

DesignCon 2017

IBIS-AMI Modeling and Simulation of Link Systems using Duobinary Signaling

Hongtao Zhang, Xilinx Inc.
hongtao@xilinx.com

Fangyi Rao, Keysight Technologies
fangyi_rao@keysight.com

Timothy De Keulenaer, BiFast
timothy@bifast.io

Ken Ly, Cisco Systems
cclly@cisco.com

Ramses Pierco, BiFast
ramses@bifast.io

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

Abstract

Duobinary signaling for high speed serial communications has regained public focus in the last several years. This is particularly true when dealing with data rate beyond 50Gbps with single channel over copper applications. With limited experience with duobinary signaling, system and SI engineers need to rely on end-to-end system level simulations. Today, IBIS-AMI specifications only support NRZ and PAM4 signaling. Hence, the existing IBIS-AMI specifications have to be modified such that duobinary signaling can also be supported.

In this paper duobinary basics are first presented. The extension of IBIS-AMI for duobinary signaling is then discussed, with details about required or suggested changes for both the TX and the RX. The feasibility of the proposed approach is demonstrated through case studies of simulating links at 100Gbps using an IBIS-AMI model on a commercial EDA tool.

s

Authors Biography

Hongtao Zhang received his Ph.D. degree in Electrical and Computer Engineering from University of California, San Diego in 2006. He joined Xilinx in 2013 and is now a senior staff Design Engineer, working on SerDes architecture development and circuit design. From 2010 to 2013, he was with SerDes design team at Oracle Corporation, where he worked on circuit design and architecture modeling. Prior to that, he worked on SerDes characterization at Texas Instruments, Dallas. His current interests are SerDes architecture development and modeling, high speed mixed-signal circuit design and optimization, and system level modeling.

Fangyi Rao is a master R&D engineer at Keysight Technologies. He received his Ph.D. degree in theoretical physics from Northwestern University. He joined Agilent/Keysight EEsoft in 2006 and works on Analog/RF and SI simulation technologies in ADS. From 2003 to 2006 he was with Cadence Design Systems, where he developed SpectreRF Harmonic Balance technology and perturbation analysis of nonlinear circuits. Prior to 2003 he worked in the areas of EM simulation, nonlinear device modeling, and medical imaging.

Timothy De Keulenaer is the CEO of BiFAST, an imec spin-off providing an electrical 112Gbps single-lane duobinary transceiver. He received the master degree in Applied Electrical Engineering from Ghent University, Belgium in 2010. From then on he has been working at the INTEC Design laboratory which is part of the Department of Information Technology at Ghent University. He received the PhD degree in Applied Electrical Engineering in 2015. His main interests are on high speed integrated circuit design and signal integrity aspects for backplane communication. He was awarded the Best Paper Award at DesignCon 2015 in the High-Speed Signal Design category and his PhD dissertation was recognized for its technological contributions receiving the Nokia Bell Labs prize in 2016.

Ken Ly received a Master of Science degree in electrical engineering in 1996 from San Jose State University. He joined Cisco right after college and had been through several job titles, which includes board design and system design. His current role at Cisco is signal integrity engineer in charge of many functions of system development including future proof design and development. He is currently looking into future SerDes architecture, design and implementation.

Ramses Pierco is the Principal Engineer at BiFAST, an IMEC spin-off startup providing an electrical 112Gbps single-lane duobinary transceiver. He received the master degree in Applied Electrical Engineering from Ghent University, Belgium in 2010. From then on he has been working at the INTEC Design laboratory, which is part of the Department of Information Technology at Ghent University. He received the PhD degree in Applied Electrical Engineering in 2015. His main interests are on analog high speed and mm Wave integrated circuit design. He received the IMEC award for Best Master Thesis in the Engineering Sciences in 2010.

Geoff Zhang received his Ph.D. in 1997 in microwave engineering and signal processing from Iowa State University, Ames, Iowa. He joined Xilinx Inc. in June, 2013. Geoff is currently Distinguished Engineer and Supervisor, in transceiver architecture and modeling under SerDes Technology Group. Prior to joining Xilinx he has employment experiences with HiSilicon, Huawei Technologies, LSI, Agere Systems, Lucent Technologies, and Texas Instruments. His current interest is in transceiver architecture modeling and system level end-to-end simulation, both electrical and optical.

1. Introduction

This paper introduces IBIS-AMI (Input/Output Buffer Information Specification Algorithmic Model Interface) modeling and simulation of duobinary signaling for high speed serial link systems. As data rate goes beyond 50Gbps, duobinary signaling starts to show its unique performance characteristics, especially for legacy backplane applications, in direct competition with NRZ and PAM4. While system engineers have been designing systems using NRZ for more than a decade, few have experience with duobinary signaling. This makes system simulation indispensable in duobinary link design, budgeting and optimization.

To accurately simulate a serial link, it is critical to model the passive link channel and SerDes (or other active component, such as repeaters) accurately, besides setting up system parameters such as data pattern, data rate, jitter, noise, etc. Inside the SerDes block one needs to model the behavior of Feed-Forward Equalizer (FFE), Continuous Time Linear Equalizer (CTLE), Decision Feedback Equalizer (DFE) and Clock Data Recovery (CDR), both implementation and adaptation algorithms. Impairments and non-idealities will also have to be represented inside the model. However, such information is typically proprietary to SerDes vendors and unavailable to the user of the model. This poses a challenge to the system level simulation.

Another challenge is model interoperability. As a third-party SerDes IP is usually required in many application scenarios, there is a need to establish a common interface standard for interoperability simulation. Still another challenge is the simulation speed. As most design specs are defined at BER of $1e-12$ or lower, designers need to run millions of bits in order to predict a link performance with good statistical confidence.

Above challenges are addressed by the IBIS-AMI standard. By defining a common SerDes model interface, the standard allows SerDes vendors to encapsulate SerDes behaviors in model executables without exposing their IP. The models can be used by system designers to perform end-to-end link simulations. Furthermore, in AMI simulations analog channels are assumed to be LTI, which can be represented by impulse responses. The highly efficient convolution and FFT method can be applied to calculating signal waveforms. As a result, millions of bits can be simulated in minutes, allowing accurate predictions of link performance at low BER.

In this paper we propose an extension of AMI for duobinary modeling and simulation. First, the standard AMI interface is reviewed. The current AMI standard only supports NRZ and PAM4 signaling. To support duobinary both the EDA and the AMI interface have to be enhanced. Specifically, on the TX side, the input waveform to the TX DLL is a NRZ signal. The input NRZ bit sequence is precoded by the EDA tool assuming precoding is required. The TX DLL adjusts the output signal swing, adds the effect of TX bandwidth impact, nonlinearity effect, and applies the necessary TX side de-emphasis such that the final received signal on the RX is duobinary.

On the RX side, the RX DLL's output waveform is a three-level duobinary signal, $\{-1, 0, +1\}$ (or $\{0, 1, 2\}$ as we have used both notations interchangeably throughout the paper), which is different from either NRZ or PAM4. Besides the output waveform and clock ticks as we are familiar with, the *AMI_GetWave* function also returns instantaneous slicer reference values and sampling time skews between slicers. They are used by the simulator to align the RX output signal and to apply de-precoding/demodulation when calculating eye diagram and raw BER.

A case study running a generated duobinary IBIS-AMI model based on the approaches illustrated in the paper in a commercial EDA simulator is presented. Through the example the feasibility and interoperability of the proposed approach is demonstrated, with details of modification, the simulation flow, and BER estimation process.

It is noted that it is not the intension of this paper to compare the performance between duobinary and NRZ or PAM4, or any other signaling schemes. The intension of the paper is to enable the capability of IBIS-AMI modeling and simulation with duobinary signaling.

2. IBIS-AMI Modeling for NRZ Signaling

Before we discuss AMI modeling for duobinary signaling, a brief review of AMI modeling of NRZ is presented in this section.

AMI defines the SerDes behavioral modeling interface and an efficient channel simulation methodology. A serial link consists of a TX, a physical channel and a RX. Each SerDes device (TX or RX) is represented by an IBIS-AMI model, which contains analog and algorithmic portions. The analog portion is a regular IBIS model, and the algorithmic part is a Dynamic Link Library (DLL) executable of a data flow model.

In a typical TX model, the analog portion models rise and fall waveforms and the output impedance and the DLL the pre-emphasis/de-emphasis. In a typical RX model, the analog portion represents the input termination and the DLL the functionalities of AGC, equalization (such as CTLE, FFE and DFE) and CDR. TX DLL output is considered an ideal voltage source, and the RX DLL input is assumed to have high impedance. Therefore, DLLs are electrically decoupled from the analog channel, which includes the TX analog model, the physical channel and the RX analog model. Furthermore, the analog channel is assumed to be LTI, thus can be represented by a combined analog channel impulse response. A graphical representation is given in Figure 1.

