

---

---

## Section 46. Serial Quad Interface (SQI)

---

---

### HIGHLIGHTS

This section of the manual contains the following major topics:

46.1	Introduction .....	46-2
46.2	SQI Module Features.....	46-2
46.3	Functional Block Description.....	46-2
46.4	Control Registers .....	46-4
46.5	SQI Operation .....	46-33
46.6	SQI Transfer Modes.....	46-35
46.7	SQI Data Flow Modes.....	46-49
46.8	Flash Instructions and Sequence Diagrams Quick Reference .....	46-50
46.9	Effects Of Reset.....	46-54
46.10	Operation in Power-Saving Modes .....	46-54
46.11	Related Application Notes.....	46-55
46.12	Revision History .....	46-56

**Note:** This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all PIC32 devices.

Please consult the note at the beginning of the “**Serial Quad Interface (SQI)**” chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: <http://www.microchip.com>

## 46.1 INTRODUCTION

The SQI module is a synchronous serial interface that provides access to serial Flash memories and other serial devices. The SQI module supports Single Lane (identical to SPI), Dual Lane, and Quad Lane interface modes.

## 46.2 SQI MODULE FEATURES

The Serial Quad Interface (SQI) module offers the following key features:

- Supports Single, Dual, and Quad Lane modes
- Supports Single Data Rate (SDR) and Double Data Rate (DDR) modes
- Programmable command sequence
- Data transfer modes:
  - DMA mode
  - Programmed I/O (PIO) mode
- eXecute-In-Place (XIP)<sup>(1)</sup>
- Supports SPI Mode 0 and Mode 3
- Programmable Clock Polarity (CPOL) and Clock Phase (CPHA) bits
- Supports automatic memory status check in DMA and PIO modes
- Supports up to two Chip Selects
- Supports up to four bytes of Flash address
- Programmable interrupt thresholds
- 32-byte transmit data buffer
- 32-byte receive data buffer
- 4-word controller buffer

## 46.3 FUNCTIONAL BLOCK DESCRIPTION

The SQI module, which is an industry standard synchronous serial link, helps communicate with multiple SPI compatible devices such as serial EEPROMs and serial Flash devices.

The SQI module has three interfaces, one external to the device (SQI Bus Interface) that connects to the external Flash memories or other serial devices, and two internal (Bus Slave interface for control register reads/writes and Bus Master for data transfers), as illustrated in [Figure 46-1](#).

The SQI bus interface consists of four data lines (SQID3-SQID0), a clock line (SQICLK), and two select lines (SQICS0 and SQICS1). As mentioned earlier, the SQI module supports Single Lane (SPI mode), Dual Lane, and Quad Lane modes of operation.

The SQI module operates in both Single Data Rate (SDR) and Double Data Rate (DDR) modes. In DDR mode, the data transition occurs on both edges of the clock providing double the throughput.

**Note:** The SQI module is a half-duplex, synchronous serial interface when in Master mode of operation.

1. This feature is not available on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine availability.

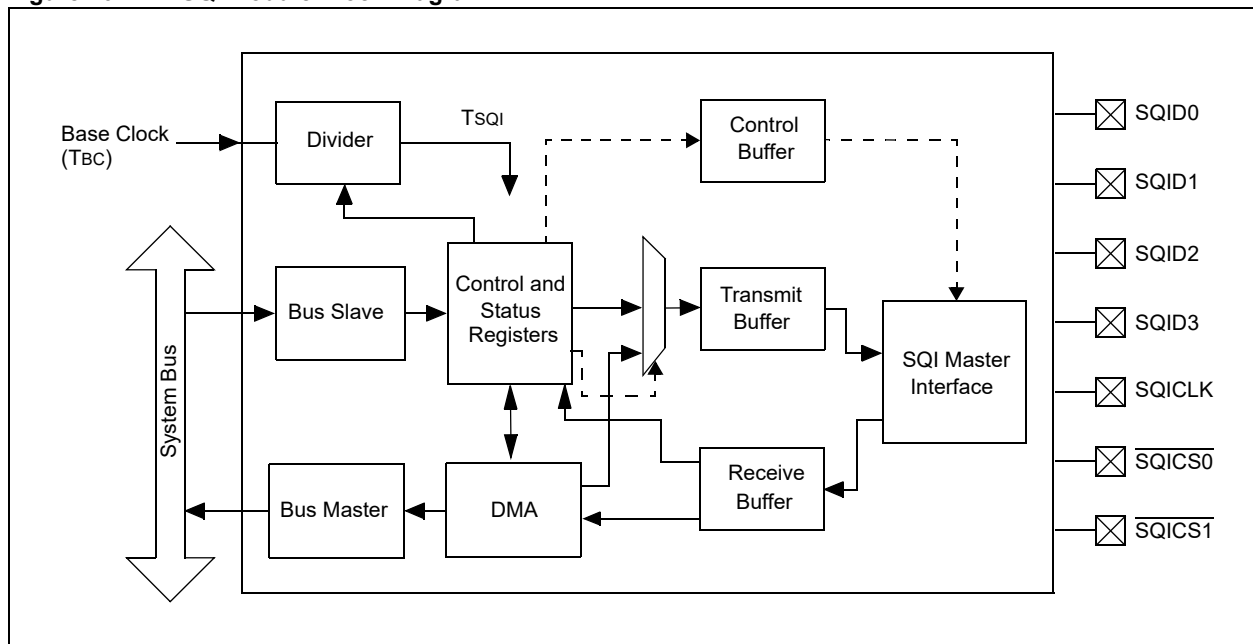
The SQI module has configurable transmit and receive buffers, programmable baud rates through the internal clock divider, clock phase, and clock polarity control for efficient data operations. Transmit and receive buffers can be accessed through SQI1TXDATA and SQI1RXDATA registers. Similarly, the control buffer can be accessed through the SQI1CON register and is mainly used to pipeline the operations. The SQI module operates in three transfer modes: DMA, PIO, and XIP. All three modes use the control buffer to pipeline the command/data sequences on the SQI bus.

The SQI module supports two data flow modes: SPI Mode 0 and Mode 3. Each transfer mode (XIP/PIO/DMA) can use any of the data flow modes as desired by the application.

DMA and PIO modes are typically used to transfer the data to and from external serial Flash memory, whereas, eXecute In Place (XIP)<sup>(1)</sup> mode is used to execute the code out of the external serial Flash memory. DMA mode uses the internal DMA engine and buffer descriptors to transfer data between source and destination memory spaces off-loading the Host processor during which time, accessing SQI1TXDATA, SQI1RXDATA, and SQI1CON functionally will not yield expected results. However, PIO mode engages the Host processor to access the contents of the external serial Flash memory using a bit-band method through the transmit and receive data registers. Refer to section 46.6 “SQI Transfer Modes” for a detailed description of each transfer mode.

The SQI module supports automatic memory status check reducing software burden. Figure 46-1 shows a block diagram of the SQI module.

**Figure 46-1: SQI Module Block Diagram**



1. This feature is not available on all devices. Refer to the “SQI” chapter of the specific device data sheet to determine availability.

## 46.4 CONTROL REGISTERS

The SQI module for PIC32 devices contains the following Special Function Registers (SFRs):

- **SQI1XCON1: SQI XIP Control Register 1**<sup>(1)</sup>

This register is used to control the SQI operation in XIP mode and the user application should program this register prior to entering XIP mode.
- **SQI1XCON2: SQI XIP Control Register 2**<sup>(1)</sup>

This register is used to control the SQI operation in XIP mode and the user application should program this register prior to entering XIP mode.
- **SQI1CFG: SQI Configuration Registers**

This register can be read at any time and must be written only when the SQI port is not actively transmitting and receiving data. This register is used to configure the SQI module in all modes of operation.
- **SQI1CON: SQI Control Register**

This register can be read or written at any time and is used to configure the SQI module. The SQI control buffer can be accessed using this register in the PIO mode of operation. Up to four commands can be stacked inside the SQI control buffer.
- **SQI1CLKCON: SQI Clock Control Register**

This read/write register is used to generate SQI clock from the SQI base clock. It can be read and written at any time. The behavior of the clock is not deterministic if changed in the middle of a transfer. This register should be configured with desired clock rate before entering any mode of operation.
- **SQI1CMDTHR: SQI Command Threshold Register**

This register is used to program the buffer depth thresholds before the designated transmit or receive command is executed by the SQI module. This register is mostly used when the SQI module is in PIO mode of operation.
- **SQI1INTTHR: SQI Interrupt Threshold Register**

This register is used to program the interrupt and buffer depth thresholds. If interrupt enable is not set for the corresponding bits, the interrupt signal will not be asserted; however, the status signal reflects the value and it can be polled for appropriate action.
- **SQI1INTEN: SQI Interrupt Enable Register**

This register is used to set the mask for interrupt generation. The status values will be set regardless of the interrupt mask value.
- **SQI1INTSTAT: SQI Interrupt Status Register**

This register provides interrupt status information. Bits that provide information about the SQI module are read-only; these bits are set and cleared by the hardware. Read/write bits are set by hardware when an interrupt condition occurs and must be cleared by software.
- **SQI1TXDATA: SQI Transmit Data Buffer Register**

This register is used to write data to the transmit buffer and to issue commands to the device that is attached to the SQI.
- **SQI1RXDATA: SQI Receive Data Buffer Register**

This register contains the read contents from the external serial devices.
- **SQI1STAT1: SQI Status Register 1**

This register provides the transmit/receive buffer levels.
- **SQI1STAT2: SQI Status Register 2**

This register is used to monitor buffer overflow/underflow events and for data bus debugging.
- **SQI1BDCON: SQI Buffer Descriptor Control Register**

This register controls the Buffer Descriptor and is used in the DMA mode of operation.

---

1. This feature is not available on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine availability.

- **SQI1BDCURADD: SQI Buffer Descriptor Current Address Register**  
This register provides the current descriptor address being processed by the DMA.
- **SQI1BDBASEADD: SQI Buffer Descriptor Base Address Register**  
This register contains the address of the first buffer descriptor. This register must be updated only when the Buffer Descriptor DMA is idle.
- **SQI1BDSTAT: SQI Buffer Descriptor Status Register**  
This register contains the current descriptor control word and provides DMA status information.
- **SQI1BDPOLLCON: SQI Buffer Descriptor Poll Control Register**  
This register determines the number of cycles the DMA would wait before refetching the descriptor control word if the previous descriptor fetched was disabled.
- **SQI1BDTXDSTAT: SQI Buffer Descriptor DMA Transmit Status Register**  
This register provides the Buffer Descriptor DMA transmit status.
- **SQI1BDRXDSTAT: SQI Buffer Descriptor DMA Receive Status Register**  
This register provides the Buffer Descriptor DMA receive status.
- **SQI1THR: SQI Threshold Control Register**  
This register is used to program the threshold value for the SQI control buffer.
- **SQI1INTSIGEN: SQI Interrupt Signal Enable Register**  
This register acts as a second level gate for the interrupts. Interrupt bits in both the SQI1INTEN and SQI1INTSEN registers should be set simultaneously to trigger an interrupt.
- **SQI1TAPCON: SQI TAP Control Register**  
This register provides clock to data timing control at higher interface speeds.
- **SQI1MEMSTAT: SQI Memory Status Control Register**  
This register control the Flash memory status check command and RDY/BUSY bit positions.
- **SQI1XCON3: SQI XIP Control Register 3<sup>(1)</sup> and SQI1XCON4: SQI XIP Control Register 4<sup>(1)</sup>**  
These optional XIP mode registers provides additional command queuing. Up to three commands with status check options are supported.

Table 46-1 provides a brief summary of the related SQI module registers. Corresponding registers appear after the summary, followed by a detailed description of each bit.

1. This register is not available on all devices. Refer to the “SQI” chapter in the specific device data sheet to determine availability.

Table 46-1: Serial Quadrature Interface (SQI) Register Map

Register Name	Bit Range	Bits																
		31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
SQI1 XCON1 <sup>(1)</sup>	31:16	—	—	SDRCMD	DDRDATA	DDR DUMMY	DDR MODE	DDR ADDR	DDRCMD	DUMMYBYTES[2:0]			ADDRBYTES[2:0]			READOPCODE[7:6]		
	15:0	READOPCODE[5:0]					TYPEDATA[1:0]			TYPEDUMMY[1:0]		TYPEMODE[1:0]		TYPEADDR[1:0]		TYPECMD[1:0]		
SQI1 XCON2 <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	DEVSEL[1:0]			MODEBYTES[1:0]			MODECODE[7:0]										
SQI1CFG	31:16	—	—	—	—	—	—	CSEN[1:0]		SQIEN	—	DATAEN[1:0]		CONBUFRST	RXBUFRST	TXBUFRST	RESET	
	15:0	—	—	—	BURSTEN	—	HOLD	WP	—	—	—	LSBF	CPOL	CPHA	MODE[2:0]			
SQI1CON	31:16	—	—	—	—	—	—	—	SCHECK	DDRMODE	DASSERT	DEVSEL[1:0]		LANEMODE[1:0]		CMDINIT[1:0]		
	15:0	TXRXCOUNT[15:0]																
SQI1 CLKCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKDIV[10:8]		
	15:0	CLKDIV[7:0]																
SQI1 CMDTHR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	TXCMDTHR[5:0]					RXCMDTHR[5:0]											
SQI1 INTTHR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	TXINTTHR[5:0]					RXINTTHR[5:0]											
SQI1 INTEN	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	—	—	—	—	DMAEIE	PKT COMPIE	BD DONEIE	CON THRIE	CON EMPTYIE	CON FULLIE	RX THRIE	RX FULLIE	RX EMPTYIE	TX THRIE	TX FULLIE	TX EMPTYIE	
SQI1 INTSTAT	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	—	—	—	—	DMAEIF	PKT COMPIF	BD DONEIF	CON THRIF	CON EMPTYIF	CON FULLIF	RX THRIF	RX FULLIF	RX EMPTYIF	TX THRIF	TX FULLIF	TX EMPTYIF	
SQI1 TXDATA	31:16	TXDATA[31:16]																
	15:0	TXDATA[15:0]																
SQI1 RXDATA	31:16	RXDATA[31:16]																
	15:0	RXDATA[15:0]																
SQI1 STAT1	31:16	—	—	—	—	—	—	—	—	—	—	TXBUFFFREE[5:0]						
	15:0	—	—	—	—	—	—	—	—	—	—	RXBUFCNT[5:0]						
SQI1 STAT2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	CMDSTAT[1:0]			
	15:0	CONAVAIL[3:0]					SDID3	SDID2	SDID1	SDID0	—	—	RXUN	TXOV				
SQI1 BDCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	START	POLLEN	DMAEN	
SQI1BD CURADD	31:16	BDCURRADDR[31:16]																
	15:0	BDCURRADDR[15:0]																
SQI1BD BASEADD	31:16	BDADDR[31:16]																
	15:0	BDADDR[15:0]																
SQI1BD STAT	31:16	—	—	—	—	—	—	—	—	—	—	BDSTATE[3:0]			DMA START	DMAACTV		
	15:0	BDCON[15:0]																
SQI1BD POLLCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	15:0	POLLCON[15:0]																

**Note 1:** This register is not available on all devices. Refer to the “SQI” chapter in the specific device data sheet to determine availability.

**Table 46-1: Serial Quadrature Interface (SQI) Register Map (Continued)**

Register Name	Bit Range	Bits															
		31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
SQI1BD TXDSTAT	31:16	—	—	—	TXSTATE[3:0]			—	—	—	TXBUFCNT[5:0]						
	15:0	—	—	—	—	—	—	TXCURBUFLN[8:0]									
SQI1BD RXDSTAT	31:16	—	—	—	RXSTATE[3:0]			—	—	—	RXBUFCNT[5:0]						
	15:0	—	—	—	—	—	—	RXCURBUFLN[8:0]									
SQI1THR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	THRES[3:0]				
SQI1INT SIGEN	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	DMAEISE	PKT COMPISE	BD DONEISE	CON THRISE	CON EMPTYISE	CON FULLISE	RX THRISE	RX FULLISE	RX EMPTYISE	TX THRISE	TX FULLISE	TX EMPTYISE
SQI1 TAPCON	31:16	—	—	DDRCLKINDLY[5:0]				SDRDATINDLY[3:0]				DDRDATINDLY[3:0]					
	15:0	—	—	SDRCLKINDLY[5:0]				DATAOUTDLY[3:0]				CLKOUTDLY[3:0]					
SQI1 MEMSTAT	31:16	—	—	—	—	—	—	—	—	—	—	—	STATPOS	TYPESTAT[1:0]		STATBYTES[1:0]	
	15:0	STATCMD[15:0]															
SQI1 XCON3 <sup>(1)</sup>	31:16	—	—	—	INIT1 SCHECK	INIT1COUNT[1:0]		INIT1TYPE[1:0]		INIT1CMD3[7:0]							
	15:0	INIT1CMD2[7:0]								INIT1CMD1[7:0]							
SQI1 XCON4 <sup>(1)</sup>	31:16	—	—	—	INIT2 SCHECK	INIT2COUNT[1:0]		INIT2TYPE[1:0]		INIT2CMD3[7:0]							
	15:0	INIT2CMD2[7:0]								INIT2CMD1[7:0]							

**Note 1:** This register is not available on all devices. Refer to the “SQI” chapter in the specific device data sheet to determine availability.

