PEX 87xx

Errata

Version 1.14
September 25, 2015

DB05-000345-01

For a comprehensive list of changes to this document, see the Revision History.

Corporate Headquarters
San Jose, CA

Website
www.avagotech.com

Table of Contents

Chapter 1: Errata List

NOTE Throughout this document, unless specified otherwise, "Switch" is used to indicate the PEX 87xx switch.

NOTE For affected products, silicon revisions, and device documentation versions, see Appendix A, Reference Information.

1.1 STRAP_I2C_SMBUS_EN_P Pin Does Not Enable/Disable the SMBUS ARP Mode as Expected

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: Strap pin STRAP_I2C_SMBUS_EN_P does not control the SMBUS ARP enable/disable functionality.

Solution/Workaround: Software/EEPROM/I2C can control the SMBUS ARP mode by programming Port0 Offset 2C8h bit[8].

Impact: STRAP_I2C_SMBUS_EN_P should not be used to control the SMBUS ARP mode.

1.2 ASPM L0s Will Not Work at Gen1, Gen2, and Gen3 Rates

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: When the receive side of the device comes out of L0s, it goes into recovery.

PEX 87xx receivers needs more time to lock than what the max NFTS allows. For this reason the ASPM L0s will not work with the device.

Solution/Workaround: Use ASPM L1 for power saving in the system.

Impact: Power savings cannot be achieved when there are short periods of inactivity. For longer periods of inactivity, power savings can be achieved by ASPM L1.

1.3 When Virtual Switch Mode Is Enabled, VS_PERST# Does Not Initialize the Serial Hot-Plug Register

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: In virtual switch mode, the PEX 87xx scans IO expander and its values when PEX_PERST# deasserts. The PEX 87xx does not re-scan the IO expander again when VS_PERST# deasserts. In this case, the IO expander's setting does not get loaded into the Serial Hot-plug register. In other words, the initial values can be wrong.

Solution/Workaround: Connect VS_PERST# to either HP_BUTTON or HP_MRLn inputs of one of the serial expanders with the corresponding hot-plug pin functionality disabled. The hot-plug pin functionality can be disabled either by EEPROM loading "Slot capability" register or by software clearing the corresponding event enable bit in "slot control" register.

Impact: Serial Hot-plug register reset value could be incorrect after VS_PERST# deassertion.

1.4 PEX 87xx Software Initiated Up-configuration Does Not Work Reliably with Electrical Idle Inference Mode

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX 87xx enable electrical idle inference mode by default. Hardware/Software initiated link width down-configuration works fine with default mode. But link width up-configuration does not reliably bring-up the link to its full width due to analog CDR (Clock and Data Recovery) circuit limitation with electrical idle inference mode.

Solution/Workaround:

1. PEX 87xx autonomous/software initiated link width up-configuration works without Electrical Idle inference mode. EEPROM or Software can be used to disable EI_inference mode (EI_inference is needed for long channels).
2. Software workaround for El_inference mode:

   Down-configuration:

   a. Change the target_link speed to Gen1 (write to port offset 0x98 Bits [3:0] to 0x1).
   b. Change the current operating link speed to Gen1 (write to port offset 0x78 Bit[5] to 1b).
   c. Disable the El_inference mode (PEX 87xx Port0/Port8/Port16 offset 0x204 Bits[15:0] to 0x0000 [see note]).
   d. Drop the Link Width.

   Up-configuration:

   a. Bring-up the Link to full width.
   b. Enable the El_inference mode (PEX 87xx Port0/Port8/Port16 offset 0x204 Bits[15:0] to 0xFFFF [see note]).
   c. Change the target_link speed to Gen3 (write to port offset 0x98 Bits[3:0] to 0x3).
   d. Change the current operating link speed to Gen3 (write port offset 0x78 Bit[5] to 1b).

NOTE Note: Each bit controls the corresponding lane. If the lane(s) are not part of the specified port, leave the default value for the corresponding lanes.

Impact: Software initiated link width up-configuration does not work reliably with Electrical Idle Inference mode. Note, this erratum has no impact on hardware triggered link width changes (link up, bad links etc.).

1.5 PEX 87xx Broken Lane(s) Issue for the Specified Conditions

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description:

1. If the PEX 87xx port links up in the lane reversal mode and lane0 is broken, the link does not negotiate down. Instead, the Switch port times-out in configuration state and goes back to detect state.

2. When two PEX 87xx devices are connected to each other (in x2, x4, x8, and x16 configuration) and the middle lane(s) are broken, but both of the PEX 87xx transmitters still detect the receivers (possible if there is a retimer on the link) the link will never come up. The link should have down-trained and come up as a x1 link but it does not.

Solution/Workaround:

1. None
2. Disable the termination seen by the PEX 87xx transmitter for the broken lane(s).

Impact:

1. Surprise DL_Down event
2. Link never comes up

1.6 PEX 87xx Could Log Receiver Errors at Gen3 Data Rate in Asynchronous Systems with Protocol Aware Retimers that Send Variable Length SKP OS

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: The PCIe device must transmit SKP OS block with 16 SKP symbols at Gen3 data rate. This rule does not apply to re-timers. Protocol aware re-timers could be adding or deleting the SKP symbols when communicating to two different PCIe devices in an asynchronous clocking system. If the scrambled EDS (End of Data stream) pattern looks like SKP OS followed by the actual SKP OS, block alignment logic does not detect SKP at all. The block alignment works fine if the SKP OS has full 130-bit block for this case. But if the SKP OS block is not equal to full block size, the block aligner loses its alignment because it does not detect SKP OS properly.

Solution/Workaround:

1. Synchronous clocking systems are not exposed to this errata.
2. Ensure Re-timer + RC/other PCIe device are in one clock domain and PEX 87xx is in a different clock domain.

Impact: Once the port loses block alignment, it logs receiver error, eventually go to recovery state, and achieves the block alignment in recovery state. There is no speed/width change or no DL_Down due to this state transition

1.7 STRAP_GEN1_GEN2_P Pin Does Not Enable Gen2 Only Mode With Port Bifurcation Feature Enabled

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: STRAP_GEN1_GEN2_P pin is supposed to enable Gen1 only, Gen1/Gen2 only, or Gen1/Gen2/Gen3 data rate mode.

STRAP_GEN1_GEN2_P pin does not enable Gen1/Gen2 only mode.

Solution/Workaround: EEPROM can be used to program Port0 offset 0x46C, Bit[12] to 0, and also to program each port's offset 0x74 Bits[3:0] to 2

Impact: STRAP_GEN1_GEN2_P pin should not be used to configure Gen1/Gen2 only mode.

