Managing Power in Today's Embedded ASIC or SoC Designs

Kishore Mishra
President and CEO
ASIC Architect

CC Hung
Product Line Manager
Mentor Graphics

Outline

- Understanding the basics of power consumption
  - ✓ Ingredients of power
  - ✓ How the ingredients can be manipulated to reduce power consumption
- Understanding the basics of power management (PM)
  - ✓ How PM is done at global level
  - ✓ How PM is done at local levels
- Applying PM in PCIe® for embedded ASIC or SoC designs
  - ✓ How embedded chip designers are faced with PM challenges?
  - ✓ How different PM approaches can be applied?
  - ✓ How the various PM modes work?
  - ✓ How other methods can be applied for further power saving?
- Conclusion

Basics of Power Consumption

- Power dissipation = Kfcv2. Power is consumed when ..
  - ✓ voltage is being applied - proportional to square of voltage
  - ✓ clock frequency – proportional to frequency
  - ✓ nodes are switching – clock is running
- Power Savings through voltage manipulation
  - ✓ Reduce operating voltage
  - ✓ Have separate power (voltage) wells, and at times, cut off supplies to most except one for wake-up logic
- Power Savings through clock manipulation
  - ✓ Operate on a lower clock frequency
  - ✓ Stop clock at times on local as well broader level