# PIC32 Family Reference Manual

**Register 46-1: SQI1XCON1: SQI XIP Control Register 1**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	SDRCMD <sup>(3)</sup>	DDR DATA <sup>(3)</sup>	DDR DUMMY <sup>(3)</sup>	DDR MODE <sup>(3)</sup>	DDR ADDR <sup>(3)</sup>	DDR CMD <sup>(2,3)</sup>
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DUMMYBYTES[2:0]			ADDRBYTES[2:0]			READOPCODE[7:6]	
15:8							R/W-0	R/W-0
	READOPCODE[5:0]						TYPEDATA[1:0]	
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TYPEDUMMY[1:0]		TYPEMODE[1:0]		TYPEADDR[1:0]		TYPECMD[1:0]	

**Legend:**

R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'  
-n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

bit 31-30 **Unimplemented:** Read as '0'

bit 29 **SDRCMD:** SQI Command in SDR Mode bit<sup>(3)</sup>  
1 = SQI command is in SDR mode and SQI data is in DDR mode  
0 = SQI command is in DDR mode and SQI data is in SDR mode

bit 28 **DDRDATA:** SQI Data DDR Mode bit<sup>(3)</sup>  
1 = SQI data bytes are transferred in DDR mode  
0 = SQI data bytes are transferred in SDR mode

bit 27 **DDR DUMMY:** SQI Dummy DDR Mode bit<sup>(3)</sup>  
1 = SQI dummy bytes are transferred in DDR mode  
0 = SQI dummy bytes are transferred in SDR mode

bit 26 **DDRMODE:** SQI DDR Mode bit<sup>(3)</sup>  
1 = SQI mode bytes are transferred in DDR mode  
0 = SQI mode bytes are transferred in SDR mode

bit 25 **DDRADDR:** SQI Address Mode bit<sup>(3)</sup>  
1 = SQI address bytes are transferred in DDR mode  
0 = SQI address bytes are transferred in SDR mode

bit 24 **DDRCMD:** SQI DDR Command Mode bit<sup>(2,3)</sup>  
1 = SQI command bytes are transferred in DDR mode  
0 = SQI command bytes are transferred in SDR mode

bit 23-21 **DUMMYBYTES[2:0]:** Transmit Dummy Bytes bits  
111 = Transmit seven dummy bytes after the address bytes  
.  
.  
.  
011 = Transmit three dummy bytes after the address bytes  
010 = Transmit two dummy bytes after the address bytes  
001 = Transmit one dummy bytes after the address bytes  
000 = Transmit zero dummy bytes after the address bytes

- Note 1:** This register is not available on all devices. Refer to the “SQI” chapter of the specific device data sheet to determine availability.
- 2:** When DDRCMD is set to '0', the SQI module will ignore the value in the SDRCMD bit.
- 3:** This bit is not available on all devices. Refer to the “SQI” chapter of the specific device data sheet to determine availability.

**Register 46-1: SQI1XCON1: SQI XIP Control Register 1 (Continued)**bit 20-18 **ADDRBYTES[2:0]**: Address Cycle bits

111 = Reserved

.  
.  
.

101 = Reserved

100 = Four address bytes

011 = Three address bytes

010 = Two address bytes

001 = One address bytes

000 = Zero address bytes

bit 17-10 **READOPCODE[7:0]**: Op code Value for Read Operation bits

These bits contain the 8-bit op code value for read operation.

bit 9-8 **TYPEDATA[1:0]**: SQI Type Data Enable bits

The boot controller will receive the data in Single Lane, Dual Lane, or Quad Lane.

11 = Reserved

10 = Quad Lane mode data is enabled

01 = Dual Lane mode data is enabled

00 = Single Lane mode data is enabled

bit 7-6 **TYPEDUMMY[1:0]**: SQI Type Dummy Enable bits

The boot controller will send the dummy in Single Lane, Dual Lane, or Quad Lane.

11 = Reserved

10 = Quad Lane mode dummy is enabled

01 = Dual Lane mode dummy is enabled

00 = Single Lane mode dummy is enabled

bit 5-4 **TYPEMODE[1:0]**: SQI Type Mode Enable bits

The boot controller will send the mode in Single Lane, Dual Lane, or Quad Lane.

11 = Reserved

10 = Quad Lane mode is enabled

01 = Dual Lane mode is enabled

00 = Single Lane mode is enabled

bit 3-2 **TYPEADDR[1:0]**: SQI Type Address Enable bits

The boot controller will send the address in Single Lane, Dual Lane, or Quad Lane.

11 = Reserved

10 = Quad Lane mode address is enabled

01 = Dual Lane mode address is enabled

00 = Single Lane mode address is enabled

bit 1-0 **TYPECMD[1:0]**: SQI Type Command Enable bits

The boot controller will send the command in Single Lane, Dual Lane, or Quad Lane.

11 = Reserved

10 = Quad Lane mode command is enabled

01 = Dual Lane mode command is enabled

00 = Single Lane mode command is enabled

**Note 1:** This register is not available on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine availability.

**2:** When DDRCMD is set to ‘0’, the SQI module will ignore the value in the SDRCMD bit.

**3:** This bit is not available on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine availability.

# PIC32 Family Reference Manual

**Register 46-2: SQ1XCON2: SQI XIP Control Register 2**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	—	DEVSEL[1:0]		MODEBYTES[1:0]	
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	MODECODE[7:0]							

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

bit 31-12 **Unimplemented:** Read as '0'

bit 11-10 **DEVSEL[1:0]:** Device Select bits

- 11 = Reserved
- 10 = Reserved
- 01 = Device 1 is selected
- 00 = Device 0 is selected

bit 9-8 **MODEBYTES[1:0]:** Mode Byte Cycle Enable bits

- 11 = Three cycles
- 10 = Two cycles
- 01 = One cycle
- 00 = Zero cycles

bit 7-0 **MODECODE[7:0]:** Mode Code Value bits

These bits contain the 8-bit code value for the mode bits.

**Note 1:** This register is not available on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine availability.

**Register 46-3: SQI1CFG: SQI Configuration Registers**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
	—	—	—	—	—	—	CSEN[1:0]	
23:16	R/W-0	U-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
	SQIEN	—	DATAEN[1:0]		CONBUFRST <sup>(2)</sup>	RXBUFRST <sup>(2)</sup>	TXBUFRST <sup>(2)</sup>	RESET
15:8	U-0	r-0	r-0	R/W-0	r-0	R/W-0	R/W-0	U-0
	—	—	—	BURSTEN <sup>(1)</sup>	—	HOLD	WP	—
7:0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	LSBF	CPOL	CPHA	MODE[2:0]		

<b>Legend:</b>	HC = Hardware Cleared	r = Reserved
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

- bit 31-26 **Unimplemented:** Read as '0'
- bit 25-24 **CSEN[1:0]:** Chip Select Output Enable bits
  - 11 = Chip Select 0 and Chip Select 1 are used
  - 10 = Chip Select 1 is used (Chip Select 0 is not used)
  - 01 = Chip Select 0 is used (Chip Select 1 is not used)
  - 00 = Chip Select 0 and Chip Select 1 are not used
- bit 23 **SQIEN:** SQI Enable bit
  - 1 = SQI module is enabled
  - 0 = SQI module is disabled
- bit 22 **Unimplemented:** Read as '0'
- bit 21-20 **DATAEN[1:0]:** Data Output Enable bits
  - 11 = Reserved
  - 10 = SQID3-SQID0 outputs are enabled
  - 01 = SQID1 and SQID0 data outputs are enabled
  - 00 = SQID0 data output is enabled
- bit 19 **CONBUFRST:** Control Buffer Reset bit<sup>(2)</sup>
  - 1 = A reset pulse is generated clearing the control buffer
  - 0 = A reset pulse is not generated
- bit 18 **RXBUFRST:** Receive Buffer Reset bit<sup>(2)</sup>
  - 1 = A reset pulse is generated clearing the receive buffer
  - 0 = A reset pulse is not generated
- bit 17 **TXBUFRST:** Transmit Buffer Reset bit<sup>(2)</sup>
  - 1 = A reset pulse is generated clearing the transmit buffer
  - 0 = A reset pulse is not generated
- bit 16 **RESET:** Software Reset Select bit
 

This bit is automatically cleared by the SQI module. All of the internal state machines and buffer pointers are reset by this reset pulse.

  - 1 = A reset pulse is generated
  - 0 = A reset pulse is not generated
- bit 15 **Unimplemented:** Read as '0'
- bit 14-13 **Reserved:** Must be programmed as '0'

**Note 1:** This bit must be programmed as '1'.  
**Note 2:** This bit is not available on all devices. Refer to the “SQI” chapter of the specific device data sheet to determine availability.

# PIC32 Family Reference Manual

---

## Register 46-3: SQ1CFG: SQI Configuration Registers (Continued)

- bit 12 **BURSTEN:** Burst Configuration bit<sup>(1)</sup>  
1 = Burst is enabled  
0 = Burst is not enabled
- bit 11 **Reserved:** Must be programmed as '0'
- bit 10 **HOLD:** Hold bit  
In Single Lane or Dual Lane mode, this bit is used to drive the SQID3 pin, which can be used for devices with a HOLD input pin. The meaning of the values for this bit will depend on the device to which SQID3 is connected.
- bit 9 **WP:** Write Protect bit  
In Single Lane or Dual Lane mode, this bit is used to drive the SQID2 pin, which can be used with devices with a write-protect pin. The meaning of the values for this bit will depend on the device to which SQID2 is connected.
- bit 8-6 **Unimplemented:** Read as '0'
- bit 5 **LSBF:** Data Format Select bit  
1 = LSB is sent or received first  
0 = MSB is sent or received first
- bit 4 **CPOL:** Clock Polarity Select bit  
1 = Active-low SQICLK (SQICLK high is the Idle state)  
0 = Active-high SQICLK (SQICLK low is the Idle state)
- bit 3 **CPHA:** Clock Phase Select bit  
1 = SQICLK starts toggling at the start of the first data bit  
0 = SQICLK starts toggling at the middle of the first data bit
- bit 2-0 **MODE[2:0]:** Mode Select bits  
111 = Reserved  
•  
•  
•  
100 = Reserved  
011 = XIP mode is selected (when this mode is entered, the module behaves as if executing in place (XIP), but uses the register data to control timing)  
010 = DMA mode is selected  
001 = CPU mode is selected (the module is controlled by the CPU in PIO mode. This mode is entered when leaving Boot or XIP mode)  
000 = Reserved

- Note 1:** This bit must be programmed as '1'.  
**2:** This bit is not available on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine availability.

Register 46-4: SQI1CON: SQI Control Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	r-0	R/W-0
	—	—	—	—	—	—	—	SCHECK <sup>(1,2)</sup>
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DDRMODE <sup>(1)</sup>	DASSERT	DEVSEL[1:0]		LANEMODE[1:0]		CMDINIT[1:0]	
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TXRXCOUNT[15:8]							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	TXRXCOUNT[7:0]							

<b>Legend:</b>	r = Reserved
R = Readable bit	W = Writable bit
-n = Value at POR	'1' = Bit is set
	U = Unimplemented bit, read as '0'
	'0' = Bit is cleared
	x = Bit is unknown

bit 31-26 **Unimplemented:** Read as '0'

bit 25 **Reserved:** Must be programmed as '0'

bit 24 **SCHECK:** Flash Status Check bit<sup>(1,2)</sup>

- 1 = Check the status of the Flash
- 0 = Do not check the status of the Flash

bit 23 **DDRMODE:** Double Data Rate Mode bit<sup>(1)</sup>

- 1 = Set the SQI transfers to DDR mode
- 0 = Set the SQI transfers to SDR mode

bit 22 **DASSERT:** Chip Select Assert bit

- 1 = Chip Select is deasserted after transmission or reception of the specified number of bytes
- 0 = Chip Select is not deasserted after transmission or reception of the specified number of bytes

bit 21-20 **DEVSEL[1:0]:** SQI Device Select bits

- 11 = Reserved
- 10 = Reserved
- 01 = Select Device 1
- 00 = Select Device 0

bit 19-18 **LANEMODE[1:0]:** SQI Lane Mode Select bits

- 11 = Reserved
- 10 = Quad Lane mode
- 01 = Dual Lane mode
- 00 = Single Lane mode

bit 17-16 **CMDINIT[1:0]:** Command Initiation Mode Select bits

If it is Transmit, commands are initiated based on a write to the transmit register or the contents of TX buffer. If CMDINIT is Receive, commands are initiated based on reads to the read register or RX buffer availability.

- 11 = Reserved
- 10 = Receive
- 01 = Transmit
- 00 = Idle

bit 15-0 **TXRXCOUNT[15:0]:** Transmit/Receive Count bits

These bits specify the total number of bytes to transmit or received (based on CMDINIT).

**Note 1:** When this bit is set to '1', the SQI module uses the SQI1MEMSTAT register to control the status check command process.

**Note 2:** This bit is not available on all devices. Refer to the "SQI" chapter of the specific device data sheet to determine availability.

# PIC32 Family Reference Manual

**Register 46-5: SQ1CLKCON: SQI Clock Control Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
	—	—	—	—	—	CLKDIV[10:8] <sup>(1,2)</sup>		
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CLKDIV[7:0] <sup>(1,2)</sup>							
7:0	U-0	U-0	U-0	U-0	U-0	U-0	R-0	R/W-0
	—	—	—	—	—	—	STABLE	EN

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

bit 31-19 **Unimplemented:** Read as '0'

bit 18-8 **CLKDIV[10:0]:** SQI Clock TsQI Frequency Select bit<sup>(1,2)</sup>

- 10000000000 = Base clock TBC is divided by 2048
- 01000000000 = Base clock TBC is divided by 1024
- 00100000000 = Base clock TBC is divided by 512
- 00010000000 = Base clock TBC is divided by 256
- 00001000000 = Base clock TBC is divided by 128
- 00000100000 = Base clock TBC is divided by 64
- 00000010000 = Base clock TBC is divided by 32
- 00000001000 = Base clock TBC is divided by 16
- 00000000100 = Base clock TBC is divided by 8
- 00000000010 = Base clock TBC is divided by 4
- 00000000001 = Base clock TBC is divided by 2
- 00000000000 = Base clock TBC

Setting these bits to '00000000' specifies the highest frequency of the SQI clock.

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **STABLE:** TsQI Clock Stable Select bit

This bit is set to '1' when the SQI clock, TsQI, is stable after writing a '1' to the EN bit.

- 1 = TsQI clock is stable
- 0 = TsQI clock is not stable

bit 0 **EN:** TsQI Clock Enable Select bit

When clock oscillation is stable, the SQI module will set the STABLE bit to '1'.

- 1 = Enable the SQI clock (TsQI) (when clock oscillation is stable, the SQI module sets the STABLE bit to '1')
- 0 = Disable the SQI clock (TsQI) (the SQI module should stop its clock to enter a low power state); SFRs can still be accessed, as they use PBCLK5

**Note 1:** Refer to the “**Electrical Characteristics**” chapter in the specific device data sheet for the maximum clock frequency specifications.

**2:** The clock divider values are not the same on all devices. Refer to the “**SQI**” chapter of the specific device data sheet to determine the actual values.

