Understanding PCIe® 2.0 Bandwidth Management

Betty Luk
Member of Technical Staff
Gord Caruk
Fellow
AMD

Agenda

- Does PCIe® 2.0 = 5GT/s?
- Link Speed Changes
- Link Width Changes
- Link Bandwidth Notification Capability
- Programming Model Considerations
- Summary

Does PCIe 2.0 = 5GT/s?

- 5GT/s support is OPTIONAL
- A PCIe 2.0 device may support only 2.5GT/s
- How do we know a device is PCIe 2.0?
  - ✓ PCI Express Capabilities Register
    - Capability version number = 02h
- How do we know a PCIe 2.0 device supports 5GT/s?
  - ✓ Link Capabilities Register
    - Maximum supported link speeds = 0010b (both 2.5GT/s and 5GT/s supported)

PCIe 2.0 devices may OPTIONALLY support 5GT/s!

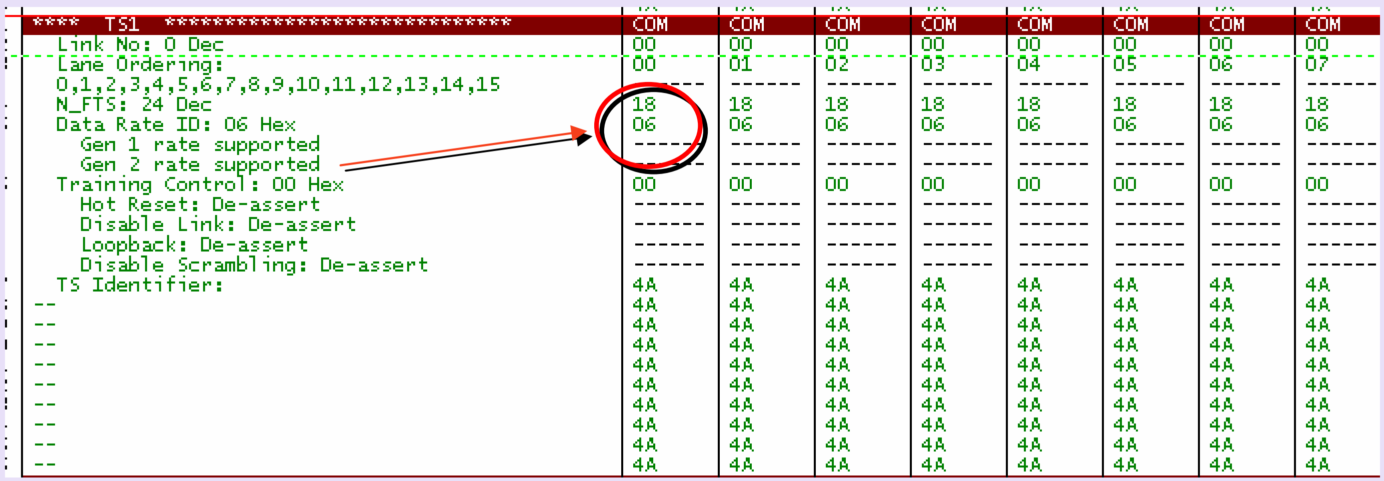
Does 5GT/s support = 5GT/s operation?

- From Section 6.11, Link Speed Management:
  - ✓ "… the Upstream component must attempt to maintain the Link at the Target Link Speed, or at the highest speed supported by both components on the Link…"
  - ✓ "During any given speed negotiation it is possible that one or both components will advertise a subset of all speeds supported…"
  - ✓ "If the Hardware Autonomous Speed Disable bit in the Link Control 2 register is clear, the component is permitted to autonomously adjust the Link speed using implementation specific criteria"

No requirement that link must (always) operate at 5GT/s!

When does the link operate at 5GT/s?

- When both sides advertise 5GT/s support in training sets



- To check the operating speed of the link:
  - ✓ Link Status register current link speed:
    - 0001b for 2.5GT/s
    - 0010b for 5GT/s

Agenda

- Does PCIe 2.0 = 5GT/s?
- Link Speed Changes
- Link Width Changes
- Link Bandwidth Notification Capability
- Programming Model Considerations
- Summary

Link Speed Changes

- Hardware initiated
  - ✓ Speed change due to unreliable link
    - When link cannot operate reliably at higher link speed
    - Eg. L0 at 5GT/s -> Recovery -> speed change to 2.5GT/s
  - ✓ Hardware autonomous speed changes
    - Speed changes not related to dropping link speed for reliability reasons
    - Catch-all for implementation specific speed changes
    - Disabled through hardware autonomous speed disable bit in link control 2 register

Spec method for speed change

- From the upstream component:
  - ✓ Use the target link speed in link control 2 register to set the maximum link speed
  - ✓ Write 1 to retrain link bit in link control register to initiate speed change
- No method is specified for downstream component
  - ✓ No specified way to change the speed advertised
  - ✓ No specified way to initiate speed change

Speed change initiated from downstream component is implementation specific.

Speed Change Bit

- New handshake in Recovery for speed change bit
  - ✓ Do not set the speed change bit if other side never advertised greater than 2.5GT/s
  - ✓ Other side must respond with speed change bit set (regardless of whether speed change actually occurs)
  - ✓ Otherwise 24ms timeout
    - – to Detect (causes link down)

