EN35SXR256A (2PC)
256 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
- 256 M-bit Serial Flash
  - 256 M-bit / 32,768 KByte / 131,072 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO, WP#, HOLD#
  - Dual SPI: CLK, CS#, DQ_0, DQ_1, WP#, HOLD#
  - Quad SPI: CLK, CS#, DQ_0, DQ_1, DQ_2, DQ_3
  - Default QE=1 (Quad Enable), WP#, HOLD# disable
- High performance
  - Full voltage range
    - 104 MHz clock rate for Single/Dual/Quad I/O Fast Read
  - Regulated voltage range: 1.8-1.95 volt
    - 133 MHz clock rate for Quad I/O Fast Read
- Low power consumption
  - 12 mA typical active current
  - 1 µA typical power down current
- Uniform Sector Architecture
  - 8,192 sectors of 4-Kbyte
  - 1,024 blocks of 32-Kbyte
  - 512 blocks of 64-Kbyte
  - Any sector or block can be erased individually
- Software and Hardware Write Protection
  - Write Protect all or portion of memory via software
  - Enable/Disable protection with WP# pin
- High performance program/erase speed
  - Page program time: 0.5 ms typical
  - Sector erase time: 40 ms typical
  - Half Block erase time 200 ms typical
  - Block erase time 300 ms typical
  - Chip erase time: 120 seconds typical
- 3byte address and 4byte address switch
- Volatile Status Register Bits
- Lockable 3x512 byte OTP security sector
- Support Serial Flash Discoverable Parameters (SFDP) signature
- Read Unique ID Number
- Replay-Protected Monotonic Counter (RPMC)
- Minimum 100K endurance cycle
- Data retention time 20 years
- 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 256 Megabit (32,768K-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.

By providing the ability to protect and unprotect blocks, a system can unprotect blocks to modify their content 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.

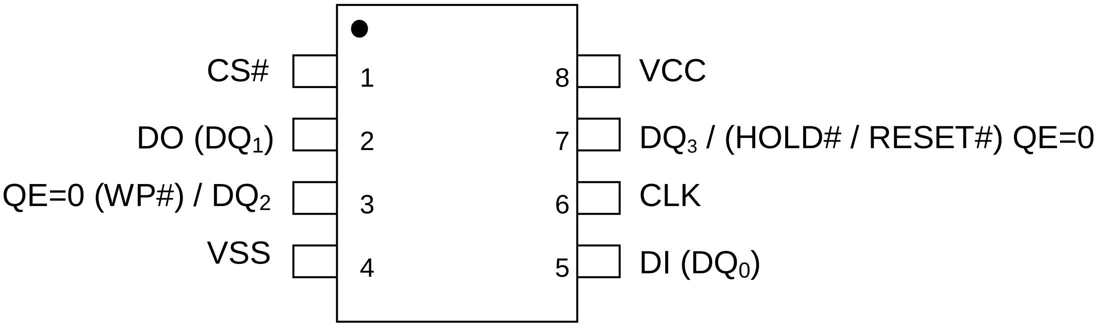
The device is designed to allow either single Sector/Block at a time or full chip erase operation. The device can be configured to protect part of the memory as the software protected mode. The device scans sustain a minimum of 100K program/erase cycles on each sector or block.

Contents

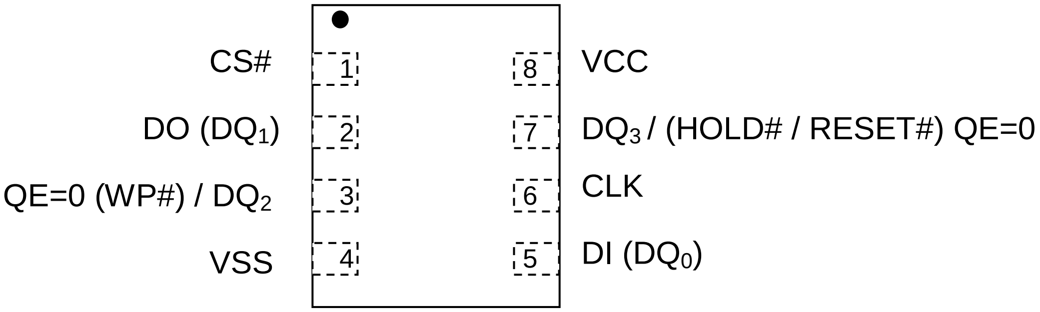
List of Figures

List of Tables

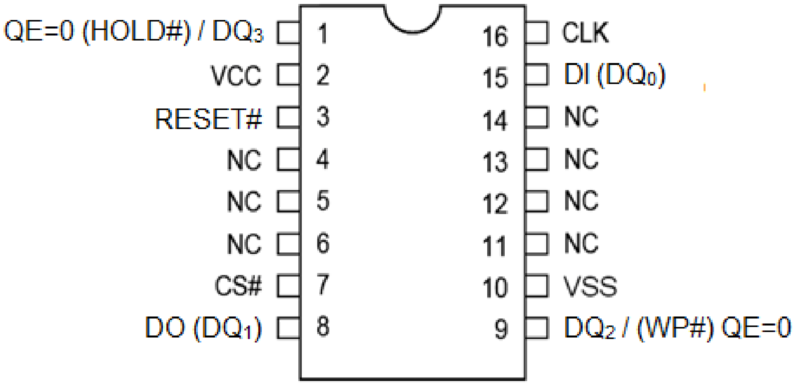
CONNECTION DIAGRAMS (TOP VIEW)



8 – LEAD SOP



8 – LEAD VDFN / WSON



16 – LEAD SOP

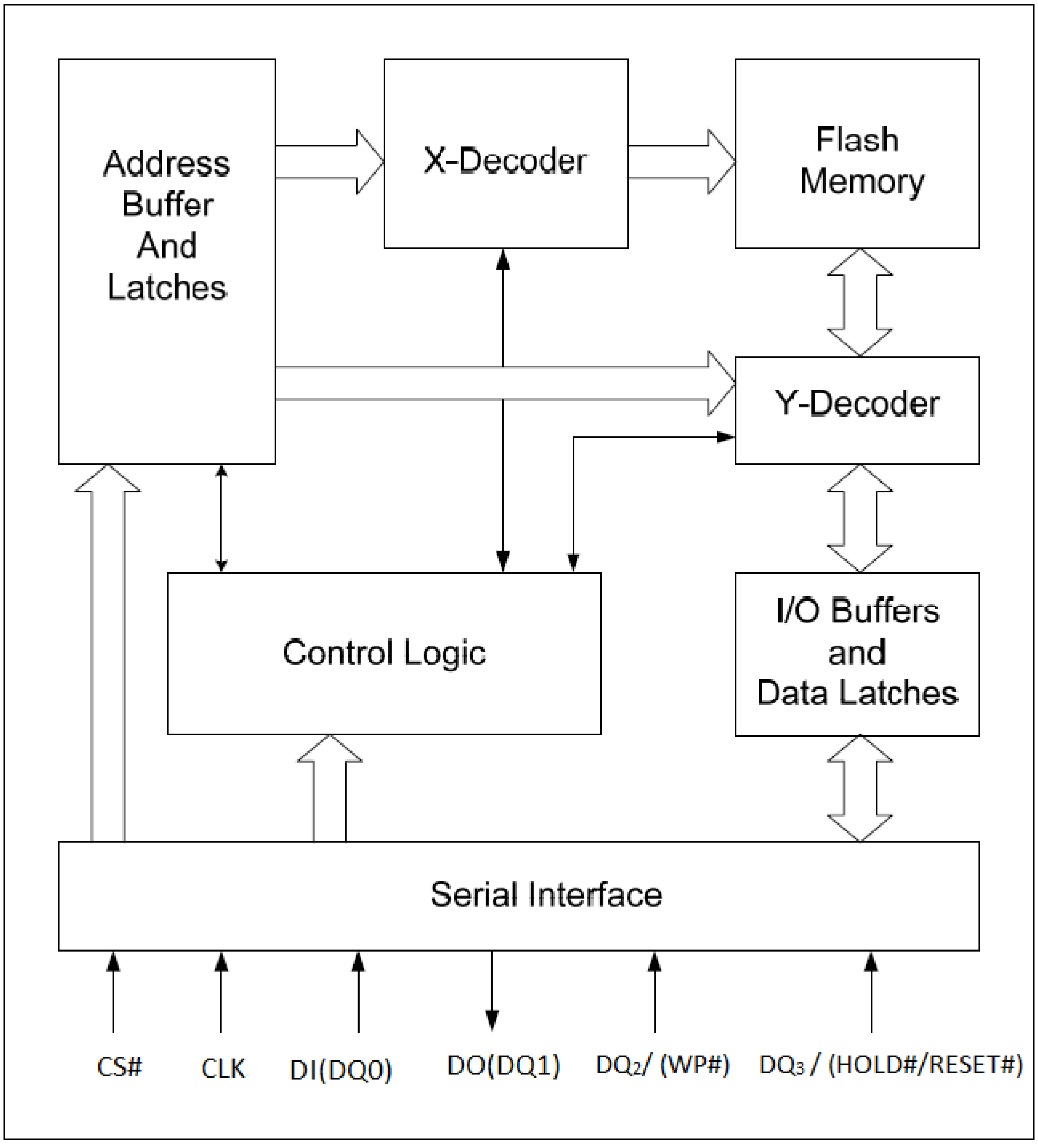
Table 1. Pin Names

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) [*2]
DQ_3 / (HOLD# / RESET#)	Data Input Output 3 / (HOLD# / RESET# pin) [*2]
DQ_3 / (HOLD#)	Data Input Output 3 / (HOLD# pin) [*2] (only SOP16)
RESET#	RESET# pin (only SOP16)
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# or HOLD# functions can be available by setting QE=0 for Standard/Dual SPI mode.
3. RESET# only work at SOP16 package

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.
3. RESET# only work at SOP16 package

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 device s 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, TB, BP3, 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 (DQ_2) 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=0 and HRSW=0(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 while it is actively selected. When QE=0 and HRSW=0(default), the RESET# pin is disabled. 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 or Quad Output operation. For SOP16 package, the RESET# pin is a dedicated hardware reset pin regardless of device settings or operation states. If the hardware reset function is not used, this pin can be left floating or connected to V_{CC} in the system.

Set RESET# to low for a minimum period 1us (t_{HRST}) will interrupt any on-going instructions to have the device to initial state. The device can accept new instructions again in 28us (t_{HRSL}) after RESET# back to high.

Table 2. Pin definition setting with QE and HRSW bits at 8 and 16 pin package

	HRSW=0	HRSW=1
8pin		
QE =0	HOLD/DQ3 WP/DQ2	RESET/DQ3 WP/DQ2
QE =1	DQ3 DQ2	DQ3 DQ2
16 pin		
QE =0	PIN1 HOLD/DQ3 PIN3 RESET PIN9 WP/DQ2	PIN1 RESET/DQ3 PIN3 RESET PIN9 WP/DQ2
QE =1	PIN1 DQ3 PIN3 RESET PIN9 DQ2	PIN1 DQ3 PIN3 RESET PIN9 DQ2

MEMORY ORGANIZATION

The memory is organized as:

- 33,554,432 bytes
- Uniform Sector Architecture
  512 blocks of 64-Kbyte
  1,024 sectors of 32-Kbyte
  8,192 sectors of 4-Kbyte
  131,072 pages (256 bytes each)

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

Table 3. Uniform Block Sector Architecture

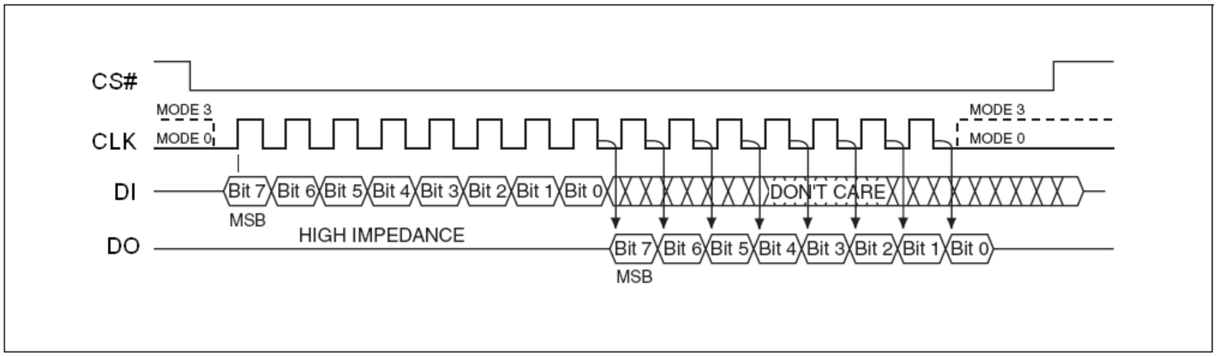
64KB Block	32KB Block	Sector	Address range		64KB Block	32KB Block	Sector	Address range	
511	1023	8191	1FFF000h	1FFFFFFh	255	511	4095	0FFF000h	0FFFFFFh
	1022	⋮	⋮	⋮		510	⋮	⋮	⋮
		8176	1FF0000h	1FF0FFFh			4080	0FF0000h	0FF0FFFh
510	1021	8175	1FEF000h	1FEFFFFh	254	509	4079	0FEF000h	0FEFFFFh
	1020	⋮	⋮	⋮		508	⋮	⋮	⋮
		8160	1FE0000h	1FE0FFFh			4064	0FE0000h	0FE0FFFh
509	1019	8159	1FDF000h	1FDFFFFh	253	507	4063	0FDF000h	0FDFFFFh
	1018	⋮	⋮	⋮		506	⋮	⋮	⋮
		8144	1FD0000h	1FD0FFFh			4048	0FD0000h	0FD0FFFh
508	1017	8143	1FCF000h	1FCFFFFh	252	505	4047	0FCF000h	0FCFFFFh
	1016	⋮	⋮	⋮		504	⋮	⋮	⋮
		8128	1FC0000h	1FC0FFFh			4032	0FC0000h	0FC0FFFh
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
259	519	4159	103F000h	103FFFFh	3	7	63	003F000h	003FFFFh
	518	⋮	⋮	⋮		6	⋮	⋮	⋮
		4144	1030000h	1030FFFh			48	0030000h	0030FFFh
258	517	4143	102F000h	102FFFFh	2	5	47	002F000h	002FFFFh
	516	⋮	⋮	⋮		4	⋮	⋮	⋮
		4128	1020000h	1020FFFh			32	0020000h	0020FFFh
257	515	4127	101F000h	101FFFFh	1	3	31	001F000h	001FFFFh
	514	⋮	⋮	⋮		2	⋮	⋮	⋮
		4112	1010000h	1010FFFh			16	0010000h	0010FFFh
256	513	4111	100F000h	100FFFFh	0	1	15	000F000h	000FFFFh
	512	⋮	⋮	⋮		0	⋮	⋮	⋮
		4096	1000000h	1000FFFh			0	0000000h	0000FFFh

OPERATING FEATURES

Standard SPI Modes

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

Figure 2. SPI Modes



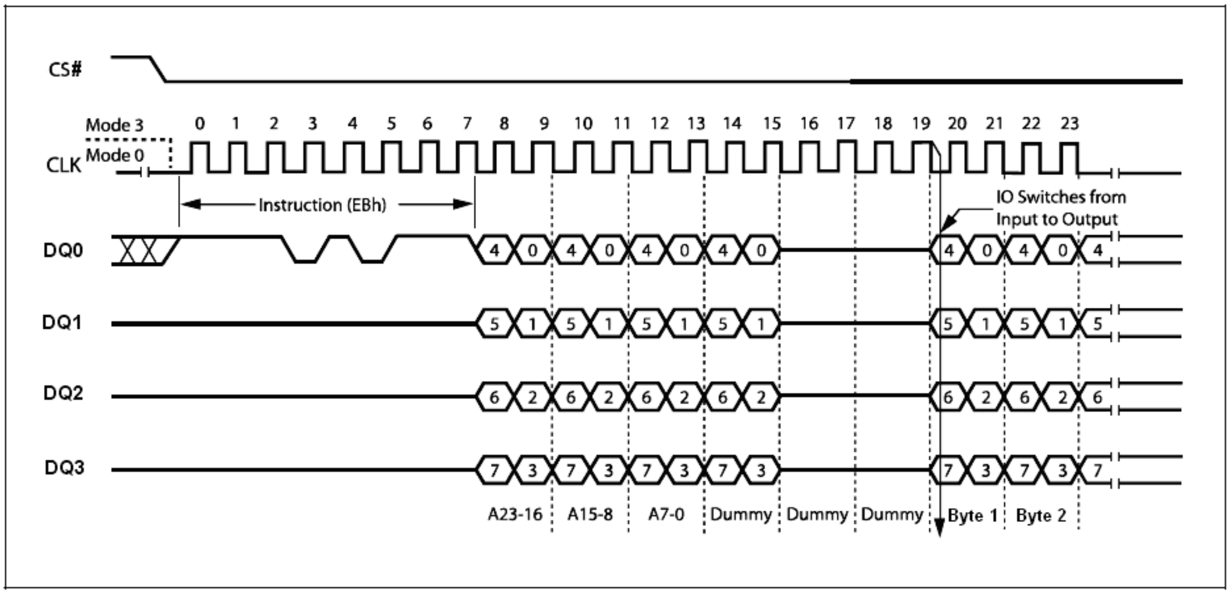
Dual SPI Instruction

The device supports Dual SPI operation when using the “Dual Output Fast Read and Dual I/ O FAST_READ” (3Bh/3Ch and BBh/BCh) 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 Instruction or Quad Output SPI Instruction

The device supports Quad output operation when using the Quad I/O Fast Read (EBh/ECh) or Quad Output Fast Read(6Bh/6Ch).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



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

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 or Write Status Register 2/3 (WRSR2/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, TB, BP3, BP2, BP1, BP0) bits allow part of the memory to be configured as read-only. This is the Software Protected Mode (SPM).
- The Write Protect (WP#) signal allows CMP, TB, BP3, BP2, BP1, BP0 bits and Status Register Protect (SRP) bit to be protected. This is the Hardware Protected Mode (HPM).
- In addition to the low power consumption feature, the Deep Power-down mode offers extra software protection from inadvertent Write, Program and Erase instructions, as all instructions are ignored except one particular instruction (the Release from Deep Power-down instruction).

Table 4. Protected Area Sizes Sector Organization

Status Register Content [1]						Memory Content [2]			
CMP	**TB**	**BP3**	**BP2**	**BP1**	**BP0**	**Protect Areas**	**Addresses**	**Density**	**Portion**
0	X	0	0	0	0	None	None	None	None
0	0	0	0	0	1	Block 511	1FF0000h-1FFFFFFh	64KB	Upper 1/512
0	0	0	0	1	0	Block 510 to 511	1FE0000h-1FFFFFFh	128KB	Upper 1/256
0	0	0	0	1	1	Block 508 to 511	1FC0000h-1FFFFFFh	256KB	Upper 1/128
0	0	0	1	0	0	Block 504 to 511	1F80000h-1FFFFFFh	512KB	Upper 1/64
0	0	0	1	0	1	Block 496 to 511	1F00000h-1FFFFFFh	1MB	Upper 1/32
0	0	0	1	1	0	Block 480 to 511	1E00000h-1FFFFFFh	2MB	Upper 1/16
0	0	0	1	1	1	Block 448 to 511	1C00000h-1FFFFFFh	4MB	Upper 1/8
0	0	1	0	0	0	Block 384 to 511	1800000h-1FFFFFFH	8MB	Upper 1/4
0	0	1	0	0	1	Block 256 to 511	1000000h-1FFFFFFh	16MB	Upper 1/2
0	1	0	0	0	1	Block 0	0000000h-000FFFFh	64KB	Lower 1/512
0	1	0	0	1	0	Block 0 to 1	0000000h-001FFFFh	128KB	Lower 1/256
0	1	0	0	1	1	Block 0 to 3	0000000h-003FFFFh	256KB	Lower 1/128
0	1	0	1	0	0	Block 0 to 7	0000000h-007FFFFh	512KB	Lower 1/64
0	1	0	1	0	1	Block 0 to 15	0000000h-00FFFFFh	1MB	Lower 1/32
0	1	0	1	1	0	Block 0 to 31	0000000h-01FFFFFh	2MB	Lower 1/16
0	1	0	1	1	1	Block 0 to 63	0000000h-03FFFFFh	4MB	Lower 1/8
0	1	1	0	0	0	Block 0 to 127	0000000h-07FFFFFh	8MB	Lower 1/4
0	1	1	0	0	1	Block 0 to 255	0000000h-0FFFFFFh	16MB	Lower 1/2
0	x	1	1	0	x	Block 0 to 511	0000000h-1FFFFFFh	32MB	All
0	x	1	x	1	x	Block 0 to 511	0000000h-1FFFFFFh	32MB	All
1	X	0	0	0	0	Block 0 to 511	0000000h-1FFFFFFh	All	All
1	0	0	0	0	1	Block 0 to 510	0000000h-1FEFFFFh	32,704 KB	Lower 511/512
1	0	0	0	1	0	Block 0 to 509	0000000h-1FDFFFFh	32,640 KB	Lower 255/256
1	0	0	0	1	1	Block 0 to 507	0000000h-1FBFFFFh	32,512 KB	Lower 127/128
1	0	0	1	0	0	Block 0 to 503	0000000h-1F7FFFFh	32,256 KB	Lower 63/64
1	0	0	1	0	1	Block 0 to 495	0000000h-1EFFFFFh	31MB	Lower 31/32
1	0	0	1	1	0	Block 0 to 479	0000000h-1DFFFFFh	30MB	Lower 15/16
1	0	0	1	1	1	Block 0 to 447	0000000h-1BFFFFFh	28MB	Lower 7/8
1	0	1	0	0	0	Block 0 to 383	0000000h-17FFFFFh	24MB	Lower 3/4
1	0	1	0	0	1	Block 0 to 255	0000000h-0FFFFFFh	16MB	Lower 1/2
1	1	0	0	0	1	Block 1 to 511	0010000h-1FFFFFFh	32,704 KB	Upper 511/512
1	1	0	0	1	0	Block 2 to 511	0020000h-1FFFFFFh	32,640 KB	Upper 255/256
1	1	0	0	1	1	Block 4 to 511	0040000h-1FFFFFFh	32,512 KB	Upper 127/128
1	1	0	1	0	0	Block 8 to 511	0080000h-1FFFFFFh	32,256 KB	Upper 63/64
1	1	0	1	0	1	Block 16 to 511	0100000h-1FFFFFFh	31MB	Upper 31/32
1	1	0	1	1	0	Block 32 to 511	0200000h-1FFFFFFh	30MB	Upper 15/16
1	1	0	1	1	1	Block 64 to 511	0400000h-1FFFFFFh	28MB	Upper 7/8
1	1	1	0	0	0	Block 128 to 511	0800000h-1FFFFFFh	24MB	Upper 3/4
1	1	1	0	0	1	Block 256 to 511	1000000h-1FFFFFFh	16MB	Upper 1/2

1	x	1	1	0	x	None	None	None	None
1	x	1	x	1	x	None	None	None	None

Note:

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

INSTRUCTIONS

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

The instruction set is listed in Instruction Set table. Every instruction sequence starts with a one-byte instruction code. Depending on the instruction, it might be followed by address bytes, or data bytes, or both or none. Chip Select (CS#) must be driven High after the last bit of the instruction sequence has been shifted in. In the case of a Read Data Bytes (READ), Read Data Bytes at Higher Speed (Fast_Read), Dual Output Fast Read, Dual I/O Fast Read, Quad Output Fast Read, Quad Input/Output FAST_READ, Read Status Register (RDSR), Read Status Register 2 (RDSR2), Read Status Register 3 (RDSR3), Read Status Register 4 (RDSR4), Read Extended Register 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 (or 32-bit, depends on mode state) 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 5. Instruction Set

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
RSTEN	66h						
RST (1)	99h						
Write Resume	30h/7Ah						
Write Suspend	B0h/75h						
Write Enable (WREN)	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)	(SR3.7-SR3.0)			
Write Status Register 2 (WRSR2)	31h/01h	SR2.7-SR2.0					
Write Status Register 3 (WRSR3)	C0h/11h/01h	SR3.7-SR3.0					
Deep Power-down	B9h						
Release from Deep Power-down, and read Device ID (RES)	ABh	dummy	dummy	dummy	(ID7-ID0)		(6)
Release from Deep Power-down (RDP)							
Manufacturer/ Device ID	90h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(7)
				01h	(ID7-ID0)	(M7-M0)	
Manufacturer/ Device ID by Dual I/O	92h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(7)
				01h	(ID7-ID0)	(M7-M0)	
Manufacturer/ Device ID by Quad I/O	94h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(7)
				01h	(ID7-ID0)	(M7-M0)	
Read Identification (RDID)	9Fh	(M7-M0)	(ID15-ID8)	(ID7-ID0)	(8)		
Read SFDP mode and Unique ID Number	5Ah	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(Next Byte) Continuous
Read Extended address Mode	C8h	(SR7-SR0) (4)					
Write Extended address Register	C5h	SR7-SR0					
Enter 4byte address mode	B7h						
Exit 4byte address mode	E9h						

Note:

1. RST command only executed if RSTEN command is executed first. Any intervening command will disable Reset.
2. Volatile Status Register Write Enable command must precede WRSR command without any intervening commands to write data to Volatile Status Register.
3. Data bytes are shifted with Most Significant Bit first. Byte fields with data in parenthesis “( )” indicate data being read from the device on the DO pin.
4. The Status Register contents will repeat continuously until CS# terminates the instruction.
5. The Device ID will repeat continuously until CS# terminates the instruction.
6. The Manufacturer ID and Device ID bytes will repeat continuously until CS# terminates the instruction.
   00h on Byte 4 starts with MID and alternate with DID, 01h on Byte 4 starts with DID and alternate with MID.
7. (M7-M0): Manufacturer, (ID15-ID8): Memory Type, (ID7-ID0): Memory Capacity.
8. WRSR (01h) support 8 or 16 or 24 bit register value input for status register, status register 2 and status register 3.

Table 6. Instruction Set (Read Instruction)

Instruction Name	OP Code	Address bits	Dummy bits / Clocks (Default)	Data Out	Remark
Read Data [1]	03h	24/32 bits	0	(D7-D0, …)	(Next Byte) continuous
Fast Read [1]	0Bh	24/32 bits	8 bits / 8 clocks	(D7-D0, …)	(Next Byte) continuous
Dual Output Fast Read [1]	3Bh	24/32 bits	8 bits / 8 clocks	(D7-D0, …)	(one byte Per 4 clocks, continuous)
Dual I/O Fast Read [1]	BBh	24/32 bits	8 bits / 4 clocks	(D7-D0, …)	(one byte Per 4 clocks, continuous)
Quad Output Fast Read [1]	6Bh	24/32 bits	8 bits / 8 clocks	(D7-D0, …)	(one byte per 2 clocks, continuous)
Quad I/O Fast Read [1]	EBh	24/32 bits	24 bits / 6 clocks	(D7-D0, …)	(one byte per 2 clocks, continuous)
Wrap read [1]	1Bh	24/32 bits	8 bits / 8 clocks	(D7-D0, …)	(Next Byte) continuous
Wrap Read with 4byte address	1Ch	32 bits	8 bits / 8 clocks	(D7-D0, …)	(Next Byte) continuous
Read Data with 4 bytes address	13h	32 bits	0	(D7-D0, …)	(Next Byte) continuous
Fast Read with 4 bytes address	0Ch	32 bits	8 bits / 8 clocks	(D7-D0, …)	(Next Byte) continuous
Dual Output Fast Read with 4 bytes address	3Ch	32 bits	8 bits / 8 clocks	(D7-D0, …)	(one byte Per 4 clocks, continuous)
Dual I/O Fast Read with 4 bytes address	BCh	32 bits	8 bits / 4 clocks	(D7-D0, …)	(one byte Per 4 clocks, continuous)
Quad Output Fast Read with 4 bytes address	6Ch	32 bits	8 bits / 8 clocks	(D7-D0, …)	(one byte per 2 clocks, continuous)
Quad I/O Fast Read with 4 bytes address	ECh	32 bits	24 bits / 6 clocks	(D7-D0, …)	(one byte per 2 clocks, continuous)
DDR Fast Read [1]	0Dh	24 /32 bits	8 bits / 4 clocks	(D7-D0, …)	(8 bits per 4 clocks, continuous)
DDR Dual I/O Fast Read [1]	BDh	24 /32 bits	8 bits / 2 clocks	(D7-D0, …)	(8 bits per 2 clock, continuous)
DDR Quad I/O Fast Read [1]	EDh	24 /32 bits	24 bits / 3 clocks	(D7-D0, …)	(8 bits per 1 clock, continuous)
DDR Read Burst with Wrap [1]	1Dh	24 /32 bits	8 bits / 4 clocks	(D7-D0, …)	(8 bits per 4 clock, continuous)
Read OTP array [1]	48h	24 /32 bits	8 bits / 8 clocks	(D7-D0, …)	(Next Byte) continuous

Note:

1. The address cycles default is 3-byte address mode. If using 4byte address, please issue enter 4byte address mode first.

Table 7. Instruction Set (Program Instruction)

Instruction Name	OP Code	Address bits	Dummy bits / Clocks (Default)	Data Out	Remark
Page Program (PP) (1)	02h	24/32 bits	0	(D7-D0, …)	(Next Byte) continuous
Quad Input Page Program (QPP) (1)	32h	24/32 bits	0	(D7-D0, …)	(one byte per 2 clocks, continuous)
Page Program (PP) with 4 bytes address	12h	32 bits	0	(D7-D0, …)	(Next Byte) continuous
Quad Input Page Program (QPP) with 4 bytes address	34h	32 bits	0	(D7-D0, …)	(one byte per 2 clocks, continuous)
DDR Mode Page Program (1)	D2h	24 /32 bits	0	(D7-D0, …)	(8 bits per 1 clock, continuous)
Program OTP array (1)	42h	24 /32 bits	0	(D7-D0, …)	(Next Byte) continuous

Note:

1. The address cycles default is 3-byte address mode. If using 4byte address, please issue enter 4byte address mode first.

Table 8. Instruction Set (Erase Instruction)

Instruction Name	OP Code	Address bits	Remark
Sector Erase (SE) (1)	20h	24/32 bits	
32K Half Block Erase (HBE) (1)	52h	24/32 bits	
64K Block Erase (BE) (1)	D8h	24/32 bits	
Chip Erase (CE)	C7h/ 60h	0	
Sector Erase (SE) with 4 bytes address	21h	32 bits	
32K Half Block Erase (HBE) with 4 bytes address	5Ch	32 bits	
64K Block Erase (BE) with 4 bytes address	DCh	32 bits	
Erase OTP array (1)	44h	24/32 bits	

Note:

1. The address cycles default is 3-byte address mode. If using 4byte address, please issue enter 4byte address mode first.

Table 9. 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 Counte rData [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.
5. All RPMC instructions are not support HOLD# function.

