

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

BIRD ID#: 123.12  
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 1, 2011; January 3, 2012  
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 Dj, Tx Sj, Tx\_Sj\_ ~~frequency~~ Frequency, Rx\_Clock\_Recovery\_Mean, Rx\_Clock\_Recovery\_Rj, Rx Clock Recovery Dj, Rx\_Clock\_Recovery\_Sj, Rx\_Clock\_Recovery\_DCD, Rx\_Rj, Rx Dj, Rx Sj, Rx\_DCD, and Rx\_Noise.

There are three sources of jitter that are accounted for using the parameters introduced in this BIRD; Tx Jitter, Rx Clock Data Recovery (CDR) Jitter and Rx Reference Clock Jitter. The Rx CDR has the ability to filter low frequency Tx Jitter and Rx Reference Clock Jitter. The parameters defined in this BIRD assume that the three sources of jitter are independent. IBIS 5.0 already defines parameters Tx Jitter, Tx DCD and Rx Clock PDF. Tx DCD is clarified in this BIRD. The parameters Tx Rj, Tx Dj, Tx Sj, and Tx Sj Frequency have similar functionality to the existing Tx Jitter, but offer more granularity in defining the various components of Tx Jitter. Similarly, Rx Clock Recovery Mean, Rx Clock Recovery Rj, Rx Clock Recovery Dj, Rx Clock Recovery Sj, and Rx Clock Recovery DCD offer increased granularity in defining the CDR behavior when doing statistical analysis, and when Rx AMI GetWave does not return clock times. Rx Rj, Rx Dj, Rx Sj, and Rx DCD describe jitter components that do not exist in IBIS 5.0, and offer the model maker a means to inform the EDA tool about how much extra jitter it should add to sampling instants. That is, these parameters indicate how much, as well as what type of, jitter is present in the actual device, but not reflected in the model. Rx Noise is unique among the parameters being introduced by this BIRD, in that it describes variations in the amplitude of the received signal, as opposed to variations in the time of sampling instants.

There are other possible methods of describing jitter. These include defining Tx and Rx Jitter Spectral Density distributions and applying Rx Jitter explicitly as

reference clock transition times. These advanced methods of handling jitter are left for a future BIRD.

Each parameter defined in the BIRD has included both a verbal and a mathematical description of how that parameter would affect the Tx transition times, Rx CDR and Rx jitter not accounted for in the CDR. In the AMI statistical flow, these impairments are treated as distributions which affect both the clock PDF and statistical eye. In the AMI time domain flow, the EDA tool may apply these jitter parameters directly to the Tx stimulus input and the Rx clock times using the given equations. Since these jitter parameters are independent, the EDA tool may use other statistical methods to account for these impairments.

Please note that even if no intrinsic jitter were present in the Tx and Rx, one would still experience an eye that has apparent jitter at the Rx data decision point. This jitter is caused by ISI, which is, typically, non-zero despite the efforts of Tx and Rx Equalization.

The model maker may assume that any and all non-zero values passed in these parameters will be used by the EDA tool in one of the following two phases:

- 1) Input stimulus generation, or
- 2) Post-processing of simulation results.

The definition of Tx DCD is clarified, and the allowed Usage is changed to Info.

\*\*\*\*\*

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

Tx\_DCD

~~Note that all of the parameters defined in this BIRD may be declared in the Model\_Specific section of the .ami file to allow the use of some legacy models. However, using these parameters in the Model\_Specific section of the .ami file is considered legacy use and will likely be deprecated in IBIS versions beyond 5.1.~~

\*\*\*\*\*

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) canmust be of Usage Info.
|           or Out. It can be of Type Float andor UI and can have Data
|           Format either Value, List, Rangeand, Corner, Increment, or
|           Steps. 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 transmitterEDA
|           tool by modifying
|           model. the stimulus input or by post processing the simulation
|           results.
|
|           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)<sup>n</sup>  
n\*bit time is the ideal time of the n<sup>th</sup> clock.  
Time(n) is the time of the n<sup>th</sup> clock modified when creating input waveforms for the Tx.  
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 stimulus 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.

~~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, Format either Value, List, Range, Corner, Increment, or Steps, and Usage ~~either~~ Info ~~or~~ Out which defines the standard deviation, in seconds or UI, of an white Gaussian phase noise process at the transmitter which is to be added to the behavior

implemented ~~directly by the transmitter model~~by the EDA tool by modifying the stimulus input or by post processing the simulation results.

Example:

```
(Tx_Rj (Usage Info) (Corner 0.005 0.006 0.004) (Type UI)
  (Description "TXTx Random Jitter in UI.")
)
```

"Tx\_Sj

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 white Gaussian distribution centered at zero with a standard deviation of 1.

