Introducing Xsim

Overview

Xsim is designed as a true single-engine mixed-mode simulator with a unified structure. It is based on electrical (circuit-level) matrix solution, with event-driven (logic-level) and functional (behavioral-level) techniques as accelerations for digital and analog circuits, respectively. The key to its automatic circuit partitioning and dynamic mode switching capabilities is based on the *subcircuit expansion* approach in which the higher-level models correspond directly to their equivalent lower-level subcircuits.

Features

- *Electrical (circuit-level) simulation*
  - Efficient vector sparse-matrix solution for solving bordered block diagonal (BBD) matrix as a result of subcircuit expansion.
  - Implemented device models: diode; MOS level 1, 2, 3; BJT Gummel–Poon; AIM HFET, MESFET, HBT.
  - A unified composite bipolar-MOS (BiMOS) model for deep submicron CMOS.
  - SPICE netlist compatible.
  - Device model bypass and incremental matrix update.
- *Digital (gate-level) simulation*
  - Two representations of a logic gate (**`U`** card): *behavioral* (Boolean, **`.model`** card) and *structural* (subcircuit, **`.subckt`** card).
  - *Dynamic delay model* based on user-netlist dependent fanout and run-time dependent input transition time:

    $$t_d = t_{di}(T_i, N_i, C_L) + R_{eff}(T_i, N_i, C_L)C_L$$

    where t_d is the gate delay (including low-to-high and high-to-low delays); t_{di} is the intrinsic (unloaded) delay and R_{eff} is the effective resistance, which are a function of the input transition time T_i and the number of triggered inputs N_i determined at run time; C_L is the fanout capacitance determined from user netlist.
  - Logic parameters (input and output capacitances and intrinsic delays) are extracted from the low-level subcircuit, either by pre-configuration or during the simulation.
- *Analog (behavioral-level) simulation*
  - Built-in behavioral functionals in the SPICE netlist format.
- *Mixed-signal multi-level simulation*
  - *Mixed-signal*: analog and digital circuits can be freely mixed.
  - *Mixed-level*: gate←circuit→behavioral, with gate and behavioral levels as abstractions.
  - *Mixed-method*: sparse-matrix, relaxation, event-driven, selective-trace.
  - *Mixed-technology*: different logic families can be mixed in the same circuit, and the interface will be automatically modeled by full subcircuit to ensure consistency.
- *Automatic circuit partitioning*
  - Partitioning of analog and digital blocks is done automatically by the simulator.
  - The partitioning can change at run time.
- *Dynamic mode switching*
  - Digital gates can be simulated in "**`analog`**" mode (equivalent to replacing all **`U`** cards by **`X`** cards) or "**`digital`**" mode (Boolean plus dynamically-extracted delays).
  - When the mode is set to "**`mixed`**", this switching happens for each gate at run time, based on the input signal quality.
- *Hierarchical probing*
  - Node voltages and branch currents at any level can be probed.
  - Device internal probes (e.g., BJT internal V_{BE}, MOS substrate diode current) are also available.

Applications

- *Large digital circuits*: Using full "**`digital`**" mode for logic synthesis with vendor-supplied delay parameters, and then, subcircuit model for back-annotated timing verification.

- *Mixed-signal interface circuits*: Using "**`mixed`**" mode for data-converter systems with large ratio of digital/analog circuitry.

- *PCB-level system simulation*: Large systems with mixed digital (including mixed logic families) and analog functional blocks.

Mixed-Signal Multi-Level Simulation of VLSI Circuits

— An Implicit mixed-mode solution

Introduction

An emerging area of growth, in the overall chip market, is for mixed-signal application-specific integrated circuits (ASIC's). A new generation of mixed-mode simulators has evolved to apply different algorithms to different circuit blocks. However, most of them are either "glued" (two simulators running concurrently) or "explicit" (the designer must specify which components are digital and which are analog) — the burden of selecting the most appropriate algorithm for each block rests with the designer. Requiring the designer to specify in advance which components are digital and which are analog is like asking him to simulate the circuit prior to running the simulation. The information required to do this is often the very information the designer is seeking from the simulation.

Binding a digital and an analog simulator into a "solution" is obviously no solution. Relying on the designer to partition the circuits is not practical either. What a designer needs is a truly "implicit" single-engine simulator that can simulate a circuit at different levels of abstraction, with "digital speed" whenever possible and "analog accuracy" wherever necessary.

Mixed-Mode Simulator

To electrons, there are no "analog" or "digital" parts — they are all simply electronic circuits following the same laws of physics in their operations. In fact, digital parts are simply analog parts that are overdriven. But why do we need a mixed-mode circuit simulator? The answer is: If analog simulators were fast enough, no one would need digital simulators. Unfortunately, they are not, and they will never be.

In addition, complex electronic systems are designed in a hierarchical manner, so they should also be simulated and verified at different levels of abstraction. For example, in the analog domain, large analog subsystems (such as op-amps, filters,

oscillators, etc.) are frequently used as building blocks. They should be simulated using functional (or behavioral) models if simulation speed is of major concern.

