ISSN:0975-9646

Link Initialization and Training in MAC Layer of PCIe 3.0

Chandana K N , Karunavathi R K

Department of E&CE, Bangalore Institute of Technology
Bangalore, Karnataka, India

***Abstract*— The serial protocols like PCI Express and USB have evolved over the years to provide very high operating speeds and throughput. This evolution has resulted in their physical layer protocol becoming very complex. One of the most essential processes at physical layer is link initialization and training process. In the PCI Express devices, this process establishes many important tasks such as link width negotiation, link data rate negotiation, bit lock per lane, symbol lock/block alignment per lane, etc. All these functions are accomplished by Link Training & Status State Machine (LTSSM), which observes the stimulus from remote link partner as well as the current state of the link, and responds accordingly.**

***Keywords*—Encoding, Link Training and Status State Machine, Ordered Sets, PCIe 3.0, Scrambling, Verification Introduction**

I. INTRODUCTION

The PCIe 3.0 architecture utilizes very efficient and productive algorithms for maintaining reliable link, highly optimized power consumption and extremely fast and flawless data transfer rate. The Link Training Status State Machine has been employed as the foremost workhorse in these regards. Its functions and provisions contribute matchlessly towards the super speed high class performance. The LTSSM tunes and trains the PCIe link for reliable data transfer. It also implements various algorithms for link's reliability maintenance and is also responsible to recover the link from any errors as may arise. It also plays key role in power management by greatly reducing link's power consumption and nullifying any conditions that waste power. The LTSSM also performs operations for making the link ready for data transaction in the very beginning when the device is plugged in. Hence LTSSM is the Data Flow GatewayControl for the device. The work also includes the development and verification of MAC Layer of PCIe 3.0 device. The LTSSM communicate and co-ordinates with almost all the layers of the device namely the PHY, the MAC, the link layer and also the master controller.

We use a PCI Express LTSSM whitebox reference model, which is a part of the bigger UVM-based testbench environment. The LTSSM reference model observes the same physical layer traffic as the DUT, behaves as per the PCI Express Base Specification and also predicts the possible state transitions. As opposed to the Black Box tetsbench which has no idea about the state of DUT's internal blocks, this model is aware of DUT's LTSSM state and values of useful LTSSM parameters.

The PCI Express defines the state behaviour and relevant state transitions so that there can be multiple state transitions conditions to transition to the same next state. For some of the sub-states, there are multiple state transition paths that lead to different next states. To trigger all the required state transitions and transition conditions, we use a mixture of directed and constrained random stimulus generation. As each and every statement in the PCIe Base Specification description of LTSSM requires attention, we create a detailed coverage for all sub-states that includes all state transition paths, transition reasons/conditions, transmit rules, stimulus etc.

II. LITERATURE REVIEW

All In the world of communication protocols, PCI-Express presents throughput in 2.5 GT/s, 5.0 GT/s and 8.0 GT/s. It is important to not forget the purpose of each protocol. PCIe is a high-speed serial computer expansion bus standard designed to replace the older PCI, PCI-X, and AGP bus standards. PCIe has numerous improvements over the older standards, including higher maximum system bus throughput, lower I/O pin count and smaller physical footprint, better performance scaling for bus devices, a more detailed error detection and reporting mechanism and native hot-plug functionality. More recent revisions of the PCIe standard provide hardware support for I/O virtualization [1],[2],[3],[4].

As the SuperSpeed USB 3.0 protocols are intended for dual simplex transmission lines, for the sake of parallel transactions, there is an absolute need of having the architecture which supports such protocols. [5] [6] have developed a fully synthezied LTSSM (Link Layer and Transition State Machine) and also interfaced it with previously developed MAC layer. The layered architecture of USB 3.0 communication protocols itself turned out helpful in structuring verification effort to enhance it.

[7] describes a method to implement the data link layer of the PCIe 3.0. The data link layer is involved in the exchange of packets at the DLL level with a state machine for flow control and initialization.

A novel Multi-mode Serial Link Controller (MMSLC) for logic physical layer (PHY) and data link layer (DLL) of USB 3.0, PCle 2.0 and SATA 3.0 is introduced in [8]. This approach exploits the relationship between protocols' similarity with circuit flexibility and its real-time requirements with effective circuit area usage, as verified in this paper. Our results show that this architecture is capable of achieving the high-speed requirements of around 500MHz symbol rate for serial link protocols and realize area reduction over conventional link controllers running each protocol individually.

With the evolution of modern verification methodologies, system-level verification using constrained-random stimulus is a high priority, especially in very large communication applications. A key goal to address is providing fast, effective test coverage. In order to generate stimulus automatically to cover all coverage bins more quickly in the verification process, especially in very large communication applications, a novel method which combines the benefits of GA and coverage-driven verification methodology is proposed. By analyzing the real time coverage results from the simulation and thereafter intelligently modifying the corresponding stimulus, this novel method iteratively improves coverage. As a result, the GA can more effectively generate stimulus. The experimental results from both a C-based testbench and a real application (PCIe system) prove that the proposed GA method can streamline the verification effort and sharply reduce simulation time to achieve thorough coverage [10].

