DesignCon 2014

High Speed Serial Link Simulation based on Dynamic Modeling

Xiaoqing Dong, Huawei Technologies
dongxiaoqing82@huawei.com

Wenyi Jin, Xilinx Inc.
wenyij@xilinx.com

Hang (Paul) Yan, Huawei Technologies
paul.yanhang@huawei.com

Daochun (Moore) Mo, Huawei Technologies
modaochun@huawei.com

Geoff Zhang, Xilinx Inc.
geoff.zhang@xilinx.com

Abstract

Today's high speed serial link simulators only perform static simulations. Static implies that only a fixed set of interconnect s-parameter files and a given silicon PVT corner behaviors are modeled and simulated. Thus, static simulation cannot always predict system performance under changing environment, in which system characteristics do change.

Dynamic simulation implies that interconnect parameters and silicon behavioral properties are changing with the environment. This is true because when the environmental parameters change, so does the performance of various blocks, such as equalizers, CDR, offset canceller, analog bandwidth, latch sensitivity, latency, and so on. Obviously, many static simulations do not add up to a dynamic simulation.

In this paper, we are proposing a fundamentally different modeling concept such that the above mentioned issues are addressed. Relying on the new modeling philosophy system bottlenecks could be more easily identified during simulation without having to wait until hardware test.

Authors Biography

Xiaoqing Dong joined Huawei Technologies in 2006 as a signal integrity research engineer. She has been working on high speed link SI simulation and measurement. She received her bachelor and master degrees in Communications and Information System from Harbin Institute of Technology, China, for research in Information and Communication Engineering.

Wenyi Jin received his Ph.D. from the University of Hawaii at Manoa in 2006 in EE with a focus on Information Theory and Coding. He joined Xilinx in 2011 and has been working on high speed SerDes architecture and modeling. From 2006 to 2011, he worked in LSI as a SerDes system architect. His current interests include CDR, FEC, Equalization, and simulation methodology.

Hang (Paul) Yan joined Huawei Technologies in 1998 as a signal integrity engineer. His work includes design and analysis of high speed component, validation and simulation of backplane channels, power integrity analysis of PCB and packaging technology. He is a high-speed interconnect system architect in the engineering department in Huawei. Paul has a bachelor degree in micro-electrical technology from Southeast University, China.

Daochun (Moore) Mo joined Huawei Technologies in 1997 as a hardware test engineer and design engineer in the area of high speed system research. His current interests include passive channel analysis, such as connectors, cables, PCB, etc., and high speed system verification. He received his bachelor degree in electromagnetic field and microwave technology from Beijing Broadcasting Institute, China.

Geoff Zhang received his Ph.D. in 1997 in Microwave Engineering and Signal Processing from Iowa State University. He joined Xilinx SerDes Technology Group in 2013 to supervise the SerDes architecture and modeling group. Since 1997 Geoff has worked at Xilinx, Huawei, LSI, Agere Systems, Lucent, and Texas Instruments. His current work involves SerDes architecture modeling and high speed system level analysis.

Introduction

High speed serial link modeling has improved tremendously in the past decade. There are both public simulators and analysis tools, such as StatEye [1], and many proprietary models from both system equipment manufacturers and SerDes IP designers, such as IBM's HSSCDR, Xilinx's SimLab, Broadcom's LinkEye, and Rambus' LinkLab, etc. In general, these models are doing pretty well, except that they can only simulate with themselves, i.e., the TX and the RX models at least need to belong to the same vendor.

To solve the co-sim problem the industry established a simulation platform, the IBIS-AMI modeling [2]. There are about half a dozen EDA vendors who are supporting this practice. The most commonly used are Agilent ADS/SystemVue, Cadence/Sigrity SystemSI, SiSoft QCD, AnSys Designer, Mentor Graphics HyperLynx, and so on.

On the simulation methodology side, high speed link simulations can be performed in time-domain, statistical domain, semi-analytical, and/or some kind of combinations. However, all of them belong to a family we call "static" simulation. Static implies that only a fixed set of interconnect s-parameter files and a given silicon PVT corner model are simulated at a time. Thus, static simulation cannot always predict well system performance under changing environment.

In this paper, we propose a different simulation concept, which we call the "dynamic" simulation. By dynamic we imply that interconnect parameters and silicon corner properties are allowed to change as simulation proceeds. As we know when environmental parameters change, so do all the settings inside SerDes chip, such as equalizers, CDR, offset canceller, analog bandwidth, latch sensitivity, latency and timing, adaptation, and so on.

The proposed method can be readily adopted by IBIS-AMI simulation environment with minor modification to extend its analysis capability.

Static Simulation Background

A very important test with equipment manufacturers is called temperature ramping test. In this test the system is put in a temperature controlled unit, the temperature chamber, and various tests are performed. A likely scenario is that temperature ramps up and down at a predefined rate, say 1°C/min, dwelling at the lowest temperature (e.g., -40°C) and the highest temperature (e.g., +65°C) for a couple of hours. The setup is illustrated in Figure 1. The range of temperature is dictated by the targeted applications requested by service providers.

Let's exam a real case we encountered. The system runs at 12.5Gbps through a backplane system with two connectors. Both simulation and lab test showed that system ran with margin at low (-20°C), room (25°C), and high (80°C) temperatures. However, during the temperature ramping test it was found that if the system was started at low temperature, errors started to show up in some links when the temperature increased above 70°C. However, if the system was started at room or high temperatures, the tested system did not have errors.

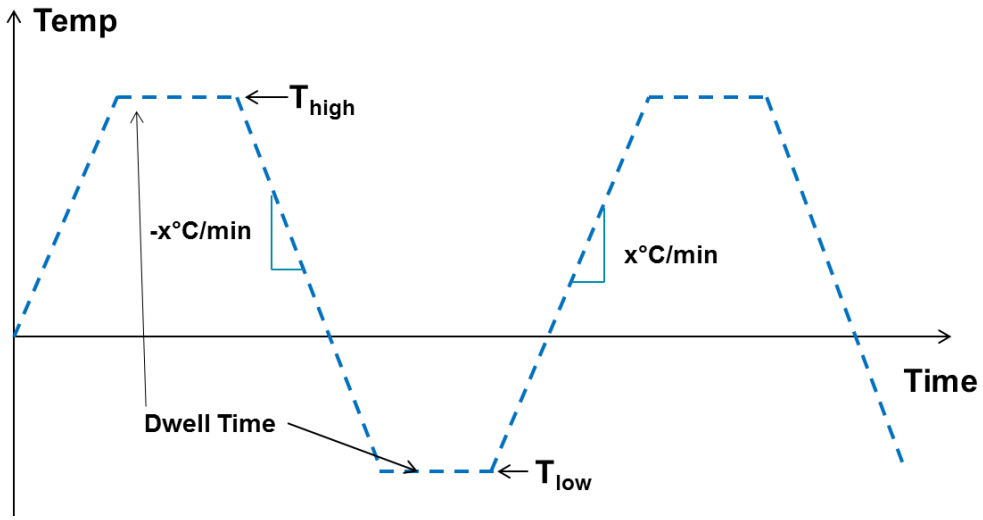


Figure 1. Illustration of temperature ramping profile.

After some investigation, it was discovered that the main cause is due to the insertion loss change of more than 6dB at the Nyquist frequency (6.25GHz) when temperature goes from the lowest to the highest. The simulated insertion loss profiles are given in Figure 2. The insertion loss covers from the TX BGA pad to the RX BGA pad.

