



IS25LE080G

IS25LE040G

8/4Mb

**SERIAL FLASH MEMORY WITH 166MHZ MULTI I/O SPI &
QUAD I/O QPI DTR INTERFACE WITH ON-CHIP ECC**

PRELIMINARY DATA SHEET

8/4Mb

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

PRELIMINARY INFORMATION

FEATURES

- **Industry Standard Serial Interface**

- IS25LE080G: 8Mbit/1Mbyte
- IS25LE040G: 4Mbit/512Kbyte
- 256 bytes per Programmable Page
- Supports standard SPI, Fast, Dual, Dual I/O, Quad, Quad I/O, SPI DTR, Dual I/O DTR, Quad I/O DTR, and QPI
- Supports Serial Flash Discoverable Parameters (SFDP)

- **High Performance Serial Flash (SPI)**

- 66MHz Normal Read
- Up to 166MHz Fast Read
- Up to 80MHz DTR (Dual Transfer Rate)
- 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
- **1-bit ERROR Detection and Correction per 32-bit boundary (with ECC)**
- ERR#: Error Indication Signal ⁽²⁾

- **Flexible & Efficient Memory Architecture**

- Chip Erase with Uniform: Sector/Block Erase (4/32/64 Kbyte)
- Program 1 to 256 Bytes per Page
- Program/Erase Suspend & Resume

- **Efficient Read and Program modes**

- Continuous Read 8/16/32/64-Byte Burst Wrap
- Selectable Burst Length
- QPI for Reduced Instruction Overhead

- **Low Power with Wide Temp. Ranges**

- Single Voltage Supply
IS25LE: 2.70V to 3.60V
- 4 mA Active Read Current
- Temp Grades:
Extended: -40°C to +105°C
Auto Grade (A3): -40°C to +125°C

- **Advanced Security Protection**

- Software and Hardware Write 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⁽¹⁾**

- H = 24-ball TFBGA 6x8mm (5x5 ball array)
- B = 8-pin SOIC 208mil (Call Factory)
- N = 8-pin SOIC 150mil (Call Factory)
- T = 8-contact USON 4x3mm
- W = KGD
- Green Package (RoHS Compliant, Halogen-Free) and TSCA Compliant

Notes:

1. Call Factory for other package options available

2. ERR# is supported by 24BGA and KGD

GENERAL DESCRIPTION

The IS25LE080G/040G 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. 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.

Initial state of the memory array is erased (all bits are set to 1) when shipped from the factory.

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.

Automatic ECC Protection (1-bit Error Correction) applies per every 32-bit to maximize reliability.
When ECC is ON, the device allows only single programming operation to each 4-byte ECC boundary between erase operations.

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).

Multi 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.

QPI

The device supports Quad Peripheral Interface (QPI) operations only when the device is switched from Standard/Dual/Quad SPI mode to QPI mode using the enter QPI (35h) instruction. The typical SPI protocol requires that the byte-long instruction code being shifted into the device only via SI pin in eight serial clocks. The QPI mode utilizes all four I/O pins to input the instruction code thus requiring only two serial clocks. This can significantly reduce the SPI instruction overhead and improve system performance. Only QPI mode or SPI/Dual/Quad mode can be active at any given time. Enter QPI (35h) and Exit QPI (F5h) instructions are used to switch between these two modes, regardless of the non-volatile Quad Enable (QE) bit status in the Status Register. Power Reset or Software Reset will return the device into the standard SPI mode. SI and SO pins become bidirectional I/O0 and I/O1, and WP# and HOLD# pins become I/O2 and I/O3 respectively during QPI mode.

DTR

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

TABLE OF CONTENTS

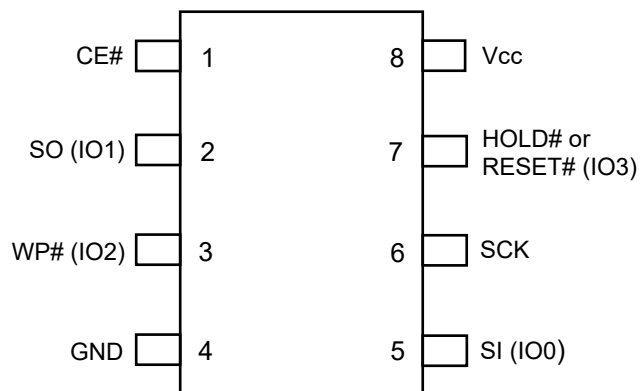
FEATURES	2
GENERAL DESCRIPTION	3
TABLE OF CONTENTS	4
1. PIN CONFIGURATION	7
2. PIN DESCRIPTIONS	9
3. BLOCK DIAGRAM	10
4. SPI MODES DESCRIPTION	11
5. SYSTEM CONFIGURATION	13
5.1 BLOCK/SECTOR ADDRESSES	13
5.2 Serial Flash Discoverable Parameters	14
6. REGISTERS	19
6.1 STATUS REGISTER	19
6.2 FUNCTION REGISTER	22
6.3 READ REGISTER AND EXTENDED REGISTER	23
6.4 ECC REGISTER	27
7. PROTECTION MODE	28
7.1 HARDWARE WRITE PROTECTION	28
7.2 SOFTWARE WRITE PROTECTION	28
8. DEVICE OPERATION	29
8.1 NORMAL READ OPERATION (NORD, 03h)	32
8.2 FAST READ OPERATION (FRD, 0Bh)	34
8.3 HOLD OPERATION	36
8.4 FAST READ DUAL I/O OPERATION (FRDIO, BBh)	36
8.5 FAST READ DUAL OUTPUT OPERATION (FRDO, 3Bh)	39
8.6 FAST READ QUAD OUTPUT OPERATION (FRQO, 6Bh)	40
8.7 FAST READ QUAD I/O OPERATION (FRQIO, EBh)	42
8.8 PAGE PROGRAM OPERATION (PP, 02h)	46
8.9 QUAD INPUT PAGE PROGRAM OPERATION (PPQ, 32h/38h)	48
8.10 ERASE OPERATION	49
8.11 SECTOR ERASE OPERATION (SER, D7h/20h)	50
8.12 BLOCK ERASE OPERATION (BER32K:52h, BER64K:D8h)	51
8.13 CHIP ERASE OPERATION (CER, C7h/60h)	53
8.14 WRITE ENABLE OPERATION (WREN, 06h)	54
8.15 WRITE DISABLE OPERATION (WRDI, 04h)	55
8.16 READ STATUS REGISTER OPERATION (RDSR, 05h)	56
8.17 WRITE STATUS REGISTER OPERATION (WRSR, 01h)	57
8.18 READ FUNCTION REGISTER OPERATION (RDFR, 48h)	58
8.19 WRITE FUNCTION REGISTER OPERATION (WRFR, 42h)	59
8.20 ENTER QUAD PERIPHERAL INTERFACE (QPI) MODE OPERATION (QPIEN, 35h; QPIDI, F5h) ..	60

8.21 PROGRAM/ERASE SUSPEND & RESUME.....	61
8.22 ENTER DEEP POWER DOWN (DP, B9h).....	63
8.23 RELEASE DEEP POWER DOWN (RDPD, ABh).....	64
8.24 SET READ PARAMETERS OPERATION (SRPNV: 65h, SRPV: C0h/63h).....	65
8.25 SET EXTENDED READ PARAMETERS OPERATION (SERPNV: 85h, SERPV: 83h).....	67
8.26 READ READ PARAMETERS OPERATION (RDRP, 61h).....	68
8.27 READ EXTENDED READ PARAMETERS OPERATION (RDERP, 81h).....	69
8.28 CLEAR EXTENDED READ PARAMETERS OPERATION (CLERP, 82h).....	70
8.29 READ PRODUCT IDENTIFICATION (RDID, ABh).....	71
8.30 READ PRODUCT IDENTIFICATION BY JEDEC ID OPERATION (RDJDID, 9Fh; RDJDIDQ, AFh).....	73
8.31 READ DEVICE MANUFACTURER AND DEVICE ID OPERATION (RDMDID, 90h).....	74
8.31 READ UNIQUE ID NUMBER (RDUID, 4Bh).....	75
8.32 READ SFDP OPERATION (RDSFDP, 5Ah).....	76
8.33 NO OPERATION (NOP, 00h).....	76
8.34 SOFTWARE RESET (RESET-ENABLE (RSTEN, 66h) AND RESET (RST, 99h)) AND HARDWARE RESET.....	77
8.35 SECURITY INFORMATION ROW.....	78
8.36 INFORMATION ROW ERASE OPERATION (IRER, 64h).....	79
8.37 INFORMATION ROW PROGRAM OPERATION (IRP, 62h).....	80
8.38 INFORMATION ROW READ OPERATION (IRRD, 68h).....	81
8.39 FAST READ DTR MODE OPERATION In SPI MODE (FRDTR, 0Dh).....	82
8.40 FAST READ DUAL IO DTR MODE OPERATION (FRDDTR, BDh).....	84
8.41 FAST READ QUAD IO DTR MODE OPERATION IN SPI MODE (FRQDTR, EDh).....	87
8.42 SECTOR LOCK/UNLOCK FUNCTIONS.....	91
8.43 WRITE REGISTER OPERATION (WRECC, B5h).....	93
8.44 READ ECC REGISTER OPERATION (RDECC, B3h).....	94
8.45 CLEAR ECC REGISTER OPERATION (CLECC, B6h).....	95
8.46 ERR# OPERATION.....	96
9. ELECTRICAL CHARACTERISTICS.....	97
9.1 ABSOLUTE MAXIMUM RATINGS ⁽¹⁾	97
9.2 OPERATING RANGE.....	97
9.3 DC CHARACTERISTICS.....	98
9.4 AC MEASUREMENT CONDITIONS.....	99
9.5 PIN CAPACITANCE.....	99
9.6 AC CHARACTERISTICS.....	100
9.7 SERIAL INPUT/OUTPUT TIMING.....	102
9.8 POWER-UP AND POWER-DOWN.....	105
9.9 PROGRAM/ERASE PERFORMANCE.....	106
9.10 RELIABILITY CHARACTERISTICS.....	106
10. PACKAGE TYPE INFORMATION.....	107
10.1 8-Pin JEDEC 208mil Broad Small Outline Integrated Circuit (SOIC) Package (B).....	107

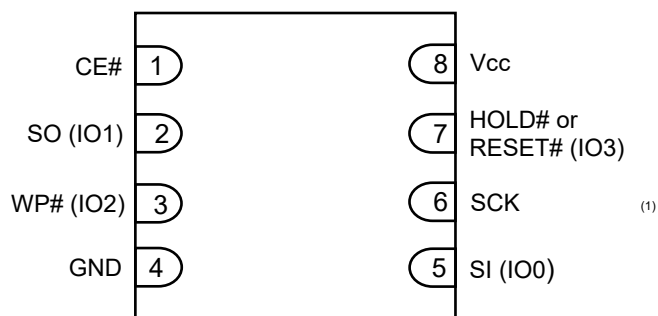


10.2 8-Pin JEDEC 150mil Broad Small Outline Integrated Circuit (SOIC) Package (N).....	108
10.3 24-BALL THIN PROFILE FINE PITCH BGA 6x8mm 5x5 BALL ARRAY (H)	109
10.4 8-CONTACT Ultra-Thin Small Outline No-Lead (USON) Package 4x3mm (T)	110
11. ORDERING INFORMATION - Valid Part Numbers.....	111

1. PIN CONFIGURATION



8-pin SOIC 208mil (Package: B)
8-pin SOIC 150mil (Package: N)



8-pin USON 4x3mm (Package: T)

Pin	Function
A2	NC
A3	NC
A4	NC/RESET# ⁽²⁾
A5	ERR# ⁽²⁾
B1	NC
B2	SCK
B3	GND
B4	VCC
B5	NC
C1	NC
C2	CE#
C3	NC
C4	WP#(IO2)
C5	NC
D1	NC
D2	SO(IO1)
D3	SI(IO0)
D4	HOLD# or RESET# (IO3) ⁽¹⁾
D5	NC
E1	NC
E2	NC
E3	NC
E4	NC
E5	NC

Notes:

1. The pin can be configured as Hold# or Reset# by setting P7 bit of the Read Register. Pin default is Hold# (IO3).
2. **Call factory for RESET#**

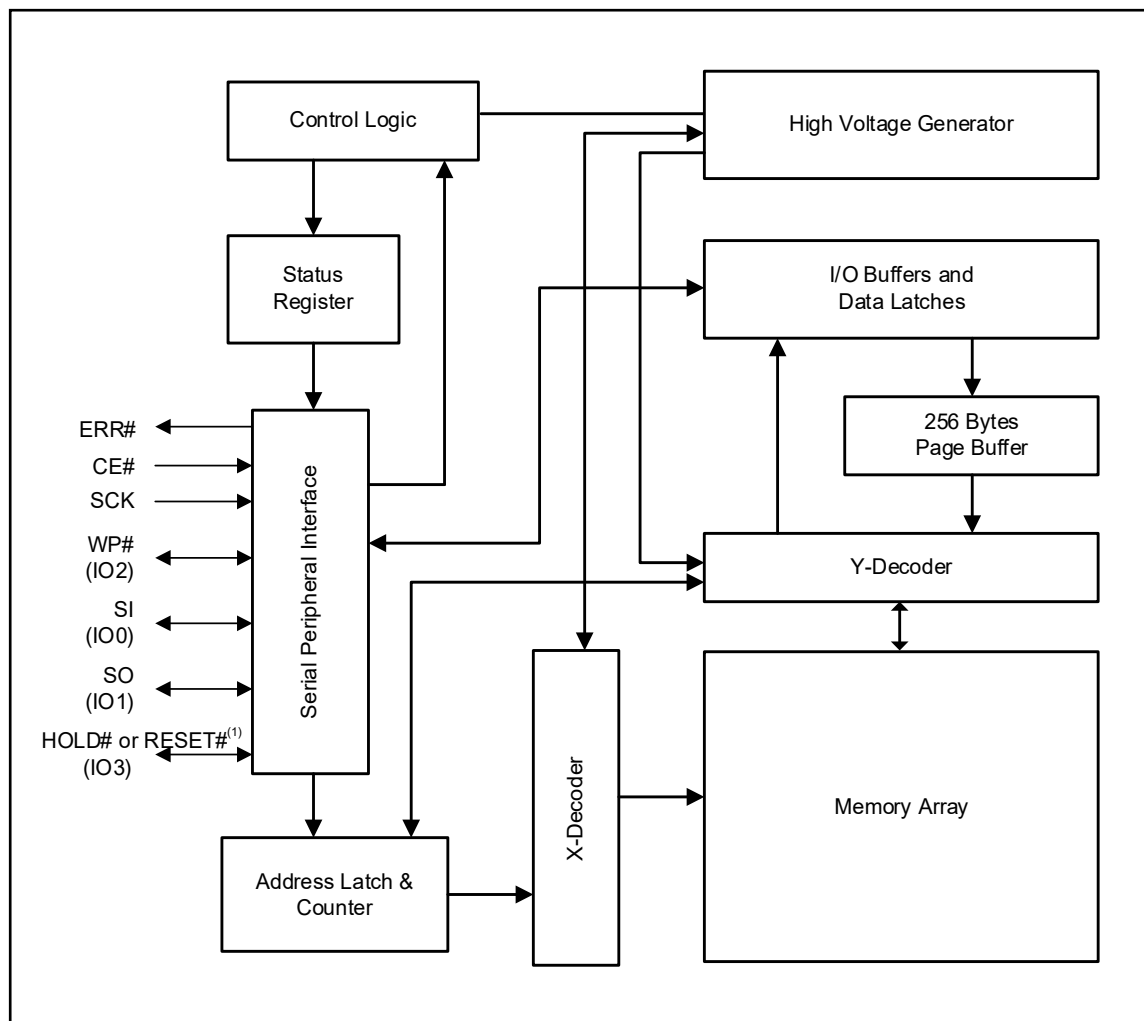
2. PIN DESCRIPTIONS

SYMBOL	TYPE	DESCRIPTION
CE#	INPUT	Chip Enable: The Chip Enable (CE#) pin enables and disables the devices operation.
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.
ERR# ⁽¹⁾	OUTPUT	ECC Error Indication Signal: Indicates ECC Event occurrence. Open Drain. <i>It is internally pulled-up and may be left floating.</i>
RESET#	INPUT	<i>This dedicated RESET# is available in 24-ball BGA package only.</i>
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 unconnected.

Note

1. ERR# is supported by 24 BGA and KGD

3. BLOCK DIAGRAM



Note1: According to the P7 bit setting in Read Register, either **HOLD#** (P7=0) or **RESET#** (P7=1) pin can be selected.

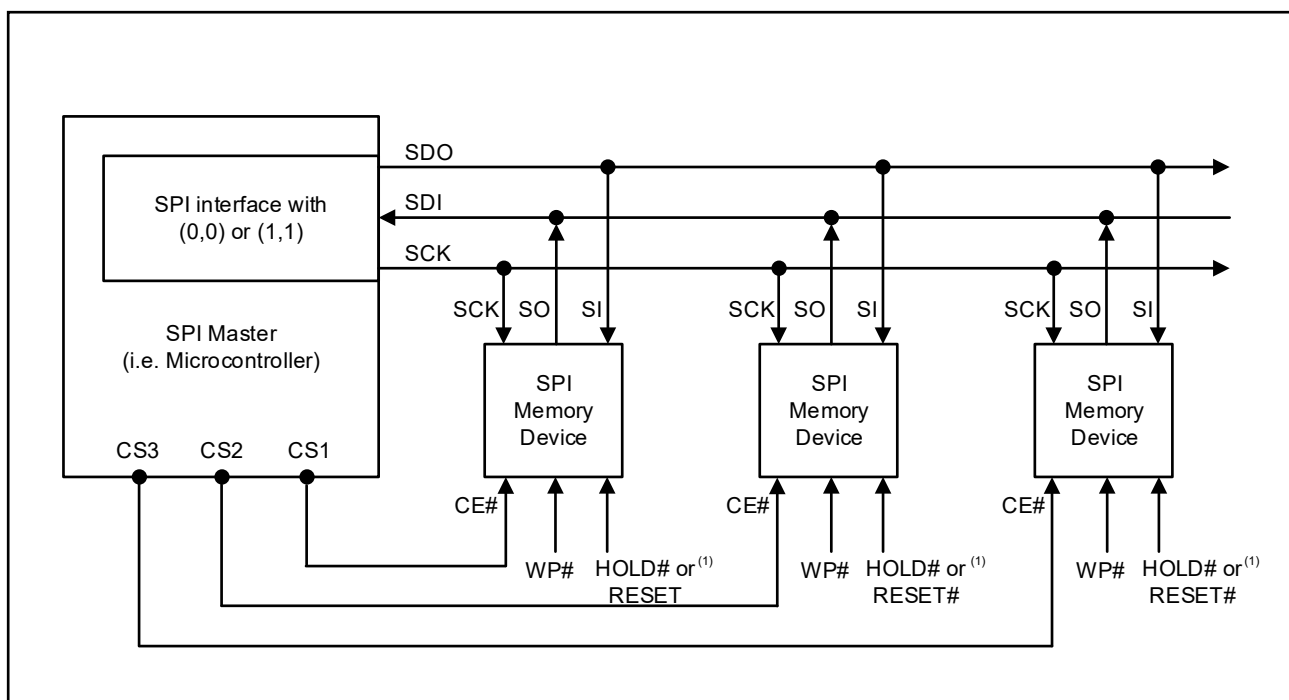
4. SPI MODES DESCRIPTION

Multiple IS25LE080G/040G 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)

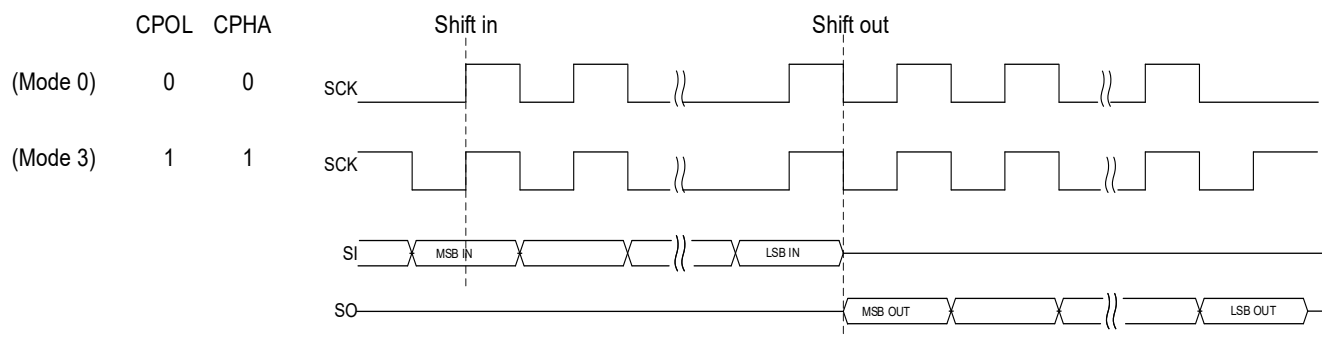
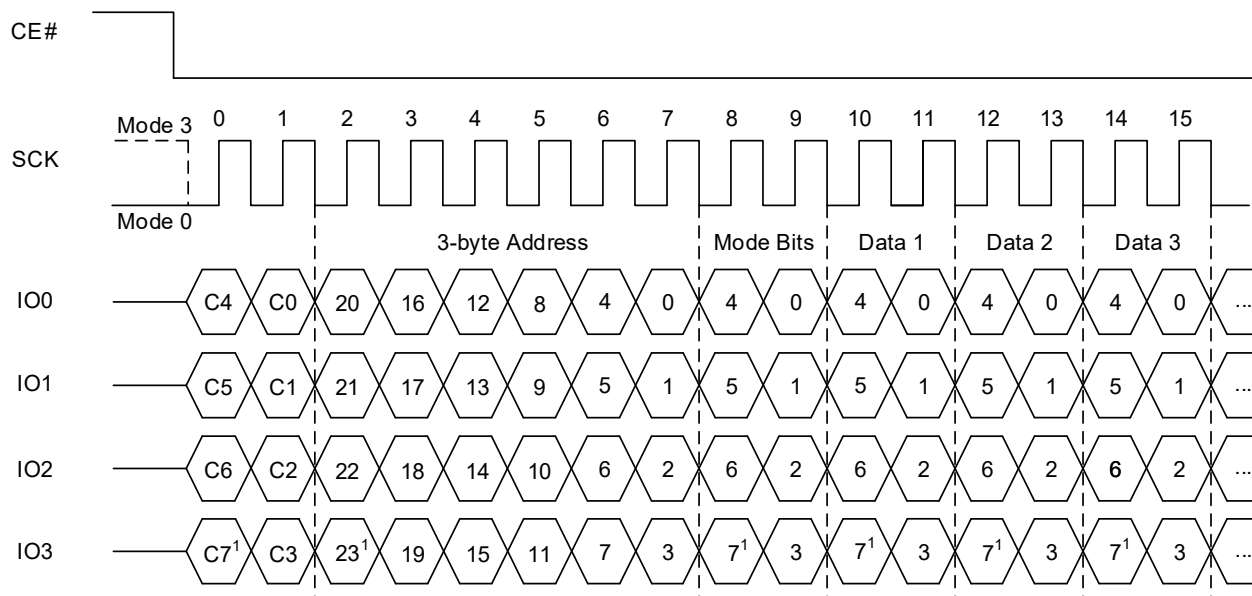
The difference between these two modes is the clock polarity. When the SPI master is in stand-by mode, the serial clock remains at “0” (SCK = 0) for Mode 0 and the clock remains at “1” (SCK = 1) for Mode 3. Please refer to Figure 4.2 and Figure 4.3 for SPI and QPI mode. In both modes, the input data 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# pins become bidirectional IO0, IO1, IO2, and IO3 respectively during Quad I/O or QPI mode.

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 (Kbytes)	Address Range
4Mb	8Mb	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 3	Block 6	Sector 48	4	030000h – 030FFFh
				:	:	:
			Block 7	:	:	:
				Sector 63	4	03F000h – 03FFFFh
		:	:	:	:	:
		Block 7	Block 14	Sector 112	4	070000h – 070FFFh
				:	:	:
			Block 15	:	:	:
				Sector 127	4	07F000h – 07FFFFh
		:	:	:	:	:
		Block 15	Block 30	Sector 240	4	0F0000h – 0F0FFFh
				:	:	:
			Block 31	:	:	:
				Sector 255	4	0FF000h – 0FFFFFh

5.2 SERIAL FLASH DISCOVERABLE PARAMETERS

The Serial Flash Discoverable Parameters (SFDP) standard defines the structure of the SFDP database within the memory device. SFDP is the standard of JEDEC JESD216.

The JEDEC-defined header with Parameter ID FF00h and related Basic Parameter Table is mandatory. Additional parameter headers and tables are optional.

Table 5.2 Signature and Parameter Identification Data Values

Description		Address (Byte)	Address (Bit)	Data
SFDP Signature		00h	7:0	53h
		01h	15:8	46h
		02h	23:16	44h
		03h	31:24	50h
SFDP Revision	Minor	04h	7:0	06h
	Major	05h	15:8	01h
Number of Parameter Headers (NPH)		06h	23:16	00h
Unused		07h	31:24	FFh
Parameter ID LSB		08h	7:0	00h
Parameter Minor Revision		09h	15:8	06h
Parameter Major Revision		0Ah	23:16	01h
Parameter Table Length (in DWPRDs)		0Bh	31:24	10h
Basic Flash Parameter Table Pointer (PTP)		0Ch	7:0	30h
		0Dh	15:8	00h
		0Eh	23:16	00h
Parameter ID MSB		0Fh	31:24	FFh

Table 5.3 JEDEC Basic Flash Parameter Table

Description	Address (Byte)	Address (Bit)	Data
Minimum Sector Erase Sizes	30h	1:0	01b
Write Granularity		2	1b
Volatile Status Register Block Protect bits		3	0b
Write Enable Instruction Select for writing to Volatile Status Register		4	0b
Unused		7:5	111b
4KB Erase Instruction	31h	15:8	20h
Supports (1-1-2) Fast Read	32h	16	1b
Address Bytes		18:17	00b
Supports Double Transfer Rate (DTR) Clocking		19	1b
Supports (1-2-2) Fast Read		20	1b
Supports (1-4-4) Fast Read		21	1b
Supports (1-1-4) Fast Read		22	1b
Unused		23	1b
Reserved	33h	31:24	FFh
Flash memory Density (bits)	34h	7:0	FFh
	35h	15:8	FFh
Flash memory Density (bits) 8Mb	36h	23:16	7Fh
	36h	23:16	3Fh
Flash memory Density (bits)	37h	31:24	00h
1-4-4 Fast Read Wait Cycle Count	38h	4:0	00100b
1-4-4 Fast Read Mode bit Cycle Count		7:5	010b
1-4-4 Fast Read Instruction	39h	15:8	EBh
1-1-4 Fast Read Wait Cycle Count	3Ah	20:16	01000b
1-1-4 Fast Read Mode bit Cycle Count		23:21	000b
1-1-4 Fast Read Instruction	3Bh	31:24	6Bh
1-1-2 Fast Read Wait Cycle Count	3Ch	4:0	01000b
1-1-2 Fast Read Mode bit Cycle Count		7:5	000b
1-1-2 Fast Read Instruction	3Dh	15:8	3Bh
1-2-2 Fast Read Wait Cycle Count	3Eh	20:16	00000b
1-2-2 Fast Read Mode bit Cycle Count		23:21	100b
1-2-2 Fast Read Instruction	3Fh	31:24	BBh

Table 5.3 JEDEC Basic Flash Parameter Table (Continued)

Description	Address (Byte)	Address (Bit)	Data
Supports (2-2-2) Fast Read	40h	0	0
Reserved		3:1	111b
Supports (4-4-4) Fast Read		4	1
Reserved		7:5	111b
Reserved		43:41h	31:8
Reserved	45:44h	15:0	FFFFh
2-2-2 Fast Read Wait Cycle Count	46h	20:16	00000b
2-2-2 Fast Read Mode bit Cycle Count		23:21	000b
2-2-2 Fast Read Instruction	47h	31:24	FFh
Reserved	49:48h	15:0	FFFFh
4-4-4 Fast Read Wait Cycle Count	4Ah	20:16	00100b
4-4-4 Fast Read Mode bit Cycle Count		23:21	010b
4-4-4 Fast Read Instruction	4Bh	31:24	EBh
Erase Type 1 Size (4KB)	4Ch	7:0	0Ch
Erase Type 1 Instruction	4Dh	15:8	20h
Erase Type 2 Size (32KB)	4Eh	23:16	0Fh
Erase Type 2 Instruction	4Fh	31:24	52h
Erase Type 3 Size (64KB)	50h	7:0	10h
Erase Type 3 Instruction	51h	15:8	D8h
Erase Type 4 Size (256KB)	52h	23:16	00h
Erase Type 4 Instruction	53h	31:24	FFh
Multiplier from typical erase time to maximum erase time	57:54h	3:0	0011b
Sector Type 1 ERASE time (typ)		8:4	00100b
		10:9	01b
Sector Type 2 ERASE time (typ)		15:11	00110b
		17:16	01b
Sector Type 3 ERASE time (typ)		22:18	01001b
		24:23	01b
Sector Type 4 ERASE time (typ)		29:25	00000b
		31:30	00b

Table 5.3 JEDEC Basic Flash Parameter Table (Continued)

Description		Address (Byte)	Address (Bit)	Data
Multiplier from typical time to maximum time for page or byte PROGRAM		58h	3:0	0010b
Page size			7:4	1000b
Page Program Typical time		5Ah:59h	12:8	11000b
			13	0b
Byte Program Typical time, first byte			17:14	0111b
			18	0b
Byte Program Typical time, additional byte			22:19	0000b
			23	0b
Chip Erase, Typical time	8Mb	5Bh	28:24	00111b
	4Mb		28:24	00011b
Units			30:29	01b
Reserved			31	1b
Prohibited Operations During Program Suspend		5Ch	3:0	1100b
Prohibited Operations During Erase Suspend			7:4	1110b
Reserved		5Eh:5Dh	8	1b
Program Resume to Suspend Interval			12:9	0110b
Suspend in-progress program max latency			17:13	01100b
			19:18	10b
Erase Resume to Suspend Interval			23:20	0110b
Suspend in-progress erase max latency		5Fh	28:24	01100b
			30:29	10b
Suspend /Resume supported			31	0b
Program Resume Instruction		60h	7:0	7Ah
Program Suspend Instruction		61h	15:8	75h
Resume Instruction		62h	23:16	7Ah
Suspend Instruction		63h	31:24	75h
Reserved		64h	1:0	11b
Status Register Polling Device Busy			7:2	111101b

Table 5.3 JEDEC Basic Flash Parameter Table (Continued)

Description	Address (Byte)	Address (Bit)	Data
Exit Deep Power-down to next operation delay	67h:65h	12:8	00010b
Exit Deep Power-down to next operation delay Units		14:13	01b
Exit Deep Power-down Instruction		22:15	ABh
Enter Deep Power-down Instruction		30:23	B9h
Deep Power-down Supported		31	0b
4-4-4 mode disable sequences (QPIDI)	69h:68h	3:0	1010b
4-4-4 mode enable sequences (QPIEN)		8:4	00100b
0-4-4 Mode Supported		9	1b
0-4-4 Mode Exit Method		15:10	110000b
0-4-4 Mode Entry Method:	6Ah	19:16	1100b
Quad Enable Requirements (QER)		22:20	010b
Hold or RESET Disable		23	0b
Reserved	6Bh	31:24	FFh
Volatile or Non-Volatile Register and Write Enable (WREN) Instruction for Status Register 1	6Ch	6:0	1100001b
Reserved		7	1b
Soft Reset and Rescue Sequence Support	6Eh:6Dh	13:8	110000b
Exit 4-Byte Addressing		23:14	110000000 0b
Enter 4-Byte Addressing	6Fh	31:24	10000000b

6. REGISTERS

The device has various sets of Registers: Status, Function, Read, Extended Read.
When the register is read continuously, the same data is output repeatedly until CE# goes HIGH.

