

## Designing with Cypress Quad SPI (QSPI) F-RAM™

**Author: Shivendra Singh**

**Associated Part Family: CY15x102QSx, CY15x104QSx, CY15x108QSx**

AN218375 discusses Quad SPI (QSPI) and shows how to design with Cypress' QSPI F-RAM. QSPI is an enhancement of the standard SPI protocol that provides up to four times the data throughput at higher frequencies while maintaining the compact form factor of the standard serial SPI. System designs using QSPI F-RAM devices occupy less board space with enhanced data throughput and features in a low-pin-count package, thus reduces overall system development and integration cost.

### Contents

1	Introduction.....	1	4.4	Dummy Phase.....	11
2	QSPI F-RAM Signals and Interface .....	2	4.5	Data Phase.....	12
	2.1 QSPI F-RAM Signaling Details .....	2	5	Power on Reset (POR) and Initialization.....	15
	2.2 QSPI F-RAM Interface Description .....	3	5.1	Power Cycle and Power on Reset (POR).....	15
3	System Interface.....	5	5.2	Device Initialization.....	16
4	Command Protocol.....	7	6	Hardware/Software Reset.....	17
	4.1 Opcode Phase .....	8	6.1	Hardware Reset (RESET) .....	17
	4.2 Address Phase.....	8	6.2	Default Recovery (JEDEC SPI Reset).....	18
	4.3 Mode Phase.....	10	6.3	Software Reset.....	18

## 1 Introduction

In order to manage the wide range of multimedia, graphics, and other data intensive content, embedded systems have evolved to offer more sophisticated features. These features place extra demands on the often-limited on-chip memory of the host controller (or MCU). External memories with parallel interface have long been used to extend the on-chip MCU's storage limitations. Memories with a parallel address/data bus come in high pin-count packages and require more pins on the controller to communicate with.

Cypress has developed the Quad SPI (QSPI) Ferroelectric RAM (F-RAM) to mitigate the above challenges by offering a high-performance memory in a compact-footprint and high-speed serial access. The QSPI F-RAM can be used as nonvolatile RAM for storing critical system parameters, system code, images and icons among other things, accessible at a high speed. Unlike a traditional RAM or a buffered memory, the F-RAM provides instantaneous non-volatility, and hence eliminates any system power backup that is required to save critical states of a system before any sudden power loss or power glitch.

High reliability applications such as industrial controls and automation, computing, automotive safety systems, medical equipment and imaging, and high-end data loggers are often designed with a dedicated power back-up so that they can capture all critical transactions in the last moment without any risk of losing them in any circumstance. The QSPI F-RAM can play a pivotal role in such system architectures by eliminating any power back-up required for storing any last moment critical transactions into a nonvolatile memory such as hard disk or flash for a safe power up during subsequent power up cycle.

The QSPI F-RAM supports single data rate (SDR) for all its SPI interface options up to 108 MHz, while it also supports the double data rate (DDR) up to 54 MHz but for specific opcodes. For more details, see the QSPI F-RAM datasheet. The 54-MHz DDR interface offers the same data throughput as the 108-MHz SDR but at half the frequency. Some systems preferably use DDR at reduced frequency than the high-speed SDR, which helps reducing the system core and I/O frequencies, thus system power without compromising the data throughput.

## 2 QSPI F-RAM Signals and Interface

The QSPI F-RAM is a low-pin-count serial interface device which supports various SPI interface options that include traditional (or single-channel) SPI, extended SPI, and exclusive Dual SPI (DPI) and Quad SPI (QPI) which are either enabled through dedicated opcodes or through configuration settings via its configuration registers.

### 2.1 QSPI F-RAM Signaling Details

The QSPI F-RAM is available in compact 8-pin package footprints – 8-pin wide SOIC (EIAJ) and 8-pin Grid QFN (GQFN). The QSPI F-RAM offers multiplexed I/Os and control pins to support all the above SPI interface options through the 8-pin package. Table 1 describes the details of the SPI signals and their mapping to corresponding SPI interface.

Figure 1. QSPI F-RAM Block Diagram

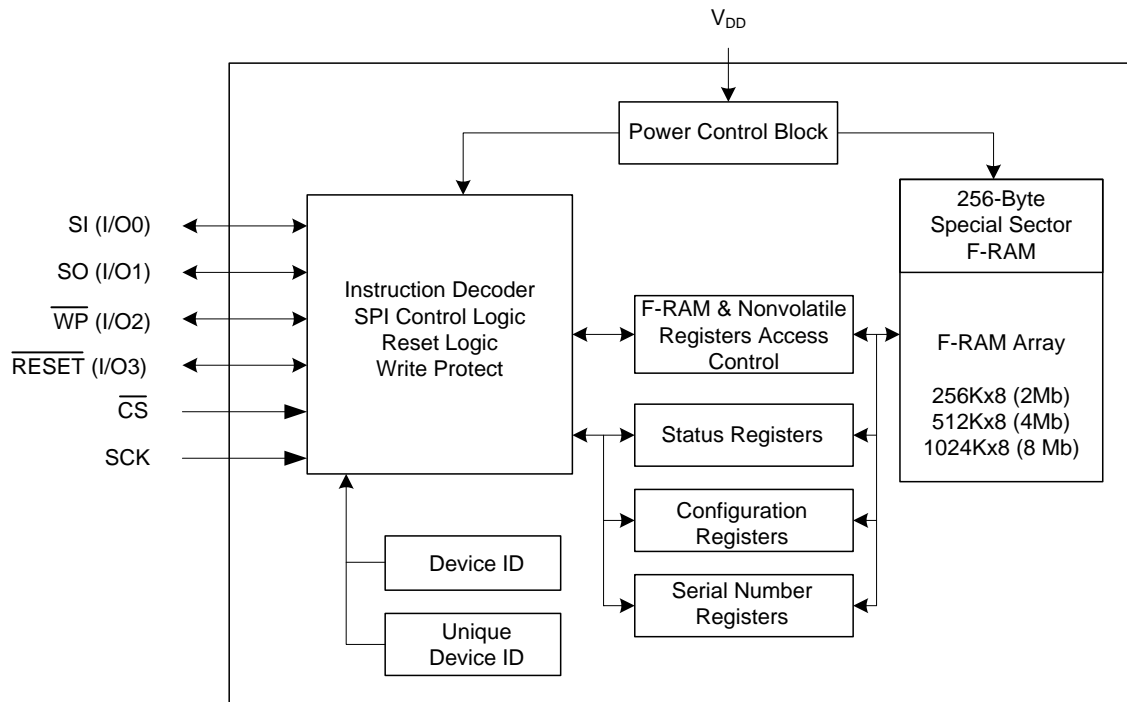


Table 1. QSPI F-RAM Signals

Signal Name	Type	Signal Description	Signal Mapping for Different SPI Interfaces		
			Single SPI	Dual SPI (Enhance SPI Dual and DPI)	Quad SPI (Enhance SPI Quad and QPI)
SCK	Input	Serial Clock	SCK		
$\overline{\text{CS}}$	Input	Chip Select	$\overline{\text{CS}}$		
SI / (I/O0)	Input	Serial Input in standard SPI	SI	I/O0	I/O0
	Input/ Output	I/O0 in dual or quad modes			
SO / (I/O1)	Input	Serial output in standard SPI	SO	I/O1	I/O1

Signal Name	Type	Signal Description	Signal Mapping for Different SPI Interfaces		
			Single SPI	Dual SPI (Enhance SPI Dual and DPI)	Quad SPI (Enhance SPI Quad and QPI)
	Input/ Output	I/O1 in dual or quad modes			
$\overline{WP}$ / (I/O2)	Input	Hardware write protection in standard SPI	$\overline{WP}$	$\overline{WP}$	I/O2
	Input/ Output	I/O2 in quad SPI mode			
$\overline{RESET}$ / (I/O3)	Input	Hardware reset in standard SPI and DPI	$\overline{RESET}$	$\overline{RESET}$	I/O3
	Input/ Output	I/O3 in quad SPI mode			

## 2.2 QSPI F-RAM Interface Description

Low pin count serial interface devices reduce the total number of pins required to interface with the host system by serially transferring all control, address, mode (if applicable), and data. This reduces the cost of the package, signal switching power, and the host side I/O count, so that host can use extra I/Os to enable additional features.

The QSPI F-RAM supports the following four modes:

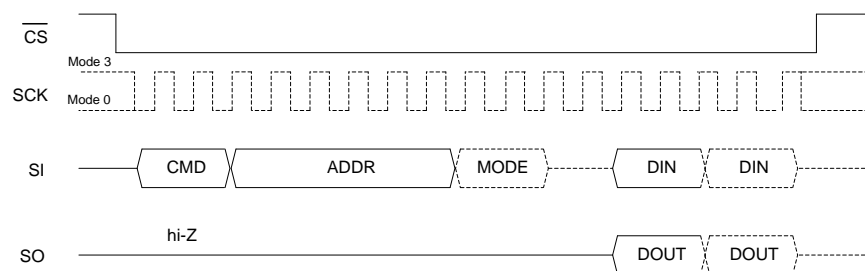
1. Single-channel SPI
2. Extended SPI
3. Dual SPI
4. Quad SPI

The command I/O count, address I/O count, mode I/O count (if applicable), and data I/O count vary for each SPI mode and represented as (w, x, y, z) where w is the I/O count for command, x is the I/O count for address, y is the I/O count for mode byte (if applicable), and z is the I/O count for input and output.

