TMS320VC5501/5502/5509/5510 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide

Literature Number: SPRU592A
December 2002
Revised February 2003

Texas Instruments

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 that 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.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Read This First

About This Manual

This manual describes the type of multichannel buffered serial ports (McBSP) available on the TMS320C55x™ DSPs. The McBSPs provide a direct serial interface between a C55x™ DSP and other devices in a system. For the number of McBSPs available, see the data sheet for your C55x DSP.

Notational Conventions

This document uses the following conventions.

- In most cases, hexadecimal numbers are shown with the suffix h. For example, the following number is a hexadecimal 40 (decimal 64):

  40h

  Similarly, binary numbers often are shown with the suffix b. For example, the following number is the decimal number 4 shown in binary form:

  0100b

- If a signal or pin is active low, it has an overbar. For example, the $\overline{\text{RESET}}$ signal is active low.

Related Documentation From Texas Instruments

The following documents describe the C55x devices and related support tools. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

TMS320C55x Technical Overview (literature number SPRU393). This overview is an introduction to the TMS320C55x DSPs, the latest generation of fixed-point DSPs in the TMS320C5000™ DSP platform. Like the previous generations, this processor is optimized for high performance and low-power operation. This book describes the CPU architecture, low-power enhancements, and embedded emulation features.

TMS320C55x DSP CPU Reference Guide (literature number SPRU371) describes the architecture, registers, and operation of the CPU for the TMS320C55x DSPs.

TMS320C55x DSP Peripherals Reference Guide (literature number SPRU317) describes the peripherals, interfaces, and related hardware that are available on TMS320C55x DSPs.

TMS320C55x DSP Algebraic Instruction Set Reference Guide (literature number SPRU375) describes the TMS320C55x DSP algebraic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the mnemonic instruction set.

TMS320C55x DSP Mnemonic Instruction Set Reference Guide (literature number SPRU374) describes the TMS320C55x DSP mnemonic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the algebraic instruction set.

TMS320C55x Optimizing C/C++ Compiler User's Guide (literature number SPRU281) describes the TMS320C55x™ C/C++ Compiler. This C/C++ compiler accepts ISO standard C and C++ source code and produces assembly language source code for TMS320C55x devices.

TMS320C55x Assembly Language Tools User's Guide (literature number SPRU280) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for TMS320C55x devices.

TMS320C55x DSP Programmer's Guide (literature number SPRU376) describes ways to optimize C and assembly code for the TMS320C55x DSPs and explains how to write code that uses special features and instructions of the DSPs.

Trademarks

TMS320, TMS320C5000, TMS320C55x, and C55x are trademarks of Texas Instruments.

Trademarks are the property of their respective owners.

Contents

Figures

Tables

Chapter 1

Introduction to McBSP

This chapter offers an introduction on multichannel buffered serial ports (McBSPs) for the TMS320C55x DSPs.

Topic **Page**

1.1 Introduction

The TMS320C55x DSPs provide multiple high-speed, multichannel buffered serial ports (McBSPs) that allow direct interface to other C55x DSPs, codecs, and other devices in a system. For the number of McBSPs available, see the data sheet for your C55x DSP.

1.2 Key Features of the McBSP

The McBSP provides:

- Full-duplex communication
- Double-buffered transmission and triple-buffered reception, which allow a continuous data stream
- Independent clocking and framing for reception and for transmission
- The capability to send interrupts to the CPU and to send DMA events to the DMA controller
- 128 channels for transmission and for reception
- Multichannel selection modes that enable you to allow or block transfers in each of the channels
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices
- Support for external generation of clock signals and frame-synchronization (frame-sync) signals
- A programmable sample rate generator for internal generation and control of clock signals and frame-sync signals
- Programmable polarity for frame-sync pulses and for clock signals
- Direct interface to:
  - T1/E1 framers
  - MVIP switching compatible and ST-BUS compliant devices including:
    - MVIP framers
    - H.100 framers
    - SCSA framers
  - IOM-2 compliant devices

  - ■ AC97 compliant devices (The necessary multiphase frame capability is provided.)
  - ■ IIS compliant devices
  - ■ SPI devices

- ❑ 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.

- ❑ μ-law and A-law companding

- ❑ The option of transmitting/receiving 8-bit data with the LSB first

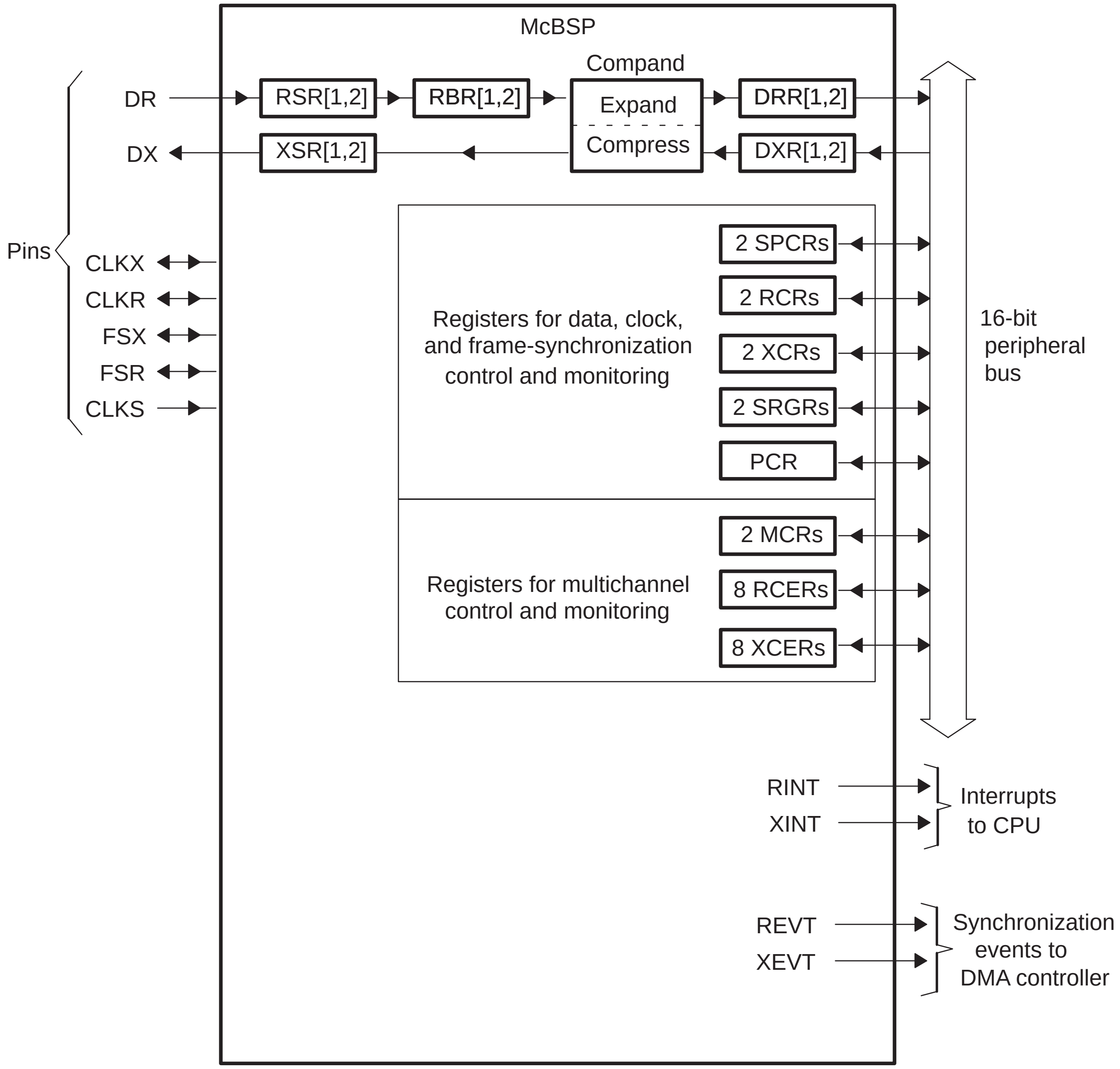
- ❑ Status bits for flagging exception/error conditions

- ❑ The capability to use the McBSP pins as general-purpose I/O pins

1.3 Block Diagram of the McBSP

The McBSP consists of a data-flow path and a control path connected to external devices by seven pins as shown in Figure 1−1.

Figure 1−1. Conceptual Block Diagram of the McBSP



Data is communicated to devices interfaced with the McBSP via the data transmit (DX) pin for transmission and the data receive (DR) pin for reception. Control information in the form of clocking and frame synchronization is communicated via the following pins: CLKX (transmit clock), CLKR (receive clock), FSX (transmit frame sync), and FSR (receive frame sync).

The CPU and the DMA controller communicate with the McBSP through 16-bit-wide registers accessible via the internal peripheral bus. The CPU or the 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 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 CPU or the 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 1−1 are registers for controlling McBSP operation. Details about these registers are available in Chapter 12, *McBSP Registers*.

1.4 McBSP Pins

The following table describes the McBSP interface pins. In the Type column, I = Input, O = Output, Z = High impedance.

Pin	Possible States	Possible Uses
CLKR	I/O/Z	Supplying or reflecting the receive clock; supplying the input clock of the sample rate generator; general-purpose I/O
CLKX	I/O/Z	Supplying or reflecting the transmit clock; supplying the input clock of the sample rate generator; general-purpose I/O
CLKS	I	Supplying the input clock of the sample rate generator; general-purpose input CLKS is not available on all devices and/or packages. Refer to the device−specific data manual for information on CLKS support. Devices that do not support CLKS also do not support any of the functions associated with CLKS.
DR	I	Receiving serial data; general-purpose input
DX	O/Z	Transmitting serial data; general-purpose output
FSR	I/O/Z	Supplying or reflecting the receive frame-sync signal; controlling sample rate generator synchronization for the case when GSYNC = 1
FSX	I/O/Z	Supplying or reflecting the transmit frame-sync signal; general-purpose I/O

On some C55x DSPs, some McBSP interface pins may be multiplexed with other pin functions. See the device-specific data manual for more information.

Chapter 2

McBSP Operation

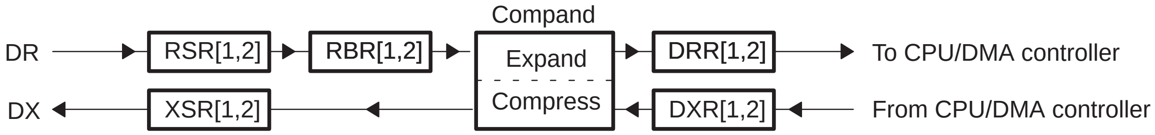
This chapter details the operation of the McBSP; the way the McBSP transmits or receives all data.

Topic **Page**

2.1 Data Transfer Processor of a McBSP

Figure 2−1 shows a diagram of the McBSP data transfer paths. 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 fits in 16 bits.

Figure 2−1. McBSP Data Transfer Paths



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). Once a full word is received, the content of RSR1 is copied to receive buffer register 1 (RBR1), only if RBR1 is not full with previous data. RBR1 is then copied to data receive register 1 (DRR1), unless the previous content of DRR1 has not been read by the CPU or the DMA controller. If the companding feature of the McBSP is implemented, the required word length is 8 bits and receive data is expanded into the appropriate format before being passed from RBR1 to DRR1.

Transmit data is written by the CPU or the DMA controller to data transmit register 1 (DXR1). If there is no previous data 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.

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 into RSR2 and RSR1. Once the full word is received, the contents of RSR2 and RSR1 are copied to RBR2 and RBR1, respectively, only 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 CPU or the DMA controller. The CPU or the DMA controller must read data from DRR2 first and then from DRR1. When DRR1 is read, the next RBR-to-DRR copy occurs.

For reception, the RJUST field in register SPCR1 controls the way the received data is aligned in DRR2 and DRR1. For transmission, the CPU or the 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.

2.2 Companding (Compressing and Expanding) Data

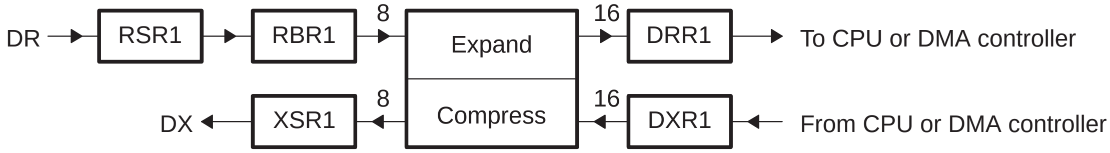
Companding (COMpressing and exPANDing) hardware allows compression and expansion of data in either μ-law or A-law format. The companding standard employed in the United States and Japan is μ-law. The European companding standard is referred to as A-law. The specifications for μ-law and A-law log PCM 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 CPU or 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 set to 8−bit mode. 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 2−2 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 2s-complement format.

Figure 2−2. Companding Processes



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, 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 2−3.

Figure 2−3. 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 2−4.

Figure 2−4. A-Law Transmit Data Companding Format

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

2.2.2 Capability to Compand Internal Data

If the McBSP is otherwise unused (the serial port transmit and receive sections are reset), the companding hardware can compand internal data. This can be used to:

- Convert linear to the appropriate μ-law or A-law format.
- Convert μ-law or A-law to the linear format.
- Observe the quantization effects in companding by transmitting linear data, and compressing and re-expanding this data. This is useful only if both XCOMPAND and RCOMPAND enable the same companding format.

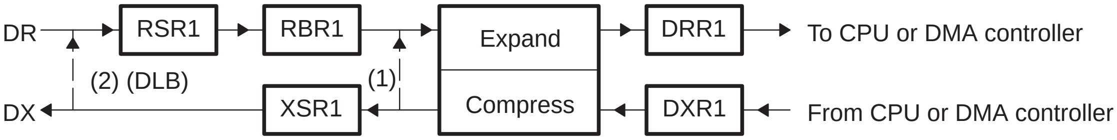
Figure 2−5 shows two methods by which the McBSP can compand internal data. Data paths for these two methods are used to indicate:

- When both the transmit and receive sections of the serial port are reset, DRR1 and DXR1 are connected internally through the companding logic. Values from DXR1 are compressed, as selected by XCOMPAND, and then expanded, as selected by RCOMPAND. Note that RRDY and XRDY bits are not set. However, data is available in DRR1 within four CPU clocks after being written to DXR1. This method is indicated as (1) in Figure 2−5 below.

  The advantage of this method is its speed. The disadvantage is that there is no synchronization available to the CPU and DMA to control the flow. Note that DRR1 and DXR1 are internally connected if the (X/R)COMPAND bits are set to 10b or 11b (compand using μ-law or A-law).

- The McBSP is enabled in digital loopback mode with companding appropriately enabled by RCOMPAND and XCOMPAND. Receive and transmit interrupts (RINT when RINTM = 0 and XINT when XINTM = 0) or synchronization events (REVT and XEVT) allow synchronization of the CPU or DMA to these conversions, respectively. Here, the time for this companding depends on the serial bit rate selected. This method is indicated as (2) in Figure 2−5 below.

Figure 2−5. Two Methods by Which the McBSP Can Compand Internal Data



2.2.3 Reversing Bit Order: Option to Transfer 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 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 set to 0, 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.

2.3 Clocking and Framing Data

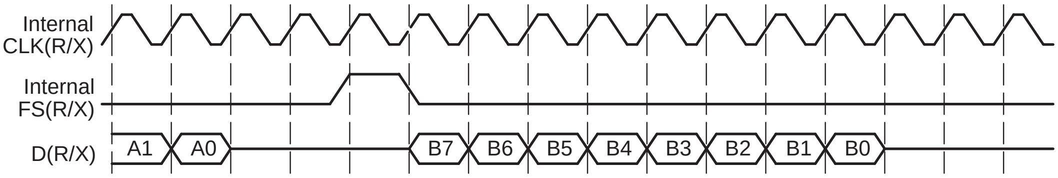
This section explains basic concepts and terminology important for understanding how McBSP data transfers are timed and delimited.

2.3.1 Clocking

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 following example, the clock signal controls the timing of each bit transfer on the pin.



Note:

The maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. For more information on programming the frequency of the slow peripheral clock, see the device-specific data manual for detailed information on the McBSP timing requirements.

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 (CLKGDV).

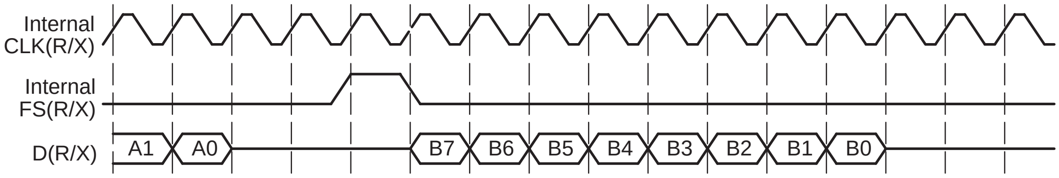
2.3.2 Serial Words

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 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.

In the following example, an 8-bit word size was defined (see bits 7 through 0 of word B being transferred).



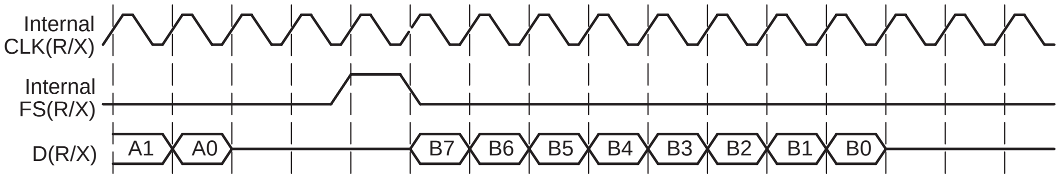
2.3.3 Frames and Frame Synchronization

One or more words are transferred in a group called a **frame**. You 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 (frame-sync) signals to determine when each frame is received/transmitted. When a pulse occurs on a frame-sync 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-sync signal (FSR) initiate frame transfers on DR. Pulses on the transmit frame-sync signal (FSX) 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 the following example, a 1-word frame is transferred when a frame-sync 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-sync signal may be high for an arbitrary number of clock cycles. Only after the signal is recognized to have gone inactive, and then active again, does the next frame synchronization occur.

2.3.4 Detecting Frame-Sync Pulses, Even in the Reset State

The McBSP can send receive and transmit interrupts to the CPU to indicate specific events in the McBSP. To facilitate detection of frame synchronization, these interrupts can be sent in response to frame-sync 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 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 CPU clock and then sent to the CPU in the form of RINT and XINT at the point at which they feed the receiver and transmitter of the serial port. Consequently, a new frame-synchronization pulse can be detected, and after this occurs the CPU can take the serial port out of reset safely.

2.3.5 Ignoring Unexpected Frame-Sync Pulses

The McBSP can be configured to ignore transmit and/or receive frame-synchronization pulses. To have the receiver or transmitter recognize frame-sync pulses, clear the appropriate frame-sync ignore bit (RFIG = 0 for the receiver, XFIG = 0 for the transmitter). To have the receiver or transmitter ignore frame-sync pulses until the desired frame length or number of words is reached, set the appropriate frame-sync ignore bit (RFIG = 1 for the receiver, XFIG = 1 for the transmitter).

2.3.6 Frame Frequency

The frame frequency is determined by the period between frame-synchronization pulses and is defined as shown by Equation 2−1.

Equation 2−1. Frame Frequency of a McBSP

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

The frame frequency may 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.

2.3.7 Maximum Frame Frequency

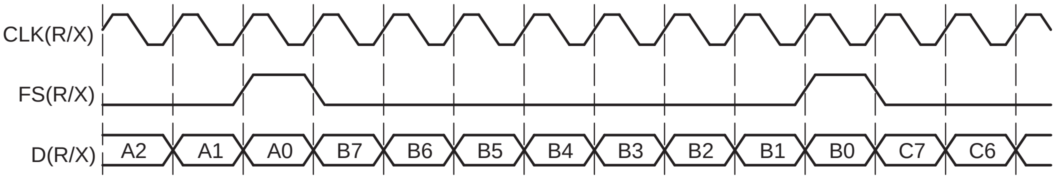
The minimum number of clock cycles between frame synchronization pulses is equal to the number of bits transferred per frame. The maximum frame frequency is defined as shown by Equation 2−2.

Equation 2−2. Maximum Frame Frequency of a McBSP

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

Figure 2−6 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 2−6. McBSP Operating at Maximum Packet Frequency



If there is a 1-bit data delay as shown in this figure, the frame-synchronization pulse overlaps the last bit transmitted in the previous frame. Effectively, this permits a continuous stream of data, making frame-synchronization pulses redundant. Theoretically, only an initial frame-synchronization pulse is required to initiate a multipacket transfer.

The McBSP supports operation of the serial port in this fashion by ignoring the successive frame-sync pulses. Data is clocked in to the receiver, or clocked out of the transmitter, during every clock cycle.

Note:

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

Note:

On the TMS320VC5501/5502, when external clocks are used for receive or transmit, one cycle is required between words. So the maximum packet frequency shown above is not possible.

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 user might define a frame 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 user to compose frames for custom applications, or in general, to maximize the efficiency of data transfers.

2.4.1 Number of Phases, Words, and Bits Per Frame

Table 2−1 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. The maximum number of bits (serial port clock cycles) per frame is 4096.

Table 2−1. McBSP Register Bits That Determine the Number of Phases, Words, and Bits Per Frame

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

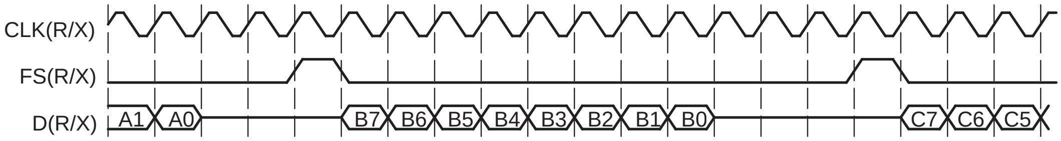
2.4.2 Single-Phase Frame Example

Figure 2−7 shows an example of a single-phase data frame comprising one 8-bit word. Since 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-sync signals
- ❑ (R/X)DATDLY = 01b: 1-bit data delay

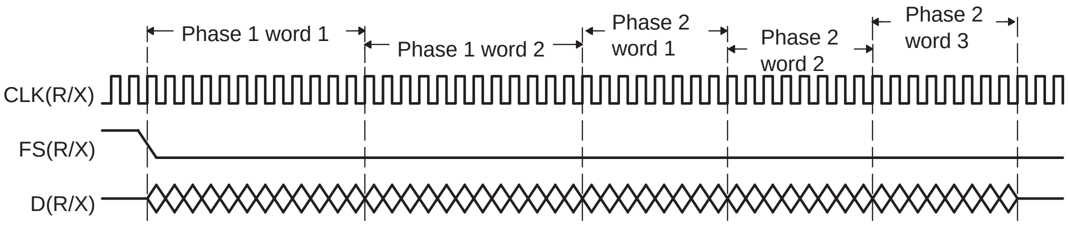
Figure 2−7. Single-Phase Frame for a McBSP Data Transfer



2.4.3 Dual-Phase Frame Example

Figure 2−8 shows an example of a frame where the first phase consists of 2 words of 12 bits each, followed by a second phase of three words of 8 bits each. Note that the entire bit stream in the frame is contiguous. There are no gaps either between words or between phases.

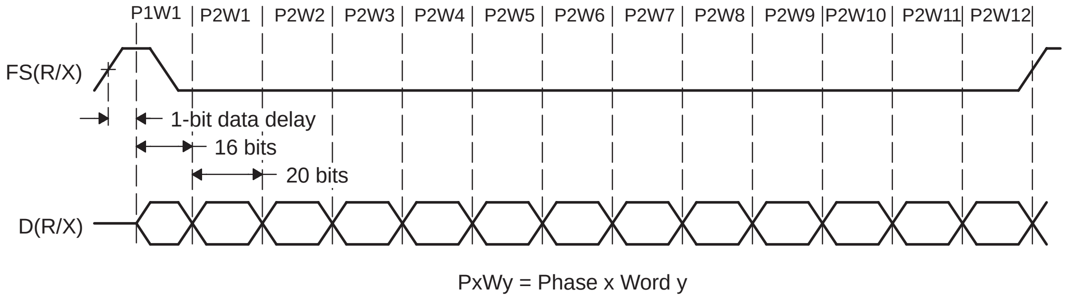
Figure 2−8. Dual-Phase Frame for a McBSP Data Transfer



2.4.4 Implementing the AC97 Standard with a Dual-Phase Frame

Figure 2−9 shows an example of the Audio Codec '97 (AC97) standard, which uses the dual-phase frame feature. Notice that words, not individual bits, are shown on the D(R/X) signal. The first phase (P1) consists of a single 16-bit word. The second phase (P2) consists of twelve 20-bit words. The phase configurations are listed after the figure.

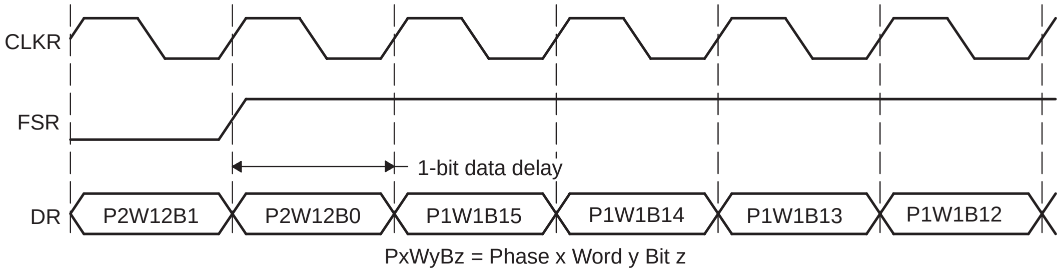
Figure 2−9. Implementing the AC97 Standard with a Dual-Phase Frame



- ❑ (R/X)PHASE = 1: Dual-phase frame
- ❑ (R/X)FRLEN1 = 0000000b: 1 word in phase 1
- ❑ (R/X)WDLEN1 = 010b: 16 bits per word in phase 1
- ❑ (R/X)FRLEN2 = 0001011b: 12 words in phase 2
- ❑ (R/X)WDLEN2 = 011b: 20 bits per word in phase 2
- ❑ CLKRP/CLKXP= 0: Receive data sampled on falling edge of internal CLKR / transmit data clocked on rising edge of internal CLKX
- ❑ FSRP/FSXP = 0: Active-high frame-sync signal
- ❑ (R/X)DATDLY = 01b: Data delay of 1 clock cycle (1-bit data delay)

Figure 2−10 shows the timing of an AC97-standard data transfer near frame synchronization. In this figure, individual bits are shown on D(R/X). Specifically, the figure shows the last two bits of phase 2 of one frame and the first four bits of phase 1 of the next frame. Regardless of the data delay, data transfers can occur without gaps. The first bit of the second frame (P1W1B15) immediately follows the last bit of the first frame (P2W12B0). Because a 1-bit data delay has been chosen, the transition on the frame-sync signal can occur when P2W12B0 is transferred.

Figure 2−10. Timing of an AC97-Standard Data Transfer Near Frame Synchronization



Note:

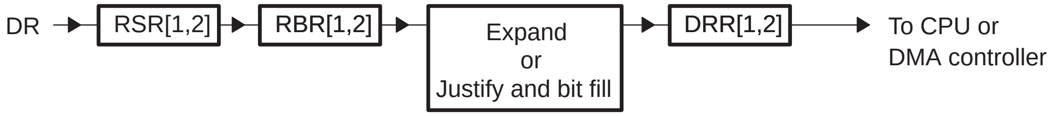
On the TMS320VC5501/5502, when external clocks are used for receive or transmit, one cycle is required between words. So the maximum packet frequency shown above is not possible.

2.5 McBSP Reception

This section explains the fundamental process of reception in the McBSP. For more details on how to configure the receiver, see Chapter 7, *Receiver Configuration*.

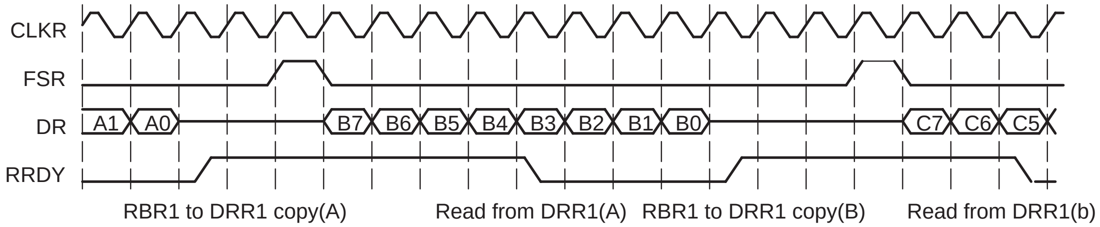
Figure 2−11 and Figure 2−12 show how reception occurs in the McBSP. Figure 2−11 shows the physical path for the data. Figure 2−12 is a timing diagram showing signal activity for one possible reception scenario. A description of the process follows the figures.