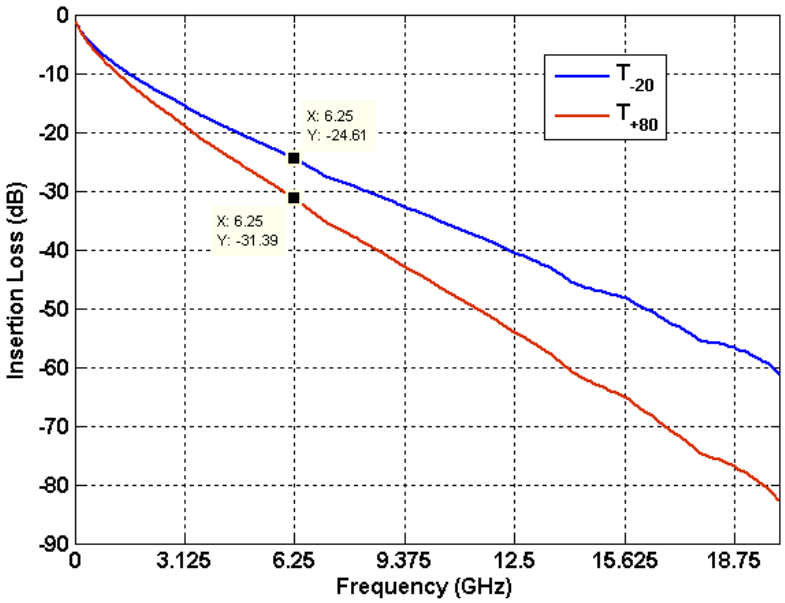


Figure 2. Insertion loss profiles at the low and high temperature.

Further work revealed that this SerDes receiver has both CTLE and DFE. While the DFE tap coefficients are adaptive all the time, the CTLE settings are not. They are auto-tuned at link startup after reset based on some eye metric searching algorithm. After that the CTLE settings remain unchanged unless a reset or forced auto-tuning is initiated. The changing environment plus the receiver design created the problem encounterd.

In order to duplicate what was observed, we modified the SerDes RX model in the following manner: We added a switch in the model such that the CTLE auto-tuning is either enabled or disabled. When enabled, the CTLE parameter searching works based on the designed algorithm. When the auto-tuning is disabled, the CTLE settings use the pre-programmed values. The simulation time has to set long enough for the auto-tuning process to finish and DFE adaption converges.

After the modification, we did simulations for the following two cases, Case 1 and Case 2. The TX side settings are always fixed, i.e., the same swing amplitude and same de-emphasis level. The pre- and post-cursor de-emphasis in our example together provides about 6dB equalization. The TX output eye diagram is shown in Figure 3.

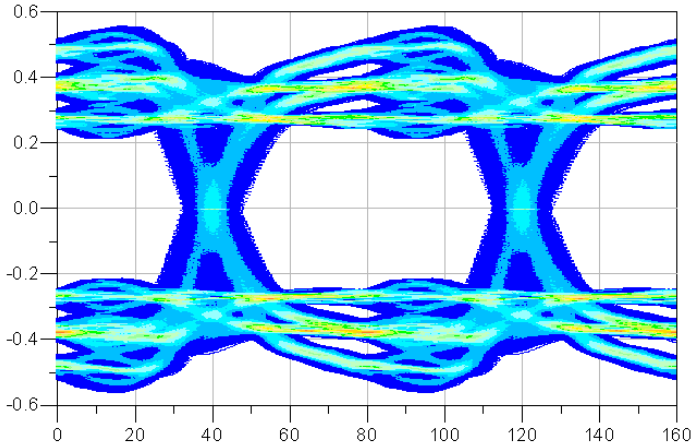
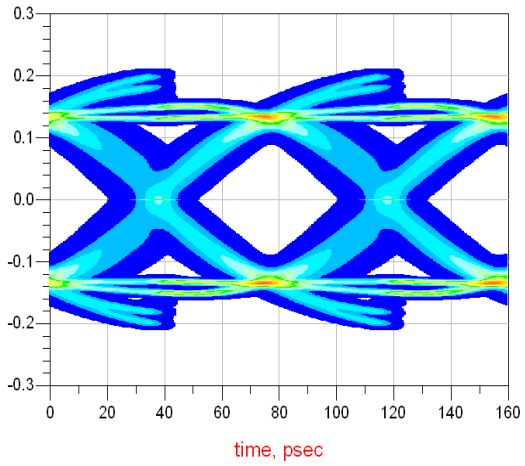


Figure 3. TX output eye diagram for the static simulation example.

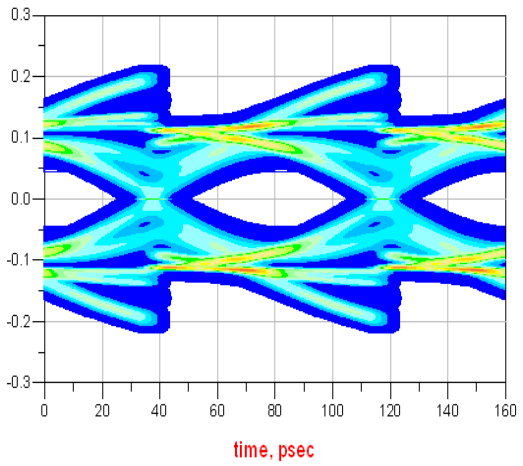
Case 1: System is started at low temperature

We simulate with CTLE auto-tuning enabled at low temp. Then we simulate at high temp with CTLE auto-tuning disabled, using the CTLE settings found from the low temp. The results are shown in Figure 4. We observe the following from Case 1:

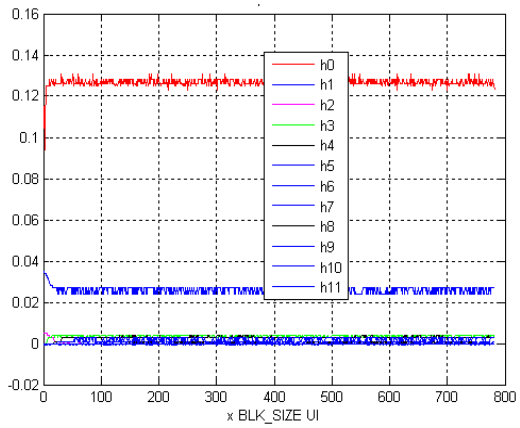
- When the system is started at low temp, the CTLE (Peaking and Boost) are auto-tuned for the given channel. The system shows a good margin.
- When the temperature increases to 80°C, the channel loss is more than 6dB higher. However, the CTLE does not provide adequate peaking, although there is still more equalization capability the CTLE can deliver. This is seen from the much higher relative values of DFE h1 and h2 with respect to h0, which is the error latch value.
- More importantly, CTLE does not provide enough boost when the signal strength is weakened due to extra loss. The result is that h0 settles at a much lower value, around 90mV, instead of 130mV at the low temp.
- All in all, when the system is started at low temp, it loses a lot of margin at high temp, due to the fact that CTLE is not continuously adapting to adjust its contribution.