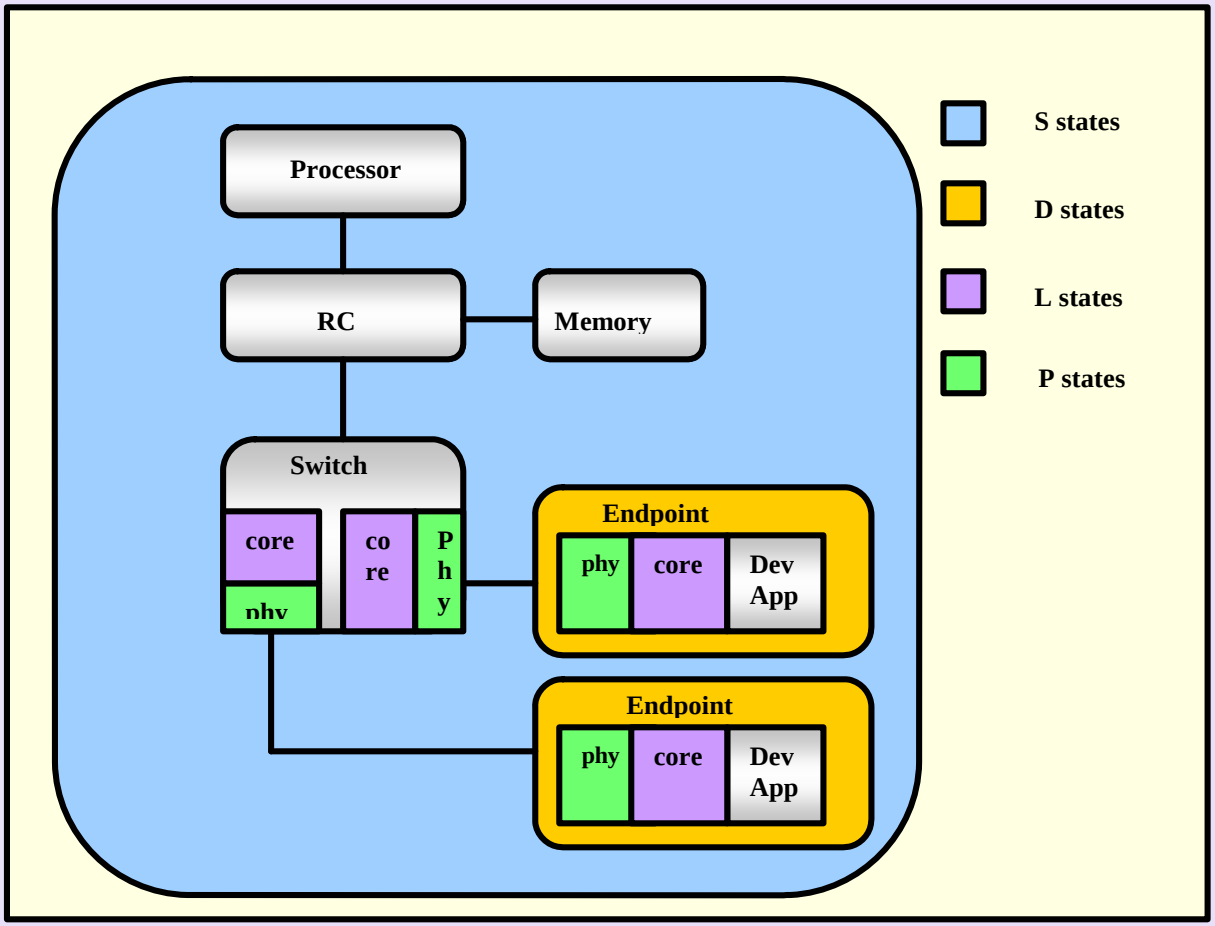
Basics of Power Management

- PM is a coordinated effort among system components
  - ✓ OS, pm drivers, platform power/clock manager, chipset and device
  - ✓ System power states S0, S1, S2, S3, S4, S5
  - ✓ Device power states D0, D1, D2, D3
  - ✓ Link States L0, L0s, L1, L2, L3
- System States (S0, S1, S2, S3, S4, S5)
  - ✓ Has system wide impact
  - ✓ Many components are powered down
  - ✓ Coordinated effort among system components
- Device states (D0, D1, D2 and D3)
  - ✓ Impacts at a device level
  - ✓ OS, power management driver, device drivers are involved
  - ✓ PM is also done at local levels
- Link states (L0, L0s, L1, L2, L3)
  - ✓ Impacts the link
  - ✓ PM is done at local levels
  - ✓ CLKREQ# clock control

PM States Relationship Table

ACPI System States	Device States	Link States	Phy States	Comments
S0	**D0**	**L0**	**P0**	▪ Fully working states. ▪ Power is on ▪ Clocks running
		L0s (ASPM)	**P0S**	▪ ASPM (without software interaction) ▪ lower savings and lower exit latency ▪ Clocks running
		L1 (ASPM)	**P1**	▪ ASPM (without software interaction) ▪ medium savings and medium exit latency ▪ Clock running without clkreq# ▪ Clock off with clkreq# on
S1 -Desktop	**D1, D2, D3Hot**	**L1 (Software)**	**P1**	▪ Similar to L1ASPM ▪ Device application logic may be in sleep
S1 -Mobile	**D1, D2, D3Hot**	**L2**	**P2**	▪ Starts by putting device into D3hot ▪ Link goes to L2 after completing L2L3 ready protocol ▪ Power may be removed
S3: suspend to memory **S4: suspend to disk**	**D3Cold**	**L2**	**P2**	▪ Many components powered down ▪ Auxiliary power is available ▪ Can wake up through WAKE# or BEACON
		L3	**P3**	▪ Auxiliary power is not available ▪ No wake up possible
S5	**D3Cold**	**L3**	**P3**	▪ System completely off

PM States Relationship Picture



PCIe Power Management for ASIC or SoC designs

- How embedded chip designers are faced with PM challenges?
- How different PM approaches can be applied?
- How the various PM modes work?
- How other methods can be applied for further power saving?

How Embedded Designers are Faced with PM Challenges?

- Increasing complexity ASIC or SoC designs
- Embedded Processor – ARM, MIPS, ARC, DSP
- Multiple Clocks
- Multiple Resets
- Multiple Sources of Power
- Hungry for Power Save to its lowest level

How Different PM Approaches Can be Applied?

- Logical:
  - ✓ Adhering the Low Power Design Techniques
- Physical:
  - ✓ Using some of the state of the art tools at the netlist level.
- Functional:
  - ✓ Adhering to the standard and defined protocols
- System:
  - ✓ Understanding the functionality of the system
- What we will cover:
  - ✓ Functional and System Level Power Management

