TMS320C6000 DSP
Multichannel Audio Serial Port (McASP)

Reference Guide

Literature Number: SPRU041J
November 2008

Contents

List of Figures

List of Tables

Preface

SPRU041J–November 2008

Read This First

About This Manual

This document describes the multichannel audio serial port (McASP) in the digital signal processors (DSPs) of the TMS320C6000™ DSP family. The McASP functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT).

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion. Reads from this reserved bit always return the default value. Writes to this reserved bit are ignored. If writing to this field, always write the default value for future device compatibility.

31	
Reserved	Bit
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Related Documentation From Texas Instruments

The following documents describe the C6000™ devices and related support tools. Copies of these documents are available on the Internet at www.ti.com. *Tip*: Enter the literature number in the search box provided at www.ti.com.

The current documentation that describes the C6000 devices, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

SPRU732 — *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C64x and TMS320C64x+ digital signal processors (DSPs) of the TMS320C6000 DSP family. The C64x/C64x+ DSP generation comprises fixed-point devices in the C6000 DSP platform. The C64x+ DSP is an enhancement of the C64x DSP with added functionality and an expanded instruction set.

SPRU733 — *TMS320C67x/C67x+ DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C67x and TMS320C67x+ digital signal processors (DSPs) of the TMS320C6000 DSP platform. The C67x/C67x+ DSP generation comprises floating-point devices in the C6000 DSP platform. The C67x+ DSP is an enhancement of the C67x DSP with added functionality and an expanded instruction set.

SPRU190 — *TMS320C6000 DSP Peripherals Overview Reference Guide.* Provides an overview and briefly describes the peripherals available on the TMS320C6000 family of digital signal processors (DSPs).

SPRU197 — *TMS320C6000 Technical Brief.* Provides an introduction to the TMS320C62x and TMS320C67x digital signal processors (DSPs) of the TMS320C6000 DSP family. Describes the CPU architecture, peripherals, development tools and third-party support for the C62x and C67x DSPs.

SPRU395 — *TMS320C64x Technical Overview.* Provides an introduction to the TMS320C64x digital signal processors (DSPs) of the TMS320C6000 DSP family.

SPRU198 — *TMS320C6000 Programmer's Guide.* Reference for programming the TMS320C6000 digital signal processors (DSPs). Before you use this manual, you should install your code generation and debugging tools. Includes a brief description of the C6000 DSP architecture and code development flow, includes C code examples and discusses optimization methods for the C code, describes the structure of assembly code and includes examples and discusses optimizations for the assembly code, and describes programming considerations for the C64x DSP.

SPRU301 — *TMS320C6000 Code Composer Studio Tutorial.* This tutorial introduces you to some of the key features of Code Composer Studio. Code Composer Studio extends the capabilities of the Code Composer Integrated Development Environment (IDE) to include full awareness of the DSP target by the host and real-time analysis tools. This tutorial assumes that you have Code Composer Studio, which includes the TMS320C6000 code generation tools along with the APIs and plug-ins for both DSP/BIOS and RTDX. This manual also assumes that you have installed a target board in your PC containing the DSP device.

SPRU273 — *TMS320C6x Peripheral Support Library Programmer's Reference.* Describes the TMS320C6000 digital signal processor (DSP) peripheral support library of functions and macros. The C6000 DSP peripheral support library is a collection of macros and functions for programming the C6000 DSP registers and peripherals using the C programming language. This document serves as a reference for the C programmer in creating code for the C6000 DSP.

SPRU401 — *TMS320C6000 Chip Support Library API Reference Guide.* Describes the TMS320C6000 chip support library (CSL) that is a set of application programming interfaces (APIs) used to configure and control all on-chip peripherals. CSL is intended to make it easier for developers by eliminating much of the tedious work usually needed to get algorithms up and running in a real system.

Trademarks

TMS320C6000, C6000 are trademarks of Texas Instruments.

Overview

This chapter provides an overview of the multichannel audio serial port (McASP) in the digital signal processors (DSPs) of the TMS320C6000™ DSP family. Included are the features of the McASP, protocols the McASP supports, and definitions of terms used within this document.

The multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT).

The McASP consists of transmit and receive sections that may operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats. The McASP module also includes up to 16 serializers that can be individually enabled to either transmit or receive. In addition, all of the McASP pins can be configured as general-purpose input/output (GPIO) pins.

Topic **Page**

1.1 Features

Features of the McASP include:

- Two independent clock generator modules for transmit and receive
  - Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
- Independent transmit and receive modules, each includes:
  - Programmable clock and frame sync generator
  - TDM streams from 2 to 32, and 384 time slots
  - Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits
  - Data formatter for bit manipulation
- Individually assignable serial data pins (up to 16 pins)
- Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components
- Wide variety of I2S and similar bit-stream format
- Integrated digital audio interface transmitter (DIT) supports:
  - S/PDIF, IEC60958-1, AES-3 formats
  - Up to 16 transmit pins
  - Enhanced channel status/user data RAM
- 384-slot TDM with external digital audio interface receiver (DIR) device
  - For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
- Extensive error checking and recovery
  - Transmit underruns and receiver overruns due to the system not meeting real-time requirements
  - Early or late frame sync in TDM mode
  - Out-of-range high-frequency master clock for both transmit and receive
  - External error signal coming into the AMUTEIN input
  - DMA error due to incorrect programming

1.2 Protocols Supported

The McASP supports a wide variety of protocols.

- Transmit section supports
  - Wide variety of I2S and similar bit-stream formats
  - TDM streams from 2 to 32 time slots
  - S/PDIF, IEC60958-1, AES-3 formats
- Receive section supports
  - Wide variety of I2S and similar bit-stream formats
  - TDM streams from 2 to 32 time slots
  - TDM stream of 384 time slots specifically designed for easy interface to external digital interface receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe)

The transmit and receive sections may each be individually programmed to support the following options on the basic serial protocol:

- Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length

- First-bit data delay: 0, 1, 2 bit clocks
- Left/right alignment of word inside slot
- Bit order: MSB first or LSB first
- Bit mask/pad/rotate function
  - Automatically aligns data for DSP internally in either Q31 or integer formats
  - Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit)

In DIT mode, additional features of the transmitter are:

- Transmit-only mode- 384 time slots (subframe) per frame
- Bi-phase encoded 3.3 V output
- Support for consumer and professional applications
- Channel status RAM (384 bits)
- User data RAM (384 bits)
- Separate valid bit (V) for subframe A, B

On C6000 DSPs in I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels.

On C6000 DSPs in DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

1.3 System Level Connections

Figure 1-1 through Figure 1-5 show examples of McASP usage in digital audio encoder/decoder systems.

Figure 1-1. McASP to Parallel 2-Channel DACs

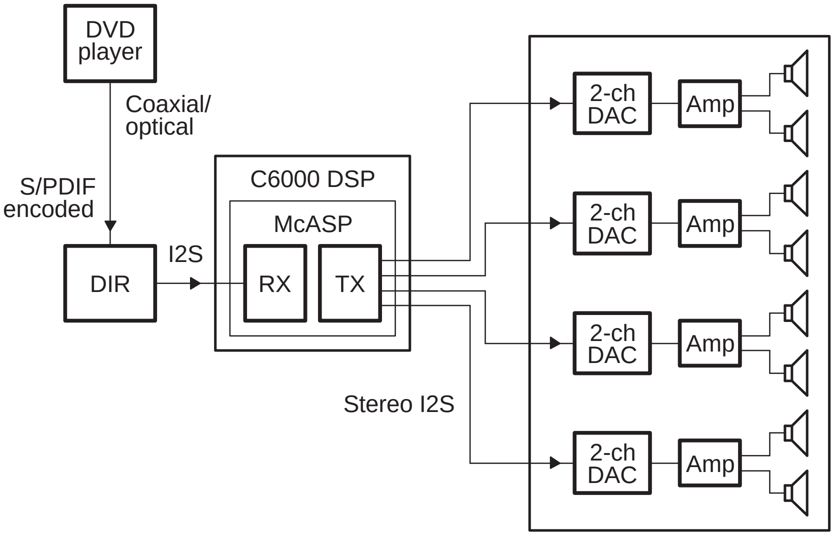


Figure 1-2. McASP to 6-Channel DAC and 2-Channel DAC

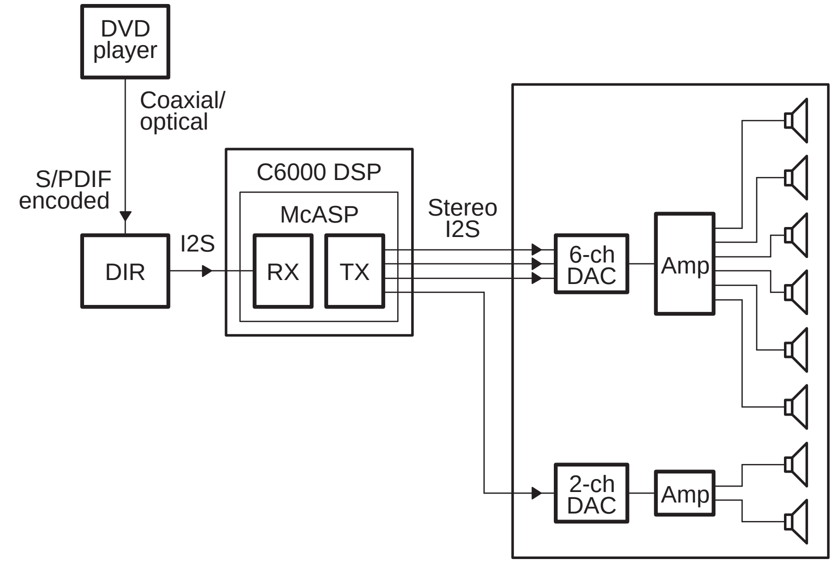


Figure 1-3. McASP to Digital Amplifier

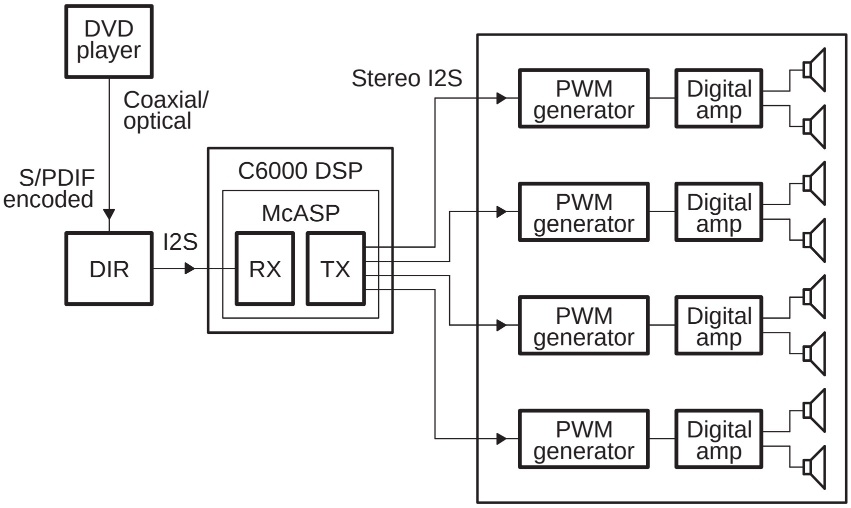


Figure 1-4. McASP as Digital Audio Encoder

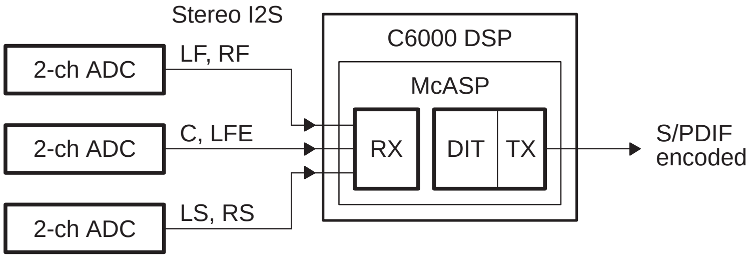
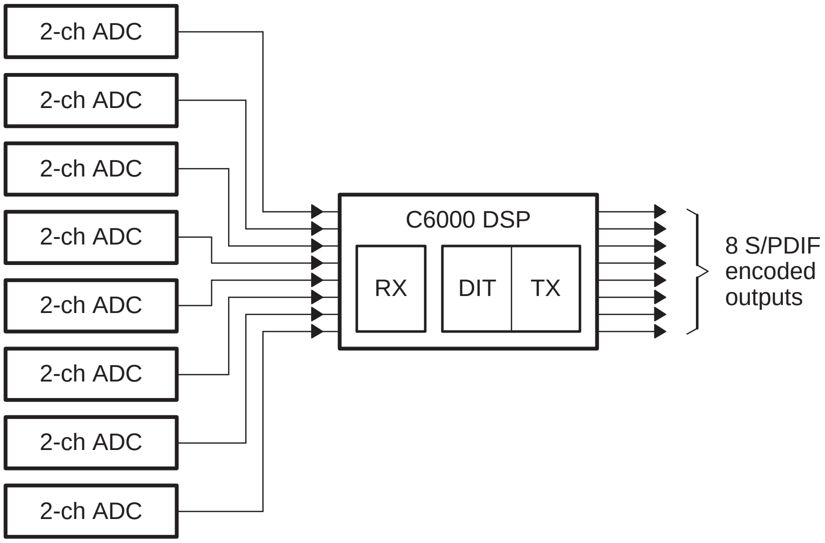


Figure 1-5. McASP as 16 Channel Digital Processor



1.4 Considerations When Using a McASP

The following is a list of things to be considered for systems using a McASP:

1.4.1 Clocks

For each receive and transmit section:

- External or internal generated bit clock and high frequency clock?
- If internally generated, what is the bit clock speed and the high frequency clock speed?
- Clock polarity?
- External or internal generated frame sync?
- If internally generated, what is frame sync speed?
- Frame sync polarity?
- Frame sync width?
- Transmit and receive sync or asynchronous?

1.4.2 Data Pins

For each pin of each McASP:

- McASP or GPIO?
- Input or output?

1.4.3 Data Format

For each transmit and receive data:

- Internal numeric representation (integer, Q31 fraction)?
- I2S or DIT (transmit only)?
- Time slot delay (0, 1, or 2 bit)?
- Alignment (left or right)?
- Order (MSB first, LSB first)?
- Pad (if yes, pad with what value)?
- Slot size?

- Rotate?
- Mask?

1.4.4 Data Transfers

- Internal: DMA or CPU?
- External: TDM or burst?
- Bus: configuration bus (CFG) or data port (DAT)?

1.5 Definition of Terms

The serial bit stream transmitted or received by the McASP is a long sequence of 1s and 0s, either output or input on one of the audio transmit/receive pins (AXR[n]). However, the sequence has a hierarchical organization that can be described in terms of frames of data, slots, words, and bits.

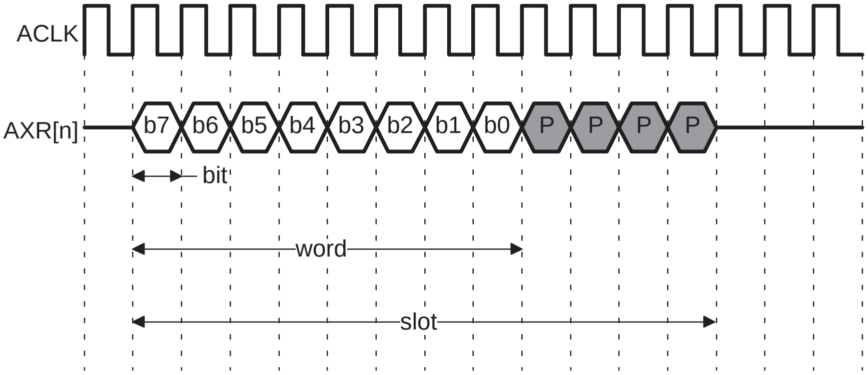
A basic synchronous serial interface consists of three important components: clock, frame sync, and data. Figure 1-6 shows two of the three basic components—the clock (ACLK) and the data (AXR[n]). Figure 1-6 does not specify whether the clock is for transmit (ACLKX) or receive (ACLKR) because the definitions of terms apply to both receive and transmit interfaces. In operation, the transmitter uses ACLKX as the serial clock, and the receiver uses ACLKR as the serial clock. Optionally, the receiver can use ACLKX as the serial clock when the transmitter and receiver of the McASP are configured to operate synchronously.

Bit A bit is the smallest entity in the serial data stream. The beginning and end of each bit is marked by an edge of the serial clock. The duration of a bit is a serial clock period. A 1 is represented by a logic high on the AXR[n] pin for the entire duration of the bit. A 0 is represented by a logic low on the AXR[n] pin for the entire duration of the bit.

Word A word is a group of bits that make up the data being transferred between the DSP and the external device. Figure 1-6 shows an 8-bit word.

Slot A slot consists of the bits that make up the word, and may consist of additional bits used to pad the word to a convenient number of bits for the interface between the DSP and the external device. In Figure 1-6, the audio data consists of only 8 bits of useful data (8-bit word), but it is padded with 4 zeros (12-bit slot) to satisfy the desired protocol in interfacing to an external device. Within a slot, the bits may be shifted in/out of the McASP on the AXR[n] pin either MSB or LSB first. When the word size is smaller than the slot size, the word may be aligned to the left (beginning) of the slot or to the right (end) of the slot. The additional bits in the slot not belonging to the word may be padded with 0, 1, or with one of the bits (the MSB or the LSB typically) from the data word. These options are shown in Figure 1-7.

Figure 1-6. Definition of Bit, Word, and Slot

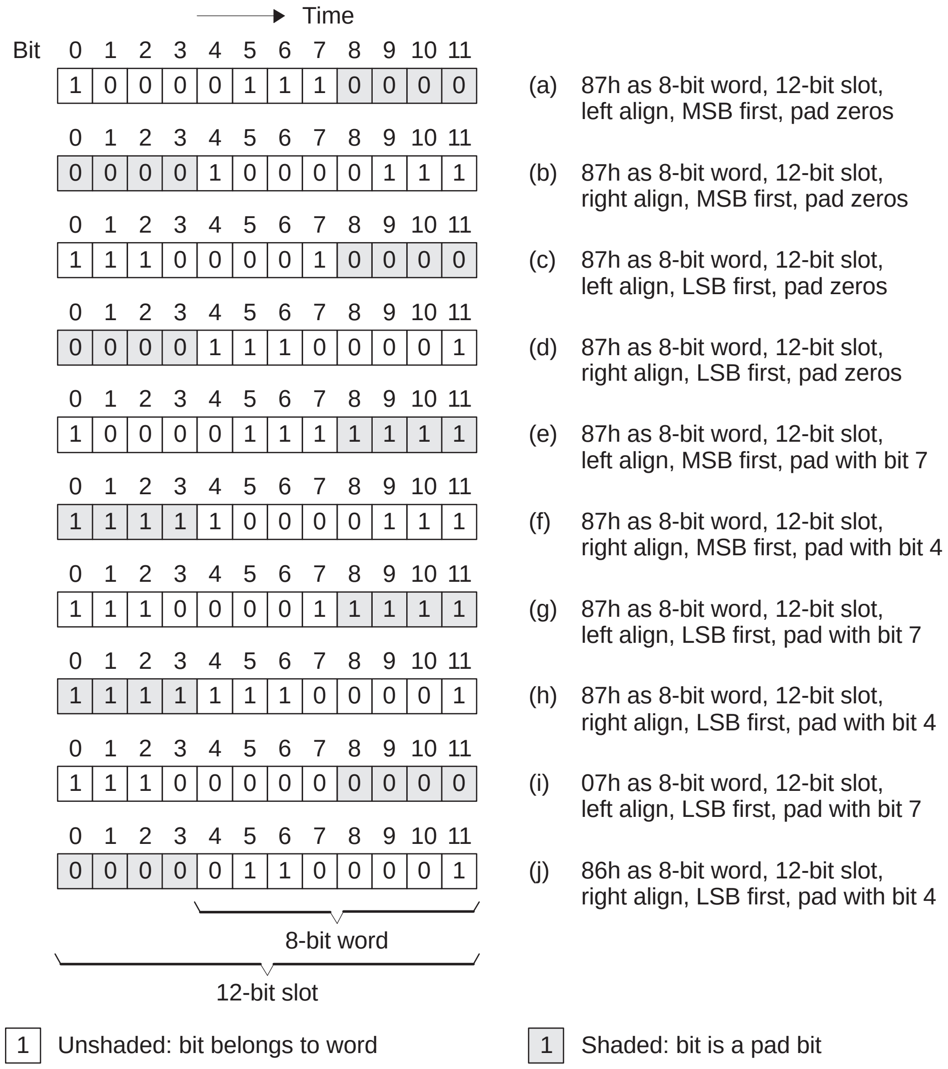


(1) b7:b0 - bits. Bits b7 to b0 form a word.

(2) P - pad bits. Bits b7 to b0, together with the four pad bits, form a slot.

(3) In this example, the data is transmitted MSB first, left aligned.

Figure 1-7. Bit Order and Word Alignment Within a Slot Examples



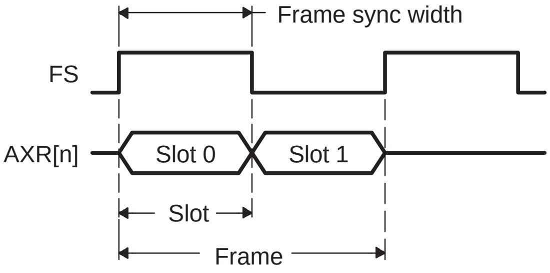
Bit	0	1	2	3	4	5	6	7	8	9	10	11	
	1	0	0	0	0	1	1	1	0	0	0	0	(a) 87h as 8-bit word, 12-bit slot, left align, MSB first, pad zeros
	0	0	0	0	1	0	0	0	0	1	1	1	(b) 87h as 8-bit word, 12-bit slot, right align, MSB first, pad zeros
	1	1	1	0	0	0	0	1	0	0	0	0	(c) 87h as 8-bit word, 12-bit slot, left align, LSB first, pad zeros
	0	0	0	0	1	1	1	0	0	0	0	1	(d) 87h as 8-bit word, 12-bit slot, right align, LSB first, pad zeros
	1	0	0	0	0	1	1	1	1	1	1	1	(e) 87h as 8-bit word, 12-bit slot, left align, MSB first, pad with bit 7
	1	1	1	1	1	0	0	0	0	1	1	1	(f) 87h as 8-bit word, 12-bit slot, right align, MSB first, pad with bit 4
	1	1	1	0	0	0	0	1	1	1	1	1	(g) 87h as 8-bit word, 12-bit slot, left align, LSB first, pad with bit 7
	1	1	1	1	1	1	1	0	0	0	0	1	(h) 87h as 8-bit word, 12-bit slot, right align, LSB first, pad with bit 4
	1	1	1	0	0	0	0	0	0	0	0	0	(i) 07h as 8-bit word, 12-bit slot, left align, LSB first, pad with bit 7
	0	0	0	0	0	1	1	0	0	0	0	1	(j) 86h as 8-bit word, 12-bit slot, right align, LSB first, pad with bit 4

The third basic element of a synchronous serial interface is the frame synchronization signal, also referred to as frame sync in this document.

Frame A frame contains one or multiple slots, as determined by the desired protocol. Figure 1-8 shows an example frame of data and the frame definitions. Figure 1-8 does not specify whether the frame sync (FS) is for transmit (AFSX) or receive (AFSR) because the definitions of terms apply to both receive and transmit interfaces. In operation, the transmitter uses AFSX and the receiver uses AFSR. Optionally, the receiver can use AFSX as the frame sync when the transmitter and receiver of the McASP are configured to operate synchronously.

This section only shows the generic definition of the frame sync. See Section 1.6, Section 1.7, and Section 3.2.1 for details on the frame sync formats required for the different transfer modes and protocols (burst mode, TDM mode and I2S format, DIT mode and S/PDIF format).

Figure 1-8. Definition of Frame and Frame Sync Width



(1) In this example, there are two slots in a frame, and FS duration of slot length is shown.

Other terms used throughout the document:

TDM Time-division multiplexed. See Section 1.6 for details on the TDM protocol.

DIR Digital audio interface receive. The McASP does not natively support receiving in the S/PDIF format. The McASP supports I2S format output by an external DIR device.

DIT Digital audio interface transmit. The McASP supports transmitting in S/PDIF format on up to all data pins configured as outputs.

I2S Inter-Integrated Sound protocol, commonly used on audio interfaces. The McASP supports the I2S protocol as part of the TDM mode (when configured as a 2-slot frame).

Slot or Time Slot For TDM format, the term time slot is interchangeable with the term slot defined in this section. For DIT format, a McASP time slot corresponds to a DIT subframe.

1.6 TDM Format

The McASP transmitter and receiver support the multichannel, synchronous time-division-multiplexed (TDM) format via the TDM transfer mode. Within this transfer mode, a wide variety of serial data formats are supported, including formats compatible with devices using the Inter-Integrated Sound (I2S) protocol. This section briefly discusses the TDM format and the I2S protocol.

1.6.1 TDM Format

The TDM format is typically used when communicating between integrated circuit devices on the same printed circuit board or on another printed circuit board within the same piece of equipment. For example, the TDM format is used to transfer data between the DSP and one or more analog-to-digital converter (ADC), digital-to-analog converter (DAC), or S/PDIF receiver (DIR) devices.

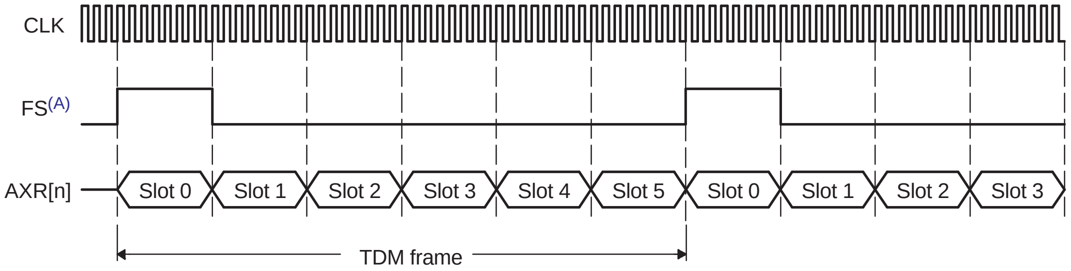
The TDM format consists of three components in a basic synchronous serial transfer: the clock, the data, and the frame sync. In a TDM transfer, all data bits (AXR[n]) are synchronous to the serial clock (ACLKX or ACLKR). The data bits are grouped into words and slots (as defined in Section 1.5). The "slots" are also commonly referred to as "time slots" or "channels" in TDM terminology. A frame consists of multiple slots (or channels). Each TDM frame is defined by the frame sync signal (AFSX or AFSR). Data transfer is continuous and periodic, since the TDM format is most commonly used to communicate with data converters that operate at a fixed sample rate.

There are no delays between slots. The last bit of slot N is followed immediately on the next serial clock cycle with the first bit of slot N + 1, and the last bit of the last slot is followed immediately on the next serial clock with the first bit of the first slot. However, the frame sync may be offset from the first bit of the first slot with a 0, 1, or 2-cycle delay.

It is required that the transmitter and receiver in the system agree on the number of bits per slot, since the determination of a slot boundary is not made by the frame sync signal (although the frame sync marks the beginning of slot 0 and the beginning of a new frame).

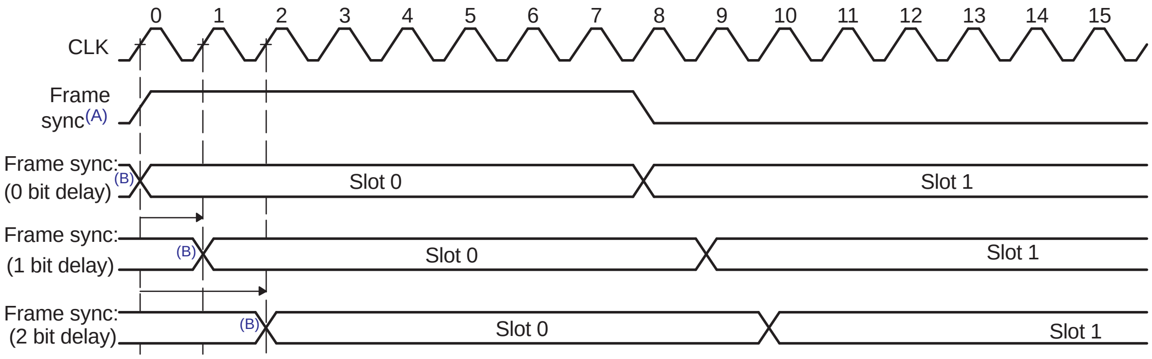
Figure 1-9 shows the TDM format. Figure 1-10 shows the different bit delays from the frame sync.

Figure 1-9. TDM Format–6 Channel TDM Example



A FS duration of slot is shown. FS duration of single bit is also supported.

Figure 1-10. TDM Format Bit Delays from Frame Sync



A FS duration of slot is shown. FS duration of single bit is also supported.

B Last bit of last slot of previous frame. No gap between this bit and the first bit of slot 0 is allowed.

In a typical audio system, one frame of data is transferred during each data converter sample period f_S. To support multiple channels, the choices are to either include more time slots per frame (thus operating with a higher bit clock rate), or to use additional data pins to transfer the same number of channels (thus operating with a slower bit clock rate).

For example, a particular six channel DAC may be designed to transfer over a single serial data pin AXR[n] as shown in Figure 1-9. In this case the serial clock must run fast enough to transfer a total of 6 channels within each frame period. Alternatively, a similar six channel DAC may be designed to use three serial data pins AXR[0,1,2], transferring two channels of data on each pin during each sample period (Figure 1-11). In the latter case, if the sample period remains the same, the serial clock can run three times slower than the former case. The McASP is flexible enough to support either type of DAC.

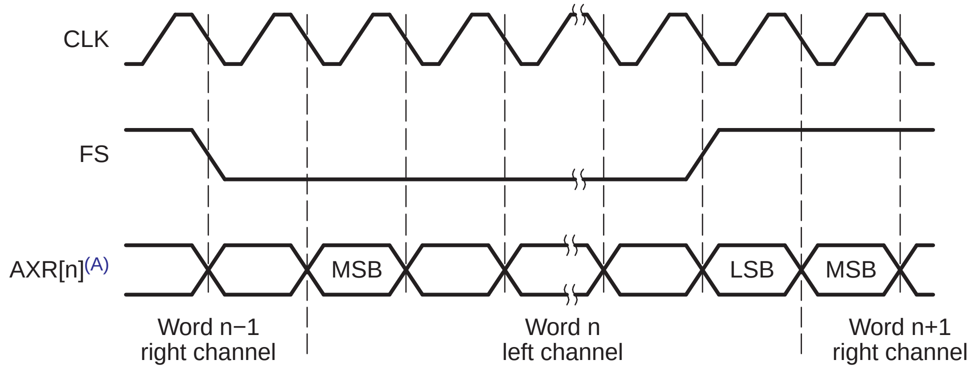
1.6.2 Inter-Integrated Sound (I2S) Format

The inter-integrated sound (I2S) format is used extensively in audio interfaces. The TDM transfer mode of the McASP supports the I2S format when configured to 2 slots per frame.