The SystemVerilog Language Reference Manual and Universal Verification Methodology User's Guide has been referred wherever required for language constructs [11][12].

III. Link Initialization and Training

The architecture of PCIe is classified in terms of three discrete logical layers: the Transaction Layer, the Data Link Layer, and the Physical Layer. Link Initialization and Training is a Physical Layer control process that configures and initializes a device's Physical Layer, port and associated Link so that normal packet traffic can proceed on the Link. This process is automatically initialized after reset without any software involvement. A sub-set of Link re-training is initiated automatically as a result of wake up event from a low power mode or due to an error condition that render the Link inoperable. The location of Link Training and Status State Machine (LTSSM) is the Physical Layer sub-block responsible for the Link Initialization and Training process as shown in Fig 3-1.

The top level states of the LTSSM are shown in Fig 3-2. Each state consists if sub-states, that taken together, comprise that state. The first LTSSM state that is entered after exiting fundamental reset or Hot reset is the Detect State. The LTSSM consists of 11 top level states as briefed below:

1. **Detect** The purpose of this state is to detect when a far end termination is present. This state can be entered at any time if directed.
2. **Polling** The Port transmits training Ordered Sets and responds to the received training Ordered Sets. In this state, bit lock and Symbol lock are established and Lane polarity is configured.
3. **Configuration** In this state, both the Transmitter and Receiver are sending and receiving data at the negotiated data rate. The Lanes of a Port configure into a Link through a width and Lane negotiation sequence.
4. **Recovery** In Recovery, both the Transmitter and Receiver are sending and receiving data using the configured Link and Lane number as well as the previously supported data rate(s).Recovery allows a configured Link to change the data rate of operation if desired, re-establish bit lock, Symbol lock or Block alignment, and Lane-to-Lane de-skew.

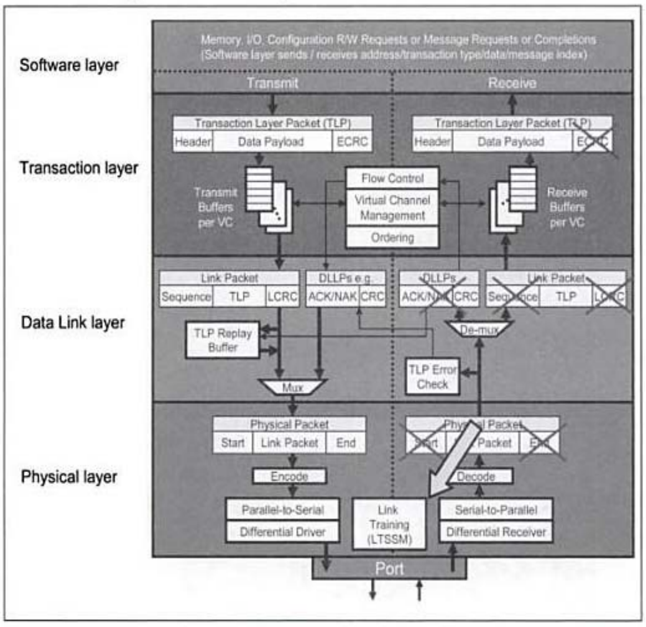

Fig 1 LTSSM Location

5. **L0** L0 is the normal operational state where data and control packets can be transmitted and received. All power management states are entered from this state.
6. **L0s** L0s is intended as a power savings state. L0s allows a Link to quickly enter and recover from a power conservation state without going through Recovery.
7. **L1** L1 is intended as a power savings state. The L1 state allows an additional power savings over L0s at the cost of additional resume latency.
8. **L2** Power can be aggressively conserved in L2. Most of the Transmitter and Receiver may be shutOff. Main power and clocks are not guaranteed, but Aux power is available.
9. **Disabled** The intent of the Disabled state is to allow a configured Link to be disabled until directed or Electrical Idle is exited (i.e., due to a hot removal and insertion) after entering Disabled.
10. **Loopback** Loopback is intended for test and fault isolation use. Only the entry and exit behavior is specified, all other details are implementation specific. Loopback can operate on either a per Lane or configured Link basis.
11. **Hot Reset** A Link can enter Hot Reset if directed by a higher Layer. A Link can also reach the Hot Reset state by receiving two consecutive TS1 Ordered Sets with the Hot Reset bit asserted.

