THE DATASHEET OF

IS25LP256D-JLLE-TR

www.win-source.net 0086-755-83957316

IS25LP256D IS25WP256D

256Mb

SERIAL FLASH MEMORY WITH 166/104MHZ MULTI I/O SPI & QUAD I/O QPI DTR INTERFACE

DATA SHEET

256Mb

SERIAL FLASH MEMORY WITH 166MHZ MULTI I/O SPI & QUAD I/O QPI DTR INTERFACE

FEATURES

- **Industry Standard Serial Interface**
  - IS25LP256D: 256Mbit/32Mbyte
  - IS25WP256D: 256Mbit/32Mbyte
  - 3 or 4 Byte Addressing Mode
  - Supports Standard SPI, Fast, Dual, Dual I/O, Quad, Quad I/O, SPI DTR, Dual I/O DTR, Quad I/O DTR, and QPI
  - Software & Hardware Reset
  - Supports Serial Flash Discoverable Parameters (SFDP)

- **High Performance Serial Flash (SPI)**
  - 80MHz Normal Read
  - Up to166Mhz Fast Read:
    -166MHz at Vcc=2.7V to 3.6V (1)
    -133MHz at Vcc=2.3V to 3.6V
    -104MHz at Vcc=1.65V to 1.95V
  - Up to 80MHz DTR (Dual Transfer Rate)
  - Equivalent Throughput of 664 Mb/s
  - Selectable Dummy Cycles
  - Configurable Drive Strength
  - Supports SPI Modes 0 and 3
  - More than 100,000 Erase/Program Cycles
  - More than 20-year Data Retention

- **Flexible & Efficient Memory Architecture**
  - Chip Erase with Uniform Sector/Block Erase (4/32/64 Kbyte)
  - Program 1 to 256 Byte per Page
  - Program/Erase Suspend & Resume

- **Efficient Read and Program modes**
  - Low Instruction Overhead Operations
  - Continuous Read 8/16/32/64 Byte Burst Wrap
  - Selectable Burst Length
  - QPI for Reduced Instruction Overhead
  - AutoBoot Operation

- **Low Power with Wide Temp. Ranges**
  - Single Voltage Supply
    IS25LP: 2.30V to 3.60V
    IS25WP: 1.65V to 1.95V
  - 7.5 mA Active Read Current
  - 10 µA Standby Current
  - 1 µA Deep Power Down
  - Temp Grades:
    Extended: -40°C to +105°C
    Auto Grade (A3): 40°C to +125°C

- **Advanced Security Protection**
  - Software and Hardware Write Protection
  - Advanced Sector/Block Protection
  - Top/Bottom Block Protection
  - Power Supply Lock Protection
  - 4x256 Byte Dedicated Security Area with OTP User-lockable Bits
  - 128 bit Unique ID for Each Device (Call Factory)

- **Industry Standard Pin-out & Packages**
  - M =16-pin SOIC 300mil
  - L = 8-contact WSON 8x6mm
  - G = 24-ball TFBGA (4x6 ball array)
  - H = 24-ball TFBGA (5x5 ball array)
  - KGD (Call Factory)

Note:
1. 166MHz at Mode 0 , and 133MHz at Mode 3

GENERAL DESCRIPTION

The IS25LP256D and IS25WP256D Serial Flash memory offers a versatile storage solution with high flexibility and performance in a simplified pin count package. ISSI's "Industry Standard Serial Interface" Flash is for systems that require limited space, a low pin count, and low power consumption. The device is accessed through a 4-wire SPI Interface consisting of a Serial Data Input (SI), Serial Data Output (SO), Serial Clock (SCK), and Chip Enable (CE#) pins, which can also be configured to serve as multi-I/O (see pin descriptions).

The device supports Dual and Quad I/O as well as standard, Dual Output, and Quad Output SPI. Clock frequencies of up to 166MHz allow for equivalent clock rates of up to 664MHz (166MHz x 4) which equates to 83Mbytes/s of data throughput. The IS25xP series of Flash adds support for DTR (Double Transfer Rate) commands that transfer addresses and read data on both edges of the clock. These transfer rates can outperform 16-bit Parallel Flash memories allowing for efficient memory access to support XIP (execute in place) operation.

The memory array is organized into programmable pages of 256 bytes. This family supports page program mode where 1 to 256 bytes of data are programmed in a single command. QPI (Quad Peripheral Interface) supports 2-cycle instruction further reducing instruction times. Pages can be erased in groups of 4Kbyte sectors, 32Kbyte blocks, 64Kbyte blocks, and/or the entire chip. The uniform sector and block architecture allows for a high degree of flexibility so that the device can be utilized for a broad variety of applications requiring solid data retention.

GLOSSARY

Standard SPI
In this operation, a 4-wire SPI Interface is utilized, consisting of Serial Data Input (SI), Serial Data Output (SO), Serial Clock (SCK), and Chip Enable (CE#) pins. Instructions are sent via the SI pin to encode instructions, addresses, or input data to the device on the rising edge of SCK. The SO pin is used to read data or to check the status of the device. This device supports SPI bus operation modes (0,0) and (1,1).

Mutil I/O SPI
Multi-I/O operation utilizes an enhanced SPI protocol to allow the device to function with Dual Output, Dual Input and Output, Quad Output, and Quad Input and Output capability. Executing these instructions through SPI mode will achieve double or quadruple the transfer bandwidth for READ and PROGRAM operations.

Quad I/O QPI
The device enables QPI protocol by issuing an "Enter QPI mode (35h)" command. The QPI mode uses four IO pins for input and output to decrease SPI instruction overhead and increase output bandwidth. SI and SO pins become bidirectional IO0 and IO1, and WP# and HOLD# pins become IO2 and IO3 respectively during QPI mode. Issuing an "Exit QPI (F5h) command will cause the device to exit QPI mode. Power Reset or Hardware/Software Reset can also return the device into the standard SPI mode.

DTR
In addition to SPI and QPI features, the device also supports Fast READ DTR operation. Fast READ DTR operation allows high data throughput while running at lower clock frequencies. Fast READ DTR operation uses both rising and falling edges of the clock for address inputs, and data outputs, resulting in reducing input and output cycles by half.

Programmable drive strength and Selectable burst setting.
The device offers programmable output drive strength and selectable burst (wrap) length features to increase the efficiency and performance of READ operation. The driver strength and burst setting features are controlled by setting the Read Registers. A total of six different drive strengths and four different burst sizes (8/16/32/64 Byte) are available for selection.

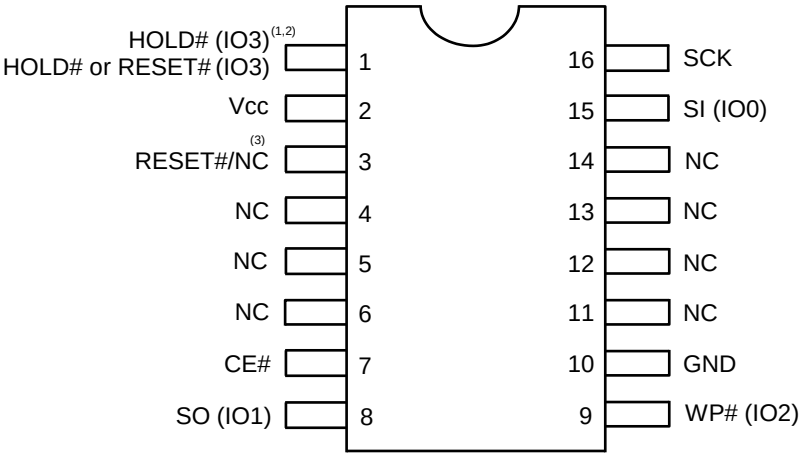
TABLE OF CONTENTS

FEATURES 2
GENERAL DESCRIPTION 3
TABLE OF CONTENTS 4
1. PIN CONFIGURATION 7
2. PIN DESCRIPTIONS 9
3. BLOCK DIAGRAM 11
4. SPI MODES DESCRIPTION 12
5. SYSTEM CONFIGURATION 14
5.1 BLOCK/SECTOR ADDRESSES 14
6. REGISTERS 15
6.1 STATUS REGISTER 15
6.2 FUNCTION REGISTER 19
6.3 READ REGISTER AND EXTENDED REGISTER 20
6.3.1 READ REGISTER 20
6.3.2 EXTENDED READ REGISTER 23
6.4 AUTOBOOT REGISTER 25
6.5 BANK ADDRESS REGISTER 26
6.6 ADVANCED SECTOR/BLOCK PROTECTION (ASP) RELATED REGISTER 27
6.6.1 ADVANCED SECTOR/BLOCK PROTECTION REGISTER (ASPR) 27
6.6.2 PASSWORD REGISTER 28
6.6.3 PPB LOCK REGISTER 28
6.6.4 PPB REGISTER 29
6.6.5 DYB REGISTER 29
7. PROTECTION MODE 30
7.1 HARDWARE WRITE PROTECTION 30
7.2 SOFTWARE WRITE PROTECTION 30
7.2.1 BLOCK PROTECTION BITS 30
7.2.2 ADVANCED SECTOR/BLOCK PROTECTION (ASP) 31
8. DEVICE OPERATION 38
8.1 COMMAND OVERVIEW 38
8.2 COMMAND SET SUMMARY 39
8.3 NORMAL READ OPERATION (NORD, 03h or 4NORD, 13h) 48
8.4 FAST READ OPERATION (FRD, 0Bh or 4FRD, 0Ch) 51
8.5 HOLD OPERATION 55
8.6 FAST READ DUAL I/O OPERATION (FRDIO, BBh or 4FRDIO, BCh) 56
8.7 FAST READ DUAL OUTPUT OPERATION (FRDO, 3Bh or 4FRDO, 3Ch) 60
8.8 FAST READ QUAD OUTPUT OPERATION (FRQO, 6Bh or 4FRQO 6Ch) 63
8.9 FAST READ QUAD I/O OPERATION (FRQIO, EBh or 4FRQIO, ECh) 66

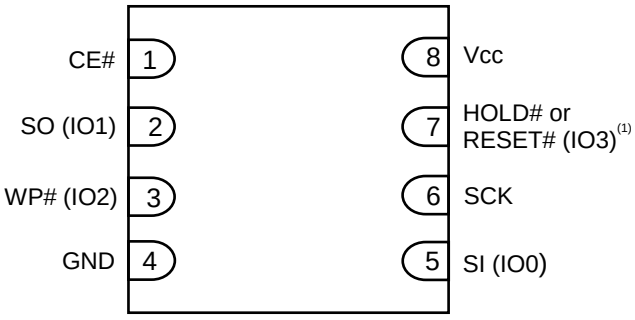
8.10 PAGE PROGRAM OPERATION (PP, 02h or 4PP, 12h) 73
8.11 QUAD INPUT PAGE PROGRAM OPERATION (PPQ, 32h/38h or 4PPQ, 34h/3Eh) 76
8.12 ERASE OPERATION 77
8.13 SECTOR ERASE OPERATION (SER, D7h/20h or 4SER, 21h) 78
8.14 BLOCK ERASE OPERATION (BER32K:52h or 4BER32K:5Ch, BER64K:D8h or 4BER64K:DCh) 80
8.15 CHIP ERASE OPERATION (CER, C7h/60h) 83
8.16 WRITE ENABLE OPERATION (WREN, 06h) 84
8.17 WRITE DISABLE OPERATION (WRDI, 04h) 85
8.18 READ STATUS REGISTER OPERATION (RDSR, 05h) 86
8.19 WRITE STATUS REGISTER OPERATION (WRSR, 01h) 87
8.20 READ FUNCTION REGISTER OPERATION (RDFR, 48h) 88
8.21 WRITE FUNCTION REGISTER OPERATION (WRFR, 42h) 89
8.22 ENTER QUAD PERIPHERAL INTERFACE (QPI) MODE OPERATION (QIOEN,35h; QIODI,F5h) 90
8.23 PROGRAM/ERASE SUSPEND & RESUME 91
8.24 ENTER DEEP POWER DOWN (DP, B9h) 94
8.25 RELEASE DEEP POWER DOWN (RDPD, ABh) 95
8.26 SET READ PARAMETERS OPERATION (SRPNV: 65h, SRPV: C0h/63h) 96
8.27 SET EXTENDED READ PARAMETERS OPERATION (SERPNV: 85h, SERPV: 83h) 98
8.28 READ READ PARAMETERS OPERATION (RDRP, 61h) 99
8.29 READ EXTENDED READ PARAMETERS OPERATION (RDERP, 81h) 100
8.30 CLEAR EXTENDED READ REGISTER OPERATION (CLERP, 82h) 101
8.31 READ PRODUCT IDENTIFICATION (RDID, ABh) 102
8.32 READ PRODUCT IDENTIFICATION BY JEDEC ID OPERATION (RDJDID, 9Fh; RDJDIDQ, AFh) 104
8.33 READ DEVICE MANUFACTURER AND DEVICE ID OPERATION (RDMDID, 90h) 105
8.34 READ UNIQUE ID NUMBER (RDUID, 4Bh) 106
8.35 READ SFDP OPERATION (RDSFDP, 5Ah) 107
8.36 NO OPERATION (NOP, 00h) 107
8.37 SOFTWARE RESET (RESET-ENABLE (RSTEN, 66h) AND RESET (RST, 99h)) AND HARDWARE RESET 108
8.38 SECURITY INFORMATION ROW 109
8.39 INFORMATION ROW ERASE OPERATION (IRER, 64h) 110
8.40 INFORMATION ROW PROGRAM OPERATION (IRP, 62h) 111
8.41 INFORMATION ROW READ OPERATION (IRRD, 68h) 112
8.42 FAST READ DTR MODE OPERATION (FRDTR, 0Dh or 4FRDTR, 0Eh) 113
8.43 FAST READ DUAL IO DTR MODE OPERATION (FRDDTR, BDh or 4FRDDTR, BEh) 118
8.44 FAST READ QUAD IO DTR MODE OPERATION IN SPI MODE (FRQDTR, EDh or 4FRQDTR, EEh) 122
8.45 SECTOR LOCK/UNLOCK FUNCTIONS 130
8.46 AUTOBOOT 133
8.47 READ BANK ADDRESS REGISTER OPERATION (RDBR: 16h/C8h) 137

8.48 WRITE BANK ADDRESS REGISTER OPERATION (WRBRNV: 18h, WRBRV: 17h/C5h) 138
8.49 ENTER 4-BYTE ADDRESS MODE OPERATION (EN4B, B7h) 139
8.50 EXIT 4-BYTE ADDRESS MODE OPERATION (EX4B, 29h) 140
8.51 READ DYB OPERATION (RDDYB, FAh or 4RDDYB, E0h) 141
8.52 WRITE DYB OPERATION (WRDYB, FBh or 4WRDYB, E1h) 143
8.53 READ PPB OPERATION (RDPPB, FCh or 4RDPPB, E2h) 145
8.54 PROGRAM PPB OPERATION (PGPPB, FDh or 4PGPPB, E3h) 146
8.55 ERASE PPB OPERATION (ERPPB, E4h) 148
8.56 READ ASP OPERATION (RDASP, 2Bh) 149
8.57 PROGRAM ASP OPERATION (PGASP, 2Fh) 150
8.58 READ PPB LOCK BIT OPERATION (RDPLB, A7h) 151
8.59 WRITE PPB LOCK BIT OPERATION (WRPLB, A6h) 152
8.60 SET FREEZE BIT OPERATION (SFRZ, 91h) 153
8.61 READ PASSWORD OPERATION (RDPWD, E7h) 154
8.62 PROGRAM PASSWORD OPERATION (PGPWD, E8h) 155
8.63 UNLOCK PASSWORD OPERATION (UNPWD, E9h) 156
8.64 GANG SECTOR/BLOCK LOCK OPERATION (GBLK, 7Eh) 157
8.65 GANG SECTOR/BLOCK UNLOCK OPERATION (GBUN, 98h) 158
9. ELECTRICAL CHARACTERISTICS 159
9.1 ABSOLUTE MAXIMUM RATINGS (1) 159
9.2 OPERATING RANGE 159
9.3 DC CHARACTERISTICS 160
9.4 AC MEASUREMENT CONDITIONS 161
9.5 PIN CAPACITANCE 161
9.6 AC CHARACTERISTICS 162
9.7 SERIAL INPUT/OUTPUT TIMING 164
9.8 POWER-UP AND POWER-DOWN 166
9.9 PROGRAM/ERASE PERFORMANCE 167
9.10 RELIABILITY CHARACTERISTICS 167
10. PACKAGE TYPE INFORMATION 168
10.1 8-CONTACT ULTRA-THIN SMALL OUTLINE NO-LEAD (WSON) PACKAGE 8x6mm (L) 168
10.2 16-LEAD PLASTIC SMALL OUTLINE PACKAGE (300 MILS BODY WIDTH) (M) 169
10.3 24-BALL THIN PROFILE FINE PITCH BGA 6x8mm 4x6 BALL ARRAY (G) 170
10.4 24-BALL THIN PROFILE FINE PITCH BGA 6x8mm 5x5 BALL ARRAY (H) 171
11. ORDERING INFORMATION – Valid Part Numbers 172

1. PIN CONFIGURATION

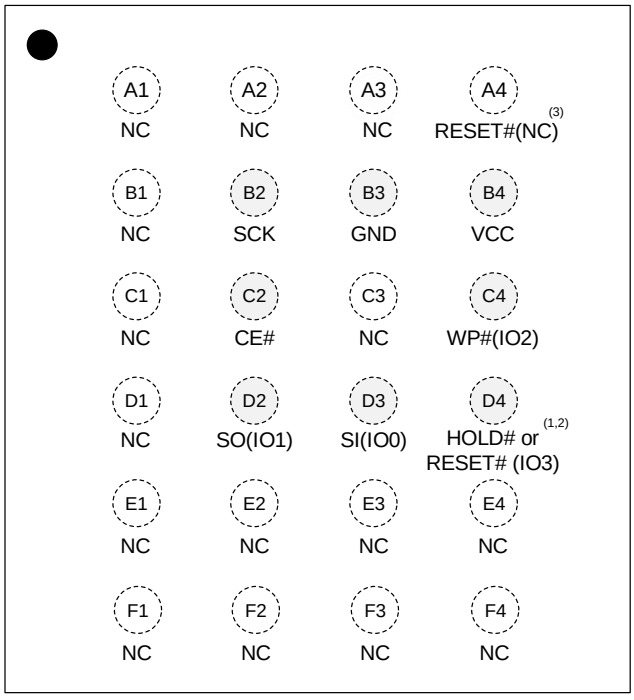


16-pin SOIC 300mil



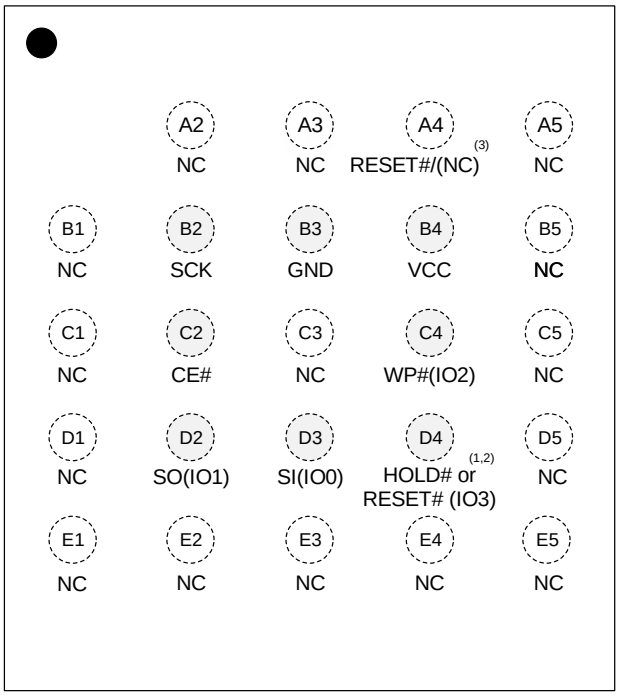
8-contact WSON 8x6mm

Top View, Balls Facing Down



24-ball TFBGA, 4x6 Ball Array (Package:G)

Top View, Balls Facing Down



24-ball TFBGA, 5x5 Ball Array (Package:H)

Notes:

1. **The pin can be configured as Hold# or Reset# by setting P7 bit of the Read Register. Pin default is Hold#.**
2. **Dedicated RESET# pin is available in devices with a dedicated part number, in these devices Pin 1 (16-pin SOIC) or ball D4 (24-ball TFBGA) are set to Hold# regardless of P7 setting of the Read Register.**

3. **For compatibility, the pin can be disabled on dedicated part numbers by setting Bit0 (Dedicated RESET# Disable) of the Function Register to "1" (default is "0" from factory on dedicated part numbers). An internal pull-up resistor exists and the pin may be left floating if not used.**

16-pin SOIC / 24-ball TFBGA	Device with HOLD#/RESET#	Device with dedicated RESET#
Pin1 / Ball D4	Hold#(IO3) or RESET#(IO3) by P7 bit setting	Hold#(IO3) only regardless of P7 bit setting
Pin3 / Ball A4	NC	RESET#
Part Number Option	J	R or P

2. PIN DESCRIPTIONS

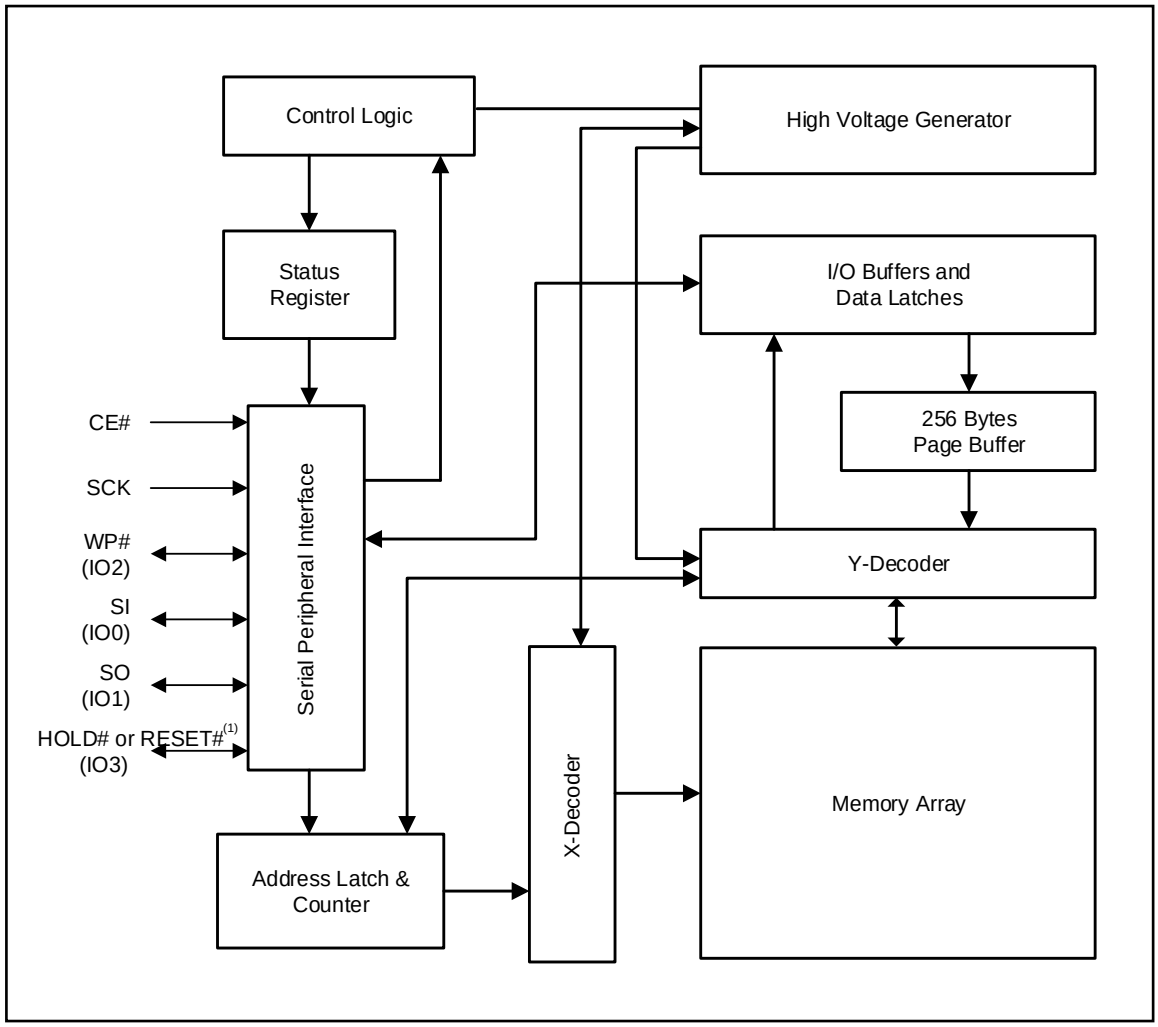
For the device with Hold#/RESET#

SYMBOL	TYPE	DESCRIPTION
CE#	INPUT	**Chip Enable:** The Chip Enable (CE#) pin enables and disables the devices operation. When CE# is high the device is deselected and output pins are in a high impedance state. When deselected the devices non-critical internal circuitry power down to allow minimal levels of power consumption while in a standby state. When CE# is pulled low the device will be selected and brought out of standby mode. The device is considered active and instructions can be written to, data read, and written to the device. After power-up, CE# must transition from high to low before a new instruction will be accepted. Keeping CE# in a high state deselects the device and switches it into its low power state. Data will not be accepted when CE# is high.
SI (IO0), SO (IO1)	INPUT/ OUTPUT	**Serial Data Input, Serial Output, and IOs (SI, SO, IO0, and IO1):** This device supports standard SPI, Dual SPI, and Quad SPI operation. Standard SPI instructions use the unidirectional SI (Serial Input) pin to write instructions, addresses, or data to the device on the rising edge of the Serial Clock (SCK). Standard SPI also uses the unidirectional SO (Serial Output) to read data or status from the device on the falling edge of the serial clock (SCK). In Dual and Quad SPI mode, SI and SO become bidirectional IO pins to write instructions, addresses or data to the device on the rising edge of the Serial Clock (SCK) and read data or status from the device on the falling edge of SCK. Quad SPI instructions use the WP# and HOLD# pins as IO2 and IO3 respectively.
WP# (IO2)	INPUT/ OUTPUT	**Write Protect/Serial Data IO (IO2):** The WP# pin protects the Status Register from being written in conjunction with the SRWD bit. When the SRWD is set to “1” and the WP# is pulled low, the Status Register bits (SRWD, QE, BP3, BP2, BP1, BP0) are write-protected and vice-versa for WP**#** high. When the SRWD is set to “0”, the Status Register is not write-protected regardless of WP# state. When the QE bit is set to “1”, the WP**#** pin (Write Protect) function is not available since this pin is used for IO2.
HOLD# or RESET# (IO3)	INPUT/ OUTPUT	**HOLD# or RESET#/Serial Data IO (IO3):** When the QE bit of Status Register is set to “1”, HOLD**#** pin or RESET**#** is not available since it becomes IO3. When QE=0 the pin acts as HOLD# or RESET# and either one can be selected by the P7 bit setting in Read Register. HOLD# will be selected if P7=0 (Default) and RESET# will be selected if P7=1. The HOLD# pin allows the device to be paused while it is selected. It pauses serial communication by the master device without resetting the serial sequence. The HOLD# pin is active low. When HOLD# is in a low state and CE# is low, the SO pin will be at high impedance. Device operation can resume when HOLD# pin is brought to a high state. RESET# pin is a hardware RESET signal. When RESET# is driven HIGH, the memory is in the normal operating mode. When RESET# is driven LOW, the memory enters reset mode and output is High-Z. If RESET# is driven LOW while an internal WRITE, PROGRAM, or ERASE operation is in progress, data may be lost.
SCK	INPUT	**Serial Data Clock:** Synchronized Clock for input and output timing operations.
Vcc	POWER	**Power:** Device Core Power Supply
GND	GROUND	**Ground:** Connect to ground when referenced to Vcc
NC	Unused	**NC:** Pins labeled “NC” stand for “No Connect” and should be left uncommitted.

For the device with dedicated RESET#

SYMBOL	TYPE	DESCRIPTION
CE#	INPUT	Same as the description in previous page
SI (IO0), SO (IO1)	INPUT/ OUTPUT	Same as the description in previous page
WP# (IO2)	INPUT/ OUTPUT	Same as the description in previous page
HOLD# (IO3)	INPUT/ OUTPUT	**HOLD#/Serial Data IO (IO3):** When the QE bit of Status Register is set to "1", HOLD**#** pin is not available since it becomes IO3. When QE=0 the pin acts as HOLD# regardless of the P7 bit of Read Register. The HOLD# pin allows the device to be paused while it is selected. It pauses serial communication by the master device without resetting the serial sequence. The HOLD# pin is active low. When HOLD# is in a low state and CE# is low, the SO pin will be at high impedance. Device operation can resume when HOLD# pin is brought to a high state.
RESET#	INPUT	**RESET#:** This pin is available only for dedicated parts (Call Factory for 24-ball TFBGA 6x8mm packages). The RESET# pin is a hardware RESET signal. When RESET# is driven HIGH, the memory is in the normal operating mode. When RESET# is driven LOW, the memory enters reset mode and output is High-Z. If RESET# is driven LOW while an internal WRITE, PROGRAM, or ERASE operation is in progress, data may be lost.
SCK	INPUT	Same as the description in previous page
Vcc	POWER	Same as the description in previous page
GND	GROUND	Same as the description in previous page
NC	Unused	Same as the description in previous page

3. BLOCK DIAGRAM



Notes:

1. **According to the P7 bit setting in Read Register, either HOLD# (P7=0) or RESET# (P7=1) pin can be selected.**
2. **SI and SO pins become bidirectional IO0 and IO1 respectively during Dual I/O mode and SI, SO, WP#, and HOLD#/RESET# pins become bidirectional IO0, IO1, IO2, and IO3 respectively during Quad I/O or QPI mode.**
3. **In case of 16-pin SOIC and 24-ball TFBGA packages, dedicated RESET# function is supported without sharing with HOLD# pin for the dedicated parts. See the Ordering Information for the dedicated RESET# pin**

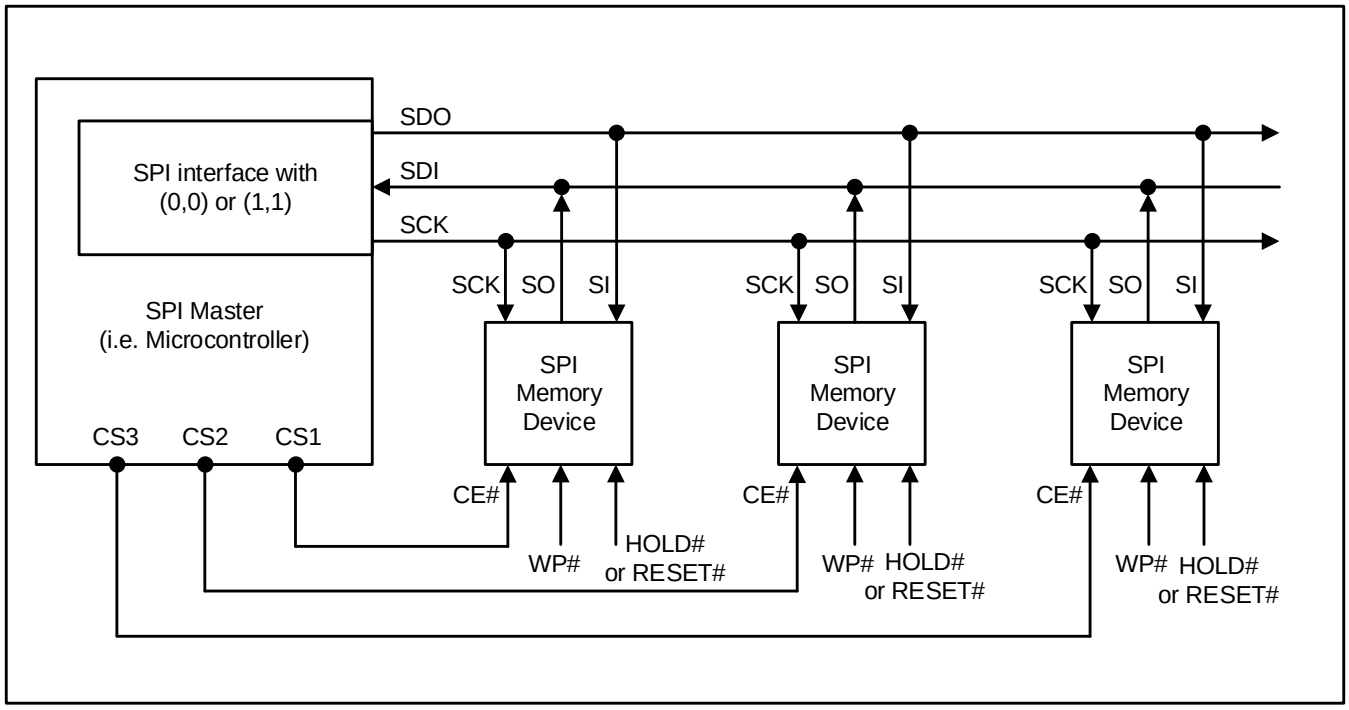
4. SPI MODES DESCRIPTION

Multiple IS25LP256D devices or multiple IS25WP256D devices can be connected on the SPI serial bus and controlled by a SPI Master, i.e. microcontroller, as shown in Figure 4.1. The devices support either of two SPI modes:

Mode 0 (0, 0)
Mode 3 (1, 1)

The difference between these two modes is the clock polarity. When the SPI master is in stand-by mode, the serial clock remains at "0" (SCK = 0) for Mode 0 and the clock remains at "1" (SCK = 1) for Mode 3. Please refer to Figure 4.2 and Figure 4.3 for SPI and QPI mode. In both modes, the input data is latched on the rising edge of Serial Clock (SCK), and the output data is available from the falling edge of SCK.

Figure 4.1 Connection Diagram among SPI Master and SPI Slaves (Memory Devices)



Notes:

1. **According to the P7 bit setting in Read Register, either HOLD# (P7=0) or RESET# (P7=1) pin can be selected.**
2. **SI and SO pins become bidirectional IO0 and IO1 respectively during Dual I/O mode and SI, SO, WP#, and HOLD#/RESET# pins become bidirectional IO0, IO1, IO2, and IO3 respectively during Quad I/O or QPI mode.**
3. **In case of 16-pin SOIC and 24-ball TFBGA packages, dedicated RESET# function is supported without sharing with HOLD# pin for the dedicated parts. See the Ordering Information for the dedicated RESET# pin**

Figure 4.2 SPI Mode Support

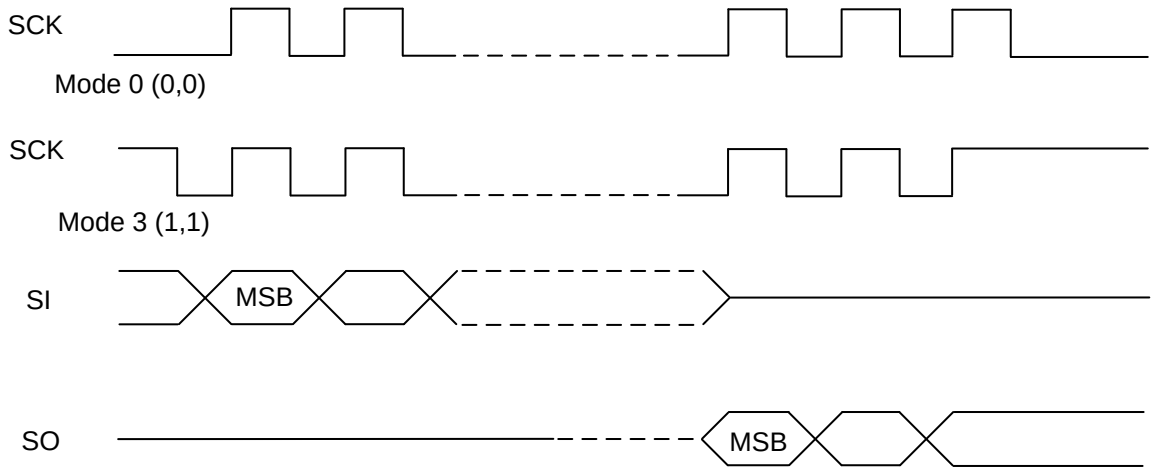
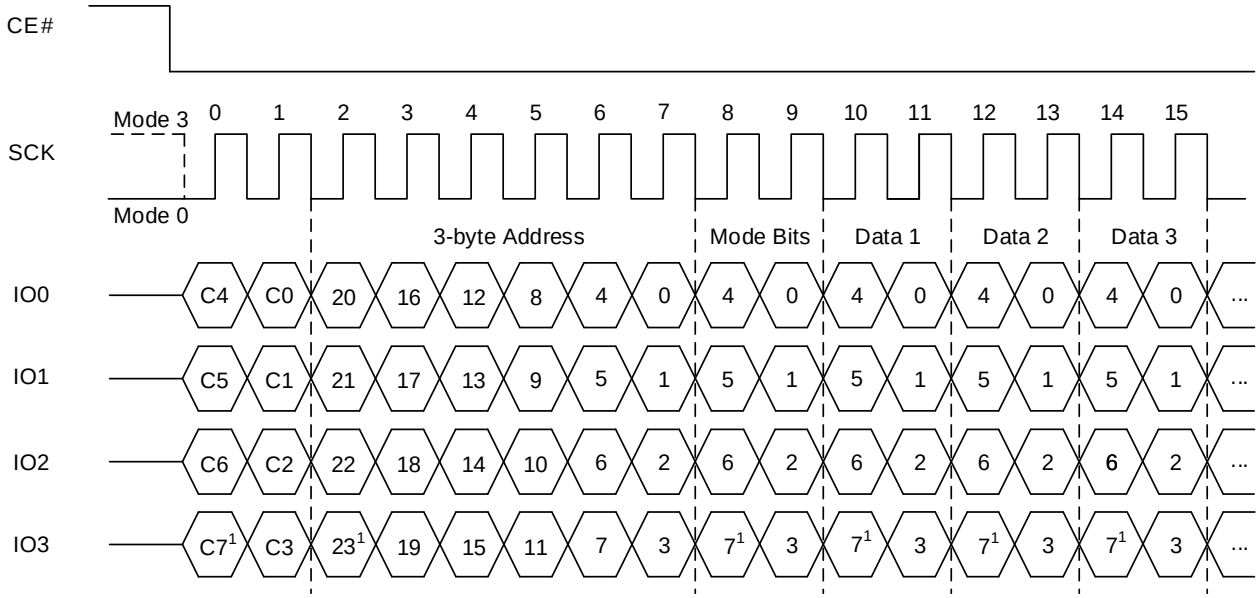


Figure 4.3 QPI Mode Support



Note1: MSB (Most Significant Bit)

5. SYSTEM CONFIGURATION

The memory array is divided into uniform 4 Kbyte sectors or uniform 32/64 Kbyte blocks (a block consists of eight/sixteen adjacent sectors respectively).

Table 5.1 illustrates the memory map of the device. The Status Register controls how the memory is protected.

5.1 BLOCK/SECTOR ADDRESSES

Table 5.1 Block/Sector Addresses

Memory Density	Block No. (64Kbyte)	Block No. (32Kbyte)	Sector No.	Sector Size (Kbyte)	Address Range
256Mb	Block 0	Block 0	Sector 0	4	000000h - 000FFFh
			:	:	:
		Block 1	:	:	:
			Sector 15	4	00F000h - 00FFFFh
	Block 1	Block 2	Sector 16	4	010000h - 010FFFh
			:	:	:
		Block 3	:	:	:
			Sector 31	4	01F000h - 01FFFFh
	Block 2	Block 4	Sector 32	4	020000h - 020FFFh
			:	:	:
		Block 5	:	:	:
			Sector 47	4	02F000h - 02FFFFh
	:	:	:	:	:
	Block 254	Block 508	Sector 4064	4	FE0000h – FE0FFFh
			:	:	:
		Block 509	:	:	:
			Sector 4079	4	FEF000h – FEFFFFh
	Block 255	Block 510	Sector 4080	4	FF0000h – FF0FFFh
			:	:	:
		Block 511	:	:	:
			Sector 4095	4	FFF000h – FFFFFFh
	:	:	:	:	:
	Block 510	Block 1020	Sector 8160	4	1FE0000h – 1FE0FFFh
			:	:	:
		Block 1021	:	:	:
			Sector 8175	4	1FEF000h – 1FEFFFFh
	Block 511	Block 1022	Sector 8176	4	1FF0000h – 1FF0FFFh
			:	:	:
		Block 1023	:	:	:
			Sector 8191	4	1FFF000h – 1FFFFFFh

6. REGISTERS

The device has many sets of Registers such as Status, Function, Read, AutoBoot, and so on.

6.1 STATUS REGISTER

Status Register Format and Status Register Bit Definitions are described in Tables 6.1 & 6.2.

Table 6.1 Status Register Format

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	SRWD	QE	BP3	BP2	BP1	BP0	WEL	WIP
Default	0	0	0	0	0	0	0	0

Table 6.2 Status Register Bit Definition

Bit	Name	Definition	Read-/Write	Type
Bit 0	WIP	Write In Progress Bit: "0" indicates the device is ready (default) "1" indicates a write cycle is in progress and the device is busy	R	Volatile
Bit 1	WEL	Write Enable Latch: "0" indicates the device is not write enabled (default) "1" indicates the device is write enabled	R/W[1]	Volatile
Bit 2	BP0	Block Protection Bit: (See Tables 6.4 for details) "0" indicates the specific blocks are not write-protected (default) "1" indicates the specific blocks are write-protected	R/W	Non-Volatile
Bit 3	BP1			
Bit 4	BP2			
Bit 5	BP3			
Bit 6	QE	Quad Enable bit: “0” indicates the Quad output function disable (default) “1” indicates the Quad output function enable	R/W	Non-Volatile
Bit 7	SRWD	Status Register Write Disable: (See Table 7.1 for details) "0" indicates the Status Register is not write-protected (default) "1" indicates the Status Register is write-protected	R/W	Non-Volatile

Note: WEL bit can be written by WREN and WRDI commands, but cannot by WRSR command.

The BP0, BP1, BP2, BP3, QE, and SRWD are non-volatile memory cells that can be written by a Write Status Register (WRSR) instruction. The default value of the BP0, BP1, BP2, BP3, QE, and SRWD bits were set to “0” at factory. The Status Register can be read by the Read Status Register (RDSR).

The function of Status Register bits are described as follows:

WIP bit: Write In Progress (WIP) is read-only, and can be used to detect the progress or completion of a Program, Erase, Write/Set Non-Volatile/OTP Register, or Gang Sector/Block Lock/Unlock operation. WIP is set to “1” (busy state) when the device is executing the operation. During this time the device will ignore further instructions except for Read Status/Function/Extended Read Register and Software/Hardware Reset instructions. In addition to the instructions, an Erase/Program Suspend instruction also can be executed during a Program or Erase operation. When an operation has completed, WIP is cleared to “0” (ready state) whether the operation is successful or not and the device is ready for further instructions.

WEL bit: Write Enable Latch (WEL) indicates the status of the internal write enable latch. When WEL is “0”, the internal write enable latch is disabled and the Write operations described in Table 6.3 are inhibited. When WEL is “1”, the Write operations are allowed. WEL is set by a Write Enable (WREN) instruction. Each Write Non-Volatile Register, Program and Erase instruction must be preceded by a WREN instruction. The volatile register related commands such as the Set Volatile Read Register, the Set Volatile Extended Read Register, the Write Volatile Bank

Address Register, and WRDYB don't require to set WEL to "1". WEL can be reset by a Write Disable (WRDI) instruction. It will automatically reset after the completion of any Write operation.

Table 6.3 Instructions requiring WREN instruction ahead

Instructions must be preceded by the WREN instruction		
Name	**Hex Code**	**Operation**
PP	02h	Serial Input Page Program (3-byte or 4-byte Address)
4PP	12h	Serial Input Page Program (4-byte Address)
PPQ	32h/38h	Quad Input Page Program (3-byte or 4-byte Address)
4PPQ	34h/3Eh	Quad Input Page Program (4-byte Address)
SER	D7h/20h	Sector Erase 4KB (3-byte or 4-byte Address)
4SER	21h	Sector Erase 4KB (4-byte Address)
BER32 (32KB)	52h	Block Erase 32KB (3-byte or 4-byte Address)
4BER32 (32KB)	5Ch	Block Erase 32KB (4-byte Address)
BER64 (64KB)	D8h	Block Erase 64KB (3-byte or 4-byte Address)
4BER64 (64KB)	DCh	Block Erase 64KB (4-byte Address)
CER	C7h/60h	Chip Erase
WRSR	01h	Write Status Register
WRFR	42h	Write Function Register
SRPNV	65h	Set Read Parameters (Non-Volatile)
SERPNV	85h	Set Extended Read Parameters (Non-Volatile)
IRER	64h	Erase Information Row
IRP	62h	Program Information Row
WRABR	15h	Write AutoBoot Register
WRBRNV	18h	Write Non-Volatile Bank Address Register
PGPPB	FDh	Write PPB (3-byte or 4-byte Address)
4PGPPB	E3h	Write PPB (4-byte Address)
ERPPB	E4h	Erase PPB
PGASP	2Fh	Program ASP
WRPLB	A6h	Write PPB Lock Bit
SFRZ	91h	Set FREEZE bit
PGPWD	E8h	Program Password

BP3, BP2, BP1, BP0 bits: The Block Protection (BP3, BP2, BP1 and BP0) bits are used to define the portion of the memory area to be protected. Refer to Table 6.4 for the Block Write Protection (BP) bit settings. When a defined combination of BP3, BP2, BP1 and BP0 bits are set, the corresponding memory area is protected. Any program or erase operation to that area will be inhibited.

Note: A Chip Erase (CER) instruction will be ignored unless all the Block Protection Bits are "0"s.

SRWD bit: The Status Register Write Disable (SRWD) bit operates in conjunction with the Write Protection (WP#) signal to provide a Hardware Protection Mode. When the SRWD is set to "0", the Status Register is not write-protected. When the SRWD is set to "1" and the WP**#** is pulled low (V_{IL}), the bits of Status Register (SRWD, QE, BP3, BP2, BP1, BP0) become read-only, and a WRSR instruction will be ignored. If the SRWD is set to "1" and WP# is pulled high (V_{IH}), the Status Register can be changed by a WRSR instruction.

QE bit: The Quad Enable (QE) is a non-volatile bit in the Status Register that allows quad operation. When the QE bit is set to "0", the pin WP# and HOLD#/RESET# are enabled. When the QE bit is set to "1", the IO2 and IO3 pins are enabled.

WARNING: The QE bit must be set to "0" if WP# or HOLD#/RESET# pin (or ball) is tied directly to the power supply.

Table 6.4 Block (64Kbyte) assignment by Block Write Protect (BP) Bits

Status Register Bits				Protected Memory Area (256M, 512Blocks)	
BP3	**BP2**	**BP1**	**BP0**	**TBS(T/B selection) = 0, Top area**	**TBS(T/B selection) = 1, Bottom area**
0	0	0	0	0 (None)	0 (None)
0	0	0	1	1 (1 block: 511st)	1 (1 block: 0th)
0	0	1	0	2 (2 blocks: 510th and 511st)	2 (2 blocks: 0th and 1st)
0	0	1	1	3 (4 blocks: 508th to 511st)	3 (4 blocks: 0th to 3rd)
0	1	0	0	4 (8 blocks: 504th to 511st)	4 (8 blocks: 0th to 7th)
0	1	0	1	5 (16 blocks: 496th to 511st)	5 (16 blocks: 0th to 15th)
0	1	1	0	6 (32 blocks: 480th to 511st)	6 (32 blocks: 0th to 31st)
0	1	1	1	7 (64 blocks: 448th to 511st)	7 (64 blocks: 0th to 63rd)
1	0	0	0	8 (128 blocks: 384th to 511st)	8 (128 blocks: 0th to 127th)
1	0	0	1	9 (256 blocks: 256th to 511st)	9 (256 blocks: 0th to 255th)
1	0	1	x	10-11 (512 blocks: 0th to 511st) All blocks	10-11 (512 blocks: 0th to 511st) All blocks
1	1	x	x	12-15 (512 blocks: 0th to 511st) All blocks	12-15 (512 blocks: 0th to 511st) All blocks

Note: x is don't care

6.2 FUNCTION REGISTER

Function Register Format and Bit definition are described in Table 6.5 and Table 6.6.

Table 6.5 Function Register Format

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	IRL3	IRL2	IRL1	IRL0	ESUS	PSUS	TBS	Dedicated RESET# Disable
Default	0	0	0	0	0	0	0	0 or 1

Table 6.6 Function Register Bit Definition

Bit	Name	Definition	Read /Write	Type
Bit 0	Dedicated RESET# Disable	Dedicated RESET# Disable bit “0” indicates Dedicated RESET# was enabled “1” indicates Dedicated RESET# was disabled	R/W for 0 R for 1	OTP
Bit 1	TBS	Top/Bottom Selection. (See Table 6.4 for details) “0” indicates Top area. “1” indicates Bottom area.	R/W	OTP
Bit 2	PSUS	Program suspend bit: “0” indicates program is not suspend “1” indicates program is suspend	R	Volatile
Bit 3	ESUS	Erase suspend bit: "0" indicates Erase is not suspend "1" indicates Erase is suspend	R	Volatile
Bit 4	IR Lock 0	Lock the Information Row 0: “0” indicates the Information Row can be programmed “1” indicates the Information Row cannot be programmed	R/W	OTP
Bit 5	IR Lock 1	Lock the Information Row 1: “0” indicates the Information Row can be programmed “1” indicates the Information Row cannot be programmed	R/W	OTP
Bit 6	IR Lock 2	Lock the Information Row 2: “0” indicates the Information Row can be programmed “1” indicates the Information Row cannot be programmed	R/W	OTP
Bit 7	IR Lock 3	Lock the Information Row 3: “0” indicates the Information Row can be programmed “1” indicates the Information Row cannot be programmed	R/W	OTP

Note: Once OTP bits of Function Register are written to “1”, it cannot be modified to “0” any more.

Dedicated RESET# Disable bit: The default status of the bit is dependent on part number. The device with dedicated RESET# can be programmed to “1” to disable dedicated RESET**#** function to move RESET**#** function to Hold#/RESET# pin (or ball). So the device with dedicated RESET# can be used for dedicated RESET# application and HOLD#/RESET# application.

TBS bit: BP0~3 area assignment can be changed from Top (default) to Bottom by setting TBS bit to “1”. However, once Bottom is selected, it cannot be changed back to Top since TBS bit is OTP. See Table 6.4 for details.

PSUS bit: The Program Suspend Status bit indicates when a Program operation has been suspended. The PSUS changes to “1” after a suspend command is issued during the program operation. Once the suspended Program resumes, the PSUS bit is reset to “0”.

ESUS bit: The Erase Suspend Status bit indicates when an Erase operation has been suspended. The ESUS bit is “1” after a suspend command is issued during an Erase operation. Once the suspended Erase resumes, the ESUS bit is reset to “0”.

IR Lock bit 0 ~ 3: The default is “0” so that the Information Row can be programmed. If the bit is set to “1”, it cannot be changed back to “0” again since IR Lock bits are OTP.

6.3 READ REGISTER AND EXTENDED REGISTER

Read Register format and bit definitions are described below. Read Register and Extended Read Register consist of a pair of rewritable non-volatile register and volatile register, respectively. During power up sequence, volatile register will be loaded with the value of non-volatile value.

6.3.1 READ REGISTER

Table 6.7 and Table 6.8 define all bits that control features in SPI/QPI modes. HOLD#/RESET# pin selection (P7) bit is used to select HOLD# pin or RESET# pin in SPI mode when QE="0" for the device with HOLD#/RESET#. When QE=1 or in QPI mode, P7 bit setting will be ignored since the pin becomes IO3.
For 16-pin SOIC or 24-ball TFBGA with dedicated RESET# device (Dedicated RESET# Disable bit in Functional Register is "0"), HOLD# will be selected regardless of P7 bit setting when QE="0" in SPI mode.

The SET READ PARAMETERS Operations (SRPNV: 65h, SRPV: C0h or 63h) are used to set all the Read Register bits, and can thereby define HOLD#/RESET# pin (or ball) selection, dummy cycles, and burst length with wrap around. SRPNV is used to set the non-volatile register and SRPV is used to set the volatile register.

Table 6.7 Read Register Parameter Bit Table

	P7	P6	P5	P4	P3	P2	P1	P0
	HOLD#/ RESET#	Dummy Cycles	Dummy Cycles	Dummy Cycles	Dummy Cycles	Wrap Enable	Burst Length	Burst Length
Default	0	0	0	0	0	0	0	0

Table 6.8 Read Register Bit Definition

Bit	Name	Definition	Read-/Write	Type
P0	Burst Length	Burst Length	R/W	Non-Volatile and Volatile
P1	Burst Length	Burst Length	R/W	Non-Volatile and Volatile
P2	Burst Length Set Enable	Burst Length Set Enable Bit: "0" indicates disable (default) "1" indicates enable	R/W	Non-Volatile and Volatile
P3	Dummy Cycles	Number of Dummy Cycles: Bits1 to Bit4 can be toggled to select the number of dummy cycles (1 to 15 cycles)	R/W	Non-Volatile and Volatile
P4	Dummy Cycles		R/W	Non-Volatile and Volatile
P5	Dummy Cycles		R/W	Non-Volatile and Volatile
P6	Dummy Cycles		R/W	Non-Volatile and Volatile
P7	HOLD#/ RESET#	HOLD#/RESET# function selection Bit: "0" indicates the HOLD# function is selected (default) "1" indicates the RESET# function is selected	R/W	Non-Volatile and Volatile

Table 6.9 Burst Length Data

	P1	P0
8 bytes	0	0
16 bytes	0	1
32 bytes	1	0
64 bytes	1	1

Table 6.10 Wrap Function

Wrap around boundary	**P2**
Whole array regardless of P1 and P0 value	0
Burst Length set by P1 and P0	1

Table 6.11 Read Dummy Cycles vs Max Frequency

For IS25LP256D (3.0V) (6, 7)

P[6:3]	Dummy Cycles[2,3]	Fast Read[5] 0Bh/0Ch		Fast Read Dual Output 3Bh/3Ch	Fast Read Dual IO BBh/BCh	Fast Read Quad Output 6Bh/6Ch	Fast Read Quad IO EBh/ECh	FRDTR 0Dh/0Eh	FRDDTR BDh/BEh	FRQDTR EDh/EEh
		SPI	QPI	SPI	SPI	SPI	SPI, QPI	SPI/QPI	SPI[4]	SPI, QPI
0	Default[1]	**166MHz**	**81MHz**	**166MHz**	**104MHz**	**145MHz**	**81MHz**	**80/69MHz**	**60MHz**	**69MHz**
1	1	98MHz	23MHz	75MHz	52MHz	63MHz	23MHz	50/11MHz	29MHz	11MHz
2	2	110MHz	34MHz	84MHz	80MHz	75MHz	34MHz	63/23MHz	40MHz	23MHz
3	3	122MHz	46MHz	98MHz	98MHz	87MHz	46MHz	75/34MHz	52MHz	34MHz
4	4	133MHz	58MHz	133MHz	**104MHz**	98MHz	58MHz	80/46MHz	**60MHz**	46MHz
5	5	145MHz	69MHz	140MHz	122MHz	110MHz	69MHz	80/58MHz	75MHz	58MHz
6	6	156MHz	**81MHz**	150MHz	133MHz	122MHz	**81MHz**	80/**69**MHz	80MHz	69MHz
7	7	166MHz	93MHz	166MHz	145MHz	133MHz	93MHz	80/80MHz	80MHz	80MHz
8	8	**166MHz**	104MHz	**166MHz**	156MHz	**145MHz**	104MHz	**80**/80MHz	80MHz	80MHz
9	9	166MHz	122MHz	166MHz	166MHz	156MHz	122MHz	80/80MHz	80MHz	80MHz
10	10	166MHz	127MHz	166MHz	166MHz	166MHz	127MHz	80/80MHz	80MHz	80MHz
11	11	166MHz	139MHz	166MHz	166MHz	166MHz	139MHz	80/80MHz	80MHz	80MHz
12	12	166MHz	151MHz	166MHz	166MHz	166MHz	151MHz	80/80MHz	80MHz	80MHz
13	13	166MHz	162MHz	166MHz	166MHz	166MHz	162MHz	80/80MHz	80MHz	80MHz
14	14	166MHz	166MHz	166MHz	166MHz	166MHz	166MHz	80/80MHz	80MHz	80MHz
15	15	166MHz	166MHz	166MHz	166MHz	166MHz	166MHz	80/80MHz	80MHz	80MHz

For IS25WP256D (1.8V) (8)

P[6:3]	Dummy Cycles[2,3]	Fast Read[5] 0Bh/0Ch		Fast Read Dual Output 3Bh/3Ch	Fast Read Dual IO BBh/BCh	Fast Read Quad Output 6Bh/6Ch	Fast Read Quad IO EBh/ECh	FRDTR 0Dh/0Eh	FRDDTR BDh/BEh	FRQDTR EDh/EEh
		SPI	QPI	SPI	SPI	SPI	SPI	SPI/QPI	SPI[4]	SPI
0	Default[1]	**104MHz**	**79MHz**	**104MHz**	**104MHz**	**104MHz**	**81MHz**	**80/58MHz**	**60MHz**	**69MHz**
1	1	98MHz	23MHz	75MHz	52MHz	63MHz	23MHz	50/11MHz	29MHz	11MHz
2	2	104MHz	34MHz	84MHz	80MHz	75MHz	34MHz	63/23MHz	40MHz	23MHz
3	3	104MHz	46MHz	98MHz	98MHz	87MHz	46MHz	75/34MHz	52MHz	34MHz
4	4	104MHz	58MHz	104MHz	**104MHz**	98MHz	58MHz	80/46MHz	**60MHz**	46MHz
5	5	104MHz	69MHz	104MHz	104MHz	104MHz	69MHz	80/58MHz	75MHz	58MHz
6	6	104MHz	**79MHz**	104MHz	104MHz	104MHz	**81MHz**	80/**58**MHz	80MHz	69MHz
7	7	104MHz	85MHz	104MHz	104MHz	104MHz	93MHz	80/62MHz	80MHz	80MHz
8	8	**104MHz**	85MHz	**104MHz**	104MHz	**104MHz**	104MHz	**80**/66MHz	80MHz	80MHz
9	9	104MHz	91MHz	104MHz	104MHz	104MHz	104MHz	80/69MHz	80MHz	80MHz
10	10	104MHz	94MHz	104MHz	104MHz	104MHz	104MHz	80/80MHz	80MHz	80MHz
11	11	104MHz	99MHz	104MHz	104MHz	104MHz	104MHz	80/80MHz	80MHz	80MHz
12	12	104MHz	104MHz	104MHz	104MHz	104MHz	104MHz	80/80MHz	80MHz	80MHz
13	13	104MHz	104MHz	104MHz	104MHz	104MHz	104MHz	80/80MHz	80MHz	80MHz
14	14	104MHz	104MHz	104MHz	104MHz	104MHz	104MHz	80/80MHz	80MHz	80MHz
15	15	104MHz	104MHz	104MHz	104MHz	104MHz	104MHz	80/80MHz	80MHz	80MHz

Notes:

1. Default dummy cycles are as follows.

Operation	Command		Dummy Cycles		Comment
	SPI mode	QPI mode	SPI mode	QPI mode	
Fast Read	0Bh/0Ch	0Bh/0Ch	8	6	RDUID, RDSFDP, IRRD instructions are also applied.
Fast Read Dual Output	3Bh/3Ch	-	8	-	
Fast Read Quad Output	6Bh/6Ch	-	8	-	
Fast Read Dual IO	BBh/BCh	-	4	-	
Fast Read Quad IO	EBh/ECh	EBh/ECh(9)	6	6	
Fast Read DTR	0Dh/0Eh	0Dh/0Eh	8	6	
Fast Read Dual IO DTR	BDh/BEh	-	4	-	
Fast Read Quad IO DTR	EDh/EEh	EDh/EEh(9)	6	6	

2. Enough number of dummy cycles must be applied to execute properly the AX read operation.

3. Must satisfy bus I/O contention. For instance, if the number of dummy cycles and AX bits cycles are same, then X must be Hi-Z.

4. QPI mode is not available for FRDDTR command.

5. RDUID, RDSFDP, IRRD instructions are also applied.

6. Max frequency is 166MHz at Vcc=2.7V~3.6V, Mode 0

7. Max frequency is 133MHz at Vcc=2.7V~3.6V, Mode 3 and 133MHz at Vcc=2.3V~3.6V at Mode 0

8. Max frequency is 104MHz for 1.8V device at Mode 0 and Mode 3

9. EBh/ECh command and EDh/EEh command are not supported in QPI mode for 1.8V device.

6.3.2 EXTENDED READ REGISTER

Table 6.12 and Table 6.13 define all bits that control features in SPI/QPI modes. The ODS2, ODS1, ODS0 (EB7, EB6, EB5) bits provide a method to set and control driver strength. The four bits (EB3, EB2, EB1, EB0) are read-only bits and may be checked to know what the WIP status is or whether there is an error during an Erase, Program, or Write/Set Register operation. These bits are not affected by SERPNV or SERPV commands. EB4 bit remains reserved for future use.

The SET EXTENDED READ PARAMETERS Operations (SERPNV: 85h, SERPV: 83h) are used to set all the Extended Read Register bits, and can thereby define the output driver strength used during READ modes. SRPNV is used to set the non-volatile register and SRPV is used to set the volatile register.

Table 6.12 Extended Read Register Bit Table

	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
	ODS2	ODS1	ODS0	Reserved	E_ERR	P_ERR	PROT_E	WIP
Default	1	1	1	1	0	0	0	0

Table 6.13 Extended Read Register Bit Definition

Bit	Name	Definition	Read-/Write	Type
EB0	WIP	Write In Progress Bit: Has exactly same function as the bit0 (WIP) of Status Register “0”: Ready, “1”: Busy	R	Volatile
EB1	PROT_E	Protection Error Bit: "0" indicates no error "1" indicates protection error in an Erase or a Program operation	R	Volatile
EB2	P_ERR	Program Error Bit: "0" indicates no error "1" indicates an Erase operation failure or protection error	R	Volatile
EB3	E_ERR	Erase Error Bit: "0" indicates no error "1" indicates a Program operation failure or protection error	R	Volatile
EB4	Reserved	Reserved	R	Reserved
EB5	ODS0	Output Driver Strength: Output Drive Strength can be selected according to Table 6.14	R/W	Non-Volatile and Volatile
EB6	ODS1		R/W	Non-Volatile and Volatile
EB7	ODS2		R/W	Non-Volatile and Volatile

Table 6.14 Driver Strength Table

ODS2	ODS1	ODS0	Description	Remark
0	0	0	Reserved	
0	0	1	12.50%	
0	1	0	25%	
0	1	1	37.50%	
1	0	0	Reserved	
1	0	1	75%	
1	1	0	100%	
1	1	1	50%	Default

WIP bit: The definition of the WIP bit is exactly same as the one of Status Register.

