OMAP5912 Multimedia Processor Multichannel Buffered Serial Ports (McBSPs) Reference Guide

Literature Number: SPRU762C
March 2005

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI’s standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Preface

Read This First

About This Manual

This document describes the three multichannel buffered serial ports (McBSPs) available on the OMAP5912 device.

Notational Conventions

This document uses the following conventions.

❑ Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

Documentation that describes the OMAP5912 device, related peripherals, and other technical collateral, is available in the OMAP5912 Product Folder on TI’s website: www.ti.com/omap5912.

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

Figures

Tables

This page is intentionally left blank.

Multichannel Buffered Serial Ports (McBSPs)

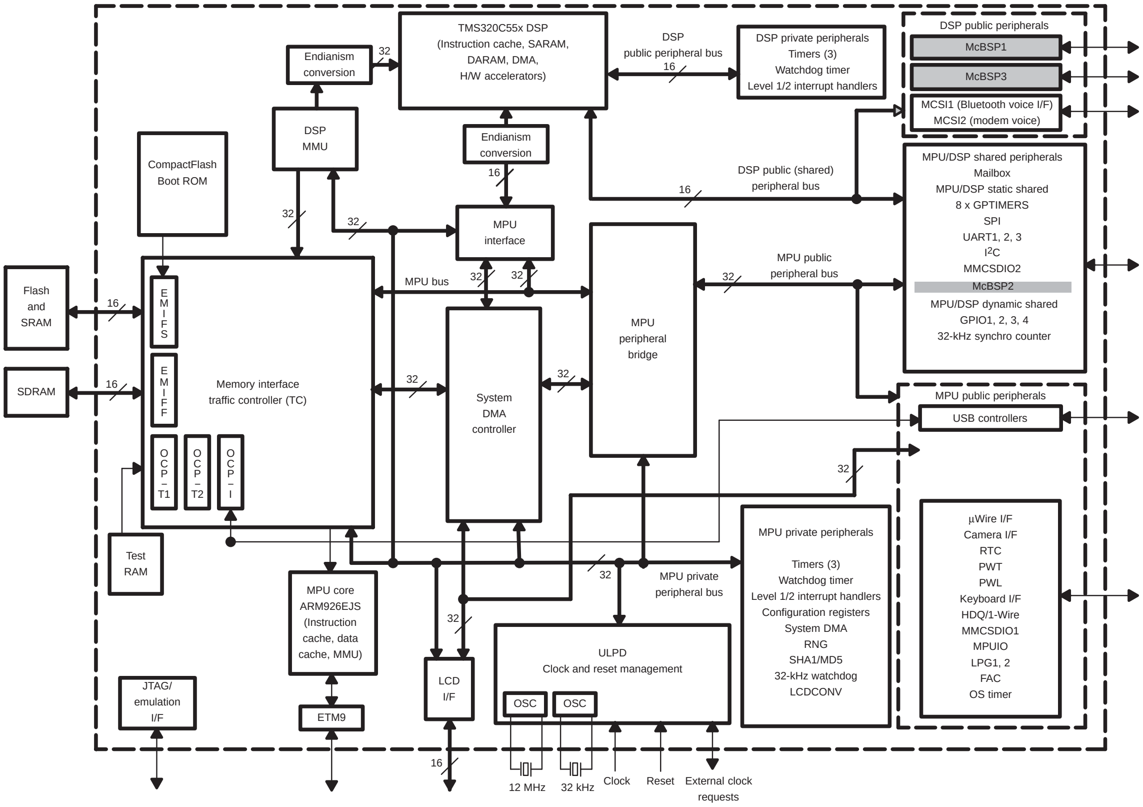
This document describes the three multichannel buffered serial ports (McBSPs) available on the OMAP5912 device.

1 Introduction/Feature Overview

The OMAP5912 device provides three application-specific multichannel buffered serial ports (McBSPs) that allow direct interface to audio codecs and other similar devices in a system.

Figure 1 shows the processor with the McBSP area highlighted.

Figure 1. McBSP Area



Section 2 describes the details of McBSP architecture and structure. Sections 3 through 6 detail major supported operational use cases for the OMAP5912 McBSPs. Section 7 describes the McBSP registers.

Note:

McBSP information applicable for all three McBSPs on the OMAP5912 device will be referred to as McBSP. Information for a specific McBSP will be presented either as McBSP1, McBSP2, or McBSP3.

1.1 Purpose of the Peripheral

McBSP1 and McBSP3 are primarily used as serial communication ports to interface to audio codecs, analog interface chips (AICs), and A/D and D/A devices. They are DSP public peripherals, and hence can be directly accessed by the DSP and DSP DMA. McBSP1 and McBSP3 may also be accessed by the MPU and system DMA controller via the MPU interface. The interface clock source for these two McBSPs can be either DSPXOR_CK or DSPPER_CK.

McBSP2 can be also used to interface to audio codecs, analog interface chips (AICs), and A/D and D/A devices. In addition, it features independent framing and clocking pins for receive and transmit. McBSP2 is a shared peripheral, and is connected to both the MPU public peripheral bus and the DSP public peripheral bus. The interface clock source for McBSP2 is ARMPER_CK.

For examples and more details on the supported McBSP use cases, see sections 3 through 6.

1.2 Key Features of the McBSPs

All three McBSPs feature:

- ❑ Double-buffered transmission and triple-buffered reception, which allow a continuous data stream.
- ❑ DSP DMA and system DMA support for both receive and transmit transfers.
- ❑ Multichannel selection modes that enable or disable block transfers in each of the channels using the DXZ pin, with 128 channels for transmission and for reception.
- ❑ Programmable polarity for frame-synchronization pulses and for clock signals. (For slave mode restrictions, see section 2.2.3.4.)
- ❑ µ-law and A-law companding.

- ❑ Status bits for flagging exception/error conditions.
- ❑ A programmable sample rate generator for internal generation and control of clock signals and frame-synchronization signals.
- ❑ The capability to use the McBSP pins as general-purpose I/O pins.
- ❑ Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices, configured as master or slave.
- ❑ A wide selection of data sizes: 8, 12, 16, 20, 24, and 32 bits.

Note:

A value of the chosen data size is referred to as a *serial word* or *word* throughout the McBSP documentation. Elsewhere, *word* is used to describe a 16-bit value.

Additional McBSP1 features/limitations

- ❑ The sample rate generator has the option of utilizing an external reference clock provided on the CLKS pin, but does not feature a CLKR or a FSR pin.
- ❑ Master/slave for transmission, slave for reception

Additional McBSP2 features/limitations

- ❑ Full duplex communication with independent external clocking and framing for reception and transmission: CLKX, CLKR, FSX, FSR pins available.
- ❑ System DMA only supports 16-bit transfers to McBSP2. However, the DSP DMA does support up to 32-bit transfers.

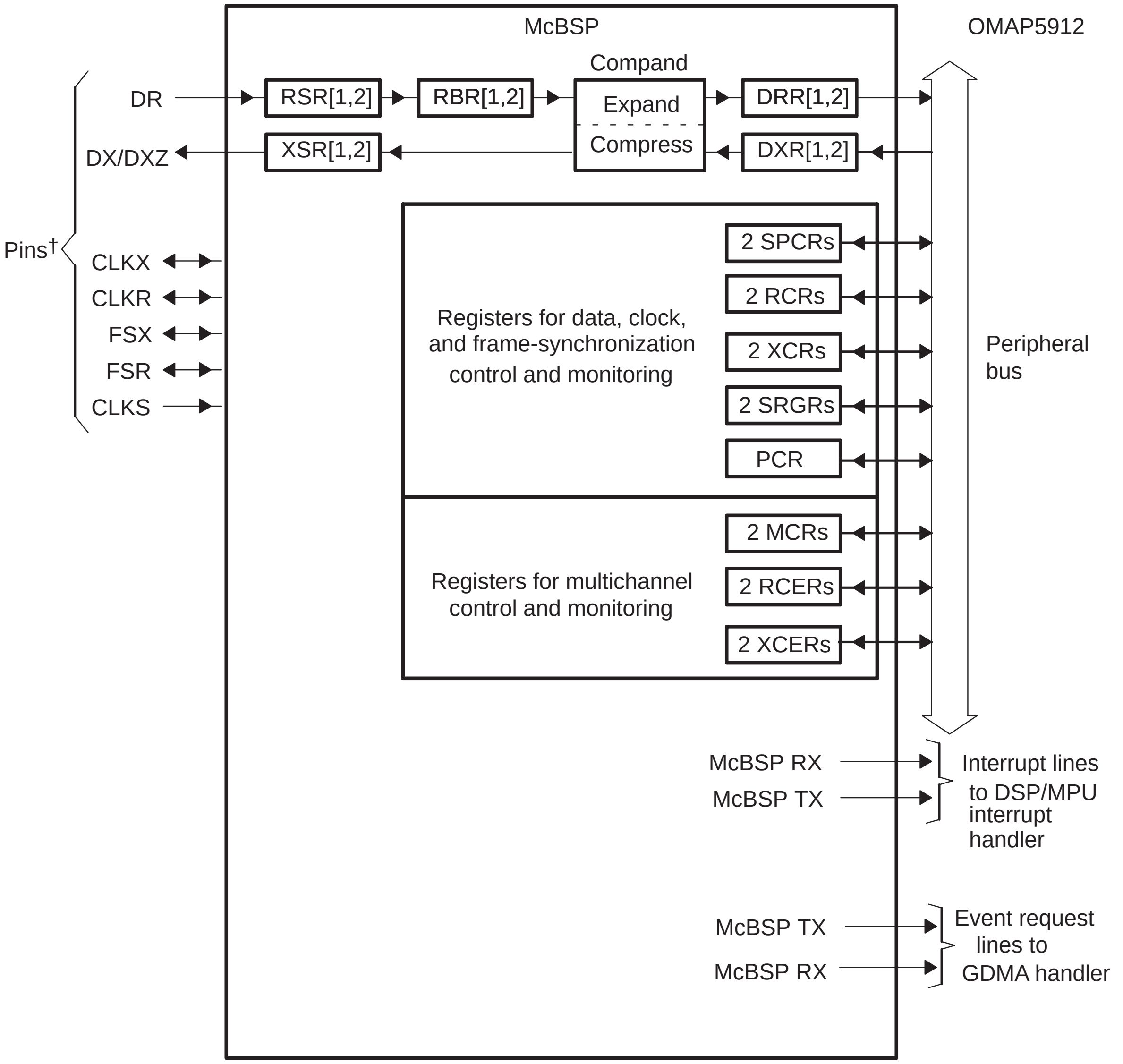
Additional McBSP3 features/limitations

- ❑ Only supports 4-pin connection
- ❑ It does not feature a CLKR or a FSR pin.

1.3 Functional Block Diagram

The McBSP consists of a data-flow path and a control path connected to external devices by a set of pins that depends on the particular McBSP used. Figure 2 shows a conceptual block diagram displaying all the pins that are potentially available on the McBSP module. Note that not all pins are available on every McBSP. For the McBSP signal names and pins available on each McBSP, see section 2.3.

Figure 2. McBSP Functional Block Diagram


McBSP
OMAP5912
Compand
DR
RSR[1,2]
RBR[1,2]
Expand
Compress
DRR[1,2]
DX/DXZ
XSR[1,2]
DXR[1,2]
Pins†
CLKX
CLKR
FSX
FSR
CLKS
Registers for data, clock, and frame-synchronization control and monitoring
2 SPCRs
2 RCRs
2 XCRs
2 SRGRs
PCR
Peripheral bus
Registers for multichannel control and monitoring
2 MCRs
2 RCERs
2 XCERs
McBSP RX
McBSP TX
Interrupt lines to DSP/MPU interrupt handler
McBSP TX
McBSP RX
Event request lines to GDMA handler

† Each McBSP features a different set of available pins. For pin availability and for signal routing for non-available pins, see sections 1.3.1, 1.3.2, and 1.3.3.

Serial data is communicated to devices interfaced with the McBSP via the data transmit DX pin (or DXZ in multichannel mode) for transmission and via the data receive (DR) pin for reception. Control information in the form of bit clock and frame synchronization is communicated via the following pins: CLKX (bit transmit clock), CLKR (bit receive clock), FSX (transmit frame synchronization), and FSR (receive frame synchronization). See section 2.3 for a list of available pins for each McBSP and for the pin naming convention for this document.

The DSP core, MPU core, and DSP/system DMA controller communicate with the McBSP through 16-bit-wide registers accessible via the internal peripheral bus. The DSP core, MPU core, or DSP/system DMA controller writes the data to be transmitted to the data transmit registers (DXR1, DXR2). Data written to the DXRs is shifted out to DX (or DXZ, in multichannel mode) via the transmit shift registers (XSR1, XSR2). Similarly, receive data on the DR pin is shifted into the receive shift registers (RSR1, RSR2) and copied into the receive buffer registers (RBR1, RBR2). The contents of the RBRs is then copied to the DRRs, which can be read by the DSP core, MPU core, or DSP/system DMA controller. This allows simultaneous movement of internal and external data communications.

DRR2, RBR2, RSR2, DXR2, and XSR2 are not used (written, read, or shifted) if the serial word length is 8 bits, 12 bits, or 16 bits. For larger word lengths, these registers are needed to hold the most significant bits. The remaining registers in Figure 2 control McBSP operation. The DSP DMA and system DMA source/destination address should be programmed to DRR2/DXR2 for transfers larger than 16-bits, or DRR1/DXR1 for transfers of 16-bits or smaller.

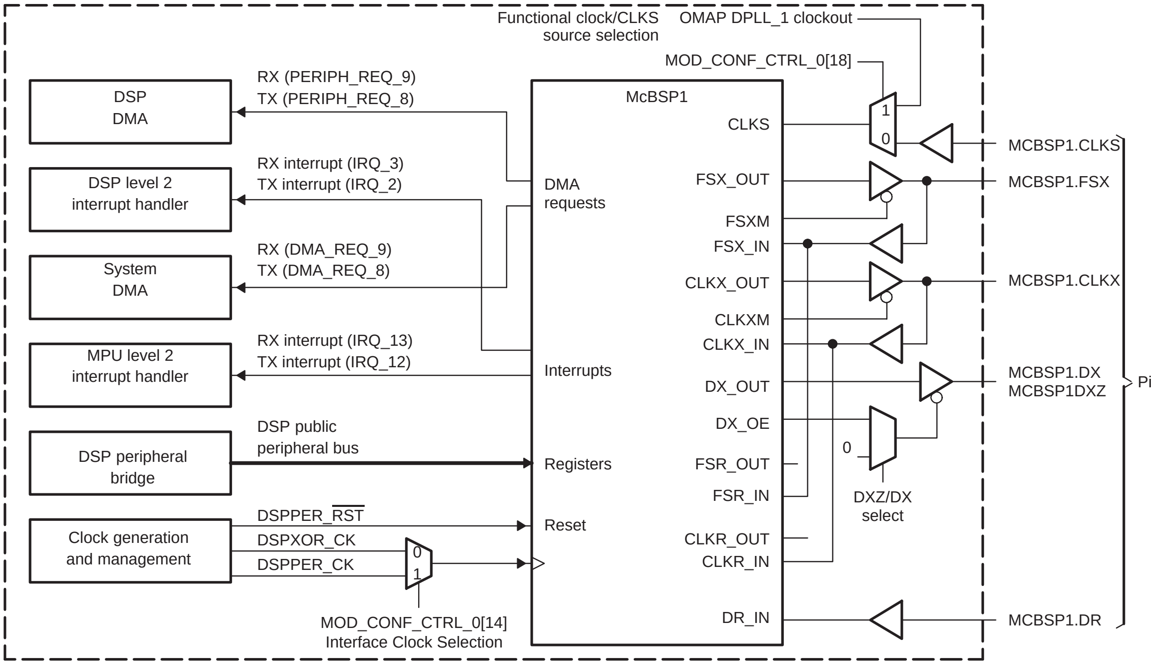
Note:

The transmit output (DX) pins do not go to high impedance when the transmitter is not actively sending data. In other words, the OMAP5912 device always actively drives the DX pins. Therefore, the dedicated pin for multichannel mode is the DXZ pin, which will go to high impedance when a given channel is disabled.

1.3.1 McBSP1 Functional Diagram

Figure 3 shows the McBSP1 functional diagram. It displays the internal signal routing (external to the McBSP module), the external interface pins, OMAP5912 internal bus, internal clock source, DSP/system DMA event requests (OMAP5912), and interrupts associated with McBSP1.

Figure 3. McBSP1 Functional Diagram



The FSR_in is internally connected to the FSX_in, and CLKR_in is internally connected to the CLKX_in signal as shown in Figure 3. The Module Configuration Control 0 Register controls clock selection for this McBSP.

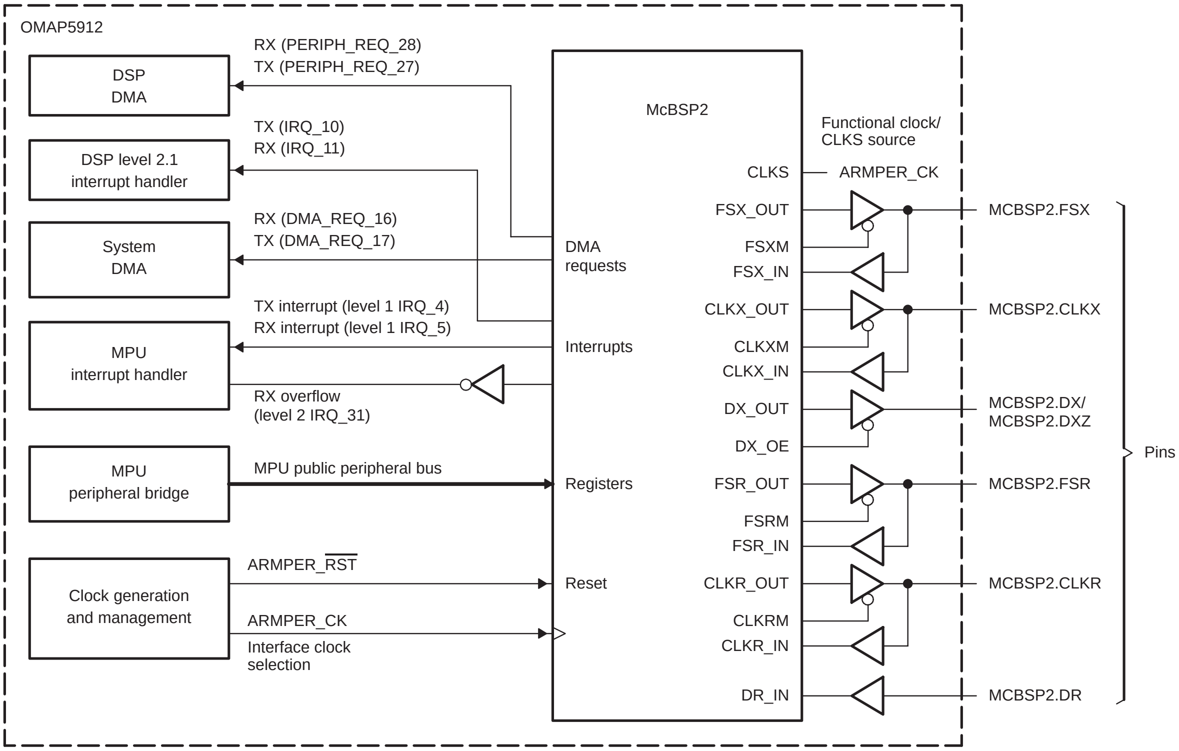
McBSP1 can be configured as a master/slave for transmission, but only as a slave for reception. McBSP1 also features the CLKS pin, which serves as an external source to the Sample Rate Generator. See Figure 6 for details on frame, clock, and data signals within the McBSP module.

McBSP1 must have both the functional clock and the interface clock enabled for proper operation. The functional clock is the McBSP module clock, which is used for the peripheral internal logic clocking. The interface clock is used to clock data movement in the peripheral bus. Figure 3 shows the clock selection for McBSP1. For more details on clock generation and management, see *OMAP5912 Multimedia Processor Initialization Reference Guide* (SPRU752) and *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

1.3.2 McBSP2 Functional Diagram

Figure 4 shows the McBSP2 functional diagram. It displays the external interface pins, OMAP5912 internal bus, internal clock source, DSP/system DMA event requests (OMAP5912), and interrupts associated with McBSP2.

Figure 4. McBSP2 Functional Diagram



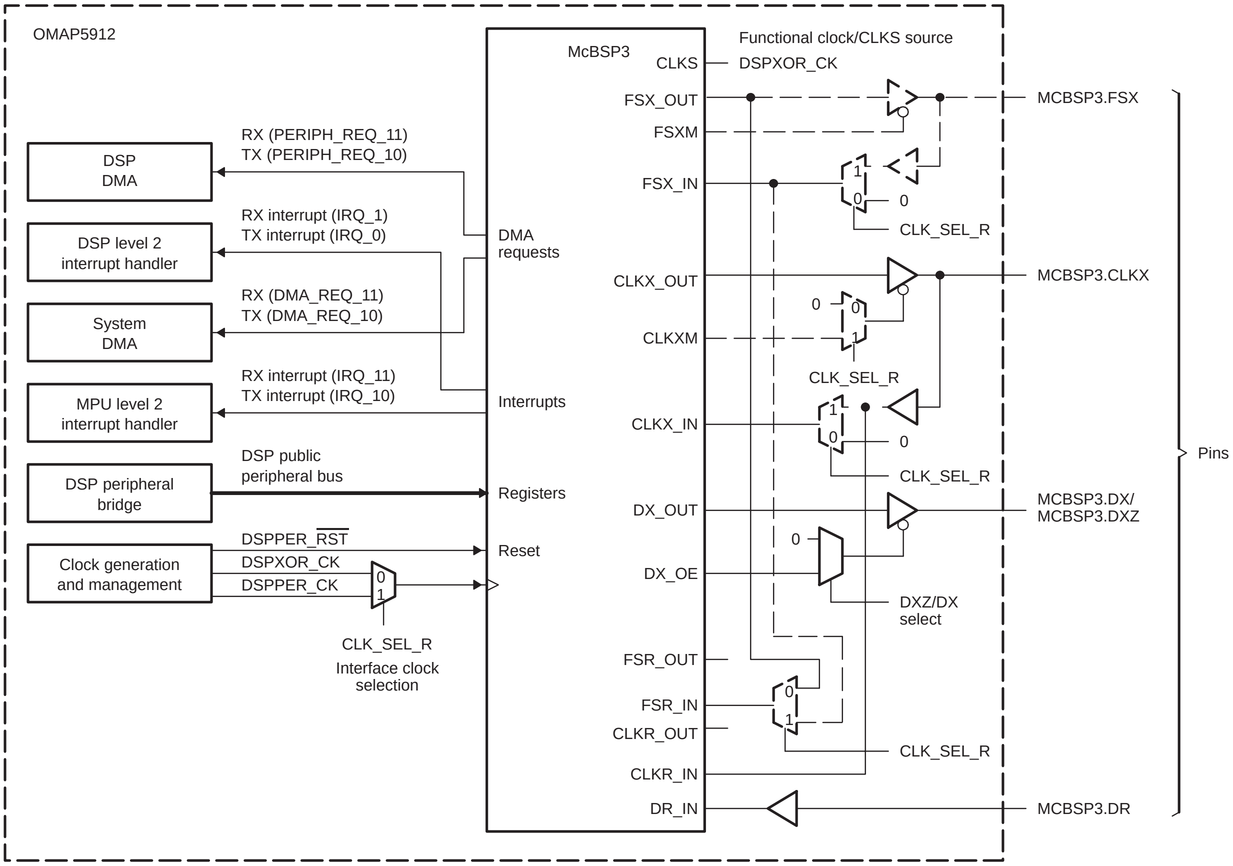
McBSP2 features CLKR and FSR pins that allow full duplex communications with independent reception and transmission clock and frame signals. See Figure 6 for details on frame, clock, and data signals within the McBSP module. The system DMA only supports 16-bit transfers to the McBSP2 data register.

McBSP2 must have both the functional clock and the interface clock enabled for proper operation. The functional clock is the McBSP module clock, which is used for the peripheral internal logic clocking. The interface clock is used to clock data movement in the peripheral bus. Figure 4 shows the clock selection for McBSP2. The Module Configuration Control 0 Register controls the clock selection for this McBSP. For more details on clock generation and management, see the *OMAP5912 Multimedia Processor Initialization Reference Guide* (SPRU752) and the *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

1.3.3 McBSP3 Functional Diagram

Figure 5 shows the McBSP3 functional diagram. It displays the external interface pins, OMAP5912 internal bus, internal clock source, DSP/system DMA event requests (OMAP5912), and interrupts associated with McBSP3.

Figure 5. McBSP3 Functional Diagram



McBSP3 can be configured in two interface modes. The solid lines show the 3 pins mode reset connection configuration. The dotted lines show the typical 4 pins mode configuration.

❑ 3 pins mode (default): This is the internal routing default mode for McBSP3 coming out of a device reset. This configuration is not supported.

❑ 4 pins mode (typical configuration): In this case, McBSP3 can be configured as master/slave for transmission, but only as a slave for reception. This mode utilizes the paths shown as dashed lines in Figure 5.

The signal routing for this mode is the following: MCBSP3.FSX and MCBSP3.CLKX are bidirectional pins, and FSR and CLKR are not available as pins (internal feedback from FSX and CLKX, respectively).

Setting bit 19 (CONF_MOD_MCBSP3_CLK_SEL_R) in the CONF_MOD_CTRL_0 register selects the 4 pin mode interface and interface clock. This is illustrated as CLK_SEL_R in Figure 5. See Figure 6 for details on frame, clock, and data signals within the McBSP module.

McBSP3 must have both the functional clock and the interface clock enabled for proper operation. The functional clock is the McBSP module clock, which is used for the peripheral internal logic clocking. The interface clock is used to clock data movement in the peripheral bus. Figure 5 shows the clock selection for McBSP3. For more details on clock generation and management, see the *OMAP5912 Multimedia Processor Initialization Reference Guide* (SPRU752) and the *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

1.4 Supported Use Case Statement

The McBSPs in the OMAP5912 device are intended to operate on typical industry-standard audio codecs such as I2S compliant devices, as well as analog interface chips (AICs), and other serially connected A/D and D/A devices that typically run on the operational frequency similar to the examples provided in sections 3 through 6:

- Section 3: Interfacing McBSP2 to Two AIC33 Devices
- Section 4: McBSP3 SPI Mode
- Section 5: Interfacing McBSP1 to Another Multichannel Device
- Section 6: Interfacing McBSP2 to Two Codecs With the DSP

Using the McBSPs in configurations that significantly deviate from the presented examples is not supported.

1.5 Industry Standards Compliance Statement

The OMAP5912 McBSPs may be successfully interfaced with the following industry standards:

- I2S Compliant
  - Transmitter and receiver device need to agree on the word size.
- SPI Compatible
  - McBSP SPI Master mode cannot support multiple slaves due to the absence of the slave-enable signal.
  - Programmable element length, but fixed for given transfer.
  - SPI settings supported: (See section 2.4 for details)
    - Clock low inactive state without delay: The McBSP transmits data on the rising edge of CLKX and receives data on the falling edge of CLKR.
    - Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of CLKX and receives data on the rising edge of CLKR.
    - High inactive state without delay: The McBSP transmits data on the falling edge of CLKX and receives data on the rising edge of CLKR.
    - High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of CLKX and receives data on the falling edge of CLKR.

Sections 3 through 6 provide examples of how to configure the McBSPs to interface with these industry standards.

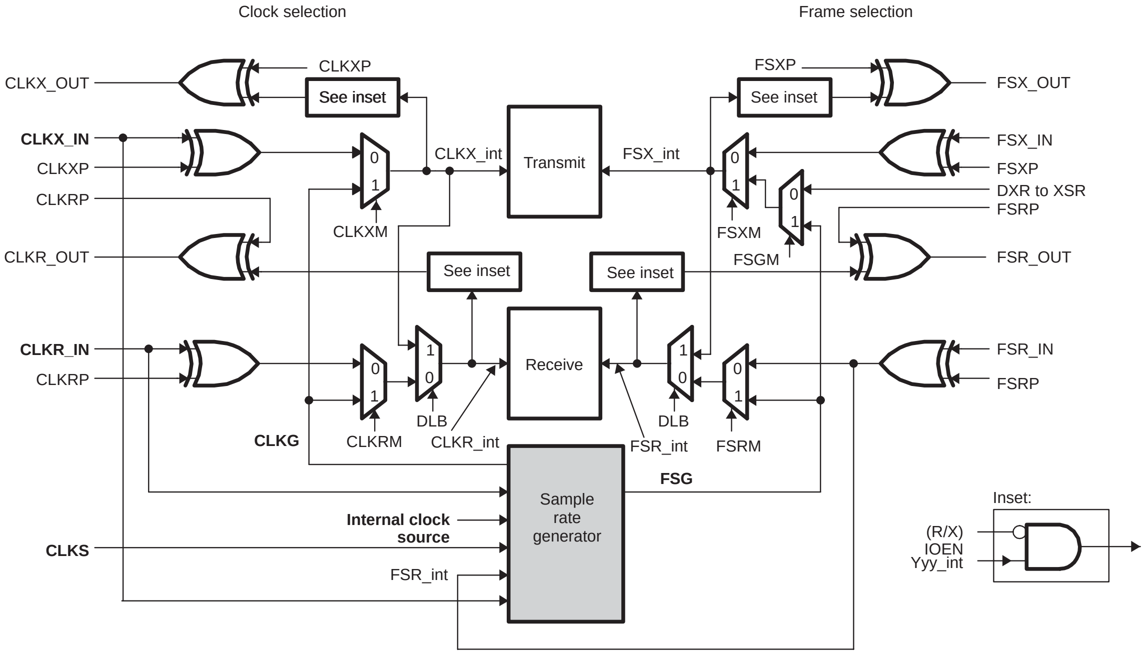
2 McBSP Architecture

This section describes the architectural and structural details of the McBSP, including the clock control, general operation overview, signal descriptions, receiver and transmitter configuration, interrupt and DSP/system DMA event request support, as well as supported McBSP modes and features.

2.1 Clock Control: McBSP Sample Rate Generator

Each McBSP contains a sample rate generator that can be used to generate an internal data clock (CLKG) and an internal frame-synchronization signal (FSG). CLKG can be used for bit shifting on the data receive (DR) pin and/or the data transmit (DX) pin. FSG can be used to initiate frame transfers on DR and/or DX. Figure 6 is a conceptual block diagram of the clock and frame generation for the McBSP.

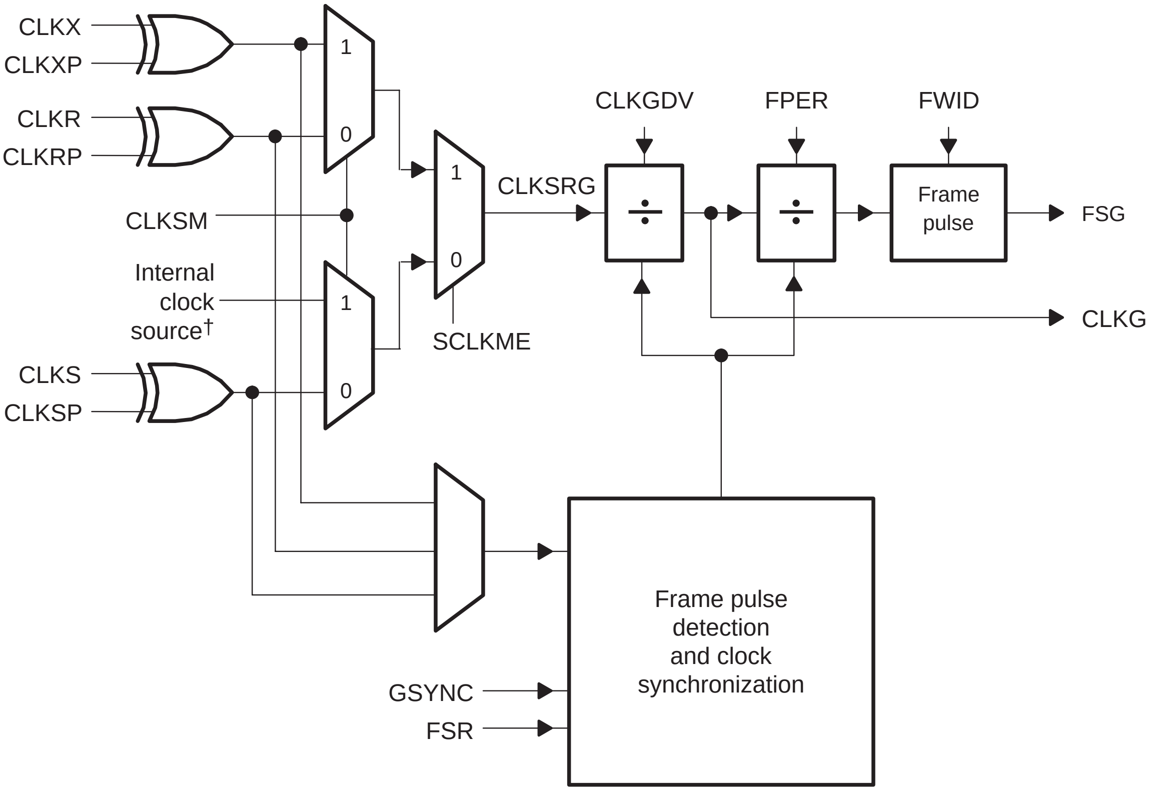
Figure 6. Clock and Frame Generation



CLKR_int, CLKX_int, FSR_int, and FSX_int represent the internal CLKR, internal CLKX, internal FSR, and internal FSX signal respectively. The following sections describe the configuration of these internal clock and frame signals.

Figure 7 is a conceptual block diagram of the sample rate generator.

Figure 7. Conceptual Block Diagram of the Sample Rate Generator



† This source depends on the choice of interface clock.

The source clock for the sample rate generator (labeled CLKSRG in the diagram) can be supplied as follows:

❑ McBSP1: By internal clock source (DSPXOR_CK, DSPPER_CK) or by an external pin (CLKS, CLKX)

❑ McBSP2: By internal clock source (ARMPER_CK) or by an external pin (CLKR, CLKX)

❑ McBSP3: By internal clock source (DSPXOR_CK) or by an external pin (CLKX)

The source is selected with the SCLKME bit in PCR and the CLKSM bit in SRGR2. If an external signal is used, the polarity of the incoming signal can be inverted with the appropriate polarity bit (CLKSP in SRGR2, CLKXP in PCR, or CLKRP in PCR).

The sample rate generator has a three-stage clock divider that gives CLKG and FSG programmability. The three stages provide:

❑ Clock divide-down. The source clock is divided according to the CLKGDV bits in SRGR1 to produce CLKG.

❑ Frame period divide-down. CLKG is divided according to the FPER bits in SRGR2 to control the period from the start of a frame-pulse to the start of the next pulse.

❑ Frame-synchronization pulse-width countdown. CLKG cycles are counted according to the FWID bits in SRGR1 to control the width of each frame-synchronization pulse.

Note:

The McBSP module cannot operate at a frequency faster than 1/2 the functional clock frequency. Choose an input clock frequency and a CLKDV value such that CLKG is less than or equal to 1/2 the functional clock frequency. This is a restriction on the McBSP module itself. To sustain a given data rate on the McBSP, care must be taken with the interface clock frequency and other bus data traffic. For more details, see the usage cases presented in sections 3 through 6.

In addition to the three-stage clock divider, the sample rate generator has a frame-synchronization pulse detection and clock synchronization module that allows synchronization of the clock divide-down with an incoming frame-synchronization pulse on the FSR pin. This feature is enabled or disabled with the GSYNC bit in SRGR2.

For details on getting the sample rate generator ready for operation, see section 2.1.4.

2.1.1 Clock Generation in the Sample Rate Generator

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both. Use of the sample rate generator to drive clocking is controlled by the clock mode bits (CLKRM and CLKXM) in the pin control register (PCR). When a clock mode bit is set (CLKRM = 1 for reception, CLKXM = 1 for transmission), the corresponding data clock (CLKR for reception, CLKX for transmission) is driven by the internal sample rate generator output clock (CLKG). For CLKRM = 0 and CLKXM = 0, the CLKR and CLKX signals need to be driven by an external clock.

The result of CLKRM = 1 and CLKXM = 1 on the McBSP is partially affected by the use of the digital loopback mode and the clock stop (SPI) mode, respectively, as described in Table 1. The digital loopback mode (section 2.5.4) is selected with the DLB bit in SPCR1. The clock stop mode (described in section 2.4) is selected with the CLKSTP bits in SPCR1. When using the sample rate generator as a clock source, ensure the sample rate generator is enabled ($\overline{\text{GRST}}$ = 1).

Table 1. Effects of DLB and CLKSTP on Clock Modes

Mode Bit Settings		**Effect**
CLKRM = 1	DLB = 0 (Digital loopback mode disabled)	CLKR is an output pin driven by the sample rate generator output clock (CLKG).
	DLB = 1 (Digital loopback mode enabled)	CLKR is an output pin driven by internal CLKX. The source for CLKX depends on the CLKXM bit.
CLKXM = 1	CLKSTP = 00b or 01b (Clock stop (SPI) mode disabled)	CLKX is an output pin driven by the sample rate generator output clock (CLKG).
	CLKSTP = 10b or 11b (Clock stop (SPI) mode enabled)	The McBSP is a master in an SPI system. Internal CLKX drives internal CLKR and the shift clocks of any SPI-compliant slave devices in the system. CLKX is driven by the internal sample rate generator.

See Figure 6 for more details on the digital loopback mode.

2.1.1.1 Choosing an Input Clock

The sample rate generator must be driven by an input clock signal from one of the four sources that can be selected with the SCLKME bit in PCR and the CLKSM bit in SRGR2 (see Table 2). When CLKSM = 1, the minimum divide-down value in CLKGDV bits is 1. CLKGDV is described in section 2.1.1.3.

Note:

The McBSP module cannot operate at a frequency faster than 1/2 the functional clock frequency. Choose an input clock frequency and a CLKDV value such that CLKG is less than or equal to 1/2 the functional clock frequency. This is a restriction on the McBSP module itself. To sustain a given data rate on the McBSP, care must be taken with the interface clock frequency and other bus data traffic. For more details, see the usage cases presented in sections 3 through 6.

Table 2. Choosing an Input Clock for the Sample Rate Generator with the SCLKME and CLKSM Bits

SCLKME	CLKSM	Input Clock for Sample Rate Generator
0	0	Signal on CLKS pin
0	1	Internal clock source
1	0	Signal on CLKR pin
1	1	Signal on CLKX pin

For details on pin availability for each McBSP, see section 2.3.

The internal clock source depends on which McBSP is used, as seen in Table 3.

Table 3. Internal Clock Sources for McBSP1, McBSP2, and McBSP3

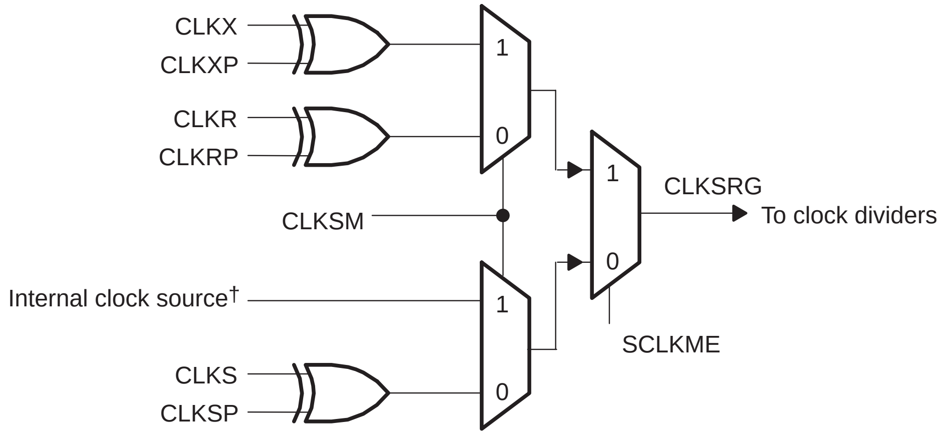
McBSP	Internal Clock Source
McBSP1	DSPXOR_CK or DSPPER_CK
McBSP2	ARMPER_CK
McBSP3	DSPXOR_CK or DSPPER_CK

The Module Configuration Control 0 Register (MOD_CONF_CTRL_0) controls the internal clock selection.

2.1.1.2 Choosing a Polarity for the Input Clock

As shown in Figure 8, when the input clock is received externally from a pin you can choose the polarity of the input clock. The rising edge of CLKSRG generates CLKG and FSG, but you can determine which edge of the input clock causes a rising edge on CLKSRG. Table 4 describes the polarity options.

Figure 8. Possible Inputs to the Sample Rate Generator and the Polarity Bits



† This source depends on the choice of interface clock.

Table 4. Polarity Options for the Input to the Sample Rate Generator

Input Clock	Polarity Option	Effect
Signal on CLKS pin	CLKSP = 0 in SRGR2	Rising edge on CLKS pin generates transitions on CLKG and FSG.
	CLKSP = 1 in SRGR2	Falling edge on CLKS pin generates transitions on CLKG and FSG.
Internal clock source	Always positive polarity	Rising edge of internal clock source generates transitions on CLKG and FSG.
Signal on CLKR pin	CLKRP = 0 in PCR	Falling edge on CLKR pin generates transitions on CLKG and FSG.
	CLKRP = 1 in PCR	Rising edge on CLKR pin generates transitions on CLKG and FSG.
Signal on CLKX pin	CLKXP = 0 in PCR	Rising edge on CLKX pin generates transitions on CLKG and FSG.
	CLKXP = 1 in PCR	Falling edge on CLKX pin generates transitions on CLKG and FSG.

See Figure 6 and Figure 8 for more details.

2.1.1.3 Choosing a Frequency for the Output Clock (CLKG)

The input clock (internal clock source or external clock) can be divided down by a programmable value to drive CLKG. Regardless of the source to the sample rate generator, the rising edge of CLKSRG (see Figure 6) generates CLKG and FSG.

The first divider stage of the sample rate generator creates the output clock from the input clock. This divider stage uses a counter that is preloaded with the divide-down value in the CLKGDV bits in SRGR1. The output of this stage is the data clock (CLKG). CLKG has the frequency represented by the following equation.

Equation 1. CLKG Frequency

$$\text{CLKG frequency} = \frac{\text{Input clock frequency}}{(\text{CLKGDV} + 1)}$$

Thus, the input clock frequency is divided by a value between 1 and 256. When CLKGDV is odd or equal to 0, the CLKG duty cycle is 50%. When CLKGDV is an even value, 2p, representing an odd divide-down, the high-state duration is p+1 cycles and the low-state duration is p cycles.

Note:

The McBSP module cannot operate at a frequency faster than 1/2 the functional clock frequency. Choose an input clock frequency and a CLKDV value such that CLKG is less than or equal to 1/2 the functional clock frequency. This is a restriction on the McBSP module itself. To sustain a given data rate on the McBSP, care must be taken with the interface clock frequency and other bus data traffic. For more details, see the usage cases presented in sections 3 through 6.

2.1.1.4 Keeping CLKG Synchronized to an External Input Clock

When an external signal is selected to drive the sample rate generator (see section 2.1.1.1), the GSYNC bit in SRGR2 and the FSR pin can be used to configure the timing of the output clock (CLKG) relative to the input clock.

GSYNC = 1 ensures that the McBSP and an external device are dividing down the input clock with the same phase relationship. If GSYNC = 1, an inactive-to-active transition on the FSR pin triggers a resynchronization of CLKG and generation of FSG.

For more details about synchronization, see section 2.1.3.

2.1.2 Frame Synchronization Generation in the Sample Rate Generator

The sample rate generator can produce a frame-synchronization signal (FSG) for use by the receiver, the transmitter, or both.

If you want the **receiver** to use FSG for frame synchronization, ensure that FSRM = 1. (When FSRM = 0, receive frame synchronization is supplied via the FSR pin.)

If you want the **transmitter** to use FSG for frame synchronization, you must set the following:

- ❑ FSXM = 1 in PCR: This indicates that transmit frame synchronization is supplied by the McBSP itself rather than from the FSX pin.
- ❑ FSGM = 1 in SRGR2: This indicates that when FSXM = 1, transmit frame synchronization is supplied by the sample rate generator. (When FSGM = 0 and FSXM = 1, the transmitter uses frame-synchronization pulses generated every time data is transferred from DXR[1,2] to XSR[1,2].)

In either case, the sample rate generator must be enabled ($\overline{\text{GRST}}$ = 1) and the frame-synchronization logic in the sample rate generator must be enabled ($\overline{\text{FRST}}$ = 1).

2.1.2.1 Choosing the Width of the Frame-Synchronization Pulse on FSG

Each pulse on FSG has a programmable width. You program the FWID bits in SRGR1, and the resulting pulse width is (FWID + 1) CLKG cycles, where CLKG is the output clock of the sample rate generator.

2.1.2.2 Controlling the Period Between the Starting Edges of Frame-Synchronization Pulses on FSG

You can control the amount of time from the starting edge of one FSG pulse to the starting edge of the next FSG pulse. This period is controlled in one of two ways, depending on the configuration of the sample rate generator.

- ❑ If the sample rate generator is using an external input clock and GSYNC = 1 in SRGR2, FSG pulses in response to an inactive-to-active transition on the FSR pin. Thus, the frame-synchronization period is controlled by an external device.
- ❑ Otherwise, you program the FPER bits in SRGR2, and the resulting frame-synchronization period is (FPER + 1) CLKG cycles, where CLKG is the output clock of the sample rate generator.

2.1.2.3 Keeping FSG Synchronized to an External Clock

When an external signal is selected to drive the sample rate generator (see section 2.1.1.1 on page 28), the GSYNC bit in SRGR2 and the FSR pin can be used to configure the timing of FSG pulses.

GSYNC = 1 ensures that the McBSP and an external device are dividing down the input clock with the same phase relationship. If GSYNC = 1, an inactive-to-active transition on the FSR pin triggers a resynchronization of CLKG and generation of FSG.

See section 2.1.3 for more details about synchronization.

2.1.3 Synchronizing Sample Rate Generator Outputs to an External Clock

The sample rate generator can produce a clock signal (CLKG) and a frame-synchronization signal (FSG) based on an input clock signal that is either the internal clock source signal or a signal at the CLKS, CLKR, or CLKX pin. When an external clock is selected to drive the sample rate generator, the GSYNC bit in SRGR2 and the FSR pin can be used to control the timing of CLKG and the pulsing of FSG relative to the chosen input clock.

Set GSYNC = 1 for the McBSP and an external device to divide down the input clock with the same phase relationship. If GSYNC = 1:

- An inactive-to-active transition on the FSR pin triggers a resynchronization of CLKG and a pulsing of FSG.
- CLKG always begins with a high state after synchronization.
- FSR is always detected at the same edge of the input clock signal that generates CLKG, no matter how long the FSR pulse is.
- The FPER bits in SRGR2 are ignored because the frame-synchronization period on FSG is determined by the arrival of the next frame-synchronization pulse on the FSR pin.

If GSYNC = 0, CLKG runs freely and is not resynchronized, and the frame-synchronization period on FSG is determined by FPER.

2.1.3.1 Synchronization Examples

Figure 9 and Figure 10 show the clock and frame-synchronization operation with various polarities of CLKS (the chosen input clock) and FSR. These figures assume FWID = 0 in SRGR1, for an FSG pulse that is one CLKG cycle wide. The FPER bits in SRGR2 are not programmed; the period from the start of a frame-synchronization pulse to the start of the next pulse is determined by the arrival of the next inactive-to-active transition on FSR. Each of the figures shows what happens to CLKG when it is initially synchronized and GSYNC = 1, and when it is not initially synchronized and GSYNC = 1. The second figure has a slower CLKG frequency (it has a larger divide-down value in the CLKGDV bits in SRGR1).

Figure 9. CLKG Synchronization and FSG Generation When GSYNC = 1 and CLKGDV = 1

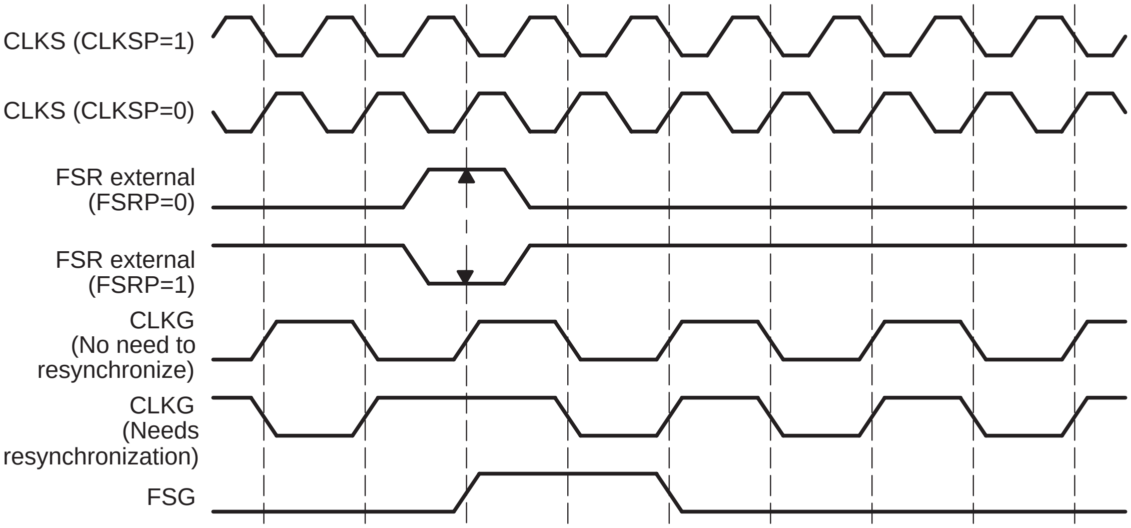
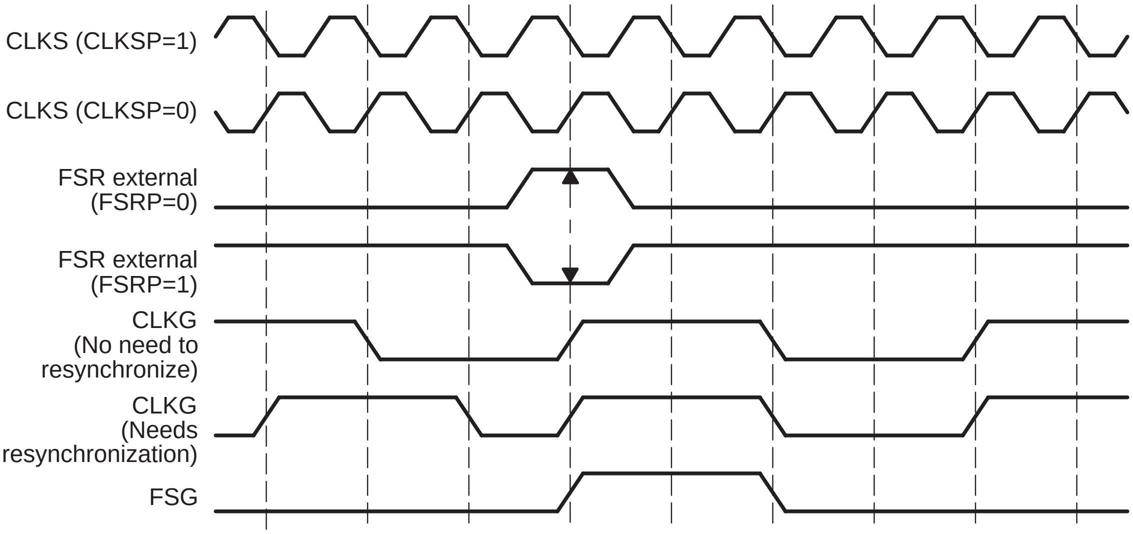


Figure 10. CLKG Synchronization and FSG Generation When GSYNC = 1 and CLKGDV = 3



2.1.4 Reset and Initialization Procedure for the Sample Rate Generator

To reset and initialize the sample rate generator:

Step 1: Place the McBSP/sample rate generator in reset.

During an OMAP5912 device reset, the sample rate generator, the receiver, and the transmitter reset bits ($\overline{\text{GRST}}$, $\overline{\text{RRST}}$, and $\overline{\text{XRST}}$) are automatically forced to 0. Otherwise, during normal operation, the sample rate generator can be reset by setting $\overline{\text{GRST}}$ = 0 in SPCR2, provided that CLKG and/or FSG are not used by any portion of the McBSP. Reset the receiver ($\overline{\text{RRST}}$ = 0 in SPCR1) and the transmitter ($\overline{\text{XRST}}$ = 0 in SPCR2) as well.

If $\overline{\text{GRST}}$ = 0 due to an OMAP5912 device reset, CLKG is driven by the internal clock source divided by 2, and FSG is driven inactive-low. If $\overline{\text{GRST}}$ = 0 due to program code, CLKG and FSG are driven low (inactive).

Step 2: Program the registers that affect the sample rate generator.

Program the sample rate generator registers (SRGR1 and SRGR2) as required for your application. If necessary, other control registers can be loaded with desired values, provided the respective portion of the McBSP (the receiver or transmitter) is in reset.

After the sample rate generator registers are programmed, wait two CLKSRG cycles. This ensures proper synchronization internally.

Step 3: Enable the sample rate generator (take it out of reset).

In SPCR2, set $\overline{\text{GRST}}$ = 1 to enable the sample rate generator.

After the sample rate generator is enabled, wait two CLKG cycles for the sample rate generator logic to stabilize.

On the next rising edge of CLKSRG, CLKG transitions to 1 and starts clocking with a frequency equal to:

Equation 2. CLKG Frequency

$$\text{CLKG frequency} = \frac{\text{Input clock frequency}}{(\text{CLKGDV} + 1)}$$

The input clock is selected with the SCLKME bit in PCR and the CLKSM bit in SRGR2 in one of the configurations in Table 5.

Table 5. Input Clock Selection for Sample Rate Generator

SCLKME	CLKSM	Input Clock for Sample Rate Generator
0	0	Signal on CLKS pin
0	1	Internal clock source
1	0	Signal on CLKR pin
1	1	Signal on CLKX pin

Step 4: If necessary, enable the receiver and/or the transmitter.

If necessary, remove the receiver and/or transmitter from reset by setting $\overline{\text{RRST}}$ and/or $\overline{\text{XRST}}$ = 1.

Step 5: If necessary, enable the frame-synchronization logic of the sample rate generator.

After the transmitter has been preloaded with data (DXR[1/2]), set $\overline{\text{FRST}}$ = 1 in SPCR2 if an internally generated frame-synchronization pulse is required. FSG is generated with an active-high edge after the programmed number of CLKG clocks (FPER + 1) have elapsed.

2.2 McBSP General Operation

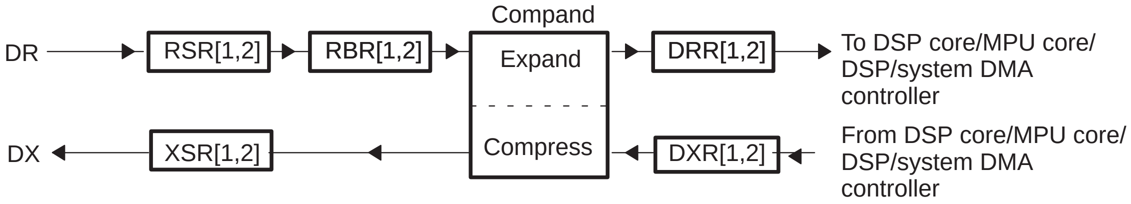
This section addresses the following topics:

- Data transfer process
- Clocking and framing data
- Frame phases
- McBSP reception
- McBSP transmission
- Interrupts and DSP/system DMA event requests generated by McBSPs

2.2.1 Data Transfer Process

Figure 11 shows a diagram of the McBSP data transfer paths. The McBSP receive operation is triple-buffered, and transmit operation is double-buffered. The use of registers varies, depending on whether the defined length of each serial word is 16 bits.

Figure 11. McBSP Data Transfer Paths


Compand
DR
RSR[1,2]
RBR[1,2]
Expand
DRR[1,2]
To DSP core/MPU core/ DSP/system DMA controller
DX
XSR[1,2]
Compress
DXR[1,2]
From DSP core/MPU core/ DSP/system DMA controller

2.2.1.1 Data Transfer Process for Word Length of 8, 12, or 16 Bits

If the word length is 16 bits or smaller, only one 16-bit register is needed at each stage of the data transfer paths. The registers DRR2, RBR2, RSR2, DXR2, and XSR2 are not used (written, read, or shifted).