I2S format is specifically designed to transfer a stereo channel (left and right) over a single data pin AXR[n]. "Slots" are also commonly referred to as "channels". The frame width duration in the I2S format is the same as the slot size. The frame signal is also referred to as "word select" in the I2S format. Figure 1-11 shows the I2S protocol.

The McASP supports transfer of multiple stereo channels over multiple AXR[n] pins.

Figure 1-11. Inter-Integrated Sound (I2S) Format



A 1 to 16 data pins may be supported.

1.7 S/PDIF Coding Format

The McASP transmitter supports the S/PDIF format with 3.3V biphase-mark encoded output. The S/PDIF format is supported by the digital audio interface transmit (DIT) transfer mode of the McASP. This section briefly discusses the S/PDIF coding format.

1.7.1 Biphase-Mark Code (BMC)

In S/PDIF format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin AXR[n]. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. These two logical states form a cell. The duration of the cell, which equals to the duration of the data bit, is called a time interval. A logical 1 is represented by two transitions of the signal within a time interval, which corresponds to a cell with logical states 01 or 10. A logical 0 is represented by one transition within a time interval, which corresponds to a cell with logical states 00 or 11. In addition, the logical level at the start of a cell is inverted from the level at the end of the previous cell. Figure 1-12 and Table 1-1 show how data is encoded to the BMC format.

As shown in Figure 1-12, the frequency of the clock is twice the unencoded data bit rate. In addition, the clock is always programmed to 128 x̲nbsp;f_S, where f_s is the sample rate (see Section 1.7.3 for details on how this clock rate is derived based on the S/PDIF format). The device receiving in S/PDIF format can recover the clock and frame information from the BMC signal.

Figure 1-12. Biphase-Mark Code (BMC)

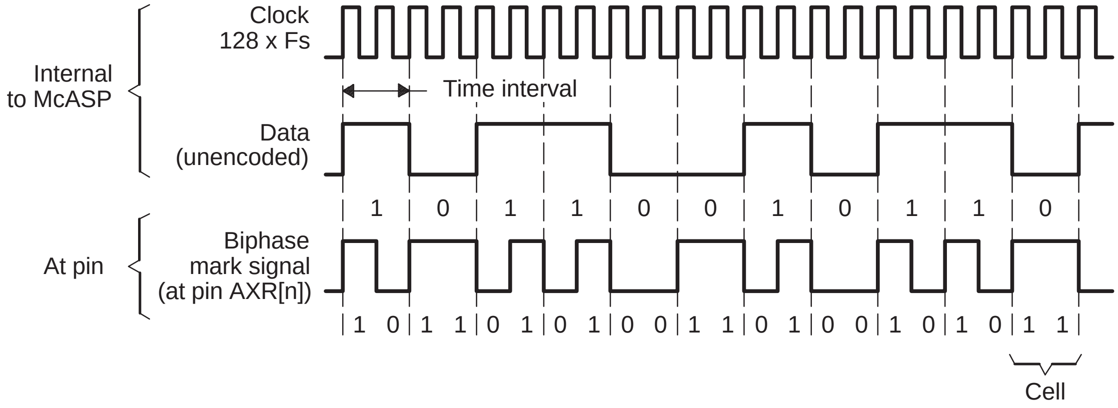


Table 1-1. Biphase-Mark Encoder

Data (Unencoded)	Previous State at Pin AXR[n]	BMC-Encoded Cell Output at AXR[n]
0	0	11
0	1	00
1	0	10
1	1	01

1.7.2 Subframe Format

Every audio sample transmitted in a subframe consists of 32 S/PDIF time intervals (or cells), numbered from 0 to 31. Figure 1-13 shows a subframe.

- **Time intervals 0-3** carry one of the three permitted preambles to signify the type of audio sample in the current subframe. The preamble is *not* encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See Table 1-2.
- **Time intervals 4-27** carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by time interval 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in time interval 4. When a 20-bit coding range is used, time intervals 8-27 carry the audio sample word with the LSB in time interval 8. Time intervals 4-7 may be used for other applications and are designated auxiliary sample bits.
- If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.
- **Time interval 28** carries the validity bit (V) associated with the main data field in the subframe.
- **Time interval 29** carries the user data channel (U) associated with the main data field in the subframe.
- **Time interval 30** carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.
- **Time interval 31** carries a parity bit (P) such that time intervals 4-31 carry an even number of 1s and an even number of 0s (even parity). As shown in Table 1-2, the preambles (time intervals 0-3) are also defined with even parity.

As shown in Table 1-2, the McASP DIT only generates one polarity of preambles and it assumes the previous logical state to be 0. This is because the McASP assures an even-polarity encoding scheme when transmitting in DIT mode. If an underrun condition occurs, the DIT resynchronizes to the correct logic level on the AXR[n] pin before continuing with the next transmission.

Figure 1-13. S/PDIF Subframe Format

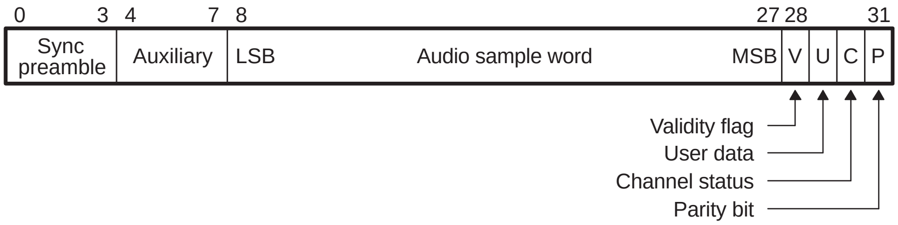


Table 1-2. Preamble Codes

Preamble Code[1]	Previous Logical State	Logical States on pin AXR[n][2]	Description
B (or Z)	0	1110 1000	Start of a block and subframe 1

[1] Historically, preamble codes are referred to as B, M, W. For use in professional applications, preambles are referred to as Z, X, Y, respectively.

[2] The preamble is not BMC encoded. Each logical state is synchronized to the serial clock. These 8 logical states make up time slots (cells) 0 to 3 in the S/PDIF stream.

Table 1-2. Preamble Codes (continued)

Preamble Code[(1)]	Previous Logical State	Logical States on pin AXR[n][(2)]	Description
M (or X)	0	1110 0010	Subframe 1
W (or Y)	0	1110 0100	Subframe 2

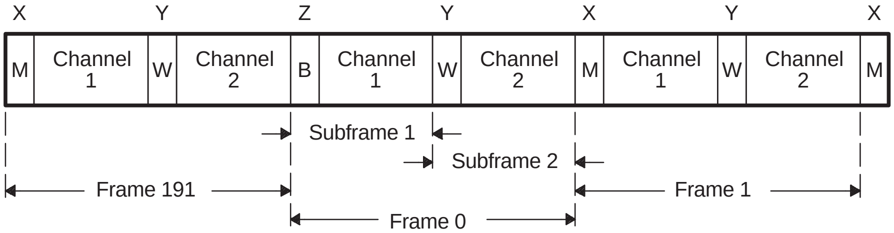
1.7.3 Frame Format

An S/PDIF frame is composed of two subframes (Figure 1-14). For linear coded audio applications, the rate of frame transmission normally corresponds exactly to the source sampling frequency f_S. The S/PDIF format clock rate is therefore 128 ɤnbsp;f_S (128 = 32 cells/subframe ɤnbsp;2 clocks/cell ɤnbsp;2 subframes/sample). For example, for an S/PDIF stream at a 192 kHz sampling frequency, the serial clock is 128 ɤnbsp;192 kHz = 24.58 MHZ.

In 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive subframes. Both subframes contain valid data. The first subframe (**left** or **A** channel in stereophonic operation and **primary** channel in monophonic operation) normally starts with preamble M. However, the preamble of the first subframe changes to preamble B once every 192 frames to identify the start of the block structure used to organize the channel status information. The second subframe (**right** or **B** channel in stereophonic operation and **secondary** channel in monophonic operation) always starts with preamble W.

In single-channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried in the first subframe and may be duplicated in the second subframe. If the second subframe is not carrying duplicate data, cell 28 (validity bit) is set to logical 1.

Figure 1-14. S/PDIF Frame Format



Architecture

This chapter discusses the architecture of the functional units of the McASP.

Topic Page

2.1 Overview

Figure 2-1 shows the major blocks of the McASP. The McASP has independent receive/transmit clock generators and frame sync generators, error-checking logic, and up to 16 serial data pins. Refer to the device-specific data manual for the number of data pins available on your device.

All the McASP pins on the device may be individually programmed as general-purpose I/O (GPIO) if they are not used for serial port functions.

The McASP includes the following pins:

- Serializers
  - Data pins AXR[n]: Up to sixteen per McASP
- Transmit clock generator:
  - AHCLKX: McASP transmit high-frequency master clock
  - ACLKX: McASP transmit bit clock
- Transmit Frame Sync Generator
  - AFSX: McASP transmit frame sync or left/right clock (LRCLK)
- Receive clock generator:
  - AHCLKR: McASP receive high-frequency master clock
  - ACLKR: McASP receive bit clock
- Receive Frame Sync Generator
  - AFSR: McASP receive frame sync or left/right clock (LRCLK)
- Mute in/out:
  - AMUTEIN: McASP mute input (from external device)
  - AMUTE: McASP mute output
  - Data pins AXR[n]

2.2 Clock and Frame Sync Generators

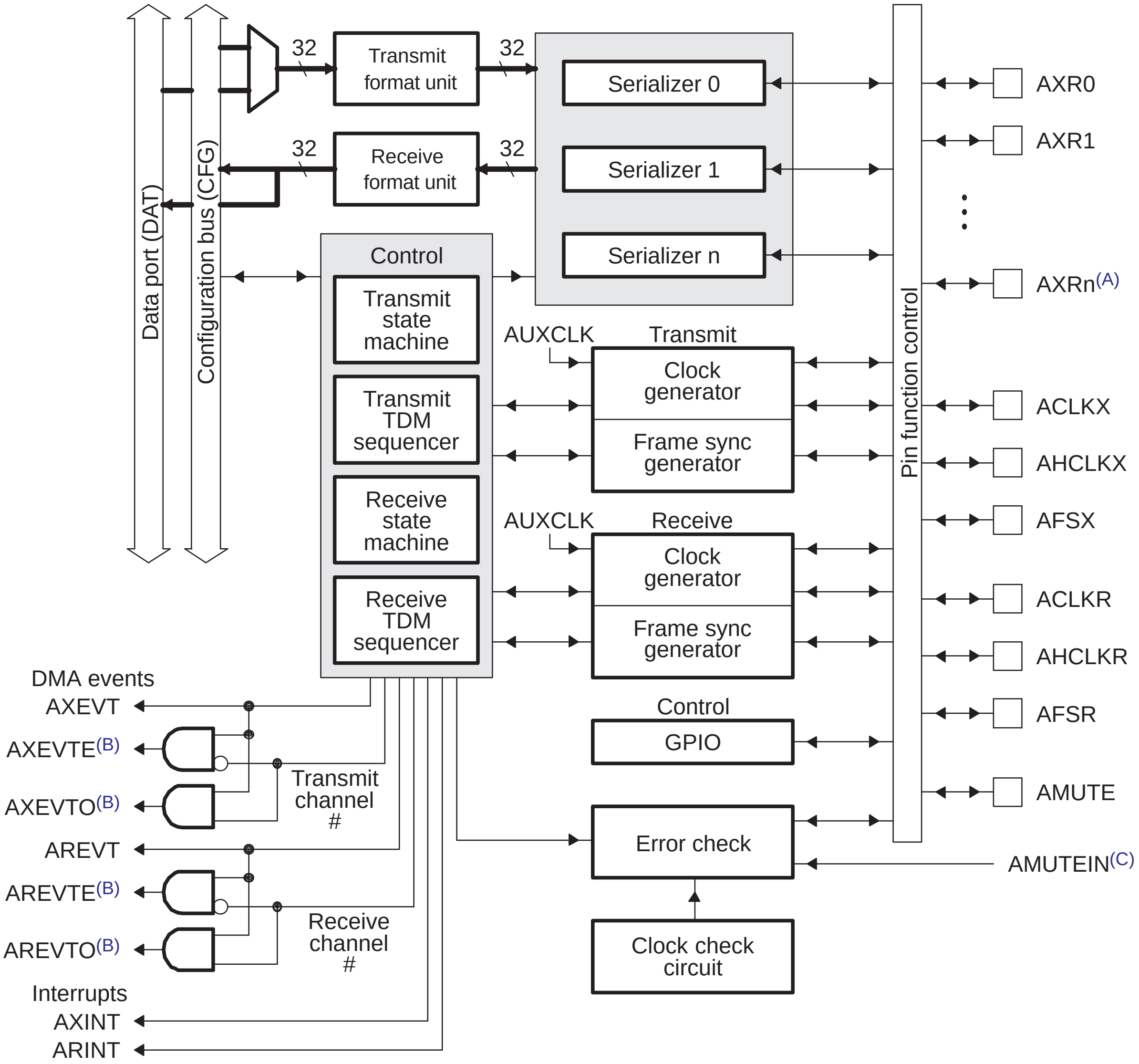
The McASP clock generators are able to produce two independent clock zones: transmit and receive clock zones. The serial clock generators may be programmed independently for the transmit section and the receive section, and may be completely asynchronous to each other. The serial clock (clock at the bit rate) may be sourced:

- Internally - by passing through two clock dividers off the internal clock source (AUXCLK)
- Externally - directly from ACLKR/X pin
- Mixed - an external high-frequency clock is input to the McASP on either the AHCLKX or AHCLKR pins, and divided down to produce the bit rate clock

In the internal/mixed cases, the bit rate clock is generated internally and should be driven out on the ACLKX (for transmit) or ACLKR (for receive) pins. In the internal case, an internally-generated high-frequency clock may be driven out onto the AHCLKX or AHCLKR pins to serve as a reference clock for other components in the system.

The McASP requires a minimum of a bit clock and a frame sync to operate, and provides the capability to reference these clocks from an external high-frequency master clock. In DIT mode, it is possible to use only internally-generated clocks and frame syncs.

Figure 2-1. McASP Block Diagram



A On DA6x DSP, McASP has 16 serial data pins, *n* = 15.
On C6000 DSP, McASP has 8 serial data pins, *n* = 7.

B Check the device data manual to see if AXEVTO/AREVTO and AXEVTE/AREVTE are supported. These signals are optional.

C AMUTEIN is not a dedicated McASP pin, but typically comes from one of the DSP's external interrupt pins.

2.2.1 Transmit Clock

The transmit bit clock, ACLKX, (Figure 2-2) may be either externally sourced from the ACLKX pin or internally generated, as selected by the CLKXM bit. If internally generated (CLKXM = 1), the clock is divided down by a programmable bit clock divider (CLKXDIV) from the transmit high-frequency master clock (AHCLKX).

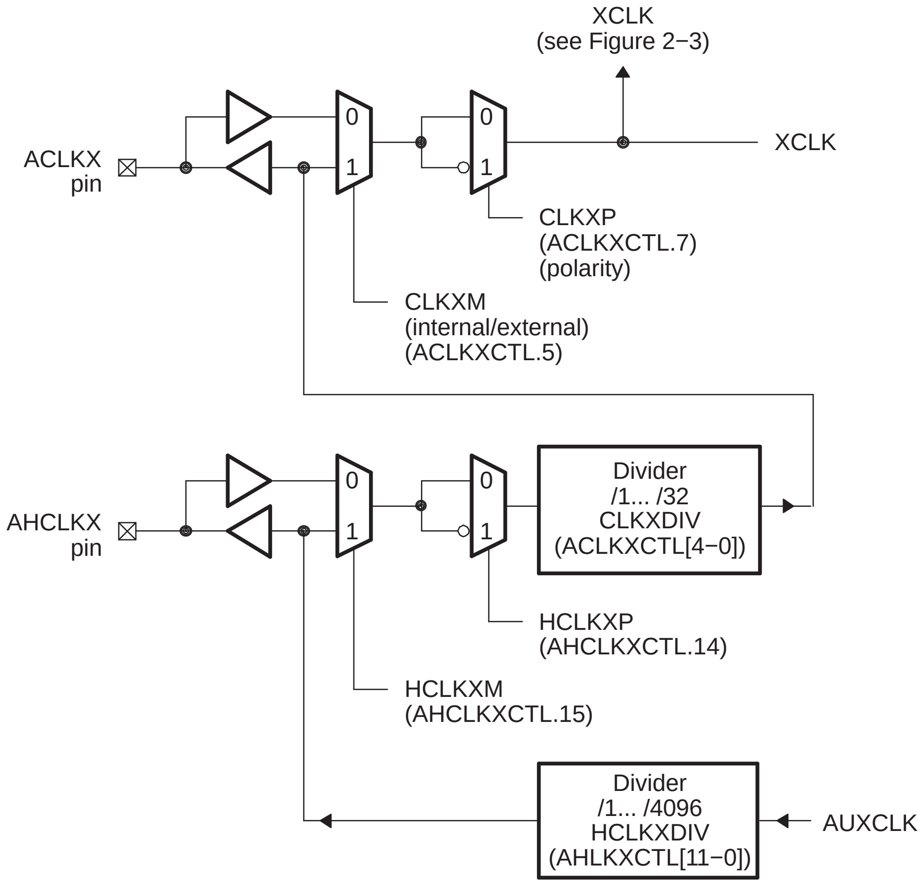
Internally, the McASP always shifts transmit data at the rising edge of the internal transmit clock, XCLK, (Figure 2-2). The CLKXP mux determines if ACLKX needs to be inverted to become XCLK. If CLKXP = 0, the CLKXP mux directly passes ACLKX to XCLK. As a result, the McASP shifts transmit data at the rising edge of ACLKX. If CLKXP = 1, the CLKX mux passes the inverted version of ACLKX to XCLK. As a result, the McASP shifts transmit data at the falling edge of ACLKX.

The transmit high-frequency master clock, AHCLKX, may be either externally sourced from the AHCLKX pin or internally generated, as selected by the HCLKXM bit. If internally generated (HCLKXM = 1), the clock is divided down by a programmable high clock divider (HCLKXDIV) from McASP internal clock source AUXCLK. The transmit high-frequency master clock may be (but is not required to be) output on the AHCLKX pin where it is available to other devices in the system.

The transmit clock configuration is controlled by the following registers:

- ACLKXCTL
- AHCLKXCTL

Figure 2-2. Transmit Clock Generator Block Diagram



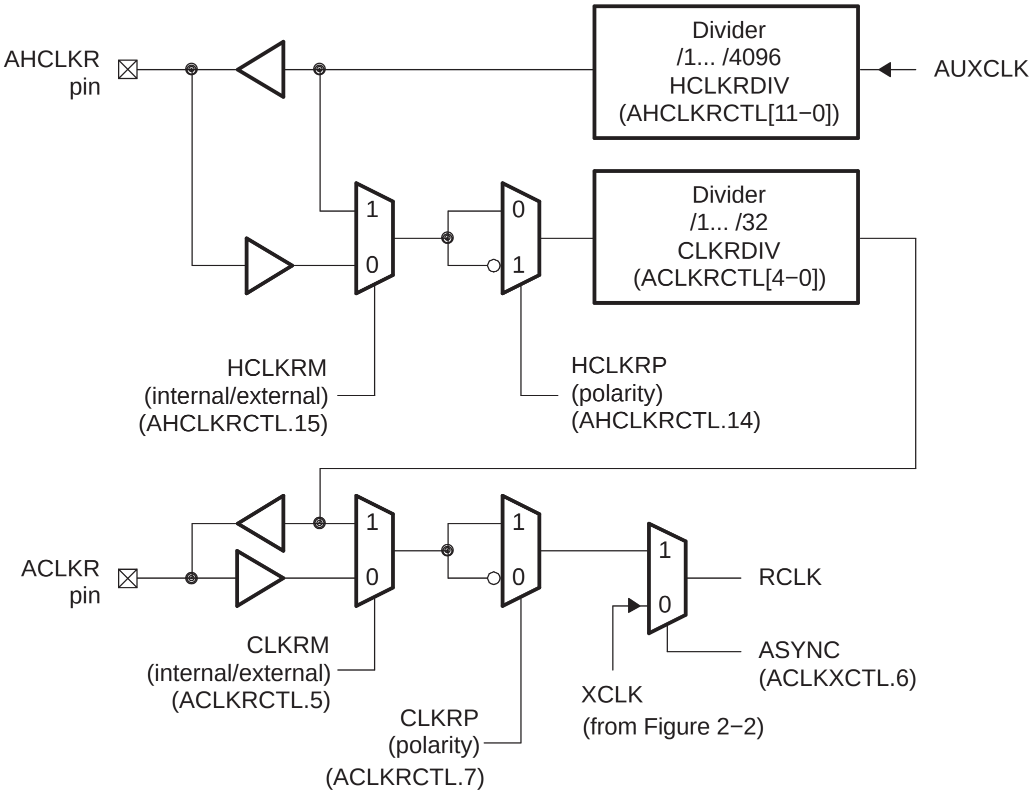
2.2.2 Receive Clock

The receiver also has the option to operate synchronously from the ACLKX and AFSX signals. This is achieved when the ASYNC bit in the transmit clock control register (ACLKXCTL) is cleared to 0 (see Figure 2-3). The receiver may be configured with different polarity (CLKRP) and frame sync data delay options from those options of the transmitter.

The receive clock configuration is controlled by the following registers:

- ACLKRCTL
- AHCLKRCTL

Figure 2-3. Receive Clock Generator Block Diagram



2.2.3 Frame Sync Generator

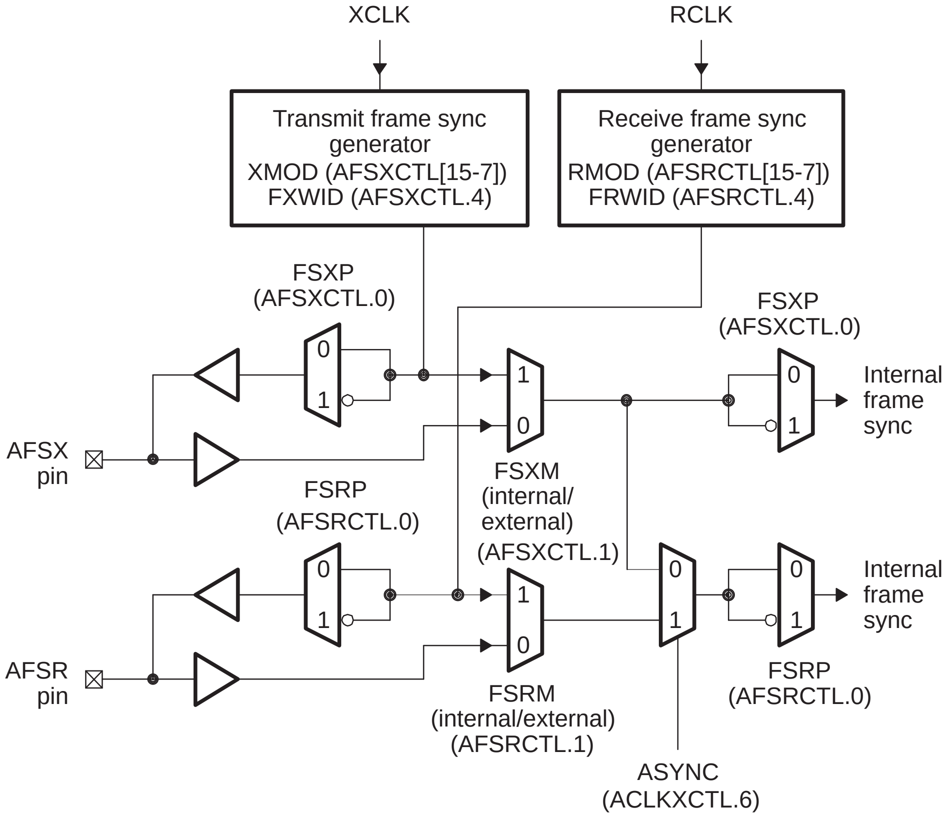
There are two different modes for frame sync: burst and TDM. A block diagram of the frame sync generator is shown in Figure 2-4. The frame sync options are programmed by the receive and transmit frame sync control registers (AFSRCTL and AFSXCTL). The options are:

- Internally-generated or externally-generated
- Frame sync polarity: rising edge or falling edge
- Frame sync width: single bit or single word
- Bit delay: 0, 1, or 2 cycles before the first data bit

The transmit frame sync pin is AFSX and the receive frame sync pin is AFSR. A typical usage for these pins is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.

Regardless if the AFSX/AFSR is internally generated or externally sourced, the polarity of AFSX/AFSR is determined by FSXP/FSRP, respectively, to be either rising or falling edge. If FSXP/FSRP = 0, the frame sync polarity is rising edge. If FSXP/FSRP = 1, the frame sync polarity is falling edge.

Figure 2-4. Frame Sync Generator Block Diagram



2.2.4 Clocking Examples

Some examples of processes using the McASP clocking and frame flexibility are:

- Receive data from a DVD at 48 kHz, but output up-sampled or decoded audio at 96 kHz or 192 kHz. This could be accomplished by inputting a high-frequency master clock (for example, 512 x̲nbsp;receive FS), receiving with an internally-generated bit clock ratio of divide-by-8, and transmitting with an internally-generated bit clock ratio of divide-by-4 or divide-by-2.
- Transmit/receive data based on one sample rate (for example, 44.1 kHz), and transmit/receive data at a different sample rate (for example, 48 kHz).

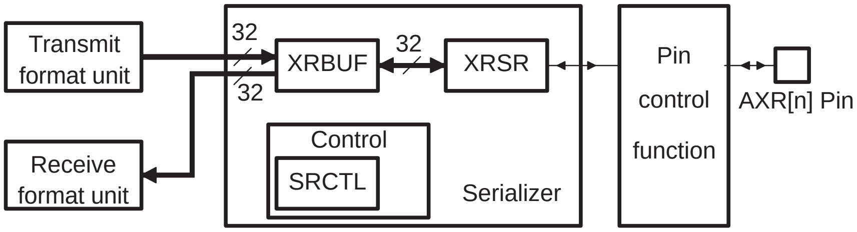
2.3 Serializers

The serializers take care of shifting serial data in and out of the McASP. Each serializer consists of a shift register (XRSR), a data buffer (XRBUF), a control register (SRCTL), and logic to support the data alignment options of the McASP. For each serializer, there is a dedicated serial data pin (AXR[n]) and a dedicated control register (SRCTL[n]). The control register allows the serializer to be configured as a transmitter, receiver, or as inactive. When configured as a transmitter the serializer shifts out data to the serial data pin AXR[n]. When configured as a receiver, the serializer shifts in data from the AXR[n] pin. The serializer is clocked from the transmit/receive section clock (ACLKX/ACLKR) if configured to transmit/receive respectively.

All serializers that are configured to transmit operate in lock-step. Similarly, all serializers that are configured to receive also operate in lock-step. This means that at most there are two zones per McASP, one for transmit and one for receive.

Figure 2-5 shows the block diagram of the serializer and its interface to other units within the McASP.

Figure 2-5. Individual Serializer and Connections Within McASP



For receive, data is shifted in through the AXR[n] pin to the shift register XRSR. Once the entire slot of data is collected in the XRSR, the data is copied to the data buffer XRBUF. The data is now ready to be read by the DSP through the RBUF register, which is an alias of the XRBUF for receive. When the DSP reads from the RBUF, the McASP passes the data from RBUF through the receive format unit and returns the formatted data to the DSP.

For transmit, the DSP services the McASP by writing data into the XBUF register, which is an alias of the XRBUF for transmit. The data automatically passes through the transmit format unit before actually reaching the XRBUF register in the serializer. The data is then copied from XRBUF to XRSR, and shifted out from the AXR[n] synchronously to the serial clock.

In DIT mode, in addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.).

The serializer configuration is controlled by SRCTL[n].

2.4 Format Unit

The McASP has two data formatting units, one for transmit and one for receive. These units automatically remap the data bits within the transmitted and received words between a natural format for the DSP (such as a Q31 representation) and the required format for the external serial device (such as "I2S format"). During the remapping process, the format unit also can mask off certain bits or perform sign extension.

Since all transmitters share the same data formatting unit, the McASP only supports one transmit format at a time. For example, the McASP will not transmit in "I2S format" on serializer 0, while transmitting "Left Justified" on serializer 1. Likewise, the receiver section of the McASP only supports one data format at a time, and this format applies to all receiving serializers. However, the McASP can transmit in one format while receiving in a completely different format.

This formatting unit consists of three stages:

- Bit mask and pad (masks off bits, performs sign extension)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB first or LSB first)

Figure 2-6 shows a block diagram of the receive formatting unit, and Figure 2-7 shows the transmit formatting unit. Note that the order in which data flows through the three stages is different between the transmit and receive formatting units.

