One idle to rule them all?

Idle management
of CPUs
&
IO devices

Kevin Hilman, Linaro
khilman@kernel.org

Two separate worlds

CPUs

- CPU idle framework
- `cpu_[cluster_]pm_*()`

- not scaling well for SMP or multi-cluster (c.f. coupled idle states)

IO devices

- Runtime PM
- auto-suspend
- PM domains
- generic PM domains (genpd)

One idle to rule them all

What if…

- use runtime PM for CPUs
- and CPU-connected "stuff"
  - interrupt controllers (ARM GIC)
  - floating-point units
  - CPU-local cache (L1$)

- model clusters with genpd
  - CPUs are just "devices" in the genpd
  - genpd includes shared resources (e.g. L2$)

Runtime PM

Next steps, discussion

- genpd evolution
  - locking simplification (Ulf Hansson, merged)
  - removing intermediate states (Ulf Hansson, merged)
  - CPU PM domains, IRQ-safe genpd support (Lina Iyer, posted)

- CPU PM notifiers: `cpu_[cluster_]_pm_[enter|exit]()`
  - used for IRQ chips, floating-point units, PMUs, wakeups, etc.
  - can/should we use runtime PM instead? (runtime PM callbacks instead of notifiers)

- `pm_genpd_attach_cpuidle()`
  - no more users? kill it.

Next steps, discussion (cont)...

- genpd: needs to support multiple levels
    - currently only supports on/off
    - CPU/clusters have more levels (e.g. retention, C-states)
    - IO devices (D-states)
    - RFC by Axel Haslam (BayLibre)

- ACPI 6: low-power idle (LPI)
    - supports hierarchical idle
    - seems to map better to genpd than CPUidle (c.f. Fig 8-46, 6.0 spec)

Credits, Thank you

Collaboration, discussion, review, ...

- ○ Rafael Wysocki
- ○ Ulf Hansson
- ○ Geert Uytterhoeven
- ○ Lorenzo Pieralisi
- ○ Lina Iyer
- ○ Sudeep Holla
- ○ Axel Haslam