Investigation of Formal Verification for Self-Healing Analog/RF Systems

A. Rajhans, M. Althoff, B. Krogh, L. Pileggi, X. Li
Carnegie Mellon University

Motivation for formal verification

<table>
<thead>
<tr>
<th>ANALYSIS TASK</th>
<th>ANALYSIS METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of a single operating point</td>
<td>Simulation</td>
</tr>
<tr>
<td>Analysis of the correctness of design</td>
<td>Simulate one particular behavior</td>
</tr>
<tr>
<td>Analysis with process variations</td>
<td>Monte Carlo simulation</td>
</tr>
<tr>
<td>Analysis robustness against process variations</td>
<td>Simulate many behaviors</td>
</tr>
<tr>
<td>Analysis over complete post-disturbance tuning range</td>
<td>Formal verification</td>
</tr>
</tbody>
</table>

How we can use formal verification

Verification-aided design of self-tuned components

- e.g., self-tuning PLL
- e.g., self-tuned locking

Decompose the locking specification into two parts

- Transient verification
  - Discrete-time verification of whether all behaviors enter the invariant target
- Invariant verification
  - Identify regions of state space that guarantee staying in the invariant
  - This becomes a target set for transient verification

Simulate the behavioral model

- Phenomenological model: For never smoldering, one between the two states, it’s a simple twoiarface scenario. Turn off.

Verification approach

- Decompose the locking specification into two parts
  - Transient verification
    - Discrete-time verification of whether all behaviors enter the invariant target
  - Invariant verification
    - Identify regions of state space that guarantee staying in the invariant
    - This becomes a target set for transient verification

Challenges in reachability analysis

- Hybrid dynamics:
  - Verification complexity: exponential in the number of continuous state variables for polyhedral computations
  - With zonotopes (polyhedra with special structure) computations*, there’s more speedup in continuous reachability (social complexity), but complexity still exponential for hybrid dynamics
- Very long transient:
  - Thousands of discrete transitions; over-approximation becomes less accurate with each discrete transition
- Liveness specification (locking):
  - Need to verify infinite (infinite-time) behavior
  - Over-approximation grows with time

Invariant verification: Forward-backward iteration

- PHAVer (Polyhedral Hybrid Automaton Verifier)
  - Uses exact rational arithmetic up to arbitrary precision.
  - Supports forward and backward reachability computation
  - However, needs to over-approximate linear dynamics by (even simpler) piecewise constant bounds on derivatives.
  - Reachability analysis with cycle unwrapped

Target application: self-healing PLL

- Verify locking behavior over
  - arbitrary initial states
  - range of parameter values
  - with self-healing logic

Behavioral model:
- Continuous state variables: \( p_0, p_1 \)
- Discrete switching due to charge pump

Invariant verification using PHAVer*

- Completion of invariant and transient verification
- More detailed model including
  - Charge pump saturation
  - VG0 non-linearity
- Compositional verification: digital-analog decoupling

Next Steps

- Component verification: digital-analog decoupling
- More detailed model including
  - Charge pump saturation
  - VG0 non-linearity

2010 Annual Review