![MICROCHIP]

Section 20. Data Converter Interface (DCI)

HIGHLIGHTS

This section of the manual contains the following topics:

20

Data Converter Interface (DCI)

20.1   INTRODUCTION

The Data Converter Interface (DCI) module allows simple interfacing between dsPIC33F devices and audio devices, such as audio coder/decoders (codecs), Analog-to-Digital Converters (ADC), and Digital-to-Analog Converters (DAC).

The following interfaces are supported:

• Framed Synchronous Serial Transfer (Single or Multi-Channel)
• Inter-IC Sound (I^2S) Interface
• AC-Link Compliant Mode

Many codecs intended for use in audio applications support sampling rates between 8 kHz and 48 kHz and use one of the interface protocols listed above. The DCI automatically handles the interface timing associated with these codecs. No overhead from the CPU is required until the requested amount of data has been transmitted and/or received by the DCI module.

The data word length for the DCI is programmable up to 16 bits to match the data size of the audio application. However, many codecs have data word sizes greater than 16 bits. The DCI can support long data word lengths. The DCI is configured to transmit or receive the long word in multiple 16-bit time slots. This operation is transparent to the user application. The long data word is stored in consecutive register locations.

The DCI can support up to 16 time slots in a data frame, for a maximum frame size of 256 bits. Control bits for each time slot in the data frame determine whether the DCI transmits or receives during the time slot.

The dsPIC33F DMA module allows for direct transfer of data between the dual port SRAM and DCI transmit and receive registers.

20.2   CONTROL REGISTER DESCRIPTIONS

The DCI has five Control registers and one Status register:

• **DCICON1: Data Converter Interface Module Control Register 1**
  This register controls the DCI module enable and mode bits.
• **DCICON2: Data Converter Interface Module Control Register 2**
  This register controls the DCI module word length, data frame length and buffer setup.
• **DCICON3: Data Converter Interface Module Control Register 3**
  This register controls the DCI module bit clock generator setup.
• **DCISTAT: Data Converter Interface Module Status Register**
  This register provides the DCI module status information.
• **RSCON: Receive Slot Enable Register**
  This register enables the active frame time slot control for data reception.
• **TSCON: Transmit Slot Enable Register**
  This register enables the active frame time slot control for data transmission.

In addition to these Control and Status registers, there are four transmit registers, TXBUF0 through TXBUF3, and four receive registers, RXBUF0 through RXBUF3.

Register 20-1:   DCICON1: Data Converter Interface Module Control Register 1

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
DCIEN	—	DCISIDL	—	DLOOP	CSCKD	CSCKE	COFSD
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
UNFM	CSDOM	DJST	—	—	—	COFSM<1:0>	
bit 7							bit 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 15   **DCIEN:** DCI Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14   **Reserved:** Read as '0'

bit 13   **DCISIDL:** DCI Stop in Idle Control bit
1 = Module halts in CPU Idle mode
0 = Module continues to operate in CPU Idle mode

bit 12   **Reserved:** Read as '0'

bit 11   **DLOOP:** Digital Loopback Mode Control bit
1 = Digital Loopback mode is enabled, CSDI and CSDO pins internally connected
0 = Digital Loopback mode is disabled

bit 10   **CSCKD:** Sample Clock Direction Control bit
1 = CSCK pin is an input when DCI module is enabled
0 = CSCK pin is an output when DCI module is enabled

bit 9   **CSCKE:** Sample Clock Edge Control bit
1 = Data changes on serial clock falling edge, sampled on serial clock rising edge
0 = Data changes on serial clock rising edge, sampled on serial clock falling edge

bit 8   **COFSD:** Frame Synchronization Direction Control bit
1 = COFS pin is an input when DCI module is enabled
0 = COFS pin is an output when DCI module is enabled

bit 7   **UNFM:** Underflow Mode bit
1 = Transmit last value written to the transmit registers on a transmit underflow
0 = Transmit '0's on a transmit underflow

bit 6   **CSDOM:** Serial Data Output Mode bit
1 = CSDO pin is tri-stated during disabled transmit time slots
0 = CSDO pin drives '0's during disabled transmit time slots

bit 5   **DJST:** DCI Data Justification Control bit
1 = Data transmission/reception begins during the same serial clock cycle as the frame
    synchronization pulse
0 = Data transmission/reception begins one serial clock cycle after frame synchronization pulse

bit 4-2   **Reserved:** Read as '0'

bit 1-0   **COFSM<1:0>:** Frame Sync Mode bits
11 = 20-bit AC-Link mode
10 = 16-bit AC-Link mode
01 = I^{2}S Frame Sync mode
00 = Multi-Channel Frame Sync mode

20

**Data Converter
Interface (DCI)**

Register 20-2: **DCICON2: Data Converter Interface Module Control Register 2**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0
—	—	—	—	BLEN<1:0>		—	COFSG3
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
COFSG<2:0>			—	WS<3:0>			
bit 7							bit 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 15-12 **Reserved:** Read as '0'

bit 11-10 **BLEN<1:0>:** Buffer Length control bits
11 = Four data words are buffered between interrupts
10 = Three data words are buffered between interrupts
01 = Two data words are buffered between interrupts
00 = One data word is buffered between interrupts

bit 9 **Reserved:** Read as '0'

bit 8-5 **COFSG<3:0>:** Frame Sync Generator control bits
1111 = Data frame has 16 words
•
•
•
0010 = Data frame has 3 words
0001 = Data frame has 2 words
0000 = Data frame has 1 word

bit 4 **Reserved:** Read as '0'

bit 3-0 **WS<3:0>:** DCI Data Word Size bits
1111 = Data word size is 16 bits
•
•
•
0100 = Data word size is 5 bits
0011 = Data word size is 4 bits
0010 = Invalid Selection. Do not use, unexpected results may occur
0001 = Invalid Selection. Do not use, unexpected results may occur
0000 = Invalid Selection. Do not use, unexpected results may occur

Register 20-3:   DCICON3: Data Converter Interface Module Control Register 3

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	BCG<11:8>			
bit 15							bit 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
BCG<7:0>							
bit 7							bit 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 15-12   **Reserved:** Read as '0'

bit 11-0   **BCG<11:0>:** DCI Bit Clock Generator Control bits

Register 20-4: DCISTAT: Data Converter Interface Module Status Register

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	SLOT<3:0>			
bit 15							bit 8

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	ROV	RFUL	TUNF	TMPTY
bit 7							bit 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 15-12 **Reserved:** Read as '0'

bit 11-8 **SLOT<3:0>:** DCI Slot Status bits
1111 = Slot 15 is currently active
•
•
•
0010 = Slot 2 is currently active
0001 = Slot 1 is currently active
0000 = Slot 0 is currently active

bit 7-4 **Reserved:** Read as '0'

bit 3 **ROV:** Receive Overflow Status bit
1 = A receive overflow has occurred for at least one receive register
0 = A receive overflow has not occurred

bit 2 **RFUL:** Receive Buffer Full Status bit
1 = New data is available in the receive registers
0 = The receive registers have old data

bit 1 **TUNF:** Transmit Buffer Underflow Status bit
1 = A transmit underflow has occurred for at least one transmit register
0 = A transmit underflow has not occurred

bit 0 **TMPTY:** Transmit Buffer Empty Status bit
1 = The transmit registers are empty
0 = The transmit registers are not empty

Register 20-5: RSCON: Receive Slot Enable Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RSE15	RSE14	RSE13	RSE12	RSE11	RSE10	RSE9	RSE8
bit 15							bit 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
RSE7	RSE6	RSE5	RSE4	RSE3	RSE2	RSE1	RSE0
bit 7							bit 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 15-0 **RSE<15:0>:** Receive Slot Enable bits
1 = CSDI data is received during the individual time slot 15
0 = CSDI data is ignored during the individual time slot 15
-
-
-
1 = CSDI data is received during the individual time slot 0
0 = CSDI data is ignored during the individual time slot 0

Register 20-6: TSCON: Transmit Slot Enable Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSE15	TSE14	TSE13	TSE12	TSE11	TSE10	TSE9	TSE8
bit 15							bit 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
TSE7	TSE6	TSE5	TSE4	TSE3	TSE2	TSE1	TSE0
bit 7							bit 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 15-0 **TSE<15:0>:** Transmit Slot Enable Control bits
1 = Transmit buffer contents are sent during the individual time slot 15
0 = CSDO pin is tri-stated or driven to '0' during the individual time slot 15, depending on the state of the CSDOM bit
-
-
-
1 = Transmit buffer contents are sent during the individual time slot 0
0 = CSDO pin is tri-stated or driven to '0' during the individual time slot 0, depending on the state of the CSDOM bit

20

**Data Converter Interface (DCI)**

20.3 CODEC INTERFACE BASICS AND TERMINOLOGY

At a minimum, every codec application has a controller and a codec device. The interface protocols supported by the DCI require the use of a Frame Synchronization (FS) signal (COFS on dsPIC33F devices) to initiate a data transfer between the two devices. In most cases, the rising edge of FS starts a new data transfer. Either device can produce FS. The device that generates FS is the master device. Conceptually, the master device is not required to be the transmitting or receiving device.

Figure 20-1 shows connection examples. The frequency of the FS signal is usually the system sampling rate, fs.

> **Note:** The details given in this section are not specific to the DCI module. This discussion provides some background and terminology related to the digital serial interface protocols found in most codec devices.

Figure 20-1:   Codec Connection Example

Note: Codec oscillator circuit generates the SCK signal.
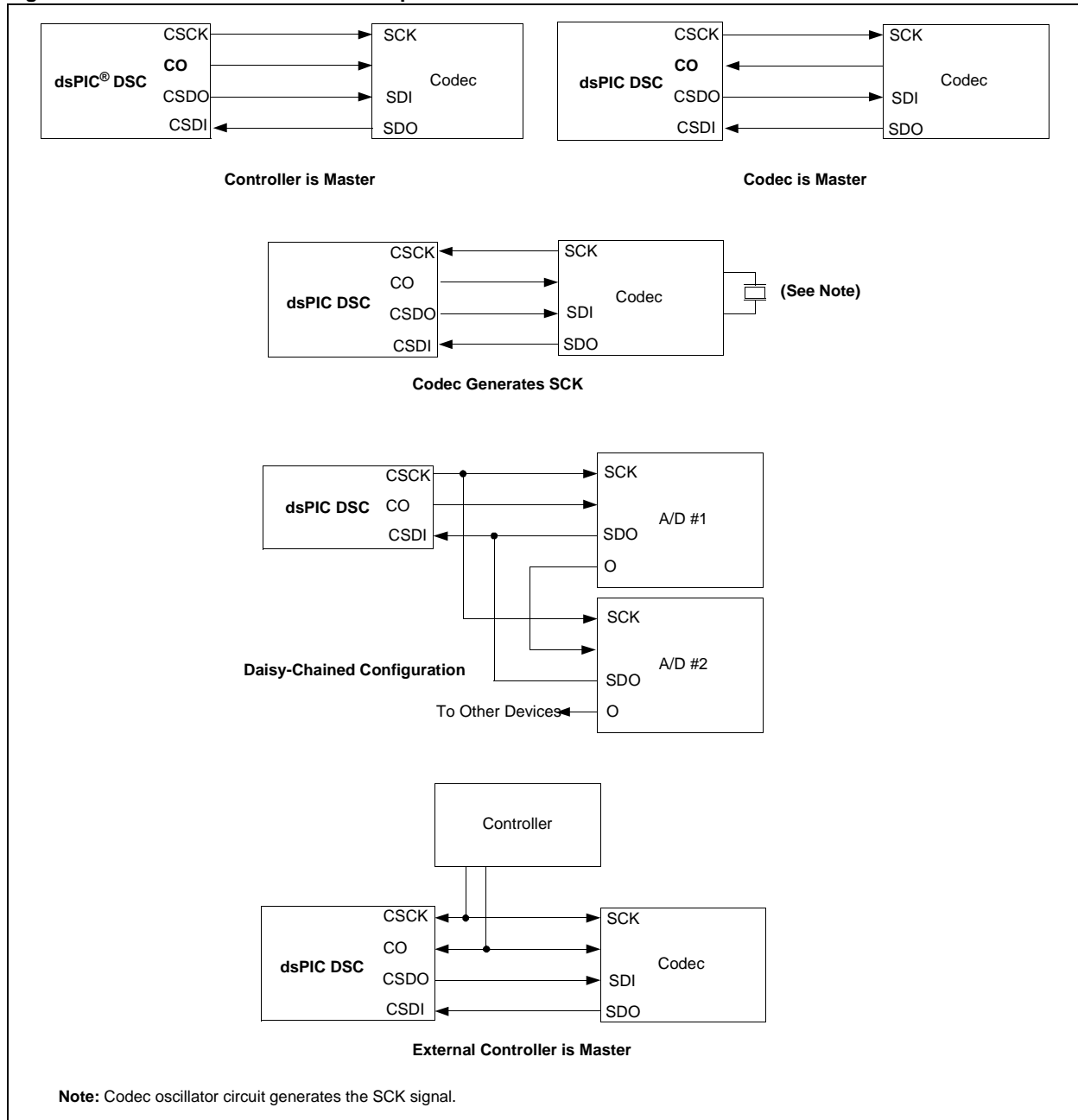
20.3.1   Serial Transfer Clock

All interfaces have a serial transfer clock, SCK (CSCK pin on dsPIC33F devices). The SCK signal can be generated by any of the connected devices or can be provided externally. In some systems, SCK is also referred to as the bit clock. For codecs that offer high signal fidelity, it is common for the SCK signal to be derived from the crystal oscillator on the codec device. The protocol defines the edge of SCK from which data is sampled. The master device generates the FS signal with respect to SCK.

The period of the FS signal delineates one data frame. This period is the same as the data sample period. The number of SCK cycles that occur during the data frame depends on the type of codec selected. The ratio of the SCK frequency to the system sample rate is expressed as a ratio of n, where n is the number of SCK periods per data frame.

20.3.2   Data Transfer and Time Slots

Data is transferred via the Serial Data Out (SDO) and Serial Data Input (SDI) signals (the CSDO and CSDI pins on dsPIC33F devices). One advantage of using a framed interface protocol is that multiple data words can be transferred during each sample period, or data frame. For example, consider a 16-bit codec with four input channels. The codec would need to transmit four 16-bit words within one FS period. This results in 64 SCK cycles per FS period and n = 64.

Time slots can be used for multiple codec data channels or control information. Furthermore, multiple devices can be multiplexed on the same serial data pins. Each slave device is programmed to place its data on the serial data connection during the proper time slot. The output of each slave device is tri-stated at all other times to permit other devices to use the serial bus.

Some devices allow the FS signal to be daisy-chained via Frame Synchronization Output (FSO) pins. Figure 20-1 shows a typical daisy-chained configuration. When the transfer from the first slave device is complete, a FS pulse is sent to the second device in the chain via its FSO pin. This process continues until the last device in the chain sends data. The controller (master) device should be programmed for a data frame size that accommodates the largest of the data words to be transferred.

20.3.2.1   DATA TRANSFER TIMING

Figure 20-2 shows the timing for a typical data transfer. Most protocols begin the data transfer one SCK cycle after the FS signal is detected. This example uses a 16-fs clock (fs is the sampling frequency) and transfers four 4-bit data words per frame.

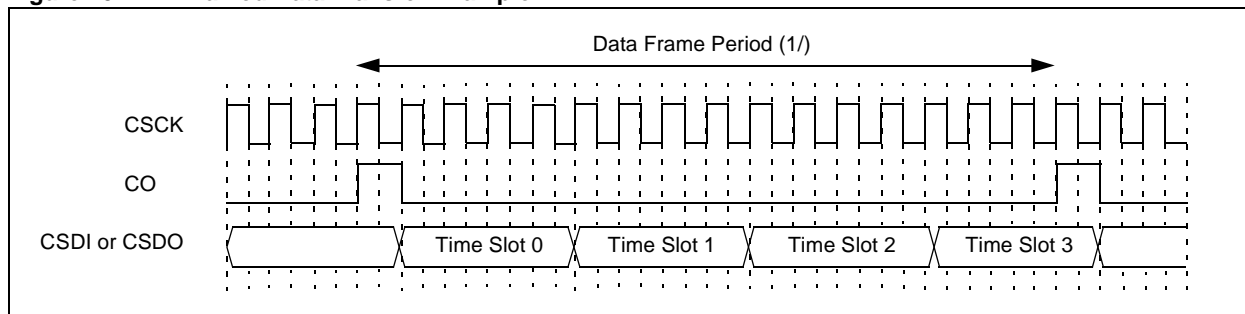**Figure 20-2:   Framed Data Transfer Example**

Figure 20-3 shows the timing for a typical data transfer with daisy-chained devices. This example uses a 16-fs SCK frequency and transfers two 8-bit data words per frame. After the FS pulse is detected, the first device in the chain transfers the first 8-bit data word and generates the FSO signal at the end of the transfer. The FSO signal begins the transfer of the second data word from the second device in the chain.

Figure 20-3:   Daisy-Chained Data Transfer Example
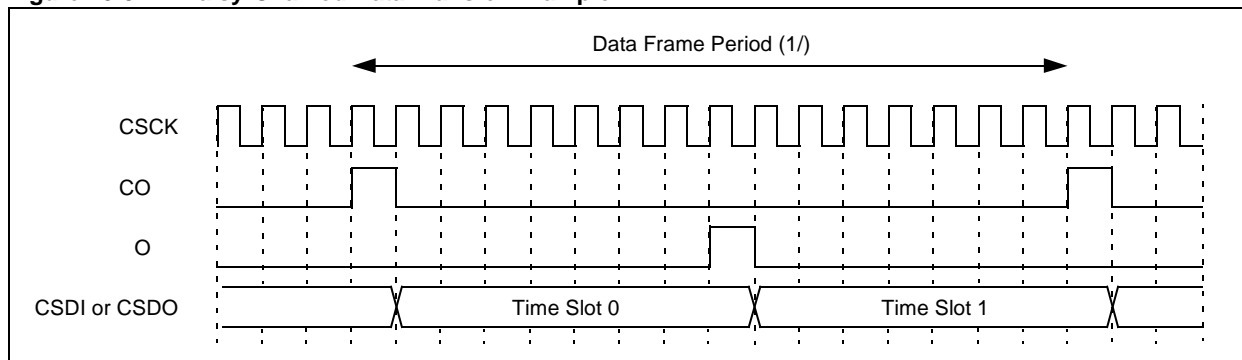
20.3.3   FS Pulse

The FS pulse has a minimum active time of one SCK period so that the slave device can detect the start of the data frame. The duty cycle of the FS pulse can vary depending on the specific protocol used to mark certain boundaries in the data frame.

As an example, the I^{2}S protocol uses a FS signal that has a 50% duty cycle. The I^{2}S protocol is optimized for the transfer of two data channels (left and right channel audio information). The edges of the FS signal mark the boundaries of the left- and right-channel data words.

As another example, the AC-Link protocol uses a FS signal that is high for 16 SCK periods and low for 240 SCK periods. The edges of the AC-Link FS signal mark the boundaries of control information and data in the frame.

