Application Note

AM62x Audio System Design Guide

ABSTRACT

Texas Instruments Audio DSP SoCs for premium audio applications feature the multichannel audio serial port (McASP). The McASP is very flexible in its configuration options to enable a variety of multi-zone and multichannel audio systems. This document provides an overview of digital audio formats, McASP configuration options, and common system implementations.

How to Use this Document

This document is meant to provide a base level understanding of audio data transmission, TI audio peripherals, and how to use these peripherals for a multizone audio system. The document explains the fundamentals for audio system design, starting with detailing how digital audio is sent or received and applying that to the McASP configuration to enable a variety of different system use cases.

Chapter Overview	Chapter link
Basic explanations of digital audio formats supported by the McASP	Section 1
Basic overview of the McASP peripheral and the various configuration options available	Section 2
Detailed explanations of the AM62x-specific implementation of McASP instances for different use cases	Section 3
Highlights the two biggest considerations for layout design of the McASP	Section 4
Examples of different configurations of the McASP for practical use cases involving external components	Section 5
Key audio system design takeaways	Section 6

Table of Contents

Trademarks

1 Digital Audio Formats

Digital audio data is communicated through 3-wire formats. The three signals required for audio transmission are bit clock, frame sync, and serial audio data. Multiple audio slots are contained within a frame of data. Audio channels are assigned to unique slots to send multichannel audio over a single bus. A single audio data frame contains a single samples for each channel being transferred. The various digital audio formats will define how the frame of audio data is organized and transferred between devices. In general, all audio frame formats are defined by the following characteristics:

- Falling or rising edge of frame sync indicates frame start
- Delay between frame sync edge and data transmission
- Frame sync width
- Number of unique audio channels per frame
- Slot size in bits per channel
- Word depth in bits per slot
- Bitstream order (MSB or LSB first)
- Bit clock polarity for sampling audio data

Note

There are many generic names for the signals of digital audio. Frame sync (FS) may also be referred to as LRCLK, WCLK, or Word Select. Bit clock may also be referred to as serial clock or audio clock. In this document, the standard naming of frame sync and bit clock is used.

Slots of audio may have words that are less than the total slot bit width. When the word depth is smaller than the slot size, then the word can be aligned to the left of the slot or the right of the slot. Because of different alignment options, it is important to have a complete understanding of how the bits are being padded to interpret the audio data.

Figure 1-1 shows an example audio slot that has a most significant bit first serial bitstream, 24-bit word depth (left-aligned), and 32-bit slot size.

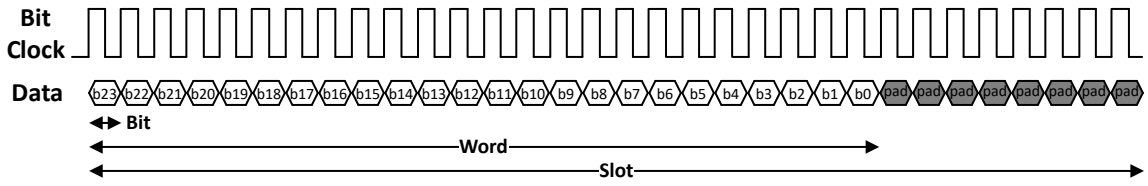

Figure 1-1. Bit, Word, and Slot in a Frame

1.1 I²S

Inter IC Sound (I²S) is a standard digital audio protocol specifically defined for stereo audio. Stereo audio means that each digital audio frame consists of two channels: left and right. An I²S frame is defined by the following characteristics:

- Falling edge of frame sync indicates frame start
- 1-bit clock cycle delay between frame sync falling edge and data transmission
- Single word frame sync width
- Two channels per frame
- Most significant bit first serial bitstream order
- Data shifted out on falling edge of bit clock
- Data sampled in on rising edge of bit clock

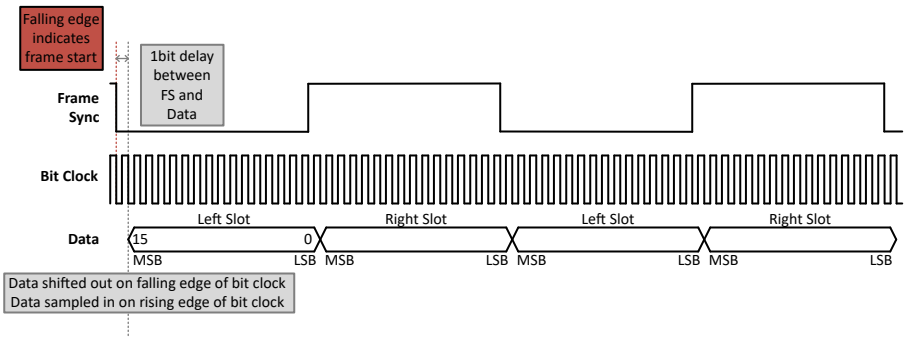

Figure 1-2. I2S Timing Diagram

1.2 TDM

Time Division Multiplexing (TDM) is a standard digital audio protocol for multichannel audio transmission. Typically, TDM is followed by a number that indicates the channel count per audio frame, such as TDM4. TDM does not have a standardized format, but a typical TDM frame is defined by the following characteristics:

- Rising edge of frame sync indicates frame start
- 1-bit clock cycle delay between frame sync rising edge and data transmission
- Single bit frame sync width
- Most significant bit first serial bitstream order
- Data shifted out on falling edge of bit clock
- Data sampled in on rising edge of bit clock

