\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BIRD ID#: 123.2

ISSUE TITLE: IBIS-AMI New Reserved Parameters for Jitter/Noise

AUTHOR: Walter Katz, Mike Steinberger, Todd Westerhoff, SiSoft

DATE SUBMITTED: October 20, 2010

DATE REVISED: April 14, 2011

DATE ACCEPTED BY IBIS OPEN FORUM:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STATEMENT OF THE ISSUE:

Model developers and EDA vendors building IBIS-AMI models using the IBIS 5.0 specification have come across a number of modeling issues that are not addressed in IBIS 5.0. In order to deliver models and EDA tools that meet end-user demands for model accuracy and functionality, EDA vendors have defined "extensions" to add new capabilities to IBIS-AMI models. Unfortunately, EDA vendors have had to use proprietary (and different) syntax to add these capabilities to models, limiting model portability between different EDA tools.

This BIRD proposes new syntax for the .ami control file that improves model functionality and accuracy. Including this syntax in the IBIS standard will allow creation of accurate, compliant IBIS-AMI models that are readily portable between commercial EDA simulators.

The parameters defined in this document are to be added in Section 6c of the

IBIS 5.0 specification as new Reserved\_Parameters.

Jitter, Noise and Clock Modeling

Tx\_Rj, Tx\_Sj, Tx\_Sj\_frequency, Rx\_Clock\_Recovery\_Mean, Rx\_Clock\_Recovery\_Rj,

Rx\_Clock\_Recovery\_Sj, Rx\_Clock\_Recovery\_DCD, Rx\_Rj, Rx\_Sj, Rx\_DCD, Rx\_Noise,

The following parameter exists in the IBIS 5.0 specification but its definition is replaced using the text in this BIRD:

Tx\_DCD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

On page 146 replace:

| Tx\_DCD:

|

| Tx\_DCD (Transmit Duty Cycle Distortion) can be of Usage Info

| and Out. It can be of Type Float and UI and can have Data

| Format of Value, Range and Corner. It tells the EDA platform

| the maximum percentage deviation of the duration of a

| transmitted pulse from the nominal pulse width. Example of

| TX\_DCD declaration is:

|

| (Tx\_DCD (Usage Info)(Type Float)

| (Format Range <typ> <min> <max>))

with:

| Tx\_DCD:

|

| Tx\_DCD (Transmit Duty Cycle Distortion) can be of Usage Info

| or Out. It can be of Type Float and UI and can have Data

| Format of Value, Range and Corner. It defines half the peak

| to peak clock duty cycle distortion, in seconds or UI, to be

| added to the behavior implemented directly by the transmitter

| model.

|

| Example of TX\_DCD declaration is:

|

| (Tx\_DCD (Usage Info)(Corner 0.008 0.016 0.005)(Type UI)

| (Description "TX Duty Cycle Distortion in UI.")

| )

Time(n)=n\*bit\_time+Tx\_DCD\*(-1)n

Time(n) is the time of the nth possible transition.

Note that all equations using jitter parameters that can be defined as UI shall be assumed to seconds in these formulae.

The following text is added immediately before Table 1 on page 148: Jitter, Noise and Clock Parameters

The following optional Reserved Parameters are used to specify impairments for the transmitter output. These budgets specify the impairment as measured at the TX output (i.e. the transmitter output is expected to be directly modulated by these amounts). This data is used by the simulator to either modify the input presented to the algorithmic model or when post-processing the results from the model; the budget values specified by these parameters are not passed directly to the model itself.

"Tx\_Rj" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines the standard deviation, in seconds or UI, of an uncorrelated Gaussian phase noise process at the transmitter which is to be added to the behavior implemented directly by the transmitter model. \*

Example:

 (Tx\_Rj (Usage Info)(Corner 0.005 0.006 0.004)(Type UI)

 (Description "TX Random Jitter in UI.")

Time(n)=n\*bit\_time+Tx\_Rj\*gaussian\_rand()

gaussian\_rand() is a function that returns floating point numbers between –inf and \_inf. The distribution of these numbers shall be an uncorrelated Gaussian distribution centered at zero with a Sigma of 1.

"Tx\_Sj" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines half the peak to peak amplitude, in seconds or UI, of a sinusoidal jitter which is to be added to the behavior implemented directly by the transmitter model.

Example:

 (Tx\_Sj (Usage Info)(Corner 0.05 0.07 0.4)(Type UI)

 (Description "TX Sinusoidal Jitter in UI.")

)

"Tx\_Sj\_frequency" is an AMI parameter of Type Float and Usage either Info or Out which defines the frequency, in Hertz, of the sinusoidal jitter at the transmitter.

Example:

 (Tx\_Sj\_Frequency (Usage Info)(Corner 6.5E7 6.5E7 6.5E7)(Type UI)

 (Description "TX Sinusoidal Jitter Frequency in Hz.")

