Scheduling Semantics (v1.03 3/21/01)

A Verilog-AMS simulation consists of a number of (analog and digital) processes communicating via events, shared memory and conservative nodes. Analog processes which share conservative nodes are "solved" jointly and can be viewed as a "macro" process, there may be any number "macro" processes, and it is left up to the implementation whether it solves them in a single matrix, multiple matrices or uses other techniques but it should abide by the accuracy stipulated in the disciplines and analog functions.

Concurrency

Most (current) simulators are single-threaded in execution, meaning that although the semantics of Verilog-AMS imply processes are active concurrently, the reality is that they are not. If an implementation is genuinely multithreaded, it should not evaluate processes that directly share memory concurrently as there are no data locking semantics in Verilog-AMS.

Analog Macro Process Semantics

An analog "macro" process interacts with other processes through events and shared variables. When it is initially activated, it will attempt to predicate a potential "solution" at a a future time (the "acceptance time") and will store (but not communicate) values[1] for all nodes at that time, and will schedule a "wake up" event for the acceptance time, the process is then inactive until woken up or it receives an event from another process. If it is woken up by its own "wake up" event it calculates a new solution point, acceptance time etc. and deactivates. If it is woken up prior to acceptance time by an event that disturbs its current solution it will cancel its own "wake up" event, accept at the wake-up time and recalculate its solution and schedule a new "wake up" event for the new acceptance time.
If the analog process identifies potential "crossings" then it will schedule it's wake-up event for the time of the first such event rather than the acceptance time. If the analog process is woken by such a crossing event it will communicate any related events at that time and de-activate, rescheduling it's wake-up for the next crossing or acceptance; events to external process generated from analog events are not communicated until the global simulation time reaches the time of the analog event.
If the time to acceptance is infinite then no "wake up" event needs to be scheduled[2].
As with digital processes, analog processes are insensitive to changes in variables i.e. a change in a variable does not force re-evaluation of the process, neither are they implicitly sensitive to digital signals used in procedural code, only changes in signals in '@' expressions will trigger re-evaluation prior to scheduled wake-up. The statement '@*;'[3] in an analog process will force it to re-evaluate on any digital signal used in the block being changed.

External Access to Analog Macro Process Data

Values of nodes in an analog process are piecewise-linear, the value returned is the linear interpolation for the time of the request between the last accepted solution point and the next (potential) solution point. Other values are given as the value at the last process de-activation.

---

1. Or derivatives w.r.t. time used to calculate the values.
2. The case when all derivatives are zero.
3. Verilog 2000.

PWL/Digital Boundary Event Scheduling

Events created for digital processes in the analog domain are handled immediately by their receiving processes. The time at which these events occur is not bound to any multiple of the digital kernel's time resolution ("tick"), but are handled as a seperate "delta" cycle. Any zero-delay digital events created by such an event that return to the analog domain carry the same timestamp as the original event from the analog domain. Events scheduled with a non-zero delay in digital processes activated by analog events will be "snapped" to the nearest multiple of the digital kernel time resolution; no real-value time scheduling is performed by a Verilog digital kernel.