Designing an ACPI/SBS Laptop

By Derek Sip,
Hitachi PC Corporation

Designing an ACPI/SBS Laptop

- In this section, we will focus on system implementation & precautions in designing ACPI/SBS compliant architectures in portable systems. Instead of reiterating SBS specifications and it's advantages, we will discuss different approaches and implementations

Smart Battery Overview

- System Management Bus (SMBus) uses two-wire (I2C like) as it back bone
- SMBALERT# is an optional signal most portable systems use.
  - It is a wired-OR line used by slave-only devices.
  - Most SMBus devices (e.g. thermal sensor) use it

Smart Battery Overview

- The host cannot directly reset slave devices
- The host can time out slave devices by purposely violating TLOW:MEXT more than 10msec within one byte
- SMBSUS# is an optional signal that is usually not used in laptops.
  - SMB devices are typically powered during suspend

Smart Battery Overview

- Devices know the SMBus is alive if both SMBCLK & SMBDATA lines are high
- I2C bus logic levels are relative to Vcc
- SMBus logic levels are fixed at 0.6 and 1.4 V
  - Relative to Vcc = 2.0 volts

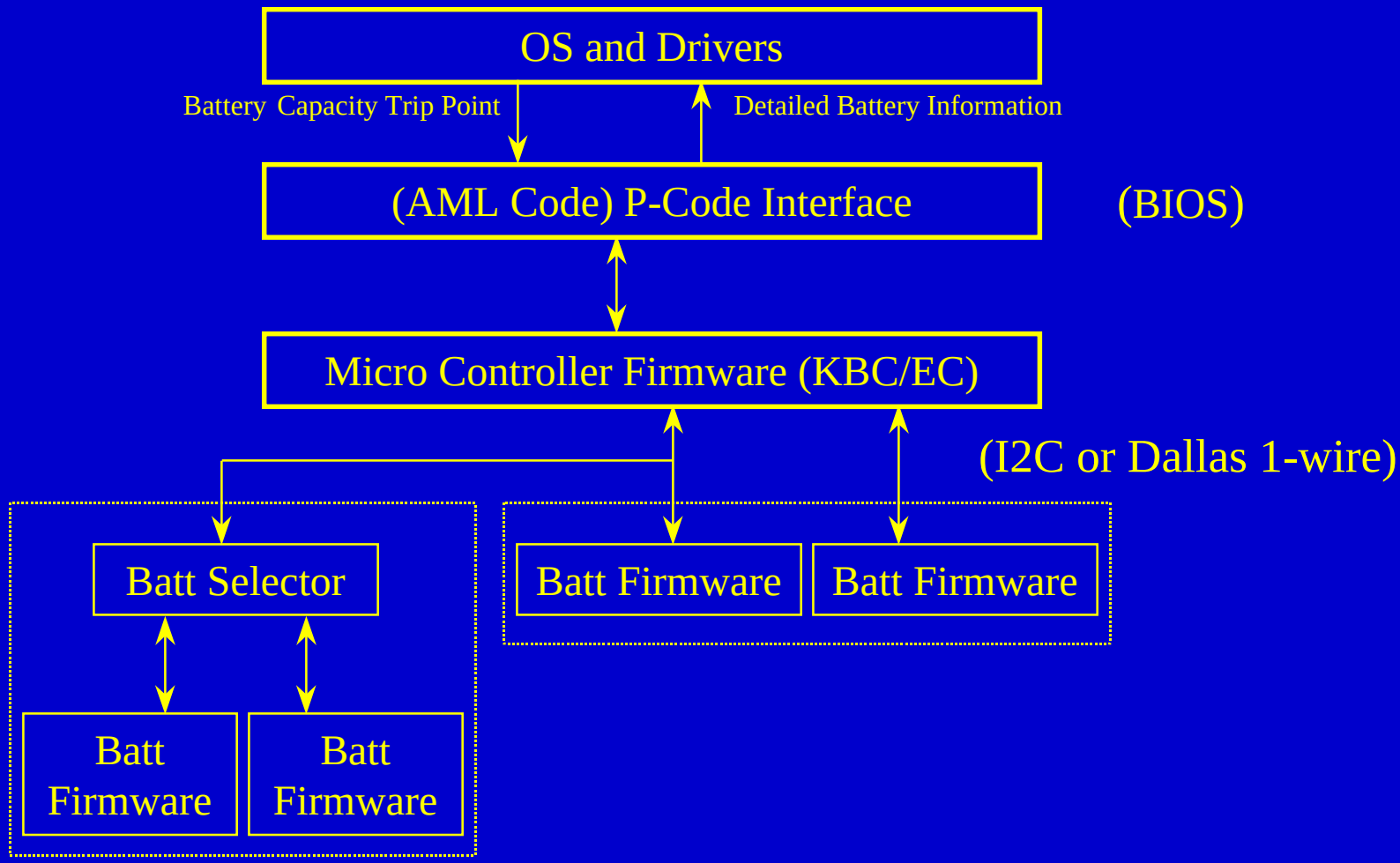
ACPI Requirements for Battery Subsystem

- Battery management policy moves from APM BIOS to ACPI compatible OS.
  - The OS is warned when the low battery point and critical low battery point are reached.
    - H/W shut off recommended
  - The OS queries the battery for remaining capacity
  - The OS does not control charge stopping
    - Done with message from battery to charger

ACPI Requirements for Battery Subsystem

- An ACPI compatible battery system needs to have one of the following:
  - (1) Smart Battery Subsystem Interface
  - (2) Control Method Interface

Communications Model for CM Battery System



Pros & Cons of CMBatt

- CMBatt Pros:
  - Complete control of system design since we can guarantee that only proprietary batteries can be used in the system
  - Fewer commands need to be supported
  - Lack of standard battery selector to support more than two batteries

Pros & Cons of CMBatt

- CMBatt Cons:
  - Large system development cost and time
  - Becomes obsolete when new standard or technology comes out
  - Incompatible with industrial standard
  - Hard to support multiple battery technologies

ACPI SBS System Implementation

- An ACPI compatible Smart Battery System consists of the following:
  - (1) An SMBus host controller (0x8)
  - (2) At least one smart battery (0xB)
  - (3) A smart battery charger (0x9)
  - (4) A smart battery selector (0xA)
    - Optional with one battery system

