
PCI
SIG®

Common Pitfalls in PCI Express® Designs

Jitendra Puri (JP)

Engineering Director

Agenda

- Introduction
- Migration from PCI/PCI-X® to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Introduction

- Tell any design engineer that he/she is coding bugs along with the design !!
  - ✓ Every one is going to get offended !
- But, that's the reality.
- Common pitfalls based upon bugs found during Verification of several PCI Express designs
- Suggested Defect Prevention steps

Agenda

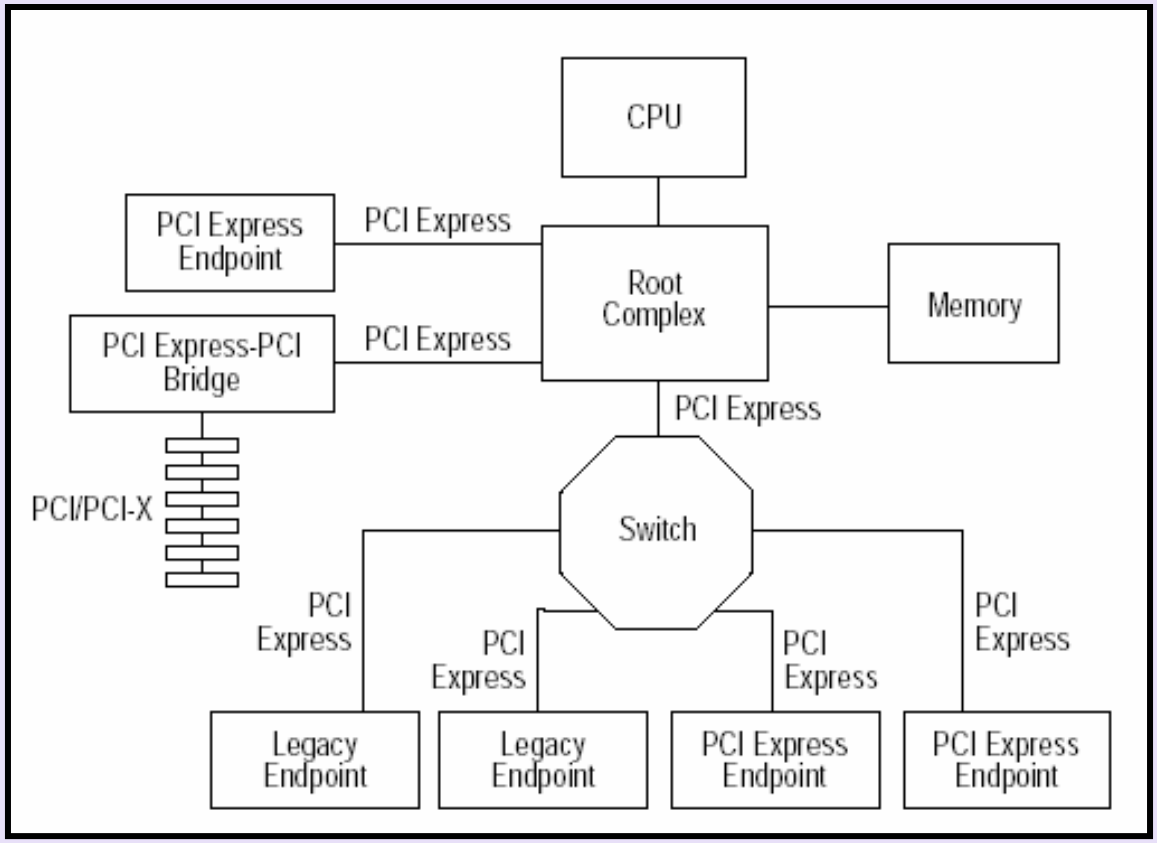
- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Migration From PCI/PCI-X to PCI Express…

