EN25S64
64 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
- 64 M-bit Serial Flash
  - 64 M-bit / 8,192 KByte /32,768 pages
  - 256 bytes per programmable page
- Standard, Dual or Quad SPI
  - Standard SPI: CLK, CS#, DI, DO, WP#
  - Dual SPI: CLK, CS#, DQ_0, DQ_1, WP#
  - Quad SPI: CLK, CS#, DQ_0, DQ_1, DQ_2, DQ_3
- High performance
  - 104MHz clock rate for one data bit
  - 80MHz clock rate for two data bits
  - 80MHz clock rate for four data bits
- Burst Modes
  - Continuous linear burst
  - 8/16/32/64/128/256 linear burst with wrap-around
- Low power consumption
  - 12 mA typical active current
  - 1μA typical power down current
- Uniform Sector Architecture:
  - 2048 sectors of 4-Kbyte
  - 128 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: 0.7ms typical
  - Sector erase time: 40ms typical
  - Block erase time 300ms typical
  - Chip erase time: 34 Seconds typical
- Write Suspend and Write Resume
- Lockable 512 byte OTP security sector
- Minimum 100K endurance cycle
- Package Options
  - 8 pins VSOP 200mil body width
  - 8 contact VDFN
  - All Pb-free packages are RoHS compliant
- Industrial temperature Range

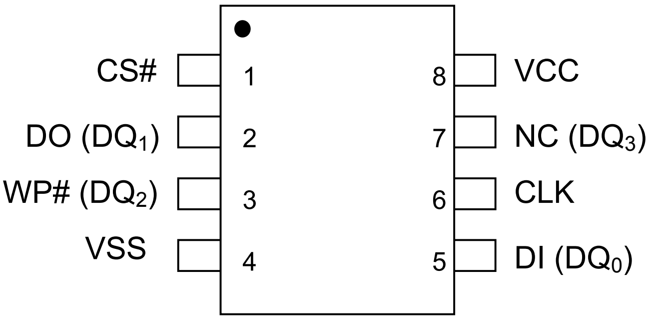
GENERAL DESCRIPTION

The EN25S64 is a 64 Megabit (8,192K-byte) Serial Flash memory, with advanced write protection mechanisms. The EN25S64 supports the standard Serial Peripheral Interface (SPI), and a high performance Dual output as well as Dual, Quad I/O using SPI pins: Serial Clock, Chip Select, Serial DQ_0 (DI) and DQ_1(DO), DQ_2(WP#) and DQ_3(NC). SPI clock frequencies of up to 80MHz are supported allowing equivalent clock rates of 160MHz for Dual Output and 320MHz for Quad Output when using the Dual/Quad Output Fast Read instructions. The memory can be programmed 1 to 256 bytes at a time, using the Page Program instruction.

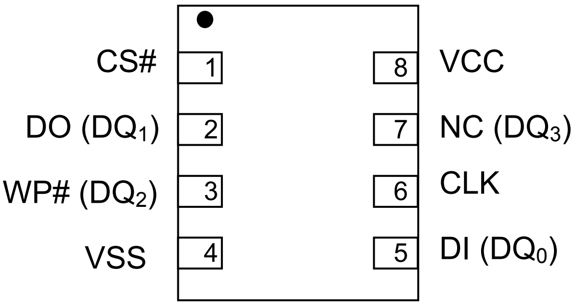
The EN25S64 also offers a sophisticated method for protecting individual blocks against erroneous or malicious program and erase operations. By providing the ability to individually protect and unprotect blocks, a system can unprotect a specific block to modify its contents while keeping the remaining blocks of the memory array securely protected. This is useful in applications where program code is patched or updated on a subroutine or module basis or in applications where data storage segments need to be modified without running the risk of errant modifications to the program code segments.

The EN25S64 is designed to allow either single Sector/Block at a time or full chip erase operation. The EN25S64 can be configured to protect part of the memory as the software protected mode. The device can sustain a minimum of 100K program/erase cycles on each sector or block.

Figure.1 CONNECTION DIAGRAMS



8 - LEAD VSOP



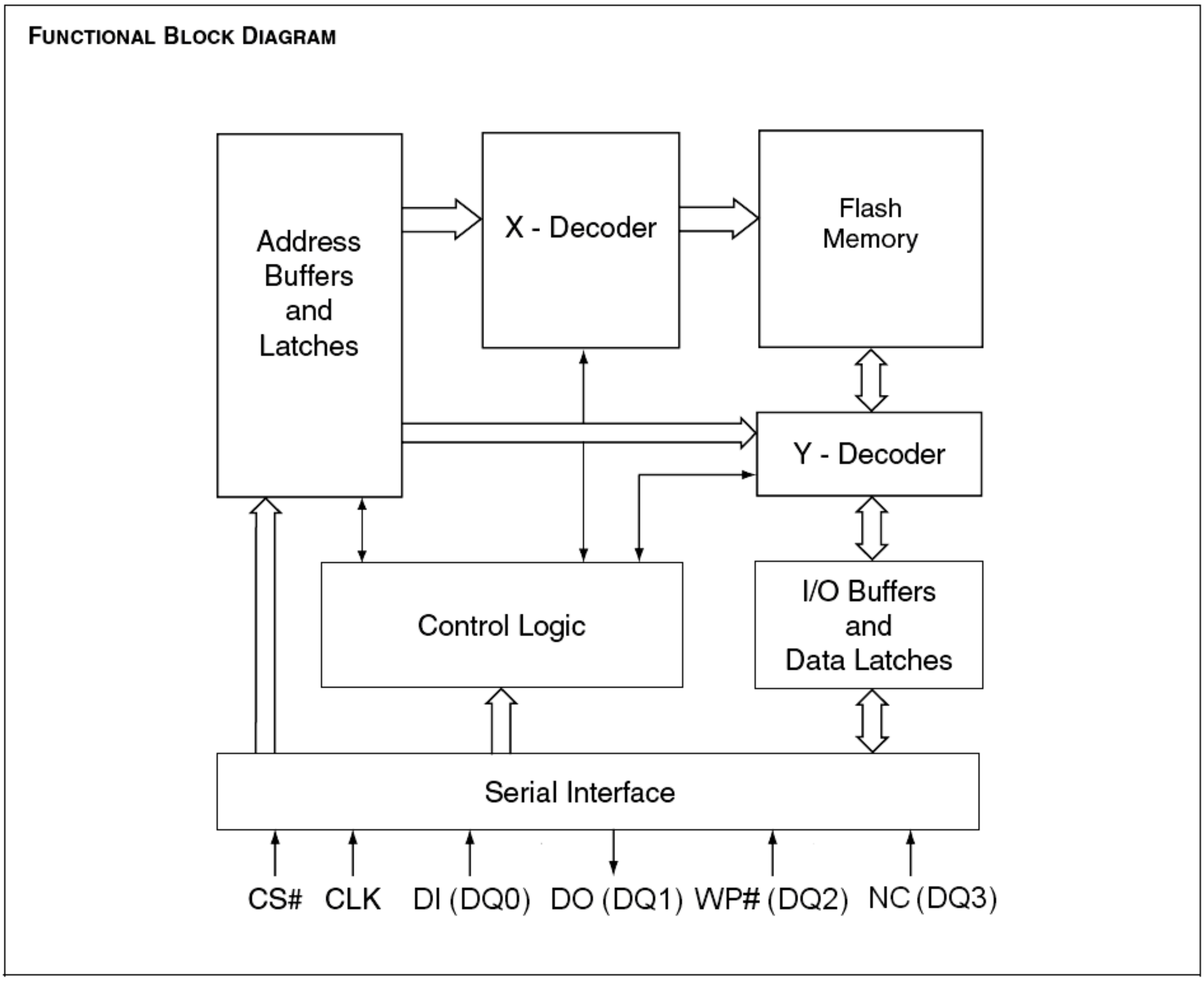
8 - LEAD VDFN

Table 1. Pin Names

Symbol	Pin Name
CLK	Serial Clock Input
DI (DQ_0)	Serial Data Input (Data Input Output 0) *1
DO (DQ_1)	Serial Data Output (Data Input Output 1) *1
CS#	Chip Enable
WP# (DQ_2)	Write Protect (Data Input Output 2) *2
NC(DQ_3)	Not Connect (Data Input Output 3) *2
Vcc	Supply Voltage (1.65-1.95V)
Vss	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.

Figure 2. BLOCK DIAGRAM



Note:

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

SIGNAL DESCRIPTION

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

The EN25S64 support standard SPI, Dual SPI and Quad SPI operation. Standard SPI instructions use the unidirectional DI (input) pin to serially write instructions, addresses or data to the device on the rising edge of the Serial Clock (CLK) input pin. Standard SPI also uses the unidirectional DO (output) to read data or status from the device on the falling edge CLK.

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

Serial Clock (CLK)

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

Chip Select (CS#)

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

Write Protect (WP#)

The Write Protect (WP#) pin can be used to prevent the Status Register from being written. Used in conjunction with the Status Register's Block Protect (BP0, BP1, BP2, BP3) bits and Status Register Protect (SRP) bits, a portion or the entire memory array can be hardware protected. The WP# function is only available for standard SPI and Dual SPI operation, when during Quad SPI, this pin is the Serial Data IO (DQ_2) for Quad I/O operation.

MEMORY ORGANIZATION

The memory is organized as:

- 8,388,608 bytes
- Uniform Sector Architecture
  128 blocks of 64-Kbyte
  2,048 sectors of 4-Kbyte
  32,768 pages (256 bytes each)

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

Table 2. Uniform Block Sector Architecture (1/4)

Block	Sector	Address range	
127	2047	7FF000h	7FFFFFh
	⋮	⋮	⋮
	2032	7F0000h	7F0FFFh
126	2031	7EF000h	7EFFFFh
	⋮	⋮	⋮
	2016	7E0000h	7E0FFFh
125	2015	7DF000h	7DFFFFh
	⋮	⋮	⋮
	2000	7D0000h	7D0FFFh
124	1999	7CF000h	7CFFFFh
	⋮	⋮	⋮
	1984	7C0000h	7C0FFFh
123	1983	7BF000h	7BFFFFh
	⋮	⋮	⋮
	1968	7B0000h	7B0FFFh
122	1967	7AF000h	7AFFFFh
	⋮	⋮	⋮
	1952	7A0000h	7A0FFFh
121	1951	79F000h	79FFFFh
	⋮	⋮	⋮
	1936	790000h	790FFFh
120	1935	78F000h	78FFFFh
	⋮	⋮	⋮
	1920	780000h	780FFFh
119	1919	77F000h	77FFFFh
	⋮	⋮	⋮
	1904	770000h	770FFFh
118	1903	76F000h	76FFFFh
	⋮	⋮	⋮
	1888	760000h	760FFFh
117	1887	75F000h	75FFFFh
	⋮	⋮	⋮
	1872	750000h	750FFFh
116	1871	74F000h	74FFFFh
	⋮	⋮	⋮
	1856	740000h	740FFFh
115	1855	73F000h	73FFFFh
	⋮	⋮	⋮
	1840	730000h	730FFFh
114	1839	72F000h	72FFFFh
	⋮	⋮	⋮
	1824	720000h	720FFFh
113	1823	71F000h	71FFFFh
	⋮	⋮	⋮
	1808	710000h	710FFFh
112	1807	70F000h	70FFFFh
	⋮	⋮	⋮
	1972	700000h	700FFFh

Block	Sector	Address range	
111	1791	6FF000h	6FFFFFh
	⋮	⋮	⋮
	1776	6F0000h	6F0FFFh
110	1775	6EF000h	6EFFFFh
	⋮	⋮	⋮
	1760	6E0000h	6E0FFFh
109	1759	6DF000h	6DFFFFh
	⋮	⋮	⋮
	1744	6D0000h	6D0FFFh
108	1743	6CF000h	6CFFFFh
	⋮	⋮	⋮
	1728	6C0000h	6C0FFFh
107	1727	6BF000h	6BFFFFh
	⋮	⋮	⋮
	1712	6B0000h	6B0FFFh
106	1711	6AF000h	6AFFFFh
	⋮	⋮	⋮
	1696	6A0000h	6A0FFFh
105	1695	69F000h	69FFFFh
	⋮	⋮	⋮
	1680	690000h	690FFFh
104	1679	68F000h	68FFFFh
	⋮	⋮	⋮
	1664	680000h	680FFFh
103	1663	67F000h	67FFFFh
	⋮	⋮	⋮
	1648	670000h	670FFFh
102	1647	66F000h	66FFFFh
	⋮	⋮	⋮
	1632	660000h	660FFFh
101	1631	65F000h	65FFFFh
	⋮	⋮	⋮
	1616	650000h	650FFFh
100	1615	64F000h	64FFFFh
	⋮	⋮	⋮
	1600	640000h	640FFFh
99	1599	63F000h	63FFFFh
	⋮	⋮	⋮
	1584	630000h	630FFFh
98	1583	62F000h	62FFFFh
	⋮	⋮	⋮
	1568	620000h	620FFFh
97	1567	61F000h	61FFFFh
	⋮	⋮	⋮
	1552	610000h	610FFFh
96	1551	60F000h	60FFFFh
	⋮	⋮	⋮
	1536	600000h	600FFFh

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

Block	Sector	Address range	
95	1535	5FF000h	5FFFFFh
	⋮	⋮	⋮
	1520	5F0000h	5F0FFFh
94	1519	5EF000h	5EFFFFh
	⋮	⋮	⋮
	1504	5E0000h	5E0FFFh
93	1503	5DF000h	5DFFFFh
	⋮	⋮	⋮
	1488	5D0000h	5D0FFFh
92	1487	5CF000h	5CFFFFh
	⋮	⋮	⋮
	1472	5C0000h	5C0FFFh
91	1471	5BF000h	5BFFFFh
	⋮	⋮	⋮
	1456	5B0000h	5B0FFFh
90	1455	5AF000h	5AFFFFh
	⋮	⋮	⋮
	1440	5A0000h	5A0FFFh
89	1439	59F000h	59FFFFh
	⋮	⋮	⋮
	1424	590000h	590FFFh
88	1423	58F000h	58FFFFh
	⋮	⋮	⋮
	1408	580000h	580FFFh
87	1407	57F000h	57FFFFh
	⋮	⋮	⋮
	1392	570000h	570FFFh
86	1391	56F000h	56FFFFh
	⋮	⋮	⋮
	1376	560000h	560FFFh
85	1375	55F000h	55FFFFh
	⋮	⋮	⋮
	1360	550000h	550FFFh
84	1359	54F000h	54FFFFh
	⋮	⋮	⋮
	1344	540000h	540FFFh
83	1343	53F000h	53FFFFh
	⋮	⋮	⋮
	1328	530000h	530FFFh
82	1327	52F000h	52FFFFh
	⋮	⋮	⋮
	1312	520000h	520FFFh
81	1311	51F000h	51FFFFh
	⋮	⋮	⋮
	1296	510000h	510FFFh
80	1295	50F000h	50FFFFh
	⋮	⋮	⋮
	1280	500000h	500FFFh

Block	Sector	Address range	
79	1279	4FF000h	4FFFFFh
	⋮	⋮	⋮
	1264	4F0000h	4F0FFFh
78	1263	4EF000h	4EFFFFh
	⋮	⋮	⋮
	1248	4E0000h	4E0FFFh
77	1247	4DF000h	4DFFFFh
	⋮	⋮	⋮
	1232	4D0000h	4D0FFFh
76	1231	4CF000h	4CFFFFh
	⋮	⋮	⋮
	1216	4C0000h	4C0FFFh
75	1215	4BF000h	4BFFFFh
	⋮	⋮	⋮
	1200	4B0000h	4B0FFFh
74	1119	4AF000h	4AFFFFh
	⋮	⋮	⋮
	1184	4A0000h	4A0FFFh
73	183	49F000h	49FFFFh
	⋮	⋮	⋮
	1168	490000h	490FFFh
72	1167	48F000h	48FFFFh
	⋮	⋮	⋮
	1152	480000h	480FFFh
71	1151	47F000h	47FFFFh
	⋮	⋮	⋮
	1136	470000h	470FFFh
70	1135	46F000h	46FFFFh
	⋮	⋮	⋮
	1120	460000h	460FFFh
69	1119	45F000h	45FFFFh
	⋮	⋮	⋮
	1104	450000h	450FFFh
68	1103	44F000h	44FFFFh
	⋮	⋮	⋮
	1088	440000h	440FFFh
67	1087	43F000h	43FFFFh
	⋮	⋮	⋮
	1072	430000h	430FFFh
66	1071	42F000h	42FFFFh
	⋮	⋮	⋮
	1056	420000h	420FFFh
65	1055	41F000h	41FFFFh
	⋮	⋮	⋮
	1040	410000h	410FFFh
64	1039	40F000h	40FFFFh
	⋮	⋮	⋮
	1024	400000h	400FFFh

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

Block	Sector	Address range	
63	1023	3FF000h	3FFFFFh
	⋮	⋮	⋮
	1008	3F0000h	3F0FFFh
62	1007	3EF000h	3EFFFFh
	⋮	⋮	⋮
	992	3E0000h	3E0FFFh
61	991	3DF000h	3DFFFFh
	⋮	⋮	⋮
	976	3D0000h	3D0FFFh
60	975	3CF000h	3CFFFFh
	⋮	⋮	⋮
	960	3C0000h	3C0FFFh
59	959	3BF000h	3BFFFFh
	⋮	⋮	⋮
	944	3B0000h	3B0FFFh
58	943	3AF000h	3AFFFFh
	⋮	⋮	⋮
	928	3A0000h	3A0FFFh
57	927	39F000h	39FFFFh
	⋮	⋮	⋮
	912	390000h	390FFFh
56	911	38F000h	38FFFFh
	⋮	⋮	⋮
	896	380000h	380FFFh
55	895	37F000h	37FFFFh
	⋮	⋮	⋮
	880	370000h	370FFFh
54	879	36F000h	36FFFFh
	⋮	⋮	⋮
	864	360000h	360FFFh
53	863	35F000h	35FFFFh
	⋮	⋮	⋮
	848	350000h	350FFFh
52	847	34F000h	34FFFFh
	⋮	⋮	⋮
	832	340000h	340FFFh
51	831	33F000h	33FFFFh
	⋮	⋮	⋮
	816	330000h	330FFFh
50	815	32F000h	32FFFFh
	⋮	⋮	⋮
	800	320000h	320FFFh
49	799	31F000h	31FFFFh
	⋮	⋮	⋮
	784	310000h	310FFFh
48	783	30F000h	30FFFFh
	⋮	⋮	⋮
	768	300000h	300FFFh

Block	Sector	Address range	
47	767	2FF000h	2FFFFFh
	⋮	⋮	⋮
	752	2F0000h	2F0FFFh
46	751	2EF000h	2EFFFFh
	⋮	⋮	⋮
	736	2E0000h	2E0FFFh
45	735	2DF000h	2DFFFFh
	⋮	⋮	⋮
	720	2D0000h	2D0FFFh
44	719	2CF000h	2CFFFFh
	⋮	⋮	⋮
	704	2C0000h	2C0FFFh
43	703	2BF000h	2BFFFFh
	⋮	⋮	⋮
	688	2B0000h	2B0FFFh
42	687	2AF000h	2AFFFFh
	⋮	⋮	⋮
	672	2A0000h	2A0FFFh
41	671	29F000h	29FFFFh
	⋮	⋮	⋮
	656	290000h	290FFFh
40	655	28F000h	28FFFFh
	⋮	⋮	⋮
	640	280000h	280FFFh
39	639	27F000h	27FFFFh
	⋮	⋮	⋮
	624	270000h	270FFFh
38	623	26F000h	26FFFFh
	⋮	⋮	⋮
	608	260000h	260FFFh
37	607	25F000h	25FFFFh
	⋮	⋮	⋮
	592	250000h	250FFFh
36	591	24F000h	24FFFFh
	⋮	⋮	⋮
	576	240000h	240FFFh
35	575	23F000h	23FFFFh
	⋮	⋮	⋮
	560	230000h	230FFFh
34	559	22F000h	22FFFFh
	⋮	⋮	⋮
	544	220000h	220FFFh
33	543	21F000h	21FFFFh
	⋮	⋮	⋮
	528	210000h	210FFFh
32	527	20F000h	20FFFFh
	⋮	⋮	⋮
	512	200000h	200FFFh

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

Block	Sector	Address range	
31	511	1FF000h	1FFFFFh
	⋮	⋮	⋮
	496	1F0000h	1F0FFFh
30	495	1EF000h	1EFFFFh
	⋮	⋮	⋮
	480	1E0000h	1E0FFFh
29	479	1DF000h	1DFFFFh
	⋮	⋮	⋮
	464	1D0000h	1D0FFFh
28	463	1CF000h	1CFFFFh
	⋮	⋮	⋮
	448	1C0000h	1C0FFFh
27	447	1BF000h	1BFFFFh
	⋮	⋮	⋮
	432	1B0000h	1B0FFFh
26	431	1AF000h	1AFFFFh
	⋮	⋮	⋮
	416	1A0000h	1A0FFFh
25	415	19F000h	19FFFF
	⋮	⋮	⋮
	400	190000h	190FFFh
24	399	18F000h	18FFFFh
	⋮	⋮	⋮
	384	180000h	180FFFh
23	383	17F000h	17FFFFh
	⋮	⋮	⋮
	368	170000h	170FFFh
22	367	16F000h	16FFFFh
	⋮	⋮	⋮
	352	160000	160FFFh
21	351	15F000	15FFFFh
	⋮	⋮	⋮
	336	150000h	150FFFh
20	335	14F000h	14FFFFh
	⋮	⋮	⋮
	320	140000h	140FFFh
19	319	13F000h	13FFFFh
	⋮	⋮	⋮
	304	130000h	130FFFh
18	303	12F000h	12FFFFh
	⋮	⋮	⋮
	288	120000h	120FFFh
17	287	11F000h	11FFFFh
	⋮	⋮	⋮
	272	110000h	110FFFh
16	271	10F000h	10FFFFh
	⋮	⋮	⋮
	256	100000h	100FFFh

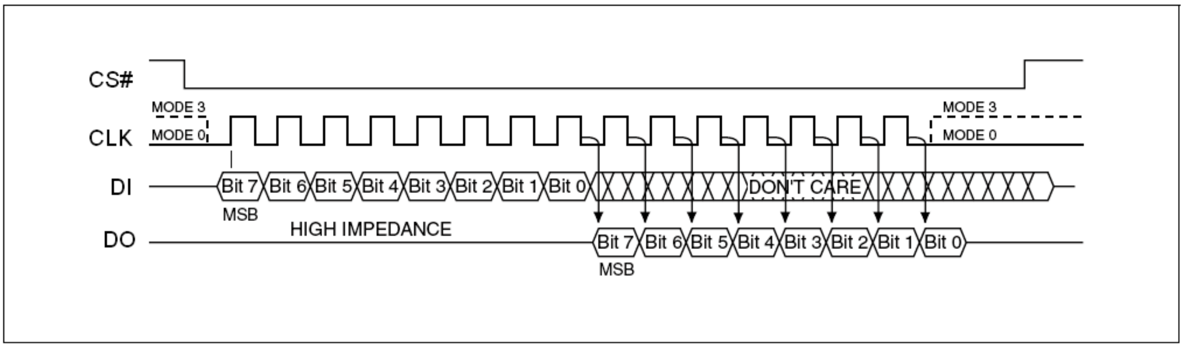
Block	Sector	Address range	
15	255	0FF000h	0FFFFFh
	⋮	⋮	⋮
	240	0F0000h	0F0FFFh
14	239	0EF000h	0EFFFFh
	⋮	⋮	⋮
	224	0E0000h	0E0FFFh
13	223	0DF000h	0DFFFFh
	⋮	⋮	⋮
	208	0D0000h	0D0FFFh
12	207	0CF000h	0CFFFFh
	⋮	⋮	⋮
	192	0C0000h	0C0FFFh
11	191	0BF000h	0BFFFFh
	⋮	⋮	⋮
	176	0B0000h	0B0FFFh
10	175	0AF000h	0AFFFFh
	⋮	⋮	⋮
	160	0A0000h	0A0FFFh
9	159	09F000h	09FFFFh
	⋮	⋮	⋮
	144	090000h	090FFFh
8	143	08F000h	08FFFFh
	⋮	⋮	⋮
	128	080000h	080FFFh
7	127	07F000h	07FFFFh
	⋮	⋮	⋮
	112	070000h	070FFFh
6	111	06F000h	06FFFFh
	⋮	⋮	⋮
	96	060000h	060FFFh
5	95	05F000h	05FFFFh
	⋮	⋮	⋮
	80	050000h	050FFFh
4	79	04F000h	04FFFFh
	⋮	⋮	⋮
	64	040000h	040FFFh
3	63	03F000h	03FFFFh
	⋮	⋮	⋮
	48	030000h	030FFFh
2	47	02F000h	02FFFFh
	⋮	⋮	⋮
	32	020000h	020FFFh
1	31	01F000h	01FFFFh
	⋮	⋮	⋮
	16	010000h	010FFFh
0	15	00F000h	00FFFFh
	⋮	⋮	⋮
	4	004000h	004FFFh
	3	003000h	003FFFh
	2	002000h	002FFFh
	1	001000h	001FFFh
	0	000000h	000FFFh

OPERATING FEATURES

Standard SPI Modes

The EN25S64 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 Figure 3, concerns the normal state of the CLK signal when the SPI bus master is in standby and data is not being transferred to the Serial Flash. For Mode 0 the CLK signal is normally low. For Mode 3 the CLK signal is normally high. In either case data input on the DI pin is sampled on the rising edge of the CLK. Data output on the DO pin is clocked out on the falling edge of CLK.