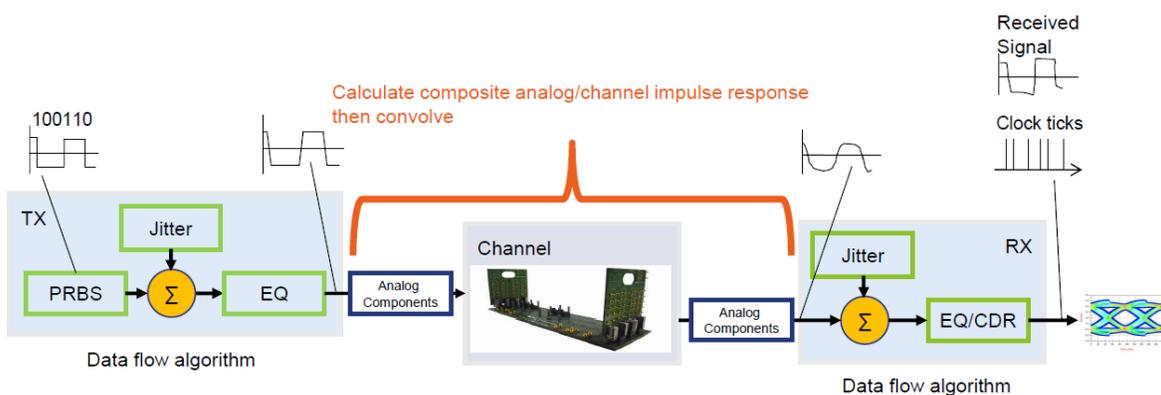


Figure 1. Graphical representation of IBIS-AMI modeling.

In AMI simulations, the TX DLL input is a square wave switching between 0.5V and -0.5V that represents the data pattern. The TX output is convolved with the analog channel impulse response. The highly efficient Fast Fourier Transform (FFT) algorithm can be employed in the convolution calculation. The resulting signal is the input to the RX DLL, which applies

equalizations and CDR to it in bit-by-bit fashion and returns the equalized signal and the recovered clocks. The RX output is sampled at each clock time and compared with the reference voltage at 0V and the transmitted bit to compute the BER. If the RX DLL has the AMI_GetWave function, the RX signal processing is performed inside the function. In a typical setting, the RX input waveform is divided into segments. The simulator repeatedly calls *AMI_GetWave*, using sequentially the waveform of each segment as the input of each function call until all segments are processed.

The above procedure is maintained while proposed modifications are added to duobinary signal modeling and simulation in the IBIS-AMI environment.

3. Introduction to Duobinary Signaling

3.1 Duobinary basics

The duobinary signaling format is an alternative to NRZ signaling, first proposed by Lender in 1963 [1], which limits the needed bandwidth in order to achieve higher serial data rates. As with PAM4 signaling this is done by trading off between signal-to-noise ratio (SNR) and bandwidth by going to a multi-level signaling. However, in case of duobinary only three signal levels are required to achieve the same power spectral density (PSD) of a PAM4 signal for the same data rate. In other words, there is a better SNR to bandwidth trade-off in case of duobinary as shown in Figure 2. This has allowed the successful demonstration of both 56Gbps and 100Gbps duobinary across backplane channels and twin-axial cable [2] [3].

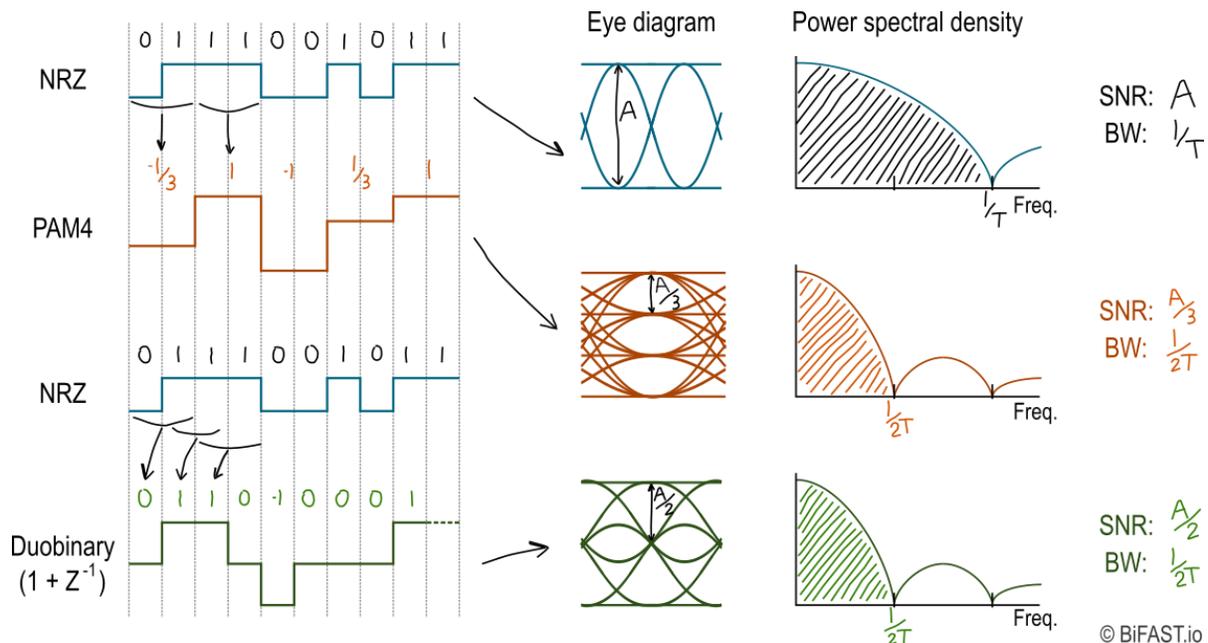


Figure 2: Eye diagram and PSD for an NRZ, PAM4 and duobinary signal.

In order to get a duobinary signal the consecutive bits of an NRZ stream are added together (corresponding to the function $1+z^{-1}$) which gives a symbol rate equal to the NRZ bit rate. By doing this high speed signal transitions are avoided as it is impossible to go directly from a +1 to a -1 symbol or vice versa. It is this attribute of a duobinary signal which gives the factor two reduction in the required bandwidth.

Achieving a duobinary signal can be done in two different ways. In one approach, the duobinary stream can be created at the transmitter by putting an NRZ stream through a delay-and-add block. In another, a low-pass filter can be applied to an NRZ stream which adds the right amount of inter-symbol interference (ISI) similar to that created by $(1+z^{-1})$. As shown in Figure 3 the wanted low-pass filter is created by combining the channel response with some equalization at both the TX and the RX side.