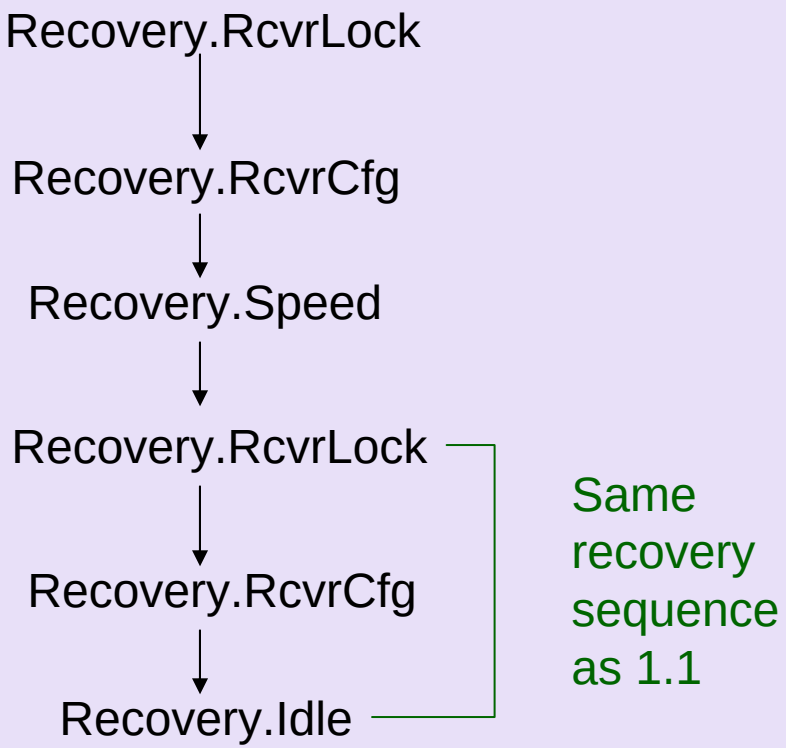
RC sets speed change bit

Packet				COM	Link	Lane	N_FTS	Training Control	Data Rate	TS1 Symbols	Time Delta	Time Stamp
1071879	R→	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s, Speed Change	D10.2 ...	32.000 ns	0090 . 553 264 720 s
1071880	R←	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s	D10.2 ...	32.000 ns	0090 . 553 264 752 s
1071881	R→	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s, Speed Change	D10.2 ...	32.000 ns	0090 . 553 264 784 s
1071882	R←	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s	D10.2 ...	32.000 ns	0090 . 553 264 816 s
1071883	R→	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s, Speed Change	D10.2 ...	32.000 ns	0090 . 553 264 848 s
1071884	R←	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s, Speed Change	D10.2 ...	32.000 ns	0090 . 553 264 880 s
1071885	R→	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s, Speed Change	D10.2 ...	32.000 ns	0090 . 553 264 912 s
1071886	R←	2.5 x16	TS1	K28.5	0	**	24	00000	2.5 GT/s, 5 GT/s, Speed Change	D10.2 ...	32.000 ns	0090 . 553 264 944 s

EP responds with speed change bit set

Speed Change Sequence

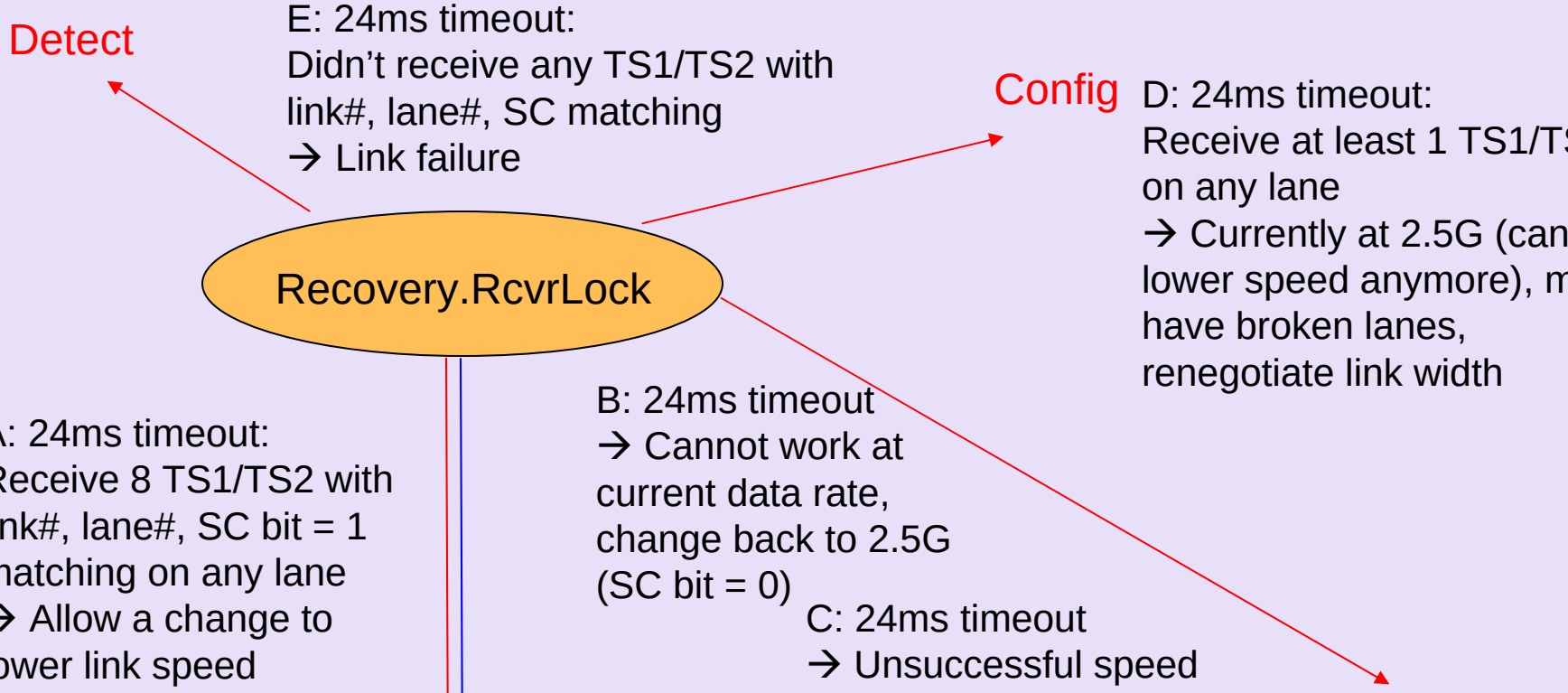
- Successful speed change sequence:



Unsuccessful Speed Change

- What happens if speed change is unsuccessful?
  1. LTSSM will try to lower the link speed first
  2. Then LTSSM will try to renegotiate to a lower link width
  3. Link down (undesirable!)

