Fault-Model Based Structural Analog Testing
To be Covered

- Analog fault models
- Analog Fault Simulation
  - DC fault simulation
  - AC fault simulation
- Analog Automatic Test-Pattern Generation
  - Using Sensitivities
  - Using Signal Flow Graphs
- Summary
Types of Structural Faults

- **Catastrophic (hard):**
  - Component is completely open or completely shorted.
  - Easy to test for.

- **Parametric (soft):**
  - Analog $R$, $C$, $L$, $K_n$, or $K_p$ ($K$ is transistor transconductance $\sim W/L$) is outside of its tolerance box.
  - Very hard to test for.
An Example

- Consider an amplification circuit:
  - First amplification stage comprises R1, R2 and the first OPAMP.
  - Second stage is a high pass filter comprising of C1 and R3.
  - Third stage is a low-pass filter comprising of R4, R5, C2 and the second OPAMP.
- The functional parameters of interest during test are shown on next slide.
  - Only two of these are single parametric faults.
  - Remaining ones are multiple parametric faults.
First stage gain
High-pass filter gain
High-pass filter cutoff $f$
Low-pass AC voltage gain
Low-pass DC voltage gain
Low-pass filter cutoff $f$

$R_2 / R_1$
$R_3$ and $C_1$
$C_1$
$R_4$, $R_5$, & $C_2$
$R_4$ and $R_5$
$C_2$
Basic Idea

- In DSP-based analog testing, no analog fault models were used.
- Here, we use the fault models, and tests are generated for “specific” multiple faults.
- Present status:
  - DSP-based testing remains the most important.
  - Structural testing method gaining acceptance as a supplement to the DSP-based methods.
Levels of Abstraction

- **Structural Level**
  - Structural View – Transistor schematic
  - Behavioral View – System of non-linear partial differential equations for netlist

- **Functional Level**
  - Structural View – *Signal Flow Graph*
  - Behavioral View – Analog network transfer function
Analog Test Types

Specifications:
- There is no general design technique for all analog circuits.
- No universal set of performance specifications.

Tests can be classified into three categories:
- Design characterization – Does design meet specifications?
- Diagnostic – Find cause of failures
- Production tests – Test large numbers of linear or mixed-signal circuits
Analog Fault Simulation

- Needed to evaluate the fault coverage and effectiveness of a set of analog test waveforms, which may be manually or automatically generated.

- Three kinds of simulation; if one passes we move on to the next:
  - DC fault simulation of non-linear circuits.
  - AC fault simulation of linear circuits.
  - Transient or time-domain fault simulation.
DC Analog Fault Simulation of Nonlinear Circuits
Introduction

- DC testing of analog circuits is attractive, since it requires less expensive testers and less testing time.
  - Useful to analyze how well a DC test can detect a given fault list.
  - Done by solving a set of non-linear equations using PSPICE, which converges after many iterations.
  - Several techniques reduce fault simulation CPU time while improving convergence.
Motivation for Analog Fault Simulation

- At present, no viable analog circuit synthesis tools exist.
  - Analog design done manually by experienced designers, using rules of thumb.
  - Analog fault simulation is extremely useful for what-if analysis.
    - What would happen if the value of R3 is out of the specification by 3%?
  - Analog fault simulation is far more computationally intensive than ordinary analog circuit simulation.
Various DC Fault Simulation Techniques

- Complementarity pivoting
- One-step relaxation
- Simulation by fault ordering
Complementarity Pivoting

Basic steps involved:

- Model all non-linear devices with piecewise-linear I-V characteristics (ideal diodes).
- Represent open, short, and parametric faults with switches.
  - Normally closed or normally open.
- Formulate fault simulation as the complementarity problem using n-port network theory.
- Solve the resulting complementarity problem with Lemke’s pivoting algorithm.
The circuit-under-test is modeled with linear resistors, controlled sources, DC independent sources, switches and ideal diodes.