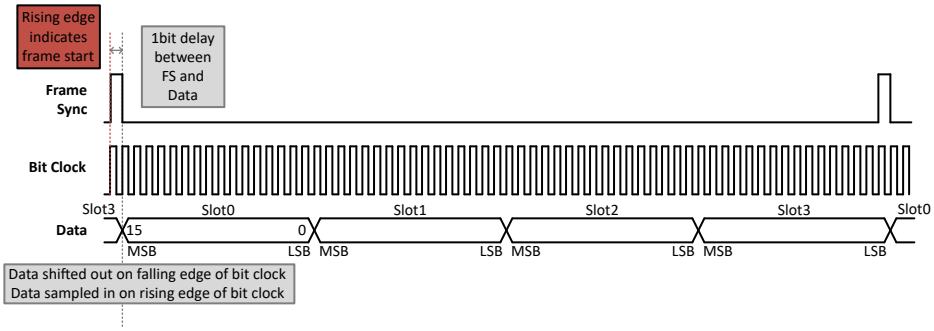

Figure 1-3. TDM4 Timing Diagram

2 McASP Overview

The Multichannel Audio Serial Port (McASP) was designed to optimize multichannel and multi-zone audio communication. The McASP peripheral consists of signals for transmit and receive bit clocks (ACLK[X/R]), transmit and receive frame syncs (AFS[X/R]), and up to sixteen audio transmit/receive serializers (AXR). The McASP also has internal paths for sourcing a root clock and programmable dividers that can be used to generate the appropriate frequencies of bit clock and frame sync. The McASP has one AUXCLK that can be used to generate the internal transmit and receive high frequency clocks (AHCLK[X/R]). The high frequency clocks are used to internally generate the bit clock and frame sync. Alternatively, the McASP can be configured to receive a bit clock and/or frame sync from an external source.

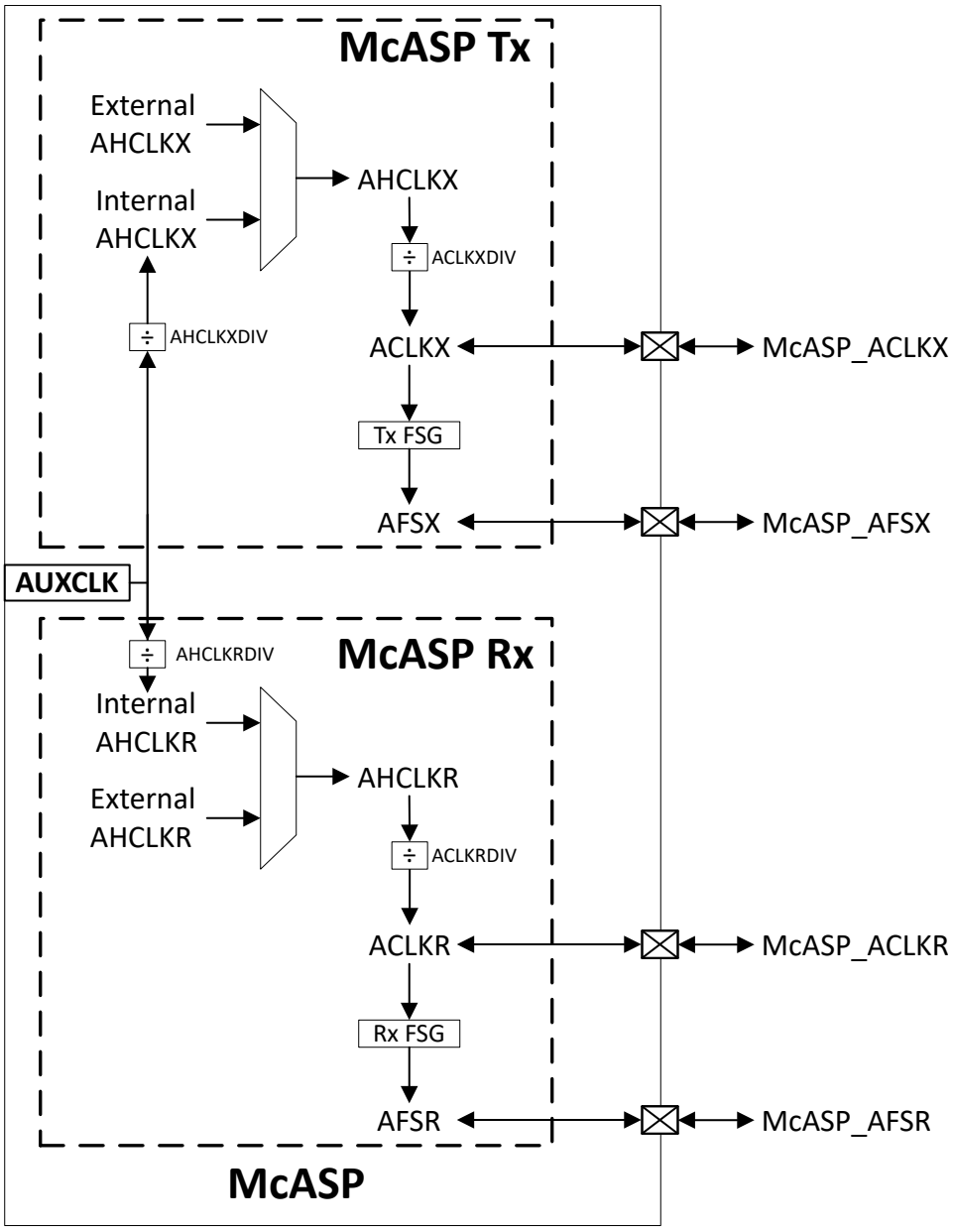

Figure 2-1. McASP General Overview

Each McASP supports the following features:

- Audio data transmit and reception with independent clock zones
- Up to 16 serializers for audio transmit and receive (AXR)
  - Serializer count per McASP instance varies depending on the SoC implementation.
  - For example, on AM62x, McASP2 has all 16 serializers available while McASP0 only has four serializers available.
- 32-bit buffer per serializer for transmit and receive operations
- Clock loss detection
- Configuration options for audio frame format
  - Slot count
  - Slot size in bits
  - Bit masking for active word depth less than slot size
  - Active slot masking
  - Frame sync to data delay in terms of bit clock cycles
  - Frame sync polarity and width
  - Bit clock polarity
  - Serial data bitstream order

Each McASP also can operate in sync mode where the ACLKX and AFSX are internally routed to the ACLKR and AFSR. Sync mode enables use cases such as a codec that has one clock domain for data transmission and reception.