20.4 DCI OPERATION

Figure 20-4 shows a simplified block diagram of the DCI module. The module consists of a transmit/receive shift register connected to a small range of memory buffers via a buffer control unit. This arrangement allows the DCI to support various codec serial protocols. The DCI shift register is 16 bits wide. Data is transmitted and received by the DCI Most Significant bit (MSb) first.

Figure 20-4:   DCI Module Block Diagram

20

**Data Converter
Interface (DCI)**
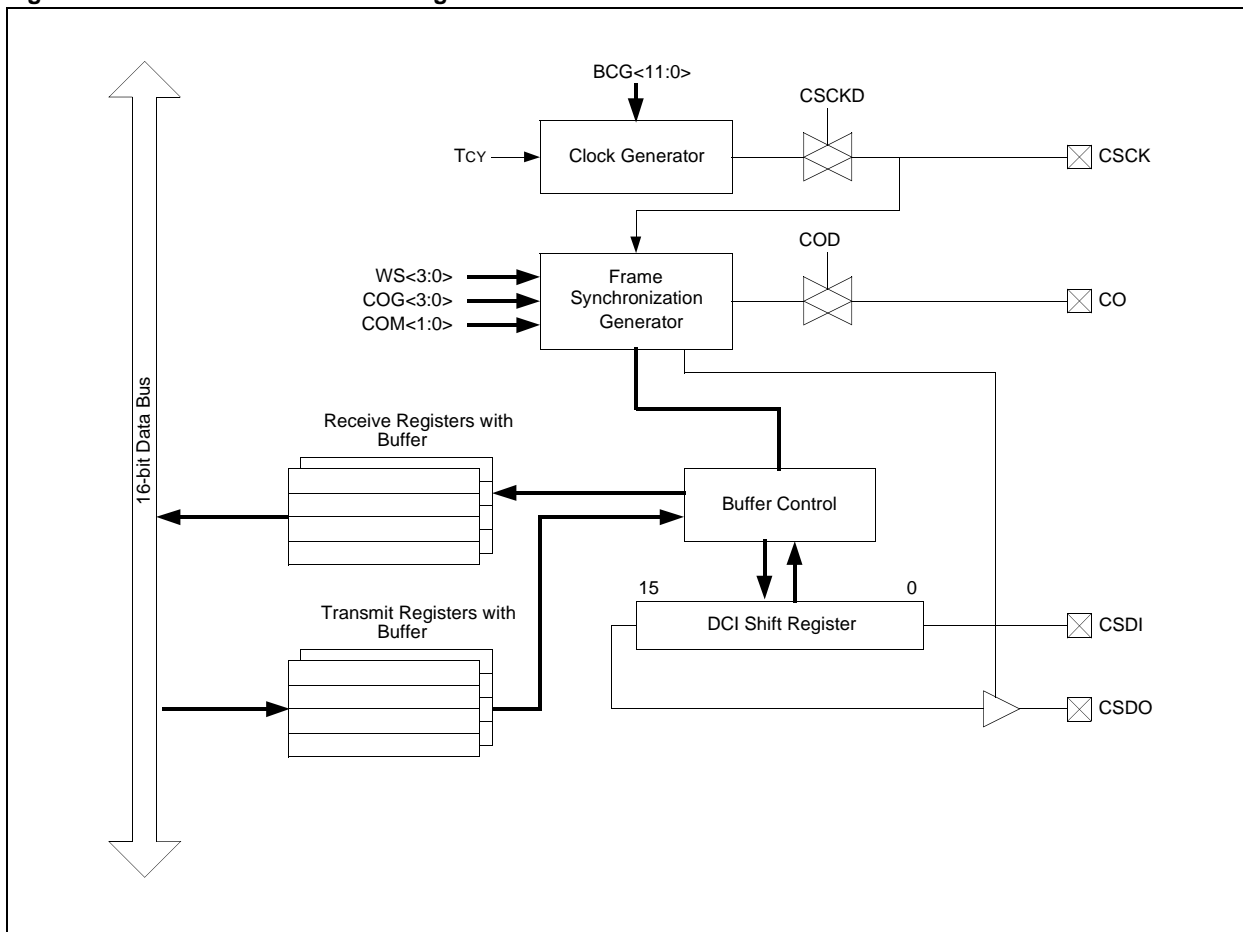
20.4.1 DCI Pins

Four I/O pins (CSCK, CSDO, CSDI and COFS) are associated with the DCI module. The DCI module, when enabled, controls the data direction of each of the four pins.

20.4.1.1 CSCK PIN

The CSCK pin provides the serial clock connection for the DCI. The CSCK pin can be configured as an input or output using the CSCKD control bit (DCICON1<10>).

- When the CSCK pin is configured as an output (CSCKD = 0), the serial clock is derived from the dsPIC33F system clock source and supplied to external devices by the DCI
- When the CSCK pin is configured as an input (CSCKD = 1), the serial clock must be provided by an external device

20.4.1.2 CSDO PIN

The Serial Data Output (CSDO) pin is configured as an output-only pin when the module is enabled. The CSDO pin drives the serial bus whenever data is to be transmitted. The CSDO pin can be tri-stated or driven to '0' during serial clock periods when data is not transmitted, depending on the state of the Serial Data Output Mode (CSDOM) control bit (DCICON1<6>). The tri-state option allows other devices to be multiplexed onto the CSDO connection.

20.4.1.3 CSDI PIN

The Serial Data Input (CSDI) pin is configured as an input-only pin when the module is enabled.

20.4.1.4 COFS PIN

The Frame Synchronization (COFS) pin is used to synchronize data transfers that occur on the CSDO and CSDI pins. The COFS pin is bidirectional and can be configured as an input or an output. The data direction for the COFS pin is determined by the COFSD control bit (DCICON1<8>):

- When the COFSD bit is cleared, the COFS pin is an output. The DCI module generates FS pulses to initiate a data transfer. The DCI is the master device for this configuration.
- When the COFSD bit is set, the COFS pin becomes an input. Incoming synchronization signals to the module initiate data transfers. The DCI is a slave device when the COFSD control bit is set.

20.4.2 Module Enable

The DCI module is enabled or disabled by setting or clearing the DCI Module Enable (DCIEN) control bit (DCICON1<15>). Clearing the DCIEN control bit resets the module. All counters associated with serial clock generation, FS and the buffer control logic are reset. For additional information, refer to **20.5.1.1 "DCI Start-up and Data Buffering"** and **20.5.1.2 "DCI Disable"**.

When enabled, the DCI controls the data direction for the CSCK, CSDI, CSDO and COFS I/O pins associated with the module. The PORT, LAT and TRIS register values for these I/O pins are overridden by the DCI module when the DCIEN bit (DCICON1<15>) is set.

20.4.3 Bit Clock Generator

The DCI module has a dedicated 12-bit time base that produces the bit clock. The bit clock rate (period) is set by writing a non-zero 12-bit value to the DCI Bit Clock Generator (BCG) control bits (DCICON3<11:0>). When the BCG bits are set to zero, the bit clock is disabled.

> **Note:** The CSCK I/O pin is controlled by the DCI module if the DCIEN bit is set **OR** the Bit Clock Generator is enabled by writing a non-zero value to DCICON3<11:0>. This allows the BCG to be operated independently of the DCI module.

When the CSCK pin is controlled by the DCI module, the corresponding PORT, LAT and TRIS control register values for the CSCK pin are overridden and the data direction for the CSCK pin is controlled by the CSCKD control bit (DCICON1<10>).

- If the serial clock for the DCI is provided by an external device, set the BCG bits (DCICON3<11:0>) to '0' and the CSCKD bit to '1'
- If the serial clock is generated by the DCI module, set the BCG control bits (DCICON3<11:0>) to a non-zero value (refer to Equation 20-1) and set the CSCKD control bit (DCICON1<10>) to '0'

Equation 20-1 provides the formula for the bit clock frequency.

Equation 20-1: DCI Bit Clock Generator Value

$$BCG<11:0> = \frac{F_{CY}}{2\ F_{CSCK}} - 1$$

The required bit clock frequency is determined by the system sampling rate and frame size. Typical bit clock frequencies range from 16 to 512 times the converter sample rate, depending on the data converter and the communication protocol used.

As an example, consider a dsPIC33F device running at 40 MIPS. The DCI module is required to interface with a 16-bit codec, which is configured for a sampling rate of 8 kHz. Therefore, the FS Period = 1/8 kHz = 125 microseconds.

The codec sends two 16-bit words in every frame and the frame occurs at the sampling frequency. Two 16-bit words in a frame requires the bit period to be (125 microseconds/(2 x 16)) = 3.960625 microseconds. Therefore, the clock frequency for this codec is FCSCK = (1/3.960625 microseconds) = 256 kHz.

The Bit Clock Generator (BCG) value for the DCI module using Equation 20-1 is BCG = [40000000/(2 x 256000)] – 1 = 77.

20.4.4 Sample Clock Edge Selection

The Sample Clock Edge (CSCKE) control bit (DCICON1<9>) determines the sampling edge for the serial clock signal.

- If the CSCKE bit is cleared (default), data is sampled on the falling edge of the CSCK signal. The AC-Link protocols and most multi-channel formats require that data be sampled on the falling edge of the CSCK signal
- If the CSCKE bit is set, data is sampled on the rising edge of CSCK. The I^2S protocol requires that data is sampled on the rising edge of the serial clock signal

20

Data Converter Interface (DCI)

20.4.5   Frame Synchronization Mode Control Bits

The type of interface protocol supported by the DCI is selected using the FS mode (COFSM) control bits (DCICON1<1:0>). Table 20-1 provides a selection of operating modes.

Table 20-1:   Operating Modes

Mode	DCICON1<1:0> Value	For Information, See Section...
Multi-channel	00	**20.5.4 "Multi-Channel Operation"**
I²S	01	**20.5.5 "I2S Operation"**
AC-Link (16-bit)	10	**20.5.6 "AC-Link Operation"**
AC-Link (20-bit)	11	

20.4.6   Word-Size Selection Bits

The DCI Data Word Size (WS) bits (DCICON2<3:0>) determine the number of bits in each DCI data word. This is the length of each time slot in the frame. Any data length from 4 to 16 bits can be selected. Word size greater than 16 bits can be processed by enabling multiple time slots. For details, refer to **20.5.3 "Data Packing for Long Data Word Support"**.

Note:	The WS control bits are used only in the Multi-Channel and I²S modes. These bits have no effect in AC-Link mode since the data slot sizes are fixed by the protocol.

20.4.7   Frame Synchronization Generator

The Frame Synchronization Generator (FSG) is a 4-bit counter that sets the frame length in data words. The period for the FSG is set by writing the Frame Synchronization Generator (COFSG) control bits (DCICON2<8:5>). Equation 20-2 provides the FSG period (in serial clock cycles) determined by the formula.

Equation 20-2:   Frame Length In CSCK Cycles

$$FrameLength = (WS\langle 3:0\rangle + 1) \bullet (COG\langle 3:0\rangle + 1)$$

A data frame may include time slots during which no data is transferred. As an example, a 16-bit codec requires a control word to be received 16 clock cycles (Time Slot 2) after receiving the 16-bit data word (Time Slot 0). The codec also transmits a data word on its output line in Time Slot 0 (refer to Figure 20-5).

The total frame length is three words or 48 clock cycles (16 clock cycles per word x 3 words). To communicate with this codec, these DCICON register bits must be set as follows:

- Word Size WS (DCICON2<3:0>) = `1111` (16-bit)
- Frame Synchronization Generator COFSG (DCICON2<8:5>) = `0010` (3 words)

Even though no data is transmitted during time slot 1, the frame length must accommodate for the disabled time slot (a time slot during which no data is transmitted or received).

Frame lengths up to 16 data words can be selected. The frame length in serial clock periods vary up to a maximum of 256 depending on the word size selected.

Note:	The COFSG control bits have no effect in AC-Link mode, since the frame length is set to 256 serial clock periods by the protocol.

Figure 20-5:   DCI Timing with WS = `1111` and COFSG = `0010`
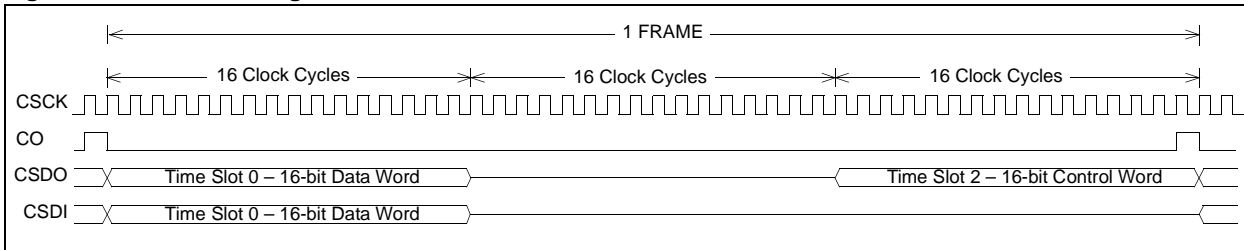
20.4.8 Transmit and Receive Registers

The DCI has four transmit registers, TXBUF0 through TXBUF3, and four receive registers, RXBUF0 through RXBUF3. All of the transmit and receive registers are memory mapped.

20.4.8.1 BUFFER DATA ALIGNMENT

Data values are always stored left justified in the TXBUF and RXBUF registers, since audio PCM data is represented as a signed 2's complement fractional number. If the programmed DCI word size is less than 16 bits, the unused Least Significant bits (LSb) in the receive registers are set to '0' by the module. The module ignores the unused LSbs in the transmit register.

20.4.8.2 TRANSMIT AND RECEIVE BUFFERS

The transmit and receive registers each have a set of buffers that are not accessible by the user software. Effectively, each transmit and receive buffer location is double-buffered. The DCI transmits data from the transmit buffers and writes received data to the receive buffers. The buffers allow the user software to read and write the RXBUF and TXBUF registers, while the DCI uses data from the buffers.

> **Note:** TXBUF0 through TXBUF3 registers are write only registers and should not be read by the user.

20.4.9 DCI Buffer Control Unit

The DCI module contains a buffer control unit that transfers data between the buffer memory and the serial shift register. The buffer control unit also transfers data between the buffer memory and the TXBUF and RXBUF registers. The buffer control unit allows the DCI to queue the transmission and reception of multiple data words without CPU overhead.

The DCI generates an interrupt each time a transfer between the buffer memory and the TXBUF and RXBUF registers takes place. The number of data words buffered between interrupts is determined by the Buffer Length (BLEN) control bits (DCICON2<11:10>). The size of the transmit and receive buffering can vary from 1 to 4 data words using the BLEN control bits.

Each time a data transfer takes place between the DCI shift register and the buffer memory, the DCI buffer control unit is incremented to point to the next buffer location. If the number of transmitted or received data words is equal to the BLEN value + 1, the following occurs:

1. The buffer control unit is reset to point to the first buffer location.
2. The received data held in the receive buffer is transferred to the RXBUF registers.
3. The data in the TXBUF registers is transferred to the buffer.
4. A CPU interrupt is generated.

The DCI buffer control unit always accesses the same relative location in the transmit and receive buffers. For example, if the DCI is transmitting data from TXBUF3, any data received during that time slot is written to RXBUF3.

20

**Data Converter
Interface (DCI)**

Figure 20-6:   DCI Buffer Control Unit
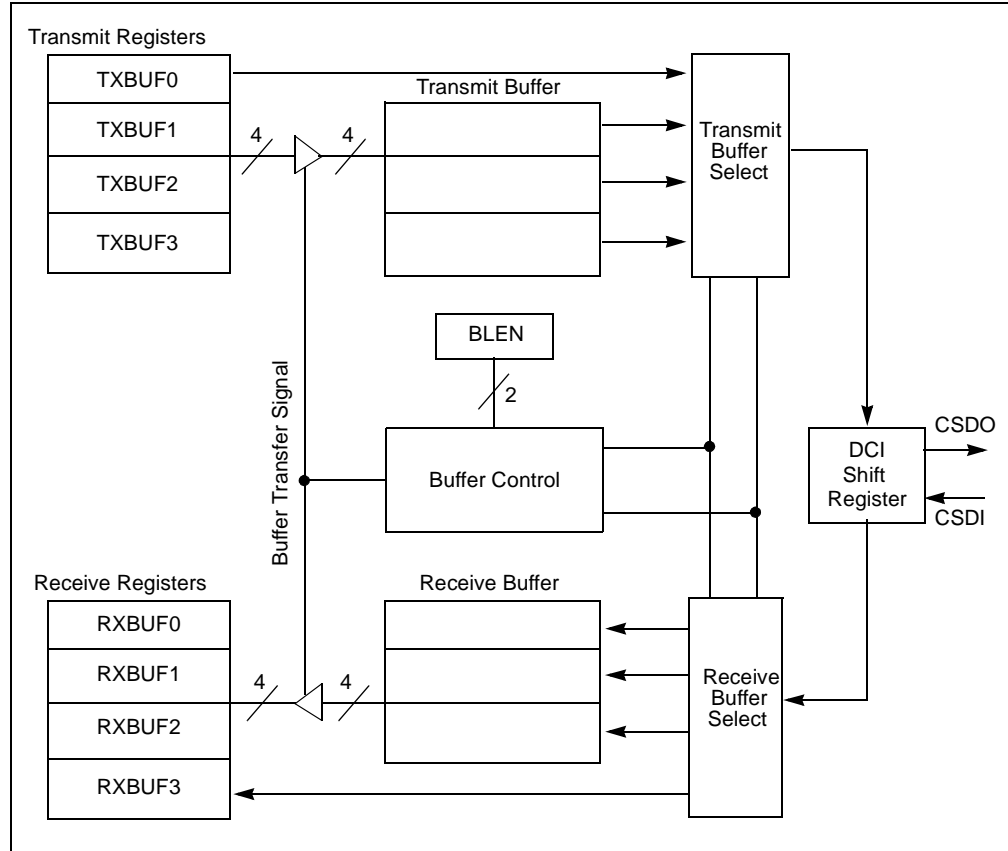
20.4.10   Transmit Slot Enable Bits

The Transmit Slot Enable (TSCON) register has the Transmit Slot Enable (TSE) control bits (TSCON<15:0>) that can enable up to 16 time slots for transmission. The size of each time slot is determined by the DCI Data Word Size (WS) bits (DCICON2<3:0>) and can vary up to 16 bits.

If a transmit time slot is enabled via one of the TSE bits (TSEx = 1), the content of the current transmit buffer location is loaded into the CSDO shift register and the DCI buffer control unit increments to point to the next buffer location. At least one transmit time slot must be enabled for data to be transmitted. If a disabled time slot is encountered, the buffer pointer increments without transmitting the contents of the corresponding TXBUFx register.

Not all TSE control bits affect the module operation if the selected frame size has less than 16 data slots. The Most Significant TSE control bits are not used. For example, if COFSG<3:0> = 0111 (8 data slots per frame), TSE8 through TSE15 have no effect on the DCI operation.

20.4.10.1 CSDO MODE CONTROL

During disabled transmit time slots, the CSDO pin can drive 0's or can be tri-stated, depending on the state of the CSDOM bit (DCICON1<6>). A given transmit time slot is disabled if its corresponding TSEx bit is cleared in the TSCON register.

