

SV-DC – Roadmap

contribution by Achim Bauer (exl-modeling), 17th August 2010

abbreviations:

CM connect module
oomr out-of-module-reference
rv real-value

Executive Summary

...

Motivation and Use Cases

Following is a “golden” list of real-value scenarios that are of major practical importance. A methodology or EDA flow, that can handle these features, i.m.o. might be considered as fully capable for discrete simulation (validation and verification) of complex mixed (real and digital values) systems.

- multiple rv-ports drive a common net
→ resolve multi-driver nets
- resolve a (mixed) net that is connected with both real and digital ports
→ deal with type incompatibilities, e.g. due to model “plug & play” or netlist composition
- assert compatibility of power/supply domains around a net
→ e.g. to verify missing/inappropriate level shifters
- derive a dense time function for a net’s slewed/delayed transition
→ if the inertial behavior of nets is a critical part of a system’s operation
- ability to handle/simulate scattered/distributed analog components
→ to “digest” the lumped Rs and Cs that often show up in complex SoC netlists
- oomr rv-net contribution or overdrive/force from the testbench
→ full controllability, directed verification to enhance coverage
- assertions should be able to read the rv-net’s state via oomr or bind
→ e.g. read linear coefficients of net’s dense time function to perform a cross-like operation
- using the sophisticated rv-resolution also for nets with pure digital drivers and loads
→ simulate a supply/RC-sensitive timing of particular digital components

Requirements

Following a few aspects/features shall be mentioned without further explanation/consideration.

resolution process

The resolution process read/write/exchanges information with all modules connected to the net, evaluates the net's state, passes appropriate digital values to digital loads and schedules the next time step to be simulated. It shall take into account all the hierarchies that are part of the net. Access functions (system functions) like \$driving_strength, \$loading_strength, \$driver_count, \$driver_state, @(driver_update()), \$struct_entry(n)... might be specified.

real↔digital conversion

Do we need automated CM insertion within a discrete case for real↔digital conversion?
Or might "mixed" discrete digital/real-nets be resolved without CM placement?
Might connect modules hide/block (struct) information required for a clean resolution?

struct type

struct nets (entries: logic, real, integer, string, events), arrays/busses/vectors of structs containing variable values or properties e.g. for actual voltage or current, driving (R) and loading (C) strength, nominal supply voltage, ask & eval-events, signal identifier...
Structs should represent a bidirectional interface for information exchange and synchronization between the net/node and its contributors.

state of the rv-net

net state may contain quasi-continuous $V(t)$ signal function, tolerances, message strings (e.g. warnings), actual time constant ($\sim RC$), next time-point for reevaluation, pointers or dynamic arrays (e.g. a series of measurements)...

oomr

oomr read and write of the resolved nets, e.g. to access state information, like $V(t)$, impedance, or to trigger an evaluation

Timeline and Vision

...