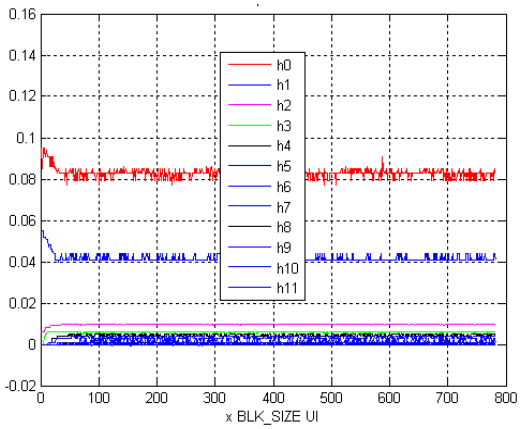


(a) Eye at low temp with BER=8.78E-36;



(b) Eye at high temp with BER=3.37E-13





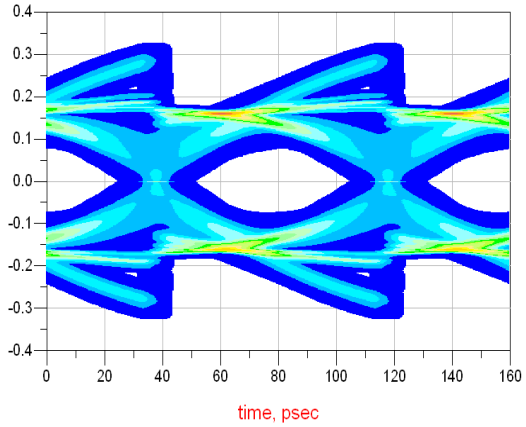
(c) Converged DFE settings for low and high temp.

Figure 4. Simulated results for Case 1.

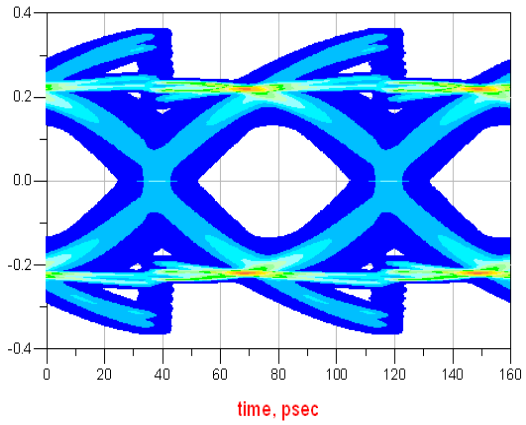
Case 2: System is started at high temperature

We simulate with CTLE auto-tuning enabled at high temp. Then we simulate at low temp with CTLE auto-tuning disabled using the CTLE settings found from high temp. The results are shown in Figure 5. We observe the following from Case 2:

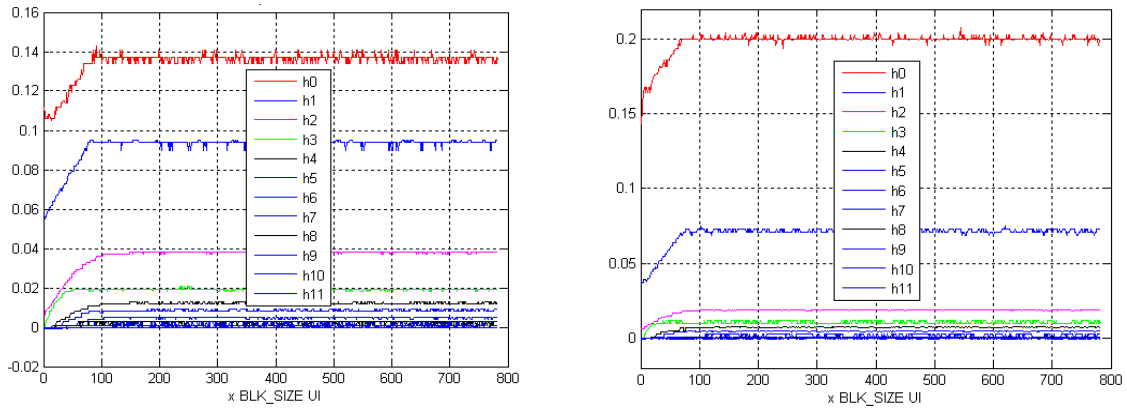
- When the system is started at high temp, the CTLE (Peaking and Boost) are auto-tuned for the given channel. The system shows a good margin.
- When the temperature decreases to -20°C, the channel loss becomes less and the signal is stronger reaching the RX. The h0 value is larger and DFE tap coefficients are smaller, implying that less ISI cancelation task is required from the DFE.
- For this particular case, even though the CTLE was tuned at high temp, at low temp the system actually achieves more performance margin, based on this particular design.
- As a result, when the system is started at high temp, it works robustly in performance for all temperatures.



(a) Eye at high temp with BER=6.32E-20;



(b) Eye at low temp with BER=2.51E-27



(c) Converged DFE settings for high and low temp.

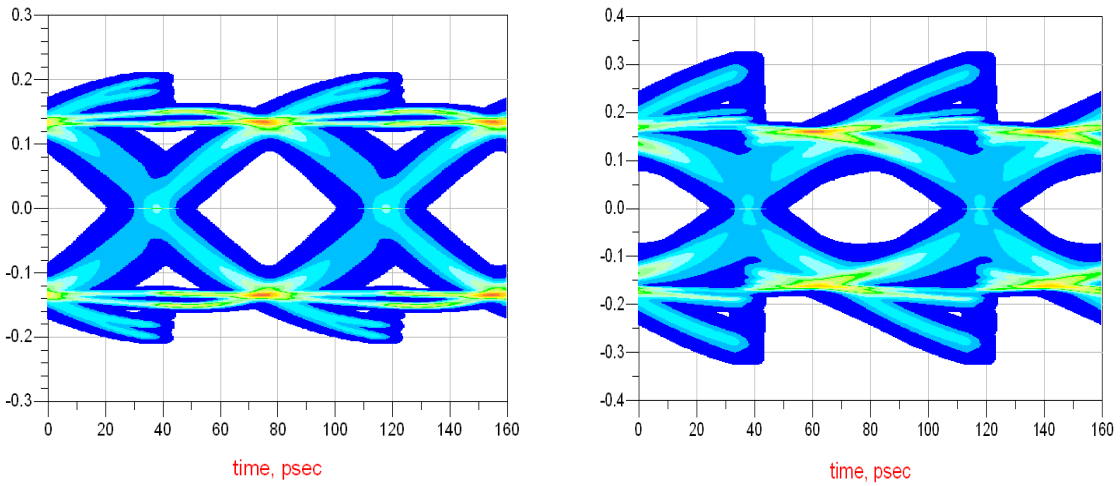
Figure 5. Simulated results for Case 2.

Relating Case 1 and Case 2 to reality

Now, most users of a targeted SerDes model are not doing the way described above. What they most likely do is to plug in high temperature link parameters into the simulation setup. If they also have options to change SerDes model conditions, they will do that. They will then simulate the link performance. When this is finished, they repeat the simulation for the low temperature, and maybe room temperature.

The result is we will not see (b) in Figure 4 or Figure 5. Only (a) in the two figures is available, unless the SerDes vendor provided enough knobs, and the user has a good picture of the link and knows how to manipulate simulation settings. As a result, the likely outcome is we only see the imagined performance for low and high temperatures re-plotted in Figure 6. We then confidently conclude that the link has good performance margin to cover the whole range of temperature range.

We need to find a solution to avoid such scenario to happen. Dynamic simulation is the method for this purpose. We will discuss its details in the second half of this paper.



(a) Eye at low temp with BER=8.78E-36; **(b) Eye at high temp with BER=6.32E-20**

Figure 6. Eyes and BER are likely to be interpreted and accepted.