**Register 46-6: SQI1CMDTHR: SQI Command Threshold Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	TXCMDTHR[5:0]					
7:0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	RXCMDTHR[5:0] <sup>(1)</sup>					

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

bit 31-14 **Unimplemented:** Read as '0'

bit 13-8 **TXCMDTHR[5:0]:** Transmit Command Threshold bits

In transmit initiation mode, the SQI module performs a transmit operation when transmit command threshold bytes are present in the TX buffer. These bits should usually be set to '1' for normal Flash commands, and set to a higher value for page programming. For 16-bit mode, the value should be a multiple of 2.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RXCMDTHR[5:0]:** Receive Command Threshold bits<sup>(1)</sup>

In receive initiation mode, the SQI module attempts to perform receive operations to fetch the receive command threshold number of bytes in the receive buffer. If space for these bytes is not present in the buffer, the SQI will not initiate a transfer. For 16-bit mode, the value should be a multiple of 2.

If software performs any reads, thereby reducing the buffer count, hardware would initiate a receive transfer to make the buffer count equal to the value in these bits. If software would not like any more words latched into the buffer, command initiation mode needs to be changed to Idle before any buffer reads by software.

In the case of Boot/XIP mode, the SQI module will use the System Bus burst size, instead of the receive command threshold value.

**Note 1:** These bits should only be programmed when a receive is not active (i.e., during Idle mode or a transmit).

# PIC32 Family Reference Manual

**Register 46-7: SQI1INTTHR: SQI Interrupt Threshold Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	TXINTTHR[5:0]					
7:0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	RXINTTHR[5:0]					

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 31-14 **Unimplemented:** Read as '0'

bit 13-8 **TXINTTHR[5:0]:** Transmit Interrupt Threshold bits

A transmit interrupt is set when the transmit buffer has more space than the set number of bytes. For 16-bit mode, the value should be a multiple of 2.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RXINTTHR[5:0]:** Receive Interrupt Threshold bits

A receive interrupt is set when the receive buffer count is larger than or equal to the set number of bytes. For 16-bit mode, the value should be multiple of 2.

**Register 46-8: SQI1INTEN: SQI Interrupt Enable Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
23:16	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
15:8	U-0 —	U-0 —	U-0 —	U-0 —	R/W-0 DMAEIE	R/W-0 PKTCOMPIE	R/W-0 BDDONEIE	R/W-0 CONTHRIE
7:0	R/W-0 CONEMPTYIE	R/W-0 CONFULLIE	R/W-0 RXTHRIE	R/W-0 RXFULLIE	R/W-0 RXEMPTYIE	R/W-0 TXTHRIE	R/W-0 TXFULLIE	R/W-0 TXEMPTYIE

<b>Legend:</b>	HS = Hardware Set
R = Readable bit	W = Writable bit      U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

- bit 31-12 **Unimplemented:** Read as '0'
- bit 11 **DMAEIE:** DMA Bus Error Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 10 **PKTCOMPIE:** DMA Buffer Descriptor Packet Complete Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 9 **BDDONEIE:** DMA Buffer Descriptor Done Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 8 **CONTHRIE:** Control Buffer Threshold Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 7 **CONEMPTYIE:** Control Buffer Empty Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 6 **CONFULLIE:** Control Buffer Full Interrupt Enable bit  
This bit enables an interrupt when the receive buffer is full.  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 5 **RXTHRIE:** Receive Buffer Threshold Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 4 **RXFULLIE:** Receive Buffer Full Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 3 **RXEMPTYIE:** Receive Buffer Empty Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 2 **TXTHRIE:** Transmit Threshold Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 1 **TXFULLIE:** Transmit Buffer Full Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled
- bit 0 **TXEMPTYIE:** Transmit Buffer Empty Interrupt Enable bit  
1 = Interrupt is enabled  
0 = Interrupt is disabled

# PIC32 Family Reference Manual

**Register 46-9: SQ1INTSTAT: SQI Interrupt Status Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
	—	—	—	—	DMA EIF	PKT COMPIF	BD DONEIF	CON THRIF
7:0	R/W-1, HS	R/W-0, HS	R/W-1, HS	R/W-0, HS	R/W-1, HS	R/W-1, HS	R/W-0, HS	R/W-1, HS
	CON EMPTYIF	CON FULLIF	RXTHRIF <sup>(1)</sup>	RXFULLIF	RX EMPTYIF	TXTHRIF	TXFULLIF	TX EMPTYIF

<b>Legend:</b>	HS = Hardware Set
R = Readable bit	W = Writable bit
-n = Value at POR	U = Unimplemented bit, read as '0'
	'1' = Bit is set
	'0' = Bit is cleared
	x = Bit is unknown

bit 31-12 **Unimplemented:** Read as '0'

bit 11 **DMAEIF:** DMA Bus Error Interrupt Flag bit

- 1 = DMA bus error has occurred
- 0 = DMA bus error has not occurred

bit 10 **PKTCOMPIF:** DMA Buffer Descriptor Processor Packet Completion Interrupt Flag bit

- 1 = DMA BD packet is complete
- 0 = DMA BD packet is in progress

bit 9 **BDDONEIF:** DMA Buffer Descriptor Done Interrupt Flag bit

- 1 = DMA BD process is done
- 0 = DMA BD process is in progress

bit 8 **CONTHRIF:** Control Buffer Threshold Interrupt Flag bit

- 1 = The control buffer has more than THRES words of space available
- 0 = The control buffer has less than THRES words of space available

bit 7 **CONEMPTYIF:** Control Buffer Empty Interrupt Flag bit

- 1 = Control buffer is empty
- 0 = Control buffer is not empty

bit 6 **CONFULLIF:** Control Buffer Full Interrupt Flag bit

- 1 = Control buffer is full
- 0 = Control buffer is not full

bit 5 **RXTHRIF:** Receive Buffer Threshold Interrupt Flag bit<sup>(1)</sup>

- 1 = Receive buffer has more than RXINTTHR words of space available
- 0 = Receive buffer has less than RXINTTHR words of space available

bit 4 **RXFULLIF:** Receive Buffer Full Interrupt Flag bit

- 1 = Receive buffer is full
- 0 = Receive buffer is not full

bit 3 **RXEMPTYIF:** Receive Buffer Empty Interrupt Flag bit

- 1 = Receive buffer is empty
- 0 = Receive buffer is not empty

**Note 1:** In the case of Boot/XIP mode, the POR value of the receive buffer threshold is zero. Therefore, this bit will be set to a '1', immediately after a POR until a read request on the System Bus bus is received.

**Note:** The bits in the register are cleared by writing a '1' to the corresponding bit position.

**Register 46-9: SQI1INTSTAT: SQI Interrupt Status Register (Continued)**

- bit 2     **TXTHRIF:** Transmit Buffer Threshold Interrupt Flag bit  
          1 = Transmit buffer has more than TXINTTHR words of space available  
          0 = Transmit buffer has less than TXINTTHR words of space available
- bit 1     **TXFULLIF:** Transmit Buffer Full Interrupt Flag bit  
          1 = The transmit buffer is full  
          0 = The transmit buffer is not full
- bit 0     **TXEMPTYIF:** Transmit Buffer Empty Interrupt Flag bit  
          1 = The transmit buffer is empty  
          0 = The transmit buffer has content

**Note 1:** In the case of Boot/XIP mode, the POR value of the receive buffer threshold is zero. Therefore, this bit will be set to a '1', immediately after a POR until a read request on the System Bus bus is received.

**Note:** The bits in the register are cleared by writing a '1' to the corresponding bit position.







**Register 46-14: SQI1BDCON: SQI Buffer Descriptor Control Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
	—	—	—	—	—	START	POLLEN	DMAEN

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

- bit 31-3 **Unimplemented:** Read as '0'
- bit 2 **START:** Buffer Descriptor Processor Start bit
  - 1 = Start the buffer descriptor processor
  - 0 = Disable the buffer descriptor processor
- bit 1 **POLLEN:** Buffer Descriptor Poll Enable bit
  - 1 = BDP poll enabled
  - 0 = BDP poll is not enabled
- bit 0 **DMAEN:** DMA Enable bit
  - 1 = DMA is enabled
  - 0 = DMA is disabled

**Register 46-15: SQI1BDCURADD: SQI Buffer Descriptor Current Address Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	BDCURRADDR[31:24]							
23:16	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	BDCURRADDR[23:16]							
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	BDCURRADDR[15:8]							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	BDCURRADDR[7:0]							

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

- bit 31-0 **BDCURRADDR[31:0]:** Current Buffer Descriptor Address bits  
 These bits contain the address of the current descriptor being processed by the Buffer Descriptor Processor.



**Register 46-18: SQI1BDPOLLCON: SQI Buffer Descriptor Poll Control Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	POLLCON[15:8]							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	POLLCON[7:0]							

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

bit 31-16 **Unimplemented:** Read as '0'

bit 15-0 **POLLCON[15:0]:** Buffer Descriptor Processor Poll Status bits

These bits indicate the number of cycles the BDP would wait before refetching the descriptor control word if the previous descriptor fetched was disabled.

**Register 46-19: SQI1BDTXDSTAT: SQI Buffer Descriptor DMA Transmit Status Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R-x	R-x	R-x	R-x	U-0
	—	—	—	TXSTATE[3:0]				—
23:16	U-0	U-0	U-0	R-x	R-x	R-x	R-x	R-x
	—	—	—	TXBUFCNT[4:0]				—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	TXCURBUFLen[7:0]							

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

bit 31-29 **Unimplemented:** Read as '0'

bit 28-25 **TXSTATE[3:0]:** Current DMA Transmit State Status bits

These bits provide information on the current DMA receive states.

bit 24-21 **Unimplemented:** Read as '0'

bit 20-16 **TXBUFCNT[4:0]:** DMA Buffer Byte Count Status bits

These bits provide information on the internal buffer space.

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **TXCURBUFLen[7:0]:** Current DMA Transmit Buffer Length Status bits

These bits provide the length of the current DMA transmit buffer.



# Section 46. Serial Quad Interface (SQI)

**Register 46-22: SQI1INTSIGEN: SQI Interrupt Signal Enable Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	—	DMAEISE	PKT COMPISE	BD DONEISE	CON THRISE
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CON EMPTYISE	CON FULLISE	RX THRISE	RX FULLISE	RX EMPTYISE	TX THRISE	TX FULLISE	TX EMPTYISE

**Legend:**

R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'  
 -n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

- bit 31-12 **Unimplemented:** Read as '0'
- bit 11 **DMAEISE:** DMA Bus Error Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 10 **PKTCOMPISE:** DMA Buffer Descriptor Packet Complete Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 9 **BDDONEISE:** DMA Buffer Descriptor Done Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 8 **CONTHRISE:** Control Buffer Threshold Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 7 **CONEMPTYISE:** Control Buffer Empty Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 6 **CONFULLISE:** Control Buffer Full Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 5 **RXTHRISE:** Receive Buffer Threshold Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 4 **RXFULLISE:** Receive Buffer Full Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 3 **RXEMPTYISE:** Receive Buffer Empty Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 2 **TXTHRISE:** Transmit Buffer Threshold Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 1 **TXFULLISE:** Transmit Buffer Full Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled
- bit 0 **TXEMPTYISE:** Transmit Buffer Empty Interrupt Signal Enable bit  
 1 = Interrupt signal is enabled  
 0 = Interrupt signal is disabled

# PIC32 Family Reference Manual

**Register 46-23: SQI1TAPCON: SQI TAP Control Register**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	DDRCLKINDLY[5:0]					
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SDRDATINDLY[3:0]				DDRDATINDLY[3:0]			
15:8	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	SDRCLKINDLY[5:0]					
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DATAOUTDLY[3:0]				CLKOUTDLY[3:0]			

**Legend:**

R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'  
 -n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

bit 31-30 **Unimplemented:** Read as '0'

bit 29-24 **DDRCLKINDLY[5:0]:** SQI Clock Input Delay in DDR Mode bits

These bits are used to add fractional delays to SQI Clock Input while sampling the incoming data in DDR mode.

111111 = 64 taps added on clock input  
 111110 = 63 taps added on clock input

- 
- 
- 

000001 = 2 taps added on clock input  
 000000 = 1 tap added on clock input

bit 23-20 **SDRDATINDLY[3:0]:** SQI Data Input Delay in SDR Mode bits

These bits are used to add fractional delays to SQI Data Output while writing the data to the Flash in SDR mode.

1111 = 16 taps added on data input  
 1110 = 15 taps added on data input

- 
- 
- 

0001 = 2 taps added on data input  
 0000 = 1 tap added on data input

bit 19-16 **DDRDATINDLY[3:0]:** SQI Data Output Delay in DDR Mode bits

These bits are used to add fractional delays to SQI Data Output while writing the data to the Flash in DDR mode.

1111 = 16 taps added on data input  
 1110 = 15 taps added on data input

- 
- 
- 

0001 = 2 taps added on data input  
 0000 = 1 tap added on data input

bit 15-14 **Unimplemented:** Read as '0'

**Note:** This register is not available on all devices. Refer to the “SQI” chapter in the specific device data sheet to determine availability.

**Register 46-23: SQI1TAPCON: SQI TAP Control Register (Continued)**bit 13-8 **SDRCLKINDLY[5:0]:** SQI Clock Input Delay in SDR Mode bits

These bits are used to add fractional delays to SQI Clock Input while sampling the incoming data in DDR mode.

111111 = 64 taps added on clock input

111110 = 63 taps added on clock input

•  
•  
•

000001 = 2 taps added on clock input

000000 = 1 tap added on clock input

bit 7-4 **DATAOUTDLY[3:0]:** SQI Data Output Delay bits

These bits are used to add fractional delays to SQI Data Output while writing the data to the Flash in all modes of operation.

1111 = 16 taps added on data output

1110 = 15 taps added on data output

•  
•  
•

0001 = 2 taps added on data output

0000 = 1 tap added on data output

bit 3-0 **CLKOUTDLY[3:0]:** SQI Clock Output Delay bits

These bits are used to add fractional delays to SQI Clock Output while writing the data to the Flash in all modes of operation.

1111 = 16 taps added on clock output

1110 = 15 taps added on clock output

•  
•  
•

0001 = 2 taps added on clock output

0000 = 1 tap added on clock output

**Note:** This register is not available on all devices. Refer to the “SQI” chapter in the specific device data sheet to determine availability.



**Register 46-25: SQI1XCON3: SQI XIP Control Register 3**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	INIT1SCHECK	INIT1COUNT[1:0]	INIT1TYPE[1:0]		
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INIT1CMD3[7:0]							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INIT1CMD2[7:0]							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INIT1CMD1[7:0]							

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

bit 31-29 **Unimplemented:** Read as '0'

bit 28 **INIT1SCHECK:** Flash Initialization 1 Command Status Check bit

- 1 = Check the status after executing the INIT1 commands
- 0 = Do not check the status

bit 27-26 **INIT1COUNT[1:0]:** Flash Initialization 1 Command Count bits

- 11 = INIT1CMD1, INIT1CMD2, and INIT1CMD3 are sent
- 10 = INIT1CMD1 and INIT1CMD2 are sent, but INIT1CMD3 is still pending
- 01 = INIT1CMD1 is sent, but INIT1CMD2 and INIT1CMD3 are still pending
- 00 = No commands are sent

bit 25-24 **INIT1TYPE[1:0]:** Flash Initialization 1 Command Type bits

- 11 = Reserved
- 10 = INIT1 commands are sent in Quad Lane mode
- 01 = INIT1 commands are sent in Dual Lane mode
- 00 = INIT1 commands are sent in Single Lane mode

bit 23-16 **INIT1CMD3[7:0]:** Flash Initialization Command 3 bits

Third command of the Flash initialization.

bit 15-8 **INIT1CMD2[7:0]:** Flash Initialization Command 2 bits

Second command of the Flash initialization.

bit 7-0 **INIT1CMD1[7:0]:** Flash Initialization Command 1 bits

First command of the Flash initialization.

**Note 1:** This register is not available on all devices. Refer to the “SQI” chapter in the specific device data sheet to determine availability.

**2:** Some Flash devices require write enable and sector unprotect commands before write/read operations, and this register is useful in working with those Flash types (XIP mode only).