PROT_E bit: The Protection Error bit indicates whether an Erase or Program operation has attempted to modify a protected array sector or block, or to access a locked Information Row region. When the bit is set to "1" it indicates that there was an error or errors in previous Erase or Program operations. See Table 6.15 for details.

P_ERR bit: The Program Error bit indicates whether a Program operation has succeeded or failed, or whether a Program operation has attempted to program a protected array sector/block or a locked Information Row region. When the bit is set to "1" it indicates that there was an error or errors in previous Program or Write/Set Non-Volatile Register operations. See Table 6.15 for details.

E_ERR bit: The Erase Error bit indicates whether an Erase operation has succeeded or failed, or whether an Erase operation has attempted to erase a protected array sector/block or a locked Information Row region. When the bit is set to "1" it indicates that there was an error or errors in previous Erase or Write/Set Non-Volatile Register operations. See Table 6.15 for details.

Table 6.15 Instructions to set PROT_E, P_ERR, or E_ERR bit

Instructions	Description
PP/4PP/PPQ/4PPQ/PGPPB/ 4PGPPB/PGPWD	The commands will set the P_ERR if there is a failure in the operation. Attempting to program within the protected array sector/block or within an erase suspended sector/block will result in a programming error with P_ERR and PROT_E set to "1".
IRP	The command will set the P_ERR if there is a failure in the operation. In attempting to program within a locked Information Row region, the operation will fail with P_ERR and PROT_E set to 1.
PGASP	The command will set the P_ERR if there is a failure in the operation. Attempting to program ASPR[2:1] after the Protection Mode is selected or attempting to program ASPR[2:1] = 00b will result in a programming error with P_ERR and PROT_E set to "1".
UNPWD	If the UNPWD command supplied password does not match the hidden internal password, the UNPWD operation fails in the same manner as a programming operation on a protected sector/block and sets the P_ERR and PROT_E to "1".
WRSR/WRABR/SRPNV/ SERPNV/WRBRNV	The update process for the non-volatile register bits involves an erase and a program operation on the non-volatile register bits. If either the erase or program portion of the update fails, the related error bit (P_ERR or E_ERR) will be set to "1". Only for WRSR command, when Status Register is write-protected by SRWD bit and WP# pin, attempting to write the register will set PROT_E and E_ERR to "1".
WRFR	The commands will set the P_ERR if there is a failure in the operation.
SER/4SER/BER32K/ 4BER32K/BER64K/ 4BER64K/CER/IRER/ERPPB	The commands will set the E_ERR if there is a failure in the operation. E_ERR and PROT_E will be set to "1" when the user attempts to erase a protected main memory sector/block or a locked Information Row region. Chip Erase (CER) command will set E_ERR and PROT_E if any blocks are protected by Block Protection bits (BP3~BP0). But Chip Erase (CER) command will not set E_ERR and PROT_E if sectors/blocks are protected by ASP (DYB bits or PPB bits) only.

Notes:
1. **OTP bits in the Function Register and TBPARM (OTP bit) in the ASP Register may only be programmed to "1". Writing of the bits back to "0" is ignored and no error is set.**
2. **Read only bits and partially protected bits by FREEZE bit in registers are never modified by a command so that the corresponding bits in the Write/Set Register command data byte are ignored without setting any error indication.**
3. **Once the PROT_E, P_ERR, and E_ERR error bits are set to "1", they remains set to "1" until they are cleared to "0" with a Clear Extended Read Register (CLERP) command. This means that those error bits must be cleared through the CLERP command. Alternatively, Hardware Reset, or Software Reset may be used to clear the bits.**
4. **Any further command will be executed even though the error bits are set to "1".**

6.4 AUTOBOOT REGISTER

AutoBoot Register Bit (32 bits) definitions are described in Table 6.16.

Table 6.16 AutoBoot Register Parameter Bit Table

Bits	Symbols	Function	Type	Default Value	Description
AB[31:5]	ABSA	AutoBoot Start Address	Non-Volatile	0000000h	32 byte boundary address for the start of boot code access
AB[4:1]	ABSD	AutoBoot Start Delay	Non-Volatile	0h	Number of initial delay cycles between CE# going low and the first bit of boot code being transferred, and it is the same as dummy cycles of FRD (QE=0) or FRQIO (QE=1). Example: The number of initial delay cycles is 8 (QE=0) or 6 (QE=1) when AB[4:1]=0h (Default setting).
AB0	ABE	AutoBoot Enable	Non-Volatile	0	1 = AutoBoot is enabled 0 = AutoBoot is not enabled

6.5 BANK ADDRESS REGISTER

Related Commands: Read Volatile Bank Address Register (RDBR 16h/C8h), Write Volatile Bank Address Register (WRBRV 17h/C5h), Write Non-Volatile Bank Address Register (WRBRNV 18h), Enter 4-byte Address Mode (EN4B B7h), and Exit 4-byte Address Mode (EX4B 29h).

Bank Address Register Bit (8 bits) definitions are described in Table 6.17 and Table 6.18.

Table 6.17 Bank Address Register Bit Table

	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	EXTADD	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	BA24
Default	0	0	0	0	0	0	0	0

Table 6.18 Bank Address Register Bit Definition

Bit	Name	Definition	Read-/Write	Type
BA0	BA24	Enables 128Mb segment selection in 3-byte addressing "0" indicates lower 128Mb segment is selected. "1" indicates upper 128Mb segment is selected.	R/W	Non-Volatile and Volatile
BA1	Reserved	Reserved	R	Reserved
BA2	Reserved	Reserved	R	Reserved
BA3	Reserved	Reserved	R	Reserved
BA4	Reserved	Reserved	R	Reserved
BA5	Reserved	Reserved	R	Reserved
BA6	Reserved	Reserved	R	Reserved
BA7	EXTADD	3-byte or 4-byte addressing selection Bit: "0" indicates 3-byte addressing. "1" indicates 4-byte addressing.	R/W	Non-Volatile and Volatile

BA24: The Bank Address Register supplies additional high order bits of the main flash array byte boundary address for legacy commands that supply only the low order 24 bits of address. The Bank Address is used as the high bits of address (above A23) for all 3-byte address commands when EXTADD=0. The Bank Address is not used when EXTADD = 1 and traditional 3-byte address commands are instead required to provide all four bytes of address.

EXTADD: Extended Address (EXTADD) controls the address field size for legacy SPI commands. When shipped from factory, it is cleared to “0” for 3 bytes (24 bits) of address. When set to “1”, the legacy commands will require 4 bytes (32 bits) for the address field.

6.6 ADVANCED SECTOR/BLOCK PROTECTION (ASP) RELATED REGISTER

6.6.1 ADVANCED SECTOR/BLOCK PROTECTION REGISTER (ASPR)

Related Commands: Read ASP (RDASP 2Bh) and Program ASP (PGASP 2Fh).

Advanced Sector/Block Protection (ASP) Register Bit (16 bits) definitions are described in Tables 6.19 and 6.20.

Table 6.19 Advanced Sector/Block Protection Register (ASPR) Bit Table

	15	7 to 14	6	5	4	3	2	1	0
	TBPARM	Reserved	Reserved	Reserved	Reserved	Reserved	PWDMLB	PSTMLB	Reserved
Default	1	1	1	1	1	1	1	1	1

Table 6.20 Advanced Sector/Block Protection Register (ASPR) Bit Definition

Bit	Name	Definition	Read-/Write	Type
0	Reserved	Reserved	R	Reserved
1	PSTMLB	Persistent Protection Mode Lock Bit “0” = Persistent Protection Mode permanently enabled. “1” = Persistent Protection Mode not permanently enabled.	R/W	OTP
2	PWDMLB	Password Protection Mode Lock Bit “0” = Password Protection Mode permanently enabled. “1” = Password Protection Mode not permanently enabled.	R/W	OTP
3:14	Reserved	Reserved	R	Reserved
15	TBPARM	Configures Parameter Sectors location “0” = 4KB physical sectors at top, (high address) “1” = 4KB physical sectors at bottom (Low address)	R/W	OTP

The Advanced Sector/Block Protection Register (ASPR) is used to permanently configure the behavior of Advanced Sector/Block Protection (ASP) features and parameter sectors location.

PWDMLB (ASPR[2]) and PSTMLB (ASPR[1]) bits: When shipped from the factory, all devices default ASP to the Persistent Protection Mode, with all sectors unprotected, when power is applied. The device programmer or host system must then choose which sector/block protection method to use. Programming either of the Protection Mode Lock Bits locks the part permanently in the selected mode:

- ASPR[2:1] = 11 = No ASP mode selected, Persistent Protection Mode is the default.
- ASPR[2:1] = 10 = Persistent Protection Mode permanently selected.
- ASPR[2:1] = 01 = Password Protection Mode permanently selected.
- ASPR[2:1] = 00 = Illegal condition, attempting to program both bits to zero results in a programming failure and the program operation will abort. It will result in a programming error with P_ERR set to 1.

As a result, PWDMLB and PSTMLB are mutually exclusive, only one may be programmed to zero.

ASPR programming rules:

- If the Password Protection Mode is chosen, the password must be programmed prior to setting the corresponding bit.
- Once the Protection Mode is selected, the ASPR[2:1] bits are permanently protected from programming and no further change to the ASPR[2:1] is allowed. Attempting to program ASPR[2:1] after selected will result in a programming error with P_ERR set to 1. The programming time of the ASPR is the same as the typical page programming time. The system can determine the status of the ASPR programming operation by reading the WIP bit in the Status Register or Extended Read Register.
- TBPARM bit can be programmed even after ASPR[2:1] bits are programmed while the FREEZE bit in the PPB Lock Register is “0”.

TBPARM bit: TBPARM defines the logical location of the parameter block. The parameter block consists of thirty-two 4KB sectors, which replace two 64KB blocks. When TBPARM is set to a "0" the parameter block is in the top of the memory array address space. When TBPARM is set to a "1" the parameter block is at the Bottom of the array. TBPARM is OTP and set to a "1" when shipped from factory. If TBPARM is programmed to "0", an attempt to change it back to "1" will fail and ignore the Program.

The desired state of TBPARM must be selected during the initial configuration of the device during system manufacture; before the first program or erase operation on the main flash array. TBPARM must not be programmed after programming or erasing is done in the main flash array.

TBS can be programmed independent of TBPARM. Therefore, the user can elect to store parameter information from the bottom of the array and protect boot code starting at the top of the array, and vice versa. Or the user can select to store and protect the parameter information starting from the top or bottom together.

6.6.2 PASSWORD REGISTER

Related Commands: Read Password (RDPWD E7h), Program Password (PGPWD E8h), and Unlock Password (UNPWD, E9h).

Table 6.21 Password Register Bit Definition

Bit	Name	Definition	Default	Read-/Write	Type
0:63	PSWD	64 bit hidden password: The password is no longer readable after the password protection mode is selected by programming ASPR bit 2 to zero.	FFFFFFFF-FFFFFFFFh	R/W	OTP

6.6.3 PPB LOCK REGISTER

Related Commands: Read PPB Lock Bit (RDPLB A7h), Write PPB Lock Bit (WRPLB A6h), and Set FREEZE Bit (SFRZ 91h).

Table 6.22 PPB Lock Register Bit Definition

Bit	Name	Definition	Default	Read-/Write	Type
0	PPBLK	PPB Lock bit: Protect PPB Array "0" = PPB array protected until next power cycle or Hardware Reset "1" = PPB array may be programmed or erased.	Persistent: 1 Password: 0	R/W	Volatile
1:6	Reserved	Reserved	Reserved	R	Reserved
7	FREEZE	Lock current state of BP3-0 bits in Status Register, TBS in Function Register and TBPARM in ASPR, and Information Row (IR) regions. "1" = Locked "0" = Un-locked	0	R/W	Volatile

PPBLK bit: The PPB Lock bit is a volatile bit for protecting all PPB bits. When cleared to 0, it locks all PPBs, when set to 1, it allows the PPBs to be changed. The WRPLB command is used to clear the PPB Lock bit to 0. The PPB Lock bit must be cleared to 0 only after all the PPBs are configured to the desired settings.

In Persistent Protection mode, the PPB Lock bit is set to 1 during POR or Hardware Reset. When cleared to 0, no software command sequence can set the PPB Lock bit to 1, only another Hardware Reset or power-up can set the PPB Lock bit.

In the Password Protection mode, the PPB Lock bit is cleared to 0 during POR or Hardware Reset. The PPB Lock bit can only be set to 1 by the Unlock Password command.

FREEZE bit: FREEZE bit, when set to "1", locks the current state of BP3-0 in Status Register, TBS in the Function Register, TBPARM in the Advanced Sector/Block Protection Register, and the Information Row. This prevents writing, programming, or erasing these areas. As long as FREEZE remains cleared to logic "0", BP3-0 in Status Register, TBS in the Function Register, and TBPARM in the Advanced Sector/Block Protection Register are writable and the Information Row is programmable. Once FREEZE has been written to a logic "1" it can only be cleared to a logic "0" by a power-on cycle or a Hardware Reset. Software Reset will not affect the state of FREEZE. The FREEZE is volatile and the default state of FREEZE after power-on is "0". The FREEZE can be set to "1" by a SFRZ command.

6.6.4 PPB REGISTER

Related Commands: Read PPB (RDPPB FCh or 4RDPPB E2h)), Program PPB (PGPPB FDh or 4PGPPB E3h), and Erase PPB (ERPPB E4h).

Table 6.23 PPB Register Bit Definition

Bit	Name	Definition	Default	Read-/Write	Type
0:7	PPB	Read or Program per sector/block PPB: 00h = PPB for the sector/block addressed by the RDPPB or PGPPB command is programmed to "0", protecting that sector/block from program or erase operations. FFh = PPB for the sector/block addressed by the RDPPB or PGPPB command is erased to "1", not protecting that sector/block from program or erase operations.	FFh	R/W	Non-Volatile

6.6.5 DYB REGISTER

Related Commands: Read DYB (RDDYB FAh or 4RDDYB E0h) and Write DYB (WRDYB FBh or 4WRDYB E1h).

Table 6.24 DYB Register Bit Definition

Bit	Name	Definition	Default	Read-/Write	Type
0:7	DYB	Read or Write per sector/block DYB: 00h = DYB for the sector/block addressed by the RDDYB or WRDYB command is cleared to "0", protecting that sector/block from program or erase operations. FFh = DYB for the sector/block addressed by the RDDYB or WRDYB command is set to "1", not protecting that sector/block from program or erase operations.	FFh	R/W	Volatile

7. PROTECTION MODE

The device supports hardware and software write-protection mechanisms.

7.1 HARDWARE WRITE PROTECTION

The Write Protection (WP#) pin provides a hardware write protection method for BP3, BP2, BP1, BP0, SRWD, and QE in the Status Register. Refer to the section 6.1 STATUS REGISTER.

Write inhibit voltage (V_{WI}) is specified in the section 9.7 POWER-UP AND POWER-DOWN. All write sequence will be ignored when Vcc drops to V_{WI}.

Table 7.1 Hardware Write Protection on Status Register

SRWD	WP#	Status Register
0	Low	Writable
1	Low	Protected
0	High	Writable
1	High	Writable

Note: Before the execution of any program, erase or Write Status/Function Register instruction, the Write Enable Latch (WEL) bit must be enabled by executing a Write Enable (WREN) instruction. If the WEL bit is not enabled, the program, erase or write register instruction will be ignored.

7.2 SOFTWARE WRITE PROTECTION

The device also provides two kinds of software write protection feature. One is Block Protection by Block Protection bits (BP3, BP2, BP1, BP0) and another is Advanced Sector/Block Protection (ASP). When Block Protection is enabled (i.e., any BP3-0 are set to “1”), Advanced Sector/Block Protection (ASP) can still be used to protect sectors/blocks not protected by the Block Protection scheme. In the case that both ASP and Block Protection are used on the same sector/block the logical OR of ASP and Block Protection related to the sector/block is used.

Warning: ASP and Block Protection should not be used concurrently. Use one or the other, but not both.

7.2.1 BLOCK PROTECTION BITS

The device provides a software write protection feature. The Block Protection bits (BP3, BP2, BP1, BP0) allow part or the whole memory area to be write-protected. For details, see 6.1 Status Register.

7.2.2 ADVANCED SECTOR/BLOCK PROTECTION (ASP)

There are two ways to implement software Advanced Sector/Block Protection on this device: Password Protection method or Persistent Protection methods. Through these two protection methods, user can disable or enable the programming or erasing operation to any or all blocks including 32 top 4K sectors or 32 bottom 4K sectors. The Figure 7.1 shows an overview of these methods.

Every main flash array block/top sector/bottom sector has a non-volatile (PPB) and a volatile (DYB) protection bit associated with it. When either bit is 0, the sector is protected from program and erase operations.

The PPB bits are protected from program and erase when the PPB Lock bit is “0”. The PPB bits are erased so that all main flash array sectors are unprotected when shipped from factory.

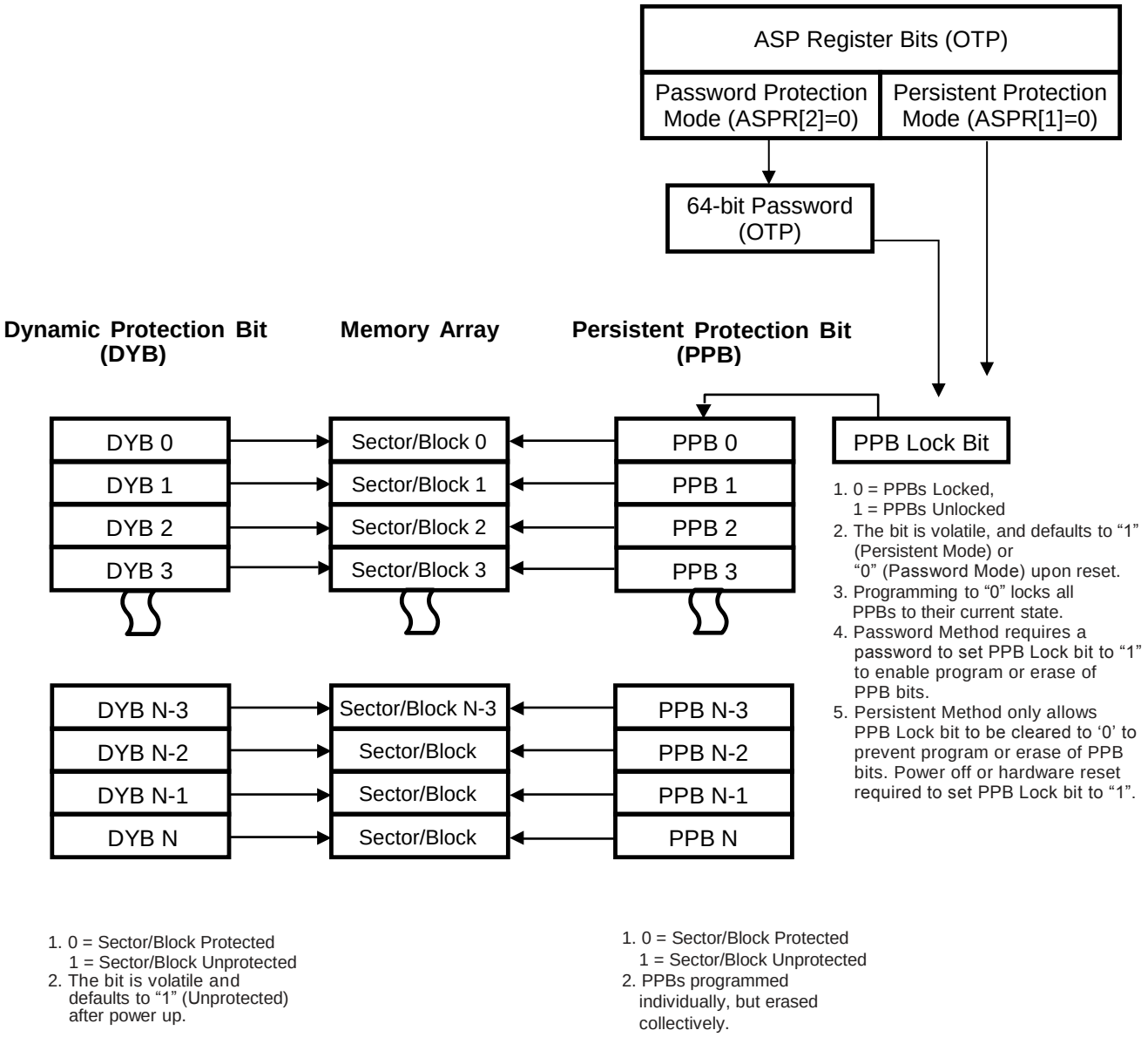
There are two methods for managing the state of the PPB Lock bit, Persistent Protection and Password Protection.

The Persistent Protection Mode sets the PPB Lock bit to “1” during power up or Hardware Reset so that the PPB bits are unprotected. There is a WRPLB command to clear the PPB Lock bit to “0” to protect the PPB bits. There is no command in the Persistent Protection method to set the PPB Lock bit therefore the PPB Lock bit will remain at “0” until the next power up or Hardware Reset. The Persistent Protection method allows boot code the option of changing sector protection by programming or erasing the PPB, then protecting the PPB from further change for the remainder of normal system operation by clearing the PPB Lock bit. This is sometimes called Boot-code controlled sector protection.

The Password Protection Mode requires use of a password to control PPB protection. In the Password Protection Mode, the PPB Lock bit is cleared to “0” during power up or Hardware Reset to protect the PPB bits. A 64-bit password may be permanently programmed and hidden for the Password Protection Mode. The UNPWD command can be used to provide a password for comparison with the hidden password. If the password matches the PPB Lock bit is set to “1” to unprotect the PPB. The WRPLB command can be used to clear the PPB Lock bit to “0”. After clearing the PPB Lock bit to “0”, the UNPWD command can be used again to unprotect the PPB.

The selection of the PPB Lock bit management method is made by programming OTP bits in the ASP Register so as to permanently select the method used.

Figure 7.1 Advanced Sector/Block Protection



Note:
1. 256M: N = 542 = 32 (32 Top 4K sectors or 32 Bottom 4K sectors) + 510 (510 64K blocks)

Table 7.2 PPB/DYB and Sector/Block mapping (TBPARM = 1)

Memory Density	PPB Group	DYB Group	Block No. (64Kbyte)	Sector No.	Sector Size (Kbyte)	Address Range
256Mb	PPB 0	DYB 0	Block 0	Sector 0	4	000000h - 000FFFh
	:	:		:	:	:
	:	:		:	:	:
	PPB 15	DYB 15		Sector 15	4	00F000h - 00FFFFh
	PPB 16	DYB 16	Block 1	Sector 16	4	010000h - 010FFFh
	:	:		:	:	:
	:	:		:	:	:
	PPB 31	DYB 31		Sector 31	4	01F000h - 01FFFFh
	PPB 32	DYB 32	Block 2	Sector 32	4	020000h - 020FFFh
				:	:	:
				:	:	:
				Sector 47	4	02F000h - 02FFFFh
	:	:	:	:	:	:
	PPB 284	DYB 284	Block 254	Sector 4064	4	FE0000h – FE0FFFh
				:	:	:
				:	:	:
				Sector 4079	4	FEF000h – FEFFFFh
	PPB 285	DYB 285	Block 255	Sector 4080	4	FF0000h – FF0FFFh
				:	:	:
				:	:	:
				Sector 4095	4	FFF000h – FFFFFFh
	:	:	:	:	:	:
	PPB 540	DYB 540	Block 510	Sector 8160	4	1FE0000h – 1FE0FFFh
				:	:	:
				:	:	:
				Sector 8175	4	1FEF000h – 1FEFFFFh
	PPB 541	DYB 541	Block 511	Sector 8176	4	1FF0000h – 1FF0FFFh
				:	:	:
				:	:	:
				Sector 8191	4	1FFF000h – 1FFFFFFh

Table 7.3 PPB/DYB and Sector/Block mapping (TBPARM = 0)

Memory Density	PPB Group	DYB Group	Block No. (64Kbyte)	Sector No.	Sector Size (Kbyte)	Address Range
256Mb	PPB 0	DYB 0	Block 0	Sector 0	4	000000h - 000FFFh
				:	:	:
				:	:	:
				Sector 15	4	00F000h - 00FFFFh
	PPB 1	DYB 1	Block 1	Sector 16	4	010000h - 010FFFh
				:	:	:
				:	:	:
				Sector 31	4	01F000h - 01FFFFh
	PPB 2	DYB 2	Block 2	Sector 32	4	020000h - 020FFFh
				:	:	:
				:	:	:
				Sector 47	4	02F000h - 02FFFFh
	:	:	:	:	:	:
	PPB 254	DYB 254	Block 254	Sector 4064	4	FE0000h – FE0FFFh
				:	:	:
				:	:	:
				Sector 4079	4	FEF000h – FEFFFFh
	PPB 255	DYB 255	Block 255	Sector 4080	4	FF0000h – FF0FFFh
				:	:	:
				:	:	:
				Sector 4095	4	FFF000h – FFFFFFh
	:	:	:	:	:	:
	PPB 510	DYB 510	Block 510	Sector 8160	4	1FE0000h – 1FE0FFFh
	:	:		:	:	:
	:	:		:	:	:
	PPB 525	DYB 525		Sector 8175	4	1FEF000h – 1FEFFFFh
	PPB 525	DYB 526	Block 511	Sector 8176	4	1FF0000h – 1FF0FFFh
	:	:		:	:	:
	:	:		:	:	:
	PPB 541	DYB 541		Sector 8191	4	1FFF000h – 1FFFFFFh

Persistent Protection Bits (PPBs)

The Persistent Protection Bits (PPBs) are unique for each sector/block and non-volatile (refer to Figure 7.1, Table 7.2, and Table 7.3). It is programmed individually but must be erased as a group, similar to the way individual words may be programmed in the main array but an entire sector/block must be erased at the same time. The PPBs have the same endurances as the Flash memory. Preprogramming and verification prior to erasure are handled by the device, and therefore do not require system monitoring. Programming a PPB bit requires the typical page programming time. Erasing all the PPBs requires typical sector erase time. During PPB bit programming and PPB bit erasing, status is available by reading the Status Register or Extended Read Register. Reading of a PPB bit requires the initial access time of the device.

Notes:
1. Each PPB is individually programmed to “0” and all are erased to “1” in parallel.
2. The PPB Lock bit must be cleared first before changing the status of a PPB.
3. While programming PPB, array data cannot be read from any sectors/blocks.
4. When reading the PPB of the desired sector/block the address should be location zero within the sector/block. The high order address bits not used must be zero.
5. There are no means for individually erasing a specific PPB and no specific sector/block address is required for this operation.
6. The state of the PPB for a given sector/block can be verified by using a PPB Read command.
7. When the parts are first shipped, the PPBs are cleared (erased to “1”).

Dynamic Protection Bits (DYBs)

Dynamic Protection Bits (DYBs) are volatile and unique for each sector/block and can be individually modified. DYBs only control the protection for unprotected sectors/blocks that have their PPBs cleared (erased to “1”). By issuing the Write DYB command, the DYBs are cleared to “0” or set to “1”, thus placing each sector/block in the protected or unprotected state respectively. This feature allows software to easily protect sectors/blocks against inadvertent changes, yet does not prevent the easy removal of protection when changes are needed. The DYBs can be set or cleared as often as needed as they are volatile bits.

Persistent Protection Bit (PPB) Lock Bit

The PPB Lock bit is a volatile bit for protecting all PPB bits. When cleared to “0”, it locks all PPBs and when set to “1”, it allows the PPBs to be changed. . If the PPB Lock bit is “0”, the PPB Program or Erase command does not execute and fails without programming or erasing the PPB.

In Persistent Protection mode, the PPB Lock bit is set to “1” during power up or Hardware Reset. When cleared to “0”, no software command sequence can set the PPB Lock bit to “1”, only another Hardware Reset or power-up can set the PPB Lock bit.

In the Password Protection mode, the PPB Lock bit is cleared to “0” during power up or a Hardware Reset during power up or a Hardware Reset during power up or a Hardware Reset. The PPB Lock bit can only be set to “1” by the Password Unlock command.

The PPB Lock bit must be cleared to “0” only after all PPBs are configured to the desired settings.

Sector/Block Protection States Summary

Each sector in specific blocks and each of all other blocks except for the specific blocks can be in one of the following protection states:

- Unlocked – The sector/block is unprotected and protection can be changed by a simple command. The protection state defaults to unprotected after a power cycle, software reset, or hardware reset.
- Dynamically Locked – A sector/block is protected and protection can be changed by a simple command. The protection state is not saved across a power cycle or reset.
- Persistently Locked – A sector/block is protected and protection can only be changed if the PPB Lock bit is set to "1". The protection state is non-volatile and saved across a power cycle or reset. Changing the protection state requires programming and or erase of the PPB bits**.**

Table 7.4 contains all possible combinations of the DYB, PPB, and PPB Lock bit relating to the status of the sector/block. In summary, if the PPB Lock bit is locked (cleared to "0"), no changes to the PPBs are allowed. The PPB Lock bit can only be unlocked (set to "1") through a Hardware Reset or power cycle.

Table 7.4 Sector/Block Protection States

		Protection Bit values		Assigned Sector/Block State
		PPB	**DYB**	
"0" = Locked or Protected "1" = Unlocked or Unprotected		1	1	Unprotected
		1	0	Protected
		0	1	Protected
		0	0	Protected
PPB Lock Bit	1	Changeable	Changeable	
	0	NOT changeable	Changeable	

Persistent Protection Mode

The Persistent Protection Mode sets the PPB Lock bit to "1" during power up or Hardware Reset so that the PPB bits are unprotected by a device Hardware Reset. Software Reset does not affect the PPB Lock bit. The WRPLB command can clear the PPB Lock bit to "0" to protect the PPB. There is no command to set the PPB Lock bit therefore the PPB Lock bit will remain at "0" until the next power up or Hardware Reset.

Password Protection Mode

The Password Protection Mode allows an even higher level of security than the Persistent Protection Mode by requiring a 64-bit password for unlocking the device PPB Lock bit. In addition to this password requirement, after power up or Hardware Reset, the PPB Lock bit is cleared to "0" to maintain the password mode of operation. Successful execution of the Unlock Password command by entering the entire password sets the PPB Lock bit to "1", allowing for sector/block PPBs modifications.

Notes:

1. The password is all "1"s when shipped from Factory. It is located in its own memory space and is accessible through the use of the Program Password and Read Password commands.
2. Once the Password is programmed and verified, the Password Protection Mode Lock Bit (ASPR[2]=0) in ASP Register must be programmed in order to prevent reading or modifying the password. After the Password Protection Mode Lock Bit is programmed, all further Program and Read commands to the password region are disabled and these commands are ignored so that there is no means to verify what the password is. Password verification is only allowed before selecting the Password Protection Mode.
3. The Program Password Command is only capable of programming "0"s. Programming a "1" after a cell is programmed as a "0" results in the cell left as a "0" with no programming error.
4. All 64-bit password combinations are valid as a password.
5. The Protection Mode Lock Bits in ASP Register are not erasable because they are OTP.
6. The exact password must be entered in order for the unlocking function to occur. If the password provided by Unlock Password command does not match the hidden internal password, the unlock operation fails in the same manner as a programming operation on a protected sector/block. The P_ERR and PROT_E are set to 1 and the PPB Lock bit remains cleared to 0. In this case it is a failure to change the state of the PPB Lock bit because it is still protected by the lack of a valid password.
7. The Unlock Password command cannot be accepted any faster than once every 100µs ± 20µs. This makes it take an unreasonably long time (58 million years) for a hacker to run through all the 64-bit combinations in an attempt to correctly match a password. The Read Status Register command or the Read Extended Read Register may be used to read the WIP bit to determine when the device has completed the Unlock Password command or is ready to accept a new password command. When a valid password is provided the Unlock Password command does not insert the 100µs delay before returning the WIP bit to zero.
8. If the password is lost after selecting the Password Protection Mode, there is no way to set the PPB Lock bit.

8. DEVICE OPERATION

8.1 COMMAND OVERVIEW

The device utilizes an 8-bit instruction register. Refer to Table 8.4. Instruction Set for details on instructions and instruction codes. All instructions, addresses, and data are shifted in with the most significant bit (MSB) first on Serial Data Input (SI) or Serial Data IOs (IO0, IO1, IO2, IO3). The input data on SI or IOs is latched on the rising edge of Serial Clock (SCK) for normal mode and both of rising and falling edges for DTR mode after Chip Enable (CE#) is driven low (V_{IL}). Every instruction sequence starts with a one-byte instruction code and is followed by address bytes and/or dummy cycles (configurable) and/or data bytes, depending on the type of instruction. CE# must be driven high (V_{IH}) after the last bit of the instruction sequence has been shifted in to end the operation.

Commands are structured as follows:

- Each command begins with a byte (eight bits) instruction.
- The instruction may be stand alone or may be followed by address bits to select a location within one of several address spaces in the device. The address may be either a 24-bit or 32-bit byte boundary address.
- The SPI interface with Multiple IO provides the option for each transfer of address and data information to be done one, two, or four bits in parallel. This enables a tradeoff between the number of signal connections (IO bus width) and the speed of information transfer. If the host system can support a two or four bit wide IO bus the memory performance can be increased by using the instructions that provide parallel two bit (dual) or parallel four bit (quad) transfers.
- The width of all transfers following the instruction are determined by the instruction sent.
- All single bit or parallel bit groups are transferred in most to least significant bit order.
- Some instructions send Mode Bits following the address to indicate that the next command will be of the same type with an implied, rather than an explicit, instruction. The next command thus does not provide an instruction byte, only a new address and mode bits. This reduces the time needed to send each command when the same command type is repeated in a sequence of commands.
- The address or Mode Bits may be followed by Dummy Cycles before read data is returned to the host.
- Dummy Cycles may be zero to several SCK cycles. In fact, Mode Bits will be counted as a part of Dummy Cycles.
- All instruction, address, Mode, and data information is transferred in byte granularity. Addresses are shifted into the device with the Most Significant Byte first. All data is transferred with the lowest address byte sent first. Following bytes of data are sent in lowest to highest byte address order i.e. the byte address increments.
- All attempts to read the flash memory array during a program, erase, or a write cycle (embedded operations) are ignored. The embedded operation will continue to execute without any affect. A very limited set of commands are accepted during an embedded operation. These are discussed in the individual command descriptions. While a program, erase, or write operation is in progress, it is recommended to check that the Write In Progress (WIP) bit is “0” before issuing most commands to the device, to ensure the new command can be accepted.
- Depending on the command, the time for execution varies. A command to read status information from an executing command is available to determine when the command completes execution and whether the command was successful.
- Following are some general signal relationship descriptions to keep in mind.
  - The host always controls the Chip Enable (CE#), Serial Clock (SCK), and Serial Input (SI) - SI for single bit wide transfers. The memory drives Serial Output (SO) for single bit read transfers. The host and memory alternately drive the IO0-IO3 signals during Dual and Quad transfers.
  - All commands begin with the host selecting the memory by driving CE# low before the first rising edge of SCK. CE# is kept low throughout a command and when CE# is returned high the command ends.

  Generally, CE# remains low for 8-bit transfer multiples to transfer byte granularity information. All commands will not be accepted if CE# is returned high not at an 8-bit boundary.

8.2 COMMAND SET SUMMARY

Extended Addressing

To accommodate addressing above 128Mb (24-bit), there are three options:

1. New instructions with 4-byte (32-bit) address. See Table 8.2.
2. 4-byte addressing with the 3-byte address instructions:
   For backward compatibility to the 3-byte address instructions, the standard instructions can be used in conjunction with the EXTADD Bit in the Bank Address Register (BAR[7]) or Enter 4-byte Address Mode to switch from 3 bytes to 4 bytes of address field. When EXTADD bit is set to 1 or Enter 4-byte Address Mode command is issued only in the case that EXTADD bit = 0, the instructions are changed to require 4-byte (32-bit) for the address field. See Table 8.3.
3. 3-byte addressing with the 3-byte address instructions:
   For backward compatibility to the 3-byte addressing, the standard instructions can be used in conjunction with the Bank Address Register. See Table 8.3.
   - The Bank Address Register is used to switch between 128Mbit (16Mbyte) banks of memory, the standard 3-byte address selects an address within the bank selected by the Bank Address Register.
     - The host system writes the Bank Address Register to access beyond the first 128Mbit of memory.
     - This applies to read, erase, and program commands.
   - The Bank Address Register provides the high order (4th) byte of address, which is used to address the available memory at addresses greater than 128Mbit.
   - Bank Address Register bits are volatile.
     - On power up, the default is Bank0 (the lowest address 16 Mbytes).
   - For Read, the device will continuously transfer out data until the end of the array.
     - There is no bank to bank delay.
     - The Bank Address Register is not updated.
     - The Bank Address Register value is used only for the initial address of an access.

Table 8.1 Bank Address Map

Bank Address Register Bit 0	Bank	Memory Array Address Range	
0	0	00000000h	00FFFFFFh
1	1	01000000h	01FFFFFFh

Table 8.2 New Instruction Set with 4-byte address

Instruction Name	Operation	Code	Address Mode
4NORD	4-byte Address Normal Read Mode	13h	4-byte Address
4FRD	4-byte Address Fast Read Mode	0Ch	4-byte Address
4FRDIO	4-byte Address Fast Read Dual I/O	BCh	4-byte Address
4FRDO	4-byte Address Fast Read Dual Output	3Ch	4-byte Address
4FRQIO	4-byte Address Fast Read Quad I/O	ECh	4-byte Address
4FRQO	4-byte Address Fast Read Quad Output	6Ch	4-byte Address
4FRDTR	4-byte Address Fast Read DTR Mode	0Eh	4-byte Address
4FRDDTR	4-byte Address Fast Read Dual I/O DTR	BEh	4-byte Address
4FRQDTR	4-byte Address Fast Read Quad I/O DTR	EEh	4-byte Address
4PP	4-byte Address Serial Input Page Program	12h	4-byte Address
4PPQ	4-byte Address Quad Input Page Program	34h/3Eh	4-byte Address
4SER	4-byte Address Sector Erase	21h	4-byte Address
4BER32 (32KB)	4-byte Address Block Erase 32KB	5Ch	4-byte Address
4BER64 (64KB)	4-byte Address Block Erase 64KB	DCh	4-byte Address
4SECUNLOCK	4-byte Address Sector Unlock	25h	4-byte Address
4RDDYB	4-byte Address Read DYB	E0	4-byte Address
4WRDYB	4-byte Address Write DYB	E1	4-byte Address
4RDPPB	4-byte Address Read PPB	E2	4-byte Address
4PGPPB	4-byte Address Program PPB	E3	4-byte Address

Table 8.3 Instruction Set with 3-byte or 4-byte address according to EXTADD Bit setting

Instruction Name	Operation	Code	Address Mode	
			EXTADD (BAR[7] = 1	EXTADD (BAR[7]) = 0
NORD	Normal Read Mode	03h	4-byte Address	3-byte Address
FRD	Fast Read Mode	0Bh	4-byte Address	3-byte Address
FRDIO	Fast Read Dual I/O	BBh	4-byte Address	3-byte Address
FRDO	Fast Read Dual Output	3Bh	4-byte Address	3-byte Address
FRQIO	Fast Read Quad I/O	EBh	4-byte Address	3-byte Address
FRQO	Fast Read Quad Output	6Bh	4-byte Address	3-byte Address
FRDTR	Fast Read DTR Mode	0Dh	4-byte Address	3-byte Address
FRDDTR	Fast Read Dual I/O DTR	BDh	4-byte Address	3-byte Address
FRQDTR	Fast Read Quad I/O DTR	EDh	4-byte Address	3-byte Address
PP	Serial Input Page Program	02h	4-byte Address	3-byte Address
PPQ	Quad Input Page Program	32h/38h	4-byte Address	3-byte Address
SER	Sector Erase	D7h/20h	4-byte Address	3-byte Address
BER32 (32KB)	Block Erase 32KB	52h	4-byte Address	3-byte Address
BER64 (64KB)	Block Erase 64KB	D8h	4-byte Address	3-byte Address
SECUNLOCK	Sector Unlock	26h	4-byte Address	3-byte Address
RDDYB	Read DYB	FA	4-byte Address	3-byte Address
WRDYB	Write DYB	FB	4-byte Address	3-byte Address
RDPPB	Read PPB	FC	4-byte Address	3-byte Address
PGPPB	Program PPB	FD	4-byte Address	3-byte Address

Table 8.4 All Instruction Set

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
NORD	Normal Read Mode (3-byte Address)	SPI	03h	A <23:16>	A <15:8>	A <7:0>	Data out		
NORD	Normal Read Mode (4-byte Address)	SPI	03h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data out	
4NORD	4-byte Address Normal Read Mode	SPI	13h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data out	
FRD	Fast Read Mode (3-byte Address)	SPI QPI	0Bh	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Data out	
FRD	Fast Read Mode (4-byte Address)	SPI QPI	0Bh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Data out
4FRD	4-byte Address Fast Read Mode	SPI QPI	0Ch	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Data out
FRDIO	Fast Read Dual I/O (3-byte Address)	SPI	BBh	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh(1),(2) Dual	Dual Data out	
FRDIO	Fast Read Dual I/O (4-byte Address)	SPI	BBh	A <31::24>	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh(1),(2) Dual	Dual Data out
4FRDIO	4-byte Address Fast Read Dual I/O	SPI	BCh	A <31::24>	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh(1),(2) Dual	Dual Data out
FRDO	Fast Read Dual Output (3-byte Address)	SPI	3Bh	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Dual Data out	
FRDO	Fast Read Dual Output (4-byte Address)	SPI	3Bh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Dual Data out
4FRDO	4-byte Address Fast Read Dual Output	SPI	3Ch	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Dual Data out
FRQIO	Fast Read Quad I/O (3-byte Address)	SPI QPI	EBh	A <23:16> Quad	A <15:8> Quad	A <7:0> Quad	AXh(1), (2) Quad	Quad Data out	
FRQIO	Fast Read Quad I/O (4-byte Address)	SPI QPI	EBh	A <31::24> Quad	A <23:16> Quad	A <15:8> Quad	A <7:0> Quad	AXh(1), (2) Quad	Quad Data out
4FRQIO	4-byte Address Fast Read Quad I/O	SPI QPI	ECh	A <31::24> Quad	A <23:16> Quad	A <15:8> Quad	A <7:0> Quad	AXh(1), (2) Quad	Quad Data out
FRQO	Fast Read Quad Output (3-byte Address)	SPI	6Bh	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Quad Data out	
FRQO	Fast Read Quad Output (4-byte Address)	SPI	6Bh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Quad Data out
4FRQO	4-byte Address Fast Read Quad Output	SPI	6Ch	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Quad Data out

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
FRDTR	Fast Read DTR Mode (3-byte Address)	SPI QPI	0Dh	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Dual Data out	
FRDTR	Fast Read DTR Mode (4-byte Address)	SPI QPI	0Dh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Dual Data out
4FRDTR	4-byte Address Fast Read DTR Mode	SPI QPI	0Eh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Dummy(1) Byte	Dual Data out
FRDDTR	Fast Read Dual I/O DTR (3-byte Address)	SPI	BDh	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh(1), (2) Dual	Dual Data out	
FRDDTR	Fast Read Dual I/O DTR (4-byte Address)	SPI	BDh	A <31::24>	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh(1), (2) Dual	Dual Data out
4FRDDTR	4-byte Address Fast Read Dual I/O DTR	SPI	BEh	A <31::24>	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh(1), (2) Dual	Dual Data out
FRQDTR	Fast Read Quad I/O DTR (3-byte Address)	SPI QPI	EDh	A <23:16>	A <15:8>	A <7:0>	AXh(1), (2) Quad	Quad Data out	
FRQDTR	Fast Read Quad I/O DTR (4-byte Address)	SPI QPI	EDh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	AXh(1), (2) Quad	Quad Data out
4FRQDTR	4-byte Address Fast Read Quad I/O DTR	SPI QPI	EEh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	AXh(1), (2) Quad	Quad Data out
PP	Input Page Program (3-byte Address)	SPI QPI	02h	A <23:16>	A <15:8>	A <7:0>	PD (256byte)		
PP	Input Page Program (4-byte Address)	SPI QPI	02h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	PD (256byte)	
4PP	4-byte Address Input Page Program	SPI QPI	12h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	PD (256byte)	
PPQ	Quad Input Page Program (3-byte Address)	SPI	32h/38h	A <23:16>	A <15:8>	A <7:0>	Quad PD (256byte)		
PPQ	Quad Input Page Program (4-byte Address)	SPI	32h/38h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Quad PD (256byte)	
4PPQ	4-byte Address Quad Input Page Program	SPI	34h/3Eh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Quad PD (256byte)	
SER	Sector Erase (3-byte Address)	SPI QPI	D7h/20h	A <23:16>	A <15:8>	A <7:0>			
SER	Sector Erase (4-byte Address)	SPI QPI	D7h/20h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
4SER	4-byte Address Sector Erase	SPI QPI	21h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
BER32 (32KB)	Block Erase 32Kbyte (3-byte Address)	SPI QPI	52h	A <23:16>	A <15:8>	A <7:0>			
BER32 (32KB)	Block Erase 32Kbyte (4-byte Address)	SPI QPI	52h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
4BER32 (32KB)	4-byte Address Block Erase 32Kbyte	SPI QPI	5Ch	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
BER64 (64KB)	Block Erase 64Kbyte (3-byte Address)	SPI QPI	D8h	A <23:16>	A <15:8>	A <7:0>			
BER64 (64KB)	Block Erase 64Kbyte (4-byte Address)	SPI QPI	D8h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
4BER64 (64KB)	4-byte Address Block Erase 64Kbyte	SPI QPI	DCh	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
CER	Chip Erase	SPI QPI	C7h/60h						
WREN	Write Enable	SPI QPI	06h						
WRDI	Write Disable	SPI QPI	04h						
RDSR	Read Status Register	SPI QPI	05h	Data out					
WRSR	Write Status Register	SPI QPI	01h	Data in					
RDFR	Read Function Register	SPI QPI	48h	Data out					
WRFR	Write Function Register	SPI QPI	42h	Data in					
QIOEN	Enter QPI mode	SPI	35h						
QIODI	Exit QPI mode	QPI	F5h						
PERSUS	Suspend during program/erase	SPI QPI	75h/B0h						
PERRSM	Resume program/erase	SPI QPI	7Ah/30h						
DP	Deep Power Down	SPI QPI	B9h						
RDID, RDPD	Read ID / Release Power Down	SPI QPI	ABh	XXh(3)	XXh(3)	XXh(3)	ID7-ID0		
SRPNV	Set Read Parameters (Non-Volatile)	SPI QPI	65h	Data in					
SRPV	Set Read Parameters (Volatile)	SPI QPI	C0h/63h	Data in					
SERPNV	Set Extended Read Parameters (Non-Volatile)	SPI QPI	85h	Data in					

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
SERPV	Set Extended Read Parameters (Volatile)	SPI QPI	83h	Data in					
RDRP	Read Read Parameters (Volatile)	SPI QPI	61h	Data out					
RDERP	Read Extended Read Parameters (Volatile)	SPI QPI	81h	Data out					
CLERP	Clear Extended Read Register	SPI QPI	82h						
RDJDID	Read JEDEC ID Command	SPI QPI	9Fh	MF7-MF0	ID15-ID8	ID7-ID0			
RDMDID	Read Manufacturer & Device ID	SPI QPI	90h	XXh(3)	XXh(3)	00h	MF7-MF0	ID7-ID0	
						01h	ID7-ID0	MF7-MF0	
RDJDIDQ	Read JEDEC ID QPI mode	QPI	AFh	MF7-MF0	ID15-ID8	ID7-ID0			
RDUID	Read Unique ID	SPI QPI	4Bh	A(4) <23:16>	A(4) <15:8>	A(4) <7:0>	Dummy Byte	Data out	
RDSFDP	SFDP Read	SPI QPI	5Ah	A <23:16>	A <15:8>	A <7:0>	Dummy Byte	Data out	
NOP	No Operation	SPI QPI	00h						
RSTEN	Software Reset Enable	SPI QPI	66h						
RST	Software Reset	SPI QPI	99h						
IRER	Erase Information Row	SPI QPI	64h	A <23:16>	A <15:8>	A <7:0>			
IRP	Program Information Row	SPI QPI	62h	A <23:16>	A <15:8>	A <7:0>	PD (256byte)		
IRRD	Read Information Row	SPI QPI	68h	A <23:16>	A <15:8>	A <7:0>	Dummy Byte	Data out	
SECUNLOCK	Sector Unlock (3-byte Address)	SPI QPI	26h	A <23:16>	A <15:8>	A <7:0>			
SECUNLOCK	Sector Unlock (4-byte Address)	SPI QPI	26h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
4SECUNLOCK	4-byte Address Sector Unlock	SPI QPI	25h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
SECLOCK	Sector Lock	SPI QPI	24h						
RDABR	Read AutoBoot Register	SPI QPI	14h						
WRABR	Write AutoBoot Register	SPI QPI	15h	Data in 1	Data in 2	Data in 3	Data in 4		

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
RDBR	Read Bank Address Register (Volatile)	SPI QPI	16h/C8h	Data out					
WRBRV	Write Bank Address Register (Volatile)	SPI QPI	17h/C5h	Data in					
WRBRNV	Write Bank Address Register (Non-Volatile)	SPI QPI	18h	Data in					
EN4B	Enter 4-byte Address Mode	SPI QPI	B7h						
EX4B	Exit 4-byte Address Mode	SPI QPI	29h						
RDDYB	Read DYB (3-byte Address)	SPI QPI	FAh	A <23:16>	A <15:8>	A <7:0>	Data out		
RDDYB	Read DYB (4-byte Address)	SPI QPI	FAh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data out	
4RDDYB	4-byte Address Read DYB	SPI QPI	E0h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data out	
WRDYB	Write DYB (3-byte Address)	SPI QPI	FBh	A <23:16>	A <15:8>	A <7:0>	Data in		
WRDYB	Write DYB (4-byte Address)	SPI QPI	FBh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data in	
4WRDYB	4-byte Address Write DYB	SPI QPI	E1h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data in	
RDPPB	Read PPB (3-byte Address)	SPI	FCh	A <23:16>	A <15:8>	A <7:0>	Data out		
RDPPB	Read PPB (4-byte Address)	SPI	FCh	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data out	
4RDPPB	4-byte Address Read PPB	SPI	E2h	A <31::24>	A <23:16>	A <15:8>	A <7:0>	Data out	
PGPPB	Program PPB (Individually) (3-byte Address)	SPI QPI	FDh	A <23:16>	A <15:8>	A <7:0>			
PGPPB	Program PPB (Individually) (4-byte Address)	SPI QPI	FDh	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
4PGPPB	4-byte Address Program PPB (Individually)	SPI QPI	E3h	A <31::24>	A <23:16>	A <15:8>	A <7:0>		
ERPPB	Erase PPB (as a group)	SPI QPI	E4h						
RDASP	Read ASP	SPI QPI	2Bh	Data out (2 byte)					
PGASP	Program ASP	SPI QPI	2Fh	PD (2 byte)					
RDPLB	Read PPB Lock Bit	SPI QPI	A7h	Data out					
WRPLB	Write PPB Lock Bit	SPI QPI	A6h						
SFRZ	Set FREEZE bit	SPI QPI	91h						

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
RDPWD	Read Password	SPI QPI	E7h	Data out (8 byte)					
PGPWD	Program Password	SPI QPI	E8h	PD (8 byte)					
UNPWD	Unlock Password	SPI QPI	E9h	Data in (8 byte)					
GBLK	Set all DYB bits (Gang Sector/ Block Lock)	SPI QPI	7Eh						
GBUN	Clear all DYB bits (Gang Sector/ Block Unlock)	SPI QPI	98h						

Notes:
1. **The number of dummy cycles depends on the value setting in the Table 6.11 Read Dummy Cycles.**
2. **AXh has to be counted as a part of dummy cycles. X means "don't care".**
3. **XX means "don't care".**
4. **A<23:9> are "don't care" and A<8:4> are always "0".**

8.3 NORMAL READ OPERATION (NORD, 03h or 4NORD, 13h)

The Normal Read (NORD) instruction is used to read memory contents at a maximum frequency of 80MHz.

- 03h (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 03h (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 13h is followed by a 4-byte address (A31-A0)

The Normal Read instruction code is transmitted via the SI line, followed by three (A23 - A0) or four (A31 – A0) address bytes of the first memory location to be read as above. A total of 24 or 32 address bits are shifted in, but only A_{VMSB} (Valid Most Significant Bit) - A_0 are decoded. The remaining bits (A31 – A_{VMSB+1}) are ignored. The first byte addressed can be at any memory location. Upon completion, any data on the SI will be ignored. Refer to Table 8.5 for the related Address Key.

The first byte data (D7 - D0) is shifted out on the SO line, MSB first. A single byte of data, or up to the whole memory array, can be read out in one Normal Read instruction. The address is automatically incremented by one after each byte of data is shifted out. The read operation can be terminated at any time by driving CE# high (VIH) after the data comes out. When the highest address of the device is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read in one continuous Read instruction.

If the Normal Read instruction is issued while an Erase, Program or Write operation is in process (WIP=1) the instruction is ignored and will not have any effects on the current operation.

Table 8.5 Address Key

Mode	**Valid Address**	**256Mb**
3 byte address	A_{VMSB}(2)–A_0	A23-A0
4 byte address		A24-A0 (A31-A25=X)(2)

Notes:

1. **X=Don't Care**
2. **A_{VMSB} is a Valid MSB. In 4 byte address for 256Mb, A31 is an MSB, and A24 is a Valid MSB.**

Figure 8.1 Normal Read Sequence (03h [EXTADD=0], 3-byte address)

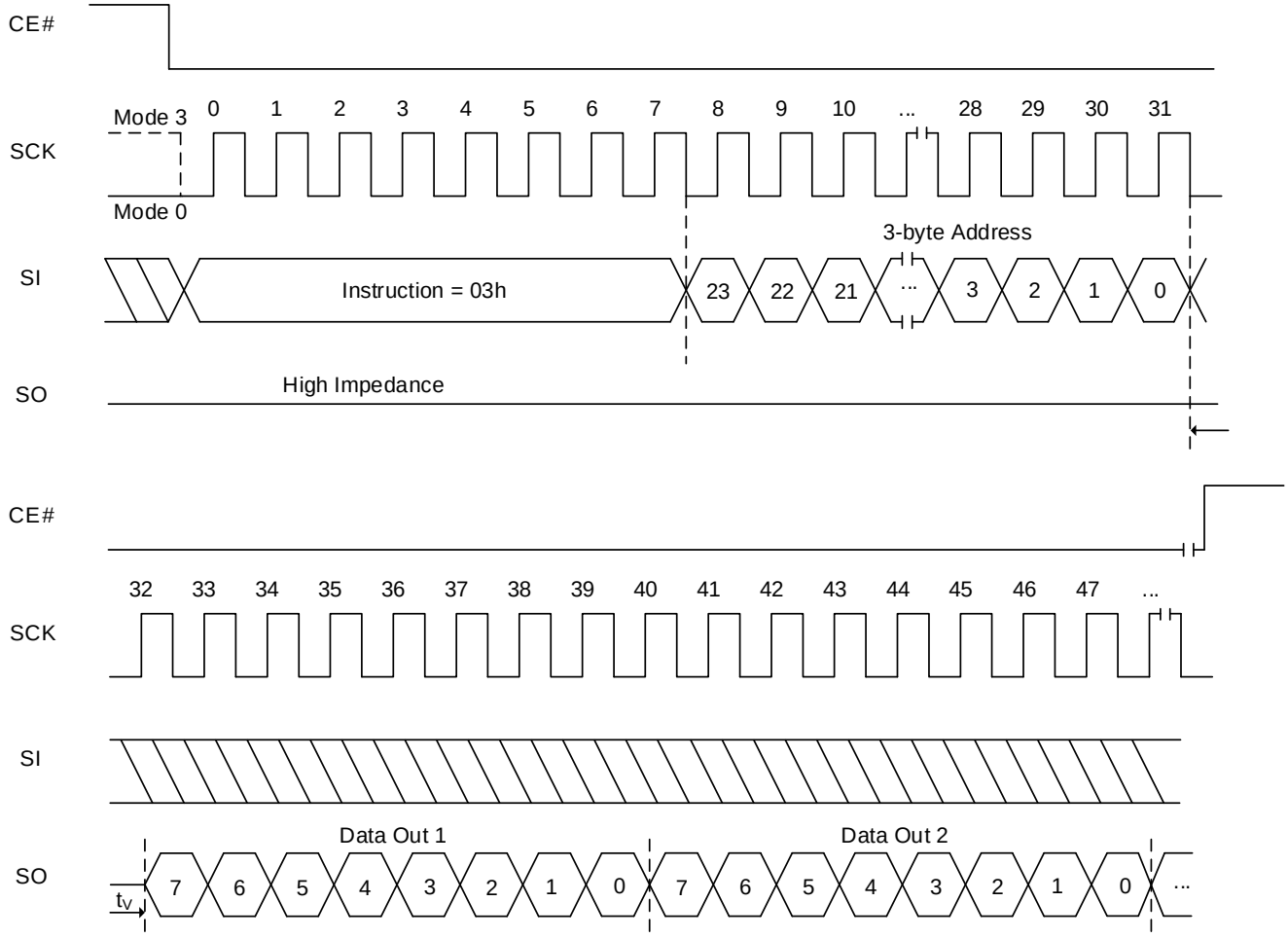
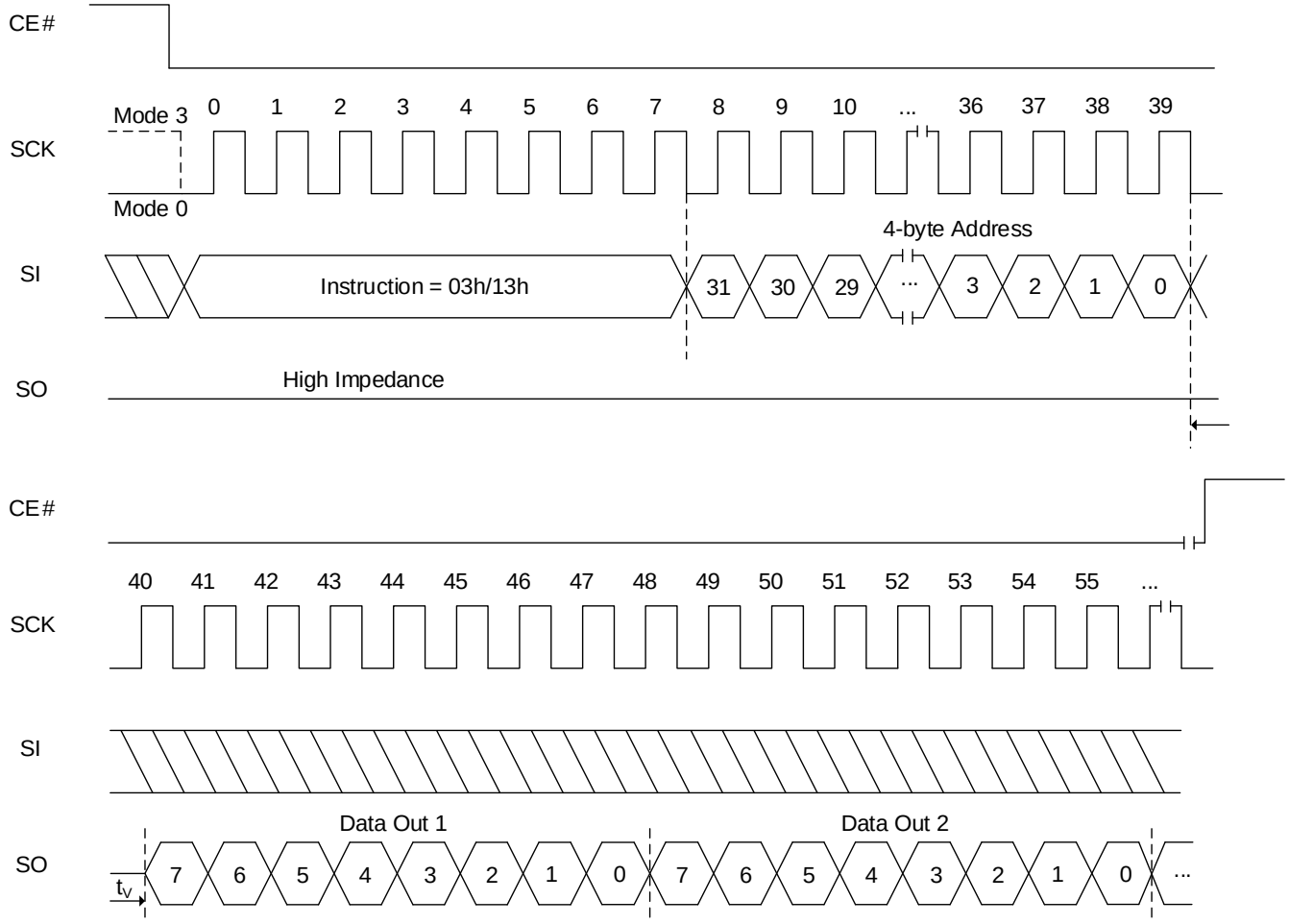


Figure 8.2 Normal Read Sequence (03h [EXTADD=1] or 13h, 4-byte address)



8.4 FAST READ OPERATION (FRD, 0Bh or 4FRD, 0Ch)

The Fast Read (FRD, 4FRD) instruction is used to read memory data at up to a 166MHz clock.

