**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 222

**ISSUE TITLE:** Clock Times Clarifications

**REQUESTOR:**  Arpad Muranyi, Siemens EDA

**DATE SUBMITTED:** November 8, 2022

**DATE REVISED:**

**DATE ACCEPTED:** December 9, 2022

**DEFINITION OF THE ISSUE:**

The one and only definition for “clock\_times” in the IBIS v7.1 specification appears on pg. 229. This definition was written for SerDes Rx AMI executable models, where the clock\_times argument was only used as an output by the AMI\_GetWave function.

IBIS v7.1 introduced a new AMI parameter “Rx\_Use\_Clock\_Input” for clocked Rx AMI executable models, which is described on pg. 253. For clocked Rx AMI executable models, the clock\_times vector is defined as an input as well as an output. The clock waveform or the clock time values are passed into the clocked Rx AMI executable model through the clock\_times argument of the AMI\_GetWave function.

The main problem with the clock\_times section on pg. 229 is that it only describes the content of this argument when it is used as an output. The specification does not discuss the content and rules of the clock\_times argument when it is used as an input. This can potentially confuse the reader of the specification and may result in erroneous models or EDA vendor implementations for clocked Rx AMI executable models.

**SUMMARY OF PROPOSED CHANGES:**

The changes proposed below extend the definition and rules of the clock\_times argument of the AMI\_GetWave function to cover the situation when it is used as an input for clocked Rx AMI executable models with the “Wave” or “Times” options in the Rx\_Use\_Clock\_Input AMI parameter.

**PROPOSED CHANGES:**

**On pg. 229 of IBIS v7.1**, replace the following text:

“Vector to return clock times. The clock times are referenced to the start of the simulation (the first AMI\_GetWave call). The clock\_times vector is allocated by the EDA tool and is guaranteed to be greater than the number of clocks expected during the AMI\_GetWave call. The clock times are exactly symbol\_time/2 before the input data signal is sampled. The algorithmic model will return non-negative clock\_times values, and place -1 after the last valid clock tick in the clock\_times vector during each AMI\_GetWave call. If there are no valid clock ticks for the duration of an AMI\_GetWave call, a single entry of -1 will be returned in the clock\_times vector. The units of clock\_times are seconds.

The clock ticks represented by clock times should be strictly monotonic, both within the clock\_times vector returned from a single call to AMI\_GetWave and between successive calls to AMI\_GetWave. That is, within a given clock\_times vector each successive valid value is greater than the value that preceded it, and the first valid value from a given call to AMI\_GetWave must be greater than the last valid value from the preceding call to AMI\_GetWave. Any non-strictly-monotonic behavior of clock times (including two identical values) should be considered by the EDA tool as an algorithmic model failure.

Each valid value in the clock\_times vector shall be used to sample the output waveform by adding to it symbol\_time/2, regardless of whether that waveform sample occurs in the waveform segment being returned by the current call to AMI\_GetWave, or in the waveform segment to be returned by the next AMI\_GetWave call. Care should be taken in implementation of clock\_times to ensure that the calculations used always maintain full double-precision floating-point accuracy across multi-million-bit simulations.

Although clock\_times will generally be related to the unit interval for the primary SerDes channel being simulated, there is no requirement that there be any relationship between the clock ticks generated by clock\_times and the actual waveform returned in the primary channel. It is possible for the CDR to go out of lock, resulting in clock ticks that have no definite relationship to the output wave. It is possible for the CDR to be suppressed for an undefined number of bits until the output of the first clock tick. In the case of a receiver without a CDR, it is possible for only -1 to ever be output during all AMI\_GetWave calls.”