Figure 2−11. 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 2−12. McBSP Reception Signal Activity



CLKR: Internal receive clock
FSR: Internal receive frame-sync signal
DR: Data on DR pin
RRDY: Status of receiver ready bit

The following process describes how data travels from the DR pin to the CPU or to the DMA controller:

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

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

   In the preceding timing diagram (Figure 2−12), 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.

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 CPU or the 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 CPU or the 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 needed (word length larger than 16 bits), the CPU or the 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 the following topics in Chapter 4 for more details:

❑ *Overrun in the Receiver*

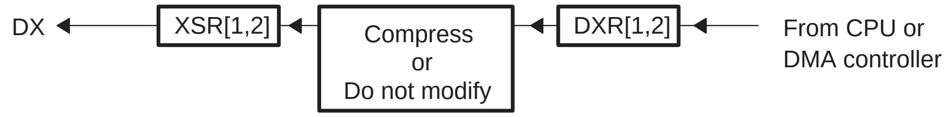
❑ *Unexpected Receive Frame-Sync Pulse*

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 Chapter 8, *Transmitter Configuration*.

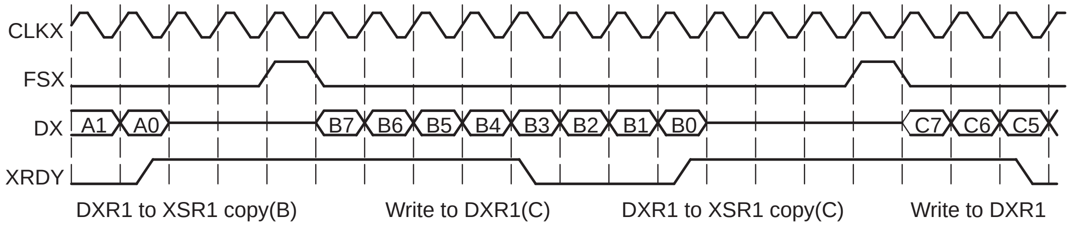
Figure 2−13 and Figure 2−14 show how transmission occurs in the McBSP. Figure 2−13 shows the physical path for the data. Figure 2−14 is a timing diagram showing signal activity for one possible transmission scenario. A description of the process follows the figures.

Figure 2−13. McBSP Transmission Physical Data Path



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

Figure 2−14. McBSP Transmission Signal Activity



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

1) The CPU or the 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.

Note:

If both DXRs are needed (word length larger than 16 bits), the CPU or the 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 CPU or the 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-sync pulse on internal FSX.

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

   In the preceding timing diagram (Figure 2−14), 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 the following topics in Chapter 4 for more details:

❑ *Overwrite in the Transmitter*

❑ *Underflow in the Transmitter*

❑ *Unexpected Transmit Frame-Sync Pulse*

2.7 Interrupts and DMA Events Generated by a McBSP

The McBSP sends notification of important events to the CPU and DMA controller via the internal signals shown in Table 2−2.

Table 2−2. Interrupts and DMA Events Generated by a McBSP

Internal Signal	Description
RINT	Receive interrupt The McBSP can send a receive interrupt request to CPU based upon a selected condition in the receiver of the McBSP (a condition selected by the RINTM bits of SPCR1).
XINT	Transmit interrupt The McBSP can send a transmit interrupt request to CPU based upon a selected condition in the transmitter of the McBSP (a condition selected by the XINTM bits of SPCR2).
REVT	Receive synchronization event An REVT signal is sent to the DMA controller when data has been received in the data receive registers (DRRs).
XEVT	Transmit synchronization event An XEVT signal is sent to the DMA controller when the data transmit registers (DXRs) are ready to accept the next serial word for transmission.

Chapter 3

Sample Rate Generator of the McBSP

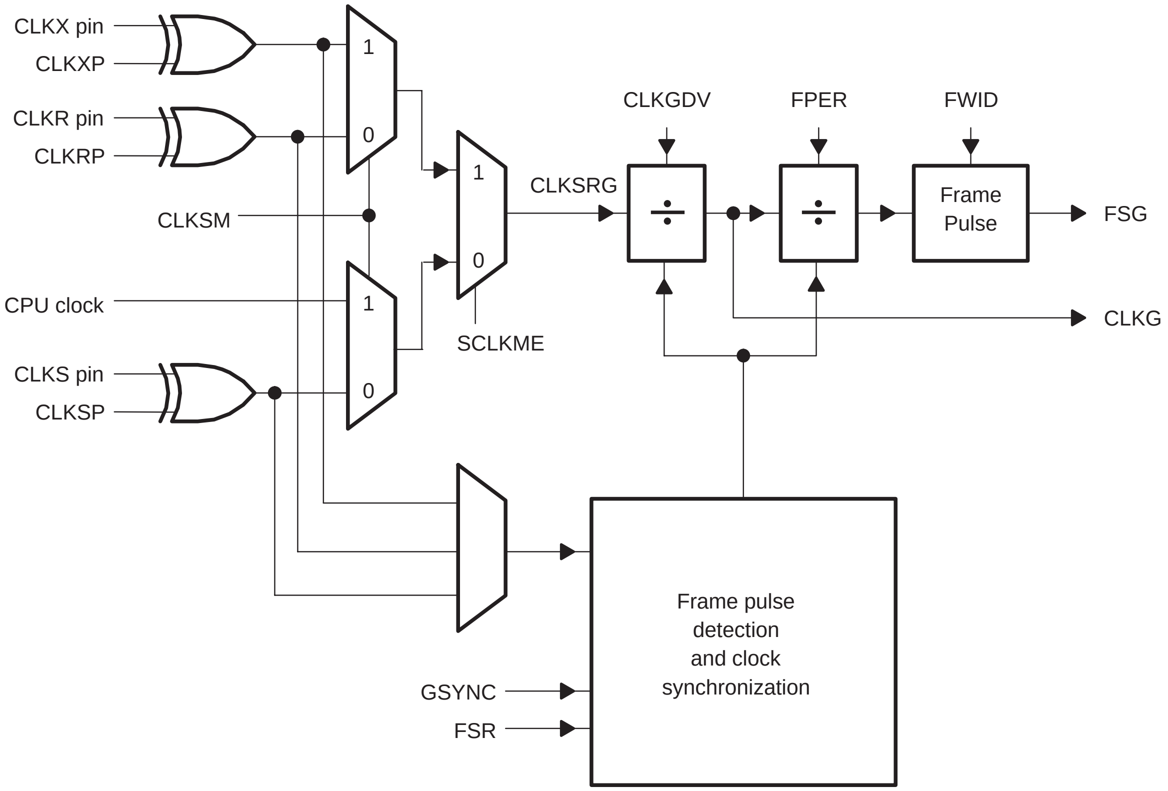
This chapter gives information on the use of the sample rate generator to drive clocking, and provides the appropriate clocking examples for support.

Topic **Page**

3.1 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 3−1 is a conceptual block diagram of the sample rate generator.

Figure 3−1. Conceptual Block Diagram of the Sample Rate Generator



The source clock for the sample rate generator (labeled CLKSRG in the diagram) can be supplied by the CPU clock or by an external pin (CLKS, CLKX, or CLKR). The source is selected with the SCLKME bit of PCR and the CLKSM bit of SRGR2. If a pin is used, the polarity of the incoming signal can be inverted with the appropriate polarity bit (CLKSP of SRGR2, CLKXP of PCR, or CLKRP of PCR).

The sample rate generator has a 3-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 of SRGR1 to produce CLKG.
- ❑ Frame period divide down. CLKG is divided according to the FPER bits of SRGR2 to control the period from the start of a frame-sync pulse to the start of the next pulse.
- ❑ Frame-sync pulse width countdown. CLKG cycles are counted according to the FWID bits of SRGR1 to control the width of each frame-sync pulse.

Note:

The maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. For more information on programming the frequency of the slow peripheral clock, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* data manual (SPRS206) or the *TMS320VC5502 Fixed-Point Digital Signal Processor* data manual (SPRS166). Other timing limitations may also apply. See the device-specific data manual for detailed information on the McBSP timing requirements.

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 (CLKGDV).

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

3.2 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 to 1 (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).

Note that the effects of CLKRM = 1 and CLKXM = 1 on the McBSP are partially affected by the use of the digital loopback mode and the clock stop (SPI) mode, respectively. The digital loopback mode is selected with the DLB bit of SPCR1. The clock stop mode is selected with the CLKSTP bits of SPCR1.

When using the sample rate generator as a clock source, make sure the sample rate generator is enabled (GRST = 1).

Table 3−1. 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.

3.2.1 Choosing an Input Clock

The sample rate generator must be driven by an input clock signal from one of the four sources selectable with the SCLKME bit of PCR and the CLKSM bit of SRGR2 (see Table 3−2). When CLKSM = 1, the minimum divide down value in CLKGDV bits should be 1.

Note:

The maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. For more information on programming the frequency of the slow peripheral clock, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* data manual (SPRS206) or the *TMS320VC5502 Fixed-Point Digital Signal Processor* data manual (SPRS166). Other timing limitations may also apply. See the device-specific data manual for detailed information on the McBSP timing requirements.

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 (CLKGDV).

Table 3−2. 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	CPU clock
1	0	Signal on CLKR pin
1	1	Signal on CLKX pin

3.2.2 Choosing a Polarity for the Input Clock

As shown in Figure 3−2, when the input clock is received 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. The polarity options and their effects are described in Table 3−3.

Figure 3−2. Possible Inputs to the Sample Rate Generator and The Polarity Bits

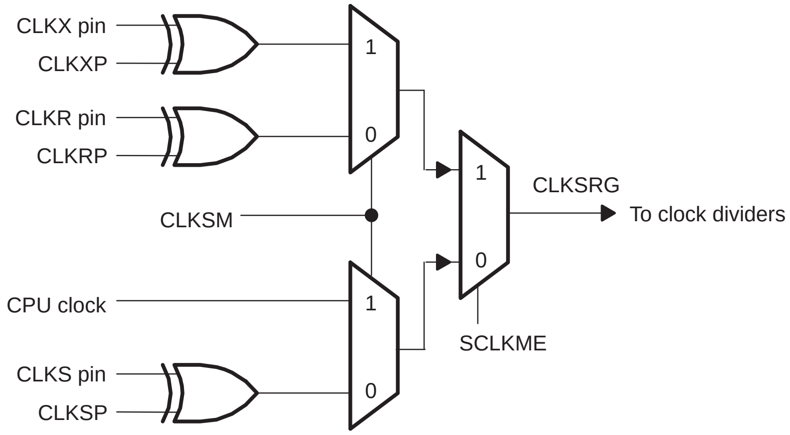


Table 3−3. Polarity Options for the Input to the Sample Rate Generator

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

3.2.3 Choosing a Frequency for the Output Clock (CLKG)

The input clock (CPU clock 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 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 of SRGR1. The output of this stage is the data clock (CLKG). CLKG has the frequency represented by the following equation.

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

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 maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. For more information on programming the frequency of the slow peripheral clock, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* data manual (SPRS206) or the *TMS320VC5502 Fixed-Point Digital Signal Processor* data manual (SPRS166). Other timing limitations may also apply. See the device-specific data manual for detailed information on the McBSP timing requirements.

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 (CLKGDV).

3.2.4 Keeping CLKG Synchronized to an External Input Clock

When an external signal is selected to drive the sample rate generator, 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.

3.3 Frame Sync Generation in the Sample Rate Generator

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

If you want the **receiver** to use FSG for frame synchronization, set 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:

- ❑ 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-sync 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 (GRST = 1) and the frame-sync logic in the sample rate generator must be enabled (FRST = 1).

3.3.1 Choosing the Width of the Frame-Sync Pulse on FSG

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

3.3.2 Controlling the Period Between the Starting Edges of Frame-Sync 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-sync period is controlled by an external device.
- ❑ Otherwise, you program the FPER bits of SRGR2, and the resulting frame-sync period is (FPER + 1) CLKG cycles, where CLKG is the output clock of the sample rate generator.

3.4 Synchronizing Sample Rate Generator Outputs to an External Clock

The sample rate generator can produce a clock signal (CLKG) and a frame-sync signal (FSG) based on an input clock signal that is either the CPU clock 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 of 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.

Make GSYNC = 1 when you want the McBSP and an external device to be synchronized 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 of SRGR2 are ignored because the frame-sync period on FSG is determined by the arrival of the next frame-sync pulse on the FSR pin.

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

3.4.1 Synchronization Examples

Figure 3−3 and Figure 3−4 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 1 CLKG cycle wide. The FPER bits of SRGR2 are not programmed; the period from the start of a frame-sync pulse to the start of the next pulse is determined by the arrival of the next inactive-to-active transition on the FSR pin. 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 of SRGR1).

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

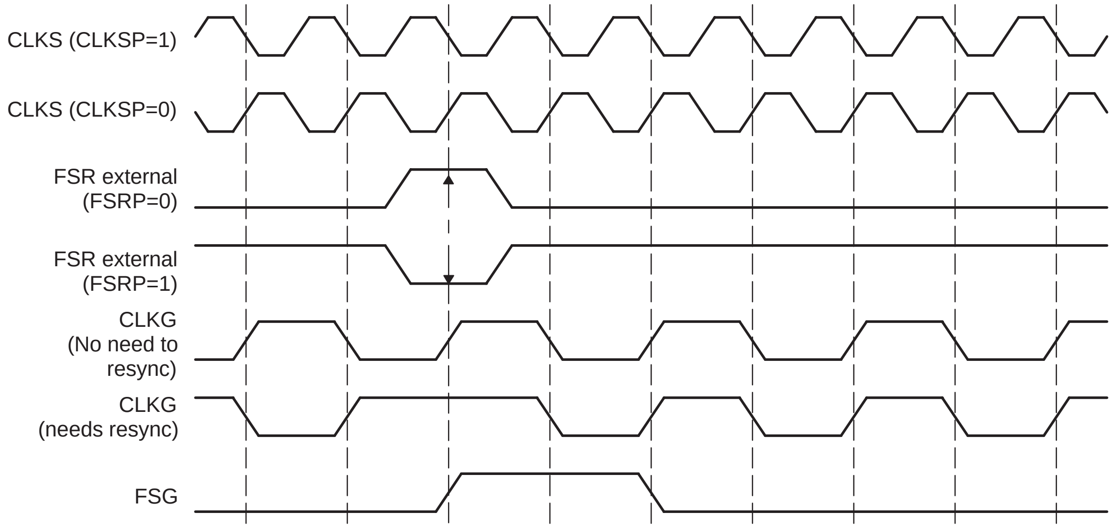
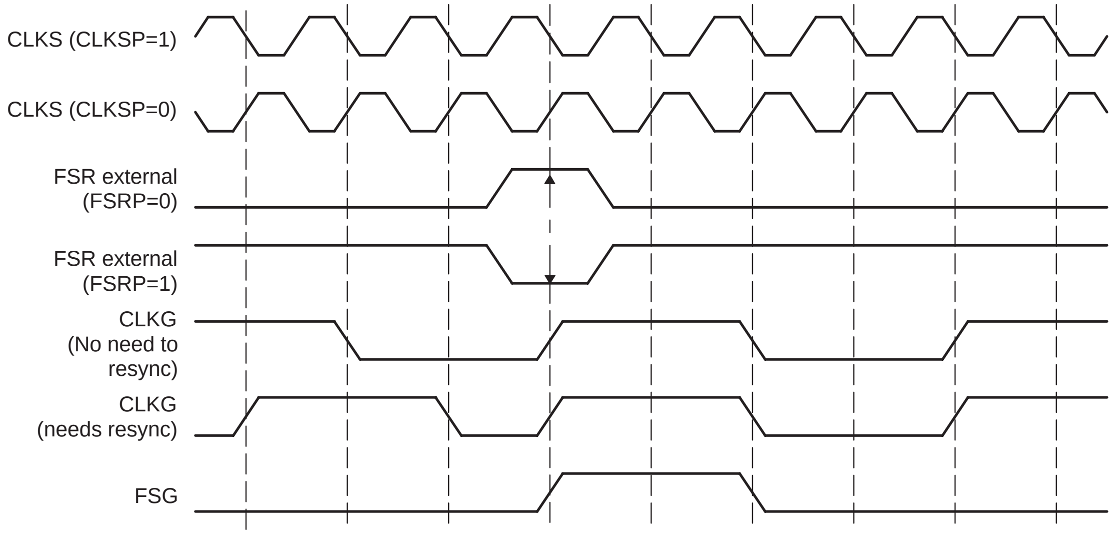


Figure 3−4. CLKG Synchronization and FSG Generation When GSYNC = 1 and CLKGDV = 3



3.5 Reset and Initialization Procedure for the Sample Rate Generator

To reset and initialize the sample rate generator:

1) **Place the McBSP/sample rate generator in reset.**

   During a DSP reset, the sample rate generator, the receiver, and the transmitter reset bits (GRST, RRST, and XRST) are automatically forced to 0. Otherwise, during normal operation, the sample rate generator can be reset by making GRST = 0 in SPCR2, provided that CLKG and/or FSG is not used by any portion of the McBSP. Depending on your system you may also want to reset the receiver (RRST = 0 in SPCR1) and reset the transmitter (XRST = 0 in SPCR2).

   If GRST = 0 due to a DSP reset, CLKG is driven by the CPU clock divided by 2, and FSG is driven inactive-low. If GRST = 0 due to program code, CLKG and FSG are driven low (inactive).

2) **Program 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 2 CLKSRG cycles. This ensures proper synchronization internally.

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

   In SPCR2, make GRST = 1 to enable the sample rate generator.

   After the sample rate generator is enabled, wait 2 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

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

   where the input clock is selected with the SCLKME bit of PCR and the CLKSM bit of SRGR2:

SCLKME	CLKSM	Input Clock For Sample Rate Generator
0	0	Signal on CLKS pin
0	1	CPU clock
1	0	Signal on CLKR pin
1	1	Signal on CLKX pin

4) **If necessary, enable the receiver and/or the transmitter.**

   If necessary, remove the receiver and/or transmitter from reset by setting RRST and/or XRST = 1.

5) **If necessary, enable the frame-sync logic of the sample rate generator.**

   After the required data acquisition setup is done (DXR[1/2] is loaded with data), set FRST = 1 in SPCR2 if an internally generated frame-sync pulse is required. FSG is generated with an active-high edge after the programmed number of CLKG clocks (FPER + 1) have elapsed.

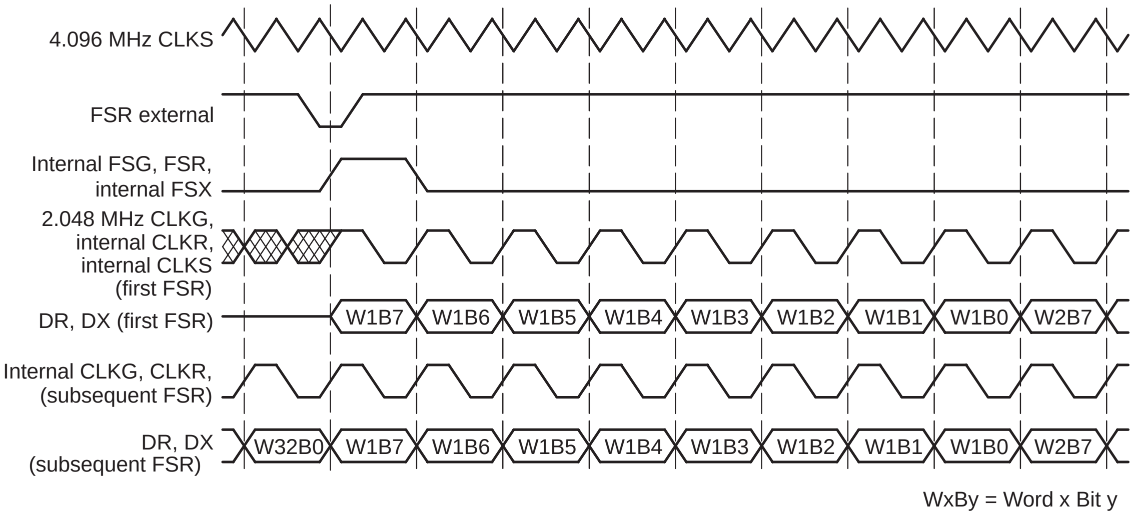
3.6 Sample Rate Generator Clocking Examples

This section shows three examples of using the sample rate generator to clock data during transmission and reception.

3.6.1 Double-Rate ST-Bus Clock

Figure 3−5 shows McBSP configuration to be compatible with the Mitel ST-Bus. Note that this operation is running at maximum frame frequency.

Figure 3−5. ST-BUS and MVIP Clocking Example



For this McBSP configuration:

- ❑ DLB = 0: Digital loopback mode off, CLKSTP = 00b: Clock stop mode off, and CLKRM/CLKXM = 1: Internal CLKR/CLKX generated internally by sample rate generator
- ❑ GSYNC = 1: Synchronize CLKG with external frame-sync signal input on FSR pin. CLKG is not synchronized until the frame-sync signal is active. FSR is regenerated internally to form a minimum pulse width.
- ❑ SCLKME = 0 and CLKSM = 1: External clock signal at CLKS pin drives the sample rate generator
- ❑ CLKSP = 1: Falling edge of CLKS generates CLKG and thus internal CLK(R/X)
- ❑ CLKGDV = 1: Frequency of receive clock (shown as CLKR) is half CLKS frequency

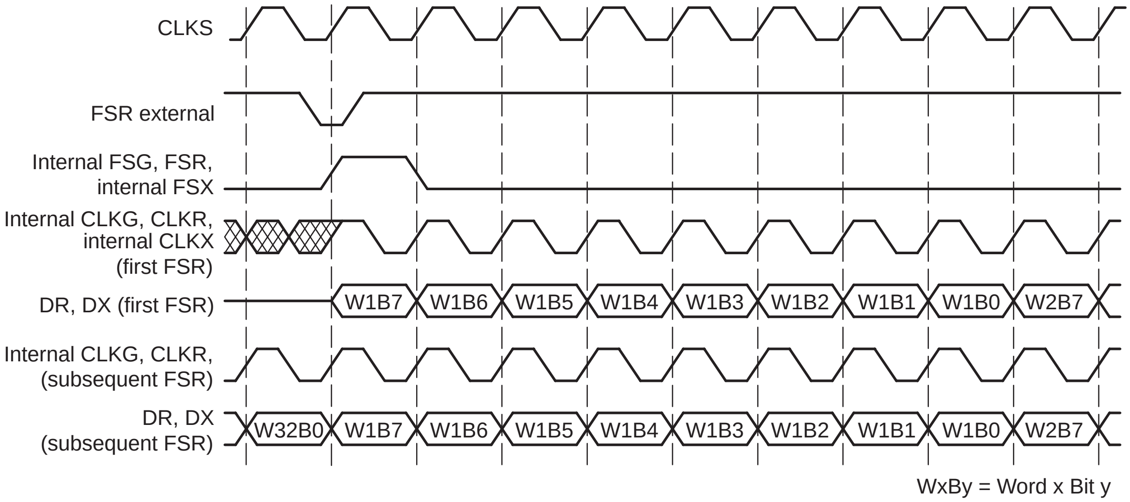
- ❑ FSRP/FSXP = 1: Active-low frame-sync pulse
- ❑ RFRLEN1/XFRLEN1 = 11111b: 32 words per frame
- ❑ RWDLEN1/XWDLEN1 = 0: 8 bits per word
- ❑ RPHASE/XPHASE = 0: Single-phase frame and thus (R/X)FRLEN2 and (R/X)WDLEN2 are ignored
- ❑ RDATDLY/XDATDLY = 0: No data delay

3.6.2 Single-Rate ST-Bus Clock

The example in Figure 3−6 is the same as the double-rate ST-bus clock example in section 3.6.1 except that:

- ❑ CLKGDV = 0: CLKS drives internal CLK(R/X) without any divide down (single-rate clock).
- ❑ CLKSP = 0: Rising edge of CLKS generates CLKG and internal CLK(R/X)

Figure 3−6. Single-Rate Clock Example



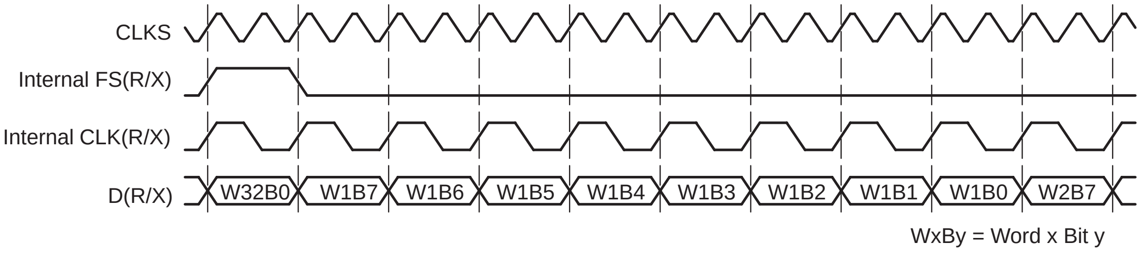
The rising edge of CLKS is used to detect the external FSR pulse, which is used to resynchronize internal McBSP clocks and generate a frame-sync pulse for internal use. The internal frame-sync pulse is generated so that it is wide enough to be detected on the falling edge of internal clocks.

3.6.3 Other Double-Rate Clock

The example in Figure 3−7 is the same as the double-rate ST-bus clock example in section 3.6.1 except that:

- ❑ CLKSP = 0: Rising edge of CLKS generates CLKG and thus CLK(R/X)
- ❑ CLKGDV = 1: Frequency of CLKG (and thus internal CLKR and internal CLKX) is half CLKS frequency
- ❑ FSRM/FSXM = 0: Frame synchronization is externally generated. The frame-sync pulse is wide enough to be detected.
- ❑ GSYNC = 0: CLKS drives CLKG. CLKG runs freely; it is not resynchronized by a pulse on the FSR pin.
- ❑ FSRP/FSXP = 0: Active-high input frame-sync signal
- ❑ RDATDLY/XDATDLY = 1: Data delay of one bit

Figure 3−7. Double-Rate Clock Example



Chapter 4

McBSP Exception/Error Conditions

This chapter provides a detailed explanation and listing of exception or error conditions associated with the McBSP.

Topic **Page**

4.1 McBSP Exception/Error Conditions

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

❑ **Receiver Overrun (RFULL = 1 in SPCR1)**. 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 may arrive at this time on DR will replace the contents of the RSR(s), and thus, the previous word is lost. The RSR(s) continue to be overwritten as long as new data arrives on DR and DRR1 is not read.

❑ **Unexpected Receive Frame-Sync Pulse (RSYNCERR = 1 in SPCR1)**. This occurs during reception when RFIG = 0 and an unexpected frame-sync pulse occurs. An unexpected frame-sync 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.

❑ **Transmitter Data Overwrite**. This occurs when the CPU or 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.

❑ **Transmitter Underflow (XEMPTY = 0 in SPCR2)**. If a new frame-sync signal arrives before new data is loaded into DXR1, the previous data in the DXR(s) is sent again. This will continue for every new frame-sync pulse that arrives until DXR1 is loaded with new data.

❑ **Unexpected Transmit Frame-Synch Pulse (XSYNCERR = 1 in SPCR2)**. This occurs during transmission when XFIG = 0 and an unexpected frame-sync pulse occurs. An unexpected frame-sync 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 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.

4.2 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:

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

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

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

As described in the section on McBSP reception, Chapter 13, 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 will 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 cycles before the end of the third word is shifted into the RSR1.

Important: If both DRRs are needed (word length larger than 16 bits), the CPU or the 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.

Note that 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:

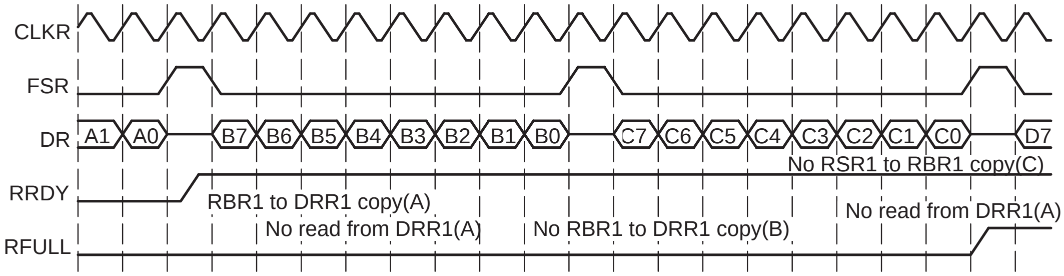
- The CPU or DMA controller reads DRR1.
- The receiver is reset individually (RRST = 0) or as part of a DSP reset.