Figure 3. SPI Modes



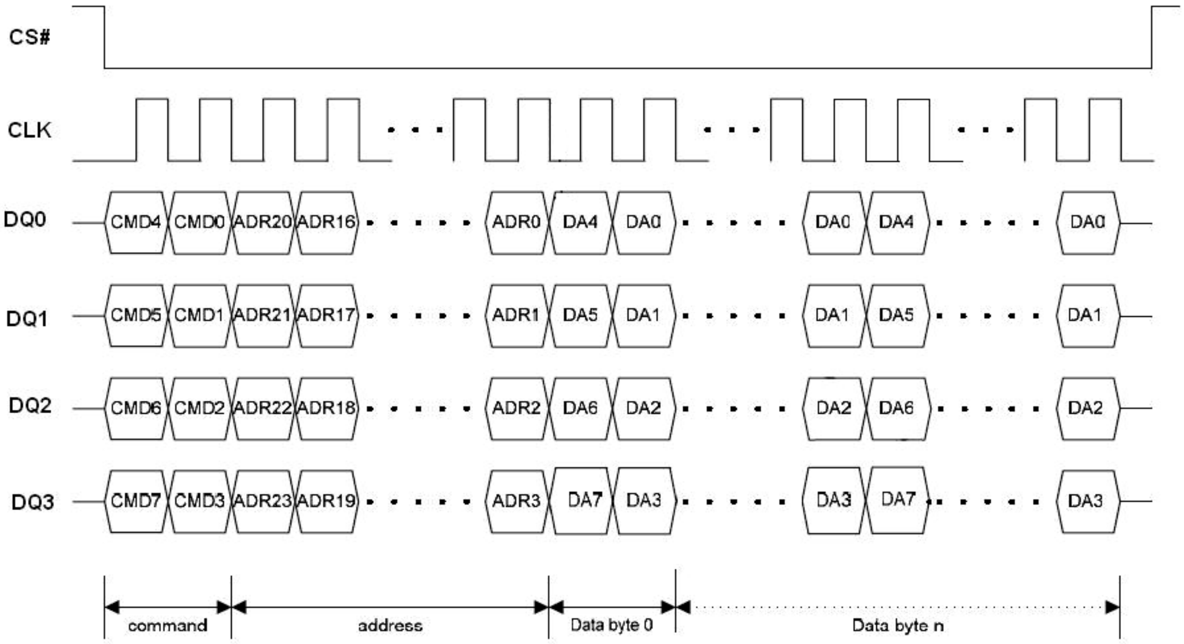
Dual SPI Instruction

The EN25S64 supports Dual SPI operation when using the “ Dual Output Fast Read and Dual I/ O FAST_READ “ (3Bh and BBh) instructions. These instructions allow data to be transferred to or from the Serial Flash memory at two to three times the rate possible with the standard SPI. The Dual Read instructions are ideal for quickly downloading code from Flash to RAM upon power-up (code-shadowing) or for application that cache code-segments to RAM for execution. The Dual output feature simply allows the SPI input pin to also serve as an output during this instruction. When using Dual SPI instructions the DI and DO pins become bidirectional I/O pins; DQ_0 and DQ_1. All other operations use the standard SPI interface with single output signal.

Quad SPI Instruction

The EN25S64 supports Quad output operation when using the Quad I/O Fast Read (EBh).This instruction allows data to be transferred to or from the Serial Flash memory at four to six times the rate possible with the standard SPI. The Quad Read instruction offer a significant improvement in continuous and random access transfer rates allowing fast code-shadowing to RAM or for application that cache code-segments to RAM for execution. The EN25S64 also supports full Quad Mode function while using the Enable Quad Peripheral Interface mode (EQPI) (38h). When using Quad SPI instruction the DI and DO pins become bidirectional I/O pins; DQ_0 and $DQ_{1,}$ and the WP# and NC pins become DQ_2 and DQ_3 respectively.

Figure 4. Quad SPI Modes



Page Programming

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

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

Sector Erase, Block Erase and Chip Erase

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

Polling During a Write, Program or Erase Cycle

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

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

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

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

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

Status Register and Suspend Status Register

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

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

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

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

WPDIS bit. The Write Protect disable (WPDIS) bit, non-volatile bit, when it is reset to “0” (factory default) to enable WP# function or is set to “1” to disable WP# function (can be floating during SPI mode.)

SRP bit / OTP_LOCK bit The Status Register Protect (SRP) bit operates in conjunction with the Write Protect (WP#) signal. The Status Register Protect (SRP) bit and Write Protect (WP#) signal allow the device to be put in the Hardware Protected mode. In this mode, the non-volatile bits of the Status Register (SRP, BP3, BP2, BP1, BP0) become read-only bits.

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

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

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

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

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 EN25S64 provides the following data protection mechanisms:

- Power-On Reset and an internal timer (t_{PUW}) can provide protection against inadvertent changes while the power supply is outside the operating specification.
- Program, Erase and Write Status Register instructions are checked that they consist of a number of clock pulses that is a multiple of eight, before they are accepted for execution.
- All instructions that modify data must be preceded by a Write Enable (WREN) instruction to set the Write Enable Latch (WEL) bit. This bit is returned to its reset state by the following events:
  - Power-up
  - Write Disable (WRDI) instruction completion or Write Status Register (WRSR) instruction completion or Page Program (PP) instruction completion or Sector Erase (SE) instruction completion or Block Erase (BE) instruction completion or Chip Erase (CE) instruction completion
- The Block Protect (BP3, BP2, BP1, BP0) bits allow part of the memory to be configured as read-only. This is the Software Protected Mode (SPM).
- The Write Protect (WP#) signal allows the Block Protect (BP3, BP2, BP1, BP0) bits and Status Register Protect (SRP) bit to be protected. This is the Hardware Protected Mode (HPM).
- In addition to the low power consumption feature, the Deep Power-down mode offers extra software protection from inadvertent Write, Program and Erase instructions, as all instructions are ignored except one particular instruction (the Release from Deep Power-down instruction).

Table 3. Protected Area Sizes Sector Organization

Status Register Content				Memory Content			
BP3 Bit	**BP2 Bit**	**BP1 Bit**	**BP0 Bit**	**Protect Areas**	**Addresses**	**Density(KB)**	**Portion**
0	0	0	0	None	None	None	None
0	0	0	1	Block 0 to 125	000000h-7DFFFFh	8064KB	Lower 126/128
0	0	1	0	Block 0 to 123	000000h-7BFFFFh	7936KB	Lower 124/128
0	0	1	1	Block 0 to 119	000000h-77FFFFh	7680KB	Lower 120/128
0	1	0	0	Block 0 to 111	000000h-6FFFFFh	7168KB	Lower 112/128
0	1	0	1	Block 0 to 95	000000h-5FFFFFh	6144KB	Lower 96/128
0	1	1	0	Block 0 to 63	000000h-3FFFFFh	4096KB	Lower 64/128
0	1	1	1	All	000000h-7FFFFFh	8192KB	All
1	0	0	0	None	None	None	None
1	0	0	1	Block 126 to 127	7E0000h-7FFFFFh	128KB	Upper 2/128
1	0	1	0	Block 124 to 127	7C0000h-7FFFFFh	256KB	Upper 4/128
1	0	1	1	Block 120 to 127	780000h-7FFFFFh	512KB	Upper 8/128
1	1	0	0	Block 112 to 127	700000h-7FFFFFh	1024KB	Upper 16/128
1	1	0	1	Block 96 to 127	600000h-7FFFFFh	2048KB	Upper 32/128
1	1	1	0	Block 64 to 127	400000h-7FFFFFh	4096KB	Upper 64/128
1	1	1	1	All	000000h-7FFFFFh	8192KB	All

INSTRUCTIONS

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

The instruction set is listed in Table 4. Every instruction sequence starts with a one-byte instruction code. Depending on the instruction, this might be followed by address bytes, or by data bytes, or by both or none. Chip Select (CS#) must be driven High after the last bit of the instruction sequence has been shifted in. In the case of a Read Data Bytes (READ), Read Data Bytes at Higher Speed (Fast_Read), Dual Output Fast Read (3Bh), Dual I/O Fast Read (BBh), Quad Input/Output FAST_READ (EBh), Read Status Register (RDSR), Read Suspend Status Register (RDSSR) or Release from Deep Power-down, and Read Device ID (RDI) instruction, the shifted-in instruction sequence is followed by a data-out sequence. Chip Select (CS#) can be driven High after any bit of the data-out sequence is being shifted out.

In the case of a Page Program (PP), Sector Erase (SE), Block Erase (BE), Chip Erase (CE), Write Status Register (WRSR), Write Enable (WREN), Write Disable (WRDI) or Deep Power-down (DP) instruction, Chip Select (CS#) must be driven High exactly at a byte boundary, otherwise the instruction is rejected, and is not executed. That is, Chip Select (CS#) must driven High when the number of clock pulses after Chip Select (CS#) being driven Low is an exact multiple of eight. For Page Program, if at any time the input byte is not a full byte, nothing will happen and WEL will not be reset.

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

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

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

Table 4A. Instruction Set

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
RSTEN	66h						
RST(1)	99h						
EQPI	38h						
RSTQIO(2) Release Quad I/O or Fast Read Enhanced Mode	FFh						
Write Enable	06h						
Write Disable / Exit OTP mode	04h						
Read Status Register	05h	(S7-S0)(3)					continuous(4)
Read Suspend Status Register	09h	(S7-S0)(3)					continuous(4)
Write Status Register	01h	S7-S0					
Page Program	02h	A23-A16	A15-A8	A7-A0	D7-D0	Next byte	continuous
Write Suspend	B0h						
Write Resume	30h						
Sector Erase / OTP erase	20h	A23-A16	A15-A8	A7-A0			
Block Erase	D8h	A23-A16	A15-A8	A7-A0			
Chip Erase	C7h/ 60h						
Deep Power-down	B9h						
Release from Deep Power-down, and read Device ID	ABh	dummy	dummy	dummy	(ID7-ID0)		(5)
Release from Deep Power-down							
Manufacturer/ Device ID	90h	dummy	dummy	00h	(M7-M0)	(ID7-ID0)	(6)
				01h	(ID7-ID0)	(M7-M0)	
Read Identification	9Fh	(M7-M0)	(ID15-ID8)	(ID7-ID0)	(7)		
Enter OTP mode	3Ah						

Notes:

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

Table 4B. Instruction Set (Read Instruction)

Instruction Name	Byte 1 Code	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
Read Data	03h	A23-A16	A15-A8	A7-A0	(D7-D0)	(Next byte)	continuous
Fast Read	0Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(Next Byte) continuous
Dual Output Fast Read	3Bh	A23-A16	A15-A8	A7-A0	dummy	(D7-D0, …) (1)	(one byte per 4 clocks, continuous)
Dual I/O Fast Read	BBh	A23-A8(2)	A7-A0, dummy (2)	(D7-D0, …) (1)			(one byte per 4 clocks, continuous)
Quad I/O Fast Read	EBh	A23-A0, dummy (4)	(dummy, D7-D0) (5)	(D7-D0, …) (3)			(one byte per 2 clocks, continuous)
Set Burst	C0h, (D7-D0) (6)						(one byte per 2 clocks, continuous)
Read Burst with wrap	0Ch, A23-A0,	dummy (7), (D7-D0) (3)	(D7-D0, …) (3)				(one byte per 2 clocks, continuous)
Fast Read Burst with wrap	0Dh, A23-A0,	dummy (8),	(D7-D0, …) (3)				(one byte per 2 clocks, continuous)

Notes:

1. Dual Output data
   DQ_0 = (D6, D4, D2, D0)
   DQ_1 = (D7, D5, D3, D1)
2. Dual Input Address
   DQ_0 = A22, A20, A18, A16, A14, A12, A10, A8 ; A6, A4, A2, A0, dummy 6, dummy 4, dummy 2, dummy 0
   DQ_1 = A23, A21, A19, A17, A15, A13, A11, A9 ; A7, A5, A3, A1, dummy 7, dummy 5, dummy 3, dummy 1
3. Quad Data
   DQ_0 = (D4, D0, ……)
   DQ_1 = (D5, D1, ……)
   DQ_2 = (D6, D2, …...)
   DQ_3 = (D7, D3, …...)
4. Quad Input Address
   DQ_0 = A20, A16, A12, A8, A4, A0, dummy 4, dummy 0
   DQ_1 = A21, A17, A13, A9, A5, A1, dummy 5, dummy 1
   DQ_2 = A22, A18, A14, A10, A6, A2, dummy 6, dummy 2
   DQ_3 = A23, A19, A15, A11, A7, A3, dummy 7, dummy 3
5. Quad I/O Fast Read Data
   DQ_0 = (dummy 12, dummy 8, dummy 4, dummy 0, D4, D0)
   DQ_1 = (dummy 13, dummy 9, dummy 5, dummy 1, D5, D1)
   DQ_2 = (dummy 14, dummy 10, dummy 6, dummy 2, D6, D2)
   DQ_3 = (dummy 15, dummy 11, dummy 7, dummy 3, D7, D3)
6. Set burst and Wrap Length

Table 5. Burst length configuration table

Data to setup	Burst length	Burst wrap (A[7:A0]) address range
00h	8 Bytes (default)	00-07H, 08-0FH, 10-17H, 18-1FH...
01h	16 Bytes	00-0FH, 10-1FH, 20-2FH, 30-3FH...
02h	32 Bytes	00-1FH, 20-3FH, 40-5FH, 60-7FH...
03h	64 Bytes	00-3FH, 40-7FH, 80-BFH, C0-FFH
04h	128 Bytes	00-7FH, 80-FFH
05h	256 Bytes	00-FFH

If input data is not between 00h~05h or user does not Set Burst, the Burst length will be 8 Bytes in default.

7. Two dummy cycles (4 clocks) are necessary for Read Burst with Wrap mode.
8. Three dummy cycles (6 clocks) are necessary for Fast Read Burst with Wrap mode.

Table 6. Manufacturer and Device Identification

OP Code	(M7-M0)	(ID15-ID0)	(ID7-ID0)
ABh			76h
90h	1Ch		76h
9Fh	1Ch	3817h	

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 EN25S64 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 and the Suspend Status register to data = 00h, see Figure 5 for SPI Mode and Figure 5.1 for Quad Mode. A device reset during an active Program or Erase operation aborts the operation, which can cause the data of the targeted address range to be corrupted or lost. Depending on the prior operation, the reset timing may vary. Recovery from a Write operation requires more software latency time (t_{SR}) than recovery from other operations.

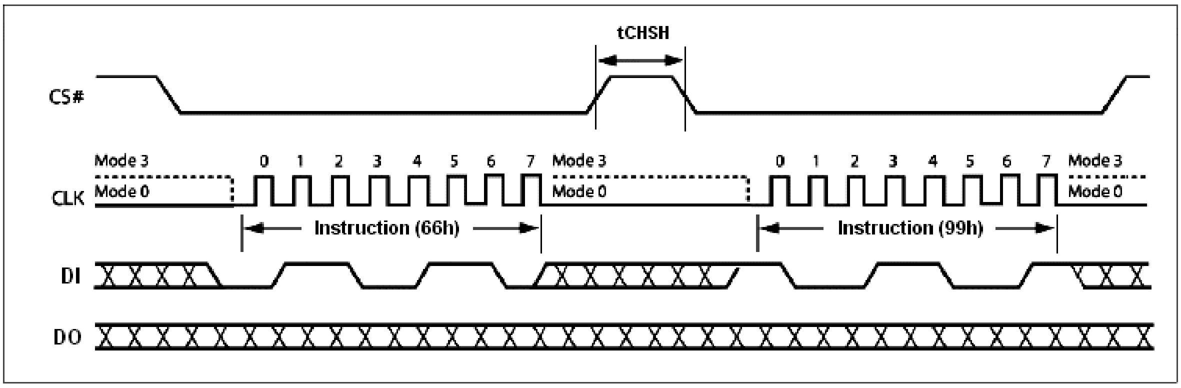


Figure 5. Reset-Enable and Reset Sequence Diagram

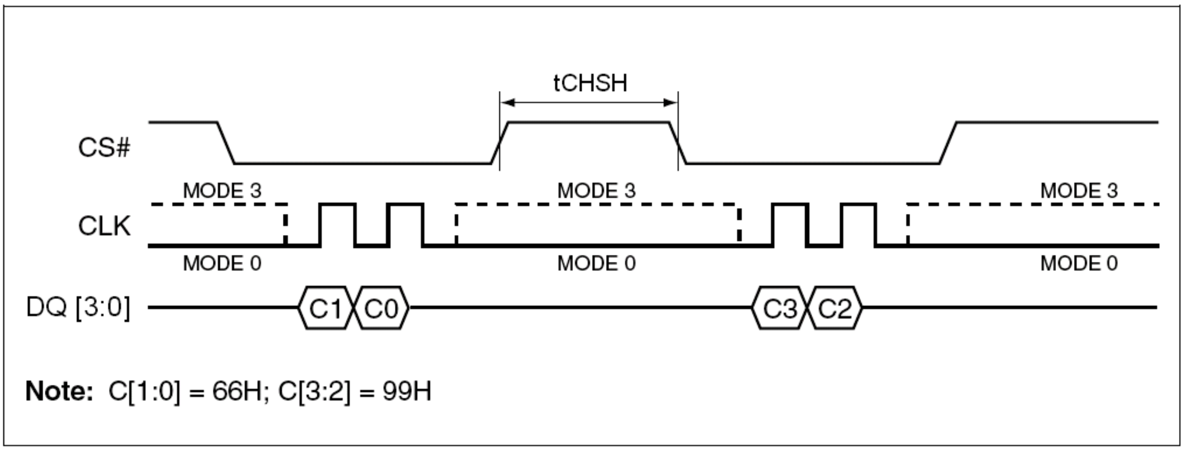
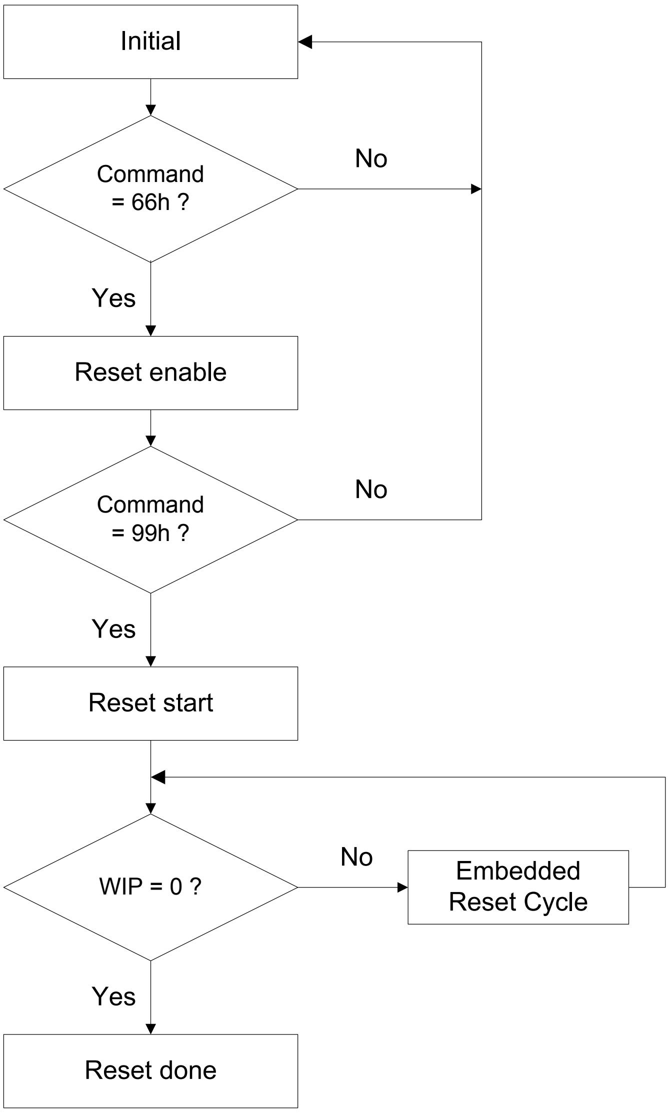


Figure 5.1 . Reset-Enable and Reset Sequence Diagram under EQPI Mode

Software Reset Flow



Note:

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

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

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

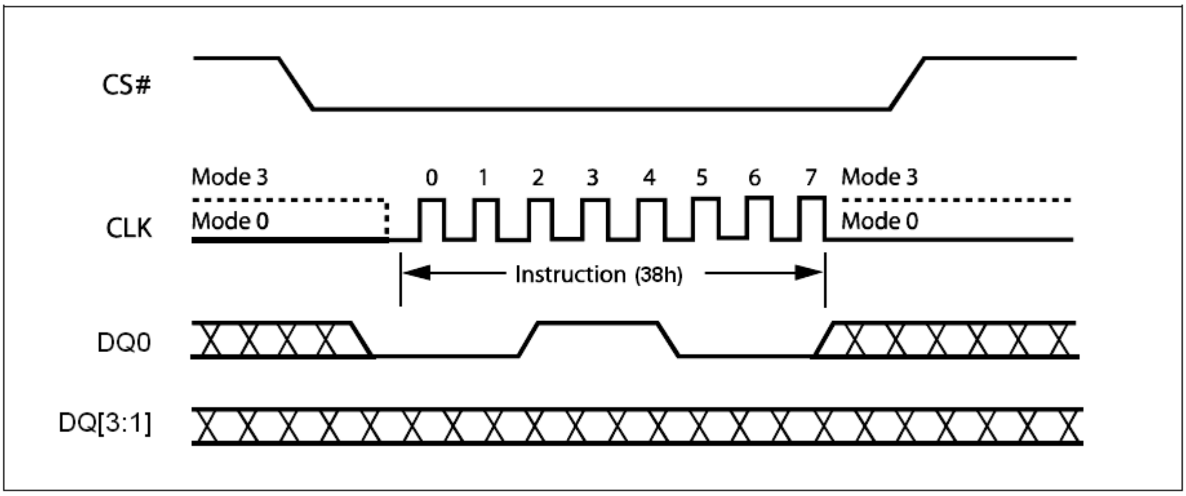


Figure 6. Enable Quad Peripheral Interface mode Sequence Diagram

Reset Quad I/O (RSTQIO) (FFh)

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

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

Note:

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

Write Enable (WREN) (06h)

The Write Enable (WREN) instruction (Figure 7) sets the Write Enable Latch (WEL) bit. The Write Enable Latch (WEL) bit must be set prior to every Page Program (PP), Sector Erase (SE), Block Erase (BE), Chip Erase (CE) and Write Status Register (WRSR) instruction.
The Write Enable (WREN) instruction is entered by driving Chip Select (CS#) Low, sending the instruction code, and then driving Chip Select (CS#) High.