# PIC32 Family Reference Manual

**Register 46-26: SQI1XCON4: SQI XIP Control Register 4**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	INIT2SCHECK	INIT2COUNT[1:0]		INIT2TYPE[1:0]	
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INIT2CMD3[7:0]							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INIT2CMD2[7:0]							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INIT2CMD1[7:0]							

**Legend:**

R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'  
-n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

bit 31-29 **Unimplemented:** Read as '0'

bit 28 **INIT2SCHECK:** Flash Initialization 2 Command Status Check bit

- 1 = Check the status after executing the INIT2 commands
- 0 = Do not check the status

bit 27-26 **INIT2COUNT[1:0]:** Flash Initialization 2 Command Count bits

- 11 = INIT2CMD1, INIT2CMD2, and INIT2CMD3 are sent
- 10 = INIT2CMD1 and INIT2CMD2 are sent, but INIT2CMD3 is still pending
- 01 = INIT2CMD1 is sent, but INIT2CMD2 and INIT2CMD3 are still pending
- 00 = No commands are sent

bit 25-24 **INIT2TYPE[1:0]:** Flash Initialization 2 Command Type bits

- 11 = Reserved
- 10 = INIT2 commands are sent in Quad Lane mode
- 01 = INIT2 commands are sent in Dual Lane mode
- 00 = INIT2 commands are sent in Single Lane mode

bit 23-16 **INIT2CMD3[7:0]:** Flash Initialization Command 3 bits

Third command of the Flash initialization.

bit 15-8 **INIT2CMD2[7:0]:** Flash Initialization Command 2 bits

Second command of the Flash initialization.

bit 7-0 **INIT2CMD1[7:0]:** Flash Initialization Command 1 bits

First command of the Flash initialization.

**Note 1:** This register is not available on all devices. Refer to the “**SQI**” chapter in the specific device data sheet to determine availability.

**2:** Some Flash devices require write enable and sector unprotect commands before write/read operations and this register is useful in working with those Flash types (XIP mode only).

## 46.5 SQI OPERATION

**Note:** Not all features discussed in this section are available in all devices. Refer to the specific device data sheet to determine availability.

As previously mentioned in **46.3 “Functional Block Description”**, the SQI module is primarily used to communicate with serial Flash memory devices. The serial Flash devices support operations, such as ERASE, READ, and WRITE through a set of command sequences, which are issued by a host controller, in this case SQI. The SQI module facilitates these command sequences through the following prominent interface features:

- Single, Dual, or Quad lane modes
- Single Data Rate (SDR) or Double Data Rate (DDR) speeds
- SPI Mode 0 or Mode 3
- DMA, PIO, or XIP transfer modes
- Flash status check
- Tap delays at high interface speeds

### 46.5.1 Single, Dual or Quad lane modes

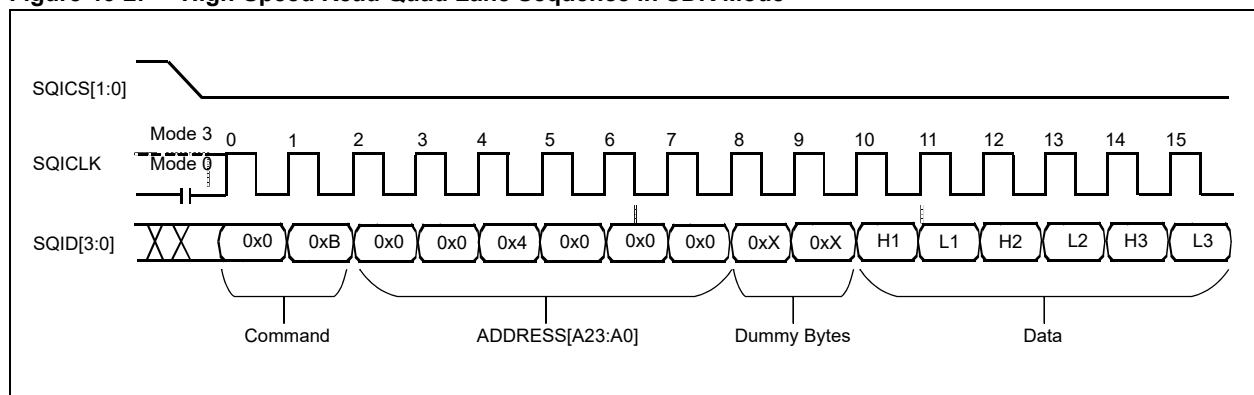
The lane modes (single/dual/quad), as the names imply set the interface to exercise transactions using single (SQID0), dual (SQID0, SQID1) or quad (SQID0-SQID3) data lanes. The majority of serial Flash devices provide commands specifically to exercise the transactions in a specific lane mode (e.g., JEDEC-ID to read device ID in single lane mode and QJID to read the same device ID in quad lane mode in case of SST26VF series devices).

### 46.5.2 Single Data Rate (SDR) or Double Data Rate (DDR) speeds

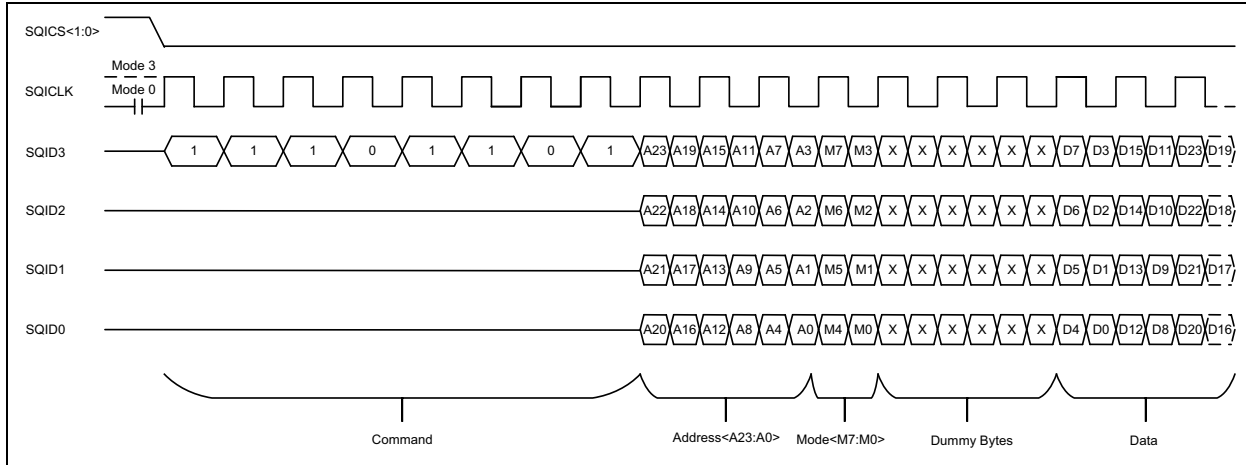
In SDR mode, the data transaction occurs only on the rising edge of the clock, whereas in DDR mode the transactions occur on both the rising and falling edge of the clock, providing double the throughput. Some serial Flash devices support commands specifically aimed at DDR mode (e.g., the 4DDRQIOR command to read from Flash using quad I/O in DDR mode in case of Spansion memories supporting DDR mode).

Figure 46-2 and Figure 46-3 show the high-speed read quad lane sequence in SDR mode and DDR mode, respectively.

**Figure 46-2: High-Speed Read Quad Lane Sequence in SDR Mode**



**Figure 46-3: High-Speed Read Quad Lane Sequence in DDR Mode**



### 46.5.3 SPI Mode 0 or Mode 3

The SQI module supports the two most prominent SPI data flow modes, Mode 0 and Mode 3, which are controlled by the CPOL bit (SQI1CFG[4]) and the CPHA bit (SQI1CFG[3]). For additional details refer to [46.7 “SQI Data Flow Modes”](#).

### 46.5.4 DMA, PIO or XIP transfer modes

The SQI module operates in three transfer modes: DMA, PIO and XIP. For additional details on the modes of transfer, refer to [46.6 “SQI Transfer Modes”](#).

### 46.5.5 Flash Status Check

The SQI module supports a hardware-based Flash status check in DMA, PIO, and XIP modes, thereby reducing the burden on software. The status check option is user-configurable, and checks the status of the Flash by automatically reading the Flash status register and checks the RDY/BUSY status flag. If a specific Flash command (i.e., programming or erase type of operations) requires a status check, the user can use this feature, and when enabled, the SQI module will not proceed with the next command in the queue until the status check on the current command returns the RDY state.

In DMA and PIO modes, the SQI module uses the SCHECK bit (BD\_CTRL[27]) in DMA mode and the DDRCMD bit (SQI1CON[24]) in PIO mode in combination with the SQI1MEMSTAT register to handle the Flash status check operation.

In XIP mode, the INIT1SCHECK (SQI1XCON3[28]) and INIT2SCHECK (SQI1XCON4[28]) bits in combination with different commands in the same registers facilitates the Flash status check operation.

### 46.5.6 Tap Delays at High Interface Speeds

The SQI module provides a tap control register, SQI1TAPCON, which can be used to adjust the timing between the SQICLK and SQID0-SQID3 signals to compensate for the data delays control the setup and hold times at higher speeds. Each tap adds a certain delay on the signal that can be used to control the clock and data relationship, and may be useful in certain instances to compensate for the PCB routing delays. Refer to the **“Electrical Characteristics”** chapter in the specific device data sheet for the exact delay that each tap element adds.

**Note:** By default, the SQI1TAPCON register is set to work at optimal interface speeds and requires no user programming. However, advanced users interested in controlling the SQICLK and SQIDx pins to the application can use this feature.

## 46.6 SQI TRANSFER MODES

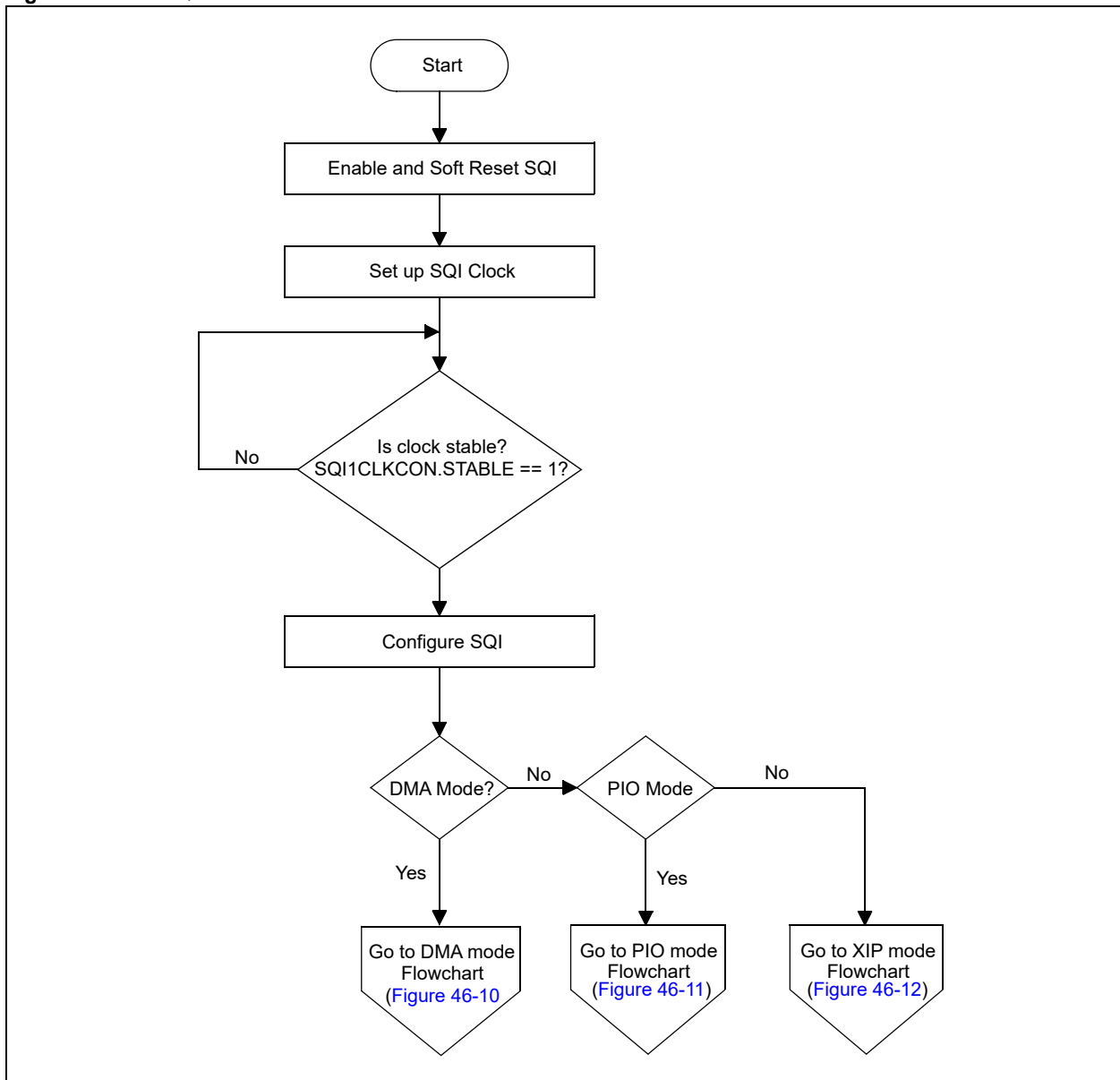
**Note:** To avoid cache coherency problems on devices with L1 cache, all SQI buffers described in this section must only be accessed from the KSEG1 segment.

The SQI module operates in three transfer modes: DMA, PIO, and XIP. As mentioned earlier, DMA and PIO modes are typically used to transfer data, whereas XIP mode is used to execute the code out of the attached serial Flash memory space. DMA mode uses the internal DMA engine and linked-list type of structures to transfer data between source and destination memory spaces, making it a little more automated, resulting in less software overhead and CPU intervention. DMA mode can be considered as a high throughput data transfer mode. In PIO mode, the CPU can access the contents of the attached serial memory device through the SQI transmit data and receive data registers with status and interrupt flag assistance.

Each transfer mode can use either of the data flow modes (Mode 0, Mode 3) for transactions. Refer to section 46.7 “SQI Data Flow Modes” for additional details.

Before initiating any transfer mode the SQI needs to be initialized properly. Figure 46-4 shows the SQI Initialization Flow diagram.

Figure 46-4: SQI Initialization Flowchart



Initialization code examples are provided in [Example 46-1](#) and [Example 46-2](#).

## Example 46-1: SQI Initialization Code

```
{
    CFGCONbits.TROEN=0;           // Disable trace outputs (SQI pins share trace)

    // REFCLKO2 is assumed to be SQI base clock
    If (!REFO2CONbits.ACTIVE) // Check if REFCLKO2 divider circuit is active
    {
        REFO2CONbits.RODIV=1;      // Set the divider to 1 (SYSCLK/2)
        REFO2CONbits.ON=1;        // Turn on the divider circuit
        while (REFO2CONbits.DIVSWEN); // Wait for divide to occur
        REFO2CONbits.OE=1;        // Output enable
    }

    SQI1CFG = 0x80010000;          // Enable and reset SQI

    SQI1CFG = 0x82209019;          // Configure SQI

    SQI1CLKCON |= (0x01 << 8);    // Set divider to 1 (TBC/2)

    SQI1CLKCON = 0x00000001;      // Enable clock circuit

    while (!SQI1CLKCONbits.STABLE) // Wait for clock to be stable
}
}
```

## Example 46-2: Serial Flash Initialization Code (PIO Mode)

```
// Flash Commands
#define SST26VF_ERASE          0xC7
#define SST26VF_EQIO          0x38
#define SST26VF_WEN           0x06
#define SST26VF_NOP           0

{
    uint32_t blockProtectLoop;

    SQI1THR = 0x00000100;        // Set control buffer threshold to 1 word
    SQI1CMDTHR = 0x00000404;     // Set SQI TX/RX command threshold to 4 bytes
    SQI1INTTHR = 0x00000404;     // Set SQI TX/RX interrupt threshold to 4 bytes

    // Set up control buffer to set SQI Flash in quad lane mode and send
    // erase the flash command
    SQI1CON = 0x00510001;        // NOP (single lane)
    SQI1CON = 0x00510001;        // 1-byte EQIO (single lane)
    SQI1CON = 0x00590001;        // 1 byte for enable flash write (quad lane)
    SQI1CON = 0x00590001;        // 1 byte for erase (quad lane)

    // Write to transmit buffer
    SQI1TXBUF = SST26VF_ERASE << 24 |
                SST26VF_WEN << 16 |
                SST26VF_EQIO << 8 |
                SST26VF_NOP;

    // Wait for 38 ms for erase through a timer delay loop

    SQI1CON = 0x00590001;        // NOP (quad lane)
    SQI1CON = 0x00590001;        // NOP (quad lane)
    SQI1CON = 0x00590001;        // WREN (quad lane)
    SQI1CON = 0x00190001;        // 1 byte for block unprotect command (quad lane)
    SQI1CON = 0x0019000C;        // 12 bytes for unprotect bytes

    // Write to transmit buffer
    SQI1TXBUF = SST26VF_BLKUP << 24 |
                SST26VF_WEN << 16 |
                SST26VF_NOP << 8 |
                SST26VF_NOP;

    SQI1TXBUF = 0x00000000;
    SQI1TXBUF = 0x00000000;
    SQI1TXBUF = 0x00000000;
}
}
```

**Note:** This sequence is applicable to the SST26VF032 device. Each Flash device may require its own initialization sequence.

