Contribution ID: **41** Type: **not specified**

What to do with kconfig.socs?

Tuesday, 13 September 2022 16:10 (30 minutes)

The goal of kconfig.socs originally was to have SOC_FOO symbols so that a user "can just push a button and have everything they need to boot", which was implemented via selects. This sort of behaviour for a kconfig symbol is at odds to other architectures and not maintainable in the long term as the number of SoCs grows and/or the select dependencies change.

As things stand, different SOC_FOO symbols have different behaviour:
- some directly select the drivers if a prereq is set
- others use SOC_FOO symbol as a prerequisite to expose drivers during configuration
- some enable prequisites to ensure drivers will be exposed & rely on a depends on SOC_FOO + default SOC_FOO comination in the driver's kconfig entry to enable the driver itself.

It would be great to have a discussion and settle on a single, consistent approach for SOC_FOO symbols (or if someone has a better idea for a replacement...) before it becomes unwieldy.

Secondly, depending on what is decided on, what should the scope of the symbol be?
Should it enable a bare minimum for boot, and then expose other options as possibilities?
Or should it turn on all bells/whistles for that SoC?

Primary authors: DOOLEY, Conor; DABBELT, Palmer (Google)

Presenter: DOOLEY, Conor

Session Classification: RISC-V MC

Track Classification: LPC Microconference: RISC-V MC