Dynamic Simulation Concept

The above examples belong to the traditional simulation we call static simulation. Each simulation has its own set of initial conditions. We can model different environment and/or device PVT to a pretty good accuracy for estimating a link channel performance [3]. However, in most applications the environment is changing [4], and we need to capture this changing impact, which the static simulation cannot deliver.

In this paper we propose a novel approach to model the changing system. The fundamental concept is the following: If we start the simulation of a system at temperature T_1, the interconnect model and the silicon model are based on this temperature. When temperature changes to T_2, we cannot simply change conditions at T_2. Doing so is equivalent to making two static simulations. It is very important to note that adding a series of static simulations does not produce a dynamic simulation.

To better explain the difference between static simulation and dynamic simulation, let's look at a couple examples. First, we use CTLE as an example to illustrate the concept. A simplified one stage CTLE is used for this purpose. As shown in Figure 7, if currently the CTLE is running at setting *K*, the simulation has to include the two neighboring settings, *k*-1 and *k*+1, simultaneously, in order not to introduce signal discontinuity to the system when there is a call to change the CTLE setting.

As simulation goes on, if the adaptation algorithm requires to update the CTLE setting from *k* to *k*-1, then the output from CTLE *k*-1 is switched into the "mux" output. Meanwhile, *K*-2 is switched into the simulation and *K*+1 is dropped out of the simulation. In short, at least the neighboring settings need to be simulated simultaneously. This represents the fundamental concept of dynamic simulation.

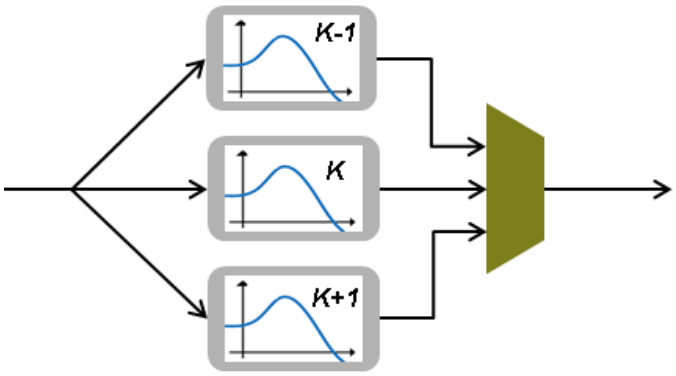


Figure 7. Illustration of a CTLE block used for dynamic simulation.

However, different blocks may not be treated identically as above. Let's explore another example. Assume we have a backplane channel whose characteristics change with temperature, which changes slowly with time. If the end-to-end channel is represented by its impulse response (terminations are included), we can implement it in the form of a FIR filter. In a static simulation, there is only one fixed set of FIR taps (with pre-defined oversampling rate). However, for dynamic modeling, since the system changes with time, the FIR tap coefficients also change with time. This is depicted in Figure 8.

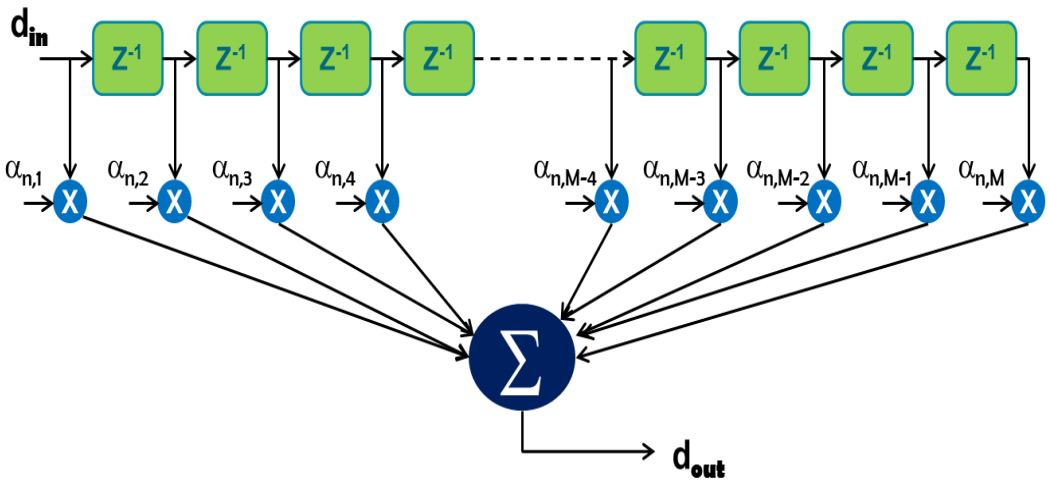


Figure 8. FIR filter representing a passive link channel.

It is important to remember that one set of filter coefficients cannot simply be replaced by another set. Doing so would destroy all the initial status and cause discontinuity in the signal at the channel output; after all, this does not exist in physics. Such discontinuity could ripple through many blocks. The effect is not really predictable, depending on many factors.

Now, we represent the link from T_{low} to T_{high} by a total of *N* states. *N* is so chosen that the insertion loss difference between the two neighboring states is small enough so as not to introduce too much disturbance to the signal (Figure 9). Of course, too large the *N* would increase simulation time.

FIR_1	$\alpha_{1,1}$	$\alpha_{1,2}$	$\alpha_{1,3}$	...	...	...	...	$\alpha_{1,M-1}$	$\alpha_{1,M}$
FIR_2	$\alpha_{2,1}$	$\alpha_{2,2}$	$\alpha_{2,3}$	...	...	...	...	$\alpha_{2,M-1}$	$\alpha_{2,M}$
...	...	...	...	...	...	...	...	...	...
FIR_N	$\alpha_{N,1}$	$\alpha_{N,2}$	$\alpha_{N,3}$	...	...	...	...	$\alpha_{N,M-1}$	$\alpha_{N,M}$

Figure 9. A bank of FIR-tap coefficients.

Another type of modeling is for some calibration loops, for example, the data sampler offset. If the calibration only works at startup, the calibrated result is optimal for the current condition. As temperature changes the calibration is not re-performed again, resulting in suboptimal performance. Sometimes it could be worse than no cancellation is applied. The model should include this effect and implement it as some conditional branches in the model.

In the next section, we will provide an example to provide a clearer picture of dynamic simulation.

Dynamic Simulation Examples

In this section we consider a serial link at 10Gbps undergoing a temperature ramping test. The environment changes from -25°C to 85°C. Channel s-parameters for this example at different temperatures are simulated, and the insertion loss is correlated at 25°C with the measured data. We choose 10°C as the step size. The insertion loss profiles are shown in Figure 10. There are a total of 12 cases from -25°C to 85°C, i.e., [-25, -15, -5, 5, 15, 25, 35, 45, 55, 65, 75, 85] °C.