If the McASP is in asynchronous mode, then the serializer IO direction determines which bit clock and frame sync is used for interpreting the audio data frame.

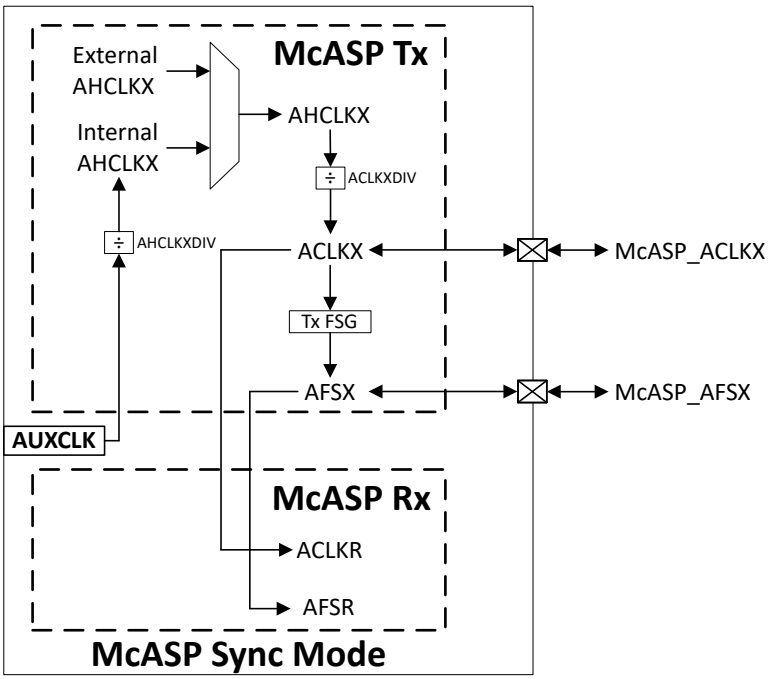

Figure 2-2. McASP Sync Mode

3 McASP Connections for AM62x Devices

The AM62x is a family of Arm-based Processors that features three unique McASPs for audio applications. The entire mapping of internal connections for the McASP on AM62x devices is shown in Figure 3-1.

Note

Each McASP has an independent ACLK and AFS for the transmit and receive clock zone but only one AUXCLK.

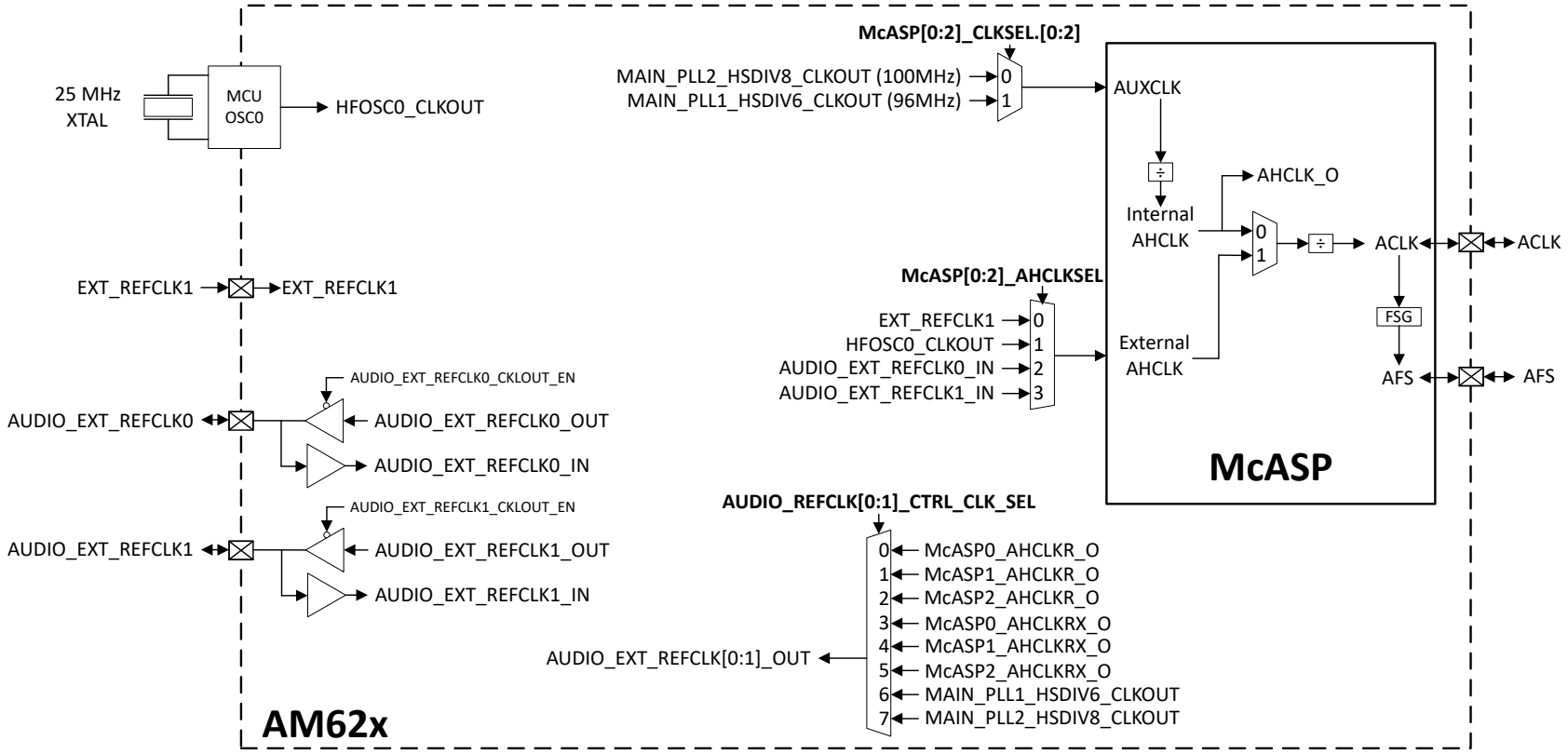

Figure 3-1. AM62x McASP Connections

3.1 McASP Common Configurations

The McASP bit clock (ACLK) and frame sync (AFS) are both bidirectional such that the McASP can either be the clock controller or clock peripheral. The following sections detail all the available options for each clocking configuration.