- If the CSDOM bit (DCICON1<6>) is cleared (default), the CSDO pin drives 0's onto the CSDO pin during disabled time slot periods. This mode is used when there are only two devices (one master and one slave) attached to the serial bus

- If the CSDOM bit (DCICON1<6>) is set, the CSDO pin is tri-stated during unused time slot periods. This mode allows multiple dsPIC33F devices to share the same CSDO line in a multiplexed application. Each device on the CSDO line is configured so that it transmits data only during specific time slots. No two devices should transmit data during the same time slot

20.4.11 Receive Slot Enable Bits

The Receive Slot Enable (RSCON) register contains the Receive Slot Enable (RSE) control bits (RSCON<15:0>) used to enable up to 16 time slots for reception. The size of each receive time slot is determined by the WS control bits (DCICON2<3:0>) and can vary from 4 to 16 bits.

If a receive time slot is enabled via one of the RSE bits (RSEx = 1), the shift register contents are written to the current DCI receive buffer location and the buffer control logic advances to the next available buffer location. At least one receive time slot must be enabled for data to be received. If a disabled time slot is encountered, the buffer pointer increments without receiving the contents of the corresponding RXBUFx register.

Data is not packed in the receive memory buffer locations if the selected word size is less than 16 bits. Each received slot data word is stored in a separate 16-bit buffer location. Data is always stored in a left-justified format in the receive memory buffer. Therefore, if the word size is 8-bit, the received data is stored in bit 15 through bit 8 of the RXBUFx register.

20.4.12 DCI Buffer Control Unit Operation

The DCI module allows read and write operations while it is in the process of transmitting or receiving data. Data is written to the TXBUFx registers and read from the RXBUFx registers. The following shows an example of internal DCI read/write operation, for the case of BLEN = 01 (Buffer length = 2).

Figure 20-7 shows when the DCI module is disabled, no data is received or transmitted.

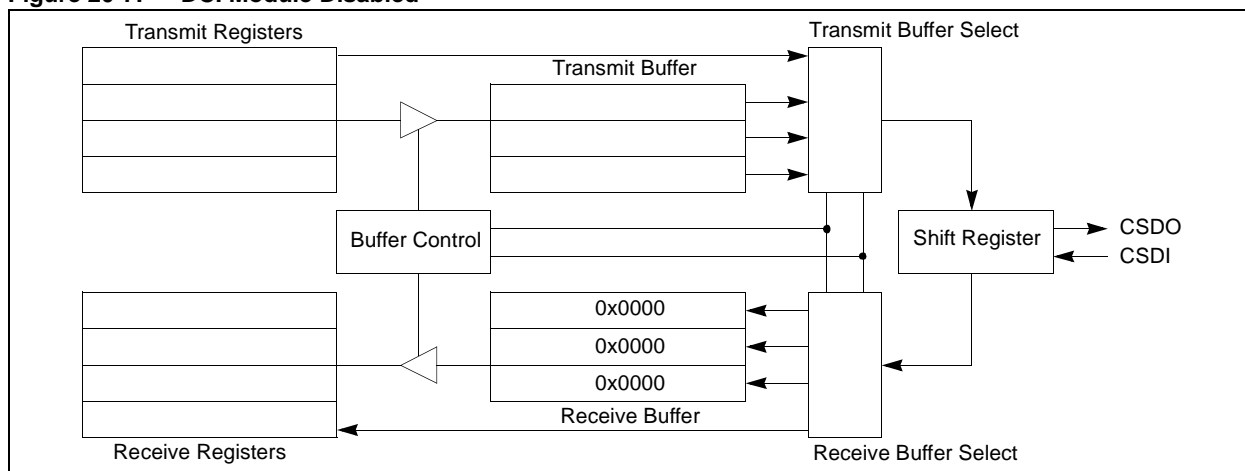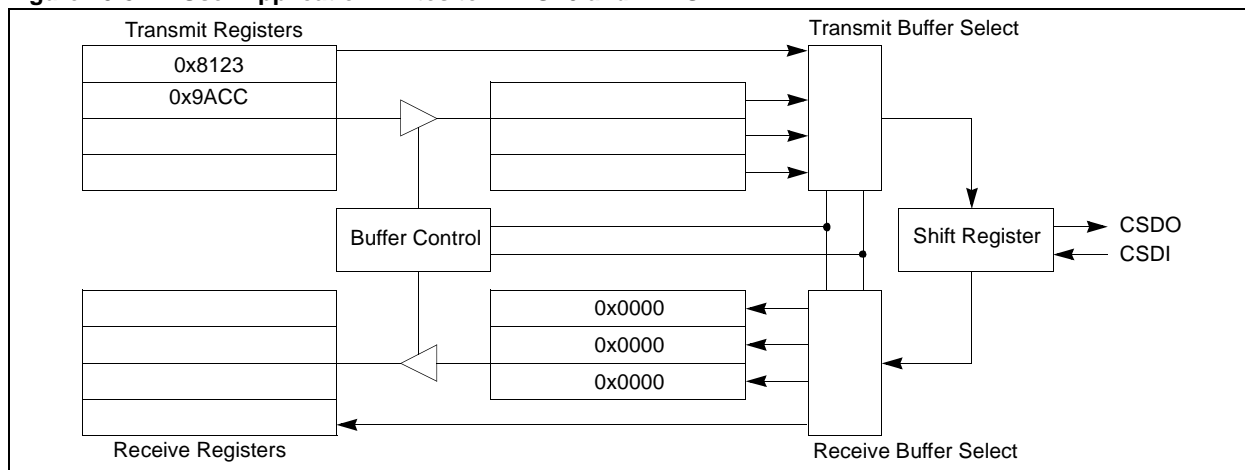**Figure 20-7: DCI Module Disabled**

Figure 20-8 shows the state of the transmit registers after the application has written data to the TXBUF0 and TXBUF1 registers.

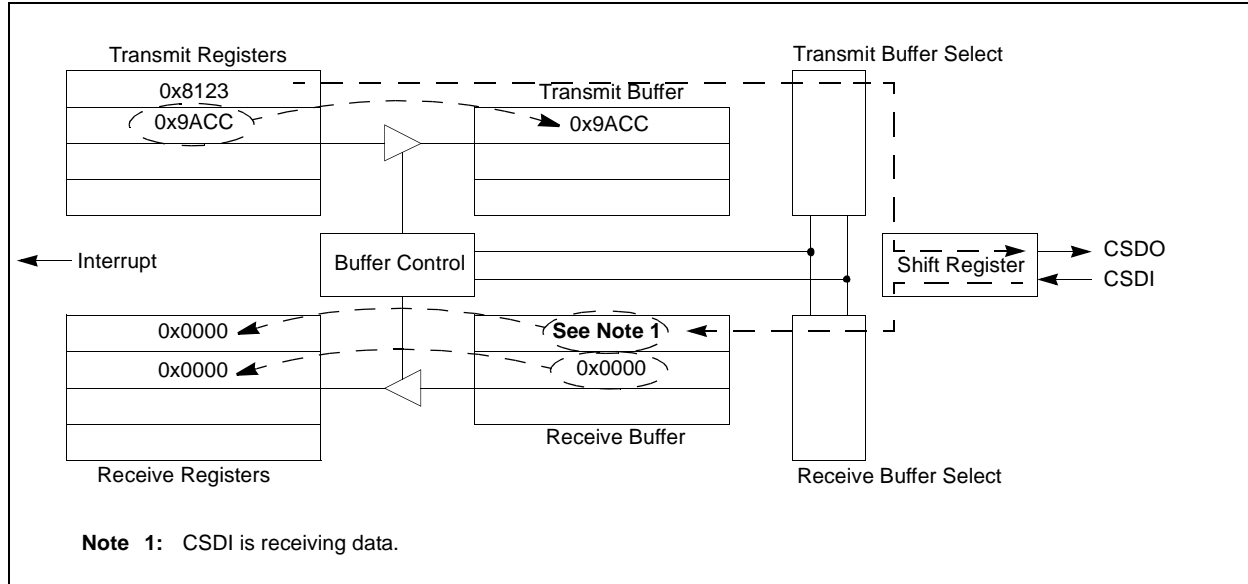**Figure 20-8: User Application Writes to TXBUF0 and TXBUF1**

20

Data Converter Interface (DCI)

When the DCI module is enabled, the CPU receives the DCI interrupt after three clock cycles. When this occurs, data in TXBUF0 shifts to the shift register and data in TXBUF1 is shifted to the transmit Buffer (refer to Figure 20-9). The DCI module will start shifting data out on the CSDO pin.

The contents of the receive buffers are shifted to the RXBUF receive registers. Since no data was received when the module was disabled, these values read as '0'. The RXBUFx registers read '0' until the next interrupt, at which time data from the receive buffers is transferred to these registers. The module will start overwriting data in the receive buffer with data received on the CSDI pin.

Figure 20-9:   DCI Module Enabled

Note 1:   CSDI is receiving data.

The user application writes new data to TXBUF0 and TXBUF1. Note that writing data to the TXBUFx register does not affect the current transmit operation.The second data word is shifted to the shift register. Figure 20-10 shows a new data word is received over the CSDI line into the receive buffers.

Figure 20-10:   User Application Writes to TXBUF0 and TXBUF1, DCI Module Starts Transmitting Second Word

Note 1:   CSDI is receiving data.
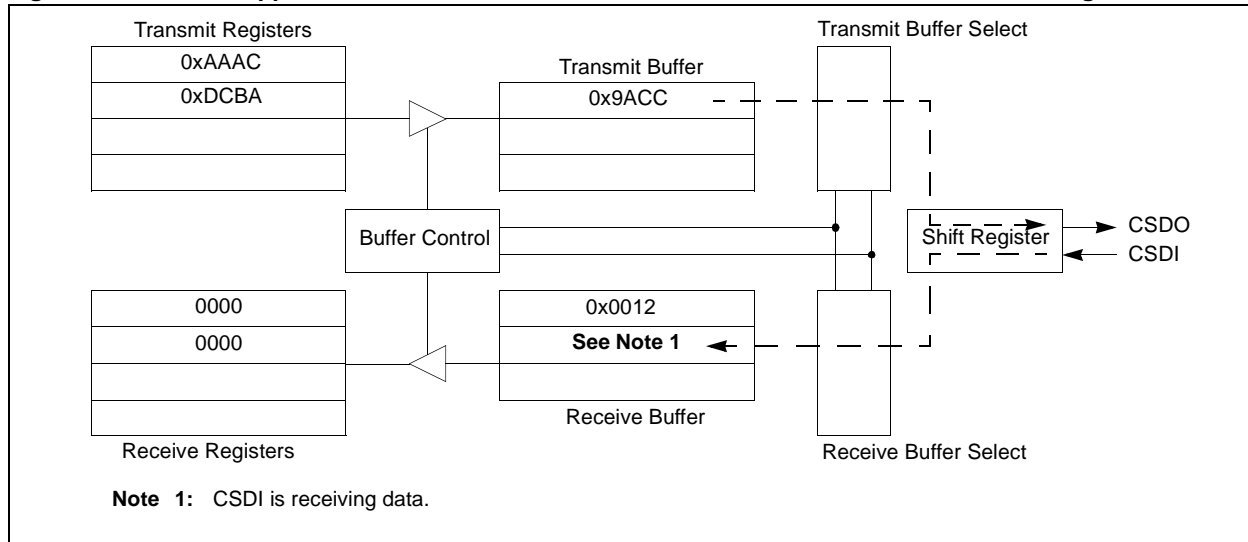
The module has completed transmit/receive operations of BLEN + 1 words, which causes an interrupt. The data in the receive buffers is copied to the RXBUFx registers. Figure 20-11 shows the data in the TXBUF0 is shifted to the shift register and the contents of the TXBUF1 register is copied to the transmit buffer. This cycle repeats with every DCI interrupt.

Figure 20-11: DCI Module Transmit/Receives Two Words

Note 1: 0x0012 data is overwritten by data received on CSDI.
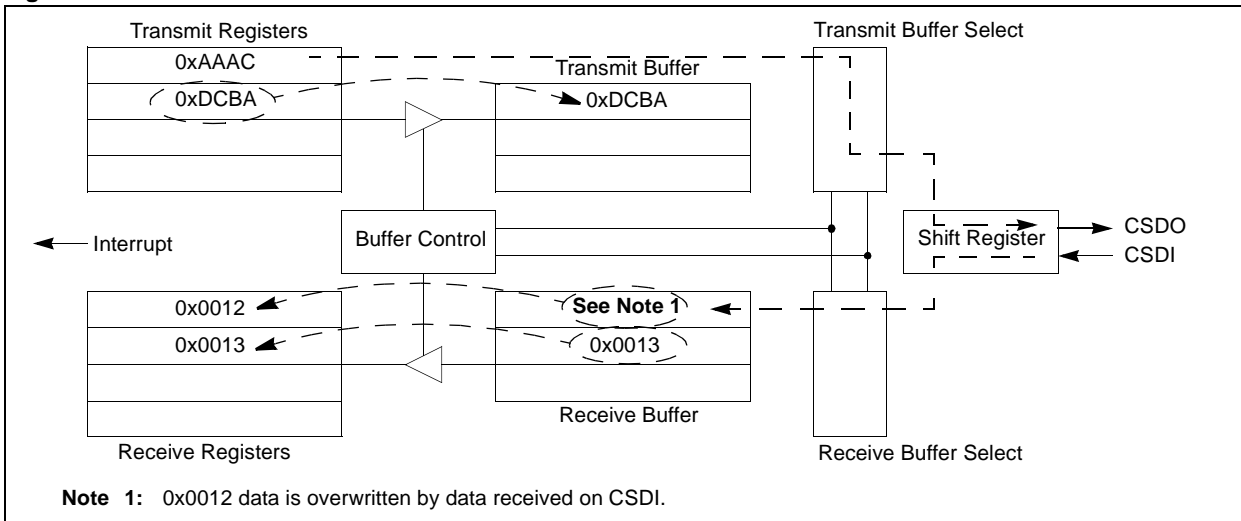
20.4.13   TSCON and RSCON Operation with Buffer Control Unit

The slot enable bits in the TSCON and RSCON registers function independently, with the exception of the buffer control logic. For each time slot in a data frame, the buffer location is advanced if either the TSEx or the R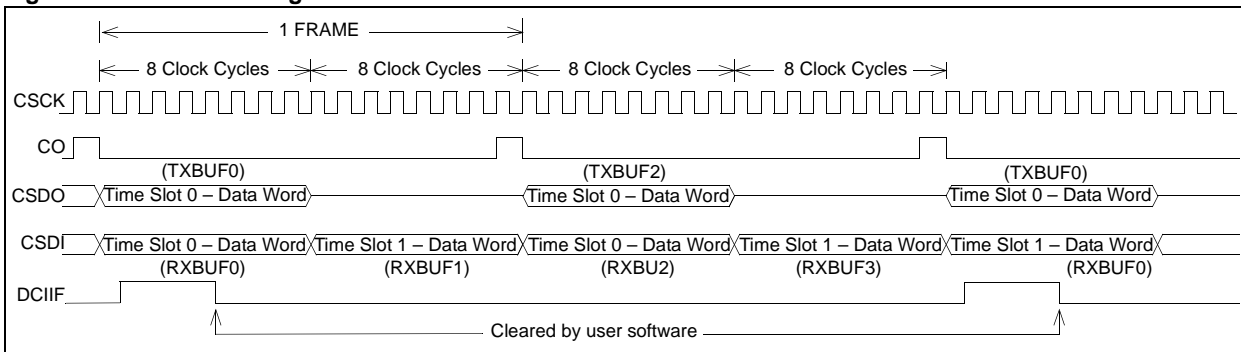SEx bit is set for the current time slot. That is, the buffer control unit synchronizes the transmit and receive buffering, so that the transmit and receive buffer location is always the same for each time slot in the data frame.

If the TSEx bit and the RSEx bit are both set for every time slot used in the data frame, the DCI will transmit and receive equal amounts of data.

In some applications, the number of data words transmitted during a frame may not equal the number of words received. Consider an example where the DCI is configured for a 2-word data frame, with Transmit Slot 0 enabled (TSCON = 0x0001) and Receive Slots 0 and 1 enabled (RSCON = 0x0003). The DCI module is configured to interrupt on four words (BLEN = 11). The frame size is set to 2 data words per frame (COFSG = 0001) and the data word size is set to 8 bits (WS = 0111). Figure 20-12 shows the timing diagram for this example, and Figure 20-13 shows the corresponding DCI buffer operation. This configuration allows the DCI to transmit one data word per frame and receive two data words per frame. Since two data words are received for each data word transmitted, the user software writes to every other transmit buffer location. Specifically, only TXBUF0 and TXBUF2 are used to transmit data.

Figure 20-12: DCI Timing with TSCON = 0x0001 and RSCON = 0x0003

20

Data Converter Interface (DCI)

Figure 20-13:   DCI Buffer Operation: TSCON = 0x0001, RSCON = 0x0003, BLEN<1:0> = 11

	Transmit Registers			Receive Registers	
TXBUF0	Data Word 1 (Time Slot 0)		RXBUF0	Data Word 1 (Time Slot 0)	
TXBUF1	——		RXBUF1	Data Word 2 (Time Slot 1)	
TXBUF2	Data Word 2 (Time Slot 0)		RXBUF2	Data Word 3 (Time Slot 0)	
TXBUF3	——		RXBUF3	Data Word 4 (Time Slot 1)	

Note: User software writes to TXBUF0 and TXBUF2. TXBUF1 and TXBUF3 are not used by transmit logic.
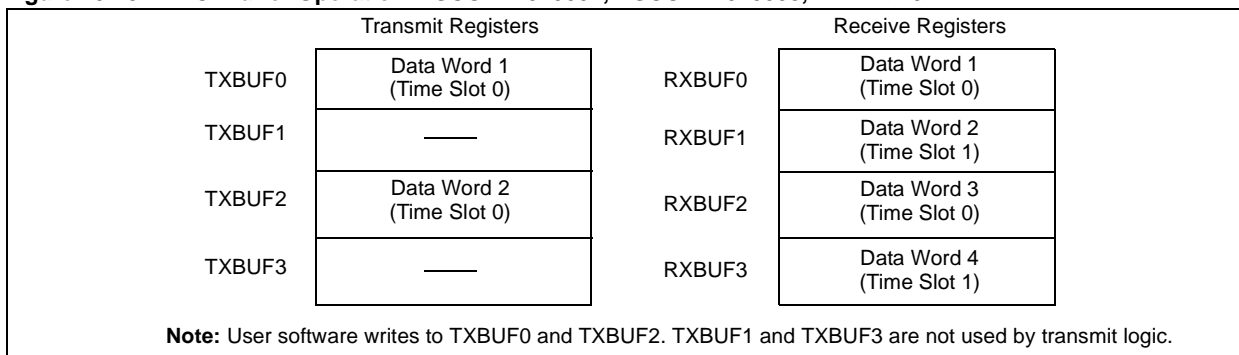
The buffer control resets to point to the first buffer location when BLEN + 1 buffers are written to and a CPU interrupt is generated, or TXBUF3 and RXBUF3 are processed and the buffer pointer must jump to the first buffer location.