Table 10. Manufacturer and Device Identification

OP Code	(M7-M0)	(ID15-ID0)	(ID7-ID0)
ABh			18h
90h/92h/94h	1Ch		18h
9Fh	1Ch	7819h	

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. 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 4. Reset-Enable and Reset Sequence Diagram

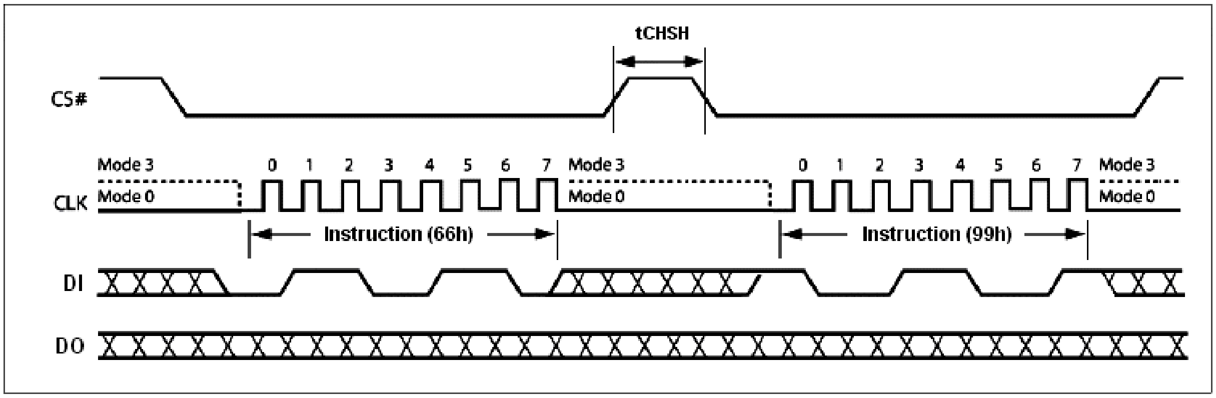
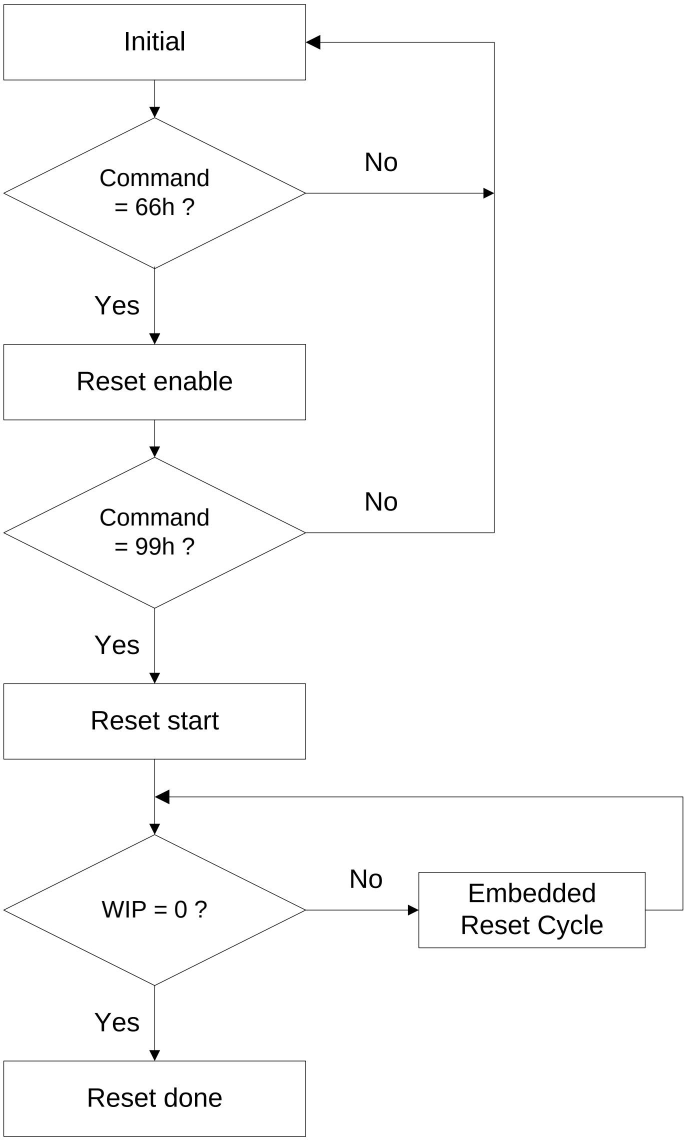


Figure 5. Software Reset Flow



Note:

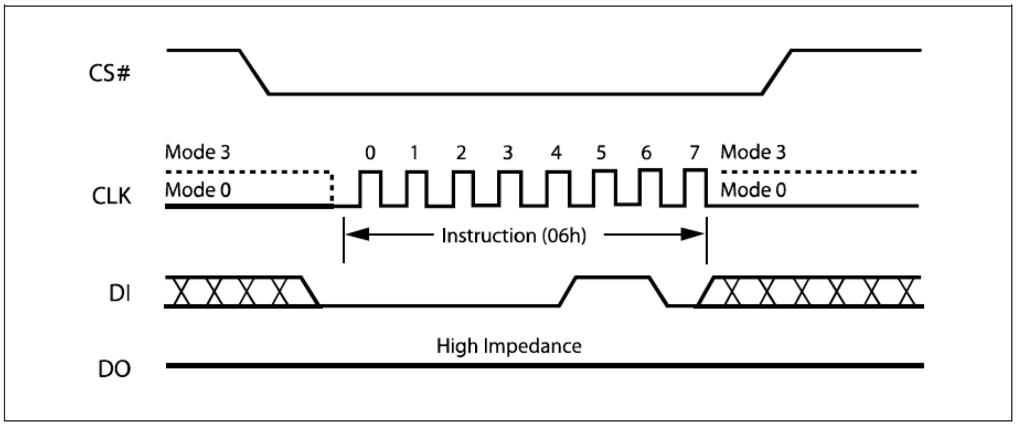
1. Reset-Enable (RSTEN) (66h) and Reset (RST) (99h) commands need to match standard SPI mode.
2. The reset command could be executed during embedded program and erase process back to SPI mode.
3. This flow can not release the device from Deep power down mode.
4. The Status Register Bits and Status Register 2/3 Bits will reset to default value after reset done.
5. If user reset device during erase, the embedded reset cycle software reset latency will take about 28 us in worst case.
6. User can’t do software reset command while doing 4K/32K erase operation.

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/ WRSR2/WRSR3/WRSR4) instruction.

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

Figure 6. Write Enable Instruction Sequence Diagram

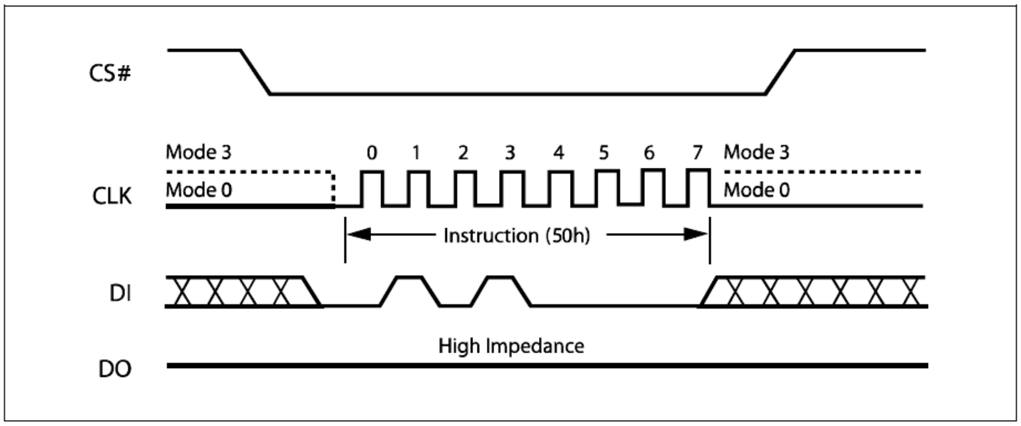


Volatile Status Register Write Enable (50h)

This feature enable user to change memory protection schemes quickly without waiting for the typical non-volatile bit write cycles or affecting the endurance of the Status Register non-volatile bits. The Volatile Status Register Write Enable (50h) command won't set the Write Enable Latch (WEL) bit, it is only valid for 'Write Status Register' (01h) command to change the Volatile Status Register bit values.

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

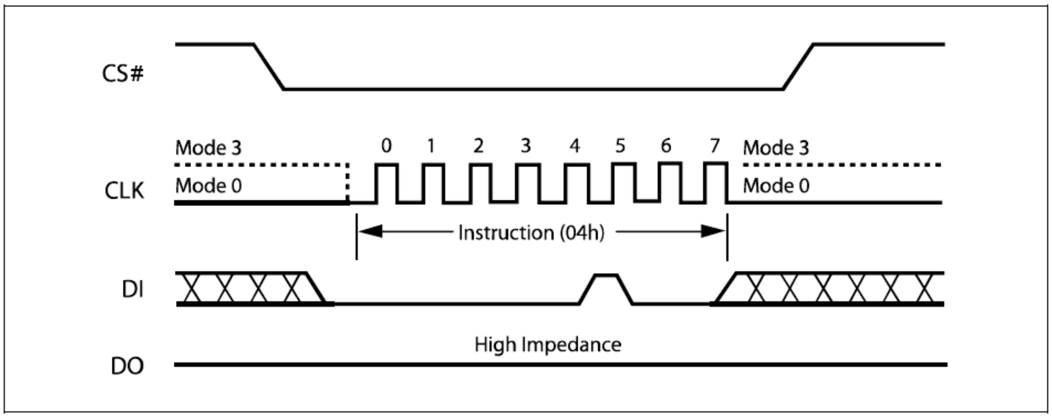
Figure 7. Volatile Status Register Write Enable Instruction Sequence Diagram



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.

Figure 8. Write Disable Instruction Sequence Diagram



Read Status Register (RDSR) (05h)

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

Figure 9. Read Status Register Instruction Sequence Diagram

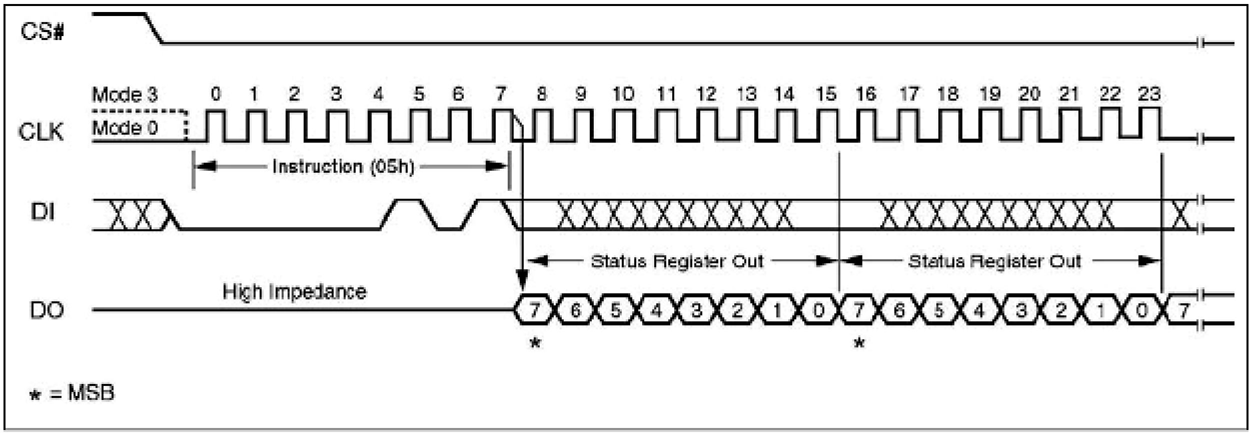


Table 11. Status Register Bit Locations

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
SRP bit (Status Register Protect)	**TB bit** (Top / Bottom Protect)	**BP3 bit** (Block Protected)	**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 = Bottom 0 = Top (default 0)	(note 1)	(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 bits of the Status Register (TB, BP3, BP2, BP1, BP0 and CMP) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

TB bit. The Top/Bottom Protect Bit (TB) controls if the Block Protect Bits (BP3, 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.

BP3, BP2, BP1, BP0 bits. The Block Protect (BP3, BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against Program and Erase instructions. These bits are written with the Write Status Register (WRSR) instruction. When one or both of the Block Protect (BP3, BP2, BP1, BP0) bits is set to 1, the relevant memory area (as defined in Protected Area Sizes Sector Organization table.) becomes protected against Page Program (PP/PP4B), Quad Input Page Program (QPP/QPP4B), Sector Erase (SE/SE4B) and Block Erase (HBE/HBE4B//BE/BE4B) instructions. The Block Protect (BP3, 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, TB, BP3, BP2, BP1, BP0) bits.

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 program/erase 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.

Figure 10. Read Status Register 2 Instruction Sequence Diagram

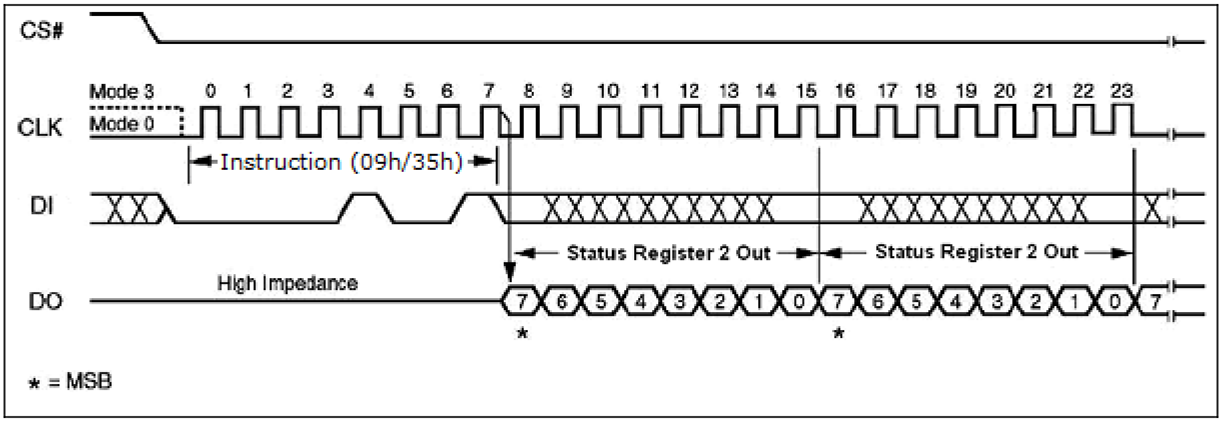


Table 12. Status Register 2 Bit Locations

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 volatile bit is “0” at Power-up or after reset.
2. See the “Protected Area Sizes Sector Organization” table.
3. When executed the (RDSR2) (09h/35h) command, the WIP (SR2.0) value is the same as WIP (SR0) in Status Register Bit Locations 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 Status Register 2. It is used in conjunction with TB, BP3, BP2, BP1, BP0 bits to provide mode flexibility for the array protection. 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 WRSR2 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 WRSR2 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 WRSR2 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 Read/Write bit in the Status Register 2 to disable WP# and Hold#/RESET# before Quad operation. When it is “0”, the WP# and HOLD#/RESET# are enabled. On the other hand, while QE bit is “1” (factory default), the WP# and HOLD#/RESET# are disabled.

No matter QE is “0” or “1”, the system can executes Quad Input/Output FAST_READ (EBh) command directly. User can use Flash Programmer to set QE bit as “1” and then the host system can let WP# and HOLD# keep floating in SPI mode.

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 program/erase 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.

Figure 11. Read Status Register 3 Instruction Sequence Diagram

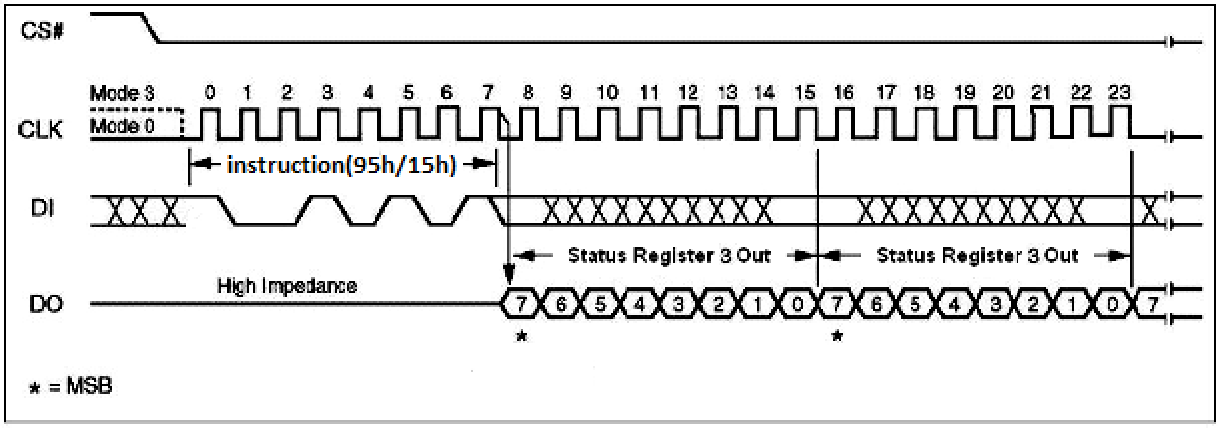


Table 13. Status Register 3 Bit Locations

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**		**Blank check bit**	**4byteP bit** (address mode select)	**4byte bit** (Byte of Address flag)
1 = RESET# enable 0 = HOLD# enable (default 0)	00 = 67% (default) 01 = 100% 10 = 83% (1/2) drive 11 = 50% (1/3) drive		00 = 8 Bytes(default) 01 = 16 Bytes 10 = 32 Bytes 11 = 64 Bytes		**1 = flash is blank after ship out (default)** 0 = flash had been programmed	1 = 4byte address mode 0= 3byte address mode (default 0)	1 = 4 byte address 0 = 3 byte address
Non-volatile/volatile bit	Non-volatile/volatile bit		Non-volatile/volatile bit		Indicator bit	Non-volatile bit	Indicator 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.

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

4byteP bit. The 4byteP bit is a non-volatile bit that determines the initial address mode when the device is powered on or reset. This bit is only used during the power on or device reset initialization period, and it is only writable by the non-volatile Write Status sequence (06h + C0h/11h/01h). When 4byteP=0 (factory default), the device will power up into 3-Byte Address Mode, the Extended Address Register must be used to access memory regions beyond 128Mb. When 4byteP=1, the device will power up into 4-Byte Address Mode directly.

4byte bit. The 4byte flag bit indicates bit is a read only bit in the Status Register 3 that indicates which address mode the device is currently operating in. When 4byte=0, the device is in the 3-Byte Address Mode, when 4byte=1, the device is in the 4-Byte Address Mode.

Write Status Register (WRSR) (01h)

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

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

The instruction sequence is shown in Write Status Register Instruction Sequence Diagram figure and Write Status Register Instruction Sequence Diagram (multi byte) figure. The Write Status Register (WRSR) instruction has no effect on S1 and S0 of the Status Register. Chip Select (CS#) must be driven High after the eighth bit or 16th or 24th 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 SRP bit, TB bit and Block Protect (BP3, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in 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.

WRSR instruction also support multi byte for setting status register, status register2 and status register 3.

Figure 12. Write Status Register Instruction Sequence Diagram

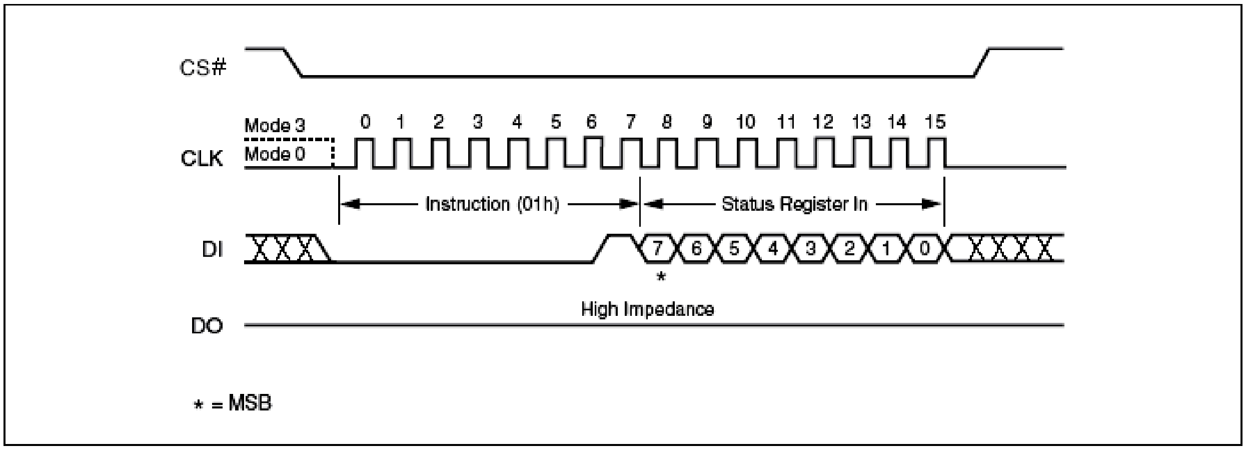
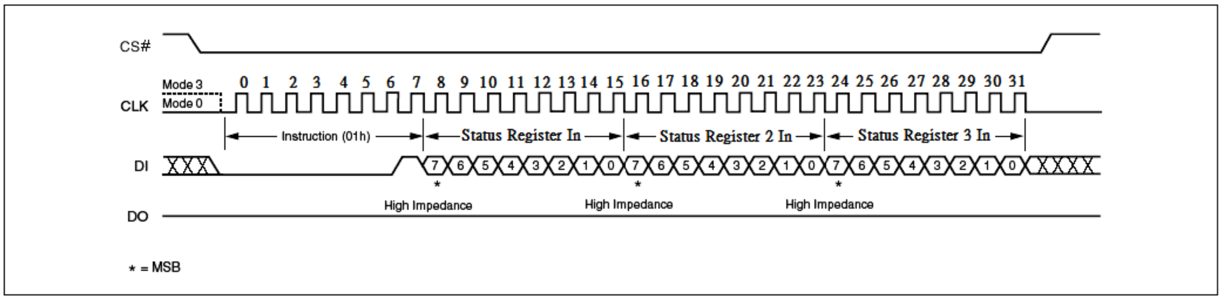


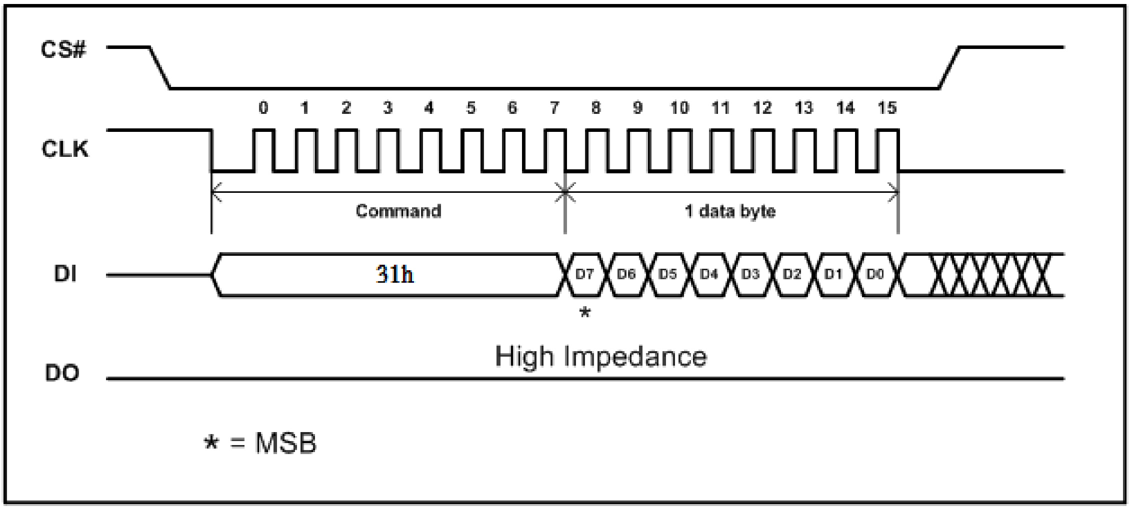
Figure 13. Write Status Register Instruction Sequence Diagram (multi byte)



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, most significant nibble first.
01h (WRSR) command also can set status register 2.

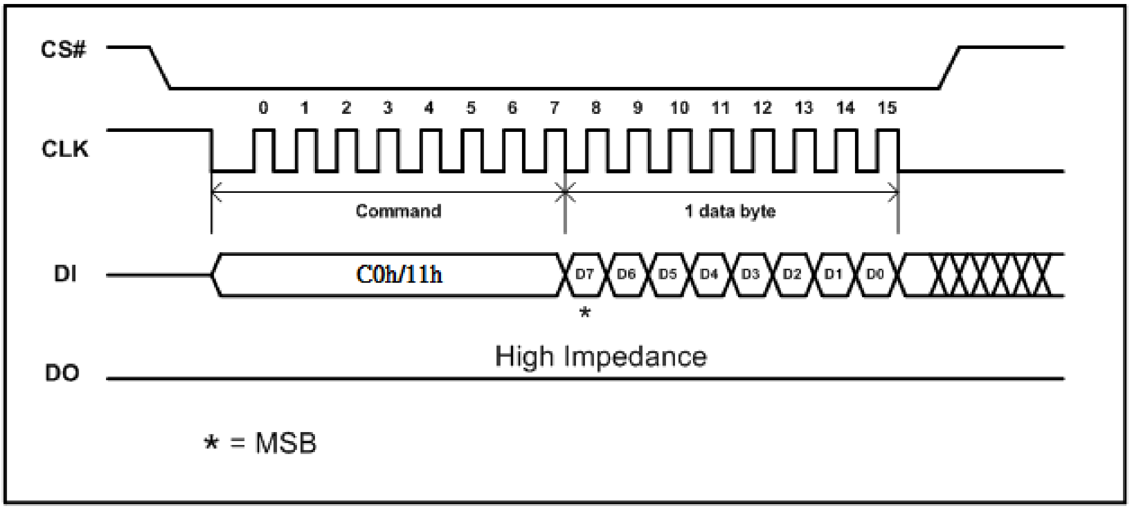
Figure 14. Write Status Register 2 Instruction Sequence Diagram



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

The Write Status Register 3 (C0h/11h) command can be used to set output drive strength in I/O pins, HOLD/RESET# selection, burst read length setting and 4byteP. To set these bits to the host driver CS# low, sends the Write Status Register 3 (C0h or 11h) and one data byte, then drivers CS# high, most significant nibble first.
01h (WRSR) command also can set status register 3.

Figure 15. Write Status Register 3 Instruction Sequence Diagram



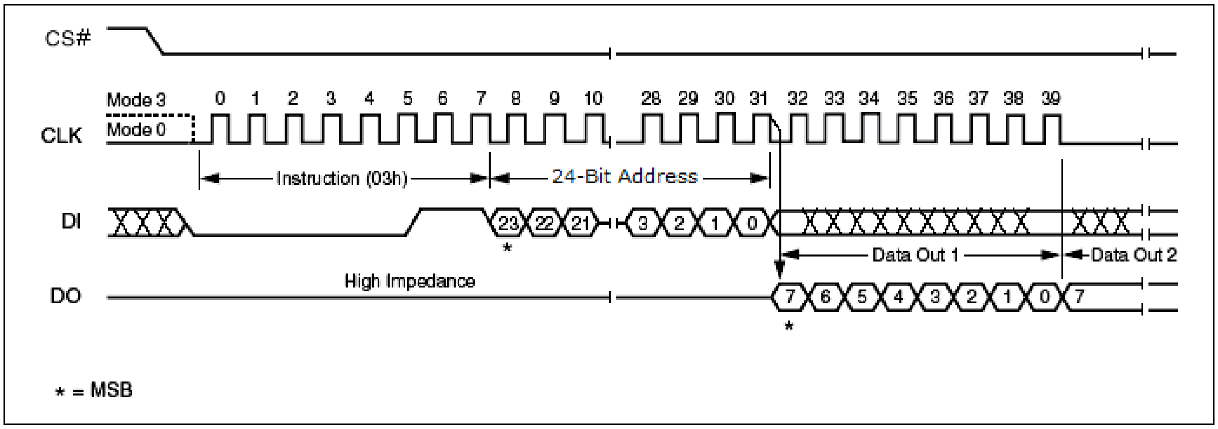
Read Data Bytes (READ) (03h)

The device is first selected by driving Chip Select (CS#) Low. The instruction code for the Read Data Bytes (READ) instruction is followed by a 3-byte address (A23-A0) or 4-byte address (A31-A0) (depending on mode state), 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Read Data Bytes (03h). The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single Read Data Bytes (READ) instruction. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

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

Figure 16. Read Data Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode. For 4-byte address mode, the address cycles will be increased.