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

4.2.1 Example of the Overrun Condition

Figure 4−1 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.

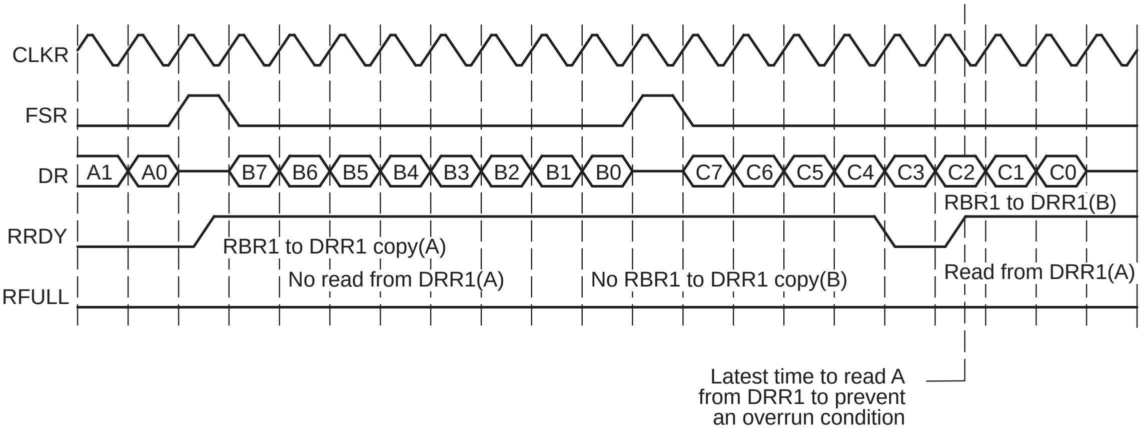
Figure 4−1. Overrun in the McBSP Receiver



4.2.2 Example of Preventing the Overrun Condition

Figure 4−2 shows the case where 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 4−2. Overrun Prevented in the McBSP Receiver



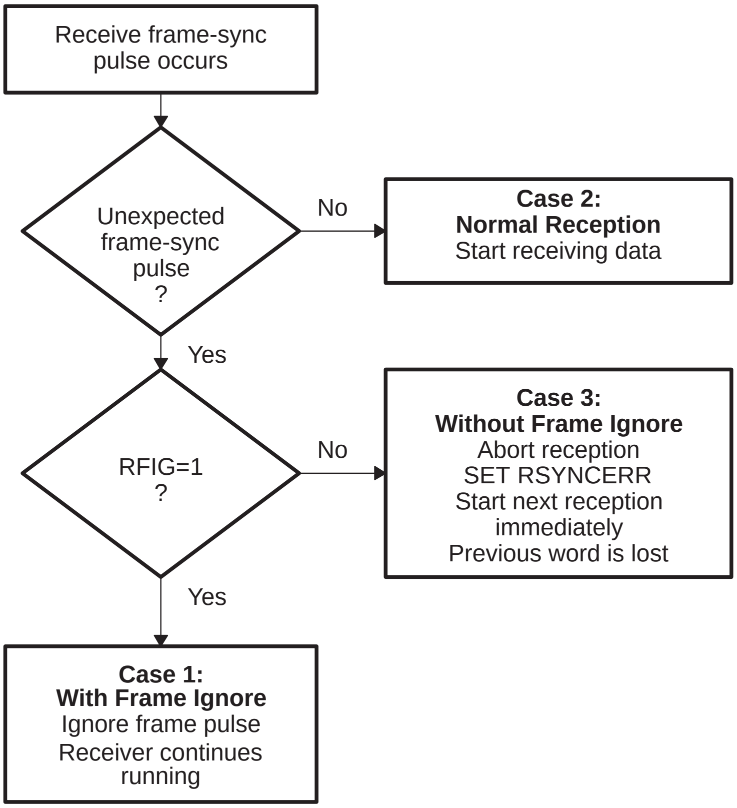
4.3 Unexpected Receive Frame-Sync Pulse

This section discusses how the McBSP responds to all receive frame-sync pulses, including an unexpected pulse. It also provides examples of a frame-sync error and an example of how to prevent such an error.

4.3.1 Possible Responses to Receive Frame-Sync Pulses

Figure 4−3 shows the decision tree that the receiver uses to handle all incoming frame-sync pulses. The figure assumes that the receiver has been started (RRST = 1 in SPCR1). Case 3 in the figure is the case in which an error occurs.

Figure 4−3. Possible Responses to Receive Frame-Sync 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-sync pulses are ignored, and the reception continues.
- **Case 2:** Normal serial port reception. Reception continues normally because the frame-sync 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 pulse after the receiver is enabled (RRST = 1 in SPCR1).
- The FSR pulse is the first pulse after DRR[1,2] is read, clearing a receiver full (RFULL = 1 in SPCR1) condition.
- The serial port is in the interpacket intervals. 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-sync pulses not ignored). Unexpected frame-sync pulses can originate from an external source or from the internal sample rate generator.

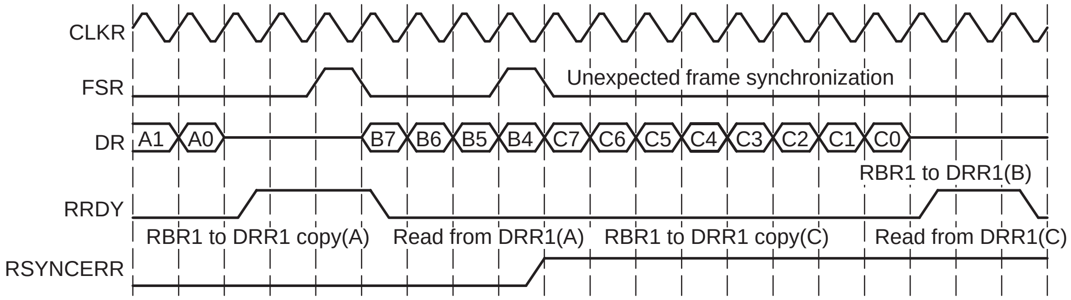
If a frame-sync pulse starts the transfer of a new frame before the current frame is fully received, this pulse is treated as an unexpected frame-sync pulse, and the receiver sets the receive frame-sync error bit (RSYNCERR) in SPCR1. RSYNCERR can be cleared only by a receiver reset or by a write of 0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

If you want the McBSP to notify the CPU of receive frame-sync errors, you can set a special receive interrupt mode with the RINTM bits of SPCR1. When RINTM = 11b, the McBSP sends a receive interrupt (RINT) request to the CPU each time that RSYNCERR is set.

4.3.2 Example of an Unexpected Receive Frame-Sync Pulse

Figure 4−4 shows an unexpected receive frame-sync pulse during normal operation of the serial port, with time intervals between data packets. When the unexpected frame-sync 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 (RINT) request to the CPU.

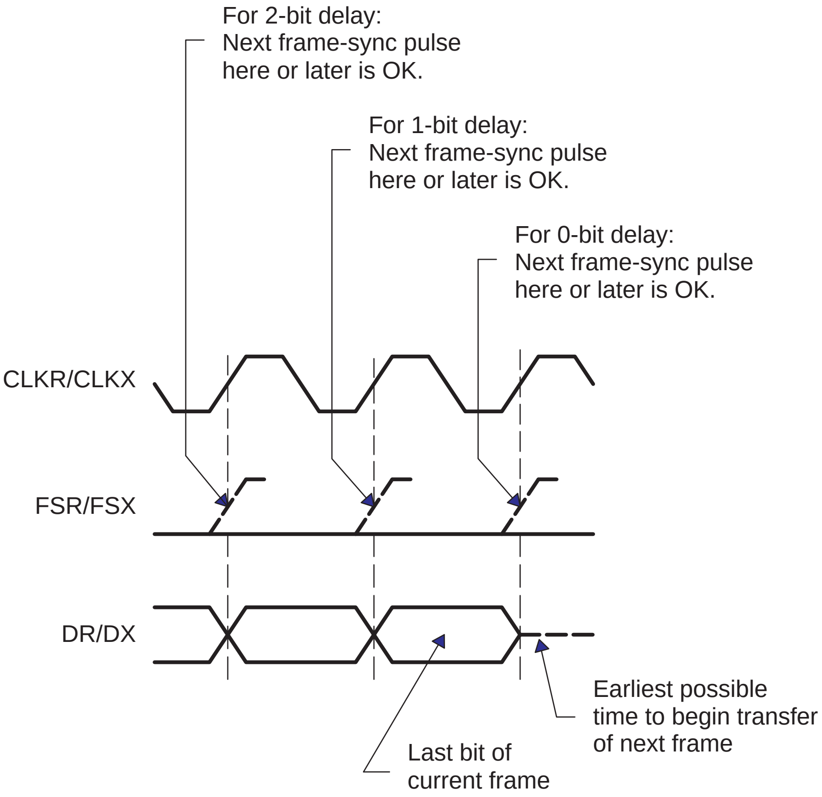
Figure 4−4. An Unexpected Frame-Sync Pulse During a McBSP Reception



4.3.3 Preventing Unexpected Receive Frame-Sync Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKR cycles, depending on the value in the RDATDLY bits of RCR2. For each possible data delay, Figure 4−5 shows when a new frame-sync pulse on FSR can safely occur relative to the last bit of the current frame.

Figure 4−5. Proper Positioning of Frame-Sync Pulses



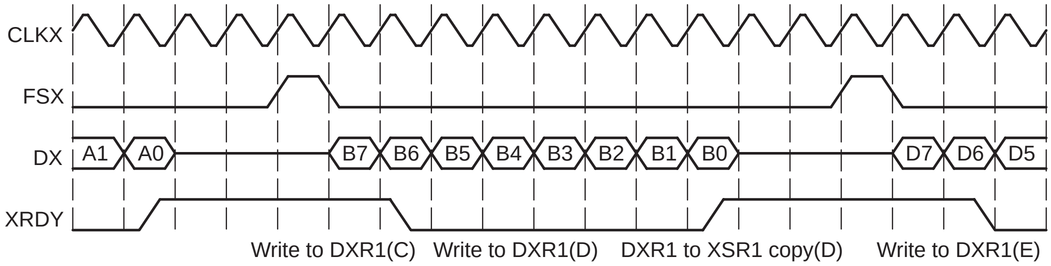
4.4 Overwrite in the Transmitter

After the CPU or DMA controller writes data to the DXR(s), the transmitter must then copy that data to 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.

4.4.1 Example of the Overwrite Condition

Figure 4−6 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 4−6. Data in the McBSP Transmitter Overwritten and Thus Not Transmitted



4.4.2 Preventing Overwrites

You can prevent CPU overwrites by making the CPU:

- ❑ 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 (XINT) before writing to the DXR(s). When XINTM = 00b in SPCR2, the transmitter sends XINT to the CPU each time XRDY is set.

You can prevent DMA overwrites by synchronizing DMA transfers to the transmit synchronization event XEVT. The transmitter sends an XEVT signal each time XRDY is set.

4.5 Underflow in the Transmitter

The McBSP indicates a transmitter empty (or underflow) condition by clearing the XEMPTY bit in SPCR2. Either of the following events activates XEMPTY (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 XRST = 0 in SPCR2, or by a DSP 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-sync signal until a new value is loaded into DXR1 by the CPU or the DMA controller.

Note:

If both DXRs are needed (word length larger than 16 bits), the CPU or the 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.

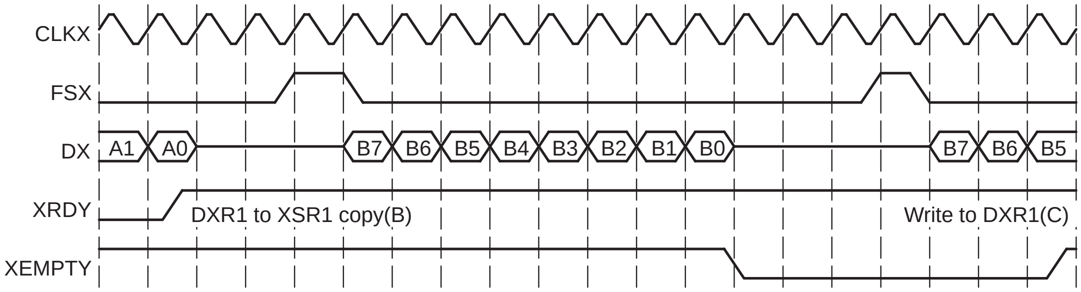
XEMPTY is deactivated (XEMPTY = 1) when a new word in DXR1 is transferred to XSR1. If FSXM = 1 in PCR and FSGM = 0 in SRGR2, the transmitter generates a single internal FSX pulse in response to a DXR-to-XSR copy. Otherwise, the transmitter waits for the next frame-sync pulse before sending out the next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

When the transmitter is taken out of reset (XRST = 1), it is in a transmitter ready (XRDY = 1 in SPCR2) and transmitter empty (XEMPTY = 0) state. If DXR1 is loaded by the CPU or the 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-sync pulse is generated or detected. Alternatively, if a transmit frame-sync pulse is detected before DXR1 is loaded, zeros will be output on DX.

4.5.1 Example of the Underflow Condition

Figure 4−7 shows an underflow condition. After B is transmitted, DXR1 is not reloaded before the subsequent frame-sync pulse. Thus, B is again transmitted on DX.

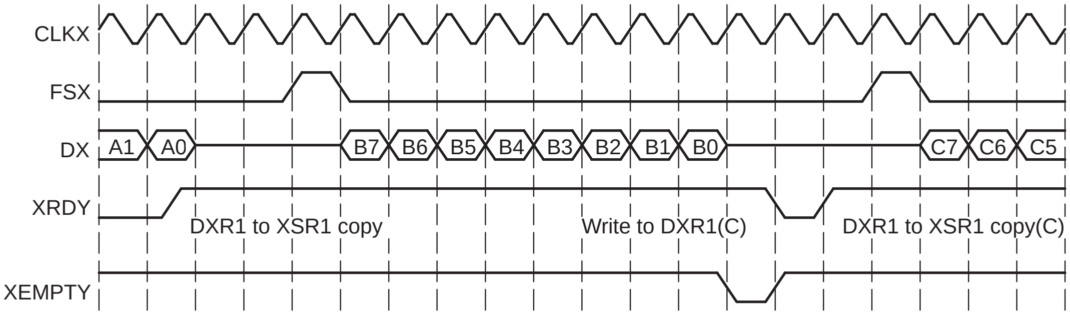
Figure 4−7. Underflow During McBSP Transmission



4.5.2 Example of Preventing the Underflow Condition

Figure 4−8 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-sync pulse. As a result, there is no underflow; B is not transmitted twice.

Figure 4−8. Underflow Prevented in the McBSP Transmitter



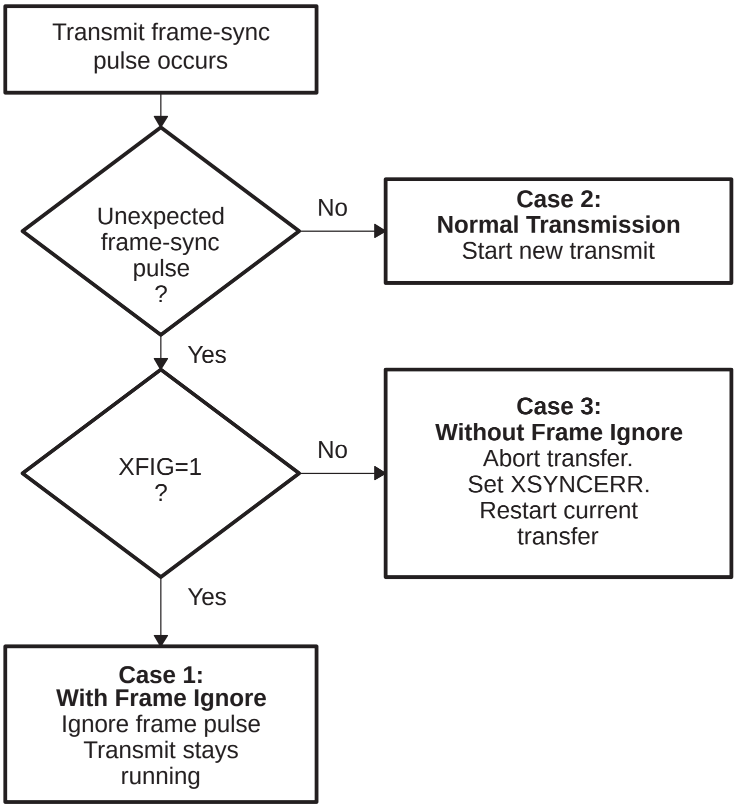
4.6 Unexpected Transmit Frame-Sync Pulse

This section discusses how the McBSP responds to any transmit frame-sync pulses, including an unexpected pulse. It also provides examples of a frame-sync error and an example of how to prevent such an error.

4.6.1 Possible Responses to Transmit Frame-Sync Pulses

Figure 4−9 shows the decision tree that the transmitter uses to handle all incoming frame-sync pulses. The figure assumes that the transmitter has been started (XRST = 1 in SPCR2). Case 3 in the figure is the case in which an error occurs.

Figure 4−9. Possible Responses to Transmit Frame-Sync 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. Unexpected transmit frame-sync pulses are ignored, and the transmission continues.

- **Case 2:** Normal serial port transmission. Transmission continues normally because the frame-sync 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 (XRST = 1).
  - The serial port is in the interpacket intervals. The programmed data delay for transmission (programmed with the XDATDLY bits of 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-sync pulses not ignored). Unexpected frame-sync pulses can originate from an external source or from the internal sample rate generator.

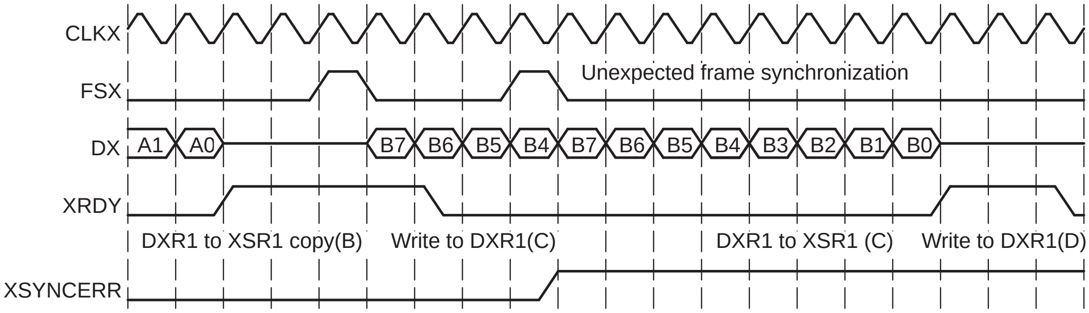
  If a frame-sync pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame-sync pulse, and the transmitter sets the transmit frame-sync error bit (XSYNCERR) in SPCR2. XSYNCERR can be cleared only by a transmitter reset or by a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

  If you want the McBSP to notify the CPU of frame-sync errors, you can set a special transmit interrupt mode with the XINTM bits of SPCR2. When XINTM = 11b, the McBSP sends a transmit interrupt (XINT) request to the CPU each time that XSYNCERR is set.

4.6.2 Example of an Unexpected Transmit Frame-Sync Pulse

Figure 4−10 shows an unexpected transmit frame-sync pulse during normal operation of the serial port, with intervals between the data packets. When the unexpected frame-sync pulse occurs, the XSYNCERR bit is set and because no new data has been passed to XSR1 yet, the transmission of data B is restarted. In addition, if XINTM = 11b, the McBSP sends a transmit interrupt (XINT) request to the CPU.

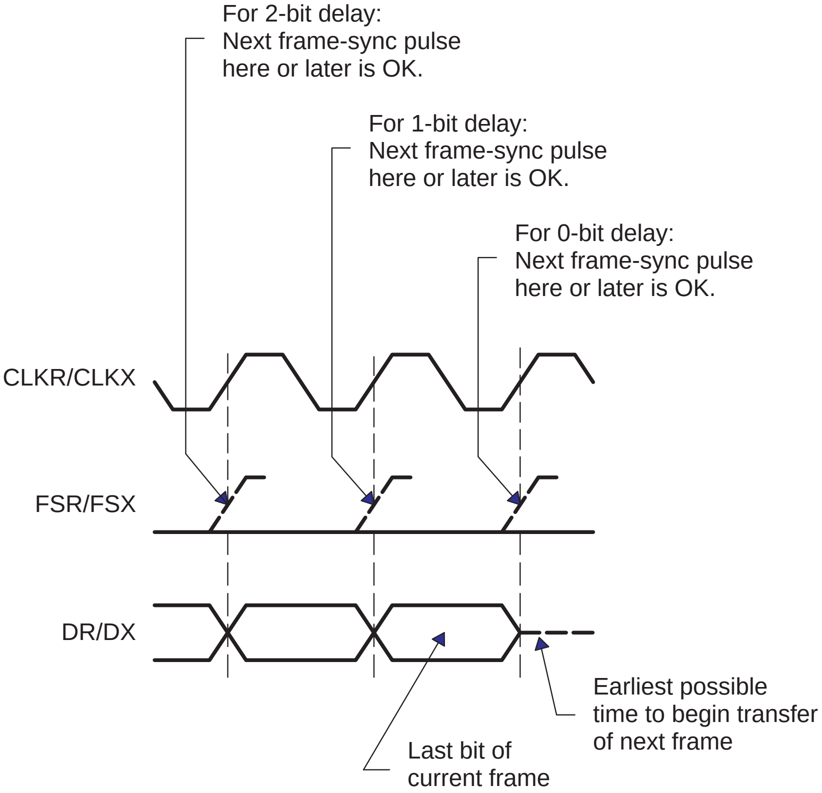
Figure 4−10. An Unexpected Frame-Sync Pulse During a McBSP Transmission



4.6.3 Preventing Unexpected Transmit Frame-Sync Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKX cycles, depending on the value in the XDATDLY bits of XCR2. For each possible data delay, Figure 4−11 shows when a new frame-sync pulse on FSX can safely occur relative to the last bit of the current frame.

Figure 4−11. Proper Positioning of Frame-Sync Pulses



Chapter 5

Multichannel Selection Modes

This chapter defines and provides the functions and all related information concerning the multichannel selection modes.

Topic **Page**

5.1 Channels, Blocks, and Partitions

A McBSP **channel** is a time slot for shifting in/out the bits of one serial word. 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 2-partition mode, 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, 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 2 receive partitions (A and B) and 8 transmit partitions (A–H).

5.2 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 and three transmit multichannel selection modes. These modes are described in Chapter 12.

5.3 Configuring a Frame for Multichannel Selection

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

- Select a single-phase frame (RPHASE/XPHASE = 0). Each frame represents a TDM data stream.
- Set a frame length (in RFRLEN1/XFRLEN1) that includes the highest-numbered channel that will 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). If XFRLEN1 = 39 in this case, the receiver creates 40 time slots per frame but only receives data during time slots 0, 15, and 39 of each frame.

5.4 Using Two Partitions

For multichannel selection operation in the receiver and/or the transmitter, you can use two partitions or eight partitions. 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-sync 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-sync pulse occurs, the next frame is transferred, beginning with the channels in partition A.

5.4.1 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, 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:

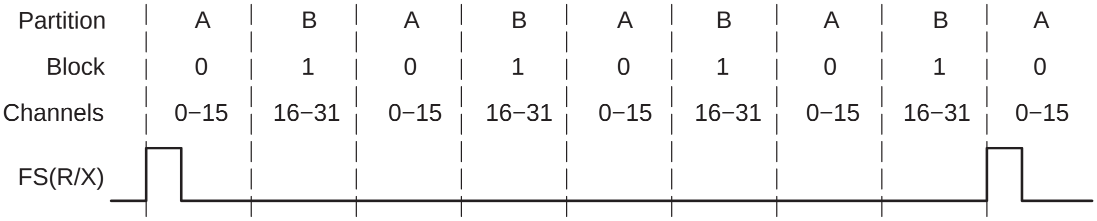
❑ Assign an even-numbered channel block (0, 2, 4, or 6) to transmit partition A by writing to the XPABLK bits. In one of the transmit multichannel selection modes, the channels in this partition are controlled by transmit channel enable register A (XCERA).

❑ 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 5−1 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-sync 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 5−1. Alternating Between the Channels of Partition A and the Channels of Partition B

2-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 next, you can dynamically change which blocks of channels are assigned to the partitions.

5.4.2 Reassigning Blocks During Reception/Transmission

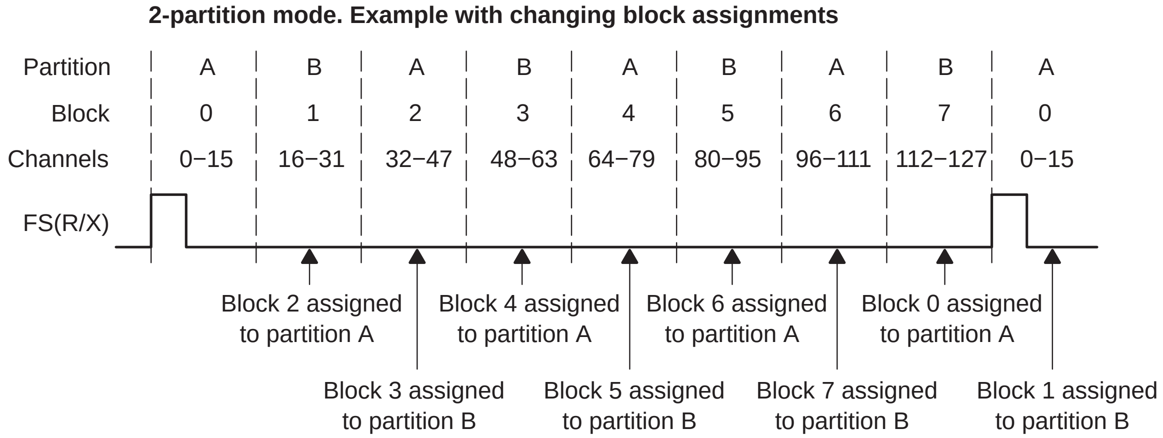
If you want to use more than 32 channels, you can change which channel blocks are assigned to partitions A and B during the course of a data transfer. However, these changes must be carefully timed. While a partition is being transferred, its the associated block assignment bits cannot be modified, and its associated channel enable register cannot be modified. For example, if block 3 is being transferred and block 3 is assigned to partition A, you cannot modify (R/X)PABLK to assign different channels to partition A, and you cannot modify (R/X)CERA to change the channel configuration for partition A. Several features of the McBSP 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 which partition is active. 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 CPU. In response to the interrupt, the CPU can then check the RCBLK/XCBLK bits and update the inactive partition.

Figure 5−2 shows an example of reassigning channels throughout a data transfer. In response to a frame-sync pulse, the McBSP alternates between partitions A and B. Whenever partition B is active, the CPU changes the block assignment for partition A. Whenever, partition A is active, the CPU changes the block assignment for partition B.

Figure 5−2. Reassigning Channel Blocks Throughout a McBSP Data Transfer


2-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

5.5 Using Eight Partitions

For multichannel selection operation in the receiver and/or the transmitter, you can use eight partitions or two partitions. If you choose the 8-partition mode (RMCME = 1 for reception, XMCME = 1 for transmission), McBSP partitions are activated in the following order: A, B, C, D, E, F, G, H. In response to a frame-sync 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-sync 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 5−1 and Table 5−2. These assignments cannot be changed. The tables also show the registers used to control the channels in the partitions.