The buffer control increments the buffer pointer when all bits in an enabled time slot have been processed, or the bits in the time slot exceed 16 bits. The pointer to the TXBUF increments in synchronization with the RXBUF pointer.

20.4.14   TSCON and RSCON operation with DMA

The DMA module on dsPIC33F devices can be configured to transfer data directly between the dual port SRAM and the DCI TXBUF0 and RXBUF0 registers, without CPU intervention. The BLEN bits (DCICON2<11:10>) should be set to '0' for correct operation. Although the DCI module uses only TXBUF0 and RXBUF0 for operation in this mode, it is still possible to have multi-word frames and multiple time slots. The user application must ensure that data stored in memory corresponds to the enabled time slots.

Figure 20-12 is an example of the DCI codec communication. Here the DCI module has 1 transmit time slot (TS0) enabled, and 2 receive time slots (RS0 and RS1) enabled. The word length is 8 bits (WS = 0111) and the frame size is 2 words (COFSG = 0001). With BLEN = 0, the DCI module requests for a DMA transfer on every word. The DCI module is configured for 8-bit word size and transmits data MSb first. Therefore, the data in DPSRAM should be organized such that the 8-bit data to be transmitted is placed in the Most Significant Byte (MSB) of the 16-bit word. To meet the timing criteria shown in Figure 20-12, the transmit data memory in DPSRAM must additionally be organized such that every other word represents data to be transmitted.

- **Transfer 1**
  The DMA module places the contents of DPSRAM in TXBUF0 and the contents of RXBUF0 into DPSRAM. TXBUF0 and RXBUF0 data corresponds to time slot 0. The DMA pointer will increment. Since the data word size is 8-bits, the received data is stored in the upper 8 bits of the DPSRAM word (refer to Figure 20-14).

- **Transfer 2**
  The DMA module will place the contents of DPSRAM in TXBUF0 and the contents of RXBUF0 into DPSRAM. This data corresponds to time slot 1. Since transmit time slot 1 is disabled, the DCI module will not transmit the data. However, because receive time slot 1 is enabled, the RXBUF0 register will contain data received on CSDI pin (refer to Figure 20-15). The data is placed in DPSRAM and the DMA pointer will increment.

- **Transfer 3**
  Since the frame length is 2 words, the DCI module will assert the COFS signal. The DMA module places the contents of DPSRAM in TXBUF0 and the contents of RXBUF0 into DPSRAM. TXBUF0 and RXBUF0 data corresponds to time slot 0. The DMA pointer will increment. Since the data word size is 8-bits, the received data is stored in the upper 8 bits of the DPSRAM word (refer to Figure 20-16).

- **Transfer 4**
  The DMA module places the contents of DPSRAM in TXBUF0 and the contents of RXBUF0 into DPSRAM. This data corresponds to time slot 1. Since transmit time slot 1 is disabled, the DCI module will not transmit the data. However, because receive time slot 1 is enabled, the RXBUF0 register will contain data received on the CSDI pin (refer to Figure 20-17). The data is placed in DPSRAM and the DMA pointer will increment.
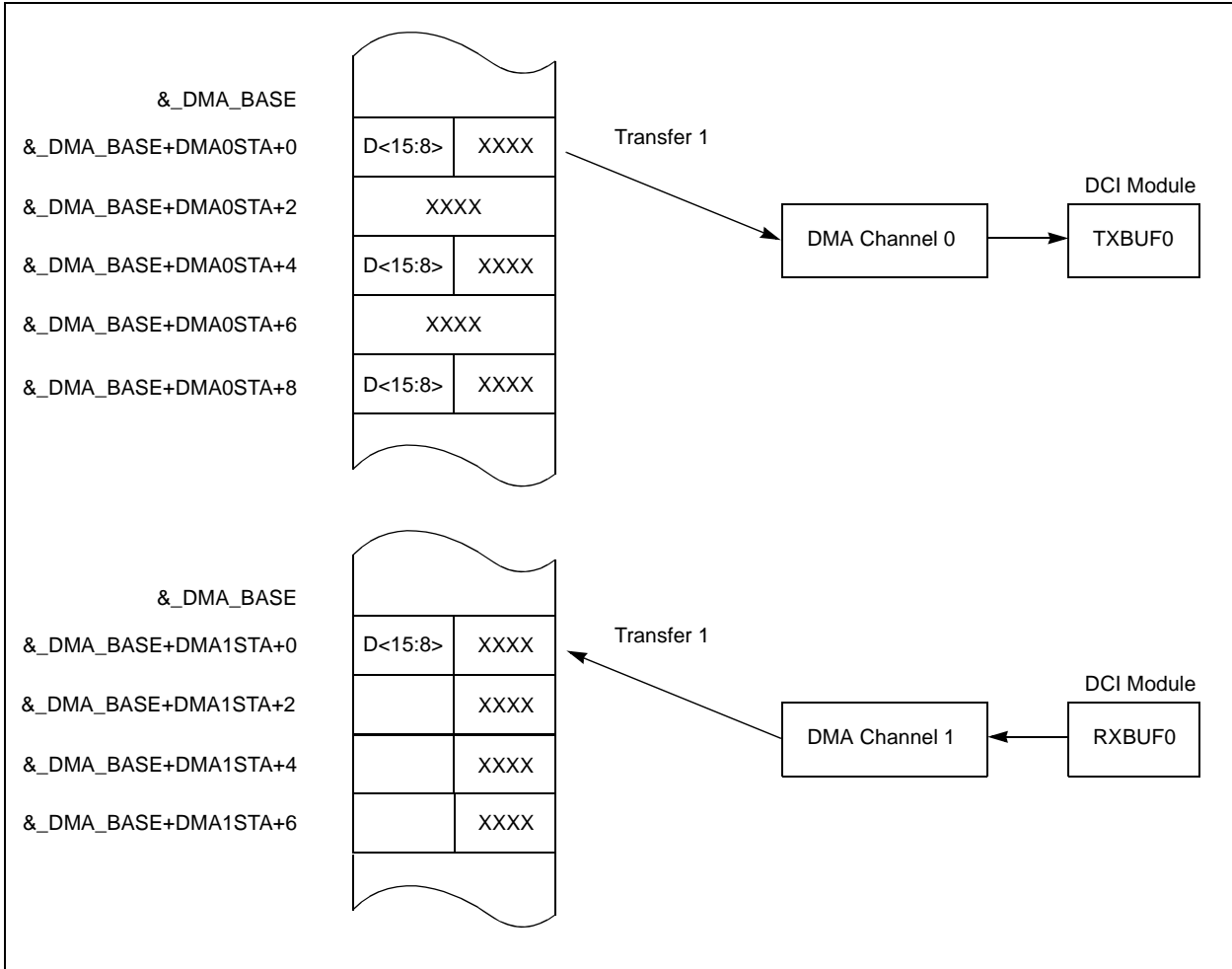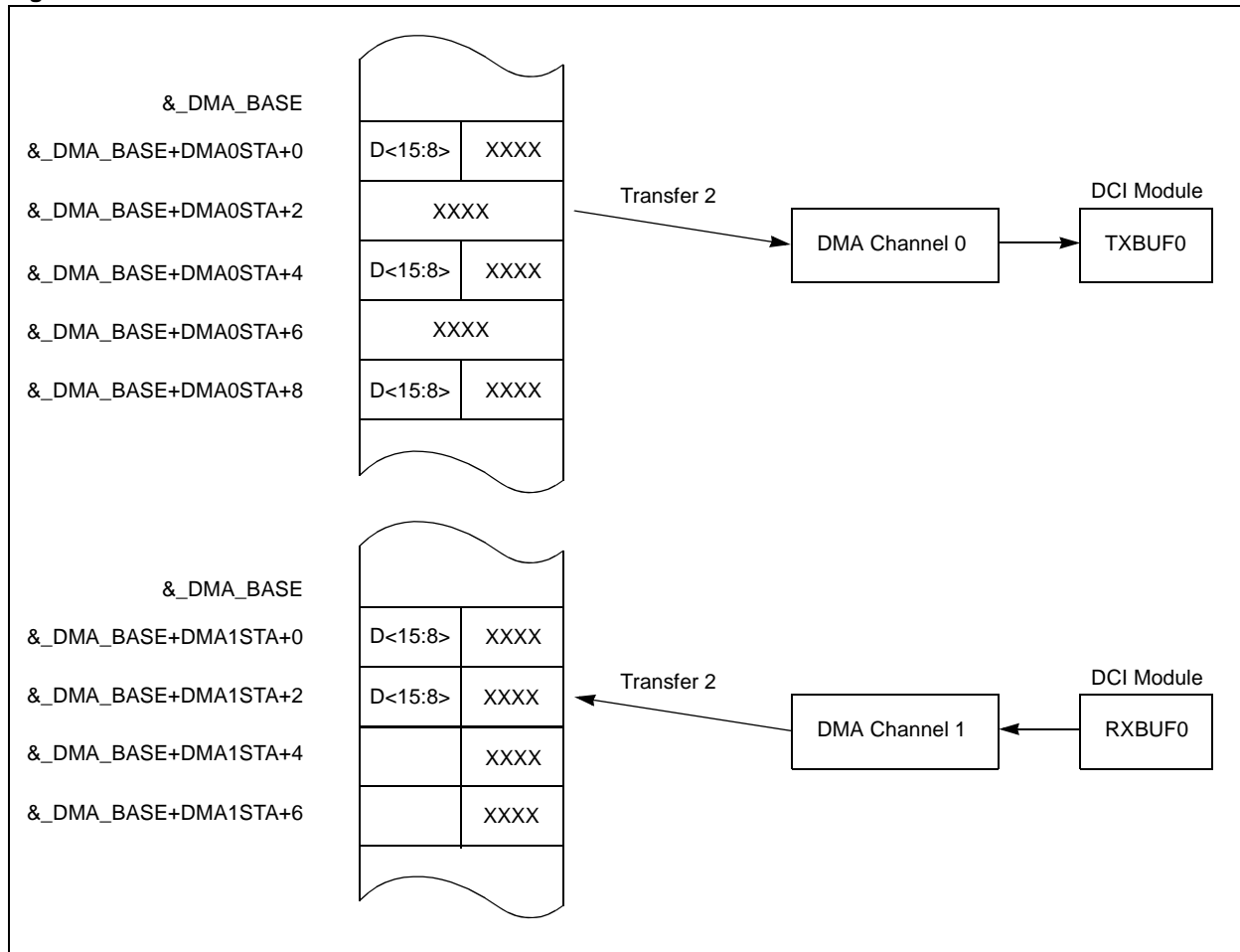
Figure 20-14: Transfer 1: Time Slot 0

Figure 20-15:   Transfer 2: Time Slot 1

Figure 20-16:   Transfer 3: Time Slot 0
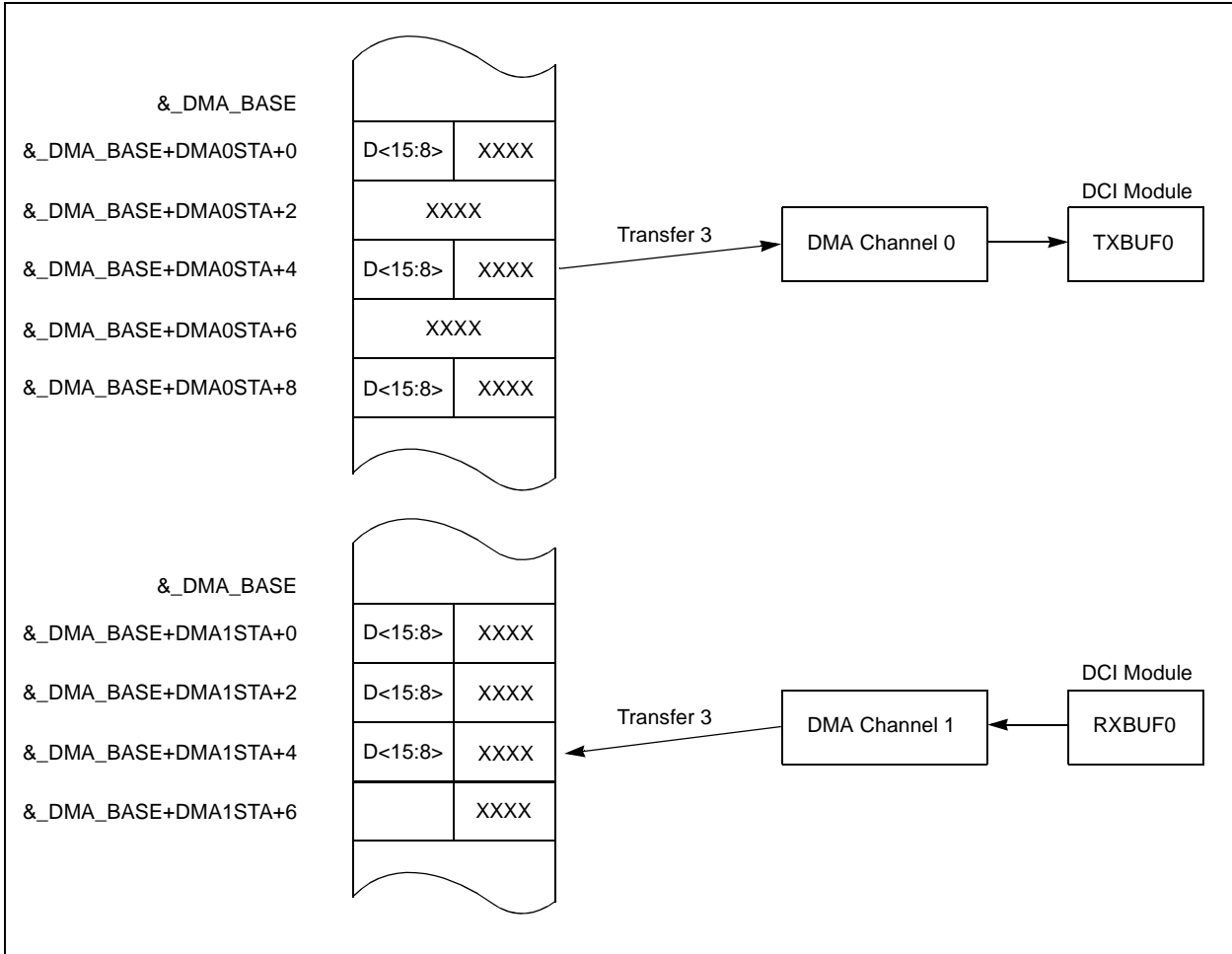
Figure 20-17: Transfer 4: Time Slot 1
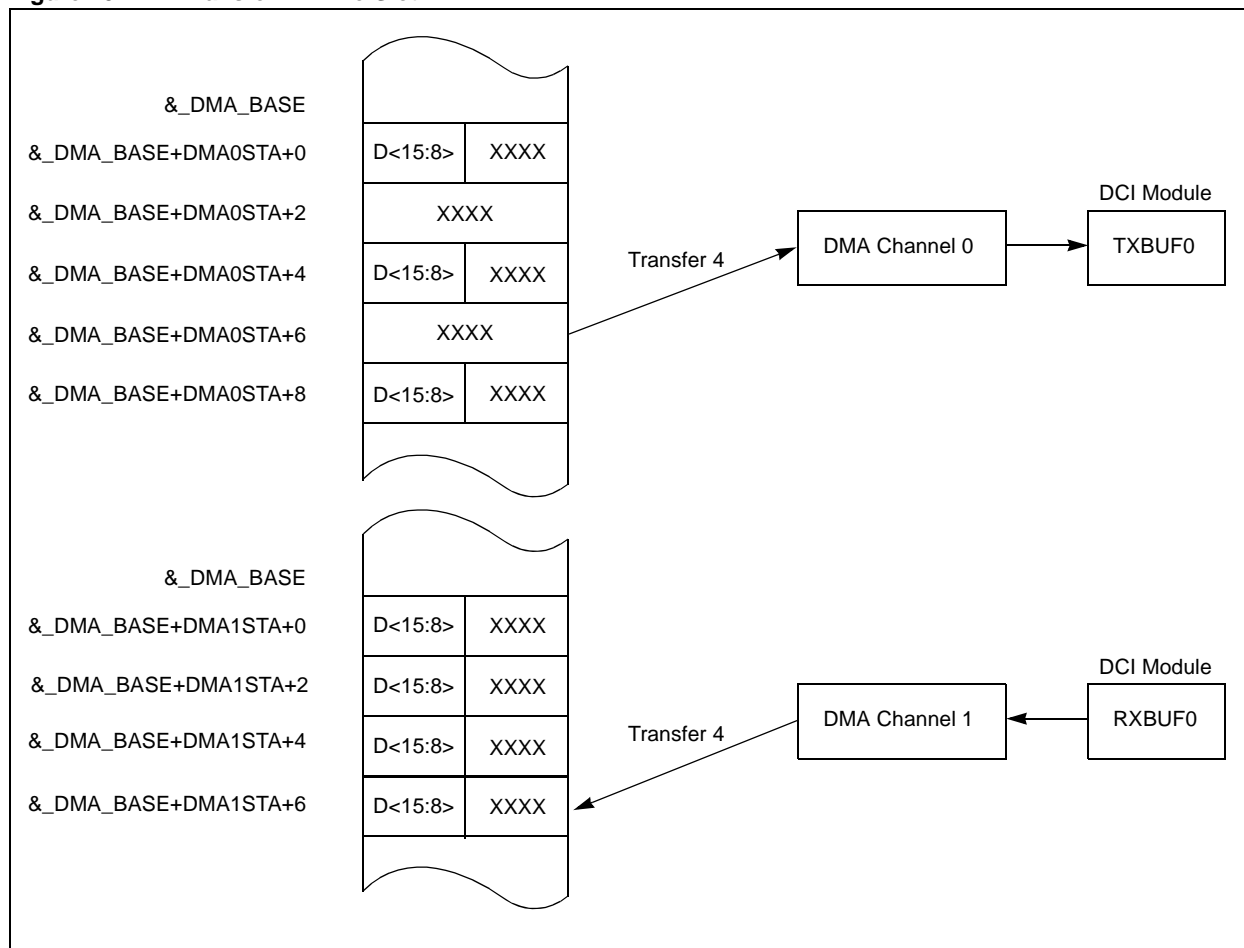
20.4.15 Receive Status Bits

The two receive status bits, Receive Buffer Full (RFUL) and Receive Overflow (ROV), indicate the status only for register locations that are enabled for use by the module. This is a function of the BLEN control bits (DCICON2<11:10>). If the buffer length is set to less than four words, the unused buffer locations do not affect the receive status bits.

The RFUL status bit (DCISTAT<2>) is read-only and indicates that new data is available in the receive registers. The RFUL bit is cleared automatically when all RXBUF registers in use have been read by the user software.

The ROV status bit (DCISTAT<3>) is read-only and indicates that a receive overflow has occurred for at least one of the receive register locations. A receive overflow occurs when the RXBUF register location is not read by the user software before new data is transferred from the buffer memory. When a receive overflow occurs, the old contents of the register are overwritten. The ROV status bit is cleared automatically when the register that caused the overflow is read.

20.4.16 Transmit Status Bits

The two transmit status bits, Transmit Buffer Empty (TMPTY) and Transmit Buffer Underflow (TUNF), indicate the status only for register locations that are used by the module. If the buffer length is set to less than four words, for example, the unused register locations do not affect the transmit status bits.