- PCI limited to 64-bit, 66-MHz
- PCI-X mode-1 limited to 64-bit,133-MHz
- PCI-X mode-2 surely higher speeds but
  - ✓ Complex IO PAD structure
  - ✓ High Bus Switching speed
- PCI Express is offering high bandwidth per pin
- Scalable performance via aggregated Lanes

Migration From PCI/PCI-X to PCI Express…

- PCI Express is an effort to move everything to a common IO standard



Migration From PCI/PCI-X to PCI Express

- An obvious choice for all the designers who worked on PCI and PCI-X

Agenda

- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Error Prone Areas in PCI Express Design…

- Experienced PCI Designers know what is ..
  - ✓ Configuration Space
  - ✓ Configuration cycles, Memory cycles
  - ✓ Device number, Bus number, Base Addresses
  - ✓ TAG, Split cycles etc.

- Majority of the designers are/were very comfortable with the Transaction Layer concepts/terminology

Error Prone Areas in PCI Express Design…

- More error prone areas that need special attention are the “newer” concepts
- Based on the analysis of the bugs unearthed during the Verification of several designs, the Error prone areas are:
  - ✓ Physical Layer
    - LTSSM design
      - Incorrect LTSSM state transitions
  - ✓ Data Link Layer
    - Flow Control
      - Update FCs not being sent for VCx

Error Prone Areas in PCI Express Design…

- Credit considerations for Message Packets

– DL_Inactive status consideration
  - Data Link Layer not getting reset on entry to DL_Inactive

– Replay mechanism
  - Incorrect re-transmission from Retry buffer

✓ Transaction Layer

– AER
  - Incorrect setting of configuration register bits

– RCB Boundary handling
  - Incorrect interpretation of RCB parameter

– Incorrect usage of Max_Read_Request_Size register in the receive logic

Error Prone Areas in PCI Express Design

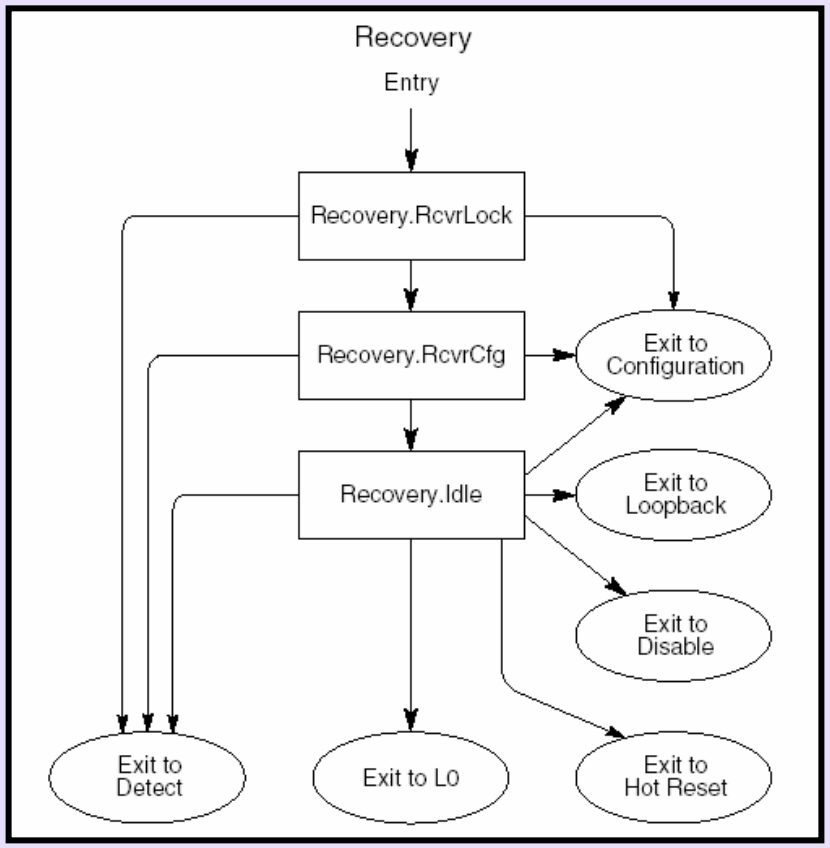
- ✓ Power Management
  - ASPM L0s entry
    - Tx_L0s and Rx_L0s not kept independent
  - ASPM L0s exit
    - Incorrect number of N_FTS sent