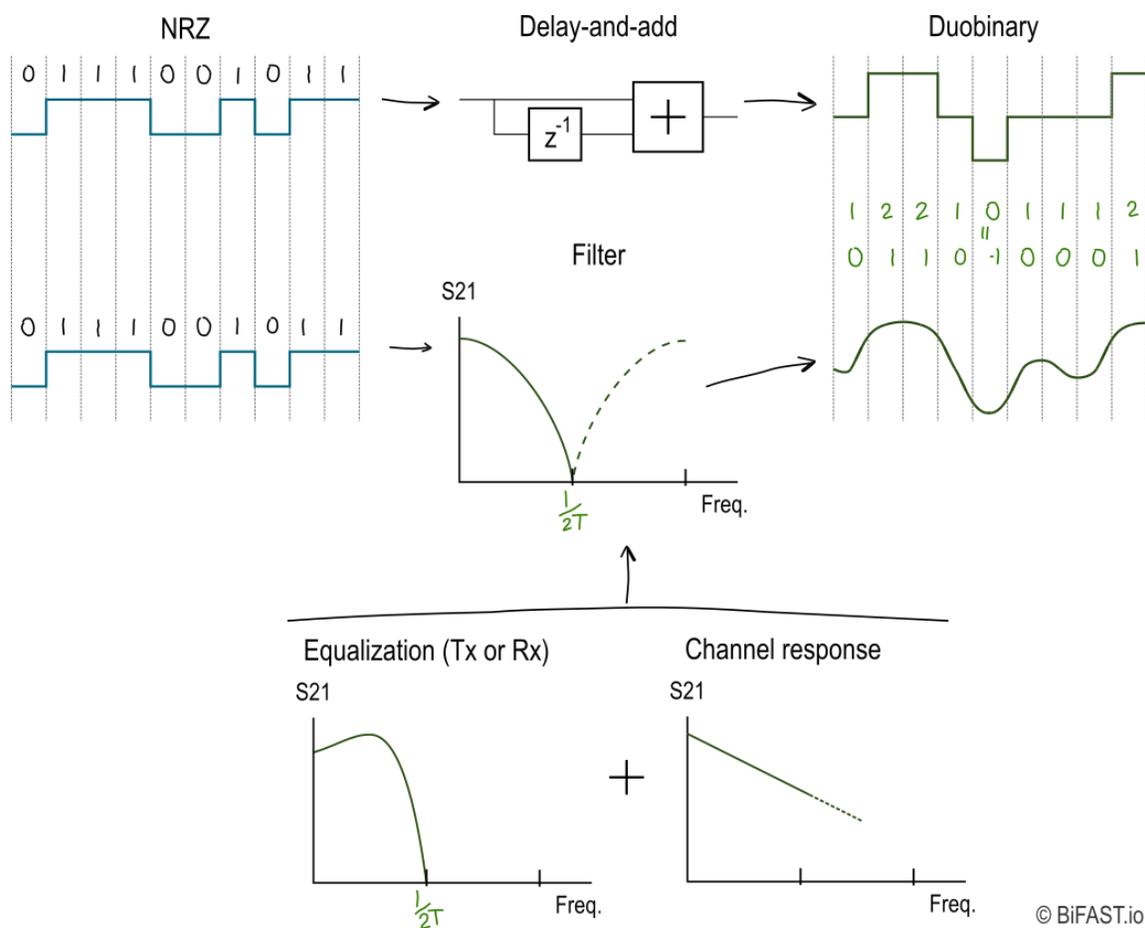


Figure 3: Duobinary signal formation from binary by a delay-and-add block or by filtering.

By choosing the second method, i.e., creating the $1+z^{-1}$ function by means of leveraging channel insertion loss, a certain portion of the channel loss does not need to be compensated. This is significant in that less equalization is required for a given channel. In addition, without using the $1+z^{-1}$ block allows the direct use an NRZ transmitter so that backwards compatibility with NRZ design is achieved.

The function used to create duobinary inherently leads to error propagation as the determination of current bit requires the correct interpretation of the previous bit. This is solved in a real-life system by means of precoding at the transmitter, which also provides easy de-precoding (or demodulation) on the receive side. Implementation of precoding and demodulation of duobinary is shown in Figure 4.

Precoding is implemented by a simple XOR gate and a single bit delay at the full rate. However, precoding can also be done at the half or quarter rate before the serialization. Demodulation is easy as only two slicers followed by a XOR gate are required. To avoid the XOR operation at the highest rate the slicer outputs can also be de-serialized allowing the XOR to work at half or quarter rate [4]. More details on signal detection are provided in the next section.

In summary, in this paper for duobinary modeling,

1. $1+z^{-1}$ is not physically implemented, but is the leverage of the channel loss and TX/RX equalization.
2. Precoding and demodulation, shown in Figure 4, are assumed to be implemented to avoid error propagation. For modeling both should be done in EDA for interoperability.

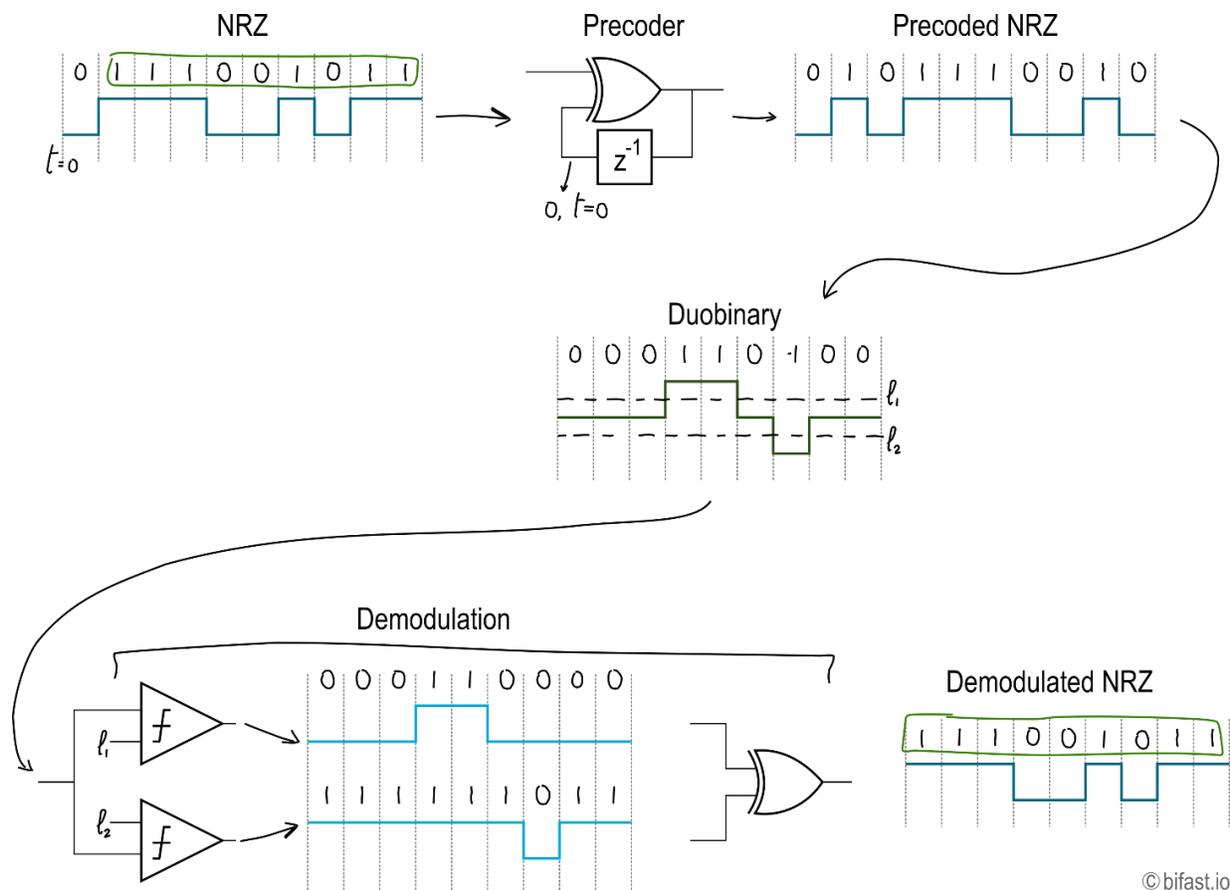


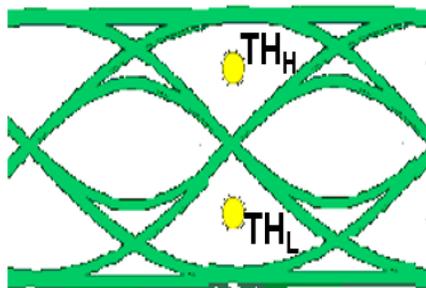
Figure 4: Possible implementation of duobinary precoder and demodulation.

3.2 Detecting duobinary signal

To reduce channel loss induced ISI and to ensure signal is sampled at the center of the eye, a duobinary SerDes employs equalization (transmitter de-emphasis, receiver equalization, such as CTLE, FFE, and DFE), and timing recovery.

The RX architecture could be analog based or ADC/DSP based. What is unique in duobinary, as shown in Figure 4 and Figure 5, are two data slicers, TH_H and TH_L , used in the RX to distinguish the signal levels. There are usually a set of error slicers or the crossing slicers for the adaptation loop to function. They are not shown in the figure.

The decision logic is also given in Figure 5, where $s(k)$ represents the input analog or digitized signal to the data slicer at the k^{th} bit and $d(k)$ the detected symbol level. Once the decision is made, duobinary signal can be decoded back into binary bits. It should be pointed out that the RX can adaptively adjust the values of the two data slicer levels, leading to time varying TH_H and TH_L . More often than not, however, TH_H and TH_L are set to be the same in value but opposite in polarity, in design to reduce design complexity and operating power.



```
if  $s(k) > TH_H$  or  $s(k) < TH_L$ 
     $d(k) = 0$ ;
else
     $d(k) = 1$ ;
endif
```

Figure 5. Illustration of duobinary signal sampling and decision logic with precoding.

It is worth pointing out that if no precoding is used on the TX side, then there is no de-precoding on the RX side. The decision logic in Figure 5 has to be modified to that shown in Figure 6. Obviously, the problem without precoding is error propagation as the current bit, the k^{th} bit, is partially dependent on the previous bit, the $(k-1)^{\text{th}}$ -bit.

```
if  $s(k) > TH_H$ 
     $d(k) = 1$ ;
elseif  $s(k) < TH_L$ 
     $d(k) = 0$ ;
else
    if  $d(k-1) = 1$ 
         $d(k) = 0$ ;
    else
         $d(k) = 1$ ;
    endif
endif
```

Figure 6. Illustration of duobinary signal sampling and decision logic without precoding.

