**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 229

**ISSUE TITLE:** AMI Test Data Support

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

**DATE SUBMITTED:** Dececmber 19, 2023

**DATE REVISED:**

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

The [Test Load] and [Test Data] keywords for traditional IBIS define structures for representing analog waveforms, as well as the circuits outside the buffer and package models used in generating these waveforms. Unfortunately, these keywords are insufficient for use with IBIS AMI models; UI, step size, and model-specific parameters must be specified along with the test load information to remove ambiguity in how the waveforms are generated. Further, the waveforms themselves may be too large for convenient in-line representation in a .ibs file.

Here a new keyword is proposed to define input and output test data for use with IBIS AMI simulation. The keyword’s subparameters include the required information that would be passed by the simulation tool into the IBIS AMI models. The keyword also supports providing input waveforms, where appropriate, and reporting associated output waveforms generated by the model-maker as separate files.

Channel correlation will be handled by separate updates to the [Test Load] and [Test Data] keywords.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. Define all parameters that would naturally be defined by the model-maker and simulation tool based on .ami parameter ranges plus the parameters outside the .ami file that are required by the AMI\_Init and AMI\_GetWave function calls
 |  |
| 1. Support a separate file that enumerates the actual parameter string to be sent to the model from the tool after configuration
 |  |
| 1. Support a separate file that enumerates the parameter string expected from the model and returned to the tool
 |  |
| 1. Support a separate file that defines the waveform data to be expected as output from the AMI\_Init and/or AMI\_GetWave function calls.
 |  |
| 1. Support a separate file that defines the stimulus to be used for AMI\_Init and AMI\_GetWave function calls.
 |  |
| 1. Support an optional separate input file that defines clock times or waveform where appropriate to the model and configuration used
 |  |
| 1. Support a separate output file that defines clock times or clock waveform where appropriate to the model and configuration used
 |  |
| 1. Exclude any results that rely in channel information not present in the input parameters and input stimulus file.
 |  |

**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 |
| [AMI Test Configuration]  | New |  |

**PROPOSED CHANGES:**

**The following text should be added after Section 10.11 as defined in IBIS 7.2.**

*Keywords:* **[AMI Test Configuration]**

*Required:* No, and illegal before IBIS 7.3.

*Description:* Defines example input stimulus and resulting output waveform data plus AMI configuration information for testing of the associated [Algorithmic Model].

The intent of [AMI Test Configuration] is to enable direct comparison by a simulation tool of the output of the associated AMI model, with appropriate configuration and stimulus information, against a “Golden Waveform” provided by the model maker. The enabled comparison is solely at the AMI level and does not involve simulating using a channel model or buffer loading.

*Sub-Params:* Type, Direction, AMI\_input\_parameters\_file, Input\_IR\_file, Input\_waveform\_file, Golden\_IR\_file, Golden\_waveform\_file, Clock\_input\_file, Clock\_output\_file, AMI\_output\_parameters\_file, Executable\_index

*Usage Rules:* The [AMI Test Configuration] keyword takes a single name argument, which is required, as it supports selection of specific input and output configurations for a given buffer and test waveform combination. The [AMI Test Configuration] argument shall be fewer than 1000 characters in length and shall not be duplicated for any other [AMI Test Configuration] keywords appearing under the same [Algorithmic Model] keyword. A given [Algorithmic Model] may contain any number of [AMI Test Configuration] sections, representing different simulation conditions.

The placement of this keyword within the hierarchy of IBIS is shown below:

├── **[Component]**

├── **[Model]**

│ └── **[Algorithmic Model]** Executable, Executable\_Rx,

**│ │** Exectuable\_Tx

│ ├───── **[AMI Test Configuration]** Type, Direction,

**│ │** AMI\_input\_parameters\_file,

**│ │** Input\_IR\_file,

**│ │** Input\_waveform\_file,

**│ │** Golden\_IR\_file,

**│ │** Golden\_waveform\_file,

│ **│** Clock\_input\_file,

**│ │** Clock\_output\_file,

**│ │** AMI\_output\_parameters\_file,

**│ │** Executable\_index

│ └───── **[End Algorithmic Model]**

The Type subparameter is required, and its arguments shall be one of either “Statistical” or “Time\_domain”. This indicates the kind of configuration data required, the impulse response and/or waveform input data required, and impulse response or waveform output to expect from a simulation of the buffer being tested.