Table 3-1 lists the most common use cases for configuring McASP. The AM62x SoCs have many options for generating, sourcing, and receiving clocks for the audio data frame formatting.

Note

For bit clock and frame sync, **internally generated** refers to internally referenced signals that are output at the SoC level for McASP clock controller applications while **externally generated** means that the signals are configured as inputs at the SoC level for McASP clock peripheral applications.

Table 3-1. McASP Use Case Matrix

Description	AHCLK	Bit Clock	Frame Sync	McASP	Example
McASP clock controller with internal audio PLL reference	AM62x Devices do not feature an internal audio PLL reference for generating audio bit clock and frame sync frequencies.				
McASP clock controller with external AUXCLK reference	AM62x Devices do not feature an option to route an external clock source to the McASP AUXCLK				
McASP clock controller with external AHCLK reference	Externally Generated	Internally Generated	Internally Generated	Section 3.1.1.1	Figure 5-1
McASP clock peripheral		Externally Generated	Externally Generated	Section 3.1.2	Figure 5-2

Figure 3-2 shows a more detailed view of the available options.

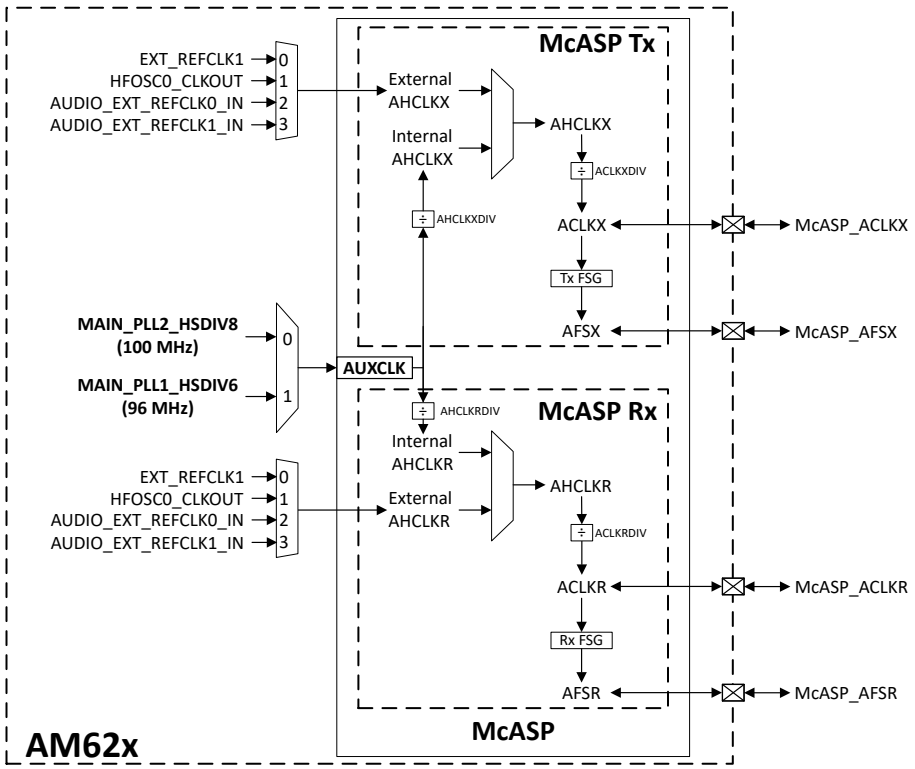

Figure 3-2. McASP Detailed Overview

3.1.1 McASP as a Clock Controller

If the McASP is configured as a clock controller, then the bit clock and frame sync signals are configured as outputs. The SDK driver defines bit clock and frame sync as outputs when the source is set to **Internally Generated**. This means that the bit clock is internally generated from the high clock and that the frame sync is generated based on the bit clock. The high clock of a TX or RX domain feature many options to best fit audio system requirements.

The AUXCLK is a single clock reference that can be sourced to both the TX and RX domains. The AM62x devices have the McASP AUXCLK input tied to the internal PLL outputs for 100 MHz and 96 MHz.

Note

Because the AM62x AUXCLK input does not feature an audio frequency input, the AUXCLK (and internally generated AHCLK) should not be used on AM62x devices.

When the AHCLK is internally generated, then the AHCLK can be routed as an output on any of the AUDIO_EXT_REFCLK pins for a high-frequency reference.

Note

The AHCLK must be configured to reference AUXCLK to internally generate the AHCLK in order to use AHCLK as an output on AUDIO_EXT_REFCLK. Therefore, AHCLK as an output on AUDIO_EXT_REFCLK is not an option on AM62x devices for most applications.