Agenda

- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Bug Details…

- **LTSSM design:** Incorrect LTSSM State Transitions
  - Transition Control bits viz. *Hot Reset, Loopback, Disable Link, Disable Scrambling* coming as part of TS OS are ignored
    - DUT makes transition to L0 instead of the other desired state
    - LTSSM deadlock
    - Unnecessary Timeouts



Bug Details…

- ✓ **Flow Control:** Update FCs not being sent for VCx
  - Update FCs for VC0 are being normally sent without fail as the same is initialized by default
  - VCx is enabled during the run time
  - Update FCs for VCx are not getting scheduled properly
    - Starvation of credits for VCx blocking the traffic flow

Bug Details…

- **Flow Control:** Credit considerations for Message Packets
  - Credit information not updated for Message Packets
  - Certain Message packets not supported by the DUT are simply ignored
    - SSPL
    - Vendor Defined Messages
  - Not taken into consideration while sending the Update FC information
    - Results in Credits not getting freed up and hence starvation

Bug Details…

- ✓ **DL_Inactive status consideration:** Data Link Layer not getting reset on entry to DL_Inactive
  - NEXT_TRANSMIT_SEQ, ACKD_SEQ, REPLAY_NUM, NEXT_RCV_SEQ, etc are not set to default values
  - Retry buffer not freed up on re-establishing link
  - The TL packets received from the application layer are remembered and are sent out even when link layer is in DL_Down state (FC_INIT1)
    - The first packet won't go with sequence number = 0
    - The contents of retry buffer would get transmitted
    - Credit logic totally out of sync

Bug Details…

- **Replay mechanism**: Incorrect re-transmission from Retry buffer
  - Does not block acceptance of new packets from the Transaction layer
    - This packet appears in the middle of the replay
  - Incorrect handling of ACK/NAK during the course of replay
    - Ends up sending the ACKed packets again

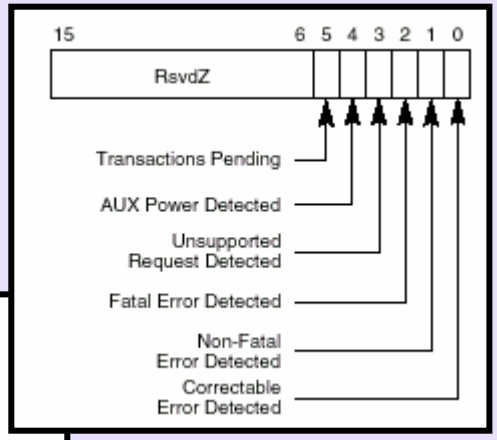
Bug Details…

- ✓ **AER**: Incorrect setting of configuration register bits
  - When AER is implemented, the bits in the AER registers are updated right, but certain register to be updated in the standard PCI configuration space are left out