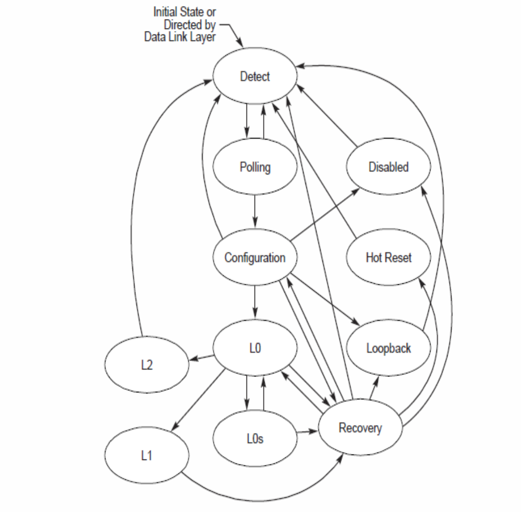

Fig 2 Main State Diagram for LTSSM

IV. Verification Architecture

The LTSSM has been designed and verified using UVM methodology. The verification architecture is as shown in Fig 4-1

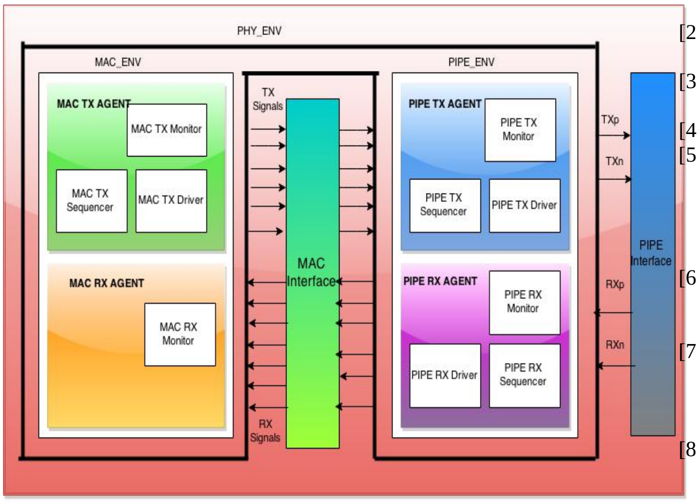

Fig 4-1 Verification Architecture

The MAC driver has the driving LTSSM which will keep track of the state machine transitions, whereas the MAC monitor monitors the arrival and sampling of the data packets and ordered sets from the upper layers at the transmit side of the MAC agent.

The receive side of the MAC agent samples the data packets and ordered sets from the PIPE interface and drives it to the upper layers. The data packets are scrambled and descrambled in the MAC agents.

The transmit side of the PIPE agent encodes and serializes the data packets on to the PIPE interface. The receive side of the PIPE agent decodes and deserializes the data packets from the PIPE interface.

V. Conclusion

The LTSSM has been designed and verified using SystemVerilog UVM Methodology. This LTSSM has been interfaced with the existing MAC Layer and proved effective use. Following the latest PCIe 3.0 specifications, the designed LTSSM can easily be hooked up with other layers. The functionality of LTSSM is also verified, via simulation, along with integrated MAClayer. The layered architecture of PCIe 3.0 communication protocols itself turned out helpful in structuring verification effort to enhance it. The layers can be verified separately with minimal overhead in the test development effort.

Acknowledgment

The authors wish to thank Mindtree Limited. This work was supported in part by a grant from Mindtree Limited.

References

[1] "*PCI Express ® Base Specification*", Revision 3.0, Version 1.0, November 10, 2010.

[2] "*PHY Interface for the PCI Express, SATA and USB 3.1 Architectures*", Version 4.0, Intel Corporation, 2011.

[3] Ravi Budruk, Don Anderson & Tom Sanely, 2004. "*PCI Express System Architecture*", Mindshare Inc., pp 419-434.

[4] "PCI-SIG Developer's Conference", PCI-SIG.

[5] Hasan Baig, Muhammad AsrarAlam, Jeong-A Lee, "*Integrated LTSSM (Link Training & Status State Machine) and MAC Layer of USB 3.0 Device for Reliable SuperSpeed Data Transactions*", Research Notes in Information Science (RNIS Volume9, Number1, May 2012.

[6] Rohith Kumar, Hardik Trivedi, NitishAlok, "*Design and Verification of USB 3.0 Link Layer (LTSSM)*",International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014, 4916-4921

[7] M. Aguilar, A. Veloz and M. Guzman, "*Proposal of implementation of the data link layer of PCI-express*", Proceedings of 1st International Conference on Electrical and Electronics Engineering, pp. 64, June 2004.

[8] Lei Wang ,Pawankumar Hegde, Vishal Nawathe, Roman Staszewski, Paras Balsara, Vajin Oklabdzija, "*Design of a Link-Controller architecture for Multiple Serial Link Protocols*", IEEE 2010, 266 -271

[9] Julien Saadé, Frédéric Pétrot, André Picco, Joel Huloux, Abdelaziz Goulahsen, "*A System-Level Overview and Comparison of Three High-Speed Serial Links: USB 3.0, PCI Express 2.0 and LLI 1.0*", IEEE 2013, 147 – 152

[10] Wang Jiawen, Liu Zhigui, Wang Suliang, Liu Yang, Li Yufei, Yang Hao, "*Coverage-Directed Stimulus Generation Using a Genetic Algorithm*", ISOCC 2013, IEEE Transactions, 298 – 301

[11] "*SystemVerilog 3.1a Language Reference Manual*", Accellara - Extensions to Verilog-2001, 2004

[12] "*Universal Verification Methodology (UVM)*", User's Guide - 1.1, Acellera, 2011