High Level Modeling Of Physical Layer Noise Parameters Using SystemC

Prem Kumar Lohani
Whizchip Labs
SOIS,
Manipal, India

Ranjani K
Whizchip Labs
SOIS,
Manipal, India

Ravi Shankar R
Whizchip Design Technologies
Bangalore, India

Sundaresan C
SOIS
Manipal University
Manipal, India

Chaitanya CVS
SOIS
Manipal University
Manipal, India

***Abstract*—SystemVerilog and SystemC are extensively used for design and Verification in VLSI industry. This paper propose a method to combine SystemVerilog and SystemC code in a single hardware/software simulation which allows design teams to leverage abstract representations of system function as it increases system simulations speed. Both languages interoperate through an intermediate layer of abstraction known as Transaction Level Models (TLMs). This paper develops Universal Verification Methodology (UVM) TLM environment for SV and SC communication in the system modeling.**

Keywords—SystemVerilog, SystemC, TLM;

I. INTRODUCTION

Communication protocol such as Ethernet [6], PCI express [8] or USB [7] provides exchange of information between devices. As per the OSI model, physical layer provides actual interconnection between devices. In physical layer due to thermal and environmental characteristics of the physical medium, noise is generated and distorts the information carried by the physical medium.

The physical medium noise constraints must be considered to provide the real time environment for communication protocol during simulation or verification. Consideration of noise or timing parameter during the simulation will result in more accurate verification of communication protocol.

In general, physical layer involves relationship between a device and a transmission medium, such as copper or fibre optical cable. The major functions and services performed by the physical layer are: Establishment & termination of a connection to a communications medium and Conversion between the representation of digital data in user equipment and the corresponding signals transmitted over a communications channel. These signals are operating over the physical cabling.

During the transmission, physical and electrical characteristics of devices and physical medium affect the information to be exchanged. These electrical and thermal characteristics generate noise parameters such as jitter, skew etc. Due to above mentioned parameters, timing violation occurs in the communication protocol and are responsible for data loss.

The above effect of noise parameters are observed in real time. To accommodate such real time scenario during the simulation for comprehensive studies creates requirement of generalized noise channel model

In order to allow for comprehensive studies on the physical layer noise parameters, high-level modelling of the noise effects can be used for the performance analysis in circuits and systems which are prone to errors. The modelling is based on a SystemC description because of its system level modelling capabilities and of its fine time granularity.

The rest of the paper is organized as follows. Section 2 discusses about the noise channel, noise parameters, noise channel models. Section 3 discusses about results and the last section discusses our conclusion.

II. NOISE CHANNEL MODEL

The top level block diagram for the noise model is shown in figure 1.

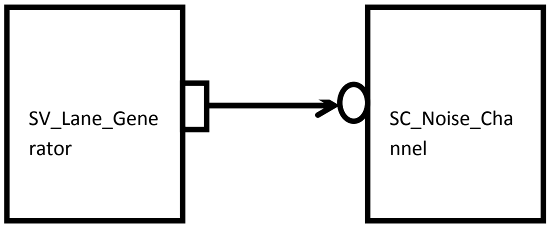

Figure 1: Top Module

There are two main components at the top level, which are developed in two different languages SystemVerilog (SV) [2] and SystemC (SC) [1] [5], and both are communicating using Transaction Level Modelling (TLM) [9]. These components are described as follows:

A. SV_Lane_Generator

This component generates lanes value as transaction and developed in UVM [3] [4]. Transactions are randomized in this component and sent out through TLM port.

B. SC_Noise_Channel

This component receives lane as transaction generated by SV_Lane_Generator. This component models noise channel in SystemC by computing noise parameter (such as jitter, skew) for each lane and update transaction with computed noise parameter. Component receives transaction through TLM export.

III. Noise Channel Implementation

The systems has modelled in such way that it can accommodate the real time noise parameters during the transmission problem such as transition time, jitter, lane skew, cable delay and return loss.

A. Transition Time:

Transition time (Rise time and fall time) is defined as time taken by the signal to change from 20% to 80% or 80% to 20% signal level respectively, of isolated edges.

B. Jitter:

Jitter may be caused by the electromagnetic interference (EMI) and the crosstalk with carriers of other signal lane [10]. The measured jitter at the transmit and receive compliance point shall be less than the maximum Total Jitter, maximum Deterministic Jitter, Random Jitter and duty tolerance Jitter. Accuracy of Jitter is specified in ppm. Jitter can vary between specified ranges. Calculations of jitter accuracy range in time unit is as follows:

$$\textbf{Jitter accuracy} = \textbf{jitter} * \frac{\textbf{ppm value}}{10^6} \qquad (1)$$

The following figure shows the flow chart for jitter model.