)

Time(n)=n\*bit\_time+Tx\_Sj\*sin((n\*bit\_time\*2\*Pi)\*Tx\_Sj\_Frequency)

The following optional Reserved Parameters are used to specify characteristics of the receiver’s recovered clock when the model does not return clock\_ticks information from an AMI\_Getwave call. This data is used by the simulator when post-processing the results from the model; the budget values specified by these parameters are not passed directly to the model itself.

"Rx\_Clock\_Recovery\_Mean" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines a static offset, in seconds or UI, *between the recovered clock and the median threshold crossing time* \* in the eye diagram plus one half bit period.

Example:

 (Rx\_Clock\_Recovery\_Mean (Usage Info)(Value 0.05)

 (Type UI)(Description "Recovered Clock offset in UI.")

)

actual\_time=ideal\_time+Rx\_Clock\_Recovery\_Mean

ideal\_time is determined by EDA tool from eye generated from

impulse response output of Rx\_Init

"Rx\_Clock\_Recovery\_Rj" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines the standard deviation, in seconds or UI, of a Gaussian phase noise exhibited by the recovered clock.

Example:

 (Rx\_Clock\_Recovery\_Rj (Usage Info)(Corner 0.005 0.006 0.004)

 (Type UI)(Description "RX Random Clock Jitter in UI.")

)

actual\_time=ideal\_time+Rx\_Clock\_Recovery\_Rj\*gaussian\_rand()

"Rx\_Clock\_Recovery\_Sj" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines half the peak to peak variation, in seconds or UI, of a sinusoidal phase noise exhibited by the recovered clock.

Example:

 (Rx\_Clock\_Recovery\_Sj (Usage Info)(Corner 0.05 0.07 0.4)(Type UI)

 (Description "RX Sinusoidal Jitter in UI."))

The following distribution shall be used if the Rx\_Sj\_Frequency is not defined.

clock\_times(n)=clock\_times(n)+Rx\_Clock\_Recovery\_Sj\*sin(Pi\*rand())

rand()is a function that returns floating point numbers between –.5 and +.5. The distribution of these numbers shall be an uncorrelated uniform distribution between -.5 and .5.

Jitter PDF(dt) = 1/sqrt(1-(dt/Rx\_Sj)^2)

 (Note: Dual-Dirac)

I I I I

 I I I I

 I I I I

 I I I I

 I--------------I

I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

 I

I

"Rx\_Clock\_Recovery\_Sj\_frequency" is an AMI parameter of Type Float and Usage either Info or Out which defines the frequency, in Hertz, of the sinusoidal phase noise driven by impairments external to the receiver.

Example:

 (Rx\_Clock\_Recovery\_Sj\_frequency (Usage Info)

 (Corner 6.5E7 6.5E7 6.5E7)(Type Float)

 (Description "Rx Clock Recovery Sinusoidal Jitter Frequency in Hz.")

)

 clock\_times(n)=clock\_times(n)+Rx\_Sj\*sin(clock\_times(n)\*2\*Pi\*Rx\_Sj\_**Frequency**)

"Rx\_Clock\_Recovery\_DCD" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines half the peak to peak variation, in seconds or UI, of a clock duty cycle distortion exhibited by the recovered clock.

Example:

 (Rx\_Clock\_Recovery\_DCD (Usage Info)(Corner 0.008 0.016 0.005)

 (Type UI)(Description "RX Duty Cycle Distortion in UI.")

)

actual\_time=ideal\_time+Rx\_Clock\_Recovery\_DCD\*(-1)n

The following optional Reserved Parameters are used to modify the statistics associated with receiver’s recovered clock when the model returns clock ticks information from an AMI\_Getwave call. This data is used by the simulator when post-processing the results from the model; the budget values specified by these parameters are not passed directly to the model itself. Rx\_Rj, Rx\_DCD, Rx\_Sj and Rx\_Sj\_Frequency represent the jitter associated with the CDR reference Clock. These impairments external to the receiver that are input to the RX CDR, but are not included in the CDR clock\_times output. This number can represent either the reference clock jitter, or a budgeted reference clock jitter.

"Rx\_Rj" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines the standard deviation, in seconds or UI, of a Gaussian phase noise driven by impairments external to the receiver that are input to the RX CDR, but are not included in the CDR clock\_times output. This phase noise is to be accounted for in both Statistical and Time-Domain simulations.

Example:

 (Rx\_Rj (Usage Info)(Corner 0.005 0.006 0.004)(Type UI)

 (Description "RX Random Jitter in UI.")

)

clock\_times(n)=clock\_times(n)+Rx\_Rj \*gaussian\_rand()

clock\_times(n) is the times returned by Rx AMI\_Getwave

"Rx\_Sj" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines half the peak to peak variation, in seconds or UI, of a sinusoidal phase noise driven by impairments external to the receiver that are input to the RX CDR, but are not included in the CDR clock\_times output. This phase noise is to be accounted for in both Statistical and Time-Domain simulations.