Receive data arrives on the DR pin and is shifted into receive shift register 1 (RSR1), upon reception of a frame synchronization signal. When a full word is received, the content of RSR1 is copied to the receive buffer register 1 (RBR1), if RBR1 is not full with previous data. RBR1 is then copied to the data receive register 1 (DRR1), unless the previous content of DRR1 has not been read by the DSP core, MPU core, or DSP/system DMA controller. If the companding feature of the McBSP is used, the required word length is 8 bits and receive data is expanded into the appropriate format before being passed from RBR1 to DRR1. For more details about reception, see section 2.2.5.

Transmit data is written by the DSP core, MPU core, or DSP/system DMA controller to the data transmit register 1 (DXR1). If there is no previous data the in transmit shift register (XSR1), the value in DXR1 is copied to XSR1; otherwise, DXR1 is copied to XSR1 when the last bit of the previous data is shifted out on the DX pin. If selected, the companding module compresses 16-bit data into the appropriate 8-bit format before passing it to XSR1. After transmit frame synchronization, the transmitter begins shifting bits from XSR1 to the DX pin. For more details about transmission, see section 2.2.6.

2.2.1.2 Data Transfer Process for Word Length of 20, 24, or 32 Bits

If the word length is larger than 16 bits, two 16-bit registers are needed at each stage of the data transfer paths. The registers DRR2, RBR2, RSR2, DXR2, and XSR2 are needed to hold the most significant bits.

Receive data arrives on the DR pin and is shifted first into RSR2 and then into RSR1, upon reception of a frame synchronization signal. When the full word is received, the contents of RSR2 and RSR1 are copied to RBR2 and RBR1, respectively, if RBR1 is not full. Then the contents of RBR2 and RBR1 are copied to DRR2 and DRR1, respectively, unless the previous content of DRR1 has not been read by the DSP core, MPU core, or DSP/system DMA controller. The DSP core, MPU core, or DSP/system DMA controller must read data from DRR2 first and then from DRR1. When DRR1 is read, the next RBR-to-DRR copy occurs. For more details about reception, see section 2.2.5.

For transmission, the DSP core, MPU core, or DSP/system DMA controller must write data to DXR2 first and then to DXR1. When new data arrives in DXR1, if there is no previous data in XSR1, the contents of DXR2 and DXR1 are copied to XSR2 and XSR1, respectively. Otherwise, the contents of the DXRs are copied to the XSRs when the last bit of the previous data is shifted out on the DX pin. After transmit frame synchronization, the transmitter begins shifting bits from the XSRs to the DX pin. For more details about transmission, see section 2.2.6.

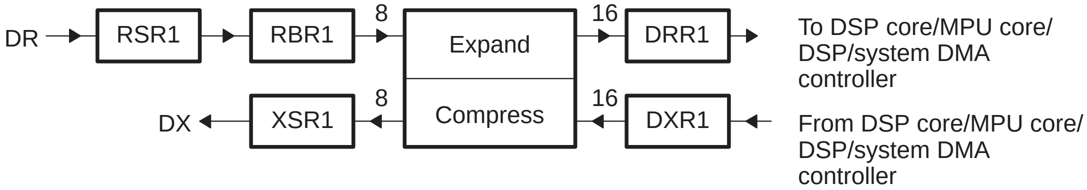
2.2.2 Companding (Compressing and Expanding) Data

Companding (COMpressing and exPANDing) hardware allows compression and expansion of data in either μ-law or A-law format. The companding standard employed in the United States and Japan is μ-law. The European companding standard is referred to as A-law. The specifications for μ-law and A-law log PCM are part of the CCITT G.711 recommendation.

A-law and μ-law allow 13 bits and 14 bits of dynamic range, respectively. Any values outside this range are set to the most positive or most negative value. Thus, for companding to work best, the data transferred to and from the McBSP via the DSP core, MPU core, or DSP/system DMA controller must be at least 16 bits wide. The μ-law and A-law formats both encode data into 8-bit code words. Companded data is always 8 bits wide; the appropriate word length bits (RWDLEN1, RWDLEN2, XWDLEN1, XWDLEN2) must therefore be cleared, indicating an 8-bit wide serial data stream. If companding is enabled and either of the frame phases does not have an 8-bit word length, companding continues as if the word length is 8 bits.

Figure 12 illustrates the companding processes. When companding is chosen for the transmitter, compression occurs during the process of copying data from DXR1 to XSR1. The transmit data is encoded according to the specified companding law (A-law or μ-law). When companding is chosen for the receiver, expansion occurs during the process of copying data from RBR1 to DRR1. The receive data is decoded to twos-complement format.

Figure 12. Companding Processes


DR
RSR1
RBR1
8
Expand
16
DRR1
To DSP core/MPU core/ DSP/system DMA controller
DX
XSR1
8
Compress
16
DXR1
From DSP core/MPU core/ DSP/system DMA controller

2.2.2.1 Companding Formats

For reception, the 8-bit compressed data in RBR1 is expanded to left-justified 16-bit data in DRR1. The receive sign-extension and justification mode specified in RJUST is ignored when companding is used.

For transmission using µ-law compression, the 14 data bits must be left-justified in DXR1 and the remaining two low-order bits must be filled with 0s, as shown in Figure 13.

Figure 13. µ-Law Transmit Data Companding Format

	15-2	1-0
µ-law format in DXR1	Value	00

For transmission using A-law compression, the 13 data bits must be left-justified in DXR1, with the remaining three low-order bits filled with 0s, as shown in Figure 14.

Figure 14. A-Law Transmit Data Companding Format

	15-3	2-0
A-law format in DXR1	Value	000

2.2.2.2 Reversing Bit Order: Option to Transfer LSB First

Generally, the McBSP transmits or receives all data with the most significant bit (MSB) first. However, certain 8-bit data protocols that do not use companded data require the least significant bit (LSB) to be transferred first. If you set XCOMPAND = 01b in XCR2, the bit ordering of 8-bit words is reversed (LSB first) before being sent from the serial port. If you set RCOMPAND = 01b in RCR2, the bit ordering of 8-bit words is reversed during reception. Similar to companding, this feature is enabled only if the appropriate word length bits are cleared, indicating that 8-bit words are to be transferred serially. If either phase of the frame does not have an 8-bit word length, the McBSP assumes the word length is eight bits, and LSB-first ordering is done.

2.2.3 Bit-Clock and Framing Data

This section explains basic concepts and terminology important for understanding how McBSP data transfers are timed and delimited.

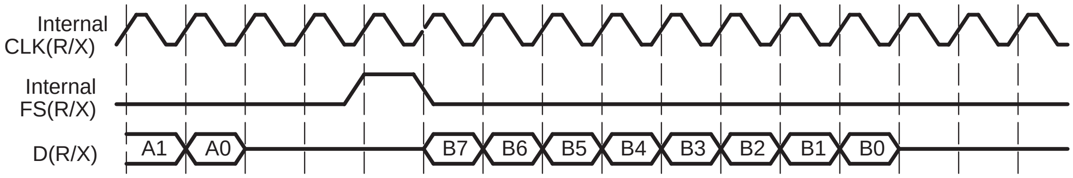
2.2.3.1 Bit-Clock

Data is shifted one bit at a time from the DR pin to the RSR(s) or from the XSR(s) to the DX pin. The time for each bit transfer is controlled by the rising or falling edge of a clock signal.

The receive clock signal (CLKR) controls bit transfers from the DR pin to the RSR(s). The transmit clock signal (CLKX) controls bit transfers from the XSR(s) to the DX pin. CLKR or CLKX can be derived from a pin at the boundary of the McBSP or derived from inside the McBSP. The polarities of CLKR and CLKX are programmable.

In the example in Figure 15, the clock signal controls the timing of each bit transfer on the pin.

Figure 15. Example of Clock Signal Control of Bit Transfer Timing



Note:

The McBSP module cannot operate at a frequency faster than 1/2 the functional clock frequency. When driving CLKX or CLKR at the pin, choose an appropriate input clock frequency. When using the internal sample rate generator for CLKX and/or CLKR, choose an appropriate input clock frequency and divide-down value (CLKDV). This is a restriction on the McBSP module itself. To sustain a given data rate on the McBSP, care must be taken with the interface clock frequency and other bus data traffic. For more details, see the usage cases presented in sections 3 through 6.

2.2.3.2 Serial Words

Bits traveling between a shift register (RSR or XSR) and a data pin (DR or DX) are transferred in a group called a serial word. You can define how many bits are in a word.

Bits coming in on the DR pin are held in RSR until RSR holds a full serial word. Only then is the word passed to RBR (and ultimately to the DRR).

During transmission, XSR does not accept new data from DXR until a full serial word has been passed from XSR to the DX pin.

Figure 15 defines an 8-bit word size (see bits 7 through 0 of word B being transferred).

2.2.3.3 Frames and Frame Synchronization

One or more words are transferred in a group called a frame. You can define how many words are in a frame.

All of the words in a frame are sent in a continuous stream. However, there can be pauses between frame transfers. The McBSP uses frame-synchronization signals to determine when each frame is received/transmitted. When a pulse occurs on a frame-synchronization signal, the McBSP begins receiving/transmitting a frame of data. When the next pulse occurs, the McBSP receives/transmits the next frame, and so on.

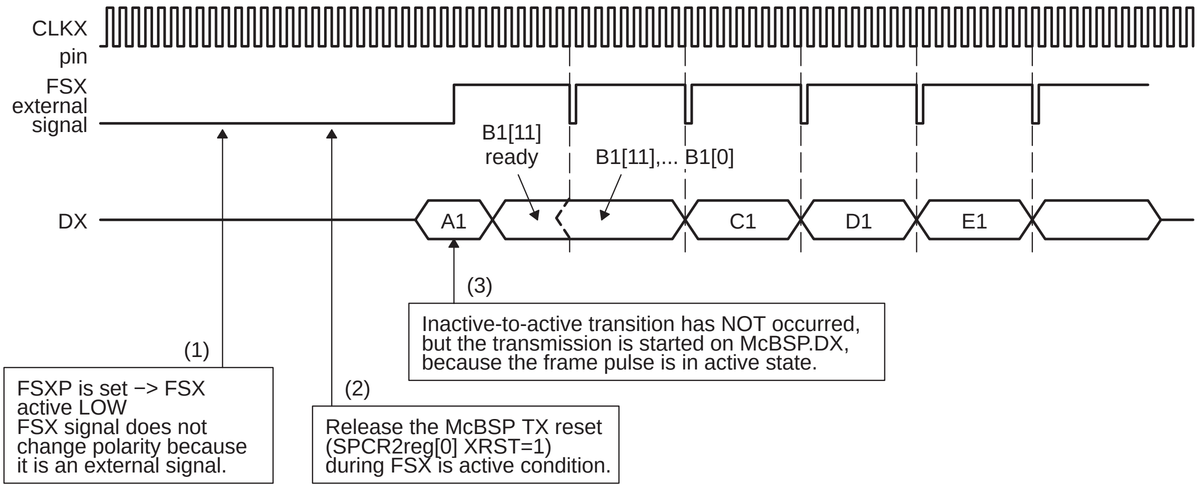
Pulses on the receive frame-synchronization (FSR) signal initiate frame transfers on DR. Pulses on the transmit frame-sync (FSX) signal initiate frame transfers on DX. FSR or FSX can be derived from a pin at the boundary of the McBSP or derived from inside the McBSP.

In Figure 15, a one-word frame is transferred when a frame-synchronization pulse occurs.

In McBSP operation, the inactive-to-active transition of the frame-synchronization signal indicates the start of the next frame. For this reason, the frame-synchronization signal may be high for an arbitrary number of clock cycles. The next frame synchronization occurs only after the signal is recognized to have gone inactive, and then active again.

An exception occurs when the transmitter and receiver are taken out of reset if the frame-synchronization signal is generated by an external device (the McBSP is a slave). Figure 16 shows the behavior.

Figure 16. Slave McBSP with FSXP=1 Data Transmission


CLKX pin
FSX external signal
DX
B1[11] ready
B1[11],... B1[0]
A1
C1
D1
E1
(1)
FSXP is set −> FSX active LOW
FSX signal does not change polarity because it is an external signal.
(2)
Release the McBSP TX reset (SPCR2reg[0] XRST=1) during FSX is active condition.
(3)
Inactive-to-active transition has NOT occurred, but the transmission is started on McBSP.DX, because the frame pulse is in active state.

Note that this behavior also holds true for the receiver and the FSR signal when the McBSP is the slave device. Therefore, data misalignment can occur if either the receiver or transmitter are released from reset while the FSR/FSX signals are in the active state (i.e. FSR/FSX are low and FSRP/FSXP=1). To resolve this problem, the receiver or transmitter must be released during the inactive state after detecting the first frame synchronization signal edge through a DSP core or MPU core interrupt (R/XINT=10b).

2.2.3.4 Detecting Frame-Synchronization Pulses, Even in Reset State

The McBSP can send receive and transmit interrupts to the DSP core or MPU core to indicate specific events in the McBSP. To facilitate detection of frame synchronization, an interrupt can be generated in response to frame-synchronization pulses. Set the appropriate interrupt mode bits to 10b (for reception, RINTM = 10b; for transmission, XINTM = 10b).

Unlike other serial port interrupt modes, this mode can operate while the associated portion of the serial port is in reset, such as activating the RX interrupt (internal McBSP signal RINT) when the receiver is in reset. In this case, FSRM/FSXM and FSRP/FSXP still select the appropriate source and polarity of frame synchronization. Thus, even when the serial port is in the reset state, these signals are synchronized to the internal clock source and then sent to the DSP core or MPU core in the form of RX and TX interrupts. Consequently, a new frame-synchronization pulse can be detected, and after this occurs, the DSP core or MPU core can safely take the serial port out of reset.

In slave mode, interrupts will always be generated on the rising edge of the FSX/FSR signal, in or out of reset. In this example, the value of the FSXP/FSRP bit does not have an effect as an external device generates the frame synchronization signal. Therefore, the FSXP/FSRP bit only defines the active state for data transmission.

2.2.3.5 Ignoring Frame-Synchronization Pulses

The McBSP can be configured to ignore unexpected transmit and/or receive frame-synchronization pulses. To have the receiver or transmitter recognize frame-synchronization pulses, clear the appropriate frame-synchronization ignore bit (RFIG = 0 for the receiver, XFIG = 0 for the transmitter). To have the receiver or transmitter ignore unexpected frame-synchronization pulses, set the appropriate frame-synchronization ignore bit (RFIG = 1 for the receiver, XFIG = 1 for the transmitter). For more details on unexpected frame-synchronization pulses, see section 2.7.2 or 2.7.5.

You can also use the frame-synchronization ignore function for data packing (for more details, see section 2.15).

2.2.3.6 Frame Frequency

The frame frequency is determined by the period between frame-synchronization pulses and is defined by Equation 3.

Equation 3. McBSP Frame Frequency

$$\text{Frame Frequency} = \frac{\text{Bit Clock Frequency}}{\text{Number of Clock Cycles Between Frame−Sync Pulses}}$$

The frame frequency can be increased by decreasing the time between frame-synchronization pulses (limited only by the number of bits per frame). As the frame transmit frequency increases, the inactivity period between the data packets for adjacent transfers decreases to zero.

2.2.3.7 Maximum Frame Frequency

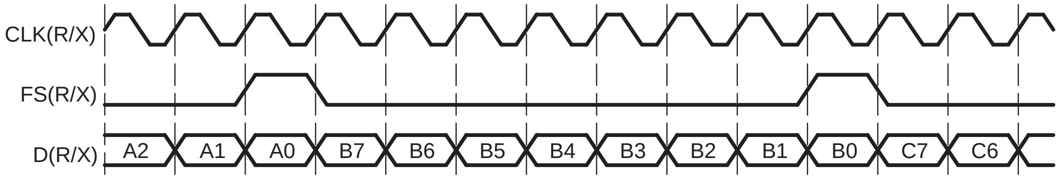
The minimum number of clock cycles between frame synchronization pulses is equal to the number of bits transferred per frame. The maximum frame frequency is defined by Equation 4.

Equation 4. McBSP Maximum Frame Frequency

$$\text{Maximum Frame Frequency} = \frac{\text{Bit Clock Frequency}}{\text{Number of Bits Per Frame}}$$

Figure 17 shows the McBSP operating at maximum packet frequency. At maximum packet frequency, the data bits in consecutive packets are transmitted contiguously with no inactivity between bits.

Figure 17. McBSP Operating at Maximum Packet Frequency



If there is a 1-bit data delay, the frame-synchronization pulse overlaps the last bit transmitted in the previous frame (Figure 17). Effectively, this permits a continuous stream of data, making frame-synchronization pulses redundant.

Note:

For XDATDLY = 0 (0-bit data delay), the first bit of data is transmitted asynchronously to the internal transmit clock signal (CLKX). For more details, see section 2.6.12.

2.2.4 Frame Phases

The McBSP allows you to configure each frame to contain one or two phases. The number of words per frame and the number of bits per word can be specified differently for each of the two phases of a frame, allowing greater flexibility in structuring data transfers. For example, a frame might be defined as consisting of one phase containing two words of 16 bits each, followed by a second phase consisting of 10 words of 8 bits each. This configuration permits the composition of frames for custom applications or, in general, maximizes the efficiency of data transfers.

2.2.4.1 Number of Phases, Words, and Bits Per Frame

Table 6 shows which bit-fields in the receive control registers (RCR1 and RCR2) and in the transmit control registers (XCR1 and XCR2) determine the number of phases per frame, the number of words per frame, and number of bits per word for each phase, for the receiver and transmitter. The maximum number of words per frame is 128 for a single-phase frame and 256 for a dual-phase frame. The number of bits per word can be 8, 12, 16, 20, 24, or 32 bits.

Table 6. McBSP Register Bits

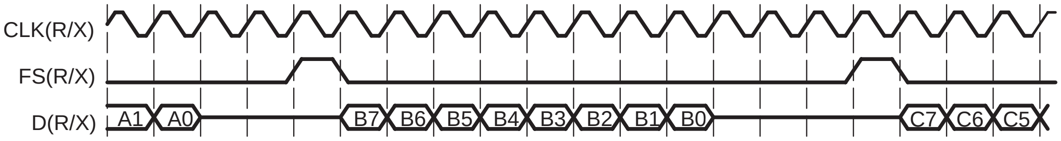
Operation	Number of Phases	Words per Frame Set With:	Bits per Word Set With:
Reception	1 (RPHASE = 0)	RFRLEN1	RWDLEN1
Reception	2 (RPHASE = 1)	RFRLEN1 and RFRLEN2	RWDLEN1 for phase 1 RWDLEN2 for phase 2
Transmission	1 (XPHASE = 0)	XFRLEN1	XWDLEN1
Transmission	2 (XPHASE = 1)	XFRLEN1 and XFRLEN2	XWDLEN1 for phase 1 XWDLEN2 for phase 2

2.2.4.2 Single-Phase Frame Example

Figure 18 shows an example of a single-phase data frame containing one 8-bit word. Because the transfer is configured for one data bit delay, the data on the DX and DR pins are available one clock cycle after FS(R/X) goes active. The figure makes the following assumptions:

- ❑ (R/X)PHASE = 0: Single-phase frame
- ❑ (R/X)FRLEN1 = 0b: 1 word per frame
- ❑ (R/X)WDLEN1 = 000b: 8-bit word length
- ❑ (R/X)FRLEN2 and (R/X)WDLEN2 are ignored
- ❑ CLK(X/R)P = 0: Receive data clocked on falling edge; transmit data clocked on rising edge
- ❑ FS(R/X)P = 0: Active-high frame-synchronization signals
- ❑ (R/X)DATDLY = 01b: 1-bit data delay

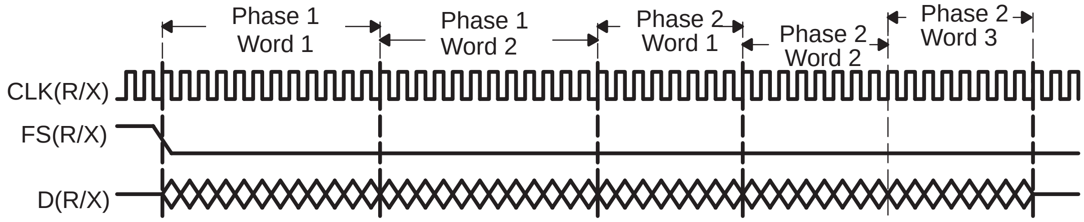
Figure 18. Single-Phase Frame for a McBSP Data Transfer



2.2.4.3 Dual-Phase Frame Example

Figure 19 shows an example of a frame where the first phase consists of two words of 12 bits each, followed by a second phase of three words of 8 bits each. The entire bit stream in the frame is contiguous. There are no gaps either between words or between phases.

Figure 19. Dual-Phase Frame for a McBSP Data Transfer

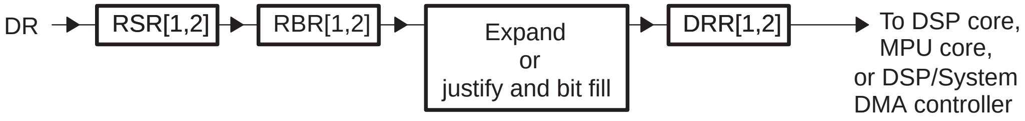


2.2.5 McBSP Reception

This section explains the fundamental process of reception in the McBSP. For details about how to program the McBSP receiver, see section 2.5.

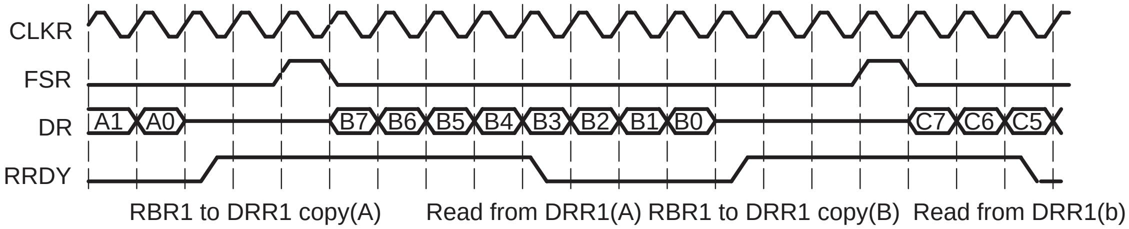
Figure 20 and Figure 21 show McBSP reception. Figure 20 shows the physical path for the data. Figure 21 is a timing diagram showing signal activity for one possible reception scenario. A description of the process follows the figures.

Figure 20. McBSP Reception Physical Data Path



RSR[1,2]: Receive shift registers 1 and 2
RBR[1,2]: Receive buffer registers 1 and 2
DRR[1,2]: Data receive registers 1 and 2

Figure 21. McBSP Reception Signal Activity



CLKR: Internal receive clock
FSR: Internal receive frame-synchronization signal
DR: Data on DR pin
RRDY: Status of receiver ready bit (high is 1)

The following process describes how data travels from the DR pin to the MPU core, DSP core, or DSP/system DMA controller:

1) The McBSP waits for a receive frame-synchronization pulse on internal FSR.

2) When the pulse arrives, the McBSP inserts the appropriate data delay that is selected with the RDATDLY bits in RCR2.

   In the preceding timing diagram (Figure 21), a 1-bit data delay is selected.

3) The McBSP accepts data bits on the DR pin and shifts them into the receive shift register(s).

   If the word length is 16 bits or smaller, only RSR1 is used. If the word length is larger than 16 bits, RSR2 and RSR1 are used and RSR2 contains the most significant bits. For details on choosing a word length, see section 2.5.8.

4) When a full word is received, the McBSP copies the contents of the receive shift register(s) to the receive buffer register(s), provided that RBR1 is not full with previous data.

   If the word length is 16 bits or smaller, only RBR1 is used. If the word length is larger than 16 bits, RBR2 and RBR1 are used and RBR2 contains the most significant bits.

5) The McBSP copies the contents of the receive buffer register(s) into the data receive register(s), provided that DRR1 is not full with previous data. When DRR1 receives new data, the receiver ready bit (RRDY) is set in SPCR1. This indicates that receive data is ready to be read by the DSP core, MPU core, or DSP/system DMA controller.

   If the word length is 16 bits or smaller, only DRR1 is used. If the word length is larger than 16 bits, DRR2 and DRR1 are used and DRR2 contains the most significant bits.

   If companding is used during the copy (RCOMPAND = 10b or 11b in RCR2), the 8-bit compressed data in RBR1 is expanded to a left-justified 16-bit value in DRR1. If companding is disabled, the data copied from RBR[1,2] to DRR[1,2] is justified and bit filled according to the RJUST bits.

6) The DSP core, MPU core, or DSP/system DMA controller reads the data from the data receive register(s). When DRR1 is read, RRDY is cleared and the next RBR-to-DRR copy is initiated.

Note:

If both DRRs are required (word length larger than 16 bits), the DSP core, MPU core, or DSP/system DMA controller must read from DRR2 first and then from DRR1. As soon as DRR1 is read, the next RBR-to-DRR copy is initiated. If DRR2 is not read first, the data in DRR2 is lost.

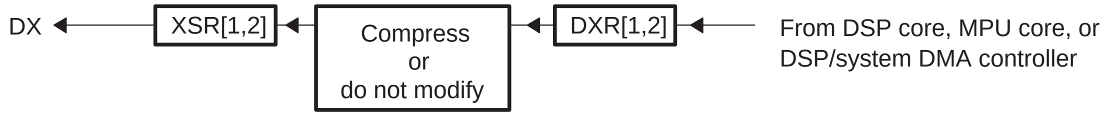
When activity is not properly timed, errors can occur. See section 2.7.1 or 2.7.2 for more details.

2.2.6 McBSP Transmission

This section explains the fundamental process of transmission in the McBSP. For details about how to program the McBSP transmitter, see section 2.6.

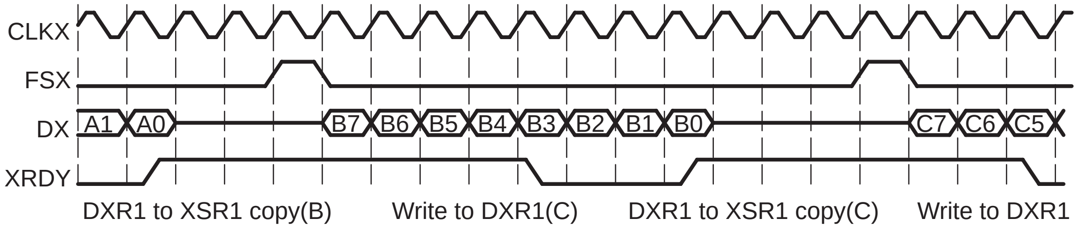
Figure 22 and Figure 23 show how transmission occurs in the McBSP. Figure 22 shows the physical path for the data. Figure 23 is a timing diagram showing signal activity for one possible transmission scenario. A description of the process follows the figures.

Figure 22. McBSP Transmission Physical Data Path



XSR[1,2]: Transmit shift registers 1 and 2   DXR[1,2]: Data transmit registers 1 and 2

Figure 23. McBSP Transmission Signal Activity



CLKX: Internal transmit clock
FSX: Internal transmit frame-synchronization signal
DX: Data on DX pin
XRDY: Status of transmitter ready bit (high is 1)

1) The DSP core, MPU core, or DSP/system DMA controller writes data to the data transmit register(s). When DXR1 is loaded, the transmitter ready bit (XRDY) is cleared in SPCR2 to indicate that the transmitter is not ready for new data.

   If the word length is 16 bits or smaller, only DXR1 is used. If the word length is larger than 16 bits, DXR2 and DXR1 are used and DXR2 contains the most significant bits. For details on choosing a word length, see section 2.6.8.

Note:

If both DXRs are needed (word length larger than 16 bits), the MPU core, DSP core, or DSP/system DMA controller must load DXR2 first and then load DXR1. As soon as DXR1 is loaded, the contents of both DXRs are copied to the transmit shift registers (XSRs), as described in the next step. If DXR2 is not loaded first, the previous content of DXR2 is passed to the XSR2.

2) When new data arrives in DXR1, the McBSP copies the content of the data transmit register(s) to the transmit shift register(s). In addition, the transmit ready bit (XRDY) is set. This indicates that the transmitter is ready to accept new data from the MPU core, DSP core, or DSP/system DMA controller.

   If the word length is 16 bits or smaller, only XSR1 is used. If the word length is larger than 16 bits, XSR2 and XSR1 are used and XSR2 contains the most significant bits.

   If companding is used during the transfer (XCOMPAND = 10b or 11b in XCR2), the McBSP compresses the 16-bit data in DXR1 to 8-bit data in the μ-law or A-law format in XSR1. If companding is disabled, the McBSP passes data from the DXR(s) to the XSR(s) without modification.

3) The McBSP waits for a transmit frame-synchronization pulse on internal FSX.

4) When the pulse arrives, the McBSP inserts the appropriate data delay that is selected with the XDATDLY bits in XCR2.

   In the preceding timing diagram (Figure 23), a 1-bit data delay is selected.

5) The McBSP shifts data bits from the transmit shift register(s) to the DX pin.

When activity is not properly timed, errors can occur. See section 2.7.3, or 2.7.4, or 2.7.5 for more details.

2.3 Signal Descriptions: McBSP Pins

Table 7 describes the McBSP interface pins. For information on using these pins for general-purpose input/output (GPIO), see section 2.9.

Note:

Note that for simplicity, the abbreviated name convention in Table 7 is used for the McBSP pin names in this document. For example, when referring to pin DX of a given McBSP, it will represent pins McBSP1.DX, McBSP2.DX, and McBSP3.DX.

Table 7. McBSP Interface Pins[†]

Pin	McBSP1	McBSP2	McBSP3	Direction	Possible Uses
CLKR		MCBSP2.CLKR		I/O	Supplying or reflecting the receive clock; supplying the input clock of the sample rate generator; general-purpose I/O.
CLKX	MCBSP1.CLKX	MCBSP2.CLKX	MCBSP3.CLKX	I/O	Supplying or reflecting the transmit clock; supplying the input clock of the sample rate generator; general-purpose I/O.
CLKS	MCBSP1.CLKS			I	Supplying the input clock of the sample rate generator; general-purpose input.
DR	MCBSP1.DR	MCBSP2.DR	MCBSP3.DR	I	Receiving serial data; general-purpose input.
DX	MCBSP1.DX	MCBSP2.DX	MCBSP3.DX	O	Transmitting serial data; general-purpose output. DX always drives this pin.
DXZ	MCBSP1.DXZ	MCBSP2.DXZ	MCBSP3.DXZ	O/Z	Transmitting serial data on multichannel mode.

[†] See sections 1.3.1, 1.3.2, and 1.3.3 for the internal signal routing for non-available pins.

Table 7. McBSP Interface Pins[†] (Continued)

Pin	McBSP1	McBSP2	McBSP3	Direction	Possible Uses
FSR		MCBSP2.FSR		I/O	Supplying or reflecting the receive frame-sync signal; controlling sample rate generator synchronization when GSYNC = 1 (see section 2.1.3); general-purpose I/O.
FSX	MCBSP1.FSX	MCBSP2.FSX	MCBSP3.FSX	I/O	Supplying or reflecting the transmit frame-sync signal; general-purpose I/O.

[†] See sections 1.3.1, 1.3.2, and 1.3.3 for the internal signal routing for non-available pins.

2.4 Protocol Description: SPI Operation Using the Clock Stop Mode

This section describes the SPI protocol and the compatibility considerations with the McBSP.

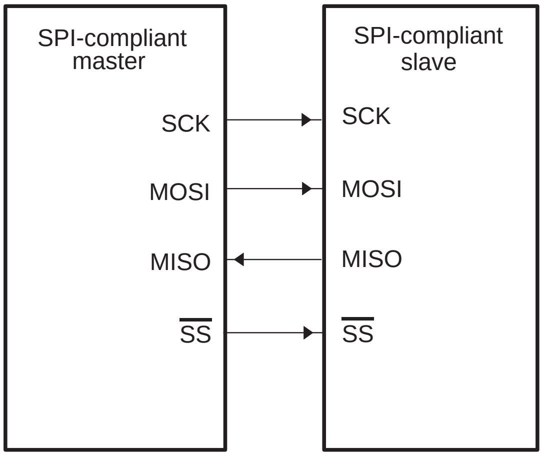
2.4.1 SPI Protocol

The SPI protocol is a master-slave configuration with one master device and one or more slave devices. The interface consists of the following four signals:

- Serial data input (also referred to as master in/slave out, or MISO)
- Serial data output (also referred to as master out/slave in, or MOSI)
- Shift-clock (also referred to as SCK)
- Slave-enable signal (also referred to as SS)

A typical SPI interface with a single slave device is shown in Figure 24.

Figure 24. Typical SPI Interface



The master device controls the flow of communication by providing shift-clock and slave-enable signals. The slave-enable signal is an optional active-low signal that enables the serial data input and output of the slave device (i.e., the device does not send out the clock).

In the absence of a dedicated slave-enable signal on the McBSP, communication between the master and slave is determined by the presence or absence of an active shift-clock. When the McBSP is operating in SPI master mode and the $\overline{SS}$ signal is not used by the slave SPI port, the slave device must remain enabled at all times, and multiple slaves cannot be used.

2.4.2 Clock Stop Mode

The clock stop mode of the McBSP provides compatibility with the SPI protocol. When the McBSP is configured in clock stop mode, the transmitter and receiver are internally synchronized so that the McBSP functions as an SPI master or slave device. The transmit clock signal (CLKX) corresponds to the serial clock signal (SCK) of the SPI protocol, while the transmit frame-synchronization signal (FSX) is used as the slave-enable signal ($\overline{SS}$).

The receive clock signal (CLKR) and receive frame-synchronization signal (FSR) are not used in the clock stop mode because these signals are internally connected to their transmit counterparts, CLKX and FSX.

2.4.3 Bits Used to Enable and Configure the Clock Stop Mode

The bits required to configure the McBSP as an SPI device are introduced in Table 8. Table 9 shows how the various combinations of the CLKSTP bit and the polarity bits CLKXP and CLKRP create four possible clock stop mode configurations. The timing diagrams in section 2.4.4 show the effects of CLKSTP, CLKXP, and CLKRP.

Table 8. Bits Used to Enable and Configure the Clock Stop Mode

Bit Field	Description
CLKSTP bits in SPCR1	These bits enable the clock stop mode and select one of two timing variations (see also Table 9).
CLKXP bit in PCR	This bit determines the polarity of the CLKX signal (see also Table 9).
CLKRP bit in PCR	This bit determines the polarity of the CLKR signal (see also Table 9).
CLKXM bit in PCR	This bit determines whether CLKX is an input signal (McBSP as slave) or an output signal (McBSP as master).
XPHASE bit in XCR2	You must use a single-phase transmit frame (XPHASE = 0).
RPHASE bit in RCR2	You must use a single-phase receive frame (RPHASE = 0).
XFRLEN1 bits in XCR1	You must use a transmit frame length of 1 serial word (XFRLEN1 = 0).
RFRLEN1 bits in RCR1	You must use a receive frame length of 1 serial word (RFRLEN1 = 0).
XWDLEN1 bits in XCR1	The XWDLEN1 bits determine the transmit packet length. XWDLEN1 must be equal to RWDLEN1 because the McBSP transmit and receive circuits are synchronized to a single clock in the clock stop mode.
RWDLEN1 bits in RCR1	The RWDLEN1 bits determine the receive packet length. RWDLEN1 must be equal to XWDLEN1 because the McBSP transmit and receive circuits are synchronized to a single clock in the clock stop mode..

Table 9. Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme

Bit Settings	Clock Scheme

CLKSTP = 00b or 01b
CLKXP = 0 or 1
CLKRP = 0 or 1
	Clock stop mode disabled. Clock enabled for non-SPI mode.

CLKSTP = 10b
CLKXP = 0
CLKRP = 0
	Low inactive state without delay: The McBSP transmits data on the rising edge of CLKX and receives data on the falling edge of CLKX.

CLKSTP = 11b
CLKXP = 0
CLKRP = 1
	Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of CLKX and receives data on the rising edge of CLKX.

CLKSTP = 10b
CLKXP = 1
CLKRP = 0
	High inactive state without delay: The McBSP transmits data on the falling edge of CLKX and receives data on the rising edge of CLKX.

CLKSTP = 11b
CLKXP = 1
CLKRP = 1
	High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of CLKX and receives data on the falling edge of CLKX.

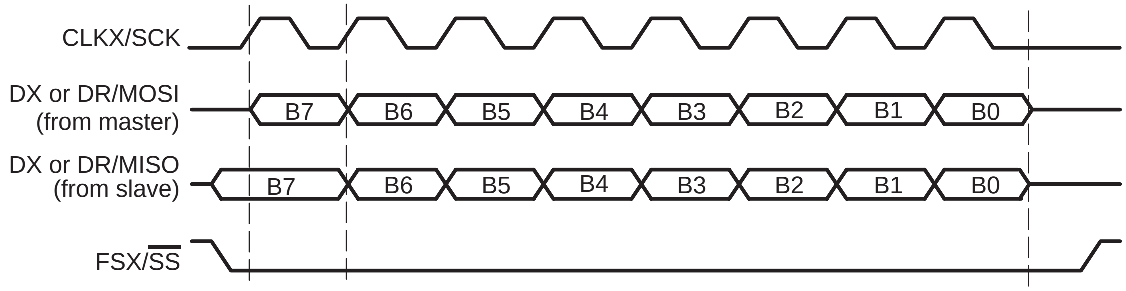
2.4.4 Clock Stop Mode Timing Diagrams

The timing diagrams for the four possible clock stop mode configurations are shown here. Notice that the frame-synchronization signal used in clock stop mode is active throughout the entire transmission as a slave-enable signal. Although the timing diagrams show 8-bit transfers, the packet length can be set to 8, 12, 16, 20, 24, or 32 bits per packet. The receive packet length is selected with the RWDLEN1 bits in RCR1, and the transmit packet length is selected with the XWDLEN1 bits in XCR1. For clock stop mode, the values of RWDLEN1 and XWDLEN1 must be the same because the McBSP transmit and receive circuits are synchronized to a single clock.

Note:

Even if multiple words are consecutively transferred, the CLKX signal is always stopped and the FSX signal returns to the inactive state after a packet transfer. When consecutive packet transfers are performed, this leads to a minimum idle time of two bit-periods between each packet transfer.

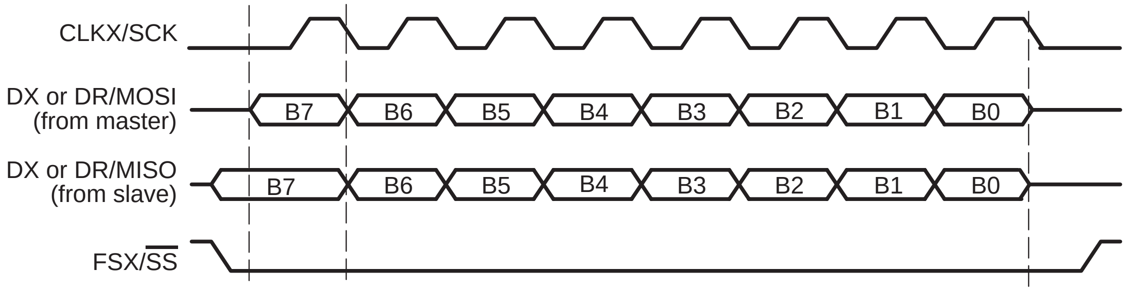
Figure 25. SPI Transfer With CLKSTP = 10b (No Clock Delay), CLKXP = 0, and CLKRP = 0



Notes: 1) If the McBSP is the SPI master (CLKXM = 1), MOSI = DX. If the McBSP is the SPI slave (CLKXM = 0), MOSI = DR.

2) If the McBSP is the SPI master (CLKXM = 1), MISO = DR. If the McBSP is the SPI slave (CLKXM = 0), MISO = DX.

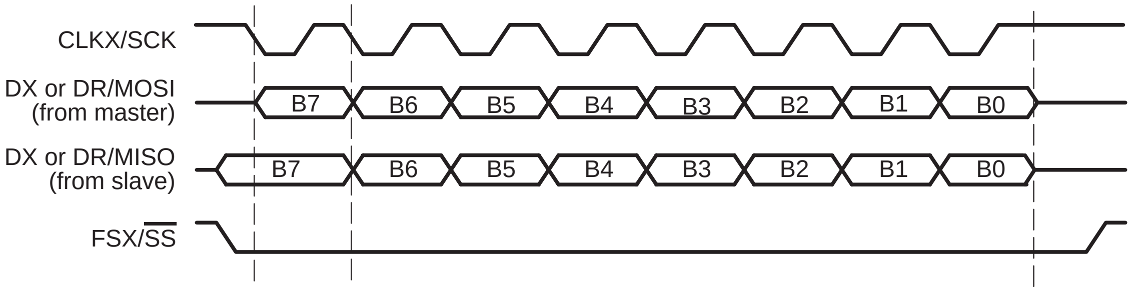
Figure 26. SPI Transfer With CLKSTP = 11b (Clock Delay), CLKXP = 0, CLKRP = 1



Notes: 1) If the McBSP is the SPI master (CLKXM = 1), MOSI = DX. If the McBSP is the SPI slave (CLKXM = 0), MOSI = DR.

2) If the McBSP is the SPI master (CLKXM = 1), MISO = DR. If the McBSP is the SPI slave (CLKXM = 0), MISO = DX.

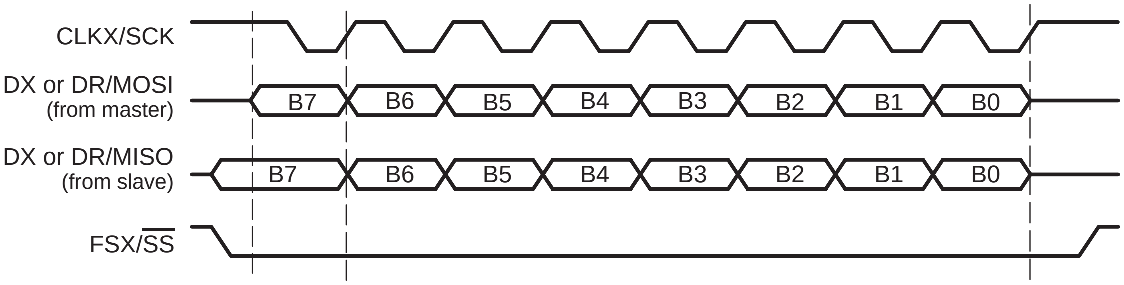
Figure 27. SPI Transfer With CLKSTP = 10b (No Clock Delay), CLKXP = 1, and CLKRP = 0



Notes: 1) If the McBSP is the SPI master (CLKXM = 1), MOSI = DX. If the McBSP is the SPI slave (CLKXM = 0), MOSI = DR.

2) If the McBSP is the SPI master (CLKXM = 1), MISO = DR. If the McBSP is the SPI slave (CLKXM = 0), MISO = DX.

Figure 28. SPI Transfer With CLKSTP = 11b (Clock Delay), CLKXP = 1, and CLKRP = 1



Notes: 1) If the McBSP is the SPI master (CLKXM = 1), MOSI=DX. If the McBSP is the SPI slave (CLKXM = 0), MOSI = DR.

2) If the McBSP is the SPI master (CLKXM = 1), MISO=DR. If the McBSP is the SPI slave (CLKXM = 0), MISO = DX.

2.4.5 Procedure for Configuring a McBSP for SPI Operation

To configure the McBSP for SPI master or slave operation:

Step 1: Place the transmitter and receiver in reset.

Clear the transmitter reset bit ($\overline{\text{XRST}}$ = 0) in SPCR2 to reset the transmitter. Clear the receiver reset bit ($\overline{\text{RRST}}$ = 0) in SPCR1 to reset the receiver.

Step 2: Place the sample rate generator in reset.

Clear the sample rate generator reset bit ($\overline{\text{GRST}}$ = 0) in SPCR2 to reset the sample rate generator.

Step 3: Program registers that affect SPI operation.

Program the appropriate McBSP registers to configure the McBSP for proper operation as an SPI master or an SPI slave. See section 2.4.6 or 2.4.7 for a list of important bits settings for McBSP as a master or slave, respectively.

Step 4: Enable the sample rate generator.

To release the sample rate generator from reset, set the sample rate generator reset bit ($\overline{\text{GRST}}$ = 1) in SPCR2.

Ensure that during the write to SPCR2, you only modify $\overline{\text{GRST}}$. Otherwise, you modify the McBSP configuration you selected in the previous step.

Step 5: Enable the transmitter and receiver.

After the sample rate generator is released from reset, wait two sample rate generator clock periods for the McBSP logic to stabilize.

If the DSP core or the MPU core services the McBSP transmit and receive buffers, then you can immediately enable the transmitter ($\overline{\text{XRST}}$ = 1 in SPCR2) and enable the receiver ($\overline{\text{RRST}}$ = 1 in SPCR1).

If the DSP/system DMA controller services the McBSP transmit and receive buffers, then you must first configure the DSP/system DMA controller. This includes enabling the channels that service the McBSP buffers. When the DSP/system DMA controller is ready, set $\overline{\text{XRST}}$ = 1 and $\overline{\text{RRST}}$ = 1.

In either case, make sure you only change $\overline{\text{XRST}}$ and $\overline{\text{RRST}}$ when you write to SPCR2 and SPCR1. Otherwise, you modify the bit settings you selected earlier in this procedure.

After the transmitter and receiver are released from reset, wait two sample rate generator clock periods for the McBSP logic to stabilize.

Step 6: If necessary, enable the frame-synchronization logic of the sample rate generator.

After the required data acquisition setup is done (DXR[1/2] is loaded with data), set $\overline{\text{FRST}}$ = 1 if an internally generated frame-synchronization pulse is required, that is, if the McBSP is the SPI master.

2.4.6 McBSP as the SPI Master

A SPI interface with the McBSP used as the master is shown in Figure 29. When the McBSP is configured as a master, the transmit output signal (DX) is used as the MOSI signal of the SPI protocol and the receive input signal (DR) is used as the MISO signal.

Figure 29. McBSP as SPI Master

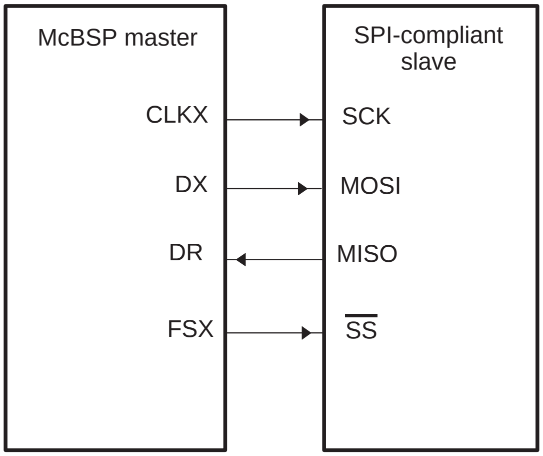


Table 10 lists the register bit values required to configure the McBSP as a master, followed by more details about the configuration requirements.

Table 10. Bit Values Required to Configure the McBSP as an SPI Master

Required Bit Setting	Description
CLKSTP = 10b or 11b	The clock stop mode (without or with a clock delay) is selected.
CLKXP = 0 or 1	The polarity of CLKX as seen on the CLKX pin is positive (CLKXP = 0) or negative (CLKXP = 1).
CLKRP = 0 or 1	The polarity of CLKR as seen on the CLKR pin is positive (CLKRP = 0) or negative (CLKRP = 1).
CLKXM = 1	The CLKX pin is an output pin driven by the internal sample rate generator. Because CLKSTP is equal to 10b or 11b, CLKR is driven internally by CLKX.

SCLKME = 0
CLKSM = 1
	The clock generated by the sample rate generator (CLKG) is derived from the internal clock source.

CLKGDV is a value from 0 to 255	CLKGDV defines the divide-down value for CLKG.
FSXM = 1	The FSX pin is an output pin driven according to the FSGM bit.
FSGM = 0	The transmitter drives a frame-synchronization pulse on the FSX pin every time data is transferred from DXR1 to XSR1.
FSXP = 1	The FSX pin is active low.

XDATDLY = 01b
RDATDLY = 01b
	This setting provides the correct setup time on the FSX signal.

When the McBSP functions as the SPI master, it controls the transmission of data by producing the serial clock signal. The clock signal on the CLKX pin is enabled only during packet transfers. When packets are not being transferred, the CLKX pin remains high or low depending on the polarity used.

For SPI master operation, the CLKX pin must be configured as an output. The sample rate generator is then used to derive the CLKX signal from the internal clock source. The clock stop mode internally connects the CLKX pin to the CLKR signal so that no external signal connection is required on the CLKR pin and both the transmit and receive circuits are clocked by the master clock (CLKX).

The data delay parameters of the McBSP (XDATDLY and RDATDLY) must be set for proper SPI master operation. A data delay value of 0 or 2 is undefined in the clock stop mode.

The McBSP can also provide a slave-enable signal ($\overline{\text{SS}}$) on the FSX pin. If a slave-enable signal is required, the FSX pin must be configured as an output and the transmitter must be configured so that a frame-synchronization pulse is generated automatically each time a packet is transmitted (FSGM = 0). The polarity of the FSX pin is programmable high or low; however, in most cases the pin must be configured active low.

When the McBSP is configured for SPI-master operation, the bit fields for frame-synchronization pulse width (FWID) and frame-synchronization period (FPER) are overridden, and custom frame-synchronization waveforms are not allowed. To view the resulting waveform produced on the FSX pin, see the timing diagrams in section 2.4.4. The signal becomes active before the first bit of a packet transfer, and remains active until the transfer of the last bit of the packet. After the packet transfer is complete, the FSX signal returns to the inactive state.

2.4.7 McBSP as an SPI Slave

Figure 30 shows a SPI interface with the McBSP used as a slave. When the McBSP is configured as a slave, DX is used as the MISO signal and DR is used as the MOSI signal.

Figure 30. SPI Interface With McBSP as Slave

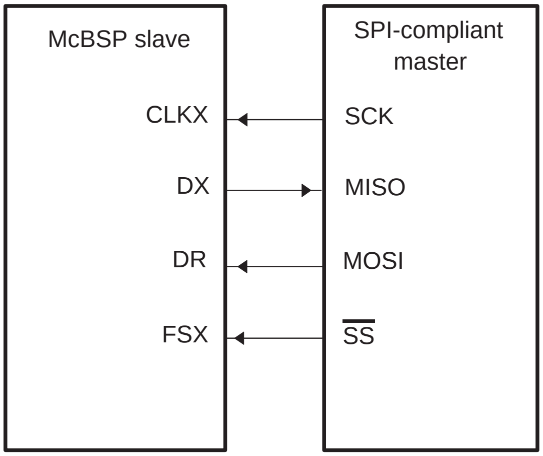


Table 11 lists the register bit values required to configure the McBSP as a slave, followed by more details about the configuration requirements.

Table 11. Bit Values Required to Configure the McBSP as an SPI Slave

Required Bit Setting	Description
CLKSTP = 10b or 11b	The clock stop mode (without or with a clock delay) is selected.
CLKXP = 0 or 1	The polarity of CLKX as seen on the CLKX pin is positive (CLKXP = 0) or negative (CLKXP = 1).
CLKRP = 0 or 1	The polarity of CLKR as seen on the CLKR pin is positive (CLKRP = 0) or negative (CLKRP = 1).
CLKXM = 0	The CLKX pin is an input pin, so that it can be driven by the SPI master. Because CLKSTP = 10b or 11b, CLKR is driven internally by CLKX.

SCLKME = 0
CLKSM = 1
	The clock generated by the sample rate generator (CLKG) is derived from the internal clock source. (The sample rate generator is used to synchronize the McBSP logic with the externally-generated master clock.)

CLKGDV = 1	The sample rate generator divides the internal clock source by 2 before generating CLKG.
FSXM = 0	The FSX pin is an input pin, so that it can be driven by the SPI master.
FSXP = 1	The FSX pin is active low.

XDATDLY = 00b
RDATDLY = 00b
	These bits must be 0s for SPI slave operation.

When the McBSP is used as an SPI slave, the master clock and slave-enable signals are generated externally by a master device. Accordingly, the CLKX and FSX pins must be configured as inputs. The CLKX pin is internally connected to the CLKR signal, so that both the transmit and receive circuits of the McBSP are clocked by the external master clock. The FSX pin is also internally connected to the FSR signal, and no external signal connections are required on the CLKR and FSR pins.

Although the CLKX signal is generated externally by the master and is asynchronous to the McBSP, the sample rate generator of the McBSP must be enabled for proper SPI slave operation. The sample rate generator must be programmed to its maximum rate of half the internal clock source rate. The CLKG rate should be at least eight times that of the SPI data rate. The internal sample rate clock is then used to synchronize the McBSP logic to the external master clock and slave-enable signals.

The McBSP requires an active edge of the slave-enable signal on the FSX input for each transfer. This means that the master device must assert the slave-enable signal at the beginning of each transfer, and de-assert the signal after the completion of each packet transfer; the slave-enable signal cannot remain active between transfers.

The data delay parameters of the McBSP must be cleared for proper SPI slave operation. A value of 1 or 2 is undefined in the clock stop mode.

2.5 Receiver Configuration

To configure the McBSP receiver, perform the following procedure:

1) Place the McBSP/receiver in reset (see section 2.5.2).

2) Program the McBSP registers for the desired receiver operation (see section 2.5.1).

3) Take the receiver out of reset (see section 2.5.2).

2.5.1 Programming the McBSP Registers for the Desired Receiver Operation

The following is a list of important tasks to be performed when you are configuring the McBSP receiver. Each task corresponds to one or more McBSP register bit fields.

- ❑ Global behavior:
  - Set the receiver pins to operate as McBSP pins
  - Enable/disable the digital loopback mode
  - Enable/disable the clock stop mode
  - Enable/disable the receive multichannel selection mode
- ❑ Data behavior:
  - Choose 1 or 2 phases for the receive frame
  - Set the receive word length(s)
  - Set the receive frame length
  - Enable/disable the receive frame-synchronization ignore function
  - Set the receive companding mode
  - Set the receive data delay
  - Set the receive sign-extension and justification mode
  - Set the receive interrupt mode
- ❑ Frame-synchronization behavior:
  - Set the receive frame-synchronization mode
  - Set the receive frame-synchronization polarity
  - Set the sample rate generator (SRG) frame-synchronization period and pulse width
- ❑ Clock behavior:
  - Set the receive clock mode
  - Set the receive clock polarity
  - Set the SRG clock divide-down value
  - Set the SRG clock synchronization mode
  - Set the SRG clock mode (choose an input clock)
  - Set the SRG input clock polarity

2.5.2 Resetting and Enabling the Receiver

The first step of the receiver configuration procedure is to reset the receiver, and the last step is to enable the receiver to take it out of reset. Table 12 describes the bits used for both of these steps.

Table 12. Register Bits Used to Reset or Enable the McBSP Receiver

Register	Bit	Name	Function	Type	Reset
SPCR1	0	$\overline{\text{RRST}}$	Receiver reset $\overline{\text{RRST}}$ = 0: The serial port receiver is disabled and in the reset state. $\overline{\text{RRST}}$ = 1: The serial port receiver is enabled.	R/W	0
SPCR2	6	$\overline{\text{GRST}}$	Sample rate generator reset $\overline{\text{GRST}}$ = 0: Sample rate generator is reset. If $\overline{\text{GRST}}$ = 0 due to an OMAP5912 reset, CLKG is driven by the internal clock source divided by 2, and FSG is driven low (inactive). If $\overline{\text{GRST}}$ = 0 due to program code, CLKG and FSG are both driven low (inactive). $\overline{\text{GRST}}$ = 1: Sample rate generator is enabled. CLKG is driven according to the configuration programmed in the sample rate generator registers (SRGR[1,2]). If $\overline{\text{FRST}}$ = 1, the generator also generates the frame-synchronization signal FSG as programmed in the sample rate generator registers.	R/W	0
SPCR2	7	$\overline{\text{FRST}}$	Frame-synchronization logic reset $\overline{\text{FRST}}$ = 0: Frame-synchronization logic is reset. The sample rate generator does not generate frame-synchronization signal FSG, even if $\overline{\text{GRST}}$ = 1. $\overline{\text{FRST}}$ = 1: If $\overline{\text{GRST}}$ = 1, frame-synchronization signal FSG is generated after (FPER + 1) number of CLKG clock cycles; all frame counters are loaded with their programmed values.	R/W	0

2.5.2.1 Reset Considerations

The serial port can be reset in the following two ways:

1) An OMAP5912 reset places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed, $\overline{\text{GRST}}$ = $\overline{\text{FRST = RRST = XRST}}$ = 0, keeping the entire serial port in the reset state.

