

ACPI AML Usage and ASL Guidelines



Graeme Gregory

4th March 2014

What is "Server Class"



- ▶ Servers should expect that any pin function control is setup by firmware (UEFI for example).
- ▶ Servers should not have complex trees of power resources (Clocks/Regulators/..) and such resources should be abstracted by ASL code.

For the purpose of this presentation we will divide ARMv8 server into two generations.

- ▶ Generation 1 those servers released using only ACPI 5.0 specifications.
- ▶ Generation 2 those servers release after the next iteration of ACPI specification implementing the new proposals.

Generation 1



- ▶ ACPI table must only contain features from version 5.0 of the ACPI specification.
- ▶ Power resources (such as clocks/regulators) must be handled by the firmware and/or ASL code.
- ▶ Muxing for SoC pins must be handled by firmware, kernel will assume pins referenced in ASL have the correct function assigned.
- ▶ CPU power control will be purely by PSCI.

Generation 1



- ▶ ASL will expose _PS0..._PS3 methods to control power state.
- ▶ ACPI will be used for system level power transitions G0...G1
- ▶ ACPI will be used for panel buttons like power via GPIO signalled events.

_PS0..._PS3 Example



```
Device (BTKL)
{
  Name (_HID, "INT3420") // _HID: Hardware ID
  Method (_STA, 0, NotSerialized) // _STA: Status
  {
    If (_OSI ("Windows 2012"))
    {
      If (LEqual (BID, BW2C))
      {
        Return (0x0F)
      }
    }

    Return (Zero)
  }

  Method (_PS0, 0, Serialized) // _PS0: Power State 0
  {
    And (GLOA, 0x7F, GLOA)
  }

  Method (_PS3, 0, Serialized) // _PS3: Power State 3
  {
    Or (GLOA, 0x80, GLOA)
  }

  Method (PSTS, 0, NotSerialized)
  {
    Return (RDGP (0x57))
  }
}
```

_PS0..._PS3 Example 2



```
Method (_PS0, 0, NotSerialized) // _PS0: Power State 0 {
    If (ISOP ()) {
        If (DGOS) {
            \VHYB (0x02, 0x00)
            Sleep (0x64)
            \VHYB (0x00, 0x01)
            Sleep (0x0A)
            \VHYB (0x0A, 0x01)
            Sleep (0x64)
            \VHYB (0x02, 0x01)
            Sleep (0x01)
            \VHYB (0x08, 0x01)
            Store (0x0A, Local0)
            Store (0x32, Local1)
            While (Local1) {
                Sleep (Local0)
                If (\LCHK (0x01)) {
                    Break
                }

                Decrement (Local1)
            }

            \VHYB (0x0A, 0x00)
            \VHYB (0x04, 0x00)
            \SWTT (0x01)
            Store (Zero, DGOS)
        } Else {
            If (LNotEqual (VSID, 0x220017AA)) {
                \VHYB (0x04, 0x00)
            }
        }
    }
}
```

Generation 1



Drivers for IP blocks shared with non server platforms.

- ▶ Drivers must operate with "dummy" regulator, any attempt to get a regulator will be replaced with "dummy" (Already Upstreamed - Mark Brown).
- ▶ Drivers must be able to operate in default state without a defined clock. (Still to be investigated).
- ▶ Drivers must operate from sensible defaults and not require fields from the DT. There is no standard method of inserting this data in ACPI.

Generation 2



- ▶ UEFI ACPI Working Group exists and is discussing proposals for the next release of ACPI specification.
- ▶ OEMs/SoC manufacturers should join UEFI and join working group to steer next ACPI specification to better support ARMv8.
- ▶ UEFI rules mean we can only discuss new proposals with other UEFI members. This limits public prototypes.

Generation 2



- ▶ Any move to make Power Resources produce-able/consume-able in ASL resource must be backed by accepted proposal to ACPI WG.
- ▶ Any advanced CPU topology/power control must be backed by accepted proposal to ACPI WG.
- ▶ This follows for any other new features.

ACPI 5.0+ Features



- ▶ linking ACPI and PSCI power state control.
- ▶ representing CPU topology
- ▶ representing performance points (cpufreq)

- ▶ Linaro's current ACPI work in kernel is leading to the implementation of Generation 1
- ▶ Linaro is involved with the ACPI WG for Generation 2.
- ▶ Linaro's aim is to implement Generation 1 features and build on this to tune and implement Generation 2 specifications.

Other Sessions



- ▶ ACPI Upstreaming 11:15am Loulan 4003-4004
- ▶ ACPI Power Management 2:55pm Loulan 4103-4104