How the Various PM Modes Work

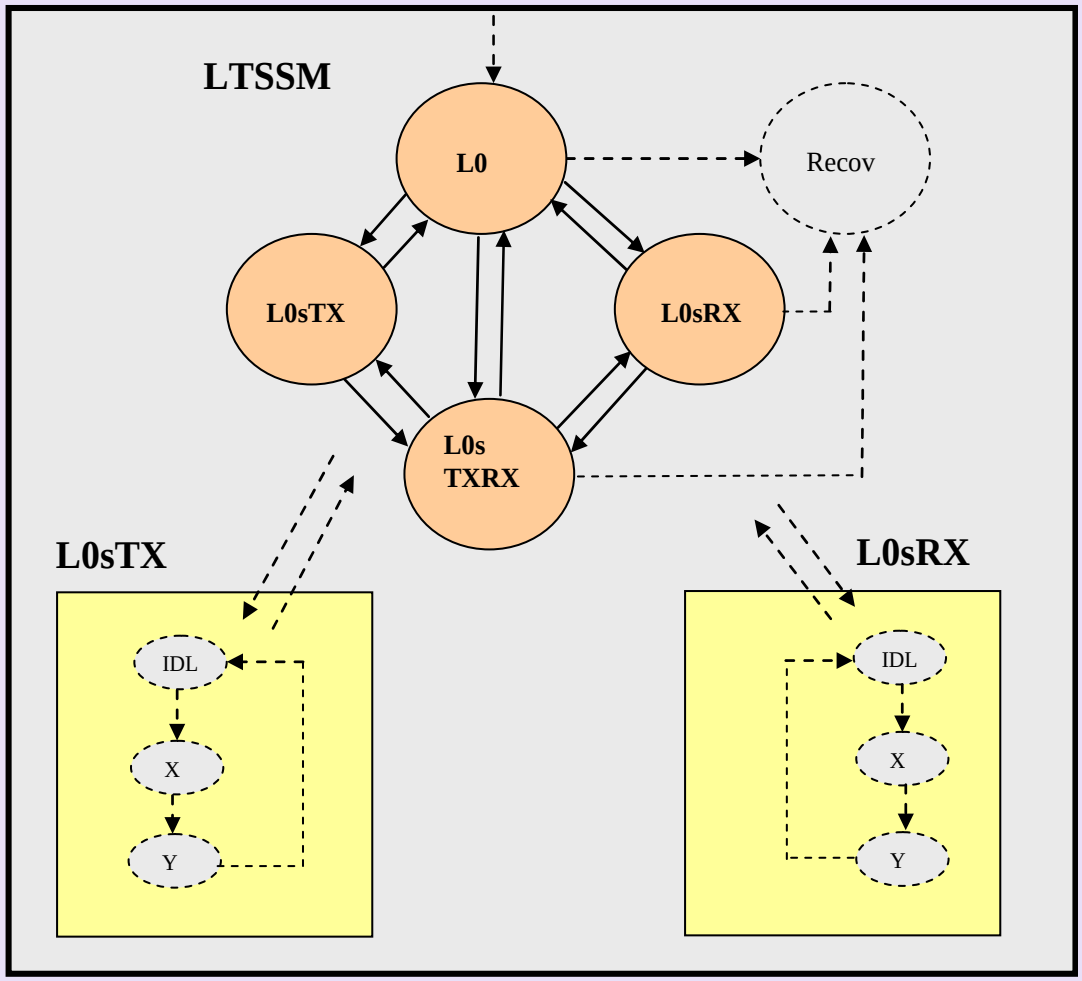
- L0s power state
  - ✓ L0sTX state
  - ✓ L0sRX state
- L1 power state – ASPM initiated
- L1 power state – Software initiated
- CLKREQ# during L1 state
- L2 power state
  - ✓ Waking up through WAKE#
  - ✓ Waking up through Beacon
  - ✓ Power well Isolation
  - ✓ PME# operation

L0s Link Power State

- TX and RX side can go into L0s states (L0sTX, L0sRX) independently
- Low exit latency
  - ✓ Intended for quick entry and quick exit
  - ✓ Exploits short-term opportunities
- 100 MHz reference clock and PLL running
- TLP and DLLP transmission not allowed
- Implementation can be tricky

Suggested L0s Implementation

- LTSSM main state machine handles L0sTX and L0sRX transitions
- L0sTX and L0sRX sub state machines feed into LTSSM state machine