6.1 STATUS REGISTER

Status Register Format and Status Register Bit Definitions are described in Table 6.1 & Table 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 Table 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

Note1: 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, or Write/Set Non-Volatile/OTP Register 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 an 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 bit 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 and the Set Volatile Extended Read Register 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
PPQ	32h/38h	Quad Input Page Program
SER	D7h/20h	Sector Erase 4KB
BER32 (32KB)	52h	Block Erase 32KB
BER64 (64KB)	D8h	Block Erase 64KB
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

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 is tied directly to the power supply.

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

Status Register Bits				BP Table, Protected Memory Area	
BP3	BP2	BP1	BP0	8Mb	4Mb
0	0	0	0	0 (None)	0 (None)
0	0	0	1	1(1 block : 15th)	1(1 block : 7th)
0	0	1	0	2(2 blocks : 14th – 15th)	2(2 blocks : 6th – 7th)
0	0	1	1	3(4 blocks : 12th – 15th)	3(4 blocks : 4th – 7th)
0	1	0	0	4(8 blocks : 8th – 15th)	4(8 blocks : 2nd – 7th)
0	1	0	1	5(12 blocks : 4th – 15th)	5(7 blocks : 1st – 7th)
0	1	1	0	6(14 blocks : 2nd – 15th)	6~7 (8 blocks: All Blocks)
0	1	1	1	7(16 blocks: All Blocks)	
1	0	0	0	8 (None)	8 (None)
1	0	0	1	9(1 block : 0th)	9(1 block : 0th)
1	0	1	0	10(2 blocks : 0th – 1st)	10(2 blocks : 0th – 1st)
1	0	1	1	11(4 blocks : 0th – 3rd)	11(4 blocks : 0th – 3rd)
1	1	0	0	12(8 blocks : 0th – 7th)	12(6 blocks : 0th – 5th)
1	1	0	1	13(12 blocks : 0th– 11th)	13(7 blocks : 0th– 6th)
1	1	1	0	14(14 blocks : 0th– 13th)	14, 15 (8 blocks : All Blocks)
1	1	1	1	15 (16 blocks : All Blocks)	

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	Reserved	Reserved
Default	0	0	0	0	0	0	0	0 ⁽¹⁾

Note: 1. Default value = 1 for 24BGA with Dedicated RESET#.

Table 6.6 Function Register Bit Definition

Bit	Name	Definition	Read /Write	Type
Bit 0	Reserved	Reserved.	R	Reserved
Bit 1	Reserved	Reserved	R	Reserved
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: 1. Once OTP bits of Function Register are written to "1", it cannot be modified to "0" any more.

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 set to "1", the Information Row can't be programmed. Once it set to "1", it cannot be changed back to "0" 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". When QE=1 or in QPI mode, P7 bit setting will be ignored since the pin becomes IO3..

The Dummy Cycle bits (P6, P5, P4, P3) define how many dummy cycles are used during various READ modes. The wrap selection bits (P2, P1, P0) define burst length with an enable bit.

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 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# pin selection Bit: "0" indicates the HOLD# pin is selected (default) "1" indicates the RESET# pin 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

P[6:3]	Dummy Cycles ^{2,3}	Fast Read ⁵ 0Bh		Fast Read Dual Output 3Bh	Fast Read Dual IO BBh	Fast Read Quad Output 6Bh	Fast Read Quad IO EBh	FRDTR 0Dh	FRDDTR BDh	FRQDTR EDh
		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	80/69MHz	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

Notes:

1. Default dummy cycles are as follows.

Operation	Command		Dummy Cycles		Comment
	Normal mode	DTR mode	Normal mode	DTR mode	
Fast Read (SPI mode)	0Bh	0Dh	8	8	RDUID, IRRD instructions are also applied.
Fast Read (QPI mode)	0Bh	0Dh	6	6	
Fast Read Dual Output	3Bh	-	8	-	
Fast Read Dual IO SPI	BBh	BDh	4	4	
Fast Read Quad Output	6Bh	-	8	-	
Fast Read Quad IO (SPI mode)	EBh	EDh	6	6	
Fast Read Quad IO (QPI mode)	EBh	EDh	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 bit cycles are same, then X must be Hi-Z.

4. QPI mode is not available for FRDDTR command.

5. RDUID, IRRD instructions are also applied.

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	0	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 Program operation failure or protection error	R	Volatile
EB3	E_ERR	Erase Error Bit: "0" indicates no error "1" indicates a Erase 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/PPQ	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.
WRSR/WRABR/SRPNV/ SERPNV	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/BER32K/BER64K/CER/ IRER	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 Block Protection bits (BP3~BP0) are not 0. .

Notes:

1. OTP bits in the Function 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 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 ECC REGISTER

Related Related Commands: Read ECC Register (RDECC B3h), Write ECC Register (WRECC B5h), Clear ECC Register (CLECC, B6h).

Default value for all register bits are “0”

Table 6.16 ECC Register Format

Address								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000000h	ERRBECC	ERRBENB	ECCCOUNTER				ECCEVENT	ECCONOFF
000001h	Reserved						ECCSTAT	IPA_ECCB
000002h	ECCFCA [A23:A16]							
000003h	ECCFCA [A15:A8]							
000004h	ECCFCA [A7:A0]							

Table 6.17 ECC Register Bit Definition

Address	Bit	Name	Definition		Read-Write	Type
000000h	0	ECCONOFF	ECC ON/OFF bit: "0" indicates ECC is ON (default) "1" indicates ECC is OFF		R/W	Volatile
	1	ECCEVENT	ECC Event bit: "0" indicates NO ECC correction event (default) "1" indicates YES ECC correction event		R	Volatile
	5:2	ECCCOUNTER	Store cumulative ECC correction occurrence. Max. 15 ECC correction occurrence can be stored, and stays 15 after further occurrence. "0000" is the default value.		R	Volatile
	6	ERRBENB	0 = ERR# OFF (Default), 1 = ERR# ON		R/W	Volatile
	7	ERRBECC	0 = ERR# LOW indicates 2-bit detection (Default) 1 = ERR# LOW indicates 1-bt correction. ERR# LOW behavior for 1-bit correction or 2-bit detection out of ECC event. Also it will determine ECC error type for ECCFCA bits. Only valid when ECCONOFF bit is 0.		R/W	Volatile
000001h	7:2	Reserved	Reserved		R	Volatile
	1	IPA_ECCB	Incremental Programming Attempt within ECC boundary bit: "0" indicates NO IPA within ECC Boundary (default) "1" indicates YES IPA with ECC Boundary. Incremental Programming within ECC boundary is not allowed, and automatically ignored when ECC is ON.		R	Volatile
	0	ECCSTAT	0 = No error (default), 1 = 2-bit Error Detection		R	Volatile
000002h	7:0	ECCFCA [23:16]	Chunk address, [A23:A16]	1 st ECC Event occurred chunk address; 1-bit correction event or 2-bit detection event will be selected by setting of ERRBECC bit (bit 7 of address 000000h)	R	Volatile
000003h	7:0	ECCFCA [15:8]	Chunk address, [A15:A8]		R	Volatile
000004h	7:0	ECCFCA [7:0]	Chunk address, [A7:A4]		R	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 9.8 POWER-UP AND POWER-DOWN. All write sequence will be ignored when V_{CC} 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 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 a software write protection feature. The Block Protection (BP3, BP2, BP1, BP0) bits allow part or the whole memory area to be write-protected.

8. DEVICE OPERATION

The device utilizes an 8-bit instruction register. Refer to Table 8.1. 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, data bytes, or both address bytes and 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.

Table 8.1 Instruction Set

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
NORD	Normal Read Mode	SPI	03h	A <23:16>	A <15:8>	A <7:0>	Data out		
FRD	Fast Read Mode	SPI QPI	0Bh	A <23:16>	A <15:8>	A <7:0>	Dummy ⁽¹⁾ Byte	Data out	
FRDIO	Fast Read Dual I/O	SPI	BBh	A <23:16> Dual	A <15:8> Dual	A <7:0> Dual	AXh ^{(1),(2)} Dual	Dual Data out	
FRDO	Fast Read Dual Output	SPI	3Bh	A <23:16>	A <15:8>	A <7:0>	Dummy ⁽¹⁾ Byte	Dual Data out	
FRQIO	Fast Read Quad I/O	SPI QPI	EBh	A <23:16> Quad	A <15:8> Quad	A <7:0> Quad	AXh ^{(1),(2)} Quad	Quad Data out	
FRQO	Fast Read Quad Output	SPI	6Bh	A <23:16>	A <15:8>	A <7:0>	Dummy ⁽¹⁾ Byte	Quad Data out	
FRDTR	Fast Read DTR Mode	SPI QPI	0Dh	A <23:16>	A <15:8>	A <7:0>	Dummy ⁽¹⁾ Byte	Dual Data out	
FRDDTR	Fast Read Dual I/O DTR	SPI	BDh	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	SPI QPI	EDh	A <23:16>	A <15:8>	A <7:0>	AXh ^{(1),(2)} Quad	Quad Data out	
PP	Input Page Program	SPI QPI	02h	A <23:16>	A <15:8>	A <7:0>	PD (256byte)		
PPQ	Quad Input Page Program	SPI	32h 38h	A <23:16>	A <15:8>	A <7:0>	Quad PD (256byte)		
SER	Sector Erase	SPI QPI	D7h 20h	A <23:16>	A <15:8>	A <7:0>			
BER32 (32KB)	Block Erase 32Kbyte	SPI QPI	52h	A <23:16>	A <15:8>	A <7:0>			
BER64 (64KB)	Block Erase 64Kbyte	SPI QPI	D8h	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	SR					
WRSR	Write Status Register	SPI QPI	01h	WSR Data					

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
RDFR	Read Function Register	SPI QPI	48h	Data out					
WRFR	Write Function Register	SPI QPI	42h	WFR Data					
QPIEN	Enter QPI mode	SPI	35h						
QPIDI	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					
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						

Instruction Name	Operation	Mode	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
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	
SECUN-LOCK	Sector Unlock	SPI QPI	26h	A <23:16>	A <15:8>	A <7:0>			
SECLOCK	Sector Lock	SPI QPI	24h						
RDECC	Read ECC register	SPI QPI	B3h	A <23:16>	A <15:8>	A <7:0>	Dummy Byte	Data out	
WRECC	Write ECC register	SPI QPI	B5h	Data in					
CLECC	Clear ECC Register	SPI QPI	B6h						

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”.

8.1 NORMAL READ OPERATION (NORD, 03h)

The NORMAL READ (NORD) instruction is used to read memory contents at a maximum frequency of 50MHz.

The NORD instruction code is transmitted via the SI line, followed by three address bytes (A23 - A0) of the first memory location to be read. A total of 24 address bits are shifted in, but only A_{VMSB} (Valid Most Significant Bit) - A₀ are decoded. The remaining bits (A23 - 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 8.2 for the related Address Key.

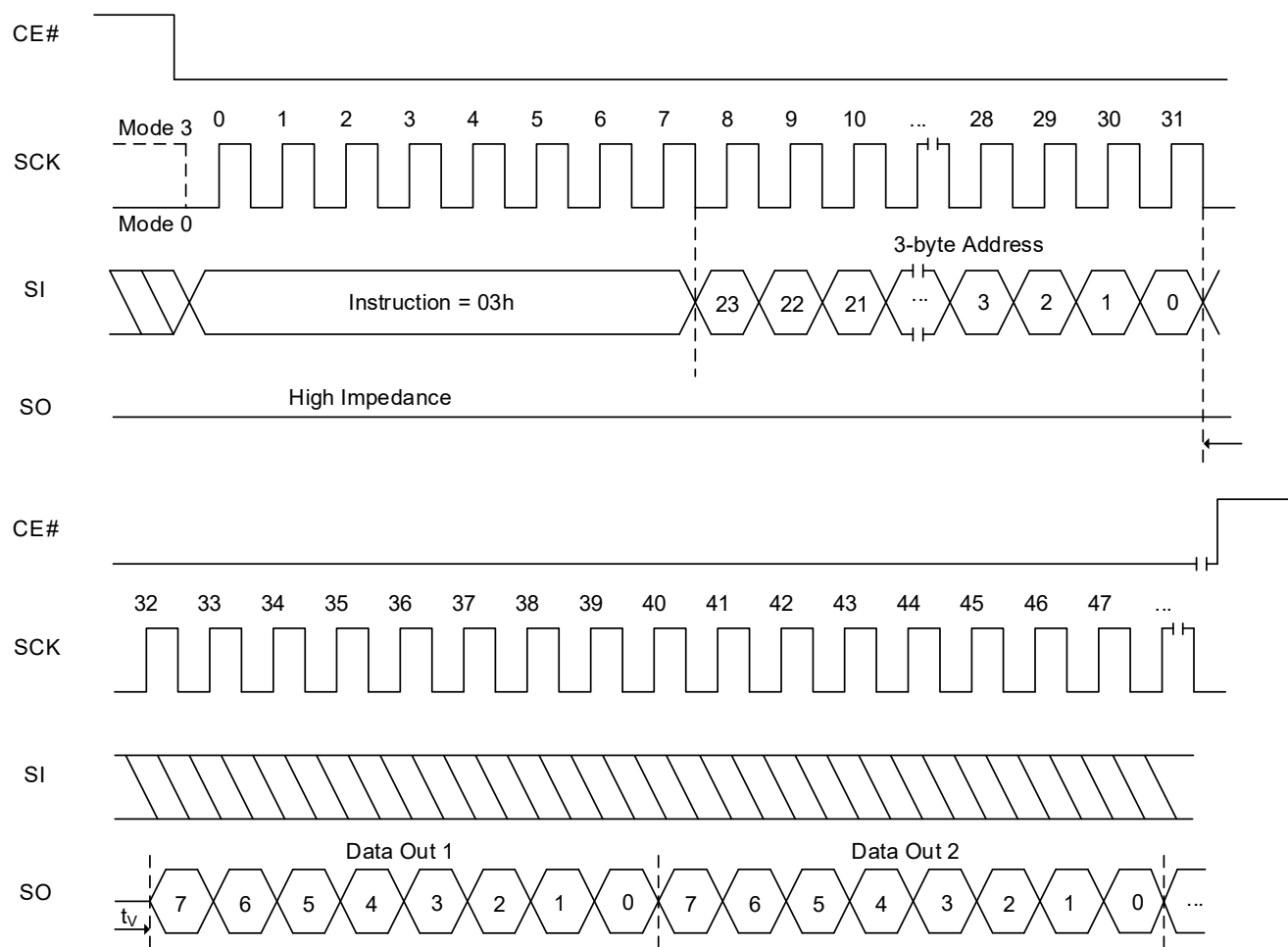
The first byte data (D7 - D0) is shifted out on the SO line, MSB first. A single byte of data, or up to the whole memory array, can be read out in one NORMAL READ instruction. The address is automatically incremented by one after each byte of data is shifted out. The read operation can be terminated 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 8.2 Address Key

Address	8Mb	4Mb
A _{VMSB} -A ₀	A19-A0 (A23-A20=X)	A18-A0 (A23-A19=X)

Note: X=Don't Care

Figure 8.1 Normal Read Sequence


8.2 FAST READ OPERATION (FRD, 0Bh)

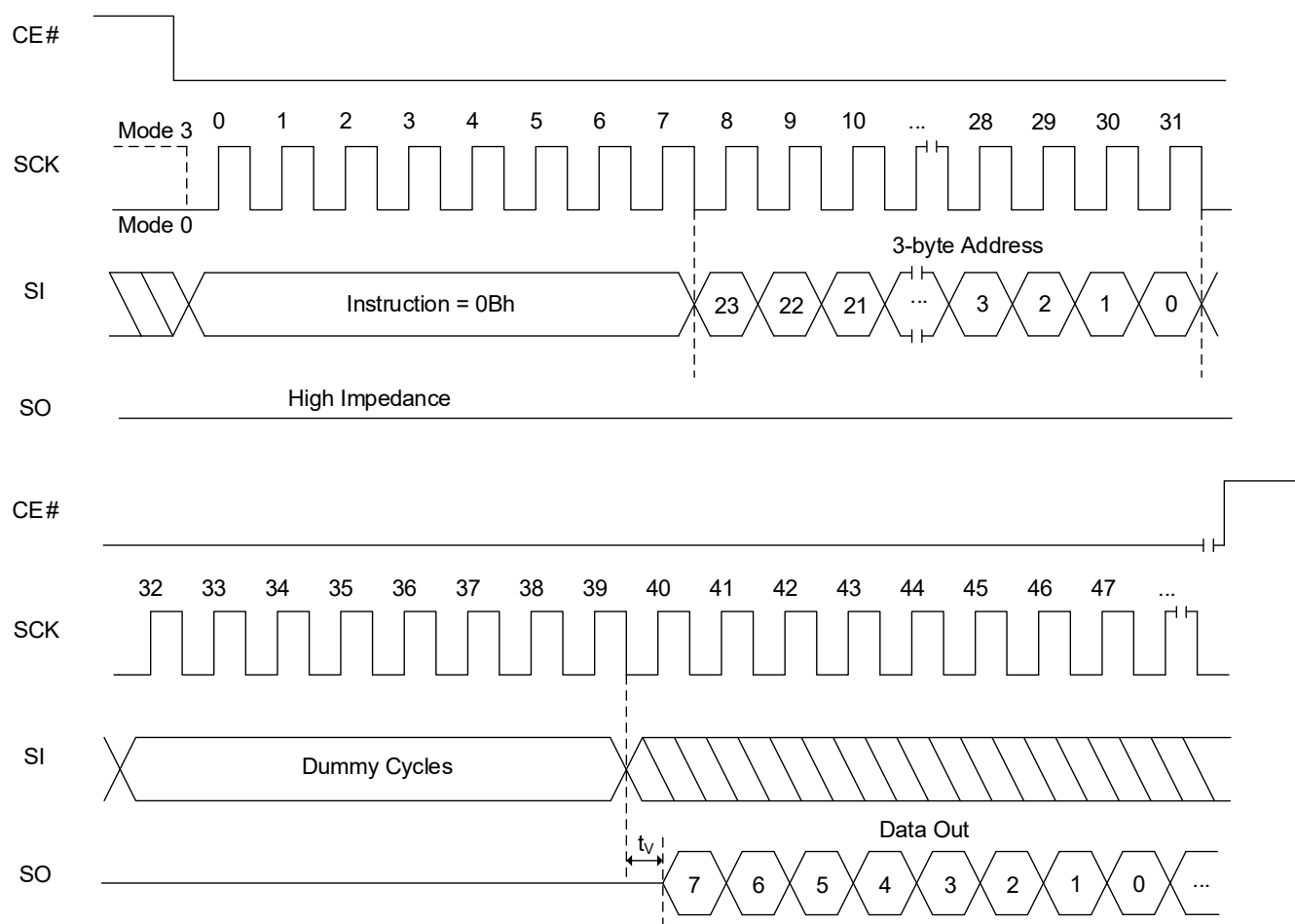
The FAST READ (FRD) instruction is used to read memory data.

The FAST READ 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 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 8.2 Fast Read Sequence



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)

The FAST READ (FRD) instruction is used also in QPI mode to read memory data.

The FAST READ instruction code (2 clocks) is followed by three address bytes (A23-A0 — 6 clocks) and dummy cycles (configurable, default is 6 cycles), 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 instruction. The FAST READ instruction in QPI mode is terminated by driving CE# high (VIH).

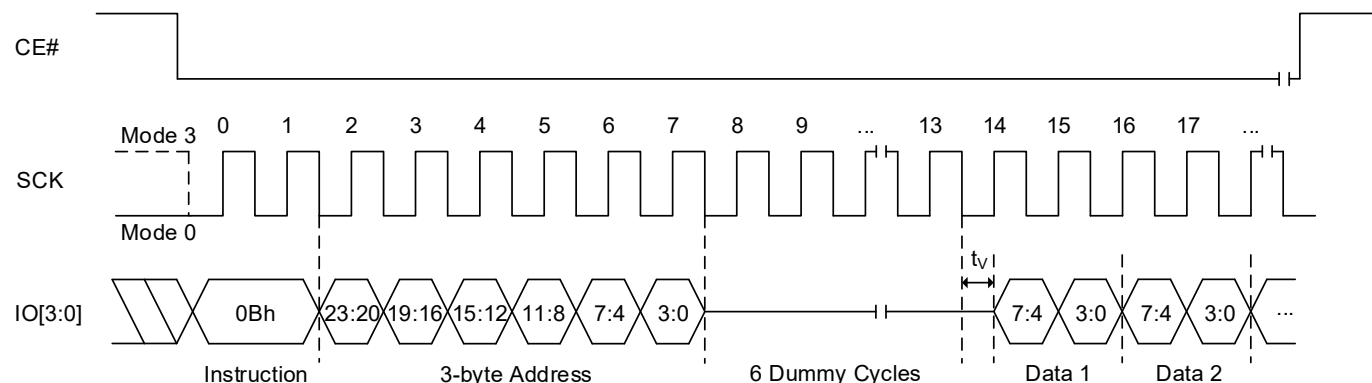
If the FAST READ instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored without affecting the current cycle.

The Fast Read sequence in QPI mode is also applied to the commands in the following table 8.3.

Table 8.3 Address Key

Instruction Name	Operation	Hex Code
FRQIO	Fast Read Quad I/O	EBh
RDUID	Read Unique ID	4Bh
IRRD	Read Information Row	68h
RDECC	Read ECC	B3h

Figure 8.3 Fast Read Sequence In QPI Mode



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

8.3 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.

8.4 FAST READ DUAL I/O OPERATION (FRDIO, BBh)

The FRDIO 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.

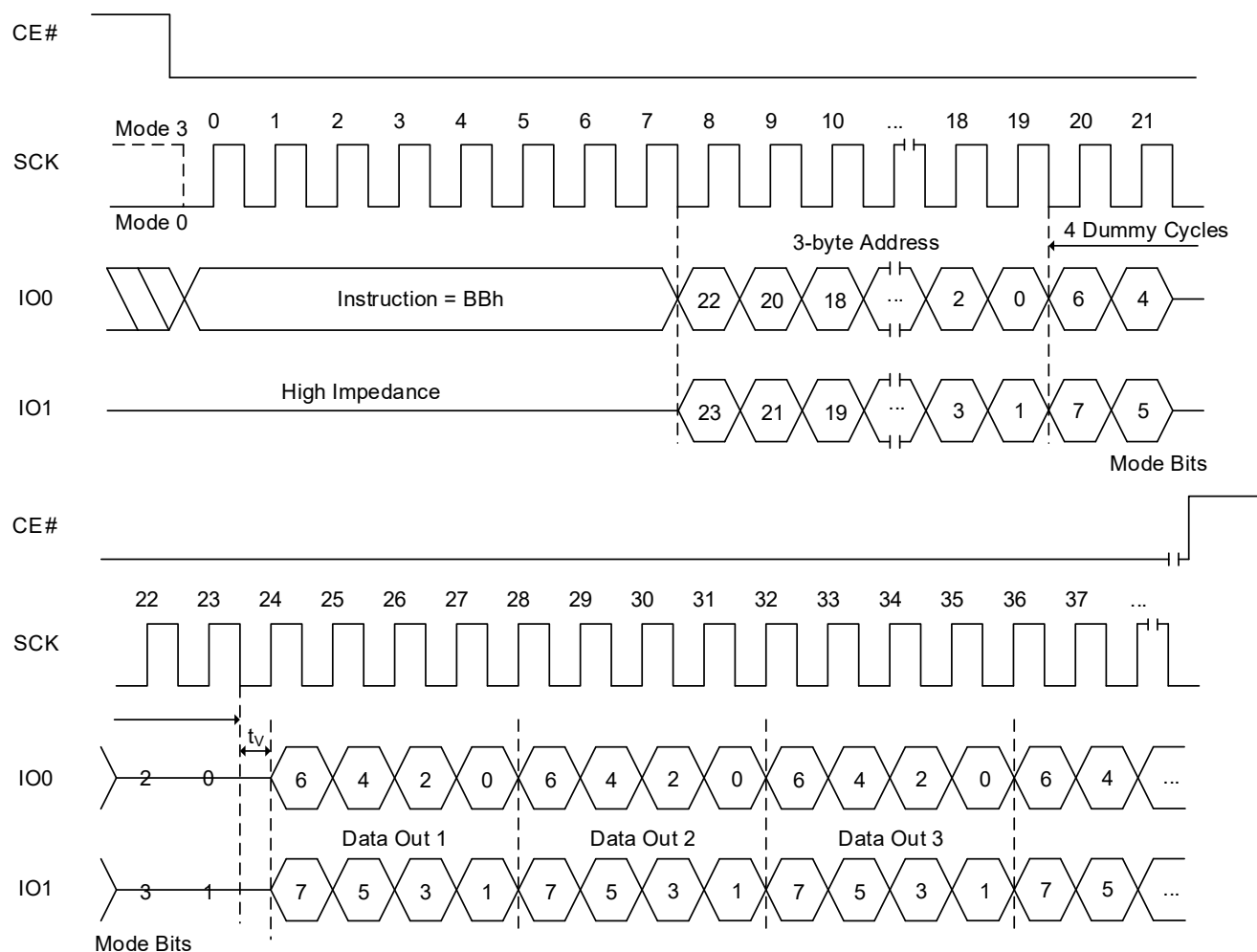
The FRDIO instruction code is followed by three address bytes (A23 – A0) and dummy cycles (configurable, default is 4 clocks), transmitted via the IO1 and IO0 lines, with each pair of bits latched-in during the rising edge of SCK. The address MSB is input on IO1, the next bit on IO0, and this shift pattern 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 MSB is output on IO1, while simultaneously the second bit is output on IO0. Figure 8.4 illustrates the timing sequence.