### 2.2.1 Single-Channel SPI

Single-channel SPI mode (1,1,1,1) uses SI (MOSI - Master Out Slave In) and SO (MISO-Master In Slave Out) pins for input and output, respectively. Opcode, address, and mode byte are transferred by the master on the SI line, while data is read by the master on SO. Figure 2 shows single channel SPI mode transfer.

Figure 2. Single Channel SPI (1,1,1,1)



### 2.2.2 Extended SPI

The extended SPI mode provides Dual data (1,1,1,2), Dual address/data or Dual I/O (1,2,2,2), Quad data (1,1,1,4), and Quad address/data or Quad I/O (1,4,4,4) operating modes.

There is no specific configuration bit to be set to enable the Extended SPI modes. However, when the command is sent for the extended SPI in Quad mode (Quad Data or Quad Address/Data), it is required to set the QUAD bit to '1' in CR1 (CR1[1]) to disable the  $\overline{WP}$  and  $\overline{RESET}$  functions which turn around these two pins as I/O2 and I/O3. SI and SO are I/O0 and I/O1 respectively for Dual and Quad I/O modes. Figure 3 to Figure 6 show Extended SPI transfers in various modes.

Figure 3. Extended SPI – Dual Data (1,1,1,2)

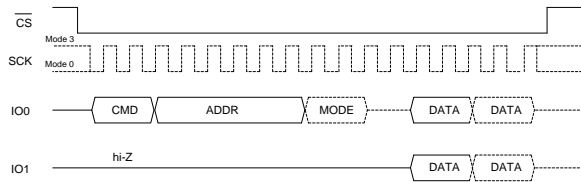


Figure 5. Extended SPI – Dual Address/Data (1,2,2,2)

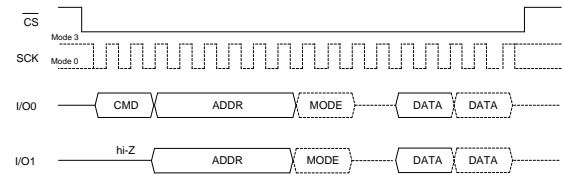


Figure 4. Extended SPI – Quad Data (1,1,1,4)

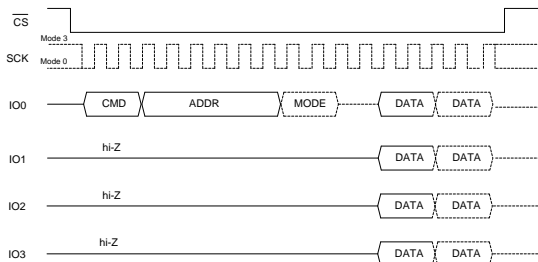
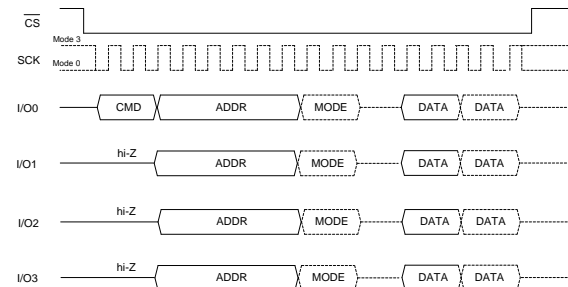


Figure 6. Extended SPI – Quad Address/Data (1,4,4,4)



### 2.2.3 Dual/Quad SPI (DPI/QPI)

The multi-channel Dual and Quad SPI or DPI (2,2,2,2) and QPI (4,4,4,4) modes are used to further enhance the SPI bandwidth by allowing opcode, address, mode, and data transfer on two or four I/Os. These modes can be enabled through Configuration Register 2 (CR2) by setting bit 6 (CR2[6] = '1') for QPI or bit 4 (CR2[4] = '1') for DPI. Once the device is configured either in DPI or QPI mode, it retains the interface configuration until it is changed by overwriting with the new configuration. For a description of configuration registers, see the QSPI F-RAM datasheet. [Figure 7](#) and [Figure 8](#) show DPI and QPI mode transfers.

Figure 7. Dual SPI – DPI (2,2,2,2)

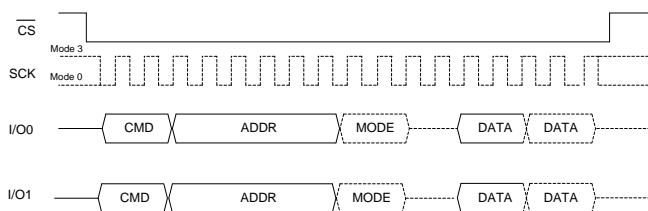
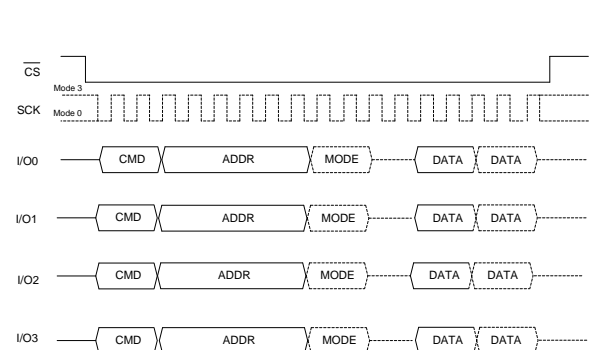


Figure 8. Quad SPI – QPI (2,2,2,2)



### 2.2.4 Quad SPI SDR and DDR

The QPI mode also supports double data rate (DDR) ( $4_{SDR}$ ,  $4_{DDR}$ ,  $4_{DDR}$ ,  $4_{DDR}$ ) through special opcodes where byte transfer occurs on both edges of the clock for address, mode, and data bytes. There is no DDR mode during the opcode phase, i.e. opcodes are always transmitted in SDR mode. Device enters DDR mode after a specific command is transmitted in SDR mode, which then determines the address, mode, and data cycles in DDR. There is no setting for enabling the DDR mode. The Quad SPI DDR mode is only supported for memory write and read operations with special opcodes.

### 3 System Interface

The QSPI F-RAM supports SPI, DPI, and QPI interfaces in a single device configurable via Configuration Register 2 (CR2). The QSPI F-RAM I/Os are multiplexed with other functions as described in Table 1. Device pins are either configured to execute a dedicated function or enabled as I/O depending on the SPI interface mode configured. Figure 9 to Figure 12 show examples of QSPI F-RAM interface with a host controller for different SPI interface options.

The signal and device nomenclature used in Figure 9 to Figure 12 are as follows:

*MOSI*: Master Out Slave In

*MISO*: Master In Slave Out

**CY15x102QS/CY15x104QS/CY15x108QS:**

CY15V102QS/CY15V104QS/CY15V108QS – 1.8 V typical parts

CY15B102QS/CY15B104QS/CY15B108QS – 3 V typical parts

Data lines (Input, Output, I/O)

Control line ( $\overline{CS}$ , SCK and controls)

(Optional connections)