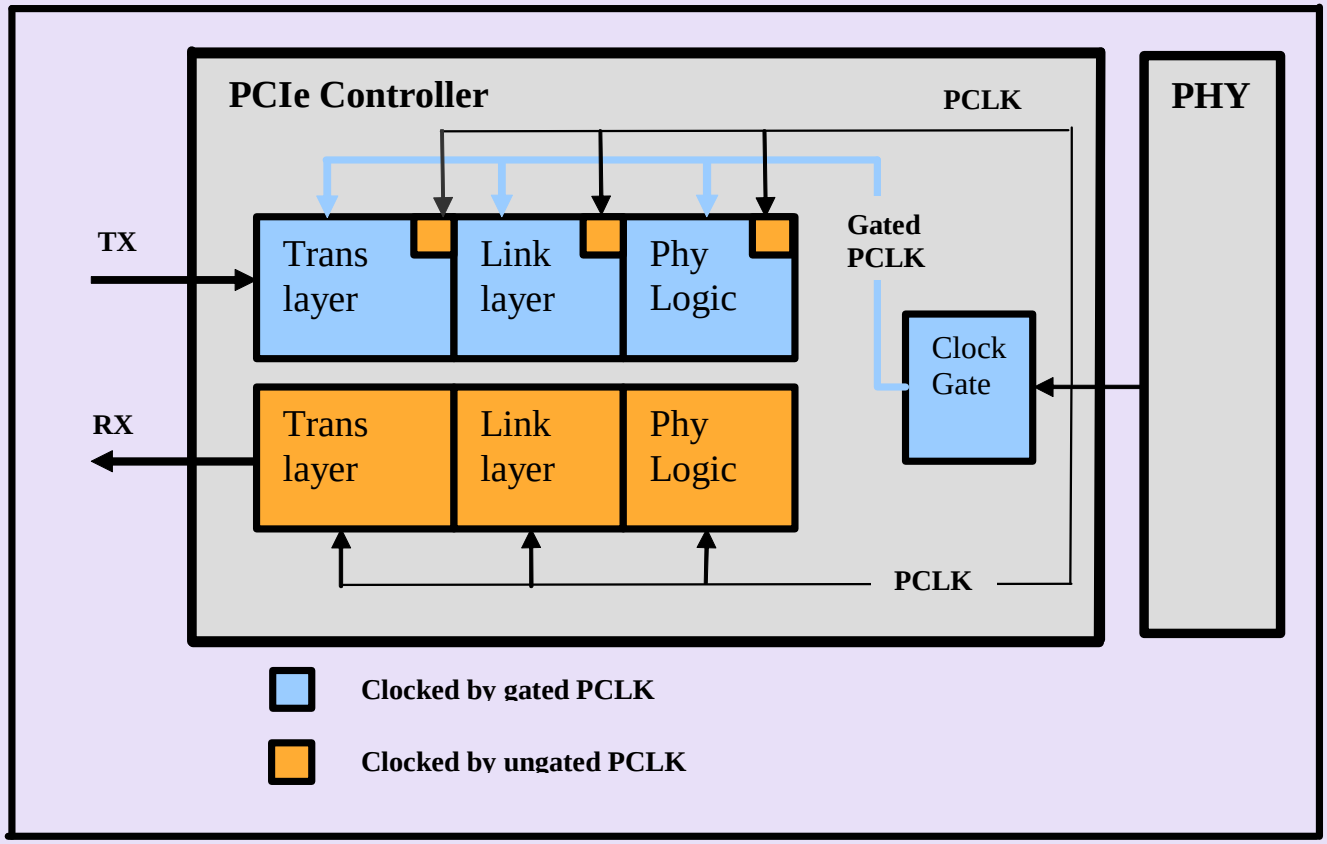
L0sTX Link Power State

- TX goes to tri-state (electrical idle)
- Internal timer tracks Transmit TLP inactivity
  - ✓ Enters L0sTX after timer expiry
  - ✓ Timer not to exceed 7 us
- Make the timer user programmable, instead of hard-coding (suggested implementation)
  - ✓ Device driver can fine tune the L0sTX entry timer
- Exits L0sTX when need to send TLP or periodic FC update
  - ✓ Reset FC update timer every time credit update is sent

Suggested Clock Gating in L0sTX

- Could employ selective clock gating inside PCIe controller
  - ✓ Selective Layers of TX logic can be clock gated
- Ungate clock when exiting L0sTX
- Ungating of clock should be quick to maintain low L0s exit latency