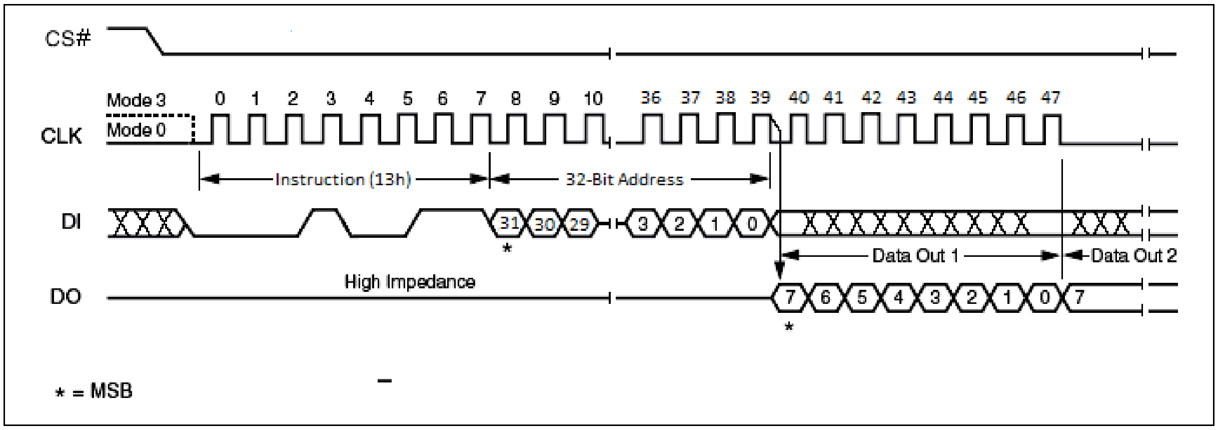
Read Data Bytes with 4byte address (READ4A) (13h)

The device is first selected by driving Chip Select (CS#) Low. The instruction code for the Read Data Bytes with 4bytes address (READ4A) instruction is followed by a 4-byte address (A31-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 with 4byte address 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 with 4byte address (READ4A) 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 with 4 byte address (READ4A) 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 with 4byte address (READ4A) 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 17. Read Data with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

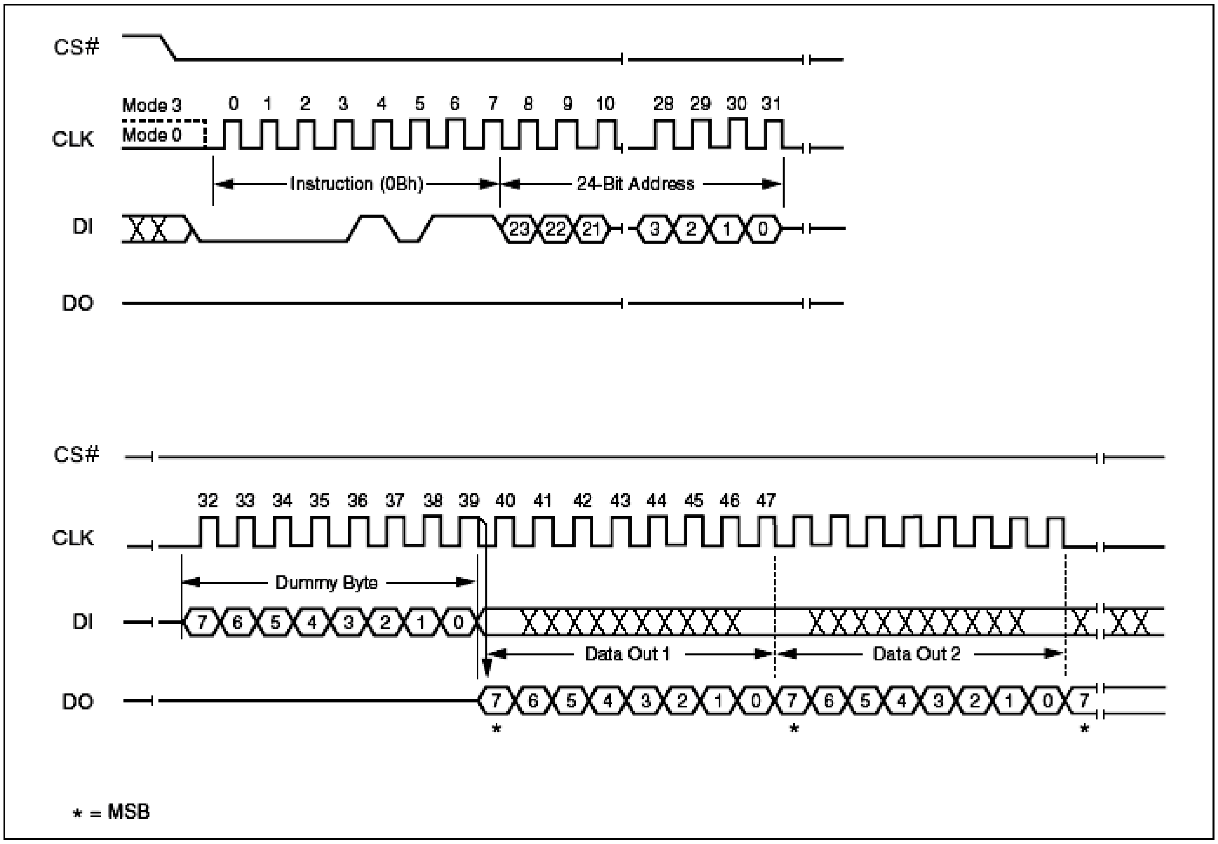
Read Data Bytes at Higher Speed (FAST_READ) (0Bh)

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) or 4-byte address (A31-A0) (depending on address mode state) 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Read Data Bytes at Higher Speed (0Bh). 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.

Figure 18. Fast Read Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode. For 4-byte address mode, the address cycles will be increased.

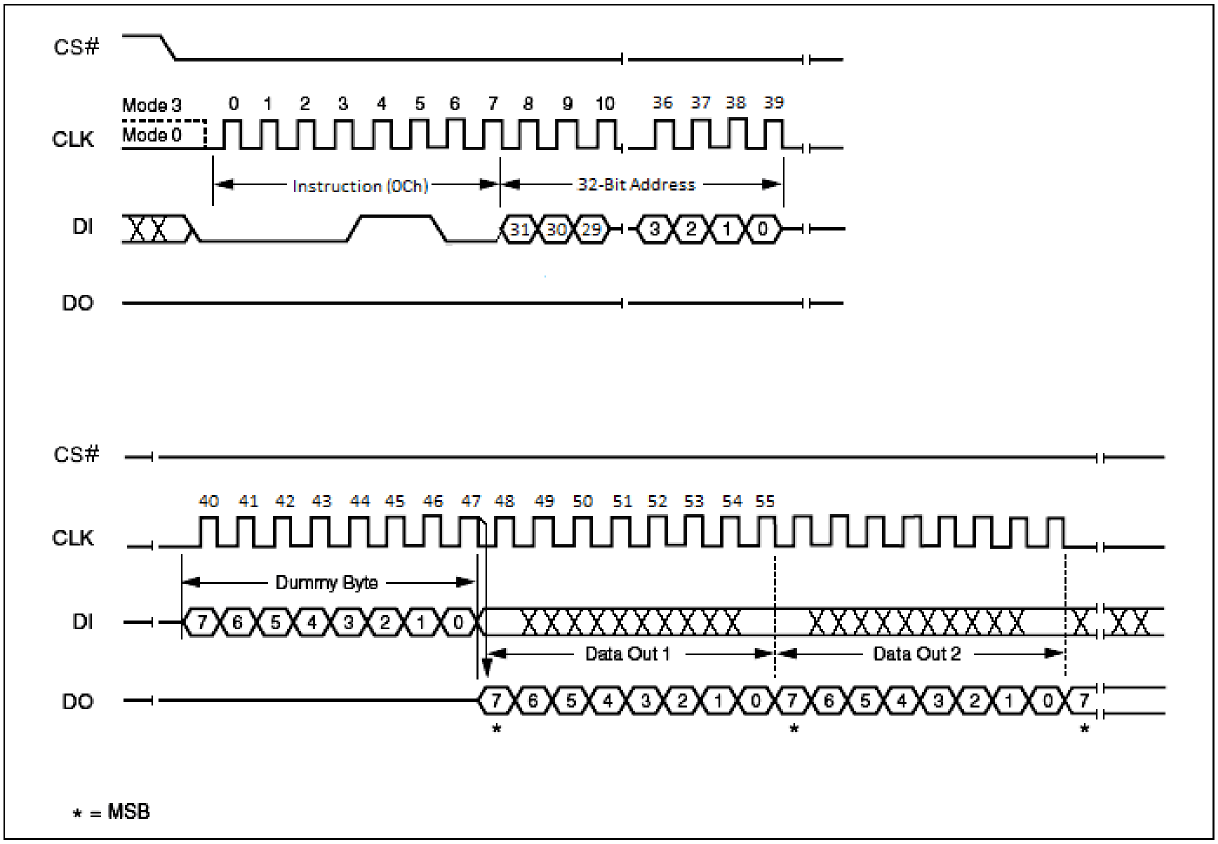
Read Data Bytes at Higher Speed with 4byte address (FAST_READ4A) (0Ch)

The device is first selected by driving Chip Select (CS#) Low. The instruction code for the Read Data Bytes at Higher Speed with 4byte address (FAST_READ4A) instruction is followed by a 4-byte address (A31-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 with 4byte address 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 with 4byte address (FAST_READ4A) 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 with 4byte address (FAST_READ4A) 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 with 4byte address (FAST_READ4A) 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 19. Fast Read with 4byte address Instruction Sequence Diagram



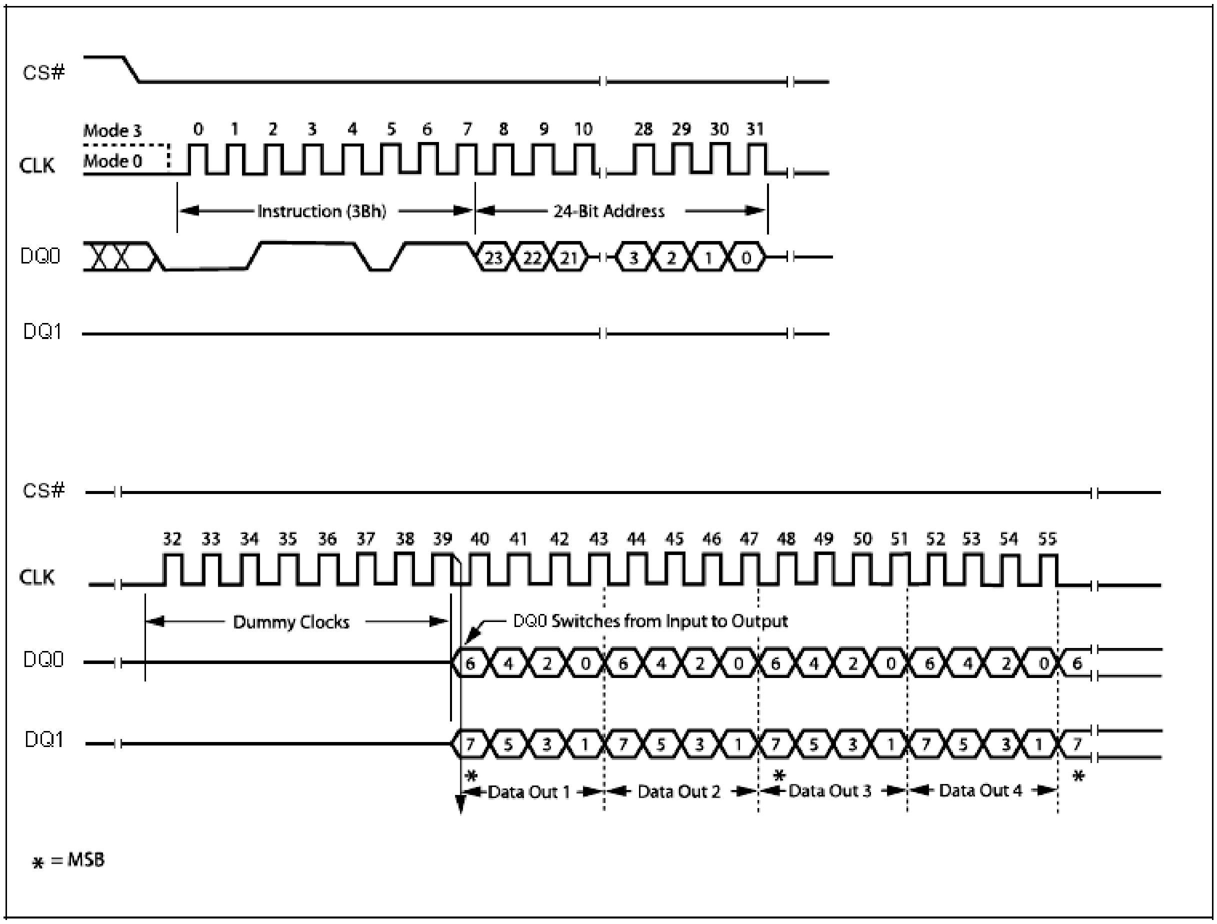
Note: This instruction is workable while in 4byte address mode.

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 device s. 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 or 32-bit address (depends on address mode state) 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 20. Dual Output Fast Read Instruction Sequence Diagram



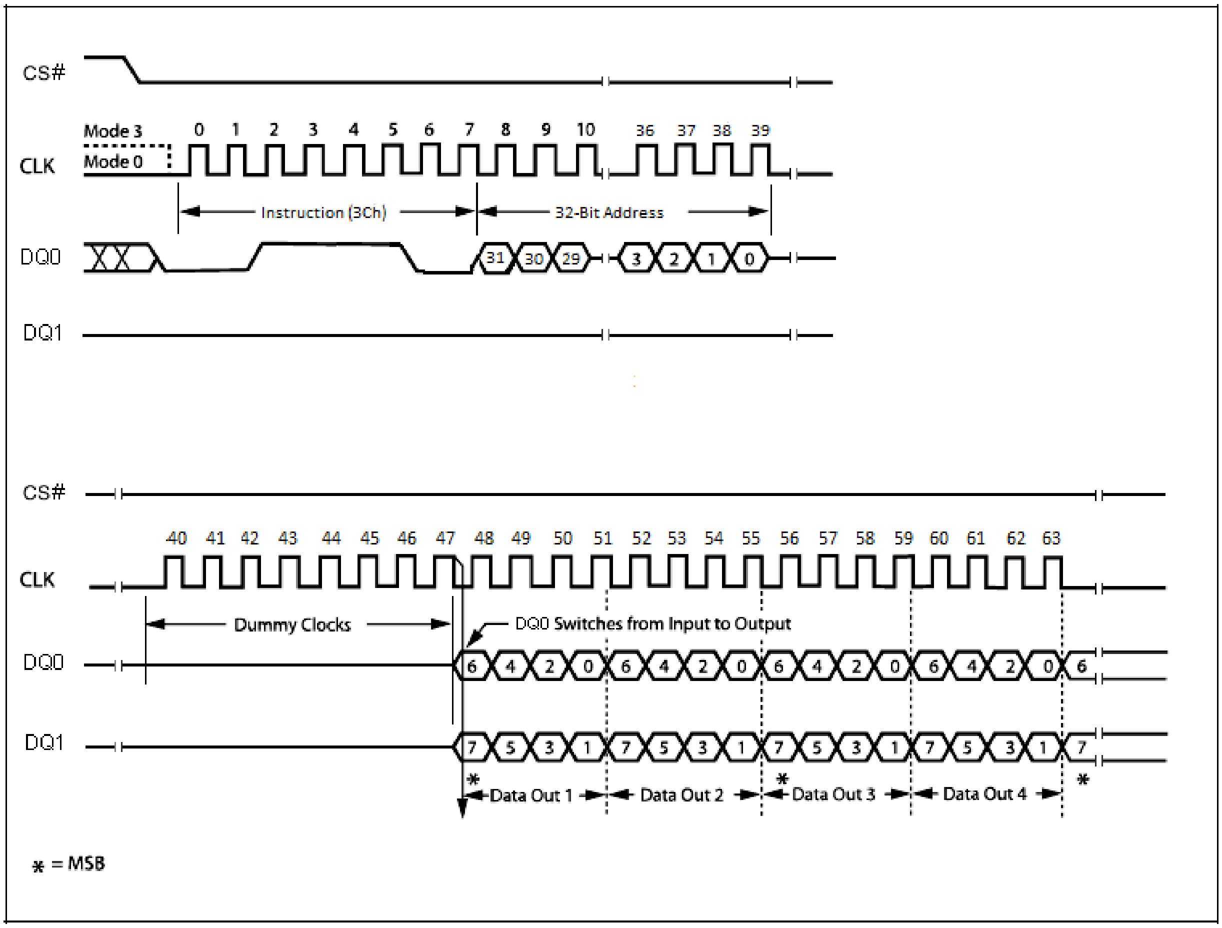
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Dual Output Fast Read with 4byte address (3Ch)

The Dual Output Fast Read with 4byte address (3Ch) is similar to the standard Fast Read with 4byte address (0Ch) 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 device's. The Dual Output Fast Read with 4byte address 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 with 4byte address instruction, the Dual Output Fast Read with 4byte address instruction can operation at the highest possible frequency of F_R (see AC Electrical Characteristics). This is accomplished by adding eight dummy clocks after the 32-bit address as shown in Dual Output Fast Read with 4byte address 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 21. Dual Output Fast Read with 4byte address Instruction Sequence Diagram



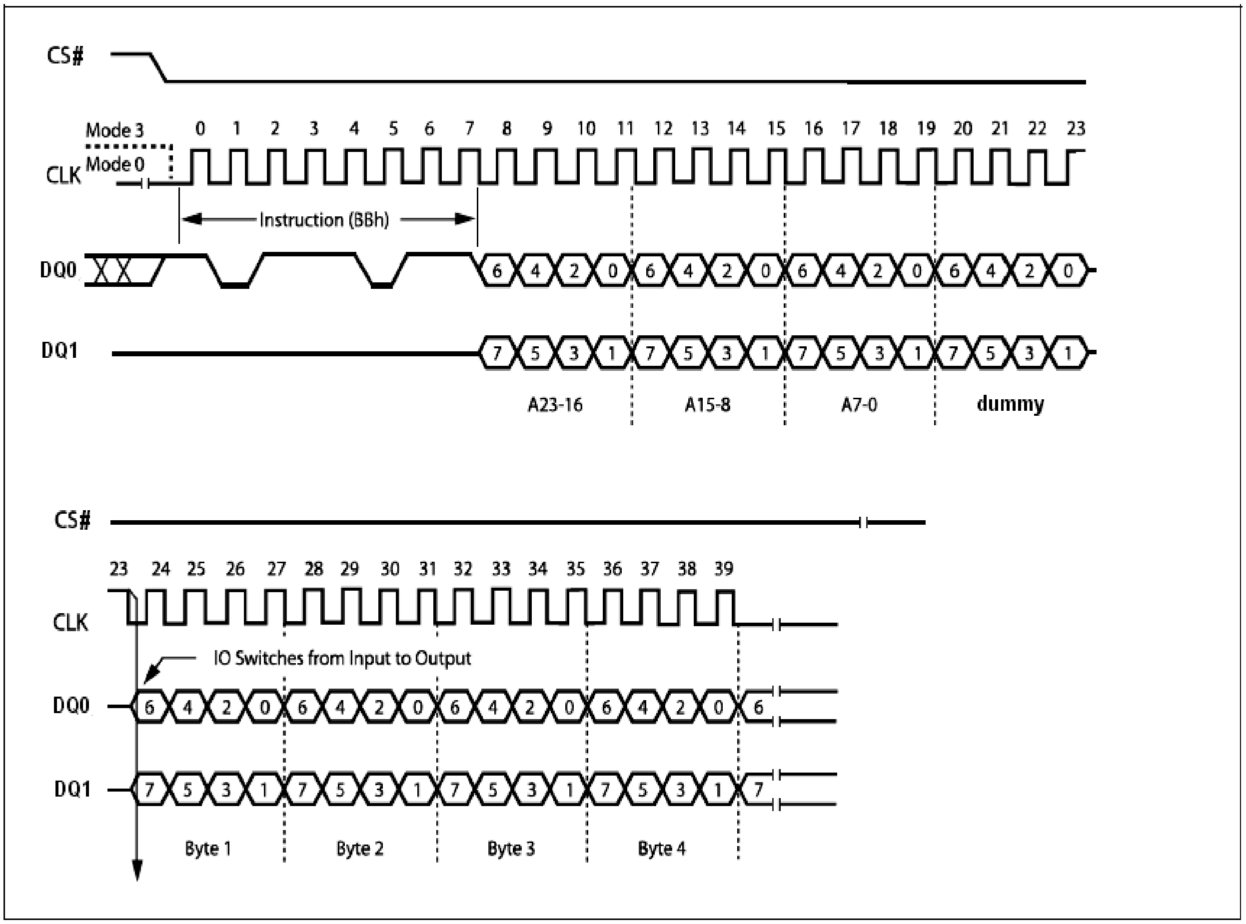
Note: This instruction is workable while in 4byte address mode.

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 or A31-0, depends on address mode state) 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Dual Input / Output FAST_READ (BBh). 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 22. Dual Input / Output Fast Read Instruction Sequence Diagram



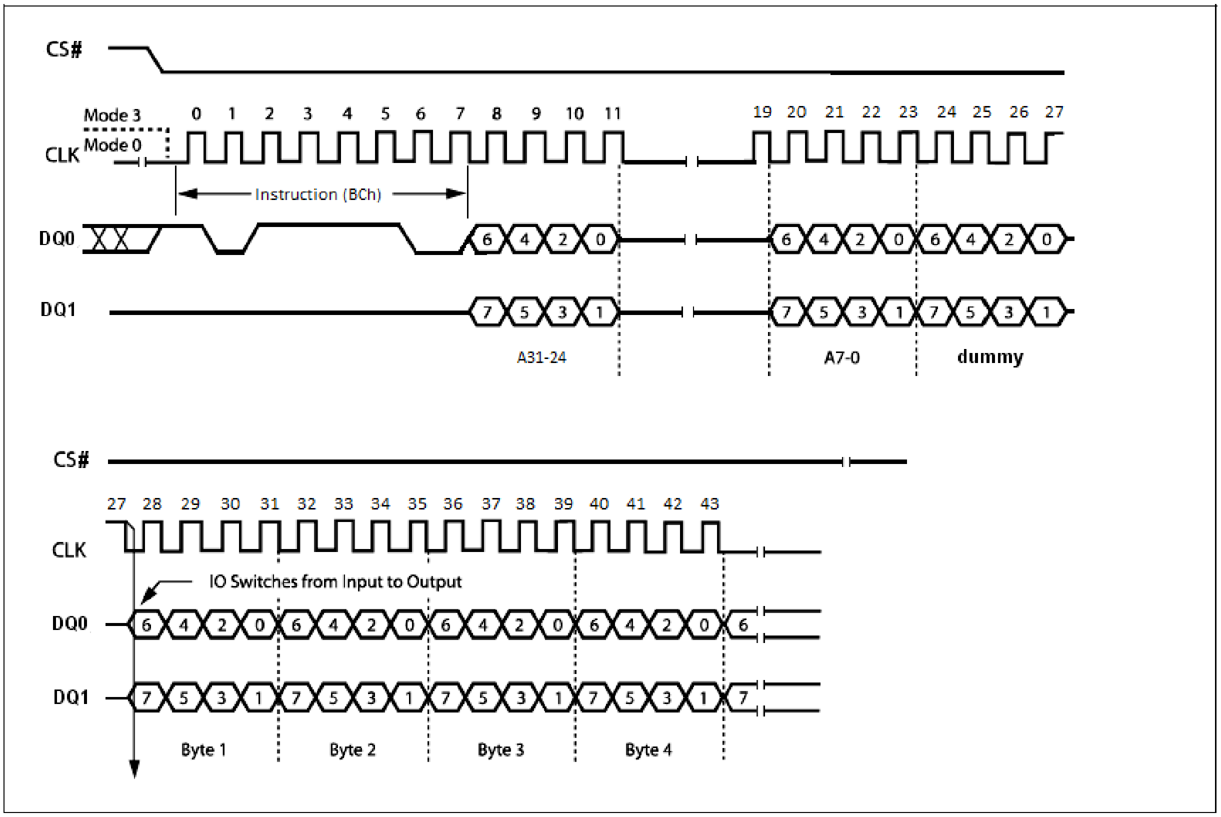
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Dual Input / Output FAST_READ with 4byte address (BCh)

The Dual I/O Fast Read with 4byte address (BCh) 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 with 4byte address (3Ch) instruction but with the capability to input the Address bits (A31-0) 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 with 4byte address 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 with 4byte address instruction. The address counter rolls over to 0 when the highest address has been reached. Once writing Dual I/O Fast Read with 4byte address 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 with 4byte address Instruction Sequence Diagram figure.

Figure 23. Dual Input / Output Fast Read with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

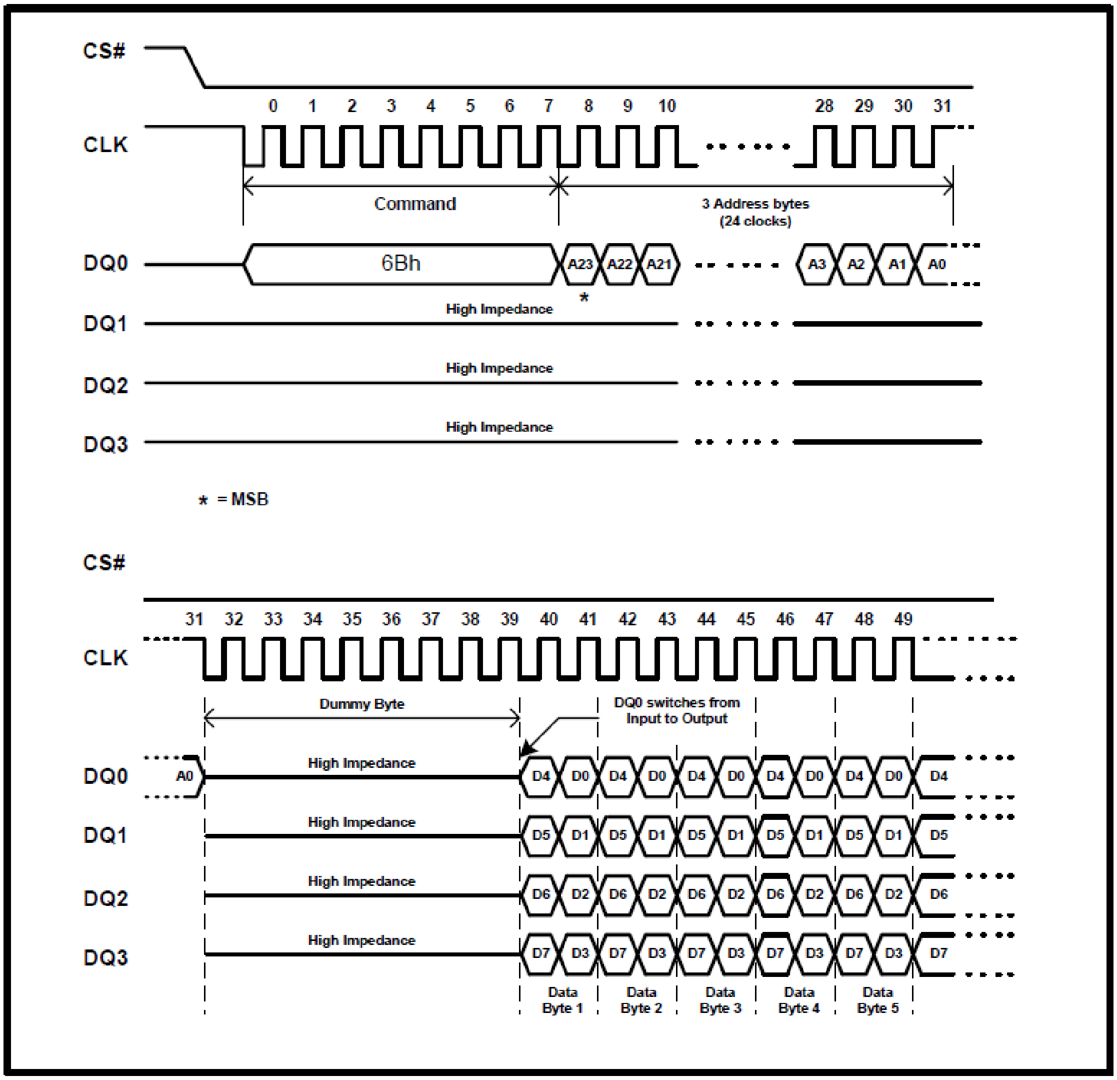
Quad Output Fast Read (6Bh)

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 Fast Read (6Bh) address is latching on rising edge of CLK, and data of every four bits (interleave on 4 I/O pins) shift out on the falling edge of CLK at a maximum frequency FR. The first address can be any location. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Quad Output FAST_READ (6Bh). 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/32-bit address on DQ0 (depends on address mode state) -> 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#/RESET# (DQ3) need to drive high before address input if WP#, HOLD# or RESET# is enable.

Figure 24. Quad Output Fast Read Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

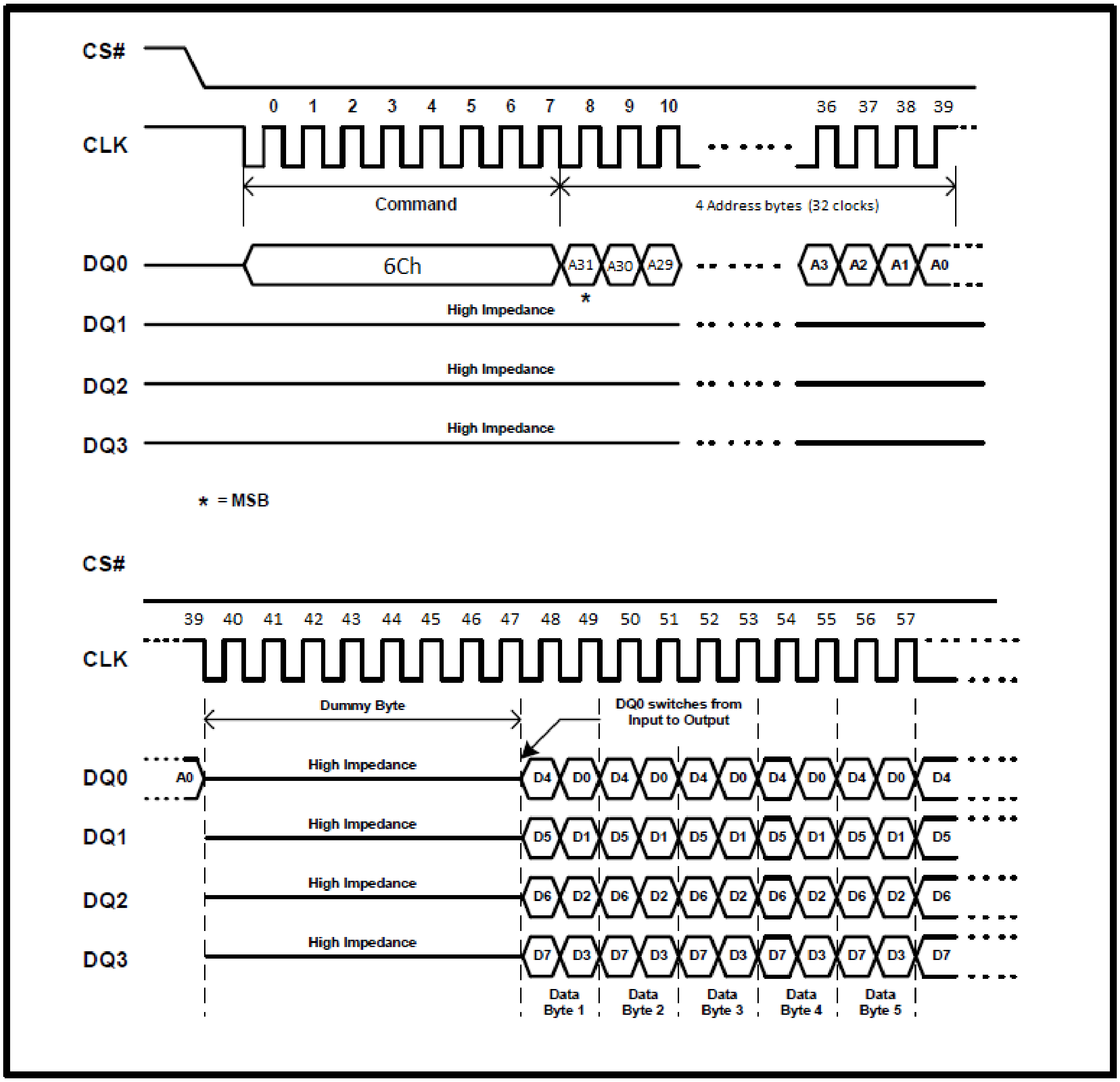
Quad Output Fast Read with 4byte address (6Ch)

The Quad Output Fast Read with 4byte address (6Ch) instruction is similar to the Dual Output Fast Read with 4byte address(3Ch) 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 Fast Read with 4byte address(6Ch) address is latching on rising edge of CLK, and data of every four bits (interleave on 4 I/O pins) shift out on the falling edge of CLK at a maximum frequency FR. The first address can be any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Quad Output Fast Read with 4byte address instruction. The address counter rolls over to 0 when the highest address has been reached.