Figure 9. System Interface with SPI Port

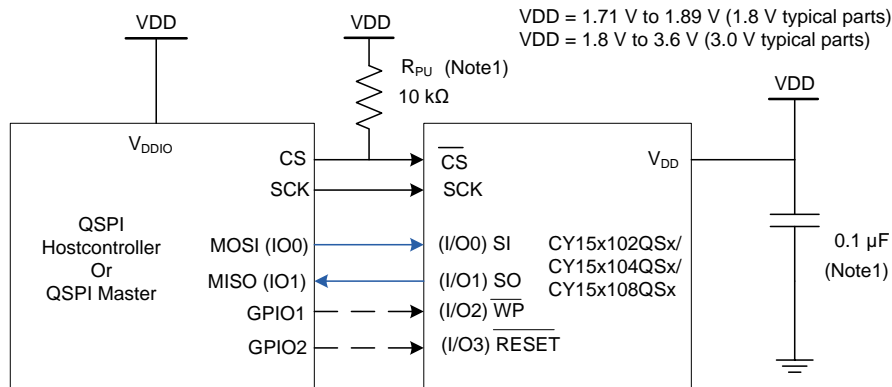


Figure 10. System interface with Dual SPI Port

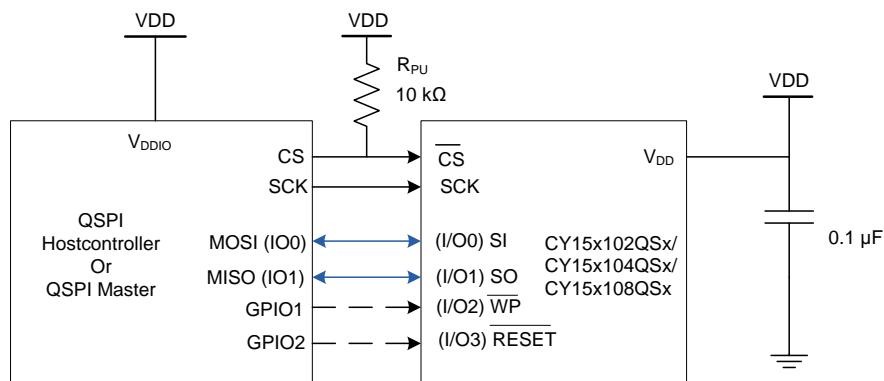


Figure 11. System Interface with a Single Quad SPI device

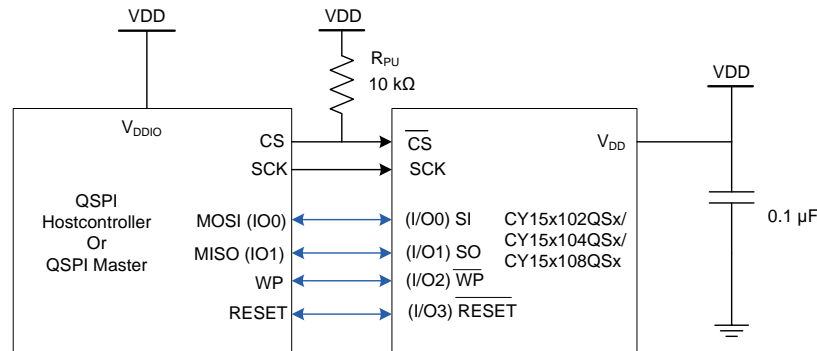
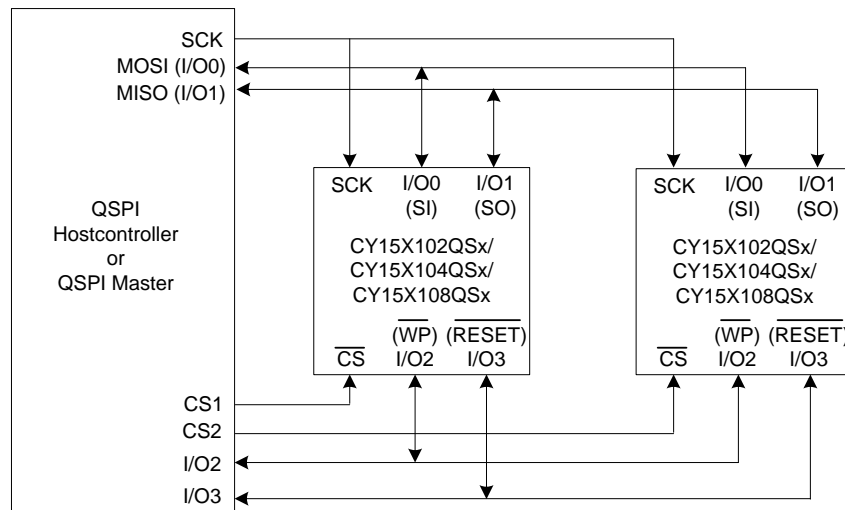


Figure 12. System interface with Multiple Quad SPI Devices


**Note: Pull-up resistor ( $R_{PU}$ ) value:**

The value of the pull-up resistor is determined as per the following equation:

$$V_{DD} - (I_{LEAK} \times R_{PU}) \geq V_{IH} (min)$$

- $V_{DD}$  – operating voltage
- $V_{IH} (min)$  is 70% of  $V_{DD}$ , per datasheet
- $I_{LEAK}$  is the total leakage current combining the leakage of all input pin/s and output pin/s in tri-state connected. In the above cases,  $I_{LEAK}$  includes the leakage current of the QSPI master output buffer in tristate mode and the control input pin of the QSPI F-RAM.

**Bypass capacitor value:**

The bypass capacitor at  $V_{DD}$  pin is used to filter out all high frequency noise generated by switching of the device circuits and I/Os. The bypass capacitor on  $V_{DD}$  ensures that the operating voltage on  $V_{DD}$  doesn't drop below  $V_{DD} (min)$  at any point of time during the operation. Cypress recommends connecting at least one 0.1  $\mu F$  on the  $V_{DD}$  pin to ensure reliable device operation.

## 4 Command Protocol

The QSPI F-RAM command cycle consists of up to five different command phases - Opcode, Address, Mode, Dummy (Latency), and Data. The number of command phases per command cycle varies from one to five depending on the opcode sent in the Opcode phase. The Opcode, Address, Mode, and Data phases are configurable in terms of number of lines 1, 2, or 4 needed to transmit them in SPI, DPI, or QPI interface, respectively. Table 2 shows the command phases for each command cycle in different SPI interfaces. Figure 13 to Figure 15 show examples of command phases for SPI, DPI, and QPI interfaces.

Table 2. Command Transmission over I/Os in Different SPI Modes

Command phases	Command transmission on I/Os (Command I/Os, Address I/Os, Mode I/Os, Data I/Os)						
	Single Channel SPI (1,1,1,1)	Extended SPI				Multi-Channel SPI	
		Dual Data (1,1,1,2)	Quad Data (1,1,1,4)	Dual I/O (1,2,2,2)	Quad I/O (1,4,4,4)	DPI (2,2,2,2)	QPI (4,4,4,4)
Opcode	SI	I/O0	I/O0	I/O0	I/O0	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3
Address	SI	I/O0	I/O0	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3
Mode	SI	I/O0	I/O0	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3
Dummy (Latency)	Fixed number of dummy SPI clocks, independent of SPI interface. 0 to 15 clocks for memory access (configurable via CR1[7:4]) 0 to 3 clocks for register access (configurable via CR5[7:6])						
Data	SI/SO	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3	I/O0, I/O1	I/O0, I/O1, I/O2, I/O3

Figure 13 to Figure 15 show an example of a read command cycle involving all five phases shown in Table 2. For details on specific commands and their phases, see the device datasheet.

Figure 13. Command Phases in SPI Interface

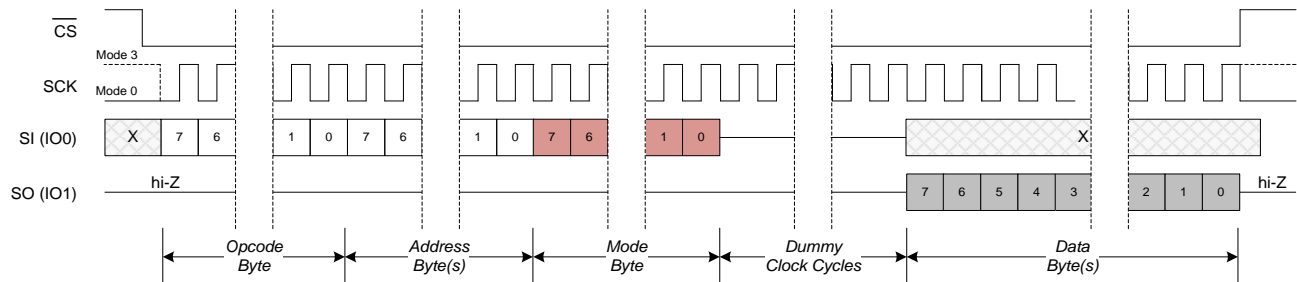


Figure 14. Command Phases in DPI Interface

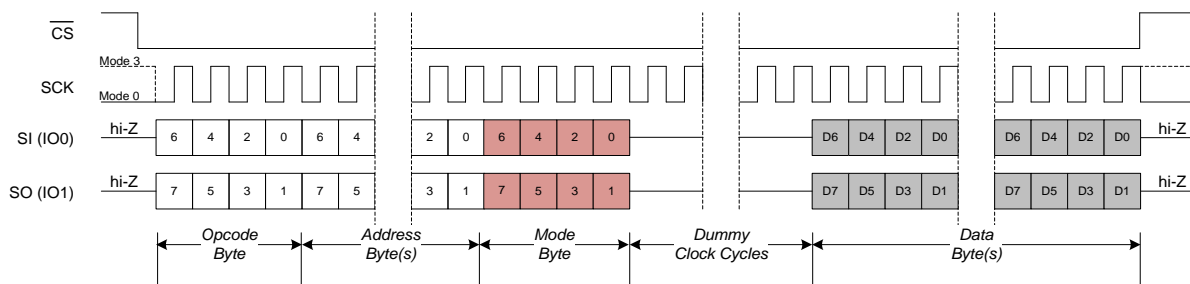
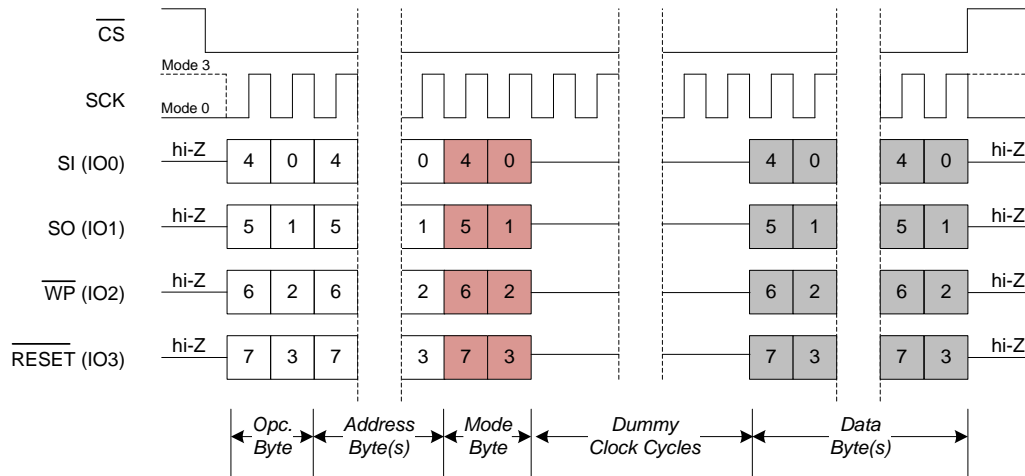