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

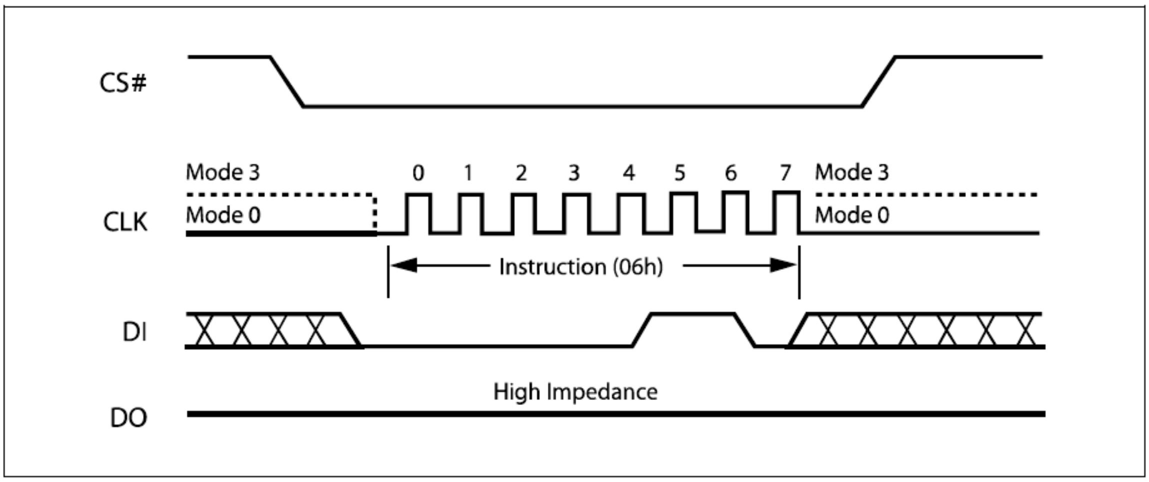


Figure 7. Write Enable Instruction Sequence Diagram

Write Disable (WRDI) (04h)

The Write Disable instruction (Figure 8) resets the Write Enable Latch (WEL) bit in the Status Register to a 0 or exit from OTP mode to normal mode. The Write Disable instruction is entered by driving Chip Select (CS#) low, shifting the instruction code “04h” into the DI pin and then driving Chip Select (CS#) high. Note that the WEL bit is automatically reset after Power-up and upon completion of the Write Status Register, Page Program, Sector Erase, Block Erase (BE) and Chip Erase instructions.

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

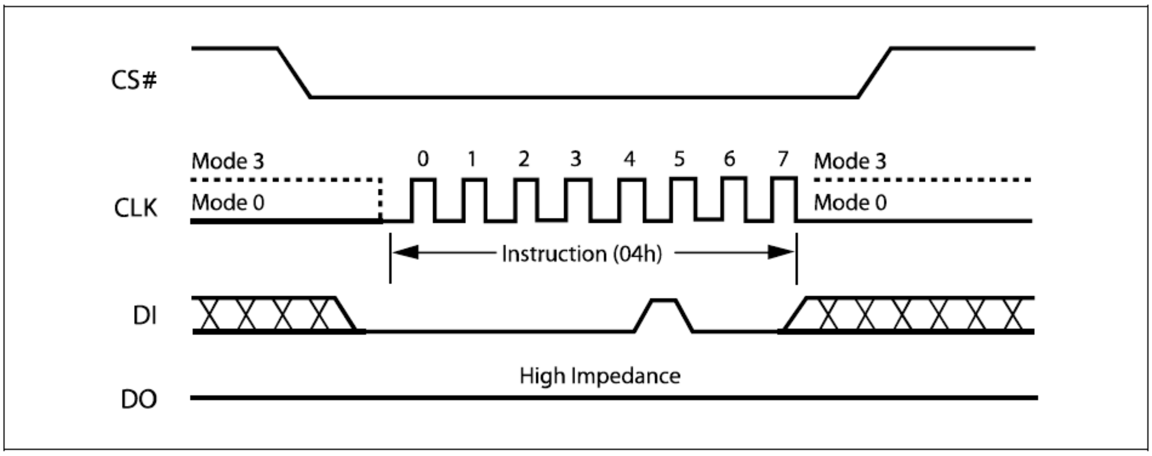


Figure 8. Write Disable Instruction Sequence Diagram

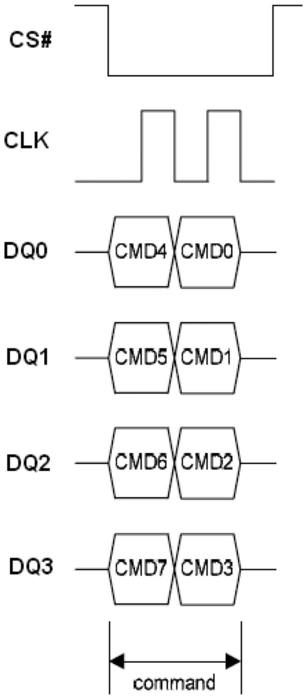


Figure 8.1 Write Enable/Disable Instruction Sequence under EQPI Mode

Read Status Register (RDSR) (05h)

The Read Status Register (RDSR) instruction allows the Status Register to be read. The Status Register may be read at any time, even while a Program, Erase or Write Status Register cycle is in progress. When one of these cycles is in progress, it is recommended to check the Write In Progress (WIP) bit before sending a new instruction to the device. It is also possible to read the Status Register continuously, as shown in Figure 9.

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

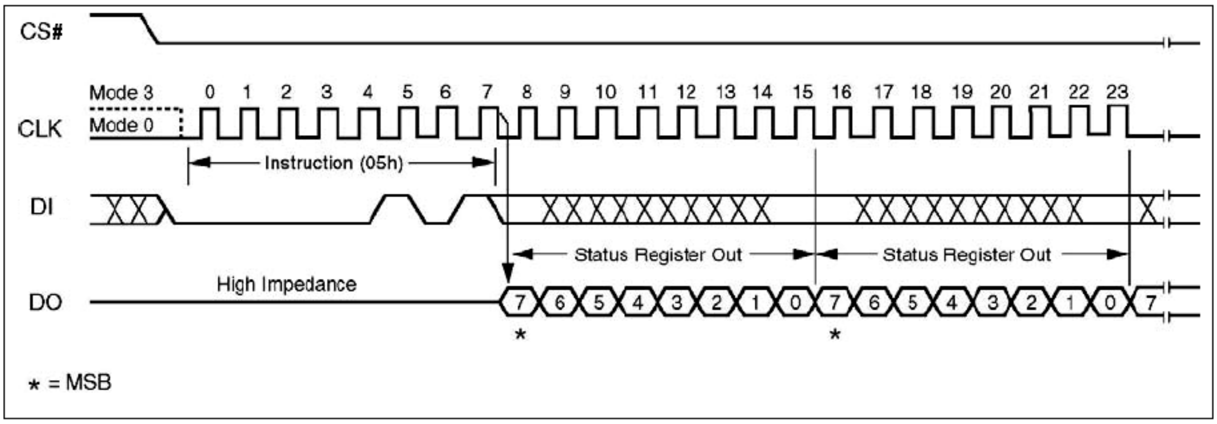


Figure 9. Read Status Register Instruction Sequence Diagram

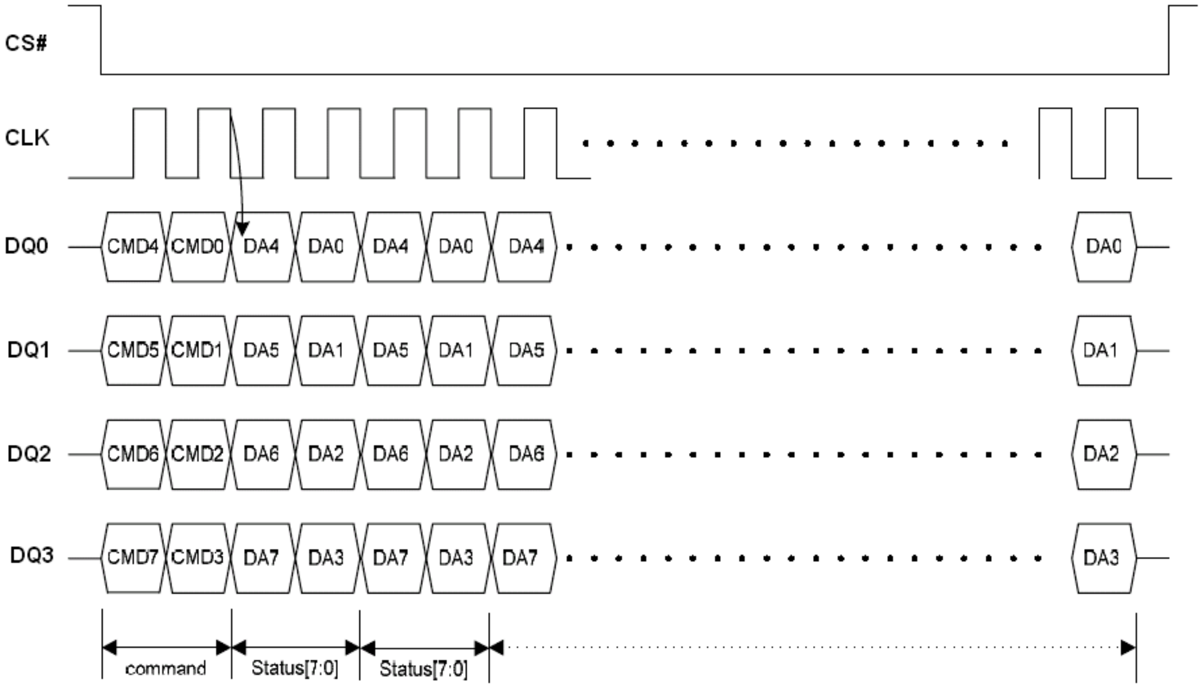


Figure 9.1 Read Status Register Instruction Sequence under EQPI Mode

Table 7. Status Register Bit Locations

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

Note

1. In OTP mode, SRP bit is served as OTP_LOCK bit.
2. See the table 3 “Protected Area Sizes Sector Organization”.
3. When executed the (RDSR) (05h) command, the WIP (S0) value is the same as WIP (S7) in table 8.

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

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

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

BP3, BP2, BP1, BP0 bits. The Block Protect (BP3, BP2, BP1, BP0) bits are non-volatile. They define the size of the area to be software protected against Program and Erase instructions. These bits are written with the Write Status Register (WRSR) instruction. When one or both of the Block Protect (BP3, BP2, BP1, BP0) bits is set to 1, the relevant memory area (as defined in Table 3.) becomes protected against Page Program (PP) Sector Erase (SE) and , Block Erase (BE), instructions. The Block Protect

(BP3, BP2, BP1, BP0) bits can be written and provided that the Hardware Protected mode has not been set. The Chip Erase (CE) instruction is executed if, and only if, all Block Protect (BP3, BP2, BP1, BP0) bits are 0.

WPDIS bit. The Write Protect disable (WPDIS) bit, non-volatile bit, when it is reset to “0” (factory default) to enable WP# function or is set to “1” to disable WP# function (can be floating during SPI mode.)

SRP bit / OTP_LOCK bit. The Status Register Protect (SRP) bit is operated in conjunction with the Write Protect (WP#) signal. The Status Register Write Protect (SRP) bit and Write Protect (WP#) signal allow the device to be put in the Hardware Protected mode (when the Status Register Protect (SRP) bit is set to 1, and Write Protect (WP#) is driven Low). In this mode, the non-volatile bits of the Status Register (SRP, BP3, BP2, BP1, BP0) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

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

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

Read Suspend Status Register (RDSSR) (09h)

The Read Suspend Status Register (RDSSR) instruction allows the Suspend Status Register to be read. The Suspend Status Register may be read at any time, even while a Write Suspend or Write Resume cycle is in progress. When one of these cycles is in progress, it is recommended to check the Write In Progress (WIP) bit before sending a new instruction to the device. It is also possible to read the Suspend Status Register continuously, as shown in Figure 10.

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

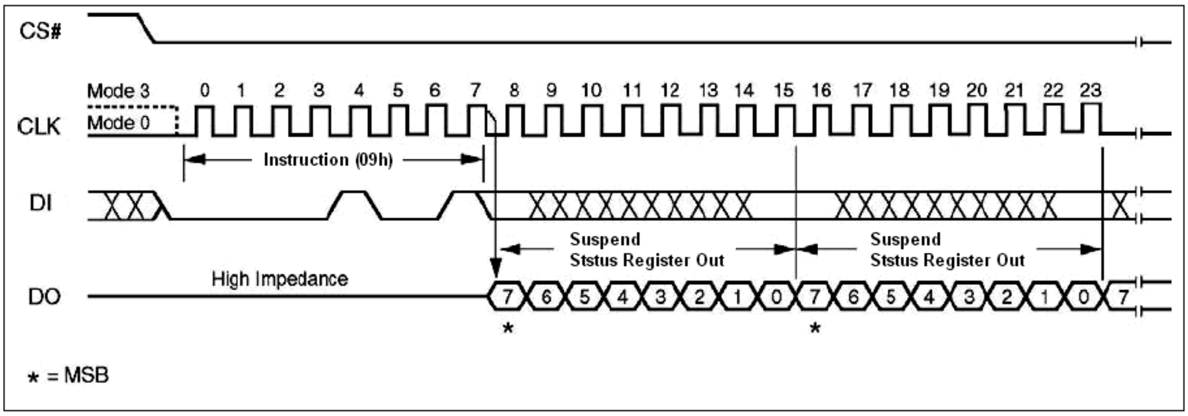


Figure 10. Read Suspend Status Register Instruction Sequence Diagram

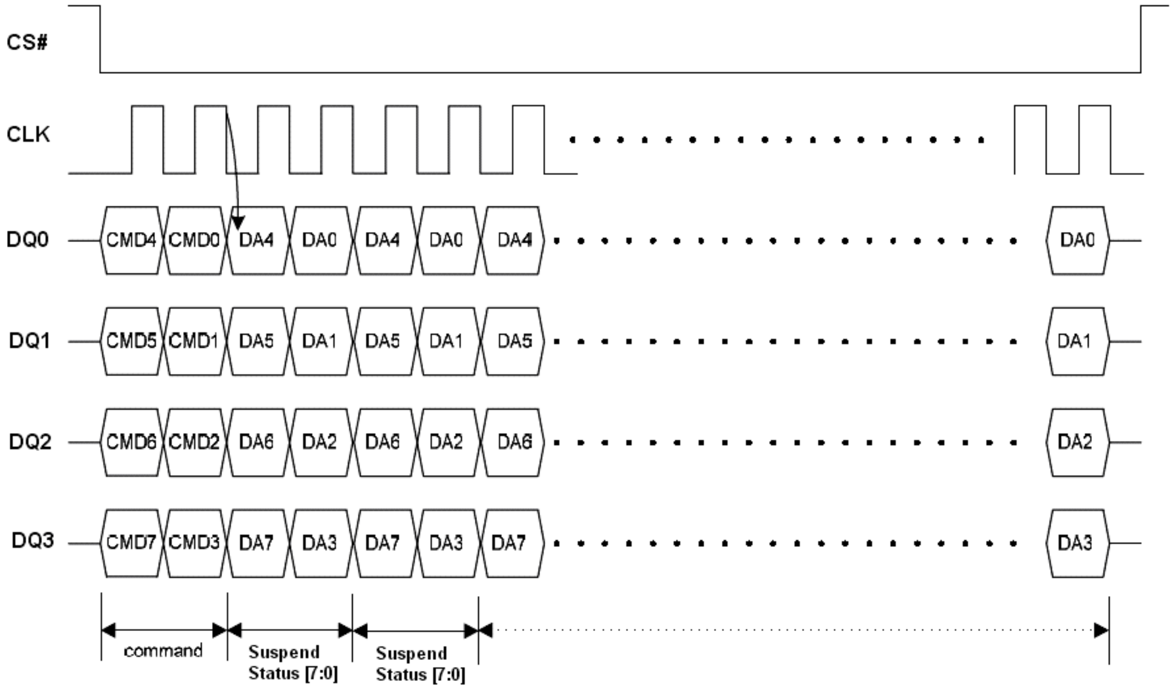


Figure 10.1 Read Suspend Status Register Instruction Sequence under EQPI Mode

Table 8. Suspend Status Register Bit Locations

S7	S6	S5	S4	S3	S2	S1	S0
WIP (Write In Progress bit) (Note 1)	Reserved bit	Fail bit index	Reserved bit	WSP (Write Suspend Program bits)	WSE (Write Suspend Erase status bit)	WEL (Write Enable Latch)	Reserved bit
1 = write operation 0 = not in write operation		1 = erase or program or WRSR failed 0 = passed		1 = Program suspended 0 = Program is not suspended	1 = Erase suspended 0 = Erase is not suspended	1 = write enable 0 = not write enable	
volatile bit		volatile bit		volatile bit	volatile bit	volatile bit	

Note:

1. When executed the (RDSSR) (09h) command, the WIP (S7) value is the same as WIP (S0) in table 7.
2. Default at Power-up is “0”

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

Reserved bit. Suspend Status register bit locations 0, 4 and 6 are reserved for future use. Current devices will read 0 for these bit locations. It is recommended to mask out the reserved bit when testing the Suspend Status Register. Doing this will ensure compatibility with future devices.

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 Suspend or Write Resume instruction is accepted.

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

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

Fail bit. The fail bit, volatile bit, it will latched high when erase or program or WRSR failed. It will be reset after new embedded program and erase cycle re-stared or power on or software reset.

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

Write Status Register (WRSR) (01h)

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

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

The instruction sequence is shown in Figure 11. The Write Status Register (WRSR) instruction has no effect on S1 and S0 of the Status Register. Chip Select (CS#) must be driven High after the eighth bit of the data byte has been latched in. If not, the Write Status Register (WRSR) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Write Status Register cycle (whose duration is t_W) is initiated. While the Write Status Register cycle is in progress, the Status Register may still be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Write Status Register cycle, and is 0 when it is completed. When the cycle is completed, the Write Enable Latch (WEL) is reset.

The Write Status Register (WRSR) instruction allows the user to change the values of the Block Protect (BP3, BP2, BP1, BP0) bits, to define the size of the area that is to be treated as read-only, as defined in Table 3. The Write Status Register (WRSR) instruction also allows the user to set or reset the Status Register Protect (SRP) bit in accordance with the Write Protect (WP#) signal. The Status Register Protect (SRP) bit and Write Protect (WP#) signal allow the device to be put in the Hardware Protected Mode (HPM). The Write Status Register (WRSR) instruction is not executed once the Hardware Protected Mode (HPM) is entered.

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

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

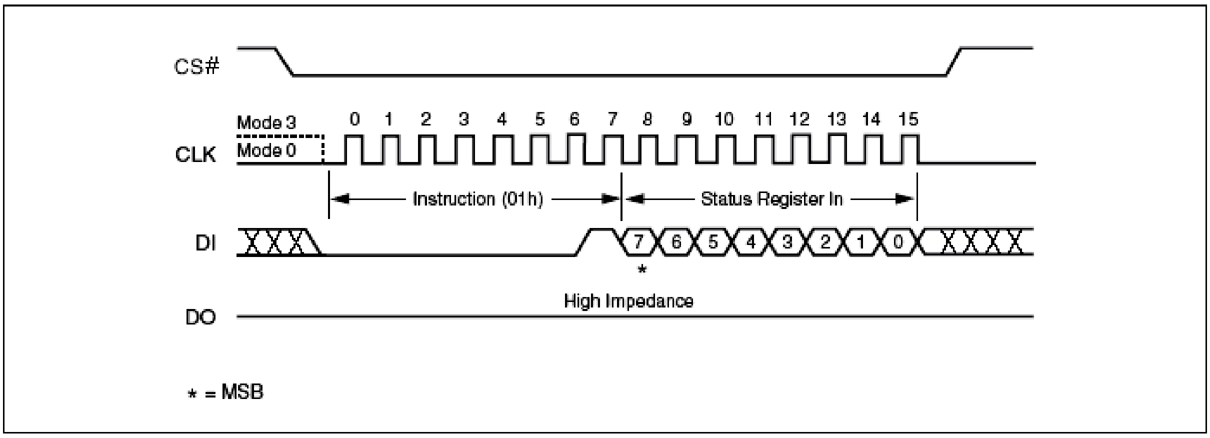


Figure 11. Write Status Register Instruction Sequence Diagram

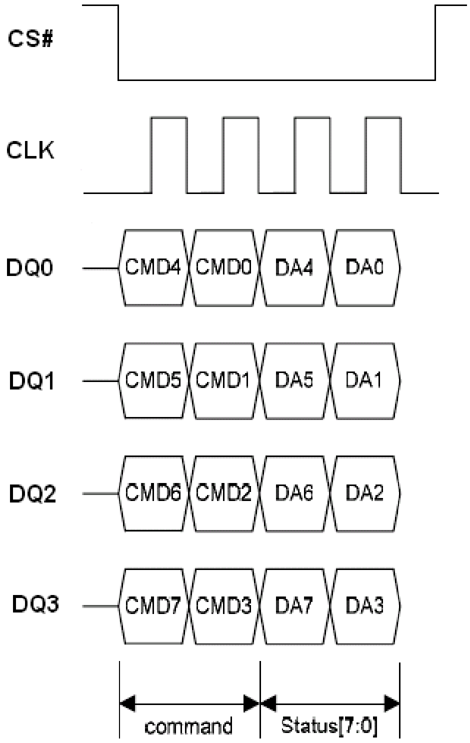


Figure 11.1 Write Status Register Instruction Sequence under EQPI Mode

Read Data Bytes (READ) (03h)

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

The instruction sequence is shown in Figure 12. The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single Read Data Bytes (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.