The TMPTY bit (DCISTAT<0>) is read-only and is set when the contents of the active TXBUF registers are transferred to the transmit buffer registers. The TMPTY bit can be polled in software to determine when the transmit registers can be written. The TMPTY bit is cleared automatically by the hardware when a write to any of the TXBUF registers in use occurs.

The TUNF bit (DCISTAT<1>) is read-only and indicates that a transmit underflow has occurred for at least one of the transmit registers in use. The TUNF bit is set when the TXBUF register contents are transferred to the transmit buffer memory and the user software did not write all of the TXBUF registers in use since the last buffer transfer. The TUNF status bit clears automatically when the TXBUF register that underflowed is written by the user software.

20.4.17 SLOT Status Bits

The SLOT status bits (DCISTAT<11:7>) indicate the current active time slot in the data frame. These bits are useful when more than four words per data frame need to be transferred. The user software can poll these status bits when a DCI interrupt occurs to determine what time slot data was last received and which time slot data should be loaded into the TXBUF registers.

20.4.18 Digital Loopback Mode

Digital Loopback mode is enabled by setting the Digital Loopback mode (DLOOP) control bit (DCICON1<11>). When the DLOOP bit is set, the module internally connects the CSDO signal to CSDI. The actual data input on the CSDI pin is ignored in Digital Loopback mode.

20.4.19 Underflow Mode Control Bit

When a transmit underflow occurs, one of two actions can occur depending on the state of the Underflow Mode (UNFM) control bit (DCICON1<7>).

- If the UNFM bit is cleared (default), the module transmits 0's on the CSDO pin during the active time slot for the buffer location. In this operating mode, the codec device attached to the DCI module is simply fed digital "silence".

- If the UNFM control bit is set, the module transmits the last data written to the buffer location. This operating mode permits the user software to send a continuous data value to the codec device without consuming software overhead.

20

**Data Converter
Interface (DCI)**

20.4.20 Data Justification Control

In most applications, the data transfer begins one serial clock cycle after the FS signal is sampled active (refer to Figure 20-18). This is the DCI module default.

Figure 20-18: Default Data Transfer
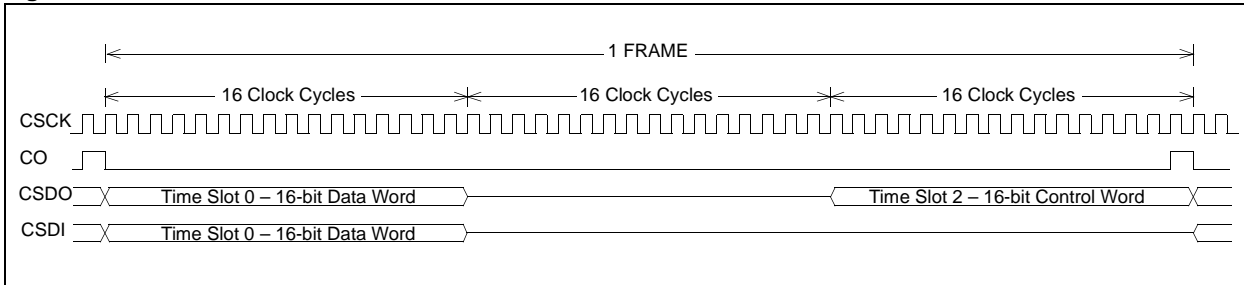
An alternate data alignment can be selected by setting the DJST control bit (DCICON1<5>). When DJST = 1, data transfers begin during the same serial clock cycle as the FS signal (refer to Figure 20-19).

Figure 20-19: Data Transfer Selection Using the DJST Control Bit
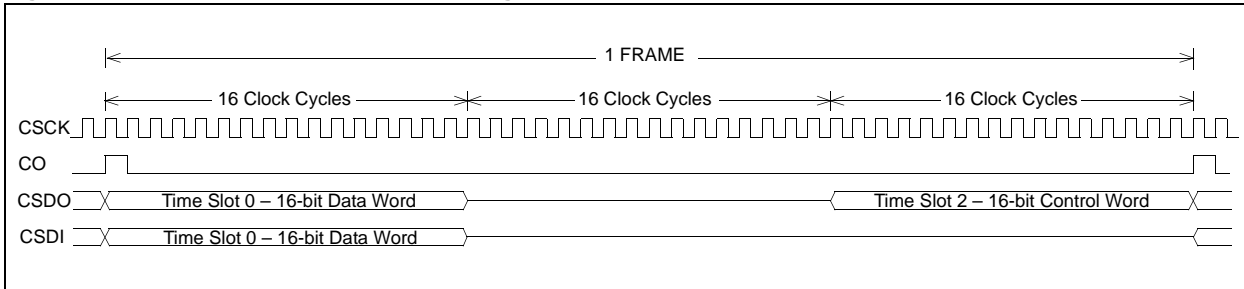
20.4.21 DCI Module Interrupts

The frequency of DCI module interrupts depends on the BLEN control bits. An interrupt is generated when the buffer length has been reached. If interrupts are enabled before the DCI module is enabled, an interrupt is generated three CSCK cycles after the module is enabled.

The DCI module also features an error interrupt. The error interrupt, if enabled, causes the CPU to interrupt when a transmit underflow or when a receive overflow event occurs.

20.5 USING THE DCI MODULE

This section explains how to configure and use the DCI with specific kinds of data converters.

20.5.1 How to Transmit and Receive Data Using the DCI Buffers, Status Bits, and Interrupts

The DCI can buffer up to four data words between CPU interrupts depending on the setting of the BLEN control bits. The buffered data can be transmitted and received in a single data frame, or across multiple data frames, depending on the TSCON and RSCON register settings. Following are four configuration examples:

1. Assume BLEN<1:0> = 00 (buffer one data word per interrupt) and TSCON = RSCON = 0x0001. This particular configuration represents the most basic setup and causes the DCI to transmit or receive one data word at the beginning of every data frame. The CPU is interrupted after every data word transmitted or received since BLEN<1:0> = 00. For details, refer to Figure 20-20.

Figure 20-20: DCI Timing with BLEN = 00 and TSCON = RSCON = 0x0001
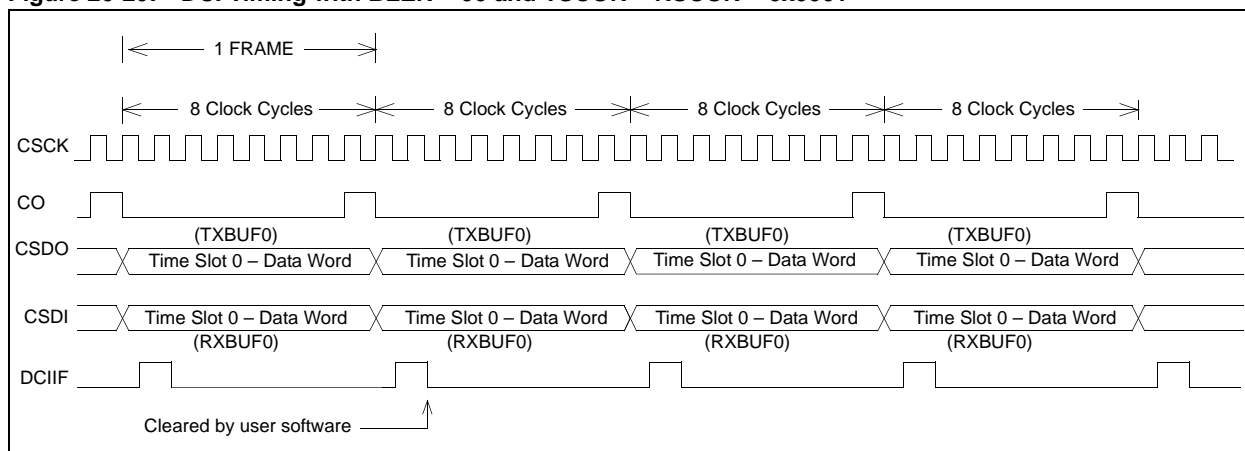
2. Assume BLEN<1:0> = 11 (buffer four data words per interrupt) and TSCON = RSCON = 0x0001. This configuration causes the DCI to transmit or receive one data word at the beginning of every data frame. A CPU interrupt is generated after four data words are transmitted or received. This configuration is useful for block processing, where multiple data samples are processed at once. For details, refer to Figure 20-21.

Figure 20-21: DCI Timing with BLEN = 11 and TSCON = RSCON = 0x0001
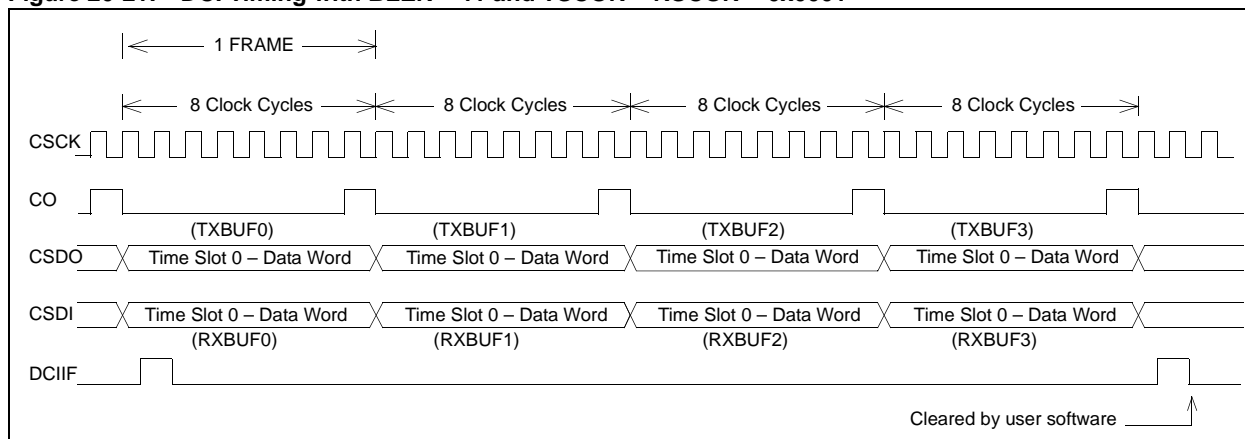
20

Data Converter Interface (DCI)

3. Assume BLEN<1:0> = 11 (buffer four data words per interrupt) and TSCON = RSCON = 0x000F. This configuration causes the DCI to transmit/receive four data words at the beginning of every data frame. A CPU interrupt is generated every data frame in this case because the DCI was set up to buffer four data words in a data frame. This configuration represents a typical multi-channel buffering setup. For details, refer to Figure 20-22.

Figure 20-22: DCI Timing with BLEN = 11 and TSCON = RSCON = 0x000F
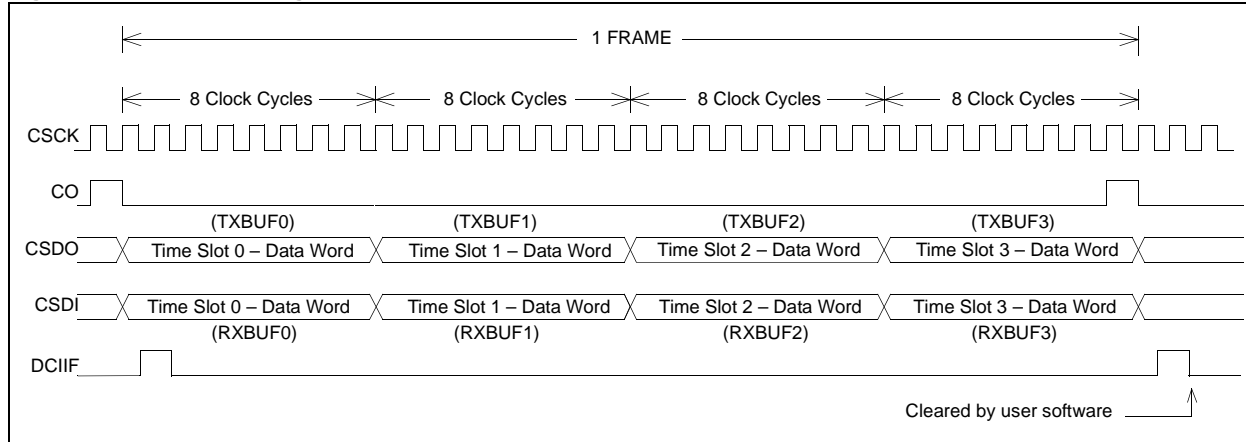
4. The DCI can also be configured to buffer more than four data words per frame. For example, assume BLEN<1:0> = 11 (buffer four data words per interrupt) and TSCON = RSCON = 0x00FF. In this configuration, the DCI transmits/receives 8 data words per data frame. An interrupt is generated twice per data frame. To determine which portion of the data is in the transmit or receive registers at each interrupt, the user software must check the SLOT status bits (DCISTAT <11:7>) in the Interrupt Service Routine (ISR) to determine the current data frame position. Figure 20-23 shows a 4-bit example for this case.

Figure 20-23: DCI Timing with BLEN = 11 and TSCON = RSCON = 0x00FF
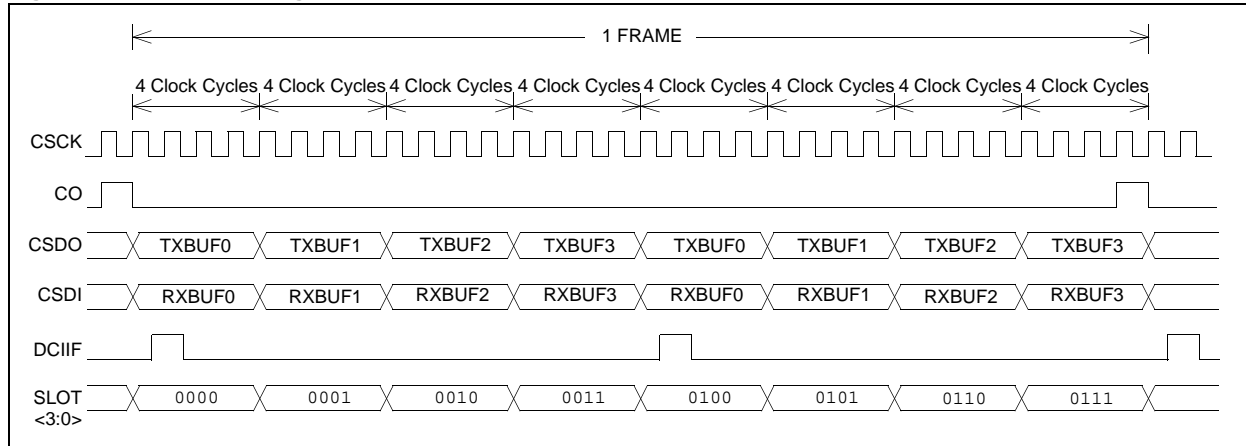
The transmit and receive registers are double-buffered, so the DCI module can work on one set of transmit and receive data, while the user software is manipulating the other set of data. Because of the double buffers, it takes three interrupt periods to receive the data, process that data, and transmit the processed data. For each DCI interrupt, the CPU processes a data word received during a prior interrupt period and generates a data word transmitted during the next interrupt period. The buffering and data processing time of the dsPIC DSC device inserts a two-interrupt period delay into the processed data. In most cases, this data delay is negligible.

The DCI status flags and CPU interrupt indicate that a buffer transfer has taken place and that it is time for the CPU to process more data. In a typical application, the following occurs each time the DCI data is processed:

1. RXBUF registers are read by the user software.
2. RFUL status bit (DCISTAT<2>) is set by the module to indicate that the receive registers contain new data.
3. RFUL bit is cleared automatically after all the active receive registers have been read.
4. User software processes the received data.
5. TMPTY status bit (DCISTAT<0>) is set to indicate that the transmit registers are ready for more data to be written.
6. Processed data is written to the TXBUF registers.

For applications that are configured to transmit and receive data (TSCON and RSCON are non-zero), the RFUL (DCISTAT<2>) and TMPTY (DCISTAT<0>) status bits can be polled in user software to determine when a DCI buffer transfer takes place.

- If the DCI is used only to transmit data (RSCON = 0), the TMPTY bit can be polled to indicate a buffer transfer
- If the DCI is configured to only receive data (TSCON = 0), the RFUL bit can be polled to indicate a buffer transfer

The DCIIF status bit (IFS2<9>) is set each time a DCI buffer transfer takes place and generates a CPU interrupt, if enabled. The DCIIF status bit is generated by the logical ORing of the RFUL (DCISTAT<2>) and TMPTY (DCISTAT<0>) status bits.

20.5.1.1 DCI START-UP AND DATA BUFFERING

For DCI start-up, first initialize the DCI control registers for the desired operating mode. Data transfers are begun by setting the DCIEN control bit (DCICON1<15>). Refer to **20.5.4 "Multi-Channel Operation"**, **20.5.5 "I2S Operation"**, and **20.5.6 "AC-Link Operation"**.

Figure 20-24 shows a timing diagram for DCI start-up. In this example, the DCI is configured for an 8-bit data word (WS<3:0> = 0111) and an 8-bit data frame (COFSG<3:0> = 0000). The buffer length is set to 1 buffer (BLEN = 00), the transmit time slot 0 is enabled (TSCON = 0x1), and the receive time slot 0 is enabled (RSCON = 0x1). In addition, the Multi-Channel mode (COFSM<1:0> = 00) is used. The steps required to transmit and receive data are described as:

1. Preload the TXBUF registers with the first data to be transmitted before the module is enabled. If the transmit data is based on data received from the codec, the user software can simply clear the TXBUF registers. This transmits digital "silence" until data is first received into the RXBUF registers from the codec.
2. Enable the DCI module by setting the DCIEN bit (DCICON1<15>). If the DCI is the master device, the data in the TXBUF registers is transferred to the transmit buffers and transmission of the first data frame commences. Otherwise, the TXBUF data is held in the transmit buffers until a FS signal is received from the master device.
3. The TMPTY bit (DCISTAT<0>) is set 3 clock cycles after the module is enabled and a DCI interrupt is generated, if enabled. At this time, the module is ready for the TXBUF registers to be reloaded with data to be transferred on the second data frame. No data has been received by the module, so the TXBUF registers are cleared again if the transmitted data is calculated from the received data. If interrupts are enabled, clear the DCIIF status bit in user software.
4. After the first data frame is transferred, the TMPTY bit (DCISTAT<0>) is set, the RFUL status bit is set, and a DCI interrupt occurs, if enabled. This is the first data word received from the device connected to the DCI.
5. The user software reads the receive register, automatically clearing the RFUL status bit. The user software also processes the received data.
6. The transmit register is written with data to be transmitted during the next data frame. The TMPTY status bit (DCISTAT<0>) is cleared automatically when the write occurs. The write data can be calculated from data received at the prior interrupt.
7. The next DCI interrupt occurs and the cycle repeats.

