A Successful Approach to PCI Express® System Debug

Betty Luk

Senior Engineer

ATI Technologies

Agenda

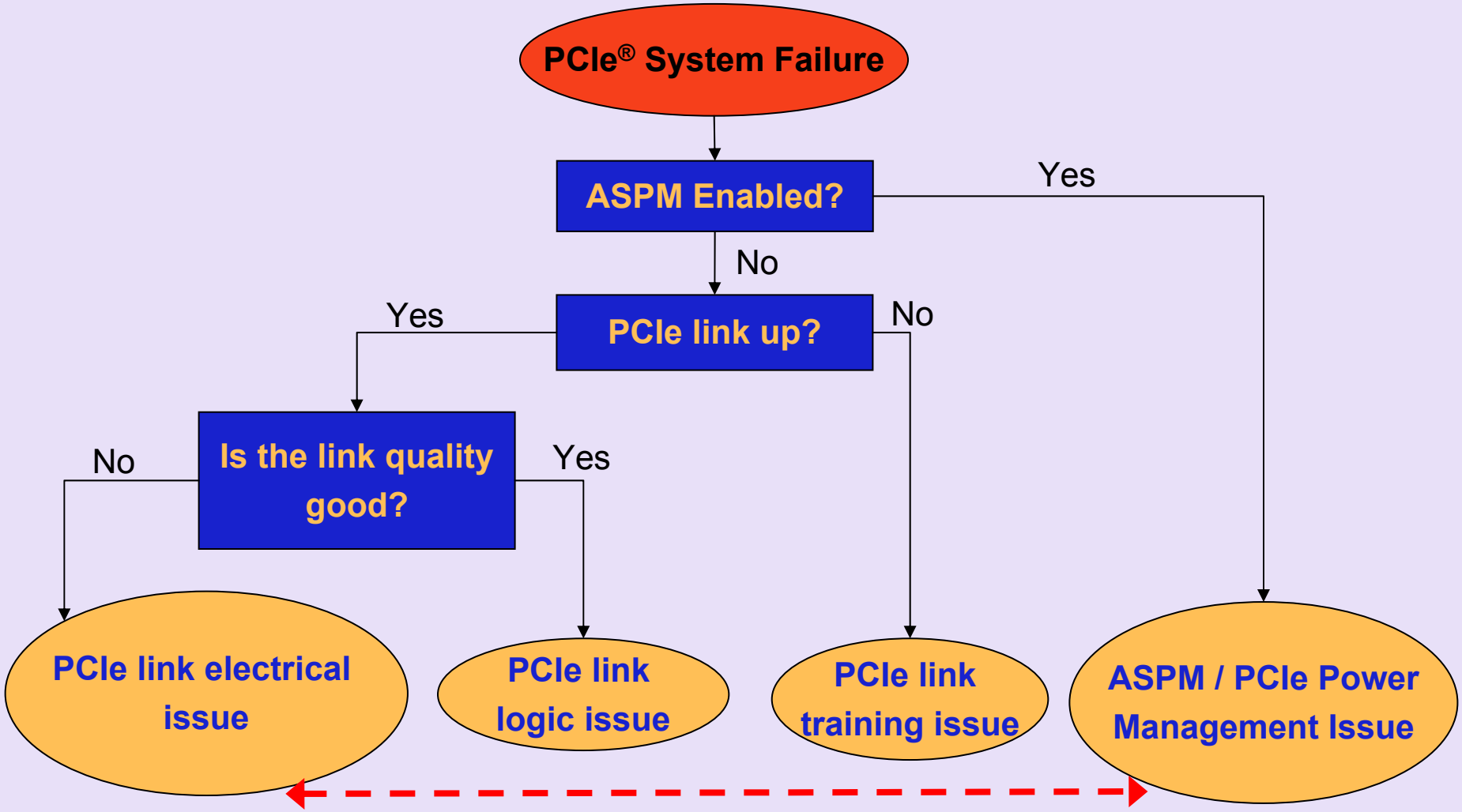
- Common PCI Express system issues
- Overview of PCI Express system debug flow
- Four debug examples
- Summary
- PCI Express debug toolkit

Common PCIe Issues

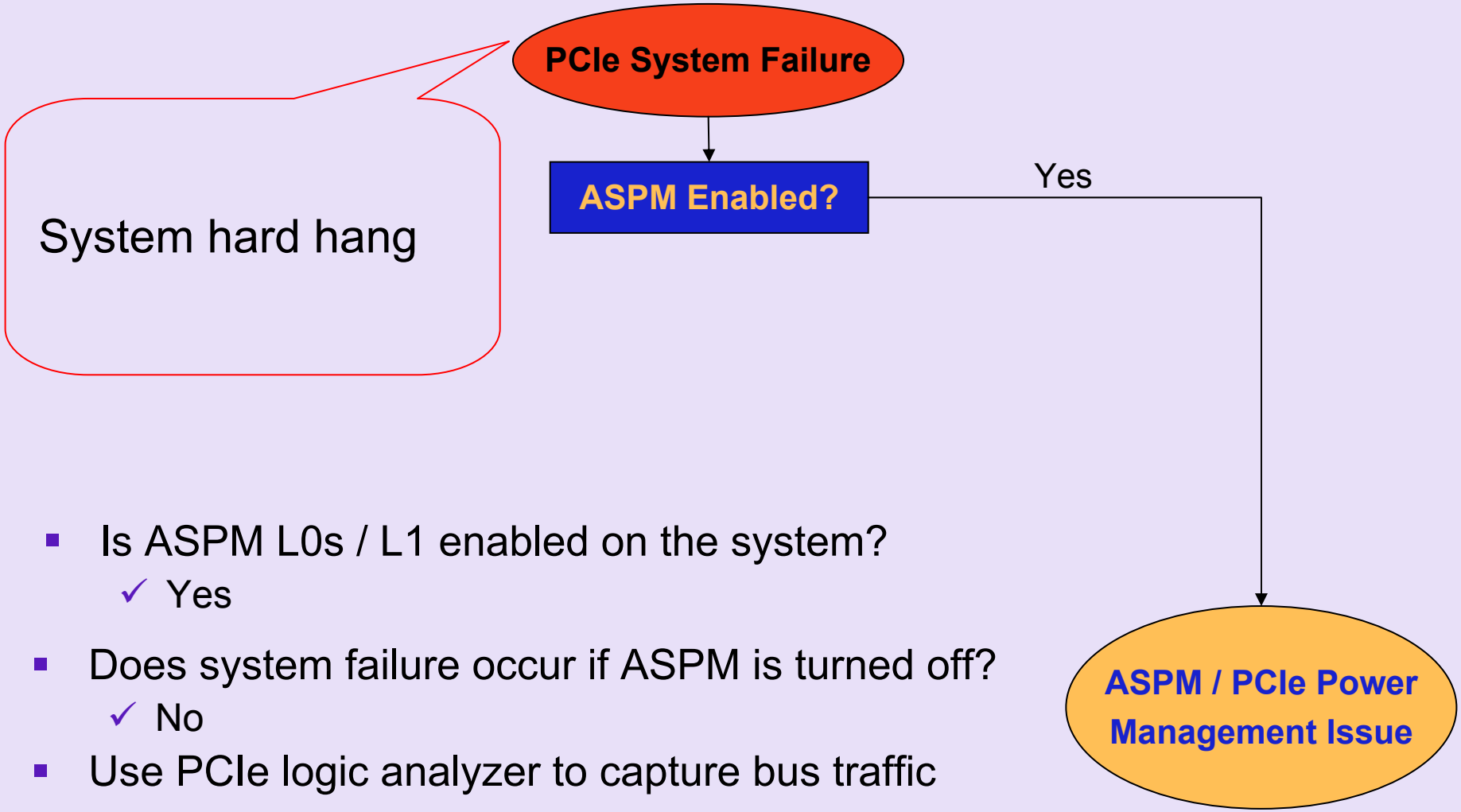
- Four categories of common PCI Express system issues:
  1. ASPM / PCIe Power Management
  2. Link Training
  3. PCIe Logic Issue
  4. Link Quality (electrical)

How do we distinguish between all these types of problems?