Recovery.RcvrLock



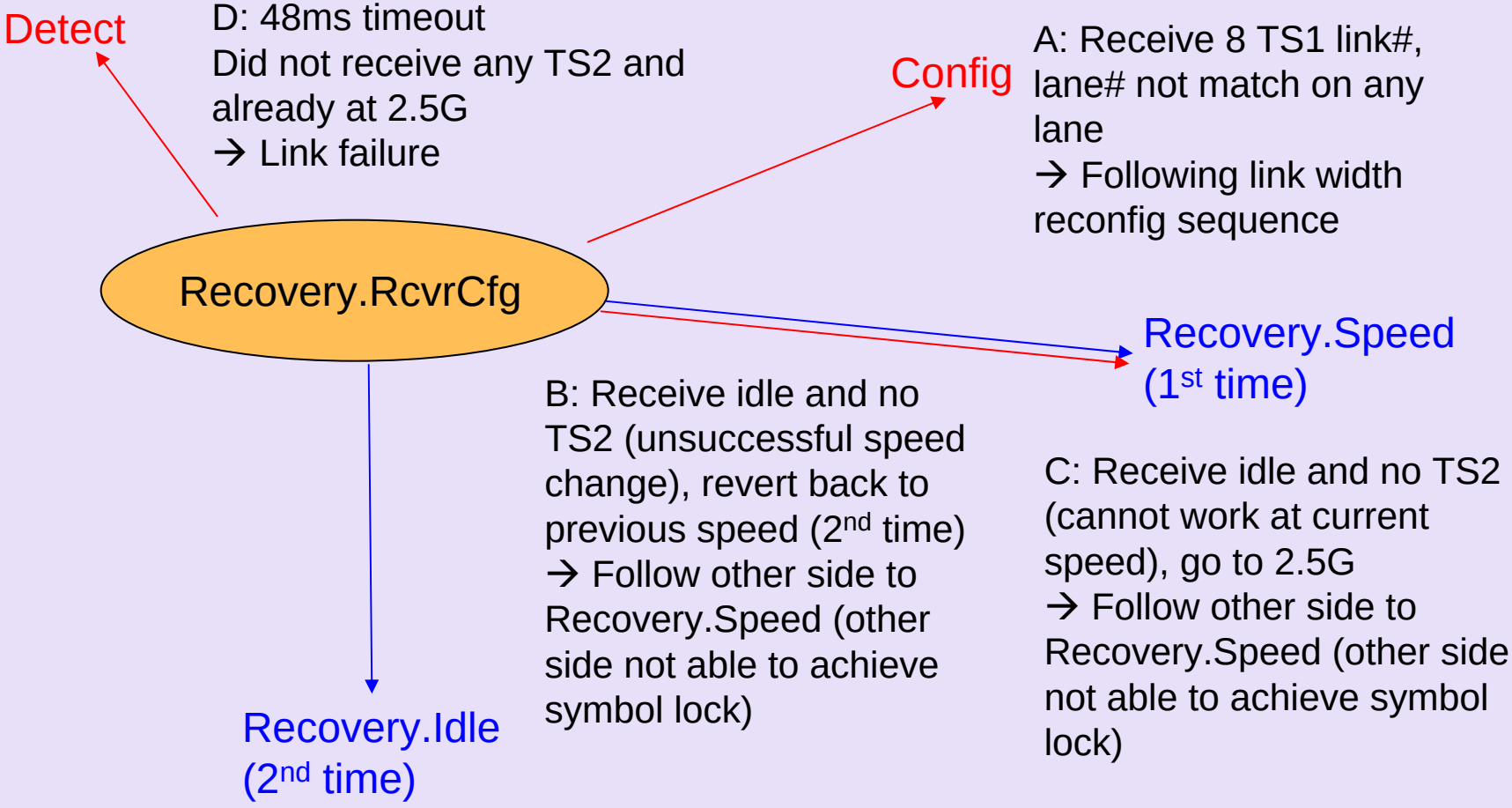
B: L0 -> **Lock -> Speed** -> Lock -> Cfg -> Idle -> L0

C: L0 -> Lock -> Cfg -> Speed -> **Lock -> Speed** -> Lock -> Cfg -> Idle -> L0

Blue arc – normal operation

Red arc – not intended for normal operation

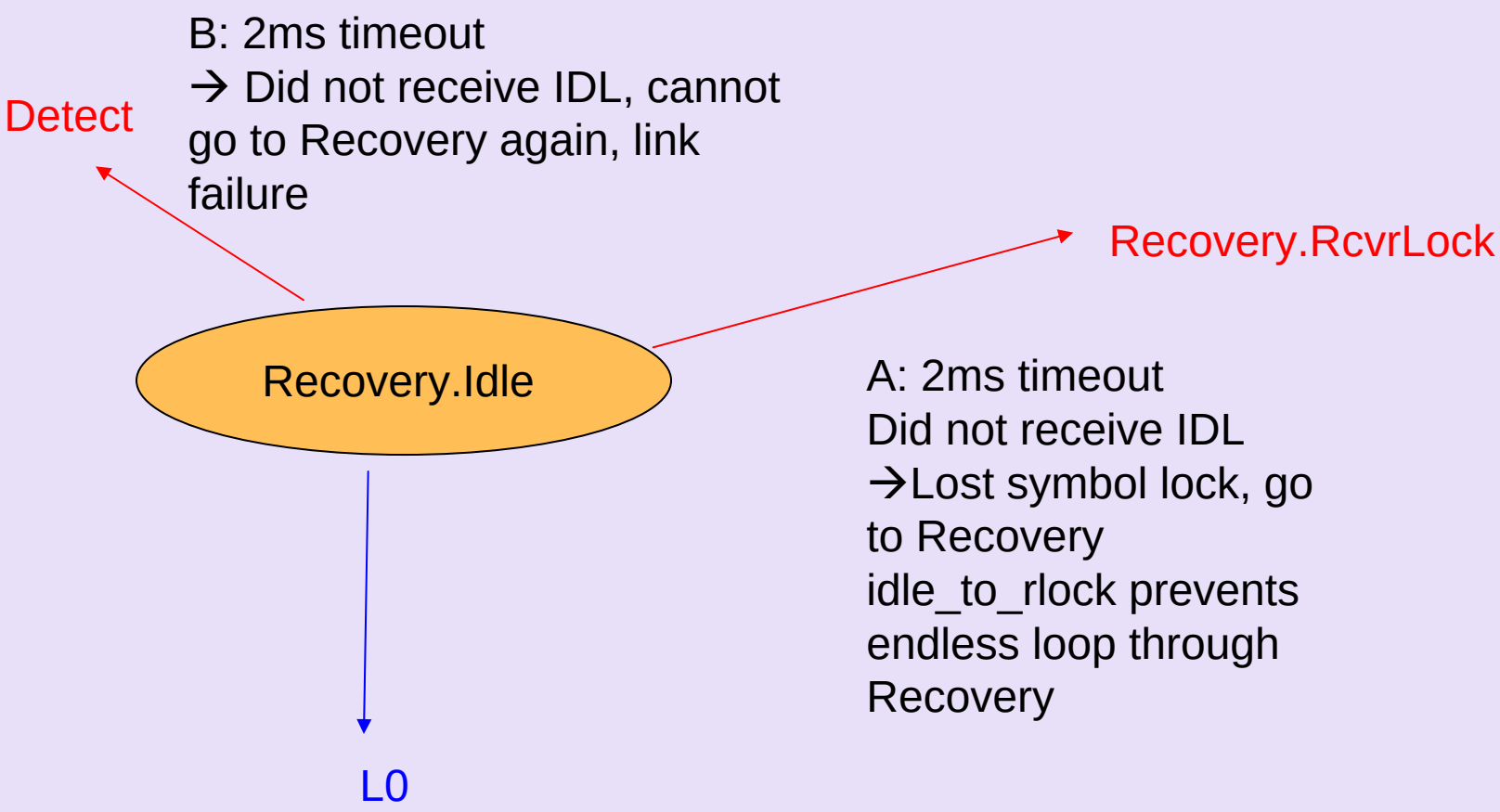
Recovery.RcvrCfg



B: L0 -> Lock -> Cfg -> Speed -> Lock -> **Cfg -> Speed** -> Lock -> Cfg -> Idle -> L0

