**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**Draft 1**

**BIRD NUMBER:**  *128.2*

**ISSUE TITLE:** *Allow AMI\_parameters\_out to pass AMI\_parameters\_in data on calls to AMI\_GetWave*

**REQUESTOR:**  *Walter Katz, SiSoft; Ambrish Varma, Cadence Design Systems, Inc.*

**DATE SUBMITTED:** *March 11, 2011*

**DATE REVISED:** *August 7, 2014; August 22, 2014*

**DATE ACCEPTED:** Rejected *October 14, 2016*

**ANALYSIS PATH/DATA THAT LED TO SPECIFICATION:**

In order to implement communications from the Rx model to the Tx AMI\_GetWave call this BIRD modifies the use of the argument AMI\_parameters\_out to point to AMI\_parameters\_in data when calling the AMI\_GetWave function.

**ANY OTHER BACKGROUND INFORMATION:**

Background information is provided through numerous e-mails on the IBIS Backchannel Reflector (see freelists.org). Also see presentations given by Walter Katz and Kumar Keshavan at the 2011 DesignCon IBIS Summit.

Replace the following text in Section 10.2.3:

Function: AMI\_GetWave

Required: No

Declaration: long AMI\_GetWave (double \*wave,

 long wave\_size,

 double \*clock\_times,

 char \*\*AMI\_parameters\_out,

 void \*AMI\_memory)

With the following text:

Function: AMI\_GetWave

Required: No

Declaration: long AMI\_GetWave (double \*wave,

 long wave\_size,

 double \*clock\_times,

 char \*\*AMI\_parameters\_inout,

 void \*AMI\_memory)

Replace the following text in Section 10.2.3:

AMI\_parameters\_out

The AMI\_parameters\_out argument is a pointer to a string pointer. Memory for the string is allocated and de-allocated by the algorithmic model. The model returns a pointer to the string as the contents of this argument. The string must be formatted using a tree structure, as described in AMI\_parameters\_in above. The AMI\_GetWave function may use this string to return parameters to the EDA tool.

While the AMI\_parameters\_out argument must always be present in the AMI\_GetWave function call and the EDA tool must always provide a valid (non-zero) address value in it, executable model files are not required to return anything at that address to the EDA tool. For this reason, the EDA tool must also initialize the memory content at that address to zero (null pointer) prior to calling the AMI\_GetWave function, so that after the execution of the function it can determine whether or not the function returned a valid string pointer at that address. If the AMI\_GetWave function does not have any parameters to return to the EDA tool, it must return a pointer at the address provided in this argument to a string which contains nothing but the root name. Note that the root name must always be included in the string.

With the following text:

AMI\_parameters\_inout

The AMI\_parameters\_inout argument is a pointer to a string pointer. On the call to the AMI\_GetWave function (‘In’ mode), AMI\_parameters\_inout shall either contain a NULL pointer, or shall point to a valid 'tree string'. Under certain circumstances (e.g., back-channel communication) the algorithmic model may elect to read the information contained by AMI\_parameters\_inout. Memory for the string is allocated and de-allocated by the EDA tool.

At the end of AMI\_Getwave call, the AMI\_parameters\_inout argument may also return dynamic information and parameters to the EDA tool (‘Out’ mode). Memory for the string passed to EDA tool is allocated and de-allocated by the algorithmic model. The model returns a pointer to the string as the contents of this argument. The string must be formatted using a tree structure, as described in AMI\_parameters\_in above. The AMI\_GetWave function may use this string to return parameters to the EDA tool.

While the AMI\_parameters\_inout argument must always be present in the AMI\_GetWave function call and the EDA tool must always provide a valid (non-zero) address value in it, executable model files are not required to return anything at that address to the EDA tool. For this reason, in the ‘Out’ mode the EDA tool must also initialize the memory content at that address to zero (null pointer) prior to calling the AMI\_GetWave function, so that after the execution of the function it can determine whether or not the function returned a valid string pointer at that address. Initialization of memory content to zero should not occur in the ‘In’ mode. If the AMI\_GetWave function does not have any parameters to return to the EDA tool, it must return a pointer at the address provided in this argument to a string which contains nothing but the root name. Note that the root name must always be included in the string.