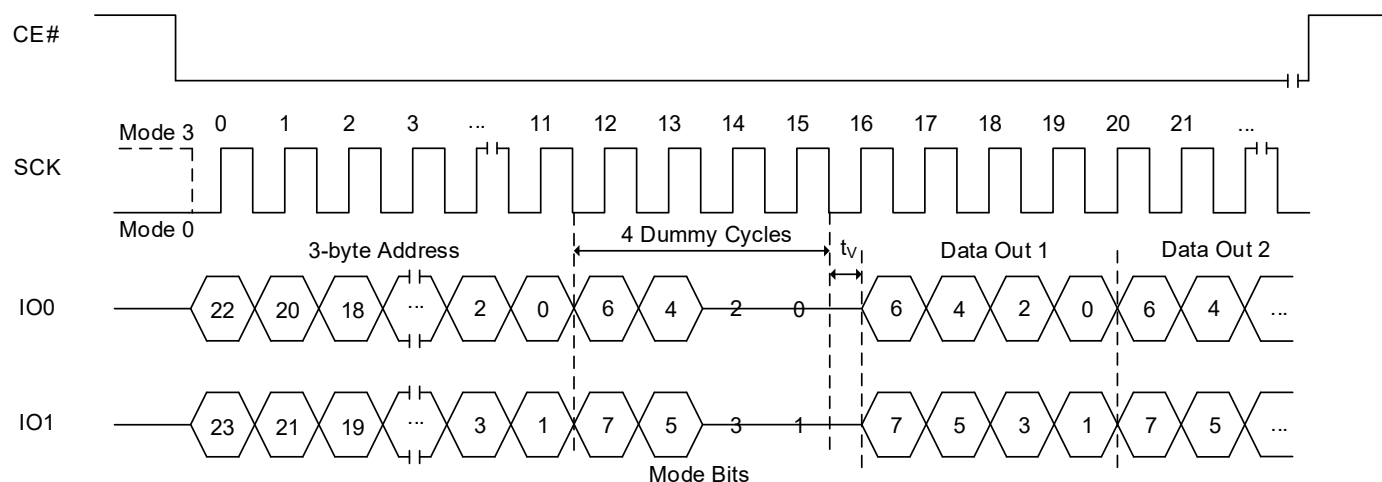
The first byte addressed can be at any memory location. The address is automatically incremented by one after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRDIO instruction. FRDIO 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 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 execution skips command code. It saves cycles as described in Figure 8.5. 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 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 8.4 Fast Read Dual I/O Sequence (with command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

Figure 8.5 Fast Read Dual I/O AX Read Sequence (without command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

8.5 FAST READ DUAL OUTPUT OPERATION (FRDO, 3Bh)

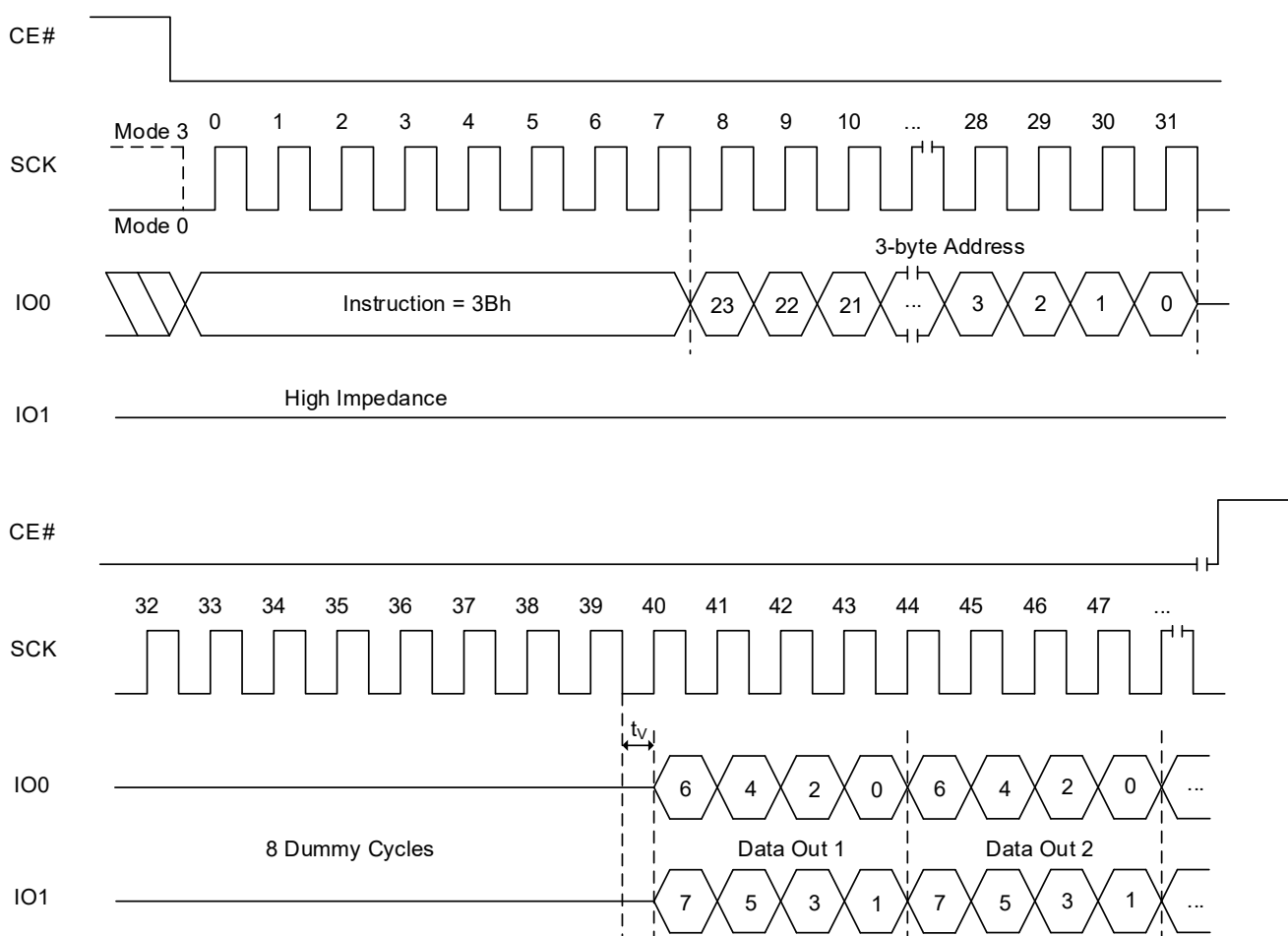
The FRDO instruction is used to read memory data on two output pins.

The FRDO instruction code is followed by three address bytes (A23 – A0) 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 instruction. The FRDO instruction is terminated by driving CE# high (VIH).

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

Figure 8.6 Fast Read Dual Output Sequence



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

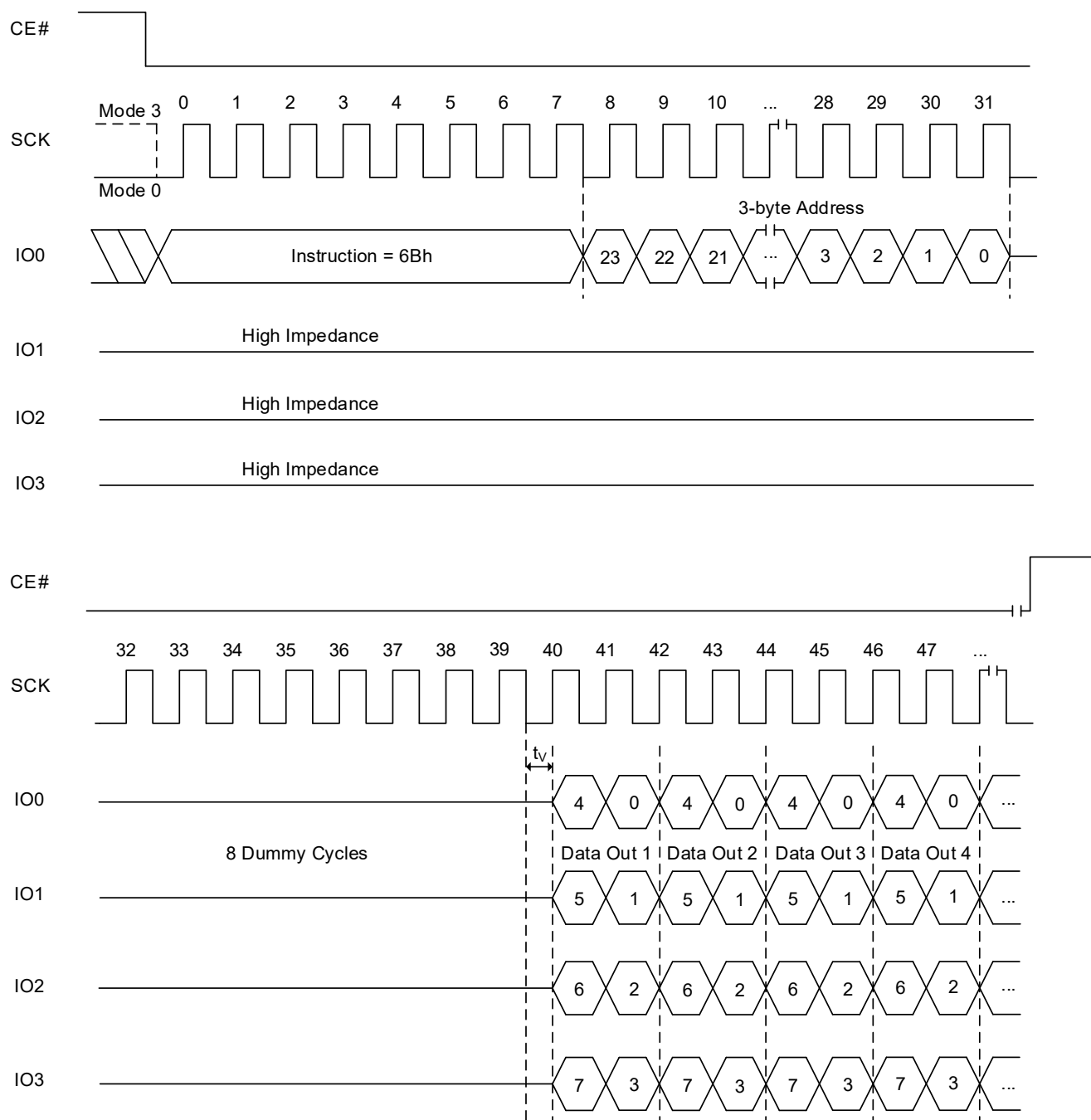
8.6 FAST READ QUAD OUTPUT OPERATION (FRQO, 6Bh)

The FRQO instruction is used to read memory data on four output pins.

The FRQO instruction code is followed by three address bytes (A23 – A0) 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 instruction. FRQO instruction is terminated by driving CE# high (VIH).

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

Figure 8.7 Fast Read Quad Output Sequence


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

8.7 FAST READ QUAD I/O OPERATION (FRQIO, EBh)

The FRQIO 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.

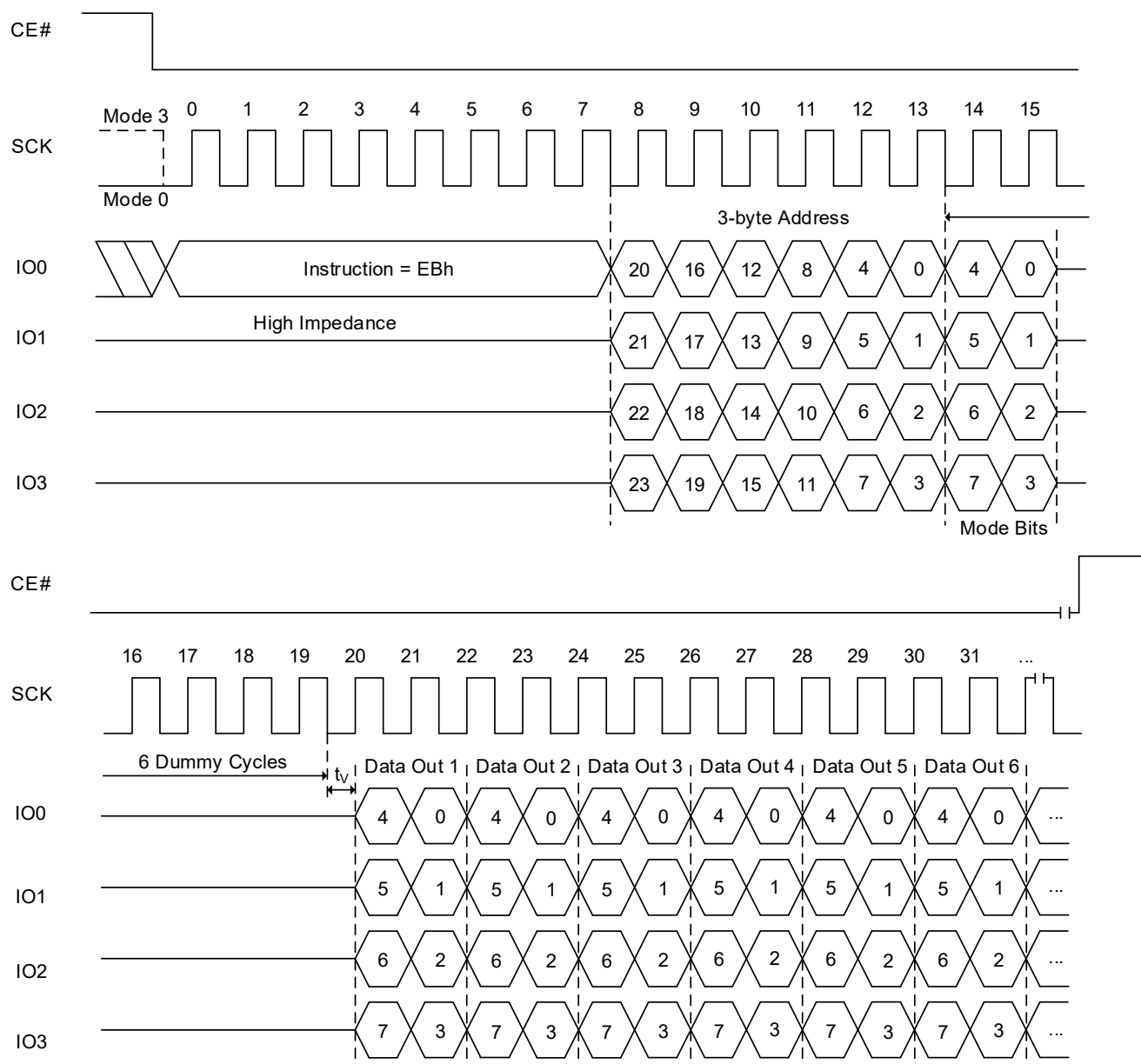
The FRQIO instruction code is followed by three address bytes (A23 – A0) and dummy cycles (configurable, default is 6 clocks), transmitted via the IO3, IO2, IO1 and IO0 lines, with each group of four bits latched-in during the rising edge of SCK. The address of MSB inputs on IO3, the next bit on IO2, 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. Figure 8.8 illustrates the timing sequence.

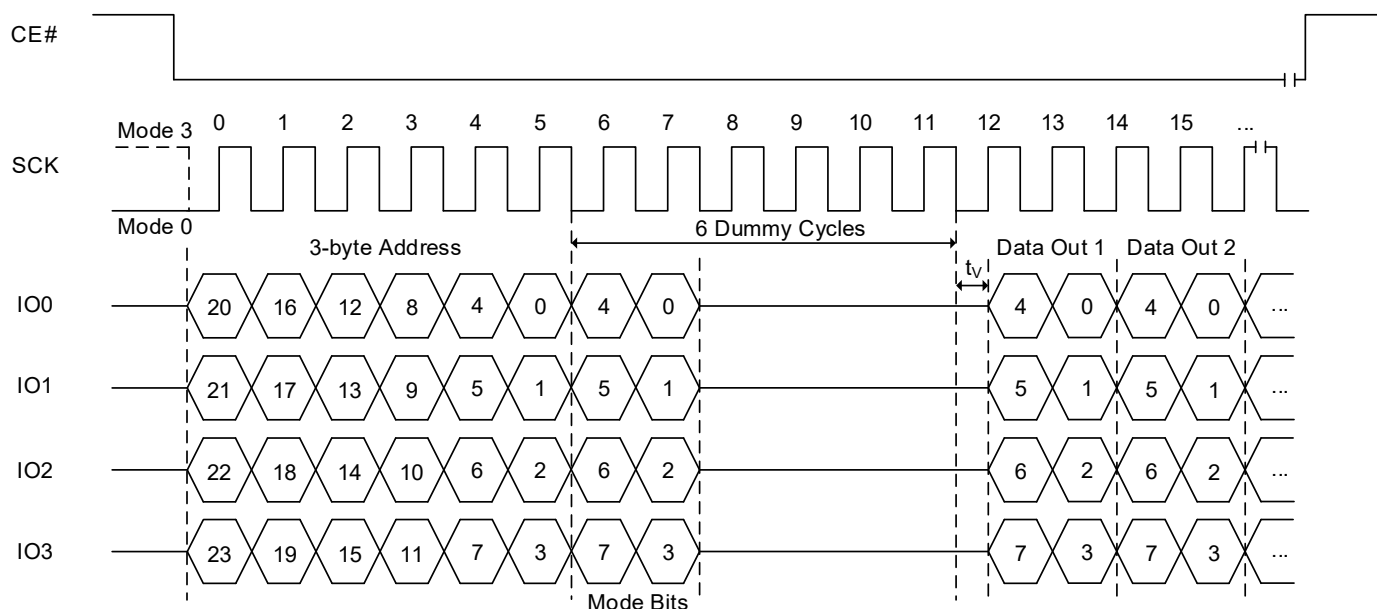
The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRQIO instruction. FRQIO 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 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 execution skips command code. It saves cycles as described in Figure 8.9. 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 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 8.8 Fast Read Quad I/O Sequence (with command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

Figure 8.9 Fast Read Quad I/O AX Read Sequence (without command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

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

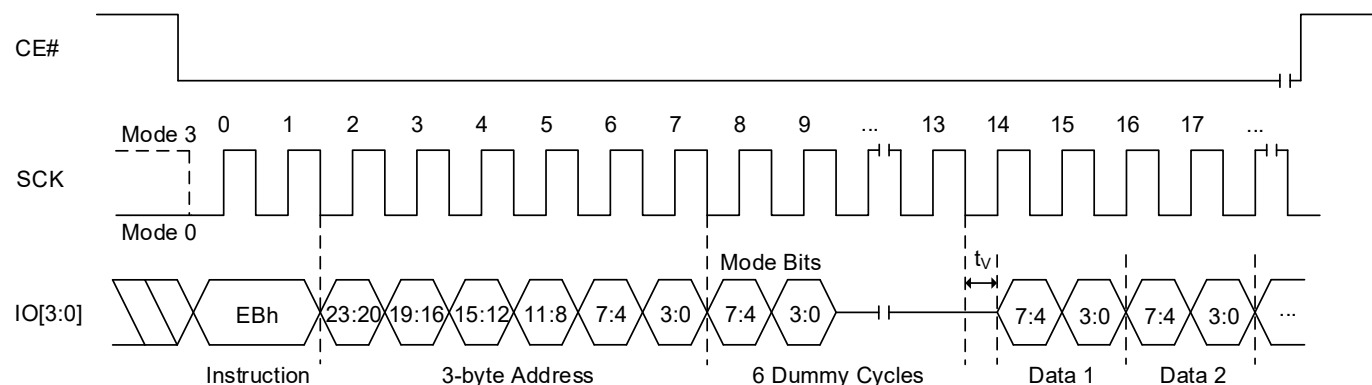
The FRQIO instruction is also used in QPI mode to read memory data.

The FRQIO instruction in QPI mode utilizes all four IO lines to input the instruction code so that only two clocks are required, while the FRQIO 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 FRQIO instruction in QPI mode. In addition, subsequent address and data out are shifted in/out via all four IO lines like the FRQIO instruction. In fact, except for the command cycle, the FRQIO operation in QPI mode is exactly same as the FRQIO 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 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 execution skips command code. It saves cycles as described in Figure 8.9. 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 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 8.10 Fast Read Quad I/O Sequence in QPI Mode



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

8.8 PAGE PROGRAM OPERATION (PP, 02h)

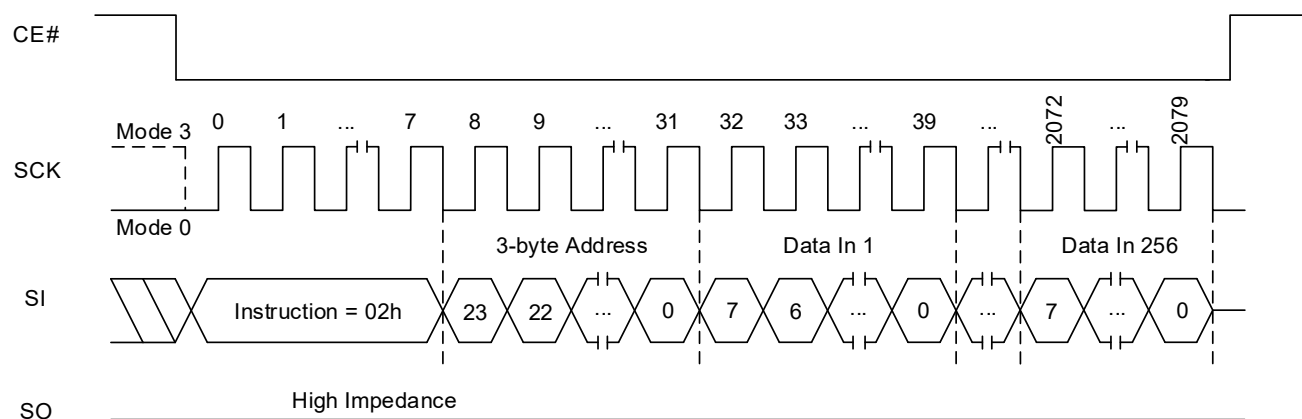
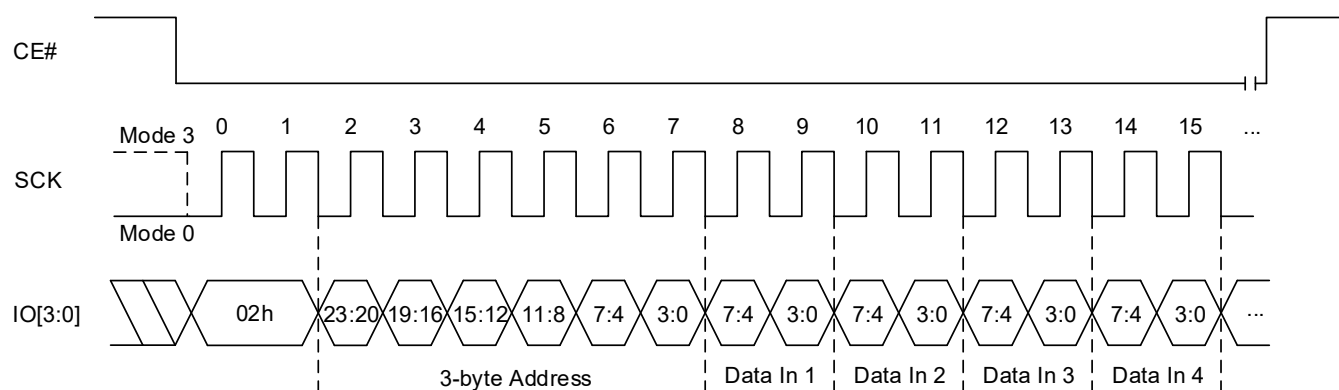
The Page Program (PP) instruction allows up to 256 bytes data to be programmed into memory in a single operation. The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection (BP3, BP2, BP1, BP0) bits. A PP instruction which attempts to program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

The PP instruction code, three address bytes 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 instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. The progress or completion of the program operation can be determined by reading the WIP bit in Status Register via a RDSR instruction. 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. When ECC is ON, ECC protection applied on each 32-bit boundary, so the device only allows a single programming operation to each 4-byte boundary between erase operations. Incremental programming is not allowed and ignored.

Figure 8.11 Page Program Sequence in SPI Mode

Figure 8.12 Page Program Sequence in QPI Mode


8.9 QUAD INPUT PAGE PROGRAM OPERATION (PPQ, 32h/38h)

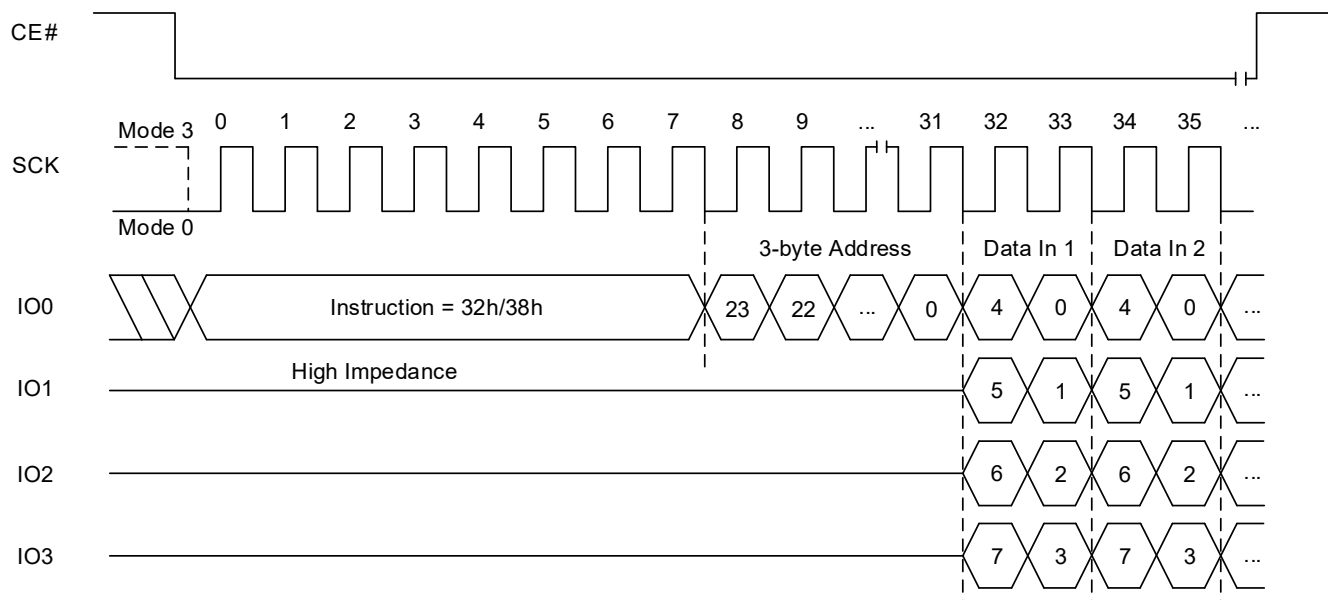
The Quad Input Page Program instruction allows up to 256 bytes data to be programmed into memory in a single operation with four pins (IO0, IO1, IO2 and IO3). The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection (BP3, BP2, BP1, BP0) bits. 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.

The Quad Input Page Program instruction code, three address bytes and program data (1 to 256 bytes) are input via the four pins (IO0, IO1, IO2 and IO3). 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. The progress or completion of the program operation can be determined by reading the WIP bit in Status Register via a RDSR instruction. 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. The same byte location or page may be programmed more than once, to incrementally change “1”s to “0”s. An erase operation is required to change “0”s to “1”s.

Figure 8.13 Quad Input Page Program Sequence



8.10 ERASE OPERATION

The Erase command sets all bits in the addressed sector or block to “1”s.

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.

8.11 SECTOR ERASE OPERATION (SER, D7h/20h)

A Sector Erase (SER) instruction erases a 4 Kbyte sector before the execution of a SER 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.

A SER instruction is entered, after CE# is pulled low to select the device and stays low during the entire instruction sequence. The SER instruction code, and three address bytes 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.

The progress or completion of the erase operation can be determined by reading the WIP bit in the Status Register using a RDSR instruction.

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 8.14 Sector Erase Sequence In QPI Mode

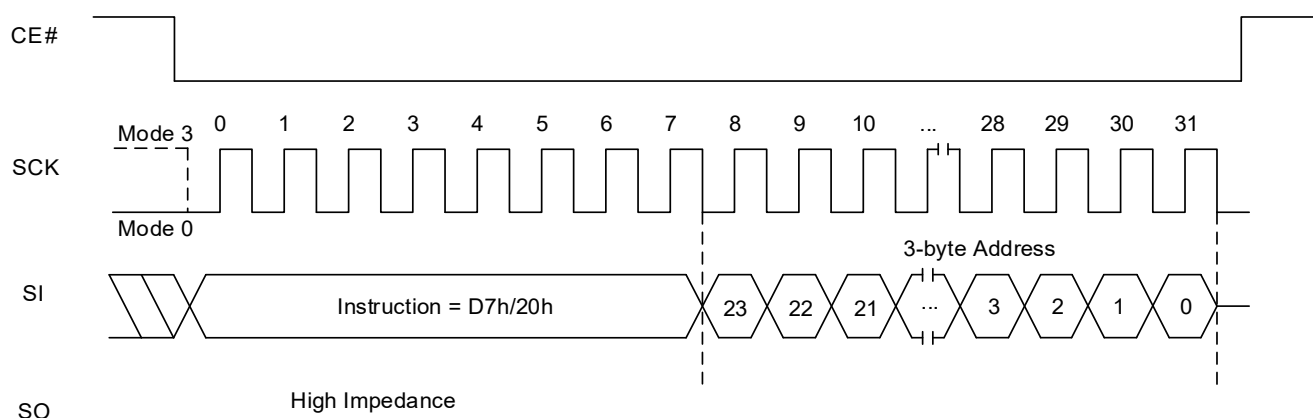
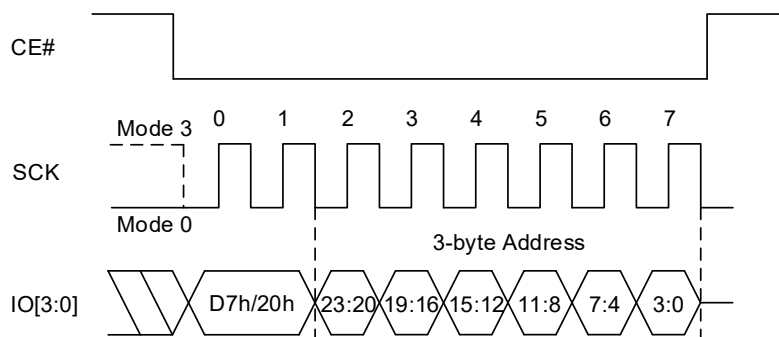


Figure 8.15 Sector Erase Sequence In QPI Mode



8.12 BLOCK ERASE OPERATION (BER32K:52h, BER64K:D8h)

A Block Erase (BER) instruction erases a 32/64Kbyte 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.

