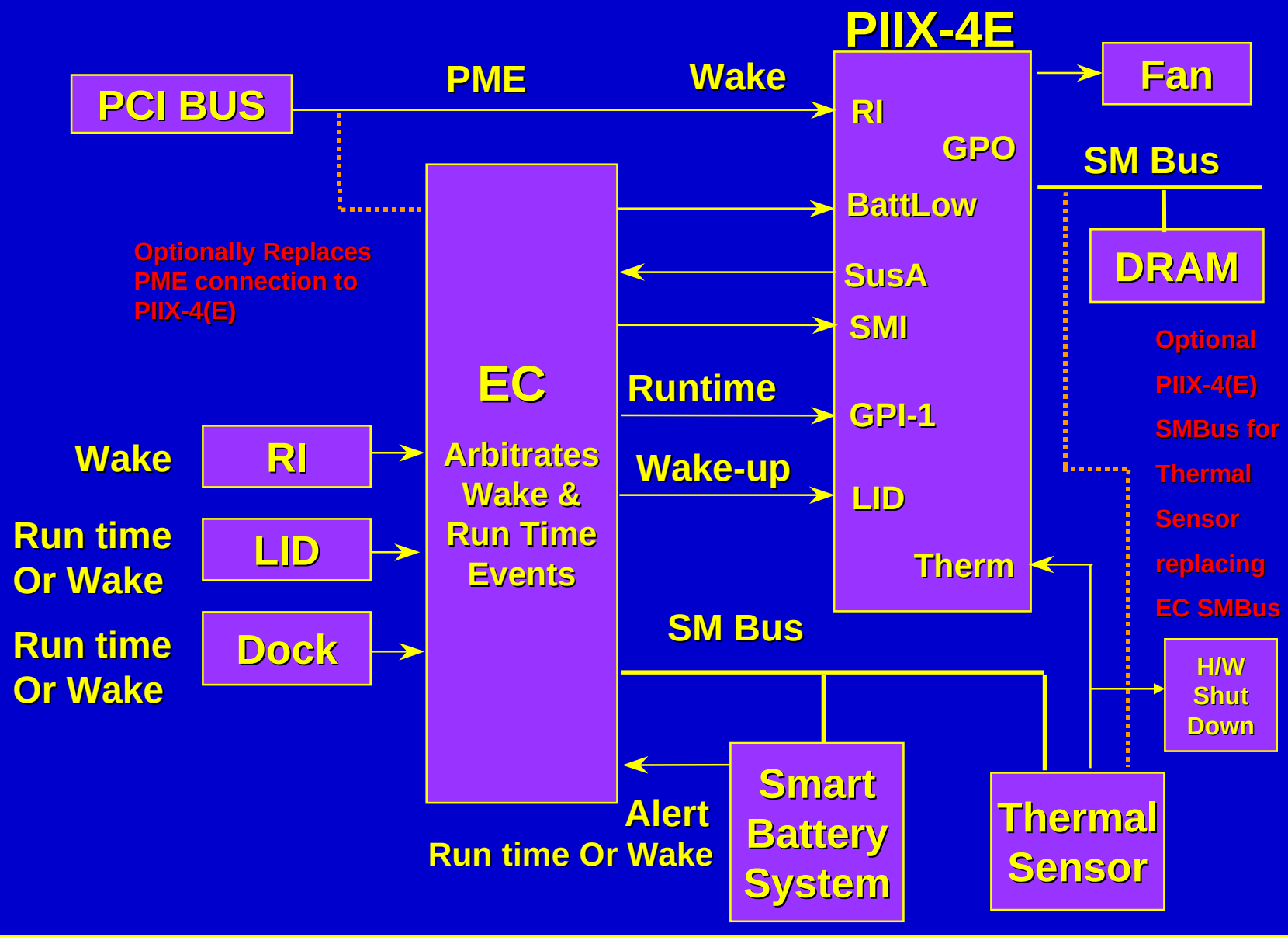
EC/PIIX-4 Architecture in ACPI systems

Phil Mummah
Intel Corporation

Agenda

- Embedded controller & Chipset
  - How connected to PIIX-4(E)
- PME
- Battery System
- Thermal architecture

EC/PIIX-4E Recommended Architecture



Embedded Controller

- EC is optional in a ACPI system
- Recommended for mobile systems
- Can use control methods to access EC operation regions to implement Control Method batteries
  - EC does not have to have ACPI SMBus Interface
    - Batteries can be compliant with SBS specification
  - Query command does not need to be supported

