



Linaro
connect
Hong Kong 2018

OSPM 2018-04

Device PM - moving forward

Ulf Hansson, Linaro



Device PM - moving forward

- An update.
- Wakeup management.
- Wakeup management and multiple idlestates per device.
- Support multiple PM domains per device.
- Next steps.



Current state

Driver PM flags:

- DPM_FLAG_NEVER_SKIP, DPM_FLAG_SMART_PREPARE, DPM_FLAG_SMART_SUSPEND, DPM_FLAG_LEAVE_SUSPENDED.
- Deployed for the ACPI and the PCI subsystems.

The runtime PM centric path:

- pm_runtime_force_suspend|resume().
- Widely deployed in drivers and in genpd.
- Improved support for parent/child devices, by Rafael.



Moving forward

- Driver PM flags vs the runtime PM centric path.
- Ideally: Find scenarios and list requirement for a common solution.
 - Avoid powering up devices during system suspend/resume.
- Wakeup management - huh!



Wakeup management

- `device_init_wakeup()`
- `device_can_wakeup()`
- `device_set_wakeup_path()`
- `device_wakeup_enable()`
- `device_wakeup_disable()`
-
-
-



Wakeup management - the problems

- PM domain (genpd) lacks details about device's wakeup configuration.
 - `device_may_wakeup() == true`, but inband or outband IRQ?
 - **Question:** When can genpd power off the PM domain?
- Same issue for wakeups configured during runtime suspend!
- What else?



Wakeup management - existing solutions

- At runtime PM suspend - always allow PM domain to power off.
 - The generic assumption - limits behaviors in drivers!
 - Prevents fine grained PM.
- GENPD_FLAG_ACTIVE_WAKEUP
 - Not sufficient to cover the generic case. Remove?
- PM_QOS_FLAG_NO_POWER_OFF.
 - Not sufficient to cover the generic case. Remove?



Wakeup management - next steps!?

- Add support for multiple idlestates per device.
 - Make both system-wide PM and runtime PM aware of them.
- Allow drivers to dynamically disable an idlestate, depending on the scenario.
- Introduce knowledge about device idlestates in genpd.
 - If deepest idlestate disabled, don't power off?



Multiple PM domains per device - today

- Today - One PM domain per device and one set of dev_pm_ops.

```
struct device {
```

```
...
```

```
    struct dev_pm_domain    *pm_domain;
```

```
...
```