Table 5−1. Receive Channel Assignment and Control When Eight Receive Partitions Are Used

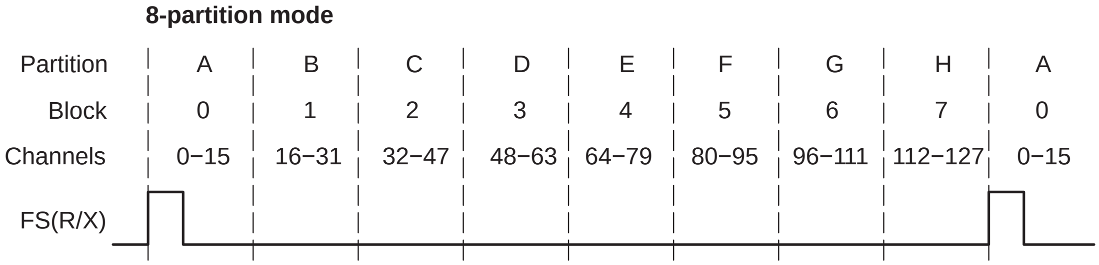
Receive Partition	Assigned Block of Receive Channels	Register Used For Channel Control
A	Block 0: channels 0 through 15	RCERA
B	Block 1: channels 16 through 31	RCERB
C	Block 2: channels 32 through 47	RCERC
D	Block 3: channels 48 through 63	RCERD
E	Block 4: channels 64 through 79	RCERE
F	Block 5: channels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

Table 5−2. Transmit Channel Assignment and Control When Eight Transmit Partitions Are Used

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 5−3 shows an example of the McBSP using the 8-partition mode. In response to a frame-sync 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 5−3. McBSP Data Transfer in the 8-Partition Mode



5.6 Receive Multichannel Selection Mode

The RMCM bit of MCR1 determines whether all channels or only selected 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 of 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 DMA synchronization event (REVT) is generated, and if the receiver interrupt mode depends on RRDY (RINTM = 00b), no interrupt is generated.

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

1) Accepts bits shifted in from the DR pin in channel 0

2) Ignores bits received in channels 1–14

3) Accepts bits shifted in from the DR pin in channel 15

4) Ignores bits received in channels 16–38

5) Accepts bits shifted in from the DR pin in channel 39

5.7 Transmit Multichannel Selection Mode

The XMCM bits of XCR2 determine whether all channels or only selected channels are enabled and unmasked for transmission. The McBSP has three transmit multichannel selection modes (XMCM = 01b, XMCM = 10b, and XMCM = 11b), which are described in the following table:

Table 5−3. 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 of 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 of 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 of 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 you have enabled only channels 0, 15, and 39. Suppose also that the frame length is 40. The McBSP…

1) Shifts data to the DX pin in channel 0
2) Places the DX pin in the high impedance state in channels 1–14
3) Shifts data to the DX pin in channel 15
4) Places the DX pin in the high impedance state in channels 16–38
5) Shifts data to the DX pin in channel 39

5.7.1 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 definitions explain 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 DX pin is held in the high impedance state; data cannot be shifted out on the DX 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 of SPCR2 is not set. Therefore, no DMA synchronization event (XEVT) is generated, and if the transmit interrupt mode depends on XRDY (XINTM = 00b in SPCR2), no interrupt is generated.

The XEMPTY bit of SPCR2 is not affected.

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

5.7.2 Activity on McBSP Pins for Different Values of XMCM

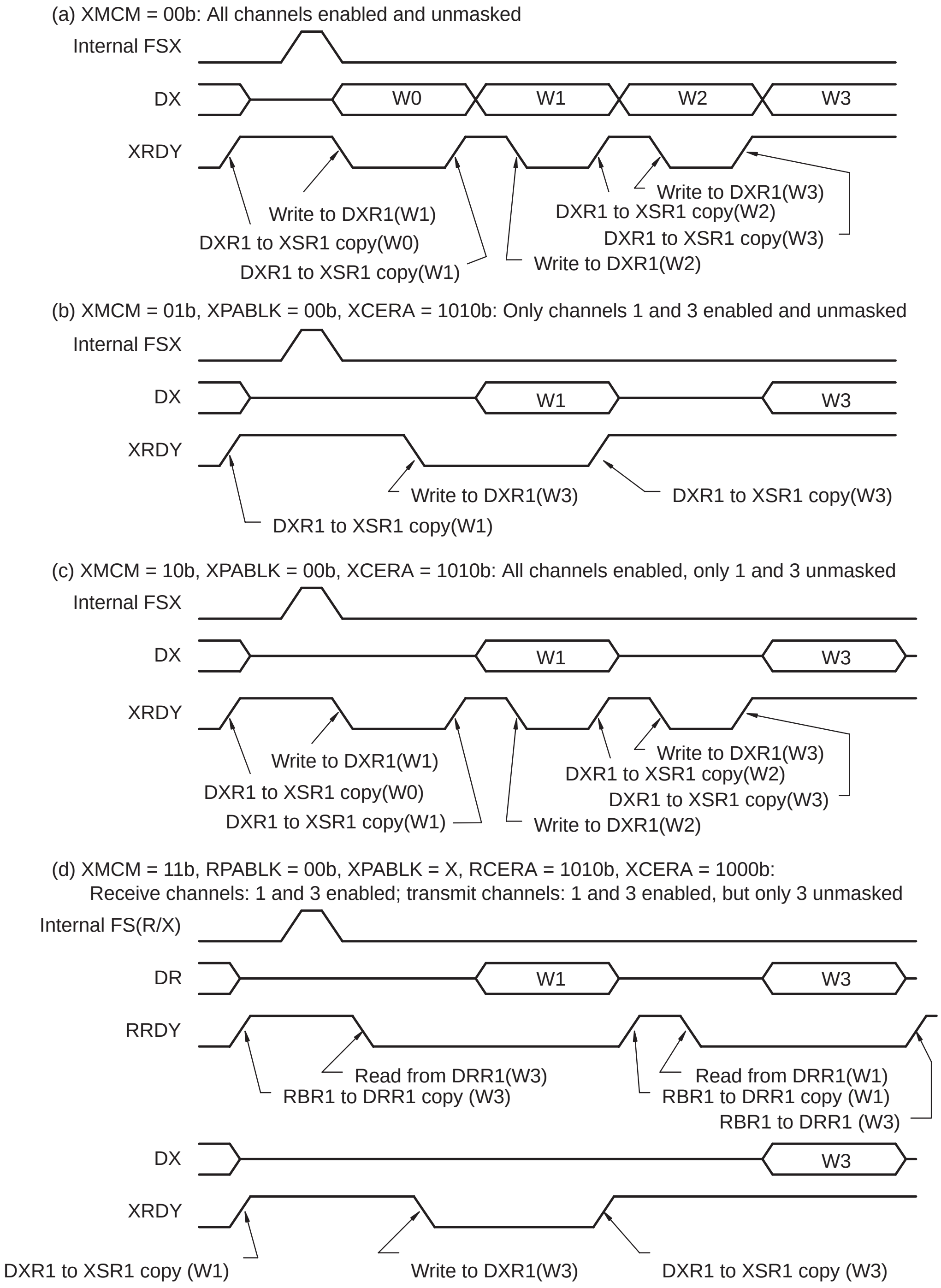
Figure 5−4 shows the activity on the McBSP pins for the various XMCM values. In all cases, the transmit frame is configured as follows:

- 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)

In the case where 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 the figure, 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 5−4. 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

5.8 Using Interrupts Between Block Transfers

When a multichannel selection mode is used, an interrupt request can be sent to the CPU 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 (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 (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 2 CPU clock cycles.

This type of interrupt is especially helpful if you are using the 2-partition mode and you want to know when you can assign a different block of channels to partition A or B.

Chapter 6

SPI Operation Using the Clock Stop Mode

This chapter describes the the function and relationship between the SPI interface and the McBSP.

Topic **Page**

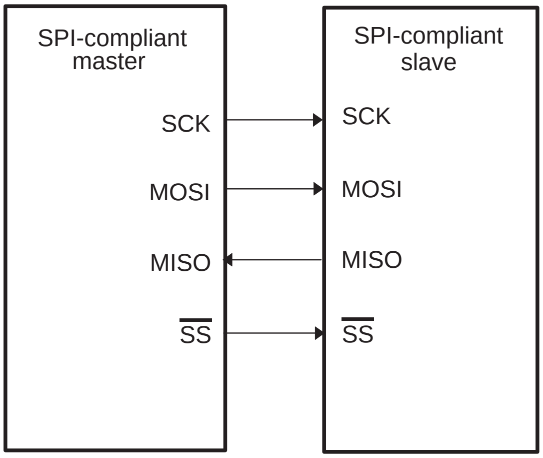
6.1 SPI Protocol

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 $\overline{\text{SS}}$)

A typical SPI interface with a single slave device is shown in Figure 6−1.

Figure 6−1. 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 (device not sending out the clock).

In the absence of a dedicated slave-enable signal, 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 $\overline{\text{SS}}$ signal is not used by the slave SPI port, the slave device must remain enabled at all times, and multiple slaves cannot be used.

6.2 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 ($\overline{\text{SS}}$).

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.

6.3 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 6−1. Table 6−2 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 6.4 show the effects of CLKSTP, CLKXP, and CLKRP.

Table 6−1. Bits Used to Enable and Configure the Clock Stop Mode

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

Table 6−2. 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.

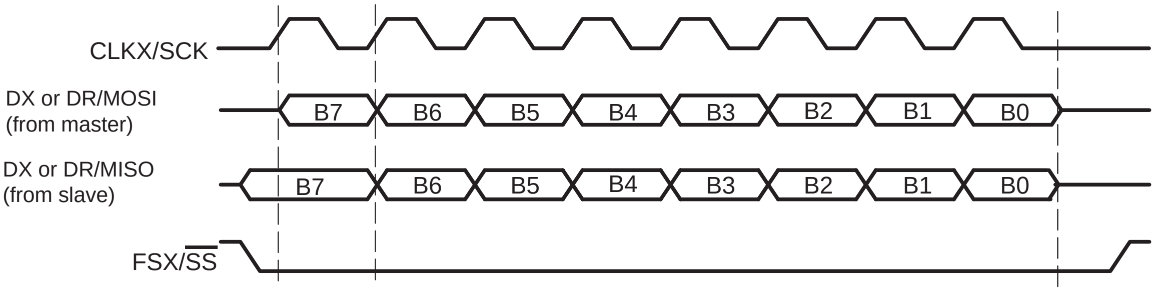
6.4 Clock Stop Mode Timing Diagrams

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 of RCR1, and the transmit packet length is selected with the XWDLEN1 bits of 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:

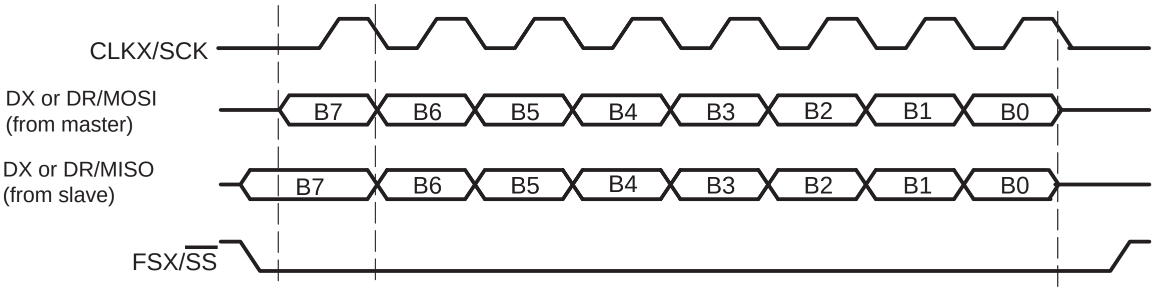
Even if multiple words are consecutively transferred, the CLKX signal is always stopped and the FSX signal returns to the inactive state after a packet transfer. When consecutive packet transfers are performed, this leads to a minimum idle time of two bit-periods between each packet transfer.

Figure 6−2. SPI Transfer With CLKSTP = 10b (no clock delay), CLKXP = 0, CLKRP = 0



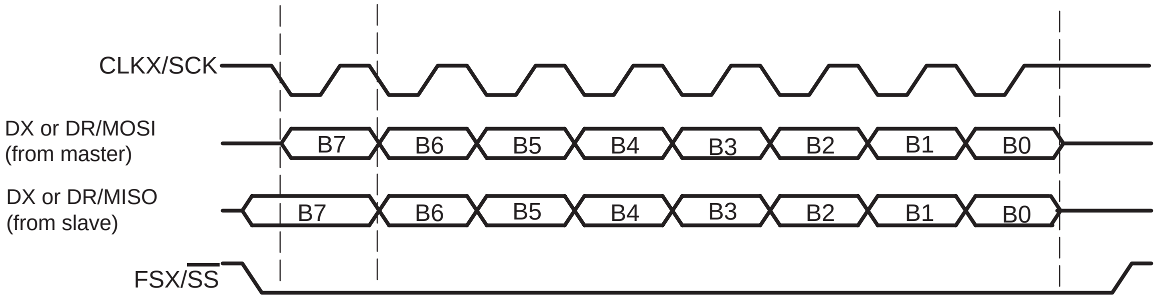
Notes: 1) If the McBSP is the SPI master (CLKXM = 1), MOSI = DX. If the McBSP is the SPI slave (CLKXM = 0), MOSI = DR.
2) If the McBSP is the SPI master (CLKXM = 1), MISO = DR. If the McBSP is the SPI slave (CLKXM = 0), MISO = DX.

Figure 6−3. 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.

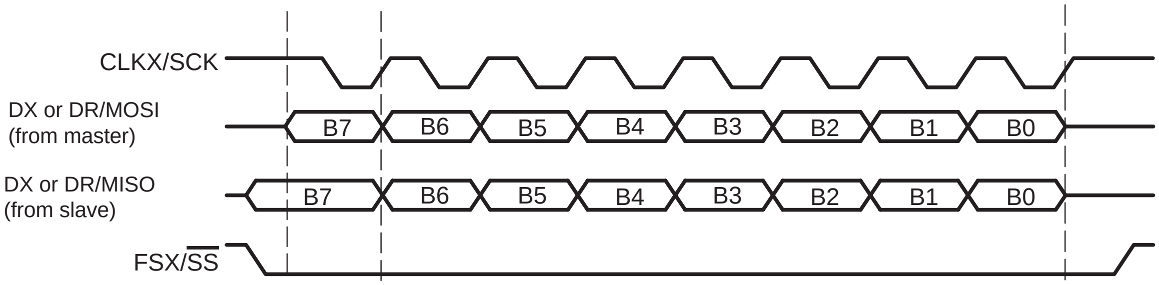
Figure 6−4. SPI Transfer With CLKSTP = 10b (no clock delay), CLKXP = 1, CLKRP = 0



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 6−5. SPI Transfer With CLKSTP = 11b (clock delay), CLKXP = 1, 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.

6.5 Procedure for Configuring a McBSP for SPI Operation

To configure the McBSP for SPI master or slave operation:

1) **Place the transmitter and receiver in reset.**

   Clear the transmitter reset bit (XRST = 0) in SPCR2, to reset the transmitter. Clear the receiver reset bit (RRST = 0) in SPCR1, to reset the receiver.

2) **Place the sample rate generator in reset.**

   Clear the sample rate generator reset bit (GRST = 0) in SPCR2, to reset the sample rate generator.

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.

4) **Enable the sample rate generator.**

   To release the sample rate generator from reset, set the sample rate generator reset bit (GRST = 1) in SPCR2.

   Make sure that during the write to SPCR2, you only modify GRST. Otherwise, you will modify the McBSP configuration you selected in the previous 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 CPU services the McBSP transmit and receive buffers, then you can immediately enable the transmitter (XRST = 1 in SPCR2) and enable the receiver (RRST = 1 in SPCR1).

   If the DMA controller services the McBSP transmit and receive buffers, then you must first configure the DMA controller (this includes enabling the channels that service the McBSP buffers). When the DMA controller is ready, make XRST = 1 and RRST = 1.

   Note: In either case, make sure you only change XRST and RRST when you write to SPCR2 and SPCR1. Otherwise, you will modify the bit settings you selected earlier in this procedure.

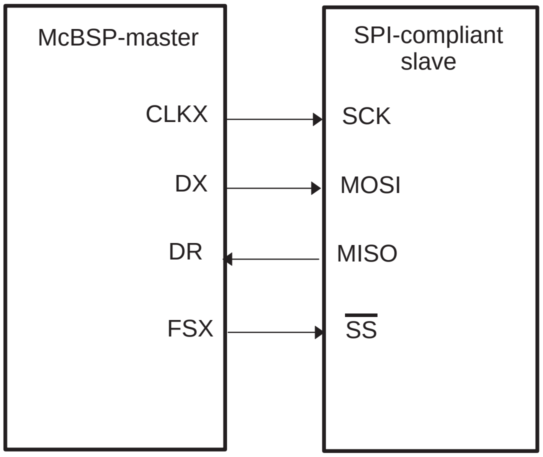
   After the transmitter and receiver are released from reset, wait two sample rate generator clock periods for the McBSP logic to stabilize.

6) **If necessary, enable the frame-sync logic of the sample rate generator.**

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

6.6 McBSP as the SPI Master

An SPI interface with the McBSP used as the master is shown in the following figure. 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.



The register bit values required to configure the McBSP as a master are listed in the following table. After the table are more details about the configuration requirements.

Table 6−3. 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 CPU clock.

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-sync 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 CPU clock. 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).

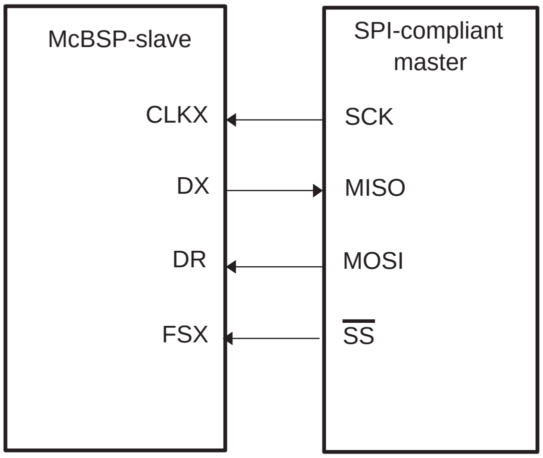
The data delay parameters of the McBSP (XDATDLY and RDATDLY) must be set to 1 for proper SPI master operation. A data delay value of 0 or 2 is undefined in the clock stop mode.

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-sync 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 should be configured active-low.

When the McBSP is configured as described for SPI-master operation, the bit fields for frame-sync pulse width (FWID) and frame-sync period (FPER) are overridden, and custom frame-sync waveforms are not allowed. The signal becomes active before the first bit of a packet transfer, and remains active until the last bit of the packet is transferred. After the packet transfer is complete, the FSX signal returns to the inactive state.

6.7 McBSP as an SPI Slave

An SPI interface with the McBSP used as a slave is shown in the following figure. When the McBSP is configured as a slave, DX is used as the MISO signal, and DR is used as the MOSI signal.



The register bit values required to configure the McBSP as a slave are listed in the following table.

Table 6−4. 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 CPU clock. (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 CPU clock 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 should be programmed to its maximum rate of half the CPU clock 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 deassert 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 set to 0 for proper SPI slave operation. A value of 1 or 2 is undefined in the clock stop mode.

Chapter 7

Receiver Configuration

This chapter describes the various methods and functions used to configure the McBSP receiver.

Topic **Page**

7.1 Configuring the McBSP Receiver

You must perform the following three steps to configure the McBSP receiver.

1) Place the McBSP/receiver in reset

2) Program the McBSP registers for the desired receiver operation

3) Take the receiver out of reset

7.2 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. Note that in the list, SRG is an abbreviation for sample rate generator.

It may be helpful to first photocopy the McBSP Register Worksheet in Chapter 13 and to fill in the blank boxes as you read the tasks.

❑ **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-sync 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

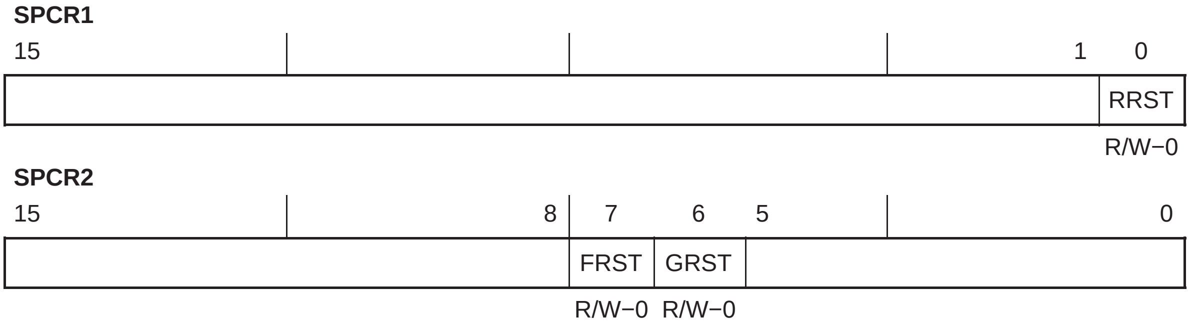
❑ **Frame-sync behavior:**
- ■ Set the receive frame-sync mode
- ■ Set the receive frame-sync polarity
- ■ Set the SRG frame-sync period and pulse width

❑ **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

7.3 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). Figure 7−1 and Table 7−1 describe the bits used for both of these steps.

Figure 7−1. Register Bits Used to Reset or Enable the McBSP Receiver



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−1. Register Bits Used to Reset or Enable the McBSP Receiver

Register	Bit	Name	Function	
SPCR1	0	RRST	Receiver Reset	
			RRST = 0	The serial port receiver is disabled and in the reset state.
			RRST = 1	The serial port receiver is enabled.
SPCR2	6	GRST	Sample Rate Generator Reset	
			GRST = 0	Sample rate generator is reset. If GRST = 0 due to a DSP reset, CLKG is driven by the CPU clock divided by 2, and FSG is driven low (inactive). If GRST = 0 due to program code, CLKG and FSG are both driven low (inactive).
			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 FRST = 1, the generator also generates the frame-sync signal FSG as programmed in the sample rate generator registers.
SPCR2	7	FRST	Frame-Sync Logic Reset	
			FRST = 0	Frame-synchronization logic is reset. The sample rate generator does not generate frame-sync signal FSG, even if GRST = 1.
			FRST = 1	If GRST = 1, frame-sync signal FSG is generated after 8 CLKG clock cycles; all frame counters are loaded with their programmed values.

7.3.1 Reset Considerations

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

1) A DSP reset ($\overline{\text{RESET}}$ signal driven low) places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed ($\overline{\text{RESET}}$ signal driven high), GRST = 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 RRST and XRST bits in the serial port control registers. The sample rate generator can be reset directly using the GRST bit in SPCR2.

Table 7−2 shows the state of McBSP pins when the serial port is reset due to a DSP reset and a direct receiver/transmitter reset.

Table 7−2. Reset State of Each McBSP Pin

Pin	Possible State(s)	State Forced By DSP Reset	State Forced By Receiver/Transmitter Reset
			Receiver Reset (RRST = 0 and 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 (XRST = 0 and GRST = 1)
DX	O/Z	High impedance	High impedance
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
CLKS	I	Input	Input

Note: In Possible State(s) column, I = Input, O = Output, Z = High impedance

7.4 Setting the Receiver Pins to Operate as McBSP Pins

The RIOEN bit, shown in Figure 7−2 and described in Table 7−3, determines whether the receiver pins are McBSP pins or general-purpose I/O pins.

Figure 7−2. Register Bit Used to Set Receiver Pins to Operate as McBSP Pins

PCR

15 13	12	11 0
	RIOEN	
	R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−3. Register Bit Used to Set Receiver Pins to Operate as McBSP Pins

Register	**Bit**	**Name**	**Function**	
PCR	12	RIOEN	Receive I/O enable	
			This bit is only applicable when the receiver is in the reset state (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 operation. The CLKS pin is a general-purpose input pin if RIOEN = XIOEN = 1 and RRST = XRST = 0.

7.5 Enabling/Disabling the Digital Loopback Mode

The DLB bit determines whether the digital loopback mode is on. DLB is shown in Figure 7−3 and described in Table 7−4.

Figure 7−3. Register Bit Used to Enable/Disable the Digital Loopback Mode

SPCR1

15	14 – 0
DLB	
R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−4. Register Bit Used to Enable/Disable the Digital Loopback Mode

Register	Bit	Name	Function	
SPCR1	15	DLB	Digital Loopback Mode	
			DLB = 0	Digital loopback mode is disabled.
			DLB = 1	Digital loopback mode is enabled.

7.5.1 About the 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 7−5. This mode allows testing of serial port code with a single DSP device; the McBSP receives the data it transmits.

Table 7−5. Receive Signals Connected to Transmit Signals in Digital Loopback Mode

This Receive Signal …	**Is Fed Internally By This Transmit Signal** …
DR (receive data)	DX (transmit data)
FSR (receive frame synchronization)	FSX (transmit frame synchronization)
CLKR (receive clock)	CLKX (transmit clock)

7.6 Enabling/Disabling the Clock Stop Mode

The CLKSTP bits determine whether the clock stop mode is on. CLKSTP is shown in Figure 7−4 and described in Table 7−6.

Figure 7−4. Register Bits Used to Enable/Disable the Clock Stop Mode

SPCR1

15 13	12 11	10 0
	CLKSTP	
	R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−6. Register Bits Used to Enable/Disable the Clock Stop Mode

Register	Bit	Name	Function	
SPCR1	12−11	CLKSTP	Clock Stop Mode	
			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

7.6.1 About the Clock Stop Mode

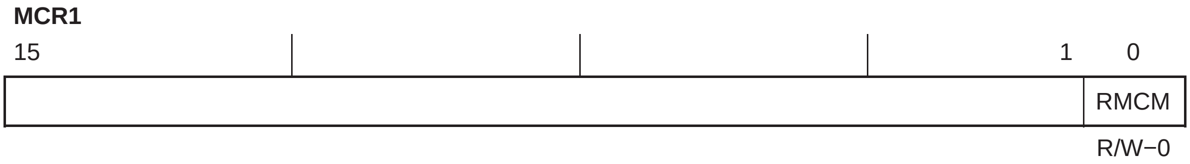
The clock stop mode supports the SPI master-slave protocol. If you will not be using the SPI protocol, you can clear CLKSTP to disable the clock stop mode.

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). 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.

7.7 Enabling/Disabling the Receive Multichannel Selection Mode

The RMCM bit determines whether the receive multichannel selection mode is on. RMCM is shown in Figure 7−5 and described in Table 7−7.

Figure 7−5. Register Bit Used to Enable/Disable the Receive Multichannel Selection Mode



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−7. Register Bit Used to Enable/Disable the Receive Multichannel Selection Mode

Register	Bit	Name	Function	
MCR1	0	RMCM	Receive Multichannel Selection Mode	
			RMCM = 0	The mode is disabled. All 128 channels are enabled.
			RMCM = 1	The mode is enabled. 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.

7.8 Choosing 1 or 2 Phases for the Receive Frame

The RPHASE bit (see Figure 7−6 and Table 7−8) determines whether the receive data frame has one or two phases.

Figure 7−6. Register Bit Used to Choose 1 or 2 Phases for the Receive Frame

RCR2

15	14	0
RPHASE		
R/W−0		

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−8. Register Bit Used to Choose 1 or 2 Phases for the Receive Frame

Register	**Bit**	**Name**	**Function**

RCR2	15	RPHASE	Receive phase number
			Specifies whether the receive frame has 1 or 2 phases.
			RPHASE = 0 Single-phase frame
			RPHASE = 1 Dual-phase frame

7.9 Setting the Receive Word Length(s)

The RWDLEN1 and RWDLEN2 bit fields (see Figure 7−7 and Table 7−9) determine how many bits are in each serial word in phase 1 and in phase 2, respectively, of the receive data frame.

Figure 7−7. Register Bits Used to Set the Receive Word Length(s)

RCR1

15 8	7 5	4 0
	RWDLEN1	
	R/W−000	

RCR2

15 8	7 5	4 0
	RWDLEN2	
	R/W−000	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−9. Register Bits Used to Set the Receive Word Length(s)

Register	Bit	Name	Function

RCR1	7−5	RWDLEN1	Receive word length 1
			Specifies the length of every serial word in phase 1 of the receive frame.
			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.
			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

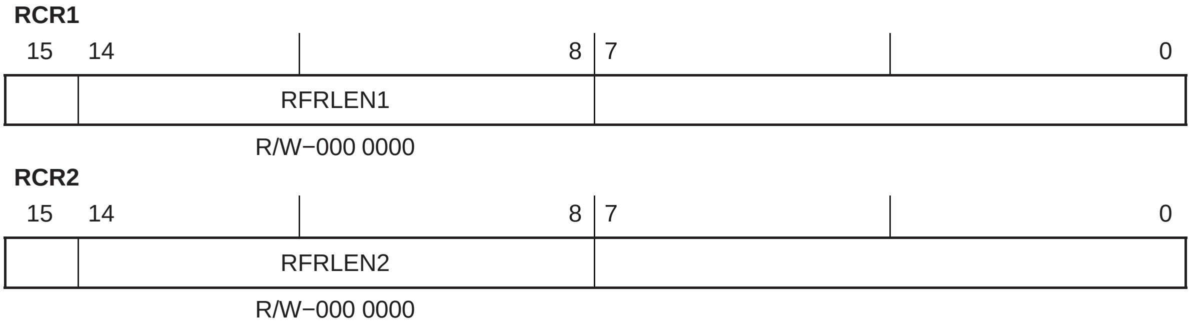
7.9.1 About the Word Length Bits

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 and RWDLEN2 is ignored. 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.

