## TMS320x281x DSP Multichannel Buffered Serial Port (McBSP) Reference Guide

Literature Number: SPRU061D May 2003 - Revised August 2011



#### Preface

### **Read This First**

#### About This Manual

This guide describes how Multichannel Buffered Serial Port (McBSP) works in the TMS320x281x devices.

#### **Related Documentation From Texas Instruments**

The following books describe the TMS320x281x and related support tools that are available on the TI website.

- TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, and TMS320C2812 Digital Signal Processors (literature number SPRS174) data sheet contains the electrical and timing specifications for these devices, as well as signal descriptions and pinouts for all of the available packages.
- **TMS320R2811 and TMS320R2812 Digital Signal Processors** (literature number SPRS257) data sheet contains the electrical and timing specifications for these devices, as well as signal descriptions and pinouts for all of the available packages.
- **TMS320C28x DSP CPU and Instruction Set Reference Guide** (literature number SPRU430) describes the central processing unit (CPU) and the assembly language instructions of the TMS320C28x<sup>™</sup> fixed-point digital signal processors (DSPs). It also describes emulation features available on these DSPs.
- **TMS320x281x Analog-to-Digital Converter (ADC) Reference Guide** (literature number SPRU060) describes the ADC module. The module is a 12-bit pipelined ADC. The analog circuits of this converter, referred to as the core in this document, include the front-end analog multiplexers (MUXs), sample-and-hold (S/H) circuits, the conversion core, voltage regulators, and other analog supporting circuits. Digital circuits, referred to as the wrapper in this document, include programmable conversion sequencer, result registers, interface to analog circuits, interface to device peripheral bus, and interface to other on-chip modules.
- TMS320x281x Boot ROM Reference Guide (literature number SPRU095) describes the purpose and features of the bootloader (factory-pro-

grammed boot-loading software). It also describes other contents of the device on-chip boot ROM and identifies where all of the information is located within that memory.

- TMS320x281x, 280x Enhanced Controller Area Network (eCAN) Reference Guide (literature number SPRU074) describes the eCAN that uses established protocol to communicate serially with other controllers in electrically noisy environments. With 32 fully configurable mailboxes and time-stamping feature, the eCAN module provides a versatile and robust serial communication interface. The eCAN module implemented in the C28x DSP is compatible with the CAN 2.0B standard (active).
- **TMS320x281x Event Manager (EV) Reference Guide** (literature number SPRU065) describes the EV modules that provide a broad range of functions and features that are particularly useful in motion control and motor control applications. The EV modules include general-purpose (GP) timers, full-compare/PWM units, capture units, and quadrature-encoder pulse (QEP) circuits.
- **TMS320x281x External Interface (XINTF) Reference Guide** (literature number SPRU067) describes the external interface (XINTF) of the 28x digital signal processors (DSPs).
- *TMS320x28xx, 28xxx Peripheral Reference Guide* (literature number SPRU566) describes the peripheral reference guides of the 28x digital signal processors (DSPs).
- TMS320x28xx, 28xxx Serial Communication Interface (SCI) Reference Guide (literature number SPRU051) describes the SCI that is a two-wire asynchronous serial port, commonly known as a UART. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format.
- TMS320x28xx, 28xxx Serial Peripheral Interface (SPI) Reference Guide (literature number SPRU059) describes the SPI – a high-speed synchronous serial input/output (I/O) port that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is used for communications between the DSP controller and external peripherals or another controller.
- **TMS320x281x System Control and Interrupts Reference Guide** (literature number SPRU078) describes the various interrupts and system control features of the 281x digital signal processors (DSPs).

- *The TMS320C28x Instruction Set Simulator Technical Overview* (literature number SPRU608) describes the simulator, available within the Code Composer Studio for TMS320C2000 IDE, that simulates the instruction set of the C28x core.
- *TMS320C28x DSP/BIOS Application Programming Interface (API) Reference Guide* (literature number SPRU625) describes development using DSP/BIOS.

#### Trademarks

Code Composer Studio and C28x are trademarks of Texas Instruments.

v

## Contents

1	Over	view	
	Introa	luces the	e multichannel buffered serial port (McBSP).
	1.1	Introdu	ction to the McBSP
		1.1.1	Key Features of the McBSP 1-2
		1.1.2	Block Diagram of the McBSP Module With FIFO 1-4
		1.1.3	McBSP Signals 1-5
	1.2	Registe	er Summary 1-7
	1.3	McBSF	P Operation
		1.3.1	Data Transfer Process of a McBSP 1-10
		1.3.2	Companding (Compressing and Expanding) Data 1-11
		1.3.3	Clocking and Framing Data 1-14
		1.3.4	Frame Phases
		1.3.5	McBSP Reception
		1.3.6	McBSP Transmission 1-22
		1.3.7	Interrupts and FIFO Events Generated by a McBSP 1-24
	1.4	Sample	e Rate Generator of the McBSP 1-26
		1.4.1	Clock Generation in the Sample Rate Generator 1-27
		1.4.2	Frame Sync Generation in the Sample Rate Generator 1-30
		1.4.3	Synchronizing Sample Rate Generator Outputs to an External Clock 1-31
		1.4.4	Reset and Initialization Procedure for the Sample Rate Generator 1-33
		1.4.5	Sample Rate Generator Clocking Examples 1-35
	1.5	McBSF	P Exception/Error Conditions 1-38
		1.5.1	Overrun in the Receiver 1-39
		1.5.2	Unexpected Receive Frame-Sync Pulse 1-40
		1.5.3	Overwrite in the Transmitter 1-43
		1.5.4	Underflow in the Transmitter 1-44
		1.5.5	Unexpected Transmit Frame-Sync Pulse 1-46
2	Multi	channel	Selection Modes
	Desci	ribes the	process for using multichannel selection.
	2.1	Channe	els, Blocks, and Partitions
		2.1.1	Multichannel Selection
		2.1.2	Configuring a Frame for Multichannel Selection 2-4
		2.1.3	Using Two Partitions 2-4
		2.1.4	Using Eight Partitions 2-6

		2.1.5	Receive Multichannel Selection Mode	. 2-8
		2.1.6	Transmit Multichannel Selection Modes	. 2-9
		2.1.7	Disabling/Enabling Versus Masking/Unmasking	2-10
		2.1.8	Activity on McBSP Pins for Different Values of XMCM	2-11
		2.1.9	Using Interrupts Between Block Transfers	2-13
	2.2	A-bis M	Node	2-14
		2.2.1	A-bis Mode Receive Operation	2-14
		2.2.2	A-bis Mode Transmit Operation	2-15
	2.3	SPI Pro		2-16
		2.3.1	Clock Stop Mode	2-16
		2.3.2	Bits Used to Enable and Configure the Clock Stop Mode	2-17
		2.3.3	Clock Stop Mode Timing Diagrams	2-18
		2.3.4	Procedure for Configuring a McBSP for SPI Operation	2-20
		2.3.5	McBSP as the SPI Master	2-21
		2.3.6	McBSP as an SPI Slave	2-23
3	Confi	gure th	e Receiver and Transmitter	. 3-1
	Desci	ribes hov	w to configure the receiver and transmitter.	
	3.1	Receiv	er Configuration	. 3-2
		3.1.1	Programming the McBSP Registers for the Desired Receiver Operation	. 3-2
		3.1.2	Resetting and Enabling the Receiver	. 3-3
		3.1.3	Set the Receiver Pins to Operate as McBSP Pins	. 3-4
		3.1.4	Enable/Disable the Digital Loopback Mode	. 3-4
		3.1.5	Digital Loopback Mode	. 3-4
		3.1.6	Clock Stop Mode	. 3-5
		3.1.7	Receive Multichannel Selection and A-bis Modes	. 3-6
		3.1.8	Choose 1 or 2 Phases for the Receive Frame	. 3-7
		3.1.9	Set the Receive Companding Mode	3-10
		3.1.10	Set the Receive Data Delay	3-11
		3.1.11	Set the Receive Sign-Extension and Justification Mode	3-13
		3.1.12	Set the Receive Interrupt Mode	3-14
		3.1.13	Set the Receive Frame-Sync Mode	3-15
		3.1.14	Set the Receive Frame-Sync Polarity	3-16
	3.2	Transm	nitter Configuration	3-27
		3.2.1	Programming the McBSP Registers for the Desired Transmitter Operation .	3-27
		3.2.2	Resetting and Enabling the Transmitter	3-28
		3.2.3	Set the Transmitter Pins to Operate as McBSP Pins	3-29
		3.2.4	Enable/Disable the Digital Loopback Mode	3-29
		3.2.5	Enable/Disable the Clock Stop Mode	3-30
		3.2.6	Enable/Disable Transmit Multichannel Selection	3-31
		3.2.7	Enable/Disable the A-bis Mode	3-32
		3.2.8	Choose 1 or 2 Phases for the Transmit Frame	3-32
		3.2.9	Set the Transmit Word Length(s)	3-32
		3.2.10	Set the Transmit Frame Length	3-33

		3.2.11 Set the Transmit Companding Mode 3-3	5
		3.2.12 Set the Transmit Data Delay 3-3	7
		3.2.13 Set the Transmit DXENA Mode 3-3	8
		3.2.14 Set the Transmit Interrupt Mode 3-38	9
		3.2.15 Set the Transmit Frame-Sync Mode 3-4	0
		3.2.16 Set the Transmit Frame-Sync Polarity 3-4	1
		3.2.17 Set the SRG Frame-Sync Period and Pulse Width 3-4	3
		3.2.18 Set the Transmit Clock Mode 3-4	4
		3.2.19 Set the SRG Clock Divide-Down Value 3-4	7
		3.2.20 Set the SRG Clock Mode (Choose an Input Clock) 3-4	8
4	Emu	ition and Reset Considerations 4-	1
	Desc	bes resetting and initializing the McBSP.	
	4.1	McBSP Emulation Mode 4-	2
		4.1.1 Resetting and Initializing a McBSP 4-	3
	4.2	Data Packing Examples 4-	8
		4.2.1 Data Packing Using Frame Length and Word Length 4-	8
		4.2.2 Data Packing Using Word Length and the Frame-Sync Ignore Function 4-	9
	4.3	GPIO Function	1
5	McB	P FIFO and Interrupts	1
	Desc	bes the FIFO interface logic.	
	5.1	McBSP FIFO Overview	2
	5.2	McBSP Functionality and Limitation Under FIFO Mode	4
	5.3	McBSP FIFO Operation	6
	5.4	McBSP Receive Interrupt Generation	8
	5.5	McBSP Transmit Interrupt Generation 5-1	0
		5.5.1 FIFO Data Register Access Constraints 5-1	1
		5.5.2 FIFO Error Flags 5-1	2
		5.5.3 McBSP IDLE Mode 5-13	3
		5.5.4 McBSP Reset Conditions 5-13	3
	5.6	McBSP FIFO Register Descriptions 5-1	4
6	McB	P Registers	1
	Desc	bes the McBSP registers and bit descriptions.	
	6.1	Data Receive and Transmit Registers	2
		6.1.1 Data Receive Registers (DRR2 and DRR1)	2
		6.1.2 How Data Travels From the Data Receive (DR) Pin to the DRRs 6-	2
		6.1.3 Data Transmit Registers (DXR2 and DXR1)	3
		6.1.4 How Data Travels From the DXRs to the Data Transmit (MDXA) Pin 6-	3
	6.2	Serial Port Control Registers (SPCR1 and SPCR2)	4
	6.3	Receive Control Registers (RCR1 and RCR2)	9
	6.4	Transmit Control Registers (XCR1 and XCR2)	4
	6.5	Sample Rate Generator Registers (SRGR1 and SRGR2) 6-1	8

Contents ix

	6.6	Multich	nannel Control Registers (MCR1 and MCR2)	6-22
	6.7	Pin Co	ntrol Register (PCR)	6-29
	6.8	Receiv	e Channel Enable Registers (RCERA - RCERH)	6-33
		6.8.1	RCERs Used in the Receive Multichannel Selection Mode	6-35
		6.8.2	RCERs Used in the A-bis Mode	6-36
	6.9	Transr	nit Channel Enable Registers (XCERA - XCERH)	6-38
		6.9.1	XCERs Used in a Transmit Multichannel Selection Mode	6-39
		6.9.2	XCERs Used in the A-bis Mode	6-41
	6.10	Regist	er Bit Summary	6-43
Α	Revis	ion His	tory	A-1
	A.1	Chang	es Made in This Revision	A-2

# Figures

1-1.       Block Diagram With FIFO Interface       1-4         1-2.       McBSP Data Transfer Paths       1-10         1-3.       Companding Processes       1-12         1-4.       µ-Law Transmit Data Companding Format       1-12         1-5.       A-Law Transmit Data Companding Format       1-12         1-5.       A-Law Transmit Data Companding Format       1-12         1-6.       Two Methods by Which the McBSP Can Compand Internal Data       1-13         1-7.       Clock Signal Control Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-14.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Reception Signal Activity       1-21         1-16.       McBSP Transmission Signal Activity       1-23
1-2.       McBSP Data Transfer Paths       1-10         1-3.       Companding Processes       1-12         1-4.       μ-Law Transmit Data Companding Format       1-12         1-5.       A-Law Transmit Data Companding Format       1-12         1-6.       Two Methods by Which the McBSP Can Compand Internal Data       1-13         1-7.       Clock Signal Control Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-15         1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-21         1-14.       Timing of an AC97-Standard Data Path       1-21         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Transmission Physical Data Path       1-23         1-17.       McBSP Transmission Signal Activity       1-23
1-3.       Companding Processes       1-12         1-4.       μ-Law Transmit Data Companding Format       1-12         1-5.       A-Law Transmit Data Companding Format       1-12         1-6.       Two Methods by Which the McBSP Can Compand Internal Data       1-13         1-7.       Clock Signal Control Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-15         1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and th
1-4.       μ-Law Transmit Data Companding Format       1-12         1-5.       A-Law Transmit Data Companding Format       1-12         1-6.       Two Methods by Which the McBSP Can Compand Internal Data       1-13         1-7.       Clock Signal Control Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-15         1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.
1-5.       A-Law Transmit Data Companding Format       1-12         1-6.       Two Methods by Which the McBSP Can Compand Internal Data       1-13         1-7.       Clock Signal Control Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-15         1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Transmission Physical Data Path       1-23         1-17.       McBSP Transmission Signal Activity       1-23         1-18.       McBSP Transmission Signal Activity       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-6.       Two Methods by Which the McBSP Can Compand Internal Data       1-13         1-7.       Clock Signal Control Waveform       1-14         1-8.       8-Bit Word Size Defined Waveform       1-15         1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-7.Clock Signal Control Waveform1-141-8.8-Bit Word Size Defined Waveform1-151-9.One-word Frame Transfer1-161-10.McBSP Operating at Maximum Packet Frequency1-171-11.Single-Phase Frame for a McBSP Data Transfer1-191-12.Dual-Phase Frame for a McBSP Data Transfer1-191-13.Implementing the AC97 Standard With a Dual-Phase Frame1-201-14.Timing of an AC97-Standard Data Transfer Near Frame Synchronization1-201-15.McBSP Reception Physical Data Path1-211-16.McBSP Reception Signal Activity1-211-17.McBSP Transmission Physical Data Path1-231-18.McBSP Transmission Signal Activity1-231-19.Sample Rate Generator Clock Selection1-261-20.Possible Inputs to the Sample Rate Generator and the Polarity Bits1-291-21.CLKG Synchronization and FSG Generation When GSYNC = 11-331-22.CLKG Synchronization and FSG Generation When GSYNC = 11-33
1-8.       8-Bit Word Size Defined Waveform       1-15         1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-9.       One-word Frame Transfer       1-16         1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-10.       McBSP Operating at Maximum Packet Frequency       1-17         1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-11.       Single-Phase Frame for a McBSP Data Transfer       1-19         1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-12.       Dual-Phase Frame for a McBSP Data Transfer       1-19         1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-13.       Implementing the AC97 Standard With a Dual-Phase Frame       1-20         1-14.       Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-14. Timing of an AC97-Standard Data Transfer Near Frame Synchronization       1-20         1-15. McBSP Reception Physical Data Path       1-21         1-16. McBSP Reception Signal Activity       1-21         1-17. McBSP Transmission Physical Data Path       1-23         1-18. McBSP Transmission Signal Activity       1-23         1-19. Sample Rate Generator Clock Selection       1-26         1-20. Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21. CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22. CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22. CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-15.       McBSP Reception Physical Data Path       1-21         1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-16.       McBSP Reception Signal Activity       1-21         1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-17.       McBSP Transmission Physical Data Path       1-23         1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
1-18.       McBSP Transmission Signal Activity       1-23         1-19.       Sample Rate Generator Clock Selection       1-26         1-20.       Possible Inputs to the Sample Rate Generator and the Polarity Bits       1-29         1-21.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-22.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33         1-23.       CLKG Synchronization and FSG Generation When GSYNC = 1       1-33
<ul> <li>1-19. Sample Rate Generator Clock Selection</li></ul>
<ul> <li>1-20. Possible Inputs to the Sample Rate Generator and the Polarity Bits</li></ul>
<ul> <li>1-21. CLKG Synchronization and FSG Generation When GSYNC = 1 and CLKGDV = 1</li></ul>
1-22. CLKG Synchronization and FSG Generation When GSYNC = 1
1-23 ST-BUS and MVIP Clocking Example
1-24 Single-Bate Clock Example 1-36
1-25 Double-Bate Clock Example
1-26 Overrun in the McBSP Receiver 1-40
1-27. Overrun Prevented in the McBSP Receiver
1-28 Possible Responses to Receive Frame-Sync Pulses 1-41
1-29 An Unexpected Frame-Sync Pulse During a McBSP Reception 1-42
1-30 Proper Positioning of Frame-Sync Pulses
1-31 Data in the McBSP Transmitter Overwritten and Thus Not Transmitted 1-44
1-32. Underflow During McBSP Transmission 1-45
1–33. Underflow Prevented in the McBSP Transmitter 1-46
1-34. Possible Responses to Transmit Frame-Sync Pulses
1–35. An Unexpected Frame-Sync Pulse During a McBSP Transmission

Contents

xi

Figures	

1-36.	Proper Positioning of Frame-Sync Pulses	1-49
2-1.	Alternating Between the Channels of Partition A and the	
	Channels of Partition B	. 2-5
2-2.	Reassigning Channel Blocks Throughout a McBSP Data Transfer	. 2-6
2-3.	McBSP Data Transfer in the 8-Partition Mode	. 2-8
2-4.	Activity on McBSP Pins for the Possible Values of XMCM	2-12
2-5.	A-bis Mode Receive Operation	2-14
2-6.	A-bis Mode Transmit Operation	2-15
2-7.	Typical SPI Interface	2-16
2-8.	SPI Transfer With CLKSTP = 10b (no clock delay), CLKXP = 0, CLKRP = 0	2-19
2-9.	SPI Transfer With CLKSTP = 11b (clock delay), CLKXP = 0, CLKRP = 1	2-19
2-10.	SPI Transfer With CLKSTP = 10b (no clock delay), CLKXP = 1, CLKRP = 0	2-19
2-11.	SPI Transfer With CLKSTP = 11b (clock delay), CLKXP = 1, CLKRP = 1	2-20
2-12.	SPI Interface With McBSP as Master	2-21
2-13.	SPI With McBSP Configured as Slave	2-23
3-1.	Unexpected Frame-Sync Pulse With (R/X)FIG = 0	. 3-9
3-2.	Unexpected Frame-Sync Pulse With (R/X)FIG = 1	3-10
3-3.	Companding Processes for Reception and for Transmission	3-10
3-4.	Range of Programmable Data Delay	3-12
3-5.	2-Bit Data Delay Used to Skip a Framing Bit	3-13
3-6.	Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge	3-18
3-7.	Frame of Period 16 CLKG Periods and Active Width of 2 CLKG Periods	3-20
3-8.	Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge	3-23
3-9	Unexpected Frame-Sync Pulse With (B/X)FIG = 0	3-34
3-10	Unexpected Frame-Sync Pulse With (R/X)FIG = 1	3-35
3-11	Companding Processes for Reception and for Transmission	3-35
3-12.	m-Law Transmit Data Companding Format	3-36
3-13.	A-Law Transmit Data Companding Format	3-36
3-14.	Range of Programmable Data Delay	3-37
3-15.	2-Bit Data Delay Used to Skip a Framing Bit	3-38
3-16.	MDXA Delay When A-bis Mode is Off	3-39
3-17.	MDXA Delays When A-bis Mode is On	3-39
3-18.	Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge	3-43
3-19.	Frame of Period 16 CLKG Periods and Active Width of 2 CLKG Periods	3-44
3-20.	Data Clocked Externally Using a Rising Edge and	• • •
0 201	Sampled by the McBSP Receiver on a Falling Edge	3-47
4-1.	Four 8-Bit Data Words Transferred To/From the McBSP	. 4-8
4-2.	One 32-Bit Data Word Transferred To/From the McBSP	. 4-9
4-3.	8-Bit Data Words Transferred at Maximum Packet Frequency	4-10
4-4.	Configuring the Data Stream of Figure 4-3 as a Continuous 32-Bit Word	4-10
5-1.	Receive Interrupt Generation	. 5-8
5-2.	Transmit Interrupt Generation	5-10

5-3.	McBSP FIFO Transmit Register (MFFTX)	5-14
5-4.	McBSP FIFO Receive Register (MFFRX)	5-15
5-5.	McBSP FIFO Control Register (MFFCT)	5-16
5-6.	McBSP FIFO Interrupt Register (MFFINT)	5-17
5-7.	McBSP FIFO Status Register (MFFST)	5-18
6-1.	Data Receive Registers (DRR2 and DRR1)	. 6-2
6-2.	Data Transmit Registers (DXR2 and DXR1)	. 6-3
6-3.	SPCR1 Register	. 6-4
6-4.	Serial Port Control 2 Register (SPCR2)	. 6-7
6-5.	RCR1 Register	. 6-9
6-6.	Receive Control 2 Register (RCR2)	6-10
6-7.	Transmit Control 2 Register (XCR2)	6-14
6-8.	XCR1 Register	6-17
6-9.	Sample Rate Generator 2 Register (SRGR2)	6-19
6-10.	Sample Rate Generator 1 Register (SRGR1)	6-21
6-11.	Multichannel Control 1 Register (MCR1)	6-22
6-12.	Multichannel Control 2 Register (MCR2)	6-25
6-13.	Pin Control Register (PCR)	6-29
6-14.	Receive Channel Enable Register (RCERA/B)	6-33
6-15.	RCER(A-G)-Receive Channel Enable Registers – A, C, E, G (for 8-partition	
	mode)	6-34
6-16.	RCER(B-H)-Receive Channel Enable Registers – B,D,F,H (for 8-partition	634
6 17	Transmit Channel English Registers A. C. E. G. (VCERA, VCERG)	6 20
0-17.	Transmit Channel Enable Degisters P. D. E. U. (XCEPP, XCEPU)	6 20
0-10.		0-39

## **Tables**

1-1.	28x Implementation Changes 1-3
1-2.	McBSP Signal Summary 1-5
1-3.	McBSP Register Summary 1-7
1-4.	McBSP Register Bits That Determine the Number of Phases, Words, and Bits Per Frame
1-5.	Interrupts and FIFO Events Generated by a McBSP
1-6.	Effects of DLB and CLKSTP on Clock Modes 1-28
1-7.	Choosing an Input Clock for the Sample Rate Generator With the SCI KME and CI KSM Bits 1-28
1-8	Polarity Options for the Input to the Sample Bate Generator 1-29
2-1	Block - Channel Assignment
2-2.	2-Partition Mode
2-3.	8-Partition Mode
2-4.	Receive Channel Assignment and Control When Eight Receive Partitions Are Used 2-7
2-5.	Transmit Channel Assignment and Control When Eight Transmit Partitions Are Used
2-6.	Selecting a Transmit Multichannel Selection Mode With the XMCM Bits 2-9
2-7.	Bits Used to Enable and Configure the Clock Stop Mode
2-8.	Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme
2-9.	Bit Values Required to Configure the McBSP as an SPI Master
2-10.	Bit Values Required to Configure the McBSP as an SPI Slave
3-1.	Reset State of Each McBSP Pin 3-4
3-2.	Register Bit Used to Enable/Disable the Digital Loopback Mode
3-3.	Receive Signals Connected to Transmit Signals in Digital Loopback Mode 3-5
3-4.	Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme
3-5.	How to Calculate the Length of the Receive Frame 3-8
3-6.	Example: Use of RJUST Field With 12-Bit Data Value 0xABC 3-13
3-7.	Example: Use of RJUST Field With 20-Bit Data Value 0xABCDE 3-14
3-8.	Select Sources to Provide the Receive Frame-Synchronization Signal and the Effect on the MFSRA Pin
3-9.	Select Sources to Provide the Receive Clock Signal and the Effect on the MCLKRA Pin
3-10.	Reset State of Each McBSP Pin 3-29
3-11.	Receive Signals Connected to Transmit Signals in Digital Loopback Mode 3-29
3-12.	Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme
3-13.	How to Calculate Frame Length 3-33
3-14.	How FSXM and FSGM Select the Source of Transmit Frame-Sync Pulses 3-40

3-15.	How the CLKXM Bit Selects the Transmit Clock and the Corresponding Status of the MCLKXA Pin	3-45
4-1.	McBSP Emulation Modes Selectable With the FREE and SOFT Bits of SPCR2	. 4-3
4-2.	Reset State of Each McBSP Pin	. 4-3
5-1.	McBSP FIFO Registers	. 5-2
5-2.	McBSP Mode Selection	. 5-5
5-3.	Receive Interrupt Sources and Signals in NonFIFO Mode and FIFO Mode	. 5-9
5-4.	Transmit Interrupt Sources and Signals in NonFIFO Mode and FIFO Mode	5-11
5-5.	Receive FIFO Read Order	5-12
5-6.	Transmit FIFO Write Order	5-12
5-7.	McBSP Error Flags	5-13
6-1.	SPCR1 Register Bit Descriptions	. 6-4
6-2.	Serial Port Control 2 Register (SPCR2) Bit Descriptions	. 6-7
6-3.	RCR1 Register Field Descriptions	. 6-9
6-4.	Frame Length Formula for Receive Control 1 Register (RCR1)	6-10
6-5.	Receive Control 2 Register (RCR2) Bit Descriptions	6-11
6-6.	Frame Length Formula for Receive Control 2 Register (RCR2)	6-13
6-7.	Transmit Control 2 Register (XCR2) Field Descriptions	6-14
6-8.	Frame Length Formula for Transmit Control 2 Register (XCR2)	6-16
6-9.	XCR1 Register Field Descriptions	6-17
6-10.	Sample Rate Generator 2 Register (SRGR2)Field Descriptions	6-19
6-11.	Sample Rate Generator 1 Register (SRGR1)	6-21
6-12.	Multichannel Control 1 Register (MCR1) Field Descriptions	6-22
6-13.	Multichannel Control 2 Register (MCR2) Field Descriptions	6-25
6-14.	Pin Control Register (PCR) Field Descriptions	6-29
6-15.	Pin Configuration	6-32
6-16.	Use of the Receive Channel Enable Registers in theReceive Multichannel Selection Mode	6-35
6-17.	Use of Receive Channel Enable Registers A and B in the A-bis Mode	6-37
6-18.	Use of the Transmit Channel Enable Registers in a Transmit Multichannel	
	Selection Mode	6-40
6-19.	Use of Transmit Channel Enable Registers A and B in the A-bis Mode	6-42
6-20.	Register Bit Summary (Base Address 0x00 7800)	6-43
6-21.	FIFO Register Bit Descriptions (Base address 0x00 7800)	6-46

### Chapter 1

## **Overview**

The multichannel buffered serial port (McBSP) provides a direct serial interface between a digital signal processor (DSP) and other devices in a system.

