EN35SXR128A (2PC)
128 Megabit 1.8V Serial Flash Memory with 4Kbyte Uniform Sector

FEATURES

- Single power supply operation
  - Full voltage range: 1.65-1.95 volt
- Serial Interface Architecture
  - SPI Compatible: Mode 0 and Mode 3
- 128 M-bit Serial Flash
  - 128 M-bit / 16,384 KByte /65,536 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO
  - Dual SPI: CLK, CS#, DQ_0, DQ_1
  - Quad SPI: CLK, CS#, DQ_0, DQ_1, DQ_2, DQ_3
  - Default QE=1(Quad Enable), WP#, HOLD# disable
- High performance
  - 1.65-1.95V
    - 104 MHz clock rate for Single/Dual/Quad I/O Fast Read
- Low power consumption
  - 8 mA typical active current
  - 0.1 μA typical power down current
- Uniform Sector Architecture:
  - 4,096 sectors of 4-Kbyte
  - 512 blocks of 32-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
- High performance program/erase speed
  - Page program time: 0.5 ms typical
  - Sector erase time: 40 ms typical
  - Half Block erase time 200 ms typical
  - Block erase time 300 ms typical
  - Chip erase time: 60 seconds typical
- Write Suspend and Write Resume
- Volatile Status Register Bits
- Lockable 3x512 byte OTP security sector
- Support Serial Flash Discoverable Parameters (SFDP) signature
- Read Unique ID Number
- Minimum 100K endurance cycle
- Data retention time 20 years
- Replay-Protected Monotonic Counter (RPMC)
- Package Options
  - 8-pin SOP 200 mil body width
  - 16-pin SOP 300 mil body width
  - 8-contact VDFN / WSON (6x5 mm)
  - 8-contact VDFN / WSON (8x6 mm)
  - All Pb-free packages are compliant RoHS, Halogen-Free and REACH.
- Industrial temperature Range

GENERAL DESCRIPTION

The device is a 128 Megabit (16,384K-byte) Serial Flash memory, with advanced write protection mechanisms. The device supports the single bit and four bits serial input and output commands via standard Serial Peripheral Interface (SPI) pins: Serial Clock, Chip Select, Serial DQ_0 (DI) and DQ_1 (DO), DQ_2 (WP#) and DQ_3 (HOLD#/RESET#). The memory can be programmed 1 to 256 bytes at a time, using the Page Program instruction.

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

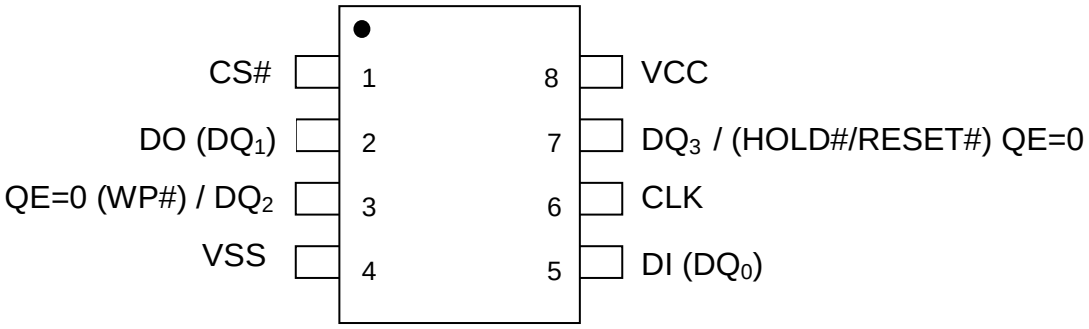
The device is designed to allow either single Sector/Block at a time or full chip erase operation. The device can be configured to protect part of the memory as the software protected mode. The device can sustain a minimum of 100K program/erase cycles on each sector or block.

Contents

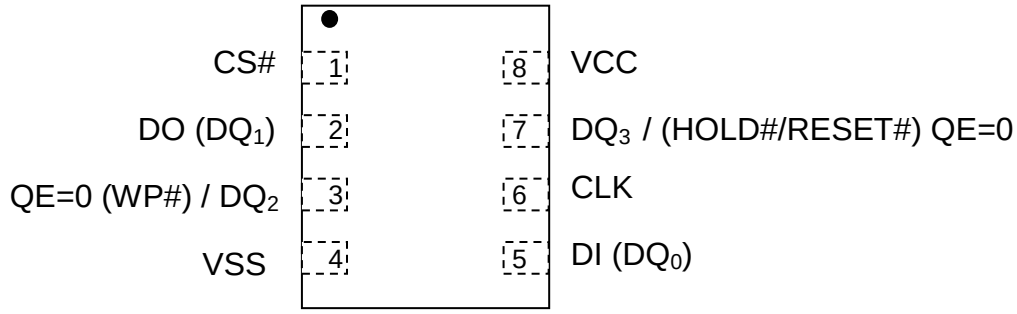
List of Figures

List of Tables

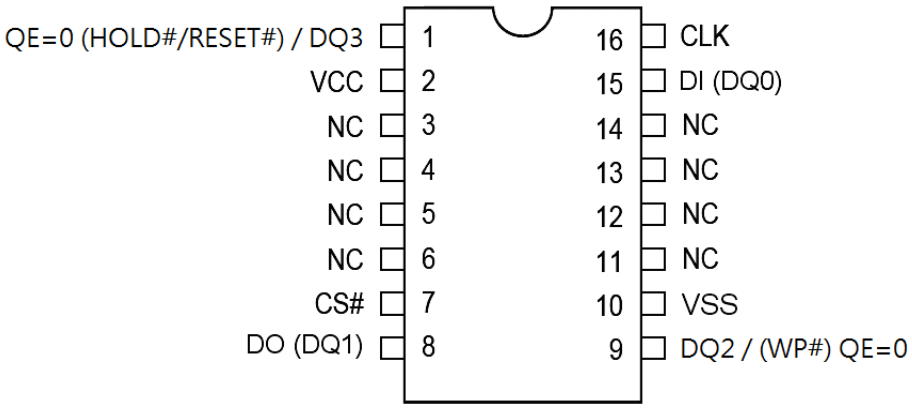
CONNECTION DIAGRAMS (TOP VIEW)



8 - LEAD SOP



8 - LEAD VDFN / WSON



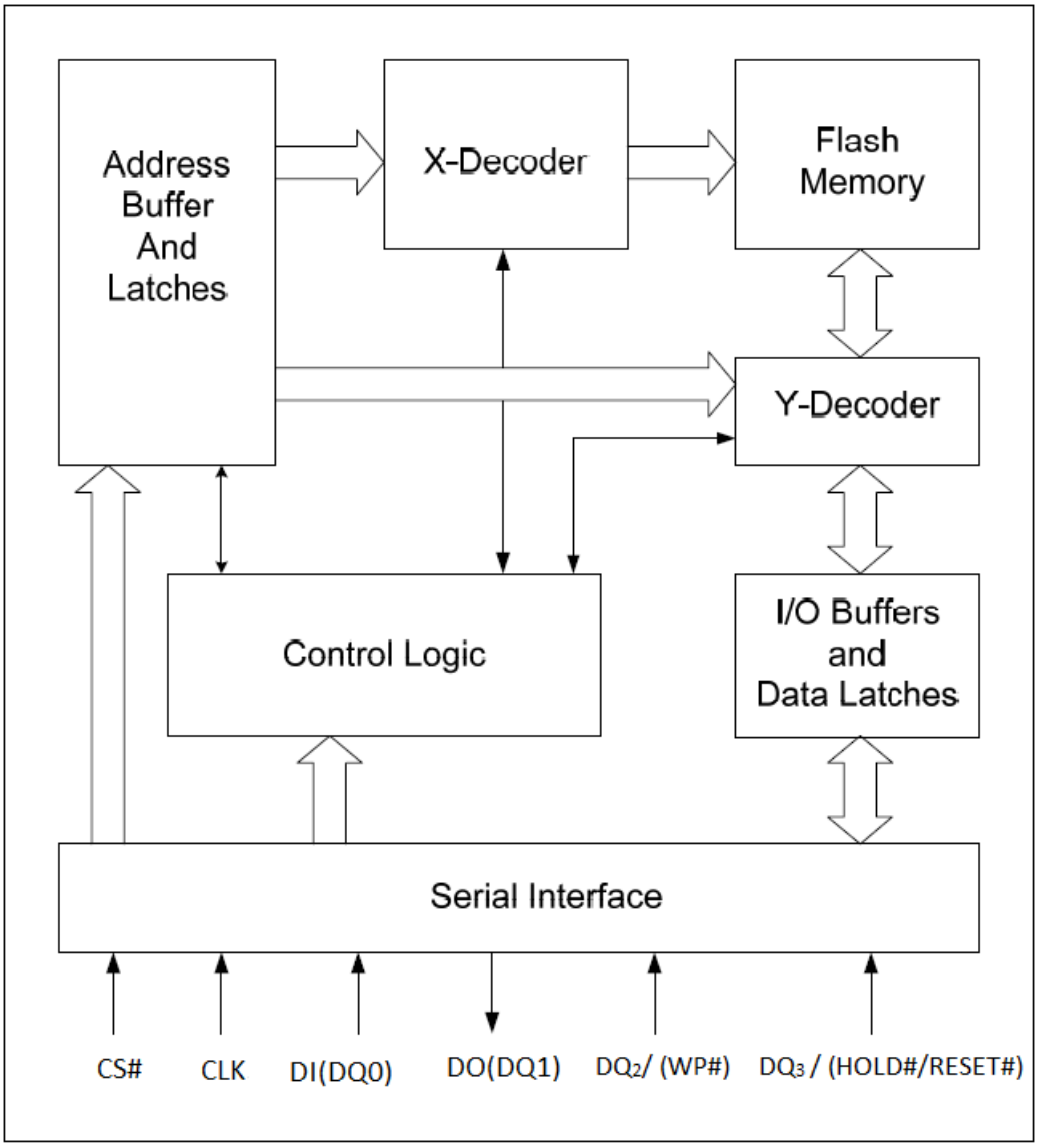
16 - LEAD SOP

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 Enable
DQ_2 / (WP#)	Data Input Output 2 / (Write Protect pin) *2
DQ_3 / (HOLD#/RESET#)	Data Input Output 3 / (HOLD#/RESET# pin) *2
V_{CC}	Supply Voltage (1.65-1.95V)
V_{SS}	Ground
NC	No Connect

Note:
1. DQ_0 and DQ_1 are used for Dual and Quad instructions.
2. DQ_0 ~ DQ_3 are used for Quad instructions.
   Default Quad Output, WP# & HOLD#/RESET# functions can be available by setting QE=0 for Standard/Dual SPI mode.

Figure 1. BLOCK DIAGRAM



Note:

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

SIGNAL DESCRIPTION

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

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

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

Serial Clock (CLK)

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

Chip Select (CS#)

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

Write Protect (WP#)

The device default setting is WP# disable by QE=1.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 (CMP, 4KBL, TB, BP2, BP1 and BP0) bits and Status Register Protect (SRP) bits, a portion or the entire memory array can be hardware protected. The WP# function is only available after setting QE=0 for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO (DQ2) for Quad I/O operation.

HOLD (HOLD#)

The device default setting is HOLD# disable by QE=1. The HOLD# pin allows the device to be paused while it is actively selected. When QE bit is "0" and HRSW bit is "0" (factory default), the HOLD# pin is enabled. When HOLD# is brought low, while CS# is low, the DO pin will be at high impedance and signals on the DI and CLK pins will be ignored (don't care). The hold function can be useful when multiple devices are sharing the same SPI signals. The HOLD# function is only available after setting QE=0 for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO (DQ3) for Quad I/O operation.

RESET (RESET#)

The device default setting is RESET# disable by QE=1. The RESET# pin allows the device to be reset by the controller. When QE bit is "0" and HRSW bit is '1' (factory default is '0'), the RESET# pin is enabled. The Hardware Reset function is only available after setting QE=0 for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO (DQ3) for Quad I/O operation. Set RESET# to low for a minimum period 1us (tHRST) will interrupt any on-going instructions to have the device to initial state. The device can accept new instructions again in 28us (tHRSL) after RESET# back to high.

MEMORY ORGANIZATION

The memory is organized as:

- 16,777,216 bytes
- Uniform Sector Architecture
  256 blocks of 64-Kbyte
  512 sectors of 32-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

64KB Block	32KB Block	Sector	Address range	
255	511	4095	FFF000h	FFFFFFh
	⋮	⋮	⋮	⋮
	510	4080	FF0000h	FF0FFFh
254	509	4079	FEF000h	FEFFFFh
	⋮	⋮	⋮	⋮
	508	4064	FE0000h	FE0FFFh
253	507	4063	FDF000h	FDFFFFh
	⋮	⋮	⋮	⋮
	506	4048	FD0000h	FD0FFFh
252	505	4047	FCF000h	FCFFFFh
	⋮	⋮	⋮	⋮
	504	4032	FC0000h	FC0FFFh
251	503	4031	FBF000h	FBFFFFh
	⋮	⋮	⋮	⋮
	502	4016	FB0000h	FB0FFFh
250	501	4015	FAF000h	FAFFFFh
	⋮	⋮	⋮	⋮
	500	4000	FA0000h	FA0FFFh
249	499	3999	F9F000h	F9FFFFh
	⋮	⋮	⋮	⋮
	498	3984	F90000h	F90FFFh
248	497	3983	F8F000h	F8FFFFh
	⋮	⋮	⋮	⋮
	496	3968	F80000h	F80FFFh
247	495	3967	F7F000h	F7FFFFh
	⋮	⋮	⋮	⋮
	494	3952	F70000h	F70FFFh
246	493	3951	F6F000h	F6FFFFh
	⋮	⋮	⋮	⋮
	492	3936	F60000h	F60FFFh
245	491	3935	F5F000h	F5FFFFh
	⋮	⋮	⋮	⋮
	490	3920	F50000h	F50FFFh
244	489	3919	F4F000h	F4FFFFh
	⋮	⋮	⋮	⋮
	488	3904	F40000h	F40FFFh
243	487	3903	F3F000h	F3FFFFh
	⋮	⋮	⋮	⋮
	486	3888	F30000h	F30FFFh
242	485	3887	F2F000h	F2FFFFh
	⋮	⋮	⋮	⋮
	484	3872	F20000h	F20FFFh
241	483	3871	F1F000h	F1FFFFh
	⋮	⋮	⋮	⋮
	482	3856	F10000h	F10FFFh
⋮	⋮	⋮	⋮	⋮

Table 2. Uniform Block Sector Architecture (Continued)

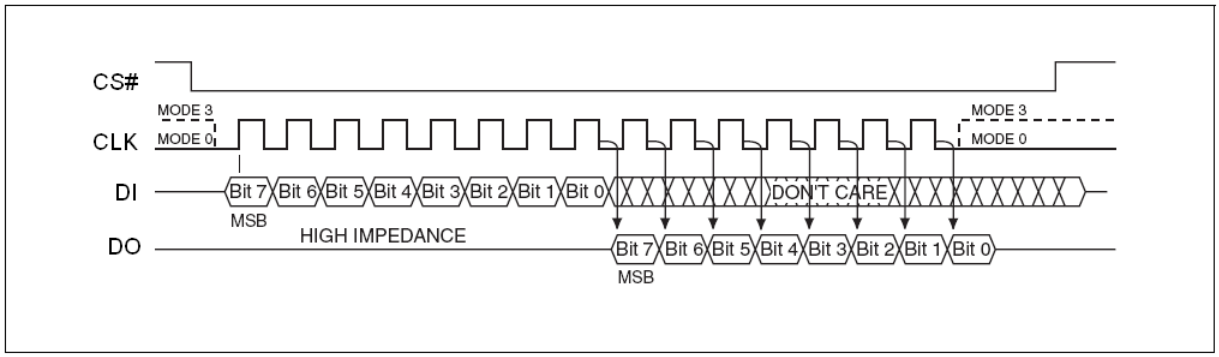
64KB Block	32KB Block	Sector	Address range	
13	27	223	0DF000h	0DFFFFh
	⋮	⋮	⋮	⋮
	26	208	0D0000h	0D0FFFh
12	25	207	0CF000h	0CFFFFh
	⋮	⋮	⋮	⋮
	24	192	0C0000h	0C0FFFh
11	13	191	0BF000h	0BFFFFh
	⋮	⋮	⋮	⋮
	22	176	0B0000h	0B0FFFh
10	21	175	0AF000h	0AFFFFh
	⋮	⋮	⋮	⋮
	20	160	0A0000h	0A0FFFh
9	19	159	09F000h	09FFFFh
	⋮	⋮	⋮	⋮
	18	144	090000h	090FFFh
8	17	143	08F000h	08FFFFh
	⋮	⋮	⋮	⋮
	16	128	080000h	080FFFh
7	15	127	07F000h	07FFFFh
	⋮	⋮	⋮	⋮
	14	112	070000h	070FFFh
6	13	111	06F000h	06FFFFh
	⋮	⋮	⋮	⋮
	12	96	060000h	060FFFh
5	11	95	05F000h	05FFFFh
	⋮	⋮	⋮	⋮
	10	80	050000h	050FFFh
4	9	79	04F000h	04FFFFh
	⋮	⋮	⋮	⋮
	8	64	040000h	040FFFh
3	7	63	03F000h	03FFFFh
	⋮	⋮	⋮	⋮
	6	48	030000h	030FFFh
2	5	47	02F000h	02FFFFh
	⋮	⋮	⋮	⋮
	4	32	020000h	020FFFh
1	3	31	01F000h	01FFFFh
	⋮	⋮	⋮	⋮
	2	16	010000h	010FFFh
0	1	15	00F000h	00FFFFh
	⋮	⋮	⋮	⋮
	0	0	000000h	000FFFh

OPERATING FEATURES

Standard SPI Modes

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

Figure 2. SPI Modes



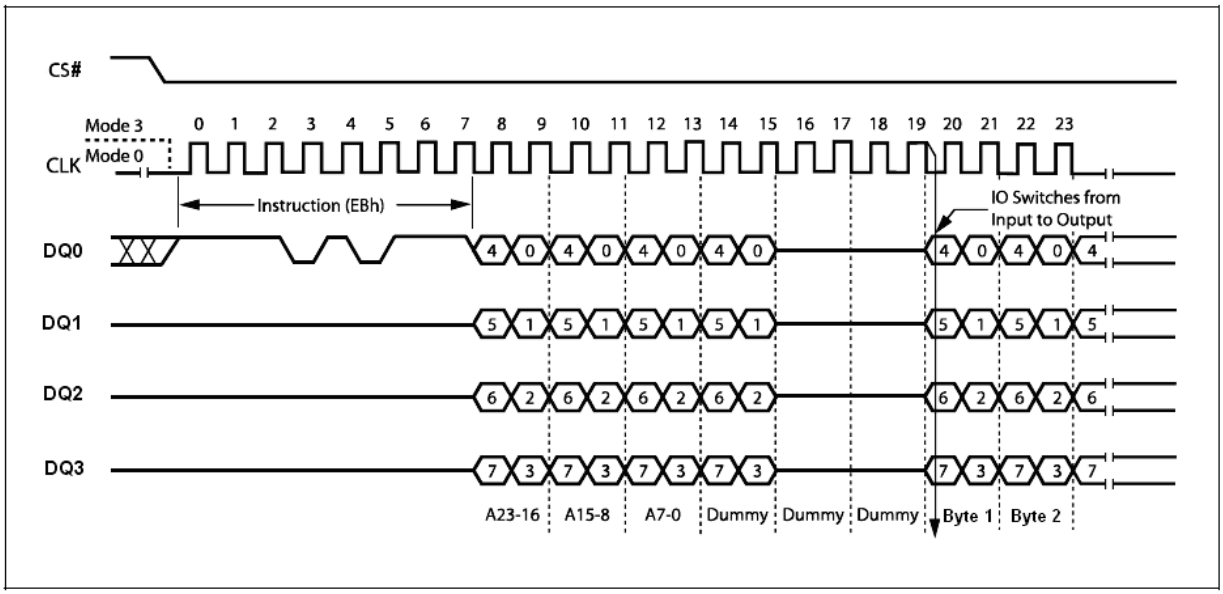
Dual SPI Instruction

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

Quad I/O SPI Modes

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

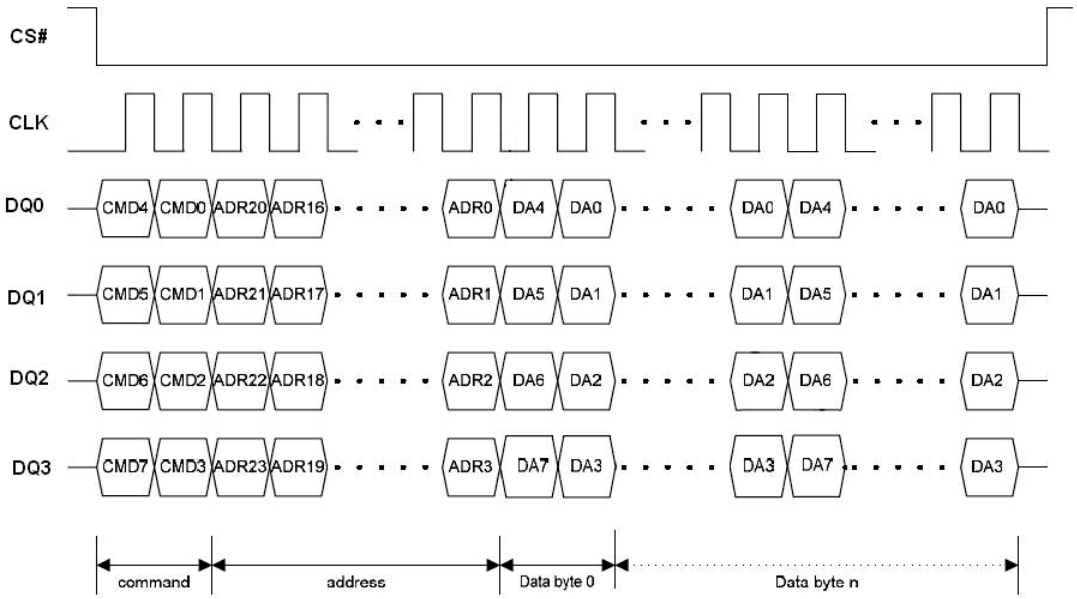
Figure 3. Quad I/O SPI Modes



Full Quad SPI Modes (QPI)

The device 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; DQ0 and DQ1, and the WP# and HOLD#/RESET# pins become DQ2 and DQ3 respectively.

Figure 4. 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, WRSR2, WRSR3), Program (PP, QPP) or Erase (SE, HBE, BE or CE) can be achieved by not waiting for the worst case delay (t_W, t_{PP}, t_{SE}, t_{HBE}, t_{BE} or t_{CE}). The Write In Progress (WIP) bit is provided in the Status Register so that the application program can monitor its value, polling it to establish when the previous Write cycle, Program cycle or Erase cycle is complete.

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

When Chip Select (CS#) is Low, the device is enabled, and in the Active Power mode. When Chip Select (CS#) is High, the device is disabled, but could remain in the Active Power mode until all internal cycles have completed (Program, Erase, Write Status Register). The device then goes into the Stand-by Power mode. The device consumption drops to I_{CC1}.
The Deep Power-down mode is entered when the specific instruction (the Enter Deep Power-down Mode (DP) instruction) is executed. The device consumption drops further to I_{CC2}. The device remains in this mode until another specific instruction (the Release from Deep Power-down Mode and Read Device ID (RDI) instruction) is executed.
All other instructions are ignored while the device is in the Deep Power-down mode. This can be used as an extra software protection mechanism, when the device is not in active use, to protect the device from inadvertent Write, Program or Erase instructions.

Write Protection

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

- Power-On Reset and an internal timer can provide protection against inadvertent changes while the power supply is outside the operating specification.
- Program, Erase and Write Status Register instructions are checked that they consist of a number of clock pulses that is a multiple of eight, before they are accepted for execution.
- All instructions that modify data must be preceded by a Write Enable (WREN) instruction to set the Write Enable Latch (WEL) bit. This bit is returned to its reset state by the following events:
  - Power-up
  - Write Disable (WRDI) instruction completion or Write Status Register (WRSR) instruction completion, Write Status Register 2 or Write Status Register 3 (WRSR3) instruction completion or Page Program (PP) or Quad Input Page Program (QPP) instruction completion or Sector Erase (SE) instruction completion or Half Block Erase (HBE) / Block Erase (BE) instruction completion or Chip Erase (CE) instruction completion
- The Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits allow part of the memory to be configured as read-only. This is the Software Protected Mode (SPM).
- The Write Protect (WP#) signal allows the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits and Status Register Protect (SRP) 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			
CMP	4KBL	T/B	SR.4	SR.3	SR.2	Protect Areas	Addresses	Density(KB)	Portion
Bit	Bit	Bit	Bit	Bit	Bit				
0	0	0	0	0	0	None	None	None	None
0	0	0	0	0	1	Block 252 to 255	FC0000h-FFFFFFh	256KB	Upper 1/64
0	0	0	0	1	0	Block 248 to 255	F80000h-FFFFFFh	512KB	Upper 1/32
0	0	0	0	1	1	Block 240 to 255	F00000h-FFFFFFh	1024KB	Upper 1/16
0	0	0	1	0	0	Block 224 to 255	E00000h-FFFFFFh	2048KB	Upper 1/8
0	0	0	1	0	1	Block 192 to 255	C00000h-FFFFFFh	4096KB	Upper 1/4
0	0	0	1	1	0	Block 128 to 255	800000h-FFFFFFh	8192KB	Upper 1/2
0	0	0	1	1	1	Block 0 to 255	000000h-FFFFFFh	16384KB	All
0	0	1	0	0	0	None	None	None	None
0	0	1	0	0	1	Block 0 to 3	000000h-03FFFFh	256KB	Lower 1/64
0	0	1	0	1	0	Block 0 to 7	000000h-07FFFFh	512KB	Lower 1/32
0	0	1	0	1	1	Block 0 to 15	000000h-0FFFFFh	1024KB	Lower 1/16
0	0	1	1	0	0	Block 0 to 31	000000h-1FFFFFh	2048KB	Lower 1/8
0	0	1	1	0	1	Block 0 to 63	000000h-3FFFFFh	4096KB	Lower 1/4
0	0	1	1	1	0	Block 0 to 127	000000h-7FFFFFh	8192KB	Lower 1/2
0	0	1	1	1	1	Block 0 to 255	000000h-FFFFFFh	16384KB	All
0	1	0	0	0	0	None	None	None	None
0	1	0	0	0	1	Block 255	FFF000h-FFFFFFh	4KB	Upper 1/4096
0	1	0	0	1	0	Block 255	FFE000h-FFFFFFh	8KB	Upper 1/2048
0	1	0	0	1	1	Block 255	FFC000h-FFFFFFh	16KB	Upper 1/1024
0	1	0	1	0	0	Block 255	FF8000h-FFFFFFh	32KB	Upper 1/512
0	1	0	1	0	1	Block 255	FF8000h-FFFFFFh	32KB	Upper 1/512
0	1	0	1	1	0	Block 255	FF8000h-FFFFFFh	32KB	Upper 1/512
0	1	0	1	1	1	Block 0 to 255	000000h-FFFFFFh	16384KB	All
0	1	1	0	0	0	None	None	None	None
0	1	1	0	0	1	Block 0	000000h-000FFFh	4KB	Lower 1/4096
0	1	1	0	1	0	Block 0	000000h-001FFFh	8KB	Lower 1/2048
0	1	1	0	1	1	Block 0	000000h-003FFFh	16KB	Lower 1/1024
0	1	1	1	0	0	Block 0	000000h-007FFFh	32KB	Lower 1/512
0	1	1	1	0	1	Block 0	000000h-007FFFh	32KB	Lower 1/512
0	1	1	1	1	0	Block 0	000000h-007FFFh	32KB	Lower 1/512
0	1	1	1	1	1	Block 0 to 255	000000h-FFFFFFh	16384KB	All

Table 3. Protected Area Sizes Sector Organization (Continued)

Status Register Content						Memory Content			
CMP	4KBL	T/B	SR.4	SR.3	SR.2	Protect Areas	Addresses	Density(KB)	Portion
Bit	Bit	Bit	Bit	Bit	Bit				
1	0	0	0	0	0	Block 0 to 255	000000h-FFFFFFh	16384KB	All
1	0	0	0	0	1	Block 0 to 251	000000h-FBFFFFh	16128KB	Lower 63/64
1	0	0	0	1	0	Block 0 to 247	000000h-F7FFFFh	15872KB	Lower 31/32
1	0	0	0	1	1	Block 0 to 239	000000h-EFFFFFh	15360KB	Lower 15/16
1	0	0	1	0	0	Block 0 to 223	000000h-DFFFFFh	14336KB	Lower 7/8
1	0	0	1	0	1	Block 0 to 191	000000h-BFFFFFh	12288KB	Lower 3/4
1	0	0	1	1	0	Block 0 to 127	000000h-7FFFFFh	8192KB	Lower 1/2
1	0	0	1	1	1	None	None	None	None
1	0	1	0	0	0	Block 0 to 255	000000h-FFFFFFh	16384KB	All
1	0	1	0	0	1	Block 4 to 255	040000h-FFFFFFh	16128KB	Upper 63/64
1	0	1	0	1	0	Block 8 to 255	080000h-FFFFFFh	15872KB	Upper 31/32
1	0	1	0	1	1	Block 16 to 255	100000h-FFFFFFh	15360KB	Upper 15/16
1	0	1	1	0	0	Block 32 to 255	200000h-FFFFFFh	14336KB	Upper 7/8
1	0	1	1	0	1	Block 64 to 255	400000h-FFFFFFh	12288KB	Upper 3/4
1	0	1	1	1	0	Block 128 to 255	800000h-FFFFFFh	8192KB	Upper 1/2
1	0	1	1	1	1	None	None	None	None
1	1	0	0	0	0	Block 0 to 255	000000h-FFFFFFh	16384KB	All
1	1	0	0	0	1	Block 0 to 255	000000h-FFEFFFh	16380KB	Lower 4095/4096
1	1	0	0	1	0	Block 0 to 255	000000h-FFDFFFh	16376KB	Lower 2047/2048
1	1	0	0	1	1	Block 0 to 255	000000h-FFBFFFh	16368KB	Lower 1023/1024
1	1	0	1	0	0	Block 0 to 255	000000h-FF7FFFh	16352KB	Lower 511/512
1	1	0	1	0	1	Block 0 to 255	000000h-FF7FFFh	16352KB	Lower 511/512
1	1	0	1	1	0	Block 0 to 255	000000h-FF7FFFh	16352KB	Lower 511/512
1	1	0	1	1	1	None	None	None	None
1	1	1	0	0	0	Block 0 to 255	000000h-FFFFFFh	16384KB	All
1	1	1	0	0	1	Block 0 to 255	001000h-FFFFFFh	16380KB	Upper 4095/4096
1	1	1	0	1	0	Block 0 to 255	002000h-FFFFFFh	16376KB	Upper 2047/2048
1	1	1	0	1	1	Block 0 to 255	004000h-FFFFFFh	16368KB	Upper 1023/1024
1	1	1	1	0	0	Block 0 to 255	008000h-FFFFFFh	16352KB	Upper 511/512
1	1	1	1	0	1	Block 0 to 255	008000h-FFFFFFh	16352KB	Upper 511/512
1	1	1	1	1	0	Block 0 to 255	008000h-FFFFFFh	16352KB	Upper 511/512
1	1	1	1	1	1	None	None	None	None

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

INSTRUCTIONS

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

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

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

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

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

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

Table 4. Instruction Set

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	n-Bytes
RSTEN	66h									
RST(1)	99h									
EQPI	38h									
RSTQIO(2)	FFh									
Write Enable (WERN)	06h									
Volatile Status Register Write Enable(3)	50h									
Write Disable (WRDI)	04h									
Read Status Register (RDSR)	05h	(SR7-SR0)(4)								continuous(5)
Read Status Register 2 (RDSR2)	09h /35h	(SR2.7-SR2.0)(4)								continuous(5)
Read Status Register 3 (RDSR3)	95h /15h	(SR3.7-SR3.0)(4)								continuous(5)
Write Status Register (WRSR)	01h	SR7-SR0	(SR2.7-SR2.0)(9)	(SR3.7-SR3.0)(9)						
Write Status Register 2 (WRSR2)	31h	SR2.7-SR2.0								
Write Status Register 3 (WRSR3)	C0h /11h	SR3.7-SR3.0								
Write Suspend	B0h/75h									
Write Resume	30h/7Ah									
Deep Power-down	B9h									
Release from Deep Power-down, and read Device ID (RES)	ABh	dummy	dummy	dummy	(ID7-ID0)					(6)
Release from Deep Power-down (RDP)										
Manufacturer/ Device ID	90h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(M7-M0)	(ID7-ID0)	(M7-M0)	(7)
				01h	(ID7-ID0)	(M7-M0)	(ID7-ID0)	(M7-M0)	(ID7-ID0)	
Manufacturer/ Device ID by Dual I/O	92h	dummy	dummy	00h	dummy	(ID7-ID0)	(ID7-ID0)	(M7-M0)	(ID7-ID0)	(7)
				01h		(M7-M0)	(M7-M0)	(ID7-ID0)	(M7-M0)	
Manufacturer/ Device ID by Quad I/O	94h	dummy	dummy	00h	dummy	dummy	dummy	(M7-M0)	(ID7-ID0)	(7)
				01h				(ID7-ID0)	(M7-M0)	
Read Identification (RDID)	9Fh	(M7-M0)	(ID15-ID8)	(ID7-ID0)	(8)					
Read OTP array	48h	A23-A16	A15-A8	A7-A0	Dummy	D7-D0	D7-D0	D7-D0	D7-D0	(Next Byte) Continuous
Program OTP array	42h	A23-A16	A15-A8	A7-A0	D7-D0	D7-D0	D7-D0	D7-D0	D7-D0	(Next Byte) Continuous
Erase OTP array	44h	A23-A16	A15-A8	A7-A0						
Read SFDP mode and Unique ID Number	5Ah	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	D7-D0	D7-D0	D7-D0	(Next Byte) continuous

Notes:

1. RST command only executed if RSTEN command is executed first. Any intervening command will disable Reset.
2. Release Full Quad SPI or Fast Read Enhanced mode. Device accepts eight-clocks command in Standard SPI mode, or two-clocks command in Full Quad SPI mode.
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# terminates 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.
9. WREN(01h) support 8 or 16 or 24 bit register value input for status register, status register 2 and status register 3

Table 5. 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)
Burst Read with Wrap	0Ch	24 bits	8 bits / 8 clocks	(D7-D0, …)	(Next Byte) continuous
DDR Fast Read	0Dh	24 bits	8 bits / 4 clocks	(D7-D0, …)	(8 bits per 4 clocks, continuous)
DDR Dual I/O Fast Read	BDh	24 bits	8 bits / 2 clocks	(D7-D0, …)	(8 bits per 2 clock, continuous)
DDR Quad I/O Fast Read	EDh	24 bits	24 bits / 3 clocks	(D7-D0, …)	(8 bits per 1 clock, continuous)
DDR Read Burst with Wrap	DCh	24 bits	8 bits / 4 clocks	(D7-D0, …)	(8 bits per 4 clock, continuous)

Table 6. 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)
DDR Mode Page Program	D2h	24 bits	0	(D7-D0, …)	(8 bits per 1 clock, continuous)

Table 7. Instruction Set (Erase Instruction)

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

Table 8. RPMC command

Instruction Name	Byte 0	Byte 1 (Cmd Type)	Byte 2	Byte 3	Byte n	Byte n
Write Root Key Register	9Bh	00h	Counter Addr [7:0]	Reserved [7:0]	Byte 4-35 Root Key [255:0]	Byte 36-63 Truncated Sign [223:0]
Update HMAC Key Register	9Bh	01h	Counter Addr [7:0]	Reserved [7:0]	Byte 4-7 Key Date [31:0]	Byte 8-39 Signature [255:0]
Increment Monotonic Counter	9Bh	02h	Counter Addr [7:0]	Reserved [7:0]	Byte 4-7 Counter Data [31:0]	Byte 8-39 Signature [255:0]
Request Monotonic Counter	9Bh	03h	Counter Addr [7:0]	Reserved [7:0]	Byte 4-15 Tag [95:0]	Byte 16-47 Signature [255:0]
Reserved Commands	9Bh	04h~FFh	Reserved			

Instruction Name	Byte 0	Byte 1	Byte 2	Byte 3-14	Byte 15-18	Byte 19-50
Read RPMC Status/Data	96h	dummy	RPMC Status [7:0]	Tag [95:0]	Counter Data [31:0]	Signature [255:0]

Note:

1. All RPMC instructions are in Standard SPI format. Each Input / Output Byte requires 8 clocks.
2. The Reserved[7:0] field for RPMC OP1 must be all 0s (00000000'b).
3. The controller may terminate the Read RPMC Status/Data instruction at any time without going through the entire data output sequence.
4. When BUSY=1, from Byte-3 and beyond, the device will output the RPMC_Status[7:0] value continuously until CS# terminates the instruction. The device will not output Tag, Counter Data & Signature fields when BUSY=1. Once BUSY becomes 0, another OP2 command must be issued to read out the correct Tag, Counter Data & Signature fields.

