

Background

Kconfig.socs today:

- 5 entries, 4 SoCs and QEMU Virt
- Canaan is a special case
- symbols named SOC_FOO
- SIFIVE & STARFIVE but also MICROCHIP_POLARFIRE

Original goals:

- "An easy place for users to say 'I want all the support for SOC X'" [2]
- None of the symbols currently do this & how newcomers expect it to work
- Large degree of churn for new selects & bloat with per SoC symbol
- Selects can override desired "m", RTC_DRV_GOLDFISH [1]

Background

Kconfig.socs today:

- 5 entries, 4 SoCs and QEMU Virt
- Canaan is a special case
- symbols named SOC_FOO
- SIFIVE & STARFIVE but also MICROCHIP_POLARFIRE

Original goals:

- "An easy place for users to say 'I want all the support for SOC X'" [2]
- None of the symbols currently do this & how newcomers expect it to work
- Large degree of churn for new selects & bloat with per SoC symbol
- Selects can override desired "m", RTC_DRV_GOLDFISH [1]

The symbols

Existing SoCs:

- MPFS: selects clk/plic directly, exposes other symbols
- STARFIVE: enables stuff via default & selects their subsystems
- SIFIVE: selects drivers iff subsystem, exposes other symbols
- CANAAN: mix of "select iff subsystem" & defaults
- VIRT: mix of "select iff subsystem" & explicit selects

My initial problem statement...

Incoming SoCs (Renesas & Allwinner)

- "Incumbents" have no ARM legacy
- Renesas & Samuel want to redefine their ARCH_FOO symbol
- ARCH_FOO already littered all over the kernel
- Why have SOC_FOO & ARCH_FOO?
- Eg. Renesas use kconfig in drivers/soc to control selects



The symbols

Existing SoCs:

- MPFS: selects clk/plic directly, exposes other symbols
- STARFIVE: enables stuff via default & selects their subsystems
- SIFIVE: selects drivers iff subsystem, exposes other symbols
- CANAAN: mix of "select iff subsystem" & defaults
- VIRT: mix of "select iff subsystem" & explicit selects

My initial problem statement...

Incoming SoCs (Renesas & Allwinner)

- "Incumbents" have no ARM legacy
- Renesas & Samuel want to redefine their ARCH_FOO symbol
- ARCH_FOO already littered all over the kernel
- Why have SOC_FOO & ARCH_FOO?
- Eg. Renesas use kconfig in drivers/soc to control selects



The symbols

Existing SoCs:

- MPFS: selects clk/plic directly, exposes other symbols
- STARFIVE: enables stuff via default & selects their subsystems
- SIFIVE: selects drivers iff subsystem, exposes other symbols
- CANAAN: mix of "select iff subsystem" & defaults
- VIRT: mix of "select iff subsystem" & explicit selects

My initial problem statement...

Incoming SoCs (Renesas & Allwinner)

- "Incumbents" have no ARM legacy
- Renesas & Samuel want to redefine their ARCH_FOO symbol
- ARCH_FOO already littered all over the kernel
- Why have SOC_FOO & ARCH_FOO?
- Eg. Renesas use kconfig in drivers/soc to control selects



Options

My preference:

- Switch from SOC_ to ARCH_ symbols
- Use default ARCH_FOO, not selects for periph drivers
- Selects only for core things like the plic?
- Beyond core, selects only for essential subsystems?
- Only add globally useful things to arch defconfig?

~~Remain as we are:~~

Over to you:

Options

My preference:

- Switch from SOC_ to ARCH_ symbols
- Use default ARCH_FOO, not selects for periph drivers
- Selects only for core things like the plic?
- Beyond core, selects only for essential subsystems?
- Only add globally useful things to arch defconfig?

~~Remain as we are:~~

Over to you:

Options

My preference:

- Switch from SOC_ to ARCH_ symbols
- Use default ARCH_FOO, not selects for periph drivers
- Selects only for core things like the plic?
- Beyond core, selects only for essential subsystems?
- Only add globally useful things to arch defconfig?

~~Remain as we are:~~

Over to you:



<https://lore.kernel.org/linux-riscv/mhng-99340121-50f8-49ca-ae6e-0f737fc4d736@palmerdabbelt-glaptop/>



<https://libera.irclog.whitequark.org/riscv/2022-05-04/>