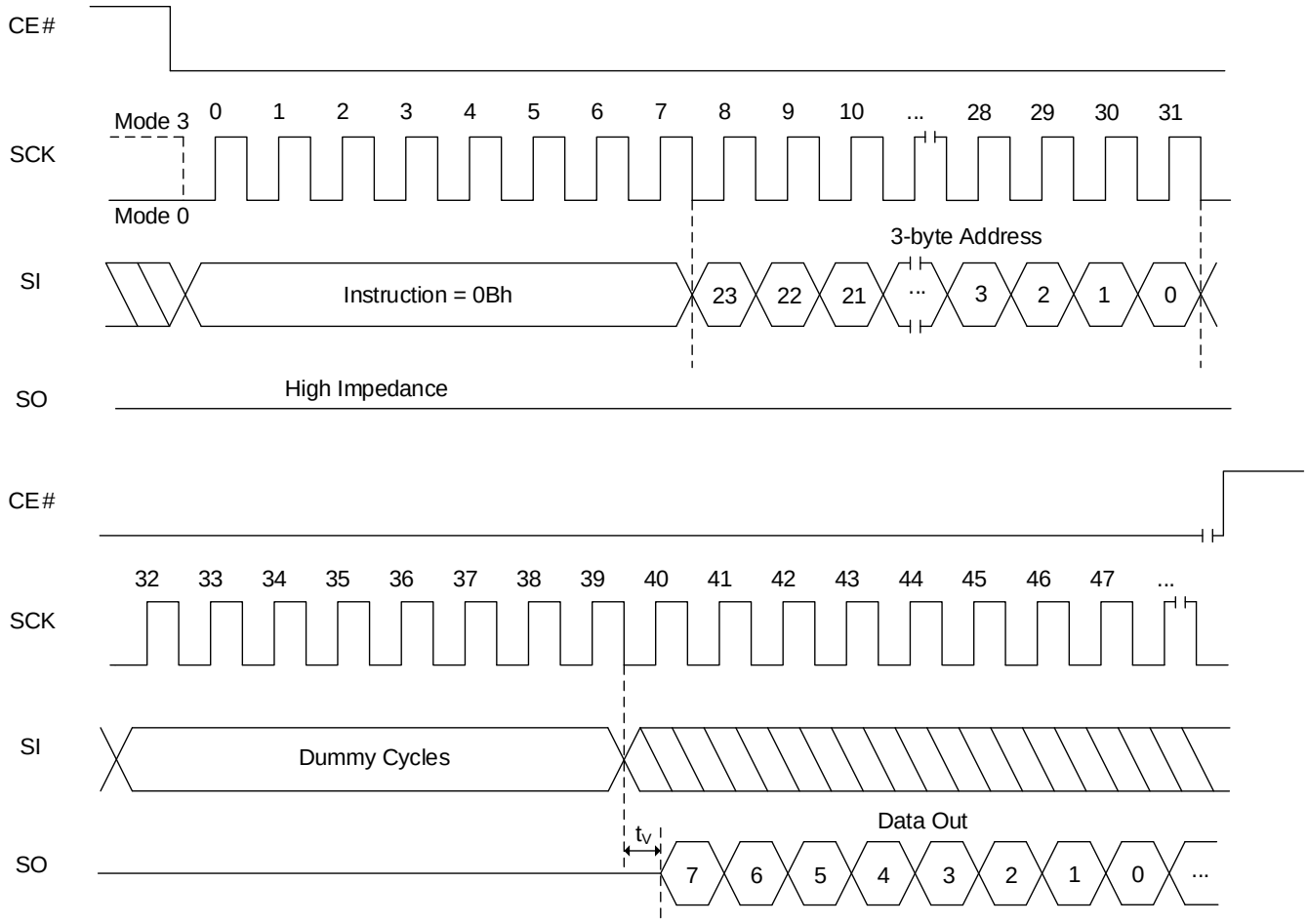
- 0Bh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 0Bh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 0Ch is followed by a 4-byte address (A31-A0)

The Fast Read instruction code is followed by three or four address bytes as above and dummy cycles (configurable, default is 8 clocks), transmitted via the SI line, with each bit latched-in during the rising edge of SCK. Then the first data byte from the address is shifted out on the SO line, with each bit shifted out at a maximum frequency f_{CT}, during the falling edge of SCK.

The first byte addressed can be at any memory location. The address is automatically incremented by one after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single Fast Read instruction. The Fast Read instruction is terminated by driving CE# high (VIH).

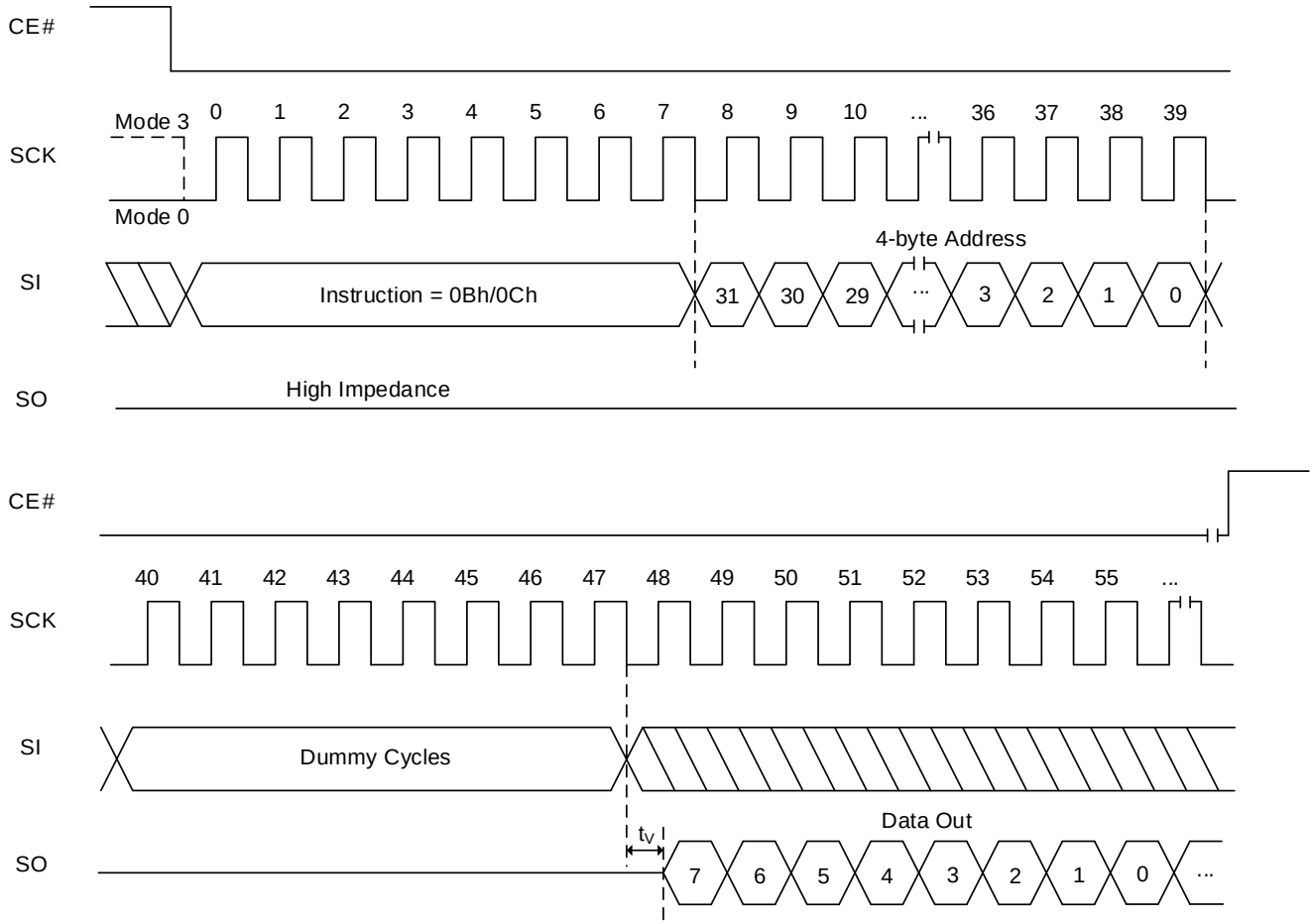
If the Fast Read instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored without affecting the current cycle.

Figure 8.3 Fast Read Sequence (0Bh [EXTADD=0], 3-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

Figure 8.4 Fast Read Sequence (0Bh [EXTADD=1] or 0Ch, 4-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

FAST READ OPERATION IN QPI MODE (FRD, 0Bh or 4FRD, 0Ch)

The Fast Read (FRD) instruction in QPI mode is used to read memory data at up to a 166MHz clock.

- 0Bh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 0Bh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 0Ch is followed by a 4-byte address (A31-A0)

The Fast Read instruction code (2 clocks) is followed by three (6 clocks) or four (8 clocks) address bytes as above and 6 dummy cycles (configurable, default is 6 clocks), transmitted via the IO3, IO2, IO1 and IO0 lines, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the IO3, IO2, IO1 and IO0 lines, with each bit shifted out at a maximum frequency f_{CT}, during the falling edge of SCK.

The first byte addressed can be at any memory location. The address is automatically incremented by one after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single Fast Read QPI instruction. The Fast Read QPI instruction is terminated by driving CE# high (VIH).

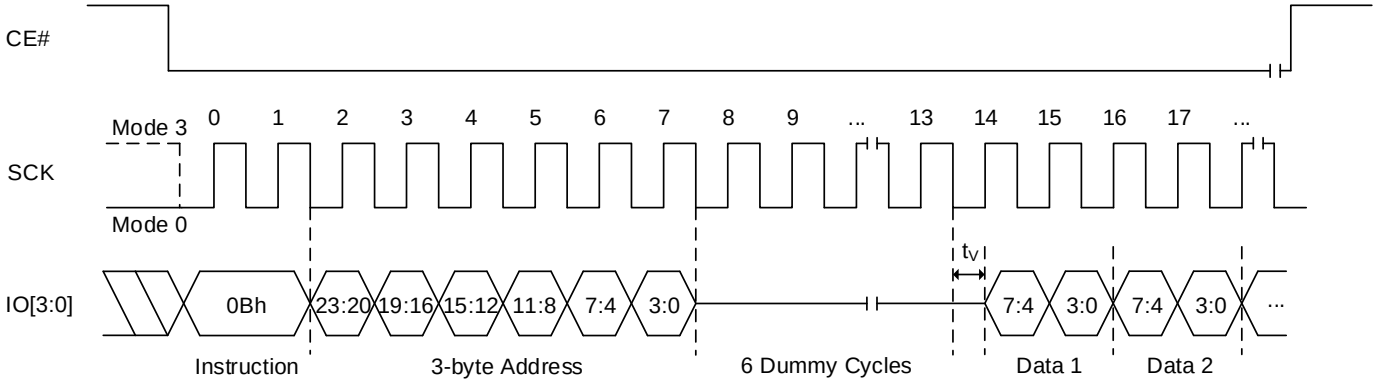
If the Fast Read instruction in QPI mode is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored without affecting the current cycle.

The Fast Read sequence in QPI mode is also applied to the commands in the following table 8.6. However, only 3-byte address mode QPI sequence is applied for RDUID, RDSFDP, and IRRD commands.

Table 8.6 Instructions that Fast Read sequence in QPI Mode is applied to

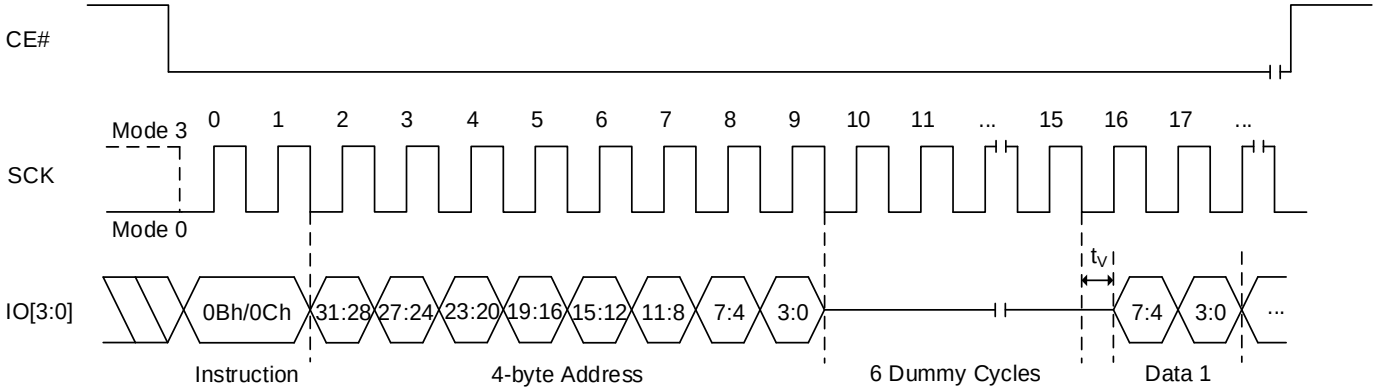
Instruction Name	Operation	Hex Code
FRQIO	Fast Read Quad I/O	EBh
RDUID	Read Unique ID	4Bh
RDSFDP	SFDP Read	5Ah
IRRD	Read Information Row	68h

Figure 8.5 Fast Read QPI Sequence (0Bh [EXTADD=0], 3-byte address)



Note: Number of dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

Figure 8.6 Fast Read Sequence In QPI Mode (0Bh [EXTADD=1] or 0Ch, 4-byte address)



Note: Number of dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

8.5 HOLD OPERATION

HOLD# is used in conjunction with CE# to select the device. When the device is selected and a serial sequence is underway, HOLD# can be used to pause the serial communication with the master device without resetting the serial sequence. To pause, HOLD# is brought low while the SCK signal is low. To resume serial communication, HOLD# is brought high while the SCK signal is low (SCK may still toggle during HOLD). Inputs to SI will be ignored while SO is in the high impedance state, during HOLD.

Note: HOLD is not supported in DTR mode or with QE=1 or when RESET# is selected for the HOLD# or RESET# pin.

Timing graph can be referenced in AC Parameters Figure 9.4.

8.6 FAST READ DUAL I/O OPERATION (FRDIO, BBh or 4FRDIO, BCh)

The Fast Read Dual I/O (FRDIO, 4FRDIO) instruction allows the address bits to be input two bits at a time. This may allow for code to be executed directly from the SPI in some applications.

- BBh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- BBh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- BCh is followed by a 4-byte address (A31-A0)

The FRDIO/4FRDIO instruction code is followed by three or four address bytes as above and dummy cycles (configurable, default is 4 clocks), transmitted via the IO1 and IO0 lines, with each pair of bits latched-in during the rising edge of SCK. The address MSB is input on IO1, the next bit on IO0, and this shift pattern continues to alternate between the two lines. Depending on the usage of AX read operation mode, a mode byte may be located after address input.

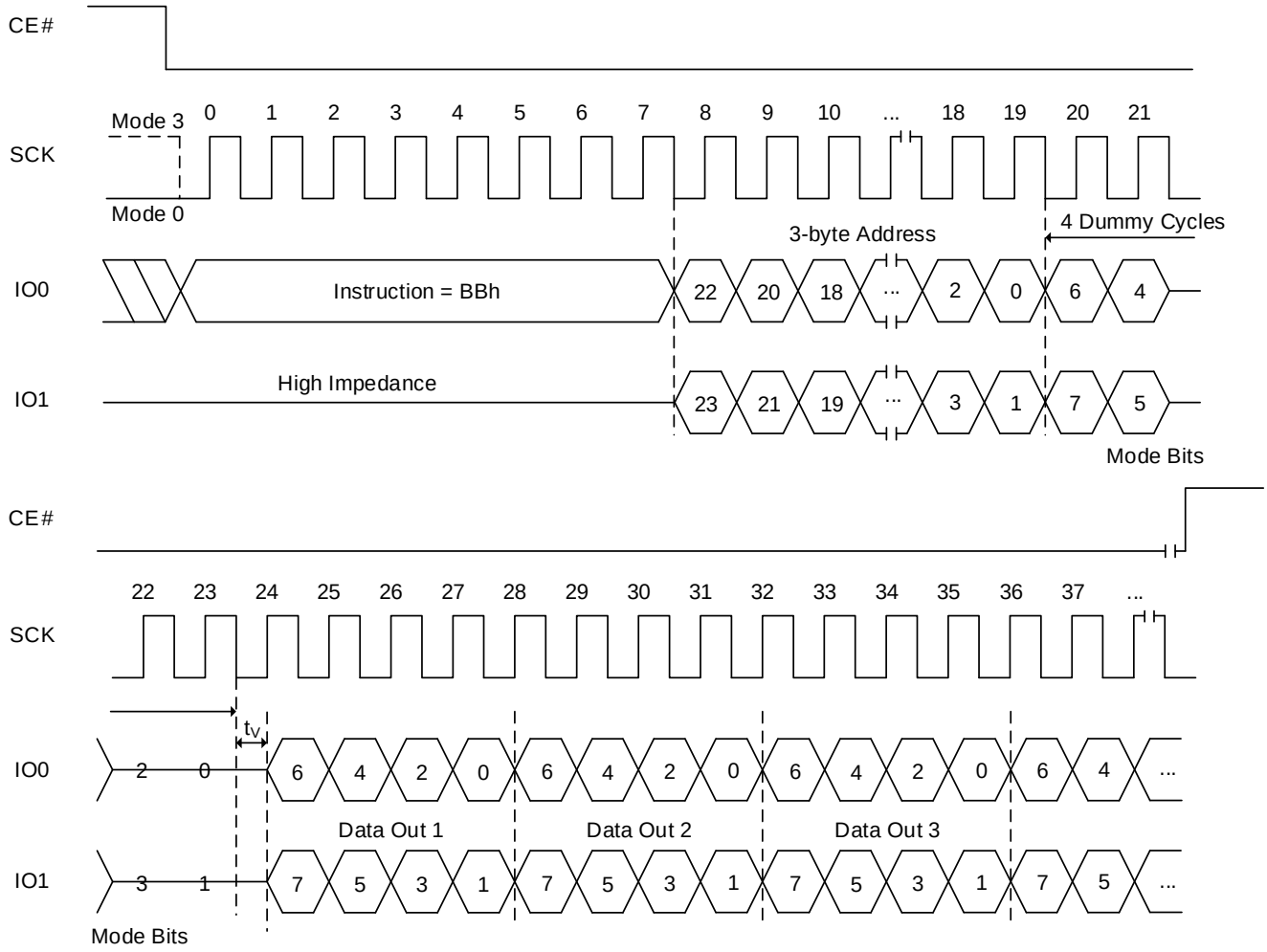
The first data byte addressed is shifted out on the IO1 and IO0 lines, with each pair of bits shifted out at a maximum frequency f_{CT}, during the falling edge of SCK. The first bit (MSB) is output on IO1, while simultaneously the second bit is output on IO0. Figures 8.7 and 8.8 illustrates the timing sequence.

The first byte addressed can be at any memory location. The address is automatically incremented by one after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRDIO/4FRDIO instruction. The FRDIO/4FRDIO instruction is terminated by driving CE# high (V_{IH}).

The device supports the AX read operation by applying mode bits during dummy period. Mode bits consist of 8 bits, such as M7 to M0. Four cycles after address input are reserved for Mode bits in FRDIO/4FRDIO execution. M7 to M4 are important for enabling this mode. M3 to M0 become don't care for future use. When M[7:4]=1010(Ah), it enables the AX read operation and subsequent FRDIO/4FRDIO execution skips command code. It saves cycles as described in Figures 8.9 and 8.10. When the code is different from AXh (where X is don't care), the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command. SPI or QPI mode configuration retains the prior setting. Mode bit must be applied during dummy cycles. Number of dummy cycles in Table 6.11 includes number of mode bit cycles. If dummy cycles are configured as 4 cycles, data output will start right after mode bit is applied.

If the FRDIO/4FRDIO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not affect the current cycle.

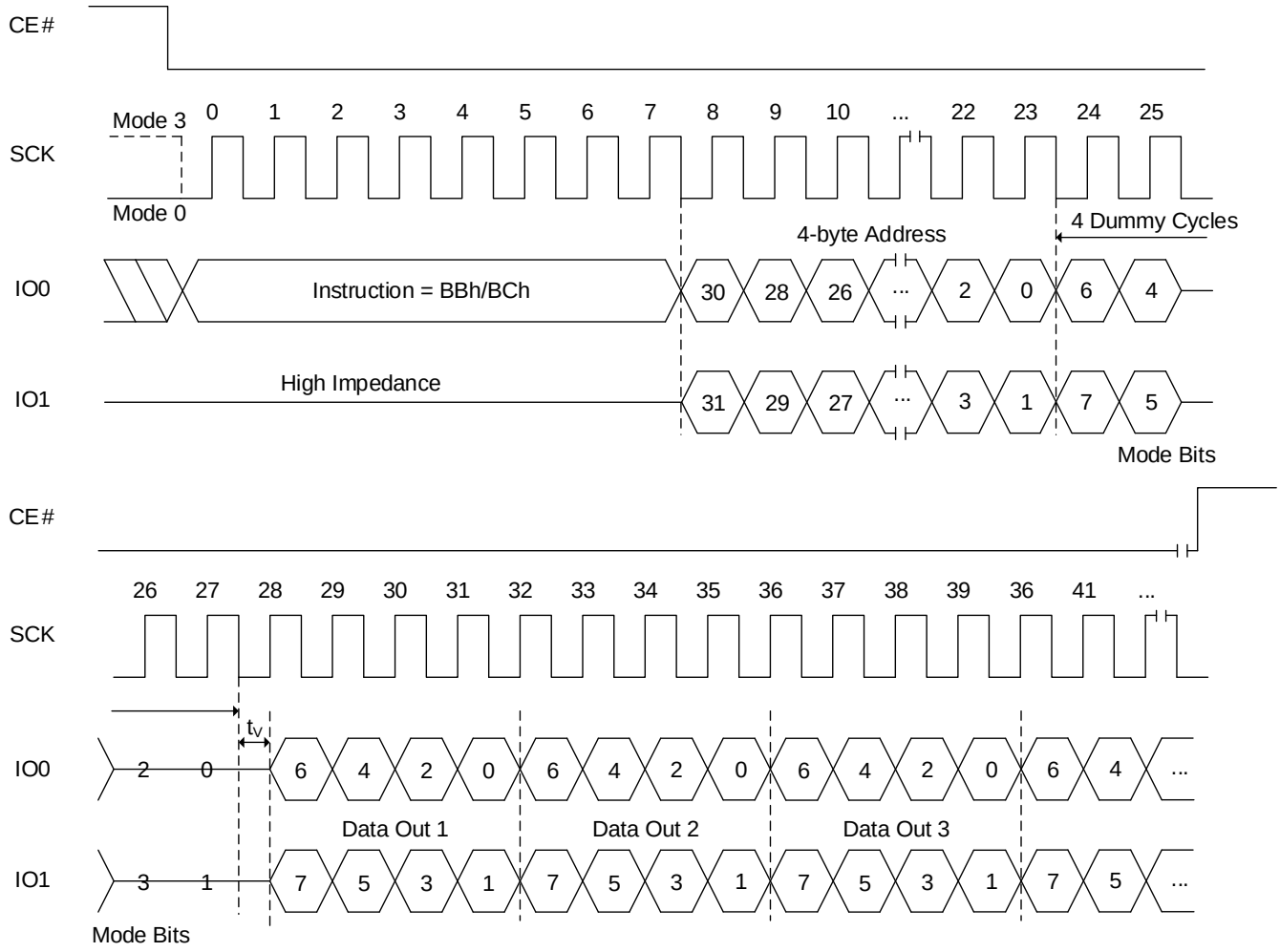
Figure 8.7 Fast Read Dual I/O Sequence (BBh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

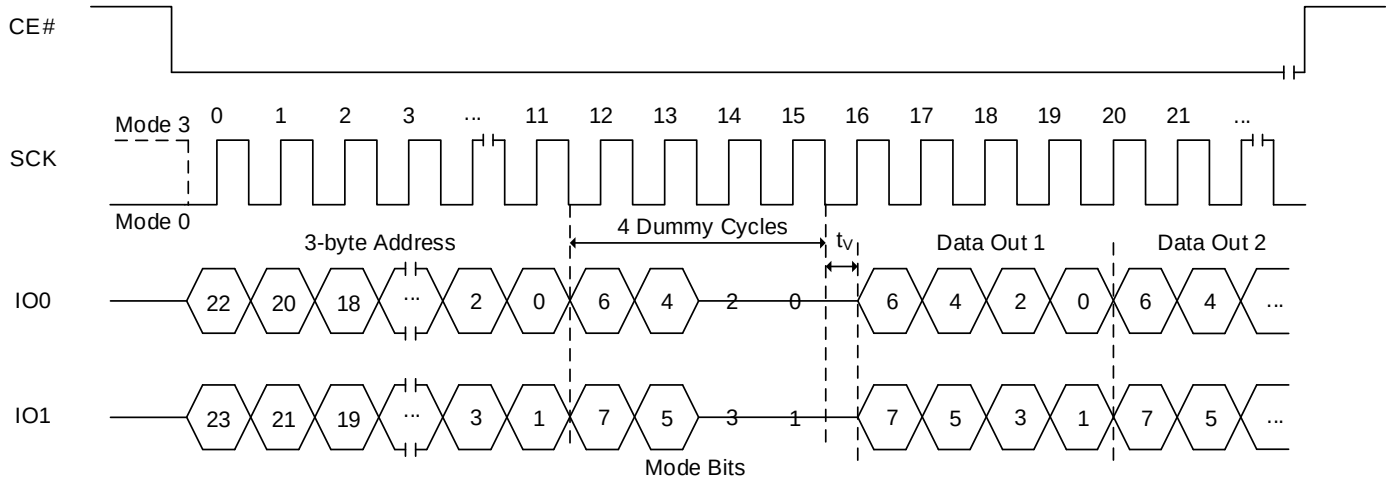
Figure 8.8 Fast Read Dual I/O Sequence (BBh [EXTADD=1] or BCh, 4-byte address)



Notes:

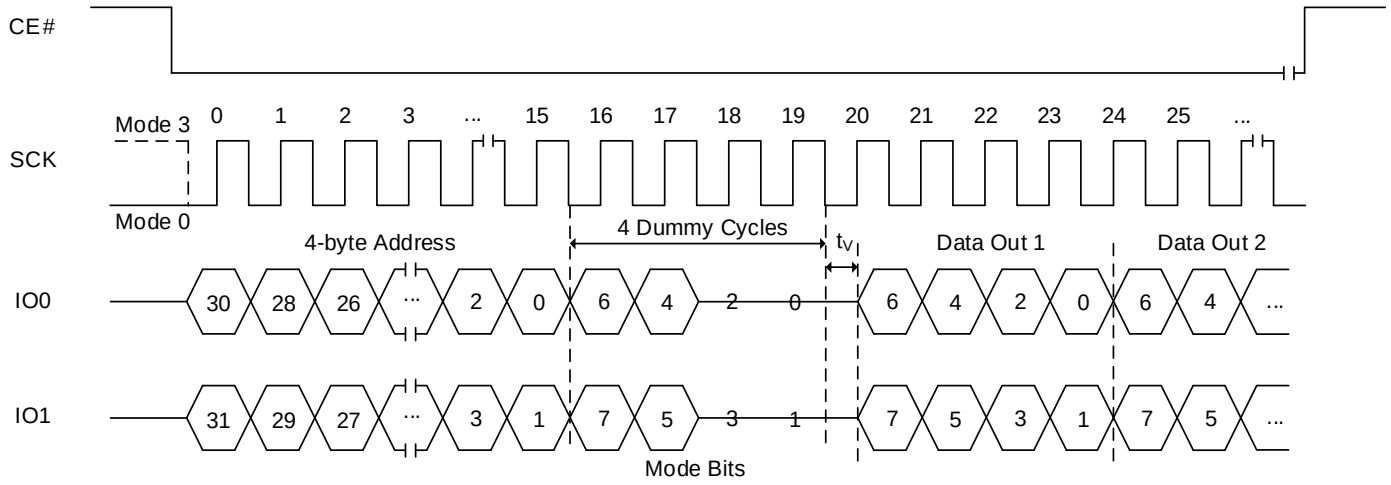
1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.9 Fast Read Dual I/O AX Read Sequence (BBh [EXTADD=0], 3-byte address)



Notes:
1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.10 Fast Read Dual I/O AX Read Sequence (BBh [EXTADD=1] or BCh, 4-byte address)



Notes:
1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

8.7 FAST READ DUAL OUTPUT OPERATION (FRDO, 3Bh or 4FRDO, 3Ch)

The FRDO/4FRDO instruction is used to read memory data on two output pins each at up to a 166MHz clock.

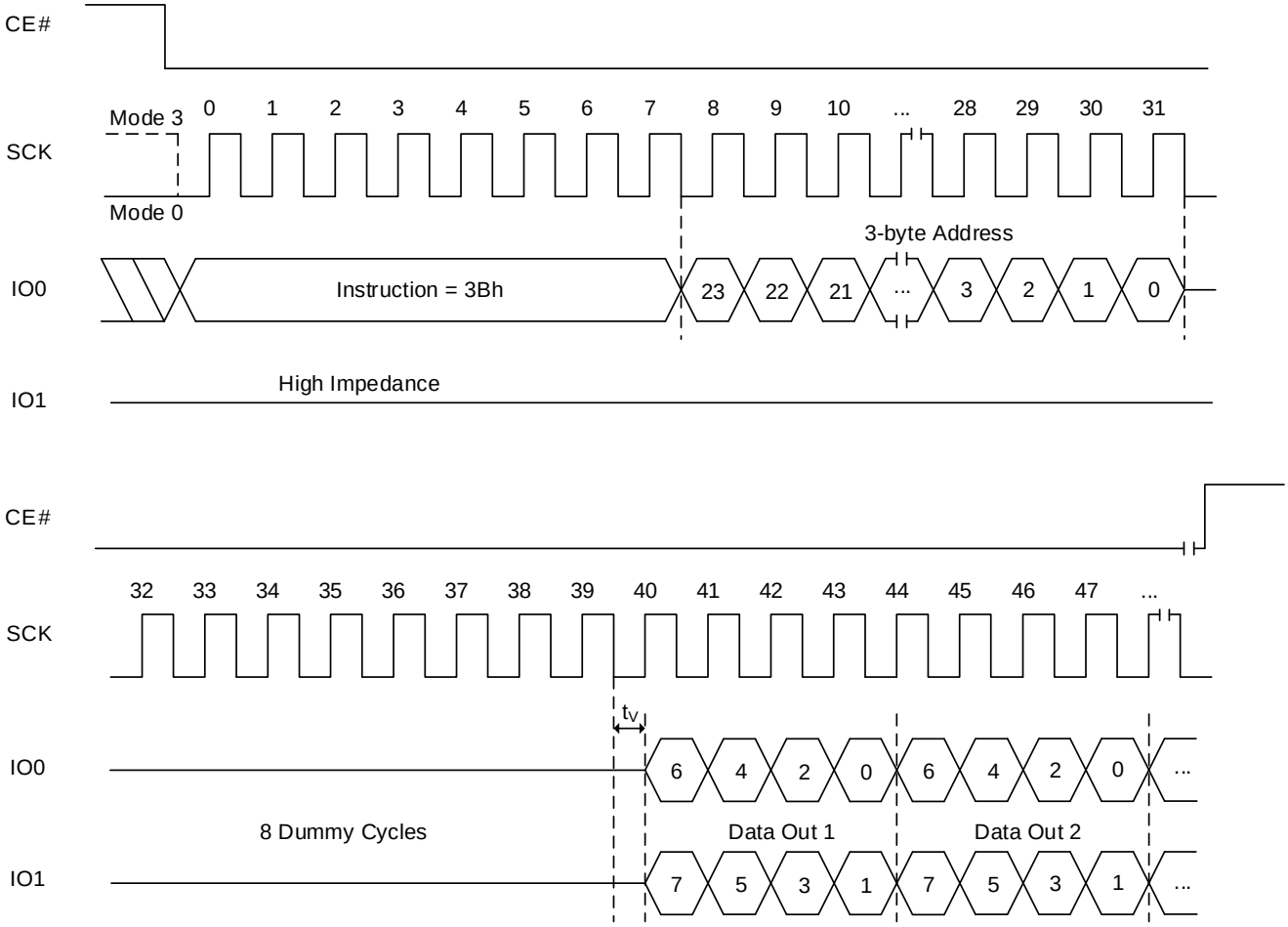
- 3Bh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 3Bh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 3Ch is followed by a 4-byte address (A31-A0)

The FRDO/4FRDO instruction code is followed by three or four address bytes as above and dummy cycles (configurable, default is 8 clocks), transmitted via the IO0 line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the IO1 and IO0 lines, with each pair of bits shifted out at a maximum frequency f_{CT}, during the falling edge of SCK. The first bit (MSB) is output on IO1. Simultaneously, the second bit is output on IO0.

The first byte addressed can be at any memory location. The address is automatically incremented by one after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRDO/4FRDO instruction. The instruction FRDO/4FRDO is terminated by driving CE# high (VIH).

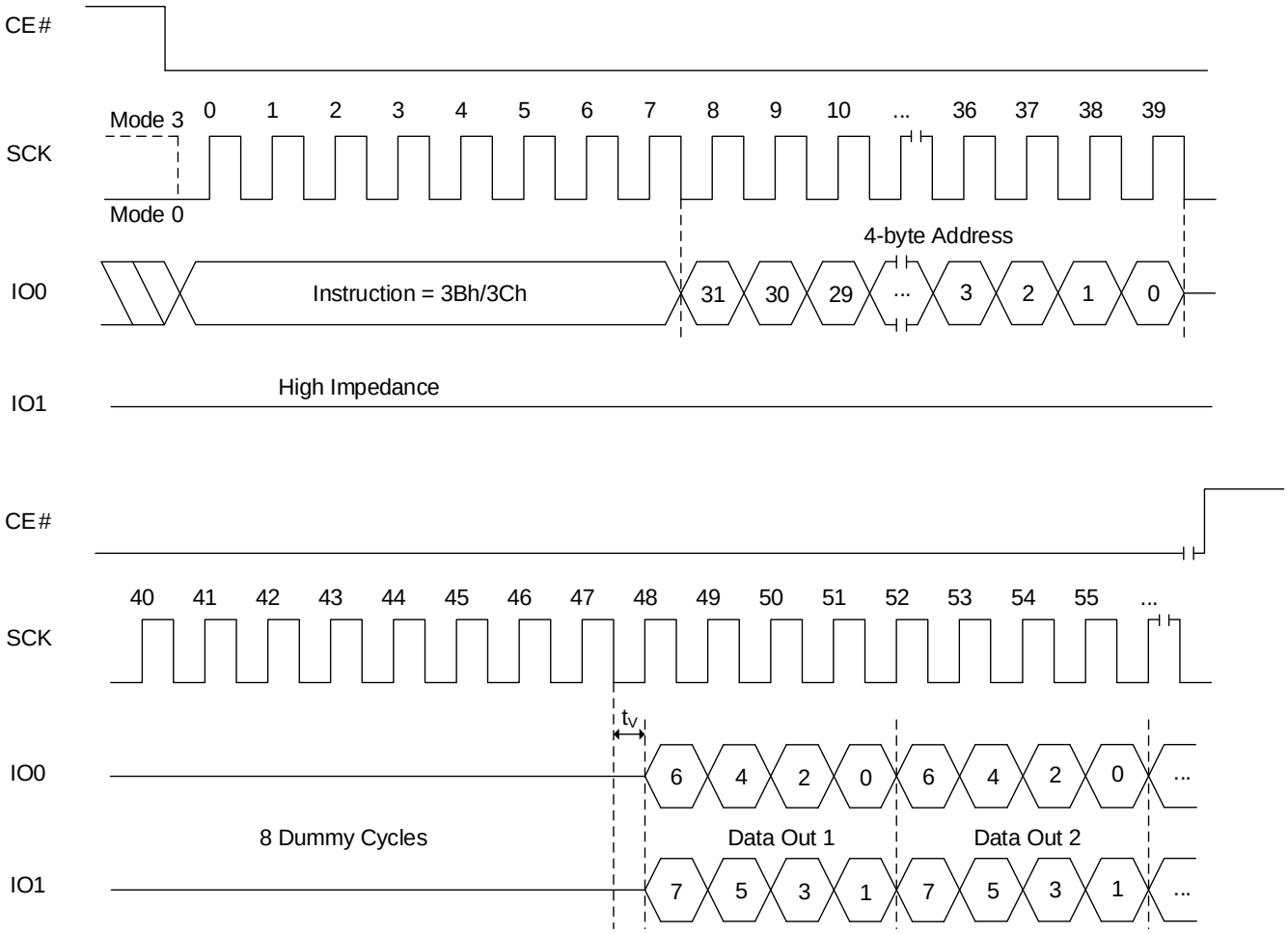
If the FRDO/4FRDO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle.

Figure 8.11 Fast Read Dual Output Sequence (3Bh [EXTADD=0], 3-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

Figure 8.12 Fast Read Dual Output Sequence (3Bh [EXTADD=1] or 3Ch, 4-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

8.8 FAST READ QUAD OUTPUT OPERATION (FRQO, 6Bh or 4FRQO 6Ch)

The FRQO/4FRQO instruction is used to read memory data on four output pins each at up to a 166 MHz clock.

A Quad Enable (QE) bit of Status Register must be set to "1" before sending the Fast Read Quad Output instruction.

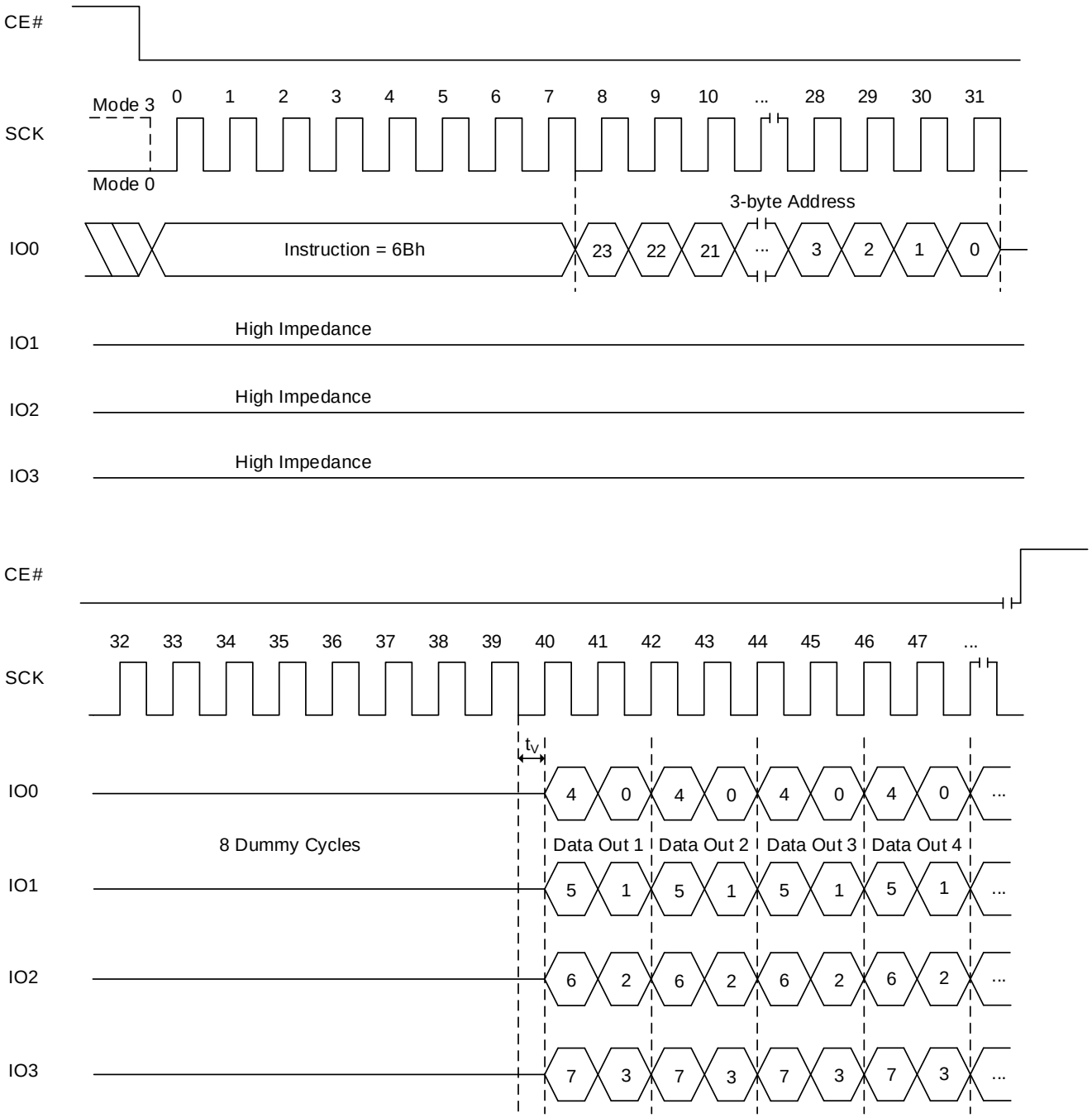
- 6Bh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 6Bh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 6Ch is followed by a 4-byte address (A31-A0)

The FRQO/4FRQO instruction code is followed by three or four address bytes as above and dummy cycles (configurable, default is 8 clocks), transmitted via the IO0 line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the IO3, IO2, IO1 and IO0 lines, with each group of four bits shifted out at a maximum frequency f_{CT}, during the falling edge of SCK. The first bit (MSB) is output on IO3, while simultaneously the second bit is output on IO2, the third bit is output on IO1, etc.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRQO/4FRQO instruction. FRQO/4FRQO instruction is terminated by driving CE# high (VIH).

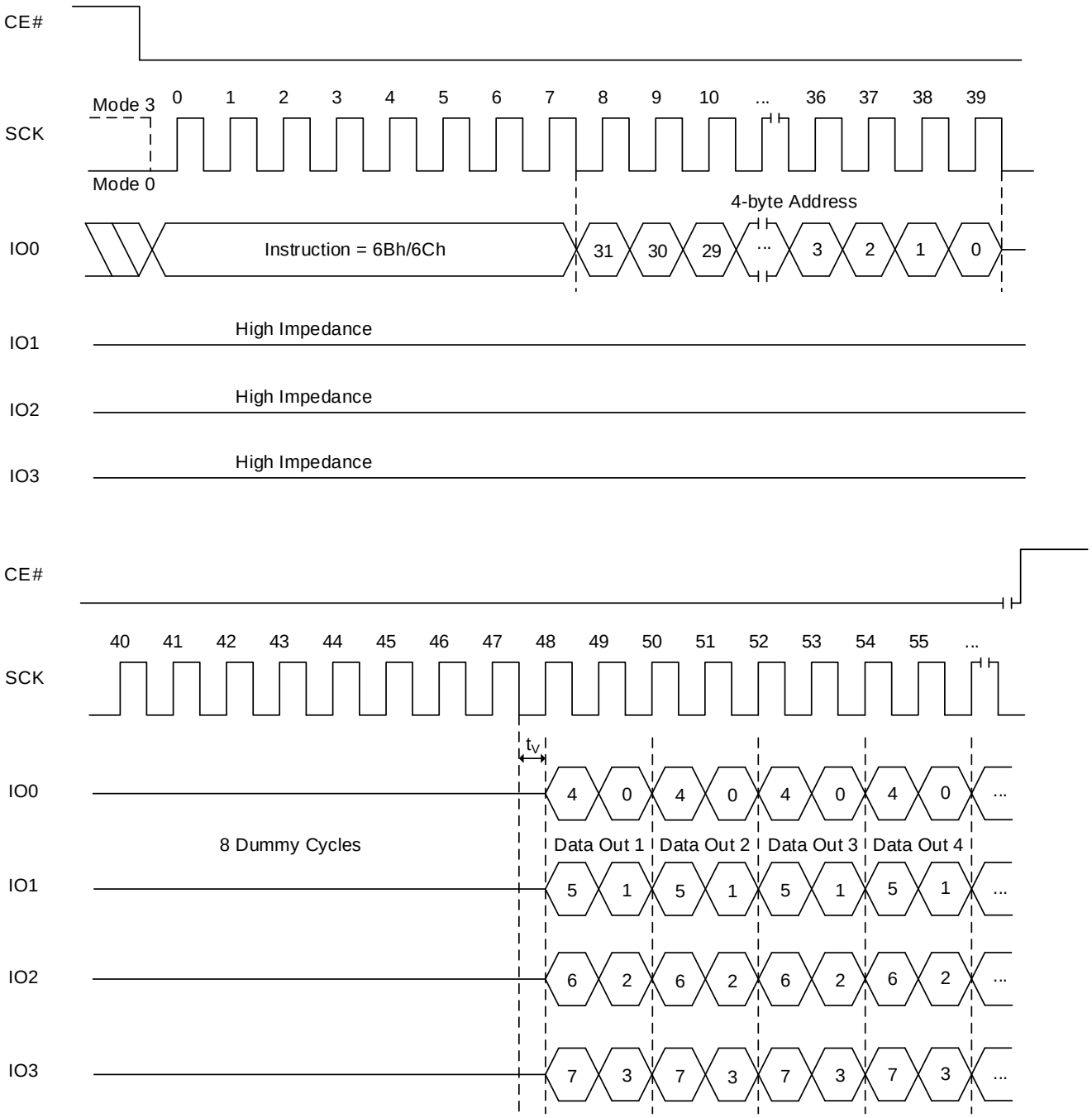
If a FRQO/4FRQO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle.

Figure 8.13 Fast Read Quad Output Sequence (6Bh [EXTADD=0], 3-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

Figure 8.14 Fast Read Quad Output Sequence (6Bh [EXTADD=1] or 6Ch, 4-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

8.9 FAST READ QUAD I/O OPERATION (FRQIO, EBh or 4FRQIO, ECh)

The FRQIO/4FRQIO instruction allows the address bits to be input four bits at a time. This may allow for code to be executed directly from the SPI in some applications.

A Quad Enable (QE) bit of Status Register must be set to "1" before sending the Fast Read Quad I/O instruction.

- EBh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- EBh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- ECh is followed by a 4-byte address (A31-A0)

The FRQIO/4FRQIO instruction code is followed by three or four address bytes as above and dummy cycles (configurable, default is 6 clocks), transmitted via the IO3, IO2, IO1 and IO0 lines, with each group of four bits latched-in during the rising edge of SCK. The address of MSB inputs on IO3, the next bit on IO2, the next bit on IO1, the next bit on IO0, and continue to shift in alternating on the four. Depending on the usage of AX read operation mode, a mode byte may be located after address input.

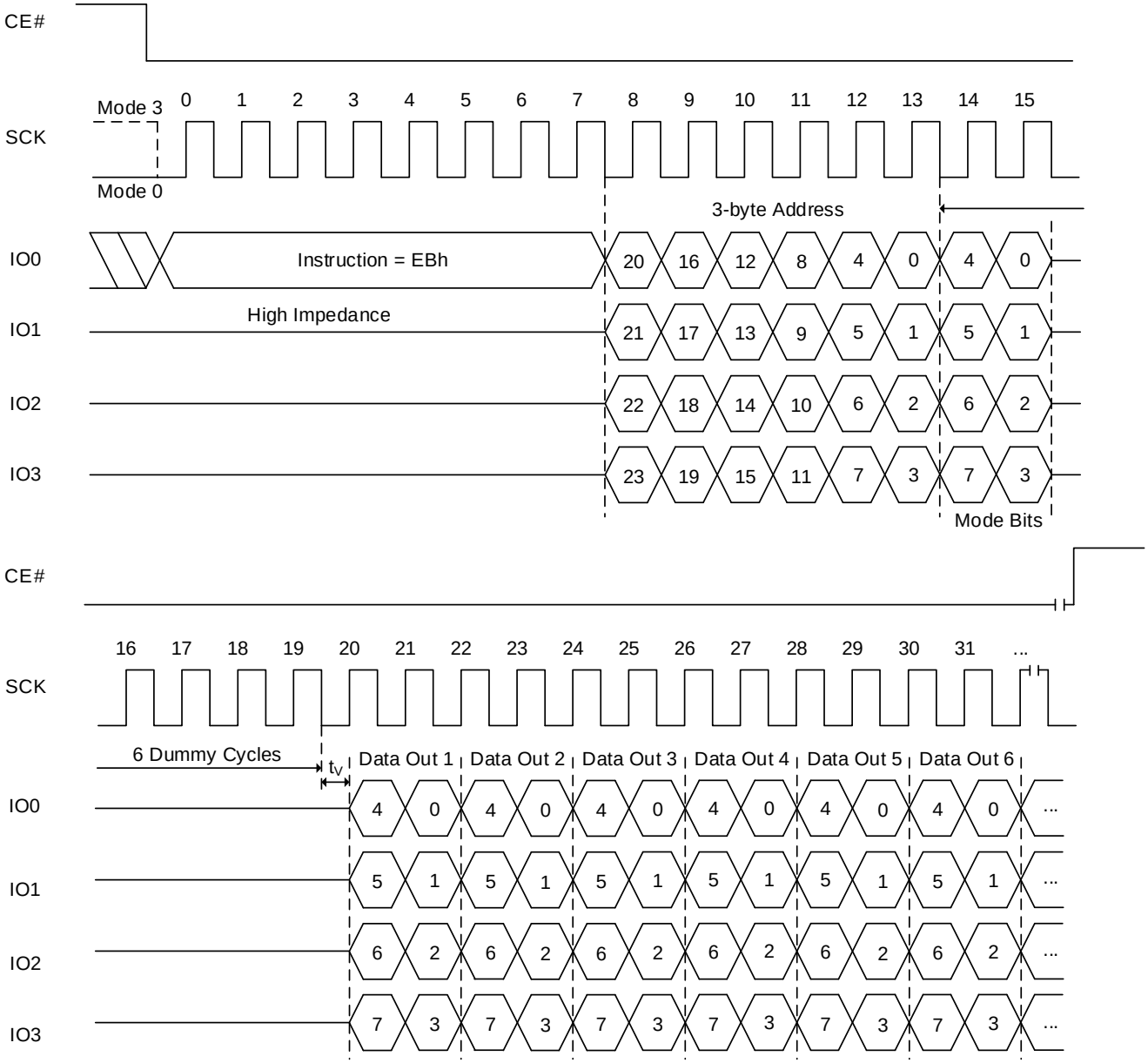
The first data byte addressed is shifted out on the IO3, IO2, IO1 and IO0 lines, with each group of four bits shifted out at a maximum frequency f_{CT}, during the falling edge of SCK. The first bit (MSB) is output on IO3, while simultaneously the second bit is output on IO2, the third bit is output on IO1, etc. Figures 8.15 and 8.16 illustrates the timing sequence.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRQIO/4FRQIO instruction. FRQIO/4FRQIO instruction is terminated by driving CE# high (V_{IH}).

The device supports the AX read operation by applying mode bits during dummy period. Mode bits consist of 8 bits, such as M7 to M0. Two cycles after address input are reserved for Mode bits in FRQIO/4FRQIO execution. M7 to M4 are important for enabling this mode. M3 to M0 become don't care for future use. When M[7:4]=1010(Ah), it enables the AX read operation and subsequent FRQIO/4FRQIO execution skips command code. It saves cycles as described in Figures 8.17 and 8.18. When the code is different from AXh (where X is don't care), the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command. SPI or QPI mode configuration retains the prior setting. Mode bit must be applied during dummy cycles. Number of dummy cycles in Table 6.11 includes number of mode bit cycles. If dummy cycles are configured as 6 cycles, data output will start right after mode bits and 4 additional dummy cycles are applied.

If the FRQIO/4FRQIO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle.

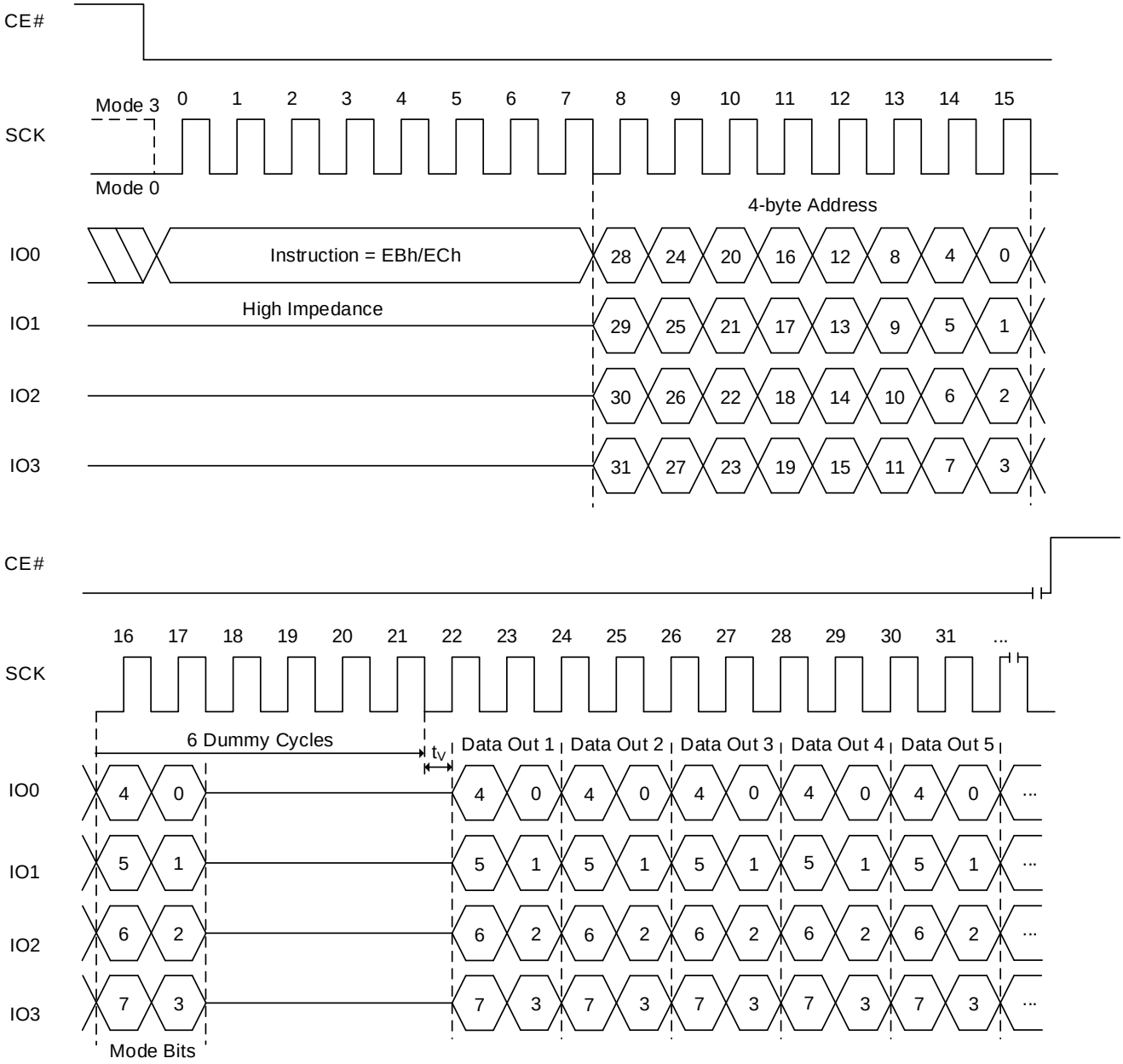
Figure 8.15 Fast Read Quad I/O Sequence (EBh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

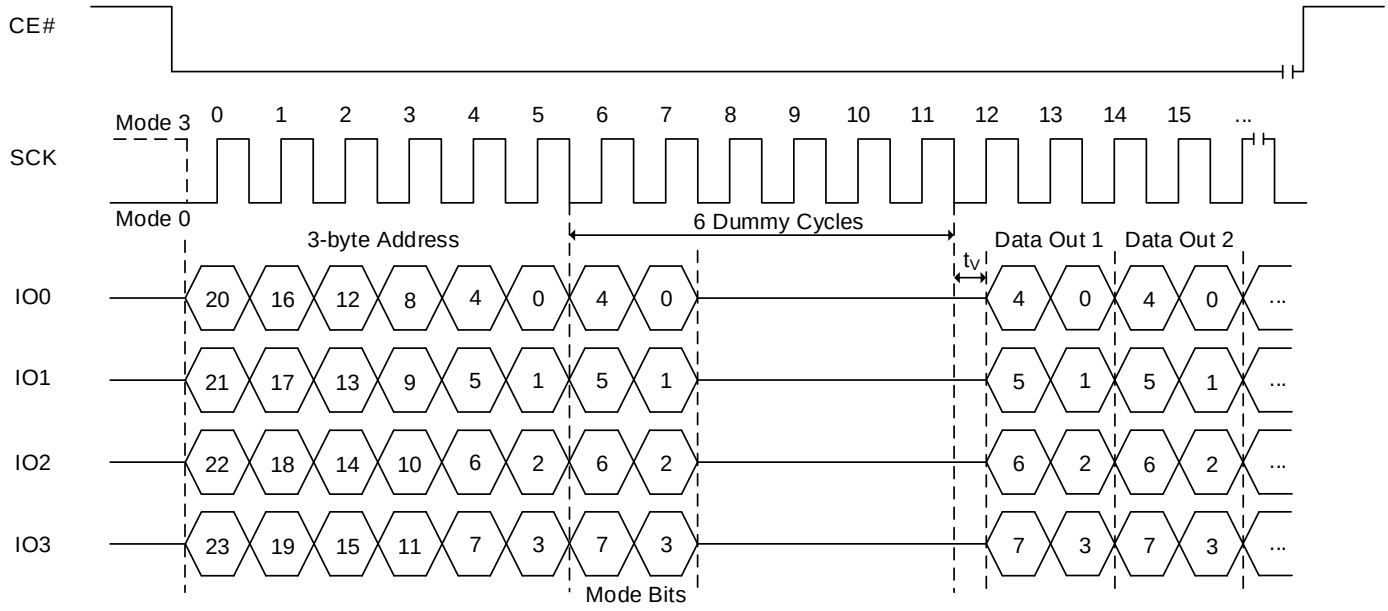
Figure 8.16 Fast Read Quad I/O Sequence (EBh [EXTADD=1] or ECh, 4-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

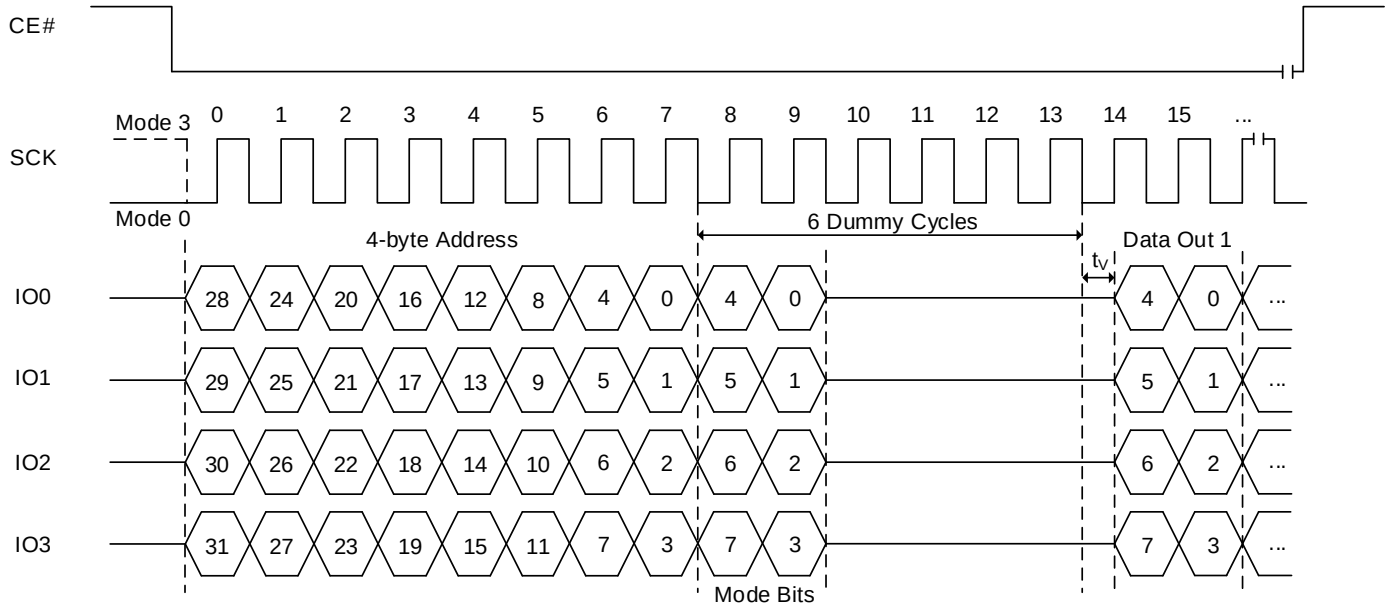
Figure 8.17 Fast Read Quad I/O AX Read Sequence (EBh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.18 Fast Read Quad I/O AX Read Sequence (EBh [EXTADD=1] or ECh, 4-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

FAST READ QUAD I/O OPERATION IN QPI MODE (FRQIO, EBh or 4FRQIO, ECh)

The FRQIO/4FRQIO instruction in QPI mode is used to read memory data at up to a 166MHz clock.

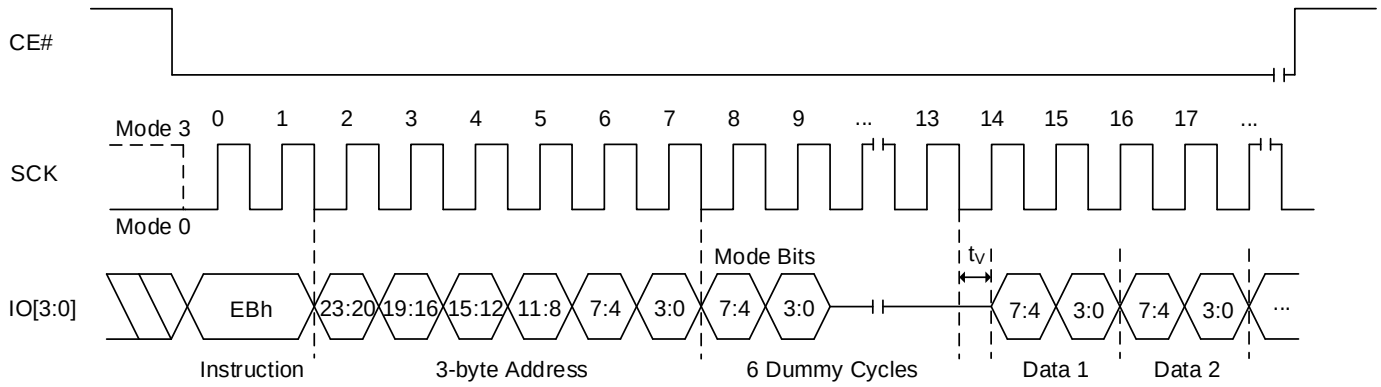
It is not required to set QE bit to "1".before Fast Read Quad I/O instruction in QPI mode.