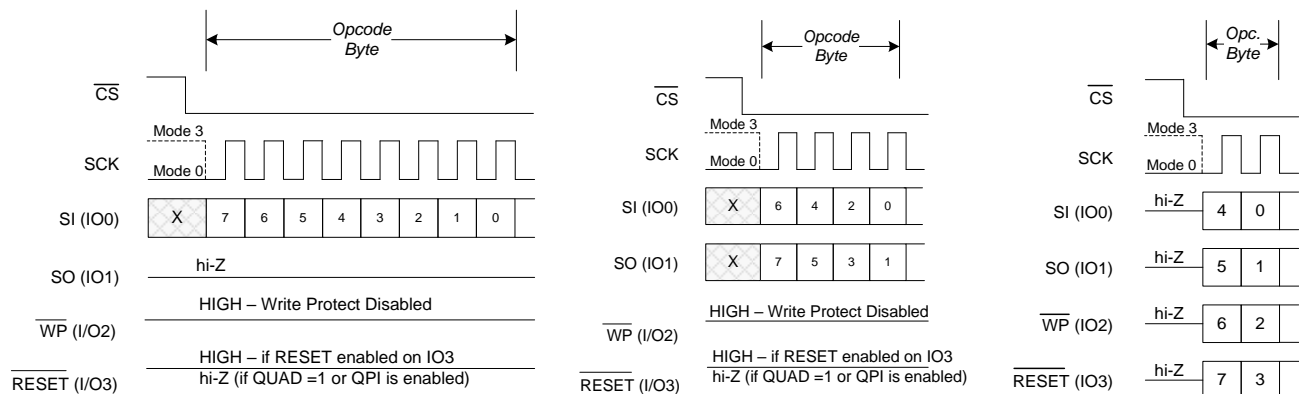
Figure 15. Command Phases in QPI Interface



#### 4.1 Opcode Phase

An 8-bit opcode is sent during the Opcode phase to initiate an intended device operation in QSPI F-RAM. Depending on the device configuration (SPI, DPI, or QPI), the opcode can be sent on one, two, or four I/Os. In some cases, where only the Opcode phase is sent, other phases should be skipped. Depending upon the SPI mode and the interface, the number of clocks to transmit the opcode will vary from two clocks (Quad, SDR) to eight clocks (in SPI, SDR).

Figure 16. Opcode Phases in SPI/DPI/QPI Interface



**Note:** A DDR access opcode is always transmitted in SDR mode. Once the opcode is decoded, the device decides whether the subsequent bytes to be received are in SDR or DDR.

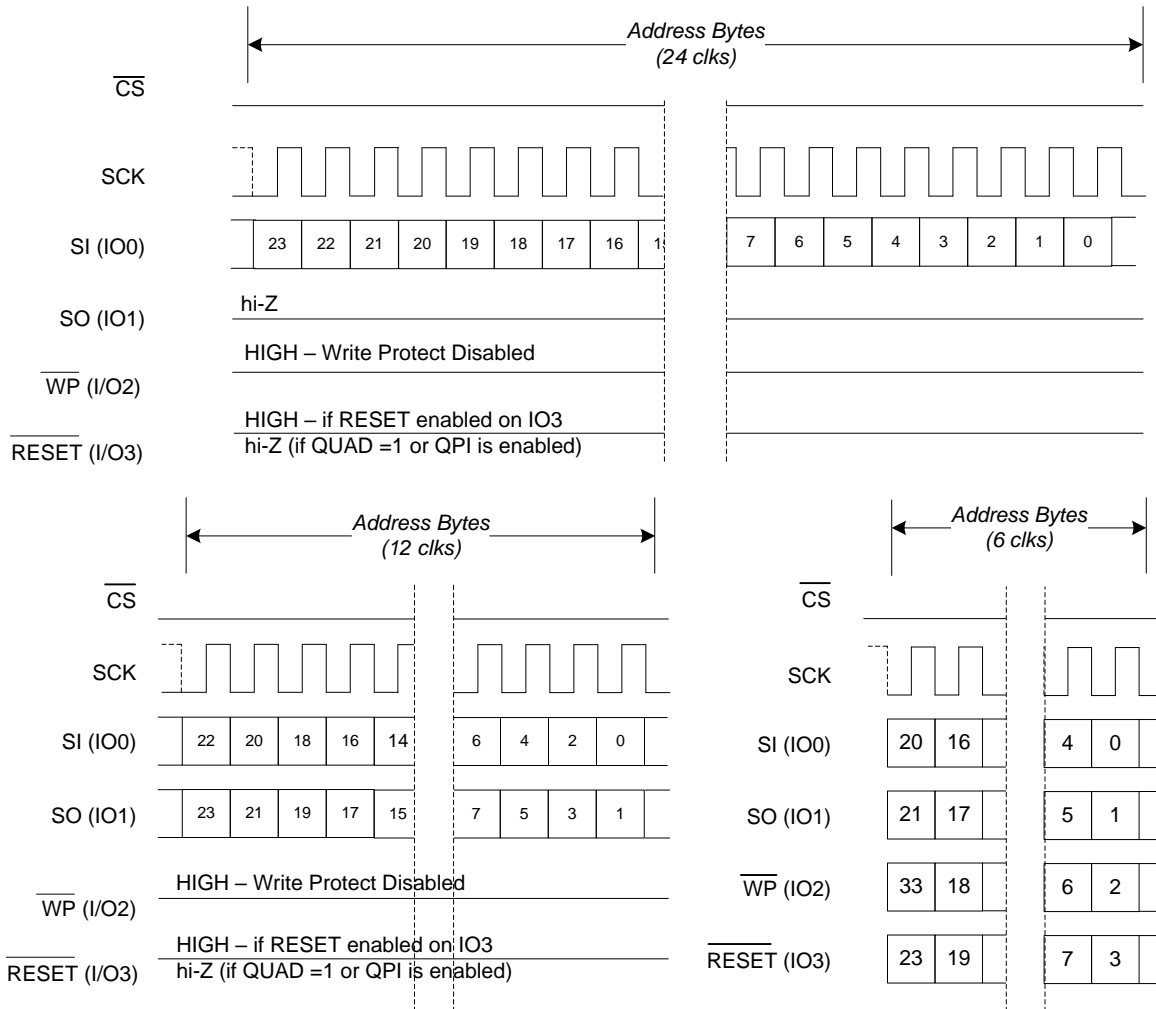
#### 4.2 Address Phase

A 3-byte address is transmitted during the Address phase. Depending upon the SPI mode and the interface type, the number of clocks to transmit the 3-byte address will vary from three clocks (Quad, DDR) to 24 clocks (SPI, SDR).