The sequence of issuing Quad Output Fast Read with 4byte address (6Ch) instruction is: CS# goes low -> sending Quad Output Fast Read with 4byte address (6Ch) instruction -> 32-bit address on DQ0 -> 8 dummy clocks -> data out interleave on DQ3, DQ2, DQ1 and DQ0 -> to end Quad Output Fast Read with 4byte address(6Ch) operation can use CS# to high at any time during data out, as shown in Quad Output Fast Read with 4byte address Instruction Sequence Diagram figure. The WP# (DQ2) and HOLD#/RESET# (DQ3) need to drive high before address input if WP#, HOLD# or RESET# is enable.

Figure 25. Quad Output Fast Read with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

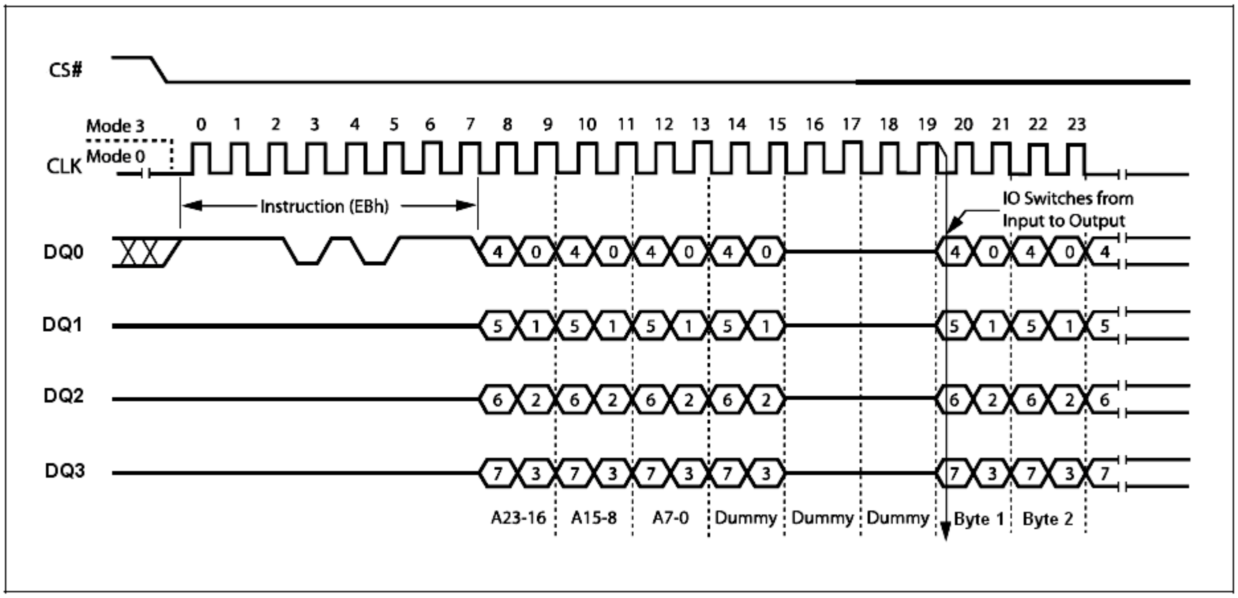
Quad Input / Output FAST_READ (EBh)

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 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Quad Input/Output FAST_READ (EBh).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(or 32-bit, depends on address mode state) 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.

Figure 26. Quad Input / Output Fast Read Instruction Sequence Diagram



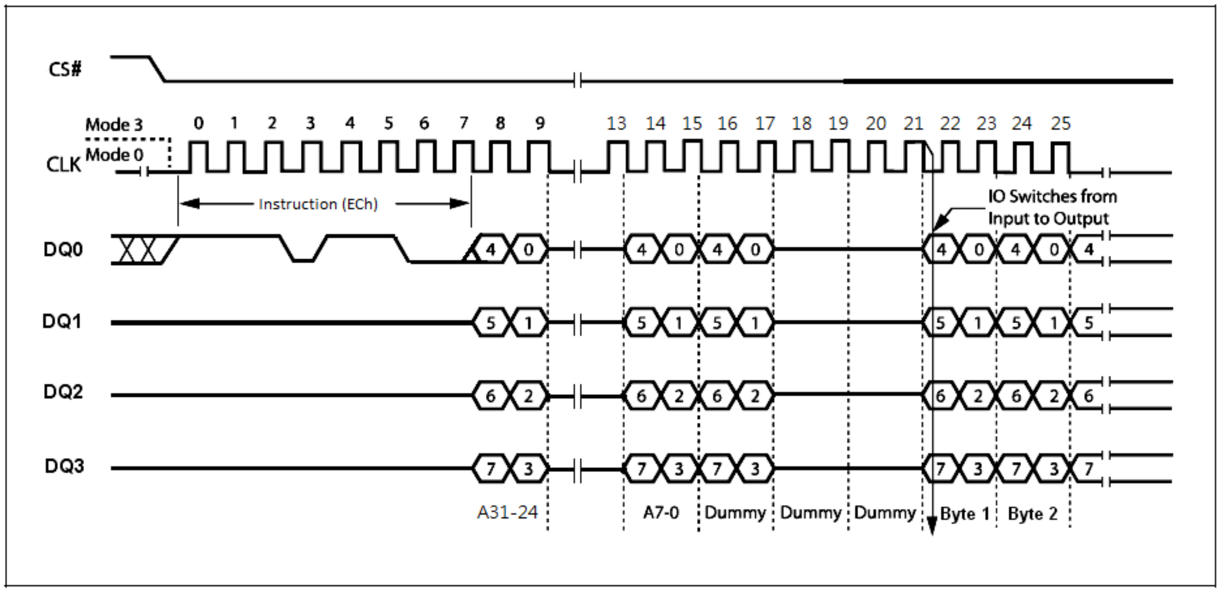
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Quad Input / Output FAST_READ with 4byte address (ECh)

The Quad Input/Output FAST_READ with 4byte address (ECh) instruction is similar to the Dual I/O Fast Read with 4byte address (BCh) 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 Input/Output FAST_READ with 4byte address (ECh) 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 with 4byte address instruction. The address counter rolls over to 0 when the highest address has been reached. Once writing Quad Input / Output FAST_READ with 4byte address 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 with 4byte address (ECh) instruction is: CS# goes low -> sending Quad Input/Output FAST_READ (EBh) instruction -> 32-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 with 4byte address (ECh) operation can use CS# to high at any time during data out, as shown in Quad Input / Output Fast Read with 4byte address Instruction Sequence Diagram figure.

Figure 27. Quad Input / Output Fast Read with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

Read Burst (1Bh)

This device supports Read Burst with wrap in SPI mode. To execute a Read Burst with wrap operation the host drivers CS# low, and sends the Read Burst with wrap (1Bh) command cycle, followed by three address bytes (or four address bytes depends on address mode state) and one dummy byte (8 clocks) in SPI mode (Read Burst Instruction Sequence Diagram figure).

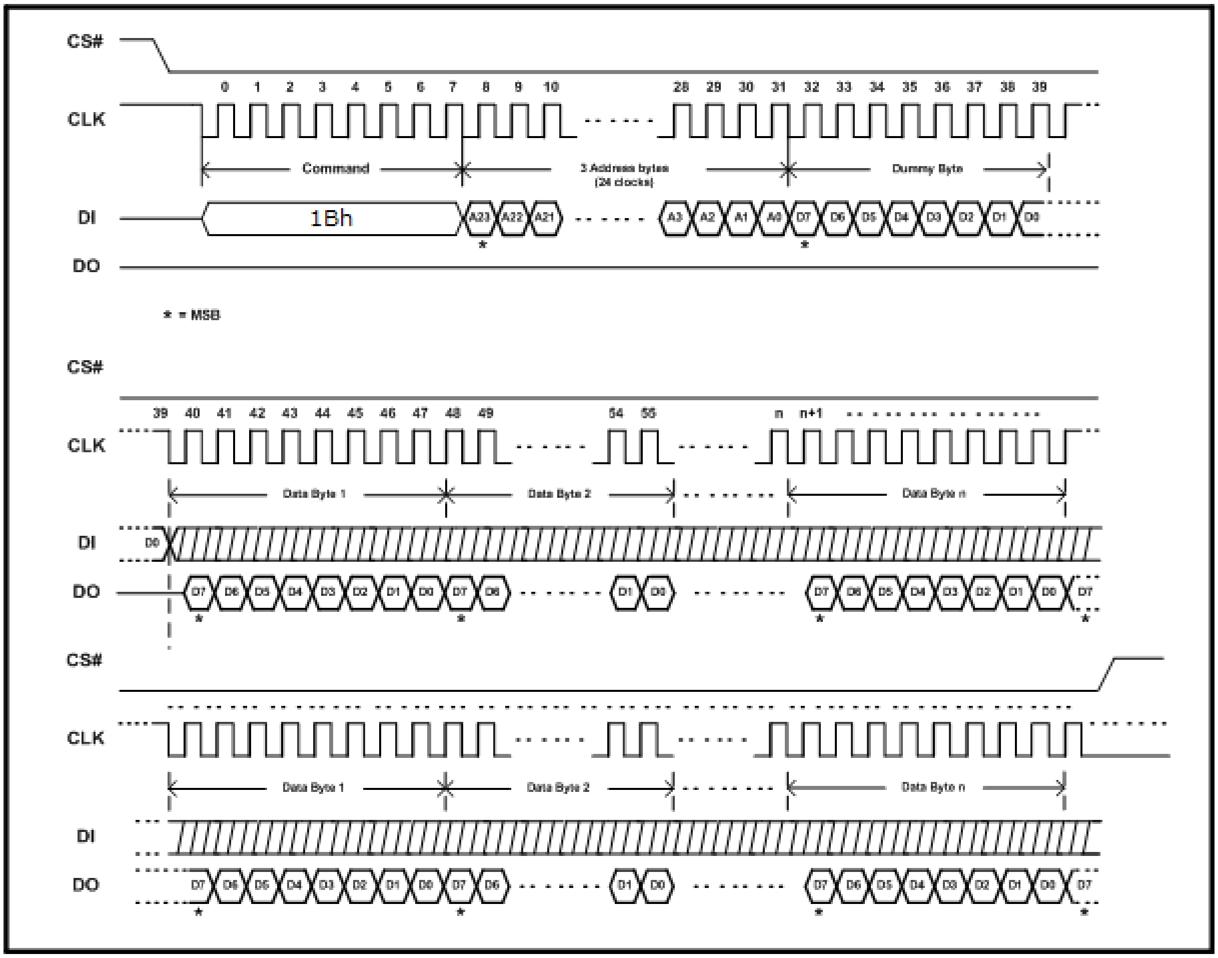
After the dummy byte, the device outputs data on the falling edge of the CLK signal starting from the specific address location. The data output stream is continuous through all addresses until terminated by a low-to high transition of CS# signal.

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.

Table 14. 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 28. Read Burst Instruction Sequence Diagram



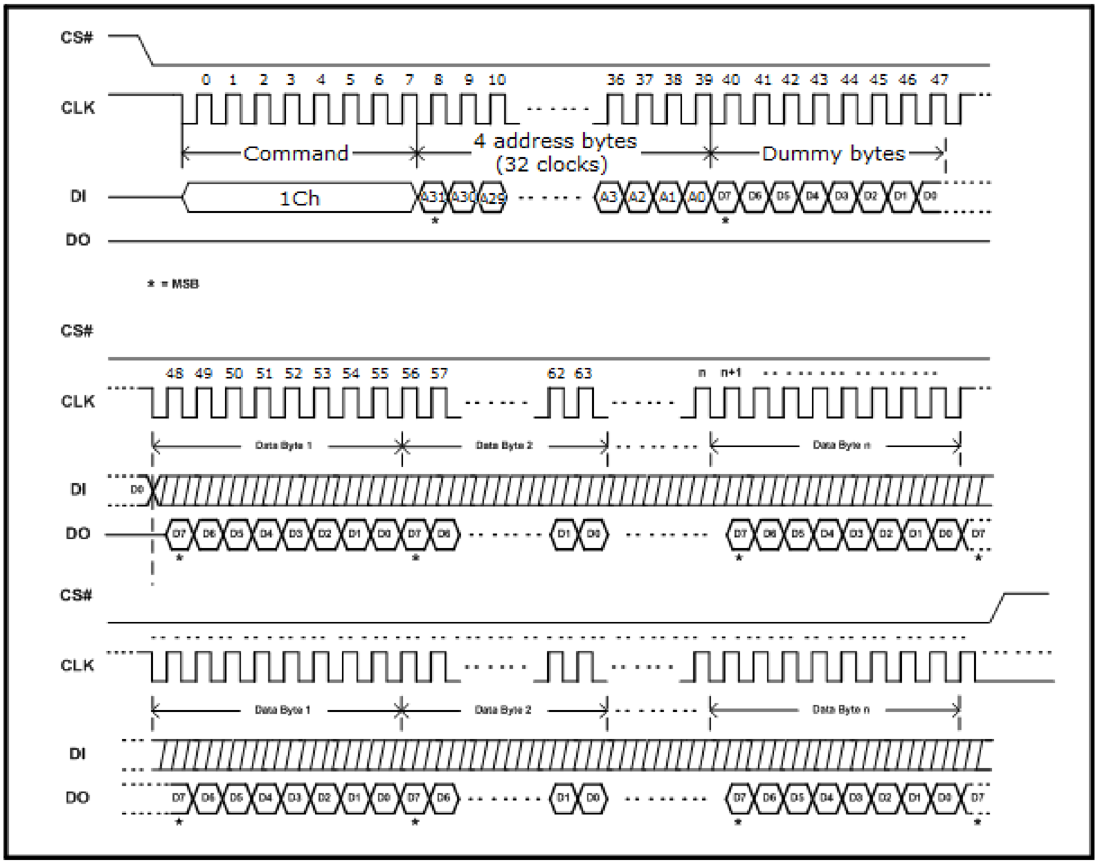
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Read Burst with 4byte address (1Ch)

This device supports Read Burst with 4byte address in SPI mode. This command is most likely Read Burst with wrap except the address byte count increase to four.

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.

Figure 29. Read Burst Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

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

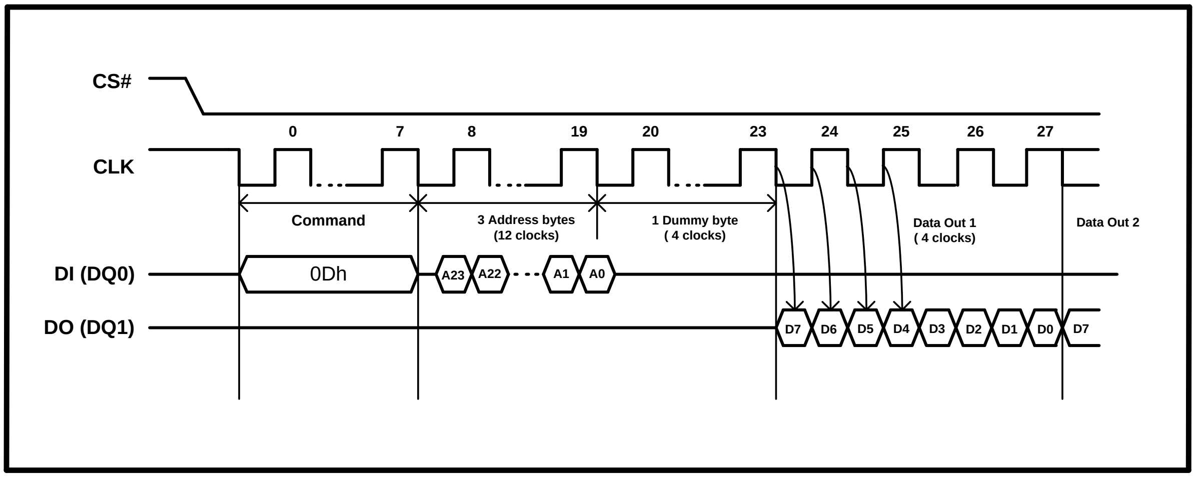
The DDR FAST_READ instruction (DDR Fast Read Instruction Sequence Diagram figure) is for doubling reading data out, signals are triggered on both rising and falling edge of clock. The address is latched on both rising and falling edge of CLK, and data of each bit shifts out on both rising and falling 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 (or 4-byte address, depends on address mode state) 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.

Figure 30. DDR Fast Read Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

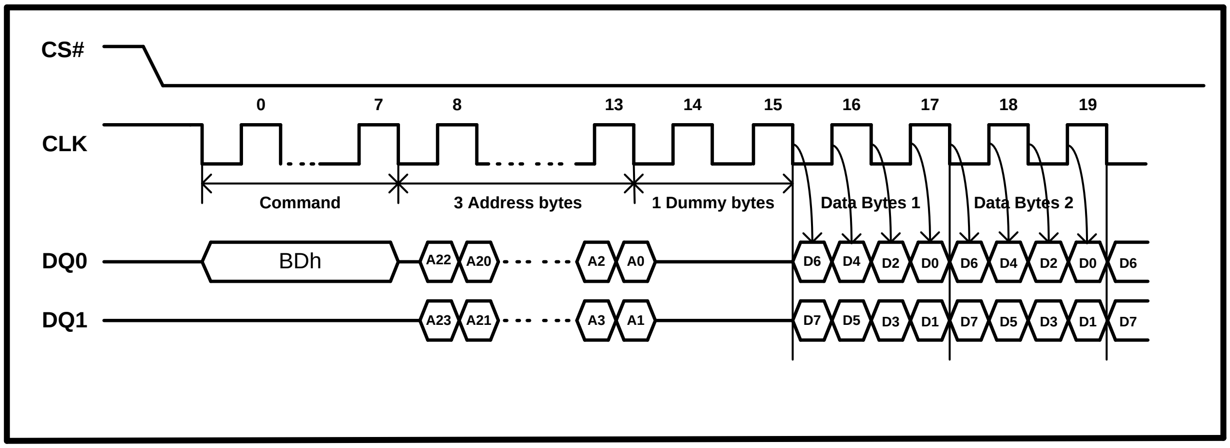
DDR Dual Input / Output FAST_READ (BDh)

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(or 32-bit address, depends on address mode state) 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 31. DDR Dual Input / Output FAST_READ Instruction Sequence Diagram



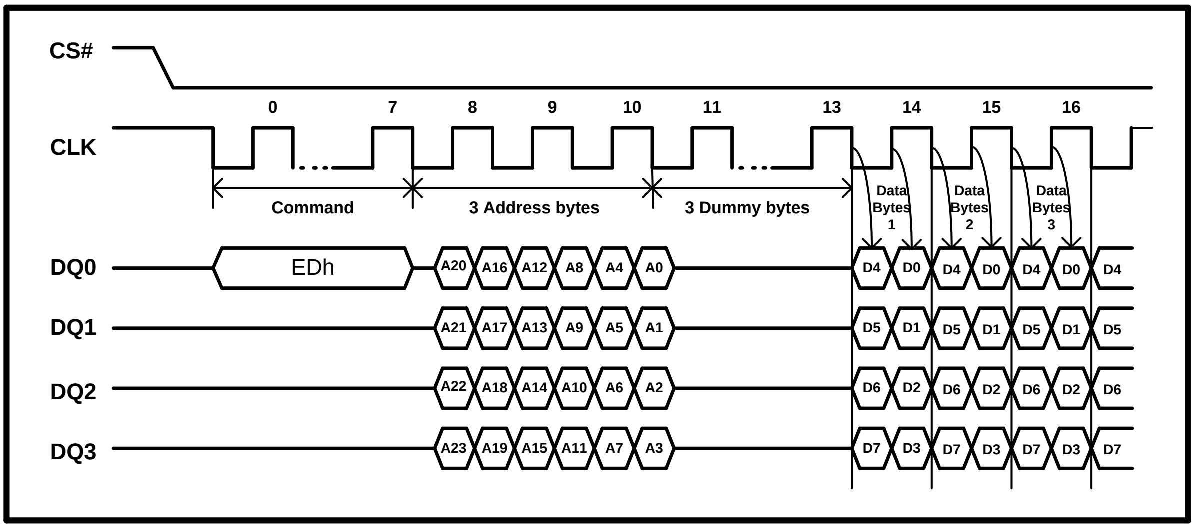
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

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 (or 32 bit address depends on address mode state) 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 32. DDR Quad Input / Output FAST_READ Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

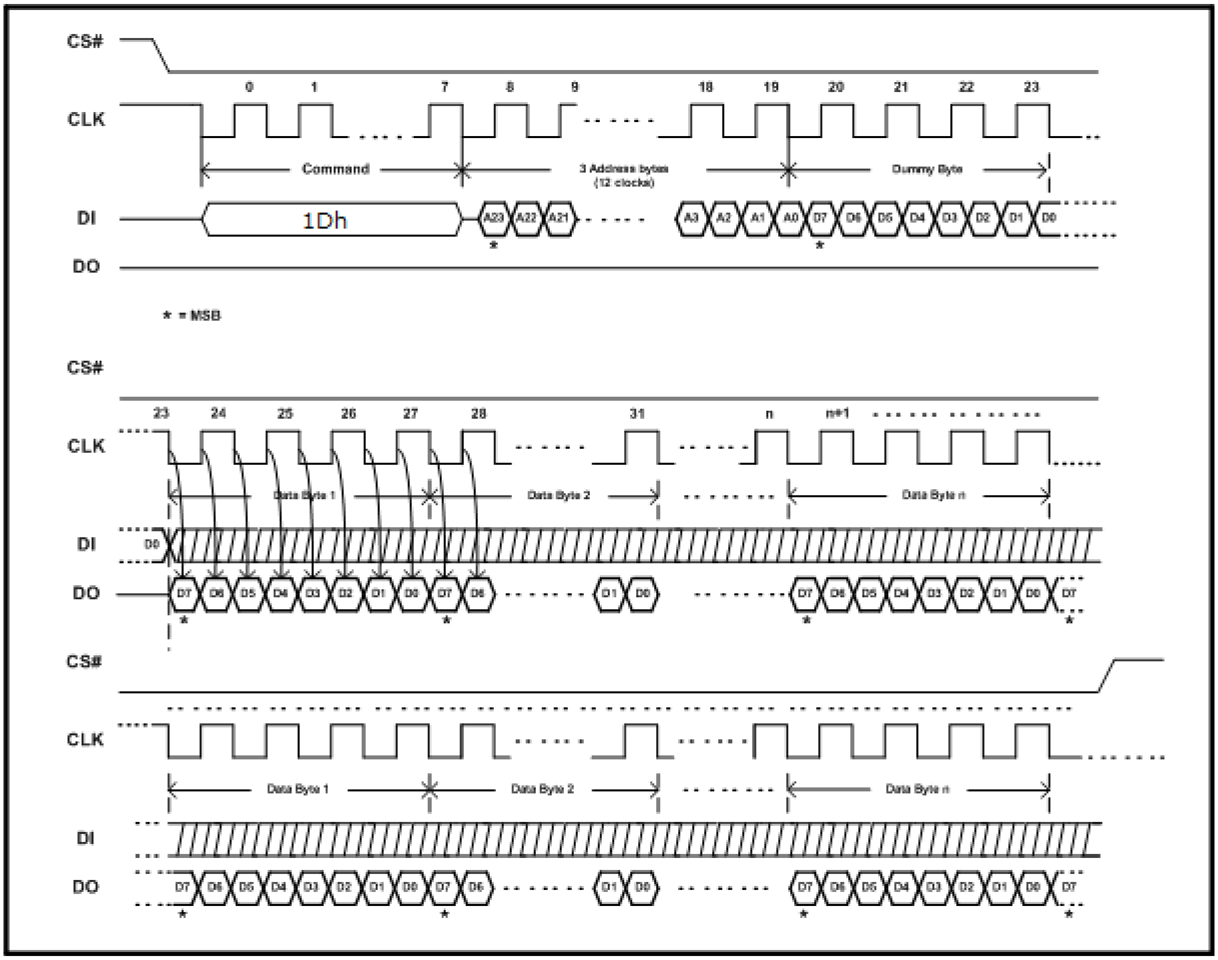
DDR Read Burst with Wrap (DQRB) (1Dh)

The DDR Read Burst with Wrap (1Dh) 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 (1Dh) 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 (1Dh) instruction is : CS# goes low -> sending DDR Read Burst with Wrap (1Dh) instruction (1-bit per clock) -> 24-bit address (or 32-bit depends on address mode state) interleave on DQ3, DQ2, DQ1 and DQ0 (8-bit per clock) -> 1 dummy bytes (4 clocks) -> data out interleave on DQ3, DQ2, DQ1 and DQ0 (8-bit per clock) -> to end DDR Read Burst with Wrap (1Dh) 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.

Figure 33. DDR Read Burst with Wrap with Wrap Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

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 (or four, depends on address mode state) 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.

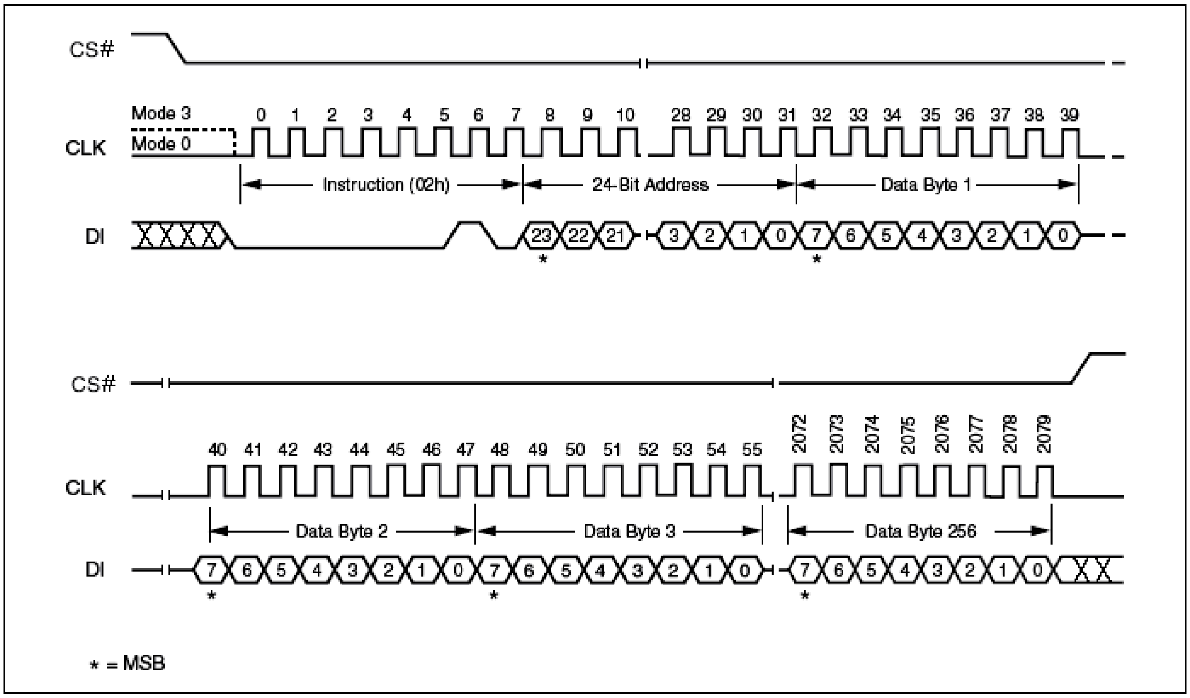
To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Page Program (02h).

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, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) is not executed.

Figure 34. Page Program Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Page Program with 4byte address (12h)

The Page Program with 4byte address 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 with 4byte address instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, four 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.

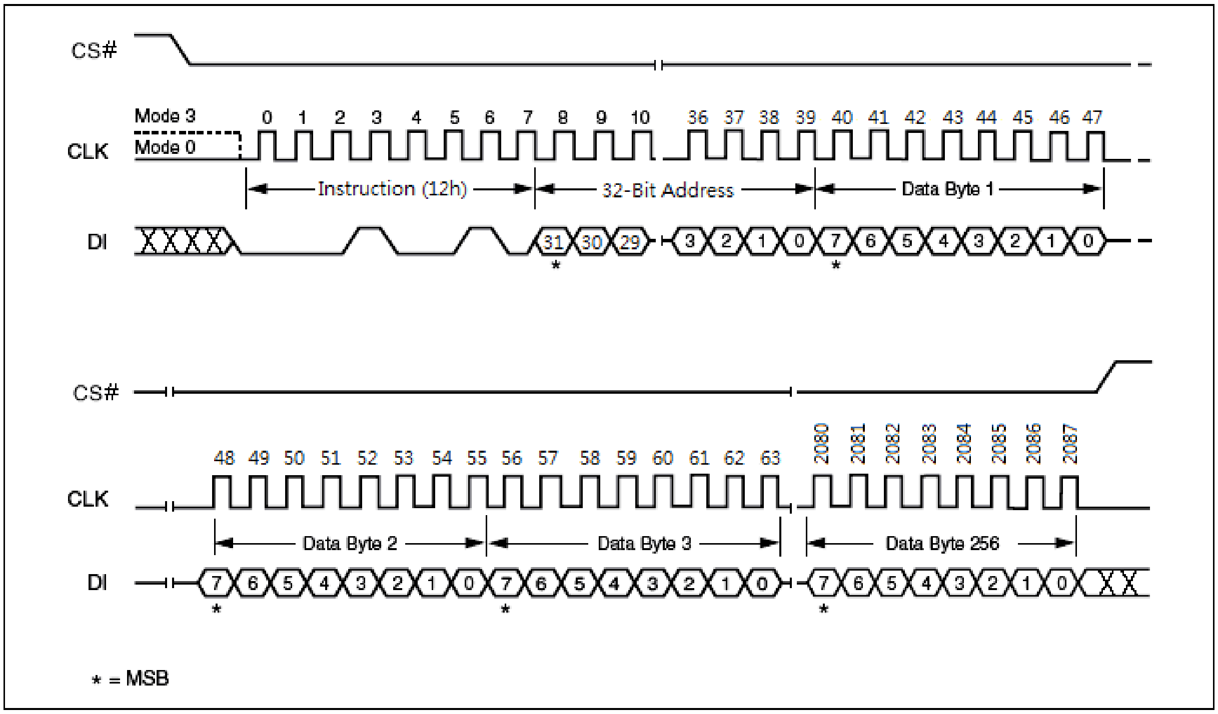
The instruction sequence is shown in Page Program with 4byte address Instruction Sequence Diagram figure. If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 Data 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 with 4byte address instruction is not executed.

As soon as Chip Select (CS#) is driven high, the self-timed Page Program with 4byte address cycle (whose duration is t_{PP}) is initiated. While the Page Program with 4byte address 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 with 4byte address 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 with 4byte address instruction applied to a page which is protected by the Block Protect (CMP, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) is not executed.