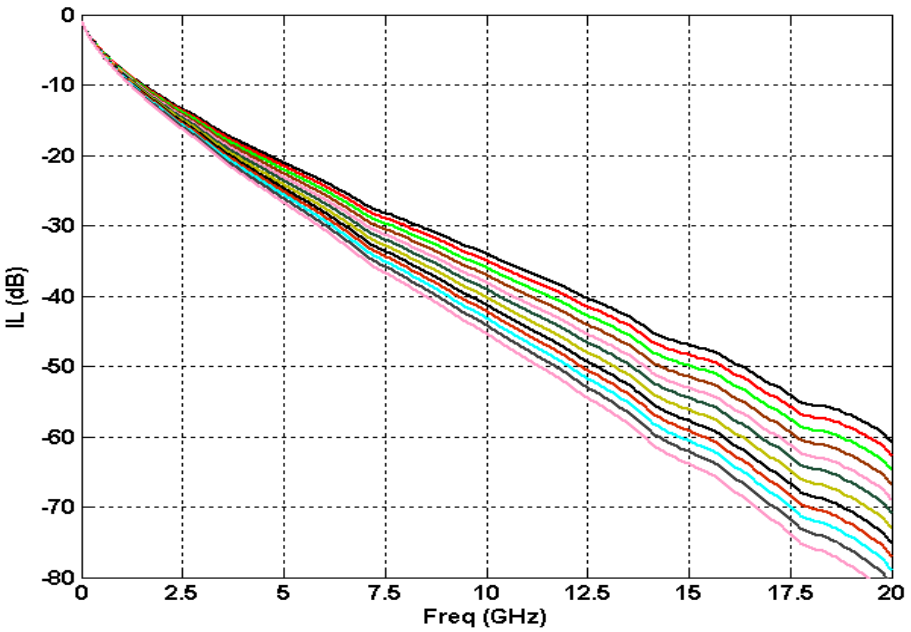


Figure 10. Insertion losses at 12 temperatures.

Besides link insertion loss change, we have also considered the following eight changes with temperature and implemented in the model (Figure 11).

TX driver bandwidth	CDR loop
RX front-end bandwidth	Latch offset
CTLE transfer functions	Intrinsic jitter
DFE loop bandwidth	Intrinsic noise

Figure 11. Blocks considered as functions of temperature.

For example, the CTLE transfer functions of one of the stages at -25°C (blue) and 85°C (red) are shown in Figure 12.

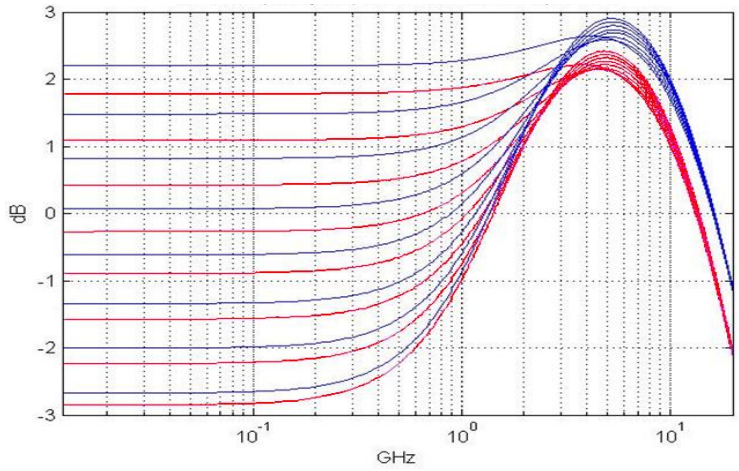


Figure 12. CTLE transfer functions at low and temperatures.

The practical problem we have to consider is the number of simulated bits. For example, if the dwell time is 2 hours, 10Gbps data stream we need to simulate 2*60*60*10e9 = 7.2E13 bits. Obviously, we can only run on a much reduced scale. However, we have to make sure the number of bits is large enough for all the loops to achieve convergence. Therefore, some planning is necessary before the simulation.

We simplify the example by assuming the ramping profile as shown in Figure 13, by removing the dwell time to shorten the simulation. We also assume that the temperature of the backplane and the device are the same without losing generality. If not we just need to have the profile and include a more complicated logic for the simulation.

In this example we decide to map 200K bits of simulation to 1 minute of running, leading to a ratio of 1 to 3 million. Hence, to complete one cycle in Figure 13 we need to simulate (24-1)*200K = 4.6M bits. The initial convergence will be seen from the final result, and will be excluded from eye plotting.

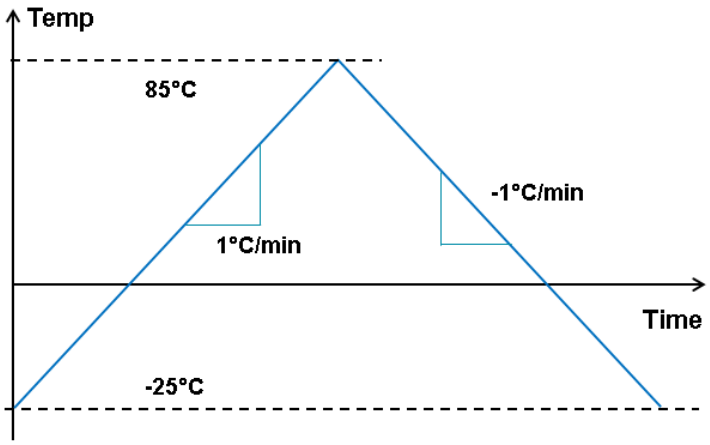


Figure 13. The temperature ramping profile for simulation.

We built the whole link model in Matlab/Simulink. The block for handling the 12 channel temperature data is shown in Figure 14, as an illustration.

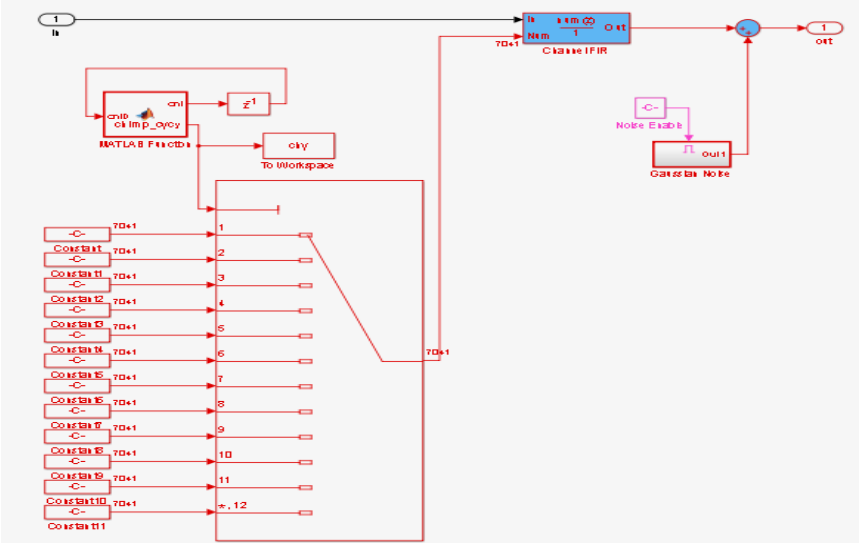


Figure 14. Simulink block for handling link channels.

The SerDes model we use has a TX 3-tap FFE de-emphasis, a RX CTLE (two peaking stages and one boost stage), and an 11-tap DFE. The TX de-emphasis is set to a fixed amount for the example here. The TX output swing is also pre-set.

For better understanding of dynamic modeling we choose to consider two CTLE Boost configurations, one has the Boost hard coded and the other allows the Boost to adapt. CTLE peaking filters and DFE taps are always adaptive throughout the simulation.

The adapted settings for the fixed Boost case are summarized in Figure 15. For more intuitive understanding, the timing is aligned for all pictures.