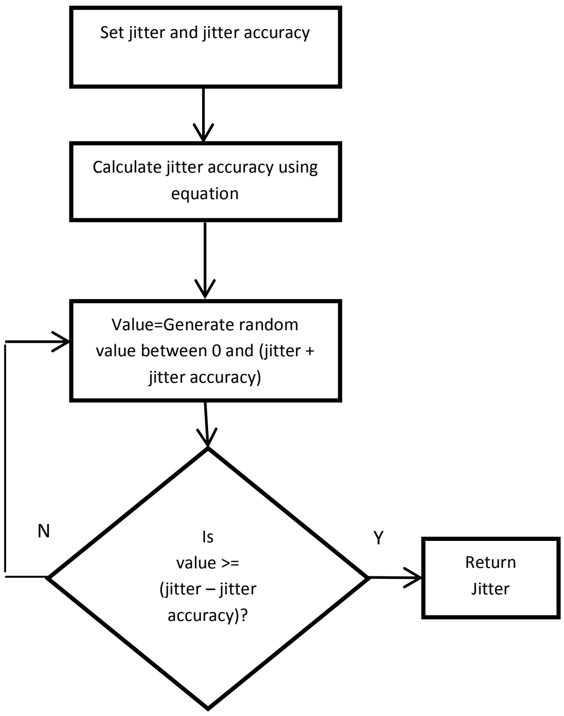

Figure 2: Flow chart Jitter model

C. Lane Skew:

Skew (or relative delay) can be introduced between lanes by both active and passive elements of a link. Skew is defined as the difference between the times of the earliest lane and latest lane for the one to zero transition of the alignment marker sync bits.

The pseudo code to model skew on lanes is given below.

Step 1 Set modify_skew = Maximum Skew.
Step 2 Receive lanes
Step 3 for (i=0; i<number of lanes; i++)
Step 4 {double x[i]←get_skew()}
Comment // Calculate skew for each lanes
Step 5 Send lanes out.

get_skew() function is used to calculate skew for all lanes based on specified maximum lane to lane skew. Addition of calculated value of lane skew value must not be greater than specified Maximum lane to lane skew.

Following are the steps to generate random skew for each lane:

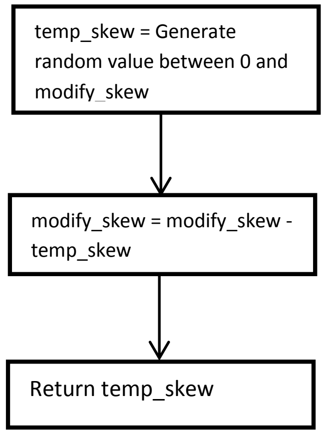

Figure 3: Flow chart Skew model

D. Cable delay:

Cable delay is the amount of time required for cable to transmit signal to travel from the transmitter to the receiver. It can be computed as the ratio between the link length and the propagation speed over the specific medium.

$$\textbf{Cable delay} = \frac{\textbf{channel length}}{\textbf{propagation speed}} \qquad (2)$$

Channel length represents the length of channel or physical medium in meter; Propagation speed depends on the physical medium of the link (that is, fibre optics, twisted-pair copper wire, etc.) in meter per second (m/s)

E. Return Loss:

Return loss is a measure of the reflected energy from a transmitted signal. These reflections are caused by impedance discontinuities in the channel. These discontinuities may be due to several things such as connectors, improper manufacture. Any energy that is reflected reduces the power of the transmitted signal. It is commonly expressed in positive dB The Larger the value, the less energy that is reflected.

Return loss can be calculated using the following equation:

$$\mathrm{R}\mathbf{L} = \mathbf{20 \log} \left| \frac{\mathbf{Zi-100\Omega}}{\mathbf{Zi+100\Omega}} \right| \qquad (3)$$

Zi = input impedance of the cable at a given frequency.

The above equation produces the value of RL as a negative number. The standards refer to all measured values in a positive format. For comparison to a specification, this conversion is accomplished by placing a negative sign in front of the equation.

$$\mathrm{RL} = -20\log\left|\frac{\mathrm{Zi}-100\Omega}{\mathrm{Zi}+100\Omega}\right| \qquad (4)$$

F. implementation

Following flow chart describes noise channel model flow in implementation of blocking transport interface (b_transport).