Suggested L0sTX Clock Gate Scheme

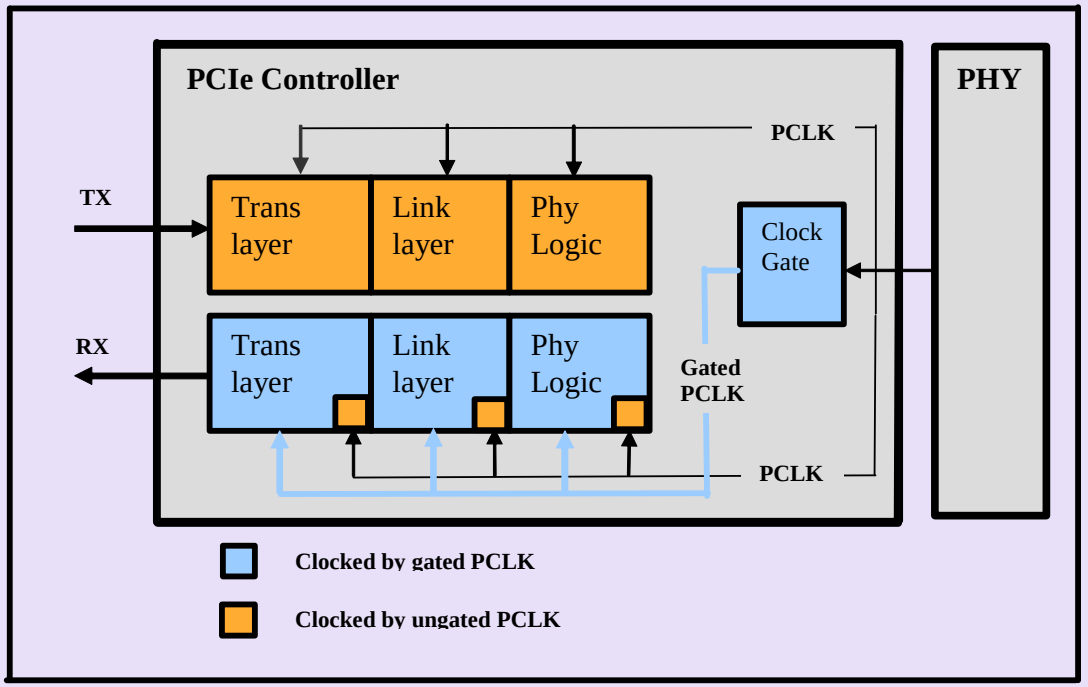


L0sRX Link Power State

- Starts when PCIe controller receives EIDLE set
- Exits L0sRX when PCIe controller receives FTS
- Low exit latency
  - ✓ Exit latency is influenced by number of FTS
    - (1 to 255 FTS)
- Design phy to handle symbol locking with low FTS
- Also make FTS user programmable instead of hard-coding (suggested implementation)

Suggested Clock Gating in L0sRX

- Selective Layers of RX logic can be clock gated
- Ungate clock when receiver exits out of electrical idle condition



L1 ASPM Power State

- When both L0s and L1 are enabled
  - ✓ Promote to L1 after link stays in L0sTX and L0sRX states for a certain amount of time
  - ✓ Make the time period user programmable instead of hard-coding (suggested implementation)
- When only L1 is enabled
  - ✓ User similar algorithm used to enter L0sTX
  - ✓ Make the TX inactivity time period user programmable instead of hard-coding (suggested implementation)

L1 ASPM Continued…

- Always initiated by endpoint
- Upstream component must accept or reject the L1ASPM entry request
- In case of acceptance, both sides sends EIDLE sets and tri-state TX lines
- In case of rejection, endpoint must wait for 10 us before sending request again.
- Exit can be initiated by both endpoint and upstream component
  - ✓ When either side wants to send TLP
- Exits to L0 through Recovery state