4. IBIS-AMI Modeling for Duobinary

Certain aspects of the IBIS-AMI standard need to be modified in order to accommodate duobinary signaling. In this section all the necessary changes are explained and details of modifications are presented. It is worth noting that there could be different approaches to the modeling. What is proposed in this paper is believed to be a natural extension to the existing IBIS-AMI standards.

4.1 TX input stimulus levels

In the proposal in this paper the TX DLL input stimulus remains the same as that for NRZ. For example, the current standard for NRZ specifies that logic high is at 0.5V and logic low at -0.5V, and the same remains for duobinary. For duobinary the EDA tool is responsible for precoding the binary data stream. For the TX DLL there is no difference whether the binary bits are precoded or not, but for the RX decision, with or without it makes a big difference. The precoding algorithm is shown in detail Figure 7.

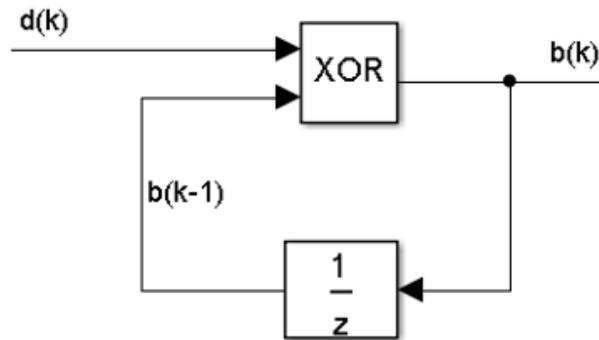


Figure 7. Data precoding block – implemented inside EDA.

An example is shown in Table 1. In general, if precoding is used, the NRZ data the TX DLL received from the EDA tool is no longer the original PRBS-n pattern for most of the n values. However, the commonly used PRBS-7 and PRBS-15 will lead to the same PRBS with only a shift.

Table 1. Example showing the relationship between $d(k)$ and $b(k)$ after precoding.

$d(k)$	x	0	0	1	0	0	1	0	1	1	1	0	1	0
$b(k)$	1	1	1	0	0	0	1	1	0	1	0	0	1	1

As described in the previous section, the TX DLL can either perform the delay-and-add operation or work with the channel to apply a low-pass filter on the input NRZ signal to generate the duobinary signal at the RX. The delay-and-add block is not implemented in this paper. This implies the equalization, jointly delivered by the TX and the RX, has less burden in terms of the amount ISI that needs to be removed.

4.2 RX slicer levels and timing skew

The proposed modification to the AMI standard for duobinary on the RX side is to add RX slicer levels to the reserved parameter list with Usage Out or InOut. Unlike in the case of NRZ, where the decision level is fixed at 0V for a differential signal, two slicer levels for a duobinary RX, TH_H and TH_L , are both output to the EDA, in addition to clock ticks. TH_H and TH_L are usually adaptive, thus they are functions of time. How CTLE, AGC, DFE, or CDR adaptation works is beyond the scope of this paper.

Duobinary decision logic must be made against the instantaneous slicer levels on a bit by bit basis. This requires the RX model to return TH_H and TH_L values in each call to the *AMI_GetWave* function. Although in theory slicer levels could differ from bit to bit, in practice they change slowly, particularly after the adaptations converge. Thus, it can be justified to return only one set of data slicers (TH_H and TH_L) for each *AMI_GetWave* call.

In IBIS 6.0 *AMI_GetWave* is defined in Figure 8. With the function signature intact, TH_H and TH_L can be returned through the *AMI_parameters_out* string. The *AMI_parameters_out* argument is a pointer to a string pointer. Memory for the string is allocated and de-allocated by the algorithmic model. The model returns a pointer to the string as the value of this argument. The content of the string is formatted as a tree structure of parameters with Usage Out and InOut. The tree structure is scalable and extendible, making it easy to add new parameters to the string.

```
long AMI_GetWave ( double *wave,  
                  long wave_size,  
                  double *clock_times,  
                  char **AMI_parameters_out,  
                  void *AMI_memory );
```

Figure 8. Definition of *AMI_GetWave*.

For optimal sampling result, the two slicers can possibly sample the signal at different times. Like slicer levels, the sampling time skew can be adjusted adaptively and time dependent. To capture the skew effect, we propose two new AMI reserved parameters, Δt_H and Δt_L , for upper and lower slicer sampling time offsets, respectively, relative to clock tick times. The slicer sample time skew is given by the difference between the two offsets. Assuming that the skew varies slowly after the adaptation converges, it is sufficient to update the offsets once in each *AMI_GetWave* call. Similar to slicer levels, the offset values are returned by the model through the *AMI_parameters_out* string argument of the *AMI_GetWave* function.

In each *AMI_GetWave*, the RX model will write name-value pairs of TH_H , TH_L , Δt_H and Δt_L into the *AMI_parameters_out* string and pass it back to the simulator, which will parse the string to extract slicer levels and sampling time offsets relative to clock tick times. Together with the offsets, clock times define the upper and lower slicer sampling times. The slicer levels are used to decide duobinary logic on bits processed in this *AMI_GetWave* call for BER calculations.

The decision logic, residing in the EDA tool, is given in Figure 5. Eye diagrams, BER contours, and voltage and timing bathtub curves can all be derived from the four sets of data, the waveforms, the clock ticks, the slicer levels and the sampling time offsets. The simulation flow is illustrated in Figure 9.

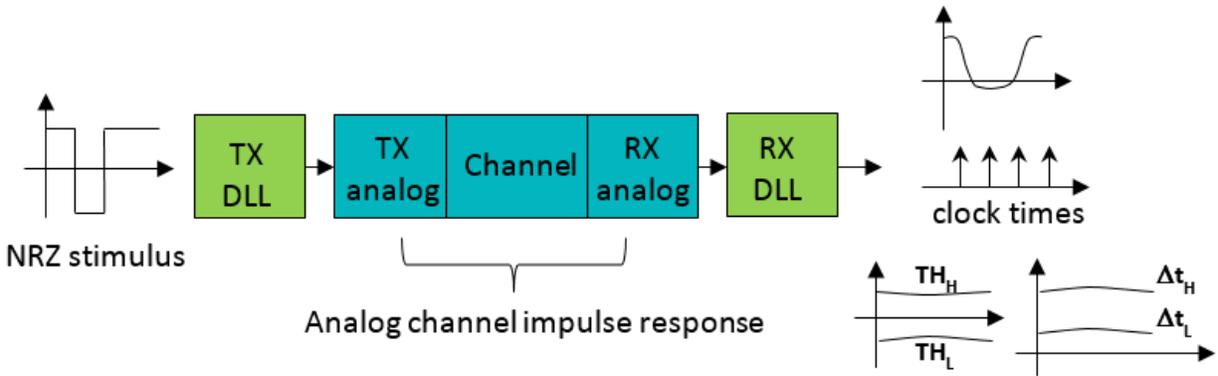


Figure 9. Duobinary link AMI simulation flow.

It should be pointed out that in the ADC-based receiver architecture the signal is sampled only once per bit. Thus, the analog waveform does not exist in the silicon. However, for the sake of AMI modeling the RX model can still generate a hypothetical analog waveform by over-sampling the signal using the same set of equalizations.

5. Duobinary Eye Measurements

5.1 Eye diagrams

The conventional eye diagram can be constructed for duobinary; there are two vertically stacked eyes in the diagram, corresponding to two decision levels. The eye diagram generation process for duobinary is identical to that for NRZ and PAM4, as shown in Figure 10. From left to right are NRZ, PAM4, and duobinary eyes centered in the plot. Hypothetical data slicers are also plotted for the three modulations as green dots. As shown in Figure 10, NRZ requires one data slicer, PAM4 requires three data slicers, while duobinary requires two data slicers.

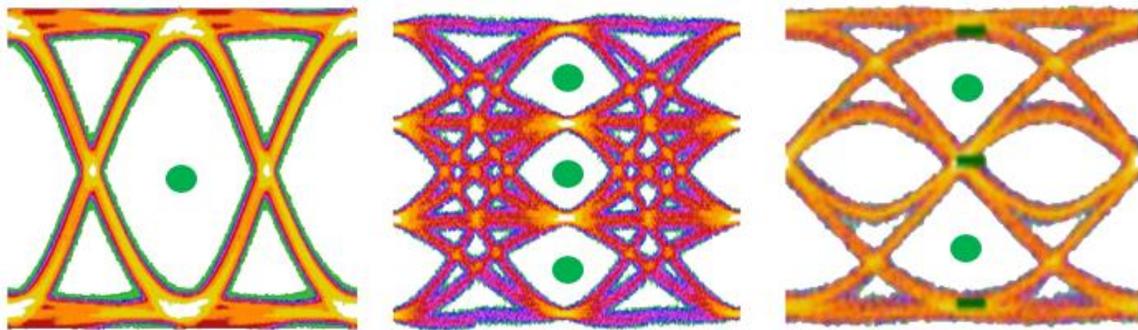


Figure 10. Eye diagrams for NRZ, PAM4, and duobinary.