 (Rx\_Sj (Usage Info)(Corner 0.05 0.07 0.04)(Type UI)

 (Description "RX Sinusoidal Jitter in UI.")

)

The following distribution shall be used if the Rx\_Sj\_Frequency is not defined.

clock\_times(n)=clock\_times(n)+Rx\_Sj\*sin(Pi\*rand())

rand() Returns random numbers between -.5 and +.5

Jitter PDF(dt) = 1/sqrt(1-(dt/Rx\_Sj)^2)

 (Note: Dual-Dirac)

I I I I

 I I I I

 I I I I

 I I I I

 I--------------I

"Rx\_Sj\_frequency" is an AMI parameter of Type Float and Usage either Info or Out which defines the frequency, in Hertz, of the sinusoidal phase noise driven by impairments external to the receiver that are input to the RX CDR, but are not included in the CDR clock\_times output.

Example:

 (Rx\_Sj\_Frequency (Usage Info)(Corner 6.5E7 6.5E7 6.5E7)(Type Float)

 (Description "Rx Sinusoidal Jitter Frequency in Hz.")

)

 clock\_times(n)=clock\_times(n)+Rx\_Sj\*sin(clock\_times(n)\*2\*Pi\*Rx\_Sj\_**Frequency**)

"Rx\_DCD" is an AMI parameter of Type either Float or UI and Usage either Info or Out which defines half the peak to peak variation, in seconds or UI, of a clock duty cycle distortion driven by impairments external to the receiver.

This phase noise is to be accounted for in both Statistical and Time-Domain simulations.

Example:

 (Rx\_DCD (Usage Info)(Corner 0.008 0.016 0.005)(Type UI)

 (Description "RX Duty Cycle Distortion in UI.")

)

clock\_times(n)=clock\_times(n)+Rx\_DCD\*(-1)n

The following optional Reserved Parameter is used to modify the statistics associated with the data input to the receiver’s sampling latch. This data is used by the simulator when post-processing the results from the model; the budget values specified by this parameter are not passed directly to the model itself.

"Rx\_Noise" is an AMI parameter of Type Float and Usage either Info or Out which defines the standard deviation, in volts into a 100 ohm differential load, of a set of independent samples of a Gaussian noise process measured at the sampling latch of a receiver.

Example:

 (Rx\_Noise (Usage Info)(Format Corner 0.0030 0.0035 0.0025) (Type Float)

 (Description "RX amplitude noise at sampling latch in V.")

)

wave(t)=wave(t)+Rx\_Noise\*gaussian\_rand()

wave(t) is the waveform returned by Rx AMI\_GetWave

Note:

The "Clock Jitter Parameters" (Rx\_Clock\_PDF, Rx\_Clock\_Recovery\_Mean, Rx\_Clock\_Recovery\_Rj, Rx\_Clock\_Recovery\_Sj, Rx\_Clock\_Recovery\_DCD , should be used by the simulator when analyzing the output of Rx AMI\_Init (for statistical analysis) or Rx AMI\_GetWave (time domain) when Rx AMI\_GetWave does not return clock\_times. When Rx AMI\_GetWave returns clock\_times, the simulator should not use the "Clock Jitter Parameters". An Rx AMI\_GetWave function should return clock\_times, unless it is a Repeater, in which case the AMI\_GetWave function may or may not return clock\_times.

Note:

The EDA Tool/Simulator shall use the values of these Jitter and Noise parameters directly if they are Usage Info. If they are Usage Out, then the EDA Tool/Simulator shall use their values generated by AMI\_Init. The model’s AMI\_GetWave function may return different values for these parameters **than** the values returned by AMI\_Init; the EDA Tool/Simulator may report the values of such parameters to the user, but the EDA Tool/Simulator may not change any inputs to AMI models or change other result of the simulation based on the values returned for the parameters in this BIRD by AMI\_GetWave.

Note:

When both an Sj and Sj\_Frequency is specified the time difference between the ideal and actual occurrence is defined by a sinusoidal function of time with a peak value of Sj and a frequency of Sj\_Frequency. When an Sj is specified, and Sj\_Frequency is not specified the time difference between the ideal and actual occurrence is defined by a sinusoidal function of time with a peak value of Sj at any frequency though avoiding commensurable ratios between S\_j\_Frequency and base frequency.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ANALYSIS PATH/DATA THAT LED TO SPECIFICATION

The parameters defined in this BIRD came from commercial IBIS-AMI model development efforts where new functionality was needed to meet customer expectations for model functionality, accuracy and performance. The parameters in this BIRD were defined by SiSoft and its semiconductor partners. These parameters are being contributed to IBIS to ensure IBIS-AMI model accuracy and portability.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ANY OTHER BACKGROUND INFORMATION:

This BIRD is being requested by the following IBIS users and model developers, in conjunction with the authors:

Cisco Systems: Upen Reddy, Doug White

Ericsson: Anders Ekholm

Broadcom: Yunong Gan

IBM: Adge Hawes

TI: Alfred Chong, Srikanth Sundaram

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*