1.8 PEX 87xx NT Link/NT Virtual Endpoint PCI Compatible Vendor-Specific Compatibility Header Is Incorrect

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PCI compatible Vendor specific capability header must present the capability length field a byte immediately following the "Next Capability pointer" field. But PEX 87xx NT Link/NT virtual endpoints PCI compatible vendor specific capability header in offset 0xC8 present the length field in next CSR offset (0xCC).

Solution/Workaround: No issue is found so far across multiple operating systems. If issue found, the capability can be bypassed using EEPROM for configuration access and use memory mapped access to configure the capability.

Impact: None.

1.9 PEX 87xx Incorrectly Logs Receive Error during Link Width Up-configuration

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX 87xx transitions the lane(s) from electrical idle state to active state during link width up-configuration. The lanes' error reporting logic should be disabled until the lanes achieve bit lock and symbol lock/block alignment. But the port logs code error as soon as the lanes exit electrical idle state.

Solution/Workaround:

1. Clear the receive error as soon as the link width up-configuration completes
2. Do not do link width up-configuration

Impact: The PEX87xx ports log unexpected receiver error after link width up-configuration

1.10 PEX 87XX Upstream Port Incorrectly Sets the Link Status Register's "Link Autonomous Bandwidth Status" Bit for Any Successful Speed Change Event

Risk Category: Low

Silicon Revisions Affected: AA

Description: PEX87xx is supposed to set upstream port Link Status register's "Link Autonomous Bandwidth Status" field if successful_speed_negotiation is set to 1b and the Autonomous Change bit (bit 6 of Symbol 4) is set to 1b for eight consecutive TS2 Ordered Sets received in Recovery.RcvrCfg state. But PEX87xx upstream port always sets Link Status register's "Link Autonomous Bandwidth Status" field for any successful speed change event.

Workaround: None

Impact: Link Status register's "Link Autonomous Bandwidth Status" field set for any successful speed change event

1.11 PEX 87xx Serial Hot-Plug Sequence Should Be Performed One Port at a Time

Risk Category: Medium

Silicon Revisions Affected: AA, AB

Description: When software performs serial hot-plug sequence for more than one port at a time by writing to the Slot Control register's Power Controller Control", "Power Indicator Control", or "Attention Indicator Control" fields, the serial hot-plug controller does not write some port write value to the IO expander.

Solution/Workaround: Software should complete serial hot-plug sequence for a port before switching to another port. Software should enable only the software mechanism for blinking attention and power indicator LEDs by programming slot control register attention and power indicator control fields to 01b/11b (to turn ON and turn OFF) at the required blink rate of 1-2Hz, instead of enabling the hardware autonomous LED blink mechanism by programming these fields to 10b.

Impact: If software writes Slot control register's "Power Controller Control", "Power Indicator Control", or "Attention Indicator Control" for multiple ports at the same time, serial hot-plug controller drops the write for some port.

1.12 Serial Hot-Plug

Risk Category: Low

Silicon Revisions Affected: AA, AB

1.12.1 8796

Summary: PEX8796 Port5, Port9, Port13, Port17, and Port21 Serial hot-plug works only if Port0 is not a x16 port in the default mode.

Description: Table 1 and Table 2 describe the expected behavior per the product specification, however a design defect places certain limitations illustrated in Table 3, Table 4, and Table 5 on default usage. An EEPROM can be used to assign the correct ports and workaround the defect.

If all PEX8796 ports are configured as x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 1 Expected Behavior

	Port																						
	0-x 16	1	2	3	4-x 16	5	6	7	8-x 16	9	10	11	12-x16	13	14	15	16-x16	17	18	19	20-x16	21	22	23
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA

If all PEX8796 ports are configured as x8 or lower width, the parallel hot-plug and serial hot-plug ports are assigned.

Table 2 Expected Behavior

	Port																						
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**	**12 is not x16**	**13**	**14**	**15**	**16 is not x16**	**17**	**18**	**19**	**20 is not x16**	**21**	**22**	**23**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

If PEX8796 port0 is x16 (irrespective of other port widths), the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to a design bug. The serial hot-plug functionality does not work for Port5, Port9, Port13, Port17, and Port21. PEX8796 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 3 Default Usage

	Port																						
	0-x 16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes

If PEX8796 Port0 is not a x16 port and all other ports are x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. The serial hot-plug functionality works fine in this case. PEX8796 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 4 Default Usage

	Port																						
	0 is not x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Parallel Hot-plug	No	Yes	No	No	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA
Serial Hot-plug	Yes	No	Yes	Yes	Yes	NA	NA	NA	Yes	NA	NA	NA	Yes	NA	NA	NA	Yes	NA	NA	NA	Yes	NA	NA	NA

If all PEX8796 ports are not x16 port, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. Both serial and parallel hot-plug functionality works in this case. PEX8796 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 5 Default Usage

	Port																							
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**	**12 is not x16**	**13**	**14**	**15**	**16 is not x16**	**17**	**18**	**19**	**20 is not x16**	**21**	**22**	**23**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

Workaround: EEPROM can be used to associate the parallel hot-plug functionality.

1. Program Port0 offset 0x33C Bits[4:0] to map parallel hot-plug to Port0 to Port3.
2. Program Port4 offset 0x33C Bits[4:0] to map parallel hot-plug to Port4 to Port7.
3. Program Port8 offset 0x33C Bits[4:0] to map parallel hot-plug to Port8 to Port11.
4. Program Port12 offset 0x33C Bits[4:0] to map parallel hot-plug to Port12 to Port15.
5. Program Port16 offset 0x33C Bits[4:0] to map parallel hot-plug to Port16 to Port19.
6. Program Port20 offset 0x33C Bits[4:0] to map parallel hot-plug to Port20 to Port23.

Impact: PEX8796 Port5, Port9, Port13, Port17, and Port21 Serial hot-plug does not work if Port0 is a x16 port in the default mode.

1.12.2 8780

Summary: PEX8780 Port5, Port9, Port13, and Port17 Serial hot-plug works only if Port0 is not a x16 port in the default mode.

Description: Table 6 andTable 7 describe the expected behavior per the product specification, however a design defect places certain limitations illustrated in Table 8, Table 9, and Table 10on default usage. An EEPROM can be used to assign the correct ports and workaround the defect.

