Sampling of analog quantities within SV – Jim Lear, 2009-07-28

The key to implement useful mixed-signal assertions is to make the appropriate data available. There are a few different types of data collections that can serve different purposes and have different costs.

1. Scalar asynchronous – similar to the existing bind proposal
2. Scalar synchronous – high precision on demand analog quantities
3. Vector asynchronous – a data set similar to that a waverform viewer would use
4. Vector synchronous – a high accuracy periodic waveform for spectral analysis

Scalar Asynchronous

In this type of observation, the accuracy of the observed quantity is not of the utmost concern. Instead, the intention is to make the observation as unobtrusive to the performance of the simulation as possible. The analog engine can calculate time points as desired, including time points that may be later than the current SV time. The SV observation of analog values will be from the last analog time point that precedes the current SV time, or some interpolated value. This observation will not alter the analog engine behavior in any way. It should look similar to the Bind proposal.

For example, verifying the enabling and disabling of a voltage regulator may require only a casual observation of the output of the regulator. A few micro Volts of error may not be of concern when the observation is knowingly conducted before the complete settling of the system.

Scalar Synchronous

Scalar synchronous sampling will force synchronization between the analog and digital engines at the demand of the digital engine to ensure the measured value is accurate at the precise time required. This is a high cost sampling in terms of performance, but produces a precise analog value. For example, extracting the RC time constant requires precision in the both the time dimension as well as the value dimension.

Vector Asynchronous

This is a finite vector of time-value pairs. This is simply a record of analog time points that precede the digital time point. For memory conservation, a limit may be required for the maximum number of samples or the maximum time contained within the vector. The collection of these time points place no constraints on the analog engine time step algorithms.

These vectors will be useful to measure time-domain behavior of analog devices, such as slew rates, peak voltages, glitches, etc.

Vector Synchronous

This vector will consist only of time-value pairs that are taken at precise periodic time points. This will be useful for spectral analysis of the transient response. Not all analog time points need to be recorded, but time points must be forced at periodic intervals.

The sampling period of this buffer must be controllable by the SV. In some cases the bandwidth of testcases will vary or spectral measurements may be unnecessary. Because there may be performance implications, the ability to disable the periodic samples should be provided. Conceivably, a very long sampling period could emulate the disabling function.

Comments

Ken Bakalar has suggested a global event be visible within SV that fires when an analog time point has occurred. A concern was raised that the SV clock precision may be so gross that multiple analog events may appear to SV to occur at the same time point. Perhaps a global high precision (double precision?) variable indicating the time of the analog time point, would allow the implementation of the vector asynchronous form of sampling. There may have

Synchronous sampling will require additional capabilities beyond the global analog events.

I would also like the ability to control crossing detection from within SV. The random configuration of the analog devices will cause thresholds to potentially vary within testcases. This might be implemented by passing a real value through an SV port to a Verilog-AMS cross operator. I'm not sure if this is legal.

Dejan Nickovic believes this should not be a part of the standard. I agree, however, I don't believe the language has the capabilities to implement the above requirements in user written code. Some language modifications may be required.

Finally, I think the sample buffers should be implemented in a layer outside of the language standard. It should be analogous to the stdio library for C. The implementation of such a library may require some language modifications, though, such as Ken's global analog event.