## 46.6.1 DMA Mode

DMA mode is a higher throughput data transfer mode, where the SQI DMA engine off-loads the Host processor by using predefined Buffer Descriptors for data transfers. Each Buffer Descriptor can handle up to 64KB of data and multiple descriptors can be chained to support larger portions of data transfers. See [Figure 46-9](#) for an example.

### 46.6.1.1 SQI BUFFER DESCRIPTORS

Buffer Descriptors are part of the SQI DMA mode of operation. In addition to the control registers described in [46.4 “Control Registers”](#), the SQI module contains four words, as shown in [Table 46-2](#) to handle the data transactions in DMA mode. Four words cumulatively are called a Buffer Descriptor (BD), and are part of the transmit and receive structure. Individually, a Buffer Descriptor describes an area within the Host memory where data is waiting to be transmitted from or received into. Host software creates a single or linked list of Buffer Descriptors based on the buffer size and places them in the system memory. Host software uses the SQI Buffer Descriptor Base Address Register (SQI1BDBASEADD) to point to the first BD of the linked list. From this point forward, each BD points to the next descriptor using the BD\_NXTPTR word. The Buffer Descriptors can be arranged either in chained fashion or in ring fashion, as shown in [Figure 46-9](#).

**Note:** If no Buffer Descriptors are available for transmission or reception, the DMA engine enters Idle mode.

- **BD\_CTRL** – Buffer Descriptor Control Word (see [Figure 46-5](#))  
This register contains the control bits to process the Buffer Descriptor. The user application should program this register prior to entering DMA mode.
- **BD\_STAT** – Buffer Descriptor Status Word (see [Figure 46-6](#))  
This register is reserved. DMA status can be monitored through the SQI1INTSTAT, SQI1BDSTAT, SQI1BDTXDSTAT and SQI1BDRXDSTAT registers.
- **BD\_BUFADDR** – Buffer Descriptor Current Buffer Address Word (see [Figure 46-7](#))  
This register contains the address pointer to the current buffer location. In the event of a transmit, this address in this register is interpreted as the source address, while in the event of a receive address, this register becomes the destination address.
- **BD\_NXTPTR** – Buffer Descriptor Next Buffer Descriptor Address Word (see [Figure 46-8](#))  
This register contains address pointer to the next Buffer Descriptor.

**Table 46-2: SQI DMA Buffer Descriptors**

Name	Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
BD_CTRL	31:24	DESCEN	DEASSERT	SQICS[1:0]		SCHECK	—	LSBF	—	
	23:16	MODE[1:0]		—	DIR	LASTBD	LASTPKT	PKTINTEN	BDDONE INTEN	
	15:8	—	—	—	—	—	—	—	BUFLEN[8]	
	7:0	BUFLEN[7:0]								
BD_STAT	31:24	—	—	—	—	—	—	—	—	
	23:16	—	—	—	—	—	—	—	—	
	15:8	—	—	—	—	—	—	—	—	
	7:0	—	—	—	—	—	—	—	—	
BD_BUFADDR	31:24	BUFADDR[31:24]								
	23:16	BUFADDR[23:16]								
	15:8	BUFADDR[15:8]								
	7:0	BUFADDR[7:0]								
BD_NXTPTR	31:24	NXTBDADDR[31:24]								
	23:16	NXTBDADDR[23:16]								
	15:8	NXTBDADDR[15:8]								
	7:0	NXTBDADDR[7:0]								

**Note:** The DMA buffer descriptors should be initialized by the CPU using an uncached virtual address. The contents of the buffer descriptors should contain only physical addresses (see [Figure 46-4](#)) as the SQI module does not understand virtual addressing.

# PIC32 Family Reference Manual

Figure 46-5: Format of BD\_CTRL – Buffer Descriptor Control Word

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31-24	DESCEN	DEASSERT	SQICS[1:0]		SCHECK <sup>(4)</sup>	—	LSBF	—
23-16	MODE[1:0]		—	DIR	LASTBD <sup>(1)</sup>	LASTPKT	PKTINTEN <sup>(2)</sup>	BDDONEINTIEN <sup>(2)</sup>
15-8	—	—	—	—	—	—	—	BUFLEN[8]
7-0	BUFLEN[7:0] <sup>(3)</sup>							

- bit 31 **DESCEN:** Buffer Descriptor Enable  
 1 = The descriptor is owned by hardware. After processing the BD, hardware resets this bit to '0'.  
 0 = The descriptor is owned by software
- bit 30 **DEASSERT:** Chip Select Assert  
 1 = Chip Select is deasserted after transmission or reception of the specified number of bytes  
 0 = Chip Select is not deasserted after transmission or reception of the specified number of bytes
- bit 29-28 **SQICS[1:0]:** SQI Chip Select  
 11 = Reserved  
 10 = Reserved  
 01 = Selects device on SQICS1  
 00 = Selects device on SQICS0
- bit 27 **Unimplemented:** Read as '0'
- bit 26 **SCHECK:** Flash Status Check  
 1 = Check the status of the Flash  
 0 = Do not check the status of the Flash
- bit 25 **LSBF:** Data Format  
 1 = LSB is sent or received first  
 0 = MSB is sent or received first
- bit 24 **Unimplemented:** Read as '0'
- bit 23-22 **MODE[1:0]:** Indicates Lane Mode  
 11 = Reserved  
 10 = Quad Lane mode  
 01 = Dual Lane mode  
 00 = Single Lane mode
- bit 21 **Unimplemented:** Read as '0'
- bit 20 **DIR:** Data Direction  
 1 = Data is transferred from an internal buffer to memory location specified by the Host  
 0 = Data is transferred from a memory location specified by the Host to an internal buffer
- bit 19 **LASTBD:** Last Buffer Descriptor<sup>(1)</sup>  
 1 = Last descriptor in the Buffer Descriptor chain  
 0 = Not the last descriptor
- bit 18 **LASTPKT:** Last Packet of the Buffer Descriptor in Progress  
 This bit is set at the same time as LASTBD and is used to trigger the packet complete interrupt (PKTCOMPIFS)  
 1 = Indicates the last packet of the data set  
 0 = Not the last packet
- bit 17 **PKTINTEN:** Packet Interrupt Enable<sup>(2)</sup>  
 1 = Packet interrupt is enabled  
 0 = Packet interrupt is not enabled
- bit 16 **BDDONEINTIEN:** Buffer Descriptor Done Interrupt Enable<sup>(2)</sup>  
 1 = Buffer Descriptor process has completed  
 0 = Buffer Descriptor process is in progress
- bit 15-9 **Unimplemented:** Read as '0'
- bit 8-0 **BUFLEN[8:0]:** Length of the Transfer Buffer in Bytes<sup>(3)</sup>  
 This field contains the length of the transfer buffer and is updated as the transfer progresses.

- Note**
- 1: When this bit is set to '1', the DMA engine ignores the value in the BD\_NXTPTR.
  - 2: For packet complete interrupt (PKTCOMPIF) and Buffer Descriptor done interrupt (BDDONEIF) to flag, PKTIEN and BDDONEIEN in the BD\_CTRL word should be set along with corresponding bits (PKTCOMPIE and BDDONEIE) in the SQI1INTEN register.
  - 3: The largest amount of data that can be referenced by a single Buffer Descriptor is 256 bytes (2<sup>BUFLEN-1</sup> bytes).
  - 4: This field is not available on all devices. Refer to the "SQI" chapter in the specific device data sheet to determine availability.

**Figure 46-6: Format of BD\_STAT – Buffer Descriptor Status Word**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31-24	—	—	—	—	—	—	—	—
23-16	—	—	—	—	—	—	—	—
15-8	—	—	—	—	—	—	—	—
7-0	—	—	—	—	—	—	—	—

bit 31      **Unimplemented:** Read as '0'

**Figure 46-7: Format of BD\_BUFADDR – Buffer Descriptor Current Buffer Address Word**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31-24	BUFADDR[31:24]							
23-16	BUFADDR[23:16]							
15-8	BUFADDR[15:8]							
7-0	BUFADDR[7:0]							

bit 31      **BUFADDR[31:0]:** Transmit/Receive Buffer Address  
 This field contains the address of the buffer in system memory. The DIR bit in BD\_CTRL indicates whether this is a transmit/receive buffer.

**Figure 46-8: Format of BD\_NXTPTR – Buffer Descriptor Next Buffer Descriptor Address Word**

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31-24	NXTBDADDR[31:24]							
23-16	NXTBDADDR[23:16]							
15-8	NXTBDADDR[15:8]							
7-0	NXTBDADDR[7:0]							

bit 31      **NXTBDADDR[31:0]:** Next Buffer Descriptor Address

As shown, in [Figure 46-9](#), the chain of Buffer Descriptors are provided as an example to read (High-Speed) 256 bytes of data from the serial Flash device attached to the SQICS0 signal in Quad Lane mode. In this example, BD0 and BD1 are configured as transmit buffer descriptors and BD2 is configured as a receive buffer descriptor. The buffer space (0x0002000) in memory that BD0 points to should be preloaded with the 4 bytes (0x0B command and 0x001000 address). The buffer space (0x00020004) in memory that BD1 points to should be preloaded with 1 byte (0x00 as dummy).

Once the DMA process is complete, Buffer 2 (0x00020100) should contain the data read from the serial Flash device. For a better understanding of buffer descriptor control value of each descriptor, refer to [Figure 46-5](#). Refer to [46.8 “Flash Instructions and Sequence Diagrams Quick Reference”](#) for additional Flash instructions.

**Figure 46-9: Buffer Descriptor Structure Example**

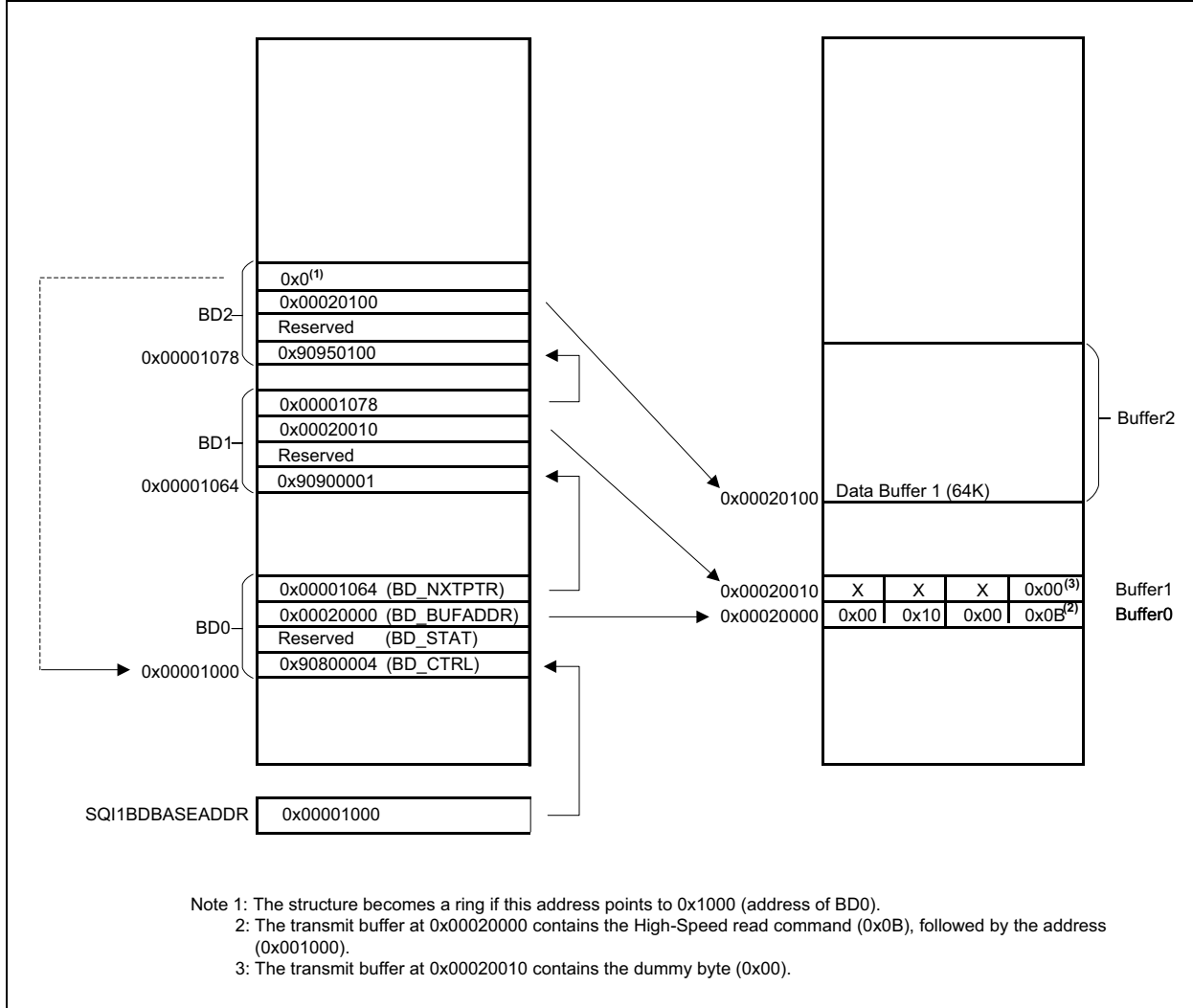
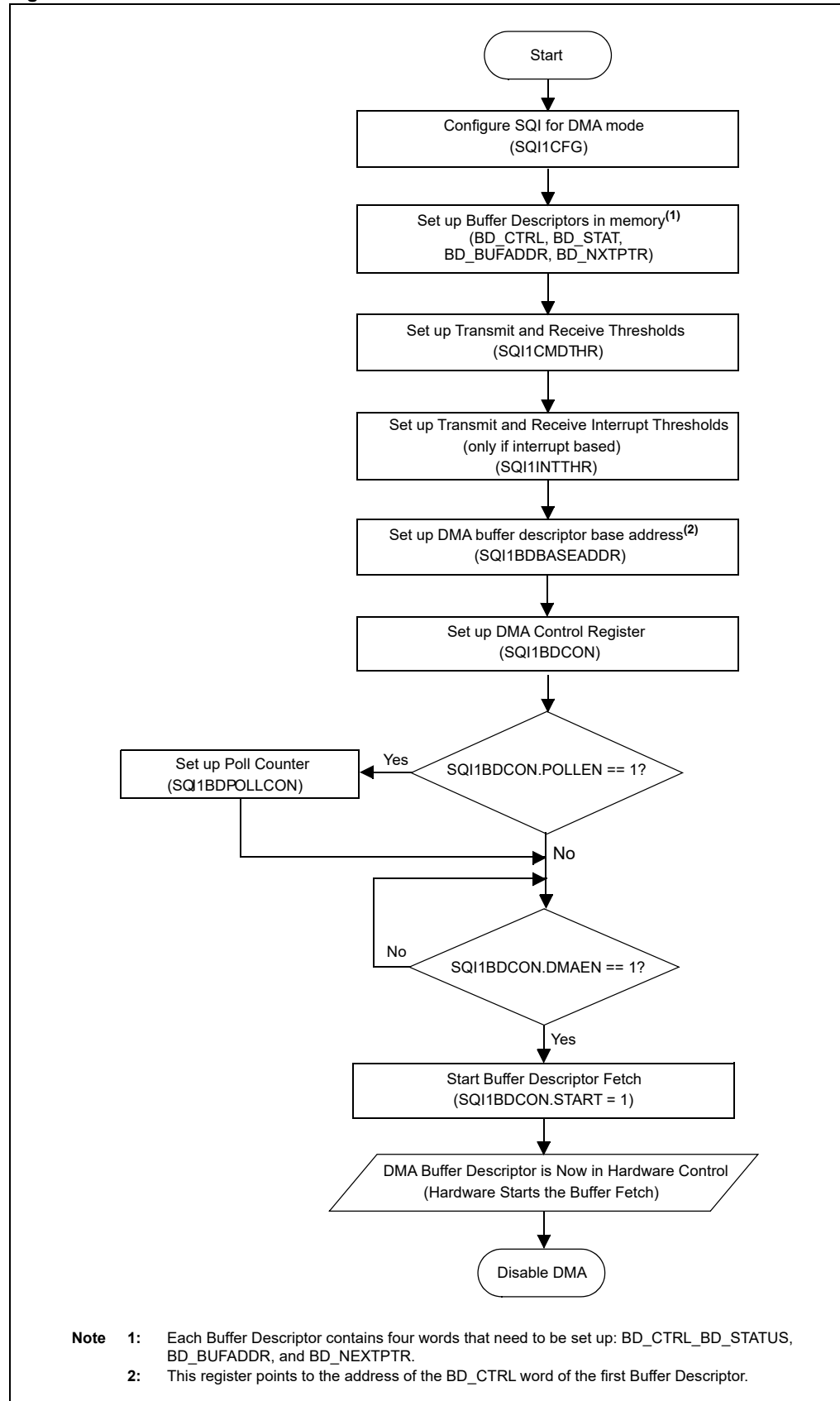


Figure 46-10 shows the DMA mode flow diagram.