If all PEX8780 ports are configured as x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 6 Expected Behavior

	Port																			
	0-x16	**1**	**2**	**3**	**4-x16**	**5**	**6**	**7**	**8-x16**	**9**	**10**	**11**	**12-x16**	**13**	**14**	**15**	**16-x16**	**17**	**18**	**19**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA

If all PEX8780 ports are configured as x8 or lower width, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 7 Expected Behavior

	Port																			
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**	**12 is not x16**	**13**	**14**	**15**	**16 is not x16**	**17**	**18**	**19**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

If PEX8780 port0 is x16 (irrespective of other port widths), the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to a design bug. The serial hot-plug functionality does not work for Port5, Port9, Port13, and Port17. PEX8780 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 8 Default Usage

	Port																			
	0-x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes

If PEX8780 Port0 is not a x16 port and all other ports are x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. The serial hot-plug functionality works fine in this case. PEX8780 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 9 Default Usage

	Port																			
	0 is not x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Parallel Hot-plug	No	Yes	No	No	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA
Serial Hot-plug	Yes	No	Yes	Yes	Yes	NA	NA	NA	Yes	NA	NA	NA	Yes	NA	NA	NA	Yes	NA	NA	NA

If all PEX8780 ports are not x16 port, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. Both serial and parallel hot-plug functionality works in this case. PEX8780 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 10 Default Usage

	Port																			
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**	**12 is not x16**	**13**	**14**	**15**	**16 is not x16**	**17**	**18**	**19**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

Workaround: EEPROM can be used to associate the parallel hot-plug functionality.

1. Program Port0 offset 0x33C Bits[4:0] to map parallel hot-plug to Port0 to Port3.
2. Program Port4 offset 0x33C Bits[4:0] to map parallel hot-plug to Port4 to Port7.
3. Program Port8 offset 0x33C Bits[4:0] to map parallel hot-plug to Port8 to Port11.
4. Program Port12 offset 0x33C Bits[4:0] to map parallel hot-plug to Port12 to Port15.
5. Program Port16 offset 0x33C Bits[4:0] to map parallel hot-plug to Port16 to Port19.

Impact: PEX8780 Port5, Port9, Port13, and Port17 Serial hot-plug does not work if Port0 is a x16 port in the default mode.

1.12.3 8764

Summary: PEX8764 Port5, Port9, and Port13 Serial hot-plug works only if Port0 is not a x16 port in the default mode.

Description: Table 11 and Table 12 describe the expected behavior per the product specification, however a design defect places certain limitations illustrated inTable 13, Table 14, and Table 15 on default usage. An EEPROM can be used to assign the correct ports and workaround the defect.

If all PEX8764 ports are configured as x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 11 Expected Behavior

	Port															
	0-x16	**1**	**2**	**3**	**4-x16**	**5**	**6**	**7**	**8-x16**	**9**	**10**	**11**	**12-x16**	**13**	**14**	**15**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA

If all PEX8764 ports are configured as x8 or lower width, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 12 Expected Behavior

	Port															
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**	**12 is not x16**	**13**	**14**	**15**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

If PEX8764 port0 is x16 (irrespective of other port widths), the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to a design bug. The serial hot-plug functionality does not work for Port5, Port9, and Port13. PEX8764 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 13 Default Usage

	Port															
	0-x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes

If PEX8764 Port0 is not a x16 port and all other ports are x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. The serial hot-plug functionality works fine in this case. PEX8764 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 14 Default Usage

	Port															
	0 is not x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Parallel Hot-plug	No	Yes	No	No	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA
Serial Hot-plug	Yes	No	Yes	Yes	Yes	NA	NA	NA	Yes	NA	NA	NA	Yes	NA	NA	NA

If all PEX8764 ports are not x16 port, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. Both serial and parallel hot-plug functionality works in this case. PEX8764 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 15 Default Usage

	Port															
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**	**12 is not x16**	**13**	**14**	**15**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

Workaround: EEPROM can be used to associate the parallel hot-plug functionality.

1. Program Port0 offset 0x33C Bits[4:0] to map parallel hot-plug to Port0 to Port3.
2. Program Port4 offset 0x33C Bits[4:0] to map parallel hot-plug to Port4 to Port7.

3. Program Port8 offset 0x33C Bits[4:0] to map parallel hot-plug to Port8 to Port11.
4. Program Port12 offset 0x33C Bits[4:0] to map parallel hot-plug to Port12 to Port15.

Impact: PEX8764 Port5, Port9, and Port13 Serial hot-plug does not work if Port0 is a x16 port in the default mode.

1.12.4 8750

Summary: PEX8750 Port5, and Port9 Serial hot-plug works only if Port0 is not a x16 port in the default mode.

Description: Table 16 and Table 17 describe the expected behavior per the product specification, however a design defect places certain limitations illustrated in Table 18, Table 19, and Table 20 on default usage. An EEPROM can be used to assign the correct ports and workaround the defect.

If all PEX8750 ports are configured as x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 16 Expected Behavior

	Port											
	0-x16	**1**	**2**	**3**	**4-x16**	**5**	**6**	**7**	**8-x16**	**9**	**10**	**11**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	NA	NA	NA	No	NA	NA	NA

If all PEX8750 ports are configured as x8 or lower width, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 17 Expected Behavior

	Port											
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

If PEX8750 port0 is x16 (irrespective of other port widths), the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to a design bug. The serial hot-plug functionality does not work for Port5, and Port9. PEX8750 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 18 Default Usage

	Port											
	0-x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	No	Yes	Yes	No	No	Yes	Yes

If PEX8750 Port0 is not a x16 port and all other ports are x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. The serial hot-plug functionality works fine in this case. PEX8750 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 19 Default Usage

	Port											
	0 is not x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Parallel Hot-plug	No	Yes	No	No	No	NA	NA	NA	No	NA	NA	NA
Serial Hot-plug	Yes	No	Yes	Yes	Yes	NA	NA	NA	Yes	NA	NA	NA

If all PEX8750 ports are not x16 port, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. Both serial and parallel hot-plug functionality works in this case. PEX8750 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 20 Default Usage

	Port											
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**	**8 is not x16**	**9**	**10**	**11**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes

Workaround: EEPROM can be used to associate the parallel hot-plug functionality.

1. Program Port0 offset 0x33C Bits[4:0] to map parallel hot-plug to Port0 to Port3.
2. Program Port4 offset 0x33C Bits[4:0] to map parallel hot-plug to Port4 to Port7.
3. Program Port8 offset 0x33C Bits[4:0] to map parallel hot-plug to Port8 to Port11.

Impact: PEX8750 Port5, and Port9 Serial hot-plug does not work if Port0 is a x16 port in the default mode.

1.12.5 8734

Summary: PEX8734 Port5 Serial hot-plug works only if Port0 is not a x16 port in the default mode.

Description: Table 21 and Table 22 describe the expected behavior per the product specification, however a design defect places certain limitations illustrated inTable 23, Table 24, and Table 25 on default usage. An EEPROM can be used to assign the correct ports and workaround the defect.