- d diodes
- m switches
- k test nodes where voltage measurements are taken
- b branches where currents are measured

These k+b measurement ports, along with the d diodes and m switches, lead to k+b+d+m pairs of complementarity variables.
One-Step Relaxation

- The exact modeling of the analog fault is avoided.
  - Inspired by success of single stuck-at fault model.
  - Fault simulator should correctly predict the faulty circuit behavior, even in the presence of measurement error.

- For DC fault simulation, one solves the equation
  \[ f(x) = 0 \]
  where \( x \) is the circuit variable vector (node voltages and branch currents, and \( f \) is a non-linear system function.)
In Spice, N-R iteration solves the equation iteratively.

- Start with initial point $x^{(0)}$ and iterating until the difference between $x^{(k)}$ and $x^{(k+1)}$ converges.

The exact algorithm is:

- Guess $x^{(0)}$.
- Solve $J(x^{(k)})(x^{(k+1)}-x^{(k)}) = -f(x^{(k)})$, for $k=0,1,2,...$ until it converges.
- $J(x^{(k)})$ is the Jacobian matrix of $f(x^{(k)})$. 
In one-step relaxation, the N-R algorithm is operated for only one step using the good circuit solution as the starting point.

Thus, the method solves the equation:

\[ J_f(x_g)(x_f^{(1)} - x_g) = -f_f(x_g) \]

where \( J_f(x_g) \) and \( f_f(x_g) \) are the Jacobian matrix and function vector \( f_f(x) \) calculated at the point \( x_g \).

\( x_f^{(1)} \) is the vector to be solved.

Experiments on 29 MCNC benchmark circuits confirmed the validity of approximate fault simulation using one-step relaxation.

Yielded same fault coverage for majority of the circuits.
Simulation by Fault Ordering

- **Fault ordering:**
  - Improves simulation convergence, and reduce the number of iterations required by N-R iteration for exact DC fault simulation.
  - Convergence speed depends critically on the starting point.

- **Basic idea:**
  - Order faults so that the results of one simulation becomes a “good” starting point for the next simulation.
- This greedy strategy is very effective for parametric fault simulation and DC power supply testing where DC supply voltages are varied.
- Results on MCNC benchmarks show that the method can reduce the total number of iterations by 5 to 10 times.
Approach by Tian & Shi:

- Order the fault list so that the result from one fault simulation reduces the number of iterations for simulating the next fault.
- They use a simple greedy heuristic to order the faults.
AC Fault Simulation
Householder’s Formula

- Analyze circuit with Modified Nodal Analysis:
  \[ T \mathbf{x} = w \]

- Equivalent faulty circuit equation:
  \[ T_f \mathbf{x}_f = w_f \]

- Formula \((T_f \text{ differs only a little from } T)\):
  \[ (A + U S W)^{-1} = A^{-1} - A^{-1} U (S^{-1} + W A^{-1} U)^{-1} W A^{-1} \]

- Reduces amount of equation solving – 10 x speedup over sparse matrix techniques
Monte-Carlo Simulation

- Perform analog simulation for randomly-generated small variations in analog circuit component values.
- Actual IC manufacturing makes good circuits deviate by such values.
- Good in practice but good and bad machines have different worst-case corners.
  - Tends to underestimate circuit response bounds – may claim faults are detectable when they are not
Analog Automatic Test-Pattern Generation
Method of ATPG Using Sensitivities

- Compute analog circuit sensitivities
- Construct analog circuit bipartite graph
- From graph, find which O/P parameters (performances) to measure to guarantee maximal coverage of parametric faults
  - Determine which O/P parameters are most sensitive to faults
- Evaluate test quality, add test points to complete the analog fault coverage
Sensitivity

- **Differential:**

\[ S_{\Delta T_j/\Delta x_i} = \frac{T_j}{\partial x_i} \frac{x_i}{T_j} \frac{\partial T_j}{\partial x_i} \frac{\Delta T_j}{\Delta x_i} \bigg|_{x_i, T_j} \Delta x_i \rightarrow 0 \]