2) The serial port transmitter and receiver can be reset directly using the $\overline{\text{RRST}}$ and $\overline{\text{XRST}}$ bits in the serial port control registers. The sample rate generator can be reset directly using the $\overline{\text{GRST}}$ bit in SPCR2.

Table 13 shows the McBSP signal state when the serial port is reset due to an OMAP5912 device reset and a direct receiver/transmitter reset. Note that the actual pin state depends on the device’s pin signal multiplexing mode, as described on the device datasheet.

Table 13. Reset State of Each McBSP Pin

Pin	Possible State(s)	State Forced By OMAP5912 Reset	State Forced By Receiver/Transmitter Reset
			Receiver reset ($\overline{\text{RRST}}$ = 0 and $\overline{\text{GRST}}$ = 1)
DR	I	Input	Input
CLKR	I/O/Z	Input	Known state if input; CLKR running if output
FSR	I/O/Z	Input	Known state if input; FSRP inactive state if output
CLKS	I/O/Z	Input	Input
			Transmitter reset ($\overline{\text{XRST}}$ = 0 and $\overline{\text{GRST}}$ = 1)
DX	O	Output	Output, drives low after bit clock provided
DXZ	O/Z	Output/high impedance	Output, high impedance (OMAP5912 only) after bit clock provided
CLKX	I/O/Z	Input	Known state if input; CLKX running if output
FSX	I/O/Z	Input	Known state if input; FSXP inactive state if output

Note: In possible state(s) column, I = input, O = output, Z = high impedance

For more details about McBSP reset conditions and effects, see section 2.11.

2.5.3 Set the Receiver Pins to Operate as McBSP Pins

The RIOEN bit, described in Table 14, determines whether the receiver pins are McBSP pins or general-purpose I/O pins.

Table 14. Register Bit Used to Set Receiver Pins to Operate as McBSP Pins

Register	Bit	Name	Function		Type	Reset
PCR	12	RIOEN	Receive I/O enable		R/W	0
			This bit is only applicable when the receiver is in the reset state ($\overline{\text{RRST}}$ = 0 in SPCR1).			
			RIOEN = 0	The DR, FSR, CLKR, and CLKS pins are configured as serial port pins and do not function as general-purpose I/O pins.		
			RIOEN = 1	The DR pin is a general-purpose input pin. The FSR and CLKR pins are general-purpose I/O pins. These serial port pins do not perform serial port operations. The CLKS pin is a general-purpose input pin if RIOEN = XIOEN = 1 and $\overline{\text{RRST}}$ = $\overline{\text{XRST}}$ = 0. For more information on using these pins as general-purpose I/O pins, see section 2.9.		

2.5.4 Enable/Disable the Digital Loopback Mode

The DLB bit determines whether the digital loopback mode is on, as described in Table 15.

Table 15. Register Bit Used to Enable/Disable the Digital Loopback Mode

Register	Bit	Name	Function		Type	Reset
SPCR1	15	DLB	Digital loopback mode		R/W	0
			DLB = 0	Digital loopback mode is disabled.		
			DLB = 1	Digital loopback mode is enabled.		

2.5.4.1 Digital Loopback Mode

In the digital loopback mode, the receive signals are connected internally through multiplexers to the corresponding transmit signals, as shown in Table 16. This mode allows testing of serial port code with a single DSP device; the McBSP receives the data it transmits.

Table 16. Receive Signals Connected to Transmit Signals in Digital Loopback Mode

Receive Signal	Receive Signal Fed Internally by-Transmit Signal
DR (receive data)	DX (transmit data)
FSR (receive frame synchronization)	FSX (transmit frame synchronization)
CLKR (receive clock)	CLKX (transmit clock)

2.5.5 Enable/Disable the Clock Stop Mode

The CLKSTP bits determine whether the clock stop mode is on. CLKSTP is described in Table 17.

Table 17. Register Bits Used to Enable/Disable the Clock Stop Mode

Register	Bit	Name	Function		Type	Reset
SPCR1	12:11	CLKSTP	Clock stop mode		R/W	00
			CLKSTP = 0Xb	Clock stop mode disabled; normal clocking for non-SPI mode.		
			CLKSTP = 10b	Clock stop mode enabled, without clock delay.		
			CLKSTP = 11b	Clock stop mode enabled, with clock delay.		

2.5.5.1 Clock Stop Mode

The clock stop mode supports the SPI master-slave protocol. If the SPI protocol will not be used, clear CLKSTP to disable the clock stop mode.

In the clock stop mode, the clock stops at the end of each data transfer. At the beginning of each data transfer, the clock starts immediately (CLKSTP = 10b) or after a half-cycle delay (CLKSTP = 11b). The CLKXP bit determines whether the starting edge of the clock on the CLKX pin is rising or falling. The CLKRP bit determines whether receive data is sampled on the rising or falling edge of the clock shown on the CLKR pin.

Table 18 summarizes the impact of CLKSTP, CLKXP, and CLKRP on serial port operation. In the clock stop mode, the receive clock is tied internally to the transmit clock, and the receive frame-synchronization signal is tied internally to the transmit frame-synchronization signal.

Table 18. Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme

Bit Settings	Clock Scheme

CLKSTP = 00b or 01b
CLKXP = 0 or 1
CLKRP = 0 or 1
	Clock stop mode disabled. Clock enabled for non-SPI mode.

CLKSTP = 10b
CLKXP = 0
CLKRP = 0
	Low inactive state without delay: The McBSP transmits data on the rising edge of CLKX and receives data on the falling edge of CLKR.

CLKSTP = 11b
CLKXP = 0
CLKRP = 1
	Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of CLKX and receives data on the rising edge of CLKR.

CLKSTP = 10b
CLKXP = 1
CLKRP = 0
	High inactive state without delay: The McBSP transmits data on the falling edge of CLKX and receives data on the rising edge of CLKR.

CLKSTP = 11b
CLKXP = 1
CLKRP = 1
	High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of CLKX and receives data on the falling edge of CLKR.

For more details on the SPI master-slave protocol, see section 2.4.

2.5.6 Enable/Disable the Receive Multichannel Selection Mode

The RMCM bit determines whether the receive multichannel selection mode is on. RMCM is described in Table 19.

Table 19. Register Bit Used to Enable/Disable the Receive Multichannel Selection Mode

Register	Bit	Name	Function		Type	Reset
MCR1	0	RMCM	Receive multichannel selection mode		R/W	0
			RMCM = 0	The mode is disabled. All 128 channels are enabled.		
			RMCM = 1	The mode is enabled. Channels can be individually enabled or disabled. Only the channels selected in the appropriate receive channel enable registers (RCERs) are enabled. The way channels are assigned to the RCERs depends on the number of receive channel partitions (2 or 8), as defined by the RMCME bit.		

For more details, see section 2.8.6.

2.5.7 Choose One or Two Phases for the Receive Frame

The RPHASE bit (see Table 20) determines whether the receive data frame has one or two phases.

Table 20. Register Bit Used to Choose One or Two Phases for the Receive Frame

Register	Bit	Name	Function		Type	Reset
RCR2	15	RPHASE	Receive phase number		R/W	0
			Specifies whether the receive frame has 1 or 2 phases.			
			RPHASE = 0	Single-phase frame		
			RPHASE = 1	Dual-phase frame		

2.5.8 Set the Receive Word Length(s)

The RWDLEN1 and RWDLEN2 bit fields (see Table 21) determine how many bits are in each serial word in phase 1 and in phase 2, respectively, of the receive data frame.

Table 21. Register Bits Used to Set the Receive Word Length(s)

Register	Bit	Name	Function		Type	Reset
RCR1	7:5	RWDLEN1	Receive word length 1. Specifies the length of every serial word in phase 1 of the receive frame.		R/W	000
			RWDLEN1 = 000	8 bits		
			RWDLEN1 = 001	12 bits		
			RWDLEN1 = 010	16 bits		
			RWDLEN1 = 011	20 bits		
			RWDLEN1 = 100	24 bits		
			RWDLEN1 = 101	32 bits		
			RWDLEN1 = 11X	Reserved		
RCR2	7:5	RWDLEN2	Receive word length 2. If a dual-phase frame is selected, RWDLEN2 specifies the length of every serial word in phase 2 of the frame.		R/W	000
			RWDLEN2 = 000	8 bits		
			RWDLEN2 = 001	12 bits		
			RWDLEN2 = 010	16 bits		
			RWDLEN2 = 011	20 bits		
			RWDLEN2 = 100	24 bits		
			RWDLEN2 = 101	32 bits		
			RWDLEN2 = 11X	Reserved		

2.5.8.1 Word Length Bits

Each frame can have one or two phases, depending on the value that you load into the RPHASE bit. If a single-phase frame is selected, RWDLEN1 selects the length for every serial word received in the frame. If a dual-phase frame is selected, RWDLEN1 determines the length of the serial words in phase 1 of the frame and RWDLEN2 determines the word length in phase 2 of the frame.

2.5.9 Set the Receive Frame Length

The RFRLEN1 and RFRLEN2 bit fields (see Table 22) determine how many serial words are in phase 1 and in phase 2, respectively, of the receive data frame.

Table 22. Register Bits Used to Set the Receive Frame Length

Register	Bit	Name	Function		Type	Reset
RCR1	14:8	RFRLEN1	Receive frame length 1		R/W	000 0000
			(RFRLEN1 + 1) is the number of serial words in phase 1 of the receive frame.			
			RFRLEN1 = 000 0000	1 word in phase 1		
			RFRLEN1 = 000 0001	2 words in phase 1		
			|	|		
			|	|		
			RFRLEN1 = 111 1111	128 words in phase 1		
RCR2	14:8	RFRLEN2	Receive frame length 2		R/W	000 0000
			If a dual-phase frame is selected, (RFRLEN2 + 1) is the number of serial words in phase 2 of the receive frame.			
			RFRLEN2 = 000 0000	1 word in phase 2		
			RFRLEN2 = 000 0001	2 words in phase 2		
			|	|		
			|	|		
			RFRLEN2 = 111 1111	128 words in phase 2		

2.5.9.1 Selected Frame Length

The receive frame length is the number of serial words in the receive frame. Each frame can have one or two phases, depending on value that you load into the RPHASE bit.

If a single-phase frame is selected (RPHASE = 0), the frame length is equal to the length of phase 1. If a dual-phase frame is selected (RPHASE = 1), the frame length is the length of phase 1, plus the length of phase 2.

The 7-bit RFRLEN fields allow up to 128 words per phase. See Table 23 for a summary of how to calculate the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization pulse.

Program the RFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For the example, if you want a phase length of 128 words in phase 1, load 127 into RFRLEN1.

Table 23. How to Calculate the Length of the Receive Frame

RPHASE	RFRLEN1	RFRLEN2	Frame Length
0	0 ≤ RFRLEN1 ≤ 127	Don’t care	(RFRLEN1 + 1) words
1	0 ≤ RFRLEN1 ≤ 127	0 ≤ RFRLEN2 ≤ 127	(RFRLEN1 + 1) + (RFRLEN2 + 1) words

2.5.10 Enable/Disable the Receive Frame-Synchronization Ignore Function

The RFIG bit (see Table 24) controls the receive frame-synchronization ignore function.

Table 24. Register Bit Used to Enable/Disable the Receive Frame-Synchronization Ignore Function

Register	Bit	Name	Function		Type	Reset
RCR2	2	RFIG	Receive frame-synchronization ignore		R/W	0
			RFIG = 0	An unexpected receive frame-synchronization pulse causes the McBSP to restart the frame transfer.		
			RFIG = 1	The McBSP ignores unexpected receive frame-synchronization pulses.		

2.5.10.1 Unexpected Frame-Synchronization Pulses and the Frame-Synchronization Ignore Function

If a frame-synchronization pulse occurs before the current frame is fully received, this pulse is treated as an unexpected frame-synchronization pulse.

When RFIG = 1, reception continues, ignoring the unexpected frame-synchronization pulses.

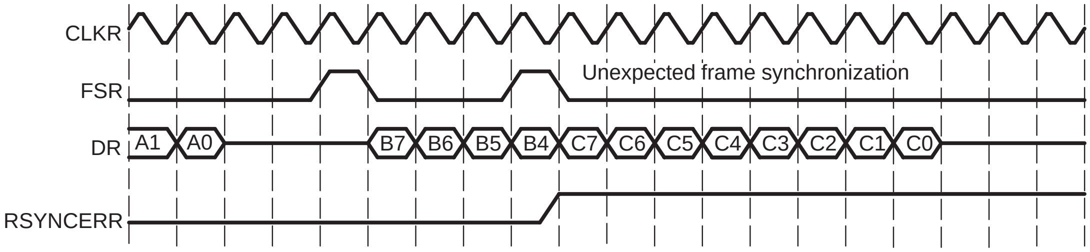
When RFIG = 0, an unexpected FSR pulse causes the McBSP to discard the contents of RSR[1,2] in favor of the new incoming data. Therefore, if RFIG = 0 and an unexpected frame-synchronization pulse occurs, the serial port aborts the current data transfer, sets RSYNCERR in SPCR1, and begins the transfer of a new data word.

For more details about the frame-synchronization error condition, see section 2.7.2.

2.5.10.2 Examples of RFIG Effects

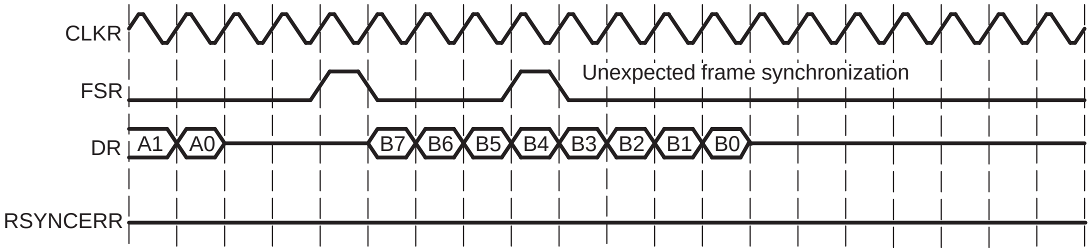
Figure 31 shows an example in which word B is interrupted by an unexpected frame-synchronization pulse when RFIG = 0. In the case of reception, the reception of B is aborted (B is lost), and a new data word (C in this example) is received after the appropriate data delay. This condition is a receive synchronization error, which sets the RSYNCERR bit.

Figure 31. Unexpected Frame-Synchronization Pulse With RFIG = 0



In contrast with Figure 31, Figure 32 shows McBSP operation when unexpected frame-synchronization signals are ignored (when RFIG = 1). In this figure, the transfer of word B is not affected by an unexpected pulse.

Figure 32. Unexpected Frame-Synchronization Pulse With RFIG = 1



2.5.11 Set the Receive Companding Mode

The RCOMPAND bits (see Table 25) determine whether companding or another data transfer option is chosen for McBSP reception.

Table 25. Register Bits Used to Set the Receive Companding Mode

Register	Bit	Name	Function		Type	Reset
RCR2	4:3	RCOMPAND	Receive companding mode		R/W	00
			Modes other than 00b are enabled only when the appropriate RWDLEN is 000b, indicating 8-bit data.			
			RCOMPAND = 00	No companding, any size data, MSB received first.		
			RCOMPAND = 01	No companding, 8-bit data, LSB received first (for details, see section 2.5.11.3).		
			RCOMPAND = 10	μ-law companding, 8-bit data, MSB received first.		
			RCOMPAND = 11	A-law companding, 8-bit data, MSB received first.		

2.5.11.1 Companding

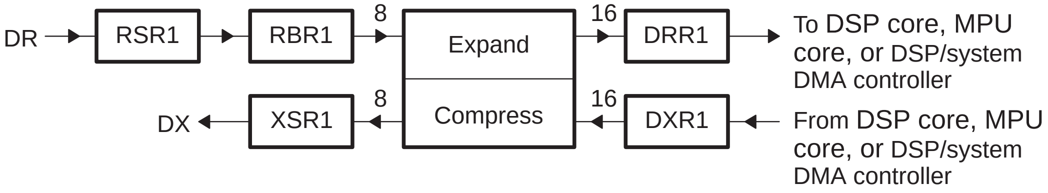
Companding (COMpressing and exPANDing) hardware allows compression and expansion of data in either μ-law or A-law format. The companding standard employed in the United States and Japan is μ-law. The European companding standard is referred to as A-law. The specifications for μ-law and A-law log PCM are part of the CCITT G.711 recommendation.

A-law and μ-law allow 13 bits and 14 bits of dynamic range, respectively. Any values outside this range are set to the most positive or most negative value. Thus, for companding to work best, the data transferred to and from the McBSP via the DSP core, MPU core, or DSP/system DMA controller must be at least 16 bits wide.

The μ-law and A-law formats both encode data into 8-bit code words. Companded data is always 8 bits wide; the appropriate word length bits (RWDLEN1, RWDLEN2, XWDLEN1, XWDLEN2) must therefore be cleared, indicating an 8-bit wide serial data stream. If companding is enabled and either of the frame phases does not have an 8-bit word length, companding continues as if the word length is 8 bits.

Figure 33 illustrates the companding processes. When companding is chosen for the transmitter, compression occurs during the process of copying data from DXR1 to XSR1. The transmit data is encoded according to the specified companding law (A-law or µ-law). When companding is chosen for the receiver, expansion occurs during the process of copying data from RBR1 to DRR1. The receive data is decoded to twos-complement format.

Figure 33. Companding Processes for Reception and for Transmission


DR
RSR1
RBR1
8
Expand
16
DRR1
To DSP core, MPU core, or DSP/system DMA controller
DX
XSR1
8
Compress
16
DXR1
From DSP core, MPU core, or DSP/system DMA controller

2.5.11.2 Format of Expanded Data

For reception, the 8-bit compressed data in RBR1 is expanded to left-justified 16-bit data in DRR1. The RJUST bit in SPCR1 is ignored when companding is used.

2.5.11.3 Option to Receive LSB First

Normally, the McBSP transmits or receives all data with the most significant bit (MSB) first. However, certain 8-bit data protocols that do not use companded data require the least significant bit (LSB) to be transferred first. If you set RCOMPAND = 01b in RCR2, the bit ordering of 8-bit words is reversed during reception. Similar to companding, this feature is enabled only if the appropriate word length bits are cleared, indicating that 8-bit words are to be transferred serially. If either phase of the frame does not have an 8-bit word length, the McBSP assumes the word length is eight bits and LSB-first ordering is done.

2.5.12 Set the Receive Data Delay

The RDATDLY bits (see Table 26) determine the length of the data delay for the receive frame.

Table 26. Register Bits Used to Set the Receive Data Delay

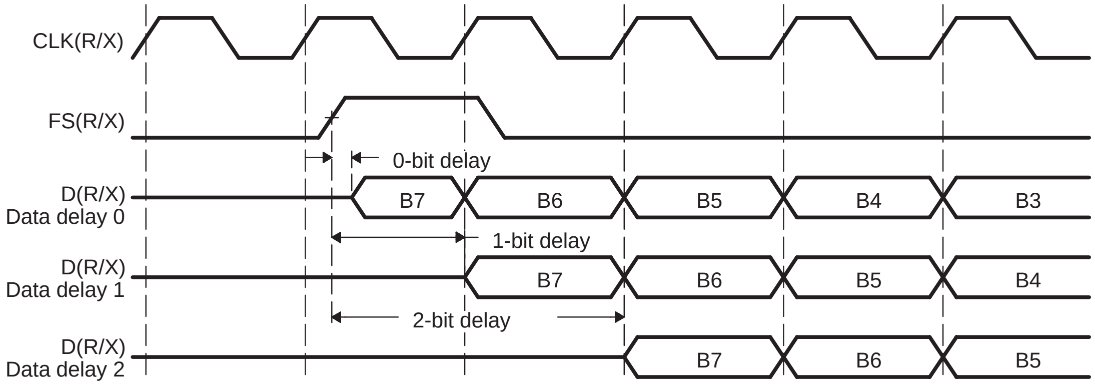
Register	Bit	Name	Function		Type	Reset
RCR2	1:0	RDATDLY	Receive data delay		R/W	00
			RDATDLY = 00	0-bit data delay		
			RDATDLY = 01	1-bit data delay		
			RDATDLY = 10	2-bit data delay		
			RDATDLY = 11	Reserved		

2.5.12.1 Data Delay

The start of a frame is defined by the first clock cycle in which frame synchronization is found to be active. The beginning of actual data reception or transmission with respect to the start of the frame can be delayed if required. This delay is called data delay.

RDATDLY specifies the data delay for reception. The range of programmable data delay is zero to two bit-clocks (RDATDLY = 00b–10b), as described in Table 26 and shown in Figure 34. In Figure 34, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on. Typically a 1-bit delay is selected, because data often follows a 1-cycle active frame-synchronization pulse.

Figure 34. Range of Programmable Data Delay



2.5.12.2 0-Bit Data Delay

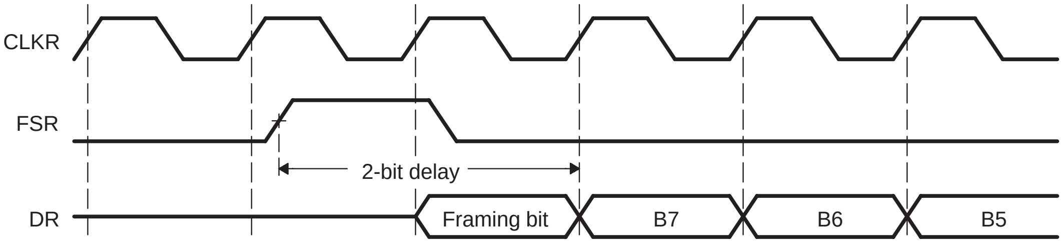
Typically, a frame-synchronization pulse is detected or sampled with respect to an edge of the internal serial clock CLK(R/X). Thus, on the following cycle or later (depending on the data delay value), data may be received or transmitted. However, in the case of 0-bit data delay, the data must be ready for reception and/or transmission on the same serial clock cycle.

In reception, receive data is sampled on the first falling edge of CLKR where an active-high internal FSR is detected, thus solving the problem. However, data transmission must begin on the rising edge of the internal CLKX clock that generated the frame synchronization. Therefore, the first data bit is assumed to be present in XSR1, and thus on DX. The transmitter then asynchronously detects the frame-synchronization signal (FSX) going active high and immediately starts driving the first bit to be transmitted on the DX pin.

2.5.12.3 2-Bit Data Delay

A data delay of two bit periods allows the serial port to interface to devices where the data stream is preceded by a framing bit. During reception of such a stream with data delay of two bits (framing bit appears after a 1-bit delay and data appears after a 2-bit delay), the serial port essentially discards the framing bit from the data stream, as shown in Figure 35. In this figure, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on.

Figure 35. 2-Bit Data Delay Used to Skip a Framing Bit



2.5.13 Set the Receive Sign-Extension and Justification Mode

The RJUST bits (see Table 27) determine whether data received by the McBSP is sign-extended and how it is justified.

Table 27. Register Bits Used to Set the Receive Sign-Extension and Justification Mode

Register	Bit	Name	Function		Type	Reset
SPCR1	14:13	RJUST	Receive sign-extension and justification mode		R/W	00
			RJUST = 00	Right justify data and zero fill MSBs in DRR[1,2].		
			RJUST = 01	Right justify data and sign-extend it into the MSBs in DRR[1,2].		
			RJUST = 10	Left justify data and zero fill LSBs in DRR[1,2].		
			RJUST = 11	Reserved.		

2.5.13.1 Sign-Extension and the Justification

RJUST in SPCR1 selects whether data in RBR[1,2] is right- or left-justified (with respect to the MSB) in DRR[1,2] and whether unused bits in DRR[1,2] are filled with zeros or with sign bits.

Table 28 and Table 29 show the effects of various RJUST values. The first table shows the effect on an example 12-bit receive-data value of 0xABC. The second table shows the effect on an example 20-bit receive-data value of 0xABCDE.

Table 28. Use of RJUST Field With 12-Bit Data Value 0xABC

RJUST	Justification	Extension	Value in DRR2	Value in DRR1
00b	Right	Zero fill MSBs	0x0000	0x0ABC
01b	Right	Sign extend data into MSBs	0xFFFF	0xFABC
10b	Left	Zero fill LSBs	0x0000	0xABC0
11b	Reserved	Reserved	Reserved	Reserved

Table 29. Use of RJUST Field With 20-Bit Data Value 0xABCDE

RJUST	Justification	Extension	Value in DRR2	Value in DRR1
00b	Right	Zero fill MSBs	0x000A	0xBCDE
01b	Right	Sign extend data into MSBs	0xFFFA	0xBCDE
10b	Left	Zero fill LSBs	0xABCD	0xE000
11b	Reserved	Reserved	Reserved	Reserved

2.5.14 Set the Receive Interrupt Mode

The RINTM bits (see Table 30) determine which event generates a receive interrupt request to the DSP core or MPU core. The receive interrupt (McBSP RX interrupt, McBSP module signal RINT) informs the DSP core or MPU core of changes to the serial port status. Four options exist for configuring this interrupt, which are set by the receive interrupt mode bits, RINTM, in SPCR1.

Table 30. Register Bits Used to Set the Receive Interrupt Mode

Register	Bit	Name	Function		Type	Reset
SPCR1	5:4	RINTM	Receive interrupt mode		R/W	00
			RINTM = 00	McBSP RX interrupt (RINT) generated when RRDY changes from 0 to 1. Interrupt on every serial word by tracking the RRDY bit in SPCR1. Regardless of the value of RINTM, RRDY can be read to detect the RRDY = 1 condition.		
			RINTM = 01	McBSP RX interrupt (RINT) generated by an end-of-block or end-of-frame condition in the receive multichannel selection mode. In the multichannel selection mode, interrupt after every 16-channel block boundary has been crossed within a frame and at the end of the frame. For details, see section 2.8.8. In any other serial transfer case, this setting is not applicable and, therefore, no interrupts are generated.		
			RINTM = 10	McBSP RX interrupt (RINT) generated by a new receive frame-synchronization pulse. Interrupt on detection of receive frame-synchronization pulses. This generates an interrupt even when the receiver is in its reset state. This is done by synchronizing the incoming frame-synchronization pulse to the internal clock source and sending it to the DSP core or MPU core via McBSP RX Interrupt.		
			RINTM = 11	McBSP RX interrupt (RINT) generated when RSYNCERR is set. Interrupt on frame-synchronization error. Regardless of the value of RINTM, RSYNCERR can be read to detect this condition. For information on using RSYNCERR, see section 2.7.2.		

For more details on the interrupt generation and mapping, see section 2.12.2.

2.5.14.1 Receive Interrupt and the Associated Modes

The receive interrupt (McBSP RX Interrupt, McBSP module signal RINT) notifies the DSP core or MPU core of serial port status changes. Four options exist for configuring this interrupt, which are set by the receive interrupt mode bits, RINTM, in SPCR1.

- RINTM = 00b. Interrupt on every serial word by tracking the RRDY bit in SPCR1. Note that regardless of the value of RINTM, RRDY can be read to detect the RRDY = 1 condition.
- RINTM = 01b. In the multichannel selection mode, interrupt after every 16-channel block boundary has been crossed within a frame and at the end of the frame. In any other serial transfer case, this setting is not applicable and, therefore, no interrupts are generated.
- RINTM = 10b. Interrupt on detection of receive frame-sync pulses. This generates an interrupt even when the receiver is in its reset state. This is done by synchronizing the incoming frame-synchronization pulse to the McBSP internal input clock and sending it to the DSP core or MPU core via McBSP RX Interrupt (McBSP module signal RINT).
- RINTM = 11b. Interrupt on frame-synchronization error. Note that regardless of the value of RINTM, RSYNCERR can be read to detect this condition.

2.5.15 Set the Receive Frame-Synchronization Mode

The bits described in Table 31 determine the source for receive frame synchronization and the function of the FSR pin.

Table 31. Register Bits Used to Set the Receive Frame Synchronization Mode

Register	Bit	Name	Function		Type	Reset
PCR	10	FSRM	Receive frame-synchronization mode		R/W	0
			FSRM = 0	Receive frame synchronization is supplied by an external source via the FSR pin.		
			FSRM = 1	Receive frame synchronization is supplied by the sample rate generator. FSR is an output pin reflecting internal FSR, except when GSYNC = 1 in SRGR2.		

Table 31. Register Bits Used to Set the Receive Frame Synchronization Mode (Continued)

Register	Bit	Name	Function		Type	Reset
SRGR2	15	GSYNC	Sample rate generator clock synchronization mode. If the sample rate generator creates a frame-synchronization signal (FSG) that is derived from an external input clock, the GSYNC bit determines whether FSG is synchronized with pulses on the FSR pin.		R/W	0
			GSYNC = 0	No clock synchronization is used: CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.		
			GSYNC = 1	Clock synchronization is used. When a pulse is detected on the FSR pin: ❑ CLKG is adjusted as necessary so that it is synchronized with the input clock on the CLKS, CLKR, or CLKX pin. ❑ FSG pulses. FSG only pulses in response to a pulse on the FSR pin. The frame-synchronization period defined in FPER is ignored. For more details, see section 2.1.3.		
SPCR1	15	DLB	Digital loopback mode		R/W	0
			DLB = 0	Digital loopback mode is disabled.		
			DLB = 1	Digital loopback mode is enabled. The receive signals, including the receive frame-synchronization signal, are connected internally through multiplexers to the corresponding transmit signals.		
SPCR1	12:11	CLKSTP	Clock stop mode		R/W	00
			CLKSTP = 0Xb	Clock stop mode disabled; normal clocking for non-SPI mode.		
			CLKSTP = 10b	Clock stop mode enabled without clock delay. The internal receive clock signal (CLKR) and the internal receive frame-synchronization signal (FSR) are internally connected to their transmit counterparts, CLKX and FSX.		
			CLKSTP = 11b	Clock stop mode enabled with clock delay. The internal receive clock signal (CLKR) and the internal receive frame-synchronization signal (FSR) are internally connected to their transmit counterparts, CLKX and FSX.		

2.5.15.1 Receive Frame-Synchronization Modes

Table 32 shows how you can select various sources to provide the receive frame-synchronization signal and the effect on the FSR pin. The polarity of the signal on the FSR pin is determined by the FSRP bit.

In digital loopback mode (DLB = 1), the transmit frame-synchronization signal is used as the receive frame-synchronization signal.

Also, in the clock stop mode, the internal receive clock signal (CLKR) and the internal receive frame-synchronization signal (FSR) are internally connected to their transmit counterparts, CLKX and FSX.

Table 32. Sources to Provide the Receive Frame-Synchronization Signal

DLB	FSRM	GSYNC	Source of Receive Frame Synchronization	FSR Pin Status
0	0	0 or 1	An external frame-synchronization signal enters the McBSP through the FSR pin. The signal is then inverted as determined by FSRP before being used as internal FSR.	Input.
0	1	0	Internal FSR is driven by the sample rate generator frame-synchronization signal (FSG).	Output. FSG is inverted as determined by FSRP before being driven out on the FSR pin.
0	1	1	Internal FSR is driven by the sample rate generator frame-synchronization signal (FSG).	Input. The external frame-synchronization input on the FSR pin is used to synchronize CLKG and generate FSG pulses.
1	0	0	Internal FSX drives internal FSR.	High impedance.
1	0 or 1	1	Internal FSX drives internal FSR.	Input. If the sample rate generator is running, external FSR is used to synchronize CLKG and generate FSG pulses.
1	1	0	Internal FSX drives internal FSR.	Output. Receive (same as transmit) frame synchronization is inverted as determined by FSRP before being driven out on the FSR pin.

2.5.16 Set the Receive Frame-Synchronization and Receive Clock Polarity

The FSRP bit (see Table 33) determines whether frame-synchronization pulses are active high or active low on the FSR pin.

Table 33. Register Bit Used to Set Receive Frame-Synchronization Polarity

Register	Bit	Name	Function		Type	Reset
PCR	2	FSRP	Receive frame-synchronization polarity		R/W	0
			FSRP = 0	Frame-synchronization pulse FSR is active high.		
			FSRP = 1	Frame-synchronization pulse FSR is active low.		

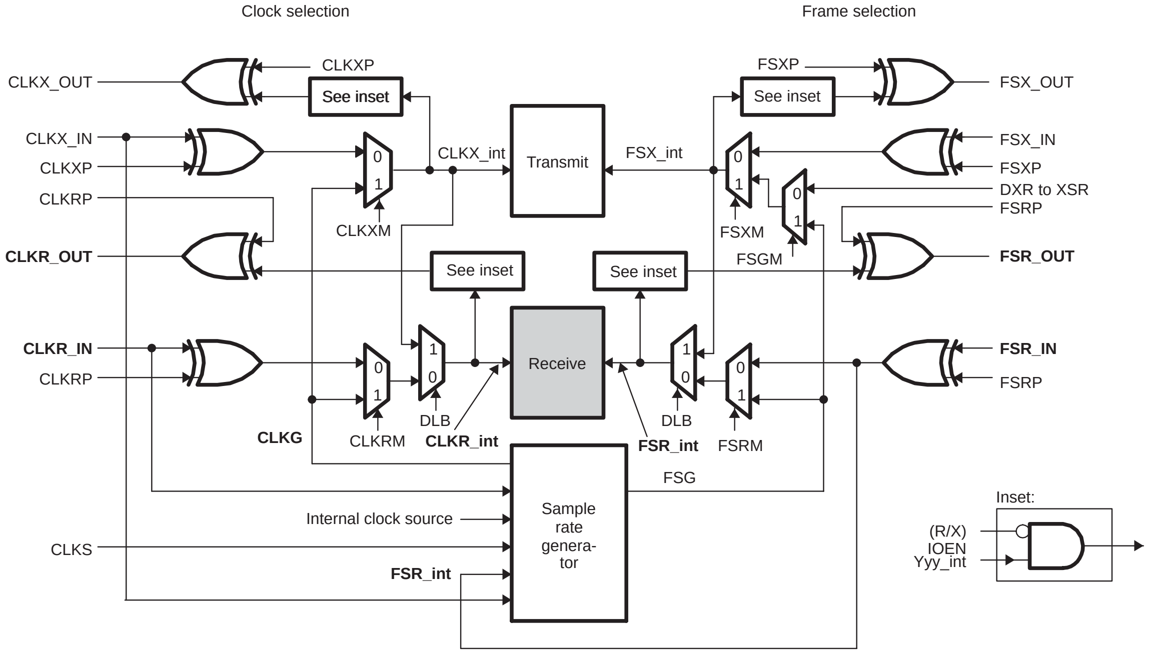
Table 34. Register Bit Used to Set Receive Clock Polarity

Register	Bit	Name	Function		Type	Reset
PCR	0	CLKRP	Receive clock polarity		R/W	0
			CLKRP = 0	Receive data sampled on falling edge of CLKR.		
			CLKRP = 1	Receive data sampled on rising edge of CLKR.		

2.5.16.1 Frame-Synchronization Pulses, Clock Signals, and Their Polarities

Receive frame-synchronization pulses can be generated internally by the sample rate generator (see section 2.1.2 and Figure 36) or driven by an external source. The source of frame synchronization is selected by programming the mode bit, FSRM, in PCR. FSR is also affected by the GSYNC bit in SRGR2. For information about the effects of FSRM and GSYNC, see section 2.5.15. Similarly, receive clocks can be selected to be inputs or outputs by programming the mode bit, CLKRM, in the PCR (see section 2.5.17). On the figure below, all _int references represent the internal version of that signal; for example, CLKR_int represents the internal CLKR.

Figure 36. Receive Frame-Synchronization Pulse Generation



When FSR and FSX are inputs (FSXM = FSRM = 0, external frame-synchronization pulses), the McBSP detects them on the internal falling edge of the clocks: internal CLKR, and internal CLKX, respectively. The receive data arriving at the DR pin is also sampled on the falling edge of internal CLKR. These internal clock signals are either derived from an external source via CLK(R/X) pins, or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When FSR and FSX are outputs, implying that they are driven by the sample rate generator, they are generated, or transition to their active state, on the rising edge of the internal clock, CLK(R/X). Similarly, data on the DX pin is output on the rising edge of internal CLKX.

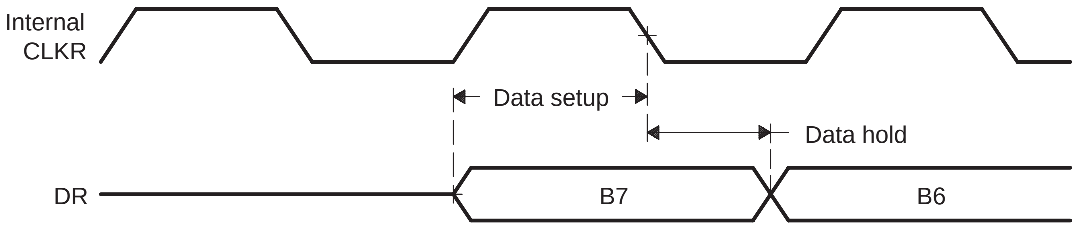
FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame-synchronization signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame synchronization (FSR/FSX are inputs to McBSP), and FSRP = FSXP = 1, the external active-low frame-synchronization signals are inverted before being sent to the receiver (internal FSR) and transmitter (internal FSX). Similarly, if internal synchronization is selected (FSR/FSX are output pins and GSYNC = 0) and the polarity bit FS(R/X)P = 1, the internal active-high frame-synchronization signals are inverted before being sent to the FS(R/X) pin.

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Data is always transmitted on the rising edge of internal CLKX. If CLKXP = 1 and external clocking is selected (CLKXM = 0 and CLKX is an input), the external falling-edge triggered input clock on CLKX is inverted to a rising-edge triggered clock before being sent to the transmitter. The external falling edge triggered clock will be seen as an internal rising edge triggered clock. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and CLKX is an output pin), the internal rising-edge triggered clock, internal CLKX, is inverted before being sent out on the CLKX pin.

Similarly, the receiver can reliably sample data that is clocked with a rising edge clock by the transmitter. The receive clock polarity bit, CLKRP, sets the edge used to sample received data. The receive data is always sampled on the falling edge of internal CLKR. Therefore, if CLKRP = 1 and external clocking is selected (CLKRM = 0 and CLKR is an input pin), the external rising-edge triggered input clock on CLKR is inverted to a falling-edge triggered clock before being sent to the receiver. If CLKRP = 1 and internal clocking is selected (CLKRM = 1), the internal falling-edge triggered clock is inverted to a rising-edge triggered clock before being sent out on the CLKR pin.

CLKRP = CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge. Figure 37 shows how data clocked by an external serial device using a rising edge can be sampled by the McBSP receiver on the falling edge of the same clock.

Figure 37. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



2.5.16.2 Frame-Synchronization Period and the Frame-Synchronization Pulse Width

The sample rate generator can produce a clock signal, CLKG, and a frame-synchronization signal, FSG. If the sample rate generator is supplying receive or transmit frame synchronization, you must program the bit fields FPER and FWID.

On FSG, the period from the start of a frame-synchronization pulse to the start of the next pulse is (FPER + 1) CLKG cycles. The 12 bits in FPER allow a frame-synchronization period of 1 to 4096 CLKG cycles, which allows up to 4096 data bits per frame. When GSYNC = 1, FPER is a don't care value.

Each pulse on FSG has a width of (FWID + 1) CLKG cycles. The eight bits in FWID allow a pulse width of 1 to 256 CLKG cycles. Program FWID to a value less than the programmed word length.

The values in FPER and FWID are loaded into separate down-counters. The 12-bit FPER counter counts down the generated clock cycles from the programmed value (4095 maximum) to 0. The 8-bit FWID counter counts down from the programmed value (255 maximum) to 0.

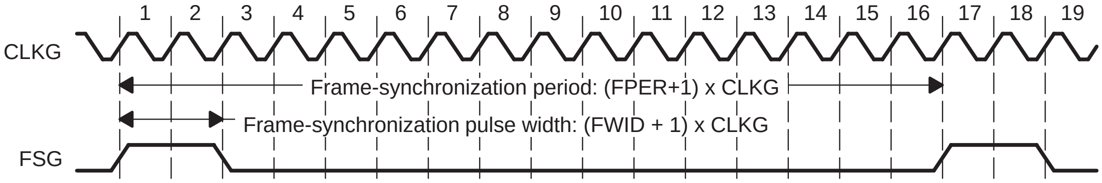
Table 35 shows settings for FPER and FWID.

Table 35. Register Bits Used to Set the SRG Frame-Synchronization Period and Pulse Width

Register	Bit	Name	Function	Type	Reset
SRGR2	11:0	FPER	Sample rate generator frame-synchronization period For the frame-synchronization signal FSG, (FPER + 1) determines the period from the start of a frame-synchronization pulse to the start of the next frame-synchronization pulse. Range for (FPER + 1) is 1 to 4096 CLKG cycles.	R/W	0000 0000 0000
SRGR1	15:8	FWID	Sample rate generator frame-synchronization pulse width This field plus 1 determines the width of each frame-synchronization pulse on FSG. Range for (FWID + 1) is 1 to 256 CLKG cycles.	R/W	0000 0000

Figure 38 shows a frame-synchronization period of 16 CLKG periods (FPER = 15 or 00001111b) and a frame-synchronization pulse with an active width of 2 CLKG periods (FWID = 1).

Figure 38. Frame of Period 16 CLKG Periods and Active Width of 2 CLKG Periods


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
CLKG
Frame-synchronization period: (FPER+1) x CLKG
Frame-synchronization pulse width: (FWID + 1) x CLKG
FSG

When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when $\overline{\text{GRST}}$ = 1 and FSGM = 1, a frame-synchronization pulse is generated. The frame width value (FWID + 1) is counted down on every CLKG cycle until it reaches 0, at which time FSG goes low. At the same time, the frame period value (FPER + 1) is also counting down. When this value reaches 0, FSG goes high, indicating a new frame.

2.5.17 Set the Receive Clock Mode

Table 36 shows the settings for bits used to set the receive clock mode.

Table 36. Register Bits Used to Set the Receive Clock Mode

Register	Bit	Name	Function		Type	Reset
PCR	8	CLKRM	Receive clock mode		R/W	0
			Case 1: Digital loopback mode not set (DLB = 0) in SPCR1.			
			CLKRM = 0	The CLKR pin is an input pin that supplies the internal receive clock (CLKR).		
			CLKRM = 1	Internal CLKR is driven by the sample rate generator of the McBSP. The CLKR pin is an output pin that reflects internal CLKR.		
			Case 2: Digital loopback mode set (DLB = 1) in SPCR1.			
			CLKRM = 0	The CLKR pin is in the high impedance state. The internal receive clock (CLKR) is driven by the internal transmit clock (CLKX). Internal CLKX is derived from the CLKXM bit in PCR.		
			CLKRM = 1	Internal CLKR is driven by internal CLKX. The CLKR pin is an output pin that reflects internal CLKR. Internal CLKX is derived from the CLKXM bit in PCR.		
SPCR1	15	DLB	Digital loopback mode		R/W	00
			DLB = 0	Digital loopback mode is disabled.		
			DLB = 1	Digital loopback mode is enabled. The receive signals, including the receive frame-synchronization signal, are connected internally through multiplexers to the corresponding transmit signals.		

Table 36. Register Bits Used to Set the Receive Clock Mode (Continued)

Register	Bit	Name	Function		Type	Reset
SPCR1	12:11	CLKSTP	Clock stop mode		R/W	00
			CLKSTP = 0Xb	Clock stop mode disabled; normal clocking for non-SPI mode.		
			CLKSTP = 10b	Clock stop mode enabled without clock delay. The internal receive clock signal (CLKR) and the internal receive frame-synchronization signal (FSR) are internally connected to their transmit counterparts, CLKX and FSX.		
			CLKSTP = 11b	Clock stop mode enabled with clock delay. The internal receive clock signal (CLKR) and the internal receive frame-synchronization signal (FSR) are internally connected to their transmit counterparts, CLKX and FSX.		

2.5.17.1 Selecting a Source for the Receive Clock and a Data Direction for the CLKR Pin

Table 37 shows how to select various sources to provide the receive clock signal and affect the CLKR pin. The polarity of the signal on the CLKR pin is determined by the CLKRP bit.

In the digital loopback mode (DLB = 1), the transmit clock signal is used as the receive clock signal. Also, the internal receive clock signal (CLKR) and the internal receive frame-synchronization signal (FSR) are internally connected to their transmit counterparts (CLKX and FSX) in the clock stop mode.

Table 37. Receive Clock Signal Source Selection

DLB in SPCR1	CLKRM in PCR	Source of Receive Clock	CLKR Pin Status
0	0	The CLKR pin is input driven by an external clock. The external clock signal is inverted as determined by CLKRP before being used.	Input.
0	1	The sample rate generator clock (CLKG) drives internal CLKR.	Output. CLKG, inverted as determined by CLKRP, is driven out on the CLKR pin.
1	0	Internal CLKX drives internal CLKR. To configure CLKX, see section 2.6.18.	High impedance.
1	1	Internal CLKX drives internal CLKR. To configure CLKX, see section 2.6.18.	Output. Internal CLKR (same as internal CLKX) is inverted as determined by CLKRP before being driven out on the CLKR pin.

2.5.18 Set the SRG Clock Divide-Down Value

Table 38. Register Bits Used to Set the Sample Rate Generator (SRG) Clock Divide-Down Value

Register	Bit	Name	Function	Type	Reset
SRGR1	7:0	CLKGDV	Sample rate generator clock divide-down value The input clock of the sample rate generator is divided by (CLKGDV + 1) to generate the required sample rate generator clock frequency. The default value of CLKGDV is 1 (divide input clock by 2).	R/W	0000 0001

2.5.18.1 Sample Rate Generator Clock Divider

The first divider stage generates the serial data bit clock from the input clock. This divider stage utilizes a counter, preloaded by CLKGDV, that contains the divide ratio value.

The output of the first divider stage is the data bit clock, which is output as CLKG and which serves as the input for the second and third stages of the divider.

CLKG has a frequency equal to 1/(CLKGDV + 1) of the sample rate generator input clock. Thus, the sample generator input clock frequency is divided by a value between 1 and 256. When CLKGDV is odd or equal to 0, the CLKG duty cycle is 50%. When CLKGDV is an even value, (2p, representing an odd divide-down) the high-state duration is p + 1 cycles and the low-state duration is p cycles.

2.5.19 Set the SRG Clock Synchronization Mode

The GSYNC bit (see Table 39) determines the SRG clock synchronization mode. This field is only relevant when using an external clock source.

Table 39. Register Bit Used to Set the SRG Clock Synchronization Mode

Register	Bit	Name	Function		Type	Reset
SRGR2	15	GSYNC	Sample rate generator clock synchronization		R/W	0
			GSYNC is used only when the input clock source for the sample rate generator is external (on the CLKS, CLKR, or CLKX pin).			
			GSYNC = 0	The sample rate generator clock (CLKG) is free running. CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.		
			GSYNC = 1	Clock synchronization is performed. When a pulse is detected on the FSR pin: ❑ CLKG is adjusted as necessary so that it is synchronized with the input clock on the CLKS, CLKR, or CLKX pin. ❑ FSG pulses. FSG **only** pulses in response to a pulse on the FSR pin. The frame-synchronization period defined in FPER is ignored.		

For more details on using the clock synchronization feature, see section 2.1.3.

2.5.20 Set the SRG Clock Mode (Choose an Input Clock)

The bits shown in Table 40 determine the source for the SRG clock. Note that only McBSP1 features the CLKS pin.

Table 40. Register Bits Used to Set the SRG Clock Mode (Choose an Input Clock)

Register	Bit	Name	Function		Type	Reset
PCR	7	SCLKME	Sample rate generator clock mode		R/W	0
SRGR2	13	CLKSM			R/W	1
			SCLKME = 0 CLKSM = 0	Sample rate generator clock derived from CLKS pin (McBSP1 only).		
			SCLKME = 0 CLKSM = 1	Sample rate generator clock derived from internal clock source. This condition is forced by an OMAP5912 reset.		
			SCLKME = 1 CLKSM = 0	Sample rate generator clock derived from CLKR pin (McBSP2 only).		
			SCLKME = 1 CLKSM = 1	Sample rate generator clock derived from CLKX pin.		

For details on the available internal sources for each McBSP, see Table 3.

2.5.20.1 SRG Clock Mode

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both, but CLKG is derived from an input clock. Table 40 shows the four possible sources of the input clock. For more details on generating CLKG, see section 2.1.1.

2.5.21 Set the SRG Input Clock Polarity

Table 41. Register Bits Used to Set the SRG Input Clock Polarity

Register	Bit	Name	Function	Type	Reset
SRGR2	14	CLKSP	CLKS pin polarity. CLKSP determines the input clock polarity when the CLKS pin supplies the input clock (SCLKME = 0 and CLKSM = 0).	R/W	0
			CLKSP = 0 Rising edge on CLKS pin generates CLKG and FSG.		
			CLKSP = 1 Falling edge on CLKS pin generates CLKG and FSG.		
PCR	1	CLKXP	CLKX pin polarity. CLKXP determines the input clock polarity when the CLKX pin supplies the input clock (SCLKME = 1 and CLKSM = 1).	R/W	0
			CLKXP = 0 Rising edge on CLKX pin generates transitions on CLKG and FSG.		
			CLKXP = 1 Falling edge on CLKX pin generates transitions on CLKG and FSG.		
PCR	0	CLKRP	CLKR pin polarity. CLKRP determines the input clock polarity when the CLKR pin supplies the input clock (SCLKME = 1 and CLKSM = 0).	R/W	0
			CLKRP = 0 Falling edge on CLKR pin generates transitions on CLKG and FSG.		
			CLKRP = 1 Rising edge on CLKR pin generates transitions on CLKG and FSG.		

Note:

Note that the polarity mode occurs at the signal input/output. The internal clock signal (CLKG) polarity remains the same, regardless of the input clock polarity. See Figure 6 and Figure 7 for more details on this feature.

2.5.21.1 Using CLKSP/CLKXP/CLKRP to Choose an Input Clock Polarity

The sample rate generator can produce a clock signal (CLKG) and a frame-synchronization signal (FSG) for use by the receiver, the transmitter, or both. To produce CLKG and FSG, the sample rate generator must be driven by an input clock signal derived from the internal clock source or from an external clock on the CLKS (McBSP1 only), CLKX, or CLKR (McBSP2 only) pin. If you use a pin, choose a polarity for that pin by using the appropriate polarity bit: CLKSP for the CLKS pin, CLKXP for the CLKX pin, and CLKRP for the CLKR pin. The polarity determines whether the rising or falling edge of the input clock generates transitions on CLKG and FSG.

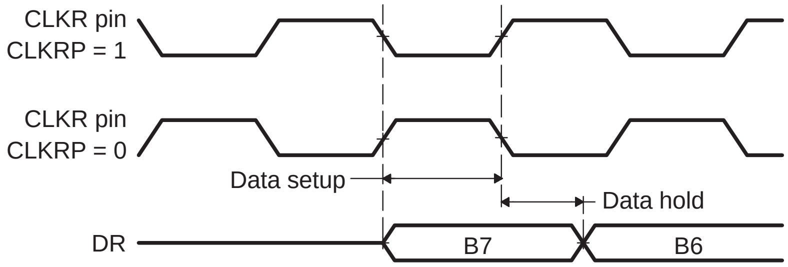
CLKXP

- ❑ If CLKXP=0, transmit data is written on the rising edge of CLKX and the external device reads the data on the falling edge of CLKX.
- ❑ If CLKXP=1, transmit data is written on the falling edge of CLKX and the external device reads the data on the rising edge of CLKX.

CLKRP

- ❑ If CLKRP=0, received data is sampled on the falling edge of CLKR.
- ❑ If CLKXR=1, received data is sampled on the rising edge of CLKR.

Figure 39. Example: Receive Clock Polarity



2.6 Transmitter Configuration

To configure the McBSP transmitter, perform the following procedure:

1) Place the McBSP/transmitter in reset (see section 2.6.2).

2) Program the McBSP registers for the desired transmitter operation (see section 2.6.1).

3) Take the transmitter out of reset (see section 2.6.2).

2.6.1 Programming the McBSP Registers for the Desired Transmitter Operation

The following is a list of important tasks to be performed to configure the McBSP transmitter. Each task corresponds to one or more McBSP register bit fields.

- Global behavior:
  - Set the transmitter pins to operate as McBSP pins
  - Enable/disable the digital loopback mode
  - Enable/disable the clock stop mode
  - Enable/disable transmit multichannel selection
- Data behavior:
  - Choose 1 or 2 phases for the transmit frame
  - Set the transmit word length(s)
  - Set the transmit frame length
  - Enable/disable the transmit frame-synchronization ignore function
  - Set the transmit companding mode
  - Set the transmit data delay
  - Set the transmit DXENA mode
  - Set the transmit interrupt mode
- Frame-synchronization behavior:
  - Set the transmit frame-synchronization mode
  - Set the transmit frame-synchronization polarity
  - Set the SRG frame-synchronization period and pulse width
- Clock behavior:
  - Set the transmit clock mode
  - Set the transmit clock polarity
  - Set the SRG clock divide-down value
  - Set the SRG clock synchronization mode
  - Set the SRG clock mode (choose an input clock)
  - Set the SRG input clock polarity

2.6.2 Resetting and Enabling the Transmitter

The first step of the transmitter configuration procedure is to reset the transmitter, and the last step is to enable the transmitter (take it out of reset). Table 42 describes the bits used for both of these steps.

Table 42. Register Bits Used to Place Transmitter in Reset

Register	Bit	Name	Function		Type	Reset
SPCR2	0	$\overline{\text{XRST}}$	Transmitter reset		R/W	0
			$\overline{\text{XRST}}$ = 0	The serial port transmitter is disabled and in the reset state.		
			$\overline{\text{XRST}}$ = 1	The serial port transmitter is enabled.		
SPCR2	6	$\overline{\text{GRST}}$	Sample rate generator reset		R/W	0
			$\overline{\text{GRST}}$ = 0	Sample rate generator is reset. If $\overline{\text{GRST}}$ = 0 due to an OMAP5912 reset, CLKG is driven by the internal clock source divided by 2, and FSG is driven low (inactive). If $\overline{\text{GRST}}$ = 0 due to program code, CLKG and FSG are both driven low (inactive).		
			$\overline{\text{GRST}}$ = 1	Sample rate generator is enabled. CLKG is driven according to the configuration programmed in the sample rate generator registers (SRGR[1,2]). If $\overline{\text{FRST}}$ = 1, the generator also generates the frame-synchronization signal FSG as programmed in the sample rate generator registers.		
SPCR2	7	$\overline{\text{FRST}}$	Frame-synchronization logic reset		R/W	0
			$\overline{\text{FRST}}$ = 0	Frame-synchronization logic is reset. The sample rate generator does not generate frame-synchronization signal FSG, even if $\overline{\text{GRST}}$ = 1.		
			$\overline{\text{FRST}}$ = 1	If $\overline{\text{GRST}}$ = 1, frame-synchronization signal FSG is generated after (FPER + 1) number of CLKG clock cycles; all frame counters are loaded with their programmed values.		

2.6.2.1 Reset Considerations

The serial port can be reset in the following two ways:

1) An OMAP5912 reset places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed, $\overline{\text{GRST}}$ = $\overline{\text{FRST}}$ = $\overline{\text{RRST}}$ = $\overline{\text{XRST}}$ = 0, keeping the entire serial port in the reset state.

2) The serial port transmitter and receiver can be reset directly using the $\overline{\text{RRST}}$ and $\overline{\text{XRST}}$ bits in the serial port control registers. The sample rate generator can be reset directly using the $\overline{\text{GRST}}$ bit in SPCR2.

Table 43 shows the state of McBSP signals when the serial port is reset due to an OMAP5912 reset and a direct receiver/transmitter reset. Note that the actual pin state depends on the device's pin signal multiplexing mode, as described on the device datasheet.

Table 43. Reset State of Each McBSP Pin

Pin	Possible State(s)	State Forced By OMAP5912 Reset	State Forced By Receiver/Transmitter Reset
			Receiver reset ($\overline{\text{RRST}}$ = 0 and $\overline{\text{GRST}}$ = 1)
DR	I	Input	Input
CLKR	I/O/Z	Input	Known state if input; CLKR running if output
FSR	I/O/Z	Input	Known state if input; FSRP inactive state if output
CLKS	I/O/Z	Input	Input
			Transmitter reset ($\overline{\text{XRST}}$ = 0 and $\overline{\text{GRST}}$ = 1)
DX	O	Output	Output, drives low after bit clock provided
DXZ	O/Z	Output/high impedance	Output, high impedance (OMAP5912 only) after bit clock provided
CLKX	I/O/Z	Input	Known state if input; CLKX running if output
FSX	I/O/Z	Input	Known state if input; FSXP inactive state if output

Note: In Possible State(s) column, I = input, O = output, Z = high impedance

For more details about McBSP reset conditions and effects, see section 2.11.

