



Advanced Audio Coding (AAC) Decoder for the ARC Processor

Data Sheet (V1.02)

– subject to change without notice –

Contact

CorePool
Fraunhofer Institute Integrated Circuits
Am Wolfsmantel 33
91058 Erlangen
Germany

Phone: +49 (0) 9131 776 777
Fax: +49 (0) 9131 776 499
Email: info@corepool.com
Internet: <http://www.corepool.com>

1 Overview

This document describes the key features of the AAC software decoder for the ARC processor core.

The Advanced Audio Coding (AAC) decoder specified in this document is designed to fulfill the requirements for a “low complexity profile” AAC audio decoder as laid down in ISO/IEC 13818-7 (Information technology – Generic coding of moving pictures and associated information). The decoder incorporates Fraunhofer IIS’s distinct knowledge in high quality audio coding as it is based on Fraunhofer IIS’s fixed point reference decoder. The decoder is designed to be standard compliant with respect to the conformance testing procedure defined in ISO/IEC 13818-4:1998 / Amd. 1, Section 2.6.

2 Architecture

The basic architecture of the AAC decoder is shown in Fig. 1 below:

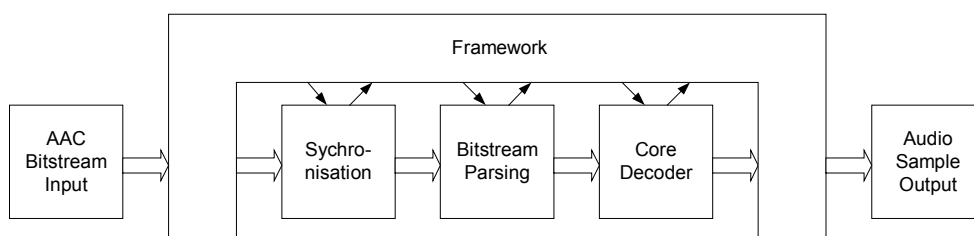


Fig. 1: Decoder Architecture

The decoder is subpartitioned into 3 different layers:

- The overall control framework.
This framework is provided as ‘C’ source code and defines the control flow in the AAC decoder. The distribution includes sample framework code for general purpose “file player” type applications. This code may be tailored for application specific requirements, control flow and error handling.
- The input and output “plug-ins”.
Generic implementation and sample code for “file I/O” is provided as ‘C’ source code. This code may be adapted or re-written to connect the AAC

decoder to the system environment, e.g. to receive data over a serial link and to “play” audio data using a D/A converter.

- The AAC core decoder functions.
These parts are provided as object code only and implement the core AAC decoder. The core decoder functions are subdivided into a number of stages (like synch detection, bitstream parsing and decoding), controlled and linked together by the framework code. Data flow within the core decoder is not made visible to the framework, but all relevant information (like status, bitstream and program information) is provided.

All AAC decoder specific functions are clearly de-coupled from the application. This enables the system integrator to adapt this decoder into its application or system specific environment, without tampering the core decoder functionality.

3 Core Decoder Specification

The decoder is designed to handle “error-free” AAC bitstreams in ADTS and ADIF format only. Additional error concealment *may* be implemented by the user as part of the control framework.

Profile	Low Complexity Profile, with L1.0.0.0 and L2.0.0.0 capabilities ¹
Sampling Rate	all rates defined in 13818-7, 8000 to 96000 samples / second ²
Bitstream Formats	ADIF and ADTS (auto detection)
Multi-Program capabilities	able to handle multi-program bitstreams and to decode a single (user selected) program
Bitstream Element handling	decodes SCE and CPE elements, processes DSE, PCE and END elements, ignores (skips) all other data stream elements The contents of DSE's is extracted and provided to the framework The contents of PCE's is extracted and provided to the framework

¹ other capabilities, like L5.1.0.1 and L7.1.0.2 are available on request

² required processing power varies with sampling rate and input data rate; maximum rate may be limited.



mix-down capabilities	not supported
CRC protection	disabled, due to extra buffer requirements

4 Software Environment

The MetaWare tools for ARC are required to compile and link the 'C' source code and object code modules provided as part of the AAC decoder distribution. Requires tools version 4.3 or later.