Figure 2-6. Receive Format Unit

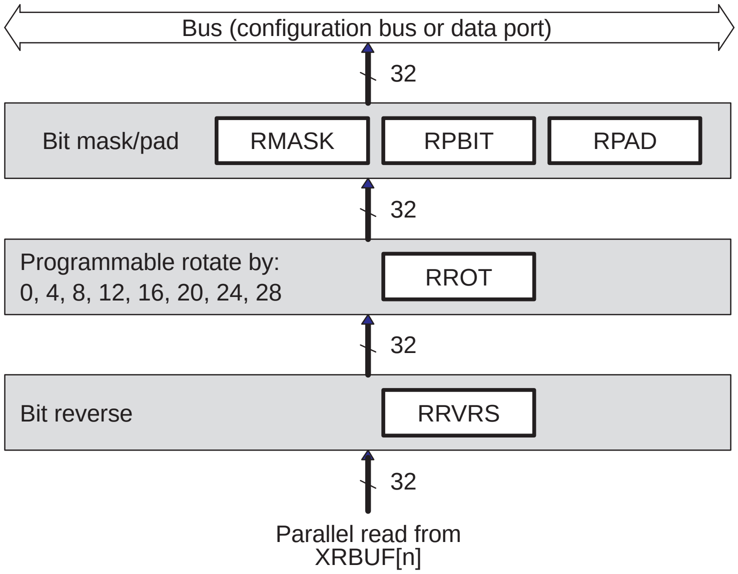
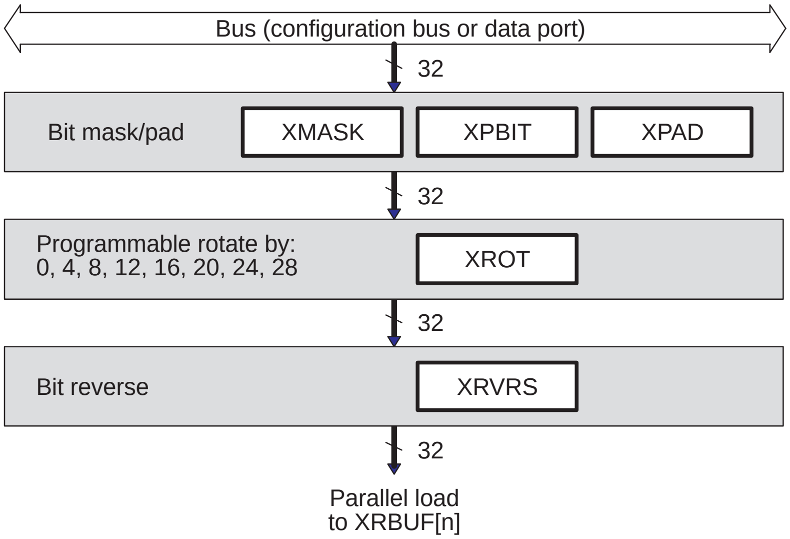


Figure 2-7. Transmit Format Unit



The bit mask and pad stage includes a full 32-bit mask register, allowing selected individual bits to either pass through the stage unchanged, or be masked off. The bit mask and pad then pad the value of the masked off bits by inserting either a 0, a 1, or one of the original 32 bits as the pad value. The last option allows for sign-extension when the sign bit is selected to pad the remaining bits.

The rotate right stage performs bitwise rotation by a multiple of 4 bits (between 0 and 28 bits), programmable by the (R/X)FMT register. Note that this is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the rotation.

The bit reversal stage either passes all 32 bits directly through, or swaps them. This allows for either MSB or LSB first data formats. If bit reversal is not enabled, then the McASP will naturally transmit and receive in an LSB first order.

Finally, note that the (R/X)DATDLY bits in (R/X)FMT also determine the data format. For example, the difference between I2S format and left-justified is determined by the delay between the frame sync edge and the first data bit of a given time slot. For I2S format, (R/X)DATDLY should be set to a 1-bit delay, whereas for left-justified format, it should be set to a 0-bit delay.

The combination of all the options in (R/X)FMT means that the McASP supports a wide variety of data formats, both on the serial data lines, and in the internal DSP representation.

Section 3.4 provides more detail and specific examples. The examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

2.5 State Machine

The receive and transmit sections have independent state machines. Each state machine controls the interactions between the various units in the respective section. In addition, the state machine keeps track of error conditions and serial port status.

No serial transfers can occur until the respective state machine is released from reset. See initialization sequence for details (Section 3.1).

The receive state machine is controlled by the RFMT register, and it reports the McASP status and error conditions in the RSTAT register. Similarly, the transmit state machine is controlled by the XFMT register, and it reports the McASP status and error conditions in the XSTAT register.

2.6 TDM Sequencer

There are separate TDM sequencers for the transmit section and the receive section. Each TDM sequencer keeps track of the slot count. In addition, the TDM sequencer checks the bits of (R/X)TDM and determines if the McASP should receive/transmit in that time slot.

If the McASP should participate (transmit/receive bit is active) in the time slot, the McASP functions normally. If the McASP should not participate (transmit/receive bit is inactive) in the time slot, no transfers between the XRBUF and XRSR registers in the serializer would occur during that time slot. In addition, the serializers programmed as transmitters place their data output pins in a predetermined state (logic low, high, or high impedance) as programmed by each serializer control register (SRCTL). Refer also to Section 3.2.2 for details on how DMA event or interrupt generations are handled during inactive time slots in TDM mode.

The receive TDM sequencer is controlled by register RTDM and reports current receive slot to RSLOT. The transmit TDM sequencer is controlled by register XTDM and reports current transmit slot to XSLOT.

2.7 Clock Check Circuit

A common source of error in audio systems is a serial clock failure due to instabilities in the off-chip DIR circuit. To detect a clock error quickly, a clock-check circuit is included in the McASP for both transmit and receive clocks, since both may be sourced from off chip.

The clock check circuit can detect and recover from transmit and receive clock failures. See Section 3.6.6 for implementation and programming details.

2.8 Pin Function Control

All McASP pins except AMUTEIN are bidirectional input/output pins. In addition, these bidirectional pins function either as McASP or general-purpose I/O (GPIO) pins. The following registers control the pin functions:

- Pin function register (PFUNC): selects pin to function as McASP or GPIO
- Pin direction register (PDIR): selects pin to be input or output
- Pin data input register (PDIN): shows data input at the pin
- Pin data output register (PDOUT): data to be output at the pin if the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1). Not applicable when the pin is configured as McASP pin (PFUNC[n] = 0).
- Pin data set register (PDSET): alias of PDOUT. Writing a 1 to PDSET[n] sets the respective PDOUT[n] to 1. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).
- Pin data clear register (PDCLR): alias of PDOUT. Writing a 1 to PDCLR[n] clears the respective PDOUT[n] to 0. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).

See the register descriptions in Section 4.1 for details on the mapping of each McASP pin to the register bits. Figure 2-8 shows the pin control block diagram.

2.8.1 McASP Pin Control-Transmit and Receive

You must correctly set the McASP GPIO registers PFUNC and PDIR, even when McASP pins are used for their serial port (non-GPIO) function.

Serial port functions include:

- Clock pins (ACLKX, ACLKR, AHCLKX, AHCLKR, AFSX, AFSR) used as clock inputs and outputs
- Serializer data pins (AXR[n]) used to transmit or receive
- AMUTE used as a mute output signal

When using these pins in their serial port function, you must clear PFUNC[n] to 0 for each pin, as opposed to PFUNC[n] = 1, which makes the pin a GPIO.

Also, certain outputs require PDIR[n] = 1, such as clock pins used as clock outputs, serializer data pins used to transmit, and AMUTE used as mute output.

Clock inputs and serializers configured to receive must have PDIR[n] = 0.

PFUNC and PDIR do not control the AMUTEIN signal, it is usually tied to a device level interrupt pin (consult device datasheet). If used as a mute input, this pin needs to be configured as an input in the appropriate peripheral (GPIO).

Finally, there is an important advantage to having separate control of pin direction (by PDIR), and the choice of internal versus external clocking (by CLKRM/CLKXM). Depending on the specific device and usage, you might select an external clock (CLKRM = 0), while enabling the internal clock divider, and the clock pin as an output in the PDIR register (PDIR[ACLKR] = 1). In this case, the bit clock is an output (PDIR[ACLKR] = 1) and, therefore, routed to the ACLKR pin. However, because CLKRM = 0, the bit clock is then routed back to the McASP module as an "external" clock source. This may result in less skew between the clock inside the McASP and the clock in the external device, thus producing more balanced setup and hold times for a particular system. As a result, this may allow a higher serial clock rate interface.

2.8.2 GPIO Pin Control

For GPIO operation, you must set the desired PFUNC[n] to 1 to indicate GPIO function. PDIR[n] must be configured to the desired direction. PDOUT, PDSET, PDCLR control the output value on the pin. PDIN always reflects the state at the pin, regardless of the PDIR and PFUNC setting.

Figure 2-8 and Figure 2-9 display the pin descriptions. The examples that follow (Example 2-1 through Example 2-4) show how the pins can be used as general-purpose input or output pins.

Figure 2-8. McASP I/O Pin Control Block Diagram

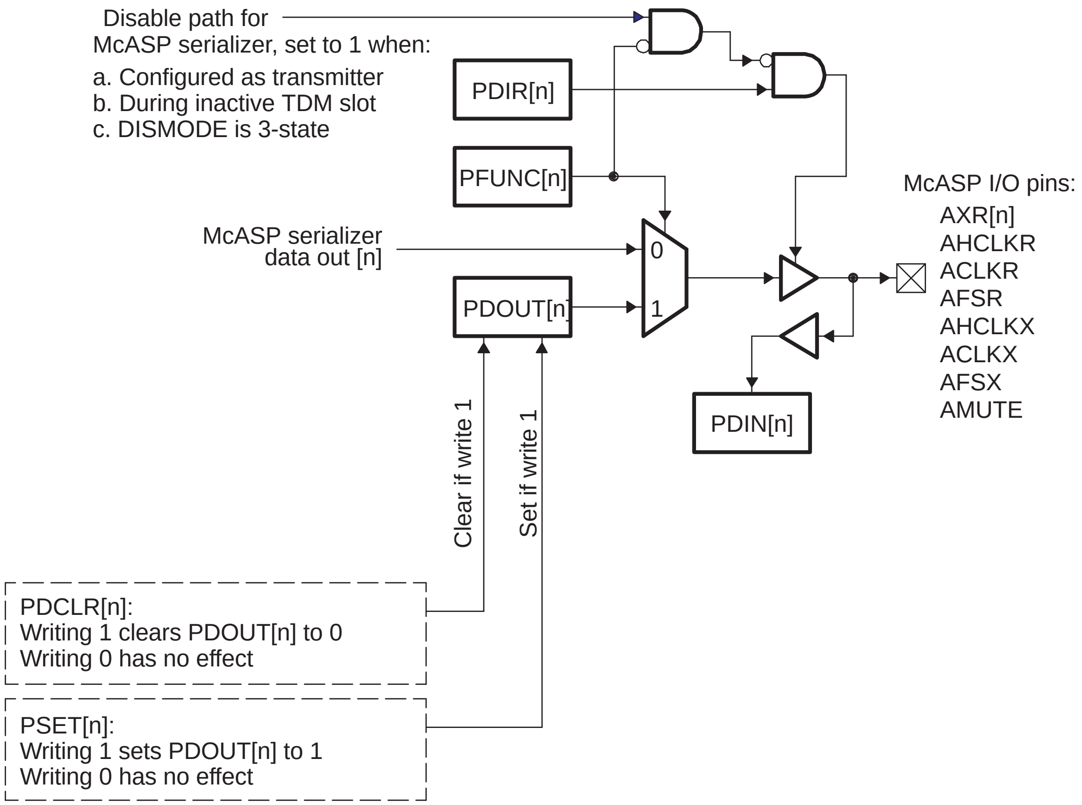


Figure 2-9. McASP I/O Pin to Control Register Mapping

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23							16
Reserved							
R-0							

15	14	13	12	11	10	9	8
AXR15[A]	AXR14[A]	AXR13[A]	AXR12[A]	AXR11[A]	AXR10[A]	AXR9[A]	AXR8[A]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A On DA6x DSP only; reserved on C6713 DSP.

Example 2-1. General-Purpose Input Pin

Because the PDIN register always reflects the state at the pin, you can read the PDIN register to obtain the pin input state. To explicitly set the pin as a general-purpose input pin, you can set the registers as follows:

- PDIR[n] = 0 (input)
- PFUNC[n] = 1 (GPIO function)

Example 2-2. General-Purpose Output Pin—Initialization Using PDOUT

All pins default as inputs. To initialize a pin as output, you should follow this sequence:

1. PDIR[n] = 0 (default as input)
2. PFUNC[n] = 1 (GPIO function)
3. PDOUT[n] = desired output value
4. PDIR[n] = 1 (change to output after desired value is configured in PDOUT[n])

Example 2-3. General-Purpose Output Pin—Change Data from 0 to 1 Using PDSET

If the pin is already configured as a general-purpose output pin driving a 0, and you want to change the output from 0 to 1, the recommended method is to use the PDSET register instead of the PDOUT register. This is because writing to the PDSET register only affects pin(s) in concern. To change a pin from 0 to 1:

- Set PDSET[n]. This sets the respective PDOUT[n].

Example 2-4. General-Purpose Output Pin—Change Data from 1 to 0 Using PDCLR

If the pin is already configured as a general-purpose output pin driving a 1, and you want to change the output from 1 to 0, the recommended method is to use the PDCLR register instead of the PDOUT register. This is because writing to the PDCLR register only affects pin(s) in concern. To change a pin from 1 to 0:

- Set PDCLR[n]. This clears the respective PDOUT[n].

Operation

This chapter discusses the operation of the McASP.

3.1 Setup and Initialization

This section discusses steps necessary to use the McASP module.

3.1.1 Transmit/Receive Section Initialization

You must follow the following steps to properly configure the McASP. If external clocks are used, they should be present prior to the following initialization steps.

1. Reset McASP to default values by setting GBLCTL = 0.
2. Configure all McASP registers except GBLCTL in the following order:
   a. Power down and emulation management: PWRDEMU.
   b. Receive registers: RMASK, RFMT, AFSRCTL, ACLKRCTL, AHCLKRCTL, RTDM, RINTCTL, RCLKCHK. If external clocks AHCLKR and/or ACLKR are used, they must be running already for proper synchronization of the GBLCTL register.
   c. Transmit registers: XMASK, XFMT, AFSXCTL, ACLKXCTL, AHCLKXCTL, XTDM, XINTCTL, XCLKCHK. If external clocks AHCLKX and/or ACLKX are used, they must be running already for proper synchronization of the GBLCTL register.
   d. Serializer registers: SRCTL[n].
   e. Global registers: Registers PFUNC, PDIR, DITCTL, DLBCTL, AMUTE. Note that PDIR should only be programmed after the clocks and frames are set up in the steps above. This is because the moment a clock pin is configured as an output in PDIR, the clock pin starts toggling at the rate defined in the corresponding clock control register. Therefore you must ensure that the clock control register is configured appropriately before you set the pin to be an output. A similar argument applies to the frame sync pins. Also note that the reset state for the transmit high-frequency clock divide register (HCLKXDIV) is divide-by-1, and the divide-by-1 clocks are not gated by the transmit high-frequency clock divider reset enable (XHCLKRST).
   f. DIT registers: For DIT mode operation, set up registers DITCSRA[n], DITCSRB[n], DITUDRA[n], and DITUDRB[n].
3. Start the respective high-frequency serial clocks AHCLKX and/or AHCLKR. This step is necessary even if external high-frequency serial clocks are used:
   a. Take the respective internal high-frequency serial clock divider(s) out of reset by setting the RHCLKRST bit for the receiver and/or the XHCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be held at 0.
   b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
4. Start the respective serial clocks ACLKX and/or ACLKR. This step can be skipped if external serial clocks are used and they are running:
   a. Take the respective internal serial clock divider(s) out of reset by setting the RCLKRST bit for the receiver and/or the XCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
5. Setup data acquisition as required:
   a. If DMA is used to service the McASP, set up data acquisition as desired and start the DMA in this step, before the McASP is taken out of reset.
   b. If CPU interrupt is used to service the McASP, enable the transmit and/ or receive interrupt as required.
   c. If CPU polling is used to service the McASP, no action is required in this step.

6. Activate serializers.
    a. Before starting, clear the respective transmitter and receiver status registers by writing XSTAT = FFFFh and RSTAT = FFFFh.
    b. Take the respective serializers out of reset by setting the RSRCLR bit for the receiver and/or the XSRCLR bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
    c. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
7. Verify that all transmit buffers are serviced. Skip this step if the transmitter is not used. Also, skip this step if time slot 0 is selected as inactive (special cases, see Figure 3-2, second waveform). As soon as the transmit serializer is taken out of reset, XDATA in the XSTAT register is set, indicating that XBUF is empty and ready to be serviced. The XDATA status causes an DMA event AXEVT to be generated, and can cause an interrupt AXINT to be generated if it is enabled in the XINTCTL register.
    a. If DMA is used to service the McASP, the DMA automatically services the McASP upon receiving AXEVT. Before proceeding in this step, you should verify that the XDATA bit in the XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the DMA.
    b. If CPU interrupt is used to service the McASP, interrupt service routine is entered upon the AXINT interrupt. The interrupt service routine should service the XBUF registers. Before proceeding in this step, you should verify that the XDATA bit in XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the CPU.
    c. If CPU polling is used to service the McASP, the XBUF registers should be written to in this step.
8. Release state machines from reset.
    a. Take the respective state machine(s) out of reset by setting the RSMRST bit for the receiver and/or the XSMRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
    b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
9. Release frame sync generators from reset. Note that it is necessary to release the internal frame sync generators from reset, even if an external frame sync is being used, because the frame sync error detection logic is built into the frame sync generator.
    a. Take the respective frame sync generator(s) out of reset by setting the RFRST bit for the receiver, and/or the XFRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
    b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
10. Upon the first frame sync signal, McASP transfers begin. The McASP synchronizes to an edge on the frame sync pin, not the level on the frame sync pin. This makes it easy to release the state machine and frame sync generators from reset.
    a. For example, if you configure the McASP for a rising edge transmit frame sync, then you do not need to wait for a low level on the frame sync pin before releasing the McASP transmitter state machine and frame sync generators from reset.

3.1.2 Separate Transmit and Receive Initialization

In many cases, it is desirable to separately initialize the McASP transmitter and receiver. For example, you may delay the initialization of the transmitter until the type of data coming in on the receiver is recognized. Or a change in the incoming data stream on the receiver may necessitate a reinitialization of the transmitter.

In this case, you may still follow the sequence outlined in Section 3.1.1, but use it for each section (transmit, receive) individually. The GBLCTL register is aliased to RGBLCTL and XGBLCTL to facilitate separate initialization of transmit and receive sections.

Also, make sure that the initialization or reinitialization sequence follows the guidelines in Table B-1.

3.1.3 Importance of Reading Back GBLCTL

In Section 3.1.1, steps 3b, 4b, 6c, 8b, and 9b state that GBLCTL should be read back until the bits that were written are successfully latched. This is important, because the transmitter and receiver state machines run off of the respective bit clocks, which are typically about tens to hundreds of times slower than the DSP's internal bus clock. Therefore, it takes many cycles between when the DSP writes to GBLCTL (or RGBLCTL and XGBLCTL), and when the McASP actually recognizes the write operation. If you skip this step, then the McASP may never see the reset bits in the global control registers get asserted and deasserted; resulting in an uninitialized McASP.

Therefore, the logic in McASP has been implemented such that once the DSP writes GBLCTL, RGBLCTL, or XGBLCTL, the resulting write is not visible by reading back GBLCTL until the McASP has recognized the change. This typically requires two bit clocks plus two DSP bus clocks to occur.

Also, if the bit clocks can be completely stopped, any software that polls GBLCTL should be implemented with a time-out. If GBLCTL does not have a time-out, and the bit clock stops, the changes written to GBLCTL will not be reflected until the bit clock restarts.

Finally, please note that while RGBLCTL and XGBLCTL allow separate changing of the receive and transmit halves of GBLCTL, they also immediately reflect the updated value (useful for debug purposes). Only GBLCTL can be used for the read back step.

3.1.4 Synchronous Transmit and Receive Operation (ASYNC = 0)

When ASYNC = 0 in ACLKXCTL, the transmit and receive sections operate synchronously from the transmit section clock and transmit frame sync signals (Figure 2-2). The receive section may have a different (but compatible in terms of slot size) data format.

When ASYNC = 0, the receive frame sync generator is internally disabled. If the AFSX pin is configured as an output, it serves as the frame sync signal for both transmit and receive. The AFSR pin should not be used because the transmit frame sync generator output, which is used by both the transmitter and the receiver when ASYNC = 0, is not propogated to the AFSR pin (Figure 2-4).

When ASYNC = 0, the transmit and receive sections must share some common settings, since they both use the same clock and frame sync signals:

- DITEN = 0 in DITCTL (TDM mode is enabled)
- The total number of bits per frame must be the same (that is, RSSZ ɤnbsp;RMOD must equal XSSZ ɤnbsp;XMOD)
- Both transmit and receive should either be specified as burst or TDM mode, but not mixed
- The settings in ACLKRCTL are irrelevant
- FSXM must match FSRM
- FXWID must match FRWID

For all other settings, the transmit and receive sections may be programmed independently.

3.1.5 Asynchronous Transmit and Receive Operation (ASYNC = 1)

When ASYNC = 1 in ACLKXCTL, the transmit and receive sections operate completely independently and have separate clock and frame sync signals (Figure 2-2, Figure 2-3, and Figure 2-4). The events generated by each section come asynchronously.

3.2 Transfer Modes

3.2.1 Burst Transfer Mode

The McASP supports a burst transfer mode, which is useful for nonaudio data such as passing control information between two DSPs. Burst transfer mode uses a synchronous serial format similar to the TDM mode. The frame sync generation is not periodic or time-driven as in TDM mode, but data driven, and the frame sync is generated for each data word transferred.

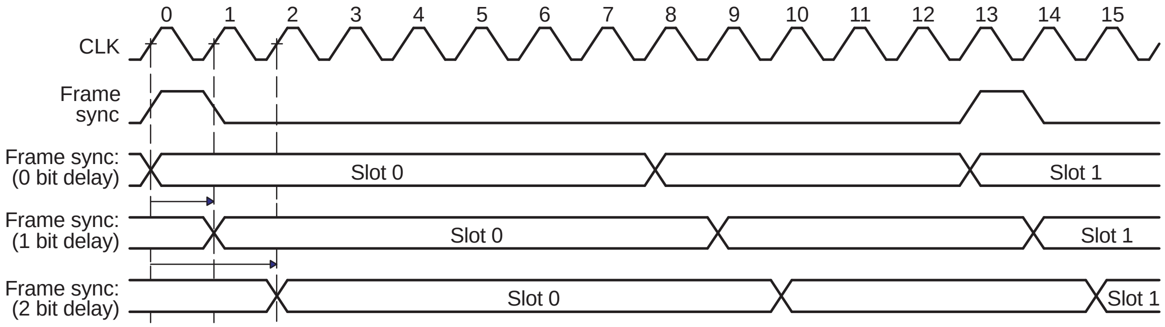
When operating in burst frame sync mode (Figure 3-1), as specified for transmit (XMOD = 0 in AFSXCTL) and receive (RMOD = 0 in AFSRCTL), one slot is shifted for each active edge of the frame sync signal that is recognized. Additional clocks after the slot and before the next frame sync edge are ignored.

In burst frame sync mode, the frame sync delay may be specified as 0, 1, or 2 serial clock cycles. This is the delay between the frame sync active edge and the start of the slot. The frame sync signal lasts for a single bit clock duration (FRWID = 0 in AFSRCTL, FXWID = 0 in AFSXCTL).

For transmit, when generating the transmit frame sync internally, the frame sync begins when the previous transmission has completed and when all the XBUF[n] (for every serializer set to operate as a transmitter) has been updated with new data.

For receive, when generating the receive frame sync internally, frame sync begins when the previous transmission has completed and when all the RBUF[n] (for every serializer set to operate as a receiver) has been read.

Figure 3-1. Burst Frame Sync Mode


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
CLK
Frame sync
Frame sync: (0 bit delay)
Slot 0
Slot 1
Frame sync: (1 bit delay)
Slot 0
Slot 1
Frame sync: (2 bit delay)
Slot 0
Slot 1

The control registers must be configured as follows for the burst transfer mode. The burst mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 3.1.1 to configure this register.
- AMUTE: Not applicable. Leave at default.
- DLBCTL: If loopback mode is desired, configure this register according to Section 3.7, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select non-DIT mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 2.4 and Section 3.4.
- RFMT/XFMT: Program all fields according to data format desired. See Section 3.4.
- AFSRCTL/AFSXCTL: Clear **RMOD/XMOD** bits to 0 to indicate burst mode. Clear **FRWID/FXWID** bits to 0 for single bit frame sync duration. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 2.2.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 2.2.
- RTDM/XTDM: Program RTDMS0/XTDMS0 to 1 to indicate one active slot only. Leave other fields at default.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Not applicable. Leave at default.
- SRCTLn: Program SRMOD to inactive/transmitter/receiver as desired. DISMOD is not applicable and should be left at default.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

3.2.2 Time-Division Multiplexed (TDM) Transfer Mode

The McASP time-division multiplexed (TDM) transfer mode supports the TDM format discussed in Section 1.6.

Transmitting data in the TDM transfer mode requires a minimum set of pins:

- ACLKX - transmit bit clock
- AFSX - transmit frame sync (or commonly called left/right clock)
- One or more serial data pins, AXR[n], whose serializers have been configured to transmit

The transmitter has the option to receive the ACLKX bit clock as an input, or to generate the ACLKX bit clock by dividing down the AHCLKX high-frequency master clock. The transmitter can either generate AHCLKX internally or receive AHCLKX as an input. See Section 2.2.1.

Similarly, to receive data in the TDM transfer mode requires a minimum set of pins:

- ACLKR - receive bit clock
- AFSR - receive frame sync (or commonly called left/right clock)
- One or more serial data pins, AXR[n], whose serializers have been configured to receive

The receiver has the option to receive the ACLKR bit clock as an input or to generate the ACLKR bit clock by dividing down the AHCLKR high-frequency master clock. The receiver can either generate AHCLKR internally or receive AHCLKR as an input. See Section 2.2.2 and Section 2.2.3.

The control registers must be configured as follows for the TDM mode. The TDM mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 3.1.1 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: If loopback mode is desired, configure this register according to Section 3.7, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select TDM mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 2.4 and Section 3.4.
- RFMT/XFMT: Program all fields according to data format desired. See Section 3.4.
- AFSRCTL/AFSXCTL: Set **RMOD/XMOD** bits to 2-32 for TDM mode. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 2.2.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 2.2.
- RTDM/XTDM: Program all fields according to the time slot characteristics desired.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Program all fields according to serializer operation desired.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

3.2.2.1 TDM Time Slots

TDM mode on the McASP can extend to support multiprocessor applications, with up to 32 time slots per frame. For each of the time slots, the McASP may be configured to participate or to be inactive by configuring XTDM and/or RTDM (this allows multiple DSPs to communicate on the same TDM serial bus).

The TDM sequencer (separate ones for transmit and receive) functions in this mode. The TDM sequencer counts the slots beginning with the frame sync. For each slot, the TDM sequencer checks the respective bit in either XTDM or RTDM to determine if the McASP should transmit/receive in that time slot.

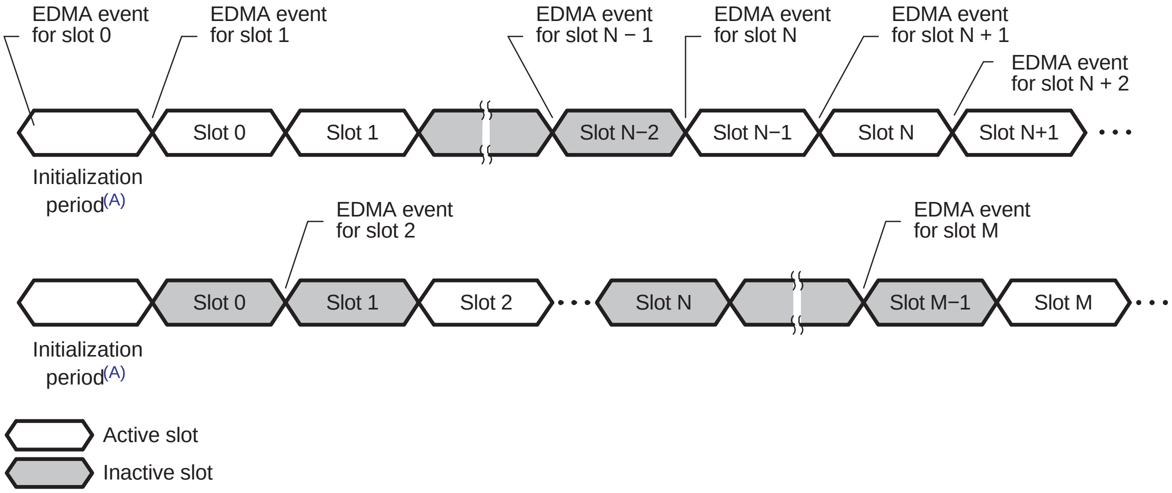
If the transmit/receive bit is active, the McASP functions normally during that time slot; otherwise, the McASP is inactive during that time slot; no update to the buffer occurs, and no event is generated. Transmit pins are automatically set to a high-impedance state, 0, or 1 during that slot, as determined by bit DISMOD in SRCTL[n].

Figure 3-2 shows when the transmit DMA event AXEVT is generated. See Section 3.3.1 for details on data ready and the initialization period indication. The transmit DMA event for an active time slot (slot N) is generated during the previous time slot (slot N - 1), regardless if the previous time slot (slot N - 1) is active or inactive.

During an active transmit time slot (slot N), if the next time slot (slot N + 1) is configured to be active, the copy from XRBUF[n] to XRSR[n] generates the DMA event for time slot N + 1. If the next time slot (slot N + 1) is configured to be inactive, then the DMA event will be delayed to time slot M - 1. In this case, slot M is the next active time slot. The DMA event for time slot M is generated during the first bit time of slot M - 1.

The receive DMA request generation does not need this capability, since the receive DMA event is generated after data is received in the buffer (looks back in time). If a time slot is disabled, then no data is copied to the buffer for that time slot and no DMA event is generated.

Figure 3-2. Transmit DMA Event (AXEVT) Generation in TDM Time Slots



A See Section 3.1.1, step 7a.

3.2.2.2 Special 384 Slot TDM Mode for Connection to External DIR

The McASP receiver also supports a 384 time slot TDM mode (DIR mode), to support S/PDIF, AES-3, IEC-60958 receiver ICs whose natural block (block corresponds to McASP frame) size is 384 samples. The advantage to using the 384 time slot TDM mode is that interrupts may be generated synchronous to the S/PDIF, AES-3, IEC-60958, such as the last slot interrupt.

The receive TDM time slot register (RTDM) should be programmed to all 1s during reception of a DIR block. Other TDM functionalities (for example, inactive slots) are not supported (only the slot counter counts the 384 subframes in a block).

To receive data in the DIR mode, the following pins are typically needed:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock). In this mode, AFSR should be connected to a DIR which outputs a start of block signal, instead of LRCLK.
- One or more serial data pins, AXR[n], whose serializers have been configured to receive.

For this special DIR mode, the control registers can be configured just as for TDM mode, except set RMOD in AFSRCTL to 384 to receive 384 time slots.