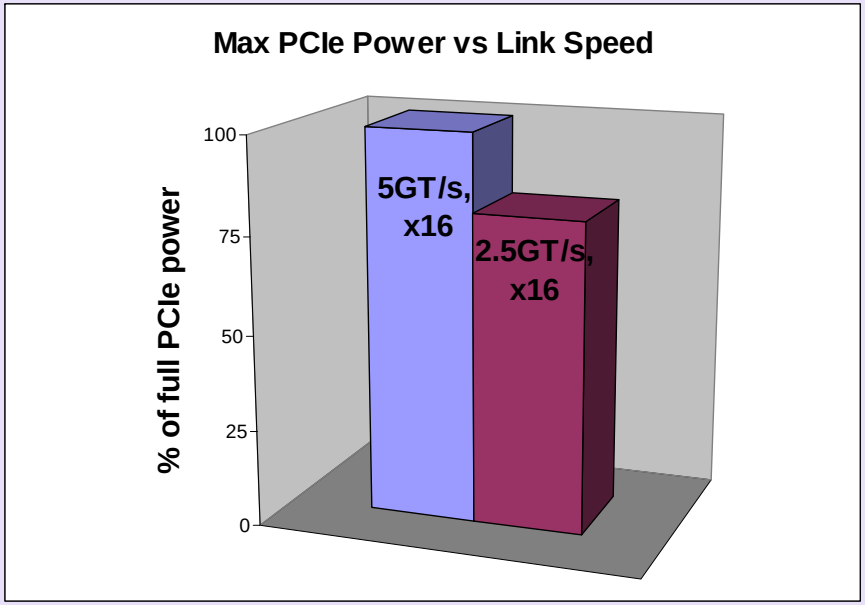
C: L0 -> Lock -> **Cfg -> Speed** -> Lock -> Cfg -> Idle -> L0

Recovery.Idle



Why change the link speed?

- To save power
  - ✓ No undesirable performance degradation
  - ✓ Performance state can be running at 5GT/s. Low power state can be running at 2.5GT/s.
  - ✓ Eg. Performance state = DVD playback
  - ✓ Eg. Low power state = Windows idle
- Reducing power consumption
  - ✓ Also reduces power needed for cooling solution



Why is power savings important?

- Battery life in mobile space
- Desktop and workstation spaces are becoming more power conscious
  - ✓ Going “green” is becoming a priority for the computer industry
  - ✓ Meeting Energy Star requirements

Image Source: www.time.com

Agenda

- Does PCIe 2.0 = 5GT/s?
- Link Speed Changes
- Link Width Changes
- Link Bandwidth Notification Capability
- Programming Model Considerations
- Summary

Link Width Changes

- LTSSM arcs always existed to reduce link width for reliability reasons
- PCIe 2.0 defines LTSSM state machine arcs to increase link width
  - ✓ Reliable and spec compliant way to dynamically change link width
  - ✓ No link down
  - ✓ Changes to Config states to specify when inactive lanes are turned on
- No programming model specified
  - ✓ Implementation specific
  - ✓ Disabled through hardware autonomous width disable bit

How to initiate link width change is implementation specific.

Upconfigure Capability

- Optional upconfiguration capability is advertised in training set
  - ✓ Down configure support is required
  - ✓ Can only initiate upconfigure if other side advertises support
  - ✓ Advertised during Config.Complete only bit 6, symbol 4 of TS2

**** TS2 ****	COM	COM	COM	COM	COM
Link No: 0 Dec	00	00	00	00	00
Lane Ordering:	00	01	02	03	04
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15	------	------	------	------	------
N_FTS: 24 Dec	18	18	18	18	18
Data Rate ID: 46 Hex	46	46	46	46	46
Gen 1 rate supported	------	------	------	------	------
Gen 2 rate supported	------	------	------	------	------
AutoChange / De-emphasis: Assert	------	------	------	------	------
Training Control: 00 Hex	00	00	00	00	00
Hot Reset: De-assert	------	------	------	------	------
Disable Link: De-assert	------	------	------	------	------
Loopback: De-assert	------	------	------	------	------
Disable Scrambling: De-assert	------	------	------	------	------
TS Identifier:	45	45	45	45	45
--	45	45	45	45	45
--	45	45	45	45	45
--	45	45	45	45	45
--	45	45	45	45	45
--	45	45	45	45	45
--	45	45	45	45	45
--	45	45	45	45	45

Bit 6 Symbol 4 of TS2

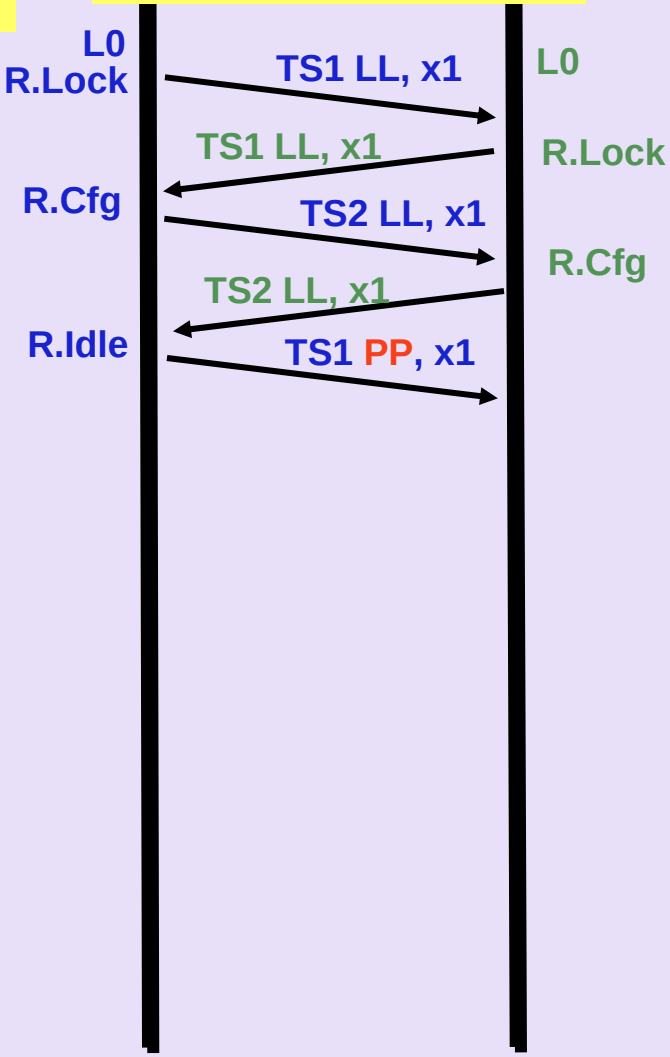
Sent by	LTSSM State	Meaning
Upstream / Downstream Component	Config.Complete	Upconfigure capability
Downstream Component	Recovery	HW autonomous BW change
Upstream Component	Recovery	De-emphasis preference
Upstream / Downstream Component	Polling.Config	De-emphasis used by other side if entering Loopback from Config