The FRQIO/4FRQIO instruction in QPI mode utilizes all four IO lines to input the instruction code so that only two clocks are required, while the FRQIO/4FRQIO instruction requires that the byte-long instruction code is shifted into the device only via IO0 line in eight clocks. As a result, 6 command cycles will be reduced by the FRQIO/4FRQIO QPI instruction. In addition, subsequent address and data out are shifted in/out via all four IO lines like the FRQIO/4FRQIO instruction. In fact, except for the command cycle, the FRQIO/4FRQIO operation in QPI mode is exactly same as the FRQIO/4FRQIO operation in SPI mode.

The device supports the AX read operation by applying mode bits during dummy period. Mode bits consist of 8 bits, such as M7 to M0. Two cycles after address input are reserved for Mode bits in FRQIO/4FRQIO execution. M7 to M4 are important for enabling this mode. M3 to M0 become don't care for future use. When M[7:4]=1010(Ah), it enables the AX read operation and subsequent FRQIO/4FRQIO execution skips command code. It saves cycles as described in Figures 8.17 and 8.18. When the code is different from AXh (where X is don't care), the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command. SPI or QPI mode configuration retains the prior setting. Mode bit must be applied during dummy cycles. Number of dummy cycles in Table 6.11 includes number of mode bit cycles. If dummy cycles are configured as 6 cycles, data output will start right after mode bits and 4 additional dummy cycles are applied.

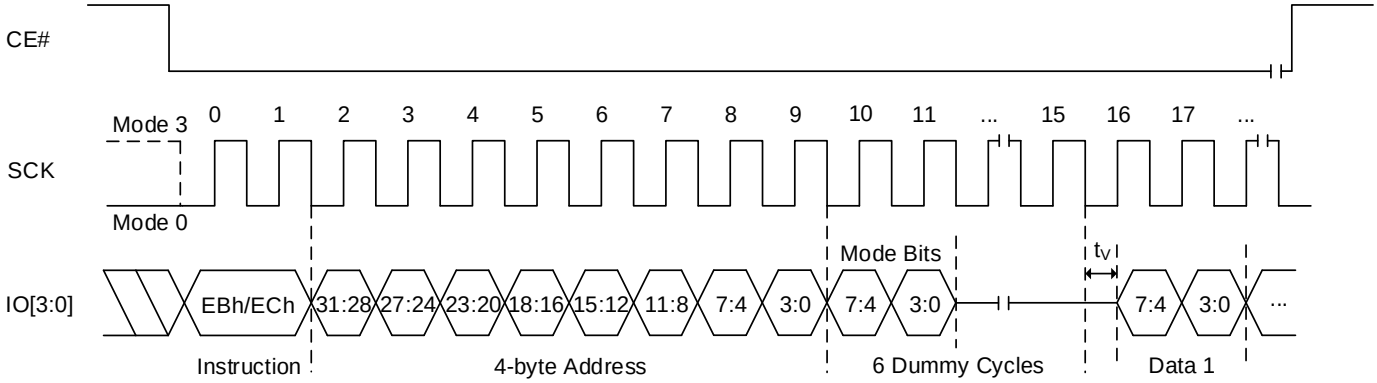
If the FRQIO/4FRQIO instruction in QPI mode is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle.

Figure 8.19 Fast Read Quad I/O Sequence In QPI Mode (EBh [EXTADD=0], 3-byte address)



Notes:
1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.**

Figure 8.20 Fast Read Quad I/O Sequence In QPI Mode (EBh [EXTADD=1] or ECh, 4-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.**

8.10 PAGE PROGRAM OPERATION (PP, 02h or 4PP, 12h)

The Page Program (PP/4PP) instruction allows up to 256 bytes data to be programmed into memory in a single operation. The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection bits (BP3, BP2, BP1, BP0) or ASP. A PP/4PP instruction which attempts to program into a page that is write-protected will be ignored. Before the execution of PP/4PP instruction, the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

- 02h (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 02h (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 12h is followed by a 4-byte address (A31-A0)

The PP/4PP instruction code, three or four address bytes as above and program data (1 to 256 bytes) are input via the SI line. Program operation will start immediately after the CE# is brought high, otherwise the PP/4PP instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. During a program operation, all instructions will be ignored except the RDSR instruction. The progress or completion of the program operation can be determined by reading the WIP bit. If the WIP bit is “1”, the program operation is still in progress. If WIP bit is “0”, the program operation has completed.

If more than 256 bytes data are sent to a device, the address counter rolls over within the same page, the previously latched data are discarded, and the last 256 bytes are kept to be programmed into the page. The starting byte can be anywhere within the page. When the end of the page is reached, the address will wrap around to the beginning of the same page. If the data to be programmed are less than a full page, the data of all other bytes on the same page will remain unchanged.

Note: A program operation can alter “1”s into “0”s, but an erase operation is required to change “0”s back to “1”s. A byte cannot be reprogrammed without first erasing the whole sector or block.

Figure 8.21 Page Program Sequence (02h [EXTADD=0], 3-byte address)

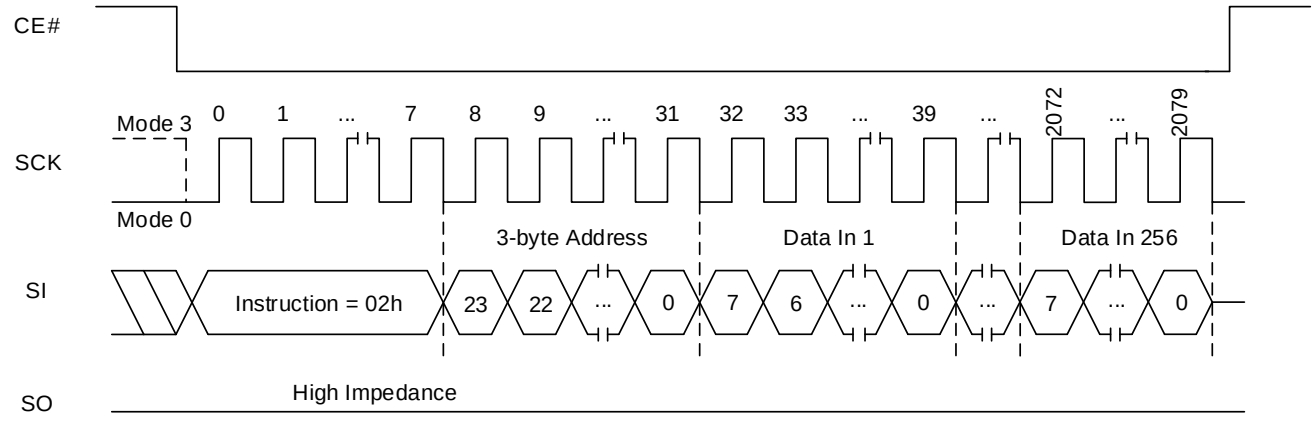


Figure 8.22 Page Program QPI Sequence (02h [EXTADD=0], 3-byte address)

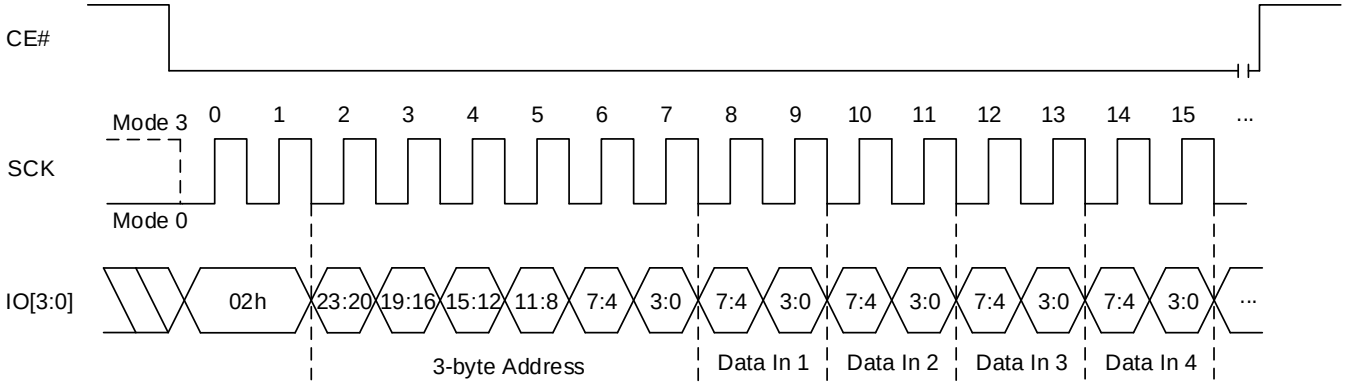


Figure 8.23 Page Program Sequence In SPI Mode (02h [EXTADD=1] or 12h, 4-byte address)

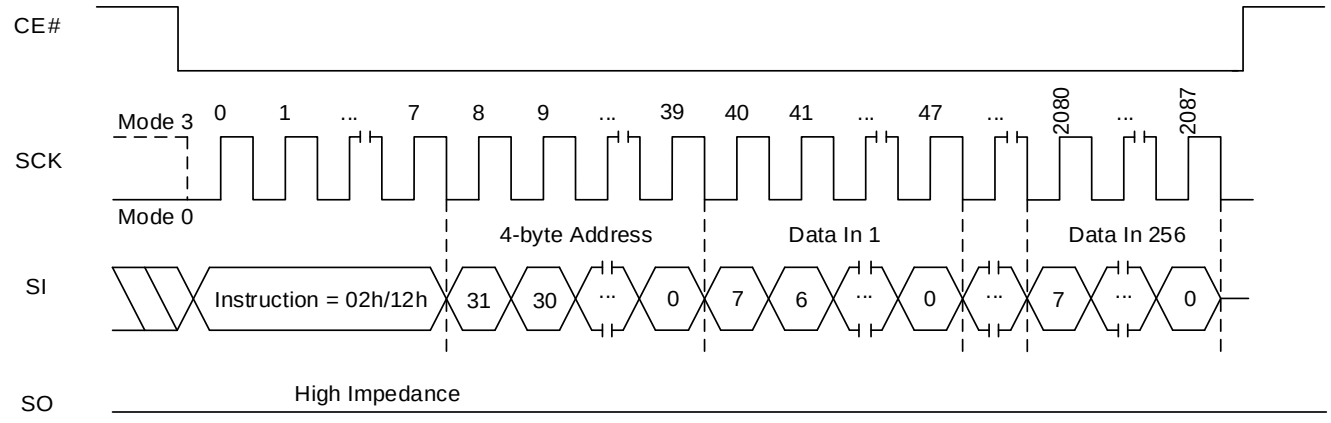
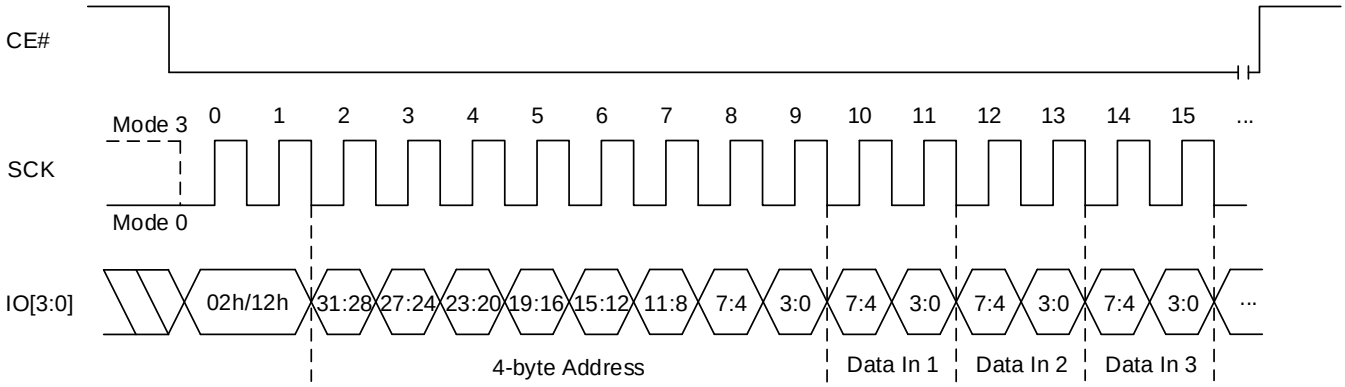


Figure 8.24 Page Program Sequence In QPI Mode (02h [EXTADD=1] or 12h, 4-byte address)



8.11 QUAD INPUT PAGE PROGRAM OPERATION (PPQ, 32h/38h or 4PPQ, 34h/3Eh)

The Quad Input Page Program instruction allows up to 256 bytes data to be programmed into memory in a single operation with four pins (IO0, IO1, IO2 and IO3). The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection (BP3, BP2, BP1, BP0) bits or ASP. A Quad Input Page Program instruction which attempts to program into a page that is write-protected will be ignored.

Before the execution of Quad Input Page Program instruction, the QE bit in the Status Register must be set to "1" and the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

- 32h/38h (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 32h/38h (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 34h/3Eh is followed by a 4-byte address (A31-A0)

Program operation will start immediately after the CE# is brought high, otherwise the Quad Input Page Program instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. During a program operation, all instructions will be ignored except the RDSR instruction. The progress or completion of the program operation can be determined by reading the WIP bit. If the WIP bit is "1", the program operation is still in progress. If WIP bit is "0", the program operation has completed.

If more than 256 bytes data are sent to a device, the address counter rolls over within the same page, the previously latched data are discarded, and the last 256 bytes data are kept to be programmed into the page. The starting byte can be anywhere within the page. When the end of the page is reached, the address will wrap around to the beginning of the same page. If the data to be programmed are less than a full page, the data of all other bytes on the same page will remain unchanged.

Note: A program operation can alter "1"s into "0"s, but an erase operation is required to change "0"s back to "1"s. A byte cannot be reprogrammed without first erasing the whole sector or block.

Figure 8.25 Quad Input Page Program operation (32h/38h [EXTADD=0], 3-byte address)

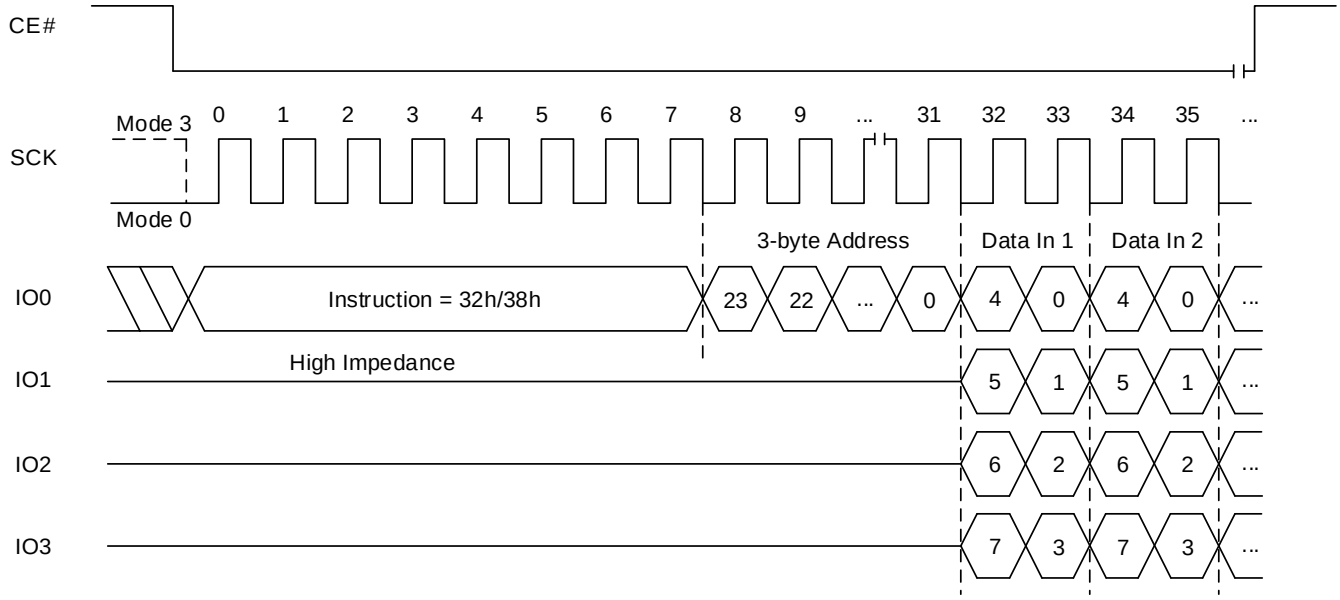
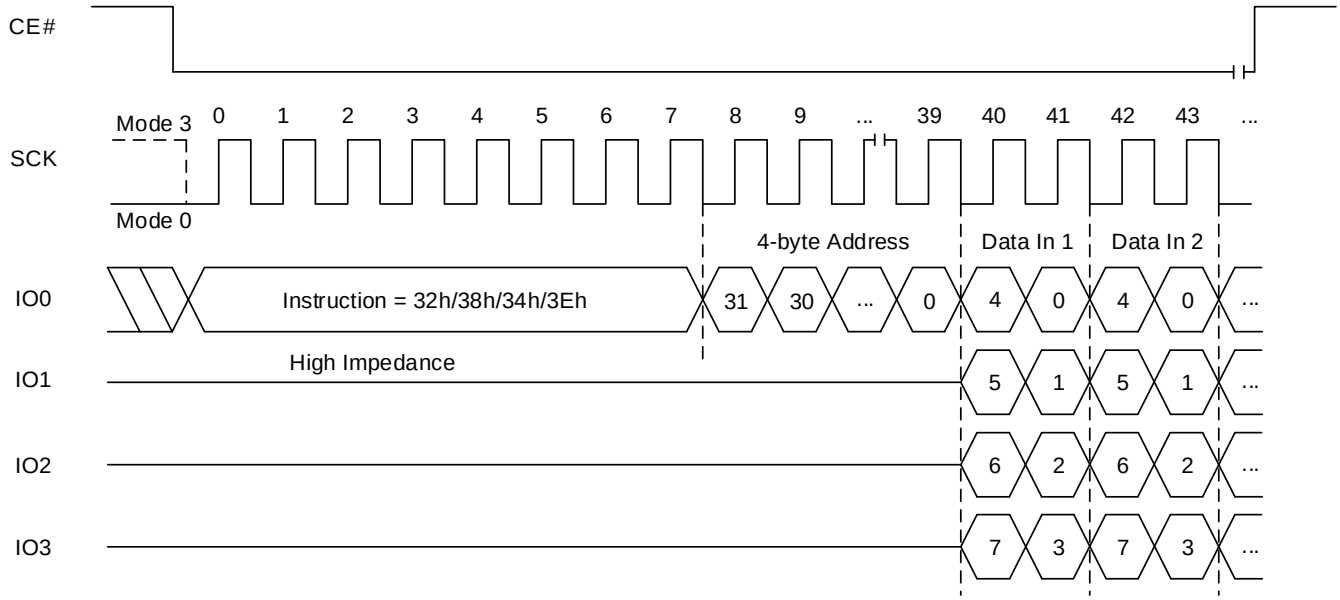


Figure 8.26 Quad Input Page Program operation (32h/38h [EXTADD=1] or 34h/3Eh, 4-byte address)



8.12 ERASE OPERATION

The memory array of the device is organized into uniform 4 Kbyte sectors or 32/64 Kbyte uniform blocks (a block consists of eight/sixteen adjacent sectors respectively).

Before a byte is reprogrammed, the sector or block that contains the byte must be erased (erasing sets bits to “1”). In order to erase the device, there are three erase instructions available: Sector Erase (SER), Block Erase (BER), and Chip Erase (CER). A sector erase operation allows any individual sector to be erased without affecting the data in other sectors. A block erase operation erases any individual block. A chip erase operation erases the whole memory array of a device. A sector erase, block erase, or chip erase operation can be executed prior to any programming operation.

8.13 SECTOR ERASE OPERATION (SER, D7h/20h or 4SER, 21h)

A Sector Erase (SER/4SER) instruction erases a 4 Kbyte sector before the execution of a SER/4SER instruction, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL bit is automatically reset after the completion of Sector Erase operation.

- D7h/20h (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- D7h/20h (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 21h is followed by a 4-byte address (A31-A0)

A SER/4SER instruction is entered, after CE# is pulled low to select the device and stays low during the entire instruction sequence The SER/4SER instruction code, and three or four address bytes as above are input via SI. Erase operation will start immediately after CE# is pulled high. The internal control logic automatically handles the erase voltage and timing.

During an erase operation, all instruction will be ignored except Read Status Register, Read Extended Read Register, Erase/Program Suspend, and Software/Hardware Reset instructions. The progress or completion of the erase operation can be determined by reading the WIP bit. If the WIP bit is “1”, the erase operation is still in progress. If the WIP bit is “0”, the erase operation has been completed.

Figure 8.27 Sector Erase Sequence (D7h/20h [EXTADD=0], 3-byte address)

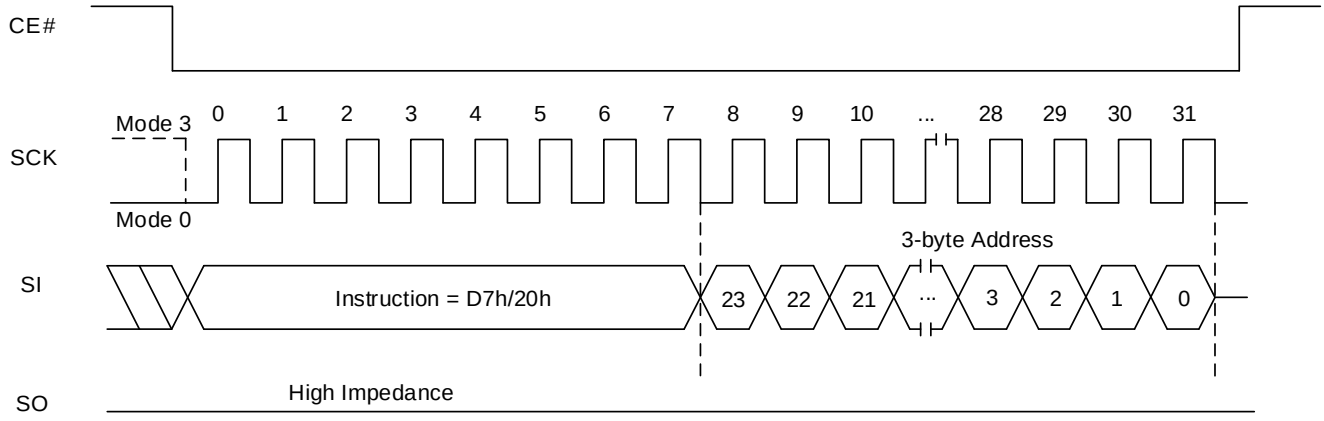


Figure 8.28 Sector Erase Sequence (D7h/20h [EXTADD=1] or 21h, 4-byte address)

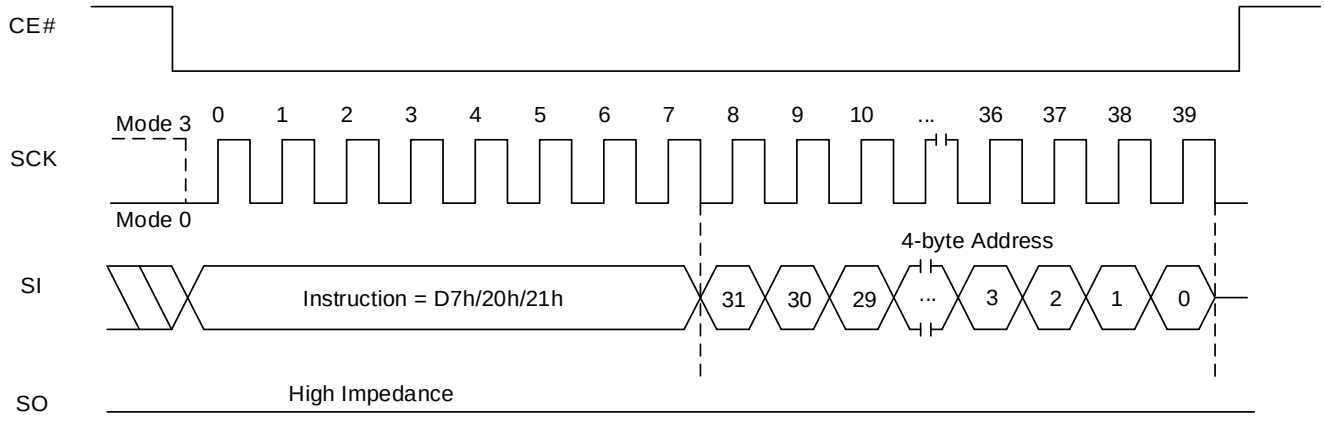


Figure 8.29 Sector Erase Sequence In QPI Mode (D7h/20h [EXTADD=0], 3-byte address)

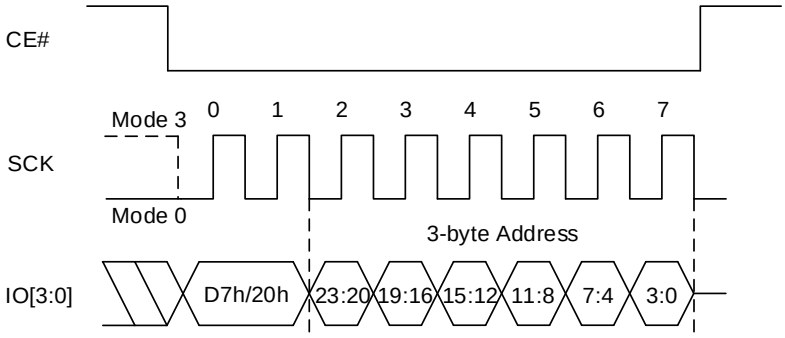
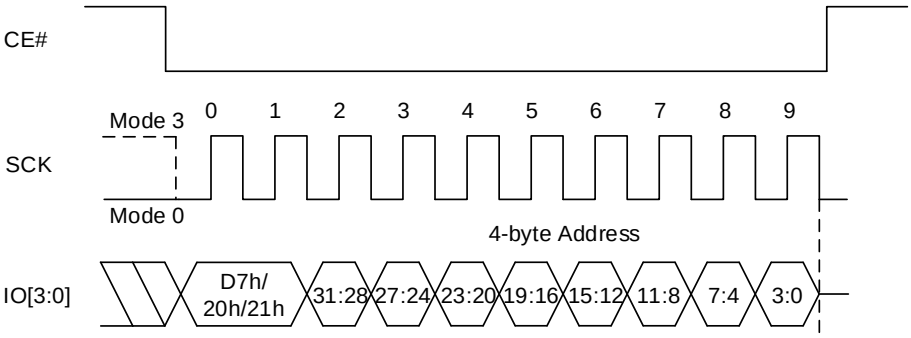


Figure 8.30 Sector Erase Sequence In QPI Mode (D7h/20h [EXTADD=1] or 21h, 4-byte address)



8.14 BLOCK ERASE OPERATION (BER32K:52h or 4BER32K:5Ch, BER64K:D8h or 4BER64K:DCh)

A Block Erase (BER) instruction erases a 32/64 Kbyte block. Before the execution of a BER instruction, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL is reset automatically after the completion of a block erase operation.

- 52h/D8h (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 52h/D8h (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 5Ch/DCh is followed by a 4-byte address (A31-A0)

The BER instruction code and three or four address bytes as above are input via SI. Erase operation will start immediately after the CE# is pulled high, otherwise the BER instruction will not be executed. The internal control logic automatically handles the erase voltage and timing.

Figure 8.31 Block Erase (64K) Sequence (D8h [EXTADD=0], 3-byte address)

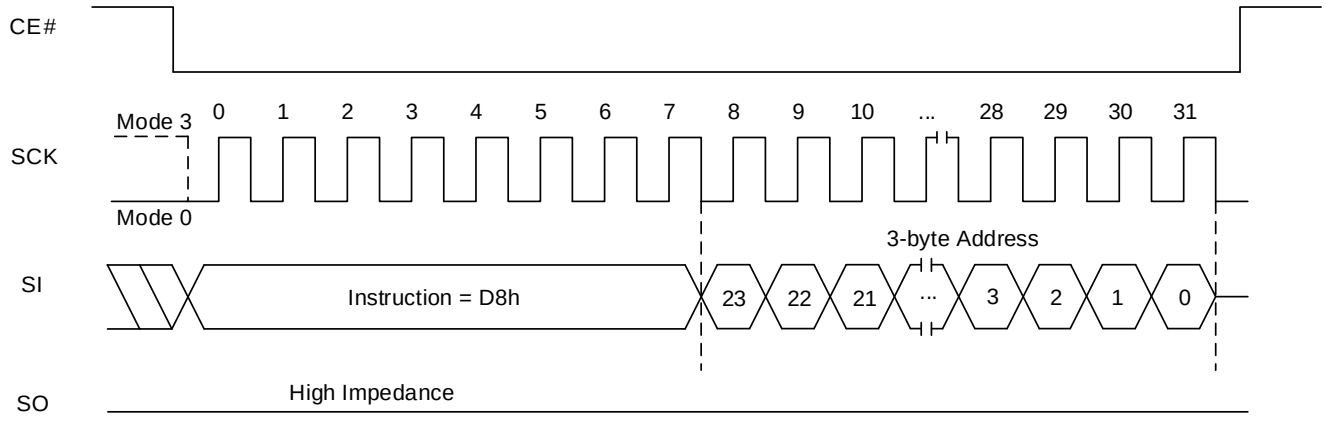


Figure 8.32 Block Erase (64K) Sequence (D8h [EXTADD=1] or DCh, 4-byte address)

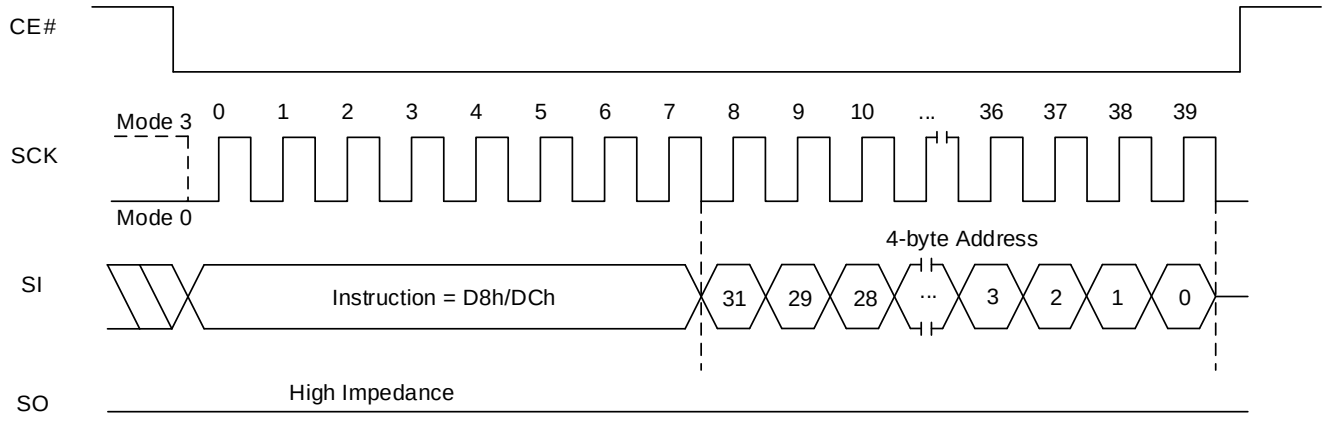


Figure 8.33 Block Erase (64K) Sequence In QPI Mode (D8h [EXTADD=0], 3-byte address)

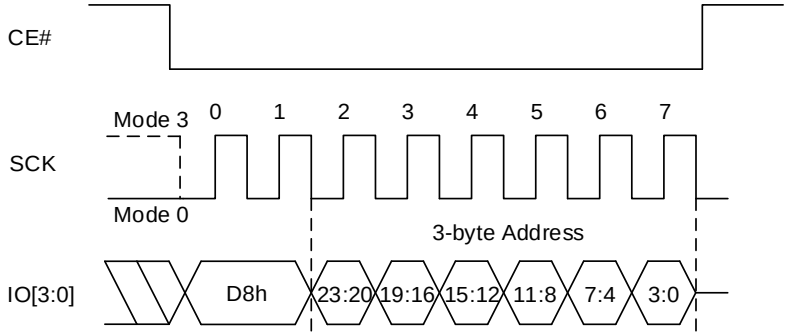


Figure 8.34 Block Erase (64K) Sequence In QPI Mode (D8h [EXTADD=1] or DCh, 4-byte address)

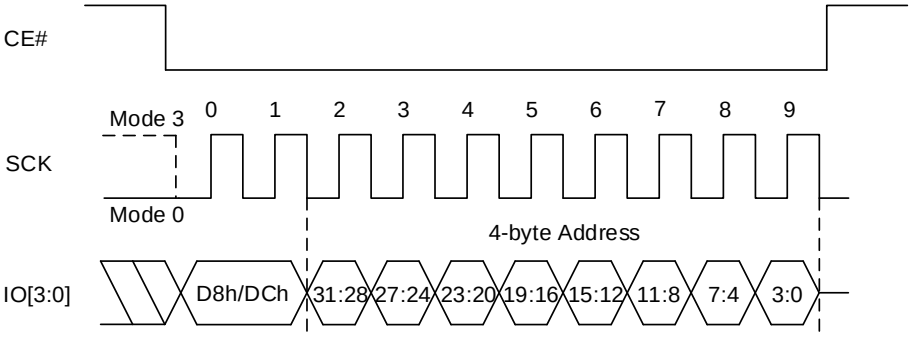


Figure 8.35 Block Erase (32K) Sequence (52h [EXTADD=0], 3-byte address)

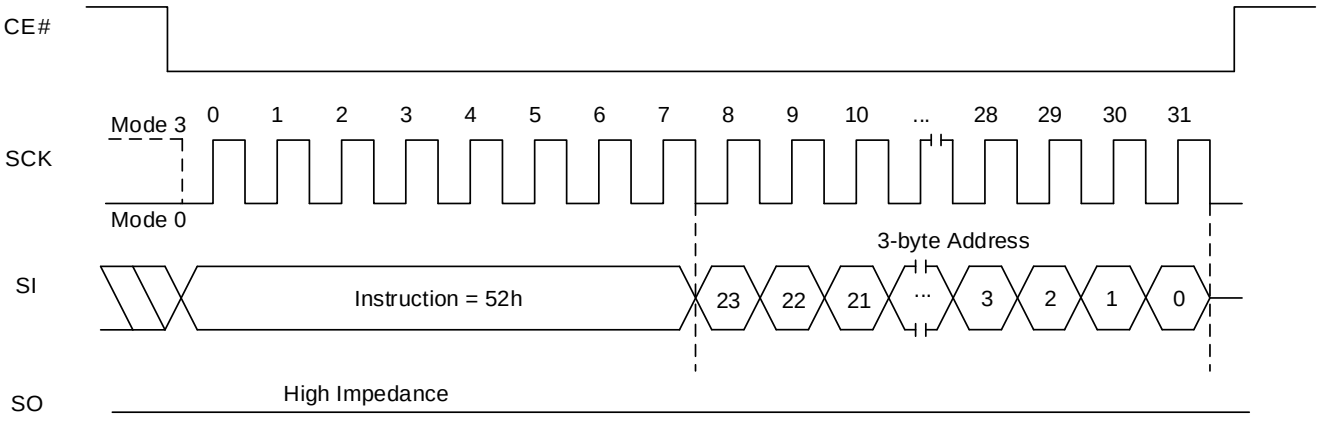


Figure 8.36 Block Erase (32K) Sequence (52h [EXTADD=1] or 5Ch, 4-byte address)

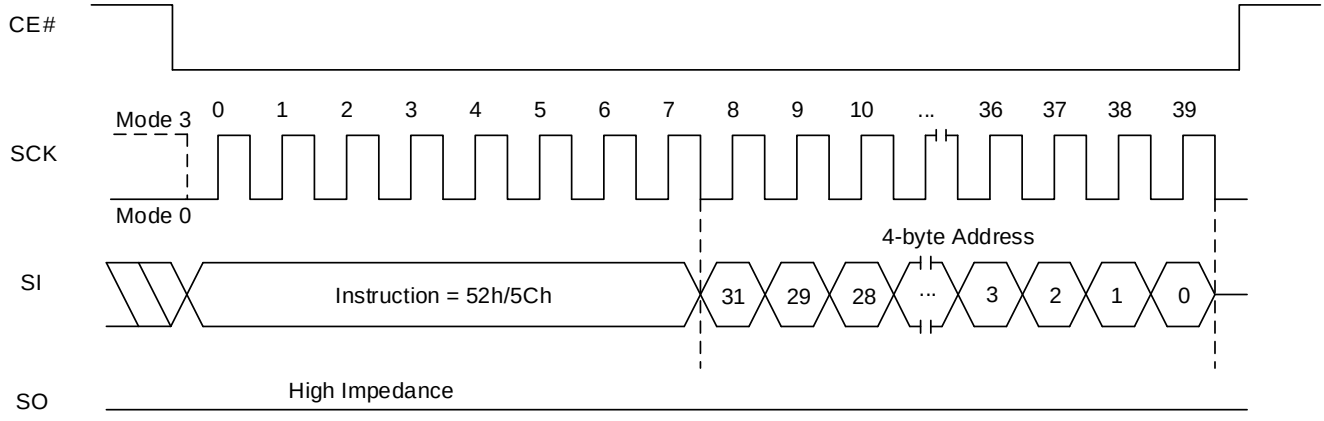


Figure 8.37 Block Erase (32K) Sequence In QPI Mode (52h [EXTADD=0], 3-byte address)

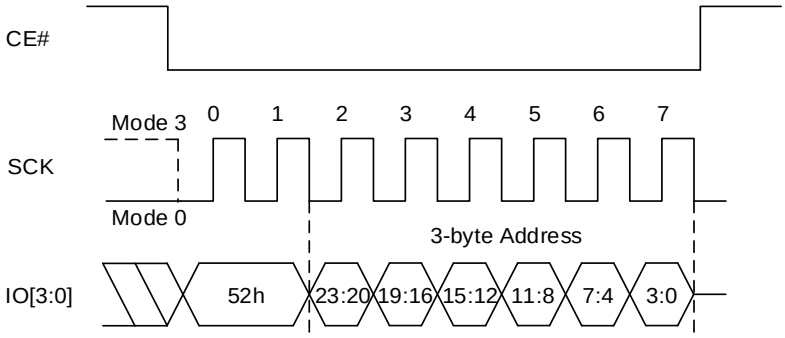
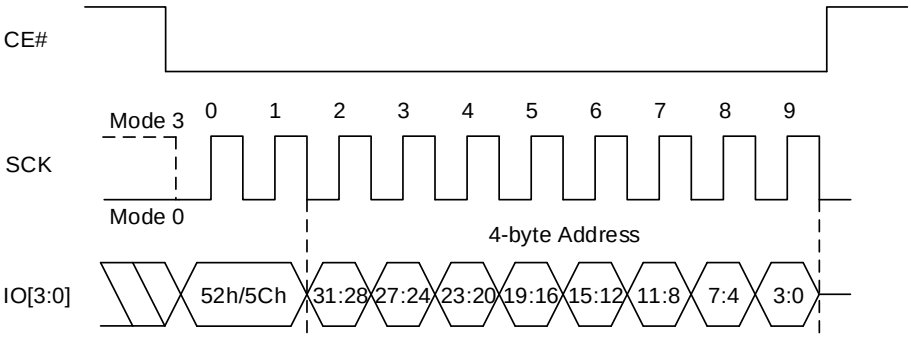


Figure 8.38 Block Erase (32K) Sequence In QPI Mode (52h [EXTADD=1] or 5Ch, 4-byte address)



8.15 CHIP ERASE OPERATION (CER, C7h/60h)

A Chip Erase (CER) instruction erases the entire memory array. Before the execution of CER instruction, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL is automatically reset after completion of a chip erase operation.

The CER instruction code is input via the SI. Erase operation will start immediately after CE# is pulled high, otherwise the CER instruction will not be executed. The internal control logic automatically handles the erase voltage and timing.

Chip Erase (CER) instruction can be executed only when Block Protection (BP3~BP0) bits are set to 0s. If the BP bits are not 0, the CER command is not executed and E_ERR and PROT_E are set.

Chip Erase (CER) instruction will skip sectors/blocks protected by ASP (DYB bits or PPB bits) and will **not** set E_ERR and PROT_E if sectors/blocks are protected by ASP only.

Figure 8.39 Chip Erase Sequence In SPI Mode

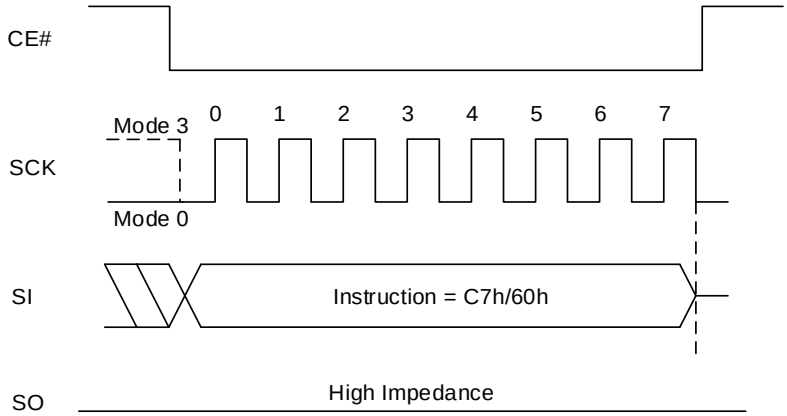
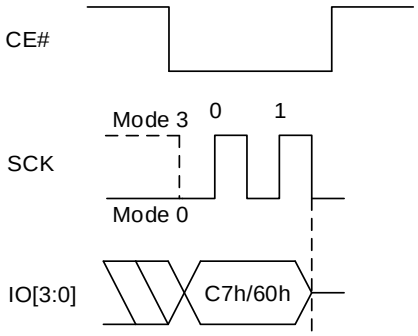


Figure 8.40 Chip Erase Sequence In QPI Mode



8.16 WRITE ENABLE OPERATION (WREN, 06h)

The Write Enable (WREN) instruction is used to set the Write Enable Latch (WEL) bit. The WEL bit is reset to the write-protected state after power-up. The WEL bit must be write enabled before any write operation, including Sector Erase, Block Erase, Chip Erase, Page Program, Program Information Row, Write Status Register, Write Function Register, Set Non-Volatile Read Register, Set Non-Volatile Extended Read Register, and Write Autoboot Register operations except for Set Volatile Read Register and Set Volatile Extended Read Register. The WEL bit will be reset to the write-protected state automatically upon completion of a write operation. The WREN instruction is required before any above operation is executed.

Figure 8.41 Write Enable Sequence In SPI Mode

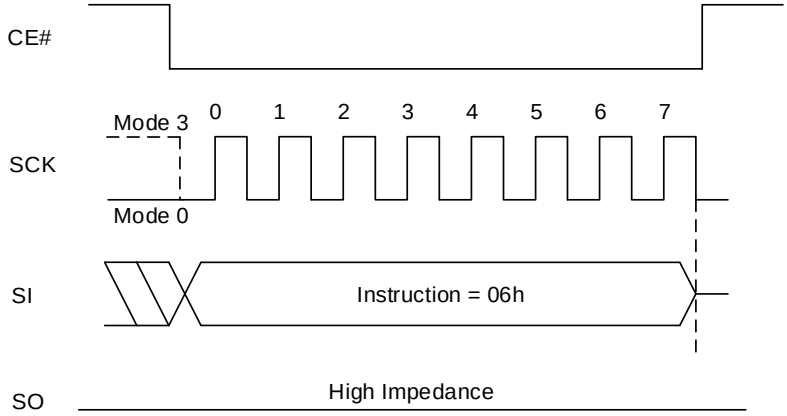
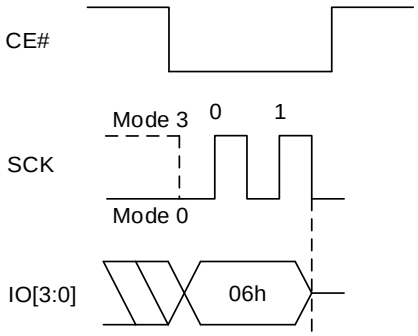


Figure 8.42 Write Enable Sequence In QPI Mode



8.17 WRITE DISABLE OPERATION (WRDI, 04h)

The Write Disable (WRDI) instruction resets the WEL bit and disables all write instructions. The WRDI instruction is not required after the execution of a write instruction, since the WEL bit is automatically reset.

Figure 8.43 Write Disable Sequence In SPI Mode

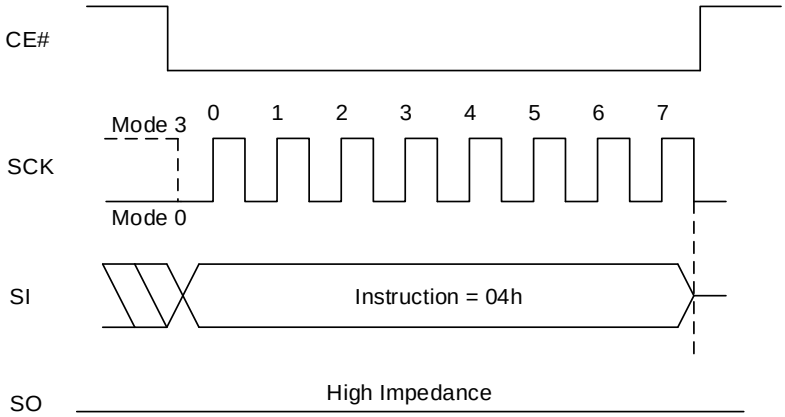
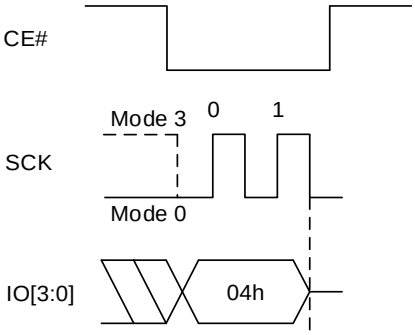


Figure 8.44 Write Disable Sequence In QPI Mode



8.18 READ STATUS REGISTER OPERATION (RDSR, 05h)

The Read Status Register (RDSR) instruction provides access to the Status Register. During the execution of a program, erase or Write Status Register operation, the RDSR instruction will be executed, which can be used to check the progress or completion of an operation by reading the WIP bit.

Figure 8.45 Read Status Register Sequence In SPI Mode

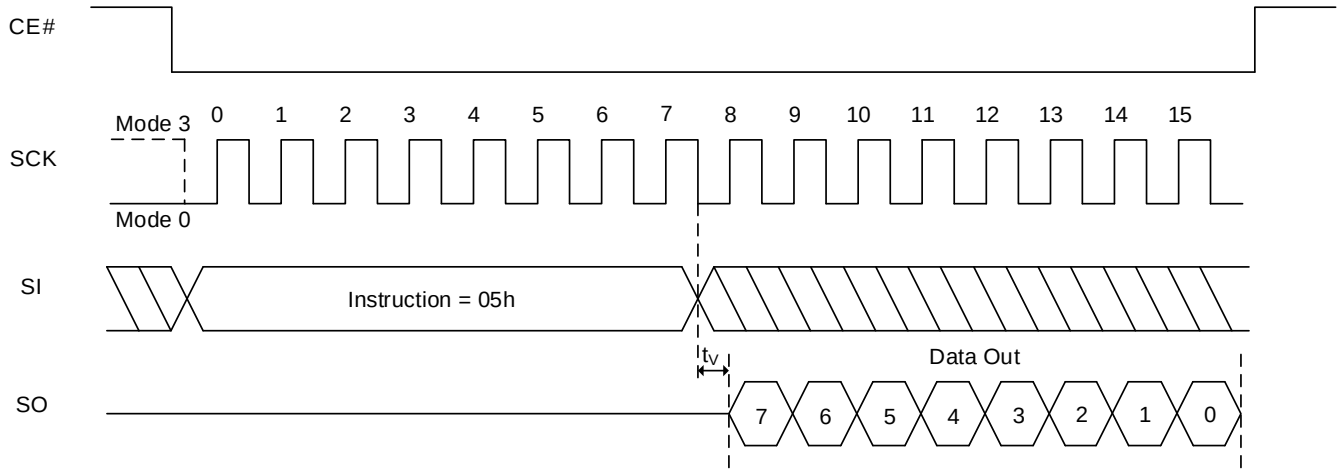
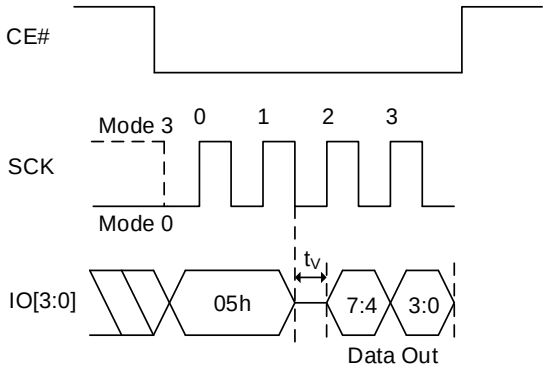


Figure 8.46 Read Status Register Sequence In QPI Mode



8.19 WRITE STATUS REGISTER OPERATION (WRSR, 01h)

The Write Status Register (WRSR) instruction allows the user to enable or disable the block protection and Status Register write protection features by writing “0”s or “1”s into the non-volatile BP3, BP2, BP1, BP0, and SRWD bits. Also WRSR instruction allows the user to disable or enable quad operation by writing “0” or “1” into the non-volatile QE bit.

Figure 8.47 Write Status Register Sequence In SPI Mode

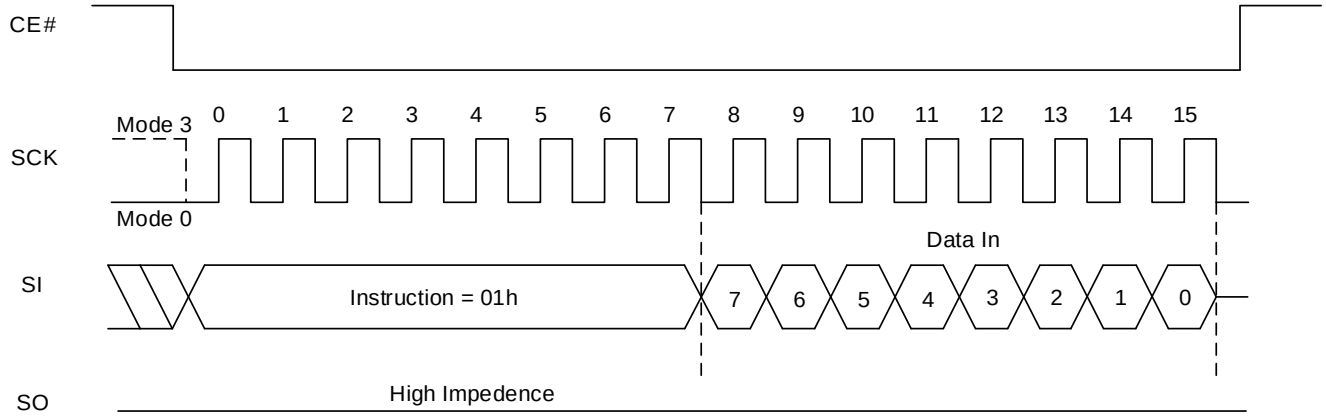
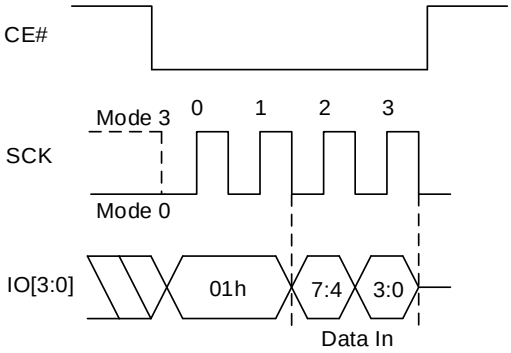


Figure 8.48 Write Status Register Sequence In QPI Mode



8.20 READ FUNCTION REGISTER OPERATION (RDFR, 48h)

The Read Function Register (RDFR) instruction provides access to the Function Register. Refer to Table 6.6 Function Register Bit Definition for more detail.

Figure 8.49 Read Function Register Sequence In SPI Mode

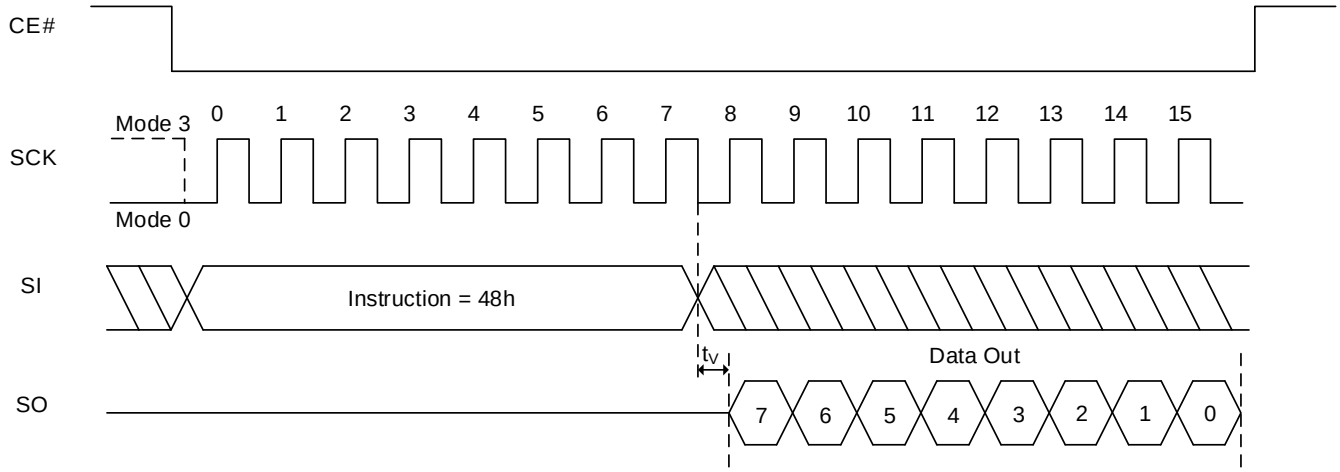
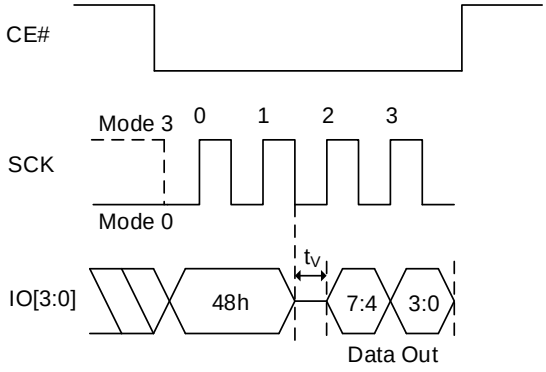


Figure 8.50 Read Function Register Sequence In QPI Mode



8.21 WRITE FUNCTION REGISTER OPERATION (WRFR, 42h)

The Write Function Register (WRFR) instruction allows the user to disable dedicated RESET# pin or ball on 16-pin SOIC or 24 ball TFBGA by setting Dedicated RESET# Disable bit to "1". Also Information Row Lock bits (IRL3~IRL0) can be set to "1" individually by WRFR instruction in order to lock Information Row.
Since Dedicated RESET# Disable bit and IRL bits are OTP, once they are set to "1", they cannot be set back to "0" again

Figure 8.51 Write Function Register Sequence In SPI Mode

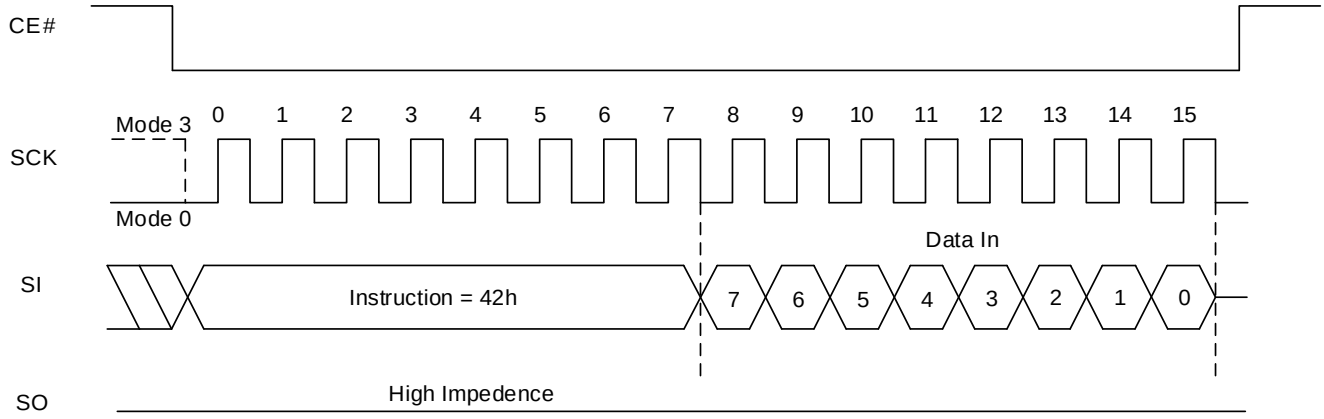
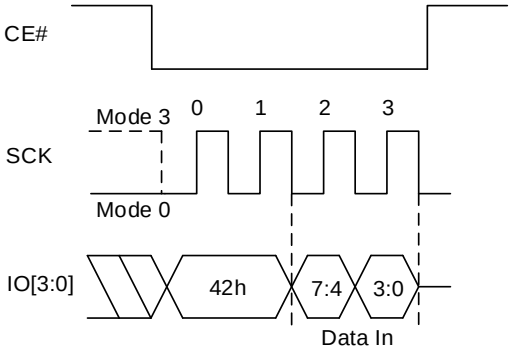


Figure 8.52 Write Function Register Sequence In QPI Mode



8.22 ENTER QUAD PERIPHERAL INTERFACE (QPI) MODE OPERATION (QIOEN,35h; QIODI,F5h)

The Enter Quad I/O (QIOEN) instruction, 35h, enables the Flash device for QPI bus operation. Upon completion of the instruction, all instructions thereafter will be 4-bit multiplexed input/output until a power cycle or an Exit Quad I/O instruction is sent to device.

The Exit Quad I/O instruction, F5h, resets the device to 1-bit SPI protocol operation. To execute an Exit Quad I/O operation, the host drives CE# low, sends the Exit Quad I/O command cycle, then drives CE# high. The device just accepts QPI (2 clocks) command cycles.

Figure 8.53 Enter Quad Peripheral Interface (QPI) Mode Sequence

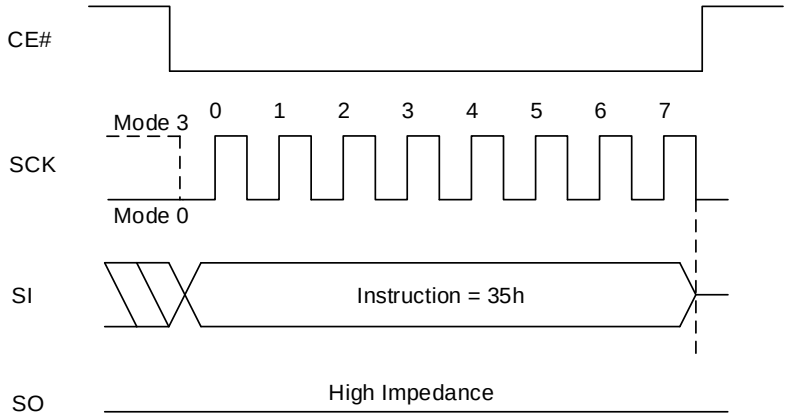
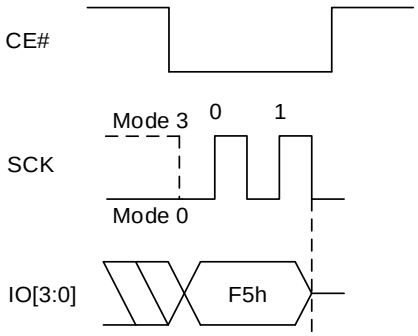


Figure 8.54 Exit Quad Peripheral Interface (QPI) Mode Sequence



8.23 PROGRAM/ERASE SUSPEND & RESUME

The device allows the interruption of Sector Erase, Block Erase, or Page Program operations to conduct other operations. 75h/B0h command for suspend and 7Ah/30h for resume will be used. (SPI/QPI all acceptable) Function Register bit2 (PSUS) and bit3 (ESUS) are used to check whether or not the device is in suspend mode.

Suspend to read ready timing: 100µs
Resume to another suspend timing: 400µs

PROGRAM/ERASE SUSPEND DURING SECTOR-ERASE OR BLOCK-ERASE (PERSUS 75h/B0h)
The Program/Erase Suspend allows the interruption of Sector Erase and Block Erase operations. After the Program/Erase Suspend, program, read related, resume and reset commands can be accepted. It is possible to nest a Program/Erase Suspend operation during a Program inside a Program/Erase Suspend operation. Refer to Table 8.7 for more detail.

To execute the Program/Erase Suspend operation, the host drives CE# low, sends the Program/Erase Suspend command cycle (75h/B0h), then drives CE# high. The Function Register indicates that the Erase has been suspended by changing the ESUS 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 or wait the specified time t_{SUS}. When ESUS bit is issued, the Write Enable Latch (WEL) bit will be reset.

PROGRAM/ERASE SUSPEND DURING PAGE PROGRAMMING (PERSUS 75h/B0h)
The Program/Erase Suspend allows the interruption of all array Program operations. After the Program/Erase Suspend command, WEL bit will be disabled, therefore only read related, resume and reset commands can be accepted. Refer to Table 8.7 for more detail.

To execute the Program/Erase Suspend operation, the host drives CE# low, sends the Program/Erase Suspend command cycle (75h/B0h), then drives CE# high. The Function Register indicates that the programming has been suspended by changing the PSUS 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 or wait the specified time t_{SUS}.

PROGRAM/ERASE RESUME (PERRSM 7Ah/30h)
The Program/Erase Resume restarts the Program or Erase command that was suspended, and changes the suspend status bit in the Function Register (ESUS or PSUS bits) back to "0". To execute the Program/Erase Resume operation, the host drives CE# low, sends the Program/Erase Resume command cycle (7Ah/30h), then drives CE# 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, 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}.