This reference guide is applicable for the McBSP found on the TMS320x281x family of processors. This includes all Flash-based, ROM-based, and RAM-based devices within the 281x family.

If you have used a McBSP in other TI DSPs, note that there are differences in this implementation, which are outlined in Table 1–1.

#### Topic

#### Page

1.1	Introduction to the McBSP 1-2
1.2	Register Summary 1-7
1.3	McBSP Operation 1-9
1.4	Sample Rate Generator of the McBSP 1-25
1.5	McBSP Exception/Error Conditions 1-38

#### 1.1 Introduction to the McBSP

The McBSP peripheral provides an interface between a 28x device and McBSP-compatible devices such as the VBAP, AIC, Combo Codecs. The external components attached to the McBSP can synchronously transmit/receive 8/16/32-bit serial data.

#### 1.1.1 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
- 128 channels for transmission and for reception
- Multichannel selection modes that enable you to allow or block transfers in each of the channels
- DMA replaced with two 16-level, 32-bit FIFOs
- Support for A-bis mode
- 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 frame-sync signals
- Programmable internal clock and frame generation
- Programmable polarity for frame-synchronization and data clocks
- Support for SPI devices

- Support fractional T1/E1. 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.
- D The option of transmitting/receiving 8-bit data with the LSB or MSB first

The McBSP module in C28x devices is adapted from TI's family of McBSPs. Although it supports most of the McBSP applications, Table 1–1 lists some limitations to this implementation.

Features of TI's McBSP family	Implementation in 28x McBSP Module
DMA supports for data transfer	DMA replaced by two 32 X 16 level FIFOs
GPIO function on McBSP pins	Supported through the GPIO module available in 28x family.
Power down-mode using IDLE-EN bit in PCR register	The clock for the McBSP logic is controlled using bit 12 of the peripheral clock control register (PCLKCR).
CLKS as external shift clock	CLKS feature is not supported. MCLKRA/MCLKXA pin provide this feature.

Table 1-1. 28x Implementation Changes

#### 1.1.2 Block Diagram of the McBSP Module With FIFO

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

Figure 1-1. Block Diagram With FIFO Interface



Data is communicated to devices interfaced with the McBSP via the data transmit (MDXA) pin for transmission and the data receive (MDRA) pin for reception. Control information in the form of clocking and frame synchronization is communicated via the following pins: MCLKXA (transmit clock), MCLKRA (receive clock), MFSXA (transmit frame sync), and MFSRA (receive frame sync).

MDRAR2, 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 are registers for controlling McBSP operation. Details about these registers are available in section 1.2 (page 1-7).

#### 1.1.3 McBSP Signals

Table 1–2 describes the McBSP interface signals. Information on using these signals for general-purpose input/output is in section 4.3 on page 4-11.

Signal Name	Type †	Reset Status	Description	
External Signals			External Signals Names	
MCLKXA	I/O/Z	Input	Transmit clock	
MCLKRA	I/O/Z	Input	Receive clock	
MDRA	I	Input	Received serial data	
MDXA	O/Z	Input	Transmitted serial data	
MFSRA	I/O/Z	Input	Receive frame synchronization	
MFSXA	I/O/Z	Input	Transmit frame synchronization	
CPU-Interrupt Signals				
MRINT			Receive interrupt to CPU or FIFO	
MXINT			Transmit interrupt to CPU or FIFO	
FIFO Events				
REVT			Receive synchronization event to FIFO	

Table 1-2. McBSP Signal Summary

<sup>†</sup> I = input, O = output, Z = high impedance

Notes: 1) CLKS signal is not supported in this implementation.

2) All McBSP pins are GPIO input upon reset.

Signal Name	Type †	Reset Status	Description
XEVT			Transmit synchronization event to FIFO
REVTA			Receive synchronization in A-bis mode in the FIFO
XEVTA			Transmit synchronization in A-bis mode in the FIFO

Table 1-2. McBSP Signal Summary (Continued)

<sup>†</sup> I = input, O = output, Z = high impedance

Notes: 1) CLKS signal is not supported in this implementation.

2) All McBSP pins are GPIO input upon reset.

#### 1.2 Register Summary

For each McBSP, the DSP contains the following registers. For the address of each register, see the data sheet for your device.

Address Offset					
0x000 xxh	16-Bit Access	Type R/W	Reset ValueHex	McBSP Communications Interface Register Description	
			Data Re	gisters, Receive, Transmit	
-	-	-		McBSP Receive Buffer Register	
-	-	-		McBSP Receive Shift Register	
-	-	-		McBSP Transmit Shift Register	
00	DRR2	R	0x0000	McBSP Data Receive Register2	
				Read First if the word size is greater than 16 bit else ignore DRR2	
01	DRR1	R	0x0000	McBSP Data Receive Register1	
				Read second if the word size is greater than 16 bit else read DRR1 only	
02	DXR2	W	0x0000	McBSP Data Transmit Register2	
				Write First if the word size is greater than 16 bit else ignore DXR2	
03	DXR1	W	0x0000	McBSP Data Transmit Register1	
				Write second if the word size is greater than 16 bit else Write to DXR1 only	
McBSP Control Registers					
04	SPCR2	R/W	0x0000	McBSP Serial Port Control Register2	
05	SPCR1	R/W	0x0000	McBSP Serial Port Control Register1	
06	RCR2	R/W	0x0000	McBSP Receive Control Register2	
07	RCR1	R/W	0x0000	McBSP Receive Control Register1	
08	XCR2	R/W	0x0000	McBSP Transmit Control Register2	
09	XCR1	R/W	0x0000	McBSP Transmit Control Register1	

Table 1-3. McBSP Register Summary

<sup>†</sup> MFFST will read 0x000A if MFSXA/MFSRA pins are left unconnected; else, it will assume pin status at reset.

Address Offset				
0x000 xxh	16-Bit Access	Type R/W	Reset ValueHex	McBSP Communications Interface Register Description
		I	Multichanne	Control Registers (Continued)
0A	SRGR2	R/W	0x2000	McBSP Sample Rate Generator Register2
0B	SRGR1	R/W	0x0001	McBSP Sample Rate Generator Register1
0C	MCR2	R/W	0x0000	McBSP Multi-Channel Register2
0D	MCR1	R/W	0x0000	McBSP Multi-Channel Register1
0E	RCERA	R/W	0x0000	McBSP Receive Channel Enable Register Partition A
0F	RCERB	R/W	0x0000	McBSP Receive Channel Enable Register Partition B
10	XCERA	R/W	0x0000	McBSP Transmit Channel Enable Register Partition A
11	XCERB	R/W	0x0000	McBSP Transmit Channel Enable Register Partition A
12	PCR1	R/W	0x0000	McBSP Pin Control Register
13	RCERC	R/W	0x0000	McBSP Receive Channel Enable Register Partition C
14	RCERD	R/W	0x0000	McBSP Receive Channel Enable Register Partition D
15	XCERC	R/W	0x0000	McBSP Transmit Channel Enable Register Partition C
16	XCERD	R/W	0x0000	McBSP Transmit Channel Enable Register Partition D
17	RCERE	R/W	0x0000	McBSP Receive Channel Enable Register Partition E
18	RCERF	R/W	0x0000	McBSP Receive Channel Enable Register Partition F
19	XCERE	R/W	0x0000	McBSP Transmit Channel Enable Register Partition E
1A	XCERF	R/W	0x0000	McBSP Transmit Channel Enable Register Partition F
1B	RCERG	R/W	0x0000	McBSP Receive Channel Enable Register Partition G
1C	RCERH	R/W	0x0000	McBSP Receive Channel Enable Register Partition H
1D	XCERG	R/W	0x0000	McBSP Transmit Channel Enable Register Partition G
1E	XCERH	R/W	0x0000	McBSP Transmit Channel Enable Register Partition H

Table 1-3. McBSP Register Summary (Continued)

<sup>†</sup> MFFST will read 0x000A if MFSXA/MFSRA pins are left unconnected; else, it will assume pin status at reset.

Address Offset						
0x000 xxh	16-Bit Access	Type R/W	Reset ValueHex	McBSP Communications Interface Register Description		
	FIFO Mode Data Registers Applicable Only in FIFO Mode					
00	DRR2	R	0x0000	McBSP Data Receive Register2 - Top of receive FIFO		
				Read First FIFO pointers will not advance		
01	DRR1	R	0x0000	McBSP Data Receive Register1-Top of receive FIFO		
				Read second for FIFO pointers to advance		
02	DXR2	W	0x0000	McBSP Data Transmit Register2-Top of transmit FIFO		
				Write First FIFO pointers will not advance		
03	DXR1	W	0x0000	McBSP Data Transmit Register1-Top of transmit FIFO		
				Write second for FIFO pointers to advance		
FIFO Control Registers						
20	MFFTX	R/W	0x2000	McBSP FIFO Transmit register		
21	MFFRX	R/W	0x201F	McBSP FIFO Receive register		
22	MFFCT	R/W	0x0000	McBSP FIFO Control register		
23	MFFINT	R/W	0x0000	McBSP FIFO Interrupt register		
24	MFFST	R/W	0x000x†	McBSP FIFO Status register		

Table 1-3. McBSP Register Summary (Continued)

<sup>†</sup> MFFST will read 0x000A if MFSXA/MFSRA pins are left unconnected; else, it will assume pin status at reset.

The registers bits are summarized in Table 6-20.

#### 1.3 McBSP Operation

This section contains the following topics:

Торіс	See
Data transfer process of a McBSP	Section 1.3.1
Companding (compressing and expanding) data	Section 1.3.2, page 1-11
Clocking and framing data	Section 1.3.3, page 1-14
Frame phases	Section 1.3.4, page 1-18
McBSP Reception	Section 1.3.5, page 1-21
McBSP Transmission	Section 1.3.6, page 1-22
Interrupts and FIFO events generated by a McBSP	Section 1.3.7, page 1-24

#### 1.3.1 Data Transfer Process of a McBSP

Figure 1–2 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 1-2. McBSP Data Transfer Paths



#### 1.3.1.1 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 MDRA 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. 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. For more details about reception, see section 1.3.5 on page 1-21.

Transmit data is written by the CPU 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 MDXA 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 MDXA pin. For more details about transmission, see section 1.3.6 on page 1-22.

#### 1.3.1.2 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 MDRA pin and is shifted into RSR2 first and then into 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. The CPU must read data from DRR2 first and then from DRR1. When DRR1 is read, the next RBR-to-DRR copy occurs. For more details about reception, see section 1.3.5 on page 1-21.

For transmission, the CPU 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 MDXA pin. After transmit frame synchronization, the transmitter begins shifting bits from the XSRs to the MDXA pin. For more details about transmission, see section 1.3.6 on page 1-22.

#### 1.3.2 Companding (Compressing and Expanding) Data

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

A-law and  $\mu$ -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 must be at least 16 bits wide.

The  $\mu$ -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 0, indicating an 8-bit wide serial data stream. If companding is enabled and either of the frame phases does not have an 8-bit word length, companding continues as if the word length is 8 bits.

Figure 1-3 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  $\mu$ -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 1-3. Companding Processes



#### 1.3.2.1 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  $\mu$ -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 1–4.

Figure 1-4. µ-Law Transmit Data Companding Format



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 1–5.

Figure 1-5. A-Law Transmit Data Companding Format



1-12 Multichannel Buffered Serial Port (McBSP)

#### 1.3.2.2 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:

- $\Box$  Convert linear to the appropriate  $\mu$ -law or A-law format.
- $\Box$  Convert  $\mu$ -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 1–6 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.

The advantage of this method is its speed. The disadvantage is that there is no synchronization available to the CPU 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  $\mu$ -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 to these conversions. The time for this companding depends on the serial bit rate selected.

Figure 1-6. Two Methods by Which the McBSP Can Compand Internal Data



SPRU061D

#### 1.3.2.3 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.

#### 1.3.3 Clocking and Framing Data

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

#### 1.3.3.1 Clocking

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

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

In Figure 1–7, the clock signal controls the timing of each bit transfer on the pin.





#### Note:

The McBSP cannot operate at a frequency faster than 1/2 the LSPCLK frequency. When driving MCLKXA or MCLKRA at the pin, choose an appropriate input clock frequency. When using the internal sample rate generator for MCLKXA and/or MCLKRA, choose an appropriate input clock frequency and divide down value (CLKGDV).

1-14 Multichannel Buffered Serial Port (McBSP)

#### 1.3.3.2 Serial Words

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

Bits coming in on the MDRA 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 MDXA pin.

In Figure 1–8, an 8-bit word size was defined (see bits 7 through 0 of word B being transferred).

#### Figure 1-8. 8-Bit Word Size Defined Waveform



#### 1.3.3.3 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 (MFSRA) initiate frame transfers on MDRA. Pulses on the transmit frame-sync signal (MFSXA) initiate frame transfers on MDXA. MFSRA or MFSXA can be derived from a pin at the boundary of the McBSP or derived from inside the McBSP.

In Figure 1–9, 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.

#### 1.3.3.4 Generating Transmit and Receive Interrupts

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).

#### 1.3.3.5 Detecting Frame-Sync Pulses, Even in the Reset State

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.

#### 1.3.3.6 Ignoring 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). For more details on unexpected frame-sync pulses, see one of the following topics:

- Unexpected Receive Frame-Sync Pulse (page 1-40)
- Unexpected Transmit Frame-Sync Pulse (page 1-46)

You can also use the frame-sync ignore function for data packing (for more details, see section 4.2.2 on page 4-9).

#### 1.3.3.7 Frame Frequency

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

Equation 1-1. Frame Frequency of a McBSP

Frame Frequency = Clock Frequency 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.

#### 1.3.3.8 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 1–2.

#### Equation 1-2. Maximum Frame Frequency of a McBSP

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

Figure 1-10 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 1-10. 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.

SPRU061D

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 (MCLKXA). For more details, see *Set the Transmit Data Delay* on page 3-37.

#### 1.3.4 Frame Phases

The McBSP allows you to configure each frame to contain one or two phases. The number of words, 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.

#### 1.3.4.1 Number of Phases, Words, and Bits Per Frame

Table 1–4 shows which bit fields in the receive control registers (RCR1 and RCR2) and in the transmit control registers (XCR1 and XCR2) determine the number of phases per frame, the number of words per frame, and number of bits per word for each phase, for the receiver and transmitter. The maximum number of words per frame is 128 for a single-phase frame and 256 for a dual-phase frame. The number of bits per word can be 8, 12, 16, 20, 24, or 32 bits.

#### Table 1-4. McBSP Register Bits That Determine the Number of Phases, Words, and Bits Per Frame

Operation	Number of Phases	Words Per Phase 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

#### 1.3.4.2 Single-Phase Frame Example

Figure 1–11 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 MDXA and MDRA 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
- $\square$  (R/X)FRLEN1 = 0b: 1 word per frame
- $\Box$  (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
- $\square$  FS(R/X)P = 0: Active-high frame-sync signals
- $\Box$  (R/X)DATDLY = 01b: 1-bit data delay

Figure 1-11. Single-Phase Frame for a McBSP Data Transfer



#### 1.3.4.3 Dual-Phase Frame Example

Figure 1–12 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.





#### 1.3.4.4 Implementing the AC97 Standard With a Dual-Phase Frame

Figure 1–13 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.

1-19



Figure 1-13. Implementing the AC97 Standard With a Dual-Phase Frame

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 1-14. Timing of an AC97-Standard Data Transfer Near Frame Synchronization



1-20 Multichannel Buffered Serial Port (McBSP)

#### 1.3.5 McBSP Reception

This section explains the fundamental process of reception in the McBSP. For details about how to program the McBSP receiver, see *Receiver Configuration* on page 3-2.

Figure 1–15 and Figure 1–16 show how reception occurs in the McBSP. Figure 1–15 shows the physical path for the data. Figure 1–16 is a timing diagram showing signal activity for one possible reception scenario. A description of the process follows the figures.

Figure 1-15. McBSP Reception Physical Data Path



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



Figure 1-16. McBSP Reception Signal Activity

owing process describes how data travels from the MDRA pin to

The following process describes how data travels from the MDRA pin to the CPU:

- 1) The McBSP waits for a receive frame-sync pulse on internal MFSRA.
- 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 1-16), a 1-bit data delay is selected.

 The McBSP accepts data bits on the MDRA 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

MCLKRA: Internal receive clock MDRA: Data on MDRA pin MFSRA: Internal receive frame-sync signal RRDY: Status of receiver ready bit (high is 1)

most significant bits. For details on choosing a word length, see *Set the Receive Word Length(s)*.

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 received data is ready to be read by the CPU.

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 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 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 for more details:

- Overrun in the Receiver (page 1-39)
- Unexpected Receive Frame-Sync Pulse (page 1-40)

#### 1.3.6 McBSP Transmission

This section explains the fundamental process of transmission in the McBSP. For details about how to program the McBSP transmitter, see *Transmitter Configuration* on page 3-27.

1-22 Multichannel Buffered Serial Port (McBSP)
Figure 1–17 and Figure 1–18 show how transmission occurs in the McBSP. Figure 1–17 shows the physical path for the data. Figure 1–18 is a timing diagram showing signal activity for one possible transmission scenario. A description of the process follows the figures.

Figure 1-17. McBSP Transmission Physical Data Path



XSR[1,2]: Transmit shift registers 1 and 2

DXR[1,2]: Data transmit registers 1 and 2

Figure 1-18. McBSP Transmission Signal Activity



MCLKXA: Internal transmit clockMDXA: Data on MDXA pinMFSXA: Internal transmit frame-sync signalXRDY: Status of transmitter ready bit (high is 1)

 The CPU writes data to the data transmit register(s). When DXR1 is loaded, the transmitter ready bit (XRDY) is cleared in SPCR2 to indicate that the transmitter is not ready for new data.

If the word length is 16 bits or smaller, only DXR1 is used. If the word length is larger than 16 bits, DXR2 and DXR1 are used, and DXR2 contains the most significant bits. For details on choosing a word length, see *Set the Transmit Word Length(s)* in Chapter 3.

#### Note:

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

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  $\mu$ -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 MFSXA.
- 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 1-18), a 1-bit data delay is selected.

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

When activity is not properly timed, errors can occur. See the following topics for more details:

- Overwrite in the Transmitter (page 1-43)
- Underflow in the Transmitter (page 1-44)
- Unexpected Transmit Frame-Sync Pulse (page 1-46)

## 1.3.7 Interrupts and FIFO Events Generated by a McBSP

The McBSP sends notification of important events to the CPU and FIFO via the internal signals shown in Table 1–5.

Table 1-5. Interrupts and FIFO 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).

 Table 1-5. Interrupts and FIFO Events Generated by a McBSP (Continued)

Internal Signal	Description
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 FIFO when data has been received in the data receive registers (DRRs).
XEVT	Transmit synchronization event
	An XEVT signal is sent to the FIFO when the data transmit registers (DXRs) are ready to accept the next serial word for transmission.
REVTA	A-bis mode receive synchronization event
	If ABIS = 1 (A-bis mode is enabled) an REVTA signal is sent to the FIFO every 16 cycles.
XEVTA	A-bis mode transmit synchronization event
	If ABIS = 1 (A-bis mode is enabled) an XEVTA signal is sent to the FIFO every 16 cycles.

# 1.4 Sample Rate Generator of the McBSP

The McBSP contains a sample rate generator (SRG) module that can be programmed to generate internal data-clock (CLKG) and an internal frame-synchronization (FSG) signal. Figure 1–19 is a conceptual block diagram that shows the clock selection options available on the McBSP. The clock selection to the module is selectable using the SCLKME bit (PCR bit 7) and CLKSM bit (SRGR2 bit 13).

Figure 1-19. Sample Rate Generator Clock Selection



CLKG can be used for bit shifting on the data receive (MDRA) pin and/or the data transmit (MDXA) pin. FSG can be used to initiate frame transfers on MDRA and/or MDXA.

The source clock for the sample rate generator (labeled CLKSRG in the diagram) can be supplied by the LSPCLK or by an external pin (MCLKXA or MCLKRA). 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 (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:

- CLKG 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.

In addition to the 3-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 MFSRA pin. This feature is enabled or disabled with the GSYNC bit of SRGR2.

For details on getting the sample rate generator ready for operation, see the reset and initialization procedure on page 1-33.

## 1.4.1 Clock Generation in the Sample Rate Generator

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both. Use of the sample rate generator to drive clocking is controlled by the clock mode bits (CLKRM and CLKXM) in the pin control register (PCR). When a clock mode bit is set to 1 (CLKRM = 1 for reception, CLKXM = 1 for transmission), the corresponding data clock (MCLKRA for reception, MCLKXA 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, as described in Table 1–6. The digital loopback mode (described in Chapter 3) 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 ( $\overline{GRST} = 1$ ).

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

Table 1-6.	Effects of DLB and	CLKSTP on	Clock Modes
------------	--------------------	-----------	-------------

## 1.4.1.1 Choosing an Input Clock

The sample rate generator must be driven by an input clock signal from one of the three sources selectable with the SCLKME bit of PCR and the CLKSM bit of SRGR2 (see Table 1–7). When CLKSM = 1, the minimum divide down value in CLKGDV bits should be 1. CLKGDV is described in section 1.4.1.3.

Table 1-7.	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	Reserved
0	1	LSPCLK
1	0	Signal on MCLKRA pin
1	1	Signal on MCLKXA pin

## 1.4.1.2 Choosing a Polarity for the Input Clock

As shown in Figure 1–20, 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 1–8.

1-28 Multichannel Buffered Serial Port (McBSP)



#### Figure 1-20. Possible Inputs to the Sample Rate Generator and the Polarity Bits

Table 1-8. Polarity Options for the Input to the Sample Rate Generator

Input Clock	Polarity Option	Effect
LSPCLK	Always positive polarity	Rising edge of CPU clock generates transitions on CLKG and FSG.
Signal on MCLKRA pin	CLKRP = 0 in PCR	Falling edge on MCLKRA pin generates transitions on CLKG and FSG.
	CLKRP = 1 in PCR	Rising edge on MCLKRA pin generates transitions on CLKG and FSG.
Signal on MCLKXA pin	CLKXP = 0 in PCR	Rising edge on MCLKXA pin generates transitions on CLKG and FSG.
	CLKXP = 1 in PCR	Falling edge on MCLKXA pin generates transitions on CLKG and FSG.

## 1.4.1.3 Choosing a Frequency for the Output Clock (CLKG)

The input clock (LSPCLK or external clock) can be divided down by a programmable value to drive CLKG. Regardless of the source to the sample rate generator, the rising edge of CLKSRG (see the sample rate generator diagram on page 1-26) 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.

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

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.

## 1.4.1.4 Keeping CLKG Synchronized to External MCLKR

When the MCKLR pin is used to drive the sample rate generator (see section 1.4.1.1), the GSYNC bit in SRGR2 and the FSRA pin can be used to configure the timing of the output clock (CLKG) relative to the input clock. Note that this feature is available only when the MCLKR pin is used to feed the external 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 MFSRA pin triggers a resynchronization of CLKG and generation of FSG.

For more details about the synchronization, see section 1.4.3 on page 1-31.

## 1.4.2 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, make sure 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 MFSXA 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 ( $\overline{GRST} = 1$ ) and the frame-sync logic in the sample rate generator must be enabled ( $\overline{FRST} = 1$ ).

## 1.4.2.1 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.

1-30 Multichannel Buffered Serial Port (McBSP)

#### 1.4.2.2 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 MFSRA 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.

#### 1.4.2.3 Keeping FSG Synchronized to an External Clock

When an external signal is selected to drive the sample rate generator (see section 1.4.1.1 on page 1-28), the GSYNC bit of SRGR2 and the MFSRA pin can be used to configure the timing of FSG pulses.

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

Section 1.4.3 has more details about the synchronization.

#### 1.4.3 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, MCLKRA, or MCLKXA pin. When an external clock is selected to drive the sample rate generator, the GSYNC bit of SRGR2 and the MFSRA 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 divide down the input clock with the same phase relationship. If GSYNC = 1:

- ☐ An inactive-to-active transition on the MFSRA pin triggers a resynchronization of CLKG and a pulsing of FSG.
- CLKG always begins with a high state after synchronization.
- MFSRA is always detected at the same edge of the input clock signal that generates CLKG, no matter how long the MFSRA pulse is.

SPRU061D

□ 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 MFSRA pin.

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

## 1.4.3.1 Operating the Transmitter Synchronously With the Receiver

When GSYNC = 1, the transmitter can operate synchronously with the receiver, provided that:

- MFSXA is programmed to be driven by FSG (FSGM = 1 in SRGR2 and FSXM = 1 in PCR). If the input MFSRA has appropriate timing so that it can be sampled by the falling edge of CLKG, it can be used, instead, by setting FSXM = 0 and connecting MFSRA to MFSXA externally.
- The sample rate generator clock drives the transmit and receive clocking (CLKRM = CLKXM = 1 in PCR).

#### 1.4.3.2 Synchronization Examples

Figure 1–21 and Figure 1–22 show the clock and frame-synchronization operation with various polarities of CLKS (the chosen input clock) and MFSRA. 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 MFSRA 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).