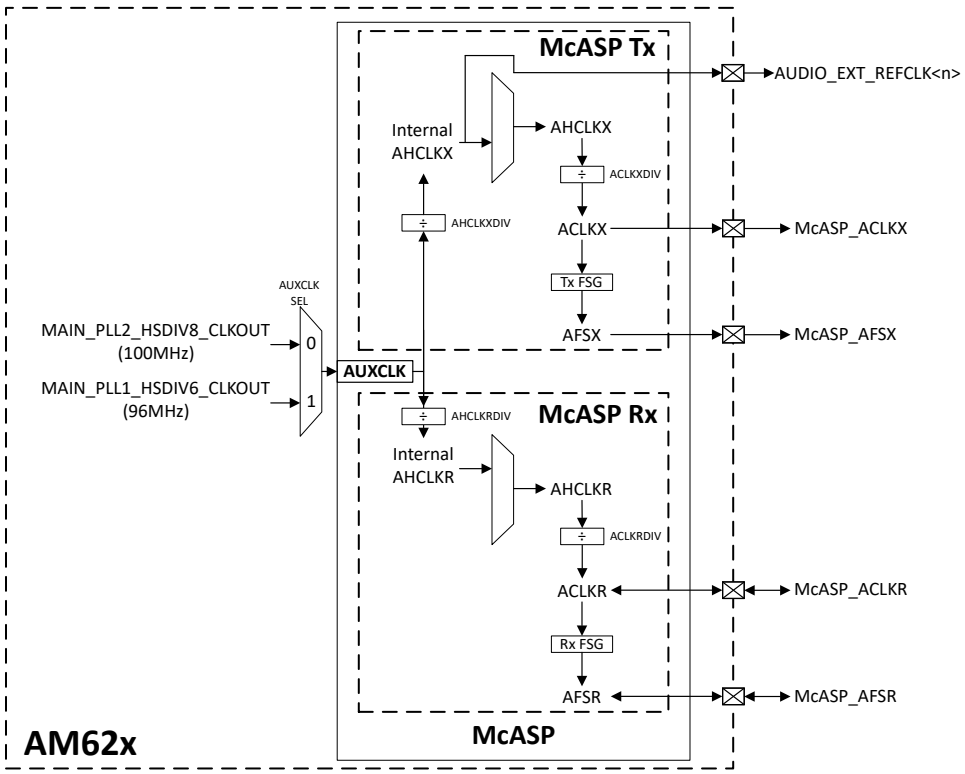

Figure 3-3. McASP Controller with AUXCLK Source

3.1.1.1 Clocks Generated using the AUDIO_EXT_REFCLK AHCLK Source

The following section details an example setup for a McASP where the bit clock and frame sync are configured as outputs and generated using an external source via AUDIO_EXT_REFCLK as a clock reference directly to the AHCLK.

Description	AHCLK	Bit Clock	Frame Sync
McASP clock controller with external AHCLK input reference	Externally Generated	Internally Generated	Internally Generated

In this example, the McASP is being configured for a 48 kHz frame sync and TDM8 frame format with 32 bit words, resulting in a bit clock frequency of 12.288 MHz. The AUXCLK is not considered when AHCLK is configured to be externally generated. Each AHCLK has a unique mux to select different external sources. The AHCLK mux is configured to point to the AUDIO_EXT_REFCLK0 source that is 24.576 MHz from an external driver. The SDK driver sets the ACLK divider based upon the number of slots, frame sync frequency, and ratio between frame sync and AHCLK.

When the AHCLK is **externally generated** then the AHCLK **can't** be output on the AUDIO_EXT_REFCLK.