PCI Express Debug Flow

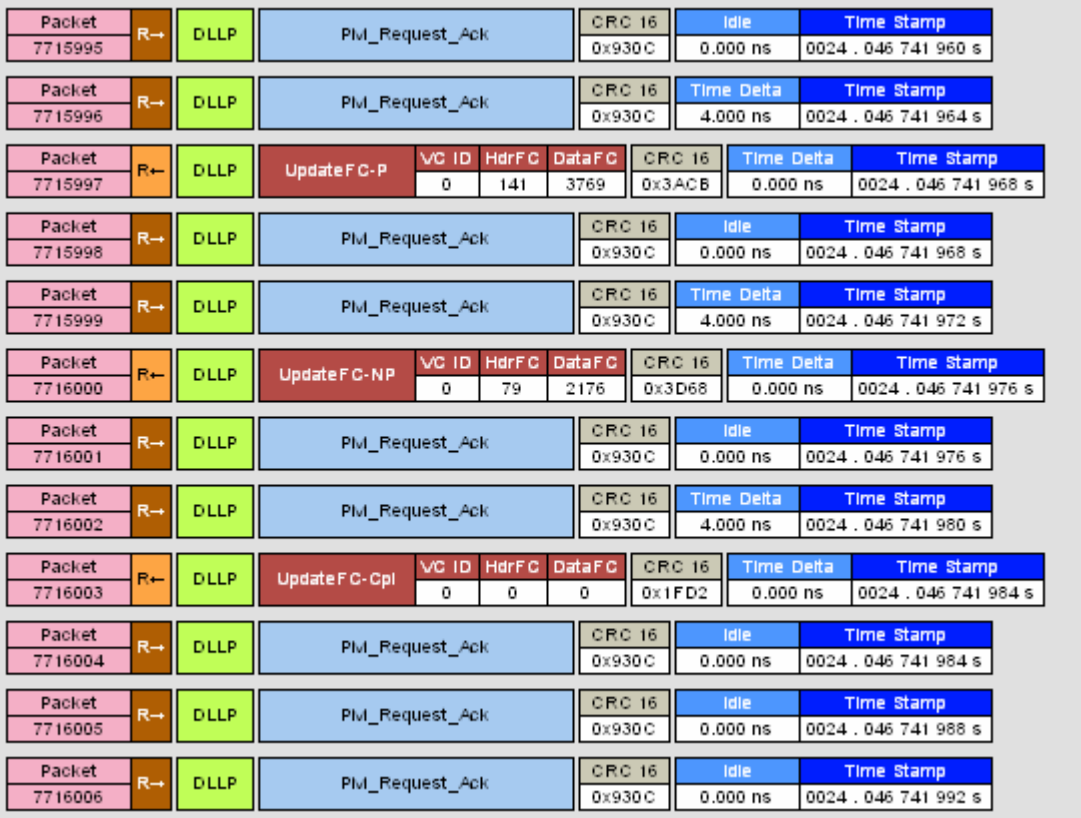


Let's Begin! Example 1



- Is ASPM L0s / L1 enabled on the system?
  - ✓ Yes
- Does system failure occur if ASPM is turned off?
  - ✓ No
- Use PCIe logic analyzer to capture bus traffic

Initial Analyzer Capture

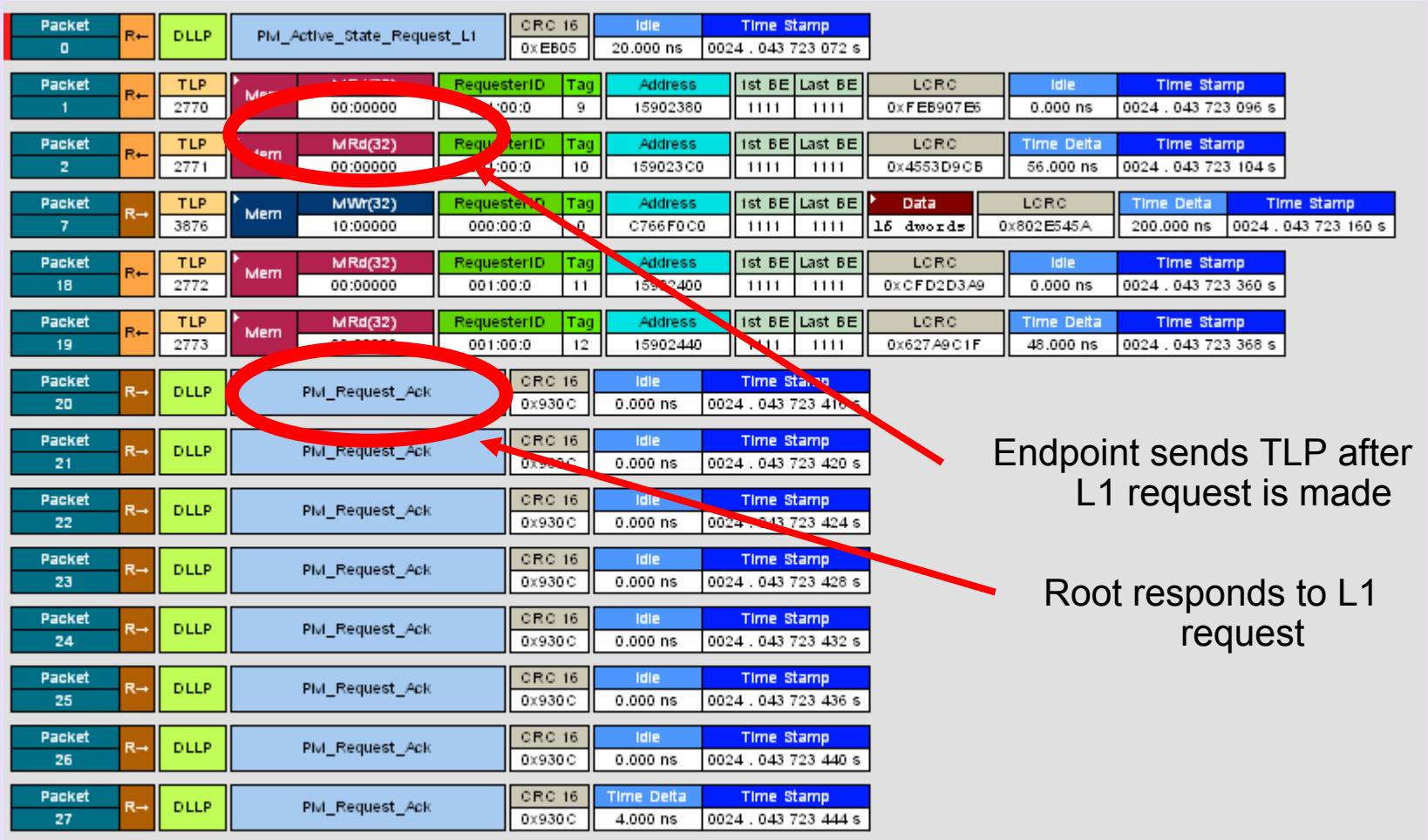


Packet												
7715995	R→	DLLP	PM_Request_Ack				CRC 16	Idle	Time Stamp			
							0x930C	0.000 ns	0024 . 046 741 960 s			
7715996	R→	DLLP	PM_Request_Ack				CRC 16	Time Delta	Time Stamp			
							0x930C	4.000 ns	0024 . 046 741 964 s			
7715997	R←	DLLP	UpdateFC-P	VC ID	HdrFC	DataFC	CRC 16	Time Delta	Time Stamp			
				0	141	3769	0x3ACB	0.000 ns	0024 . 046 741 968 s			
7715998	R→	DLLP	PM_Request_Ack				CRC 16	Idle	Time Stamp			
							0x930C	0.000 ns	0024 . 046 741 968 s			
7715999	R→	DLLP	PM_Request_Ack				CRC 16	Time Delta	Time Stamp			
							0x930C	4.000 ns	0024 . 046 741 972 s			
7716000	R←	DLLP	UpdateFC-NP	VC ID	HdrFC	DataFC	CRC 16	Time Delta	Time Stamp			
				0	79	2176	0x3D68	0.000 ns	0024 . 046 741 976 s			
7716001	R→	DLLP	PM_Request_Ack				CRC 16	Idle	Time Stamp			
							0x930C	0.000 ns	0024 . 046 741 976 s			
7716002	R→	DLLP	PM_Request_Ack				CRC 16	Time Delta	Time Stamp			
							0x930C	4.000 ns	0024 . 046 741 980 s			
7716003	R←	DLLP	UpdateFC-Cpl	VC ID	HdrFC	DataFC	CRC 16	Time Delta	Time Stamp			
				0	0	0	0x1FD2	0.000 ns	0024 . 046 741 984 s			
7716004	R→	DLLP	PM_Request_Ack				CRC 16	Idle	Time Stamp			
							0x930C	0.000 ns	0024 . 046 741 984 s			
7716005	R→	DLLP	PM_Request_Ack				CRC 16	Idle	Time Stamp			
							0x930C	0.000 ns	0024 . 046 741 988 s			
7716006	R→	DLLP	PM_Request_Ack				CRC 16	Idle	Time Stamp			
							0x930C	0.000 ns	0024 . 046 741 992 s			