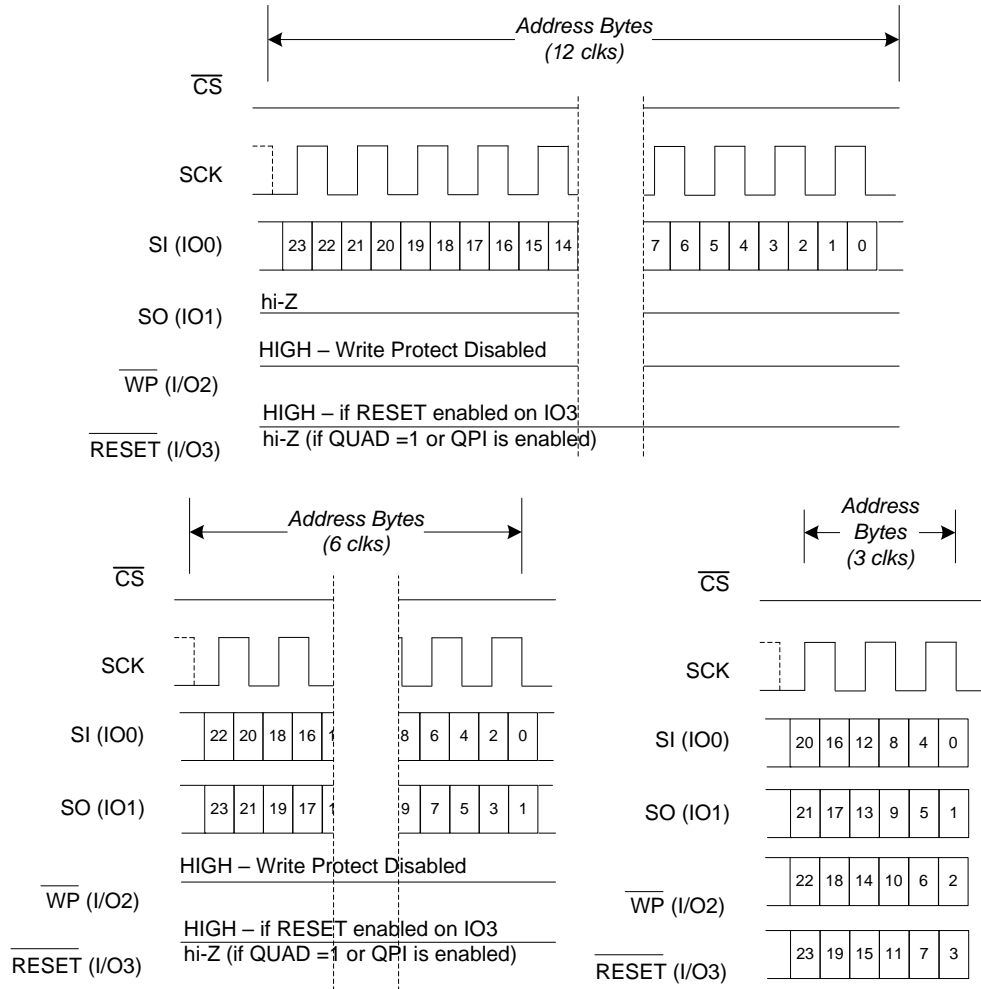
### 4.2.1 Address Phase – SDR

Figure 17. Address Phase - SDR in SPI/DPI/QPI Interface



### 4.2.2 Address Phase - DDR

Figure 18. Address Phase - DDR in SPI/DPI/QPI Interface



### 4.3 Mode Phase

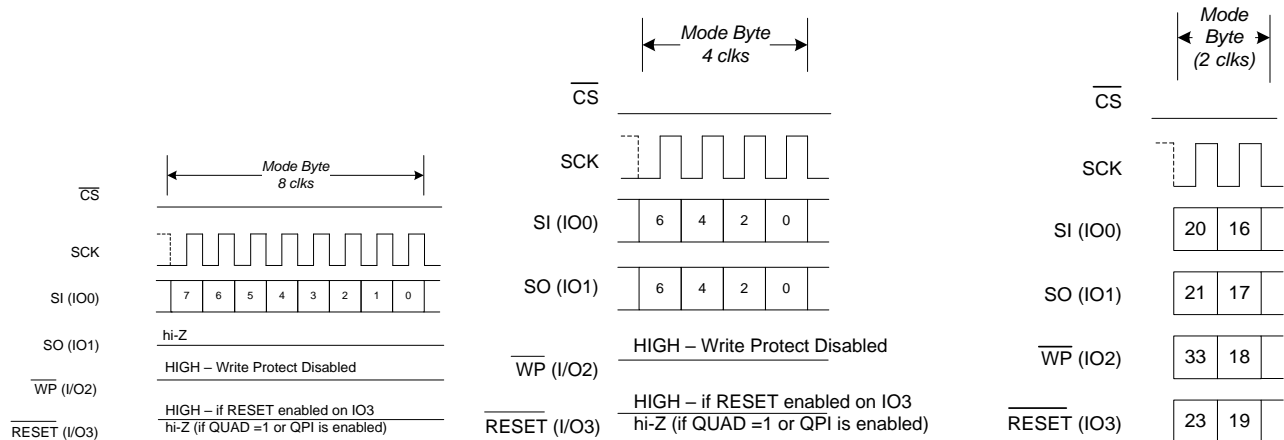
The Mode phase is applicable for all write and read commands that support Execute-In-Place (XIP). The XIP is a method of executing the program (code) directly from an external memory rather than copying or shadowing the code into RAM. When the XIP is set for a write or read command, the device stays in XIP mode after the command cycle is terminated ( $\overline{CS}$  toggle HIGH) so that the subsequent command cycle with  $\overline{CS}$  LOW directly starts with the Address phase (Opcode phase is skipped). When in XIP, the device executes the same operation as in previous cycle.

Following the opcode and 3-byte address cycles, the mode byte 0xA5 (X don't care bits) or 0xA5 (depending on the opcode) transmitted during the Mode phase keeps the device in XIP for the next command cycle. Any other value than 0xA5 or 0xA5 (!0xA5 or !0xA5) transmitted during the Mode phase will exit the XIP.

Depending upon the SPI mode and the interface type, the number of clocks to transmit the mode byte will vary from one clock (Quad, DDR) to eight clocks (SPI, SDR).

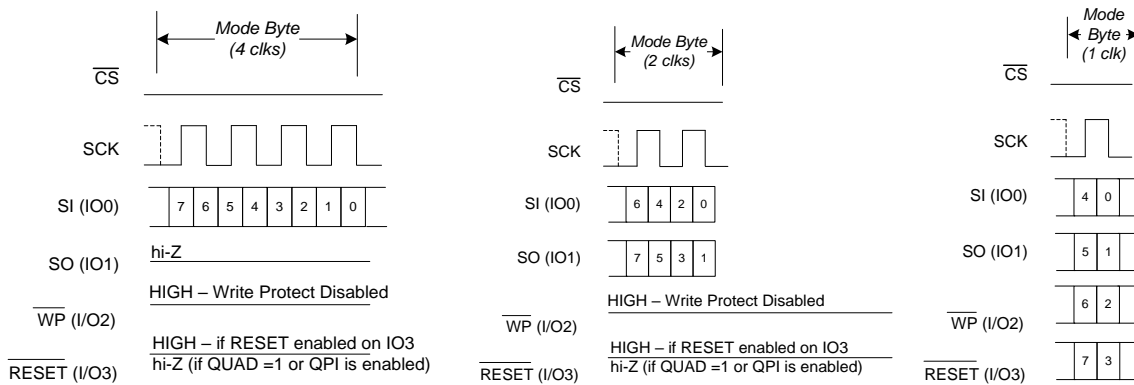
### 4.3.1 Mode Phase - SDR

Figure 19. Mode Phase - DDR in SPI/DPI/QPI Interface



### 4.3.2 Mode Phase - DDR

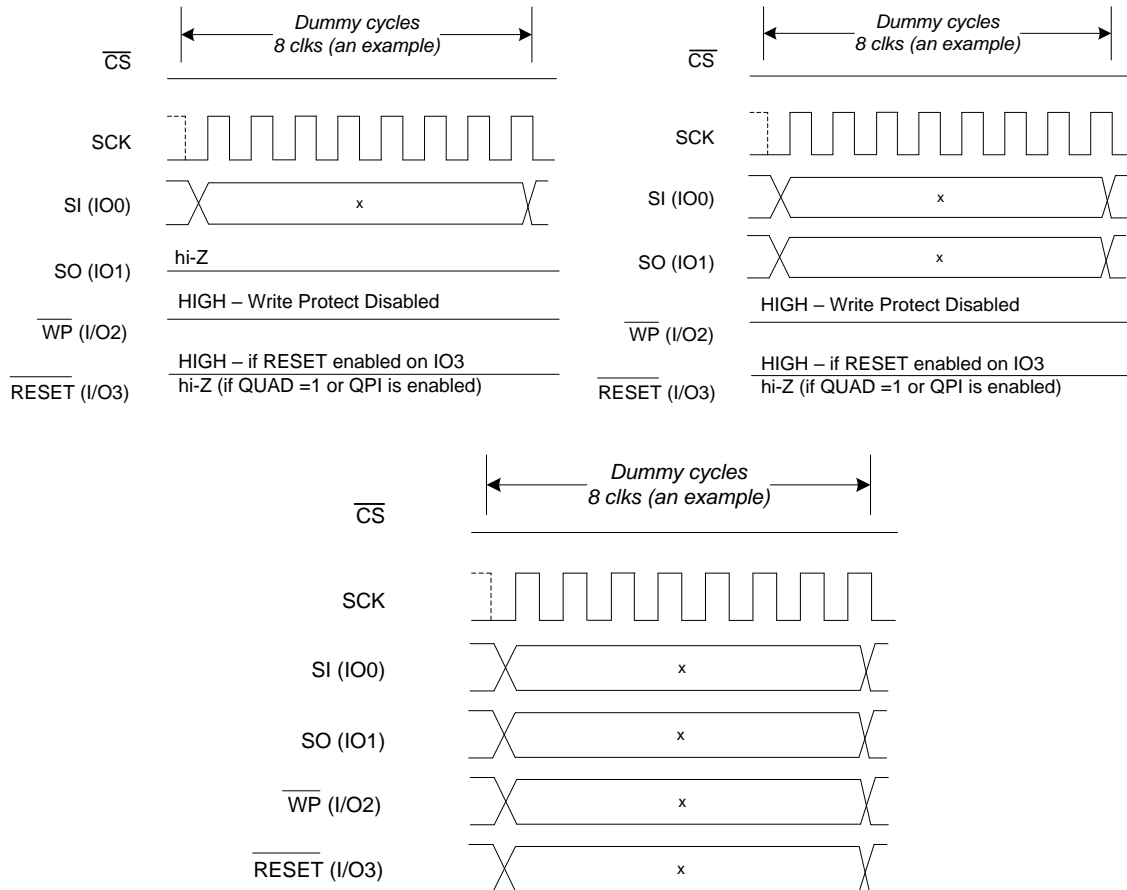
Figure 20. Mode Phase - DDR in SPI/DPI/QPI Interface



## 4.4 Dummy Phase

The Dummy phase transmits dummy clocks to provide the required clock latency. The number of SPI clocks (SCK) to be transmitted as dummy clocks is determined by the number of latency clocks programmed for the memory or the register access in respective configuration registers. The number of clocks is fixed per their programmed value and do not change with SPI mode and/or the interface type. Dummy clocks for the memory access opcodes can be set between 0 to 15 clocks through 4-bit Memory Latency Code (MLC) in Configuration Register 1 (CR1[7:4]). Similarly, dummy clocks for the register access can be set between 0 to 3 clocks through 2-bit Register Latency Code (RLC) in Configuration Register 5 (CR5[7:6]). Dummy phase is only applicable during memory and register read operations, it is not applicable during writes. The status of I/O(s) are don't care during the Dummy phase. Therefore, the host controllers can keep I/Os tristated during the Dummy phase.