Table 8.7 Instructions accepted during Suspend

Operation Suspended	Instruction Allowed		
	Name	Hex Code	Operation
Program or Erase	NORD	03h	Normal Read Mode
Program or Erase	4NORD	13h	4-byte Address Normal Read Mode
Program or Erase	FRD	0Bh	Fast Read Mode
Program or Erase	4FRD	0Ch	4-byte Address Fast Read Mode
Program or Erase	FRDIO	BBh	Fast Read Dual I/O
Program or Erase	4FRDIO	BCh	4-byte Address Fast Read Dual I/O
Program or Erase	FRDO	3Bh	Fast Read Dual Output
Program or Erase	4FRDO	3Ch	4-byte Address Fast Read Dual Output
Program or Erase	FRQIO	EBh	Fast Read Quad I/O
Program or Erase	4FRQIO	ECh	4-byte Address Fast Read Quad I/O
Program or Erase	FRQO	6Bh	Fast Read Quad Output
Program or Erase	4FRQO	6Ch	4-byte Address Fast Read Quad Output
Program or Erase	FRDTR	0Dh	Fast Read DTR Mode
Program or Erase	4FRDTR	0Eh	4-byte Address Fast Read DTR Mode
Program or Erase	FRDDTR	BDh	Fast Read Dual I/O DTR
Program or Erase	4FRDDTR	BEh	4-byte Address Fast Read Dual I/O DTR
Program or Erase	FRQDTR	EDh	Fast Read Quad I/O DTR
Program or Erase	4FRQDTR	EEh	4-byte Address Fast Read Quad I/O DTR
Erase	PP	02h	Serial Input Page Program
Erase	4PP	12h	4-byte Address Serial Input Page Program
Erase	PPQ	32h/38h	Quad Input Page Program
Erase	4PPQ	34h/3Eh	4-byte Address Quad Input Page Program
Erase	WREN	06h	Write Enable
Program or Erase	RDSR	05h	Read Status Register
Program or Erase	RDFR	48h	Read Function Register
Program or Erase	CLERP	82h	Clear Extended Read Register
Program or Erase	PERRSM	7Ah/30h	Program/Erase Resume
Erase	PERSUS	75h/B0h	Program/Erase Suspend
Program or Erase	RDID	ABh	Read Manufacturer and Product ID
Program or Erase	SRPV	C0/63h	Set Read Parameters (Volatile)
Program or Erase	SERPV	83h	Set Extended Read Parameters (Volatile)
Program or Erase	RDRP	61h	Read Read Parameters (Volatile)
Program or Erase	RDERP	81h	Read Extended Read Parameters (Volatile)
Program or Erase	RDJDID	9Fh	Read Manufacturer and Product ID by JEDEC ID Command
Program or Erase	RDMDID	90h	Read Manufacturer and Device ID
Program or Erase	RDJDIDQ	AFh	Read JEDEC ID QPI mode
Program or Erase	RDUID	4Bh	Read Unique ID Number
Program or Erase	RDSFDP	5Ah	SFDP Read

Operation Suspended	Instruction Allowed		
	Name	**Hex Code**	**Operation**
Program or Erase	NOP	00h	No Operation
Program or Erase	RSTEN	66h	Software reset enable
Program or Erase	RST	99h	Reset (Only along with RSTEN 66h)
Program or Erase	IRRD	78h	Read Information Row
Erase	SECUNLOCK	26h	Sector Unlock
Erase	4SECUNLOCK	25h	4-byte Address Sector Unlock
Erase	SECLOCK	24h	Sector Lock
Program or Erase	RDABR	14h	Read AutoBoot Register
Program or Erase	RDBR	16h/C8h	Read Bank Address Register
Program or Erase	WRBRV	17h/C5h	Write Volatile Bank Address Register
Program or Erase	EN4B	B7h	Enter 4-byte Address Mode
Program or Erase	EX4B	29h	Exit 4-byte Address Mode
Erase	RDDYB	FAh	Read DYB
Erase	4RDDYB	E0h	4-byte Address Read DYB
Erase	WRDYB	FBh	Write DYB
Erase	4WRDYB	E1h	4-byte Address Write DYB
Erase	RDPPB	FCh	Read PPB
Erase	4RDPPB	E2h	4-byte Address Read PPB

8.24 ENTER DEEP POWER DOWN (DP, B9h)

The Deep Power-down (DP) instruction is for setting the device on the minimizing the power consumption (enter into Power-down mode). During this mode, standby current is reduced from I_{sb1} to I_{sb2}. While in the Power-down mode, the device is not active and all Write/Program/Erase instructions are ignored. The instruction is initiated by driving the CE# pin low and shifting the instruction code into the device. The CE# pin must be driven high after the instruction has been latched, or Power-down mode will not engage. Once CE# pin driven high, the Power-down mode will be entered within the time duration of t_{DP}. While in the Power-down mode only the Release from Power-down/RDID instruction, which restores the device to normal operation, will be recognized. All other instructions are ignored, including the Read Status Register instruction which is always available during normal operation. Ignoring all but one instruction makes the Power Down state a useful condition for securing maximum write protection. It is available in both SPI and QPI mode.

Figure 8.55 Enter Deep Power Down Mode Sequence

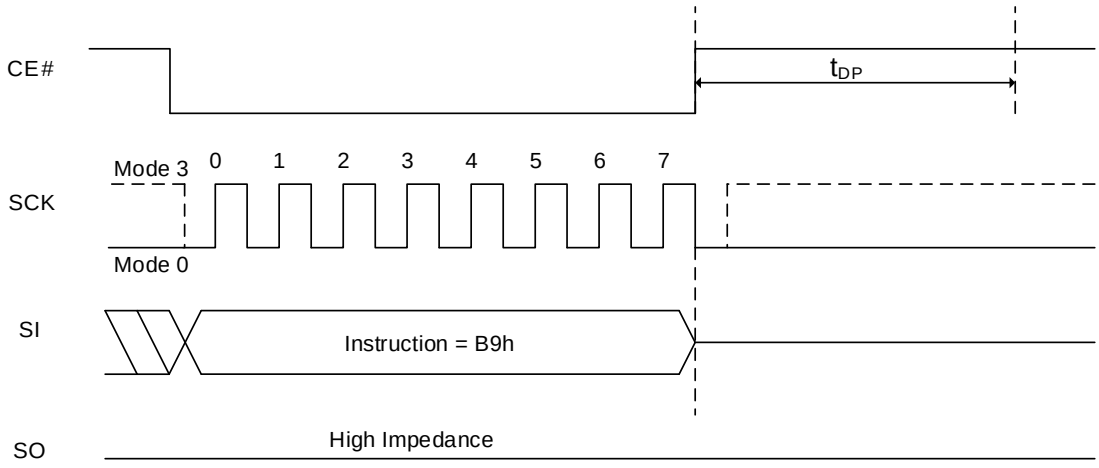
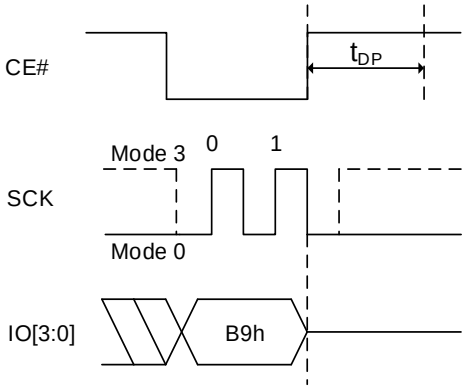


Figure 8.56 Enter Deep Power Down Mode QPI Sequence



8.25 RELEASE DEEP POWER DOWN (RDPD, ABh)

The Release Deep Power-down/Read Device ID instruction is a multi-purpose command. To release the device from the deep power-down mode, the instruction is issued by driving the CE# pin low, shifting the instruction code into the device and driving CE# high.

Releasing the device from Power-down mode will take the time duration of t_{RES1} before normal operation is restored and other instructions are accepted. The CE# pin must remain high during the t_{RES1} time duration. If the Release Deep Power-down/RDID instruction is issued while an Erase, Program or Write cycle is in progress (WIP=1) the instruction is ignored and will not have any effects on the current cycle.

Figure 8.57 Release Power Down Mode Sequence In SPI Mode

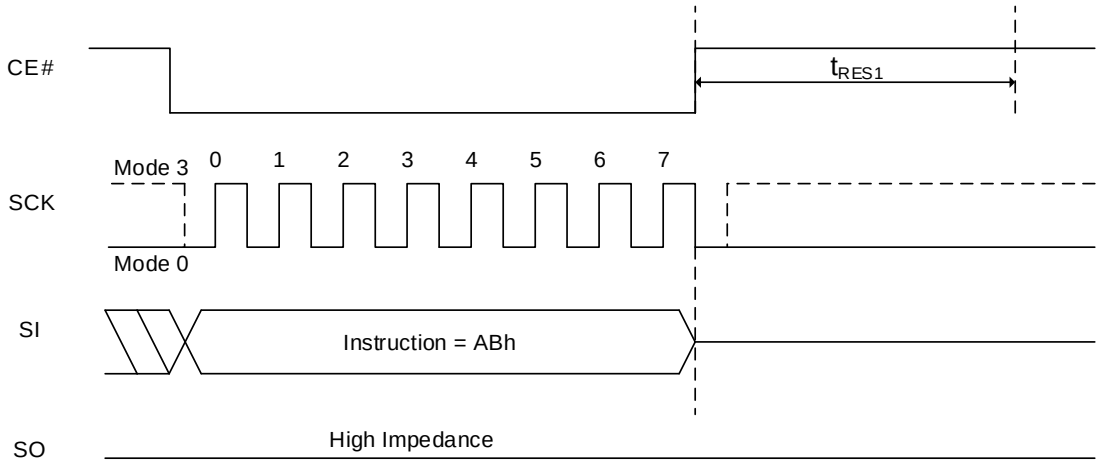
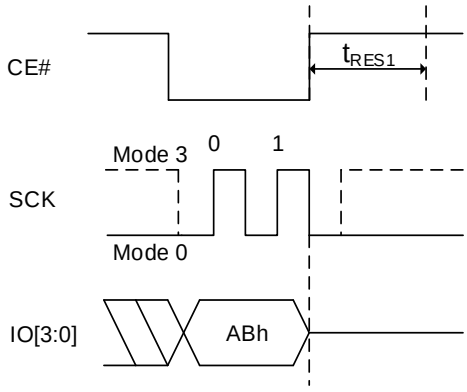


Figure 8.58 Release Power Down Mode Sequence In QPI Mode



8.26 SET READ PARAMETERS OPERATION (SRPNV: 65h, SRPV: C0h/63h)

Set Read Parameter Bits

This device supports configurable burst length and dummy cycles in both SPI and QPI mode by setting three bits (P2, P1, P0) and four bits (P6, P5, P4, P3) within the Read Register, respectively. To set those bits the SRPNV and SRPV operation instruction are used. Details regarding burst length and dummy cycles can be found in Table 6.9, Table 6.10, and Table 6.11. HOLD#/RESET# function selection (P7) bit in the Read Register can be set with the SRPNV and SRPV operation as well, in order to select HOLD#/RESET# pin as RESET# or HOLD#.
For the device with dedicated RESET# pin (or ball), RESET# pin (or ball) will be a separate pin (or ball) and it is independent of the P7 bit setting in Read Register.

SRPNV is used to set the non-volatile Read Register, while SRPV is used to set the volatile Read Register.

Note: When SRPNV is executed, the volatile Read Register is set as well as the non-volatile Read Register.

Figure 8.59 Set Read Parameters Sequence In SPI Mode

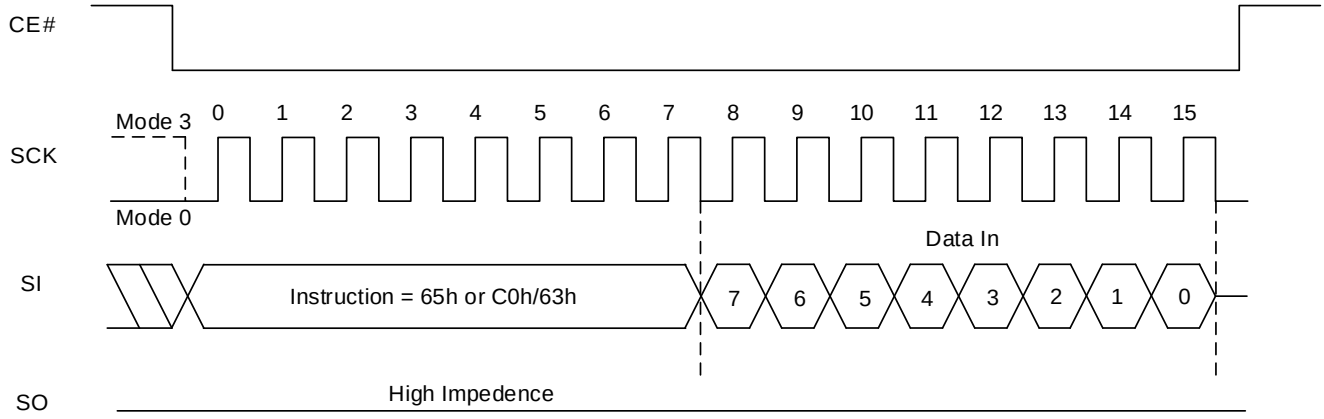
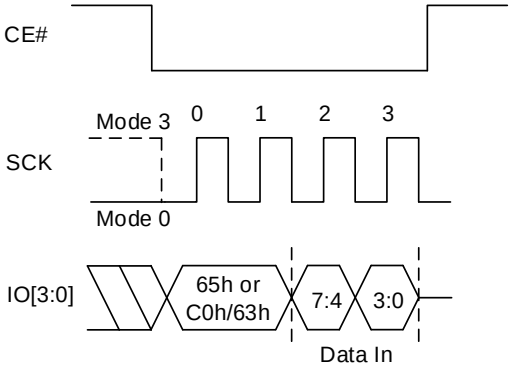


Figure 8.60 Set Read Parameters Sequence In QPI Mode



Read with "8/16/32/64-Byte Wrap Around"

The device is capable of burst read with wrap around in both SPI and QPI mode. The size of burst length is configurable by using P0, P1, and P2 bits in Read Register. P2 bit (Wrap enable) enables the burst mode feature. P0 and P1 define the size of burst. Burst lengths of 8, 16, 32, and 64 bytes are supported. By default, address increases by one up through the entire array. By setting the burst length, the data being accessed can be limited to the length of burst boundary within a 256 byte page. The first output will be the data at the initial address which is specified in the instruction. Following data will come out from the next address within the burst boundary. Once the address reaches the end of boundary, it will automatically move to the first address of the boundary. CE# high will terminate the command.

For example, if burst length of 8 and initial address being applied is 0h, following byte output will be from address 00h and continue to 01h,..,07h, 00h, 01h… until CE# terminates the operation. If burst length of 8 and initial address being applied is FEh(254d), following byte output will be from address FEh and continue to FFh, F8h, F9h, FAh, FBh, FCh, FDh, and repeat from FEh until CE# terminates the operation.

The commands, "SRPV (65h) or SRPNV (C0h or 63h)", are used to configure the burst length. If the following data input is one of "00h","01h","02h", and "03h", the device will be in default operation mode. It will be continuous burst read of the whole array. If the following data input is one of "04h","05h","06h", and "07h", the device will set the burst length as 8,16,32 and 64, respectively.

To exit the burst mode, another "C0h or 63h" command is necessary to set P2 to 0. Otherwise, the burst mode will be retained until either power down or reset operation. To change burst length, another "C0h or 63h" command should be executed to set P0 and P1 (Detailed information in Table 6.9 Burst Length Data). All read commands will operate in burst mode once the Read Register is set to enable burst mode.

Refer to Figure 8.59 and Figure 8.60 for instruction sequence.

8.27 SET EXTENDED READ PARAMETERS OPERATION (SERPNV: 85h, SERPV: 83h)

Set Read Operational Driver Strength

This device supports configurable Operational Driver Strength in both SPI and QPI modes by setting three bits (ODS0, ODS1, ODS2) within the Extended Read Register. To set the ODS bits the SERPNV and SERPV operation instructions are required. The device's driver strength can be reduced as low as 12.50% of full drive strength. Details regarding the driver strength can be found in Table 6.14.

SERPNV is used to set the non-volatile Extended Read register, while SERPV is used to set the volatile Extended Read register.

Notes:
1. **The default driver strength is set to 50%.**
2. **When SERPNV is executed, the volatile Read Extended Register is set as well as the non-volatile Read Extended Register.**

Figure 8.61 Set Extended Read Parameters Sequence In SPI Mode

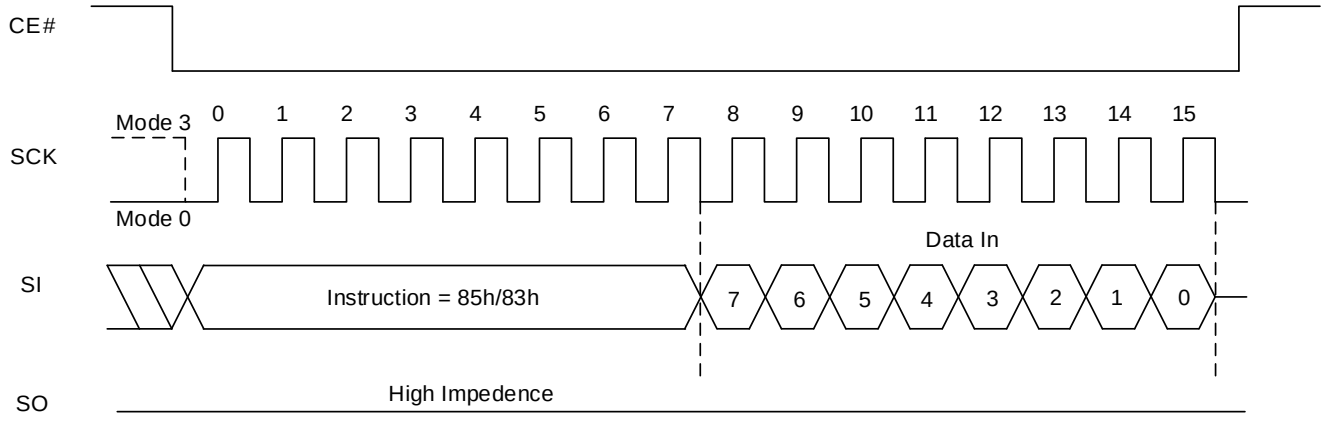
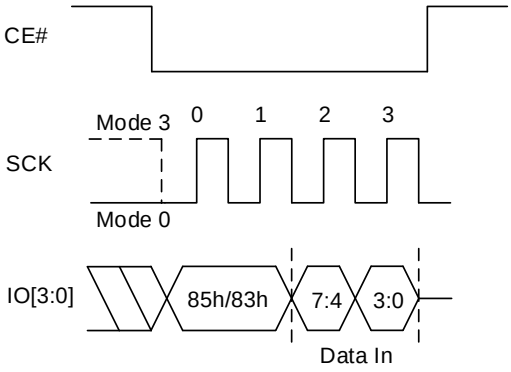


Figure 8.62 Set Extended Read Parameters Sequence In QPI Mode



8.28 READ READ PARAMETERS OPERATION (RDRP, 61h)

Prior to, or after setting Read Register, the data of the Read Register can be confirmed by the RDRP command. The instruction is only applicable for the volatile Read Register, not for the non-volatile Read Register.

Figure 8.63 Read Read Parameters Sequence In SPI Mode

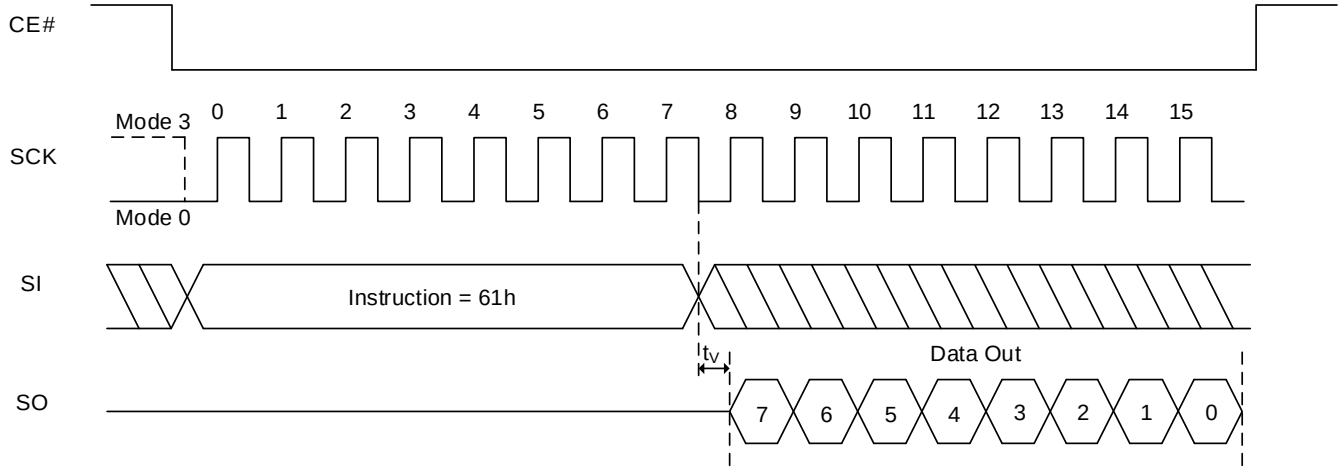
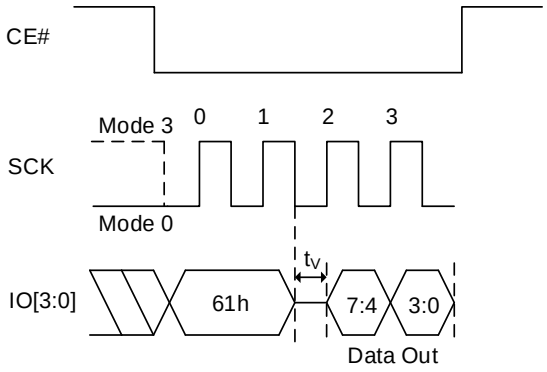


Figure 8.64 Read Read Parameters Sequence In QPI Mode



8.29 READ EXTENDED READ PARAMETERS OPERATION (RDERP, 81h)

Prior to, or after setting Extended Read Register, the data of the Extended Read Register can be confirmed by the RDERP command. The instruction is only applicable for the volatile Extended Read Register, not for the non-volatile Extended Read Register.

During the execution of a Program, Erase or Write Non-Volatile Register operation, the RDERP instruction will be executed, which can be used to check the progress or completion of an operation by reading the WIP bit.

Figure 8.65 Read Extended Read Parameters Sequence In SPI Mode

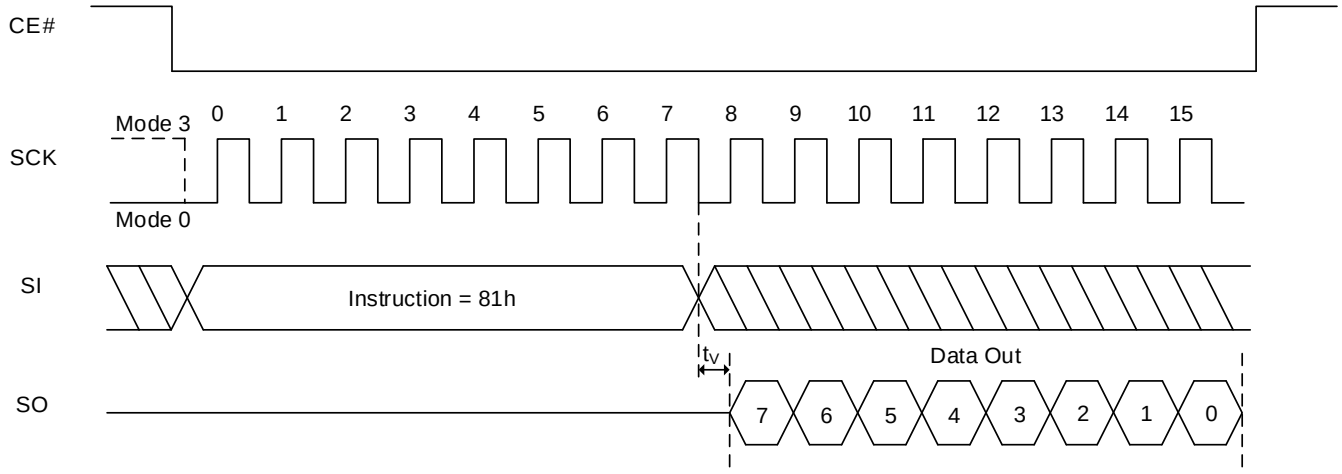
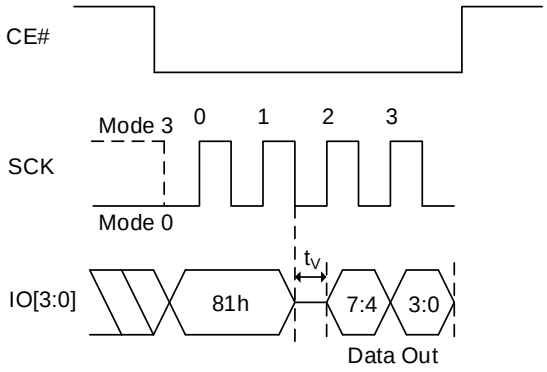


Figure 8.66 Read Extended Read Parameters Sequence In QPI Mode



8.30 CLEAR EXTENDED READ REGISTER OPERATION (CLERP, 82h)

A Clear Extended Read Register (CLERP) instruction clears PROT_E, P_ERR, and E_ERR error bits in the Extended Read Register to “0” when the error bits are set to “1”. Once the error bits are set to “1”, they remains set to “1” until they are cleared to “0” with a CLERP command.

Figure 8.67 Clear Extended Read Register Sequence In SPI Mode

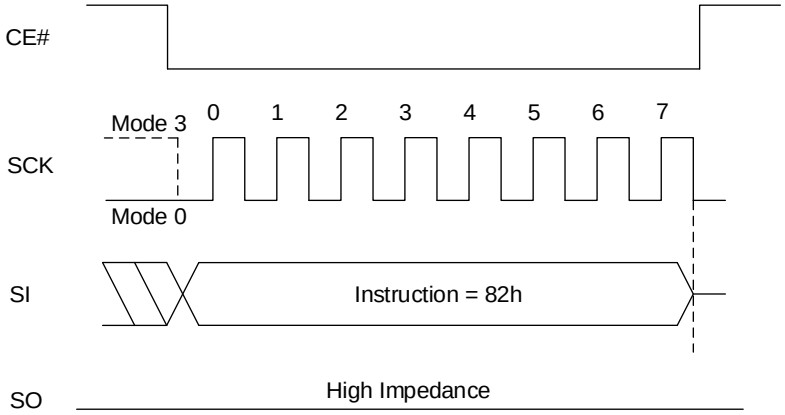
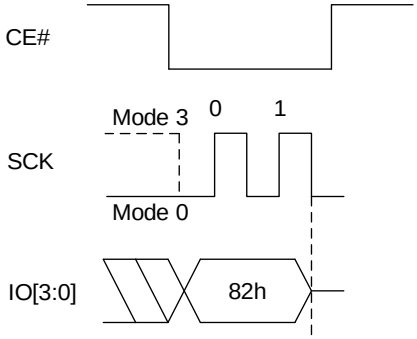


Figure 8.68 Clear Extended Read Register Sequence In QPI Mode



8.31 READ PRODUCT IDENTIFICATION (RDID, ABh)

The Release from Power-down/Read Device ID instruction is a multi-purpose instruction. It can support both SPI and QPI modes. The Read Product Identification (RDID) instruction is for reading out the old style of 8-bit Electronic Signature, whose values are shown as the table of Product Identification.

The RDID instruction code is followed by three dummy bytes, each bit being latched-in on SI during the rising SCK edge. Then the Device ID is shifted out on SO with the MSB first, each bit been shifted out during the falling edge of SCK. The RDID instruction is ended by driving CE# high. The Device ID (ID7-ID0) outputs repeatedly if additional clock cycles are continuously sent to SCK while CE# is at low.

Table 8.8 Product Identification

Manufacturer ID		**(MF7-MF0)**	
ISSI Serial Flash		9Dh	
Instruction	**ABh**	**90h**	**9Fh**
Part Number	Device ID (ID7-ID0)		Memory Type + Capacity (ID15-ID0)
IS25LP256D	18h		6019h
IS25WP256D	18h		7019h

Figure 8.69 Read Product Identification Sequence

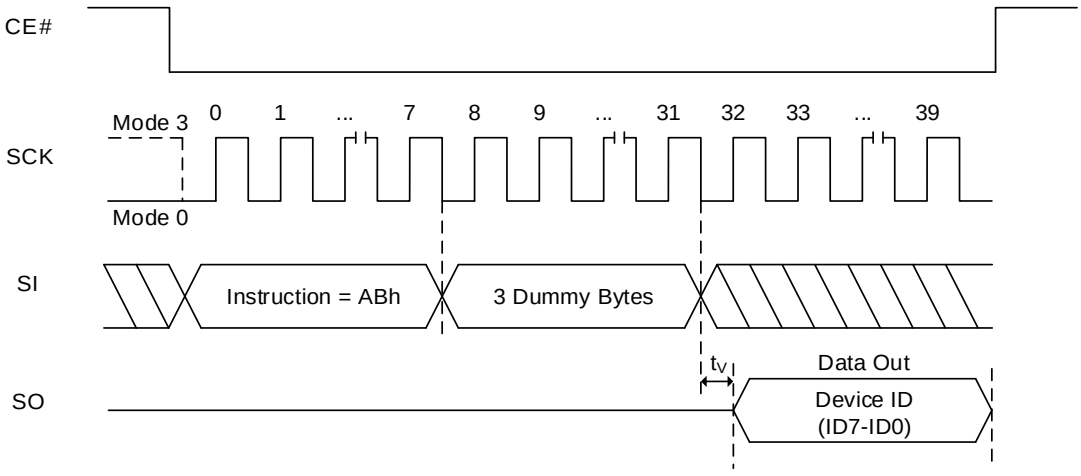
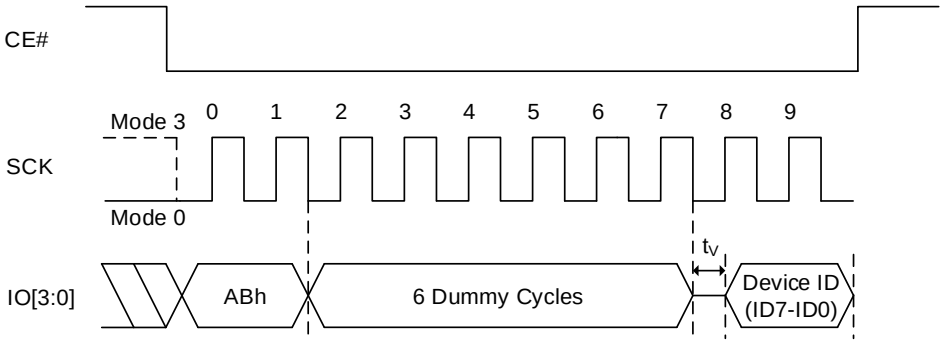


Figure 8.70 Read Product Identification Sequence In QPI Mode



8.32 READ PRODUCT IDENTIFICATION BY JEDEC ID OPERATION (RDJDID, 9Fh; RDJDIDQ, AFh)

The JEDEC ID READ instruction allows the user to read the manufacturer and product ID of devices. Refer to Table 8.8 Product Identification for Manufacturer ID and Device ID. After the JEDEC ID READ command (9Fh in SPI mode, AFh in QPI mode) is input, the Manufacturer ID is shifted out MSB first followed by the 2-byte electronic ID (ID15-ID0) that indicates Memory Type and Capacity, one bit at a time. Each bit is shifted out during the falling edge of SCK. If CE# stays low after the last bit of the 2-byte electronic ID, the Manufacturer ID and 2-byte electronic ID will loop until CE# is pulled high.

Figure 8.71 Read Product Identification by JEDEC ID Read Sequence In SPI Mode

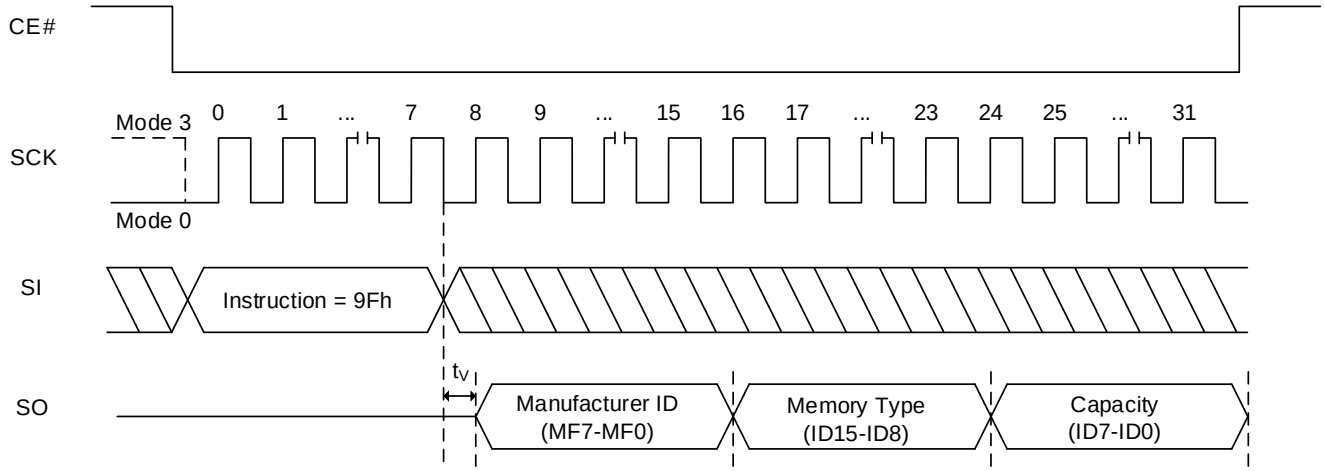
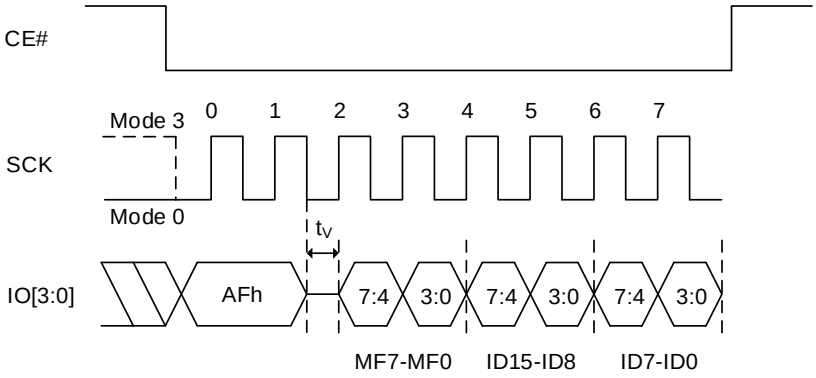


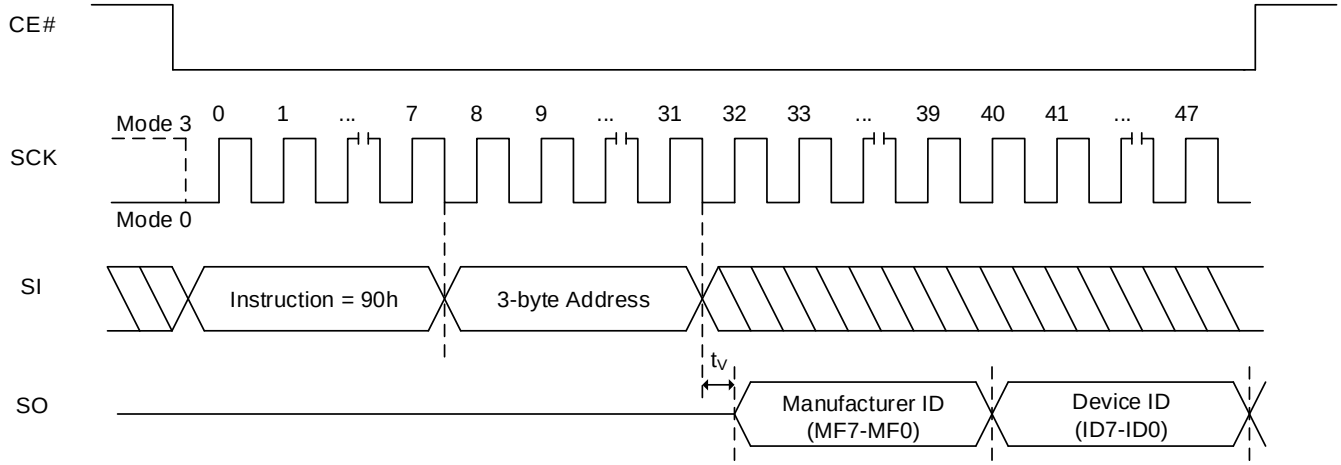
Figure 8.72 RDJDID and RDJDIDQ (Read JEDEC ID) Sequence In QPI MOde



8.33 READ DEVICE MANUFACTURER AND DEVICE ID OPERATION (RDMDID, 90h)

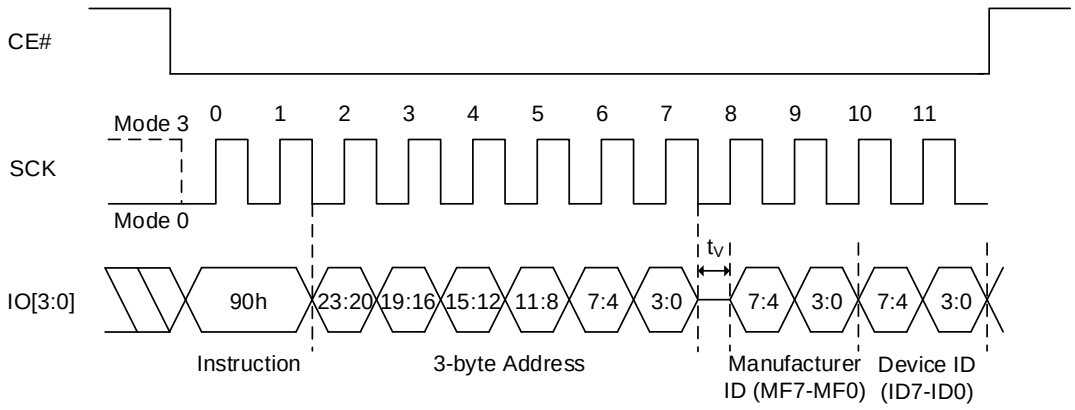
The Read Device Manufacturer and Device ID (RDMDID) instruction allows the user to read the Manufacturer and product ID of devices. Refer to Table 8.8 Product Identification for Manufacturer ID and Device ID. The RDMDID instruction code is followed by two dummy bytes and one byte address (A7~A0), each bit being latched-in on SI during the rising edge of SCK. If one byte address is initially set as A0 = 0, then the Manufacturer ID is shifted out on SO with the MSB first followed by the device ID (ID7- ID0). Each bit is shifted out during the falling edge of SCK. If one byte address is initially set as A0 = 1, then Device ID7-ID0 will be read first followed by the Manufacturer ID. The Manufacturer and Device ID can be read continuously alternating between the two until CE# is driven high.

Figure 8.73 Read Product Identification by RDMDID Sequence In SPI Mode



Notes:
1. **ADDRESS A0 = 0, will output the 1-byte Manufacturer ID (MF7-MF0) → 1-byte Device ID (ID7-ID0)**
   ADDRESS A0 = 1, will output the 1-byte Device ID (ID7-ID0) → 1-byte Manufacturer ID (MF7-MF0)
2. **The Manufacturer and Device ID can be read continuously and will alternate from one to the other until CE# pin is pulled high.**

Figure 8.74 Read Product Identification by RDMDID Sequence In QPI Mode



Notes:
1. **ADDRESS A0 = 0, will output the 1-byte Manufacturer ID (MF7-MF0) → 1-byte Device ID (ID7-ID0)**
   ADDRESS A0 = 1, will output the 1-byte Device ID (ID7-ID0) → 1-byte Manufacturer ID (MF7-MF0)
2. **The Manufacturer and Device ID can be read continuously and will alternate from one to the other until CE# pin is pulled high.**

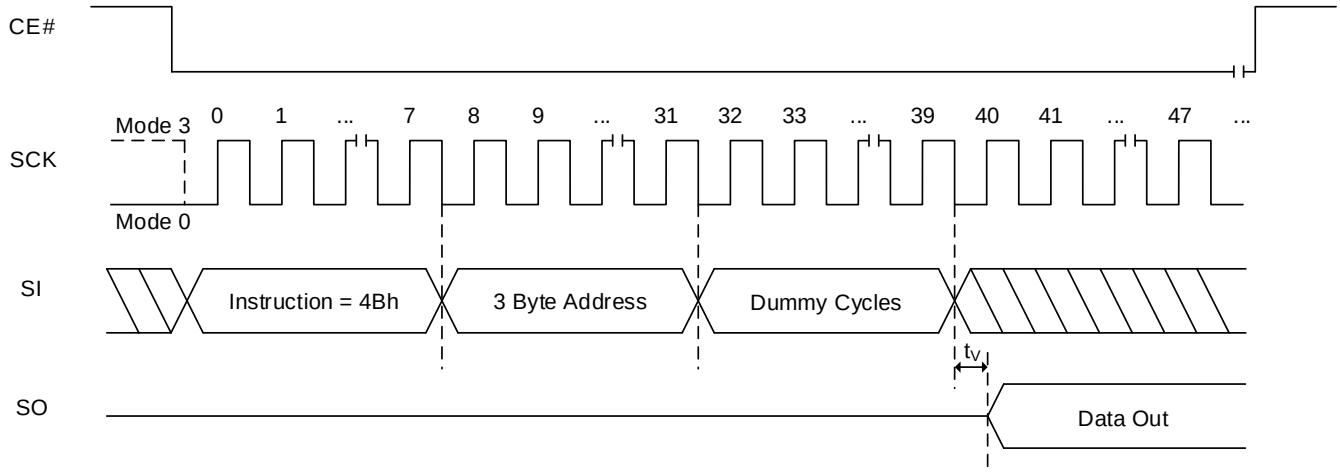
8.34 READ UNIQUE ID NUMBER (RDUID, 4Bh)

The Read Unique ID Number (RDUID) instruction accesses a factory-set read-only 16-byte number that is unique to the device. The ID number can be used in conjunction with user software methods to help prevent copying or cloning of a system. The RDUID instruction is instated by driving the CE# pin low and shifting the instruction code (4Bh) followed by 3 address bytes and dummy cycles (configurable, default is 8 clocks). After which, the 16-byte ID is shifted out on the falling edge of SCK as shown below.

As a result, the sequence of RDUID instruction is same as FAST READ. RDUID sequence in QPI mode is also same as FAST READ sequence in QPI mode except for the instruction code. Refer to the FAST READ operation in QPI mode.

Note: 16 bytes of data will repeat as long as CE# is low and SCK is toggling.

Figure 8.75 RDUID Sequence In SPI Mode



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

Table 8.9 Unique ID Addressing

A[23:16]	A[15:9]	A[8:4]	A[3:0]
XXh	XXh	00h	0h Byte address
XXh	XXh	00h	1h Byte address
XXh	XXh	00h	2h Byte address
XXh	XXh	00h	⋮
XXh	XXh	00h	Fh Byte address

Note: XX means "don't care".

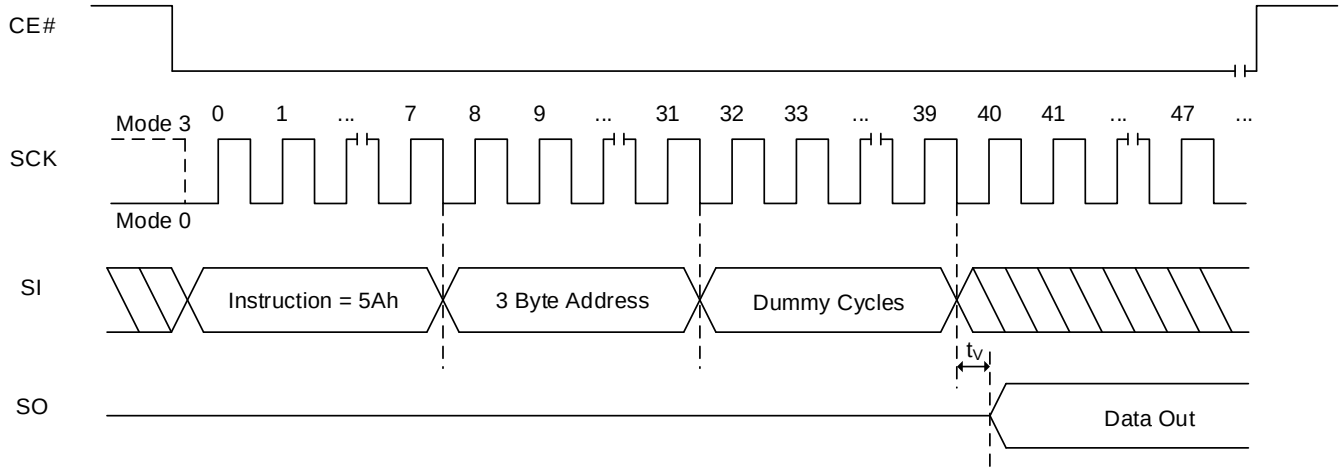
8.35 READ SFDP OPERATION (RDSFDP, 5Ah)

The Serial Flash Discoverable Parameters (SFDP) standard provides a consistent method of describing the functions and features of serial Flash devices in a standard set of internal parameter tables. These parameters can be interrogated by host system software to enable adjustments needed to accommodate divergent features from multiple vendors. For more details please refer to the JEDEC Standard JESD216A (Serial Flash Discoverable Parameters).

The sequence of issuing RDSFDP instruction is same as FAST_READ: CE# goes low → Send RDSFDP instruction (5Ah) → Send 3 address bytes on SI pin → Send dummy cycles (configurable, default is 8 clocks) on SI pin → Read SFDP code on SO → End RDSFDP operation by driving CE# high at any time during data out. Refer to ISSI's Application note for SFDP table. The data at the addresses that are not specified in SFDP table are undefined.

The sequence of RDSFDP instruction is same as FAST READ except for the instruction code. RDSFDP QPI sequence is also same as FAST READ QPI except for the instruction code. Refer to the FAST READ QPI operation.

Figure 8.76 RDSFDP (Read SFDP) Sequence



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

8.36 NO OPERATION (NOP, 00h)

The No Operation command solely cancels a Reset Enable command and has no impact on any other commands. It is available in both SPI and QPI modes. To execute a NOP, the host drives CE# low, sends the NOP command cycle (00H), then drives CE# high.

8.37 SOFTWARE RESET (RESET-ENABLE (RSTEN, 66h) AND RESET (RST, 99h)) AND HARDWARE RESET

The Software Reset operation is used as a system reset that puts the device in normal operating mode. During the Reset operation, the value of volatile registers will default back to the value in the corresponding non-volatile register. However, the volatile FREEZE bit and the volatile PPB Lock bit in the PPB Lock Register are not changed by Software Reset. In all other respects, Software Reset is the same as Hardware Reset. This operation consists of two commands: Reset-Enable (RSTEN) and Reset (RST). The 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.

Execute the CE# pin low → sends the Reset-Enable command (66h), and drives CE# high. Next, the host drives CE# low again, sends the Reset command (99h), and pulls CE# high.

Only if the RESET# pin is enabled, Hardware Reset function is available.
For the device with HOLD#/RESET#, the RESET# pin will be solely applicable in SPI mode and when the QE bit = "0". For the device with dedicated RESET**#** (Dedicated RESET**#** Disable bit is "0" in Function Register), the RESET**#** pin is always applicable regardless of the QE bit value in Status Register and HOLD#/RESET# selection bit (P7) in Read Register in SPI/QPI mode.

In order to activate Hardware Reset, the RESET# pin (or ball) must be driven low for a minimum period of t_{RESET} (100ns). Drive RESET# low for a minimum period of t_{RESET} will interrupt any on-going internal and external operations, release the device from deep power down mode**[1],** disable all input signals, force the output pin enter a state of high impedance, and reset all the read parameters.
The required wait time after activating a HW Reset before the device will accept another instruction is t_{HWRST} of 35us.

The Software/Hardware Reset during an active Program or Erase operation aborts the operation, which can result in corrupting or losing the data of the targeted address range. Depending on the prior operation, the reset timing may vary. Recovery from a Write operation will require more latency than recovery from other operations.

Note1: The Status and Function Registers remain unaffected.

Figure 8.77 Software Reset Enable and Software Reset Sequence (RSTEN, 66h + RST, 99h)

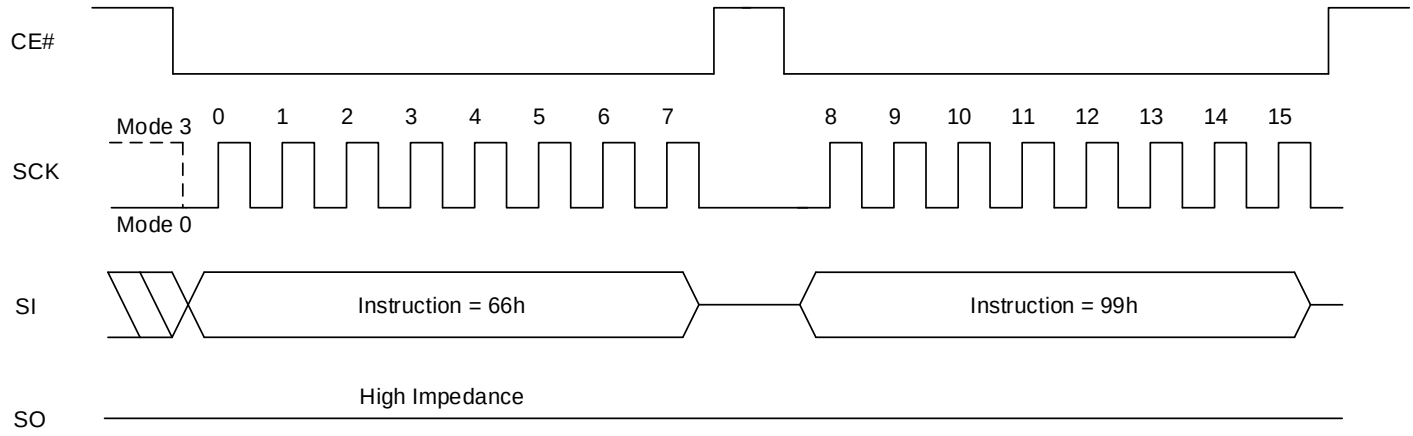
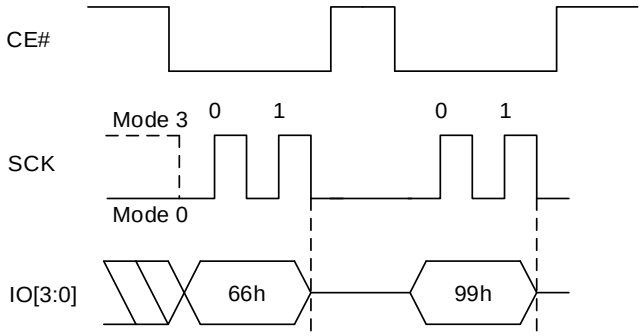


Figure 8.78 Software Reset Enable and Software Reset QPI Sequence (RSTEN, 66h + RST, 99h)



8.38 SECURITY INFORMATION ROW

The security Information Row is comprised of an additional 4 x 256 bytes of programmable information. The security bits can be reprogrammed by the user. Any program security instruction issued while an erase, program or write cycle is in progress is rejected without having any effect on the cycle that is in progress.

Table 8.10 Information Row Valid Address Range

Address Assignment	A[23:16]	A[15:8]	A[7:0]
IRL0 (Information Row Lock0)	00h	00h	Byte address
IRL1	00h	10h	Byte address
IRL2	00h	20h	Byte address
IRL3	00h	30h	Byte address

Bit 7~4 of the Function Register is used to permanently lock the programmable memory array.

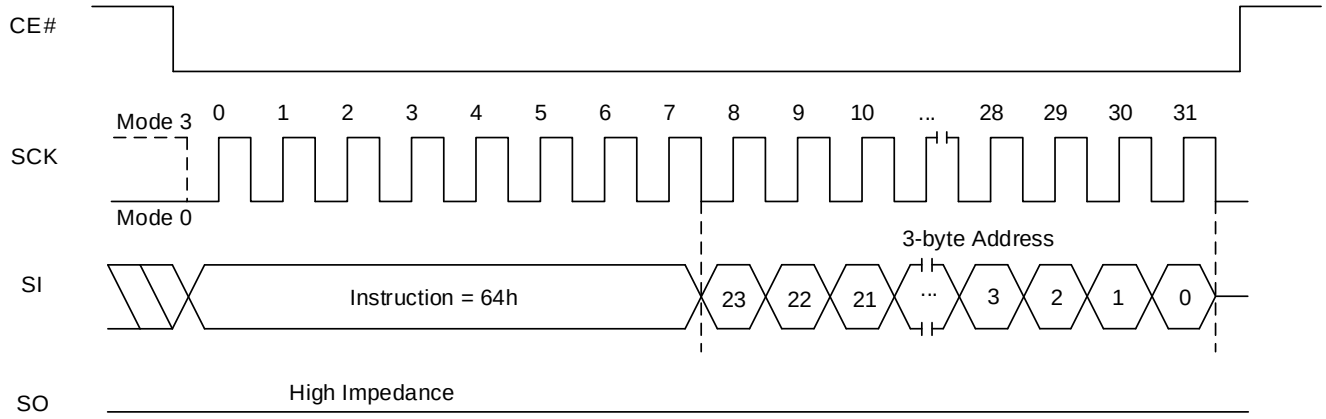
When Function Register bit IRLx = “0”, the 256 bytes of the programmable memory array can be programmed.
When Function Register bit IRLx = “1”, the 256 bytes of the programmable memory array function as read only.

8.39 INFORMATION ROW ERASE OPERATION (IRER, 64h)

Information Row Erase (IRER) instruction erases the data in the Information Row x (x: 0~3) array. Prior to the operation, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL bit is automatically reset after the completion of the operation.

The sequence of IRER operation: Pull CE# low to select the device → Send IRER instruction code → Send three address bytes → Pull CE# high. CE# should remain low during the entire instruction sequence. Once CE# is pulled high, Erase operation will begin immediately. The internal control logic automatically handles the erase voltage and timing.

Figure 8.79 IRER (Information Row Erase) Sequence



8.40 INFORMATION ROW PROGRAM OPERATION (IRP, 62h)

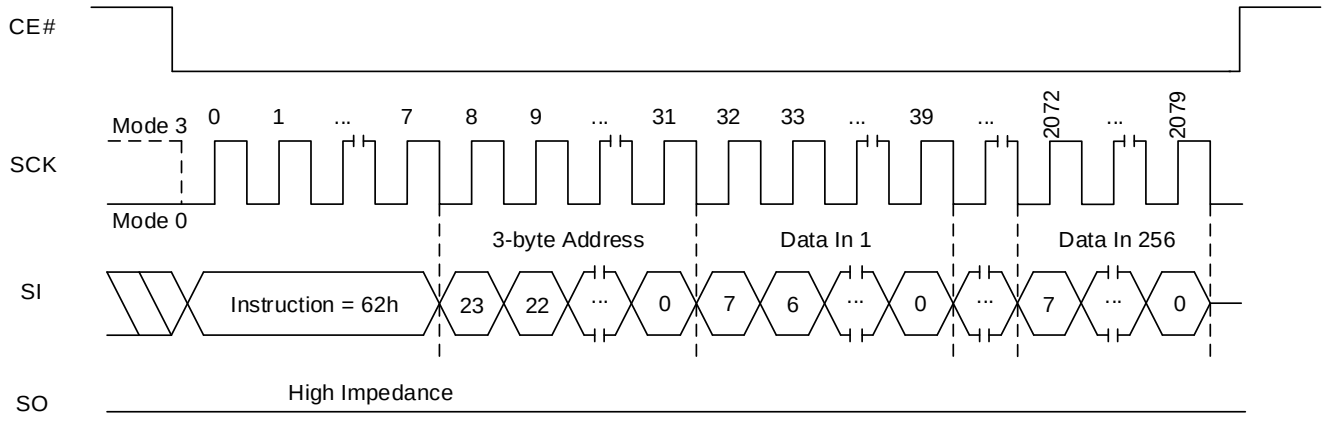
The Information Row Program (IRP) instruction allows up to 256 bytes data to be programmed into the memory in a single operation. Before the execution of IRP instruction, the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

The IRP instruction code, three address bytes and program data (1 to 256 bytes) should be sequentially input. Three address bytes has to be input as specified in the Table 8.10 Information Row Valid Address Range. Program operation will start once the CE# goes high, otherwise the IRP instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. During a program operation, all instructions will be ignored except the RDSR instruction. The progress or completion of the program operation can be determined by reading the WIP bit. If the WIP bit is "1", the program operation is still in progress. If WIP bit is "0", the program operation has completed.

If more than 256 bytes data are sent to a device, the address counter rolls over within the same page. The previously latched data are discarded and the last 256 bytes data are kept to be programmed into the page. The starting byte can be anywhere within the page. When the end of the page is reached, the address will wrap around to the beginning of the same page. If the data to be programmed are less than a full page, the data of all other bytes on the same page will remain unchanged.

Note: A program operation can alter "1"s into "0"s, but an erase operation is required to change "0"s back to "1"s. A byte cannot be reprogrammed without first erasing the corresponding Information Row array which is one of IR0~3.

Figure 8.80 IRP (Information Row Program) Sequence



8.41 INFORMATION ROW READ OPERATION (IRRD, 68h)

The IRRD instruction is used to read data from Information Row.

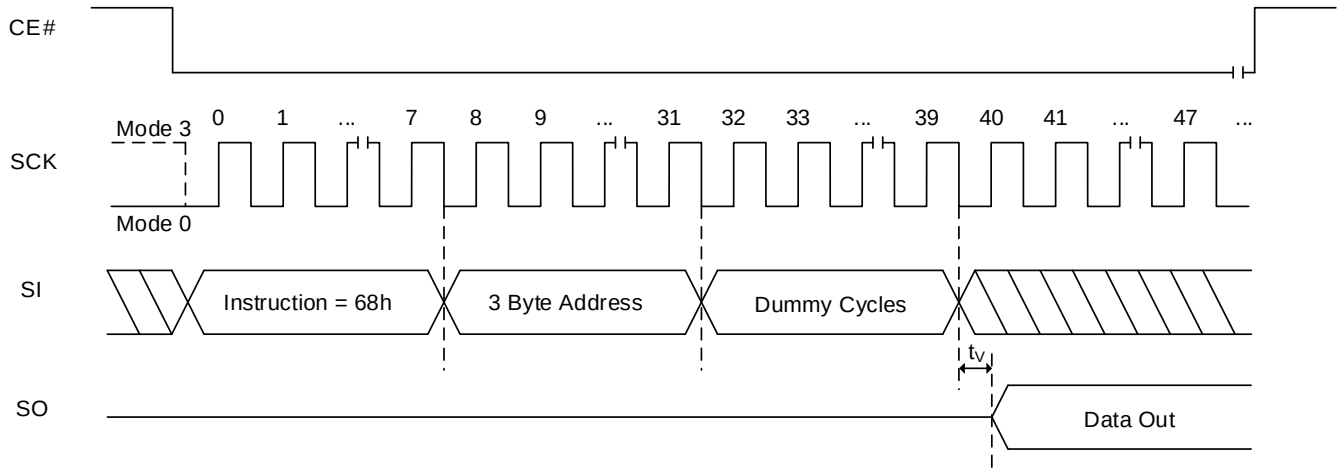
The IRRD instruction code is followed by three address bytes (A23 - A0) and dummy cycles (configurable, default is 8 clocks), transmitted via the SI line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the SO line, with each bit shifted out at a maximum frequency f_{CT}, during the falling edge of SCK.

The address is automatically incremented by one after each byte of data is shifted out. Once the address reaches the last address of each 256 byte Information Row, the next address will not be valid and the data of the address will be garbage data. It is recommended to repeat four times IRRD operation that reads 256 byte with a valid starting address of each Information Row in order to read all data in the 4 x 256 byte Information Row array. The IRRD instruction is terminated by driving CE# high (VIH).

If an IRRD instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

The sequence of IRRD instruction is same as Fast Read except for the instruction code. IRRD QPI sequence is same as Fast Read QPI except for the instruction code. Refer to the Fast Read QPI operation.
The maximum operating clock frequency for the IRRD operation is the same with Fast Read Operation.

Figure 8.81 IRRD (Information Row Read) Sequence



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

8.42 FAST READ DTR MODE OPERATION (FRDTR, 0Dh or 4FRDTR, 0Eh)

The FRDTR/4FRDTR instruction is for doubling the data in and out. Signals are triggered on both rising and falling edge of clock. The address is latched on both rising and falling edge of SCK, and data of each bit shifts out on both rising and falling edge of SCK at a maximum frequency f_{C2}. The 2-bit address can be latched-in at one clock, and 2-bit data can be read out at one clock, which means one bit at the rising edge of clock, the other bit at the falling edge of clock.