- Steady state bus capture
- Link is up, but is “stuck”
- Root repeatedly sends PM_Request_Ack DLLPs
- Endpoint sending periodic UpdateFC DLLPs

- Need to get a bus capture at the time the bus hang occurs.
- Configure analyzer to trigger on Root repeatedly sending PM_Request_Ack DLLPs

Bus Capture at Time of System Hang

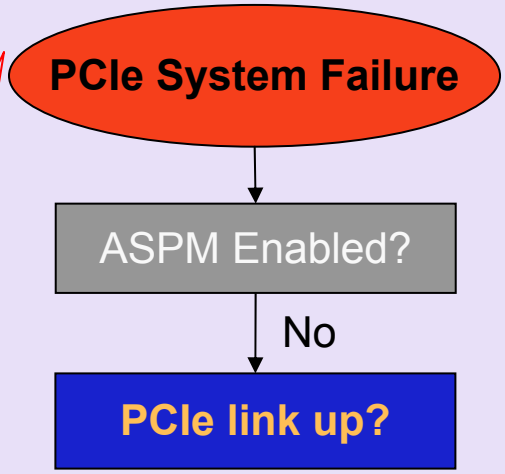


Root Cause

- At steady state, Root is continuously sending PM_Request_Ack DLLPs
- Endpoint is in L0 idle, sending UpdateFC DLLPs and SKP ordered sets
- L1 entry not complete because Endpoint does not go to electrical idle
- Endpoint should complete L1 protocol after sending PM_Active_State_Request_L1 before sending any more TLPs

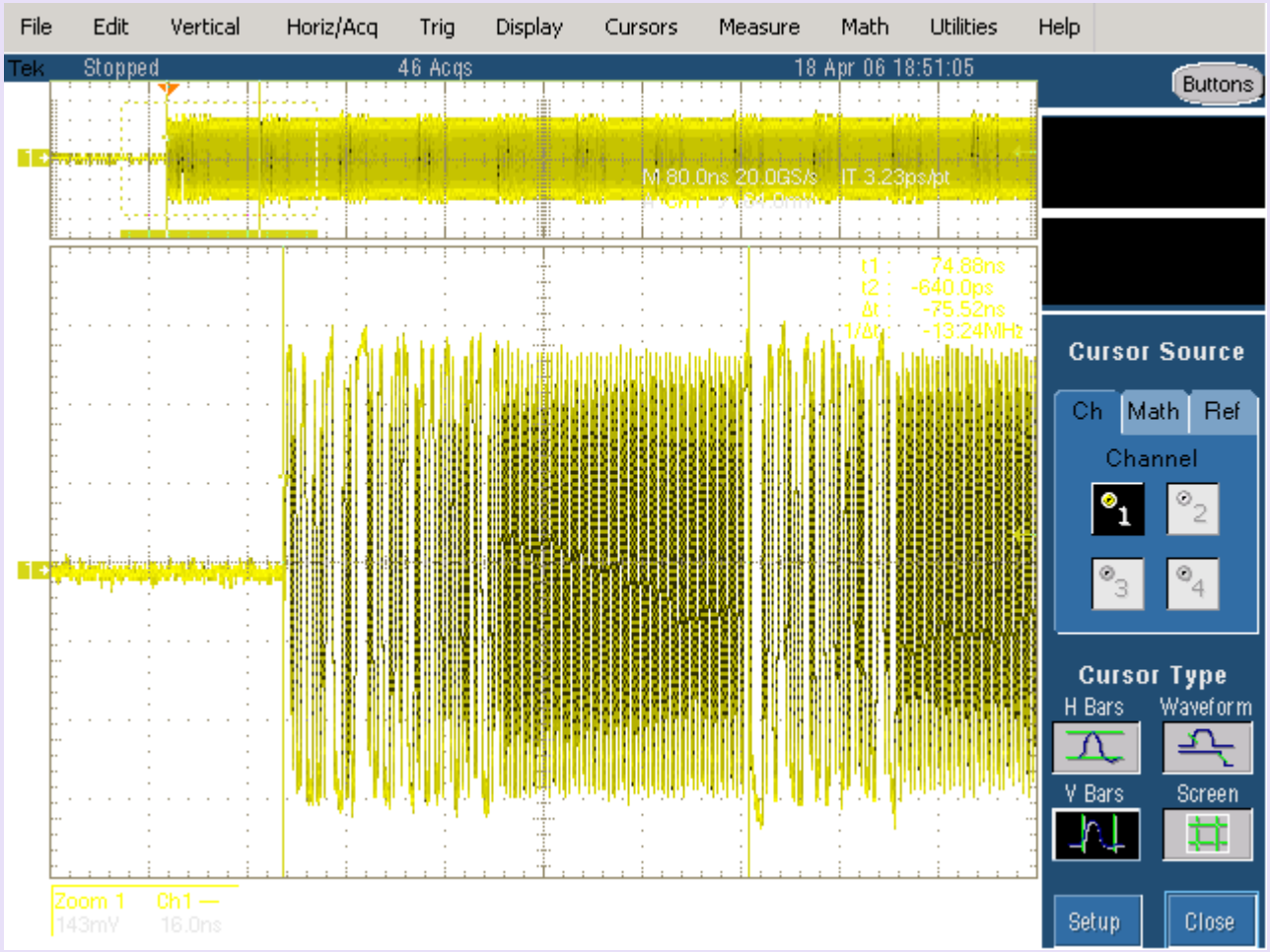
Example 2 – System Boot Failure

Intermittent system boot up failure (hang during initialization)



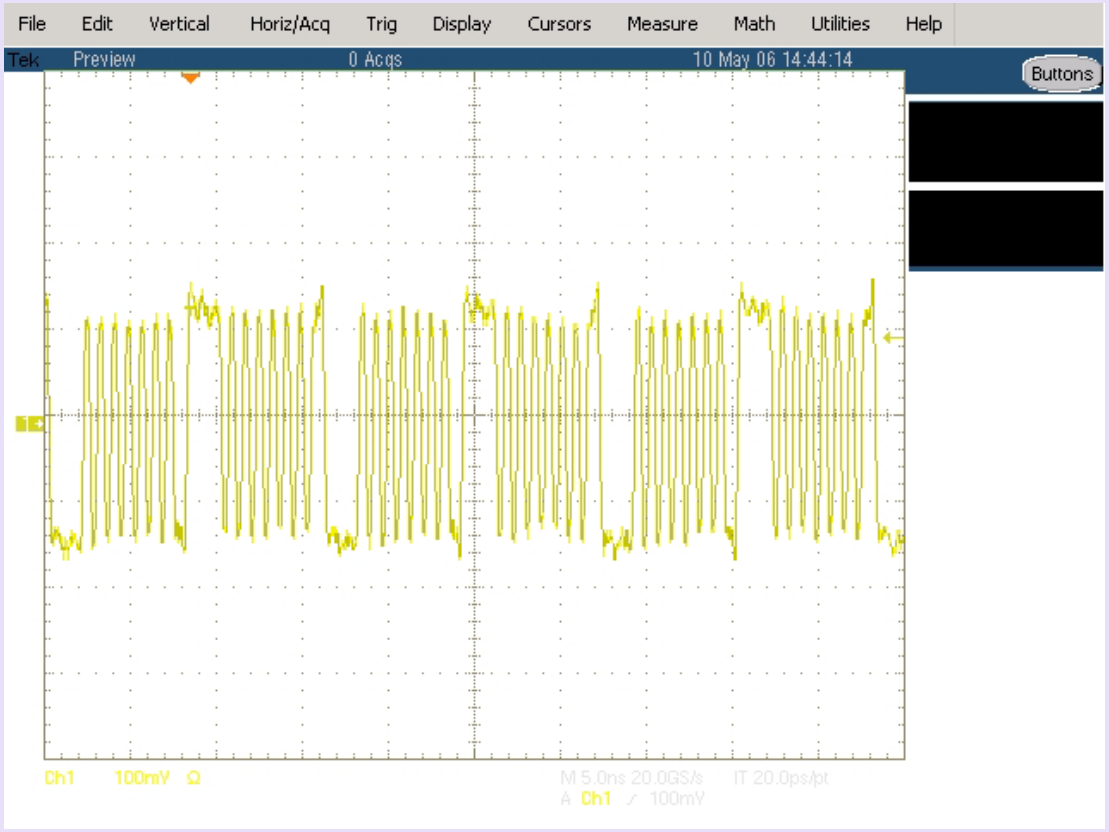
- What is the state of the PCIe link?
  - ✓ Check using a scope
  - ✓ Check using PCIe bus analyzer
    - Is LTSSM in L0? – trigger on training sets
    - Is the data link layer initialized? – any InitFC DLLPs, any UpdateFC DLLPs
  - ✓ Internal state vector

