

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 time is always  
|* greater or equal to zero. 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 the clock signal at the output of the clock  
|* recovery loop crosses the logic threshold in a full data rate CDR clocking  
|* system (i.e clock period equals UI). The effective receiver sampling  
|* point is equal to the clock_times plus 1/2 the instantaneous UI period,  
|* which is the midpoint of two adjacent clock ticks. The last  
|* valid clock of the current GetWave call is indicated  
|* by placing -1 after the last valid clock in the clock_time vector.  
|*  
|* The clock ticks represented by clock times should be strictly monotonic,  
| clock_times bird_9.txt page 1
```

|* both within the clock_times array returned from a single call to GetWave
|* and between successive calls to GetWave. That is, within a given clock_times
|* array each successive valid value is greater than the value that preceded it,
|* and the first valid value from a given call to GetWave must be greater than
|* the last valid value from the preceding call to GetWave. Any non-strict-monotonic
|* behavior of clock times (including two identical values) should be considered
|* by EDA platform as a DLL failure and should lead to simulation termination
|* with respective message.
|*
|* Each valid pair of values in the clock_times array shall be used to sample the output
|* waveform as previously described, regardless whether that waveform sample occurs
|* in the waveform segment being returned by the current call to GetWave, or the
|* waveform segment to be returned by the next 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 is
|* 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.
|*
|* 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.
|

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 previous specification was vague with respect to the meaning of one half clock period. As a result, implementations have interpreted this as being one half of a Unit Interval (UI), which is erroneous. In reality, the CDR clock is generally used to recover a stable clock from the data stream and use that as the basis for sampling (latching) the incoming waveform. Since the CDR locks on differential crossings at the edge of an eye, the sampler in a full rate CDR architecture uses the opposite edge of the recovered clock to generate the sample point. In a locked system under steady state conditions, the mean of the clock period is also the nominal period (UI) of the data stream. However, even under steady state conditions, random fluctuations in the CDR loop will cause clock deviation that can be seen as jitter, along with a subsequent shift in sample point. Under these conditions, the clock period (and sample point) will vary over time. The instantaneous clock period can be significantly less than or greater than the nominal period.
- * The current 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. Our first order calculations at Teraspeed consulting indicate that this is at the minimum an error of $\pm 5\%$ of a nominal UI. Depending on the CDR, and the way it responds to stressful data patterns, this can be larger. For any particular time slice, this means

that the center of the eye for a given single bit of data can be incorrect if the nominal UI is used for calculation of the receiver sample point, causing a skew in any derived statistics.

ANY OTHER BACKGROUND INFORMATION:

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