Formal Verification of C-element Circuits

Chao Yan*, Florent Ouchet†, Laurent Fesquet†, Katell Morin-Allory†

* Department of Computer Science
University of British Columbia, Canada
chaoyan@cs.ubc.ca

† TIMA Laboratory
Grenoble INP, UJF, CNRS, France
firstname.lastname@imag.fr
Outline

- Motivation
- The Full-Buffer Circuit
- COHO
- Reachability Computations
- Results
- Conclusion & Future Work
Circuit-Level Verification

To Validate Behaviors of
- Digital Circuits: deep submicron design, leakage currents, delay, ...
- Asynchronous Circuits: continuous time, non-uniform threshold voltages
- Analog Circuits: e.g., cellphones, cameras, GPS, MEMS, etc.

To Find Bugs
- Circuit-level bugs account for large percentage of critical bugs
  - Synchronous, digital design has become relatively low error
  - Asynchronous and analog designs lack systematic design flow and methodologies
  - Asynchronous and analog designs rely on designers’ intuition and expertise
- Noteworthy circuit bugs:
  - Intel Sandy Bridge Chipset
  - iPhone4

Problem Statement:

Circuit Verification Using Circuit-Level Models
Formal Verification

- Problems of Simulation
  - SPICE is still the main validation tool.
  - Incomplete coverage: difficult to cover all corner cases.
  - Inexact models, incomplete coverage of operating conditions and input signals.
  - Expensive: initialization, analog-digital interface.
  - Miss critical bugs: millions/billions of $$\$$!

- Role of Formal Methods
  - Provide coverage that simulation does not.
  - Provide a precise way to write specifications.
  - Verifying real circuits enables us to evaluate these specifications.

- Formal Verification: Gate-level $\rightarrow$ Circuit-Level
Full-Buffer Circuit

- From ST Microelectronics
- Quasi-delay insensitive circuit, dual-rail encoded
- Four-phase handshake protocol
  1. the receiver is ready to accept data
  2. the sender emits data through the channel
  3. the receiver acknowledge this data
  4. the sender removes the data
Does It Work for Low Power Design?

- Delay insensitive does not mean slope insensitive [ASYNC 2010].
- Erroneous switching behavior because of slow input transition.
- May generate spurious "data" if $V_{TH} < V_{TL}$ in the C-element.
Does It Work for Low Power Design?

- Delay insensitive does not mean slope insensitive [ASYNC 2010].
- Capacitance of internal node matters.
  - One internal node has a much larger/smaller capacitance than others.
  - Not only increases the propagation delay but can also stall the handshaking cycling.
Does It Work for Low Power Design?

- Delay insensitive does not mean slope insensitive [ASYNC 2010].
- Capacitance of internal node matters.
- Sufficient condition to ensure the handshake protocol.
  - Simulation cannot fully guarantee correctness.
  - How to ensure a given design works?
Outline

- Motivation
- The Full-Buffer Circuit
- СОНО
- Reachability Computations
- Results
- Conclusion & Future Work
Verification as Reachability

- Phase-space view of circuit behavior
  - Waveforms $\rightarrow$ phase-space (reachable regions)
  - An inverter example

- Circuit verification based on reachability analysis
  - Construct a mathematical model of a given circuit
  - Compute all forward reachable regions from given initial states
  - Check properties on reachable regions: safety properties, liveness properties.

- Challenges of reachability analysis
  - Representation of high-dimensional, non-convex regions
  - Solve nonlinear dynamics efficiently
**Соно**: Our Approach

- **Соно**
  - Modeling a circuit by ODEs automatically
  - Representing and manipulating high dimensional space: *projectagon*
  - Solving dynamic systems: *linear differential inclusions*.
  - All approximations overapproximate the reachable space:
    - **Соно** is *sound* for verifying safety properties.
    - False negatives are possible.
  - [http://coho.sourceforge.net](http://coho.sourceforge.net)

---

ASYNC 2011    29th April, 2011    Formal Verification of C-element Circuits – p.9/22
Circuit Model

- Transistors modeled as voltage controlled current sources.
- Current function
  - Obtained by tabulated data from HSPICE simulations.
  - Other circuit-level models are also supported
- Construct ODEs automatically
  - Compute current of each transistor: $i_1 = f(V)$. Likewise for $i_2$.
  - Compute current of each node: $i_c = i_1 + i_2$
  - $\dot{V} = C^{-1} \cdot I = F(V)$
Projectagons