Using a Scope – Training Sets



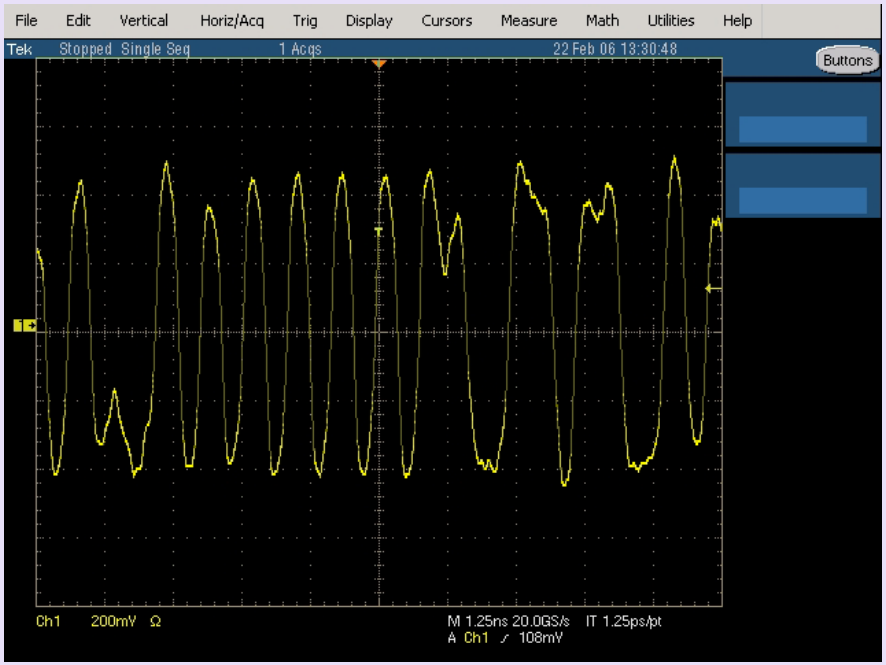
- Always send TS1 (or FTS) coming out of electrical idle
- TS1 identifier is D10.2 (toggle pattern in either disparity)
- K-Codes are not scrambled!

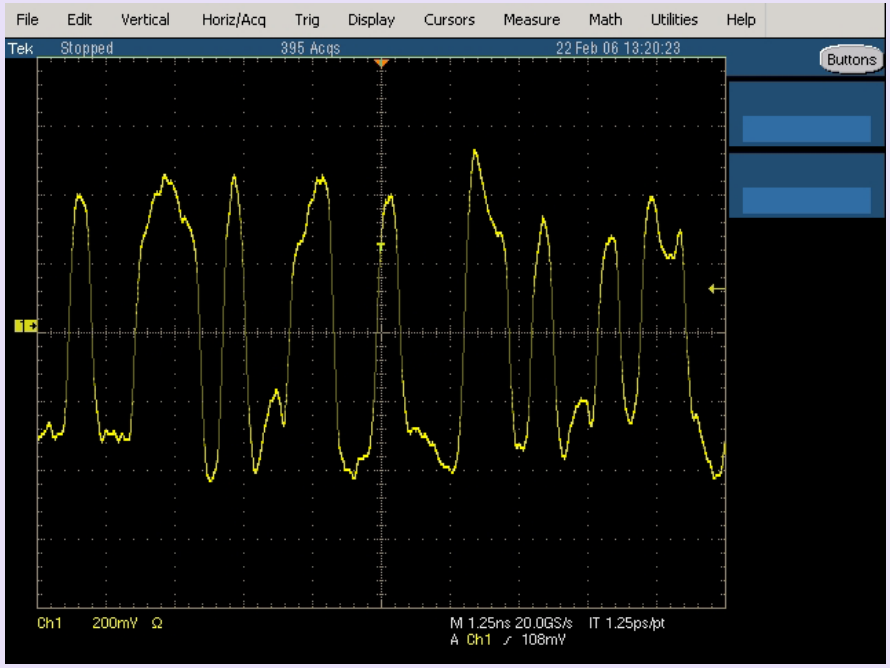
Compliance Pattern



- Compliance pattern: K28.5 D21.5 K28.5 D10.2
- Polling.Compliance – did not detect exit from electrical idle

Active Link





- PCIe link is active (but cannot distinguish what is being sent)

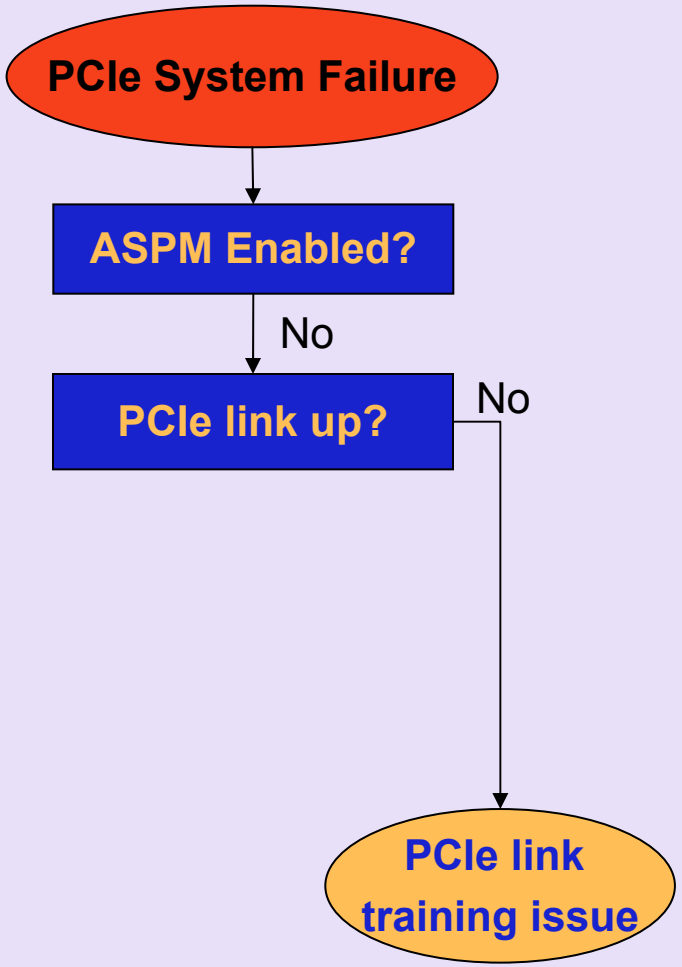
Link Training Failure Example

Packet			COM	Link	Lane	N_FTS	Training Control	TS1	Time Delta	Time Stamp
710767	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 136 s
710768	R←	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	0.000 ns	0012 . 661 143 200 s
710769	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 200 s
710770	R←	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	0.000 ns	0012 . 661 143 264 s
710771	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 264 s
710772	R←	TS2	K28.5	PAD	PAD	12	0 0 0 0	D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2	0.000 ns	0012 . 661 143 328 s
710773	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 328 s
710774	R←	TS2	K28.5	PAD	PAD	12	0 0 0 0	D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2	0.000 ns	0012 . 661 143 392 s
710775	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 392 s
710776	R←	TS2	K28.5	PAD	PAD	12	0 0 0 0	D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2	0.000 ns	0012 . 661 143 456 s
710777	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 456 s
710778	R←	TS2	K28.5	PAD	PAD	12	0 0 0 0	D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2 D05.2	0.000 ns	0012 . 661 143 520 s
710779	R→	TS1	K28.5	PAD	PAD	12	0 0 0 0	D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2 D10.2	64.000 ns	0012 . 661 143 520 s

Root

PCIe Link Not Up

- Root sees the TS1 sent by Endpoint, responds with TS2
  - ✓ Root is in Polling.Configuration
- Endpoint is only sending TS1
  - ✓ Endpoint is in Polling.Active
- Link training failure due to Endpoint not receiving TS1