#### 1.4.4 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.

SPRU061D

Multichannel Buffered Serial Port (McBSP) 1-33

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

## 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  $\overline{\text{GRST}}$  = 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

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

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	Reserved
0	1	LSPCLK
1	0	Signal on MCLKRA pin
1	1	Signal on MCLKXA 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  $\overline{\text{FRST}} = 1$  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.

#### 1.4.5 Sample Rate Generator Clocking Examples

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

#### 1.4.5.1 Double-Rate ST-Bus Clock

Figure 1-23 shows McBSP configuration to be compatible with the Mitel ST-Bus. Note that this operation is running at maximum frame frequency (described on page 1-17).



Figure 1-23. ST-BUS and MVIP Clocking Example

For this McBSP configuration:

- DLB = 0: Digital loopback mode off, CLKSTP = 00b: Clock stop mode off, and CLKR/CLKX = 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 drives the sample rate generator
- CLKSP = 1: Falling edge of input clock generates CLKG and thus internal CLKR/CLKX
- CLKGDV = 1: Frequency of receive clock (shown as CLKR) is half of input clock 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

#### 1.4.5.2 Single-Rate ST-Bus Clock

The example in Figure 1–24 is the same as the double-rate ST-bus clock example in section 1.4.5.1 except that:

- CLKGDV = 0: CLKX/CLKR driven without any divide-down (single-rate clock).
- CLKRP/CLKXP = 0: Rising edge of input clock generates CLKG and internal CLKR/CLKX.





The rising edge of the input clock is used to detect the external MFSRA pulse, which resynchronizes the internal McBSP clocks and generates a frame-sync

1-36 Multichannel Buffered Serial Port (McBSP)

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.

## 1.4.5.3 Other Double-Rate Clock

The example in Figure 1–25 is the same as the double-rate ST-bus clock example in section 1.4.5.1 except that:

- CLKRP/CLKXP = 0: Rising edge of input clock generates CLKG and thus CLKR/CLKX
- CLKGDV = 1: Frequency of CLKG (and thus internal CLKR and internal CLKX) is half input clock frequency
- □ FSRM/FSXM = 0: Frame synchronization is externally generated. The frame-sync pulse is wide enough to be detected.
- GSYNC = 0: CLKG runs freely; it is not resynchronized by a pulse on the MFSRA pin.
- **FSRP/FSXP** = 0: Active-high input frame-sync signal
- BDATDLY/XDATDLY = 1: Data delay of one bit

#### Figure 1-25. Double-Rate Clock Example



# 1.5 McBSP Exception/Error Conditions

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

- Receiver Overrun (RFULL = 1). This occurs when DRR1 has not been read since the last RBR-to-DRR copy. Consequently, the receiver does not copy a new word from the RBR(s) to the DRR(s), and the RSR(s) are now full with another new word shifted in from MDRA. Therefore, RFULL = 1 indicates an error condition wherein any new data that may arrive at this time on MDRA 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 MDRA and DRR1 is not read. For more details about overrun in the receiver, see page 1-39.
- Unexpected Receive Frame-Sync Pulse (RSYNCERR = 1). 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. For more details about receive frame-sync errors, see page 1-40.
- □ **Transmitter Data Overwrite**. This occurs when the CPU overwrites data in the DXR(s) before the data is copied to the XSR(s). The overwritten data never reaches the MDXA pin. For more details about overwrite in the transmitter, see page 1-43.
- □ **Transmitter Underflow (XEMPTY = 0)**. 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. For more details about underflow in the transmitter, see page 1-44.
- Unexpected Transmit Frame-Synch Pulse (XSYNCERR = 1). 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. For more details about transmit frame-sync errors, see page 1-46.

#### 1.5.1 Overrun in the Receiver

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

- 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 (page 1-21), 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 MDRA 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 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 reads DRR1.
- $\Box$  The receiver is reset individually (RRST = 0) or as part of a DSP reset.

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

#### 1.5.1.1 Example of the Overrun Condition

Figure 1–26 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.

SPRU061D



Figure 1-26. Overrun in the McBSP Receiver

## 1.5.1.2 Example of Preventing the Overrun Condition

Figure 1–27 shows the case where RFULL is set, but 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.





## 1.5.2 Unexpected Receive Frame-Sync Pulse

Section 1.5.2.1 shows how the McBSP responds to any receive frame-sync pulses, including an unexpected pulse. Sections 1.5.2.2 and 1.5.2.3 show an example of a frame-sync error and an example of how to prevent such an error, respectively.

## 1.5.2.1 Possible Responses to Receive Frame-Sync Pulses

Figure 1–28 shows the decision tree that the receiver uses to handle all incoming frame-sync pulses. The figure assumes that the receiver has been started ( $\overline{RRST} = 1$  in SPCR1). Case 3 in the figure is the case in which an error occurs.

1-40 Multichannel Buffered Serial Port (McBSP)



#### Figure 1-28. Possible Responses to Receive Frame-Sync Pulses

Any one of three cases can occur:

- □ **Case 1:** Unexpected internal MFSRA 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 MFSRA pulse is the first after the receiver is enabled (RRST = 1 in SPCR1).
  - The MFSRA pulse is the first 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.

## 1.5.2.2 Example of an Unexpected Receive Frame-Sync Pulse

Figure 1–29 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 1-29. An Unexpected Frame-Sync Pulse During a McBSP Reception

## 1.5.2.3 Preventing Unexpected Receive Frame-Sync Pulses

Each frame transfer can be delayed by 0, 1, or 2 MCLKRA cycles, depending on the value in the RDATDLY bits of RCR2. For each possible data delay, Figure 1–30 shows when a new frame-sync pulse on MFSRA can safely occur relative to the last bit of the current frame.

Figure 1-30. Proper Positioning of Frame-Sync Pulses



## 1.5.3 Overwrite in the Transmitter

As described in the section on McBSP transmission (page 1-22), after the CPU 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 MDXA 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.

1-43

SPRU061D

#### 1.5.3.1 Example of the Overwrite Condition

Figure 1–31 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 MDXA.

Figure 1-31. Data in the McBSP Transmitter Overwritten and Thus Not Transmitted



#### 1.5.3.2 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.

## 1.5.4 Underflow in the Transmitter

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

- 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 MDXA pin.
- The transmitter is reset (by forcing  $\overline{XRST} = 0$  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.

1-44 Multichannel Buffered Serial Port (McBSP)

#### Note:

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

When the transmitter is taken out of reset ( $\overline{XRST} = 1$ ), it is in a transmitter ready (XRDY = 1 in SPCR2) and transmitter empty ( $\overline{XEMPTY} = 0$ ) state. If DXR1 is loaded by the CPU before internal MFSXA 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 MDXA.

#### 1.5.4.1 Example of the Underflow Condition

Figure 1–32 shows an underflow condition. After B is transmitted, DXR1 is not reloaded before the subsequent frame-sync pulse. Thus, B is again transmitted on MDXA.



Figure 1-32. Underflow During McBSP Transmission

#### 1.5.4.2 Example of Preventing the Underflow Condition

Figure 1–33 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 1-33. Underflow Prevented in the McBSP Transmitter

## 1.5.5 Unexpected Transmit Frame-Sync Pulse

Section 1.5.5.1 shows how the McBSP responds to any transmit frame-sync pulses, including an unexpected pulse. Sections 1.5.5.2 and 1.5.5.3 show an examples of a frame-sync error and an example of how to prevent such an error, respectively.

#### 1.5.5.1 Possible Responses to Transmit Frame-Sync Pulses

Figure 1–34 shows the decision tree that the transmitter uses to handle all incoming frame-sync pulses. The figure assumes that the transmitter has been started ( $\overline{XRST} = 1$  in SPCR2). Case 3 in the figure is the case in which an error occurs.





Any one of three cases can occur:

- □ **Case 1:** Unexpected internal MFSXA pulses with XFIG = 1 in XCR2. 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 MFSXA pulse is the first after the transmitter is enabled ( $\overline{XRST} = 1$ ).

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.

## 1.5.5.2 Example of an Unexpected Transmit Frame-Sync Pulse

Figure 1–35 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 1-35. An Unexpected Frame-Sync Pulse During a McBSP Transmission



## 1.5.5.3 Preventing Unexpected Transmit Frame-Sync Pulses

Each frame transfer can be delayed by 0, 1, or 2 MCLKXA cycles, depending on the value in the XDATDLY bits of XCR2. For each possible data delay, Figure 1–36 shows when a new frame-sync pulse on MFSXA can safely occur relative to the last bit of the current frame.



Figure 1-36. Proper Positioning of Frame-Sync Pulses

# Chapter 2

# **Multichannel Selection Modes**

This chapter explains how to assign blocks to partitions for multichannel selection.

# Topic

# Page

2.1	Channels, Blocks, and Partitions 2-2
2.2	A-bis Mode
2.3	SPI Protocol 2-16

#### 2.1 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 (see Table 2-1 through Table 2-3).

- □ It is possible to have two receive partitions (A and B) and 8 transmit partitions (A H).
- □ McBSP can transmit/receive on selected channels.
- Each channel partition has a dedicated channel-enable register. Each bit controls whether data flow is allowed or prevented in one of the channels assigned to that partition.
- □ There are three transmit multichannel modes and one receive multichannel mode.

Table 2-1.	Block ·	- Channel	Assignment
------------	---------	-----------	------------

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

The blocks are assigned to partitions according to the selected partition mode. In the two-partition mode (described in Section 2.1.3), you assign one even-numbered block (0, 2, 4, or 6) to partition A and one odd-numbered block (1, 3, 5, or 7) to partition B. In the 8-partition mode (described in Section 2.1.4), blocks 0 through 7 are automatically assigned to partitions, A through H, respectively.

Table 2-2. 2-Partition Mode

Partition	Blocks
А	0 or 2 or 4 or 6
В	1 or 3 or 5 or 7

Taple 2-3. O-Partition Moue
-----------------------------

Partition	Blocks	Channels
А	0	0 -15
В	1	16 - 31
С	2	32 - 47
D	3	48 - 63
E	4	64 - 79
F	5	80 - 95
G	6	96 - 111
н	7	112 - 127

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

#### 2.1.1 Multichannel Selection

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

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

The McBSP has one receive multichannel selection mode (described in section 2.1.5) and three transmit multichannel selection modes (described in section 2.1.6).

## 2.1.2 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.

## 2.1.3 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.

#### 2.1.3.1 Assigning Blocks to Partitions A and B

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

For reception:

- ❑ Assign an even-numbered channel block (0, 2, 4, or 6) to receive partition A by writing to the RPABLK bits. In the receive multichannel selection mode (described in section 2.1.5), the channels in this partition are controlled by receive channel enable register A (RCERA).
- Assign an odd-numbered block (1, 3, 5, or 7) to receive partition B with the RPBBLK bits. In the receive multichannel selection mode, the channels in this partition are controlled by receive channel enable register B (RCERB).

For transmission:

- Assign an even-numbered channel block (0, 2, 4, or 6) to transmit partition A by writing to the XPABLK bits. In one of the transmit multichannel selection modes (described in section 2.1.6), 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 2–1 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 2-1. Alternating Between the Channels of Partition A and the Channels of Partition B

Partition	A	В	А	В	А	В	A	В	А
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
FS(R/X)									

#### 2-partition mode. Example with fixed block assignments

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

#### 2.1.3.2 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 associated block assignment bits cannot be modified, and its associated channel enable register cannot be modified. For example, if block 2 is being transferred and block 2 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 2–2 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 2-2. Reassigning Channel Blocks Throughout a McBSP Data Transfer



## 2.1.4 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 channels 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 2–4 and Table 2–5. These assignments cannot be changed. The tables also show the registers used to control the channels in the partitions.

Receive Partition	Assigned Block of Receive Channels	Register Used For Channel Control
А	Block 0: channels 0 through 15	RCERA
В	Block 1: channels 16 through 31	RCERB
С	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
Н	Block 7: channels 112 through 127	RCERH

Table 2-4. Receive Channel Assignment and Control When Eight Receive Partitions Are Used

Table 2-5. Transmit Channel Assignment and Control When Eight TransmitPartitions Are Used

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

Figure 2–3 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 2-3. McBSP Data Transfer in the 8-Partition Mode

#### 2.1.5 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 receive FIFO 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 MDRA pin in channel 0
- 2) Ignores bits received in channels 1–14
- 3) Accepts bits shifted in from the MDRA pin in channel 15
- 4) Ignores bits received in channels 16–38
- 5) Accepts bits shifted in from the MDRA pin in channel 39
# 2.1.6 Transmit Multichannel Selection Modes

The XMCM bits of XCR2 determine whether all channels or only selected channels are enabled and unmasked for transmission. More details on enabling and masking are in section 2.1.7. The McBSP has three transmit multichannel selection modes (XMCM = 01b, XMCM = 10b, and XMCM = 11b), which are described in the following table:

Table 2-6. Selecting a Transmit Multichannel Selection Mode With the XMCM Bits

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

# 2.1.7 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 MDXA pin is held in the high impedance state; data cannot be shifted out on the MDXA 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 transmit FIFO 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 MDXA pin.

# 2.1.8 Activity on McBSP Pins for Different Values of XMCM

Figure 2-4 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 2-4. Activity on McBSP Pins for the Possible Values of XMCM



2-12 Multichannel Selection Modes

SPRU061D

# 2.1.9 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 = 1b, no interrupt is generated unless a multichannel selection mode is on.

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.

# 2.2 A-bis Mode

In the A-bis mode (ABIS = 1 in SPCR1), the McBSP can receive and transmit up to 1024 bits on a PCM link. The receive section can extract all 1024 bits from a 1024-bit PCM frame according to a given bit-enable pattern, and generate an interrupt to the CPU when 16 enabled bits have been compacted into a word in DRR1, or when a receive frame is complete. In addition, the transmit section can expand up to 1024 bits into a 1024-bit PCM frame at a specific position, according to a given bit-enable pattern, and generate an interrupt when 16 enabled bits have been transmitted or a transmit frame is complete.

The bit-enable patterns are specified with channel enable registers A and B (RCERA and RCERB for reception, XCERA and XCERB for transmission). These registers have a different function than in the multichannel selection modes (described in section 2.1). Instead of indicating which channels will be enabled, these registers indicate which bits in the data stream will be enabled. A 1 in a given position in the (R/X)CER(A/B) register enables a corresponding bit in the receive/transmit data stream.

The A-bis mode requires a word length of 16 bits (for reception: RWDLEN1 = 010b in RCR1, for transmission: XWDLEN1 = 010b in XCR1). Otherwise, operation in the A-bis mode is undetermined.

# 2.2.1 A-bis Mode Receive Operation

In the A-bis mode, bits that are not enabled in the RCERA and RCERB registers are ignored and are not compacted in the receiver. Bits that are enabled are received and compacted. When 16 enabled bits have been received, the received word is copied from RSR1 to DRR1 and the McBSP generates an interrupt to the CPU. RCERA and RCERB alternate specifying the receive masking pattern for each of the 16 receive clocks. Figure 2-5 shows an example bit sequence for the receiver (in the figure, – indicates that the bit on the MDRA pin is ignored and thus is not passed to DRR1).

RCERA	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1																		
RCERB																		0	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	
MDRA pin	1	0	1	1	0	1	0	1	0	0	1	1	0	1	0	1		0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	
DRR1	-	0	-	1	-	1	-	1	-	0	-	1	0	-	-	1		-	-	1	-	-	1	1	0	-	-	0	-	-	0	1	1	75E3h
Note: - ind	icat	tes	tha	t th	e b	it o	n th	ne N	٨DI	RA	pin	is	ian	ore	d a	nd	th	us	is r	not	pas	se	d to	D D	۲R	1.								

Figure 2-5. A-bis Mode Receive Operation

# 2.2.2 A-bis Mode Transmit Operation

In the A-bis mode, only bits that are enabled in the XCERA and XCERB registers are transmitted out from the MDXA pin. Bits that are not enabled are not transmitted, and the MDXA pin is in the high-impedance state during that clock cycle. XCERA and XCERB alternate specifying the bit-enable pattern for each 16 clock cycles. When 16 enabled have been shifted out, the McBSP generates an interrupt to the CPU. Figure 2–6 shows an example bit sequence for the transmitter (in the figure, z indicates the high-impedance state).

Figure 2-6. A-bis Mode Transmit Operation

XCERA	0	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1																
XCERB																	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0
DXR1	1	0	1	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0
MDXA pin	z	0	1	1	0	1	z	z	z	0	0	z	z	1	1	1	z	z	1	1	z	z	z	0	1	z	z	z	0	0	z	z
Note: z indi	cat	es	the	hig	jh-i	mp	eda	ince	e st	ate																						

# 2.3 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 SPISOMI)
- Serial data output (also referred to as Master Out Slave In, or SPISIMO)
- □ Shift-clock (also referred to as SPICIK)
- Slave-enable signal (also referred to as SPISTE)

A typical SPI interface with a single slave device is shown in Figure 2-7.

# Figure 2-7. 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. In such a configuration, the slave device must remain enabled at all times, and multiple slaves cannot be used.

# 2.3.1 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 (MCLKXA) corresponds to the serial clock signal (SPISCK) of the SPI protocol, while the transmit frame-synchronization signal (MFSXA) is used as the slave-enable signal (SPISTE).

The receive clock signal (MCLKRA) and receive frame-synchronization signal (MFSRA) are not used in the clock stop mode because these signals are internally connected to their transmit counterparts, MCLKXA and MFSXA.

# 2.3.2 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 2–7. Table 2–8 shows how the various combinations of the CLKSTP bit and the polarity bits CLKXP and CLKRP create four possible clock stop mode configurations. The timing diagrams in section 2.3.3 show the effects of CLKSTP, CLKXP, and CLKRP.

Bit Field	Description
CLKSTP bits of SPCR1	Use these bits to enable the clock stop mode and to select one of two timing variations. (See also Table 2-8.)
CLKXP bit of PCR	This bit determines the polarity of the MCLKXA signal. (See also Table 2-8.)
CLKRP bit of PCR	This bit determines the polarity of the MCLKRA signal. (See also Table 2-8.)
CLKXM bit of PCR	This bit determines whether MCLKXA 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 2-7. Bits Used to Enable and Configure the Clock Stop Mode

Table 2-8. 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 MCLKXA and receives data on the falling edge of MCLKRA.
CLKSTP = 11b CLKXP = 0 CLKRP = 1	Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of MCLKXA and receives data on the rising edge of MCLKRA.
CLKSTP = 10b CLKXP = 1 CLKRP = 0	High inactive state without delay: The McBSP transmits data on the falling edge of MCLKXA and receives data on the rising edge of MCLKRA.
CLKSTP = 11b CLKXP = 1 CLKRP = 1	High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of MCLKXA and receives data on the falling edge of MCLKRA.

#### 2.3.3 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 MCLKXA signal is always stopped and the MFSXA 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 2-8. SPI Transfer With CLKSTP = 10b (no clock delay), CLKXP = 0, CLKRP = 0

Figure 2-9. SPI Transfer With CLKSTP = 11b (clock delay), CLKXP = 0, CLKRP = 1



 If the McBSP is the SPI master (CLKXM = 1), MISO = MDRA. If the McBSP is the SPI slave (CLKXM = 0), MISO = MDXA.





SPI slave (CLKXM = 0), MOSI = MDRA.

 If the McBSP is the SPI master (CLKXM = 1), MISO = MDRA. If the McBSP is the SPI slave (CLKXM = 0), MISO = MDXA.

SPRU061D

<sup>SPI slave (CLKXM = 0), MOSI = MDRA.
2) If the McBSP is the SPI master (CLKXM = 1), MISO = MDRA. If the McBSP is the</sup> 

SPI slave (CLKXM = 0), MISO = MDXA.



Figure 2-11.SPI Transfer With CLKSTP = 11b (clock delay), CLKXP = 1, CLKRP = 1

slave (CLKXM = 0), MOSI = MDRA.

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

# 2.3.4 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 ( $\overline{GRST} = 0$ ) 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. For a list of important bits settings, see one of the following topics:

- □ McBSP as the SPI Master (page 2-21)
- □ *McBSP as an SPI Slave* (page 2-23)
- 4) Enable the sample rate generator.

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

Make sure that during the write to SPCR2, you only modify <u>GRST</u>. 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 ( $\overline{XRST} = 1$  in SPCR2) and enable the receiver ( $\overline{RRST} = 1$  in SPCR1).

The FIFO has to be configured first if it must write into the McBSP transmit buffer or read from the McBSP receive buffer. After the FIFO has been configured, make XRST = 1 and  $\overline{RRST}$  = 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  $\overline{FRST} = 1$  if an internally generated frame-sync pulse is required (that is, if the McBSP is the SPI master).

# 2.3.5 McBSP as the SPI Master

An SPI interface with the McBSP used as the master is shown in Figure 2–12. When the McBSP is configured as a master, the transmit output signal (MDXA) is used as the SPISIMO signal of the SPI protocol, and the receive input signal (MDRA) is used as the SPISOMI signal.

Figure 2-12. SPI Interface With McBSP as Master



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 2-9. 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 MCLKXA as seen on the MCLKXA pin is positive (CLKXP = 0) or negative (CLKXP = 1).
CLKRP = 0 or 1	The polarity of MCLKRA as seen on the MCLKRA pin is positive (CLKRP = 0) or negative (CLKRP = 1).
CLKXM = 1	The MCLKXA pin is an output pin driven by the internal sample rate generator. Because CLKSTP is equal to 10b or 11b, MCLKRA is driven internally by MCLKXA.
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 MFSXA pin is an output pin driven according to the FSGM bit.
FSGM = 0	The transmitter drives a frame-sync pulse on the MFSXA pin every time data is transferred from DXR1 to XSR1.
FSXP = 1	The MFSXA pin is active low.
XDATDLY = 01b RDATDLY = 01b	This setting provides the correct setup time on the MFSXA 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 MCLKXA pin is enabled only during packet transfers. When packets are not being transferred, the MCLKXA pin remains high or low depending on the polarity used.

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

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 (SPISTE) on the MFSXA pin. If a slave-enable signal is required, the MFSXA 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 MFSXA 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. To see the resulting waveform produced on the MFSXA pin, see the timing diagrams in section 2.3.3. 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 MFSXA signal returns to the inactive state.

# 2.3.6 McBSP as an SPI Slave

An SPI interface with the McBSP used as a slave is shown in Figure 2–13. When the McBSP is configured as a slave, MDXA is used as the SPISOMI signal, and MDRA is used as the SPISIMO signal.





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

Table 2-10. 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 MCLKXA as seen on the MCLKXA pin is positive (CLKXP = 0) or negative (CLKXP = 1).
CLKRP = 0 or 1	The polarity of MCLKRA as seen on the MCLKRA pin is positive (CLKRP = 0) or negative (CLKRP = 1).
CLKXM = 0	The MCLKXA pin is an input pin, so that it can be driven by the SPI master. Because CLKSTP = 10b or 11b, MCLKRA is driven internally by MCLKXA.
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 MFSXA pin is an input pin, so that it can be driven by the SPI master.
FSXP = 1	The MFSXA 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 MCLKXA and MFSXA pins must be configured as inputs. The MCLKXA pin is internally connected to the MCLKRA signal, so that both the transmit and receive circuits of the McBSP are clocked by the external master clock. The MFSXA pin is also internally connected to the MFSRA signal, and no external signal connections are required on the MCLKRA and MFSRA pins.

Although the MCLKXA 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 MFSXA 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. Unlike the standard SPI, this pin cannot be tied low all the time.

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 3

# **Configure the Receiver and Transmitter**

This chapter explains how to configure the receiver/transmitter and program the registers.

# TopicPage3.1Receiver Configuration3-23.2Transmitter Configuration3-27

# 3.1 Receiver Configuration

To configure the McBSP receiver, perform the following procedure:

- 1) Place the McBSP receiver in reset (see section 3.1.2).
- 2) Program the McBSP registers for the desired receiver operation (see section 3.1.1).
- 3) Take the receiver out of reset (see section 3.1.2).

# 3.1.1 Programming the McBSP Registers for the Desired Receiver Operation

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

#### 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
- Enable/disable the A-bis 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

# 3.1.2 Resetting and Enabling the Receiver

The first step of the receiver configuration procedure is to reset the receiver, and the last step is to enable the receiver (to take it out of reset).

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

- A DSP reset (XRS signal driven low) places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed (XRS signal released), GRST = FRST = RRST = XRST = 0, keeping the entire serial port in the reset state, provided the McBSP clock is turned on.
- 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.

To get this result	Use this bit
Enable the receiver	RRST (bit 0 of the SPCR1 register) set to 1
Disable the receiver (the reset state)	RRST set to 0
Reset the sample-rate generator	GRST (bit 6 of the SPCR2 register) set to 0
Enable the sample-rate generator	GRST set to 1
Reset the frame-sync logic	FRST (bit 7 of the SPCR2 register) = 0
Enable the frame-sync logic	FRST = 1

For details on the Serial Port Control Register 1 (SPCR1) and SPCR2 see Figure 6-4 (page 6-7) and Figure 6-3 (page 6-4). Table 3-1 shows the state of McBSP pins when the serial port is reset due to a DSP reset and a direct receiver/transmitter reset.

Pin	Possible State(s)	State Forced By DSP Reset	State Forced By Receiver/Transmitter Reset
			Receiver Reset ( $\overline{RRST} = 0$ and $\overline{GRST} = 1$ )
MDRA	I	GPIO input	Input
MCLKRA	I/O/Z	GPIO input	Known state if Input; MCLKRA running if output
MFSRA	I/O/Z	GPIO input	Known state if Input; FSRP inactive state if output
			Transmitter Reset ( $\overline{XRST} = 0$ and $\overline{GRST} = 1$ )
MDXA	O/Z	GPIO input	High impedance
MCLKXA	I/O/Z	GPIO input	Known state if Input; MCLKXA running if output
MFSXA	I/O/Z	GPIO input	Known state if Input; FSXP inactive state if output

Table 3-1. Reset State of Each McBSP Pin

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

For more details about McBSP reset conditions and effects, see *Section 4.1.1 Resetting and Initializing a McBSP* on page 4-3.

# 3.1.3 Set the Receiver Pins to Operate as McBSP Pins

To configure a pin for its McBSP function, you should configure the bits of the GPxMUXn register appropriately. In addition to this, bits 12 and 13 of the PCR register must be set to 0. These bits are defined as reserved.

# 3.1.4 Enable/Disable the Digital Loopback Mode

The DLB bit determines whether the digital loopback mode is on. DLB is described in Table 3–2.

Table 3-2. Register Bit Used to Enable/Disable the Digital Loopback Mode

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

# 3.1.5 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 3–3. This mode allows testing of serial port code with a single DSP device; the McBSP receives the data it transmits.

