SoC Idling
&
CPU Cluster PM
SoC Idling & CPU Cluster PM

- Idle management of devices via runtime PM and the generic PM domain (genpd). A proven concept!
- Idle management of CPUs and clusters. Let’s have one “idle” to rule them all!
Background

- Runtime PM
- System PM
- DevPM QoS
- Generic PM domain (genpd)
Users of genpd

- SH-Mobile
- S3C64xx
- Exynos
- Linux 3.1
- Linux 3.4
- Linux 3.18
- Linux 4.5
- SH-Mobile
- S3C64xx
- Exynos
- IMX
- Ux500
- ZX
- Qcom
- BCM
- Dove
- Mediatek
- Rockchip
- Tegra
Highlights

- Genpd: Maintained by Ulf, Kevin and Rafael
- Various consolidation, fixing issues and regressions
- Genpd: Removing intermediate states from the power off sequence (4.3)
- Genpd: Enabling the runtime PM centric approach
- Genpd: Support for multiple retention states (4.6)
Next steps

- Add genpd power statistics
- Avoid needless wakeup in System PM
CPU Cluster PM - Background and motivation

More...
- CPUs
- integrated devices
- power domains
- micro controllers
- firmware

Kernel needs to evolve...
Idle management today

- The Linux kernel has two distinct ways of managing idle. The CPUIdle framework for CPUs and for all other devices: runtime PM combined with generic power domains (genpd). In addition, CPUIdle is not scaling well for multi-cluster SMP systems and heterogeneous systems like big.LITTLE. To better manage idle for modern SoCs with a hierarchical structure, we are exploring extending runtime PM and genpd to CPUs so there is a unified framework for managing idle across all devices.
SoC Idle: Today
SoC Idle: With CPU PM
Recipe

Start with upstream kernel (v4.5-rc3)

● Prep
  ○ Extend genpd domains to support multiple idle levels
  ○ Extend genpd to support IRQ-safe PM domains

● Add
  ○ Add CPU PM domain framework
  ○ Call Runtime PM from cpuidle

● Sample
  ○ Using CPU PM domains - OS-Initiated
  ○ DT changes for SoC

¹ Patches slated for merge in v4.6
CPU PM framework

- **Init**
  - Read domain topology from DT
  - Setup genpd PM domains

- **Genpd**
  - Last man determination

- **Genpd Gov:**
  - Determine cluster idle state
CPU PM framework

- Better than coupled cpuidle
  - CPUs are not woken up when ready to enter coupled state
  - Handles multi-level CPU-Domain topology
- Is not MCPM
  - MCPM handles low level race between CPUs
  - Some v7 SoCs may still need MCPM with CPU PM framework
PSCI: PC vs OSI

- Platform Coordinated (PC)
  - Default PSCI mode
  - F/W decides on CPU-Domain idle state when all CPUs are idle
  - F/W does not know of Linux CPU QoS requirements, latency and next CPU wakeup
  - F/W decision to power off domain may be detrimental to power and performance

- OS-Initiated (OSI)
  - Optional from PSCI v1.0 onwards
  - Linux decides the CPU-Domain idle state
  - The last CPU in a domain provides the idle state of the cluster, coherency
  - Linux can make a wise choice of CPU-Domain idle state knowing QoS, latency, predicted CPU wakeup etc.
OSI using CPU PM

- Query PSCI_FEATURES in F/W for OSI support
- Setup CPU PM Domains
  - Reads State-IDs for Cluster and Coherency idle levels from DT
- Callbacks for Domain ON/OFF
  - State-IDs passed from CPU PM
  - Aggregate State-IDs against the CPU
- CPU calls F/W with Composite State-ID
- NO SoC specific drivers needed for ARM v8
PSCI: Composite State-ID

- CPUidle: CPU state
- CPU PM Gov: Cluster state
- CPU PM Gov: Coherency state
Changes for Vendors: ARM64

- **DT**
  - Domain hierarchy
  - Domain idle states
- **Driver**
  - None if F/W supports OSI
Changes for Vendors: ARM32

- **DT**
  - Domain hierarchy
  - Domain idle states
- **Driver**
  - Handle power ON/OFF callbacks
  - MCPM or any race avoiding last man logic
  - Revisit SoC specific CPUIdle hacks
Genpd: IRQ-Safe Restriction

- IRQ-safe domains can only have IRQ-safe sub-domains
- Other cases with domains and devices remain the same
Status of patches

- PM / Domains: Multiple genpd states: **Merged**
- PM / Domains: IRQ safe domains: **Under review**
- CPU PM domains: New framework for CPU clusters: **Under review**
- PSCI 1.0 OS-Initiated: Support for domain hierarchy: **Under review**

RFC submission on ML:


Sample ARM v7a implementation

Results

- SoC idle saves power when CPUs are online
  - Critical power saving comes from powering off caches and peripheral h/w
  - ~20 mA @800 Mhz of power saving at idle
- ~5 μs addition to idle enter path
  - Exit latency depends on what happens at domain OFF
- Implemented and tested on DB410c / 96Board