**Figure 46-10: DMA Mode Flowchart**



Code to configure DMA mode is provided in [Example 46-3](#).

## Example 46-3: DMA Mode Configuration Code

```
// Assumptions: Flash is already written; Example code is polling based; Best throughput is
// achieved in interrupt-based code.
// Flash commands
#define SST26VF_FAST_READ          0x0B

#define BD_BUFFER1_ADDR            0xA0010000
#define BD_BUFFER2_ADDR            0xA0010010
#define BD_BUFFER1_PA_ADDR        0x00010000
#define BD_BUFFER2_PA_ADDR        0x00010010

#define FLASH_PAGE_ADDR           0x00001000

#define WRITE_BUF_ADDR            0xA0060000
#define PIC32_KVA0_TO_KVA1_VAR(v) ((typeof(v))*((unsigned long)&(v) | 0x20000000u))
#define PIC32_KVA0_TO_KVA1_PTR(v) ((typeof(v))*((unsigned long)(v) | 0x20000000u))
#define PIC32_UNCACHED_VAR(v)    PIC32_KVA0_TO_KVA1_VAR(v)
#define PIC32_UNCACHED_PTR(v)    PIC32_KVA0_TO_KVA1_PTR(v)

typedef struct
{
    unsigned int BDCon;           // Buffer Descriptor Control Word
    unsigned int BDStat;         // Buffer Descriptor Status Word - reserved
    unsigned int *BDAddr;        // Buffer Address
    struct sqiDMADesc *nextBDAddr; // Next Buffer Descriptor Address Pointer
} sqiDMADesc;

{
    sqiDMADesc pSqiDMADesc1;
    sqiDMADesc pSqiDMADesc2;
    sqiDMADesc pSqiDMADesc3;
    uint32_t checkLoop;
    uint8_t *readBufInByte = (uint8_t *) BD_BUFFER2_ADDR;
    uint8_t *writeBuf = (uint8_t *) WRITE_BUF_ADDR;
    unsigned int *buf1VAddr = (unsigned int *) BD_BUFFER1_ADDR;
    unsigned int *buf1PAAddr = (unsigned int *) BD_BUFFER1_PA_ADDR;
    unsigned int *buf2PAAddr = (unsigned int *) BD_BUFFER2_PA_ADDR;
    uint8_t tempAddress1, tempAddress2, tempAddress3;
    uint32_t errCount = 1;

    // Address manipulation
    tempAddress1 = (uint8_t) (address >> 16);
    tempAddress2 = (uint8_t) (address >> 8);
    tempAddress3 = (uint8_t) address;
    address = tempAddress1 | tempAddress2 << 8 | tempAddress3 << 16;

    // Initialize data in Buffer1 (transmit buffer)
    *buf1VAddr++ = address << 8 | SST26VF_FAST_READ;
    *buf1VAddr = 0x00000000;

    SQI1CFG = 0x8220901A;           // Configure SQI for DMA mode

    // Set up buffer descriptors in memory (SRAM)
    SQI1INTENbits.BDDONE = 1;      //Enable BDDONE interrupt (masked)

    // Set up Buffer Descriptors in Memory
    //BD1
    PIC32_UNCACHED_VAR(pSqiDMADesc1.BDCon) = 0x90800004;
    PIC32_UNCACHED_VAR(pSqiDMADesc1.BDStat) = 0;
    PIC32_UNCACHED_VAR(pSqiDMADesc1.BDAddr) = (unsigned int*) buf1PAAddr;
    PIC32_UNCACHED_VAR(pSqiDMADesc1.nextBDAddr) = (struct sqiDMADesc *)KVA_TO_PA(&pSqiDMADesc2);

    //BD2
    PIC32_UNCACHED_VAR(pSqiDMADesc2.BDCon) = 0x90900001;
    PIC32_UNCACHED_VAR(pSqiDMADesc2.BDStat) = 0;
    PIC32_UNCACHED_VAR(pSqiDMADesc2.BDAddr) = (unsigned int*) buf1PAAddr+4;
    PIC32_UNCACHED_VAR(pSqiDMADesc2.nextBDAddr) = (struct sqiDMADesc *)KVA_TO_PA(&pSqiDMADesc3);

    //BD3
    PIC32_UNCACHED_VAR(pSqiDMADesc3.BDCon) = 0x90950100;
    PIC32_UNCACHED_VAR(pSqiDMADesc3.BDStat) = 0;
    PIC32_UNCACHED_VAR(pSqiDMADesc3.BDAddr) = (unsigned int*) buf2PAAddr;
    PIC32_UNCACHED_VAR(pSqiDMADesc3.nextBDAddr) = 0;

    SQI1CMDTHR = 0x00000418;       // Set up command thresholds (4, 24 bytes)
```

**Example 46-3: DMA Mode Configuration Code (Continued)**

```
//Load first descriptor address into Buffer Descriptor base address (physical address)
SQI1BDBASEADDR = (void *)KVA_TO_PA(&pSqIDMADesc1);

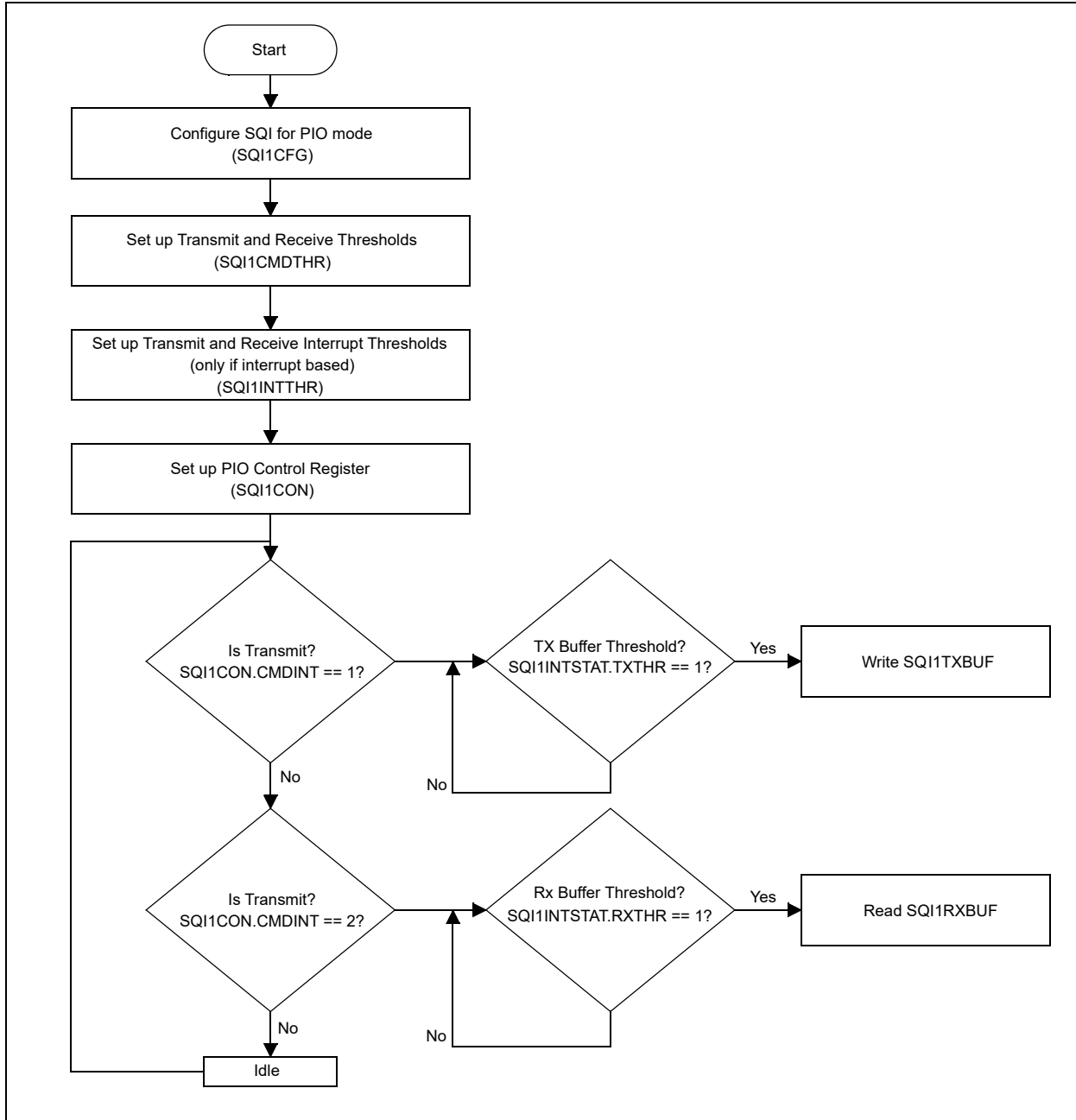
SQI1BDCON = 0x00000005;           // Enable and start DMA process

While (!SQI1STATbits.BDDONE);    // Wait for buffer process to finish
SQI1BDCON = 0;                   // Disable DMA
for (checkLoop=0; checkLoop <256; checkLoop++){
    if (*readBufInByte++ != *writeBuf++){
        errCount++;
    }
}
return errCount;
```

## 46.6.2 PIO Mode

The SQI module is controlled by the Host processor in PIO mode, indicating Host load during all PIO transactions. In PIO mode, the Host CPU configures the SQI through the SQI1CFG register and handles the control of each transaction through the SQI1CON register. The Host processor should actively monitor all of the status and interrupt bits to dynamically control the transactions. PIO mode is used for smaller portions of controlled data transfers. Figure 46-11 shows the basic PIO mode flow diagram.

Figure 46-11: PIO Mode Flowchart



Code to configure PIO mode is provided in [Example 46-4](#).

#### Example 46-4: PIO Mode Configuration Code

```
// Assumptions: Flash is already written; Example code is polling based; Best throughput is
// achieved in interrupt-based code.
// Flash commands
#define SST26VF_EQIO          0x38
#define SST26VF_WEN          0x06
#define SST26VF_NOP          0
#define SST26VF_FAST_READ0x0B

#define MAX_WRITE_BUF_DEPTH  16
#define MAX_READ_BUF_DEPTH   8
#define WRITE_BUF_ADDR       0xA0060000
#define PIO_READ_BUF_ADDR    0xA0060200
#define SQI_TXBUF_ADDR       0xBF8E2024
#define SQI_RXBUF_ADDR       0xBF8E2028
#define MAX_BUF_DEPTH        8

#define FLASH_PAGE_ADDR      0x00100000

// PIO mode write example
{
    uint32_t writeLoop, bufLoop;
    uint8_t writeLoopChar = 0;
    uint8_t * writeBufAddrChar = (uint8_t *) WRITE_BUF_ADDR;
    uint8_t * txBufChar = (uint8_t *) SQI_TXBUF_ADDR;
    uint8_t tempAddress1, tempAddress2, tempAddress3;

    // Set up transmit data
    for (writeLoop=0; writeLoop < 256; writeLoop++)
        *writeBufAddrChar++ = writeLoopChar++;

    writeBufAddrChar = (uint8_t *) WRITE_BUF_ADDR;

    // Address manipulation
    tempAddress1 = (uint8_t) (address >> 16);
    tempAddress2 = (uint8_t) (address >> 8);
    tempAddress3 = (uint8_t) address;
    address = tempAddress1 | tempAddress2 << 8 | tempAddress3 << 16;

    SQI1CFG = 0x82209019; // Configure SQI for PIO mode

    SQI1THR = 0x00000100; // Set up control threshold
    SQI1CMDTHR = 0x00000400; // Set up transmit command threshold
    SQI1INTTHR = 0x00000400; // Set up transmit interrupt threshold

    // SQI Transfer Configuration
    SQI1CON = 0x00510001; // NOP (quad lane)
    SQI1CON = 0x00510001; // NOP (quad lane)
    SQI1CON = 0x00510001; // NOP (quad lane)
    SQI1CON = 0x00590001; // WEN (quad lane)

    SQI1TXBUF = SST26VF_WEN << 24 | 0x00000000;
    // Send write command followed by address and data
    SQI1CON = 0x00190001; // 1 byte for send flash write opcode
    SQI1CON = 0x00190003; // 3 bytes of flash address
    SQI1CON = 0x00590100; // 256 bytes to be written

    SQI1TXBUF = address << 8 |
        SST26VF_PAGE_WRITE);

    // Write the data to flash
    for (writeLoop=0; writeLoop < 16; writeLoop++){
        APP_StartCoreTimer(0);
        APP_CoreTimer_Delay(400);
        while (!(PLIB_SQI_InterruptFlagGet(SQI_ID_0, SQI_TXTHR)));
        for (bufLoop=0; bufLoop < MAX_WRITE_BUF_DEPTH; bufLoop++){
            *txBufChar = *writeBufAddrChar++; // Next byte of write data
        }
        APP_StartCoreTimer(0);
        APP_CoreTimer_Delay(180000);
    }
}
```

