```
*****************************************************************************
*****************************************************************************

BIRD ID#:
ISSUE TITLE:     IBIS-AMI clock_times Clarification
REQUESTER:       Scott McMorrow, Teraspeed Consulting Group
DATE SUBMITTED:
DATE REVISED:
DATE ACCEPTED BY IBIS OPEN FORUM:

*****************************************************************************
*****************************************************************************
```

STATEMENT OF THE ISSUE:

In the Section, "NOTES ON ALGORITHMIC MODELING INTERFACE AND PROGRAMMING GUIDE", the paragraph describing clock_times has led to inconsistent and incorrect model implementation.

The suggestion is to clarify usage of the parameter consistent with the original intent.

```
*****************************************************************************
```

Replace this text:

```
|
| 3.2.2.3 clock_times
| ===================
|
| Vector to return clock times.  The clock times are referenced to the start
| of the simulation (the first AMI_GetWave call).  The time is always
| greater or equal to zero.  The last clock is indicated by putting a value
| of -1 at the end of clocks for the current wave sample.  The clock_time
| vector is allocated by the EDA platform and is guaranteed to be greater
| than the number of clocks expected during the AMI_GetWave call.  The clock
| times are the times at which clock signal at the output of the clock
| recovery loop crosses the logic threshold. It is to be assumed that the
| input data signal is sampled at exactly one half clock period after a
| clock time.
|
```

```
--------------
```

With the following text with changes noted by "|*" lines:

```
|
| 3.2.2.3 clock_times
| ===================
|
| 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 platform and is guaranteed to be greater than
|* the number of clocks expected during the AMI_GetWave call.  The clock
|* times are exactly bit_time/2 before the input data signal is sampled.
|* The DLL 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 a AMI_GetWave call a single entry of -1 will be returned in the
|* clock_times vector.
|*
```

```
|* 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-strict-monotonic
|* behavior of clock times (including two identical values) should be
|* considered by EDA platform as a DLL failure.
|*
|* Each valid value in the clock_times vector shall be used to sample the
|* output waveform by adding to it bit_time/2, regardless 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 insure 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 UI 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 1st 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.
|
```

****

ANALYSIS PATH/DATA THAT LED TO SPECIFICATION

Additional notes regarding correct clock_times usage have been included as part of this BIRD, distilled from discussions on the ibis-macro reflector.

Additional notes regarding clock_times

The original specification of clock times, and any implementations flowing from this, are technically wrong, resulting in a difference between the actual sample point in a receiver, and the calculated sample point. However, all known implementations of AMI dll models calculate the receiver sample point, and then subtract 1/2 of the bit_time or symbol UI to derive the clock_times returned to the EDA platform. The EDA platform can then add 1/2 of the bit_time to the clock_times to calculate the original sample point. This BIRD clarifies what has become the accepted usage.

If the receiver does not contain a CDR, as evidenced by lack of the output of valid clock times before AMI_Close, then the EDA platform is at liberty to present the received eye pattern in any way it sees fit. If the receiver does contain a CDR, it will eventually output valid clock_times, and the center of each eye period is defined to be clock_times + bit_time/2. The EDA platform should calculate the center of each symbol eye at each individual clock tick, to track fluctuations of the CDR period jitter. If a clock tick is missing from any symbol period, the EDA platform may calculate the missing tick based on a "golden CDR" or nominal clock period calculation for the purpose of eye display. If a "golden CDR" is used by the EDA platform for missing tick clock recovery, leading waveforms without subsequent clock ticks may be centered with the EDA clock and accumulated. However, all missing ticks should be considered as received data errors.

There is no requirement that clock times should be integer multiples of the sample interval (or time step used to represent the waveforms). There is also no requirement that there be a relationship between clock_times in the primary channel, and any additional waveform components in the wave vector, such as crosstalk. Crosstalk channels shall not be constrained to any timing relationship to the primary channel, or to the clock_times vector.

****

ANY OTHER BACKGROUND INFORMATION:

This is an editorial correction to clarify the usage of clock_times.

****