The Direction subparameter is required. Its argument specifies the direction of the “device under test” and shall be one of the following strings: “Tx” or “Rx”. Note that using “Tx” for a buffer which is incapable of a driving mode of operation is illegal. Similarly, using “Rx” for a buffer which is incapable of a receiving mode of operation is illegal. Use of “Tx” or “Rx” for an I/O buffer will identify the mode of operation for the test information and waveform(s).

The AMI\_input\_parameters\_file subparameter is required. This subparameter takes a single string argument corresponding to a filename present in the same directory as the calling .ibs file. The file’s contents consist of rows of space-separated string data, using formatting rules identical to that of the AMI\_parameters\_in string. This data is grouped in two sections. The first section, which uses the required identifier “Simulator\_parameters”, includes those parameters and values which are needed to unambiguously define the simulation conditions used to generate the golden waveforms provided and, in a normal simulation context, would be supplied by the user through a simulator user interface (i.e., AMI function signature parameters). The second section, which uses the required identifier “Model\_parameters”, includes the AMI\_parameter\_in string contents that would be generated by an EDA tool and passed to the model, specifically the Usage In and Usage InOut parameters from the .ami file contents and their associated values.

The structure is shown below; note that the line feeds shown may be omitted as in the AMI\_parameters\_in string.

For Type Statistical waveforms, the “Simulator\_parameters” section shall include four required AMI\_Init and/or AMI\_Impulse function signature parameters: Sample\_interval, Symbol\_time, Number\_of\_rows, and Aggressors, as shown below:

(Simulator\_parameters

(Sample\_interval <float>)

 (Symbol\_time <float>)

 (Number\_of\_rows <integer>)

 (Aggressors <integer>)

)

(Model\_parameters

 (<root name>

 (<Usage In or InOut parameter name> <value>)

 (<Usage In or InOut parameter name> <value>)

 (<Usage In or InOut parameter name> <value>)

 …

)

)

For Type Time\_Domain waveforms, the “Simulator\_parameters” section shall include three required AMI\_GetWave and AMI\_Init and/or AMI\_Impulse function signature parameters: Sample\_interval, Symbol\_time, and Wave\_size, as shown below

(Simulator\_parameters

(Sample\_interval <float>)

 (Symbol\_time <float>)

 (Wave\_size <integer>)

)

(Model\_parameters

 (<root name>

 (<Usage In or InOut parameter name> <value>)

 (<Usage In or InOut parameter name> <value>)

 (<Usage In or InOut parameter name> <value>)

 …

)

)

All Usage In and InOut parameters supported by the associated model’s .ami file shall be included in the AMI\_input\_parameters\_file contents. Note that these Usage In and InOut parameters are preceded by several parameters supplied as part of the AMI\_Init and AMI\_GetWave function calls; the values supplied as part of those calls should match those provided in the AMI\_input\_parameters\_file.

long AMI\_Init (double \*impulse\_matrix, long number\_of\_rows, long aggressors, double sample\_interval, double symbol\_time, char \*AMI\_parameters\_in, char \*\*AMI\_parameters\_out, void \*\*AMI\_memory\_handle, char \*\*msg)

long AMI\_GetWave (double \*wave, long wave\_size, double \*clock\_times, char \*\*AMI\_parameters\_out, void \*AMI\_memory)

Though not a requirement of AMI\_GetWave simulations generally, for the purposes of test generation using [AMI Test Configuration], wave\_size shall be a fixed value across all calls to AMI\_GetWave (see below).

No Usage Out or Usage Info parameters shall be listed in the AMI\_input\_parameters\_file contents; only Usage In and Usage InOut parameters shall be provided.

The Input\_IR\_file subparameter is required for all Direction and Type combinations. This subparameter takes a single string argument corresponding to a filename present in the same directory as the calling .ibs file.

The Input\_IR\_file contents consist of rows of whitespace-separated numeric data representing an impulse response. The values shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9). Each file shall contain at least one row. The vectors contain the impulse responses for the channel connected to the model being characterized, and optionally the crosstalk impulse responses from aggressor channels. The number of columns in the Input\_IR\_file shall be at least 1 and shall be equal to 1 plus the value of the Aggressors parameter.

The Input\_waveform\_file subparameter is required for Type Time\_domain and is prohibited for Type Statistical. The subparameter takes a single string argument corresponding to a filename present in the same directory as the calling .ibs file.

The Input\_waveform\_file contents consist of rows of numeric data representing voltages. The values shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9). The contents of the Input\_waveform\_file are equivalent to the contents of the “wave” argument of the AMI\_GetWave function; these are described in the functional reference flows of Section 10.2.2.

The Input\_waveform\_file contents for Direction Rx are equivalent to the input to the Rx AMI\_GetWave function during a normal channel simulation that includes both Tx and Rx AMI executables.