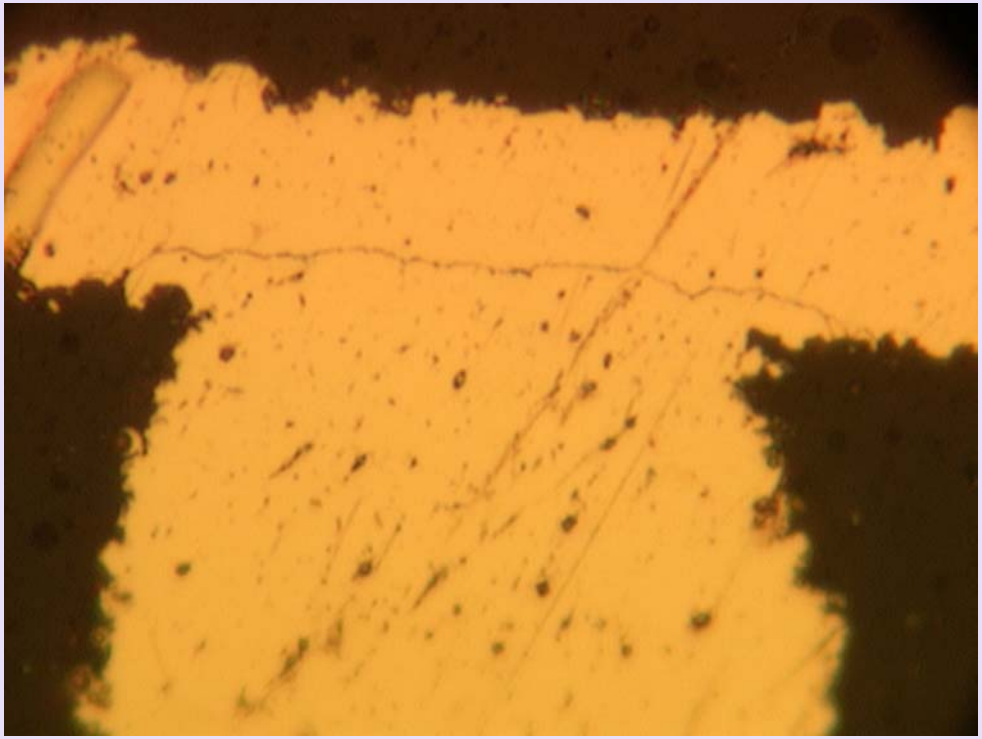
Isolate Failing Lane

- Try x1, x2, x4, x8 link training
- Try lane reversal mode
- Results:

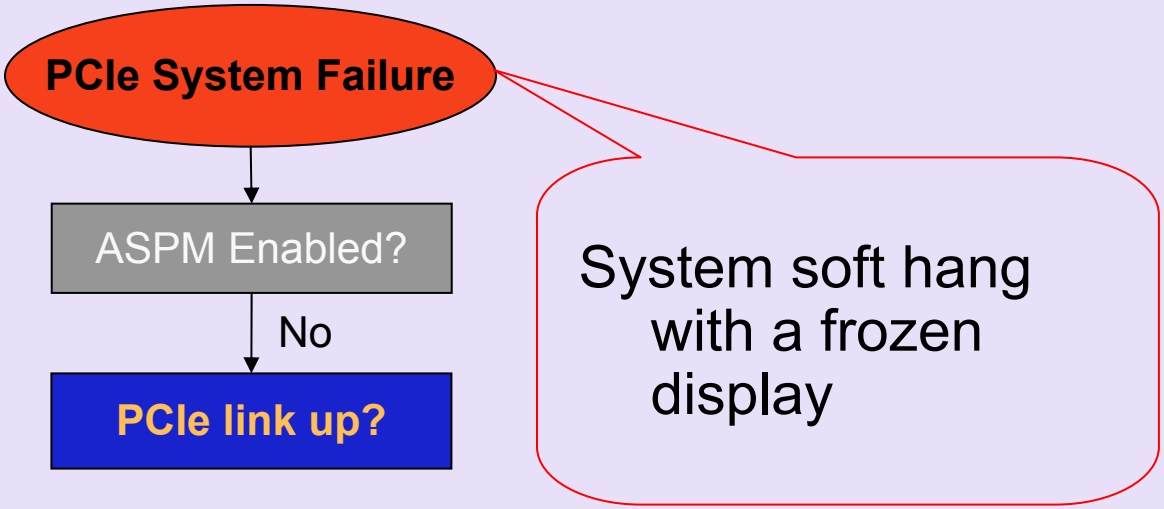
Lane Width	Forward	Reverse
x1	Pass	Pass
X2	Pass	Pass
X4	Pass	Pass
X8	Pass	Fail
X16	Fail	Fail

Root Cause

- Internal debug signals show lane 9 did not receive any data
- TDR measurement shows discontinuity on lane 9+ receive pin
- Further package analysis revealed package defect

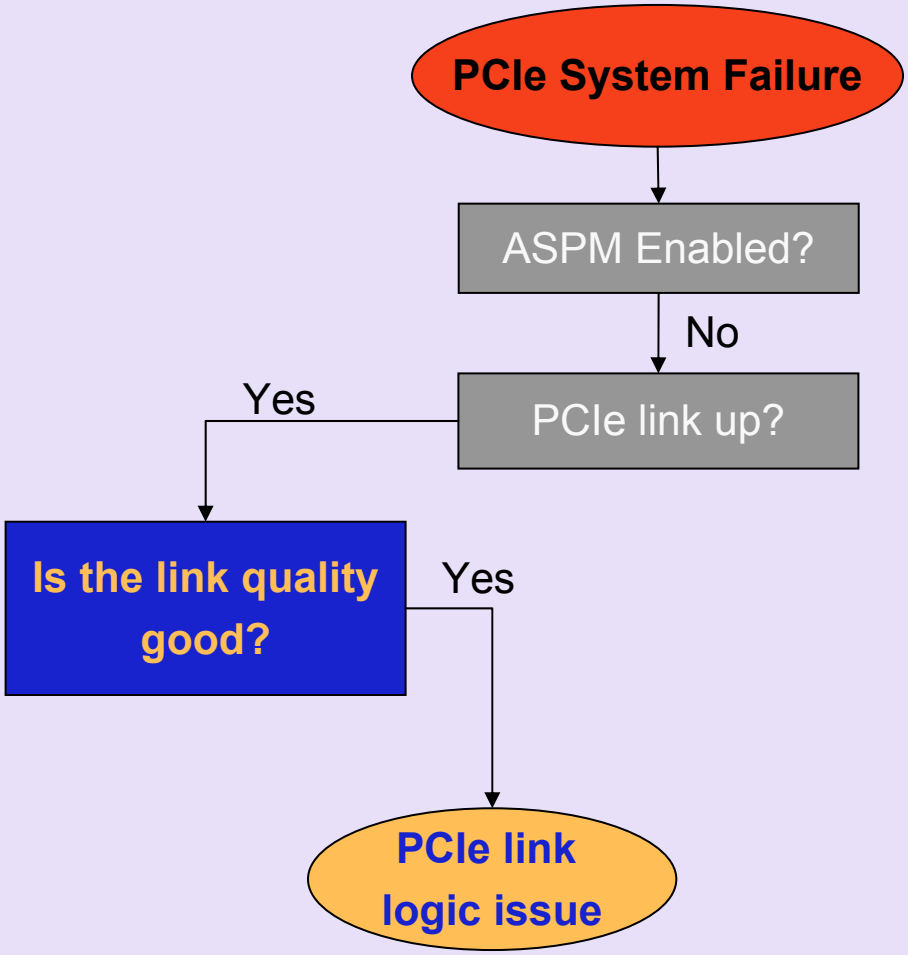


Example 3



- Software debugger can break into system
  - ✓ Issue cycles across the bus
  - ✓ PCIe link is up and operational