5.2 Bathtub curves

For NRZ signal, timing and voltage bathtub curves are used to measure link performance. For the duobinary signal, to analyze performance at different signal levels, symbol error rate (SER) is measured at each of the two eyes, and one set of bathtub curves can be plotted for each data slicer level, giving a pair of bathtub curves in total. Upper slicer bathtubs are generated with waveform segments of expected level 1 and level 2 symbols, denoted as $v_1(t)$ and $v_2(t)$, respectively. Lower slicer bathtubs are generated with $v_0(t)$ and $v_1(t)$, where $v_0(t)$ is the waveform segment of expected level 0 symbol.

Upper slicer level $TH_H(t)$, lower slicer level $TH_L(t)$, clock times $t_{clk}(n)$, upper slicer sample time offset $\Delta t_H(n)$ and lower slicer sample time offset $\Delta t_L(n)$ are used to calculate SER. Here n is the bit index. Recall that bits processed in the same AMI_GetWave call share the same TH_H , TH_L , Δt_H and Δt_L . To capture slicer level fluctuations, the instantaneous slicer level is subtracted from RX output waveform in SER calculations. Specifically, for the upper eye, waveform segments of $v_1(t)-TH_H(t)$ and $v_2(t)-TH_H(t)$ are sampled to generate bathtubs, and the segments are centered at $t_{clk}(n)+\Delta t_H(n)+UI/2$ to track upper slicer sample time jitters, where UI is the unit interval. For the lower eye, waveform segments $v_0(t)-TH_L(t)$ and $v_1(t)-TH_L(t)$ are sampled to generate bathtubs, and the segments are centered at $t_{clk}(n)+\Delta t_L(n)+UI/2$ to track lower slicer sample time jitters. The procedure is shown in Table 2. Note that since the signal is adjusted with respect to slicer levels in BER calculations, if levels vary with time, some discrepancy between bathtubs and the conventional eye is expected.

Eye	Traces	Horizontal eye center
Upper	$v_1(t)-TH_H(t)$ and $v_2(t)-TH_H(t)$	$t_{clk}(n)+\Delta t_H(n)+UI/2$
Lower	$v_0(t)-TH_L(t)$ and $v_1(t)-TH_L(t)$	$t_{clk}(n)+\Delta t_L(n)+UI/2$

Table 2. Procedure of bathtub curve construction at each slicer level.

6. Example of Using AMI Model for Duobinary

Three channels are used for 100Gbps simulations using duobinary signaling. The three channels could represent VSR, MR, and LR, approximately, as no industry standards are defined yet for the 100G speed node. Loss profiles are plotted in Figure 11. The loss at 28GHz, the Nyquist frequency for duobinary at 100Gbps, is marked.

The simulation bench setup in Keysight ADS is shown in Figure 12. The channel is represented by cascading TX on-die, TX package, channel, RX package and RX on-die S-parameters. The data rate is 100Gbps. PRBS23 is the data pattern. Of course, after the EDA applies precoding, the data pattern input to the TX DLL is no longer PRBS23. One and a half million bits are simulated for each channel, with the first half a million bits before the RX adaptation converges ignored. No crosstalk is explicitly included for the simulations, as the exercise is more of proving the concept of modeling duobinary rather than demonstrating link performance.

The TX equalization is implemented by a simple symbol spaced 3-tap FIR, with one pre and one post cursor tap. The tap weights are manually programmed. On the RX side, there is a 3-stage CTLE and an AGC, plus 20-tap DFE. Baud CDR is implemented. A baseline wander cancellation block is also implemented. All the RX side parameters are adaptively tuned to achieve highest SNR based on proprietary algorithms.

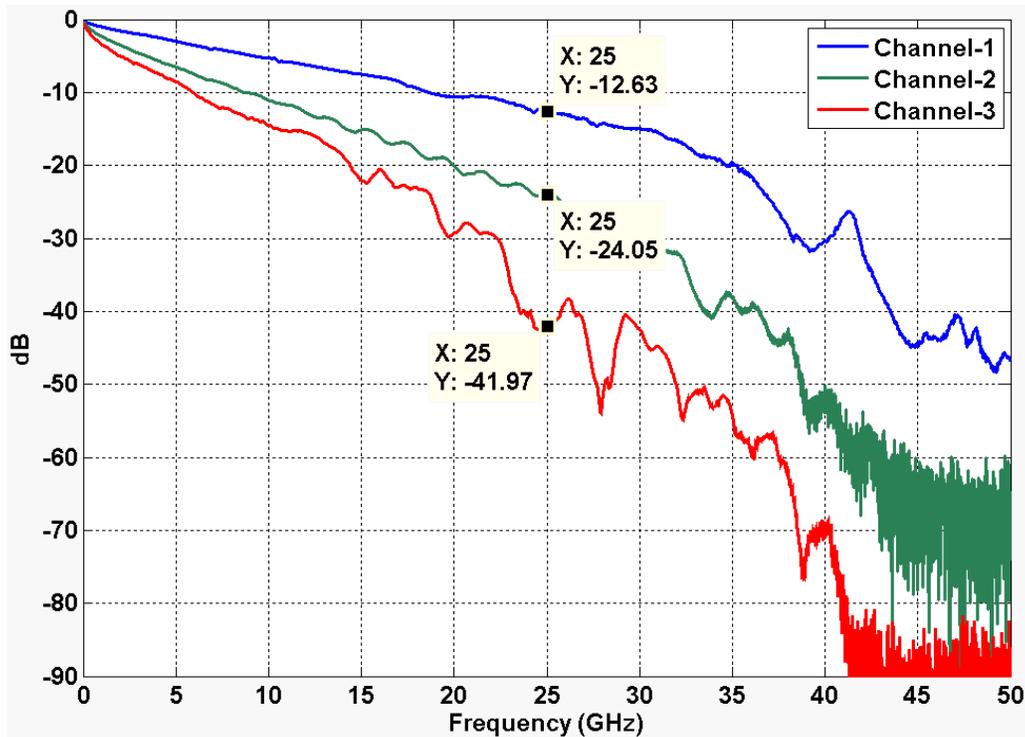


Figure 11. Three channel insertion loss profiles.

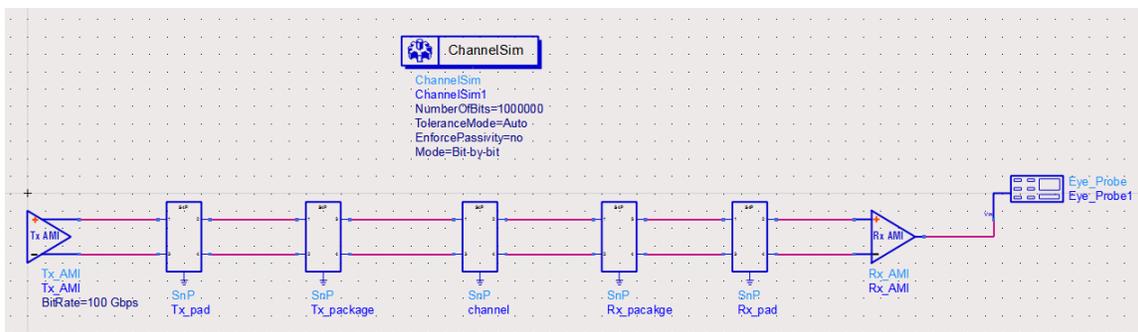
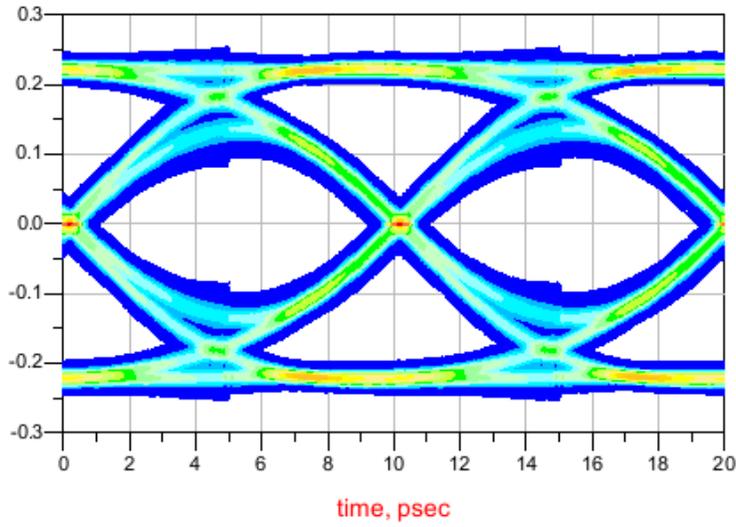
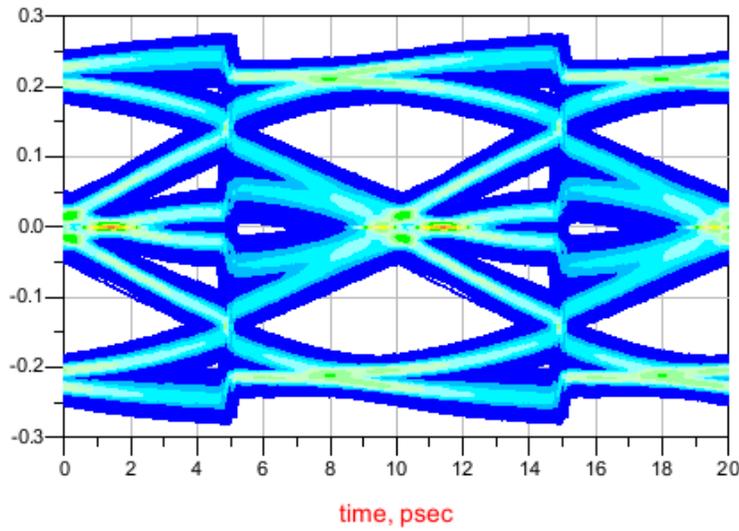


Figure 12. Simulation bench setup.