Table 3-3. Receive Signals Connected to Transmit Signals in Digital Loopback Mode

This receive signal	Is fed internally by this transmit signal …
MDRA (receive data)	MDXA (transmit data)
MFSRA (receive frame synchronization)	MFSXA (transmit frame synchronization)
MCLKRA (receive clock)	MCLKXA (transmit clock)

# 3.1.6 Clock Stop Mode

The CLKSTP bits in the serial port control registers determine whether the clock stop mode is on.

The clock stop mode supports the SPI master-slave protocol. If you will not be using the SPI protocol, you can clear the CLKSTP bits (12:11 in SPCR1) 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 MCLKXA 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 MCLKRA pin.

Table 3-4 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.

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 MCLKXA and receives data on the falling edge of MCLKRA.
CLKSTP = 11b CLKXP = 0 CLKRP = 1	Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of MCLKXA and receives data on the rising edge of MCLKRA.
CLKSTP = 10b CLKXP = 1 CLKRP = 0	High inactive state without delay: The McBSP transmits data on the falling edge of MCLKXA and receives data on the rising edge of MCLKRA.
CLKSTP = 11b CLKXP = 1 CLKRP = 1	High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of MCLKXA and receives data on the falling edge of MCLKRA.

Table 3-4. Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme

#### 3.1.7 Receive Multichannel Selection and A-bis Modes

The RMCM bit determines whether the receive multichannel selection mode is on. For more details on this mode, see *Receive Multichannel Selection Mode* on page 2-8. For more details on the multichannel control registers, see Section 6.6 (page 6-22).

For more details on A-bis mode, see page 2-14. If A-bis mode is enabled, individual bits can be enabled or disabled during reception and transmission. For transmission, the bits are controlled by transmit channel enable registers A and B (XCERA and XCERB). For reception, the bits are controlled by receive channel enable registers A and B (RCERA and RCERB).

To get this result	Use this bit …
Enable multichannel selection mode	RMCM bit in MCR1 to 1 (individual channels can be enabled or disabled with this setting)
Disable multichannel selection mode	Set RMCM to 0 so that all 128 channels are enabled
Enable A-bis mode	ABIS bit (number 6 in SPCR1) set to 1
Disable A-bis mode	ABIS set to 0

6 Configure the Receiver and Transmitter

# 3.1.8 Choose 1 or 2 Phases for the Receive Frame

The RPHASE bit in the Receive Control Register 2 (RCR2) determines whether the receive data frame has one or two phases. See Section 6.3 for details of RCR1 and RCR2.

To get this result	Use this bit…
Set receive frame to single phase	RPHASE bit in Receive FRCR2 set to 0
Set receive frame to dual-phase frame	RPHASE bit in RCR2 set to 1

# 3.1.8.1 Set the Receive Word Length(s)

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

To get this result	Use this bit…	
Specify the length of every serial word in phase 1 of the receive frame	Set RWDLEN1 (bits 7:5 of RCR1) to one of the following:	
	000	8 bits
	001	12 bits
	010	16 bits
	011	20 bits
	100	24 bits
	101	32 bits
	11x	Reserved

Each frame can have one or two phases, depending on the value that you load into the RPHASE bit. If a single-phase frame is selected, RWDLEN1 selects the length for every serial word received in the frame. If a dual-phase frame is selected, RWDLEN1 determines the length of the serial words in phase 1 of the frame, and RWDLEN2 determines the word length in phase 2 of the frame with the same bit settings. For details on the receive control registers, see Section 6.3 (page 6-9).

#### 3.1.8.2 Set the Receive Frame Length

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

To get this result Use this bit		it
Set receive frame length	Set RFRLEN1 (bits 14:8 of RCR1) one of the following: 000 0000 1 word in phase 1 000 0001 2 words in phase 1	
	111 1111	128 words in phase 1

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

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

The 7-bit RFRLEN fields allow up to 128 words per phase. See Table 3–5 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 the example, if you want a phase length of 128 words in phase 1, load 127 into RFRLEN1.

Table 3-5. 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	$0 \le \text{RFRLEN1} \le 127$	$0 \leq \text{RFRLEN2} \leq 127$	(RFRLEN1 + 1) + (RFRLEN2 + 1) words

# *3.1.8.3* Enable/Disable the Receive Frame-Sync Ignore Function

The RFIG bit controls the receive frame-sync ignore function.

If there is an unexpected receive frame-sync pulse …	Use this bit…
Restart the frame transfer	Set RFIG (bit 2 in RCR2) to 0
Ignore the unexpected pulses	Set RFIG to 1

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 MFSRA pulse causes the McBSP to discard the contents of RSR[1,2] in favor of the new incoming data. Therefore, if RFIG = 0 and an unexpected frame-sync pulse occurs, the serial port:

- 1) Aborts the current data transfer
- 2) Sets RSYNCERR in SPCR1 to 1
- 3) Begins the transfer of a new data word

For more details about the frame-sync error condition, see *Unexpected Receive Frame-Sync Pulse* on page 1-40.

#### 3.1.8.4 Examples Showing the Effects of RFIG

Figure 3–1 shows an example in which word B is interrupted by an unexpected frame-sync pulse when (R/X)FIG = 0. In the case of reception, the reception of B is aborted (B is lost), and a new data word (C in this example) is received after the appropriate data delay. This condition is a receive synchronization error, and thus sets the RSYNCERR bit.

Figure 3-1. Unexpected Frame-Sync Pulse With (R/X)FIG = 0



In contrast with Figure 3–1, Figure 3–2 shows McBSP operation when unexpected frame-sync signals are ignored (when (R/X)FIG = 1). Here, the transfer of word B is not affected by an unexpected pulse.



Figure 3-2. Unexpected Frame-Sync Pulse With (R/X)FIG = 1

# 3.1.9 Set the Receive Companding Mode

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

A-law and  $\mu$ -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 FIFO must be at least 16 bits wide.

The  $\mu$ -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 0, indicating an 8-bit wide serial data stream. If companding is enabled and either of the frame phases does not have an 8-bit word length, companding continues as if the word length is 8 bits.

Figure 3–3 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  $\mu$ -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.





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

3-10 Configure the Receiver and Transmitter

If the McBSP is otherwise unused (the serial port transmit and receive sections are reset), the companding hardware can compand internal data. See *Capability to Compand Internal Data* on page 1-13.

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

The RCOMPAND bits determine whether companding or another data transfer option is chosen for McBSP reception. Modes other than 00b are enabled only when the appropriate RWDLEN is 000b, indicating 8-bit data.

To get this result	Use this bit…	
Receive companding mode	Set RCOMPAND (bits 4:3 in RCR2) to one of the following	0
	00 No companding, any size data MSB received first	<b>1</b> ,
	01 No companding, 8-bit data, LS received first (for details, scrol down to Option to Receive LS First)	SB II B
	10 μ-law companding, 8-bit data, MSB received first	
	11 A-law companding, 8-bit data, MSB received first	

# 3.1.10 Set the Receive 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 shown in Figure 3-4. 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 3-4. Range of Programmable Data Delay

#### 3.1.10.1 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.

For reception this problem is solved because receive data is sampled on the first falling edge of MCLKRA 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 MDXA. The transmitter then asynchronously detects the frame-sync signal (MFSXA) going active high and immediately starts driving the first bit to be transmitted on the MDXA pin.

#### 3.1.10.2 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 Figure 3–5. In this figure, the data transferred is an 8-bit value with bits labeled B7, B6, B5, and so on.



Figure 3-5. 2-Bit Data Delay Used to Skip a Framing Bit

The RDATDLY bits determine the length of the data delay for the receive frame.

To get this result	Use this bit…		
Set the receive data delay	Set RDATDLY	Set RDATDLY (bits 1:0 in RCR2 to:	
	00 0-bit d	ata delay	
	01 1-bit d	ata delay	
	10 2-bit d	ata delay	
	11 Reser	ved	

# 3.1.11 Set the Receive Sign-Extension and Justification Mode

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 3-6 and Table 3-7 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 3-6. 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

SPRU061D

Configure the Receiver and Transmitter 3-13

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

Table 3-7.	Example:	Use of RJUST	Field With	20-Bit Data	Value 0xABCDE

The RJUST bits determine whether data received by the McBSP is sign extended and how it is justified.

To get this result	Use this bit…		
Set receive sign-extension and justification mode	Set R	Set RJUST (bits 14:13 in SPCR1 to:	
	00	Right justify data and zero fill MSBs in DRR[1,2]	
	01	Right justify data and sign extend it into the MSBs in DRR[1,2]	
	10	Left justify data and zero fill LSBs in DRR[1,2]	
	11	Reserved	

# 3.1.12 Set the Receive Interrupt Mode

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. For information on using RSYNCERR, see Unexpected Receive Frame-Sync Pulse on page 1-40. See section 5.4 for RINTM interrupt selection using FIFO logic.

The RINTM bits determine which event generates a receive interrupt request to the CPU.

To get this result	Use t	this bit	
Receive interrupt mode	Set R	Set RINTM (bits 5:4 in SPCR1 to:	
	00	RINT generated when RRDY changes from 0 to 1	
	01	RINT generated by an end-of-block or end-of-frame condition in the receive multichannel selection mode	
	10	RINT generated by a new receive frame-sync pulse	
	11	RINT generated when RSYNCERR is set	

#### 3.1.13 Set the Receive Frame-Sync Mode

Table 3–8 shows how you may select various sources to provide the receive frame-synchronization signal and the effect on the MFSRA pin. The polarity of the signal on the MFSRA 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.

DLB	FSRM	GSYNC	Source of Receive Frame Synchronization	MFSRA Pin Status
0	0	0 or 1	An external frame-sync signal enters the McBSP through the MFSRA 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 MFSRA 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 MFSRA 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 MFSRA 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 MFSRA pin.

Table 3-8. Select Sources to Provide the Receive Frame-Synchronization Signal and the Effect on the MFSRA Pin

# 3.1.14 Set the Receive Frame-Sync Polarity

The FSRP bit determines whether frame-synchronization (frame-sync) pulses are active high or active low on the MFSRA pin.

Receive frame-sync pulses can be either generated internally by the sample rate generator (see section 1.4.2 on page 1-30) or driven by an external source. The source of frame sync is selected by programming the mode bit, FSRM, in PCR. MFSRA is also affected by the GSYNC bit in SRGR2. For information about the effects of FSRM and GSYNC, see *Set the Receive Frame-Sync Mode* on page 3-15. Similarly, receive clocks can be selected to be inputs or outputs by programming the mode bit, CLKRM, in the PCR.

When MFSRA and MFSXA 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 MDRA 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 MFSRA and MFSXA 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 MDXA 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 MFSRA, MFSXA, MCLKRA, and MCLKXA 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 (MFSRA/MFSXA 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 (MFSRA/MFSXA 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 MCLKXA is an input), the external falling-edge triggered input clock on MCLKXA is inverted to a rising-edge triggered clock before being sent to the transmitter. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and MCLKXA is an output pin), the internal (rising-edge triggered) clock, internal CLKX, is inverted before being sent out on the MCLKXA 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 MCLKRA is an input pin), the external rising-edge triggered input clock on MCLKRA 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 MCLKRA 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 3-6 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 3-6. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



The FSRM (in PCR), GSYNC (in SRGR2), DLB and CLKSTP (both in SPCR1) determine the source for receive frame synchronization and the function of the MFSRA pin.

To get this result	Use this bit…
Frame synchronization	Set FSRM (bit 10 in PCR) to 0.
Supply receive frame synchronization by an external source via the MFSRA pin.	
Receive frame synchronization is supplied by the sample rate generator. MFSRA is an output pin reflecting internal FSR, except when GSYNC = 1 in SRGR2.	Set FSRM (bit 10 in PCR) to 1
Set sample rate generator clock synchronization mode	Set GSYNC (bit 15 in SRGR2) to 0.
Use no clock synchronization. CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.	

То	get this result …	Use this bit…
Use	e clock synchronization. When a pulse	Set GSYNC to 1.
	CLKG is adjusted as necessary so that it is synchronized with the input clock on the CLKS, MCLKRA, or MCLKXA pin.	For more details, see Synchronizing Sample Rate Generator Outputs to an External Clock on page 1-31.
	FSG pulses. FSG <b>only</b> pulses in response to a pulse on the MFSRA pin. The frame-sync period defined in FPER is ignored.	
Set Dis	t digital loopback (DLB) mode. able DLB mode	Set DLB (bit 15 in SPCR1) to 0.
Ena sigu sigu mu sigu	able digital loopback mode. The receive nals, including the receive frame-sync nal, are connected internally through Itiplexers to the corresponding transmit nals.	Set DLB to 1.
Set Dis for	t clock stop mode able clock stop mode - normal clocking non-SPI mode.	Set CLKSTP (bits 12:11 in SPCR1) to 0Xb
Ena del (CL fran inte cou	able clock stop mode without clock ay. The internal receive clock signal LKR) and the internal receive me-synchronization signal (FSR) are ernally connected to their transmit unterparts, MCLKXA and MFSXA.	Set CLKSTP (bits 12:11 in SPCR1) to 10b.
Clo The and fran inte cou	ock stop mode enabled, with clock delay. e internal receive clock signal (CLKR) d the internal receive me-synchronization signal (FSR) are ernally connected to their transmit unterparts, MCLKXA and MFSXA.	Set CLKSTP (bits 12:11 in SPCR1) to 11b.

#### 3.1.14.1 Set the SRG Frame-Sync Period and 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 3-7 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 3-7. Frame of Period 16 CLKG Periods and Active Width of 2 CLKG Periods



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.

To get this result	Use this bit…
Set the SRG frame-sync period and pulse width	FPER (bits 11:0 in SRGR2) set to range FPER + 1 (1 to 4096 CLKG cycles)
	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.
Set the sample rate generator	FWID (bits 15:8) in SRGR1
frame-sync pulse width	This field plus 1 determines the width of each frame-sync pulse on FSG.

To get this result	Use this bit
Enable receive clock mode with digital loop-back (DLB) mode not set (DLB - 1)	Set CLKRM (bit 7 in PCR) to 0.
Enable receive clock mode with DLB mode not set (DLB - 1)	Set CLKRM (bit 7 in PCR) to 0.
Enable DLB mode	Set DLB (bit 15 in SPCR1) to 1
Disable DLB	Set DLB (bit 15 in SPCR1) to 0
Enable clock stop mode with clock delay	Set CLKSTP (bits 12:11 in SPCR1) to 11b.
Enable clock stop mode without clock delay	Set CLKSTP (bits 12:11 in SPCR1) to 10b.
Disable clock stop mode (normal clocking for non-SPI mode)	Set CLKSTP (bits 12:11 in SPCR1) to 0Xb.

#### 3.1.14.2 Set the Receive Clock Mode

#### 3.1.14.3 Selecting a Source for the Receive Clock and a Data Direction for the MCLKRA Pin

Table 3–9 shows how you may select various sources to provide the receive clock signal and the effect on the MCLKRA pin. The polarity of the signal on the MCLKRA pin is determined by the CLKRP bit.

Note that in the digital loop back mode (DLB = 1), the transmit clock signal is 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 3-9. Select Sources to Provide the Receive Clock Signal and the Effect on the MCLKRA Pin

DLB in	CLKRM in	Source of Becsive Clock	MCI KBA Din Status
	i on	Source of Necerve Olock	
0	0	The MCLKRA 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 MCLKRA pin.

 Table 3-9. Select Sources to Provide the Receive Clock Signal and the Effect on the MCLKRA Pin (Continued)

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

Receive frame-sync pulses can be either generated internally by the sample rate generator (see section 1.4.2 on page 1-30) or driven by an external source. The source of frame sync is selected by programming the mode bit, FSRM, in PCR. MFSRA is also affected by the GSYNC bit in SRGR2. For information about the effects of FSRM and GSYNC, see *Set the Receive Frame-Sync Mode* on page 3-15. Similarly, receive clocks can be selected to be inputs or outputs by programming the mode bit, CLKRM, in the PCR.

When MFSRA and MFSXA 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 MDRA 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 MFSRA and MFSXA 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 MDXA 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 MFSRA, MFSXA, MCLKRA, and MCLKXA 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 (MFSRA/MFSXA 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 (MFSRA/MFSXA 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 MFSRA/MFSXA 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 MCLKXA is an input), the external falling-edge triggered input clock on MCLKXA is inverted to a rising-edge triggered clock before being sent to the transmitter. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and MCLKXA is an output pin), the internal (rising-edge triggered) clock, internal CLKX, is inverted before being sent out on the MCLKXA 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 MCLKRA is an input pin), the external rising-edge triggered input clock on MCLKRA 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 MCLKRA 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 3–8 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 3-8. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge



#### 3.1.14.4 Set the Receive Clock Polarity

To get this result	Use this bit
Sample receive data on falling edge of MCLKRA	Set CLRP (bit 0 in PCR) to 0
Sample receive data on rising edge of MCLKRA	Set CLRP (bit 0 in PCR) to 1

## 3.1.14.5 Set the SRG Clock Divide-Down Value

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.

То	get this result …	Use this bit
Set the divide down value of the sample rate generator clock		Set CLKGDV (bits 7-0 of SRGR1) to generate the required SRG clock frequency. The default value is 1 (divide input clock by 2).
Set	the SRG clock synchronization mode	Set GYSYNC (bit 15 of SRGR2)
The sample rate generator clock (CLKG) is free running. CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.		to 0
Clock synchronization is performed. When a pulse is detected on the MFSRA pin:		Set GYSYNC (bit 15 of SRGR2) to 1
CLKG is adjusted as necessary so that it is synchronized with the input clock on the MCLKRA or MCLKXA pin.		
	FSG pulses. FSG <b>only</b> pulses in response to a pulse on the MFSRA pin. The frame-sync period defined in FPER is ignored.	

GSYNC is used only when the input clock source for the sample rate generator is external—on the MCLKRA or MCLKXA pin.

For more details on using the clock synchronization feature, see *Synchronizing Sample Rate Generator Outputs to an External Clock* on page 1-31.

## 3.1.14.6 Set the SRG Clock Mode (Choose an Input Clock)

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. For more details on generating CLKG, see *Clock Generation in the Sample Rate Generator* on page 1-27.

To get this result	Use this bit
Choose sample rate generator clock derived from CLKS pin	Set SCLKME (bit 7 of PCR) and CLKSM (bit 13 of SRGR2) both to 0.
Choose sample rate generator clock derived from CPU clock. (This is the condition forced by a DSP reset.)	Set SCLKME = 0 and CLKSM = 1.

To get this result …	Use this bit…
Choose sample rate generator clock derived from MCLKRA pin	Set SCLKME = 1 and CLKSM = 0.
Choose sample rate generator clock derived from MCLKXA pin	Set SCLKME = 1 and CLKSM = 1.

#### 3.1.14.7 Set the SRG Input Clock Polarity

To get this result	Use this bit
CLKXP determines the input clock polarity when the MCLKXA pin supplies the input clock (SCLKME = 1 and CLKSM = 1).	Set CLKXP (bit 1 in the PCR register) to 0.
Rising edge on MCLKXA pin generates transitions on CLKG and FSG.	
Falling edge on MCLKXA pin generates transitions on CLKG and FSG.	Set CLKXP to 1.
MCLKRA Pin Polarity	Set CLKRP to 0.
CLKRP determines the input clock polarity when the MCLKRA pin supplies the input clock (SCLKME = 1 and CLKSM = 0).	
Falling edge on MCLKRA pin generates transitions on CLKG and FSG.	
Rising edge on MCLKRA pin generates transitions on CLKG and FSG.	Set CLKRP to 1.

#### 3.1.14.8 Using 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, MCLKXA, or MCLKRA pin. If you use a pin, choose a polarity for that pin by using the appropriate polarity bit (CLKXP for the MCLKXA pin and CLKRP for the MCLKRA pin). The polarity determines whether the rising or falling edge of the input clock generates transitions on CLKG and FSG.

# 3.2 Transmitter Configuration

To configure the McBSP transmitter, perform the following procedure:

- 1) Place the McBSP/transmitter in reset (see section 3.2.2).
- 2) Program the McBSP registers for the desired transmitter operation (see 3.2.1).
- 3) Take the transmitter out of reset (see section 3.2.2).

## 3.2.1 Programming the McBSP Registers for the Desired Transmitter Operation

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.

#### 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
- Enable/disable the A-bis mode

#### 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

## 3.2.2 Resetting and Enabling the Transmitter

The first step of the transmitter configuration procedure is to reset the transmitter, and the last step is to enable the transmitter (to take it out of reset).

To get this result	Use this bit
Enable the serial port transmitter	XRST (bit 0 of the SPCR2 register) set to 1
Disable the serial port transmitter (the reset state)	XRST set to 0
Reset the sample-rate generator	GRST (bit 6 of the SPCR2 register) set to 0
Enable the sample-rate generator	GRST set to 1
Reset the frame-sync logic	FRST (bit 7 of the SPCR2 register) = 0
Enable the frame-sync logic	FRST = 1

#### 3.2.2.1 Reset Considerations

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

- A DSP reset (RESET signal driven low) places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed (RESET signal released), 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 3-10 shows the state of McBSP pins when the serial port is reset due to a DSP reset and a direct receiver/transmitter reset.

Pin	Possible State(s)	State Forced By DSP Reset	State Forced By Receiver/Transmitter Reset
			Receiver Reset (RRST = 0 and GRST = 1)
MDRA	I	Input	Input
MCLKRA	I/O/Z	Input	Known state if Input; MCLKRA running if output
MFSRA	I/O/Z	Input	Known state if Input; FSRP inactive state if output
			Transmitter Reset (XRST = 0 and GRST = 1)
MDXA	O/Z	High impedance	High impedance
MCLKXA	I/O/Z	Input	Known state if Input; MCLKXA running if output
MFSXA	I/O/Z	Input	Known state if Input; FSXP inactive state if output

Table 3-10. Reset State of Each McBSP Pin

For more details about McBSP reset conditions and effects, see *Resetting and Initializing a McBSP* on page 4-3.

## 3.2.3 Set the Transmitter Pins to Operate as McBSP Pins

To configure a pin for its McBSP function, you should configure the bits of the GPxMUXn register appropriately. In addition to this, bits 12 and 13 of the PCR register must be set to 0. These bits are defined as reserved.

#### 3.2.4 Enable/Disable 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 3–11. This mode allows testing of serial port code with a single DSP device; the McBSP receives the data it transmits.

Table 3-11. Receive Signals Connected to Ti	Fransmit Signals in Digital Loopback Mode
---	---

	Is fed internally by
I his receive signal	this transmit signal
MDRA (receive data)	MDXA (transmit data)
MFSRA (receive frame synchronization)	MFSXA (transmit frame synchronization)
MCLKRA (receive clock)	MCLKXA (transmit clock)

The DLB bit determines whether the digital loopback mode is on.

SPRU061D

To get this result	Set this bit
Disable the digital-loopback mode	DLB (bit 15 of the SPCR1 register)
Enable the digital-loopback mode	DLB (bit 15 of the SPCR1 register)

## 3.2.5 Enable/Disable the Clock Stop Mode

The CLKSTP bits determine whether the clock stop mode is on.

To get this result	Set this bit
Disable clock stop mode (normal clocking for nonSPI mode)	CLKSTP (bits 11 and 12 of the SPCR1 register) set to 0Xb
Enable clock stop mode without clock delay	CLKSTP (bits 11 and 12 of the SPCR1 register)set to 10b
Enable clock stop mode with clock delay	CLKSTP (bits 11 and 12 of the SPCR1 register) set to 11b

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 MCLKXA 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 MCLKRA pin.

Table 3-12 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.

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 MCLKXA and receives data on the falling edge of MCLKRA.
CLKSTP = 11b CLKXP = 0 CLKRP = 1	Low inactive state with delay: The McBSP transmits data one-half cycle ahead of the rising edge of MCLKXA and receives data on the rising edge of MCLKRA.
CLKSTP = 10b CLKXP = 1 CLKRP = 0	High inactive state without delay: The McBSP transmits data on the falling edge of MCLKXA and receives data on the rising edge of MCLKRA.
CLKSTP = 11b CLKXP = 1 CLKRP = 1	High inactive state with delay: The McBSP transmits data one-half cycle ahead of the falling edge of MCLKXA and receives data on the falling edge of MCLKRA.

## Table 3-12. Effects of CLKSTP, CLKXP, and CLKRP on the Clock Scheme

# 3.2.6 Enable/Disable Transmit Multichannel Selection

To get this result	Use this bit …
Enable all channels	XMCM (bits 0 and 1 of the MCR2 register) to 00 (No channels can be disabled or masked with this setting.)
Disable all channels unless they are selected in the appropriate transit channel enable registers (XCERs).	Set XMCM to 01b. A channel is in this mode is also unmasked.
Enable all channels as masked unless they are selected in the appropriate transmit channel enable registers (XCERs).	Set XMCM to 10b.
Enable channels for symmetric transmissions and reception.	Set XMCM to 11b.

For more details, see Transmit Multichannel Selection Modes on page 2-9.

## 3.2.7 Enable/Disable the A-bis Mode

The ABIS bit determines whether the A-bis mode is on.

To get this result	Use this bit
Enable A-bis mode	ABIS (bit 6 in SPCR1) set to 1
Disable A-bis mode	ABIS set to 0

For more details, see A-bis Mode on page 2-14.

## 3.2.8 Choose 1 or 2 Phases for the Transmit Frame

To get this result	Use this bit…
Set transmit frame to single phase	XPHASE (bit 15 in XCR2 register set to 0.
Set transmit frame to dual-phase frame	XPHASE bit in XCR2 set to 1.

# 3.2.9 Set the Transmit Word Length(s)

To get this result	Use this b	it
Specify the length of every transmit word in Frame Phase 1	Set XWDLEN1 and XWDLEN2 (bit 7:5 of XCR1 XCR2 registers, respectively) to one of the following	
Transmit Word Length of Frame Phase 1	000b	8 bits
	001b	12 bits
	010b	16 bits
	011b	20 bits
	100b	24 bits
	101b	32 bits
	11xb	Reserved

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, 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.

## 3.2.10 Set the Transmit 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 3–13 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.

Table 3-13. How to Calculate Frame Length

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

**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.

To get this result	Use this bit
Set transmit frame length 1	Set XFRLEN1 (bits 14:8 of XRCR1) to one of the following:
	000 0000 1 word in phase 1
	000 0001 2 words in phase 1
	111 1111 128 words in phase 1
Set transmit frame length 2	Set XFRLEN2 (bits 14:8 of XRCR2) to one of the following:
	000 0000 1 word in phase 1
	000 0001 2 words in phase 1
	111 1111 128 words in phase 1

## 3.2.10.1 Enable/Disable the Transmit 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

See section 5.5 for XINTM interrupt selection into FIFO logic.