2.6.3 Set the Transmitter Pins to Operate as McBSP Pins

Table 44. Register Bit Used to Set Transmitter Pins to Operate as McBSP Pins

Register	Bit	Name	Function	Type	Reset
PCR	13	XIOEN	Transmit I/O enable	R/W	0
			This bit is only applicable when the transmitter is in the reset state (XRST = 0 in SPCR2).		
			XIOEN = 0 The DX, FSX, CLKX, and CLKS pins are configured as serial port pins and do not function as general-purpose I/Os.		
			XIOEN = 1 The DX pin is a general-purpose output pin. The FSX and CLKX pins are general-purpose I/O pins. These serial port pins do not perform serial port operations. The CLKS pin is a general-purpose input pin if $\overline{\text{RIOEN}}$ = $\overline{\text{XIOE}}$N = 1, and $\overline{\text{RRST}}$ = $\overline{\text{XRST}}$ = 0. For more information on using these pins as general-purpose I/O pins, see section 2.9.		

2.6.4 Enable/Disable the Digital Loopback Mode

The DLB bit determines whether the digital loopback mode is on, as described in Table 45.

Table 45. Register Bit Used to Enable/Disable the Digital Loopback Mode

Register	Bit	Name	Function	Type	Reset
SPCR1	15	DLB	Digital loopback mode	R/W	0
			DLB = 0 Digital loopback mode is disabled.		
			DLB = 1 Digital loopback mode is enabled.		

2.6.4.1 Digital Loopback Mode

In the digital loopback mode, the receive signals are connected internally through multiplexers to the corresponding transmit signals, as shown in Table 46. This mode allows testing of serial port code with a single DSP device, the McBSP receives the data it transmits.

Table 46. Receive Signals Connected to Transmit Signals in Digital Loopback Mode

Receive Signal	Receive Signal Fed Internally by Transmit Signal
DR (receive data)	DX (transmit data)
FSR (receive frame synchronization)	FSX (transmit frame synchronization)
CLKR (receive clock)	CLKX (transmit clock)

2.6.5 Enable/Disable the Clock Stop Mode

The CLKSTP bits determine whether the clock stop mode is on, as described in Table 47.

Table 47. Register Bits Used to Enable/Disable the Clock Stop Mode

Register	Bit	Name	Function		Type	Reset
SPCR1	12:11	CLKSTP	Clock stop mode		R/W	00
			CLKSTP = 0Xb	Clock stop mode disabled; normal clocking for non-SPI mode.		
			CLKSTP = 10b	Clock stop mode enabled without clock delay.		
			CLKSTP = 11b	Clock stop mode enabled with clock delay.		

2.6.5.1 Clock Stop Mode

The clock stop mode supports the SPI master-slave protocol. If you do not plan to use the SPI protocol, you can clear CLKSTP to disable the clock stop mode.

In the clock stop mode, the clock stops at the end of each data transfer. At the beginning of each data transfer, the clock starts immediately (CLKSTP = 10b) or after a half-cycle delay (CLKSTP = 11b). The CLKXP bit determines whether the starting edge of the clock on the CLKX pin is rising or falling. The CLKRP bit determines whether receive data is sampled on the rising or falling edge of the clock shown on the CLKR pin.

Table 48 summarizes the impact of CLKSTP, CLKXP, and CLKRP on serial port operation. In the clock stop mode, the receive clock is tied internally to the transmit clock, and the receive frame-synchronization signal is tied internally to the transmit frame-synchronization signal.

Table 48. Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme

Bit Settings	Clock Scheme

CLKSTP = 00b or 01b
CLKXP = 0 or 1
CLKRP = 0 or 1
	Clock stop mode disabled. Clock enabled for non-SPI mode.

CLKSTP = 10b
CLKXP = 0
CLKRP = 0
	Low inactive state without delay: The McBSP transmits data on the rising edge of CLKX and receives data on the falling edge of CLKR.

CLKSTP = 11b
CLKXP = 0
CLKRP = 1
	Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of CLKX and receives data on the rising edge of CLKR.

CLKSTP = 10b
CLKXP = 1
CLKRP = 0
	High inactive state without delay: The McBSP transmits data on the falling edge of CLKX and receives data on the rising edge of CLKR.

CLKSTP = 11b
CLKXP = 1
CLKRP = 1
	High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of CLKX and receives data on the falling edge of CLKR.

For more details on the SPI master-slave protocol, see section 2.4.

2.6.6 Enable/Disable Transmit Multichannel Selection

The XMCM bits described in Table 49 are used to select one of the three transmit multichannel selection modes, or to disable transmit multichannel selection.

Table 49. Register Bits Used to Enable/Disable Transmit Multichannel Selection

Register	Bit	Name	Function		Type	Reset
MCR2	1:0	XMCM	Transmit multichannel selection		R/W	00
			XMCM = 00b	No transmit multichannel selection mode is on. All channels are enabled and unmasked. No channels can be disabled or masked.		
			XMCM = 01b	All channels are disabled unless they are selected in the appropriate transmit channel enable registers (XCERs). If enabled, a channel in this mode is also unmasked. The XMCME bit determines whether 32 channels or 128 channels are selectable in the XCERs.		
			XMCM = 10b	All channels are enabled, but they are masked unless they are selected in the appropriate transmit channel enable registers (XCERs). The XMCME bit determines whether 32 channels or 128 channels are selectable in the XCERs.		
			XMCM = 11b	This mode is used for symmetric transmission and reception. All channels are disabled for transmission unless they are enabled for reception in the appropriate receive channel enable registers (RCERs). Once enabled, they are masked unless they are also selected in the appropriate transmit channel enable registers (XCERs). The XMCME bit determines whether 32 channels or 128 channels are selectable in the RCERs and XCERs.		

2.6.7 Choose One or Two Phases for the Transmit Frame

The XPHASE bit described in Table 50 is used to choose one or two phases for the transmit frame.

Table 50. Register Bit Used to Choose 1 or 2 Phases for the Transmit Frame

Register	Bit	Name	Function	Type	Reset
XCR2	15	XPHASE	Transmit phase number	R/W	0
			Specifies whether the transmit frame has 1 or 2 phases.		
			XPHASE = 0 Single-phase frame.		
			XPHASE = 1 Dual-phase frame.		

2.6.8 Set the Transmit Word Length(s)

The XWDLEN1 and XWDLEN2 fields (Table 51) are used to set the transmit word length(s).

Table 51. Register Bits Used to Set the Transmit Word Length(s)

Register	Bit	Name	Function	Type	Reset
XCR1	7:5	XWDLEN1	Transmit word length of frame phase 1	R/W	000
			XWDLEN1 = 000b 8 bits		
			XWDLEN1 = 001b 12 bits		
			XWDLEN1 = 010b 16 bits		
			XWDLEN1 = 011b 20 bits		
			XWDLEN1 = 100b 24 bits		
			XWDLEN1 = 101b 32 bits		
			XWDLEN1 = 11Xb Reserved		
XCR2	7:5	XWDLEN2	Transmit word length of frame phase 2	R/W	000
			XWDLEN2 = 000b 8 bits		
			XWDLEN2 = 001b 12 bits		
			XWDLEN2 = 010b 16 bits		
			XWDLEN2 = 011b 20 bits		
			XWDLEN2 = 100b 24 bits		
			XWDLEN2 = 101b 32 bits		
			XWDLEN2 = 11Xb Reserved		

2.6.8.1 Word Length Bits

Each frame can have one or two phases, depending on the value loaded in the RPHASE bit. If a single-phase frame is selected, XWDLEN1 selects the length for every serial word transmitted in the frame. If a dual-phase frame is selected, XWDLEN1 determines the length of the serial words in phase 1 of the frame, and XWDLEN2 determines the word length in phase 2 of the frame.

2.6.9 Set the Transmit Frame Length

Table 52. Register Bits Used to Set the Transmit Frame Length

Register	Bit	Name	Function		Type	Reset
XCR1	14:8	XFRLEN1	Transmit frame length 1		R/W	000 0000
			(XFRLEN1 + 1) is the number of serial words in phase 1 of the transmit frame.			
			XFRLEN1 = 000 0000	1 word in phase 1		
			XFRLEN1 = 000 0001	2 words in phase 1		
			|	|		
			|	|		
			XFRLEN1 = 111 1111	128 words in phase 1		
XCR2	14:8	XFRLEN2	Transmit frame length 2		R/W	000 0000
			If a dual-phase frame is selected, (XFRLEN2 + 1) is the number of serial words in phase 2 of the transmit frame.			
			XFRLEN2 = 000 0000	1 word in phase 2		
			XFRLEN2 = 000 0001	2 words in phase 2		
			|	|		
			|	|		
			XFRLEN2 = 111 1111	128 words in phase 2		

2.6.9.1 Selected Frame Length

The transmit frame length is the number of serial words in the transmit frame. Each frame can have one or two phases, depending on the value loaded into the XPHASE bit.

If a single-phase frame is selected (XPHASE = 0), the frame length is equal to the length of phase 1. If a dual-phase frame is selected (XPHASE = 1), the frame length is the length of phase 1 plus the length of phase 2.

The 7-bit XFRLEN fields allow up to 128 words per phase. See Table 53 for how to calculate the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization pulse.

Note:

Program the XFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For example, if you want a phase length of 128 words in phase 1, load 127 into XFRLEN1.

Table 53. How to Calculate Frame Length

XPHASE	XFRLEN1	XFRLEN2	Frame Length
0	0 ≤ XFRLEN1 ≤ 127	Don't care	(XFRLEN1 + 1) words
1	0 ≤ XFRLEN1 ≤ 127	0 ≤ XFRLEN2 ≤ 127	(XFRLEN1 + 1) + (XFRLEN2 + 1) words

2.6.10 Enable/Disable the Transmit Frame-Synchronization Ignore Function

Table 54. Register Bit Used to Enable/Disable the Transmit Frame-Synchronization Ignore Function

Register	Bit	Name	Function		Type	Reset
XCR2	2	XFIG	Transmit frame-synchronization ignore		R/W	0
			XFIG = 0	An unexpected transmit frame-synchronization pulse causes the McBSP to restart the frame transfer.		
			XFIG = 1	The McBSP ignores unexpected transmit frame-synchronization pulses.		

2.6.10.1 Unexpected Frame-Synchronization Pulses and Frame-Synchronization Ignore

If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame-synchronization pulse.

If XFIG = 1, normal transmission continues and unexpected frame-synchronization signals are ignored.

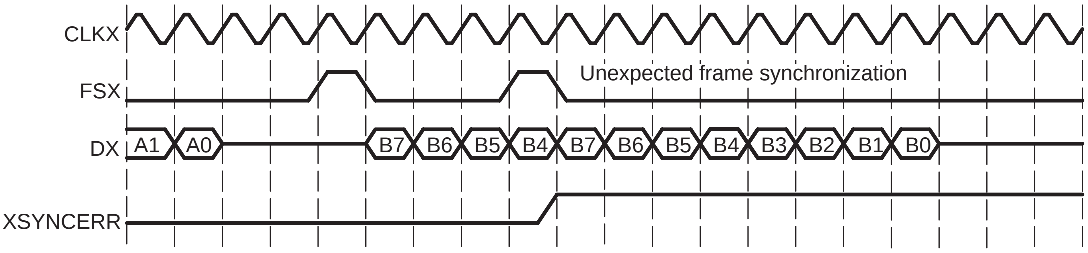
When XFIG = 0 and an unexpected frame-synchronization pulse occurs, the serial port aborts the present transmission, sets XSYNCERR to 1 in SPCR2, and reinitiates transmission of the current word that was aborted.

For more details about the frame-synchronization error condition, see section 2.7.5.

2.6.10.2 Examples Showing the Effects of XFIG

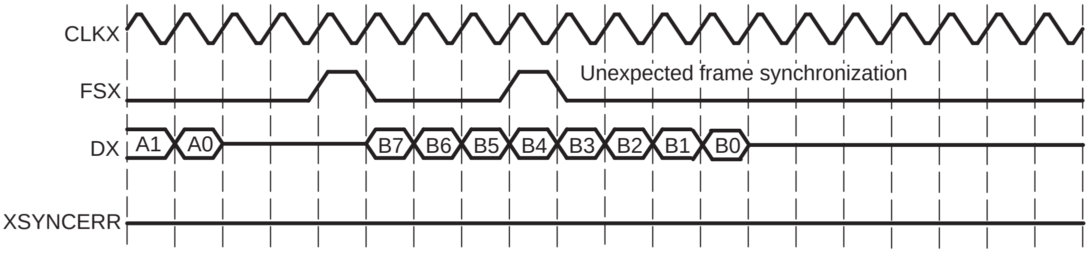
Figure 40 shows an example in which word B is interrupted by an unexpected frame-synchronization pulse when XFIG = 0. In the case of transmission, the transmission of B is aborted, and B is re-sent . This condition is a transmit synchronization error, which sets the XSYNCERR bit. No new data has been written to DXR[1,2]; therefore, the McBSP transmits B again.

Figure 40. Unexpected Frame-Synchronization Pulse With XFIG = 0



In contrast with Figure 40, Figure 41 shows McBSP operation when unexpected frame-synchronization signals are ignored if XFIG = 1. In this figure, the transfer of word B is not affected by an unexpected frame-synchronization pulse.

Figure 41. Unexpected Frame-Synchronization Pulse With XFIG = 1



2.6.11 Set the Transmit Companding Mode

Table 55. Register Bits Used to Set the Transmit Companding Mode

Register	Bit	Name	Function		Type	Reset
XCR2	4:3	XCOMPAND	Transmit companding mode		R/W	00
			Modes other than 00b are enabled only when the appropriate XWDLEN is 000b, indicating 8-bit data.			
			XCOMPAND = 00b	No companding, any size data, MSB transmitted first.		
			XCOMPAND = 01b	No companding, 8-bit data, LSB transmitted first (for details, see section 2.5.11.3).		
			XCOMPAND = 10b	μ-law companding, 8-bit data, MSB transmitted first.		
			XCOMPAND = 11b	A-law companding, 8-bit data, MSB transmitted first.		

2.6.11.1 Companding

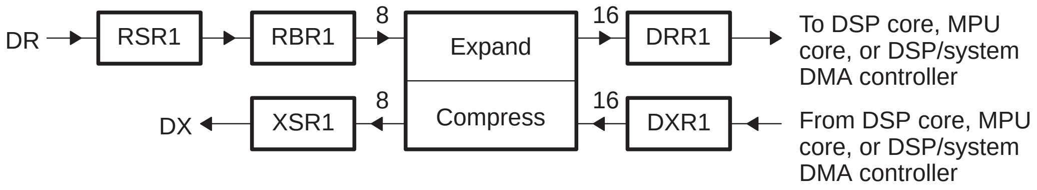
Companding (COMpressing and exPANDing) hardware allows compression and expansion of data in either μ-law or A-law format. The companding standard employed in the United States and Japan is μ-law. The European companding standard is referred to as A-law. The specifications for μ-law and A-law log PCM are part of the CCITT G.711 recommendation.

A-law and μ-law allow 13 bits and 14 bits of dynamic range, respectively. Any values outside this range are set to the most positive or most negative value. Thus, for companding to work best, the data transferred to and from the McBSP via the DSP core, MPU core, or DSP/system DMA controller must be at least 16 bits wide.

The μ-law and A-law formats both encode data into 8-bit code words. Companded data is always 8 bits wide; the appropriate word length bits (RWDLEN1, RWDLEN2, XWDLEN1, XWDLEN2) must therefore be cleared, indicating an 8-bit wide serial data stream. If companding is enabled and either of the frame phases does not have an 8-bit word length, companding continues as if the word length is 8 bits.

Figure 42 illustrates the companding processes. When companding is chosen for the transmitter, compression occurs during the process of copying data from DXR1 to XSR1. The transmit data is encoded according to the specified companding law (A-law or μ-law). When companding is chosen for the receiver, expansion occurs during the process of copying data from RBR1 to DRR1. The receive data is decoded to twos-complement format.

Figure 42. Companding Processes for Reception and for Transmission


DR
RSR1
RBR1
8
Expand
16
DRR1
To DSP core, MPU core, or DSP/system DMA controller
DX
XSR1
8
Compress
16
DXR1
From DSP core, MPU core, or DSP/system DMA controller

2.6.11.2 Data Format for Compression

For transmission using μ-law compression, make sure the 14 data bits are left-justified in DXR1, with the remaining two low-order bits filled with 0s as shown in Figure 43.

Figure 43. µ-Law Transmit Data Companding Format

	15-2	1-0
µ-law format in DXR1	Value	00

For transmission using A-law compression, make sure the 13 data bits are left-justified in DXR1, with the remaining three low-order bits filled with 0s as shown in Figure 44.

Figure 44. A-Law Transmit Data Companding Format

	15-3	2-0
A-law format in DXR1	Value	000

2.6.11.3 Option to Transmit LSB First

Typically, the McBSP transmits or receives all data with the most significant bit (MSB) first. However, certain 8-bit data protocols (that do not use companded data) require the least significant bit (LSB) to be transferred first. If you set XCOMPAND = 01b in XCR2, the bit ordering of 8-bit words is reversed (LSB first) before being sent from the serial port. Similar to companding, this feature is enabled only if the appropriate word length bits are cleared, indicating that 8-bit words are to be transferred serially. If either phase of the frame does not have an 8-bit word length, the McBSP assumes the word length is eight bits and does LSB-first ordering.

2.6.12 Set the Transmit Data Delay

Use the XDATDLY bits (see Table 56) to select a delay of 0, 1, or 2 bits after a transmit frame-sync pulse is detected.

Table 56. Register Bits Used to Set the Transmit Data Delay

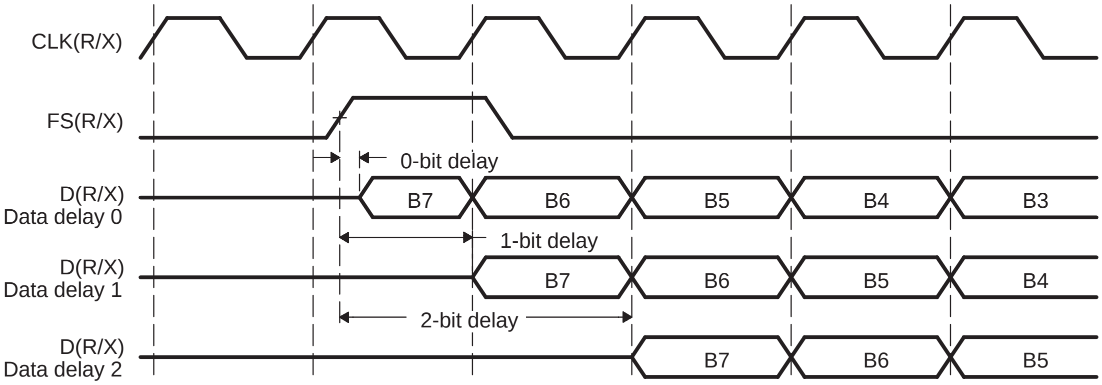
Register	Bit	Name	Function		Type	Reset
XCR2	1:0	XDATDLY	Transmitter data delay		R/W	00
			XDATDLY = 00	0-bit data delay		
			XDATDLY = 01	1-bit data delay		
			XDATDLY = 10	2-bit data delay		
			XDATDLY = 11	Reserved		

2.6.12.1 Data Delay

The start of a frame is the first clock cycle with active frame synchronization. The beginning of actual data reception or transmission with respect to the start of the frame can be delayed if necessary. This delay is called data delay.

XDATDLY specifies the data delay for transmission. The range of programmable data delay is zero to two bit-clocks (XDATDLY = 00b–10b), with CLK(R/X) = FS(R/X) = 0, as described in Table 56 and Figure 45. In this figure, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on. Typically a 1-bit delay is selected, because data often follows a 1-cycle active frame-synchronization pulse.

Figure 45. Range of Programmable Data Delay



2.6.12.2 0-Bit Data Delay

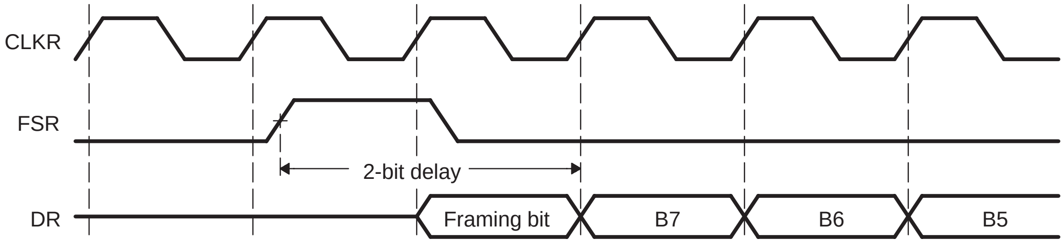
Normally, a frame-synchronization pulse is detected or sampled with respect to an edge of the serial clock internal CLK(R/X). Thus, on the following cycle or later (depending on the data delay value), data can be received or transmitted. However, in the case of 0-bit data delay, the data must be ready for reception and/or transmission on the same serial clock cycle.

In reception, receive data is sampled on the first falling edge of CLKR where an active-high internal FSR is detected, so the problem is solved. However, data transmission must begin on the rising edge of the internal CLKX clock that generated the frame synchronization. Therefore, the first data bit is assumed to be present in XSR1, and thus DX. The transmitter then asynchronously detects the frame synchronization, FSX, going active high and immediately starts driving the first bit to be transmitted on the DX pin.

2.6.12.3 2-Bit Data Delay

A data delay of two bit-periods allows the serial port to interface with devices where the data stream is preceded by a framing bit. During reception of such a stream with a data delay of two bits (framing bit appears after a 1-bit delay and data appears after a 2-bit delay), the serial port essentially discards the framing bit from the data stream, as shown in Figure 46. In this figure, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on.

Figure 46. 2-Bit Data Delay Used to Skip a Framing Bit



2.6.13 Set the Transmit DXENA Mode

The DXENA bit (see Table 57) controls the delay enabler on the DXZ pin. DXZ is used for multichannel mode. The DXENA mode is not supported on OMAP5912.

Table 57. Register Bit Used to Set the Transmit DXENA (DX Delay Enabler) Mode

Register	Bit	Name	Function		Type	Reset
SPCR1	7	DXENA	DXZ delay enabler mode		R/W	0
			DXENA = 0	DXZ delay enabler is off.		

2.6.13.1 DXENA Mode

The DXENA mode is not supported on OMAP5912, and hence data contention can be an issue when tying two or more McBSP DXZ pins together in multichannel mode.

To avoid contention, program the McBSP to transfer one extra (dummy) element or channel than necessary. Disable this additional (dummy) channel via the XCER registers, as it does not represent data of interest. A disabled channel has its DXZ in high-impedance state. The dummy channel will provide the necessary dead time between transfers from two McBSPs, and thus prevent contention.

Note that contention is not an issue when tying DR pins together, as the McBSPs do not drive this pin.

2.6.14 Set the Transmit Interrupt Mode

The transmitter interrupt (McBSP TX Interrupt, McBSP module signal XINT) signals the DSP core or MPU core of changes to the serial port status. Four options exist for configuring this interrupt. The options are set by the transmit interrupt mode bits, XINTM, in SPCR2.

Table 58. Register Bits Used to Set the Transmit Interrupt Mode

Register	Bit	Name	Function		Type	Reset
SPCR2	5:4	XINTM	Transmit interrupt mode		R/W	00
			XINTM = 00	McBSP TX interrupt (XINT) generated when XRDY changes from 0 to 1.		
			XINTM = 01	McBSP TX interrupt (XINT) generated by an end-of-block or end-of-frame condition in a transmit multichannel selection mode. In any of the transmit multichannel selection modes, interrupt after every 16-channel block boundary has been crossed within a frame and at the end of the frame. For details, see section 2.8.8. In any other serial transfer case, this setting is not applicable and, therefore, no interrupts are generated.		
			XINTM = 10	McBSP TX interrupt (XINT) generated by a new transmit frame-synchronization pulse. Interrupt on detection of each transmit frame-synchronization pulse. This generates an interrupt even when the transmitter is in its reset state. This is done by synchronizing the incoming frame-synchronization pulse to the internal clock source and sending it to the DSP core or MPU core via XINT.		
			XINTM = 11	McBSP TX interrupt (XINT) generated when XSYNCERR is set. Interrupt on frame-synchronization error. Regardless of the value of XINTM, XSYNCERR can be read to detect this condition. For more information on using XSYNCERR, see section 2.7.5.		

For more details on interrupt generation and mapping, see section 2.12.

2.6.15 Set the Transmit Frame-Synchronization Mode

The bits described in Table 59 determine the source for transmit frame synchronization and the FXR pin function.

Table 59. Register Bits Used to Set the Transmit Frame-Synchronization Mode

Register	Bit	Name	Function		Type	Reset
PCR	11	FSXM	Transmit frame-synchronization mode		R/W	0
			FSXM = 0	Transmit frame synchronization is supplied by an external source via the FSX pin.		
			FSXM = 1	Transmit frame synchronization is supplied by the McBSP, as determined by the FSGM bit in SRGR2.		
SRGR2	12	FSGM	Sample rate generator transmit frame-synchronization mode Used when FSXM = 1 in PCR.		R/W	0
			FSGM = 0	The McBSP generates a transmit frame-synchronization pulse when the content of DXR[1,2] is copied to XSR[1,2].		
			FSGM = 1	The transmitter uses frame-synchronization pulses generated by the sample rate generator. Program the FWID bits to set the width of each pulse. Program the FPER bits to set the frame-synchronization period.		

2.6.15.1 Transmit Frame-Synchronization Modes

Table 60 shows how FSXM and FSGM select the source of transmit frame-synchronization pulses. The three choices are:

- External frame-synchronization input
- Sample rate generator frame-synchronization signal (FSG)
- Internal signal that indicates a DXR-to-XSR copy has been made

Table 60 also shows the effect of each bit setting on the FSX pin. The FSXP bit determines the polarity of the signal on the FSX pin.

Table 60. Selecting the Source of Transmit Frame-Synchronization Pulses

FSXM	FSGM	Source of Transmit Frame Synchronization	FSX Pin Status
0	0 or 1	An external frame-synchronization signal enters the McBSP through the FSX pin. The signal is then inverted by FSXP before being used as internal FSX.	Input.
1	1	Internal FSX is driven by the sample rate generator frame-synchronization signal (FSG).	Output. FSG is inverted by FSXP before being driven out on FSX pin.
1	0	A DXR-to-XSR copy causes the McBSP to generate a transmit frame-synchronization pulse that is 1 cycle wide.	Output. The generated frame-synchronization pulse is inverted as determined by FSXP before being driven out on FSX pin.

2.6.15.2 Other Considerations

If the sample rate generator creates a frame-synchronization signal (FSG) that is derived from an external input clock, the GSYNC bit determines whether FSG is kept synchronized with pulses on the FSR pin. For more details, see section 2.1.3.

In the clock stop mode (CLKSTP = 10b or 11b), the McBSP can act as a master or as a slave in the SPI protocol. If the McBSP is a master and must provide a slave-enable signal ($\overline{\text{SS}}$) on the FSX pin, make sure that FSXM = 1 and FSGM = 0 so that FSX is an output and is driven active for the duration of each transmission. If the McBSP is a slave, make sure that FSXM = 0 so that the McBSP can receive the slave-enable signal on the FSX pin.

2.6.16 Set the Transmit Frame-Synchronization and Clock Polarity

Table 61. Register Bit Used to Set Transmit Frame-Synchronization Polarity

Register	Bit	Name	Function		Type	Reset
PCR	3	FSXP	Transmit frame-synchronization polarity		R/W	0
			FSXP = 0	Frame-synchronization pulse FSX is active high.		
			FSXP = 1	Frame-synchronization pulse FSX is active low.		

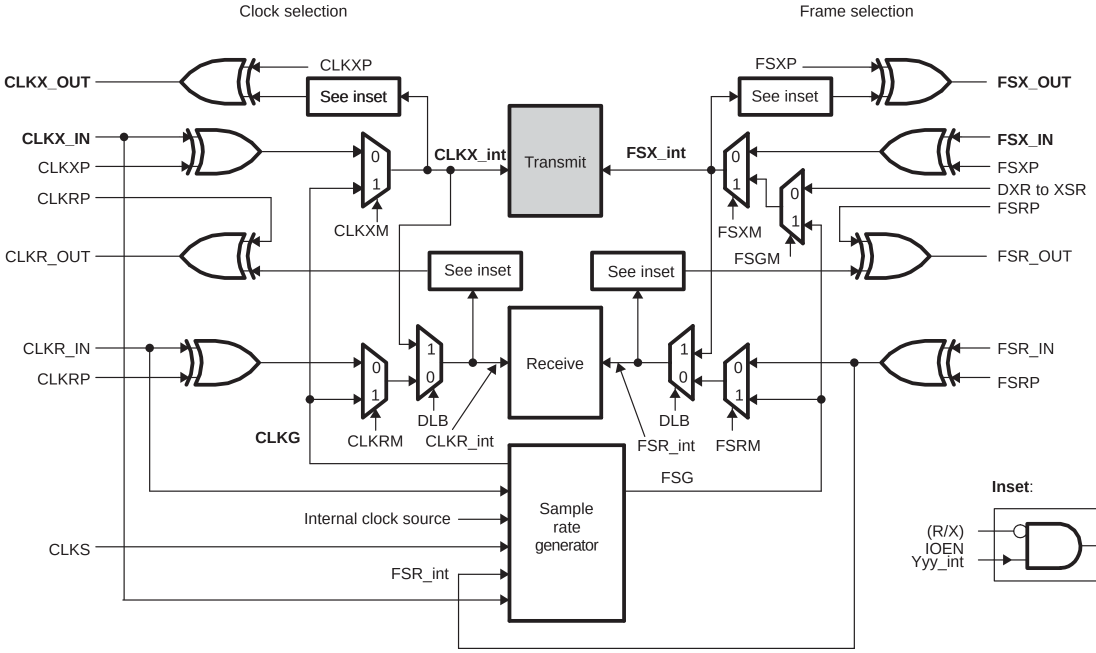
Table 62. Register Bit Used to Set Transmit Clock Polarity

Register	Bit	Name	Function		Type	Reset
PCR	1	CLKXP	Transmit clock polarity		R/W	0
			CLKXP = 0	Transmit data sampled on rising edge of CLKX.		
			CLKXP = 1	Transmit data sampled on falling edge of CLKX.		

2.6.16.1 Frame Synchronization Pulses, Clock Signals, and Their Polarities

Transmit frame-synchronization pulses can be generated internally by the sample rate generator (see section 2.1.2 and Figure 47) or driven by an external source. The source of frame synchronization is selected by programming the mode bit, FSXM, in PCR. FSX is also affected by the FSGM bit in SRGR2. For information about the effects of FSXM and FSGM, see section 2.6.15. Similarly, you can select whether transmit clocks are inputs or outputs by programming the mode bit, CLKXM, in PCR (see section 2.6.18). In Figure 47, all _int references represent the internal version of that signal; for example, CLKR_int represents the internal CLKR.

Figure 47. Transmit Clock and Frame-Synchronization Pulse Generation



When FSR and FSX are inputs (FSXM = FSRM = 0, external frame-synchronization pulses), the McBSP detects them on the internal falling edge of the internal CLKR and internal CLKX clocks, respectively. The receive data arriving at the DR pin is also sampled on the falling edge of internal CLKR. These internal clock signals are either derived from external source via the CLK(R/X) pins, or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When FSR and FSX are outputs, implying that they are driven by the sample rate generator, they are generated (transition to their active state) on the rising edge of the internal clock, CLK(R/X). Similarly, data on the DX pin is output on the rising edge of internal CLKX.

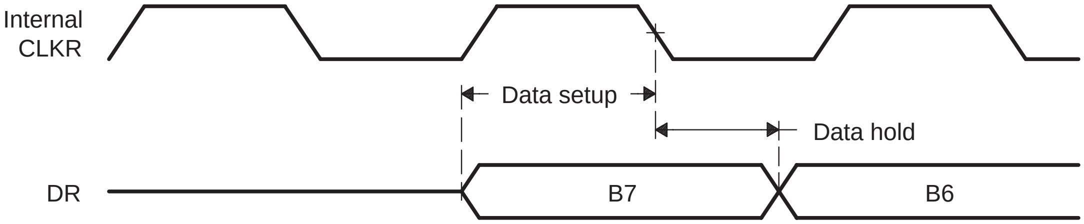
FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame-synchronization signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame synchronization (FSR/FSX are inputs to McBSP) and FSRP = FSXP = 1, the external active-low frame-synchronization signals are inverted before being sent to the receiver (internal FSR) and transmitter (internal FSX). Similarly, if internal synchronization is selected (FSR/FSX are output pins and GSYNC = 0) and the polarity bit FS(R/X)P = 1, the internal active-high frame-synchronization signals are inverted before being sent to the FS(R/X) pin.

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Data is always transmitted on the rising edge of internal CLKX. If CLKXP = 1 and external clocking is selected (CLKXM = 0 and CLKX is an input), the external falling-edge triggered input clock on CLKX is inverted to a rising-edge triggered clock before being sent to the transmitter. The external falling edge triggered clock will be seen as an internal rising edge triggered clock. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and CLKX is an output pin), the internal (rising-edge triggered) clock, internal CLKX, is inverted before being sent out on the CLKX pin.

Similarly, the receiver can reliably sample data that is clocked with a rising edge clock by the transmitter. The receive clock polarity bit, CLKRP, sets the edge used to sample received data. The receive data is always sampled on the falling edge of internal CLKR. Therefore, if CLKRP = 1, and external clocking is selected (CLKRM = 0 and CLKR is an input pin), the external rising-edge triggered input clock on CLKR is inverted to a falling-edge triggered clock before being sent to the receiver. If CLKRP = 1, and internal clocking is selected (CLKRM = 1), the internal falling-edge triggered clock is inverted to a rising-edge triggered clock before being sent out on the CLKR pin.

CLKRP = CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge from the transmitter to ensure valid setup and hold of data around this edge. Figure 48 shows how data clocked by an external serial device using a rising edge can be sampled by the McBSP receiver on the falling edge of the same clock.

Figure 48. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



2.6.17 Set the SRG Frame-Synchronization Period and Pulse Width

Table 63. Register Bits Used to Set SRG Frame-Synchronization Period and Pulse Width

Register	Bit	Name	Function	Type	Reset
SRGR2	11:0	FPER	Sample rate generator frame-synchronization period For the frame-synchronization signal FSG, (FPER + 1) determines the period from the start of a frame-synchronization pulse to the start of the next frame-synchronization pulse. Range for (FPER + 1): 1 to 4096 CLKG cycles.	R/W	0000 0000 0000
SRGR1	15:8	FWID	Sample rate generator frame-synchronization pulse width This field plus 1 determines the width of each frame-synchronization pulse on FSG. Range for (FWID + 1): 1 to 256 CLKG cycles.	R/W	0000 0000

2.6.17.1 Frame-Synchronization Period and Frame-Synchronization Pulse Width

The sample rate generator can produce a clock signal, CLKG, and a frame-synchronization signal, FSG. If the sample rate generator is supplying receive or transmit frame synchronization, you must program FPER and FWID.

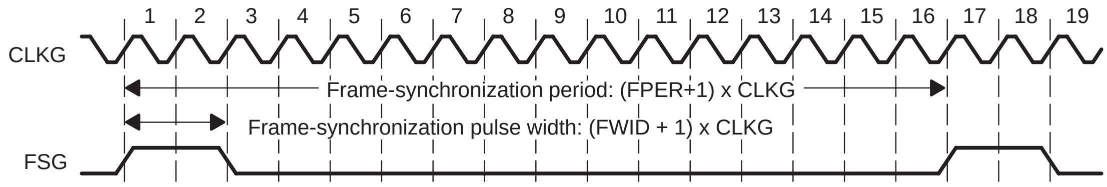
On FSG, the period from the start of a frame-synchronization pulse to the start of the next pulse is (FPER + 1) CLKG cycles. The 12 bits in FPER allow a frame-synchronization period of 1 to 4096 CLKG cycles, permitting up to 4096 data bits per frame. When GSYNC = 1, FPER is a don’t care value.

Each pulse on FSG has a width of (FWID + 1) CLKG cycles. The eight bits in FWID allow a pulse width of 1 to 256 CLKG cycles. Program FWID to a value less than the programmed word length.

The values in FPER and FWID are loaded into separate down-counters. The 12-bit FPER counter counts down the generated clock cycles from the programmed value (4095 maximum) to 0. The 8-bit FWID counter counts down from the programmed value (255 maximum) to 0.

Figure 49 shows a frame-synchronization period of 16 CLKG periods (FPER = 15 or 00001111b) and a frame-synchronization pulse with an active width of 2 CLKG periods (FWID = 1).

Figure 49. Frame of Period 16 CLKG Periods and Active Width of 2 CLKG Periods


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
CLKG
Frame-synchronization period: (FPER+1) x CLKG
Frame-synchronization pulse width: (FWID + 1) x CLKG
FSG

When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when $\overline{\text{GRST}}$ = 1 and FSGM = 1, a frame-synchronization pulse is generated. The frame width value (FWID + 1) is counted down on every CLKG cycle until it reaches 0, at which time FSG goes low. At the same time, the frame period value (FPER + 1) is also counting down. When this value reaches 0, FSG goes high, indicating a new frame.

2.6.18 Set the Transmit Clock Mode

The CLKXM bit, described in Table 64, determines the source for the transmit clock and the function of the CLKX pin.

Table 64. Register Bit Used to Set the Transmit Clock Mode

Register	Bit	Name	Function		Type	Reset
PCR	9	CLKXM	Transmit clock mode		R/W	0
			CLKXM = 0	The transmitter gets its clock signal from an external source via the CLKX pin.		
			CLKXM = 1	The CLKX pin is an output pin driven by the sample rate generator of the McBSP.		

2.6.18.1 Selecting a Source for the Transmit Clock and a Data Direction for the CLKX Pin

Table 65 shows how the CLKXM bit selects the transmit clock and the corresponding status of the CLKX pin. The polarity of the signal on the CLKX pin is determined by the CLKXP bit.

Table 65. How the CLKXM Bit Selects the Transmit Clock and the Corresponding Status of the CLKX Pin

CLKXM in PCR	Source of Transmit Clock	CLKX Pin Status
0	Internal CLKX is driven by an external clock on the CLKX pin. CLKX is inverted as determined by CLKXP before being used.	Input.
1	Internal CLKX is driven by the sample rate generator clock, CLKG.	Output. CLKG, inverted as determined by CLKXP, is driven out on CLKX.

2.6.18.2 Other Considerations

If the sample rate generator creates a clock signal (CLKG) that is derived from an external input clock, the GSYNC bit determines whether CLKG is kept synchronized with pulses on the FSR pin. For more details, see section 2.1.3.

In the clock stop mode (CLKSTP = 10b or 11b), the McBSP can act as a master or as a slave in the SPI protocol. If the McBSP is a master, ensure that CLKXM = 1, so that CLKX is an output that supplies the master clock to any slave devices. If the McBSP is a slave, make sure that CLKXM = 0, so that CLKX is an input to accept the master clock signal.

2.6.19 Set the SRG Clock Divide-Down Value

The CLKGDV field described in Table 66, is used to set the sample rate generator clock divide-down value.

Table 66. Register Bits Used to Set Sample Rate Generator (SRG) Clock Divide-Down Value

Register	Bit	Name	Function	Type	Reset
SRGR1	7:0	CLKGDV	Sample rate generator clock divide-down value The input clock of the sample rate generator is divided by (CLKGDV + 1) to generate the required sample rate generator clock frequency. The default value of CLKGDV is 1 (divide input clock by 2).	R/W	0000 0001

2.6.19.1 Sample Rate Generator Clock Divider

The first divider stage generates the serial data bit clock from the input clock. This divider stage utilizes a counter, preloaded by CLKGDV, that contains the divide ratio value.

The output of the first divider stage is the data bit clock, which is output as CLKG and which serves as the input for the second and third stages of the divider.

CLKG has a frequency equal to 1/(CLKGDV + 1) of the sample rate generator input clock. Thus, the sample generator input clock frequency is divided by a value between 1 and 256. When CLKGDV is odd or equal to 0, the CLKG duty cycle is 50%. When CLKGDV is an even value, 2p, representing an odd divide-down, the high-state duration is p+1 cycles and the low-state duration is p cycles.

2.6.20 Set the SRG Clock Synchronization Mode

The GSYNC bit (see Table 67) determines the SRG clock synchronization mode.

Table 67. Register Bit Used to Set the SRG Clock Synchronization Mode

Register	Bit	Name	Function	Type	Reset
SRGR2	15	GSYNC	Sample rate generator clock synchronization GSYNC is used only when the input clock source for the sample rate generator is external- on the CLKS (McBSP1), CLKR (McBSP2), or CLKX pin. GSYNC = 0 The sample rate generator clock (CLKG) is free running. CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles. GSYNC = 1 Clock synchronization is performed. When a pulse is detected on the FSR pin: ❑ CLKG is adjusted so that it is synchronized with the input clock on the CLKS (McBSP1), CLKR (McBSP2), or CLKX pin. ❑ FSG pulses. FSG only pulses in response to a pulse on the FSR pin. The frame-synchronization period defined in FPER is ignored.	R/W	0

For more details on using the clock synchronization feature, see section 2.1.3.

2.6.21 Set the SRG Clock Mode (Choose an Input Clock)

The bits described in Table 68 are used to select the source for the SRG clock. Note that McBSP1 features the CLKS pin and McBSP2 features the CLKR pin only.

Table 68. Register Bits Used to Set the SRG Clock Mode (Choose an Input Clock)

Register	Bit	Name	Function		Type	Reset
PCR SRGR2	7 13	SCLKME CLKSM	Sample rate generator clock mode		R/W R/W	0 1
			SCLKME = 0 CLKSM = 0	Sample rate generator clock derived from CLKS (McBSP1) pin.		
			SCLKME = 0 CLKSM = 1	Sample rate generator clock derived from internal clock source. (This is the condition forced by an OMAP5912 reset.)		
			SCLKME = 1 CLKSM = 0	Sample rate generator clock derived from CLKR (McBSP2) pin.		
			SCLKME = 1 CLKSM = 1	Sample rate generator clock derived from CLKX pin.		

2.6.21.1 SRG Clock Mode

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both, but CLKG is derived from an input clock. Table 68 shows the four possible sources of the input clock. For more details on generating CLKG, see section 2.1.1.

2.6.22 Set the SRG Input Clock Polarity

Table 69. Register Bits Used to Set the SRG Input Clock Polarity

Register	Bit	Name	Function	Type	Reset
SRGR2	14	CLKSP	CLKS pin polarity. CLKSP determines the input clock polarity when the CLKS pin supplies the input clock (SCLKME = 0 and CLKSM = 0). CLKSP = 0 Rising edge on CLKS pin generates CLKG and FSG. CLKSP = 1 Falling edge on CLKS pin generates CLKG and FSG.	R/W	0
PCR	1	CLKXP	CLKX pin polarity. CLKXP determines the input clock polarity when the CLKX pin supplies the input clock (SCLKME = 1 and CLKSM = 1). CLKXP = 0 Rising edge on CLKX pin generates transitions on CLKG and FSG. CLKXP = 1 Falling edge on CLKX pin generates transitions on CLKG and FSG.	R/W	0
PCR	0	CLKRP	CLKR pin polarity. CLKRP determines the input clock polarity when the CLKR pin supplies the input clock (SCLKME = 1 and CLKSM = 0). CLKRP = 0 Falling edge on CLKR pin generates transitions on CLKG and FSG. CLKRP = 1 Rising edge on CLKR pin generates transitions on CLKG and FSG.	R/W	0

Note that the polarity mode is performed at the signal input/output. The internal clock signal (CLKG) polarity remains the same regardless of the input clock polarity. See Figure 6 and Figure 7 for more details on the effect of this feature.

2.6.22.1 Using CLKSP/CLKXP/CLKRP to Choose an Input Clock Polarity

The sample rate generator can produce a clock signal (CLKG) and a frame-synchronization signal (FSG) for use by the receiver, the transmitter, or both. To produce CLKG and FSG, the sample rate generator must be driven by an input clock signal derived from the internal clock source or from an external clock on the CLKS, CLKX, or CLKR pin. If you use a pin, choose a polarity for that pin by using the appropriate polarity bit (CLKSP for the CLKS pin, CLKXP for the CLKX pin, CLKRP for the CLKR pin). The polarity determines whether the rising or falling edge of the input clock generates transitions on CLKG and FSG.

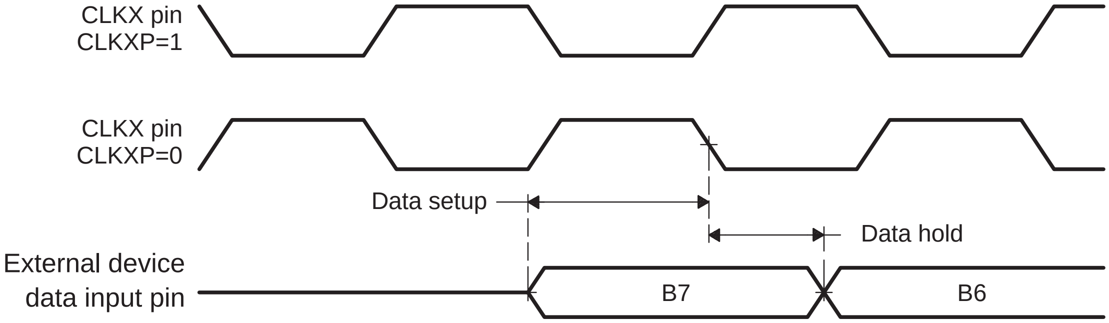
CLKXP

- ❑ If CLKXP=0, transmit data is written on the rising edge of CLKX and the external device reads the data on the falling edge CLKX.
- ❑ If CLKXP=1, transmit data is written on the falling edge of CLKX and the external device reads the data on the rising edge of CLKX.

CLKRP

- ❑ If CLKRP=0, received data is sampled on the falling edge of CLKR.
- ❑ If CLKXR=1, received data is sampled on the rising edge of CLKR.

Figure 50. Example: Transmit Clock Polarity



2.7 McBSP Exception/Error Conditions

There are five serial port events that can constitute a system error:

- ❑ Receiver overrun (RFULL = 1)

  This occurs when DRR1 has not been read since the last RBR-to-DRR copy. Consequently, the receiver does not copy a new word from the RBR(s) to the DRR(s), and the RSR(s) are now full with another new word shifted in from DR. Therefore, RFULL = 1 indicates an error condition wherein any new data that can arrive at this time on DR replaces the contents of the RSR(s), and the previous word is lost. The RSRs continue to be overwritten as long as new data arrives on DR and DRR1 is not read. McBSP2 features the RX overflow MPU interrupt upon a RFULL = 1 condition. For more details about overrun in the receiver, see section 2.7.1.

❑ Unexpected receive frame-synchronization pulse (RSYNCERR = 1)

  This occurs during reception when RFIG = 0 and an unexpected frame-synchronization pulse occurs. An unexpected frame-synchronization pulse is one that begins the next frame transfer before all the bits of the current frame have been received. Such a pulse causes data reception to abort and restart. If new data has been copied into the RBR(s) from the RSR(s) since the last RBR-to-DRR copy, this new data in the RBR(s) is lost. This is because no RBR-to-DRR copy occurs; the reception has been restarted. For more details about receive frame-synchronization errors, see section 2.7.2.

❑ Transmitter data overwrite

  This occurs when the DSP core, MPU core, or DSP/system DMA controller overwrites data in the DXR(s) before the data is copied to the XSR(s). The overwritten data never reaches the DX pin. For more details about overwrite in the transmitter, see section 2.7.3.

❑ Transmitter underflow ($\overline{\text{XEMPTY}}$ = 0)

  If a new frame-synchronization signal arrives before new data is loaded into DXR1, the previous data in the DXR(s) is sent again. This procedure continues for every new frame-synchronization pulse that arrives until DXR1 is loaded with new data. For more details about underflow in the transmitter, see section 2.7.4.

❑ Unexpected transmit frame-synchronization pulse (XSYNCERR = 1)

  This occurs during transmission when XFIG = 0 and an unexpected frame-synchronization pulse occurs. An unexpected frame-synchronization pulse is one that begins the next frame transfer before all the bits of the current frame have been transferred. Such a pulse causes the current data transmission to abort and restart if XFIG = 0. If new data has been written to the DXR(s) since the last DXR-to-XSR copy, the current value in the XSR(s) is lost. For more details about transmit frame-synchronization errors, see section 2.7.5.

2.7.1 Overrun in the Receiver

RFULL = 1 in SPCR1 indicates that the receiver has experienced overrun and is in an error condition. RFULL is set when all of the following conditions are met:

❑ DRR1 has not been read since the last RBR-to-DRR copy (RRDY = 1).

❑ RBR1 is full and an RBR-to-DRR copy has not occurred.

❑ RSR1 is full and an RSR1-to-RBR copy has not occurred.

As described in the section 2.2.5, data arriving on DR is continuously shifted into RSR1 (for word length of 16 bits or smaller) or RSR2 and RSR1 (for word length larger than 16 bits). Once a complete word is shifted into the RSR(s), an RSR-to-RBR copy can occur only if the previous data in RBR1 has been copied to DRR1. The RRDY bit is set when new data arrives in DRR1 and is cleared when that data is read from DRR1. Until RRDY = 0, the next RBR-to-DRR copy does not take place, and the data is held in the RSR(s). New data arriving on the DR pin is shifted into RSR(s), and the previous content of the RSR(s) is lost.

You can prevent the loss of data if DRR1 is read no later than 2.5 CLKR cycles before the end of the third word is shifted into the RSR1.

Note:

If both DRRs are needed (word length larger than 16 bits), the DSP core, MPU core, or DSP/system DMA controller must read from DRR2 first and then from DRR1. As soon as DRR1 is read, the next RBR-to-DRR copy is initiated. If DRR2 is not read first, the data in DRR2 is lost.

McBSP2 is capable of interrupting the MPU when an overrun condition occurs on the receiver. See section 2.12.2.2 for more interrupt mapping information.

After the receiver starts running from reset, a minimum of three words must be received before RFULL is set. Either of the following events clears the RFULL bit and allows subsequent transfers to be read properly:

❑ The DSP core, MPU core, or DSP/system DMA controller reads DRR1.

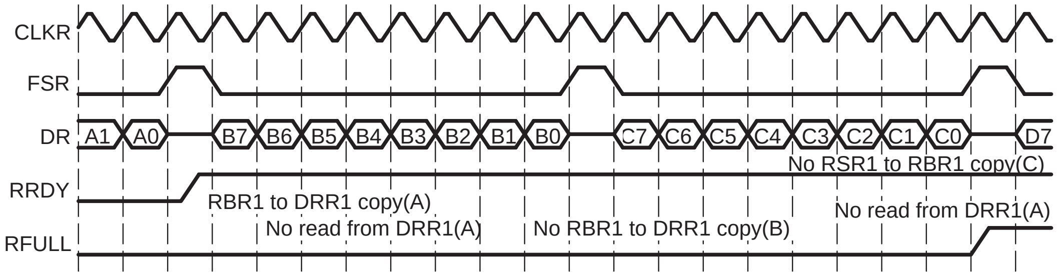
❑ The receiver is reset individually ($\overline{\text{RRST}}$ = 0) or as part of an OMAP5912 reset.

Another frame-synchronization pulse is required to restart the receiver.

2.7.1.1 Example of Overrun Condition

Figure 51 shows the receive overrun condition. Because serial word A is not read from DRR1 before serial word B arrives in RBR1, B is not transferred to DRR1 yet. Another new word (C) arrives and RSR1 is full with this data. DRR1 is finally read, but not earlier than 2.5 cycles before the end of word C. Therefore, new data (D) overwrites word C in RSR1. If DRR1 is not read in time, the next word can overwrite D.

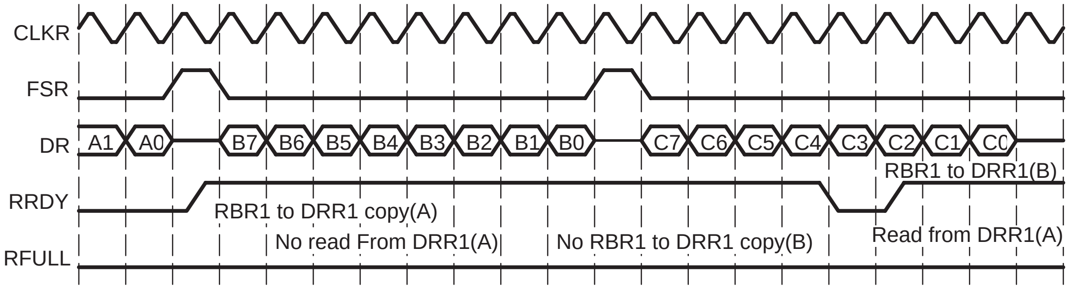
Figure 51. Overrun in the McBSP Receiver



2.7.1.2 Example of Preventing Overrun Condition

Figure 52 shows the case where RFULL is not set, and the overrun condition is prevented by a read from DRR1 at least 2.5 cycles before the next serial word (C) is completely shifted into RSR1. This ensures that an RBR1-to-DRR1 copy of word B occurs before receiver attempts to transfer word C from RSR1 to RBR1.

Figure 52. Overrun Prevented in the McBSP Receiver



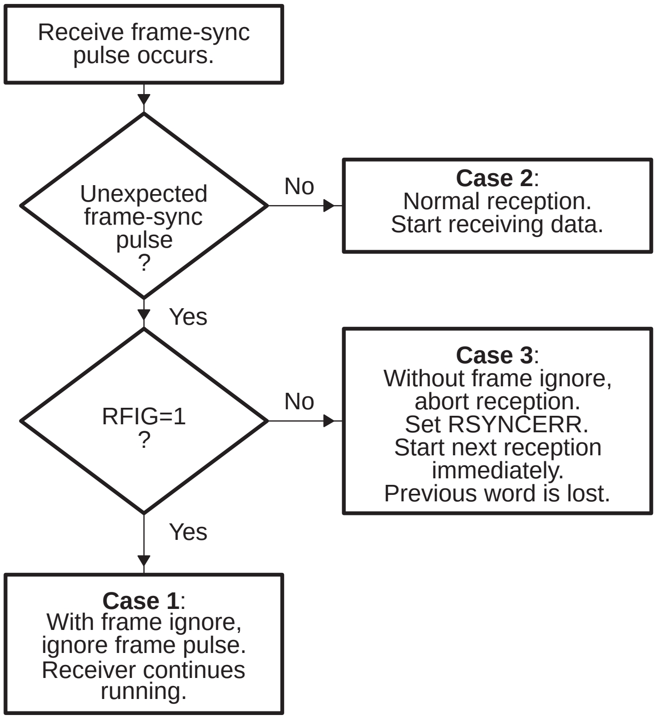
2.7.2 Unexpected Receive Frame-Synchronization Pulse

Section 2.7.2.1 shows how the McBSP responds to any receive frame-synchronization pulses, including an unexpected pulse. Sections 2.7.2.2 and 2.7.2.3 show an example of a frame-synchronization error and an example of how to prevent such an error, respectively.

2.7.2.1 Possible Responses to Receive Frame-Synchronization Pulses

Figure 53 shows the decision tree that the receiver uses to handle all incoming frame-synchronization pulses. The figure assumes that the receiver has been started ($\overline{\text{RRST}}$ = 1 in SPCR1). Case 3 in the figure is the case in which an error occurs.

Figure 53. Possible Responses to Receive Frame-Synchronization Pulses


Receive frame-sync pulse occurs.
Unexpected frame-sync pulse ?
No
Case 2: Normal reception. Start receiving data.
Yes
RFIG=1 ?
No
Case 3: Without frame ignore, abort reception. Set RSYNCERR. Start next reception immediately. Previous word is lost.
Yes
Case 1: With frame ignore, ignore frame pulse. Receiver continues running.

Any one of three cases can occur:

- ❏ Case 1: Unexpected internal FSR pulses with RFIG = 1 in RCR2. Receive frame-synchronization pulses are ignored, and the reception continues.
- ❏ Case 2: Normal serial port reception. Reception continues normally because the frame-synchronization pulse is not unexpected. There are three possible reasons why a receive operation might not be in progress when the pulse occurs:
  - ■ The FSR pulse is the first after the receiver is enabled ($\overline{\text{RRST}}$ = 1 in SPCR1).
  - ■ The FSR pulse is the first after DRR[1,2] is read, clearing a receiver full (RFULL = 1 in SPCR1) condition.
  - ■ The serial port is between data frames. The programmed data delay for reception (programmed with the RDATDLY bits in RCR2) may start during these interpacket intervals for the first bit of the next word to be received. Thus, at maximum frame frequency, frame synchronization can still be received 0 to 2 clock cycles before the first bit of the synchronized frame.

