

# ASRC

## Audio Sample Rate Converter



The ASRC core is a compact and high-performance audio sample rate converter. It accurately converts digital audio signals between different sample rates while maintaining signal integrity and minimizing distortion. Supporting both asynchronous and synchronous conversion modes, the low latency ASRC can be used in real-time streaming applications as well as in high-speed batch processing environments. The core efficiently handles multi-channel audio streams, making it ideal for professional audio systems, telecommunications, and embedded applications that require seamless audio synchronization across different formats.

The asynchronous mode dynamically estimates the input and output sample rates and adjusts the conversion ratio in real time. This is particularly useful in live audio applications where input and output clocks are not synchronized, such as professional audio mixers, AV receivers, and VoIP systems. The synchronous mode, on the other hand, operates as a fixed-ratio converter, suitable for applications where the sample rate ratio is predetermined, such as high-speed offline processing of stored audio files.

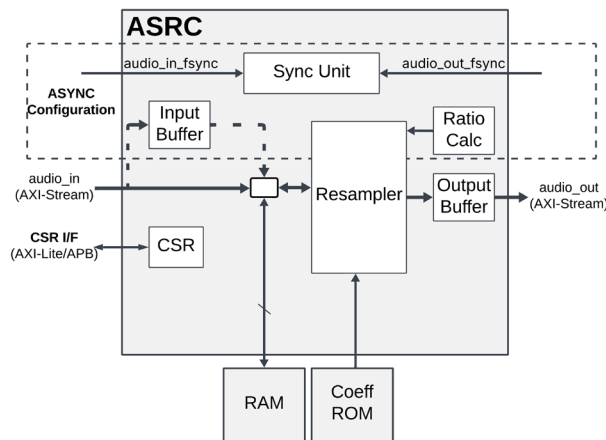
The ASRC Core supports conversion ratios from 1:24 to 24:1, allowing it to process a wide range of industry-standard sample rates, including 8 kHz (telephony), 44.1 kHz (CD quality), 48 kHz (broadcast), and 192 kHz (high-resolution audio). It is capable of handling tens to hundreds of Time-Division Multiplexed (TDM) audio channels, depending on the system clock frequency, making it a scalable solution for multi-channel processing.

A key performance metric of the ASRC Core is its low distortion and noise performance, achieving an average Total Harmonic Distortion Plus Noise (THD+N) of -130 dB, ensuring transparent and high-fidelity audio conversion. By using a linear interpolation method to compute filter coefficients in real time, memory requirements are significantly reduced while still effectively approximating the performance of an ideal, but large, look-up table.

Designed for ease of integration, the ASRC Core supports AMBA® AXI4-Stream interfaces for audio data and multiple control interface options, including AXI-Lite, APB, and a Generic microcontroller interface. It is ready for either ASIC or FPGA implementations, with optimized resource usage to enable deployment in cost-sensitive and power-constrained environments. Interrupt-driven error detection for input overflow, output underflow, and incorrect channel number ensures stable and reliable operation in critical applications.

The comprehensive set of deliverables includes Verilog RTL or netlist, testbenches, simulation scripts, synthesis constraints, and a software driver, making it easy for developers to integrate, verify, and deploy within their systems.

### Block Diagram



### FEATURES

#### Real-Time & High-Speed Modes

- **Asynchronous Mode:** Ideal for real-time audio streaming with automatic sample rate detection
- **Synchronous Mode:** Functions as an accelerator for batch processing of stored audio files

#### Multi-Channel Audio Processing

- Supports tens to hundreds of TDM channels
- Scalable based on system clock frequency

#### Fast Synchronization and Low-Latency

- Sub 100ms sync time at 100 MHz in asynchronous mode
- Updates conversion ratio dynamically to compensate for minor clock variations in streaming applications
- Less than one sample group delay after reset, ensuring rapid synchronization

#### Wide Range of Audio Formats

- Input/output sample rates from 8KHz up to 192KHz
- Sample depth up to 24-bit

#### Ultra-Low Distortion

- Industry-leading THD+N on average of -130 dB with as few as 37 filter taps
- Never higher than -126 dB for conversions between 8kHz and 192kHz
- Prevents artifacts caused by jitter or drift

#### Easy to Integrate

- Standardized interfaces
  - Audio Data: AXI4-Stream input/output
  - Control: 32-bit AXI-Lite, APB, generic microcontroller interface
- LINT-clean, scan-ready design with clean CDC boundaries

## Applications

- **Professional and Consumer Audio Systems** such as digital mixers, AV receivers, or music production hardware.
- **Broadcast and Live Streaming** to ensure smooth audio conversion across diverse media formats.
- **Automotive Infotainment** for synchronizing multiple audio sources within advanced vehicle sound systems.
- **Gaming and VR audio processing** where high-fidelity spatial sound is paramount to provide realistic immersive experiences.
- **Telecommunications and VoIP** to adapt audio sample rates in real-time communication systems.

## Key Benefits

- **Studio-Grade Audio Quality**—Professional 24-bit support for crystal-clear sound reproduction.
- **Scalable and Efficient**—Handles multiple audio streams while maintaining low power consumption.
- **Fast Synchronization**—Sub 100ms sync time in asynchronous mode ensures quality real-time conversion.
- **Easy integration**—Designed for straightforward system integration.

## Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

## Deliverables

- Verilog RTL code or targeted FPGA netlist.
- Verilog testbench and simulation scripts.
- FPGA synthesis scripts and timing constraints.
- ASIC synthesis scripts and timing constraints.
- Bare-metal software driver and example user firmware.
- Comprehensive user and system integration documentation.
- Upon request: linting, simulation coverage, synthesis reports, and FPGA testing environment.

## Size and Performance

The ASRC can be mapped to any AMD® FPGA (provided sufficient silicon resources are available). The FPGA resources requirements depend on the core configuration. With a sample depth of 24-bits, max sample ratio of 24:1, 16 channels, and an APB configuration interface, the core utilizes about 2300 LUTs and 36 RAMB36 of Memory. Please contact CAST to get characterization data for your target configuration and technology