Figure 35. Page Program with 4byte address Instruction Sequence Diagram



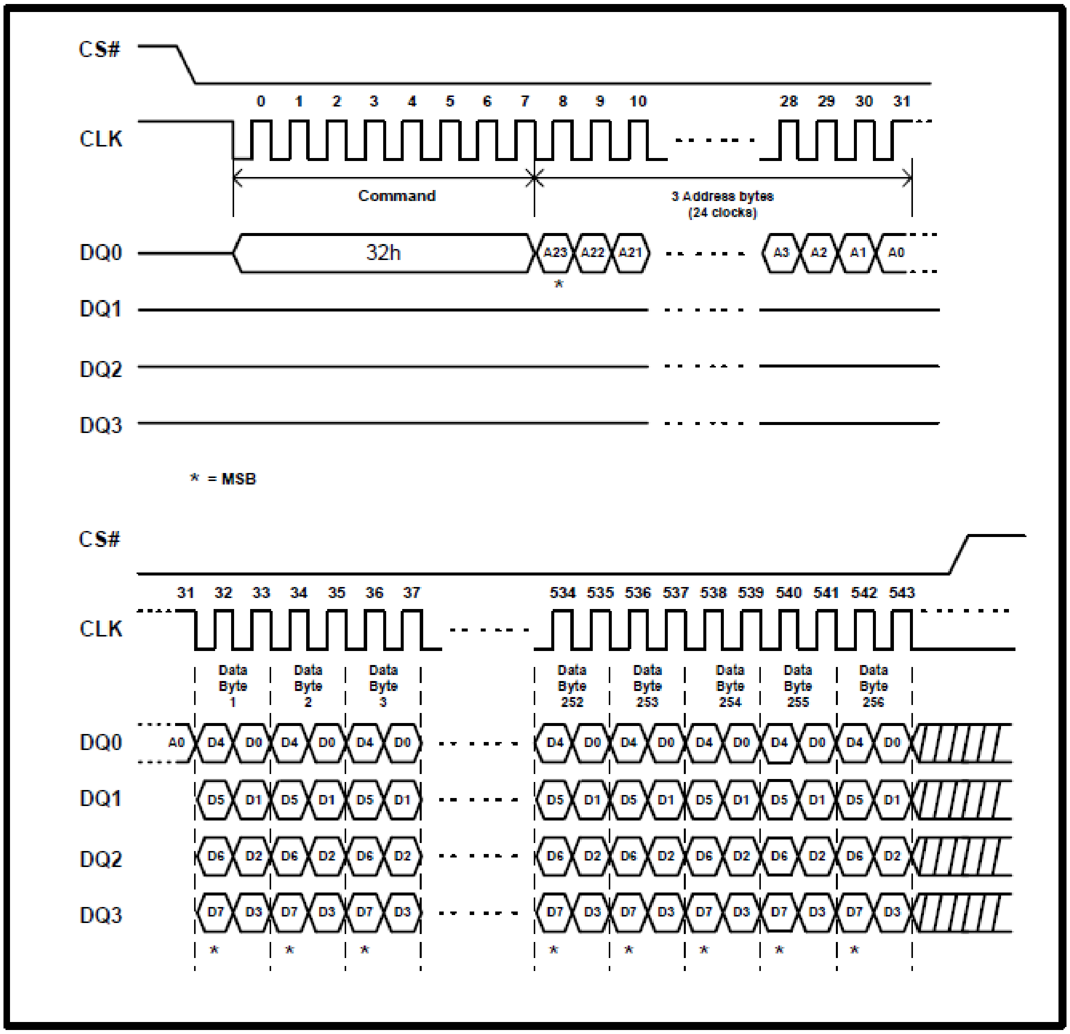
Note: This instruction is workable while in 4byte address mode.

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 access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Quad Input Page Program (32h).

To use Quad Page Program (QPP) the WP#, HOLD#, RESET# Disable (QE) bits in Status Register 2 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) (or 32-bit address (A31-0), depends on address mode state) 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 36. Quad Input Page Program Instruction Sequence Diagram (SPI Mode only)



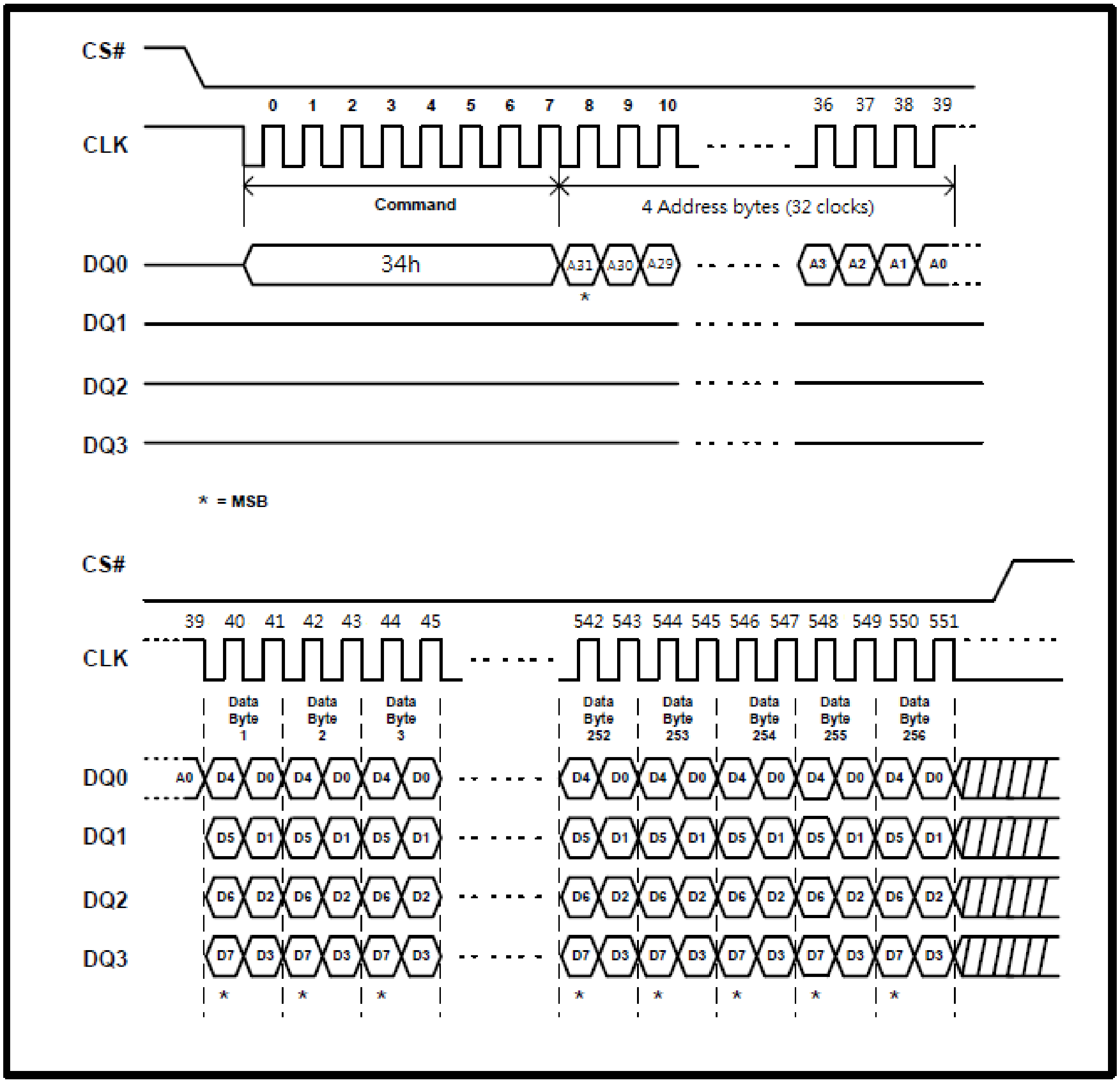
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Quad Input Page Program with 4byte address (34h)

The Quad Input Page Program with 4byte address 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 Input Page Program with 4byte address 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 Input Page Program with 4byte address instruction since the inherent page program time is much greater than the time it take to clock- in the data.

To use Quad Input Page Program with 4byte address the WP#, HOLD# and RESET# Disable (QE) bits in Status Register 2 must be set to 1. A Write Enable instruction must be executed before the device will accept the Quad Input Page Program with 4byte address instruction (SR.1, WEL=1). The instruction is initiated by driving the CS# pin low then shifting the instruction code “34h” followed by a 32-bit address (A31-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 Input Page Program with 4byte address are identical to standard Page Program with 4byte address. The Quad Input Page Program with 4byte address instruction sequence is shown in Quad Input Page Program with 4byte address Instruction Sequence Diagram (SPI Mode only) figure.

Figure 37. Quad Input Page Program with 4byte address Instruction Sequence Diagram (SPI Mode only)



Note: This instruction is workable while in 4byte address mode.

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(or 32-bit, depends on address mode state) interleave on DQ3, DQ2, DQ1 and DQ0 (2-bit per clock) -> data in interleave on DQ3, DQ2, DQ1 and 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.

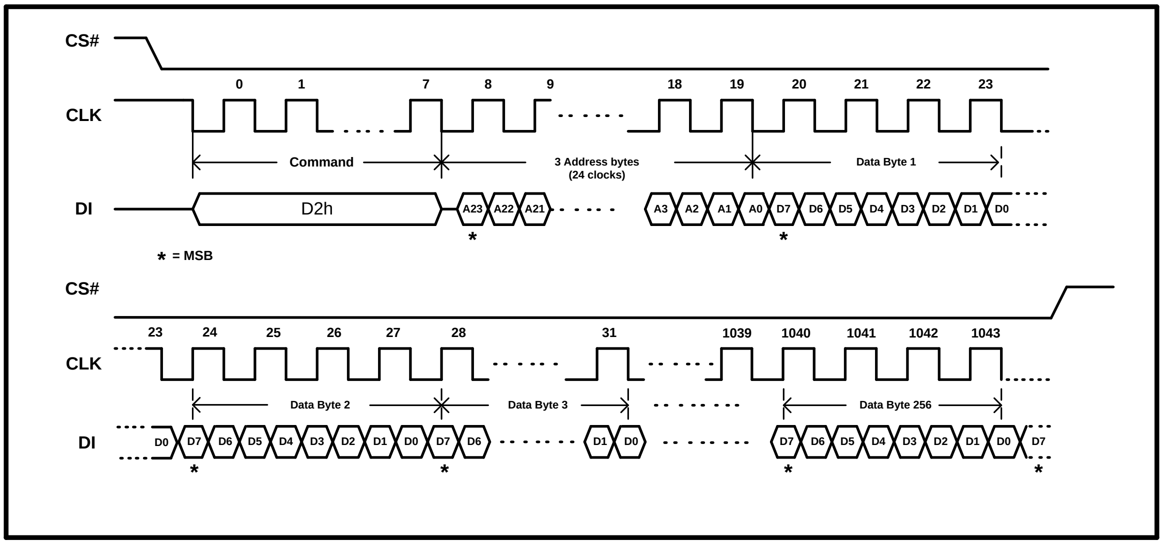
The instruction sequence is shown in DDR Page Program Instruction Sequence Diagram figure. If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 Data bytes are sent 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.

Figure 38. DDR Page Program Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Write Suspend (B0h/75h)

Write Suspend allows the interruption of Sector Erase, Block Erase or Page Program operations in order to erase, program, or 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.

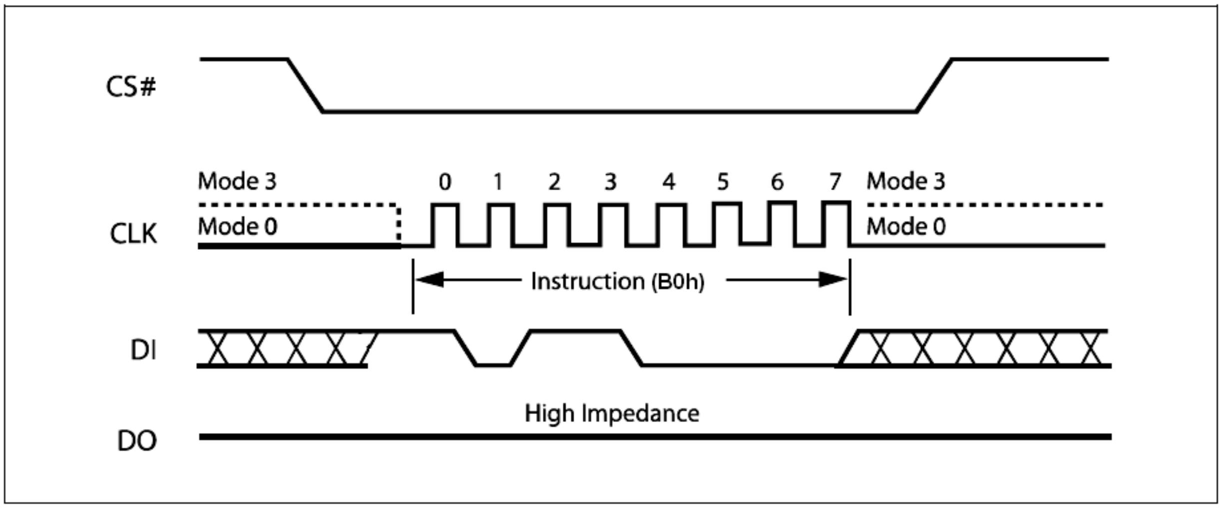
Only one write operation can be suspended at a time; if an operation is already suspended, the device will ignore the Write Suspend command. Write Suspend during Chip Erase is ignored; Chip Erase is not a valid command while a write is suspended.

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 39. 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 sector(s) will output unknown data because the Sector or Block Erase will be incomplete.

To execute a Write Suspend operation, the host drives CS# low, sends the Write Suspend command cycle (B0h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. The Suspend Status register indicates that the erase has been suspended by changing the WSE bit from “0” to “1”, but the device will not accept another command until it is ready. To determine when the device will accept a new command, poll the WIP bit in the Suspend Status register or after issue program suspend command, latency time 20us is needed before issue another command. For “Suspend to Read”, “Resume to Read”, “Resume to Suspend” timing specification please note 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 commands pointing to the suspended 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 Suspend Status register indicates that the programming has been suspended by changing the WSP bit from "0" to "1", but the device will not accept another command until it is ready. To determine when the device will accept a new command, poll the WIP bit in the Suspend Status register or after issue program suspend command, latency time 20us 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.

Figure 40. Suspend to Read Latency

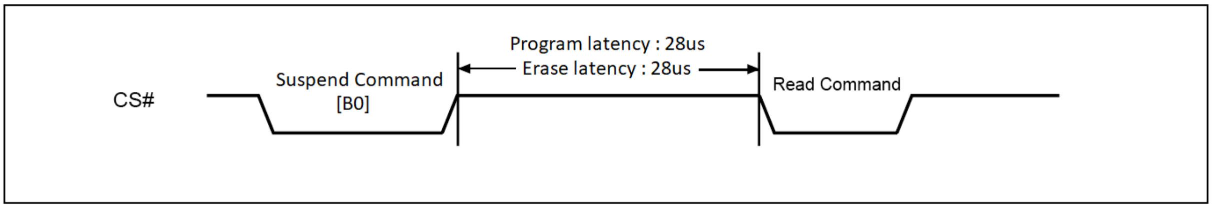


Figure 41. Resume to Read Latency

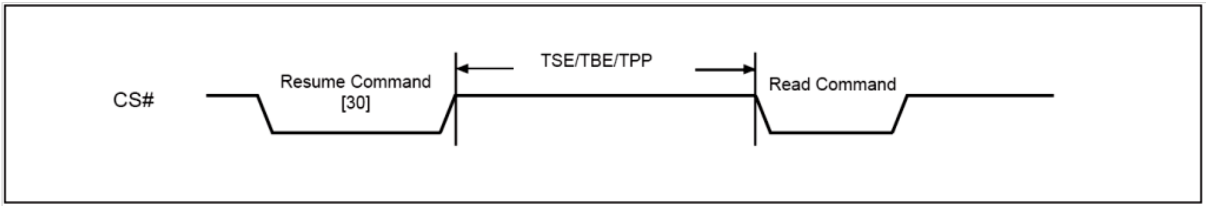
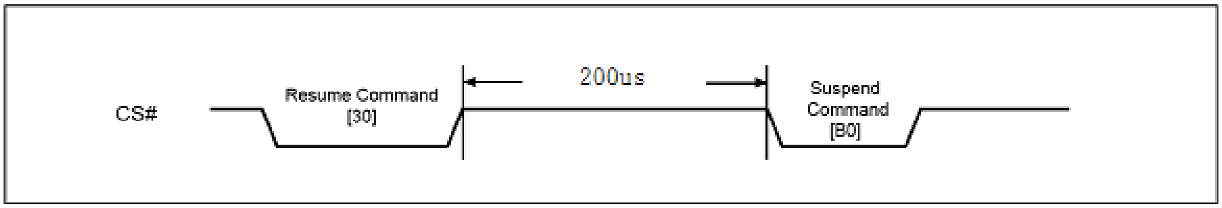


Figure 42. Resume to Suspend Latency



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), 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 Suspend Status register, or wait the specified time t_{SE}, t_{HBE}, t_{BE} or t_{PP} for Sector 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 200us.

Figure 43. Write Resume Instruction Sequence Diagram

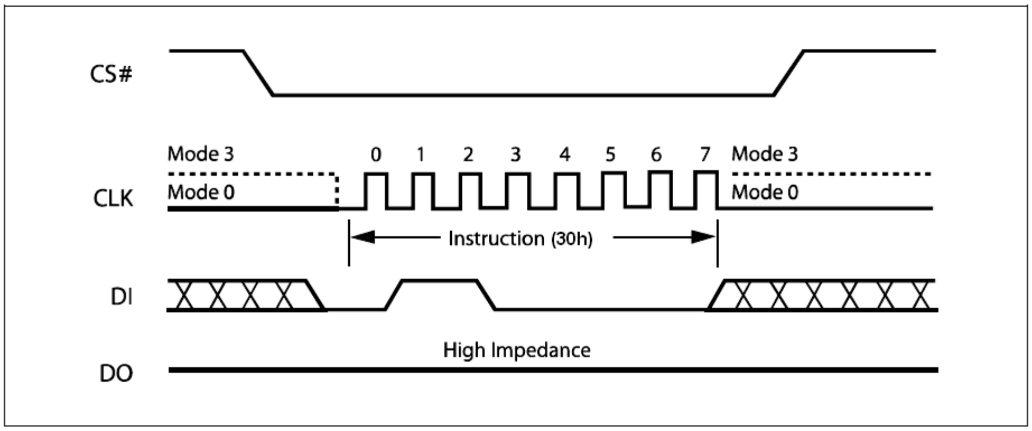
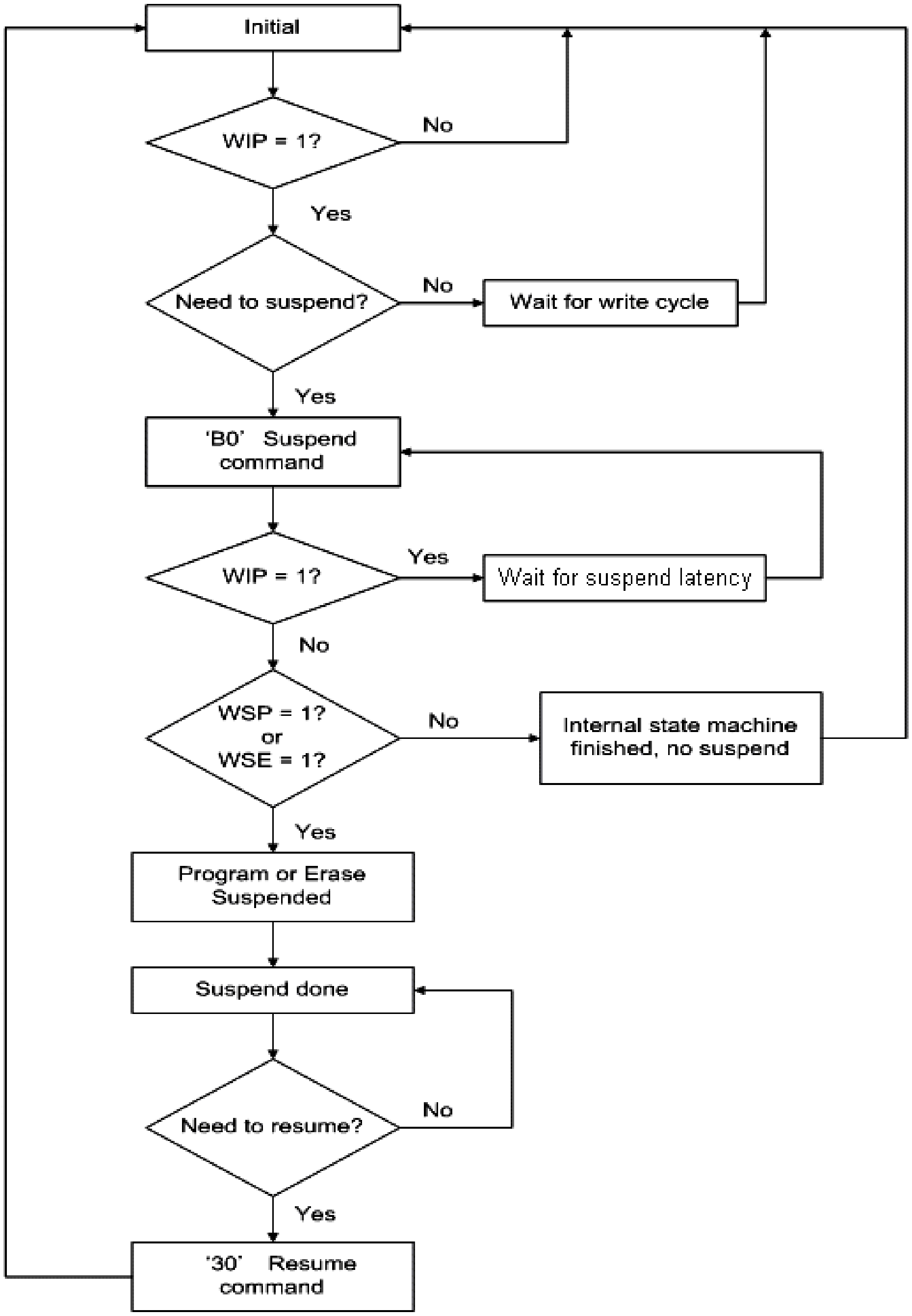


Figure 44. Write Suspend/Resume Flow



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 20 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)

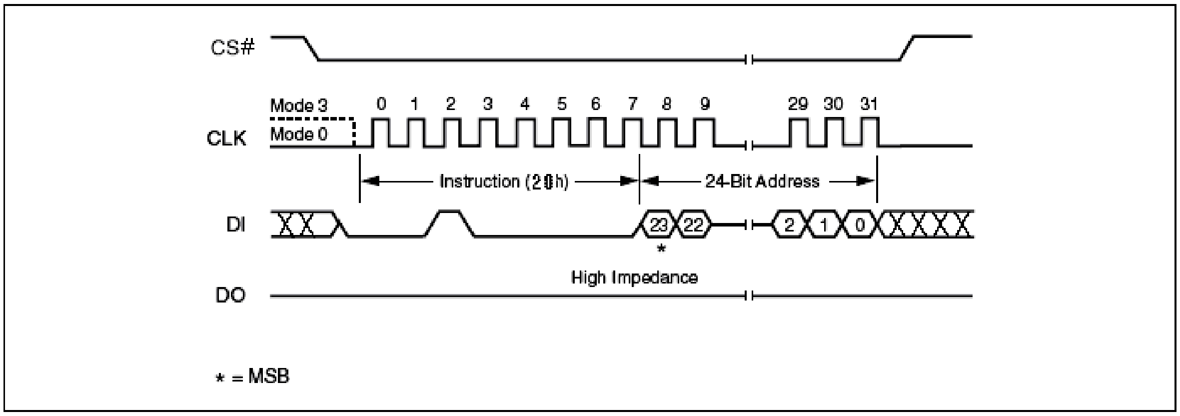
The Sector Erase (SE) instruction sets to 1 (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Sector Erase (SE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three(or four, depends on address modes state) 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before Sector Erase (20h).

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, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) will be ignored.

Figure 45. Sector Erase Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

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

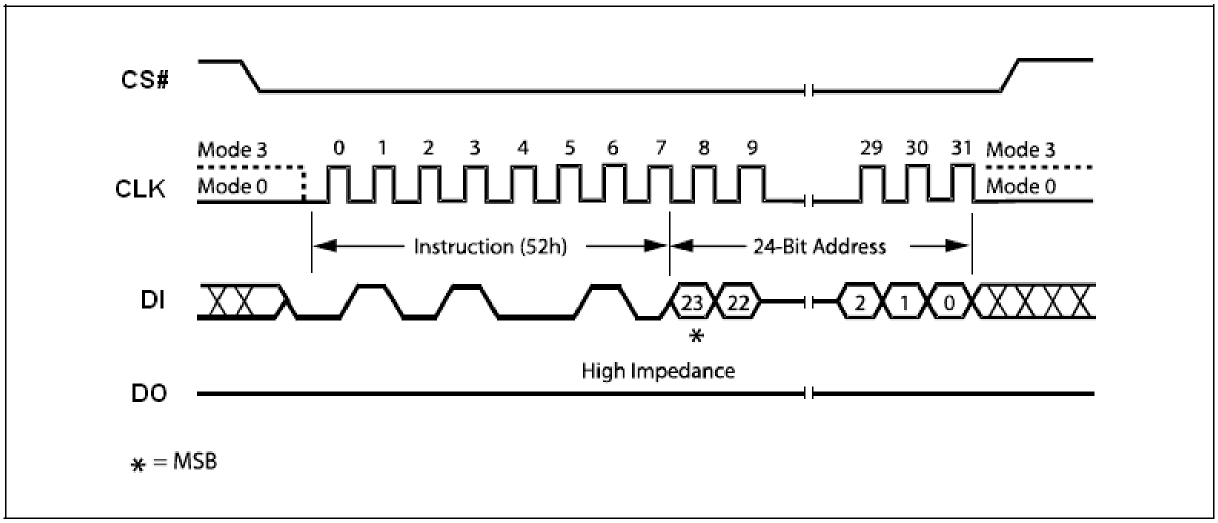
The Half Block Erase (HBE) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Half Block Erase (HBE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three (or four, depends on address mode state) 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before 32KB Half Block Erase (52h).

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, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) will be ignored.

Figure 46. 32KB Half Block Erase Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

64KB Block Erase (BE) (D8h)

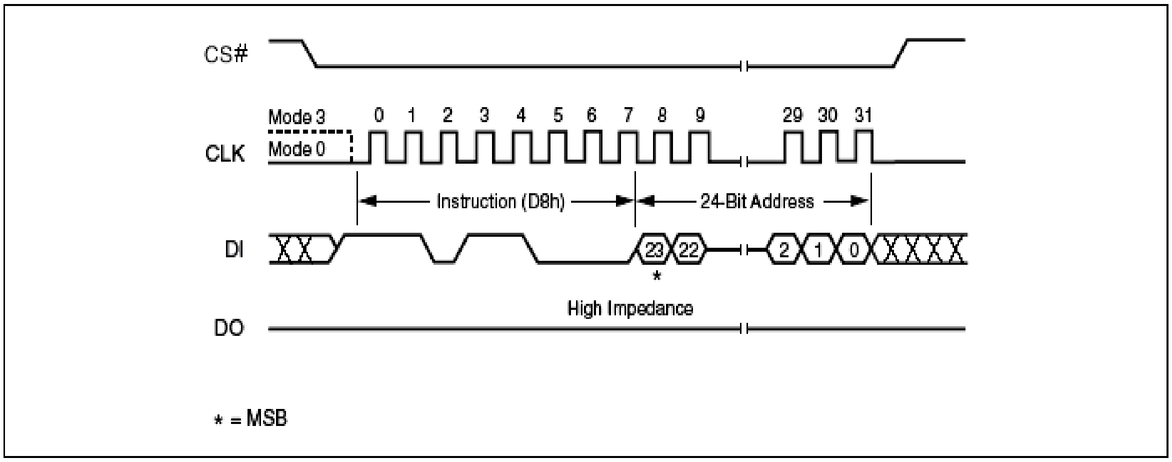
The Block Erase (BE) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Block Erase (BE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three (or four, depends on address mode state) 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. To access higher address (larger than 128Mb) in 3 byte address mode, user can issue Write Extended Register (C5h) operation before 64KB Block Erase (D8h).

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, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) is not executed.

Figure 47. 64KB Block Erase Instruction Sequence Diagram



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Sector Erase with 4byte address (SE4B) (21h)

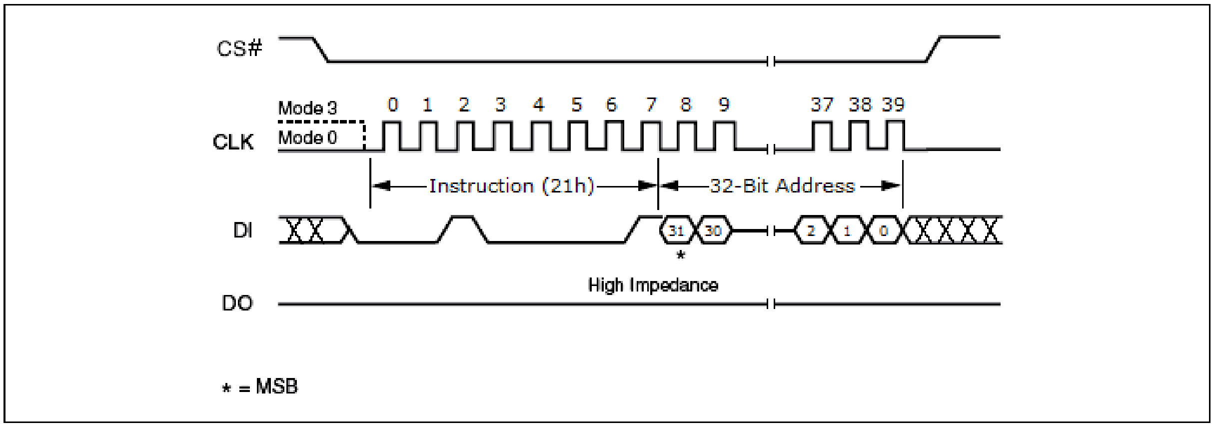
The Sector Erase with 4byte address (SE4B) instruction sets to 1 (FFh) all bits inside the chosen sector. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Sector Erase with 4byte address (SE4B) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and four 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 with 4byte address (SE4B) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in Sector Erase with 4byte address 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 with 4byte address (SE4B) 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 with 4byte address (SE4B) instruction applied to a sector which is protected by the Block Protect (CMP, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) will be ignored.

