PCI Express® 1.1 Protocols & Software

Beijing	April 12, 2005
Tokyo	April 15, 2005
Taipei	April 18, 2005

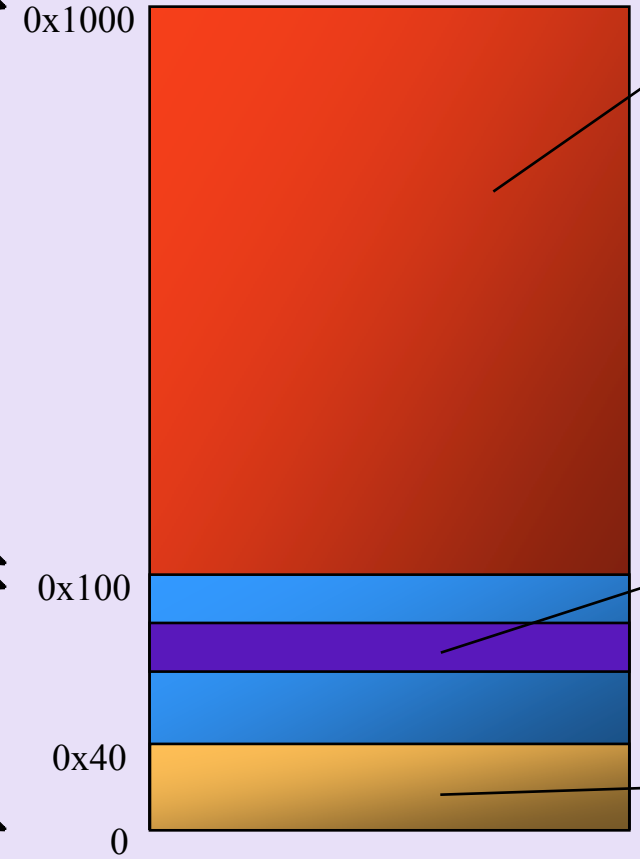
PCIe Protocol/Configuration Basics

- Configuration Space
- Interrupts
- Power Management
- Power Negotiation
- Boot Considerations

Configuration Space

PCI Express Extended Configuration Space ***(Not available on legacy OS)***

PCI-Compatible Configuration Space (Available on PC-compatible platforms with legacy OS via CF8/CFC)



Extended configuration space for PCI Express parameters capabilities (First extended capability begins at offset 0x100)

PCI Express Capability Structure

PCI 2.x Compatible Configuration Header

Configuration Space

- PCI-Compatible and Extended Config Space

- Do not rely on Extended Config space to be available in legacy environments
  - ✓ **Extended Configuration Space access may not be available in legacy OS scenarios**
  - ✓ **If access to Extended Configuration Space elements is really needed, design for aliasing elements through a BAR or PCI-Compatible Configuration Space region**

- Note that registers critical to device functionality are all located in PCI-Compatible Configuration Space
  - ✓ **PCI Express Capability Structure located below 256 bytes**

Configuration Space

- Specification defined capabilities are for system software use
  - ✓ **Device-specific software should rely on OS services for access to these registers**
  - ✓ **Writes to spec-defined capabilities are not recommended for software other than system software (firmware / OS)**
    - Improper use can cause upgrade problems during OS migration
- Recommend that device-specific registers be located in BAR regions

Interrupts

- PCI Express supports three interrupt mechanisms:
  - ✓ **INTx: level triggered**
  - ✓ **MSI: edge triggered**
  - ✓ **MSI-X: edge triggered; newly added via ECN**
- INTx, MSI, & MSI-X are mutually exclusive
  - ✓ **Enabling MSI or MSI-X disables INTx**
  - ✓ **SW is prohibited from enabling MSI & MSI-X concurrently**
  - ✓ **MSI/MSI-X are not controlled by (INTx) interrupt disable bit**
  - ✓ **MSI/MSI-X messages are memory write requests and can be disabled by clearing the BME (Bus Master Enable) bit**
  - ✓ **MSI/MSI-X/INTx interrupts can only be signaled in D0 state**