Figure 21. Mode Phase - DDR in SPI/DPI/QPI Interface

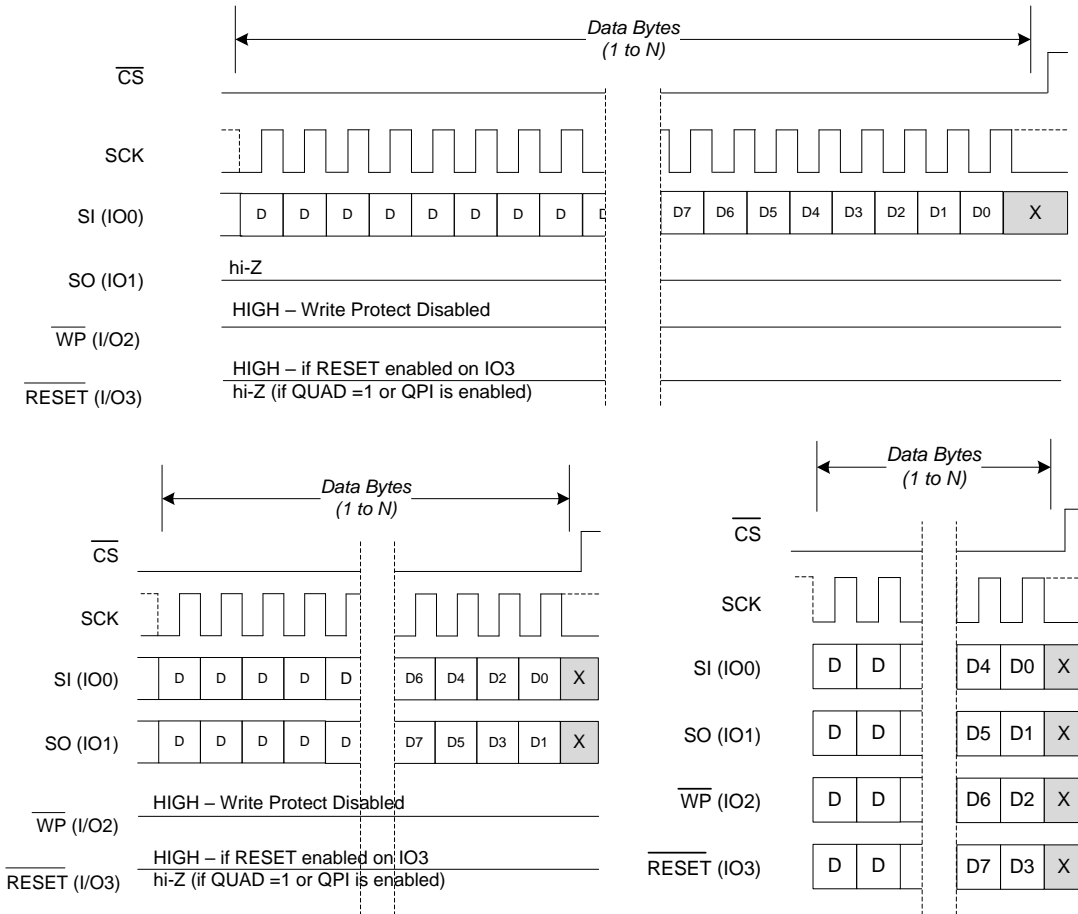


#### 4.5 Data Phase

The Data byte phase either transmits data to the QSPI F-RAM during write operations or receives data from QSPI F-RAM during read operations. The data length can vary from one byte to the entire memory array. Depending upon the SPI mode and the interface type, the number of clocks to transmit one data byte will vary from one clock (Quad, DDR) to eight clocks (SPI, SDR).

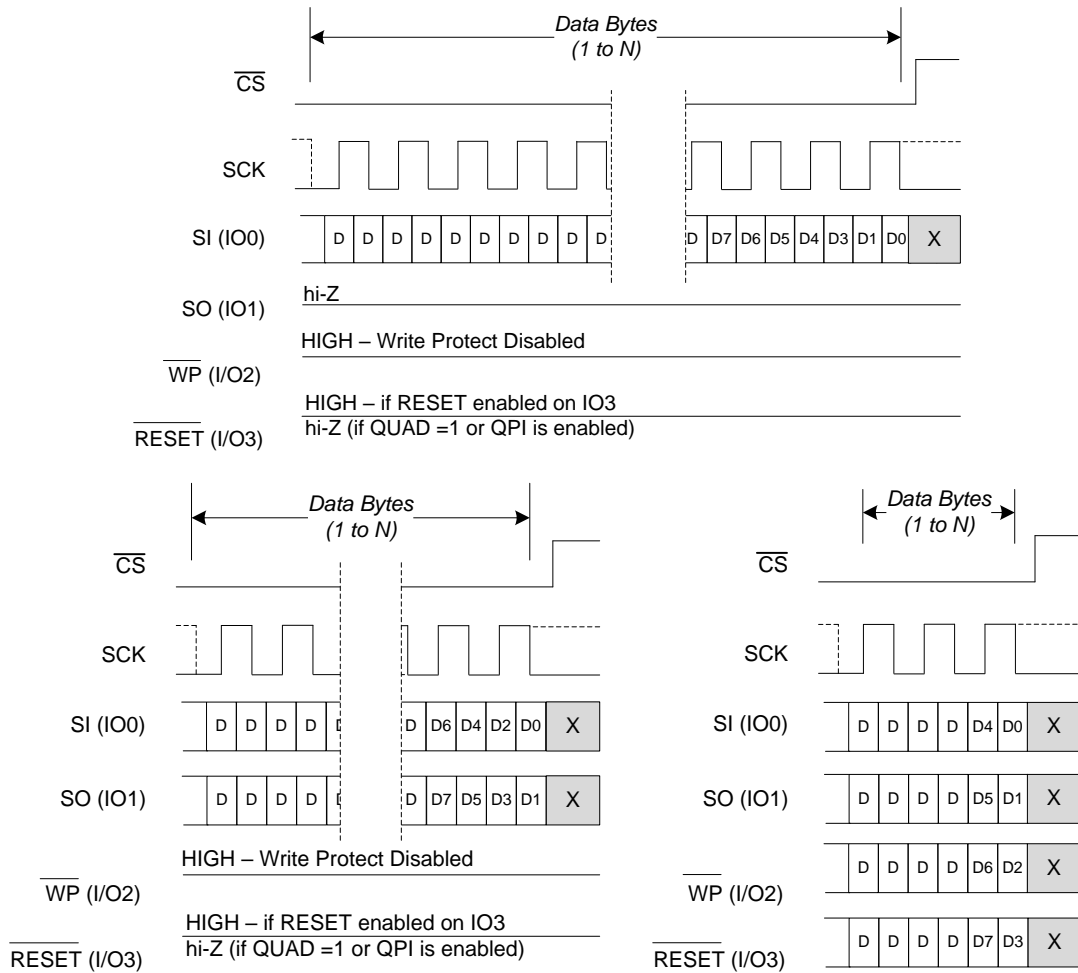
## 4.5.1 Data Phase -SDR

Figure 22. Data Phase - SDR in SPI/DPI/QPI Interface



### 4.5.2 Data Phase - DDR

Figure 23. Data Phase - DDR in SPI/DPI/QPI Interface



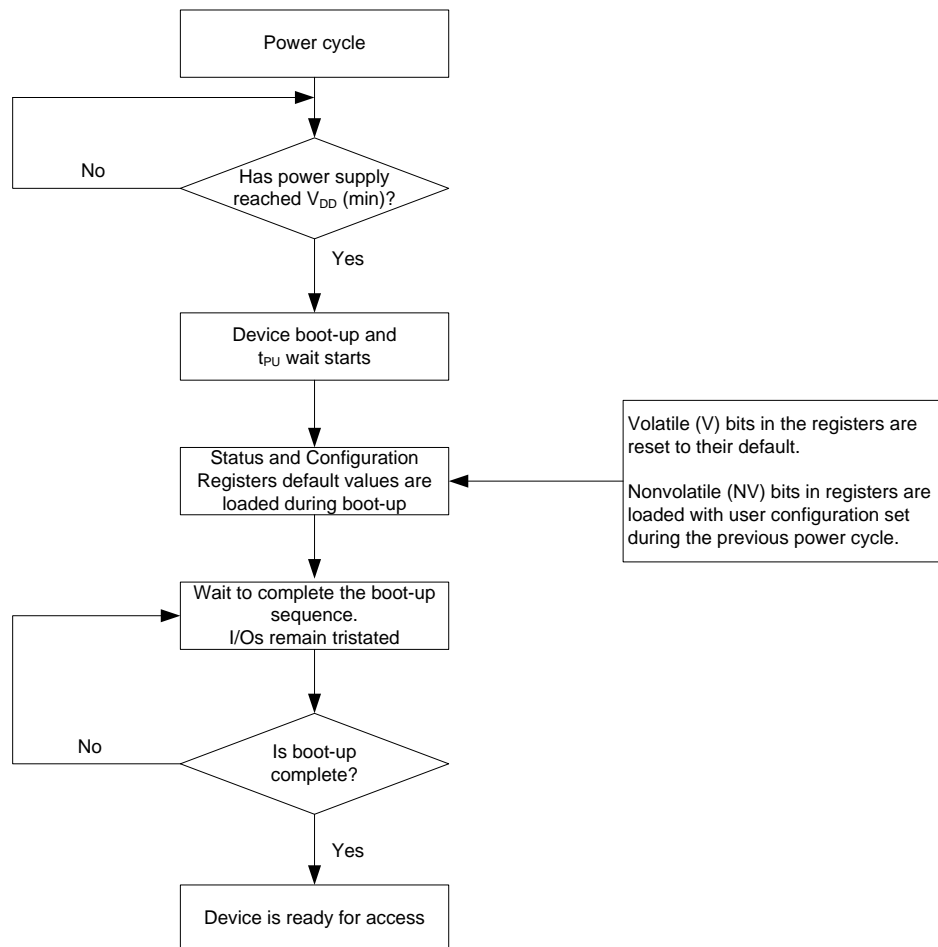
## 5 Power on Reset (POR) and Initialization

### 5.1 Power Cycle and Power on Reset (POR)

When the QSPI F-RAM encounters a power cycle event, it starts its boot up cycle and ignores all commands until the  $t_{PU}$  time has elapsed after  $V_{DD}$  rises above the minimum  $V_{DD}$ .