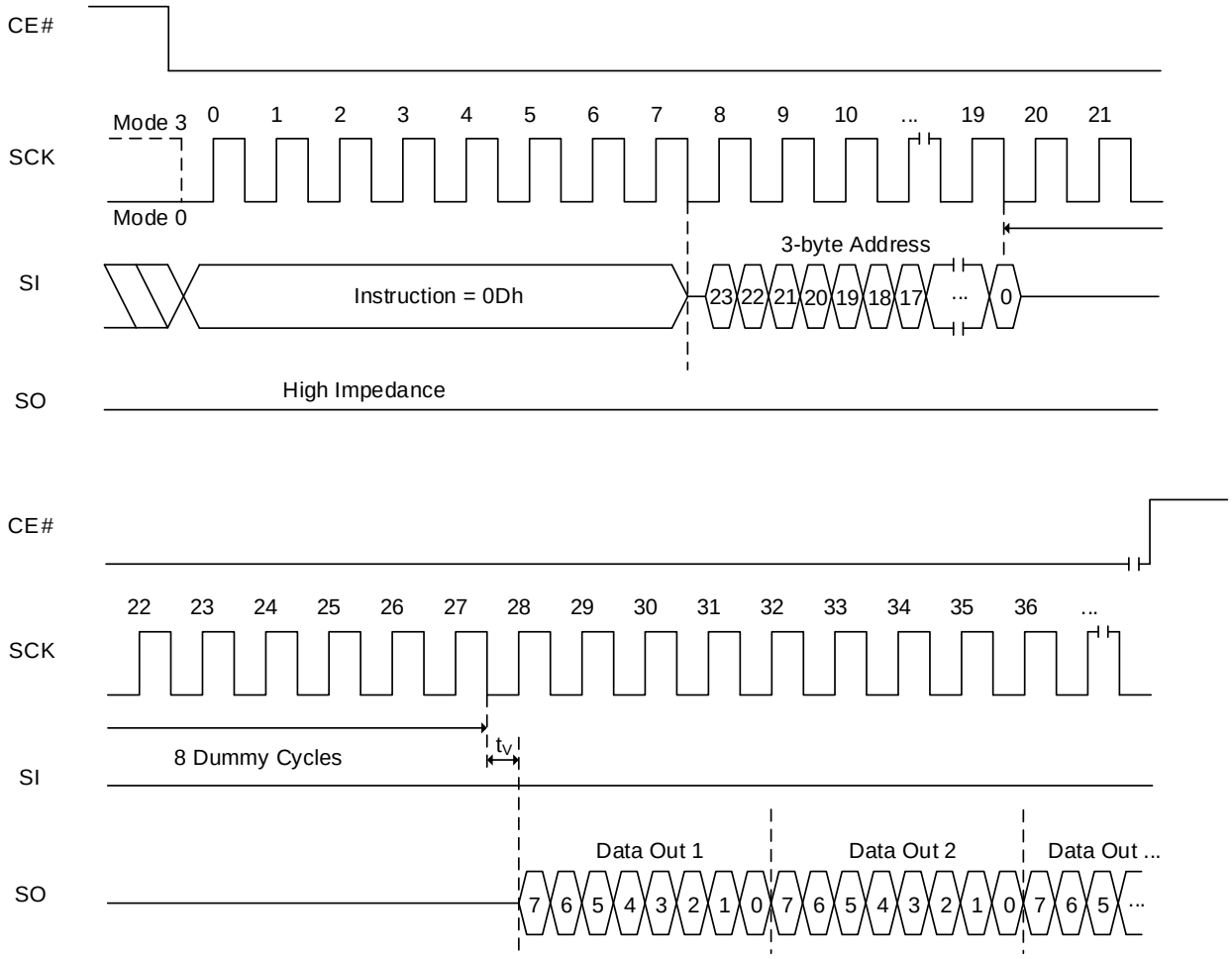
The first address byte can be at any location. The address is automatically increased to the next higher address after each byte of data is shifted out, so the whole memory can be read out in a single FRDTR/4FRDTR instruction. The address counter rolls over to 0 when the highest address is reached.

- 0Dh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 0Dh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 0Eh is followed by a 4-byte address (A31-A0)

The sequence of issuing FRDTR/4FRDTR instruction is: CE# goes low → Sending FRDTR/4FRDTR instruction code (1bit per clock) → 3-byte or 4-byte address on SI (2-bit per clock) as above → 8 dummy clocks (configurable, default is 8 clocks) on SI → Data out on SO (2-bit per clock) → End FRDTR/4FRDTR operation via driving CE# high at any time during data out.

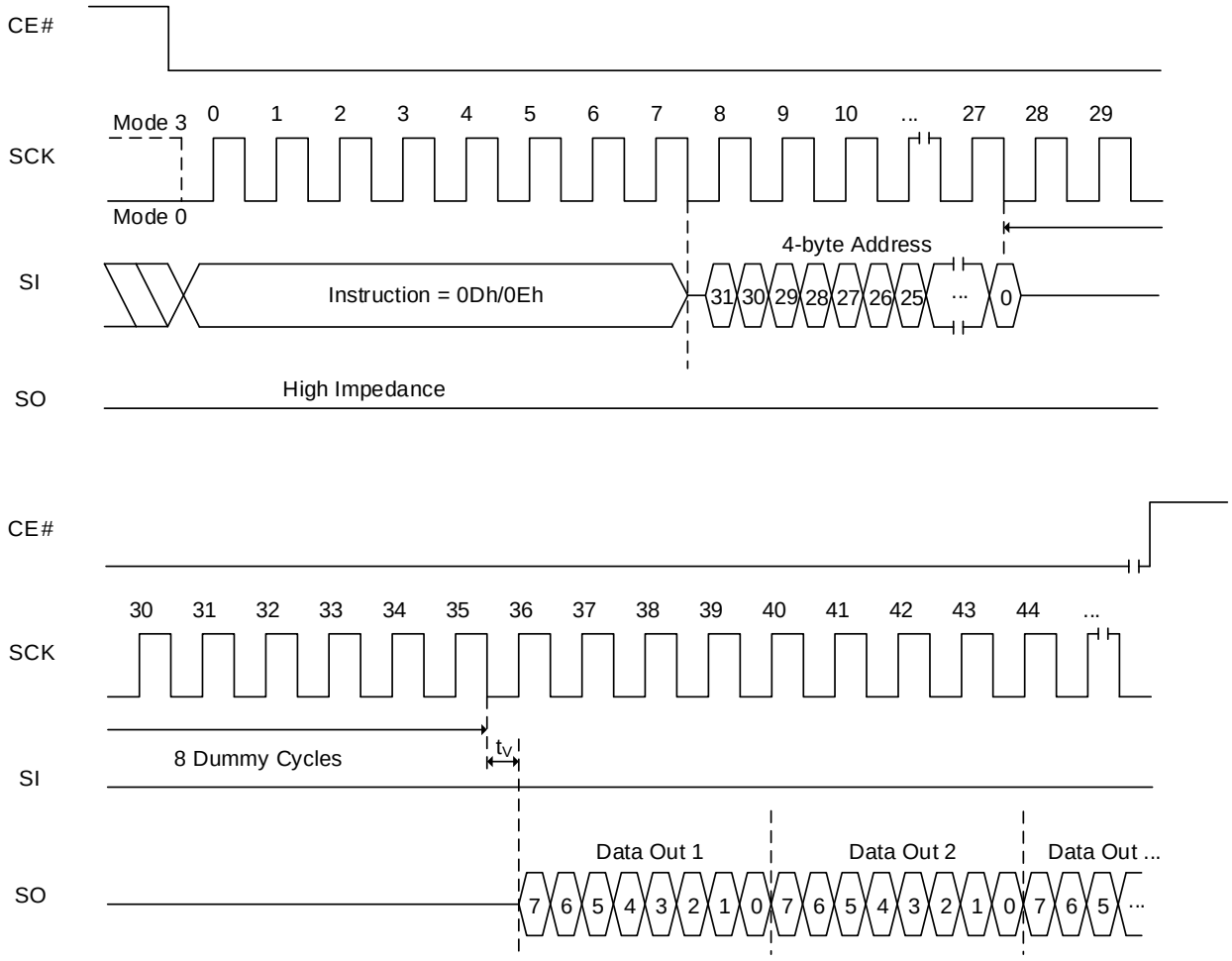
While a Program/Erase/Write Status Register cycle is in progress, FRDTR/4FRDTR instruction will be rejected without any effect on the current cycle.

Figure 8.82 FRDTR Sequence (0Dh [EXTADD=0], 3-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

Figure 8.83 FRDTR Sequence (0Dh [EXTADD=1] or 0Eh, 4-byte address)



Note: Dummy cycles depends on Read Parameter setting. Detailed information in Table 6.11 Read Dummy Cycles.

FAST READ DTR MODE OPERATION IN QPI MODE (FRDTR, 0Dh or 4FRDTR, 0Eh)

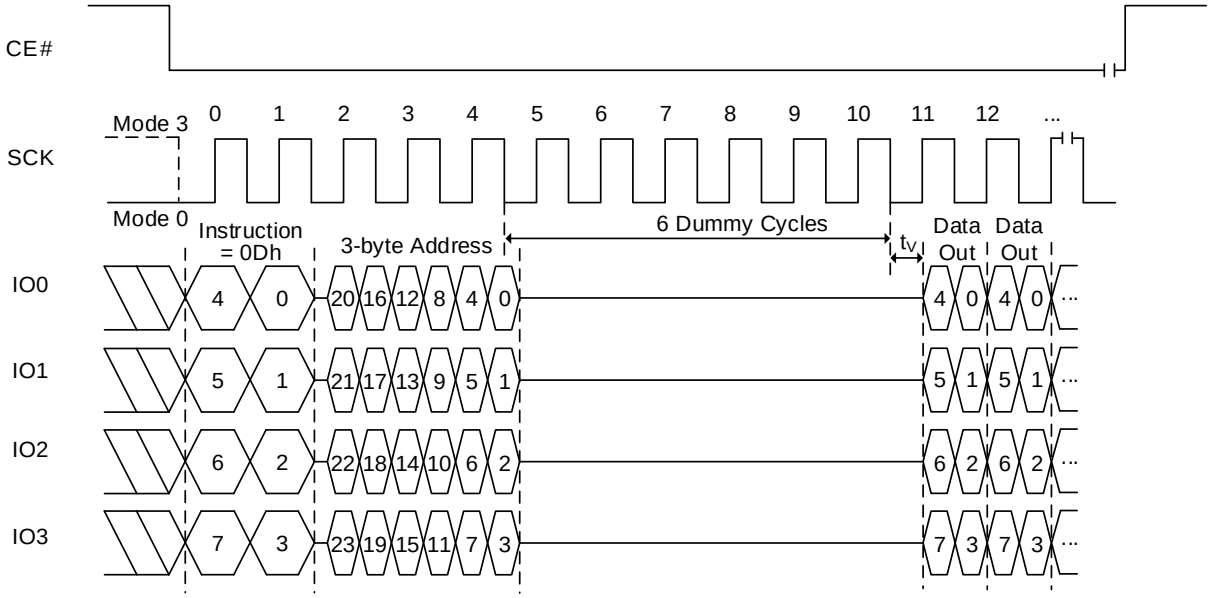
The FRDTR/4FRDTR instruction in QPI mode utilizes all four IO lines to input the instruction code so that only two clocks are required, while the FRDTR/4FRDTR instruction in SPI mode requires that the byte-long instruction code is shifted into the device only via IO0 (SI) line in eight clocks. In addition, subsequent address and data out are shifted in/out via all four IO lines unlike the FRDTR/4FRDTR instruction. Eventually this operation is same as the FRQDTR/4FRQDTR in QPI mode, but the only different thing is that AX mode is not available in the FRDTR/4FRDTR operation in QPI mode.

- 0Dh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 0Dh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 0Eh is followed by a 4-byte address (A31-A0)

The sequence of issuing FRDTR/4FRDTR QPI instruction is: CE# goes low → Sending FRDTR/4FRDTR QPI instruction (4-bit per clock) → 24-bit or 32-bit address interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) as above → 6 dummy clocks (configurable, default is 6 clocks) → Data out interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → End FRDTR/4FRDTR QPI operation by driving CE# high at any time during data out.

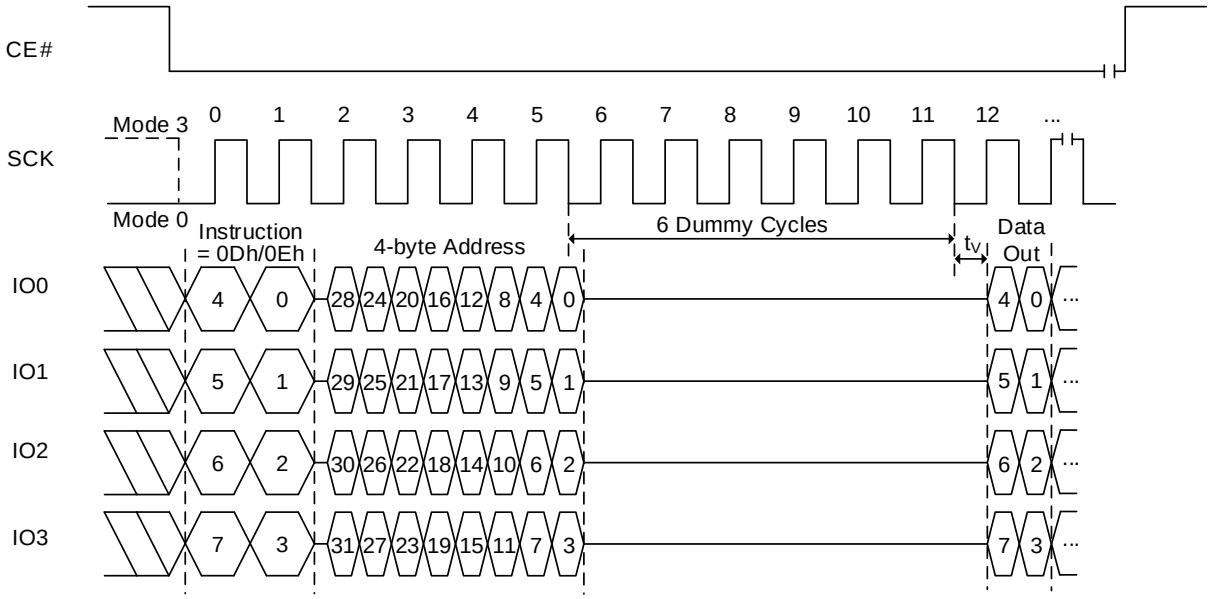
If the FRDTR/4FRDTR instruction in QPI mode is issued while an Erase, Program or Write cycle is in process is in progress (WIP=1), the instruction will be rejected without any effect on the current cycle.

Figure 8.84 FRDTR Sequence In QPI Mode ((0Dh [EXTADD=0], 3-byte address)



Notes:
1. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
2. **Sufficient dummy cycles are required to avoid I/O contention.**

Figure 8.85 FRDTR Sequence In QPI Mode (0Dh [EXTADD=1] or 0Eh, 4-byte address)



Notes:
1. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
2. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

8.43 FAST READ DUAL IO DTR MODE OPERATION (FRDDTR, BDh or 4FRDDTR, BEh)

The FRDDTR/4FRDDTR instruction enables Double Transfer Rate throughput on dual I/O of the device in read mode. The address (interleave on dual I/O pins) is latched on both rising and falling edge of SCK, and the data (interleave on dual I/O pins) shift out on both rising and falling edge of SCK at a maximum frequency f_{T2}. The 4-bit address can be latched-in at one clock, and 4-bit data can be read out at one clock, which means two bits at the rising edge of clock, the other two bits at the falling edge of clock.

The first address byte can be at any location. The address is automatically increased to the next higher address after each byte of data is shifted out, so the whole memory can be read out with a single FRDDTR/4FRDDTR instruction. The address counter rolls over to 0 when the highest address is reached. Once writing FRDDTR/4FRDDTR instruction, the following address/dummy/data out will perform as 4-bit instead of previous 1-bit.

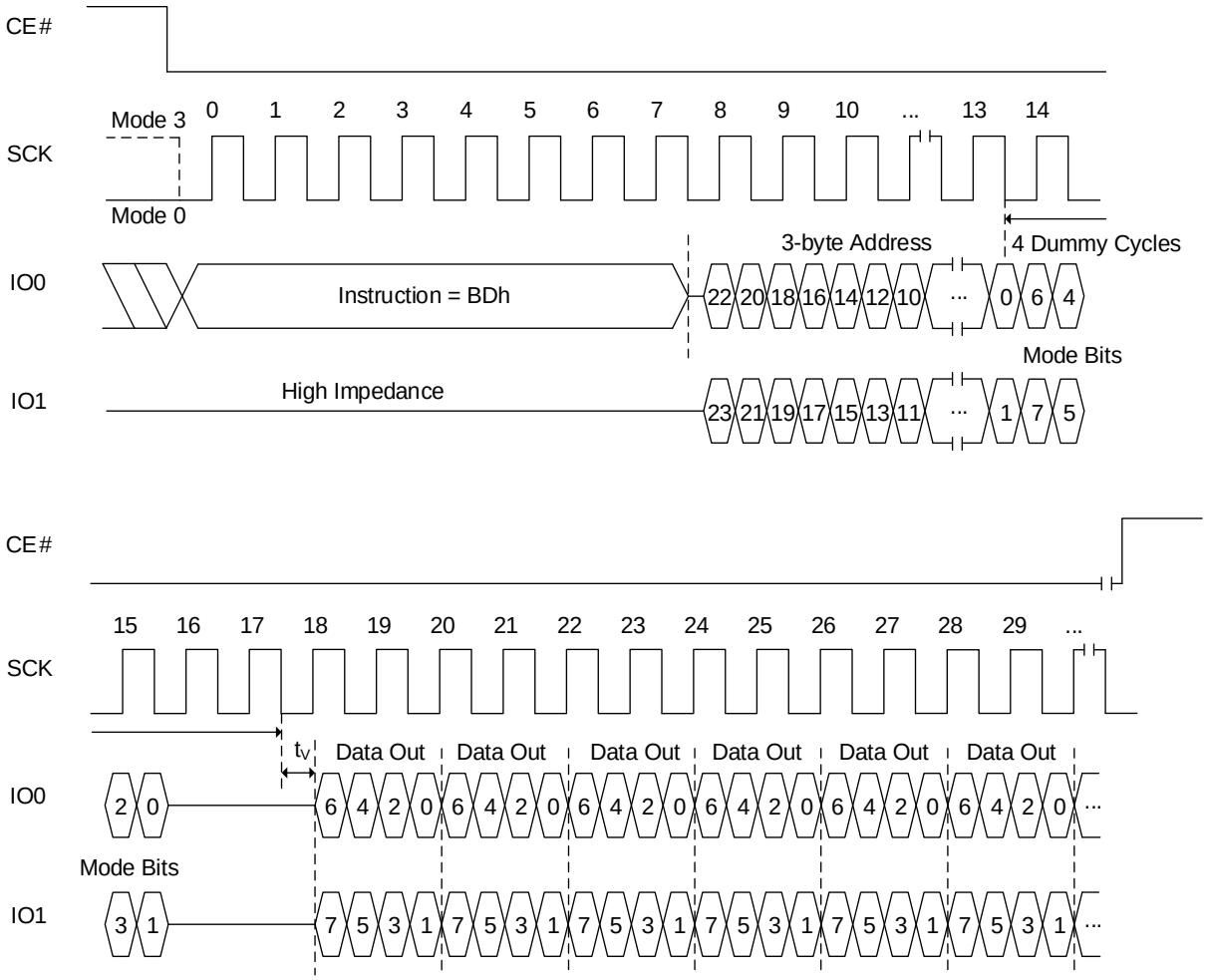
- BDh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- BDh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- BEh is followed by a 4-byte address (A31-A0)

The sequence of issuing FRDDTR/4FRDDTR instruction is: CE# goes low → Sending FRDDTR/4FRDDTR instruction (1-bit per clock) → 24-bit or 32-bit address interleave on IO1 & IO0 (4-bit per clock) as above → 4 dummy clocks (configurable, default is 4 clocks) on IO1 & IO0 → Data out interleave on IO1 & IO0 (4-bit per clock) → End FRDDTR/4FRDDTR operation via pulling CE# high at any time during data out (Please refer to Figures 8.86 and 8.87 for 2 x I/O Double Transfer Rate Read Mode Timing Waveform).

If AXh (where X is don't care) is input for the mode bits during dummy cycles, the device will enter AX read operation mode which enables subsequent FRDDTR/4FRDDTR execution skips command code. It saves cycles as described in Figures 8.88 and 8.89. When the code is different from AXh (where X is don't care), the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command.

If the FRDDTR/4FRDDTR instruction is issued while an Erase, Program or Write cycle is in process is in progress (WIP=1), the instruction will be rejected without any effect on the current cycle.

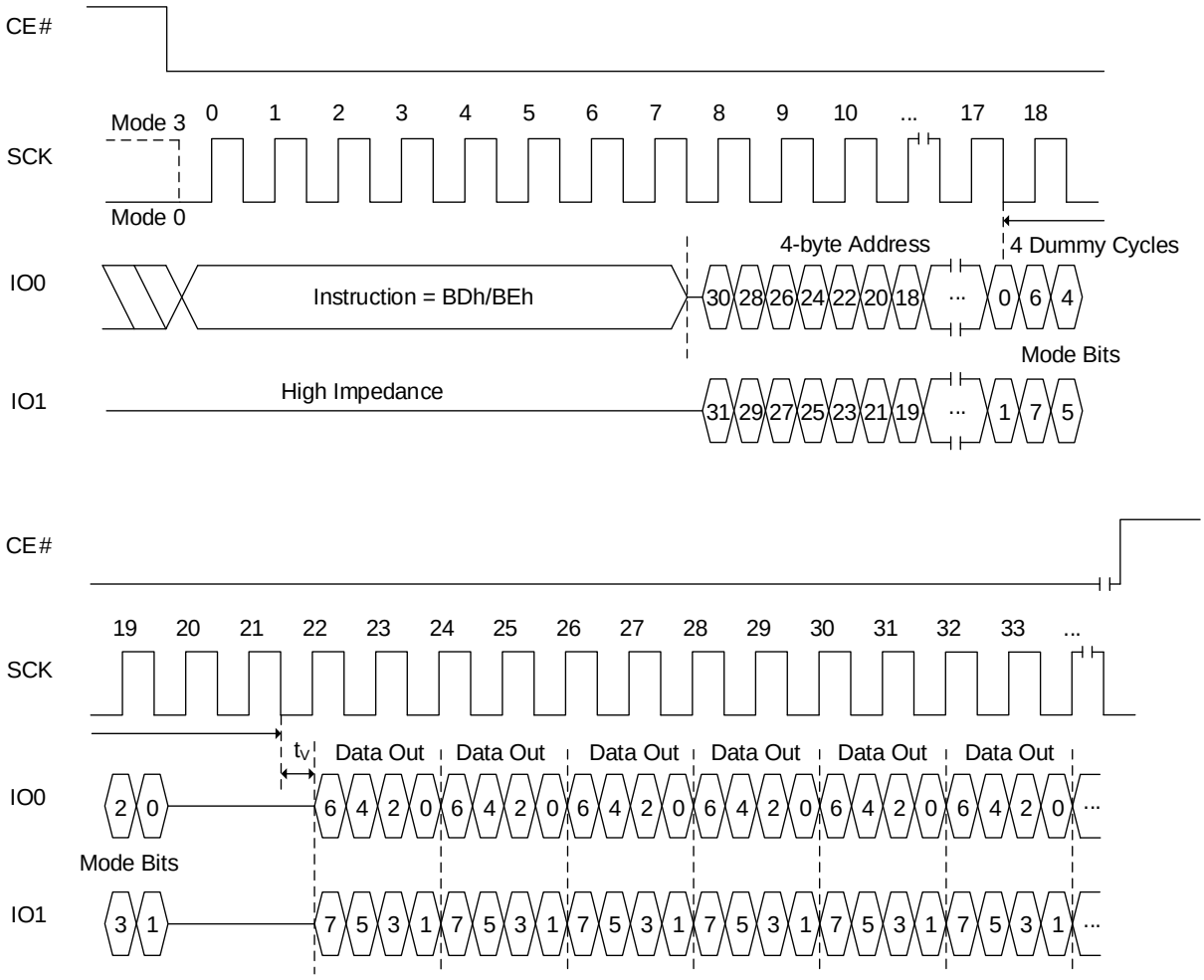
Figure 8.86 FRDDTR Sequence (BDh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

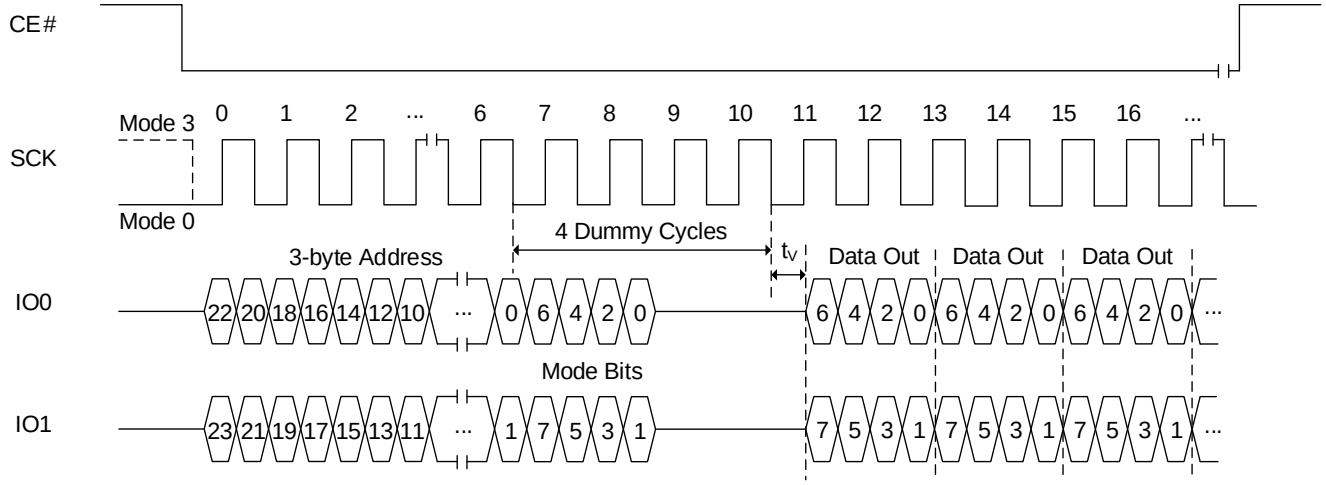
Figure 8.87 FRDDTR Sequence (BDh [EXTADD=1] or BEh, 4-byte address)



Notes:

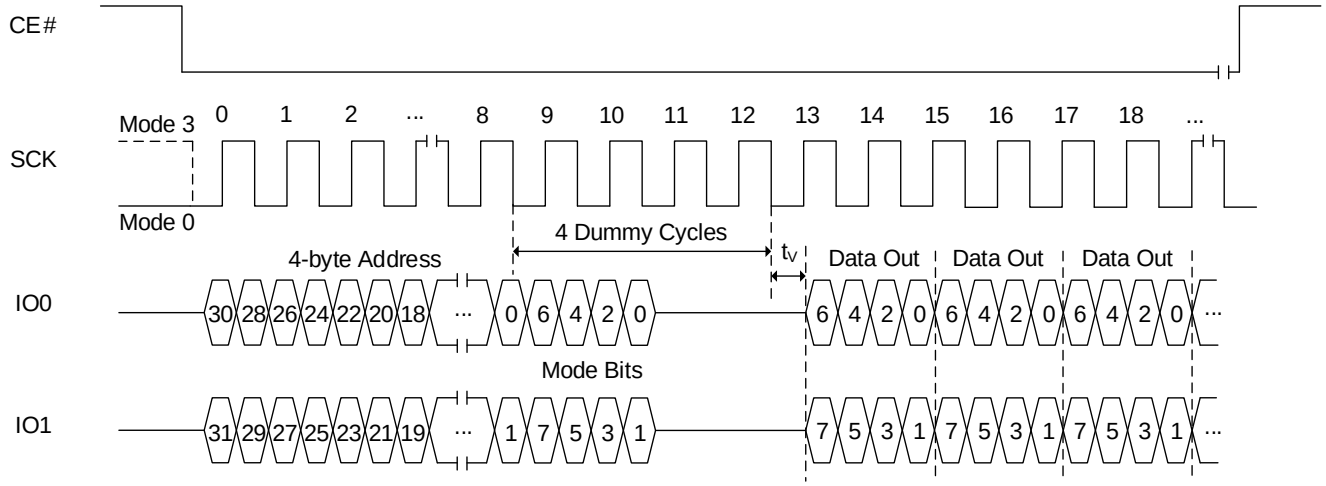
1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.88 FRDDTR AX Read Sequence (BDh [EXTADD=0], 3-byte address)



Notes:
1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.89 FRDDTR AX Read Sequence (BDh [EXTADD=1] or BEh, 4-byte address)



Notes:
1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

8.44 FAST READ QUAD I/O DTR MODE OPERATION IN SPI MODE (FRQDTR, EDh or 4FRQDTR, EEh)

The FRQDTR/4FRQDTR instruction enables Double Transfer Rate throughput on quad I/O of the device in read mode.

A Quad Enable (QE) bit of Status Register must be set to "1" before sending the Fast Read Quad I/O DTR instruction.

The address (interleave on 4 I/O pins) is latched on both rising and falling edge of SCK, and data (interleave on 4 I/O pins) shift out on both rising and falling edge of SCK at a maximum frequency f_{Q2}. The 8-bit address can be latched-in at one clock, and 8-bit data can be read out at one clock, which means four bits at the rising edge of clock, the other four bits at the falling edge of clock.

The first address byte can be at any location. The address is automatically increased to the next higher address after each byte data is shifted out, so the whole memory can be read out with a single FRQDTR/4FRQDTR instruction. The address counter rolls over to 0 when the highest address is reached. Once writing FRQDTR/4FRQDTR instruction, the following address/dummy/data out will perform as 8-bit instead of previous 1-bit.

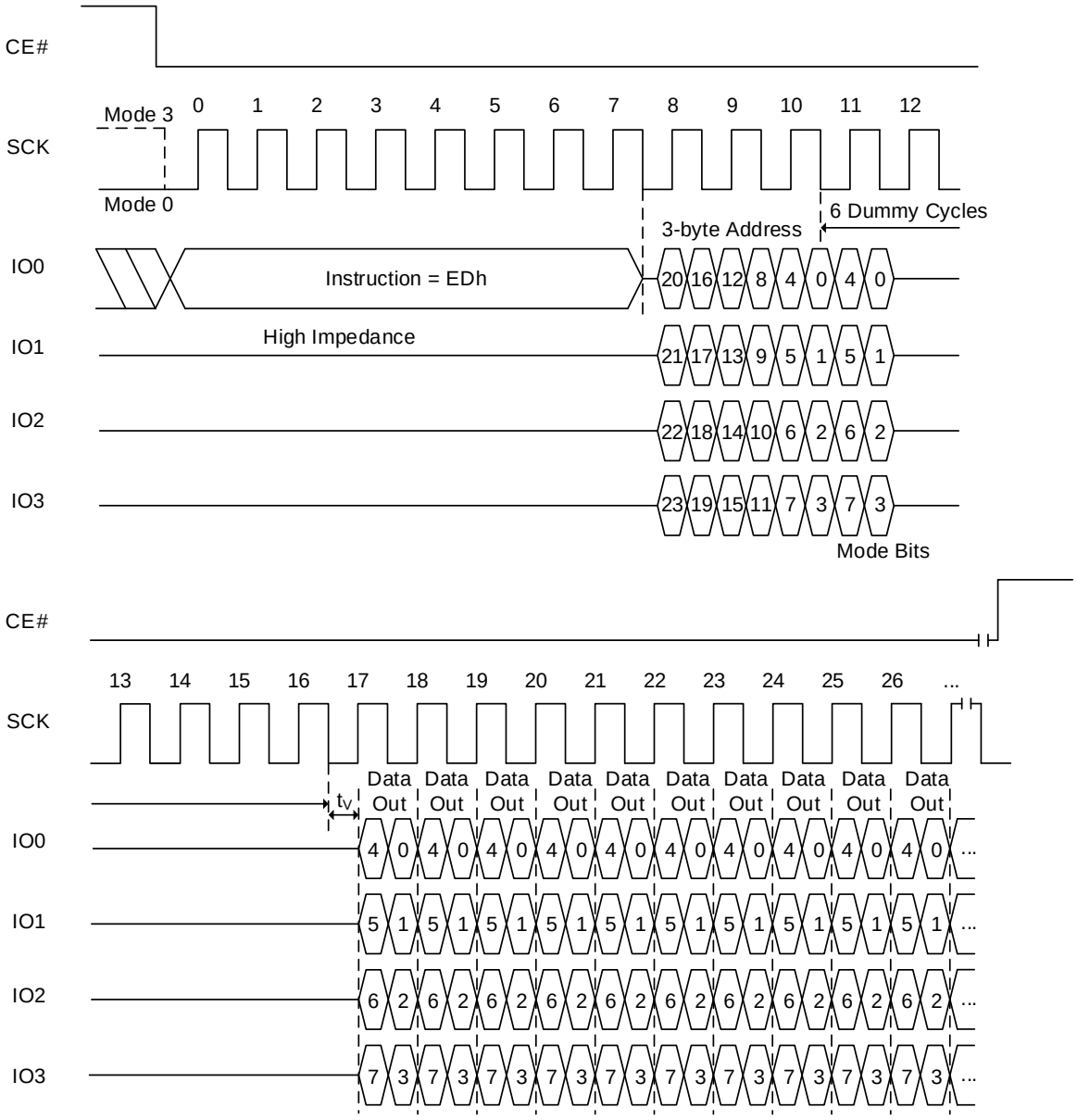
- EDh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- EDh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- EEh is followed by a 4-byte address (A31-A0)

The sequence of issuing FRQDTR/4FRQDTR instruction is: CE# goes low → Sending FRQDTR/4FRQDTR instruction (1-bit per clock) → 24-bit or 32-bit address interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) as above → 6 dummy clocks (configurable, default is 6 clocks) → Data out interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → End FRQDTR/4FRQDTR operation by driving CE# high at any time during data out.

If AXh (where X is don't care) is input for the mode bits during dummy cycles, the device will enter AX read operation mode which enables subsequent FRQDTR/4FRQDTR execution skips command code. It saves cycles as described in Figures 8.92 and 8.93. When the code is different from AXh (where X is don't care), the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command.

If the FRQDTR/4FRQDTR instruction is issued while an Erase, Program or Write cycle is in process is in progress (WIP=1), the instruction will be rejected without any effect on the current cycle.

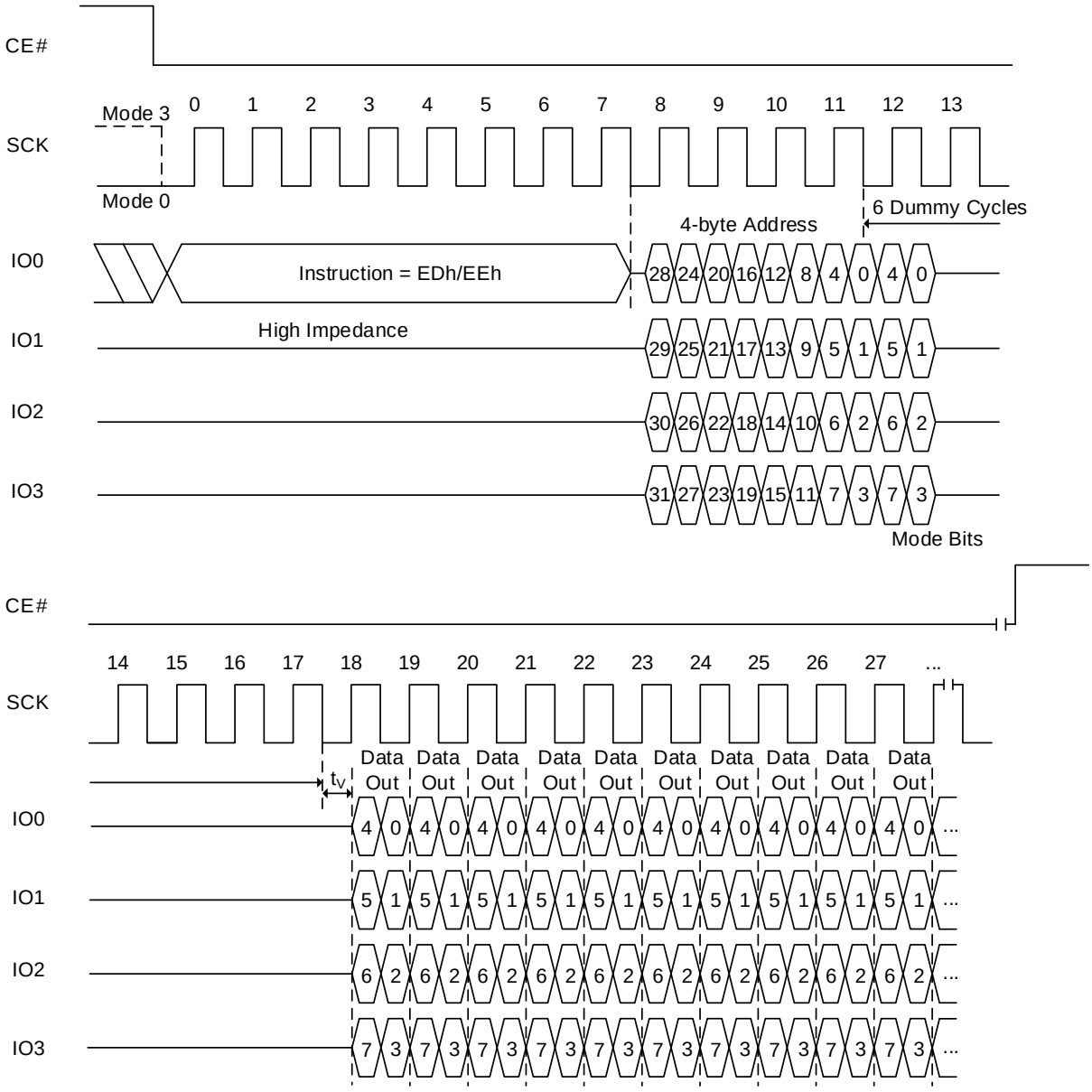
Figure 8.90 FRQDTR Sequence In SPI Mode (EDh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

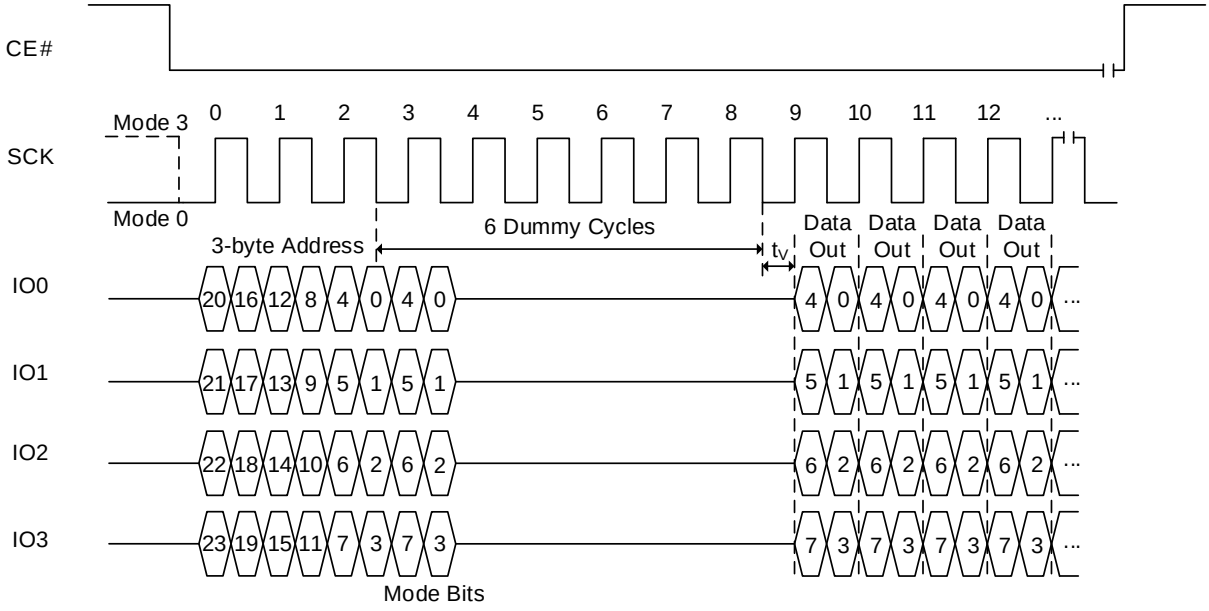
Figure 8.91 FRQDTR Sequence In SPI Mode (EDh [EXTADD=1] or EEh, 4-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

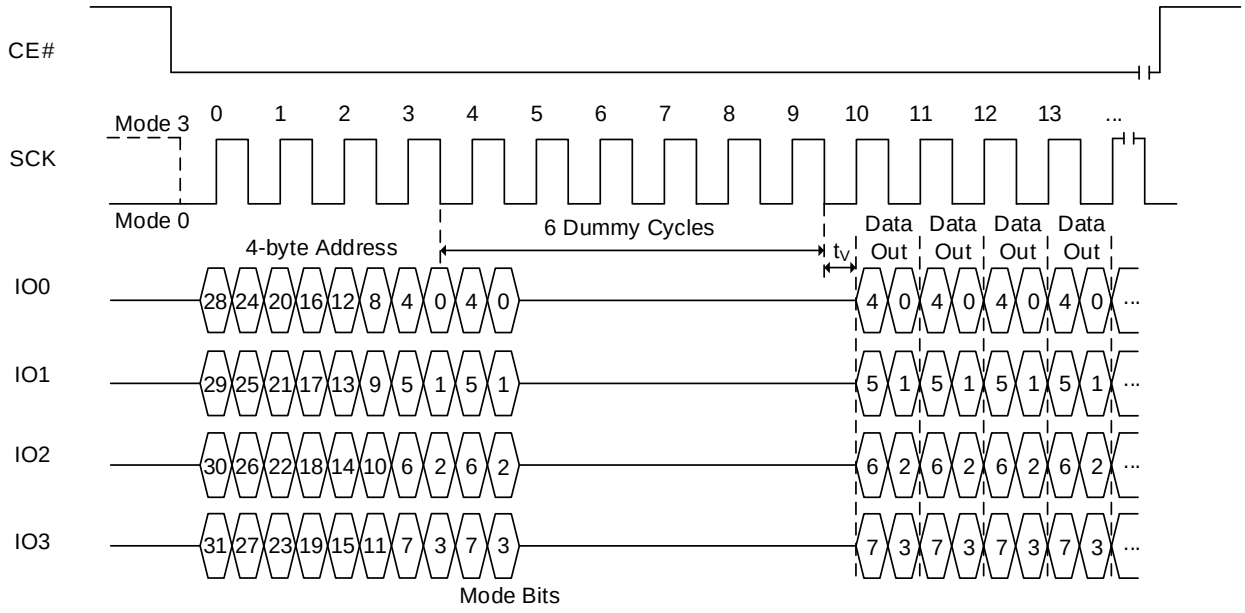
Figure 8.92 FRQDTR AX Read Sequence (EDh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.93 FRQDTR AX Read Sequence (EDh [EXTADD=1] or EEh, 4-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it will keep executing the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

FAST READ QUAD IO DTR OPERATION IN QPI MODE (FRQDTR, EDh OR 4FRQDTR, EEh)

The FRQDTR/4FRQDTR instruction in QPI mode utilizes all four IO lines to input the instruction code so that only two clocks are required, while the FRQDTR/4FRQDTR instruction in SPI mode requires that the byte-long instruction code is shifted into the device only via IO0 line in eight clocks. As a result, 6 command cycles will be reduced by the FRQDTR/4FRQDTR instruction in QPI mode. In addition, subsequent address and data out are shifted in/out via all four IO lines like the FRQDTR/4FRQDTR instruction. In fact, except for the command cycle, the FRQDTR/4FRQDTR operation in QPI mode is exactly same as the FRQDTR/4FRQDTR operation in SPI mode.

It is not required to set QE bit to "1".before Fast Read Quad I/O DTR instruction in QPI mode.

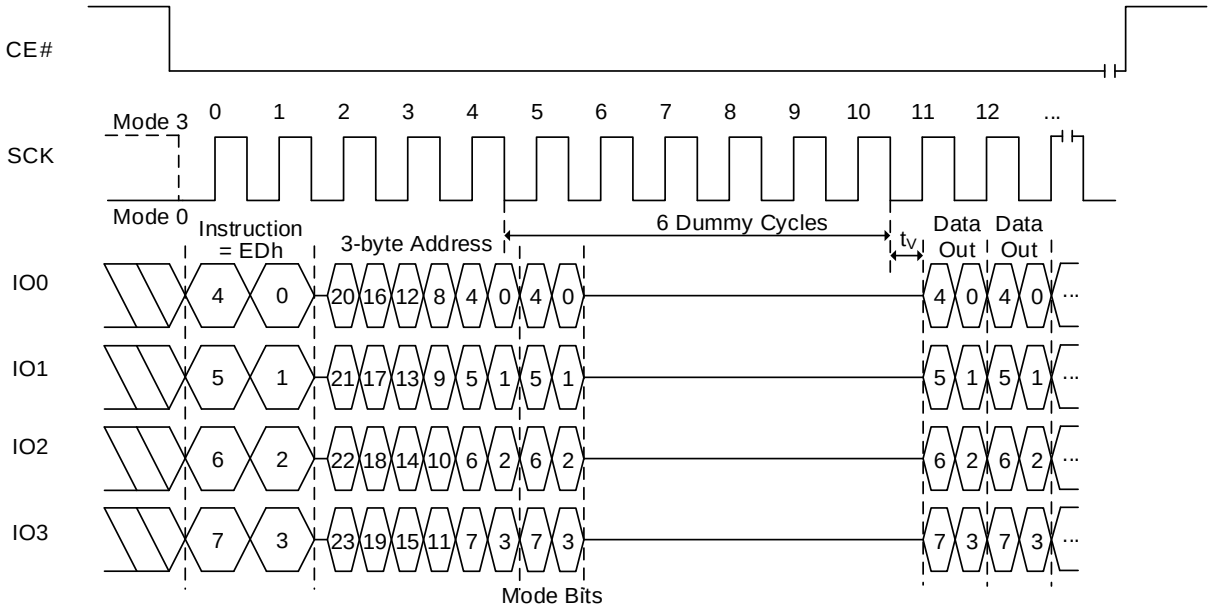
- EDh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- EDh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- EEh is followed by a 4-byte address (A31-A0)

The sequence of issuing FRQDTR/4FRQDTR instruction is: CE# goes low → Sending FRQDTR/4FRQDTR instruction (4-bit per clock) → 24-bit or 32-bit address interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) as above → 6 dummy clocks (configurable, default is 6 clocks) → Data out interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → End FRQDTR/4FRQDTR operation by driving CE# high at any time during data out.

If AXh (where X is don't care) is input for the mode bits during dummy cycles, the device will enter AX read operation mode which enables subsequent FRQDTR/4FRQDTR in QPI mode execution skips command code. When the code is different from AXh (where X is don't care), the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command.

If the FRQDTR/4FRQDTR instruction in QPI mode is issued while an Erase, Program or Write cycle is in process is in progress (WIP=1), the instruction will be rejected without any effect on the current cycle.

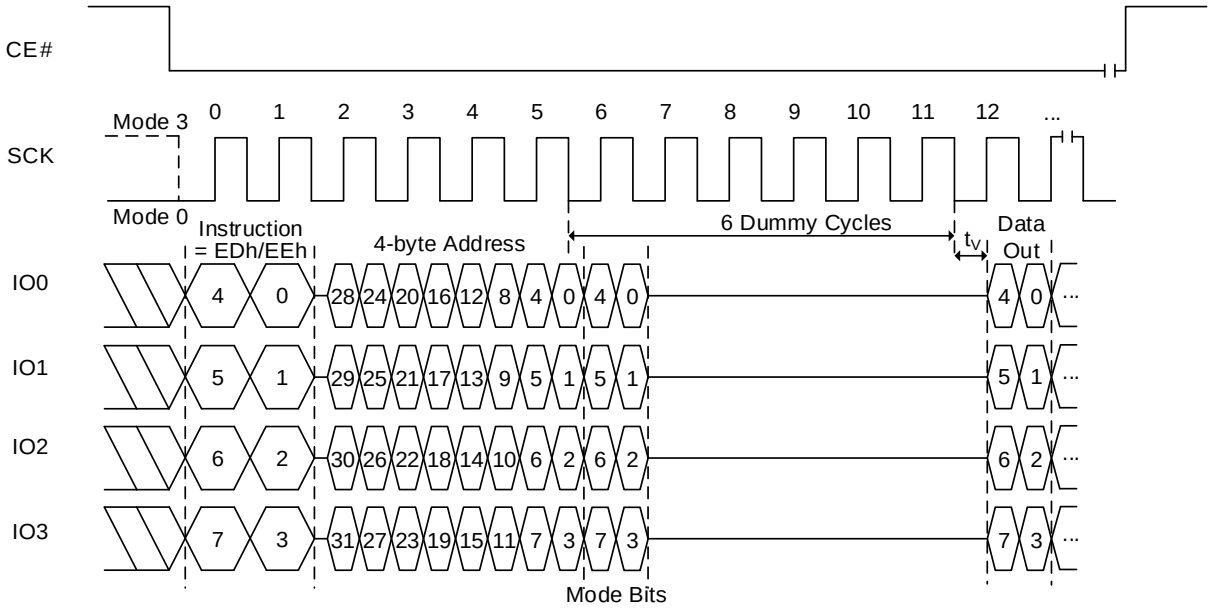
Figure 8.94 FRQDTR Sequence In QPI Mode (EDh [EXTADD=0], 3-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

Figure 8.95 FRQDTR Sequence In QPI Mode (EDh [EXTADD=1] or EEh, 4-byte address)



Notes:

1. **If the mode bits=AXh (where X is don't care), it can execute the AX read mode (without command). When the mode bits are different from AXh, the device exits the AX read operation.**
2. **Number of dummy cycles depends on clock speed. Detailed information in Table 6.11 Read Dummy Cycles.**
3. **Sufficient dummy cycles are required to avoid I/O contention. If the number of dummy cycles and AX bits cycles are same, then X should be Hi-Z.**

8.45 SECTOR LOCK/UNLOCK FUNCTIONS

SECTOR UNLOCK OPERATION (SECUNLOCK, 26h or 4SECUNLOCK, 25h)

The Sector Unlock command allows the user to select a specific sector to allow program and erase operations. This instruction is effective when the blocks are designated as write-protected through the BP0-BP3 bits in the Status Register and TBS bit in the Function Register. Only one sector can be enabled at any time. To enable a different sector, a previously enabled sector must be disabled by executing a Sector Lock command.

- 26h (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- 26h (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- 25h is followed by a 4-byte address (A31-A0)

The instruction code is followed by a 24-bit or 32-bit address specifying the target sector as above, but A0 through A11 are not decoded. The remaining sectors within the same block remain as read-only.

Figure 8.96 Sector Unlock Sequence In SPI Mode (26h [EXTADD=0], 3-byte address)

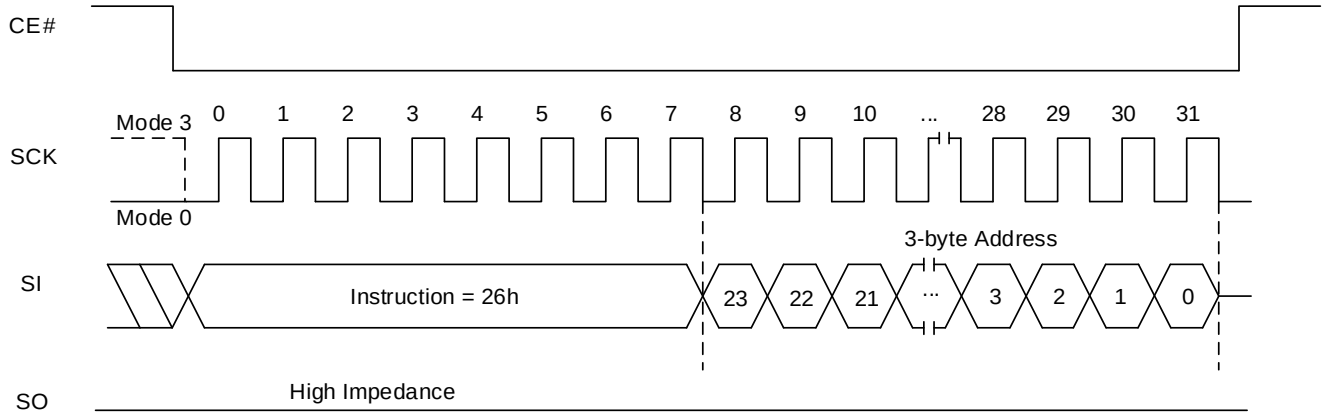


Figure 8.97 Sector Unlock Sequence In SPI Mode (26h [EXTADD=1] or 25h, 4-byte address)

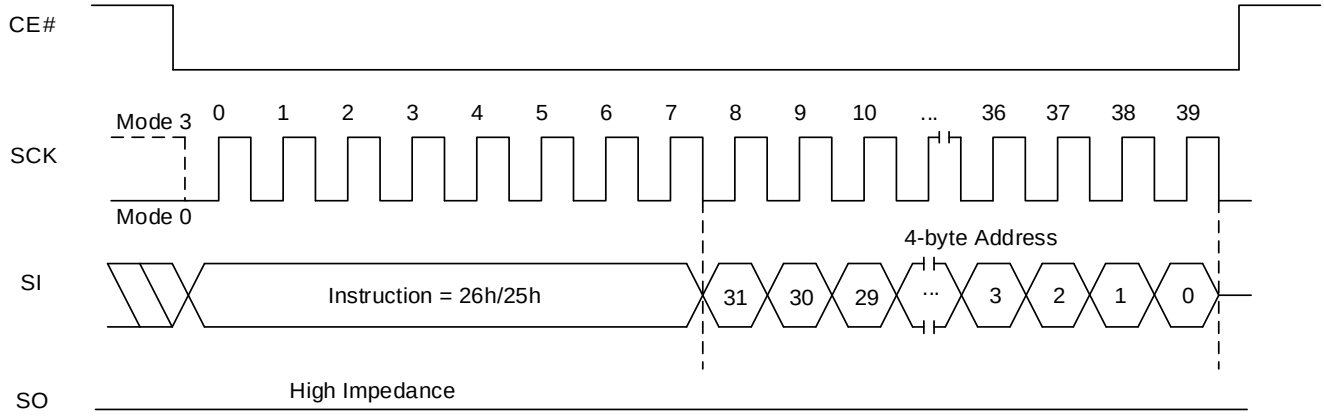


Figure 8.98 Sector Unlock Sequence In QPI Mode (26h [EXTADD=0], 3-byte address)

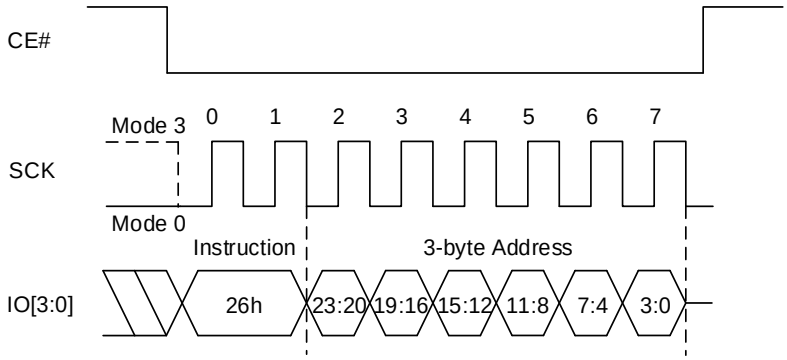
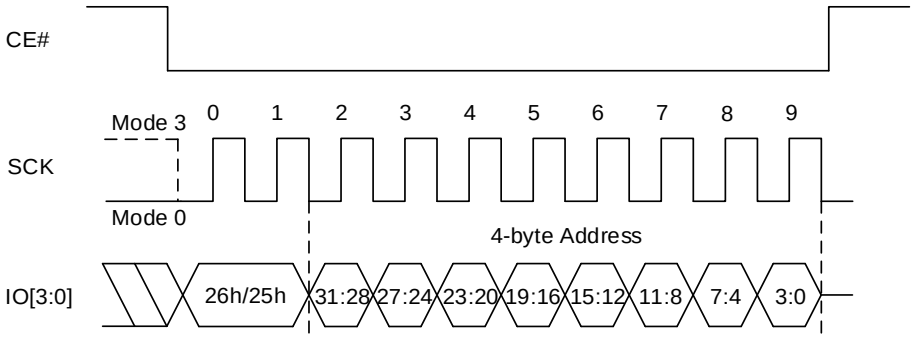


Figure 8.99 Sector Unlock Sequence In QPI Mode (26h [EXTADD=1] or 25h, 4-byte address)



SECTOR LOCK OPERATION (SECLOCK, 24h)

The Sector Lock command relocks a sector that was previously unlocked by the Sector Unlock command. The instruction code does not require an address to be specified, as only one sector can be enabled at a time. The remaining sectors within the same block remain in read-only mode.

Figure 8.100 Sector Lock Sequence In SPI Mode

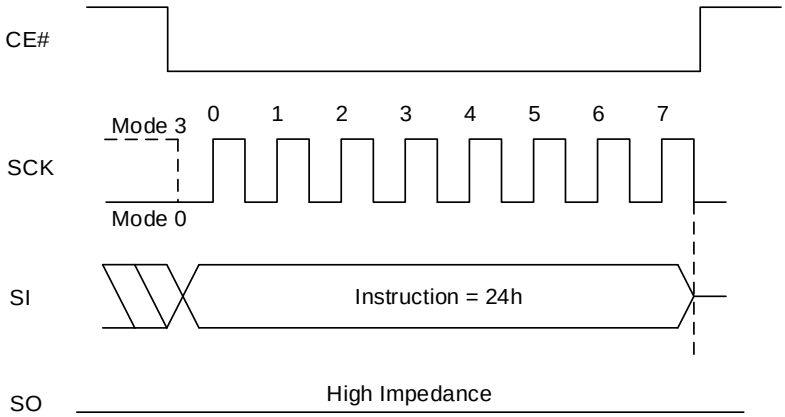
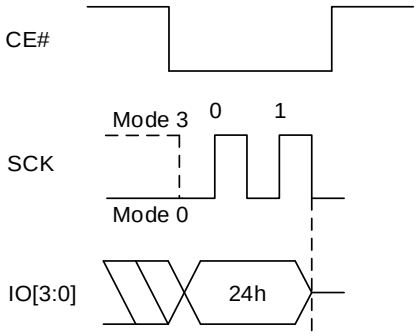


Figure 8.101 Sector Lock Sequence In QPI Mode



8.46 AUTOBOOT

SPI devices normally require 32 or more cycles of command and address shifting to initiate a read command. And, in order to read boot code from an SPI device, the host memory controller or processor must supply the read command from a hardwired state machine or from some host processor internal ROM code.
Parallel NOR devices need only an initial address, supplied in parallel in a single cycle, and initial access time to start reading boot code.
The AutoBoot feature allows the host memory controller to take boot code from the device immediately after the end of reset, without having to send a read command. This saves 32 or more cycles and simplifies the logic needed to initiate the reading of boot code.

- As part of the Power-up Reset, Hardware Reset, or Software Reset process the AutoBoot feature automatically starts a read access from a pre-specified address. At the time the reset process is completed, the device is ready to deliver code from the starting address. The host memory controller only needs to drive CE# signal from high to low and begin toggling the SCK signal. The device will delay code output for a pre-specified number of clock cycles before code streams out.

  – The Auto Boot Start Delay (ABSD) field of the AutoBoot register specifies the initial delay if any is needed by the host.
  – The host cannot send commands during this time.
  – If QE bit (Bit 6) in the Status Register is set to “1”, Fast Read Quad I/O operation will be selected and initial delay is the same as dummy cycles of Fast Read Quad I/O Read operation. If it is set to “0”, Fast Read operation will be applied and initial delay is the same as dummy cycles of Fast Read operation. Maximum operation frequency will be the same with Fast Read Operation.

- The starting address of the boot code is selected by the value programmed into the AutoBoot Start Address (ABSA) field of the AutoBoot Register.

  – Data will continuously shift out until CE# returns high.

- At any point after the first data byte is transferred, when CE# returns high, the SPI device will reset to standard SPI mode; able to accept normal command operations.

  – A minimum of one byte must be transferred.
  – AutoBoot mode will not initiate again until another power cycle or a reset occurs.

- An AutoBoot Enable bit (ABE) is set to enable the AutoBoot feature.

The AutoBoot register bits are non-volatile and provide:
- The starting address set by the AutoBoot Start Address (ABSA).
- The number of initial delay cycles, set by the AutoBoot Start Delay (ABSD) 4-bit count value.

Figure 8.102 AutoBoot Sequence (QE = 0)

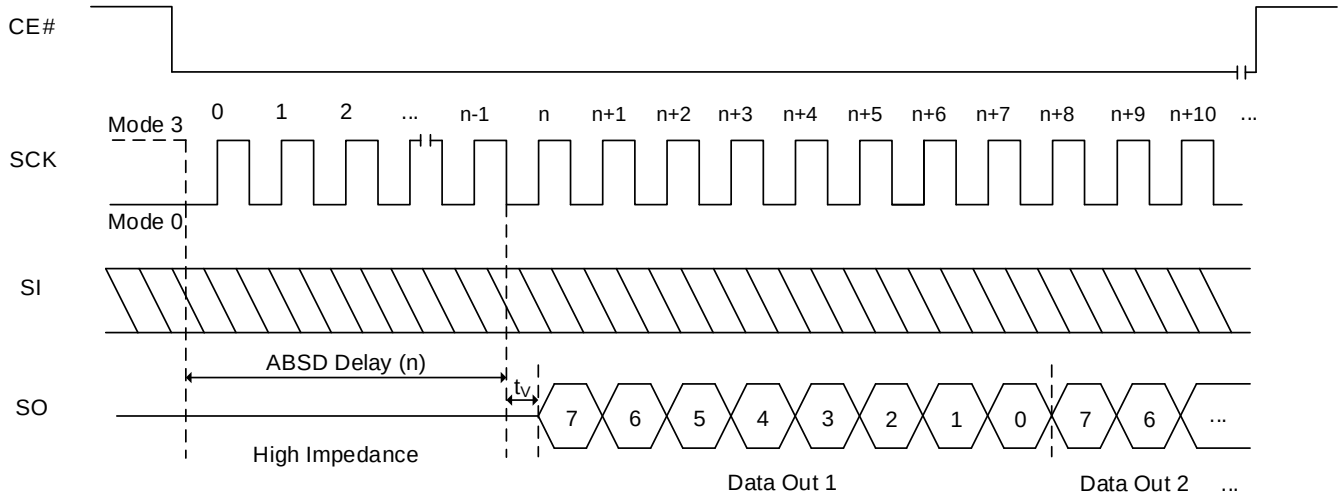
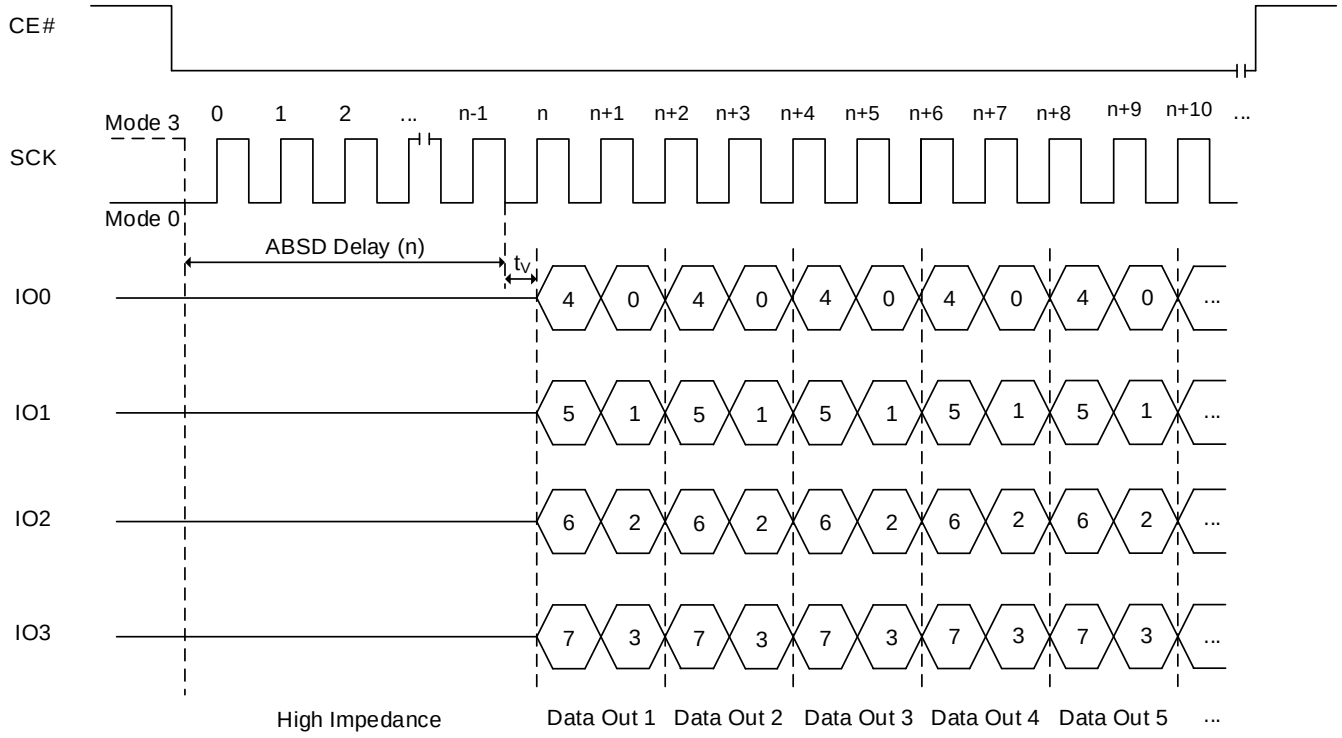


Figure 8.103 AutoBoot Sequence (QE = 1)



AUTOBOOT REGISTER READ OPERATION (RDABR, 14h)

The AutoBoot Register Read command is shifted in. Then the 32bit AutoBoot Register is shifted out, least significant byte first, most significant bit of each byte first. It is possible to read the AutoBoot Register continuously by providing multiples of 32bits.