Check Link Quality

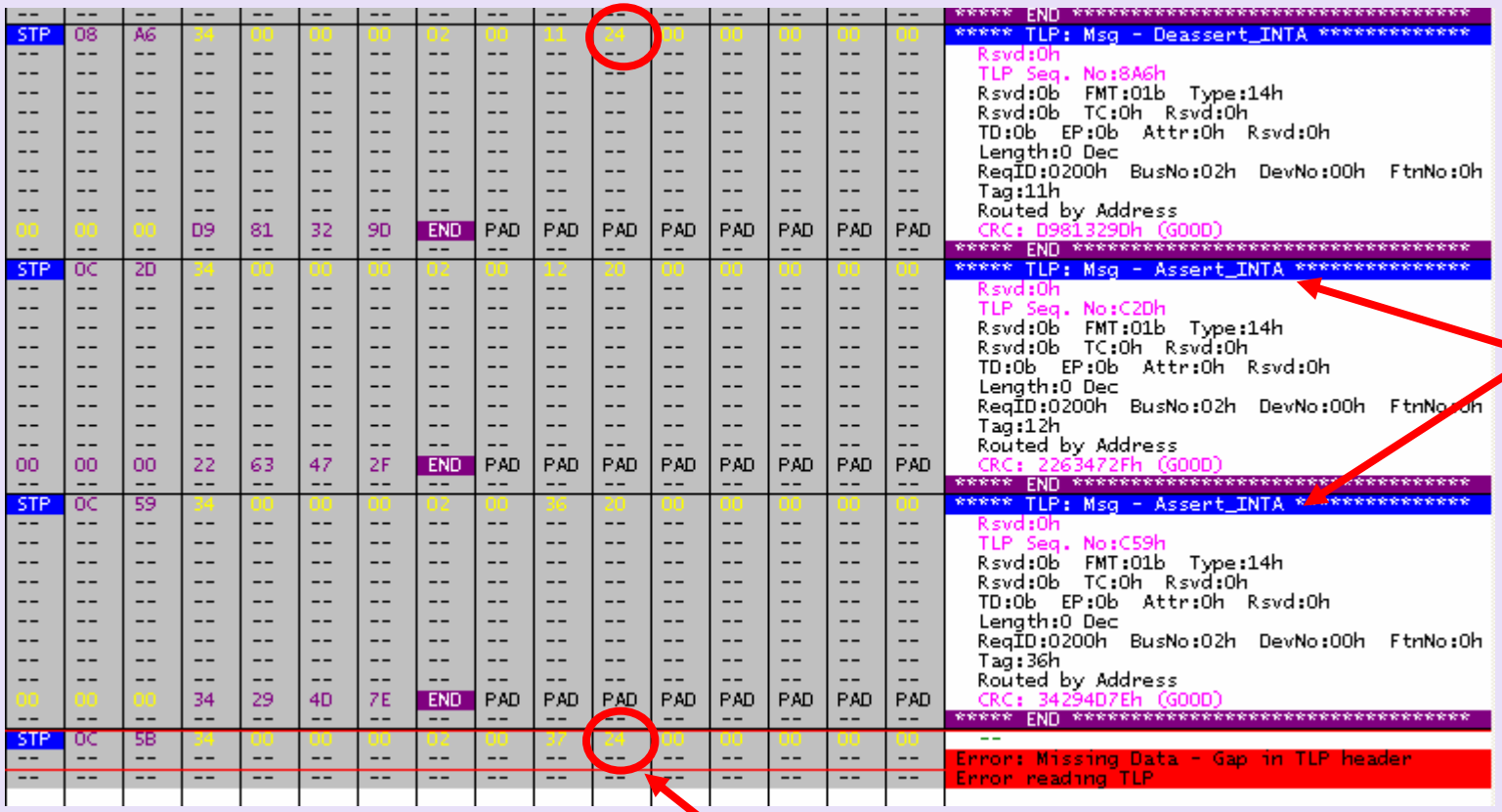


- No indication of a link quality problem (details discussed later)
- Need further logic analysis

PCIe Logic issue

- Software debugging shows that there is a pending interrupt for the device, but no interrupt is pending at the system
  - ✓ Look at interrupt messages
- Analyzer setup:
  - ✓ Conditional storage (only store Assert_INTA and Deassert_INTA messages) and stop capture when hang occurs

Bus Capture

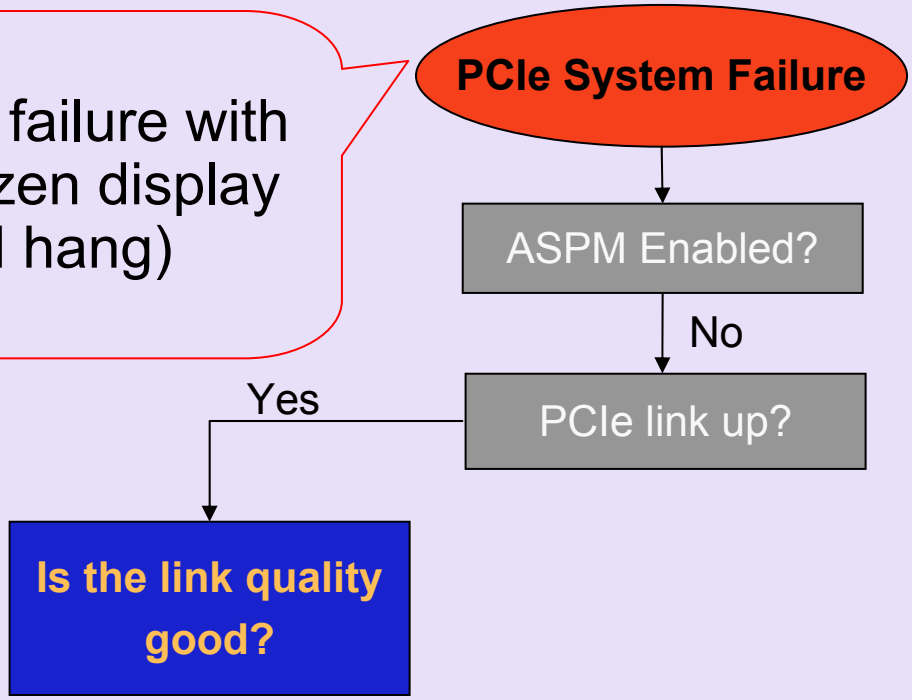


Root Cause

- Sending back-to-back Assert_INTA messages is not a violation of PCIe specification
- Assert_INTA message is like sending the virtual INTA wire active
- One pair of Assert_INTA and Deassert_INTA is out of order
- System does not see the second Assert_INTA
- Hence interrupt is “lost” and software is hung

Example 4

System failure with a frozen display (hard hang)



- How do we determine if the PCIe link quality is good?
  - ✓ Direct method – measuring the eye diagram, BER testing
  - ✓ Indirect method – link quality indicators

Link Quality Indicators

- PCIe errors as defined by the spec
- Correctable Error Status Register
  - ✓ Replay Timer Timeout
  - ✓ Replay Num Rollover
  - ✓ Receiver Errors
- Status bits are set when the error condition is hit once
- Have to write to register to clear it
- Only indicates the presence of the error, but not how many times it happened

More Link Quality Indicators

- NAKs
  - ✓ Trigger on a NAK using logic analyzer
  - ✓ Internal NAK counters
- Link Recoveries
  - ✓ Trigger on training sets using logic analyzer
  - ✓ Internal debug signals and counters
- Link going to detect state
  - ✓ Trigger on electrical idle, TS1 with PADs
  - ✓ Internal LTSSM state vector