Table 9. Manufacturer and Device Identification

OP Code	(M7-M0)	(ID15-ID0)	(ID7-ID0)
ABh			77h
90h/92h/94h	1Ch		77h
9Fh	1Ch	7818h	

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

A successful command execution will reset the status register, see Reset-Enable and Reset Sequence Diagram figure for SPI Mode and Reset-Enable and Reset Sequence Diagram under QPI Mode figure for Quad 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 5. Reset-Enable and Reset Sequence Diagram

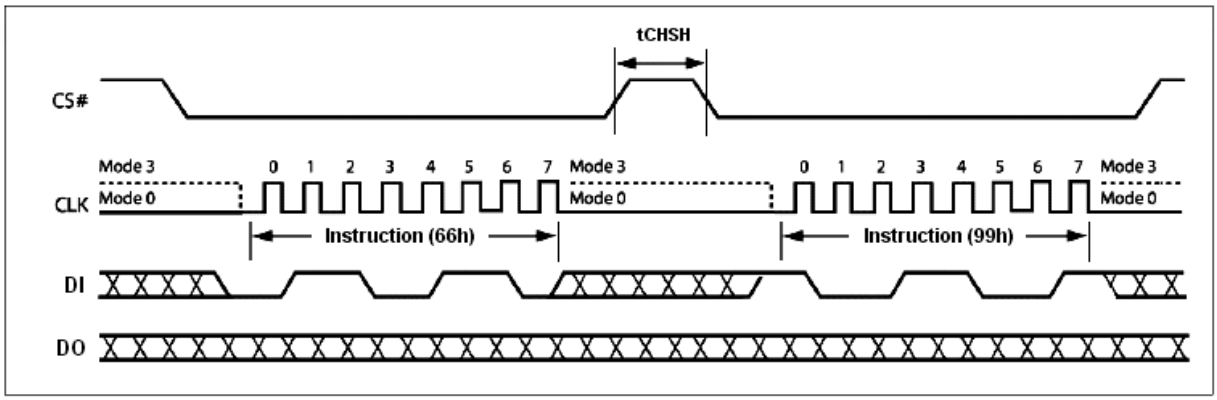


Figure 6. Reset-Enable and Reset Sequence Diagram under QPI Mode

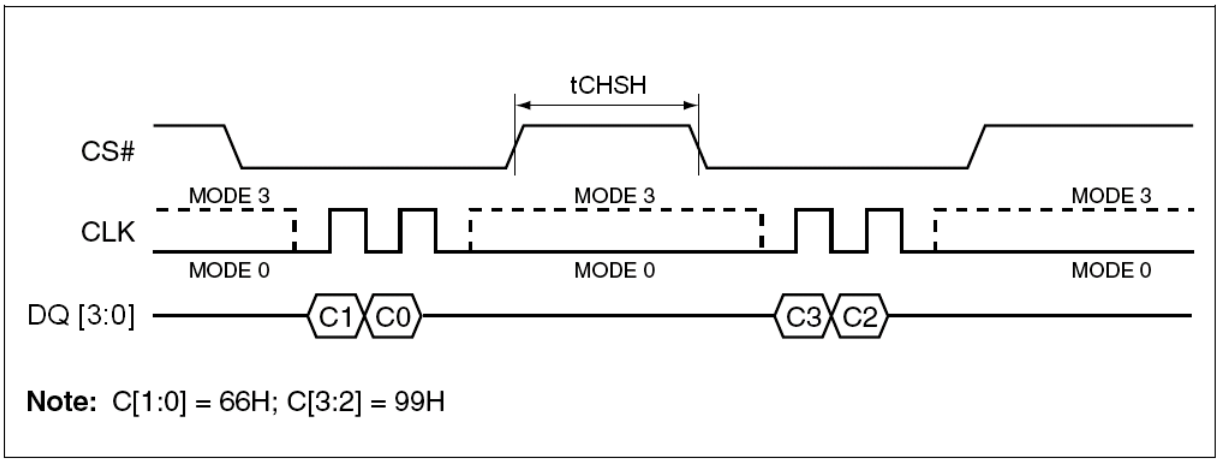
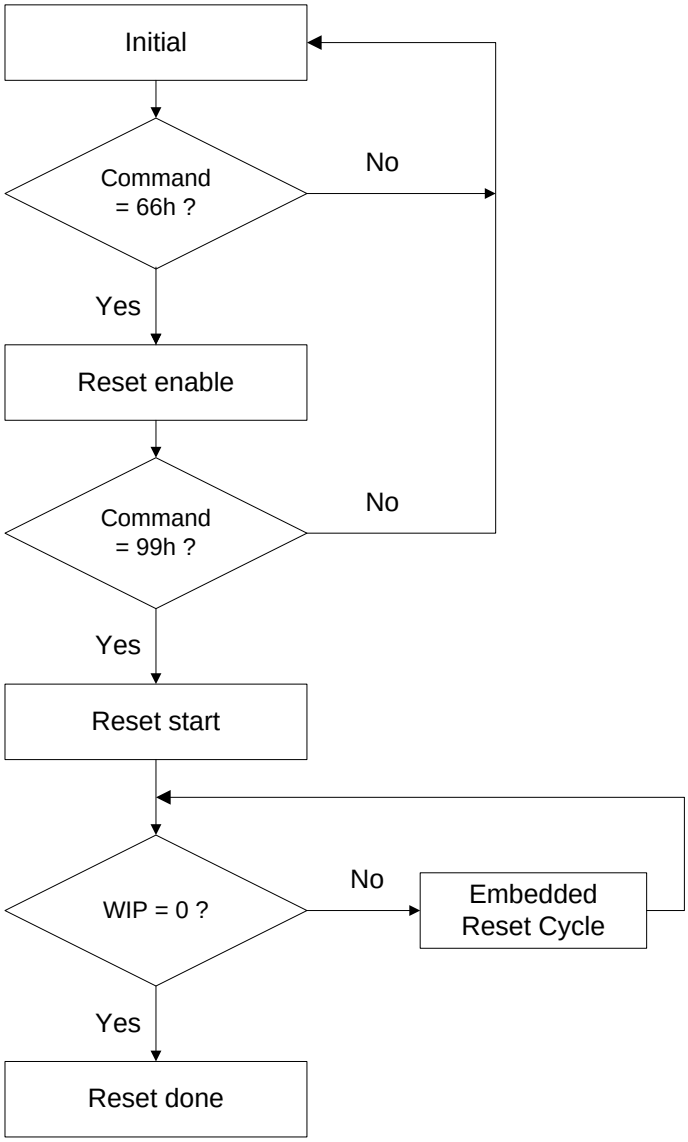


Figure 7. Software Reset Flow



Note:

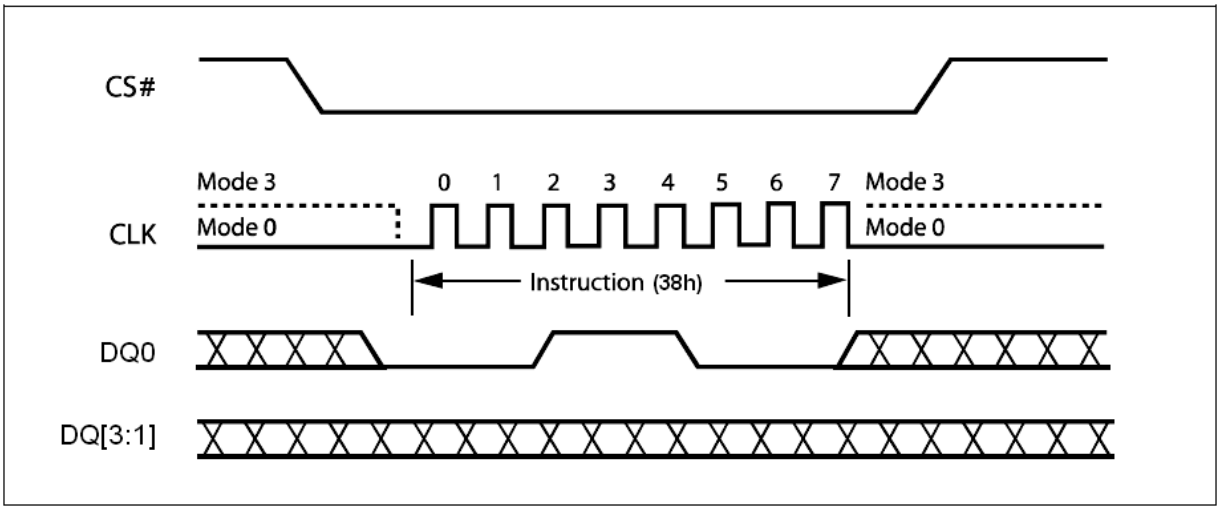
1. Reset-Enable (RSTEN) (66h) and Reset (RST) (99h) commands need to match standard SPI or QPI (quad) mode.
2. Continue (Enhance) EB mode need to use quad Reset-Enable (RSTEN) (66h) and quad Reset (RST) (99h) commands.
3. If user is not sure it is in SPI or Quad mode, we suggest to execute sequence as 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, Continue EB mode and suspend mode to back to SPI mode.
5. This flow can 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.

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 Enable Quad Peripheral Interface mode Sequence Diagram figure. The device did not support the Read Data Bytes (READ) (03h)**,** Dual Output Fast Read (3Bh) and Dual Input/Output FAST_READ (BBh) and Quad Output Fast Read (6Bh) modes while the Enable Quad Peripheral Interface mode (EQPI) (38h) turns on.

Figure 8. 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 FFh 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.

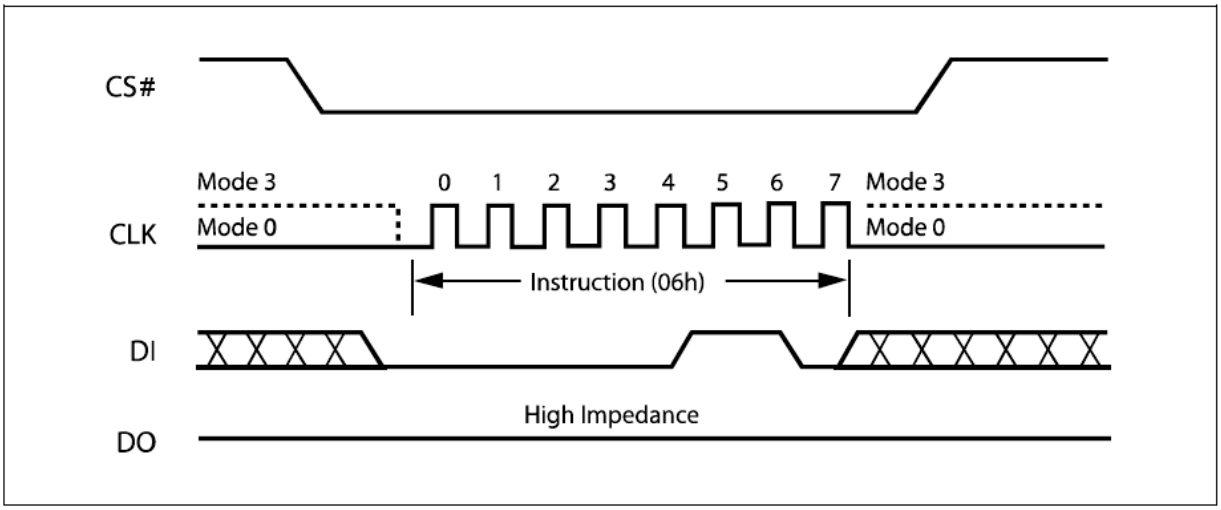
Note:
If the system is in the Quad I/O Fast Read Enhance Mode in QPI Mode, it is necessary to execute FFh command by two times. The first FFh command is to release Quad I/O Fast Read Enhance Mode, and the second FFh command is to release EQPI Mode.

Write Enable (WREN) (06h)

The Write Enable (WREN) instruction (Write Enable Instruction Sequence Diagram figure) sets the Write Enable Latch (WEL) bit. The Write Enable Latch (WEL) bit must be set prior to every Page Program (PP), Quad Input Page Program (QPP), Sector Erase (SE), Block Erase (HBE/BE), Chip Erase (CE) and Write Status Register (WRSR/ WRSR3) instruction.
The Write Enable (WREN) instruction is entered by driving Chip Select (CS#) Low, sending the instruction code, and then driving Chip Select (CS#) High.

The instruction sequence is shown in Write Enable/Disable Instruction Sequence under QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 9. Write Enable Instruction Sequence Diagram



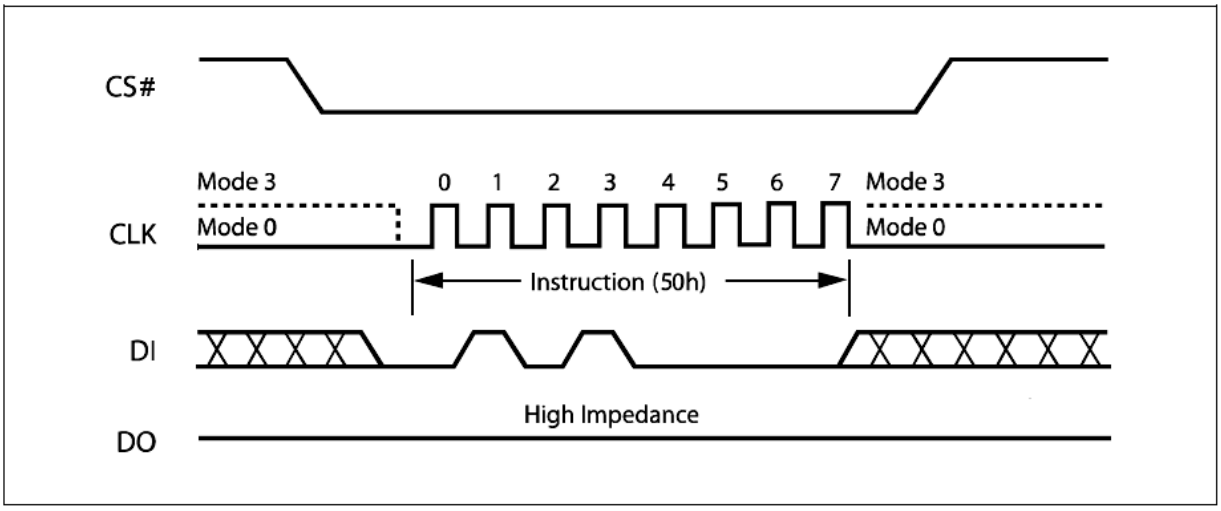
Volatile Status Register Write Enable (50h)

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), WRSR2 and WRSR3 commands 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) or WRSR2 or WRSR3. The Status Register bits will be refresh to Volatile Status Register (SR[7:2] or SR2[7:0]) or SR3[7:0]) within t_{SHSL2} (50ns). Upon power off or the execution of a Software/Hardware Reset, the volatile Status Register bit values will be lost, and the non-volatile Status Register bit values will be restored. The instruction sequence is shown in Volatile Status Register Write Enable Instruction Sequence Diagram figure.

The instruction sequence is shown in Write Enable/Disable Instruction Sequence under QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 10. Volatile Status Register Write Enable Instruction Sequence Diagram



Write Disable (WRDI) (04h)

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

The instruction sequence is shown in Write Enable/Disable Instruction Sequence under QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 11. Write Disable Instruction Sequence Diagram

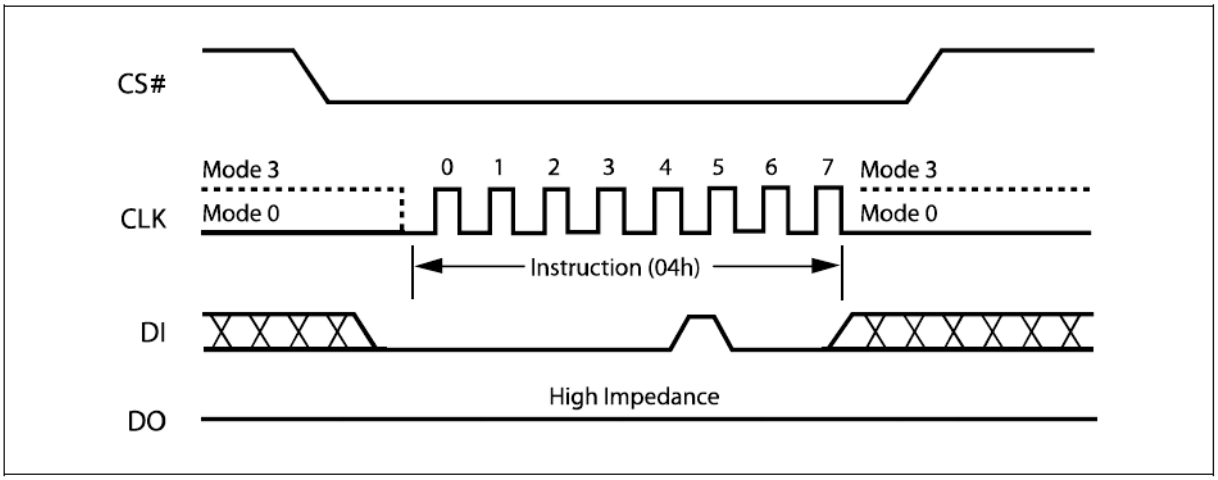
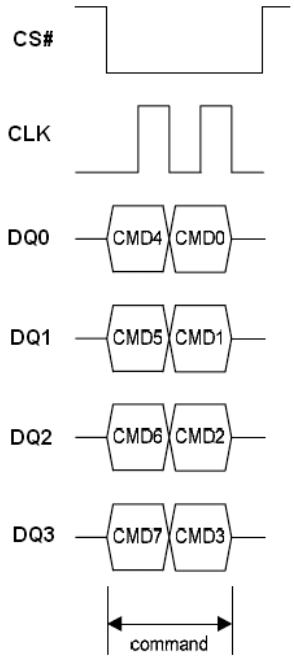


Figure 12. Write Enable/Disable Instruction Sequence under QPI Mode



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 Read Status Register Instruction Sequence Diagram figure.

The instruction sequence is shown in Read Status Register Instruction Sequence under QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 13. Read Status Register Instruction Sequence Diagram

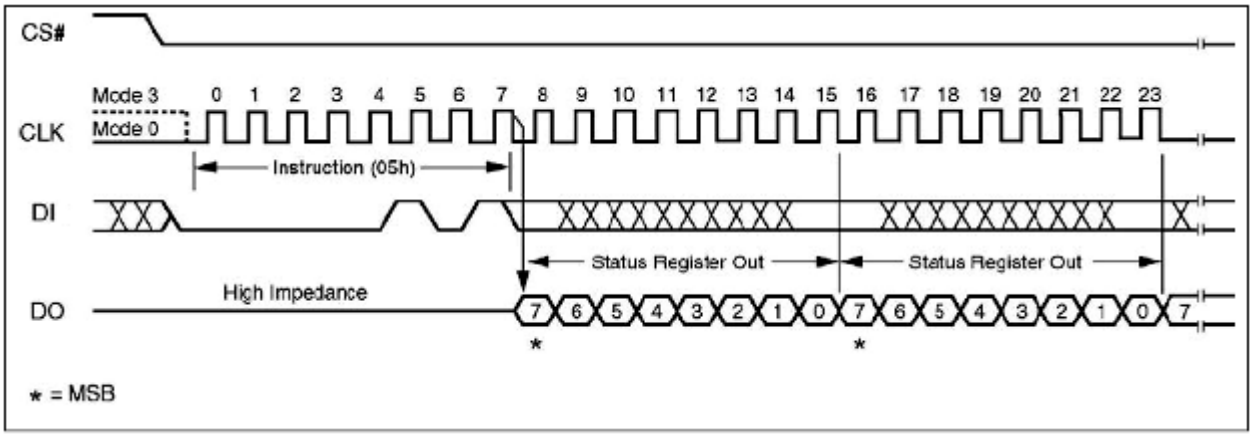


Figure 14. Read Status Register Instruction Sequence under QPI Mode

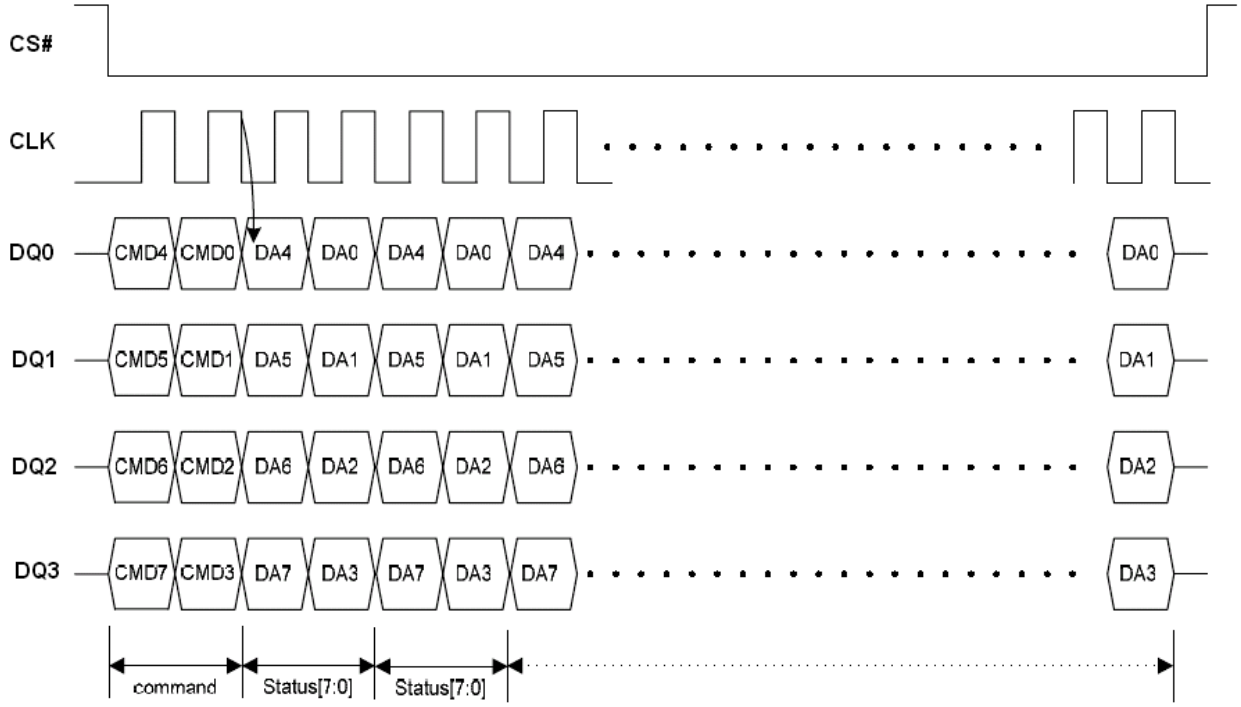


Table 10. Status Register Bit Locations

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

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

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

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, CMP, 4KBL, TB, BP2, BP1, BP0) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

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

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

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

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.

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.

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

The Read Status Register 2 (RDSR2) instruction allows the Status Register 2 to be read. The Status Register 2 may be read at any time, even while a Write Suspend or Write Resume cycle is in progress. When one of these cycles is in progress, it is recommended to check the Write In Progress (WIP) bit before sending a new instruction to the device. It is also possible to read the Status Register 2 continuously, as shown in Read Status Register 2 Instruction Sequence Diagram figure.

The instruction sequence is shown in Read Status Register 2 Instruction Sequence under QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 15. Read Status Register 2 Instruction Sequence Diagram

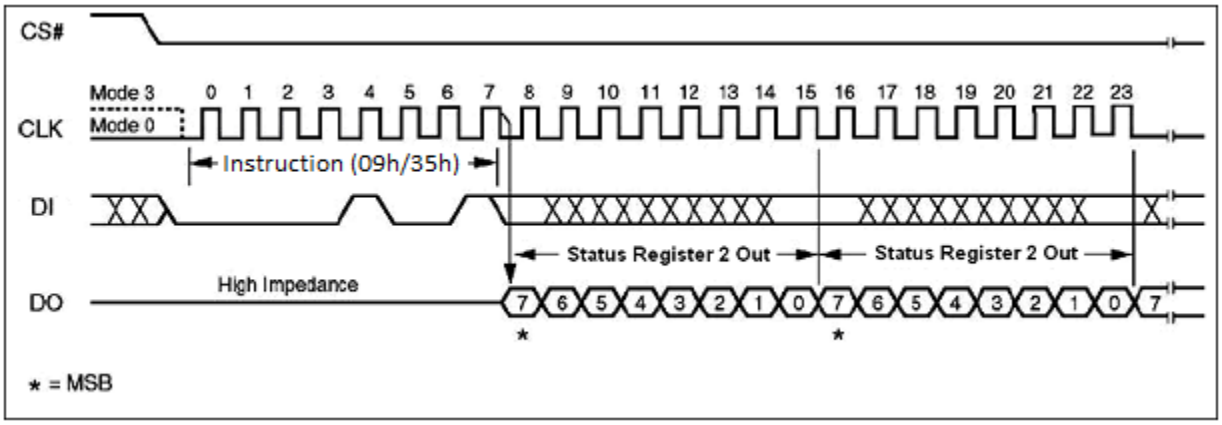


Figure 16. Read Status Register 2 Instruction Sequence under QPI Mode

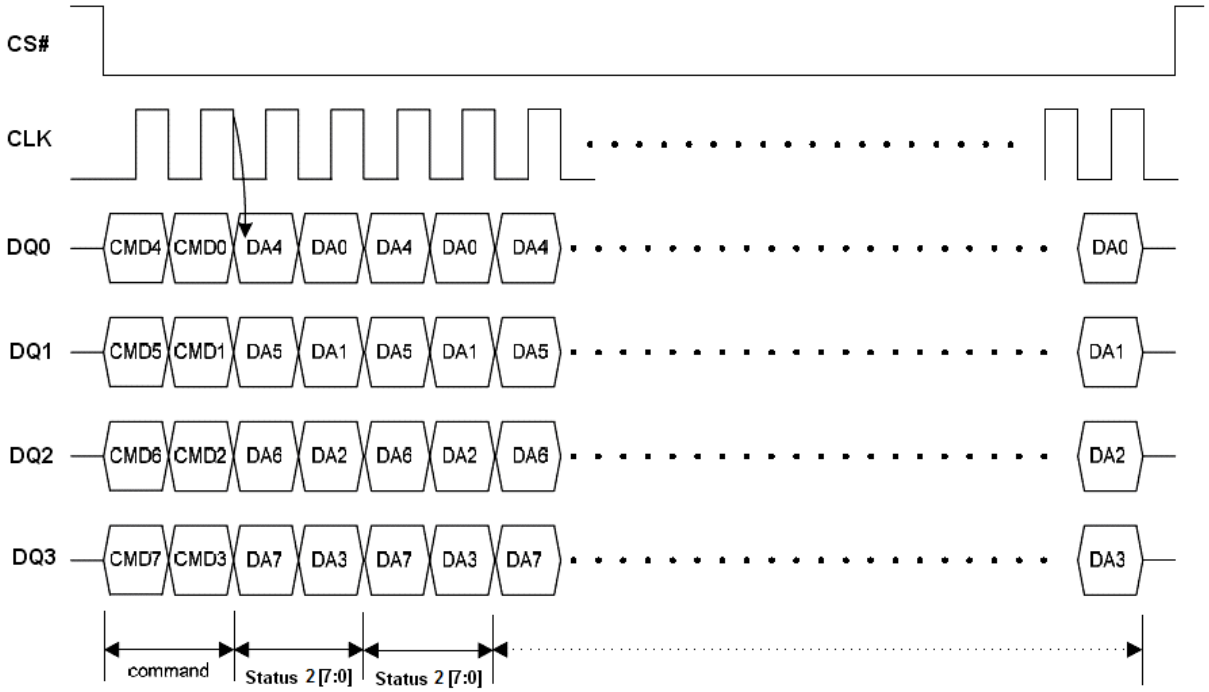


Table 11. Status Register 2 Bit Locations

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

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

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

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

CMP bit. The Complement Protect bit (CMP) is a non-volatile bit in the status register. It is used in conjunction with 4KBL, TB, BP2, BP1 and BP0 bits to provide more flexibility for the array protection. This bit only can be programmed one time. Once CMP is set to 1, it cannot change again. The default setting is CMP=0.

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.

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.

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.

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”.

QE bit. The Quad Enable (QE) bit is a non-volatile bit and the default value is “1” to support Quad Input/Output Fast Read (EBh) or EQPI (38h) directly without additional setting. However, WP# and HOLD#/RESET# can be enabled by setting QE=”0”.

Reserved bit. Status Register 2 bit locations SR2.0 is 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 Suspend Status Register. Doing this will ensure compatibility with future devices.

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

The Read Status Register 3 (RDSR3) instruction allows the Status Register 3 to be read. The Status Register 3 may be read at any time, even while 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 3 continuously, as shown in Read Status Register 3 Instruction Sequence Diagram figure.

The instruction sequence is shown in Read Status Register 3 Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 17. Read Status Register 3 Instruction Sequence Diagram

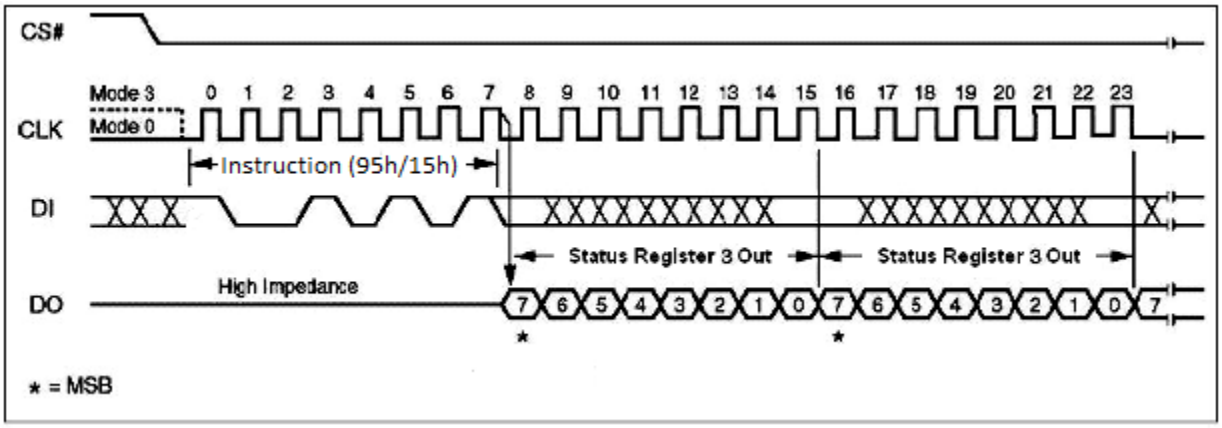


Figure 18. Read Status Register 3 Instruction Sequence in QPI Mode

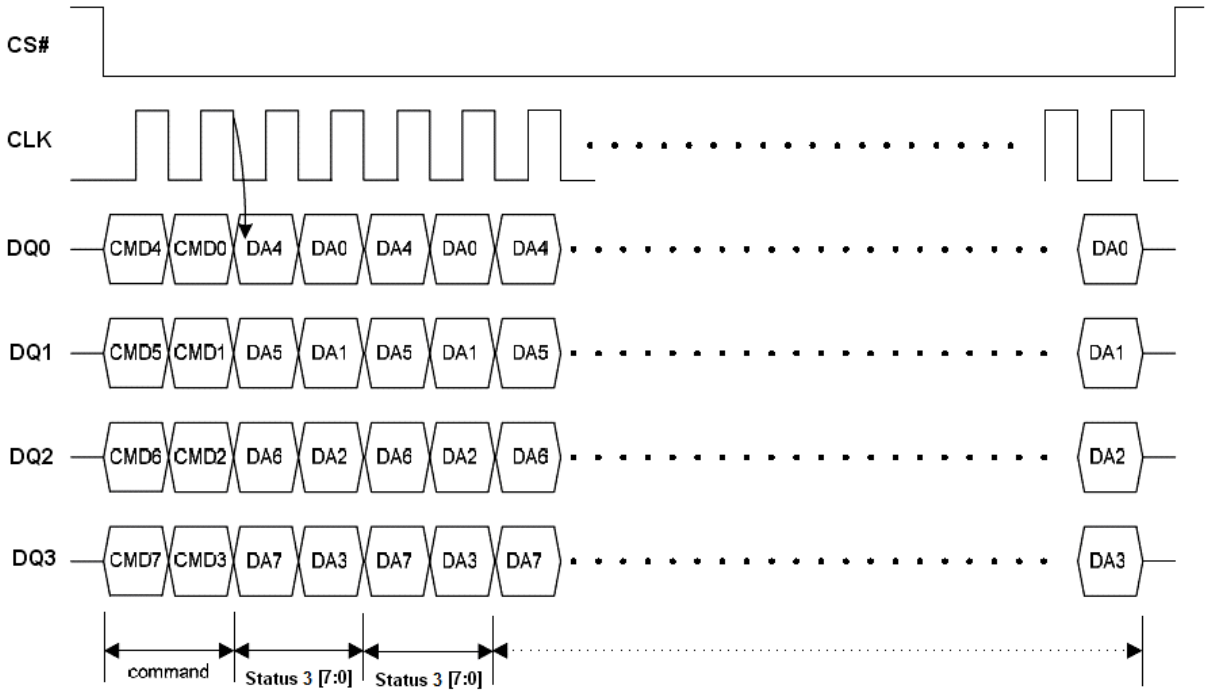


Table 12. Status Register 3 Bit Locations

SR3.7	SR3.6	SR3.5	SR3.4	SR3.3	SR3.2	SR3.1	SR3.0
HRSW bit (HOLD#/RESET# switch)	**Output Drive Strength bit**		**Burst Length bit**		**Reserved bit**		
1 = RESET# enable 0 = HOLD# enable (default 0)	00 = 67% (default) 01 = 100% 10 = 50% (1/2) drive 11 = 33% (1/3) drive		00 = 8 Bytes (default) 01 = 16 Bytes 10 = 32 Bytes 11 = 64 Bytes				
Non-volatile / volatile bit	Non-volatile / volatile bit		Non-volatile / volatile bit				

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

HRSW bit. The HOLD#/RESET# switch bit (HRSW bit), Non-Volatile / Volatile bit, the HRSW bit is used to determine whether HOLD# or RESET# function should be implemented on the hardware pin. When it is “0” (factory default), the pin acts as HOLD#; when it is “1”, the pin acts as RESET#. However, HOLD# or RESET# functions are only available when QE bit is “0”. If QE bit is set to “1”, the HOLD# and RESET# functions are disabled, the pin acts as a dedicated data I/O pin.

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

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

Reserved bit. SR3.2, SR3.1 and SR3.0 are reserved for future use.

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 or data bytes on Serial Data Input (DI). The WRSR instruction also support multi bytes data input to set other status registers.

The instruction sequence is shown in Write Status Register Instruction Sequence Diagram figure. 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, 16th or 24th 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 (CMP, 4KBL, TB, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in Protected Area Sizes Sector Organization table. The Write Status Register (WRSR) instruction also allows the user to set or reset the Status Register Protect (SRP) bit in accordance with the Write Protect (WP#) signal. The Status Register Protect (SRP) bit and Write Protect (WP#) signal allow the device to be put in the Hardware Protected Mode (HPM). The Write Status Register (WRSR) instruction is not executed once the Hardware Protected Mode (HPM) is entered.