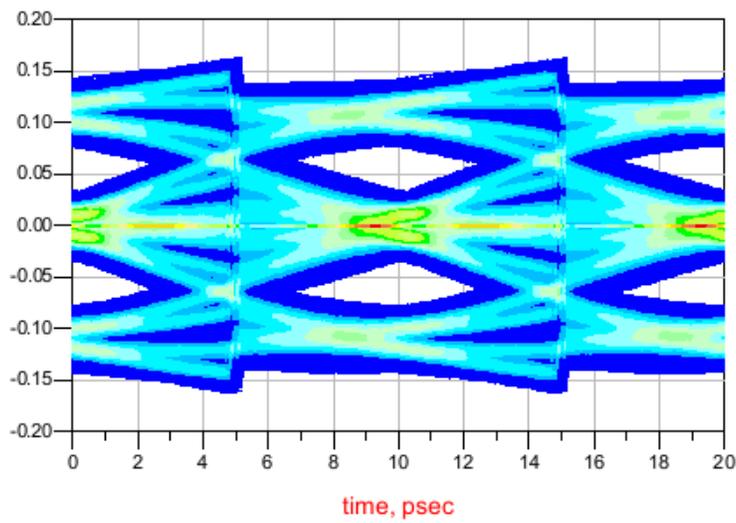
Figure 13 shows the eye diagrams from the three channels. They are processed by the EDA based on the waveforms and clock ticks exported from the RX DLL. Figure 14 shows the RX equalized output waveform with an over-sample rate of 64 in channel 2. It is worth noting that in the VSR channel, without delay-and-add operation in the TX, both the eye and the waveform resemble those of an NRZ signal, which is as expected. It implies that the same transmit equalization scheme can be used to transmit both NRZ and duobinary signals, allowing designers to choose between NRZ and duobinary receivers.



(a) Channel 1 (VSR)

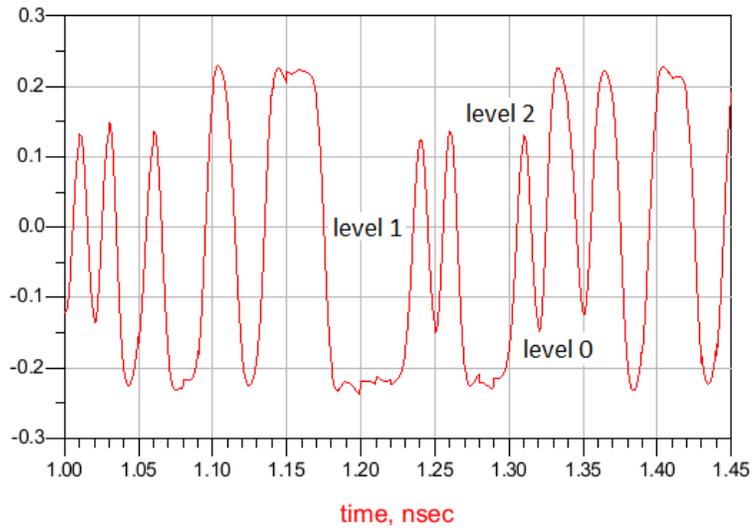


(b) Channel 2 (MR)

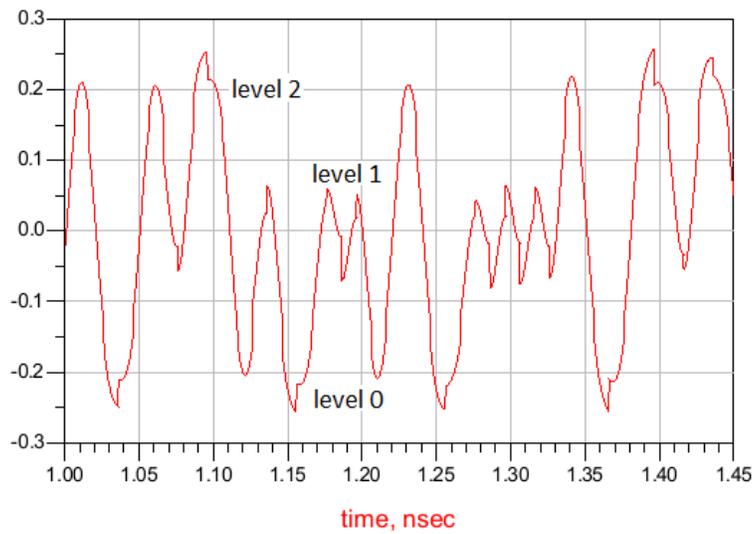


(c) Channel 3 (LR)

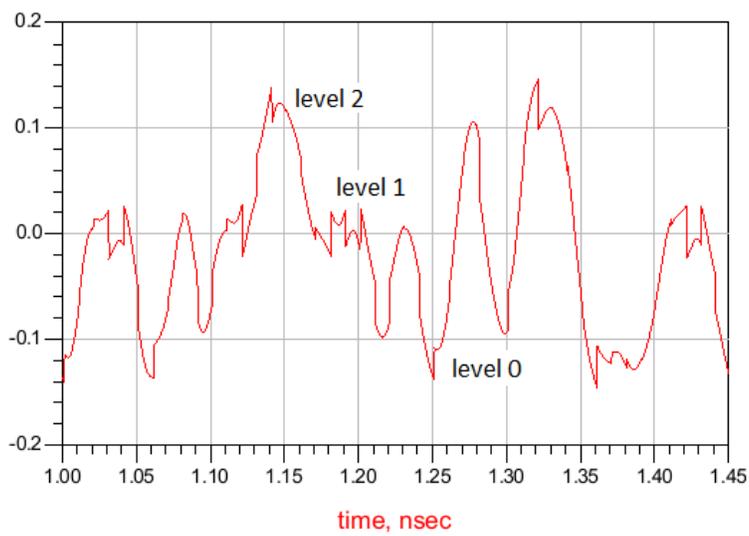
Figure 13. Duobinary eye diagrams for the three channels.



(a) Channel 1 (VSR)



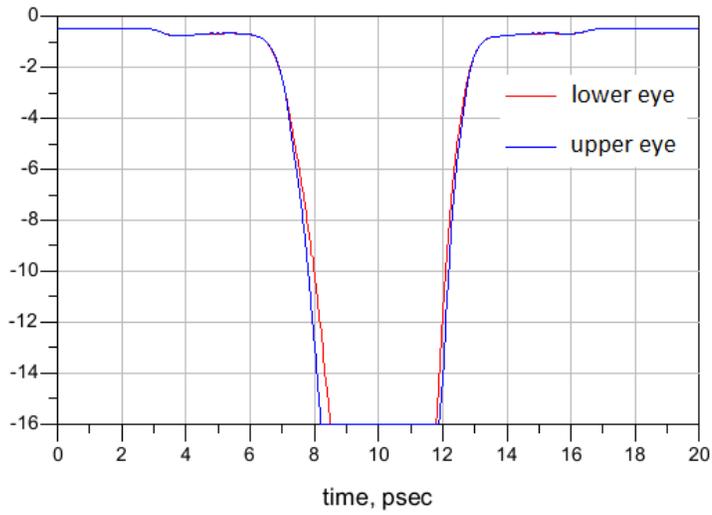
(b) Channel 2 (MR)



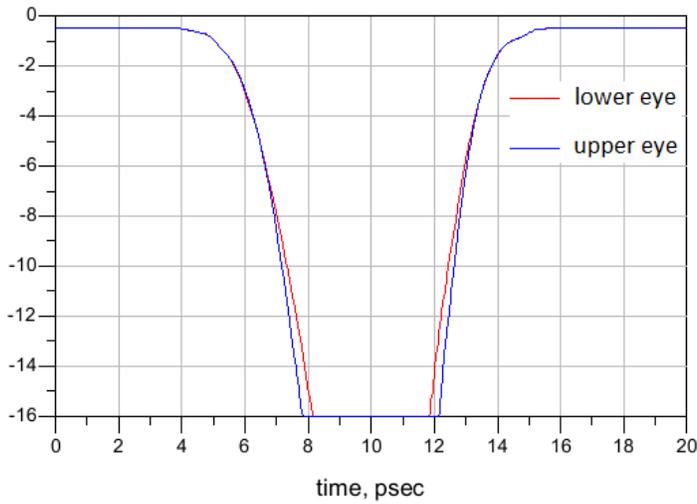
(c) Channel 3 (LR)

Figure 14. RX equalized output waveforms of the three channels.