The boot up cycle includes reloading all internal configurations and settings and keeping the device ready for access. Figure 24 shows the device boot up cycle after POR. QSPI F-RAM registers are loaded with default values as shown in Table 5.

Figure 24. Power Cycle and POR



In case the device fails to boot up correctly after a power cycle, reinitiating the power cycle or hardware reset by asserting  $\overline{RESET}$  LOW will restart the boot up cycle again and will take no more than the  $t_{PU}$  time to complete. After  $t_{PU}$ , if  $\overline{CS}$  is HIGH, the device enters Standby mode and draws the standby current ( $I_{SB}$ ). The device can be configured to enter Deep Power Down mode after  $t_{PU}$  if the DPDPOR bit in CR4 (CR4[2]) is set to '1'.

The WIP bit (SR1[0]) cannot be used to poll the device readiness after the POR event or hardware reset events because device is still not accessible for command until the  $t_{PU}$  time is over. However, if the WIP status remains HIGH even after  $t_{PU}$  time, this indicates device didn't boot up correctly (boot error). Once the boot error occurs, the device enters the following default state:

- The interface mode is set to Single SPI (SDR), irrespective of DPI, QPI bits status in CR2
- Register latency is set to three clock cycle (max value)
- Output impedance is set to 45- $\Omega$

- RDSR1 and RDAR commands are allowed (in SPI SDR mode only) to read the SR1. All other commands will remain disabled and will return undefined data if executed
- Reading SR1 returns 0x61 as boot error signature

The device will require a power cycle to restart the boot up sequence.

## 5.2 Device Initialization

1. QSPI F-RAMs are shipped from factory in default SPI mode.
2. To operate the QSPI F-RAM either in DPI or QPI mode, the host controller needs to configure the QSPI F-RAM to the desired DPI or QSPI interface by updating the DPI or QPI bit in CR2. Table 3 lists the details on setting the SPI, DPI, or QPI interface. CR2 is a nonvolatile register; therefore, the interface mode setting doesn't change after a power cycle or hardware/software reset. To change the interface mode, the host controller needs to overwrite the previous setting with a new interface setting.
3. The SPI, DPI, or QPI mode can be enabled via writing to CR2 either using the WRR command or WRAR command. There is no dedicated opcode to enable any specific interface mode or reset to a default SPI, which is offered as a feature in some QSPI interface devices where the interface configuration is set into volatile space that is not retained during the power cycle.
4. There is no special setting for the extended SPI dual or quad modes. Once the device receives a specific command in single-channel SPI, it determines whether to switch to either dual data, dual I/O; or quad data, quad I/O modes. To access the extended SPI quad mode, the QUAD bit must be set to '1' through CR1[1].
5. The QSPI F-RAM supports impedance selection through CR4 [7:5] on output driver. The device is shipped with 30-Ω default setting. The impedance value can be adjusted to achieve better signal integrity on application boards.
6. The SPI host controller should always start either in SPI Mode 0 (POL = '0' and PHA = '0') or SPI Mode 3 (POL = '1' and PHA = '1') as shown in Figure 25. SPI Mode 1 and SPI Mode 2 are not supported in QSPI F-RAM.

Figure 25. SPI Mode 0 and Mode 3 Timing

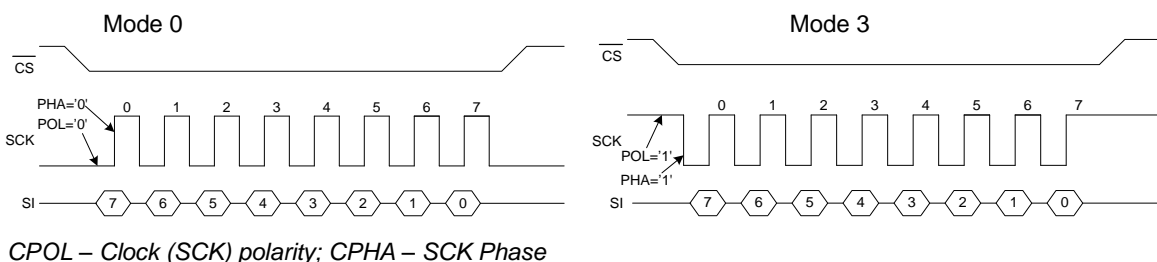


Table 3. SPI Interface Mode Setting

QUAD <sup>[Note 2]</sup>	CR1[1]	DPI CR2[4]	QPI CR2[6]	Operational Mode
0		0	0	SPI, Extended SPI (DUAL)
1		0	0	SPI, Extended SPI (DUAL/QUAD)
X		1	0	DPI
X		0	1	QPI
0		1	1	SPI <sup>[Note 3]</sup> , Extended SPI (DUAL) – Not a recommended SPI configuration
1		1	1	SPI <sup>[Note 3]</sup> , Extended SPI (DUAL/QUAD) – Not a recommended SPI configuration

**Note:** QUAD = '1' reconfigures I/O to QUAD mode and affects WP and  $\overline{\text{RESET}}$ /IO3 operations.

**Note:** Register reading will always return what is written to it



## 6 Hardware/Software Reset

### 6.1 Hardware Reset ( $\overline{\text{RESET}}$ )

$\overline{\text{RESET}}$  is multiplexed on I/O3 pin and the respective functionality on the I/O3 pin is enabled or disabled depending on the IO3Reset bit and interface mode bit setting as shown in Table 4. The  $\overline{\text{RESET}}$  input, when enabled, has an internal weak pull-up resistor hence allowed to be left floating if not used. This pull-up resistor gets disabled when the pin is configured as I/O3. The  $\overline{\text{RESET}}$  signal should never be tied LOW even if  $\overline{\text{RESET}}$  functionality is disabled because it will increase the leakage current due to the internal pull-up. QUAD bit CR1 [1] = 0 enables the hardware reset feature on the multiplexed (I/O3)/  $\overline{\text{RESET}}$  pin.

The  $\overline{\text{RESET}}$  pin is multiplexed on I/O3 in QPI mode. When using the hardware ( $\overline{\text{RESET}}$ ) in QPI mode, the CR2 [5] bit must be set to '1' to enable I/O3 as the  $\overline{\text{RESET}}$  input when CS is HIGH. Figure 26 and Figure 27 show the  $\overline{\text{RESET}}$  timing in different SPI modes. Table 5 shows the status of the registers after a hardware reset. Memory array status doesn't change after Hardware reset.

In a shared bus configuration, if the  $\overline{\text{RESET}}$  function is enabled, the device will reset every time the IO3/  $\overline{\text{RESET}}$  pin toggles LOW ( $\overline{\text{CS}}$  for the device is HIGH). Therefore, the  $\overline{\text{RESET}}$  functionality must be disabled in a shared bus configuration in QPI or extended SPI quad modes.

Table 4. IO3/ $\overline{\text{RESET}}$  Function

Interface Mode	IO3 / $\overline{\text{RESET}}$ Function		
	IO3 / Reset = 0 (IO3Reset Disable)	IO3 / Reset = 1 (IO3Reset Enable)	IO3 / Reset = 1 (IO3Reset Enable)
	$\overline{\text{CS}}$ = LOW/HIGH	$\overline{\text{CS}}$ = LOW	$\overline{\text{CS}}$ = HIGH
SPI (QUAD=0)	No function	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
DPI(QUAD=0)	No function	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
DPI(QUAD=1)	NA	IO3	$\overline{\text{RESET}}$
SPI (QUAD=1)	IO3	IO3	$\overline{\text{RESET}}$
QPI	IO3	IO3	$\overline{\text{RESET}}$

Figure 26.  $\overline{\text{RESET}}$  Timing in SPI (QUAD bit = 1) and QPI Mode

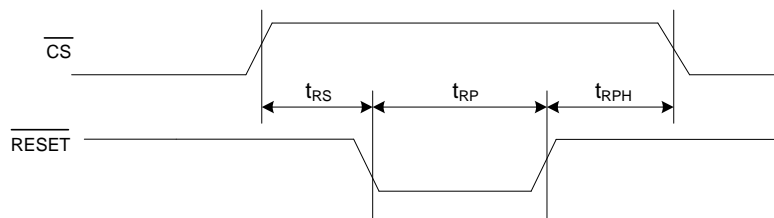
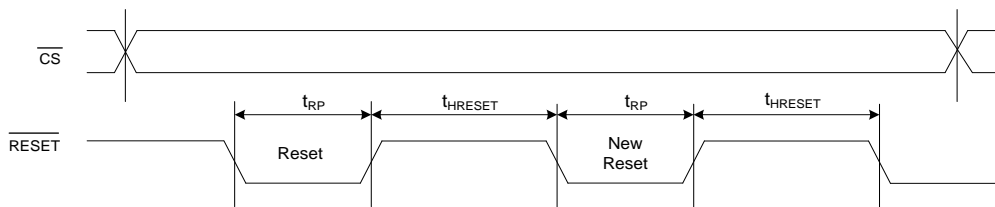


Figure 27.  $\overline{\text{RESET}}$  Timing in SPI (with QUAD bit = 0) and DPI Mode