The instruction sequence is shown in Write Status Register Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 19. Write Status Register Instruction Sequence Diagram

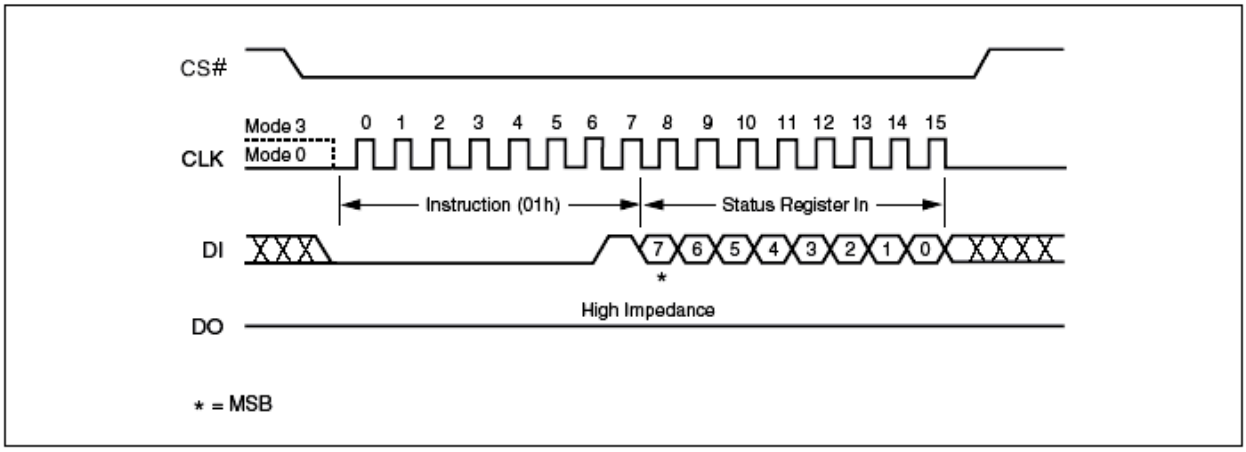


Figure 20. Write Status Register Instruction Sequence Diagram (multi bytes)

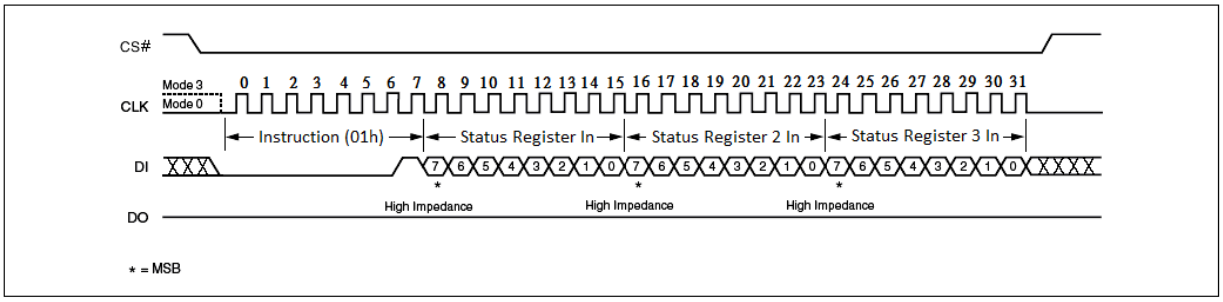


Figure 21. Write Status Register Instruction Sequence in QPI Mode

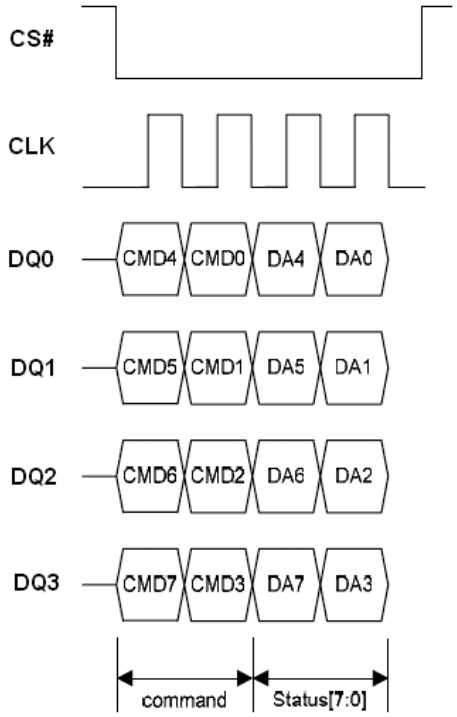
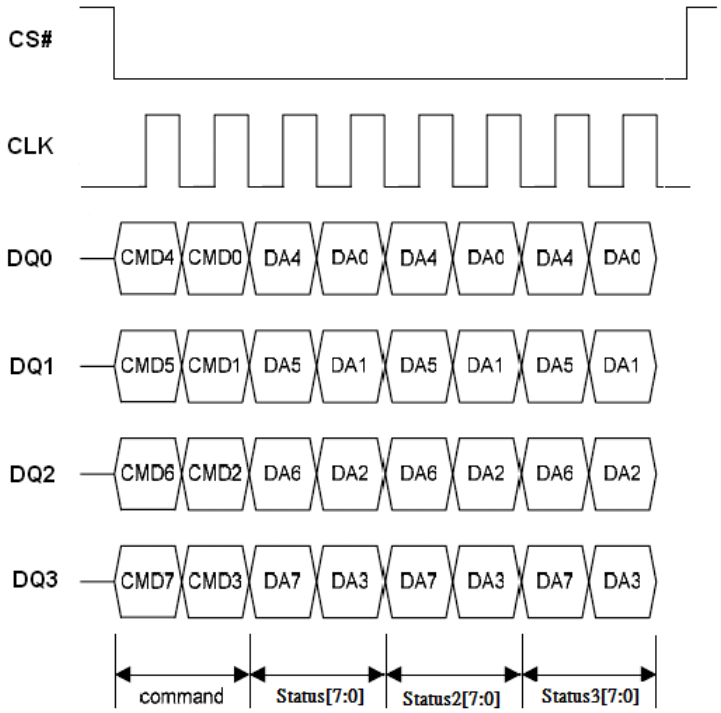


Figure 22. Write Status Register Instruction Sequence in QPI Mode (multi bytes)



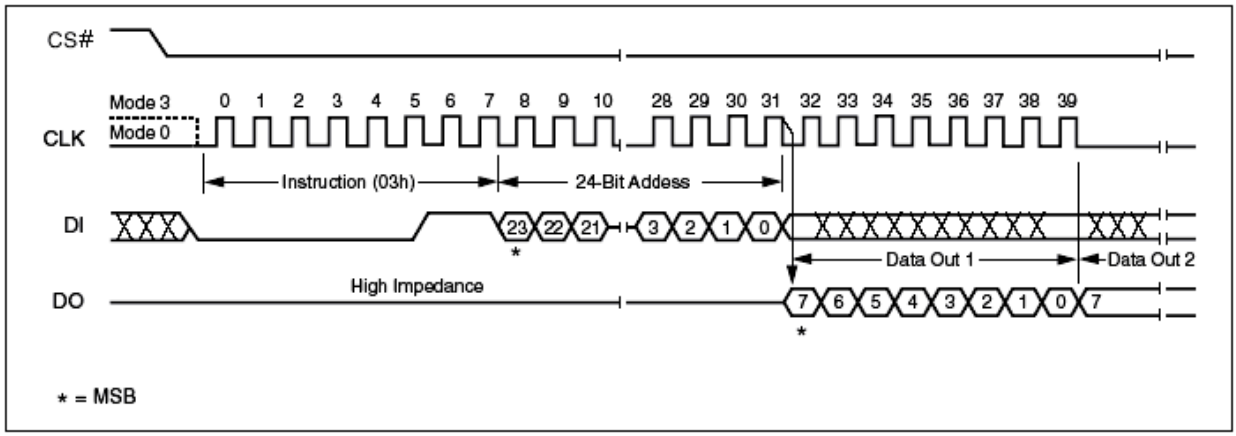
Read Data Bytes (READ) (03h)

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

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

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

Figure 23. Read Data Instruction Sequence Diagram



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

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

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

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

The instruction sequence is shown in Fast Read Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 24. Fast Read Instruction Sequence Diagram

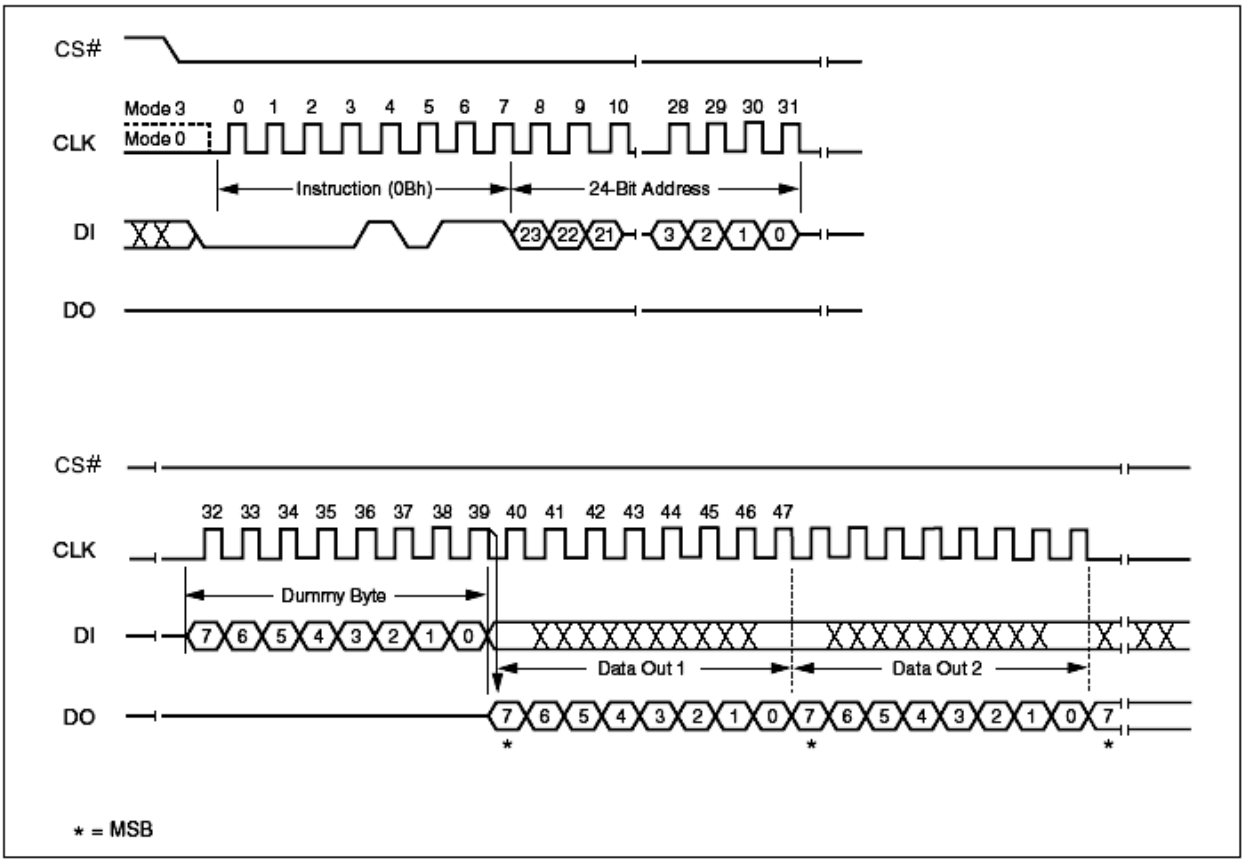
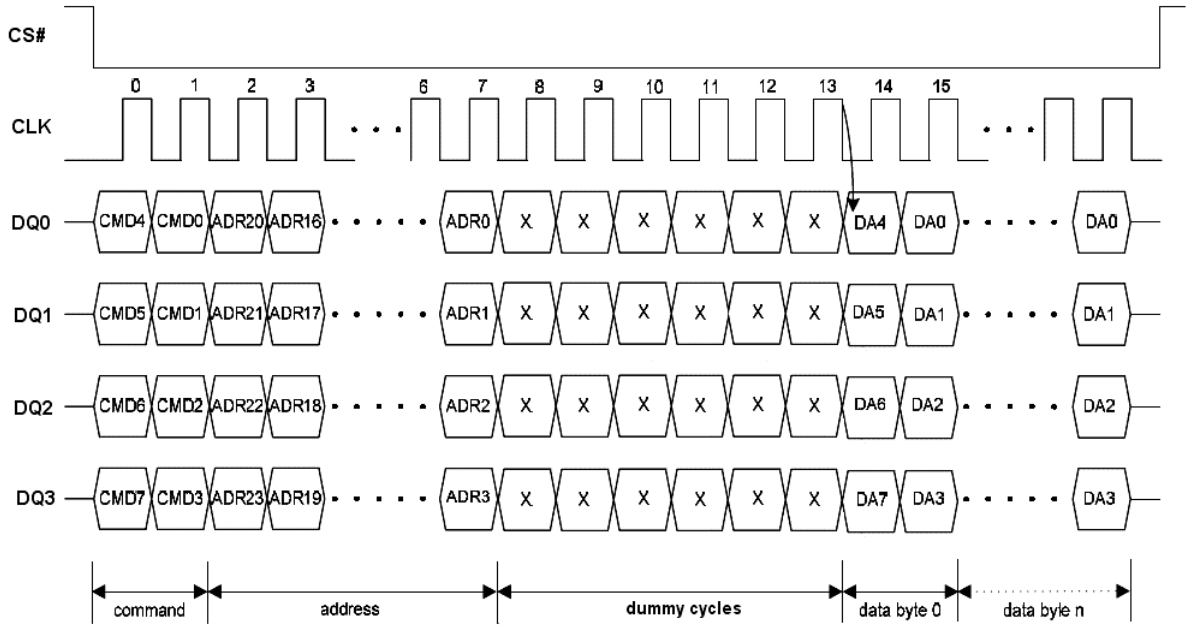


Figure 25. Fast Read Instruction Sequence in QPI Mode



DDR Read Data Bytes at Higher Speed (DDR FAST_READ) (0Dh)

The DDR FAST_READ instruction (DDR Fast Read Instruction Sequence Diagram figure) is for doubling reading data out, signals are triggered on both rising and falling edge of clock. The address is latched on both rising and falling edge of CLK, and data of each bit shifts out on both rising and falling edge of CLK at a maximum frequency F_R. The 2-bit address can be latched-in at one clock, and 2-bit data can be read out at one clock, which means one bit at rising edge of clock, the other bit at falling edge of clock. The first address byte 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 DDR FAST_READ instruction. The address counter rolls over to 0 when the highest address has been reached.

The sequence of issuing DDR FAST_READ instruction is: CS# goes low -> sending DDR FAST_READ instruction code (1 bit per clock) -> 3-byte address on DI (2-bit per clock) -> 1 dummy byte (default) on DI -> data out on DO (2-bit per clock) -> to end DDR FAST_READ operation can use CS# to high at any time during data out.

While Program/ Erase/ Write Status Register cycle is in progress, DDR FAST_READ instruction is rejected without any impact on the Program/ Erase/ Write Status Register current cycle.

The instruction sequence is shown in DDR Fast Read Instruction Sequence Diagram in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 26. DDR Fast Read Instruction Sequence Diagram

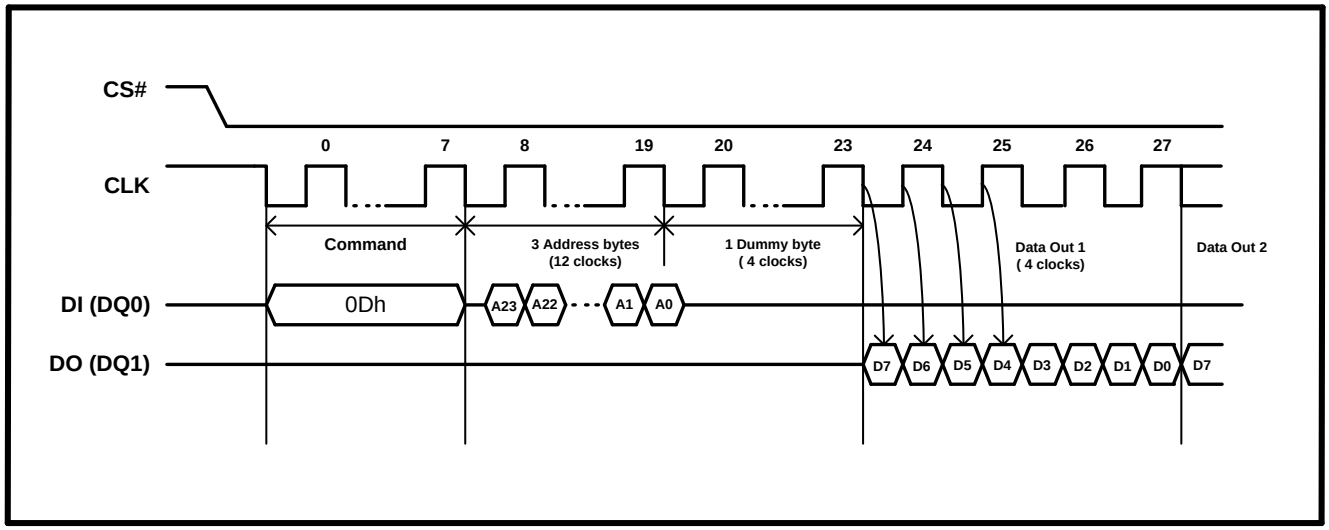
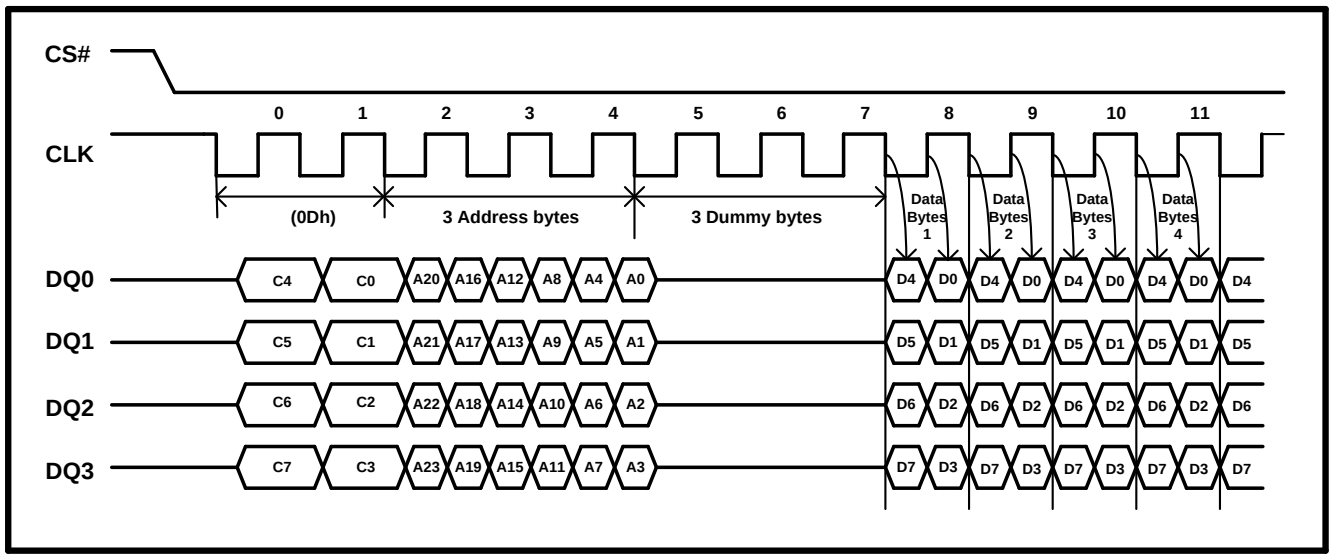


Figure 27. DDR Fast Read Instruction Sequence Diagram in QPI Mode

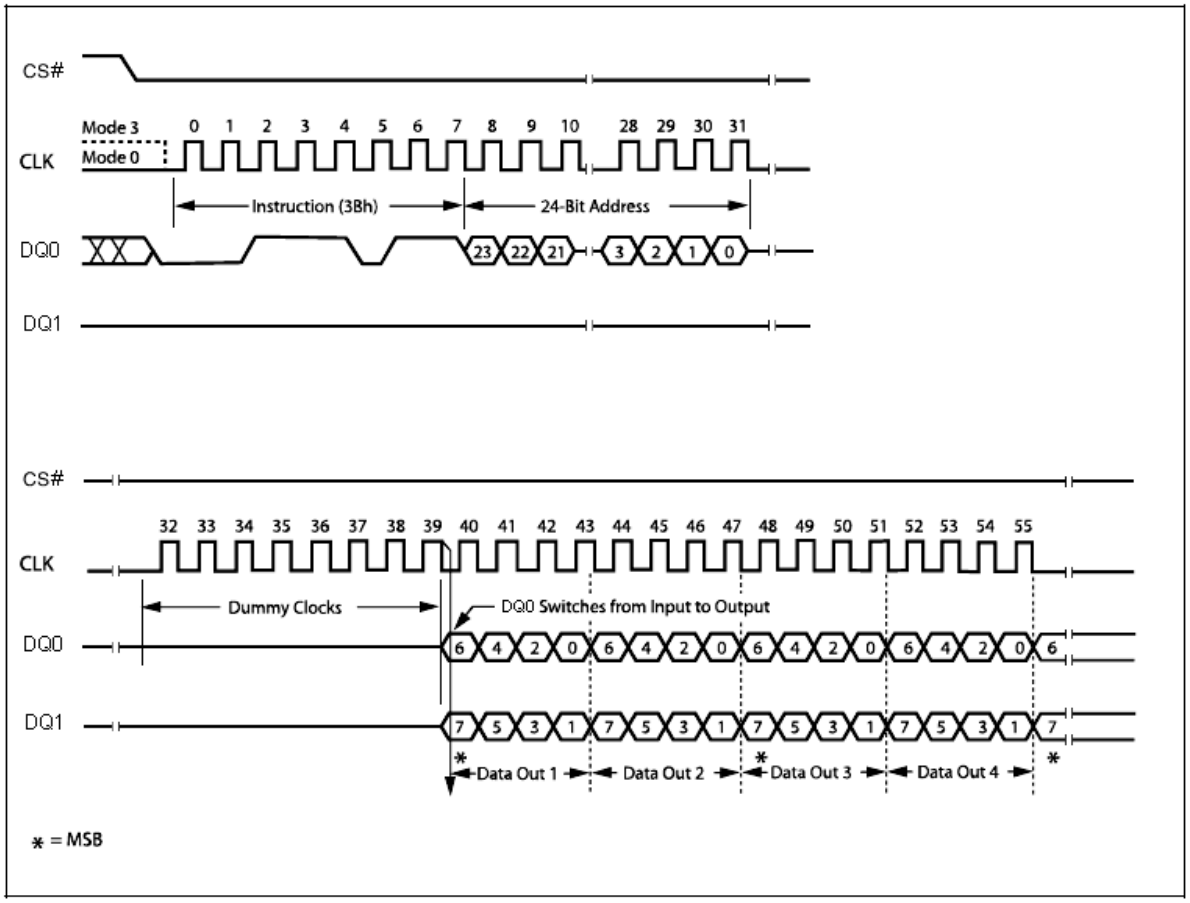


Dual Output Fast Read (3Bh)

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 device at twice the rate of standard SPI devices. The Dual Output Fast Read instruction is ideal for quickly downloading code from to RAM upon power-up or for applications that cache code-segments to RAM for execution.

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

Figure 28. Dual Output Fast Read Instruction Sequence Diagram

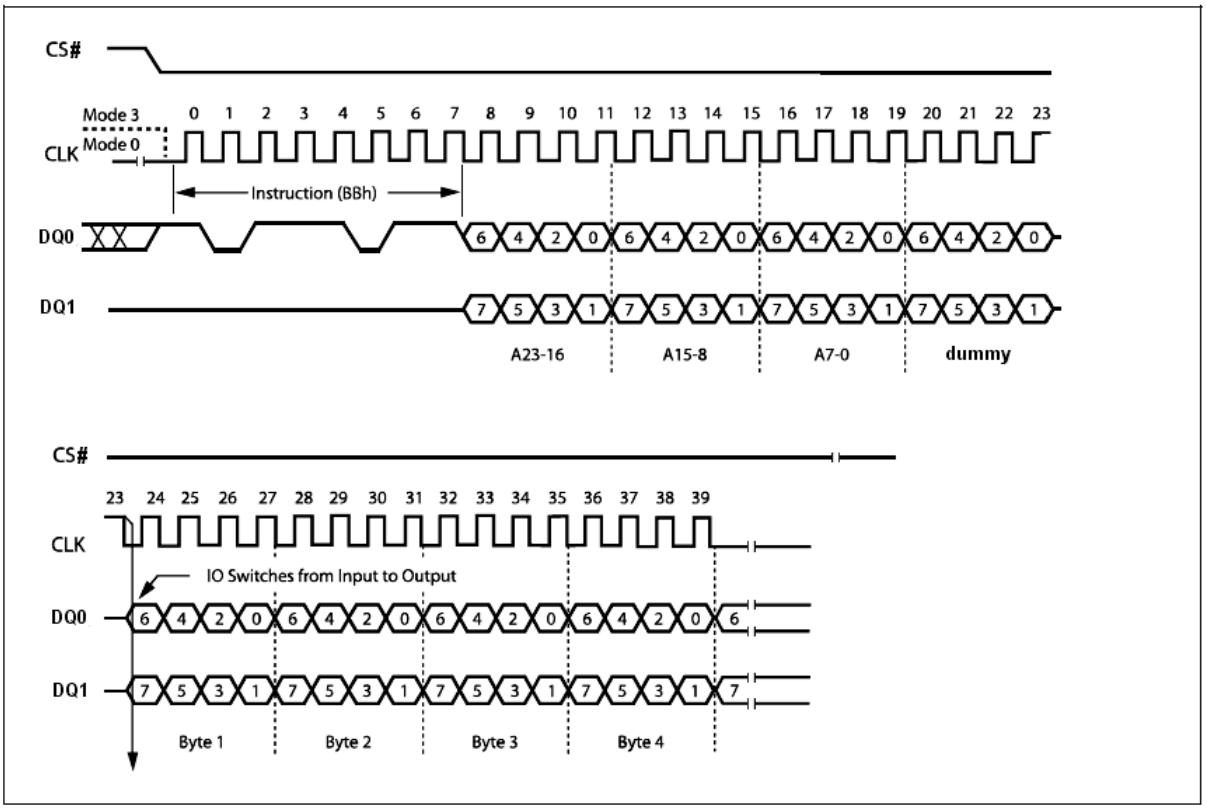


Dual Input / Output FAST_READ (BBh)

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

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

Figure 29. Dual Input / Output Fast Read Instruction Sequence Diagram



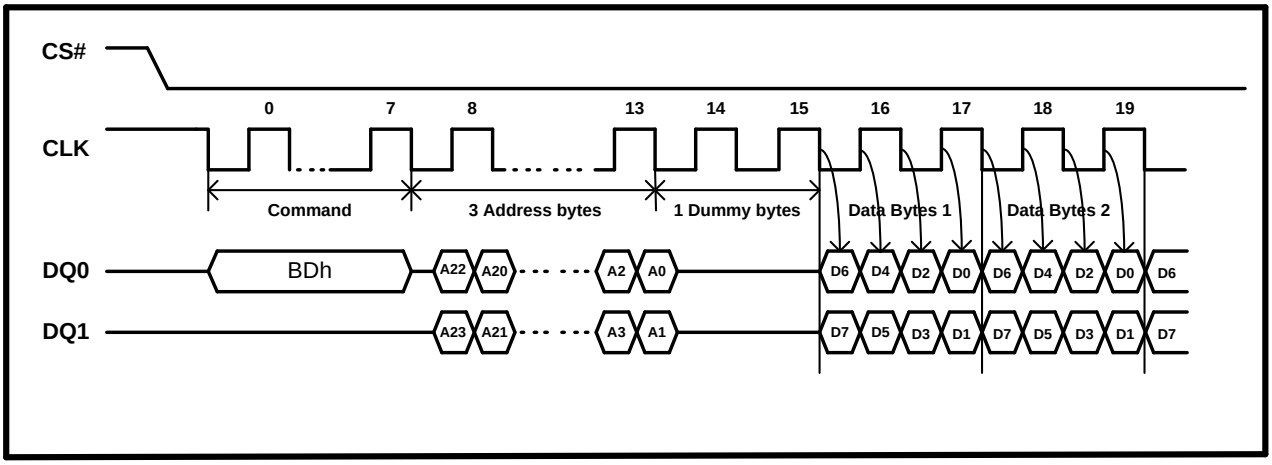
DDR Dual Input / Output FAST_READ (BDh)

The DDR Dual Input / Output FAST_READ (BDh) instruction enables Double Data Rate throughput on dual I/O of Serial Flash in read mode. The address (interleave on dual I/O pins) is latched on both rising and falling edge of CLK, and data (interleave on dual I/O pins) shift out on both rising and falling edge on CLK at a maximum frequency F_R. The 4-bit address can be latched-in at one clock, and 4-bit data can be read out at one clock, which means two bits at rising edge of clock, the other two bits at falling edge of clock. The first address byte 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 DDR Dual Input / Output FAST_READ (BDh) instruction. The address counter rolls over 0 when the highest address has been reached. Once writing DDR Dual Input / Output FAST_READ (BDh) instruction, the following address/dummy/data out will perform as 4-bit instead of previous 1-bit.

The sequence of issuing DDR Dual Input / Output FAST_READ (BDh) instruction is : CS# goes low -> sending DDR Dual Input / Output FAST_READ (BDh) instruction (1-bit per clock) -> 24-bit bit address interleave on DQ3, DQ2, DQ1 and DQ0 (4-bit per clock) -> 1 dummy byte (2 clocks) -> data out interleave on DQ3, DQ2, DQ1 and DQ0 (4-bit per clock) -> to end DDR Dual Input / Output FAST_READ (BDh) operation can use CS# to high at any time during data out, as shown in DDR Dual Input / Output FAST_READ Instruction Sequence Diagram figure.

Figure 30. DDR Dual Input / Output FAST_READ Instruction Sequence Diagram



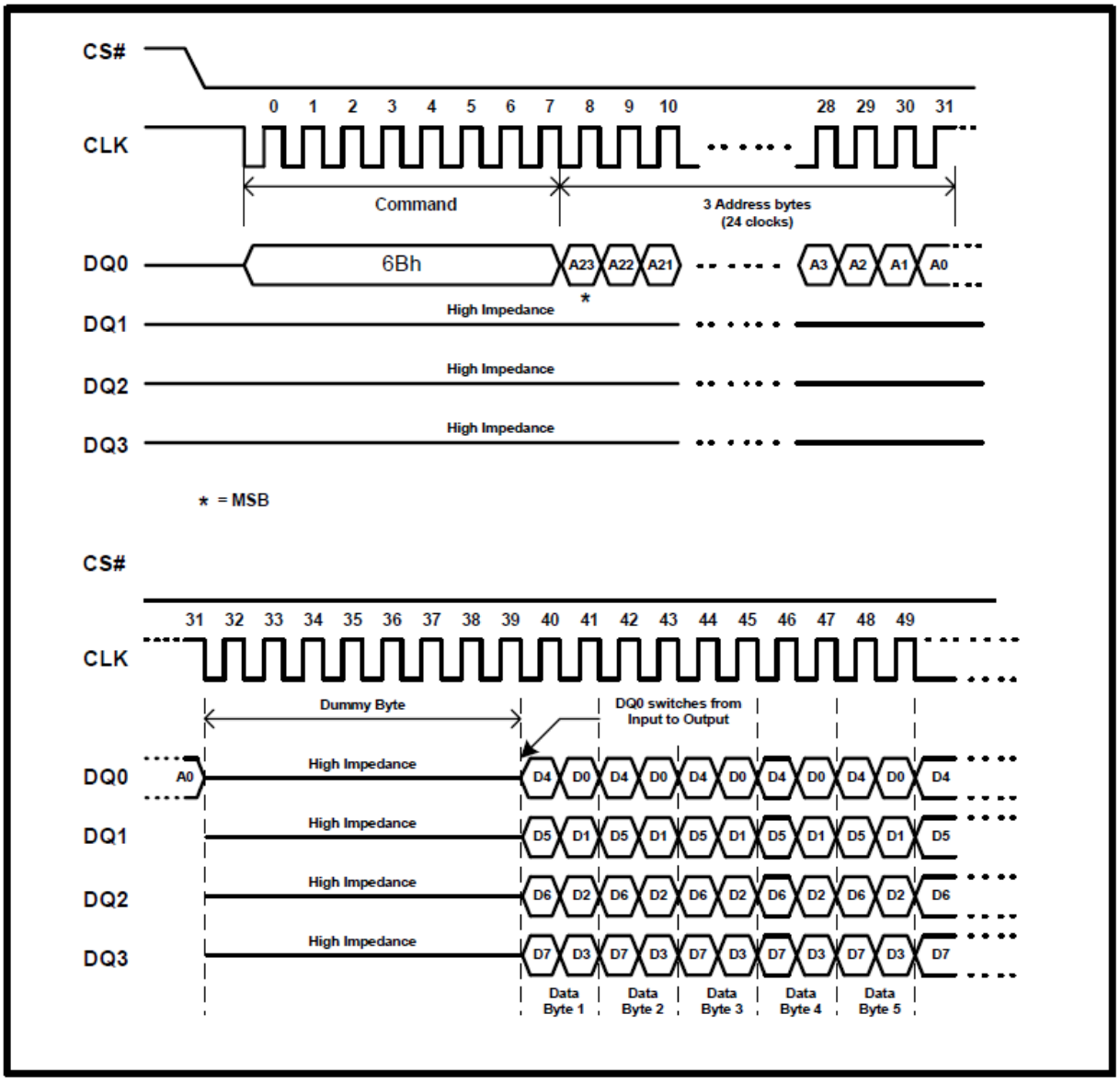
Quad Output Fast Read (6Bh)

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 F_R. The first address can be any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Quad Output Fast Read instruction. The address counter rolls over to 0 when the highest address has been reached.

The sequence of issuing Quad Output Fast Read (6Bh) instruction is: CS# goes low -> sending Quad Output Fast Read (6Bh) instruction -> 24-bit address on 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 Quad Output Fast Read Instruction Sequence Diagram figure. The WP# (DQ2) and HOLD# (DQ3) need to drive high before address input if QE bit in Status Register is 0.