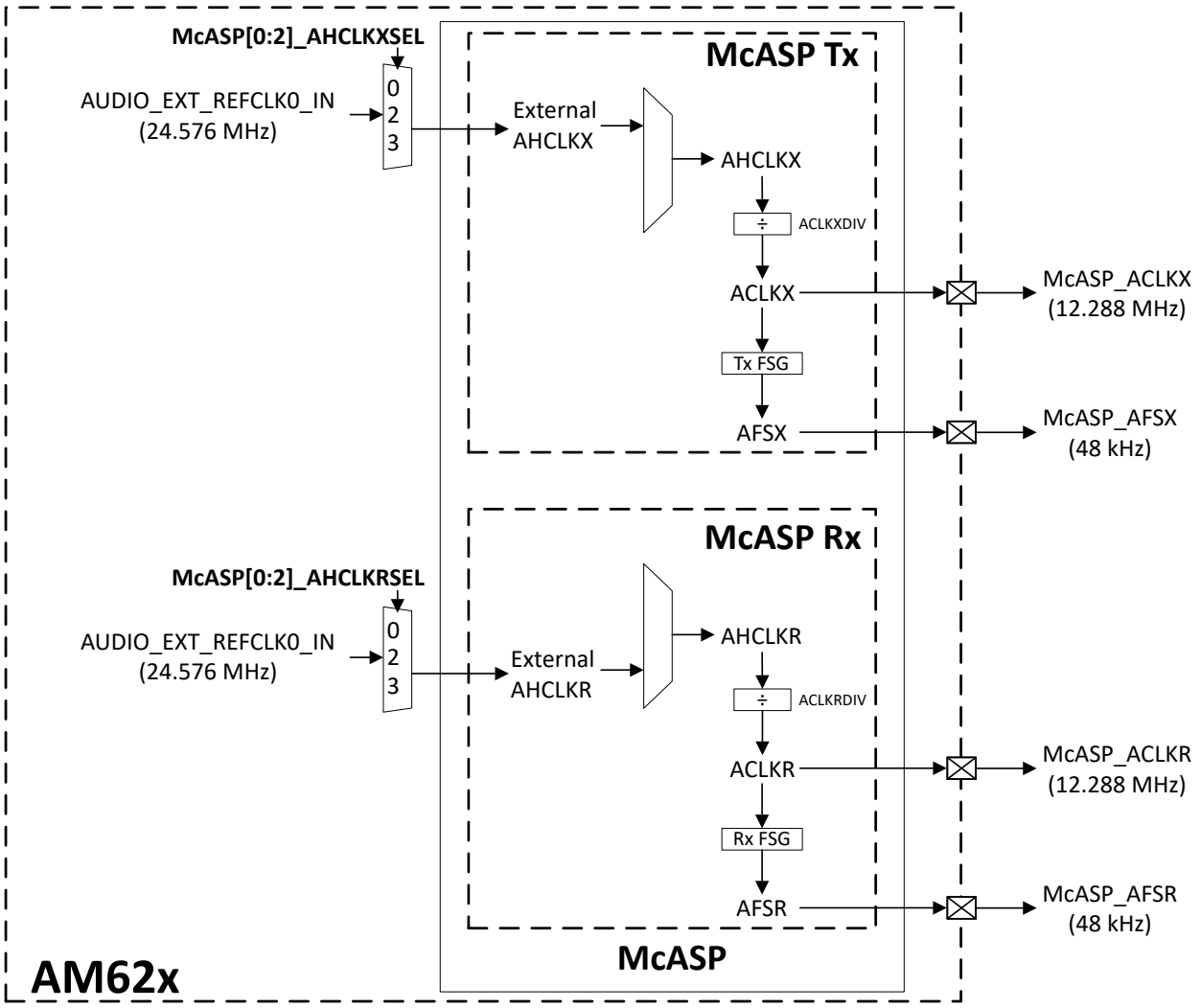

Figure 3-4. McASP Clock Controller with AUDIO_EXT_REFCLK0 AHCLK Reference

3.1.2 McASP as Clock Peripheral

The following section details an example setup for a McASP where the bit clock and frame sync are configured as inputs.

Description	AHCLK	Bit Clock	Frame Sync
McASP clock peripheral		Externally Generated	Externally Generated

In this example, the McASP is being configured for a 48 kHz frame sync and TDM8 frame format with 32-bit words, resulting in a bit clock frequency of 12.288 MHz. The AHCLK settings do not matter in this case. The SDK driver must be configured to represent the expected frequency for bit clock and frame sync for proper audio data transmission.

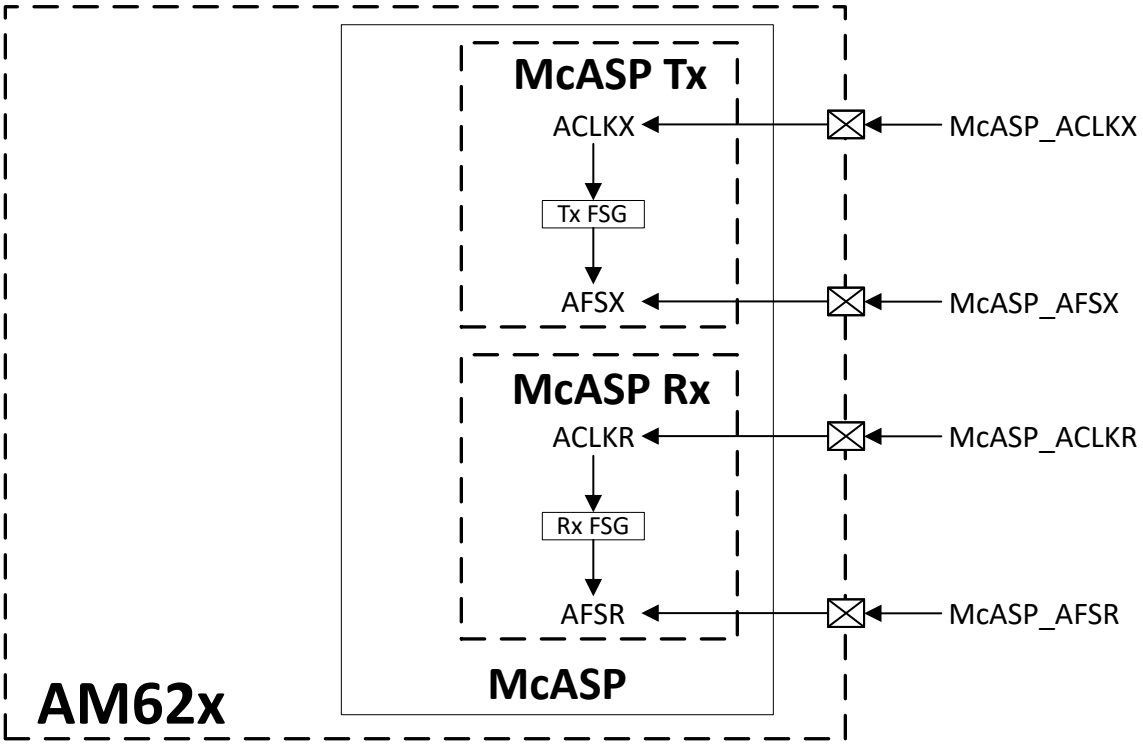

Figure 3-5. McASP Clock Peripheral

4 McASP Layout Considerations

The McASP is designed to be able to interface with multiple audio devices simultaneously using a single clock domain. However, the signal integrity of the clock and data signals may be impacted depending on the layout implementation. This chapter highlights the two most important layout considerations for the McASP on the AM62x devices.

Note

Regardless of layout implementation parameters, the McASP signal layout should always be simulated to ensure the clock and data signals meet the datasheet timing requirements.

4.1 McASP Signals Shared with Bootmode Logic

The AM62x has 16 bootmode signals that are used by the ROM to determine which peripheral is being used for boot and other boot configuration parameters. The 16 bootmode signals are tied to a specific pad of the SoC, and the AM62x uses most of the McASP2 interface pads for the bootmode pads.

Each bootmode pad requires an external pull up or pull down resistor to define a digital logic high or low state for the associated bootmode signal during the power up sequence.