- **Incremental:**

\[ \rho_{x_i} = \frac{T_j}{x_i} \frac{x_i}{T_j} \frac{\Delta T_j}{\Delta x_i} \]

- **\( T_j \) – performance parameter**
- **\( x_i \) – network element**
Circuit Model
## Incremental Sensitivity Matrix of Circuit

<table>
<thead>
<tr>
<th></th>
<th>R₁</th>
<th>R₂</th>
<th>C₁</th>
<th>R₃</th>
<th>R₄</th>
<th>R₅</th>
<th>C₂</th>
<th>A₁</th>
<th>A₂</th>
<th>fc₁</th>
<th>A₃</th>
<th>A₄</th>
<th>fc₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.91</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.58</td>
<td>0.38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.91</td>
<td>-0.89</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.96</td>
<td>0.48</td>
<td>-0.48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.97</td>
<td>-0.97</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.88</td>
<td>-0.91</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

A₁, A₂, fc₁, A₃, A₄, fc₂
Bipartite Graph of Circuit
Single Fault Best and Worst-Case Deviations

\[ \begin{align*}
A1 & \quad \{ \\
& \quad \quad 5 \leq \frac{\Delta R_1}{R_1} \leq 15.98 \\
& \quad \quad 5 \leq \frac{\Delta R_2}{R_2} \leq 14.1 \\
& \quad \quad 5 \leq \frac{\Delta R_3}{R_3} \leq 20.27 \\
& \quad \quad 5 \leq \frac{\Delta C_1}{C_1} \leq 11.6 \\
& \quad \quad 5 \leq \frac{\Delta R_4}{R_4} \leq 15 \\
& \quad \quad 5 \leq \frac{\Delta R_5}{R_5} \leq 15 
\end{align*} \]

\[ \begin{align*}
fc1 & \quad \{ \\
& \quad \quad 5 \leq \frac{\Delta R_3}{R_3} \leq 14.81 \\
& \quad \quad 5 \leq \frac{\Delta C_1}{C_1} \leq 15.2 \\
& \quad \quad 5 \leq \frac{\Delta R_5}{R_5} \leq 14.65 \\
& \quad \quad 5 \leq \frac{\Delta C_2}{C_2} \leq 13.96 \\
& \quad \quad 5 \leq \frac{\Delta R_4}{R_4} \leq 15 \\
& \quad \quad 5 \leq \frac{\Delta C_2}{C_2} \leq 35 
\end{align*} \]
Weighted Bipartite Graph
Generates tests and defines parametric faults for analog circuits

ATPG Approach:
- Backtraces signals from circuit outputs (specified with magnitude/phase tolerance) through circuit using signal flow graph (SFG)
  - Inverts the SFG to allow backtracing
- Evaluates internal waveforms using an output waveform sample set by evaluating SFG
Test Generation via Reverse Simulation

- Find good circuit signal values at all nodes using good output waveform
- Find bad circuit signal values at all nodes using bad output waveform (use extrema of tolerance box for magnitude or phase)
- Finds faulty value of analog component necessary to drive output waveform out of tolerance box
  - Mark all corresponding edges to fault
  - Compute modified SFG weights that give good value after bad edges in inverted SFG
Integrator Example

- Basic integrator circuit with ideal OPAMP
Signal Flow Graph Inversion

- SFG represents analog network equations
  \[ value (i) = \sum (parent \ node \ value) \times (edge \ weight) \]
- May be inverted
  \[ x_2 = ax_1 + bx_3 + cx_4 \quad x_1 = \frac{1}{a} x_2 - \frac{b}{a} x_3 - \frac{c}{a} x_4 \]

- SFG inversion algorithm follows from Balabanian’s example (1969)
SFG Inversion Algorithm