The BER instruction code and three address bytes 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 8.16 Block Erase (64K) Sequence In SPI Mode

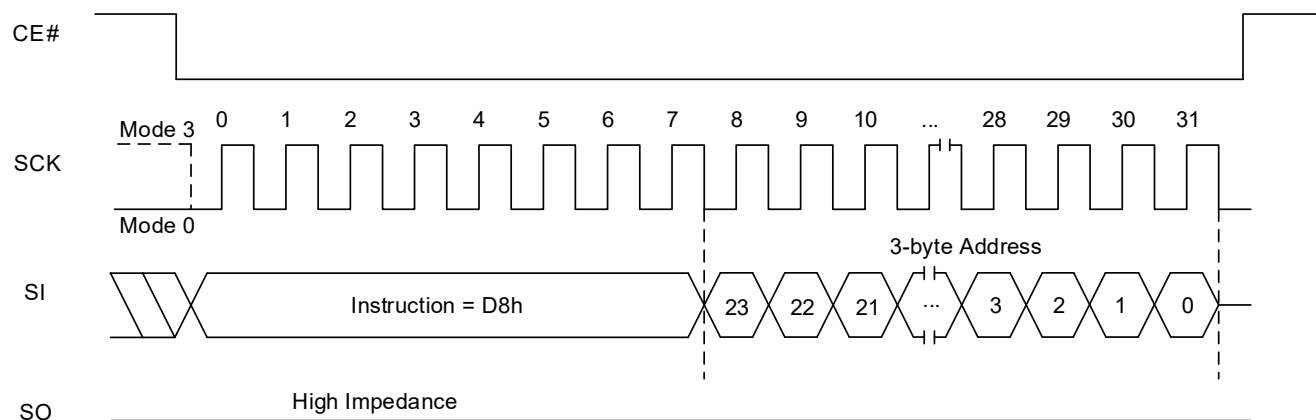


Figure 8.17 Block Erase (64K) Sequence In QPI Mode

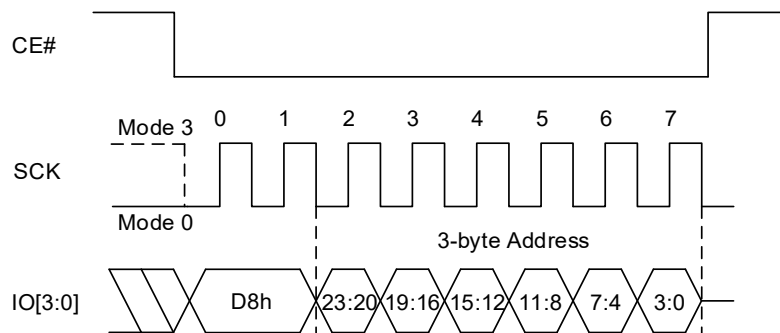
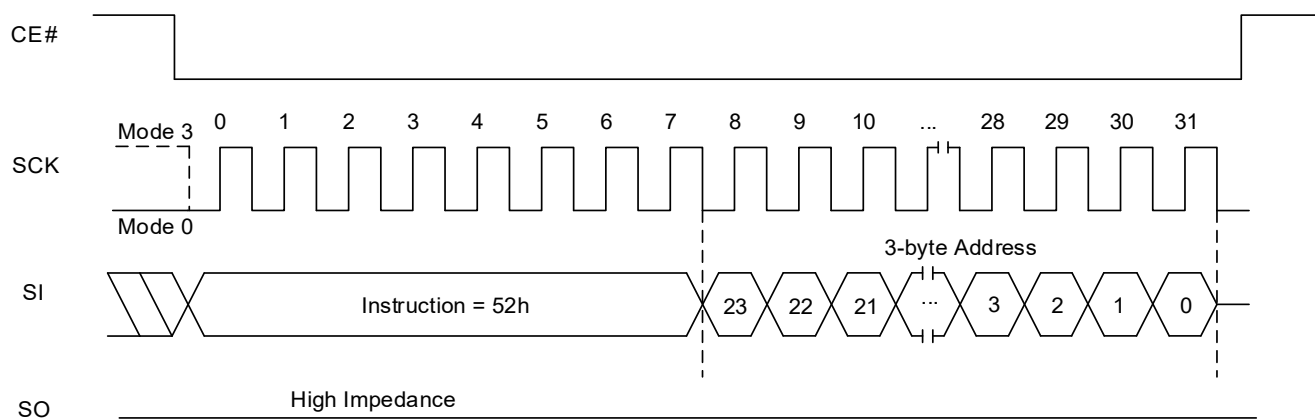
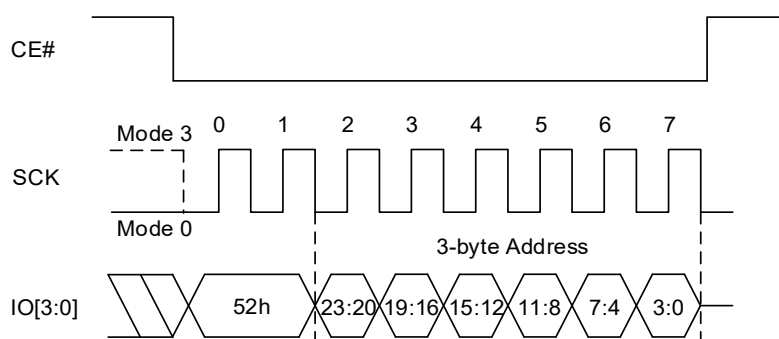


Figure 8.18 Block Erase (32K) Sequence In SPI Mode

Figure 8.19 Block Erase (32K) Sequence In QPI Mode


8.13 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.

Figure 8.20 Chip Erase Sequence In SPI Mode

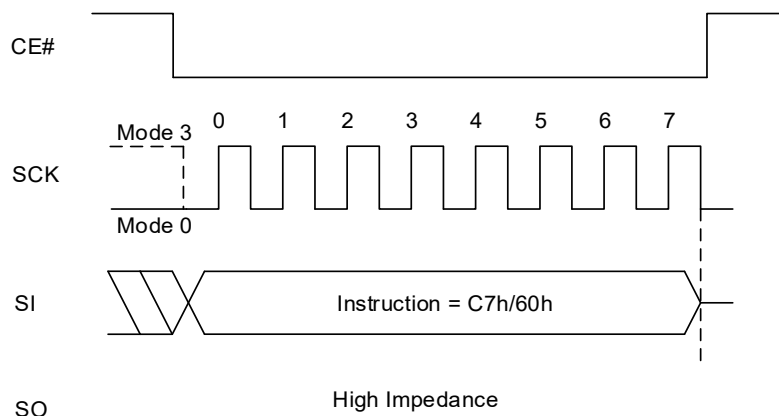
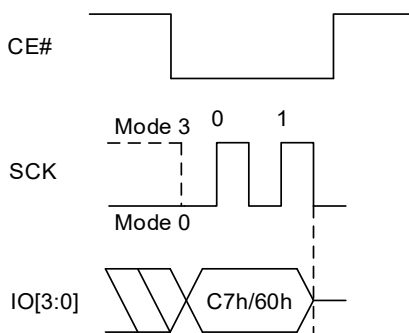


Figure 8.21 Chip Erase Sequence In QPI Mode



8.14 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, and Set non-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 8.22 Write Enable Sequence In SPI Mode

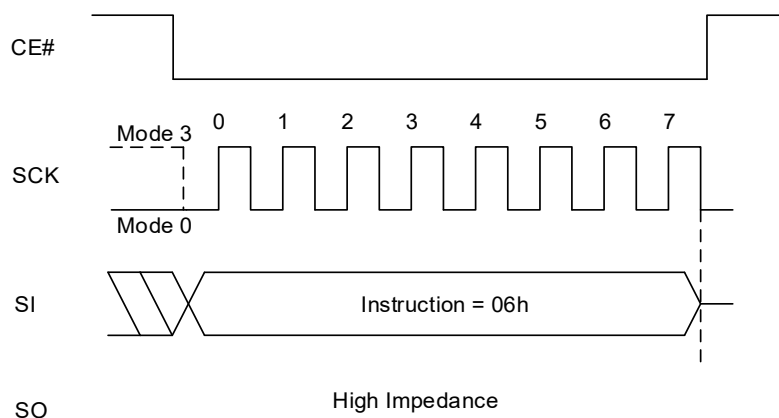
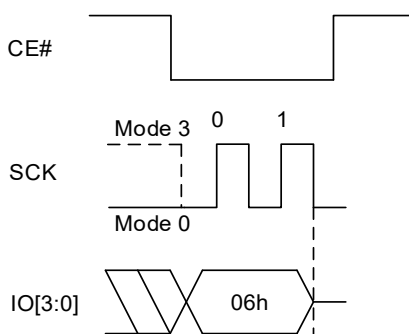


Figure 8.23 Write Enable Sequence In QPI Mode



8.15 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 8.24 Write Disable Sequence In SPI Mode

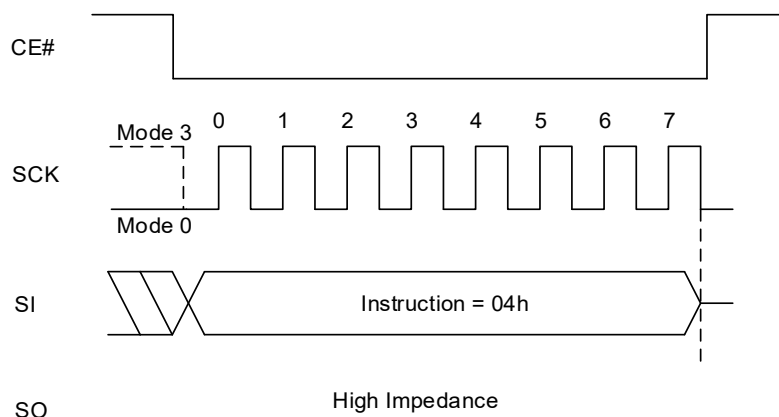
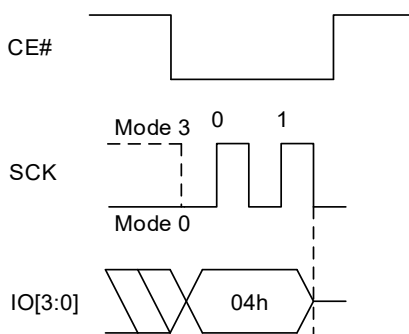


Figure 8.25 Write Disable Sequence In QPI Mode



8.16 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, RDSR instruction can be used to check the progress or completion of an operation by reading the WIP bit of Status Register.

Figure 8.26 Read Status Register Sequence In SPI Mode

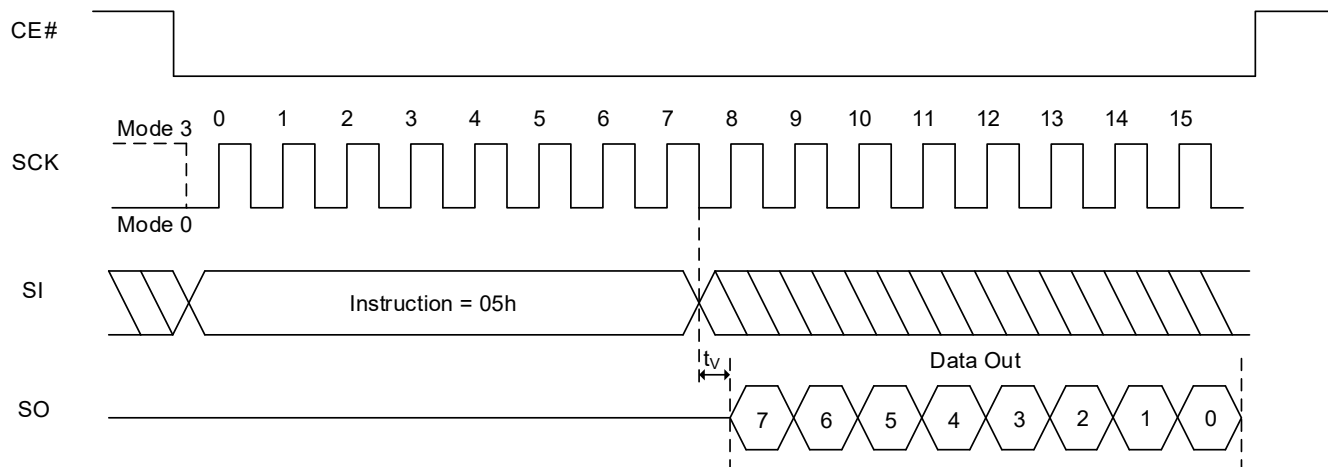
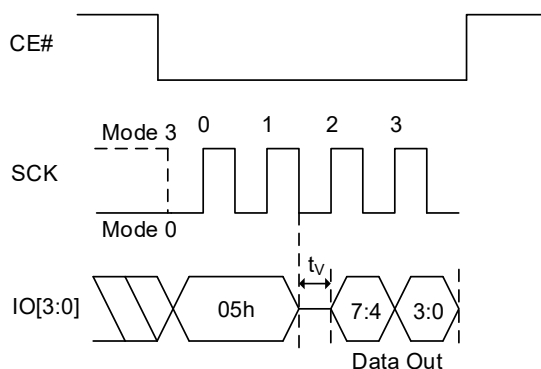


Figure 8.27 Read Status Register Sequence In QPI Mode



8.17 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 8.28 Write Status Register Sequence In SPI Mode

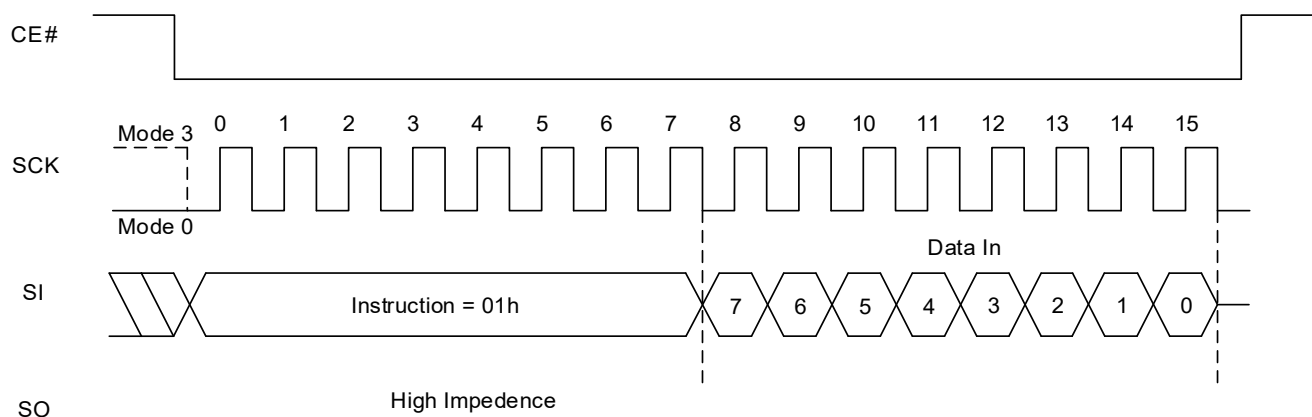
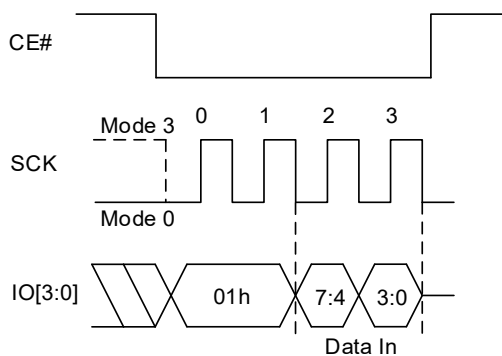


Figure 8.29 Write Status Register Sequence In QPI Mode



8.18 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 8.30 Read Function Register Sequence In SPI Mode

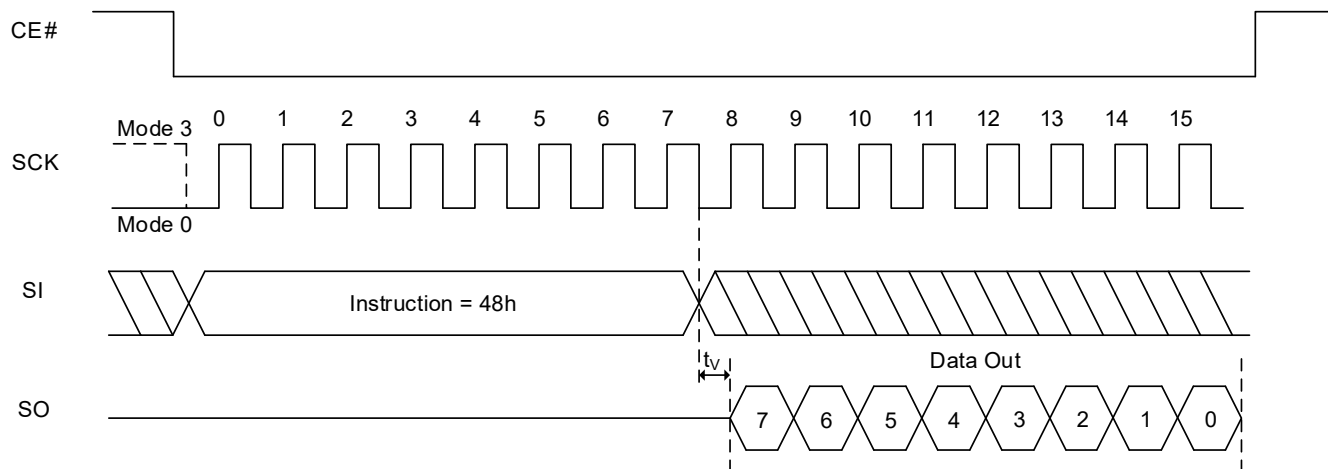
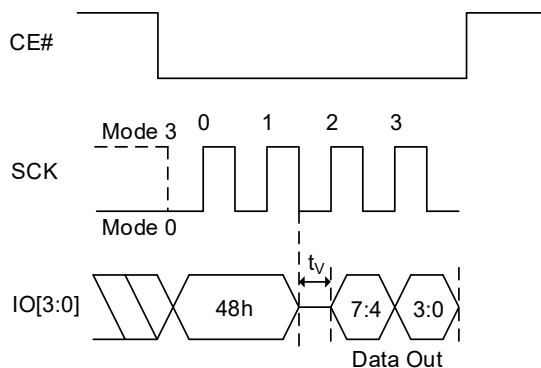


Figure 8.31 Read Function Register Sequence In QPI Mode



8.19 WRITE FUNCTION REGISTER OPERATION (WRFR, 42h)

Information Row Lock bits (IRL3~IRL0) can be set to “1” individually by WRFR instruction in order to lock Information Row. Since IRL bits are OTP, once it is set to “1”, it cannot set back to “0” again.

Figure 8.32 Write Function Register Sequence In SPI Mode

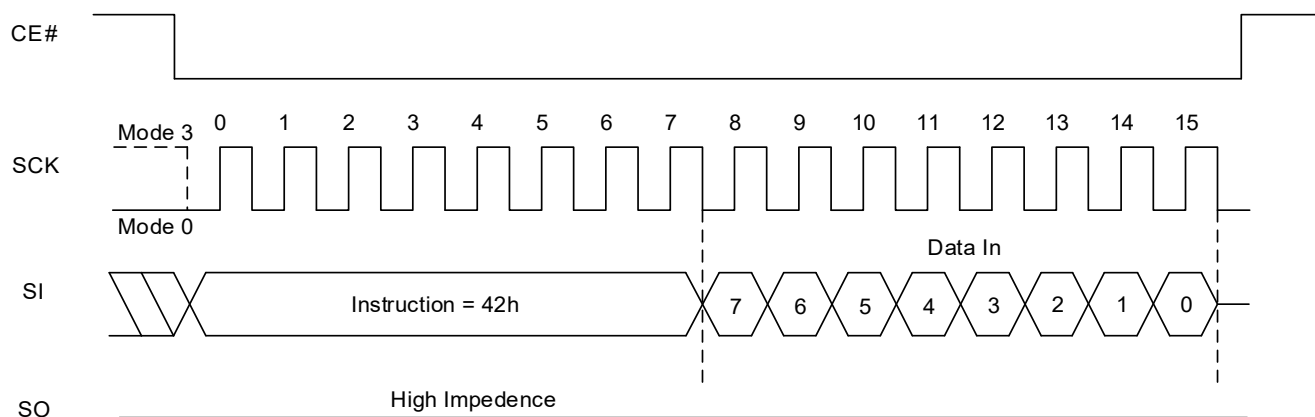
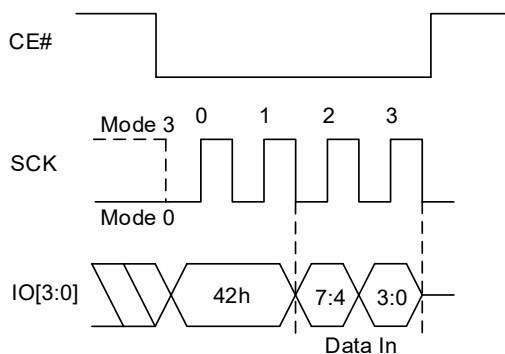


Figure 8.33 Write Function Register QPI Sequence In QPI Mode



8.20 ENTER QUAD PERIPHERAL INTERFACE (QPI) MODE OPERATION (QPIEN, 35h; QPIDI, F5h)

The Enter Quad Peripheral Interface (QPIEN) 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 Peripheral Interface instruction is sent to device.

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

Figure 8.34 Enter Quad Peripheral Interface (QPI) Mode Sequence

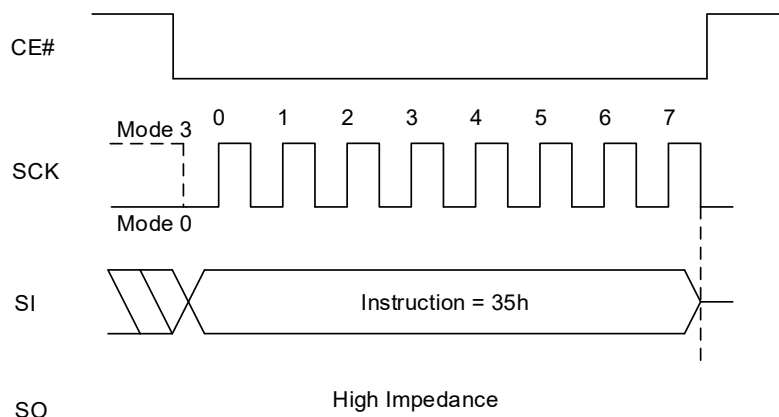
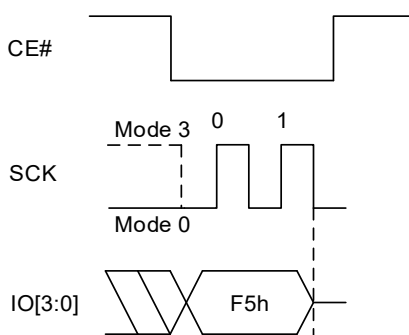


Figure 8.35 Exit Quad Peripheral Interface (QPI) Mode Sequence



8.21 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 (t_{SUS}): 100 μ s (TYP)

Resume to another suspend timing (t_{RS}): 80 μ s (TYP)

SUSPEND DURING SECTOR-ERASE OR BLOCK-ERASE (PERSUS 75h/B0h)

The Suspend command allows the interruption of Sector Erase and Block Erase operations. But Suspend command will be ignored during Chip Erase operation. After the Suspend command, other commands include array read operation can be accepted.

But Write Status Register command (01h) and Erase instructions are not allowed during Erase Suspend. Also, array read for being erased sector/block is not allowed.

To execute Erase Suspend operation, the host drives CE# low, sends the Suspend command cycle (75h/B0h), then drives CE# high. The Function Register indicates that the Erase has been suspended by setting 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 set to "1", the Write Enable Latch (WEL) bit clears to "0".

SUSPEND DURING PAGE PROGRAMMING (PERSUS 75h/B0h)

The Suspend command also allows the interruption of all array Program operations. After the Suspend command, other commands include array read operation can be accepted can be accepted.

But Write Status Register instruction (01h) and Program instructions are not allowed during Program Suspend. Also, array read for being programmed page is not allowed.

To execute the Program Suspend operation, the host drives CE# low, sends the Suspend command cycle (75h/B0h), then drives CE# high. The Function Register indicates that the programming has been suspended by setting 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} . When PSUS bit is set to "1", the Write Enable Latch (WEL) bit clears to "0".

PROGRAM/ERASE RESUME (PERRSM 7Ah/30h)

The Program/Erase Resume restarts the Program or Erase command that was suspended, and clears the suspend status bit in the Function Register (ESUS or PSUS bits) 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 issue another Erase Suspend operation after Erase Resume operation, Erase Resume to another Erase Suspend delay (t_{RS}) is required, but it could require longer Erase time to complete Erase operation.

To determine if the internal, self-timed Write operation completed, poll the WIP bit.

