Application Report
SLAA469A–September 2010–Revised June 2019

Audio Serial Interface Configurations for Audio Devices

Jorge Arbona, Uttam Agarwal — Audio Converter Products

ABSTRACT

An audio serial interface (ASI) provides a means to transfer non-buffered audio data between processors and audio converters. This data is typically encoded in PCM twos complement format, although other format variations may be possible to achieve companding for lower data rate transfers. Audio converters based on the delta-sigma (ΔΣ) architecture require an internal master clock that operates at a much faster rate than the target sample rate. Although there are several means to obtain this master clock, take care to ensure that this clock does not drift with respect to the ASI. This application report discusses several configurations that prevent such situations.

Contents

List of Figures

1 Introduction

Each system has different requirements when it comes to interfacing to an audio device ASI. The most common configurations are the master and slave modes. When the audio device ASI is configured in master mode, its bit clock (BCLK) and word clock (WCLK) pins are output. In slave mode, BCLK and WCLK are inputs to the device ASI. This relationship might seem straightforward. However, take care when the ASI is configured in slave mode to ensure that the oversampled data that is decimated always fall within the correct target rate time slot.

If a master clock is a free-running clock and it is fed to a converter, it is not frequency-locked to the frame clock (WCLK) of an independent ASI. Any deviation from the ideal eventually results in a skipped or repeated sample (assuming that the architecture repeats samples). For example, if a host processor provides an ideal 48-kHz WCLK with respect to absolute time, its respective ideal master clock could be exactly (128 • WCLK) = 6.144 MHz. If a master clock from a non-ideal crystal is provided directly to the converter modulator with a 0.001% error, this clock could result in 6.14393856 MHz. Eventually this slower clock results in a repeated sample out of the ASI bus. Of course, there is no such thing as an ideal master or ASI clocks.

[1] All trademarks are the property of their respective owners.

Figure 1 illustrates a simplified case in which a hypothetical analog-to-digital converter (ADC) operating at Nyquist frequency (for simplification purposes) results in a duplicated sample on the ASI bus. In this example, the hypothetical converter ideally hands over its data on the middle of a frame on the falling edge of the master clock. These data should then be ready to be transferred in the beginning of the next frame. As shown in Figure 1, the master clock is actually slower than the ideal. This configuration eventually drifts the clock enough (with respect to the ASI frame) such that there is a frame that does not receive new data (as shown at the end of frame #3).

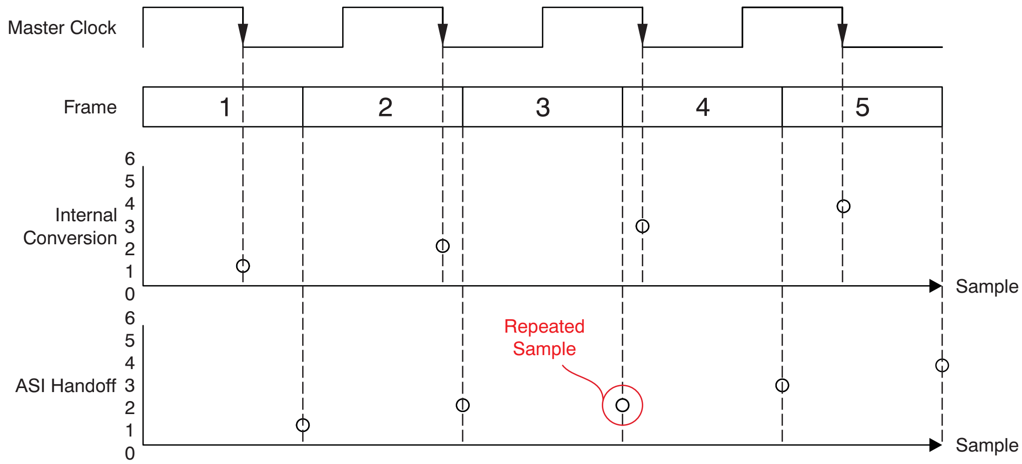

Figure 1. Repeated Sample (Master Clock Slower than Ideal)

If the master clock is faster than the ideal, then two ADC samples may be written within a single frame resulting in a skipped sample through the ASI bus. For the DAC data received from the ASI bus, a faster modulator clock than the ideal results in duplicate samples and a slower modulator clock results in skipped samples at the modulator output.

2 ASI Configurations

As a general requirement, the internal master clock of a converter must be frequency-locked with the ASI frame clock (typically available through WCLK pin). This condition does not mean that the master clock and the ASI frame need to be phase-locked. What is important is that these clocks do not drift over time with respect to each other.

