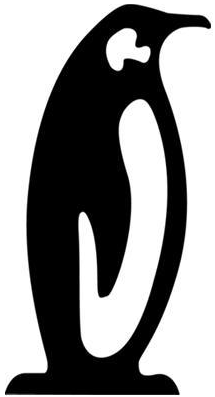


東京 2025 LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025

Power Sequencing for Enumerable Busses

What's Still Missing From the Device Tree and Kernel

東京 2025
LINUX PLUMBERS CONFERENCE
TOKYO, JAPAN / DEC. 11-13, 2025

Chen-Yu Tsai

- Works on ARM-based Chromebooks at Google
- Maintainer of wireless-regdb project
- Maintainer of Allwinner SoC within the kernel on the side

What is an Enumerable Bus?

- Devices can be "enumerated" directly by the host controller without additional hardware descriptions, provided they are operational.
- Making the device "operational" might require additional setup

- SD/MMC/SDIO
- USB
- PCIe
- MDIO

What's the Problem Exactly?

Device tree has *some* support for describing power supplies, clocks and resets for some bus types in a generic fashion.

But the implementation (Linux) doesn't work for all cases.

And what about the bus types that don't have generic descriptions?

MMC / SD / SDIO

MMC Power Sequencing Bindings

Power Sequencing node:

```
&sdhci0_pwrseq: pwrseq {
    compatible = "mmc-pwrseq-simple";
    reset-gpios =
        <&gpio1 12 GPIO_ACTIVE_LOW>;
    clocks = <&clk_32768_ck>;
    clock-names = "ext_clock";
};
```

MMC controller node:

```
mmc3: mmc@1c12000 {
    ...
    vmmc-supply = <&reg_vmmc3>;
    vqmmc-supply = <&reg_vddio>;
    mmc-pwrseq = <&sdhci0_pwrseq>;
};
```

SD/MMC/SDIO - Status Quo

- SD/MMC cards require both core and I/O voltage
- eMMC chips have a reset pin
- SDIO devices may require extra clocks and pin toggles

- First bus type to gain generic (across host drivers) power sequencing in 2015 [1]
- Support onboard SDIO-based WiFi controllers as the main goal
- Generic case works well, unless driver was accidentally disabled

[1] https://lore.kernel.org/all/1421658784-11980-1-git-send-email-ulf.hansson@linaro.org/

USB

USB Connectors and Devices

usb-connector.yaml

- Type C
  - Parsed and handled by type-C drivers
- Type B
  - ID pin, VBUS regulator and detection handled by **PHY drivers**
  - or "**gpio-usb-b-connector**"
- Type A
  - "**usb-a-connector**" is not handled

- Onboard devices
  - usb-device.yaml
  - Onboard USB device driver

- How is VBUS handled for type A host ports?

USB Type A Host Ports and VBUS

Described under USB host

```
&xhci0 {
    vbus-supply = <&usb_vbus>;
};
```

From mt8195-cherry.dtsi

Described under USB PHY

```
&usbphy {
    usb0_vbus-supply =
        <&reg_vcc5v>;
    usb1_vbus-supply =
        <&reg_usb_vbus>;
};
```

From sun55i-a527-cubie-a5e.dts

USB Type A Host Ports and VBUS

Combination of both.

```
&usb {
        vbus-supply = <&usb_pwr_en>;
};

&usb2_phy1 {
        /* Enable the hub ... */
        phy-supply = <&hub_5v>;
};
```

From meson-g12b-odroid-n2.dtsi

Or none.

Regulators set as always on.

USB Type A Host Ports and VBUS

Described under USB PHY as PHY supply

```
&u2phy0 {
        u2phy0_otg: otg-port {
                phy-supply = <&vcc_vbus_typec1>;
        };

        u2phy0_host: host-port {
                phy-supply = <&vcc5v0_host>;
        };
};
```

From rk3399-roc-pc-plus.dts

PCIe

PCIe - Host handling of device power

- Device power supplies described in the host controller node

```
&pcie2x1l2 {
    reset-gpios = <...>;
    vpcie3v3-supply =
        <&vcc3v3_pcie2x1l2>;
};

&pcie3x4 {
    reset-gpios = <...>;
    vpcie3v3-supply =
        <&vcc3v3_pcie30>;
};
```

From rk3588-rock-5b-5bp-5t.dtsi

PCIe Slot Power Description

Represent a PCIe slot as a PCI bridge:

```
&pcie_host {
    pcie@0 {
        compatible = "pciclass,0604";
        vpcie1v8-supply = <&wlan_vddio>;
        vpcie3v3-supply = <&wlan_pmu>;
    };
};
```

Could this also describe onboard PCIe devices integrated in a standard way, i.e. hard-wired as if a slot existed?