The time duration represented by the contents of the Input\_waveform\_file shall match the time duration represented by the contents of the Golden\_waveform\_file (see below). Note that the time values are implied based on the assumption of simulation starting at time = 0 and the use of a fixed timestep specified by the Sample\_interval parameter. The time duration represented by the contents of Input\_waveform\_file and the Golden\_waveform\_file shall be identical and shall be an integer multiple of the wave\_size parameter (the block size; see below).

The number of columns in the contents of the Input\_waveform\_file shall be 1.

The Golden\_IR\_file subparameter is required for Type Statistical and is optional for Type Time\_domain. Its argument is a single string corresponding to a filename present in the same directory as the calling .ibs file. The Golden\_IR\_file contents shall consist of rows of whitespace-separated numeric data representing an impulse response. In the case of Type Statistical, this data will be compared to the output generated by the algorithmic model after the input IR data has been processed by the AMI\_Init function. The values shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9).

The number of values per row (columns) in the Golden\_IR\_file contents shall correspond to the sum of the number of crosstalk channels represented plus 1 for the main communication channel; as the data represents a matrix of impulse responses, a time column is not included. Note that the time values are implied based on the assumption of simulation starting at time = 0 and the use of a fixed timestep specified by the Sample\_interval parameter. The number of whitespace-separated columns in the contents of the Golden\_impulse\_file shall match the number of whitespace-separated columns in the contents of the Input\_IR\_file. The time duration represented by the contents of the Input\_IR\_file and the Golden\_IR\_file shall be identical.

The Golden\_waveform\_file subparameter is required for Type Time\_domain and is prohibited for Type Statistical. The data consists of a single column of analog waveform values representing the concatenated output of an integer number of calls to the associated model’s AMI\_GetWave function if it exists. The values shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9). The time duration shall be an integer multiple of the wave\_size parameter; this is because data will be generated through successive calls to the AMI\_GetWave function, if present, where the size of each block of data passed to and received from the function in each call is specified by wave\_size. The time duration represented by the contents of the Input\_waveform\_file and the Golden\_waveform\_file shall be identical. Note that the time values are implied based on the assumption of simulation starting at time = 0 and the use of a fixed timestep specified by the Sample\_interval parameter.

Note that time domain simulations, and therefore Type Time\_domain comparisons, can be executed if AMI\_GetWave is not present for a given model; in this case, time domain data is generated by the EDA simulation tool by convolving bitstream stimulus information as provided in the Input\_waveform\_file and an impulse response. In the case where Init\_Returns\_Impulse for the model is True, this impulse response is the filtered output from the statistical model represented by the AMI\_Init function. In this situation, providing Golden\_IR\_file data would be strongly recommended for Type Time\_domain, to allow direct examination of the filtered impulse response. Type Time\_domain analysis is possible in cases where Init\_Returns\_Impulse for the model is False. Here, the model does not filter the incoming impulse response, so the bitstream stimulus information as provided in the Input\_waveform\_file would be convolved with the Input\_IR\_file to produce the Golden\_waveform\_file. Golden\_IR\_file data, if present, would be identical to the Input\_IR\_file data.

The Clock\_input\_file subparameter is required if the Direction subparameter value is “Rx”, the associated model uses the Rx\_Use\_Clock\_Input parameter, and the Type subparameter value is “Time\_Domain”. Clock\_input\_file is prohibited for all other combinations of Direction and Type. Its argument is a single string corresponding to a filename present in the same directory as the calling .ibs file. The contents of the Clock\_input\_file consist of multiple rows of numeric data representing a vector of clock information provided to the Rx model; note that only a single value (a single column) is present per row. In cases where Rx\_Use\_Clock\_Input is set to “Times”, multiple blocks of clock information comprise the file, with adjacent blocks separated by a value of -1, and the entire file ending with the -1 value. The number of clock values may differ from the number of UI in each waveform block whose size is given by wave\_size. The vector shall contain increasing clock times (except for the values of -1) which may or may not increase according to a regular interval. With the exception of the values of -1, the values in the file shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9).

In cases where Rx\_Use\_Clock\_Input is set to “Wave”, the contents of the Clock\_input\_file are the input clock waveform to the buffer, where the time duration represented by the contents of the Clock\_input\_file shall match time duration represented by the associated Golden\_waveform\_file. The Clock\_input\_file contents shall use a uniform time step identical to that in the contents of the Golden\_waveform\_file. The values in the file shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9). Only one value per row is permitted.