7.10 Setting the Receive Frame Length

The RFRLEN1 and RFRLEN2 bit fields (see Figure 7−8 and Table 7−10) determine how many serial words are in phase 1 and in phase 2, respectively, of the receive data frame.

Figure 7−8. Register Bits Used to Set the Receive Frame Length



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−10. Register Bits Used to Set the Receive Frame Length

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

7.10.1 About the 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 the 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 7−11 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.

Note: 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.

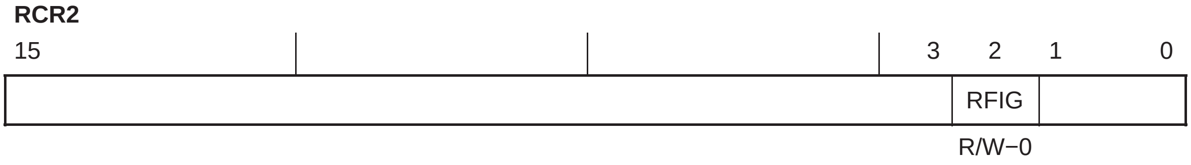
Table 7−11. How to Calculate the Length of the Receive Frame

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

7.11 Enabling/Disabling the Receive Frame-Sync Ignore Function

The RFIG bit (see Figure 7−9 and Table 7−12) controls the receive frame-sync ignore function.

Figure 7−9. Register Bit Used to Enable/Disable the Receive Frame-Sync Ignore Function



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−12. Register Bit Used to Enable/Disable the Receive Frame-Sync Ignore Function

Register	Bit	Name	Function	
RCR2	2	RFIG	Receive Frame-Sync Ignore	
			RFIG = 0	An unexpected receive frame-sync pulse causes the McBSP to restart the frame transfer.
			RFIG = 1	The McBSP ignores unexpected receive frame-sync pulses.

7.11.1 About Unexpected Frame-Sync Pulses and the Frame-Sync Ignore Function

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

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

7.12 Setting the Receive Companding Mode

The RCOMPAND bits (see Figure 7−10 and Table 7−13) determine whether companding or another data transfer option is chosen for McBSP reception.

Figure 7−10. Register Bits Used to Set the Receive Companding Mode

RCR2

15 5	4 3	2 0
	RCOMPAND	
	R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−13. Register Bits Used to Set the Receive Companding Mode

Register	Bit	Name	Function
RCR2	4−3	RCOMPAND	Receive companding mode 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, scroll down to Option to Receive LSB First) RCOMPAND = 10 µ-law companding, 8-bit data, MSB received first RCOMPAND = 11 A-law companding, 8-bit data, MSB received first

7.13 Setting the Receive Data Delay

The RDATDLY bits (see Figure 7−11 and Table 7−14) determine the length of the data delay for the receive frame.

Figure 7−11. Register Bits Used to Set the Receive Data Delay

RCR2

15 2	1 0
	RDATDLY
	R/W−00

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−14. Register Bits Used to Set the Receive Data Delay

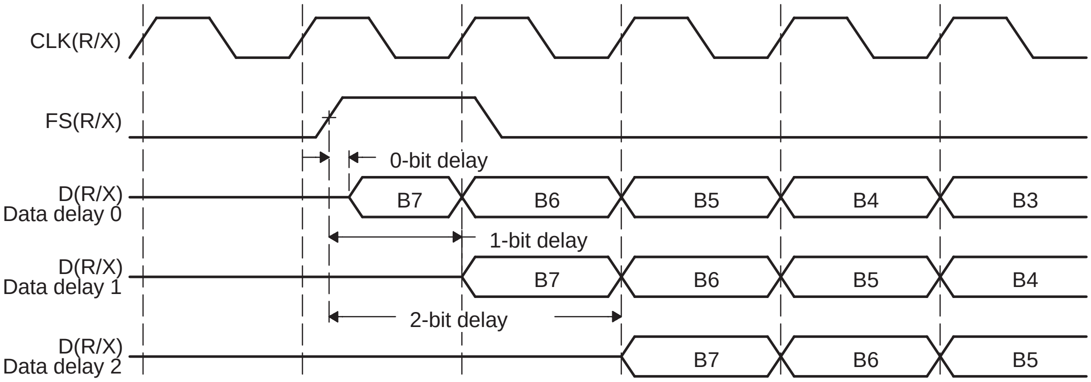
Register	Bit	Name	Function	
RCR2	1−0	RDATDLY	Receive data delay	
			RDATDLY = 00	0-bit data delay
			RDATDLY = 01	1-bit data delay
			RDATDLY = 10	2-bit data delay
			RDATDLY = 11	Reserved

7.13.1 About the 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 7−14 and shown in Figure 7−12. 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-sync pulse.

Figure 7−12. Range of Programmable Data Delay



7.13.2 0-Bit Data Delay

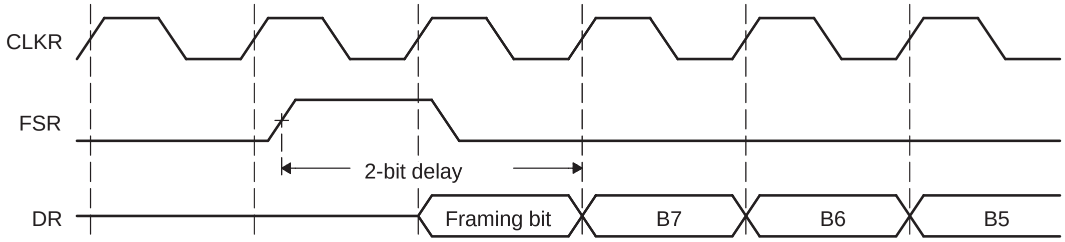
Normally, a frame-sync pulse is detected or sampled with respect to an edge of internal serial clock CLK(R/X). Thus, on the following cycle or later (depending on the data delay value), data may be received or transmitted. However, in the case of 0-bit data delay, the data must be ready for reception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

This problem is solved for reception because receive data is sampled on the first falling edge of CLKR where an active-high internal FSR is detected. 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-sync signal (FSX) going active high and immediately starts driving the first bit to be transmitted on the DX pin.

7.13.3 2-Bit Data Delay

A data delay of two bit periods allows the serial port to interface to different types of T1 framing 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 7−13. In this figure, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on.

Figure 7−13. 2-Bit Data Delay Used to Skip a Framing Bit



7.14 Setting the Receive Sign-Extension and Justification Mode

The RJUST bits (see Figure 7−14 and Table 7−15) determine whether data received by the McBSP is sign extended and how it is justified.

Figure 7−14. Register Bits Used to Set the Receive Sign-Extension and Justification Mode

SPCR1

15	14–13	12–0
	RJUST	
	R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−15. Register Bits Used to Set the Receive Sign-Extension and Justification Mode

Register	Bit	Name	Function	
SPCR1	14−13	RJUST	Receive Sign-Extension and Justification Mode	
			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

7.14.1 About the 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 how unused bits in DRR[1,2] are filled—with zeros or with sign bits.

Table 7−16 and Table 7−17 show the effects of various RJUST values. The first table shows the effect on an example 12-bit receive-data value 0xABC. The second table shows the effect on an example 20-bit receive-data value 0xABCDE.

Table 7−16. Example: 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 7−17. Example: 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

7.15 Setting the Receive Interrupt Mode

The RINTM bits (see Figure 7−15 and Table 7−18) determine which event generates a receive interrupt request to the CPU.

Figure 7−15. Register Bits Used to Set the Receive Interrupt Mode

SPCR1

15	6	5	4	3	0
		RINTM			
		R/W−00			

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−18. Register Bits Used to Set the Receive Interrupt Mode

Register	Bit	Name	Function	
SPCR1	5–4	RINTM	Receive Interrupt Mode	
			RINTM = 00	RINT generated when RRDY changes from 0 to 1
			RINTM = 01	RINT generated by an end-of-block or end-of-frame condition in the receive multichannel selection mode
			RINTM = 10	RINT generated by a new receive frame-sync pulse
			RINTM = 11	RINT generated when RSYNCERR is set

7.15.1 About the Receive Interrupt and the Associated Modes

The receive interrupt (RINT) signals the CPU of changes to the serial port status. Four options exist for configuring this interrupt. The options 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.

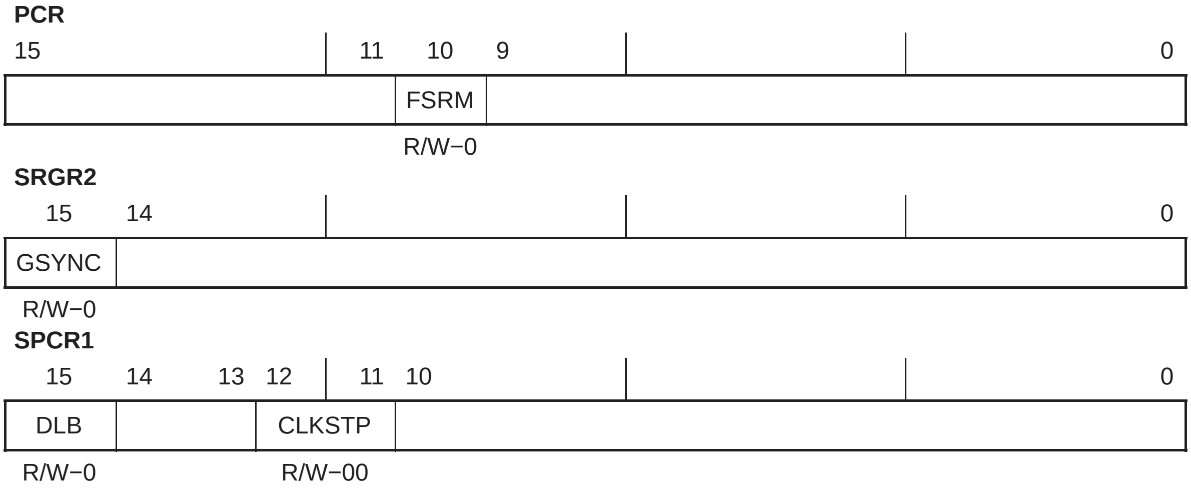
❑ RINTM = 10b. Interrupt on detection of receive frame-sync pulses. This generates an interrupt even when the receiver is in its reset state. This is done by synchronizing the incoming frame-sync pulse to the CPU clock and sending it to the CPU via RINT.

❑ RINTM = 11b. Interrupt on frame-synchronization error. Note that regardless of the value of RINTM, RSYNCERR can be read to detect this condition.

7.16 Setting the Receive Frame-Sync Mode

The bits shown in Figure 7−16 and described in Table 7−19 determine the source for receive frame synchronization and the function of the FSR pin.

Figure 7−16. Register Bits Used to Set the Receive Frame Sync Mode



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−19. Register Bits Used to Set the Receive Frame Sync Mode

Register	Bit	Name	Function	
PCR	10	FSRM	Receive Frame-Synchronization Mode	
			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 7−19. Register Bits Used to Set the Receive Frame Sync Mode(Continued)

Register	**Bit**	**Name**	**Function**	
SRGR2	15	GSYNC	Sample Rate Generator Clock Synchronization Mode	
			If the sample rate generator creates a frame-sync 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.	
			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-sync period defined in FPER is ignored.
SPCR1	15	DLB	Digital Loopback Mode	
			DLB = 0	Digital loopback mode is disabled.
			DLB = 1	Digital loopback mode is enabled. The receive signals, including the receive frame-sync signal, are connected internally through multiplexers to the corresponding transmit signals.
SPCR1	12−11	CLKSTP	Clock Stop Mode	
			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.

7.16.1 About the Receive Frame-Sync Modes

Table 7−20 shows how you may 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.

Note that in the digital loop back mode (DLB = 1), the transmit frame-sync signal is used as the receive frame-sync 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 7−20. Select Sources to Provide the Receive Frame-Synchronization Signal and . the Effect on the FSR Pin

DLB	FSRM	GSYNC	Source of Receive Frame Synchronization	FSR Pin Status
0	0	0 or 1	An external frame-sync 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-sync 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-sync signal (FSG).	Input. The external frame-sync 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.

7.17 Setting the Receive Frame-Sync Polarity

The FSRP bit (see Figure 7−17 and Table 7−21) determines whether frame-synchronization (frame-sync) pulses are active high or active low on the FSR pin.

Figure 7−17. Register Bit Used to Set Receive Frame-Sync Polarity

PCR

15	3	2	1	0
		FSRP		
		R/W−0		

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−21. Register Bit Used to Set Receive Frame-Sync Polarity

Register	Bit	Name	Function	
PCR	2	FSRP	Receive Frame-Synchronization Polarity	
			FSRP = 0	Frame-synchronization pulse FSR is active high.
			FSRP = 1	Frame-synchronization pulse FSR is active low.

7.17.1 About Frame Sync Pulses, Clock Signals, and Their Polarities

Receive frame-sync pulses can be either generated internally by the sample rate generator or driven by an external source. The source of frame sync is selected by programming the mode bit, FSRM, in PCR. FSR is also affected by the GSYNC bit in SRGR2. Similarly, receive clocks can be selected to be inputs or outputs by programming the mode bit, CLKRM, in the PCR.

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame-sync pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the DR pin is also sampled on the falling edge of internal CLKR. Note that these internal clock signals are either derived from 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 (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the DX pin is output on the rising edge of internal CLKX.

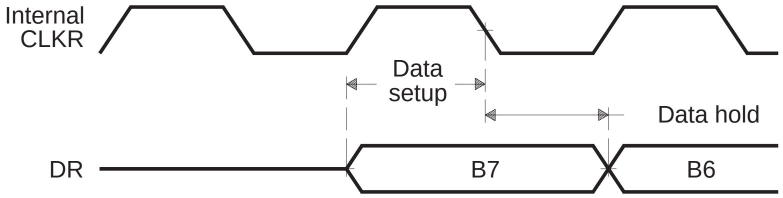
FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame-sync signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame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

Note that 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 7−18 shows how data clocked by an external serial device using a rising edge, may be sampled by the McBSP receiver on the falling edge of the same clock.

Figure 7−18. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



7.18 Setting the SRG Frame-Sync Period and Pulse Width

Figure 7−19. Register Bits Used to Set the SRG Frame-Sync Period and Pulse Width

SRGR2

15 12	11 0
	FPER
	R/W−0000 0000 0000

SRGR1

15 8	7 0
FWID	
R/W−0000 0000	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−22. Register Bits Used to Set the SRG Frame-Sync Period and Pulse Width

Register	Bit	Name	Function
SRGR2	11−0	FPER	Sample Rate Generator Frame-Sync Period For the frame-sync signal FSG, (FPER + 1) determines the period from the start of a frame-sync pulse to the start of the next frame-sync pulse. Range for (FPER + 1): 1 to 4096 CLKG cycles.
SRGR1	15−8	FWID	Sample Rate Generator Frame-Sync Pulse Width This field plus 1 determines the width of each frame-sync pulse on FSG. Range for (FWID + 1): 1 to 256 CLKG cycles.

7.18.1 About the Frame-Sync Period and the Frame-Sync Pulse Width

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

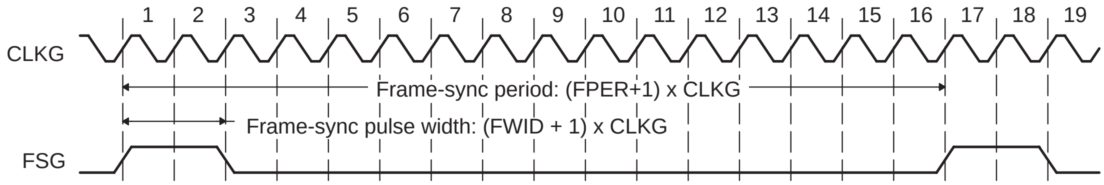
On FSG, the period from the start of a frame-sync pulse to the start of the next pulse is (FPER + 1) CLKG cycles. The 12 bits of FPER allow a frame-sync period of 1 to 4096 CLKG cycles, which allows up to 4096 data bits per frame. When GSYNC = 1, FPER is a don’t care value.

Each pulse on FSG has a width of (FWID + 1) CLKG cycles. The eight bits of FWID allow a pulse width of 1 to 256 CLKG cycles. It is recommended that FWID be programmed 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 7−20 shows a frame-sync period of 16 CLKG periods (FPER = 15 or 00001111b) and a frame-sync pulse with an active width of 2 CLKG periods (FWID = 1).

Figure 7−20. 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-sync period: (FPER+1) x CLKG
Frame-sync pulse width: (FWID + 1) x CLKG
FSG

When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when FRST = 1 and FSGM = 1, a frame-sync 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.

7.19 Setting the Receive Clock Mode

The bits shown in Figure 7−21 and described in Table 7−23 determine the source for receive clock and the function of the CLKR pin.

Figure 7−21. Register Bits Used to Set the Receive Clock Mode

PCR

15–9	8	7–0
	CLKRM	
	R/W−0	

SPCR1

15	14–13	12–11	10–0
DLB		CLKSTP	
R/W−0		R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−23. Register Bits Used to Set the Receive Clock Mode

Register	Bit	Name	Function	
PCR	8	CLKRM	Receive Clock Mode	
			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 according to the CLKXM bit of 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 according to the CLKXM bit of PCR.
SPCR1	15	DLB	Digital Loopback Mode	
			DLB = 0	Digital loopback mode is disabled.
			DLB = 1	Digital loopback mode is enabled. The receive signals, including the receive frame-sync signal, are connected internally through multiplexers to the corresponding transmit signals.

Table 7−23. Register Bits Used to Set the Receive Clock Mode (Continued)

Register	Bit	Name	Function	
SPCR1	12−11	CLKSTP	Clock Stop Mode	
			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.

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

Table 7−24 shows how you may select various sources to provide the receive clock signal and the effect on the CLKR pin. The polarity of the signal on the CLKR pin is determined by the CLKRP bit.

Note that in the digital loop back mode (DLB = 1), the transmit clock signal is also used as the receive clock 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 7−24. Select Sources to Provide the Receive Clock Signal and the Effect on the . CLKR Pin

DLB in SPCR1	CLKRM in PCR	Source of Receive Clock	CLKR Pin Status
0	0	The CLKR pin is an 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.

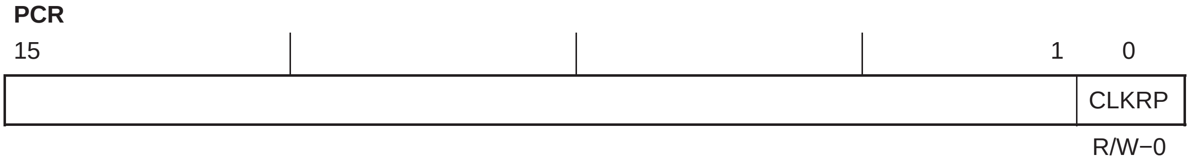
Table 7−24. Select Sources to Provide the Receive Clock Signal and the Effect on the . CLKR Pin (Continued)

DLB in SPCR1	CLKRM in PCR	Source of Receive Clock	CLKR Pin Status
1	0	Internal CLKX drives internal CLKR. To configure CLKX, see *Set the Transmit Clock Mode.*	High impedance
1	1	Internal CLKX drives internal CLKR. To configure CLKX, see *Set the Transmit Clock Mode*.	Output. Internal CLKR (same as internal CLKX) is inverted as determined by CLKRP before being driven out on the CLKR pin.

7.20 Setting the Receive Clock Polarity

The bits shown in Figure 7−22 and described in Table 7−25 determine the source for setting the receive clock polarity.

Figure 7−22. Register Bit Used to Set Receive Clock Polarity



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−25. Register Bit Used to Set Receive Clock Polarity

Register	Bit	Name	Function	
PCR	0	CLKRP	Receive Clock Polarity	
			CLKRP = 0	When CLKR is configured as a input, the external CLKR is not inverted before being used internally. When CLKR is configured as an output, the internal CLKR is not inverted before being driven on the CLKR pin. The receive data is sampled on the falling edge of the external CLKR signal.
			CLKRP = 1	When CLKR is configured as a input, the external CLKR is inverted before being used internally. When CLKR is configured as an output, the internal CLKR is inverted before being driven on the CLKR pin. The receive data is sampled on the rising edge of the external CLKR signal.

7.20.1 About Frame Sync Pulses, Clock Signals, and Their Polarities

Receive frame-sync pulses can be either generated internally by the sample rate generator or driven by an external source. The source of frame sync is selected by programming the mode bit, FSRM, in PCR. FSR is also affected by the GSYNC bit in SRGR2. Similarly, receive clocks can be selected to be inputs or outputs by programming the mode bit, CLKRM, in the PCR.

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame-sync pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the DR pin is also sampled on the falling edge of internal CLKR. Note that these internal clock signals are either derived from 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 (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the DX pin is output on the rising edge of internal CLKX.

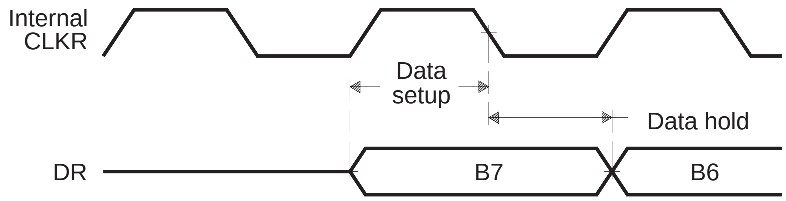
FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame-sync signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame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

Note that 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 7−23 shows how data clocked by an external serial device using a rising edge, may be sampled by the McBSP receiver on the falling edge of the same clock.

Figure 7−23. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



7.21 Setting the SRG Clock Divide-Down Value

The bits shown in Figure 7−24 and described in Table 7−26 determine the source for setting the SRG clock divide-down value.

Figure 7−24. Register Bits Used to Set the Sample Rate Generator (SRG) Clock Divide-Down Value

SRGR1

15 8	7 0
	CLKGDV
	R/W−0000 0001

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−26. Register Bits Used to Set the Sample Rate Generator (SRG) Clock Divide-Down Value

Register	Bit	Name	Function
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).

7.21.1 About the 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 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.

Note:

The maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. For more information on programming the frequency of the slow peripheral clock, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* (SPRS206) data manual or the *TMS320VC5502 Fixed-Point Digital Signal Processor* (SPRS166) data manual. Other timing limitations may also apply. Refer to the device-specific data manual for detailed information on the McBSP timing requirements.

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 (CLKGDV).

7.22 Setting the SRG Clock Synchronization Mode

The bits shown in Figure 7−25 and described in Table 7−27 determine the source for setting the SRG clock synchronization mode.

Figure 7−25. Register Bit Used to Set the SRG Clock Synchronization Mode

SRGR2

15	14	0
GSYNC		
R/W−0		

Legend: R = Read; W = Write; -*n* = Value after reset

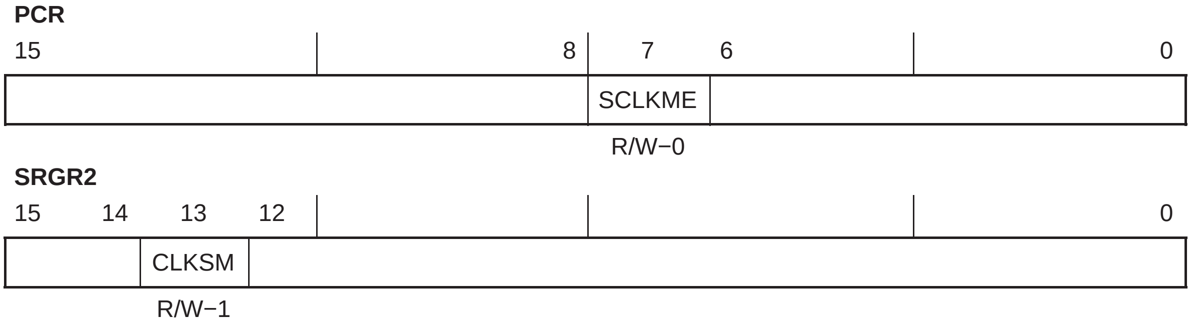
Table 7−27. Register Bit Used to Set the SRG Clock Synchronization Mode

Register	Bit	Name	Function	
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, 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-sync period defined in FPER is ignored.

7.23 Setting the SRG Clock Mode (Choose an Input Clock)

The bits shown in Figure 7−26 and described in Table 7−28 determine the source for setting the SRG clock synchronization mode.

Figure 7−26. Register Bits Used to Set the SRG Clock Mode (Choose an Input Clock)



Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−28. Register Bits Used to Set the SRG Clock Mode (Choose an Input Clock)

Register	Bit	Name	Function	
PCR SRGR2	7 13	SCLKME CLKSM	Sample Rate Generator Clock Mode	

			SCLKME = 0
			CLKSM = 0
				Sample rate generator clock derived from CLKS pin

			SCLKME = 0
			CLKSM = 1
				Sample rate generator clock derived from CPU clock (This is the condition forced by a DSP reset.)

			SCLKME = 1
			CLKSM = 0
				Sample rate generator clock derived from CLKR pin

			SCLKME = 1
			CLKSM = 1
				Sample rate generator clock derived from CLKX pin

7.23.1 About the 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. The preceding table shows the four possible sources of the input clock.

7.24 Setting the SRG Input Clock Polarity

The bits shown in Figure 7−25 and described in Table 7−27 determine the source for setting the SRG clock synchronization mode.

Figure 7−27. Register Bits Used to Set the SRG Input Clock Polarity

SRGR2

15	14	13–0
	CLKSP	
	R/W−0	

PCR

15–2	1	0
	CLKXP	CLKRP
	R/W−0	R/W−0

Legend: R = Read; W = Write; -*n* = Value after reset

Table 7−29. Register Bits Used to Set the SRG Input Clock Polarity

Register	Bit	Name	Function	
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.
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.
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.

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

The sample rate generator can produce a clock signal (CLKG) and a frame-sync 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 CPU clock 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.

Chapter 8

Transmitter Configuration

This chapter provides details on how to configure an McBSP transmitter.

Topic **Page**

8.1 Configuring the Transmitter

To configure the McBSP transmitter, perform the following procedure:

1) Place the McBSP/transmitter in reset
2) Program the McBSP registers for the desired transmitter operation
3) Take the transmitter out of reset

8.2 Programming McBSP Registers for Desired Transmitter Operations

The following is a list of important tasks to be performed when you are configuring the McBSP transmitter. Each task corresponds to one or more McBSP register bit fields. Note that in the list, SRG is an abbreviation for sample rate generator.

It may be helpful to print the McBSP Register Worksheet first and to fill it in as you read the tasks.

- ❑ **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-sync ignore function
  - ■ Set the transmit companding mode
  - ■ Set the transmit data delay
  - ■ Set the transmit DXENA mode
  - ■ Set the transmit interrupt mode
- ❑ **Frame-sync behavior:**
  - ■ Set the transmit frame-sync mode
  - ■ Set the transmit frame-sync polarity
  - ■ Set the SRG frame-sync 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

8.3 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 (to take it out of reset). Figure 8−1 and Table 8−1 describe the bits used for both of these steps.

Figure 8−1. Register Bits Used to Place Transmitter in Reset

SPCR2

15	8	7	6	5	1	0
		FRST	GRST			XRST
		R/W−0	R/W−0			R/W−0

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−1. Register Bits Used to Place Transmitter in Reset

Register	Bit	Name	Function	
SPCR2	0	XRST	Transmitter Reset	
			XRST = 0	The serial port transmitter is disabled and in the reset state.
			XRST = 1	The serial port transmitter is enabled.
SPCR2	6	GRST	Sample Rate Generator Reset	
			GRST = 0	Sample rate generator is reset. If GRST = 0 due to a DSP reset, CLKG is driven by the CPU clock divided by 2, and FSG is driven low (inactive). If GRST = 0 due to program code, CLKG and FSG are both driven low (inactive).
			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 FRST = 1, the generator also generates the frame-sync signal FSG as programmed in the sample rate generator registers.
SPCR2	7	FRST	Frame-Sync Logic Reset	
			FRST = 0	Frame-synchronization logic is reset. The sample rate generator does not generate frame-sync signal FSG, even if GRST = 1.
			FRST = 1	If GRST = 1, frame-sync signal FSG is generated after 8 CLKG clock cycles; all frame counters are loaded with their programmed values.