Figure 8.104 RDABR Sequence In SPI Mode

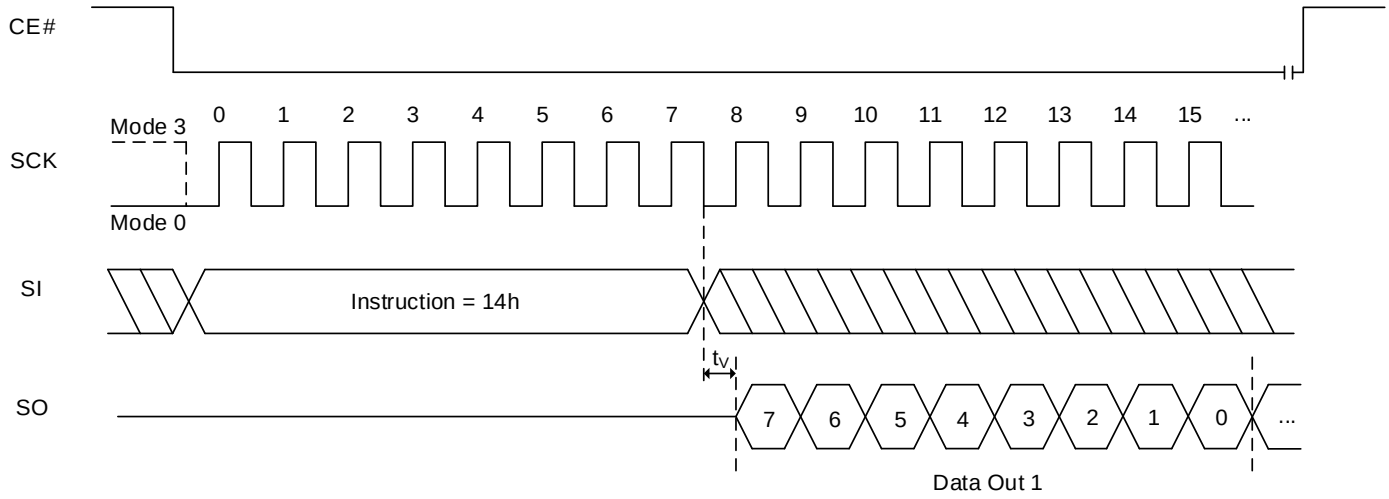
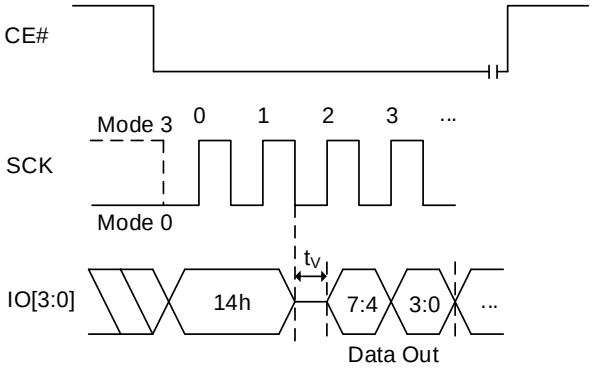


Figure 8.105 RDABR Sequence In QPI Mode



AUTOBOOT REGISTER WRITE OPERATION (WRABR, 15h)

Before the WRABR command can be accepted, a Write Enable (WREN) command must be issued and decoded by the device, which sets the Write Enable Latch (WEL) in the Status Register to enable any write operations.

The WRABR command is entered by shifting the instruction and the data bytes, least significant byte first, most significant bit of each byte first. The WRABR data is 32bits in length.

CE# must be driven high after the 32nd bit of data has been latched. If not, the WRABR command is not executed. As soon as CE# is driven high, the WRABR operation is initiated. While the WRABR operation is in progress, Status Register or Extended Read Register may be read to check the value of the Write In Progress (WIP) bit. The WIP bit is "1" during the WRABR operation, and is "0" when it is completed. When the WRABR cycle is completed, the WEL is set to "0".

Figure 8.106 WRABR Sequence In SPI Mode

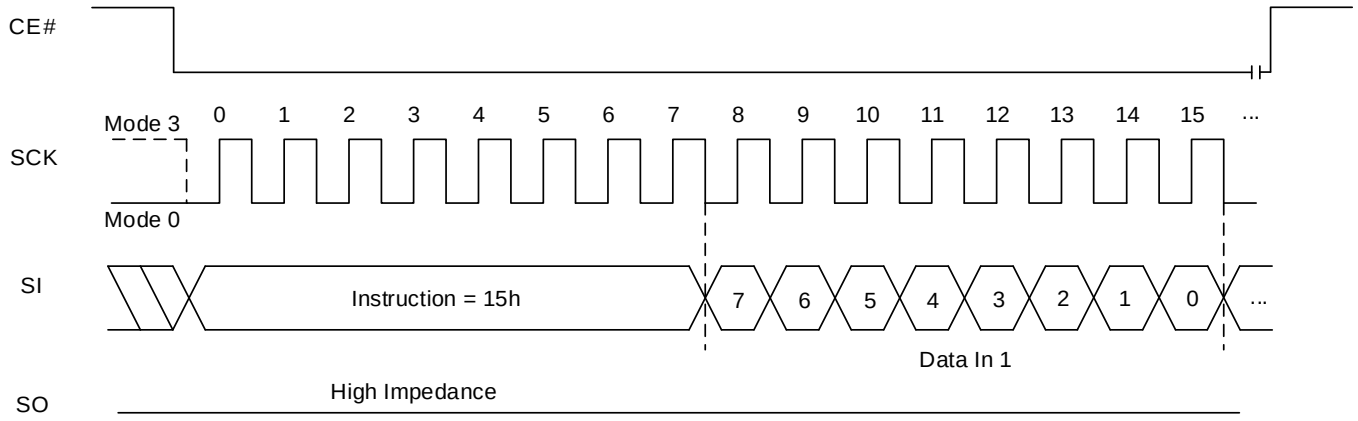
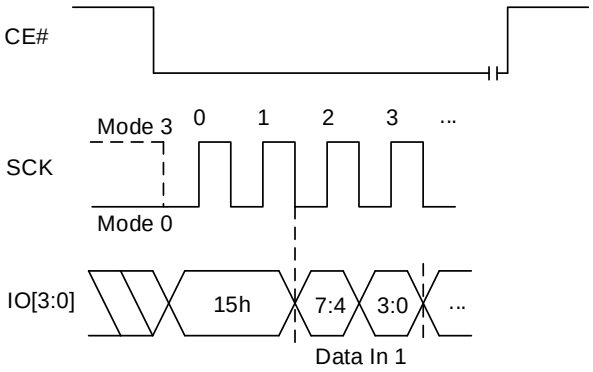


Figure 8.107 WRABR Sequence In QPI Mode



8.47 READ BANK ADDRESS REGISTER OPERATION (RDBR: 16h/C8h)

The Read Bank Address Register (RDBR) instruction allows the Bank Address Register contents to be read. RDBR is used to read only a volatile Bank Address Register.

The instruction code is first shifted in. Then the 8-bit Bank Register is shifted out. It is possible to read the Bank Address Register continuously by providing multiples of eight bits. The maximum operating clock frequency for the RDBR command is the same with Fast Read Operation.

Data is shifted in from SI and data is shifted out from SO in SPI sequence whereas data in and out is via four pins (IO0-IO3) in QPI sequence.

Figure 8.108 Read Bank Address Register Sequence In SPI Mode

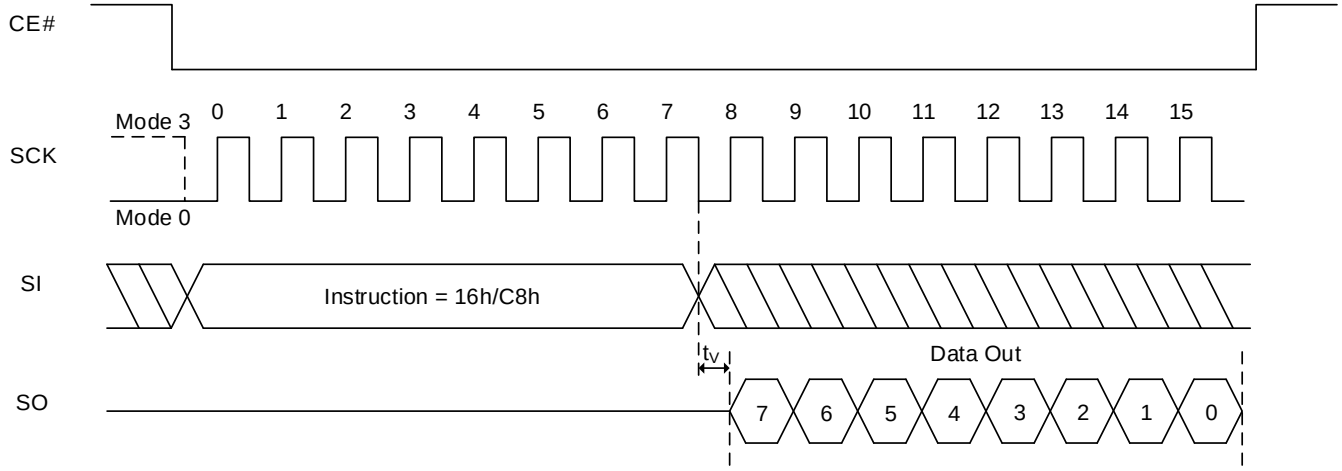
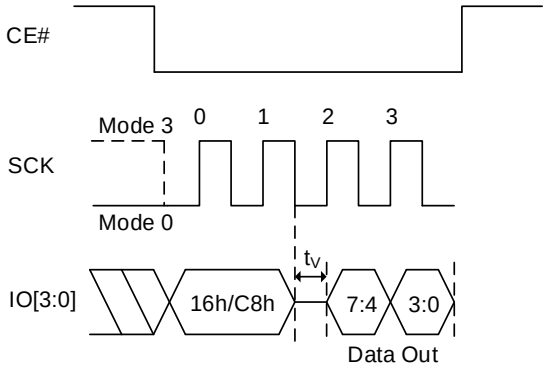


Figure 8.109 Read Bank Address Register Sequence In QPI Mode



8.48 WRITE BANK ADDRESS REGISTER OPERATION (WRBRNV: 18h, WRBRV: 17h/C5h)

The Write Bank Address Register (WRBRNV and WRBRV) instruction is used to write address bits above A23, into the Bank Address Register (BAR). WRBRNV is used to write the non-volatile Bank Address Register and WRBRV is used to write the volatile Bank Address Register. The instruction is also used to write the Extended Address Control bit (EXTADD) that is also in BAR[7]. BAR provides the high order addresses needed by devices having more than 128Mbits (16Mbytes), when using 3-byte address commands without extended addressing enabled (BAR[7] EXTADD = 0).

WRBRNV requires the Write Enable (WREN) command to precede it while WRBRV does not require WREN command.

The WRBRNV/WRBRV instruction is followed by the data byte. The Bank Address Register is one data byte in length. The Write In Progress (WIP) bit is "1" during WRBRNV operation, and is "0" when it is completed. The WRBRV command has no effect on the Write In Progress (WIP) bit. Any bank address bit reserved for the future should always be written as "0". Data is shifted in from SI and in SPI whereas data is shifted in via four pins (IO0-IO3) in QPI.

Note: When WRBRNV is executed, the volatile Bank Address Register is set as well as the non-volatile Bank Address Register.

Figure 8.110 Write Bank Address Register Sequence In SPI Mode

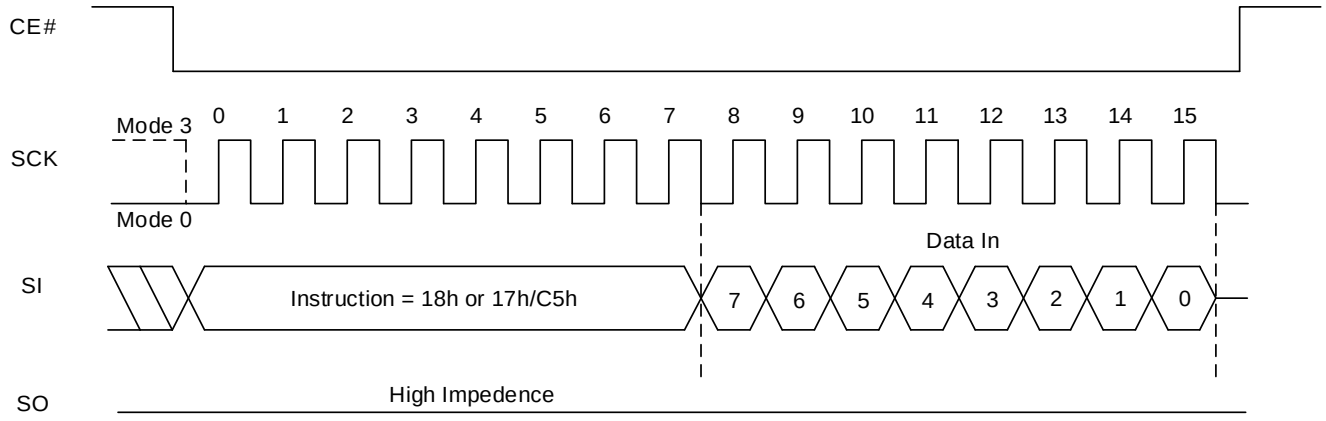
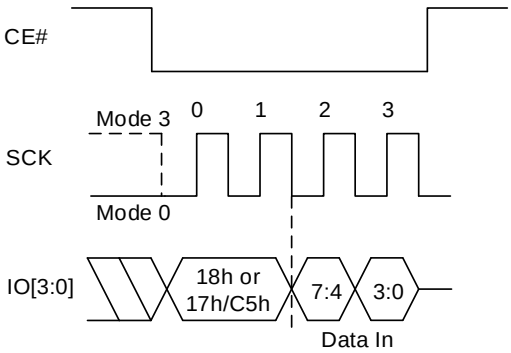


Figure 8.111 Write Bank Address Register Sequence In QPI Mode



8.49 ENTER 4-BYTE ADDRESS MODE OPERATION (EN4B, B7h)

The Enter 4-byte Address Mode instruction allows 32bit address (A31-A0) to be used to access the memory array beyond 128Mb. To execute EN4B operation, the host drives CE# low, sends the instruction code and then drives CE# high. The Exit 4-byte Address Mode instruction can be used to exit the 4-byte address mode.

Note: The EN4B instruction will set the Bit 7 (EXTADD) of the volatile Bank Address Register to "1", but will not change the non-volatile Bank Address Register.

Figure 8.112 Enter 4-byte Address Mode Sequence In SPI Mode

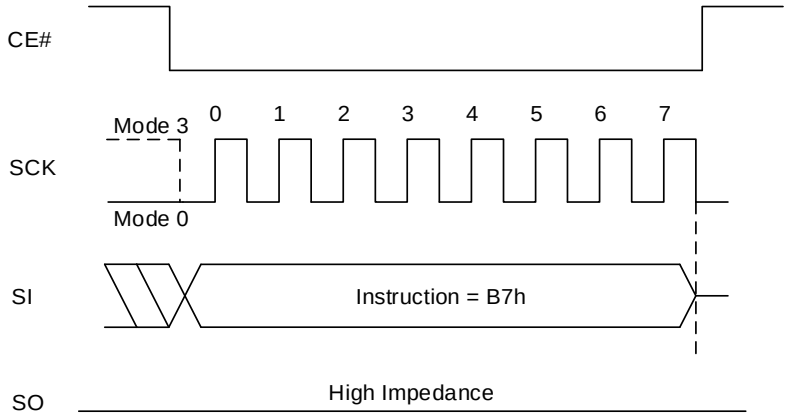
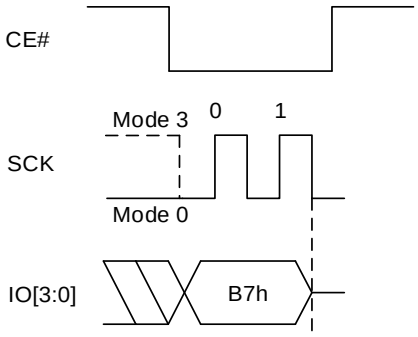


Figure 8.113 Enter 4-byte Address Mode Sequence In QPI Mode



8.50 EXIT 4-BYTE ADDRESS MODE OPERATION (EX4B, 29h)

In order to be backward compatible, the Exit 4-byte Address Mode instruction allows 24bit address (A23-A0) to be used to access the memory array up to 128Mb. The Bank Address Register must be used to access the memory array beyond 128Mb. To execute EX4B operation, the host drives CE# low, sends the instruction code and then drives CE# high.

Note: The EX4B instruction will reset the Bit 7 (EXTADD) of the volatile Bank Address Register to "0" ", but will not change the non-volatile Bank Address Register.

Figure 8.114 Exit 4-byte Address Mode Sequence In SPI Mode

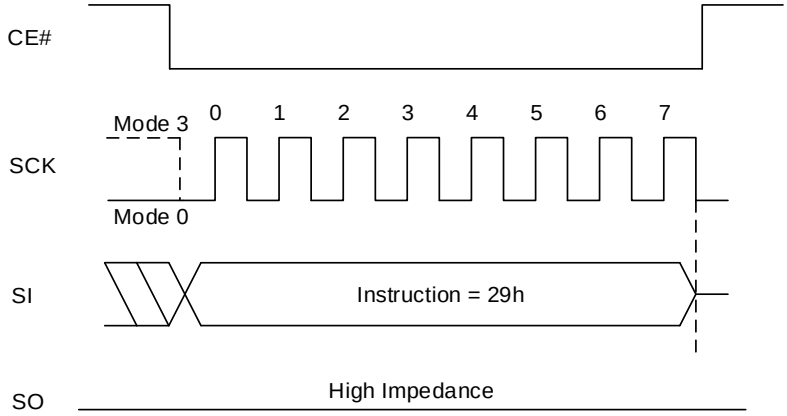
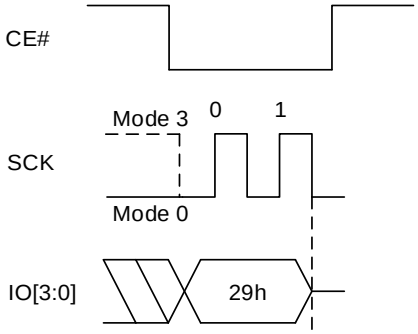


Figure 8.115 Exit 4-byte Address Mode Sequence In QPI Mode



8.51 READ DYB OPERATION (RDDYB, FAh or 4RDDYB, E0h)

FAh (EXTADD=0) is followed by a 3-byte address (A23-A0) or

- FAh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- E0h is followed by a 4-byte address (A31-A0)

The instruction is used to read Dynamic Protection Bit (DYB) status of the given sector/block. The instruction code is entered first, followed by the 24-bit or 32-bit address selecting location zero within the desired sector/block as above. Then the 8-bit DYB access register contents are shifted out. Each bit (SPI) or four bits (QPI) are shifted out at the SCK frequency by the falling edge of the SCK signal. It is possible to read the same DYB access register continuously by providing multiples of eight bits. The address of the DYB register does not increment so this is not a means to read the entire DYB array. Each location must be read with a separate Read DYB instruction. The maximum operating clock frequency for READ command is the same with Fast Read Operation.

Note: The high order address bits not used by 256M/128M must be zero. Data must be either 00h (protected) or FFh (unprotected).

Figure 8.116 Read DYB Sequence In SPI Mode (FAh [EXTADD=0], 3-byte address)

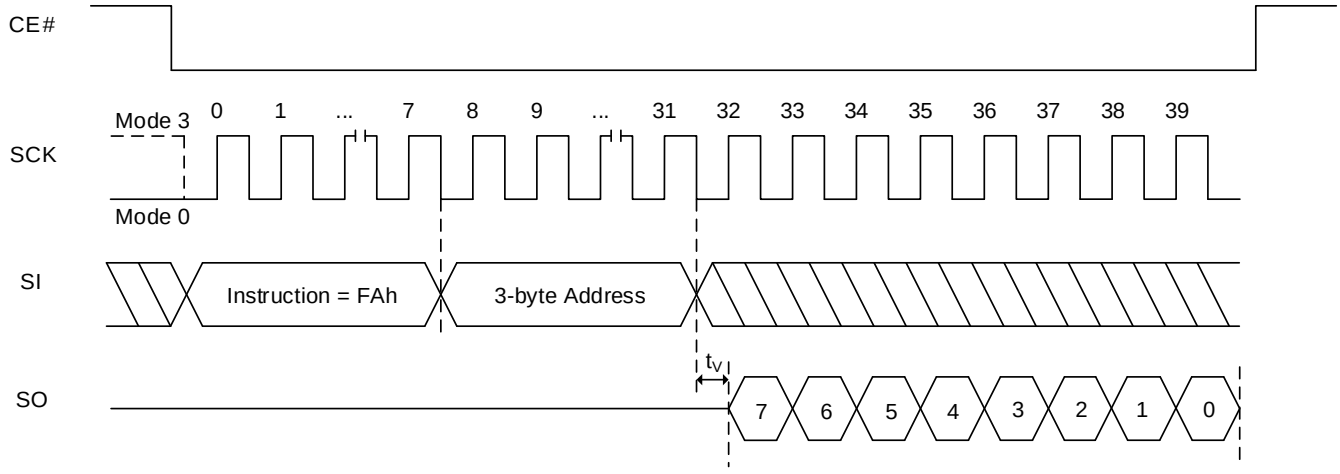


Figure 8.117 Read DYB Sequence In SPI Mode (FAh [EXTADD=1] or E0h, 4-byte address)

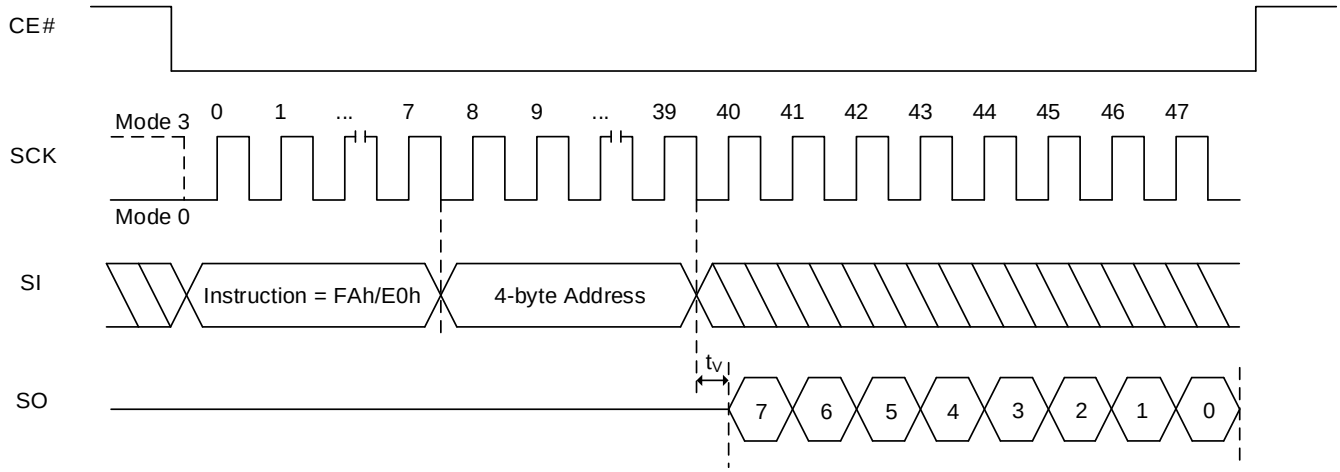


Figure 8.118 Read DYB Sequence In QPI Mode (FAh [EXTADD=0], 3-byte address)

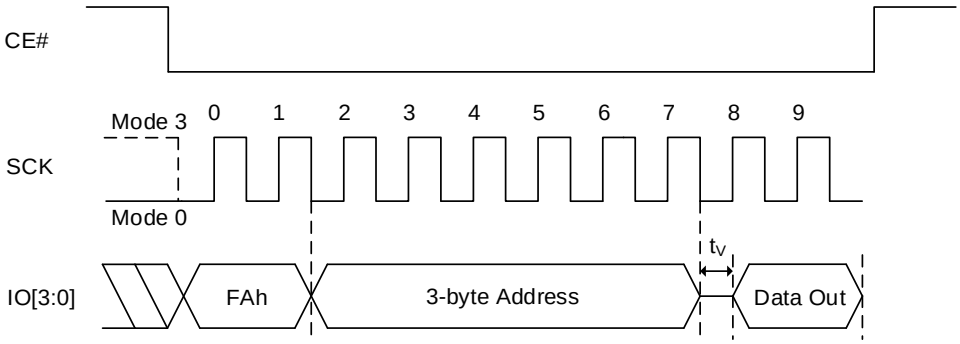
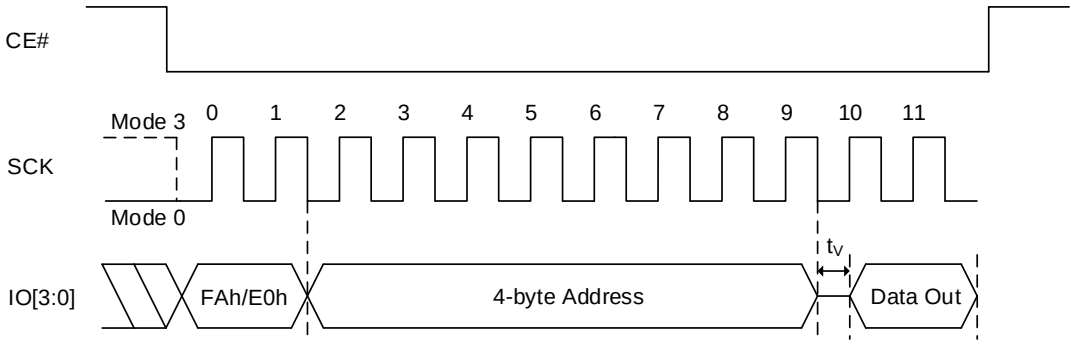


Figure 8.119 Read DYB Sequence In QPI Mode (FAh [EXTADD=0] or E0h, 4-byte address)



8.52 WRITE DYB OPERATION (WRDYB, FBh or 4WRDYB, E1h)

Before the Write DYB (WRDYB/4WRDYB) command can be accepted by the device, a Write Enable (WREN) command is not required.

- FBh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
- FBh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
- E1h is followed by a 4-byte address (A31-A0)

The WRDYB/4WRDYB command is entered by driving CE# low, followed by the instruction code, the 24-bit or 32-bit address selecting location zero within the desired sector/block as above, then the data byte. The DYB Access Register is one data byte in length.

CE# must be driven high after the eighth bit of data has been latched in. As soon as CE# is driven high, the WRDYB/4WRDYB operation is initiated.

Note: The high order address bits not used by 256M/128M must be zero. Data must be either 00h (protected) or FFh (unprotected).

Figure 8.120 Write DYB Sequence In SPI Mode (FBh [EXTADD=0], 3-byte address)

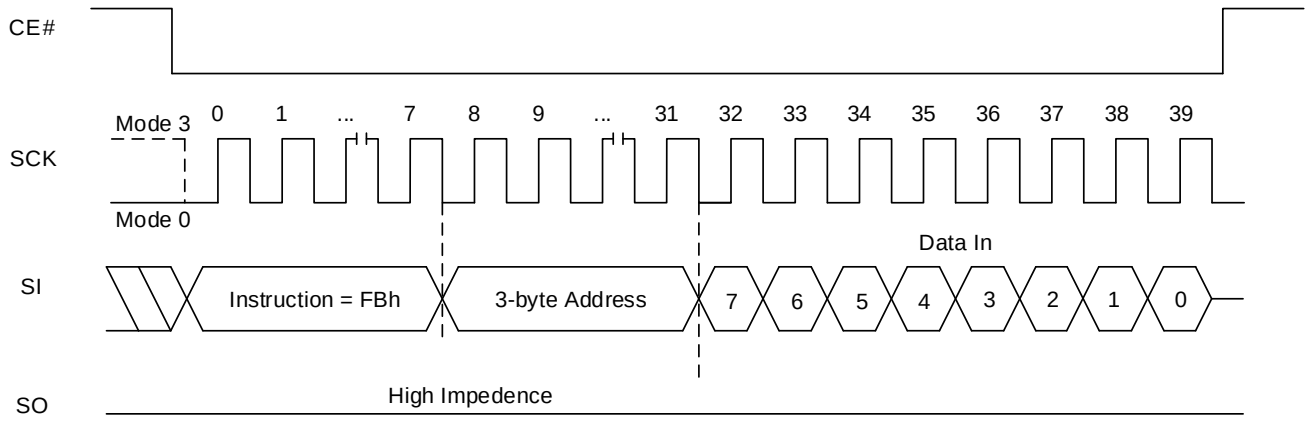


Figure 8.121 Write DYB Sequence In SPI Mode (FBh [EXTADD=1] or E1h, 4-byte address)

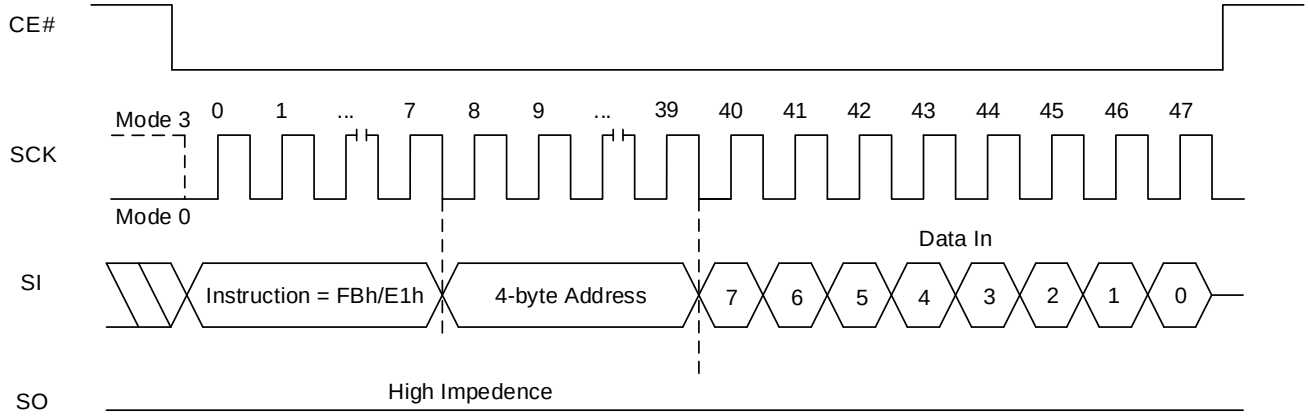


Figure 8.122 Write DYB Sequence In QPI Mode (FBh [EXTADD=0], 3-byte address)

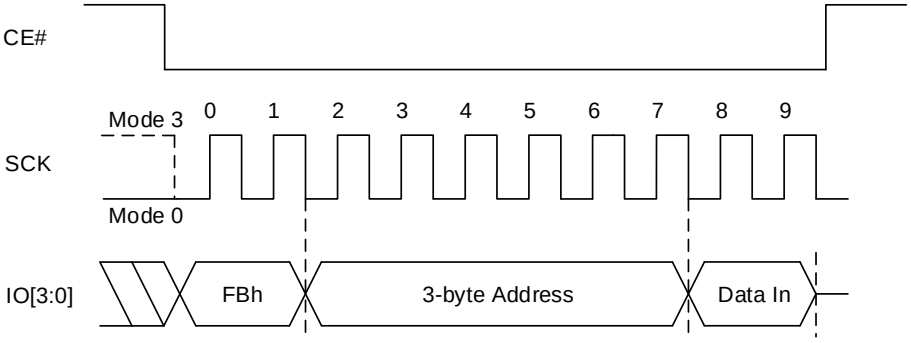
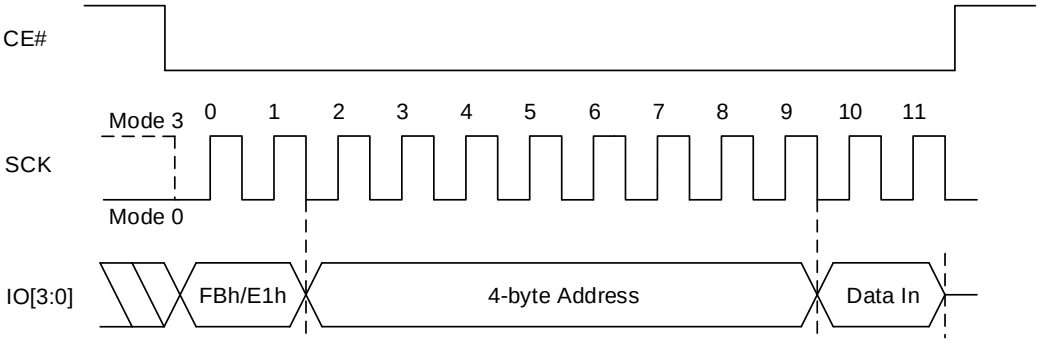


Figure 8.123 Write DYB Sequence In QPI Mode (FBh [EXTADD=1] or E1h, 4-byte address)



8.53 READ PPB OPERATION (RDPPB, FCh or 4RDPPB, E2h)

FCh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
FCh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
E2h is followed by a 4-byte address (A31-A0)

The instruction code is shifted into SI by the rising edges of the SCK signal, followed by the 24-bit or 32-bit address selecting location zero within the desired sector/block as above. Then the 8-bit PPB Access Register contents are shifted out on SO. The RDPPB/4RDPPB is supporting only SPI, not supporting QPI.

It is possible to read the same PPB Access Register continuously by providing multiples of eight bits. The address of the PPB Access Register does not increment so this is not a means to read the entire PPB array. Each location must be read with a separate Read PPB command. The maximum operating clock frequency for the Read PPB command is the same with Fast Read Operation.

Note: The high order address bits not used by 256M/128M must be zero. Data must be either 00h (protected) or FFh (unprotected).

Figure 8.124 Read PPB Sequence (FCh [EXTADD=0], 3-byte address)

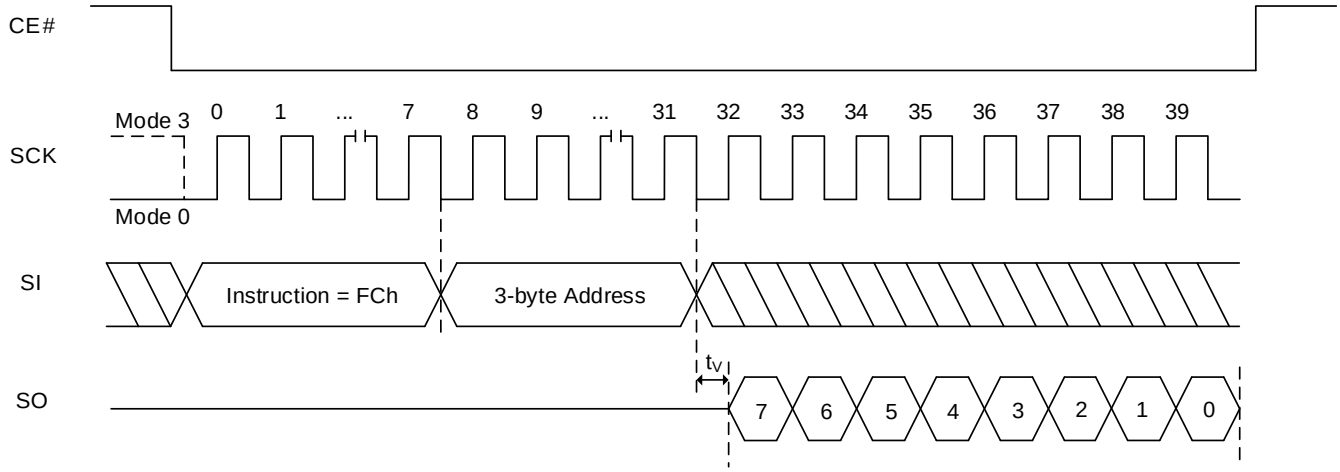
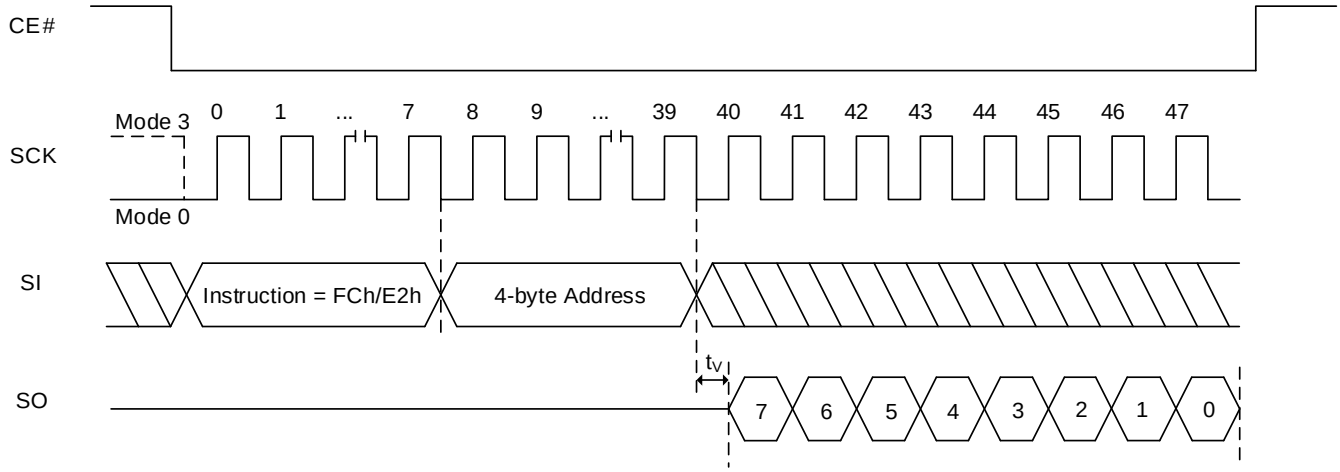


Figure 8.125 Read PPB Sequence (FCh [EXTADD=1] or E2h, 4-byte address)



8.54 PROGRAM PPB OPERATION (PGPPB, FDh or 4PGPPB, E3h)

Before the Program PPB (PGPPB/4PGPPB) command is sent, a Write Enable (WREN) command must be issued. After the WREN command has been decoded, the device will set the Write Enable Latch (WEL) in the Status Register.

FDh (EXTADD=0) is followed by a 3-byte address (A23-A0) or
FDh (EXTADD=1) is followed by a 4-byte address (A31-A0) or
E3h is followed by a 4-byte address (A31-A0)

The PGPPB/4PGPPB command is entered by driving CE# low, followed by the instruction code, followed by the 24-bit or 32-bit address selecting location zero within the desired sector/block as above.

The PGPPB/4PGPPB command affects the WIP bit in the same manner as any other programming operation. CE# must be driven high after the last bit of address has been latched in. As soon as CE# is driven high, the PGPPB/4PGPPB operation is initiated. While the PGPPB/4PGPPB operation is in progress, the Status Register or Extended Read Register may be read to check the value of the Write In Progress (WIP) bit. The WIP bit is "1" during the PGPPB/4PGPPB operation, and is "0" when it is completed. When the PGPPB/4PGPPB operation is completed, the WEL is set to "0".

Note: The high order address bits not used by 256M/128M must be zero. Data must be either 00h (protected) or FFh (unprotected).

Figure 8.126 Program PPB Sequence In SPI Mode (FDh [EXTADD=0], 3-byte address)

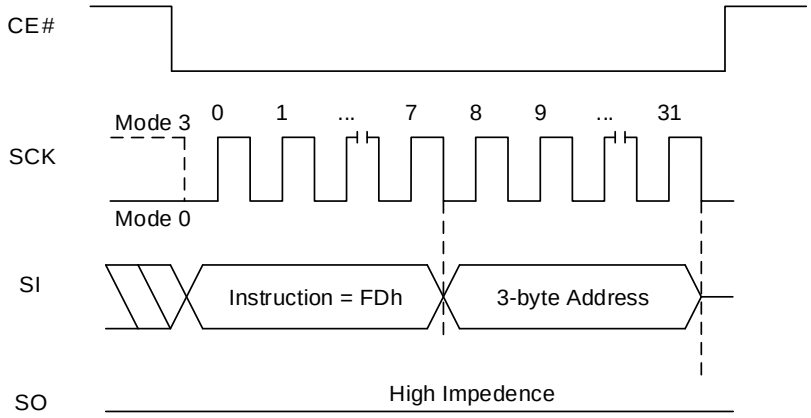


Figure 8.127 Program PPB Sequence In SPI Mode (FDh [EXTADD=1] or E3h, 4-byte address)

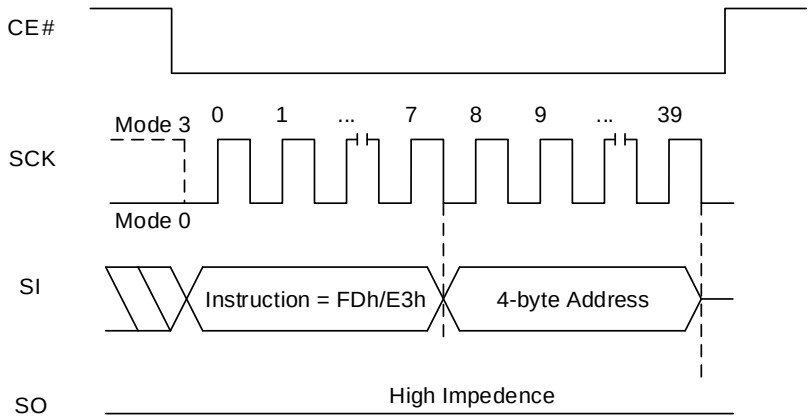


Figure 8.128 Program PPB Sequence In QPI Mode (FDh [EXTADD=0], 3-byte address)

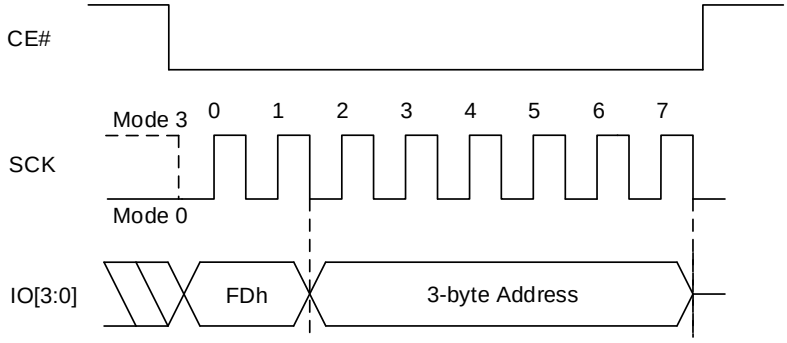
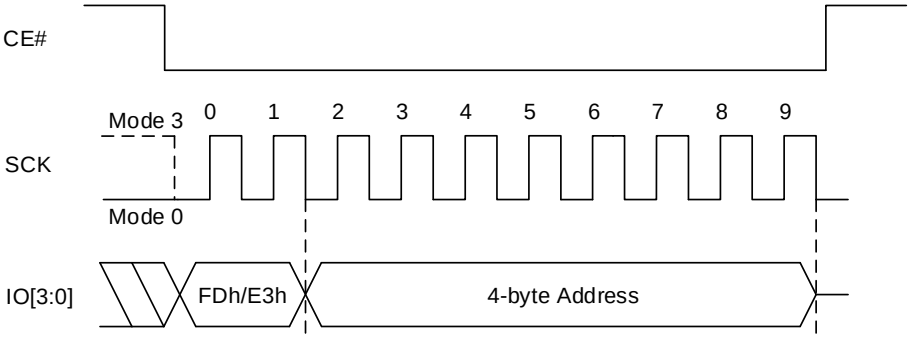


Figure 8.129 Program PPB Sequence In QPI Mode (FDh [EXTADD=1] or E3h, 4-byte address)



8.55 ERASE PPB OPERATION (ERPPB, E4h)

The Erase PPB (ERPPB) command sets all PPB bits to “1”. Before the ERPPB command can be accepted by the device, a Write Enable (WREN) command must be issued and decoded by the device, which sets the Write Enable Latch (WEL) in the Status Register to enable any write operations.

The instruction code is shifted in by the rising edges of the SCK signal. CE# must be driven high after the eighth bit of the instruction byte has been latched in. This will initiate the beginning of internal erase cycle, which involves the pre-programming and erase of the entire PPB memory array. Without CE# being driven high after the eighth bit of the instruction, the PPB erase operation will not be executed.

With the internal erase cycle in progress, the user can read the value of the Write In Progress (WIP) bit to check if the operation has been completed. The WIP bit will indicate “1” when the erase cycle is in progress and “0” when the erase cycle has been completed. When the ERPPB operation is completed, the WEL is set to “0”. Erase suspend is not allowed during PPB Erase.

Figure 8.130 Erase PPB Sequence In SPI Mode

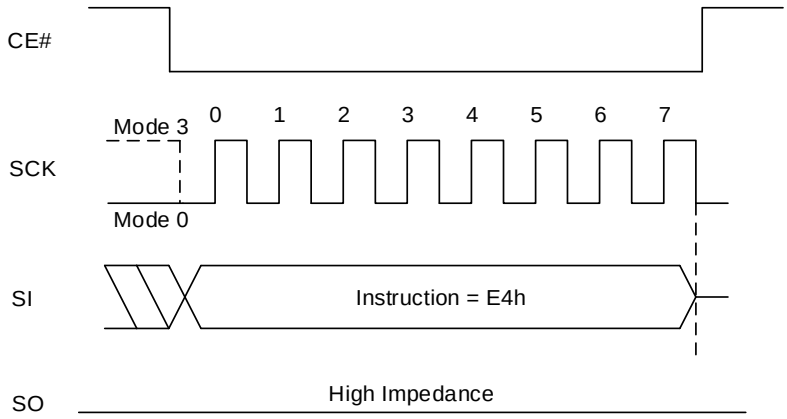
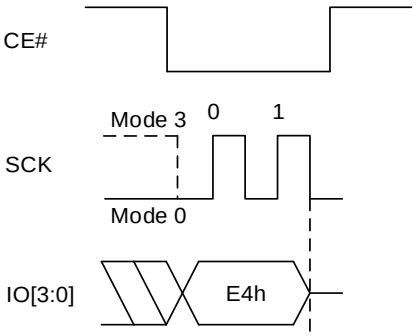


Figure 8.131 Erase PPB Sequence In QPI Mode



8.56 READ ASP OPERATION (RDASP, 2Bh)

The RDASP instruction code is shifted in by the rising edge of the SCK signal. Then the 16-bit ASP register contents is shifted out, least significant byte first, most significant bit of each byte first. Each bit is shifted out at the SCK frequency by the falling edge of the SCK signal. It is possible to read the ASP register continuously by providing multiples of 16 bits. The maximum operating clock frequency for the RDASP command is the same with Fast Read Operation.

Figure 8.132 Read ASP Sequence In SPI Mode

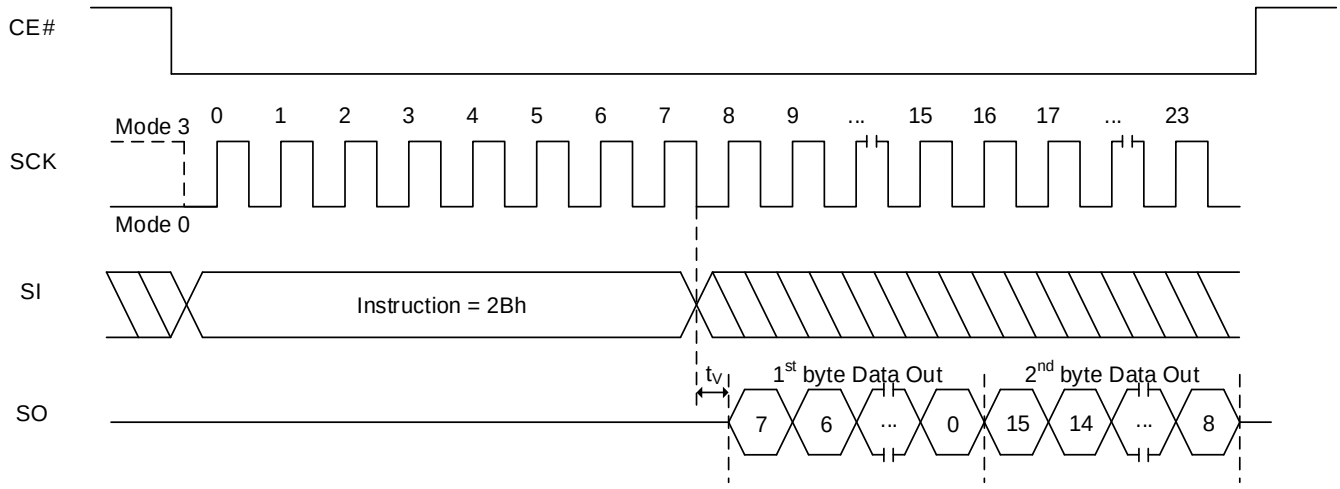
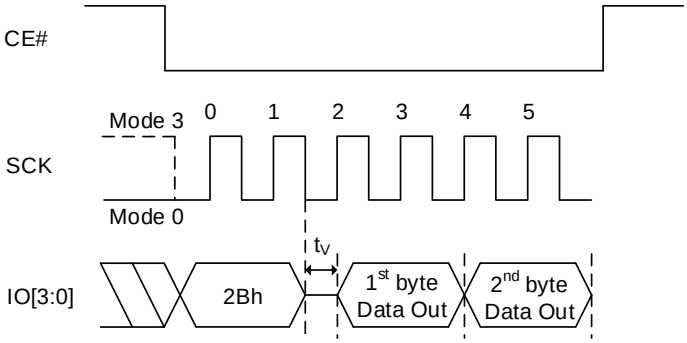


Figure 8.133 Read ASP Sequence In QPI Mode



8.57 PROGRAM ASP OPERATION (PGASP, 2Fh)

Before the Program ASP (PGASP) command can be accepted by the device, a Write Enable (WREN) command must be issued. After the WREN command has been decoded, the device will set the Write Enable Latch (WEL) in the Status Register to enable any write operations.

The PGASP command is entered by driving CE# low, followed by the instruction code and two data bytes, least significant byte first, most significant bit of each byte first. The ASP Register is two data bytes in length. The PGASP command affects the Write In Progress (WIP) bit in the same manner as any other programming operation.

CE# input must be driven high after the sixteenth bit of data has been latched in. If not, the PGASP command is not executed. As soon as CE# is driven high, the PGASP operation is initiated. While the PGASP operation is in progress, the Status Register or the Extended Read Register may be read to check the value of WIP bit. The WIP bit is "1" during the PGASP operation, and is "0" when it is completed. When the PGASP operation is completed, the WEL is set to "0".

Figure 8.134 Program ASP Sequence In SPI Mode

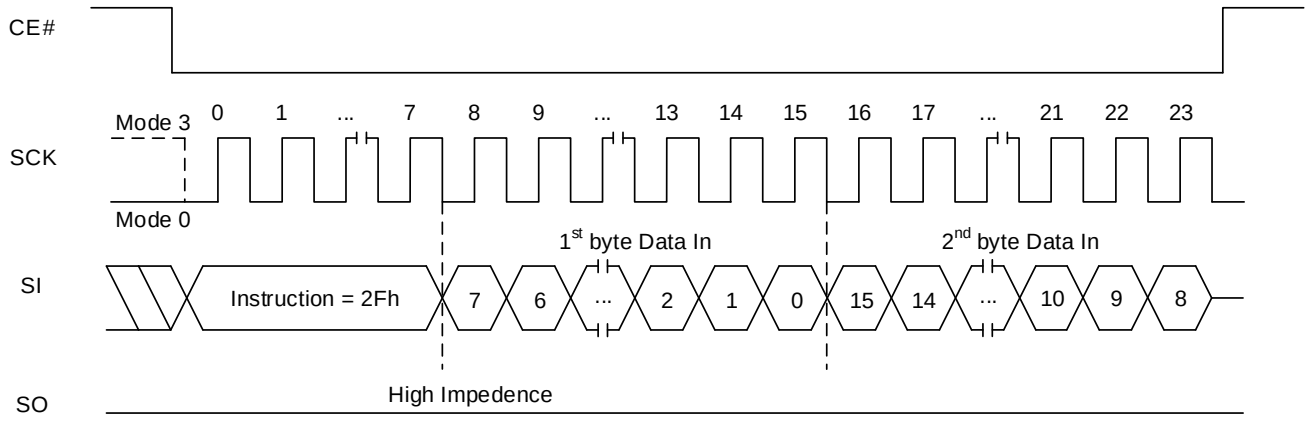
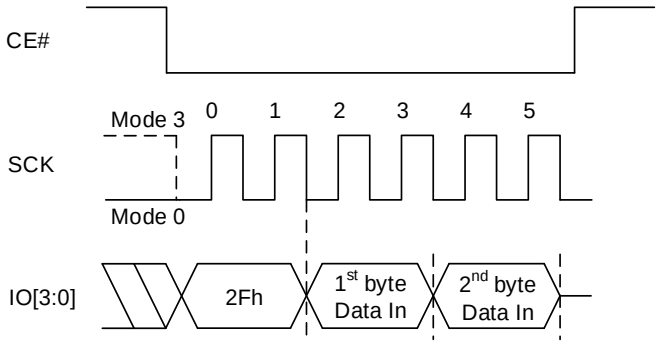


Figure 8.135 Program ASP Sequence In QPI Mode



8.58 READ PPB LOCK BIT OPERATION (RDPLB, A7h)

The Read PPB Lock Bit (RDPLB) command allows the PPB Lock Register contents to be read. It is possible to read the PPB Lock Register continuously by providing multiples of eight bits. The PPB Lock Register contents may only be read when the device is in standby state with no other operation in progress. It is recommended to check the Write In Progress (WIP) bit before issuing a new command to the device.

Figure 8.136 Read PPB Lock Bit Sequence In SPI Mode

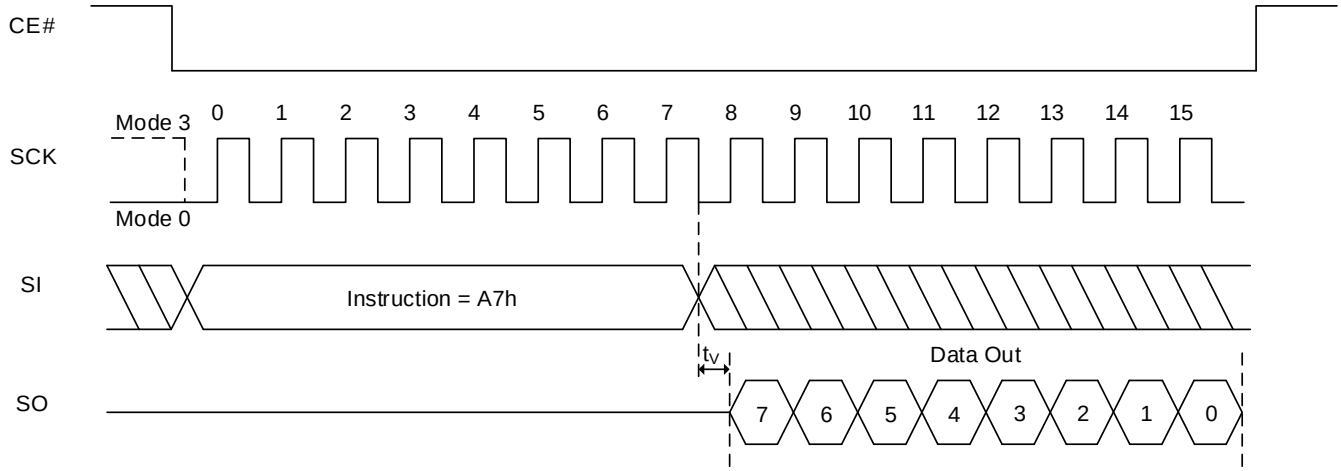
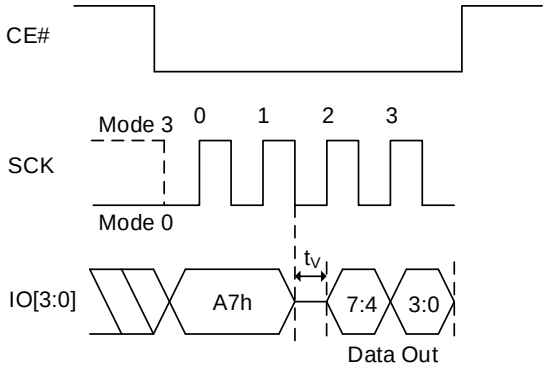


Figure 8.137 Read PPB Lock Bit Sequence In QPI Mode



8.59 WRITE PPB LOCK BIT OPERATION (WRPLB, A6h)

The Write PPB Lock Bit (WRPLB) command clears the PPB Lock (PPBLK) bit to zero. Before the WRPLB command can be accepted by the device, a Write Enable (WREN) command must be issued and decoded by the device, which sets the Write Enable Latch (WEL) in the Status Register to enable any write operations.

The WRPLB command is entered by driving CE# low, followed by the instruction code. CE# must be driven high after the eighth bit of instruction has been latched in. If not, the WRPLB command is not executed. As soon as CE# is driven high, the WRPLB operation is initiated. While the WRPLB operation is in progress, the Status Register or Extended Read Register may still be read to check the value of the Write In Progress (WIP) bit. The WIP bit is "1" during the WRPLB operation, and is "0" when it is completed. When the WRPLB operation is completed, the WEL is set to "0". The maximum clock frequency for the WRPLB command is the same with Fast Read Operation.

Figure 8.138 Write PPB Lock Bit Sequence In SPI Mode

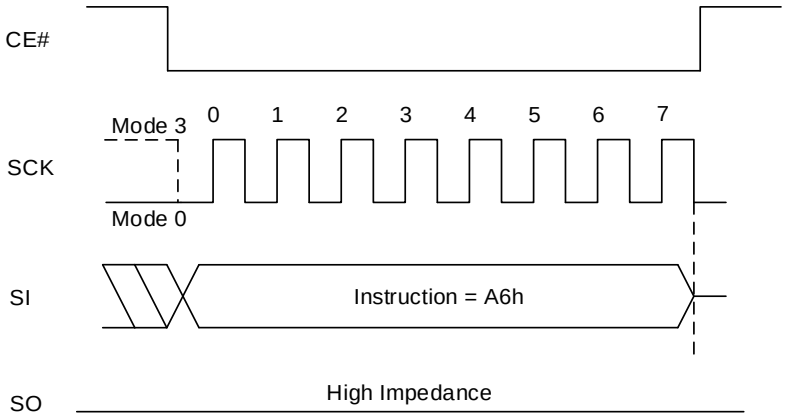
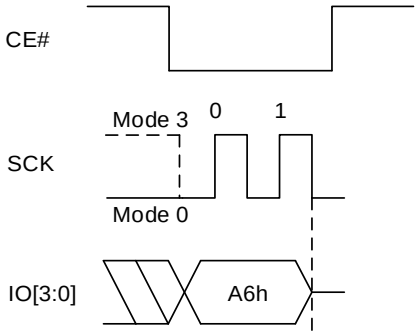


Figure 8.139 Write PPB Lock Bit Sequence In QPI Mode



8.60 SET FREEZE BIT OPERATION (SFRZ, 91h)

The Set FREEZE Bit (SFRZ) command sets FREEZE (PPB Lock Register bit7) to one. Please refer to the section 6.6.3 PPB Lock Register for more detail. Before the SFRZ command can be accepted by the device, a Write Enable (WREN) command must be issued and decoded by the device, which sets the Write Enable Latch (WEL) in the Status Register to enable any write operations.

The SFRZ command is entered by driving CE# low, followed by the instruction code. CE# must be driven high after the eighth bit of instruction has been latched in. If not, the SFRZ command is not executed. As soon as CE# is driven high, the SFRZ operation is initiated. While the SFRZ operation is in progress, the Status Register or Extended Read Register may still be read to check the value of the Write In Progress (WIP) bit. The WIP bit is "1" during the SFRZ operation, and is "0" when it is completed. When the SFRZ operation is completed, the WEL is set to "0". The maximum clock frequency for the SFRZ command is the same with Fast Read Operation.