Figure 31. Quad Output Fast Read Instruction Sequence Diagram



Quad Input / Output FAST_READ (EBh)

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

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

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

The instruction sequence is shown in Quad Input / Output Fast Read Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 32. Quad Input / Output Fast Read Instruction Sequence Diagram

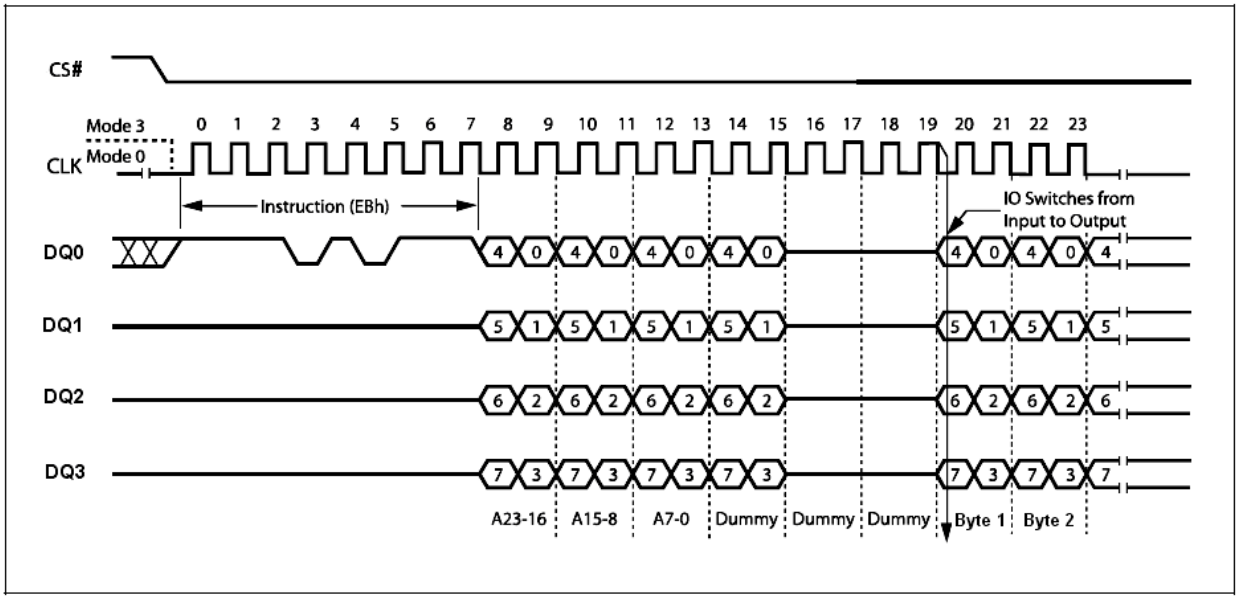
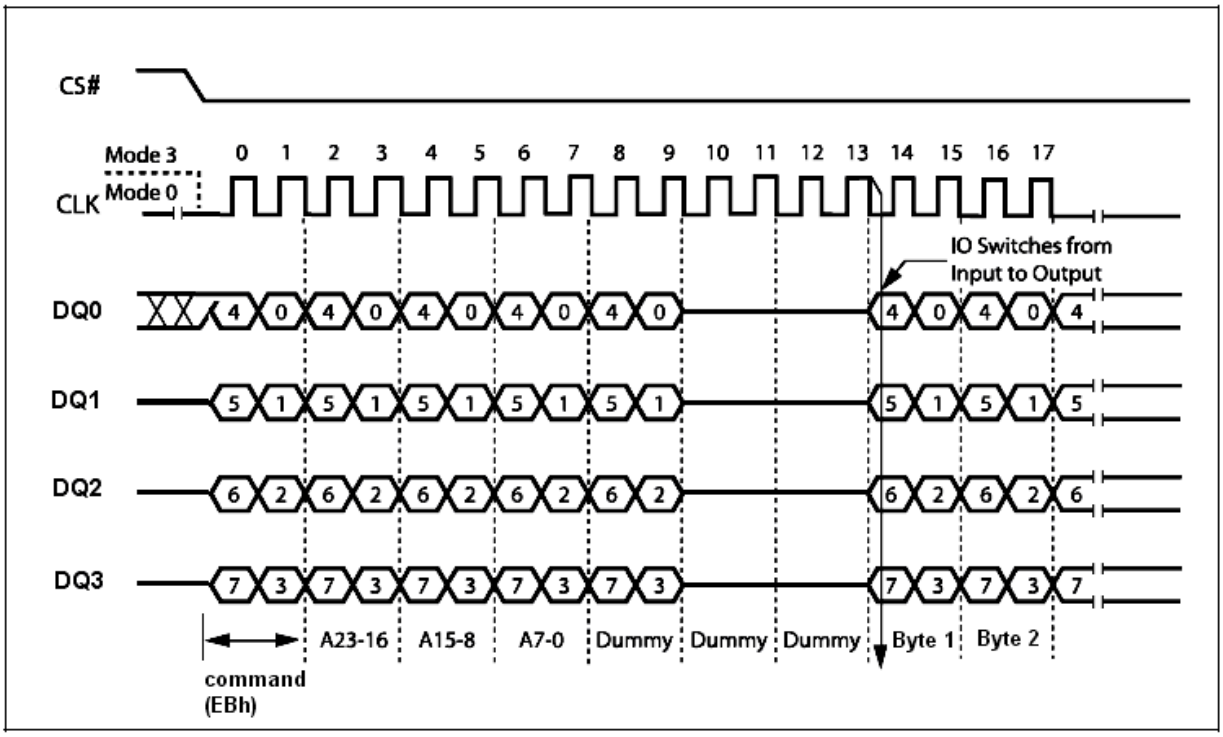


Figure 33. Quad Input / Output Fast Read Instruction Sequence in QPI Mode



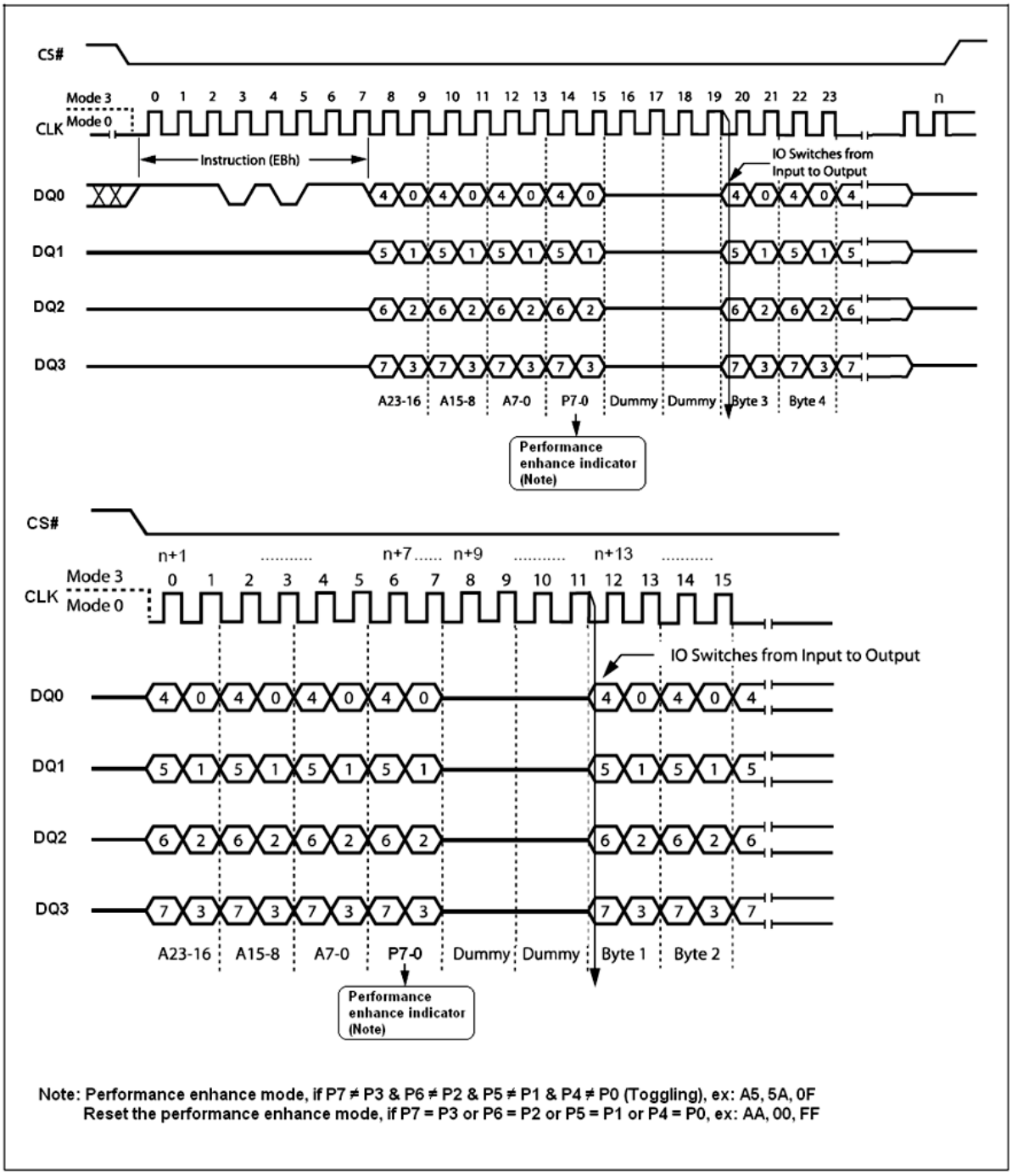
Another sequence of issuing Quad Input/Output FAST_READ (EBh) instruction especially useful in random access is : CS# goes low -> sending Quad Input/Output FAST_READ (EBh) instruction -> 24-bit address interleave on DQ_3, DQ_2, DQ_1 and DQ_0 -> performance enhance toggling bit P[7:0] -> 4 dummy clocks -> data out interleave on DQ_3, DQ_2, DQ_1 and DQ_0 till CS# goes high -> CS# goes low (reduce Quad Input/Output FAST_READ (EBh) instruction) -> 24-bit random access address, as shown in Quad Input / Output Fast Read Enhance Performance Mode Sequence Diagram figure.

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. These commands will reset the performance enhance mode. And afterwards CS# is raised or issuing FFh command (CS# goes high -> CS# goes low -> sending FFh -> CS# goes high) instead of no toggling, 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.

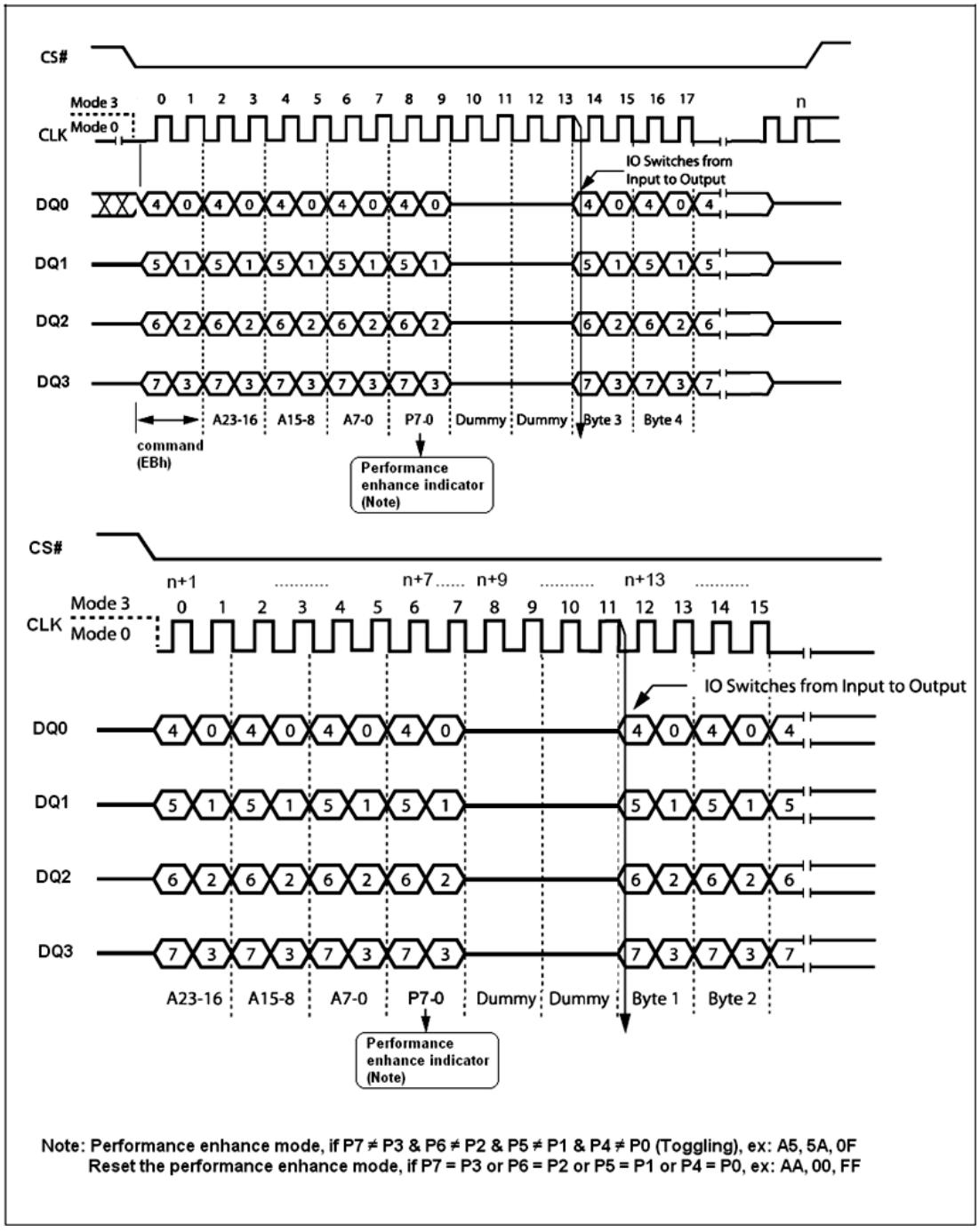
The instruction sequence is shown in Quad Input / Output Fast Read Enhance Performance Mode Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 34. 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 35. Quad Input / Output Fast Read Enhance Performance Mode Sequence in QPI Mode



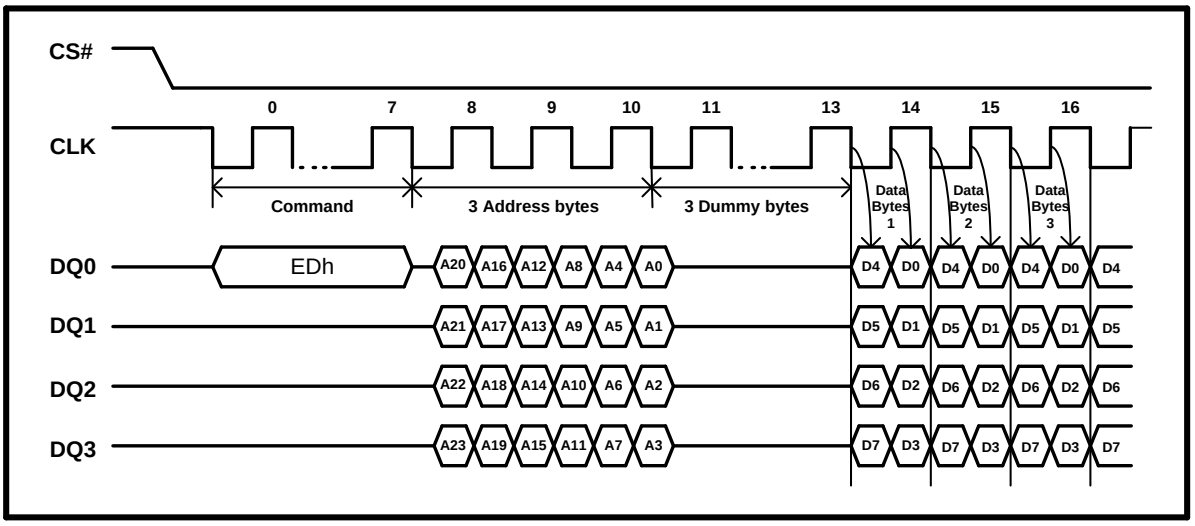
Note: Performance enhance mode, if P7 ≠ P3 & P6 ≠ P2 & P5 ≠ P1 & P4 ≠ P0 (Toggling), ex: A5, 5A, 0F
Reset the performance enhance mode, if P7 = P3 or P6 = P2 or P5 = P1 or P4 = P0, ex: AA, 00, FF

DDR Quad Input / Output FAST_READ (EDh)

The DDR Quad Input / Output FAST_READ (EDh) instruction enable Double Data Rate throughput on quad I/O of Serial Flash in read mode. The address (interleave on 4 I/O pins) is latched on both rising and falling edge of CLK, and data (interleave on 4 I/O pins) shift out on both rising and falling edge on CLK at a maximum frequency F_R. The 8-bit address can be latched-in at one clock, and 8-bit data can be read out at one clock, which means four bits at rising edge of clock, the other four bits at falling edge of clock. The first address byte 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 DDR Quad Input / Output FAST_READ (EDh) instruction. The address counter rolls over 0 when the highest address has been reached. Once writing DDR Quad Input / Output FAST_READ (EDh) instruction, the following address/dummy/data out will perform as 8-bit instead of previous 1-bit.

The sequence of issuing DDR Quad Input / Output FAST_READ (EDh) instruction is : CS# goes low -> sending DDR Quad Input / Output FAST_READ (EDh) instruction (1-bit per clock) -> 24-bit address interleave on DQ3, DQ2, DQ1 and DQ0 (8-bit per clock) -> 3 dummy byte (3 clocks) -> data out interleave on DQ3, DQ2, DQ1 and DQ0 (8-bit per clock) -> to end DDR Quad Input / Output FAST_READ (EDh) operation can use CS# to high at any time during data out, as shown in DDR Quad Input / Output FAST_READ Instruction Sequence Diagram figure.

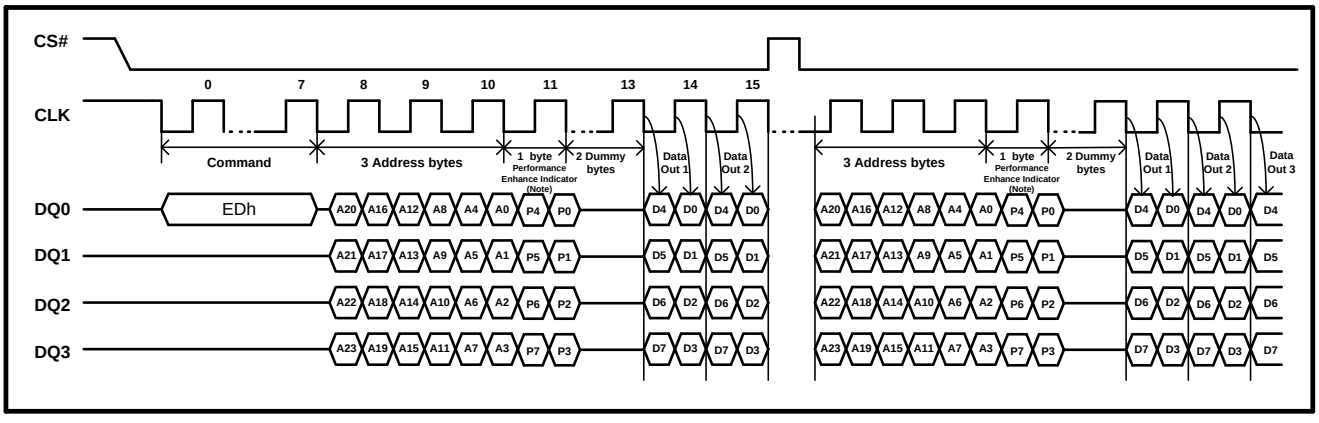
Figure 36. DDR Quad Input / Output FAST_READ Instruction Sequence Diagram



Another sequence of issuing enhanced mode of DDR Quad Input / Output FAST_READ (EDh) instruction especially useful in random access is : CS# goes low -> sending DDR Quad Input / Output FAST_READ (EDh) instruction (1-bit per clock) -> 3-byte address interleave on DQ3, DQ2, DQ1 and DQ0 (8-bit per clock) -> performance enhance toggling bit P[7:0} -> 2 dummy byte -> data out (8-bit per clock) still CS# goes high -> CS# goes low (eliminate Quad Input / Output FAST_READ) -> 24-bit random access address, as shown in DDR Quad Input / Output Fast Read Enhance Performance Mode Sequence Diagram figure.

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

Figure 37. DDR Quad Input / Output Fast Read Enhance Performance Mode Sequence Diagram



Note:

1. Performance enhance mode, if P7≠P3 & P6≠P2 & P5≠P1 & P4≠P0 (Toggling).

DDR Read Burst with Wrap (DQRB) (DCh)

The DDR Read Burst with Wrap (DCh) instruction (DDR Read Burst with Wrap with Wrap Instruction Sequence Diagram figure) enable Double Data Rate throughput on quad I/O of Serial Flash in read mode. The address (interleave on 4 I/O pins) is latched on both rising and falling edge of CLK, and data (interleave on 4 I/O pins) shift out on both rising and falling edge on CLK at a maximum frequency F_R. The 8-bit address can be latched-in at one clock, and 8-bit data can be read out at one clock, which means four bits at rising edge of clock, the other four bits at falling edge of clock. Once writing DDR Read Burst with Wrap (DCh) instruction, the following address/dummy/data out will perform as 8-bit instead of previous 1-bit.

The sequence of issuing DDR Read Burst with Wrap (DCh) instruction is: CS# goes low -> sending DDR Read Burst with Wrap (DCh) instruction (1-bit per clock) -> 24 bit address interleave on DQ0 (2-bit per clock) -> 1 dummy bytes (4 clocks) -> data out interleave on DQ1 (2-bit per clock) -> to end DDR Read Burst with Wrap (DCh) operation can use CS# to high at any time during data out.

During DDR Read Burst with Wrap, the first address byte can be at any location. The internal address point automatically increments until the last byte of the burst reached, then jumps to first byte of the burst. All bursts are aligned to addresses within the burst length, see Burst Address Range table. For example, if the burst length is 8 bytes, and the start address is 06h, the burst sequence should be: 06h, 07h, 00h, 01h, 02h, 03h, 04h, 05, 06, etc. The pattern would repeat until the command was terminated by pulling CS# as high status.

The instruction sequence is shown in DDR Read Burst with Wrap Instruction Sequence Diagram in QPI mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 38. DDR Read Burst with Wrap with Wrap Instruction Sequence Diagram

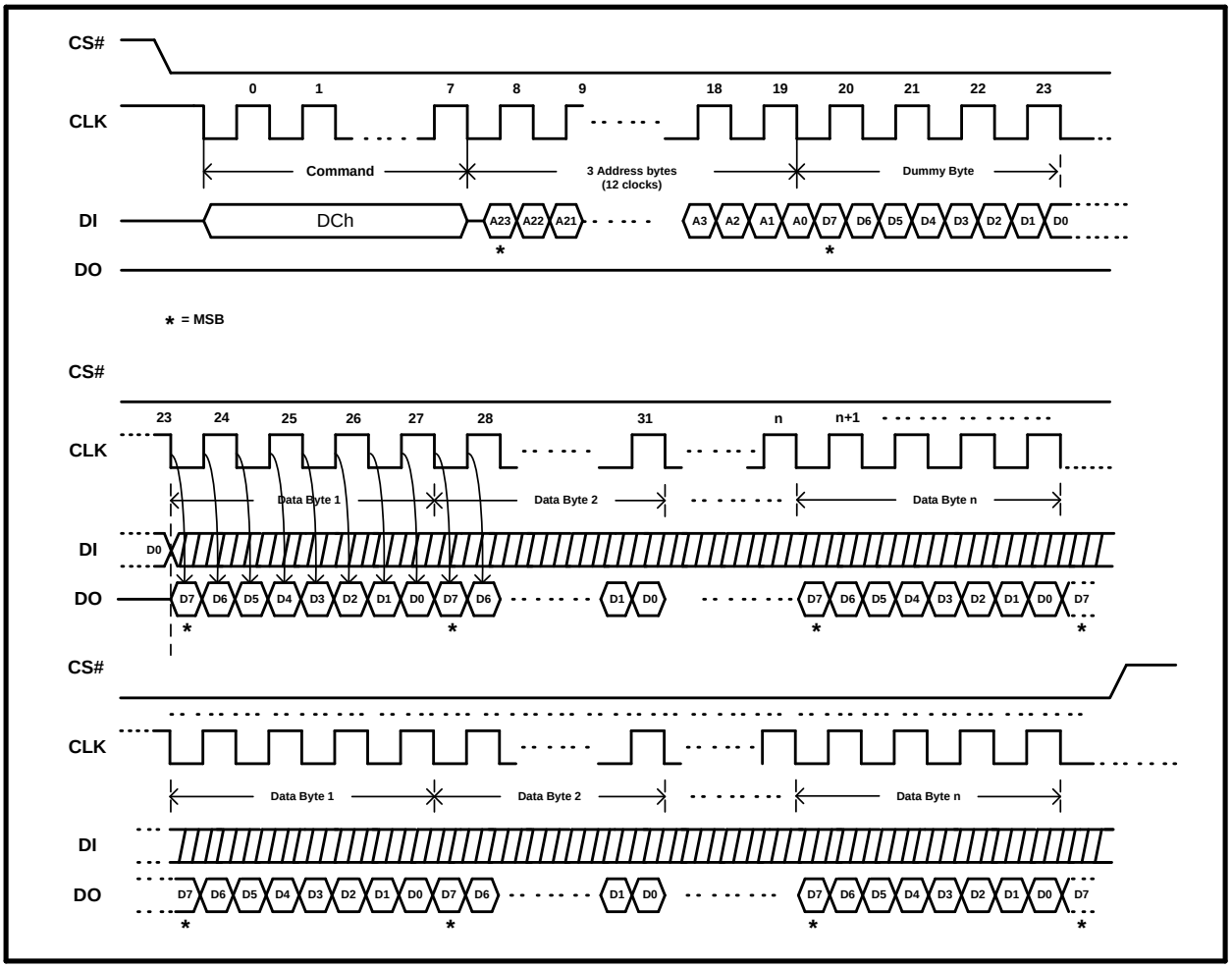
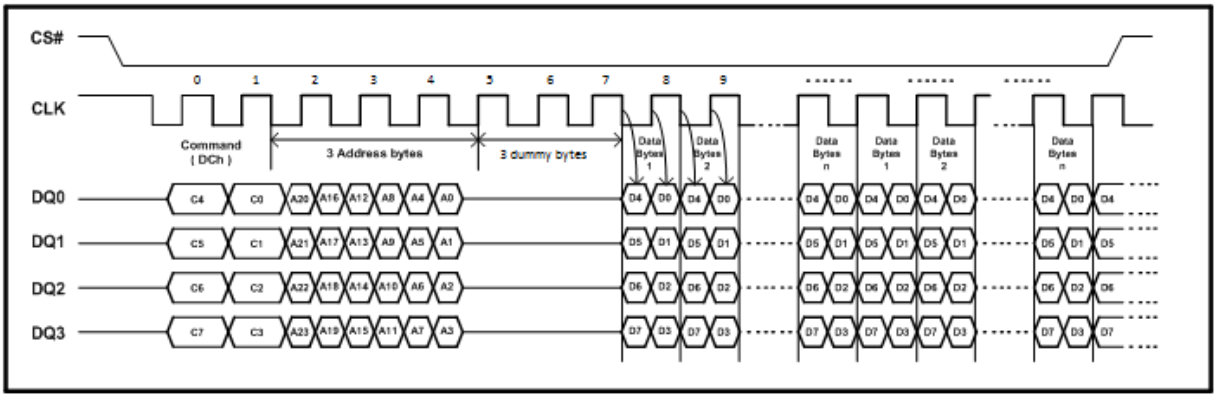


Figure 39. DDR Read Burst with Wrap Instruction Sequence Diagram in QPI mode



Read Burst (0Ch)

This device supports Read Burst with wrap in both SPI and QPI mode. To execute a Read Burst with wrap operation the host drivers CS# low, and sends the Read Burst with wrap (0Ch) command cycle, followed by three address bytes and one dummy byte (8 clocks) in SPI mode (Read Burst Instruction Sequence Diagram figure) or default three dummy bytes (6 clocks) in QPI mode (Read Burst Instruction Sequence Diagram in QPI mode figure).

After the dummy byte, the device outputs data on the falling edge of the CLK signal starting from the specific address location. The data output stream is continuous through all addresses until terminated by a low-to high transition of CS# signal.

During Read Burst, the internal address point automatically increments until the last byte of the burst reached, then jumps to first byte of the burst. All bursts are aligned to addresses within the burst length, see Burst Address Range table. For example, if the burst length is 8 bytes, and the start address is 06h, the burst sequence should be: 06h, 07h, 00h, 01h, 02h, 03h, 04h, 05, 06, etc. The pattern would repeat until the command was terminated by pulling CS# as high status.

The instruction sequence is shown in Read Burst Instruction Sequence Diagram in QPI mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Table 13. Burst Address Range

Burst length	Burst wrap (A[7:A0]) address range
8 Bytes (default)	00-07H, 08-0FH, 10-17H, 18-1FH...
16 Bytes	00-0FH, 10-1FH, 20-2FH, 30-3FH...
32 Bytes	00-1FH, 20-3FH, 40-5FH, 60-7FH...
64 Bytes	00-3FH, 40-7FH, 80-BFH, C0-FFH

Figure 40. Read Burst Instruction Sequence Diagram

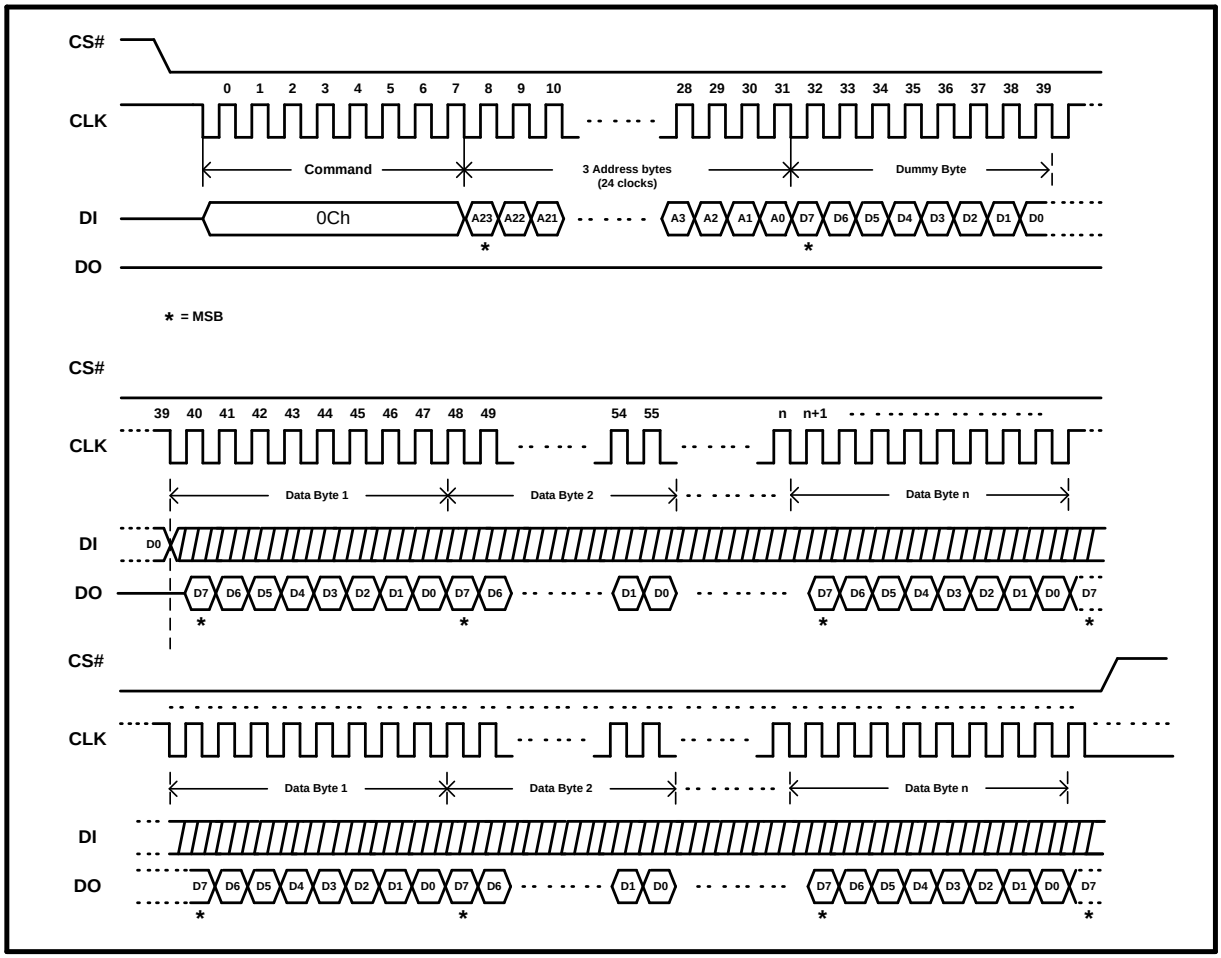
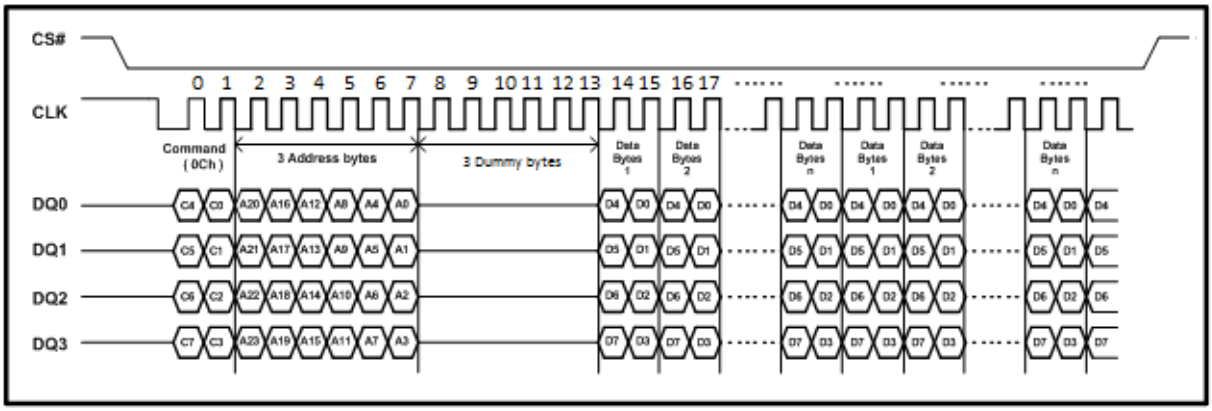


Figure 41. Read Burst Instruction Sequence Diagram in QPI mode



Write Status Register 2 (31h/01h)

The Write Status Register 2 (31h) command can be used to set SPL0/SPL1/SPL2 OTP bits, QE bit and CMP bit. To set these bits to the host driver CS# low, sends the Write Status Register 2 (31h) and one data byte, then drivers CS# high, In QPI mode, a cycle is two nibbles, or two clocks, long, most significant nibble first.

01h (WRSR) command also can set status register 2.

The instruction sequence is shown in Write Status Register 2 Instruction Sequence Diagram in QPI mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 42. Write Status Register 2 Instruction Sequence Diagram

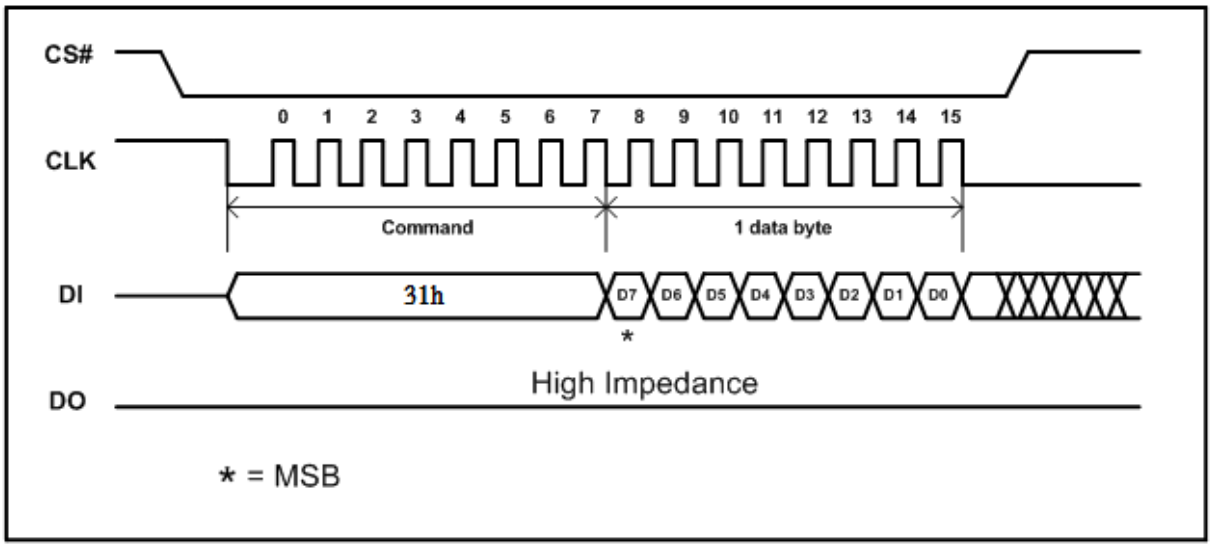
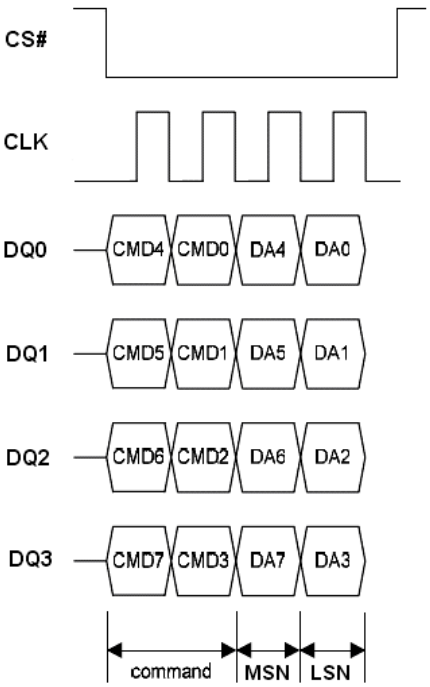


Figure 43. Write Status Register 2 Instruction Sequence Diagram in QPI mode



Note: MSN = Most Significant Nibble,
LSN = Least Significant Nibble

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

The Write Status Register 3 (C0h/11h/11h) command can be used to set output drive strength in I/O pins, HOLD#/RESET# pin selection and Burst read length setting. To set these bits, the host driver CS# low, sends the Write Status Register 3 (C0h or 11h) and one data byte, then drivers CS# high.

In QPI mode, a cycle is two nibbles, or two clocks, long, most significant nibble first. The instruction sequence is shown in Write Status Register 3 Instruction Sequence Diagram in QPI mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

01h (WRSR) command also can set status register 3.