8.3.1 Reset Considerations

The serial port can be reset in two ways:

1) A DSP reset ($\overline{\text{RESET}}$ signal driven low) places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed ($\overline{\text{RESET}}$ signal driven high), GRST = FRST = RRST = XRST = 0, keeps the entire serial port in the reset state.

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

Table 8−2 shows the state of McBSP pins when the serial port is reset due to a DSP reset and a direct receiver/transmitter reset.

Table 8−2. Reset State of Each McBSP Pin

Pin	Possible State(s)	State Forced By DSP Reset	State Forced By Receiver/Transmitter Reset
			Receiver Reset (RRST = 0 and 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 (XRST = 0 and GRST = 1)
DX	O/Z	High impedance	High impedance
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
CLKS	I	Input	Input

8.4 Setting the Transmitter Pins to Operate as McBSP Pins

The XIOEN bit shown in Figure 8−2 and Table 8−3 is used to set the transmitter pins to operate as McBSP pins.

Figure 8−2. Register Bit Used to Set Transmitter Pins to Operate as McBSP Pins

PST

15 14	13	12 0
	XIOEN	
	R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

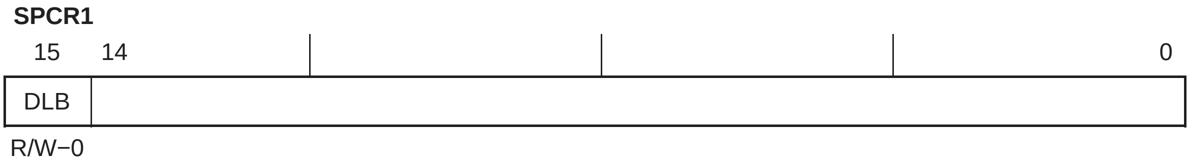
Table 8−3. Register Bit Used to Set Transmitter Pins to Operate as McBSP Pins

Register	Bit	Name	Function	
PCR	13	XIOEN	Transmit I/O enable	
			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 operation. The CLKS pin is a general-purpose input pin if RIOEN = XIOEN = 1 and RRST = XRST = 0.

8.5 Enabling/Disabling the Digital Loopback Mode

The DLB bit determines whether the digital loopback mode is on. DLB is shown in Figure 8−3 and described in Table 8−4.

Figure 8−3. Register Bit Used to Enable/Disable the Digital Loopback Mode



Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−4. Register Bit Used to Enable/Disable the Digital Loopback Mode

Register	Bit	Name	Function	
SPCR1	15	DLB	Digital Loopback Mode	
			DLB = 0	Digital loopback mode is disabled.
			DLB = 1	Digital loopback mode is enabled.

8.5.1 About the 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 8−5. This mode allows testing of serial port code with a single DSP device; the McBSP receives the data it transmits.

Table 8−5. Receive Signals Connected to Transmit Signals in Digital Loopback Mode

This Receive Signal …	Is Fed Internally By This Transmit Signal …
DR (receive data)	DX (transmit data)
FSR (receive frame synchronization)	FSX (transmit frame synchronization)
CLKR (receive clock)	CLKX (transmit clock)

8.6 Enabling/Disabling the Clock Stop Mode

The CLKSTP bits determine whether the clock stop mode is on. CLKSTP is shown in Figure 8−4 and described in Table 8−6.

Figure 8−4. Register Bits Used to Enable/Disable the Clock Stop Mode

SPCR1

15 13	12 11	10 0
	CLKSTP	
	R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−6. Register Bits Used to Enable/Disable the Clock Stop Mode

Register	Bit	Name	Function	
SPCR1	12−11	CLKSTP	Clock Stop Mode	
			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

8.6.1 About the Clock Stop Mode

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

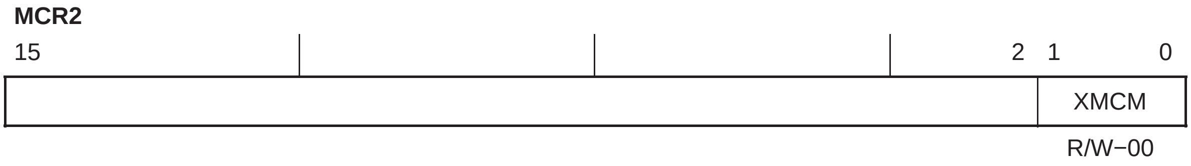
In the clock stop mode, the clock stops at the end of each data transfer. At the beginning of each data transfer, the clock starts immediately (CLKSTP = 10b) or after a half-cycle delay (CLKSTP = 11b). The CLKXP bit determines whether the starting edge of the clock on the CLKX pin is rising or falling.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

Table 6−2 on page 6-5 summarizes the impact of CLKSTP, CLKXP, and CLKRP on serial port operation. Note that in the clock stop mode, the receive clock is tied internally to the transmit clock, and the receive frame-sync signal is tied internally to the transmit frame-sync signal.

8.7 Enabling/Disabling Transmit Multichannel Selection

The XMCM bits shown in Figure 8−5 and Table 8−7, are used to enable and disable transmit multichannel selection.

Figure 8−5. Register Bits Used to Enable/Disable Transmit Multichannel Selection



Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−7. Register Bits Used to Enable/Disable Transmit Multichannel Selection

Register	Bit	Name	Function	
MCR2	1−0	XMCM	Transmit Multichannel Selection	
			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 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 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 RCERs and XCERs.

8.8 Choosing 1 or 2 Phases for the Transmit Frame

The XPHASE bit, shown in Figure 8−6 and Table 8−8, are used to choose 1 or 2 phases for the transmit frame.

Figure 8−6. Register Bit Used to Choose 1 or 2 Phases for the Transmit Frame

XCR2

15	14 … 0
XPHASE	
R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−8. Register Bit Used to Choose 1 or 2 Phases for the Transmit Frame

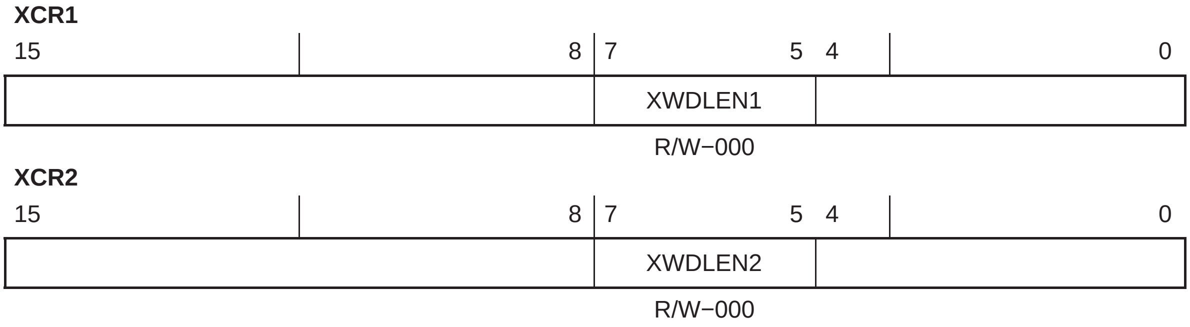
Register	Bit	Name	Function

XCR2	15	XPHASE	Transmit phase number
			Specifies whether the transmit frame has 1 or 2 phases.
			XPHASE = 0 Single-phase frame
			XPHASE = 1 Dual-phase frame

8.9 Setting the Transmit Word Length(s)

The register bits XWDLEN1, shown in Figure 8−7 and Table 8−9, are used to set the transmit word length(s).

Figure 8−7. Register Bits Used to Set the Transmit Word Length(s)



Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−9. Register Bits Used to Set the Transmit Word Length(s)

Register	Bit	Name	Function
XCR1	7−5	XWDLEN1	Transmit Word Length of Frame Phase 1
			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
			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

8.9.1 About the Word Length Bits

Each frame can have one or two phases, depending on the value that you load into the XPHASE 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.

8.10 Setting the Transmit Frame Length

The XFRLEN1 register bits, shown in Figure 8−8 and Table 8−10, are used to set the transmit frame length.

Figure 8−8. Register Bits Used to Set the Transmit Frame Length

XCR1

15	14 … 8	7 … 0
	XFRLEN1	
	R/W−000 0000	

XCR2

15	14 … 8	7 … 0
	XFRLEN2	
	R/W−000 0000	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−10. Register Bits Used to Set the Transmit Frame Length

Register	Bit	Name	Function
XCR1	14−8	XFRLEN1	Transmit frame length 1 (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 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

8.10.1 About the 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 value that you load 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 8−11 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.

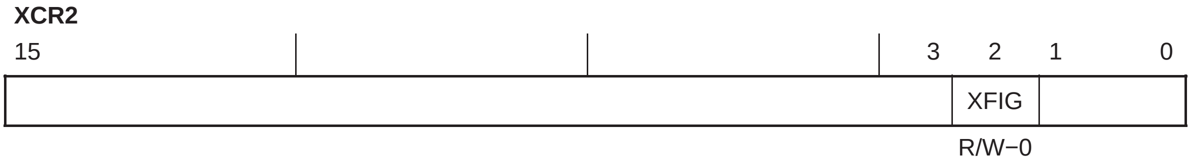
Note: Program the XFRLEN fields with [*w minus 1*], where *w* represents the number of words per phase. For the example, if you want a phase length of 128 words in phase 1, load 127 into XFRLEN1.

Table 8−11. 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

8.11 Enabling/Disabling the Transmit Frame-Sync Ignore Function

Figure 8−9. Register Bit Used to Enable/Disable the Transmit Frame-Sync Ignore Function



Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−12. Register Bit Used to Enable/Disable the Transmit Frame-Sync Ignore Function

Register	Bit	Name	Function	
XCR2	2	XFIG	Transmit Frame-Sync Ignore	
			XFIG = 0	An unexpected transmit frame-sync pulse causes the McBSP to restart the frame transfer.
			XFIG = 1	The McBSP ignores unexpected transmit frame-sync pulses.

8.11.1 About Unexpected Frame-Sync Pulses and the Frame-Sync Ignore Function

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

When XFIG = 1, normal transmission continues with unexpected frame-sync signals ignored.

When XFIG = 0 and an unexpected frame-sync pulse occurs, the serial port:

1) Aborts the present transmission

2) Sets XSYNCERR to 1 in SPCR2

3) Re-initiates transmission of the current word that was aborted

8.12 Setting the Transmit Companding Mode

The XCOMPAND register bits, shown in Figure 8−10 and Table 8−13, are used to set t he transmit companding mode.

Figure 8−10. Register Bits Used to Set the Transmit Companding Mode

XCR2

15 5	4 3	2 0
	XCOMPAND	
	R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−13. Register Bits Used to Set the Transmit Companding Mode

Register	Bit	Name	Function	
XCR2	4−3	XCOMPAND	Transmit Companding Mode	
			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, scroll down to Option to Transmit LSB First)
			XCOMPAND = 10b	µ-law companding, 8-bit data, MSB transmitted first
			XCOMPAND = 11b	A-law companding, 8-bit data, MSB transmitted first

8.13 Setting the Transmit Data Delay

Figure 8−11.Register Bits Used to Set the Transmit Data Delay

XCR2

15	2	1	0
		XDATDLY	
		R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−14. Register Bits Used to Set the Transmit Data Delay

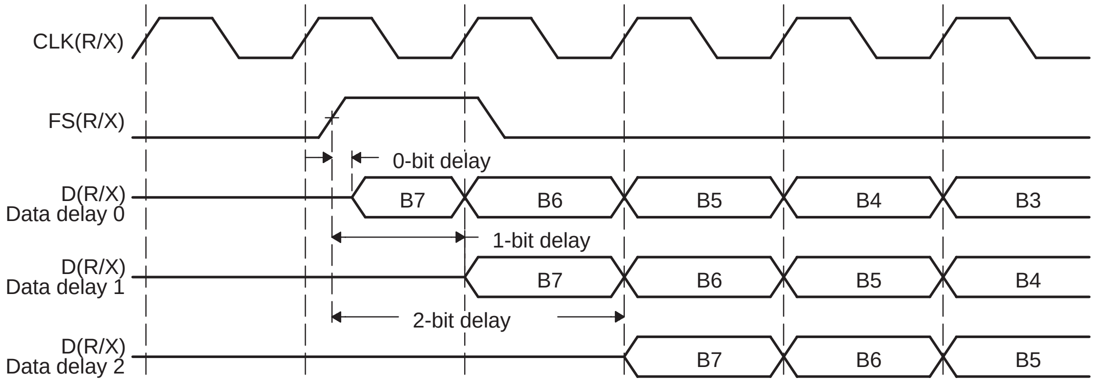
Register	Bit	Name	Function	
XCR2	1−0	XDATDLY	Transmitter data delay	
			XDATDLY = 00	0-bit data delay
			XDATDLY = 01	1-bit data delay
			XDATDLY = 10	2-bit data delay
			XDATDLY = 11	Reserved

8.13.1 About the 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.

XDATDLY specifies the data delay for transmission. The range of programmable data delay is zero to two bit-clocks (XDATDLY = 00b–10b), as described in Table 8−14 and Figure 8−12. 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-sync pulse.

Figure 8−12. Range of Programmable Data Delay



8.13.2 0-Bit Data Delay

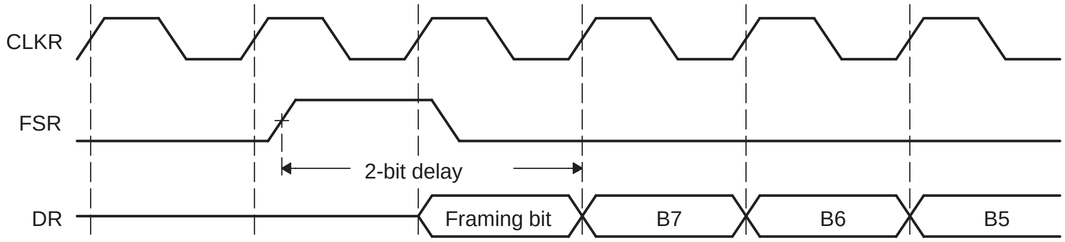
Normally, a frame-sync pulse is detected or sampled with respect to an edge of serial clock internal CLK(R/X). Thus, on the following cycle or later (depending on the data delay value), data may be received or transmitted. However, in the case of 0-bit data delay, the data must be ready for reception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

For reception, this problem is solved because receive data is sampled on the first falling edge of CLKR where an active-high internal FSR is detected. 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.

8.13.3 2-Bit Data Delay

A data delay of two bit periods allows the serial port to interface to different types of T1 framing 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 the following figure. In this figure, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on.

Figure 8−13. 2-Bit Data Delay Used to Skip a Framing Bit



8.14 Setting the Transmit DXENA Mode

The DXENA register bit, shown in Figure 8−14 and Table 8−15, is used to set the transmit DXENA (DX Delay Enabler) mode.

Figure 8−14. Register Bit Used to Set the Transmit DXENA (DX Delay Enabler) Mode

SPCR1

15 … 8	7	6 … 0
	DXENA	
	R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−15. Register Bit Used to Set the Transmit DXENA (DX Delay Enabler) Mode

Register	Bit	Name	Function	
SPCR1	7	DXENA	DX Delay Enabler Mode	
			DXENA = 0	DX delay enabler is off.
			DXENA = 1	DX delay enabler is on.

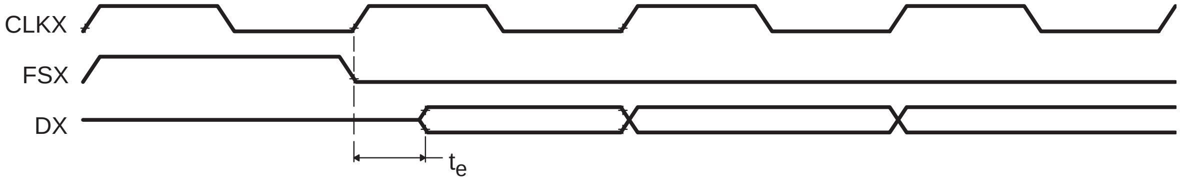
8.14.1 About the DXENA Mode

The DXENA bit controls the delay enabler on the DX pin. Set DXENA to enable an extra delay for turn-on time (for the length of the delay, see the data sheet for your TMS320C55x DSP). Note that this bit does not control the data itself, so only the first bit is delayed.

If you tie together the DX pins of multiple McBSPs, make sure DXENA = 1 to avoid having more than one McBSP transmit on the data line at one time.

The following figure shows the timing of the DX pin for DXENA = 1.

Figure 8−15. DX Delay When DXENA = 1



Note: te = extra delay for turn on time with DXENA = 1

8.15 Setting the Transmit Interrupt Mode

Figure 8−16. Register Bits Used to Set the Transmit Interrupt Mode

SPCR2

15 6	5 4	3 0
	XINTM	
	R/W−00	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−16. Register Bits Used to Set the Transmit Interrupt Mode

Register	Bit	Name	Function	
SPCR2	5−4	XINTM	Transmit Interrupt Mode	
			XINTM = 00	XINT generated when XRDY changes from 0 to 1
			XINTM = 01	XINT generated by an end-of-block or end-of-frame condition in a transmit multichannel selection mode
			XINTM = 10	XINT generated by a new transmit frame-sync pulse
			XINTM = 11	XINT generated when XSYNCERR is set

8.15.1 About the Transmitter Interrupt and the Associated Modes

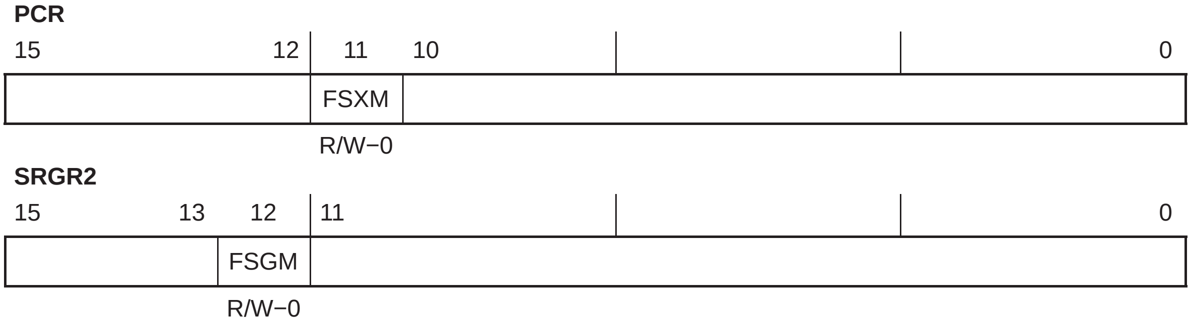
The transmitter interrupt (XINT) signals the CPU of changes to the serial port status. Four options exist for configuring this interrupt. The options are set by the transmit interrupt mode bits, XINTM, in SPCR2.

- XINTM = 00b. Interrupt on every serial word by tracking the XRDY bit in SPCR2. Note that regardless of the value of XINTM, XRDY can be read to detect the XRDY = 1 condition.
- XINTM = 01b. 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. In any other serial transfer case, this setting is not applicable and, therefore, no interrupts are generated.
- XINTM = 10b. Interrupt on detection of each transmit frame-sync pulse. This generates an interrupt even when the transmitter is in its reset state. This is done by synchronizing the incoming frame-sync pulse to the CPU clock and sending it to the CPU via XINT.
- XINTM = 11b. Interrupt on frame-synchronization error. Note that regardless of the value of XINTM, XSYNCERR can be read to detect this condition.

8.16 Setting the Transmit Frame-Sync Mode

The FSXM register bit, shown in Figure 8−17 and Table 8−17, is used to set the transmit frame-sync mode.

Figure 8−17. Register Bits Used to Set the Transmit Frame-Sync Mode



Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−17. Register Bits Used to Set the Transmit Frame-Sync Mode

Register	Bit	Name	Function	
PCR	11	FSXM	Transmit Frame-Synchronization Mode	
			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 of SRGR2.
SRGR2	12	FSGM	Sample Rate Generator Transmit Frame-Synchronization Mode	
			Used when FSXM = 1 in PCR.	
			FSGM = 0	The McBSP generates a transmit frame-sync pulse when the content of DXR[1,2] is copied to XSR[1,2].
			FSGM = 1	The transmitter uses frame-sync 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-sync period.

8.16.1 About the Transmit Frame-Sync Modes

Table 8−18 shows how FSXM and FSGM select the source of transmit frame-sync pulses. The three choices are:

❑ External frame-sync input

❑ Sample rate generator frame-sync signal (FSG).

❏ Internal signal that indicates a DXR-to-XSR copy has been made

Table 8−18 also shows the effect of each bit setting on the FSX pin. The polarity of the signal on the FSX pin is determined by the FSXP bit.

Table 8−18. How FSXM and FSGM Select the Source of Transmit Frame-Sync Pulses

FSXM	FSGM	Source of Transmit Frame Synchronization	FSX Pin Status
0	0 or 1	An external frame-sync 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-sync 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-sync pulse that is 1 cycle wide.	Output. The generated frame-sync pulse is inverted as determined by FSXP before being driven out on FSX pin.

8.16.2 Other Considerations

If the sample rate generator creates a frame-sync 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.

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.

8.17 Setting the Transmit Frame-Sync Polarity

The FSXP register bit, shown in Figure 8−18 and Table 8−19, is used to set the transmit frame-sync polarity.

Figure 8−18. Register Bit Used to Set Transmit Frame-Sync Polarity

PCR

15 4	3	2 0
	FSXP	
	R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−19. Register Bit Used to Set Transmit Frame-Sync Polarity

Register	Bit	Name	Function	
PCR	3	FSXP	Transmit Frame-Synchronization Polarity	
			FSXP = 0	Frame-synchronization pulse FSX is active high.
			FSXP = 1	Frame-synchronization pulse FSX is active low.

8.17.1 About Frame Sync Pulses, Clock Signals, and Their Polarities

Transmit frame-sync pulses can be either generated internally by the sample rate generator or driven by an external source. The source of frame sync is selected by programming the mode bit, FSXM, in PCR. FSX is also affected by the FSGM bit in SRGR2. Similarly, transmit clocks can be selected to be inputs or outputs by programming the mode bit, CLKXM, in the PCR.

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame-sync pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the DR pin is also sampled on the falling edge of internal CLKR. Note that these internal clock signals are either derived from 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 (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the DX pin is output on the rising edge of internal CLKX.

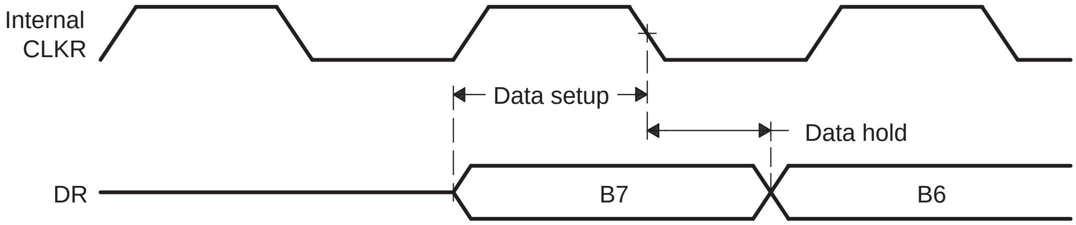
FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame-sync signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame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

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Note that 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. 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. Note that 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.

Note that 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 8−19 shows how data clocked by an external serial device using a rising edge, may be sampled by the McBSP receiver on the falling edge of the same clock.

Figure 8−19. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



8.18 Setting the SRG Frame-Sync Period and Pulse Width

The FPER register bit, shown in Figure 8−20 and Table 8−20, is used to set the SRG frame-sync period and pulse width.

Figure 8−20. Register Bits Used to Set the SRG Frame-Sync Period and Pulse Width

SRGR2

15 12	11 0
	FPER
	R/W−0000 0000 0000

SRGR1

15 8	7 0
FWID	
R/W−0000 0000	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−20. Register Bits Used to Set the SRG Frame-Sync Period and Pulse Width

Register	Bit	Name	Function
SRGR2	11−0	FPER	Sample Rate Generator Frame-Sync Period For the frame-sync signal FSG, (FPER + 1) determines the period from the start of a frame-sync pulse to the start of the next frame-sync pulse. Range for (FPER + 1): 1 to 4096 CLKG cycles.
SRGR1	15−8	FWID	Sample Rate Generator Frame-Sync Pulse Width This field plus 1 determines the width of each frame-sync pulse on FSG. Range for (FWID + 1): 1 to 256 CLKG cycles.

8.18.1 About the Frame-Sync Period and the Frame-Sync Pulse Width

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

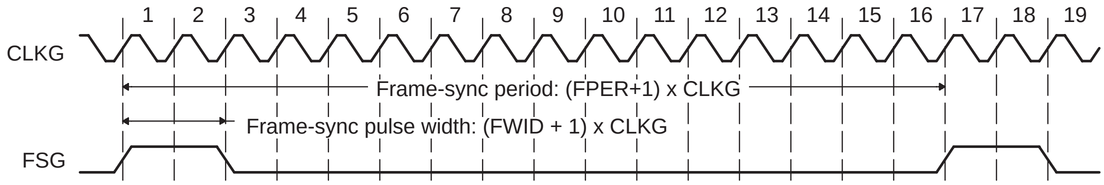
On FSG, the period from the start of a frame-sync pulse to the start of the next pulse is (FPER + 1) CLKG cycles. The 12 bits of FPER allow a frame-sync period of 1 to 4096 CLKG cycles, which allows up to 4096 data bits per frame. When GSYNC = 1, FPER is a don’t care value.

Each pulse on FSG has a width of (FWID + 1) CLKG cycles. The eight bits of FWID allow a pulse width of 1 to 256 CLKG cycles. It is recommended that FWID be programmed 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 8−21 shows a frame-sync period of 16 CLKG periods (FPER = 15 or 00001111b) and a frame-sync pulse with an active width of 2 CLKG periods (FWID = 1).

Figure 8−21. 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-sync period: (FPER+1) x CLKG
Frame-sync pulse width: (FWID + 1) x CLKG
FSG

When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when FRST = 1 and FSGM = 1, a frame-sync 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.

8.19 Setting the Transmit Clock Mode

The bit shown in Figure 8−22 and described in Table 8−21 determines the source for transmitclock and the function of the CLKX pin.

Figure 8−22. Register Bit Used to Set the Transmit Clock Mode

PCR

15 10	9	8 0
	CLKXM	
	R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−21. Register Bit Used to Set the Transmit Clock Mode

Register	Bit	Name	Function	
PCR	9	CLKXM	Transmit Clock Mode	
			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.

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

Table 8−22 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 8−22. 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.

8.19.2 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.

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 CLKXM = 1, so that CLKX is an output to supply 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.

8.20 Setting the Transmit Clock Polarity

The CLKXP register bit shown in Figure 8−23 and Table 8−23 is used to set transmit clock polarity.

Figure 8−23. Register Bit Used to Set Transmit Clock Polarity

PCR

15 … 2	1	0
	CLKXP	
	R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−23. Register Bit Used to Set Transmit Clock Polarity

Register	Bit	Name	Function	
PCR	1	CLKXP	Transmit Clock Polarity	
			CLKXP = 0	When the CLKX pin is configured as an input, the signal on the CLKX pin is not inverted before being used internally. When CLKX is configured as an output, the internal CLKX is not inverted before being driven on the CLKX pin. The transmit data is driven on the rising edge of the external CLKX signal.
			CLKXP = 1	When the CLKX pin is configured as an input, the signal on the CLKX pin is inverted before being used internally. When CLKX is configured as an output, the internal CLKX is not inverted before being driven on the CLKX pin. The transmit data is driven on the falling edge of the external CLKX signal.

8.20.1 About Frame Sync Pulses, Clock Signals, and Their Polarities

Transmit frame-sync pulses can be either generated internally by the sample rate generator or driven by an external source. The source of frame sync is selected by programming the mode bit, FSXM, in PCR. FSX is also affected by the FSGM bit in SRGR2. Similarly, transmit clocks can be selected to be inputs or outputs by programming the mode bit, CLKXM, in the PCR.

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame-sync pulses), the McBSP detects them on the internal falling edge of clock, internal

CLKR, and internal CLKX, respectively. The receive data arriving at the DR pin is also sampled on the falling edge of internal CLKR. Note that these internal clock signals are either derived from 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 (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the DX pin is output on the rising edge of internal CLKX.