Figure 48. Sector Erase with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

32KB Half Block Erase with 4byte address (HBE4B) (5Ch)

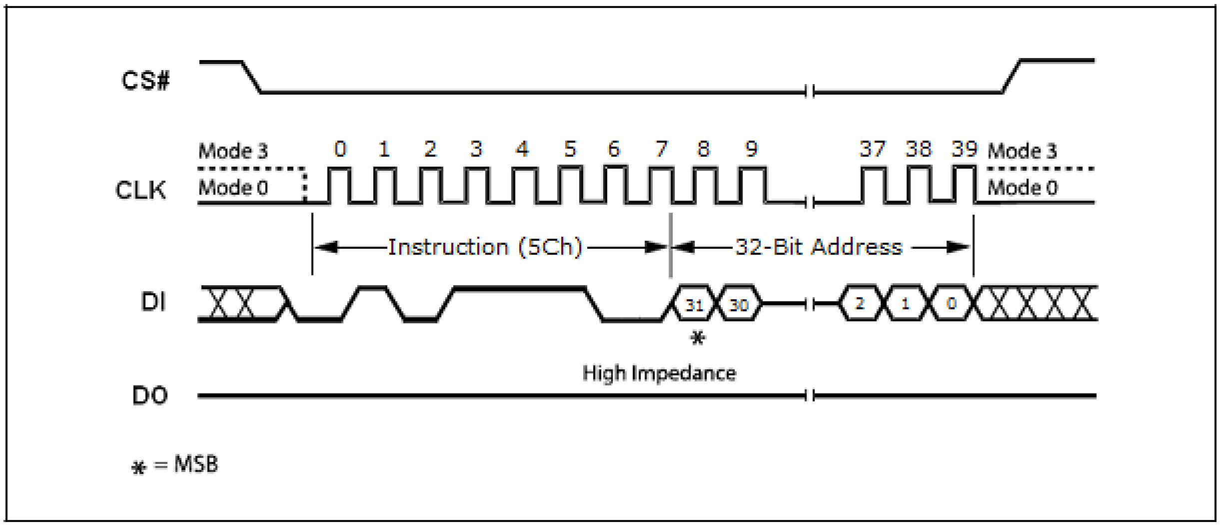
The Half Block Erase with 4byte address (HBE4B) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Half Block Erase with 4byte address (HBE4B) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and four 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 with 4byte address (HBE4B) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in 2KB Half Block Erase with 4byte address 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 with 4byte address (HBE4B) 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 with 4byte address (HBE4B) instruction applied to a block which is protected by the Block Protect (CMP, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) will be ignored.

Figure 49. 32KB Half Block Erase with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

64KB Block Erase with 4byte address (BE4B) (DCh)

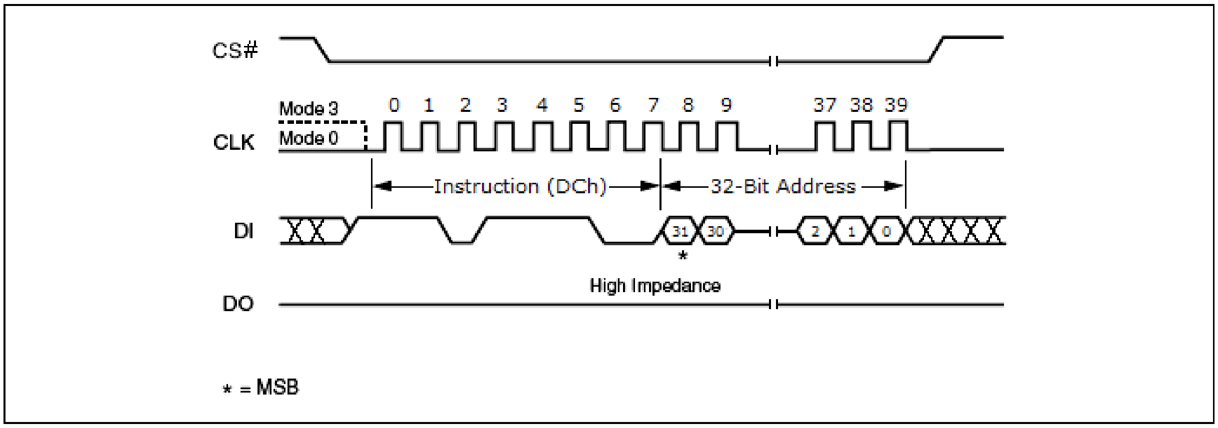
The Block Erase with 4byte address (BE4B) instruction sets to 1 (FFh) all bits inside the chosen block. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

The Block Erase with 4byte address (BE4B) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and four 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 with 4byte address (BE4B) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in 64KB Block Erase with 4byte address 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 with 4byte address (BE4B) 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 with 4byte address (BE4B) instruction applied to a block which is protected by the Block Protect (CMP, TB, BP3, BP2, BP1, BP0) bits (see Protected Area Sizes Sector Organization table) is not executed.

Figure 50. 64KB Block Erase with 4byte address Instruction Sequence Diagram



Note: This instruction is workable while in 4byte address mode.

Chip Erase (CE) (C7h/60h)

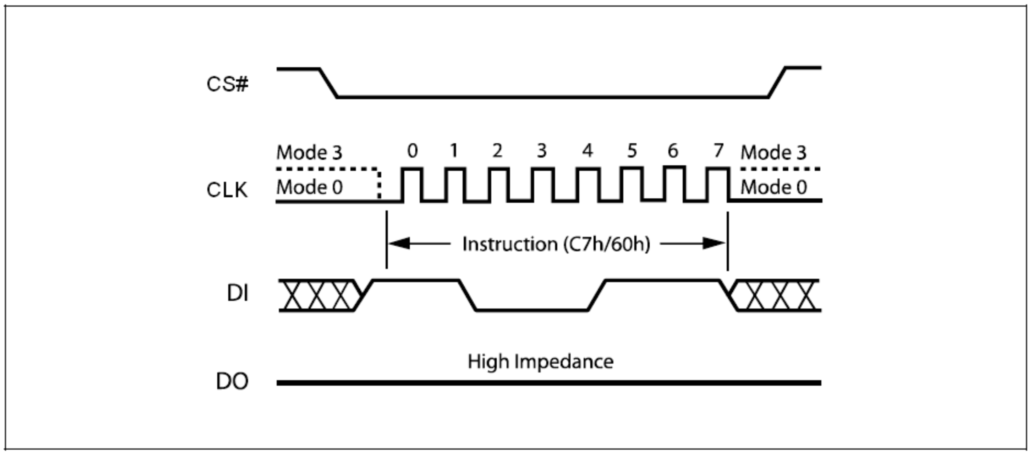
The Chip Erase (CE) instruction sets all bits to 1 (FFh). Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed. After the Write Enable (WREN) instruction has been decoded, the device sets the Write Enable Latch (WEL).

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.

Figure 51. Chip Erase Instruction Sequence Diagram

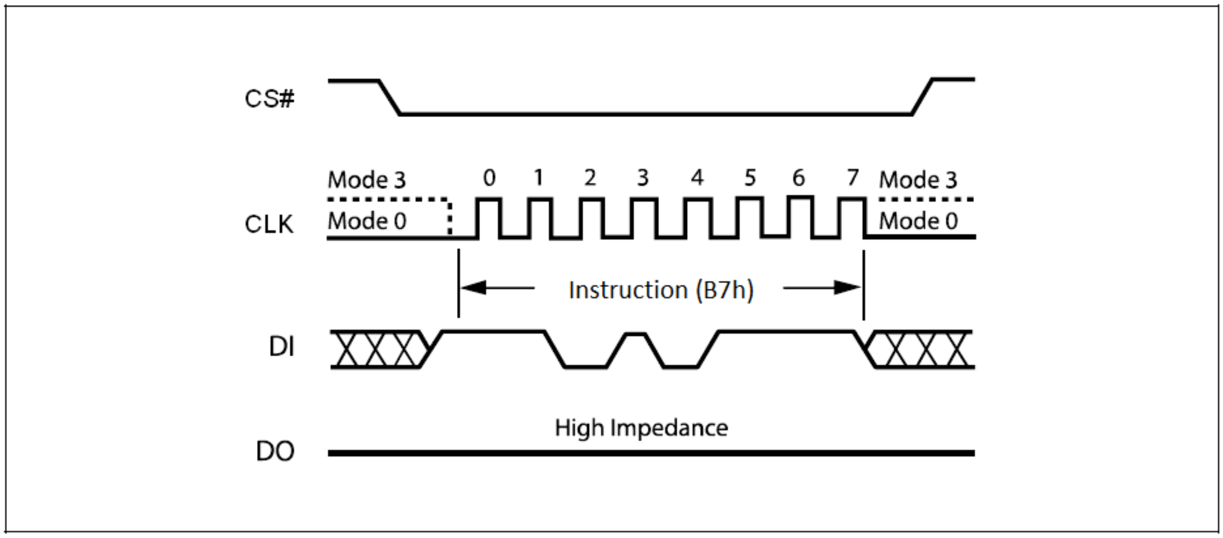


Enter 4-Byte Address Modes (B7h)

The enter 4byte address mode instruction enables accessing the address length of 32-bit for the memory area of higher density (larger than 128Mb). The device default is in 24-bit address mode; after sending out the enter 4byte address mode instruction, the 4byte bit of Status Register 3 will be automatically set to "1" to indicate the 4-byte address mode has been enabled. Once the 4-byte address mode is enabled, the address length becomes 32-bit instead of the default 24-bit. There is a method to exit the 4-byte mode: writing exit 4-byte mode instruction. All instructions are accepted normally, and just the address bit is changed from 24-bit to 32-bit.

The sequence of issuing enter 4byte address mode instruction is: CS# goes low → sending enter 4byte address mode instruction to enter 4-byte mode(automatically set 4byte bit as "1") → CS# goes high. (Enter 4byte address mode Instruction Sequence Diagram figure for SPI mode)

Figure 52. Enter 4byte address mode Instruction Sequence Diagram

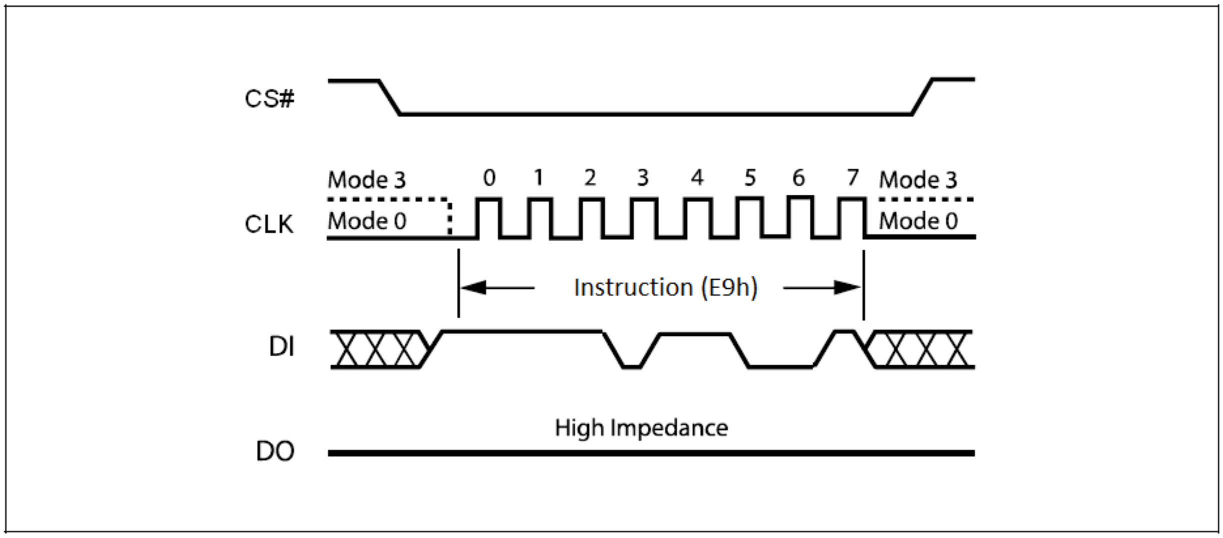


Exit 4-Byte Address Modes (E9h)

The Exit 4byte address mode instruction is executed to exit the 4-byte address mode and return to the default 3-bytes address mode. After sending out the exit 4byte address mode instruction, the 4byte bit of Status Register 3 will be cleared to be "0" to indicate the exit of the 4-byte address mode. Once exiting the 4-byte address mode, the address length will return to 24-bit.

The sequence of issuing Exit 4-Byte Address Modes instruction is: CS# goes low → sending exit 4byte address mode instruction to exit 4-byte mode (automatically clear the 4BYTE bit to be "0") → CS# goes high. (Exit 4byte address mode Instruction Sequence Diagram figure for SPI mode)

Figure 53. Exit 4byte address mode Instruction Sequence Diagram



Read Extended Address Register (C8h)

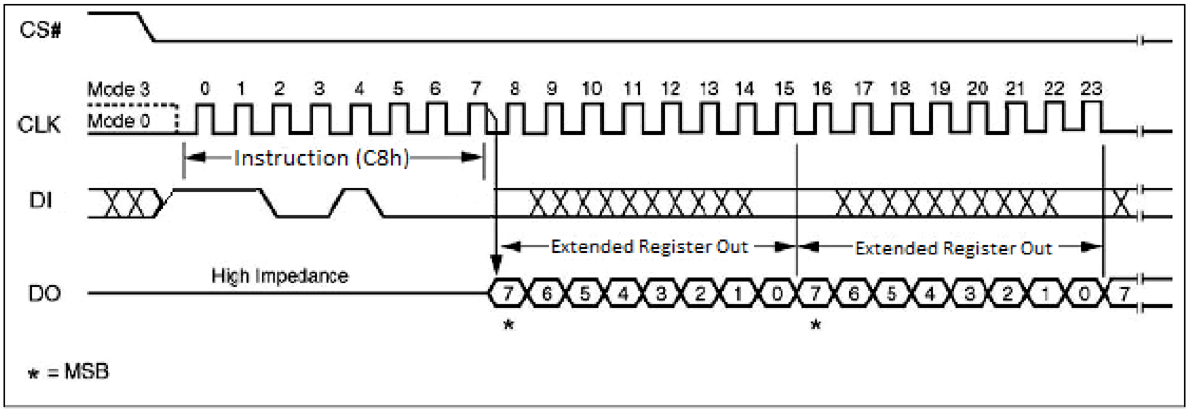
When the device is in the 3-Byte Address Mode, the Extended Address Register is used as the 4th address byte A[31:24] to access memory regions beyond 128Mb. The Read Extended Address Register instruction is entered by driving CS# low and shifting the instruction code “C8h” into the DI pin on the rising edge of CLK. The Extended Address Register bits are then shifted out on the DO pin at the falling edge of CLK with most significant bit (MSB) first as shown in Read Extended Address Register Instruction Sequence Diagram figure.

When the device is in the 4byte address Mode, the Extended Address Register is not used.

Table 15. Extended Register Bit Locations

ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
A31	A30	A29	A28	A27	A26	A25	A24
Volatile bit	Volatile bit	Volatile bit	Volatile bit	Volatile bit	Volatile bit	Volatile bit	Volatile bit

Figure 54. Read Extended Address Register Instruction Sequence Diagram



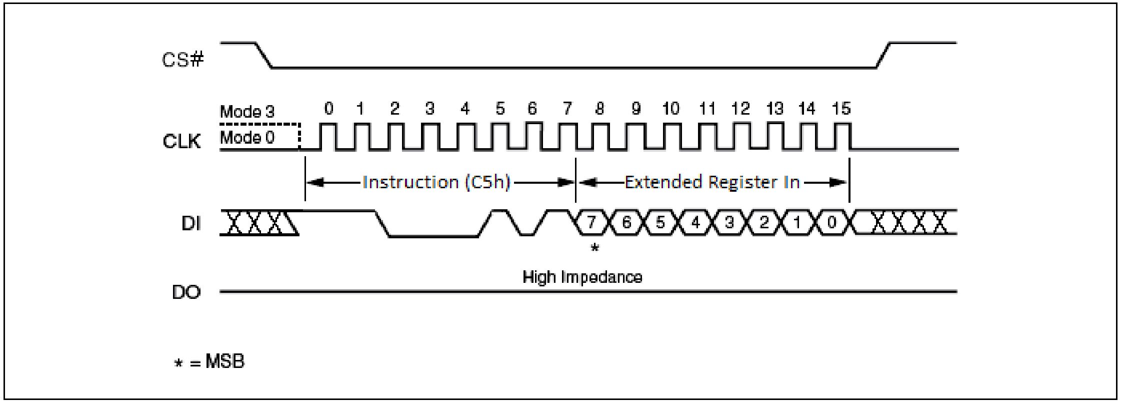
Write Extended Address Register (C5h)

The Extended Address Register is a volatile register that stores the 4th byte address (A31-A24) when the device is operating in the 3-Byte Address Mode. To write the Extended Address Register bits, a Write Enable (06h) instruction must previously have been executed for the device to accept the Write Extended Address Register instruction (Status Register bit WEL must equal 1). Once write enabled, the instruction is entered by driving CS# low, sending the instruction code “C5h”, and then writing the Extended Address Register data byte as illustrated in Write Extended Register Instruction Sequence Diagram figure.

Upon power up or the execution of a Software or Hardware Reset, the Extended Address Register bit values will be cleared to 0.

The Extended Address Register is only effective when the device is in the 3-Byte Address Mode. When the device operates in the 4byte address mode, any command with address input of A31-A24 will replace the Extended Address Register values. It is recommended to check and update the Extended Address Register if necessary when the device is switched from 4byte to 3byte address mode.

Figure 55. Write Extended Register Instruction Sequence Diagram



Deep Power-down (DP) (B9h)

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

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

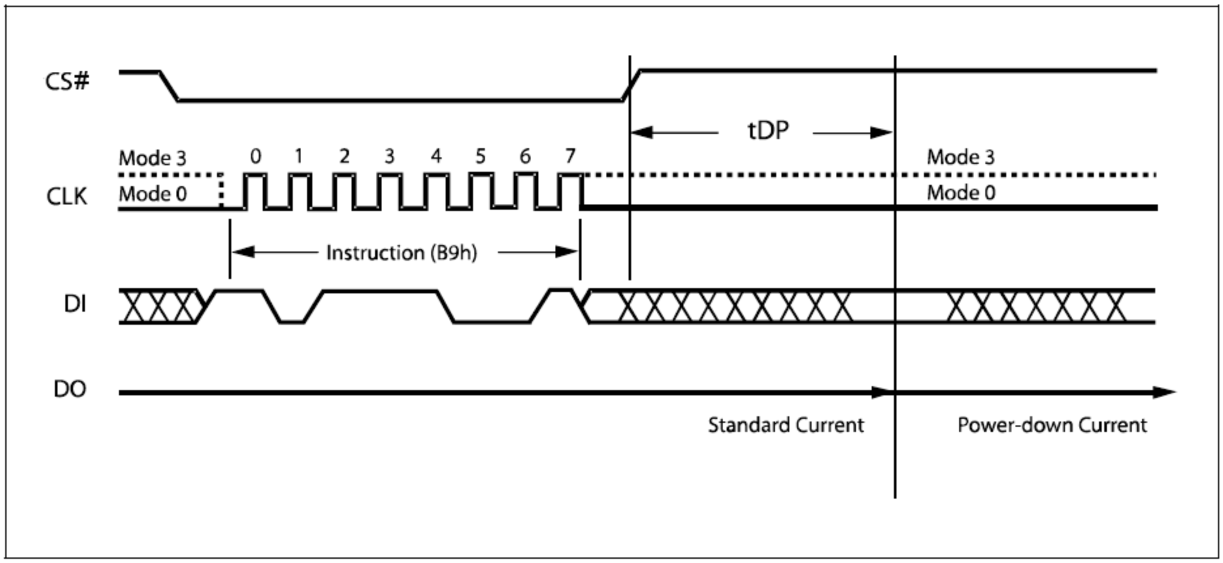
Once the device has entered the Deep Power-down mode, all instructions are ignored except the Release from Deep Power-down, Read Device ID (RDI) and Software Reset instruction which release the device from this mode. The Release from Deep Power-down and Read Device ID (RDI) instruction also allows the 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 56. Deep Power-down Instruction Sequence Diagram



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

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 57. Release Power-down Instruction Sequence Diagram

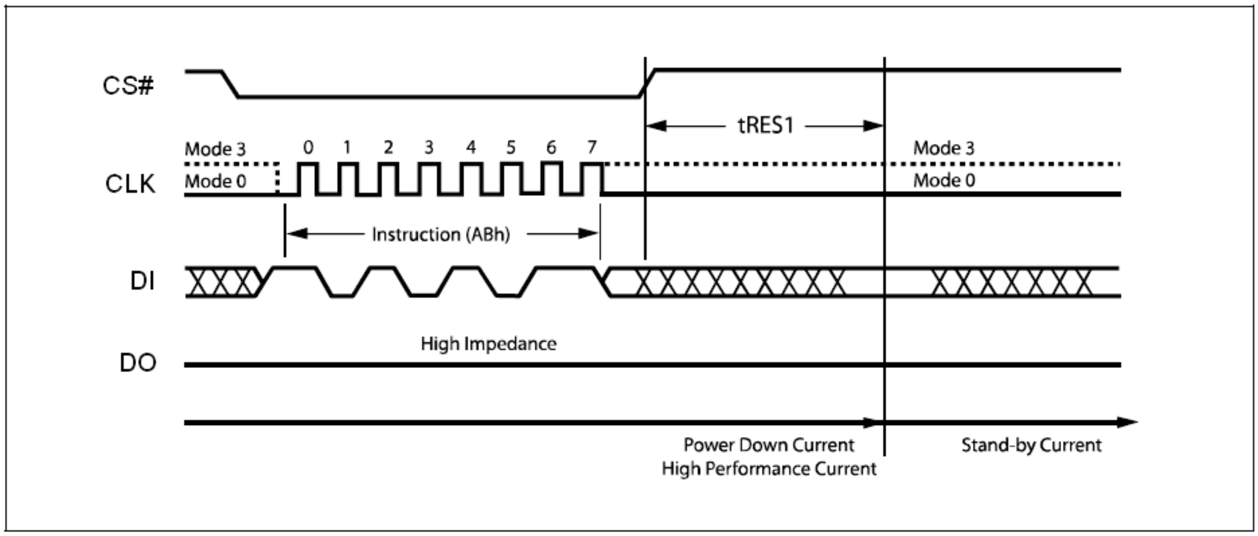
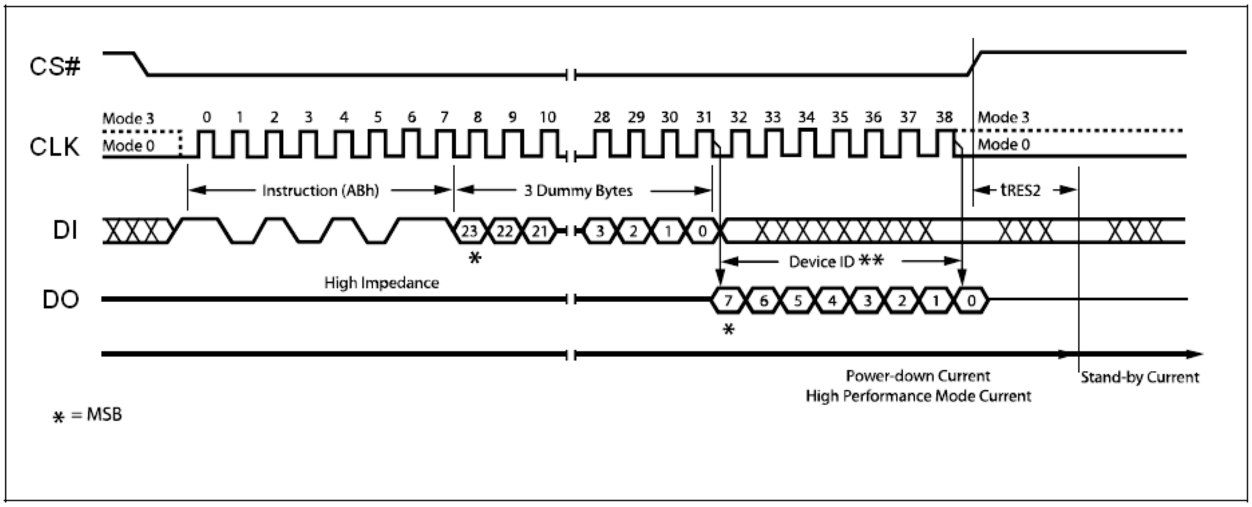


Figure 58. Release Power-down / Device ID Instruction Sequence Diagram

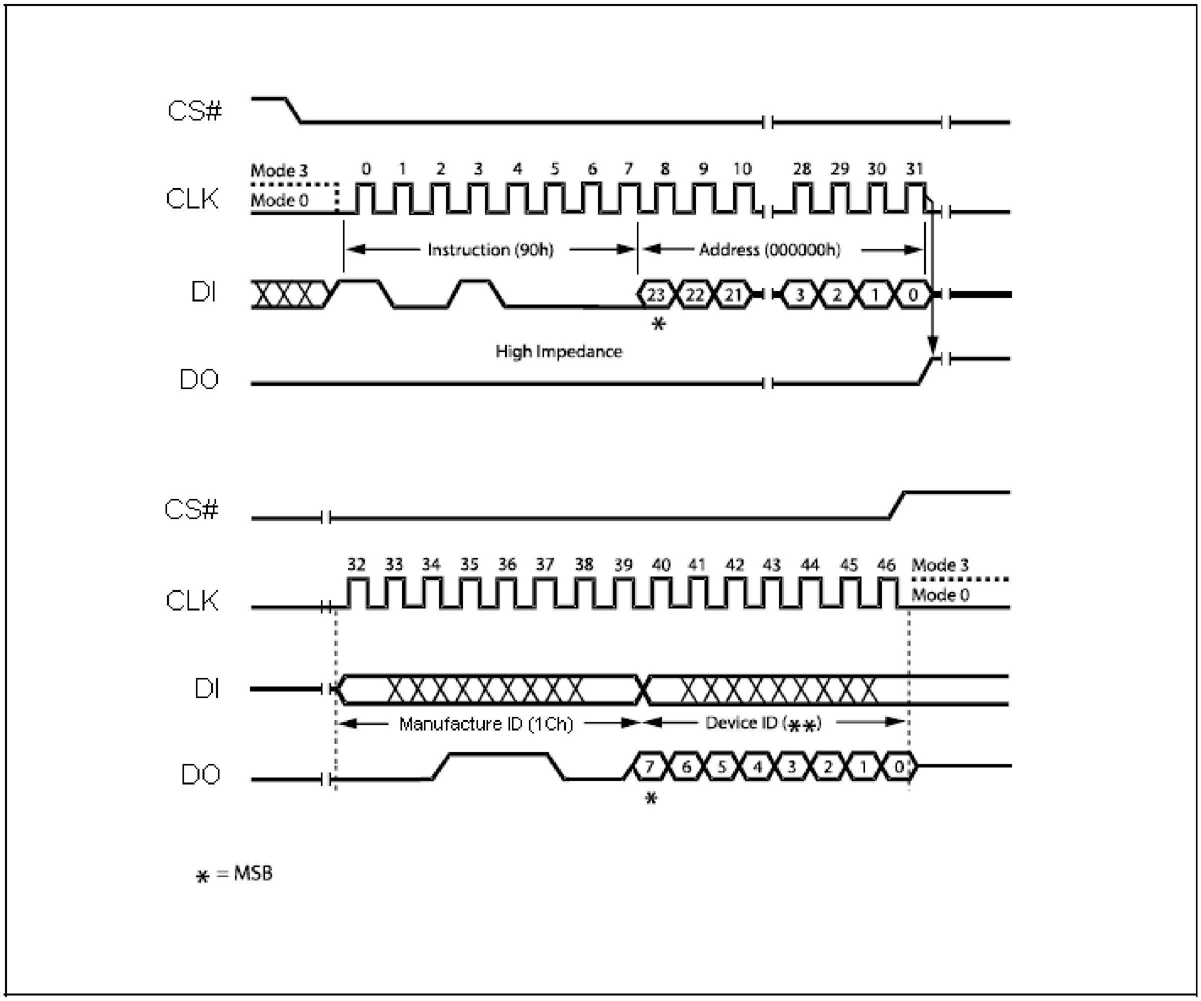


Read Manufacturer / Device ID (90h)

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.

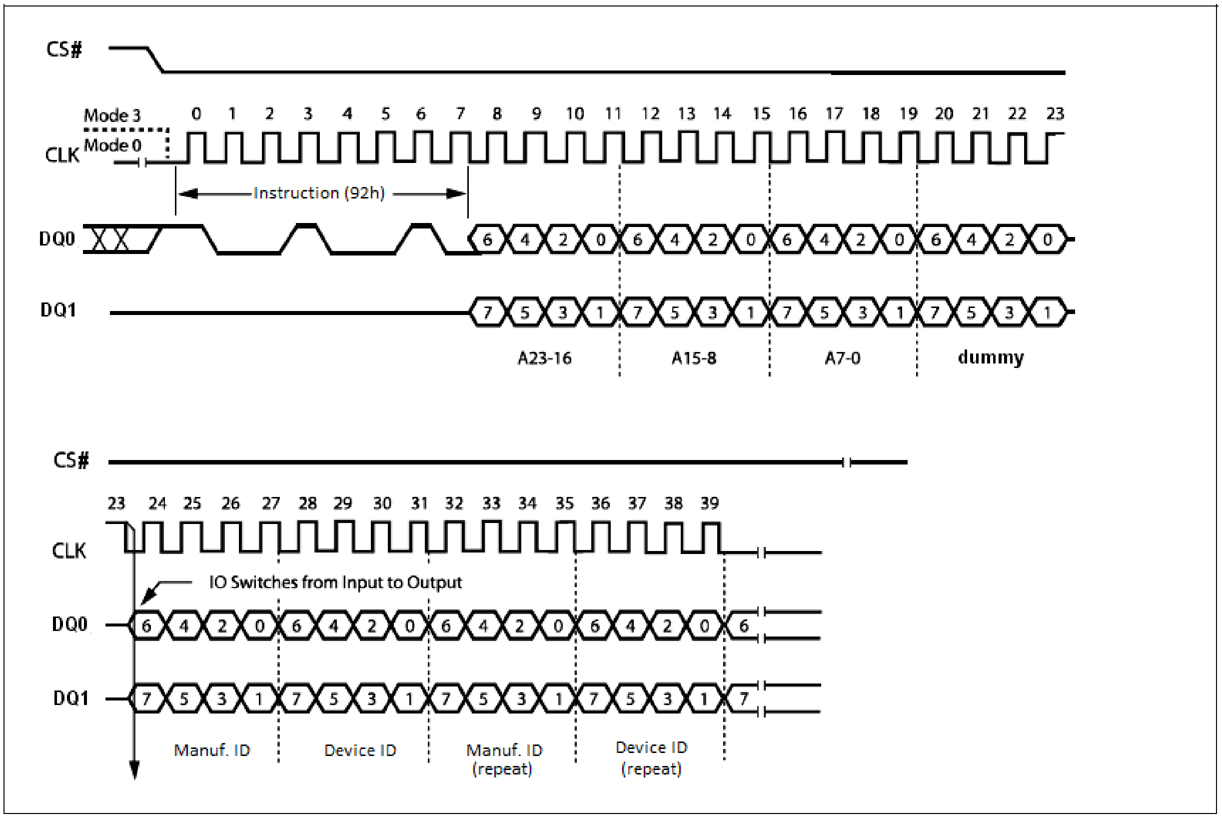
Figure 59. Read Manufacturer / Device ID Diagram



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 DQ_0 and DQ_1 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 DQ_0 and DQ_1 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.