"Tx Dj" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, and Usage ~~either~~ Info or Out which defines half the worst case peak to peak amplitude variation, in seconds or UI, of a sinusoidal jitter which is to be added to the behavior at the transmitter implemented directly by the transmitter modelEDA tool by modifying the stimulus input or by post processing the simulation results. Tx Dj shall include all deterministic and uncorrelated bounded jitter that is not accounted for by Tx DCD, and Tx Sj.

Example:

```
(Tx Dj (Usage Info) (Value .1) (Type UI)
  (Description "Tx Bounded Jitter in UI.")
)
```

Time(n)= n\*bit time+Tx Dj\*rand()

rand() is a function that returns floating point numbers between -.5 and +.5 with white uniform distribution.

"Tx Sj" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, and Usage Info 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 "TXTx Sinusoidal Jitter in UI.")
)
```

"Note: If Tx\_Sj\_frequencyFrequency is not assigned (either in the model or by the user), Tx Sj should be ignored.

"Tx Sj Frequency" is an AMI parameter of Type Float, Format Value, 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 UIFloat)
  (Description "TXTx Sinusoidal Jitter Frequency in Hz.")
)
```

~~"Tx\_DCD" is an AMI parameter of Type either Float or UI and Usage either Info or Out which 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:~~

```
(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 Sj*sin((n*bit time*2*Pi)*Tx Sj Frequency))
```

The following optional Reserved Parameters are used to specify characteristics of the ~~receiver~~ 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 when the model does not return clock times, or when Rx AMI GetWave is not used; the budget values specified by these

parameters are not passed directly to the model itself. For Rx models that do return clock times by AMI GetWave, these parameters represent the amount of jitter THAT HAD ALREADY BEEN IMPLEMENTED BY RX AMI GETWAVE AND ALREADY INCLUDED IN THE RETURNED clock times. For this reason, the EDA platform should NOT apply these jitter parameters again to the Rx clock times. These parameters are provided by the model creator to the EDA platform and end users for the sole purpose that these jitters can be properly accounted for when Rx AMI GetWave is NOT used or Rx clock times was not returned, in which cases the EDA platform is responsible to apply these jitters to the Rx output."

"Rx\_Clock\_Recovery\_Mean" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, 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 point half way between the PDF medians of consecutive eye diagram plus one half bit period zero crossings.

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 half way between the median of the eye crossing 0 on both sides of the eye.

"Rx\_Clock\_Recovery\_Rj" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, 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 and included in the clock times vector returned by the AMI GetWave function.

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\_SjRj\*gaussian rand()

"Rx\_Clock\_Recovery\_Dj" is an AMI parameter of Type either Float or UI ~~and Usage, Format either Value, List, Range, Corner, Increment, or Steps, and Usage Info ~~or Out~~ which defines half the worst case peak to peak variation, in seconds or UI, of ~~a sinusoidal phase noise exhibited by the the~~ recovered clock. Rx\_Clock\_Recovery\_Dj shall include all deterministic and uncorrelated bounded jitter that is included in the clock times vector returned by the AMI GetWave function and not accounted for by Rx\_Clock\_Recovery\_DCD and Rx\_Clock\_Recovery\_Sj.~~

Example:

```
(Rx_Clock_Recovery_Dj (Usage Info) (Value .1) (Type UI)
  (Description "Tx Bounded Jitter in UI.")
)
```

actual time = ideal time + Rx\_Clock\_Recovery\_Dj\*rand()

"Rx\_Clock\_Recovery\_Sj" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, and Usage Info which defines half the peak to peak variation, in seconds or UI, of a sinusoidal phase noise exhibited by the recovered clock and included in the clock times vector returned by the AMI GetWave function.

Example:

```
(Rx_Clock_Recovery_Sj (Usage Info) (Corner 0.05 0.07 0.4) (Type UI)
  (Description "RX Sinusoidal Jitter in UI-").")
)
```

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.

"Rx\_Clock\_Recovery\_DCD" is an AMI parameter of Type either Float or UI ~~and Usage, Format~~ either Value, List, Range, Corner, Increment, or Steps, and Usage 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 and included in the clock times vector returned by the AMI GetWave function.

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)<sup>n</sup>

The following optional Reserved Parameters are used to modify the statistics associated with ~~receiver's~~ receiver's recovered clock ~~when the model returns clock ticks information from an AMI Getwave call.~~ These parameters are used to account for jitter that is not included in either the clock times returned by Rx AMI GetWave or the Rx Clock Recovery parameters. 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" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, 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 by the EDA tool, in both Statistical and Time-Domain simulations.