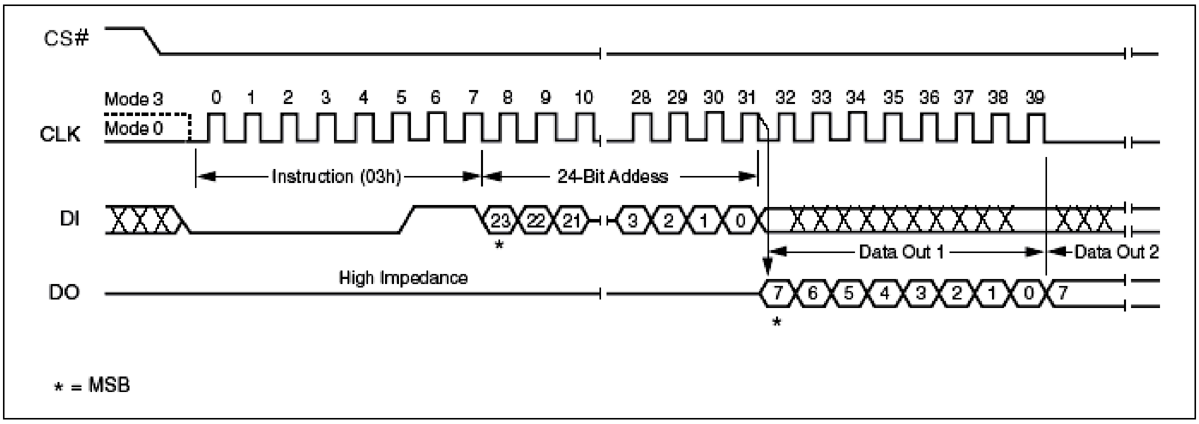


Figure 12. Read Data Instruction Sequence Diagram

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

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

The instruction sequence is shown in Figure 13. The first byte addressed can be at any location. The address is automatically incremented to the next higher address after each byte of data is shifted out. The whole memory can, therefore, be read with a single Read Data Bytes at Higher Speed (FAST_READ) instruction. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to be continued indefinitely.

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

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

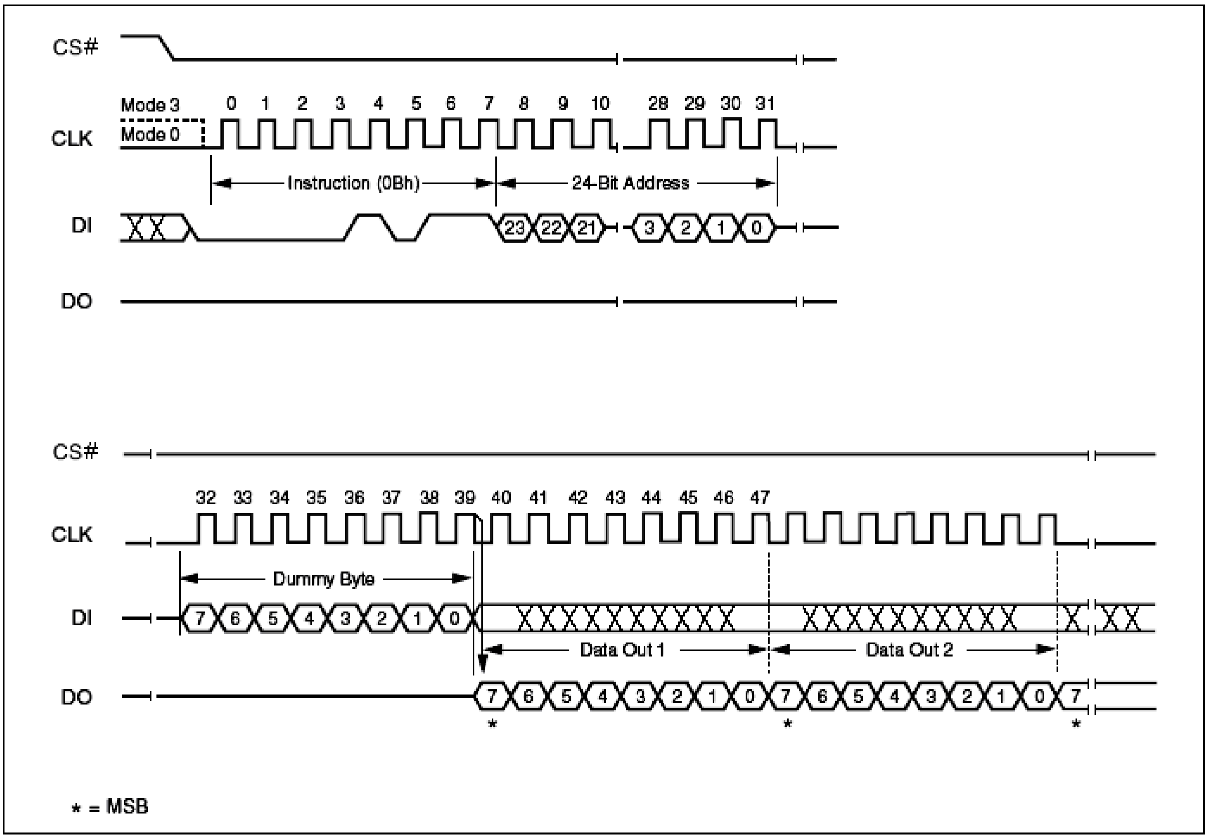


Figure 13. Fast Read Instruction Sequence Diagram

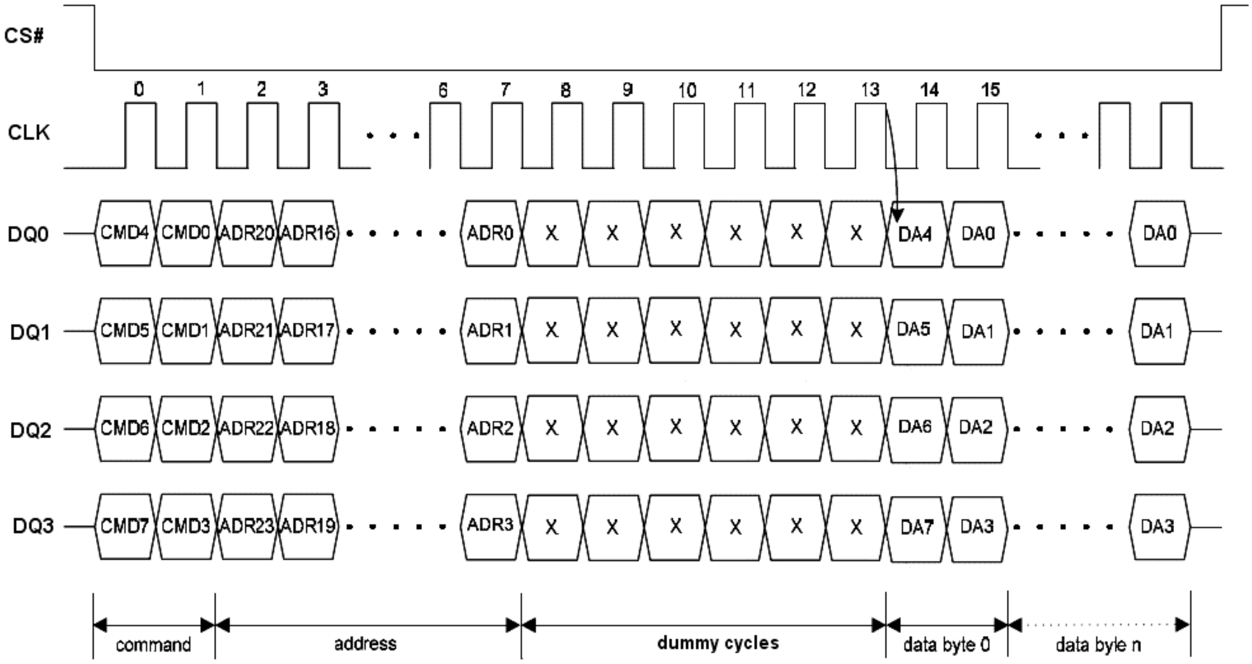


Figure 13.1 Fast Read Instruction Sequence under EQPI Mode

Dual Output Fast Read (3Bh)

The Dual Output Fast Read (3Bh) is similar to the standard Fast Read (0Bh) instruction except that data is output on two pins, DQ_0 and DQ_1, instead of just DQ_0. This allows data to be transferred from the EN25S64 at twice the rate of standard SPI devices. The Dual Output Fast Read instruction is ideal for quickly downloading code from to RAM upon power-up or for applications that cache code-segments to RAM for execution.

Similar to the Fast Read instruction, the Dual Output Fast Read instructions can operation at the highest possible frequency of FR (see AC Electrical Characteristics). This is accomplished by adding eight “dummy clocks after the 24-bit address as shown in Figure 14. 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.

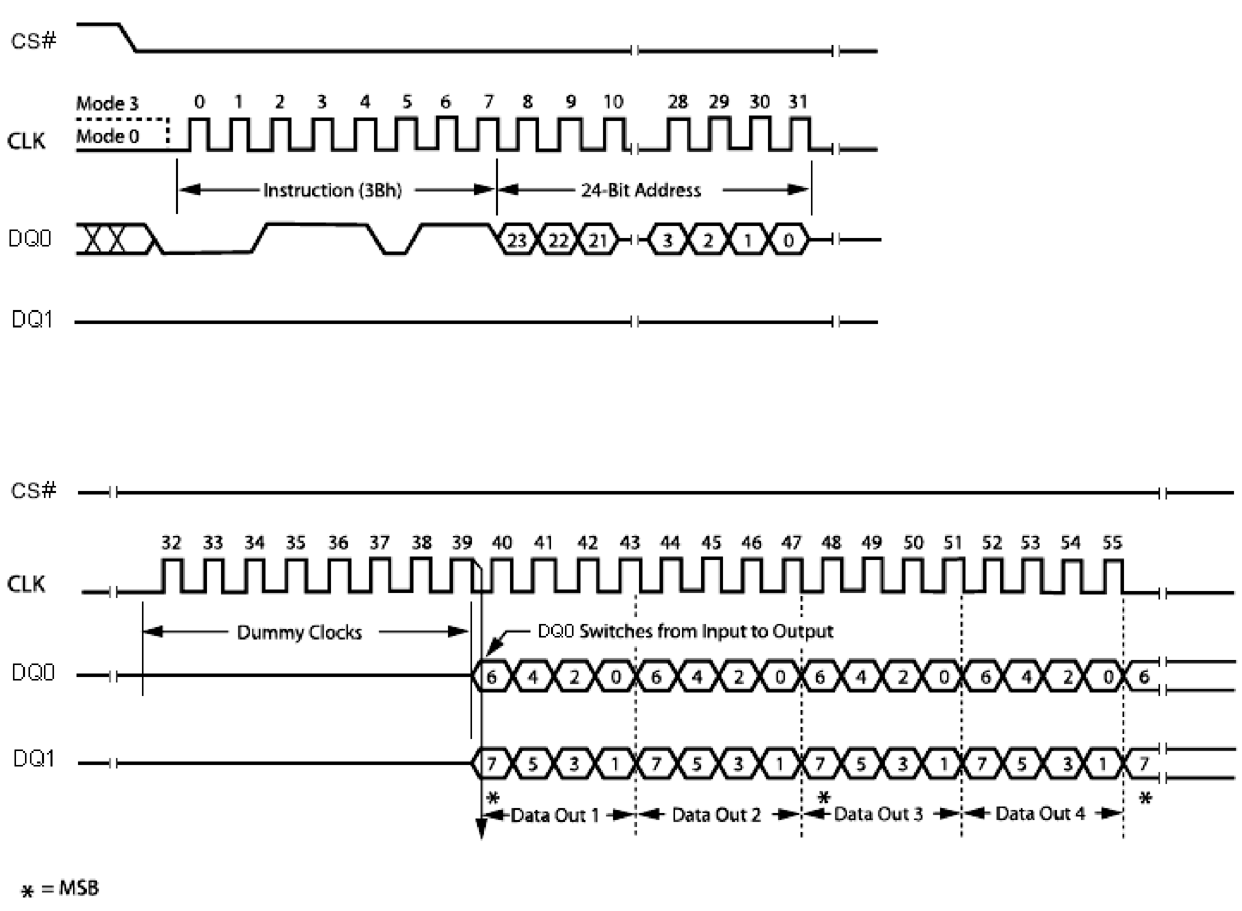


Figure 14. Dual Output Fast Read Instruction Sequence Diagram

Dual Input / Output FAST_READ (BBh)

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

The Dual I/O Fast Read instruction enable double throughput of Serial Flash in read mode. The address is latched on rising edge of CLK, and data of every two bits (interleave 2 I/O pins) shift out on the falling edge of CLK at a maximum frequency. The first address can be at any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out at a single Dual I/O Fast Read instruction. The address counter rolls over to 0 when the highest address has been reached. Once writing Dual I/O Fast Read instruction, the following address/dummy/data out will perform as 2-bit instead of previous 1-bit, as shown in Figure 15.

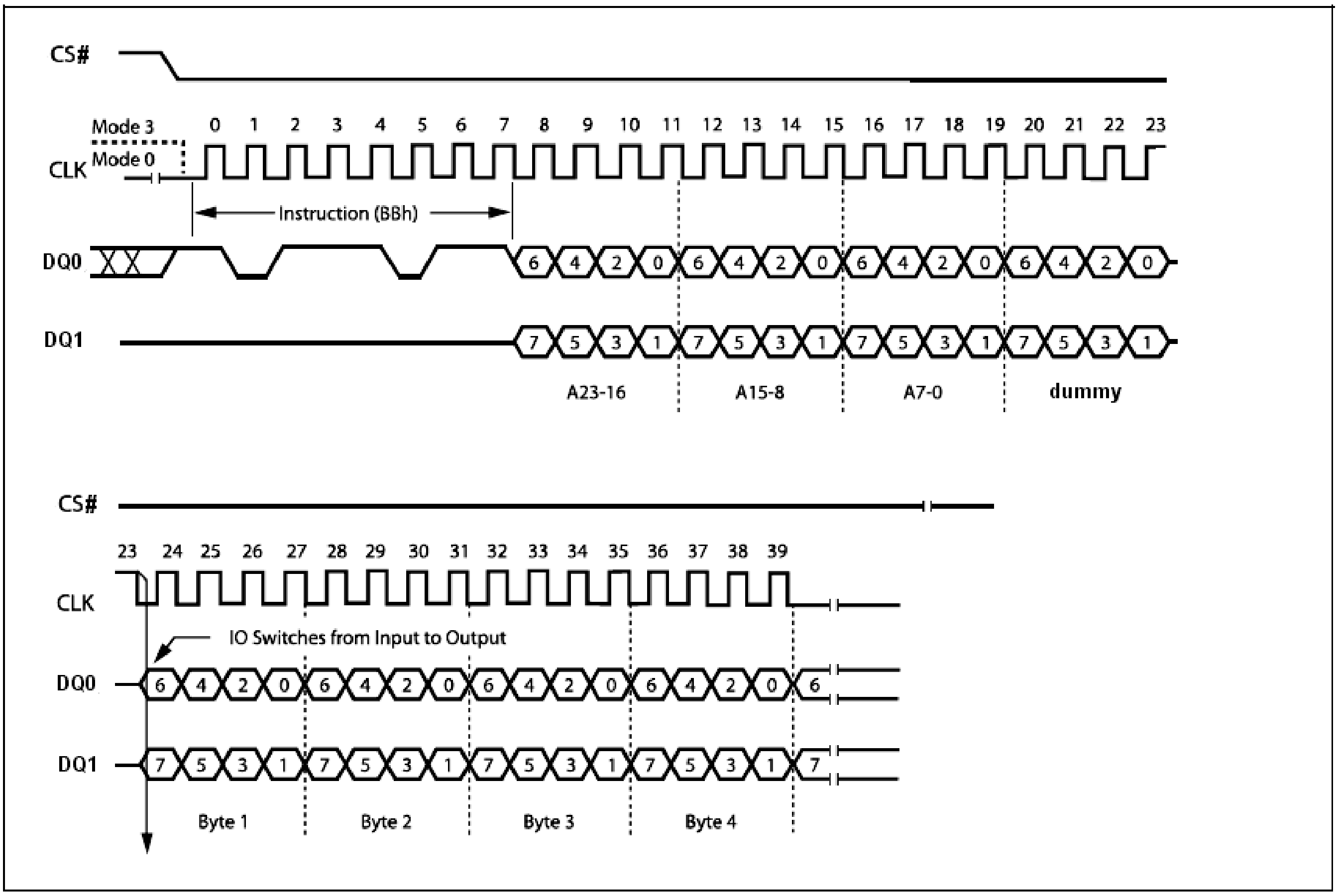


Figure 15. Dual Input / Output Fast Read Instruction Sequence Diagram

Quad Input / Output FAST_READ (EBh)

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

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

The sequence of issuing Quad Input/Output FAST_READ (EBh) instruction is: CS# goes low -> sending Quad Input/Output FAST_READ (EBh) instruction -> 24-bit address interleave on DQ_3, DQ_2, DQ_1 and DQ_0 -> 6 dummy clocks -> data out interleave on DQ_3, DQ_2, DQ_1 and DQ_0 -> to end Quad Input/Output FAST_READ (EBh) operation can use CS# to high at any time during data out, as shown in Figure 16.

The instruction sequence is shown in Figure 16.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