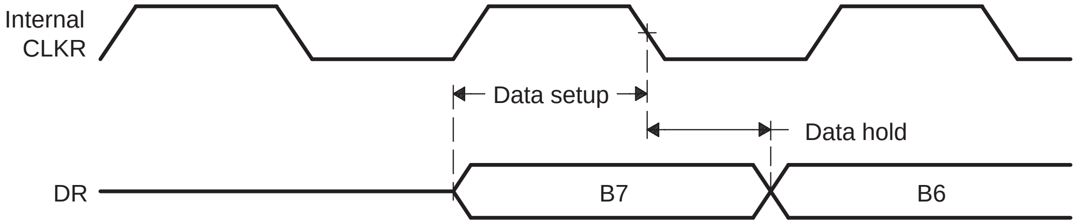
FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame-sync signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame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

Note that 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 8−24 shows how data clocked by an external serial device using a rising edge, may be sampled by the McBSP receiver on the falling edge of the same clock.

Figure 8−24. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



8.21 Setting the SRG Clock Divide-Down Value

The CLKGDV register bits shown in Figure 8−25 and Table 8−24 are used to set the sample rate generator clock divide-down value.

Figure 8−25. Register Bits Used to Set the Sample Rate Generator (SRG) Clock Divide-Down Value

SRGR1

15 8	7 0
	CLKGDV
	R/W−0000 0001

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−24. Register Bits Used to Set the Sample Rate Generator (SRG) Clock Divide-Down Value

Register	Bit	Name	Function
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).

8.21.1 About the 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 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.

Note:

The maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. For more information on programming the frequency of the slow peripheral clock, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* (SPRS206) data sheet or the *TMS320VC5502 Fixed-Point Digital Signal Processor* (SPRS166) data sheet. Other timing limitations may also apply. See the device-specific data manual for detailed information on the McBSP timing requirements.

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 (CLKGDV).

8.21.2 Set the SRG Clock Synchronization Mode

Figure 8−26. Register Bit Used to Set the SRG Clock Synchronization Mode

SRGR2

15	14	0
GSYNC		
R/W−0		

Legend: R = Read; W = Write; -*n* = Value after reset

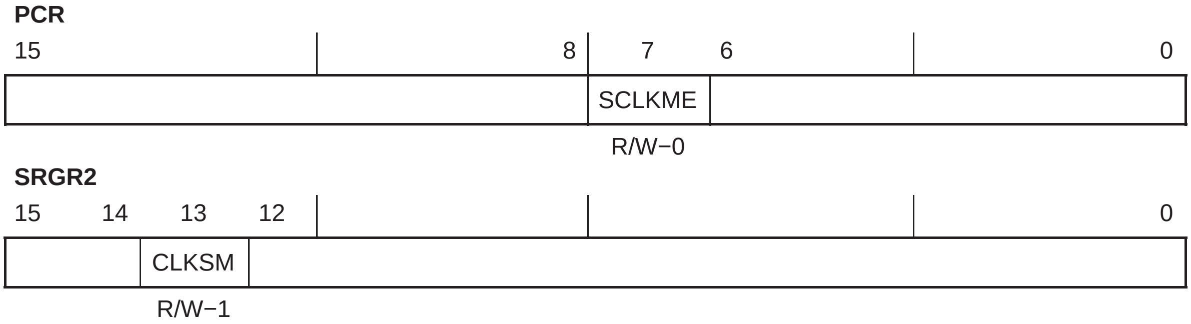
Table 8−25. Register Bit Used to Set the SRG Clock Synchronization Mode

Register	Bit	Name	Function
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, 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-sync period defined in FPER is ignored.

8.22 Setting the SRG Clock Mode (Choose an Input Clock)

The SCLKME register bits shown in Figure 8−27 and Table 8−26 are used to set the SRG clock mode.

Figure 8−27. Register Bits Used to Set the SRG Clock Mode (Choose an Input Clock)



Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−26. Register Bits Used to Set the SRG Clock Mode (Choose an Input Clock)

Register	Bit	Name	Function	
PCR SRGR2	7 13	SCLKME CLKSM	Sample Rate Generator Clock Mode	

			SCLKME = 0
			CLKSM = 0
				Sample rate generator clock derived from CLKS pin

			SCLKME = 0
			CLKSM = 1
				Sample rate generator clock derived from CPU clock (This is the condition forced by a DSP reset.)

			SCLKME = 1
			CLKSM = 0
				Sample rate generator clock derived from CLKR pin

			SCLKME = 1
			CLKSM = 1
				Sample rate generator clock derived from CLKX pin

8.22.1 About the 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. The preceding table shows the four possible sources of the input clock.

8.23 Setting the SRG Input Clock Polarity

The CLKSP register bits shown in Figure 8−28 and Table 8−27 are used to set the SRG input clock polarity.

Figure 8−28. Register Bits Used to Set the SRG Input Clock Polarity

SRGR2

15	14	13–0
	CLKSP	
	R/W−0	

PCR

15–2	1	0
	CLKXP	CLKRP
	R/W−0	R/W−0

Legend: R = Read; W = Write; -*n* = Value after reset

Table 8−27. Register Bits Used to Set the SRG Input Clock Polarity

Register	Bit	Name	Function	
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.
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.
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.

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

The sample rate generator can produce a clock signal (CLKG) and a frame-sync 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 CPU clock 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.

Chapter 9

General-Purpose I/O on the McBSP Pins

This chapter summarizes how to use the McBSP pins as general-purpose I/O pins.

Topic **Page**

9.1 Using the McBSP Pins

Table 9−1 summarizes how to use the McBSP pins as general-purpose I/O pins. All of the bits mentioned in the table except XRST and RRST are in the pin control register. XRST and RRST are in the serial port control registers.

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

- The receiver of the serial port is in reset (RRST = 0 in SPCR1).
- General-purpose I/O is 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 9−1 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 similar conditions:

- The transmitter of the serial port is in reset (XRST = 0 in SPCR2).
- General-purpose I/O is 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. The following table shows which bits in PCR are used to read from/write to these pins.

For the **CLKS pin**, all of the reset and I/O enable conditions must be met:

- Both the receiver and transmitter of the serial port are in reset (RRST = 0 and XRST = 0).
- General-purpose I/O is 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 CLKSSTAT bit in PCR.

Table 9−1. How Use 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	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

Note:

When the McBSP pins are configured as general−purpose input pins, CLKRP, CLKXP, CLKSP. FSRP and FSXP are not write−protected. If written, they will contain the written value until they are next automatically updated with the state of the associated pins. This behavior should be considered when polling these bits.

On the TMS320VC5509 and TMS320VC5510, these bits are updated on every occurrence of the CPU clock. On the TMS320VC5501 and TMS320VC5502, these bits are updated on every occurrence of the slow peripheral clock.

Chapter 10

Emulation, Power, and Reset Considerations

This chapter covers the following topics:

- ❑ How to program the McBSP's response to a breakpoint in the high-level language debugger
- ❑ How to conserve power in the DSP by placing the McBSP into its idle mode
- ❑ How to reset and initialize the various parts of the McBSP

Topic **Page**

10.1 McBSP Emulation Mode

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, upon a software breakpoint the clock continues to run and data is still 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, thus aborting a transmission. If SOFT = 1 and a breakpoint occurs while transmission is in progress, the transmission continues until completion of the word, and then the clock halts. These options are listed in Table 10−1.

The McBSP receiver functions in a similar fashion. Note that 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 10−1. McBSP Emulation Modes Selectable with the FREE and SOFT Bits of 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.

10.2 McBSP Power Management on the TMS320VC5509 and TMS320VC5510

The McBSP is placed into its idle mode with reduced power consumption when the PERIPH idle domain is idle (PERIS = 1 in ISTR) and the McBSP idle enable bit is set (IDLEEN = 1 in PCR).

In the McBSP idle mode:

❑ If the McBSP is configured to operate with internally generated clocking and frame synchronization, it will be completely stopped.

❑ If the McBSP is configured to operate with externally generated clocking and frame synchronization (either directly or through the sample rate generator), the external interface portion of the McBSP continues to function during external clock activity periods. The McBSP sends a request to activate the PERIPH and DMA idle domains when it needs to be serviced. If the domains were idle, they are made idle again after the McBSP has been serviced.

When IDLEEN = 0 in PCR, the McBSP keeps running, regardless of whether the PERIPH domain is idle.

10.3 McBSP Power Management on the TMS320VC5501 and TMS320VC5502

The McBSP is placed into its idle mode with reduced power consumption when the PERIPH idle domain is idle (PERIS = 1 in ISTR) and the McBSP idle enable bit is set (SPn = 1) in the Peripheral Idle Control Register (PICR). When the McBSP is in the Idle state, it is unable to receive or transmit data.

In the McBSP idle mode:

- ❑ If the McBSP is configured to operate with internally generated clocking and frame synchronization, it will be completely stopped.
- ❑ If the McBSP is configured to operate with externally generated clocking and frame synchronization (either directly or through the sample rate generator), the external interface portion of the McBSP continues to function during external clock activity periods. The McBSP sends a request to activate the PERIPH and DMA idle domains when it needs to be serviced. If the domains were idle, they are made idle again after the McBSP has been serviced.

When SPn= 0 in PICR, the McBSP keeps running, regardless of whether the PERIPH domain is idle.

10.4 Resetting and Initializing a McBSP

10.4.1 McBSP Pin States: DSP Reset Versus Receiver/Transmitter Reset

Table 10−2 shows the state of McBSP pins when the serial port is reset due to a DSP reset and due to a direct receiver or transmitter reset.

Table 10−2. Reset State of Each McBSP Pin

Pin	Possible State(s)	State Forced By DSP Reset	State Forced By Receiver/Transmitter Reset
			Receiver Reset (RRST = 0 and 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 (XRST = 0 and GRST = 1)
DX	O/Z	High impedance	High impedance
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
CLKS	I	Input	Input

Note: In Possible State(s) column, I = Input, O = Output, Z = High impedance

10.4.2 DSP Reset, McBSP Reset, and Sample Rate Generator Reset

When a DSP reset or a McBSP reset occurs, the McBSP 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.

- **DSP reset.** When the whole DSP is reset ($\overline{\text{RESET}}$ signal is driven low), the entire serial port, including the transmitter, receiver, and the sample rate generator, is reset. All input-only pins and three-state pins should be in a known state. The output-only pin DX is in the high-impedance state.

  The DSP reset forces the sample rate generator clock, CLKG, to have half the frequency of the CPU clock. No pulses are generated on the sample rate generator’s frame-sync 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 may be used as general-purpose I/O pins.

- **McBSP 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 in the corresponding section of the serial port stops. All input-only pins, such as DR and CLKS, and all other pins that are configured as inputs, are in a known state. The FSR and FSX pins are driven to their inactive state if they are not outputs. If the CLKR and CLKX pins are programmed as outputs, they will be driven by CLKG, provided that GRST = 1. Lastly, the DX pin will be 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 should 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) and its frame-sync signal (FSG) are 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 FRST = 1 and its frame synchronization is driven by FSG.

- **Sample rate generator reset.** The sample rate generator is reset when the DSP is reset or when GRST is loaded with 0. In the case of a DSP reset, the sample rate generator clock, CLKG, is driven by the CPU clock divided by 2, and the frame-sync 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 and FSG are driven inactive low. If you then set GRST, CLKG starts and runs as programmed. Later, if FRST = 1, FSG pulses active high after the programmed number of CLKG cycles has elapsed.

10.4.3 McBSP Initialization Procedure

The serial port initialization procedure is as follows:

1) Make XRST = RRST = FRST = GRST = 0 in SPCR[1,2]. If coming out of a DSP reset, this step is not required.

2) While the serial port is in the reset state, program only the McBSP configuration registers (not the data registers) as required.

3) Wait for two clock cycles. This ensures proper internal synchronization.

4) Set GRST = 1 to enable the sample rate generator.

5) Wait for two clock cycles. This ensures proper internal synchronization.

6) Set up data acquisition as required (such as writing to DXR[1,2]).

7) Make XRST = RRST = 1 to enable the serial port. Make sure that as you set these reset bits, you do not modify any of the other bits in SPCR1 and SPCR2. Otherwise, you will change the configuration you selected in step 2.

8) Set FRST = 1, if internally generated frame synchronization is required.

9) Wait two clock cycles for the receiver and transmitter to become active.

Alternatively, on either write (step 1 or 5), the transmitter and receiver may be placed in or taken out of reset individually by modifying the desired bit.

The above procedure for reset/initialization can be applied in general when the receiver or transmitter has to be reset during its normal operation, and also when the sample rate generator is not used for either operation.

Notes:

1) The necessary duration of the active-low period of XRST or 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] should 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]) should be loaded by the CPU or by the DMA controller only when the transmitter is enabled (XRST = 1). An exception to this rule is when these registers are used for companding internal data.

4) The bits of 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.

Chapter 11

Data Packing Examples

This section shows two ways you can implement data packing in the McBSP.

Topic **Page**

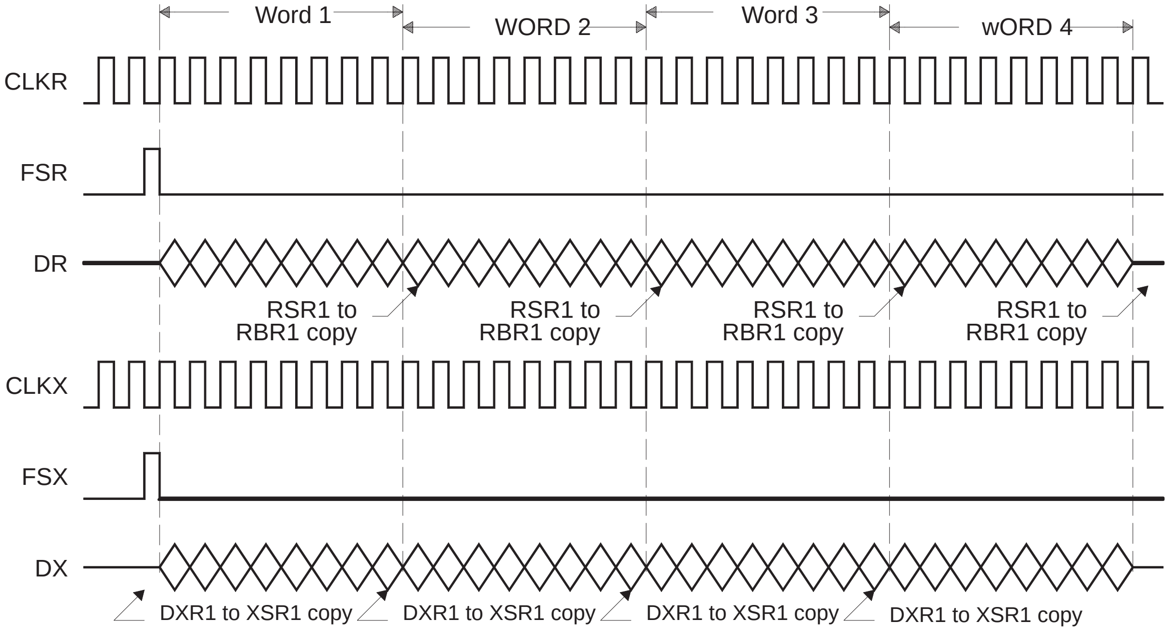
11.1 Data Packing Using Frame Length and Word Length

The 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 11−1. 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 CPU or by the DMA controller. Thus, four reads from DRR1 and four writes to DXR1 are necessary for each frame.

Figure 11−1.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 11−2. In this case:

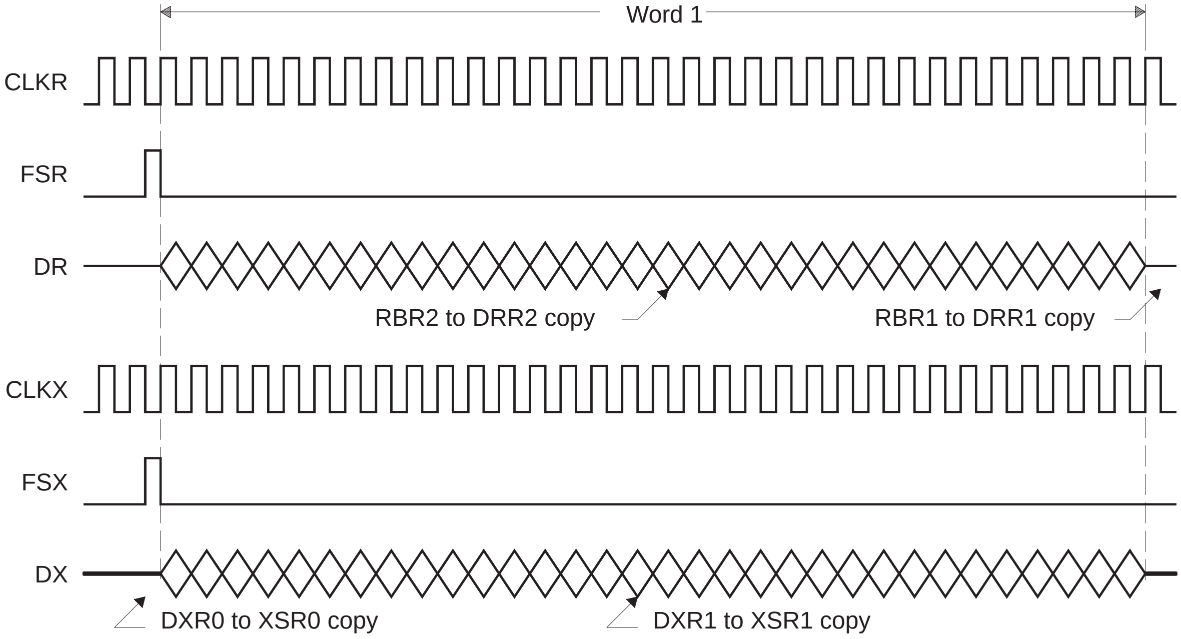
- ❑ (R/X)PHASE = 0: Single-phase frame
- ❑ (R/X)FRLEN1 = 0000000b: 1-word frame
- ❑ (R/X)WDLEN1 = 101b: 32-bit word

Two 16-bit data words are transferred to and from the McBSP by the CPU or 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, make sure you access DRR2/DXR2 before you access 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 11−2.One 32-Bit Data Word Transferred To/From the McBSP



11.2 Data Packing Using Word Length and the Frame-Sync 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-sync pulses. First, consider Figure 11−3, which shows the McBSP operating at the maximum packet frequency. Here, each frame only has a single 8-bit word. Note the frame-sync 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 11−3.8-Bit Data Words Transferred at Maximum Packet Frequency

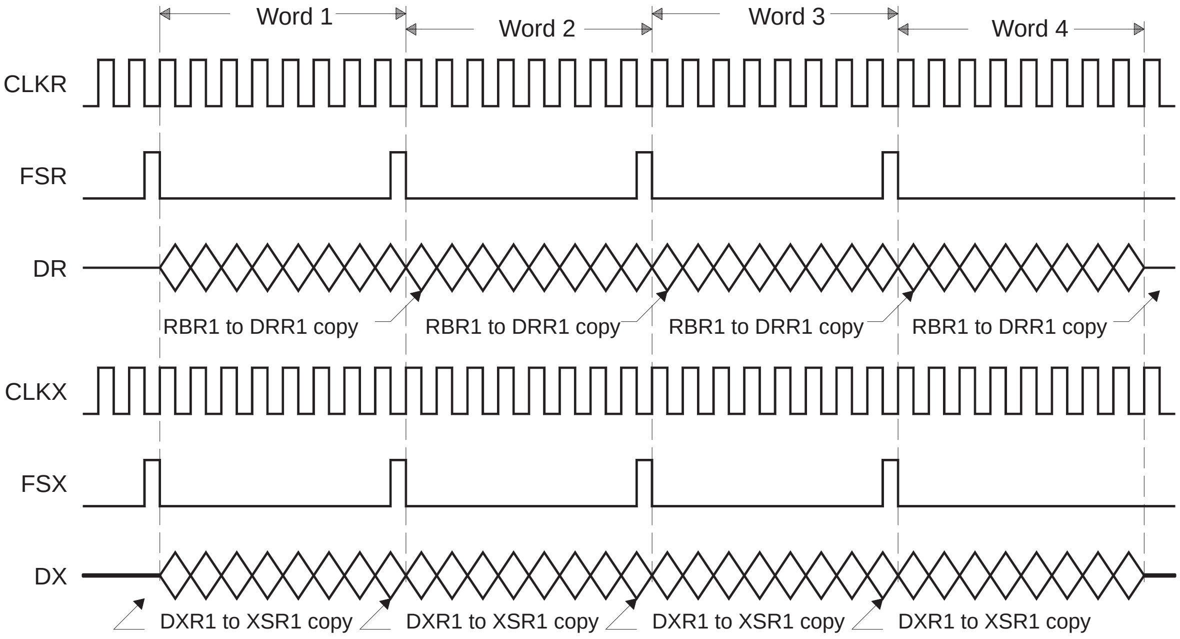
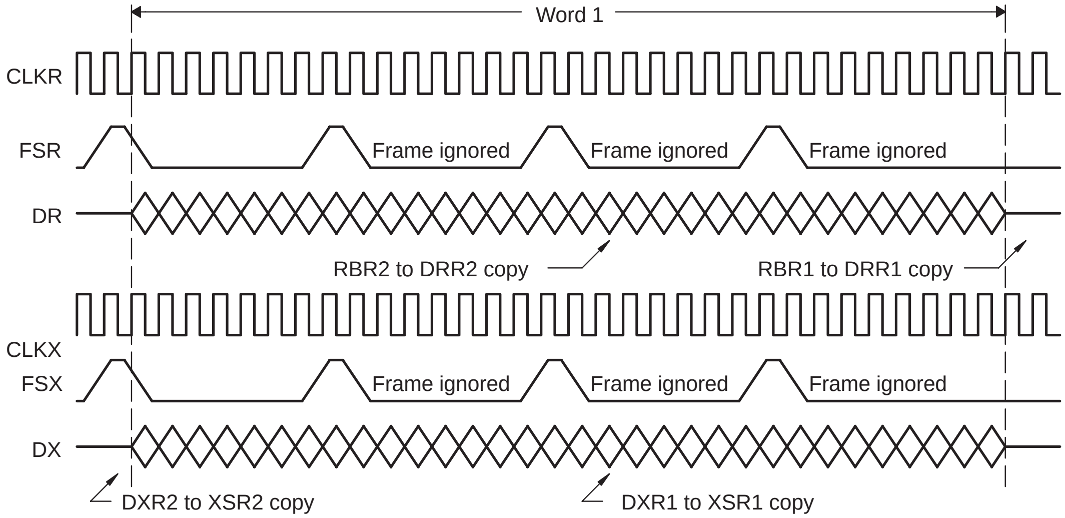


Figure 11−4 shows the McBSP configured to treat this data stream as a continuous 32-bit word. In this example, the McBSP responds to an initial frame-sync pulse. However, (R/X)FIG = 1 so that the McBSP ignores subsequent pulses. Only two read transfers or two write transfers are needed every 32 bits. This configuration effectively reduces the required bus bandwidth to half the bandwidth needed to transfer four 8-bit words.

Figure 11−4. Configuring the Data Stream of Figure 11−3 as a Continuous 32-Bit Word



Note:

The data packing method used here does not apply to the TMS320VC5501 or the TMS320VC5502. These devices require at least one inactive cycle between words so this data packing method is not possible.

Chapter 12

McBSP Registers

The McBSP registers are detailed and defined in this chapter. For each McBSP, the DSP contains the following registers. For the I/O address of each register, see the data sheet for your TMS320C55x DSP.

Topic **Page**

12.1 Data Receive Registers (DRR2 and DRR1)

The CPU or the DMA controller reads received data from one or both of the data receive registers (see Figure 12−1). 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 12−1. 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

12.1.1 How Data Travels From the Data Receive (DR) Pin to the DRRs

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 CPU or by the 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 both of the DRRs, which can be read by the CPU or by the 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.

12.2 Data Transmit Registers (DXR2 and DXR1)

For transmission, the CPU or the DMA controller writes data to one or both of the data transmit registers (see Figure 12−2). 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 12−2. 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

12.2.1 How Data Travels From the DXRs to the Data Transmit (DX) Pin

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 to both 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.

12.3 Serial Port Control Registers (SPCR1 and SPCR2)

Each McBSP has two serial port control registers of the form shown in Figure 12−3. Table 12−1 and Table 12−2 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)

Figure 12−3. Serial Port Control Registers (SPCR1 and SPCR2)

SPCR1

15	14–13	12–11	10–8
DLB	RJUST	CLKSTP	reserved
R/W-0	R/W-00	R/W-00	

7	6	5–4	3	2	1	0
DXENA	rsvd	RINTM	RSYNCERR	RFULL	RRDY	RRST
R/W-0	R/W-0	R/W-00	R/W-0	R-0	R-0	R/W-0

SPCR2

15–10	9	8
reserved	FREE	SOFT
	R/W-0	R/W-0

7	6	5–4	3	2	1	0
FRST	GRST	XINTM	XSYNCERR	XEMPTY	XRDY	XRST
R/W-0	R/W-0	R/W-00	R/W-0	R-0	R-0	R/W-0

Legend: R = Read; W = Write; -*n* = Value after reset

Table 12−1. SPCR1 Bit Descriptions

Bit	***field***[†]	Value	Description
15	DLB		Digital loopback mode bit. DLB disables or enables the digital loopback mode of the McBSP:
		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.

Table 12−1. SPCR1 Bit Descriptions (Continued)

Bit	*field*†	Value	Description
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). **Note:** 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.
		00b	Right justify the data and zero fill the MSBs.
		01b	Right justify the data and sign-extend the data into the MSBs.
		10b	Left justify the data and zero fill the LSBs.
		11b	Reserved (do not use)
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).
		00b or 01b	Clock stop mode is disabled.
		10b	Clock stop mode, without clock delay
		11b	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. DXENA controls the delay enabler for the DX pin. The enabler creates an extra delay for turn-on time (for the length of the delay, see the data sheet for your TMS320C55x DSP).
		0	DX delay enabler off
		1	DX delay enabler on

Table 12−1. SPCR1 Bit Descriptions (Continued)

Bit	***field***†	Value	Description
6	Reserved		Reserved bits (not available for your use). They are read-only bits and return 0s when read. This bit must be written as 0.
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 CPU, the CPU will service the interrupt request; otherwise, the CPU will ignore the request.
		00b	The McBSP sends a receive interrupt (RINT) request to the CPU 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]): Note: Regardless of the value of RINTM, you can check RRDY to determine whether a word transfer is complete.
		01b	In the multichannel selection mode, the McBSP sends a RINT request to the CPU after every 16-channel block is received in a frame. Outside of the multichannel selection mode, no interrupt request is sent.
		10b	The McBSP sends a RINT request to the CPU when each receive frame-sync pulse is detected. The interrupt request is sent even if the receiver is in its reset state.
		11b	The McBSP sends a RINT request to the CPU when the RSYNCERR bit is set, indicating a receive frame-sync error. Note: Regardless of the value of RINTM, you can check RSYNCERR to determine whether a receive frame-sync error occurred.
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 CPU 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-sync error occurred.
		0	No error
		1	Receive frame-sync error
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).
		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 12−1. SPCR1 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description
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 CPU 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 DMA controller.
		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 need (word length larger than 16 bits), the CPU or the 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. **Note:** This bit has a negative polarity; RRST = 0 indicates the reset state.
		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 12−2. SPCR2 Bit Descriptions

Bit	***field***†	Value	Description
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.
		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.
		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. On the TMS320VC5501 and TMS320VC5502, the SOFT operation works as described above. On the TMS320VC5510 and TMS320VC5509, support for SOFT=1 is not available. If FREE = 0 and SOFT = 1, the serial port will continue to run on an emulation breakpoint.

Table 12−2. SPCR2 Bit Descriptions (Continued)

Bit	*field*†	Value	Description
7	FRST		Frame-sync logic reset bit. The sample rate generator of the McBSP includes frame-sync logic to generate an internal frame-sync signal. You can use FRST to take the frame-sync logic into and out of its reset state. **Note:** This bit has a negative polarity; FRST = 0 indicates the reset state.
		0	If you read a 0, the frame-sync logic is in its reset state. If you write a 0, you reset the frame-sync logic. In the reset state, the frame-sync logic does not generate a frame-sync signal (FSG).
		1	If you read a 1, the frame-sync logic is enabled. If you write a 1, you enable the frame-sync logic by taking it out of its reset state. When the frame-sync logic is enabled (FRST = 1) and the sample rate generator as a whole is enabled (GRST = 1), the frame-sync logic generates the frame-sync signal FSG as programmed.
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. **Note:** This bit has a negative polarity; GRST = 0 indicates the reset state.
		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 DSP reset, CLKG is driven by the CPU clock 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-sync signal FSG as programmed in the sample rate generator registers.