3.2.3 Digital Audio Interface Transmit (DIT) Transfer Mode

In addition to the TDM and burst transfer modes, which are suitable for transmitting audio data between ICs inside the same system, the digital audio interface transmit (DIT) transfer mode of the McASP also supports transmission of audio data in the S/PDIF, AES-3, or IEC-60958 format. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode only applies to serializers configured as transmitters, not receivers. Refer to Section 1.7 for a description of the S/PDIF format.

3.2.3.1 Transmit DIT Encoding

The McASP operation in DIT mode is basically identical to the 2 time slot TDM mode, but the data transmitted is output as a biphase mark encoded bit stream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bit stream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

The transmit TDM time slot register (XTDM) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in the DIT mode, the following pins are typically needed:

- AHCLKX - transmit high-frequency master clock
- One or more serial data pins, AXR[n], whose serializers have been configured to transmit

AHCLKX is optional (the internal clock source may be used instead), but if used as a reference, the DSP provides a clock check circuit that continually monitors the AHCLKX input for stability.

If the McASP is configured to transmit in the DIT mode on more than one serial data pin, the bit streams on all pins will be synchronized. In addition, although they will carry unique audio data, they will carry the same channel status, user data, and validity information.

The actual 24-bit audio data must always be in bit positions 23-0 after passing through the first three stages of the transmit format unit.

For left-aligned Q31 data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 010 (rotate right by 8 bits)
- XRVRS = 0 (no bit reversal, LSB first)
- XMASK = FFFF FF00h-FFFF 0000h (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present)
- XPAD = 00 (pad extra bits with 0)

For right-aligned data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 000 (rotate right by 0 bits)
- XRVRS = 0 (no bit reversal, LSB first)
- XMASK = 00FF FFFFh to 0000 FFFFh (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present)
- XPAD = 00 (pad extra bits with 0)

3.2.3.2 Transmit DIT Clock and Frame Sync Generation

The DIT transmitter only works in the following configuration:

- In transmit frame control register (AFSXCTL):
  - Internally-generated transmit frame sync, FSXM = 1
  - Rising-edge frame sync, FSXP = 0
  - Bit-width frame sync, FXWID = 0
  - 384-slot TDM, XMOD = 1 1000 0000b
- In transmit clock control register (ACLKXCTL), ASYNC = 1
- In transmit bitstream format register (XFMT), XSSZ = 1111 (32-bit slot size)

All combinations of AHCLKX and ACLKX are supported.

This is a summary of the register configurations required for DIT mode. The DIT mode specific bit fields are in bold face:

- PFUNC: The data pins must be configured for McASP function. If AHCLKX is used, it must also be configured for McASP function. Other pins can be configured to function as GPIO if desired.
- PDIR: The data pins must be configured as outputs. If AHCLKX is used as an input reference, it should be configured as input. If internal clock source AUXCLK is used as the reference clock, it may be output on the AHCLKX pin by configuring AHCLKX as an output.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable for DIT operation. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 3.1.1 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: Not applicable. Loopback is not supported for DIT mode. Leave at default.
- DITCTL: **DITEN** bit must be set to 1 to enable DIT mode. Configure other bits as desired.
- RMASK: Not applicable. Leave at default.
- RFMT: Not applicable. Leave at default.
- AFSRCTL: Not applicable. Leave at default.
- ACLKRCTL: Not applicable. Leave at default.
- AHCLKRCTL: Not applicable. Leave at default.
- RTDM: Not applicable. Leave at default.
- RINTCTL: Not applicable. Leave at default.
- RCLKCHK: Not applicable. Leave at default.
- **XMASK**: Mask desired bits according to the discussion in this section, depending upon left-aligned or right-aligned internal data.
- **XFMT:XDATDLY** = 0. **XRVRS** = 0. **XPAD** = 0. **XPBIT** = default (not applicable). **XSSZ** = Fh (32-bit slot). XBUSEL = configured as desired. **XROT** bit is configured according to the discussion in this section, either 0 or 8-bit rotate.
- **AFSXCTL**: Configure the bits according to the discussion in this section.
- **ACLKXCTL**: **ASYNC** = 1. Program CLKXDIV bits to obtain the bit clock rate desired. Configure CLKXP and CLKXM bits as desired, because CLKX is not actually used in the DIT protocol.
- **AHCLKXCTL**: Program all fields according to high-frequency clock desired.
- **XTDM**: Set to FFFF FFFFh for all active slots for DIT transfers.
- XINTCTL: Program all fields according to interrupts desired.
- XCLKCHK: Program all fields according to clock checking desired.

- SRCTLn: Set **SRMOD** = 1 (transmitter) for the DIT pins. DISMOD field is don't care for DIT mode.
- **DITCSRA[n], DITCSRB[n]**: Program the channel status bits as desired.
- **DITUDRA[n], DITUDRB[n]**: Program the user data bits as desired.

3.2.3.3 DIT Channel Status and User Data Register Files

The channel status registers (DITCSRA*n* and DITCSRB*n*) and user data registers (DITUDRA*n* and DITUDRB*n*) are not double buffered. Typically the programmer uses one of the synchronizing interrupts, such as last slot, to create an event at a safe time so the register may be updated. In addition, the CPU reads the transmit TDM slot counter to determine which word of the register is being used.

It is a requirement that the software avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it will be indeterminate whether the old or new data is used to encode the bitstream.

The DIT subframe format is defined in Section 1.7.2. The channel status information (C) and User Data (U) are defined in these DIT control registers:

- DITCSRA0 to DITCSRA5: The 192 bits in these six registers contain the channel status information for the LEFT channel within each frame.
- DITCSRB0 to DITCSRB5: The 192 bits in these six registers contain the channel status information for the RIGHT channel within each frame.
- DITUDRA0 to DITUDRA5: The 192 bits in these six registers contain the user data information for the LEFT channel within each frame.
- DITUDRB0 to DITUDRB5: The 192 bits in these six registers contain the user data information for the RIGHT channel within each frame.

The S/PDIF block format is shown in Figure 1-14. There are 192 frames within a block (frame 0 to frame 191). Within each frame there are two subframes (subframe 1 and 2 for left and right channels, respectively). The channel status and user data information sent on each subframe is summarized in Table 3-1.

Table 3-1. Channel Status and User Data for Each DIT Block

Frame	Subframe	Preamble	Channel Status defined in:	User Data defined in:
Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0				
0	1 (L)	B	DITCSRA0[0]	DITUDRA0[0]
0	2 (R)	W	DITCSRB0[0]	DITUDRB0[0]
1	1 (L)	M	DITCSRA0[1]	DITUDRA0[1]
1	2 (R)	W	DITCSRB0[1]	DITUDRB0[1]
2	1 (L)	M	DITCSRA0[2]	DITUDRA0[2]
2	2 (R)	W	DITCSRB0[2]	DITUDRB0[2]
…	…	…	…	…
31	1 (L)	M	DITCSRA0[31]	DITUDRA0[31]
31	2 (R)	W	DITCSRB0[31]	DITUDRB0[31]
Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1				
32	1 (L)	M	DITCSRA1[0]	DITUDRA1[0]
32	2 (R)	W	DITCSRB1[0]	DITUDRB1[0]
…	…	…	…	…
63	1 (L)	M	DITCSRA1[31]	DITUDRA1[31]
63	2 (R)	W	DITCSRB1[31]	DITUDRB1[31]
Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2				
64	1 (L)	M	DITCSRA2[0]	DITUDRA2[0]
64	2 (R)	W	DITCSRB2[0]	DITUDRB2[0]
…	…	…	…	…
95	1 (L)	M	DITCSRA2[31]	DITUDRA2[31]
95	2 (R)	W	DITCSRB2[31]	DITUDRB2[31]
Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3				
96	1 (L)	M	DITCSRA3[0]	DITUDRA3[0]
96	2 (R)	W	DITCSRB3[0]	DITUDRB3[0]
…	…	…	…	…
127	1 (L)	M	DITCSRA3[31]	DITUDRA3[31]
127	2 (R)	W	DITCSRB3[31]	DITUDRB3[31]
Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4				
128	1 (L)	M	DITCSRA4[0]	DITUDRA4[0]
128	2 (R)	W	DITCSRB4[0]	DITUDRB4[0]
…	…	…	…	…
159	1 (L)	M	DITCSRA4[31]	DITUDRA4[31]
159	2 (R)	W	DITCSRB4[31]	DITUDRB4[31]
Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5				
160	1 (L)	M	DITCSRA5[0]	DITUDRA5[0]
160	2 (R)	W	DITCSRB5[0]	DITUDRB5[0]
…	…	…	…	…
191	1 (L)	M	DITCSRA5[31]	DITUDRA5[31]
191	2 (R)	W	DITCSRB5[31]	DITUDRB5[31]

3.3 Data Transmission and Reception

The DSP services the McASP by writing data to the XBUF register(s) for transmit operations, and by reading data from the RBUF register(s) for receive operations. The McASP sets status flag and notifies the DSP whenever data is ready to be serviced. Section 3.3.1 discusses data ready status in detail.

The XBUF and RBUF registers can be accessed through one of the two peripheral ports of the device:

- The data port (DAT): This port is dedicated for data transfers on the device.
- The configuration bus (CFG): This port is used for both data transfers and peripheral configuration control on the device.

Section 3.3.2 and Section 3.3.3 discuss how to perform transfers through the data port and the configuration bus.

Either the CPU or the DMA can be used to service the McASP through any of these two peripheral ports. The CPU and DMA usages are discussed in Section 3.3.4 and Section 3.3.5.

3.3.1 Data Ready Status and Event/Interrupt Generation

3.3.1.1 Transmit Data Ready

The transmit data ready flag XDATA bit in the XSTAT register reflects the status of the XBUF register. The XDATA flag is set when data is transferred from the XRBUF[n] buffers to the XRSR[n] shift registers, indicating that the XBUF is empty and ready to accept new data from the DSP. This flag is cleared when the XDATA bit is written with a 1, or when all the serializers configured as transmitters are written by the DSP.

Whenever XDATA is set, an DMA event AXEVT is automatically generated to notify the DMA of the XBUF empty status. An interrupt AXINT is also generated if XDATA interrupt is enabled in the XINTCTL register (See Section 3.5.1 for details).

For DMA requests, the McASP does not require XSTAT to be read between DMA events. This means that even if XSTAT already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

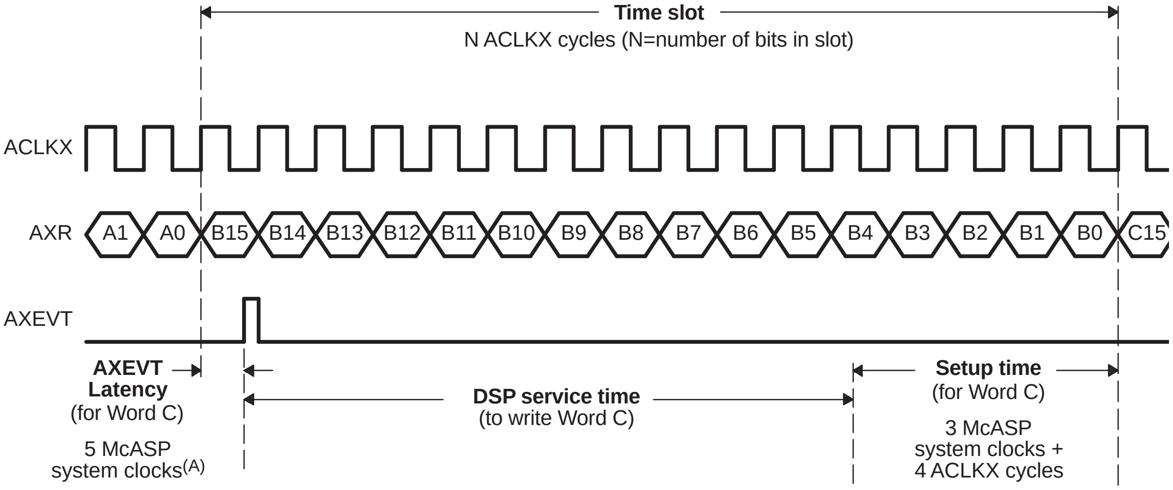
Since all serializers act in lockstep, only one DMA event is generated to indicate that all active transmit serializers are ready to be written to with new data.

Figure 3-3 shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to 5 McASP system clocks (AXEVT Latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the DSP can begin servicing the McASP by writing Word C into the XBUF (DSP Service Time). The DSP must write Word C into the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum DSP Service Time (Figure 3-3) can be calculated as:

DSP Service Time = Time Slot - AXEVT Latency - Setup Time

Figure 3-3. DSP Service Time Upon Transmit DMA Event (AXEVT)



A Refer to the device-specific data manual for the McASP system clock source. This is not the same as AUXCLK. For example, the C6713 DSP uses SYSCLK2 as the McASP system clock source.

Example 3-1. DSP Service Time Calculation for Transmit DMA Event (AXEVT)

The following is an example to show how to calculate DSP Service Time. Assume the following setup:

- Device: C6713 DSP at 300 MHZ
- McASP transmits in I2S format at 192 kHz frame rate. Assume slot size is 32 bit

With the above setup, we obtain the following parameters corresponding to Figure 3-3:

- Calculation of McASP system clock cycle:
  - C6713 DSP uses SYSCLK2 as the McASP system clock. It runs at 150 MHZ (half of device frequency)
  - Therefore, McASP system clock cycle = 1/150 MHZ = 6.7 ns
- Calculation of ACLKX clock cycle:
  - This example has two 32-bit slots per frame, for a total of 64 bits per frame
  - ACLKX clock cycle is (1/192 kHz)/64 = 81.4 ns
- Time Slot between AXEVT events:
  - For I2S format, McASP generates two AXEVT events per 192 kHz frame
  - Therefore, Time Slot between AXEVT events is (1/192 kHz)/2 = 2604 ns
- AXEVT Latency

  = 5 McASP system clocks

  = 6.7 ns × 5 = 33.5 ns
- Setup Time

  = 3 McASP system clocks + 4 ACLKX cycles

  = (6.7 ns × 3) + (81.4 ns × 4)

  = 345.7 ns
- DSP Service Time

  = Time Slot - AXEVT Latency - Setup Time

  = 2604 ns - 33.5 ns - 345.7 ns

  = 2225 ns

3.3.1.2 Receive Data Ready

Similarly, the receive data ready flag RDATA bit in the RSTAT reflects the status of the RBUF register. The RDATA flag is set when data is transferred from the XRSR[n] shift registers to the XRBUF[n] buffers, indicating that the RBUF contains received data and is ready to have the DSP read the data. This flag is cleared when the RDATA bit is written with a 1, or when all the serializers configured as receivers are read.

Whenever RDATA is set, an DMA event AREVT is automatically generated to notify the DMA of the RBUF ready status. An interrupt ARINT is also generated if RDATA interrupt is enabled in the RINTCTL register (See Section 3.5.2 for details).

For DMA requests, the McASP does not require RSTAT to be read between DMA events. This means that even if RSTAT already has the RDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Since all serializers act in lockstep, only one DMA event is generated to indicate that all active receive serializers are ready to receive new data.

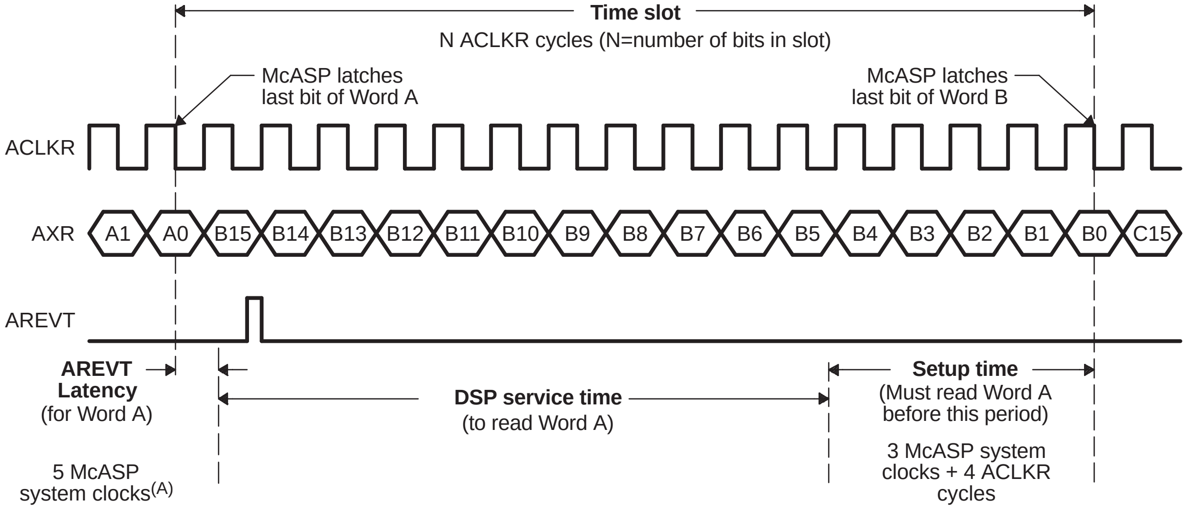
Figure 3-4 shows the timing details of when AREVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is received, the McASP sets the RDATA flag and generates an AREVT event. However, it takes up to 5 McASP system clocks (AREVT Latency) before AREVT is active at the McASP boundary. Upon AREVT, the DSP can begin servicing the McASP by reading Word A from the RBUF (DSP Service Time). The DSP must read Word A from the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum DSP Service Time (Figure 3-4) can be calculated as:

DSP Service Time = Time Slot - AREVT Latency - Setup Time

The DSP Service Time calculation for receive is similar to the calculation for transmit. See Example 3-1 for DSP Service Time calculation using transmit as an example.

Figure 3-4. DSP Service Time Upon Receive DMA Event (AREVT)


Time slot
N ACLKR cycles (N=number of bits in slot)
McASP latches last bit of Word A
McASP latches last bit of Word B
ACLKR
AXR
A1
A0
B15
B14
B13
B12
B11
B10
B9
B8
B7
B6
B5
B4
B3
B2
B1
B0
C15
AREVT
AREVT Latency (for Word A)
5 McASP system clocks(A)
DSP service time (to read Word A)
Setup time (Must read Word A before this period)
3 McASP system clocks + 4 ACLKR cycles

A Refer to the device-specific data manual for the McASP system clock source. This is not the same as AUXCLK. For example, the C6713 DSP uses SYSCLK2 as the McASP system clock source.

3.3.2 Transfers through the Data Port (DAT)

CAUTION

To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

Typically, you will access the McASP XRBUF registers through the data port. To access through the data port, simply have the CPU or DMA access the XRBUF through its data port location. Refer to the device-specific data manual for the exact memory address. Through the data port, the DMA/CPU can service all the serializers through a single address. The McASP automatically cycles through the appropriate serializers.

For transmit operations through the data port, the DMA/CPU should write to the same XBUF data port address to service all of the active transmit serializers. In addition, the DMA/CPU should write to the XBUF for all active transmit serializers in incremental (although not necessarily consecutive) order. For example, if serializers 0, 4, 5, and 7 are set up as active transmitters, the DMA/CPU should write to the XBUF data port address four times with data for serializers 0, 4, 5, and 7 upon each transmit data ready event. This exact servicing order must be followed so that data appears in the appropriate serializers.

Similarly, for receive operations through the data port, the DMA/CPU should read from the same RBUF data port address to service all of the active receive serializers. In addition, reads from the active receive serializers through the data port return data in incremental (although not necessarily consecutive) order. For example, if serializers 1, 2, 3, and 6 are set up as active receivers, the DMA/CPU should read from the RBUF data port address four times to obtain data for serializers 1, 2, 3, and 6 in this exact order, upon each receive data ready event.

When transmitting, the DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 3.6.2). Similarly, when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 3.6.3).

To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers.

3.3.3 Transfers through the Configuration Bus (CFG)

CAUTION

To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

In this method, the DMA/CPU accesses the XRBUF registers through the configuration bus address. The exact XRBUF register address for any particular serializer is determined by adding the offset for that particular serializer to the base address for the particular McASP (found in the device-specific data manual). XRBUF for the serializers configured as transmitters is given the name XBUF*n*. For example, the XRBUF associated with transmit serializer 2 is named XBUF2. Similarly, XRBUF for the serializers configured as receivers is given the name RBUF*n*.

Accessing the XRBUF registers through the data port is different because the CPU/DMA only needs to access one single address. When accessing through the configuration bus, the CPU/DMA must provide the exact XBUF*n* or RBUF*n* address for each access.

When transmitting, DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 3.6.2). Similarly when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 3.6.3).

To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers.

3.3.4 Using the CPU for McASP Servicing

The CPU can be used to service the McASP through interrupt (upon AXINT/ARINT interrupts) or through polling the XDATA bit in the XSTAT register. As discussed in Section 3.3.2 and Section 3.3.3, the CPU can access through either the data port or through the configuration bus.

To use the CPU to service the McASP through interrupts, the XSTAT/RSTAT bit must be enabled in the respective XINTCTL/RINTCTL registers, to generate interrupts AXINT/ARINT to the CPU upon data ready.

3.3.5 Using the DMA for McASP Servicing

The most typical scenario is to use the DMA to service the McASP through the data port, although the DMA can also service the McASP through the configuration bus. Two possibilities exist for using the DMA events to service the McASP:

1. Use **AXEVT/AREVT**: Triggered upon each XDATA/RDATA transition from 0 to 1.
2. Use **AXEVTO/AREVTO** and **AXEVTE/AREVTE**: Alternating AXEVT/AREVT events for odd/even slots. Upon AXEVT/AREVT, AXEVTO/AREVTO is triggered if the event is for an odd channel, and AXEVTE/AREVTE is triggered if the event is for an even channel.

Note: Check the device-specific data manual to see if AXEVTO/AREVTO and AXEVTE/AREVTE are supported. These are optional.

Figure 3-5 and Figure 3-6 show an example audio system with six audio channels (LF, RF, LS, RS, C, and LFE) transmitted from three AXR[n] pins on the McASP. Figure 3-5 and Figure 3-6 show when events AXEVT, AXEVTO, and AXEVTE are triggered. Figure 3-5 and Figure 3-6 also apply for the receive audio channels and show when events AREVT, AREVTO, and AREVTE are triggered.

Figure 3-5. DMA Events in an Audio Example–Two Events (Scenario 1)

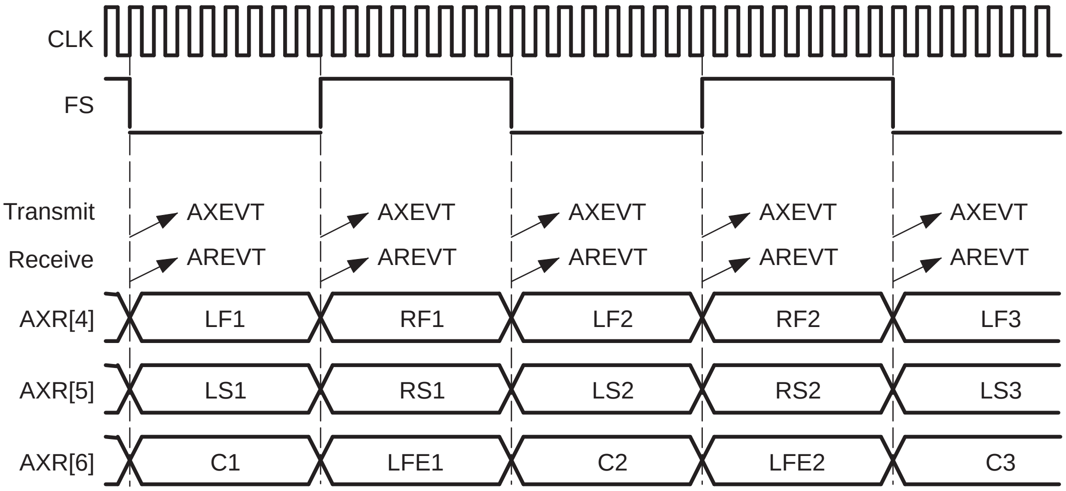
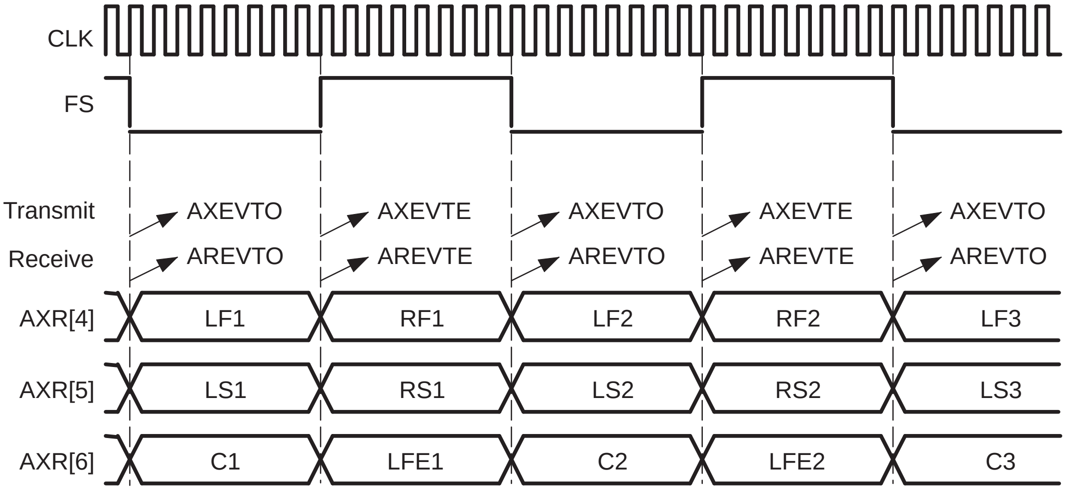


Figure 3-6. DMA Events in an Audio Example–Four Events (Scenario 2)



You can either use the DMA to service the McASP upon events AXEVT and AREVT (Figure 3-5) or upon events AXEVTO, AREVTO, AXEVTE, and AREVTE (Figure 3-6).

In scenario 1 (Figure 3-5), a DMA event AXEVT/AREVT is triggered on each time slot. In the example, AXEVT is triggered for each of the transmit audio channel time slot (Time slot for channels LF, LS, and C; and time slot for channels RF, RS, LFE). Similarly, AREVT is triggered for each of the receive audio channel time slot. Scenario 1 allows for the use of a single DMA to transmit all audio channels, and a single DMA to receive all audio channels.

In scenario 2 (Figure 3-6), two alternating DMA events are triggered for each time slot. In the example, AXEVTE (even) is triggered for the time slot for the even audio channels (LF, LS, C) and AXEVTO (odd) is triggered for the time slot for the odd audio channels (RF, RS, LFE). AXEVTO and AXEVTE alternate in time. The same is true in the receive direction with the use of AREVTO and AREVTE. This scenario allows for the use of two DMA channels (odd and even) to transmit all audio channels, and two DMA channels to receive all audio channels.

Appendix B shows example EDMA implementations of scenario 1 and scenario 2.

Here are some guidelines on using the different DMA events:

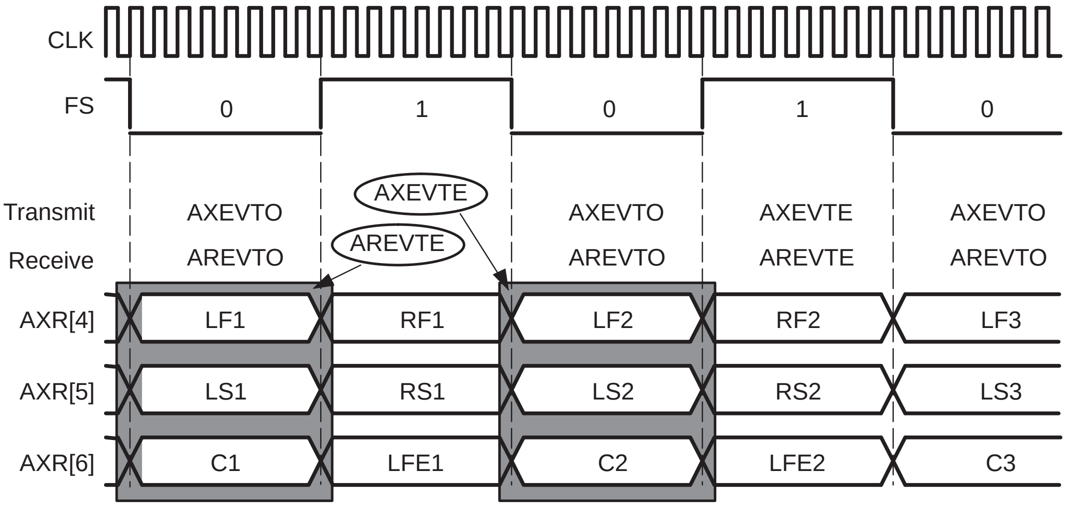
- Either use AXEVT, or the combination of AXEVTO and AXEVTE, to service the McASP. Never use all three at the same time. Similarly for receive, either use AREVT, or the combination of AREVTO and AREVTE.
- The McASP generates transmit DMA events independently from receive DMA events; therefore, separate schemes can be used for transmit and receive DMA. For example, scenario 1 could be used for the transmit data (AXEVT) and scenario 2 could be used for the receive data (AREVTO, AREVTE), and conversely.

Note the difference between DMA event generation and the CPU interrupt generation. DMA events are generated automatically upon data ready; whereas CPU interrupt generation needs to be enabled in the XINTCTL/RINTCTL register.

In Figure 3-6, scenario 2, each transmit DMA request is for data in the next time slot, while each receive DMA request is for data in the previous time slot. For example, Figure 3-7 shows a circled AXEVTE event for an even time slot transmit DMA request. The transmitter always requests a DMA transfer for data it will need to transmit during the next time slot. So, in this example, the circled event AXEVTE is a request for data for samples LF2, LS2, and C2.

On the other hand, the circled AREVTE event is an even time slot receive DMA request. The receiver always requests a DMA transfer for data it received during the previous time slot. In this example, the circled event AREVTE is a request for samples LF1, LS1, and C1.

Figure 3-7. DMA Events in an Audio Example



3.4 Formatter

3.4.1 Transmit Bit Stream Data Alignment

The McASP transmitter supports serial formats of:

- Slot (or Time slot) size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment: when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order: order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the transmit bitstream format register (XFMT):

- XRVRS: bit reverse (1) or no bit reverse (0)
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- XSSZ: transmit slot size of 8, 12, 16, 20, 24, 28, or 32 bits

XSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the XROT field.

Table 3-2 and Figure 3-8 show the XRVRS and XROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size (SLOT in Table 3-2) and word size (WORD in Table 3-2) options are multiples of 4, since the transmit rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The transmit bit mask/pad unit operates on data as an initial step of the transmit format unit (see Figure 2-7), and the data is aligned in the same representation as it is written to the transmitter by the DSP (typically Q31 or integer).