20

Data Converter Interface (DCI)

Figure 20-24:   DCI Start-up and Data Buffering Example
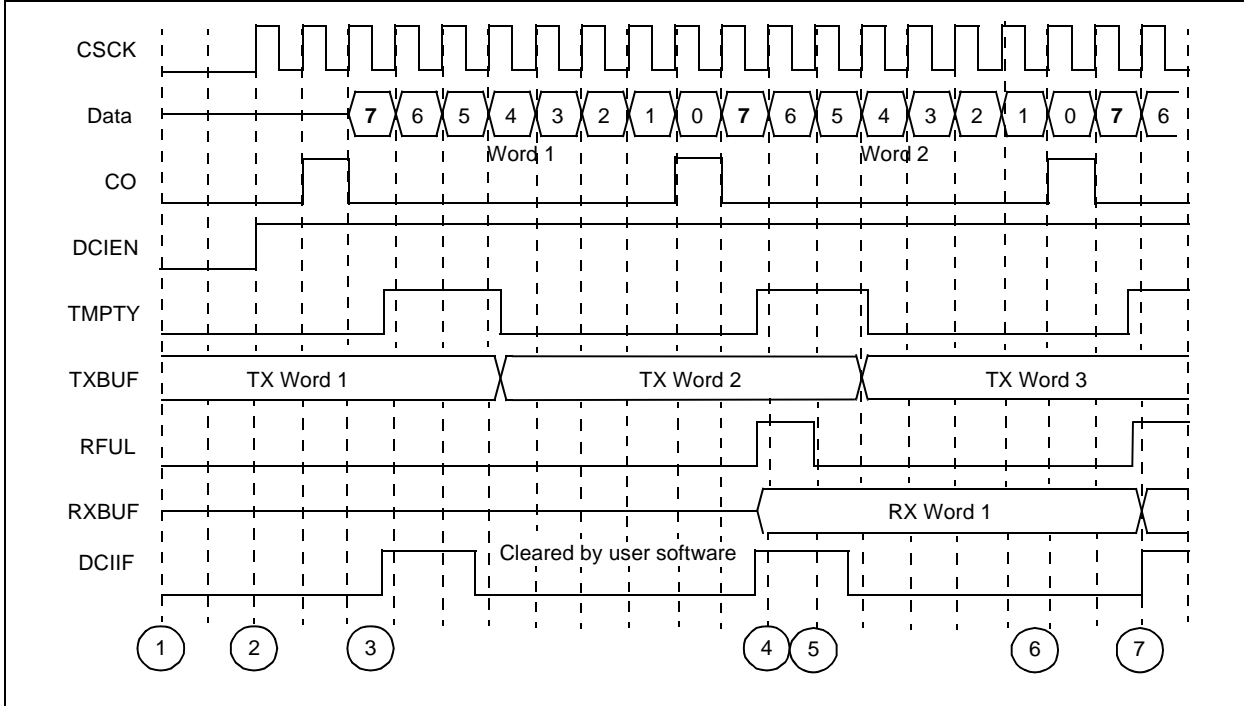
20.5.1.2   DCI DISABLE

The DCI module is disabled by clearing the DCIEN control bit (DCICON1<15>). When the DCIEN bit is cleared, the module finishes the data frame transfer in progress. An interrupt is generated if the transmit/receive buffers need to be written or read before the end of the frame.

The DCIEN bit must be cleared at least 3 CSCK cycles before the end of the frame for the module to be disabled in that frame. If the bit is not cleared in time, the module is disabled on the next frame.

Once disabled, the DCI will not generate any further FS pulses, nor will it respond to an incoming FS pulse.

When the FSG has reached the final time slot in the data frame, all state machines associated with the DCI are reset to their Idle state and control of the I/O pins associated with the module is released. The user software can poll the SLOT status bits (DCISTAT<11:8>) after the DCIEN bit (DCICON1<15>) is cleared to determine when the module is idle. The DCI is idle when SLOT<3:0> = `0000` and DCIEN = `0`.

When the module enters an Idle state, any data in the receive shadow registers is transferred to the RXBUF registers, and the RFUL and ROV status bits are affected accordingly.

Figure 20-25: DCI Timing, Module Disable

Receive buffer contents transferred to RXBUF

20.5.2 Master vs. Slave Operation

The DCI can be configured for master or slave operation. The master device generates the FS signal to initiate a data transfer. The operating mode (Master or Slave) is selected by the COFSD control bit (DCICON1<8>).

When the DCI module is operating as a master device (COFSD = 0), the COFSM mode bits (DCICON1<1:0>) determine the type of FS pulse that is generated by the FSG logic. A new FS signal is generated when the FSG resets and is output on the COFS pin.

When the DCI module is operating as a FS slave (COFSD = 1), data transfers are controlled by the device attached to the DCI module. The COFSM control bits (DCICON1<1:0>) control how the DCI module responds to incoming FS signals.

In the Multi-Channel mode, a new data frame transfer begins one serial clock cycle after the COFS pin is sampled high. The pulse on the COFS pin resets the FSG logic.

In the I^{2}S mode, a new data word is transferred one serial clock cycle after a low-to-high or a high-to-low transition is sampled on the COFS pin. A rising or falling edge on the COFS pin resets the FSG logic.

In the AC-Link mode, the tag slot and subsequent data slots for the next frame is transferred one serial clock cycle after the COFS pin is sampled high.

The COFSM (DCICON1<1:0>) and WS (DCICON2<3:0>) bits must be configured to provide the expected frame length when the module is operating in Slave mode. Once a valid FS pulse is sampled by the module on the COFS pin, an entire data frame transfer takes place. The module will not respond to further FS pulses until the current data frame transfer has fully completed.

20.5.3 Data Packing for Long Data Word Support

Many codecs have data word lengths in excess of 16 bits. The DCI natively supports word lengths up to 16 bits, but longer word lengths can be supported by enabling multiple transmit and receive slots and packing data into multiple transmit and receive buffer locations.

For example, assume that a particular codec transmits or receives 24-bit data words. This data could be transmitted and received by setting BLEN<1:0> = 01 (two data words per interrupt) and setting TSCON = 0x0003 and RSCON = 0x0003. This enables transmission and reception during the first two time slots of the data frame. The 16 MSbs of the transmit data are written to TXBUF0. The 8 LSbs of the transmit data are written left justified to TXBUF1, as shown in Figure 20-26. The value of the 8 LSbs of TXBUF1 can be written to '0'. The 24-bit data received from the codec is loaded into RXBUF0 and RXBUF1 with the same format as the transmit data. In this case, the FS signal is generated at the 32-bit intervals.

20

Data Converter Interface (DCI)

Any combination of word size and enabled time slots can be used to transmit and receive long data words in multiple transmit and receive registers. For example, the 24-bit data word example shown in Figure 20-26 could be transmitted or received in three consecutive registers by setting WS<3:0> = 0111 (word size = 8 bits), BLEN<1:0> = 10 (buffer three words between interrupts), and TSCON = RSCON = 0x0007 (transmit or receive during the first three time slots of the data frame). Each transmit and receive register would contain 8 bits of the data word (refer to Figure 20-27). If COFSG = 0010 (3 words per frame), the FS signal would be generated at 24-bit intervals.

Figure 20-26: Data Packing Example for Long Data Words

Figure 20-27: Data Packing Example for Long Data Words with WS = 0111, and TSCON = RSCON = 0x0007

20.5.4 Multi-Channel Operation

Multi-Channel mode (COFSM<1:0> = 00) is used for codecs that require a FS pulse that is driven high for one serial clock period to initiate a data transfer. One or more data words can be transferred in the data frame. The number of clock cycles between successive FS pulses depends on the device connected to the DCI module. Figure 20-28 is a timing diagram for the FS signal in Multi-Channel mode. Figure 20-2 is a timing example indicating a 4-bit word data transfer.

Figure 20-28: Frame Synchronization Timing, Multi-Channel Mode

20.5.4.1 MULTI-CHANNEL SETUP DETAILS

This section provides the steps required to configure the DCI for a codec using Multi-Channel mode. This operating mode can be used for codecs with one or more data channels. The setup is similar regardless of the number of channels.

For this setup example, a hypothetical codec is assumed. The single channel codec used for this setup example uses a 256 fs serial clock frequency with a 16-bit data word transmitted at the beginning of each frame.

The steps required for setup and operation are described below.

1. Determine the sample rate and data word size required by the codec. An 8 kHz sampling rate is assumed for this example.

2. Determine the serial transfer clock frequency required by the codec. Most codecs require a serial clock signal that is some multiple of the sampling frequency. The example codec requires a frequency that is 256 fs, or 1.024 MHz. Therefore, a FS pulse must be generated every 256 serial clock cycles to start a data transfer.

3. Configure the DCI for the serial transfer clock.
   - If the CSCK signal is generated by the DCI, clear the CSCKD control bit (DCICON1<10>) and write a value to DCICON3 that produces the correct clock frequency (refer to **20.4.3 "Bit Clock Generator"**).
   - If the CSCK signal is generated by the codec or other external source, set the CSCKD control bit and clear the DCICON3 register.

4. Clear the COFSM control bits (DCICON1<1:0>) to set the FS signal to Multi-Channel mode.

5. If the DCI is generating the FS signal (master), clear the COFSD control bit (DCICON1<8>). If the DCI is receiving the FS signal (slave), set the COFSD control bit.

6. Clear the CSCKE control bit (DCICON1<9>) to sample incoming data on the falling edge of CSCK. This is the typical configuration for most codecs. Refer to the codec data sheet to ensure the correct sampling edge is used.

7. Write the WS control bits (DCICON2<3:0>) for the desired data word size. The example codec requires WS<3:0> = `1111` for a 16-bit data word size.

8. Write the COFSG control bits (DCICON2<8:5>) for the desired number of data words per frame. The WS and COFSG control bits determine the length of the data frame in CSCK cycles (refer to **20.4.7 "Frame Synchronization Generator"**). COFSG<3:0> = `1111` is used to provide the 256-bit data frame required by the example codec.

9. Set the output mode for the CSDO pin using the CSDOM control bit (DCICON1<6>). If a single device is attached to the DCI, CSDOM can be cleared. This forces the CSDO pin to '0' during unused data time slots. You may need to set CSDOM if multiple devices are attached to the CSDO pin.

10. Write the TSCON and RSCON registers to determine which data time slots in the frame are to be transmitted and received, respectively. For this single-channel codec, use TSCON = RSCON = 0x0001 to enable transmission and reception during the first 16-bit time slot of the data frame.

11. Set the BLEN control bits (DCICON2<11:10>) to buffer the desired amount of data words. For the single-channel codec, BLEN = `00` provides an interrupt at each data frame. A higher value of BLEN could be used for this codec to buffer multiple samples between interrupts.

12. If interrupts are to be used, clear the DCIIF status bit (IFS2<9>) and set the DCIIE control bit (IEC2<9>).

13. Begin operation as described in **20.5.1.1 "DCI Start-up and Data Buffering"**.

20.5.4.2 MULTI-CHANNEL SETUP WITH DMA

The Multi-Channel DCI Setup with DMA is similar to the setup described in **20.5.4.1 "Multi-Channel Setup Details"** with the following exceptions:

- BLEN must be '0'. Setting BLEN to any other value results in unpredictable behavior.
- The DMA channels must be configured to read/write to the RXBUF0/TXBUF0 registers. For more details on configuring the DMA module, refer to **"Section 22. Direct Memory Access (DMA)"** (DS70182) of the "*dsPIC33F Family Reference Manual"*.

To use the DMA:

- One DMA channel must be configured to read from Dual Port SRAM (DPSRAM) and write to the TXBUF0 register
- A second DMA channel is configured to read from the RXBUF0 register and write to the DPSRAM
- The DMA channels must be enabled before enabling the DCI module. This ensures that the first DCI interrupt is processed by the DMA module

Figure 20-29 shows the DPSRAM organization for this example. The 16-bit data to be transmitted must be stored at consecutive locations in DPSRAM since only one time slot is enabled in the DCI module. The received data will be stored at consecutive locations in DPSRAM.
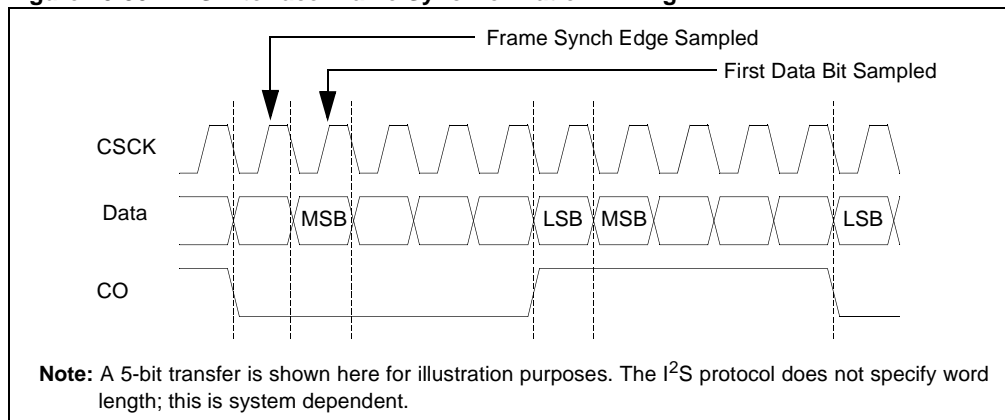
Figure 20-29: DPSRAM Organization for Multi-Channel Mode
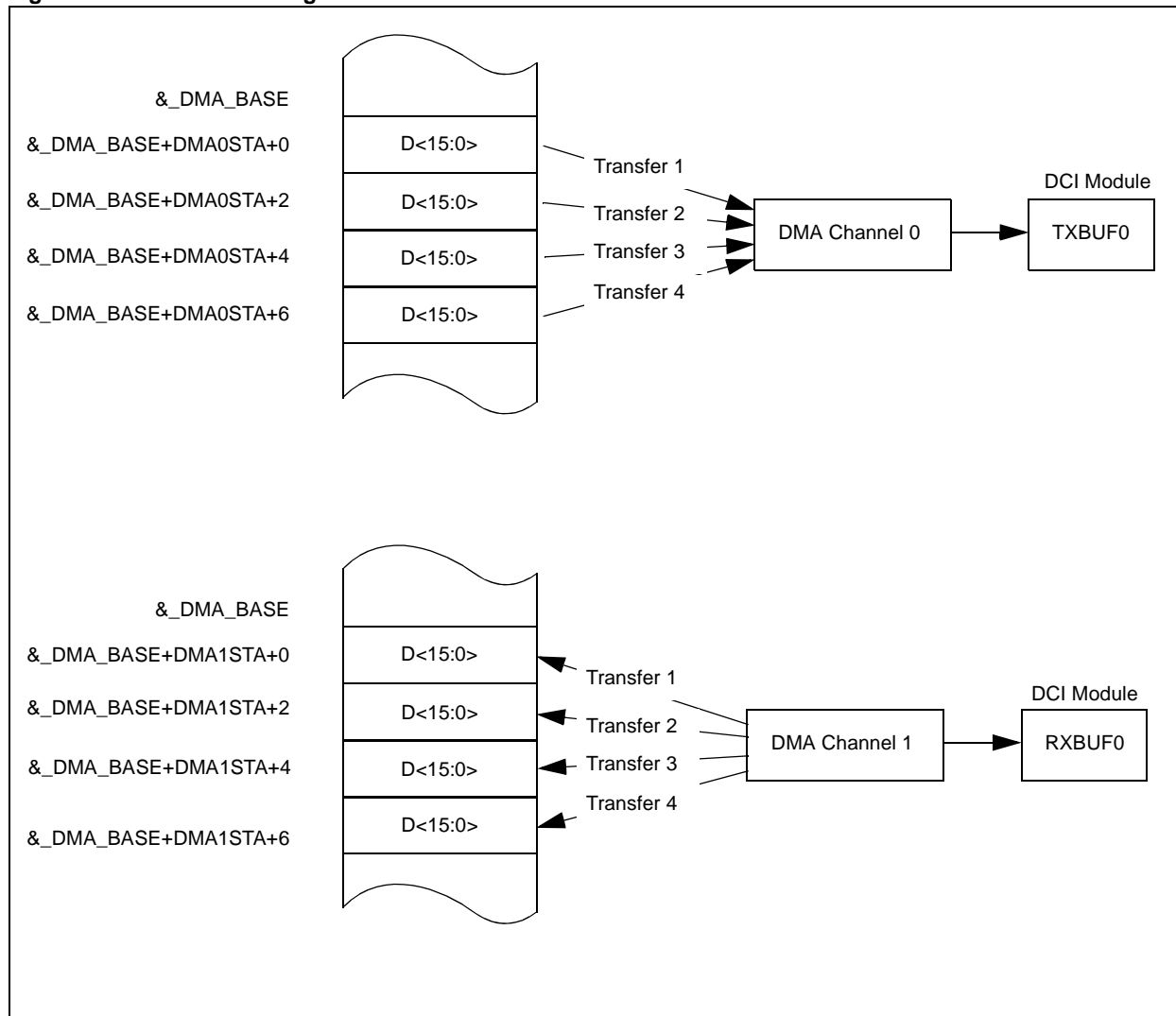
20.5.5   I²S Operation

The I²S operating mode is used for codecs that require a FS signal that has a 50% duty cycle. The period of the I²S FS signal in serial clock cycles is determined by the word size of the codec connected to the DCI module. Figure 20-30 shows the start of a new word boundary is marked by a high-to-low or a low-to-high transition edge on the COFS pin. I²S codecs are generally stereo or two-channel devices, with one data word transferred during the low time of the FS signal and the other data word transmitted during the high time.

Figure 20-30:   I²S Interface Frame Synchronization Timing

Note: A 5-bit transfer is shown here for illustration purposes. The I²S protocol does not specify word length; this is system dependent.

The DCI module is configured for I²S mode by writing a value of 01h to the COFSM<1:0> control bits in the DCICON1 SFR. When operating in the I²S mode, the DCI module generates FS signals with a 50% duty cycle. Each edge of the FS signal marks the boundary of a new data word transfer. The user software must also select the frame length and data word size using the COFSG and WS control bits in DCICON2.

20.5.5.1   I²S SETUP DETAILS

This section provides the steps required to configure the DCI for an I²S codec. For this example, a hypothetical I²S codec is assumed.

The I²S codec in this setup example uses a 64 fs serial clock frequency, with two 16-bit data words during the data frame. Therefore, the frame length is 64 CSCK cycles, with the COFS signal high for 32 cycles and low for 32 cycles. The first data word is transmitted one CSCK cycle after the falling edge of COFS, and the second data word is transmitted one CSCK cycle after the rising edge of COFS.

1. Determine the sample rate used by the codec to determine the CSCK frequency. It is assumed in this example that fs is 48 kHz.
2. Determine the serial transfer clock frequency required by the codec. The example codec requires a frequency that is 64 fs, or 3.072 MHz.
3. The DCI must be configured for the serial transfer clock. If the CSCK signal is generated by the DCI, clear the CSCKD control bit (DCICON1<10>) and write a value to DCICON3 that produces the correct clock frequency (refer to **20.4.3 "Bit Clock Generator"**). If the CSCK signal is generated by the codec or other external source, set the CSCKD control bit and clear the DCICON3 register.
4. Set COFSM<1:0> = 01 to set the FS signal to I²S mode.
5. If the DCI is generating the FS signal (master), clear the COFSD control bit (DCICON1<8>). If the DCI is receiving the FS signal (slave), set the COFSD control bit.
6. Set the CSCKE control bit (DCICON1<9>) to sample incoming data on the rising edge of CSCK. This is the typical configuration for most I²S codecs.
7. Write the WS control bits (DCICON2<3:0>) for the desired data word size. For the example codec, use WS<3:0> = 1111 for a 16-bit data word size.