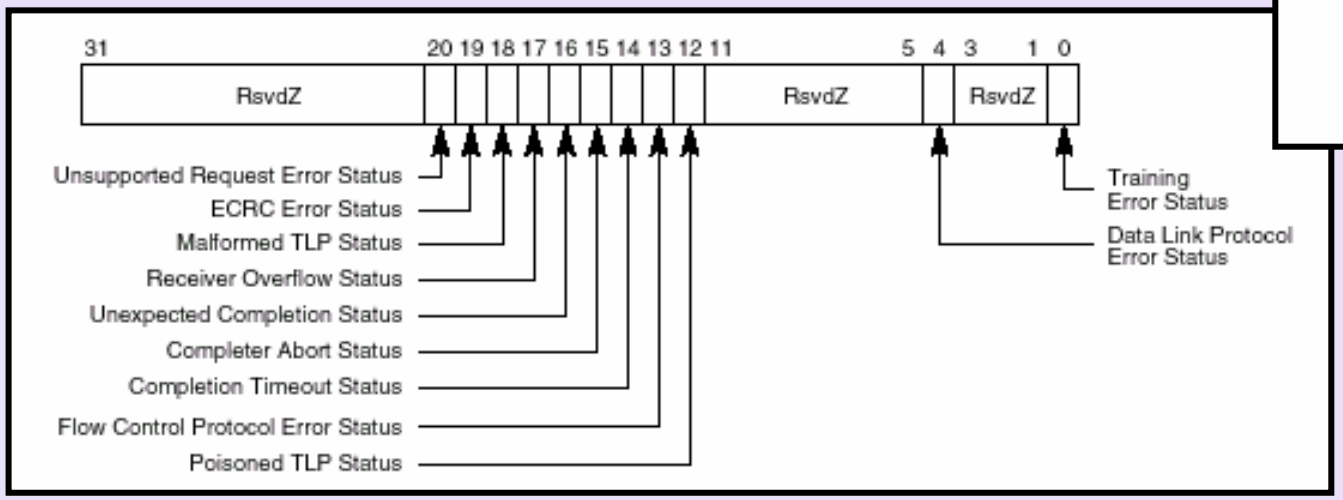
Signaled System Error – See Section 7.5.1.7.

This bit is set when a device sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR Enable bit in the Command register is 1.

Default value of this field is 0.



Device Status Register



Uncorrectable Error Status Register

Bug Details…

- ✓ **RCB Boundary handling**: Incorrect interpretation of RCB parameter
  - EP designs assume that they can also send 64-byte aligned completions when RCB bit set to 0
    - Incorrectly send out completions broken to 64-byte boundary
    - Malformed for the receiver

Endpoints:

Read Completion Boundary (RCB) – May be set by configuration software to indicate the RCB value of the Root Port upstream from the Endpoint. Refer to Section 2.3.1.1 for the definition of the parameter RCB.

Defined encodings are:

0b	64 byte
1b	128 byte

Receivers may optionally check for violations of RCB. If a Receiver implementing this check determines that a Completion violates this rule, it must handle the Completion as a Malformed TLP

Bug Details…

- ✓ Incorrect usage of Max_Read_Request_Size register in the receive logic
  - This is wrongly interpreted and its value is used in receive logic also
    - The received read requests' size gets wrongly compared with this
    - Well formed read request gets treated as Malformed read request

Max_Read_Request_Size – This field sets the maximum Read Request size for the Device as a Requester. The Device must not generate read requests with size exceeding the set value. Defined encodings for this field are:

Bug Details

- ✓ **ASPM L0s entry :** Tx_L0s and Rx_L0s not kept independent
  - The TX is forced to go to L0s when the RX is in L0s
    - Blocking the packet transmission

> It is possible for the Transmit side of one component on a Link to be in L0s while the Transmit side of the other component on the Link is in L0.

Bug Details

- ✓ **ASPM L0s exit:** Incorrect number of N_FTS sent
  - The transmitter sends out N_FTS equal to the number it itself advertised and not based on the number it received
    - The device does not get the required time for transition to L0 from L0s
    - Might go to recovery after the N_FTS timeout

Agenda

- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

The Most Occurring Bugs…

- And the WINNER is…..
  - ✓ The received SSPL message not taken into account while sending update FCs.
    - Majority of the EP design specs say that the “data received as part of SSPL would be ignored”
    - Somehow the whole of packet gets ignored including the Header and the Data Credits consumed
    - The UpdateFC going back does not take into account the above mentioned Credits
    - The Other device finds lesser available Credits with the Target device

The Most Occurring Bugs…

- Impact of the “SSPL” bug on the system…
  - ✓ Caused the TX of the other device to believe that lesser Header/Data credits are available
  - ✓ Direct impact on performance
  - ✓ In one of the designs, the system was to have 64-byte and 128-byte Writes. But with the Credit not being freed in terms of Update for the SSPL, system would see only 64-byte Writes, and never 128-byte Writes…. Performance Degradation….