Software-initiated L1

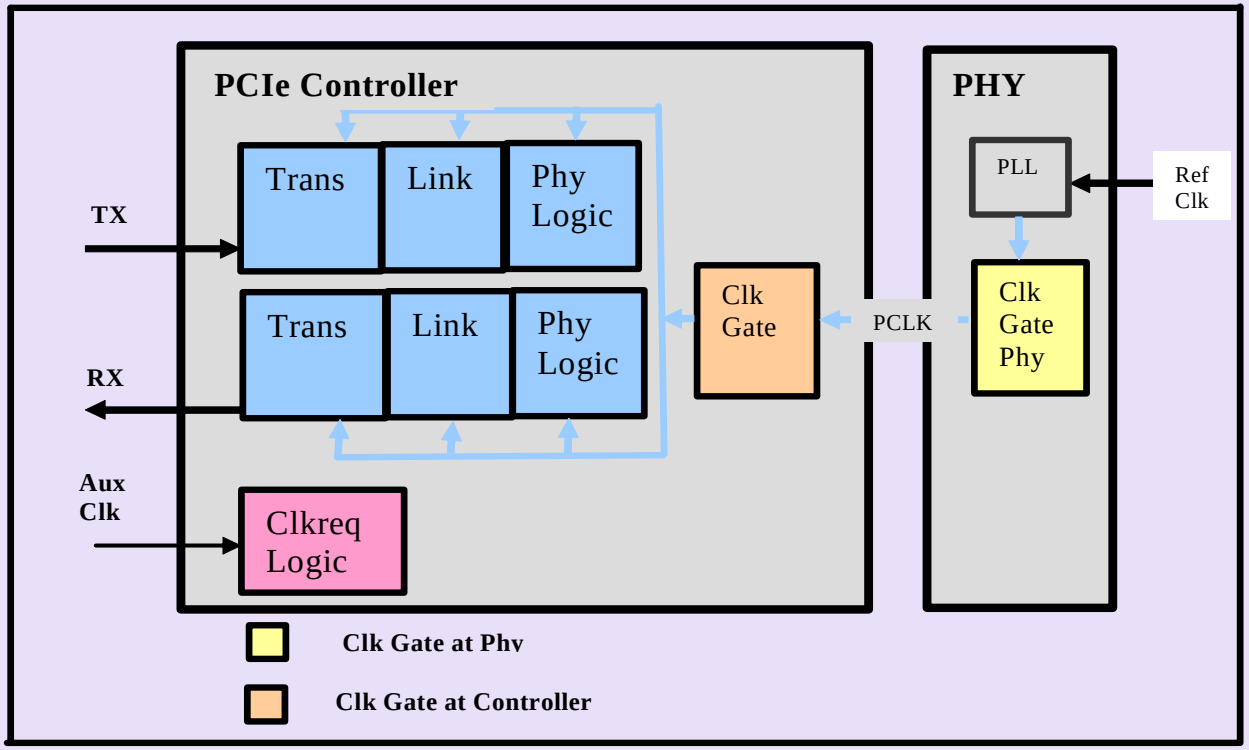
- Software programs the device into D3hot (can be D1 or D2 also)
- Endpoint initiates L1 entry
- Upstream component must accept L1 request
- The entry process from here is similar to L1 ASPM
- Exit is also similar to L1ASPM exit
  - ✓ When either side wants to send a TLP
- Exit to L0 through Recovery state
- Application can put its user logic to sleep in D3hot state

CLKREQ# Clock Control

- Supported in mobile platform
- 100 MHz reference clock may be turned off
  - ✓ This depends on platform-specific implementation
- PLL will be shut-off and PCLK gated from PHY
- Higher power savings due PLL off
- Higher exit latency due PLL re-locking

Clock Gating during CLKREQ#

- Clk gating done at Phy.
- Can be done at Controller also (suggested implementation)



L2 Entry

- RC sends PME_Turn_Off message
  - ✓ Can happen after software puts device in D3hot state
  - ✓ Also can happen during device in D0 state
- Endpoint sends back sends PME_TO_ACK message TLP
- Endpoint readies itself for eventual power and clock removal
- Endpoint continuously sends PM_ENTER_L2L3 DLLP
  - ✓ till it gets Acknowledgement or power is removed
- Core power is removed, clock is stopped
- Device maintains wake-up logic in separate power well