❑ Case 3: Unexpected receive frame synchronization with RFIG = 0, frame-synchronization pulses are not ignored. Unexpected frame-synchronization pulses can originate from an external source or from the internal sample rate generator when the frame period is shorter than the frame length.

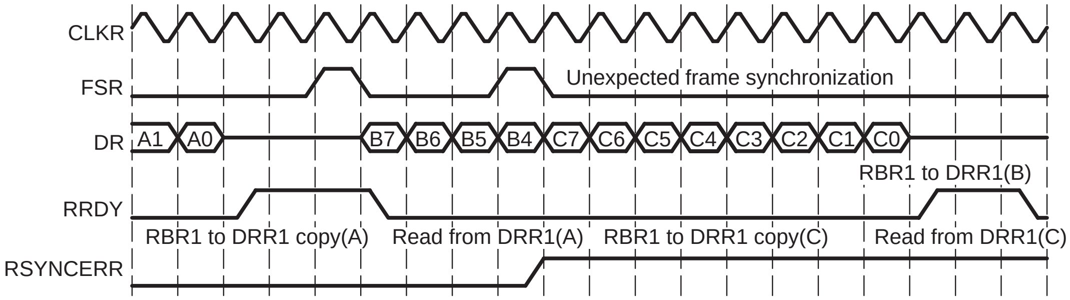
If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully received, this pulse is treated as an unexpected frame-synchronization pulse, and the receiver sets the receive frame-synchronization error bit (RSYNCERR) in SPCR1. RSYNCERR can be cleared only by a receiver reset or by a write of 0 to this bit.

If you want the McBSP to notify the DSP core or MPU core of receive frame-synchronization errors, you can set a special receive interrupt mode with the RINTM bits in SPCR1. When RINTM = 11b, the McBSP sends a McBSP RX Interrupt (internal McBSP signal RINT) request to the DSP core or MPU core each time that RSYNCERR is set.

2.7.2.2 Example of Unexpected Receive Frame-Synchronization Pulse

Figure 54 shows an unexpected receive frame-synchronization pulse during normal operation of the serial port, with RFIG = 0 and time intervals between data packets. When the unexpected frame-synchronization pulse occurs, the RSYNCERR bit is set, the reception of data B is aborted, and the reception of data C begins. In addition, if RINTM = 11b, the McBSP sends a receive interrupt (McBSP RX Interrupt, McBSP module signal RINT) request to the DSP core or MPU core.

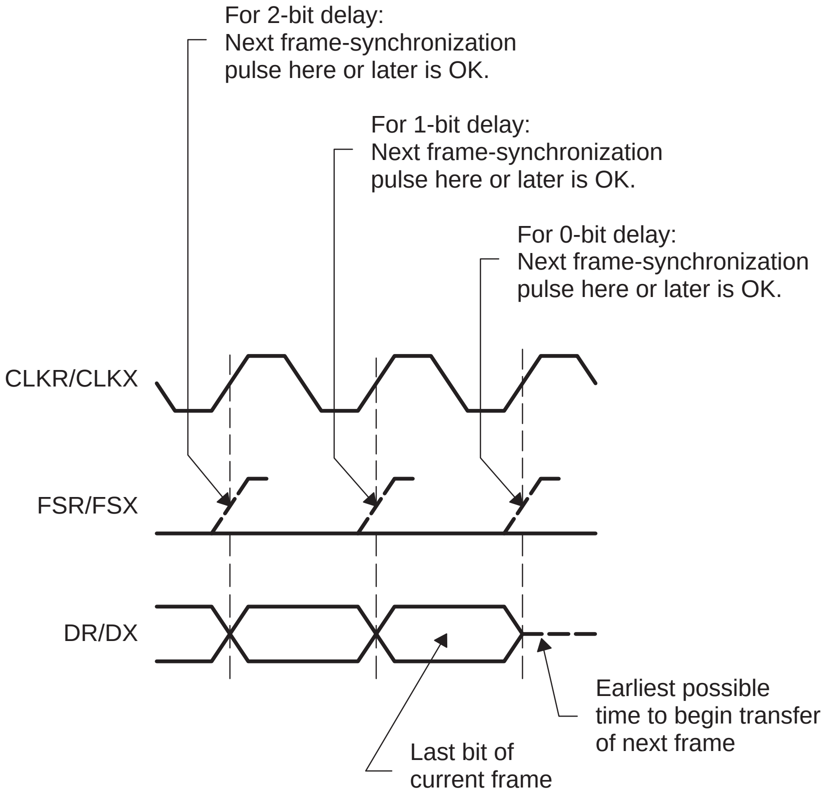
Figure 54. Unexpected Frame-Synchronization Pulse During a McBSP Reception



2.7.2.3 Preventing Unexpected Receive Frame-Synchronization Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKR cycles, depending on the value in the RDATDLY bits in RCR2. For each possible data delay, Figure 55 shows when a new frame-synchronization pulse on FSR can safely occur relative to the last bit of the current frame.

Figure 55. Proper Positioning of Frame-Synchronization Pulses


For 2-bit delay:
Next frame-synchronization
pulse here or later is OK.
For 1-bit delay:
Next frame-synchronization
pulse here or later is OK.
For 0-bit delay:
Next frame-synchronization
pulse here or later is OK.
CLKR/CLKX
FSR/FSX
DR/DX
Earliest possible
time to begin transfer
of next frame
Last bit of
current frame

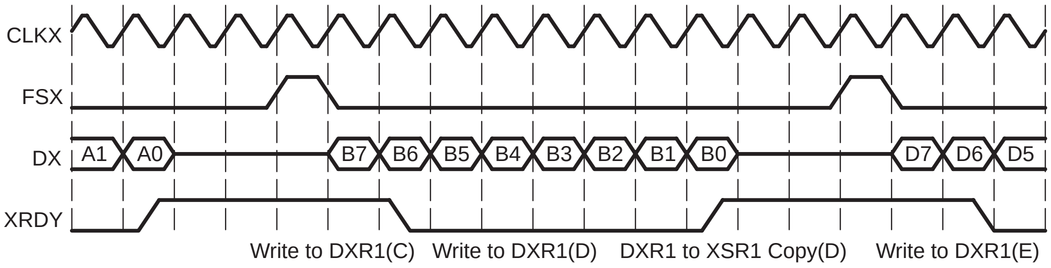
2.7.3 Overwrite in the Transmitter

As described in section 2.2.6 on McBSP transmission, the transmitter must copy the data previously written to the DXR(s) by the DSP core, MPU core, or DSP/system DMA controller into the XSR(s) and then shift each bit from the XSR(s) to the DX pin. If new data is written to the DXR(s) before the previous data is copied to the XSR(s), the previous data in the DXR(s) is overwritten and thus lost.

2.7.3.1 Example of Overwrite Condition

Figure 56 shows what happens if the data in DXR1 is overwritten before being transmitted. Initially, DXR1 is loaded with data C. A subsequent write to DXR1 overwrites C with D before C is copied to XSR1. Thus, C is never transmitted on DX.

Figure 56. Data in the McBSP Transmitter Overwritten and Not Transmitted



2.7.3.2 Preventing Overwrites

You can prevent DSP core or MPU core overwrites by making the DSP core/MPU core:

- Poll for XRDY = 1 in SPCR2 before writing to the DXR(s). XRDY is set when data is copied from DXR1 to XSR1 and is cleared when new data is written to DXR1.
- Wait for a transmit interrupt (McBSP TX Interrupt, McBSP module signal XINT) before writing to the DXR(s). When XINTM = 00b in SPCR2, the transmitter sends the McBSP TX Interrupt to the DSP core or MPU core each time XRDY is set.

You can prevent DSP/system DMA overwrites by synchronizing DSP/system DMA transfers to the transmit synchronization event through the McBSP TX DSP/system DMA request line (internal McBSP signal XEVT). The transmitter sends a request signal to the DSP/system DMA each time XRDY is set.

For more details on DSP/system DMA synchronization event request lines, see section 2.13.

2.7.4 Underflow in the Transmitter

The McBSP indicates a transmitter empty (or underflow) condition by clearing the $\overline{\text{XEMPTY}}$ bit in SPCR2. Either of the following events activates $\overline{\text{XEMPTY}}$ ($\overline{\text{XEMPTY}}$ = 0):

- DXR1 has not been loaded since the last DXR-to-XSR copy, and all bits of the data word in the XSR(s) have been shifted out on the DX pin.
- The transmitter is reset (by forcing $\overline{\text{XRST}}$ = 0 in SPCR2, or by an OMAP5912 reset) and is then restarted.

In the underflow condition, the transmitter continues to transmit the old data that is in the DXR(s) for every new transmit frame-synchronization signal until a new value is loaded into DXR1 by the DSP core, MPU core, or DSP/system DMA controller.

Note:

If both DXRs are needed (word length larger than 16 bits), the DSP core, MPU core, or DSP/system DMA controller must load DXR2 first and then load DXR1. As soon as DXR1 is loaded, the contents of both DXRs are copied to the transmit shift registers (XSRs). If DXR2 is not loaded first, the previous content of DXR2 is passed to the XSR2.

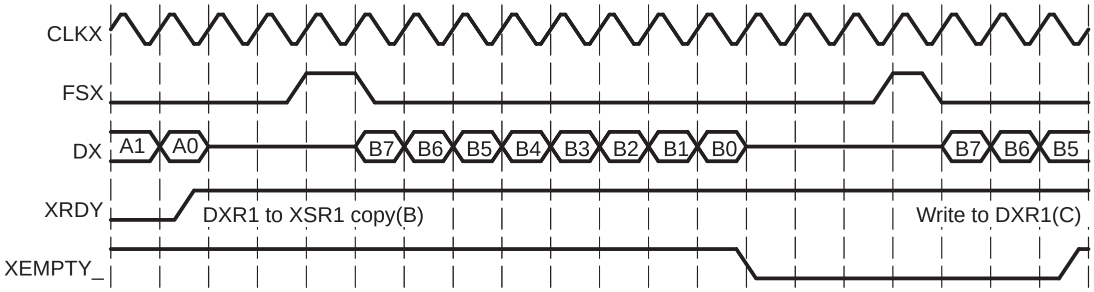
$\overline{\text{XEMPTY}}$ is deactivated ($\overline{\text{XEMPTY}}$ = 1) when a new word in DXR1 is transferred to XSR1. If FSXM = 1 in PCR and FSGM = 0 in SRGR2, the transmitter generates a single internal FSX pulse in response to a DXR-to-XSR copy. Otherwise, the transmitter waits for the next frame-synchronization pulse before sending out the next frame on DX.

When the transmitter is taken out of reset ($\overline{\text{XRST}}$ = 1), it is in a transmitter ready (XRDY = 1 in SPCR2) and transmitter empty ($\overline{\text{XEMPTY}}$ = 0) state. If DXR1 is loaded by the DSP core, MPU core, or DSP/system DMA controller before internal FSX goes active high, a valid DXR-to-XSR transfer occurs. This allows for the first word of the first frame to be valid even before the transmit frame-synchronization pulse is generated or detected. Alternatively, if a transmit frame-synchronization pulse is detected before DXR1 is loaded, zeros are output on DX.

2.7.4.1 Example of the Underflow Condition

Figure 57 shows an underflow condition. After B is transmitted, DXR1 is not reloaded before the subsequent frame-synchronization pulse. Thus, B is again transmitted on DX.

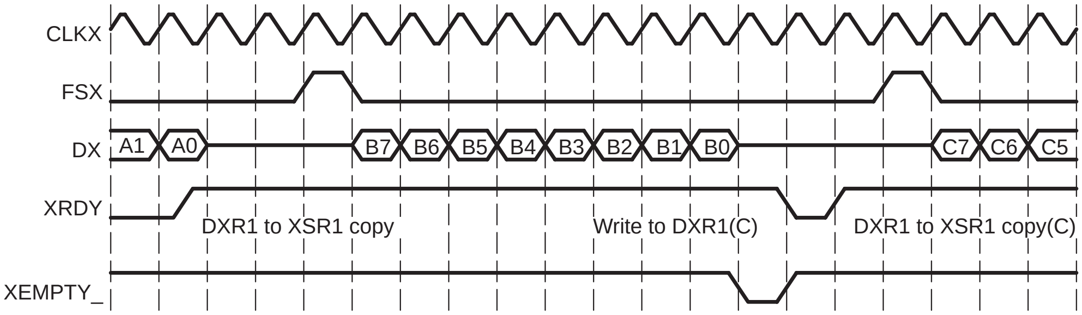
Figure 57. Underflow During McBSP Transmission



2.7.4.2 Example of Preventing Underflow Condition

Figure 58 shows the case of writing to DXR1 just before an underflow condition would otherwise occur. After B is transmitted, C is written to DXR1 before the next frame-synchronization pulse. As a result, there is no underflow; B is not transmitted twice.

Figure 58. Underflow Prevented in the McBSP Transmitter



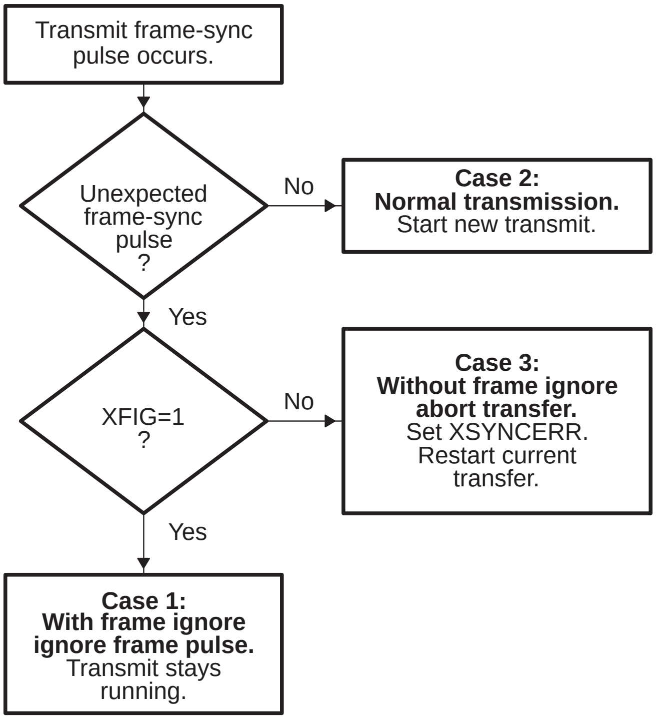
2.7.5 Unexpected Transmit Frame-Synchronization Pulse

Section 2.7.5.1 shows how the McBSP responds to any transmit frame-synchronization pulses, including an unexpected pulse. Sections 2.7.5.2 and 2.7.5.3 show an example of a frame-synchronization error and an example of how to prevent such an error, respectively.

2.7.5.1 Possible Responses to Transmit Frame-Synchronization Pulses

Figure 59 shows the decision tree that the transmitter uses to handle all incoming frame-synchronization pulses. The figure assumes that the transmitter has been started ($\overline{\text{XRST}}$ = 1 in SPCR2). Case 3 represents the case in which an error occurs.

Figure 59. Possible Responses to Transmit Frame-Synchronization Pulses


Transmit frame-sync pulse occurs.
Unexpected frame-sync pulse ?
No
Case 2: Normal transmission. Start new transmit.
Yes
XFIG=1 ?
No
Case 3: Without frame ignore abort transfer. Set XSYNCERR. Restart current transfer.
Yes
Case 1: With frame ignore ignore frame pulse. Transmit stays running.

Any one of three cases can occur:

❑ Case 1: Unexpected internal FSX pulses with XFIG = 1 in XCR2. Transmit frame-synchronization pulses are ignored, and the transmission continues.

❑ Case 2: Normal serial port transmission. Transmission continues normally because the frame-synchronization pulse is not unexpected. There are two possible reasons why a transmit operations might not be in progress when the pulse occurs:

  - This FSX pulse is the first after the transmitter is enabled ($\overline{\text{XRST}}$ = 1).
  - The serial port is between data frames. The programmed data delay for transmission (programmed with the XDATDLY bits in XCR2) may start during these interpacket intervals before the first bit of the previous word is transmitted. Thus, at maximum packet frequency, frame synchronization can still be received 0 to 2 clock cycles before the first bit of the synchronized frame.

❑ Case 3: Unexpected transmit frame synchronization with XFIG = 0 (frame-synchronization pulses not ignored). Unexpected frame-synchronization pulses can originate from an external source or from the internal sample rate generator.

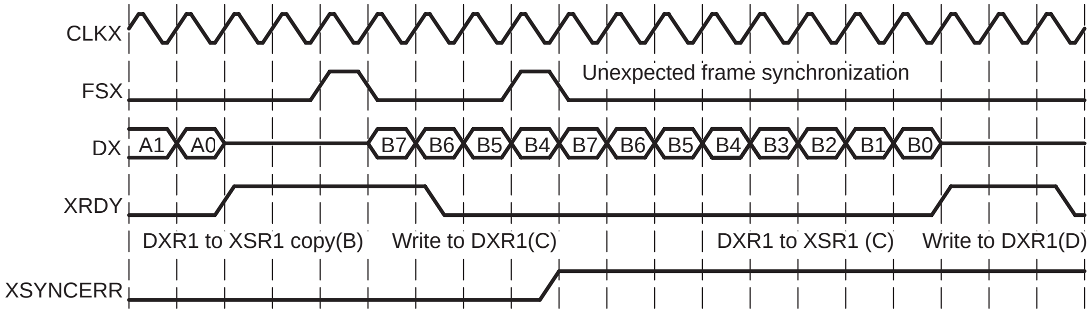
If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame-synchronization pulse, and the transmitter sets the transmit frame-synchronization error bit (XSYNCERR) in SPCR2. XSYNCERR can be cleared only by a transmitter reset or by a write of 0 to this bit.

If you want the McBSP to notify the DSP core or MPU core of frame-synchronization errors, you can set a special transmit interrupt mode with the XINTM bits in SPCR2. When XINTM = 11b, the McBSP sends a transmit interrupt (McBSP TX Interrupt, McBSP module signal XINT) request to the DSP core or MPU core each time that XSYNCERR is set.

2.7.5.2 Example of Unexpected Transmit Frame-Synchronization Pulse

Figure 60 shows an unexpected transmit frame-synchronization pulse during normal operation of the serial port with intervals between the data packets. When the unexpected frame-synchronization pulse occurs, the XSYNCERR bit is set and the transmission of data B is restarted because no new data has been passed to XSR1 yet. In addition, if XINTM = 11b, the McBSP sends a transmit interrupt (McBSP TX Interrupt, McBSP module signal XINT) request to the DSP core or MPU core.

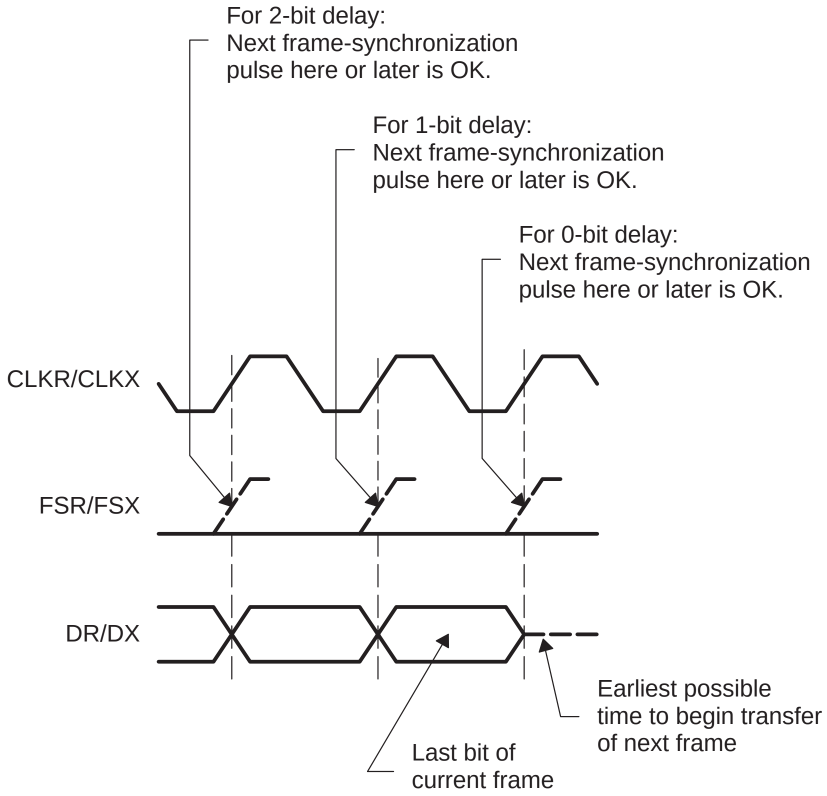
Figure 60. An Unexpected Frame-Synchronization Pulse During a McBSP Transmission



2.7.5.3 Preventing Unexpected Transmit Frame-Synchronization Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKX cycles, depending on the value in the XDATDLY bits in XCR2. For each possible data delay, Figure 61 shows when a new frame-synchronization pulse on FSX can safely occur relative to the last bit of the current frame.

Figure 61. Proper Positioning of Frame-Synchronization Pulses


For 2-bit delay:
Next frame-synchronization pulse here or later is OK.
For 1-bit delay:
Next frame-synchronization pulse here or later is OK.
For 0-bit delay:
Next frame-synchronization pulse here or later is OK.
CLKR/CLKX
FSR/FSX
DR/DX
Earliest possible time to begin transfer of next frame
Last bit of current frame

2.8 Multichannel Selection Modes

This section discusses the multichannel selection modes for the McBSP. Before using this McBSP feature, the DX pin must be configured for hi-Z control (via the pin multiplexing). That is, use the MCBSPn.DXZ pin instead of the MCBSPn.DX pin.

2.8.1 Channels, Blocks, and Partitions

A McBSP channel is a time slot for shifting the bits of one serial word in or out. Each McBSP supports up to 128 channels for reception and 128 channels for transmission.

In the receiver and in the transmitter, the 128 available channels are divided into eight blocks that each contain 16 contiguous channels:

Block 0: Channels 0–15
Block 1: Channels 16–31
Block 2: Channels 32–47
Block 3: Channels 48–63
Block 4: Channels 64–79
Block 5: Channels 80–95
Block 6: Channels 96–111
Block 7: Channels 112–127

The blocks are assigned to **partitions** according to the selected partition mode. In the two-partition mode (described in section 2.8.4), you assign one even-numbered block (0, 2, 4, or 6) to partition A and one odd-numbered block (1, 3, 5, or 7) to partition B. In the 8-partition mode (described in section 2.8.5), blocks 0 through 7 are automatically assigned to partitions A through H, respectively.

The number of partitions for reception and the number of partitions for transmission are independent. For example, it is possible to use two receive partitions (A and B) and eight transmit partitions (A–H).

2.8.2 Multichannel Selection

When a McBSP uses a time-division multiplexed (TDM) data stream while communicating with other McBSPs or serial devices, the McBSP may need to receive and/or transmit on only a few channels. To save memory and bus bandwidth, you can use a multichannel selection mode to prevent data flow in some of the channels.

Each channel partition has a dedicated channel enable register. If the appropriate multichannel selection mode is on, each bit in the register controls whether data flow is allowed or prevented in one of the channels that is assigned to that partition.

The McBSP has one receive multichannel selection mode (described in section 2.8.6) and three transmit multichannel selection modes (described in section 2.8.7).

2.8.3 Configuring a Frame for Multichannel Selection

Before you enable a multichannel selection mode, make sure you properly configure the data frame:

1) Select a single-phase frame (RPHASE/XPHASE = 0). Each frame represents a TDM data stream.

2) Set a frame length (in RFRLEN1/XFRLEN1) that includes the highest-numbered channel to be used. For example, if you plan to use channels 0, 15, and 39 for reception, the receive frame length must be at least 40 (RFRLEN1 = 39). For instance, if XFRLEN1 = 39, the receiver creates 40 time slots per frame but only receives data during time slots 0, 15, and 39 of each frame.

2.8.4 Using Two Partitions

For multichannel selection operation in the receiver and/or the transmitter, you can use two partitions or eight partitions (described in section 2.8.5). If you choose the 2-partition mode (RMCME = 0 for reception, XMCME = 0 for transmission), McBSP channels are activated using an alternating scheme. In response to a frame-synchronization pulse, the receiver or transmitter begins with the channels in partition A and then alternates between partitions B and A until the complete frame has been transferred. When the next frame-synchronization pulse occurs, the next frame is transferred beginning with the channels in partition A.

2.8.4.1 Assigning Blocks to Partitions A and B

For reception, any two of the eight receive-channel blocks can be assigned to receive partitions A and B, which means up to 32 receive channels can be enabled at any given point in time. Similarly, any two of the eight transmit-channel blocks (up 32 enabled transmit channels) can be assigned to transmit partitions A and B.

For reception:

❑ Assign an even-numbered channel block (0, 2, 4, or 6) to receive partition A by writing to the RPABLK bits. In the receive multichannel selection mode (described in section 2.8.6), the channels in this partition are controlled by receive channel enable register A (RCERA).

❑ Assign an odd-numbered block (1, 3, 5, or 7) to receive partition B with the RPBBLK bits. In the receive multichannel selection mode, the channels in this partition are controlled by receive channel enable register B (RCERB).

For transmission:

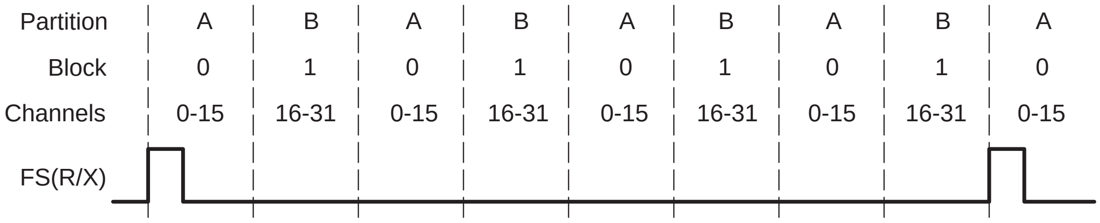
- Assign an even-numbered channel block (0, 2, 4, or 6) to transmit partition A by writing to the XPABLK bits. In one of the transmit multichannel selection modes (described in section 2.8.7), the channels in this partition are controlled by transmit channel enable register A (XCERA).

- Assign an odd-numbered block (1, 3, 5, or 7) to transmit partition B with the XPBBLK bits. In one of the transmit multichannel selection modes, the channels in this partition are controlled by transmit channel enable register B (XCERB).

Figure 62 shows an example of alternating between the channels of partition A and the channels of partition B. Channels 0-15 have been assigned to partition A, and channels 16-31 have been assigned to partition B. In response to a frame-synchronization pulse, the McBSP begins a frame transfer with partition A and then alternates between partitions B and A until the complete frame is transferred.

Figure 62. Alternating Between the Channels of Partition A and the Channels of Partition B

Two-partition mode. Example with fixed block assignments



Partition	A	B	A	B	A	B	A	B	A
Block	0	1	0	1	0	1	0	1	0
Channels	0-15	16-31	0-15	16-31	0-15	16-31	0-15	16-31	0-15

As explained in section 2.8.4.2, you can dynamically change which blocks of channels are assigned to the partitions.

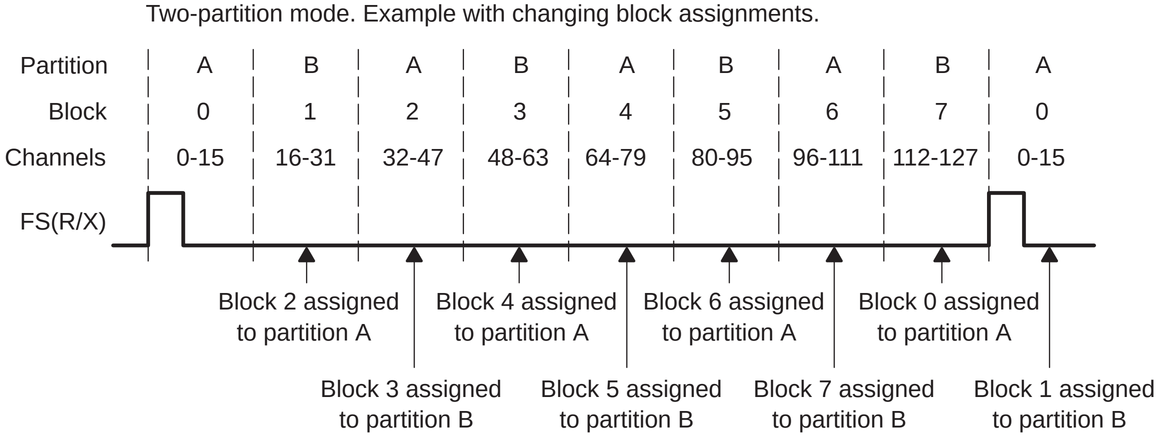
2.8.4.2 Reassigning Blocks During Reception/Transmission

If you want to use more than 32 channels, you can change which channel blocks are assigned to partitions A or B during the course of a data transfer. However, these changes must be carefully timed. While a partition is being transferred, its associated block assignment bits cannot be modified and its associated channel enable register cannot be modified. For example, if block 3 is being transferred and block 3 is assigned to partition A, neither RPABLK nor XPABLK can be modified to assign different channels to partition A nor can (R/X)CERA change the channel configuration for partition A. The following features of the McBSP can help you time the reassignment.

- ❑ The block of channels currently involved in reception/transmission (the current block) is reflected in the RCBLK/XCBLK bits. Your program can poll these bits to determine the active partition. When a partition is not active, it is safe to change its block assignment and channel configuration.
- ❑ At the end of every block, at the boundary of two partitions, an interrupt can be sent to the DSP core or MPU core. In response to the interrupt, the DSP core or MPU core can then check the RCBLK/XCBLK bits and update the inactive partition. See section 2.8.8.

Figure 63 shows an example of reassigning channels throughout a data transfer. In response to a frame-synchronization pulse, the McBSP alternates between partitions A and B. Whenever partition B is active, the DSP core or MPU core changes the block assignment for partition A. Whenever partition A is active, the DSP core or MPU core changes the block assignment for partition B.

Figure 63. Reassigning Channel Blocks Throughout a McBSP Data Transfer



Two-partition mode. Example with changing block assignments.
Partition
A
B
A
B
A
B
A
B
A
Block
0
1
2
3
4
5
6
7
0
Channels
0-15
16-31
32-47
48-63
64-79
80-95
96-111
112-127
0-15
FS(R/X)
Block 2 assigned to partition A
Block 4 assigned to partition A
Block 6 assigned to partition A
Block 0 assigned to partition A
Block 3 assigned to partition B
Block 5 assigned to partition B
Block 7 assigned to partition B
Block 1 assigned to partition B

2.8.5 Using Eight Partitions

For multichannel selection operation in the receiver and/or the transmitter, you can use eight partitions or two partitions (described in section 2.8.4). If you choose the 8-partition mode (RMCME = 1 for reception, XMCME = 1 for transmission), McBSP channels are activated in the following order: A, B, C, D, E, F, G, H. In response to a frame-synchronization pulse, the receiver or transmitter begins with the channels in partition A and then continues with the other partitions in order until the complete frame has been transferred. When the next frame-synchronization pulse occurs, the next frame is transferred, beginning with the channels in partition A.

In the 8-partition mode, the (R/X)PABLK and (R/X)PBBLK bits are ignored and the 16-channel blocks are assigned to the partitions as shown in Table 70 and Table 71. These assignments cannot be changed. The tables also show the registers used to control the channels in the partitions.

Table 70. Receive Channel Assignment and Control With Eight Receive Partitions

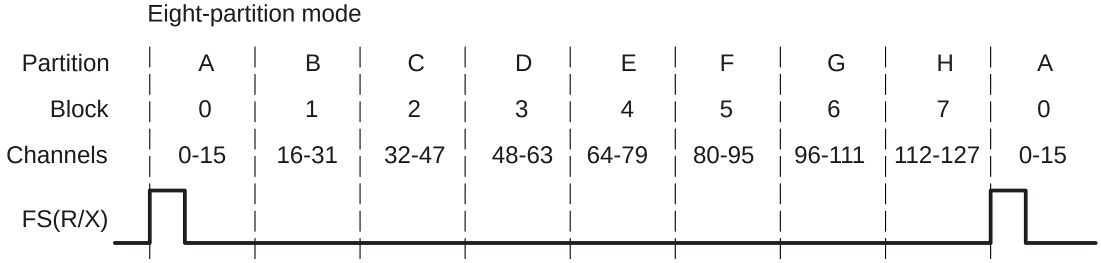
Receive Partition	Assigned Block of Receive Channels	Register Used For Channel Control
A	Block 0: channels 0 through 15	RCERA
B	Block 1: channels 16 through 31	RCERB
C	Block 2: channels 32 through 47	RCERC
D	Block 3: channels 48 through 63	RCERD
E	Block 4: channels 64 through 79	RCERE
F	Block 5: channels 80 through 95	RCERF
G	Block 6: channels 96 through 111	RCERG
H	Block 7: channels 112 through 127	RCERH

Table 71. Transmit Channel Assignment and Control With Eight Transmit Partitions

Transmit Partition	Assigned Block of Transmit Channels	Register Used For Channel Control
A	Block 0: channels 0 through 15	XCERA
B	Block 1: channels 16 through 31	XCERB
C	Block 2: channels 32 through 47	XCERC
D	Block 3: channels 48 through 63	XCERD
E	Block 4: channels 64 through 79	XCERE
F	Block 5: channels 80 through 95	XCERF
G	Block 6: channels 96 through 111	XCERG
H	Block 7: channels 112 through 127	XCERH

Figure 64 shows an example of the McBSP using the 8-partition mode. In response to a frame-synchronization pulse, the McBSP begins a frame transfer with partition A and then activates B, C, D, E, F, G, and H to complete a 128-word frame.

Figure 64. McBSP Data Transfer in the 8-Partition Mode



Eight-partition mode

Partition	A	B	C	D	E	F	G	H	A
Block	0	1	2	3	4	5	6	7	0
Channels	0-15	16-31	32-47	48-63	64-79	80-95	96-111	112-127	0-15

2.8.6 Receive Multichannel Selection Mode

The RMCM bit in MCR1 determines which channels are enabled for reception. When RMCM = 0, all 128 receive channels are enabled and cannot be disabled. When RMCM = 1, the receive multichannel selection mode is enabled. In this mode:

- Channels can be individually enabled or disabled. The only channels enabled are those selected in the appropriate receive channel enable registers (RCERs). The way channels are assigned to the RCERs depends on the number of receive channel partitions (2 or 8), as defined by the RMCME bit in MCR1.

- If a receive channel is disabled, any bits received in that channel are passed only as far as the receive buffer register(s) (RBR(s)). The receiver does not copy the content of the RBR(s) to the DRR(s), and as a result, does not set the receiver ready bit (RRDY). Therefore, no DSP/system DMA synchronization event request (internal McBSP signal REVT) is generated and no interrupt is generated if the receiver interrupt mode depends on RRDY (RINTM = 00b).

As an example of how the McBSP behaves in the receive multichannel selection mode, suppose only channels 0, 15, and 39 are enabled, and that the frame length is 40. The McBSP:

- Accepts bits shifted in from the DR pin in channel 0.
- Ignores bits received in channels 1-14.
- Accepts bits shifted in from the DR pin in channel 15.
- Ignores bits received in channels 16-38.
- Accepts bits shifted in from the DR pin in channel 39.

2.8.7 Transmit Multichannel Selection Modes

The XMCM bits in XCR2 determine which channels are enabled and unmasked for transmission. More details on enabling and masking are in section 2.8.7.1. The McBSP has three transmit multichannel selection modes described in the following table: XMCM = 01b, XMCM = 10b, and XMCM = 11b.

Table 72. Selecting a Transmit Multichannel Selection Mode With the XMCM Bits

XMCM	Transmit Multichannel Selection Mode
00b	No transmit multichannel selection mode is on. All channels are enabled and unmasked. No channels can be disabled or masked.
01b	All channels are disabled unless they are selected in the appropriate transmit channel enable registers (XCERs). If enabled, a channel in this mode is also unmasked. The XMCME bit in MCR2 determines whether 32 channels or 128 channels are selectable in XCERs.
10b	All channels are enabled, but they are masked unless they are selected in the appropriate transmit channel enable registers (XCERs). The XMCME bit in MCR2 determines whether 32 channels or 128 channels are selectable in XCERs.
11b	This mode is used for symmetric transmission and reception. All channels are disabled for transmission, unless they are enabled for reception in the appropriate receive channel enable registers (RCERs). Once enabled, they are masked, unless they are also selected in the appropriate transmit channel enable registers (XCERs). The XMCME bit in MCR2 determines whether 32 channels or 128 channels are selectable in RCERs and XCERs.

As an example of how the McBSP behaves in a transmit multichannel selection mode, suppose that XMCM = 01b (all channels disabled unless individually enabled) and that only channels 0, 15, and 39 are enabled. Suppose also that the frame length is 40. The McBSP:

- Shifts data to the DXZ pin in channel 0.
- Places the DXZ pin in the high impedance state in channels 1-14.
- Shifts data to the DXZ pin in channel 15.
- Places the DXZ pin in the high impedance state in channels 16-38.
- Shifts data to the DXZ pin in channel 39.

2.8.7.1 Disabling/Enabling Versus Masking/Unmasking

For transmission, a channel can be:

- Enabled and unmasked (transmission can begin and can be completed).
- Enabled but masked (transmission can begin but cannot be completed).
- Disabled (transmission cannot occur).

The following list defines the channel control options:

Enabled channel A channel that can begin transmission by passing data from the data transmit register(s) (DXR(s)) to the transmit shift registers (XSR(s)).

Masked channel A channel that cannot complete transmission. The DXZ pin is held in the high impedance state; data cannot be shifted out on the DXZ pin.

In systems where symmetric transmit and receive provides software benefits, this feature allows transmit channels to be disabled on a shared serial bus. A similar feature is not needed for reception because multiple receptions cannot cause serial bus contention.

Disabled channel A channel that is not enabled. A disabled channel is also masked.

Because no DXR-to-XSR copy occurs, the XRDY bit in SPCR2 is not set. Therefore, no DSP/system DMA synchronization event request (McBSP TX request, internal McBSP signal XEVT) is generated, and if the transmit interrupt mode depends on XRDY (XINTM = 00b in SPCR2), no interrupt is generated.

The $\overline{\text{XEMPTY}}$ bit in SPCR2 is not affected.

Unmasked channel A channel that is not masked. Data in the XSR(s) is shifted out on the DXZ pin.

2.8.7.2 Activity on McBSP Pins for Different Values of XMCM

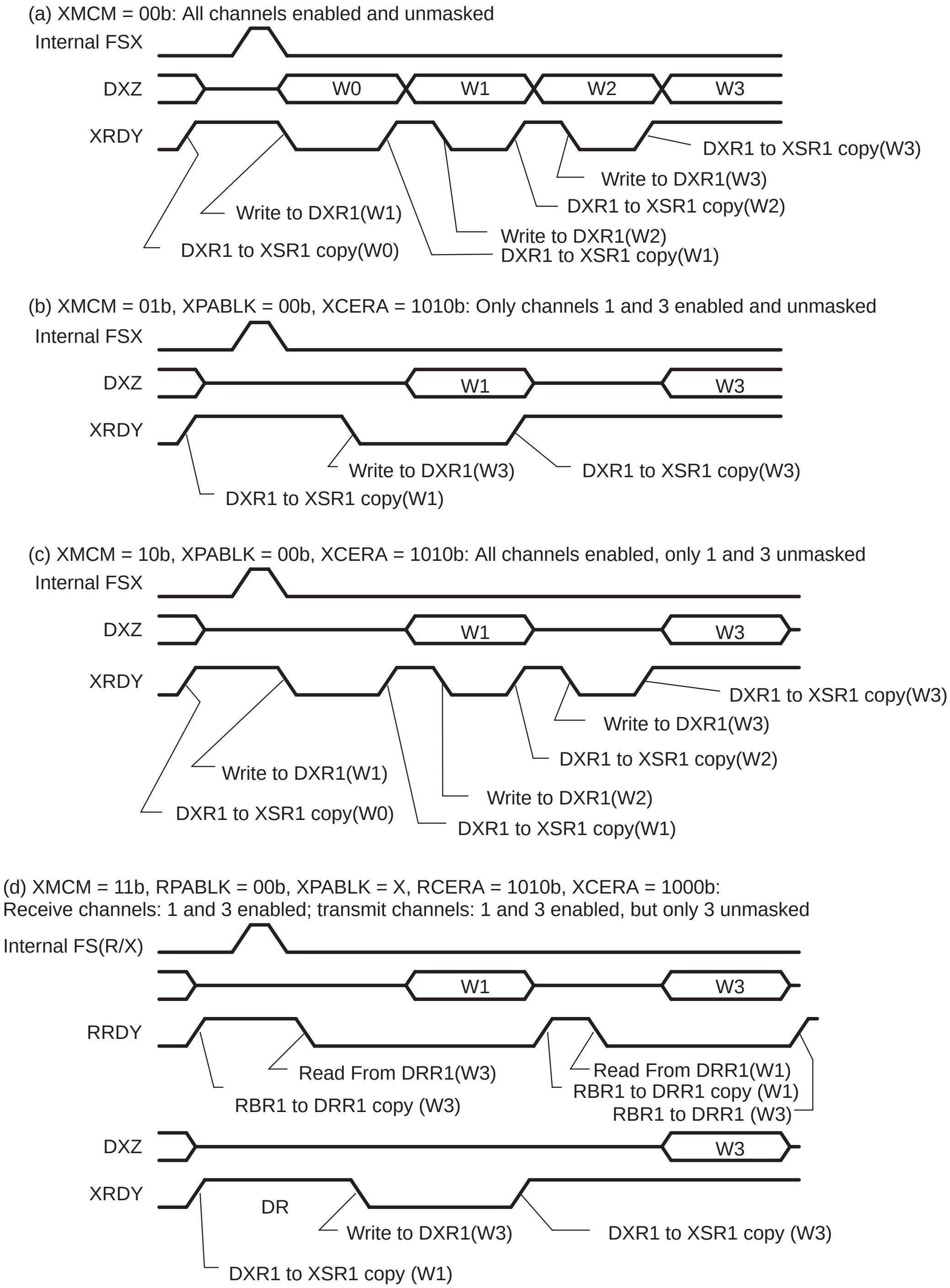
Figure 65 shows the activity on the McBSP pins for the various XMCM values. In all cases, the transmit frame is configured as follows:

- XPHASE = 0: Single-phase frame (required for multichannel selection modes)
- XFRLEN1 = 0000011b: 4 words per frame
- XWDLEN1 = 000b: 8 bits per word
- XMCME = 0: 2-partition mode (only partitions A and B used)

When XMCM = 11b, transmission and reception are symmetric, which means the corresponding bits for the receiver (RPHASE, RFRLEN1, RWDLEN1, and RMCME) must have the same values as XPHASE, XFRLEN1, and XWDLEN1, respectively.

In Figure 65, the arrows showing where the various events occur are only sample indications. Wherever possible, there is a time window in which these events can occur.

Figure 65. Activity on McBSP Pins for the Possible Values of XMCM



(a) XMCM = 00b: All channels enabled and unmasked

(b) XMCM = 01b, XPABLK = 00b, XCERA = 1010b: Only channels 1 and 3 enabled and unmasked

(c) XMCM = 10b, XPABLK = 00b, XCERA = 1010b: All channels enabled, only 1 and 3 unmasked

(d) XMCM = 11b, RPABLK = 00b, XPABLK = X, RCERA = 1010b, XCERA = 1000b:
Receive channels: 1 and 3 enabled; transmit channels: 1 and 3 enabled, but only 3 unmasked

2.8.8 Using Interrupts Between Block Transfers

When a multichannel selection mode is used, an interrupt request can be sent to the DSP core or MPU core at the end of every 16-channel block (at the boundary between partitions and at the end of the frame). In the receive multichannel selection mode, a receive interrupt (McBSP RX Interrupt, McBSP module signal RINT) request is generated at the end of each block transfer, if RINTM = 01b. In any of the transmit multichannel selection modes, a transmit interrupt (McBSP TX Interrupt, McBSP module signal XINT) request is generated at the end of each block transfer if XINTM = 01b. When RINTM/XINTM = 01b, no interrupt is generated unless a multichannel selection mode is on.

These interrupt pulses are active high and last for two internal input clock cycles.

This type of interrupt is especially helpful if you are using the two-partition mode (described in section 2.8.4) and you want to know when you can assign a different block of channels to partition A or B.

2.9 General-Purpose I/O on the McBSP Pins

Table 73 summarizes how to use the McBSP pins as general-purpose I/O pins. All of the bits mentioned in the table except $\overline{\text{XRST}}$ and $\overline{\text{RRST}}$ are in the pin control register (PCR). $\overline{\text{XRST}}$ and $\overline{\text{RRST}}$ are in the serial port control registers (SPCRs).

To use receiver pins CLKR, FSR, and DR as general-purpose I/O pins rather than as serial port pins, you must meet two conditions:

- The receiver of the serial port must be in reset ($\overline{\text{RRST}}$ = 0 in SPCR1).
- General-purpose I/O must be enabled for the serial port receiver (RIOEN = 1 in PCR).

The CLKR and FSR pins can be individually configured as either input or output pins with the CLKRM and FSRM bits, respectively. The DR pin can only be an input pin. Table 73 shows which bits in PCR are used to read from/write to these pins.

For the transmitter pins CLKX, FSX, and DX, you must meet two conditions:

- The transmitter of the serial port must be in reset ($\overline{\text{XRST}}$ = 0 in SPCR2).
- General-purpose I/O must be enabled for the serial port transmitter (XIOEN = 1 in PCR).

The CLKX and FSX pins can be individually configured as input or output pins with the CLKXM and FSXM bits, respectively. The DX pin can only be an output pin. Table 73 shows which bits in PCR are used to read from/write to these pins.

For the CLKS pin, you must meet these reset and I/O enable conditions:

- Both the receiver and transmitter of the serial port must be in reset ($\overline{\text{RRST}}$ = 0 and $\overline{\text{XRST}}$ = 0).
- General-purpose I/O must be enabled for both the receiver and the transmitter (RIOEN = 1 and XIOEN = 1).

The CLKS pin can only be an input pin. To read the status of the signal on the CLKS pin, read the CLKS_STAT bit in PCR.

Table 73. Using McBSP Pins for General-Purpose Input/Output

Pin	General-Purpose Use Enabled by This Bit Combination	Selected as Output When …	Output Value Driven From This Bit	Selected as Input When …	Input Value Read From This Bit
CLKX	$\overline{\text{XRST}}$ = 0 XIOEN = 1	CLKXM = 1	CLKXP	CLKXM = 0	CLKXP
FSX	$\overline{\text{XRST}}$ = 0 XIOEN = 1	FSXM = 1	FSXP	FSXM = 0	FSXP
DX	$\overline{\text{XRST}}$ = 0 XIOEN = 1	Always	DX_STAT	Never	Does not apply
CLKR	$\overline{\text{RRST}}$ = 0 RIOEN = 1	CLKRM = 1	CLKRP	CLKRM = 0	CLKRP
FSR	$\overline{\text{RRST}}$ = 0 RIOEN = 1	FSRM = 1	FSRP	FSRM = 0	FSRP
DR	$\overline{\text{RRST}}$ = 0 RIOEN = 1	Never	Does not apply	Always	DRSTAT
CLKS	$\overline{\text{RRST}}$ = $\overline{\text{XRST}}$ = 0 RIOEN = XIOEN = 1	Never	Does not apply	Always	CLKSSTAT

2.10 Reset Considerations

The serial port can be reset by an OMAP5912 device reset or by directly using software through the $\overline{\text{RRST}}$, $\overline{\text{XRST}}$ and $\overline{\text{GRST}}$ bits in the serial port control registers.

When the McBSP is reset in either way, the machine is reset to its initial state, including reset of all counters and status bits. The receive status bits include RFULL, RRDY, and RSYNCERR. The transmit status bits include XEMPTY, XRDY, and XSYNCERR.

Table 74 shows the state of McBSP signals when the serial port is reset due to an OMAP5912 reset and a direct receiver/transmitter reset. Note that the actual pin state depends on the device's pin signal multiplexing mode, as described on the device datasheet.

Table 74. Reset State of Each McBSP Pin

Pin	Possible State(s)	State Forced By OMAP5912 Reset	State Forced By Receiver/Transmitter Reset
			Receiver reset ($\overline{\mathrm{RRST}}$ = 0 and $\overline{\mathrm{GRST}}$ = 1)
DR	I	Input	Input
CLKR	I/O/Z	Input	Known state if input; CLKR running if output
FSR	I/O/Z	Input	Known state if input; FSRP inactive state if output
CLKS	I/O/Z	Input	Input
			Transmitter reset ($\overline{\mathrm{XRST}}$ = 0 and $\overline{\mathrm{GRST}}$ = 1)
DX	O	Output	Output, drives low after bit clock provided
DXZ	O/Z	Output/high impedance	Output, high impedance (OMAP5912 only) after bit clock provided
CLKX	I/O/Z	Input	Known state if input; CLKX running if output
FSX	I/O/Z	Input	Known state if input; FSXP inactive state if output

Note: In Possible State(s) column, I = input, O = output, Z = high impedance

2.10.1 Software Reset Considerations: Using $\overline{\mathrm{RRST}}$, $\overline{\mathrm{XRST}}$, and $\overline{\mathrm{GRST}}$ to Reset the Serial Port

The serial port transmitter and receiver can be directly reset using the RRST and XRST bits in the serial port control registers. The sample rate generator can be reset directly using the GRST bit in SPCR2.

❑ RRST and XRST reset. When the receiver and transmitter reset bits, RRST and XRST, are loaded with 0s, the respective portions of the McBSP are reset and activity stops in the corresponding section of the serial port. All input-only pins such as DR and CLKS, and all other input pins are in a known state. The FSR and FSX pins are driven to their inactive states if they are not outputs. If the CLKR and CLKX pins are programmed as outputs, they are driven by CLKG, provided that GRST = 1. Lastly, the DXZ pin is in the high-impedance state when the transmitter and/or the device is reset.

During normal operation, the sample rate generator is reset if the GRST bit is cleared. GRST must be 0 only when neither the transmitter nor the receiver is using the sample rate generator. In this case, the internal sample rate generator clock (CLKG) is driven high for OMAP5912 and its frame-synchronization signal (FSG) is driven inactive low.

When the sample rate generator is not in the reset state (GRST = 1), pins FSR and FSX are in an inactive state when RRST = 0 and XRST = 0, respectively, even if they are outputs driven by FSG. This ensures that when only one portion of the McBSP is in reset, the other portion can continue operation when GRST = 1 and its frame synchronization is driven by FSG.

❑ GRST reset. The sample rate generator is reset when the DSP is reset or when GRST is loaded with 0. In the case of an OMAP5912 reset, the sample rate generator clock, CLKG, is driven by the internal clock source divided by 2 and the frame-synchronization signal, FSG, is driven inactive low.

  When neither the transmitter nor the receiver is fed by CLKG and FSG, you can reset the sample rate generator by clearing GRST. In this case, CLKG is driven high and FSG is driven inactive low. If you then set GRST, CLKG starts and runs as programmed. Later, if GRST = 1, FSG pulses active high after the programmed number of CLKG cycles has elapsed. If the sample rate generator is reset after CLKG has been toggling, CLKG will keep the level it had when the sample rate generator was reset.

2.10.2 Hardware Reset Considerations: OMAP5912 Device Reset

An OMAP5912 device reset places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed, GRST = FRST = RRST = XRST = 0 keep the entire serial port in the reset state.

When the whole DSP is reset with a device reset, the entire serial port, including the transmitter, receiver, and the sample rate generator, is reset. All input-only pins and 3-state pins must be in a known state. The output-only pin DXZ is in the high-impedance state.

The OMAP5912 reset forces the sample rate generator clock, CLKG, to have the frequency of half the internal clock source. No pulses are generated on the sample rate generator’s frame-synchronization signal, FSG.

When the device is pulled out of reset, the serial port remains in the reset state. In this state the DR and DX pins can be used as general-purpose I/O pins. as described in section 2.9.

2.11 Initialization

This section provides both the general initialization procedure for the McBSP, and the specialized procedure when the transmitter is used and the external device provides the transmit frame synchronization FSX.

2.11.1 General McBSP Initialization Procedure

This section provides the general initialization procedure:

1) Ensure that no portion of the McBSP is using the internal sample rate generator signal CLKG and the internal frame-synchronization generator signal FSG (GRST = FRST = 0). The respective portion of the McBSP needs to be in reset (XRST = 0 and/or RRST = 0). If coming out of an OMAP5912 reset, this step is not required.

2) Program the control registers as required. Ensure the internal sample rate generator and the internal frame-synchronization generator are still in reset (GRST = FRST = 0). Also ensure the respective portion of the McBSP is still in reset in this step (XRST = 0 and/or RRST = 0).

3) Wait for proper internal synchronization. If the external device provides the bit clock, wait for two CLKR or CLKX cycles. If the McBSP generates the bit clock as a clock master, wait for two CLKSRG cycles. In this case, the clock source to the sample rate generator (CLKSRG) is selected by the CLKSM bit in the SRGR2 register and the SCLKME bit in the PCR register.

4) Skip this step if the bit clock is provided by the external device. This step only applies if the McBSP is the bit clock master and the internal sample rate generator is used.

   a) Start the sample rate generator by setting the GRST bit. Wait two CLKG bit clocks for synchronization. CLKG is the output of the sample rate generator.

   b) On the next rising edge of CLKSRG, CLKG transitions to 1 and starts clocking with a frequency equal to 1/(CLKGDV + 1) of the sample rate generator source clock CLKSRG.

5) Skip this step if the transmitter is not used. If the transmitter is used, a transmit synchronization error (XSYNCERR) may occur when it is enabled for the first time after device reset. To clear any potential XSYNCERR that occurs on the transmitter at this time:

   a) Set the XRST bit to enable the transmitter.

   b) Wait for any unexpected frame synchronization errors to occur. If the external device provides the bit clock, wait for two CLKR or CLKX cycles. If the McBSP generates the bit clock as a clock master, wait for two CLKG cycles. The unexpected frame synchronization error (XSYNCERR), if any, occurs within this time period.

   c) Disable the transmitter (XRST = 0). This clears any outstanding XSYNCERR.

6) Set up data acquisition as required:

   a) If the DSP/system DMA is used to service the McBSP, set up data acquisition as desired and start the DSP/system DMA in this step, before the McBSP is taken out of reset.

   b) If an interrupt is used to service the McBSP, enable the transmit and/or receive interrupt as required.

   c) If XRDY/RRDY polling is used to service the McBSP, no action is required in this step.

7) Set the XRST bit and/or the RRST bit to enable the corresponding section of the McBSP. The McBSP is now ready to transmit and/or receive.

   a) If the DSP/system DMA is used to service the McBSP, it services the McBSP automatically upon receiving the McBSP RX or TX DSP/system DMA event request (internal McBSP signal R/XEVT).

   b) If an interrupt is used to service the McBSP, the interrupt service routine is automatically entered upon receiving the McBSP TX Interrupt (internal McBSP signal XINT) and/or McBSP RX Interrupt (internal McBSP signal RINT).

   c) If polling is used to service the McBSP, it can do so now by polling the XRDY and/or RRDY bit.

8) If the internal frame-synchronization generator is used (FSGM = 1), proceed to the additional steps below to turn on the internal frame-synchronization generator. Initialization is complete if any one of the following is true:

   a) The external device generates frame synchronization FSX and/or FSR. The McBSP is now ready to transmit and/or receive upon receiving external frame synchronization.

   b) The McBSP generates transmit frame synchronization FSX upon each DXR-to-XSR copy. The internal frame synchronization generator is not used (FSGM = 0).

Additional steps to turn on the internal frame synchronization generator (only applies if FSGM = 1):

9) Skip this step if the transmitter is not used. If the transmitter is used, ensure that DXR is serviced before you start the internal frame synchronization generator. You can do so by checking that XEMPTY = 1 (XSR is not empty) in SPCR2.

10) Set the FRST bit to start the internal frame synchronization generator. The internal frame synchronization signal FSG is generated on a CLKG active edge after 2 CLKG clocks cycles have elapsed.

Notes:

1) The necessary duration of the active-low period of $\overline{\text{XRST}}$ or $\overline{\text{RRST}}$ is at least two CLKR/CLKX cycles.