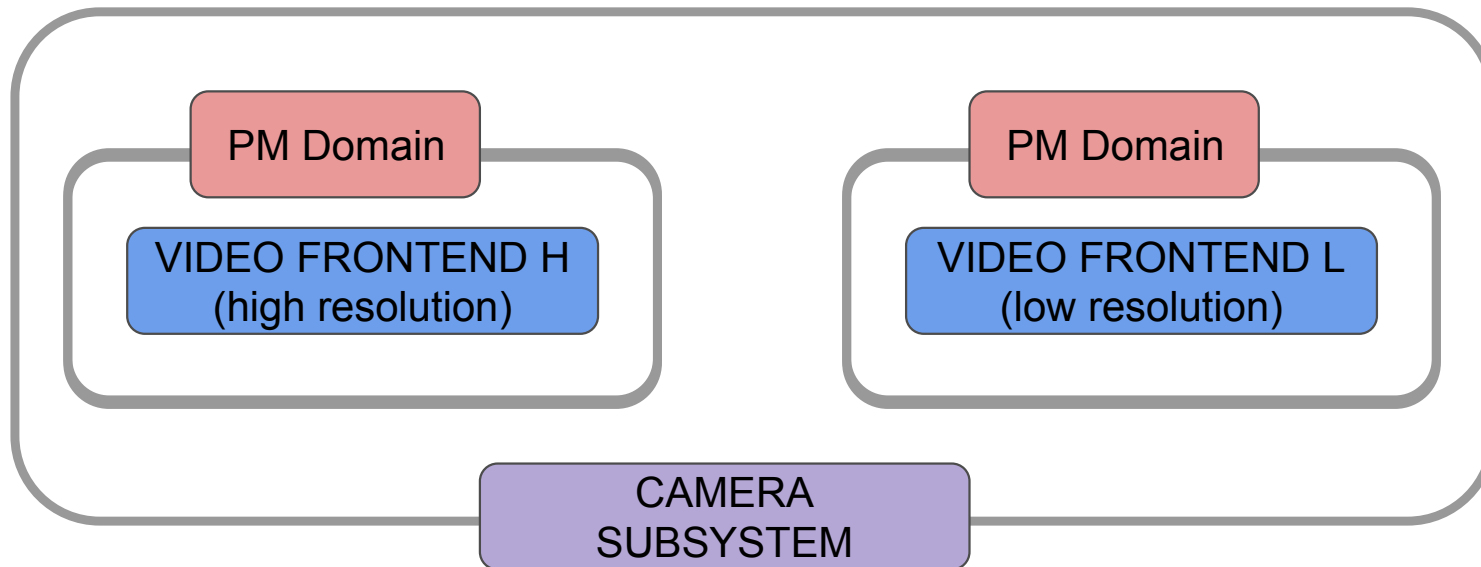
```
}
```

- Convenient and transparent.
 - Device attach/detach to PM domain is transparent to drivers.
 - Driver only need runtime PM deployment to manage PM domains.
- But more flexibility is needed...



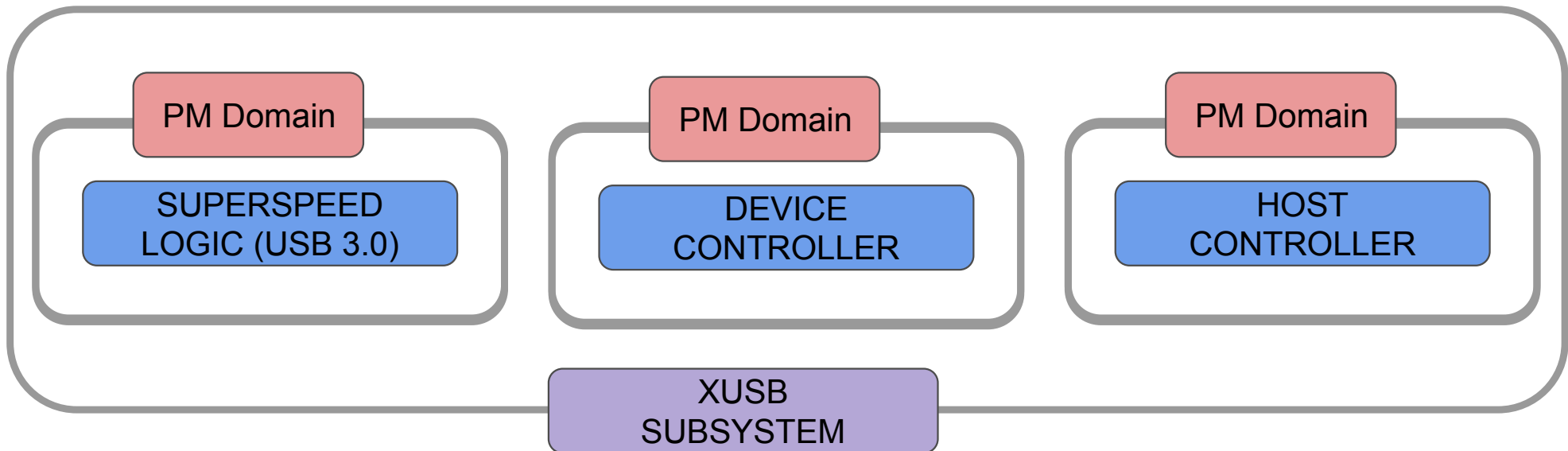
Multiple PM domains per device - why?

- QCOM APQ8096 - Camera subsystem



Multiple PM domains per device - why?

- The Tegra124/210 XUSB subsystem



Multiple PM domains per device - why?

- More...
 - QCOM modem boot use case
 - Renesas using clk **and** power domains.
- Attempts so far..
 - <https://lwn.net/Articles/718263/>
 - <https://lkml.org/lkml/2017/3/28/753>



Multiple PM domains per device - how?

- **No option:** Existing infrastructure, one PM domain pointer per device isn't easy to extend to an “array of pointers”.
- **Option 1:** `pm_domain_get|put()` + `pm_domain_enable|disable()`.
 - + It's generic and any PM domain providers could be registered.
 - - May get over-engineered and certainly a bit complicated.
- **Option 2:** `pm_genpd_get|put()` + `pm_genpd_enable|disable()`.
 - - Works only for genpd.
 - + Rather simple to extend genpd.
 - + Combine old and new use cases.
- **Option 3: ?**





Linaro
connect

Hong Kong 2018

Thank You!

OSPM 2018-04

Ulf Hansson, Linaro

