SiFive

RISC-V Platform Power Management Interface Discussion

Linux Plumbers Conference 2018
Paul Walmsley

What about power management is CPU architecture specific?

What about power management is CPU architecture specific?

Almost nothing …

What about power management is CPU architecture specific?

Almost nothing …

… except that, in recent years, those defining a CPU architecture have also defined recommended methods for architecture binaries to interact with platforms (the “interface method”)

What about power management is CPU architecture specific?

Almost nothing …

… except that, in recent years, those defining a CPU architecture have also defined recommended methods for architecture binaries to interact with platforms (the “interface method”)

These interface methods are usually paired with a set of recommended platform operations to implement (“the operations”)

Why have standard CPU PM operations?

- **Portable binaries**
  - OS distributions
  - Emulators
  - Simulators

Why have standard CPU PM operations?

- **Portable binaries**
  - OS distributions
  - Emulators
  - Simulators

- **Boot one kernel image on multiple vendor hardware platforms; push *some* platform variation behind the SBI**

What are some common interface methods?

- MMIO
- Special instructions (MSRs, coprocessor moves, etc.)
- Triggering an exception
- Jumps

What are some common interface methods?

- MMIO
- Special instructions (MSRs, coprocessor moves, etc.)
- Triggering an exception
- Jumps

Almost nothing about these methods has any bearing on the power management operations themselves

What’s important are the operations

The operations define the functionality

What’s important are the operations

The operations define the functionality

So what operations should be defined?

What’s important are the operations

The operations define the functionality

So what operations should be defined?

Start with stakeholders

Power management interface stakeholders

Power management interface stakeholders

- **Anyone wanting binary portability across:**
  - Different platform hardware
  - Hypervisors
  - Emulators

Power management interface stakeholders

- **Anyone wanting binary portability across:**
  - Different platform hardware
  - Hypervisors
  - Emulators

- **Hardware power management engineers**

Power management interface stakeholders

- **Anyone wanting binary portability across:**
  - Different platform hardware
  - Hypervisors
  - Emulators

- **Hardware power management engineers**

- **Software power management engineers**

Why define new PM operations?

So why not reuse PM operations from previous platform specifications?

Why define new PM operations?

So why not reuse PM operations from previous platform specifications?

- **The hardware is more sophisticated**

Why define new PM operations?

So why not reuse PM operations from previous platform specifications?

- The hardware is more sophisticated
- The software is more sophisticated

Why define new PM operations?

So why not reuse PM operations from previous platform specifications?

- The hardware is more sophisticated
- The software is more sophisticated
- Some stakeholders haven’t had input in the past

Why define new PM operations?

So why not reuse PM operations from previous platform specifications?

- **The hardware is more sophisticated**
- **The software is more sophisticated**
- **Some stakeholders haven’t had input in the past**
- **RISC-V “big tent” philosophy**

Why define new PM operations?

So why not reuse PM operations from previous platform specifications?

- **The hardware is more sophisticated**
- **The software is more sophisticated**
- **Some stakeholders haven’t had input in the past**
- **RISC-V “big tent” philosophy**

Learn (and borrow) from the past. Build something better!

An aside about RISC-V philosophy

“Big tent”

- The goal is to create recommendations and reference specifications, not mandatory requirements

An aside about RISC-V philosophy

“Big tent” approach

- The goal is to create recommendations and reference specifications, not mandatory requirements
- Example: if someone wants to take a 2010-era SoC with a MMIO platform interface and replace the proprietary CPU cores with RISC-V cores, we cannot (and do not wish to) compel the use of the RISC-V PM specification

An aside about RISC-V philosophy

“Big tent” approach

- **The goal is to create recommendations and reference specifications, not mandatory requirements**
- **Example: if someone wants to take a 2010-era SoC with a MMIO platform interface and replace the proprietary CPU cores with RISC-V cores, we cannot (and do not wish to) compel the use of the RISC-V PM specification**
- **Example: if someone wishes to build a system with no platform firmware, that is fine too**

So what should be defined?

- **Some people think it should be very minimal**
  - Hotplug and nothing more

So what should be defined?

- **Some people think it should be very minimal**
  - Hotplug and nothing more
- **Some people think it should be maximal**
  - No MMIO devices at all

So what should be defined?

- **Some people think it should be very minimal**
  - Hotplug and nothing more
- **Some people think it should be maximal**
  - No MMIO devices at all
- **This talk proposes that we will start small and build up**

So what should be defined?

- **Some people think it should be very minimal**
  - Hotplug and nothing more
- **Some people think it should be maximal**
  - No MMIO devices at all
- **This talk proposes that we will start small and build up**
- **Support querying the specific PM SBI features that are available**

Starting points

- **PM feature queries**
- **CPU hotplug**
  - CPU hotplug & unplug
  - Query CPU hotplug state
- **Platform reset/shutdown**
- **CPU idle**
  - Maximum wakeup latency/Expected sleep residency
  - Explicit states?
- **CPU suspend**
- **Thoughts?**

Fini