Upconfiguration sequence

EP initating upconfigure

Link is operating in x1

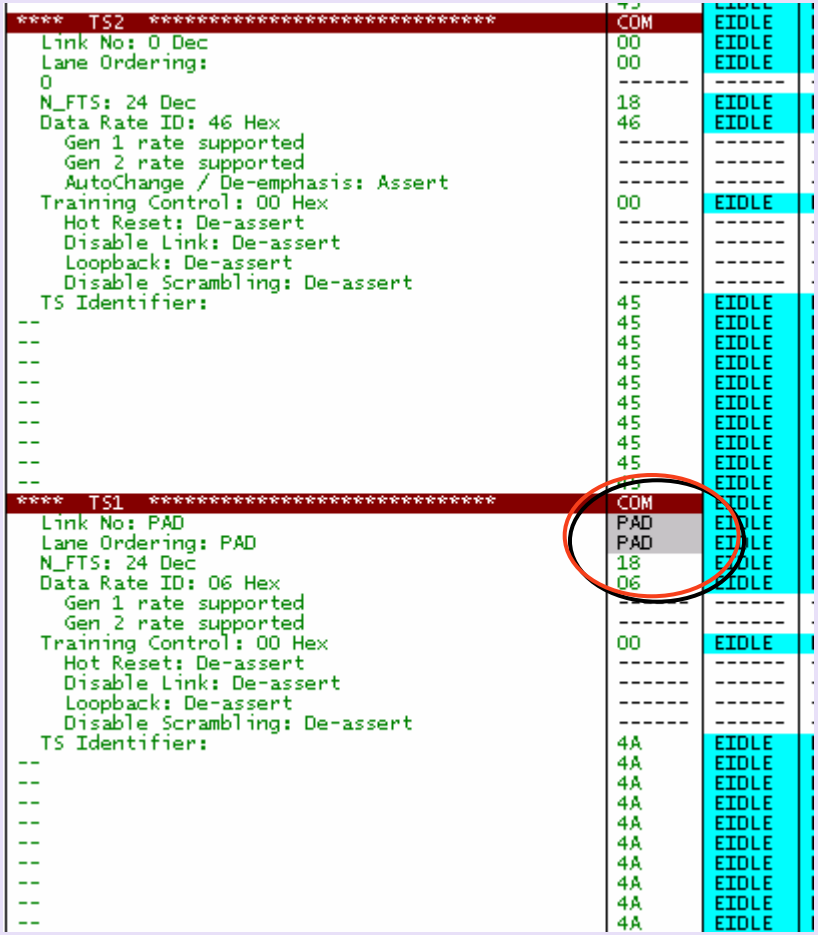
RC is the follower



Send PAD in R.Idle to initiate upconfigure

Initiating upconfig

- Sending TS1 PAD in Recovery.Idle to drive LTSSM to Config (to initiate link width upconfigure)

