chip Erase Time (max.) from 90s to 140s on page 47.</li></ol>	2011/07/01
I	Add Automotive (-40°C to +125°C) grade temperature option.	2011/09/06
J	Add VDFN 8 ( 5 x 6 mm ) package option.	2011/09/19