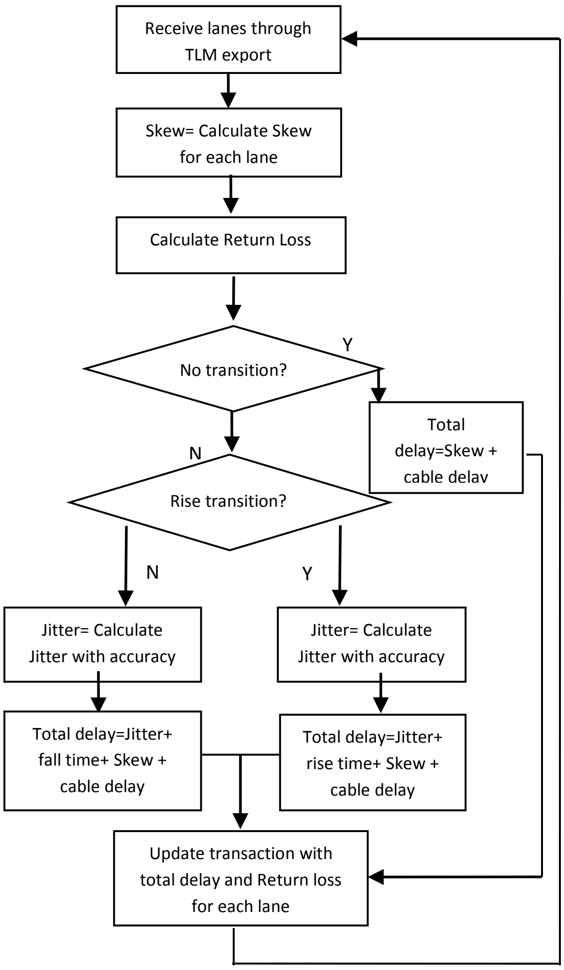

Figure 4: Flow Chart of Noise Channel Model

IV. RESULTS AND DESCRIPTION

UVM generates randomized bit value for each lane as transaction and sends to SystemC noise channel model. SystemC models computes delay based on timing and noise parameters specified by user and updates total delay for each lane as transaction.

Following timing and noise parameters are considered during the simulation. They are rise time in ps; fall time in ps; Jitter in ps; Jitter accuracy in ppm; Maximum lane to lane skew in ns; Channel length in meter and propagation speed in m/s for cable delay; Input impedance of cable at a given frequency in ohms for return loss

For the specified timing and noise parameter SystemC computes total delay for each lane to propagate corresponding lane value. Following is the description shows simulation results for specified value of noise parameters for the experiment setup.

Number of lanes = 4; Rise time = 2.0ps; Fall time = 1.0ps; Jitter = 2.0ps; Jitter accuracy = 100000ppm; Skew = 29.0ns. To calculate cable delay, the Channel length = 10m; Propagation speed = 200000000 (in m/s depends on channel). To calculate return loss, Input impedance = 80.0ohms (for a given frequency)

Initially all lanes are initialized to Zero value. We have generated nine packets of lane values and passed to SC model for delay computation. SystemC model Calculates lane to lane skew for each lane in picosecond as per the algorithm described in lane skew during simulation.

Table 1: Calculated lane to lane skew

Lane	Calculated lane to lane skew in ps
0	26569.7
1	1545
2	635.056
3	250.29

SystemC model calculated Cable delay as 50000ps; return loss in as 19.0849dB. The total delay for each lane based on described formula and updates transaction. All lanes simulation is tabled in Table 2.

Table 2: Calculated total delay

Total Calculated delay in ps			
Lane 3	**Lane 2**	**Lane 1**	**Lane 0**
50254.4	50639.1	51548.8	76569.7
50250.3	50637.9	51548.2	76569.7
50250.3	50639	51549.1	76569.7
50253.2	50638.1	51548	76573.8
50254.3	50635.1	51545	76569.7
50250.3	50639.1	51549	76572.5
50250.3	50637.9	51547.9	76573.6
50250.3	50638.9	51549.1	76569.7
50253.3	50635.1	51545	76569.7

V. CONCLUSION

This paper describes the implementation of generalized noise channel model at transaction level using SystemC and verified for noise parameters like jitter, lane to lane skew, transition time, cable delay which will be useful for performance analysis of communication protocol at higher abstraction level.

REFERENCES

[1] "SystemC 2.0.1 Language Reference Manual", Rev. 1.0, Open SystemC Initiative, 2003.

[2] "SystemVerilog—Unified Hardware Design, Specification and Verification Language", IEEE Computer Society, February, 2013.

[3] "Universal Verification Methodology (UVM) 1.1 User's Guide", Accellera, May, 2011.

[4] "Universal Verification Methodology (UVM) 1.1 Class Reference", Accellera, June, 2011.

[5] Victor Berman, "A Tale of Two Languages_ SystemC and SystemVerilog", Chip Design Magazine, July, 2005.

[6] "IEEE 802.3ba 40 and100 Gigabit Ethernet Architecture", IEEE Standard, June, 2010.

[7] "Universal Serial Bus 3.0 Specification", Revision 1.0, November, 2008.

[8] Mindshare Inc, Ravi Budruk, Don Anderson, and Tom Shanley, "PCI Express System Architecture" Addison-Wesley Professional, 1st edition, September, 2003.

[9] John Aynsley,"OSCI TLM-2.0 Language Reference Manual", Doulos, July, 2009.

[10] Agla, K.A., Toepfer, H. ; Wetzstein, O. ; Ortlepp, T., "Modeling of noise-induced jitter in low-power mixed-signal circuits using a SystemC highlevel description", Telecommunications Forum (TELFOR), pp. 943-946, Belgrade, 2012