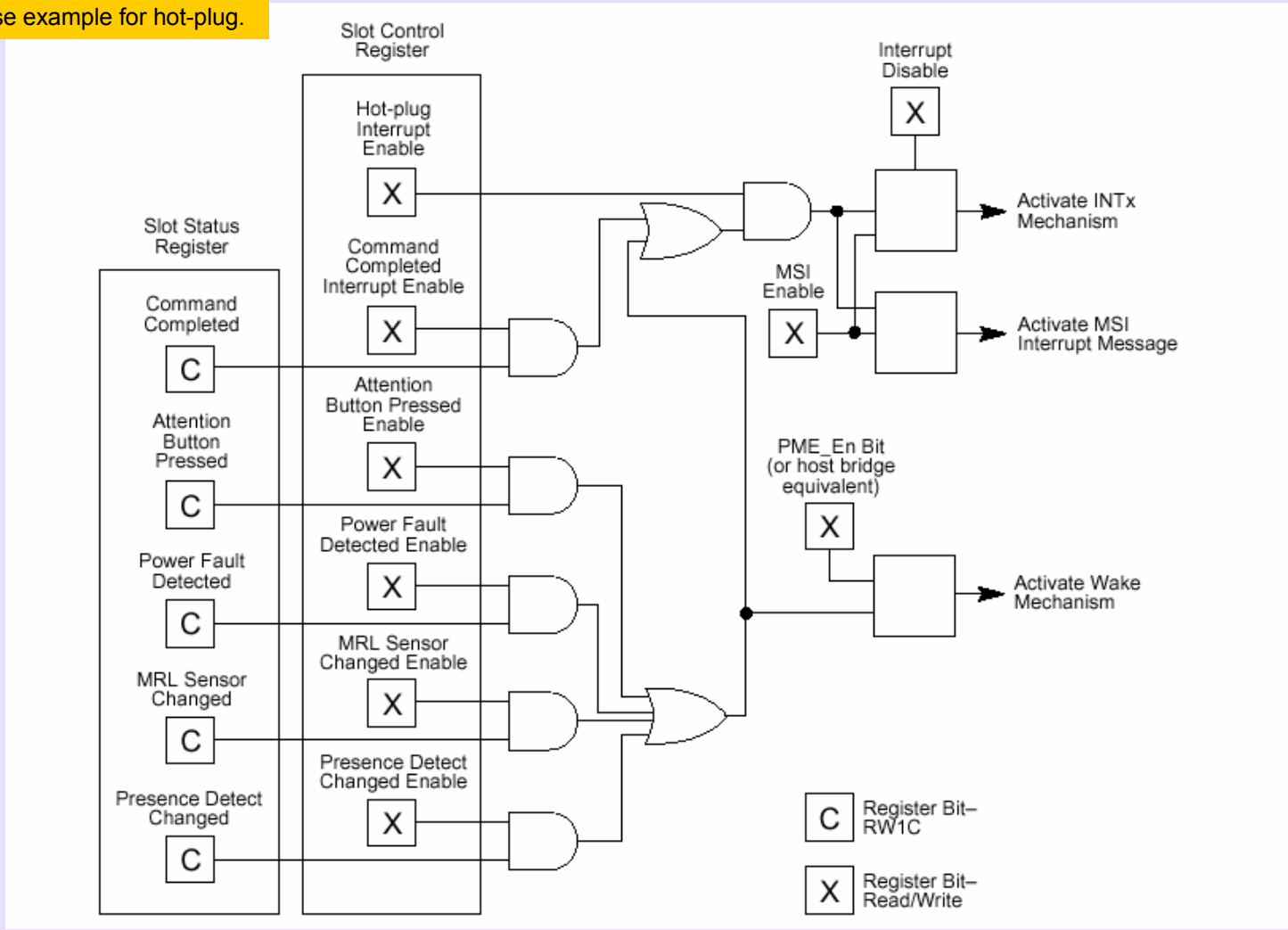
Interrupts

- Level triggered interrupts can result in interrupt storms
  - ✓ **Especially high risk in virtual wire scenarios if de-assert messages not sent correctly**
  - ✓ **Be sure to:**
    - De-assert INTx in low-power states
    - De-assert INTx when source becomes masked
    - De-assert INTx when interrupts become disabled
  - ✓ **Asserts / De-asserts must be sent in pairs**