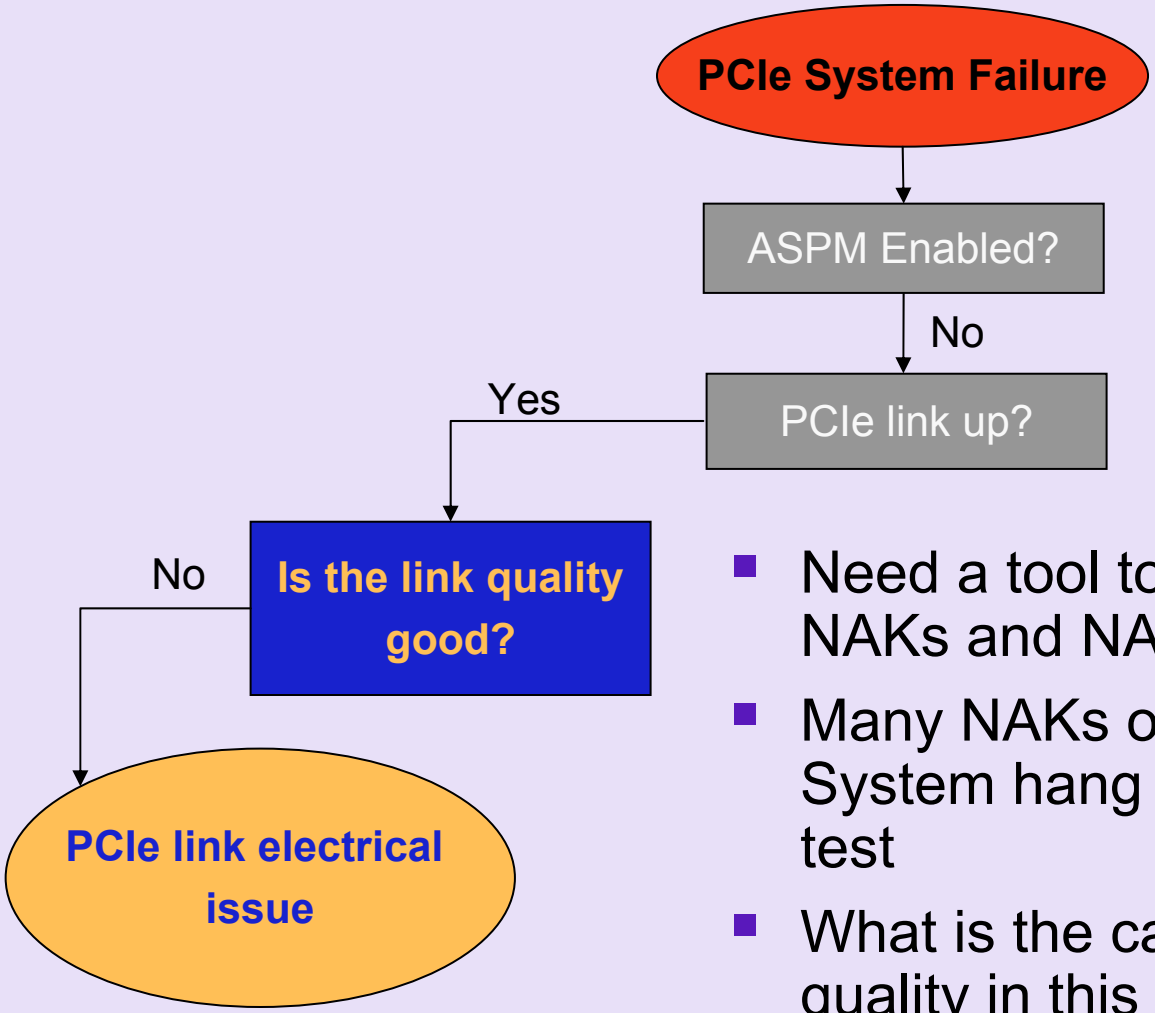
Are NAKs always bad?

- NAKs are generated when:
  - ✓ Bad CRC
  - ✓ Bad sequence number
  - ✓ Physical layer signals receiver error
  - ✓ EDB received (but not a nullified TLP)
- BER 10^{-12}
  - ✓ 1 error / 400s
  - ✓ Less than 1 NAK / 6.7 minutes!
- NAKs and replays are part of the PCI Express protocol
- Should be handled by design
- NAKs are not bad as long as forward progress is made
  - ✓ Performance not noticeably degraded

When are NAKs bad?

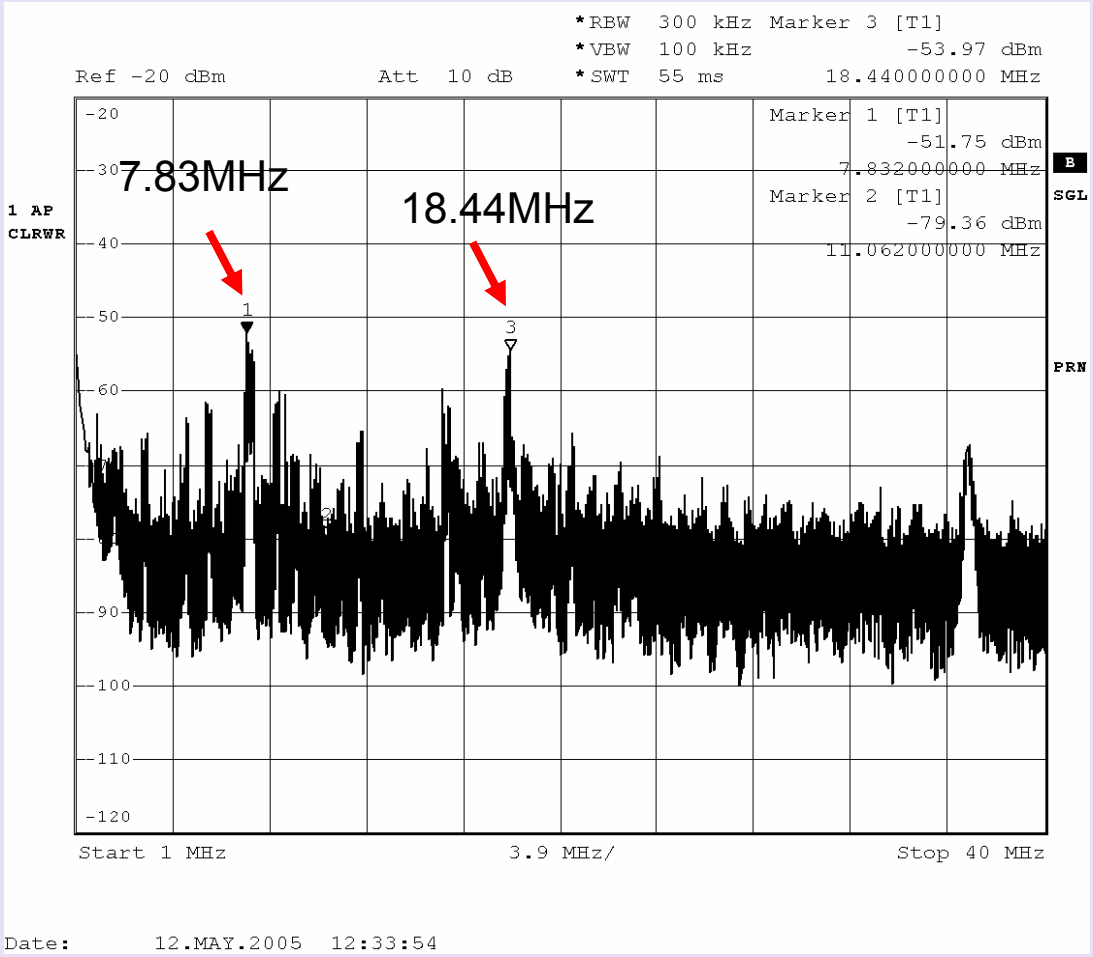
- LINK DOWN is (almost) always bad!
  - ✓ Lost data, system crash
- How to catch link down?
  - ✓ Same as before – looking at link state
  - ✓ Trigger on TS1 PADs
  - ✓ Look at internal LTSSM states (Detect, Polling, Configuration)
- High number of NAKs and link recoveries indicate a risk of link down
  - ✓ How many is “high”?
  - ✓ Absolute threshold difficult to define
  - ✓ Use RELATIVE numbers
  - ✓ Check that performance is acceptable

Back to Example 4…



- Need a tool to monitor the number of NAKs and NAK rate
- Many NAKs on the link prior to System hang when running stressful test
- What is the cause of the poor link quality in this system?