The Most Occurring Bugs…

- And the next in line is…..
  - ✓ Various timers not set properly if link is established with lane count < maximum supported
    - Assuming maximum supported lane count to be 8, but link is established at 4, various timers are still kept as per 8 lane configuration
      - AckNak_LATENCY_TIMER
      - REPLAY_TIMER

The Most Occurring Bugs…

- Impact of “Timer” bug on the system…

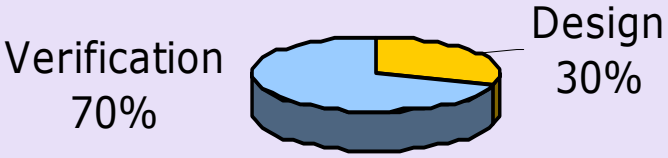
  - ✓ AckNak_LATENCY_TIMER
    - The ACK/NAK is still all right, it is scheduled faster
    - This may cause the TX to come out of L0s faster than normal

Agenda

- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Defect Prevention…

- The real Mantra is “Defect Prevention”
- Aim is to prevent injection of these bugs at the first place itself
- Spend some more time in Design phase and drastically cut the Verification efforts
- Review error prone areas more thoroughly



Defect Prevention

- Use proven Verification IP
  - ✓ Having Compliance Test Suite
  - ✓ Extensive error injection and detection capabilities
  - ✓ Ability to create any real life test scenario
- Test suite based on PCI-SIG provided Compliance Checklists should be a key component of the verification effort

Agenda

- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Compliance Checklist (design phase)

- Compliance checklist to be discussed as part of Design Kick-Off Meetings
- Not very easy always to derive finer points from a 400+ page specifications
- Compliance checklist would be handy here

TXN.02.12#02 **All Message Requests use the Msg Type field encoding, except for the Vendor_Defined messages, which can use either Msg or MsgD, and the Set Slot Power Limit message which uses MsgD.**

✓ Handling it in Receive logic would have avoided the bug

Compliance Checklist (Tracking)

- Make Compliance Checklist as part of the “Traceability Matrix”