Figure 44. Write Status Register 3 Instruction Sequence Diagram

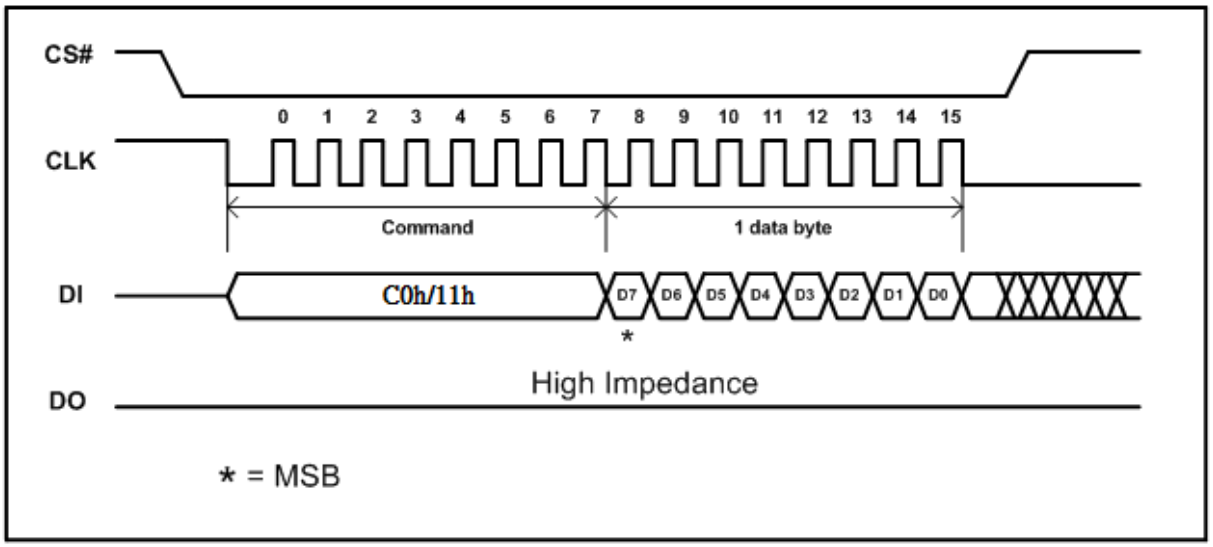
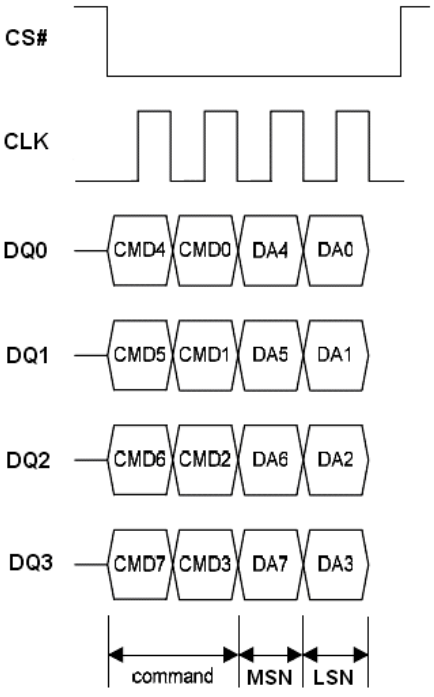


Figure 45. Write Status Register 3 Instruction Sequence Diagram in QPI mode



Note: MSN = Most Significant Nibble, LSN = Least Significant Nibble

Page Program (PP) (02h)

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

The Page Program (PP) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, three address bytes and at least one data byte on Serial Data Input (DI). If the 8 least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same page (from the address whose 8 least significant bits (A7-A0) are all zero). Chip Select (CS#) must be driven Low for the entire duration of the sequence.

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

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

As soon as Chip Select (CS#) is driven high, the self-timed Page Program cycle (whose duration is t_{PP}) is initiated. While the Page Program cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Page Program cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.

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

The instruction sequence is shown in Program Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 46. Page Program Instruction Sequence Diagram

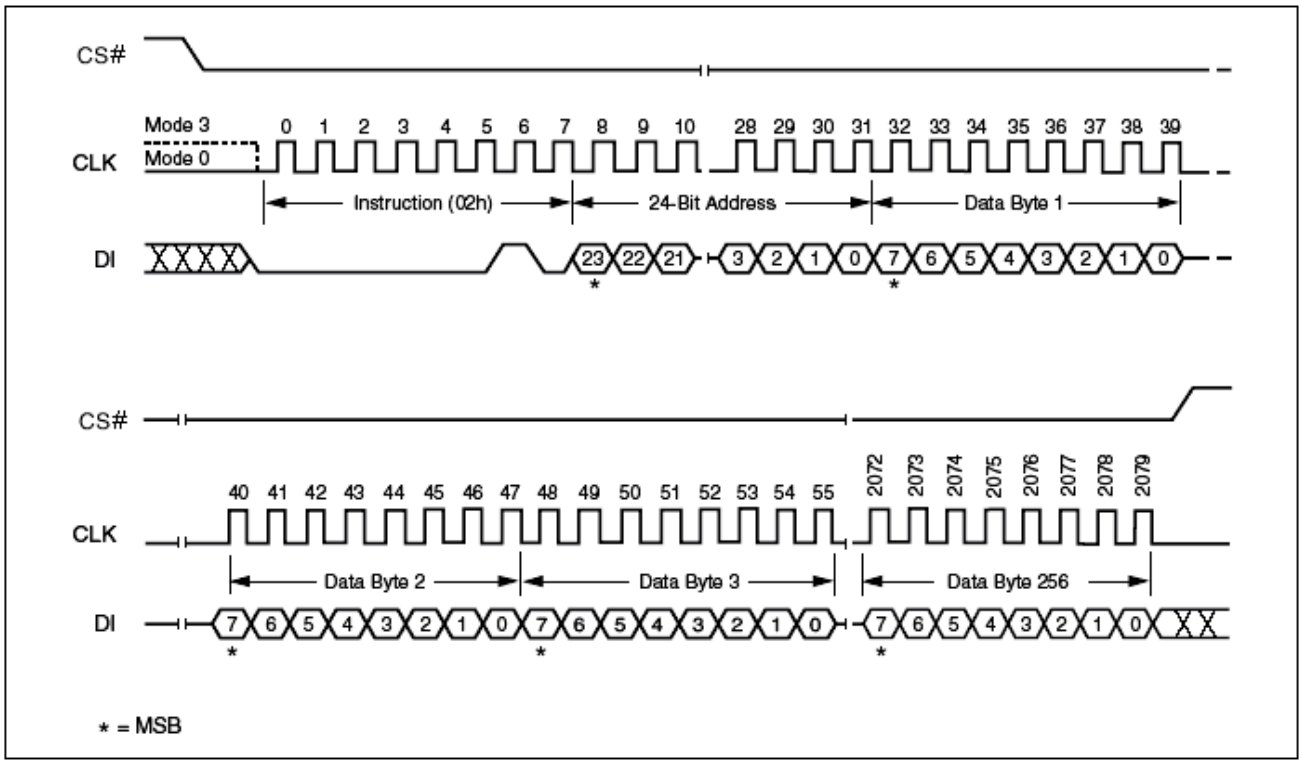
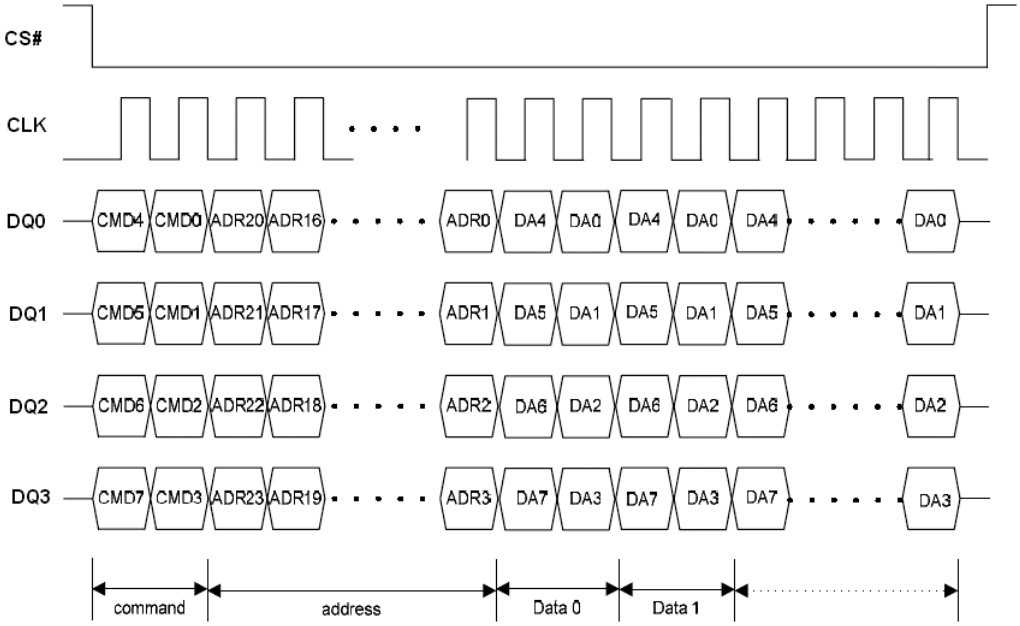


Figure 47. Program Instruction Sequence in QPI Mode

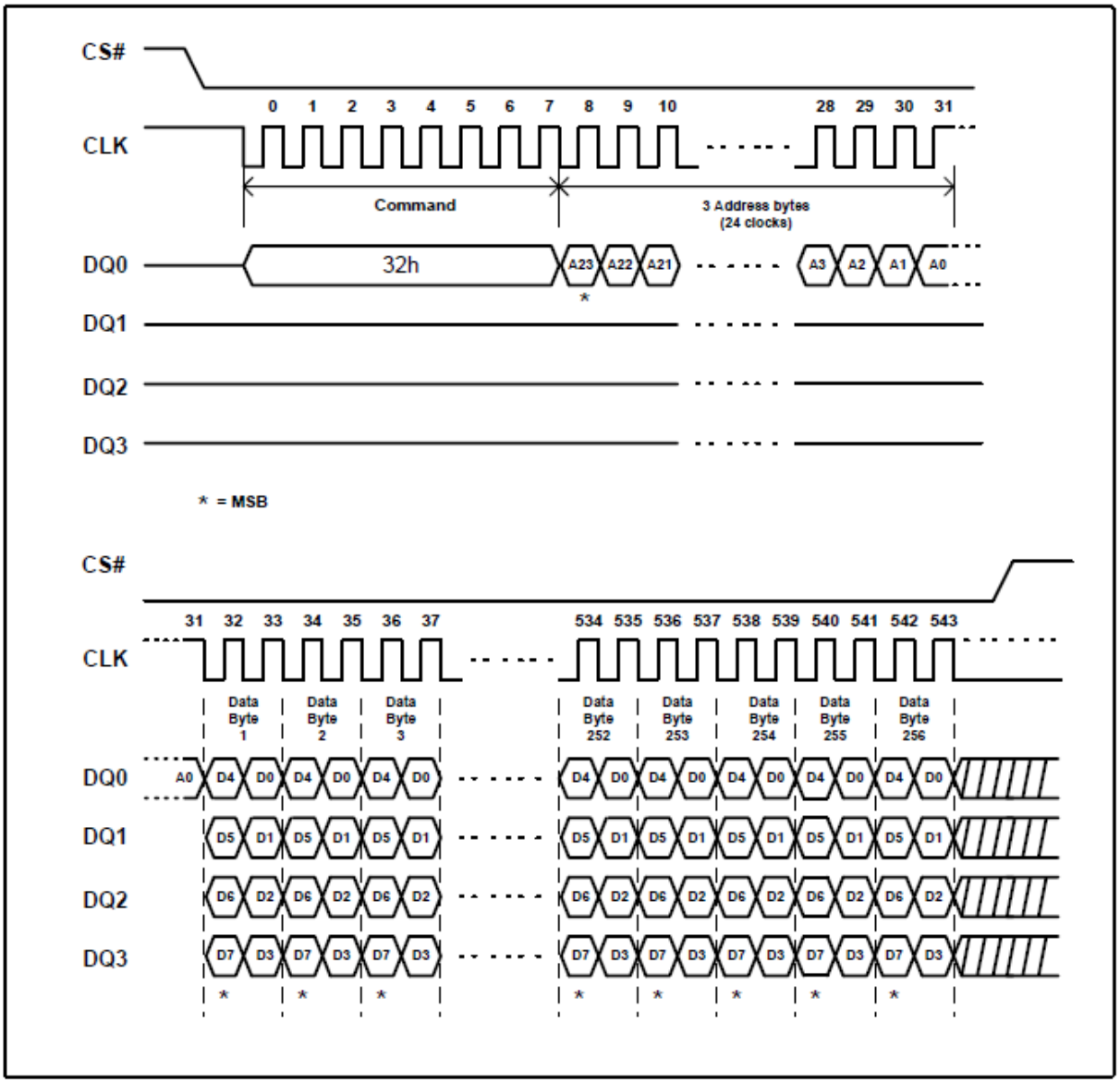


Quad Input Page Program (QPP) (32h)

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: DQ0, DQ1, DQ2 and DQ3. The Quad Page Program can improve performance for PROM Programmer and applications that have slow clock speeds < 5MHz. Systems with faster clock speed will not realize much benefit for the Quad Page Program instruction since the inherent page program time is much greater than the time it take to clock- in the data.

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

Figure 48. Quad Input Page Program Instruction Sequence Diagram (SPI Mode only)



DDR Page Program (DPP) (D2h)

The DDR Page Program (DPP) instruction enable Double Data Rate throughput on quad I/O of Serial Flash in Program mode. The address (interleave on 4 I/O pins) is latched on both rising and falling edge of CLK, and data (interleave on 4 I/O pins) shift out on both rising and falling edge on CLK at a maximum frequency F_R. The 8-bit address can be latched-in at one clock, and 8-bit data can be read out at one clock, which means four bits at rising edge of clock, the other four bits at falling edge of clock. Once writing DDR Page Program (DPP) instruction, the following address /data in will perform as 8-bit instead of previous 1-bit.

DDR Page Program (DPP) 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 sequence of issuing DDR Page Program (D2h) instruction is: CS# goes low -> sending DDR Page Program (D2h) instruction (1-bit per clock) -> 24-bit address interleave on DQ0 (2-bit per clock) -> data in interleave on DQ0 (2-bit per clock) -> to end DDR Page Program (D2h) operation can use CS# to high at any time during data out.

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

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

As soon as Chip Select (CS#) is driven high, the self-timed DDR Page Program cycle (whose duration is t_{DPP}) is initiated. While the DDR 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 DDR 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 DDR Page Program (PP) instruction applied to a page which is protected by the Block Protect bits (see Protected Area Sizes Sector Organization table) is not executed.

The instruction sequence is shown in DDR Page Program Instruction Sequence Diagram in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 49. DDR Page Program Instruction Sequence Diagram

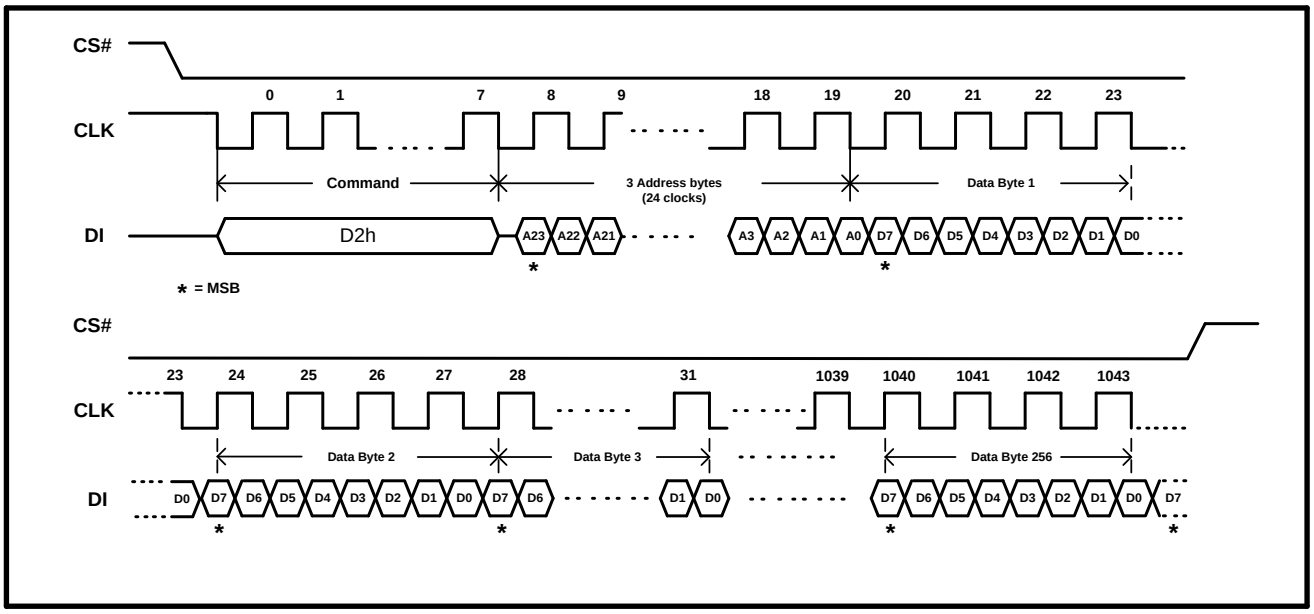
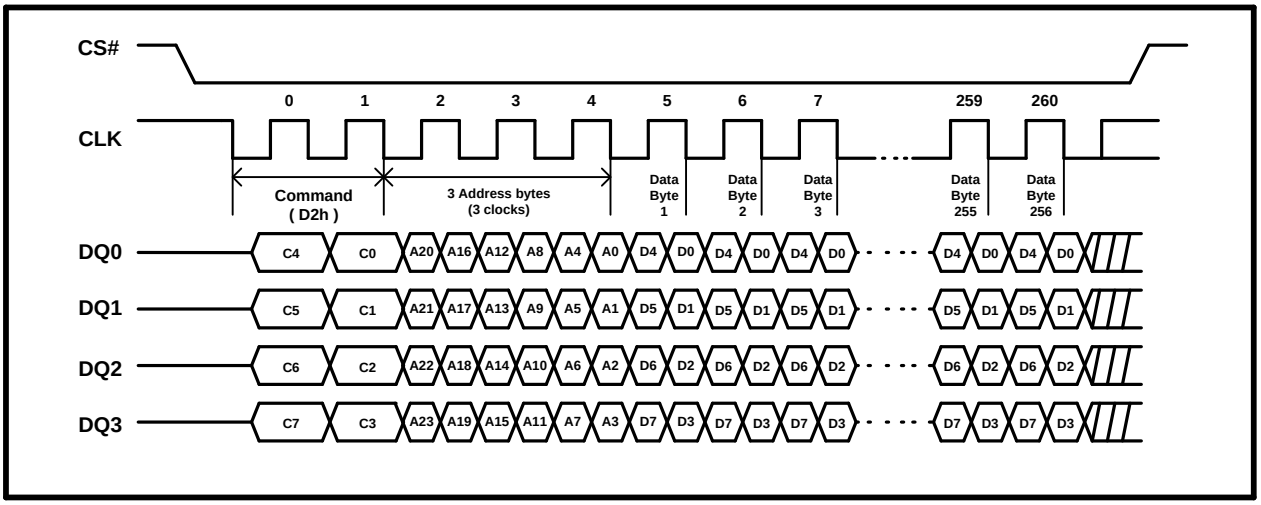


Figure 50. DDR Page Program Instruction Sequence Diagram in QPI Mode



Write Suspend (B0h/75h)

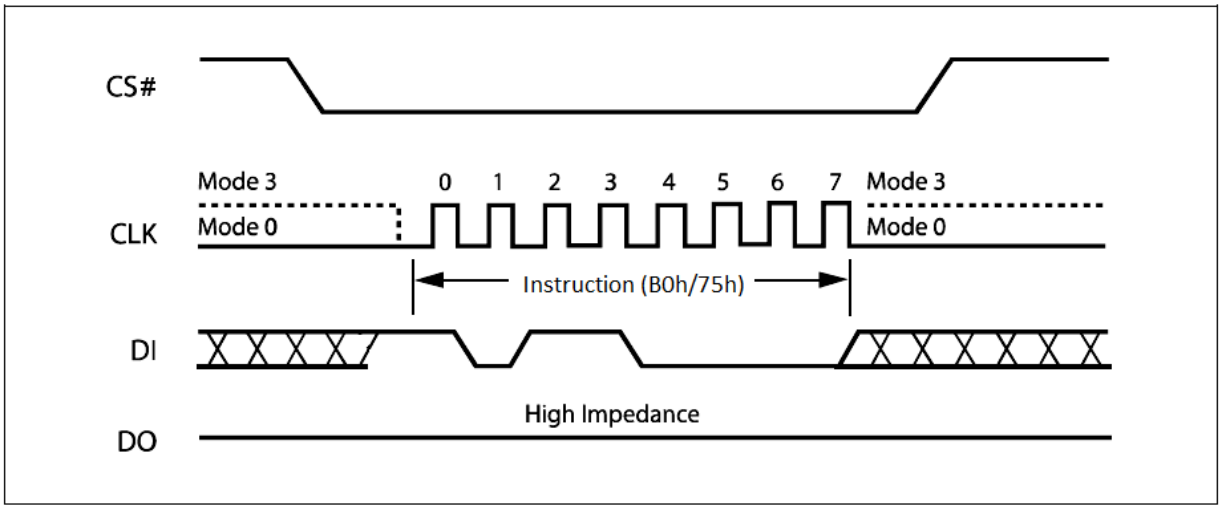
Write Suspend allows the interruption of Sector Erase, Block Erase or Page Program operations in order to read data in another portion of memory. The original operation can be continued with Write Resume command. The instruction sequence is shown in Write Suspend Instruction Sequence Diagram figure.

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

Suspend to suspend ready timing: 28 us.
Resume to another suspend timing: min 0.3 us, typ. 200 us.

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

Figure 51. 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 program or read any block that was not being erased. The device will ignore any programming commands pointing to the suspended sector(s). Any attempt to read from the suspended block will output unknown data because the Sector or Block Erase will be incomplete.

To execute a Write Suspend operation, the host drives CS# low, sends the Write Suspend command cycle (B0h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. The 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 or after issue program suspend command, latency time 28 us is needed before issue another command. For “Suspend to Read”, “Resume to Read”, “Resume to Suspend” timing specification please note Suspend to Read Latency, Resume to Read Latency and Resume to Suspend Latency figure.

Write Suspend During Page Programming

Issuing a Write Suspend instruction during Page Programming allows the host to erase any sector or read any page that is not being programmed. Erase command pointing to the suspend sector(s) will be ignored. Any attempt to read from the suspended page will output unknown data because the program will be incomplete.

To execute a Write Suspend operation, the host drives CS# low, sends the Write Suspend command cycle (B0h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. The 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 or after issue program suspend command, latency time 28 us is needed before issue another command. For “Suspend to Read”, “Resume to Read”, “Resume to Suspend” timing specification please note Suspend to Read Latency, Resume to Read Latency and Resume to Suspend Latency figure.

The instruction sequence is shown in Write Suspend / Resume Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 52. Suspend to Read Latency

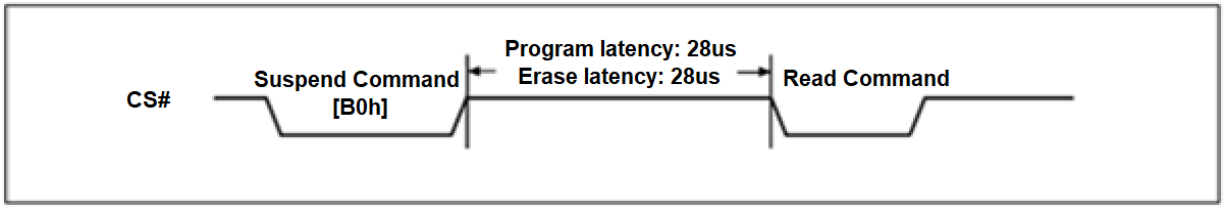


Figure 53. Resume to Read Latency

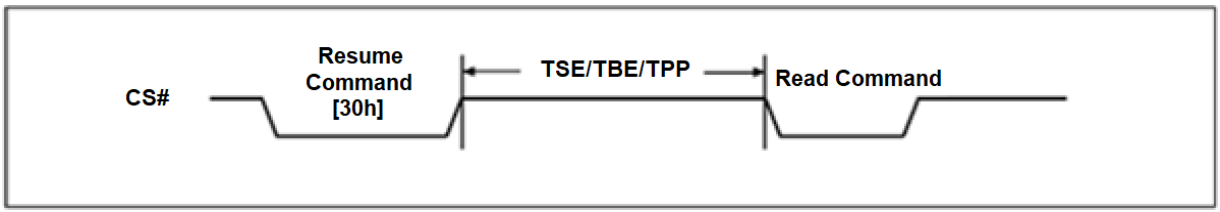
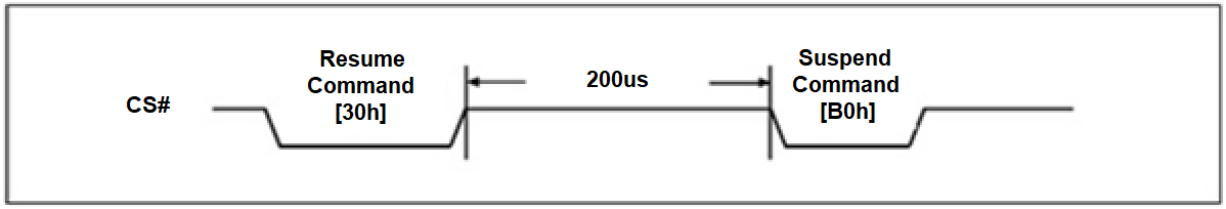


Figure 54. Resume to Suspend Latency



Write Resume (30h/7Ah)

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 Write Resume Instruction Sequence Diagram figure. To execute a Write Resume operation, the host drives CS# low, sends the Write Resume command cycle (30h or 7Ah), 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, 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 200 us.

The instruction sequence is shown in Write Suspend / Resume Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 55. Write Resume Instruction Sequence Diagram

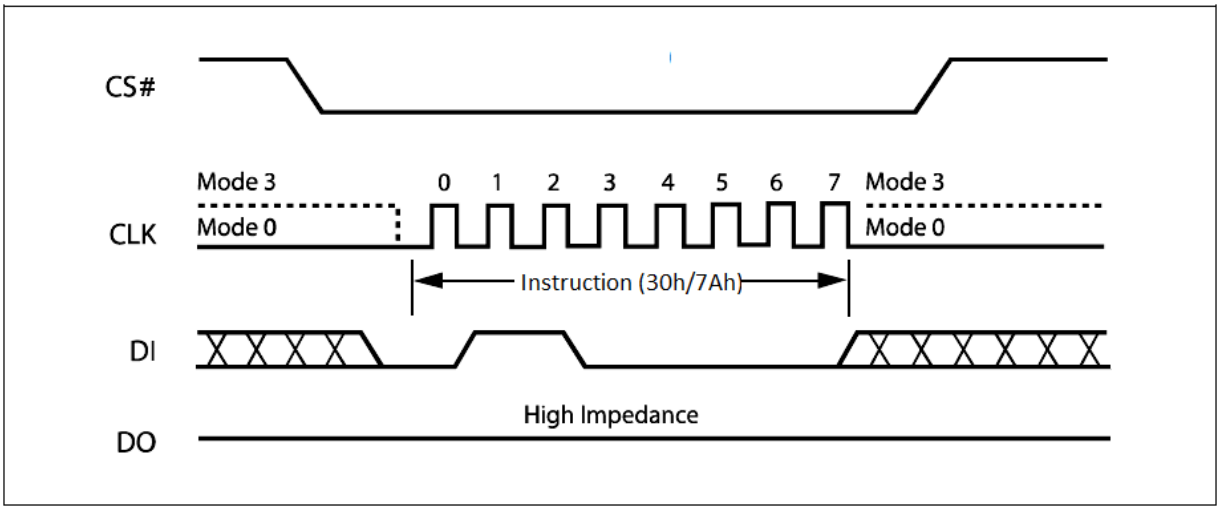


Figure 56. Write Suspend / Resume Instruction Sequence in QPI Mode

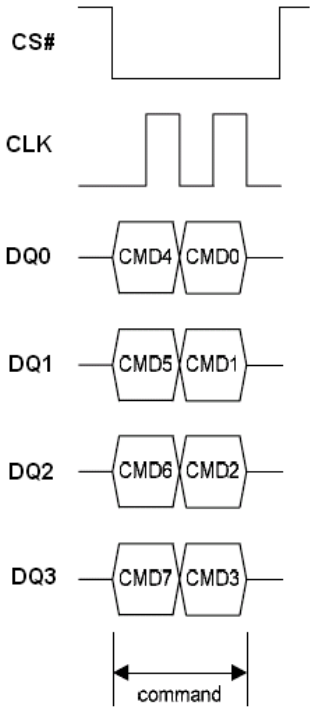
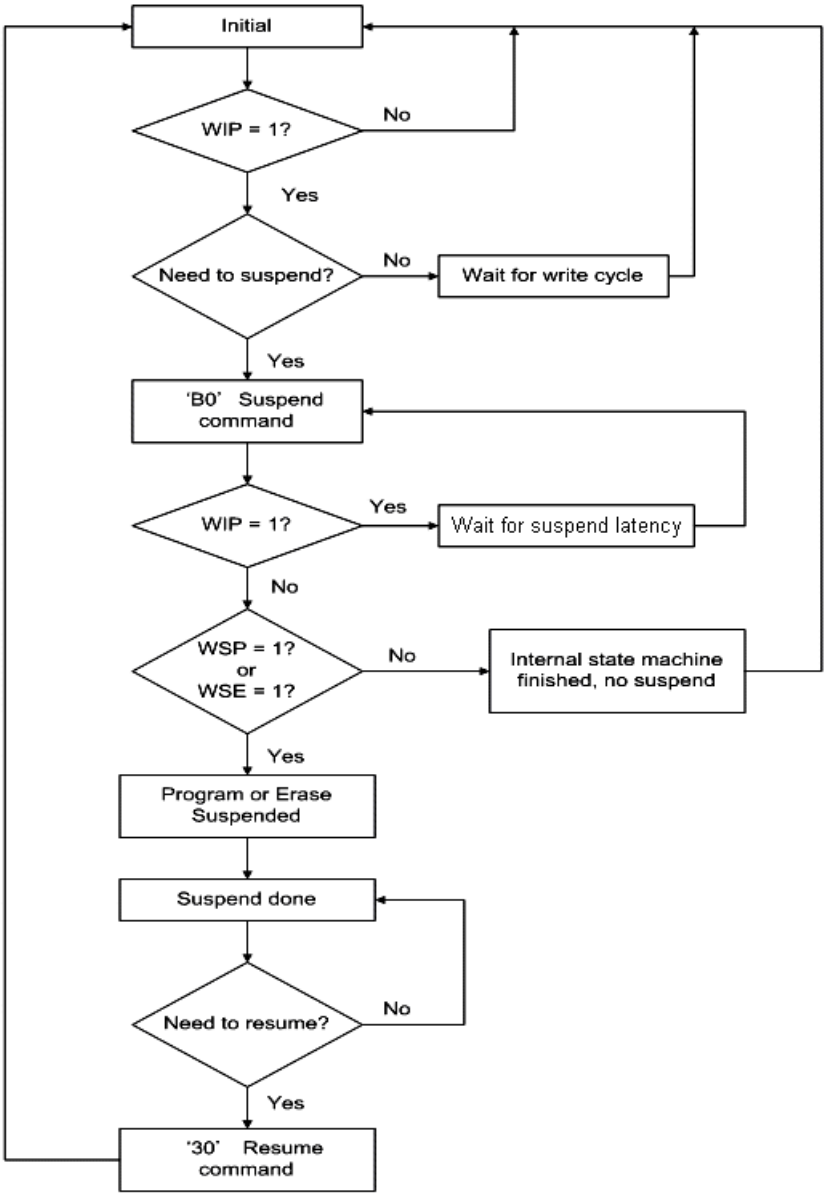


Figure 57. Write Suspend/Resume Flow



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 28 us 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).

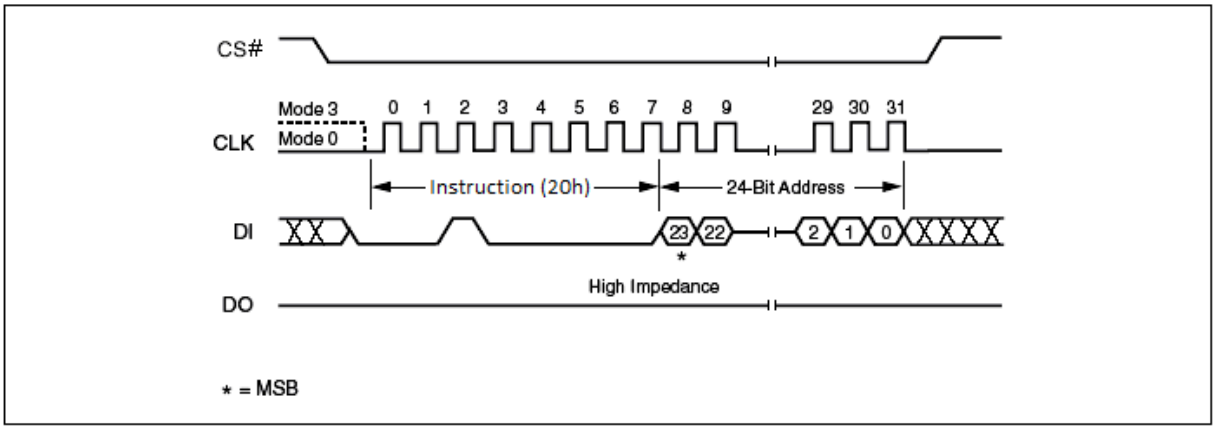
The Sector Erase (SE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Sector (see Uniform Block Sector Architecture table) is a valid address for the Sector Erase (SE) instruction. Chip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

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

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

The instruction sequence is shown in Block / Sector Erase Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 58. Sector Erase Instruction Sequence Diagram



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

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

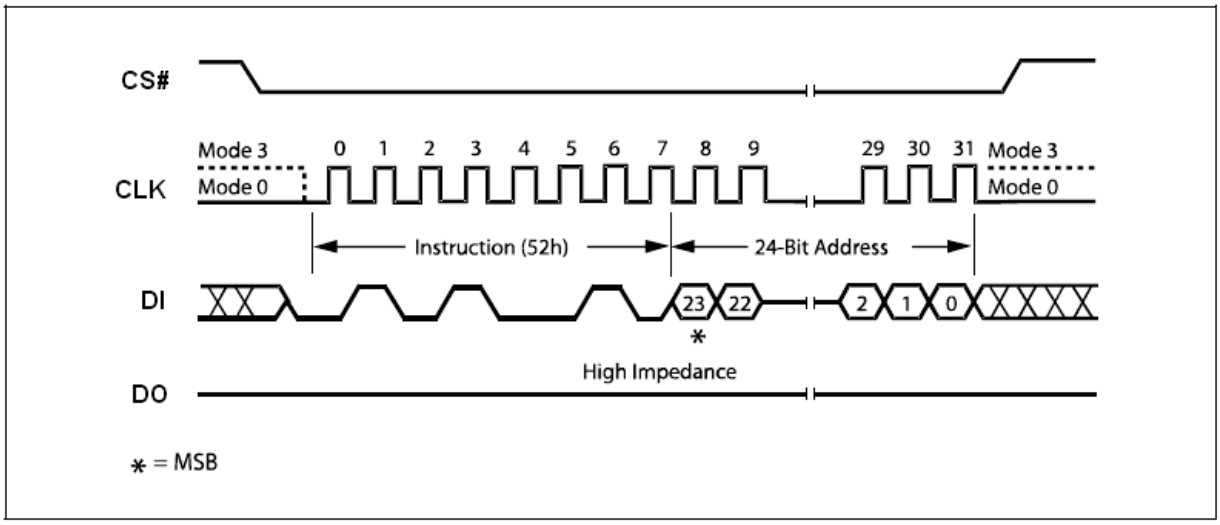
The Half Block Erase (HBE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Block (see Uniform Block Sector Architecture table) is a valid address for the Half Block Erase (HBE) instruction.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

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

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

The instruction sequence is shown in Block / Sector Erase Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 59. 32KB Half Block Erase Instruction Sequence Diagram



64KB Block Erase (BE) (D8h)

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

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

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

A Block Erase (BE) instruction applied to a block which is protected by the Block Protect (CMP, 4KBL, TB, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) or Boot Lock feature is not executed.