- COHO projects high dimensional polyhedron onto two-dimensional subspaces.
- Projectagons are efficiently manipulated using two-dimensional geometry computation algorithms.
- Projectagon faces correspond to projection polygon edges.
Solving dynamic systems

- Approximate the nonlinear ODEs by *linear differential inclusions*:

\[
A \begin{bmatrix} \dot{v} \\ \text{in} \end{bmatrix} + b - u \leq \dot{v} \leq A \begin{bmatrix} \dot{v} \\ \text{in} \end{bmatrix} + b + u
\]

- Efficient
  - Compute forward reachable region using the Maximum Principle
  - Only matrix computations, no integration

- Accurate
  - Work on each face rather than the whole projectagon
Basic Step of COHO

- A bounding projectagon is obtained by moving each face forward in time.
- The advanced face is projected onto two-dimensional subspaces to maintain the structure of projectagon.
- Computation continues until no new reachable region found.
Reachable Regions of C-elements

- Partition the circuit to C-elements, NORs, and inverters.
- Compute reachable regions of C-element circuits.
  - How to model input transitions?
  - How to perform reachability computation automatically?
  - How to improve performance?
- Verify properties based on Brockett’s annulus.
Specifying an Input Signal

- Brockett’s Annulus

Region 1 represents a logical low signal.
Region 2 represents a monotonically rising signal.
Region 3 represents a logical high signal.
Region 4 represents a monotonically falling signal.
Brockett’s annulus allows entire families of signals to be specified.

- Reachability Computation
  - Map continuous trajectories to discrete sequences.
  - Minimum dwell time.
Specifying Multiple Inputs

- Consider All Input Transitions
  - All combinations: $4^d$ states.
  - How to record the time of each trajectory?

- Over-approximation
  - Allow trajectories to leave region 1(3) a little earlier.
  - Two types of transitions.
Reachability Computation

- Tedious and Error-prone
  - Write codes for each circuit verification
  - Computation is expensive: difficult to find stupid bugs

- A Standard Interface
  - Based on hybrid automata
  - Templates: one-input circuits, two-input circuits . . .
  - Check point

- Optimize performance and accuracy
  - Slicing: partition region $2(4)$ into smaller intervals
  - Assume-guarantee: divide one automaton into several smaller ones
  - Macro-models: simplified models for speeding up computations
Outline

- Motivation
- The Full-Buffer Circuit
- \text{ОНО}
- Reachability Computations
- Results
- Conclusion & Future Work
Results of C-elements

- Verified Properties
  - Established an invariant set
  - C-element’s analog behavior conforms to the digital abstraction
  - The output signal satisfies the same Brockett’s annulus with the input
  - Verification failed if the keeper inverter is too large or small

- Performance
Results of C-elements

- Verified Properties
  - Established an invariant set
  - C-element’s analog behavior conforms to the digital abstraction
  - The output signal satisfies the same Brockett’s annulus with the input
  - Verification failed if the keeper inverter is too large or small

- Performance
Results of C-elements

- Verified Properties
- Performance
  - Still expensive (1-2 days)
  - Run-time depends on the number of dimensions
  - 1-2G bytes memory
Verifying the Full-Buffer Circuit

- Verification Based on Brockett’s Annulus
  - Apply the same Brockett’s annulus to specify inputs of C-elements, NORs, and inverters.
  - Verify outputs of C-elements, NORs and inverters satisfy the same Brockett's annulus.
  - All internal signals oscillate properly.
  - The full-buffer circuit guarantees the handshake protocol.
  - Can be applied to verify other similar circuits.

- Verification of “Large” Circuits
  - Decomposition helps!
  - Brockett’s annuli provide a specification method to higher level.
Conclusions

- Simulation cannot find all potential bugs
  - Extra data when slope is slow.
  - Non-oscillation when capacitance is huge/tiny.

- Reachability analysis can formally verify circuit-level properties.
  - An efficient and robust reachability computation tool.
  - Verified practical circuits.

- Larger circuits can be verified by combining reachability analysis and other methods.
  - Reachability computation of high-dimensional systems is expensive.
  - Reduce dimensionality by decomposition.
  - Partition computations by assume-guarantee strategy.
Future Work

- Circuit-level formal verification
  - Leveraging Designer’s Intent [Kim, Jeeradit, et al.]
  - Parameterized verification
  - Point verification
  - Static analysis

- Verification of Mixed Signal Circuits
  - Digital-analog interface
  - Small-“a” analog design

- Specification

- Parallel Computing

Thank You!