If all PEX8734 ports are configured as x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 21 Expected Behavior

	Port							
	0-x16	**1**	**2**	**3**	**4-x16**	**5**	**6**	**7**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	NA	NA	NA

If all PEX8734 ports are configured as x8 or lower width, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table.

Table 22 Expected Behavior

	Port							
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes

If PEX8734 port0 is x16 (irrespective of other port widths), the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to a design bug. The serial hot-plug functionality does not work for Port5. PEX8734 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 23 Default Usage

	Port							
	0-x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Parallel Hot-plug	Yes	No	No	No	Yes	No	No	No
Serial Hot-plug	No	NA	NA	NA	No	No	Yes	Yes

If PEX8734 Port0 is not a x16 port and all other ports are x16, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. The serial hot-plug functionality works fine in this case. PEX8734 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 24 Default Usage

	Port							
	0 is not x16	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Parallel Hot-plug	No	Yes	No	No	No	NA	NA	NA
Serial Hot-plug	Yes	No	Yes	Yes	Yes	NA	NA	NA

If all PEX8734 ports are not x16 port, the parallel hot-plug and serial hot-plug ports are assigned as defined in the following table due to design bug. Both serial and parallel hot-plug functionality works in this case. PEX8734 does not have this bug if EEPROM is used for parallel hot-plug assignment.

Table 25 Default Usage

	Port							
	0 is not x16	**1**	**2**	**3**	**4 is not x16**	**5**	**6**	**7**
Parallel Hot-plug	No	Yes	No	No	No	Yes	No	No
Serial Hot-plug	Yes	No	Yes	Yes	Yes	No	Yes	Yes

Workaround: EEPROM can be used to associate the parallel hot-plug functionality.

1. Program Port0 offset 0x33C Bits[4:0] to map parallel hot-plug to Port0 to Port3.
2. Program Port4 offset 0x33C Bits[4:0] to map parallel hot-plug to Port4 to Port7.

Impact: PEX8734 Port5 Serial hot-plug does not work if Port0 is a x16 port in the default mode.

1.13 Subset of Pins Are Accessible by BSCAN, an Errata Prevents Access to Remaining Pins

Risk Category: Low

Silicon Revisions Affected: AA

Devices Impacted: PEX8796, PEX8780, PEX8764, PEX8750, and PEX8734

Description: The first 16 lanes of SerDes and a partial set of digital pins can be tested using JTAG. DeviceID read, Bypass instruction, etc., are not impacted. Please refer to version 1.0 of BSDL files to see what pins can be tested by BSCAN.

Solution/Workaround: Functional tests need to be used to test pin connectivity on the remaining pins.

Impact: Both DC and AC JTAG are affected by this.

1.14 PEX87xx Downstream Port Does Not Change the Link Speed from Gen2 to Gen3 in Single Retrain for the Specified Rate Change Sequence

Risk Category: Low

Silicon Revisions Affected: AA

Description: If software changes PEX87xx downstream link data rate from Gen1 to Gen2 to Gen3 by changing both upstream and downstream component's target_link_speed, Gen1 to Gen2 speed change works fine. But PEX87xx downstream port does not initiate speed change from Gen2 to Gen3 with single retrain. It expects software to write "retrain link" bit twice to initiate Gen2 to Gen3 speed change.

Solution/Workaround:

1. Keep the downstream component's "target_link_speed" to it's maximum supported data rate. Change only PEX87xx downstream port's "target_link_speed" to change the link data rate.
2. If Software needs to change the downstream component's target_link_speed, it needs to write PEX87xx downstream port "Retrain Link" bit two times.

Impact: PEX87xx downstream port does not initiate Gen2 to Gen3 speed change with single "Retrain Link".

1.15 PEX8xxx Implements MC_Base_Address as Defined in the Description

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PCIe specification states MC_Base_Address should be implemented as 2 ^ MC_Index_Position. But PEX87xx implements MC_Base_Address as follows:

1. If MC_Num_Group is 0x0, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 0)
2. If MC_Num_Group is 0x1, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 1)
3. If MC_Num_Group is 0x2 to 0x3, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 2)
4. If MC_Num_Group is 0x4 to 0x7, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 3)
5. If MC_Num_Group is 0x8 to 0xF, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 4)

6. If MC_Num_Group is 0x10 to 0x1F, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 5)
7. If MC_Num_Group is 0x20 to 0x3F, MC_Base_Address is implemented as 2 ^ (MC_Index_Position + 6)

This causes no multicast hit if the MC_Base_Address programming is not aligned to the power of MC_Index_Position+"n" (where n is 0 to 6).

Solution/Workaround: MC_Base_Address programming needs to be aligned to the power of MC_Index_Position+'n"

Impact: No multicast hit if the MC_Base_Address programming is not aligned to the power of MC_Index_Position+"n"

1.16 PEX87xx Generated MSI-64 TLP Payload Might Not Be the MSI Data Register Value with Specific Incoming Traffic Pattern

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx internally generated 64-bit MSI TLP payload supposed to carry the MSI Data Register value with vector related modification. But it generates 64-bit MSI TLP with upper MSI address as payload value if the corresponding PEX87xx port receives a specific traffic pattern.

Solution/Workaround:

1. Use INTx mechanism
2. Use MSI-32 bit mechanism by programming PEX87xx Port's offset 0x48 Bit[23] to 0b using EEPROM/I2C
3. Use MSI-64 bit mechanism by ignoring the MSI-64 TLP's payload. But modify the software to poll the PEX87xx port's interrupt event instead of relying on MSI-64 TLP payload's to derive the information.

Impact: MSI-64 TLP payload may contain incorrect value

1.17 Under an Extreme Corner Case Condition, PEX87xxAA Could Corrupt Internally Generated TLPs and/or TLPs Being Routed by the Switch

Risk Category: Low

Silicon Revisions Affected: AA

Description: The following sequence of TLPs could cause TLP corruption for either the internally generated TLP and/or TLPs routed by the Switch.

- PEX87xxAA generates an internal TLP immediately followed by (no gap)
- The PEX87xxAA Port0/Port1/Port2/Port3 receives at least 20 back to back short TLPs(less than 7 DWs, Header and payload together) with a specific alignment with no gap followed by
- PEX87xxAA generating another internal TLP immediately (no gap).

Solution/Workaround: Disable level sensitive interrupts & avoid back to back configuration transactions with normal traffic. For systems using NT, disable doorbell interrupts.