**Corresponding text in ver7\_2\_draft3.docx** (highlights indicate changes from IBIS v7.1 due to incorporating BIRDs affecting this section):

Vector to return clock times. The clock times are referenced to the start of the simulation (the first AMI\_GetWave call). The clock\_times vector is allocated by the EDA tool and is guaranteed to be greater than the number of clocks expected during the AMI\_GetWave call. The sample times equal clock\_times + ½ UI + offset, where offset is defined by Reserved Parameters PAM\_Offsets or PAM4\_UpperEyeOffset, PAM4\_CenterEyeOffset and PAM4\_LowerEyeOffset. In the absence of these parameters, offset is assumed to be 0. The algorithmic model will return non-negative clock\_times values, and place -1 after the last valid clock time in the clock\_times vector during each AMI\_GetWave call. If there are no valid clock times for the duration of an AMI\_GetWave call, a single entry of -1 will be returned in the clock\_times vector. If an Rx AMI\_GetWave only returns -1 during all AMI\_GetWave calls then the model does not generate clock times.

Except for Redriver receiver models, it is highly recommended that all receiver models return valid clock times. A receiver model that specifies the Rx\_Use\_Clock\_Input parameter must return valid clock times. The units of clock\_times are seconds.

The clock time values within the clock\_times vector should be strictly monotonic, both within the clock\_times vector returned from a single call to AMI\_GetWave and between successive calls to AMI\_GetWave. That is, within a given clock\_times vector each successive valid value is greater than the value that preceded it, and the first valid value from a given call to AMI\_GetWave must be greater than the last valid value from the preceding call to AMI\_GetWave. Any non-strictly-monotonic behavior of clock times (including two identical values) should be considered by the EDA tool as an algorithmic model failure.

Each valid value in the clock\_times vector shall be used to sample the output waveform by adding to it symbol\_time/2, regardless of whether that waveform sample occurs in the waveform segment being returned by the current call to AMI\_GetWave, or in the waveform segment to be returned by the next AMI\_GetWave call. Care should be taken in implementation of clock\_times to ensure that the calculations used always maintain full double-precision floating-point accuracy across multi-million-bit simulations.

Although the clock times will generally be related to the unit interval for the primary SerDes channel being simulated, there is no requirement that there be any relationship between the clock times in the clock\_times vector and the actual waveform returned in the primary channel. It is possible for the CDR to go out of lock, resulting in clock times that have no definite relationship to the output wave. It is possible for the CDR to be suppressed for an undefined number of bits until the first clock time value is written to the clock\_times vector.

**With** (proposed changes in this BIRD are in cyan):

“Vector to return clock times, or, when Rx\_Use\_Clock\_Input is set to “Times” or “Wave”, a vector to input clock times or a clock waveform, respectively. If a model does not have the Rx\_Use\_Clock\_Input parameter, it is considered to have a “single input” Rx AMI\_GetWave function. If a model has the Rx\_Use\_Clock\_Input parameter that is set to “Times” or “Wave”, it is considered to have a “dual input” Rx AMI\_GetWave function. The clock times or clock waveform are referenced to the start of the simulation (the first AMI\_GetWave call). The Memory for clock\_times vector is allocated by the EDA tool and must be large enough to hold the clock times or clock waveform expected during the AMI\_GetWave calls.

***Change with respect to IBIS 7.1:***

When the clock times are the output of single or dual input Rx AMI\_GetWave functions, they are exactly symbol\_time/2 before the input data signal is sampled.

***Change with respect to text in ver7\_2\_draft3.docx:***

When the clock times are the output of single or dual input Rx AMI\_GetWave functions, the sampling times equal clock\_times + ½ UI + offset, where offset is defined by Reserved Parameters PAM\_Offsets or PAM4\_UpperEyeOffset, PAM4\_CenterEyeOffset and PAM4\_LowerEyeOffset. In the absence of these parameters, offset is assumed to be 0.

***Double strike through text is from IBIS v7.1, and is replaced by text in ver7\_2\_draft3.docx:***

All clock time values must be non-negative except the value after the last valid clock time, which shall be -1 to indicate the end of the clock\_times vector during each AMI\_GetWave call. If there are no valid clock ticks for the duration of an AMI\_GetWave call, a single entry of -1 will be returned in the clock\_times vector. If an Rx AMI\_GetWave only returns -1 during all AMI\_GetWave calls then the model does not generate clock times.