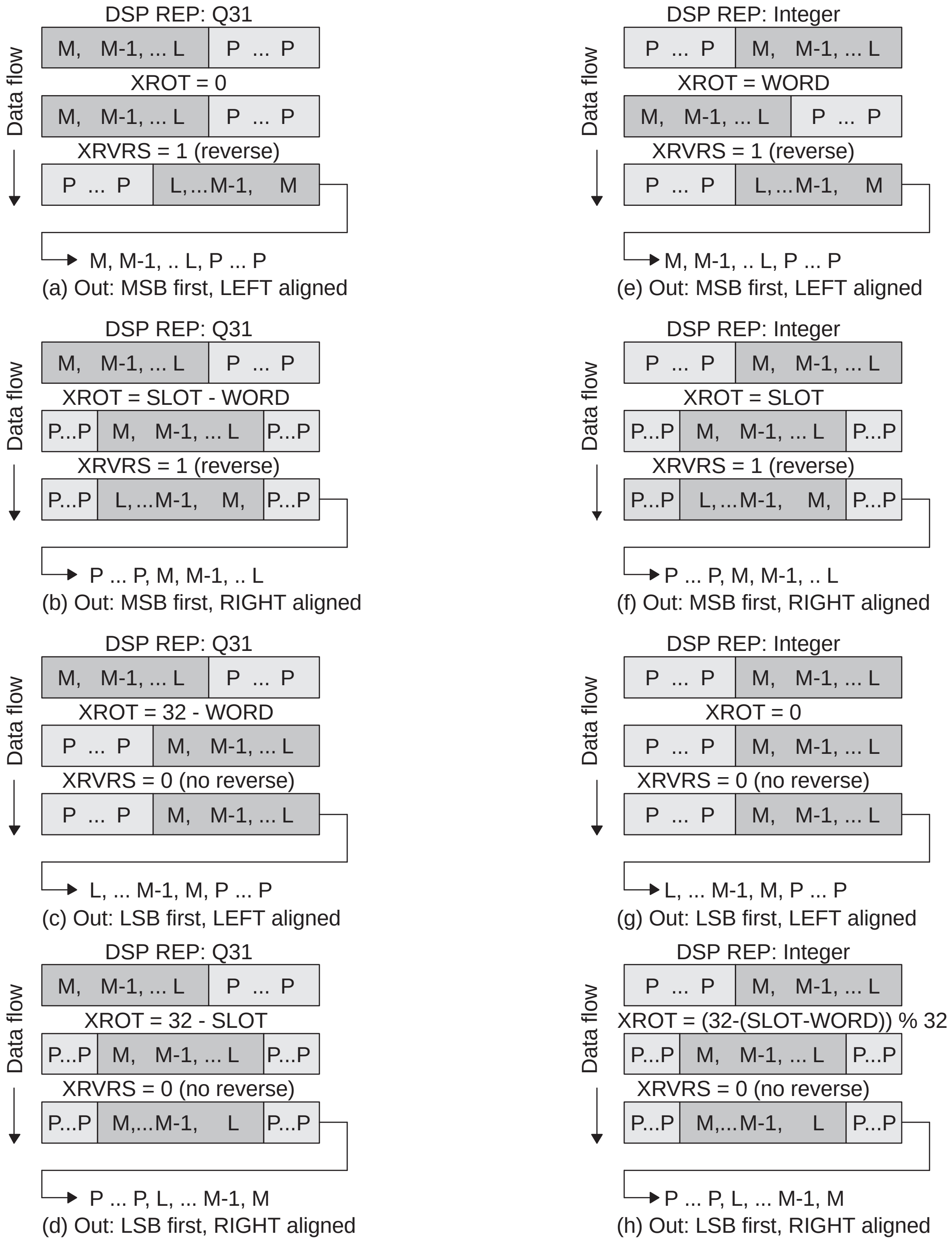
Table 3-2. Transmit Bitstream Data Alignment

				XFMT Bit	
Figure 3-8	Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	XROT[1]	XRVRS
(a)[2]	MSB first	Left aligned	Q31 fraction	0	1
(b)	MSB first	Right aligned	Q31 fraction	SLOT - WORD	1
(c)	LSB first	Left aligned	Q31 fraction	32 - WORD	0
(d)	LSB first	Right aligned	Q31 fraction	32 - SLOT	0
(e)[2]	MSB first	Left aligned	Integer	WORD	1
(f)	MSB first	Right aligned	Integer	SLOT	1
(g)	LSB first	Left aligned	Integer	0	0
(h)	LSB first	Right aligned	Integer	(32 - (SLOT - WORD)) % 32	0

[1] WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator

[2] To transmit in I2S format, use MSB first, left aligned, and also select XDATDLY = 01 (1 bit delay)

Figure 3-8. Data Flow Through Transmit Format Unit, Illustrated



(a) Out: MSB first, LEFT aligned

(b) Out: MSB first, RIGHT aligned

(c) Out: LSB first, LEFT aligned

(d) Out: LSB first, RIGHT aligned

(e) Out: MSB first, LEFT aligned

(f) Out: MSB first, RIGHT aligned

(g) Out: LSB first, LEFT aligned

(h) Out: LSB first, RIGHT aligned

3.4.2 Receive Bit Stream Data Alignment

The McASP receiver supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the receive bitstream format register (RFMT):

- RRVRS: bit reverse (1) or no bit reverse (0)
- RROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- RSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits

RSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the RROT field.

Table 3-3 and Figure 3-9 show the RRVRS and RROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size and word size options are multiples of 4; since the receive rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The receive bit mask/pad unit operates on data as the final step of the receive format unit (see Figure 2-6), and the data is aligned in the same representation as it is read from the receiver by the DSP (typically Q31 or integer).

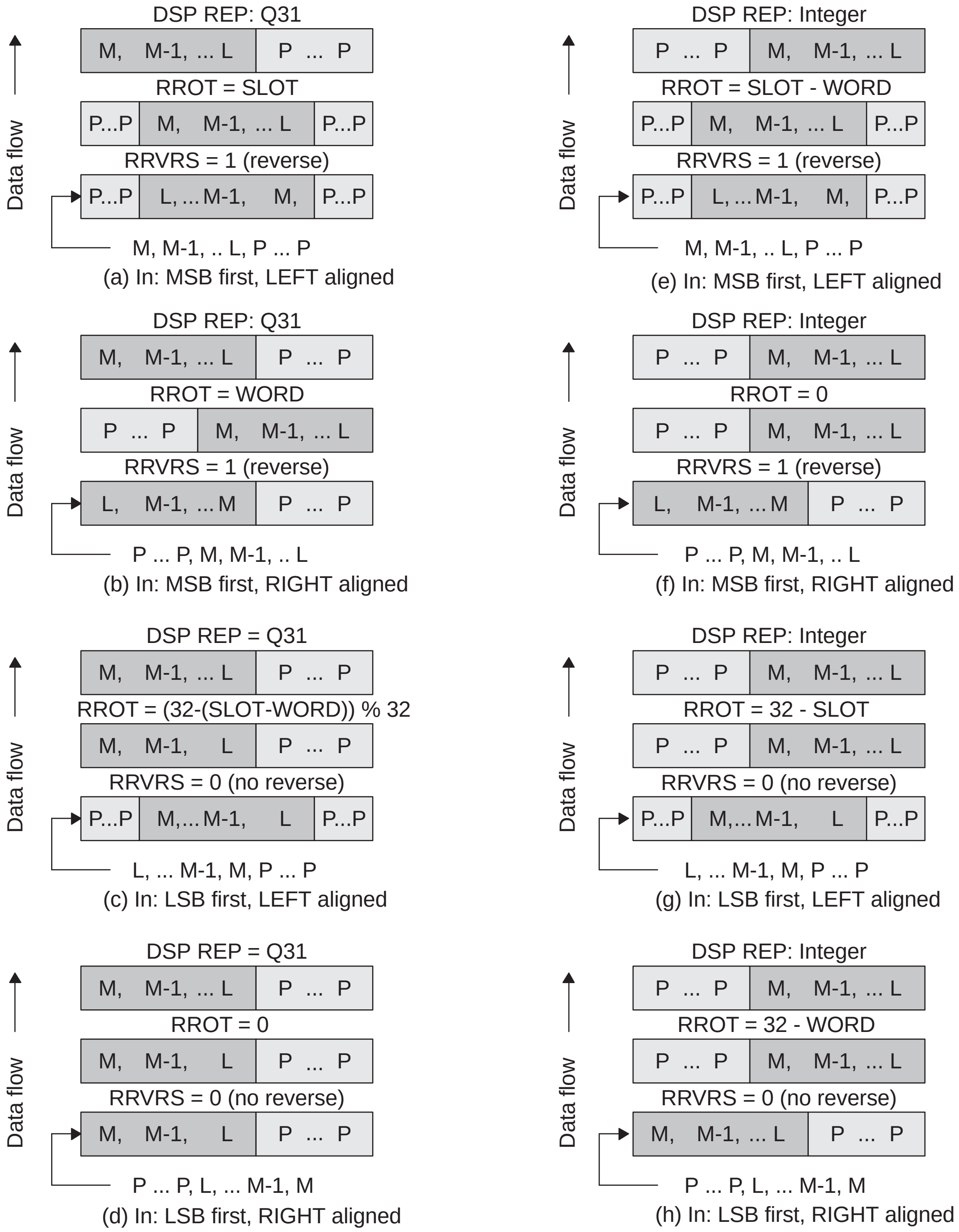
Table 3-3. Receive Bitstream Data Alignment

				RFMT Bit	
Figure 3-9	Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	RROT[1]	RRVRS
(a)[2]	MSB first	Left aligned	Q31 fraction	SLOT	1
(b)	MSB first	Right aligned	Q31 fraction	WORD	1
(c)	LSB first	Left aligned	Q31 fraction	(32 - (SLOT - WORD)) % 32	0
(d)	LSB first	Right aligned	Q31 fraction	0	0
(e)[2]	MSB first	Left aligned	Integer	SLOT - WORD	1
(f)	MSB first	Right aligned	Integer	0	1
(g)	LSB first	Left aligned	Integer	32 - SLOT	0
(h)	LSB first	Right aligned	Integer	32 - WORD	0

[1] WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator

[2] To transmit in I2S format, select MSB first, left aligned, and also select RDATDLY = 01 (1 bit delay)

Figure 3-9. Data Flow Through Receive Format Unit, Illustrated



(a) In: MSB first, LEFT aligned

(e) In: MSB first, LEFT aligned

(b) In: MSB first, RIGHT aligned

(f) In: MSB first, RIGHT aligned

(c) In: LSB first, LEFT aligned

(g) In: LSB first, LEFT aligned

(d) In: LSB first, RIGHT aligned

(h) In: LSB first, RIGHT aligned

3.5 Interrupts

3.5.1 Transmit Data Ready Interrupt

The transmit data ready interrupt (XDATA) is generated if XDATA is 1 in the XSTAT register and XDATA is also enabled in XINTCTL. Section 3.3.1 provides details on when XDATA is set in the XSTAT register.

A transmit start of frame interrupt (XSTAFRM) is triggered by the recognition of transmit frame sync. A transmit last slot interrupt (XLAST) is a qualified version of the data ready interrupt (XDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended was next-to-last TDM slot, current slot is last slot).

3.5.2 Receive Data Ready Interrupt

The receive data ready interrupt (RDATA) is generated if RDATA is 1 in the RSTAT register and RDATA is also enabled in RINTCTL. Section 3.3.2 provides details on when RDATA is set in the RSTAT register.

A receiver start of frame interrupt (RSTAFRM) is triggered by the recognition of a receiver frame sync. A receiver last slot interrupt (RLAST) is a qualified version of the data ready interrupt (RDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data in the buffer come from the last TDM time slot (the slot that just ended was last TDM slot).

3.5.3 Error Interrupts

Upon detection, the following error conditions generate interrupt flags:

- In the receive status register (RSTAT):
  - Receiver overrun (ROVRN)
  - Unexpected receive frame sync (RSYNCERR)
  - Receive clock failure (RCKFAIL)
  - Receive DMA error (RDMAERR)
- In the transmit status register (XSTAT):
  - Transmit underrun (XUNDRN)
  - Unexpected transmit frame sync (XSYNCERR)
  - Transmit clock failure (XCKFAIL)
  - Transmit DMA error (XDMAERR)

Each interrupt source also has a corresponding enable bit in the receive interrupt control register (RINTCTL) and transmit interrupt control register (XINTCTL). If the enable bit is set in RINTCTL or XINTCTL, an interrupt is requested when the interrupt flag is set in RSTAT or XSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

3.5.4 Audio Mute (AMUTE) Function

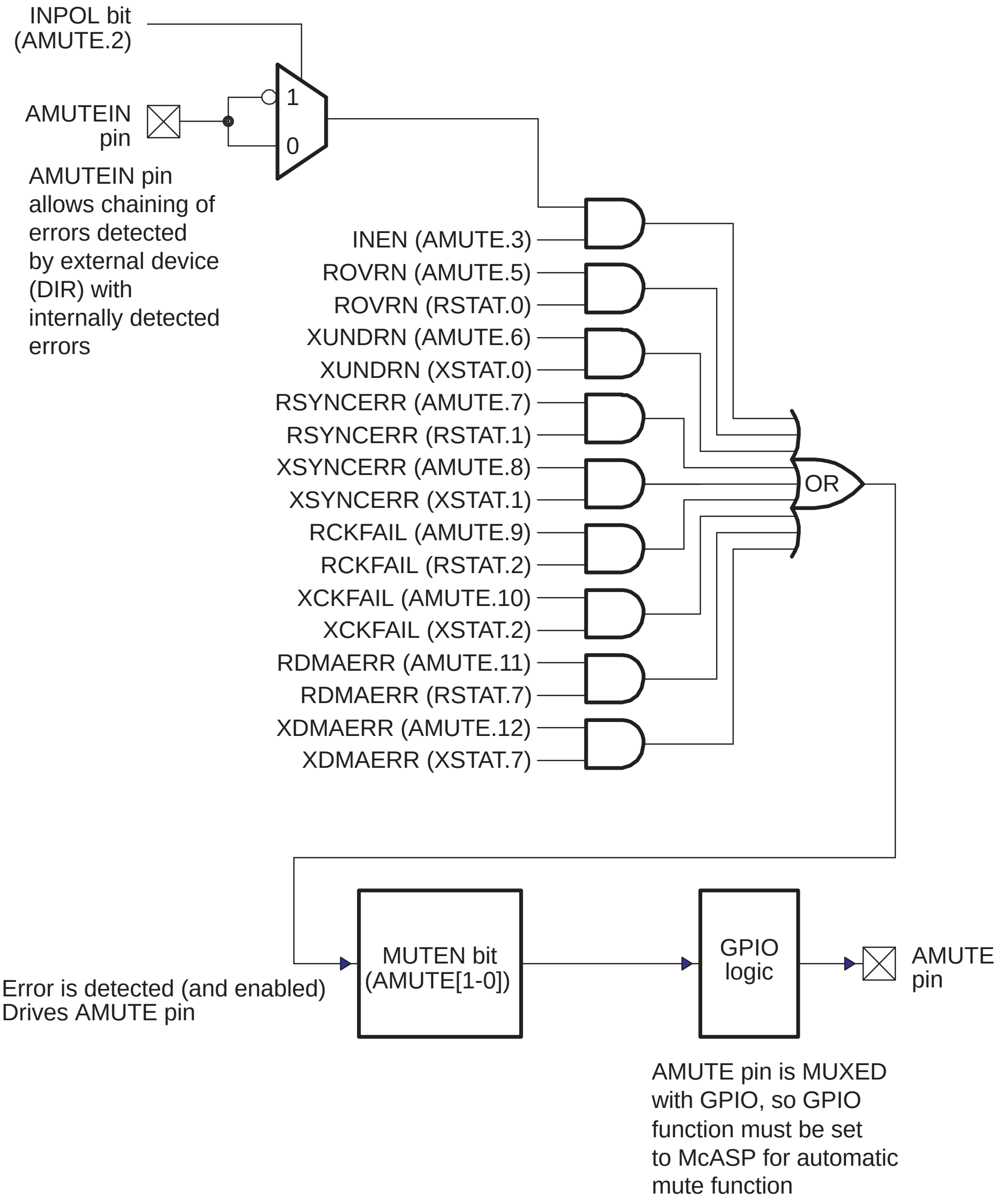
The McASP includes an automatic audio mute function (Figure 3-10) that asserts in hardware the AMUTE pin to a preprogrammed output state, as selected by the MUTEN bit in the audio mute control register (AMUTE). The AMUTE pin is asserted when one of the interrupt flags is set or an external device issues an error signal on the AMUTEIN input. Typically, the AMUTEIN input is shared with a device interrupt pin (for example EXT_INT4).

The AMUTEIN input allows the on-chip logic to consider a mute input from other devices in the system, so that all errors may be considered. The AMUTEIN input has a programmable polarity to allow it to adapt to different devices, as selected by the INPOL bit in AMUTE, and it must be enabled explicitly.

In addition to the external AMUTEIN input, the AMUTE pin output may be asserted when one of the error interrupt flags is set and its mute function is enabled in AMUTE.

When one or more of the errors is detected and enabled, the AMUTE pin is driven to an active state that is selected by MUTEN in AMUTE. The active polarity of the AMUTE pin is programmable by MUTEN (and the inactive polarity is the opposite of the active polarity). The AMUTE pin remains driven active until software clears all the error interrupt flags that are enabled to mute, and until the AMUTEIN is inactive.

Figure 3-10. Audio Mute (AMUTE) Block Diagram



3.5.5 Multiple Interrupts

This only applies to interrupts and not to DMA requests. The following terms are defined:

- **Active Interrupt Request:** a flag in RSTAT or XSTAT is set and the interrupt is enabled in RINTCTL or XINTCTL.
- **Outstanding Interrupt Request:** An interrupt request has been issued on one of the McASP transmit/receive interrupt ports, but that request has not yet been serviced.
- **Serviced:** The CPU writes to RSTAT or XSTAT to clear one or more of the active interrupt request flags.

The first interrupt request to become active for the transmitter with the interrupt flag set in XSTAT and the interrupt enabled in XINTCTL generates a request on the McASP transmit interrupt port AXINT.

If more than one interrupt request becomes active in the same cycle, a single interrupt request is generated on the McASP transmit interrupt port. Subsequent interrupt requests that become active while the first interrupt request is outstanding do not immediately generate a new request pulse on the McASP transmit interrupt port.

The transmit interrupt is serviced with the CPU writing to XSTAT. If any interrupt requests are active after the write, a new request is generated on the McASP transmit interrupt port.

The receiver operates in a similar way, but using RSTAT, RINTCTL, and the McASP receive interrupt port ARINT.

One outstanding interrupt request is allowed on each port, so a transmit and a receive interrupt request may both be outstanding at the same time.

3.6 Error Handling and Management

To support the design of a robust audio system, the McASP includes error-checking capability for the serial protocol, data underrun, and data overrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX/AHCLKR clock cycles. The timer value can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

Upon the detection of any one or more errors (software selectable), or the assertion of the AMUTEIN input pin, the AMUTE output pin may be asserted to a high or low level to immediately mute the audio output. In addition, an interrupt may be generated if desired, based on any one or more of the error sources.

3.6.1 Unexpected Frame Sync Error

An unexpected frame sync occurs when:

- In burst mode, when the next active edge of the frame sync occurs early such that the current slot will not be completed by the time the next slot is scheduled to begin.
- In TDM mode, a further constraint is that the frame sync must occur exactly during the correct bit clock (not a cycle earlier or later) and only before slot 0. An unexpected frame sync occurs if this condition is not met.

When an unexpected frame sync occurs, there are two possible actions depending upon when the unexpected frame sync occurs:

1. Early: An early unexpected frame sync occurs when the McASP is in the process of completing the current frame and a new frame sync is detected (not including overlap that occurs due to a 1 or 2 bit frame sync delay). When an early unexpected frame sync occurs:
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Current frame is not resynchronized. The number of bits in the current frame is completed. The next frame sync, which occurs after the current frame is completed, will be resynchronized.

2. Late: A late unexpected frame sync occurs when there is a gap or delay between the last bit of the previous frame and the first bit of the next frame. When a late unexpected frame sync occurs (as soon as the gap is detected):
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Resynchronization occurs upon the arrival of the next frame sync.

Late frame sync is detected the same way in both burst mode and TDM mode; however, in burst mode, late frame sync is not meaningful and its interrupt enable should not be set.

3.6.2 Buffer Underrun Error - Transmitter

A buffer underrun can only occur for serializers programmed to be transmitters. A buffer underrun occurs when the serializer is instructed by the transmit state machine to transfer data from XRBUF[n] to XRSR[n], but XRBUF[n] has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the DSP explicitly writes a 1 to the XUNDRN bit to clear the XUNDRN bit.

In DIT mode, a pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 ɿfs). By shifting out a pair of zeros, a clock may be recovered on the receiver. To recover, reset the McASP and start again with the proper initialization.

In TDM mode, during an underrun case, a long stream of zeros are shifted out causing the DACs to mute. To recover, reset the McASP and start again with the proper initialization.

3.6.3 Buffer Overrun Error - Receiver

A buffer overrun can only occur for serializers programmed to be receivers. A buffer overrun occurs when the serializer is instructed to transfer data from XRSR[n] to XRBUF[n], but XRBUF[n] has not yet been read by either the DMA or the DSP. When this occurs, the receiver state machine sets the ROVRN flag. However, the individual serializer writes over the data in the XRBUF[n] register (destroying the previous sample) and continues shifting.

An overrun is checked only once per time slot. The ROVRN flag is set when an overrun condition occurs. It is possible that an overrun occurs on one time slot but then the DSP catches up and does not cause an overrun on the following time slots. However, once the ROVRN flag is set, it remains set until the DSP explicitly writes a 1 to the ROVRN bit to clear the ROVRN bit.

3.6.4 DMA Error - Transmitter

A transmit DMA error, as indicated by the XDMAERR flag in the XSTAT register, occurs when the DMA (or CPU) writes more words to the DAT port of the McASP than it should. For each DMA event, the DMA should write exactly as many words as there are serializers enabled as transmitters.

XDMAERR indicates that the DMA (or CPU) wrote too many words to the McASP for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in XSTAT.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP transmitter and the DMA to resynchronize them.

3.6.5 DMA Error - Receiver

A receive DMA error, as indicated by the RDMAERR flag in the RSTAT register, occurs when the DMA (or CPU) reads more words from the DAT port of the McASP than it should. For each DMA event, the DMA should read exactly as many words as there are serializers enabled as receivers.

RDMAERR indicates that the DMA (or CPU) read too many words from the McASP for a given receive DMA event. Reading too few words results in a receiver overrun error setting ROVRN in RSTAT.

While RDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP receiver and the DMA to resynchronize them.

3.6.6 Clock Failure Detection

3.6.6.1 Clock-Failure Check Startup

It is expected, initially, that the clock-failure circuits will generate an error until at least one measurement has been taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not immediately be enabled, but be enabled only after a specific startup procedure. The startup procedure is:

1. For the transmit clock failure check:
   a. Configure transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (XCLKCHK).
   b. Clear transmit clock failure flag (XCKFAIL) in the transmit status register (XSTAT).
   c. Wait until first measurement is taken (> 32 AHCLKX clock periods).
   d. Verify no clock failure is detected.
   e. Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   f. After the transmit clock is measured and falls within the acceptable range, the following may be enabled:
      i. transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (XINTCTL)
      ii. transmit clock failure detect autoswitch enable bit (XCKFAILSW) in the transmit clock check control register (XCLKCHK)
      iii. mute option (XCKFAIL) in the mute control register (AMUTE)
2. For the receive clock failure check:
   a. Configure receive clock failure detect logic (RMIN, RMAX, RPS) in the receive clock check control register (RCLKCHK).
   b. Clear receive clock failure flag (RCKFAIL) in the receive status register (RSTAT).
   c. Wait until first measurement is taken (> 32 AHCLKR clock periods).
   d. Verify no clock failure is detected.
   e. Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   f. After the receive clock is measured and falls within the acceptable range, the following may be enabled:
      i. receive clock failure interrupt enable bit (RCKFAIL) in the receiver interrupt control register (RINTCTL)
      ii. mute option (RCKFAIL) in the mute control register (AMUTE)

3.6.6.2 Transmit Clock Failure Check and Recovery

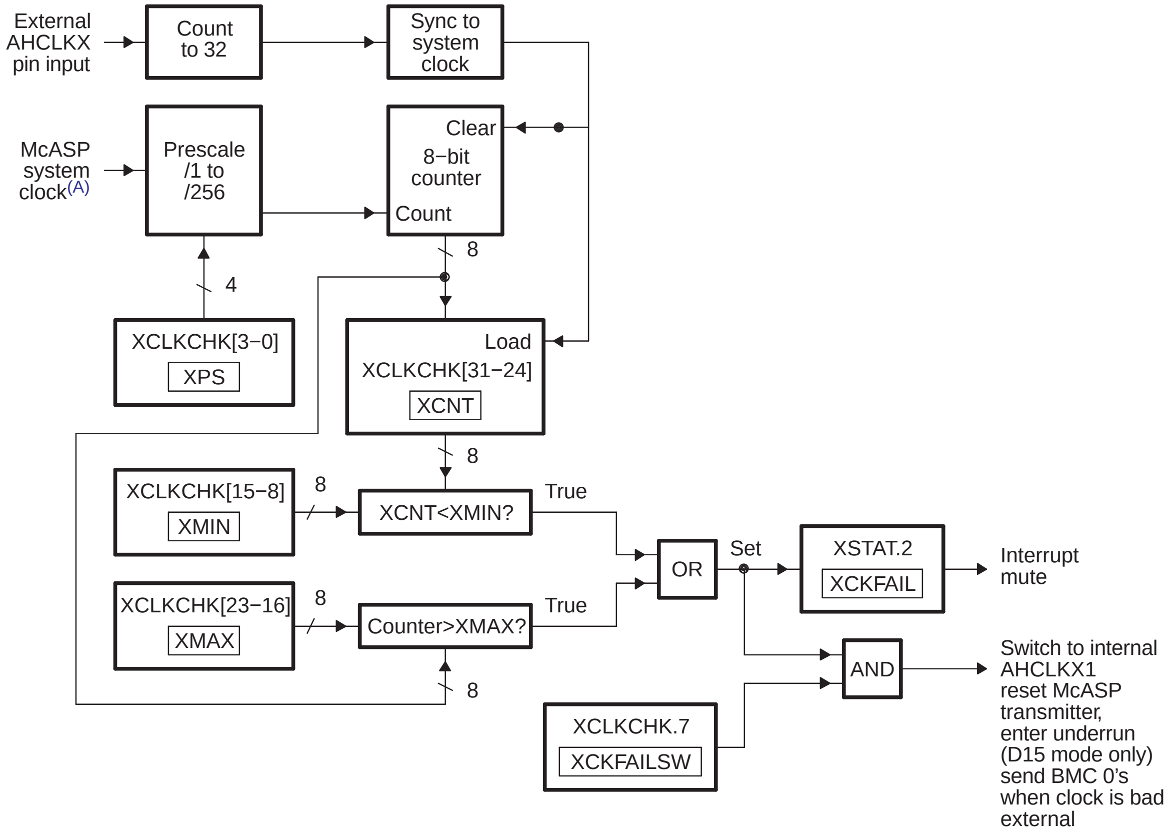
The transmit clock failure check circuit (Figure 3-11) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKX). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKX) periods, and stores the count in XCNT of the transmit clock check control register (XCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in XSTST) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than XMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (XMAX). This is in case the external clock completely stops, so that the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. Note that the XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected, or that the audio source has changed and a new sample rate is being used.

In order for the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKX is internally generated or externally sourced.

Figure 3-11. Transmit Clock Failure Detection Circuit Block Diagram



A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.

The following actions are taken if a clock failure is detected:

1. Transmit clock failure flag (XCKFAIL) in XSTAT is set. This causes an interrupt if transmit clock failure interrupt enable bit (XCKFAIL) in XINTCTL is set.

In addition (only supported for DIT mode), if the transmit clock failure detect autoswitch enable bit (XCKFAILSW) in XCLKCHK is set, the following additional steps are taken to change the clock source from external to internal:

1. High-frequency transmit clock source bit (HCLKXM) in AHCLKXCTL is set to 1 and internal serial clock divider is selected. However, AHCLKX pin direction does not change to an output while XCKFAIL is set.
2. The internal clock divider is reset, so that the next clock it produces is a full period. However, the transmit clock divide ratio bits (HCLKXDIV) in AHCLKXCTL are not affected, so the internal clock divider generates clocks at the rate configured.
3. The transmit section is reset for a single serial clock period.
4. The transmit section is released from reset and attempts to begin transmitting. If data is available, it begins transmitting immediately; otherwise, it enters the underrun state. An initial underrun is certain to occur, the pattern 1100 (BMC zeroes) should be shifted out initially.

To change back to an external clock, take the following actions:

1. Wait for the external clock to stabilize again. This can be checked by polling the transmit clock count (XCNT) in XCLKCHK.
2. Reset the transmit section according to the startup procedure in Section 3.6.6.1.