Impact: Data corruption for the PEX87xxAA internally generated TLPs and/or the TLPs being routed by the Switch

1.18 PEX87xx Incorrectly Calculates the Gen3 TS1 Parity in Response to use_preset Equalization Request from the Link Partner

Risk Category: Low

Silicon Revisions Affected: AA

Description: If PEX87xx port receives the use_preset request from the link partner during Equalization Phase2 and Phase3, it responds by transmitting the Gen3 TS1 with incorrect parity. If the link partner tunes the PEX87xx transmitter using coefficients request instead of preset requests, the parity is calculated right.

Solution/Workaround: Link partner to use coefficient request instead of use_preset request for tuning PEX87xx transmitter

Impact: The TS1 transmitted by PEX87xx will get rejected by the link partner

1.19 See Erratum 26

1.20 Register A30h Accessible from all VS's in Virtual Switch Mode

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: The register A30h is a virtual switch register. This register is accessible from any VS that owns NT0. However it is not accessible from a VS that owns NT1.

Solution/Workaround: This register needs to be written from the upstream/NT port of the VS that owns NT0

Impact: Virtual switch owning NT1 cannot access register A30h if it does not own NT0

1.21 PEX8796/8780/8764 Does Not Load the EEPROM for Port12 or Greater Port Number in Virtual Switch Mode hot-reset, port_reset, or VS_PERST# Reset

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX8796 does not load the EEPROM for Port12 to Port23 in Virtual Switch Mode hot-reset, port reset, or VS_PERST# reset.

PEX8780 does not load the EEPROM for Port12 to Port19 in Virtual Switch Mode hot-reset, port reset, or VS_PERST# reset.

PEX8764 does not load the EEPROM for Port12 to Port15 in Virtual Switch Mode hot-reset, port reset, or VS_PERST# reset.

EEPROM loading works fine in Base Mode for all reset types, and in Virtual Switch Mode for PEX_PERST# reset.

Solution/Workaround: None if the configuration space register (CSR) is resetted by port specific hot-reset, port-reset, or VS_PERST#.

If the configuration space register is not resetted by port specific hot-reset, port reset, or VS_PERST#, EEPROM loading from PEX_PERST# stays unless software changes the value.

Impact: EEPROM does not get loaded for the specified ports.

1.22 See Erratum 26

1.23 False Detection of Receiver for Long Cable Lengths

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: 87xx devices can falsely detect receivers when there is no endpoint connected to them if there is a long cable attached to the Tx pins(high trace capacitance). This has been seen when cables attached to Tx pins are typically more than 5 feet in length. With this modification Cables in excess of 3-5 feet can still result in false detection.

Solution/Workaround: Write the following values in register BFC:

- 0x8E097F for Quad0
- 0xAE097F for Quad 1
- 0xCE097F for Quad2
- 0xEE097F for Quad3

Impact: Customers using long cables to PCie links can falsely detect receivers when there is nothing connected at the end of the cable.

1.24 87xx WRR Port Arbitration Table Inconsistencies

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: 87xx WRR port arbitration table implementation has the following issues.

1. If upstream port implements the WRR port arbitration table, WRR port arbitration table has to have atleast one entry programmed with upstream port number for configuration space access to work
2. If WRR port arbitration table is implemented by station's Port1, Port2, Port3, or Port4 upstream port
   - Configuration write to WRR port arbitration table is ignored and responds with successful completion status
   - Configuration read to WRR port arbitration table returns all 0s.

Solution/Workaround: For issue1: There is no issue with hardware default assignment. If the customer reprograms the WRR port arbitration table slots assignment, program the upstream port number to at least one slot of WRR port arbitration table.

For issue2: Enable Port0 as an upstream port with no upstream port assignment on station's Port1, Port2, Port3, and Port4, or use memory mapped access to access the WRR port arbitration table if station's Port1, Port2, Port3, or Port4 are assigned as upstream port(s)

Impact: For issue1: If upstream port number is not programmed to WRR port arbitration table, all capella1/capella2 configuration space access gets completion timeout

For issue2: Configuration access does not access the WRR port arbitration table

1.25 PEX87xx Upstream Port's Lane Reversal with Link Width Down Negotiation Does Not Work for the Specified Condition

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx upstream port does not link up if its lanes are reversed, the remote device tries to link-up in lower link width, and the remote device sends training sequence 1 (TS1) with Link number set to non-PAD on all the lower lanes that it intends to negotiate-out in configuration.Linkwidth.Start state. Typically, the remote device sends Link number set to PAD on all the lanes that it intends to negotiate-out in configuration.linkwidth.start. If PEX87xx upstream port receives TS1 with Link number set to PAD in configuration.linkwidth.start, it links-up fine.

Solution/Workaround:

1. Disable the Lane reversal.
2. Configure the PEX87xx upstream port programmed link width to the remote device's Link width.

Impact: PEX87xx upstream port does not link up for the above specified condition

1.26 PEX87xx Gen3, x16 ports with Sequences of Back-to-Back Short TLPs along with Long TLP(s) Could Result in Transmitted TLPs with LCRC Error and Possible Surprise Link Down when Aggregating traffic

Risk Category: Medium

Silicon Revisions Affected: AA, AB

Description: If PEX87xx is configured as a Gen3 x16 port, it could transmit TLPs with LCRC error depending on the traffic load, pattern, and if that pattern has multiple back to back short TLPs with long TLP(s) interspersed. The problem was observed as the result of an Ethernet NIC (T10Diff) loopback test which creates and uses encapsulated (T10Diff) packets atop PCIe. This test creates high traffic loads and short packets on the downstream ports of the switch which get aggregated by PEX87xx Gen3 x16 upstream port (Aggregation port Gen3, x16). The aggregation of these packets by PEX87xx Gen3 x16 upstream port highlighted the problem. For Gen3 x16 port(s), if more than 9 back to back short TLPs (20bytes or less, Header + PayLoad) is followed by long TLP(s), and if traffic is sufficient to create internal FIFO backpressure, the port could send a LCRC corrupted packet out and stay in TLP transmission mode until a physical PERST#. The port does not respond to any recovery protocol after this condition. The port goes into this unexpected state due to the logic compressing the short TLPs to efficiently utilize the egress link bandwidth if the ingress link width bandwidth is not fully utilized efficiently.

Solution/Workaround: Program Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBE8 Bit[20] to 1b. This disables logic from compressing the short TLPs. The port transmits the TLP out as it receives on its ingress port(s)

Impact: Gen3 x16 port sends out LCRC corrupted packets, stays in TLP transmission mode forever and doe s not respond to any physical layer or data link layer requests.

1.27 Bandwidth Limitations for Short TLPs in Station-to-Station Traffic

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: TLPs with less than 128 Bytes of payloads can have marginal bandwidth issues when they are going from one station to the other in 4 or 6 station devices. Kindly refer to the Capella Bandwidth limitation White paper that describes it in detail.