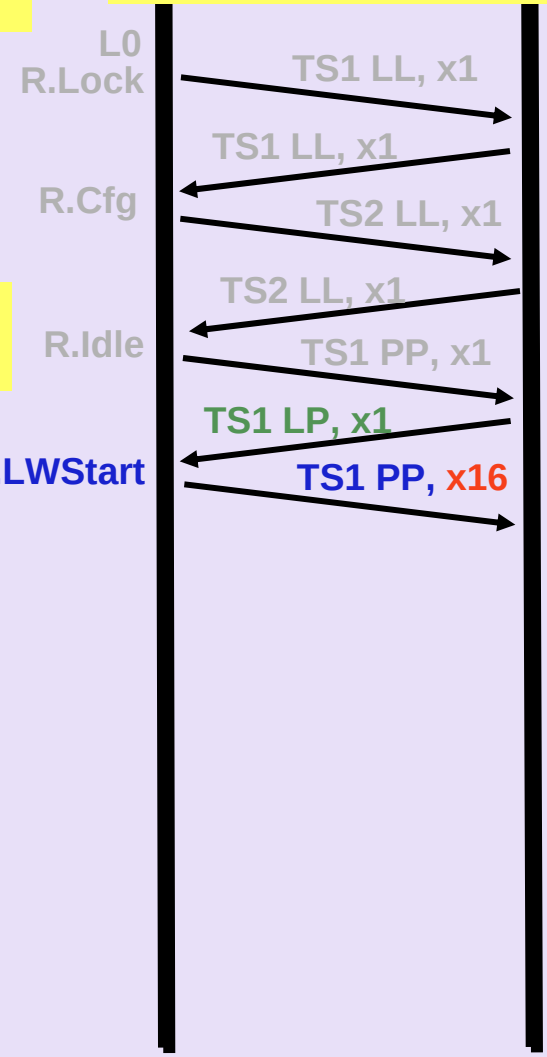


Upconfiguration sequence

EP initating upconfigure

Link is operating in x1

RC is the follower



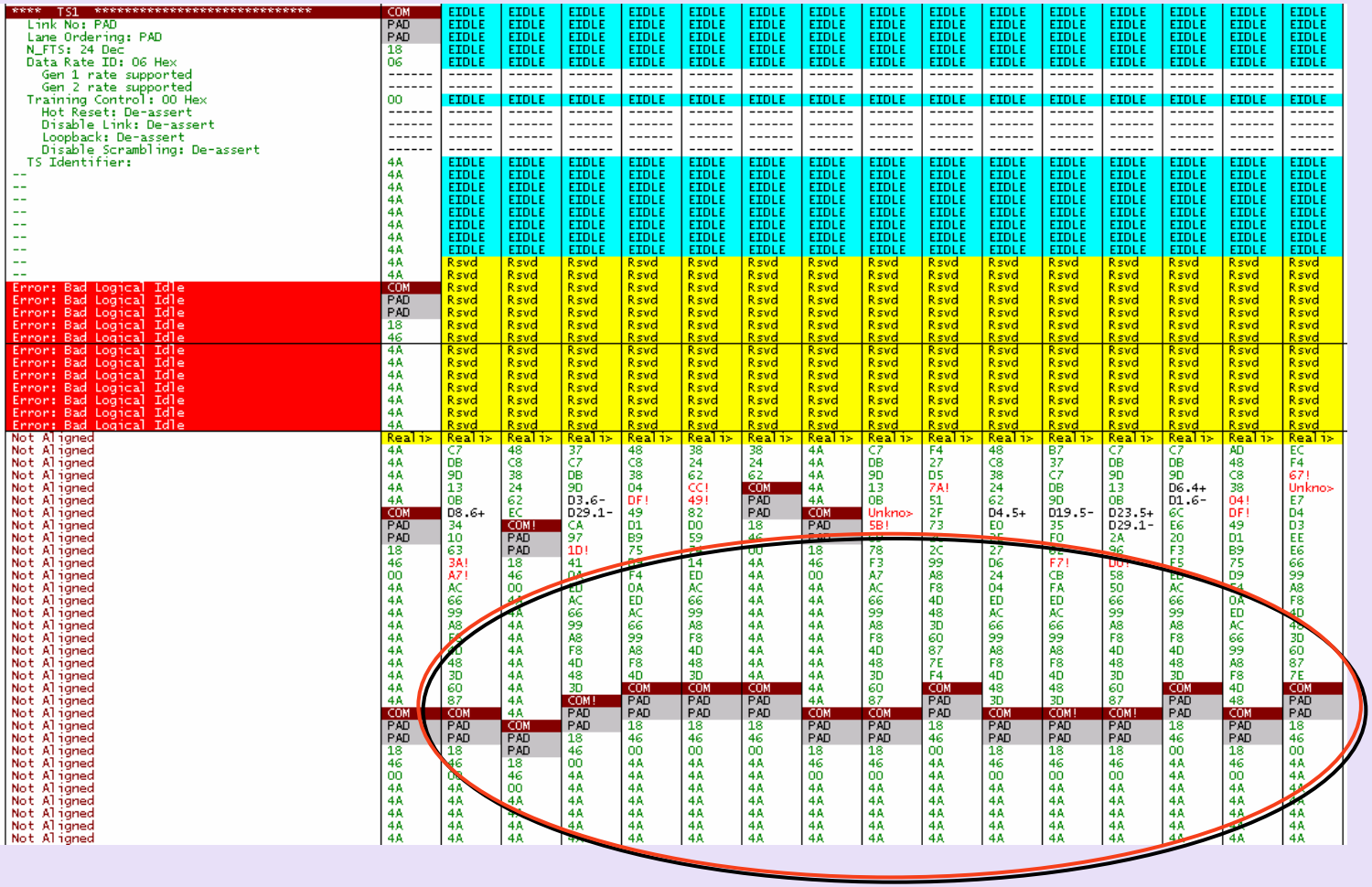
Send PAD in R.Idle to initiate upconfigure