3.6.6.3 Receive Clock Failure Check and Recovery

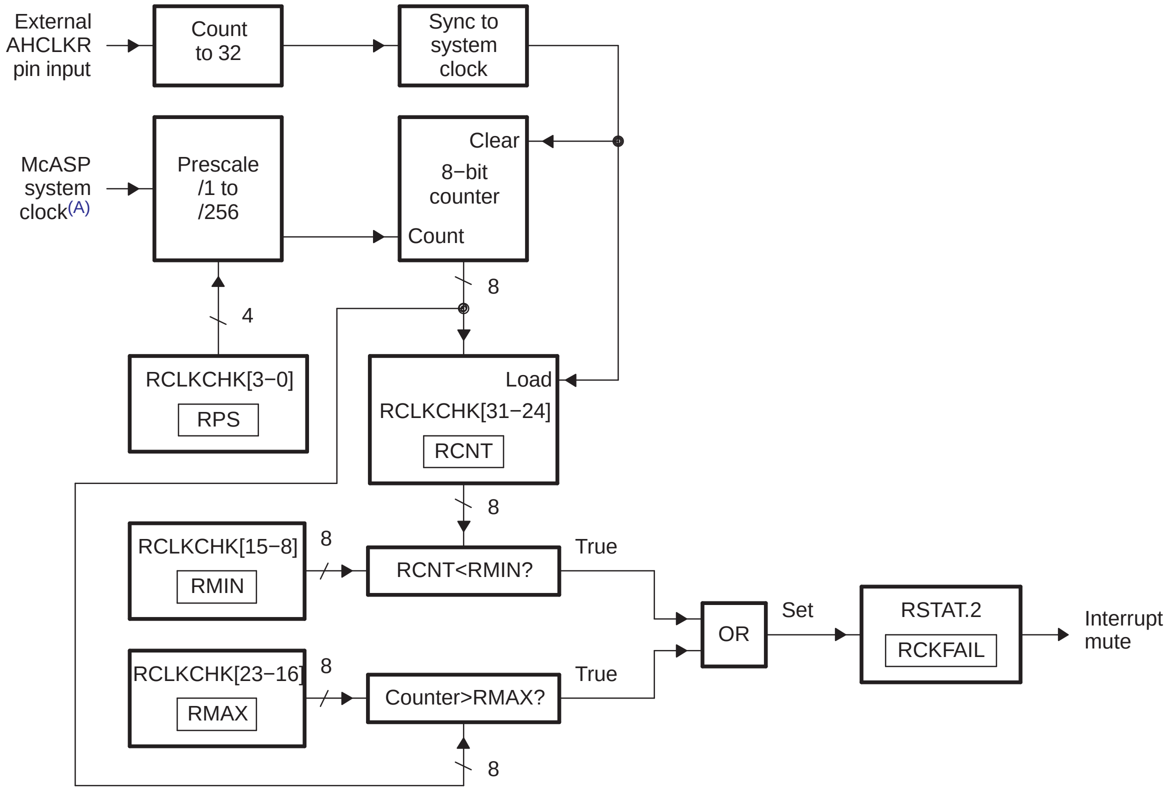
The receive clock failure check circuit (Figure 3-12) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKR). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKR) periods, and stores the count in RCNT of the receive clock check control register (RCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (RMIN) and automatically flags an interrupt (RCKFAIL in RSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than RMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (RMAX). This is in case the external clock completely stops, so that the counter value is not copied to RCNT. An out-of-range maximum condition occurs when the count is greater than RMAX. Note that the RMIN and RMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

In order for the receive clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKR is internally generated or externally sourced.

Figure 3-12. Receive Clock Failure Detection Circuit Block Diagram



A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.

3.7 Loopback Modes

The McASP features a digital loopback mode (DLB) that allows testing of the McASP code in TDM mode with a single DSP device. In loopback mode, output of the transmit serializers is connected internally to the input of the receive serializers. Therefore, you can check the receive data against the transmit data to ensure that the McASP settings are correct. Digital loopback mode applies to TDM mode only (2 to 32 slots in a frame). It does not apply to DIT mode (XMOD = 180h) or burst mode (XMOD = 0).

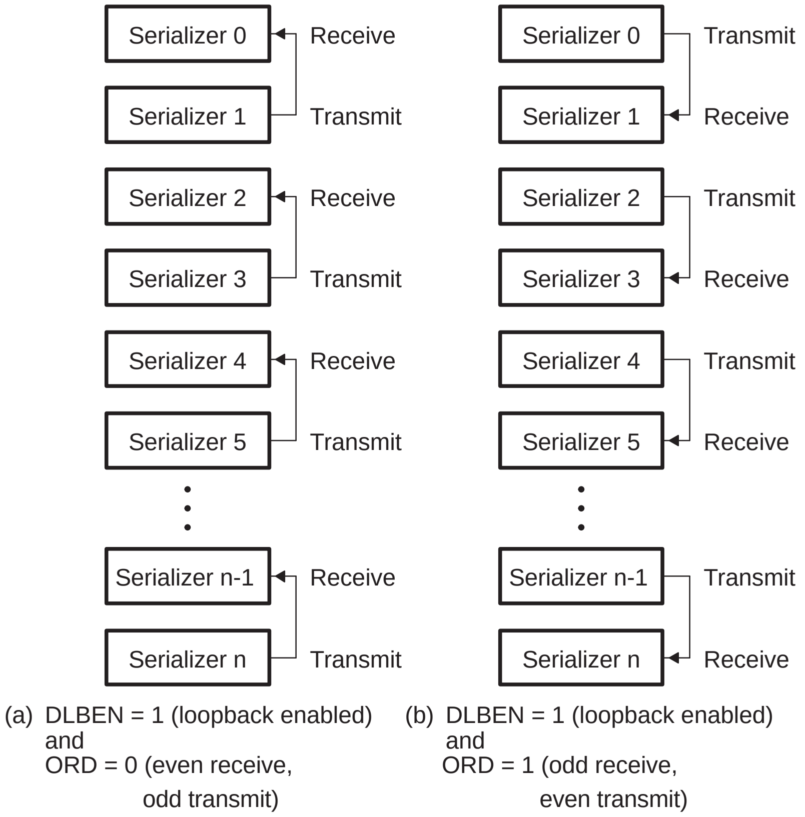
Figure 3-13 shows the basic logical connection of the serializers in loopback mode. Two types of loopback connections are possible, selected by the ORD bit in the digital loopback control register (DLBCTL) as follows:

- ORD = 0: Outputs of odd serializers are connected to inputs of even serializers. If this mode is selected, you should configure odd serializers to be transmitters and even serializers to be receivers.
- ORD = 1: Outputs of even serializers are connected to inputs of odd serializers. If this mode is selected, you should configure even serializers to be transmitters and odd serializers to be receivers.

Data can be externally visible at the I/O pin of the transmit serializer if the pin is configured as a McASP output pin by setting the corresponding PFUNC bit to 0 and PDIR bit to 1.

In loopback mode, the transmit clock and frame sync are used by both the transmit and receive sections of the McASP. The transmit and receive sections operate synchronously. This is achieved by setting the MODE bit of the DLBCTL register to 01b and the ASYNC bit of the ACLKXCTL register to 0.

Figure 3-13. Serializers in Loopback Mode



3.7.1 Loopback Mode Configurations

This is a summary of the settings required for digital loopback mode for TDM format:

- The DLBEN bit in DLBCTL must be set to 1 to enable loopback mode.
- The MODE bits in DLBCTL must be set to 01b for both the transmit and receive sections to use the transmit clock and frame sync generator.
- The ORD bit in DLBCTL must be programmed appropriately to select odd or even serializers to be transmitters or receivers. The corresponding serializers must be configured accordingly.
- The ASYNC bit in ACLKXCTL must be cleared to 0 to ensure synchronous transmit and receive operations.
- RMOD field in AFSRCTL and XMOD field in AFSXCTL must be set to 2h to 20h to indicate TDM mode. Loopback mode does not apply to DIT or burst mode.

Chapter 4

SPRU041J–November 2008

Registers

This chapter describes the registers of the McASP.

4.1 Registers

Control registers for the McASP are summarized in Table 4-1. The control registers are accessed through the configuration bus of the device. The receive buffer registers (RBUF) and transmit buffer registers (XBUF) can also be accessed through the data port of the device, as listed in Table 4-2. See the device-specific data manual for the memory address of these registers.

Table 4-1. McASP Registers Accessed Through Configuration Bus

Offset	Acronym	Register Description	Section
0h	PID	Peripheral identification register	Section 4.2
4h	PWRDEMU	Power down and emulation management register	Section 4.3
10h	PFUNC	Pin function register	Section 4.4
14h	PDIR	Pin direction register	Section 4.5
18h	PDOUT	Pin data output register	Section 4.6
1Ch	PDIN	Read returns: Pin data input register	Section 4.7
1Ch	PDSET	Writes affect: Pin data set register (alternate write address: PDOUT)	Section 4.8
20h	PDCLR	Pin data clear register (alternate write address: PDOUT)	Section 4.9
44h	GBLCTL	Global control register	Section 4.10
48h	AMUTE	Audio mute control register	Section 4.11
4Ch	DLBCTL	Digital loopback control register	Section 4.12
50h	DITCTL	DIT mode control register	Section 4.13
60h	RGBLCTL	Receiver global control register: Alias of GBLCTL, only receive bits are affected - allows receiver to be reset independently from transmitter	Section 4.14
64h	RMASK	Receive format unit bit mask register	Section 4.15
68h	RFMT	Receive bit stream format register	Section 4.16
6Ch	AFSRCTL	Receive frame sync control register	Section 4.17
70h	ACLKRCTL	Receive clock control register	Section 4.18
74h	AHCLKRCTL	Receive high-frequency clock control register	Section 4.19
78h	RTDM	Receive TDM time slot 0-31 register	Section 4.20
7Ch	RINTCTL	Receiver interrupt control register	Section 4.21
80h	RSTAT	Receiver status register	Section 4.22
84h	RSLOT	Current receive TDM time slot register	Section 4.23
88h	RCLKCHK	Receive clock check control register	Section 4.24
8Ch	REVTCTL	Receiver DMA event control register	Section 4.25
ACh	XGBLCTL	Transmitter global control register. Alias of GBLCTL, only transmit bits are affected - allows transmitter to be reset independently from receiver	Section 4.26
A4h	XMASK	Transmit format unit bit mask register	Section 4.27
A8h	XFMT	Transmit bit stream format register	Section 4.28
ACh	AFSXCTL	Transmit frame sync control register	Section 4.29

Table 4-1. McASP Registers Accessed Through Configuration Bus (continued)

Offset	Acronym	Register Description	Section
B0h	ACLKXCTL	Transmit clock control register	Section 4.30
B4h	AHCLKXCTL	Transmit high-frequency clock control register	Section 4.31
B8h	XTDM	Transmit TDM time slot 0-31 register	Section 4.32
BCh	XINTCTL	Transmitter interrupt control register	Section 4.33
C0h	XSTAT	Transmitter status register	Section 4.34
C4h	XSLOT	Current transmit TDM time slot register	Section 4.35
C8h	XCLKCHK	Transmit clock check control register	Section 4.36
CCh	XEVTCTL	Transmitter DMA event control register	Section 4.37
100h	DITCSRA0	Left (even TDM time slot) channel status register (DIT mode) 0	Section 4.39
104h	DITCSRA1	Left (even TDM time slot) channel status register (DIT mode) 1	Section 4.39
108h	DITCSRA2	Left (even TDM time slot) channel status register (DIT mode) 2	Section 4.39
10Ch	DITCSRA3	Left (even TDM time slot) channel status register (DIT mode) 3	Section 4.39
110h	DITCSRA4	Left (even TDM time slot) channel status register (DIT mode) 4	Section 4.39
114h	DITCSRA5	Left (even TDM time slot) channel status register (DIT mode) 5	Section 4.39
118h	DITCSRB0	Right (odd TDM time slot) channel status register (DIT mode) 0	Section 4.40
11Ch	DITCSRB1	Right (odd TDM time slot) channel status register (DIT mode) 1	Section 4.40
120h	DITCSRB2	Right (odd TDM time slot) channel status register (DIT mode) 2	Section 4.40
124h	DITCSRB3	Right (odd TDM time slot) channel status register (DIT mode) 3	Section 4.40
128h	DITCSRB4	Right (odd TDM time slot) channel status register (DIT mode) 4	Section 4.40
12Ch	DITCSRB5	Right (odd TDM time slot) channel status register (DIT mode) 5	Section 4.40
130h	DITUDRA0	Left (even TDM time slot) channel user data register (DIT mode) 0	Section 4.41
134h	DITUDRA1	Left (even TDM time slot) channel user data register (DIT mode) 1	Section 4.41
138h	DITUDRA2	Left (even TDM time slot) channel user data register (DIT mode) 2	Section 4.41
13Ch	DITUDRA3	Left (even TDM time slot) channel user data register (DIT mode) 3	Section 4.41
140h	DITUDRA4	Left (even TDM time slot) channel user data register (DIT mode) 4	Section 4.41
144h	DITUDRA5	Left (even TDM time slot) channel user data register (DIT mode) 5	Section 4.41
148h	DITUDRB0	Right (odd TDM time slot) channel user data register (DIT mode) 0	Section 4.42
14Ch	DITUDRB1	Right (odd TDM time slot) channel user data register (DIT mode) 1	Section 4.42
150h	DITUDRB2	Right (odd TDM time slot) channel user data register (DIT mode) 2	Section 4.42
154h	DITUDRB3	Right (odd TDM time slot) channel user data register (DIT mode) 3	Section 4.42
158h	DITUDRB4	Right (odd TDM time slot) channel user data register (DIT mode) 4	Section 4.42
15Ch	DITUDRB5	Right (odd TDM time slot) channel user data register (DIT mode) 5	Section 4.42
180h	SRCTL0	Serializer control register 0	Section 4.38
184h	SRCTL1	Serializer control register 1	Section 4.38
188h	SRCTL2	Serializer control register 2	Section 4.38
18Ch	SRCTL3	Serializer control register 3	Section 4.38
190h	SRCTL4	Serializer control register 4	Section 4.38
194h	SRCTL5	Serializer control register 5	Section 4.38
198h	SRCTL6	Serializer control register 6	Section 4.38
19Ch	SRCTL7	Serializer control register 7	Section 4.38
1A0h	SRCTL8	Serializer control register 8	Section 4.38
1A4h	SRCTL9	Serializer control register 9	Section 4.38
1A8h	SRCTL10	Serializer control register 10	Section 4.38
1ACh	SRCTL11	Serializer control register 11	Section 4.38
1B0h	SRCTL12	Serializer control register 12	Section 4.38
1B4h	SRCTL13	Serializer control register 13	Section 4.38

Table 4-1. McASP Registers Accessed Through Configuration Bus (continued)

Offset	Acronym	Register Description	Section
1B8h	SRCTL14	Serializer control register 14	Section 4.38
1BCh	SRCTL15	Serializer control register 15	Section 4.38
200h	XBUF0	Transmit buffer register for serializer 0	Section 4.43
204h	XBUF1	Transmit buffer register for serializer 1	Section 4.43
208h	XBUF2	Transmit buffer register for serializer 2	Section 4.43
20Ch	XBUF3	Transmit buffer register for serializer 3	Section 4.43
210h	XBUF4	Transmit buffer register for serializer 4	Section 4.43
214h	XBUF5	Transmit buffer register for serializer 5	Section 4.43
218h	XBUF6	Transmit buffer register for serializer 6	Section 4.43
21Ch	XBUF7	Transmit buffer register for serializer 7	Section 4.43
220h	XBUF8	Transmit buffer register for serializer 8	Section 4.43
224h	XBUF9	Transmit buffer register for serializer 9	Section 4.43
228h	XBUF10	Transmit buffer register for serializer 10	Section 4.43
22Ch	XBUF11	Transmit buffer register for serializer 11	Section 4.43
230h	XBUF12	Transmit buffer register for serializer 12	Section 4.43
234h	XBUF13	Transmit buffer register for serializer 13	Section 4.43
238h	XBUF14	Transmit buffer register for serializer 14	Section 4.43
23Ch	XBUF15	Transmit buffer register for serializer 15	Section 4.43
280h	RBUF0	Receive buffer register for serializer 0	Section 4.44
284h	RBUF1	Receive buffer register for serializer 1	Section 4.44
288h	RBUF2	Receive buffer register for serializer 2	Section 4.44
28Ch	RBUF3	Receive buffer register for serializer 3	Section 4.44
290h	RBUF4	Receive buffer register for serializer 4	Section 4.44
294h	RBUF5	Receive buffer register for serializer 5	Section 4.44
298h	RBUF6	Receive buffer register for serializer 6	Section 4.44
29Ch	RBUF7	Receive buffer register for serializer 7	Section 4.44
2A0h	RBUF8	Receive buffer register for serializer 8	Section 4.44
2A4h	RBUF9	Receive buffer register for serializer 9	Section 4.44
2A8h	RBUF10	Receive buffer register for serializer 10	Section 4.44
2ACh	RBUF11	Receive buffer register for serializer 11	Section 4.44
2B0h	RBUF12	Receive buffer register for serializer 12	Section 4.44
2B4h	RBUF13	Receive buffer register for serializer 13	Section 4.44
2B8h	RBUF14	Receive buffer register for serializer 14	Section 4.44
2BCh	RBUF15	Receive buffer register for serializer 15	Section 4.44

Table 4-2. McASP Registers Accessed Through Data Port

Hex Address	Register Name	Register Description
Read Accesses	RBUF	Receive buffer data port address. Cycles through receive serializers, skipping over transmit serializers and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. DAT BUS only if XBUSEL = 0.
Write Accesses	XBUF	Transmit buffer data port address. Cycles through transmit serializers, skipping over receive and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. DAT BUS only if RBUSEL = 0.

4.2 Peripheral Identification Register (PID)

The peripheral identification register (PID) contains identification data (class, revision, and type) for the peripheral. The PID is shown in Figure 4-1 and described in Table 4-3.

Figure 4-1. Peripheral Identification Register (PID) [Offset 0h]

31 — 24	23 — 16
Reserved[(A)]	TYPE
R-0	R-10h

15 — 8	7 — 0
CLASS	REV
R-01h	R-x[(B)]

LEGEND: R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.
B See the device-specific data manual for the default value of this field.

Table 4-3. Peripheral Identification Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
23-16	TYPE		Identifies type of peripheral.
		10h	McASP
15-8	CLASS		Identifies class of peripheral.
		1	Serial port
7-0	REV		Identifies revision of peripheral.
		x	See the device-specific data manual for the value.

4.3 Power Down and Emulation Management Register (PWRDEMU)

The power down and emulation management register (PWRDEMU) is shown in Figure 4-2 and described in Table 4-4.

Figure 4-2. Power Down and Emulation Management Register (PWRDEMU) [Offset 4h]

31	16
Reserved[A]	
R-0	

15	1	0
Reserved[A]		FREE
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-4. Power Down and Emulation Management Register (PWRDEMU) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	FREE		Free-running mode enable bit. This bit determines the state of the serial port clock during emulation halt.
		0	Reserved.
		1	Free-running mode is enabled. Peripheral ignores the emulation suspend signal and continues to function as normal. During emulation suspend, DMA requests continue to be generated and are serviced by the DMA. Error conditions are flagged as usual.

4.4 Pin Function Register (PFUNC)

The pin function register (PFUNC) specifies the function of AXR[n], ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either a McASP pin or a general-purpose input/output (GPIO) pin. The PFUNC is shown in Figure 4-3 and described in Table 4-5.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation. This includes bits that are not implemented on a particular DSP.

Figure 4-3. Pin Function Register (PFUNC) [Offset 10h]

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved[A]
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23–16
Reserved[A]
R-0

15	14	13	12	11	10	9	8
AXR15[B]	AXR14[B]	AXR13[B]	AXR12[B]	AXR11[B]	AXR10[B]	AXR9[B]	AXR8[B]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.
B On DA6x DSP only; reserved on C6713 DSP.

Table 4-5. Pin Function Register (PFUNC) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Determines if AFSR pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
30	AHCLKR		Determines if AHCLKR pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
29	ACLKR		Determines if ACLKR pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
28	AFSX		Determines if AFSX pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
27	AHCLKX		Determines if AHCLKX pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
26	ACLKX		Determines if ACLKX pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
25	AMUTE		Determines if AMUTE pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
24-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-8	AXR[15-8] [1]		Determines if AXR[n] pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
7-0	AXR[7-0]		Determines if AXR[n] pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.

[1] On DA6x DSP only; reserved on C6713 DSP.

4.5 Pin Direction Register (PDIR)

The pin direction register (PDIR) specifies the direction of AXR[n], ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either an input or an output pin. The PDIR is shown in Figure 4-4 and described in Table 4-6.

Regardless of the pin function register (PFUNC) setting, each PDIR bit must be set to 1 for the specified pin to be enabled as an output and each PDIR bit must be cleared to 0 for the specified pin to be an input.

For example, if the McASP is configured to use an internally-generated bit clock and the clock is to be driven out to the system, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output).

When AXR[n] is configured to transmit, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output). Similarly, when AXR[n] is configured to receive, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be cleared to 0 (an input).

> **CAUTION**
>
> **Writing to Reserved Bits**
>
> Writing a value other than 0 to reserved bits in this register may cause improper device operation. This includes bits that are not implemented on a particular DSP.

Figure 4-4. Pin Direction Register (PDIR) [Offset 14h]

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved[A]
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	16
Reserved[A]	
R-0	

15	14	13	12	11	10	9	8
AXR15[B]	AXR14[B]	AXR13[B]	AXR12[B]	AXR11[B]	AXR10[B]	AXR9[B]	AXR8[B]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.
B On DA6x DSP only; reserved on C6713 DSP.

Table 4-6. Pin Direction Register (PDIR) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Determines if AFSR pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
30	AHCLKR		Determines if AHCLKR pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
29	ACLKR		Determines if ACLKR pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
28	AFSX		Determines if AFSX pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
27	AHCLKX		Determines if AHCLKX pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
26	ACLKX		Determines if ACLKX pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
25	AMUTE		Determines if AMUTE pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
24-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-8	AXR[15-8][1]		Determines if AXR[n] pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
7-0	AXR[7-0]		Determines if AXR[n] pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.

[1] On DA6x DSP only; reserved on C6713 DSP.

4.6 Pin Data Output Register (PDOUT)

The pin data output register (PDOUT) holds a value for data out at all times, and may be read back at all times. The value held by PDOUT is not affected by writing to PDIR and PFUNC. However, the data value in PDOUT is driven out onto the McASP pin only if the corresponding bit in PFUNC is set to 1 (GPIO function) and the corresponding bit in PDIR is set to 1 (output). When reading data, returns the corresponding bit value in PDOUT[n], does not return input from I/O pin; when writing data, writes to the corresponding PDOUT[n] bit. The PDOUT is shown in Figure 4-5 and described in Table 4-7.

PDOUT has these aliases or alternate addresses:

- PDSET - when written to at this address, writing a 1 to a bit in PDSET sets the corresponding bit in PDOUT to 1; writing a 0 has no effect and keeps the bits in PDOUT unchanged.
- PDCLR - when written to at this address, writing a 1 to a bit in PDCLR clears the corresponding bit in PDOUT to 0; writing a 0 has no effect and keeps the bits in PDOUT unchanged.

There is only one set of data out bits, PDOUT[31-0]. The other registers, PDSET and PDCLR, are just different addresses for the same control bits, with different behaviors during writes.

> **CAUTION**
>
> **Writing to Reserved Bits**
>
> Writing a value other than 0 to reserved bits in this register may cause improper device operation. This includes bits that are not implemented on a particular DSP.

Figure 4-5. Pin Data Output Register (PDOUT) [Offset 18h]

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved[A]
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23							16
Reserved[A]							
R-0							

15	14	13	12	11	10	9	8
AXR15[B]	AXR14[B]	AXR13[B]	AXR12[B]	AXR11[B]	AXR10[B]	AXR9[B]	AXR8[B]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.
B On DA6x DSP only; reserved on C6713 DSP.

Table 4-7. Pin Data Output Register (PDOUT) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Determines drive on AFSR output pin when the corresponding PFUNC[31] and PDIR[31] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
30	AHCLKR		Determines drive on AHCLKR output pin when the corresponding PFUNC[30] and PDIR[30] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
29	ACLKR		Determines drive on ACLKR output pin when the corresponding PFUNC[29] and PDIR[29] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
28	AFSX		Determines drive on AFSX output pin when the corresponding PFUNC[28] and PDIR[28] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
27	AHCLKX		Determines drive on AHCLKX output pin when the corresponding PFUNC[27] and PDIR[27] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
26	ACLKX		Determines drive on ACLKX output pin when the corresponding PFUNC[26] and PDIR[26] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
25	AMUTE		Determines drive on AMUTE output pin when the corresponding PFUNC[25] and PDIR[25] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
24-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-8	AXR[15-8][1]		Determines drive on AXR[n] output pin when the corresponding PFUNC[n] and PDIR[n] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
7-0	AXR[7-0]		Determines drive on AXR[n] output pin when the corresponding PFUNC[n] and PDIR[n] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.

[1] On DA6x DSP only; reserved on C6713 DSP.

4.7 Pin Data Input Register (PDIN)

The pin data input register (PDIN) holds the I/O pin state of each of the McASP pins. PDIN allows the actual value of the pin to be read, regardless of the state of PFUNC and PDIR. The value after reset for registers 1 through 15 and 24 through 31 depends on how the pins are being driven. The PDIN is shown in Figure 4-6 and described in Table 4-8.

> **CAUTION**
>
> **Writing to Reserved Bits**
>
> Writing a value other than 0 to reserved bits in this register may cause improper device operation. This includes bits that are not implemented on a particular DSP.

Figure 4-6. Pin Data Input Register (PDIN) [Offset 1Ch]

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved[A]
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	16
Reserved[A]	
R-0	

15	14	13	12	11	10	9	8
AXR15[B]	AXR14[B]	AXR13[B]	AXR12[B]	AXR11[B]	AXR10[B]	AXR9[B]	AXR8[B]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[A] If writing to this field, always write the default value for future device compatibility.
[B] On DA6x DSP only; reserved on C6713 DSP.

Table 4-8. Pin Data Input Register (PDIN) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Logic level on AFSR pin.
		0	Pin is logic low.
		1	Pin is logic high.
30	AHCLKR		Logic level on AHCLKR pin.
		0	Pin is logic low.
		1	Pin is logic high.
29	ACLKR		Logic level on ACLKR pin.
		0	Pin is logic low.
		1	Pin is logic high.
28	AFSX		Logic level on AFSX pin.
		0	Pin is logic low.
		1	Pin is logic high.
27	AHCLKX		Logic level on AHCLKX pin.
		0	Pin is logic low.
		1	Pin is logic high.
26	ACLKX		Logic level on ACLKX pin.
		0	Pin is logic low.
		1	Pin is logic high.
25	AMUTE		Logic level on AMUTE pin.
		0	Pin is logic low.
		1	Pin is logic high.
24-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-8	AXR[15-8][1]		Logic level on AXR[n] pin.
		0	Pin is logic low.
		1	Pin is logic high.
7-0	AXR[7-0]		Logic level on AXR[n] pin.
		0	Pin is logic low.
		1	Pin is logic high.

[1] On DA6x DSP only; reserved on C6713 DSP.

4.8 Pin Data Set Register (PDSET)

The pin data set register (PDSET) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDSET bit sets the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic high on the pin. PDSET is useful for a multitasking system because it allows you to set to a logic high only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. The PDSET is shown in Figure 4-7 and described in Table 4-9.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation. This includes bits that are not implemented on a particular DSP.

Figure 4-7. Pin Data Set Register (PDSET) [Offset 1Ch]

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved[A]
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	16
Reserved[A]	
R-0	

15	14	13	12	11	10	9	8
AXR15[B]	AXR14[B]	AXR13[B]	AXR12[B]	AXR11[B]	AXR10[B]	AXR9[B]	AXR8[B]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[A] If writing to this field, always write the default value for future device compatibility.

[B] On DA6x DSP only; reserved on C6713 DSP.

Table 4-9. Pin Data Set Register (PDSET) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Allows the corresponding AFSR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[31] bit is set to 1.
30	AHCLKR		Allows the corresponding AHCLKR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[30] bit is set to 1.
29	ACLKR		Allows the corresponding ACLKR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[29] bit is set to 1.
28	AFSX		Allows the corresponding AFSX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[28] bit is set to 1.
27	AHCLKX		Allows the corresponding AHCLKX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[27] bit is set to 1.
26	ACLKX		Allows the corresponding ACLKX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[26] bit is set to 1.
25	AMUTE		Allows the corresponding AMUTE bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[25] bit is set to 1.
24-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-8	AXR[15-8] [1]		Allows the corresponding AXR[n] bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[n] bit is set to 1.
7-0	AXR[7-0]		Allows the corresponding AXR[n] bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[n] bit is set to 1.

[1] On DA6x DSP only; reserved on C6713 DSP.

4.9 Pin Data Clear Register (PDCLR)

The pin data clear register (PDCLR) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDCLR bit clears the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic low on the pin. PDCLR is useful for a multitasking system because it allows you to clear to a logic low only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. The PDCLR is shown in Figure 4-8 and described in Table 4-10.

> **CAUTION**
>
> **Writing to Reserved Bits**
>
> Writing a value other than 0 to reserved bits in this register may cause improper device operation. This includes bits that are not implemented on a particular DSP.

Figure 4-8. Pin Data Clear Register (PDCLR) [Offset 20h]

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved[A]
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	16
Reserved[A]	
R-0	

15	14	13	12	11	10	9	8
AXR15[B]	AXR14[B]	AXR13[B]	AXR12[B]	AXR11[B]	AXR10[B]	AXR9[B]	AXR8[B]
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
AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.
B On DA6x DSP only; reserved on C6713 DSP.

Table 4-10. Pin Data Clear Register (PDCLR) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Allows the corresponding AFSR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[31] bit is cleared to 0.
30	AHCLKR		Allows the corresponding AHCLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[30] bit is cleared to 0.
29	ACLKR		Allows the corresponding ACLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[29] bit is cleared to 0.
28	AFSX		Allows the corresponding AFSX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[28] bit is cleared to 0.
27	AHCLKX		Allows the corresponding AHCLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[27] bit is cleared to 0.
26	ACLKX		Allows the corresponding ACLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[26] bit is cleared to 0.
25	AMUTE		Allows the corresponding AMUTE bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[25] bit is cleared to 0.
24-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-8	AXR[15-8] [1]		Allows the corresponding AXR[n] bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[n] bit is cleared to 0.
7-0	AXR[7-0]		Allows the corresponding AXR[n] bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[n] bit is cleared to 0.

[1] On DA6x DSP only; reserved on C6713 DSP.

4.10 Global Control Register (GBLCTL)

The global control register (GBLCTL) provides initialization of the transmit and receive sections. The GBLCTL is shown in Figure 4-9 and described in Table 4-11.

The bit fields in GBLCTL are synchronized and latched by the corresponding clocks (ACLKX for bits 12-8 and ACLKR for bits 4-0). Before GBLCTL is programmed, you must ensure that serial clocks are running. If the corresponding external serial clocks, ACLKX and ACLKR, are not yet running, you should select the internal serial clock source in AHCLKXCTL, AHCLKRCTL, ACLKXCTL, and ACLKRCTL before GBLCTL is programmed. Also, after programming any bits in GBLCTL you should not proceed until you have read back from GBLCTL and verified that the bits are latched in GBLCTL.