2) The appropriate bits in serial port configuration registers SPCR[1,2], PCR, RCR[1,2], XCR[1,2], and SRGR[1,2] must only be modified when the affected portion of the serial port is in its reset state.

3) In most cases, the data transmit registers (DXR[1,2]) must be loaded by the DSP core, MPU core, or DSP/system DMA controller only when the transmitter is enabled ($\overline{\text{XRST}}$ = 1). An exception to this rule is when these registers are used for companding internal data (see section 2.2.2).

4) The bits in the channel control registers (MCR[1,2], RCER[A-H], XCER[A-H]) can be modified at any time as long as they are not being used by the current reception/transmission in a multichannel selection mode.

2.11.2 Special Case: External Device is the Transmit Frame Master

Care must be taken if the transmitter expects a frame synchronization from an external device. After the transmitter comes out of reset (XRST = 1), it waits for a frame synchronization from the external device. If the first frame synchronization arrives very shortly after the transmitter is enabled, the DSP core/MPU core or DSP/system DMA may not have a chance to service DXR. In this case, the transmitter shifts out the default data in XSR instead of the desired value, which has not yet arrived in DXR. This causes problems in some applications, as the first data element in the frame is invalid. The data stream appears element-shifted (the first data word may appear in the second channel instead of the first).

To ensure proper operation when the external device is the frame master, you must ensure that DXR is already serviced with the first word when a frame synchronization occurs. To do so, you can keep the transmitter in reset until the first frame synchronization is detected. Upon detection of the first frame synchronization , the McBSP can generate an interrupt to the DSP core or MPU core. Within the interrupt service routine, the transmitter is taken out of reset (XRST = 1). This ensures that the transmitter does not begin data transfers at the data pin during the first frame synchronization period. This also provides almost an entire frame period for the DSP core to service DXR with the first word before the second frame synchronization occurs. The transmitter only begins data transfers upon receiving the second frame synchronization. At this point, DXR is already serviced with the first word.

The interrupt service routine must be set up first according to the description in step 9. Then follow this modified procedure for proper initialization:

1) Ensure that no portion of the McBSP is using the internal sample rate generator signal CLKG and the internal frame synchronization generator signal FSG (GRST = FRST = 0). The respective portion of the McBSP needs to be in reset (XRST = 0 and/or RRST = 0). If coming out of an OMAP5912 reset, this step is not required.

2) Program SRGR and other control registers as required. Ensure the internal sample rate generator and the internal frame synchronization generator are still in reset (GRST = FRST = 0). Also ensure the respective portion of the McBSP is still in reset in this step (XRST = 0 and/or RRST = 0).

3) Program the XINTM bits to 10b in SPCR2 to generate an interrupt to the DSP core/MPU core upon detection of a transmit frame synchronization. Do not globally enable the McBSP TX interrupt (XINT) in this step.

4) Wait for proper internal synchronization. If the external device provides the bit clock, wait for two CLKR or CLKX cycles. If the McBSP generates the bit clock as a clock master, wait for two CLKSRG cycles. In this case, the clock source to the sample rate generator (CLKSRG) is selected by the CLKSM bit in the SRGR2 register and the SCLKME bits in the PCR register.

5) Skip this step if the bit clock is provided by the external device. This step only applies if the McBSP is the bit clock master and the internal sample rate generator is used.

   a) Start the sample rate generator by setting the GRST bit. Wait two CLKG bit clocks for synchronization. CLKG is the output of the sample rate generator.

   b) On the next rising edge of CLKSRG, CLKG transitions to 1 and starts clocking with a frequency equal to 1/(CLKGDV + 1) of the sample rate generator source clock CLKSRG.

6) A transmit synchronization error (XSYNCERR) may occur when it is enabled for the first time after device reset. To clear any potential XSYNCERR that occurs on the transmitter at this time:

   a) Set the XRST bit to enable the transmitter.

   b) Wait for any unexpected frame synchronization errors to occur. If the external device provides the bit clock, wait for two CLKR or CLKX cycles. If the McBSP generates the bit clock as a clock master, wait for two CLKG cycles. The unexpected frame synchronization error (XSYNCERR), if any, occurs within this time period.

   c) Disable the transmitter (XRST = 0). This clears any outstanding XSYNCERR.

7) Set up data acquisition as required:

   a) If the DSP/system DMA is used to service the McBSP, set up data acquisition as desired and start the DSP/system DMA in this step, before the McBSP is taken out of reset.

   b) If an interrupt is used to service the McBSP, no action is required in this step.

   c) If XRDY/RRDY polling is used to service the McBSP, no action is required in this step.

8) Globally enable the McBSP TX interrupt (McBSP module signal XINT). In this step, the McBSP transmitter is still in reset. Upon detection of the first transmit frame synchronization from the external device, the McBSP generates an interrupt to the DSP core or MPU core and the DSP core/MPU core enters the interrupt service routine (ISR). The ISR needs to perform these tasks in this order:

   a) Modify the XINTM bits to the value desired for normal McBSP operation. If DSP core or MPU core interrupt is used to service the McBSP in normal operations, ensure that the XINTM bits are modified to 0 to detect the McBSP XRDY event. If no McBSP interrupt is desired in normal operations, disable future McBSP-to-DSP core/MPU core interrupts.

   b) Set the XRST bit and/or the RRST bit to enable the respective portion of the McBSP. The McBSP is now ready to transmit and/or receive.

9) Service the McBSP:

   a) If XRDY/RRDY polling is used to service the McBSP in normal operations, it can do so upon exit from the ISR.

   b) If an interrupt is used to service the McBSP in normal operations, upon XRDY, interrupt service routine is entered. Set up the ISR to verify that XRDY = 1, and service the McBSP accordingly.

   c) If DSP/system DMA is used to service the McBSP in normal operations, it services the McBSP automatically upon receiving the McBSP TX and/or RX DSP/system DMA event request (internal McBSP signal X/REVT).

10) Upon detection of the second frame synchronization, DXR is already serviced and the transmitter is ready to transmit the valid data. The receiver is also serviced properly by the DSP core.

Note:

1) The necessary duration of the active-low period of XRST or RRST is at least two CLKR/CLKX cycles.
2) Only modify the appropriate bits in serial port configuration registers SPCR[1,2], PCR, RCR[1,2], XCR[1,2], and SRGR[1,2] when the affected portion of the serial port is in its reset state.
3) In most cases, the data transmit registers (DXR[1,2]) should be loaded by the DSP core, MPU core or by the DSP/system DMA controller only when the transmitter is enabled (XRST = 1).
4) The bits in the channel control registers, MCR[1,2], RCER[A.H], and XCER[A.H], can be modified at any time as long as they are not being used by the current reception/transmission in a multichannel selection mode.

2.12 Interrupt Support

This section describes the interrupt requests that can be generated by the McBSP, including what events can generate interrupts and how the interrupts are reported to the DSP core or MPU core.

2.12.1 Interrupt Events and Requests

The transmitter and receiver interrupt events signal the DSP core or MPU core of changes to the serial port status. There are four options for configuring this event for both the transmitter and receiver. The options are set by the following mode bits:

❑ The RINTM bits in the SPCR1 register (see Table 30) determine which event generates a receive interrupt request to the DSP core or MPU core.

❑ The XINTM bits in the SPCR2 register (see Table 58) determine which event generates a transmit interrupt request to the DSP core or MPU core.

The four options to configure the events that generate the interrupts:

❑ Receive/transmit interrupt generated when RRDY/XRDY changes from 0 to 1.

❑ Receive/transmit interrupt generated by an end-of-block or end-of-frame condition in the reception/transmission multichannel selection mode.

❑ Receive/transmit interrupt generated by a new receive/transmit frame-synchronization pulse on the rising edge only.

❑ Receive/transmit interrupt generated when RSYNCERR/XSYNCERR is set.

2.12.2 Interrupt Multiplexing

There are three level 2 interrupt controllers in OMAP5912:

❑ One MPU level 2 interrupt, also referred to as MPU interrupt level 2

❑ One DSP level 2.1 interrupt handler

❑ One DSP level 2 interrupt handler, referred as DSP interrupt level 2.0

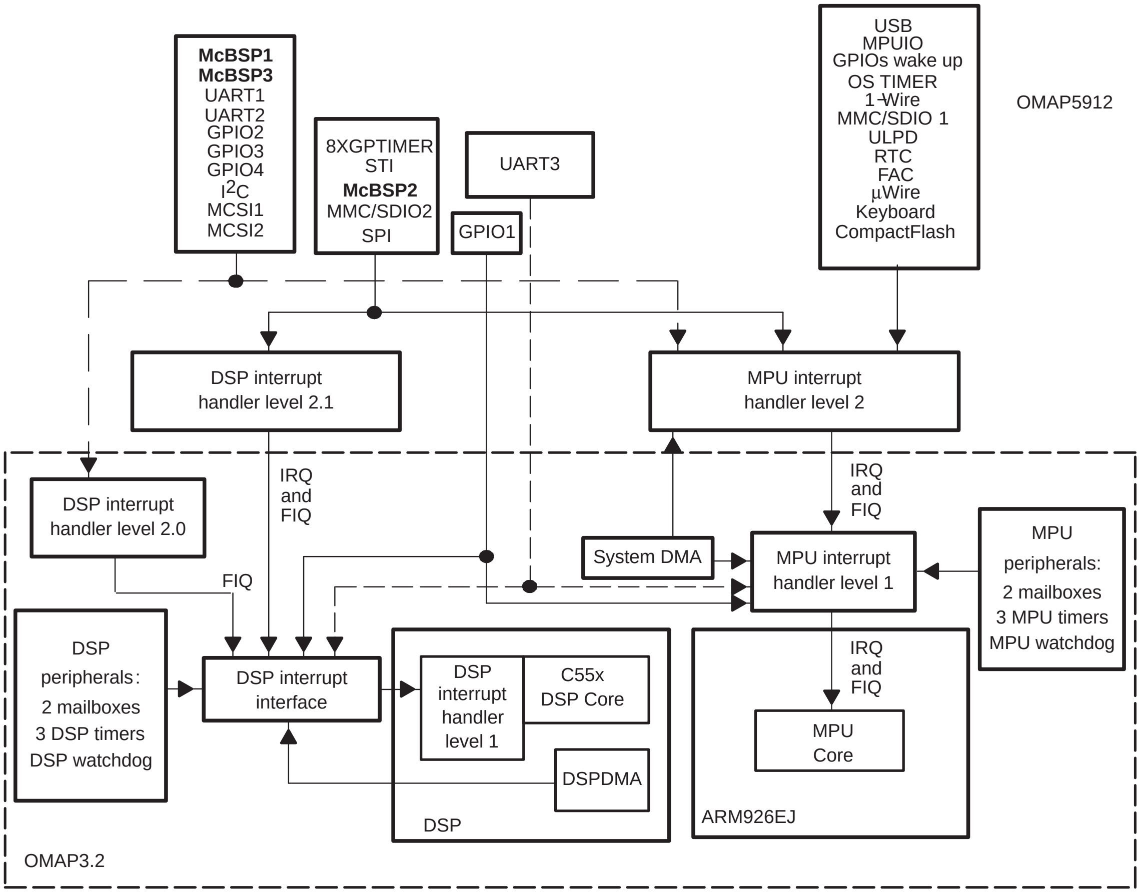
There are two level 1 interrupt controllers:

❑ One MPU level 1 interrupt handler

❑ One DSP level 1 interrupt handler

For more details on the OMAP5912 interrupt multiplexing, see the *OMAP5912 Multimedia Processor Interrupts Reference Guide* (SPRU757). Figure 66 shows how the McBSP modules can trigger an interrupt in each processor.

Figure 66. OMAP5912 Interrupt Interconnect



2.12.2.1 McBSP1 Interrupt Mapping

Table 75 identifies the McBSP1 interrupts. McBSP1 generates level 2 interrupts for both the DSP core and the MPU core.

Table 75. McBSP1 Interrupt Mapping

Incoming Interrupts	Level 2 DSP Interrupt	Level 2 MPU Interrupt
McBSP1 TX Interrupt	IRQ_02	IRQ_12
McBSP1 RX Interrupt	IRQ_03	IRQ_13

2.12.2.2 McBSP2 Interrupt Mapping

Table 76 identifies the McBSP2 interrupts. McBSP2 generates level 2.1 interrupts for the DSP and level 2 interrupts for the MPU.

Table 76. McBSP2 Interrupt Mapping

Incoming Interrupts	Level 2.1 DSP Interrupt	Level 1 MPU Interrupt	Level 2 MPU Interrupt
McBSP2 TX Interrupt	IRQ_10	IRQ_4	–
McBSP2 RX Interrupt	IRQ_11	IRQ_5	–
McBSP2 RX Overflow	–	–	IRQ_31

2.12.2.3 McBSP3 Interrupt Mapping

Table 77 identifies the McBSP3 interrupts. McBSP3 generates level 2 interrupts for both the DSP and the MPU.

Table 77. McBSP3 Interrupt Mapping

Incoming Interrupts	Level 2 DSP Interrupt	Level 2 MPU Interrupt
McBSP3 TX Interrupt	IRQ_00	IRQ_10
McBSP3 RX Interrupt	IRQ_01	IRQ_11

2.13 DSP/System DMA Event Support

In OMAP5912, the McBSP is capable of sending DSP/system DMA requests to either the DSP core GDMA Handler or MPU core GDMA Handler in order to generate DSP or system DMA events, respectively.

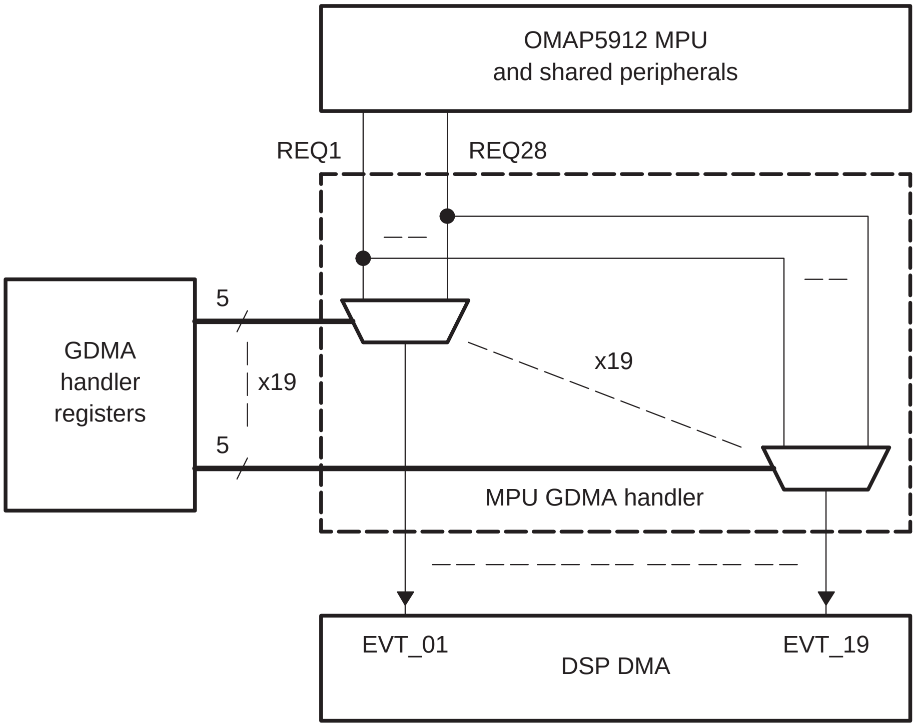
A McBSP DSP/system DMA request is generated as follows:

❑ McBSP RX request (McBSP Module RINT signal). This request is issued when the data has been received in the data receive registers (DRRs). In other words, an RRDY change from 0 to 1 will generate a DSP/system DMA receive request.

❑ McBSP TX request (McBSP Module XINT signal). This request is issued when the data transmit registers (DXRs) are ready to accept the next serial word for transmission. In other words, an XRDY change from 0 to 1 will generate a DSP/system DMA transmit request.

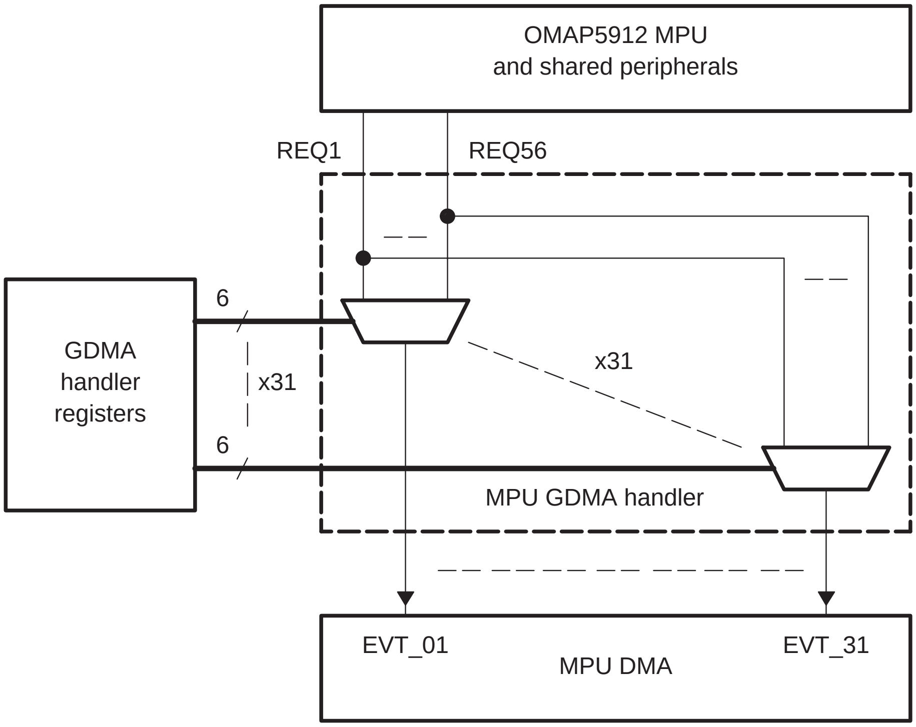
The OMAP5912 DSP core GDMA Handler acts as a crossbar which maps these McBSP requests to the 19 DSP core DMA synchronization events (see Figure 67).

Figure 67. DSP GDMA Handler



The OMAP5912 MPU core GDMA Handler acts as a crossbar so that each of the incoming DMA requests can be remapped to any of the 31 system DMA events (see Figure 68).

Figure 68. MPU GDMA Handler



For more details on the DSP/system DMA handlers, see the *OMAP5912 Multimedia Processor Direct Memory Access (DMA) Support Reference Guide* (SPRU755).

Note that the system DMA only supports 16-bit transfers to McBSP2.

The following sections describe the specific DSP/system DMA event request lines for each McBSP.

2.13.1 McBSP1 DSP/System DMA Event Request Lines

Table 78 identifies McBSP1 DSP/system DMA request lines.

Table 78. McBSP1 DSP/System DMA Request Lines

DMA Request Source	DSP DMA Request Line	System DMA Request Line- MPU
McBSP1 TX	PERIPH_REQ_08	DMA_REQ_08
McBSP1 RX	PERIPH_REQ_09	DMA_REQ_09

2.13.2 McBSP2 DSP/System DMA Event Request Lines

Table 79 identifies McBSP2 DSP/system DMA request lines.

Table 79. McBSP2 DSP/System DMA Request Lines

DMA Request Source	DSP DMA Request Line	System DMA Request Line- MPU
McBSP2 TX	PERIPH_REQ_27	DMA_REQ_16
McBSP2 RX	PERIPH_REQ_28	DMA_REQ_17

2.13.3 McBSP3 DSP/System DMA Event Request Lines

Table 80 identifies McBSP3 DSP/system DMA request lines.

Table 80. McBSP3 DSP/System DMA Request Lines

DMA Request Source	DSP DMA Request Line	System DMA Request Line- MPU
McBSP3 TX	PERIPH_REQ_10	DMA_REQ_10
McBSP3 RX	PERIPH_REQ_11	DMA_REQ_11

2.14 Emulation Considerations

FREE and SOFT are special emulation bits in SPCR2 that determine the state of the McBSP when a breakpoint is encountered in the high-level language debugger. If FREE = 1, the clock continues to run upon a software breakpoint and data is shifted out. When FREE = 1, the SOFT bit is a *don’t care.*

If FREE = 0, the SOFT bit takes effect. If SOFT = 0 when breakpoint occurs, the clock stops immediately, aborting a transmission. If SOFT = 1 and a breakpoint occurs while transmission is in progress, the transmission continues until completion of the transfer and then the clock halts. Table 81 lists these options.

The McBSP receiver functions in a similar fashion. If a mode other than the immediate stop mode (SOFT = FREE = 0) is chosen, the receiver continues running and an overrun error is possible.

Table 81. McBSP Emulation Modes Selectable with FREE and SOFT Bits in SPCR2

FREE	SOFT	McBSP Emulation Mode
0	0	Immediate stop mode (reset condition) The transmitter or receiver stops immediately in response to a breakpoint.
0	1	Soft stop mode When a breakpoint occurs, the transmitter stops after completion of the current word. The receiver is not affected.
1	0 or 1	Free run mode The transmitter and receiver continue to run when a breakpoint occurs.

2.15 Data Packing Using the McBSP

This section shows two ways to implement data packing in the McBSP.

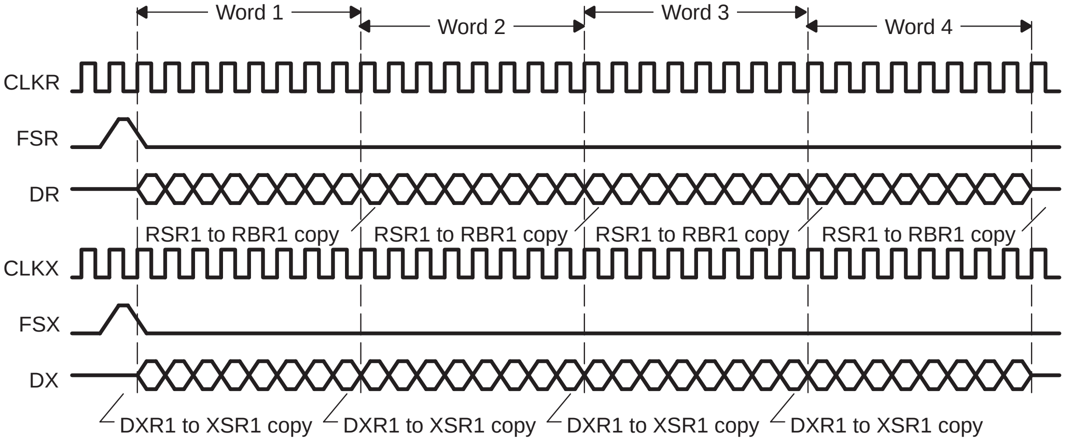
2.15.1 Data Packing Using Frame Length and Word Length

Frame length and word length can be manipulated to effectively pack data. For example, consider a situation where four 8-bit words are transferred in a single-phase frame as shown in Figure 69. In this case:

- ❏ (R/X)PHASE = 0: Single-phase frame
- ❏ (R/X)FRLEN1 = 0000011b: 4-word frame
- ❏ (R/X)WDLEN1 = 000b: 8-bit words

Four 8-bit data words are transferred to and from the McBSP by the DSP core and MPU core or by the system DMA or DSP DMA controller. Thus, four reads from DRR1 and four writes to DXR1 are necessary for each frame.

Figure 69. Four 8-Bit Data Words Transferred To/From the McBSP



This data can also be treated as a single-phase frame consisting of one 32-bit data word, as shown in Figure 70. In this case:

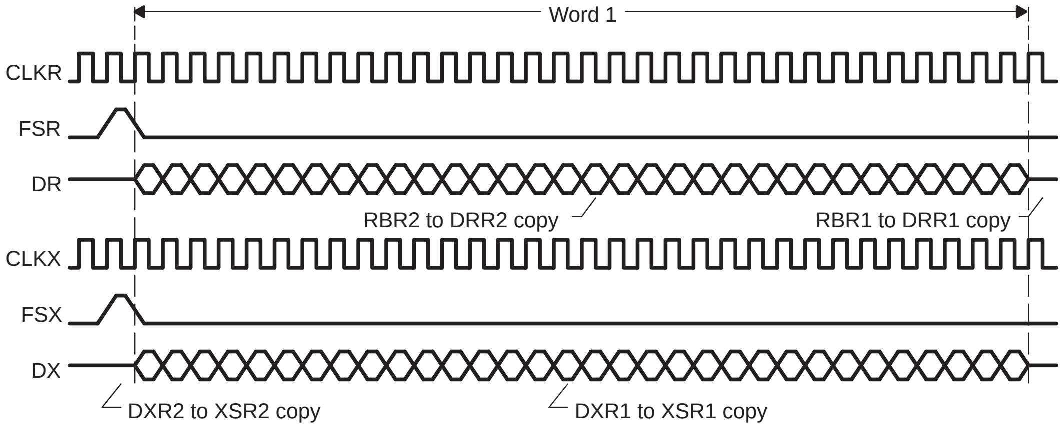
- ❑ (R/X)PHASE = 0: Single-phase frame
- ❑ (R/X)FRLEN1 = 0000000b: 1-word frame
- ❑ (R/X)WDLEN1 = 101b: 32-bit word

Two 16-bit data words are transferred to and from the McBSP by the DSP core and MPU core or system DMA or DSP DMA controller. Thus, two reads (from DRR2 and DRR1) and two writes (to DXR2 and DXR1) are necessary for each frame. This results in only half the number of transfers compared to the previous case. This manipulation reduces the percentage of bus time required for serial port data movement.

Note:

When the word length is larger than 16 bits, access DRR2/DXR2 before accessing DRR1/DXR1. McBSP activity is tied to accesses of DRR1/DXR1. During the reception of 24-bit or 32-bit words, read DRR2 and then read DRR1. Otherwise, the next RBR[1,2]-to-DRR[1,2] copy occurs before DRR2 is read. Similarly, during the transmission of 24-bit or 32-bit words, write to DXR2 and then write to DXR1. Otherwise, the next DXR[1,2]-to-XSR[1,2] copy occurs before DXR2 is loaded with new data.

Figure 70. One 32-Bit Data Word Transferred To/From the McBSP



2.15.2 Data Packing Using Word Length and the Frame-Synchronization Ignore Function

When there are multiple words per frame, you can implement data packing by increasing the word length (defining a serial word with more bits) and by ignoring frame-synchronization pulses. First, consider Figure 71, which shows the McBSP operating at the maximum packet frequency. Here, each frame only has a single 8-bit word. Notice the frame-synchronization pulse that initiates each frame transfer for reception and for transmission. For reception, this configuration requires one read operation for each word. For transmission, this configuration requires one write operation for each word.

Figure 71. 8-Bit Data Words Transferred at Maximum Packet Frequency

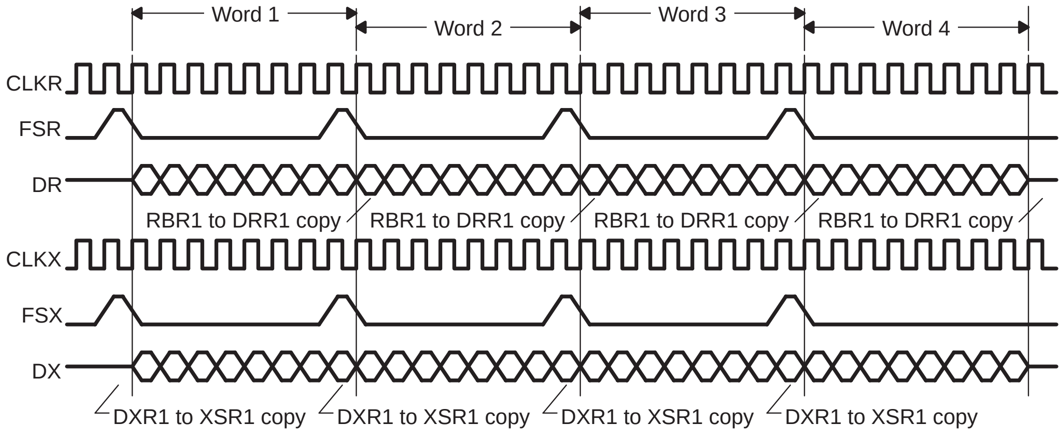
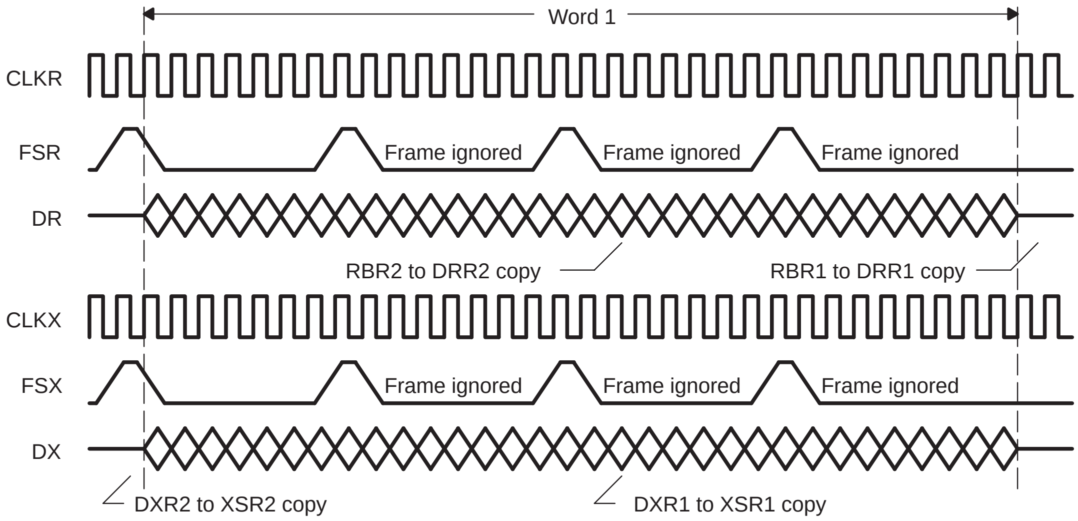


Figure 72 shows the McBSP configured to treat this data stream as a continuous 32-bit word. In this example, the McBSP responds to an initial frame-synchronization pulse. However, (R/X)FIG = 1, so that the McBSP ignores subsequent pulses. Only two read transfers or two write transfers are needed every 32 bits. This configuration effectively reduces the required bus bandwidth to half the bandwidth needed to transfer four 8-bit words.

Figure 72. Configuring the Data Stream as a Continuous 32-Bit Word



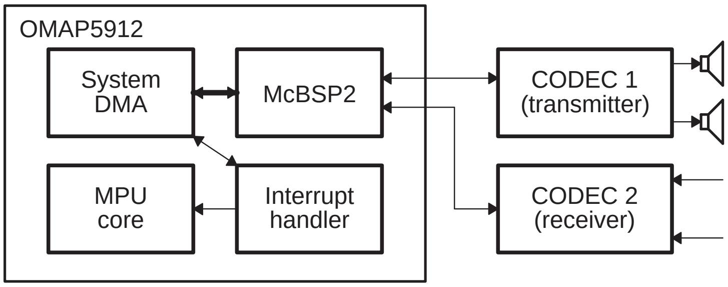
3 Interfacing McBSP2 to Two AIC33 Devices

The first supported usage case for McBSP2 is the ability to interface to two different codec devices operating at independent rates for transmit and receive. McBSP2 features transmit and receive clock and frame-synchronization pins that allow interface with external devices with independent rates. The following sections show how to interface McBSP2 to an AIC33 device using independent rates for transmit and receive.

3.1 Architectural/Operational Description

Figure 73 shows a block diagram of the architectural components involved in this usage case.

Figure 73. Architectural Block Diagram



In this example, McBSP2 transmit pins FSX, CLKX, and DX are connected to CODEC 1 and are operating at a 48 KHz sampling rate. The receive pins FSR, CLKR, and DR are connected to CODEC 2 and are operating at a 44.1 KHz sampling rate. Both codecs operate on two 16-bit elements per frame.

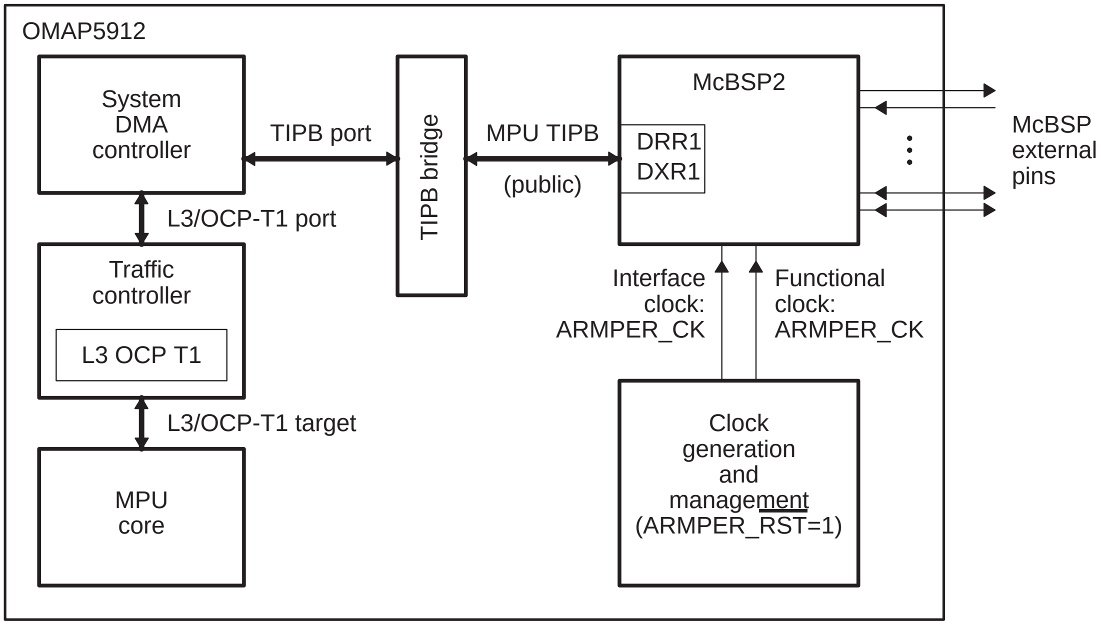
The codec used is the TLV320AIC33 low power stereo codec for portable audio/telephony. This codec can operate in I2S mode or DSP mode through an interface that requires a bit-clock, a frame synchronization, and data pins. There is a separate set of clock and frame-synchronization signals for the transmit and the receive zones. McBSP2 acts as the slave in this particular usage case.

McBSP2 is event-synchronized to the system DMA and the data is available to the MPU core through a ping-pong buffering system that operates through the use of the half-frame and end-of-block DMA MPU interrupts. The buffers are placed in the internal frame buffer memory block for the MPU (L3 OCP T1).

3.1.1 Data Path

Figure 74 shows the supported data path for this usage case.

Figure 74. Data Path



The system DMA uses McBSP2 event synchronization to move data to and from memory. McBSP2 flags the DMA with an event request when received data has arrived (or when it is ready to transmit data).

Upon data reception, the DMA receives the McBSP2 synchronization event (McBSP2 RX), indicating that data from the DRR1 register is ready to be serviced. This is accomplished by the system DMA, communicate with the serial port through the TIPB port. Communication with McBSP2 (MPU public peripheral) is performed through the TIPB bridge via the MPU TIPB public bus. Although this is a 32-bit wide bus, the system DMA can only perform 16-bit data accesses to the serial port, and thus it only utilizes the McBSP Data Receive Register 1 (DRR1). The system DMA then writes the data to the internal frame buffer via the L3/OCP-T1 port through the traffic controller. Upon completion of a half frame, the system DMA interrupts the MPU to indicate that half the frame has been received. The end-of-block interrupt also indicates that the full block has been transferred. Ping-pong buffering can be accomplished with these two interrupts by making the first half of the frame the ping buffer and the second half of the frame the pong buffer.

Upon transmitting data, the system DMA receives the McBSP2 transmit synchronization event (McBSP2 TX), indicating that DXR1 is ready to be written and transmit data. The system DMA then takes data from the internal memory buffer via the L3/OCP-T1 port through the Traffic Controller to the serial port via the TIPB port. Transmit data also is sent through the MPU TIPB public bus to McBSP2. The DMA interrupts the MPU core to indicate that a half-frame and a full-frame have been transmitted to the serial port.

For more details on the MPU TIPB public bus, TIPB bridge, and traffic controller, see *OMAP5912 Multimedia Processor OMAP 3.2 Subsystem Reference Guide* (SPRU749).

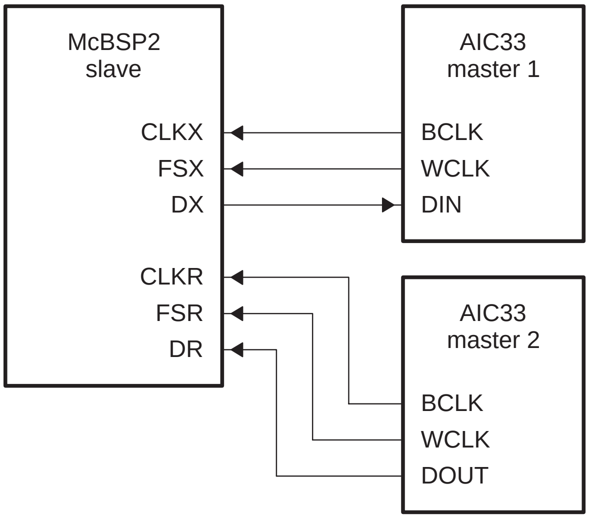
McBSP2 needs to have both the interface and functional clocks enabled for proper operation. Data movement is clocked on the MPU TIPB bus at the interface clock rate. The functional clock is the McBSP2 module clock. Both the interface and functional clock both are set to the ARMPER_CK for McBSP2. ARMPPER_$\overline{\text{RST}}$ needs to be de-asserted.

For more details on the interface/functional clocks and McBSP2 reset signals, see *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

3.2 Hardware Interface

Figure 75 shows the hardware interface between McBSP2 and the two stereo codecs.

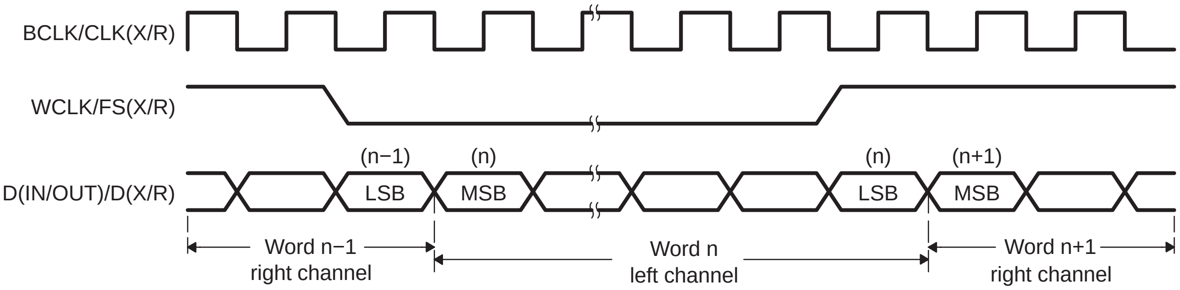
Figure 75. Hardware Interface



The AIC33 codecs act as masters, providing independent clock rates. McBSP2 is the slave for both receive and transmit.

The codecs are programmed to operate in I2S mode. Left channel data is valid on the second rising edge of the bit clock, after the falling edge of the word clock (McBSP2 frame synchronization). Right channel data is valid on the second rising edge of the bit clock after the rising edge of the word clock. Data is transmitted MSB first. Figure 76 shows the signal diagram for the I2S interface.

Figure 76. I2S Interface



From the McBSP perspective, data is handled as follows:

- Left channel data is output on the first falling edge of CLKX after the falling edge of FSX, and is sampled on the second rising edge of CLKR after the falling edge of FSR.
- Right channel data is output on the first falling edge of CLKX after the rising edge of FSX and is sampled on the second rising edge of CLKR after the rising edge of FSR.

This is accomplished by programming a data-delay of 1-bit with CLKRP/CLKXP=1 and FSRP/FSXP=1 clock/frame-synchronization polarities.

Note:

The I2S specification is flexible, so that receivers and transmitters do not have to agree on a word size. As data is transmitted MSB first and each new word is indicated by a transition in WCLK (FSR/X), the slave device can determine the appropriate word size by the master’s signal on-the-fly. However, when using the OMAP5912 device as the slave, this automatic word size detection is not possible.

All timing parameters need to be met for both device data sheets.

3.3 Software Configuration

In order to program McBSP2 as a slave I2S receiver and transmitter, the following registers must be programmed from the MPU/ARM side:

- ❑ Disable digital loopback mode in SPCR1
- ❑ Disable the clock stop mode in SPCR1
- ❑ Choose one phase for the receiver and transmitter in RCR2 and XCR2, respectively
- ❑ Set the receive and transmit frame length to 2 in the RCR1 and XCR1, respectively
- ❑ Set the receive and transmit word length to 16-bits in the RCR1 and XCR1, respectively
- ❑ Set the receive/transmit frame-synchronization ignore function to detect unexpected frame-synchronization pulses in the RCR2 and XCR2 respectively. This will set the RSYNCERR and XSYNCERR flags upon receiving unexpected frame-synchronization pulses.
- ❑ Set the receive and transmit data delay to 1-bit in RCR2 and XCR2, respectively
- ❑ Set the receive and transmit interrupt mode to generate an interrupt when a receive/transmit frame pulse is received in SPCR1 and SPCR2, respectively
- ❑ Set the receive and transmit frame-synchronization and clock polarity to 1 in PCR
  - ■ FSR/FSX is active low
  - ■ Receive data sampled on rising edge of CLKR
  - ■ Transmit data sampled on falling edge of CLKX
- ❑ Set the receive and transmit clock mode for CLKR/CLKX to operate as inputs in PCR (McBSP2 is a slave)
- ❑ Leave the sample rate generator registers at their default values and in reset, as they are not utilized in this example
- ❑ Leave all other register fields at their default values

For details on the proper McBSP initialization sequence, see section 2.11.

Program the DMA as follows:

- Receive Channel (any free channel)
  - Event synchronized to McBSP2 RX event
  - Source port: TIPB
  - Destination port: L3/OCP_T1
  - Source addressing mode: constant address
  - Destination addressing mode: post-increment address
  - Source address: McBSP2 DRR1 (MPU location 0xFFFB1002)
  - Destination address: array in internal frame buffer memory (L3 OCP T1)
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.
- Transmit Channel (any free channel)
  - Event synchronized to McBSP2 TX event
  - Source port: L3/OCP_T1
  - Destination port: TIPB
  - Source addressing mode: post-increment address
  - Destination addressing mode: constant address
  - Source address: array in internal frame buffer memory (L3 OCP T1)
  - Destination address: McBSP2 DXR1 (MPU location 0xFFFB1006)
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.

The DMA event handler can be used in its default state, as the events are mapped correctly coming out of reset. No remapping is necessary.

The interrupts used on this example are the following:

- DMA half-frame interrupt receive. Flags the reception completion of half the frame. This can be the ping buffer in memory.
- DMA end of block interrupt receive. Flags the reception completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- DMA half-frame interrupt transmit. Flags the transmission completion of half the frame. This can be the ping buffer in memory.
- DMA end of block interrupt transmit. Flags the transmission completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- McBSP TX interrupt. This is initially programmed to send an interrupt upon reception of the FSX signal. In this interrupt service routine, the transmitter must be released from reset when FSX is high, ensuring that transmission will begin on an FSX falling edge. See sections 2.2.3.3 and 2.2.3.4 for details.

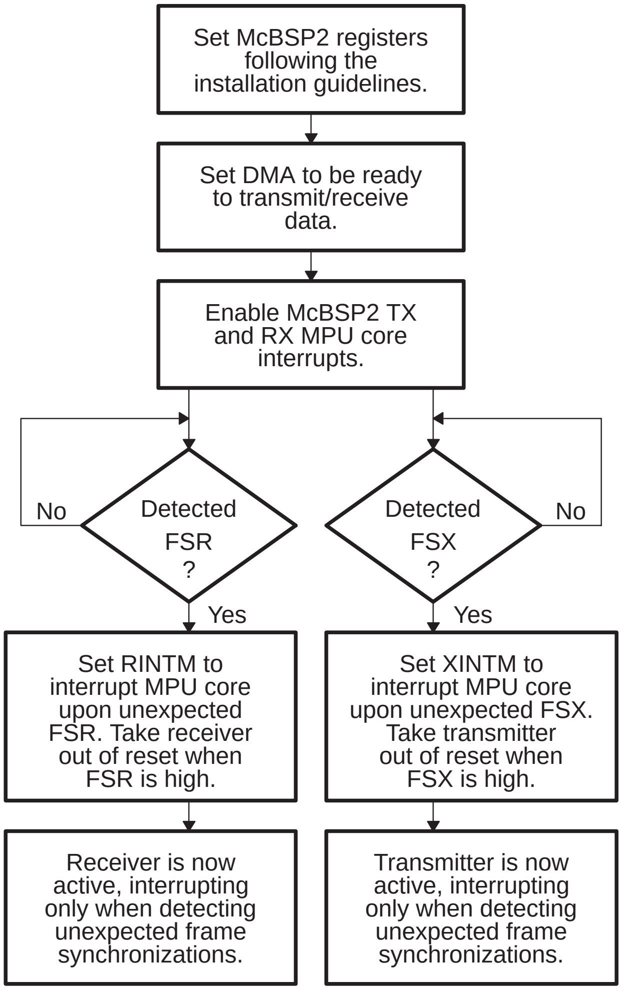
  Once the transmitter is taken out of reset in this ISR, XINTM may be programmed to detect an unexpected transmit frame synchronization.
- McBSP RX interrupt. This is initially programmed to send an interrupt upon reception of the FSR signal. In this interrupt service routine, the transmitter must be released from reset when FSR is high, ensuring that data reception will begin on an FSR falling edge. See sections 2.2.3.3 and 2.2.3.4 for details.

  Once the transmitter is taken out of reset in this ISR, RINTM may be programmed to detect an unexpected receive frame synchronization.

The interrupt handler default mapping is appropriate for this setup.

Figure 77 displays the software flow for this example.

Figure 77. Software Flow Chart


Set McBSP2 registers following the installation guidelines.
Set DMA to be ready to transmit/receive data.
Enable McBSP2 TX and RX MPU core interrupts.
No
Detected FSR ?
Detected FSX ?
No
Yes
Yes
Set RINTM to interrupt MPU core upon unexpected FSR. Take receiver out of reset when FSR is high.
Set XINTM to interrupt MPU core upon unexpected FSX. Take transmitter out of reset when FSX is high.
Receiver is now active, interrupting only when detecting unexpected frame synchronizations.
Transmitter is now active, interrupting only when detecting unexpected frame synchronizations.

Note that it is necessary to take the receiver/transmitter out of reset when the corresponding frame-synchronization is in the inactive state (*high* given FSRP/FSXP=1). To ensure this condition is true, you might need to poll FSR/X using the GPIO capability to wait for the frame-synchronization to reach the high state, and then immediately take the receiver/transmitter out of reset. This is only necessary when the McBSP is a slave and the frame-synchronization timing and polarity cause the receiver/transmitter to be taken out of reset while the frame-synchronization signal is in the active state, otherwise channels might be rotated. See sections 2.2.3.3 and 2.2.3.4 for more details.

3.4 System Traffic Considerations

This supported usage case for McBSP2 runs at 48 KHz sampling rate for the transmitter and at 44.1 KHz sampling rate for the receiver using 16-bit words for the two stereo data channels.

The McBSP2 can also be configured as a slave receiver and transmitter using the same software configuration, but at a sampling frequency of 96 KHz, when using an external device to provide the frame and clock source to both the serial port and the codec. The codec supports sampling rates up to 96KHz as a slave.

The traffic controller is running at 96 MHz, ARMPER_CK is running at 48MHz, and the TIPB strobe period is twice the traffic controller period. Furthermore, the McBSP2 data movement is the only activity on the bus.

Note that performance is affected by the number of requests to a given DMA port, traffic on the MPU public peripheral bus, and external memory access latency.

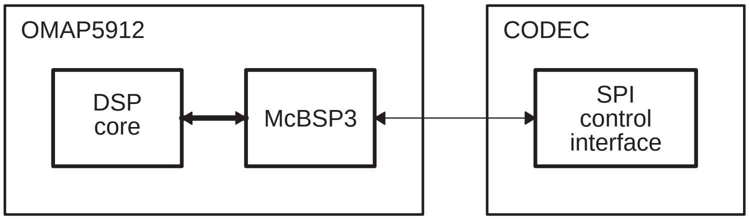
4 McBSP3 SPI Mode

The OMAP5912 McBSP may be configured as a SPI master or slave to communicate with other devices in a system. This section shows how to use one of the four clock stop mode configurations to control the TLV320AIC33 SPI-device using McBSP3. Given the pins available in this particular McBSP (CLKX, FSX, DX, and DR), it is a good candidate for use in clock stop mode to interface to the SPI device.

4.1 Architectural/Operational Description

The McBSP3 sends commands to write to the internal registers of the AIC33 device via the SPI interface. It writes to the internal registers of the codec to configure the device, and it also reads the values back to confirm the write takes place.

Figure 78. Architectural Block Diagram for McBSP3 SPI Mode

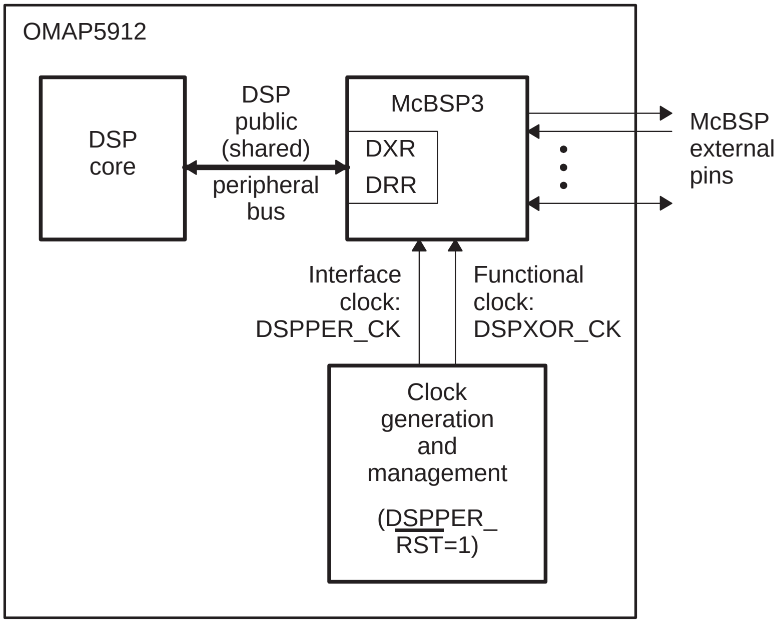


The method used is XRDY/RRDY polling by the DSP core. XRDY/RRDY is set every time the McBSP has received data or is ready to transmit data, respectively. The DSP core writes a value to the codec control register and then reads it back to ensure the write was performed correctly. McBSP3 is the master in this usage case.

4.1.1 Data Path

Figure 79 shows the data path supported for this usage case.

Figure 79. Data Path


OMAP5912
DSP core
DSP public (shared) peripheral bus
McBSP3
DXR
DRR
McBSP external pins
Interface clock: DSPPER_CK
Functional clock: DSPXOR_CK
Clock generation and management
(DSPPER_RST=1)

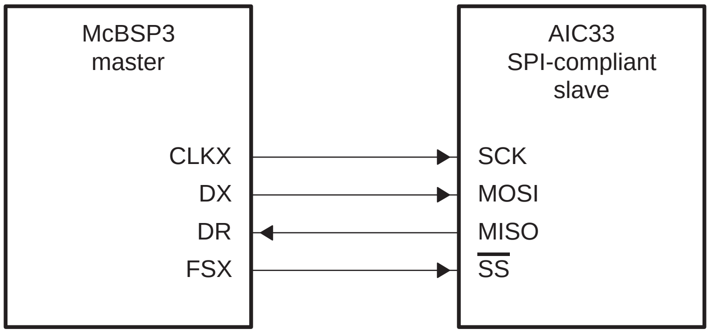
The DSP public peripheral bus (DSP TIPB bus) is a 16-bit bus that interfaces the DSP to the DSP public peripherals. Data is accessed to or from McBSP3 by the DSP core via this data path every time the McBSP is ready to transmit or data has been received through polling the XRDY/RRDY bits in the SPCR2 and SPCR1 registers, respectively.

Note that McBSP3 has to be set to operate in the 4-pins mode, which uses DSPPER_CK as the interface clock and FSX/CLKX as bidirectional pins. See section 1.3.3 for more details.

4.2 Hardware Interface

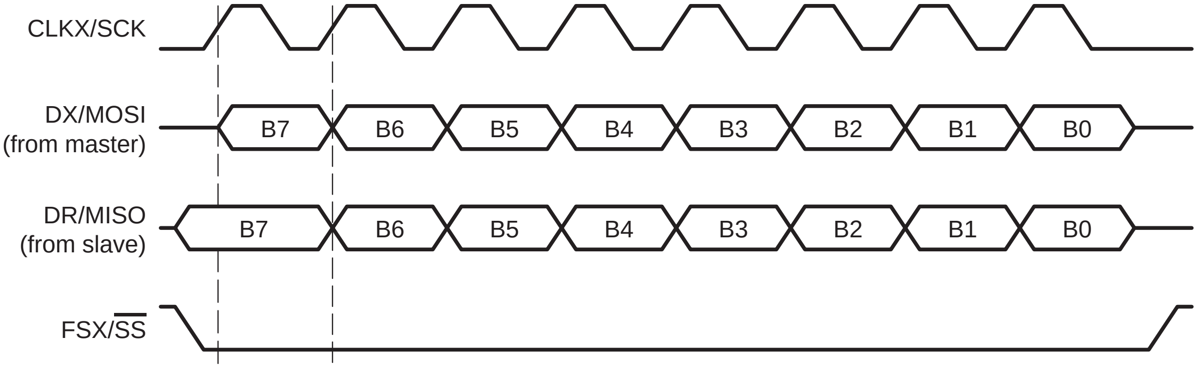
Figure 80 shows the hardware interface using McBSP3 in clock stop mode to communicate with the SPI-compliant slave.

Figure 80. McBSP3 as Master for AIC33 Control



The SPI timing interface is achieved using clock stop mode 10b, with CLKXP = CLKRP = 0, as shown in Figure 81.

Figure 81. SPI Transfer With No Clock Delay (CLKSTP=10b, CLKXP=0, CLKRP=0), 8-Bit Word Example

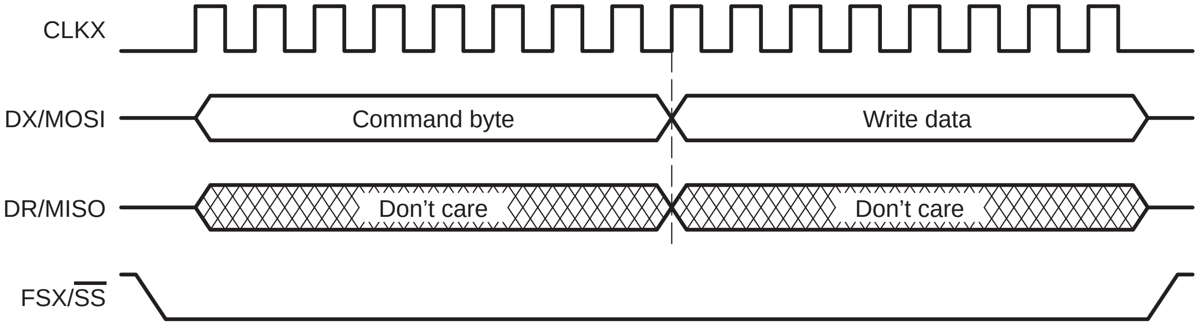


Data DX (MOSI) is clocked out of McBSP3 on the rising edge of CLKX (SCK). Received data DR (MISO) is sampled on the falling edge of CLKX (SCK). Data transmission and reception are only possible when $\overline{\text{SS}}$ is low on the SPI compliant device. For more details on the clock stop modes, see section 2.4.2.

The TLV320AIC33 is controlled by registers. Reading and writing these registers is accomplished by the use of an 8-bit command, which is sent to the MOSI pin of the device prior to the data for that register. The command word contains 7 bits that specify the register address from 0 to 127 (decimal), followed by an R/W bit, which specifies the direction of data flow on the serial bus.