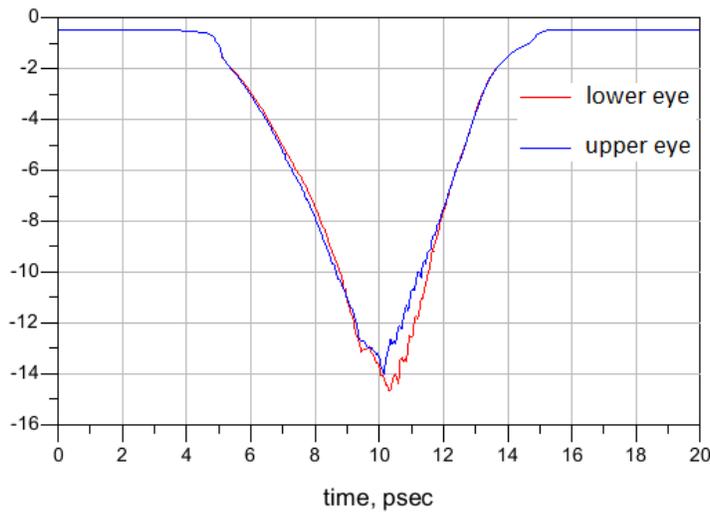
Timing and voltage bathtub curves for upper and lower eyes are plotted in Figures 15 and 16.



(a) Channel 1 (VSR)

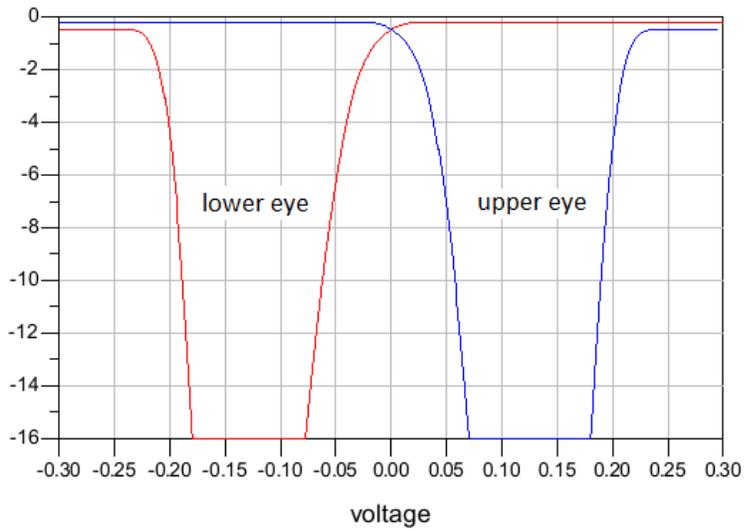


(b) Channel 2 (MR)

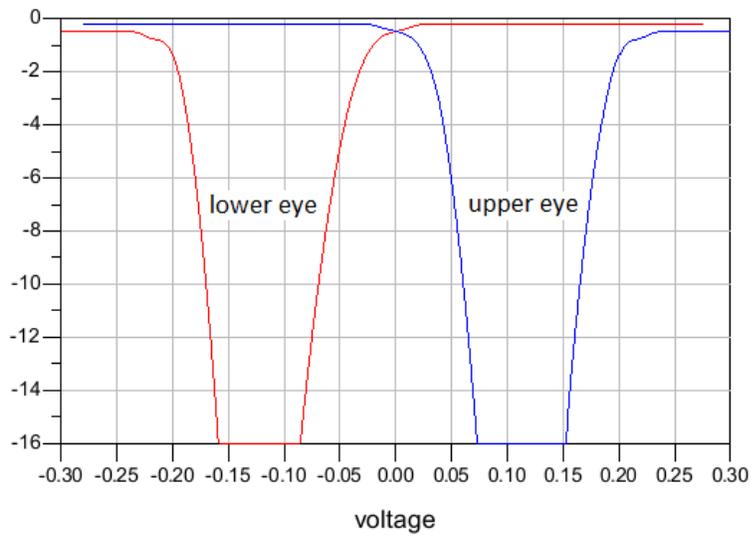


(c) Channel 3 (LR)

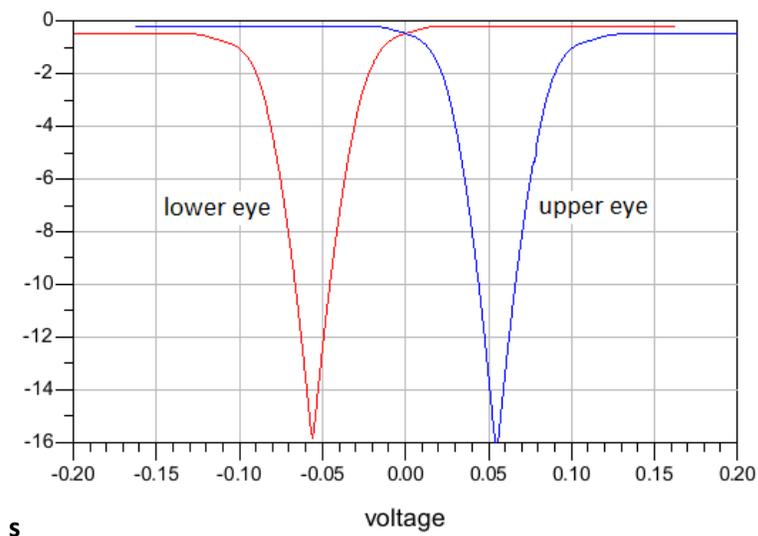
Figure 15. Timing bathtub curves of the three channels.



(a) Channel 1 (VSR)



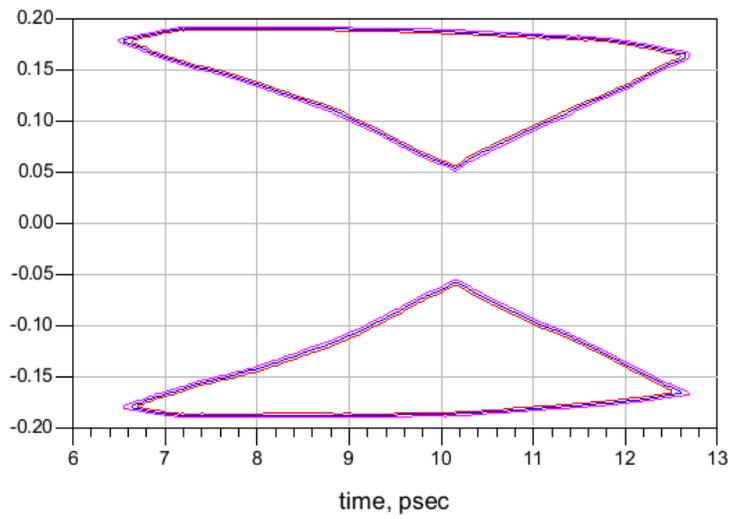
(b) Channel 2 (MR)



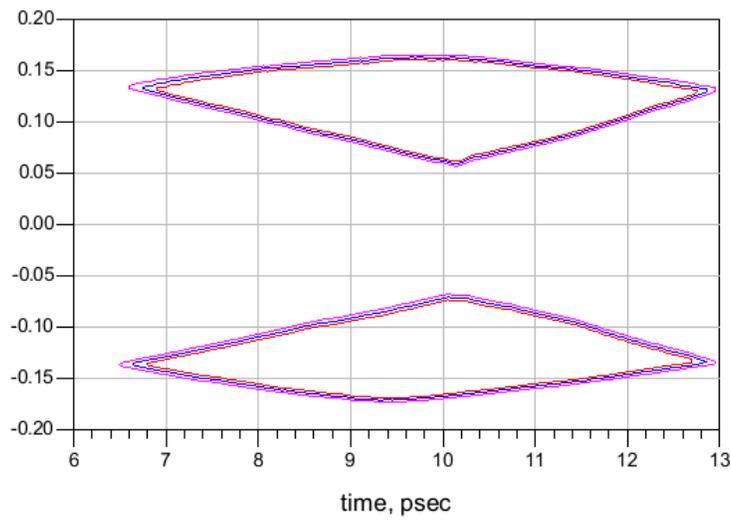
(c) Channel 3 (LR)

Figure 16. Voltage bathtub curves of the three channels.