Solution/Workaround: If port configurations can be changed, use the suggested port configurations from the white paper to reduce the effect. Flows with longer TLPs are not affected by this issue. If multiple flows are interleaved going from one station to 2 different stations, this issue will not be seen

Impact: Reduced bandwidth for short TLPs going from one station to another

1.28 PEX87xx Does Not Infer Electrical Idle in Detect State when port_disable/VS_PERST# Assertion Occur during Hot-Reset Sequence

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx does not infer electrical idle in Detect state when the port has received training sequence(s) in hot-reset state followed by port_disable/VS_PERST# assertion occur before the port goes to Detect state. Otherwise, PEX87xx infers electrical idle in Detect State.

Solution/Workaround:

1. Disable hot-reset in the system or do not assert port_disable(CSR offset 0x208[5:0])/VS_PERST# when the port is in hot-reset state
2. Disable the electrical idle inference (Station's Port0 0x204[15:0] is 0x0000 and 0x220[29:24] is 0x3f) for short channels at Gen3 data rate or any channel length at Gen1/Gen2 data rate
3. For long channels:
   a. Change the link speed to Gen1 data rate (Port's offset 0x98[1:0] is 01b followed by Link retrain-Port's offset 0x78[5] is 1b).
   b. Disable the electrical idle inference before initiating the hot-reset sequence (Station's Port0 0x204[15:0] is 0x0000 and 0x220[29:24] is 0x3f).
   c. Once the link comes-up in Gen1 data rate after completing the hot-reset sequence, enable the electrical idle inference before going into Gen3 data rate.
4. Enable the "Link downtrain disable" feature (Station's Port0 offset 0xBE4[5:0])

NOTE This workaround causes additional Link bring-up after getting configuration state timeout

Impact: PEX87xx could link-up in lower link width typically in virtual switch mode.

1.29 PEX87xx Does Not Enter ASPM-L1/PCI-PM-L1 State If the Non-Posted Payload Credit Available from Remote Device Is Less Than 1 MPS

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx is supposed to enter ASPM-L1 or PCI-PM-L1 state with other conditions satisfied if the non-posted payload credit available from remote device is sufficient enough to transmit a maximum payload architected non-posted TLP. Non-posted TLP's maximum payload size could consume up to 2 payload credits. But PEX87xx waits for 1 MPS size before it enters ASPM-L1/PCI-PM-L1 state.

Solution/Workaround: Have remote device to advertise at least 1 MPS payload credit for non-posted payload type.

Impact: No L1 state power saving if the non-posted payload credit available from remote device is less than 1 MPS. This issue was observed with the use of the Intel Haswell platform. With Haswell platform, L1 entry is not possible.

Chapter 2: Cautions

NOTE Cautions are not errata. In cautions, the switch is not violating the PCIe specification. However, the switch might be behaving unexpectedly due to another non-compliant device connected to the switch or due to ambiguity in the specification itself.

NOTE For affected products, silicon revisions, and device documentation versions, see Appendix A, Reference Information.

2.1 Link to Intel Based Systems Are Limited to 19dB Insertion Loss Channel

Risk Category: High

Silicon Revisions Affected: AA, AB

Description: Due to margin requirements when interfacing to Intel® Xeon ® processor E5-2600 product family (formerly codenamed Romley) and future Intel® Xeon® processor E5 family products based PCIe Gen3 systems, it is recommended to limit the channel insertion loss at 4GHz to 19dB (note that this is 4dB less than the maximum channel loss communicated in the Intel Product Design Guide). The transmitter may not provide enough boost and the proper proportions of pre-shoot and de-emphasis for the signal to be correctly received by the Intel receiver for channels with greater insertion loss than 19dB. This 19dB applies to the entire channel between PLX TX and Intel RX, including package loss. PLX switch package loss is 0.5dB.

NOTE Channel loss restriction does not apply in the other direction (Intel TX to PLX RX) because the PLX receivers do not have the same restriction as the Intel receivers. Loss between two PLX devices can be greater than 25 dB for well-designed channels.

2.2 PEX 87xx Port Replay Timer Equation Does Not Include "Extended Sync" Bit into the Calculation

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: The PCI Express device should adjust the replay timer value if software sets "extended sync" bit to 1'b1. This is to avoid unexpected replay timeouts and recoveries. The "extended sync" CSR bit intended for debug purpose only. PLX Switch supports the device specific register to multiply the "default" replay timer value by up to 256 times.

Solution/Workaround: Multiply the replay timer value by programming Port's offset FA8h Bits[23:16] with the proper multiplication factor (eg., Bits[23:16] == 8'h1, multiplies the replay timer by 2, Bits[23:16] == 8'h2, multiplie3s the replay timer by 4, etc.).

2.3 PEX 87xx Port Times Out in Configuration State in Lane(s) Breaks after Polling

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: The PEX 87xx port should negotiate the broken receive lane(s) in configuration state even if they are good in polling state. But the Port times out in configuration state and go to detect state. If the lane(s) is/are still broken in polling state, the PEX 87xx port negotiate down the port to the next legal width. The PEX 87xx port negotiate out the lane correctly if they are broken from polling state even if it detects receiver.

Solution/Workaround: None.

Impact: Link bring-up takes extra time for the dynamic broken lane condition.

2.4 PEX 87xx Port Sends TS1 Ordered Set with Incorrect Link Number for the Specified Broken Lane Condition with Dynamic Lane Reversal

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: The external device connected to the PEX 87xx port exchanged the right link number and lane numbers in configuration.lanenum.wait substate. If the external device receiver's lower lanes are broken in this configuration sub-state and is trying to lane reverse the Link, PEX 87xx port sends incorrect link number and finally timeout in configuration.lanenum.wait substate. PEX 87xx port go back to Detect state and come up with right link width and lane reversed configuration.

Solution/Workaround: None.

Impact: Port takes extra time to link up with dynamic broken lane and dynamic lane reversal condition.

2.5 PEX 87xx to PEX 87xx Link Does Not Come Up for Lane Reversal Configuration With Inner Broken Lane(s)

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: "Inner broken" is the lane number from 1 to n-2 (e.g. for x8 link, lane 1 to lane6 are the inner lanes). If PCIe link is in lane reversal mode with inner broken lane(s), it should negotiate out the broken lanes and link-up in lower width.

But PEX 87xx to PEX 87xx link does not link-up if the lanes are reversed and if one or more inner lanes are broken. PEX 87xx port does link-up properly with non-PLX devices if lanes are reversed and one or more inner lanes are broken.