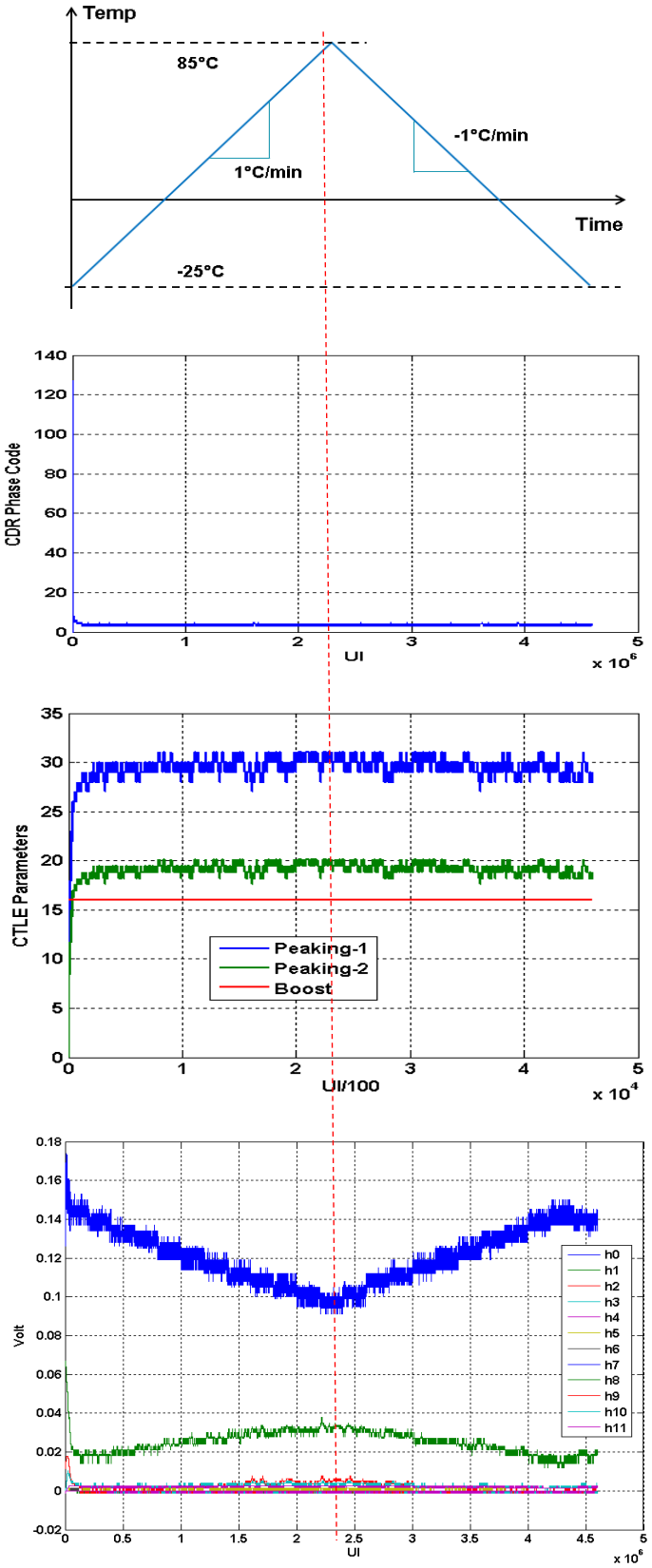


Figure 15. CDR, CTLE, and DFE convergence when Boost is forced to code 16.

We observe the following:

- The CDR settles in a tight range throughout the simulation when temperature ramps up and down.
- As the Boost is fixed, signal strength, h0, decreases when temperature rises and climbs back when temperature falls.
- The CTLE, based on a given algorithm, does not change much during temperature ramping, and Peaking-1 is saturated.
- As temperature rises, h1 increases to account for more ISI due to more channel loss. Relative to h0, h1's effect is more than its absolute value change.
- The values of h2 and h3 are small, but do increase around high temperature.
- Beyond tap 3, the contribution from the remaining DFE taps is negligible across the temperature range.

If we approximate h0 as signal strength, we can estimate the DFE equalization capability in terms of the familiar dB number. This is computed and plotted in Figure 16. Due to the fact that CTLE Peaking did not change much, and Boost was fixed, the channel insertion loss change has to be picked up by DFE. This is actually very close to the simplified estimation.

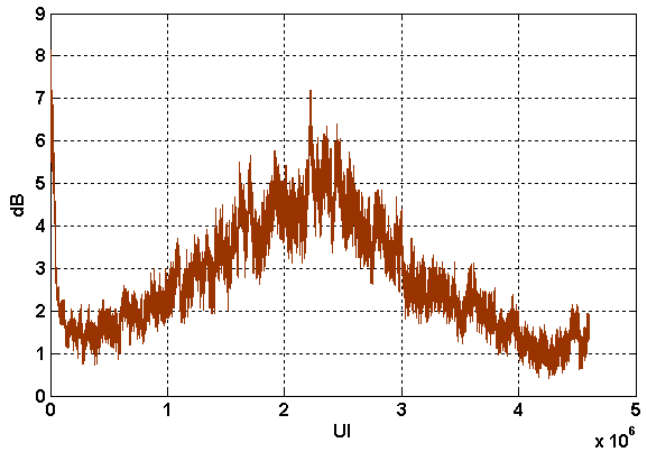


Figure 16. Estimated DFE equalization provided as time evolves.

The adapted settings for the adaptive Boost case are summarized in Figure 17.

- Whenever Boost changes (induced by a temperature change), the CDR also shifts accordingly, as the Boost block itself has phase information.
- Signal strength, h0, is directly impacted by Boost. Large Boost yields large h0.
- The CTLE does not change much during temperature ramping, and Peaking-1 is saturated.
- When h0 increases, h1 will change accordingly to deliver adequate equalization.

