Debugpci: Making PCIe Common Error Debugging Easier

Shradha Todi and Padmanabhan Rajanbabu
Introduction
Debug features in DesignWare PCIe controller
Need for a streamlined PCIe debugging procedure
pciutils – Open-Source Tool for diagnosis
dbgpci – command line based diagnostic tool
Architecture and software stack of dbgpci
Software implementation of dbgpci
Real Use-Cases
Conclusion and Future scope
Introduction

- PCIe - most prevalent interface standard for connecting high-speed components
- Critical applications utilizing PCIe shall be bulletproof and validated thoroughly for all conditions
- Debugging PCIe IP is cumbersome in the absence of IP Internal debug registers for status check
- DesignWare PCIe IP has a versatile debug and RAS-DES register set to ease debugging effort
- This presentation proposes a simple command line based diagnostic tool compatible with any Linux system
- This tool utilizes debug registers to simplify PCIe error solving methodology
- The tool also follow a streamlined debugging process, improving the debugging efficiency
Vital debug hooks provided by the controller

- LTSSM state and equalization status
- Error Injection - ECRC/LCRC/Unsupported Request
- Error detection, error logging and error handling mechanisms
- Time and event-based counters to measure the % of time the controller spends in each low power state
- TX/RX data throughput in the system
- Statistical event counters
Mainstream debugging procedure involves capturing trace using PCIe analyzers.

But why does it need an alternative?

Source: https://www.storagenewsletter.com/wp-content/uploads/2018/06/TELEDYNE_LECROY_1806_summit_m5x_2.jpg
Need for Streamlined PCIe Debugging Procedure

- Need to dump software debug registers to obtain information
- Decoding the dump in the absence of any user space tool is tedious

Source: http://trace32.com/wiki/index.php/Accessing_System_Data
pciutils – Open Source Tool for Diagnosis

- Collection of utilities to access PCI bus configuration registers
- Runs on Linux, Windows and many more platforms
- Easy to add support for platform/board specific PCIe configuration registers and commands
- The library has the following utilities:
  - `lspci`: displays information about all PCI buses and devices.
  - `setpci`: allows to read from and write to PCI device configuration registers
  - `update-pciids`: download the current version of the pci.ids file.
- Source -
  - [https://git.kernel.org/pub/scm/utils/pciutils/pciutils.git](https://git.kernel.org/pub/scm/utils/pciutils/pciutils.git)
debugpci – Command Line Based Diagnostic Tool

- **debugpci** - command line based diagnostic tool to help users dump all the required debug data

- The commands that will be used for capturing and dumping data:
  - Begin Capture - `debugpci [[[<domain>]:]<bus>]:[<slot>][.[<func>]] c`
  - Analyse capture - `debugpci [[[<domain>]:]<bus>]:[<slot>][.[<func>]] d`

- Internally captures counters for different errors, LTSSM states and silicon debug registers provided by DesignWare PCIe IP

- Presents the captured debug data in a human readable format

- Analyses the discrepancy (If any) seen in the collected data and presents the same to user with all possible root causes

- For example, if the tool checks that there was a receiver detection related timeout when reading SD_CONTROL2_REG. It will alert the user and provide input by printing following message:

  - “A Receiver detection timeout was seen. If the PHY requires more time for receiver detection, the application software can hold LTSSM in Detect.Active by setting the HOLD_LTSSM field of SD_CONTROL2_REG[0] register.”
The PCI library is a portable library for accessing PCI devices and their configuration space present in the user space layer.

Linux PCI subsystem enumerates and populates the PCI devices. The vendors have underlying low level device drivers in the kernel layer.

PCI PHY is the physical layer in the PCI architecture stack.
The main function gets the pci_access structure, initializes the PCI library and gets the device that is requested by user.

Depending on whether the user has selected option “c” or “d”, it calls the capture or dump function respectively.
The capture function enables all debug and error counters present in DWC controller for all the lanes.

Some error counters added in a static structure is shown in the figure above.
The dump function dumps all debug and error counters present in DWC controller for all the lanes by reading respective error/debug register.

Function root_cause_issue compares the read value with expected value and in case of mismatch, prints the root-cause analysis for various observed issues.
Receiver Detection - Receiver detection is used to determine if a remote PCIe link partner is available to establish a linkup

Broken Lanes - After receiver detection, if some lanes are broken, the link may not reach L0 at the desired link width

Speed Change - Sometimes link does not come up / unstable at the new speed or link falls back to a lower speed after speed change

Link Equalization - Sometimes link fails to come up at Gen3 / Gen4 data rate or is unstable after speed change to Gen3 / Gen4 data rate
- **Receiver Error** - PCIe link does not remain stable in L0 and goes down to Recovery

- **Correctable Errors** - Correctable errors are the most common consequence of poor link quality and can lead to link instability.

- **Uncorrectable Errors** - Uncorrectable errors are further classified as Fatal Errors and Non-Fatal Errors based on the Severity

- **ASPM issues** - Sometimes link is unable to properly enter low power states like L0S, L1 and it’s substates
- Tool prints all possible debug and error register values for all possible lanes.
- Against each error, it prints the number of times the error occurred.
- Against debug values like RX valid, or lane detected value 1 indicates true and value 0 indicates false.
These are errors that render the link and related hardware unreliable. These are only seen when any of the link components is severely broken, and forces link to go to detect state.

- As we can see, Protocol error status is flagged in the tool.
- The root cause is given in the last line stating the reason why protocol error can occur.

Flow Control Protocol Error: Occurs if no DLLP is received within a 200us window. This indicates that the link quality is severely deteriorated.
Real Use-Case: Framing Error

- Framing Token
  - When EDS token was expected but not received or whenever an EDS token was received but not expected.
  - When a framing error was detected in the deskew block while a packet has been in progress in token_finder.

- Unexpected STP Token
  - When Framing CRC in STP token did not match
  - When Framing Parity in STP token did not match.

- Unexpected Block
  - When RxStatus Error was detected in Datastream state
  - When Not full 16 EIEOS symbols are received in EIEOS state
After receiver detection is completed, the LTSSM goes through the following states: Polling → Configuration → Recovery before reaching L0 state at Gen1 data rate.

- If some lanes are broken after receiver detection, the link may not reach L0 at the desired link width.
- The RX valid not being set indicates there might be broken lanes as mentioned in the last line.
We were able to use this diagnostic tool even in pre-silicon emulation with ZeBu PCIe transactor for PCIe link-partner and root-caused several issues being faced during pre-silicon validation.

This tool helps in avoiding any human error while manipulating the debug data.

The users of DesignWare PCIe IP can simply use this tool for self diagnosis and share the report to Synopsys or customer for further analysis.

For every issue detected, a possible error condition is always highlighted.

Future scope involves upstreaming the source code as part of GPL license in Linux or as open source tool.
Any Questions?
THANK YOU