Solution/Workaround: None

Impact: None

2.6 Gen2 Loopback Entered From Recovery State Always Uses -6db De-emphasis

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: When Loopback state is entered from Recovery state and the port is a Loopback slave, the Selectable De-emphasis bit is not saved from the training sets received during Loopback.Entry state.

Solution/Workaround: Enter Loopback state from Configuration state.

Impact: When the link speed is Gen 2, and Loopback state is entered from Recovery state, the PEX 87xx port will only transmit at –6 dB.

2.7 PEX87xx Port Does Not Respond with Received Preset for Reserved Preset Request

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: If PEX87xx port receives the reserved preset request, it should respond by transmitting TS1s with "reject_coefficient" bit set and the received preset values. The PEX87xx transmits TS1s with "reject coefficient" bit set, but it does not transmit the received reserved preset value in the preset field.

Solution/Workaround: None.

Impact: Devices should not use reserved preset for tuning.

2.8 Software Back-to-Back EEPROM Accesses Do Not Work with Default EEPROM Clock Speed

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: If software performs back to back accesses to EEPROM control CSR offsets (0x260 and 0x264), the EEPROM controller may fail to access the EEPROM contents correctly for subsequent accesses. Hardware loading the EEPROM contents to the Configuration Space Registers(CSRs) works fine.

Solution/Workaround:

1. Change the EEPROM clock frequency from its default 1MHz to 5MHz or greater by programming port 0 CSR 0x268[3:0] to 0x2 or greater. This CSR write should be the first entry in the EEPROM.
2. Software can delay the back to back EEPROM accesses by at least 2.5us gap.

Impact: Software reads can be corrupted and writes can be dropped for subsequent accesses.

2.9 PEX87xx Memory Mapped Access to Specific NT Link Space Registers Does Not Return Completion when NT_PERST# Is Asserted

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx is supposed to return successful completion with proper data if host from upstream port side accesses the NT Link space offsets f8h, c4ch, c50h, and c54h. PEX87xx does not return any completion for these offsets.

Solution/Workaround: Avoid accessing NT link offsets f8h, c4ch, c50h, c54h if NT_PERST# is asserted.

Impact: Completion timeout occurs if host accesses NT link offsets f8h, c4ch, c50h, and c54h

2.10 PEX87xx Uses TLP FBE field Bit[0] to Write byte0 to byte3 of GPIO Offsets 0x604, 0x608, and 0x60c

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx should use TLP FBE field Bit[0] to write byte0 of GPIO offsets, TLP FBE field Bit[1] to write byte1 of GPIO offset, TLP FBE field Bit[2] to write byte2 of GPIO offset, etc. PEX87xx incorrectly uses TLP FBE field Bit[0] to write byte1, byte2, and byte3 of GPIO offsets 0x604, 0x608, and 0x60c.

Solution/Workaround:

1. Software read GPIO offsets 0x604, 0x608, and 0x60C and preserve byte1 to byte3 before changing these GPIO offset's byte0.
2. EEPROM or I^{2}C accesses do not have this issue.

Impact: If software does not want to change byte1 to byte3 of GPIO offsets 0x604, 0x608, and 0x60C, it needs to read GPIO offsets 0x604, 0x608, and 0x60C and preserve byte1 to byte3 before changing these GPIO offset byte0.

2.11 PEX87xx NT Virtual Endpoint offset 0xC3C, 0xC40, 0xC44, and 0xC48 Read Does Not Return the Right Data

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx NT virtual endpoint offset 0xC3C, 0xC40, 0xC44, and 0xC48 programming works fine, but read to this offset returns incorrect data.

Solution/Workaround: None

Impact: Incorrect read data while reading NT virtual endpoint offset 0xC3C, 0xC40, 0xC44, and 0xC48

2.12 EEPROM/I^{2}C Needs to Write PEX87xx NT Virtual Endpoint Offset 0x6C Twice

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: If the application needs to change the default "Maximum Payload Size Supported" field of the PEX87xx NT virtual endpoint, it needs to write NT virtual endpoint offset 0x6C twice using EEPROM/I^{2}C. This field is Read Only to software.

Solution/Workaround: EEPROM/I^{2}C needs to write "Maximum Payload Size Supported" field with two writes

Impact: EEPROM/I^{2}C cannot change "Maximum Payload Size Supported" field with single write

2.13 PEX 87xx ASPM L1 Exit Logs Receive Error, Bad TLP, and Bad DLLP Errors at Gen3 Data Rate

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX87xx ASPM/PCI-PM L1 exit logs Receiver Error, Bad TLP, and Bad DLLP errors at Gen3 data rate.

Solution/Workaround: Program the following registers using EEPROM, I2C, or software to fix the bit errors. These settings reduce the overall ASPM/PCI-PM L1 power savings by 5 percent.

1. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x008E2120 - L0/L1/L2/L3
2. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x00AE2120 - L4/L5/L6/L7
3. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x00CE2120 - L8/L9/L10/L11
4. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x00EE2120 - L12/L13/L14/L15
5. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x008E7008 - L0/L1/L2/L3
6. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x00AE7008 - L4/L5/L6/L7
7. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x00CE7008 - L8/L9/L10/L11
8. Port0/Port4/Port8/Port12/Port16/Port20 offset 0xBFC with 0x00EE7008 - L12/L13/L14/L15

2.14 Some Non-Compliant PCIE Gen3 Add in Cards Cannot Handle "use_preset" Bit Set as a Response for Equalization Preset Request(s)

Risk Category: Low

Silicon Revisions Affected: AA

Description: PEX87XX-AA always sets the "use_preset" bit as a response to preset requests. But add in card receiver should ignore "use_preset" bit value in response to preset requests as required by the PCIe Gen3 Specification.

Solution/Workaround: Bypass the equalization by programming Port0/Port4/Port8/Port12/Port16/Port20 offsets 0xBD8 Bit[4:0].

- Bit0 to 1'b1 to bypass Port0 equalization
- Bit1 to 1'b1 to bypass Port1 equalization
- Bit2 to 1'b1 to bypass Port2 equalization
- Bit3 to 1'b1 to bypass Port3 equalization
- Bit4 to 1'b1 to bypass Port4 equalization

Impact: PEX87xx-AA does not have a mechanism to respond with "use_preset" bit clear for equalization preset request(s).

2.15 In Virtual Switch Mode, "VS GPIO_PG Availability" Register (offset A3Ch in VS Upstream Ports) Could Show One More GPIO Than What Is Assigned to that Virtual Switch

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: In Virtual Switch mode, "VS GPIO_PG Availability" Register is supposed to show the total number of GPIOs assigned to that Virtual Switch. But PEX87xx Virtual Switch's upstream port could intermittently show one more GPIO than what is assigned to that Virtual Switch hierarchy. There is no impact to the PEX87xx Base Mode.

