ALSA SoC Layer

The overall project goal of the ALSA System on Chip (ASoC) layer is to provide better ALSA support for embedded system-on-chip processors (e.g. pxa2xx, au1x00, iMX, etc) and portable audio codecs. Prior to the ASoC subsystem there was some support in the kernel for SoC audio, however it had some limitations:-

* Codec drivers were often tightly coupled to the underlying SoC CPU. This is not ideal and leads to code duplication - for example, Linux had different wm8731 drivers for 4 different SoC platforms.

* There was no standard method to signal user initiated audio events (e.g. Headphone/Mic insertion, Headphone/Mic detection after an insertion event). These are quite common events on portable devices and often require machine specific code to re-route audio, enable amps, etc., after such an event.

* Drivers tended to power up the entire codec when playing (or recording) audio. This is fine for a PC, but tends to waste a lot of power on portable devices. There was also no support for saving power via changing codec oversampling rates, bias currents, etc.

ASoC Design

The ASoC layer is designed to address these issues and provide the following features :-

* Codec independence. Allows reuse of codec drivers on other platforms and machines.

* Easy I2S/PCM audio interface setup between codec and SoC. Each SoC interface and codec registers its audio interface capabilities with the core and are subsequently matched and configured when the application hardware parameters are known.

* Dynamic Audio Power Management (DAPM). DAPM automatically sets the codec to its minimum power state at all times. This includes powering up/down internal power blocks depending on the internal codec audio routing and any active streams.

* Pop and click reduction. Pops and clicks can be reduced by powering the codec up/down in the correct sequence (including using digital mute). ASoC signals the codec when to change power states.

* Machine specific controls: Allow machines to add controls to the sound card (e.g. volume control for speaker amplifier).

To achieve all this, ASoC basically splits an embedded audio system into 3 components :-

* Codec driver: The codec driver is platform independent and contains audio controls, audio interface capabilities, codec DAPM definition and codec IO

functions.

* Platform driver: The platform driver contains the audio DMA engine and audio interface drivers (e.g. I2S, AC97, PCM) for that platform.

* Machine driver: The machine driver handles any machine specific controls and audio events (e.g. turning on an amp at start of playback).

Documentation
=============

The documentation is spilt into the following sections:-

overview.txt: This file.

codec.txt: Codec driver internals.

DAI.txt: Description of Digital Audio Interface standards and how to configure a DAI within your codec and CPU DAI drivers.

dapm.txt: Dynamic Audio Power Management

platform.txt: Platform audio DMA and DAI.

machine.txt: Machine driver internals.

pop_clicks.txt: How to minimise audio artifacts.

clocking.txt: ASoC clocking for best power performance.