Figure 8.140 Set FREEZE Bit Sequence In SPI Mode

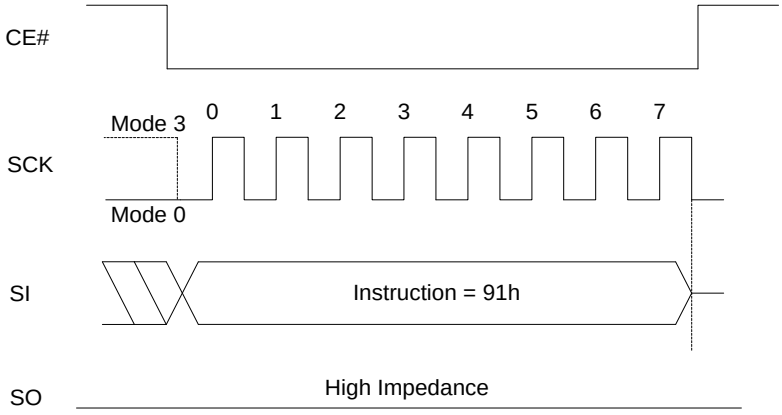
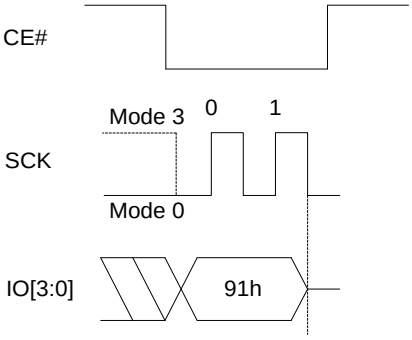


Figure 8.141 Set FREEZE Bit Sequence In QPI Mode



8.61 READ PASSWORD OPERATION (RDPWD, E7h)

The correct password value may be read only after it is programmed and before the Password Mode has been selected by programming the Password Protection Mode bit to "0" in the ASP Register (ASP[2]). After the Password Protection Mode is selected the RDPWD command is ignored.

The RDPWD command is shifted in. Then the 64-bit Password is shifted out, least significant byte first, most significant bit of each byte first. Each bit is shifted out at the SCK frequency by the falling edge of the SCK signal. It is possible to read the Password continuously by providing multiples of 64bits. The maximum operating clock frequency for the RDPWD command is the same with Fast Read Operation.

Figure 8.142 Read password Sequence In SPI Mode

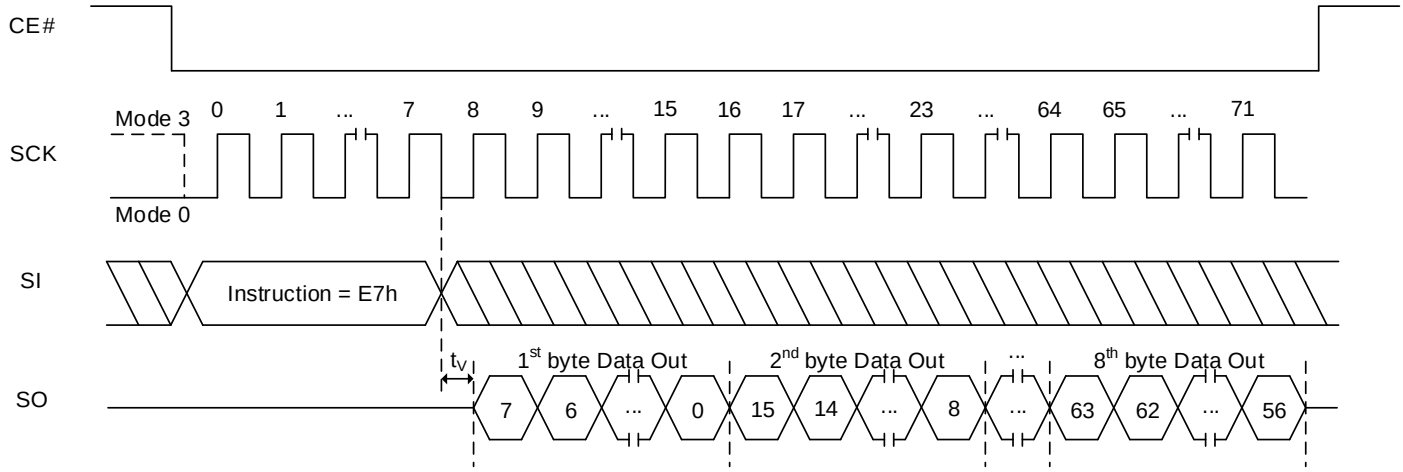
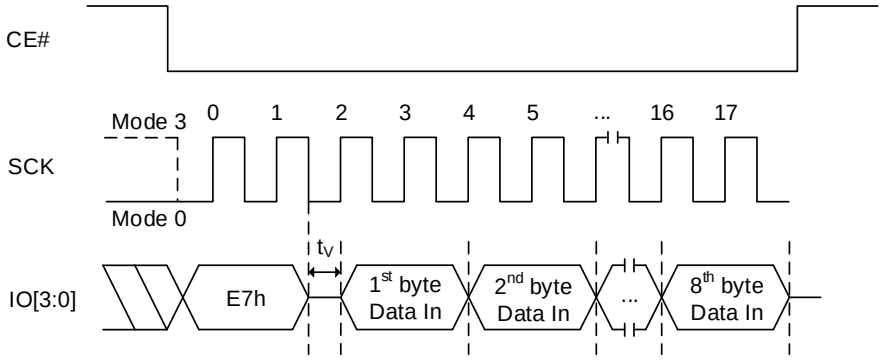


Figure 8.143 Read Password Sequence In QPI Mode



8.62 PROGRAM PASSWORD OPERATION (PGPWD, E8h)

Before the Program Password (PGPWD) command can be accepted by the device, a Write Enable (WREN) command must be issued and decoded by the device which sets the Write Enable Latch (WEL) to enable the PGPWD operation. The password can only be programmed before the Password Mode is selected by programming the Password Protection Mode bit to "0" in the ASP Register (ASP[2]). After the Password Protection Mode is selected the PGPWD command is ignored.

The PGPWD command is entered by driving CE# low, followed by the instruction code and the password data bytes, least significant byte first, most significant bit of each byte first. The password is 64bits in length.

CE# must be driven high after the 64th bit of data has been latched. If not, the PGPWD command is not executed. As soon as CE# is driven high, the PGPWD operation is initiated. While the PGPWD operation is in progress, the Status Register or Extended Read Register may be read to check the value of the Write In Progress (WIP) bit. The WIP bit is "1" during the PGPWD operation, and is "0" when it is completed. When the PGPWD operation is completed, the Write Enable Latch (WEL) is set to "0". The maximum clock frequency for the PGPWD command is the same with Fast Read Operation.

Figure 8.144 Program Password Sequence In SPI Mode

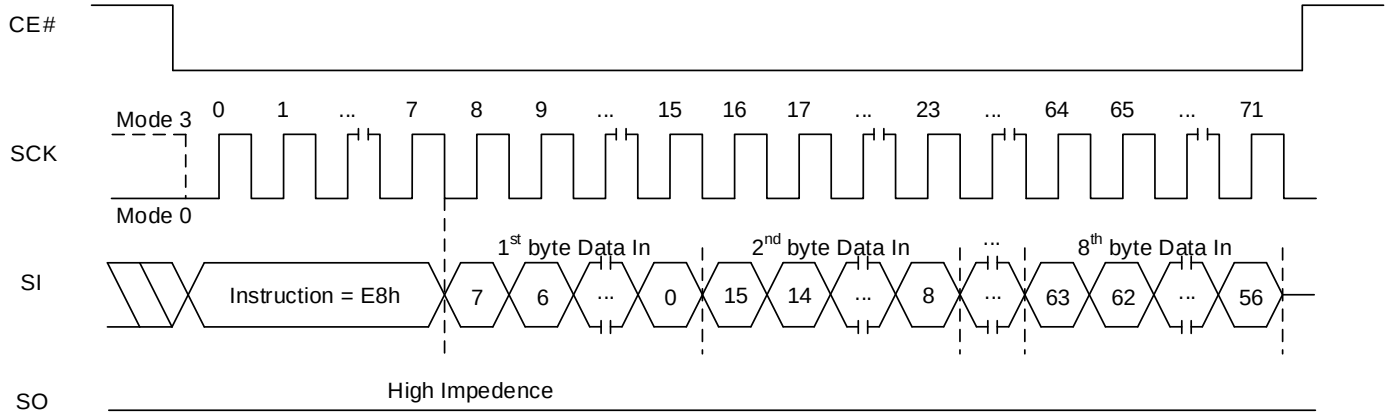
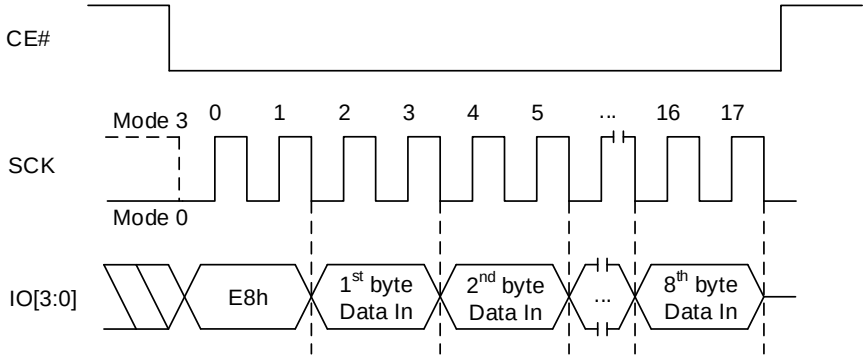


Figure 8.145 Program Password Sequence In QPI Mode



8.63 UNLOCK PASSWORD OPERATION (UNPWD, E9h)

The UNPWD command is entered by driving CE# low, followed by the instruction code and the password data bytes, least significant byte first, most significant bit of each byte first. The password is 64bits in length.

CE# must be driven high after the 64th bit of data has been latched. If not, the UNPWD command is not executed. As soon as CE# is driven high, the UNPWD operation is initiated. While the UNPWD operation is in progress, the Status Register or Extended Read Register may be read to check the value of the Write In Progress (WIP) bit. The WIP bit is "1" during the UNPWD operation, and is "0" when it is completed.

If the UNPWD command supplied password does not match the hidden password in the Password Register, the UNPWD command is ignored. This returns the device to standby state, ready for a new command such as a retry of the UNPWD command. If the password does match, the PPB Lock bit is set to "1". The maximum clock frequency for the UNPWD command is the same with Fast Read Operation.

Figure 8.146 Unlock Password Sequence In SPI Mode

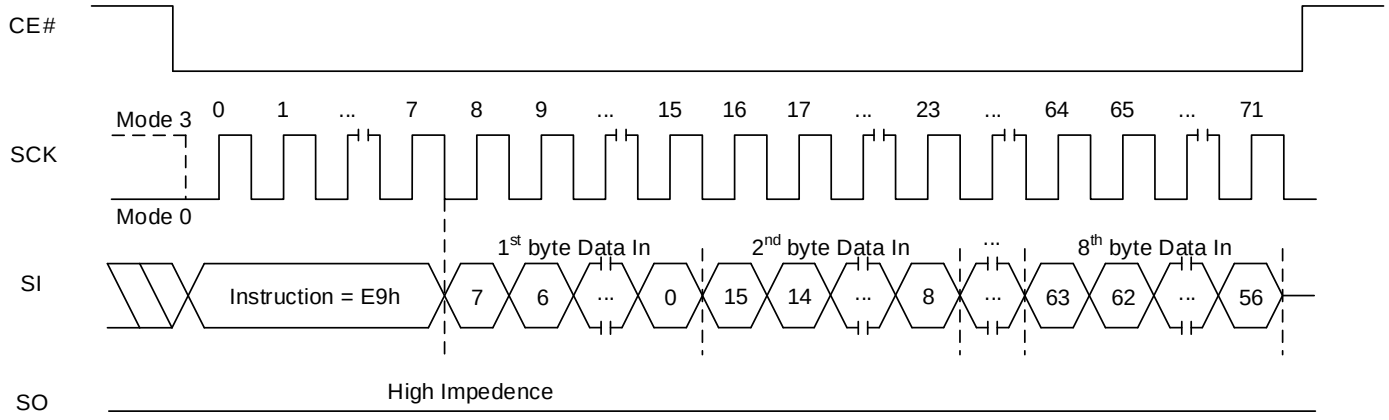
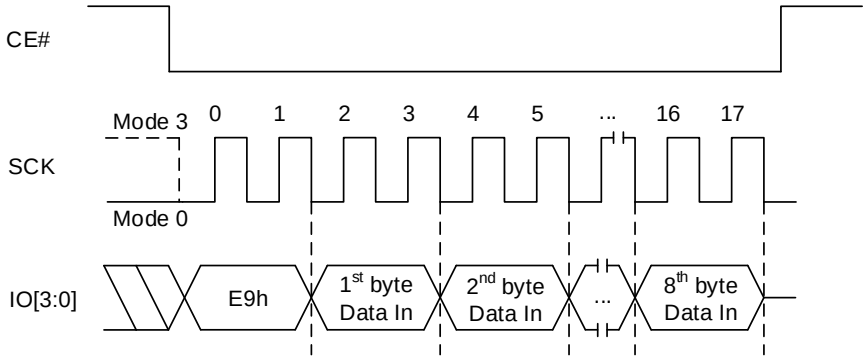


Figure 8.147 Unlock Password Sequence In SPI Mode



8.64 GANG SECTOR/BLOCK LOCK OPERATION (GBLK, 7Eh)

The Gang Sector/Block Lock (GBLK) instruction provides a quick method to set all DYB (Dynamic Protection Bit) bits to “0” at once.

The sequence of issuing GBLK instruction is: drive CE# low → send GBLK instruction code → drive CE# high. The instruction code will be shifted into the device on the rising edge of SCK.

The GBLK command is accepted in both SPI and QPI mode. The CE# must go high exactly at the byte boundary, otherwise, the instruction will be ignored. While the GBLK operation is in progress, the Status Register or Extended Read Register may be read to check the value of the Write In Progress (WIP) bit. The WIP bit is “1” during the GBLK operation, and is “0” when it is completed.

Figure 8.148 Gang Sector/Block Lock Sequence In SPI Mode

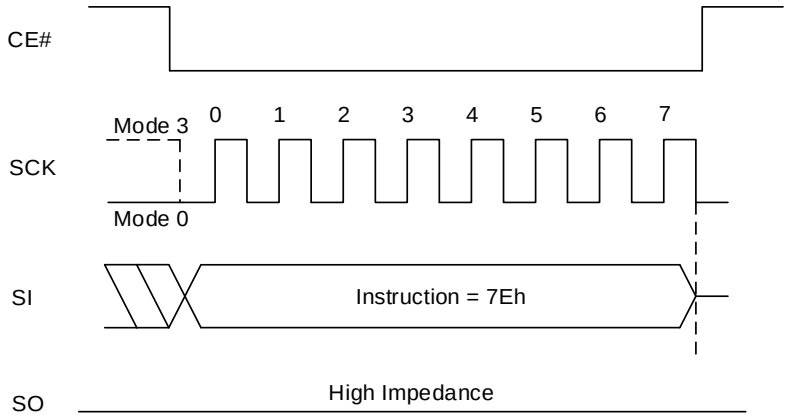
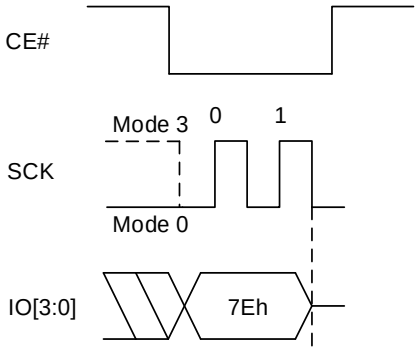


Figure 8.149 Gang Sector/Block Lock Sequence In QPI Mode



8.65 GANG SECTOR/BLOCK UNLOCK OPERATION (GBUN, 98h)

The Gang Sector/Block Unlock (GBUN) instruction provides a quick method to clear all DYB (Dynamic Protection Bit) bits to “1” at once.

The sequence of issuing GBUN instruction is: drive CE# low → send GBUN instruction code → drive CE# high. The instruction code will be shifted into the device on the rising edge of SCK.

The GBUN command is accepted in both SPI and QPI mode. The CE# must go high exactly at the byte boundary, otherwise, the instruction will be ignored and not be executed. While the GBUN operation is in progress, the Status Register or Extended Read Register may be read to check the value of the Write In Progress (WIP) bit. The WIP bit is “1” during the GBUN operation, and is “0” when it is completed.

Figure 8.150 Gang Sector/Block Unlock Sequence In SPI Mode

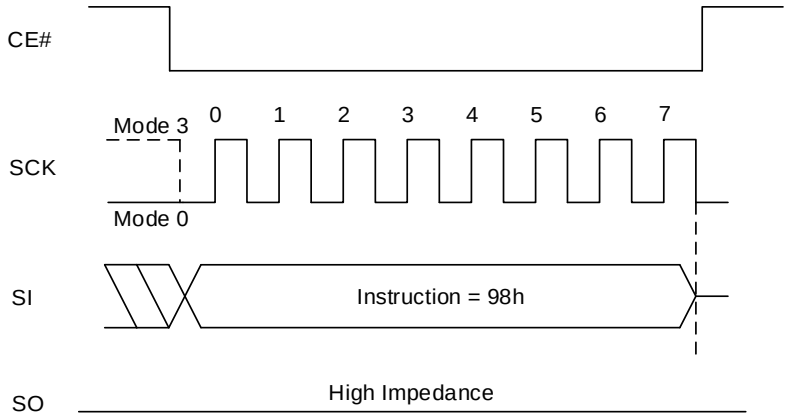
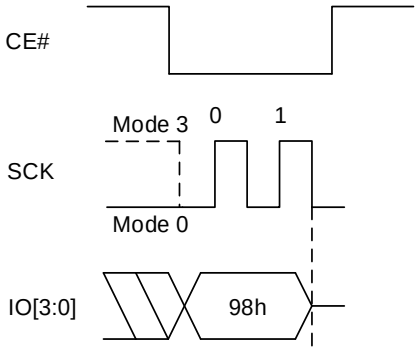


Figure 8.151 Gang Sector/Block Unlock Sequence In QPI Mode



9. ELECTRICAL CHARACTERISTICS

9.1 ABSOLUTE MAXIMUM RATINGS (1)

Storage Temperature		-65°C to +150°C
Surface Mount Lead Soldering Temperature	Standard Package	240°C 3 Seconds
	Lead-free Package	260°C 3 Seconds
Input Voltage with Respect to Ground on All Pins		-0.5V to V_{CC} + 0.5V
All Output Voltage with Respect to Ground		-0.5V to V_{CC} + 0.5V
V_{CC}	IS25LP	-0.5V to +6.0V
	IS25WP	-0.5V to +2.5V

Note:

1. **Applied conditions greater than those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.**

9.2 OPERATING RANGE

Operating Temperature	Extended Grade E	-40°C to 105°C
	Automotive Grade A3	-40°C to 125°C
V_{CC} Power Supply	IS25LP	2.7V (VMIN) – 3.6V (VMAX); 3.0V (Typ), Max 166MHz (1)
		2.3V (VMIN) – 3.6V (VMAX); 3.0V (Typ), Max 133MHz
	IS25WP	1.65V (VMIN) –1.95V (VMAX); 1.8V (Typ), Max 104MHz

Note:

1, Max frequency is 133MHz at Mode 3

9.3 DC CHARACTERISTICS

(Under operating range)

Symbol	Parameter		Condition		Min	Typ(2)	Max	Units
I_{CC1}	V_{CC} Active Read current(3)		NORD at 80MHz			7.5	9	mA
			FRD Single at 166MHz			9.5	11	
			FRD Dual at 166MHz			9	12	
			FRD Quad at 166MHz			9	15	
			FRD Single at 133MHz			9	10.5	
			FRD Dual at 133MHz			8	11	
			FRD Quad at 133MHz			10	13.5	
			FRD Quad at 83MHz			9	11	
			FRD Quad at 104MHz			10	12	
			FRD Single DTR at 80MHz			9	11	
			FRD Dual DTR at 80MHz			10	12	
			FRD Quad DTR at 80MHz			11	13	
I_{CC2}	V_{CC} Program Current		CE# = V_{CC}	85°C		25	30	mA
				105°C			30	
				125°C			30	
I_{CC3}	V_{CC} WRSR Current		CE# = V_{CC}	85°C		25	30	
				105°C			30	
				125°C			30	
I_{CC4}	V_{CC} Erase Current (SER/4SER/BER32/4BER32/ BER64/4BER64)		CE# = V_{CC}	85°C		25	30	
				105°C			30	
				125°C			30	
I_{CC5}	V_{CC} Erase Current (CE)		CE# = V_{CC}	85°C		25	30	
				105°C			30	
				125°C			30	
I_{SB1}	V_{CC} Standby Current CMOS	IS25LP	CE# = V_{CC}, CE#, RESET#(4) = V_{CC}	85°C		10	60 (6)	µA
				105°C			90 (6)	
				125°C			140	
		IS25WP		85°C		6	60 (6)	
				105°C			90 (6)	
				125°C			140	
I_{SB2}	Deep power down current	IS25LP	CE# = V_{CC}, CE#, RESET#(4) = V_{CC}	85°C		7.5	20 (6)	µA
				105°C			30 (6)	
				125°C			50	
		IS25WP	CE# = V_{CC}, CE#, RESET#(4) = V_{CC}	85°C		1	20 (6)	
				105°C			30 (6)	
				125°C			50	
I_{LI}	Input Leakage Current		V_{IN} = 0V to V_{CC}				±1(5)	µA
I_{LO}	Output Leakage Current		V_{IN} = 0V to V_{CC}				±1(5)	µA
V_{IL}(1)	Input Low Voltage				-0.5		0.3V_{CC}	V
V_{IH}(1)	Input High Voltage				0.7V_{CC}		V_{CC} + 0.3	V
V_{OL}	Output Low Voltage		I_{OL} = 100 µA				0.2	V
V_{OH}	Output High Voltage		I_{OH} = -100 µA		V_{CC} - 0.2			V

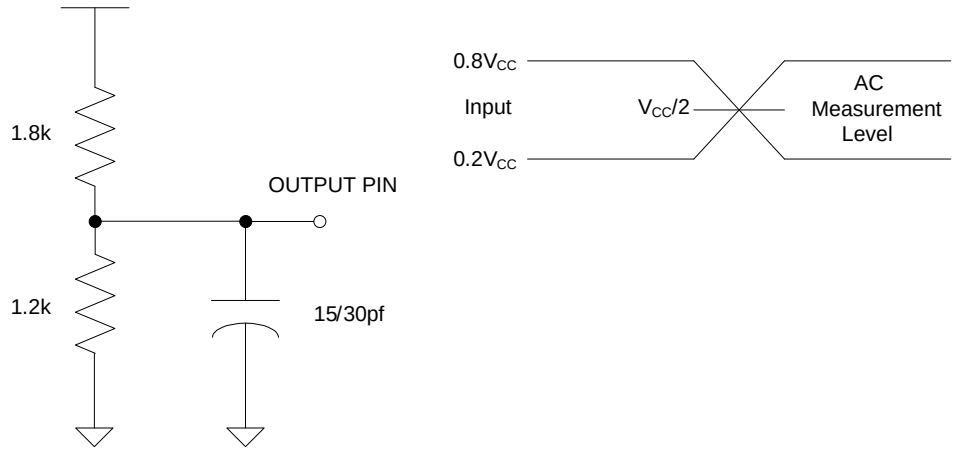
Notes:

1. **Maximum DC voltage on input or I/O pins is VCC + 0.5V. During voltage transitions, input or I/O pins may overshoot VCC by +2.0V for a period of time not to exceed 20ns. Minimum DC voltage on input or I/O pins is -0.5V. During voltage transitions, input or I/O pins may undershoot GND by -2.0V for a period of time not to exceed 20ns.**
2. **Typical values are included for reference only and are not guaranteed or tested. Typical values are measured at V_{CC} = V_{CC} (Typ), TA=25°C.**
3. **Outputs are unconnected during reading data so that output switching current is not included.**
4. **Only for the dedicated RESET# pin (or ball).**
5. **The Max of I_{LI} and I_{LO} for the dedicated RESET# pin (or ball) is ±2 µA.**
6. **These parameters are characterized and are not 100% tested.**

9.4 AC MEASUREMENT CONDITIONS

Symbol	Parameter	Min	Max	Units
CL	Load Capacitance up to 104MHz/52MHz DTR		30	pF
	Load Capacitance up to 166MHz/80MHz DTR		15	pF
TR,TF	Input Rise and Fall Times		5	ns
VIN	Input Pulse Voltages	$0.2V_{CC}$ to $0.8V_{CC}$		V
VREFI	Input Timing Reference Voltages	$0.3V_{CC}$ to $0.7V_{CC}$		V
VREFO	Output Timing Reference Voltages	$0.5V_{CC}$		V

Figure 9.1 Output test load & AC measurement I/O Waveform



9.5 PIN CAPACITANCE

(TA = 25°C, VCC=3V for IS25LP, VCC=1.8V for IS25WP, 1MHz)

Symbol	Parameter	Test Condition	IS25LP		IS25WP		Units
			Min	Max	Min	Max	
C_{IN}	Input Capacitance (CE#, SCK)	V_{IN} = 0V	-	6	-	6	pF
$C_{IN/OUT}$	Input/Output Capacitance (other pins)	$V_{IN/OUT}$ = 0V	-	8	-	10	pF

9.6 AC CHARACTERISTICS

(Under operating range, refer to section 9.4 for AC measurement conditions)

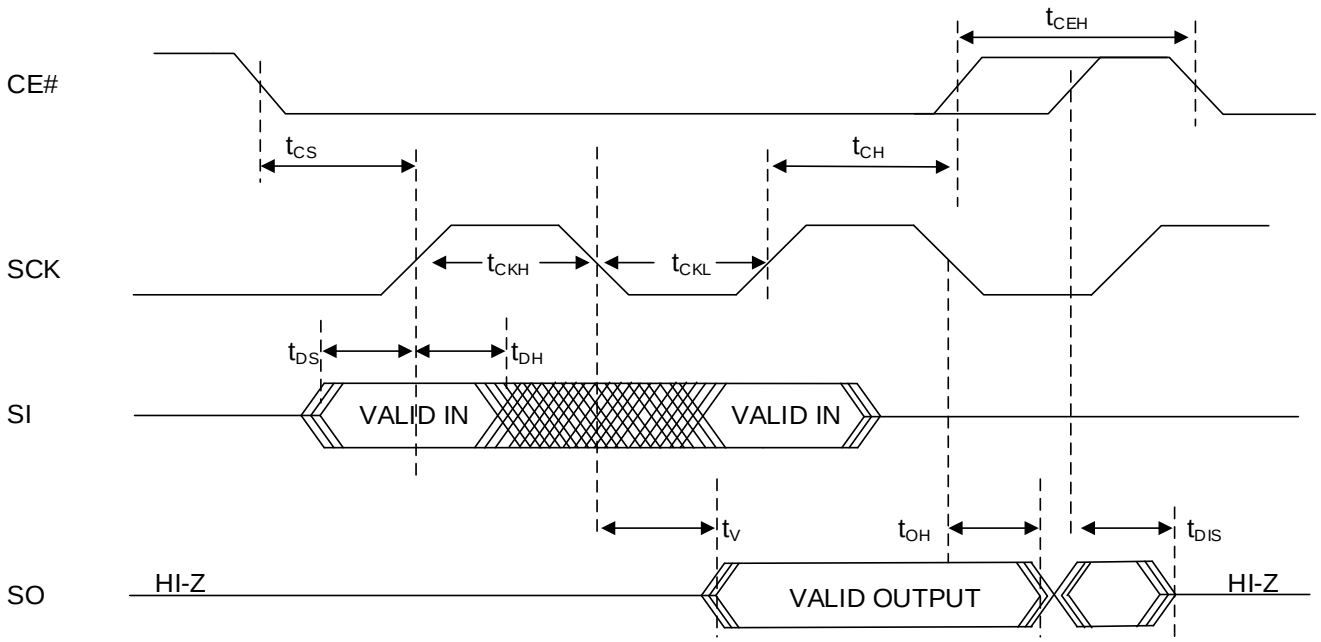
Symbol	Parameter			Min	Typ(2)	Max	Units
f_{CT}	Clock Frequency for fast read mode: SPI, Dual, Dual I/O, Quad I/O, and QPI.		Vcc=2.7V~3.6V	0		166(3)	MHz
			Vcc=2.3V~3.6V	0		133	MHz
			Vcc=1.65V~1.95V			104	MHz
f_{C2}, f_{T2}, f_{Q2}	Clock Frequency for fast read DTR: SPI DTR, Dual DTR, Dual I/O DTR, Quad I/O DTR, and QPI DTR.			0		80	MHz
f_C	Clock Frequency for read mode SPI			0		80	MHz
t_{CLCH}(1)	SCK Rise Time (peak to peak)			0.1			V/ns
t_{CHCL}(1)	SCK Fall Time (peak to peak)			0.1			V/ns
t_{CKH}	SCK High Time	For read mode		45% f_C			ns
		For others		45% $f_{CT/C2/T2/Q2}$			
t_{CKL}	SCK Low Time	For read mode		45% f_C			ns
		For others		45% $f_{CT/C2/T2/Q2}$			
t_{CEH}	CE# High Time			7			ns
t_{CS}	CE# Setup Time		IS25LP	3			ns
			IS25WP	5			
t_{CH}	CE# Hold Time			3			ns
t_{DS}	Data In Setup Time	Normal Mode		2			ns
		DTR Mode		1.5			
t_{DH}	Data in Hold Time	Normal Mode		2			ns
		DTR Mode		1.5			
t_V	Output Valid	@ 166MHz (CL = 15pF)				7	ns
		@ 133MHz (CL = 15pF)				7	
		@ 104MHz (CL = 30pF)				8	
t_{OH}	Output Hold Time			2			ns
t_{DIS}(1)	Output Disable Time					8	ns
t_{HLCH}	HOLD Active Setup Time relative to SCK			2			ns
t_{CHHH}	HOLD Active Hold Time relative to SCK			2			ns
t_{HHCH}	HOLD Not Active Setup Time relative to SCK			2			ns
t_{CHHL}	HOLD Not Active Hold Time relative to SCK			2			ns
t_{LZ}(1)	HOLD to Output Low Z					8	ns
t_{HZ}(1)	HOLD to Output High Z					8	ns
t_{EC}	Sector Erase Time (4Kbyte)				100	300	ms
	Block Erase Time (32Kbyte)				0.14	0.5	s
	Block Erase time (64Kbyte)				0.17	1.0	s
	Chip Erase Time				70	180	s
t_{PP}	Page Program Time				0.2	0.8	ms

Symbol	Parameter		Min	Typ(2)	Max	Units
t_{RES1}(1)	Release deep power down	IS25LP			3	µs
		IS25WP			5	
t_{DP}(1)	Deep power down				3	µs
t_W	Write Status Register time			2	15	ms
t_{SUS}(1)	Suspend to read ready				100	µs
t_{SRST}(1)	Software Reset recovery time				35	µs
t_{RESET}(1)	RESET# pin low pulse width		100			ns
t_{HWRST}(1)	Hardware Reset recovery time				35	µs

Notes:
1. **These parameters are characterized and not 100% tested.**
2. **Typical values are included for reference only and are not guaranteed or tested. Typical values are measured at V_{CC} = V_{CC} (Typ), TA=25°C.**
3. **Max. Frequency is 133MHz at Mode 3**

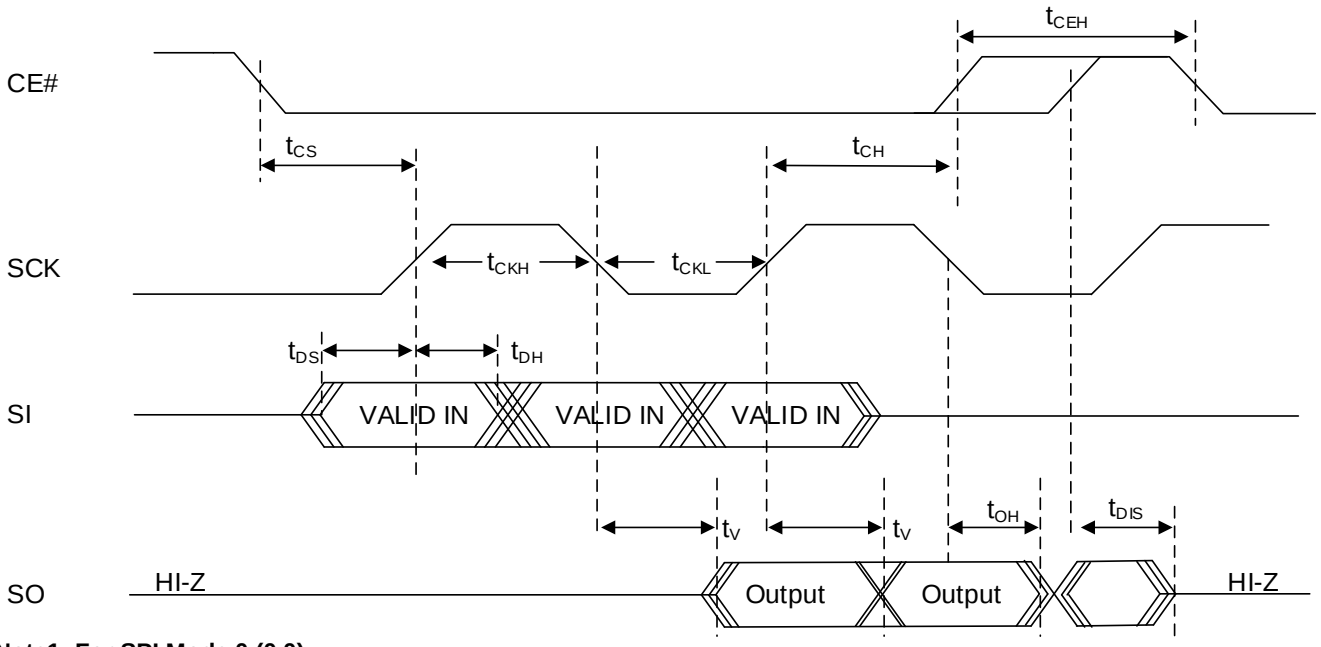
9.7 SERIAL INPUT/OUTPUT TIMING

Figure 9.2 SERIAL INPUT/OUTPUT TIMING (Normal Mode) (1)



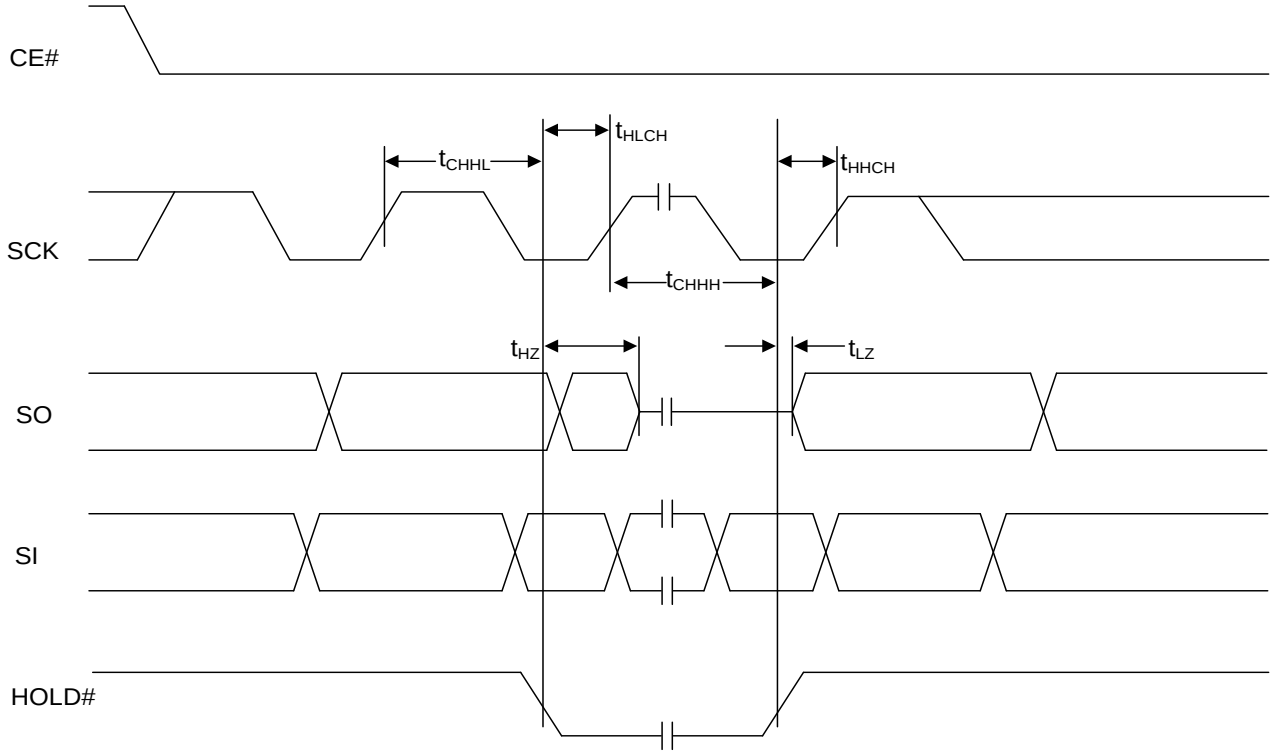
Note1: For SPI Mode 0 (0,0)

Figure 9.3 SERIAL INPUT/OUTPUT TIMING (DTR Mode) (1)



Note1: For SPI Mode 0 (0,0)

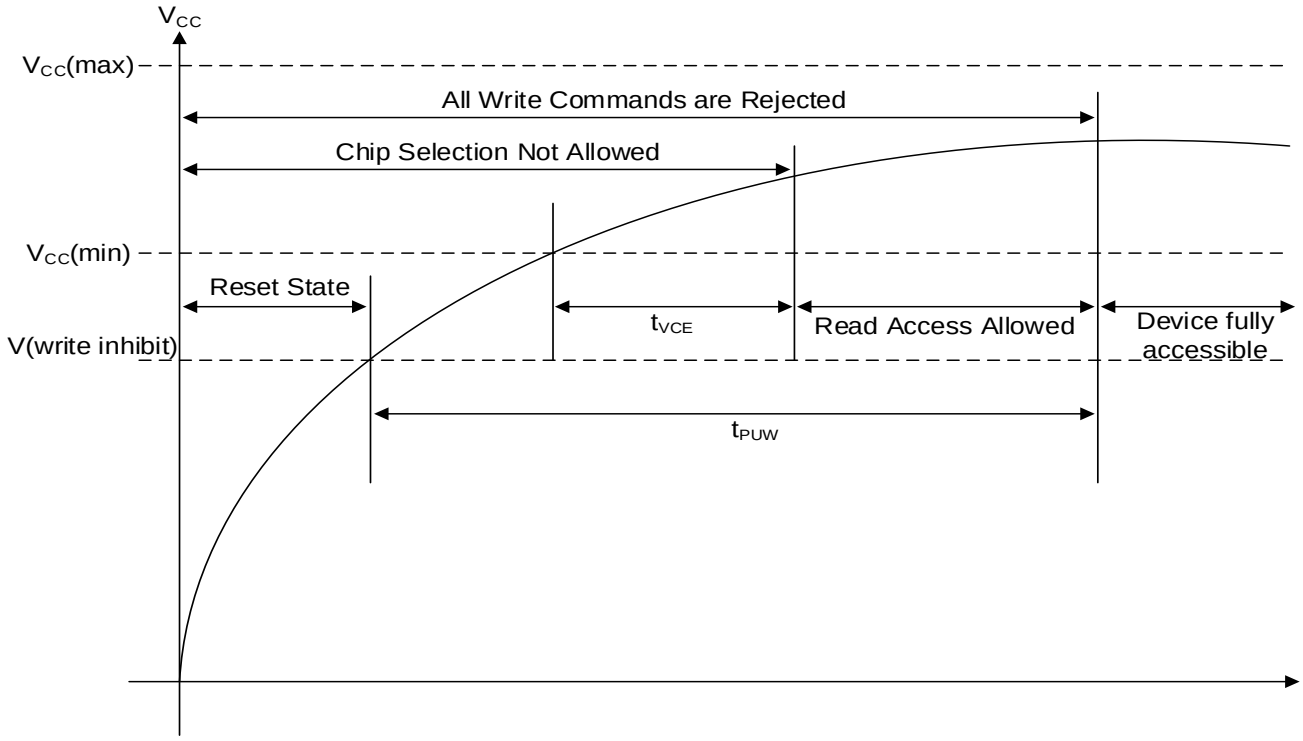
Figure 9.4 HOLD TIMING



9.8 POWER-UP AND POWER-DOWN

At Power-up and Power-down, the device must be NOT SELECTED until Vcc reaches at the right level. (Adding a simple pull-up resistor on CE# is recommended.)

Power up timing



Symbol	Parameter		Min.	Max	Unit
tVCE(1)	Vcc(min) to CE# Low		1		ms
tPUW(1)	Power-up time delay to write instruction		1	10	ms
V_{WI}(1)	Write Inhibit Voltage	IS25LP		2.1	V
		IS25WP		1.4	

Note: These parameters are characterized and not 100% tested.

9.9 PROGRAM/ERASE PERFORMANCE

Parameter	Typ	Max	Unit
Sector Erase Time (4Kbyte)	100	300	ms
Block Erase Time (32Kbyte)	0.14	0.5	s
Block Erase Time (64Kbyte)	0.17	1.0	s
Chip Erase Time	70	180	s
Page Programming Time	0.2	0.8	ms
Byte Program	8	40	µs

Note: These parameters are characterized and not 100% tested.

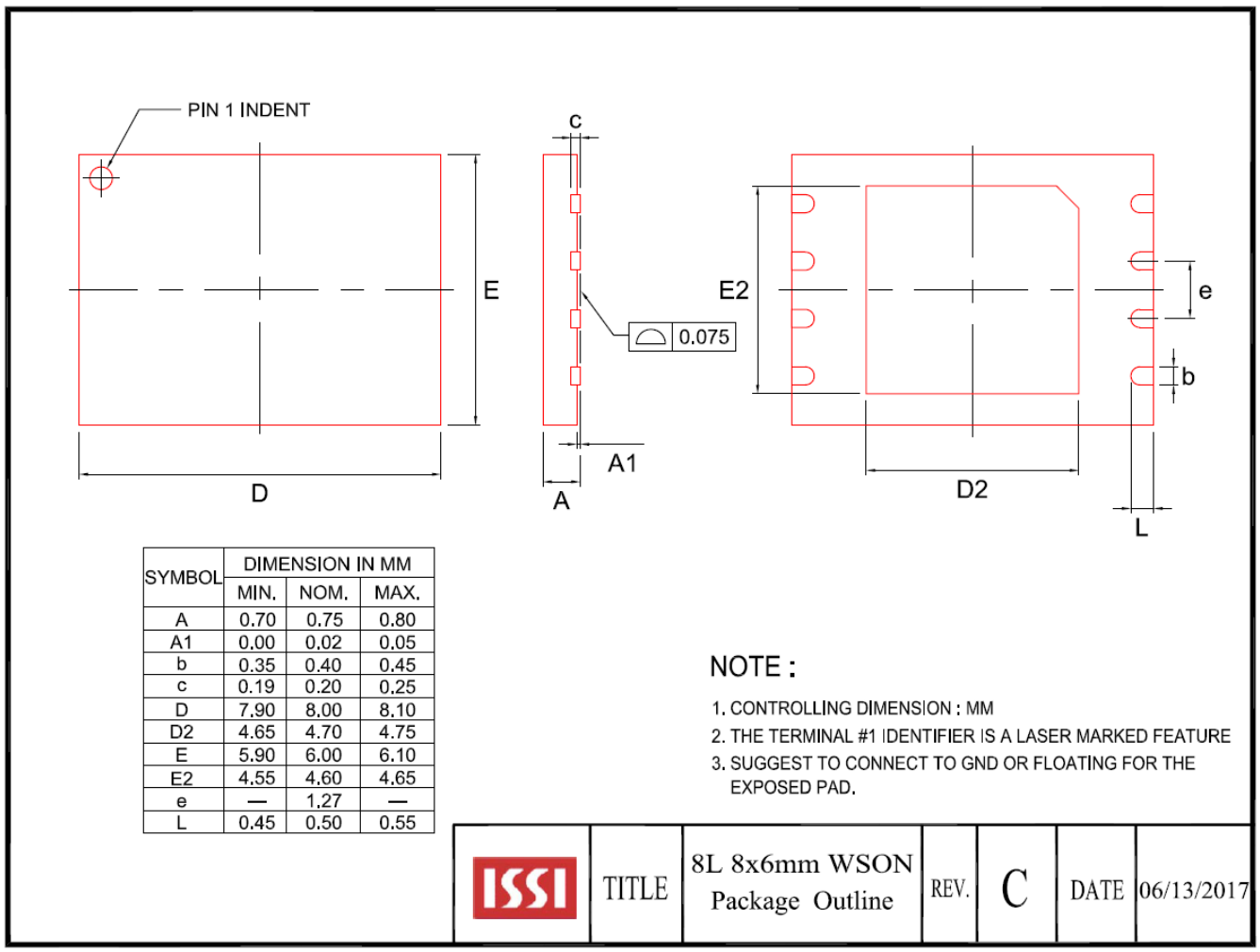
9.10 RELIABILITY CHARACTERISTICS

Parameter	Min	Unit	Test Method
Endurance	100,000	Cycles	JEDEC Standard A117
Data Retention	20	Years	JEDEC Standard A103
ESD – Human Body Model	2,000	Volts	JEDEC Standard A114
ESD – Machine Model	200	Volts	JEDEC Standard A115
Latch-Up	100 + ICC1	mA	JEDEC Standard 78

Note: These parameters are characterized and not 100% tested.

10. PACKAGE TYPE INFORMATION

10.1 8-CONTACT ULTRA-THIN SMALL OUTLINE NO-LEAD (WSON) PACKAGE 8X6MM (L)



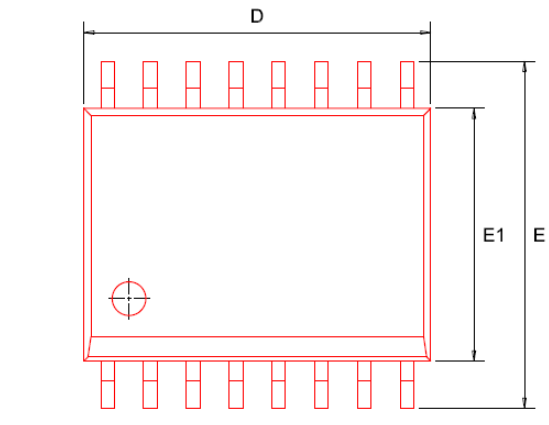
SYMBOL	DIMENSION IN MM		
	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.35	0.40	0.45
c	0.19	0.20	0.25
D	7.90	8.00	8.10
D2	4.65	4.70	4.75
E	5.90	6.00	6.10
E2	4.55	4.60	4.65
e	—	1.27	—
L	0.45	0.50	0.55

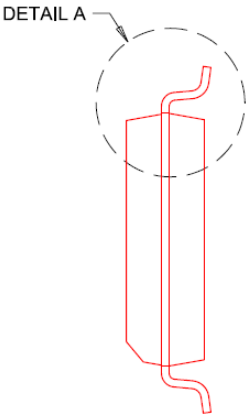
NOTE :

1. CONTROLLING DIMENSION : MM
2. THE TERMINAL #1 IDENTIFIER IS A LASER MARKED FEATURE
3. SUGGEST TO CONNECT TO GND OR FLOATING FOR THE EXPOSED PAD.

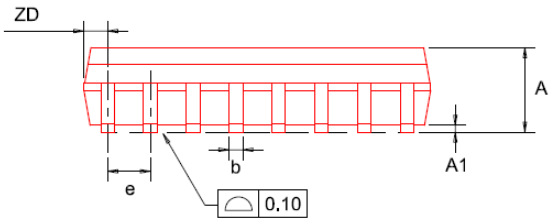
	TITLE	8L 8x6mm WSON Package Outline	REV.	C	DATE	06/13/2017

10.2 16-LEAD PLASTIC SMALL OUTLINE PACKAGE (300 MILS BODY WIDTH) (M)



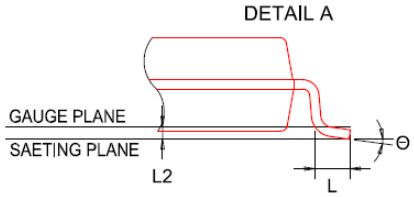


SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	—	—	2.65
A1	0.10	—	0.30
b	0.31	—	0.51
D	10.08	10.31	10.49
E	10.01	10.31	10.64
E1	7.39	7.49	7.59
e	1.27 BSC.		
L	0.38	0.81	1.27
L2	0.25 BSC.		
ZD	0.705 BSC.		
Θ	0°	—	8°



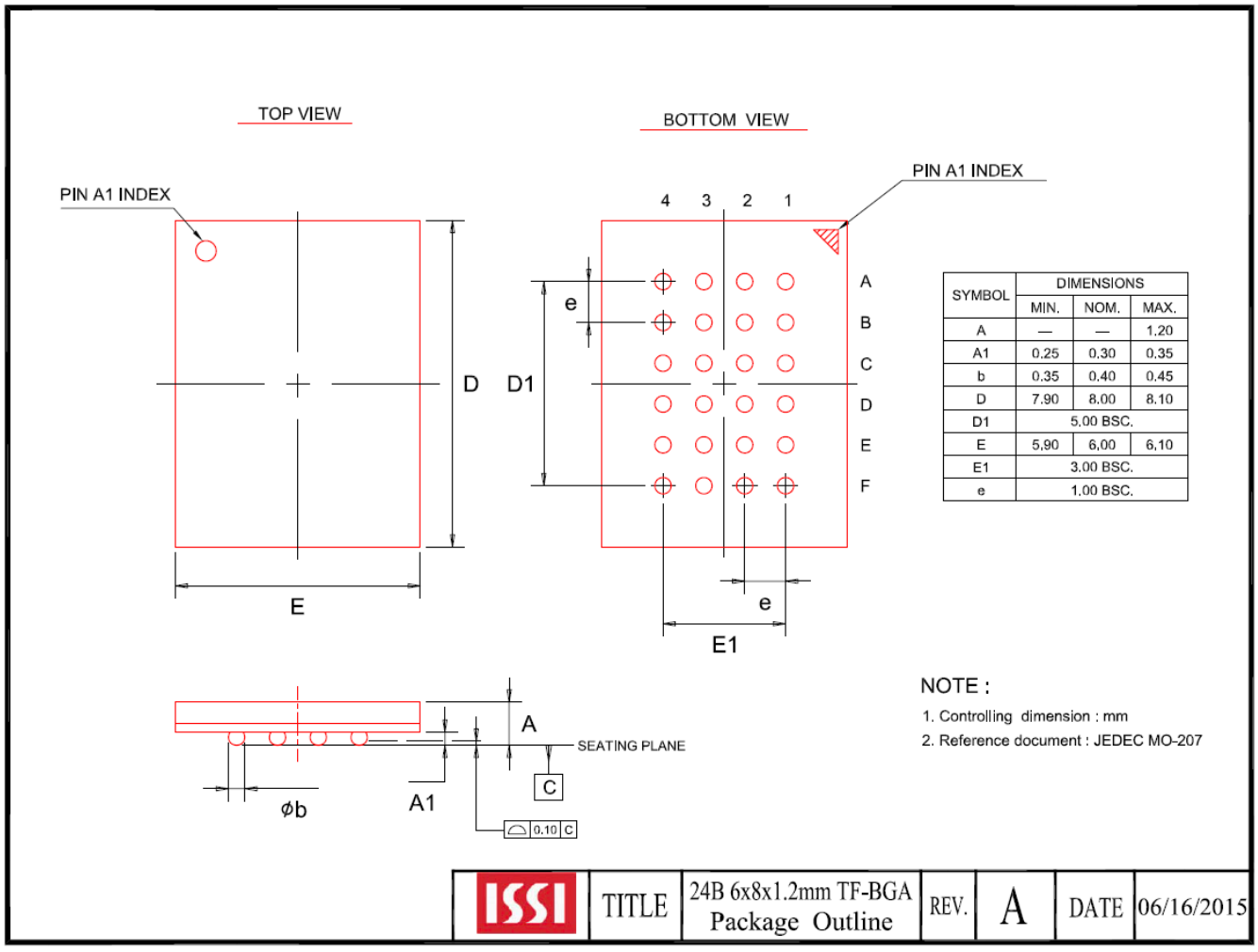
NOTE :

1. CONTROLLING DIMENSION : MM
2. DIMENSION D DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.



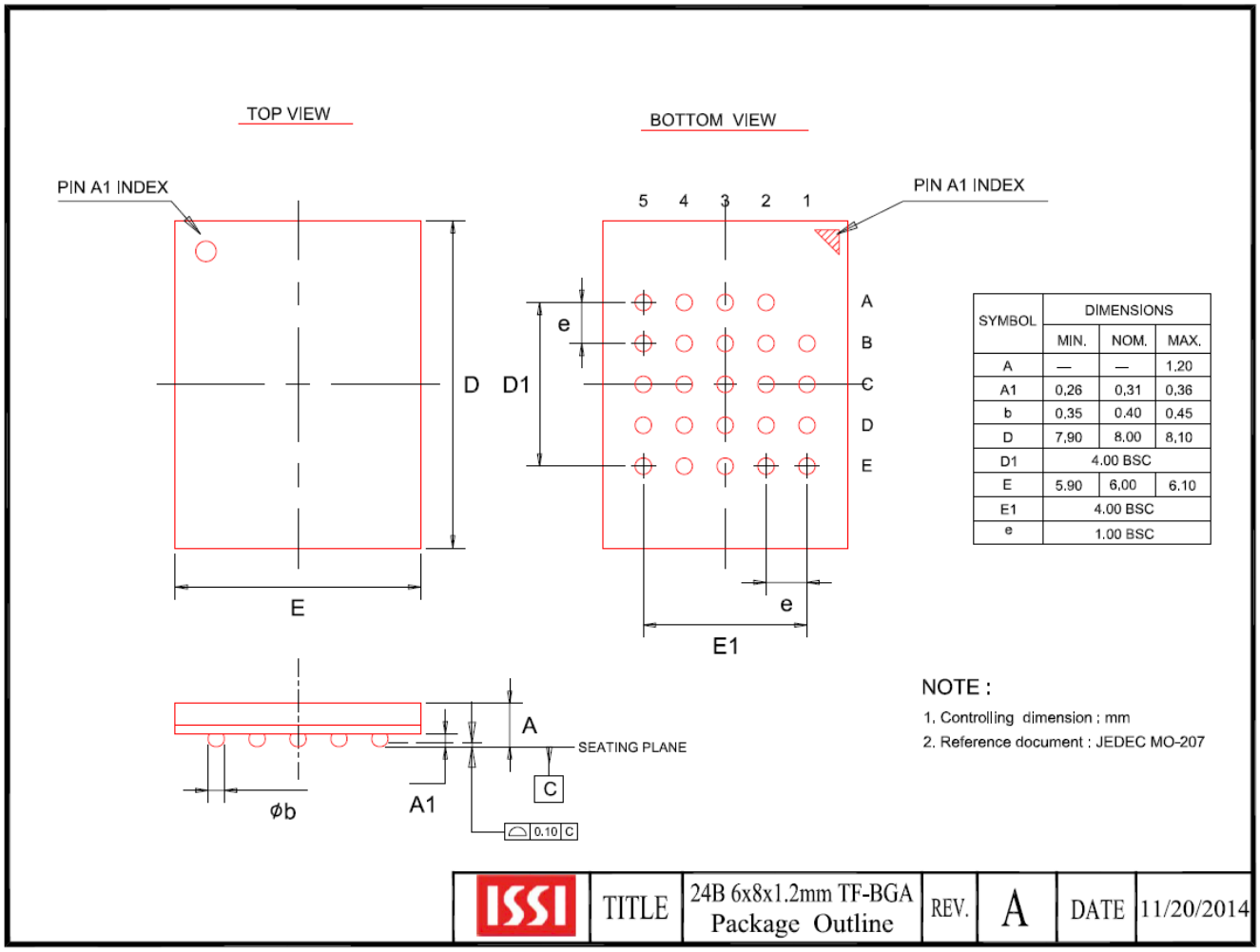
ISSI	TITLE	16L 300mil SOP Package Outline	REV.	A	DATE	06/26/2015

10.3 24-BALL THIN PROFILE FINE PITCH BGA 6X8MM 4X6 BALL ARRAY (G)



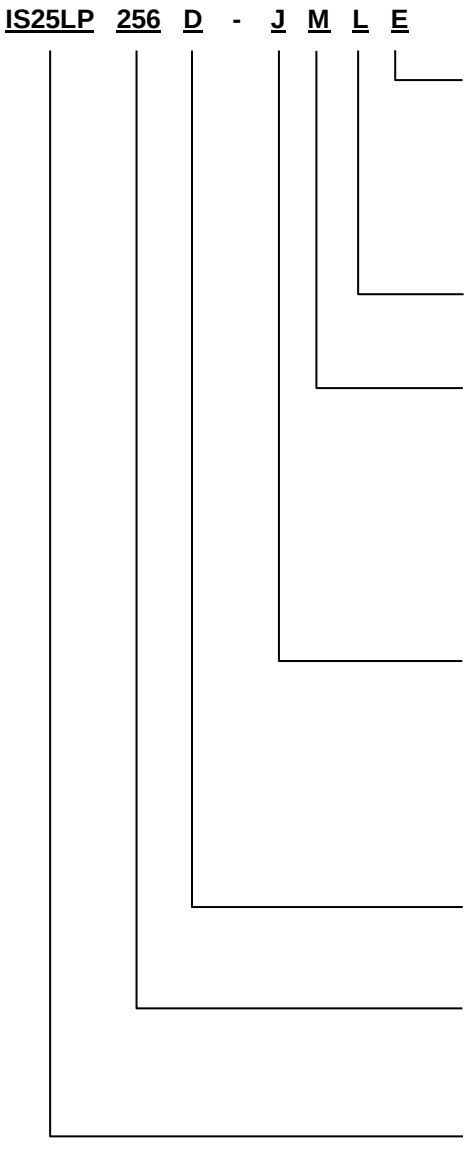
SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	—	—	1.20
A1	0.25	0.30	0.35
b	0.35	0.40	0.45
D	7.90	8.00	8.10
D1	5.00 BSC.		
E	5.90	6.00	6.10
E1	3.00 BSC.		
e	1.00 BSC.		

10.4 24-BALL THIN PROFILE FINE PITCH BGA 6X8MM 5X5 BALL ARRAY (H)



SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	—	—	1.20
A1	0.26	0.31	0.36
b	0.35	0.40	0.45
D	7.90	8.00	8.10
D1		4.00 BSC	
E	5.90	6.00	6.10
E1		4.00 BSC	
e		1.00 BSC	

11. ORDERING INFORMATION – Valid Part Numbers



TEMPERATURE RANGE
E = Extended (-40°C to +105°C)
A3 = Automotive Grade (-40°C to +125°C)

PACKAGING CONTENT
L = **RoHS compliant**

PACKAGE Type
L = 8-contact WSON (8x6mm)
M = 16-pin SOIC 300mil
G = 24-ball TFBGA 4x6 ball array
H = 24-ball TFBGA 5x5 ball array
W = KGD

Option
J = Standard
R = Dedicated RESET# pin (or ball)
Q = QE bit set to 1 (Call Factory)
E = SFDP + QE bit set to 1 (Call Factory)
G = SFDP + Dedicated RESET#(Call Factory)

Die Revision
D = Revision D

Density
256 = 256 Megabit

BASE PART NUMBER
IS = Integrated Silicon Solution Inc.
25LP = FLASH, 2.30V ~ 3.60V, QPI
25WP = FLASH, 1.65V ~ 1.95V, QPI

Density, Voltage	Frequency (MHz)	Order Part Number[3]	Package
256M, 3V	STR 166 [2], DTR 80	IS25LP256D-JLLE	8-contact WSON 8x6mm
		IS25LP256D-QLLE	8-contact WSON (8x6mm)
		IS25LP256D-ELLE	8-contact WSON (8x6mm)
		IS25LP256D-JMLE	16-pin SOIC 300mil
		IS25LP256D-RMLE	16-pin SOIC 300mil
		IS25LP256D-JGLE	24-ball TFBGA 6x8mm 4x6 ball array
		IS25LP256D-RGLE	24-ball TFBGA 6x8mm 4x6 ball array
		IS25LP256D-JHLE	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LP256D-RHLE	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LP256D-JLLA3	8-contact WSON (8x6mm)
		IS25LP256D-QLLA3	8-contact WSON (8x6mm)
		IS25LP256D-JMLA3	16-pin SOIC 300mil
		IS25LP256D-RMLA3	16-pin SOIC 300mil
		IS25LP256D-JGLA3	24-ball TFBGA 6x8mm 4x6 ball array
		IS25LP256D-RGLA3	24-ball TFBGA 6x8mm 4x6 ball array
		IS25LP256D-JHLA3	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LP256D-RHLA3	24-ball TFBGA 6x8mm 5x5 ball array
256M, 1.8V	STR 104, DTR 80	IS25WP256D-JLLE	8-contact WSON 8x6mm
		IS25WP256D-JMLE	16-pin SOIC 300mil
		IS25WP256D-RMLE	16-pin SOIC 300mil
		IS25WP256D-JGLE	24-ball TFBGA 6x8mm 4x6 ball array
		IS25WP256D-RGLE	24-ball TFBGA 6x8mm 4x6 ball array
		IS25WP256D-JHLE	24-ball TFBGA 6x8mm 5x5 ball array
		IS25WP256D-RHLE	24-ball TFBGA 6x8mm 5x5 ball array
		IS25WP256D-JLLA3	8-contact WSON (8x6mm)
		IS25WP256D-JMLA3	16-pin SOIC 300mil
		IS25WP256D-RMLA3	16-pin SOIC 300mil
		IS25WP256D-JGLA3	24-ball TFBGA 6x8mm 4x6 ball array
		IS25WP256D-RGLA3	24-ball TFBGA 6x8mm 4x6 ball array
		IS25WP256D-JHLA3	24-ball TFBGA 6x8mm 5x5 ball array
		IS25WP256D-GHLA3	24-ball TFBGA 6x8mm 5x5 ball array

Notes:
1. A3 meets AEC-Q100 requirements with PPAP.
Temp Grades: E= -40 to 105°C, A3= -40 to 125°C
2. 133MHz at Vcc = 2.3V~3.6V (Mode 0 & Mode 3) or 133MHz at Vcc = 2.7V~3.6V (Mode 3 only)

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View IS25LP256D-JLLE-TR on WIN SOURCE
- ISSI, Integrated Silicon Solution Inc Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management