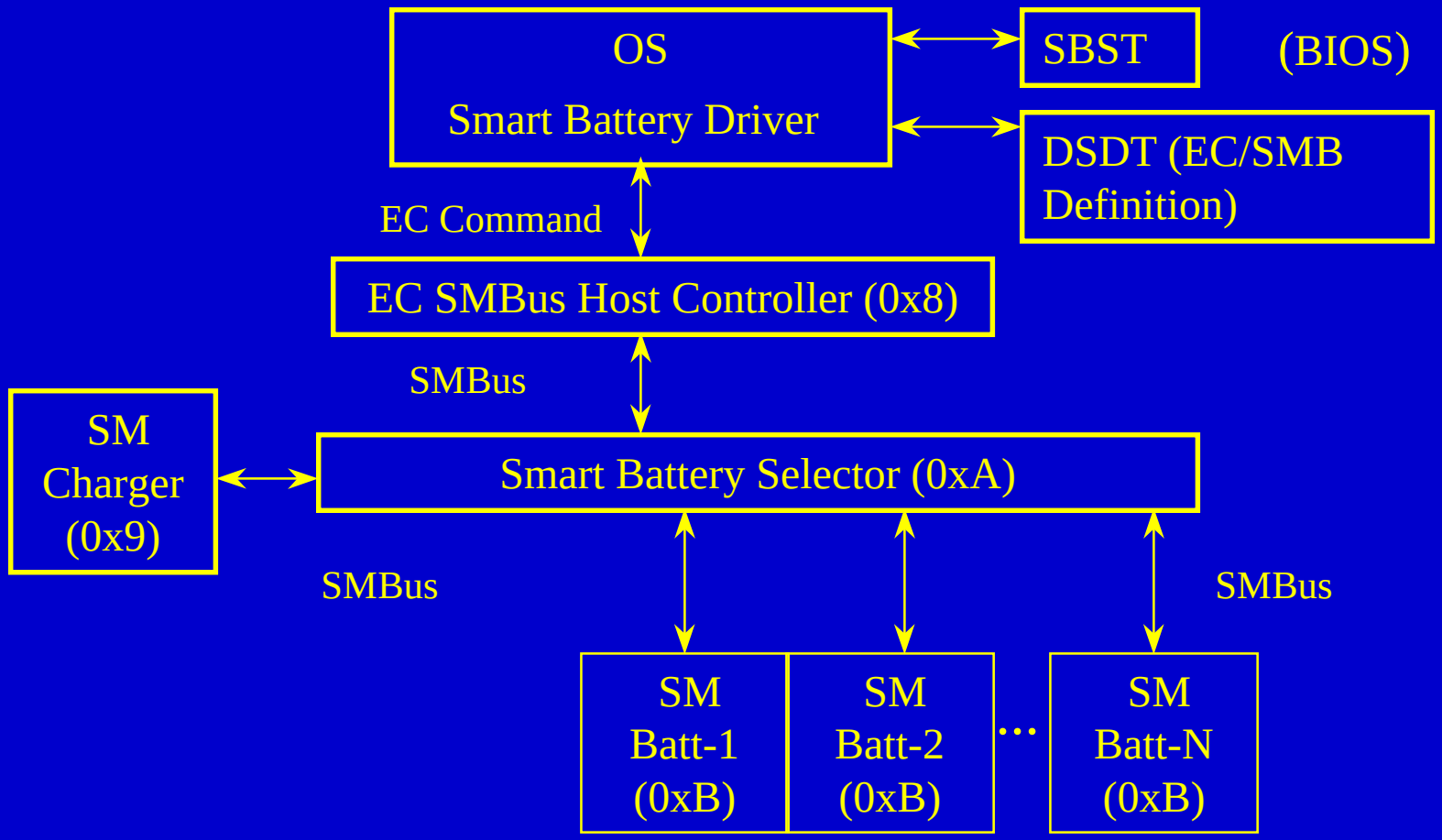
ACPI SBS System Implementation

- Sub-notebooks, Typically use only one battery.
  - No smart selector required.
  - The smart battery charger will also perform the selector function, including notifications for status change
- For low end and midrange notebooks, two battery support is usually required.

ACPI SBS System Implementation

- For high end notebooks, support for more than two batteries may be required.
  - However docking stations can be used for charging and do not need to be on the same SMBus
- In the industry today, there is no simple standard battery selector solution that supports more than two SM batteries

Communications Model for ACPI/SBS System



Communications Model for ACPI/SBS System

ASL code for this communication model for N batteries will look like this.

```
Device (Ecx) {                                          // EC
        Name (_HID, EISAID (“PNP0C09”))
        Name (_CRS, ResourceTemplate() {                // EC IO Port
                IO(Decode16, ECIO, ECIO, 0, 2) } )
        Name (_GPE,1)
        Device (SMBy) {                                 // SM Bus
                Name (_HID, “ACPI0001”)                 // Host Controller
                Name (_EC, 0x0931)                      // EC offset & Query
                Device (SBSz) {                         // SM Battery
                        Name (_HID, “ACPI0002”)
                        Name (_SBS, 0xN)  // N SM Batteries
                        } } }
```

Smart Battery System Advantages

- More accurate battery information
- Increase battery life
- Enable green conditioning for NiMH battery
- Open spec enables supporting different chemistries and multiple sources
