Circuit Level Verification of a High-Speed Toggle

Chao Yan & Mark Greenstreet

University of British Columbia
Overview

- Motivation
- Coho
  - Projection Based Reachability Analysis
  - Numerical Issues
- Verification Example
  - Toggle Circuit
  - Toggle Specification
  - Verification Using Coho

**Formal Verification of Digital Circuits Using SPICE-Level Models is Possible.**
Motivation

Design Flow

Typical Design Flow

Coding

Cell Lib.

Synthesis

equations

Tech Map

netlist

Place & Route

layout

Extract

annotated netlist

OK?

N

Y

done

FMCAD 2007 — 14 Nov 2007 — Circuit Level Verification of a High-Speed Toggle — p.3/22
Motivation

Design Flow

Typical Design Flow

Coding

Synthesis

Tech Map

Place & Route

Extract

Cell Lib.

Add a new cell

Manually Check

Simulate (SPICE)

Layout

Y done?

N reject

~ one month
Motivation

Typical Design Flow

Coding

Synthesis

equations

Tech Map

netlist

Place & Route

Layout

Extract

annotated netlist

OK?

OK?

Y

N

Y

N

Y

N

N

Y

N

Y

one month

Coho

Automatic Verification

Add a new cell

Manually Check

Simulate (SPICE)

Y

N

done?

time?

Motivation

- **Design Flow**
- **Similar Problems**
  - crosstalk analysis
  - power noise problems
  - leaky transistors
  - mixed-signal design

---

**Typical Design Flow**

- Coding
  - RTL
  - Synthesis
    - equations
  - Tech Map
    - netlist
    - layout
  - Place & Route
    - layout
  - Extract
    - annotated netlist
  - OK?

**Add a new cell**

- N
- automatic verification

**Manual Check**

- Y
- N

**Layout**

- Y
- N

**Coho**

- Y
- N

**FMCAD 2007 — 14 Nov 2007 — Circuit Level Verification of a High-Speed Toggle — p.3/22**
Reachability method for verifying real circuits

Approximate the non-linear ordinary differential equations (ODEs) in small neighborhoods by linear differential inclusions:

\[
Ax + b - u \leq \dot{x} \leq Ax + b + u
\]

Projection based representation of reachable space
Representing the Reachable Space

- Coho: Projectagons
  - Project high dimensional polyhedron onto two-dimensional subspaces.
  - A point is in the projectagon iff its projections are contained in the corresponding polygons.
  - Projectagons are efficiently manipulated using two-dimensional geometry computation algorithms.
  - Each edge of the polygon corresponds to a face of the high-dimensional polyhedron.
Representing the Reachable Space

- **Coho: Projectagons**
  - Project high dimensional polyhedron onto two-dimensional subspaces.
  - A point is in the projectagon iff its projections are contained in the corresponding polygons.
  - Projectagons are efficiently manipulated using two-dimensional geometry computation algorithms.
  - Each edge of the polygon corresponds to a face of the high-dimensional polyhedron.

- **Other approaches:**
  - symbolic hyper-rectangles (HyTech)
  - convex polyhedra (CheckMate)
  - orthogonal polyhedra (d/dt)
Reachability for Projectagons

- Extremal trajectories original from projectagon faces.
- Projectagon faces correspond to projection polygon edges.
- Coho computes time-advanced projectagons by working on one edge at a time.
Basic Step of Coho

1. Project
2. Advance
3. Compute model and time step
4. Union and simplify
5. Assemble projections
6. Create new projectagon

Basic Step of Coho

1. Project
2. Advance
3. Compute model and time step
4. Union and simplify
5. Assemble projections
6. Create new projectagon

Coho Linear Program Solver

Coho Linear Program

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b
\end{align*}
\]

\[
A^T_{\text{block}} = \begin{bmatrix}
\alpha_1 & \beta_1 \\
\alpha_2 & \beta_2 \\
\beta_2 & \alpha_2
\end{bmatrix}
\]

- Each inequality constraint corresponds to a face of the projectagon.
- One or two non-zero elements on each row of A.
- Dual is a standard form LP: \( A^T u = c \).
- Efficient linear system solver in \( O(n) \) time.
Coho Linear Program Solver

- Coho Linear Program

\[
A^T_{\text{block}} = \begin{bmatrix}
\alpha_1 & \beta_1 \\
\alpha_2 & \beta_2 \\
0 & 0 \\
\beta_2 & \alpha_2
\end{bmatrix}
\]

- Each inequality constraint corresponds to a face of the projectagon.
- One or two non-zero elements on each row of A.
- Dual is a standard form LP: \( A^T u = c \).
- Efficient linear system solver in \( O(n) \) time.