# PIC32 Family Reference Manual

## Example 46-4: PIO Mode Configuration Code (Continued)

```
// PIO mode read example
{
    uint32_t readLoop, bufLoop, checkLoop;
    uint32_t *readBuf = (uint32_t *) PIO_READ_BUF_ADDR;
    uint32_t *writeBuf = (uint32_t *) WRITE_BUF_ADDR;
    uint8_t tempAddress1, tempAddress2, tempAddress3;
    uint32_t errCount = 1;

    // Address manipulation
    tempAddress1 = (uint8_t) (address >> 16);
    tempAddress2 = (uint8_t) (address >> 8);
    tempAddress3 = (uint8_t) address;
    address = tempAddress1 | tempAddress2 << 8 | tempAddress3 << 16;

    SQI1CFG = 0x82209019;           // Configure SQI for PIO mode
    SQI1THR = 0x00000100;          // Set up control threshold
    SQI1CMDTHR = 0x00000400;       // Set up transmit command threshold
    SQI1INTTHR = 0x00000400;       // Set up transmit interrupt threshold

    SQI1CON = 0x00510001;          // NOP
    SQI1CON = 0x00510001;          // NOP
    SQI1CON = 0x00510001;          // NOP
    SQI1CON = 0x00190001;          // FAST_READ
    SQI1CON = 0x00190004;          // ADDRESS, DUMMY

    SQI1TXBUF = SST26VF_FAST_READ << 24 | 0x00000000;

    SQI1TXBUF = 0x0 << 24 | address;

    SQI1CON = 0x005A0100;          // 256 bytes of data to be read from Flash
    SQI1CMDTHR = 0x0000001F;       // Configure transmit threshold
    for (readLoop = 0; readLoop < 8; readLoop++){
    {
        while ((SQI1STAT1 & 0xFF) != 0x1F);
        for (bufLoop=0; bufLoop < MAX_READ_BUF_DEPTH; bufLoop++){
            *readBuf++ = SQI1RXBUF; // Dump received data into a buffer
        }
    }
    readBuf = (uint32_t *) PIO_READ_BUF_ADDR;

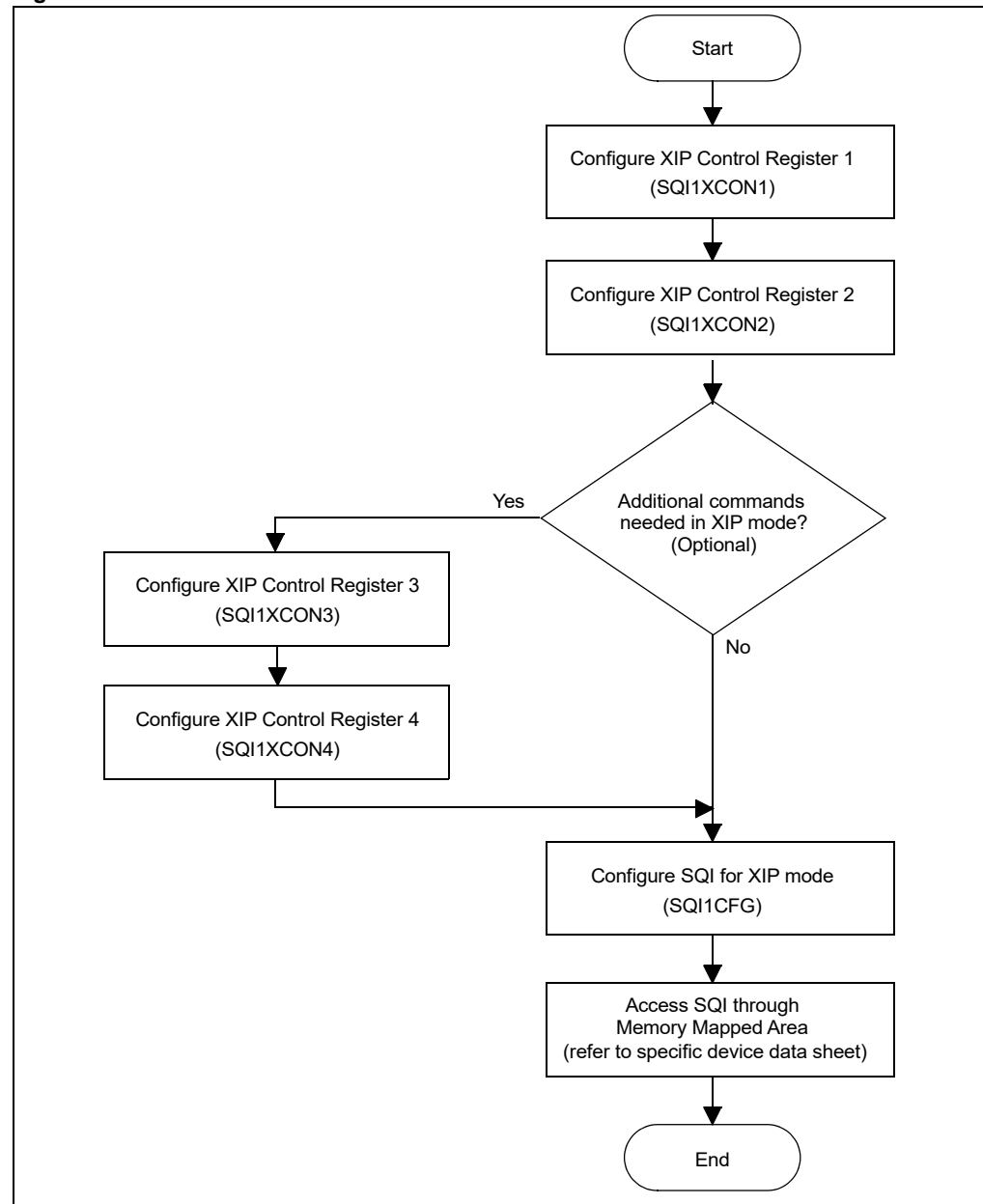
    for (checkLoop=0; checkLoop < 64; checkLoop++){
        if (*readBuf++ != *writeBuf++)
            errCount++;
    }

    return errCount;
}
```

**46.6.3 XIP Mode<sup>(1)</sup>**

XIP mode, as described in the previous sections, is used to execute code out of serial Flash devices. This mode uses an optimal timing setup for faster execution. For the SQI module to function properly in XIP mode, the Host processor should program the SQI1XCON1 and SQI1XCON2 registers as required, and then set the MODE[2:0] bits in the SQI1CFG register to switch to XIP mode. The SQI module has two additional control registers; SQI1XCON3 and SQI1XCON4, which provide additional command support that might be required by Flash prior to executing write and/or read commands. The SQI1XCON3 and SQI1XCON4 registers programming is optional. Figure 46-12 shows the XIP mode flow diagram.

**Note:** If XIP is done with CACHE enabled, make sure the SQI Flash device is in Burst mode and any reads are performed through burst read to avoid exceptions.

**Figure 46-12: XIP Mode Flowchart**

1. This feature is not available on all devices. Refer to the “SQI” chapter of the specific device data sheet to determine availability.

# PIC32 Family Reference Manual

Code to configure XIP mode is provided in [Example 46-5](#).

## Example 46-5: XIP Mode Configuration Code

```
// Assumptions: Flash is already written
// Flash Commands
#define SST26VF_FAST_READ0x0B

#define MAX_WRITE_BUF_DEPTH      16
#define MAX_READ_BUF_DEPTH      8
#define WRITE_BUF_ADDR           0xA0060000
#define XIP_READ_BUF_ADDR        0xA0060400
#define XIP_UNCACHED_ADDR_MASK   0xF0000000
#define SQI_TXBUF_ADDR           0xBF8E2024
#define SQI_RXBUF_ADDR           0xBF8E2028
#define MAX_BUF_DEPTH            8

#define FLASH_PAGE_ADDR          0x00100000
{
    uint32_t readLoop, checkLoop;
    uint32_t * writeBuf = (uint32_t *) WRITE_BUF_ADDR;
    uint32_t * sqiXIPAddr;
    uint32_t * readBuf = (uint32_t *) XIP_READ_BUF_ADDR;

    // Address decoding
    address = XIP_UNCACHED_ADDR_MASK | address;

    sqiXIPAddr = (uint32_t *) address;

    // All command set to QUAD lane mode, FASTREAD opcode, 3 address bytes and
    // 1 dummy byte
    SQI1XCON1 = 0x002C2EAA;

    //Mode code set to 0, Mode bytes set to 0, CS1 enabled
    SQI1XCON2 = 0x00000400;

    SQI1CFG = 0x8220901B;           // Configure SQI for PIO mode

    // Access the SQI flash through memory mapped address (256 bytes)
    for (readLoop = 0; readLoop < 64; readLoop++){
        *readBuf++ = *sqiXIPAddr++;
    }
}
```

## 46.7 SQI DATA FLOW MODES

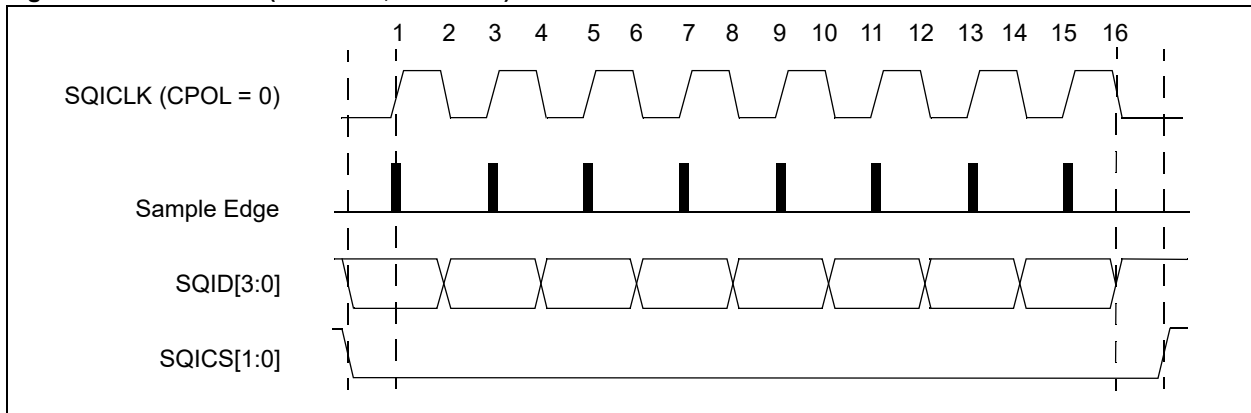
The SQI module is a synchronous SPI-compatible serial port, which can operate in typical SPI modes 0 and 3 (specified by the CPOL bit (SQI1CFG[4]) and the CPHA bit (SQI1CFG[3])).

### 46.7.1 Mode 0 and Mode 3

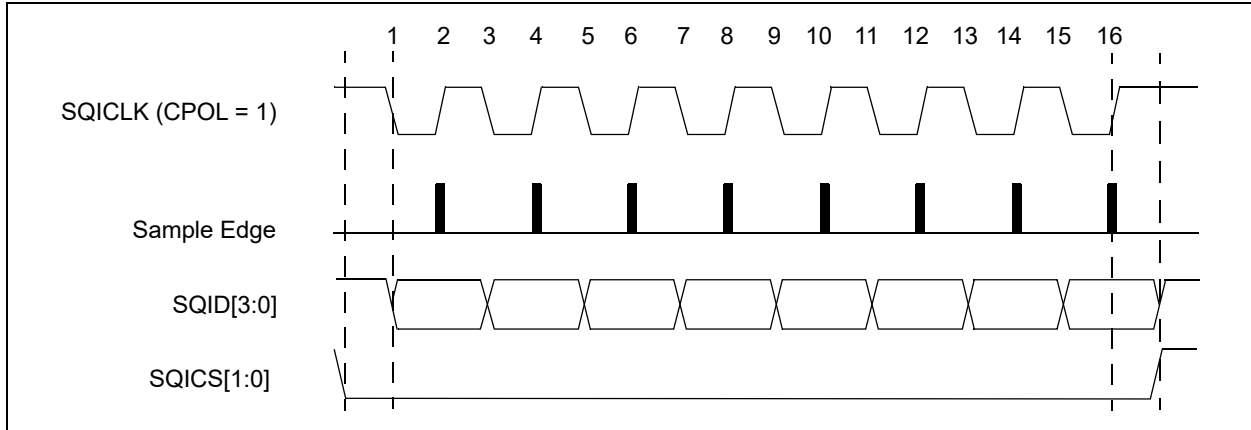
Mode 0 and Mode 3 are typical SPI modes of operation, which are differentiated by the CPOL and CPHA bit settings. When CPOL and CPHA are set to '0', the SQI module operates in Mode 0. When these two bits are set to '1', the SQI module operates in Mode 3.

As shown in Figure 46-13 and Figure 46-14, the primary difference between Mode 0 and Mode 3 concerns the state of SQI clock when the SQI controller is in Idle mode (i.e., no transfers are in progress). In Mode 0, the SQI clock stays low during Idle mode and in Mode 3, it stays high (provides a better clock edge entering active mode). In Mode 0, the SQI clock is held low at the start and the end of the SQI transfer cycle, whereas in Mode 3, the SQI clock is held high at the start and end of the transfer cycle. In both modes, the SQI data input is sampled on the rising edge of the SQI clock, and the SQI data output is clocked on the falling edge of the SQI clock.

**Figure 46-13: Mode 0 (CPHA = 0, CPOL = 0)**



**Figure 46-14: Mode 3 (CPHA = 1, CPOL = 1)**



## 46.8 FLASH INSTRUCTIONS AND SEQUENCE DIAGRAMS QUICK REFERENCE

This section provides a quick reference to a few general Flash instructions and sequence diagrams using the Microchip SST26VF016/031 Quad Flash device. See the “*Serial Quad I/O (SQI) Flash Memory SST26VF016 / SST26VF032 Data Sheet*” (DS20005017) for additional details.

Table 46-3 describes the set of Flash commands implemented by the SST26VF016/032 SQI Flash device.

**Table 46-3: Quad Flash Memory Sample Instructions**

Instruction	Description	Command Byte	Number of Address Bytes	Number of Dummy Bytes	Number of Data Bytes
RSTEN	Reset Enable	0x66	0	0	0
RST <sup>(1)</sup>	Reset Memory	0x99	0	0	0
EQIO	Enable Quad I/O	0x38	0	0	0
RSTEQIO <sup>(2)</sup>	Reset Quad I/O	0xFF	0	0	0
Read <sup>(3)</sup>	Read Memory	0x03	3	0	≥ 1
High-Speed Read/Fast Read <sup>(3)</sup>	Read Memory at Higher Speed	0x0B	3	1	≥ 1
JEDEC-ID <sup>(3,4)</sup>	Read JEDEC ID	0x9F	0	0	3 to infinity ≥ 3

- Note 1:** The RST command is only executed if the RSTEN command is executed first. Any intervening command will disable Reset.
- 2:** The Device accepts eight-clock command in SPI mode, or two-clock command in SQI mode.
- 3:** After a power cycle, Read, High-Speed Read, and JEDEC-ID Read instructions input and output cycles are SPI bus protocol.
- 4:** The Quad J-ID read wraps the three Quad J-ID Bytes of data until terminated by a low-to-high transition on the SQICS0/SQICS1 pins.

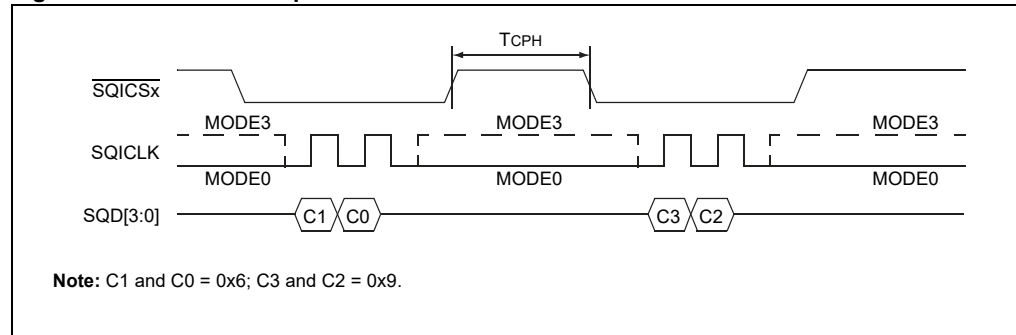
## 46.8.1 Instruction Sequence Diagrams

### 46.8.1.1 RESET-ENABLE (RSTEN) AND RESET (RST)

The Reset operation is used as a system software reset of the Flash device. Reset operation consists of two commands: Reset Enable and Reset (RST). As an example, to reset the memory, the SQI first drives the  $\overline{\text{SQICS0}}$  pin low (assuming Flash memory is connected to  $\overline{\text{SQICS0}}$ ), sends the Reset Enable command (0x66), and then drives the  $\overline{\text{SQICS0}}$  pin high. Next, the SQI drives the  $\overline{\text{SQICS0}}$  pin low again, sends the Reset command (0x99), and then drives the  $\overline{\text{SQICS0}}$  pin high, as shown in [Figure 46-15](#).

**Note:** Any command other than the Reset command after the Reset Enable command will disable the Reset Enable.

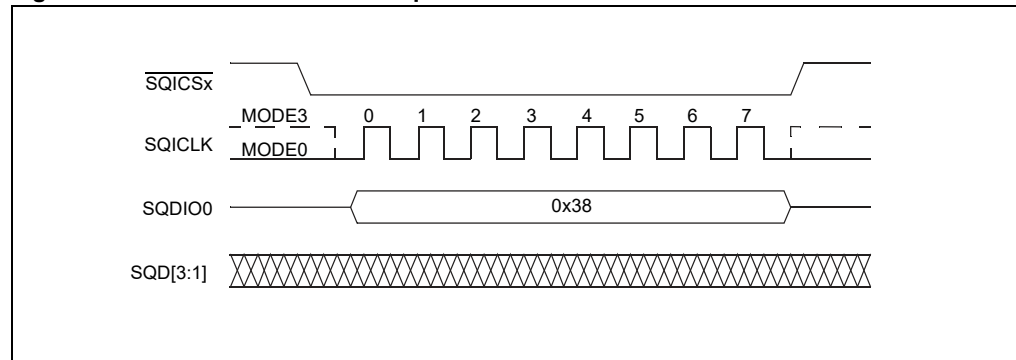
**Figure 46-15: Reset Sequence**



### 46.8.1.2 ENABLE QUAD I/O (EQIO)

The Enable Quad I/O (EQIO) instruction (0x38) enables the Flash Memory for SQI bus operation. Upon completion of the instruction, all instructions thereafter will be 4-bit multiplexed input/output until a power cycle or a "Reset Quad I/O instruction" is executed. The Reset Quad I/O instruction (0xFF) is executed in the same manner as Enable Quad I/O. Refer to [Figure 46-16](#).

