**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER: 231**

**ISSUE TITLE:** Clarifications on AMI Block Concepts

**REQUESTOR:**  Michael Mirmak, Intel Corporation

**DATE SUBMITTED:** March 26, 2024

**DATE REVISED:**

**DATE ACCEPTED:** May 31, 2024

**DEFINITION OF THE ISSUE:**

The words “block” and “segment” are used in relation to the IBIS Algorithmic Modeling Interface waveform data without clear definitions in the specification text, particularly in sections related to the Back-Channel Interface.

This BIRD makes several changes to ensure that “block” and “segment” are understood refer to the same concept, and to simplify related text. Actual definitions for these and other concepts (e.g., UI and NRZ) will be handed through a separate BIRD, BIRD 230.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. “Block” and “segment” shall clearly refer to the same concept.
 |  |
| 1. “Block” and “segment” definitions shall clarify whether the group of waveform samples is assumed to contain one complete symbol, multiple symbols, or may not need to align with the time boundaries of a symbol.
 | The current specification explicitly allows blocks to be misaligned with the edges of a UI. |

**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 |
| Identical text on pages 219, 295, 299 shall be updated to define “block” | Modified |  |
| Page 216, link “segment” and “block” concepts  | Modified |  |

**PROPOSED CHANGES:**

All page numbers below refer to the IBIS 7.2 document in Adobe PDF format. Note that the changes proposed for pages 219, 295, and 299 are identical.

On page 216, remove the text in strikeout font and add the highlighted text as shown below.

7. The EDA tool generates a time-domain digital input waveform bit pattern (stimulus). A long bit pattern (and simulation) may be broken up into multiple time segments (or blocks) by the EDA tool. For example, if one million NRZ bits are to be simulated, the analysis may be executed in ~~there can be~~ 1000 time segments of 1000 bits each. The segments are not required to be equally sized and are not required to contain an integer number of bits.

On page 219, remove the text in strikeout font and add the highlighted text as shown below.

Steps 4 through 8 can be called once or can be called multiple times to process the full analog waveform. ~~Splitting up~~ Processing the full analog waveform ~~into~~ using multiple calls, where each call passes the model a block of samples, reduces the memory requirements when ~~doing~~ performing long simulations. ~~And~~ This also allows AMI\_GetWave to return model status ~~every so many bits~~ after processing each block (note that the size of a block is not limited to the number of samples needed to define an integer number of complete symbols). Once all the blocks of the input waveform have been processed...

On page 238, remove the text in strikeout font and add the highlighted text as shown below.

**UI**

Unit Interval. One (1) UI is the inverse of the symbol rate. For example, 1 UI of 100 ps for an NRZ ~~channel~~ link transmits 10 Gb/s, while the same 100 ps UI for a PAM4 ~~channel~~ link transmits 20 Gb/s (or 10 Gsymbols/s). Parameters of Type UI accept values in units of UI (i.e., symbol time). The parameter may take on either floating-point or integer values.

On page 295, remove the text in strikeout font and add the highlighted text as shown below.

Steps 7 through 8 can be called once or can be called multiple times to process the full analog waveform. ~~Splitting up~~ Processing the full analog waveform ~~into~~ using multiple calls, where each call passes the model a block of samples, reduces the memory requirements when ~~doing~~ performing long simulations. ~~And~~ This also allows AMI\_GetWave to return model status ~~every so many bits~~ after processing each block (note that the size of a block is not limited to the number of samples needed to define an integer number of complete symbols). Once all blocks of the input waveform have been processed…

On page 299, remove the text in strikeout font and add the highlighted text as shown below.

Steps 7 through 8 can be called once or can be called multiple times to process the full analog waveform. ~~Splitting up~~ Processing the full analog waveform ~~into~~ using multiple calls, where each call passes the model a block of samples, reduces the memory requirements when ~~doing~~ performing long simulations. ~~And~~ This also allows AMI\_GetWave to return model status ~~every so many bits~~ after processing each block (note that the size of a block is not limited to the number of samples needed to define an integer number of complete symbols). Once all blocks of the input waveform have been processed…

On page 303, remove the text in strikeout font and add the highlighted text as shown below.

*Usage Rules:* BCI\_Message\_Interval\_UI may be used by the EDA tool to manage the size in UI of the block of waveform samples passed to the AMI\_GetWave function ~~block size~~ to provide better synchronization between the times a model has a message to send and the actual timing of the AMI\_GetWave block boundaries when messaging may occur. Note that a block is assumed to contain waveform samples for at least one complete UI.

BCI\_Message\_Interval\_UI ~~must~~ shall be present if BCI\_Protocol is present. BCI\_Message\_Interval\_UI ~~must~~ shall be absent if BCI\_Protocol is absent.

*Other Notes:* This parameter allows a BCI\_Protocol to define the number of training ~~bits~~ UI ~~(“dwell time”)~~ between BCI messages, which necessarily must occur at most once per AMI\_GetWave call. Protocols and models implementing them should not expect AMI\_GetWave boundaries to occur precisely when a message (e.g., for a Tx adaptation) is ready to be sent. Adaptation engines within the models must therefore be capable of performing correctly without regard to the actual AMI\_GetWave block size the EDA tool chooses.

Note that if an adaptation message is ready early in an AMI\_GetWave block the adaptation engine must wait for the message to be sent and have an effect on GetWave results before it can begin to acquire information associated with performance at the new settings to determine the next adaptation. This means the adaptation process is interrupted for the remainder of the AMI\_GetWave block, adding to the overall number of UI that must be processed in the time domain simulation to complete the adaptation. The model maker/protocol designer should choose a value of BCI\_Message\_Interval\_UI that is slightly larger than the smallest number of training UI required per adaptation.

To ensure good messaging efficiency, the EDA tool should consider choosing an AMI\_Get~~w~~Wave block size such that either a single AMI\_GetWave block or some number of concatenated AMI\_GetWave blocks spans a number of UI equal to or slightly larger than BCI\_Message\_Interval\_UI.

**BACKGROUND INFORMATION/HISTORY:**

Note that not every instance of the word “block” is addressed here. In some cases, “block” is used to refer to a functional module or section of code rather than a grouping of waveform data. These instances have not been modified here.

Some spacing issues have been corrected here without being highlighted.

Technically, BCI\_Message\_Interval\_UI is permitted to use a value of zero (integers are supported but no minimum value is explicitly defined). This value should be required to be 1 or larger, which will be done in a separate BIRD.