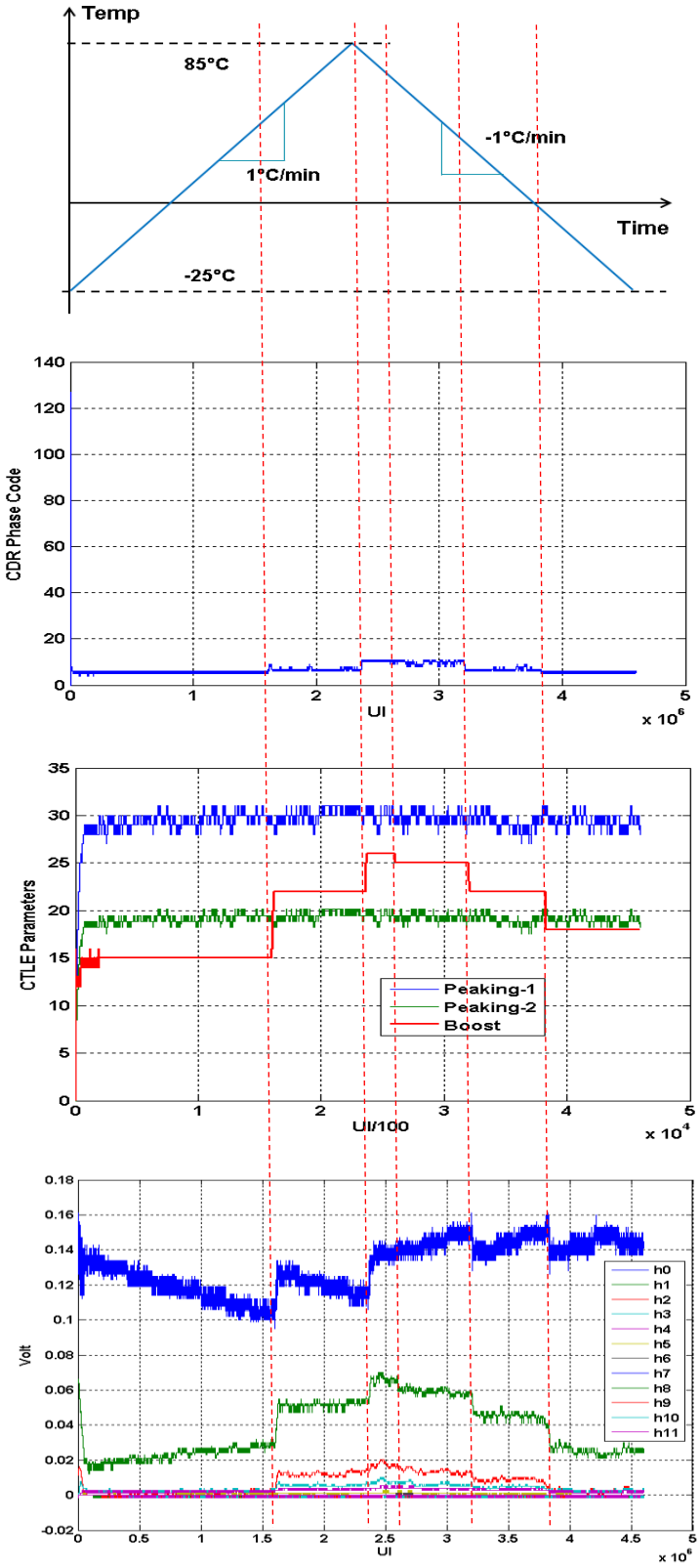


Figure 17. CDR, CTLE, and DFE convergence when Boost is allowed to adapt.

A very important observation is that for the same temperature, SerDes settings are not necessarily the same. This conclusion cannot be obtained from static simulations.

Let's zoom into one area to understand better what is happening. Let's look at 1.5 – 1.7 million bits, as shown in Figure 18. We have to note that most loop bandwidths are artificially increased in order to simulate within reasonable amount of time. So what we feel like sudden change actually takes much longer time (or takes many more bits) to complete.

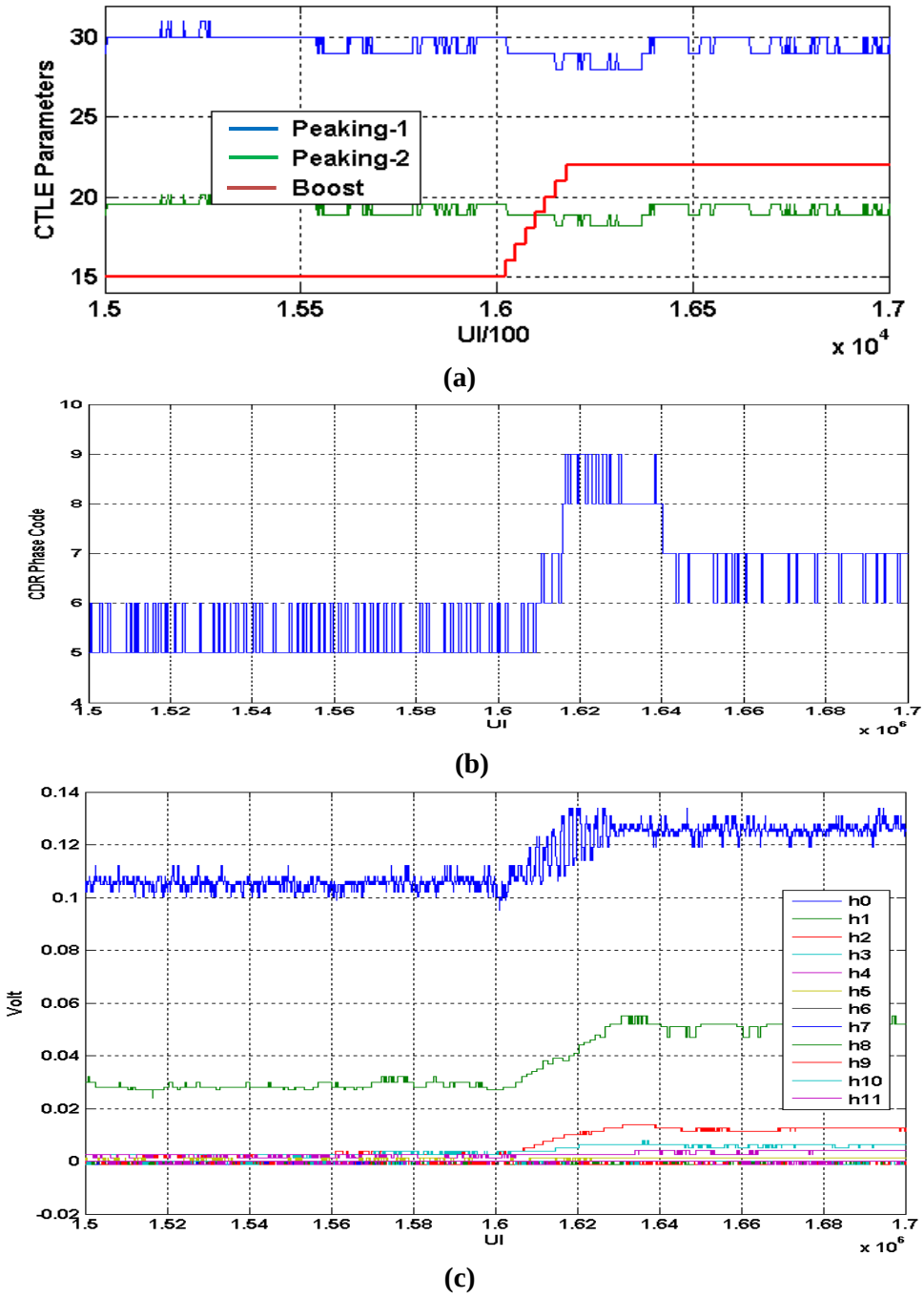
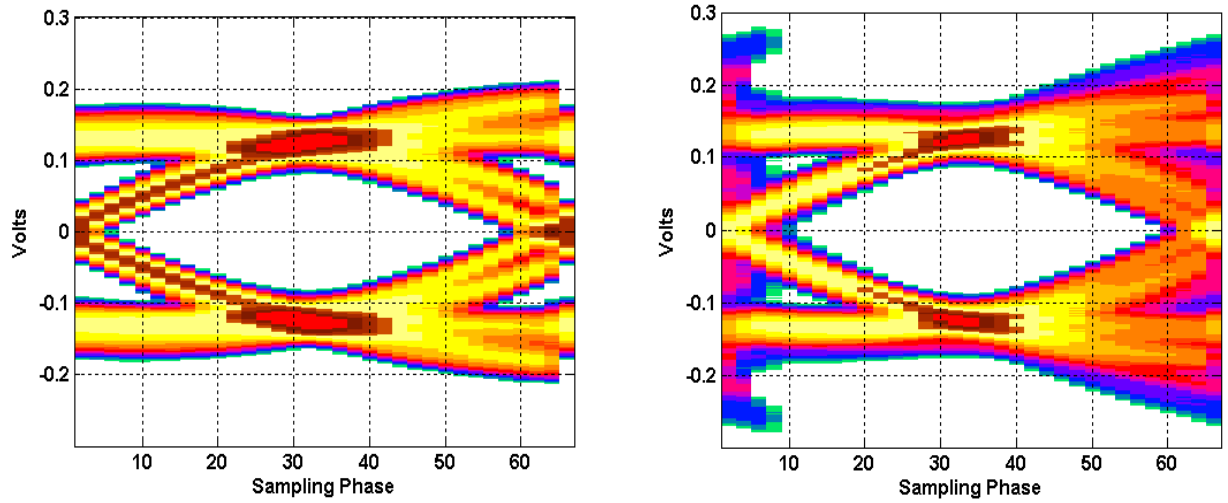


Figure 18. Zoomed in look at loop convergence.

