A Process for Serial Link Signal Integrity Analysis

Guest Author Donald Telian, Signal Integrity Consultant

Serial links have caused a number of changes in Signal Integrity (SI) tools, models, and methods. While many have collaborated over the past decade to develop the necessary tools and models for serial links, a well-defined process for using them is still emerging. This article will describe an SI analysis process that has been successfully deployed on over a dozen high-speed serial interfaces with data rates ranging from 1.5 to 6.4 Gbps.

The process described below assumes access to three basic serial link tools: (1) a Time Domain (TD) simulator, (2) S-Parameter (SP) generation and simulation, and (3) a Channel Analysis (CA) simulator. While the TD simulator might be the same one used for lower-speed busses, the newer SP and CA tools are often specifically designed for serial link SI. CA tools normally derive a stressed eye from a link's impulse response, and come in a variety of open-market and proprietary flavors. For data rates over 5 Gbps, the CA tool typically provides a way to simulate complex receive equalization.

One of the primary differences between serial link SI and traditional SI is that the tools and models do not yet automate the task of analyzing corner cases. For example, it's not possible to simply ask for "fast" and "slow" corner case analysis and expect the tools to understand what you mean. And even if they did, the data to assess those corners is not yet available in the models in any standardized way. And so we're back again to sound engineering judgment and creativity.

Table 1 defines a serial link analysis process, and each step is detailed further below.

Table 1. Serial Link System Analysis Process Steps

Step	Task	Purpose	Output
1	Collect and Connect Models	Build Link Model	Link Ready-to-Run
2	Model Sanity Check	Verify Models	TD Functional
3	Quantify Loss & Crosstalk	Understand & Gauge Link	S21 dB, mV RMS
4	Plot Impulse Response & ISP	Measure ISP, Calculate #bits	#bits for CA
5	Verify Eye Convergence	Test #bits, Confirm Coverage	CA Functional
6	Parameter Determination	Setup for Worst-Case	CA Parameters
7	Corner Case Analysis	Derive Design Margins	Eye h/w Margins

1. Collect and Connect Models. This step involves collecting and/or building all the active and passive sub-models and connecting them to form the end-to-end link. Typically the first question to answer is the format of the active SerDes Transmit (Tx) and Receive (Rx) models. Some models may only work in a certain simulator, while newer IBIS-AMI models target numerous simulators. Either way, you'll need to ensure you have viable simulation engines in place for the task at hand. For the passive elements, be sure all necessary vias, connectors, and series capacitors are included. Via models must be field-solved with ground stitching vias in place in to get an accurate description of loss. Approximate trace construction parameters as needed. If models are not available for a number of the passive elements in the link such as cables and connectors consider building a prototype and measuring (or sub-contracting the measurement of) the S-Parameters of the end-to-end assembly. At times this is simpler - and perhaps even more accurate - than cascading numerous sub-models. The output of this step is a link that is expected to simulate correctly in TD simulation.

2. Model Sanity Check. Begin by running a short time domain simulation of the assembled link. Watch for simulation errors, unreasonable voltages or DC shifts, and signals not reaching their destination. When there are problems during this step it is often necessary to temporarily delete elements from the simulation and add them back one at a time until the problematic element is isolated. S-parameter models are particularly suspect of causing problems. When most of the models are new and untested start with just the Tx and Rx and add the models to the link one at a time, ensuring that each model adds a little more time delay and a little more loss. When all elements are in place, verify that the end-to-end time delay is reasonable and the voltages look correct. Figure 1 shows Tx (red) and Rx (blue) waveforms in a 6 Gbps channel with and without equalization. It's possible the eye will be completely closed at the Rx; this does not necessarily mean the simulation is incorrect, particularly at higher speeds. The output of this step is the ability to simulate in the time domain with confidence. Without achieving that, it's likely that none of the subsequent steps can be performed reliably.

3. Quantify Loss and Crosstalk. Quantifying factors such as loss and crosstalk can help double-confirm the link simulation is performing correctly since there is a direct correlation between insertion loss (an Rx/Tx transfer function) and the Tx and Rx voltages found in step two. Generate S-Parameters and measure S21 insertion loss. Take the value derived by the tool and double-check it against both the time domain voltages at the Tx and Rx and a hand calculation that sums the loss of each individual element. This will typically match within 10%. It's also important to gain an intuitive sense that can approximate performance across link metrics such as frequency, loss, and crosstalk. The outputs of this step include channel loss expressed in either dB or as a fraction. The latter reveals exactly what fraction of Tx voltage will appear at the Rx, while dB is less straightforward (though more common). Figure 2 plots the loss in a 6 Gbps link as -16 dB. A typical way to quantify crosstalk is to calculate the Root-Mean-Squared (RMS) value of the coupled signal on a quiet net. While most waveform tools do not calculate this number, it can be derived by exporting the waveform data to a spreadsheet.

4. Plot Impulse/Pulse Response & ISP. A link's impulse response can be output from most CA tools or can be approximated by injecting a single pulse into the Tx and measuring the pulse response at the Rx. The Interconnect Storage Potential (ISP) can be measured directly from the impulse/pulse response as described in [1]. Furthermore, equations in [1] explain how to use the ISP to calculate the number of bits a high-capacity simulation will require to converge on a stable eye diagram. It is also important to verify the integrity of the link's impulse or step response or the CA tool will not work correctly. Figure 3 shows an impulse response for a channel with many discontinuities and its measured ISP. Outputs from this step are a verified channel impulse response (some times referred to as the channel's

"characterization" or "fingerprint"), the number of bits to apply in CA, and a sense of confidence that that quantity of bits will be sufficient for coverage. This saves time during subsequent simulations, since you are not running more bits than necessary.