20

Data Converter Interface (DCI)

8. Write the COFSG control bits (DCICON2<8:5>) for the desired number of data words per frame. The WS and COFSG control bits determine the length of the data frame in CSCK cycles (refer to **20.4.7 "Frame Synchronization Generator"**). For this example codec, set COFSG<3:0> = `0001`.

> **Note:** In the I^{2}S mode, the COFSG bits are set to the length of half of the data frame. For this example codec, set COFSG<3:0> = `0001` (two data words per frame) to produce a 32-bit frame. This produces an I^{2}S data frame that is 64 bits in length.

9. Set the Output mode for the CSDO pin using the CSDOM control bit (DCICON1<6>). If a single device is attached to the DCI, CSDOM can be cleared. You may need to set CSDOM if multiple devices are attached to the CSDO pin.

10. Write the TSCON and RSCON registers to determine which data time slots in the frame are to be transmitted and received, respectively. For this codec, set TSCON = 0x0001 and RSCON = 0x0001 to enable transmission and reception during the first 16-bit time slot of the 32-bit data frame. Adjacent time slots can be enabled to buffer data words longer than 16 bits.

11. Set the BLEN control bits (DCICON2<11:10>) to buffer the desired amount of data words. For a two-channel I^{2}S codec, BLEN<1:0> = `01` generates an interrupt after transferring two data words.

12. If interrupts are to be used, clear the DCIIF status bit (IFS2<9>) and set the DCIIE control bit (IEC2<9>).

13. Begin operation as described in **20.5.1.1 "DCI Start-up and Data Buffering"**. In the I^{2}S Master mode, the COFS pin is driven high after the module is enabled and begins transmitting the data loaded in TXBUF0.

20.5.5.2 I^{2}S SETUP WITH DMA.

The I^{2}S DCI setup with DMA is similar to the setup described in **20.5.5.1 "I2S Setup Details"** with following exceptions:

• BLEN must be '`0`'. Setting BLEN to any other value results in unpredictable behavior.
• The DMA channels must be configured to read or write to the RXBUF0/TXBUF0 registers. For more details on configuring the DMA module, refer to **"Section 22. Direct Memory Access (DMA)"** (DS70182) of the "*dsPIC33F Family Reference Manual*".

To use the DMA:

• One DMA channel must be configured to read from Dual Port SRAM (DPSRAM) and write to the TXBUF0 register
• A second DMA channel is configured to read from RXBUF0 register and write to the DPSRAM
• The DMA channels must be enabled before enabling the DCI module. This ensures that the first DCI Interrupt is processed by the DMA module.

Figure 20-31 shows the DPSRAM organization for this setup. The transmit data is organized as the first data word (which is transmitted at the falling edge of the COFS signal) followed by the second data word (which is transmitted at the rising edge of the COFS signal).

Figure 20-31: DPSRAM Organization for I²S Mode
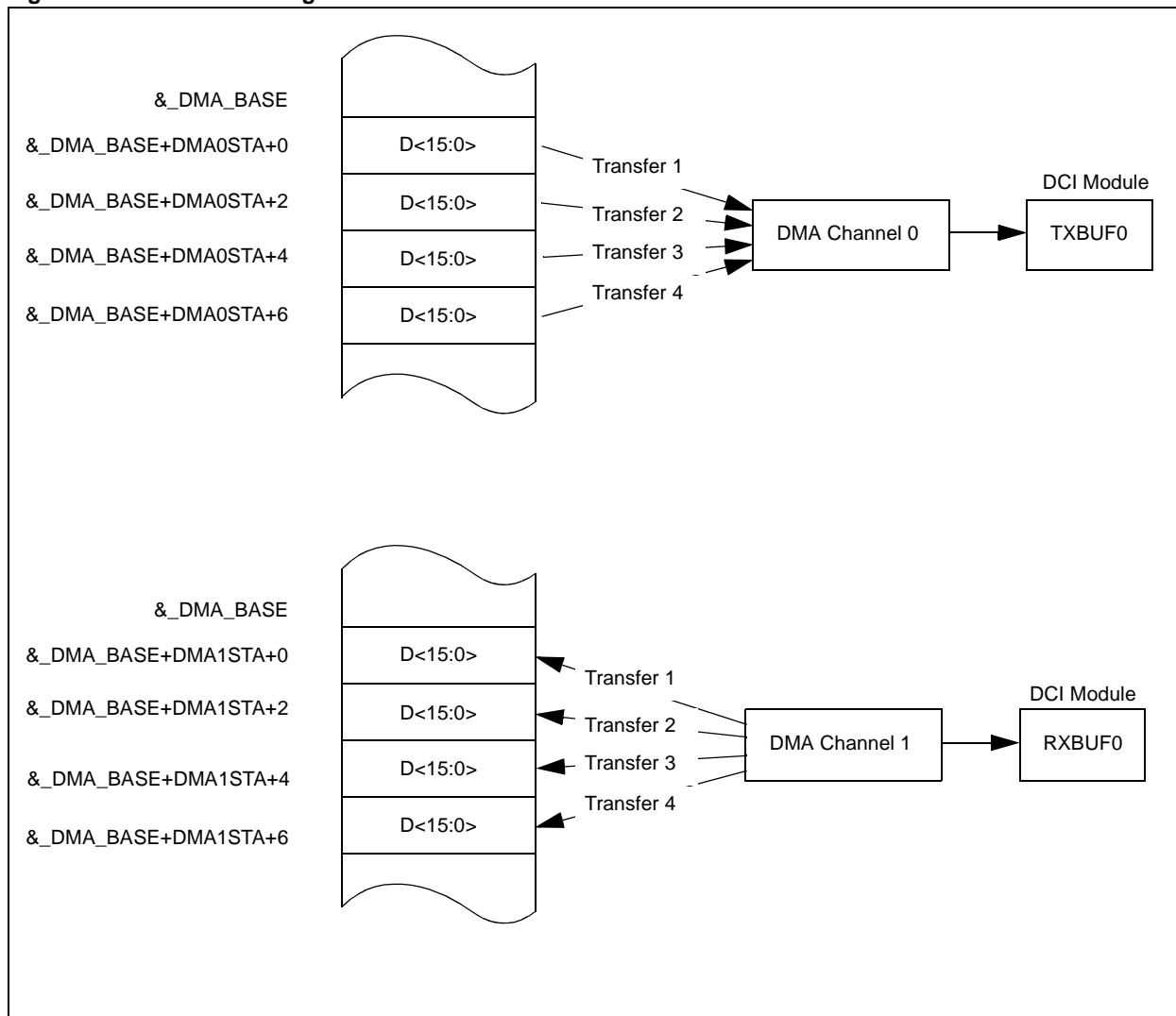
## 20.5.5.3 HOW TO DETERMINE THE I²S CHANNEL ALIGNMENT

Most I²S codecs support two channels of data, and the level of the FS signal indicates the channel transferred during that half of the data frame. The COFS pin can be polled in software using its associated PORT register to determine the present level on the pin in the DCI Interrupt Service Routine. This indicates which data is in the receive register and which data should be written to the transmit registers for transfer on the next frame.

20.5.5.4 I²S DATA JUSTIFICATION

As per the I²S specification, a data word transfer by default begins one serial clock cycle following a transition of the FS signal. An "MSb left-justified" option can be selected using the DJST control bit (DCICON1<5>).

If DJST = 1, the I²S data transfers are MSb left-justified. The MSb of the data word is presented on the CSDO pin during the same serial clock cycle as the rising or falling edge of the FS signal. After the data word has been transmitted, the state of the CSDO pin is dictated by the CSDOM (DCICON1<6>) bit.

20

Data Converter Interface (DCI)

Figure 20-32:   I²S Data Justification Options
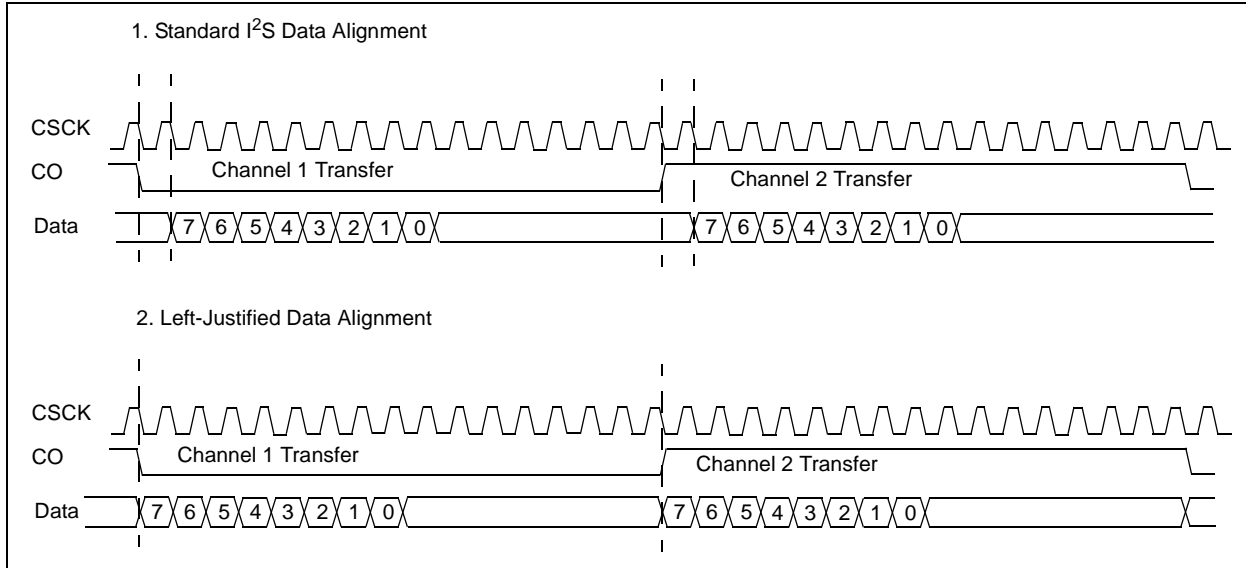
20.5.6   AC-Link Operation

This section describes how to use the DCI in the AC-Link modes. The AC-Link modes communicate with AC-'97 compliant codec devices.
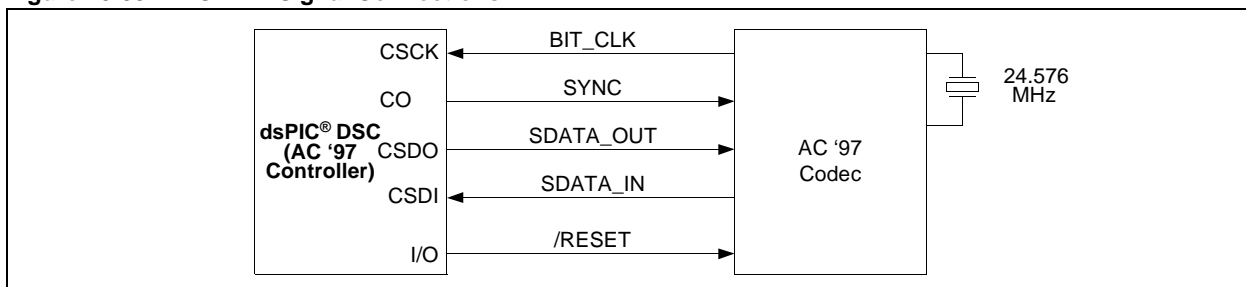
20.5.6.1   AC-LINK DATA FRAME

The AC-Link data frame is 256 bits subdivided into one 16-bit control slot, followed by twelve 20-bit data slots. Figure 20-33 shows the AC-'97 codec usually provides the serial transfer clock signal, which is derived from a crystal oscillator.

The controller receives the serial clock and generates the FS signal. The default data frame rate is 48 kHz. The FS signal used for AC-Link systems is high for 16 CSCK periods at the beginning of the data frame and low for 240 CSCK periods.

Figure 20-35 shows the data transfer begins one CSCK period after the rising edge of the FS signal. Data is sampled by the receiving device on the falling edge of CSCK. Figure 20-34 shows the control and data time slots in the AC-Link have defined uses in the protocol.

Figure 20-33:   AC-Link Signal Connections
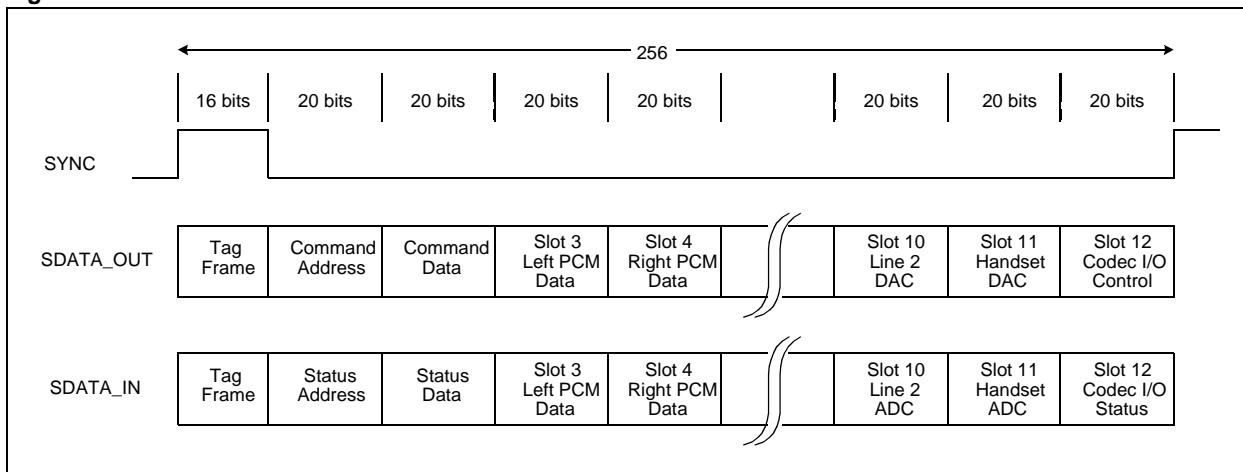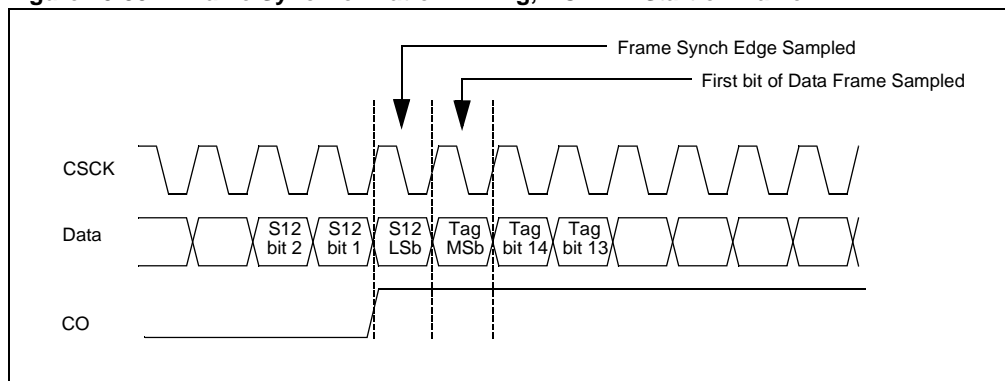
Figure 20-34: AC-Link Data Frame

Figure 20-35: Frame Synchronization Timing, AC-Link Start of Frame

The DCI module has two operating modes for the AC-Link protocol to accommodate the 20-bit data time slots. These operating modes are selected using the COFSM control bits (DCICON1<1:0>).

- To select the first AC-Link mode, called "16-bit AC-Link mode," set COFSM<1:0> = 10
- To select the second AC-Link mode, called "20-bit AC-Link mode," set COFSM<1:0> = 11

20.5.6.2 16-BIT AC-LINK MODE

In the 16-bit AC-Link mode, transmit and receive data word lengths are restricted to 16 bits to fit the DCI transmit and receive registers. This restriction only affects the 20-bit data time slots of the AC-Link protocol. For received time slots, the incoming data is truncated to 16 bits. For outgoing time slots, the module sets the 4 LSbs of the data word to '0'. This operating mode simplifies the AC-Link data frame by treating every time slot as a 16-bit time slot. The FSG maintains alignment to the time slot boundaries.

20.5.6.3 20-BIT AC-LINK MODE

The 20-bit AC-Link mode allows all bits in the data time slots to be transmitted and received, but does not maintain data alignment to the specific time slot boundaries defined in the AC-Link protocol.

The 20-bit AC-Link mode functions similarly to the Multi-Channel mode of the DCI module, except for the duty cycle of the FS signal that is produced. The AC-Link FS signal should remain high for 16 clock cycles and should be low for the following 240 cycles.

The 20-bit mode treats each 256-bit AC-Link frame as sixteen 16-bit time slots. In the 20-bit AC-Link mode, the module operates as if COFSG<3:0> = 1111 and WS<3:0> = 1111. The data alignment for 20-bit data slots is not maintained in this operating mode.

20

Data Converter Interface (DCI)

For example, an entire 256-bit AC-Link data frame can be transmitted and received in a packed fashion by setting all bits in the TSCON and RSCON registers. Since the total available buffer length is 64 bits, it takes four consecutive interrupts to transfer the AC-Link frame. The application software must keep track of the current AC-Link frame segment by monitoring the SLOT status bits (DCISTAT<11:8>).

20.5.6.4   AC-LINK SETUP DETAILS

To enable AC-Link mode, write `10h` or `11h` to the COFSM<1:0> control bits in the DCICON1 SFR. The word size selection bits (WS<3:0>) and the FSG bits (COFSG<3:0>) have no effect for the 16- and 20-bit AC-Link modes since the frame and word sizes are set by the protocol.

Most AC '97 codecs generate the clock signal that controls data transfers. Therefore, the CSCKD control bit is set in software. The COFSD control bit is cleared because the DCI generates the FS signal from the incoming clock signal. The CSCKE bit is cleared so that data is sampled on the rising edge.

The user must decide which time slots in the AC-Link data frame are to be buffered and set the TSE and RSE control bits in software accordingly. At a minimum, it is necessary to buffer the transmit and receive tag slots. Therefore, set the TSCON<0> and RSCON<1> control bits in software.

> **Note:** Only the TSCON<12:0> control bits and the RSCON<12:0> control bits have an effect in the 16-bit AC-Link mode, since an AC-Link frame has 13 time slots.

To set up the module for AC-Link mode:

1. Configure the DCI to accept the serial transfer clock from the AC'97 codec. Set the CSCKD control bit and clear the DCICON3 register.
2. Set the COFSM<1:0> control bits (DCICON1<1:0>) to '10b' or '11b' to set the desired AC-Link Frame Synchronization mode.
3. Clear the COFSD control bit (DCICON1<8>), so the DCI outputs the FS signal.
4. Clear the CSCKE control bit (DCICON1<9>) to sample incoming data on the falling edge of CSCK.