If there is an unexpected receive frame-sync pulse …	Use this bit
Allow an unexpected transmit frame sync pulse to restart the frame transfer	Set XFIG (bit 2 in XCR2) to 0
Ignore the unexpected pulses	Set XFIG to 1

#### 3.2.10.2 Examples Showing the Effects of XFIG

Figure 3–9 shows an example in which word B is interrupted by an unexpected frame-sync pulse when (R/X)FIG = 0. In the case of transmission, the transmission of B is aborted (B is lost). This condition is a transmit synchronization error, and thus sets the XSYNCERR bit. No new data has been written to DXR[1,2], and therefore, the McBSP transmits B again.

Figure 3-9. Unexpected Frame-Sync Pulse With (R/X)FIG = 0



In contrast with Figure 3–9, Figure 3–10 shows McBSP operation when unexpected frame-sync signals are ignored (when (R/X)FIG = 1). Here, the transfer of word B is not affected by an unexpected frame-sync pulse.



Figure 3-10. Unexpected Frame-Sync Pulse With (R/X)FIG = 1

## 3.2.11 Set the Transmit Companding Mode

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

A-law and  $\mu$ -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 FIFO must be at least 16 bits wide.

The  $\mu$ -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 0, indicating an 8-bit wide serial data stream. If companding is enabled and either of the frame phases does not have an 8-bit word length, companding continues as if the word length is 8 bits.

Figure 3-11 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  $\mu$ -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 3-11. Companding Processes for Reception and for Transmission



For transmission using  $\mu$ -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 3–12.

## Figure 3-12. µ-Law Transmit Data Companding Format



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 3–13.

## Figure 3-13. A-Law Transmit Data Companding Format



If the McBSP is otherwise unused (the serial port transmit and receive sections are reset), the companding hardware can compand internal data. See section 1.3.2.2 on page 1-13.

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

To get this result	Use this bit…	
Transmit companding mode	Set XCOMPAND (bits 4:3 in XCR2) to one of the following:	
	00	No companding, any size data, MSB transmitted first
	01	No companding, 8-bit data, LSB transmitted first
	10	μ-law companding, 8-bit data, MSB received first
	11	A-law companding, 8-bit data, MSB received first

## 3.2.12 Set the Transmit 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 Figure 3-14. 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 3-14. Range of Programmable Data Delay



#### 3.2.12.1 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 MCLKRA 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 MDXA. The transmitter then asynchronously detects the frame synchronization, MFSXA, going active high, and immediately starts driving the first bit to be transmitted on the MDXA pin.

#### 3.2.12.2 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 3-15. 2-Bit Data Delay Used to Skip a Framing Bit

## 3.2.13 Set the Transmit DXENA Mode

The DXENA bit controls the delay enabler on the MDXA pin. Set DXENA to enable an extra delay for turn-on time (for the length of the delay, see the data sheet for your device. Note that this bit does not control the data itself, so only the first bit is delayed, unless the A-bis mode is on. In the A-bis mode, any bit can be delayed because any bit can go from the high-impedance state to the valid state.

If you tie together the MDXA 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 two figures show the timing of the MDXA pin for DXENA = 1. The first figure shows the effect of the MDXA delay enabler when the A-bis mode is off. The second figure shows the effect of the MDXA delay enabler when the A-bis mode is on.



### 3.2.14 Set the Transmit Interrupt Mode

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. For more information on using XSYNCERR, see *Unexpected Transmit Frame-Sync Pulse* on page 1-46.

To get this result	Use	this bit	
Set the transmit interrupt mode.	Set	Set XINTM (bits 5:4 in SPCR2 to:	
	00	XINT generated when XRDY changes from 0 to 1	
	01	XINT generated by an end-of-block or end-of-frame condition in the transmit multichannel selection mode	
	10	XINT generated by a new transmit frame-sync pulse	
	11	XINT generated when XSYNCERR is set	

## 3.2.15 Set the Transmit Frame-Sync Mode

Table 3-14 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 3-14 also shows the effect of each bit setting on the MFSXA pin. The polarity of the signal on the MFSXA pin is determined by the FSXP bit.

Table 3-14. How FSXM and FSGM Select the Source of Transmit Frame-Sync Pulses

FSXM	FSGM	Source of Transmit Frame Synchronization	MFSXA Pin Status
0	0 or 1	An external frame-sync signal enters the McBSP through the MFSXA 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 MFSXA 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 MFSXA pin.

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 MFSRA pin. For more details, see *Synchronizing Sample Rate Generator Outputs to an External Clock* on page 1-31.

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 (SPISTE) on the MFSXA pin, make sure that FSXM = 1 and FSGM = 0, so that MFSXA 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 MFSXA pin.

To get this result	Use this bit…
To supply transmit frame synchronization mode by an external source via the MFSXA pin	Set FSXM (bit 11 in PCR) to 0.
To have the McBSP supply the transmit frame synchronization mode by way of the FSGM bit of SRGR2.	Set FSXM to 1.
Have the McBSP generate a transmit frame-sync pulse when the content of DXR is copied to XSR	Set FSGM (bit 12 of SRGR2) to 0. This mode is used when FSXM = 1.
Use frame-sync pulses generated by the sample rate generator	Set FSGM (bit 12 of SRGR2) to 1.

#### 3.2.16 Set the Transmit Frame-Sync Polarity

Transmit frame-sync pulses can be either generated internally by the sample rate generator (see section 1.4.2 on page 1-30) or driven by an external source. The source of frame sync is selected by programming the mode bit, FSXM, in PCR. MFSXA is also affected by the FSGM bit in SRGR2. For information about the effects of FSXM and FSGM, see *Set the Transmit Frame-Sync Mode* on page 3-40). Similarly, transmit clocks can be selected to be inputs or outputs by programming the mode bit, CLKXM, in the PCR (see *Set the Transmit Clock Mode* on page 3-44).

When MFSRA and MFSXA 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 MDRA 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 MFSRA and MFSXA 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 MDXA 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 MFSRA, MFSXA, MCLKRA, and MCLKXA 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 (MFSRA/MFSXA 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 (MFSRA/MFSXA 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 MCLKXA is an input), the external falling-edge triggered input clock on MCLKXA is inverted to a rising-edge triggered clock before being sent to the transmitter. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and MCLKXA is an output pin), the internal (rising-edge triggered) clock, internal CLKX, is inverted before being sent out on the MCLKXA 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 MCLKRA is an input pin), the external rising-edge triggered input clock on MCLKRA 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 MCLKRA 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 3-18 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.



## 3.2.17 Set the SRG Frame-Sync Period and Pulse Width

To get this result	Use this bit…
Set the 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.
	Set FPER (bits 11-0 in the SRGR2 register) from 1 to 4096 cycles.
Set the sample rate generator frame-sync pulse width	FWID (bits 15-8 in SRGR1) plus 1 determines the width of each frame-sync pulse (from 1 to 256 CLKG cycles).

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 3–19 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 3-19. Frame of Period 16 CLKG Periods and Active Width of 2 CLKG Periods



When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when  $\overline{FRST} = 1$  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.

## 3.2.18 Set the Transmit Clock Mode

To get this result	Use this bit…
Set the transmitter to get its clock signal from an external source via the MCLKXA pin.	Set CLKXM (bit 9 in the PCR register) to 0
Set the MCLKXA output pin driven by the sample rate generator	Set CLKXM (bit 9 in the PCR register) to 1

## 3.2.18.1 Selecting a Source for the Transmit Clock and a Data Direction for the MCLKXA Pin

Table 3–15 shows how the CLKXM bit selects the transmit clock and the corresponding status of the MCLKXA pin. The polarity of the signal on the MCLKXA pin is determined by the CLKXP bit.

CLKXM in PCR	Source of Transmit Clock	MCLKXA Pin Status
0	Internal CLKX is driven by an external clock on the MCLKXA pin. MCLKXA 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 MCLKXA.

Table 3-15.	How the CLKXM Bit Selects the Transmit Clock and the Corresponding
	Status of the MCLKXA Pin

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 MFSRA pin. For more details, see *Synchronizing Sample Rate Generator Outputs to an External Clock* on page 1-31.

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 MCLKXA 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 MCLKXA is an input to accept the master clock signal.

#### 3.2.18.2 Set the Transmit Clock Polarity

To get this result	Use this bit…
Sample transmit data on the rising edge of MCLKXA	Set CLKXP (bit 1 of PCR) to 0.
Sample transmit data on the falling edge of MCLKXA	Set CLKXP (bit 1 of PCR) to 1.

Transmit frame-sync pulses can be either generated internally by the sample rate generator (see section 1.4.2 on page 1-30) or driven by an external source. The source of frame sync is selected by programming the mode bit, FSXM, in PCR. MFSXA is also affected by the FSGM bit in SRGR2. For information about the effects of FSXM and FSGM, see *Set the Transmit Frame-Sync Mode*. Similarly, transmit clocks can be selected to be inputs or outputs by programming the mode bit, CLKXM, in the PCR (see *Set the Transmit Clock Mode* on page 3-44).

When MFSRA and MFSXA 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 MDRA pin is also sampled on the falling edge of internal MCLKRA. Note that these internal clock signals are either derived from external source via MCLKRA/MCLKXA pins or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When MFSRA and MFSXA 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 MDXA 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 MFSRA, MFSXA, MCLKRA, and MCLKXA 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 (MFSRA/MFSXA 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 MFSXA). Similarly, if internal synchronization (MFSRA/MFSXA 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 MFSRA/MFSXA 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 MCLKXA is an input), the external falling-edge triggered input clock on MCLKXA is inverted to a rising-edge triggered clock before being sent to the transmitter. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and MCLKXA is an output pin), the internal (rising-edge triggered) clock, internal CLKX, is inverted before being sent out on the MCLKXA 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 MCLKRA is an input pin), the external rising-edge triggered input clock on MCLKRA 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 MCLKRA 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 3-20 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 3-20. Data Clocked Externally Using a Rising Edge and Sampled by the McBSP Receiver on a Falling Edge

## 3.2.19 Set the SRG Clock Divide-Down Value

To get this result	Use this bit…
Set the divide down value of the sample rate generator clock	Set CLKGDV (bits 7-0 of SRGR1) to generate the required SRG clock frequency. The default value is 1 (divide input clock by 2).

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.

## 3.2.19.1 Set the SRG Clock Synchronization Mode

GSYNC is used only when the input clock source for the sample rate generator is external—on the MCLKRA or MCLKXA pin.

For more details on using the clock synchronization feature, see *Synchronizing Sample Rate Generator Outputs to an External Clock* on page 1-31.

То	get this result	Use this bit
The sample rate generator clock (CLKG) is free running. CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.		Set GSYNC (bit 15 of SRGR2) to 0
Clo Wh pin:	ck synchronization is performed. en a pulse is detected on the MFSRA :	Set GSYNC (bit 15 of SRGR2) to 1
	CLKG is adjusted as necessary so that it is synchronized with the input clock on the MCLKRA or MCLKXA pin.	
	FSG pulses. FSG <b>only</b> pulses in response to a pulse on the MFSRA pin. The frame-sync period defined in FPER is ignored.	

## 3.2.20 Set the SRG Clock Mode (Choose an Input Clock)

To get this result	Use this bit…
Sample rate generator clock derived from CLKS pin	Set SCLKME (bit 7 of PCR) = 0 and CLKSM (bit 13 of SRGR2) = 0
Sample rate generator clock derived from CPU clock (This is the condition forced by a DSP reset.)	SCLKME = 0 CLKSM = 1
Sample rate generator clock derived from MCLKRA pin	SCLKME = 1 CLKSM = 0
Sample rate generator clock derived from MCLKXA pin	SCLKME = 1 CLKSM = 1

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. For more details on generating CLKG, see *Clock Generation in the Sample Rate Generator* on page 1-27.

## 3.2.20.1 Set the SRG Input Clock Polarity

CLKSP determines the input clock polarity when the CLKS pin supplies the input clock (SCLKME = 0 and CLKSM = 0).

To get this result	Use this bit…
Rising edge on CLKS pin generates CLKG and FSG.	Set CLKSP (bit 14 of SRGR2) to 0
Falling edge on CLKS pin generates CLKG and FSG.	Set CLKSP (bit 14 of SRGR2) to 1

CLKXP determines the input clock polarity when the MCLKXA pin supplies the input clock (SCLKME = 1 and CLKSM = 1).

To get this result	Use this bit…
Rising edge on MCLKXA pin generates transitions on CLKG and FSG.	Set CLKXP (bit 1 of PCR) to 0
Falling edge on MCLKXA pin generates transitions on CLKG and FSG.	Set CLKXP to 1

CLKRP determines the input clock polarity when the MCLKRA pin supplies the input clock (SCLKME = 1 and CLKSM = 0).

To get this result	Use this bit…
Rising edge on MCLKRA pin generates transitions on CLKG and FSG.	Set CLKRP (bit 0 of PCR) to 1
Falling edge on MCLKRA pin generates transitions on CLKG and FSG.	Set CLKRP to 0

## 3.2.20.2 Using MCLKXAP/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 MCLKXA, or MCLKRA pin. If you use a pin, choose a polarity for that pin by using the appropriate polarity bit (CLKXP for the MCLKXA pin, CLKRP for the MCLKRA pin). The polarity determines whether the rising or falling edge of the input clock generates transitions on CLKG and FSG.

# Chapter 4

# **Emulation and Reset Considerations**

This section covers emulation mode and how to reset and initialize the various parts of the McBSP.

# Topic

# Page

4.1	McBSP Emulation Mode 4-2
4.2	Data Packing Examples 4-8
4.3	GPIO Function 4-11

# 4.1 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 transfer, and then the clock halts. These options are listed in the following table.

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.

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.

Table 4-1. McBSP Emulation Modes Selectable With the FREE and SOFT Bits of SPCR2

## 4.1.1 Resetting and Initializing a McBSP

Table 4–2 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.

Pin	Possible State(s)	State Forced By DSP Reset	State Forced By Receiver/Transmitter Reset
			Receiver Reset (RRST = 0 and $\overline{\text{GRST}}$ = 1)
MDRA	I	Input	Input
MCLKRA	I/O/Z	Input	Known state if Input; MCLKRA running if output
MFSRA	I/O/Z	Input	Known state if Input; FSRP inactive state if output
			Transmitter Reset ( $\overline{XRST} = 0$ and $\overline{GRST} = 1$ )
MDXA	O/Z	High impedance	High impedance
MCLKXA	I/O/Z	Input	Known state if Input; MCLKXA running if output
MFSXA	I/O/Z	Input	Known state if Input; FSXP inactive state if output

Table 4-2. Reset State of Each McBSP Pin

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

In the 28x family, at device reset, all I/Os default to GPIO function and generally as inputs.

When the McBSP is reset in either of the above two ways, the machine is reset to its initial state, including reset of all counters and status bits. The receive status bits include RFULL, RRDY, and RSYNCERR. The transmit status bits include  $\overline{XEMPTY}$ , XRDY, and XSYNCERR.

DSP reset. When the whole DSP is reset (XRS 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 MDXA is in the high-impedance state.

The DSP reset forces the sample rate generator clock, CLKG, to have the frequency of LSPCLK 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 MDRA and MDXA pins may be used as general-purpose I/O pins as described in section 4.3 on page 4-11.

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 MDRA, and all other pins that are configured as inputs, are in a known state. The MFSRA and MFSXA pins are driven to their inactive state if they are not outputs. If the MCLKRA and MCLKXA pins are programmed as outputs, they will be driven by CLKG, provided that GRST = 1. Lastly, the MDXA 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 ( $\overline{GRST} = 1$ ), pins MFSRA and MFSXA are in an inactive state when  $\overline{RRST} = 0$  and  $\overline{XRST} = 0$ , 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  $\overline{FRST} = 1$  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 LSPCLK, 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  $\overline{GRST}$ . In this case, CLKG and FSG are driven inactive low. If you then set  $\overline{GRST}$ , CLKG starts and runs as programmed. Later, if  $\overline{FRST} = 1$ , FSG pulses active high after the programmed number of CLKG cycles has elapsed.

## 4.1.1.1 McBSP Initialization Procedure

The serial port initialization procedure is as follows:

- 1) Make XRST = RRST = FRST = 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 up data acquisition as required (such as writing to DXR[1,2]).
- 5) Make  $\overline{\text{XRST}} = \overline{\text{RRST}} = 1$  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.
- 6) Set  $\overline{\text{FRST}} = 1$ , if internally generated frame synchronization is required.
- 7) 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 MCLKRA/MCLKXA 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 or FIFO only when the transmitter is enabled (XRST = 1). An exception to this rule is when these registers are used for companding internal data (see section 1.3.2.2 on page 1-13).
- 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.

#### 4.1.1.2 Example: Resetting the Transmitter While the Receiver is Running

The following example shows values in the control registers that reset and configure the transmitter while the receiver is running.

## Example 4-1. Resetting and Configuring the McBSP Transmitter While the McBSP Receiver is Running - Non-FIFO Mode

SPCR1 = 0001h SPCR2 = 0030h	<pre>; The receiver is running with the receive ; interrupt (RINT) triggered by the ; receiver ready bit (RRDY). The ; transmitter is in its reset state. The ; transmit interrupt (XINT) will be ; triggered by the transmit frame-sync ; error bit (XSYNCERR).</pre>
PCR = 0900h	; Transmit frame synchronization is ; generated internally according to the ; FSGM bit of SRGR2. The transmit clock ; is driven by an external source. The ; receive clock continues to be driven by ; sample rate generator. The input clock ; of the sample rate generator is supplied ; by the CLKS pin or by the CPU clock ; depending on the CLKSM bit of SRGR2.
SRGR1 = 0001h SRGR2 = 2000h	; The CPU clock is the input clock for ; the sample rate generator. The sample ; rate generator divides the CPU clock by ; 2 to generate its output clock (CLKG). ; Transmit frame synchronization is tied ; to the automatic copying of data from ; the DXR(s) to the XSR(s).
XCR1 = 0740h XCR2 = 8321h	<pre>; The transmit frame has two phases. ; Phase 1 has eight 16-bit words. Phase 2 ; has four 12-bit words. There is 1-bit ; data delay between the start of a ; frame-sync pulse and the first data bit ; transmitted.</pre>
------------------------------	--
SPCR2 = 0x0031	; The transmitter is taken out of reset.

### 4.2 Data Packing Examples

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

#### 4.2.1 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 4–1. In this case:

- (R/X)PHASE = 0: Single-phase frame
- □ (R/X)FRLEN1 = 0000011b: 4-word frame
- $\square$  (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 FIFO. Thus, four reads from DRR1 and four writes to DXR1 are necessary for each frame.

Figure 4-1. 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 4–2. 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 FIFO. 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.





#### 4.2.2 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 4–3, 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 4-3. 8-Bit Data Words Transferred at Maximum Packet Frequency

Figure 4-4 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.





# 4.3 GPIO Function

#### Note: General-purpose I/O Selection

Do not use the GPIO function using RIOEN/XIOEN bits 12 and 13 in the PCR register. This feature is not applicable to the 28x McBSP implementation; therefore, these bits are reserved on 28x devices.

# Chapter 5

# **McBSP FIFO and Interrupts**

This chapter describes the FIFO interface logic and how interrupts are handled in the McBSP module.

## Topic

## Page

## 5.1 McBSP FIFO Overview

McBSP module is interfaced with a 16x16 bit (16 level) FIFO for each of the data registers, DRR2/DRR1 and DXR2/DXR1. The top of the FIFO registers share the same addresses as of the data registers in nonFIFO mode. Each of the FIFO data registers pair has a separate set of control registers. Table 5–1 lists the register details for easy reference.

Address Offset 0x000	16-Bit	Туре	Reset Value	
xxh	Access	<b>R/W</b> <sup>†</sup>	(Hex)	McBSP FIFO Registers
				FIFO mode registers applicable only for FIFO mode in 28xx implementation
00	DRR2	R	0x0000	McBSP Data Receive Register2 - Top of receive FIFO
				Read first FIFO index will not advance
01	DRR1	R	0x0000	McBSP Data Receive Register1-Top of receive FIFO
				Read second for FIFO index to advance
02	DXR2	W	0x0000	McBSP Data Transmit Register2-Top of transmit FIFO
				Write first FIFO index will not advance
03	DXR1	W	0x0000	McBSP Data Transmit Register1-Top of transmit FIFO
				Write second for FIFO index to advance
20	MFFTX	R/W	0xA000	McBSP transmit FIFO register
21	MFFRX	R/W	0x201F	McBSP receive FIFO register
22	MFFCT	R/W	0x0000	McBSP FIFO control register
23	MFFINT	R/W	0x0000	McBSP FIFO interrupt register
24	MFFST	R/W	0x0000	McBSP FIFO status register

Table 5-1. McBSP FIFO Registers

<sup>†</sup> R = read, W = write

This section describes the following requirements for the McBSP module in FIFO mode.

- 1) McBSP functionality and limitation under FIFO mode
- 2) McBSP FIFO operation
- 3) McBSP receive interrupt generation
- 4) McBSP transmit interrupt generation
- 5) McBSP FIFO access constraints
- 6) McBSP FIFO register description

# 5.2 McBSP Functionality and Limitation Under FIFO Mode

McBSP, in its normal mode, communicates with various types of Codecs with variable word size. This mode is common and is available while the FIFO is enabled. Apart from this mode, the McBSP uses time-division multiplexed (TDM) data stream while communicating with other McBSPs or serial devices. The multichannel mode provides flexibility while transmitting/receiving selected channels or all the channels in a TDM stream.

Table 5-2 provides a quick reference to McBSP mode selection and its limitation if any in FIFO mode.

		Registe	r Bits Tha	t Decide th	e Mode S		
	MaRSD	MCR1	bit 9,0	MCR2 b	oit 9,1,0	SPCR1-6	
No.	Word Size	RMCME	RMCM	XMCME	ХМСМ	A-bis	Description of the Mode and its Function
							Normal Mode
1	8/12/16/20/24/ 32 bit words	0	0	0	00	0	All types of Codec interface will use this selection
							Multichannel Mode
2	8-bit words						2 Partition or 32-channel mode
		0	1	0	01	0	All channels are disabled,unless selected in X/RCERA/B
		0	1	0	10	0	All channels are enabled,but masked unless selected in X/RCERA/B
		0	1	0	11	0	Symmetric transmit, receive
							8 Partition or 128 Channel Mode Transmit/ Receive
							Channels selected by X/RCERA to X/RCERH bits
							Multichannel Mode is ON
		1	1	1	01	0	All channels are disabled,unless selected in XCERs
		1	1	1	10	0	All channels are enabled,but masked unless selected in XCERs
		1	1	1	11	0	Symmetric transmit, receive
							Continuous Mode - Transmit
		1	0	1	00	0	Multi-Channel Mode is OFF
							All 128 channels are active and enabled
							A-bis Mode
4	16-bit words	0	0	0	0	1	A-bis mode. McBSP can receive /transmit up to 1024 bits on PCM link. Bit selection is provide by RCERA/B , XCERA/B registers only X/BCEBC-H are not used
							ATTOLITO-IT are not used

# Table 5-2. McBSP Mode Selection

#### 5.3 McBSP FIFO Operation

The McBSP powers up without the FIFOs enabled. The FIFO mode of the McBSP can be enabled using the FIFO-enable bits in FIFO register MFFTX. The following steps explain the FIFO features and help programming the McBSP with FIFOs.

- Reset. At reset the McBSP powers up in standard McBSP mode with the FIFOs features disabled. The FIFO registers MFFTX, MFFRX, and MFFCT will remain inactive.
- 2) NonFIFO mode. The McBSP without the FIFO registers is enabled. McBSP registers can be programmed to receive and transmit data through DRR2/DRR1 and DXR2/DXR1 registers, respectively. The CPU can directly access these registers to move data from memory to these registers. Interrupt signals will be based on these register pair contents and its related flags.
- FIFO mode. FIFOs are enabled by setting the MFFENA bit to 1 in the MFFTX register.
- 4) Active registers. All the McBSP registers and McBSP FIFO registers MFFTX, MFFRX (once FIFOs are enabled), and MFFCT will be active.
- 5) Interrupts. MRINT/MXINT will generate CPU interrupts for receive and transmit conditions of the FIFO. See the interrupt chapter for details. See the table interrupt section for details on these bit selection and interrupt functionality.
- 6) Buffers. Transmit and receive data registers of the McBSP will be supplemented with four 16x16 FIFOs. The receive data registers DRR2 and DRR1 will have a 16x16bit FIFO each. Receive channel buffer from MDRA pin will stack as shown below.

MDRA pin - RSR[1,2] - RBR[1,2] - DRR[1,2]+16x16 FIFO[1,2].

Similarly the transmit data register DXR2 and DXR1 will have a 16x16bit FIFO each. Transmit channel buffer up to MDXA pin will stack as shown below.

MDXA pin - XSR[1,2] - DXR[1,2]+ 16x16 FIF0[1,2]

7) Transmit FIFO to DXR2/DXR1 data transfer will be based on XEVT/XINT interrupt signal from McBSP and its related flags.

DRR2/DRR1 to Receive FIFO data register will be based on REVT/RINT interrupt signal from McBSP and its related flags.

- 8) FIFO status bits. Both transmit and receive FIFOs have status bits TXFFST (bit 12-0) or RXFFST (bits 12-0). They define the number of words available in the FIFOs at any time. The status bits are updated on every 2 word transfer, DXR2/DXR1 and DRR2/DRR1 register, respectively. The transmit FIFO reset bit TXFIFO and receive reset bit RXFIFO will reset the FIFO pointers to zero when these bits are set 1. The FIFOs will resume operation from start once these bits are cleared to zero.
- 9) Programmable interrupt levels. Both transmit and receive FIFO can generate CPU interrupts. The interrupt trigger is generated whenever the transmit FIFO status bits TXFFST (bits 12-8) match the interrupt trigger level bits TXFFIL (bits 4-0). This provides a programmable interrupt trigger for transmit and receive sections of the McBSP. Default value for these trigger level bits will be 0x11111 for receive FIFO and 0x00000 for transmit FIFO respectively.

## 5.4 McBSP Receive Interrupt Generation

In the McBSP module, data receive and error conditions generate two sets of interrupt signals. One set is used for the CPU and the other set is for DMA. Since there is no DMA implemented with this module, the DMA interrupt signals are used by the FIFO control logic. The FIFO control logic references these signal to move the data between the FIFO registers and the actual receive registers. The following section maps the interrupt signal selection in FIFO mode and nonFIFO mode for the receive channel.





