**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** (TBD)

**ISSUE TITLE:** Separating High- and Low-Frequency Tx Random Jitter Parameters

**REQUESTOR:**  Michael Mirmak, Intel Corp.

**DATE SUBMITTED:** (Draft 1 submitted to ATM July 18, 2019)

**DATE REVISED:**

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

In high-speed system designs, not all frequencies of random jitter from a transmitter will be modified by the channel. High-frequency jitter components will be amplified by the channel, while low-frequency jitter components will pass through the channel unaffected.

IBIS 7.0 still only supports a single “Tx\_Rj” parameter, which does not distinguish between high- and low-frequency effects. As a result, any low-frequency transmitter random jitter must be manually added (using root-sum-square techniques) to the receiver random jitter. This is inconvenient and prone to error. Separate Tx random jitter high- and low-frequency components are needed.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. High- and low-frequency transmitter random jitter parameter must be available.
 |  |
| 1. Support for existing transmitter random jitter values in .ami files must be maintained.
 |  |

**SUMMARY OF PROPOSED CHANGES:**

For review purposes, the proposed changes are summarized as follows:

Table 2: IBIS Keywords, Subparameters, AMI Reserved\_Parameters, and AMI functions Affected

|  |  |  |
| --- | --- | --- |
| Specification Item | New/Modified/Other | Notes |
| Tx\_Rj\_HF, Tx\_Rj | Modified | Tx\_Rj is expanded with a new synonymous parameter to cover high-frequency jitter when low-frequency random jitter data is provided. |
| Tx\_Rj\_LF | New | A new low-frequency Tx random jitter parameter is defined. |

**PROPOSED CHANGES:**

Change the following text in IBIS 7.0 starting on PDF page 237:

*Parameter:* **Tx\_Rj**

*Required:* No, and illegal before AMI\_Version 6.0

*Direction:* Tx

*Descriptors*:

Usage: Info, Out, Dep

Type: Float, UI

Format: Value, List, Range, Corner, Increment, Steps

Default: <numeric\_literal*>*

Description:<string>

*Definition:* The standard deviation of a white Gaussian phase noise process at the transmitter which is to be added to the behavior implemented by the EDA tool by modifying the stimulus input or by post processing the simulation results. Entries are assumed to be in units of seconds when declared as Type Float.

*Usage Rules:*

*Other Notes:* Time is calculated as follows:

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

where gaussian\_rand() is a function that returns floating point numbers between -inf and +inf. The distribution of these numbers shall be a white Gaussian distribution centered at 0.0 with a standard deviation of 1.0. The EDA tool can protect against abs(Tx\_Rj\*gaussian\_rand()) > 0.5UI.

*Example:*

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

 (Description "Tx Random Jitter in UI."))

… to:

*Parameter:* **Tx\_Rj, Tx\_Rj\_HF**

*Required:* No, and illegal before AMI\_Version 6.0; Tx\_Rj\_HF is illegal before AMI\_Version 7.1

*Direction:* Tx

*Descriptors*:

Usage: Info, Out, Dep

Type: Float, UI

Format: Value, List, Range, Corner, Increment, Steps

Default: <numeric\_literal*>*

Description:<string>

*Definition:* The standard deviation of a white Gaussian phase noise process at the transmitter which is to be added to the behavior implemented by the EDA tool by modifying the stimulus input or by post processing the simulation results. Entries are assumed to be in units of seconds when declared as Type Float.

*Usage Rules:*

*Other Notes:* Time is calculated as follows:

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

where gaussian\_rand() is a function that returns floating point numbers between -inf and +inf. The distribution of these numbers shall be a white Gaussian distribution centered at 0.0 with a standard deviation of 1.0. The EDA tool can protect against abs(Tx\_Rj\*gaussian\_rand()) > 0.5UI.

In the presence of “Tx\_Rj\_LF” and if “Tx\_Rj\_HF” is not used, “Tx\_Rj” will be interpreted as including only high-frequency jitter components. The use of “Tx\_Rj\_HF” and “Tx\_Rj” simultaneously for the same model is prohibited. The use of “Tx\_Rj\_LF” by itself in a model without either “Tx\_Rj” or ”Tx\_Rj\_HF” is prohibited.

Note: Separating jitter into individual high- and low-frequency response blocks based on a single cutoff frequency is an oversimplification. An ideal treatment that would correctly take jitter amplification by the channel into account would specify the spectral density of the jitter.

*Example:*

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

 (Description "Tx Random Jitter in UI."))

Add the following immediately after the “Tx\_Rj” parameter:

*Parameter:* **Tx\_Rj\_LF**

*Required:* No, and illegal before AMI\_Version 7.1

*Direction:* Tx

Descriptors:

Usage: Info, Out, Dep

Type: Float, UI

Format: Value, List, Range, Corner, Increment, Steps

Default: <numeric\_literal>

Description: <string>

*Definition:*

The standard deviation of a white Gaussian phase noise process at the transmitter, below a critical frequency (hence “LF” or “low frequency”), which is to be added to the behavior implemented by the EDA tool by modifying the signal *at the receiver*. The receiver behavior is modified because the low frequency jitter is assumed to be unaffected by channel amplification. Note that the particular critical frequency (or frequencies) distinguishing between low- and high-frequency jitter is not relevant; only the fact that some jitter components should be immune from channel effects is important.

Entries are assumed to be in units of seconds when declared as Type Float.

*Usage Rules:
Other Notes:* Due to its nature, the stimulus for the transmitter is unaffected by “Tx\_Rj\_LF”. In the presence of “Tx\_Rj\_LF” and if “Tx\_Rj\_HF” is not used, the entry for “Tx\_Rj” will be interpreted as high-frequency jitter. The equation for Tx\_Rj when covering high-frequency jitter remains unchanged.

Note: Separating jitter into individual high- and low-frequency response blocks based on a single cutoff frequency is an oversimplification. An ideal treatment that would correctly take jitter amplification by the channel into account would specify the spectral density of the jitter.

*Example:*(Tx\_Rj (Usage Info) (Corner 0.005 0.006 0.004) (Type UI)
(Description "Tx Random Jitter in UI."))