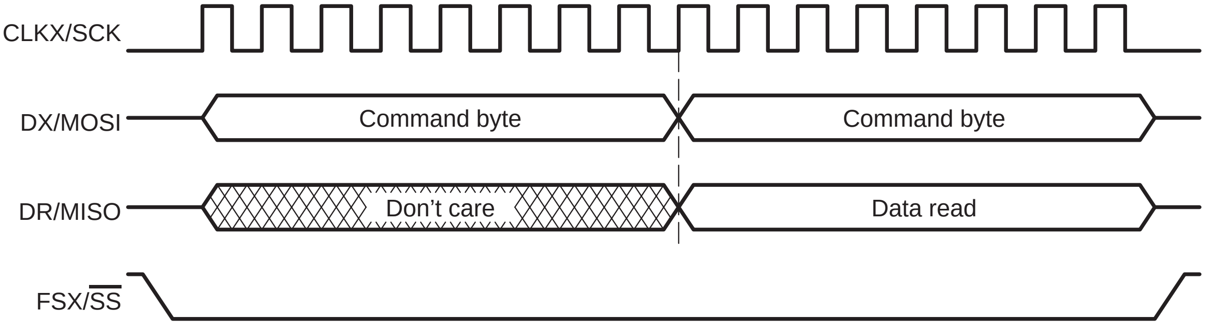
In the case of a register write, a second byte of data is sent to the MOSI pin and contains the data to be written to the register. Both the command and data bytes need to be sent while holding the $\overline{SS}$ line low. Figure 82 shows an example of a McBSP 16-bit access that performs an AIC33 register write operation. Note that the McBSP needs to ignore the 8 most significant bits of the 16-bit data read from DRR1, because the least significant 8 bits contain valid data.

Figure 82. Write Example



Reading of registers is accomplished in similar fashion. The 8-bit command word sends the 7-bit register address, followed by the R/W bit = 0 to signify a register read is occurring. The 8-bit register data is then clocked out of the MISO pin during the second 8 SCLK clocks in the frame. Figure 83 shows an example of a McBSP 16-bit access that performs an AIC33 register read operation. Although the McBSP only writes data, the receiver will simultaneously perform a data read so that the data in DRR1 can be discarded.

Figure 83. Read Example



4.3 Software Configuration

To configure McBSP3 to operate in this clock stop mode, program the following from the DSP side:

- ❑ Set the clock stop mode with clock delay to comply with the timing requirements of the AIC33 device SPI mode
  - ■ CLKSTP = 10b
  - ■ CLKXP=0
  - ■ CLKRP=0
- ❑ Configure the CLKX pin as an output pin driven by the internal sample generator
  - ■ CLKXM=1
- ❑ The clock generated by the sample rate generator (CLKG) is derived by the internal clock source
  - ■ SCLKME=0
  - ■ CLKSM=1
- ❑ Choose the divide-down value for CLKG to comply with the AIC33 SPI mode clocking requirements
  - ■ CLKGD value depends on the frequency of the internal clock source. For example, if the internal clock source equals 48MHz, a value of CLKGDV=11 (divided by 12) provides a CLKX of 4MHz.
- ❑ Configure the FSX pin as an output driven according to the FSGM bit
  - ■ FSXM=1
- ❑ Set the transmitter to drive the frame-synchronization pulse on the FSX pin every time data is transferred from DXR1 to SXR1
  - ■ FSGM=0
- ❑ Set the FSX pin as active low
  - ■ FSXP=1
- ❑ Provide the correct setup time of 1-bit delay on the FSX signal
  - ■ X/RDATDLY=01b
- ❑ Disable digital loopback mode
  - ■ DLB=0

- ❑ Configure pins to behave as McBSP pins, not as GPIO pins
  - ■ X/RIOEN=0
- ❑ Use single-phase only for SPI mode
  - ■ X/RPHASE=0
- ❑ Use a transmit word length specific for the AIC33
  - ■ X/RWDLEN1=010b (16-bit words)
- ❑ Set the transmit frame length to 1
  - ■ X/RFRLEN1=000 0000b
- ❑ Leave the remaining fields at their default values

Table 8 shows a generic summary of the McBSP software configuration for clock stop mode.

Also, the proper initialization procedure outlined in section 2.4.5 must be followed to initialize the McBSP in clock stop mode.

Once the McBSP has been initialized and taken out of reset, the DSP core can transmit data once XRDY=1 by writing to the DXR1 register (for 16-bit transfers), and receive data from the DRR1 register (for 16-bit transfers) once RRDY=1. The DSP core polls the state of the XRDY and RRDY bits.

4.4 System Traffic Considerations

This particular example is run with a CLKX frequency of 4MHz, using an interface clock (DSPPER_CK) of 96 MHz and a functional clock (DSPXOR_CK) of 12 MHz. This example was run with no other activity on the peripheral bus.

Furthermore, note that performance is affected by the number of requests to a given DMA port (e.g., the TIPB port), traffic on the DSP public peripheral bus, external memory access latency, and conflicting accesses by the DMA and core to the same internal memory blocks.

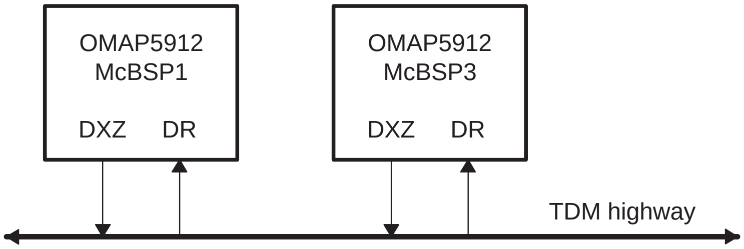
5 Interfacing McBSP1 to Another Multichannel Device

The McBSP is capable of supporting multichannel mode using the DXZ pin. The DXZ pin is capable of tri-stating when a particular channel in a time division multiplexed (TDM) data highway is enabled and masked or disabled. The following sections show how to interface McBSP1 to another multichannel device (McBSP3) using eight partition mode.

5.1 Architectural/Operational Description

Figure 84 shows a block diagram of the architectural components involved in this usage case.

Figure 84. Functional Block Diagram



Multiple multichannel devices or other DSPs can be connected via a single TDM bus to communicate specific information to a specific device or group of devices. Typical applications include messaging, broadcasting, and passing channels of processed data to a given device or devices. For the OMAP5912 McBSP, TDM transfers are achieved using the McBSP and the DMA. The serial port pins DXZ and DR are used for data transfer, whereas CLK(R/X) and FS(R/X) serve as the control signals for clocking and synchronization.

It is important to configure each device to send and receive channels of data on certain assigned time slots. As no address lines are used, the multichannel operation of the McBSP helps identify the destination of data in transit. Each McBSP is programmed to transmit data on certain time slots, and to listen to specific time slots.

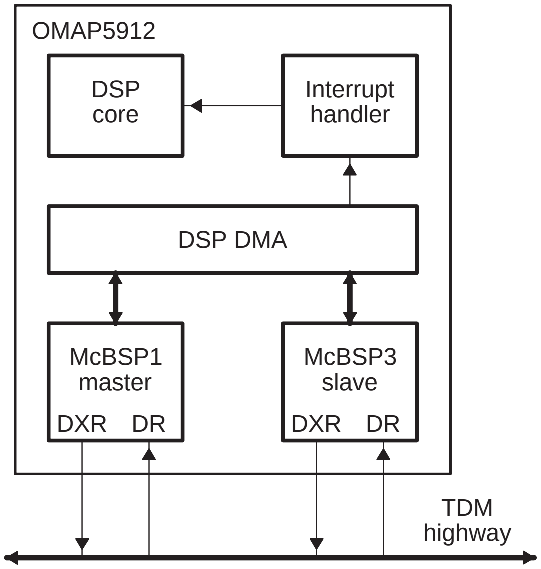
This particular example is implemented using McBSP1 and McBSP3 for one OMAP5912 device, but it can be expanded to include other multichannel devices to the TDM highway.

Note:

Caution must be taken to ensure the driver strength of the multichannel devices used is strong enough to drive the load attached. If necessary, use the external signal drivers for the particular type and number of loads.

Figure 85 shows the architectural block diagram used in this example.

Figure 85. Architectural Block Diagram



Two 16-bit words are transferred by each McBSP in the TDM highway. Each McBSP has the capability to read any of the total four channels on the line. This example shows the configuration that allows McBSP1 to transmit on channels 0 and 1, while McBSP3 transmits on channels 3 and 4. McBSP1 then receives channels 3 and 4, while McBSP3 receives channels 0 and 1. When other multichannel devices are connected to the highway, each multichannel device can listen to any of the channels being transmitted by any of the devices on the line.

To prevent contention on the data line, a dummy channel needs to be inserted between device transfer transitions. These extra channels are shown in Table 82, and further explained in section 5.2.

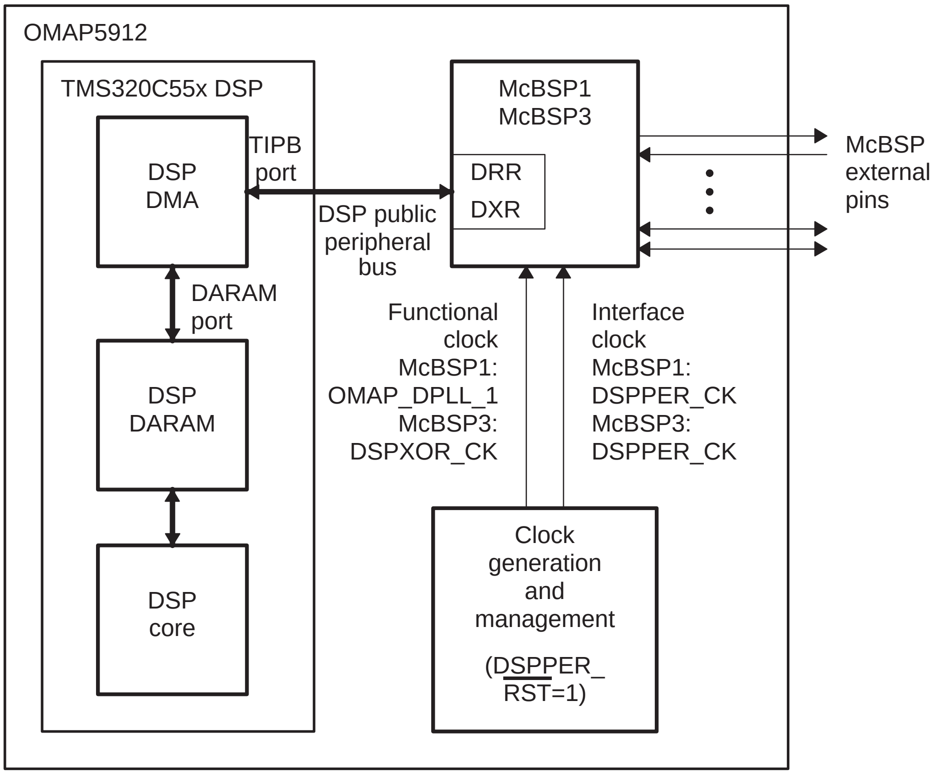
Table 82. Channel States for McBSP1 and McBSP3

Channel	Transmit State	Receive State
	McBSP1	
Channel 0	Enabled, unmasked (transmits data, generates DMA event)	Disabled (does not generate DMA event)
Channel 1	Enabled, unmasked (transmits data, generates DMA event)	Disabled (does not generate DMA event)
Channel 2 (dummy channel)	Disabled (tri-stated, does not generate DMA event)	Disabled (does not generate DMA event)
Channel 3	Disabled (tri-stated, does not generate DMA event)	Enabled (receives data, generates DMA event)
Channel 4	Disabled (tri-stated, does not generate DMA event)	Enabled (receives data, generates DMA event)
Channel 5 (dummy channel)	Disabled (tri-stated, does not generate DMA event)	Disabled (does not generate DMA event)
	McBSP3	
Channel 0	Disabled (tri-stated, does not generate DMA event)	Enabled (receives data, generates DMA event)
Channel 1	Disabled (tri-stated, does not generate DMA event)	Enabled (receives data, generates DMA event)
Channel 2 (dummy channel)	Disabled (tri-stated, does not generate DMA event)	Disabled (does not generate DMA event)
Channel 3	Enabled, unmasked (transmits data, generates DMA event)	Disabled (does not generate DMA event)
Channel 4	Enabled, unmasked (transmits data, generates DMA event)	Disabled (does not generate DMA event)
Channel 5 (dummy channel)	Disabled (tri-stated, does not generate DMA event)	Disabled (does not generate DMA event)

5.1.1 Data Path

Figure 86 shows the supported data path for this usage case.

Figure 86. Data Path



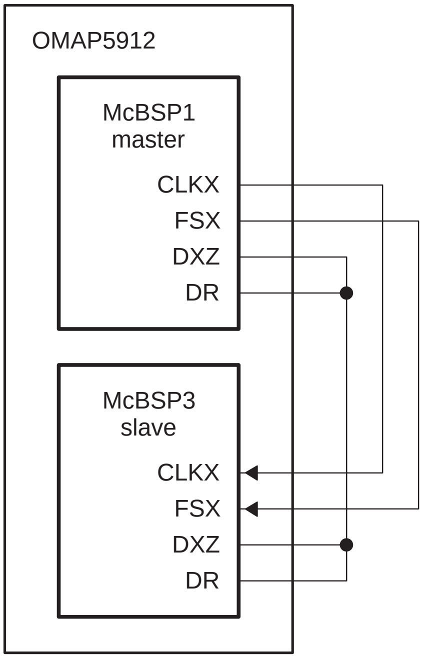
The McBSPs will generate DMA events for the channels that are enabled and unmasked, in this example. Once a DMA event is generated, the DSP DMA services the serial port via the TIPB port and reads (or writes) data from/to the internal DSP DARAM. For more details on the DSP TIPB bus and DSP DMA, see the *OMAP5910/5912 Multimedia Processor DSP Subsystem Reference Guide* (SPRU890).

McBSP1 and McBSP3 need to have both the interface and functional clocks enabled for proper operation. Data movement is clocked on the DSP TIPB bus at the interface clock rate. The functional clock is the McBSP module clock. McBSP1 has the functional clock set to the OMAP_DPLL_1 clock and the interface clock to DSPXOR_CK; McBSP3 has both the interface and functional clock set to the DSPPER_CK. DSPPER_$\overline{\text{RST}}$ needs to be de-asserted. For more details on interface/functional clocks and McBSP2 reset signals, see *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

5.2 Hardware Interface

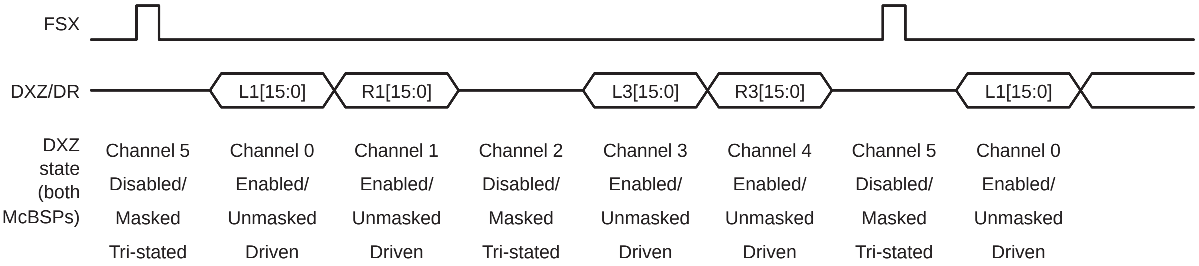
Figure 87 shows the hardware interface between McBSP1 and McBSP3 in multichannel mode.

Figure 87. Hardware Interface



McBSP1 is the master device, providing the CLKX and FSX signals. Figure 88 shows the timing relationship on the data line. Note that the dummy channel inserts dead time between McBSP1 and McBSP3 data.

Figure 88. Multichannel Timing Relationship



DXZ state (both McBSPs)	Channel 5	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 0
	Disabled/	Enabled/	Enabled/	Disabled/	Enabled/	Enabled/	Disabled/	Enabled/
	Masked	Unmasked	Unmasked	Masked	Unmasked	Unmasked	Masked	Unmasked
	Tri-stated	Driven	Driven	Tri-stated	Driven	Driven	Tri-stated	Driven

L1 and R1 correspond to the two 16-bit data elements transmitted by McBSP1. Similarly, L3 and R3 correspond to the two 16-bit data elements transmitted by McBSP3. The frame length is programmed to the length of highest numbered channel, Channel 5 in this case.

5.3 Software Configuration

The software configuration for McBSP1 and McBSP3 mainly depends on the fact that McBSP1 is the master device, and therefore the sample rate generator must generate the clock and frame signal internally. Furthermore, each McBSP can be programmed to read and write to different channels in the TDM bus from different devices.

5.3.1 McBSP1 Configuration

In order to program McBSP1 as a master transmitter (slave receiver only due to internal signal routing) in multichannel mode, configure the following registers from the DSP side:

- Disable digital loopback mode in SPCR1
- Disable the clock stop mode in SPCR1
- Enable the receive and transmit multichannel selection mode:
  - RMCM = 1, channels can be individually enabled or disabled, in the MCR1
  - XMCM = 01b, all channels are disabled unless they are selected in the appropriate transmit channel enable registers (XCERs). If a channel is enabled, a channel in this mode is also unmasked.
  - RMCME = XMCME = 1, to select 8-partition mode in the MCR1 and MCR2 registers
  - Enable registers
    - RCERA = 0x0018, to enable channels 3 and 4
    - XCERA = 0x0003, to enable channels 0 and 1
- Choose one phase for the receiver and transmitter in RCR2 and XCR2, respectively
- Set the receive and transmit word length to 16 bits in the RCR1 and XCR1, respectively
- Set the receive and transmit frame length to 6 in the RCR1 and XCR1, respectively, to accommodate up to the highest numbered channel
- Set the receive/transmit frame-synchronization ignore function to detect unexpected frame-synchronization pulses in the RCR2 and XCR2 respectively. This will set the RSYNCERR and XSYNCERR flags upon receiving unexpected frame synchronization pulses.

- Set the receive and transmit data delay to 2-bits in RCR2 and XCR2, respectively
- Set the receive and transmit interrupt mode to generate an interrupt when an unexpected frame pulse is detected in SPCR1 and SPCR2, respectively
- Frame synchronization mode set up
  - Set the transmit frame-synchronization mode to be supplied internally by the McBSP (McBSP1 is the master transmitter)
  - Set the receive frame-synchronization mode to be supplied by an external source via the FSX pin (McBSP1 FSR signal is internally connected to the FSX pin)
- Clear the receive and transmit frame-synchronization and clock polarity in PCR
  - FSR/FSX is active high
  - Receive data sampled on falling edge of CLKR
  - Transmit data sampled on rising edge of CLKX
- Set the receive clock mode for CLKR to operate as an input in PCR (McBSP1 can only be a slave receiver)
- Set the transmit clock mode for CLKX to operate as an output in PCR (McBSP1 is a master transmitter)
- Configure the sample rate generator
  - Set the frame-synchronization period to 96 bit clocks per frame to accommodate 6 channels x 16-bits/channel
  - Set the frame-synchronization pulse width to 1 bit
  - Set the transmit clock mode to drive the CLKX pin by the sample rate generator in the PCR register
  - Set the SRG clock divide-down value to the desired value to accomplish the appropriate sampling rate. For example, if the internal clock source is running at 48MHz, set CLKGDV to 24 (divided by 25) to obtain a CLKX frequency of 1.92MHz and a sampling rate of 20KHz (6 channels × 16-bits/channel for every frame).
  - Set the SRG clock mode (chose an input clock to the SRG) by setting SCLKME=0, CLKSM=1 in PCR and SRGR1, respectively, to choose the internal clock source as input source
- Leave all other register fields at their default values

For details on the proper McBSP initialization sequence, see section 2.11.

Program the DSP DMA channels that service McBSP1 as follows:

- ❑ Receive Channel (any free channel)
  - Event synchronized to McBSP1 RX event
  - Source port: TIPB
  - Destination port: DARAM
  - Source addressing mode: constant address
  - Destination addressing mode: post-increment address
  - Source address: McBSP1 DRR1
  - Destination address: array in internal DARAM
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.
- ❑ Transmit Channel (any free channel)
  - Event synchronized to McBSP1 TX event
  - Source port: DARAM
  - Destination port: TIPB
  - Source addressing mode: post-increment address
  - Destination addressing mode: constant address
  - Source address: array in internal DARAM
  - Destination address: McBSP1 DXR1
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.

The DMA event handler can be used in its default state, because the events are mapped appropriately coming out of reset and no remapping is necessary.

The McBSP1 interrupts used for this example include:

- ❑ DMA half-frame interrupt receive. Flags the reception completion of half the frame. This can be the ping buffer in memory.
- ❑ DMA end of block interrupt receive. Flags the reception completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- ❑ DMA half-frame interrupt transmit. Flags the transmission completion of half the frame. This can be the ping buffer in memory.
- ❑ DMA end of block interrupt transmit. Flags the transmission completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- ❑ McBSP TX interrupt. This interrupt is optionally used to detect unexpected frame synchronization pulses.
- ❑ McBSP RX interrupt. This interrupt is optionally used to detect unexpected frame synchronization pulses.

The interrupt handler default mapping is appropriate for this setup.

5.3.2 McBSP3 Configuration

To program McBSP3 as a slave transmitter (slave receiver only due to internal signal routing) in multichannel mode, configure the following registers from the DSP side:

- ❑ Disable digital loopback mode in SPCR1
- ❑ Disable the clock stop mode in SPCR1
- ❑ Enable the receive and transmit multichannel selection mode:
  - ■ RMCM = 1, channels can be individually enabled or disabled, in the MCR1
  - ■ XMCM = 01b, all channels are disabled unless they are selected in the appropriate transmit channel enable registers (XCERs). If a channel is enabled, a channel in this mode is also unmasked.
  - ■ RMCME = XMCME = 1, to select 8-partition mode in the MCR1 and MCR2 registers
  - ■ Enable registers
    - ▪ RCERA = 0x0003, to enable channels 0 and 1
    - ▪ XCERA = 0x0018, to enable channels 3 and 4

❑ Choose one phase for the receiver and transmitter in RCR2 and XCR2, respectively

❑ Set the receive and transmit word length to 16 bits in the RCR1 and XCR1, respectively

❑ Set the receive and transmit frame length to 6 in the RCR1 and XCR1, respectively, to accommodate up to the highest numbered channel

❑ Set the receive/transmit frame-synchronization ignore function to detect unexpected frame-synchronization pulses in the RCR2 and XCR2 respectively. This will set the RSYNCERR and XSYNCERR flags upon receiving unexpected frame synchronization pulses.

❑ Set the receive and transmit data delay to 2-bits in RCR2 and XCR2, respectively

❑ Set the receive and transmit interrupt mode to generate an interrupt when a frame pulse is detected in SPCR1 and SPCR2, respectively. This configuration will allow the receiver and transmitter to be taken out of reset at a known FSR/FSX state. See sections 2.2.3.3 and 2.2.3.4 for more details.

❑ Frame synchronization mode set up

  ■ Set the transmit frame-synchronization mode to be supplied externally (McBSP3 is the slave transmitter)

  ■ Set the receive frame-synchronization mode to be supplied by an external source via the FSX pin (McBSP3 FSR signal is internally connected to the FSX pin)

❑ Set the receive and transmit frame-synchronization and clock polarity to 0 in PCR

  ■ FSR/FSX is active high

  ■ Receive data sampled on falling edge of CLKR

  ■ Transmit data sampled on rising edge of CLKX

❑ Set the receive clock mode for CLKR to operate as an input in PCR (McBSP3 can only be a slave receiver)

❑ Set the transmit clock mode for CLKX to operate as an input in PCR (McBSP3 is a slave transmitter)

❑ Configure the sample rate generator

  ■ The SRG is not used for this device because it is a slave

❑ Leave all other register fields at their default values

For details on the proper McBSP initialization sequence, see section 2.11.

Program the DSP DMA channels that service McBSP3 as follows:

- ❑ Receive Channel (any free channel)
  - Event synchronized to McBSP3 RX event
  - Source port: TIPB
  - Destination port: DARAM
  - Source addressing mode: constant address
  - Destination addressing mode: post-increment address
  - Source address: McBSP3 DRR1
  - Destination address: array in internal DARAM
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.
- ❑ Transmit Channel (any free channel)
  - Event synchronized to McBSP3 TX event
  - Source port: DARAM
  - Destination port: TIPB
  - Source addressing mode: post-increment address
  - Destination addressing mode: constant address
  - Source address: array in internal DARAM
  - Destination address: McBSP3 DXR1
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.

The DMA event handler can be used in its default state, because the events are mapped appropriately coming out of reset and no remapping is necessary.

The interrupts used on this example are the following:

- DMA half-frame interrupt receive. Flags the reception completion of half the frame. This can be the ping buffer in memory.
- DMA end of block interrupt receive. Flags the reception completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- DMA half-frame interrupt transmit. Flags the transmission completion of half the frame. This can be the ping buffer in memory.
- DMA end of block interrupt transmit. Flags the transmission completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- McBSP TX interrupt. This is initially programmed to send an interrupt upon reception of the FSX signal. In this interrupt service routine, the transmitter must be released from reset when FSX is high, ensuring that transmission will begin on an FSX rising edge. See sections 2.2.3.3 and 2.2.3.4 for details.

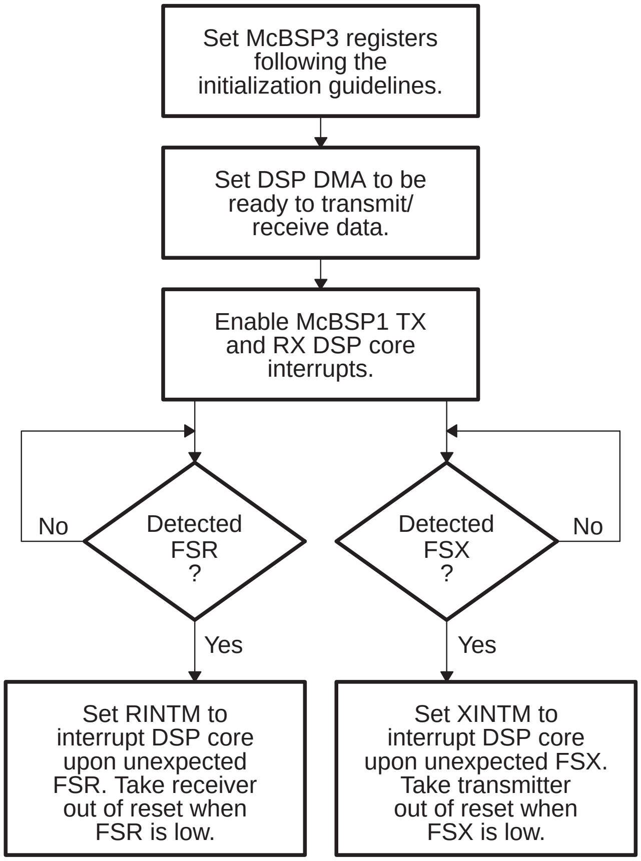
  Once the transmitter is taken out of reset in this ISR, XINTM may be programmed to detect an unexpected transmit frame synchronization.
- McBSP RX interrupt. This is initially programmed to send an interrupt upon reception of the FSR signal. In this interrupt service routine, the transmitter must be released from reset when FSR is high, ensuring that data reception will begin on an FSR rising edge. See sections 2.2.3.3 and 2.2.3.4 for details.

  Once the transmitter is taken out of reset in this ISR, RINTM may be programmed to detect an unexpected receive frame synchronization.

The interrupt handler default mapping is appropriate for this setup.

Figure 89 details the software flow used for the slave device, McBSP3.

Figure 89. McBSP3 Slave Transmitter Flow Diagram


Set McBSP3 registers following the initialization guidelines.
Set DSP DMA to be ready to transmit/ receive data.
Enable McBSP1 TX and RX DSP core interrupts.
Detected FSR ?
No
Yes
Detected FSX ?
No
Yes
Set RINTM to interrupt DSP core upon unexpected FSR. Take receiver out of reset when FSR is low.
Set XINTM to interrupt DSP core upon unexpected FSX. Take transmitter out of reset when FSX is low.

The receiver/transmitter must be taken out of reset when the frame-synchronization signal is low, given that FSXP/FSRP=0, thus ensuring data transmission/reception begins on the first rising edge of FSX/FSR. See sections 2.2.3.3 and 2.2.3.4 for more details.

5.4 System Traffic Considerations

This particular example is run with the following:

- ❑ McBSP1
  - ■ Functional clock: OMAP_DPLL_1 running at 192MHz
  - ■ Interface clock: DSPPER_CK running at 96MHz
- ❑ McBSP3
  - ■ Functional clock: DSPXOR_CK running at 12MHz
  - ■ Interface clock: DSPPER_CK running at 96MHz

The sampling frequency chosen for this example is 40 KHz with a 3.84Mhz bit-clock. It uses four DMA channels to service McBSP1 and McBSP3 transmit and receive events, with two TDM enabled channels each.

Note that performance is affected by the number of requests to a given DMA port (e.g., TIPB port), traffic on the DSP public peripheral bus, external memory access latency, and conflicting accesses by the DMA and core to the same internal memory blocks.

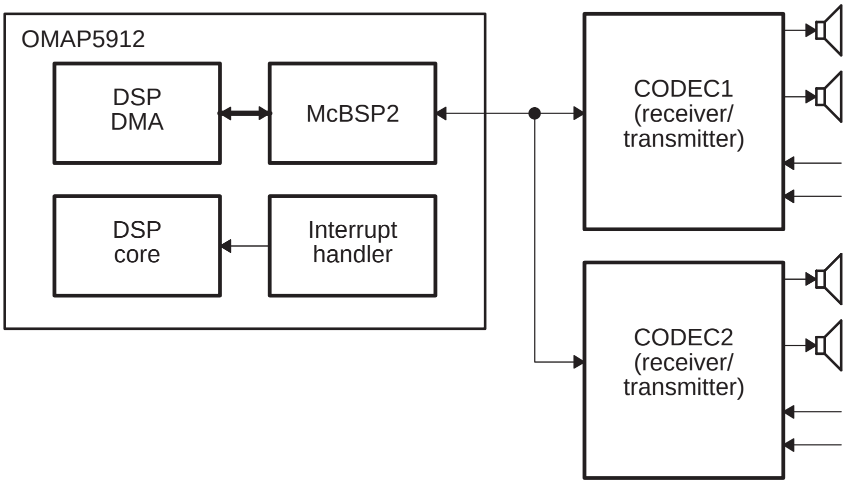
6 Interfacing McBSP2 to Two Codecs With The DSP

McBSP2 can be accessed by the DSP subsystem as shown in this section. In this specific configuration, McBSP2 is connected to two stereo codecs receiving and transmitting data.

6.1 Architectural/Operational Description

Figure 90 shows a block diagram of the architectural components involved in this usage case.

Figure 90. Architectural Description


OMAP5912
DSP DMA
McBSP2
DSP core
Interrupt handler
CODEC1 (receiver/ transmitter)
CODEC2 (receiver/ transmitter)

In this example, McBSP2 acts as the master transmitter and as a slave receiver, with the AIC33 stereo codecs operating in time-division multiplexed mode. Both codecs are receiving and transmitting stereo (2-channel) data.

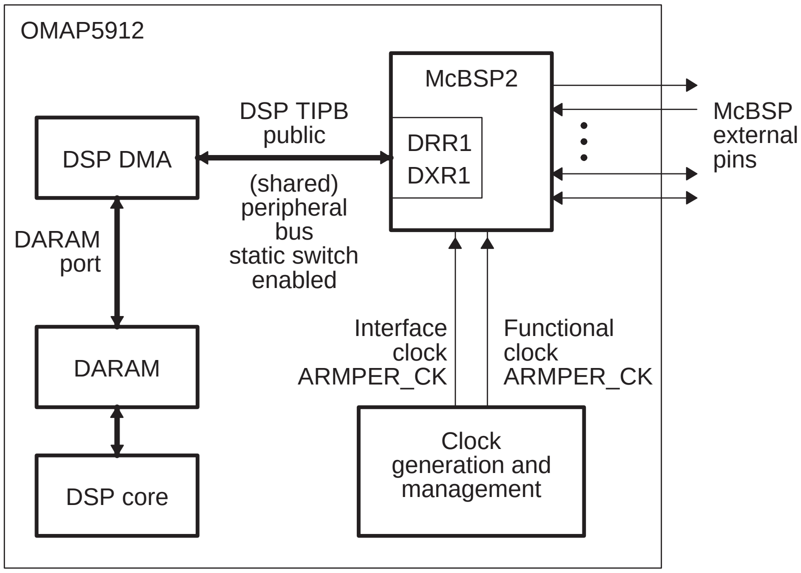
McBSP2 is serviced by the DSP DMA, and the DSP core is interrupted by the DMA once the received data is available for processing or or if it has been notified that the transmit data has been sent by the serial port. The data is stored in internal DARAM memory in the DSP subsystem. Two DMA channels are used in this example, one for transmission and one for reception.

Using the codec's programmable data delay, the first two channels are transmitted upon the detection of a frame-synchronization pulse. The second codec is programmed to transmit the data at a data delay of two stereo words, thus delivering the data immediate after the first codec sends the first two data channels. Therefore, only one McBSP data pin is needed (DX). A similar codec configuration is available for codec data reception.

6.1.1 Data Path

Figure 91 shows the DSP DMA data path for McBSP2.

Figure 91. Data Path


OMAP5912
McBSP2
DRR1
DXR1
McBSP external pins
DSP TIPB public
(shared) peripheral bus static switch enabled
DSP DMA
DARAM port
DARAM
DSP core
Interface clock ARMPER_CK
Functional clock ARMPER_CK
Clock generation and management

The DSP DMA is event synchronized to McBSP2 transfer and receive events (McBSP2 TX/ McBSP2 RX) by mapping the McBSP2 events through the DSP GDMA handler. When a data word has been received by the McBSP2, it sends an event request to the DMA event handler. The data is moved to the DSP DARAM via the DSP DMA DARAM port and from the DSP TIPB (peripheral) bus. The DSP core then has access to the DARAM, where it can be efficiently processed. The data is managed through a ping-pong buffer scheme, by auto-initializing the DSP DMA after each block is transferred.

In order to use McBSP2 (an MPU public peripheral) from the DSP side, the TI peripheral bus switch must be configured to allow DSP access.

For more details on the MPU TIPB public bus, TIPB bridge, and MPUI, see *OMAP5912 Multimedia Processor OMAP3.2 Subsystem Reference Guide* (SPRU749). For details on the DSP DMA and DSP subsystem, see *OMAP5910/5912 Multimedia Processor DSP Subsystem Reference Guide* (SPRU890).

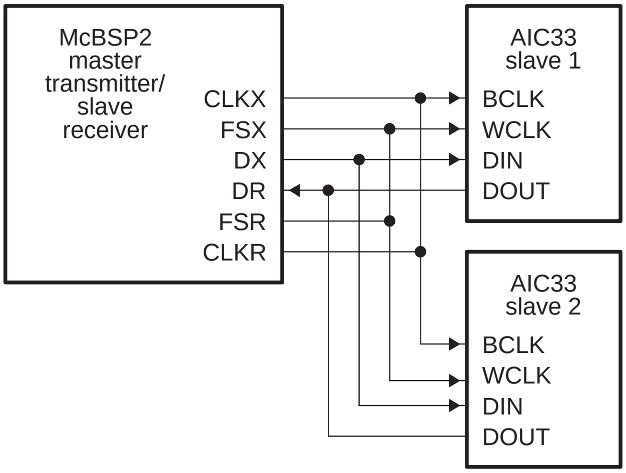
McBSP2 needs to have both the interface and functional clocks enabled for proper operation. Data movement is clocked on the MPU TIPB bus at the interface clock rate. The functional clock is the McBSP2 module clock. Both the interface and functional clock both are set to the ARMPER_CK for McBSP2. ARMPPER_$\overline{\text{RST}}$ needs to be de-asserted.

For more details on the interface/functional clocks and McBSP2 reset signals, see *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

6.2 Hardware Interface

Figure 92 shows the hardware interface between McBSP2 and the two AIC33 devices.

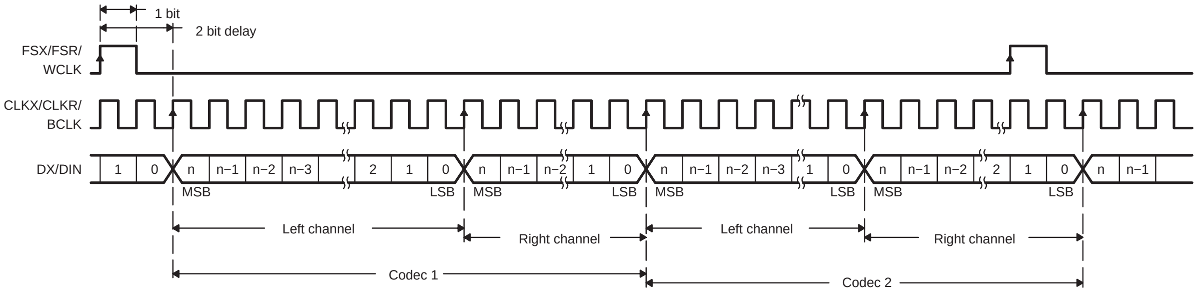
Figure 92. Hardware interface



The McBSP2 transmitter acts as the master, providing both the bit-clock and frame synchronization signals, CLKX and FSX, respectively. The McBSP2 receiver takes both of these signals as the input clock and frame-synchronization, hereafter designated as the slave receiver.

Both codecs take the CLKX and FSX signals as inputs for bit-clock and frame-synchronization BCLK and WCLK, respectively. The digital data output pins DOUT are connected together, as the codecs are programmed to work in TDM mode. The same configuration is used for the digital data input pins DIN. Figure 93 shows the data transmission format with respect to the frame-synchronization and clock signals.

Figure 93. Two AIC33s



The codecs are programmed to operate in DSP mode, which uses a frame-synchronization width of 1 bit clock. Figure 93 shows the data using a McBSP2 2-bit delay. The data delay of the codecs is programmable to any number from 0 to 255.

All timing parameters need to be met for the device data sheets.

6.3 Software Configuration

In order to program McBSP2 as a master transmitter (slave receiver given the external connection) in multichannel mode, configure the following registers with the DSP:

- Disable digital loopback mode in SPCR1
- Disable the clock stop mode in SPCR1
- Choose one phase for the receiver and transmitter in RCR2 and XCR2, respectively
- Set the receive and transmit word length to 16 bits in the RCR1 and XCR1, respectively
- Set the receive and transmit frame length to 4 words in the RCR1 and XCR1, respectively, to accommodate up to the highest numbered channel
- Set the receive/transmit frame-synchronization ignore function to detect unexpected frame-synchronization pulses in the RCR2 and XCR2 respectively. This will set the RSYNCERR and XSYNCERR flags upon receiving unexpected frame synchronization pulses.

- Set the receive and transmit data delay to 2-bits in RCR2 and XCR2, respectively
- Set the receive and transmit interrupt mode to generate an interrupt when an unexpected frame pulse is detected in SPCR1 and SPCR2, respectively
- Set up frame synchronization mode
  - Set the transmit frame-synchronization mode to be supplied internally by the McBSP (McBSP2 is the master transmitter)
  - Set the receive frame-synchronization mode to be supplied by an external source via the FSX pin (connected externally)
- Set the receive and transmit frame-synchronization and clock polarity to 0 in PCR
  - FSR/FSX is active high
  - Receive data sampled on falling edge of CLKR
  - Transmit data sampled on rising edge of CLKX
- Set the receive clock mode for CLKR to operate as an input in PCR (McBSP1 can only be a slave receiver)
- Set the transmit clock mode for CLKX to operate as an output in PCR (McBSP1 is a master transmitter)
- Configure the sample rate generator
  - Set the frame-synchronization period to 64 bit clocks per frame in order to accommodate 4 channels x 16-bits/channel
  - Set the frame-synchronization pulse width to 1 bit for this example
  - Set the transmit clock mode to drive the CLKX pin by the sample rate generator in the PCR register
  - Set the SRG clock divide-down value to the value desired to accomplish the appropriate sampling rate. For example, if using an internal clock source of 48MHz, a CLKGDV value of 7 (divided by 8) will provide a sampling rate of 93.75KHz (6MHz bit-rate with 4 channels × 16-bits/channel).

    If a clock source of 12.288MHz is supplied through the CLKR pin, a CLKGDV value of 1 (divided by 2) will provide a sampling rate of 96KHz (6.1MHz bit rate with 4 channels × 16-bits/channel).
  - Set the SRG clock mode (chose an input clock to the SRG) by setting SCLKME=0, CLKSM=1 in PCR and SRGR1, respectively, to choose the internal clock source as input source
- Leave all other register fields at their default values

For details on the proper McBSP initialization sequence, see section 2.11.

Program the DSP DMA channels that service McBSP2 as follows:

- ❑ Receive Channel (any free channel)
  - Event synchronized to McBSP2 RX event
  - Source port: TIPB
  - Destination port: DARAM
  - Source addressing mode: constant address
  - Destination addressing mode: post-increment address
  - Source address: McBSP2 DRR1
  - Destination address: array in internal DARAM
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.
- ❑ Transmit Channel (any free channel)
  - Event synchronized to McBSP2 TX event
  - Source port: DARAM
  - Destination port: TIPB
  - Source addressing mode: post-increment address
  - Destination addressing mode: constant address
  - Source address: array in internal DARAM
  - Destination address: McBSP2 DXR1
  - Data size: 16-bit
  - Enable half-frame, end-of-block, and drop-event interrupts
  - Frame length: 1 frame per block
  - Element count: 1024 words per frame (512 words per ping/pong buffer)
  - Autoinitialization: enable autoinitialization, enable repeat. By enabling autoinitialization, the ping and pong buffers must reside adjacent to each other in memory.

The DMA handler needs to be set to map the McBSP2 TX and RX DMA requests to the available events. This example maps the requests to events 18 (TX) and 7 (RX). For more information, see the *OMAP5910/5912 Multimedia Processor DSP Subsystem Reference Guide* (SPRU890).

Furthermore, the static switch needs to be set for the DSP to access McBSP2, which is an MPU public peripheral. For more information, see the *OMAP5912 Multimedia Processor Peripheral Interconnects Reference Guide* (SPRU758).

The McBSP1 interrupts used on this example are the following:

- DMA half-frame interrupt receive. Flags the reception completion of half the frame. This can be the ping buffer in memory.
- DMA end of block interrupt receive. Flags the reception completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- DMA half-frame interrupt transmit. Flags the transmission completion of half the frame. This can be the ping buffer in memory.
- DMA end of block interrupt transmit. Flags the transmission completion of the entire block. In this case, there is one frame in the block, indicating the completion of a pong buffer in memory.
- McBSP TX interrupt. This interrupt is optionally used to detect unexpected frame synchronization pulses.
- McBSP RX interrupt. This interrupt is optionally used to detect unexpected frame synchronization pulses.

6.4 System Traffic Considerations

This supported usage case for McBSP2 runs at 93.75 KHz sampling rate for the transmitter and receiver using the internal clock as a source to the sample rate generator, using 16-bit words for each of the 4 data channel (bit clock rate of 6.1MHz). The codec used only supports up to 96KHz sampling rate as a slave, which can be accomplished by using a 12.288MHz input clock on the CLKR pin. The ARMPER_CK is running at 48MHz and the McBSP2 data movement is the only activity on the bus.

Note that performance is affected by the number of requests to a given DMA port (e.g., TIPB port), traffic on the DSP public peripheral bus, external memory access latency, and conflicting accesses by the DMA and core to the same internal memory blocks.

7 McBSP Registers

The base address for each McBSP register map is as follows:

- ❑ McBSP1:
  - ■ E101:1800 (MPU memory map)
  - ■ x001:1800 (DSP memory map), byte address
- ❑ McBSP2:
  - ■ FFFB:1000 (MPU memory map)
  - ■ x001:1000 (DSP memory map), byte address
- ❑ McBSP3:
  - ■ E101:7000 (MPU memory map)
  - ■ x001:7000 (DSP memory map), byte address

Table 83 shows the accessible registers on each McBSP. Table 84 through Table 103 describe register bits.

Table 83. McBSP Registers

Name	Description	Offset
DRR2(15:0)	Data receive register 2	0x00
DRR1(15:0)	Data receive register 1	0x02
DXR2(15:0)	Data transmit register 2	0x04
DXR1(15:0)	Data transmit register 1	0x06
SPCR2(15:0)	Serial port control register 2	0x08
SPCR1(15:0)	Serial port control register 1	0x0A
RCR2(15:0)	Receive control register 2	0x0C
RCR1(15:0)	Receive control register 1	0x0E
XCR2 (15:0)	Transmit control register 2	0x10
XCR1(15:0)	Transmit control register 1	0x12
SRGR2(15:0)	Sample rate generator register 2	0x14
SRGR1(15:0)	Sample rate generator register 1	0x16
MCR2(15:0)	Multichannel register 2	0x18
MCR1(15:0)	Multichannel register 1	0x1A
RCERA(15:0)	Receive channel enable register partition A	0x1C

Table 83. McBSP Registers (Continued)

Name	Description	Offset
RCERB(15:0)	Receive channel enable register partition B	0x1E
XCERA(15:0)	Transmit channel enable register partition A	0x20
XCERB(15:0)	Transmit channel enable register partition B	0x22
PCR0(15:0)	Pin control register	0x24
RCERC(15:0)	Receive channel enable register partition C	0x26
RCERD(15:0)	Receive channel enable register partition D	0x28
XCERC(15:0)	Transmit channel enable register partition C	0x2A
XCERD(15:0)	Transmit channel enable register partition D	0x2C
RCERE(15:0)	Receive channel enable register partition E	0x2E
RCERF(15:0)	Receive channel enable register partition F	0x30
XCERE(15:0)	Transmit channel enable register partition E	0x32
XCERF(15:0)	Transmit channel enable register partition F	0x34
RCERG(15:0)	Receive channel enable register partition G	0x36
RCERH(15:0)	Receive channel enable register partition H	0x38
XCERG(15:0)	Transmit channel enable register partition G	0x3A
XCERH(15:0)	Transmit channel enable register partition H	0x3C

7.1 Data Receive Registers (DRR2 and DRR1)

The DSP core, MPU core, or DSP/system DMA controller reads received data from one or both of the data receive registers (Figure 94). If the serial word length is 16 bits or smaller, only DRR1 is used. If the serial length is larger than 16 bits, both DRR1 and DRR2 are used and DRR2 holds the most significant bits. Each frame of receive data in the McBSP can have one phase or two phases, each with its own serial word length.

DRR1 and DRR2 are I/O mapped registers; they are accessible at addresses in I/O space.

Figure 94. Data Receive Registers (DRR2 and DRR1)

DRR2

15	0
High part of receive data (for 20-, 24- or 32-bit data)	
R/W−0	

DRR1

15	0
Receive data (for 8-, 12-, or 16-bit data) or low part of receive data (for 20-, 24- or 32-bit data)	
R/W−0	

Legend: R = read; W = write; -*n* = value after reset

7.1.1 Data Travel From Data Receive Pins to the Registers

If the serial word length is 16 bits or smaller, receive data on the DR pin is shifted into receive shift register 1 (RSR1) and then copied into receive buffer register 1 (RBR1). The content of RBR1 is then copied to DRR1, which can be read by the DSP core, MPU core, or DSP/system DMA controller.

If the serial word length is larger than 16 bits, receive data on the DR pin is shifted into both of the receive shift registers (RSR2, RSR1) and then copied into both of the receive buffer registers (RBR2, RBR1). The content of the RBRs is then copied into the DRRs, which can be read by the DSP core, MPU core, or DSP/system DMA controller.

If companding is used during the copy from RBR1 to DRR1 (RCOMPAND = 10b or 11b), the 8-bit compressed data in RBR1 is expanded to a left-justified 16-bit value in DRR1. If companding is disabled, the data copied from RBR[1,2] to DRR[1,2] is justified and bit filled according to the RJUST bits.

The RSRs and RBRs are not accessible. They are not mapped to I/O space like the DRRs.

7.2 Data Transmit Registers (DXR2 and DXR1)

For transmission, the DSP core, MPU core, or DSP/system DMA controller writes data to one or both of the data transmit registers (see Figure 95). If the serial word length is 16 bits or smaller, only DXR1 is used. If the word length is larger than 16 bits, both DXR1 and DXR2 are used and DXR2 holds the most significant bits. Each frame of transmit data in the McBSP can have one phase or two phases, each with its own serial word length.

DXR1 and DXR2 are I/O mapped registers; they are accessible at addresses in I/O space.

Figure 95. Data Transmit Registers (DXR2 and DXR1)

DXR2

15 … 0
High part of transmit data (for 20-, 24- or 32-bit data)
R/W−0

DXR1

15 … 0
Transmit data (for 8-, 12-, or 16-bit data) or low part of receive data (for 20-, 24- or 32-bit data)
R/W−0

Legend: R = read; W = write; -*n* = value after reset

7.2.1 Data Travel From Registers to Data Transmit (DX) Pins

If the serial word length is 16 bits or fewer, data written to DXR1 is copied to transmit shift register 1 (XSR1). From XSR1, the data is shifted onto the DX pin one bit at a time.

If the serial word length is more than 16 bits, data written to DXR1 and DXR2 is copied into both of the transmit shift registers (XSR2, XSR1). From the XSRs, the data is shifted onto the DX pin one bit at a time.

If companding is used during the transfer from DXR1 to XSR1 (XCOMPAND = 10b or 11b), the McBSP compresses the 16-bit data in DXR1 to 8-bit data in the µ-law or A-law format in XSR1. If companding is disabled, the McBSP passes data from the DXR(s) to the XSR(s) without modification.

The XSRs are not accessible. They are not mapped to I/O space like the DXRs.

7.3 Serial Port Control Registers (SPCR1 and SPCR2)

Each McBSP has two serial port control registers. Table 84 and Table 86 describe the bits in SPCR1 and SPCR2, respectively. These I/O-mapped registers enable you to:

❑ Control various McBSP modes: digital loopback mode (DLB), sign-extension and justification mode for reception (RJUST), clock stop mode (CLKSTP), interrupt modes (RINTM and XINTM), emulation mode (FREE and SOFT)

❑ Turn on and off the DX-pin delay enabler (DXENA)

❑ Check the status of receive and transmit operations (RSYNCERR, XSYNCERR, RFULL, XEMPTY, RRDY, XRDY)

❑ Reset portions of the McBSP (RRST, XRST, FRST, GRST)

Table 84. Serial Port Control 1 Register (SPCR1)

Bit	Name	Value	Description	Type	Reset
15	DLB		Digital loopback mode bit. DLB disables or enables the digital loopback mode of the McBSP:	R/W	0
		0	Disabled Internal DR is supplied by the DR pin. Internal FSR and internal CLKR can be supplied by their respective pins or by the sample rate generator, depending on the mode bits FSRM and CLKRM.		
		1	Enabled Internal receive signals are supplied by internal transmit signals: DR connected to DX FSR connected to FSX CLKR connected to CLKX Internal DX is supplied by the DX pin. Internal FSX and internal CLKX are supplied by their respective pins or are generated internally, depending on the mode bits FSXM and CLKXM. This mode allows you to test serial port code with a single DSP. The McBSP transmitter directly supplies data, frame synchronization, and clocking to the McBSP receiver.		
14:13	RJUST		Receive sign-extension and justification mode bits. During reception, RJUST determines how data is justified and bit filled before being passed to the data receive registers (DRR1, DRR2). RJUST is ignored if you enable a companding mode with the RCOMPAND bits. In a companding mode, the 8-bit compressed data in RBR1 is expanded to left-justified 16-bit data in DRR1. For more details about the effects of RJUST, see section 2.5.13.	R/W	00
		00	Right justify the data and zero fill the MSBs.		
		01	Right justify the data and sign-extend the data into the MSBs.		
		10	Left justify the data and zero fill the LSBs.		
		11	Reserved (do not use)		

Table 84. Serial Port Control 1 Register (SPCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
12:11	CLKSTP		Clock stop mode bits. CLKSTP allows you to use the clock stop mode to support the SPI master-slave protocol. If you will not be using the SPI protocol, you can clear CLKSTP to disable the clock stop mode. In the clock stop mode, the clock stops at the end of each data transfer. At the beginning of each data transfer, the clock starts immediately (CLKSTP = 10b) or after a half-cycle delay (CLKSTP = 11b). For more details, see section 2.5.5.	R/W	00
		00/01	Clock stop mode is disabled.		
		10	Clock stop mode, without clock delay.		
		11	Clock stop mode, with half-cycle clock delay.		
10:8	Reserved		Reserved bits (not available for your use). They are read-only bits and return 0s when read.		
7	DXENA		DX delay enabler mode bit.	R/W	0
		0	DX delay enabler off supported only.		
6	Reserved		Always write 0 to this reserved bit for proper receive operation.	R/W	0

Table 84. Serial Port Control 1 Register (SPCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
5:4	RINTM		Receive interrupt mode bits. RINTM determines which event in the McBSP receiver generates a receive interrupt (RINT) request. If RINT is properly enabled inside the DSP core or MPU core, the DSP core or MPU core services the interrupt request; otherwise, the DSP core or MPU core ignores the request.	R/W	00
		00	The McBSP sends a receive interrupt (RINT) request to the DSP core or MPU core when the RRDY bit changes from 0 to 1, indicating that receive data is ready to be read (the content of RBR[1,2] has been copied to DRR[1,2]). Regardless of the value of RINTM, you can check RRDY to determine whether a word transfer is complete. The McBSP sends a RINT request to the DSP core or MPU core when 16 enabled bits have been received on the DR pin.		
		01	In the multichannel selection mode, the McBSP sends a RINT request to the DSP core or MPU core after every 16-channel block is received in a frame. Outside of the multichannel selection mode, no interrupt request is sent.		
		10	The McBSP sends a RINT request to the DSP core or MPU core when each receive frame-synchronization pulse is detected. The interrupt request is sent even if the receiver is in its reset state.		
		11	The McBSP sends a RINT request to the DSP core or MPU core when the RSYNCERR bit is set, indicating a receive frame-synchronization error. Regardless of the value of RINTM, you can check RSYNCERR to determine whether a receive frame-synchronization error occurred.		

Table 84. Serial Port Control 1 Register (SPCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
3	RSYNCERR		Receive frame-sync error bit. RSYNCERR is set when a receive frame-sync error is detected by the McBSP. If RINTM = 11b, the McBSP sends a receive interrupt (RINT) request to the DSP core or MPU core when RSYNCERR is set. The flag remains set until you write a 0 to it or reset the receiver. Caution**:** If RINTM = 11b, writing a 1 to RSYNCERR triggers a receive interrupt just as if a receive frame-synchronization error occurred.	R/W	0
		0	No error.		
		1	Receive frame-synchronization error. For more details about this error, see section 2.7.2.		
2	RFULL		Receiver full bit. RFULL is set when the receiver is full with new data and the previously received data has not been read (receiver-full condition). For more details about this condition, see section 2.7.1.	R	0
		0	No receiver-full condition.		
		1	Receiver-full condition: RSR[1,2] and RBR[1,2] are full with new data, but the previous data in DRR[1,2] has not been read.		

Table 84. Serial Port Control 1 Register (SPCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
1	RRDY		Receiver ready bit. RRDY is set when data is ready to be read from DRR[1,2]. Specifically, RRDY is set in response to a copy from RBR1 to DRR1. If the receive interrupt mode is RINTM = 00b, the McBSP sends a receive interrupt request to the DSP core or MPU core when RRDY changes from 0 to 1. Also, when RRDY changes from 0 to 1, the McBSP sends a receive synchronization event (REVT) signal to the DSP/system DMA controller.	R	0
		0	Receiver not ready. When the content of DRR1 is read, RRDY is automatically cleared.		
		1	Receiver ready: New data can be read from DRR[1,2]. Important: If both DRRs are required (word length larger than 16 bits), the DSP core, MPU core, or DSP/system DMA controller must read from DRR2 first and then from DRR1. As soon as DRR1 is read, the next RBR-to-DRR copy is initiated. If DRR2 is not read first, the data in DRR2 is lost.		
0	RRST		Receiver reset bit. You can use RRST to take the McBSP receiver into and out of its reset state. This bit has a negative polarity; RRST = 0 indicates the reset state. To read about the effects of a receiver reset, see section 2.11.	R/W	0
		0	If you read a 0, the receiver is in its reset state. If you write a 0, you reset the receiver.		
		1	If you read a 1, the receiver is enabled. If you write a 1, you enable the receiver by taking it out of its reset state.		