Modes	McBSP Interrupt Signal	Interrupt Flags	Interrupt Enables in SPCR1	Interrupt Enables	Type of Interrupt	Interrupt Line
NonFIFO Mode			RINTM Bits			
Receive	RINT	RRDY	00	RINTENA	Every word receive	MRINT
		EOBR	01	RINTENA	Every 16 channel block boundary	
		MFSRA	10	RINTENA	On every MFSRA	
		RSYNCERR	11	RINTENA	Frame sync error	
	REVTA		хх	REVTA ENA	A-bis mode update for CPU	
FIFO Mode						
Receive FIFO Interrupt		RXFFINT or RXFFOVF	xx	RXFFINT ENA	Transmit FIFO status interrupt	MRINT
	REVT				Not Used by CPU and FIFO	

Table 5-3. Receive Interrupt Sources and Signals in NonFIFO Mode and FIFO Mode

Note: x – Don't care

#### Note:

Since X/RINT, X/REVTA, and X/RXFFINT share the same CPU interrupt, it is recommended that all applications use one of the above selections for interrupt generation. If multiple interrupt enables are selected at the same time, there is a likelihood of interrupts being masked or not recognized.

# 5.5 McBSP Transmit Interrupt Generation

McBSP module data transmit and error conditions generate two sets of interrupt signals. One set is used for the CPU and the other set is for DMA. Since there is no DMA implemented with this module, the DMA interrupt signals are used by the FIFO control logic. The FIFO control logic will reference these signal to move the data between the FIFO registers and the actual receive registers. The following section maps the interrupt signal selection in FIFO mode and nonFIFO mode for the transmit channel.





Modes	McBSP Interrupt Signal	Interrupt Flags	Interrupt Enables in SPCR1	Interrupt Enables	Type of Interrupt	Interrupt Line
NonFIFO Mode			XINTM Bits			
Transmit	XINT	XRDY	00	XINTENA	Every word receive	MXINT
		EOBX	01	XINTENA	Every 16-channel block boundary	
		MFSXA	10	XINTENA	On every MFSXA	
		XSYNCERR	11	XINTENA	Frame sync error	
	XEVTA		XX	XEVTA ENA	A-bis mode update for CPU	
FIFO Mode						
Transmit FIFO Interrupt		TXFFINT	XX	TXFFINT ENA	Transmit FIFO status interrupt	MXINT
	XEVT				Not used by CPU	

Table 5-4. Transmit Interrupt Sources and Signals in NonFIFO Mode and FIFO Mode

Note: x – Don't care.

#### 5.5.1 FIFO Data Register Access Constraints

McBSP registers are allowed only 16-bit access via the peripheral bus. This includes the pair of data registers on receive and transmit channels. DRR2/DRR1 is the receive register pair and DXR2/DXR1 is the transmit register pair. Based on the word size selected (8/12/16/24/32 bit) on the receive side either the DRR2/DRR1 register pair is used or only the DRR1 is used. Similarly, on the transmit side either the DXR2/DXR1 register pair is used or only the DRR1 is used. Similarly, on the transmit side either the DXR2/DXR1 register pair is used or only the DXR1 is used. This is true in nonFIFO mode and FIFO mode of the McBSP module. The following tables explain the read/write order based on the word size. In the FIFO mode, there is only one set of FIFO pointer for both receive and transmit register pair. The FIFO pointer update is valid only if the listed order is maintained. This will guarantee the same data integrity as applicable in the nonFIFO mode of the McBSP.

	Normal/ Multichannel Mode	Receive FIFO Pointer	Normal Mode	Receive FIFO Pointer
	8/16 Bit Word	Read Order	24/32 Bit Word	l Read Order
Case1	DRR2	No change	DRR2	No change
	DRR1	Decrement	DRR1	Decrement
Case2	DRR2	No change	DRR2	No change
	DRR2, or n times	no change	DRR2 or n times	no change
	DRR1	Decrement	DRR1	Decrement
Case3	DRR1	Decrement	DRR1	no change
	DRR2	no change	DRR2	no change

Table 5-5. Receive FIFO Read Order

#### Table 5-6. Transmit FIFO Write Order

	Normal∕ Multichannel Mode	Receive FIFO Pointer	Normal Mode	Receive FIFO Pointer
	8/16 Bit Word	Write Order	24/32 Bit Word	d Write Order
Case1	DXR2	No change	DXR2	No change
	DXR1	Increment	DXR1	Increment
Case2	DXR2	No change	DXR2	No change
	DXR2, or n times	No change	DXR2 or n times	No change
	DXR1	Increment	DXR1	Increment
Case3	DXR1	Increment	DXR1	No change
	DXR2	No change	DXR2	No change

**Note:** DRR2/DRR1 and DXR2/DXR1 registers read and write constraints are native to the McBSP module. Hence maintaining these constraints in FIFO mode will leave no additional conflicts.

#### 5.5.2 FIFO Error Flags

The McBSP has several error flags both on receive and transmit channel. In FIFO mode these bits either made non functional or equivalent flags are provided. Table 5-7 explains the error flags and their new meaning in FIFO mode

Error Flags	Function in McBSP, NonFIFO Mode	Function in McBSP, FIFO Mode
RFULL	Indicates DRR2/DRR1 are not read and RXR register is overwritten	This bit will never get set, as the DRR2/DRR1 values are read into the FIFO registers. Use RXFFOVF instead for error condition.
RXFFOVF	Not applicable in nonFIFO mode	FIFO receive overflow flag that will be set whenever the receive FIFO is overflows. The data at the top (or the FIRST IN data) of the FIFO will be lost.
RSYNCERR	Indicates unexpected frame-sync condition, current data reception will abort	Indicates unexpected frame-sync condition, current data reception will abort and restart
	and restart. Use RINTM bit 11 for interrupt generation on this condition.	Use RINTM bits 11 for interrupt generation on this conditions.
XSYNCERR	Indicates unexpected frame-sync condition, current data transmission will abort and restart. Use XINTM bit 11 for interrupt generation on this condition.	Indicates unexpected frame-sync condition, current data transmission will abort and restart. Use RINTM bits 11 for interrupt generation on this condition.

Table 5-7. McBSP Error Flags

#### 5.5.3 McBSP IDLE Mode

Internal IDLE mode of the McBSP module is not to be implemented. Bit 14 in PCR register for IDLE\_EN will be a read/write bit.

#### 5.5.4 McBSP Reset Conditions

The McBSP can be reset either through hardware DSP reset or reset control on receive or transmit channel. The receive channel and transmit channel resets are controlled by RRST (bit 0 in SPCR1) and XRST (bit 0 in SPCR2) bits The McBSP module native reset state of the McBSP pins, register status bits will remain as is in both FIFO nonFIFO modes.

However, the FIFO registers have additional register bits that will control of the reset state of the FIFO pointers and register contents. RXFIFO and TXFIFO reset bits in MFFRX (bit 13) and MFFTX (bit 13) reset the FIFO pointers to zero and power up in reset state.

# 5.6 McBSP FIFO Register Descriptions

15		14		1	3	12	11	10	9	8		
Reser	ved	MFFENA		TXF Re	IFO set	TXFFST4	TXFFST3	TXFFST2	TXFFST1	TXFFST0		
R-0	)	R/W-0		R/V	V-1	R-0	R-0	R-0	R-0	R-0		
7		6		5	5	4	3	2	1	0		
TXFFI Flag	INT g	TXFFINT Clear		TXFF	IENA	TXFFIL4	TXFFIL3	TXFFIL2	TXFFIL1	TXFFIL0		
R-0	)	W-0		R/V	V-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
Legend:	R = Re	ad only; W	= V	Vrite on	y; R/W	= Read/Write; -r	n = Reset Value					
Bit(s)	Nam	e	D	escrip	tion							
15	Rese	erved	R	eserved	ł							
14	MFFI	ENA	0		McBS	P FIFO enhar	cements are	disabled and F	IFO is in rese	t.		
			1		McBSF	P FIFO enhance	ments are enab	oled.				
13	TXFI	FO Reset	0		Write	0 to reset the	FIFO pointer t	o zero, and ho	old in reset.			
			1		Re-enable Transmit FIFO operation							
12-8	TXFF	ST4-0	0	0000	Transmit FIFO is empty.							
			0	0001	1 Transmit FIFO has 1 word							
			0	0010	Transmit FIFO has 2 words							
			0	0011	Trans	Transmit FIFO has 3 words						
			0	xxxx	Trans	mit FIFO has :	k words					
			1(	0000	Trans	mit FIFO has	16 words					
7	TXFF	FINT	0		TXFIF	O interrupt ha	s not occurred	d. Read only b	it			
			1		TXFIF	O interrupt ha	s occurred. R	ead only bit				
6	TXFF	INT CLR	0		Write	0 has no effec	t on TXFIFIN	Γ flag bit, bit re	ads back a ze	ro		
			1	Write 1 to clear TXFFINT flag in bit 7								
5	TXFF	IENA	0		TX FII disabl	FO interrupt ba	ased on TXFF	IVL match (les	ss than or equ	al to) will be		
			1		TX FIFO interrupt based on TXFFIVL match (less than or equal to) will enabled.					al to) will be		
4-0	TXFF	FIL4-0	Т	XFFIL₄	1-0 Tra	nsmit FIFO in	terrupt level bi	ts				
			Transmit FIFO wi FIFO level bits (T			will generate i (TXFFIL4-0 )	nterrupt when match (less th	the FIFO stat nan or equal to	us bits (TXFFS ).	ST4-0) and		
			D	efault	value s	hould be 0x00	000					
Noto		and to be ME	20.	T bit on	d will bo	reconved in this		ntation				

Figure 5-3. McBSP FIFO Transmit Register (MFFTX)

Note: Bit 15 used to be MRST bit and will be reserved in this FIFO implementation

5-14 McBSP FIFO and Interrupts

15	14	13	12	11	10	9	8
RXFFOVF Flag	RXFFOVF Clear	RXFIFO Reset	RXFFST4	RXFFST3	RXFFST2	RXFFST1	RXFFST0
R-0	W-0	R/W-1	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
RXFFINT Flag	RXFFINT Clear	RXFFIENA	RXFFIL4	RXFFIL3	RXFFIL2	RXFFIL1	RXFFIL0
R-0	W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

Figure 5-4. McBSP FIFO Receive Register (MFFRX)

Legend: R = Read only; W = Write only; R/W = Read/Write; -n = Reset Value

Bit(s)	Name	Description					
15	RXFFOVF	0	Receive FIFO has not overflowed, read only bit.				
		1	Receive FIFO has overflowed, read only bit. More than 16 words have been received in to the FIFO, and the first received word is lost.				
14	RXFFOVFCLR	0	Write 0 has no effect on RXFFOVF flag bit. Bit reads back a zero				
		1	Write 1 to clear RXFFOVF flag in bit 15				
13	RXFIFO Reset	0	Write 0 to reset the FIFO pointer to zero, and hold in reset.				
		1	Re-enable Transmit FIFO operation				
12-8	RXFFST4-0	00000	Receive FIFO is empty.				
		00001	Receive FIFO has 1 word				
		00010	Receive FIFO has 2 words				
		00011	Receive FIFO has 3 words				
		0xxxx	Receive FIFO has x words				
		10000	Receive FIFO has 16 words.				
7	RXFFINT	0	RXFIFO interrupt has not occurred. Read only bit				
		1	RXFIFO interrupt has occurred. Read only bit				
6	RXFFINT CLR	0	Write 0 has no effect on RXFIFINT flag bit, Bit reads back a zero				
		1	Write 1 to clear RXFFINT flag in bit 7				
5	RXFFIENA	0	RX FIFO interrupt based on RXFFIVL match (less than or equal to) will be disabled				
		1	RX FIFO interrupt based on RXFFIVL match (less than or equal to) will be enabled.				
4-0	RXFFIL0 -4	RXFFIL	4-0 Receive FIFO interrupt level bits				
		Receive FIFO lev bits afte FIFO wi	FIFO will generate interrupt when the FIFO status bits (RXFFST4-0) and vel bits (RXFFIL4-0) match (greater than or equal to) default value of these r reset - 11111. This will avoid frequent interrupts, after reset, as the receive II be empty most of the time.				

SPRU061D

# Figure 5-5. McBSP FIFO Control Register (MFFCT)

15		14							8		
IACK	M					Reserved					
R/W-0						R-0					
7		6	Ę	5	4	3	2	1	0		
FFTXD	LY7	FFTXDLY6	FFTX	DLY5	FFTXDLY4	FFTXDLY3	FFTXDLY2	FFTXDLY1	FFTXDLY0		
R/W	-0	R/W-0	R/V	V-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
Legend:	Legend: R = Read only; R/W = Read/Write										
Bit(s)	Nam	ne	Reset	Desci	ription						
15	IACI	KM	0	Default value to be written							
			1	Reser	ved function.	Do not write 1	to this bit.				
14-8			0	Reser	ved						
7-0	-0 FFTXDLY7-0 0x0000				DLY7-0 are us These bits def to transmit (E (XA serial cloc um delay of 0 S.	sed only in SF ine the delay b XR2/DXR1) r k or baud cloc serial clock c	PI mode. In Mo between every register. The c k cycles. The ycles and a m	BSP mode th transfer from f lelay is define 8 bit register c aximum of 25	ey are "don't <sup>-</sup> IFO transmit ed in number could define a 6 serial clock		
				In the occur ensur McBS DXR2	FIFO mode th only after the es that the de P/SPI mode /DXR1 are no	e transfer fror shift register f lay count is b with FIFO ar t to be treated	n FIFO to (DX has completed etween two w id delay enab as one additio	R2/DXR1) reg shifting of the ords in the da led, the McB onal level of be	Jisters should last bit. This ta stream. In SP registers uffer.		

15							8			
			Rese	erved						
	R-0									
7			4	3	2	1	0			
	R	eserved		REVTA ENA	RINT ENA	XEVTA ENA	XINT ENA			
		R-0		R/W-0	R/W-0	R/W-0	R/W-0			
Legend:	R = Read only; R	/W = Read/	Write							
Bit(s)	Name	Value	Description							
15-4	Reserved	Reserve	ed							
3	REVTA ENA	Enable A-bis Fl	Enable for A-bis Receive Interrupt on every 16 CLKX/CLKR cycles. Enabled only in A-bis FIFO mode							
		0	0 A-bis Receive interrupt is disabled.							
		1	A-bis Receive interr	rupt is enabled	d.					
2	RINT ENA	Enable	for Receive Interrupt.	Enabled only	in nonFIFO m	node				
		0	0 Receive interrupt on RRDY is disabled.							
		1	Receive interrupt or	n RRDY is ena	abled.					
		This inte	errupt will be active fo	r any of the co	onditions selec	ted for RINTM	1 bits			
1	XEVTA ENA	Enable i in A-bis	Enable for A-bis Transmit Interrupt on every 16 CLKX/CLKR cycles. Enabled only in A-bis FIFO mode							
		0	A-bis transmit interr	rupt is disable	d.					
		1	A-bis transmit interr	rupt is enabled	ł.					
0	XINT ENA	Enable	for transmit Interrupt.	Enabled only	in nonFIFO m	ode				
		0	Transmit interrupt o	on XRDY is dis	abled.					
		1	Transmit interrupt o	n XRDY is en	abled.					
	This interrupt will be active for any of the conditions selected for XINTM bits.									

# Figure 5-6. McBSP FIFO Interrupt Register (MFFINT)

Figure 5-7.	McBSP FIFC	) Status	Register	(MFFST)
0				

15							8		
			Rese	erved					
			R	-0					
7			4	3	2	1	0		
	F	Reserve	b	FSR Flag	EOBR Flag	FSX Flag	EOBX Flag		
		R-0		R/W-x	R/W-0	R/W-x	R/W-0		
Legend:	R = Read only; F	R/W = Re	ad/Write						
Bit(s)	Name	Dese	cription						
15-4	Reserved Reserved								
3	FSR Flag	New RIN bit w will b	New frame-sync FSR pulse detection flag . This flag will be set irrespective of whether RINTM bits are 1/0. Can be used to detect FSR pulse without interrupt. At reset, this bit will be updated based on the MFSXA/MFSRA pin status. If left unconnected, they will be set to 1 due to the internal pullups.						
		0	FSR Frame sync puls	e not detected	k				
		1	FSR Frame sync puls	e detected					
		Write	e 0 to clear						
2	EOBR Flag	Flag End of receive block in multichannel mode . This flag will be set in RINTM bits are 0/1. Can be used to detect EOB without interr					ve of whether		
		0	EOBX End of block co	ondition(EOB)	not occurred.	Write 0 to clea	ar.		
		1	EOBX End of block co	ondition occur	red				
1	FSX Flag	New XINT	Frame sync FSX pulse o M bits are 1/0. Can be	detection flag. used to detect	This flag will be FSX without i	e set irrespecti nterrupt.	ve of whether		
		0	FSX Frame sync puls	e not detected	ł				
		1	FSX Frame sync puls	e detected					
		Write	e 0 to clear						
0	EOBX Flag	End whet	of transmit block in minimer XINTM bits are 0/1.	ultichannel mo Can be used	ode. This flag detect EOB co	will be set in andition withou	respective of ut interrupt.		
		0	EOBX End of block co	ondition (EOB	) not occurred				
		1	EOBX End of block co	ondition occur	red				
		Write	e 0 to clear						

# Chapter 6

# **McBSP Registers**

This chapter includes the register and bit descriptions.

# Topic

# Page

6.1	Data Receive and Transmit Registers
6.2	Serial Port Control Registers (SPCR1 and SPCR2)
6.3	Receive Control Registers (RCR1 and RCR2)
6.4	Transmit Control Registers (XCR1 and XCR2)
6.5	Sample Rate Generator Registers (SRGR1 and SRGR2)
6.6	Multichannel Control Registers (MCR1 and MCR2) 6-22
6.7	Pin Control Register (PCR) 6-29
6.8	Receive Channel Enable Registers (RCERA - RCERH) 6-33
6.9	Transmit Channel Enable Registers (XCERA - XCERH) 6-38
6.10	Register Bit Summary 6-43

#### 6.1 Data Receive and Transmit Registers

This section includes the Data Receive Registers (DRR2 and DRR1) and the Data Transmit Registers (DXR2 and DXR1).

#### 6.1.1 Data Receive Registers (DRR2 and DRR1)

The CPU or the FIFO controller reads received data from one or both of the data receive registers. 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.

Figure 6-1. Data Receive Registers (DRR2 and DRR1)

DRR2		
15		0
	High part of receive data (for 20-, 24- or 32-bit data)	

#### 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)		

#### 6.1.2 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 MDRA 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 FIFO.

If the serial word length is larger than 16 bits, receive data on the MDRA 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 FIFO.

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.

#### 6.1.3 Data Transmit Registers (DXR2 and DXR1)

For transmission, the CPU or the FIFO writes data to one or both of the data transmit registers. 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.

Figure 6-2. Data Transmit Registers (DXR2 and DXR1)

DXR2		
15		0
	High part of transmit data (for 20-, 24- or 32-bit data)	

#### DXR1

15		0
	Transmit data (for 8-, 12-, or 16-bit data) or Low part of transmit data (for 20-, 24- or 32-bit data)	

#### 6.1.4 How Data Travels From the DXRs to the Data Transmit (MDXA) 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 MDXA 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 MDXA 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  $\mu$ -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.

# 6.2 Serial Port Control Registers (SPCR1 and SPCR2)

Each McBSP has two serial port control registers of the form shown in Figure 6-4. and describe the bits in SPCR1 and SPCR2, respectively. These memory-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), A-bis mode (ABIS), interrupt modes (RINTM and XINTM), emulation mode (FREE and SOFT)
- Turn on and off the MDXA-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)

15	14	13	12	11	10		8
DLB	RJL	JST	CLI	KSTP		Reserved	
R/W-0	R/V	V-0	R/	W-0		R-0	
7	6	5	4	3	2	1	0
DXENA	ABIS	RIN	ITM	RSYNCERR	RFULL	RRDY	RRST
R/W-0	R/W-0	R/\	V-0	R/W-0	R-0	R-0	R/W-0

Figure 6-3. SPCR1 Register

Notes: R - Read-only access, R/W - Read/write access, -n - n is the value after a DSP reset

Table 6-1. SPCR1 Register Bit Descriptions

Bit(s)	Name	Descripti	Description			
15	DLB	Digital loo McBSP:	Digital loopback mode bit. DLB disables or enables the digital loopback mode of the McBSP:			
		0	Disabled			
			Internal MDRA is supplied by the MDRA pin. Internal MFSRA 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: MDRA connected to MDXA MFSRA connected to MFXRA MCLKRA connected to MCLKXA			

Table 6-1. SPCR1 Register Bit Descriptions(continued)	
---	--

Bit(s)	Name	Descrip	otion		
			Internal MCLKX/ dependi	MDXA is A are sup ng on the	supplied by the MDXA pin. Internal MFXRA and internal oplied by their respective pins or are generated internally, mode bits FSXM and CLKXM.
			This mo McBSP clocking	de allows transmitt to the M	s you to test serial port code with a single DSP. The er directly supplies data, frame synchronization, and cBSP receiver.
14-13	RJUST	Receive determin registers	e sign-exten nes how da s (DRR1, D	sion and ta is justi RR2).	justification mode bits. During reception, RJUST field and bit filled before being passed to the data receive
		RJUST compan 16-bit da Set the	is ignored it ding mode, ata in DRR Receive Sig	f you ena the 8-bit I. For mo gn-Extens	ble a companding mode with the RCOMPAND bits. In a compressed data in RBR1 is expanded to left-justified ore details about the effects of RJUST, see Section 3.1.11, sion and Justification Mode.
		0	0	Right-jı	ustify and zero-fill MSBs
		0	1	Right-jı	ustify the data and sign-extend the data into the MSBs
		1	0	Left-jus	stify and zero-fill LSBs
		1	1	Reserv	ed (do not use)
12-11	CLKSTP	Clock st the PCF	op mode. Ir ? register.	n SPI mo	de, this field is related to settings on CLKXP and CLKRP in
		0 0	0 Clock st	op mode	disabled. Normal clocking for non-SPI mode.
			CLKXP	CLKRP	
		1 (	0 0	0	In SPI mode: Clock starts with rising edge without delay
		1 (	) 1	0	In SPI mode: Clock starts with falling edge without delay
		1 ·	0	1	In SPI mode: Clock starts with rising edge with delay
		1 ·	1	1	In SPI mode: Clock starts with falling edge with delay

Table 6-1. SPCR1 Register Bit Descriptions(continued)

Bit(s)	Name	Description	
10-8	Reserved	Reserved bits (not available for your use). They are read-only bits and return 0s when read.	
7	DXENA	DX enabler. Enable extra delay for turn-on time. This bit controls the HI-z enable the DX pin, not the data itself, so only the first bit will be delayed in the normal n In A-bis mode, any bit can be delayed since any bit can go from Hi-Z to valid	
		0 DX enabler is off.	
		1 DX enabler is on.	
6	ABIS	ABIS mode	
		0 A-bis mode is disabled.	
		1 A-bis mode is enabled.	
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 services the interrupt request; otherwise, the CPU ignores the request.	
		0 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]):	
		Regardless of the value of RINTM, you can check RRDY to determine whether a word transfer is complete.	
		The McBSP sends an RINT request to the CPU when 16 enabled bits have been received on the DR pin.	
		1h In the multichannel selection mode, the McBSP sends an 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.	
		2h The McBSP sends an RINT request to the CPU when each receive frame-synchronization pulse is detected. The interrupt request is sent even if the receiver is in its reset state.	
		3h The McBSP sends an RINT request to the CPU when the RSYNCERR bit is set, indicating a receive frame-synchronization error.	
		Regardless of the value of RINTM, you can check RSYNCERR to determine whether a receive frame-synchronization error occurred.	
3	RSYNCERR	Receive synchronization error	
		0 No synchronization error	

Bit(s)	Name	Descript	ion
		1	Synchronization error detected by McBSP.
2	RFULL	Receiver	shiftt register (RSR) full
		0	RBR is not in overrun condition.
		1	DRR is not read, RBR is full and RSR is also full with new word.
1	RRDY	Receiver	ready
		0	Receiver is not ready.
		1	Received data is ready to be read from DRR.
0	RRST	Receiver	reset. This resets and enables the receiver.
		0	The serial port receiver is disabled and in reset state.
		1	The serial port receiver is enabled.

Table 6-1. SPCR1 Register Bit Descriptions(continued)

# Figure 6-4. Serial Port Control 2 Register (SPCR2)

15					10	9	8
Reserved							SOFT
	R-0						R/W-0
7	6	5	4	3	2	1	0
FRST	GRST	XIN	ТМ	XSYNCERR	XEMPTY	XRDY	XRST
R/W-0	R/W-0	R/V	V-0	R/W-0	R-0	R-0	R/W-0

Note: R = Read access, W = Write access, R/W - Read/write access

Table 6-2. Serial Port Control 2 Register (SPCR2) Bit Descriptions

Bit(s)	Name	Description
15-10	Reserved	Reserved bits (not available for your use). They are read-only bits and return 0s when read.
9	FREE	Free running mode (FREE and SOFT are taking effect when EMUSUSPEND is set)
		0 Free running mode is disabled
		1 Free running mode is enabled
8	SOFT	Soft bit (FREE and SOFT is taking effect when EMUSUSPEND is set)
		0 SOFT mode is disabled

Table 6-2. Serial Port Control 2 Register (SPCR2) Bit Descriptions (continued)

Bit(s)	Name	Des	Description			
		1	SOFT mode is enabled			
7	FRST	Fran	ne-synchronization logic reset			
		0	Frame Synchronization logic is reset. Frame sync signal FSG is not generated by the sample rate generator.			
		1	Frame sync signal FSG is generated after (FPER+1) number of CLKG clocks i.e. all frame counters are loaded with their programmed values.			
6	GRST	Sam	iple rate generator reset			
		0	Sample rate generator is reset.			
		1	Sample rate generator is pulled out of reset. CLKG is driven as per programmed value in Sample Rate Generator Registers (SRGR).			
5-4	XINTM	Tran	smit Interrupt mode			
		0	XINT driven by XRDY (i.e. end of word) and end of frame in A-bis mode.			
		1h	XINT generated by end-of-block or end-of-frame in multi-channel operation.			
		2h	XINT generated by a new frame synchronization			
		3h	XINT generated by XSYNCERR			
3	XSYNCERR	Tran	smit synchronization error			
		0	No synchronization error			
		1	Synchronization error detected by McBSP			
2	XEMPTY	Tran	smit shift register (XSR) empty			
		0	XSR is empty.			
		1	XSR is not empty.			
1	XRDY	Tran	smitter ready			
		0	Transmitter is not ready to accept new data from the CPU.			
		1	Transmitter is ready to accept new data from the CPU.			
0	XRST	Tran	smitter reset. This resets and enables the transmitter.			
		0	The serial port transmitter is disabled and in reset state.			
		1	The serial port transmitter is enabled.			