PCIe - Status Quo

- Standard hardware
  - PCIe power rails (3v3, 3v3aux, 1v8)
  - PERST#
  - ~~OOB control signals on M.2 slots~~
- Device specific power sequences for integrated designs (chip-on-board)
  - Qualcomm WCN WiFi/BT chipsets integrated on mainboard with all power rails from main PMIC, i.e. not standard M.2 card with 3.3V
- Host driver specific supply handling
  - Latest additions rejected in favor of:
- PCI power control framework
  - drivers/pci/pwrctrl/
  - Core added in v6.11
  - Slot driver added in v6.15
- Power sequencing framework
  - drivers/power/sequencing
  - Added in v6.11

PCIe Device Power and Link Training

- Driver waits for link up before entering PCI core
- Power enabled by PCI pwrctrl devices created by PCI core
- Current workaround
  a. Host driver ignores link up timeout
  b. Pwrctrl ready triggers bus rescan

In dw_pcie_host_init():

- dw_pcie_start_link()
  - PERST# de-asserted
- dw_pcie_wait_for_link()
- pci_host_probe()
  - pci_scan_device()
    - pci_pwrctrl_create_device()

Pwrctrl device probes:

- power on device supplies
- pci_pwrctrl_device_set_ready()
  - pci_rescan_bus()

PCIe Power Sequencing Violations

- PCI controller driver controls native PERST#
- PCI pwrctrl enables power
- PCI standard requires PERST# be asserted until power supplies are valid

In dw_pcie_host_init():

- dw_pcie_start_link()
  - PERST# de-asserted
- dw_pcie_wait_for_link()
- pci_host_probe()
  - pci_scan_device()
    - pci_pwrctrl_create_device()

Pwrctrl device probes:

- power on device supplies
- pci_pwrctrl_device_set_ready()
  - pci_rescan_bus()

PCIe Solutions on the Horizon

Changes proposed by Mani

- New **pwrctrl** APIs to interleave power control with host controller actions
- M.2 E-key slot binding and driver
  - USB not covered in proposal
- M.2 M-key slot binding and driver

MDIO

Generic MDIO Resets

Reset in MDIO device node:

```
mdio {
    ethernet-phy@1 {
        reset-gpios =
            <&gpio1 12 GPIO_ACTIVE_LOW>;
        reset-assert-us = <500>;
        reset-deassert-us = <100>;
    };
};
```

Reset in MDIO bus node:

```
mdio {
    reset-gpios =
        <&gpio1 12 GPIO_ACTIVE_LOW>;
    reset-delay-us = <500>;
    reset-post-delay-us = <100>;
};
```

Generic MDIO Resets - Two Descriptions?

"*... the RESET de-assertion cannot be done in the PHY driver as the PHY will not probe till its reset is de-asserted. So do the RESET de-assertion in the MDIO bus driver.*"

commit 69226896ad63 ("mdio_bus: Issue GPIO RESET to PHYs.")

PHY doesn't respond while it's reset is asserted. MDIO bus scan gets no response, and no device is created.

Work-around: encode the PHY ID in the DT compatible (ethernet-phy binding). However, the actual PHY ID will then not be read, which means if the driver wants to know e.g. the revision, or during production the PHY changes, it will require DT to change.

From Russell King - https://lore.kernel.org/all/aJy_qUbmqoOG-GBC@shell.armlinux.org.uk/

Clocks for PHYs on MDIO bus

- Only under PHY node
- Supported by some PHY drivers

Power Supplies for MDIO devices?

Supply property under controller node

- vcc-supply
  - davicom,dm9000.yaml
- phy-supply
  - allwinner,sun7i-a20-gmac.yaml
  - allwinner,sun8i-a83t-emac.yaml
  - rockchip,emac.yaml
  - rockchip-dwmac.yaml

Nothing generic at the moment.

Do we want something generic?

One core power supply and one (optional) I/O voltage supply is what is most common for Ethernet PHYs.

Ethernet switch chips are more complicated.

Improvements for MDIO?

- Reset GPIOs for non-Ethernet-PHY MDIO devices work already

- Ethernet PHY devices want PHY ID when created
  - Rework PHY framework to split the MDIO power sequencing from PHY ID read-out
- Add generic power supply and clock handling if desired

Summary

Device Tree Recap

Generic description works well

- MMC/SD/SDIO
- USB type B/C connectors

Generic description exists but implementation has issues

- PCIe
- MDIO

Generic description exists but don't work

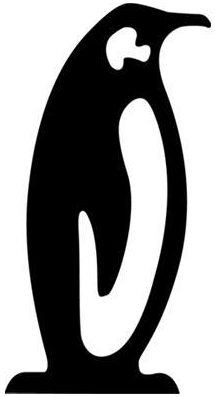
- USB type A connectors

Custom descriptions

- Integrated PCIe devices (chip-on-board)
- USB on-board devices
- VBUS as property of USB host or PHY

Questions for Maintainers and Users

- Do we want more generic descriptions?
  - M.2 slots (WIP)
  - MDIO power supplies
- If generic descriptions exist, should we migrate everyone to them?
  - PCIe slots and devices
  - USB type A host ports
- If generic descriptions exist but don't work, should we fix them?
  - PCIe slot power (WIP)
  - MDIO device reset pins
  - USB type A host ports



東京 2025
LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025