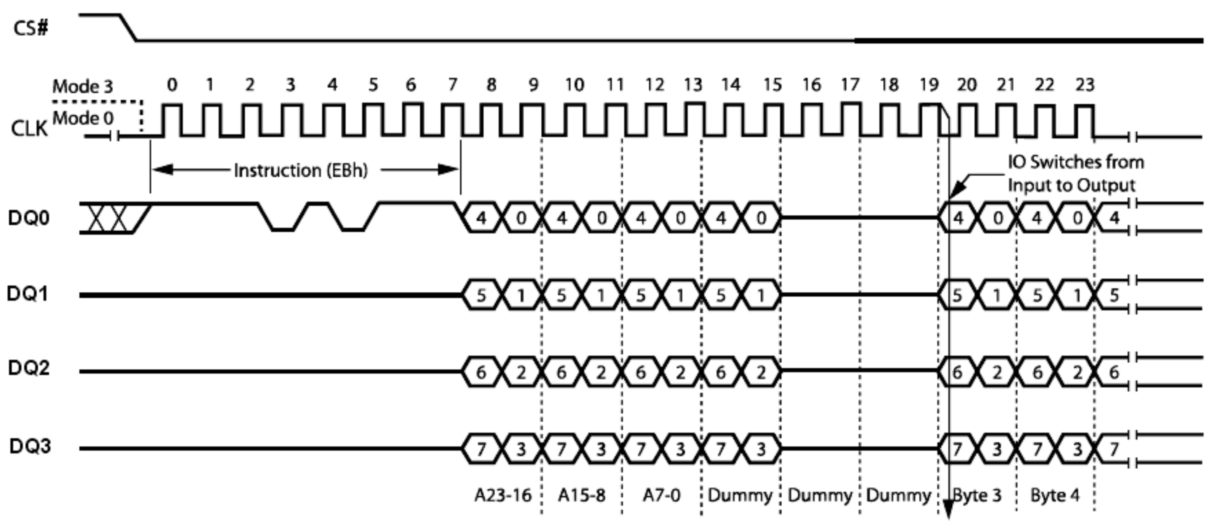


Figure 16. Quad Input / Output Fast Read Instruction Sequence Diagram

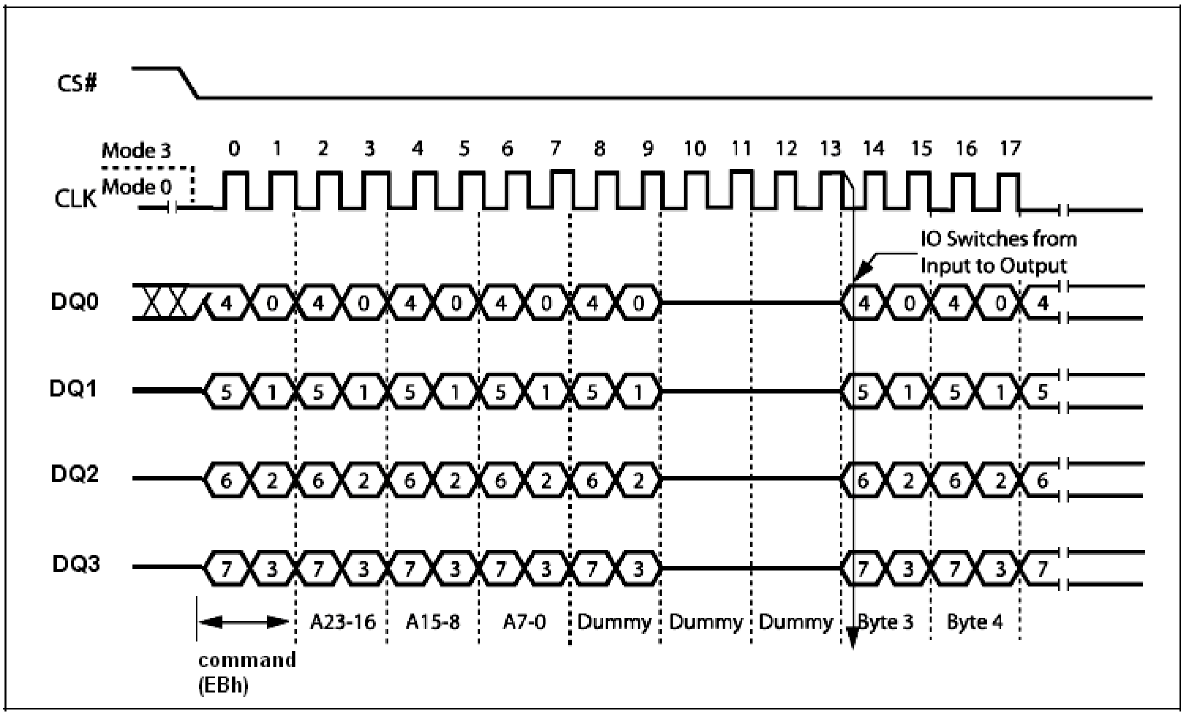


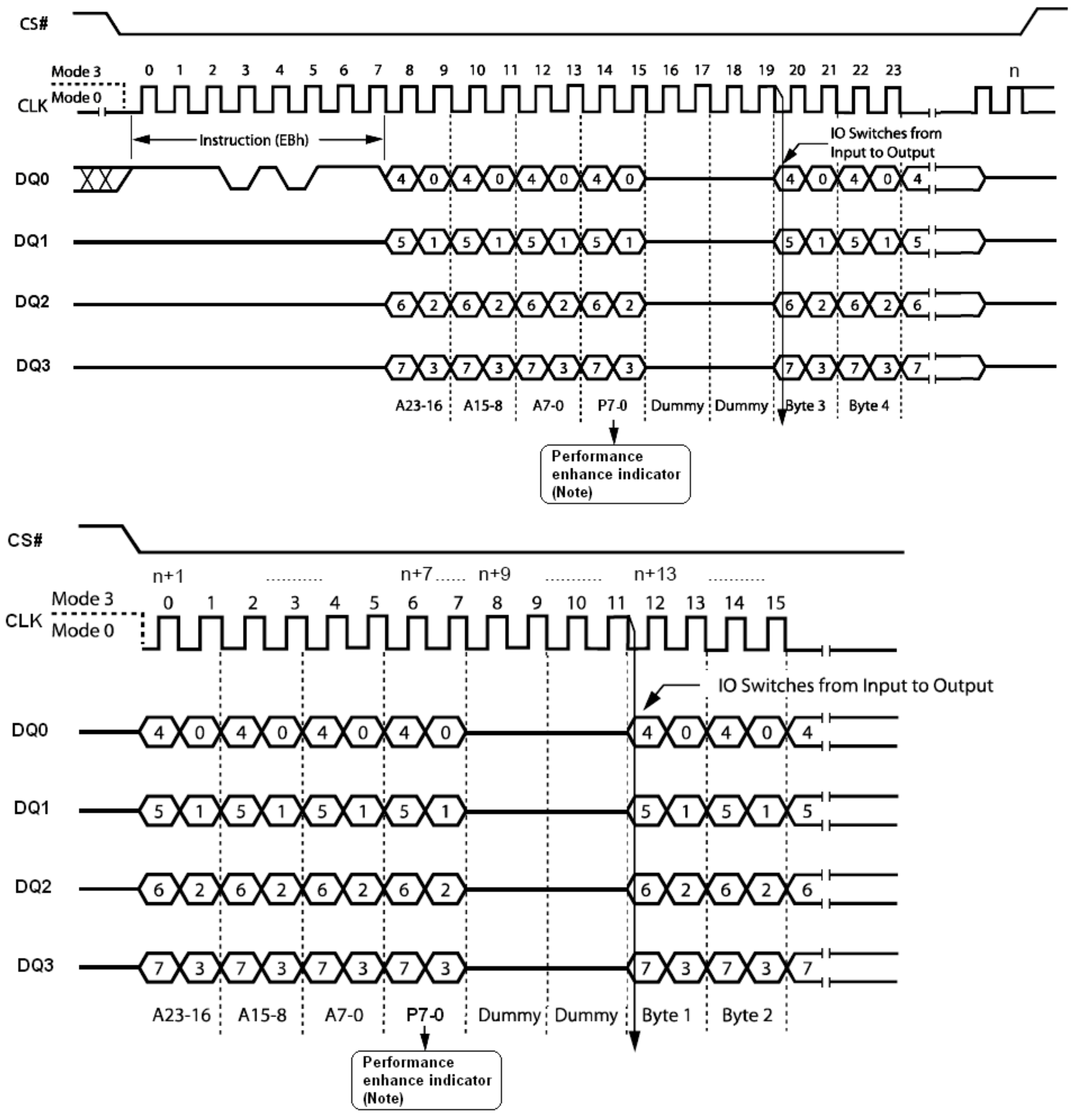
Figure 16.1. Quad Input / Output Fast Read Instruction Sequence under EQPI Mode

Another sequence of issuing Quad Input/Output FAST_READ (EBh) instruction especially useful in random access is : CS# goes low -> sending Quad Input/Output FAST_READ (EBh) instruction -> 24-bit address interleave on DQ_3, DQ_2, DQ_1 and DQ_0 -> performance enhance toggling bit P[7:0] -> 4 dummy clocks -> data out interleave on DQ_3, DQ_2, DQ_1 and DQ_0 till CS# goes high -> CS# goes low (reduce Quad Input/Output FAST_READ (EBh) instruction) -> 24-bit random access address, as shown in Figure 17.

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

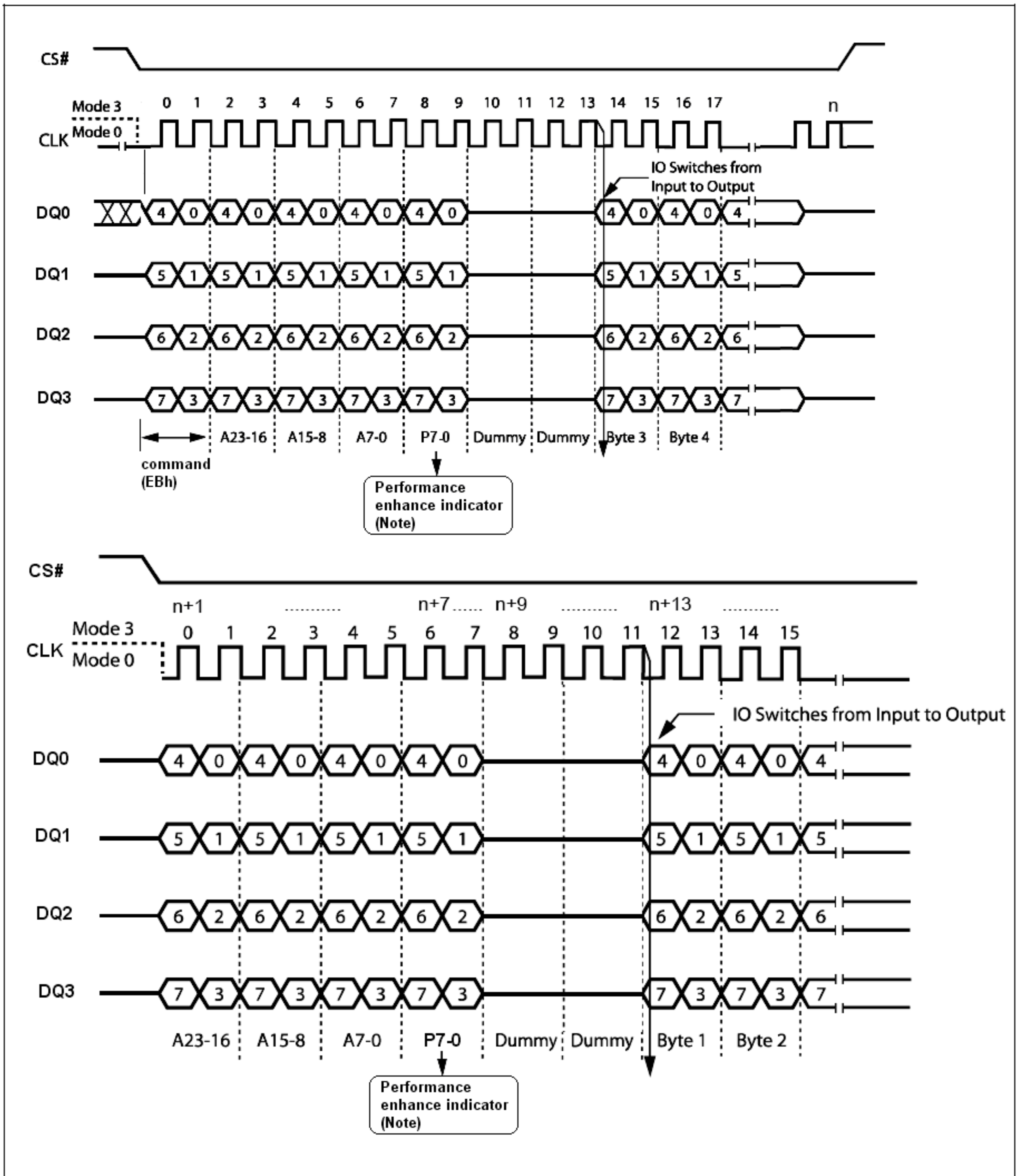
While Program/ Erase/ Write Status Register is in progress, Quad Input/Output FAST_READ (EBh) instruction is rejected without impact on the Program/ Erase/ Write Status Register current cycle.

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



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

Figure 17. Quad Input/Output Fast Read Enhance Performance Mode Sequence Diagram



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

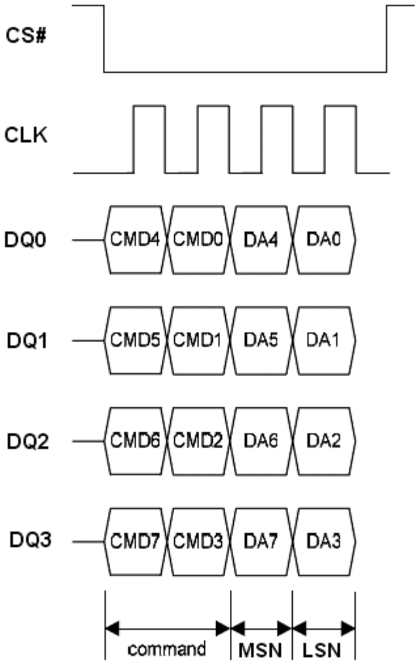
Figure 17.1 Quad Input/Output Fast Read Enhance Performance Mode Sequence under EQPI Mode

Set Burst (C0h)

The Set Burst command specifies the number of bytes to be output during a Read Bust command before the device wraps around. To set the burst length the host driver CS# low, sends the Set Burst command cycle (C0h) and one data cycle, then drivers CS# high. A cycle is two nibbles, or two clocks, long, most significant nibble first. After power-up or reset, the burst length is set to 8 bytes (00h), please refer to Table 9 for burst length data and Figure 18 for the sequence.

Table 9. Burst Length Data

Burst length	High Nibble (H0)	Low Nibble (L0)
8 Bytes (default)	0h	0h
16 Bytes	0h	1h
32 Bytes	0h	2h
64 Bytes	0h	3h
128 Bytes	0h	4h
256 Bytes	0h	5h



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

Figure 18. Set Burst Instruction Sequence Diagram

Read Burst (0Ch)

To execute a Read Burst operation the host drivers CS# low, and sends the Read Burst command cycle (0Ch), followed by three address cycles and two dummy cycles (4 clocks). Each of cycle is consisted of two nibbles (clocks) long, most significant nibble first,

After the dummy cycle, 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 bust length, see Table 10. 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 10. 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
128 Bytes	00-7FH, 80-FFH
256 Bytes	00-FFH

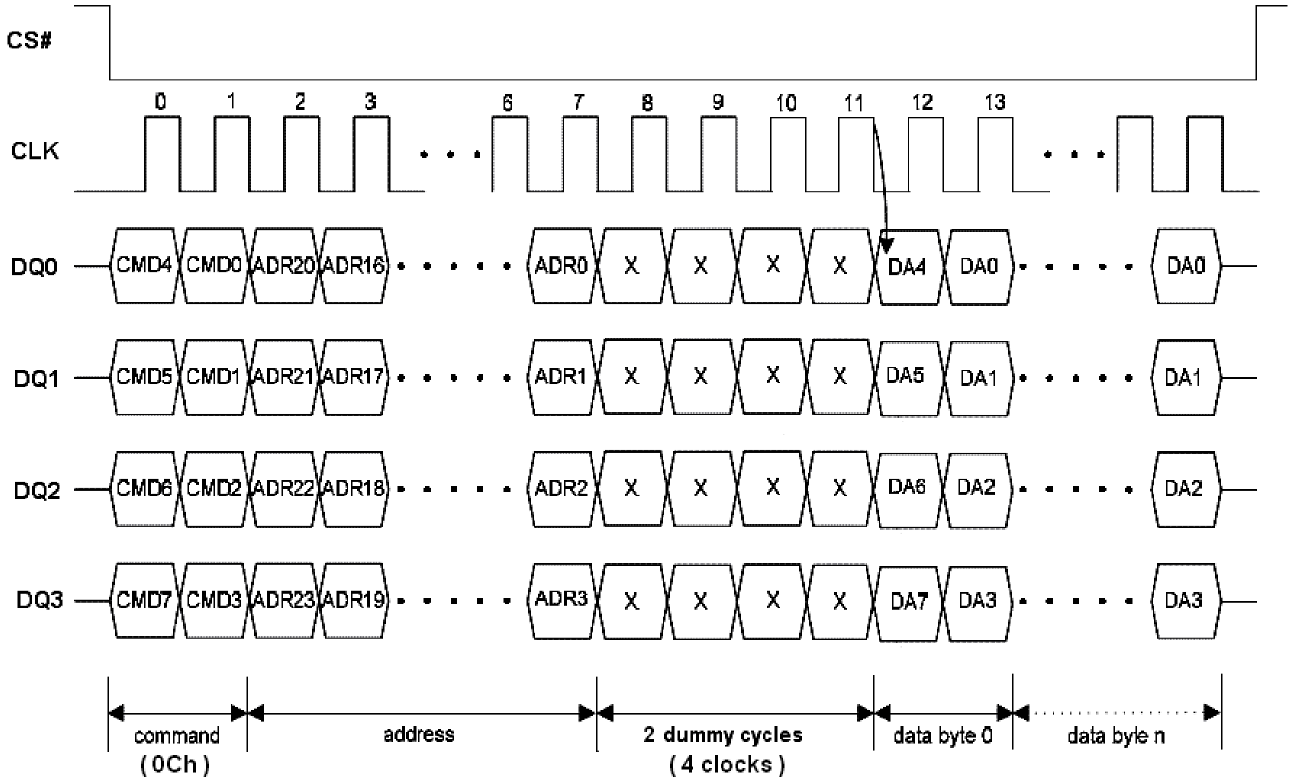


Figure 19. Read Burst Instruction Sequence Diagram (0Ch : 2 dummy cycles / 4 clocks)

Fast Read Burst (0Dh)

To execute a Fast Read Burst operation the host drivers CS# low, and sends the Read Burst command cycle (0Dh), followed by three address cycles and three dummy cycles (6 clocks). Each of cycle is consisted of two nibbles (clocks) long, most significant nibble first,

After the dummy cycle, 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 bust length, see Table 10. 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.

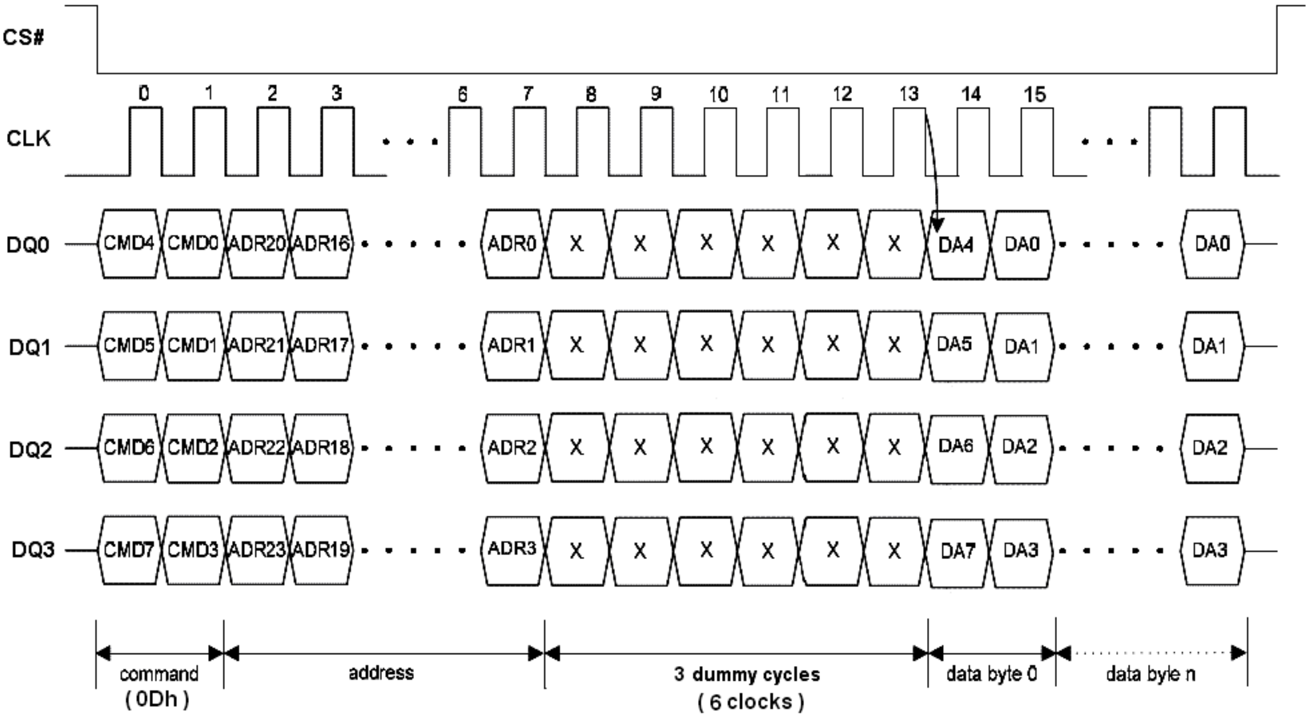


Figure 20. Fast Read Burst Instruction Sequence Diagram (0Dh : 3 dummy cycles / 6 clocks)

Page Program (PP) (02h)

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

The Page Program (PP) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, three address bytes and at least one data byte on Serial Data Input (DI). If the 8 least significant address bits (A7-A0) are not all zero, all transmitted data that goes beyond the end of the current page are programmed from the start address of the same page (from the address whose 8 least significant bits (A7-A0) are all zero). Chip Select (CS#) must be driven Low for the entire duration of the sequence.
The instruction sequence is shown in Figure 21. If more than 256 bytes are sent to the device, previously latched data are discarded and the last 256 data bytes are guaranteed to be programmed correctly within the same page. If less than 256 Data bytes are sent to device, they are correctly programmed at the requested addresses without having any effects on the other bytes of the same page.

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

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

A Page Program (PP) instruction applied to a page which is protected by the Block Protect (BP3, BP2, BP1, BP0) bits (see Table 3) is not executed.

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

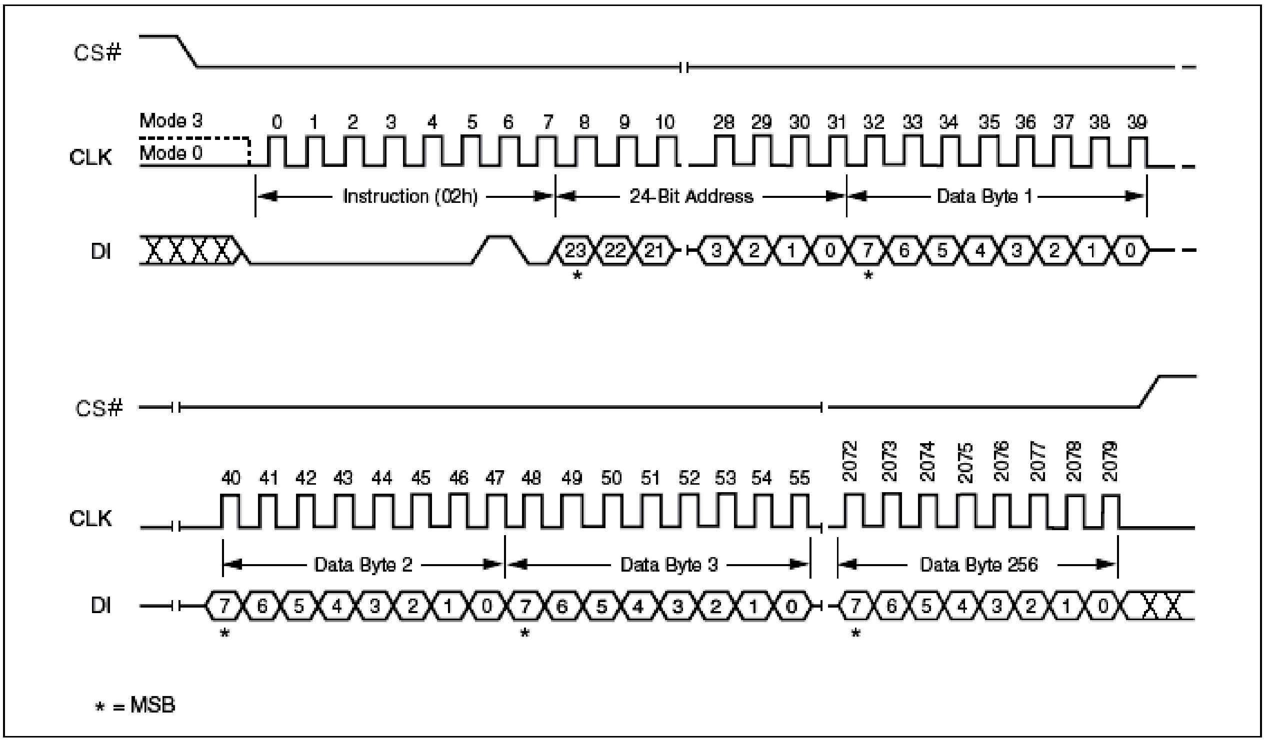


Figure 21. Page Program Instruction Sequence Diagram

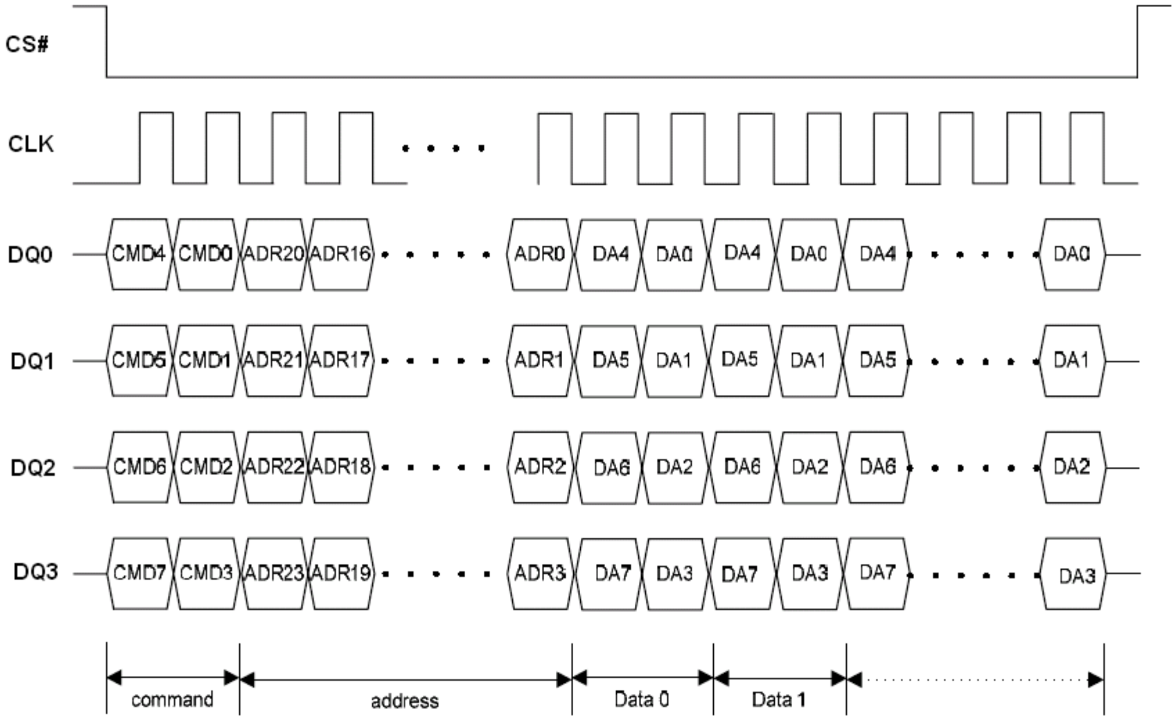


Figure 21.1 Program Instruction Sequence under EQPI Mode

Write Suspend (B0h)

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 Figure 22.