Solution/Workaround: Each host checks it's "VS GPIO_PG Input Data" Register bits based on the pre-assigned GPIOs to its virtual hierarchy

Impact: Software should check only the assigned bits in "VS GPIO_PG Input Data" Register

2.16 PEX87xx Performance Monitoring Limitations

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: If PEX8718 and PEX8714 configured for 5-port mode, the performance monitoring has the following limitation.

1. Port4 DLL Egress performance counters cannot be monitored, and Port0 to Port3 DLL Egress performance counter(s) storage in debug RAM is shifted by an Entry.
2. If Port4 TIC (transaction layer ingress control block) receives TLPs, it inadvertently updates the Port0 TIC performance counters. This corrupts the Port0 TIC performance counter values.
3. Port4 TIC performance counters cannot be monitored.

Performance monitor does not work for Non-Transparent(NT) port.

TEC posted header performance counter could be off by one DW (double word) per multicast TLP.

Solution/Workaround: None

Impact: There is no impact to mission mode traffic (debug mode only impact).

2.17 PEX8734 Lanes 16 to 31 Are Reversed

Risk Category: Low

Silicon Revisions Affected: AA, AB

Description: PEX8734 lanes 16 to 31 are reversed as described in the following table.

Documented Lane Ordering	Actual Lane Ordering
Lane16	Lane31
Lane17	Lane30
Lane18	Lane29
Lane19	Lane28
Lane20	Lane27
Lane21	Lane26
Lane22	Lane25
Lane23	Lane24
Lane24	Lane23
Lane25	Lane22

Documented Lane Ordering	Actual Lane Ordering
Lane26	Lane21
Lane27	Lane20
Lane28	Lane19
Lane29	Lane18
Lane30	Lane17
Lane31	Lane16

Solution/Workaround:

- No programmable option to fix this reversal in PEX8734.
- Board design could reverse the lanes from "Actual Lane Ordering" column specified lane ordering from the preceding table.

Impact: PEX8734 lane assignments as shown in the above table are reversed. For example, when PEX8734 Port4 to Port7 is configured as x4x4x4x4, Port 4 gets lanes 31:28, Port 5 gets lanes 27:24, Port 6 gets lanes 23:20 and Port 7 gets lanes 19:16 (in reverse order).

2.18 PEX87xx Does Not Transmit EIOS (Electrical Idle Order Set) after Hot-Reset

Risk Category: None

Silicon Revisions Affected: AA, AB

Description: According to PCIe Base SPEC, EIOS transmission after hot-reset before entering electrical idle state is optional. PEX87xx does not implement this optional functionality

Solution/Workaround: None

Impact: None

2.19 PEX87xx PRBS Function Does Not Work

Description: PEX87xx PRBS checker expects the preamble detection followed by PRBS pattern check. PEX87xx PRBS checker never detects the preamble pattern due to a design bug.

Workaround: None

Impact: PEX87xx PRBS function does not work.

Appendix A: Reference Information

A.1 Affected Products and Silicon Revisions

This document details errata for the following products and silicon revisions.

Product	Description	Revision	Status
PEX 8714	12-Lane, 5-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012
PEX 8718	16-Lane, 5-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012
PEX 8734	32-Lane, 8-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012
PEX 8750	48-Lane, 12-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012
PEX 8764	64-Lane, 16-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012
PEX 8780	80-Lane, 20-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012
PEX 8796	96-Lane, 24-Port PCIe Gen 3 Switch	AA	Samples in Dec 2012

A.2 Device Documentation Version

The following documentation is the baseline functional description of the silicon.

Document	Version	Description	Publication Date
PEX 8714 Data Book	1.0	Data Book	June 2013
PEX 8718 Data Book	1.0	Data Book	June 2013
PEX 8734 Data Book	1.0	Data Book	June 2013
PEX 8750 Data Book	1.0	Data Book	June 2013
PEX 8764 Data Book	1.0	Data Book	June 2013
PEX 8780 Data Book	1.0	Data Book	June 2013
PEX 8796 Data Book	1.0	Data Book	June 2013

Revision History

Version 1.14, September 25, 2015

- Added Section 1.29.

Version 1.13, June 2015

- Added Section 2.19.

Version 1.12, August 25, 2014

- Added Section 1.28.

Version 1.11, August 12, 2014

- Added Section 1.27.
- Section 1.19 and Section 1.22 have been consolidated into Section 1.26
- Modified Section 1.25.

Version 1.10, August 4, 2014

- Added Section 1.25.

Version 1.09, July 28, 2014

- Replaced Section 2.16 with completely new write-up.

Version 1.08, June 13, 2014

- Added the following errata:
  - Section 1.21
  - Section 1.22
  - Section 1.23
  - Section 1.24

Version 1.07, January 29, 2014

- Updated Section 1.11, and changed Risk Category (from Low) to Medium
- Added Section 1.19
- Added Section 1.20
- Added Section 2.18.

Version 1.06, October 15, 2013

- Made minor changes to Section 2.17.

Version 1.05, October 14, 2013

- Added Section 2.17.

Version 1.04, August 14, 2013

- Added Section 2.16.
- Added Section 1.18.

Version 1.03, July 11, 2013

- Section 2.11, changed NT0 to NT
- Section 2.14, changed to indicate that only AA revision is impacted (i.e. caution 14 is fixed in AB revision).

Version 1.02, June 24, 2013

- Changed data book version to 1.0 and publication date to June 2013 in section B above,
- Added Section 1.17.

Version 1.01, June 14, 2013

- Added Section 1.16.

Version 1.00, June 10, 2013

- Added Section 1.15.

- Version 0.99, May 16, 2013
- Added Section 1.14.
- Added Section 2.15.

Version 0.98, May 6, 2013

- Added Section 2.9 through Section 2.12.
- Removed ASPM L1 as Section 1.14, and added Section 2.13 for this topic.
- Added Section 2.14.

Version 0.97, March 26, 2013

- Added Section 1.14.

Version 0.96, February 6, 2013

- Added Section 1.13.

Version 0.95, January 24, 2013

- Added Section 2.8.
- Added Section 1.12.

Version 0.94, January 16, 2013

- Added Section 2.7.

Version 0.93, December 21, 2012

- Added Section 1.11

Version 0.92, December 19, 2012

- Removed reference to Section 2.1 in Section 1.4.
- Minor modifications in Section 1.10.

Version 0.91, Dece,mber 13, 2012

- Added Section 1.10.

Version 0.90, October 29, 2012

Initial release of the document.

Avago
TECHNOLOGIES