Figure 4-9. Global Control Register (GBLCTL) [Offset 44h]

31-16
Reserved[A]
R-0

15-13	12	11	10	9	8
Reserved[A]	XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7-5	4	3	2	1	0
Reserved[A]	RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-11. Global Control Register (GBLCTL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XFRST		Transmit frame sync generator reset enable bit.
		0	Transmit frame sync generator is reset.
		1	Transmit frame sync generator is active. When released from reset, the transmit frame sync generator begins counting serial clocks and generating frame sync as programmed.
11	XSMRST		Transmit state machine reset enable bit.
		0	Transmit state machine is held in reset. AXR[n] pin state: If PFUNC[n] = 0 and PDIR[n] = 1; then the serializer drives the AXR[n] pin to the state specified for inactive time slot (as determined by DISMOD bits in SRCTL).
		1	Transmit state machine is released from reset. When released from reset, the transmit state machine immediately transfers data from XRBUF[n] to XRSR[n]. The transmit state machine sets the underrun flag (XUNDRN) in XSTAT, if XRBUF[n] have not been preloaded with data before reset is released. The transmit state machine also immediately begins detecting frame sync and is ready to transmit. Transmit TDM time slot begins at slot 0 after reset is released.
10	XSRCLR		Transmit serializer clear enable bit. By clearing then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun will occur.
		0	Transmit serializers are cleared.
		1	Transmit serializers are active. When the transmit serializers are first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in XSTAT is set to indicate XBUF is ready to be written.
9	XHCLKRST		Transmit high-frequency clock divider reset enable bit.
		0	Transmit high-frequency clock divider is held in reset and passes through its input as divide-by-1..
		1	Transmit high-frequency clock divider is running.

Table 4-11. Global Control Register (GBLCTL) Field Descriptions (continued)

Bit	Field	Value	Description
8	XCLKRST		Transmit clock divider reset enable bit.
		0	Transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input.
		1	Transmit clock divider is running.
7-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	RFRST		Receive frame sync generator reset enable bit.
		0	Receive frame sync generator is reset.
		1	Receive frame sync generator is active. When released from reset, the receive frame sync generator begins counting serial clocks and generating frame sync as programmed.
3	RSMRST		Receive state machine reset enable bit.
		0	Receive state machine is held in reset.
		1	Receive state machine is released from reset. When released from reset, the receive state machine immediately begins detecting frame sync and is ready to receive. Receive TDM time slot begins at slot 0 after reset is released.
2	RSRCLR		Receive serializer clear enable bit. By clearing then setting this bit, the receive buffer is flushed.
		0	Receive serializers are cleared.
		1	Receive serializers are active.
1	RHCLKRST		Receive high-frequency clock divider reset enable bit.
		0	Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1.
		1	Receive high-frequency clock divider is running.
0	RCLKRST		Receive high-frequency clock divider reset enable bit.
		0	Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input.
		1	Receive clock divider is running.

4.11 Audio Mute Control Register (AMUTE)

The audio mute control register (AMUTE) controls the McASP audio mute (AMUTE) output pin. The value after reset for register 4 depends on how the pins are being driven. The AMUTE is shown in Figure 4-10 and described in Table 4-12.

Figure 4-10. Audio Mute Control Register (AMUTE) [Offset 48h]

31–16
Reserved[A]
R-0

15–13	12	11	10	9	8
Reserved[A]	XDMAERR	RDMAERR	XCKFAIL	RCKFAIL	XSYNCERR
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1–0
RSYNCERR	XUNDRN	ROVRN	INSTAT	INEN	INPOL	MUTEN
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-12. Audio Mute Control Register (AMUTE) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XDMAERR		If transmit DMA error (XDMAERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of transmit DMA error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of transmit DMA error, AMUTE is active and is driven according to MUTEN bit.
11	RDMAERR		If receive DMA error (RDMAERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of receive DMA error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of receive DMA error, AMUTE is active and is driven according to MUTEN bit.
10	XCKFAIL		If transmit clock failure (XCKFAIL), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of transmit clock failure is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of transmit clock failure, AMUTE is active and is driven according to MUTEN bit
9	RCKFAIL		If receive clock failure (RCKFAIL), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of receive clock failure is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of receive clock failure, AMUTE is active and is driven according to MUTEN bit.
8	XSYNCERR		If unexpected transmit frame sync error (XSYNCERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of unexpected transmit frame sync error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of unexpected transmit frame sync error, AMUTE is active and is driven according to MUTEN bit.
7	RSYNCERR		If unexpected receive frame sync error (RSYNCERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of unexpected receive frame sync error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of unexpected receive frame sync error, AMUTE is active and is driven according to MUTEN bit.

Table 4-12. Audio Mute Control Register (AMUTE) Field Descriptions (continued)

Bit	Field	Value	Description
6	XUNDRN		If transmit underrun error (XUNDRN), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of transmit underrun error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of transmit underrun error, AMUTE is active and is driven according to MUTEN bit.
5	ROVRN		If receiver overrun error (ROVRN), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of receiver overrun error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of receiver overrun error, AMUTE is active and is driven according to MUTEN bit.
4	INSTAT		Determines drive on AXR[n] pin when PFUNC[n] and PDIR[n] bits are set to 1.
		0	AMUTEIN pin is inactive.
		1	AMUTEIN pin is active. Audio mute in error is detected.
3	INEN		Drive AMUTE active when AMUTEIN error is active (INSTAT = 1).
		0	Drive is disabled. AMUTEIN is ignored by AMUTE.
		1	Drive is enabled (active). INSTAT = 1 drives AMUTE active.
2	INPOL		Audio mute in (AMUTEIN) polarity select bit.
		0	Polarity is active high. A high on AMUTEIN sets INSTAT to 1.
		1	Polarity is active low. A low on AMUTEIN sets INSTAT to 1.
1-0	MUTEN	0-3h	AMUTE pin enable bit (unless overridden by GPIO registers).
		0	AMUTE pin is disabled, pin goes to tri-state condition.
		1h	AMUTE pin is driven high if error is detected.
		2h	AMUTE pin is driven low if error is detected.
		3h	Reserved

4.12 Digital Loopback Control Register (DLBCTL)

The digital loopback control register (DLBCTL) controls the internal loopback settings of the McASP in TDM mode. The DLBCTL is shown in Figure 4-11 and described in Table 4-13.

Figure 4-11. Digital Loopback Control Register (DLBCTL) [Offset 4Ch]

31	16
Reserved[A]	
R-0	

15	4	3	2	1	0
Reserved[A]		MODE		ORD	DLBEN
R-0		R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[A] If writing to this field, always write the default value for future device compatibility.

Table 4-13. Digital Loopback Control Register (DLBCTL) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3-2	MODE	0-3h	Loopback generator mode bits. Applies only when loopback mode is enabled (DLBEN = 1).
		0	Default and reserved on loopback mode (DLBEN = 1). When in non-loopback mode (DLBEN = 0), MODE should be left at default (00). When in loopback mode (DLBEN = 1), MODE = 00 is reserved and not applicable.
		1h	Transmit clock and frame sync generators used by both transmit and receive sections. When in loopback mode (DLBEN = 1), MODE must be 01.
		2h-3h	Reserved.
1	ORD		Loopback order bit when loopback mode is enabled (DLBEN = 1).
		0	Odd serializers N + 1 transmit to even serializers N that receive. The corresponding serializers must be programmed properly.
		1	Even serializers N transmit to odd serializers N+1 that receive. The corresponding serializers must be programmed properly.
0	DLBEN		Loopback mode enable bit.
		0	Loopback mode is disabled.
		1	Loopback mode is enabled.

4.13 Digital Mode Control Register (DITCTL)

The DIT mode control register (DITCTL) controls DIT operations of the McASP. The DITCTL is shown in Figure 4-12 and described in Table 4-14.

Figure 4-12. Digital Mode Control Register (DITCTL) [Offset 50h]

31	16
Reserved[A]	
R-0	

15	4	3	2	1	0
Reserved[A]		VB	VA	Rsvd[A]	DITEN
R-0		R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[A] If writing to this field, always write the default value for future device compatibility.

Table 4-14. Digital Mode Control Register (DITCTL) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3	VB		Valid bit for odd time slots (DIT right subframe).
		0	V bit is 0 during odd DIT subframes.
		1	V bit is 1 during odd DIT subframes.
2	VA		Valid bit for even time slots (DIT left subframe).
		0	V bit is 0 during even DIT subframes.
		1	V bit is 1 during even DIT subframes.
1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	DITEN		DIT mode enable bit. DITEN should only be changed while the XSMRST bit in GBLCTL is in reset (and for startup, XSRCLR also in reset). However, it is not necessary to reset the XCLKRST or XHCLKRST bits in GBLCTL to change DITEN.
		0	DIT mode is disabled. Transmitter operates in TDM or burst mode.
		1	DIT mode is enabled. Transmitter operates in DIT encoded mode.

4.14 Receiver Global Control Register (RGBLCTL)

Alias of the global control register (GBLCTL). Writing to the receiver global control register (RGBLCTL) affects only the receive bits of GBLCTL (bits 4-0). Reads from RGBLCTL return the value of GBLCTL. RGBLCTL allows the receiver to be reset independently from the transmitter. The RGBLCTL is shown in Figure 4-13 and described in Table 4-15. See Section 4.10 for a detailed description of GBLCTL.

Figure 4-13. Receiver Global Control Register (RGBLCTL) [Offset 60h]

31–16
Reserved[A]
R-0

15–13	12	11	10	9	8
Reserved[A]	XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0	R-0	R-0	R-0	R-0	R-0

7–5	4	3	2	1	0
Reserved[A]	RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-15. Receiver Global Control Register (RGBLCTL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XFRST	x	Transmit frame sync generator reset enable bit. A read of this bit returns the XFRST bit value of GBLCTL. Writes have no effect.
11	XSMRST	x	Transmit state machine reset enable bit. A read of this bit returns the XSMRST bit value of GBLCTL. Writes have no effect.
10	XSRCLR	x	Transmit serializer clear enable bit. A read of this bit returns the XSRCLR bit value of GBLCTL. Writes have no effect.
9	XHCLKRST	x	Transmit high-frequency clock divider reset enable bit. A read of this bit returns the XHCLKRST bit value of GBLCTL. Writes have no effect.
8	XCLKRST	x	Transmit clock divider reset enable bit. a read of this bit returns the XCLKRST bit value of GBLCTL. Writes have no effect.
7-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	RFRST		Receive frame sync generator reset enable bit. A write to this bit affects the RFRST bit of GBLCTL.
		0	Receive frame sync generator is reset.
		1	Receive frame sync generator is active.
3	RSMRST		Receive state machine reset enable bit. A write to this bit affects the RSMRST bit of GBLCTL.
		0	Receive state machine is held in reset.
		1	Receive state machine is released from reset.
2	RSRCLR		Receive serializer clear enable bit. A write to this bit affects the RSRCLR bit of GBLCTL.
		0	Receive serializers are cleared.
		1	Receive serializers are active.
1	RHCLKRST		Receive high-frequency clock divider reset enable bit. A write to this bit affects the RHCLKRST bit of GBLCTL.
		0	Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1.
		1	Receive high-frequency clock divider is running.

Table 4-15. Receiver Global Control Register (RGBLCTL) Field Descriptions (continued)

Bit	Field	Value	Description
0	RCLKRST		Receive clock divider reset enable bit. A write to this bit affects the RCLKRST bit of GBLCTL.
		0	Receive clock divider is held in reset.
		1	Receive clock divider is running.

4.15 Receive Format Unit Bit Mask Register (RMASK)

The receive format unit bit mask register (RMASK) determines which bits of the received data are masked off and padded with a known value before being read by the CPU or DMA. The RMASK is shown in Figure 4-14 and described in Table 4-16.

Figure 4-14. Receive Format Unit Bit Mask Register (RMASK) [Offset 64h]

31	30	29	28	27	26	25	24
RMASK31	RMASK30	RMASK29	RMASK28	RMASK27	RMASK26	RMASK25	RMASK24
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	22	21	20	19	18	17	16
RMASK23	RMASK22	RMASK21	RMASK20	RMASK19	RMASK18	RMASK17	RMASK16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
RMASK15	RMASK14	RMASK13	RMASK12	RMASK11	RMASK10	RMASK9	RMASK8
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
RMASK7	RMASK6	RMASK5	RMASK4	RMASK3	RMASK2	RMASK1	RMASK0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 4-16. Receive Format Unit Bit Mask Register (RMASK) Field Descriptions

Bit	Field	Value	Description
31-0	RMASK[31-0]		Receive data mask enable bit.
		0	Corresponding bit of receive data (after passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (RPAD and RPBIT bits in RFMT).
		1	Corresponding bit of receive data (after passing through reverse and rotate units) is returned to CPU or DMA.

4.16 Receive Bit Stream Format Register (RFMT)

The receive bit stream format register (RFMT) configures the receive data format. The RFMT is shown in Figure 4-15 and described in Table 4-17.

Figure 4-15. Receive Bit Stream Format Register (RFMT) [Offset 68h]

31 18	17 16
Reserved[A]	RDATDLY
R-0	R/W-0

15	14 13	12 8	7 4	3	2 0
RRVRS	RPAD	RPBIT	RSSZ	RBUSEL	RROT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-17. Receive Bit Stream Format Register (RFMT) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
17-16	RDATDLY	0-3h	Receive bit delay.
		0	0-bit delay. The first receive data bit, AXR[n], occurs in same ACLKR cycle as the receive frame sync (AFSR).
		1h	1-bit delay. The first receive data bit, AXR[n], occurs one ACLKR cycle after the receive frame sync (AFSR).
		2h	2-bit delay. The first receive data bit, AXR[n], occurs two ACLKR cycles after the receive frame sync (AFSR).
		3h	Reserved.
15	RRVRS		Receive serial bitstream order.
		0	Bitstream is LSB first. No bit reversal is performed in receive format bit reverse unit.
		1	Bitstream is MSB first. Bit reversal is performed in receive format bit reverse unit.
14-13	RPAD	0-3h	Pad value for extra bits in slot not belonging to the word. This field only applies to bits when RMASK[n] = 0.
		0	Pad extra bits with 0.
		1h	Pad extra bits with 1.
		2h	Pad extra bits with one of the bits from the word as specified by RPBIT bits.
		3h	Reserved.
12-8	RPBIT	0-1Fh	RPBIT value determines which bit (as read by the CPU or DMA from RBUF[n]) is used to pad the extra bits. This field only applies when RPAD = 2h.
		0	Pad with bit 0 value.
		1h-1Fh	Pad with bit 1 to bit 31 value.

Table 4-17. Receive Bit Stream Format Register (RFMT) Field Descriptions (continued)

Bit	Field	Value	Description
7-4	RSSZ	0-Fh	Receive slot size.
		0-2h	Reserved
		3h	Slot size is 8 bits.
		4h	Reserved
		5h	Slot size is 12 bits.
		6h	Reserved
		7h	Slot size is 16 bits.
		8h	Reserved
		9h	Slot size is 20 bits.
		Ah	Reserved
		Bh	Slot size is 24 bits
		Ch	Reserved
		Dh	Slot size is 28 bits.
		Eh	Reserved
		Fh	Slot size is 32 bits.
3	RBUSEL		Selects whether reads from serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port.
		0	Reads from XRBUF[n] originate on data port. Reads from XRBUF[n] on configuration bus are ignored.
		1	Reads from XRBUF[n] originate on configuration bus. Reads from XRBUF[n] on data port are ignored.
2-0	RROT	0-7h	Right-rotation value for receive rotate right format unit.
		0	Rotate right by 0 (no rotation).
		1h	Rotate right by 4 bit positions.
		2h	Rotate right by 8 bit positions.
		3h	Rotate right by 12 bit positions.
		4h	Rotate right by 16 bit positions.
		5h	Rotate right by 20 bit positions.
		6h	Rotate right by 24 bit positions.
		7h	Rotate right by 28 bit positions.

4.17 Receive Frame Sync Control Register (AFSRCTL)

The receive frame sync control register (AFSRCTL) configures the receive frame sync (AFSR). The AFSRCTL is shown in Figure 4-16 and described in Table 4-18.

Figure 4-16. Receive Frame Sync Control Register (AFSRCTL) [Offset 6Ch]

31–16
Reserved[(A)]
R-0

15–7	6–5	4	3–2	1	0
RMOD	Reserved[(A)]	FRWID	Reserved[(A)]	FSRM	FSRP
R/W-0	R-0	R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-18. Receive Frame Sync Control Register (AFSRCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-7	RMOD	0-1FFh	Receive frame sync mode select bits.
		0	Burst mode
		1h	Reserved
		2h-20h	2-slot TDM (I2S mode) to 32-slot TDM
		21h-17Fh	Reserved
		180h	384-slot TDM (external DIR IC inputting 384-slot DIR frames to McASP over I2S interface)
		181h-1FFh	Reserved
6-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	FRWID		Receive frame sync width select bit indicates the width of the receive frame sync (AFSR) during its active period.
		0	Single bit
		1	Single word
3-2	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
1	FSRM		Receive frame sync generation select bit.
		0	Externally-generated receive frame sync
		1	Internally-generated receive frame sync
0	FSRP		Receive frame sync polarity select bit.
		0	A rising edge on receive frame sync (AFSR) indicates the beginning of a frame.
		1	A falling edge on receive frame sync (AFSR) indicates the beginning of a frame.

4.18 Receive Clock Control Register (ACLKRCTL)

The receive clock control register (ACLKRCTL) configures the receive bit clock (ACLKR) and the receive clock generator. The ACLKRCTL is shown in Figure 4-17 and described in Table 4-19.

Figure 4-17. Receive Clock Control Register (ACLKRCTL) [Offset 70h]

31 16
Reserved[A]
R-0

15 8	7	6	5	4 0
Reserved[A]	CLKRP	Rsvd[A]	CLKRM	CLKRDIV
R-0	R/W-0	R-0	R/W-1	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-19. Receive Clock Control Register (ACLKRCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	CLKRP		Receive bitstream clock polarity select bit.
		0	Falling edge. Receiver samples data on the falling edge of the serial clock, so the external transmitter driving this receiver must shift data out on the rising edge of the serial clock.
		1	Rising edge. Receiver samples data on the rising edge of the serial clock, so the external transmitter driving this receiver must shift data out on the falling edge of the serial clock.
6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	CLKRM		Receive bit clock source bit.
		0	External receive clock source from ACLKR pin.
		1	Internal receive clock source from output of programmable bit clock divider.
4-0	CLKRDIV	0-1Fh	Receive bit clock divide ratio bits determine the divide-down ratio from AHCLKR to ACLKR.
		0	Divide-by-1
		1h	Divide-by-2
		2h-1Fh	Divide-by-3 to divide-by-32

4.19 Receive High-Frequency Clock Control Register (AHCLKRCTL)

The receive high-frequency clock control register (AHCLKRCTL) configures the receive high-frequency master clock (AHCLKR) and the receive clock generator. The AHCLKRCTL is shown in Figure 4-18 and described in Table 4-20.

Figure 4-18. Receive High-Frequency Clock Control Register (AHCLKRCTL) [Offset 74h]

31 16
Reserved[A]
R-0

15	14	13 12	11 0
HCLKRM	HCLKRP	Reserved[A]	HCLKRDIV
R/W-1	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-20. Receive High-Frequency Clock Control Register (AHCLKRCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15	HCLKRM		Receive high-frequency clock source bit.
		0	External receive high-frequency clock source from AHCLKR pin.
		1	Internal receive high-frequency clock source from output of programmable high clock divider.
14	HCLKRP		Receive bitstream high-frequency clock polarity select bit.
		0	Rising edge. AHCLKR is not inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin.
		1	Falling edge. AHCLKR is inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin.
13-12	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
11-0	HCLKRDIV	0-FFFh	Receive high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKR.
		0	Divide-by-1
		1h	Divide-by-2
		2h-FFFh	Divide-by-3 to divide-by-4096

4.20 Receive TDM Time Slot Register (RTDM)

The receive TDM time slot register (RTDM) specifies which TDM time slot the receiver is active. The RTDM is shown in Figure 4-19 and described in Table 4-21.

Figure 4-19. Receive TDM Time Slot Register (RTDM) [Offset 78h]

31	30	29	28	27	26	25	24
RTDMS31	RTDMS30	RTDMS29	RTDMS28	RTDMS27	RTDMS26	RTDMS25	RTDMS24
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	22	21	20	19	18	17	16
RTDMS23	RTDMS22	RTDMS21	RTDMS20	RTDMS19	RTDMS18	RTDMS17	RTDMS16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
RTDMS15	RTDMS14	RTDMS13	RTDMS12	RTDMS11	RTDMS10	RTDMS9	RTDMS8
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
RTDMS7	RTDMS6	RTDMS5	RTDMS4	RTDMS3	RTDMS2	RTDMS1	RTDMS0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 4-21. Receive TDM Time Slot Register (RTDM) Field Descriptions

Bit	Field	Value	Description
31-0	RTDMS[31-0]		Receiver mode during TDM time slot *n*.
		0	Receive TDM time slot *n* is inactive. The receive serializer does not shift in data during this slot.
		1	Receive TDM time slot *n* is active. The receive serializer shifts in data during this slot.

4.21 Receiver Interrupt Control Register (RINTCTL)

The receiver interrupt control register (RINTCTL) controls generation of the McASP receive interrupt (RINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates RINT. The RINTCTL is shown in Figure 4-20 and described in Table 4-22. See Section 4.22 for a description of the interrupt conditions.

Figure 4-20. Receiver Interrupt Control Register (RINTCTL) [Offset 7Ch]

31–8
Reserved[A]
R-0

7	6	5	4	3	2	1	0
RSTAFRM	Reserved[A]	RDATA	RLAST	RDMAERR	RCKFAIL	RSYNCERR	ROVRN
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-22. Receiver Interrupt Control Register (RINTCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	RSTAFRM		Receive start of frame interrupt enable bit.
		0	Interrupt is disabled. A receive start of frame interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive start of frame interrupt generates a McASP receive interrupt (RINT).
6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	RDATA		Receive data ready interrupt enable bit.
		0	Interrupt is disabled. A receive data ready interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive data ready interrupt generates a McASP receive interrupt (RINT).
4	RLAST		Receive last slot interrupt enable bit.
		0	Interrupt is disabled. A receive last slot interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive last slot interrupt generates a McASP receive interrupt (RINT).
3	RDMAERR		Receive DMA error interrupt enable bit.
		0	Interrupt is disabled. A receive DMA error interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive DMA error interrupt generates a McASP receive interrupt (RINT).
2	RCKFAIL		Receive clock failure interrupt enable bit.
		0	Interrupt is disabled. A receive clock failure interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive clock failure interrupt generates a McASP receive interrupt (RINT).
1	RSYNCERR		Unexpected receive frame sync interrupt enable bit.
		0	Interrupt is disabled. An unexpected receive frame sync interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. An unexpected receive frame sync interrupt generates a McASP receive interrupt (RINT).
0	ROVRN		Receiver overrun interrupt enable bit.
		0	Interrupt is disabled. A receiver overrun interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receiver overrun interrupt generates a McASP receive interrupt (RINT).

4.22 Receiver Status Register (RSTAT)

The receiver status register (RSTAT) provides the receiver status and receive TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated. The RSTAT is shown in Figure 4-21 and described in Table 4-23.

Figure 4-21. Receiver Status Register (RSTAT) [Offset 80h]

31 9	8
Reserved[(A)]	RERR
R-0	R-0

7	6	5	4	3	2	1	0
RDMAERR	RSTAFRM	RDATA	RLAST	RTDMSLOT	RCKFAIL	RSYNCERR	ROVRN
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = bit is cleared by writing a 1, writing a 0 has no effect; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-23. Receiver Status Register (RSTAT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
8	RERR		RERR bit always returns a logic-OR of:ROVRN | RSYNCERR | RCKFAIL | RDMAERR Allows a single bit to be checked to determine if a receiver error interrupt has occurred.
		0	No errors have occurred.
		1	An error has occurred.
7	RDMAERR		Receive DMA error flag. RDMAERR is set when the CPU or DMA reads more serializers through the data port in a given time slot than were programmed as receivers. Causes a receive interrupt (RINT), if this bit is set and RDMAERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Receive DMA error did not occur.
		1	Receive DMA error did occur.
6	RSTAFRM		Receive start of frame flag. Causes a receive interrupt (RINT), if this bit is set and RSTAFRM in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	No new receive frame sync (AFSR) is detected.
		1	A new receive frame sync (AFSR) is detected.
5	RDATA		Receive data ready flag. Causes a receive interrupt (RINT), if this bit is set and RDATA in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	No new data in RBUF.
		1	Data is transferred from XRSR to RBUF and ready to be serviced by the CPU or DMA. When RDATA is set, it always causes a DMA event (AREVT).
4	RLAST		Receive last slot flag. RLAST is set along with RDATA, if the current slot is the last slot in a frame. Causes a receive interrupt (RINT), if this bit is set and RLAST in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Current slot is not the last slot in a frame.
		1	Current slot is the last slot in a frame. RDATA is also set.
3	RTDMSLOT		Returns the LSB of RSLOT. Allows a single read of RSTAT to determine whether the current TDM time slot is even or odd.
		0	Current TDM time slot is odd.
		1	Current TDM time slot is even.

Table 4-23. Receiver Status Register (RSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
2	RCKFAIL		Receive clock failure flag. RCKFAIL is set when the receive clock failure detection circuit reports an error (see Section 3.6.6). Causes a receive interrupt (RINT), if this bit is set and RCKFAIL in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Receive clock failure did not occur.
		1	Receive clock failure did occur.
1	RSYNCERR		Unexpected receive frame sync flag. RSYNCERR is set when a new receive frame sync (AFSR) occurs before it is expected. Causes a receive interrupt (RINT), if this bit is set and RSYNCERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Unexpected receive frame sync did not occur.
		1	Unexpected receive frame sync did occur.
0	ROVRN		Receiver overrun flag. ROVRN is set when the receive serializer is instructed to transfer data from XRSR to RBUF, but the former data in RBUF has not yet been read by the CPU or DMA. Causes a receive interrupt (RINT), if this bit is set and ROVRN in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Receiver overrun did not occur.
		1	Receiver overrun did occur.

4.23 Current Receive TDM Time Slot Registers (RSLOT)

The current receive TDM time slot register (RSLOT) indicates the current time slot for the receive data frame. The RSLOT is shown in Figure 4-22 and described in Table 4-24.

Figure 4-22. Current Receive TDM Time Slot Registers (RSLOT) [Offset 84h]

31	16
Reserved[(A)]	
R-0	

15	10	9	0
Reserved[(A)]		RSLOTCNT	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-24. Current Receive TDM Time Slot Registers (RSLOT) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
9-0	RSLOTCNT	0-17Fh	Current receive time slot count. Legal values: 0 to 383. TDM function is not supported for > 32 time slots. However, TDM time slot counter may count to 383 when used to receive a DIR block (transferred over TDM format).

4.24 Receive Clock Check Control Register (RCLKCHK)

The receive clock check control register (RCLKCHK) configures the receive clock failure detection circuit. The RCLKCHK is shown in Figure 4-23 and described in Table 4-25.

Figure 4-23. Receive Clock Check Control Register (RCLKCHK) [Offset 88h]

31 – 24	23 – 16
RCNT	RMAX
R-0	R/W-0

15 – 8	7 – 4	3 – 0
RMIN	Reserved[A]	RPS
R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-25. Receive Clock Check Control Register (RCLKCHK) Field Descriptions

Bit	Field	Value	Description
31-24	RCNT	0-FFh	Receive clock count value (from previous measurement). The clock circuit continually counts the number of DSP system clocks for every 32 receive high-frequency master clock (AHCLKR) signals, and stores the count in RCNT until the next measurement is taken.
23-16	RMAX	0-FFh	Receive clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If the current counter value is greater than RMAX after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.
15-8	RMIN	0-FFh	Receive clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If RCNT is less than RMIN after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.
7-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3-0	RPS	0-Fh	Receive clock check prescaler value.
		0	McASP system clock divided by 1
		1h	McASP system clock divided by 2
		2h	McASP system clock divided by 4
		3h	McASP system clock divided by 8
		4h	McASP system clock divided by 16
		5h	McASP system clock divided by 32
		6h	McASP system clock divided by 64
		7h	McASP system clock divided by 128
		8h	McASP system clock divided by 256
		9h-Fh	Reserved

4.25 Receiver DMA Event Control Register (REVTCTL)

The receiver DMA event control register (REVTCTL) is shown in Figure 4-24 and described in Table 4-26.

CAUTION

DSP specific registers

Accessing REVTCTL not implemented on a specific DSP may cause improper device operation.

Figure 4-24. Receiver DMA Event Control Register (REVTCTL) [Offset 8Ch]

31 – 16
Reserved[A]
R-0

15 – 1	0
Reserved[A]	RDATDMA
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-26. Receiver DMA Event Control Register (REVTCTL) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	RDATDMA		Receive data DMA request enable bit. If writing to this field, always write the default value of 0.
		0	Receive data DMA request is enabled.
		1	Reserved.

4.26 Transmitter Global Control Register (XGBLCTL)

Alias of the global control register (GBLCTL). Writing to the transmitter global control register (XGBLCTL) affects only the transmit bits of GBLCTL (bits 12-8). Reads from XGBLCTL return the value of GBLCTL. XGBLCTL allows the transmitter to be reset independently from the receiver. The XGBLCTL is shown in Figure 4-25 and described in Table 4-27. See Section 4.10 for a detailed description of GBLCTL.

Figure 4-25. Transmitter Global Control Register (XGBLCTL) [Offset A0h]

31–16
Reserved[A]
R-0

15–13	12	11	10	9	8
Reserved[A]	XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7–5	4	3	2	1	0
Reserved[A]	RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[A] If writing to this field, always write the default value for future device compatibility.

Table 4-27. Transmitter Global Control Register (XGBLCTL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0-FFh	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XFRST		Transmit frame sync generator reset enable bit. A write to this bit affects the XFRST bit of GBLCTL.
		0	Transmit frame sync generator is reset.
		1	Transmit frame sync generator is active.
11	XSMRST		Transmit state machine reset enable bit. A write to this bit affects the XSMRST bit of GBLCTL.
		0	Transmit state machine is held in reset.
		1	Transmit state machine is released from reset.
10	XSRCLR		Transmit serializer clear enable bit. A write to this bit affects the XSRCLR bit of GBLCTL.
		0	Transmit serializers are cleared.
		1	Transmit serializers are active.
9	XHCLKRST		Transmit high-frequency clock divider reset enable bit. A write to this bit affects the XHCLKRST bit of GBLCTL.
		0	Transmit high-frequency clock divider is held in reset.
		1	Transmit high-frequency clock divider is running.
8	XCLKRST		Transmit clock divider reset enable bit. A write to this bit affects the XCLKRST bit of GBLCTL.
		0	Transmit clock divider is held in reset.
		1	Transmit clock divider is running.
7-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	RFRST	x	Receive frame sync generator reset enable bit. A read of this bit returns the RFRST bit value of GBLCTL. Writes have no effect.
3	RSMRST	x	Receive state machine reset enable bit. A read of this bit returns the RSMRST bit value of GBLCTL. Writes have no effect.
2	RSRCLR	x	Receive serializer clear enable bit. A read of this bit returns the RSRSCLR bit value of GBLCTL. Writes have no effect.