- Simplex-based linear program solver:
  - Reduce accumulated error by computing tableau matrix directly from input data.
  - Use Interval Arithmetic for well-conditioned problems.
  - Use Arbitrary Precision Rational Computation for ill-conditioned problems.
Summary of Coho

Summary of Basic Algorithm:
- The ODE model of circuit is approximated by linear differential inclusions.
- Use projectagons to represent reachable set.
- Coho is sound: all approximations overestimate the reachable space.
- Extensive use of linear programming.

Numerical Problems:
- Ill-conditioned linear programs
  - Exploit LP structure of Coho’s LPs.
  - Use hybrid approach of interval and arbitrary-precision arithmetic.
- Polygon intersection/union difficult with nearly-parallel edges.
  - Use hybrid approach of interval and arbitrary-precision arithmetic already implemented for LPs.
Circuit Verification

- Circuit description
  - Use MSPICE – a Matlab package that provides simple spice-like functionality.
  - Allows us to use same model for simulation and verification.
  - Simulate first:
    - Do not attempt to verify a incorrect system.
    - Have a rough idea of the reachable space to guide the verification.
    - Helps explain verification failures.

- Compute reachable set by Coho

- Verify design specification using reachable set
Toggle Circuit

- Start from the state where $\Phi$ is low, $x$ is low, $y$, $z$ are high
- $\Phi$ rises to high, $z$ falls to low
- $\Phi$ falls to low, $x$ rises to high
- $\Phi$ rises to high, $y$ falls to low, $z$ rises to high, $x$ falls to low
- $\Phi$ falls to low, $y$ rises to high

<table>
<thead>
<tr>
<th>step</th>
<th>$\Phi$</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Toggle Circuit

Start from the state where $\Phi$ is low, $x$ is low, $y$, $z$ are high

- $\Phi$ rises to high, $z$ falls to low
- $\Phi$ falls to low, $x$ rises to high
- $\Phi$ rises to high, $y$ falls to low, $z$ rises to high, $x$ falls to low
- $\Phi$ falls to low, $y$ rises to high

<table>
<thead>
<tr>
<th>step</th>
<th>$\Phi$ x y z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>1</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>3</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td>4</td>
<td>0 0 1 1</td>
</tr>
</tbody>
</table>
Toggle Circuit

<table>
<thead>
<tr>
<th>Step</th>
<th>( \Phi \times y \times z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>1</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>3</td>
<td>1 0 0 1</td>
</tr>
<tr>
<td>4</td>
<td>0 0 1 1</td>
</tr>
</tbody>
</table>

- Start from the state where \( \Phi \) is low, \( x \) is low, \( y, z \) are high.
- \( \Phi \) rises to high, \( z \) falls to low.
- \( \Phi \) falls to low, \( x \) rises to high.
- \( \Phi \) rises to high, \( y \) falls to low, \( z \) rises to high, \( x \) falls to low.
- \( \Phi \) falls to low, \( y \) rises to high
## Toggle Circuit

### Diagram

![Diagram of a toggle circuit](image)

### Table

<table>
<thead>
<tr>
<th>Step</th>
<th>$\Phi$</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Notes

- Start from the state where $\Phi$ is low, $x$ is low, $y$, $z$ are high
- $\Phi$ rises to high, $z$ falls to low
- $\Phi$ falls to low, $x$ rises to high
- $\Phi$ rises to high, $y$ falls to low, $z$ rises to high, $x$ falls to low
- $\Phi$ falls to low, $y$ rises to high
**Toggle Circuit**