- Off load more development effort to battery vendors

Smart Battery System Advantages

- Allowing non-power related devices in the SM Bus to enable the system to partition more nicely, like putting LM75 for battery related thermal sensor in SM bus instead of another generic bus

Smart Battery System Disadvantages

- According to our experience, I still do not trust the SM battery data at this stage.
  - Some batteries still have to twist the firmware a little bit to make it work with the systems
- Since SMBus is a common bus and all the batteries have the same address, debugging for multiple batteries environment is getting difficult.

Precautions in Designing SBS into Portable Systems

- Lion charge is basically constant voltage (e.g 12.3V) with current foldback.
  - Full charge detected with current
- Battery may rely on FULL_CHARGED status (0x0020) to inform battery fully charged and then set to 100mA.
- Others may set it at 150ma and rely on OVER CHARGED_ALARM (0x8000) to signal full charge.
- Systems using the first system may not fully charge the battery

Precautions in Designing SBS into Portable Systems

- The OS will poll the SM bus status in a fixed interval. For master only EC (e.g. M38813Mx), it will also poll the SM batteries in a fixed interval. In this early stage, some batteries cannot handle bursting of polling.

Conclusions

- Development effort will greatly be reduced if we only implement SBS instead of both SBS and CM interface
- SM batteries are the major component that we cannot control in system point of view since user can plug in any unqualified battery
- Somebody has to control SM battery compliance.
  - Should we have an independent test lab for qualification