Table 12−2. SPCR2 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description
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 CPU will service the interrupt request; otherwise, the CPU will ignore the request.
		00b	The McBSP sends a transmit interrupt (XINT) request to the CPU 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]): **Note:** Regardless of the value of XINTM, you can check XRDY to determine whether a word transfer is complete.
		01b	In the multichannel selection mode, the McBSP sends an XINT request to the CPU 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 CPU when each transmit frame-sync pulse is detected. The interrupt request is sent even if the transmitter is in its reset state.
		11b	The McBSP sends an XINT request to the CPU when the XSYNCERR bit is set, indicating a transmit frame-sync error. Note: Regardless of the value of XINTM, you can check XSYNCERR to determine whether a transmit frame-sync error occurred.
3	XSYNCERR		Transmit frame-sync error bit. XSYNCERR is set when a transmit frame-sync error is detected by the McBSP. If XINTM = 11b, the McBSP sends a transmit interrupt (XINT) request to the CPU when XSYNCERR is set. The flag remains set until you write a 0 to it or reset the transmitter. **Caution:** if XINTM = 11b, writing a 1 to XSYNCERR triggers a transmit interrupt just as if a transmit frame-sync error occurred.
		0	No error
		1	Transmit frame-sync error

Table 12−2. SPCR2 Bit Descriptions (Continued)

Bit	*field*†	Value	Description
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). **Note:** This bit has a negative polarity; a transmitter-empty condition is indicated by XEMPTY = 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.
		1	No transmitter-empty condition
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 CPU 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 DMA controller.
		0	Transmitter not ready When DXR1 is loaded, XRDY is automatically cleared.
		1	Transmitter ready: DXR[1,2] is ready to accept new data. **Important:** If both DXRs are needed (word length larger than 16 bits), the CPU or the 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. **Note:** This bit has a negative polarity; XRST = 0 indicates the reset state.
		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.

12.4 Receive Control Registers (RCR1 and RCR2)

Each McBSP has two receive control registers of the form shown in Figure 12−4. Table 12−3 and Table 12−4 describe the bis of 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-sync ignore function (RFIG)
- ❑ Choose a receive data delay (RDATDLY)

Figure 12−4. Receive Control Registers (RCR1 and RCR2)

RCR1

15	14 8
reserved	RFRLEN1
R−0	R/W−0

7 5	4 0
RWDLEN1	reserved
R/W−000	R−0

RCR2

15	14 8
RPHASE	RFRLEN2
R/W−0	R/W−0

7 5	4 3	2	1 0
RWDLEN2	RCOMPAND	RFIG	RDATDLY
R/W-000	R/W−00	R/W−0	R/W−00

Legend: R = Read; W = Write; -*n* = Value after reset

Table 12−3. RCR1 BIt Descriptions

Bit	*field*[†]	Value	Description
15	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.
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 value that you load into the RPHASE bit. If a single-phase frame is selected, RFRLEN1 in RCR1 selects the number of serial words 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 the following table for a summary of how you determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period. **Note:** 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.

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 12−3. RCR1 BIt Descriptions (Continued)

Bit	*field*[†]	Value	Description
7–5	RWDLEN1		Receive word length 1. Each frame of receive data 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 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.
		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.

Table 12−4. RCR2 Bit Descriptions

Bit	*field*†	Value	Description
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.
		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.
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 value that you load into the RPHASE bit. If a single-phase frame is selected, RFRLEN1 in RCR1 selects the number of serial words 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 the following table 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. **Note:** 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 2, load 127 into RFRLEN2.

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 12−4. RCR2 BIt Descriptions (Continued)

Bit	***field***[†]	Value	Description
7–5	RWDLEN2		Receive word length 2. Each frame of receive data 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 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.
		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	RCOMPAND		Receive companding mode bits. Companding (COMpress and exPAND) hardware allows compression and expansion of data in either μ-law or A-law format. RCOMPAND allows you to choose one of the following companding modes for the McBSP receiver:
		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-sync ignore bit. If a frame-sync pulse starts the transfer of a new frame before the current frame is fully received, this pulse is treated as an unexpected frame-sync pulse. Setting RFIG causes the serial port to ignore unexpected frame-sync signals during reception.

Table 12−4. RCR2 BIt Descriptions (Continued)

Bit	***field***†	Value	Description
		0	Frame-sync 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-sync ignore. An unexpected FSR pulse is ignored. Reception continues uninterrupted.
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.
		00b	0-bit data delay
		01b	1-bit data delay
		10b	2-bit data delay
		11b	Reserved (do not use)

12.5 Transmit Control Registers (XCR1 and XCR2)

Each McBSP has two transmit control registers of the form shown in Figure 12−5. Table 12−5 and Table 12−6 describe the bits of 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)

Figure 12−5. Transmit Control Registers (XCR1 and XCR2)

XCR1

<table>
<tr><td>15</td><td>14</td><td>8</td></tr>
<tr><td>reserved</td><td colspan="2">XFRLEN1</td></tr>
<tr><td>R−0</td><td colspan="2">R/W−0</td></tr>
</table>

<table>
<tr><td>7</td><td>5</td><td>4</td><td>0</td></tr>
<tr><td colspan="2">XWDLEN1</td><td colspan="2">reserved</td></tr>
<tr><td colspan="2">R/W−000</td><td colspan="2">R−0</td></tr>
</table>

XCR2

<table>
<tr><td>15</td><td>14</td><td>8</td></tr>
<tr><td>XPHASE</td><td colspan="2">XFRLEN2</td></tr>
<tr><td>R/W−0</td><td colspan="2">R/W−0</td></tr>
</table>

<table>
<tr><td>7</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">XWDLEN2</td><td colspan="2">XCOMPAND</td><td>XFIG</td><td colspan="2">XDATDLY</td></tr>
<tr><td colspan="2">R/W-000</td><td colspan="2">R/W−00</td><td>R/W−0</td><td colspan="2">R/W−00</td></tr>
</table>

Legend: R = Read; W = Write; -*n* = Value after reset

Table 12−5. XCR1 Bit Descriptions

Bit	*field*[†]	Value	Description
15	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.
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 value that you load into the XPHASE bit. If a single-phase frame is selected, XFRLEN1 in XCR1 selects the number of serial words 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 the following table for a summary of how you determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period. **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.

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 12−5. XCR1 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description
7–5	XWDLEN1		Transmit word length 1. Each frame of transmit data can have one or two phases, depending on the value that you load 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.
		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.

Table 12−6. XCR2 Bit Descriptions

Bit	*field*[†]	Value	Description
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.
		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 value that you load into the XPHASE bit. If a single-phase frame is selected, XFRLEN1 in XCR1 selects the number of serial words 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 the following table 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. **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.

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 12−6. XCR2 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description
7–5	XWDLEN2		Transmit word length 2. Each frame of transmit data can have one or two phases, depending on the value that you load 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.
		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. XCOMPAND allows you to choose one of the following companding modes for the McBSP transmitter.
		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 12−6. XCR2 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description
2	XFIG		Transmit frame-sync ignore bit. If a frame-sync pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame-sync pulse. Setting XFIG causes the serial port to ignore unexpected frame-sync pulses during transmission.
		0	Frame-sync 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-sync ignore. An unexpected FSX pulse is ignored. Transmission continues uninterrupted.
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.
		00b	0-bit data delay
		01b	1-bit data delay
		10b	2-bit data delay
		11b	Reserved (do not use)

12.6 Sample Rate Generator Registers (SRGR1 and SRGR2)

Each McBSP has two sample rate generator registers of the form shown in Figure 12−6. Table 12−7 and Table 12−8 describe the bits of SRGR1 and SRGR2, respectively. The sample rate generator can generate a clock signal (CLKG) and a frame-sync 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 of PCR)
- Divide down the frequency of CLKG (CLKGDV)
- Select whether internally-generated transmit frame-sync pulse are driven by FSG or by activity in the transmitter (FSGM).
- Specify the width of frame-sync 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 of SRGR2 allows you to make CLKG synchronized to an external frame-sync signal on the FSR pin, so that CLKG is kept in phase with the input clock.

Figure 12−6. Sample Rate Generator Registers (SRGR1 and SRGR2)

SRGR1

15		8
	FWID	
	R/W−0	

7		0
	CLKGDV	
	R/W−1	

SRGR2

15	14	13	12	11
GSYNC	CLKSP	CLKSM	FSGM	FPER
R/W-0	R/W-0	R/W-1	R/W-0	R/W−0

	0
FPER	
R/W−0	

Legend: R = Read; W = Write; -*n* = Value after reset

Table 12−7. SRGR1 Bit Descriptions

<table>
<tr><th>Bit</th><th><i>field</i>[†]</th><th>Value</th><th>Description</th></tr>
<tr><td>15–8</td><td>FWID</td><td>0−255</td><td>Frame-sync pulse width bits for FSG
The sample rate generator can produce a clock signal, CLKG, and a frame-sync signal, FSG. For frame-sync pulses on FSG, (FWID + 1) is the pulse width in CLKG cycles. The eight bits of 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-sync pulses on FSG is defined by the FPER bits.</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>CPU clock</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 DSP reset forces the CLKG frequency to 1/2 the input clock frequency (CLKGDV = 1), and the reset selects the CPU clock as the input clock.
The maximum frequency for the McBSP on the TMS320VC5509 and TMS320VC5510 is 1/2 the CPU clock frequency. The maximum frequency for the McBSP on the TMS320VC5501 and TMS320VC5502 is 1/2 the frequency of the slow peripheral clock. See the device-specific data manual for timing requirements for the McBSP.</td></tr>
</table>

Table 12−8. SRGR2 Bit Descriptions

Bit	***field***†	Value	Description	Description
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-sync signal (FSG) generated by the sample rate generator are made dependent on pulses on the FSR pin.	
		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. ❑ FSG pulses. FSG **only** pulses in response to a pulse on the FSR pin. The frame-sync period defined in FPER is ignored.	
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-sync signal (FSG) that are generated by the sample rate generator:	
		0	A rising edge on the CLKS pin	
		1	A falling edge on the CLKS pin	

Table 12−8. SRGR2 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description	Description	
13	CLKSM		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 DSP reset selects the CPU clock as the input clock and forces the CLKG frequency to 1/2 the CPU clock frequency.		
		0	The input clock for the sample rate generator is taken from the CLKS pin or from the CLKR pin, depending on the value of the SCLKME bit of PCR:		
			SCLKME	**CLKSM**	**Input Clock For Sample Rate Generator**
			0	0	Signal on CLKS pin
			1	0	Signal on CLKR pin
		1	The input clock for the sample rate generator is taken from the CPU clock or from the CLKX pin, depending on the value of the SCLKME bit of PCR:		
			SCLKME	**CLKSM**	**Input Clock For Sample Rate Generator**
			0	1	CPU clock
			1	1	Signal on CLKX pin

Table 12−8. SRGR2 Bit Descriptions (Continued)

Bit	***field***†	Value	Description Description
12	FSGM		Sample rate generator transmit frame-sync 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-sync pulses.
		0	If FSXM = 1, the McBSP generates a transmit frame-sync pulse when the content of DXR[1,2] is copied to XSR[1,2].
		1	If FSXM = 1, the transmitter uses frame-sync 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 period between pulses.
11–0	FPER	0–4095	Frame-sync period bits for FSG. The sample rate generator can produce a clock signal, CLKG, and a frame-sync signal, FSG. The period between frame-sync pulses on FSG is (FPER + 1) CLKG cycles. The 12 bits of FPER allow a frame-sync 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-sync pulse on FSG is defined by the FWID bits.

12.7 Multichannel Control Registers (MCR1 and MCR2)

Each McBSP has two multichannel control registers of the form shown in Figure 12−7. MCR1 has control and status bits (with an R prefix) for multichannel selection operation in the receiver. MCR2 contains the same type of bits (bit with an X prefix) for the transmitter. The bits of MCR1 and MCR2 are described in Table 12−9 and Table 12−10, 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)

Figure 12−7. Multichannel Control Registers (MCR1 and MCR2)

MCR1

15 10	9	8
reserved	RMCME	RPBBLK
R−0	R/W-0	R/W-00

7	6 5	4 2	1	0
RPBBLK	RPABLK	RCBLK	reserved	RMCM
R/W-00	R/W-00	R-000	R−0	R/W-0

MCR2

15 10	9	8
reserved	XMCME	XPBBLK
R−0	R/W-0	R/W-00

7	6 5	4 2	1 0
XPBBLK	XPABLK	XCBLK	XMCM
R/W-00	R/W-00	R-000	R/W-00

Legend: R = Read; W = Write; -*n* = Value after reset

Table 12−9. MCR1 Bit Descriptions

Bit	***field***[†]	Value	Description
15−10	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.
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.
		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 12−9. MCR1 Bit Descriptions (Continued)

Bit	***field*** †	Value	Description
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. **Note:** 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).
		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
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.
		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

Table 12−9. MCR1 Bit Descriptions (Continued)

Bit	*field*†	Value	Description
4–2	RCBLK		Receive current block indicator. RCBLK indicates which block of 16 channels is involved in the current McBSP reception:
		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.
0	RMCM		Receive multichannel selection mode bit. RMCM determines whether all channels or only selected channels are enabled for reception:
		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 12−10. MCR2 Bit Descriptions

Bit	*field*[†]	Value	Description
15−10	Reserved	0	Reserved bits (not available for your use). They are read-only bits and return 0s when read.
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).
		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
		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

Table 12−10. MCR2 Bit Descriptions (Continued)

Bit	*field*[†]	Value	Description
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. **Note:** 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).
		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
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.
		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

Table 12−10. MCR2 Bit Descriptions (Continued)

Bit	***field***[†]	Value	Description
4–2	XCBLK		Transmit current block indicator. XCBLK indicates which block of 16 channels is involved in the current McBSP transmission:
		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
1–0	XMCM		Transmit multichannel selection mode bits. XMCM determines whether all channels or only selected channels are enabled and unmasked for transmission.
		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.

12.8 Pin Control Register (PCR)

Each McBSP has one pin control register of the form shown in Figure 12−8. Table 12−11 describes the bits of 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 of ICR). For the TMS320VC5509 and TMS320VC5510, this capability is provided in the PCR. On the TMS320VC5501 and TMS320VC5502, this capability is provided in the Peripheral Idle Control Register (PICR). For more information on the TMS320VC5501 implementation, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* data manual (SPRS206); for the TMS320VC5502 implementation, see the *TMS320VC5502 Fixed-Point Digital Signal Processor* data manual (SPRS166).
- 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-sync 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 of 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-sync 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)

Figure 12−8. Pin Control Register (PCR) for TMS320VC5509 and TMS320VC5510

15	14	13	12	11	10	9	8
reserved	IDLEEN	XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM
R−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
SCLKME	CLKSSTAT	DXSTAT	DRSTAT	FSXP	FSRP	CLKXP	CLKRP
R/W-0	R−0	R/W-0	R−0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read; W = Write; -*n* = Value after reset

Figure 12−9. Pin Control Register (PCR) for TMS320VC5501 and TMS320VC5502

15	14	13	12	11	10	9	8
reserved	reserved	XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM
R−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
SCLKME	CLKSSTAT	DXSTAT	DRSTAT	FSXP	FSRP	CLKXP	CLKRP
R/W-0	R−0	R/W-0	R−0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read; W = Write; -*n* = Value after reset

Table 12−11. PCR Bit Descriptions

Bit	***field***[†]	Value	Description
15	Reserved	0	Reserved bit (not available for your use). It is a read-only bit and returns a 0 when read.
14	IDLEEN		Idle enable bit. If the PERIPH idle domain is configured to be idle and IDLEEN = 1, the McBSP stops and enters a low-power state. On the TMS320VC5501 and TMS320VC5502, this bit is reserved and should be written as 0. The IDLEEN function is implemented in the Peripheral Idle Control Register (PICR). For more information on the PICR, see the *TMS320VC5501 Fixed-Point Digital Signal Processor* data manual (SPRS206) or the *TMS320VC5502 Fixed-Point Digital Signal Processor* data manual (SPRS166).
		0	The McBSP remains active when the PERIPH domain is idled.
		1	If the PERIPH domain is idle (PERIS = 1 in the idle status register), the McBSP is stopped in a low-power state.
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 the table that follows the RIOEN bit description.
		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 the table that follows the RIOEN bit description. XRST and RRST are in the serial port control registers, but all other bits mentioned in this table are in the pin control register.
		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.

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

Table 12−11.PCR Bit Descriptions (Continued)

Bit	field[†]	Value	Description
11	FSXM		Transmit frame-sync mode bit. FSXM determines whether transmit frame-sync pulses are supplied externally or internally. The polarity of the signal on the FSX pin is determined by the FSXP bit.
		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 of SRGR2.
10	FSRM		Receive frame-sync mode bit. FSRM determines whether receive frame-sync pulses are supplied externally or internally. The polarity of the signal on the FSR pin is determined by the FSRP bit.
		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.

Table 12−11.PCR Bit Descriptions (Continued)

Bit	field[†]	Value	Description
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):
		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.
			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.

Table 12−11.PCR Bit Descriptions (Continued)

<table>
<tr><th>Bit</th><th>field[†]</th><th>Value</th><th>Description</th></tr>
<tr><td>8</td><td>CLKRM</td><td></td><td>Receive clock mode bit. The role of CLKRM and the resulting effect on the CLKR pin depend on whether the McBSP is in the digital loopback mode (DLB = 1).
<b>Note:</b> The polarity of the signal on the CLKR pin is determined by the CLKRP bit.
<b>NOT in digital loopback mode (DLB = 0):</b></td></tr>
<tr><td></td><td></td><td>0</td><td>The CLKR pin is an input pin that supplies the internal receive clock (CLKR).</td></tr>
<tr><td></td><td></td><td>1</td><td>Internal CLKR is driven by the sample rate generator of the McBSP. The CLKR pin is an output pin that reflects internal CLKR.
<b>In digital loopback mode (DLB = 1):</b></td></tr>
<tr><td></td><td></td><td>0</td><td>The CLKR pin is in the <b>high impedance state</b>. The internal receive clock (CLKR) is driven by the internal transmit clock (CLKX). CLKX is derived according to the CLKXM bit.</td></tr>
<tr><td></td><td></td><td>1</td><td>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.</td></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></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 CPU clock, depending on the value of the CLKSM bit of 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>CPU clock</td></tr>
</table></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 of 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></tr>
</table>

Table 12−11.PCR Bit Descriptions (Continued)

Bit	field†	Value	Description
6	CLKSSTAT		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).
		0	The signal on the CLKS pin is low.
		1	The signal on the CLKS pin is high.
5	DXSTAT		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).
		0	Drive the signal on the DX pin low.
		1	Drive the signal on the DX pin high.
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).
		0	The signal on DR pin is low.
		1	The signal on DR pin is high.
3	FSXP		Transmit frame-sync polarity bit. FSXP determines the polarity of FSX as seen on the FSX pin.
		0	Transmit frame-sync pulses are active high.
		1	Transmit frame-sync pulses are active low.
2	FSRP		Receive frame-sync polarity bit. FSRP determines the polarity of FSR as seen on the FSR pin.
		0	Receive frame-sync pulses are active high.
		1	Receive frame-sync pulses are active low.

Table 12−11.PCR Bit Descriptions (Continued)

Bit	field[†]	Value	Description
1	CLKXP		Transmit clock polarity bit. CLKXP determines the polarity of CLKX as seen on the CLKX pin.
		0	Transmit data is driven on the rising edge of CLKX.
		1	Transmit data is driven on the falling edge of CLKX.
0	CLKRP		Receive clock polarity bit. CLKRP determines the polarity of CLKR as seen on the CLKR pin.
		0	When the CLKR pin is configured as an input, the external CLKR is not inverted before being used internally and the receive data is sampled on the falling edge of CLKR. When the CLKR pin is configured an as output, the internal CLKR is not inverted before being driven on the pin.
		1	When the CLKR pin is configured as an input, the external CLKR is inverted before being used internally and the receive data is sampled on the rising edge of CLKR. When the CLKR pin is configured an as output, the internal CLKR is inverted before being driven on the pin.

12.9 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 12−10. There is one for each of the receive partitions: A, B, C, D, E, F, G, and H. Table 12−12 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 12−10. Format of the 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 12−12. Description For Bit x of a Receive Channel Enable Register (x = 0, 1, 2, ..., or 15)

Bit	Field	Value	Description
x	RCEx		Receive channel enable bit.
			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.

12.9.1 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 12−13 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 12−13. 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)
128 (RMCME = 1)	RCERA	Block 0	RCE0 RCE1 RCE2 : RCE15	Channel 0 Channel 1 Channel 2 : Channel 15
	RCERB	Block 1	RCE0 RCE1 RCE2 : RCE15	Channel 16 Channel 17 Channel 18 : Channel 31

Table 12−13. Use of the Receive Channel Enable Registers (Continued)

Number of Selectable Channels	Block Assignments		Channel Assignments	
	RCERx	**Block Assigned**	**Bit in RCERx**	**Channel Assigned**
	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

12.10 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 12−11. There is one for each of the transmit partitions: A, B, C, D, E, F, G, and H. Table 12−14 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 12−11. Format of the 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 12−14. Description For Bit x of a Transmit Channel Enable Register (x = 0, 1, 2, ..., or 15)

Bit	Field	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.

12.10.1 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 12−15. The table shows which block of channels is assigned to each XCER that is used. For each XCER, the table shows which channels 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 12−15. 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
	XCERD	Block 3	XCE0 XCE1 XCE2 : XCE15	Channel 48 Channel 49 Channel 50 : Channel 63

Table 12−15. 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**
	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

Chapter 13

McBSP Register Worksheet

This register worksheet is meant to be printed and used as a guide for configuring the McBSP registers. Each figure on the worksheet provides space in every register field for entering the binary value that will need to be loaded into that field. For read-only fields, you can use 0s or 1s. When all of the fields have been filled in, you can use the line above the register figure to record the corresponding hexadecimal value to load into the register during initialization.

Topic **Page**

13.1 General Control Registers

SPCR1 − Initialization Value: ____

15	14−13	12−11	10−8
DLB	RJUST	CLKSTP	Reserved
			Read-only

7	6	5−4	3	2	1	0
DXENA	Reserved	RINTM	RSYNCERR	RFULL	RRDY	RRST
				Read-only	Read-only	

SPCR2 − Initialization Value: ____

15−10	9	8
Reserved	FREE	SOFT
Read-only		

7	6	5−4	3	2	1	0
FRST	GRST	XINTM	XSYNCERR	XEMPTY	XRDY	XRST
				Read-only	Read-only	

PCR − Initialization Value: ____

15	14	13	12	11	10	9	8
Reserved	IDLEEN†	XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM
Read-only							

†Reserved on 5501 and 5502

7	6	5	4	3	2	1	0
SCLKME	CLKSSTAT	DXSTAT	DRSTAT	FSXP	FSRP	CLKXP	CLKRP
	Read-only		Read-only				

RCR1 − Initialization Value: ____

15	14−8	7−5	4−0
Reserved	RFRLEN1	RWDLEN1	Reserved
Read-only			Read-only

RCR2 − Initialization Value: ____

15	14−8	7−5	4−3	2	1−0
RPHASE	RFRLEN2	RWDLEN2	RCOMPAND	RFIG	RDATDLY

XCR1 − Initialization Value: ____

15	14−8	7−5	4−0
Rsvd	XFRLEN1	XWDLEN1	Reserved
Read-only			Read-only

XCR2 − Initialization Value: ____

15	14−8	7−5	4−3	2	1−0
XPHASE	XFRLEN2	XWDLEN2	XCOMPAND	XFIG	XDATDLY

SRGR1 − Initialization Value: ____

15−8	7−0
FWID	CLKGDV

SRGR2 − Initialization Value: ____

15	14	13	12	11−0
GSYNC	CLKSP	CLKSM	FSGM	FPER

13.2 Multichannel Selection Control Registers

MCR1 − Initialization Value: ____

15−10	9	8−7	6−5	4−2	1	0
Reserved	RMCME	RPBBLK	RPABLK	RCBLK	Reserved	RMCM
Read-only				Read-only	Read-only	

MCR2 − Initialization Value: ____

15−10	9	8−7	6−5	4−2	1−0
Reserved	XMCME	XPBBLK	XPABLK	XCBLK	XMCM
Read-only				Read-only	

RCERA − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________

RCERB − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________	Channel ________

RCERC − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel 47	Channel 46	Channel 45	Channel 44	Channel 43	Channel 42	Channel 41	Channel 40

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel 39	Channel 38	Channel 37	Channel 36	Channel 35	Channel 34	Channel 33	Channel 32

RCERD − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel 63	Channel 62	Channel 61	Channel 60	Channel 59	Channel 58	Channel 57	Channel 56

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel 55	Channel 54	Channel 53	Channel 52	Channel 51	Channel 50	Channel 49	Channel 48

RCERE − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel 79	Channel 78	Channel 77	Channel 76	Channel 75	Channel 74	Channel 73	Channel 72

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel 71	Channel 70	Channel 69	Channel 68	Channel 67	Channel 66	Channel 65	Channel 64

RCERF − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel 95	Channel 94	Channel 93	Channel 92	Channel 91	Channel 90	Channel 89	Channel 88

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel 87	Channel 86	Channel 85	Channel 84	Channel 83	Channel 82	Channel 81	Channel 80

RCERG − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel 111	Channel 110	Channel 109	Channel 108	Channel 107	Channel 106	Channel 105	Channel 104

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel 103	Channel 102	Channel 101	Channel 100	Channel 99	Channel 98	Channel 97	Channel 96

RCERH − Initialization Value: ____

15	14	13	12	11	10	9	8
RCE15	RCE14	RCE13	RCE12	RCE11	RCE10	RCE9	RCE8
Channel 127	Channel 126	Channel 125	Channel 124	Channel 123	Channel 122	Channel 121	Channel 120

7	6	5	4	3	2	1	0
RCE7	RCE6	RCE5	RCE4	RCE3	RCE2	RCE1	RCE0
Channel 119	Channel 118	Channel 117	Channel 116	Channel 115	Channel 114	Channel 113	Channel 112

XCERA − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel __________	Channel __________	Channel __________	Channel __________	Channel __________	Channel __________	Channel __________	Channel __________

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel __________	Channel __________	Channel __________	Channel __________	Channel __________	Channel __________	Channel __________	Channel __________

XCERB − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel _________	Channel _________	Channel _________	Channel _________	Channel _________	Channel _________	Channel _________	Channel _________

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel _________	Channel _________	Channel _________	Channel _________	Channel _________	Channel _________	Channel _________	Channel _________

XCERC − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel 47	Channel 46	Channel 45	Channel 44	Channel 43	Channel 42	Channel 41	Channel 40

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel 39	Channel 38	Channel 37	Channel 36	Channel 35	Channel 34	Channel 33	Channel 32

XCERD − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel 63	Channel 62	Channel 61	Channel 60	Channel 59	Channel 58	Channel 57	Channel 56

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel 55	Channel 54	Channel 53	Channel 52	Channel 51	Channel 50	Channel 49	Channel 48

XCERE − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel 79	Channel 78	Channel 77	Channel 76	Channel 75	Channel 74	Channel 73	Channel 72

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel 71	Channel 70	Channel 69	Channel 68	Channel 67	Channel 66	Channel 65	Channel 64

XCERF − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel 95	Channel 94	Channel 93	Channel 92	Channel 91	Channel 90	Channel 89	Channel 88

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel 87	Channel 86	Channel 85	Channel 84	Channel 83	Channel 82	Channel 81	Channel 80

XCERG − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel 111	Channel 110	Channel 109	Channel 108	Channel 107	Channel 106	Channel 105	Channel 104

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel 103	Channel 102	Channel 101	Channel 100	Channel 99	Channel 98	Channel 97	Channel 96

XCERH − Initialization Value: ____

15	14	13	12	11	10	9	8
XCE15	XCE14	XCE13	XCE12	XCE11	XCE10	XCE9	XCE8
Channel 127	Channel 126	Channel 125	Channel 124	Channel 123	Channel 122	Channel 121	Channel 120

7	6	5	4	3	2	1	0
XCE7	XCE6	XCE5	XCE4	XCE3	XCE2	XCE1	XCE0
Channel 119	Channel 118	Channel 117	Channel 116	Channel 115	Channel 114	Channel 113	Channel 112

Appendix A

Revision History

This document was revised to SPRU592A from SPRU592, which was released in December 2002.

The scope of the revisions was limited to adding support for the TMS320VC5501 device.

A.1 Changes Made in This Revision

The following changes were made in this revision:

Page	Additions/Modifications/Deletions
Global	Added the TMS320VC5501 device.

Index