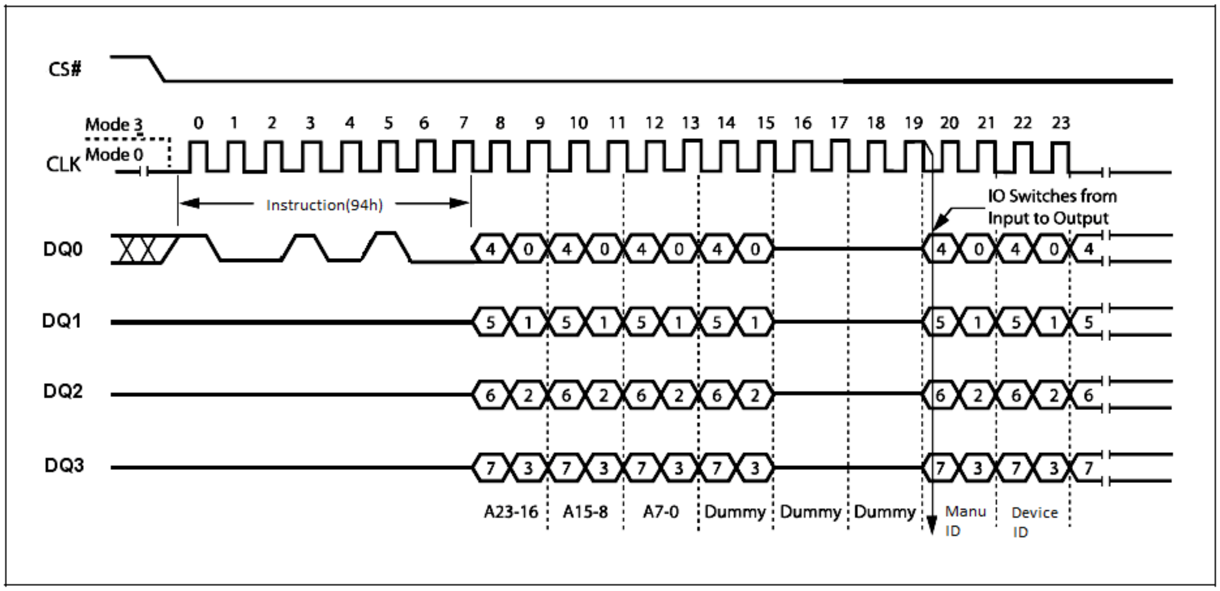
Figure 60. Read Manufacturer / Device ID by Dual I/O Diagram



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 61. Read Manufacturer / Device ID by Quad I/O Diagram



Read Identification (RDID) (9Fh)

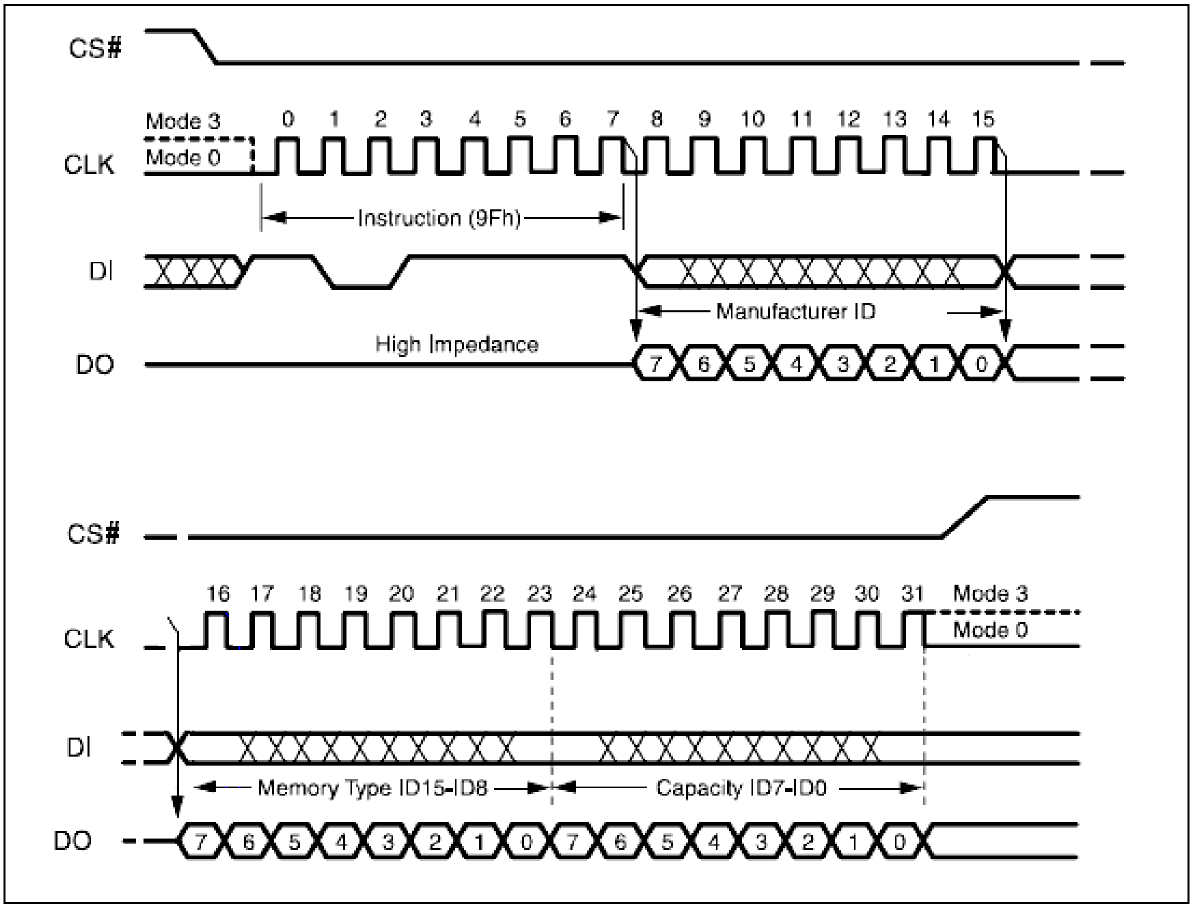
The Read Identification (RDID) instruction allows the 8-bit manufacturer identification to be read, followed by two bytes of device identification. The device identification indicates the memory type in the first byte, and the memory capacity of the device in the second byte.

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.

Figure 62. Read Identification (RDID)



Program OTP array (42h)

The Program OTP array operation is similar to the Page Program instruction. It allows from one byte to 256 bytes of security register data to be programmed at previously erased (FFh) memory locations. A Write Enable instruction must be executed before the device will accept the Program OTP array Instruction. The instruction is initiated by driving the CS# pin low then shifting the instruction code "42h" followed by a 24-bit address (A23-A0)(or 32-bit address, depends on address mode state) 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.

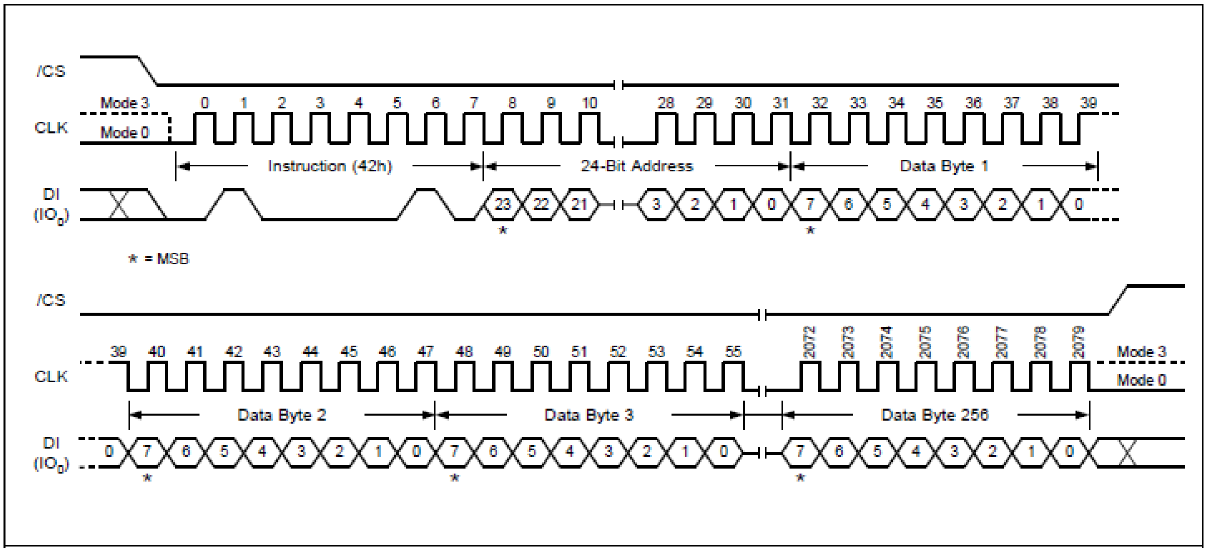
The Program OTP array instruction sequence is shown in Program OTP array figure. The OTP array Lock Bits (SPL0-SPL3) in Status Register2 can be used to OTP protect the OTP array data. Once a lock bit is set to 1, the corresponding OTP array will be permanently locked, Program OTP array instruction to that register will be ignored.

Table 16. OTP Sector Address

Lock bit	Sector	Sector Size	Address Range (4byte address mode)	Address Range (3byte address mode) Extended Address , EA0=1
SPL0	8191	512 byte	1FFF000h –1FFF1FFh	FFF000h –FFF1FFh
SPL1	8190	512 byte	1FFE000h – 1FFE1FFh	FFE000h – FFE1FFh
SPL2	8189	512 byte	1FFD000h – 1FFD1FFh	FFD000h – FFD1FFh

Note. The OTP sector is mapping to sector 8191, 8190 and 8189.

Figure 63. Program OTP array



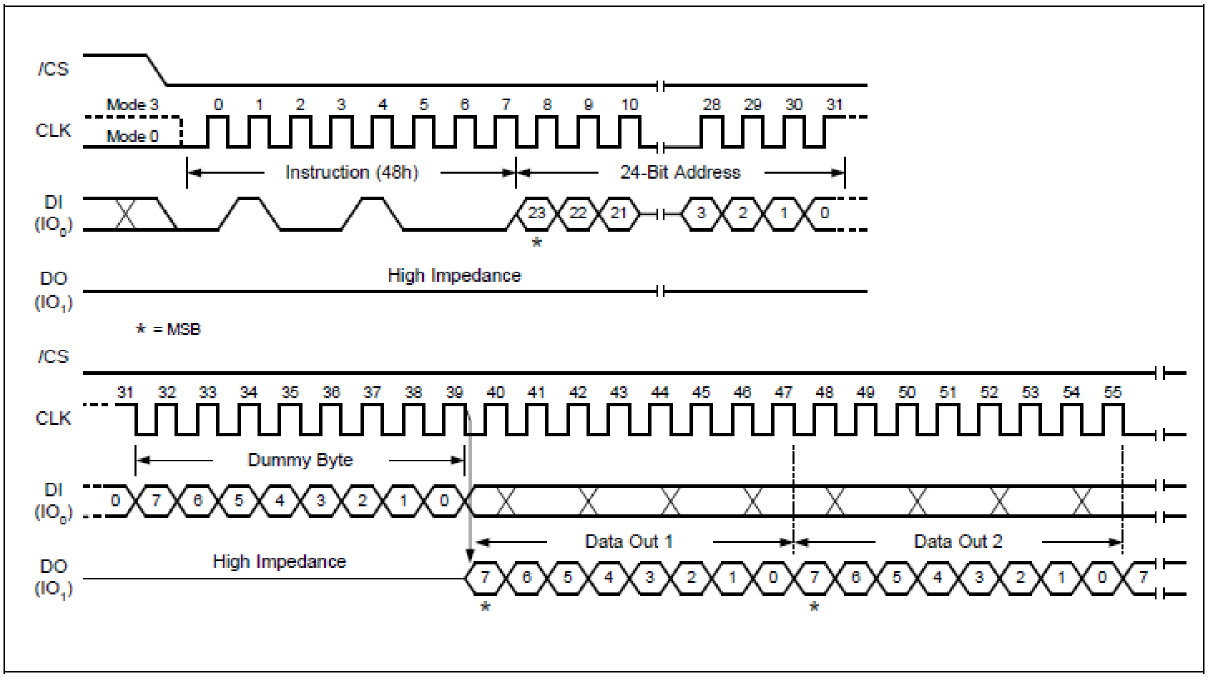
Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

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) (or 32-bit address, depends on address mode state) 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).

Figure 64. Read OTP array



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

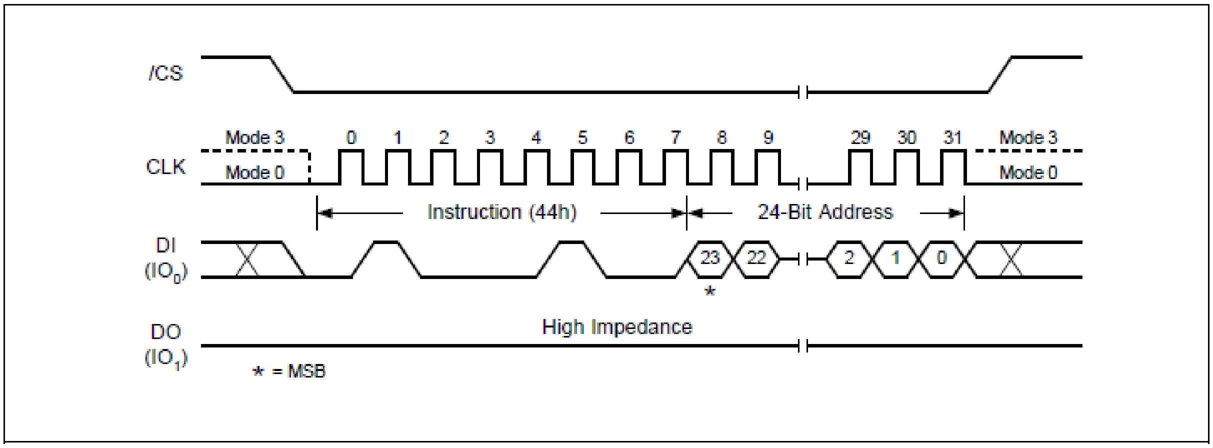
Erase OTP array (44h)

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) (or 32-bit address, depends on address mode state) 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.

Figure 65. Erase OTP array



Note: The above address cycles are base on 3-byte address mode, for 4-byte address mode, the address cycles will be increased.

Read SFDP Mode and Unique ID Number (5Ah)

Read SFDP mode

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 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.

The SFDP version is 216B.

Figure 66. Read SFDP Mode and Unique ID Number Instruction Sequence Diagram

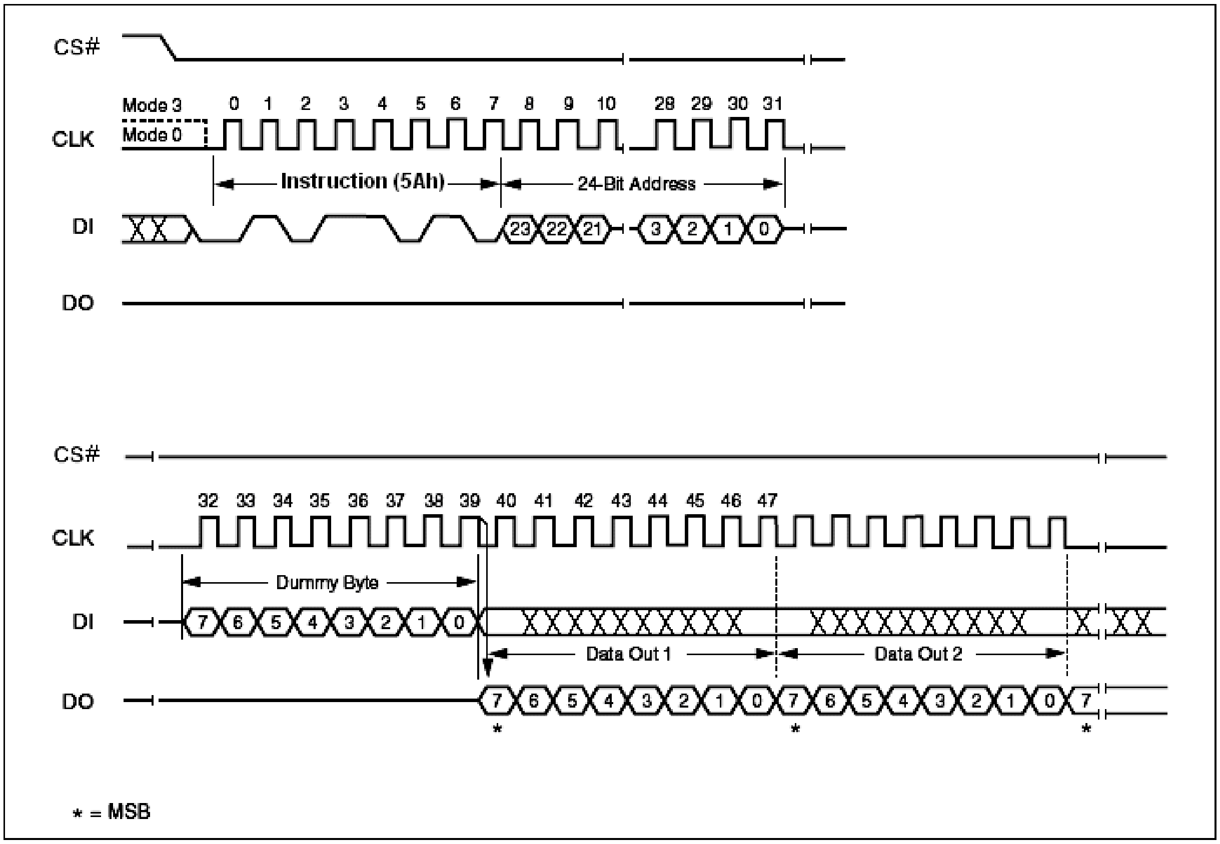


Table 17. Serial Flash Discoverable Parameters (SFDP) Signature and Parameter Identification Data Value (Advanced Information)

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	4 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 double word)	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 double word)	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 18. 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 01	01b	E5h	00 = reserved 01 = 4KB erase 10 = reserved 11 = 64KB erase
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	FBh	0 = not supported 1 = supported
Address Byte Number of bytes used in addressing for flash array read, write and erase.		17 18	01b		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
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 18. Parameter ID (0) (Advanced Information) 2/16

Description	Address (h) (Byte Mode)	Address (Bit)	Data (h)	Comment
Flash Memory Density	37h : 34h	31 : 00	0FFFFFFFh	256 Mbits

Table 18. 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 18. 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 18. 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	EEh	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	0b		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 18. 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 18. 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	00000b	00h	4 dummy clocks
(4-4-4) Fast Read Number of Mode Bits		23:21	000b		8 mode bits
(4-4-4) Fast Read Opcode Opcode for quad input opcode/address, quad output data Fast Read.	4Bh	31 : 24	FFh	FFh	Must Enter EQPI Mode Firstly

Table 18. 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 18. 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 18. 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)		07:04	00010b		count
	55h	08		62h	
		10:09	01b		units : 00b: 1 ms, 01b: 16 ms, 10b: 128 ms, 11b: 1 s
Erase type 2 Erase, typical time (typical time = (count + 1)*units)		15:11	01100b		count
	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 18. 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	11110b	DEh	count
		30:29	10b		Units : 00b:16ms, 01b:256ms, 10b:4s, 11b:64s
Reserved		31	1b		Reserved

Table 18. 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, 01bus
Suspend/Resume supported		31	0b		0:support:1us, 10b:8us, 11b:64ed 1:not supported

Table 18. 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 18. 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 18. 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	0000b	00h	xxx1b: issue FFh instruction 1xxxb: issue the Soft Reset 66/99 sequence
4-4-4 mode enable sequences		07:04	00000b		x_xx1xb: issue instruction 38h
	69h	08		90h	
0-4-4 mode supported		09	0b		0: not supported 1:supported
0-4-4 mode Exit Method		15:10	100100b		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	1000b	48h	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		23	0b		0:not supported
Reserved	6Bh	31:24	FFh	FFh	Reserved

Table 18. 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	50h	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	00101b		xx_xxxx_xxx1b: issue instruction E9h to exit 4-Byte address mode (write enable instruction 06h is not required) x1_xxxx_xxxxb: Reserved 1x_xxxx_xxxxb: Reserved
	6Eh	18:16		C1h	
		23:19	11000b		
Enter 4-Byte Address	6Fh	31:24	10100101b	A5h	xxxx_xxx1b: issue instruction B7h (preceding write enable not required)

Table 19. 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	1b	FFh	0:not supported 1:supported
Support for 1-1-1 Fast Read Command, Instruction=0Ch		01	1b		0:not supported 1:supported
Support for 1-1-2 Fast Read Command, Instruction=3Ch		02	1b		0:not supported 1:supported
Support for 1-2-2 Fast Read Command, Instruction=BCh		03	1b		0:not supported 1:supported
Support for 1-1-4 Fast Read Command, Instruction=ECh		04	1b		0:not supported 1:supported
Support for 1-4-4 Fast Read Command, Instruction=ECh		05	1b		0:not supported 1:supported
Support for 1-1-1 Page Program Command, Instruction=12h		06	1b		0:not supported 1:supported
Support for 1-1-4 Page Program Command, Instruction=34h		07	1b		0:not supported 1:supported
Support for 1-4-4 Page Program Command, Instruction=3Eh	C1h	08	0b	0Eh	0:not supported 1:supported
Support for Erase Command-Type 1 size, instruction lookup in next Dword		09	1b		0:not supported 1:supported
Support for Erase Command-Type 2 size, instruction lookup in next Dword		10	1b		0:not supported 1:supported
Support for Erase Command-Type 3 size, instruction lookup in next Dword		11	1b		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	
Support for volatile individual sector lock Write command, Instruction=E1h		17	0b		
Support for non-volatile individual sector lock read command, Instruction=E2h		18	0b		
Support for non-volatile indivdual sector lock write command, Instrucion=E3h		19	0b		
Reserved		23:20	1111b		
Reserved	C3h	31:24	FFh	FFh	
Instruction for Erase Type 1	C4h	07:00	21h	21h	
Instruction for Erase Type 2	C5h	15:08	5Ch	5Ch	
Instruction for Erase Type 3	C6h	23:16	DCh	DCh	
Instruction for Erase Type 4	C7h	31:24	FFh	FFh	

Table 20. 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 21. 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	**1600=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	99h		
		11:08		F9h	
Program Suspend/Resume		12	1b		**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	1Bh	1Bh	
Wrap Read data length	117h	31:24	64h	64h	
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	FF		
		09:08		CBh	
Individual block lock Volatile protect bit default protect status		10	0b		**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 22. 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.

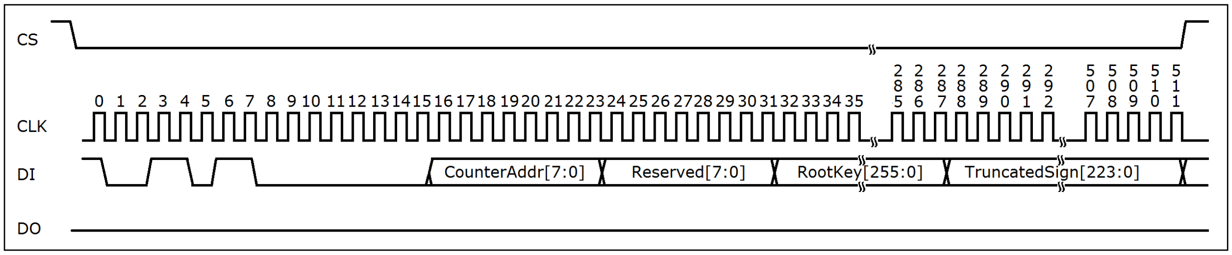
- After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:
- Payload size is correct. (include OP1 is 64 bytes)
- Counter Address falls within the range of supported counters.
- The Root Key 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 67. 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.

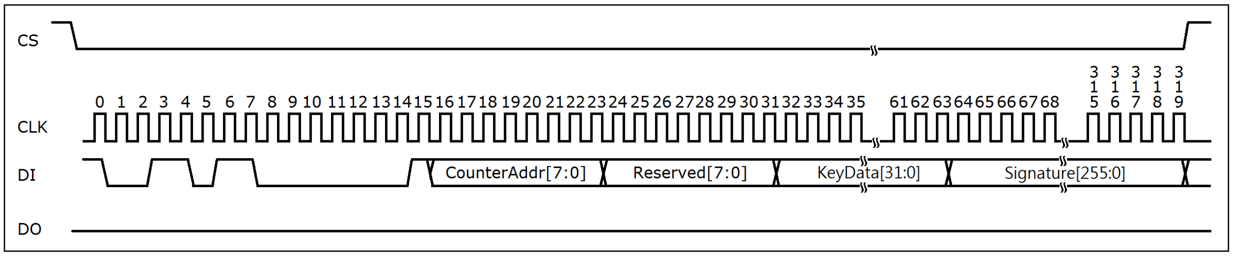
After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:

- 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 68. Update HMAC Key Instruction Sequence Diagram



Increment Monotonic Counter (9Bh + 02h)

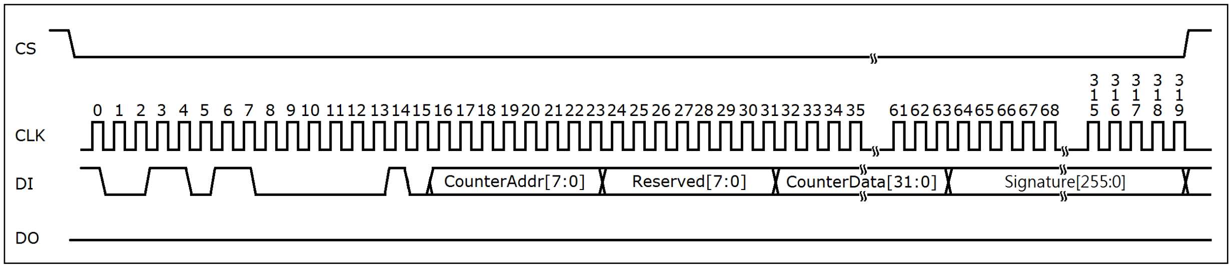
This command is used by the SPI Flash Controller to increment the Monotonic counter by 1 inside the SPI Flash Device.

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 69. 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.

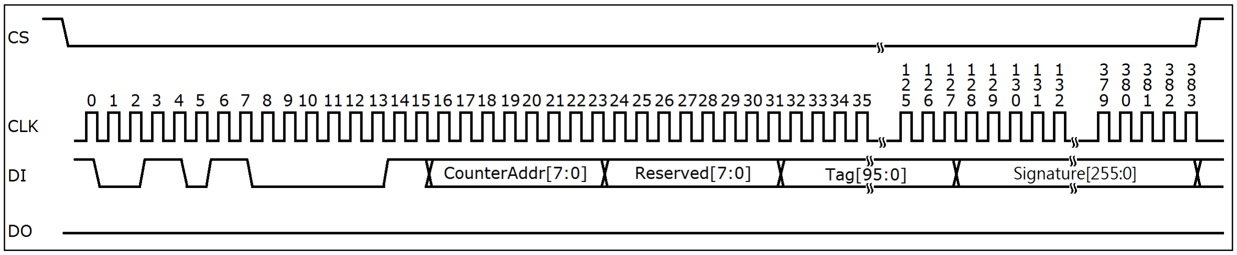
After the command is issued on the interface the SPI Flash device must ensure that the received transaction is error free. This includes checking following conditions:

- 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 70. Request Monotonic Counter Instruction Sequence Diagram



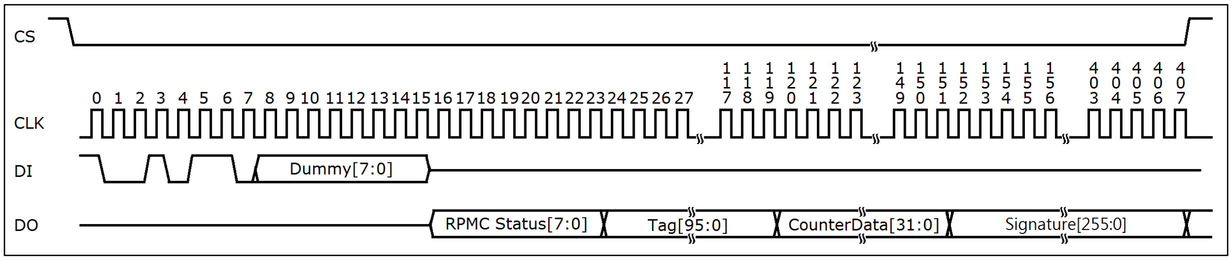
Reserved RPMC Commands (9Bh + 04h~FFh)

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)

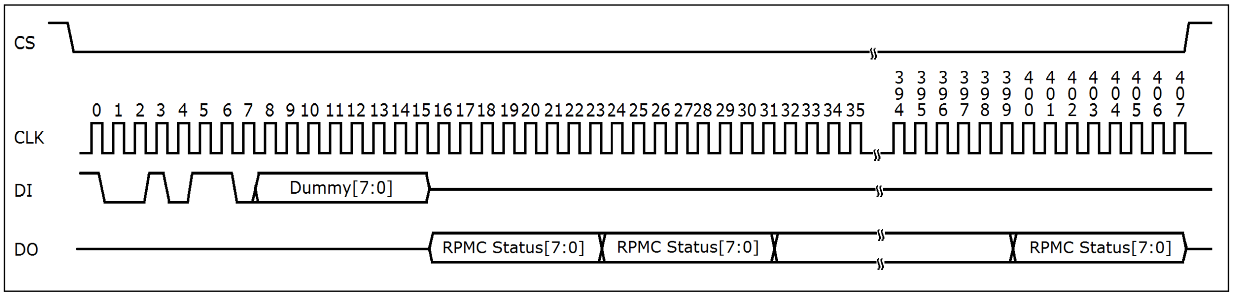
This command is used by the SPI Flash Controller to read the RPMC status from any previously issued OP1 command. In addition, if previous OP1 command is Request Monotonic Counter and if SPI Flash returns successful completion (BUSY=0) in the RPMC Status Register, then it must also return valid values 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 71. 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, Counter Data & Signature fields.

Figure 72. 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 (0V to 1.65V in less than 165 ms). See Power-Up Timing table and Power-up Timing figure for more information.