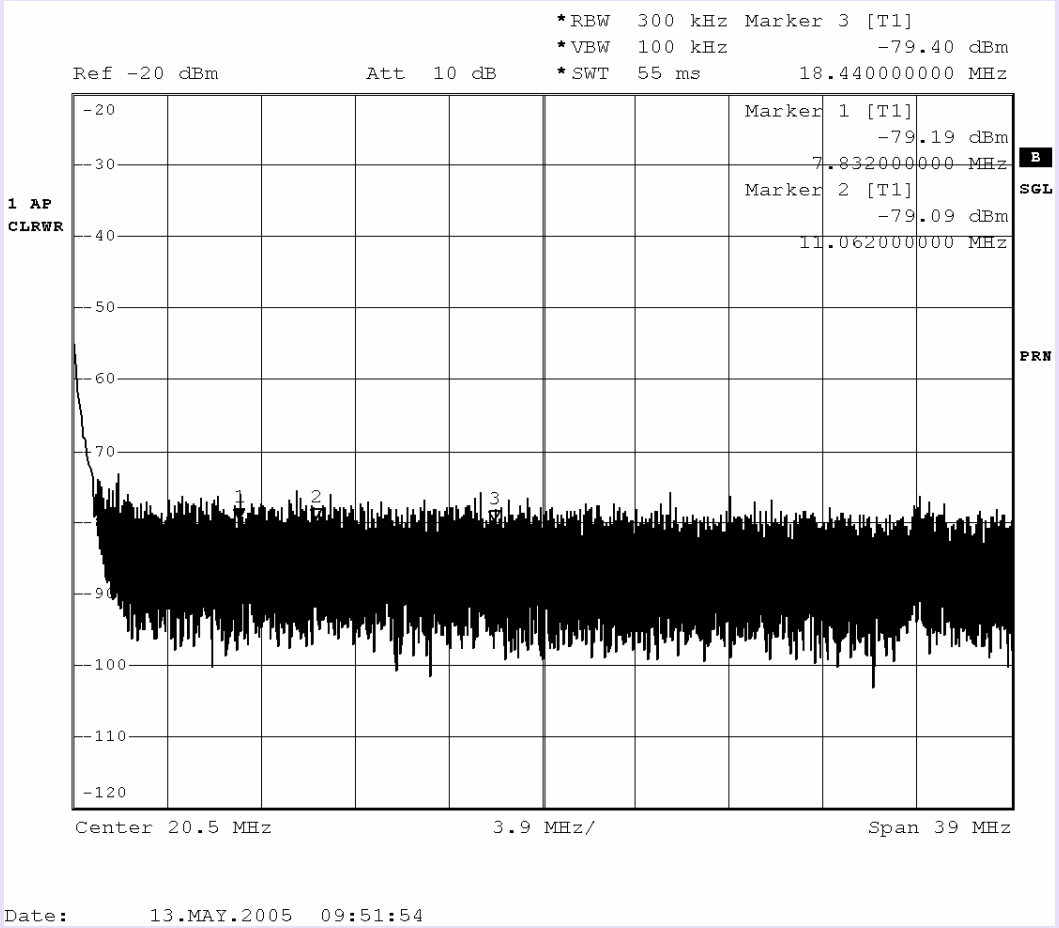
Probing the Power Supply



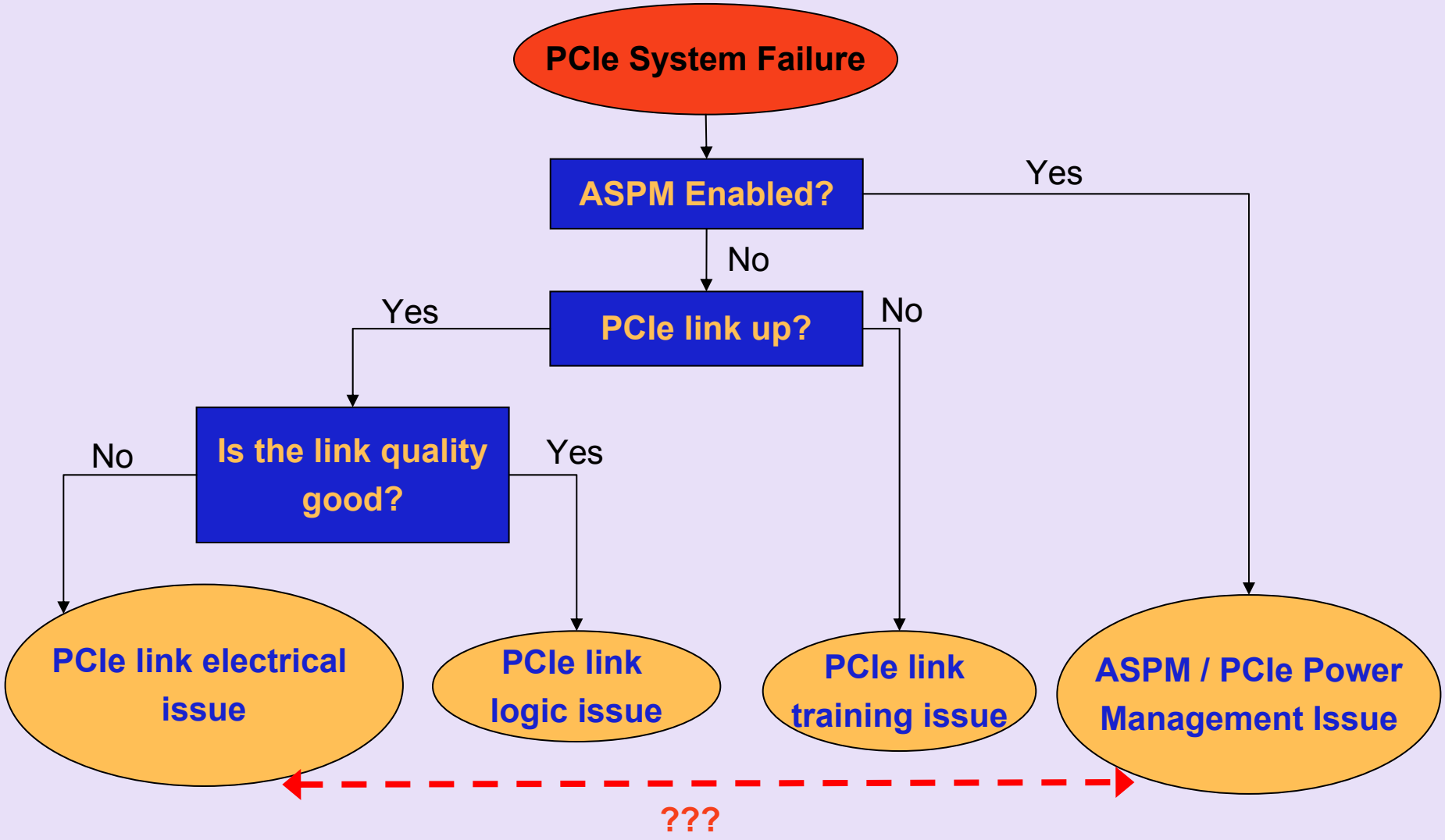
- Probing at package level
- Two noise peaks when rest of chip has a lot of activity

Root Cause

- Add 1.0uF decoupling package caps
- Add 1.0uF decoupling PCB caps
- No NAKs on the link
- Power looks clean with no obvious frequency component



PCI Express Debug Flow



ASPM and Link Quality

- Many ASPM issues are also link quality related
- ASPM L0s and L1 are stressful for the PCIe transmitter and receiver
  - ✓ In and out of electrical idle
  - ✓ Regaining symbol lock
  - ✓ Turning on and off the PLL
- Harder to debug (ASPM enabled on embedded / mobile systems with no access to PCIe bus)
- Need to make use of the link quality indicators
  - ✓ Remember that with L1->Recovery->L0!

Summary

- ASPM enabled?
  - ✓ Does the system fail if ASPM is turned off?
  - ✓ Is the problem related to L1, L0s (which side)?
- PCIE link up?
  - ✓ Software debugger available? Issue a transaction across the bus
  - ✓ PCIe analyzer to look at bus traffic
  - ✓ If all else fails – use a scope!
- Potential link quality problem indicators
  - ✓ Correctable errors
  - ✓ NAKs
  - ✓ Replays (check for replay timer timeout correctable error)
  - ✓ Link recoveries

PCI Express System Debug Toolbox

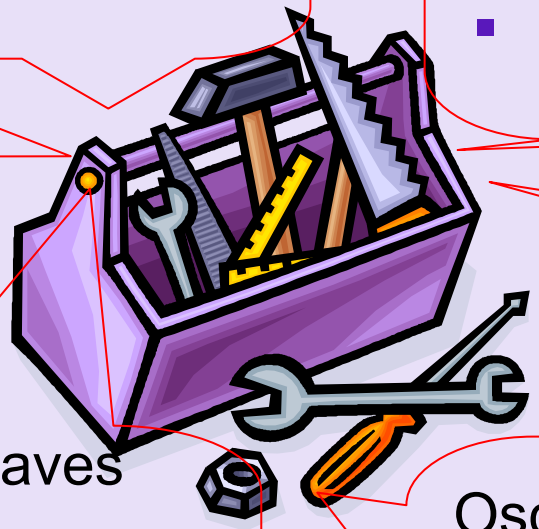
Design-in debug tools

- Link state, internal debug signals, NAK counters

Different link widths

- Riser cards
- Tape up unused lanes
- Remove AC coupling caps

PCIe error reporting



SW debugger

Logic analyzer must-haves

- Conditional storage
- State machine if-then-else triggers
- Counters

Oscilloscope

- Check link state
- Eye diagram
- Power noise

Thank you for attending the
PCI-SIG Developers Conference 2006.

For more information please go to
www.pcisig.com

Acknowledgements

- Ivana Konvalinka
- Stephen Ma
- Steve Manning
- Jagdish Patel
- Gord Caruk