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

Suspend to suspend ready timing: 20us.

Resume to another suspend timing: 1ms.

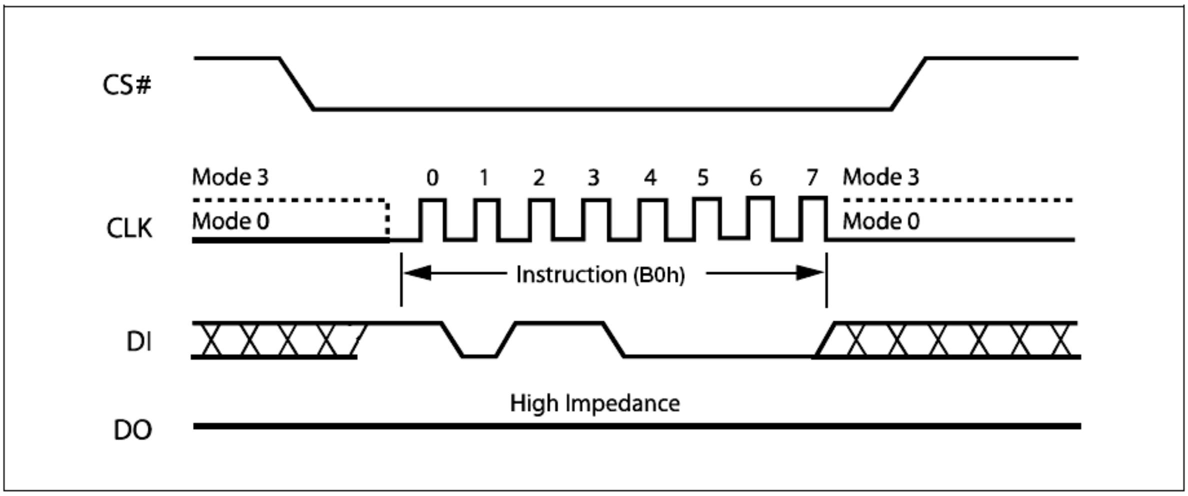


Figure 22. Write Suspend Instruction Sequence Diagram

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 sector 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 out put unknown data because the Sector or Block Erase will be incomplete.

To execute a Write Suspend operation, the host drives CS# low, sends the Write Suspend command cycle (B0h), then drives CS# high. A cycle is two nibbles long, most significant nibble first. The 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 Figure 23.1, 23.2 and 23.3.

Write Suspend During Page Programming

Issuing a Write Suspend instruction during Page Programming allows the host to erase or read any sector 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 Figure 23.1, 23.2 and 23.3.

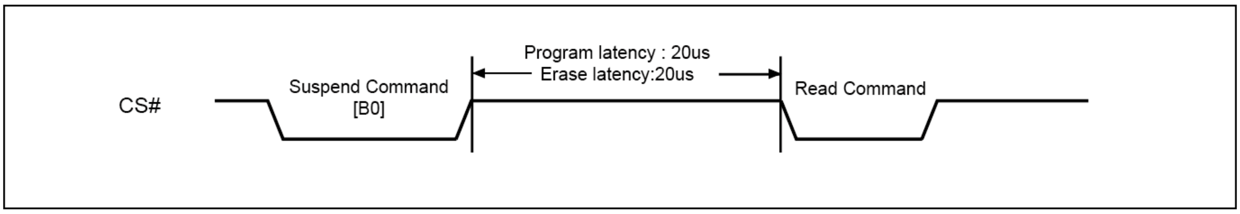


Figure 23.1 Suspend to Read Latency

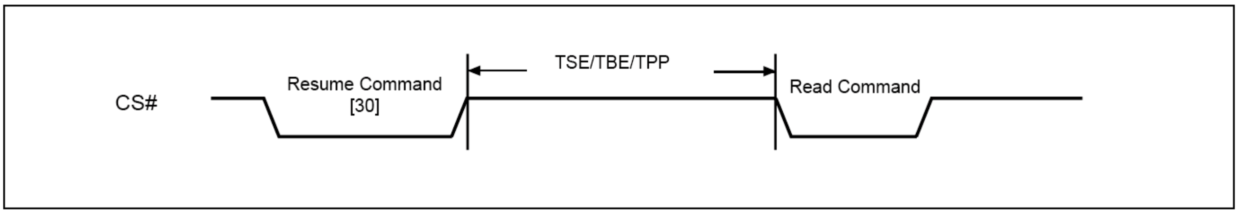


Figure 23.2 Resume to Read Latency

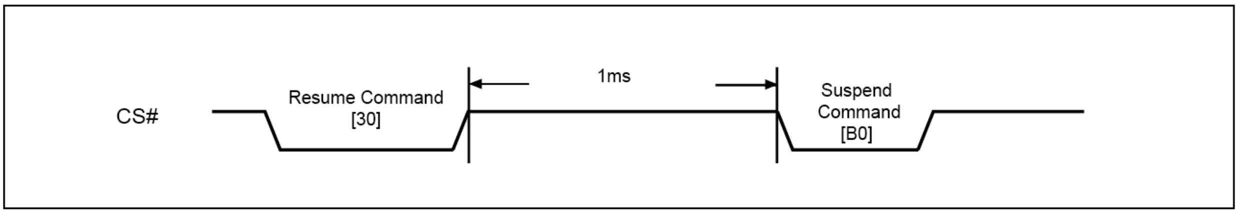


Figure 23.3 Resume to Suspend Latency

The instruction sequence is shown in Figure 23.4 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Write Resume (30h)

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

The instruction sequence is shown in Figure 23. 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_{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_{BE} or t_{PP}. Resume to another suspend operation requires latency time of 1ms.

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

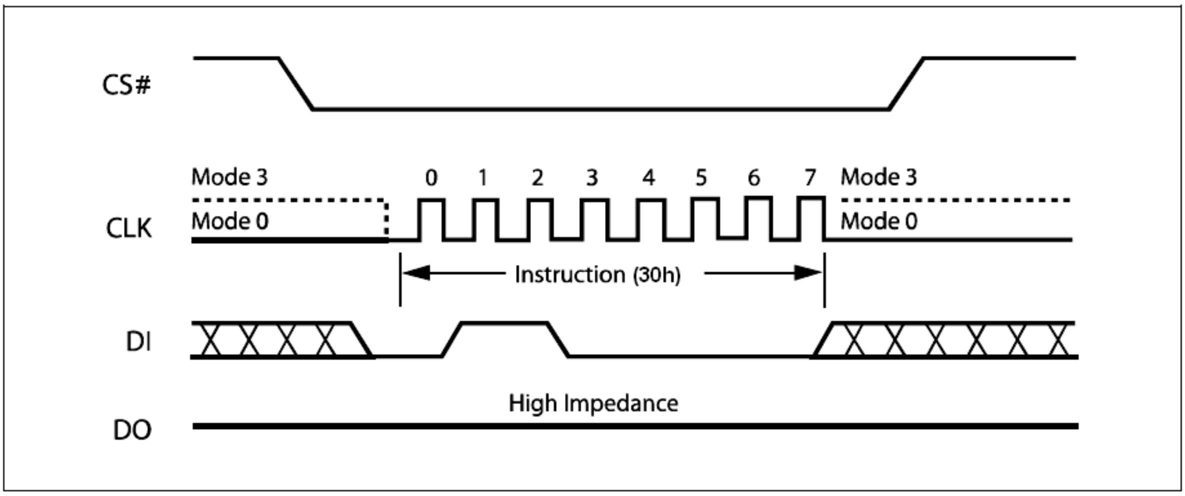


Figure 23. Write Resume Instruction Sequence Diagram

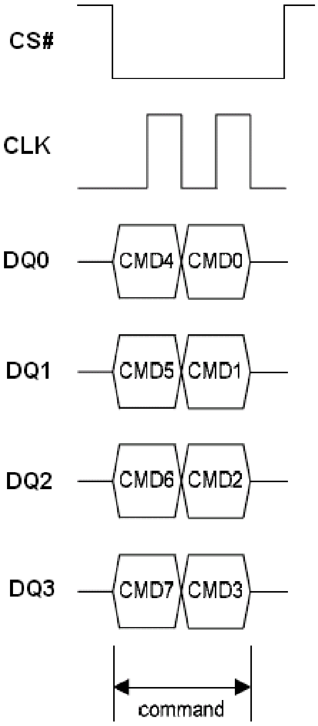


Figure 23.4 Write Suspend/Resume Instruction Sequence under EQPI Mode

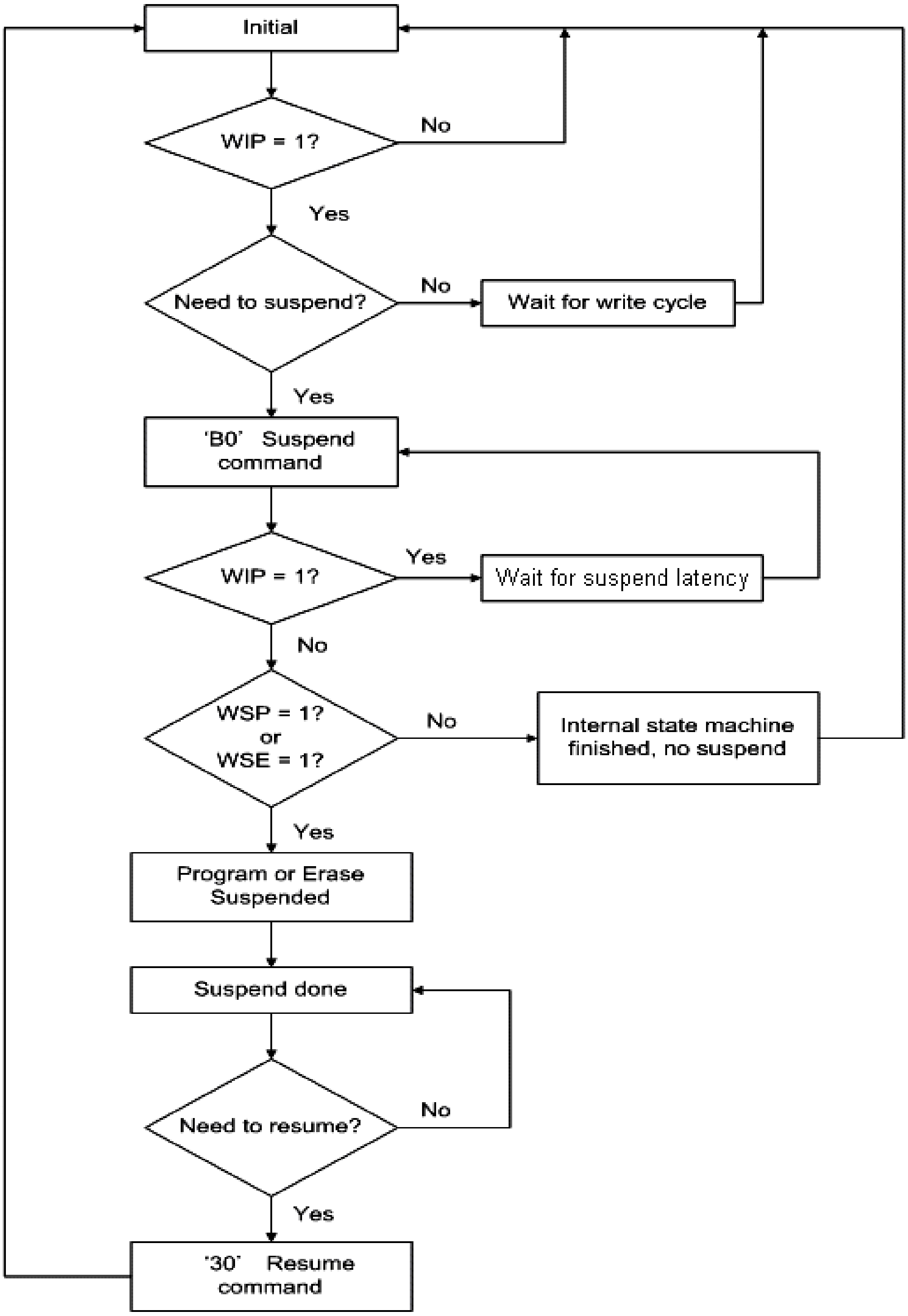


Figure 24. Write Suspend/Resume Flow

Note:

1. The 'WIP' can be either checked by command '09'or '05' polling.
2. 'Wait for write cycle' can be referring to maximum write cycle time or polling the WIP.
3. 'Wait for suspend latency', after issue program suspend command, latency time 20us is needed before issue another command or polling the WIP.
4. The 'WES' and 'WSE' can be checked by command '09' polling.
5. 'Suspend done' means the chip can do further operations allowed by suspend spec.

Sector Erase (SE) (20h)

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

The Sector Erase (SE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Sector (see Table 2) is a valid address for the Sector Erase (SE) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in Figure 25. Chip Select (CS#) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Sector Erase (SE) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Sector Erase cycle (whose duration is t_{SE}) is initiated. While the Sector Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Sector Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.
A Sector Erase (SE) instruction applied to a sector which is protected by the Block Protect (BP3, BP2, BP1, BP0) bits (see Table 3) is not executed.

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

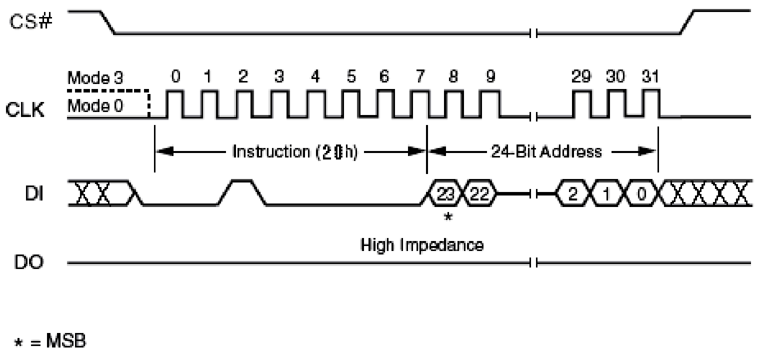


Figure 25. Sector Erase Instruction Sequence Diagram

Block Erase (BE) (D8h)

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

The Block Erase (BE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code, and three address bytes on Serial Data Input (DI). Any address inside the Block (see Table 2) is a valid address for the Block Erase (BE) instruction. Chip Select (CS#) must be driven Low for the entire duration of the sequence.

The instruction sequence is shown in Figure 26. Chip Select (CS#) must be driven High after the eighth bit of the last address byte has been latched in, otherwise the Block Erase (BE) instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Block Erase cycle (whose duration is t_{BE}) is initiated. While the Block Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Block Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.
A Block Erase (BE) instruction applied to a block which is protected by the Block Protect (BP3, BP2, BP1, BP0) bits (see Table 3) is not executed.

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

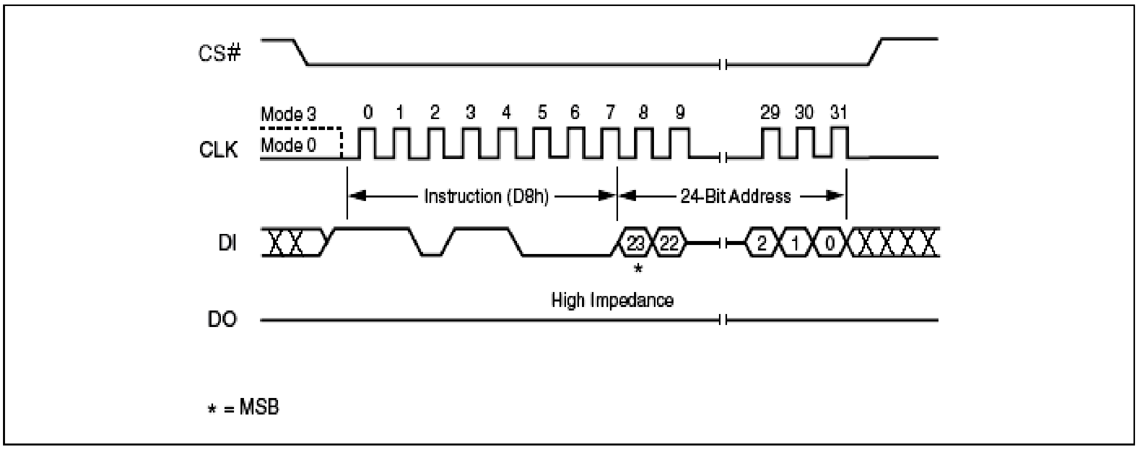


Figure 26. Block Erase Instruction Sequence Diagram

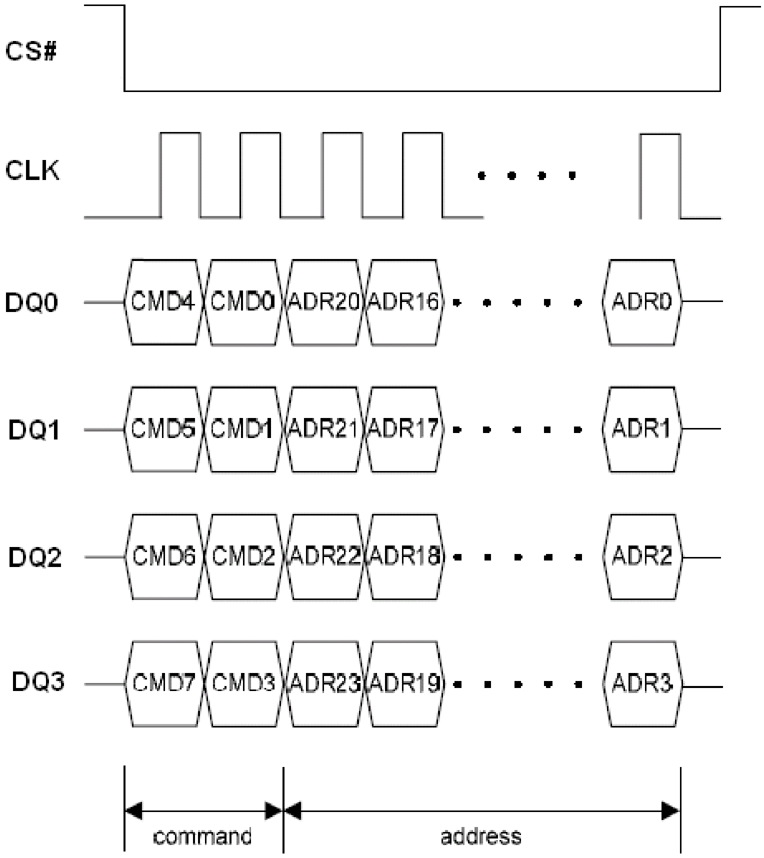


Figure 26.1 Block/Sector Erase Instruction Sequence under EQPI Mode

Chip Erase (CE) (C7h/60h)

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

The Chip Erase (CE) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code on Serial Data Input (DI). Chip Select (CS#) must be driven Low for the entire duration of the sequence.
The instruction sequence is shown in Figure 27. Chip Select (CS#) must be driven High after the eighth bit of the instruction code has been latched in, otherwise the Chip Erase instruction is not executed. As soon as Chip Select (CS#) is driven High, the self-timed Chip Erase cycle (whose duration is t_{CE}) is initiated. While the Chip Erase cycle is in progress, the Status Register may be read to check the value of the Write In Progress (WIP) bit. The Write In Progress (WIP) bit is 1 during the self-timed Chip Erase cycle, and is 0 when it is completed. At some unspecified time before the cycle is completed, the Write Enable Latch (WEL) bit is reset.
The Chip Erase (CE) instruction is executed only if all Block Protect (BP3, BP2, BP1, BP0) bits are 0. The Chip Erase (CE) instruction is ignored if one, or more blocks are protected.

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

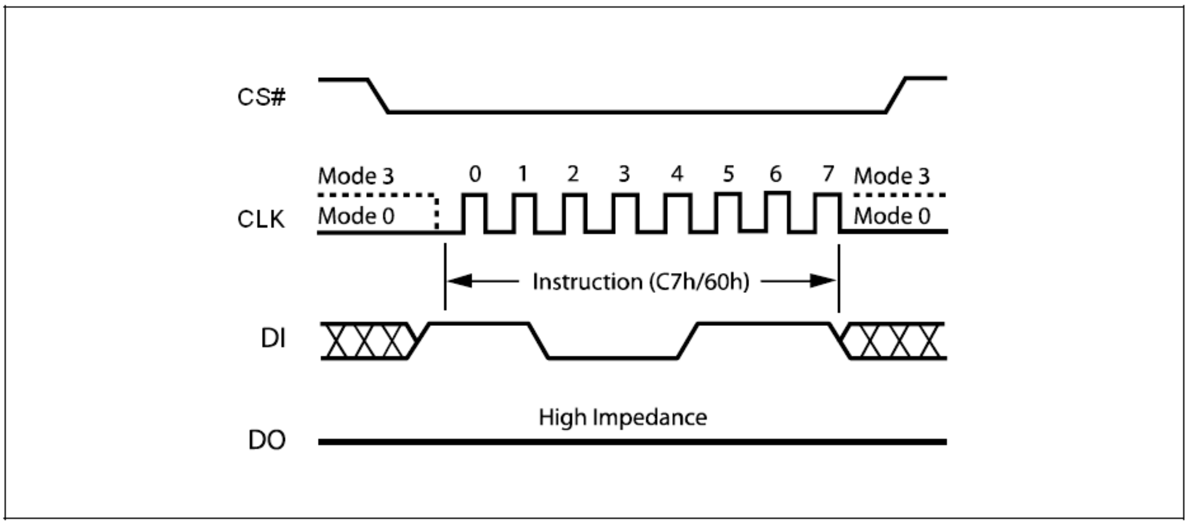


Figure 27. Chip Erase Instruction Sequence Diagram

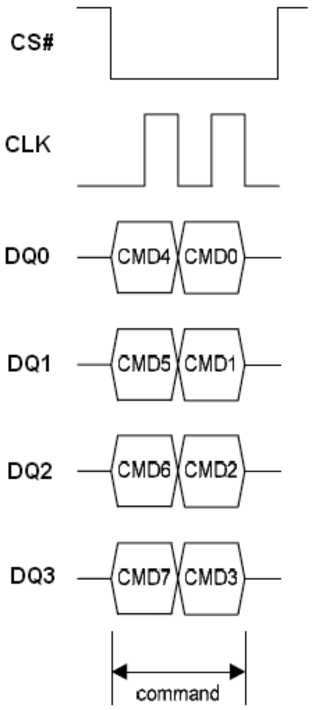


Figure 27.1 Chip Erase Sequence under EQPI Mode

Deep Power-down (DP) (B9h)

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

Driving Chip Select (CS#) High deselects the device, and puts the device in the Standby mode (if there is no internal cycle currently in progress). But this mode is not the Deep Power-down mode. The Deep Power-down mode can only be entered by executing the Deep Power-down (DP) instruction, to reduce the standby current (from I_{CC1} to I_{CC2}, as specified in Table 13.)