Example:

```
(Rx_Rj (Usage Info) (Corner 0.005 0.006 0.004) (Type UI)
(Description "RXR Random Jitter in UI.")
)
```

~~"Rx\_Sj" is an AMI parameter of Type either Float or UI and Usage either Info or Out~~ clock times(n)=clock times(n)+Rx Rj \*gaussian rand()  
clock times(n) is the times returned by Rx AMI Getwave

"Rx Dj" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, and Usage Info which defines the worst case peak to peak variation, in seconds or UI, of the recovered clock, not including the random jitter specified by Rx Rj, Rx Sj, or Rx DCD . Rx Dj shall include all deterministic and uncorrelated bounded jitter that is not accounted for by either

Rx clock times, Rx Rj, or Rx Clock Recovery parameters. This phase noise is to be accounted for by the EDA tool in both Statistical and Time-Domain simulations.

Example:

```
(Rx Dj (Usage Info) (Value .1) (Type UI)  
(Description "Tx Bounded Jitter in UI.")  
)
```

actual time = ideal time + Rx Dj\*rand()

"Rx Sj" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, and Usage Info which defines half the peak to peak variation, in seconds or UI, of a sinusoidal phase noise ~~driven by impairments external to~~, but are not included in the receiver.  
CDR clock times output. This phase noise is to be accounted for by the EDA tool in both Statistical and Time-Domain simulations.

```
(Rx_Sj (Usage Info) (Corner 0.05 0.07 0.404) (Type UI)  
(Description "RX Sinusoidal Jitter in UI.")
```

+

```
)
```

actual time = ideal time + Rx Sj\*sin(Pi\*rand())  
rand() Returns random numbers between -.5 and +.5

"Rx\_DCD" is an AMI parameter of Type either Float or UI, Format either Value, List, Range, Corner, Increment, or Steps, 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 by the EDA tool 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.")
```

)

-actual time = ideal time + Rx DCD\*(-1)<sup>n</sup>  
n is the n<sup>th</sup> clock

The following optional Reserved Parameter is used to modify the statistics associated with the data input to the ~~receiver~~ receiver's sampling latch- (a.k.a. 'slicer'). 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, Format either Value, List, Range, Corner, Increment, or Steps, and Usage either Info or Out which defines the standard deviation, in ~~volts into a 100 ohm differential loadVolts~~, of a ~~set of independent samples of a white~~ Gaussian ~~noiserandom~~ process, which is to be added by the EDA tool to the signal measured at the sampling latch of a receiver.

Example:

```
(Rx_Noise (Usage Info) (Format Corner 0.0030 0.0035 0.0025Value .010) (Type Float) (Description "RXRx amplitude noise at sampling latch in VVolts."))
```

~~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\_Pad" is an AMI parameter of Type Float and Usage either Info or Out which defines the spectral density of the additive white Gaussian noise at the input to a receiver buffer in volts/sqrt(Hz). Ignored for a driver.~~

~~Example:~~

```
(Rx_Noise_Pad (Usage Info) (Format Value 7e-9) (Type Float) (Description "Rx spectral density of the additive white Gaussian noise at the input to a receiver buffer in volts/sqrt(Hz)."))  
+wave(t)=wave(t)+Rx_Noise*gaussian_rand()  
wave(t) is the waveform returned by Rx AMI GetWave
```

If Rx Noise is Usage Out, then the EDA tool shall use the value returned by Rx AMI Init if Rx AMI GetWave is not used. If Rx AMI GetWave is used, then the EDA tool may apply the value returned by each AMI GetWave call to the waveform returned by that call to AMI GetWave, or use the average value of Rx Noise returned by all calls to AMI GetWave (after Ignore Bits), or the value of Rx Noise returned by the last call to AMI GetWave.

Note:

The "~~Rx Clock Jitter Recovery~~ Parameters" (Rx\_Clock\_PDF, Rx\_Clock\_Recovery\_Mean, Rx\_Clock\_Recovery\_Rj, Rx\_Clock\_Recovery\_Sj, ~~Dj~~, Rx Clock Recovery Sj and 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

"Rx Clock Recovery Parameters".

~~"Clock Jitter Parameters".~~

Note:

The "Rx Jitter Parameters" (Rx Rj, Rx Dj, Rx Sj and Rx DCD, should be used by the simulator when analyzing the output of either Rx AMI Init (for statistical analysis) or Rx AMI GetWave (for time domain analysis).

Tables summarizing the rules for the jitter, noise and sensitivity parameters for information only.