At around 1.6M-th bit, the Boost decided to increase. This change triggers changes in CDR phase and DFE tap coefficients. The CTLE actually also tracked the "disturbance" to some extent as well.

It is seen that when Boost increases, so does h0 (blue curve in Figure 18 (c)). In order to achieve the desired equalization, tap h1 (green curve in Figure 18 (c)), tap h2 (red curve in Figure 18 (c)), etc. also increased. The increase of DFE tap coefficients in strength actually went a bit ahead, resulting in CTLE's decrease in peaking values for some time. When Boost settled, CTLE and DFE also followed. For all other "abrupt" changes of Boost, increasing or decreasing, similar interpretations hold.

Figure 19 shows the accumulated eye diagrams for the two cases described above. The first 100K bits are ignored to allow for the adaptation to converge. Adapted setting variations, including CDR sampling phase, are reflected in the eye.



(a) Boost fixed **(b) Boost adaptive**

Figure 19. Data eye at sampler accumulated over 4.3M bits.

All the actions happen automatically inside the SerDes. They together are trying to improve the signal integrity based on designed algorithms. However, in reality, this may not be the case. The fine differences may not be able to be revealed by static simulations. Our example shows the value of the proposed concept of dynamic modeling.

For this example, when Boost is allowed to adapt, the system actually sacrifices a little timing margin. However, this is device dependent (hardware and adaptation algorithm). On the other hand, to freeze Boost requires that we configure a good Boost setting, and that we are aware of and understand well of other impacts and consequences.

For the case in which Boost is adaptive, let's now zoom into the area around 1.4M to 1.8M bits so that we have a window of 400K bits. We will plot four eye diagrams with bits overlapping, as shown in Figure 20.

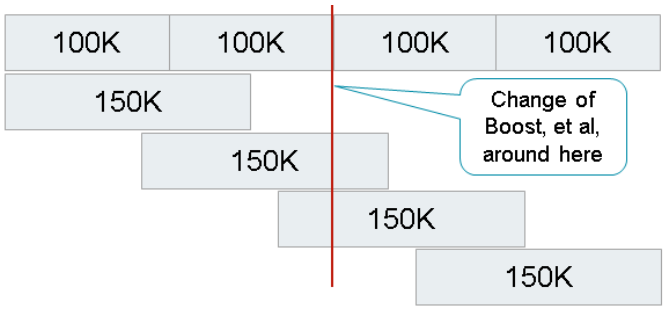


Figure 20. Bit sequence from which eye diagrams are generated.

The eye diagrams are given in Figure 21. The same scale is used for easy comparison. Before and after transitions, (a) and (d), the eyes are cleaner. During transitions, (b) and (c), we see some relative shifting between clock and data. However, the degradation is minimal. Particularly, the actual loop bandwidths are smaller. In addition, the increased Boost is clearly visible. This will not be revealed in static simulations.

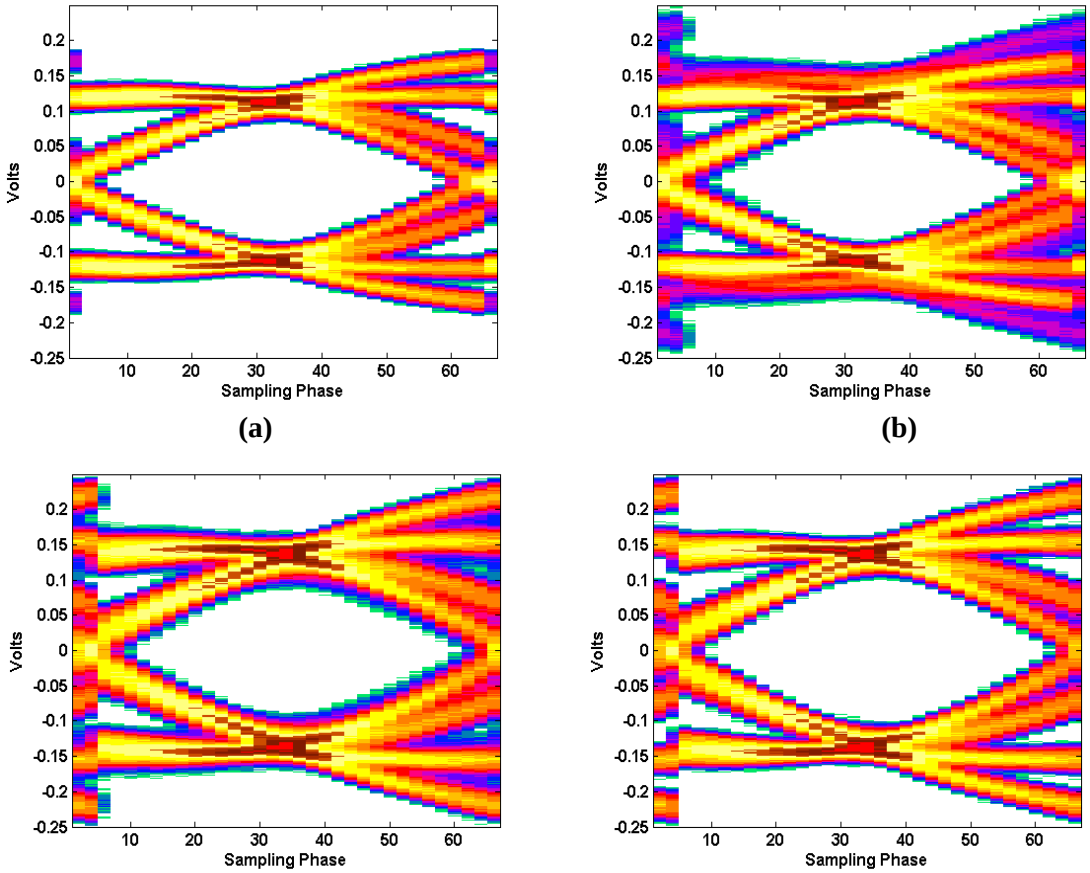


Figure 21. Four eye diagrams showing the impact from adaptation parameter change.

Conclusions

We have shown in the paper the basic concept of static simulation and dynamic simulation through examples. Although static simulation can reveal most problems within a high speed link system, there are situations in which dynamic modeling can do more. The downside of dynamic simulation is its longer simulation time and requirement of more depth knowledge of the whole system and capability to model. In addition, a good planning prior to simulation is very important to achieve what is desired. Dynamic modeling can be applied to IBIS-AMI without much effort.

References

[1] http://www.stateye.org/

[2] http://eda.org/pub/ibis/ver5.0/ver5_0.pdf

[3] Ravi Kollipara, Ben Chia, Qi Lin, and Jared Zerbe, "Impact of Manufacturint Parametric Variations on Backplane System Performance", DesignCon 2005.

[4] Greg Sheets and John D'Ambrosia, "The Impact of Environmental Conditions on Channel Performance", DesignCon 2004.

[5] Huawei internal documents for Interconnect Design

[6] SimLab: Xilinx internal high speed serial link simulator