# 6.3 Receive Control Registers (RCR1 and RCR2)

Each McBSP has two receive control registers of the form shown in Figure 6–5 and Figure 6–6. These memory-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)

15	14					8
Reserved				RFRLEN1		
R/W-0				R/W-0		
7		5	4			0
	RWDLEN1			Rese	rved	
R/W-0				R-	0	

Figure 6-5. RCR1 Register

Notes: R - Read-only access, R/W - Read/write access, -n - n is the value after a DSP reset

Table 6-3. RCR1 Register Field Descriptions

Bit(s)	Name	Description
15	Reserved	Reserved bits (not available for your use). They are read-only bits and return 0s when read.
14-8	RFRLEN1	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 (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, RFRLEN1 determines the number of serial words in phase 1 of the frame, and RFRLEN2 in RCR2 determines the number of words in phase 2 of the frame.
		The 7-bit RFRLEN fields allow up to 128 words per phase. See Table 6-4 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.
		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 6-3.	RCR1	Register	Field	Descriptions	(continued)
------------	------	----------	-------	--------------	-------------

Bit(s)	Name	Descri	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.				
		0	8 bits			
		1h	12 bits			
		2h	16 bits			
		3h	20 bits			
		4h	24 bits			
		5h	32 bits			
		6h-7h	Reserved			
4-0	Reserved	Reserve when rea	d Reserved bits (not available for your use). They are read-only bits and return 0s ad.			

Table 6-4. Frame Length Formula for Receive Control 1 Register (RCR1)

RPHASE	REFLEN1	RFLEN2	Frame Length
0	0 ≤ RFRLEN1 ≤ 127	Not used	(RFRLEN1 + 1) words
1	$0 \leq \text{RFRLEN1} \leq 127$	$0 \le \text{RFRLEN1} \le 127$	(RFRLEN1 + 1) + (RFRLEN2 + 1) words

Figure 6-6.	Receive	Control 2	Register	(RCR2)
				۰ ·

15	14						8
RPHASE	RFRLEN2						
R/W-0				R/W-0			
7		5	4	3	2	1	0
RWDLEN2			RCOMPAND		RFIG	RDATDLY	
R/W-0			R/W-0		R/W-0	RW-0	

Notes: R - Read-only access, R/W - Read/write access, -n - n is the value after a DSP reset

Table 6-5. Receive Control 2 Register (RCR2) Bit Descriptions

Bit(s)	Name	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	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 (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, RFRLEN1 determines the number of serial words in phase 1 of the frame, and RFRLEN2 in RCR2 determines the number of words in phase 2 of the frame. The 7-bit RFRLEN fields allow up to 128 words per phase. See Table 6-6 for a summary of how to determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period.		
		Program the RFRLEN fields with [w minus 1], where w represents the number of words per phase. For example, if you want a phase length of 128 words in phase 2, load 127 into RFRLEN2.		
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.		
		0 8 bits		
		1h 12 bits		
		2h 16 bits		
		3h 20 bits		
		4h 24 bits		
		5h 32 bits		
		6h-7h Reserved (do not use)		

Table 6-5. Receive Control 2 Register (RCR2) Bit Descriptions (continued)

Bit(s)	Name	Descri	ption		
4-3	RCOMPAND	Receive companding mode bits. Companding (COMpress and exPAND) hardware allows compression and expansion of data in either $\mu$ -law or A-law format. RCOMPAND allows you to choose one of the following companding modes for the McBSP receiver:			
		0	No companding, any size data, MSB received first		
		1h	No companding, 8-bit data, LSB received first		
		2h	$\mu$ -law companding, 8-bit data, MSB received first		
		3h	A-law companding, 8-bit data, MSB received first		
2	RFIG	Receive the tran treated	ive frame-synchronization ignore bit. If a frame-synchronization pulse starts ansfer of a new frame before the current frame is fully received, this pulse is an unexpected frame-synchronization pulse.		
		Setting signals	RFIG causes the serial port to ignore unexpected frame-synchronization during reception.		
		0	Frame-synchronization detect. An unexpected MFSRA pulse causes the receiver to discard the contents of RSR[1,2] in favor of the new incoming data. The receiver:		
			1) Aborts the current data transfer		
			2) Sets RSYNCERR in SPCR1		
			3) Begins the transfer of a new data word		
		1	Frame-synchronization ignore. An unexpected MFSRA pulse is ignored. Reception continues uninterrupted.		
1-0	RDATDLY	Receive cycles a frame.	ive data delay bits. RDATDLY specifies a data delay of 0, 1, or 2 receive close after frame-synchronization and before the reception of the first bit of the e.		
		0	0-bit data delay		
		1h	1-bit data delay		
		2h	2-bit data delay		
		3h	Reserved (do not use)		
RPHASE	RFRLEN1	RFRLEN2	Frame Length		
--------	----------------------------------	----------------------------------	-------------------------------------		
0	$0 \leq \text{RFRLEN1} \leq 127$	Not used	(RFRLEN1 + 1) words		
	$0 \leq \text{RFRLEN1} \leq 127$	$0 \leq \text{RFRLEN2} \leq 127$	(RFRLEN1 + 1) + (RFRLEN2 + 1) words		

Table 6-6. Frame Length Formula for Receive Control 2 Register (RCR2)

### 6.4 Transmit Control Registers (XCR1 and XCR2)

Each McBSP has two transmit control registers of the form shown in Figure 6-7. These 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 6-7. Transmit Contr	ol 2 Register (XCR2)
----------------------------	----------------------

15	14	14 8					
XPHASE	XFRLEN2						
R/W-0		R/W-0					
7		5	4	3	2	1	0
XWDLEN2		XCOMPAND		XFIG	XDA	TDLY	
R/W-0		R/W-0		R/W-0	R/W-0		

Notes: R - Read-only access, R/W - Read/write access, -X - X is the value after a DSP reset

Bit(s)	Name	Descrip	scription nsmit phase number bit. XPHASE determines whether the transmit frame has				
15	XPHASE	Transmi one pha and the XWDLE there ar	it phase number bit. XPHASE determines whether the transmit frame has ase or two phases. For each phase you can define the serial word length number of serial words in the phase. To set up phase 1, program EN1 (word length) and XFRLEN1 (number of words). To set up phase 2 (if e 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.				

Table 6-7. Transmit Control 2 Register (XCR2) Field Descriptions (continued)

Bit(s)	Name	Descri	otion		
		0-7h	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 (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, XFRLEN1 determines the number of serial words in phase 1 of the frame and XFRLEN2 in XCR2 determines the number of words in phase 2 of the frame. The 7-bit XFRLEN fields allow up to 128 words per phase. See Table 6-8 for a summary of how to determine the frame length. This length corresponds to the number of words or logical time slots or channels per frame-synchronization period.		
			Program the XFRLEN fields with [w minus 1], where w represents the number of words per phase. For example, if you want a phase length of 128 words in phase 1, load 127 into XFRLEN1.		
7-5	XWDLEN2	0-7h	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.		
		0	8 bits		
		1h	12 bits		
		2h	16 bits		
		3h	20 bits		
		4h	24 bits		
5h 32 bits		5h	32 bits		
		6h-7h	Reserved (do not use)		
4-2	XCOMPAND	Transm allows o	it companding mode bits. Companding (COMpress and exPAND) hardware compressionand expansion of data in either $\mu$ -law or A-law format.		
		XCOMI McBSP	PAND allows you to choose one of the following companding modes for the transmitter.		
		00b	No companding, any size data, MSB transmitted first		
		01b	No companding, 8-bit data, LSB transmitted first		
		10b	$\mu$ -law companding, 8-bit data, MSB transmitted first		

## Table 6-7. Transmit Control 2 Register (XCR2) Field Descriptions (continued)

Bit(s)	Name	Descri	otion		
		11b	A-law companding, 8-bit data, MSB transmitted first		
1	XFIG	Transm the tran is treate	it frame-synchronization ignore bit. If a frame-synchronization pulse starts isfer of a new frame before the current frame is fully transmitted, this pulse ed as an unexpected frame-synchronization pulse.		
		Setting 2 transmis	XFIG causes the serial port to ignore unexpected frame-synchronization pulses during ssion.		
		0	<ul> <li>Frame-synchronization detect. An unexpected FSX pulse causes the transmitter to discard the content of XSR[1,2]. The transmitter:</li> <li>1) Aborts the present transmission</li> <li>2) Sets XSYNCERR in SPCR2</li> <li>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.</li> </ul>		
		1	Frame-synchronization ignore. An unexpected FSX pulse is ignored. Transmission continues uninterrupted.		
0	XDATDLY	Transm clock cy of the f	mit data delay bits. XDATDLY specifies a data delay of 0, 1, or 2 transmit cycles after frame synchronization and before the transmission of the first frame.		
		0	0-bit data delay		
		1h	1-bit data delay		
		2h	2-bit data delay		
		Зh	Reserved (do not use)		

Table 6-8. Frame Length Formula for Transmit Control 2 Register (XCR2)

XPHASE	XFRLEN1	XFRLEN2	Frame Length
0	$0 \leq XFRLEN1 \leq 127$	Not used	(XFRLEN1 + 1) words
1	$0 \leq XFRLEN1 \leq 127$	$0 \leq XFRLEN2 \leq 127$	(XFRLEN1 + 1) + (XFRLEN2 + 1) words

15 14 8 Reserved XFRLEN1 R/W-0 R/W-0 7 5 4 0 XWDLEN1 Reserved R/W-0 R-0 Note: R = Read access, W = Write access, C = Clear only, -n = value after reset; Table 6-9. XCR1 Register Field Descriptions Bit(s) Name Description 15 Reserved Reserved bit. Read-only; returns 0 when read 0 Valid to write a zero not a 1 1 Reserved value 14-8 XFRLEN1 0-7h 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 (8, 12, 16, 20, 24, or 32 bits per word) in the frame. If a dual-phase frame is selected, XFRLEN1 determines the number of serial words in phase 1 of the frame and XFRLEN2 in XCR2 determines the number of words in phase 2 of the frame. The 7-bit XFRLEN fields allow up to 128 words per phase. See Table 6-8 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. 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. 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. 0 8 bits 1h 12 bits 2h 16 bits 5h 32 bits 6h-7h Reserved (do not use)

#### Figure 6-8. XCR1 Register

|--|

Bit(s)	Name	Descrip	tion
4-0	Reserved	Reserve	d (read-only bits that return 0 when read)
		Зh	20 bits
		4h	24 bits

#### 6.5 Sample Rate Generator Registers (SRGR1 and SRGR2)

Each McBSP has two sample rate generator registers of the form shown in Figure 6–9. The sample rate generator can generate a clock signal (CLKG) and a frame-sync signal (FSG). The 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 pulses 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 MCLKRA or MCLKXA pin) provides the input clock source for the sample rate generator:

- □ If the MCLKXA/MCLKRA pin is used, 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 MFSRA pin, so that CLKG is kept in phase with the input clock.

## Figure 6-9. Sample Rate Generator 2 Register (SRGR2)

15	14	13	12	11			0		
GSYNC	Res	CLKSM	FSGM	FPER					
R/W-0	R/W-1	R/W-1	R/W-0			R/W-0			
Note: F	ote: R - Read-only access, R/W - Read/write access, -X - X is the value after a DSP reset								
Table (	6-10.	Sam	ole Rat	te Generat	or 2 Regi	ster (SRGR2)Field Descriptions			
Bit(s)	Name	e V	alue	Descriptio	n				
15	GSYN	IC		Clock syncl clock sourc MCLKXA p	hronization ce for the s in.	mode bit for CLKG. GSYNC is used only when the sample rate generator is external on the MCLK	∍input RA or		
				When GSY signal (FSC pulses on t	/NC = 1, th G) generate he MFSRA	e clock signal (CLKG) and the frame-synchroni d by the sample rate generator are made depend pin.	zation ent on		
			0	No clock synchronization					
				CLKG oscil cycles.	CLKG oscillates without adjustment, and FSG pulses every (FPER + 1) CLKG cycles.				
			1	Clock sync CLKG clock c	hronization is adjustec on the MCL ulses. FSG	l as necessary so that it is synchronized with the KRA or MCLKXA pin. only pulses in response to a pulse on the MFSF	input ≀A pin.		
				The frame-	synchroniz	ation period defined in FPER is ignored.			
14	Reser	ved R	eserved						
13	CLKS	М		Sample rate generator input clock mode bit. The sample rate generator can accept an input clock signal and divide it down according to CLKGDV to produce an output clock signal, CLKG. The frequency of CLKG is:					
				CLKG frequency = (input clock frequency)/ (CLKGDV + 1)					
				CLKSM is used in conjunction with the SCLKME bit to determine the source for the input clock.					
				A reset selects the CPU clock as the input clock and forces the CLKG frequency to $1/2$ the LSPCLK frequency.					
			0	The input MCLKRA	clock for t pin, depei	he sample rate generator is taken from the nding on the value of the SCLKME bit of PC	R:		
				SCLKME	CLKSM	Input Clock For Sample Rate Generator			
				0	0	Reserved			
				1	0	Signal on MCLKRA pin			

 Table 6-10. Sample Rate Generator 2 Register (SRGR2)Field Descriptions (continued)

 Bit(s) Name
 Value
 Description

 1
 The input clock for the sample rate generator is taken from the LSPCLK or from the MCLKXA pin, depending on the value of the

SCLKME CLKSM Input Clock for Sample Rate Generator

0 1 LSPCLK

SCLKME bit of PCR:

1

1 Signal on MCLKXA pin

- 12 FSGM Sample rate generator transmit frame-synchronization mode bit. The transmitter can get frame synchronization from the MFSXA pin (FSXM = 0) or from inside the McBSP (FSXM = 1). When FSXM = 1, the FSGM bit determines how the McBSP supplies frame-synchronization pulses.
  - 0 If FSXM = 1, the McBSP generates a transmit frame-synchronization pulse when the content of DXR[1, 2] is copied to XSR[1, 2].
  - 1 If FSXM = 1, the transmitter uses frame-synchronization pulses generated by the sample rate generator. 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-FFh Frame-synchronization period bits for FSG. The sample rate generator can produce a clock signal, CLKG, and a frame-synchronization signal, FSG. The period between frame-synchronization pulses on FSG is (FPER + 1) CLKG cycles. The 12 bits of FPER allow a frame-synchronization period of 1 to 4096 CLKG cycles:

≤ FPER ≤ 4095

 $\leq$  (FPER + 1)  $\leq$  4096 CLKG cycles

15 8	7 0
FWID	CLKGDV
R/W-0	R/W-0

Figure 6-10. Sample Rate Generator 1 Register (SRGR1)

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

## Table 6-11. Sample Rate Generator 1 Register (SRGR1)

Bit(s)	Name	Value	Descriptio	on			
15-8	FWID	0-FFh	Frame-syn	Frame-synchronization pulse width bits for FSG			
			The sample rate generator can produce a clock signal, CLKG, and a frame-synchronization signal, FSG. For frame-synchronization pulses on FSG, (FWID + 1) is the pulse width in CLKG cycles. The eight bits of FWID allow a pulse width of 1 to 256 CLKG cycles:				
		0	≤ FWID ≤	≤255			
		1	≤ (FWID -	+ 1) ≤ 25	6 CLKG cycles		
			The period the FPER	l between t bits.	he frame-synchronization pulses on FSG is defined by		
7-0	CLKGDV	0-FFh	Divide-down value for CLKG. The sample rate generator can accept an inpu clock signal and divide it down according to CLKGDV to produce an output clock signal, CLKG. The frequency of CLKG is:				
			CLKG freq	uency = (Ir	nput clock frequency)/ (CLKGDV + 1)		
			The input of	clock is sele	ected by the SCLKME and CLKSM bits:		
			SCLKME	CLKSM	Input Clock for Sample Rate Generator		
			0	0	Reserved		
			0	1	LSPCLK		
			1	0	Signal on MCLKRA pin		
			1	1	Signal on MCLKXA pin		

## 6.6 Multichannel Control Registers (MCR1 and MCR2)

Each McBSP has two multichannel control registers. MCR1 (Figure 6–11) has control and status bits (with an R prefix) for multichannel selection operation in the receiver. MCR2 (Figure 6–12) contains the same type of bits (bit with an X prefix) for the transmitter. These 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 6-11. Multichannel Control 1 Register (MCR1)

15					10	9	8
	RMCME	RPBBLK					
	R-0					R/W-0	R/W-0
7	6	5	4		2	1	0
RPBBLK	RPABLK		RCBLK		Reserved	RMCM	
R/W-0	R/\	N-0	R-0			R-0	R/W-0

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

Table 6-12. Multichannel Control 1 Register (MCR1)Field Descriptions

Bit(s)	Name	Value	Description				
15-10	Reserved	Reserver when r	erved bits (not available for your use). They are read-only bits and return 0s n read.				
9	RMCME	Receiv be indi	ceive multichannel partition mode bit. RMCME is only applicable if channels can individually enabled or disabled for reception ( $RMCM = 1$ ).				
		RMCN individ	MCME determines whether only 32 channels or all 128 channels are to I advidually selectable.				
		🗋 Ri va	MCME = 0 and XMCME = 0: (normal multichannel selection mode) (Default lue) Maximum 32 channels can be enabled at one time.				
		□ RI ch	MCME = 1 and XMCME = 1: (128 channel selection mode) Maximum 128 annels can be enabled at one time. All other modes reserved.				

## Table 6-12. Multichannel Control 1 Register (MCR1)Field Descriptions (continued)

Bit(s)	Name	Value	Description
		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
8-7	RPBBLK	0-3h	Receive/transmit 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.

## Table 6-12. Multichannel Control 1 Register (MCR1)Field Descriptions (continued)

Bit(s)	Name	Value	Description
			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).
		0	Block 1: channels 16 through 31
		1h	Block 3: channels 48 through 63
		2h	Block 5: channels 80 through 95
		3h	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.
		0	Block 0: channels 0 through 15
		1h	Block 2: channels 32 through 47
		2h	Block 5: channels 64 through 79
		3h	Block 7: channels 96 through 111
4-2	RCBLK		Receive current block indicator. RCBLK indicates which block of 16 channels is involved in the current McBSP reception:
		0	Block 0: channels 0 through 15
		1h	Block 1: channels 16 through 31
		2h	Block 2: channels 32 through 47
		3h	Block 3: channels 48 through 63
		4h	Block 4: channels 64 through 79
		5h	Block 5: channels 80 through 95
		6h	Block 6: channels 96 through 111
		7h	Block 7: channels 112 through 127
1	Reserved		Reserved bits (not available for your use). They are read-only bits and return 0s when read.

Table 6-12. Multichannel Control 1 Register (MCR1)Field Descriptions (continued)

Bit(s)	Name	Value	Description
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	Multichanneled 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.

### Figure 6-12. Multichannel Control 2 Register (MCR2)

15					10	9	8
		XMCME	XPBBLK				
		R	R/W-0	R/W-0			
7	6	5	4		2	1	0
XPBBLK	XPABLK		XCBLK		XM	ICM	
R/W-0	R/\	V-0	R-0			R/	N-0

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset

Table 6-13. Multichannel Control 2 Register (MCR2) Field Descriptions

Bit(s)	Name	Value	Description
15-10	Reserved		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 B

## Table 6-13. Multichannel Control 2 Register (MCR2) Field Descriptions (continued)

Bit(s)	Name	Value	Description
		1	8-partition mode. All partitions (A through H) are used. You can control up to 128 channels in the transmit multichannel selection mode selected with the XMCM bits. You control the channels with the appropriate transmit channel enable registers:
			XCERA: Channels 0 through 15 XCERB: Channels 16 through 31 XCERC: Channels 32 through 47 XCERD: Channels 48 through 63 XCERE: Channels 64 through 79 XCERF: Channels 80 through 95 XCERG: Channels 96 through 111 XCERH: Channels 112 through 127
8-7	XPBBLK	0-3h	Transmit partition B block bits XPBBLK is only applicable if channels can be individually disabled/enabled and masked/unmasked (XMCM is nonzero) and the 2-partition mode is selected (XMCME = 0). Under these conditions, the McBSP transmitter can transmit or withhold data in any of the 32 channels that are assigned to partitions A and B of the transmitter. The 128 transmit channels of the McBSP are divided equally among 8 blocks (0 through 7). When XPBBLK is applicable, use XPBBLK to assign one of the odd-numbered blocks (1, 3, 5, or 7) to partition B, as shown in the following table. Use the XPABLK bit to assign one of the even-numbered blocks (0, 2, 4, or 6) to partition A. If you want to use more than 32 channels, you can change block assignments dynamically. You can assign a new block to one partition while the transmitter is handling activity in the other partition. For example, while the block in partition A is active, you can change which block is assigned to partition B. The XCBLK bits are regularly updated to indicate which block is active. When XMCM = 11b (for symmetric transmission and reception), the transmitter uses the receive block bits (RPABLK and RPBBLK) rather than the transmit block bits (XPABLK and XPBBLK).
		0	Block 1: channels 16 through 31
		1h	Block 3: channels 48 through 63
		2h	Block 5: channels 80 through 95
		3h	Block 7: channels 112 through 127

## Table 6-13. Multichannel Control 2 Register (MCR2) Field Descriptions (continued)

Bit(s)	Name	Value	Description
6-5	XPABLK	0-3h	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.
		0	Block 0: channels 0 through 15
		1h	Block 2: channels 32 through 47
		2h	Block 4: channels 64 through 79
		Зh	Block 6: channels 96 through 111
4-2	XCBLK		Transmit current block indicator. XCBLK indicates which block of 16 channels is involved in the current McBSP transmission:
		0	Block 0: channels 0 through 15
		<b>1</b> h	Block 1: channels 16 through 31
		2h	Block 2: channels 32 through 47
		3h	Block 3: channels 48 through 63
		4h	Block 4: channels 64 through 79
		5h	Block 5: channels 80 through 95
		6h	Block 6: channels 96 through 111
		7h	Block 7: channels 112 through 127
1-0	XMCM	0-3h	Transmit multichannel selection mode bits. XMCM determines whether all channels or only selected channels are enabled and unmasked for transmission.
		0	No transmit multichannel selection mode is on. All channels are enabled and unmasked. No channels can be disabled or masked.
		1h	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.

#### Table 6-13. Multichannel Control 2 Register (MCR2) Field Descriptions (continued)

#### Bit(s) Name Value Description

2h 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.

3h 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.

#### 6.7 Pin Control Register (PCR)

Each McBSP has one pin control register (PCR). Figure 6–13 describes the bits of PCR. This register enables you to:

- 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)
- 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)

The pin control register (PCR) is shown in Figure 6-13 and described in Table 6-14.

#### Figure 6-13. Pin Control Register (PCR)

15			12	11	10	9	8
	Rese	erved		FSXM	FSRM	CLKXM	CLKRM
	R	-0		R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
SCLKME	Reserved	DX_STAT	DR_STAT	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

**Note:** R = Read access, W = Write access, C = Clear only, -n = value after reset;

Table 6-14. Pin Control Register (PCR) Field Descriptions

Bit(s)	Name	Value	Description
15-12	Reserved		Reserved bit (not available for your use). It is a read-only bit and returns a 0 when read.
11	FSXM		Transmit frame-synchronization mode bit. FSXM determines whether transmit frame-synchronization pulses are supplied externally or internally. The polarity of the signal on the MFSXA pin is determined by the FSXP bit.
		0	Transmit frame synchronization is supplied by an external source via the MFSXA pin.

# Table 6-14. Pin Control Register (PCR) Field Descriptions (continued)

Bit(s)	Name	Value	Description
		1	Transmit frame synchronization is generated internally by the Sample Rate generator, as determined by the FSGM bit of SRGR2.
10	FSRM		Receive frame-synchronization mode bit. FSRM determines whether receive frame-synchronization pulses are supplied externally or internally. The polarity of the signal on the MFSRA pin is determined by the FSRP bit.
		0	Receive frame synchronization is supplied by an external source via the MFSRA pin.
		1	Receive frame synchronization is supplied by the sample rate generator. MFSRA is an output pin reflecting internal FSR, except when $GSYNC = 1$ in SRGR2.
9	CLKXM		Transmit clock mode bit. CLKXM determines whether the source for the transmit clock is external or internal, and whether the MCLKXA pin is an input or an output. The polarity of the signal on the MCLKXA 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 MCLKXA pin.
		1	Internal CLKX is driven by the sample rate generator of the McBSP. The MCLKXA 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 MCLKXA pin. The internal receive clock (MCLKRA) 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 MCLKXA pin to drive the shift clock of the SPI-compliant slaves in the system. Internal CLKX also drives the internal receive clock (MCLKRA), so that both the transmitter and the receiver are controlled by the internal master clock.
8	CLKRM		Receive clock mode bit. The role of CLKRM and the resulting effect on the MCLKRA pin depend on whether the McBSP is in the digital loopback mode $(DLB = 1)$ .
			The polarity of the signal on the MCLKRA pin is determined by the CLKRP bit.
			Not in digital loopback mode (DLB = 0):
		0	The MCLKRA pin is an input pin that supplies the internal receive clock (MCLKRA).