Table 8.4 Instructions accepted during Suspend

Operation Suspended	Instruction Allowed		
	Name	Hex Code	Operation
Program or Erase	NORD	03h	Read Data Bytes from Memory at Normal Read Mode
Program or Erase	FRD	0Bh	Read Data Bytes from Memory at Fast Read Mode
Program or Erase	FRDIO	BBh	Fast Read Dual I/O
Program or Erase	FRDO	3Bh	Fast Read Dual Output
Program or Erase	FRQIO	EBh	Fast Read Quad I/O
Program or Erase	FRQO	6Bh	Fast Read Quad Output
Program or Erase	FRDTR	0Dh	Fast Read DTR Mode
Program or Erase	FRDDTR	BDh	Fast Read Dual I/O DTR
Program or Erase	FRQDTR	EDh	Fast Read Quad I/O DTR
Erase	PP	02h	Serial Input Page Program
Erase	PPQ	32h/38h	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	Resume program/erase
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
Program or Erase	NOP	00h	No Operation
Program or Erase	RSTEN	66h	Software reset enable
Program or Erase	RST	99h	Reset (Only along with 66h)
Program or Erase	IRRD	68h	Read Information Row
Erase	SECUNLOCK	26h	Sector Unlock
Erase	SECLOCK	24h	Sector Lock

8.22 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 8.36 Enter Deep Power Down Mode Sequence In SPI Mode

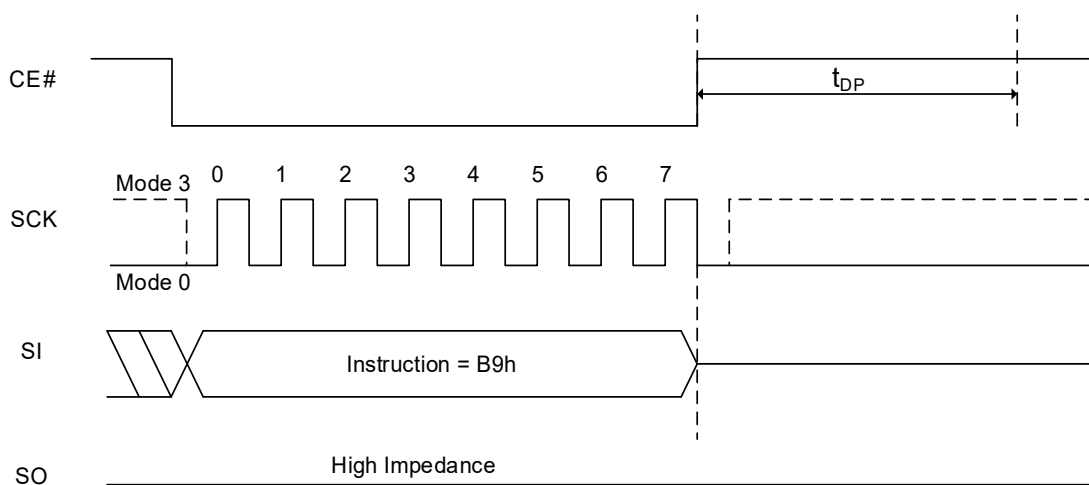
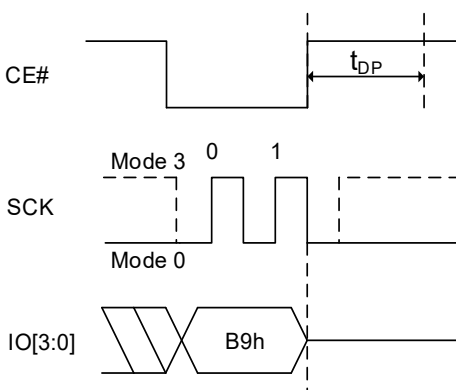


Figure 8.37 Enter Deep Power Down Mode Sequence In QPI Mode



8.23 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 Power-down mode, the instruction is issued by driving the CE# pin low, shifting the instruction code “ABh” 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 8.38 Release Deep Power Down Mode Sequence In SPI Mode

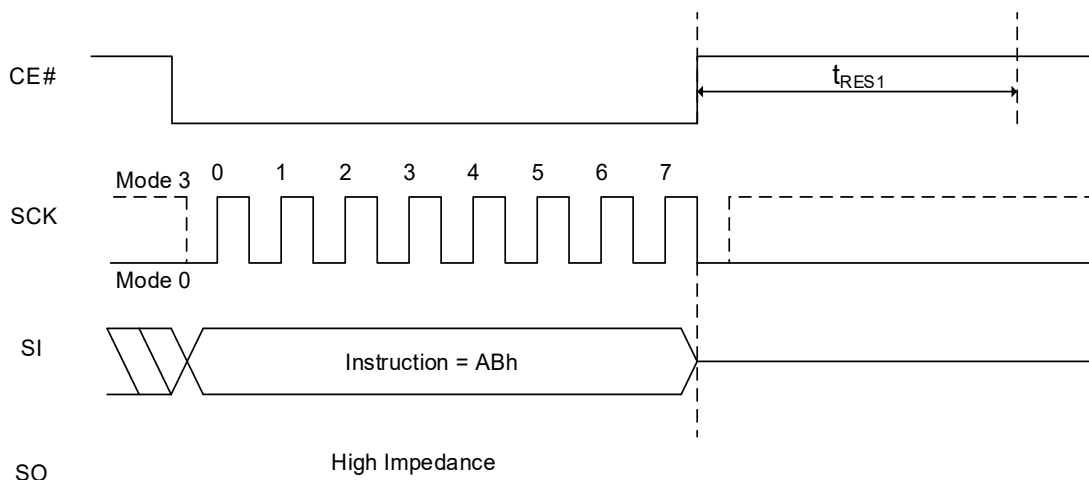
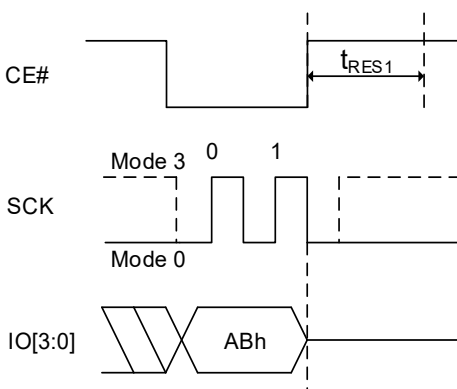


Figure 8.39 Release Deep Power Down Mode Sequence In QPI Mode



8.24 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# pin selection (P7) bit in the Read Register can be set with the SRPNV and SRPV operation as well, in order to select RESET# pin instead of HOLD# pin.

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 8.40 Set Read Parameters Sequence In SPI Mode

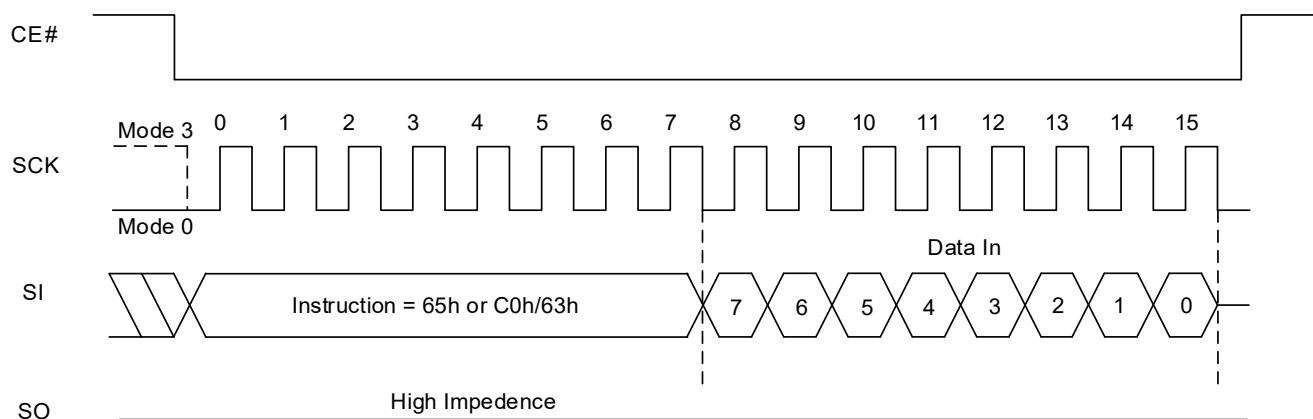
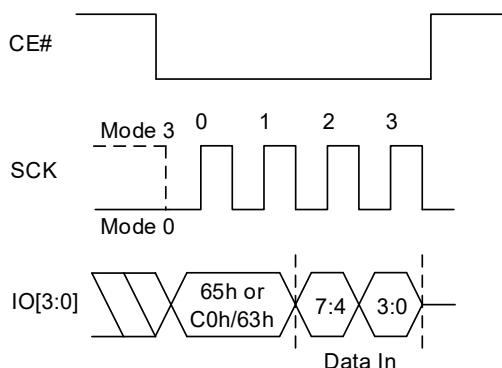


Figure 8.41 Set Read Parameters Sequence In QPI Mode



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 8.40 and Figure 8.41 for instruction sequence.

8.25 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 8.42 Set Extended Read Parameters Sequence In SPI Mode

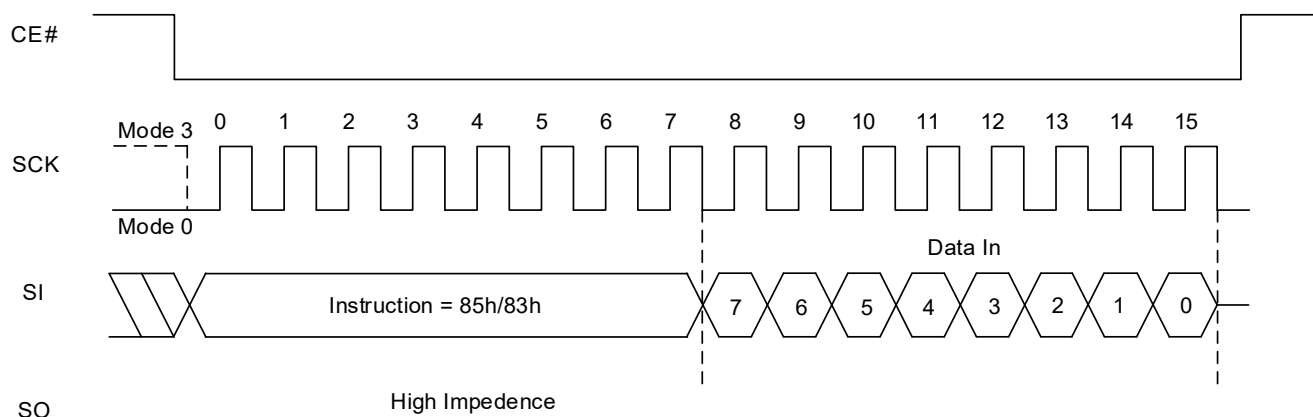
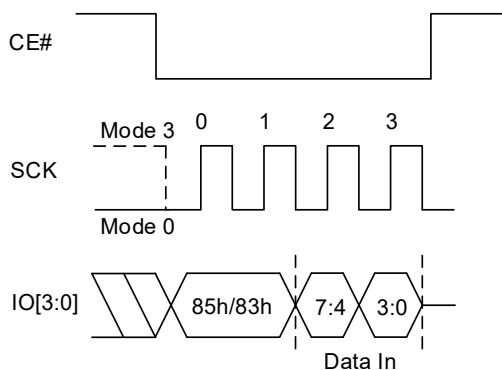


Figure 8.43 Set Extended Read Parameters Sequence In QPI Mode



8.26 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 8.44 Read Read Parameters Sequence In SPI Mode

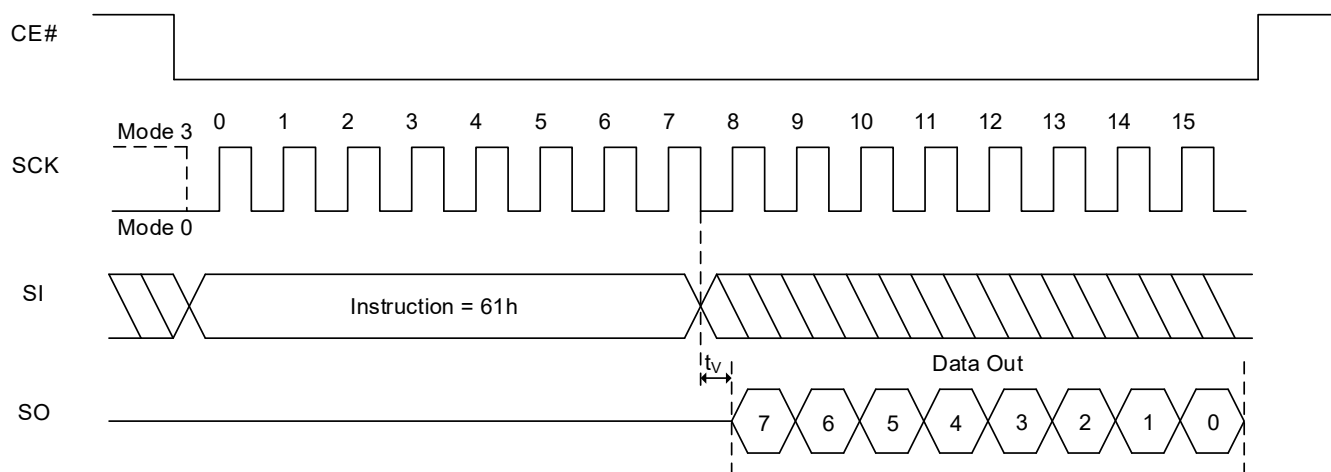
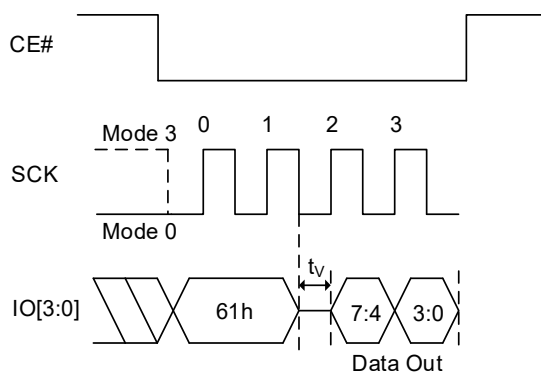


Figure 8.45 Read Read Parameters Sequence In QPI Mode



8.27 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 8.46 Read Extended Read Parameters Sequence In SPI Mode

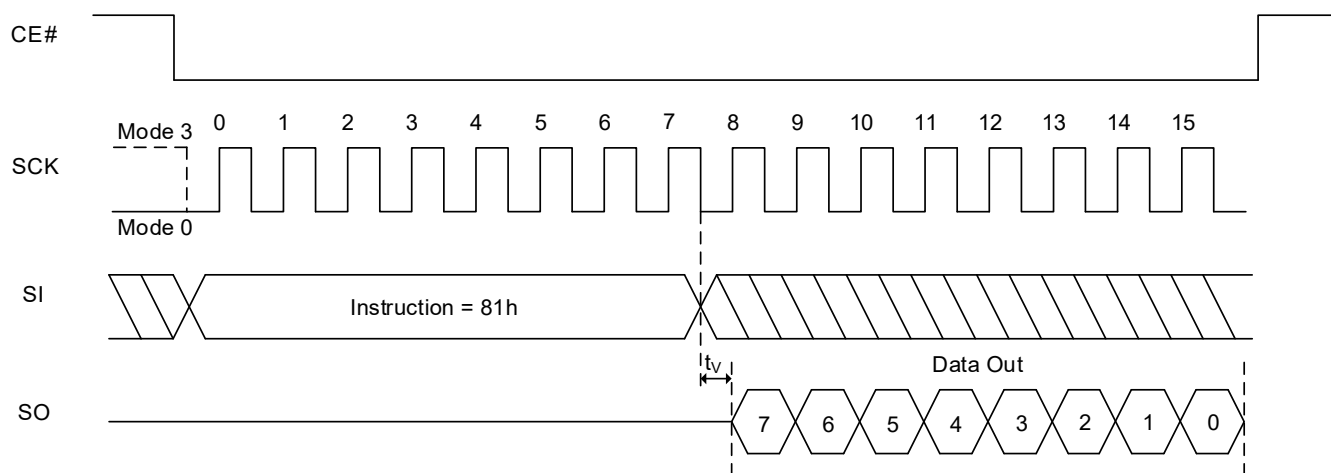
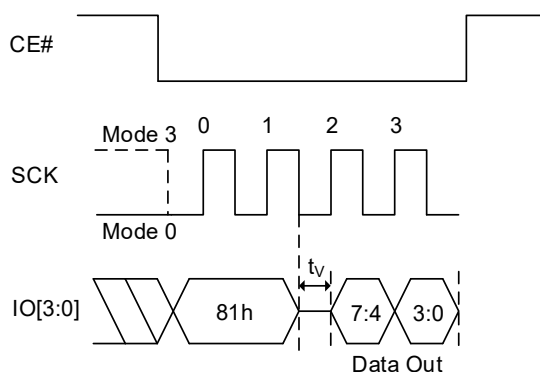


Figure 8.47 Read Extended Read Parameters Sequence In QPI Mode



8.28 CLEAR EXTENDED READ PARAMETERS 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 remain set to “1” until they are cleared to “0” with a CLERP command.

Figure 8.48 Clear Extended Read Parameters Sequence In SPI Mode

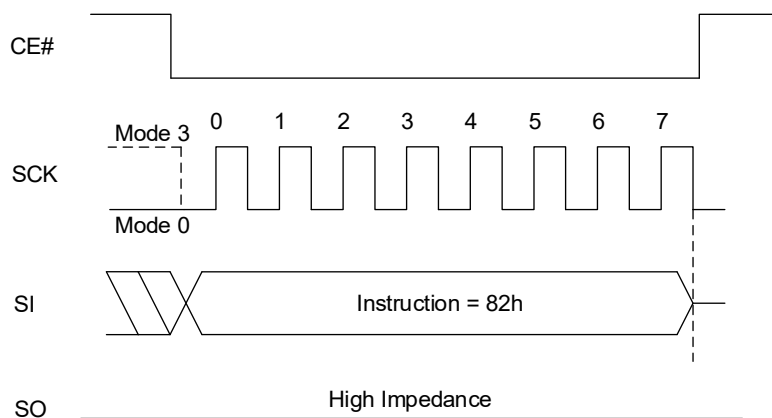
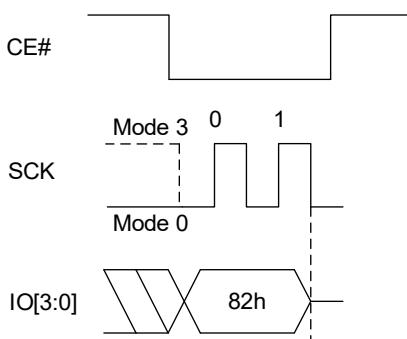


Figure 8.49 Clear Extended Read Parameters Sequence In QPI Mode



8.29 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 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 8.5 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)	
IS25LE080G	13h	6014h	
IS25LE040G	12h	6013h	

Figure 8.50 RDID (Read Product Identification) Sequence In SPI Mode

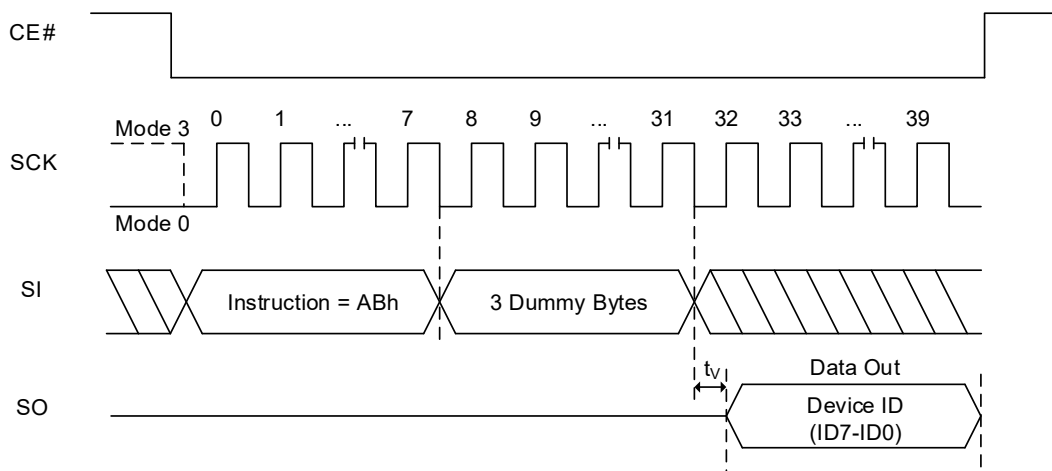
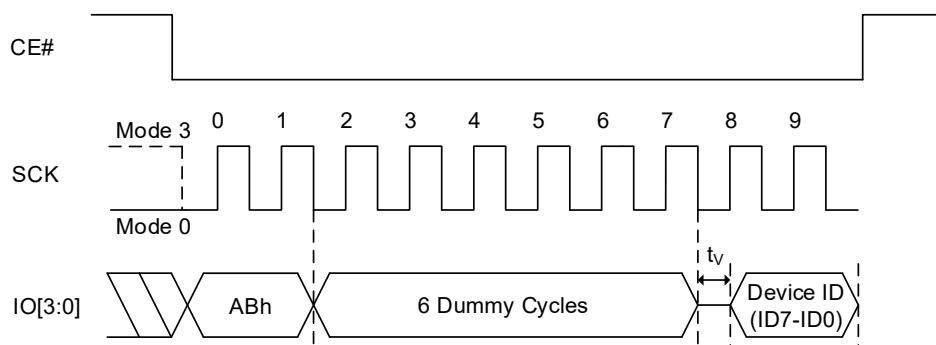


Figure 8.51 RDID (Read Product Identification) Sequence In QPI Mode


8.30 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 8.5 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 8.52 RDJDID (Read JEDEC ID In SPI Mode) Sequence

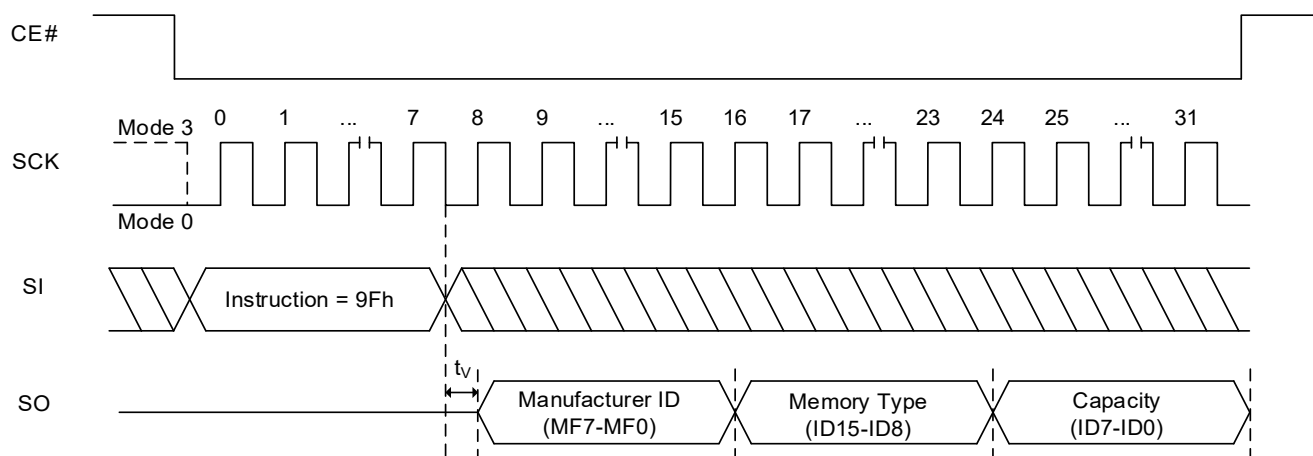
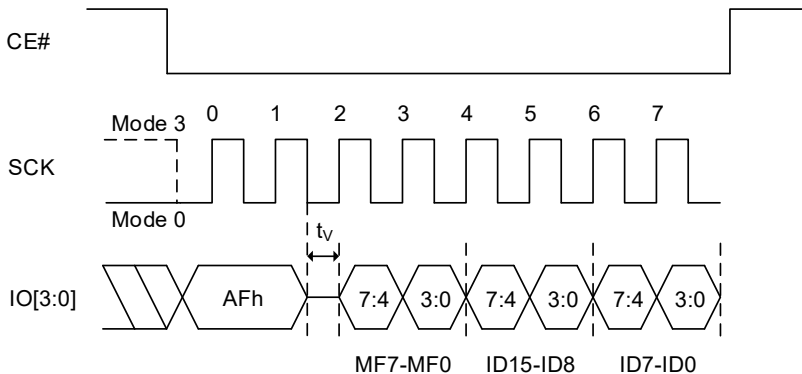


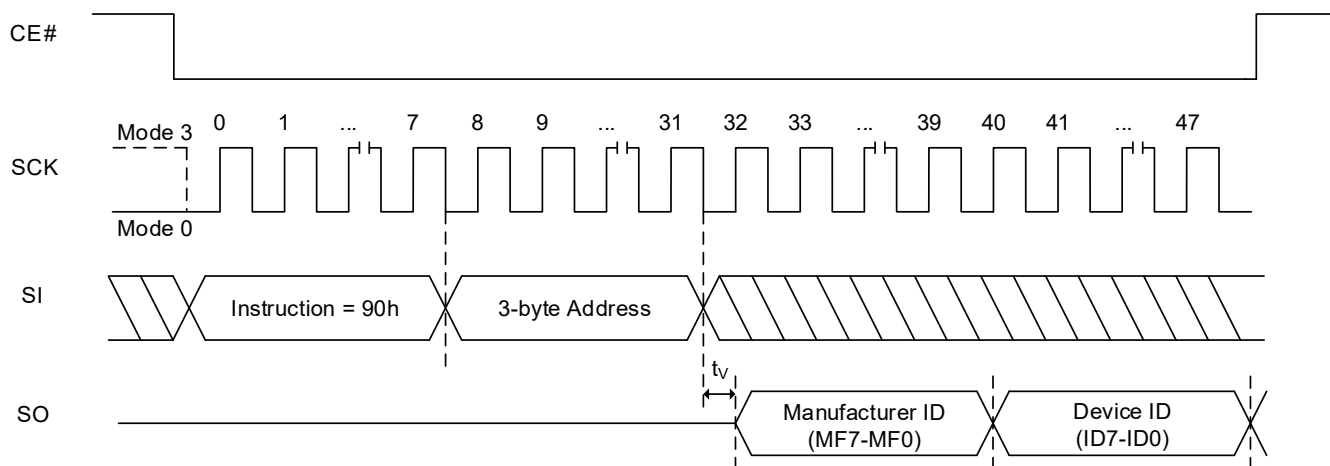
Figure 8.53 RDJDIDQ (Read JEDEC ID In QPI Mode) Sequence



8.31 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 8.5 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 ID 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.

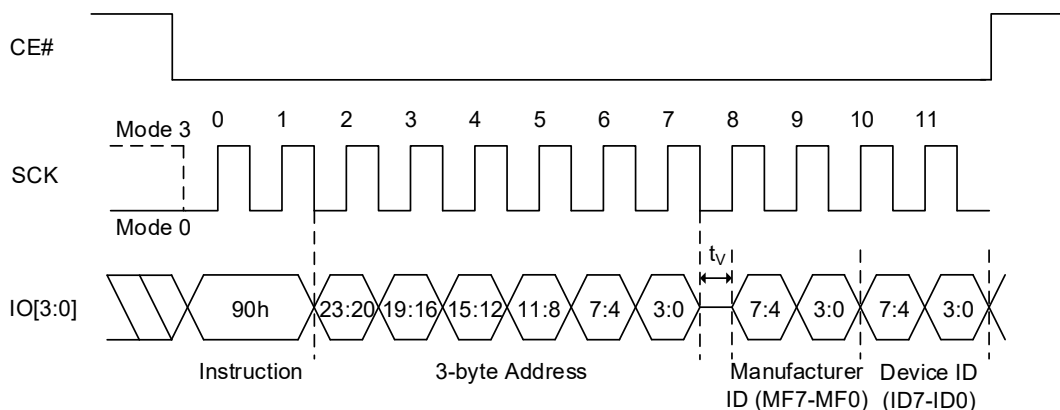
Figure 8.54 RDMDID (Read Product Identification) Sequence In SPI Mode



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 8.55 RDMDID (Read Product Identification) Sequence In QPI Mode



Notes:

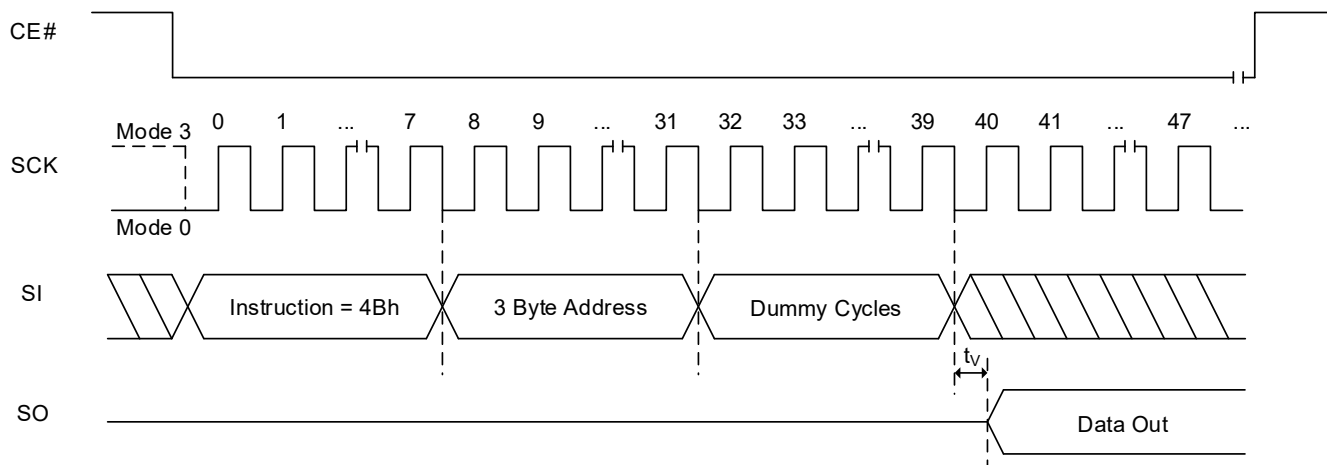
1. ADDRESS A0 = 0, will output the 1-byte Manufacturer ID (MF7-MF0) → 1-byte Device ID (ID7-ID0)
ADDRESS A0 = 1, will output the 1-byte Device ID (ID7-ID0) → 1-byte Manufacturer ID (MF7-MF0)
2. The Manufacturer and Device ID can be read continuously and will alternate from one to the other until CE#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

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

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

Figure 8.56 RDUID Sequence



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

Table 8.6 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”.

8.32 READ SFDP OPERATION (RDSFDP, 5Ah)

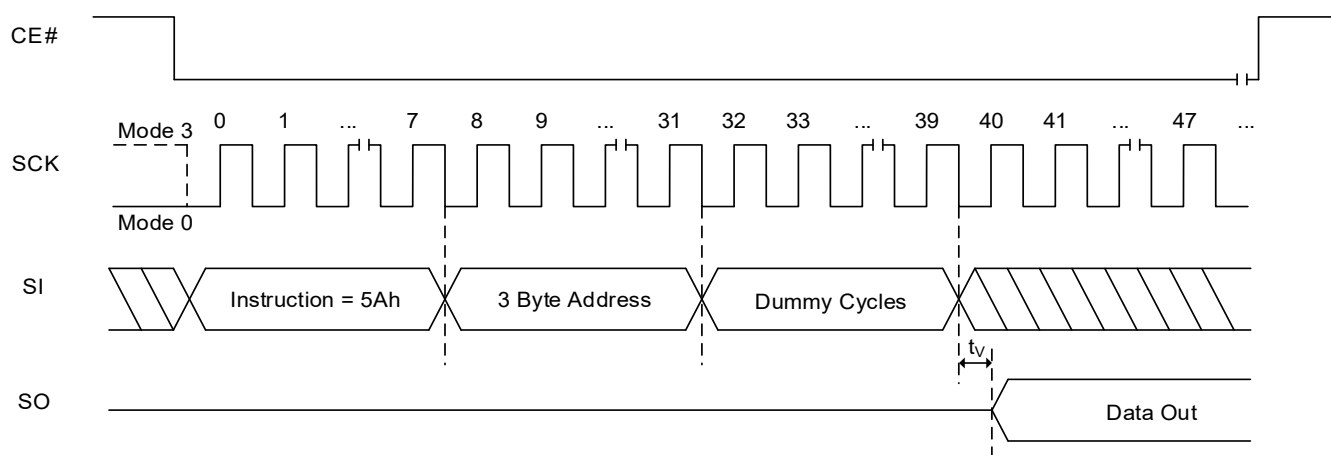
The Serial Flash Discoverable Parameters (SFDP) standard provides a consistent method of describing the functions and features of serial Flash devices in a standard set of internal parameter tables. These parameters can be interrogated by host system software to enable adjustments needed to accommodate divergent features from multiple vendors. For more details please refer to the JEDEC Standard JESD216 (Serial Flash Discoverable Parameters).