Figure 73. Power-up Timing

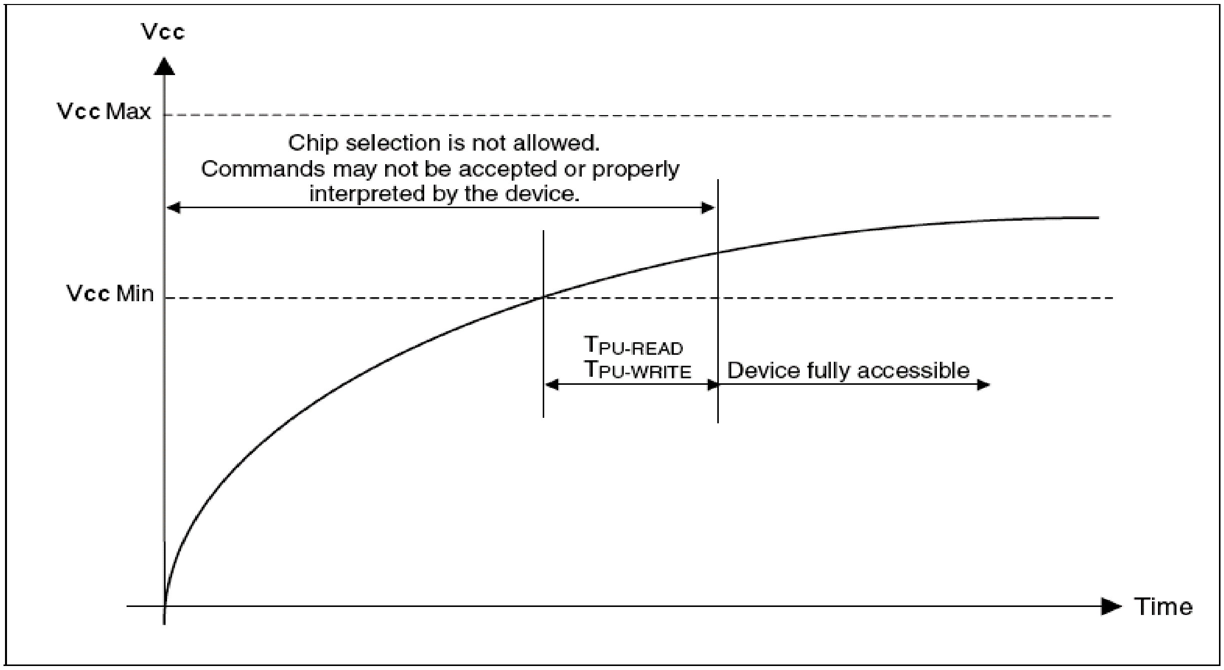


Table 23. Power-Up Timing

Symbol	Parameter	Min.	Unit
$T_{PU\text{-}READ}$ *1	V_{CC} Min to Read Operation	100	µs
$T_{PU\text{-}WRITE}$ *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 24. DC Characteristics

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

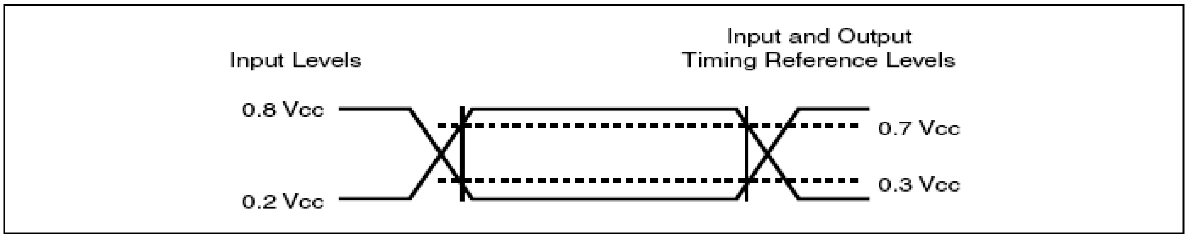
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
I_{LI}	Input Leakage Current		-	1	± 2	µA
I_{LO}	Output Leakage Current		-	1	± 2	µA
I_{CC1}	Standby Current	CS# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	-	1	20	µA
I_{CC2}	Deep Power-down Current	CS# = V_{CC}, V_{IN} = V_{SS} or V_{CC}	-	1	20	µA
I_{CC3}	Operating Current (READ)	CLK = 0.1 V_{CC} / 0.9 V_{CC} at 104 MHz, DQ = open	-	12	18	mA
		CLK = 0.1 V_{CC} / 0.9 V_{CC} at 133 MHz, Quad Output Read, DQ = open	-	20	30	mA
		CLK = 0.1 V_{CC} / 0.9 V_{CC} at 104 MHz, Quad Output Read, DQ = open	-	16	22	mA
I_{CC4}	Operating Current (PP)	CS# = V_{CC}	-	9	30	mA
I_{CC5}	Operating Current (WRSR/WRSR3)	CS# = V_{CC}	-	-	25	mA
I_{CC6}	Operating Current (SE)	CS# = V_{CC}	-	13	25	mA
I_{CC7}	Operating Current (HBE/BE)	CS# = V_{CC}	-	13	25	mA
I_{CC8}	Operating Current (CE)	CS# = V_{CC}	-	13	25	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		0.7 V_{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

AC Measurement Conditions

Table 25. 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 74. AC Measurement I/O Waveform



AC Timing Input / Output Conditions

Table 26. 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 Clock Frequency for: PP, QPP, SE, HBE, BE, CE, DP, RES, RDP, WREN, WRDI, WRSR, WRSR3, Fast Read RDSR, RDSR3, RDID, Dual Output Fast Read, Dual I/O Fast Read, Quad I/O Fast Read	D.C.	-	104	MHz
		Serial DTR Clock Frequency for: DDR Fast Read, DDR Read Burst with Wrap, DDR Mode Page Program, DDR Dual I/O Fast Read, DDR Quad I/O Fast Read	D.C.	-	66	MHz
		Serial Quad Clock Frequency for: Quad Output Fast Read, Quad I/O Fast Read (V_{CC}: 1.8-1.95V)	D.C.	-	133	MHz
		Serial DTR Clock Frequency for: DDR Fast Read, DDR Read Burst with Wrap, DDR Mode Page Program, DDR Dual I/O Fast Read, DDR Quad I/O Fast Read (V_{CC}: 1.8-1.95V)	D.C.	-	71	MHz
f_R		Serial Clock Frequency for READ	D.C.	-	50	MHz
F_R	fc	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	30	-	-	ns
		CS# High Time (RPMC)	50	-	-	ns
t_{SHSL} *2	t_{CSH}	Volatile Register Write Time	50	-	-	ns
t_{SHQZ} *2	t_{DIS}	Output Disable Time	-	-	7	ns
		Output Disable Time (RPMC)	-	-	10	ns
t_{CLQX}	t_{HO}	Output Hold Time	1.5	-	-	ns
t_{DVCH}	t_{DSU}	Data In Setup Time	2	-	-	ns
t_{CHDX}	t_{DH}	Data In Hold Time	3	-	-	ns
		Data In Hold Time (RPMC)	3	-	-	ns

Table 26. AC Characteristics (Continued)

Symbol	Alt	Parameter	Min	Typ	Max	Unit
t_{HLCH}		HOLD# Low Setup Time (relative to CLK)	5	-	-	ns
t_{HHCH}		HOLD# High Setup Time (relative to CLK)	5	-	-	ns
t_{CHHH}		HOLD# Low Hold Time (relative to CLK)	5	-	-	ns
t_{CHHL}		HOLD# High Hold Time (relative to CLK)	5	-	-	ns
t_{HLQZ} *2	t_{HZ}	HOLD# to Output High-Z	-	-	6	ns
t_{HHQX} *2	t_{LZ}	HOLD# to Output Low-Z	-	-	6	ns
t_{CLQV}	t_V	Output Valid from CLK for 30 pF	-	-	8	ns
		Output Valid from CLK for 15 pF	-	-	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	-	120	400	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	-	-	35	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 75. Serial Output Timing

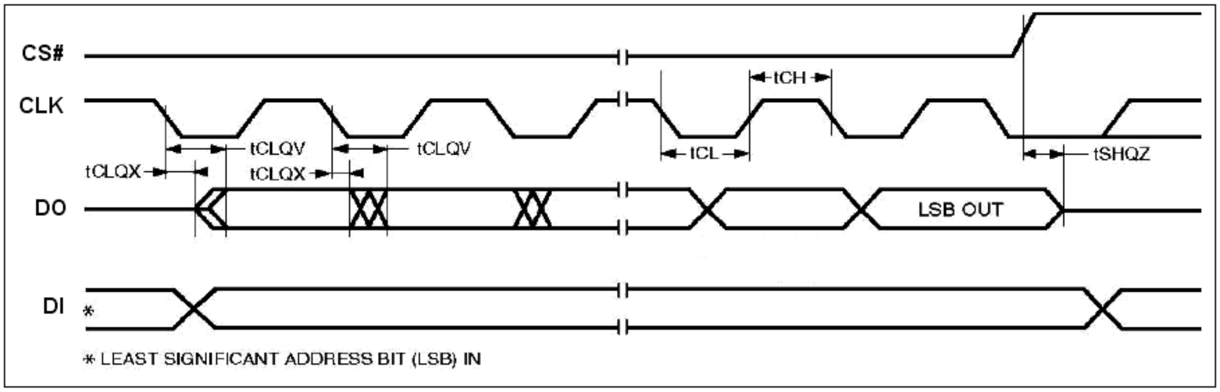


Figure 76. Input Timing

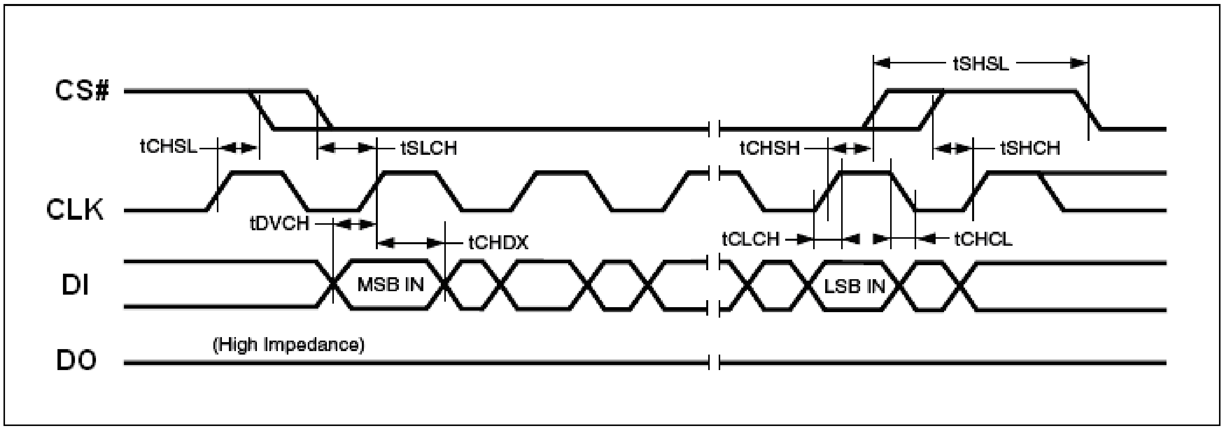


Figure 77. Hold Timing

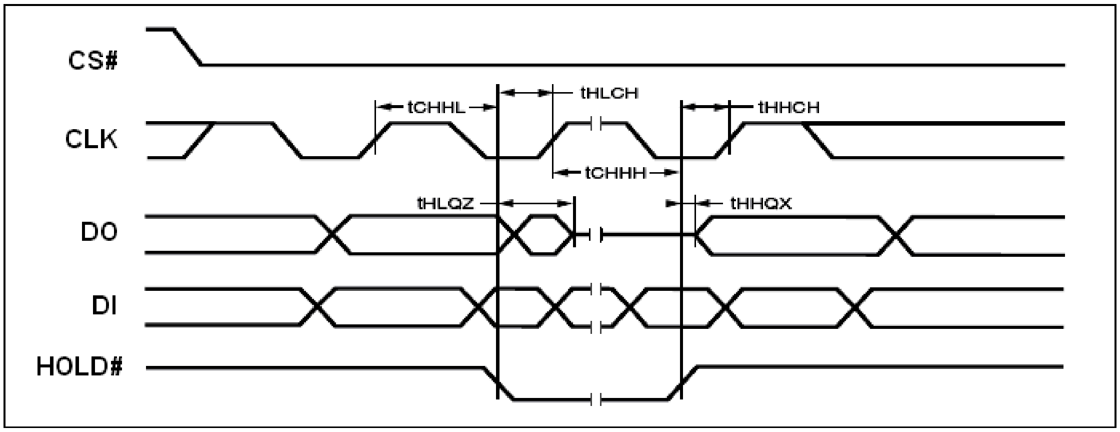


Figure 78. Serial Output Timing for Double Data Rate Mode

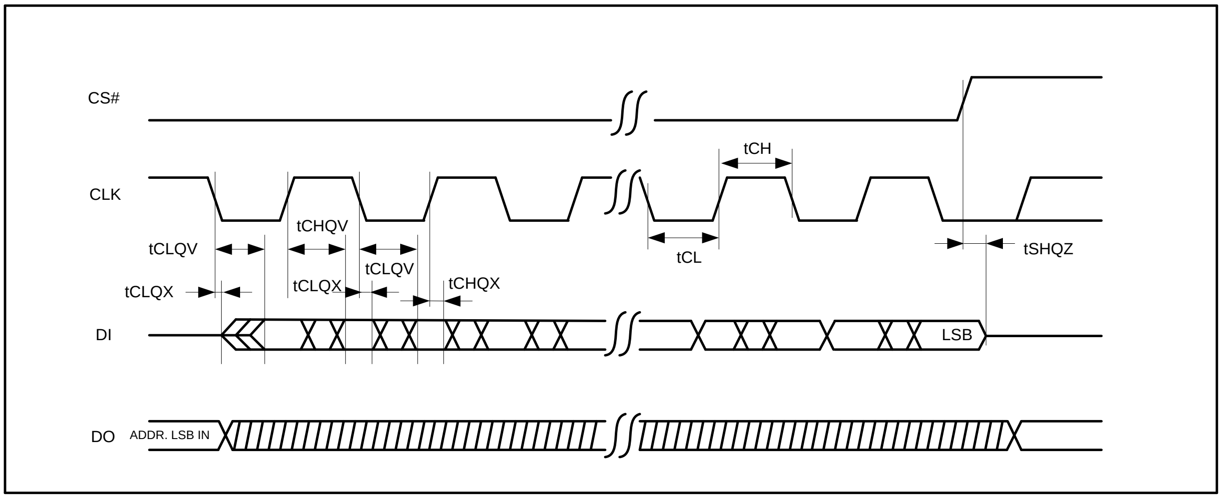
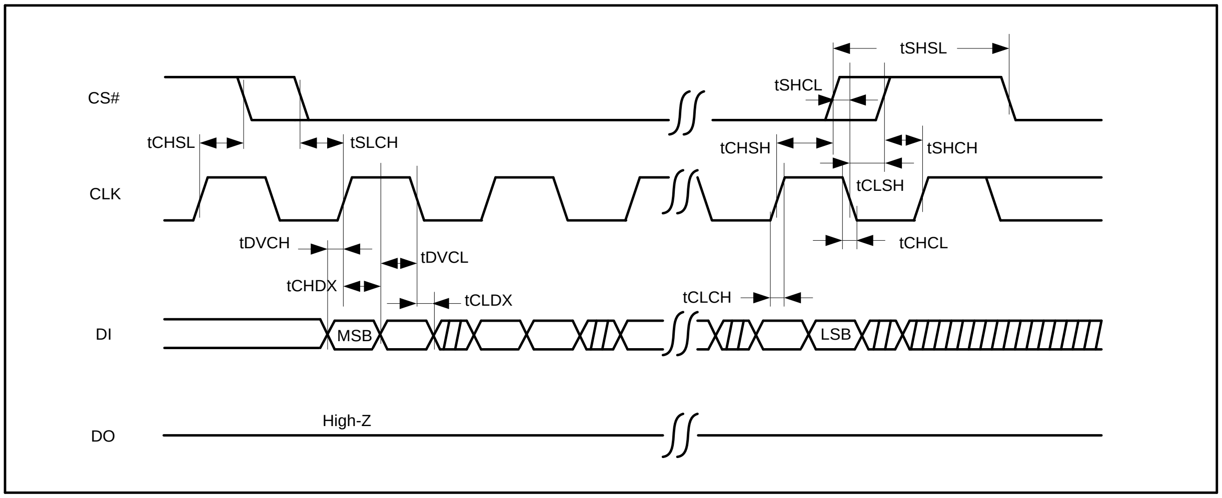


Figure 79. Serial Input Timing for Double Data Rate Mode



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 27. 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]

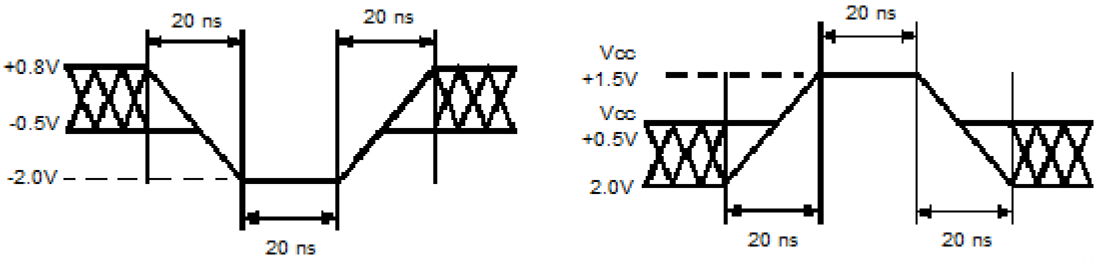
Table 28. Operating Conditions

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 80. Overshoot Waveform



Maximum Negative Overshoot Waveform

Maximum Positive Overshoot Waveform

INPUT / OUTPUT CAPACITANCE

Table 29. CAPACITANCE

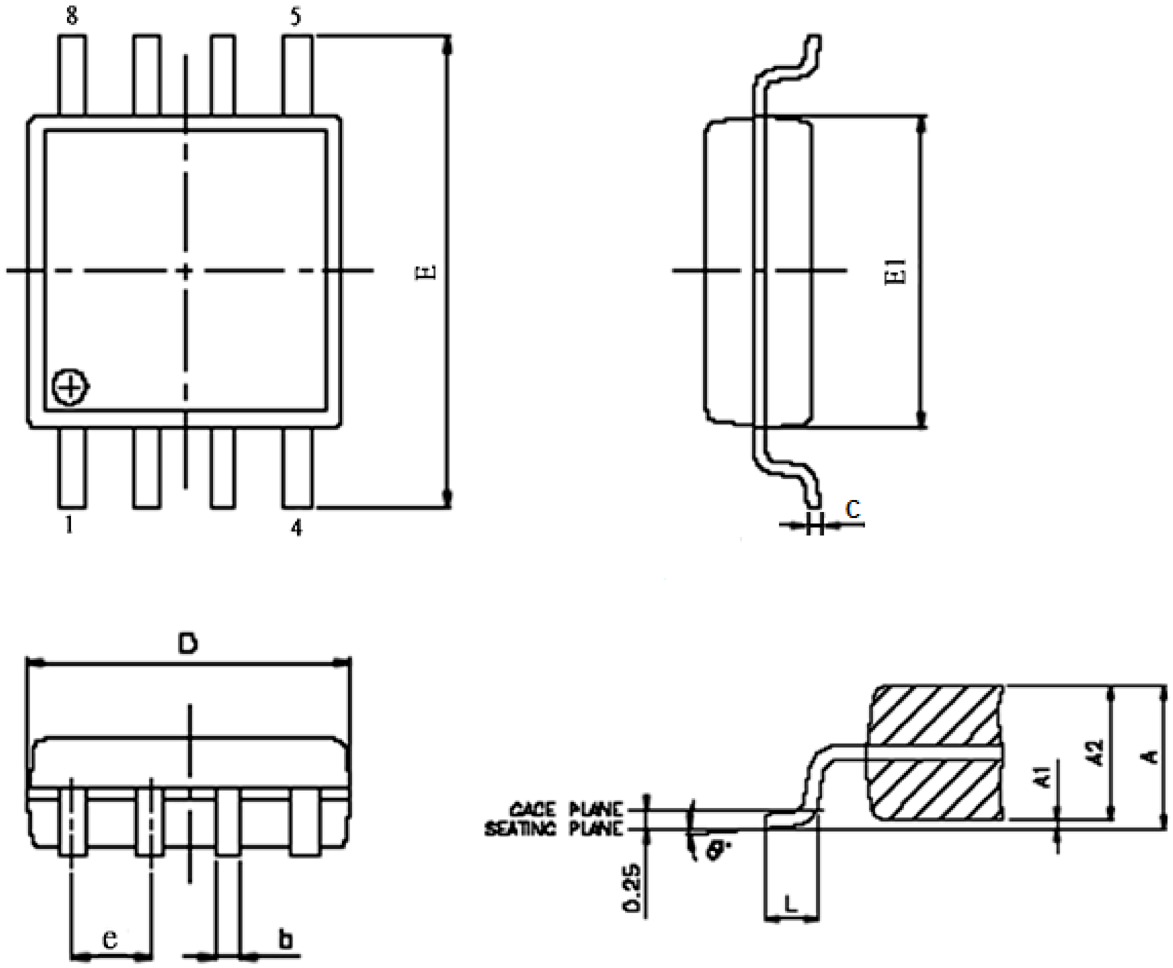
(V_{CC} = 1.65-1.95V)

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

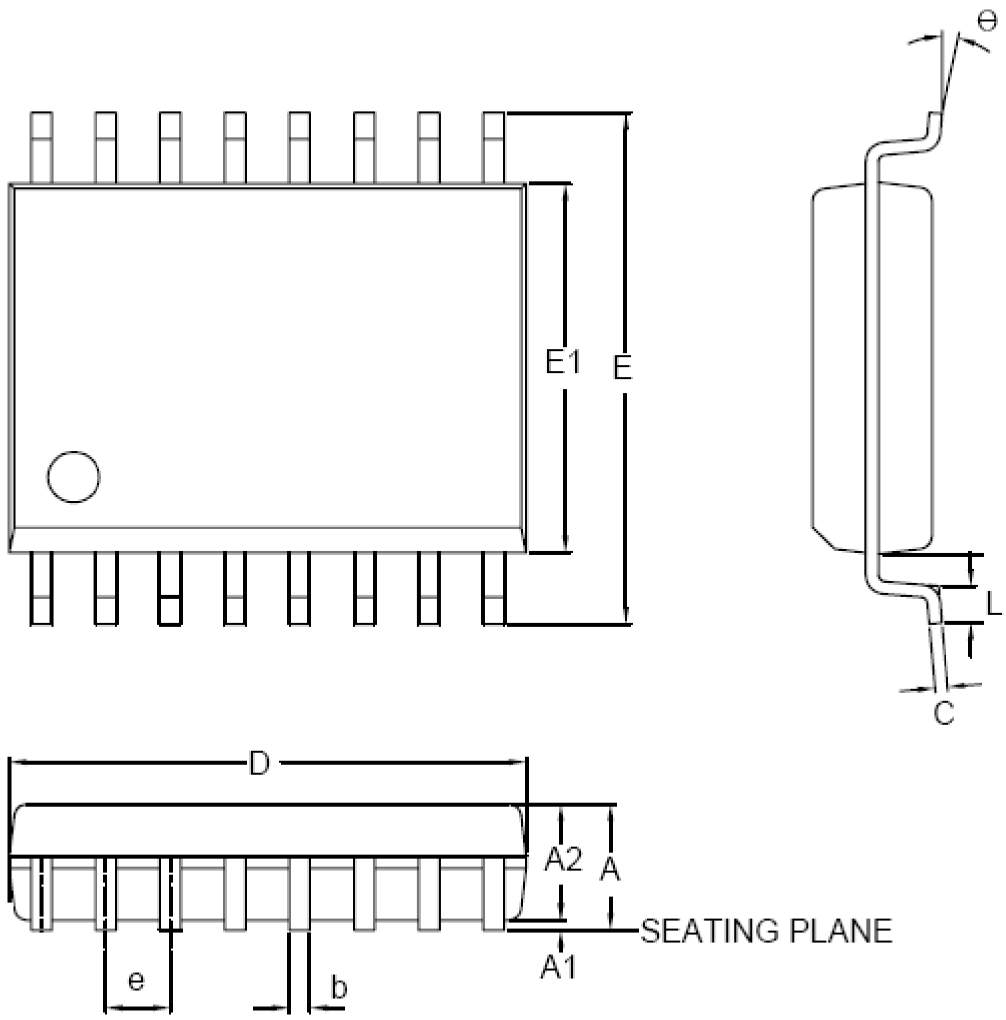
Figure 81. SOP 200 mil (official name = 208 mil)



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.

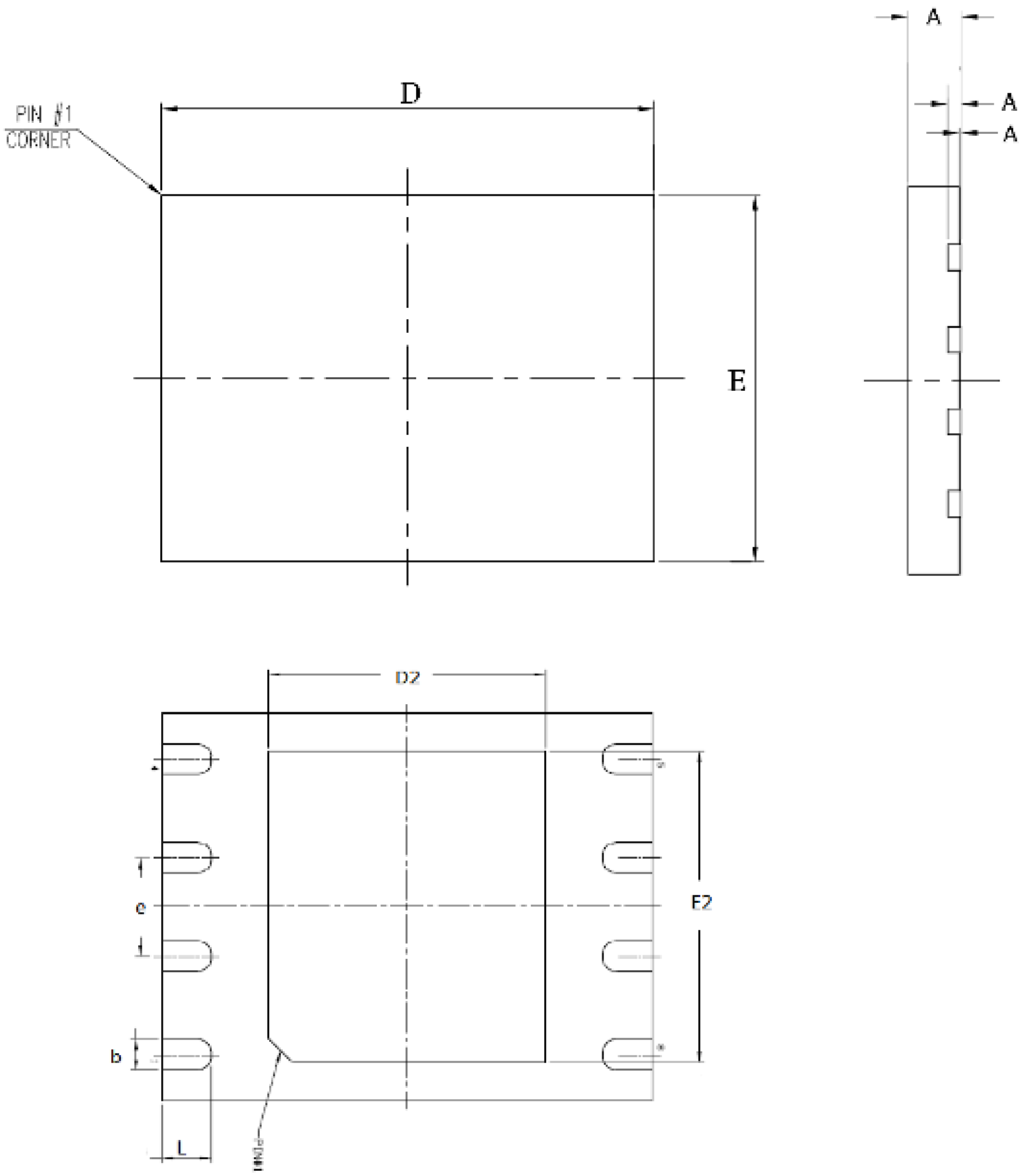
Figure 82. 16-LEAD SOP 300 mil



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 83. 8-LEAD VDFN / WSON (6x5 mm)

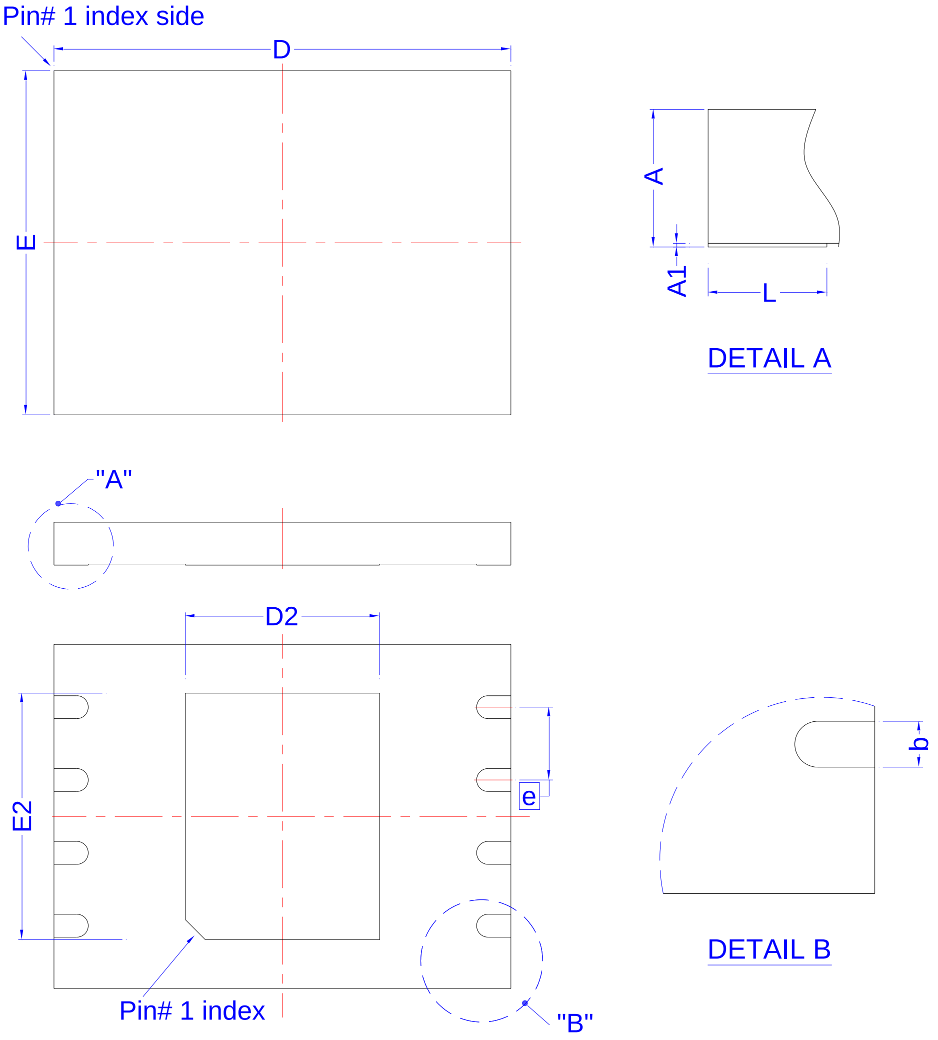


Controlling dimensions are in millimeters (mm).

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 84. 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	0.45	0.014	0.016	0.018
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	0.45	0.50	0.55	0.018	0.020	0.022

Controlling dimension: millimeter
(Revision date: Sep 07 2021)

ORDERING INFORMATION

EN35SXR256A - 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
256 = 256 Megabit (32,768K x 8)
A = version identifier

Revisions List

Revision No	Description	Date
0.1	Initial Release	2023/04/13
0.2	Remove enhance and EQPI mode	2023/05/18
1.0	Delete Preliminary	2023/07/13

Important Notice

All rights reserved.

No part of this document may be reproduced or duplicated in any form or by any means without the prior permission of ESMT.

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.