- Start from the state where $\Phi$ is low, $x$ is low, $y$, $z$ are high
- $\Phi$ rises to high, $z$ falls to low
- $\Phi$ falls to low, $x$ rises to high
- $\Phi$ rises to high, $y$ falls to low, $z$ rises to high, $x$ falls to low
- $\Phi$ falls to low, $y$ rises to high

<table>
<thead>
<tr>
<th>step</th>
<th>$\Phi$</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Toggle Circuit

- Start from the state where $\Phi$ is low, $x$ is low, $y$, $z$ are high
- $\Phi$ rises to high, $z$ falls to low
- $\Phi$ falls to low, $x$ rises to high
- $\Phi$ rises to high, $y$ falls to low, $z$ rises to high, $x$ falls to low
  $\Phi$ falls to low, $y$ rises to high

<table>
<thead>
<tr>
<th>step</th>
<th>$\Phi$</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Specification

With a "well-behaved" clock input:

- The reachable space is a collection of trajectories whose period is twice that of the clock;

- The output is "well-behaved" like the clock.
Region 1 represents a logical low signal. The signal may wander in a small interval.

- 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.
Region 1 represents a logical low signal. The signal may wander in a small interval.

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.
Region 1 represents a logical low signal. The signal may wander in a small interval.

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.
Region 1 represents a logical low signal. The signal may wander in a small interval.

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.
Region 1 represents a logical low signal. The signal may wander in a small interval.

- 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.
The left and right boundaries of region 1 give min and max logical low level.

The left and right boundaries of region 3 give min and max logical high level.

The upper boundary of region 2 gives the minimum rise time.

The lower boundary of region 2 gives the maximum rise time.

The upper and lower boundaries of region 4 give the maximum and minimum fall times respectively.
Circuit Models

- Model MOS circuits as a collection of voltage controlled current sources
- Current function is obtained by simulating TSMC 180nm, 1.8 volt, bulk CMOS process
- Linearize the current function

\[ AV + b - u \leq ids(V) \leq AV + b + u \]

- Time derivative of voltage

\[ \dot{V} = C^{-1} \cdot I_c \]
Verification Strategy

Separate verification into four phases
- One phase for each transition of $\Phi$.  
- Assume bounding hyperrectangle for start of phase.  
- Establish bounding hyperrectangle at end of phase.  
- Containment establishes invariant set.  
- Allows parallel execution and parallel debugging.

Use invariant set to show that Brockett-ring at input implies Brockett-ring at output.

BUT overapproximation errors need to be managed
- Slicing  
- Multiple models
Slicing

Partition the reachable space along a critical variable:

- Large range of $\Phi$ leads to large approximation error in linear model.
- Partition reachable space by intervals of $\Phi$. 
Multiple Models

- Negative current from source to drain caused by overapproximation of linearization.
- Use multiple models to reduce error.
- Intersect the projectagons to eliminate non-physical states.
The Invariant Set

- Red: Hyperrectangles at beginning of each phase.
- Blue: Hyperrectangles at end of each phase.
- An invariant set with twice the period of the clock has been established.
Construct the brockett annulus for z, ignoring the inverter.

Perform a separate reachability analysis for the output inverter.

Arbitrary ripple counter.
Construct the brockett annulus for z, ignoring the inverter

Perform a separate reachability analysis for the output inverter

Arbitrary ripple counter
Experience from the Toggle

- Coho works for moderate dimensional systems.
- Topological properties provide a mathematically rigorous abstraction from continuous to discrete models.
- Leakage current included in the circuit model.
  - We found that we needed to add keepers to the circuit.
- Slicing and multiple models improved accuracy of linearization to enable successful verification.
- Seven-dimensional record sets a record – looks like we have headroom for more.
- Verification process currently involves substantial manual effort – more automation needed before useful in practice.
Conclusion

- Demonstrate a new reachability method to verify a real circuit
- Model the circuit with SPICE-level, non-linear differential equations.
- Projection based representation of reachable space
- Digital behavior corresponds to topological properties of invariant sets in the continuous space

Future Work

- Improve performance
- Exploit parallelism
- Develop more accurate circuit model
- Verify more circuits
- Apply Coho to hybrid systems
- Compare with other tools