The sequence of issuing RDSFDP instruction in SPI mode is:

CE# goes low → Send RDSFDP instruction (5Ah) → Send 3 address bytes on SI pin → 8 dummy cycles 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.

RDSFDP Sequence in QPI mode, has 8 dummy cycles before SFDP code, too.

Figure 8.57 RDSFDP (Read SFDP) Sequence in SPI mode



8.33 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.

8.34 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. 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 when the RESET# pin is enabled, Hardware Reset function is available.

For the device with HOLD#/RESET#, the RESET# pin will be solely applicable when the QE bit = "0" and HOLD#/RESET# selection bit (P7) in Read Register is set to "1".

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 mode¹, 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 8.58 Software Reset Enable and Software Reset Sequence In SPI Mode (RSTEN, 66h + RST, 99h)

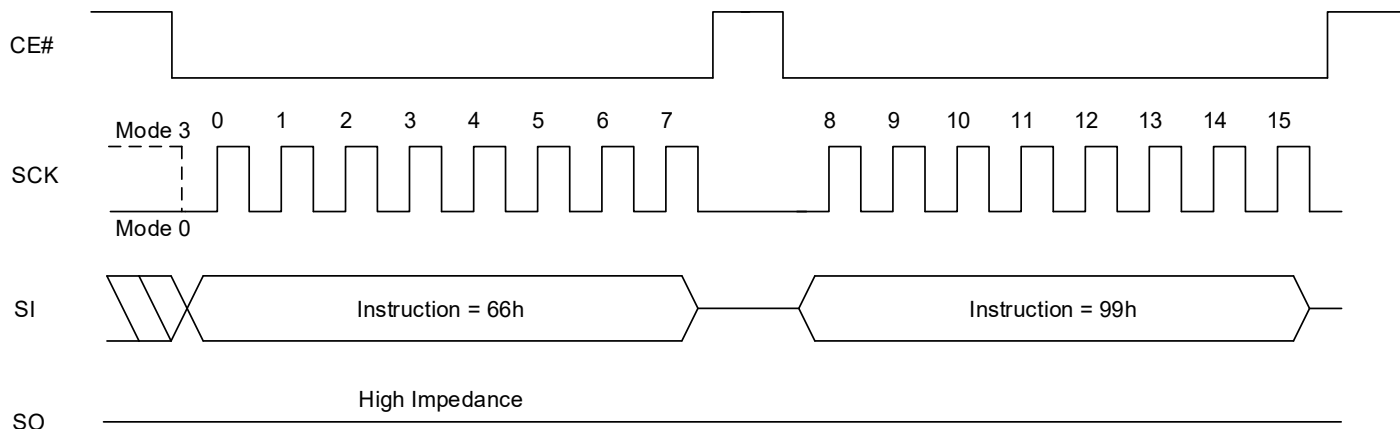
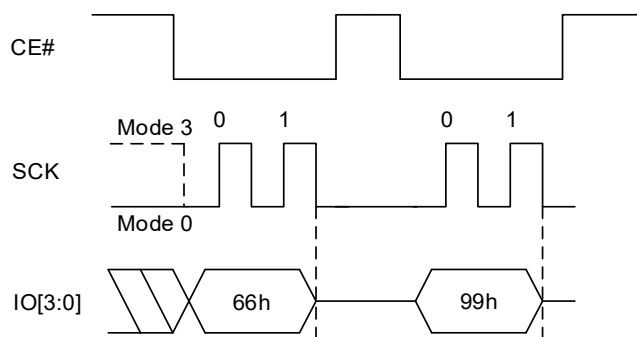


Figure 8.59 Software Reset Enable and Software Reset Sequence In QPI Mode (RSTEN, 66h + RST, 99h)


8.35 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 8.7 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.

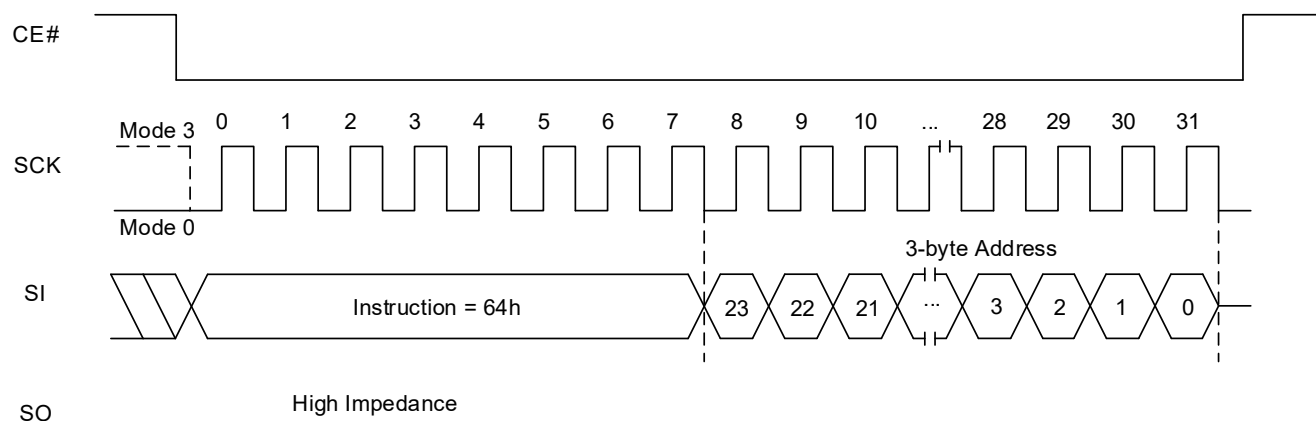
When Function Register bit IRLx = "0", the 256 bytes of the programmable memory array can be programmed.
When Function Register bit IRLx = "1", the 256 bytes of the programmable memory array function as read only.

8.36 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 8.60 IRER (Information Row Erase) Sequence



8.37 INFORMATION ROW PROGRAM OPERATION (IRP, 62h)

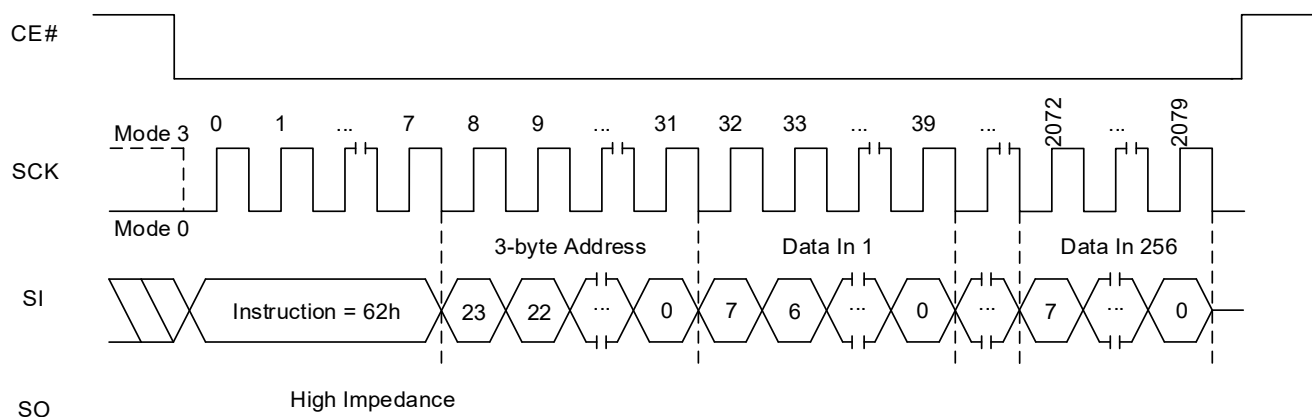
The Information Row Program (IRP) instruction allows up to 256 bytes data to be programmed into the memory in a single operation. Before the execution of IRP instruction, the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

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 8.7 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. The progress or completion of the program operation can be determined by reading the WIP bit in Status Register via a RDSR instruction. 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, The same byte location or Information Row array may be programmed more than once to incrementally change "1" to "0"s. An erase operation is required to change "0"s back to "1"s.

Figure 8.61 IRP (Information Row Program) Sequence



8.38 INFORMATION ROW READ OPERATION (IRRD, 68h)

The IRRD instruction is used to read memory data.

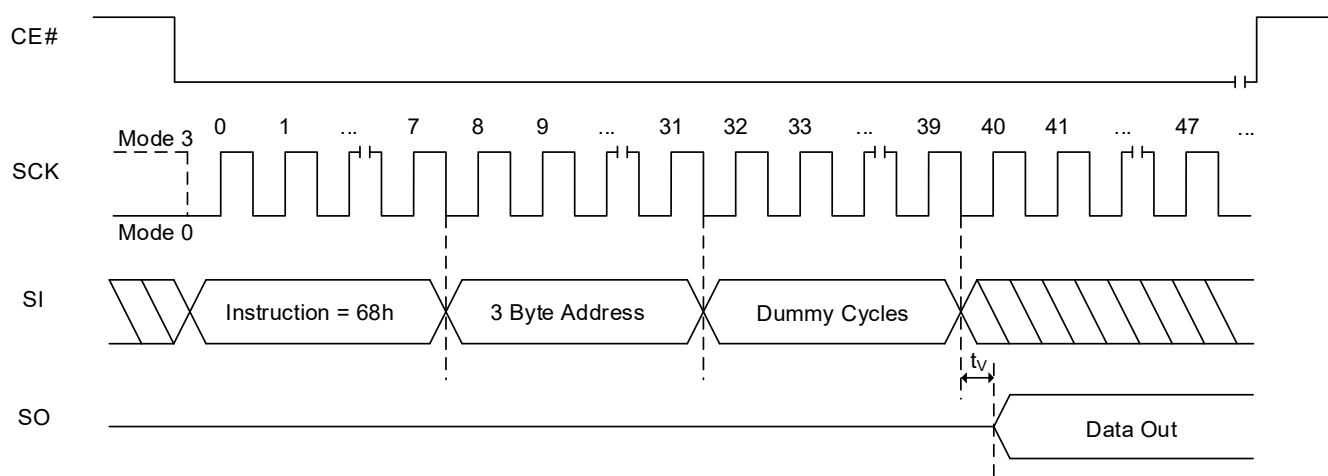
The IRRD instruction code is followed by three address bytes (A23 - A0) and dummy cycles (configurable, default is 8 clocks), transmitted via the SI line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the SO line, with each bit shifted out 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 also same as FAST READ QPI except for the instruction code. Refer to the FAST READ QPI operation.

Figure 8.62 IRRD (Information Row Read) Sequence



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

8.39 FAST READ DTR MODE OPERATION IN SPI MODE (FRDTR, 0Dh)

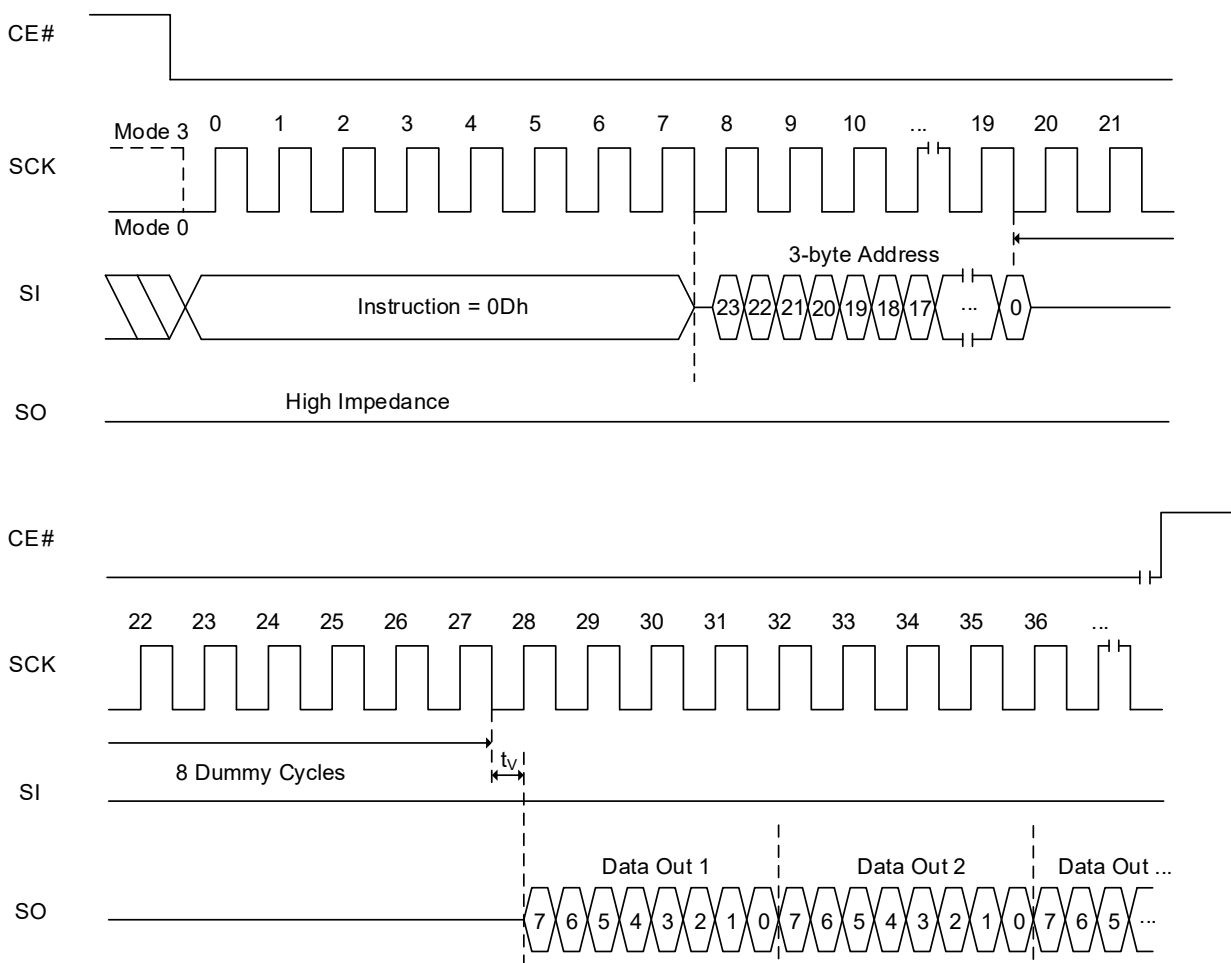
The FRDTR instruction is for doubling the data in and out. Signals are triggered on both rising and falling edge of clock. The address is latched on both rising and falling edge of SCK, and data of each bit shifts out on both rising and falling edge of SCK at a maximum frequency. The 2-bit address can 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 instruction. The address counter rolls over to 0 when the highest address is reached.

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

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

Figure 8.63 FRDTR Sequence In SPI Mode



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

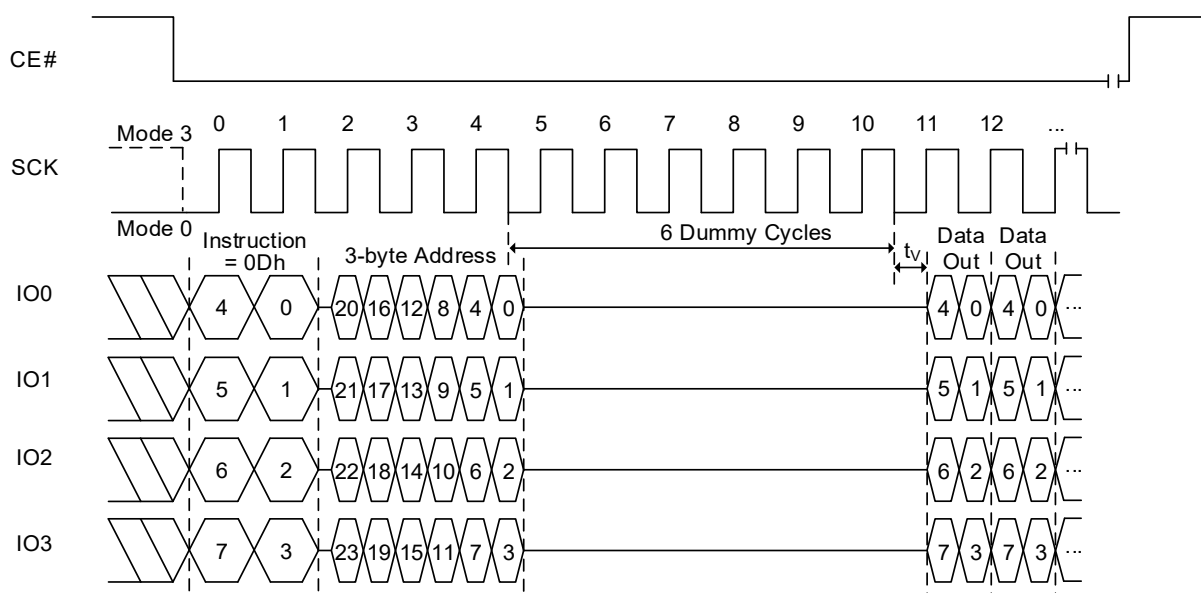
FAST READ DTR MODE OPERATION IN QPI MODE (FRDTR, 0Dh)

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

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

If the FRDTR QPI instruction is issued while a Program/Erase/Write Status Register cycle is in progress (WIP=1), the instruction will be rejected without any effect on the current cycle.

Figure 8.64 FRDTR Sequence In QPI Mode



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.

8.40 FAST READ DUAL IO DTR MODE OPERATION (FRDDTR, BDh)

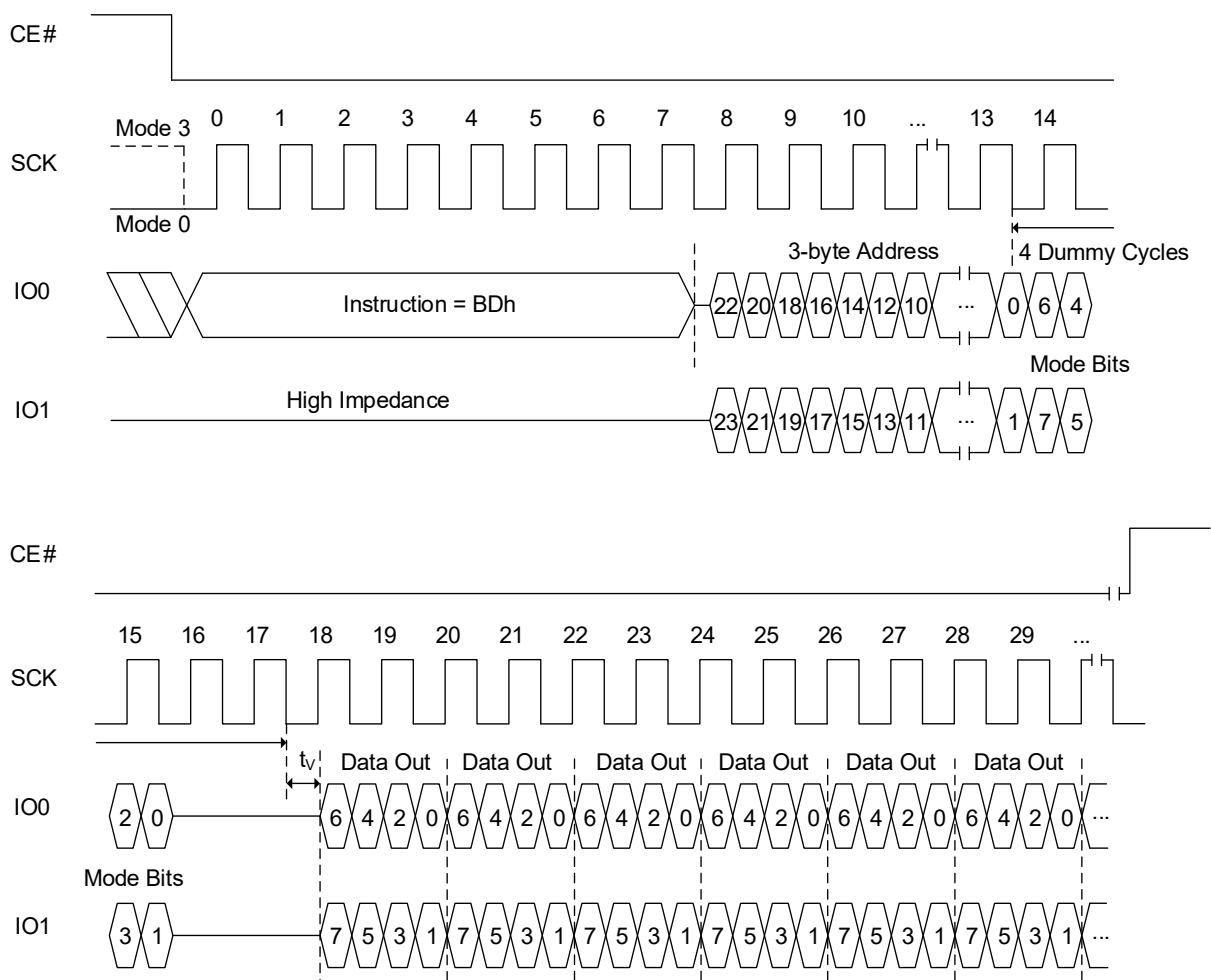
The FRDDTR instruction enables Double Transfer Rate throughput on dual I/O of the device in read mode. The address (interleave on dual I/O pins) is latched on both rising and falling edge of SCK, and the data (interleave on dual I/O pins) shift out on both rising and falling edge of SCK at a maximum frequency. 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 instruction. The address counter rolls over to 0 when the highest address is reached. Once writing FRDDTR instruction, the following address/dummy/data out will perform as 4-bit instead of previous 1-bit.

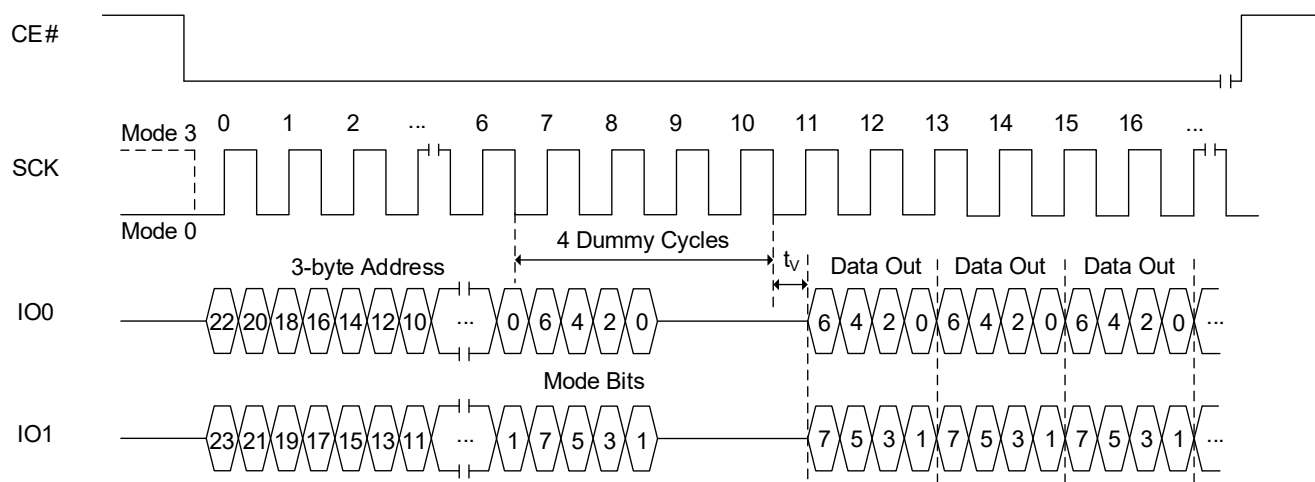
The sequence of issuing FRDDTR instruction is: CE# goes low → Sending FRDDTR instruction (1-bit per clock) → 24-bit address interleave on IO1 & IO0 (4-bit per clock) → 4 dummy clocks (configurable, default is 4 clocks) on IO1 & IO0 → Data out interleave on IO1 & IO0 (4-bit per clock) → End FRDDTR operation via pulling CE# high at any time during data out (Please refer to Figure 8.64 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 execution skips command code. It saves cycles as described in Figure 8.65. When the code is different from AXh, the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command.

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

Figure 8.65 FRDDTR Sequence (with command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

Figure 8.66 FRDDTR AX Read Sequence (without command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

8.41 FAST READ QUAD IO DTR MODE OPERATION IN SPI MODE (FRQDTR, EDh)

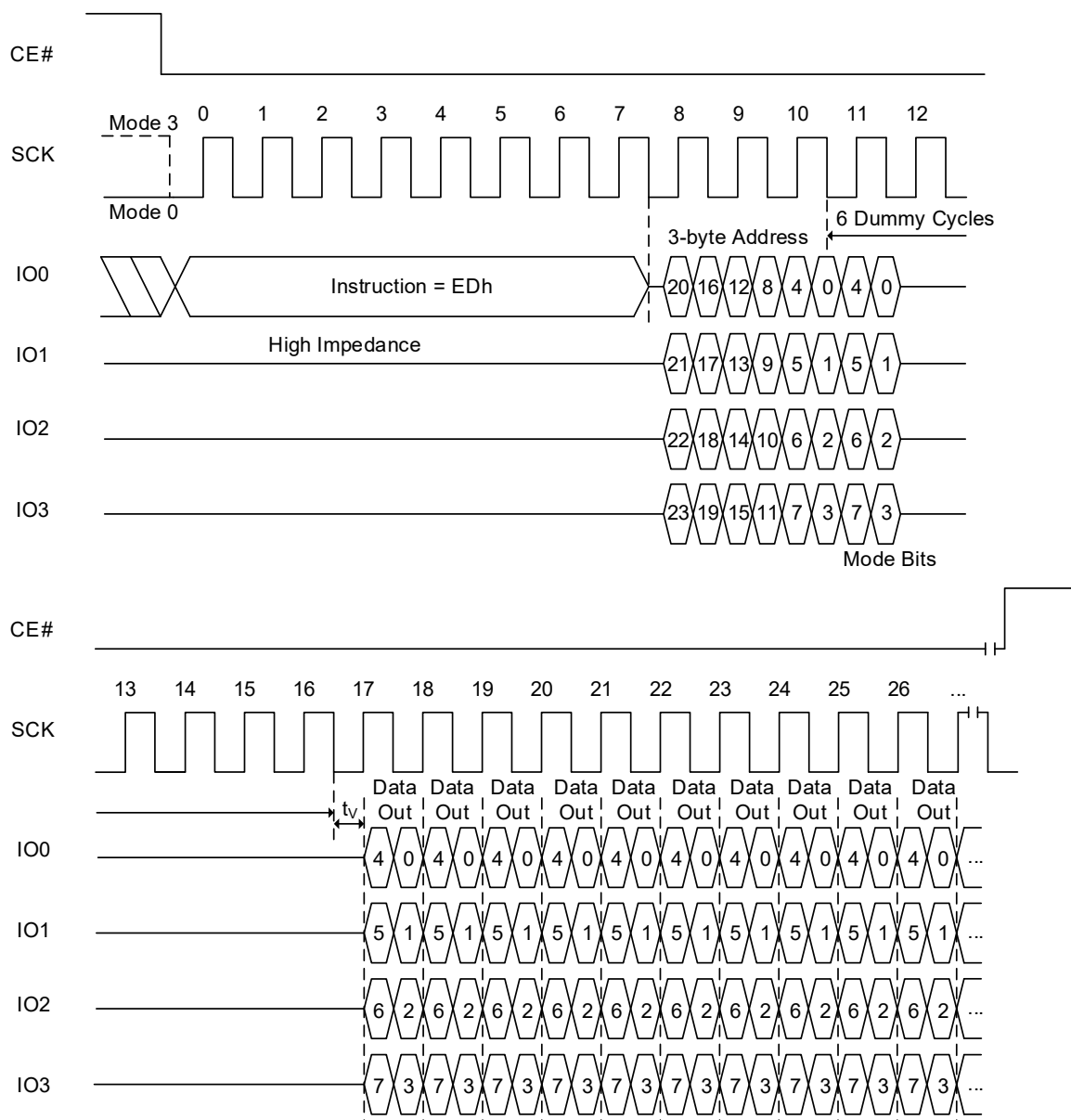
The FRQDTR instruction enables Double Transfer Rate throughput on quad I/O of the device in read mode. A Quad Enable (QE) bit of Status Register must be set to "1" before sending the FRQDTR instruction. The address (interleave on 4 I/O pins) is latched on both rising and falling edge of SCK, and data (interleave on 4 I/O pins) shift out on both rising and falling edge of SCK at a maximum frequency. 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 instruction. The address counter rolls over to 0 when the highest address is reached. Once writing FRQDTR instruction, the following address/dummy/data out will perform as 8-bit instead of previous 1-bit.