- Switches must synthesize de-asserts as necessary
  - ✓ **For example, if attached device is surprise removed**

Interrupts: Design Example

NOTE: Example intended to illustrate device interrupt logic. Do not use example for hot-plug.



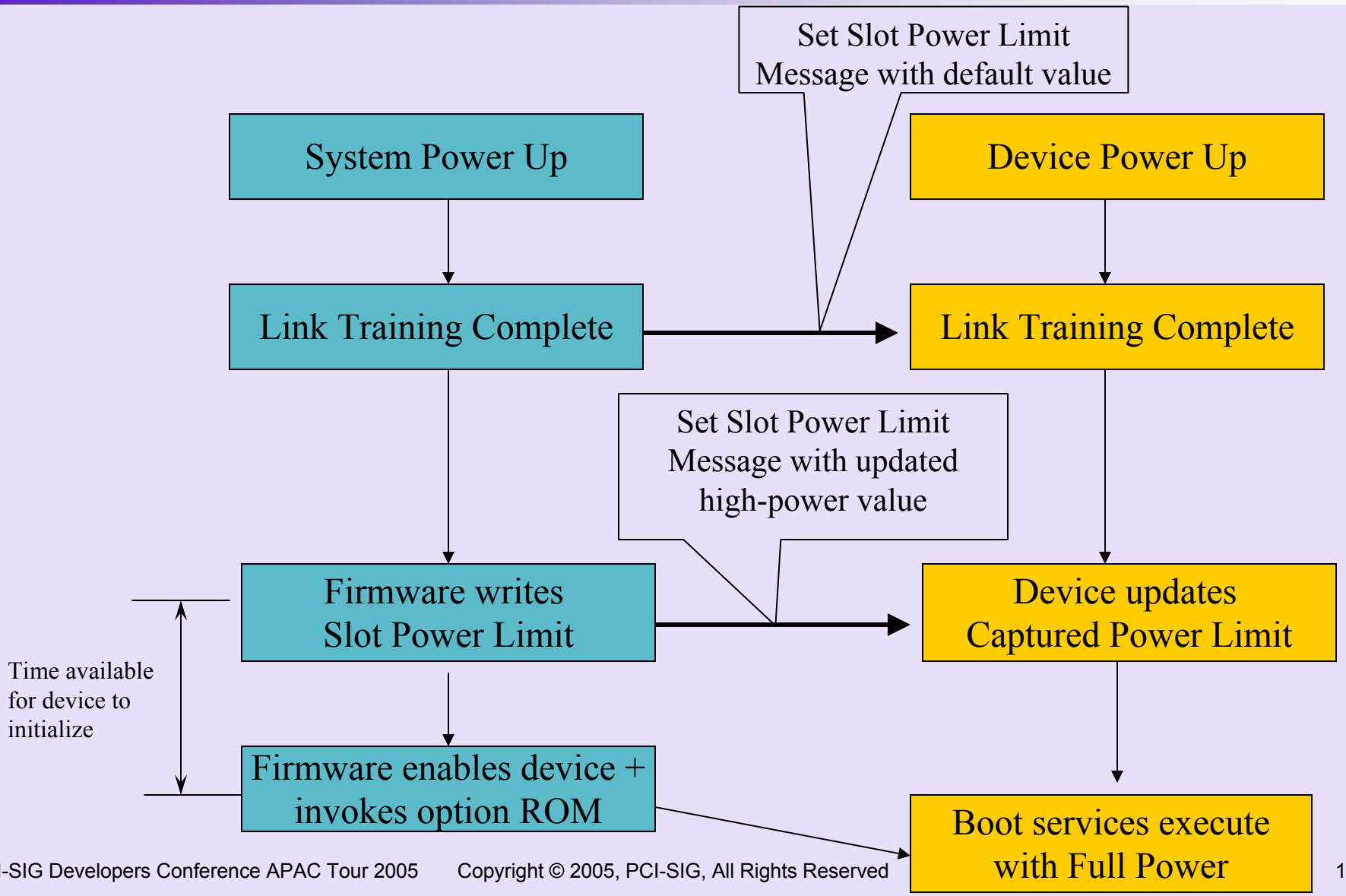
Power Management

- Devices cannot source memory on IO requests when in a non-D0 state
  - ✓ **Remember to de-assert any pending INTx interrupts when transitioning out of D0 to a low-power state**
- PME_Turn_Off message can be received at any time
  - ✓ **Not just in non-D0 states**
  - ✓ **Simply indicates to device that power and clocks are going to be removed**
    - For example, can be received on system shutdown
- PME sequence is a two-stage process
  - ✓ **Two-stage process:**
    - Wakeup is separated from request for service
  - ✓ **Wakeup is responsible for getting clocks and power**
  - ✓ **Service is requested through PME message**

Power Negotiation

- Slot Power Notification mechanism
  - ✓ **Flexible and scalable mechanism to notify a PCI Express adapter of additional available slot power**
- Conditions that trigger the message:
  - ✓ **On a Configuration Write to the Slot Capabilities register (when the Data Link Layer reports DL_Up status)**
  - ✓ **Anytime when a Link transitions from a non-DL_Up status to a DL_Up status**

Power Negotiation: Event Flow



Power Negotiation Considerations

- Devices strongly encouraged to provide minimum level of functionality within the CEM specification power
  - ✓ **Driver or Option ROM can detect available power through captured power limit registers**
- Power Budgeting Extended Capability can allow flexibility in power redistribution
  - ✓ **Implementation specific dynamic power redistribution**
  - ✓ **Devices are strongly encouraged to implement this capability**

