**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:**

**ISSUE TITLE:** New GetWave API for Clock Forwarding Modeling

**REQUESTOR:**  Fangyi Rao, Keysight Technologies

Wendem Beyene, Intel

**DATE SUBMITTED:**

**DATE REVISED:**

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

In DDR systems, DQ Rx uses DQS Rx output signal as a forwarded clock to clock the slicer and the DFE. To model such clock forwarding functionality, the DQ Rx model needs two input waveforms, one for DQ, and the other for DQS. This BIRD proposes a new API AMI\_GetWave2 and a new AMI Reserved Parameter GetWave2\_Exists to enable modeling of clock forwading.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. Allow the Rx model to model clock forwarding. |  |

**SUMMARY OF PROPOSED CHANGES:**

Add new API AMI\_GetWave2 and new AMI Reserved Parameter GetWave2\_Exists

**PROPOSED CHANGES:**

*Parameter:* **GetWave2\_Exists**

*Required:* No, and illegal before AMI\_Version 7.2

*Direction:* Rx

*Descriptors:*

Usage:                   Info

Type:                     Boolean

Format:                 Value

Default:                 <Boolean\_literal>

Description:<string>

*Definition:* Tells the EDA tool whether the AMI\_GetWave2 is implemented in this model.

*Usage Rules:* If omitted, the default is “False”. If Init\_Returns\_Impulse is set to “False”, then GetWave\_Exists or GetWave2\_Exists or both shall be set to “True”.

*Other Notes:*

If both GetWave\_Exists and GetWave2\_Exists are “True”, then model consumer (including EDA tool) decides whether to use AMI\_GetWave or AMI\_GetWave2 in simulation.

*Example:*

(GetWave2\_Exists (Usage Info) (Type Boolean) (Default True)

(Description "Model implements AMI\_GetWave2 function")

)

(GetWave2\_Exists (Usage Info) (Type Boolean) (Value True)

(Description "Model implements AMI\_GetWave2 function")

)

*Function:* **AMI\_GetWave2**

*Required:* No

*Declaration:* long AMI\_GetWave2 (double \*wave,

double \*wave\_clk,

long wave\_size,

double \*clock\_times,

char \*\* AMI\_parameters\_out,

void \* AMI\_memory)

*Arguments:*

**wave**

Same as for AMI\_GetWave. See **AMI\_GetWave**.

**wave\_clk**

“wave\_clk” points to a memory location where a uniformly sampled vector of a time domain waveform is stored. The sample spacing and the number of samples of the “wave\_clk” waveform are the same as those of the waveform of the AMI\_GetWave2 argument “wave”, respectively. The waveform pointed to by the “wave\_clk” argument is input. The algorithmic model can use the waveform to control the timing of sampling and equalization on the waveform of the AMI\_GetWave2 argument “wave”. If the sampling timing is controlled by “wave\_clk”, the algorithmic model is expected to derive values in the AMI\_GetWave2 argument “clock\_times” from “wave\_clk”. Note that the algorithmic model can internally process the waveform of “wave\_clk” before it is used.

The EDA tool provides the waveform of “wave\_clk”. In the case of DQ Rx model in a DDR channel, the input waveform of “wave\_clk” is the DLL output waveform of the DQS Rx that provides the forwarded clock to that DQ Rx.

**wave\_size**

Same as for AMI\_GetWave. See **AMI\_GetWave**.

**clock\_times**

Same as for AMI\_GetWave. See **AMI\_GetWave**.

**AMI\_parameters\_out**

Same as for AMI\_GetWave. See **AMI\_GetWave**.

**AMI\_memory**

Same as for AMI\_GetWave. See **AMI\_GetWave**.

**ReturnValue**

Same as for AMI\_GetWave. See **AMI\_GetWave**.

Note:

AMI\_GetWave2 shall be implemented only by Rx models that need to model clock forwarding. In DDR systems, DQ Rx models can implement AMI\_GetWave2. DQS, command-address, conotrol and clock Rx models shall not implement AMI\_GetWave2 because these signals do not use forwarded clock.

**BACKGROUND INFORMATION/HISTORY:**