5. Verify Eye Convergence. CA simulation is run during this step, as various bit-stream lengths are overlaid to verify the eye converges correctly with the number of bits calculated in step 4. This value can be double-confirmed by comparing the number of bits derived with the knee on the bathtub curve plot. As such, the outputs of this step are a verified number of bits and a functional CA environment.

6. Parameter Determination. In order to derive meaningful results and design margins it's important to input correct parameters into CA for items such as jitter, bit patterns, duty cycle distortion (DCD), and crosstalk. There are many components of jitter, and the industry does not always use consistent terminology. During this step you will determine the various sources of jitter imposed by the Tx such as random, deterministic, sinusoidal, and periodic jitter and map them into the capabilities of the CA tool. Explore what types of data patterns are driven on the link (such as 8b/10b) or if there is a certain pattern (such as CJTPAT) that the link suggests to use. Find out what tap values are possible and how to apply them. Much of this data is extracted from datasheets and other specifications, and typically does not yet reside in the models. The outputs from this step are a complete set of CA parameters necessary to derive worst-case eye diagrams and design margins.

7. Corner Case Analysis. This step runs CA to derive relevant eye diagrams and bathtub curves. From these, the necessary design margins are determined. Total jitter or eye width is often specified at a certain BER or quantity of bits (e.g., 1e12 or 1e15) which typically must be extracted from a bathtub curve. If tolerances in the interconnect need to be tested, additional CA responses or characterizations must be derived. Tx tap values can often be explored and adjusted at this stage without a new characterization. Compare the CA results derived during this step with link specifications. Insufficient margin may cause you to iterate all or parts of this process. If the link is performing acceptably, be sure to capture all essential tap settings, component selections, and routing requirements/assumptions and communicate them to the correct firmware, procurement, or layout engineer.

The process described above has been applied successfully on serial links conforming to nine different industry standards operating at data rates from 1.5 to 6.4 Gbps. For a collection of design case studies detailing the proper application of this process on interfaces such as PCI Express, eSATA, and SAS-2 please visit the published works section at www.siguys.com.

References:

[1] "New Techniques for Designing and Analyzing Multi-GigaHertz Serial Links", Telian, Wang, Maramis, Chung – DesignCon 2005

Donald Telian is an independent Signal Integrity Consultant specializing in the analysis and implementation of multi-gigabit serial links. Before starting his consulting business in 2005, Donald architected and established serial link tools, model formats, and design methodologies for numerous electronics companies around the world. He has over 25 years of SI experience, has published numerous works in the field, and has taught SI techniques to thousands of engineers in more than 15 countries. Donald can be reached at telian@sti.net.

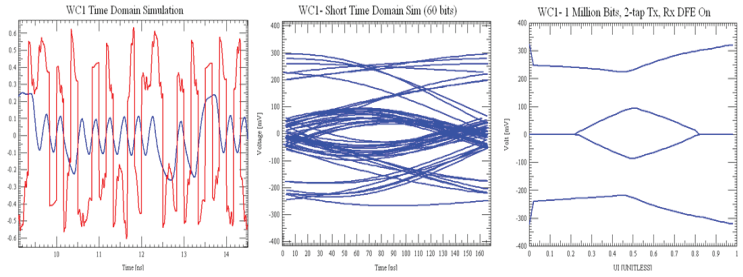

Figure 1

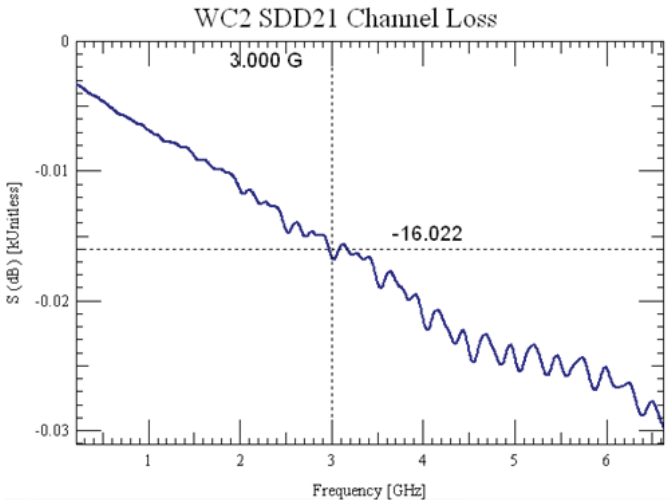

Figure 2

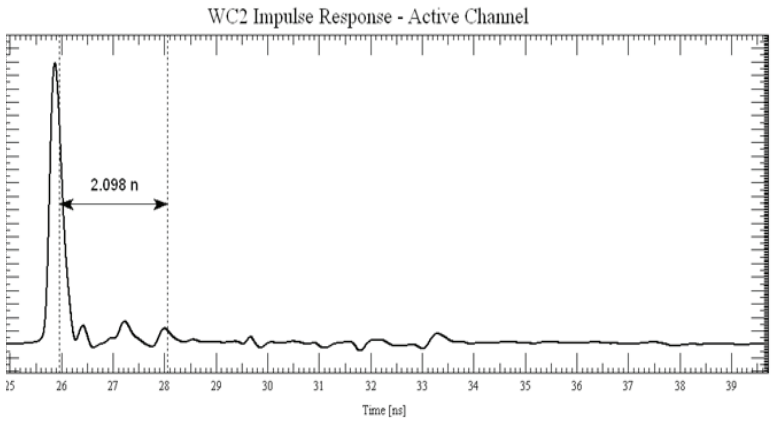

Figure 3