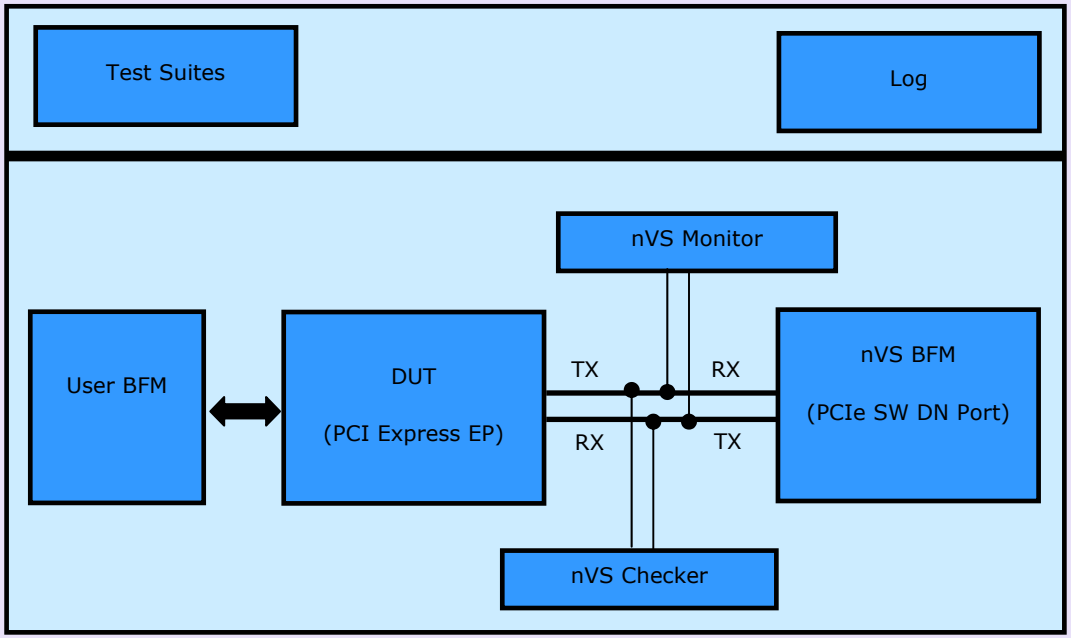
Check Point #	Description	Section in High Level Design	Section in Detailed Design	Code	Test Case #
TXN.02.00#02	All TLP fields marked Reserved in the Base Specification must be filled with all 0s when a TLP is formed.	2.1	TL-2.3	nvs_ex_ttl.v	TL-21
TXN.02.00#03	Values in TLP reserved fields must be ignored by receivers from a functional view, but are included in all LCRC and ECRC calculations.	2.3	TL-2.4	nvs_ex_ttl.v nvs_ex_tdl.v	TL-25 DL-10
. . .					

Compliance Checklist (Test suite)…

- Compliance Suite available with 3rd Party Verification IPs
- Directed Test Suite based on the Compliance Checklist
- Map each and every check-point to a test case
- Self-checking Test case
- Randomness in even in the directed tests to generate a newer test pattern in every run

Compliance Checklist (Test suite)…

- Typical Verification environment will have
  - ✓ Bus Functional Model (to generate and receive traffic)
  - ✓ Protocol Checker
  - ✓ Bus Monitor (to show the formatted log)
  - ✓ Test Suites (including Compliance)



Compliance Checklist (Test suite)

```
//------------------------------------------------------------------
// TXN.2.2#1
// All required and implemented optional checks for malformed TLPs must be detected and reported
// the receiver as errors associated with the receiving port.
//------------------------------------------------------------------
`CHK0.RX.do_cfg(`EX_TLP_MSG_EFT,1'b0);
`CHK0.TX.do_cfg(`EX_TLP_MAL_LENGTH_NE_PAYLOAD,1'b0);

`p0.do_err(`EX_TLP_MAL_LENGTH_NE_PAYLOAD,`EX_LEN_PAYLOAD_DIFF);

get_random(range_bfm1_mem32,ex_max_data_sz_1,random_address,random_bc);
`p0.do_cmd(`EX_MWR,(`EX_UP_PORT_MEM32_BADDR+random_address),random_bc);

wait_for_err_msg_ft(`EX_MAX_DELAY,status);

if(status)
        begin
                print("log","Case 15(a) -> EP has not sent Fatal Error Msg");
                ->event_tp_compliance_error;
        end
@(nvs_ex_tests.bfm_idle);
```

Agenda

- Introduction
- Migration from PCI/PCI-X to PCI Express
- Error Prone Areas in PCI Express Design
- Bug Details
- The Most Occurring Bugs
- Defect Prevention
- Compliance Checklist
- Summary

Summary

- PCI Express has been a natural upgrade for PCI/PCI-X designers
- “To Err is Human”… the bugs are going to be there !
- Review error prone areas more thoroughly
- Use back-to-back operation of Verification IP to improve understanding
- Use Verification IP with proven ability to detect bugs
- Compliance test suite to be one of the key components of the verification effort

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

For more information please go to www.pcisig.com

Common Pitfalls in PCI Express Designs

Jitendra Puri (JP)

Engineering Director


PCI
SIG®