Defining terms

The term "mixed-mode" has been used in a very broad sense, which implies one or more of the following interpretations:

Mixed-signal: Analog and digital circuitry with distinctively different waveforms (voltages vs. logic states).

Mixed-level: Same circuit described at different levels of abstraction (e.g., circuit, switch, gate, behavioral, etc.).

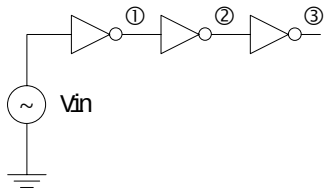
Mixed-method: Different simulation algorithms (e.g., matrix, integration, relaxation, event driven, selective trace, etc.) applied to different parts of a circuit.

Mixed-precision: Multiple precision (e.g., 2-, 3-, 6-, or 9-state logic) used at different stages of a design.

Mixed-technology: Different logic families (e.g., NMOS, CMOS, TTL, ECL, etc.) mixed in the same circuit description.

A trivial example

The following "trivial" example shows a string of inverters driven by an analog stimulus. Suppose that the source is a sine function with increasing amplitude.



Question: What are the waveforms at nodes ①, ②, and ③?

Trivial answer: Use full-circuit simulation such as SPICE.

The "digital" solution: For event-driven gate level model, the initial waveform at node 1 may be wrong, and results from that point on would be meaningless.

The mixed-mode scenario: The following two questions should be answered:

- *Which* gate(s) should be simulated as "analog"? (*automatic* circuit partitioning)
- *When* should it (they) be simulated as "analog"? (*dynamic* mode switching)

A wish list

- *Simulator architecture*
  - A common paradigm and a unified data structure should be used at all levels of abstraction.
  - A consistent representation for signals over all simulation levels as well as representation of time should be defined.
  - A flexible and extensible structure so that algorithms and device models can be added or removed easily.
- *Choice of simulation levels*
  - Ideally, one would prefer to combine all the levels into one simulator: electrical, switch, gate, behavioral, etc.
  - It is desirable that the simulator can make intelligent choice of appropriate simulation levels; and the user can specify his own choice as well.
- *Automatic circuit partitioning*
  - The simulator (not the user) should be able to determine which portions of the circuit must be simulated at the most detailed level and which portions will profit from simulation at a higher level without noticeable loss of accuracy.
  - It is also nice if the user can designate circuit blocks to be simulated at a level of his choice.
  - Higher-level model should correspond to its lower-level equivalent, and its parameters should be extracted automatically and dynamically from the lower-level model.
- *Dynamic mode switching*
  - The simulator should be able to switch any elements or blocks from a lower level to a higher level when it is appropriate, and switch back when necessary.
  - This switching should happen automatically and, more importantly, dynamically during a simulation.
- *Hierarchical probing of variables*
  - Circuit variables (voltages and currents) should be obtainable at each level of abstraction.
  - Both analog (waveforms) and digital (logic states) signals should be available at request.
- *User interface*
  - It should accommodate all the levels on the same schematic capture.
  - It should handle different waveforms (analog or digital) for post-processing.
  - Adding new components (macromodels, behavioral models) should be simple.
  - A consistent netlist format should be used at all levels of abstraction.

Myth or reality?

The challenge in mixed-mode simulation is not in simulating different types of electronic circuitry, but in combining distinctively different algorithms in a unified structure. There are simulators that work efficiently at different levels of abstraction, such as behavioral modeling language, gate-level logic simulator, switch-level timing analyzer, circuit-level simulator. The question is: Is it possible to combine all these into a single-engine simulator?

The items in the above Wish List sound too good to be true. The simulator introduced next (**Xsim**) shows that it is not a myth but a reality.

References

[1] X. Zhou, "Mixed-Signal Multi-Level Circuit Simulation: An Implicit Mixed-Mode Solution," (*Invited Paper*), *Proc. VSDT2000*, Indian Institute of Technology, Bombay, India, Dec. 10-11, 2000, pp. 10–15.

[2] T. Tang and X. Zhou, "Multi-Level Digital/Mixed-Signal Simulation with Automatic Circuit Partition and Dynamic Delay Calculation," *JMSM*, vol. 1, no. 2, pp. 83–89, Dec. 1999.

[3] T. Tang and X. Zhou, "A Dynamic Timing Delay for Accurate Gate-Level Circuit Simulation," *Proc. MWSCAS-96*, Iowa, USA, Aug. 1996, pp. 325–327.

For more information, please contact:
Dr. Xing Zhou
School of Electrical & Electronic Engineering
Nanyang Technological University, Singapore 639798
Tel: (65) 6790-4532. *Fax*: (65) 6791-2687.
Email: exzhou@ntu.edu.sg
http://www.ntu.edu.sg/home/exzhou/Research/XSIM/