Other Considerations and Robustness Guidelines

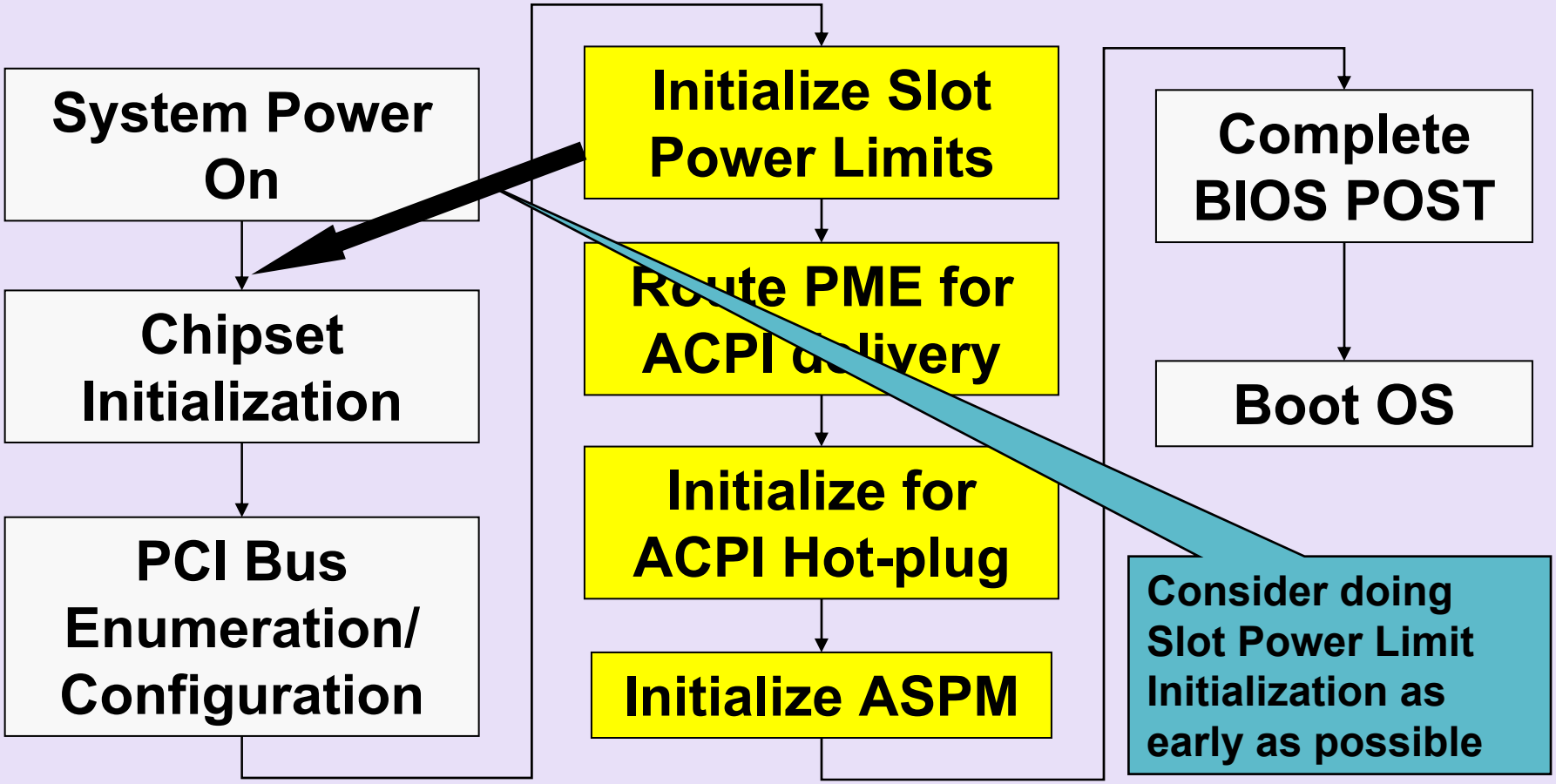
- Don't rely on access to extended configuration space for PCI compatible device operation
- IO BARs can be closed by system software for native PCI Express devices
  - ✓ **Device may support an IO BAR for booting in legacy environments**
  - ✓ **But… Native PCI Express device must alias IO resources through MMIO as per PCI Express specification**
- Do not hang on unexpected TLPs
  - ✓ **For example, power changes or hot-plug messages**
- Do not assume that TLPs can only arrive in certain states
  - ✓ **For example, PME_Turn_Off**
- Not sending de-asserts correctly for INTx can cause interrupt storms

Legacy software compatibility and transition strategy to PCIe-aware software environment must be considered

Firmware Guidelines

- Firmware must correctly communicate configuration space information to operating system
- Firmware support required for legacy compatible support in the absence of a PCI Express-aware OS
  - ✓ **ExpressCard* hot-plug through ACPI**
  - ✓ **ACPI PME handling**
- Firmware must implement functionality for transition of services to PCI Express aware OS
  - ✓ **Native OS PME handling**
  - ✓ **Native OS hot-plug**
- Firmware support required for correct power negotiation
  - ✓ **Power budgeting can enable firmware to optimize power distribution at boot**

Firmware Initialization Example



Firmware plays a key role in enabling new PCIe features

PCIe Protocol/Config Summary

- Legacy software compatibility and transition strategy to PCI Express-aware software environment must be considered at all times
- MSI/MSI-X have major advantages over INTx, but correct interrupt handling to avoid loss of edge-triggered interrupts is not intuitive
- Implement MSI-X instead of MSI when multiple vectors are merited
- Firmware plays key role in enabling new PCI Express features


PCI
SIG™