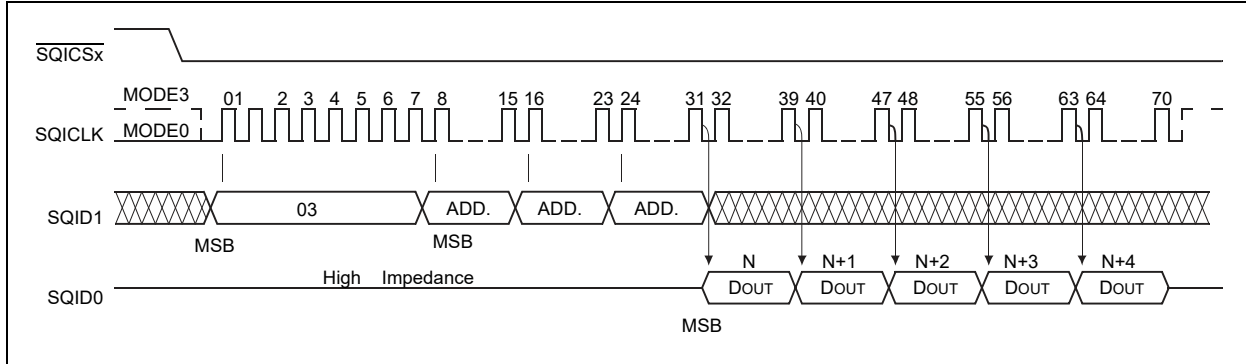
**Figure 46-16: Enable Quad I/O Sequence**



## 46.8.1.3 READ

The Read instruction (0x03) is only supported in SPI bus protocol Modes 0 or 3. This instruction is not supported in Dual/Quad Lane modes. The memory outputs the data starting from the specified address location, and then continuously streams the data output through all addresses until terminated by a low-to-high transition on the  $\overline{\text{SQICS0}}$  or  $\overline{\text{SQICS1}}$  pin. The Read instruction is initiated by executing an 8-bit command (0x03), followed by a 24-bit address, as shown in Figure 46-17. Chip Select will remain active-low for the duration of the Read cycle.

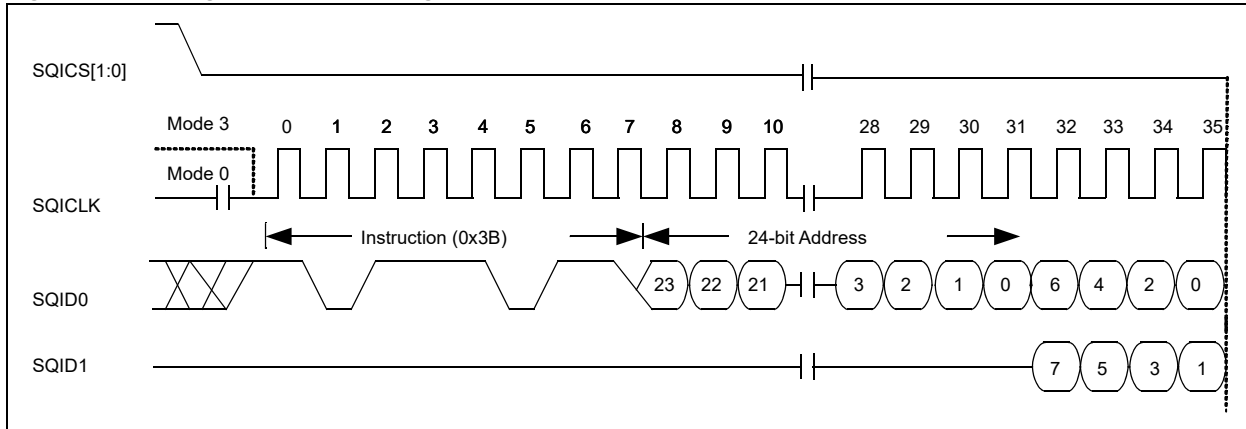
**Figure 46-17: Read Sequence**



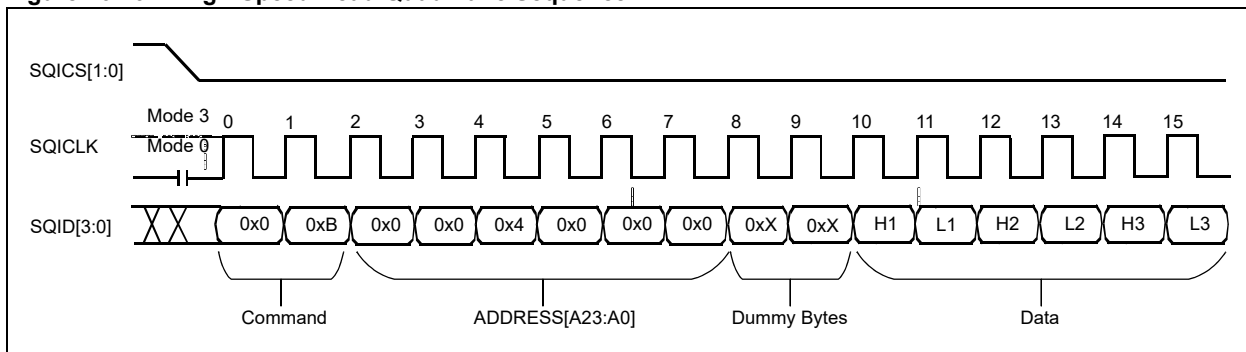
## 46.8.1.4 HIGH-SPEED/FAST QUAD READ

The High-Speed Read (0x0B) instruction is supported in both the SPI bus protocol and the SQI bus protocol (Dual/Quad Lane mode). Figure 46-18 shows the timing diagram of the Quad Lane High Speed Read instruction. To execute this instruction with the SST16VF series Quad Flash, the SQI module would first issue an Enable Quad I/O instruction (refer to 46.8.1.2 “Enable Quad I/O (EQIO)”) followed by a High Speed Read instruction (0x0B).

**Figure 46-18: High-Speed Read Single Lane Sequence**



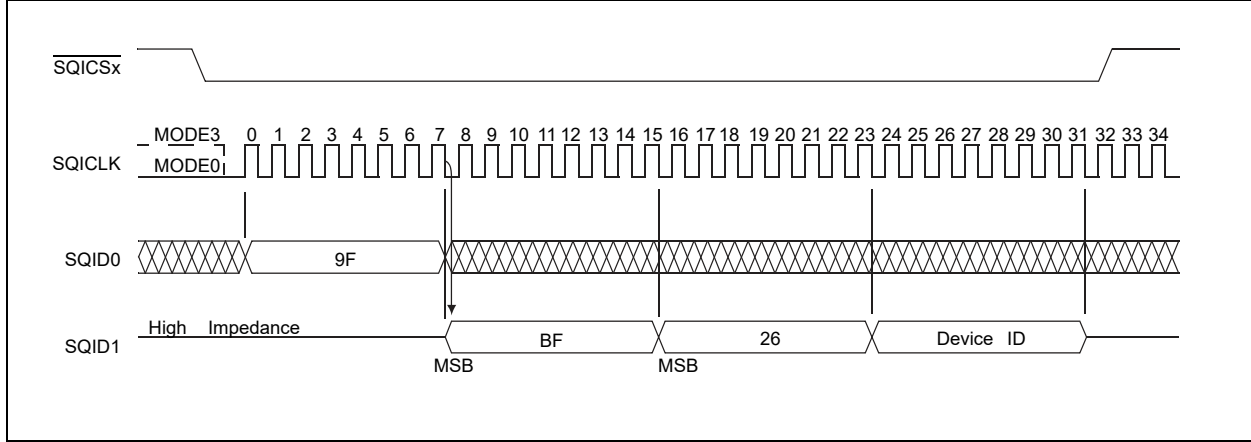
**Figure 46-19: High-Speed Read Quad Lane Sequence**



## 46.8.1.5 JEDEC-ID READ

The JEDEC-ID Read instruction reads the manufacturer and device identifications, and is executed by driving the SQICSt pin low (assuming the Flash is connected to SQICSt), and then sending the JEDEC-ID instruction (0x9F). The memory outputs the device ID (up to 3 bytes), immediately after the instruction, as shown in Figure 46-20.

**Figure 46-20: Device ID Read Sequence – Microchip SST Flash**



## 46.9 EFFECTS OF RESET

### 46.9.1 Device Reset ( $\overline{\text{MCLR}}$ )

All SQI registers are forced to their reset states upon a device Reset. When the asynchronous reset input goes active, the SQI module resets:

- All read/write bits in all registers (SQI1XCON1, SQI1XCON2, SQI1CFG, SQI1CON, etc.)
- The transmit, receive, and control buffers to the empty state
- The SQI1TXDATA and SQI1RXDATA registers
- The SQI1CLKCON register, setting  $T_{SQI}$  to  $T_{BC}/2$

### 46.9.2 Software Reset

A software reset is achieved through the RESET bit of the SQI1CFG register. In this case, the SQI module behaves as if a device Reset has occurred and performs the same reset actions that are described in [46.9.1 “Device Reset \(MCLR\)”](#).

### 46.9.3 Power-on Reset

All of the SQI control registers are forced to their reset states when a Power-on Reset occurs.

### 46.9.4 Watchdog Timer Reset

During a Watchdog Timer reset, the SQI module behaves as if a device Reset has occurred and performs the same reset actions that are described in [46.9.1 “Device Reset \(MCLR\)”](#).

## 46.10 OPERATION IN POWER-SAVING MODES

### 46.10.1 Sleep Mode

When the device enters Sleep mode, the SQI module is disabled and placed into a low-power state where no clocking occurs in the module. Depending on the device, the state of the SQI module is either preserved or reset when the device exits Sleep mode. Refer to the “**Power-Saving Features**” chapter in the specific device data sheet for details.

### 46.10.2 Idle Mode

When the device enters Idle mode, the SQI module clocks still operate and the peripheral is functional. However, as the CPU is halted, the SQI module operation is limited to certain functions. BD DMA transfers should run uninterrupted when the device is in Idle mode.

### 46.10.3 Debug Mode

The behavior of the SQI module is unaltered in Debug mode.

## 46.11 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC32 device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Serial Quad Interface (SQI) module are:

Title	Application Note #
No related application notes at this time.	N/A

**Note:** Please visit the Microchip web site ([www.microchip.com](http://www.microchip.com)) for additional application notes and code examples for the PIC32 family of devices.

## 46.12 REVISION HISTORY

### Revision A (November 2013)

This is the initial released version of this document.

### Revision B (June 2014)

This revision includes the following updates:

- The Preliminary status was removed
- The SERMODE bit was eliminated (see [Table 46-1](#))
- The DEASSERT Buffer Descriptor Control Word was added (see [Figure 46-5](#))
- The following figures were added:
  - [Figure 46-10: “DMA Mode Flowchart”](#)
  - [Figure 46-11: “PIO Mode Flowchart”](#)
  - [Figure 46-12: “XIP Mode Flowchart”](#)
- The following code examples were added:
  - [Example 46-3: “DMA Mode Configuration Code”](#)
  - [Example 46-4: “PIO Mode Configuration Code”](#)
  - [Example 46-5: “XIP Mode Configuration Code”](#)
- Minor updates to text and formatting were incorporated throughout the document

### Revision C (May 2015)

This revision includes the following updates:

- Extensive updates to [Table 46-1: “Serial Quadrature Interface \(SQI\) Register Map”](#)
- Extensive updates to all registers ([Register 46-1](#) through [Register 46-26](#))
- [46.5 “SQI Operation”](#) was added
- Minor updates to text and formatting were incorporated throughout the document

### Revision D (September 2020)

This revision includes the following updates:

- Note related to non availability of eXecute-In-place (XIP) feature is added in following sections and register tables:
  - [Section 46.2 “SQI Module Features”](#)
  - [Section 46.3 “Functional Block Description”](#)
  - [Register 46-1](#)
  - [Register 46-2](#)
  - [Register 46-25](#)
  - [Register 46-26](#)
  - [Section 46.6.3 “XIP Mode\(\)”](#)

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

### Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTracker, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Minda, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQR, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2020, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-5224-6556-0

For information regarding Microchip's Quality Management Systems, please visit [www.microchip.com/quality](http://www.microchip.com/quality).



# MICROCHIP

## Worldwide Sales and Service

### AMERICAS

**Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://www.microchip.com/support>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

#### Atlanta

Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

#### Austin, TX

Tel: 512-257-3370

#### Boston

Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

#### Chicago

Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

#### Dallas

Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

#### Detroit

Novi, MI  
Tel: 248-848-4000

#### Houston, TX

Tel: 281-894-5983

#### Indianapolis

Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453  
Tel: 317-536-2380

#### Los Angeles

Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608  
Tel: 951-273-7800

#### Raleigh, NC

Tel: 919-844-7510

#### New York, NY

Tel: 631-435-6000

#### San Jose, CA

Tel: 408-735-9110  
Tel: 408-436-4270

#### Canada - Toronto

Tel: 905-695-1980  
Fax: 905-695-2078

### ASIA/PACIFIC

**Australia - Sydney**  
Tel: 61-2-9868-6733

**China - Beijing**  
Tel: 86-10-8569-7000

**China - Chengdu**  
Tel: 86-28-8665-5511

**China - Chongqing**  
Tel: 86-23-8980-9588

**China - Dongguan**  
Tel: 86-769-8702-9880

**China - Guangzhou**  
Tel: 86-20-8755-8029

**China - Hangzhou**  
Tel: 86-571-8792-8115

**China - Hong Kong SAR**  
Tel: 852-2943-5100

**China - Nanjing**  
Tel: 86-25-8473-2460

**China - Qingdao**  
Tel: 86-532-8502-7355

**China - Shanghai**  
Tel: 86-21-3326-8000

**China - Shenyang**  
Tel: 86-24-2334-2829

**China - Shenzhen**  
Tel: 86-755-8864-2200

**China - Suzhou**  
Tel: 86-186-6233-1526

**China - Wuhan**  
Tel: 86-27-5980-5300

**China - Xian**  
Tel: 86-29-8833-7252

**China - Xiamen**  
Tel: 86-592-2388138

**China - Zhuhai**  
Tel: 86-756-3210040

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-3090-4444

**India - New Delhi**  
Tel: 91-11-4160-8631

**India - Pune**  
Tel: 91-20-4121-0141

**Japan - Osaka**  
Tel: 81-6-6152-7160

**Japan - Tokyo**  
Tel: 81-3-6880-3770

**Korea - Daegu**  
Tel: 82-53-744-4301

**Korea - Seoul**  
Tel: 82-2-554-7200

**Malaysia - Kuala Lumpur**  
Tel: 60-3-7651-7906

**Malaysia - Penang**  
Tel: 60-4-227-8870

**Philippines - Manila**  
Tel: 63-2-634-9065

**Singapore**  
Tel: 65-6334-8870

**Taiwan - Hsin Chu**  
Tel: 886-3-577-8366

**Taiwan - Kaohsiung**  
Tel: 886-7-213-7830

**Taiwan - Taipei**  
Tel: 886-2-2508-8600

**Thailand - Bangkok**  
Tel: 66-2-694-1351

**Vietnam - Ho Chi Minh**  
Tel: 84-28-5448-2100

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4485-5910  
Fax: 45-4485-2829

**Finland - Espoo**  
Tel: 358-9-4520-820

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Garching**  
Tel: 49-8931-9700

**Germany - Haan**  
Tel: 49-2129-3766400

**Germany - Heilbronn**  
Tel: 49-7131-72400

**Germany - Karlsruhe**  
Tel: 49-721-625370

**Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Germany - Rosenheim**  
Tel: 49-8031-354-560

**Israel - Ra'anana**  
Tel: 972-9-744-7705

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Italy - Padova**  
Tel: 39-049-7625286

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Norway - Trondheim**  
Tel: 47-7288-4388

**Poland - Warsaw**  
Tel: 48-22-3325737

**Romania - Bucharest**  
Tel: 40-21-407-87-50

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**Sweden - Gothenberg**  
Tel: 46-31-704-60-40

**Sweden - Stockholm**  
Tel: 46-8-5090-4654

**UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820