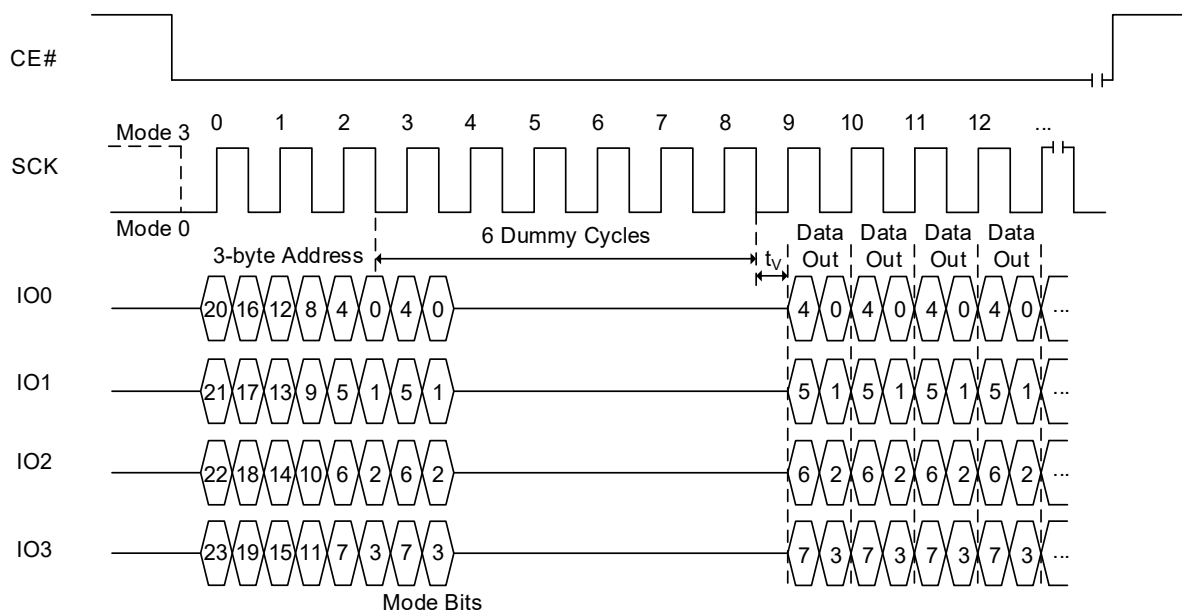
The sequence of issuing FRQDTR instruction is: CE# goes low → Sending FRQDTR instruction (1-bit per clock) → 24-bit address interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → 6 dummy clocks (configurable, default is 6 clocks) → Data out interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → End FRQDTR 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 execution skips command code. It saves cycles as described in Figure 8.68. When the code is different from AXh, the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command.

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

Figure 8.67 FRQDTR Sequence (with command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

Figure 8.68 FRQDTR Sequence (without command decode cycles)

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 bit cycles are same, then X should be Hi-Z.

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

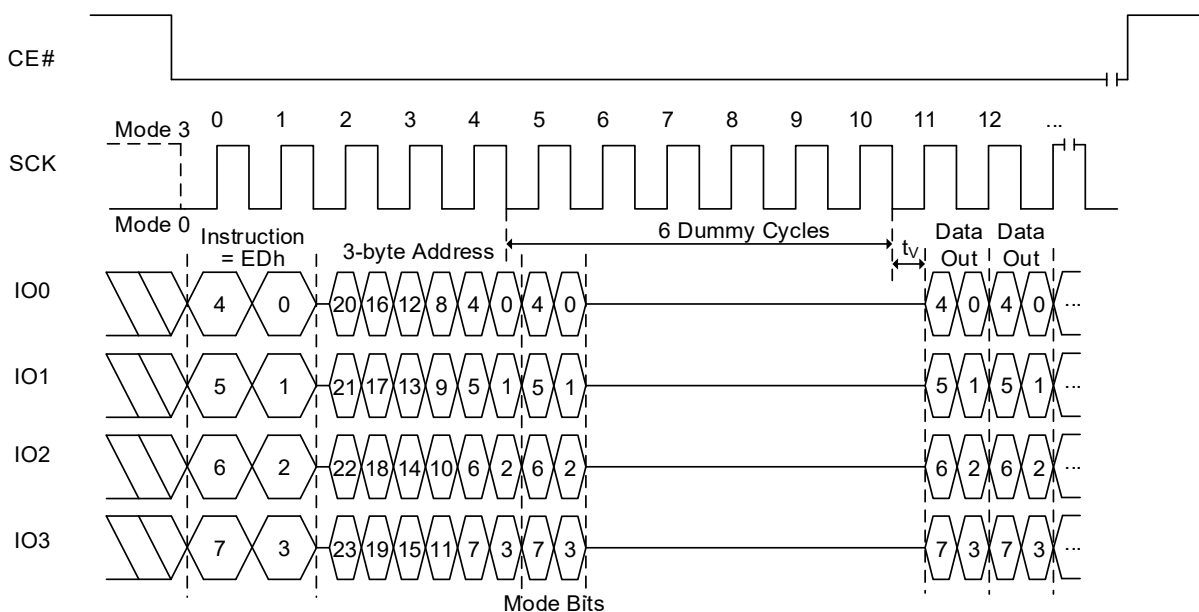
The FRQDTR QPI instruction utilizes all four IO lines to input the instruction code so that only two clocks are required, while the FRQDTR instruction requires that the byte-long instruction code is shifted into the device only via IO0 line in eight clocks. As a result, 6 command cycles will be reduced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

The sequence of issuing FRQDTR QPI instruction is: CE# goes low → Sending FRQDTR QPI instruction (4-bit per clock) → 24-bit address interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → 6 dummy clocks (configurable, default is 6 clocks) → Data out interleave on IO3, IO2, IO1 & IO0 (8-bit per clock) → End FRQDTR QPI 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 QPI execution skips command code. It saves cycles as described in Figure 8.68. When the code is different from AXh, the device exits the AX read operation. After finishing the read operation, device becomes ready to receive a new command.

If the FRQDTR QPI instruction is issued while a Program/Erase/Write Status Register cycle is in progress (WIP=1), the instruction will be rejected without any effect on the current cycle.

Figure 8.69 FRQDTR QPI Sequence (with command decode cycles)



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 bit cycles are same, then X should be Hi-Z.

8.42 SECTOR LOCK/UNLOCK FUNCTIONS

SECTOR UNLOCK OPERATION (SECUNLOCK, 26h)

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, BP1, BP2, and BP3 bits in the Status 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. The instruction code is followed by a 24-bit address specifying the target sector, but A0 through A11 are not decoded. The remaining sectors within the same block remain as read-only.

Figure 8.70 Sector Unlock Sequence In SPI Mode

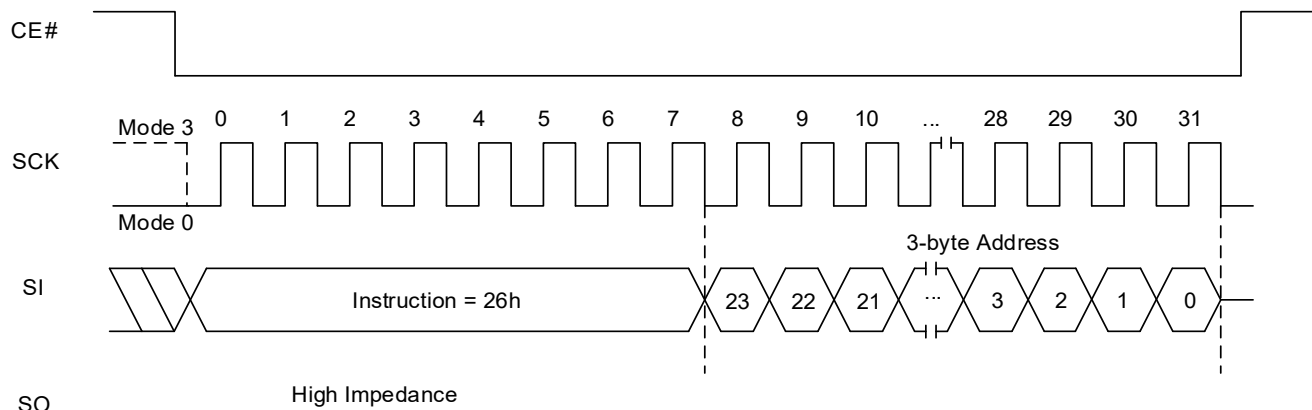
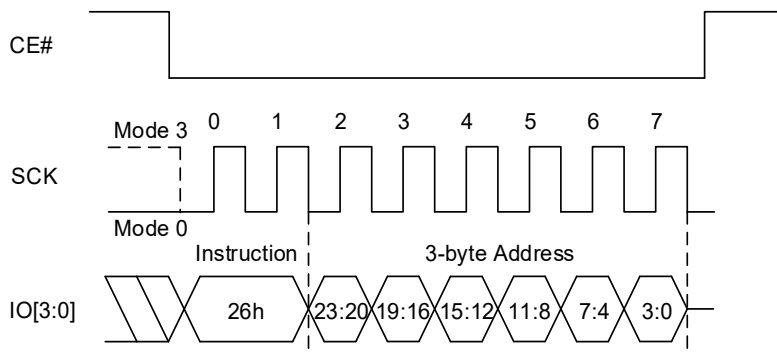
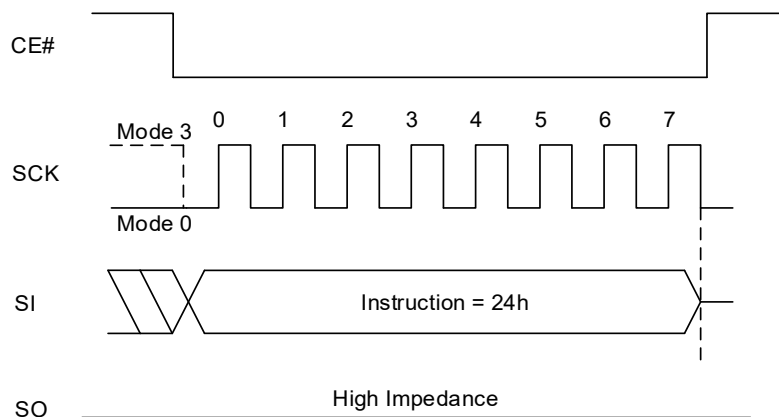
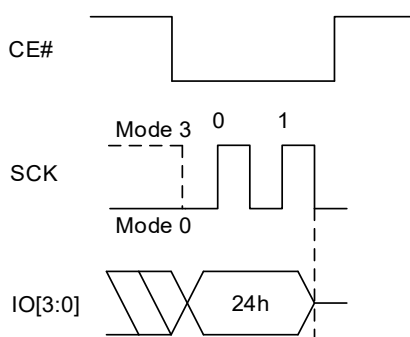


Figure 8.71 Sector Unlock QPI Sequence In QPI Mode



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.

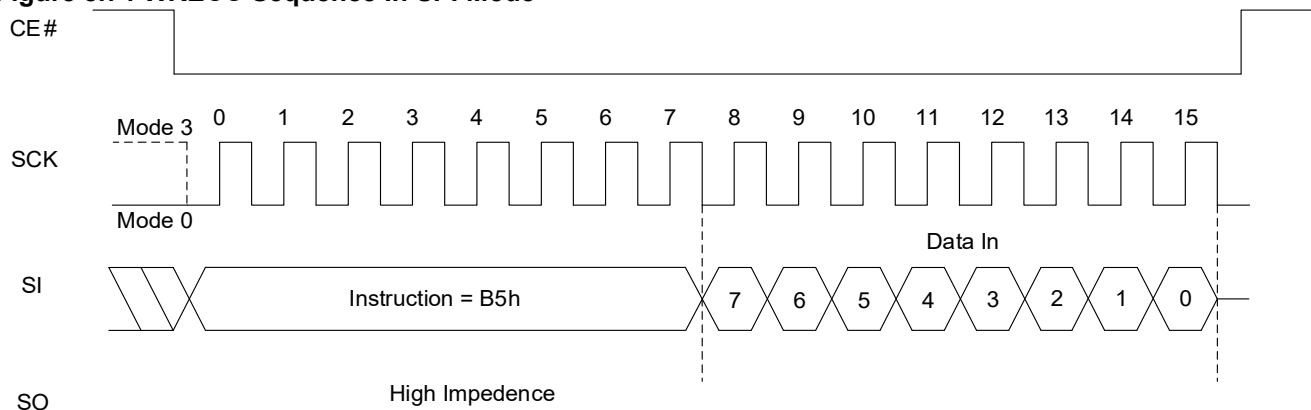
Figure 8.72 Sector Lock Sequence In SPI Mode

Figure 8.73 Sector Lock QPI Sequence In QPI Mode


8.43 WRITE REGISTER OPERATION (WRECC, B5h)

The WRECC command (B5h) is entered by shifting the instruction and data byte on SI.

CE# must be driven to the logic high state after the eighth (8th) bit of data has been latched. If not, the WRECC command is not executed. As soon as CE# is driven to the logic high state, the WRECC operation is initiated with no delays.

Figure 8.74 WRECC Sequence In SPI Mode



8.44 READ ECC REGISTER OPERATION (RDECC, B3h)

The RDECC 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 ECC Register, the next address will not be valid and the data of the address will be garbage data.

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

Figure 8.75 Read ECC Register Sequence In SPI Mode

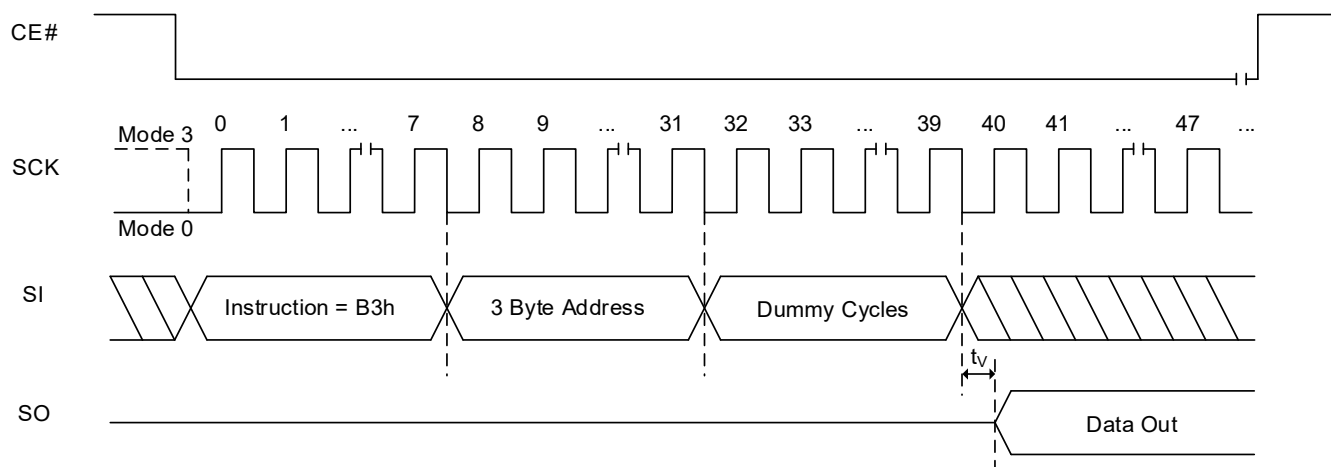
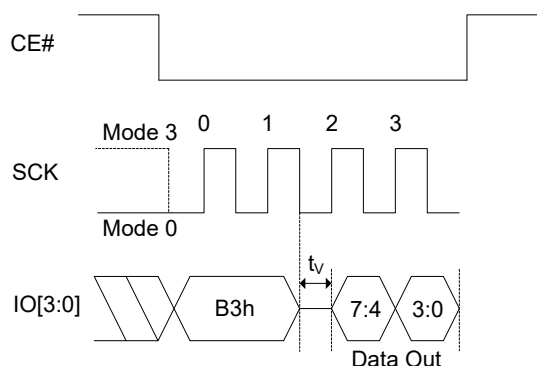


Figure 8.76 Read ECC Register In QPI Mode



8.45 CLEAR ECC REGISTER OPERATION (CLECC, B6h)

A Clear ECC Register (CLECC) instruction clears ECCEVENT, ECCOUNTER, and IPA_ECCB, ECCSTAT, and ECCFCA[23:0] bits in the ECC Register to “0”. Once any bits are set to “1”, they remain set to “1” until they are cleared to “0” with a CLECC command.

Figure 8.77 Clear ECC Register Sequence In SPI Mode

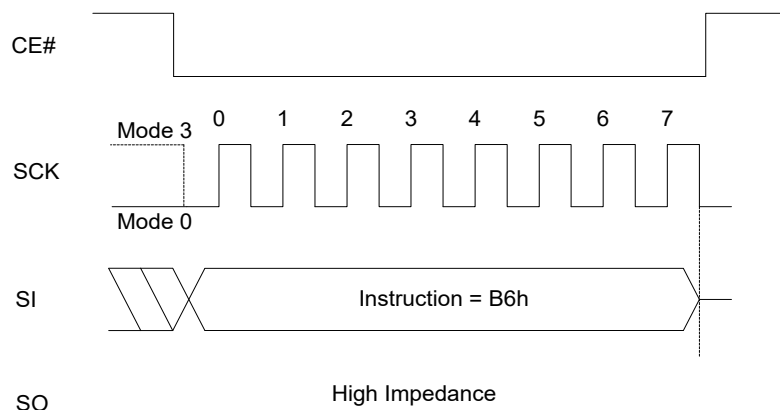
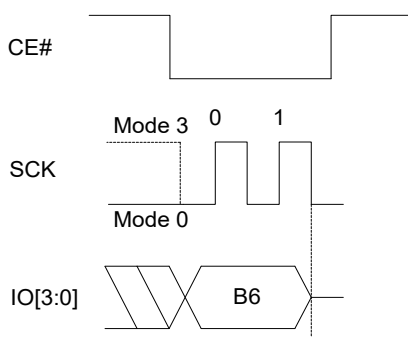


Figure 8.78 Clear ECC Register Sequence In QPI Mode



8.46 ERR# OPERATION

The ERR# pin is a real time indication signal for the ECC event.

The ERR# pin is designed as an open drain structure. (Internally pulled-up)

In normal situation, the ERR# is in **HIGH impedance state**, and **ERR# signal is pulled HIGH with internal pull-up**. Once ECC event occurs, the ERR# pin will be pulled LOW and stays LOW until RESET the device or until CLRERR command (B6h) is issued.

When ERR# signal goes LOW after detecting ECC error, especially when 2-bit detection error occurred, ERR# signal must be LOW before the end of ECC chunk read data for host to block wrong read data from the device.

So valid ERR# LOW signal must be within ECC chunk read data (tERR).

tERR is from beginning of ECC chunk read data to ERR# LOW, and is maximum 2 clock cycles.

Table 6.18

Symbol	Parameter	Min	Typ	Max	Units
tERR	ERR# Access time from beginning of ECC chunk	-		2	Clock

The user can select ERR# corresponding ECC event type; 1-bit correction or 2-bit detection by setting ERRBECC bit (bit 7 of address 000000h in ECC register).

ERRBECC bit also selects ECC event type for storage of ECC occurred first chunk address location.

If it is set to "0",

- ERR# signal reacts only to the 2-bit detection only.
- The address for first 2-bit detection occurred location will be stored on the ECC register (address 000002h~000004h).

To confirm error type (2-bit detection) after ERR# signal goes LOW, host could check ECCSTAT bits (bit [0] of address 000001h in ECC register).

The ERR# signal goes to high-z state by CLRERR command (B6h), which also clears all volatile register values ECC error.

9. ELECTRICAL CHARACTERISTICS

9.1 ABSOLUTE MAXIMUM RATINGS ⁽¹⁾

Storage Temperature	-65°C to +150°C
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}	-0.5V to +6.0V
Electrostatic Discharge Voltage (Human Body Model) ⁽²⁾	-2000V to +2000V

Notes:

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.
2. ANSI/ESDA/JEDEC JS-001

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	2.7V (V_{MIN}) – 3.6V (V_{MAX}); 3.0V (Typ)	

9.3 DC CHARACTERISTICS

(Under operating range)

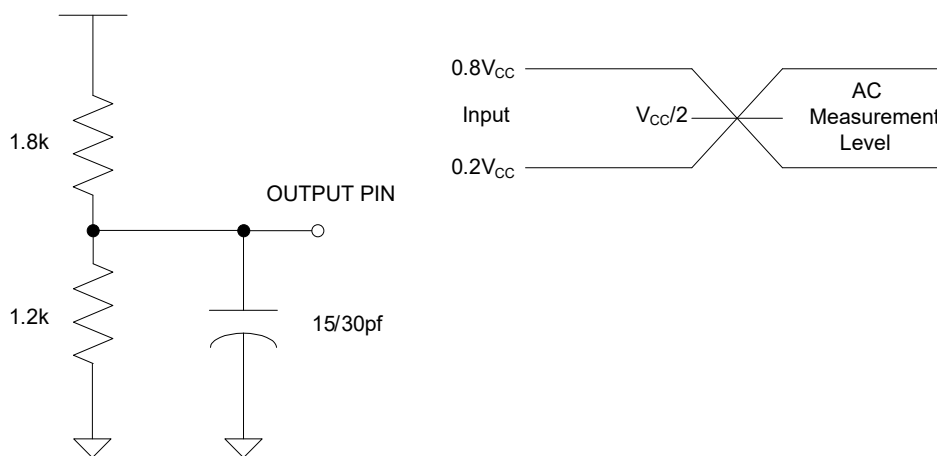
Symbol	Parameter	Condition		Min	Typ ⁽²⁾	Max	Units
I _{CC1}	V _{CC} Active Read current ⁽³⁾	NORD at 66MHz			6	12	mA
		FRD Single at 166MHz			8	14	
		FRD Dual at 166MHz			10	15	
		FRD Quad at 166MHz			13	17	
		FRD Quad at 133MHz			11	16	
		FRD Quad at 104MHz			10	15	
		FRD Single DTR at 80MHz			8	14	
		FRD Dual DTR at 80MHz			10	15	
		FRD Quad DTR at 80MHz			13	17	
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/BER32/BER64)	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	CE# = V _{CC} , V _{IN} = GND or V _{CC}	85°C		10	18 ⁽⁴⁾	μA
			105°C			24 ⁽⁴⁾	
			125°C			36	
I _{SB2}	Deep power down current	CE# = V _{CC} , V _{IN} = GND or V _{CC}	85°C		8	15 ⁽⁴⁾	μA
			105°C			20 ⁽⁴⁾	
			125°C			25	
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 ⁽⁶⁾	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 $V_{CC} + 0.5V$. During voltage transitions, input or I/O pins may overshoot V_{CC} 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 $V_{CC} = V_{CC} (Typ)$, $T_A = 25^{\circ}C$.
3. Outputs are unconnected during reading data so that output switching current is not included.
4. These parameters are characterized and are not 100% tested.
5. 2 μA for RESET# pin.
6. 2 μA for ERR# pin.

9.4 AC MEASUREMENT CONDITIONS

Symbol	Parameter	Min	Max	Units
CL	Load Capacitance		30	pF
			10	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

($T_A = 25^{\circ}C$, $V_{CC} = 3V$, 1MHz)

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

Notes:

1. These parameters are characterized and are not 100% tested.

9.6 AC CHARACTERISTICS

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

Symbol	Parameter			Min	Typ ⁽³⁾	Max	Units
f _{CT}	Clock Frequency for fast read: SPI, Dual, Dual I/O, Quad I/O, and QPI.			0		166	MHz
	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 (03h)			0		66	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 (03h)		0.45 x 1/f _{Cmax}			ns
		For others		0.45 x 1/f _{CTmax}			
t _{CKL}	SCK Low Time	For read (03h)		0.45 x 1/f _{Cmax}			ns
		For others		0.45 x 1/f _{CTmax}			
t _{CEH}	CE# High Time			7			ns
t _{CS}	CE# Setup Time			5			ns
t _{CH}	CE# Hold Time			3			ns
t _{SHCH}	CE# Not Active Setup Time			3			ns
t _{CHSL}	CE# Not Active Hold Time			3			ns
t _{DS}	Data In Setup Time	STR		2			ns
		DTR		1.5			
t _{DH}	Data in Hold Time	STR		2			ns
		DTR		1.5			
t _V	Output Valid	-40°C to 85°C	@10pF			5.5 ⁽¹⁾	ns
			@30pF			7.0 ⁽¹⁾	
		-40°C to 125°C	@10pF			6.5	
			@30pF			8.0	
t _{OH}	Output Hold Time			2			ns
t _{DIS} ⁽¹⁾	Output Disable Time					8	ns
t _{WHSL} ⁽³⁾	Write Protect Setup Time			20			ns
t _{SHWL} ⁽³⁾	Write Protect Hold Time			100			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					12	ns
t _{HZ} ⁽¹⁾	HOLD to Output High Z					12	ns



IS25LE080G/040G

Symbol	Parameter	Min	Typ ⁽²⁾	Max	Units
t _{RES1} ⁽¹⁾	Release deep power down			3	μs
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 _{RS} ⁽¹⁾	Resume to next suspend		80	-	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. Only applicable as a constraint for a WRITE STATUS REGISTER command when SRWD is set at 1.