The instruction sequence is shown in Block / Sector Erase Instruction Sequence in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 60. 64KB Block Erase Instruction Sequence Diagram

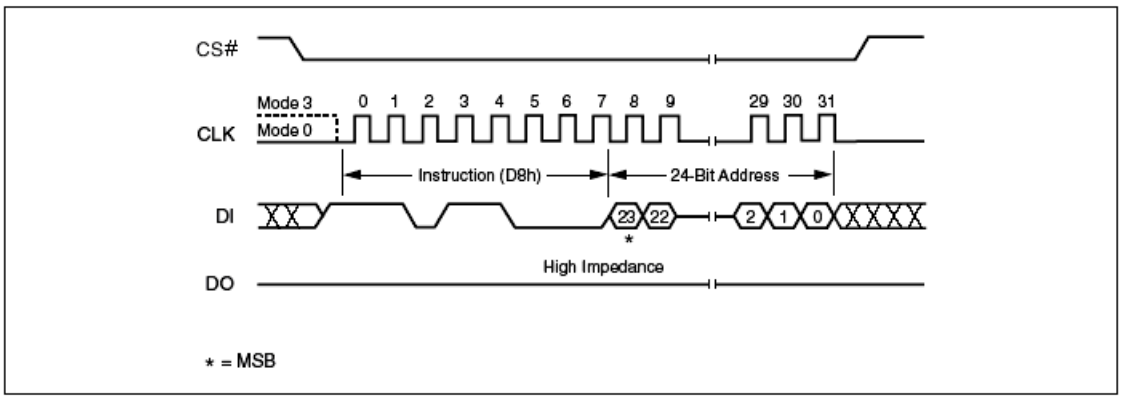
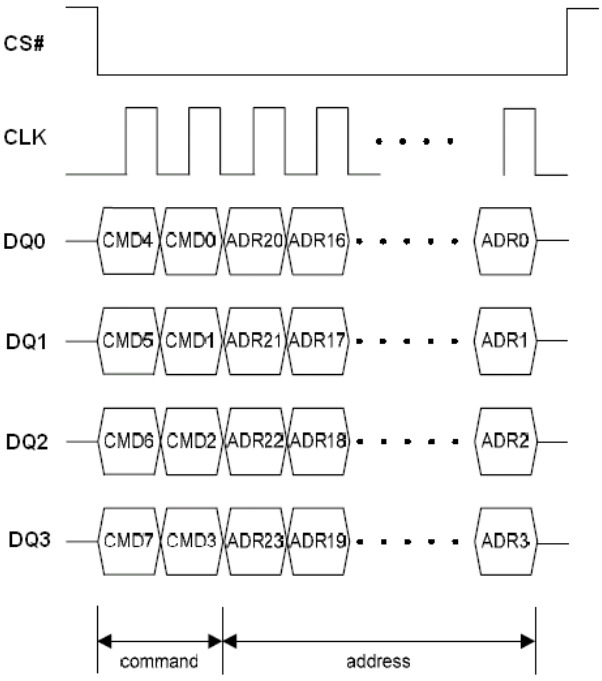


Figure 61. Block / Sector Erase Instruction Sequence in QPI Mode



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

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

The instruction sequence is shown in Chip Erase Sequence under EQPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 62. Chip Erase Instruction Sequence Diagram

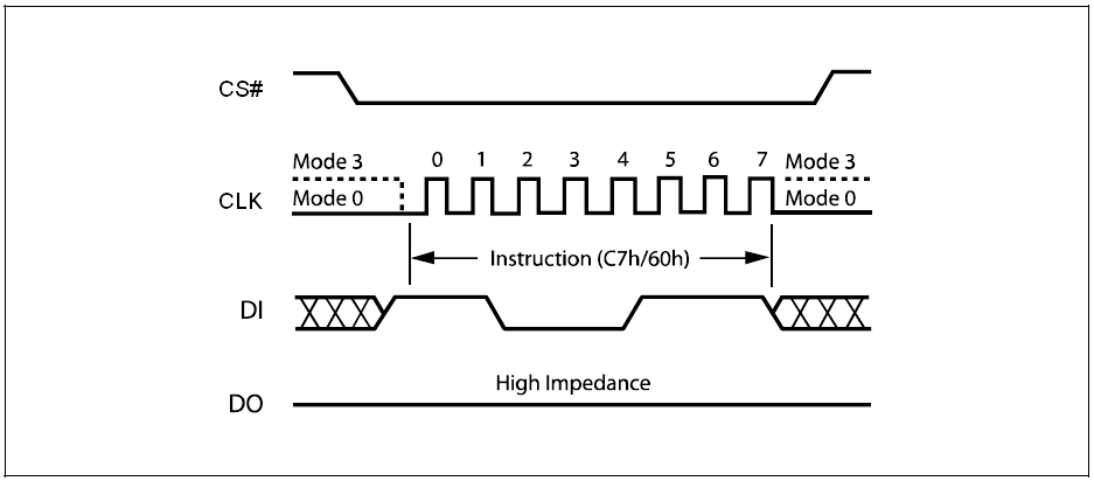
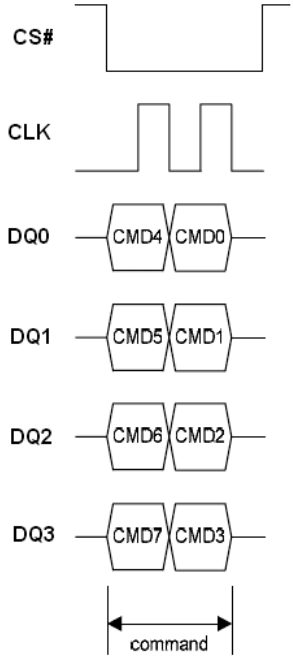


Figure 63. Chip Erase Sequence under EQPI Mode



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 DC Characteristics table.)

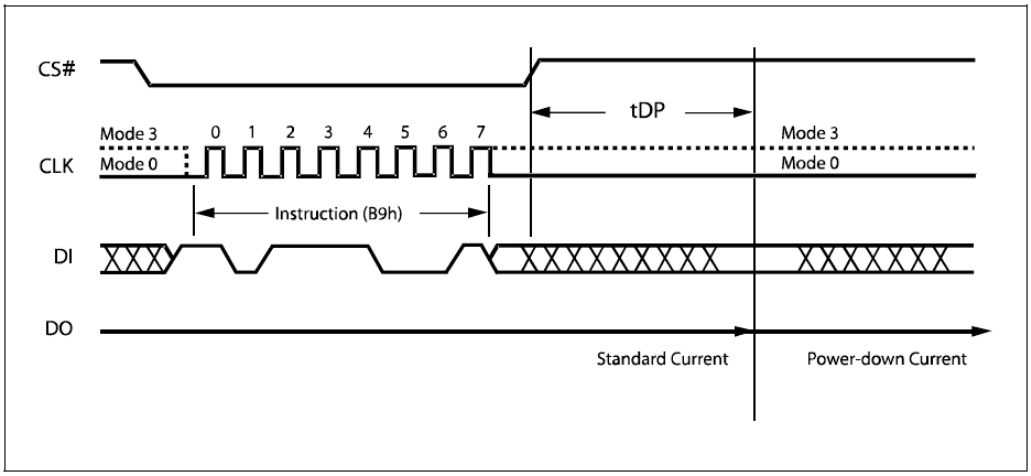
Once the device has entered the Deep Power-down mode, all instructions are ignored except the Release from Deep Power-down, Read Device ID (RDI) and Software Reset instruction which release the device from this mode. The Release from Deep Power-down and Read Device ID (RDI) instruction also allows the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

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

The instruction sequence is shown in Deep Power-down Instruction Sequence Diagram figure. Chip Select (CS#) must be driven High after the eighth bit of the instruction code has been latched in, otherwise the Deep Power-down (DP) instruction is not executed. As soon as Chip Select (CS#) is driven High, it requires a delay of t_{DP} before the supply current is reduced to I_{CC2} and the Deep Power-down mode is entered.

Any Deep Power-down (DP) instruction, while an Erase, Program or Write cycle is in progress, is rejected without having any effects on the cycle that is in progress.

Figure 64. Deep Power-down Instruction Sequence Diagram



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

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

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

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

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

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

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

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

Figure 65. Release Power-down Instruction Sequence Diagram

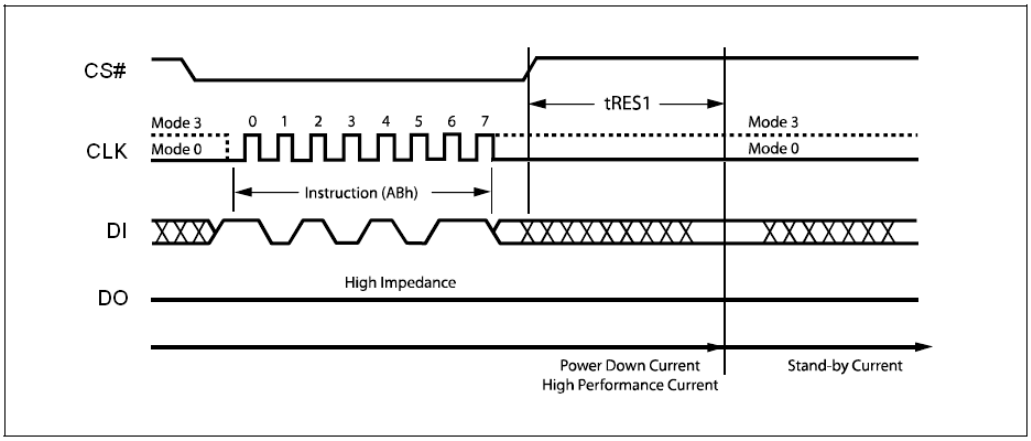
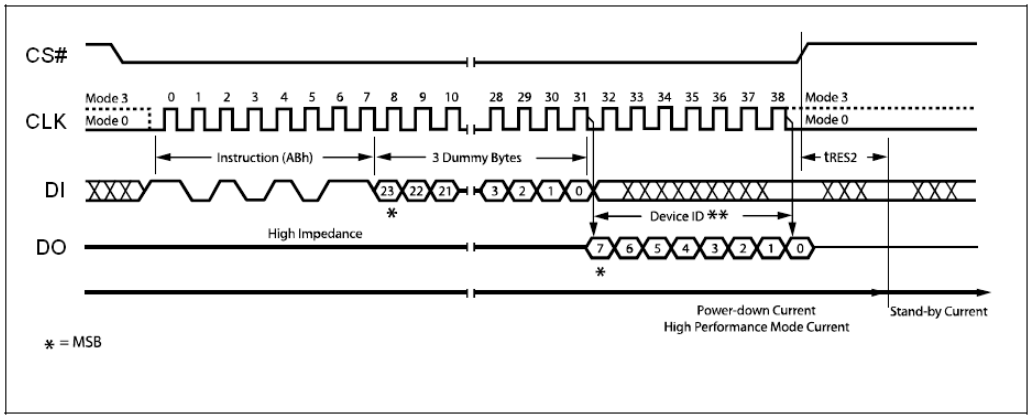


Figure 66. Release Power-down / Device ID Instruction Sequence Diagram



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 (1Ch) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Read Manufacturer / Device ID Diagram figure. The Device ID values for the device are listed in Manufacturer and Device Identification table. If the 24-bit address is initially set to 000001h the Device ID will be read first

The instruction sequence is shown in Read Manufacturer / Device ID Diagram in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 67. Read Manufacturer / Device ID Diagram

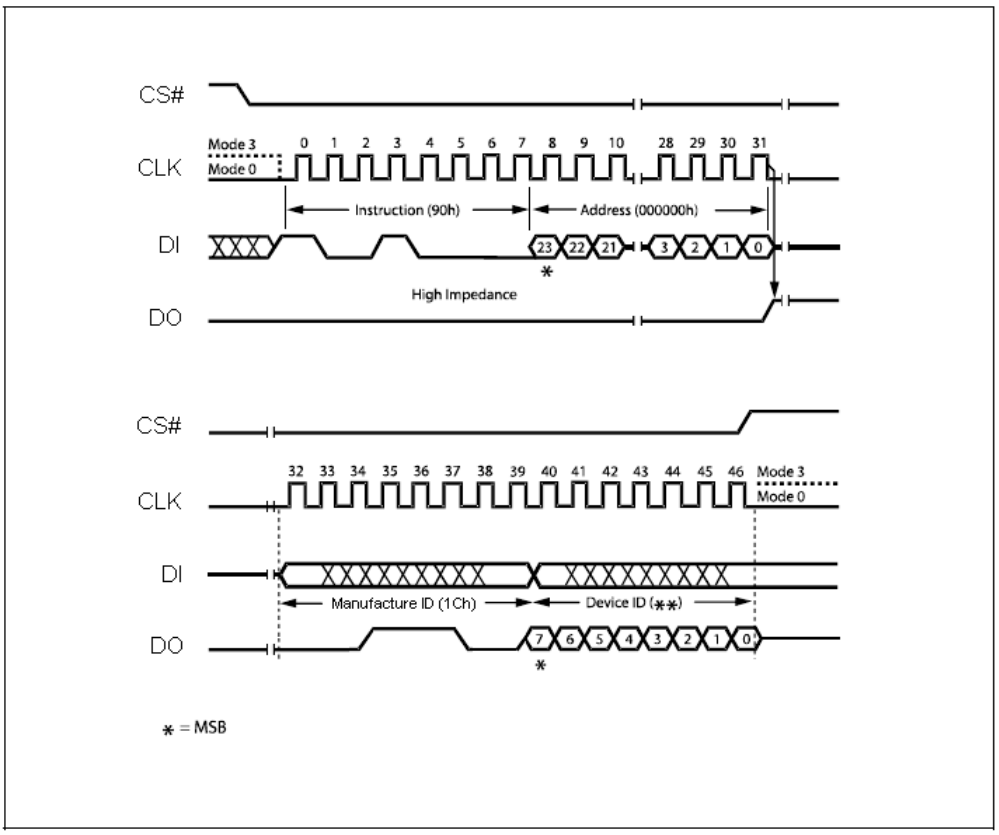
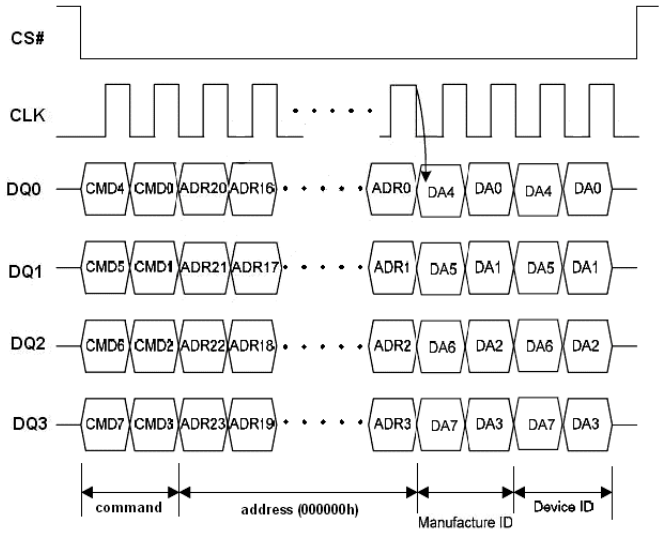


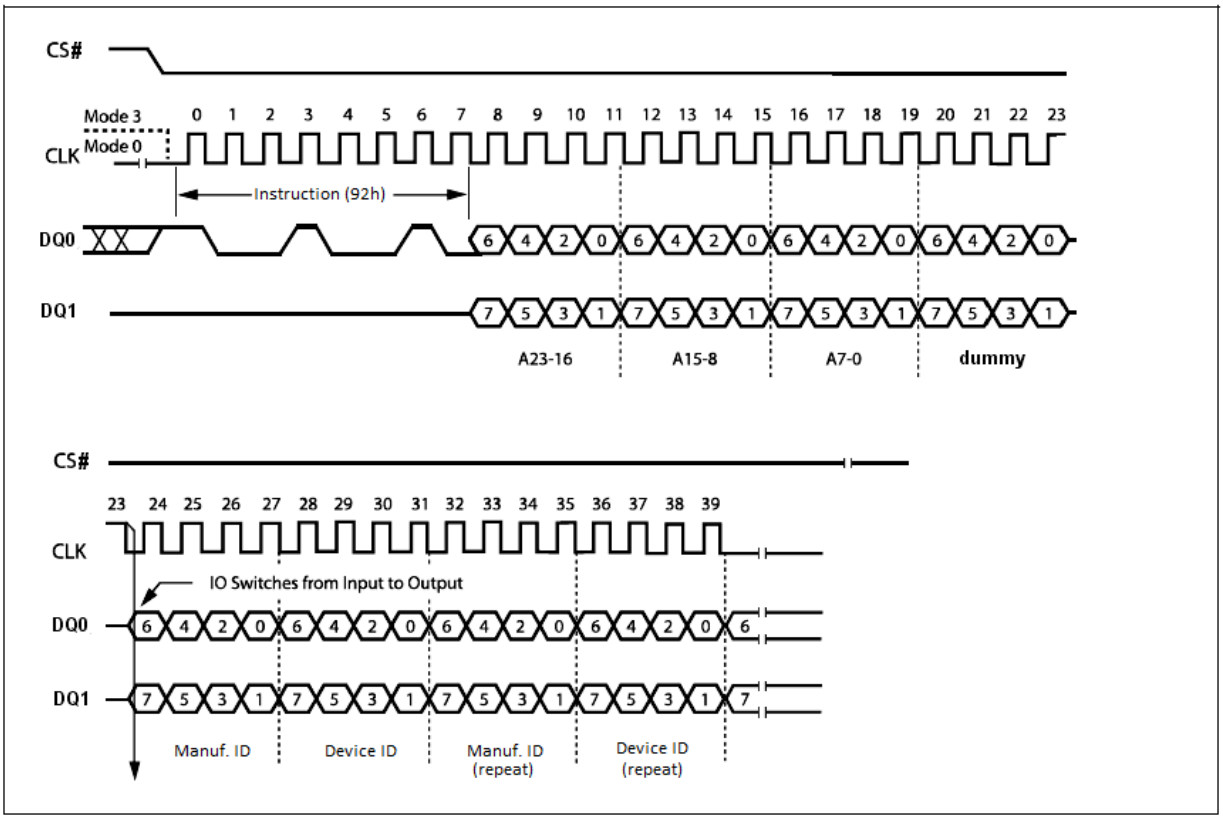
Figure 68. Read Manufacturer / Device ID Diagram in QPI Mode



Read Manufacturer / Device ID by Dual I/O (92h)

The Read Manufacturer/Device ID by Dual I/O instruction is very similar to the Dual Input / Output FAST_READ instruction. The instruction is initiated by driving the CS# pin low and shifting the instruction code “92h” followed by a 24-bit address (A23-A0) of 000000h by using DQ0 and DQ1 and one byte dummy . After which, the Manufacturer ID for (1Ch) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first byte using DQ0 and DQ1 as shown in Read Manufacturer / Device ID by Dual I/O Diagram figure. The Device ID values for the device are listed in Manufacturer and Device Identification table. If the 24-bit address is initially set to 000001h, the Device ID will be read first.

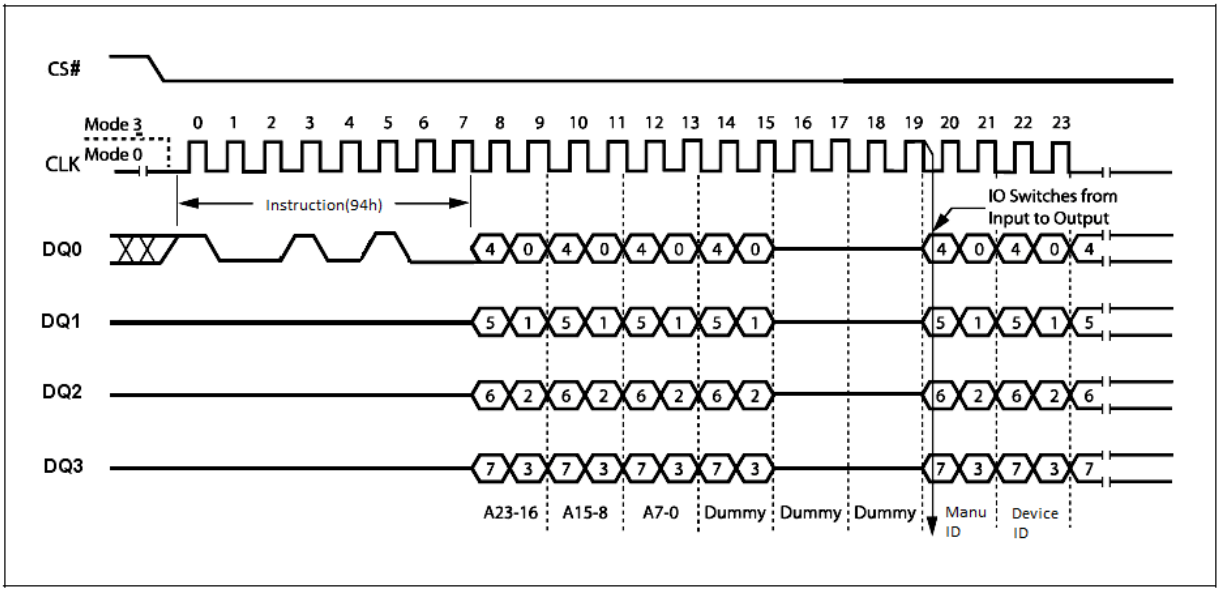
Figure 69. Read Manufacturer / Device ID by Dual I/O Diagram



Read Manufacturer / Device ID by Quad I/O (94h)

The Read Manufacturer/Device ID by Quad I/O instruction is very similar to the Quad IO Fast Read instruction. The instruction is initiated by driving the CS# pin low and shifting the instruction code “94h” followed by a 24-bit address (A23-A0) of 000000h and three byte of dummy. After which, the Manufacturer ID for (1Ch) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Read Manufacturer / Device ID by Quad I/O Diagram figure. The Device ID values for the device are listed in Manufacturer and Device Identification table. If the 24-bit address is initially set to 000001h, the Device ID will be read first.

Figure 70. Read Manufacturer / Device ID by Quad I/O Diagram



Read Identification (RDID) (9Fh)

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

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

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

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

The instruction sequence is shown in Read Identification (RDID) in QPI Mode figure while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Figure 71. Read Identification (RDID)

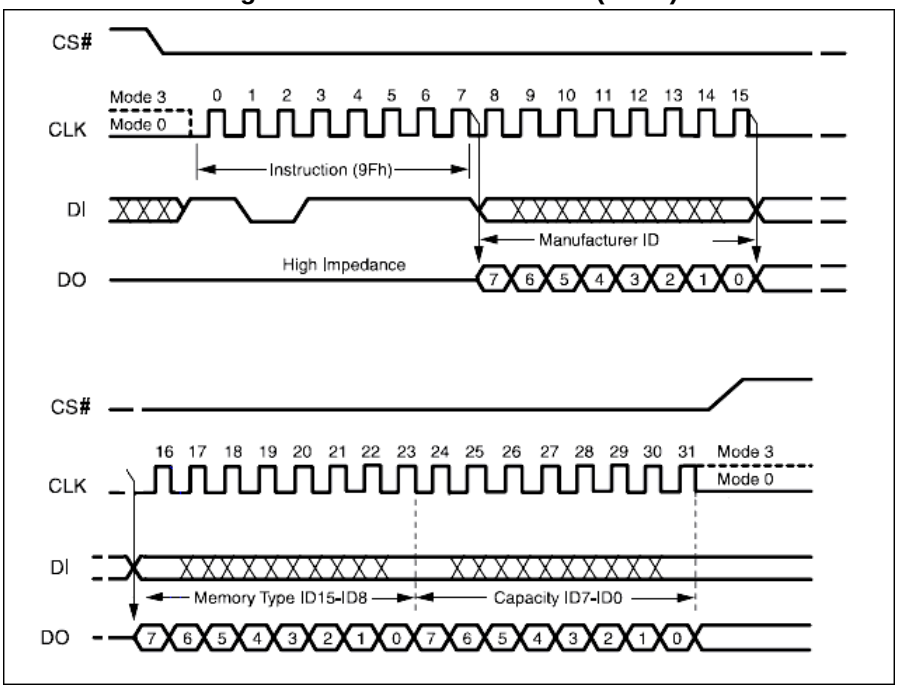
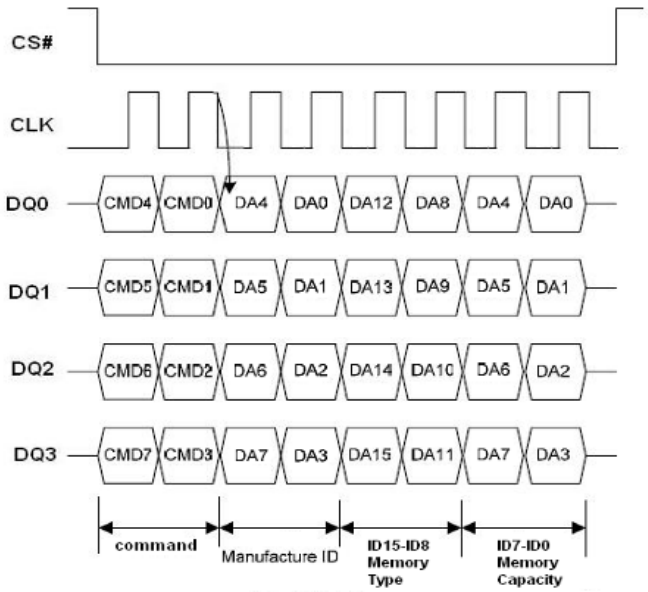


Figure 72. Read Identification (RDID) in QPI Mode



Program OTP array (42h)

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

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

This command also supports QPI mode.

Table 14. OTP Sector Address

Sector	Sector Size	Address Range
4095	512 byte	FFF000h – FFF1FFh
4094	512 byte	FFE000h – FFE1FFh
4093	512 byte	FFD000h – FFD1FFh

Note: The OTP sector is mapping to sector 4095, 4094 and 4093

Figure 73. Program OTP array

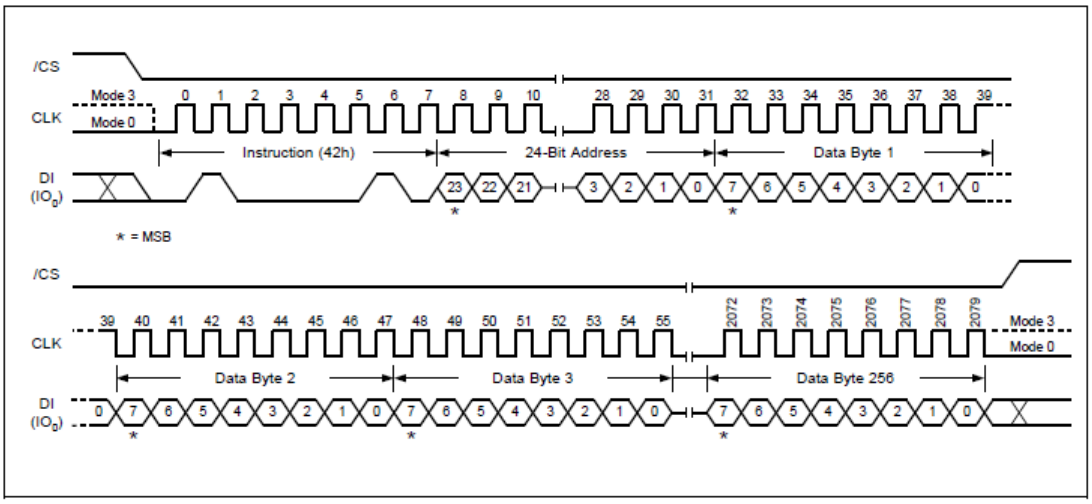
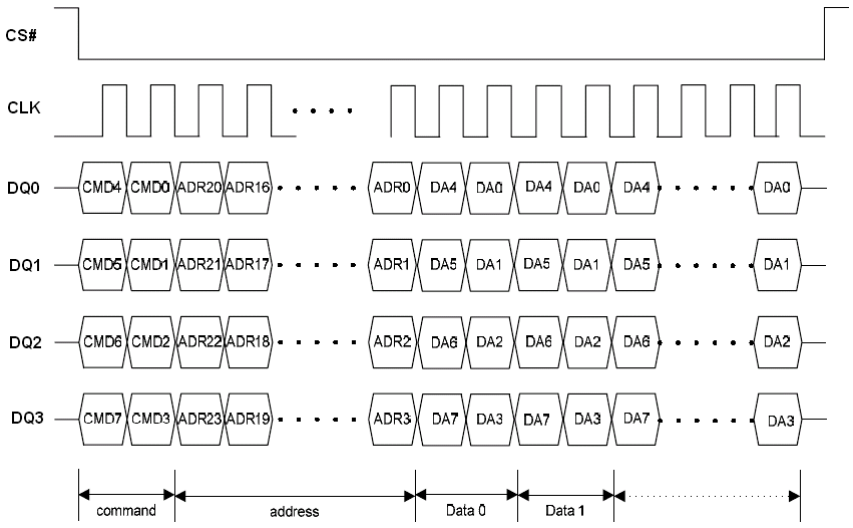


Figure 74. Program OTP array (QPI mode)



Read OTP array (48h)

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

This command also supports QPI mode.

Figure 75. Read OTP array

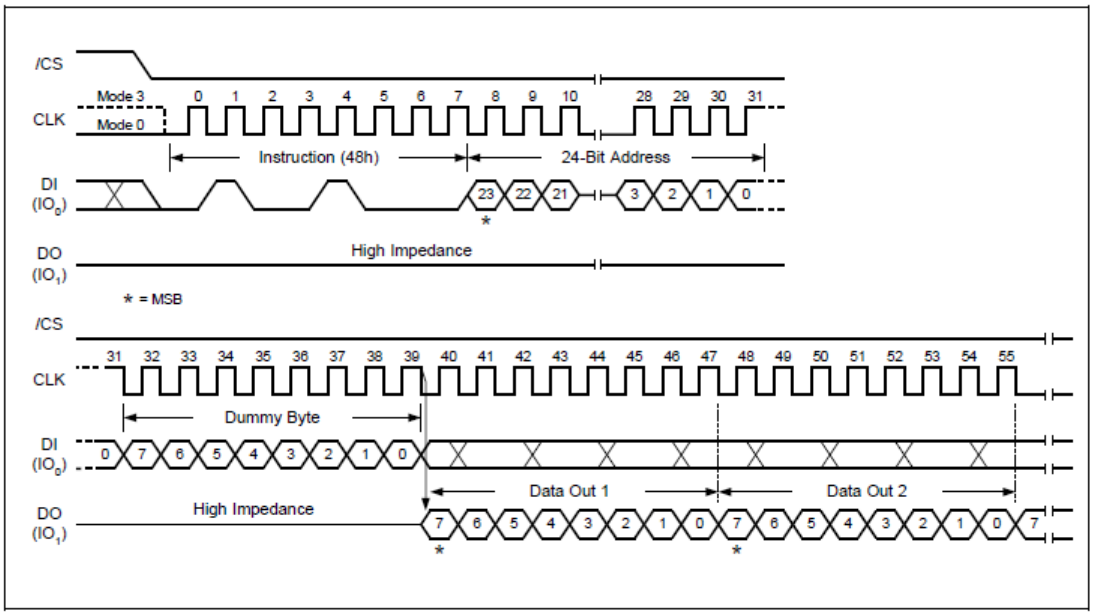
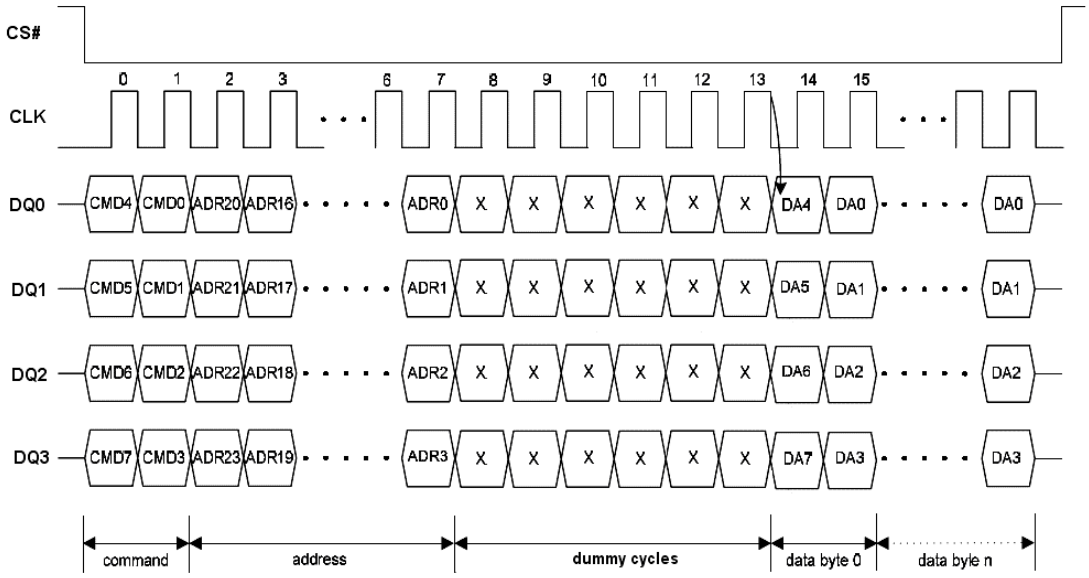


Figure 76. Read OTP array (QPI mode)



Erase OTP array (44h)

The device offers three set of 512-byte OTP array which can be erased and programmed individually. These registers may be used by the system manufacturers to store security and other important information separately from the main memory array.

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

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

This command supports QPI mode.