Once the device has entered the Deep Power-down mode, all instructions are ignored except the Release from Deep Power-down and Read Device ID (RDI) instruction. This releases the device from this mode. The Release from Deep Power-down and Read Device ID (RDI) instruction also allows the Device ID of the device to be output on Serial Data Output (DO).
The Deep Power-down mode automatically stops at Power-down, and the device always Powers-up in the Standby mode. The Deep Power-down (DP) instruction is entered by driving Chip Select (CS#) Low, followed by the instruction code on Serial Data Input (DI). Chip Select (CS#) must be driven Low for the entire duration of the sequence.
The instruction sequence is shown in Figure 28.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.

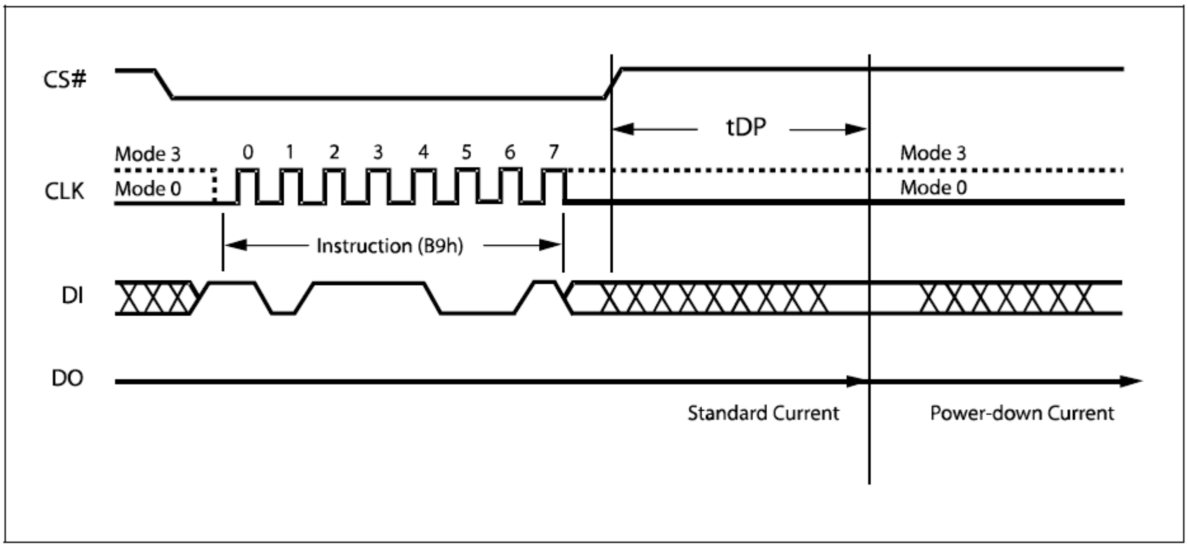


Figure 28. Deep Power-down Instruction Sequence Diagram

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

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

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

When used only to release the device from the power-down state, the instruction is issued by driving the CS# pin low, shifting the instruction code “ABh” and driving CS# high as shown in Figure 29. After the time duration of t_{RES1} (See AC Characteristics) the device will resume normal operation and other instructions will be accepted. The CS# pin must remain high during the t_{RES1} time duration.

When used only to obtain the Device ID while not in the power-down state, the instruction is initiated by driving the CS# pin low and shifting the instruction code “ABh” followed by 3-dummy bytes. The Device ID bits are then shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Figure 30. The Device ID value for the EN25S64 are listed in Table 6. The Device ID can be read continuously. The instruction is completed by driving CS# high.

When Chip Select (CS#) is driven High, the device is put in the Stand-by Power mode. If the device was not previously in the Deep Power-down mode, the transition to the Stand-by Power mode is immediate. If the device was previously in the Deep Power-down mode, though, the transition to the Standby Power mode is delayed by t_{RES2}, and Chip Select (CS#) must remain High for at least t_{RES2} (max), as specified in Table 15. 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.

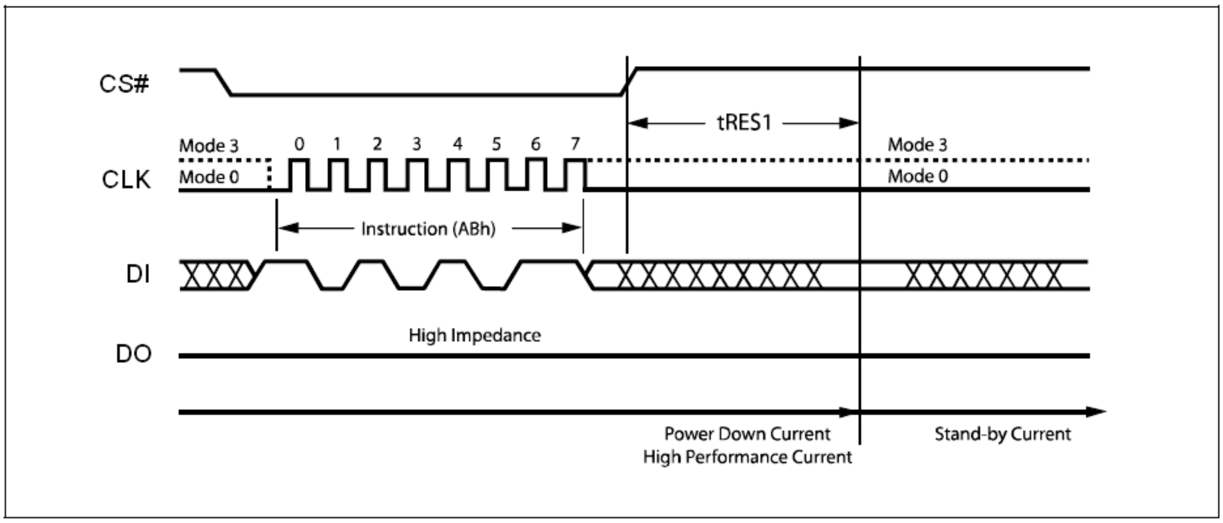


Figure 29. Release Power-down Instruction Sequence Diagram

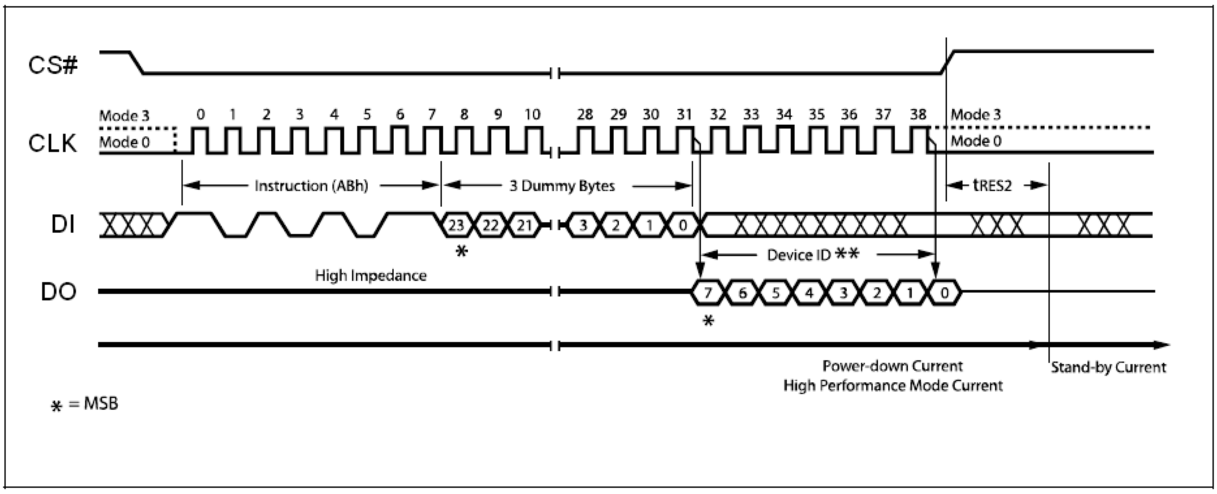


Figure 30. Release Power-down / Device ID Instruction Sequence Diagram

Read Manufacturer / Device ID (90h)

The Read Manufacturer/Device ID instruction is an alternative to the Release from Power-down / Device ID instruction that provides both the JEDEC assigned manufacturer ID and the specific device ID.
The Read Manufacturer/Device ID instruction is very similar to the Release from Power-down / Device ID instruction. The instruction is initiated by driving the CS# pin low and shifting the instruction code "90h" followed by a 24-bit address (A23-A0) of 000000h. After which, the Manufacturer ID for Eon (1Ch) and the Device ID are shifted out on the falling edge of CLK with most significant bit (MSB) first as shown in Figure 31. The Device ID values for the EN25S64 are listed in Table 6. If the 24-bit address is initially set to 000001h the Device ID will be read first

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

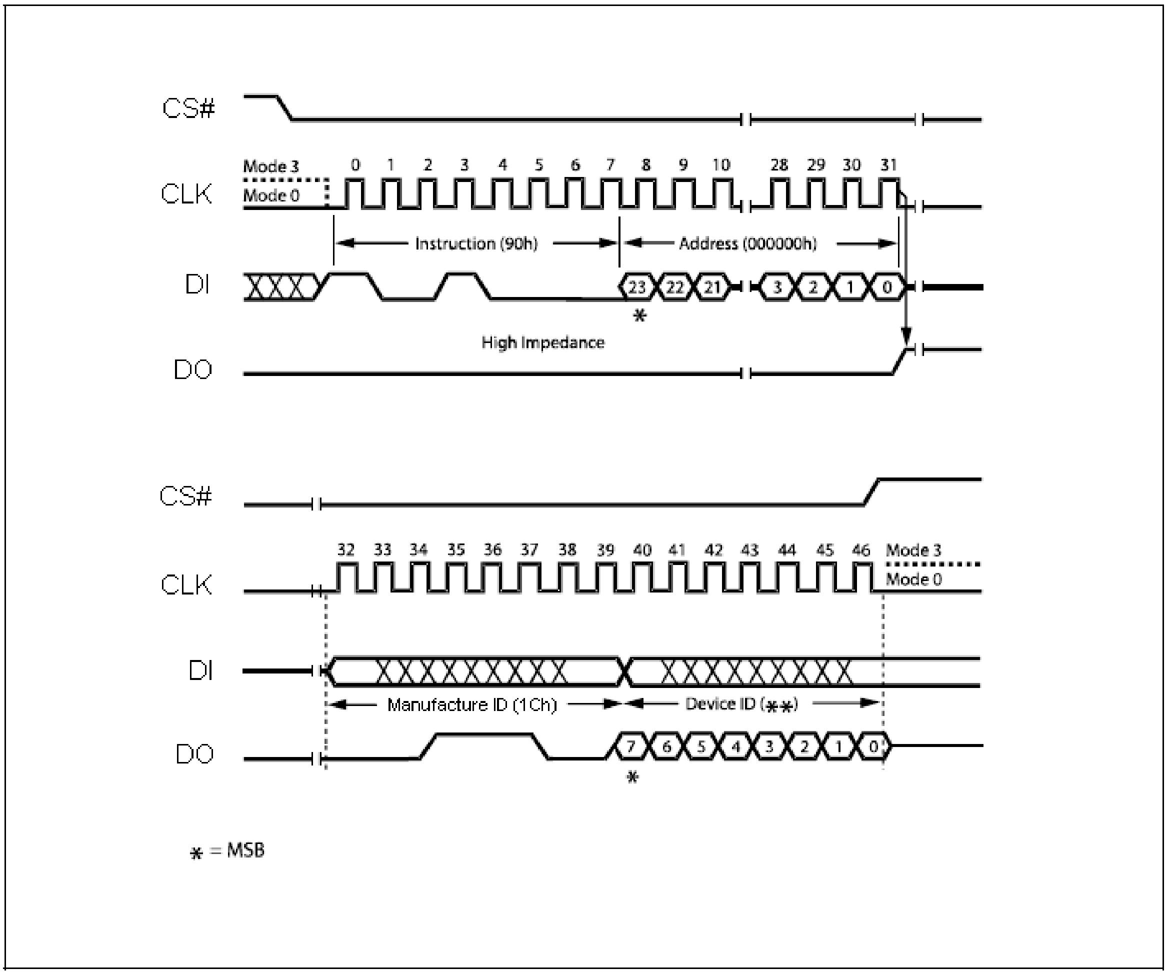


Figure 31. Read Manufacturer / Device ID Diagram

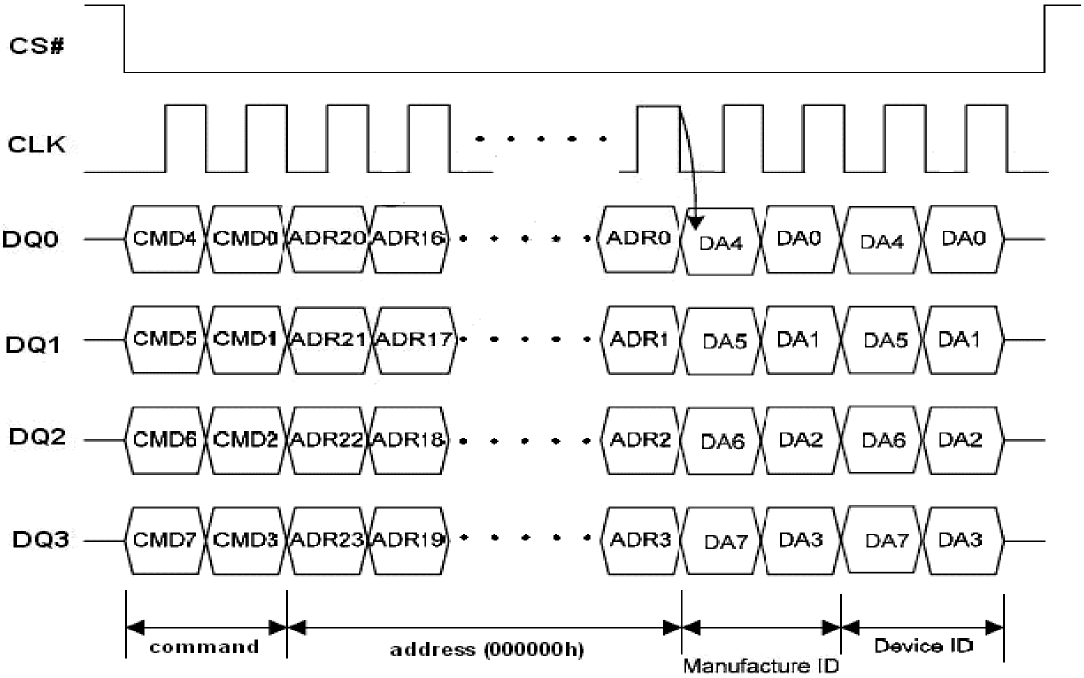


Figure 31.1. Read Manufacturer / Device ID Diagram under EQPI Mode

Read Identification (RDID) (9Fh)

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

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

The device is first selected by driving Chip Select Low. Then, the 8-bit instruction code for the instruction is shifted in. This is followed by the 24-bit device identification, stored in the memory, being shifted out on Serial Data Output , each bit being shifted out during the falling edge of Serial Clock . The instruction sequence is shown in Figure 32. The Read Identification (RDID) instruction is terminated by driving Chip Select High at any time during data output.

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

The instruction sequence is shown in Figure 32.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

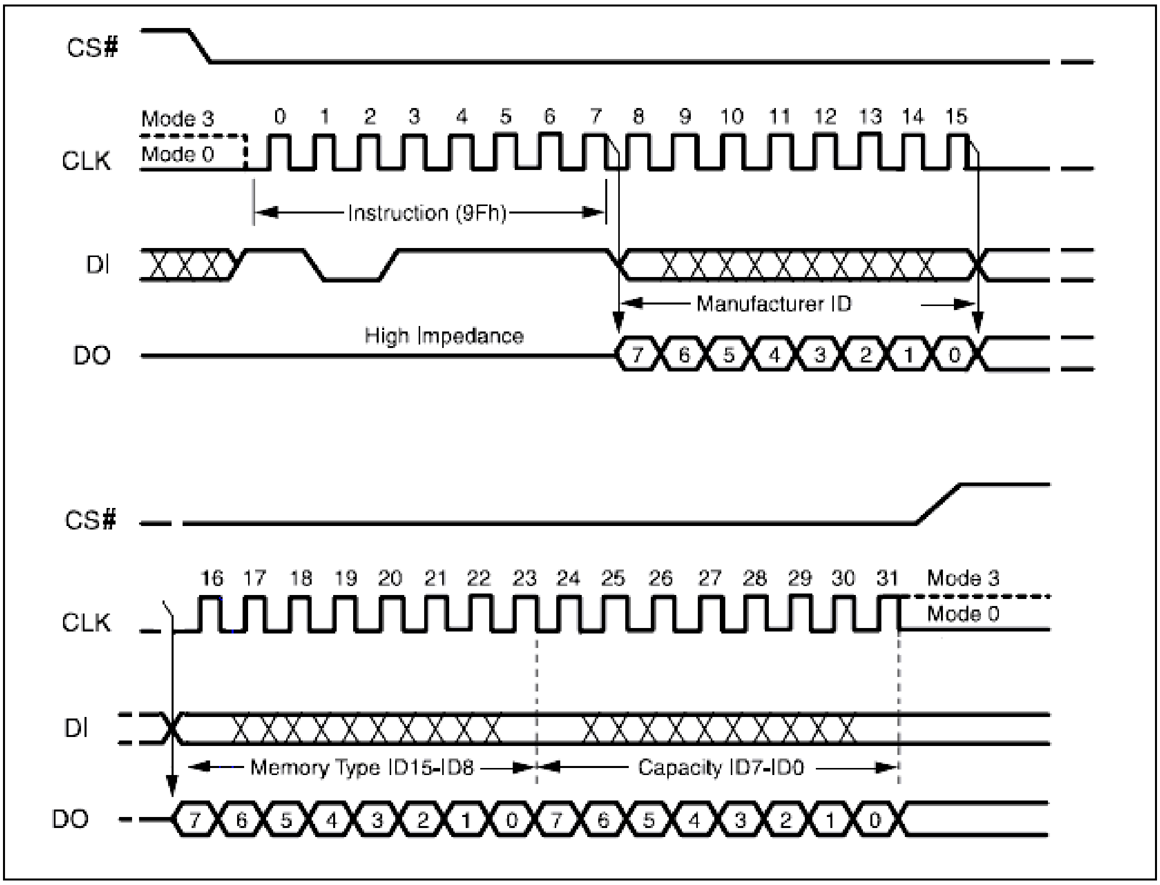


Figure 32. Read Identification (RDID)

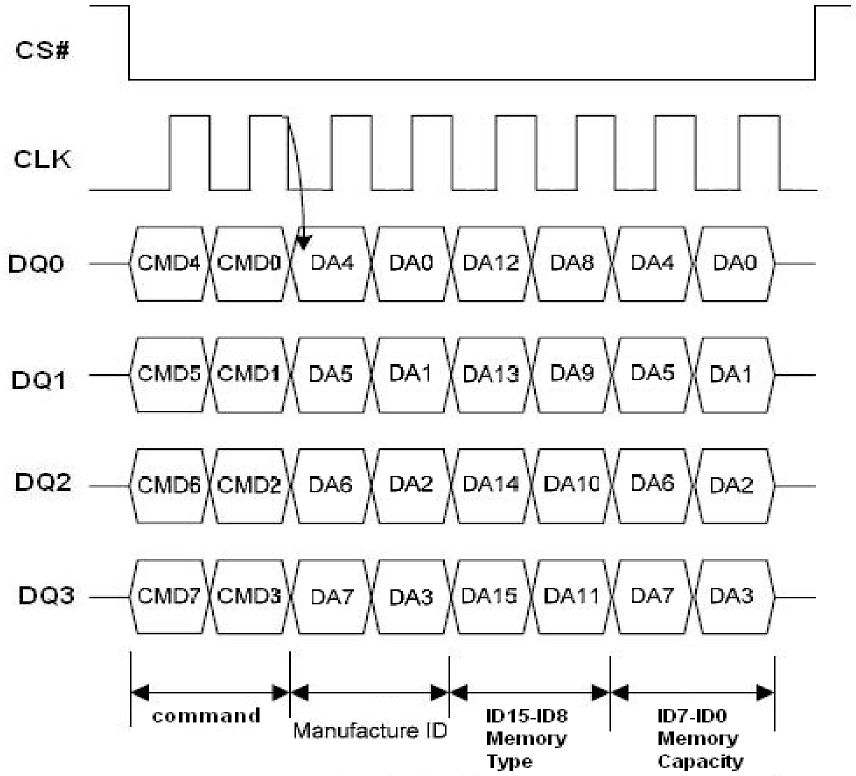


Figure 32.1. Read Identification (RDID) under EQPI Mode

Enter OTP Mode (3Ah)

This Flash has an extra 512 bytes OTP sector, user must issue ENTER OTP MODE command to read, program or erase OTP sector. After entering OTP mode, the OTP sector is mapping to sector 2047, **SRP bit** becomes OTP_LOCK bit and can be read with RDSR command. Program / Erase command will be disabled when OTP_LOCK bit is ‘1’

WRSR command will ignore the input data and program OTP_LOCK bit to 1.
User must clear the protect bits before enter OTP mode.
OTP sector can only be program and erase before OTP_LOCK bit is set to ‘1’ and BP [3:0] = ‘0000’ In OTP mode, user can read other sectors, but program/erase other sectors only allowed when OTP_LOCK bit equal to ‘0’.
User can use WRDI (04H) command to exit OTP mode.

Erase OTP Command (20h)

User can use Sector Erase (20h) command only to erase OTP data.

The instruction sequence is shown in Figure 33.1 while using the Enable Quad Peripheral Interface mode (EQPI) (38h) command.