- Start at a primary input, $x_1$, a source node
- Reverse the direction of the outgoing edge from $x_1$ to $x_2$ and change the weight to $1/a$
- Redirect all edges incident on $x_2$ to $x_1$ and change weights appropriately
- Continue for all source nodes, from all inputs, until the output becomes a source

*Inverted SFG Properties:*
- Equivalent to original SFG
- A feed-forward network – graph cycles cut
- Represents set of integral equations, solved by numerical differentiation
- May be an unstable system
Graphs for Integrator

- SFG part after fault has faulty value
- Bad signal does not disappear, circuits are linear
- Method applicable to all circuits representable with SFGs (1\textsuperscript{st} and 2\textsuperscript{nd} order)
- Backtrace over all paths from outputs to inputs
- 2\textsuperscript{nd} order approximation for \( s \) differential operator

\[ V_{in}(s) \quad \begin{array}{ccc} & \text{1/R} & C \\ & \text{1/s} & \\
\end{array} \quad \begin{array}{ccc} & \text{Vout}(s) & \text{Vin(s)} \\ & \text{2} & \\
\end{array} \quad \text{Vout}(s) \]

Original SFG

\[ -1/R_f C \]

Faulty Edge

Good Edge

Inverted SFG

\[ \text{2nd order approximation for s differential operator} \]
Analog Fault Definition

- Want to find parametric fault value for R1
- Use good & bad node values for all nodes from reverse analog simulation
- For parametric fault definition in inverted SFG
  - Use good values for nodes before fault
  - Use bad values for nodes after fault
  - Linear equation in 1 variable for each component
  - Manipulate component equations symbolically to get component tolerance
Calculation of $R1$ Tolerance to Cause Fault

$V_{in}(s)$ \[1\] $V_{out}(s)$ $V_{in}(s)$ $V_{out}(s)$

Original SFG

Inverted SFG

$R_{1}$ Tolerance = $\text{goodval} (R_{1})$ - $\text{badval} (R_{1})$

$\text{goodval} (1) - R_{1} C + \text{badval} (3) - R_{f} C = \text{badval} (2)$

$\text{badval} (R_{1}) = C \left( \text{badval} (2) + \text{badval} (3) / R_{f} C \right)$

$\text{goodval} (R_{1}) = C \left( \text{goodval} (2) + \text{goodval} (3) / R_{f} C \right)$

$R_{1}$ Tolerance = $\text{goodval} (R_{1}) - \text{badval} (R_{1})$
SFG ATPG Results

- $R_1 = 10 \, K\Omega$, $R_f = 100 \, K\Omega$, $C = 0.01 \, \mu F$
- Output tolerance = +10%, used SPICE output
- Calculated test signal and component deviations
- Deviations analogous to fault coverage

<table>
<thead>
<tr>
<th>Component</th>
<th>Allowed Value</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>9.09 KW</td>
<td>-9.1%</td>
</tr>
<tr>
<td>$R_f$</td>
<td>80.99 KW</td>
<td>-19.01%</td>
</tr>
<tr>
<td>$C$</td>
<td>0.0093 mF</td>
<td>-7.0%</td>
</tr>
</tbody>
</table>
Generated Test Waveform

ACTUAL SPICE INPUT

ACTUAL SPICE OUTPUT

TEST SIGNAL (FROM SFG INVERSION)

Voltage

Time (ms)
Summary of SFG Method

- Works for multiple input, multiple output circuits
- Handles single and multiple parametric faults, and catastrophic faults
  - Symbolic solution too difficult for multiple parametric fault tolerance – use iterative method with simulation to obtain deviation
- Extended to cover transistor biasing faults in analog circuits
- Extended to analog multipliers and comparators
Summary

- Analog model-based testing – Just starting to get some acceptance
  - Structural test with a fault model
  - Offers advantage of testing specific parametric and catastrophic faults
- Analog DSP-based testing – Main stream
  - Functional test without fault model
- Problem is worsening – 22-bit A/D converters coming, expected to sample at 1 GHz