Table 4-27. Transmitter Global Control Register (XGBLCTL) Field Descriptions (continued)

Bit	Field	Value	Description
1	RHCLKRST	x	Receive high-frequency clock divider reset enable bit. A read of this bit returns the RHCLKRST bit value of GBLCTL. Writes have no effect.
0	RCLKRST	x	Receive clock divider reset enable bit. A read of this bit returns the RCLKRST bit value of GBLCTL. Writes have no effect.

4.27 Transmit Format Unit Bit Mask Register (XMASK)

The transmit format unit bit mask register (XMASK) determines which bits of the transmitted data are masked off and padded with a known value before being shifted out the McASP. The XMASK is shown in Figure 4-26 and described in Table 4-28.

Figure 4-26. Transmit Format Unit Bit Mask Register (XMASK) [Offset A4h]

31	30	29	28	27	26	25	24
XMASK31	XMASK30	XMASK29	XMASK28	XMASK27	XMASK26	XMASK25	XMASK24
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	22	21	20	19	18	17	16
XMASK23	XMASK22	XMASK21	XMASK20	XMASK19	XMASK18	XMASK17	XMASK16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
XMASK15	XMASK14	XMASK13	XMASK12	XMASK11	XMASK10	XMASK9	XMASK8
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
XMASK7	XMASK6	XMASK5	XMASK4	XMASK3	XMASK2	XMASK1	XMASK0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 4-28. Transmit Format Unit Bit Mask Register (XMASK) Field Descriptions

Bit	Field	Value	Description
31-0	XMASK[31-0]		Transmit data mask enable bit.
		0	Corresponding bit of transmit data (before passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (XPAD and XPBIT bits in XFMT), which is transmitted out the McASP in place of the original bit.
		1	Corresponding bit of transmit data (before passing through reverse and rotate units) is transmitted out the McASP.

4.28 Transmit Bit Stream Format Register (XFMT)

The transmit bit stream format register (XFMT) configures the transmit data format. The XFMT is shown in Figure 4-27 and described in Table 4-29.

Figure 4-27. Transmit Bit Stream Format Register (XFMT) [Offset A4h]

31	18	17	16
Reserved[A]		XDATDLY	
R-0		R/W-0	

15	14	13	12	8	7	4	3	2	0
XRVRS	XPAD		XPBIT		XSSZ		XBUSEL	XROT	
R/W-0	R/W-0		R/W-0		R/W-0		R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-29. Transmit Bit Stream Format Register (XFMT) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
17-16	XDATDLY	0-3h	Transmit sync bit delay.
		0	0-bit delay. The first transmit data bit, AXR[n], occurs in same ACLKX cycle as the transmit frame sync (AFSX).
		1h	1-bit delay. The first transmit data bit, AXR[n], occurs one ACLKX cycle after the transmit frame sync (AFSX).
		2h	2-bit delay. The first transmit data bit, AXR[n], occurs two ACLKX cycles after the transmit frame sync (AFSX).
		3h	Reserved.
15	XRVRS		Transmit serial bitstream order.
		0	Bitstream is LSB first. No bit reversal is performed in transmit format bit reverse unit.
		1	Bitstream is MSB first. Bit reversal is performed in transmit format bit reverse unit.
14-13	XPAD	0-3h	Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0.
		0	Pad extra bits with 0.
		1h	Pad extra bits with 1.
		2h	Pad extra bits with one of the bits from the word as specified by XPBIT bits.
		3h	Reserved
12-8	XPBIT	0-1Fh	XPBIT value determines which bit (as written by the CPU or DMA to XBUF[n]) is used to pad the extra bits before shifting. This field only applies when XPAD = 2h.
		0	Pad with bit 0 value.
		1-1Fh	Pad with bit 1 to bit 31 value.

Table 4-29. Transmit Bit Stream Format Register (XFMT) Field Descriptions (continued)

Bit	Field	Value	Description
7-4	XSSZ	0-Fh	Transmit slot size.
		0-2h	Reserved
		3h	Slot size is 8 bits.
		4h	Reserved
		5h	Slot size is 12 bits.
		6h	Reserved.
		7h	Slot size is 16 bits.
		8h	Reserved.
		9h	Slot size is 20 bits.
		Ah	Reserved.
		Bh	Slot size is 24 bits.
		Ch	Reserved.
		Dh	Slot size is 28 bits.
		Eh	Reserved.
		Fh	Slot size is 32 bits.
3	XBUSEL		Selects whether writes to serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port.
		0	Writes to XRBUF[n] originate from the data port. Writes to XRBUF[n] from the configuration bus are ignored with no effect to the McASP.
		1	Writes to XRBUF[n] originate from the configuration bus. Writes to XRBUF[n] from the data port are ignored with no effect to the McASP.
2-0	XROT	0-7h	Right-rotation value for transmit rotate right format unit.
		0	Rotate right by 0 (no rotation).
		1h	Rotate right by 4 bit positions.
		2h	Rotate right by 8 bit positions.
		3h	Rotate right by 12 bit positions.
		4h	Rotate right by 16 bit positions.
		5h	Rotate right by 20 bit positions.
		6h	Rotate right by 24 bit positions.
		7h	Rotate right by 28 bit positions.

4.29 Transmit Frame Sync Control Register (AFSXCTL)

The transmit frame sync control register (AFSXCTL) configures the transmit frame sync (AFSX). The AFSXCTL is shown in Figure 4-28 and described in Table 4-30.

Figure 4-28. Transmit Frame Sync Control Register (AFSXCTL) [Offset ACh]

31	16
Reserved[A]	
R-0	

15	7	6	5	4	3	2	1	0
XMOD		Reserved[A]		FXWID	Reserved[A]		FSXM	FSXP
R/W-0		R-0		R/W-0	R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-30. Transmit Frame Sync Control Register (AFSXCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-7	XMOD	0-1FFh	Transmit frame sync mode select bits.
		0	Burst mode
		1h	Reserved
		2h-20h	2-slot TDM (I2S mode) to 32-slot TDM
		21h-17Fh	Reserved
		180h	384-slot DIT mode
		181h-1FFh	Reserved
6-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	FXWID		Transmit frame sync width select bit indicates the width of the transmit frame sync (AFSX) during its active period.
		0	Single bit
		1	Single word
3-2	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
1	FSXM		Transmit frame sync generation select bit.
		0	Externally-generated transmit frame sync
		1	Internally-generated transmit frame sync
0	FSXP		Transmit frame sync polarity select bit.
		0	A rising edge on transmit frame sync (AFSX) indicates the beginning of a frame.
		1	A falling edge on transmit frame sync (AFSX) indicates the beginning of a frame.

4.30 Transmit Clock Control Register (ACLKXCTL)

The transmit clock control register (ACLKXCTL) configures the transmit bit clock (ACLKX) and the transmit clock generator. The ACLKXCTL is shown in Figure 4-29 and described in Table 4-31.

Figure 4-29. Transmit Clock Control Register (ACLKXCTL) [Offset B0h]

31	16
Reserved[(A)]	
R-0	

15	8	7	6	5	4	0
Reserved[(A)]		CLKXP	ASYNC	CLKXM	CLKXDIV	
R-0		R/W-0	R/W-1	R/W-1	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-31. Transmit Clock Control Register (ACLKXCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	CLKXP		Transmit bitstream clock polarity select bit.
		0	Rising edge. External receiver samples data on the falling edge of the serial clock, so the transmitter must shift data out on the rising edge of the serial clock.
		1	Falling edge. External receiver samples data on the rising edge of the serial clock, so the transmitter must shift data out on the falling edge of the serial clock.
6	ASYNC		Transmit/receive operation asynchronous enable bit.
		0	Synchronous. Transmit clock and frame sync provides the source for both the transmit and receive sections.
		1	Asynchronous. Separate clock and frame sync used by transmit and receive sections.
5	CLKXM		Transmit bit clock source bit.
		0	External transmit clock source from ACLKX pin.
		1	Internal transmit clock source from output of programmable bit clock divider.
4-0	CLKXDIV	0-1Fh	Transmit bit clock divide ratio bits determine the divide-down ratio from AHCLKX to ACLKX.
		0	Divide-by-1
		1h	Divide-by-2
		2h-1Fh	Divide-by-3 to divide-by-32

4.31 Transmit High-Frequency Clock Control Register (AHCLKXCTL)

The transmit high-frequency clock control register (AHCLKXCTL) configures the transmit high-frequency master clock (AHCLKX) and the transmit clock generator. The AHCLKXCTL is shown in Figure 4-30 and described in Table 4-32.

Figure 4-30. Transmit High-Frequency Clock Control Register (AHCLKXCTL) [Offset B4h]

31	16
Reserved[A]	
R-0	

15	14	13 12	11 0
HCLKXM	HCLKXP	Reserved[A]	HCLKXDIV
R/W-1	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[A] If writing to this field, always write the default value for future device compatibility.

Table 4-32. Transmit High-Frequency Clock Control Register (AHCLKXCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15	HCLKXM		Transmit high-frequency clock source bit.
		0	External transmit high-frequency clock source from AHCLKX pin.
		1	Internal transmit high-frequency clock source from output of programmable high clock divider.
14	HCLKXP		Transmit bitstream high-frequency clock polarity select bit.
		0	Rising edge. AHCLKX is not inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin.
		1	Falling edge. AHCLKX is inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin.
13-12	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
11-0	HCLKXDIV	0-FFFh	Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKX.
		0	Divide-by-1
		1h	Divide-by-2
		2h-FFFh	Divide-by-3 to divide-by-4096

4.32 Transmit TDM Time Slot Register (XTDM)

The transmit TDM time slot register (XTDM) specifies in which TDM time slot the transmitter is active. TDM time slot counter range is extended to 384 slots (to support SPDIF blocks of 384 subframes). XTDM operates modulo 32, that is, XTDMS specifies the TDM activity for time slots 0, 32, 64, 96, 128, etc. The XTDM is shown in Figure 4-31 and described in Table 4-33.

Figure 4-31. Transmit TDM Time Slot Register (XTDM) [Offset B8h]

31	30	29	28	27	26	25	24
XTDMS31	XTDMS30	XTDMS29	XTDMS28	XTDMS27	XTDMS26	XTDMS25	XTDMS24
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	22	21	20	19	18	17	16
XTDMS23	XTDMS22	XTDMS21	XTDMS20	XTDMS19	XTDMS18	XTDMS17	XTDMS16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
XTDMS15	XTDMS14	XTDMS13	XTDMS12	XTDMS11	XTDMS10	XTDMS9	XTDMS8
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
XTDMS7	XTDMS6	XTDMS5	XTDMS4	XTDMS3	XTDMS2	XTDMS1	XTDMS0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 4-33. Transmit TDM Time Slot Register (XTDM) Field Descriptions

Bit	Field	Value	Description
31-0	XTDMS[31-0]		Transmitter mode during TDM time slot *n*.
		0	Transmit TDM time slot *n* is inactive. The transmit serializer does not shift out data during this slot.
		1	Transmit TDM time slot *n* is active. The transmit serializer shifts out data during this slot according to the serializer control register (SRCTL).

4.33 Transmitter Interrupt Control Register (XINTCTL)

The transmitter interrupt control register (XINTCTL) controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT. The XINTCTL is shown in Figure 4-32 and described in Table 4-34. See Section 4.34 for a description of the interrupt conditions.

Figure 4-32. Transmitter Interrupt Control Register (XINTCTL) [Offset BCh]

31–8
Reserved[(A)]
R-0

7	6	5	4	3	2	1	0
XSTAFRM	Reserved[(A)]	XDATA	XLAST	XDMAERR	XCKFAIL	XSYNCERR	XUNDRN
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-34. Transmitter Interrupt Control Register (XINTCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	XSTAFRM		Transmit start of frame interrupt enable bit.
		0	Interrupt is disabled. A transmit start of frame interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit start of frame interrupt generates a McASP transmit interrupt (XINT).
6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	XDATA		Transmit data ready interrupt enable bit.
		0	Interrupt is disabled. A transmit data ready interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit data ready interrupt generates a McASP transmit interrupt (XINT).
4	XLAST		Transmit last slot interrupt enable bit.
		0	Interrupt is disabled. A transmit last slot interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit last slot interrupt generates a McASP transmit interrupt (XINT).
3	XDMAERR		Transmit DMA error interrupt enable bit.
		0	Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT).
2	XCKFAIL		Transmit clock failure interrupt enable bit.
		0	Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT).
1	XSYNCERR		Unexpected transmit frame sync interrupt enable bit.
		0	Interrupt is disabled. An unexpected transmit frame sync interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. An unexpected transmit frame sync interrupt generates a McASP transmit interrupt (XINT).
0	XUNDRN		Transmitter underrun interrupt enable bit.
		0	Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT).

4.34 Transmitter Status Register (XSTAT)

The transmitter status register (XSTAT) provides the transmitter status and transmit TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated. The XSTAT is shown in Figure 4-33 and described in Table 4-35.

Figure 4-33. Transmitter Status Register (XSTAT) [Offset C0h]

31 9	8
Reserved[(A)]	XERR
R-0	R-0

7	6	5	4	3	2	1	0
XDMAERR	XSTAFRM	XDATA	XLAST	XTDMSLOT	XCKFAIL	XSYNCERR	XUNDRN
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = bit is cleared by writing a 1, writing a 0 has no effect; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-35. Transmitter Status Register (XSTAT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
8	XERR		XERR bit always returns a logic-OR of:XUNDRN | XSYNCERR | XCKFAIL | XDMAERR Allows a single bit to be checked to determine if a transmitter error interrupt has occurred.
		0	No errors have occurred.
		1	An error has occurred.
7	XDMAERR		Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more serializers through the data port in a given time slot than were programmed as transmitters. Causes a transmit interrupt (XINT), if this bit is set and XDMAERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Transmit DMA error did not occur.
		1	Transmit DMA error did occur.
6	XSTAFRM		Transmit start of frame flag. Causes a transmit interrupt (XINT), if this bit is set and XSTAFRM in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	No new transmit frame sync (AFSX) is detected.
		1	A new transmit frame sync (AFSX) is detected.
5	XDATA		Transmit data ready flag. Causes a transmit interrupt (XINT), if this bit is set and XDATA in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	XBUF is written and is full.
		1	Data is copied from XBUF to XRSR. XBUF is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT).
4	XLAST		Transmit last slot flag. XLAST is set along with XDATA, if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT), if this bit is set and XLAST in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Current slot is not the last slot in a frame.
		1	Current slot is the last slot in a frame. XDATA is also set.
3	XTDMSLOT		Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd.
		0	Current TDM time slot is odd.
		1	Current TDM time slot is even.

Table 4-35. Transmitter Status Register (XSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
2	XCKFAIL		Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error (see Section 3.6.6). Causes a transmit interrupt (XINT), if this bit is set and XCKFAIL in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Transmit clock failure did not occur.
		1	Transmit clock failure did occur.
1	XSYNCERR		Unexpected transmit frame sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT), if this bit is set and XSYNCERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Unexpected transmit frame sync did not occur.
		1	Unexpected transmit frame sync did occur.
0	XUNDRN		Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF to XRSR, but XBUF has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT), if this bit is set and XUNDRN in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Transmitter underrun did not occur.
		1	Transmitter underrun did occur. See Section 3.6.2 for details on McASP action upon underrun conditions.

4.35 Current Transmit TDM Time Slot Register (XSLOT)

The current transmit TDM time slot register (XSLOT) indicates the current time slot for the transmit data frame. The XSLOT is shown in Figure 4-34 and described in Table 4-36.

Figure 4-34. Current Transmit TDM Time Slot Register (XSLOT) [Offset C4h]

31	16
Reserved[(A)]	
R-0	

15	10	9	0
Reserved[(A)]		XSLOTCNT	
R-0		R-17Fh	

LEGEND: R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-36. Current Transmit TDM Time Slot Register (XSLOT) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
9-0	XSLOTCNT	0-17Fh	Current transmit time slot count. Legal values: 0 to 383. During reset, this counter value is 383 so the next count value, which is used to encode the first DIT group of data, will be 0 and encodes the B preamble. TDM function is not supported for >32 time slots. However, TDM time slot counter may count to 383 when used to transmit a DIT block.

4.36 Transmit Clock Check Control Register (XCLKCHK)

The transmit clock check control register (XCLKCHK) configures the transmit clock failure detection circuit. The XCLKCHK is shown in Figure 4-35 and described in Table 4-37.

Figure 4-35. Transmit Clock Check Control Register (XCLKCHK) [Offset C8h]

31 — 24	23 — 16
XCNT	XMAX
R-0	R/W-0

15 — 8	7	6 — 4	3 — 0
XMIN	XCKFAILSW	Reserved[A]	XPS
R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-37. Transmit Clock Check Control Register (XCLKCHK) Field Descriptions

Bit	Field	Value	Description
31-24	XCNT	0	Transmit clock count value (from previous measurement). The clock circuit continually counts the number of DSP system clocks for every 32 transmit high-frequency master clock (AHCLKX) signals, and stores the count in XCNT until the next measurement is taken.
23-16	XMAX	0-FFh	Transmit clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.
15-8	XMIN	0-FFh	Transmit clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.
7	XCKFAILSW		Transmit clock failure detect autoswitch enable bit.
		0	Transmit clock failure detect autoswitch is disabled.
		1	Transmit clock failure detect autoswitch is enabled.
6-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3-0	XPS	0-Fh	Transmit clock check prescaler value.
		0	McASP system clock divided by 1
		1h	McASP system clock divided by 2
		2h	McASP system clock divided by 4
		3h	McASP system clock divided by 8
		4h	McASP system clock divided by 16
		5h	McASP system clock divided by 32
		6h	McASP system clock divided by 64
		7h	McASP system clock divided by 128
		8h	McASP system clock divided by 256
		9h-Fh	Reserved

4.37 Transmitter DMA Event Control Register (XEVTCTL)

The transmitter DMA event control register (XEVTCTL) is shown in Figure 4-36 and described in Table 4-38.

CAUTION

DSP specific registers

Accessing XEVTCTL not implemented on a specific DSP may cause improper device operation.

Figure 4-36. Transmitter DMA Event Control Register (XEVTCTL) [Offset CCh]

31 – 16
Reserved[A]
R-0

15 – 1	0
Reserved[A]	XDATDMA
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-38. Transmitter DMA Event Control Register (XEVTCTL) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	XDATDMA		Transmit data DMA request enable bit. If writing to this field, always write the default value of 0.
		0	Transmit data DMA request is enabled.
		1	Reserved.

4.38 Serializer Control Registers (SRCTL*n*)

Each serializer on the McASP has a serializer control register (SRCTL). There are up to 16 serializers per McASP. The SRCTL is shown in Figure 4-37 and described in Table 4-39.

> **CAUTION**
>
> **DSP specific registers**
>
> Accessing SRCTL*n* not implemented on a specific DSP may cause improper device operation.

Figure 4-37. Serializer Control Registers (SRCTL*n*) [Offset 180h-1BCh]

31-16
Reserved[A]
R-0

15-6	5	4	3-2	1-0
Reserved[A]	RRDY	XRDY	DISMOD	SRMOD
R-0	R-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A If writing to this field, always write the default value for future device compatibility.

Table 4-39. Serializer Control Registers (SRCTL*n*) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	RRDY		Receive buffer ready bit. RRDY indicates the current receive buffer state. Always reads 0 when programmed as a transmitter or as inactive. If SRMOD bit is set to receive (2h), RRDY switches from 0 to 1 whenever data is transferred from XRSR to RBUF.
		0	Receive buffer (RBUF) is empty.
		1	Receive buffer (RBUF) contains data and needs to be read before the start of the next time slot or a receiver overrun occurs.
4	XRDY		Transmit buffer ready bit. XRDY indicates the current transmit buffer state. Always reads 0 when programmed as a receiver or as inactive. If SRMOD bit is set to transmit (1h), XRDY switches from 0 to 1 when XSRCLR in GBLCTL is switched from 0 to 1 to indicate an empty transmitter. XRDY remains set until XSRCLR is forced to 0, data is written to the corresponding transmit buffer, or SRMOD bit is changed to receive (2h) or inactive (0).
		0	Transmit buffer (XBUF) contains data.
		1	Transmit buffer (XBUF) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs.
3-2	DISMOD	0-3h	Serializer pin drive mode bit. Drive on pin when in inactive TDM slot of transmit mode or when serializer is inactive. This field only applies if the pin is configured as a McASP pin (PFUNC = 0).
		0	Drive on pin is 3-state.
		1h	Reserved
		2h	Drive on pin is logic low.
		3h	Drive on pin is logic high.
1-0	SRMOD	0-3h	Serializer mode bit.
		0	Serializer is inactive.
		1h	Serializer is transmitter.
		2h	Serializer is receiver.
		3h	Reserved

4.39 DIT Left Channel Status Registers (DITCSRA0-DITCSRA5)

The DIT left channel status registers (DITCSRA) provide the status of each left channel (even TDM time slot). Each of the six 32-bit registers (Figure 4-38) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

Figure 4-38. DIT Left Channel Status Registers (DITCSRA0-DITCSRA5) [Offset 100h-114h]

31	16
DITCSRA*n*	
R/W-0	

15	0
DITCSRA*n*	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

4.40 DIT Right Channel Status Registers (DITCSRB0-DITCSRB5)

The DIT right channel status registers (DITCSRB) provide the status of each right channel (odd TDM time slot). Each of the six 32-bit registers (Figure 4-39) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

Figure 4-39. DIT Right Channel Status Registers (DITCSRB0-DITCSRB5) [Offset 118h-12Ch]

31	16
DITCSRB*n*	
R/W-0	

15	0
DITCSRB*n*	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

4.41 DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5)

The DIT left channel user data registers (DITUDRA) provides the user data of each left channel (even TDM time slot). Each of the six 32-bit registers (Figure 4-40) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

Figure 4-40. DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5) [Offset 130h-144h]

31	16
DITUDRA*n*	
R/W-0	

15	0
DITUDRA*n*	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

4.42 DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5)

The DIT right channel user data registers (DITUDRB) provides the user data of each right channel (odd TDM time slot). Each of the six 32-bit registers (Figure 4-41) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

Figure 4-41. DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5) [Offset 148h-15Ch]

31	16
DITUDRB*n*	
R/W-0	

15	0
DITUDRB*n*	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

4.43 Transmit Buffer Registers (XBUF*n*)

The transmit buffers for the serializers (XBUF) hold data from the transmit format unit. For transmit operations, the XBUF (Figure 4-42) is an alias of the XRBUF in the serializer.

CAUTION

DSP specific registers

Accessing XBUF registers not implemented on a specific DSP may cause improper device operation.

Figure 4-42. Transmit Buffer Registers (XBUFn) [Offset 200h-21Ch]

31	16
XBUF*n*	
R/W-0	

15	0
XBUF*n*	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

4.44 Receive Buffer Registers (RBUF*n*)

The receive buffers for the serializers (RBUF) hold data from the serializer before the data goes to the receive format unit. For receive operations, the RBUF (Figure 4-43) is an alias of the XRBUF in the serializer.

CAUTION

DSP specific registers

Accessing XBUF registers not implemented on a specific DSP may cause improper device operation.

Figure 4-43. Receive Buffer Registers (RBUF*n*) [Offset 280h-2BCh]

31	16
RBUF*n*	
R/W-0	

15	0
RBUF*n*	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Appendix A

SPRU041J–November 2008

EDMA Examples

This appendix shows example EDMA implementations to service the McASP. See Section 3.3.5 for a general description of McASP servicing.

Topic **Page**

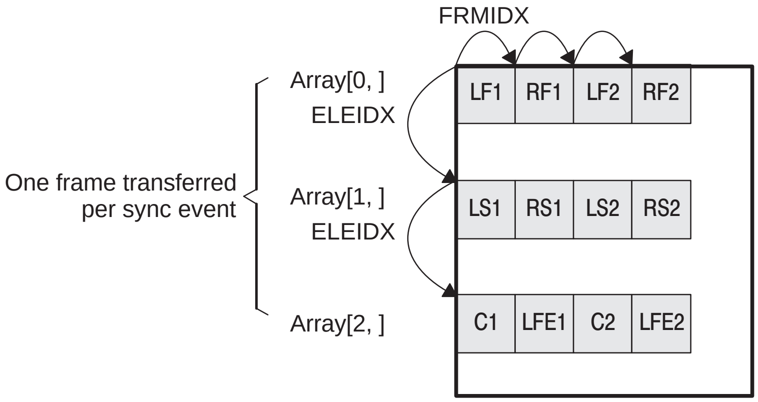
A.1 EDMA Implementation Guidelines

Each of the six events: AXEVT, AXEVTO, AXEVTE, AREVT, AREVTO, and AREVTE (for each McASP), can be configured to any EDMA channel by use of the EDMA selector control register (see the device-specific data manual for details).

A.2 EDMA Implementation of Scenario 1

As discussed in Section 3.3.5 and Figure 3-5, the EDMA can service the McASP upon events AXEVT and AREVT (Figure 3-5, Scenario 1), or upon events AXEVTO, AREVTO, AXEVTE, and AREVTE (Figure 3-5, Scenario 2). Figure A-1 shows an example implementation of scenario 1 using the EDMA.

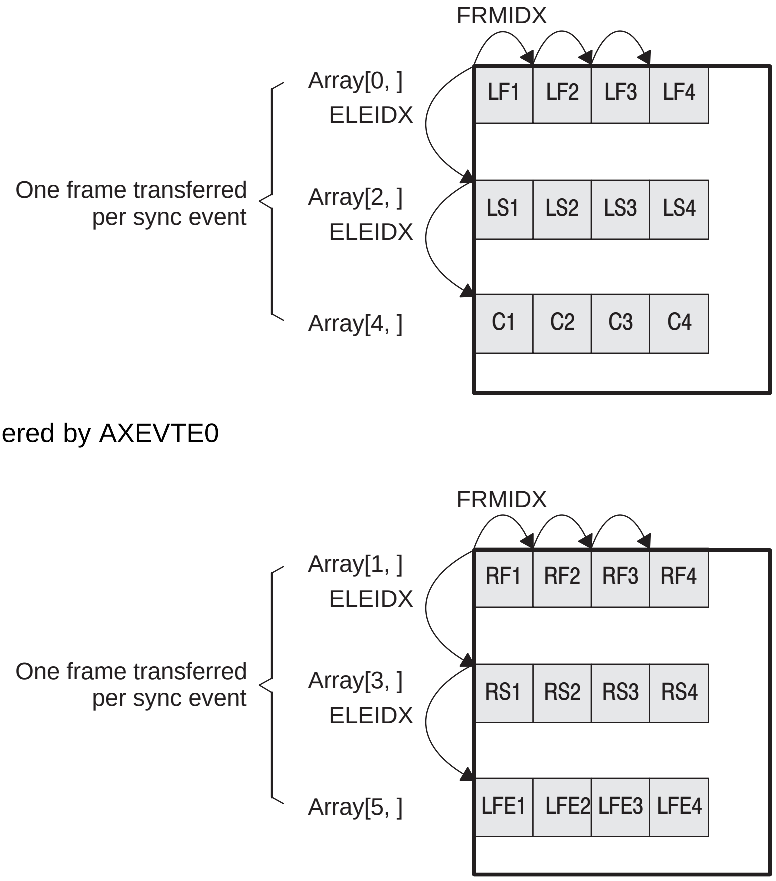
Figure A-1. EDMA Event Triggered on Each Time Slot (AXEVT/AREVT)



A.3 EDMA Implementation of Scenario 2

As discussed in Section 3.3.5 and Figure 3-5, the EDMA can service the McASP upon events AXEVT and AREVT (Figure 3-5, Scenario 1), or upon events AXEVTO, AREVTO, AXEVTE, and AREVTE (Figure 3-5, Scenario 2). Figure A-2 shows an example implementation of scenario 2 using the EDMA.

Figure A-2. Two Alternating EDMA Events Triggered for Each Time Slot



(a) Channel Triggered by AXEVTE0

(b) Channel Triggered by AXEVTO0

Register Bit Restrictions

Some bit fields (see Table B-1) have restrictions on when they may be changed. These restrictions take the form of certain registers that must be asserted in GBLCTL. Once these registers have been asserted, the user may then, and only then, change the desired bit field.

Table B-1. Bits With Restrictions on When They May be Changed

To Change ... Register	To Change ... Bit Field	... these registers must be asserted in GBLCTL									
		HCLKRRST	RGRST	RSRCLR	RSMRST	RFRST	HCLKXRST	XGRST	XSRCLR	XSMRST	XFRST
DITCTL	DITEN									x	x
XFMT	XSSZ									x	
XFMT	XDATDLY				x					x	
RFMT	RSSZ				x						
RFMT	RDATDLY				x						
AFSXCTL	FSXP									x	x
AFSXCTL	FSXM									x	x
AFSXCTL	FXWID									x	x
AFSXCTL	XMOD									x	x
AFSRCTL	FSRP				x	x					
AFSRCTL	FSRM				x	x					
AFSRCTL	FRWID				x	x					
AFSRCTL	RMOD				x	x					
ACLKXCTL	CLKXDIV							x	x	x	x
ACLKXCTL	CLKXM								x	x	x
ACLKXCTL	ASYNC				x	x					
ACLKXCTL	CLKXP								x	x	x
ACLKRCTL	CLKRDIV		x	x	x	x					
ACLKRCTL	CLKRM			x	x	x					
ACLKRCTL	CLKRP			x	x	x					
AHCLKXCTL	HCLKXDIV						x	x	x	x	x
AHCLKXCTL	HCLKXP						x	x	x	x	x
AHCLKXCTL	HCLKXM						x	x	x	x	x
AHCLKRCTL	HCLKRDIV	x	x	x	x	x					
AHCLKRCTL	HCLKRP	x	x	x	x	x					
AHCLKRCTL	HCLKRM	x	x	x	x	x					
DLBCTL	DLBEN			x	x	x			x	x	x
DLBCTL	ORD			x	x	x			x	x	x
DLBCTL	MODE			x	x	x			x	x	x

Revision History

Table C-1 lists the changes made since the previous version of this document.

Table C-1. Document Revision History

Reference	Additions/Modifications/Deletions
Figure 1-9	Changed figure.
Section 4.25	Changed paragraph.
Table 4-26	Changed Description of RDATDMA bit 0.
Section 4.37	Changed paragraph.
Table 4-38	Changed Description of XDATDMA bit 0.

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**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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