> **Note:** The word size selection bits (WS<3:0>) and the FSG bits (COFSG<3:0>) have no effect for the 16- and 20-bit AC-Link modes, since the frame and word sizes are set by the protocol.

5. Clear the CSDOM control bit (DCICON1<6>).
6. Write the TSCON and RSCON registers to determine which data time slots in the frame are to be transmitted and received, respectively. This depends on which data time slots in the AC-Link protocol is used. At a minimum, communication on slot 0 (tag slot) is required. For additional information, refer to the discussion in **20.5.6.2 "16-bit AC-Link Mode"**, **20.5.6.3 "20-bit AC-Link Mode"** and the Appendix of this manual.
7. Set the BLEN control bits (DCICON2<11:10>) to buffer the desired amount of data words. For the single channel codec, BLEN = `00` provides an interrupt at each data frame. A higher value of BLEN could be used for this codec to buffer multiple samples between interrupts.
8. If interrupts are to be used, clear the DCIIF status bit (IFS2<9>) and set the DCIIE control bit (IEC2<9>).
9. Begin operation as described in **20.5.1.1 "DCI Start-up and Data Buffering"**.

20.6   DCI CONFIGURATION CODE EXAMPLE

This section describes the configuration of the DCI module for operation in Slave mode with a 16-bit codec. Figure 20-36 shows the timing diagram for the codec. The control registers of the codec can be accessed by inputting a control word 16 clock cycles after the data word is received. The codec also outputs a status word 16 clock cycles after the data word is transmitted. The codec is configured to be a master and will drive the COFS and CSCK pins of the DCI module.

Figure 20-36:   Codec Timing Diagram for DCI Configuration Code Example
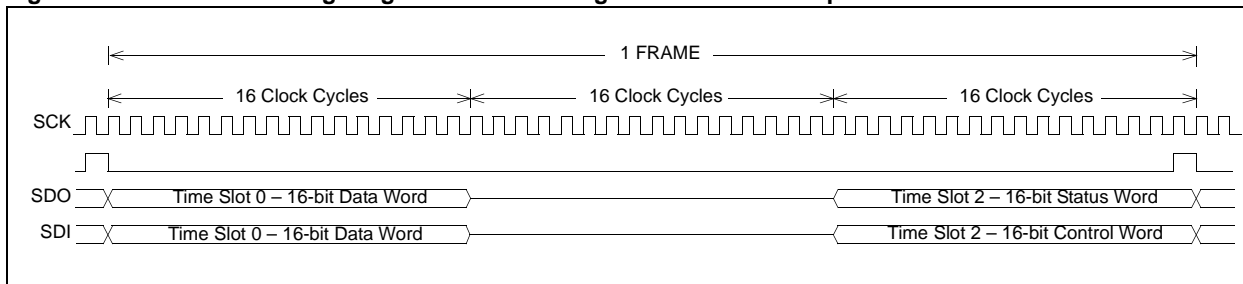
Since the codec transmits and receives the same number of time slots in a frame (TSCON = RSCON), the user application can write to successive TXBUF registers to transmit data and read from successive RXBUF registers.

Example 20-1 shows the code example for configuring the DCI module to interface with this codec. The code example sets up the DCI module to interrupt at two-word intervals (BLEN = 1). The application must write to two buffers or read from two buffers per interrupt in this scheme.

Alternatively, the user application could set up the module to interrupt at one-word intervals (BLEN = 0). This results in two interrupts per frame and requires the user application to process only one buffer per interrupt.

20

**Data Converter
Interface (DCI)**

Example 20-1:   DCI Configuration Code Example

```
#include "p33Fxxxx.h"

/* Device configuration registers */
_FGS(GWRP_OFF & GCP_OFF);
_FOSCSEL(FNOSC_PRIPLL);
_FOSC(FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMD_XT);
_FWDT(FWDTEN_OFF);

int main(void)
{
    RSCONbits.RSE2=1;       /* Enable Receive Time Slot 2 */
    RSCONbits.RSE0=1;       /* Enable Receive Time Slot 0 */


    TSCONbits.TSE2=1;       /* Enable Transmit Time Slot 2 */
    TSCONbits.TSE0=1;       /* Enable Transmit Time Slot 0 */

    DCICON1bits.COFSM = 0; /* Multichannel Frame Sync mode */
    DCICON1bits.DJST = 0;  /* Data TX/RX is begun one serial clock cycle after frame sync pulse */
    DCICON1bits.CSCKE = 0; /* Data changes on rising edge sampled on falling edge of CSCK */
    DCICON1bits.COFSD = 1; /* Frame sync driven by codec */
    DCICON1bits.CSCKD = 1; /* Clock is input to DCI from codec */

    DCICON2bits.BLEN = 1;  /* Two data words will be buffered between interrupts */
    DCICON2bits.COFSG = 2; /* Data frame has 3 words */
    DCICON2bits.WS = 15;   /* Data word size is 16 bits*/

    DCICON3 = 0;           /* BCG value is zero since clock is driven by codec */


    IPC15bits.DCIIP=6;     /* Enable the interrupts */
    IFS3bits.DCIIF=0;
    IEC3bits.DCIIE=0;

    TXBUF0= 0x0001;        /* This is the data word */
    TXBUF1= 0x0002;        /* This is the control word */

    DCICON1bits.DCIEN = 1;/* Enable the module */

    while(1);
}

void __attribute__((__interrupt__, no_auto_psv)) _DCIInterrupt(void)
{
    int dataWord;
    int statusWord;

    IFS3bits.DCIIF = 0;
    TXBUF0 = 0x0001;       /* Write some data */
    TXBUF1 = 0x0002;       /* This is the control word */

    dataWord = RXBUF0;     /* Read the data word */
    statusWord = RXBUF1;   /* Read the status word */
}
```

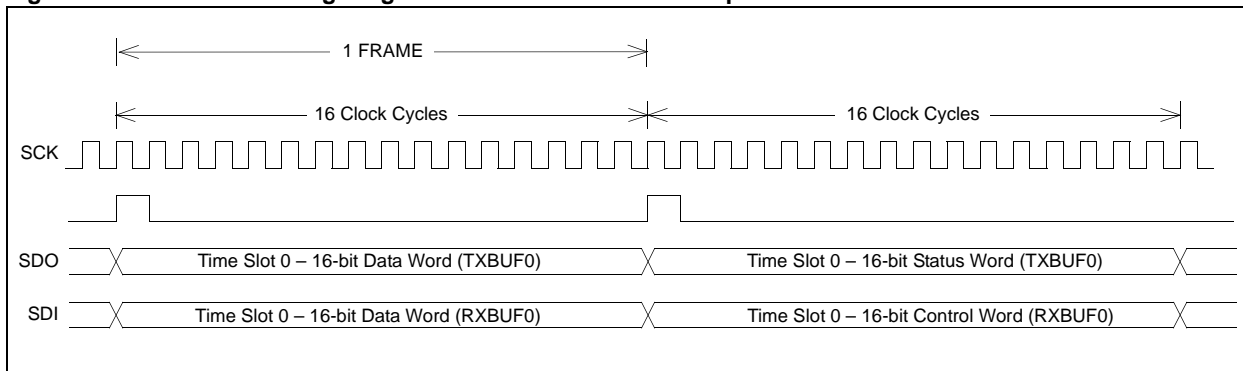## 20.7   DATA TRANSFER TO DCI MODULE BUFFERS USING DMA

The Direct Memory Access (DMA) module on dsPIC33F devices can be used to transfer data from DPSRAM to the DCI module buffers without user application intervention. At least two DMA channels would be needed for this purpose. One DMA channel reads data from the receive registers while the other channel writes data to the transmit registers. Both DMA channels use the DCI Transfer Done interrupt.

Since the DCI RXBUF and TXBUF registers are 16-bit registers, the DMA channels should be set up for word transfer. To write byte values to the DCI module, user software must first left-shift them to the upper byte of the word.

Example 20-2 shows the code that configures the DMA for continuous Ping-Pong Buffer mode. Figure 20-37 shows the timing diagram of the DCI codec communication. In Ping-Pong Buffer mode, the DMA module alternates the memory locations where the data frames are stored. This mechanism facilitates processing on one data frame while a processed data frame is being transmitted and a new data frame is being received. The DCI module requests the DMA module for a transfer on every transfer complete interrupt.

For details on the DMA module, refer to **Section 22. "Direct Memory Access (DMA)"** (DS70182).

Figure 20-37:   Codec Timing Diagram for DCI-DMA Code Example

Example 20-2: Data Transfer to DCI Module Buffers Using DMA Code Example

```
#include "p33Fxxxx.h"

/* Device configuration registers */
_FGS(GWRP_OFF & GCP_OFF);
_FOSCSEL(FNOSC_PRIPLL);
_FOSC(FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMD_XT);
_FWDT(FWDTEN_OFF);

#define FCY 40000000
#define CODEC_SAMPLE_RATE 8000
#define DCI_BCG_VALUE( ( (FCY/32) / CODEC_SAMPLE_RATE ) - 1 )
#define FRAME 80

int txBufferA[FRAME] __attribute__((space(dma)));
int txBufferB[FRAME] __attribute__((space(dma)));
int rxBufferA[FRAME] __attribute__((space(dma)));
int rxBufferB[FRAME] __attribute__((space(dma)));

volatile int rxBufferIndicator = 0;

void DCIInit(void);
void processRxData(int * sourceBuffer, int * targetBuffer);
void DMAInit(void);

int main (void)
{
    CLKDIV = 0;                 /* Set up for 40 MIPS */
    PLLFBD = 30;
    while (!OSCCONbits.LOCK);

    DMAInit();
    DCIInit();

    while(1);

}

void DCIInit(void)
{
    TSCON = 0x0001;            /* Only one transmit time slot */
    RSCON = 0x0001;            /* Only one receive time slot */

    DCICON1 = 0;
    DCICON1bits.DCIEN = 1;     /* Module is enabled */
    DCICON1bits.DCISIDL = 0;   /* Continue operation in idle */
    DCICON1bits.DLOOP = 0;     /* Loopback mode is disabled */
    DCICON1bits.CSCKD = 0;     /* DCI is master - CSCK is output */
    DCICON1bits.CSCKE = 0;     /* Data is sampled on falling edge */
    DCICON1bits.COFSD = 0;     /* DCI is master - COFS is output */
    DCICON1bits.UNFM = 0;      /* Transmit zeroes on TX underflow */
    DCICON1bits.CSDOM = 0;     /* Transmit 0 on disabled time slots */
    DCICON1bits.DJST = 1;      /* COFS and CSDO start together */
    DCICON1bits.COFSM = 0;     /* DCI mode is multi-channel FS mode */

    DCICON2 = 0;
    DCICON2bits.BLEN = 0;      /* Interrupt on one buffer */
    DCICON2bits.COFSG = 0;     /* Data frame has one word */
    DCICON2bits.WS = 0xF;      /* Word size is 16 bits */

    DCICON3 = DCI_BCG_VALUE;

    _DCIIE = 0;                /* Disabled since DMA is used */

}
void DMAInit(void)
{
    /* DMA 0 - DPSRAM to DCI */
```

Example 20-2:   Data Transfer to DCI Module Buffers Using DMA Code Example (Continued)

```
    DMA0CONbits.SIZE = 0;      /* Word transfers */
    DMA0CONbits.DIR = 1;       /* From DPSRAM to DCI */
    DMA0CONbits.AMODE = 0;     /* Register Indirect with post-increment mode */
    DMA0CONbits.MODE = 2;      /* Continuous ping pong mode enabled */
    DMA0CONbits.HALF = 0;      /* Interrupt when all the data has been moved */
    DMA0CONbits.NULLW = 0;
    DMA0REQbits.FORCE = 0;     /* Automatic transfer */
    DMA0REQbits.IRQSEL = 0x3C;/* Codec transfer done */

    DMA0STA =__builtin_dmaoffset(txBufferA);
    DMA0STB =__builtin_dmaoffset(txBufferB);

    DMA0PAD = (int)&TXBUF0;
    DMA0CNT = FRAME-1;

    /* DMA 2 - DCI to DPSRAM */

    DMA2CONbits.SIZE = 0;      /* Word transfers */
    DMA2CONbits.DIR = 0;       /* From DCI to DPSRAM */
    DMA2CONbits.HALF = 0;      /* Interrupt when all the data has been moved */
    DMA2CONbits.NULLW = 0;     /* No NULL writes - Normal Operation */
    DMA2CONbits.AMODE = 0;     /* Register Indirect with post-increment mode */
    DMA2CONbits.MODE = 2;      /* Continuous mode ping pong mode enabled */

    DMA2REQbits.FORCE = 0;     /* Automatic transfer */
    DMA2REQbits.IRQSEL = 0x3C;/* Codec transfer done */

    DMA2STA =__builtin_dmaoffset(rxBufferA);
    DMA2STB =__builtin_dmaoffset(rxBufferB);

    DMA2PAD = (int)&RXBUF0;
    DMA2CNT = FRAME-1;

    _DMA2IP = 5;
    _DMA2IE = 1;

    DMA0CONbits.CHEN = 1;      /* Enable the channel */
    DMA2CONbits.CHEN = 1;
}

void processRxData(int * sourceBuffer, int * targetBuffer)
{
    /* This procedure loops back the received data to the */
    /* the codec output. The user application could process */
    /* this data as per application requirements. */

    int index;

    for(index = 0;index < FRAME;index ++)
    {
        targetBuffer[index] = sourceBuffer[index];
    }

}

void __attribute__((__interrupt__,no_auto_psv)) _DMA2Interrupt(void)
{
    _DMA2IF = 0;               /* Received one frame of data */

    if(rxBufferIndicator == 0)
    {
        processRxData(rxBufferA,txBufferA);
    }
    else
    {
        processRxData(rxBufferB,txBufferB);
    }
    rxBufferIndicator ^= 1;    /* Toggle the indicator */

}
```

20

**Data Converter
Interface (DCI)**

20.8   OPERATION IN POWER-SAVING MODES

20.8.1   CPU Idle Mode

The DCI module can optionally continue to operate while the CPU is in Idle mode. The DCISIDL control bit (DCICON1<13>) determines whether the DCI module operates when the CPU is in Idle mode.

*   If the DCISIDL control bit is cleared (default), the module continues to operate normally in Idle mode
*   If the DCISIDL bit is set, the module halts when the CPU enters Idle mode

20.8.2   Sleep Mode

The DCI will not operate while the device is in Sleep mode if the CSCK signal is derived from the device instruction clock, T_{CY}.

However, the DCI module can operate while in Sleep mode and wake the CPU when the CSCK signal is supplied by an external device (CSCKD = 1). The DCI Interrupt Enable bit, DCIIE, must be set to allow a wake-up event from Sleep mode. When the DCI Interrupt Flag, DCIIF, is set, the device wakes up from Sleep mode. If the DCI interrupt priority level is greater than the current CPU priority, program execution resumes from the DCI ISR. Otherwise, execution resumes with the instruction following the PWRSAV instruction that previously entered Sleep mode.

20.8.3   Doze Mode

The DCI module is not affected by Doze mode. However, the processor may not have sufficient time to respond to a DCI interrupt while in Doze mode.

20.9 REGISTERS ASSOCIATED WITH DCI

Table 20-2 lists the registers associated with the DCI module.

Table 20-2: DCI Register Map

Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on all Resets
IFS2	—	—	—	FLTBIF	FLTAIF	LVDIF	DCIIF	QEIIF	PWMIF	C2IF	INT4IF	INT3IF	OC8IF	OC7IF	OC6IF	OC5IF	0000 0000 0000 0000
IEC2	—	—	—	FLTBIE	FLTAIE	LVDIE	DCIIE	QEIIE	PWMIE	C2IE	INT4IE	INT3IE	OC8IE	OC7IE	OC6IE	OC5IE	0000 0000 0000 0000
IPC10	—	FLTAIP<2:0>			—	LVDIP<2:0>			—	DCIIP<2:0>			—	QEIIP<2:0>			0100 0100 0100 0100
DCICON1	DCIEN	—	DCISIDL	—	DLOOP	CSCKD	CSCKE	COFSD	UNFM	CSDOM	DJST	—	—	—	COFSM<1:0>		000- -000 000- --00
DCICON2	—	—	—	—	BLEN<1:0>		—	COFSG<3:0>				—	WS<3:0>				---- 00-0 000- 0000
DCICON3	—	—	—	—	BCG<11:0>												---- 0000 0000 0000
DCISTAT	—	—	—	—	SLOT<3:0>				—	—	—	—	ROV	RFUL	TUNF	TMPTY	---- 0000 ---0 0000
TSCON	TSE15	TSE14	TSE13	TSE12	TSE11	TSE10	TSE9	TSE8	TSE7	TSE6	TSE5	TSE4	TSE3	TSE2	TSE1	TSE0	0000 0000 0000 0000
RSCON	RSE15	RSE14	RSE13	RSE12	RSE11	RSE10	RSE9	RSE8	RSE7	RSE6	RSE5	RSE4	RSE3	RSE2	RSE1	RSE0	0000 0000 0000 0000
RXBUF0	Receive 0 Data Register																uuuu uuuu uuuu uuuu
RXBUF1	Receive 1 Data Register																uuuu uuuu uuuu uuuu
RXBUF2	Receive 2 Data Register																uuuu uuuu uuuu uuuu
RXBUF3	Receive 3 Data Register																uuuu uuuu uuuu uuuu
TXBUF0	Transmit 0 Data Register																0000 0000 0000 0000
TXBUF1	Transmit 1 Data Register																0000 0000 0000 0000
TXBUF2	Transmit 2 Data Register																0000 0000 0000 0000
TXBUF3	Transmit 3 Data Register																0000 0000 0000 0000

Legend: r = Reserved, x = Unknown, u = Unchanged. Shaded locations indicate reserved space in SFR map for future module expansion. Read reserved locations as '0's.

Note: SFR addresses may vary from one dsPIC33F device to another. For more information, consult the device specific data sheet.

20.10   DESIGN TIPS

Question 1: *Can the DCI support data word lengths greater than 16 bits?*

Answer: Yes. A long data word can be transmitted and received using multiple transmit and receive registers. For details, refer to **20.5.3 "Data Packing for Long Data Word Support"**.

Question 2: *Can the dsPIC33F use the codec clock source?*

Answer: Yes. The codec clock must be connected to the correct clock input pin on the dsPIC33F device. This synchronizes the dsPIC33F with the codec.

Question 3: *What is HD Audio, and is it supported by the DCI module?*

Answer: Intel® High Definition Audio, is a new protocol for codec interfaces. The DCI module does not support this protocol.

20.11 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 dsPIC33F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the DCI module are:

Title   **Application Note #**

No related application notes at this time.

> **Note:** For additional Application Notes and code examples for the dsPIC33F device family, visit the Microchip web site (www.microchip.com).

20.12   REVISION HISTORY

Revision A (May 2007)

This is the initial released version of this document.

Revision B (September 2008)

This revision incorporates the following content updates:

- Figures:
  - Data Packing Example for Long Data Words (see Figure 20-26): TXBUF0 contents reads the Data Word as 23:8.
- Additional minor corrections such as language and formatting updates are incorporated in the entire document.