Turn on inactive lanes

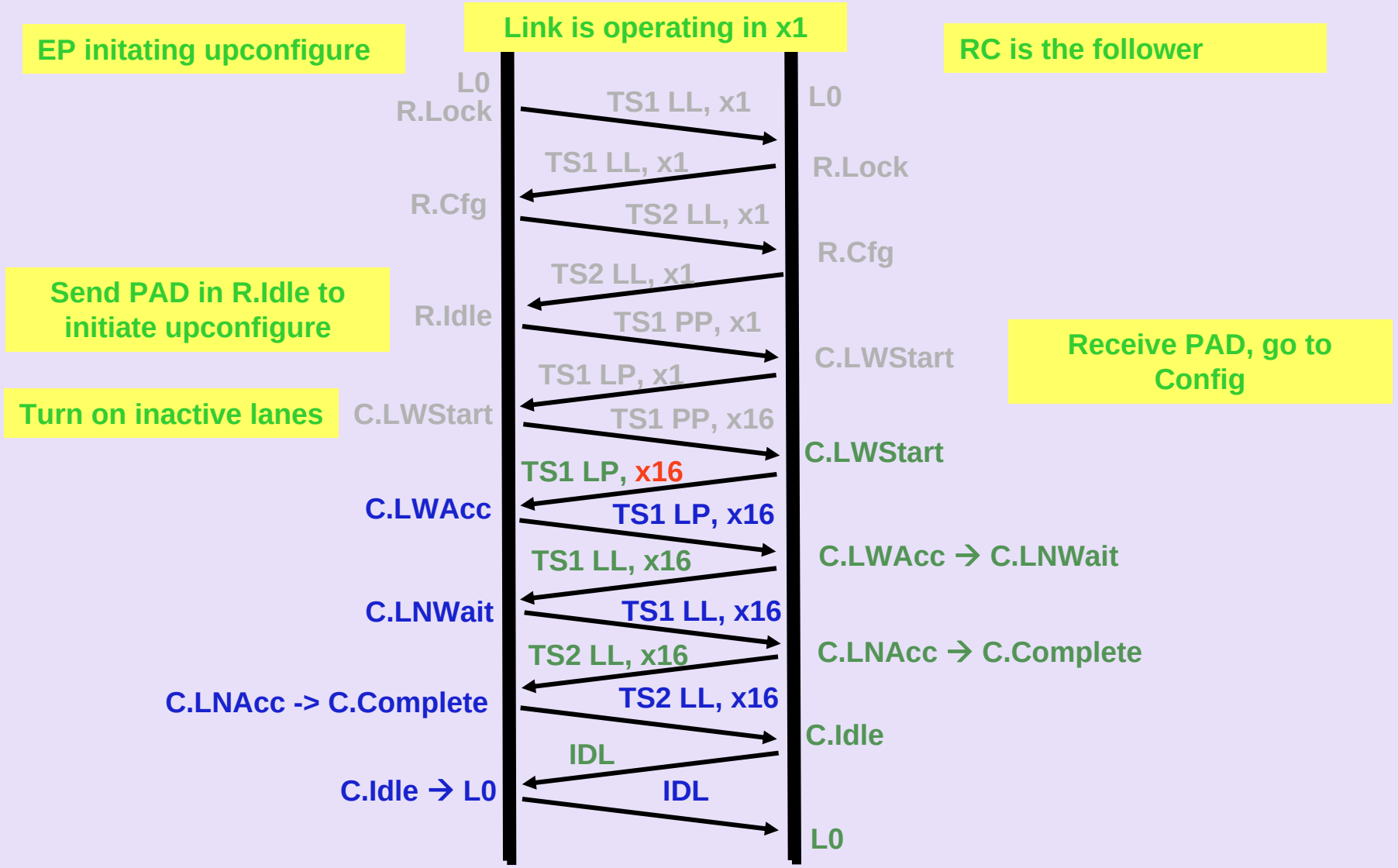
Receive PAD, go to Config

Turning on inactive lanes

- Turning on inactive lanes in Config.LWStart

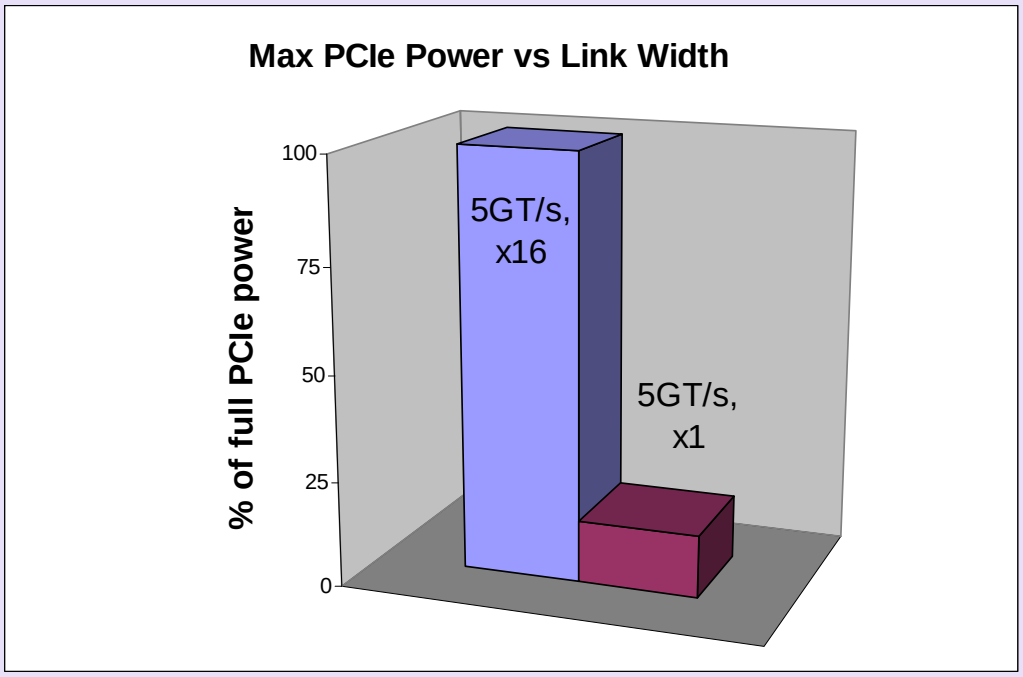


Upconfiguration sequence



Why change the link width?

- To save power
  - ✓ Battery life for mobile platforms
  - ✓ Reducing idle power in desktop and workstations
- Bandwidth requirements
  - ✓ Full bandwidth is not required at all times
  - ✓ Windows Vista Premium logo requires PCIe x4 bandwidth for Aero support



Considerations for link up/down configure

- What about the unused lanes?
  - ✓ Turn off unused transmitters and receivers for maximum power savings
  - ✓ From section 4.2.6.3.5 (Configuration.Complete)
    - It is recommended that the Receiver terminations of these Lanes be left on. If they are not left on, they must be turned on when the LTSSM enters the Recovery.RcvrCfg substate until it reaches the Config.Complete substate if upconfigure_capable is set to 1b to allow for potential Link width upconfiguration.
  - ✓ Recommendation:
    - Turn off unused lanes and turn on receiver terminations in Recovery
- "Limiting" the link width
  - ✓ Turn off receiver terminations
  - ✓ Negotiate lane out of configured link by sending PAD
  - ✓ To maintain a lower power state

Agenda

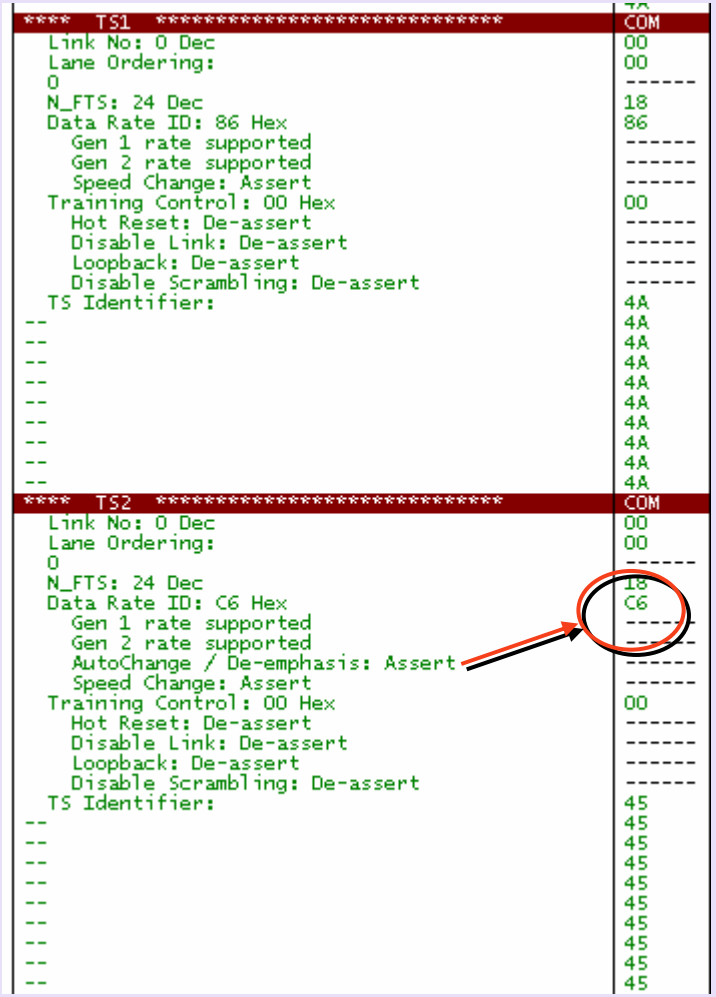
- Does PCIe 2.0 = 5GT/s?
- Link Speed Changes
- Link Width Changes
- Link Bandwidth Notification Capability
- Programming Model Considerations
- Summary