Because the McASP2 signals are shared with the bootmode logic, it is very important to review and ensure the following:

- The audio device connected to the McASP2 interface does not have the ability to drive on the bootmode signals during the initial power up or in the event of a reset. For example, if PORz is asserted for the AM62x, then the McASP2 audio device should be held in reset as well until the boot sequence is completed.
  - An external driver on the bootmode logic during power up or reset will lead to unpredictable bootmode states.
- The external pull resistors should be placed in line with the signal trace such that they do not introduce a stub. Figure 4-1 shows examples of pull resistors with and without trace stubs, where the green implementation should be replicated for designs.
  - Stubs on the signal traces, especially on the bit clock, will impact the reliability of the audio data as signal reflections caused by the stub can lead to timing errors and signal distortion.

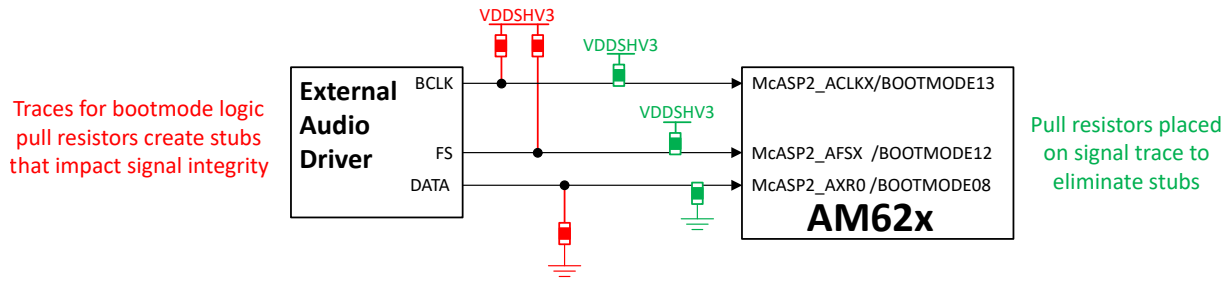

Figure 4-1. McASP Signal Trace Stubs

4.2 McASP Topology for Multiple Devices in Single Clock Domain

The McASP will often be designed into a system where many audio devices share a single clock domain. For example, the TAS6754 is a 4-channel amplifier that can support TDM16. This means that a single McASP's bit clock, frame sync, and data pin can be shared by up to 4 amplifiers. The layout design of these three signals will impact the performance and reliability of the interface.

Figure 4-2 shows three different signal topologies for connecting a bit clock signal to four different amplifiers.

- If the flyby topology is used, then the trace stubs created by each drop on the bus should be uniform in length and as short as possible to reduce reflections. This topology could lead to signal integrity issues depending on the clock frequency and trace length
- A clock fanout buffer, such as the LMK1C1104, is the recommended approach for sharing a clock signal across multiple devices. By redriving the clock, the fanout buffer results in a clock signal with signal integrity that is near the performance of a point-to-point trace.
- A balanced-T or star topology is where a single bus is split into branches that are equivalent in length for each of the devices. The branches that are created should be as short as possible with each stub created for a device being uniform in length.

Note

Regardless of topology, it is always recommended to include a series termination resistor on all McASP signals in close proximity the driver to surpress signal reflections and maintain signal integrity.

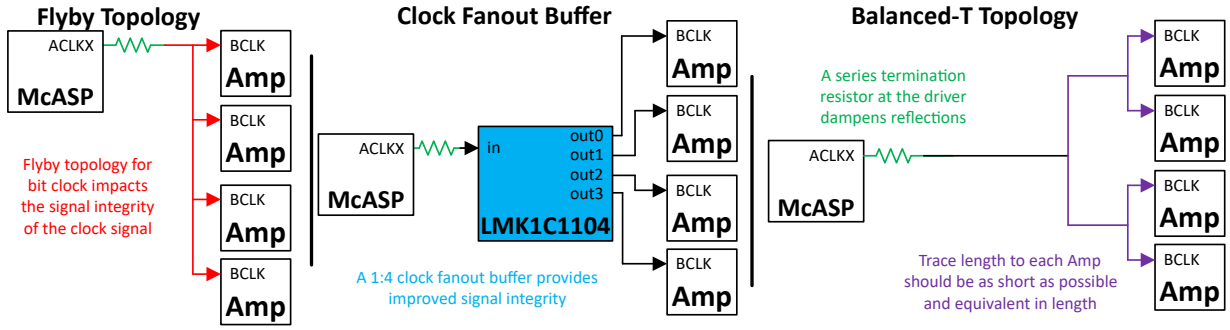

Figure 4-2. Clock Topologies for Multi-Device Audio Systems

5 McASP Practical Examples

5.1 Audio Playback using AUDIO_EXT_REFCLK for Two Clock Domains

Figure 5-1 shows a simple example of how the McASP can use a single internal reference to send and receive audio data across multiple domains. The McASP is operating in asynchronous mode, but because the root clock source is the same for the transmit and receive domain, there is no risk of buffer overrun or underrun (as long as the audio data frame formatting is the same on the input and output).

For this system, an external LVCMOS oscillator (such as the LMK6CE024576) is used to generate an audio clock rate frequency of 24.576 MHz. In this case, the TX and RX domains both have AHCLK configured as externally generated (from AUDIO_EXT_REFCLK0) while ACLK and AFS are configured to be internally generated.

The audio data frame is four audio channels for a single TDM4 stream and, assuming that the word depth is 32 bits, then the bit clock can be calculated based on the product of 4 channels of 32 bit words that are sampled at 48kHz = 4*32*48,000 = 6.144 MHz.