## 6.2 Default Recovery (JEDEC SPI Reset)

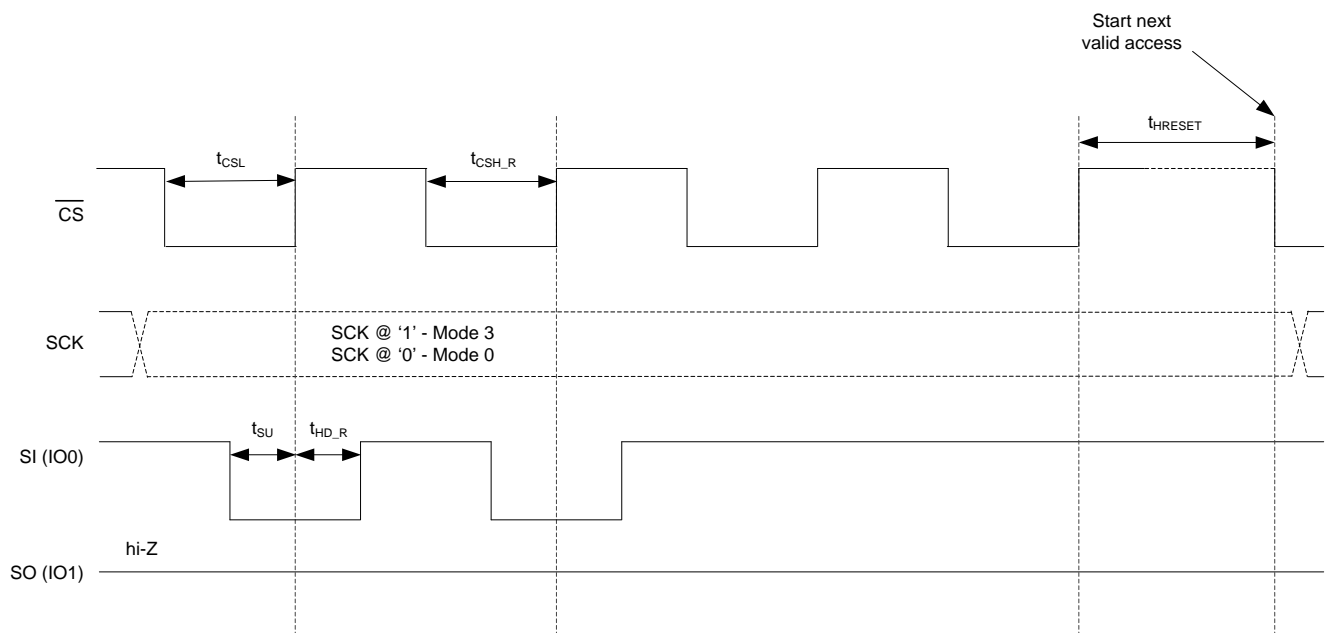
Default Recovery is a JEDEC signaling protocol that initiates a hardware reset independent of the device's operating I/O mode. It brings the device to its default mode per the setting of status and configuration registers. [Table 5](#) shows the status of the registers after the default recovery is initiated. Memory array status doesn't change after JEDEC reset.

The default recovery steps are as follows:

1.  $\overline{CS}$  toggles LOW to select the SPI slave.
2. SCK remains stable either in a HIGH or LOW state.
3. SI (I/O0) toggles HIGH to LOW, simultaneously with  $\overline{CS}$  going LOW. Other I/Os (I/O1, I/O2, and I/O3) remain don't care.
4.  $\overline{CS}$  is driven HIGH while I/O0 remain LOW.
5. Repeat the above steps 1 to 4 each time alternating the state of SI (I/O0) at the falling edge of  $\overline{CS}$  for a total of four times.
6. Reset occurs after the 4<sup>th</sup>  $\overline{CS}$  goes HIGH (inactive).

[Figure 28](#) shows the timing details. For timing parameter values, see the device datasheet.

Figure 28. Default Recovery (JEDEC SPI Reset Timing)



## 6.3 Software Reset

The software reset instruction resets the QSPI F-RAM status bits and all suspended operations. All nonvolatile registers setting remain unchanged. [Table 5](#) lists the QSPI F-RAM register status after a software reset. Software reset requires two opcode cycles - RSTEN (66h) first followed by RST (99h) to initiate the software reset cycle. It takes  $t_{RESET}$  to complete the software reset cycle.

The RSTEN command must always precede the RST command to initiate software reset. Any command, other than the RST command, following the RSTEN command will clear the reset enable condition. The next software reset instruction must start again with the RSTEN command. [Figure 29](#) shows the software reset timing. [Table 5](#) lists registers status after software reset. Memory array status doesn't change after Software reset.

Figure 29. Software Reset Timing Diagram

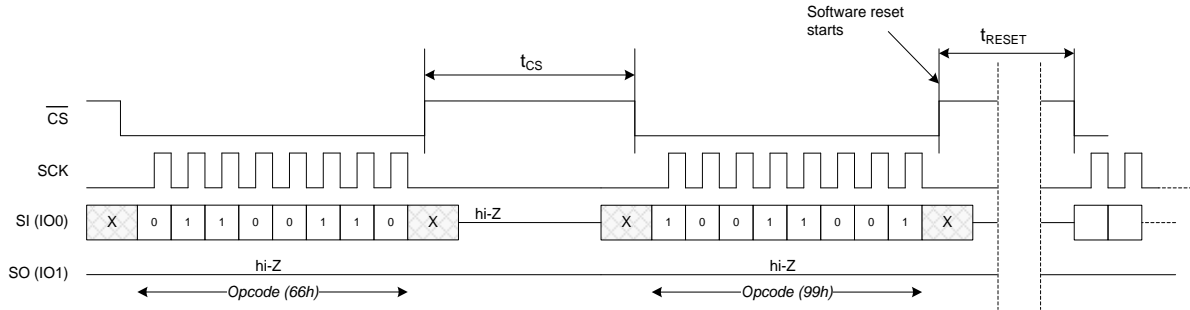


Figure 30. QSPI F-RAM Reset Flowchart (JEDEC SPI)

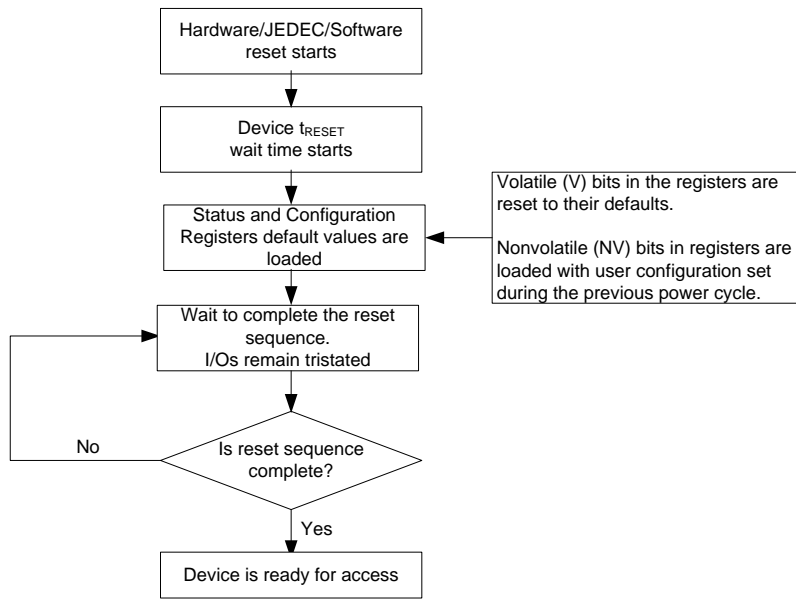


Table 5. QSPI F-RAM Register Status After Reset

Reset Function	I/O Requirements	Status Registers	Configuration Register	ECC Status	CRC Reg	Bus CRC (BCRC)	ECC Count Reg (ECCDC)	ADDR Trap Reg (ADDTRAP)	I/O Modes
<b>Power On Reset</b>	$\overline{CS} = 1'b1$ Other Inputs –Ignored All outputs – Tristated	SR1- No change SR2- 0x00	CR1, CR2, CR4, CR5 Load default values	Load – 0x00	Load – 0x00	Load – 0xFF	Load – 0x00	Load – 0x00	No change
<b>Hardware Reset</b>	$\overline{CS} = 1'b1$ Other Inputs –Ignored All outputs – Tristated	SR1- No change SR2- 0x00	CR1, CR2, CR4, CR5 Load default values	Load – 0x00	Load – 0x00	Load – 0xFF	Load – 0x00	Load – 0x00	No change
<b>Software Reset</b>	Command (RSTEN, RST)	SR1- No change SR2- 0x00	CR1, CR2, CR4, CR5 Load default values	Load – 0x00	Load – 0x00	Load – 0xFF	Load – 0x00	Load – 0x00	No change
<b>JEDEC Reset (Default Recovery)</b>	$\overline{CS}$ and SI (IO0) = Toggle Other Inputs –Ignored All outputs – Tristated	SR1- No change SR2- 0x00	CR1, CR2, CR4, CR5 Load default values	Load – 0x00	Load – 0x00	Load – 0xFF	Load – 0x00	Load – 0x00	No change

## Document History

Document Title: AN218375 - Designing with Cypress Quad SPI (QSPI) F-RAM™

Document Number: 002-18375

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5847526	ZSK	08/07/2017	Initial release

## Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

### Products

ARM® Cortex® Microcontrollers	<a href="http://cypress.com/arm">cypress.com/arm</a>
Automotive	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
Interface	<a href="http://cypress.com/interface">cypress.com/interface</a>
Internet of Things	<a href="http://cypress.com/iot">cypress.com/iot</a>
Memory	<a href="http://cypress.com/memory">cypress.com/memory</a>
Microcontrollers	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
Power Management ICs	<a href="http://cypress.com/pmic">cypress.com/pmic</a>
Touch Sensing	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB Controllers	<a href="http://cypress.com/usb">cypress.com/usb</a>
Wireless Connectivity	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

### PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

### Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

### Technical Support

[cypress.com/support](http://cypress.com/support)

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.