Table 6-14.	Pin Control Register	(PCR) Fie	ld Descriptions	(continued)
-------------	----------------------	-----------	-----------------	-------------

Bit(s)	Name	Value			Description
		1	Internal MC MCLKRA pi	LKRA is d n is an out	riven by the sample rate generator of the McBSP. The put pin that reflects internal CLKR.
			In digital loo	pback mod	de (DLB = 1):
		0	The MCLKF (MCLKRA) according to	RA pin is i is driven b the CLKX	n the high impedance state. The internal receive clock by the internal transmit clock (CLKX). CLKX is derived M bit.
		1	Internal CLk reflects inter	KR is driver rnal MCLK	by internal CLKX. The MCLKRA pin is an output pin that RA. CLKX is derived according to the CLKXM bit.
7	SCLKME		Sample rate produce a c	e generato lock signal	r input clock mode bit. The sample rate generator can , CLKG. The frequency of CLKG is:
			CLKG freq.	= (Input clo	ock frequency) / (CLKGDV + 1)
			SCLKME is	used in co	njunction with the CLKSM bit to select the input clock.
			SCLKME	CLKSM	Input Clock For Sample Rate Generator
			0	0	Reserved
			0	1	LSPCLK
			The input clo from the MC	ock for the CLKXA pin,	sample rate generator is taken from the MCLKRA pin or depending on the value of the CLKSM bit of SRGR2:
			SCLKME	CLKSM	Input Clock For Sample Rate Generator
			1	0	Signal on MCLKRA pin
			1	1	Signal on MCLKXA pin
6	Reserved		Reserved		
5	DX_STAT		DX pin statu DX by writin	us bit. Whe	n DXSTAT is applicable, you can toggle the signal on AT.
			DXSTAT is o is configured (XIOEN = 1)	only applica d for use as ).	able when the transmitter is in reset (XRST = 0) and DX s a general-purpose output pin
		0	Drive the sig	gnal on the	DX pin low.
		1	Drive the sig	gnal on the	DX pin high.
4	DR_STAT		MDRA pin s MDRA pin.	tatus bit. V	Vhen DRSTAT is applicable, it reflects the level on the
			DRSTAT is MDRA is co (RIOEN = 1	only applic nfigured fo ).	able when the receiver is in reset (RRST = 0) and r use as a general-purpose input pin

Table 6-14. Pin Control Register (PCR) Field Descriptions (continued)

Bit(s)	Name	Value	Description
		0	The signal on MDRA pin is low.
		1	The signal on MDRA pin is high.
3	FSXP		Transmit frame-synchronization polarity bit. FSXP determines the polarity of FSX as seen on the MFSXA pin.
		0	Transmit frame-synchronization pulses are active high.
		1	Transmit frame-synchronization pulses are active low.
2	FSRP		Receive frame-synchronization polarity bit. FSRP determines the polarity of FSR as seen on the MFSRA pin.
		0	Receive frame-synchronization pulses are active high.
		1	Receive frame-synchronization pulses are active low.
1	CLKXP		Transmit clock polarity bit. CLKXP determines the polarity of CLKX as seen on the MCLKXA pin.
		0	Transmit data is sampled on the rising edge of CLKX.
		1	Transmit data is sampled on the falling edge of CLKX.
0	CLKRP		Receive clock polarity bit. CLKRP determines the polarity of CLKR as seen on the MCLKRA pin.
		0	Receive data is sampled on the falling edge of MCLKRA.
		1	Receive data is sampled on the rising edge of MCLKRA.

Table 6-15. Pin Configuration

Pin	Selected as Output When	Selected as Input When
MCLKXA	CLKXM = 1	CLKXM = 0
MFSXA	FSXM = 1	FSXM = 0
MCLKRA	CLKRM = 1	CLKRM = 0
MFSRA	FSRM = 1	FSRM = 0

### 6.8 Receive Channel Enable Registers (RCERA - RCERH)

Each McBSP has eight receive channel enable registers of the form shown in Figure 6-14. There is one for each of the receive partitions: A, B, C, D, E, F, G, and H.

These memory-mapped registers are only used when the receiver is configured to allow individual enabling and disabling of the channels (RMCM = 1) or when the receiver needs bit-enable patterns for the A-bis mode (ABIS = 1). For more details about the way these registers are used be sure to read the topics at the end of this section:

RCERs Used in the Receive Multichannel Selection Mode (page 6-35)
 RCERs Used in the A-bis Mode (page 6-36)

Figure 6-14	Receive	Channel	Fnable	Reaister	(RCFRA/B	)
0					· · · ·	

15	14	13	12	11	10	9	8
RCEA15	RCEA14	RCEA13	RCEA12	RCEA11	RCEA10	RCEA9	RCEA8
 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
RCEA7	RCEA6	RCEA5	RCEA4	RCEA3	RCEA2	RCEA1	RCEA0
 R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

Bit(s)	Name	Description			
	RCERA	Rece	ive Channel Enable Register A		
15-0	RCEA <i>n</i> 0 ≤ n ≤ 15	Receive channel enable			
		0	Disables reception of <i>n</i> th channel in an even-numbered block in partition A		
		1	Enables reception of <i>n</i> th channel in an even-numbered block in partition A		
	RCERB	Rece	ive Channel Enable Register B		
15-0	RCER <i>n</i> 0 ≤ n ≤ 15	Receive/transmit channel enable			
		0	Disables reception of <i>n</i> th channel in odd-numbered block in partition B.		
		1	Enables reception of <i>n</i> th channel in odd-numbered block in partition B.		

Figure 6-15. RCER(A-G)-Receive Channel Enable Registers - A, C, E, G (for 8-partition mode)

15	14	13	12	11	10	9	8
RCERx15	RCERx14	RCERx13	RCERx12	RCERx11	RCERx10	RCERx9	RCERx8
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
RCERX/	RCERx6	RCERx5	RCERx4	RCERx3	RCERx2	RCERx1	RCERx0

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

Name	Desc	ription
RCERx	Receive channel enable	
	0	Disables reception of <i>n</i> th channel in partition x.
	1	Enables reception of <i>n</i> th channel in partition x.
	Name RCERx	NameDescRCERxRece01

Note: Receive channel enable register bit layout for 128 channels x= Partition A, C, E, G; n = bit 15-0 y = Partition B, D, F, H; n = bit 15-0

# Figure 6-16. RCER(B-H)-Receive Channel Enable Registers - B,D,F,H (for 8-partition mode)

15	14	13	12	11	10	9	8
RCERy15	RCERy14	RCERy13	RCERy12	RCERy11	RCERy10	RCERy9	RCERy8
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
		-	•	0	_	•	-
RCERy7	RCERy6	RCERy5	RCERy4	RCERy3	RCERy2	RCERy1	RCERy0

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

Bit(s)	Name	Description
15-0	RCERy	Receive channel enable

- 0 Disables reception of *n*th channel in partition y.
- 1 Enables reception of *n*th channel in partition y.

Note: Receive channel enable register bit layout for 128 channels x= Partition A, C, E, G; n = bit 15-0 y = Partition B, D, F, H; n = bit 15-0

#### 6.8.1 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 (see Table 6–16).

Selectable Channels	Block Assignments	Channel Assignments
32 (RMCME = 0)	RCERA: Channels n–(n + 15) (channels assigned with the RPABLK bits)	RCEA0: Channel n RCEA1: Channel (n + 1) RCEA2: Channel (n + 2) :
		RCEA15: Channel (n + 15)
	RCERB: Channels m–(m + 15) (channels assigned with the RPBBLK bits)	RCEB0: Channel m RCEB1: Channel (m + 1) RCEB2: Channel (m + 2) : RCEB15: Channel (m + 15)
	Other RCERs not used	-
128 (RMCME = 1)	RCERA: Block 0	RCEA0: Channel 0 RCEA1: Channel 1 RCEA2: Channel 2
		RCEA15: Channel 15
	RCERB: Block 1	RCEB0: Channel 16 RCEB1: Channel 17 RCEB2: Channel 18 : RCEB15: Channel 31
	RCERC: Block 2	RCEC0: Channel 32 RCEC1: Channel 33 RCEC2: Channel 34 : RCEC15: Channel 47
	RCERD: Block 3	RCED0: Channel 48 RCED1: Channel 49 RCED2: Channel 50 : RCED15: Channel 63

Table 6-16. Use of the Receive Channel Enable Registers in theReceive MultichannelSelection Mode

Selectable Channels	Block Assignments	Channel Assignments
128 (continued)	RCERE: Block 4	RCEE0: Channel 64 RCEE1: Channel 65 RCEE2: Channel 66 : RCEE15: Channel 79
	RCERF: Block 5	RCEF0: Channel 80 RCEF1: Channel 81 RCEF2: Channel 82 : RCEF15: Channel 95
	RCERG: Block 6	RCEG0: Channel 96 RCEG1: Channel 97 RCEG2: Channel 98 : RCEG15: Channel 111
	RCERH: Block 7	RCEH0: Channel 112 RCEH1: Channel 113 RCEH2: Channel 114 : RCEH15: Channel 127

 Table 6-16.
 Use of the Receive Channel Enable Registers in theReceive Multichannel

 Selection Mode (Continued)

#### 6.8.2 RCERs Used in the A-bis Mode

In A-bis mode operation, only RCERA and RCERB are used. Sixteen bits at a time are passed to the receiver. As each of the 16 bits arrives on the DR pin (the MSB, bit 15, arrives first), the bit is stored or ignored, depending on the bit-enable pattern in RCERA or RCERB. The first 16 bits that arrive are handled according to the bit-enable pattern in RCERA. For example, if RCEA6 = 1 and all the other bits of RCERA are 0s, only bit 6 is stored. Each of the next 16 bits is stored or ignored according to the bit-enable pattern in RCERB. The receiver alternately uses RCERA and RCERB for consecutive 16-bit words.

Table 6–17 shows how bits of RCERA correspond to the bits of the incoming word.

Register	Bits
RCERA	RCEA15: Enables or masks the first bit of the incoming word RCEA14: Enables or masks bit 14 of the incoming word RCEA13: Enables or masks bit 13 of the incoming word : RCEA0: Enables or masks the last bit of the incoming word
RCERB	RCEB15: Enables or masks the first bit of the incoming word RCEB14: Enables or masks bit 14 of the incoming word RCEB13: Enables or masks bit 13 of the incoming word : RCEB0: Enables or masks the last bit of the incoming word

Table 6-17. Use of Receive Channel Enable Registers A and B in the A-bis Mode

## 6.9 Transmit Channel Enable Registers (XCERA - XCERH)

Each McBSP has eight transmit channel enable registers of the form shown in the following figure. There is one for each of the transmit partitions: A, B, C, D, E, F, G, and H. provides a general bit description that applies to each of the transmit channel enable registers.

These mapped registers are only used when transmitter is configured to allow individual disabling/enabling and masking/unmasking of the channels (XMCM is nonzero) or when the transmitter needs bit-enable patterns for the A-bis mode (ABIS = 1). For more details about the way these registers are used, read the topics at the end of this section:

XCERs Used in a Transmit Multichannel Selection Mode (page 6-39)
 XCERs Used in the A-bis Mode (page 6-41)

Figure 6-17. Transmit Channel Enable Registers A. C. E. G (XCERA-XCERG)

15	14	13	12	11	10	9	8
XCERx15	XCERx14	XCERx13	XCERx12	XCERx11	XCERx10	XCERx9	XCERx8
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
XCERx7	XCERx6	XCERx5	XCERx4	XCERx3	XCERx2	XCERx1	XCERx0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

Bit(s)	Name	Description			
15-0	XCERx	Transmit channel enable register			
		0 Disables transmission of <i>n</i> th channel in partition x.			

1 Enables transmission of *n*th channel in partition x.

Note: Transmit channel enable register bit layout for 128 channels x= Partition A, C, E, G; n = bit 15-0 y = Partition B, D, F, H; n = bit 15-0

15	14	13	12	11	10	9	8
XCERy15	XCERy14	XCERy13	XCERy12	XCERy11	XCERy10	XCERy9	XCERy8
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
XCERy7	XCERy6	XCERy5	XCERy4	XCERy3	XCERy2	XCERy1	XCERy0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Figure 6-18. Transmit Channel Enable Registers–B, D, F, H (XCERB-XCERH)

Note: R = Read access, W = Write access, C = Clear only, -n = value after reset;

Bit(s	s) Name	Desc	ription			
15-	0 XCERy	Trans	Transmit channel enable register			
		0	Disables transmission of <i>n</i> th channel in partition y.			
		1	Enables transmission of <i>n</i> th channel in partition y.			
Note:	Transmit channel enable x= Partition A, C, E, G; n $\gamma$ = Partition B, D, F, H; n	register = bit 15 = bit 15	bit layout for 128 channels -0 -0			

#### 6.9.1 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, as shown in the following table.

#### 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 6-18.	Use of the Transmit Channel Enable Registers in a Transmit Multichannel
S	election Mode

Selectable Channels	Block Assignments	Channel Assignments
32 (XMCME = 0)	XCERA: Channels $n_{(n + 15)}$ (channels assigned with the XPABLK bits for XMCM = 01b or 10b; assigned with the RPABLK bits for XMCM = 11b)	XCEA0: Channel n XCEA1: Channel (n + 1) XCEA2: Channel (n + 2) : XCEA15: Channel (n + 15)
	XCERB: Channels m–(m + 15) (channels assigned with the XPBBLK bits for XMCM = 01b or 10b; assigned with the RPBBLK bits for XMCM = 11b)	XCEB0: Channel m XCEB1: Channel (m + 1) XCEB2: Channel (m + 2) : XCEB15: Channel (m + 15)
	Other XCERs not used	_
128 (XMCME = 1)	XCERA: Block 0	XCEA0: Channel 0 XCEA1: Channel 1 XCEA2: Channel 2 : XCEA15: Channel 15
	XCERB: Block 1	XCEB0: Channel 16 XCEB1: Channel 17 XCEB2: Channel 18 : XCEB15: Channel 31
	XCERC: Block 2	XCEC0: Channel 32 XCEC1: Channel 33 XCEC2: Channel 34 :
128 (continued)	XCERD: Block 3	XCEC15: Channel 47 XCED0: Channel 48 XCED1: Channel 49 XCED2: Channel 50 : XCED15: Channel 63
	XCERE: Block 4	XCEE0: Channel 64 XCEE1: Channel 65 XCEE2: Channel 66 : XCEE15: Channel 79

Selectable Channels	Block Assignments	Channel Assignments
	XCERF: Block 5	XCEF0: Channel 80
		XCEF1: Channel 81
		XCEF2: Channel 82
		:
		XCEF15: Channel 95
	XCERG: Block 6	XCEG0: Channel 96
		XCEG1: Channel 97
		XCEG2: Channel 98
		:
		XCEG15: Channel 111
	XCERH: Block 7	XCEH0: Channel 112
		XCEH1: Channel 113
		XCEH2: Channel 114
		:
		XCEH15: Channel 127

 Table 6-18.
 Use of the Transmit Channel Enable Registers in a Transmit Multichannel

 Selection Mode (Continued)

#### 6.9.2 XCERs Used in the A-bis Mode

In A-bis mode operation, only XCERA and XCERB are used. Sixteen bits at a time are passed to data transmit register 1 (DXR1) by the CPU or FIFO. Each of the 16 bits is transmitted on the DX pin or is ignored, depending on the bit-enable pattern in XCERA or XCERB. The first 16 bits that enter the transmitter are handled according to the bit-enable pattern in XCERA. For example, if XCEA13 = 1 and all the other bits of XCERA are 0s, only bit 13 is transmitted. Each of the next 16 bits is transmitted or ignored according to the bit-enable pattern in XCERB. The transmitted or ignored according to the bit-enable pattern in XCERB. The transmitted or ignored according to the bit-enable pattern in XCERB. The transmitter alternately uses XCERA and XCERB for consecutive 16-bit words.

Table 6–19 shows how bits of XCERA and XCERB correspond to the bits of a word written to DXR1 (MSB = most significant bit, LSB = least significant bit).

Register	Bits
XCERA	XCEA15: Enables or masks the MSB of the word written to DXR1 XCEA14: Enables or masks bit 14 of the word written to DXR1 XCEA13: Enables or masks bit 13 of the word written to DXR1
	XCEA0: Enables or masks the LSB of the word written to DXR1
XCERB	XCEB15: Enables or masks the MSB of the word written to DXR1 XCEB14: Enables or masks bit 14 of the word written to DXR1 XCEB13: Enables or masks bit 13 of the word written to DXR1
	: XCEB0: Enables or masks the LSB of the word written to DXR1

Table 6-19. Use of Transmit Channel Enable Registers A and B in the A-bis Mode

# 6.10 Register Bit Summary

Register Address	MSB bits	15	14	13	12	11	10	9	8
Offset 0x00	LSB bits	7	6	5	4	3	2	1	0
7800	DRR2	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
		Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7801	DRR1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7802	DXR2	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
		Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
7803	DXR1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
7804	SPCR2			Reserved					SOFT
				R∙	-0			R/W-0	R/W-0
		FRST	GRST	XIN	ТМ	XSYNCERR	XEMPTY	XRDY	XRST
		R/W-0	R/W-0	R/V	/-0	R/W-0	R-0	R-0	R/W-0
7805	SPCR1	DLB	RJU	JST	CLK	STP		Reserved	
		R/W-0	R/V	V-0	R/V	V-0		R-0	
		DXENA	ABIS	RIN	ТМ	RSYNCERR	RFULL	RRDY	RRST
		R/W-0	R/W-0	R/V	/-0	R/W-0	R-0	R-0	R/W-0
7806	RCR2	RPHASE				RFRLEN2			
		R/W-0				R/W-0			
			RWDLEN2		RCON	IPAND	RFIG	RDA	TDLY
			R/W-0         R/W-0         R/W-0         R/W-0					/-0	
7807	RCR1	Reserved				RFRLEN1			
		R/W-0	R/W-0						
			RWDLEN1 Reserved						
		1	R/W-0				R-0		
7808	XCR2	XPHASE				XFRLEN2			
		R/W-0							

Table 6-20. Register Bit Summary (Base Address 0x00 7800)

Register Address	MSB bits	15	14	13	12	11	10	9	8		
Offset 0x00	LSB bits	7	6	5	4	3	2	1	0		
			XWDLEN2 XCOMPAND XFIG XD			XDA	ГDLY				
			R/W-0		R/W-0						
7809	XCR1	Reserved		XFRLEN1							
		R/W-0				R/W-0					
			XWDLEN1				Reserved				
			R/W-0				R-0				
780A	SRGR2	GSYNC	Reserved	CLKSM	FSGM		FP	FPER			
		R/W-0	R/W-0	R/W-1	R/W-0		R/V	V-0			
					FP	ER					
					R/V	V-0					
780B	SRGR1				FV	/ID					
					R/V	V-0					
					CLK	GDV					
			R/W-0								
780C	MCR2			Reserved					XPBBLK		
				R∙	R/W-0	R/W-0					
		XPBBLK	XPA	BLK	XMCM						
		R/W-0	R/W-0 R-0					R/W-0			
780D	MCR1			Rese	erved			RMCME	RPBBLK		
				R	-0			R/W-0	R/W-0		
		RPBBLK	RPA	BLK		RCBLK		Reserved	RMCM		
		R/W-0	R/V	V-0		R-0		R-0	R/W-0		
780E	RCERA	RCEA15	RCEA14	RCEA13	RCEA12	RCEA11	RCEA10	RCEA9	RCEA8		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
		RCEA7	RCEA6	RCEA5	RCEA4	RCEA3	RCEA2	RCEA1	RCEA0		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
780F	RCERB	RCEB15	RCEB14	RCEB13	RCEB12	RCEB11	RCEB10	RCEB9	RCEB8		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
		RCEB7	RCEB6	RCEB5	RCEB4	RCEB3	RCEB2	RCEB1	RCEB0		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
7810	XCERA	XCEA15	XCEA14	XCEA13	XCEA12	XCEA11	XCEA10	XCEA9	XCEA8		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
		XCEA7	XCEA6	XCEA5	XCEA4	XCEA3	XCEA2	XCEA1	XCEA0		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
7811	XCERB	XCEB15	XCEB14	XCEB13	XCEB12	XCEB11	XCEB10	XCEB9	XCEB8		
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		

Table 6-20. Register Bit Summary (Continued) (Base Address 0x00 7800) (Continued)

Register Address	MSB bits	15	14	13	12	11	10	9	8
Offset 0x00	LSB bits	7	6	5	4	3	2	1	0
		XCEB7	XCEB6	XCEB5	XCEB4	XCEB3	XCEB2	XCEB1	XCEB0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7812	PCR		Rese	erved		FSXM	FSRM	CLKXM	CLKRM
			R	-0		R/W-0	R/W-0	R/W-0	R/W-0
		SCLKME	Reserved	DX_STAT	DR_STAT	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
7813	RCERC	RCEC15	RCEC14	RCEC13	RCEC12	RCEC11	RCEC10	RCEC9	RCEC8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		RCEC7	RCEC6	RCEC5	RCEC4	RCEC3	RCEC2	RCEC1	RCEC0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7814	RCERD	RCED15	RCED14	RCED13	RCED12	RCED11	RCED10	RCED9	RCED8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		RCED7	RCED6	RCED5	RCED4	RCED3	RCED2	RCED1	RCED0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7815	XCERC	XCEC15	XCEC14	XCEC13	XCEC12	XCEC11	XCEC10	XCEC9	XCEC8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		XCEC7	XCEC6	XCEC5	XCEC4	XCEC3	XCEC2	XCEC1	XCEC0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7816	XCERD	XCED15	XCED14	XCED13	XCED12	XCED11	XCED10	XCED9	XCED8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		XCED7	XCED6	XCED5	XCED4	XCED3	XCED2	XCED1	XCED0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7817	RCERE	RCEB15	RCEB14	RCEB13	RCEB12	RCEB11	RCEB10	RCEB9	RCEB8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		RCEB7	RCEB6	RCEB5	RCEB4	RCEB3	RCEB2	RCEB1	RCEB0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7818	RCERF	RCEF15	RCEF14	RCEF13	RCEF12	RCEF11	RCEF10	RCEF9	RCEF8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		RCEF7	RCEF6	RCEF5	RCEF4	RCEF3	RCEF2	RCEF1	RCEF0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7819	XCERE	XCEB15	XCEB14	XCEB13	XCEB12	XCEB11	XCEB10	XCEB9	XCEB8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		XCEB7	XCEB6	XCEB5	XCEB4	XCEB3	XCEB2	XCEB1	XCEB0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
781A	XCERF	XCEF15	XCEF14	XCEF13	XCEF12	XCEF11	XCEF10	XCEF9	XCEF8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Table 6-20. Register Bit Summary (Continued) (Base Address 0x00 7800) (Continued)

SPRU061D

Register Address	MSB bits	15	14	13	12	11	10	9	8
Offset 0x00	LSB bits	7	6	5	4	3	2	1	0
		XCEF7	XCEF6	XCEF5	XCEF4	XCEF3	XCEF2	XCEF1	XCEF0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
781B	RCERG	RCEG15	RCEG14	RCEG13	RCEG12	RCEG11	RCEG10	RCEG9	RCEG8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		RCEG7	RCEG6	RCEG5	RCEG4	RCEG3	RCEG2	RCEG1	RCEG0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
781C	RCERH	RCEH15	RCEH14	RCEH13	RCEH12	RCEH11	RCEH10	RCEH9	RCEH8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		RCEH7	RCEH6	RCEH5	RCEH4	RCEH3	RCEH2	RCEH1	RCEH0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
781D	XCERG	XCEG15	XCEG14	XCEG13	XCEG12	XCEG11	XCEG10	XCEG9	XCEG8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		XCEG7	XCEG6	XCEG5	XCEG4	XCEG3	XCEG2	XCEG1	XCEG0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
781E	XCERH	XCEH15	XCEH14	XCEH13	XCEH12	XCEH11	XCEH10	XCEH9	XCEH8
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
		XCEH7	XCEH6	XCEH5	XCEH4	XCEH3	XCEH2	XCEH1	XCEH0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

 Table 6-20.
 Register Bit Summary (Continued) (Base Address 0x00 7800) (Continued)

The registers bits are summarized in Table 6-21.

Table 6-21. FIFO Register Bit Descriptions (Base address 0x00 7800)

Register Address- Offset 0x000	MSB bits	15	14	13	12	11	10	9	8
	LSB bits	7	6	5	4	3	2	1	0
00	DRR2	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
		R-0							
		Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
		R-0							
01	DRR1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		R-0							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		R-0							
02	DXR2	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
		W-0							
		Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

Register Address-	MSB bits	15	14	13	12	11	10	9	8
Offset 0x000	LSB bits	7	6	5	4	3	2	1	0
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
03	DXR1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
20	MFFTX	Reserved	MFFENA	TXFIFO Reset	TXFFST4	TXFFST3	TXFFST2	TXFFST1	TXFFST0
		R-0	R/W-0	R/W-1	R-0	R-0	R-0	R-0	R-0
		TXFFINT Flag	TXFFINT CLR	TXFFIENA	TXFFIL4	TXFFIL3	TXFFIL2	TXFFIL1	TXFFIL0
		R-0	W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
21	MFFRX	RXFFOVF Flag	RXFFOVF CLR	RXFIFO Reset	RXFFST4	RXFFST3	RXFIFST2	RXFFST1	RXFFST0
		R-0	W-0	R/W-1	R-0	R-0	R-0	R-0	R-0
		RXFFINT- Flag	RXFFINT-C LR	RXFFIENA	RXFFIL4	RXFFIL3	RXFFIL2	RXFFIL1	RXFFIL0
		R-0	W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
22	MFFCT				Rese	erved			
					R	-0			
		FFTXDLY7	FFTXDLY6	FFTXDLY5	FFTXDLY4	FFTXDLY3	FFTXDLY2	FFTXDLY1	FFTXDLY0
		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	MFFINT				Rese	erved			
					R	-0		r	
			Rese	erved		REVTA ENA	RINT ENA	XEVTA ENA	XINT ENA
			R-	-0	1	R/W-0	R/W-0	R/W-0	R/W-0
24	MFFST				Rese	erved			
					R	-0			
			Rese	erved		FSR Flag	EOBR Flag	FSX Flag	EOBX Flag
			R-	-0		R/W-x	R/W-0	R/W-x	R/W-0

Table 6-21. FIFO Register Bit Descriptions (Base address 0x00 7800) (Continued)

# Appendix A

# **Revision History**

This document was revised to accomodate the following technical changes as described on the next page.
## A.1 Changes Made in This Revision

The following changes were made in this revision:

Page	Additions/Modifications/Deletions			
Global	Global change to update signal names to TI nomenclature			
Global	Deleted all instances of "I/O-mapped registers."			
1-30	1-30 Keeping CLKG Synchronized to External MCLKR			
	Changed heading and first sentence to include "external MCLKR."			
1-33	Deleted sentence, "therefore, the CLK(R/X) pin must not be driven by any oher driving source.			
1-33	Deleted CLKXX from figures 1-21 to 1-25			
3-3	Added "provided the McBSP clock is turned on" to the end of the sentence.			

## **IMPORTANT NOTICE**

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

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

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

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

Reproduction of TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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

Products		Applications	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connctivity	www.ti.com/wirelessconnectivity		

**TI E2E Community Home Page** 

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2011, Texas Instruments Incorporated