Table 85. Serial Port Control 2 Register (SPCR2)

Bit	Name	Value	Description	Type	Reset
15:10	Reserved		Reserved bits (not available for your use). They are read-only bits and return 0s when read.		
9	FREE		Free run bit. When a breakpoint is encountered in the high-level language debugger, FREE determines whether the McBSP transmit and receive clocks continue to run or whether they are affected as determined by the SOFT bit. When one of the clocks stops, the corresponding data transfer (transmission or reception) stops.	R/W	0
		0	The McBSP transmit and receive clocks are affected as determined by the SOFT bit.		
		1	Free run. The McBSP transmit and receive clocks continue to run.		
8	SOFT		Soft stop bit. When FREE = 0, SOFT determines the response of the McBSP transmit and receive clocks when a breakpoint is encountered in the high-level language debugger. When one of the clocks stops, the corresponding data transfer (transmission or reception) stops.	R/W	0
		0	Hard stop. The McBSP transmit and receive clocks are stopped immediately.		
		1	Soft stop. The McBSP transmit clock stops after completion of the current serial word transfer. The McBSP receive clock is not affected.		

Table 85. Serial Port Control 2 Register (SPCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
7	FRST		Frame-synchronization logic reset bit. The sample rate generator of the McBSP includes frame-synchronization logic to generate an internal frame-synchronization signal. You can use FRST to take the frame-synchronization logic into and out of its reset state. This bit has a negative polarity; FRST = 0 indicates the reset state.	R/W	0
		0	If you read a 0, the frame-synchronization logic is in its reset state. If you write a 0, you reset the frame-synchronization logic. In the reset state, the frame-synchronization logic does not generate a frame-synchronization signal (FSG).		
		1	If you read a 1, the frame-synchronization logic is enabled. If you write a 1, you enable the frame-synchronization logic by taking it out of its reset state. When the frame-synchronization logic is enabled (FRST = 1) and the sample rate generator as a whole is enabled (GRST = 1), the frame-synchronization logic generates the frame-synchronization signal FSG as programmed.		

Table 85. Serial Port Control 2 Register (SPCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
6	GRST		Sample rate generator reset bit. You can use GRST to take the McBSP sample rate generator into and out of its reset state. This bit has a negative polarity; GRST = 0 indicates the reset state. To read about the effects of a sample rate generator reset, see section 2.11.	R/W	0
		0	If you read a 0, the sample rate generator is in its reset state. If you write a 0, you reset the sample rate generator. If GRST = 0 due to a reset, CLKG is driven by the internal clock source divided by 2, and FSG is driven low (inactive). If GRST = 0 due to program code, CLKG and FSG are both driven low (inactive).		
		1	If you read a 1, the sample rate generator is enabled. If you write a 1, you enable the sample rate generator by taking it out of its reset state. When enabled, the sample rate generator generates the clock signal CLKG as programmed in the sample rate generator registers. If FRST = 1, the generator also generates the frame-synchronization signal FSG as programmed in the sample rate generator registers.		

Table 85. Serial Port Control 2 Register (SPCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
5:4	XINTM		Transmit interrupt mode bits. XINTM determines which event in the McBSP transmitter generates a transmit interrupt (XINT) request. If XINT is properly enabled, the DSP core or MPU core services the interrupt request; otherwise, the DSP core or MPU core ignores the request.	R/W	00
		00b	The McBSP sends a transmit interrupt (XINT) request to the DSP core or MPU core when the XRDY bit changes from 0 to 1, indicating that transmitter is ready to accept new data (the content of DXR[1,2] has been copied to XSR[1,2]). Regardless of the value of XINTM, you can check XRDY to determine whether a word transfer is complete. The McBSP sends an XINT request to the DSP core or MPU core when 16 enabled bits have been transmitted on the DX pin.		
		01b	In the multichannel selection mode, the McBSP sends an XINT request to the DSP core or MPU core after every 16-channel block is transmitted in a frame. Outside of the multichannel selection mode, no interrupt request is sent.		
		10b	The McBSP sends an XINT request to the DSP core or MPU core when each transmit frame-synchronization pulse is detected. The interrupt request is sent even if the transmitter is in its reset state. (On the rising edge of the frame-synchronization only.)		
		11b	The McBSP sends an XINT request to the DSP core or MPU core when the XSYNCERR bit is set, indicating a transmit frame-synchronization error. Regardless of the value of XINTM, you can check XSYNCERR to determine whether a transmit frame-synchronization error occurred.		

Table 85. Serial Port Control 2 Register (SPCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
3	XSYNCERR		Transmit frame-synchronization error bit. XSYNCERR is set when a transmit frame-synchronization error is detected by the McBSP. If XINTM = 11b, the McBSP sends a transmit interrupt (XINT) request to the DSP core or MPU core when XSYNCERR is set. The flag remains set until you write a 0 to it or reset the transmitter. If XINTM = 11b, writing a 1 to XSYNCERR triggers a transmit interrupt just as if a transmit frame-synchronization error occurred. For details about this error, see section 2.7.5.	R/W	0
		0	No error.		
		1	Transmit frame-synchronization error.		
2	XEMPTY		Transmitter empty bit. XEMPTY is cleared when the transmitter is ready to send new data but no new data is available (transmitter-empty condition). This bit has a negative polarity; a transmitter-empty condition is indicated by XEMPTY = 0.	R	0
		0	Transmitter-empty condition. Typically this indicates that all the bits of the current word have been transmitted but there is no new data in DXR1. XEMPTY is also cleared if the transmitter is reset and then restarted. For more details about this error condition, see section 2.7.4.		
		1	No transmitter-empty condition.		

Table 85. Serial Port Control 2 Register (SPCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
1	XRDY		Transmitter ready bit. XRDY is set when the transmitter is ready to accept new data in DXR[1,2]. Specifically, XRDY is set in response to a copy from DXR1 to XSR1. If the transmit interrupt mode is XINTM = 00b, the McBSP sends a transmit interrupt (XINT) request to the DSP/ARM when XRDY changes from 0 to 1. Also, when XRDY changes from 0 to 1, the McBSP sends a transmit synchronization event (XEVT) signal to the DSP/system DMA controller.	R	0
		0	Transmitter not ready. When DXR1 is loaded, XRDY is automatically cleared.		
		1	Transmitter ready: DXR[1,2] is ready to accept new data. If both DXRs are needed (word length larger than 16 bits), the DSP/ARM or the DSP/system DMA controller must load DXR2 first and then load DXR1. As soon as DXR1 is loaded, the contents of both DXRs are copied to the transmit shift registers (XSRs), as described in the next step. If DXR2 is not loaded first, the previous content of DXR2 is passed to the XSR2.		
0	XRST		Transmitter reset bit. You can use XRST to take the McBSP transmitter into and out of its reset state. This bit has a negative polarity; XRST = 0 indicates the reset state. To read about the effects of a transmitter reset, see section 2.11.	R/W	0
		0	If you read a 0, the transmitter is in its reset state. If you write a 0, you reset the transmitter.		
		1	If you read a 1, the transmitter is enabled. If you write a 1, you enable the transmitter by taking it out of its reset state.		

7.4 Receive Control Registers (RCR1 and RCR2)

Each McBSP has two receive control registers. Table 86 and Table 88 describe the bits in RCR1 and RCR2, respectively. These I/O-mapped registers enable you to:

- ❑ Specify one or two phases for each frame of receive data (RPHASE)
- ❑ Define two parameters for phase 1 and, if necessary, phase 2: the serial word length (RWDLEN1, RWDLEN2) and the number of words (RFRLEN1, RFRLEN2)
- ❑ Choose a receive companding mode, if any (RCOMPAND)
- ❑ Enable or disable the receive frame-synchronization ignore function (RFIG)
- ❑ Choose a receive data delay (RDATDLY)

Table 86. Receive Control 1 Register (RCR1)

Bit	Name	Value	Description	Type	Reset
15	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.	R	0
14:8	RFRLEN1	0−127	Receive frame length 1 (1 to 128 words). Each frame of receive data can have one or two phases, depending on the value loaded into the RPHASE bit. If a single-phase frame is selected, RFRLEN1 in RCR1 selects the number of serial words (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, RFRLEN1 determines the number of serial words in phase 1 of the frame, and RFRLEN2 in RCR2 determines the number of words in phase 2 of the frame. The 7-bit RFRLEN fields allow up to 128 words per phase. See Table 87 for a summary of how to determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period. Program the RFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For example, if you want a phase length of 128 words in phase 1, load 127 into RFRLEN1.	R/W	0
7:5	RWDLEN1		Receive word length 1. Each frame of receive data can have one or two phases, depending on the value loaded into the RPHASE bit. If a single-phase frame is selected, RWDLEN1 in RCR1 selects the length for every serial word received in the frame. If a dual-phase frame is selected, RWDLEN1 determines the length of the serial words in phase 1 of the frame, and RWDLEN2 in RCR2 determines the word length in phase 2 of the frame.	R/W	000
		000b	8 bits		
		001b	12 bits		
		010b	16 bits		
		011b	20 bits		
		100b	24 bits		
		101b	32 bits		
		other	Reserved (do not use)		
4:0	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.	R	0

Table 87. Frame Length Formula for Receive Control 1 Register (RCR1)

RPHASE	RFRLEN1	RFRLEN2	Frame Length
0	0 ≤ RFRLEN1 ≤ 127	Not used	(RFRLEN1 + 1) words
1	0 ≤ RFRLEN1 ≤ 127	0 ≤ RFRLEN2 ≤ 127	(RFRLEN1 + 1) + (RFRLEN2 + 1) words

Table 88. Receive Control 2 Register (RCR2)

Bit	Name	Value	Description	Type	Reset
15	RPHASE		Receive phase number bit. RPHASE determines whether the receive frame has one phase or two phases. For each phase you can define the serial word length and the number of serial words in the phase. To set up phase 1, program RWDLEN1 (word length) and RFRLEN1 (number of words). To set up phase 2 (if there are two phases), program RWDLEN2 and RFRLEN2.	R/W	0
		0	Single-phase frame. The receive frame has only one phase, phase 1.		
		1	Dual-phase frame. The receive frame has two phases, phase 1 and phase 2.		

Table 88. Receive Control 2 Register (RCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
14:8	RFRLEN2	0−127	Receive frame length 2 (1 to 128 words). Each frame of receive data can have one or two phases, depending on the value loaded into the RPHASE bit. If a single-phase frame is selected, RFRLEN1 in RCR1 selects the number of serial words (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, RFRLEN1 determines the number of serial words in phase 1 of the frame, and RFRLEN2 in RCR2 determines the number of words in phase 2 of the frame. The 7-bit RFRLEN fields allow up to 128 words per phase. See Table 89 for a summary of how to determine the frame length. This length corresponds to the number of words, or logical time slots, or channels per frame-synchronization period. Program the RFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For example, for a phase length of 128 words in phase 2, load 127 into RFRLEN2.	R/W	0
7:5	RWDLEN2		Receive word length 2. Each frame of receive data can have one or two phases, depending on the value loaded into the RPHASE bit. If a single-phase frame is selected, RWDLEN1 in RCR1 selects the length for every serial word received in the frame. If a dual-phase frame is selected, RWDLEN1 determines the length of the serial words in phase 1 of the frame, and RWDLEN2 in RCR2 determines the word length in phase 2 of the frame.	R/W	000
		000b	8 bits		
		001b	12 bits		
		010b	16 bits		
		011b	20 bits		
		100b	24 bits		
		101b	32 bits		
		other	Reserved (do not use)		

Table 88. Receive Control 2 Register (RCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
4:3	RCOMPAND		Receive companding mode bits. Companding (COMpress and exPAND) hardware allows compression and expansion of data in either µ-law or A-law format. For more details about these companding modes, see section 2.2.2. RCOMPAND allows you to choose one of the following companding modes for the McBSP receiver:	R/W	00
		00b	No companding, any size data, MSB received first		
		01b	No companding, 8-bit data, LSB received first		
		10b	µ-law companding, 8-bit data, MSB received first		
		11b	A-law companding, 8-bit data, MSB received first		
2	RFIG		Receive frame-synchronization ignore bit. If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully received, this pulse is treated as an unexpected frame-synchronization pulse. For more details about the frame-synchronization error condition, see section 2.7.2. Setting RFIG causes the serial port to ignore unexpected frame-synchronization signals during reception. For more details on the effects of RFIG, see section 2.5.10.1.	R/W	0
		0	Frame-synchronization detect. An unexpected FSR pulse causes the receiver to discard the contents of RSR[1,2] in favor of the new incoming data. The receiver: 1) Aborts the current data transfer 2) Sets RSYNCERR in SPCR1 3) Begins the transfer of a new data word		
		1	Frame-synchronization ignore. An unexpected FSR pulse is ignored. Reception continues uninterrupted.		

Table 88. Receive Control 2 Register (RCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
1:0	RDATDLY		Receive data delay bits. RDATDLY specifies a data delay of 0, 1, or 2 receive clock cycles after frame-synchronization and before the reception of the first bit of the frame. For more details, see section 2.5.12.	R/W	00
		00b	0-bit data delay		
		01b	1-bit data delay		
		10b	2-bit data delay		
		11b	Reserved (do not use)		

Table 89. Frame Length Formula for RCR2

RPHASE	RFRLEN1	RFRLEN2	Frame Length
0	$0 \leq \text{RFRLEN1} \leq 127$	Not used	(RFRLEN1 + 1) words
1	$0 \leq \text{RFRLEN1} \leq 127$	$0 \leq \text{RFRLEN2} \leq 127$	(RFRLEN1 + 1) + (RFRLEN2 + 1) words

7.5 Transmit Control Registers (XCR1 and XCR2)

Each McBSP has two transmit control registers. Table 90 and Table 92 describe the bits in XCR1 and XCR2, respectively. These I/O-mapped registers enable you to:

❑ Specify one or two phases for each frame of transmit data (XPHASE)

❑ Define two parameters for phase 1 and, if necessary, phase 2: the serial word length (XWDLEN1, XWDLEN2) and the number of words (XFRLEN1, XFRLEN2)

❑ Choose a transmit companding mode, if any (XCOMPAND)

❑ Enable or disable the transmit frame-sync ignore function (XFIG)

❑ Choose a transmit data delay (XDATDLY)

Table 90. Transmit Control 1 Register (XCR1)

Bit	Name	Value	Description	Type	Reset
15	Reserved	0	Reserved bit. Read-only; returns 0 when read.	R	0
14:8	XFRLEN1	0−127	Transmit frame length 1 (1 to 128 words). Each frame of transmit data can have one or two phases, depending on the value loaded into the XPHASE bit. If a single-phase frame is selected, XFRLEN1 in XCR1 selects the number of serial words (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, XFRLEN1 determines the number of serial words in phase 1 of the frame, and XFRLEN2 in XCR2 determines the number of words in phase 2 of the frame. The 7-bit XFRLEN fields allow up to 128 words per phase. See Table 91 for a summary of how to determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period. Program the XFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For example, for a phase length of 128 words in phase 1, load 127 into XFRLEN1.	R/W	0
7:5	XWDLEN1		Transmit word length 1. Each frame of transmit data can have one or two phases, depending on the value loaded into the XPHASE bit. If a single-phase frame is selected, XWDLEN1 in XCR1 selects the length for every serial word transmitted in the frame. If a dual-phase frame is selected, XWDLEN1 determines the length of the serial words in phase 1 of the frame, and XWDLEN2 in XCR2 determines the word length in phase 2 of the frame.	R/W	000
		000b	8 bits		
		001b	12 bits		
		010b	16 bits		
		011b	20 bits		
		100b	24 bits		
		101b	32 bits		
		other	Reserved (do not use)		
4:0	Reserved	0	Reserved bits. They are read-only bits and return 0s when read.	R	0

Table 91. Frame Length Formula for Transmit Control 1 Register (XCR1)

XPHASE	XFRLEN1	XFRLEN2	Frame Length
0	0 ≤ XFRLEN1 ≤ 127	Not used	(XFRLEN1 + 1) words
1	0 ≤ XFRLEN1 ≤ 127	0 ≤ XFRLEN2 ≤ 127	(XFRLEN1 + 1) + (XFRLEN2 + 1) words

Table 92. Transmit Control 2 Register (XCR2)

Bit	Name	Value	Description	Type	Reset
15	XPHASE		Transmit phase number bit. XPHASE determines whether the transmit frame has one phase or two phases. For each phase you can define the serial word length and the number of serial words in the phase. To set up phase 1, program XWDLEN1 (word length) and XFRLEN1 (number of words). To set up phase 2 (if there are two phases), program XWDLEN2 and XFRLEN2.	R/W	0
		0	Single-phase frame The transmit frame has only one phase, phase 1.		
		1	Dual-phase frame The transmit frame has two phases, phase 1 and phase 2.		
14:8	XFRLEN2	0–127	Transmit frame length 2 (1 to 128 words). Each frame of transmit data can have one or two phases, depending on the value loaded into the XPHASE bit. If a single-phase frame is selected, XFRLEN1 in XCR1 selects the number of serial words (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, XFRLEN1 determines the number of serial words in phase 1 of the frame, and XFRLEN2 in XCR2 determines the number of words in phase 2 of the frame. The 7-bit XFRLEN fields allow up to 128 words per phase. See Table 93 for a summary of how to determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period. Program the XFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For example, for a phase length of 128 words in phase 1, load 127 into XFRLEN1.	R/W	0

Table 92. Transmit Control 2 Register (XCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
7:5	XWDLEN2		Transmit word length 2. Each frame of transmit data can have one or two phases, depending on the value loaded into the XPHASE bit. If a single-phase frame is selected, XWDLEN1 in XCR1 selects the length for every serial word transmitted in the frame. If a dual-phase frame is selected, XWDLEN1 determines the length of the serial words in phase 1 of the frame and XWDLEN2 in XCR2 determines the word length in phase 2 of the frame.	R/W	000
		000b	8 bits		
		001b	12 bits		
		010b	16 bits		
		011b	20 bits		
		100b	24 bits		
		101b	32 bits		
		other	Reserved (do not use)		
4:3	XCOMPAND		Transmit companding mode bits. Companding (COMpress and exPAND) hardware allows compression and expansion of data in either µ-law or A-law format. For more details, see section 2.2.2. XCOMPAND allows you to choose one of the following companding modes for the McBSP transmitter. For more details about these companding modes, see section 2.2.2.	R/W	00
		00b	No companding, any size data, MSB transmitted first		
		01b	No companding, 8-bit data, LSB transmitted first		
		10b	µ-law companding, 8-bit data, MSB transmitted first		
		11b	A-law companding, 8-bit data, MSB transmitted first		

Table 92. Transmit Control 2 Register (XCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
2	XFIG		Transmit frame-synchronization ignore bit. If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame-synchronization pulse. For more details about the frame-synchronization error condition, see section 2.7.5. Setting XFIG causes the serial port to ignore unexpected frame-synchronization pulses during transmission. For more details on the effects of XFIG, see section 2.6.10.	R/W	0
		0	Frame-synchronization detect. An unexpected FSX pulse causes the transmitter to discard the content of XSR[1,2]. The transmitter: 1) Aborts the present transmission. 2) Sets XSYNCERR in SPCR2. 3) Begins a new transmission from DXR[1,2]. If new data was written to DXR[1,2] since the last DXR[1,2]-to-XSR[1,2] copy, the current value in XSR[1,2] is lost. Otherwise, the same data is transmitted.		
		1	Frame-synchronization ignore. An unexpected FSX pulse is ignored. Transmission continues uninterrupted.		

Table 92. Transmit Control 2 Register (XCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
1:0	XDATDLY		Transmit data delay bits. XDATDLY specifies a data delay of 0, 1, or 2 transmit clock cycles after frame synchronization and before the transmission of the first bit of the frame. For more details, see section 2.6.12.	R/W	00
		00b	0-bit data delay		
		01b	1-bit data delay		
		10b	2-bit data delay		
		11b	Reserved (do not use)		

Table 93. Frame Length Formula for Transmit Control 2 Register (XCR2)

XPHASE	XFRLEN1	XFRLEN2	Frame Length
0	0 ≤ XFRLEN1 ≤ 127	Not used	(XFRLEN1 + 1) words
1	0 ≤ XFRLEN1 ≤ 127	0 ≤ XFRLEN2 ≤ 127	(XFRLEN1 + 1) + (XFRLEN2 + 1) words

7.6 Sample Rate Generator Registers (SRGR1 and SRGR2)

Each McBSP has two sample rate generator registers. Table 94 and Table 95 describe the bits in SRGR1 and SRGR2, respectively. The sample rate generator can generate a clock signal (CLKG) and a frame-synchronization signal (FSG). The I/O-mapped registers SRGR1 and SRGR2 enable you to:

- ❑ Select the input clock source for the sample rate generator (CLKSM, in conjunction with the SCLKME bit in PCR).
- ❑ Divide down the frequency of CLKG (CLKGDV).
- ❑ Select whether internally-generated transmit frame-synchronization pulses are driven by FSG or by activity in the transmitter (FSGM).
- ❑ Specify the width of frame-synchronization pulses on FSG (FWID) and specify the period between those pulses (FPER).

When an external source (via the CLKS, CLKR, or CLKX pin) provides the input clock source for the sample rate generator:

- If the CLKS pin provides the input clock, the CLKSP bit in SRGR2 allows you to select whether the rising edge or the falling edge of CLKS triggers CLKG and FSG. If the CLKX/CLKR pin is used instead of the CLKS pin, the polarity of the input clock is selected with CLKXP/CLKRP of PCR.
- The GSYNC bit in SRGR2 allows you to make CLKG synchronized to an external frame-synchronization signal on the FSR pin, so that CLKG is kept in phase with the input clock.

Table 94. Sample Rate Generator 1 Register (SRGR1)

<table>
<tr><th>Bit</th><th>Name</th><th>Value</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td>15:8</td><td>FWID</td><td>0−255</td><td>Frame-synchronization pulse width bits for FSG
The sample rate generator can produce a clock signal, CLKG, and a frame-synchronization signal, FSG. For frame-synchronization pulses on FSG, (FWID + 1) is the pulse width in CLKG cycles. The eight bits in FWID allow a pulse width of 1 to 256 CLKG cycles:
0 ≤ FWID ≤ 255
1 ≤ (FWID + 1) ≤ 256 CLKG cycles
The period between the frame-synchronization pulses on FSG is defined by the FPER bits.</td><td>R/W</td><td>0</td></tr>
<tr><td>7:0</td><td>CLKGDV</td><td>0−255</td><td>Divide-down value for CLKG. The sample rate generator can accept an input clock signal and divide it down according to CLKGDV to produce an output clock signal, CLKG. The frequency of CLKG is:
CLKG frequency = (Input clock frequency)/ (CLKGDV + 1)
The input clock is selected by the SCLKME and CLKSM bits:<table>
<tr><th>SCLKME</th><th>CLKSM</th><th>Input Clock For Sample Rate Generator</th></tr>
<tr><td>0</td><td>0</td><td>Signal on CLKS pin</td></tr>
<tr><td>0</td><td>1</td><td>Internal clock source</td></tr>
<tr><td>1</td><td>0</td><td>Signal on CLKR pin</td></tr>
<tr><td>1</td><td>1</td><td>Signal on CLKX pin</td></tr>
</table>A reset forces the CLKG frequency to 1/2 the input clock frequency (CLKGDV = 1), and selects the internal clock source as the input clock.</td><td>R/W</td><td>1</td></tr>
</table>

Table 95. Sample Rate Generator 2 Register (SRGR2)

Bit	Name	Value	Description	Type	Reset
15	GSYNC		Clock synchronization mode bit for CLKG. GSYNC is used only when the input clock source for the sample rate generator is external− on the CLKS, CLKR, or CLKX pin. When GSYNC = 1, the clock signal (CLKG) and the frame-synchronization signal (FSG) generated by the sample rate generator are made dependent on pulses on the FSR pin.	R/W	0
		0	No clock synchronization. CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.		
		1	Clock synchronization. CLKG is adjusted as necessary so that it is synchronized with the input clock on the CLKS, CLKR, or CLKX pin, and FSG pulses. FSG only pulses in response to a pulse on the FSR pin. The frame-synchronization period defined in FPER is ignored. For more details, see section 2.1.3.		
14	CLKSP		CLKS pin polarity bit. CLKSP is used only when the CLKS pin is the input clock source for the sample rate generator. The bit determines which edge of CLKS drives the clock signal (CLKG) and the frame-synchronization signal (FSG) that are generated by the sample rate generator:	R/W	0
		0	A rising edge on the CLKS pin.		
		1	A falling edge on the CLKS pin.		

Table 95. Sample Rate Generator 2 Register (SRGR2) (Continued)

<table>
<tr><th>Bit</th><th>Name</th><th>Value</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td>13</td><td>CLKSM</td><td></td><td>Sample rate generator input clock mode bit. The sample rate generator can accept an input clock signal and divide it down according to CLKGDV to produce an output clock signal, CLKG. The frequency of CLKG is:
CLKG frequency = (input clock frequency)/ (CLKGDV + 1)
CLKSM is used in conjunction with the SCLKME bit to determine the source for the input clock.
A reset selects the internal clock source as the input clock and forces the CLKG frequency to 1/2 the internal clock source frequency.</td><td>R/W</td><td>1</td></tr>
<tr><td></td><td></td><td>0</td><td>The input clock for the sample rate generator is either taken from the CLKS pin or the CLKR pin, depending on the value of the SCLKME bit in PCR:
<table>
<tr><th>SCLKME</th><th>CLKSM</th><th>Input Clock For Sample Rate Generator</th></tr>
<tr><td>0</td><td>0</td><td>Signal on CLKS pin</td></tr>
<tr><td>1</td><td>0</td><td>Signal on CLKR pin</td></tr>
</table></td><td></td><td></td></tr>
<tr><td></td><td></td><td>1</td><td>The input clock for the sample rate generator is either taken from the Internal clock or the CLKX pin, depending on the value of the SCLKME bit in PCR:
<table>
<tr><th>SCLKME</th><th>CLKSM</th><th>Input Clock For Sample Rate Generator</th></tr>
<tr><td>0</td><td>1</td><td>Internal clock</td></tr>
<tr><td>1</td><td>1</td><td>Signal on CLKX pin</td></tr>
</table></td><td></td><td></td></tr>
</table>

Table 95. Sample Rate Generator 2 Register (SRGR2) (Continued)

Bit	Name	Value	Description	Type	Reset
12	FSGM		Sample rate generator transmit frame-synchronization mode bit. The transmitter can get frame synchronization from the FSX pin (FSXM = 0) or from inside the McBSP (FSXM = 1). When FSXM = 1, the FSGM bit determines how the McBSP supplies frame-synchronization pulses.	R/W	0
		0	If FSXM = 1, the McBSP generates a transmit frame-synchronization pulse when the content of DXR[1,2] is copied to XSR[1,2].		
		1	If FSXM = 1, the transmitter uses frame-synchronization pulses generated by the sample rate generator, which programs the FWID bits to set the width of each pulse, and programs the FPER bits to set the period between pulses.		
11:0	FPER	0–4095	Frame-synchronization period bits for FSG. The sample rate generator can produce a clock signal, CLKG, and a frame-synchronization signal, FSG. The period between frame-synchronization pulses on FSG is (FPER + 1) CLKG cycles. The 12 bits in FPER allow a frame-synchronization period of 1 to 4096 CLKG cycles: $0 \le \text{FPER} \le 4095$ $1 \le (\text{FPER} + 1) \le 4096$ CLKG cycles The width of each frame-synchronization pulse on FSG is defined by the FWID bits.	R/W	0

7.7 Multichannel Control Registers (MCR1 and MCR2)

Each McBSP has two multichannel control registers. MCR1 has control and status bits (with an R prefix) for multichannel selection operation in the receiver. MCR2 contains the same type of bits (but with an X prefix) for the transmitter. The bits in MCR1 and MCR2 are described in Table 96 and Table 97, respectively. These I/O-mapped registers enable you to:

- Enable all channels or only selected channels for reception (RMCM).
- Choose which channels are enabled/disabled and masked/unmasked for transmission (XMCM).
- Specify whether two partitions (32 channels at a time) or eight partitions (128 channels at a time) can be used (RMCME for reception, XMCME for transmission).

- ❑ Assign blocks of 16 channels to partitions A and B when the 2-partition mode is selected (RPABLK and RPBBLK for reception, XPABLK and XPBBLK for transmission).
- ❑ Determine which block of 16 channels is currently involved in a data transfer (RCBLK for reception, XCBLK for transmission).

Table 96. Multichannel Control 1 Register (MCR1)

Bit	Name	Value	Description	Type	Reset
15:10	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.	R	0
9	RMCME		Receive multichannel partition mode bit. RMCME is only applicable if channels can be individually enabled or disabled for reception (RMCM = 1). RMCME determines whether only 32 channels or all 128 channels are to be individually selectable.	R/W	0
		0	2-partition mode Only partitions A and B are used. You can control up to 32 channels in the receive multichannel selection mode (RMCM = 1). Assign 16 channels to partition A with the RPABLK bits. Assign 16 channels to partition B with the RPBBLK bits. You control the channels with the appropriate receive channel enable registers: RCERA: Channels in partition A RCERB: Channels in partition B		
		1	8-partition mode All partitions (A through H) are used. You can control up to 128 channels in the receive multichannel selection mode. You control the channels with the appropriate receive channel enable registers: RCERA: Channels 0 through 15 RCERB: Channels 16 through 31 RCERC: Channels 32 through 47 RCERD: Channels 48 through 63 RCERE: Channels 64 through 79 RCERF: Channels 80 through 95 RCERG: Channels 96 through 111 RCERH: Channels 112 through 127		

Table 96. Multichannel Control 1 Register (MCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
8:7	RPBBLK		Receive partition B block bits RPBBLK is only applicable if channels can be individually enabled or disabled (RMCM = 1) and the 2-partition mode is selected (RMCME = 0). Under these conditions, the McBSP receiver can accept or ignore data in any of the 32 channels that are assigned to partitions A and B of the receiver. The 128 receive channels of the McBSP are divided equally among 8 blocks (0 through 7). When RPBBLK is applicable, use RPBBLK to assign one of the odd-numbered blocks (1, 3, 5, or 7) to partition B. Use the RPABLK bits to assign one of the even-numbered blocks (0, 2, 4, or 6) to partition A. If you want to use more than 32 channels, you can change block assignments dynamically. You can assign a new block to one partition while the receiver is handling activity in the other partition. For example, while the block in partition A is active, you can change which block is assigned to partition B. The RCBLK bits are regularly updated to indicate which block is active. When XMCM = 11b (for symmetric transmission and reception), the transmitter uses the receive block bits (RPABLK and RPBBLK) rather than the transmit block bits (XPABLK and XPBBLK).	R/W	00
		00b	Block 1: channels 16 through 31		
		01b	Block 3: channels 48 through 63		
		10b	Block 5: channels 80 through 95		
		11b	Block 7: channels 112 through 127		

Table 96. Multichannel Control 1 Register (MCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
6:5	RPABLK		Receive partition A block bits RPABLK is only applicable if channels can be individually enabled or disabled (RMCM = 1) and the 2-partition mode is selected (RMCME = 0). Under these conditions, the McBSP receiver can accept or ignore data in any of the 32 channels that are assigned to partitions A and B of the receiver. See the description for RPBBLK (bits 8-7) for more information about assigning blocks to partitions A and B.	R/W	00
		00b	Block 0: channels 0 through 15		
		01b	Block 2: channels 32 through 47		
		10b	Block 4: channels 64 through 79		
		11b	Block 6: channels 96 through 111		
4:2	RCBLK		Receive current block indicator. RCBLK indicates which block of 16 channels is involved in the current McBSP reception:	R	000
		000b	Block 0: channels 0 through 15		
		001b	Block 1: channels 16 through 31		
		010b	Block 2: channels 32 through 47		
		011b	Block 3: channels 48 through 63		
		100b	Block 4: channels 64 through 79		
		101b	Block 5: channels 80 through 95		
		110b	Block 6: channels 96 through 111		
		111b	Block 7: channels 112 through 127		
1	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.	R	0

Table 96. Multichannel Control 1 Register (MCR1) (Continued)

Bit	Name	Value	Description	Type	Reset
0	RMCM		Receive multichannel selection mode bit. RMCM determines whether all channels or only selected channels are enabled for reception:	R/W	0
		0	All 128 channels are enabled.		
		1	Multichannel selection mode. Channels can be individually enabled or disabled. The only channels enabled are those selected in the appropriate receive channel enable registers (RCERs). The way channels are assigned to the RCERs depends on the number of receive channel partitions (2 or 8), as defined by the RMCME bit.		

Table 97. Multichannel Control 2 Register (MCR2)

Bit	Name	Value	Description	Type	Reset
15:10	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.	R	0
9	XMCME		Transmit multichannel partition mode bit. XMCME determines whether only 32 channels or all 128 channels are to be individually selectable. XMCME is only applicable if channels can be individually disabled/enabled or masked/unmasked for transmission (XMCM is nonzero).	R/W	0
		0	2-partition mode. Only partitions A and B are used. You can control up to 32 channels in the transmit multichannel selection mode selected with the XMCM bits. If XMCM = 01b or 10b, assign 16 channels to partition A with the XPABLK bits. Assign 16 channels to partition B with the XPBBLK bits. If XMCM = 11b (for symmetric transmission and reception), assign 16 channels to receive partition A with the RPABLK bits. Assign 16 channels to receive partition B with the RPBBLK bits. You control the channels with the appropriate transmit channel enable registers: XCERA: Channels in partition A XCERB: Channels in partition B		

Table 97. Multichannel Control 2 Register (MCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
9 (Cont)		1	8-partition mode. All partitions (A through H) are used. You can control up to 128 channels in the transmit multichannel selection mode selected with the XMCM bits. You control the channels with the appropriate transmit channel enable registers: XCERA: Channels 0 through 15 XCERB: Channels 16 through 31 XCERC: Channels 32 through 47 XCERD: Channels 48 through 63 XCERE: Channels 64 through 79 XCERF: Channels 80 through 95 XCERG: Channels 96 through 111 XCERH: Channels 112 through 127		
8:7	XPBBLK		Transmit partition B block bits XPBBLK is only applicable if channels can be individually disabled/enabled and masked/unmasked (XMCM is nonzero) and the 2-partition mode is selected (XMCME = 0). Under these conditions, the McBSP transmitter can transmit or withhold data in any of the 32 channels that are assigned to partitions A and B of the transmitter. The 128 transmit channels of the McBSP are divided equally among 8 blocks (0 through 7). When XPBBLK is applicable, use XPBBLK to assign one of the odd-numbered blocks (1, 3, 5, or 7) to partition B, as shown in the following table. Use the XPABLK bit to assign one of the even-numbered blocks (0, 2, 4, or 6) to partition A. If you want to use more than 32 channels, you can change block assignments dynamically. You can assign a new block to one partition while the transmitter is handling activity in the other partition. For example, while the block in partition A is active, you can change which block is assigned to partition B. The XCBLK bits are regularly updated to indicate which block is active. When XMCM = 11b (for symmetric transmission and reception), the transmitter uses the receive block bits (RPABLK and RPBBLK) rather than the transmit block bits (XPABLK and XPBBLK).	R/W	00
		00b	Block 1: channels 16 through 31		
		01b	Block 3: channels 48 through 63		
		10b	Block 5: channels 80 through 95		
		11b	Block 7: channels 112 through 127		

Table 97. Multichannel Control 2 Register (MCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
6:5	XPABLK		Transmit partition A block bits. XPABLK is only applicable if channels can be individually disabled/enabled and masked/ unmasked (XMCM is nonzero) and the 2-partition mode is selected (XMCME = 0). Under these conditions, the McBSP transmitter can transmit or withhold data in any of the 32 channels that are assigned to partitions A and B of the transmitter. See the description for XPBBLK (bits 8-7) for more information about assigning blocks to partitions A and B.	R/W	00
		00b	Block 0: channels 0 through 15		
		01b	Block 2: channels 32 through 47		
		10b	Block 4: channels 64 through 79		
		11b	Block 6: channels 96 through 111		
4:2	XCBLK		Transmit current block indicator. XCBLK indicates which block of 16 channels is involved in the current McBSP transmission:	R	000
		000b	Block 0: channels 0 through 15		
		001b	Block 1: channels 16 through 31		
		010b	Block 2: channels 32 through 47		
		011b	Block 3: channels 48 through 63		
		100b	Block 4: channels 64 through 79		
		101b	Block 5: channels 80 through 95		
		110b	Block 6: channels 96 through 111		
		111b	Block 7: channels 112 through 127		

Table 97. Multichannel Control 2 Register (MCR2) (Continued)

Bit	Name	Value	Description	Type	Reset
1:0	XMCM		Transmit multichannel selection mode bits. XMCM determines whether all channels or only selected channels are enabled and unmasked for transmission. For more details on how the channels are affected, see section 2.8.7.	R/W	00
		00b	No transmit multichannel selection mode is on. All channels are enabled and unmasked. No channels can be disabled or masked.		
		01b	All channels are disabled unless they are selected in the appropriate transmit channel enable registers (XCERs). If enabled, a channel in this mode is also unmasked. The XMCME bit determines whether 32 channels or 128 channels are selectable in XCERs.		
		10b	All channels are enabled, but they are masked unless they are selected in the appropriate transmit channel enable registers (XCERs). The XMCME bit determines whether 32 channels or 128 channels are selectable in XCERs.		
		11b	This mode is used for symmetric transmission and reception. All channels are disabled for transmission unless they are enabled for reception in the appropriate receive channel enable registers (RCERs). Once enabled, they are masked unless they are also selected in the appropriate transmit channel enable registers (XCERs). The XMCME bit determines whether 32 channels or 128 channels are selectable in RCERs and XCERs.		

7.8 Pin Control Register (PCR)

Each McBSP has one pin control register. Table 98 describes the bits in PCR. This I/O-mapped register enables you to:

❑ Allow the McBSP to enter a low-power mode when the idle instruction is executed (IDLEEN, in conjunction with the PERI bit in ICR).

❑ Specify whether McBSP pins can be used as general-purpose I/O pins when the transmitter and/or receiver is in its reset state (XIOEN and RIOEN).

❑ Choose a frame-synchronization mode for the transmitter (FSXM) and for the receiver (FSRM).

- Choose a clock mode for transmitter (CLKXM) and for the receiver (CLKRM).
- Select the input clock source for the sample rate generator (SCLKME, in conjunction with the CLKSM bit in SRGR2).
- Read or write data when the CLKS, DX, and DR pins are configured as general-purpose I/O pins (CLKSSTAT, DXSTAT, and DXSTAT).
- Choose whether frame-synchronization signals are active low or active high (FSXP for transmission, FSRP for reception).
- Specify whether data is sampled on the falling edge or the rising edge of the clock signals (CLKXP for transmission, CLKRP for reception).

Table 98. Pin Control Register (PCR)

Bit	Name	Value	Description	Type	Reset
15	Reserved	0	Reserved bit (not available for your use). It is a read-only bit and returns a 0 when read.	R	0
14	Reserved		Reserved bit. For proper operation, you must only write a 0 to this bit.	R/W	0
13	XIOEN		Transmit I/O enable bit. When the transmitter is in reset (XRST = 0), XIOEN can configure certain McBSP pins as general-purpose I/O (GPIO) pins. For a summary, see Table 14.	R/W	0
		0	The CLKX, FSX, DX, and CLKS pins are serial port pins.		
		1	If XRST = 0, the CLKX, FSX, and DX pins are GPIO pins. The CLKS is also a GPIO pin if RRST = 0 and RIOEN = 1.		
12	RIOEN		Receive I/O enable bit. When the receiver is in reset (RRST = 0), RIOEN can configure certain McBSP pins as general-purpose I/O (GPIO) pins . For a summary, see Table 14. XRST and RRST are in the serial port control registers, but all other bits mentioned in this table are in the pin control register.	R/W	0
		0	The CLKR, FSR, DR, and CLKS pins are serial port pins.		
		1	If RRST = 0, the CLKR, FSR, and DR pins are GPIO pins. The CLKS is also a GPIO pin if XRST = 0 and XIOEN = 1.		

Table 98. Pin Control Register (PCR) (Continued)

Bit	Name	Value	Description	Type	Reset
11	FSXM		Transmit frame-synchronization mode bit. FSXM determines whether transmit frame-synchronization pulses are supplied externally or internally. The polarity of the signal on the FSX pin is determined by the FSXP bit.	R/W	0
		0	Transmit frame synchronization is supplied by an external source via the FSX pin.		
		1	Transmit frame synchronization is supplied by the McBSP, as determined by the FSGM bit in SRGR2.		
10	FSRM		Receive frame-synchronization mode bit. FSRM determines whether receive frame-synchronization pulses are supplied externally or internally. The polarity of the signal on the FSR pin is determined by the FSRP bit.	R/W	0
		0	Receive frame synchronization is supplied by an external source via the FSR pin.		
		1	Receive frame synchronization is supplied by the sample rate generator. FSR is an output pin reflecting internal FSR, except when GSYNC = 1 in SRGR2.		
9	CLKXM		Transmit clock mode bit. CLKXM determines whether the source for the transmit clock is external or internal, and whether the CLKX pin is an input or an output. The polarity of the signal on the CLKX pin is determined by the CLKXP bit. In the clock stop mode (CLKSTP = 10b or 11b), the McBSP can act as a master or as a slave in the SPI protocol. If the McBSP is a master, make sure that CLKX is an output. If the McBSP is a slave, make sure that CLKX is an input. Not in clock stop mode (CLKSTP = 00b or 01b):	R/W	0
		0	The transmitter gets its clock signal from an external source via the CLKX pin.		
		1	Internal CLKX is driven by the sample rate generator of the McBSP. The CLKX pin is an output pin that reflects internal CLKX.		

Table 98. Pin Control Register (PCR) (Continued)

Bit	Name	Value	Description	Type	Reset
9 (Cont.)			In clock stop mode (CLKSTP = 10b or 11b):		
		0	The McBSP is a slave in the SPI protocol. The internal transmit clock (CLKX) is driven by the SPI master via the CLKX pin. The internal receive clock (CLKR) is driven internally by CLKX, so that both the transmitter and the receiver are controlled by the external master clock.		
		1	The McBSP is a master in the SPI protocol. The sample rate generator drives the internal transmit clock (CLKX). Internal CLKX is reflected on the CLKX pin to drive the shift clock of the SPI-compliant slaves in the system. Internal CLKX also drives the internal receive clock (CLKR), so that both the transmitter and the receiver are controlled by the internal master clock.		
8	CLKRM		Receive clock mode bit. The role of CLKRM and its effect on the CLKR pin depend on whether the McBSP is in the digital loopback mode (DLB = 1). The polarity of the signal on the CLKR pin is determined by the CLKRP bit. Not in digital loopback mode (DLB = 0):	R/W	0
		0	The CLKR pin is an input pin that supplies the internal receive clock (CLKR).		
		1	Internal CLKR is driven by the sample rate generator of the McBSP. The CLKR pin is an output pin that reflects internal CLKR.		
			In digital loopback mode (DLB = 1):		
		0	The CLKR pin is in the high impedance state. The internal receive clock (CLKR) is driven by the internal transmit clock (CLKX). CLKX is derived according to the CLKXM bit.		
		1	Internal CLKR is driven by internal CLKX. The CLKR pin is an output pin that reflects internal CLKR. CLKX is derived according to the CLKXM bit.		

Table 98. Pin Control Register (PCR) (Continued)

<table>
<tr><th>Bit</th><th>Name</th><th>Value</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td>7</td><td>SCLKME</td><td></td><td>Sample rate generator input clock mode bit. The sample rate generator can produce a clock signal, CLKG. The frequency of CLKG is:
CLKG freq. = (Input clock frequency) / (CLKGDV + 1)
SCLKME is used in conjunction with the CLKSM bit to select the input clock.</td><td>R/W</td><td>0</td></tr>
<tr><td></td><td></td><td>0</td><td>The input clock for the sample rate generator is taken from the CLKS pin or from the Internal clock, depending on the value of the CLKSM bit in SRGR2:
<table>
<tr><th>SCLKME</th><th>CLKSM</th><th>Input Clock For
Sample Rate Generator</th></tr>
<tr><td>0</td><td>0</td><td>Signal on CLKS pin</td></tr>
<tr><td>0</td><td>1</td><td>Internal clock</td></tr>
</table></td><td></td><td></td></tr>
<tr><td></td><td></td><td>1</td><td>The input clock for the sample rate generator is taken from the CLKR pin or from the CLKX pin, depending on the value of the CLKSM bit in SRGR2:
<table>
<tr><th>SCLKME</th><th>CLKSM</th><th>Input Clock For
Sample Rate Generator</th></tr>
<tr><td>1</td><td>0</td><td>Signal on CLKR pin</td></tr>
<tr><td>1</td><td>1</td><td>Signal on CLKX pin</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>6</td><td>CLKSSTAT</td><td></td><td>CLKS pin status bit. When CLKSSTAT is applicable, it reflects the level on the CLKS pin.
CLKSSTAT is only applicable when the transmitter and receiver are both in reset (XRST = RRST = 0) and CLKS is configured for use as a general-purpose input pin (XIOEN = RIOEN = 1).</td><td>R</td><td>0</td></tr>
<tr><td></td><td></td><td>0</td><td>The signal on the CLKS pin is low.</td><td></td><td></td></tr>
<tr><td></td><td></td><td>1</td><td>The signal on the CLKS pin is high.</td><td></td><td></td></tr>
<tr><td>5</td><td>DXSTAT</td><td></td><td>DX pin status bit. When DXSTAT is applicable, you can toggle the signal on DX by writing to DXSTAT. DXSTAT is only applicable when the transmitter is in reset (XRST = 0) and DX is configured for use as a general-purpose output pin (XIOEN = 1).</td><td>R/W</td><td>0</td></tr>
<tr><td></td><td></td><td>0</td><td>Drive the signal on the DX pin low.</td><td></td><td></td></tr>
<tr><td></td><td></td><td>1</td><td>Drive the signal on the DX pin high.</td><td></td><td></td></tr>
</table>

Table 98. Pin Control Register (PCR) (Continued)

Bit	Name	Value	Description	Type	Reset
4	DRSTAT		DR pin status bit. When DRSTAT is applicable, it reflects the level on the DR pin. DRSTAT is only applicable when the receiver is in reset (RRST = 0) and DR is configured for use as a general-purpose input pin (RIOEN = 1).	R	0
		0	The signal on DR pin is low.		
		1	The signal on DR pin is high.		
3	FSXP		Transmit frame-synchronization polarity bit. FSXP determines the polarity of FSX as seen on the FSX pin.	R/W	0
		0	Transmit frame-synchronization pulses are active high.		
		1	Transmit frame-synchronization pulses are active low.		
2	FSRP		Receive frame-synchronization polarity bit. FSRP determines the polarity of FSR as seen on the FSR pin.	R/W	0
		0	Receive frame-synchronization pulses are active high.		
		1	Receive frame-synchronization pulses are active low.		
1	CLKXP		Transmit clock polarity bit. CLKXP determines the polarity of CLKX as seen on the CLKX pin.	R/W	0
		0	Transmit data is sampled on the rising edge of CLKX.		
		1	Transmit data is sampled on the falling edge of CLKX.		
0	CLKRP		Receive clock polarity bit. CLKRP determines the polarity of CLKR as seen on the CLKR pin.	R/W	0
		0	Receive data is sampled on the falling edge of CLKR.		
		1	Receive data is sampled on the rising edge of CLKR.		

Table 99. Bit Configuration for GPIOs

Pin	General-Purpose Use Enabled By This Bit Combination	Selected as Output When …	Output Value Driven From This Bit	Selected as Input When…	Input Value Read From This Bit
CLKX	XRST = 0 XIOEN = 1	CLKXM = 1	CLKXP	CLKXM = 0	CLKXP
FSX	XRST = 0 XIOEN = 1	FSXM = 1	FSXP	FSXM = 0	FSXP
DX	XRST = 0 XIOEN = 1	Always	DXSTAT	Never	Does not apply
CLKR	RRST = 0 RIOEN = 1	CLKRM = 1	CLKRP	CLKRM = 0	CLKRP
FSR	RRST = 0 RIOEN = 1	FSRM = 1	FSRP	FSRM = 0	FSRP
DR	RRST = 0 RIOEN = 1	Never	Does not apply	Always	DRSTAT
CLKS	RRST = XRST = 0 RIOEN = XIOEN = 1	Never	Does not apply	Always	CLKSSTAT

7.9 Receive Channel Enable Registers (RCERA, RCERB, RCERC, RCERD, RCERE, RCERF, RCERG, RCERH)

Each McBSP has eight receive channel enable registers of the format shown in Figure 96. There is one enable register for each of the receive partitions: A, B, C, D, E, F, G, and H. Table 100 provides a summary description that applies to any bit x of a receive channel enable register.

These I/O-mapped registers are only used when the receiver is configured to allow individual enabling and disabling of the channels (RMCM = 1).

Figure 96. Receive Channel Enable Registers (RCERA...RCERH)

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = read; W = write; -*n* = value after reset

Table 100. Receive Channel Enable Registers (RCERA...RCERH)

Bit	Name	Value	Description
x	RCEx		Receive channel enable bit. The role of this bit depends on whether it is used to support the receive multichannel selection mode reception.
			For receive multichannel selection mode (RMCM = 1):
		0	Disable the channel that is mapped to RCEx.
		1	Enable the channel that is mapped to RCEx.

7.9.1 RCERs Used in the Receive Multichannel Selection Mode

For multichannel selection operation, the assignment of channels to the RCERs depends on whether 32 or 128 channels are individually selectable, as defined by the RMCME bit. For each of these two cases, Table 102 shows which block of channels is assigned to each of the RCERs used. For each RCER, the table shows which channel is assigned to each of the bits.

Table 101. Pin Control Register (PCR)

Bit	Name	Value	Description	Type	Reset
15	Reserved	0	Reserved bit (not available for your use). It is a read-only bit and returns a 0 when read.	R	0

Table 102. Use of the Receive Channel Enable Registers

Number of Selectable Channels	Block Assignments		Channel Assignments	
	RCERx	Block Assigned	Bit in RCERx	Channel Assigned
32 (RMCME = 0)	RCERA	Channels n to (n + 15) The block of channels is chosen with the RPABLK bits.	RCE0 RCE1 RCE2 : RCE15	Channel n Channel (n + 1) Channel (n + 2) : Channel (n + 15)
	RCERB	Channels m to (m + 15) The block of channels is chosen with the RPBBLK bits.	RCE0 RCE1 RCE2 : RCE15	Channel m Channel (m + 1) Channel (m + 2) : Channel (m + 15)

Table 102. Use of the Receive Channel Enable Registers (Continued)

Number of Selectable Channels	Block Assignments		Channel Assignments	
	RCERx	Block Assigned	Bit in RCERx	Channel Assigned
128 (RMCME = 1)	RCERA	Block 0	RCE0	Channel 0
			RCE1	Channel 1
			RCE2	Channel 2
			:	:
			RCE15	Channel 15
	RCERB	Block 1	RCE0	Channel 16
			RCE1	Channel 17
			RCE2	Channel 18
			:	:
			RCE15	Channel 31
	RCERC	Block 2	RCE0	Channel 32
			RCE1	Channel 33
			RCE2	Channel 34
			:	:
			RCE15	Channel 47
	RCERD	Block 3	RCE0	Channel 48
			RCE1	Channel 49
			RCE2	Channel 50
			:	:
			RCE15	Channel 63
	RCERE	Block 4	RCE0	Channel 64
			RCE1	Channel 65
			RCE2	Channel 66
			:	:
			RCE15	Channel 79
	RCERF	Block 5	RCE0	Channel 80
			RCE1	Channel 81
			RCE2	Channel 82
			:	:
			RCE15	Channel 95
	RCERG	Block 6	RCE0	Channel 96
			RCE1	Channel 97
			RCE2	Channel 98
			:	:
			RCE15	Channel 111
	RCERH	Block 7	RCE0	Channel 112
			RCE1	Channel 113
			RCE2	Channel 114
			:	:
			RCE15	Channel 127

7.10 Transmit Channel Enable Registers (XCERA, XCERB, XCERC, XCERD, XCERE, XCERF, XCERG, XCERH)

Each McBSP has eight transmit channel enable registers of the form shown in Figure 97. There is one for each of the transmit partitions: A, B, C, D, E, F, G, and H. Table 103 provides a summary description that applies to each bit XCEx of a transmit channel enable register.

The I/O-mapped XCERs are only used when the transmitter is configured to allow individual disabling/enabling and masking/unmasking of the channels (XMCM is nonzero).

Figure 97. Transmit Channel Enable Registers (XCERA...XCERH)

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = read; W = write; -*n* = value after reset

Table 103. Transmit Channel Enable Registers (XCERA...XCERH)

Bit	Name	Value	Description
x	XCEx		Transmit channel enable bit. The role of this bit depends on which transmit multichannel selection mode is selected with the XMCM bits.
			For multichannel selection when XMCM = 01b (all channels disabled unless selected):
		0	Disable and mask the channel that is mapped to XCEx.
		1	Enable and unmask the channel that is mapped to XCEx.
			For multichannel selection when XMCM = 10b (all channels enabled but masked unless selected):
		0	Mask the channel that is mapped to XCEx.
		1	Unmask the channel that is mapped to XCEx.
			For multichannel selection when XMCM = 11b (all channels masked unless selected):
		0	Mask the channel that is mapped to XCEx. Even if the channel is enabled by the corresponding receive channel enable bit, this channel's data cannot appear on the DX pin.
		1	Unmask the channel that is mapped to XCEx. If the channel is also enabled by the corresponding receive channel enable bit, full transmission can occur.

7.10.1 XCERs Used in a Transmit Multichannel Selection Mode

For multichannel selection operation, the assignment of channels to the XCERs depends on whether 32 or 128 channels are individually selectable, as defined by the XMCME bit. These two cases are shown in Table 104. The table shows which block of channels is assigned to each XCER that is used. For each XCER, the table shows which channel is assigned to each of the bits.

Note:

When XMCM = 11b (for symmetric transmission and reception), the transmitter uses the receive channel enable registers (RCERs) to enable channels and uses the XCERs to unmask channels for transmission.

Table 104. Use of the Transmit Channel Enable Registers in a Transmit Multichannel Selection Mode

Number of Selectable Channels	**Block Assignments**		**Channel Assignments**	
	XCERx	**Block Assigned**	**Bit in XCERx**	**Channel Assigned**
32 (XMCME = 0)	XCERA	Channels n to (n + 15) When XMCM = 01b or 10b, the block of channels is chosen with the XPABLK bits. When XMCM = 11b, the block is chosen with the RPABLK bits.	XCE0 XCE1 XCE2 : XCE15	Channel n Channel (n + 1) Channel (n + 2) : Channel (n + 15)
	XCERB	Channels m to (m + 15) When XMCM = 01b or 10b, the block of channels is chosen with the XPBBLK bits. When XMCM = 11b, the block is chosen with the RPBBLK bits.	XCE0 XCE1 XCE2 : XCE15	Channel m Channel (m + 1) Channel (m + 2) : Channel (m + 15)
128 (XMCME = 1)	XCERA	Block 0	XCE0 XCE1 XCE2 : XCE15	Channel 0 Channel 1 Channel 2 : Channel 15
	XCERB	Block 1	XCE0 XCE1 XCE2 : XCE15	Channel 16 Channel 17 Channel 18 : Channel 31
	XCERC	Block 2	XCE0 XCE1 XCE2 : XCE15	Channel 32 Channel 33 Channel 34 : Channel 47

Table 104. Use of the Transmit Channel Enable Registers in a Transmit Multichannel Selection Mode (Continued)

Number of Selectable Channels	**Block Assignments**		**Channel Assignments**	
	XCERx	**Block Assigned**	**Bit in XCERx**	**Channel Assigned**
	XCERD	Block 3	XCE0	Channel 48
			XCE1	Channel 49
			XCE2	Channel 50
			:	:
			XCE15	Channel 63
	XCERE	Block 4	XCE0	Channel 64
			XCE1	Channel 65
			XCE2	Channel 66
			:	:
			XCE15	Channel 79
	XCERF	Block 5	XCE0	Channel 80
			XCE1	Channel 81
			XCE2	Channel 82
			:	:
			XCE15	Channel 95
	XCERG	Block 6	XCE0	Channel 96
			XCE1	Channel 97
			XCE2	Channel 98
			:	:
			XCE15	Channel 111
	XCERH	Block 7	XCE0	Channel 112
			XCE1	Channel 113
			XCE2	Channel 114
			:	:
			XCE15	Channel 127

Revision History

Table 105 lists the changes made since the previous version of this document.

Table 105. Document Revision History

Page	Additions/Modifications/Deletions
Global	Entire document updated for new content model for reference guides. New sections added for examples. Index was removed.