Figure 77. Erase OTP array

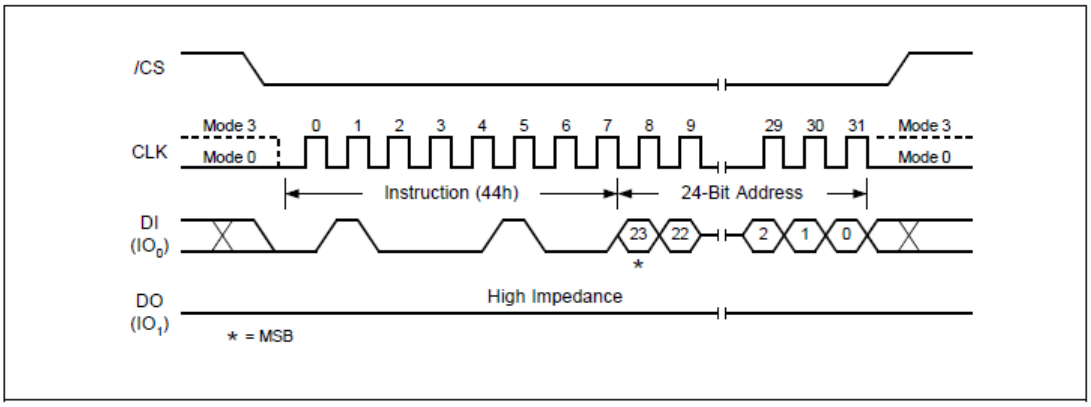
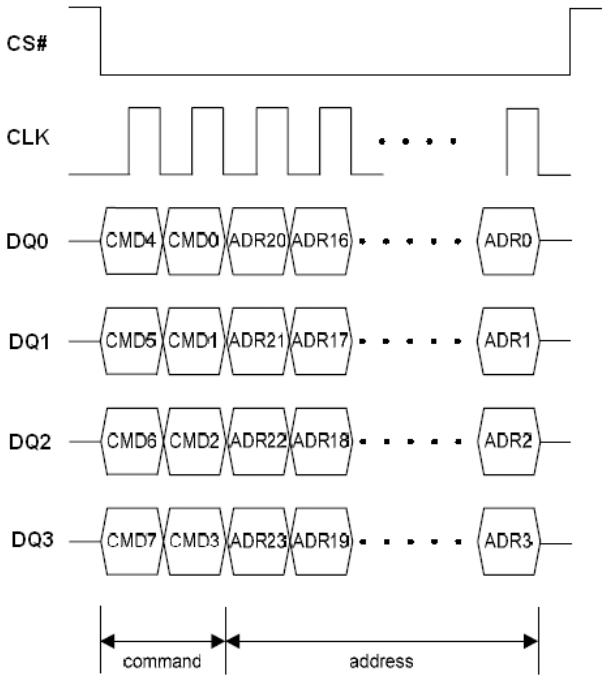


Figure 78. Erase OTP array (QPI mode)



Read SFDP Mode and Unique ID Number (5Ah)

Read SFDP mode

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

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

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

Figure 79. Read SFDP Mode and Unique ID Number Instruction Sequence Diagram

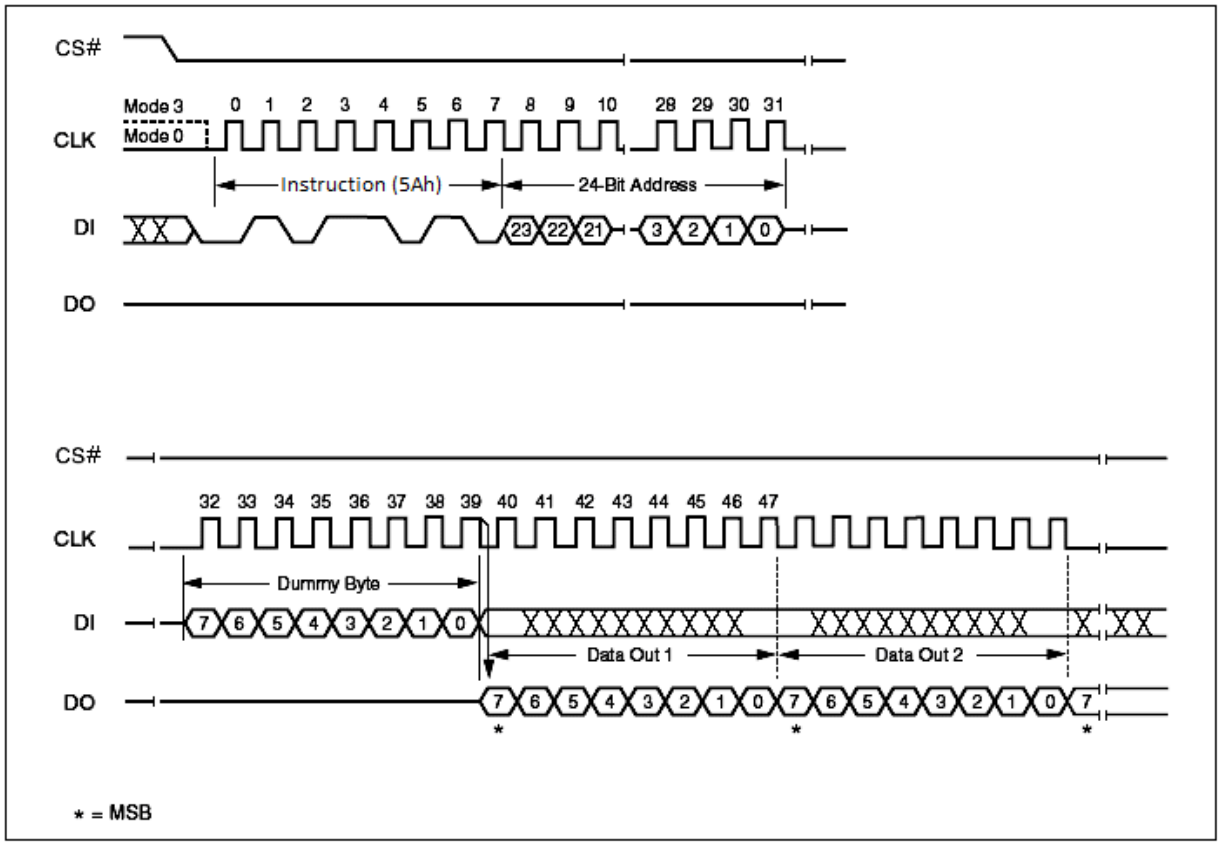


Table 15. Serial Flash Discoverable Parameters (SFDP) Signature and Parameter Identification Data Value (Advanced Information)

Description	Address (h) (Byte Mode)	Address (Bit)	Data (h)	Comment
SFDP Signature	00h	07 : 00	53h	Signature [31:0]: Hex: 50444653
	01h	15 : 08	46h	
	02h	23 : 16	44h	
	03h	31 : 24	50h	
SFDP Minor Revision Number	04h	07 : 00	06h	Star from 0x06
SFDP Major Revision Number	05h	15 : 08	01h	Star from 0x01
Number of Parameter Headers (NPH)	06h	23 : 16	03h	3 parameter header
Unused	07h	31 : 24	FFh	Reserved
ID Number	08h	07 : 00	00h	JEDEC ID
Parameter Table Minor Revision Number	09h	15 : 08	06h	Star from 0x06
Parameter Table Major Revision Number	0Ah	23 : 16	01h	Star from 0x01
Parameter Table Length (in DW)	0Bh	31 : 24	10h	16 DWORDs
Parameter Table Pointer (PTP)	0Ch	07 : 00	30h	000030h
	0Dh	15 : 08	00h	
	0Eh	23 : 16	00h	
Unused	0Fh	31 : 24	FFh	Reserved
ID Number (Vender ID)	10h	07:00	1Ch	
Parameter Table Minor revision Number	11h	15:08	00h	Start from 00h
Parameter Table Major Revision Number	12h	23:16	01h	Start from 01h
Parameter Table Length (in DW)	13h	31:24	04h	
Parameter Table Pointer (PTP)	14h	07:00	10h	
	15h	15:08	01h	
	16h	23:16	00h	
Unused	17h	31:24	FFh	
ID Number (4byte address)	18h	07:00	84h	
Parameter Table Minor revision Number	19h	15:08	00h	
Parameter Table Major Revision Number	1Ah	23:16	01h	
Parameter Table Length (in DW)	1Bh	31:24	02h	
Parameter Table Pointer (PTP)	1Ch	07:00	C0h	
	1Dh	15:08	00h	
	1Eh	23:16	00h	
Unused	1Fh	31:24	FFh	
ID Number (RPMC)	20h	07:00	03h	RPMC
Parameter Table Minor revision Number	21h	15:08	00h	
Parameter Table Major Revision Number	22h	23:16	01h	
Parameter Table Length (in DW)	23h	31:24	02h	
Parameter Table Pointer (PTP)	24h	07:00	F0h	
	25h	15:08	00h	
	26h	23:16	00h	
Unused	27h	31:24	FFh	

Table 16. Parameter ID (0) (Advanced Information) 1/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Block / Sector Erase sizes Identifies the erase granularity for all Flash Components	30h	00	01b	E5h	00 = reserved 01 = 4KB erase 10 = reserved 11 = 64KB erase
		01			
Write Granularity		02	1b		0 = No, 1 = Yes
Volatile Status Register Block Protect bits		03	0b		0: Block Protect bits in device's status register are solely non-volatile or may be programmed either as volatile using the 50h instruction for write enable or non-volatile using the 06h instruction for write enable.
Write Enable Opcode Select for Writing to Volatile Status Register		04	0b		0: 50h 1: 06h
Unused		07:05	111b		Reserved
4 Kilo-Byte Erase Opcode	31h	15:08	20h	20h	4 KB Erase Support (FFh = not supported)
Supports (1-1-2) Fast Read Device supports single input opcode & address and dual output data Fast Read	32h	16	1b	F9h	0 = not supported 1 = supported
Address Byte Number of bytes used in addressing for flash array read, write and erase.		17	00b		00 = 3-Byte 01 = 3- or 4-Byte (e.g. defaults to 3-Byte mode; enters 4-Byte mode on command) 10 = 4-Byte 11 = reserved
		18			
Supports Double Data Rate (DDR) Clocking Indicates the device supports some type of double transfer rate clocking.		19	1b		0 = not supported 1 = supported
Supports (1-2-2) Fast Read Device supports single input opcode, dual input address, and dual output data Fast Read		20	1b		0 = not supported 1 = supported
Supports (1-4-4) Fast Read Device supports single input opcode, quad input address, and quad output data Fast Read		21	1b		0 = not supported 1 = supported
Supports (1-1-4) Fast Read Device supports single input opcode & address and quad output data Fast Read		22	1b		0 = not supported 1 = supported
Unused		23	1b		Reserved
Unused	33h	31:24	FFh	FFh	Reserved

Table 16. Parameter ID (0) (Advanced Information) 2/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (h)	Comment
Flash Memory Density	37h : 34h	31 : 00	07FFFFFFh	128 Mbits

Table 16. Parameter ID (0) (Advanced Information) 3/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
(1-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output	38h	04:00	00100b	44h	4 dummy clocks
Quad Input Address Quad Output (1-4-4) Fast Read Number of Mode Bits		07:05	010b		8 mode bits
(1-4-4) Fast Read Opcode Opcode for single input opcode, quad input address, and quad output data Fast Read.	39h	15:08	EBh	EBh	
(1-1-4) Fast Read Number of Wait states (dummy clocks) needed before valid output	3Ah	20:16	01000b	08h	8 dummy clocks
(1-1-4) Fast Read Number of Mode Bits		23:21	000b		Not Supported
(1-1-4) Fast Read Opcode Opcode for single input opcode & address and quad output data Fast Read.	3Bh	31 : 24	6Bh	6Bh	

Table 16. Parameter ID (0) (Advanced Information) 4/16

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

Table 16. Parameter ID (0) (Advanced Information) 5/16

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

Table 16. Parameter ID (0) (Advanced Information) 6/16

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

Table 16. Parameter ID (0) (Advanced Information) 7/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Reserved. These bits default to all 1's	49h : 48h	15 : 00	FFh	FFh	Reserved
(4-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output	4Ah	20:16	00100b	44h	4 dummy clocks
(4-4-4) Fast Read Number of Mode Bits		23:21	010b		8 mode bits
(4-4-4) Fast Read Opcode Opcode for quad input opcode/address, quad output data Fast Read.	4Bh	31 : 24	EBh	EBh	Must Enter EQPI Mode Firstly

Table 16. Parameter ID (0) (Advanced Information) 8/16

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

Table 16. Parameter ID (0) (Advanced Information) 9/16

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

Table 16. Parameter ID (0) (Advanced Information) 10/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Multiplier form typical erase time to maximum erase time (max time = 2*(count+1)*erase typical time)	54h	03:00	0100b	24h	count
Erase type 1 Erase, typical time (typical time = (count + 1)*units)	55h	07:04	00010b	62h	count
		08			units : 00b: 1 ms, 01b: 16 ms, 10b: 128 ms, 11b: 1 s
		10:09	01b		
		15:11	01100b		count
Erase type 2 Erase, typical time (typical time = (count + 1)*units)	56h	17:16	01b	C9h	units : 00b: 1 ms, 01b: 16 ms, 10b: 128 ms, 11b: 1 s
Erase type 3 Erase, typical time (typical time = (count + 1)*units)		22:18	10010b		count
		23	01b		Units: 00b: 1 ms, 01b: 16 ms, 10b: 128 ms, 11b: 1 s
	57h	24		00h	
Erase type 4 Erase, typical time (typical time = (count + 1)*units)		29:25	00000b		count
		31:30	00b		Units : 00b: 1 ms, 01b: 16 ms, 10b: 128 ms, 11b: 1 s

Table 16. Parameter ID (0) (Advanced Information) 11/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Multiplier from typical time to max time for Page or byte program (maximum time = 2 * (count + 1)*typical time)	58h	03:00	0010b	82h	count
Page Size		07:04	1000b		Page
Page Program typical time (typical page program time = (count+1)*units)	59h	12:08	00111b	E7h	count
		13	1b		Units : 0:8us, 1:64us
Byte Program typical time, first byte (first byte typical time = (count+1)*units)		15:14	0111b		count
	5Ah	17:16		39h	
		18	0b		Units : 0:1us, 1:8us
Byte Program typical time, additional byte (additional byte time = (count+1)*units)		22:19	0111b		count
		23	0b		Units: 0:1us, 1:8us
Chip Erase, typical time	5Bh	28:24	01111b	CFh	count
		30:29	10b		Units : 00b:16ms, 01b:256ms, 10b:4s, 11b:64s
Reserved		31	1b		Reserved

Table 16. Parameter ID (0) (Advanced Information) 12/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Prohibited Operations During Program suspend	5Ch	03:00	0100b	44h	xxx0b: May not initiate a new erase anywhere (erase nesting not permitted) xx0xb: May not initiate a new page program anywhere (program nesting not permitted) x1xxb: May not initiate a read in the program suspended page size 1xxxb: The erase and program restrictions in bits 1:0 are sufficient
Prohibited Operations During Erase suspend		07:04	0100b		xxx0b: May not initiate a new erase anywhere (erase nesting not permitted) xx0xb: May not initiate a page program anywhere x0xxb: Refer to vendor datasheet for read restrictions 0xxxb: Additional erase or program restrictions apply
Reserved	5Dh	08	1b	87h	reserved
Program Resume to Suspend interval		12:09	0011b		Count of fixed units of 64us
Suspend in-progress program max latency (max latency=(count+1)*untis		15:13	11100b		count
	5Eh	17:16		37h	
		19:18	01b		Units : 00b:128ns, 01b:1us, 10b;8us, 11b:64us
Erase resume to Suspend interval (latency=(count+1)*64us)		23:20	0011b		Count of fixed units of 64us
Suspend in-progress erase max latency	5Fh	28:24	11100b	3Ch	count
		30:29	01b		Units : 00b: 128ns, 01b:1us, 10b:8us, 11b:64us
Suspend/Resume supported		31	0b		0:supported 1:not supported

Table 16. Parameter ID (0) (Advanced Information) 13/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (h)	Comment
Program Resume Instruction	60h	07:00	30h	
Program Suspend Instruction	61h	15:08	B0h	
Resume Instruction	62h	23:16	30h	
Suspend Instruction	63h	31:24	B0h	

Table 16. Parameter ID (0) (Advanced Information) 14/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Reserved	64h	01:00	11b	F7h	Reserved
Status Register Polling Device Busy		07:02	111101b		Bit 2: Read WIP bit [0] by 05h Read instruction Bit 3: Read bit 7 of Status Register by 70h Read instruction (0=not supported 1=support) Bit 07:04, Reserved: 1111b
Exit Deep Power down to next operation delay (delay=(count+1)*units)	65h	12:08	00010b	A2h	count
		14:13	01b		Units : 00b:128ns, 01b:1us, 10b:8us, 11b:64us
Exit Deep Power down Instruction		15	10101011b (ABh)		
	66h	22:16		D5h	
Enter Deep Power down Instruction		23	10111001b (B9h)		
	67h	30:24		5Ch	
Deep Power down Supported		31	0b		0:suppored 1:not supported

Table 16. Parameter ID (0) (Advanced Information) 15/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
4-4-4 mode disable sequences	68h	03:00	1001b	29h	xxx1b: issue FFh instruction 1xxxb: issue the Soft Reset 66/99 sequence
4-4-4 mode enable sequences		07:04	00010b		x_xx1xb: issue instruction 38h
		08			
0-4-4 mode supported		09	1b		0: not supported 1:supported
0-4-4 mode Exit Method	69h	15:10	100101b	96h	xx_xxx1b: Mode Bits[7:0] = 00h will terminate this mode at the end of the current read operation. xx_xx1xb: If 3-Byte address active, input Fh on DQ0-DQ3 for 8 clocks. If 4-Byte address active, input Fh on DQ0-DQ3 for 10 clocks. xx_x1xxb: Reserved xx_1xxxb: Input Fh (mode bit reset) on DQ0-DQ3 for 8 clocks. x1_xxxxb: Mode Bit[7:0]≠Axh 1x_xxxxb: Reserved
0-4-4 Mode entry Method	6Ah	19:16	1001b	49h	xxx1b: Mode Bits[7:0] = A5h Note: QE must be set prior to using this mode x1xxb: Mode Bit[7:0]=Axh 1xxxb: Reserved
Quad Enable Requirements		22:20	100b		000b: No QE bit. Detects 1-1-4/1-4-4 reads based on instruction 010b: QE is bit 6 of Status Register. where 1=Quad Enable or 0=not Quad Enable 111b: Not Supported
HOLD or RESET Disable by bit 4 of Ext Register		23	0b		0:not supported
Reserved	6Bh	31:24	FFh	FFh	Reserved

Table 16. Parameter ID (0) (Advanced Information) 16/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Volatile or Non-Volatile Register and Write Enable Instruction for Status Register 1	6Ch	06:00	1101000b	E8h	xxx_1xxxb: Non-Volatile/Volatile status register 1 powers-up to last written value in the nonvolatile status register, use instruction 06h to enable write to non-volatile status register. Volatile status register may be activated after power-up to override the non-volatile status register, use instruction 50h to enable write and activate the volatile status register. x1x_xxxxb: Reserved 1xx_xxxxb: Reserved NOTE If the status register is read-only then this field will contain all zeros in bits 4:0.
Reserved		07	1b		reserved
Soft Reset and Rescue Sequence Support	6Dh	13:08	010000b	10h	x1_xxxxb: issue reset enable instruction 66h, then issue reset instruction 99h. The reset enable, reset sequence may be issued on 1, 2, or 4 wires depending on the device operating mode.
		09			
		10			
		11			
		12			
		13			
Exit 4-byte Address		15:14	00000b		x1_xxxx_xxxxb: Reserved 1x_xxxx_xxxxb: Reserved
	6Eh	18:16		C0h	
		23:19	11000b		
Enter 4-Byte Address	6Fh	31:24	10000000b	80h	1xxx_xxxxb: Reserved

Table 17. Parameter ID (1) (Advanced Information-4byte address instruction)

Description (4byte address instruction)	Address (h) (Byte Mode)	Address (Bit)	Data (b/h)	Data (h)	Comment
Support for 1-1-1 Read Command, Instruction=13h	C0h	00	0b	00h	0:not supported 1:supported
Support for 1-1-1 Fast Read Command, Instruction=0Ch		01	0b		0:not supported 1:supported
Support for 1-1-2 Fast Read Command, Instruction=3Ch		02	0b		0:not supported 1:supported
Support for 1-2-2 Fast Read Command, Instruction=BCh		03	0b		0:not supported 1:supported
Support for 1-1-4 Fast Read Command, Instruction=ECh		04	0b		0:not supported 1:supported
Support for 1-4-4 Fast Read Command, Instruction=ECh		05	0b		0:not supported 1:supported
Support for 1-1-1 Page Program Command, Instruction=12h		06	0b		0:not supported 1:supported
Support for 1-1-4 Page Program Command, Instruction=34h		07	0b		0:not supported 1:supported
Support for 1-4-4 Page Program Command, Instruction=3Eh	C1h	08	0b	00h	0:not supported 1:supported
Support for Erase Command-Type 1 size, instruction loop in next Dword		09	0b		0:not supported 1:supported
Support for Erase Command-Type 2 size, instruction lookup in next Dword		10	0b		0:not supported 1:supported
Support for Erase Command-Type 3 size, instruction lookup in next Dword		11	0b		0:not supported 1:supported
Support for Erase Command-Type 4 size, instruction lookup in next Dword		12	0b		0:not supported 1:supported
Support for 1-1-1 DTR Read Command, Instruction=0Eh		13	0b		0:not supported 1:supported
Support for 1-2-2 DTR Read Command, Instruction=BEh		14	0b		0:not supported 1:supported
Support for 1-4-4 DTR Read Command, Instruction=EEh		15	0b		0:not supported 1:supported
Support for volatile individual sector lock Read command, Instruction=E0h	C2h	16	0b	F0h	0=not supported
Support for volatile individual sector lock Write command, Instruction=E1h		17	0b		0=not supported
Support for non-volatile individual sector lock read command, Instruction=E2h		18	0b		0=not supported
Support for non-volatile indivdual sector lock write command, Instrucion=E3h		19	0b		0=not supported
Reserved		23:20	1111b		
Reserved	C3h	31:24	FFh	FFh	
Instruction for Erase Type 1	C4h	07:00	FFh	FFh	
Instruction for Erase Type 2	C5h	15:08	FFh	FFh	
Instruction for Erase Type 3	C6h	23:16	FFh	FFh	
Instruction for Erase Type 4	C7h	31:24	FFh	FFh	

Table 18. RPMC Parameter ID

Description	Address (h) (Byte Mode)	Address (Bit)	Data (Binary)	Data (Word)	Comment
Flash Hardening	F0h	00	0b	38h	Flash Hardening is supported
MC_Size		01	0b		Monotonic counter size is 32bit
Busy_Polling_Method		02	0b		Poll for OP1 busy using OP2 RPMC status
Reserved		03	1b		Must be 1
Number_Counter-1		07:04	0011b		Number of support counter-1 Suggest value=3
OP1	F1h	15 : 08	9Bh	9Bh	OP1
OP2	F2h	23 : 16	96h	96h	OP2
Update_Rate	F3h	27 : 24	0000b	F0h	Update Rate=5*2**update_Rate (s)
Reserved		31 : 28	1111b		Must be 1
Read Counter Polling Delay (Typ) to calculate HMAC two times	F4h	04 : 00	**0 1010b**	**AAh**	Counter
		06:05	**01b**		Unit 01: 16us
		07	**1b**		Reserved
Write Counter Polling Short Delay	F5h	12 : 08	**1 0100b**	**B4h**	Polling_Short_delay_write_counter
		14 :13	**01b**		Unit 01 16us
		15	**1b**		Reserved
Write Counter Polling Long Delay	F6h	20 : 16	**1 1001b**	**B9h**	Polling_long_delay_write_counter
		22 : 21	**01b**		Unit 01: 16ms
		23	**1b**		Reserved
Reserved	F7h	31 : 24	FFh	FFh	Must be 1

Table 19. Parameter ID (2) (Advanced Information-ESMT flash parameter)

Description (ESMT Flash Parameter Tables)	Address (h) (Byte Mode)	Address (Bit)	Data (h/b)	Data (h)	Comment
V_{CC} Supply Max Voltage	111h:110h	07:00 15:08	00h 20h	00h 20h	2000h=1.95V
V_{CC} Supply Min Voltage	113h:112h	23:16 31:24	00h 16h	00h 16h	1600h=1.65V
HW RESET# pin	115h:114h	00	1b	9Fh	0:not support 1:supported
HW HOLD# pin		01	1b		0:not support 1:supported
Deep Power down Supported		02	1b		0:not support 1:supported
SW Reset		03	1b		0:not support 1:supported
SW Reset Instruction		07:04 11:08	99h		
Program Suspend/Resume		12	1b	F9h	0:not support 1:supported
Erase Suspend/Resume		13	1b		0:not support 1:supported
Unused		14	1b		
Wrap Read Mode		15	1b		0:not support 1:supported
Wrap Read Instruction	116h	23:16	0Ch	0Ch	
Wrap Read data length	117h	31:24	64h	64h	64h:8B&16B &32B&64B
Individual block lock	11Bh:118h	00	0b	FCh	0:not support 1:supported
Individual block lock bit		01	0b		0:volatile 1:nonvolatile
Individual block lock Instruction		07:02 09:08	FFh		
Individual block lock Volatile protect bit default protect status		10	0b	CBh	0:protecct 1:unprotect
Secured OTP		11	1b		0:not support 1:supported
Read Lock		12	0b		0:not support 1:supported
Permanent Lock		13	0b		0:not support 1:supported
Unused		15:14	11b		
Unused		31:16	FFh	FFh	
Unused	11F:11Ch		FFh	FFh	

Read Unique ID Number

The Read Unique ID Number instruction accesses a factory-set read-only 96-bit number that is unique to each device. The ID number can be used in conjunction with user software methods to help prevent copying or cloning of a system. The Read Unique ID instruction is initiated by driving the CS# pin low and shifting the instruction code “5Ah” followed by a three bytes of addresses, 0x1E0h, and one byte of dummy clocks. After which, the 96-bit ID is shifted out on the falling edge of CLK.

Table 20. Unique ID Number

Description	Address (h) (Byte Mode)	Address (Bit)	Data (h)	Comment
Unique ID Number	1E0h : 1EBh	95 : 00	By die	

Write Root Key Register (9Bh + 00h)

This command is used by the SPI Flash Controller to initialize the Root Key Register corresponding to the received Counter Address with the received Root Key. It is expected to be used in an OEM manufacturing environment when the SPI Flash Controller and SPI Flash are powered together for the first time.

After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:

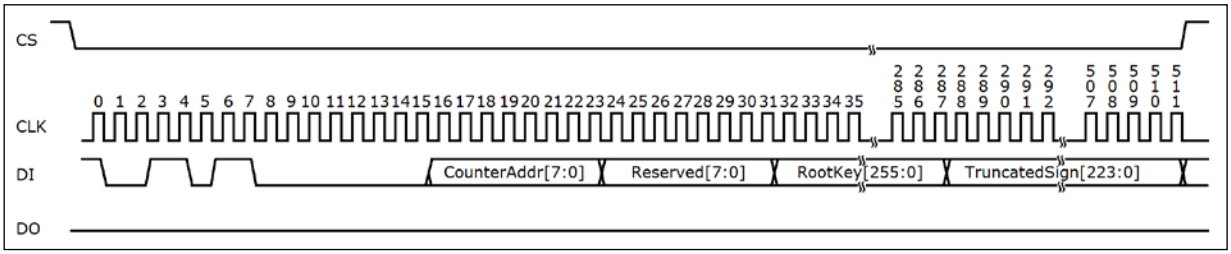
- Payload size is correct. (include OP1 is 64 bytes)
- Counter Address falls within the range of supported counters.
- The Root Key Register corresponding to the requested Counter Address was previously ninitialized. [Root_Key_Reg_Init_State[Counter_Address] = 0xFFh]
- Truncated signature field is the same as least significant 224 bits of HMAC-SHA-256 based signature computed based on received input parameters.

If the received transaction is error free SPI Flash device successfully executes the command and posts “successful completion” in the RPMC Status Register. This command must be executed to ensure that power cycling in the middle of command execution is properly handled. This requires that the internal state tracking the root key register initialization is written as the last operation of the command execution. (Root_Key_Reg_Init_State[Counter_Address] = 0)

Root Key Register Write with root key is = 256’hFF…FF is used as a temporary key. When this request is received error-free Root_Key_Reg_Init_State[Counter_Address] is not affected. Instead only the corresponding Monotonic Counter is initialized to 0 if previously uninitialized. This state is tracked as separate state using MC_Init_State[Counter_Address]. This state is used to leave the monotonic counters at the current value when an error free Root Key Register Write operation is received. (Both 256’hFF..FF and non 256’hFF..FF)

Once this command is successfully executed with a non 256’hFF..FF Root Key, the device will not accept the “Write Root Key Register” command any more, and the Root Key value cannot be read out by any instructions.

Figure 80. Write Root Key Instruction Sequence Diagram



Update HMAC Key (9Bh + 01h)

This command is used by the SPI Flash Controller to update the HMAC-Key register corresponding to the received Counter Address with a new HMAC key calculated based on received input. This command must be issued once only on every power cycle event on the interface. This allows the HMAC key storage to be implemented using volatile memory. Status register busy indication is expected to indicate busy for double the amount of Read_Counter_Polling_Delay specified in SFDP table since this command performs two distinct HMAC-SHA-256 computations.

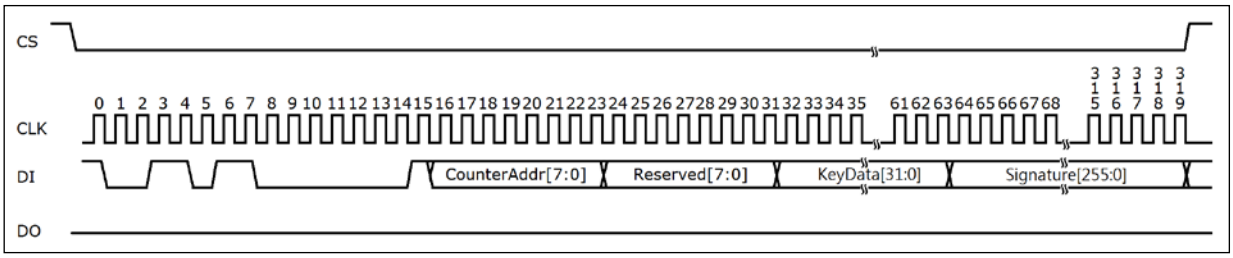
After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:

- Payload size is correct. (including OP1 = 40 bytes)
- Counter Address falls within the range of supported counters.
- The Monotonic Counter corresponding to the requested Counter Address was previously initialized.
- Signature matches the HMAC-SHA-256 based signature computed based on received input parameters. This command performs two HMAC-SHA-256 operations.

If the received transaction is error free, the SPI Flash device successfully executes the command and posts “successful completion” in the RPMC Status Register.

If the received transaction has errors, the SPI Flash device does not execute the transaction and posts the corresponding error in the RPMC Status Register.

Figure 81. Update HMAC Key Instruction Sequence Diagram



Increment Monotonic Counter (9Bh + 02h)

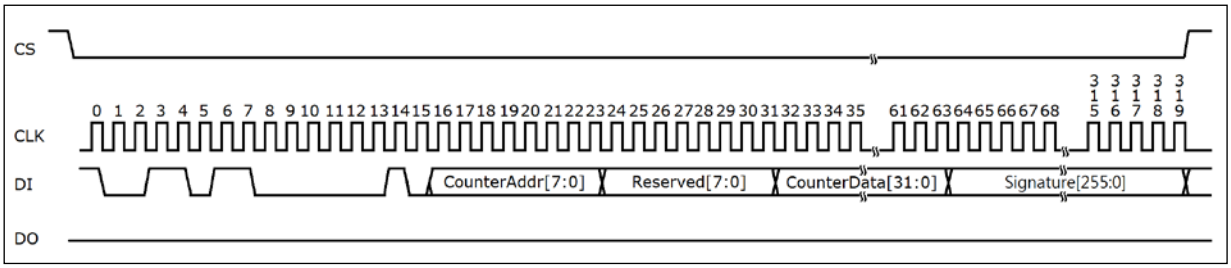
This command is used by the SPI Flash Controller to increment the Monotonic counter by 1 inside the SPI Flash Device.

After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:

- Payload size is correct. (including OP1 = 40 bytes)
- Counter Address falls within the range of supported counters.
- The Monotonic Counter corresponding to the requested Counter Address was previously initialized.
- The HMAC Key Register corresponding to the requested Counter Address was previously initialized.
- The requested Signature matches the HMAC-SHA-256 based signature computed based on received input parameters.
- The received Counter Data matches the current value of the counter read from the SPI Flash.

If the received transaction is error free, the SPI Flash device successfully executes the command and posts “successful completion” in the RPMC Status Register. The increment counter implementation should make sure that the counter increment operation is performed in a power glitch aware manner.

Figure 82. Increment Monotonic Counter Instruction Sequence Diagram



Request Monotonic Counter (9Bh + 03h)

This command is used by the SPI Flash Controller to request the Monotonic counter value inside the SPI Flash Device.

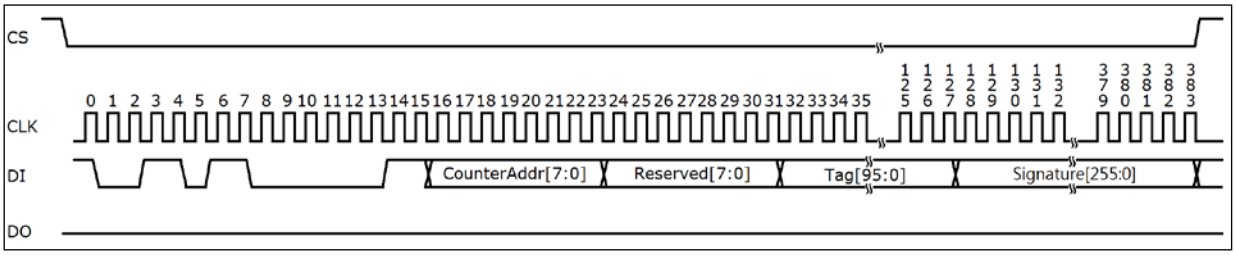
After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:

- Payload size is correct. (including OP1 = 48 bytes)
- Counter Address falls within the range of supported counters.
- The Monotonic Counter corresponding to the requested Counter Address was previously initialized.
- The HMAC Key Register corresponding to the requested Counter Address was previously initialized.
- The requested Signature matches the HMAC-SHA-256 based signature computed based on received input parameters.

If the received transaction is error free, the SPI Flash device successfully executes the command and posts “successful completion” in the RPMC Status Register. In response to this command, the SPI flash reads the monotonic counter addressed by counter address. It calculates HMAC-SHA-256 signatures the second time, based on following parameters.

- HMAC Message[127:0] = Tag [95:0], Counter_Data_Read[31:0]
- HMAC Key[255:0] = HMAC_Key_Register[Counter_Address][255:0]

Figure 83. Request Monotonic Counter Instruction Sequence Diagram



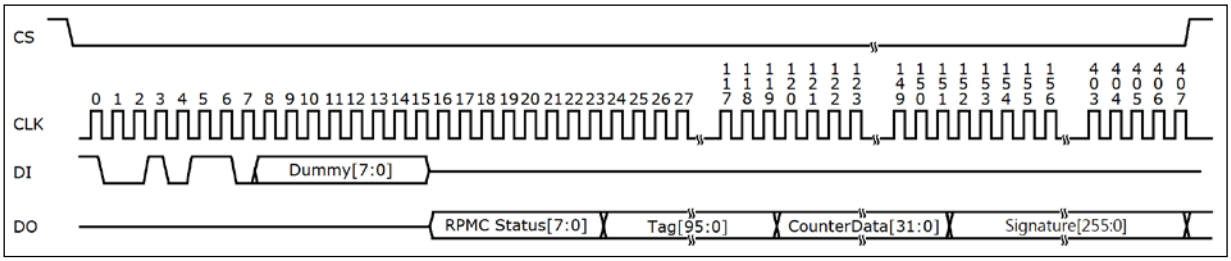
Reserved RPMC Commands (9Bh + 04h~FFh)

If the SPI Flash Controller issues any of the reserved command-types, the SPI Flash Device must return Error status in the RPMC Status Register. It asserts bit 2 to indicate that a reserved command-type was issued.

Read RPMC Status / Data (96h)

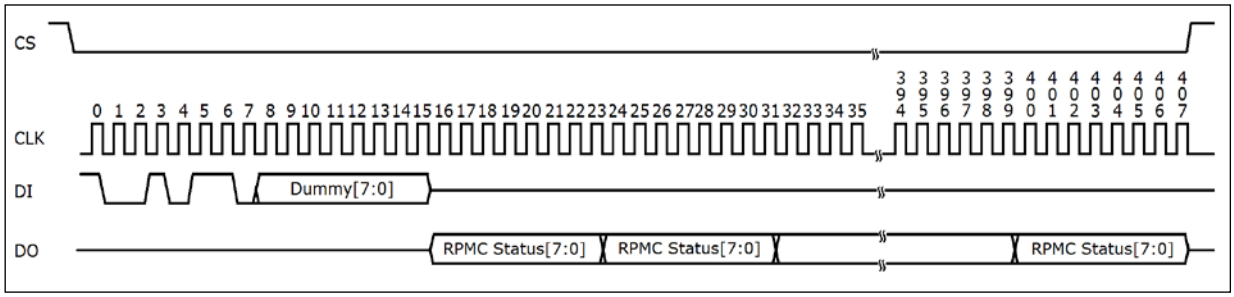
This command is used by the SPI Flash Controller to read the RPMC status from any previously issued OP1 command. In addition, if previous OP1 command is Request Monotonic Counter and if SPI Flash returns successful completion (BUSY=0) in the RPMC Status Register, then it must also return valid values in the Tag, Counter Data and Signature field. If there're other error flags, the values returned in Tag, Counter and Signature field are invalid. The controller may abort the read prematurely prior to completely reading the entire payload. This may occur when the controller wants to simply read the RPMC status or when it observes an error being returned in the RPMC status field. The controller may also continue reading past the defined payload size of 49 bytes. Since this is an error condition, the SPI Flash may return any data past the defined payload size. The controller must ignore the data.

Figure 84. Read RPMC Status / Data Instruction Sequence Diagram (BUSY=0)



When BUSY=1, from Byte-3 and beyond, the device will output the RPMC_Status[7:0] value continuously until CS# terminates the instruction. The device will not output Tag, CounterData & Signature fields when BUSY=1. Once BUSY becomes 0, another OP2 command must be issued to read out the correct Tag, CounterData & Signature fields.

Figure 85. Read RPMC Status Instruction Sequence Diagram (BUSY=1)



Power-up Timing

All functionalities and DC specifications are specified for a V_{CC} ramp rate of greater than 1V per 100 ms. See Power-Up Timing table and Power-Up Timing table figure for more information.