In TI's AIC family of devices, the ASI is composed of a bit clock, word clock, data in, and data out. The bit clock (typically the BCLK pin) clocks DOUT (ADC) data, and latches DIN (DAC) data for each word clock (WCLK) frame. The ASI bus has timing requirements itself (which can be found in the respective device data sheet). However, these are not related to the MCLK, only to the ASI itself.

2.1 ASI Slave Mode

In slave mode, the host processor generates the bit clock and word clock from a system clock. To obtain an audio clock, the system clock is often synthesized to a number divisible by 44.1 kHz or 48 kHz. Figure 2 shows such example. This configuration ensures that the frame clock (accessible at the device WCLK pin) does not drift with respect to the master clock generated by the frequency synthesizer.

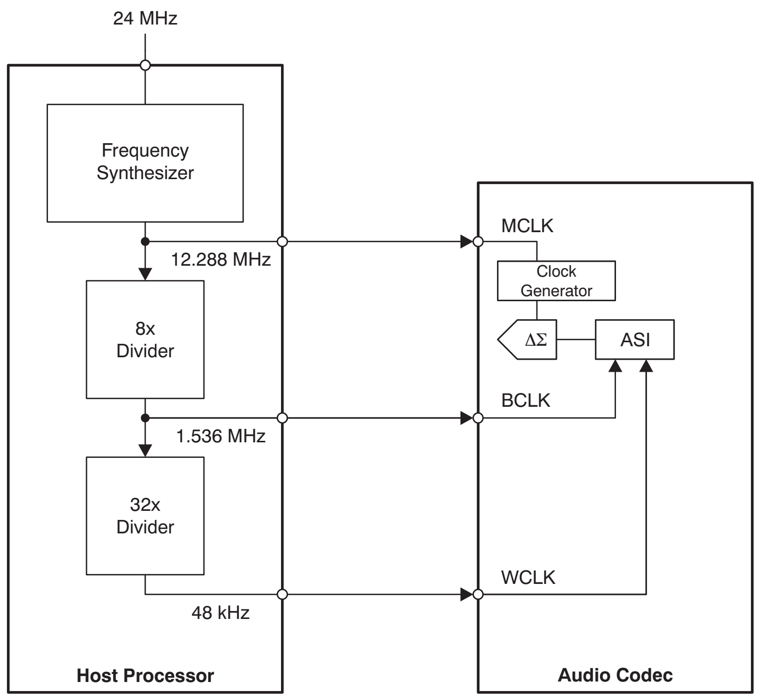

Figure 2. ASI Slave Mode

For cases where a master clock output is not available, obtaining the audio device MCLK directly from the frequency synthesizer source (for example, 24-MHz clock shown in Figure 2) may or may not be suitable in some applications. Some hosts may receive another clock, such as a USB start of frame (SOF) tick, as a reference which may continually change the phase of the ASI bus relative to the external clock.

Figure 3 shows a configuration that should be avoided. Because the crystal and the frame clock are independent of each other, they eventually drift and cause skipped and repeated samples.

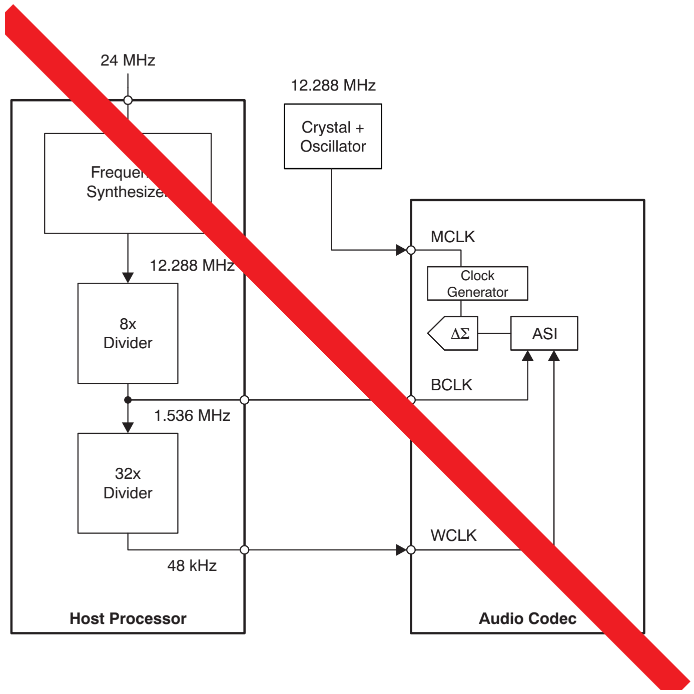

Figure 3. ASI Slave Mode (Independent Master Clock)

Some audio devices, such as the TAS2505 and TAS2557, are capable of deriving the internal master clock from an external BCLK, as shown in Figure 4. However, BCLK must be fast enough to be within the PLL input frequency specification. For low WCLK frequencies, this condition can be solved by increasing the number of BCLK cycles per WCLK frame enough to satisfy these requirements.