```

*****
+-----+-----+
| General Rules | Allowed Usage |
+-----+-----+
| Reserved Parameter | Required Default | Info In Out InOut |
+-----+-----+
| Tx Jitter | No No Jitter | X |
| Tx Dj | No 0 | X |
| Tx Rj | No 0 | X |
| Tx Sj | No 0 | X |
| Tx DCD | No 0 | X |
| Tx Sj Frequency | No Undefined | X |
| Rx Receiver Sensitivity | No 0 | X X |
| Rx Clock PDF | No Clock Centered | X |
| Rx Clock Recovery Mean | No 0 | X |
| Rx Clock Recovery Dj | No 0 | X |
| Rx Clock Recovery Rj | No 0 | X |
| Rx Clock Recovery Sj | No 0 | X |
| Rx Clock Recovery DCD | No 0 | X |
| Rx Dj | No 0 | X |
| Rx Rj | No 0 | X |
| Rx Sj | No 0 | X |
| Rx DCD | No 0 | X |
| Rx Noise | No 0 | X X |
+-----+-----+

```

Table 1: General Rules and Allowed Usage for Reserved Parameters

	Data Type				
Reserved Parameter	Float	UI	Integer	String	Boolean
Tx Jitter	X	X			
Tx Dj	X	X			
Tx Rj	X	X			
Tx Sj	X	X			
Tx DCD	X	X			
Tx Sj Frequency	X				
Rx Receiver Sensitivity	X				
Rx Clock PDF	X	X			
Rx Recovery Mean	X	X			
Rx Clock Recovery Dj	X	X			
Rx Clock Recovery Rj	X	X			
Rx Clock Recovery Sj	X	X			
Rx Clock Recovery DCD	X	X			
Rx Dj	X	X			
Rx Rj	X	X			
Rx Sj	X	X			
Rx DCD	X	X			
Rx Noise	X				

Table 2: Allowed Data Types for Reserved Parameters

	Data Format										
Reserved Parameter	V	R	C	L	I	S	G	D	D	T	
	a	a	o	i	n	t	a	u	j	a	
	l	n	r	s	c	e	u	a	R	b	
	u	g	n	t	r	p	s	l	j	l	
*	e	e	e		e	s	s	-		e	
*			r		m		i	D			
*					e		a	i			
*					n		n	r			
*					t			a			
*								c			
Tx Jitter								X	X	X	X
Tx Dj	X	X	X	X	X	X					
Tx Rj	X	X	X	X	X	X					
Tx Sj	X	X	X	X	X	X					
Tx DCD	X	X	X	X	X	X					
Tx Sj Frequency	X	X	X	X	X	X					
Rx Receiver Sensitivity	X	X	X	X	X	X					
Rx Clock PDF								X	X	X	X
Rx Recovery Mean	X	X	X	X	X	X					
Rx Clock Recovery Dj	X	X	X	X	X	X					
Rx Clock Recovery Rj	X	X	X	X	X	X					
Rx Clock Recovery Sj	X	X	X	X	X	X					
Rx Clock Recovery DCD	X	X	X	X	X	X					
Rx Dj	X	X	X	X	X	X					
Rx Rj	X	X	X	X	X	X					
Rx Sj	X	X	X	X	X	X					
Rx DCD	X	X	X	X	X	X					
Rx Noise	X	X	X	X	X	X					

Table 3: Allowed Data Format for Reserved Parameters

\*\*\*\*\*

With the exception of the "Table" format, the Tx Jitter parameter has been essentially superseded by the Reserved Parameters Tx Rj, Tx Dj, Tx Sj, Tx Sj Frequency, and Tx DCD, which enable SerDes transmitter jitter to be specified in greater detail. It is recommended for AMI model developers to use these preferred jitter parameters when possible instead of Tx Jitter.

With the exception of the "Table" format, the Rx Clock PDF parameter has been essentially superseded by the Reserved Parameters Rx Clock Recovery Rj, Rx Clock Recovery Dj, Rx Clock Recovery Sj, and Rx Clock Recovery DCD, which enable SerDes receiver jitter to be specified in greater detail. It is recommended for AMI model developers to use these preferred jitter parameters when possible instead of Rx Clock PDF.

\*\*\*\*\*

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.  
~~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

Markup copies of this document, in Adobe PDF\* and Microsoft Word\* format, are available at:  
[http://www.eda.org/ibis/birds/bird123.2/bird123.2\\_markup.pdf](http://www.eda.org/ibis/birds/bird123.2/bird123.2_markup.pdf)  
[http://www.eda.org/ibis/birds/bird123.2/bird123.2\\_markup.docx](http://www.eda.org/ibis/birds/bird123.2/bird123.2_markup.docx)

\*\*\*\*\*