Table 11. OTP Sector Address

Sector	Sector Size	Address Range
2047	512 byte	7FF000h – 7FF1FFh

Note: The OTP sector is mapping to sector 2047

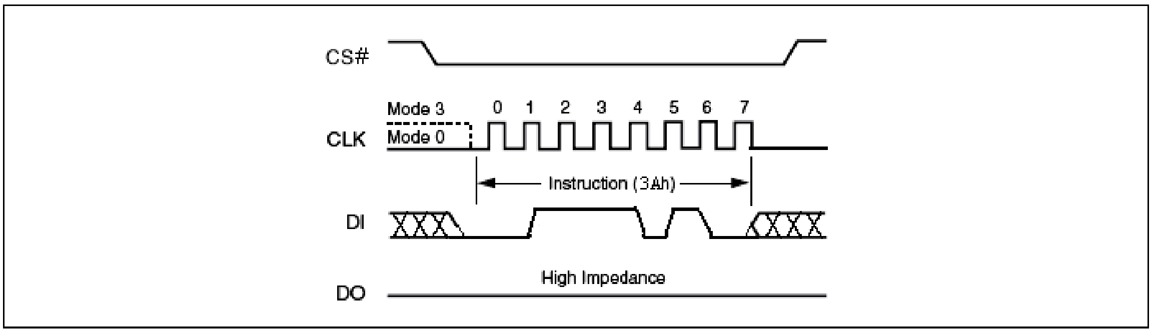


Figure 33. Enter OTP Mode

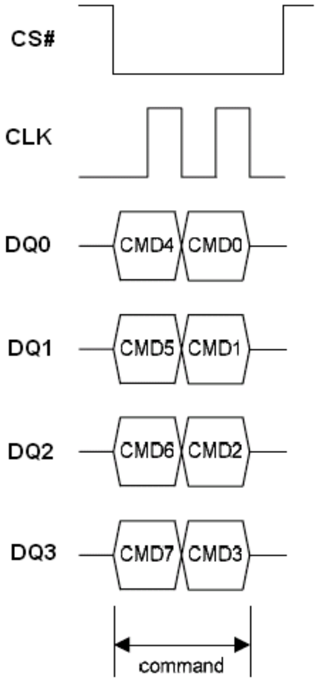


Figure 33.1 Enter OTP Mode Sequence under EQPI Mode

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 270 ms). See Table 12 and Figure 34 for more information.

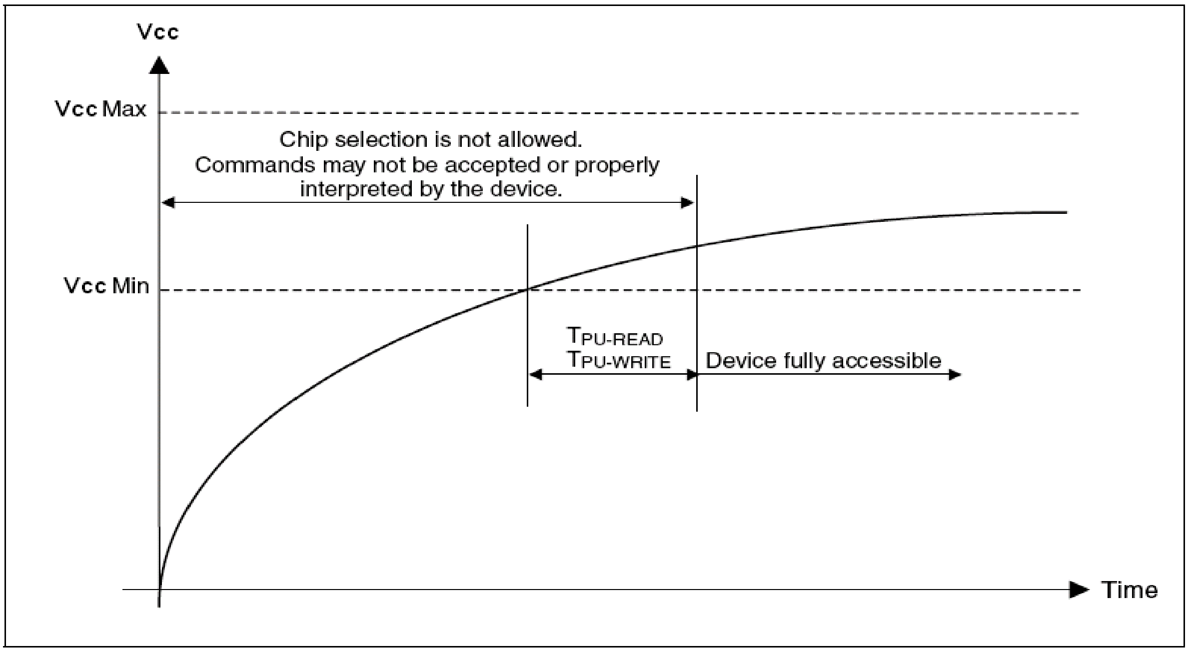


Figure 34. Power-up Timing

Table 12. 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).

Table 13. DC Characteristics

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

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
I_{LI}	Input Leakage Current			2	μA
I_{LO}	Output Leakage Current			2	μA
I_{CC1}	Standby Current	CS# = V_{CC}, V_{IN} = V_{SS} or V_{CC}		10	μA
I_{CC2}	Deep Power-down Current	CS# = V_{CC}, V_{IN} = V_{SS} or V_{CC}		5	μA
I_{CC3}	Operating Current (READ)	CLK = 0.1 V_{CC} / 0.9 V_{CC} at 104MHz, DQ = open		20	mA
		CLK = 0.1 V_{CC} / 0.9 V_{CC} at 80MHz, DQ = open		25	mA
I_{CC4}	Operating Current (PP)	CS# = V_{CC}		30	mA
I_{CC5}	Operating Current (WRSR)	CS# = V_{CC}		15	mA
I_{CC6}	Operating Current (SE)	CS# = V_{CC}		15	mA
I_{CC7}	Operating Current (BE)	CS# = V_{CC}		15	mA
V_{IL}	Input Low Voltage		– 0.5	0.2 V_{CC}	V
V_{IH}	Input High Voltage		0.7V_{CC}	V_{CC}+0.4	V
V_{OL}	Output Low Voltage	I_{OL} = 100 μA, Vcc=Vcc Min.		0.3	V
V_{OH}	Output High Voltage	I_{OH} = –100 μA , Vcc=Vcc Min.	V_{CC}-0.2		V

Table 14. 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

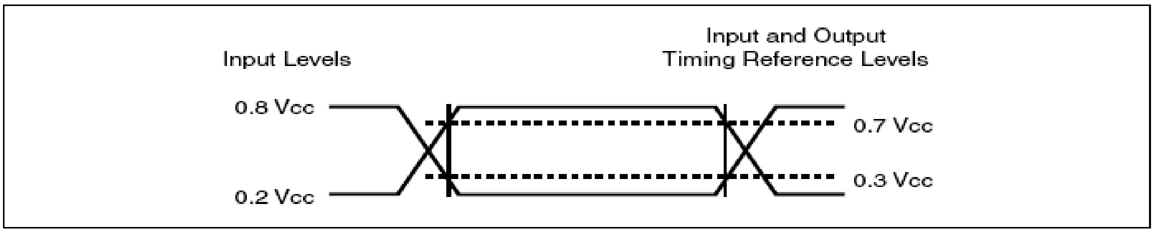


Figure 35. AC Measurement I/O Waveform

Table 15. 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 Clock Frequency for: FAST_READ, PP, SE, BE, DP, RES, WREN, WRDI, WRSR, Fast Read Burst	D.C.		104	MHz
		Serial Clock Frequency for: Read Burst, Dual Output Fast Read and Quad I/O Fast Read	D.C.		80	MHz
f_R		Serial Clock Frequency for READ, RDSR, RDSSR, RDID	D.C.		50	MHz
t_{CH}[1]		Serial Clock High Time	6			ns
t_{CL}[1]		Serial Clock Low Time	6			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 for read CS# High Time for program/erase	30 40			ns
t_{SHQZ}[2]	t_{DIS}	Output Disable Time			6	ns
t_{CLQX}	t_{HO}	Output Hold Time	0			ns
t_{DVCH}	t_{DSU}	Data In Setup Time	2			ns
t_{CHDX}	t_{DH}	Data In Hold Time	5			ns
t_{CLQV}	t_V	Output Valid from CLK			8	ns
t_{WHSL}[3]		Write Protect Setup Time before CS# Low	20			ns
t_{SHWL}[3]		Write Protect Hold Time after CS# High	100			ns
t_{DP}[2]		CS# High to Deep Power-down Mode			3	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		20	50	ms
t_{PP}		Page Programming Time		0.7	5	ms
t_{SE}		Sector Erase Time		0.04	0.3	s
t_{BE}		Block Erase Time		0.3	2	s
t_{CE}		Chip Erase Time		34	100	s
	t_{SR}	Software Reset Latency WIP = write operation			28	μs
		WIP = not in write operation			0	μs

Note: 1. t_{CH} + t_{CL} must be greater than or equal to 1/ f_C

2. Value guaranteed by characterization, not 100% tested in production.

3. Only applicable as a constraint for a Write status Register instruction when Status Register Protect Bit is set at 1.

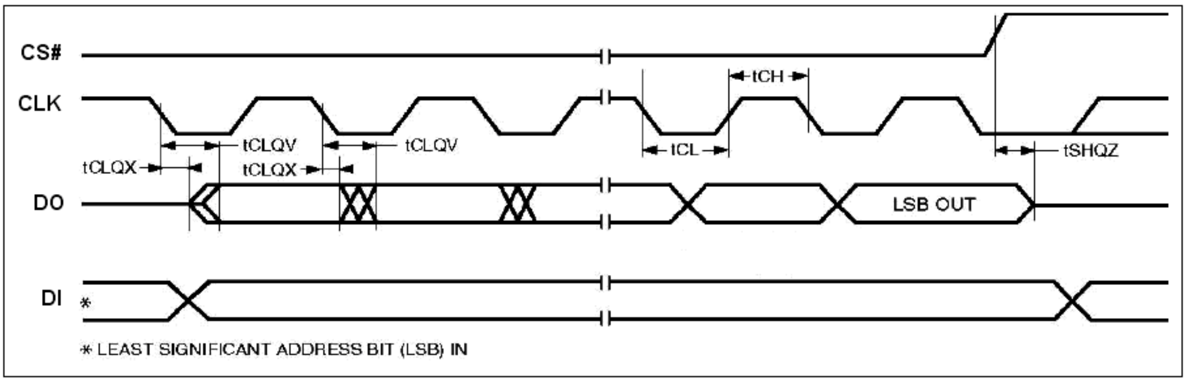


Figure 36. Serial Output Timing

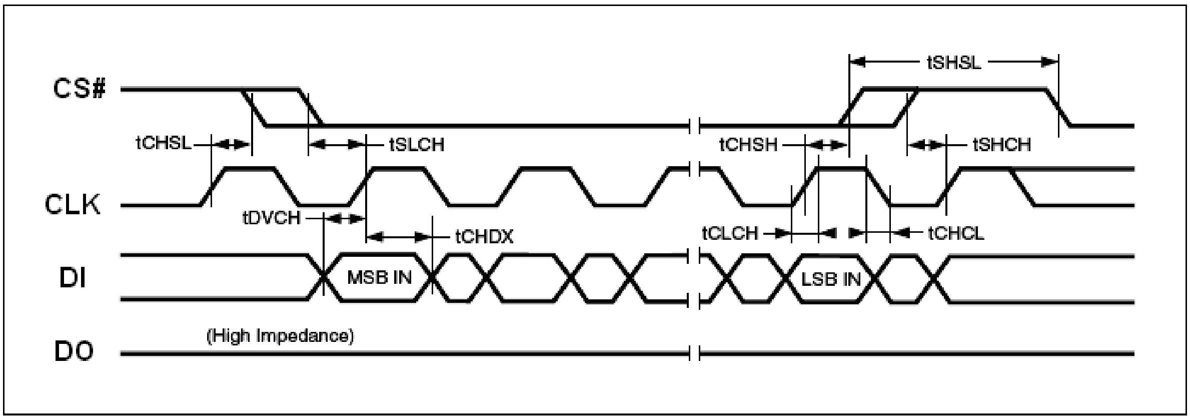


Figure 37. Input Timing

ABSOLUTE MAXIMUM RATINGS

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

Parameter	Value	Unit
Storage Temperature	-65 to +150	°C
Plastic Packages	-65 to +125	°C
Output Short Circuit Current[1]	200	mA
Input and Output Voltage (with respect to ground) [2]	-0.5 to Vcc+0.5	V
Vcc	-0.5 to Vcc+0.5	V

Notes:

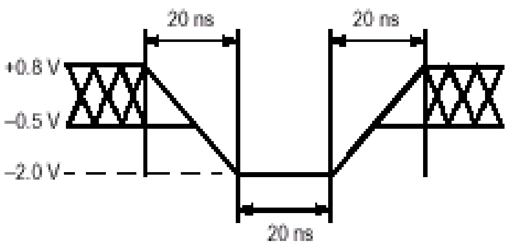
1. No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.
2. Minimum DC voltage on input or I/O pins is –0.5 V. During voltage transitions, inputs may undershoot V_{ss} to –1.0V for periods of up to 50ns and to –2.0 V for periods of up to 20ns.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

RECOMMENDED OPERATING RANGES [1]

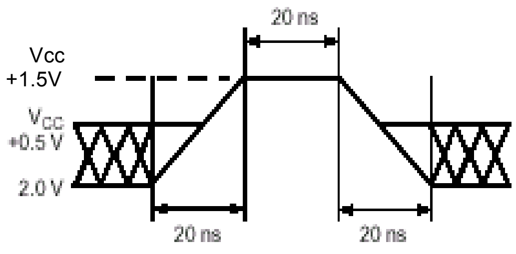
Parameter	Value	Unit
Ambient Operating Temperature Industrial Devices	-40 to 85	°C
Operating Supply Voltage Vcc	Full: 1.65 to 1.95	V

Notes:

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



Maximum Negative Overshoot Waveform



Maximum Positive Overshoot Waveform

Table 16. DATA RETENTION and ENDURANCE

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

Table 17. CAPACITANCE

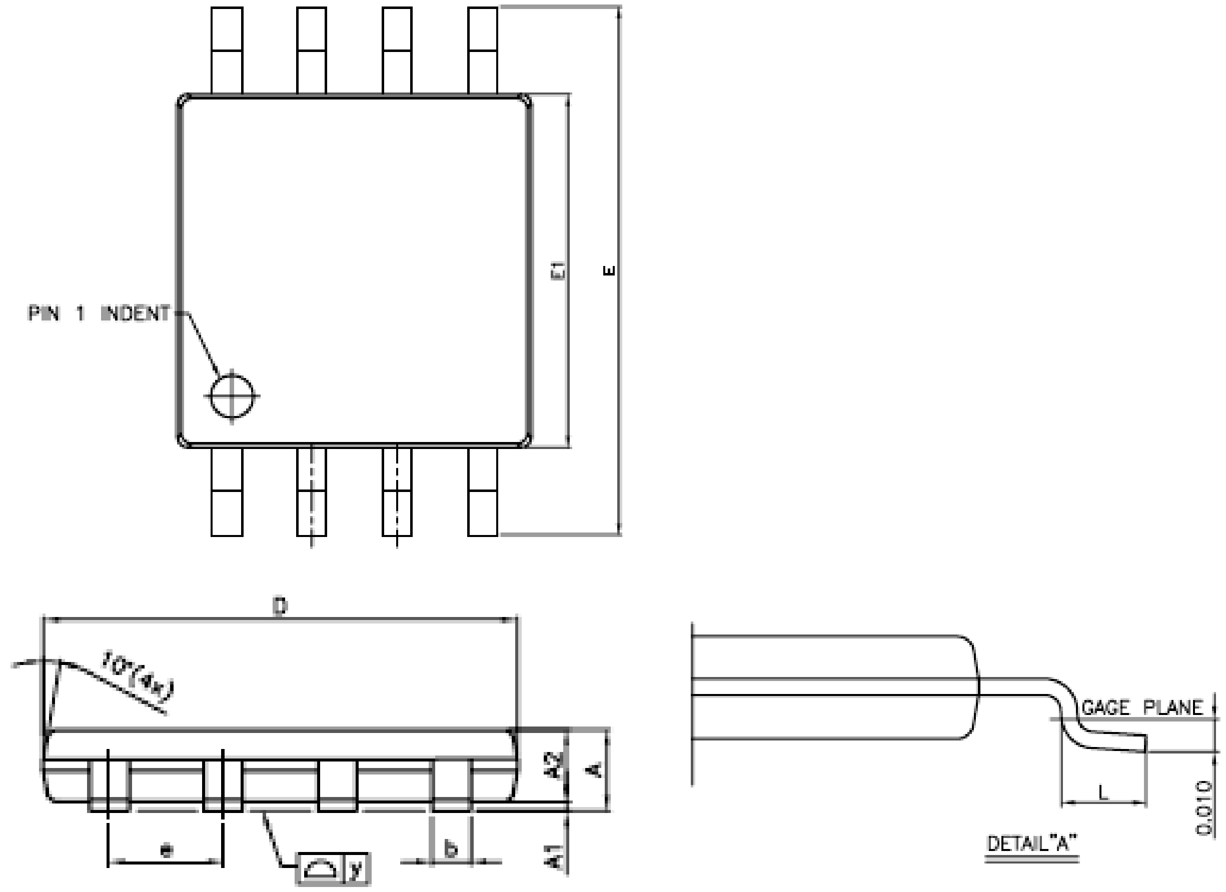
(V_{CC} = 1.65-1.95V)

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 20MHz.

PACKAGE MECHANICAL

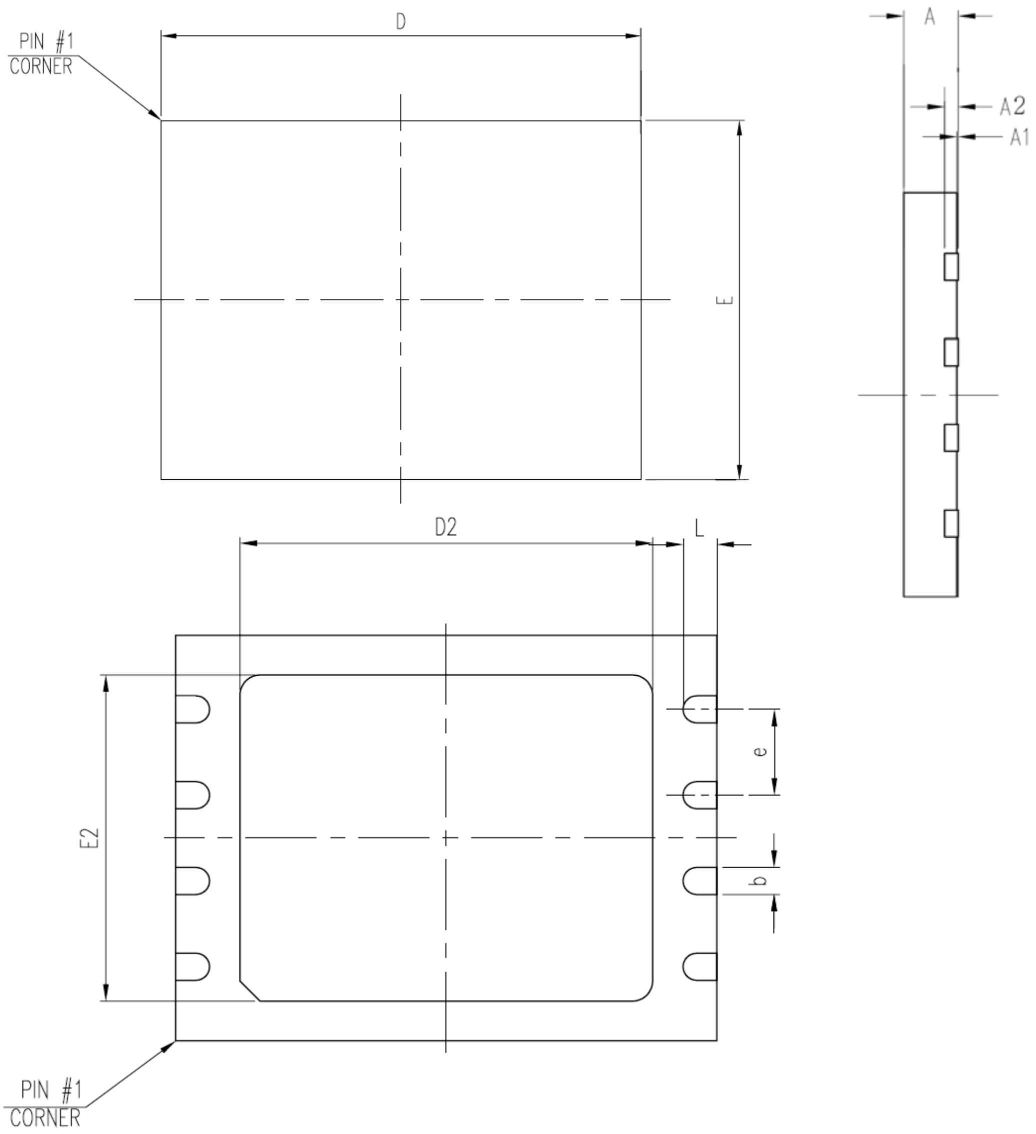
Figure 38. VSOP 200 mil (Official name = 208 mil)



SYMBOL	DIMENSION IN MM		
	MIN.	NOR	MAX
A	--	--	1.00
A1	0.05	0.10	0.15
A2	0.75	0.80	0.85
D	5.18	5.28	5.38
E	7.70	7.90	8.10
E1	5.18	5.28	5.38
e	- - -	1.27	- - -
b	0.35	0.42	0.48
L	0.5	0.65	0.80
θ	0	--	10

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

Figure 39. VDFN8 (5x6mm)



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

Purpose

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

Eon products’ Top Marking

cFeon

cFeon Top Marking Example:

cFeon

Part Number: XXXX-XXX
Lot Number: XXXXX
Date Code: XXXXX

For More Information

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

ORDERING INFORMATION

EN25S64 - 104 R I P

PACKAGING CONTENT
P = RoHS compliant

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

PACKAGE
R = 8-pin 200mil VSOP
W = 8-pin VDFN (5x6mm)

SPEED
104 = 104 MHz

BASE PART NUMBER
EN = Eon Silicon Solution Inc.
25S = 1.8V Serial Flash with 4KB Uniform-Sector
64 = 64 Megabit (8192K x 8)

Revisions List

Revision No	Description	Date
A	Initial Release	2010/12/27
B	Add package option of 8-pin 200mil VSOP	2011/02/08
C	Revise t_{SHSL} (CS# High Time for read) from 10ns to 30ns on page 54.	2011/02/25
D	1. Correct the typo of 6 dummy clocks on page 30. 2. Update Figure 24. Write Suspend/Resume Flow on page 41.	2011/05/25
E	1. Add the note “5. This flow cannot release the device from Deep power down mode.” on page 18. 2. Update Table 15. AC Characteristics on page 54.	2011/06/28
F	Remove the 8-pin 200mil SOP package option.	2011/07/06
G	Supplement the description of suspend and resume latency timing on page 38, 39, 40 and 41.	2011/09/23