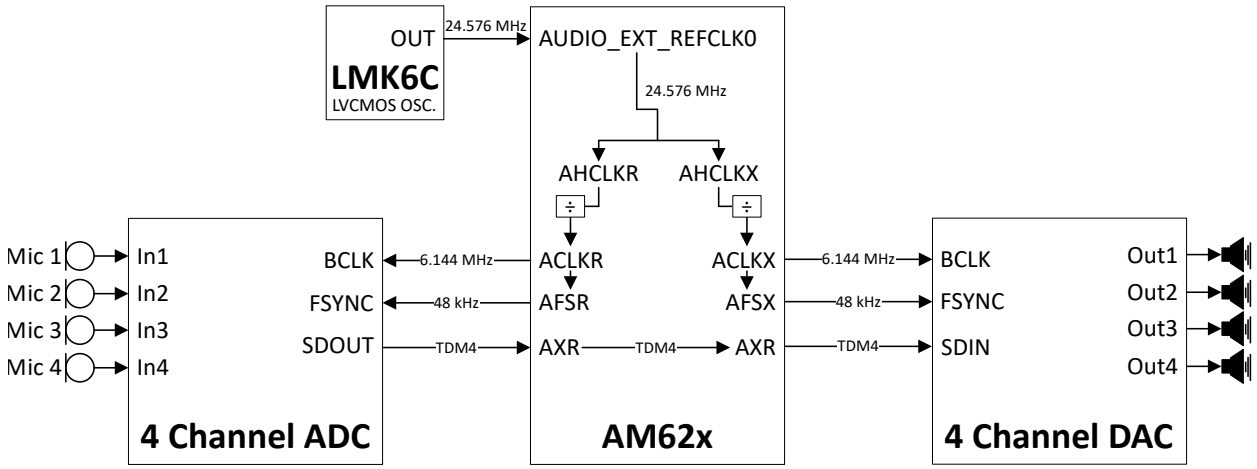

Figure 5-1. ADC DAC Audio Playback

5.2 Audio Playback with External Clock Source and McASP SYNC mode

Figure 5-2 shows a simple example of how a McASP can send and receive audio data with only a single clock reference. The McASP is operating in synchronous mode meaning that the transmit bit clock and frame sync are internally routed to the receive bit clock and frame sync, respectively. The internal routing of the RX domain allows for a single McASP instance to have serializers for input and output of audio data as long as all audio data streams have the same frame formatting.

For this system, a 4-channel codec is the clock controller for both the bit clock and frame sync. The TX and RX domains are in SYNC mode and have ACLK and AFS configured to be externally generated. If bit clock and frame sync are externally generated, the AHCLK is not required for operation and can be considered a "don't care" value.

The audio data frame is four audio channels for a single TDM4 stream and, assuming that the word depth is 32 bits, then the bit clock can be calculated based on the product of 4 channels of 32 bit words that are sampled at 48 kHz = 4*32*48,000 = 6.144 MHz.

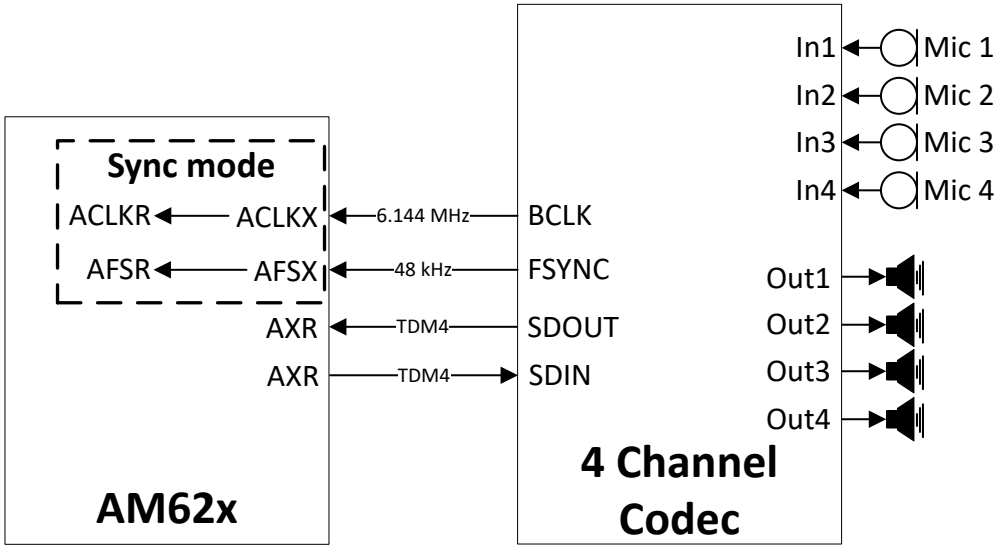

Figure 5-2. Codec Playback with McASP as Clock Peripheral in Sync Mode

6 Key Audio System Design Takeaways

- The McASP has two modes of operation for clock synchronization:
  - Sync mode: where the ACLKX and AFSX signals are internally routed to ACLKR and AFSR and all audio data is sent and received with a single clock domain.
  - Asynchronous mode: The TX and RX clock domains are independent of each other and the audio data clock domains are determined by the serializer IO direction.
- Multizone audio systems ideally have one clock reference for all generated bit clocks and frame syncs in order to avoid audio data buffering issues. For the AM62x devices, the clock reference must be provided by an external source.
  - Either the McASPs need to be configured to internally reference an AUDIO_EXT_REFCLK input or they need to have the bit clock and frame syncs configured to be externally generated.
    - If the external source does not have a device level high-frequency reference, then the bit clock must also be routed to an AUDIO_EXT_REFCLK input to enable other McASP instances with the same reference.
- Carefully review all bootmode signals that are shared with McASP signals to ensure that there aren't any unnecessary trace stubs introduced on the clock or data signal lines.
- For clock and data signals that are shared across multiple devices, ensure that the layout topology does not impact the performance of the signal.
  - Always simulate the signal with the proposed layout topology to ensure the reliability and integrity of the audio data transfers.

7 References

- *AM62D-Q1 Signal Processing Microprocessor*
- *AM62D-Q1 Signal Processing Microprocessor Data Sheet*
- *AM62D-Q1 Signal Processing Microprocessor Technical Reference Manual*
- *AM620-Q1 Signal Processing Microprocessor*
- *AM620-Q1 Signal Processing Microprocessor Data Sheet*
- *AM620-Q1 Signal Processing Microprocessor Technical Reference Manual*

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025