L2 Exit

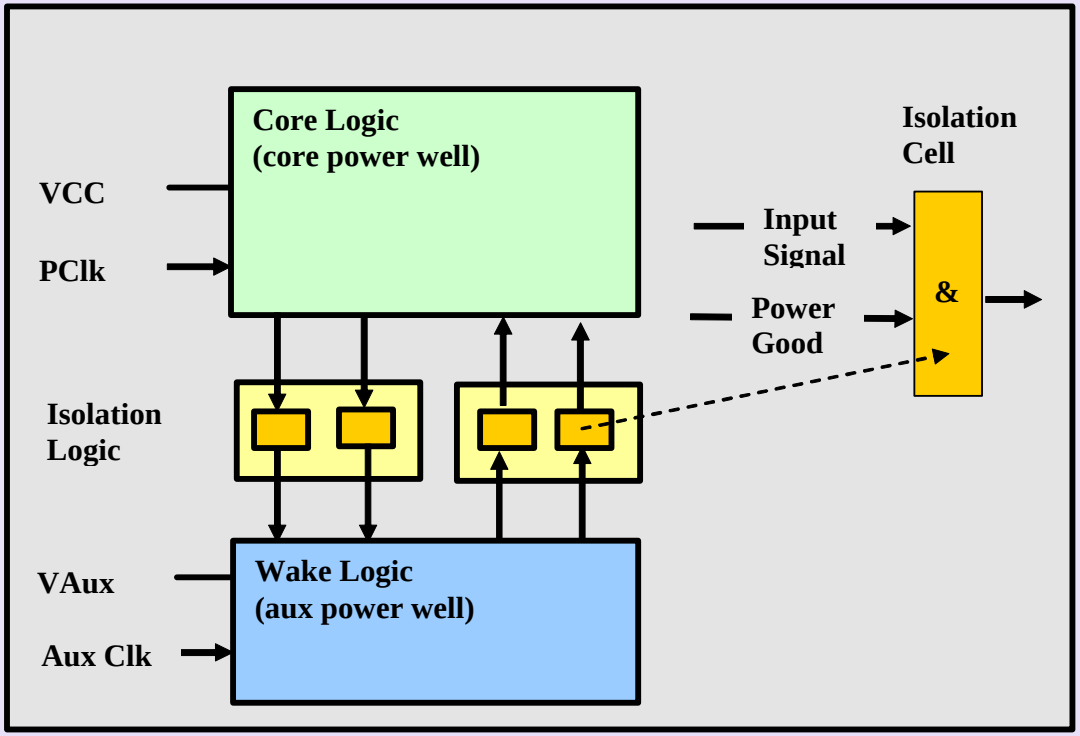
- Device wants to exit out of L2 and send TLP
- EP (device) asserts WAKE# signal OR
- EP Sends BEACON to upstream component
- System power manager restores power and clock
- EP sends PME virtual message
  - ✓ Similar in concept to the PCI PME# pin
  - ✓ Instead it goes out as a message TLP
- Software brings device to D0 state

How Other Methods Can be Applied for Further Power Saving

- Power Well Isolation
- Running PCLK at lower frequency

Suggested Power Well Isolation

- Power isolation required for signals crossing power wells to avoid floating inputs
  - ✓ Forced to zero through isolation cells – AND gates



Running PCLK at Lower Frequency

- Reduced clock frequency – Example: operate at
  - ✓ 125 MHz and 2xwide datapath versus –dual pipe mode
  - ✓ 250 MHz 1xwide datapath – single pipe mode
- Pros: Power savings due to lower operating frequency
- Cons: Higher gate-count and higher latency in datapath
- Mobile application might prefer lower frequency and interconnect/switch might prefer lower latency

Conclusion

- PCI Express provides many tools to maximize power savings
  - ✓ System states
  - ✓ Device states
  - ✓ Link/Phy states
- Aggressive clock gating Controller during L0sTX, L0sRX and L1 is possible
- Make the various inactive timers user programmable so that device driver can fine tune based on application
- Plan separate power wells with proper isolation techniques to save power during D3cold state
- Determine if 125 MHz PCLK frequency suits the latency requirements
- Have user-defined override bits for debuggability

About the Authors

- kishore@asic-architectinc.com
  - ✓ President and CEO, ASIC Architect, Inc.
- cc_hung@mentor.com
  - ✓ Product Line Manager , Mentor Graphics

Thank you for attending the PCI-SIG Developers Conference Europe 2007.

For more information please go to www.pcisig.com