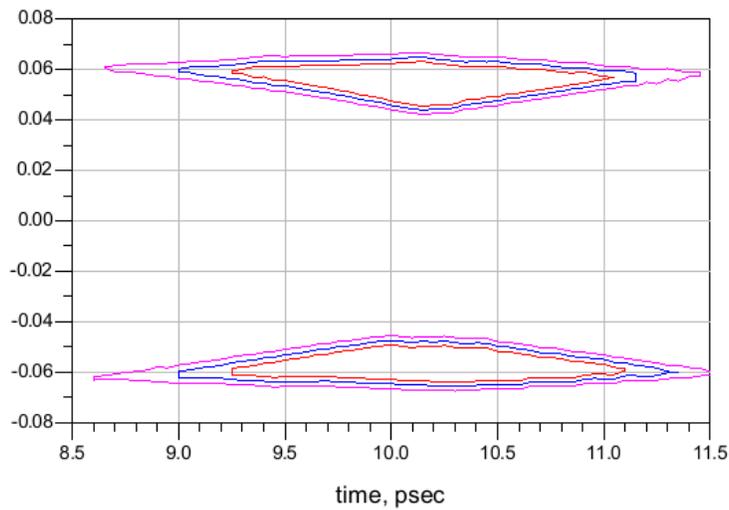
Contours at SER $1e-10$, $1e-11$ and $1e-12$ of upper and lower eyes are shown in Figure 17.



(a) Channel 1 (VSR)



(b) Channel 2 (MR)



(c) Channel 3 (LR)

Figure 17. Contours at SER $1e-10$, $1e-11$ and $1e-12$ of the three channels.

Figure 18 shows the data slicer and CDR phase adaptation profiles. The VSR, MR, and LR channels are plotted in (a), (b), and (c), respectively. The results indicate that both slicer and CDR adaptations converge after the first 100K bits in all three cases. Notice that in this RX model TH_H and TH_L are symmetrical with $TH_H = -TH_L$, as shown in the plots.

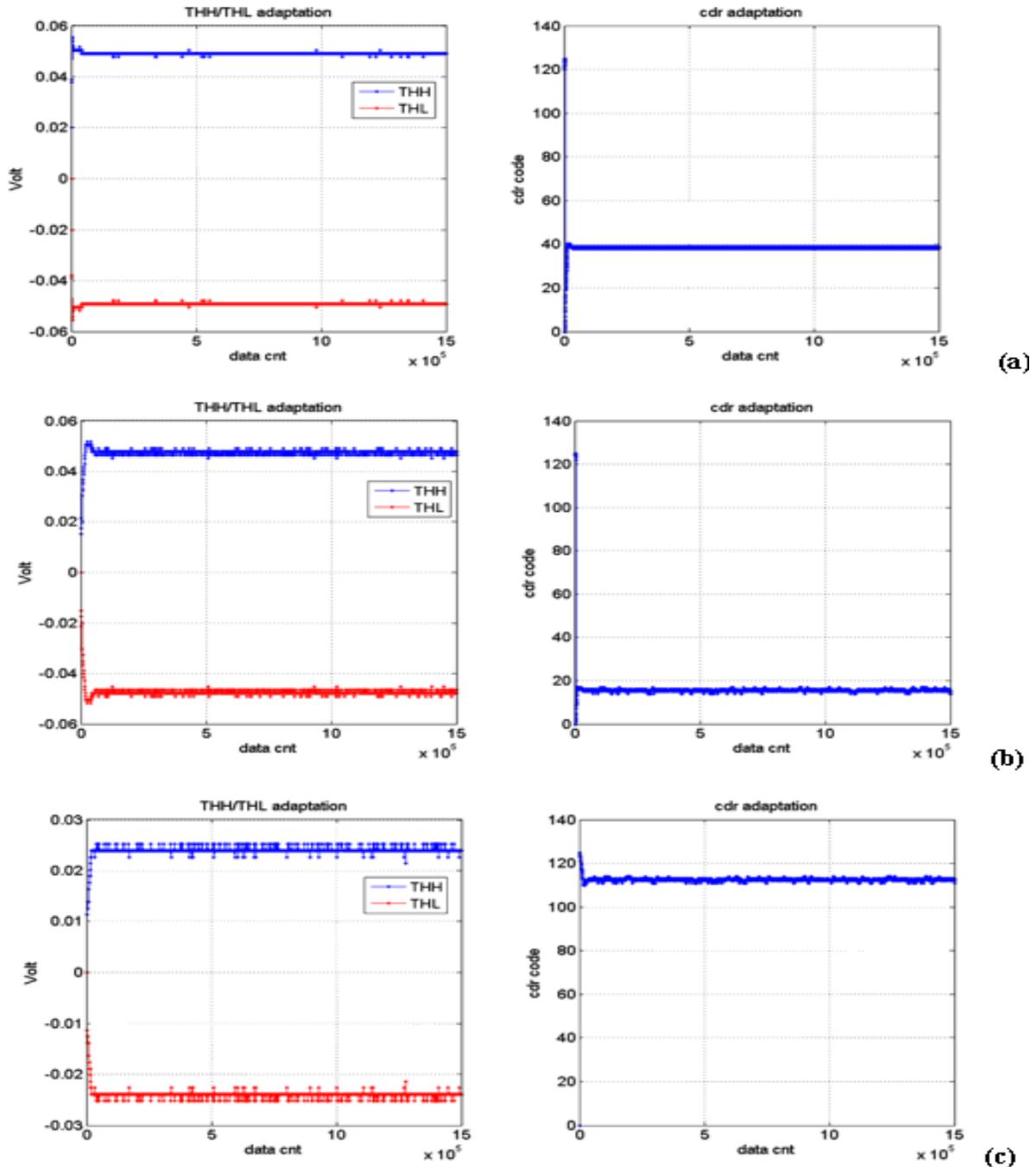


Figure 18. Data slicer and CDR adaptation profiles of (a) VSR, (b) MR, and (c) LR channels.

7. Conclusions and Future Work

In this paper, an extension to the IBIS AMI standard to include duobinary signal modeling and simulation is proposed. Specifically, the input waveform to the TX DLL remains an NRZ signal, although it is precoded from the source NRZ pattern. That is to say in the context of this paper precoding is always assumed. The TX DLL works exactly the same as an NRZ TX DLL model. The RX output waveform is a three-level duobinary signal. RX needs to send out two data slicer levels, two slicer sampling time offsets plus clock times to the EDA tool using parameter *AMI_parameters_out* through *AMI_GetWave* function calls. The RX output signal can be demodulated in a straightforward manner by the EDA tool for the overall system BER performance estimation.

The proposal is verified through simulations of duobinary signaling on three channels with low, medium and high losses using a modified ADS platform. Results of simulated waveforms and eye diagrams are presented. Two vertical bathtub curves, two horizontal bathtub curves, and two sets of SER contours are calculated for upper and lower eyes to fully evaluate the performance margin of the link system.

In the proposed approach slicer levels and sampling time offsets are updated by the RX DLL once per *AMI_GetWave* call. This is justified as a stable and settled reference adaptation loop is not expected to deviate much from bit to bit. If an unstable reference adaptation loop is expected or if the detailed reference adaptation settling behavior is to be observed, the AMI model would need to pass on the entire adaptation sequence to the EDA tool, which cannot be supported using the existing IBIS-AMI standard.

To support NRZ, PAM4, and duobinary signaling schemes at the same time, an EDA tool should be able to switch between the three modulation modes and can automatically pass the information to the IBIS-AMI model. To calculate true SER/BER, the received and decoded signal is correlated with the transmitted signal. The EDA tool should also have options for including or excluding precoding/de-precoding blocks. The decision algorithm is different accordingly and is taken care of on the EDA tool side.

References

- [1] A. Lender, "The duo-binary technique for high-speed data transmission", IEEE Trans. Communications Electronics, vol. 82, no. 2, pp. 214-218, May 1963
- [2] Timothy De Keulenaer, et al, "56+ Gb/s Serial Transmission using Duobinary Signaling", DesignCon 2015
- [3] Joris Van Kerrebrouck, et al, "100 Gb/s Serial Transmission over Copper using Duobinary Signaling", DesignCon 2016
- [4] Jeffrey Sinsky, et al, "Circuitry and method for multi-level signals", US 9369317 B2
- [5] Hongtao Zhang, et al, "IBIS-AMI Modeling and Simulation of 56G PAM4 Link Systems", DesignCon 2015