The Clock\_output\_file subparameter is required if both the Direction subparameter value is “Rx” and the Type subparameter value is “Time\_Domain”. Clock\_output\_file is prohibited for all other combinations of Direction and Type. Its argument is a single string corresponding to a filename present in the same directory as the calling .ibs file. The Clock\_output\_file contents consist of rows of numeric values representing a vector of clock output information from the model. Note that multiple blocks of clock information comprise the file, with adjacent blocks separated by a value of -1. The Clock\_output\_file shall have at least one row; in the case of the contents consisting of a single row, the value shall be -1 (representing a receiver that does not generate clock output). The final value of the Clock\_output\_file time values shall be -1. With the exception of the -1 values, the values in the file shall be floating-point numbers in the standard ANSI “C” notation (e.g., 2.0e-9); while representing the -1 values in floating-point format is permitted, it is not preferred. The number of clock values may differ from the number of UI in each waveform block, whose size is given by wave\_size.

The AMI\_output\_parameters\_file subparameter is optional but strongly recommended. Its argument is a single string corresponding to a filename present in the same directory as the calling .ibs file.

The AMI\_output\_parameters\_file contents consist of rows of whitespace-separated string data, using formatting rules identical to that of the AMI\_parameters\_out string. This data contains the AMI\_parameters\_out string contents that would be generated by the model and passed to the EDA tool. This includes the Usage Out and Usage InOut parameters from the model AMI file contents and their associated values. An example of the structure for Type Statistical is shown below; note that the line feeds shown may be omitted as in the AMI\_parameters\_out string.

(<root name>

 (<Usage Out or InOut parameter name> <value>)

 (<Usage Out or InOut parameter name> <value>)

 (<Usage Out or InOut parameter name> <value>)

 …

)

For Type Time\_domain, because outputs will be generated for each block (instance of wave\_size data being processed by the AMI\_GetWave model, preceded by a single execution of the AMI\_Init function), the AMI\_output\_parameters\_file will consist of groups of parameters which are repeated in each group; the parameters output from the first instance of wave\_size data will be appended to the parameters output from execution of the AMI\_Init function, the parameters output from the second instance of wave\_size data will be appended to the output of the first, the third will be appended to the output of the second, and so on. Each parameter in each block of output parameters shall be preceded by an integer block index value, starting with one and increasing by one with each subsequent block added. This preserves the data for each block allowing correlation of output data as it changes across blocks.

 (<root name>

 (<block index> <Usage Out or InOut parameter name 1> <value>)

 (<block index> <Usage Out or InOut parameter name 2 > <value>)

 (<block index> <Usage Out or InOut parameter name 3> <value>)

 …

 (<block index+1> <Usage Out or InOut parameter name 1> <value>)

 (<block index+1> <Usage Out or InOut parameter name 2> <value>)

 (<block index+1> <Usage Out or InOut parameter name 3> <value>)

 …

)

The Executable\_index parameter is required for all Direction and Type combinations. Its argument is a single non-zero positive integer. This value corresponds to the specific Executable, Executable\_Rx, and/or Executable\_Tx lines under the same [Algorithmic Model] keyword, listing the architecture, compiler, etc. combination used to generate the input and output data provided for the [AMI Test Configuration] keyword. The integer argument to Executable\_index shall not be larger than the number of Executable, Executable\_Rx, and/or Executable\_Tx lines under the same [Algorithmic Model] keyword.

*Other Notes:*  Note that “Statistical” and “Time\_domain” Types are distinct from the capabilities supported by the AMI\_Init/AMI\_Impulse and AMI\_GetWave functions. Type Statistical assumes that statistical processing is performed by the model through the AMI\_Init function; Type Statistical is supported for models regardless of whether they contain or omit the AMI\_GetWave function. AMI\_Init is a required function of all models, but not all models containing AMI\_Init support Statistical analysis. As time domain simulations often involve both AMI\_Init and AMI\_GetWave functions, it is assumed that the EDA tool will execute the appropriate AMI\_Init function for named models using Type Time\_domain before executing AMI\_GetWave, if present. As noted above, Type Time\_domain may be supported for a model only including the AMI\_Init function (and excluding AMI\_GetWave) through the EDA simulation tool convolving the input stimulus bitstream pattern with an impulse response.

The EDA tool is assumed to pass the contents of the Clock\_input\_file into the Rx AMI executable models in cases where the associated model’s AMI parameter file contains “Rx\_Use\_Clock\_Input” values of “Time” or “Wave”.