Link Bandwidth Notification Capability

- Required for root ports and switches that support:
  - ✓ Wider than x1 link
  - ✓ Multiple link speeds
- Link capabilities register
  - ✓ Link bandwidth notification capability = 1b if supported
- Link status register indicates if link speed or link width has changed
  - ✓ Link bandwidth management status
    - If retrain bit was set in root port
    - Link reliability reasons
  - ✓ Link autonomous bandwidth status
    - Non link reliability reasons
    - Autonomous change bit set by endpoint

Autonomous Change Bit

- Notifies upstream component that BW change is not caused by link reliability issue
- Downstream component sets bit 6, symbol 4
  - ✓ TS1 – in Configuration
  - ✓ TS2 – in Recovery



Link Bandwidth Notification Capability cont'd…

- Notify software that link bandwidth has changed
  - ✓ Push model – through interrupt
  - ✓ Pull model – through polling of status bits
- Option to generate interrupt when link speed or link width is changed
  - ✓ Link control register
    - – Link bandwidth management interrupt enable
    - – Link autonomous bandwidth interrupt enable
- When this is used
  - ✓ Monitoring and maintenance
    - – Signal when maintenance is required for a device that is no longer functioning at optimal bandwidth
  - ✓ Can also be used for test and debug
    - – During testing and validation to monitor if link speed or link width changes unexpectedly

Agenda

- Does PCIe 2.0 = 5GT/s?
- Link Speed Changes
- Link Width Changes
- Link Bandwidth Notification Capability
- Programming Model Considerations
- Summary

Programming Model Considerations

- Link speed and link width changes can be initiated by both sides of the link
  - ✓ Both sides may be asynchronously changing speed and width
  - ✓ Cannot cause a hang or (undesired) performance degradation
    - Example:
      - Link is operating at 5GT/s
      - EP lowers the link speed to 2.5GT/s
      - RC attempts to raise the link speed back up to 5GT/s continuously
      - Link is constantly going through recovery
    - Component is not allowed to attempt to change link speed within 200ms of a failed speed change
- Lower link speed by advertising only subset of link speeds supported
- Lower link width by turning off lanes that are no longer used

Recommended Usage Model

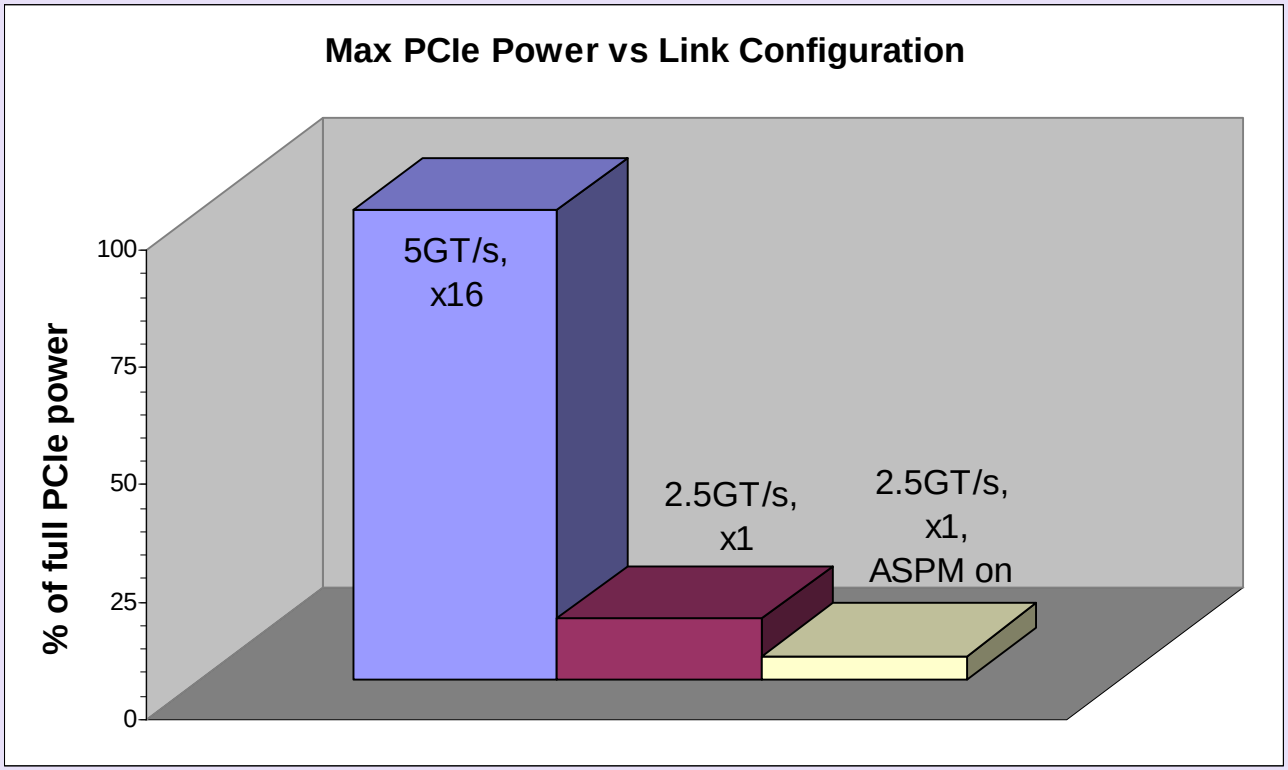
- Initial link-up at 2.5GT/s
  - ✓ Reduce power
  - ✓ Backwards compatible with some problematic PCIe 1.1 devices
- Who should be controlling the link speed and link width?
  - ✓ Recommend using endpoint control
    - When to do speed / width changes depends on access patterns
    - EP hardware + software driver
  - ✓ Need to intelligently manage bandwidth and power based on workload

Recommended Usage Model cont'd

- Endpoint device controls link speed and link width
  - ✓ Bring the link speed up to 5GT/s when the workload requires higher bandwidth
  - ✓ Hardware mechanism
    - Faster
    - Use if speed switching is expected to occur often
    - When no software is available
  - ✓ Software mechanism
    - Less cost in hardware
    - More flexible
  - ✓ Can use combination of hardware and software mechanisms

Putting it all together…

- Combine features together to reduce power:



- ✓ Lower link speed
- ✓ Lower link width
- ✓ Enable ASPM
- ✓ Lower operating voltage and clocks

Summary

- Reduce power to minimum
  - ✓ As long as bandwidth and performance isn't crippled
- Implement a mechanism for controlling link speed and link width from endpoint device
  - ✓ Make use of hardware and software mechanisms
- Provide (and advertise) support for link width upconfigure in multi-lane devices
  - ✓ Initial link up at maximum link width
- For interoperability, stress test link speed changes
  - ✓ With ASPM enabled

Thank you for attending the
PCI-SIG Developers Conference 2008

For more information please go to
www.pcisig.com