Figure 86. Power-up Timing

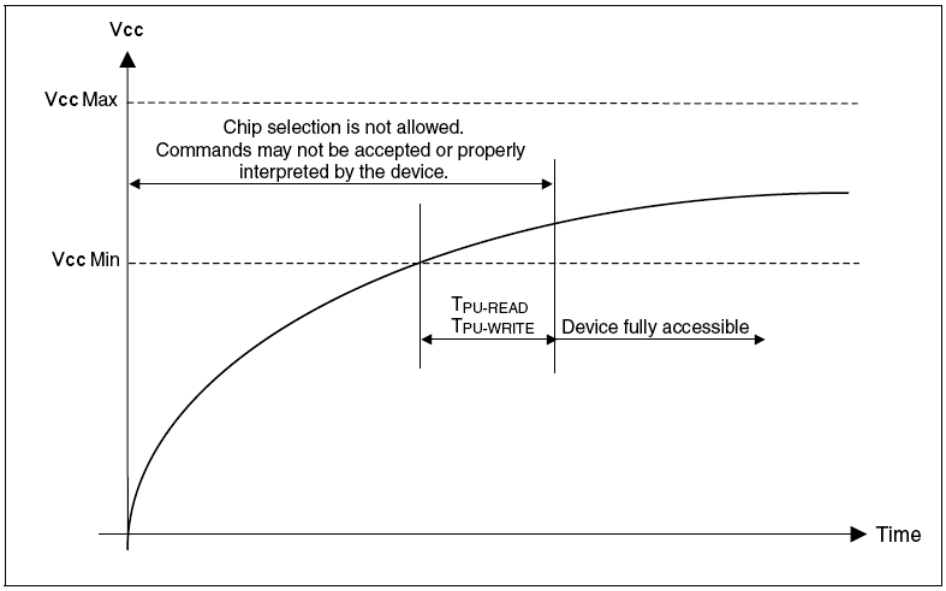


Table 21. Power-Up Timing

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

Note:

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

INITIAL DELIVERY STATE

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

DC Characteristics and Operating Conditions

Table 22. DC Characteristics

(T_A = -40°C to 85°C; V_{CC} = 1.65-1.95V)

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}	-	0.1	10	µA
I_{CC2}	Deep Power-down Current	CS# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	-	0.1	10	µA
I_{CC3}	Operating Current (READ)	CLK = 0.1 V_{CC} / 0.9 V_{CC} at 104 MHz, DQ = open	-	8	13	mA
		CLK = 0.1 V_{CC} / 0.9 V_{CC} at 104 MHz for Quad Output Read, DQ = open	-	11	16	mA
I_{CC4}	Operating Current (PP)	CS# = V_{CC}	-	15	30	mA
I_{CC5}	Operating Current (WRSR/WRSR3)	CS# = V_{CC}	-	1	15	mA
I_{CC6} *1	Operating Current (SE)	CS# = V_{CC}	-	5	10	mA
I_{CC7} *1	Operating Current (HBE/BE)	CS# = V_{CC}	-	5	10	mA
I_{CC9}	RPMC OP1 Write Only	CS# = V_{CC}	-	15	20	mA
I_{CC10}	RPMC OP1 & Array Read	CLK = 0.1 V_{CC} / 0.9 V_{CC}, DQ = open	-	19	40	mA
I_{CC11}	RPMC OP1 & Array Program / Erase	CS# = V_{CC}	-	35	45	mA
I_{CC12}	RPMC OP2 Read Only	CLK = 0.1 V_{CC} / 0.9 V_{CC}	-	-	30	mA
I_{CC13}	RPMC OP2 & Array Program / Erase	CLK = 0.1 V_{CC} / 0.9 V_{CC}	-	30	45	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} = 100 µA, V_{CC} = V_{CC} Min	-	-	0.3	V
V_{OH}	Output High Voltage	I_{OH} = -100 µA, V_{CC} = V_{CC} Min	V_{CC} - 0.2	-	-	V

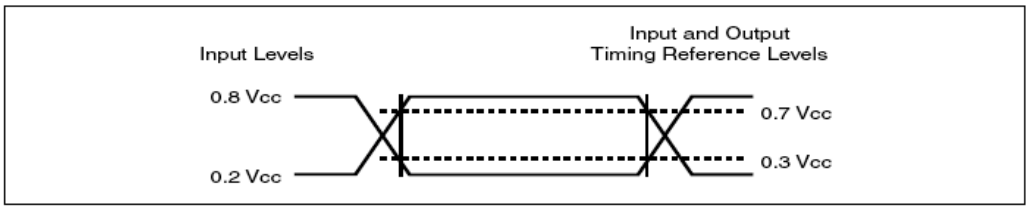
Note:
1. Erase current measure on all cells = '0' state.

AC Timing Input / Output Conditions

Table 23. AC Measurement Conditions

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

Figure 87. AC Measurement I/O Waveform



AC Timing Input / Output Conditions

Table 24. AC Characteristics

(T_A = -40°C to 85°C; V_{CC} = 1.65-1.95V)

Symbol	Alt	Parameter	Min	Typ	Max	Unit
F_R	f_C	Serial SDR SPI Clock Frequency for: PP, QPP, SE, HBE, BE, CE, DP, RES, RDP, WREN, WRDI, WRSR, WRSR2, WRSR3, Fast Read, RDSR, RDSR2, RDSR3, RDID	D.C.	-	104	MHz
		Serial SDR Dual/Quad Clock Frequency for: PP, QPP, SE, HBE, BE, CE, DP, RES, RDP, WREN, WRDI, WRSR, WRSR3, RDSR, RDSR2, RDSR3, RDID, Fast Read, Dual Output Fast Read, Dual I/O Fast Read, Quad Output Fast Read, Quad I/O Fast Read	D.C.	-	104	MHz
		Serial DDR SPI Clock Frequency for: DDR Fast Read, DDR Read Burst with Wrap, DDR Mode Page Program	D.C.	-	52	MHz
		Serial DDR Dual/Quad Clock Frequency for: DDR Fast Read, DDR Dual I/O Fast Read, DDR Quad I/O Fast Read, DDR Read Burst with Wrap, DDR Mode Page Program	D.C.	-	52	MHz
f_R		Serial Clock Frequency for READ	D.C.	-	50	MHz
F_R	f_c	Serial Clock Frequency for RPMC	D.C.		80	MHz
t_{CH} *1		Serial Clock High Time	3.5	-	-	ns
t_{CL} *1		Serial Clock Low Time	3.5	-	-	ns
t_{CH} *1		Serial Clock High Time (RPMC)	5	-	-	ns
t_{CL} *1		Serial Clock Low Time (RPMC)	5	-	-	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	5	-	-	ns
t_{CHSH}		CS# Active Hold Time	5	-	-	ns
t_{SHCH}		CS# Not Active Setup Time	5	-	-	ns
t_{CHSL}		CS# Not Active Hold Time	5	-	-	ns
t_{SHSL}	t_{CSH}	CS# High Time CS# High Time (RPMC)	30 50	-	-	ns
t_{SHSL} *2	t_{CSH}	Volatile Register Write Time	50	-	-	ns
t_{SHQZ} *2	t_{DIS}	Output Disable Time Output Disable Time (RPMC)	-	-	6 10	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 Data In Hold Time (RPMC)	3 3	-	-	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

Table 24. AC Characteristics (Continued)

Symbol	Alt	Parameter		Min	Typ	Max	Unit
t_{HLQZ} *2	t_{HZ}	HOLD# to Output High-Z		-	-	6	ns
t_{HHQX} *2	t_{LZ}	HOLD# to Output Low-Z		-	-	6	ns
t_{CLQV}	t_V	Output Valid from CLK for 30pF		-	-	8	ns
		Output Valid from CLK for 15pF		-	-	6	ns
		Output Valid from CLK for RPMC		-	-	7.5	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		-	10	50	ms
t_{PP}		Page Programming Time		-	0.5	3	ms
t_{SE}		Sector Erase Time		-	0.04	0.3	s
t_{HBE}		Half Block Erase Time		-	0.2	1	s
t_{BE}		Block Erase Time		-	0.3	2	s
t_{CE}		Chip Erase Time		-	60	200	s
t_{KEY}		RPMC Write Root Key Register		-	280	500	µs
t_{HMAC}		RPMC Update HMAC Key Register		-	140	400	µs
t_{INC1}		RPMC Increment Monotonic Counter		-	120	300	µs
t_{INC2}		RPMC Increment Monotonic Counter (Counter Switching)		-	120	**400**	ms
t_{REQ}		RPMC Request Monotonic Counter		-	150	400	µs
t_{HRST}		RESET# low period to reset the device		1	-	-	µs
t_{HRSL}		RESET# high to next instruction		28	-	-	µs
t_{SHRV}		Deselect to RESET# valid in quad mode		8	-	-	ns
	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 88. Serial Output Timing

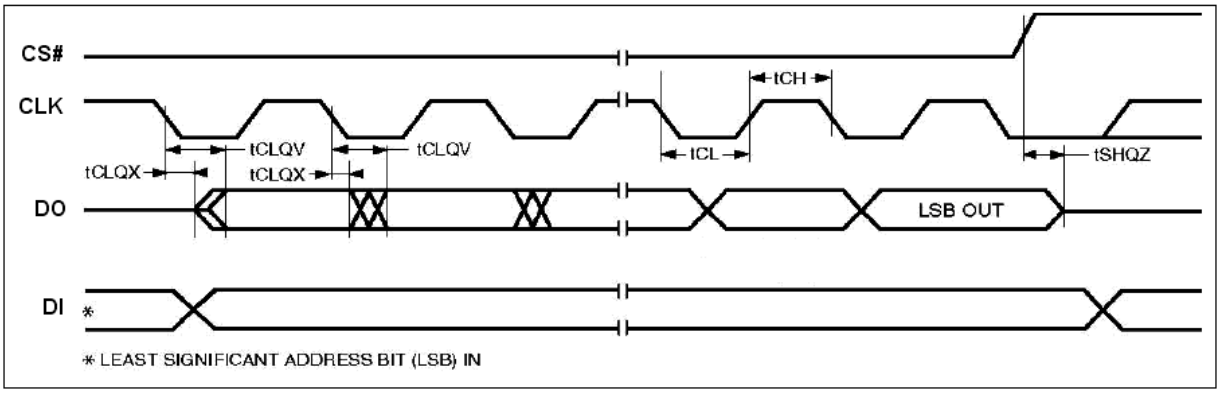


Figure 89. Input Timing

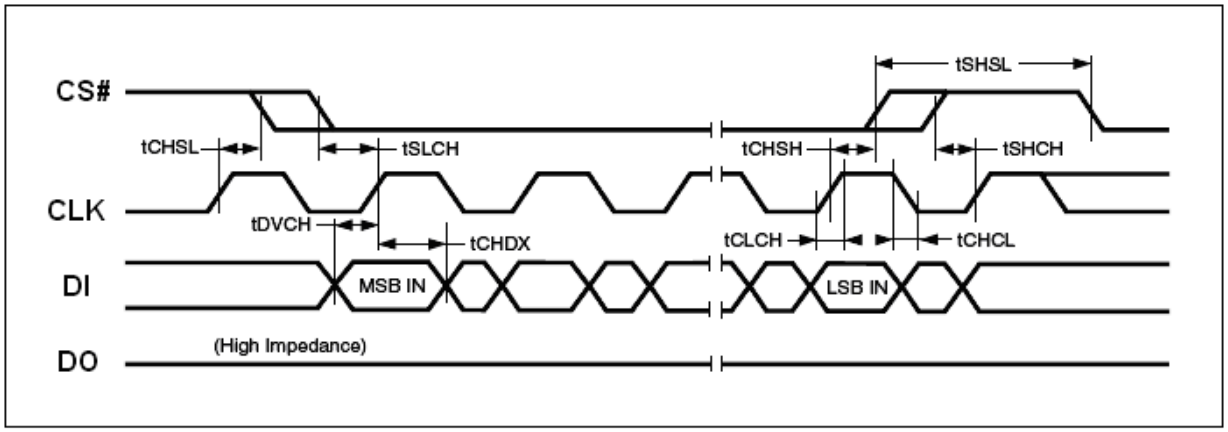


Figure 90. Hold Timing

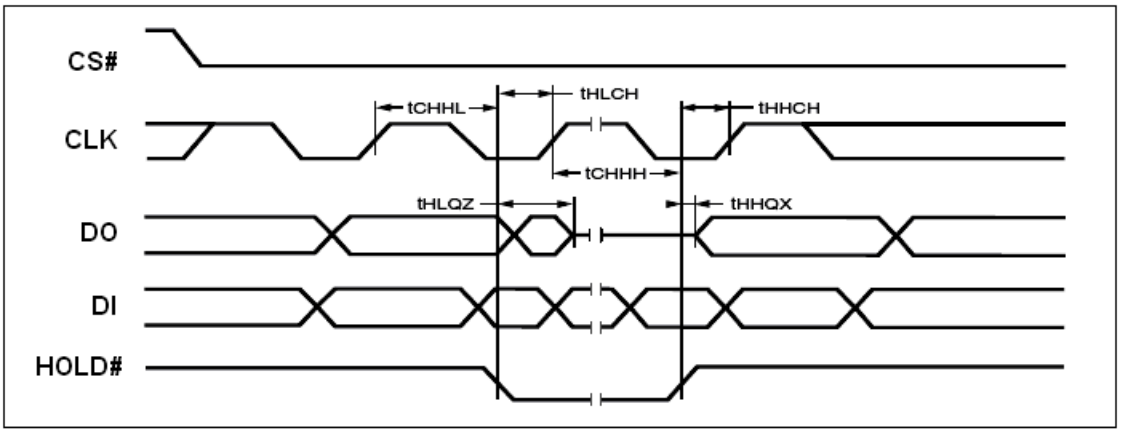


Figure 91. Reset Timing

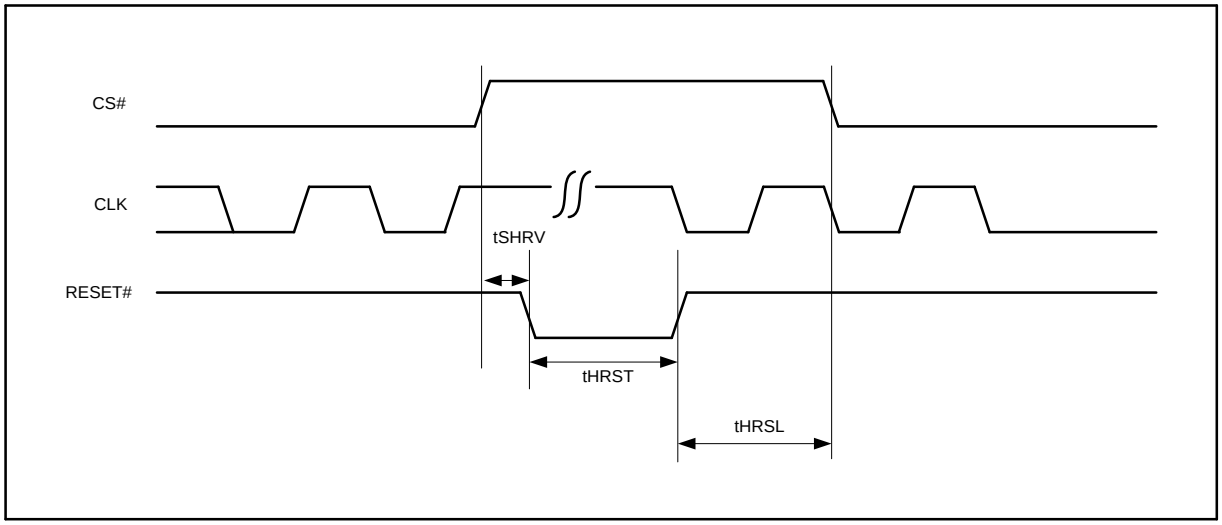


Figure 92. Serial Output Timing for Double Data Rate Mode

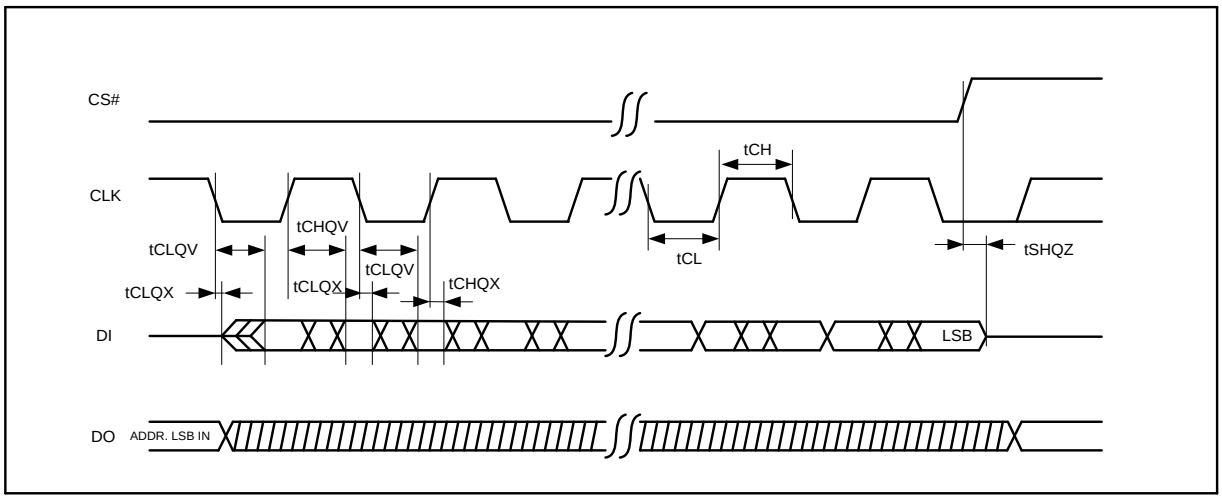
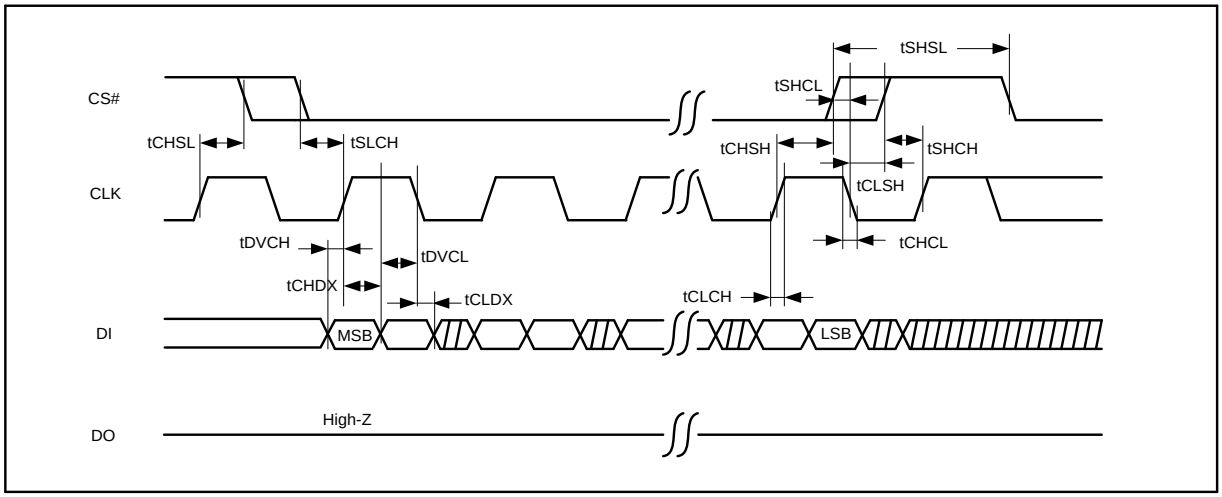


Figure 93. Serial Input Timing for Double Data Rate Mode



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.

Table 25. Absolute Ratings

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

Note:

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

RECOMMENDED OPERATING RANGES [*1]

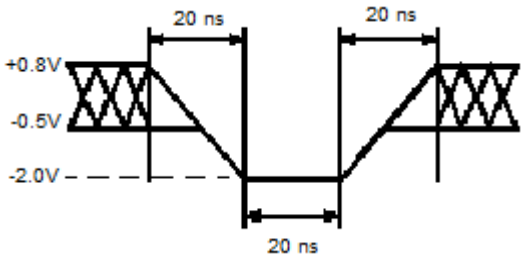
Table 26. Operating Conditions

Parameter	Value	Unit
Ambient Operating Temperature Industrial Devices	-40 to 85	°C
Operating Supply Voltage V_{CC}	Full: 1.65 to 1.95	V

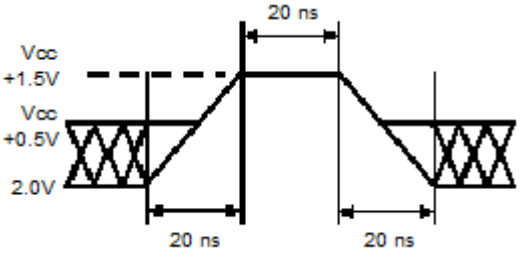
Note:

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

Figure 94. Overshoot Waveform



Maximum Negative Overshoot Waveform



Maximum Positive Overshoot Waveform

Input / Output Capacitance

Table 27. CAPACITANCE

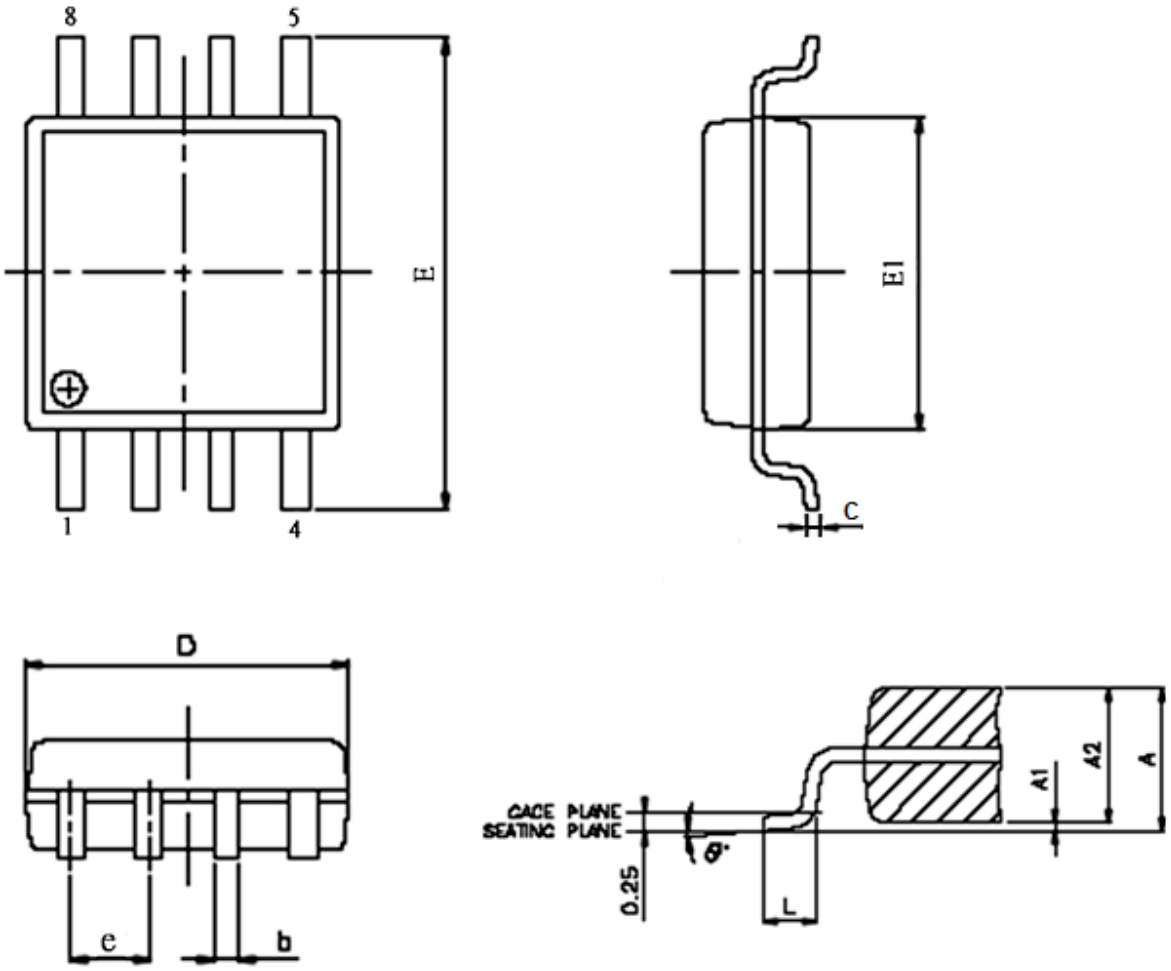
(V_{CC} = 1.65-1.95V)

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

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

PACKAGE MECHANICAL

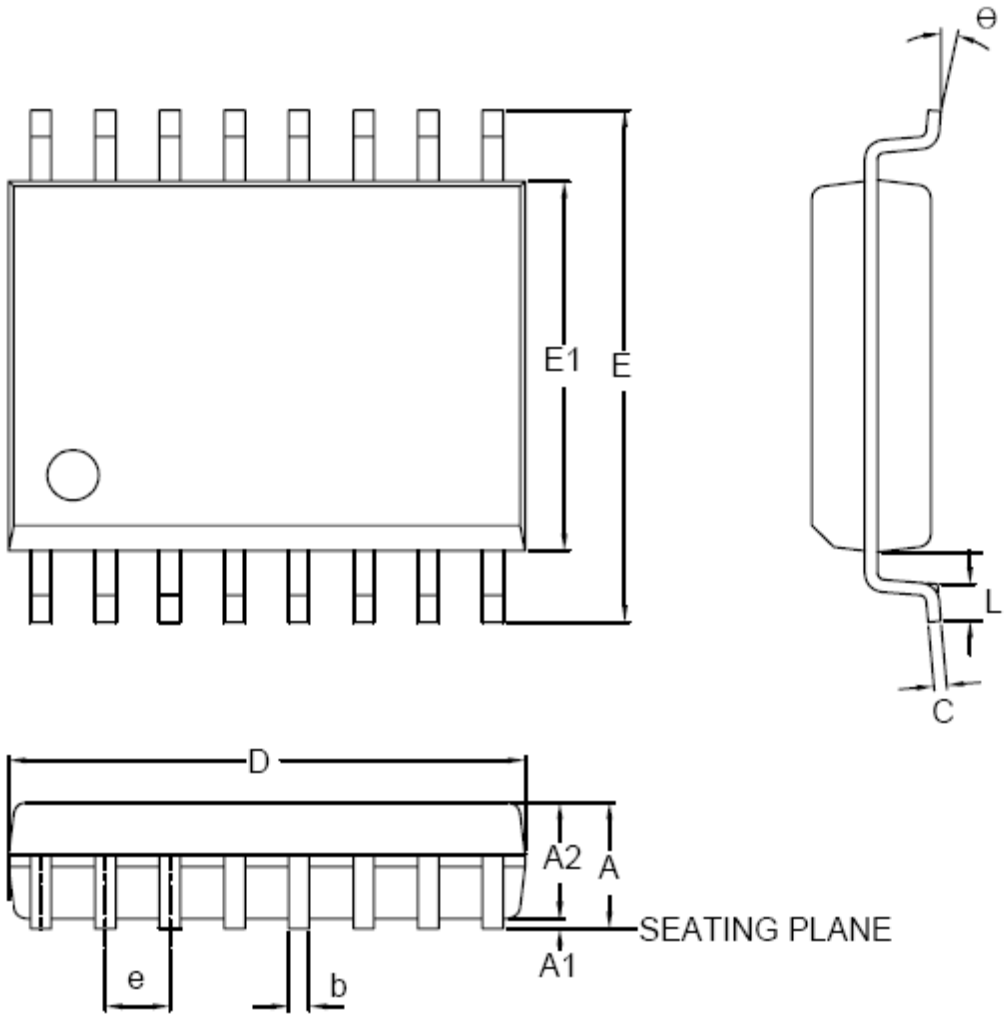
Figure 95. SOP 200 mil (official name = 208 mil)



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.

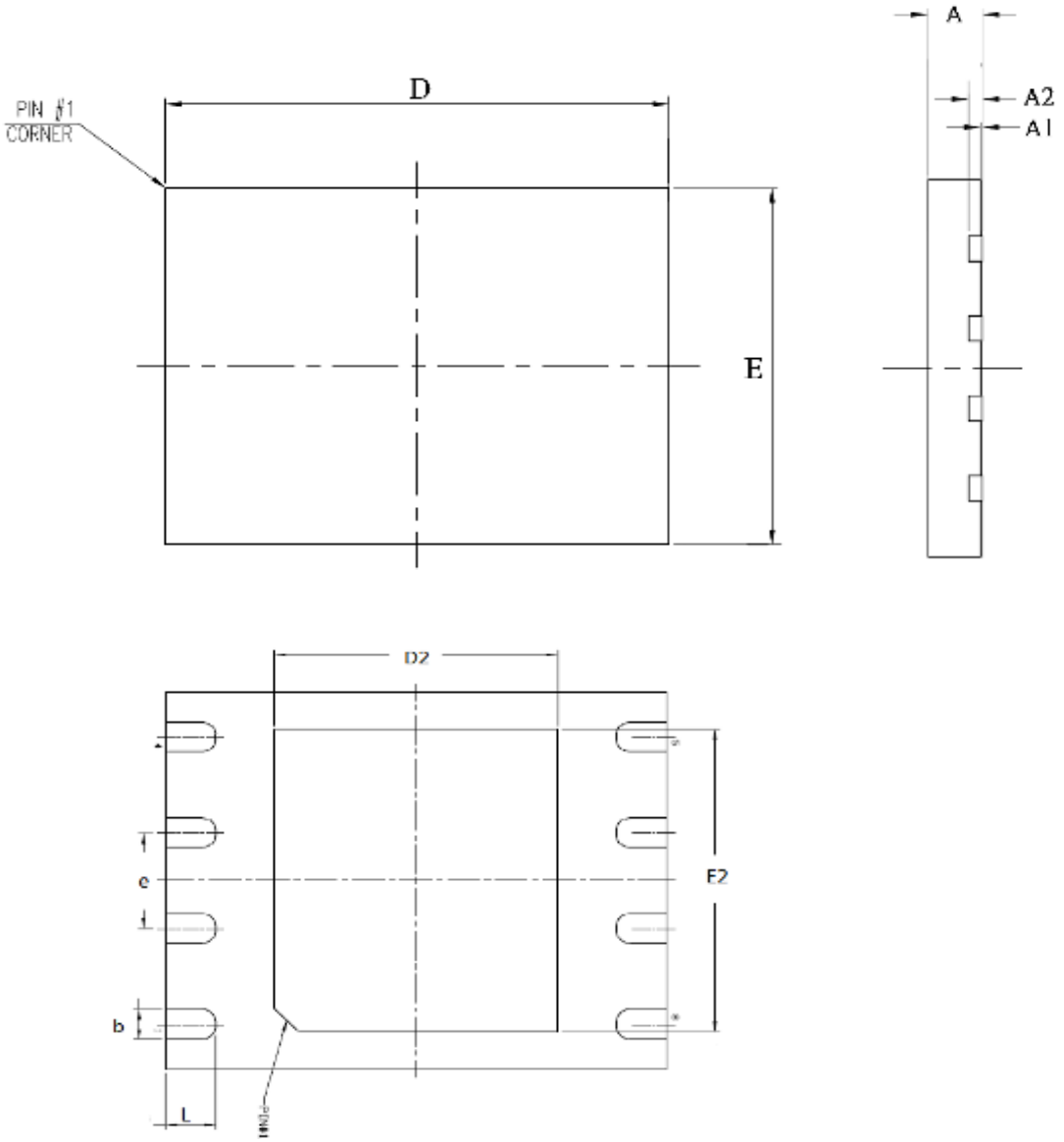
Figure 96. 16 LEAD SOP 300 mil



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	0^0	5^0	8^0

Note : 1. Coplanarity: 0.1 mm

Figure 97. 8-LEAD VDFN / WSON (6x5 mm)

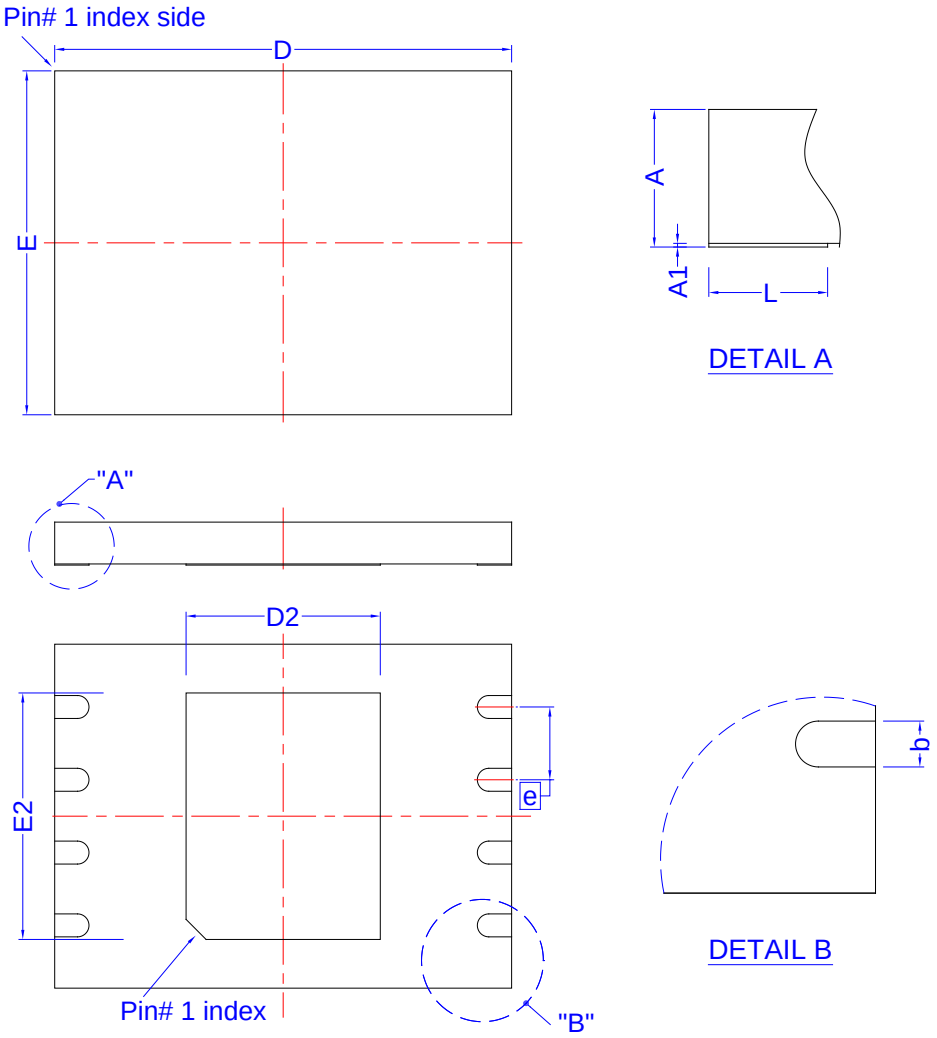


Controlling dimensions are in millimeters (mm).

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 98. 8-LEAD VDFN / WSON (8x6 mm)



Symbol	Dimension in mm			Dimension in inch		
	Min	Norm	Max	Min	Norm	Max
A	0.70	0.75	0.80	0.028	0.030	0.031
A1	0.00	0.02	0.05	0.000	0.001	0.002
b	0.35	0.40	0.48	0.014	0.016	0.019
D		8.00 BSC			0.315 BSC	
D2	3.30	3.40	3.50	0.130	0.134	0.138
E		6.00 BSC			0.236 BSC	
E2	4.20	4.30	4.40	0.165	0.169	0.173
e		1.27 BSC			0.050 BSC	
L	0.40	0.50	0.60	0.016	0.020	0.024

Controlling dimension: millimeter
(Revision date: Jul 14 2022)

ORDERING INFORMATION

EN35SXR128A - 104 H I P 2PC

DIFFERENTIATION CODE
2PC = Quad Enable

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

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

PACKAGE
H = 8-pin 200 mil SOP
F = 16-pin 300 mil SOP
W = 8-contact VDFN / WSON (6x5 mm)
Y = 8-contact VDFN / WSON (8x6 mm)

SPEED
104 = 104 MHz

BASE PART NUMBER
EN = Eon Silicon Solution Inc.
35SX = 1.8V Serial Flash with 4KB Uniform-Sector
R = RPMC
128 = 128 Megabit (16,384K x 8)
A = version identifier

Revisions List

Revision No	Description	Date
1.0	Initial Release	2023/04/07

Important Notice

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

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

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

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