The structure of the [AMI Test Configuration] keyword is intended to ensure that output can be generated unambiguously by any EDA tool using the Input\_IR\_file, AMI\_input\_parameters\_file, and, where appropriate, Input\_waveform\_file and/or Clock\_input\_file provided, for comparison against the contents of the Golden\_IR\_file, Golden\_waveform\_file, AMI\_output\_parameters\_file, and Clock\_output\_file (where appropriate).

The permitted combinations of [AMI Test Configuration] Direction and Type subparameters with other subparameters are shown in the table below.

Table 3 - [AMI Test Configuration] Subparameter Rules

|  |  |  |  |
| --- | --- | --- | --- |
| **Subparameter** | **Direction** | **Type** | **Required, Optional, or Prohibited** |
| Input\_IR\_file | Rx, Tx | Statistical,Time\_domain | Required |
| AMI\_input\_parameters\_file | Rx, Tx | Statistical,Time\_domain | Required |
| Input\_waveform\_file | Rx, Tx | Statistical | Prohibited  |
| Rx, Tx | Time\_domain | Required |
| Golden\_IR\_file | Rx, Tx | Statistical | Required |
| Rx, Tx | Time\_domain | Optional |
| Golden\_waveform\_file | Rx, Tx | Statistical | Prohibited |
| Rx, Tx | Time\_domain | Required |
| Clock\_input\_file | Rx | Time\_domain | Required if the Rx\_Use\_Clock\_Input parameter is present, Prohibited otherwise |
| Tx | Statistical,Time\_domain | Prohibited |
| Clock\_output\_file | Rx | Time\_domain | Required |
| Tx | Statistical,Time\_domain | Prohibited |
| AMI\_output\_parameters\_file  | Rx, Tx | Statistical,Time\_domain | Optional |
| Executable\_index | Rx, Tx | Statistical,Time\_domain | Required |

*Examples:*

[Algorithmic Model]

Executable\_Tx Solaris\_cc\_64 libtx\_getwave.so tx\_getwave.ami

Executable\_Rx Solaris\_cc\_64 libtx\_getwave.so rx\_getwave.ami

|

Executable\_Tx Windows\_VisualStudio\_64 tx\_getwave.dll tx\_getwave.ami

Executable\_Rx Windows\_VisualStudio\_64 rx\_getwave.dll rx\_getwave.ami

|

| The Executable\_index value below identifies the data for AMI Test

| Configuration as having been collected using the libtx\_getwave.so

| executable.

|

[AMI Test Configuration] Typ\_corner\_statistical

Type Statistical

Direction Tx

Input\_IR\_file four\_tap\_input\_IR.txt

AMI\_input\_parameters\_file four\_tap\_tx\_params\_typ\_stat.txt

Golden\_IR\_file four\_tap\_output\_IR\_typ.txt

AMI\_output\_parameters\_file four\_tap\_output\_params\_typ\_stat.txt

Executable\_index 3

|

| The Executable\_index value below identifies the data for AMI Test

| Configuration as having been collected using the tx\_getwave.dll

| executable.

|

[AMI Test Configuration] Typ\_corner\_time\_domain

Type Time\_domain

Direction Tx

Input\_IR\_file four\_tap\_input\_IR.txt

Input\_waveform\_file four\_tap\_input\_bits.txt

AMI\_input\_parameters\_file four\_tap\_tx\_params\_typ\_TD.txt

Golden\_IR\_file four\_tap\_output\_IR\_typ.txt

Golden\_waveform\_file four\_tap\_output\_wave\_typ\_TD.txt

AMI\_output\_parameters\_file four\_tap\_output\_params\_typ\_TD.txt

Executable\_index 1

| A separate [AMI Test Configuration] file could be provided with a

| different number of crosstalk nodes represented by different

| Input\_IR\_file and Golden\_waveform\_file entries, but with

| otherwise identical parameters and a different name.

[End Algorithmic Model]

**BACKGROUND INFORMATION/HISTORY:**

This BIRD targets correlation of algorithmic models by explicitly defining the stimulus to the model so that the model’s output may be compared to example output supplied by the model maker. The generation and correlation of waveforms based on channels may be addressed in a separate BIRD.

Note that models of Type Statistical will most likely use the parameter Init\_Returns\_Impulse with value True, though this is not required.

Checking of root name as requested in BUG227 for the ibischk7 parser is enabled by this keyword.

The material here was revised thanks to significant input from Fangyi Rao of Keysight Technologies, Ambrish Varma of Cadence Design Systems, and Arpad Muranyi of Siemens EDA.

The text was approved without objection for submission to the IBIS Open Forum by the IBIS Advanced Technical Modeling Task Group on December 19, 2023.