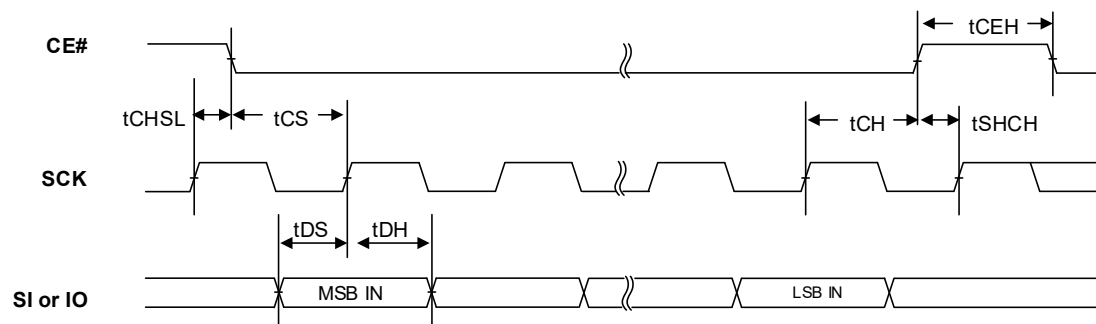
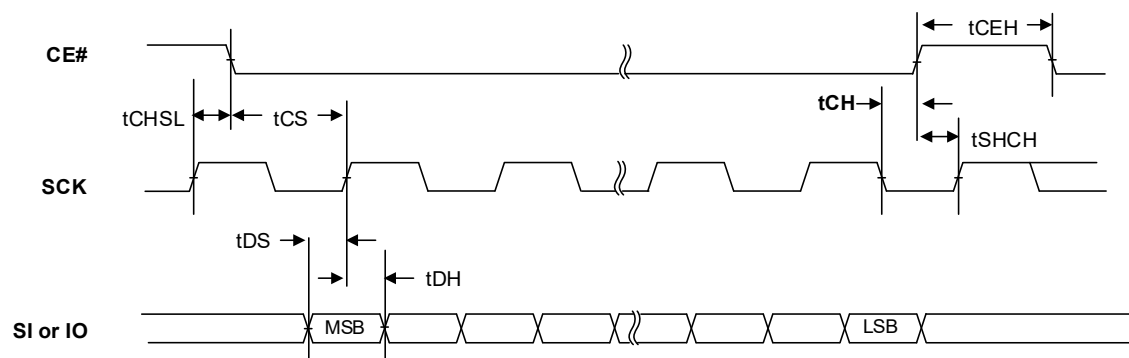
9.7 SERIAL INPUT/OUTPUT TIMING
Figure 9.2 SERIAL INPUT TIMING (STR Mode)

Figure 9.3 SERIAL INPUT TIMING (DTR Mode)


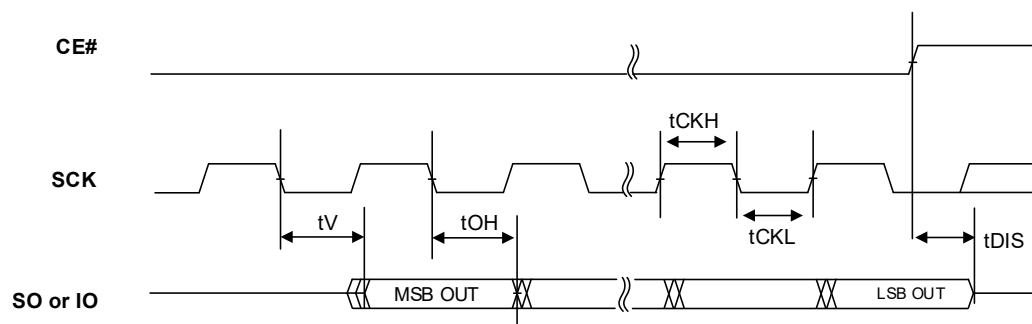
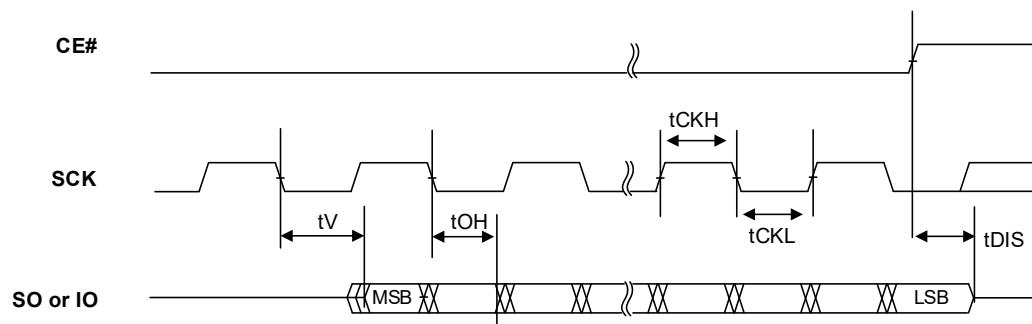
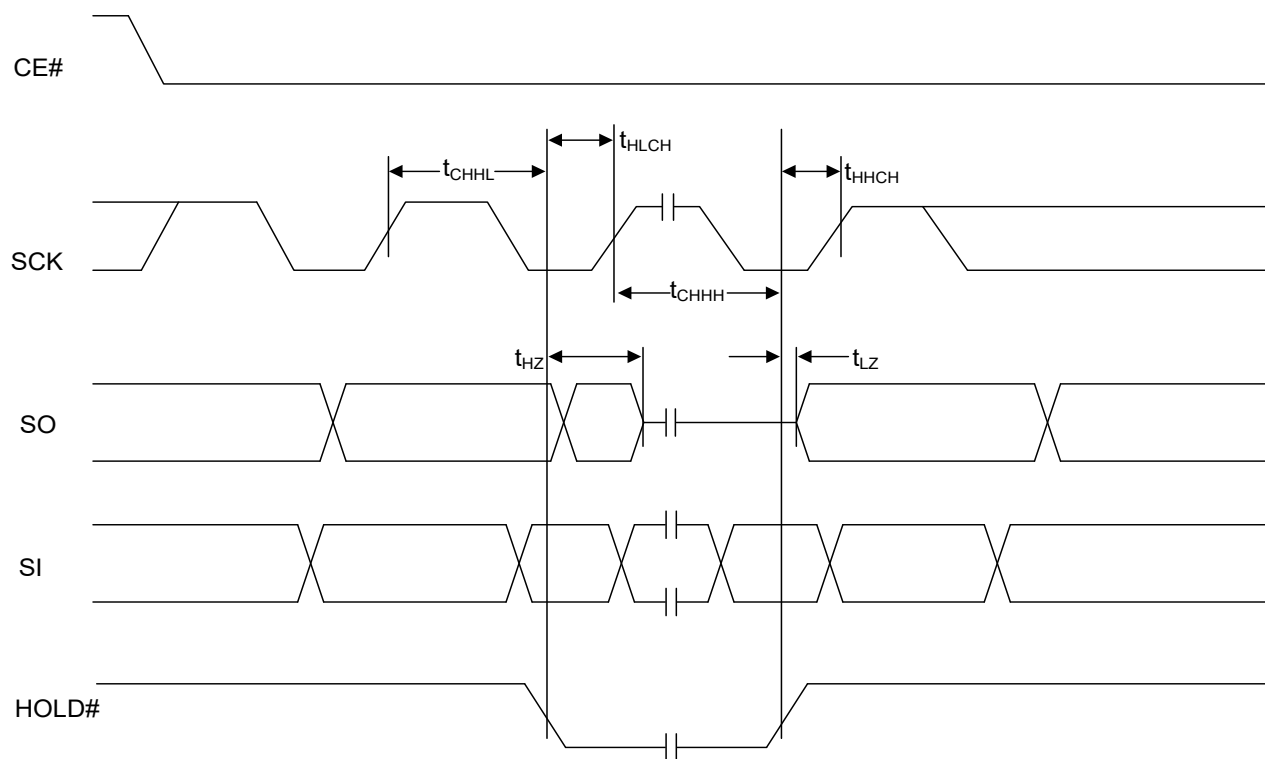
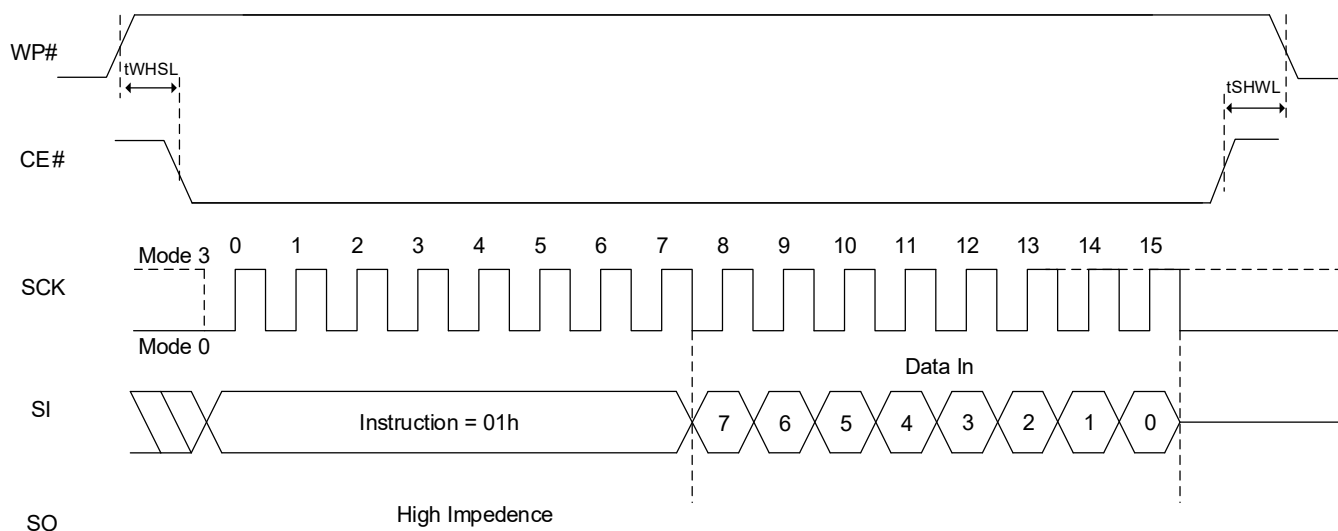
Figure 9.4 OUTPUT TIMING (STR Mode)

Figure 9.5 OUTPUT TIMING (DTR Mode)


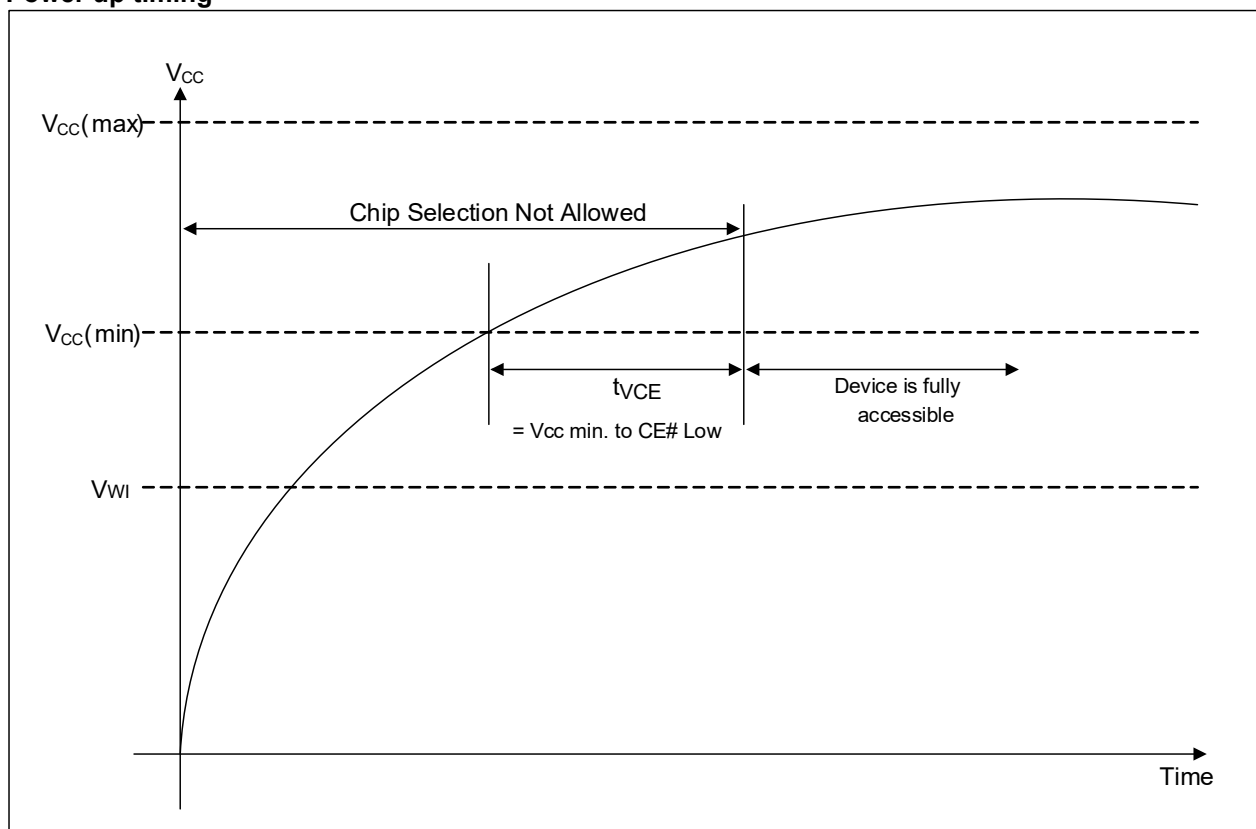
Figure 9.6 HOLD TIMING

Figure 9.7 WRITE PROTECT SETUP AND HOLD TIMING DURING WRITE STATUS REGISTER (SRWD=1)


Note: WP# must be kept high until the embedded operation finish.

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
$t_{VCE}^{(1)}$	Vcc(min) to CE# Low	300		us
$V_{WI}^{(1)}$	Write Inhibit Voltage		2.1	V

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

9.9 PROGRAM/ERASE PERFORMANCE

Parameter		Typ	Max	Unit
Sector Erase Time (4Kbyte)		50	300	ms
Block Erase Time (32Kbyte)		0.15	0.5	s
Block Erase Time (64Kbyte)		0.20	1.0	s
Chip Erase Time	4Mb	1	3	s
	8Mb	2	6	
Page Programming Time		0.2	0.8	ms
Byte Program		8	40	μs

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

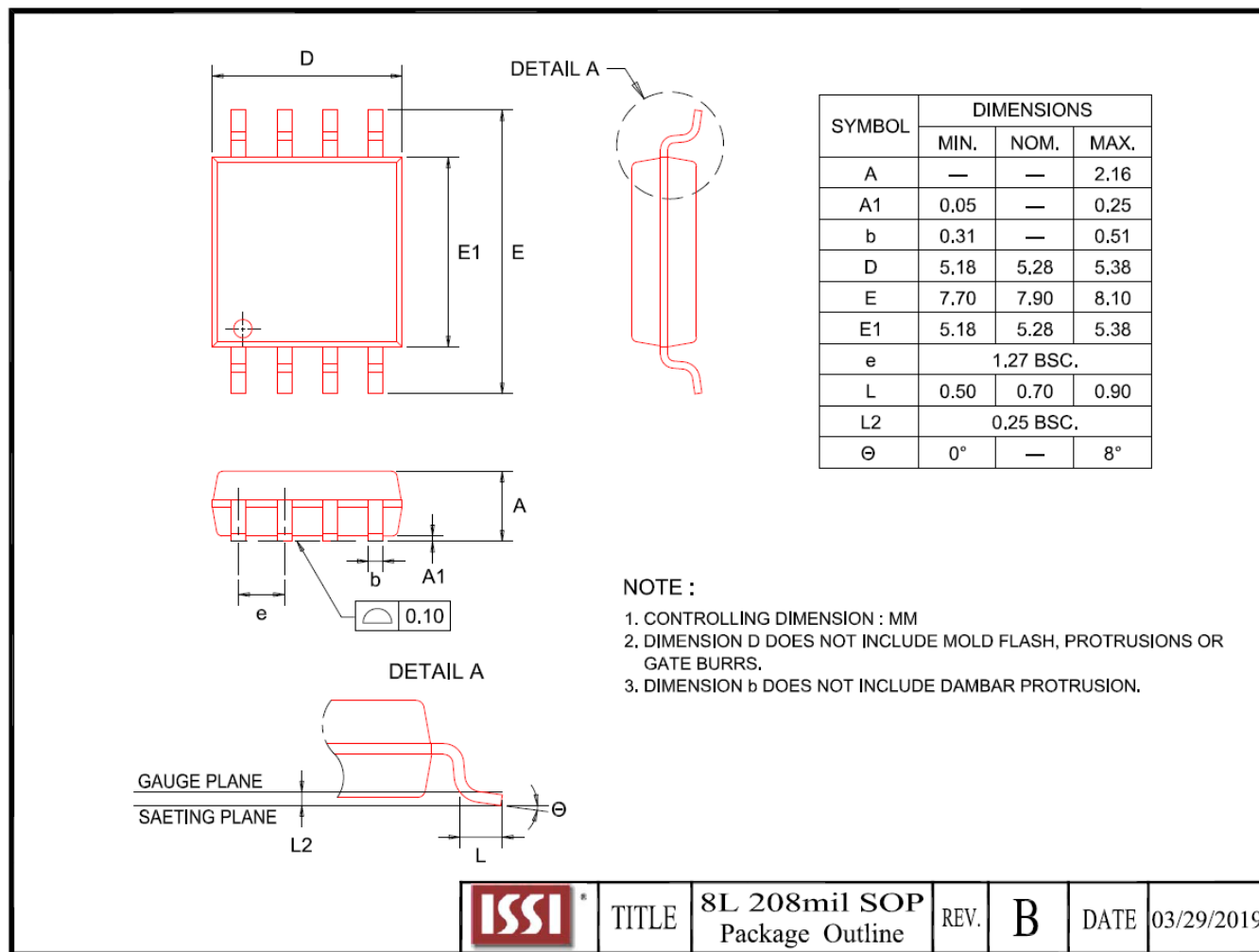
9.10 RELIABILITY CHARACTERISTICS

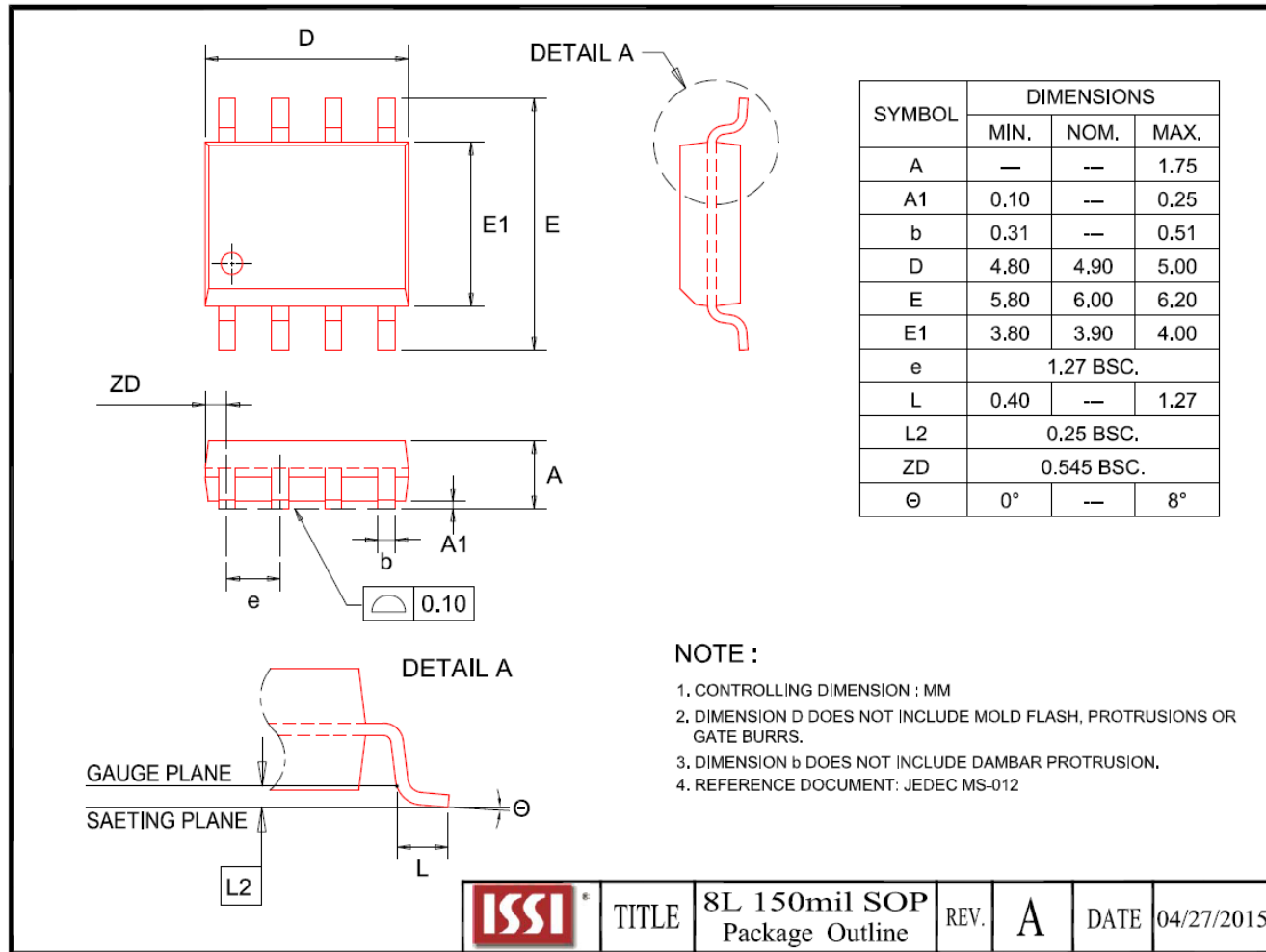
Parameter	Min	Max	Unit	Test Method
Endurance	100,000	-	Cycles	JEDEC Standard A117
Data Retention	20	-	Years	JEDEC Standard A117
Latch-Up	-100	+100	mA	JEDEC Standard 78

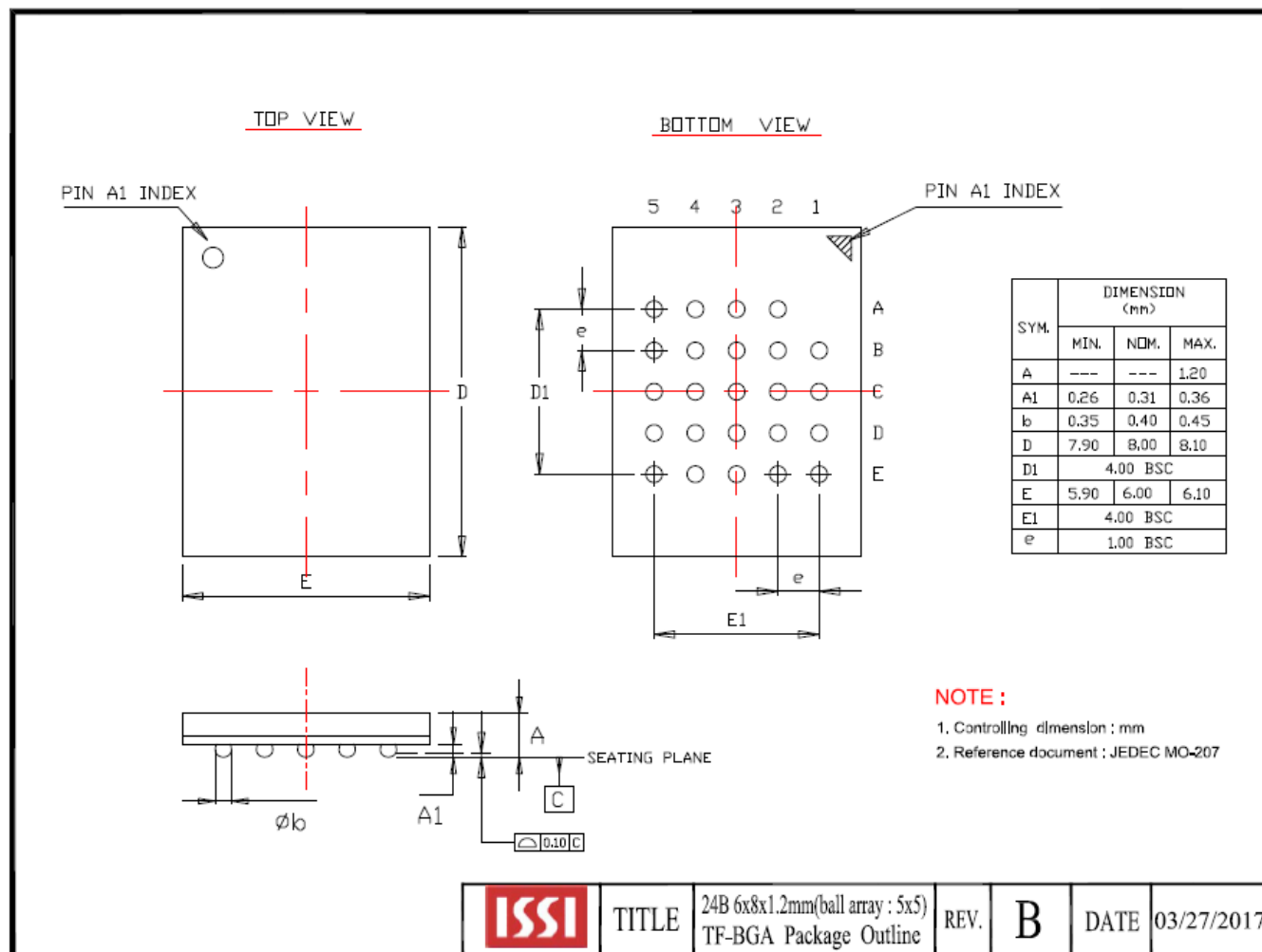
Note: These parameters are characterized and not 100% tested.

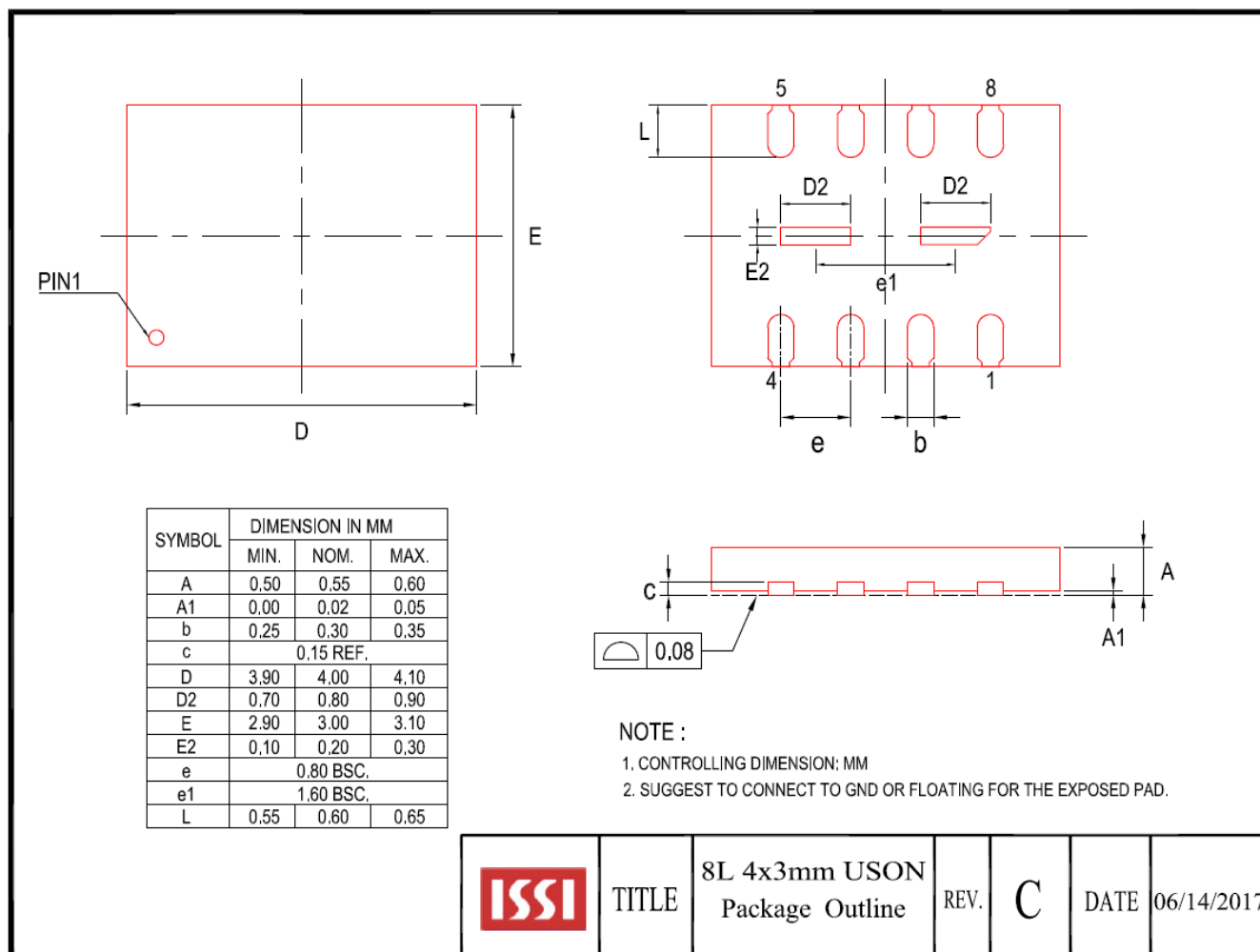
10. PACKAGE TYPE INFORMATION

10.1 8-PIN JEDEC 208MIL BROAD SMALL OUTLINE INTEGRATED CIRCUIT (SOIC) PACKAGE (B)



10.2 8-PIN JEDEC 150MIL BROAD SMALL OUTLINE INTEGRATED CIRCUIT (SOIC) PACKAGE (N)


10.3 24-BALL THIN PROFILE FINE PITCH BGA 6X8MM 5X5 BALL ARRAY (H)


10.4 8-CONTACT ULTRA-THIN SMALL OUTLINE NO-LEAD (USON) PACKAGE 4X3MM (T)


Note:

1. Please [click here](#) to refer to Application Note (AN25D011, Thin USON/WSO/XSON package handling precautions) for assembly guidelines.

11. ORDERING INFORMATION - Valid Part Numbers

IS25LE 080 G - J U L E - TR

PACKING

TR = Tape and Reel

TY = Tray

Blank = Bulk

TEMPERATURE RANGE

E = Extended (-40°C to +105°C)

A3 = Automotive Grade (-40°C to +125°C)

PACKAGING CONTENT

L = Green Package (RoHS Compliant, Halogen-free) and TSCA Compliant

PACKAGE Type ⁽¹⁾

H = 24-ball TFBGA (6x8mm) 5x5 ball array

B = 8-pin SOIC 208mil (Call Factory)

N = 8-pin SOIC 150mil (Call Factory)

T = 8-contact USON 4x3mm

W = KGD (Call Factory)

Option

J = Standard

Q = QE bit set to 1 (Call Factory)

R = Dedicated RESET# for 24-ball BGA (Call Factory)

Die Revision

G = Revision G

Density

080 = 8 Megabit

040 = 4 Megabit

BASE PART NUMBER

IS = Integrated Silicon Solution Inc.

25LE = FLASH, 2.70V ~ 3.60V, QPI & On-Chip ECC

Note:

1. Call Factory for other package options available



Density, Voltage	Frequency (MHz)	Order Part Number ⁽¹⁾	Package
8Mb, 3V	STR 166, DTR 80	IS25LE080G-JHLE	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LE080G-JTLE	8-contact USON 4x3mm
		IS25LE080G-JHLA3	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LE080G-JTLA3	8-contact USON 4x3mm

Density, Voltage	Frequency (MHz)	Order Part Number ⁽¹⁾	Package
4Mb, 3V	STR 166, DTR 80	IS25LE040G-JHLE	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LE040G-JTLE	8-contact USON 4x3mm
		IS25LE040G-JHLA3	24-ball TFBGA 6x8mm 5x5 ball array
		IS25LE040G-JTLA3	8-contact USON 4x3mm

Notes:**1. A3 meets AEC-Q100 requirements with PPAP.****Temp Grades: E= -40 to 105°C, A3= -40 to 125°C**