Embedded Controller
With ACPI Compliant SMBus I/F

- EC must support query commands
- EC must support SMBus host controller I/F registers
  - Requires 40 registers
- No control methods required
- OS driver can access battery system directly

Embedded Controller Events

- **ACPI compliant EC uses runtime GPE to communicate with EC driver**
  - **Only used in S0**
  - **EC needs H/W mechanism to know system power state**
    - **S/W (Control methods) can NOT be used to determine system power state**
  - **Signals:**
    - **Transactions**
    - **Query events**
    - **Operation region access**
  - **Must be separate GPE from all other GPEs asserted by EC**
- **All other events are simply GPEs**

Embedded Controller Events

- Terminology "Wake events" are *device* wake events
  - May be asserted in any system state (S0-S4)
  - Wake devices must have _PRW in name space
    - Query events cannot wake system
  - If the system is asleep, a *device* wake event may wake the system
    - Only wakes if enabled by a driver

Run Time Battery Event Process

- Battery asserts alert to EC
- EC will assert run time event to PIIX-4E
- SCI is generated
- OS will detect it is an EC event by comparing with ASL table
  - DSDT table list event owners
- OS send query command to EC

Run Time Battery Event Process

Continued

- EC will return query number for battery event
  - Event numbers are defined in DSDT
- OS will run smart battery driver
  - OS will read SMBus directly and take appropriate action

Wake Up Battery Event

- Battery assert alert to EC
- EC asserts wake up GPE of PIIX-4E
- PIIX-4E exits suspend and enters resume state
  - Execution starts at power-on reset vector for wake up from S2 - S4 sleep states
- BIOS restores system back to POST state and passes control to OS

Wake Up Battery Event

Continued

- OS detects asserted (wake up) GPE and runs EC's _Lxx
- _Lxx accesses EC IO space to find the waking device(s).
- _Lxx issues notification to OS on behalf of the waking device(s).
- OS knows the waking device(s) and invokes the smart battery driver

Embedded Controller

- SMI can be used for hot keys
  - Can be used for anything not defined in ACPI Specification
    - e.g. Back light control
  - **Do not use hot keys for system power state changes**
    - Power button & User I/F used to request power state changes
      - OS changes power states
      - Can NOT change system power state without using OS mechanism
- Use drivers & Apps instead of SMI

PME

- Recommended directly connected to GPE input
  - OS will scan and clear PME_STS while asserted
- Alternate design: PME can go through Embedded Controller
  - Latency of EC must be short
    - OS will continue to scan PCI bus until PME cleared
    - Should respond as if EC is passing PME signal through (emulate a wire)

Thermal Control

- Thermal control should follow ACPI thermal model for mobile systems
  - OS makes cooling decisions
  - Performance and quiet trade off made by OS

Thermal Control

Continued

- Multiple temperature trip points recommended for mobile
  - Passive (throttling)
    - One Passive zone
  - Active
    - Multiple active zones available
  - Critical
- Alternate Design: Can have fan directly controlled by thermal sensor
  - Don't include thermal zones in name space

Thermal SCIs

- Passive trip point should generate SCI
  - OS will throttle system
  - OS will read thermal sensor every sampling period as defined in _TSP
  - OS will stop throttling and stop reading thermal sensor when temp falls below passive trip point
    - Hysterisis can be put in by passive control method

Thermal SCIs

Continued

- If passive trip point is below active trip points
  - Active trip point may generate SCI
  - OS will detect crossing active trip point while reading thermal sensor during sampling period
  - Active control method will be run
  - Throttling will continue until temperature falls below passive trip point

Thermal SCIs

Continued

- If passive trip point is above active trip points
  - Active trip points must generate SCI

Thermal Sensor

- Variations of implementation
  - Thermal sensor behind the EC
    - EX: LM-75
      - Has one trip point
      - EC can emulate multiple trip points
  - Thermal sensor connects to SMBus on PIIX-4(E)
    - Multiple trip point sensor
      - Ex: Maxim 1617

Thermal Sensor

- Software Critical Temp
  - OS will shut down system
  - Application data not saved
- Hardware Critical Temp
  - Higher temperature than S/W critical
  - Must turn machine off
    - Does not depend on state of system
    - Does not require s/w to run
    - Don't depend on EC unless has watch dog timer

Summary

- Smart Battery system can share SMBus with other devices
- Embedded Controller needs to be H/W compliant
- Smart batteries can be accessed with control methods or OS smart battery driver
  - Firmware can be upgraded at a later time