Except for Redriver receiver models, it is highly recommended that all receiver models return valid clock times. A receiver model that specifies the Rx\_Use\_Clock\_Input parameter shall return valid clock times. The units of the clock time values is seconds and the unit of the waveform samples is volts.

The clock ticks represented by clock times The clock time values within the clock\_times vector should be strictly monotonic, both within the clock\_times vector in a single call to AMI\_GetWave and between successive calls to AMI\_GetWave. That is, within a given clock\_times vector, each successive valid value must be greater than the value that preceded it, and the first valid value from a given call to AMI\_GetWave must be greater than the last valid value from the preceding call to AMI\_GetWave.

Any non-strictly-monotonic behavior of clock times (including two identical values) should be considered an algorithmic model failure by the EDA tool.

Each valid value in the clock\_times vector returned by single input Rx AMI\_GetWave function calls or dual input Data Rx AMI\_GetWave function calls shall be used to sample the output waveform of the AMI\_GetWave function by adding symbol\_time/2 ½ UI to the clock times, regardless of whether that waveform sample occurs in the waveform segment being returned by the current call to AMI\_GetWave, or in the waveform segment to be returned by the next AMI\_GetWave call. Care should be taken in implementation of clock\_times to ensure that the calculations used always maintain full double-precision floating-point accuracy across multi-million-symbol simulations.

In SerDes receivers it is possible for the CDR to go out of lock, resulting in clock ticks times that have no definite relationship to the output wave. It is possible for the CDR to be suppressed for an undefined number of symbols until the output of the first clock tick. In the case of a SerDes receiver without a CDR, it is possible for only -1 to ever be output during all AMI\_GetWave calls first clock time value is written to the clock\_times vector.”

On pg. 253, change:

*Definition:* Specifies the content of the Data Rx AMI\_GetWave clock\_times input supported by the model. The three possible content types are: (1) to be ignored, (2) the clock\_times and (3) the wave output of the Clock Rx AMI\_GetWave. If this parameter is present in the .ami file, the EDA tool is responsible to pass the selected value to the AMI\_Init function.

To:

*Definition:* Specifies the content of the Data Rx AMI\_GetWave clock\_times input supported by the model. The three possible content types are: (1) to be ignored, (2) the clock\_times and (3) the wave output of the Clock Rx AMI\_GetWave. For types (2) and (3) the clock\_times input represents the external clock used by the Data Rx AMI\_GetWave function. If this parameter is present in the .ami file, the EDA tool is responsible to pass the selected value to the AMI\_Init function.

On pg. 254, change:

*Other Notes:* The wave input to both Data and Clock shall have the same block size and sample\_interval. For the “Wave” option, if the Clock does not have a DLL or has a DLL without an AMI\_GetWave, then the EDA tool should effectively insert a passthrough Clock AMI\_GetWave function to make the clock waveform available for the Data AMI\_GetWave clock\_times input. For the “Times” option, the Clock shall have a DLL with an AMI\_GetWave that returns clock\_times.

To:

*Other Notes:* For the “Wave” option, the Data and Clock input waveforms shall have the same block size and sample\_interval, and their sample times shall cover the same time span in each AMI\_GetWave call.

For the “Wave” or “Times” option, the Clock shall have an AMI executable model with an AMI\_GetWave function. When the Data model’s Rx\_Use\_Clock\_Input parameter includes the “Times” option, the corresponding Clock AMI\_GetWave function shall return a clock times vector.

The EDA tool should pass the clock times vector or clock waveform generated by each of the Clock AMI\_GetWave function calls into the corresponding dual input Data Rx AMI\_GetWave function call without altering it in any way.

**BACKGROUND INFORMATION/HISTORY:**

The changes proposed here were discussed extensively in the Advanced Technology Task Group (ATM) meetings in October 2022.