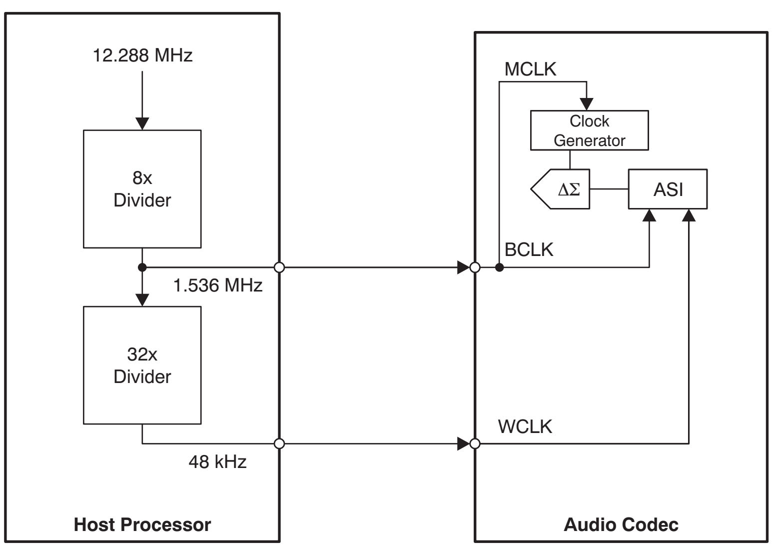

Figure 4. ASI Slave Mode (Generating Master Clock from BCLK)

2.2 ASI Master Mode

In master mode, the BCLK and WCLK are outputs from the audio device. These clocks are derived from an external master clock, such as an oscillator, as shown in Figure 5. The ASI bus is derived from the device master clock input which prevents drift between both.

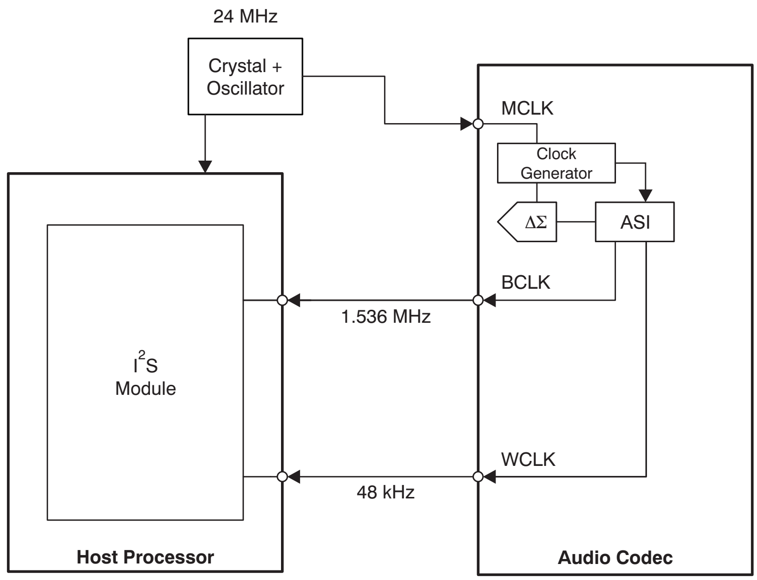

Figure 5. ASI Master Mode

2.3 ASI Hybrid Modes

Two additional modes are possible in most AIC family devices:

- **BCLK is the output and WCLK is the input.**

  In this case, BCLK is internally derived from MCLK and sent to the host processor, which in turn should generate a WCLK that conforms to the ASI specification of the device. BCLK does not drift with respect to MCLK; thus, the generated WCLK does also not drift.

  Additional information about this configuration can be found in the *Configuring I2S to Generate BCLK from Codec Devices & WCLK from McBSP Port Application Report*.

- **BCLK is the input and WCLK is the output.**

  This mode has similar constraints to the ones mentioned in Section 2.1. In this mode, the audio device monitors the BCLK pin to keep the WCLK output timing within the ASI bus specification. BCLK must not drift with respect to the master clock to ensure that the generated WCLK does not drift with respect to MCLK.

  The internal master clock can also be generated from the BCLK (similar to Figure 4, but with WCLK as an output).

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (September 2010) to A Revision	**Page**
• Changed instances of "codec" to "device".	1
• Edited application report for clarity.	1
• Changed TLV320AICxxx to TAS25xx.	4

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, 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 will 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 (www.ti.com/legal/termsofsale.html) 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.