Approximate property checking of mixed-signal circuits

Parijat Mukherjee, Texas A&M University
Chirayu Amin, Intel Corporation
Peng Li, Texas A&M University

TAU 2014
Thursday, March 6th 2014
Outline

• Mixed-signal design
• Property checking
• Why approximate?
  • Interpreting “failure probability”
  • Approximate property checking
  • Implementation details
• Conclusion : Challenges and future work
Mixed-signal design

• Significant analog components
  – Inherently complex: Continuous variables
  – Extremely large state space
  – Simulation computationally expensive

• Why talk of it now?
  – Circuit complexity
  – Process variability
  – Frequency band of interest
  – Effect of Jitter / Noise etc

INCREASING!
Does a circuit “always” perform its intended function?

Violations are observed here

- Input Parameters
  - Input Signal parameters
  - Process parameters
  - Initial Conditions
    - #(time steps)
- Circuit
  - Complex
  - Non-linear
  - Internal state space
    - Grows with
      - #(time steps)
- Output Properties
  - Output Signals
  - Specifications
    - Do output signals violate specifications?

May not always capture the intended function (specification)
Does a circuit “always” perform its intended function?

Violations are observed here

Input Parameters
- Input Signal parameters
- Process parameters
- Initial Conditions

Henceforth referred to as out-of-specification failures

May not always capture the intended function (specification)
Detection and Diagnosis

- Outputs
  - Probability of failure
  - Debug information
    - Counterexamples
    - Important parameters
    - Failure patterns

Does circuit meet specifications over its entire range of operation?

Why doesn’t it meet target specifications?

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Statistical property checking

• What is the statistical quantity varied?
  – Any parameter that can change circuit behavior

• What is the statistical quantity checked?
  – Output signals vs. specifications
Addressing circuit complexity

Circuit Size

Failure Probability

Can be simulated via SPICE

Cannot be simulated via SPICE

Circuit level techniques

Hierarchical decomposition via models

System level techniques

Warning: Figure may not be drawn to scale

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Addressing circuit complexity

Can be simulated via SPICE

Cannot be simulated via SPICE

Circuit level techniques

System level techniques

Hierarchical decomposition via models

Warning: Figure may not be drawn to scale

So what’s the catch?

• Rejection of potentially important information
• Correlations between components overlooked

Solution

• Decompose only when absolutely necessary

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
**Existing work - Detection**

Warning: Figure may not be drawn to scale

- **Standard Monte Carlo**
- **Yield Estimation**
- **System level techniques**

Circuit Size

Failure Probability

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14

Warning: Figure may not be drawn to scale.
Outline

• Mixed-signal design
• Property checking
• Why approximate?
• Interpreting “failure probability”
• Approximate property checking
• Implementation details
• Conclusion : Challenges and future work
Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
There exists a mapping between Parameters and Properties!

Parameter and property spaces

Parameter space

Property space

Model

There exists a mapping between Parameters and Properties!
Failure can be observed directly in the property space!
Parameter and property spaces

Interactions between parameters give rise to failures!

\[ P(F|X1 \in \text{red}, X2 \in \text{red}) = 1 \]

\[ P(F|Y1) = 0 \]

\[ P(F|Y1) = 1 \]
The real picture

Parameter space

Property space

Multiple violations over multiple parameter combinations

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Failure probability estimation

Statistical framework

Approximate failure probability

Parameter space
Waveform generator
Model
Waveform analyzer
Property space

Simulation may be expensive!
How approximate is approximate?
Requirements

• A statistical framework that:
  – Requires low number of SPICE simulations
  – Can bound the failure probability uncertainty
  – Can deal with varying input distributions
  – Does not need to assume an output distribution
  – Does not assume an input-output relationship
  – Can exit early with an approximate bounded estimate

• By assuming that:
  – Failure regions are more or less contiguous
    (Few probable failure regions vs. many unlikely ones)
What is Failure probability?

\[ P(failure) = \frac{\text{Area}(\text{red})}{(\text{Area}(\text{red}) + \text{Area}(\text{green}))} \]

\[ P(F) = \int_Y P(F|y)p(y)\,dy \]

Easy to do via sampling :- Monte Carlo Simulations!

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Where does uncertainty arise?
Part 1

Have we followed the output distribution exactly?
Have the relative areas been estimated correctly?

Only relevant metric: Number of samples that fall into each zone!
Adaptive sampling

SPICE is Expensive!

Few Strategic Samples (Arbitrary distribution)

Needs some sort of knowledge representation scheme:

- Sampling distribution
- Failure Model

Failure Estimate

Parameter

Circuit

Property
Model driven sampling

Accuracy $\propto$ How well model has captured failure boundaries

SPICE is Expensive!

Inexpensive!

Extensive Resampling (Parameter distribution)

Few Strategic Samples (Arbitrary distribution)
Where does uncertainty arise? Part 2

Do we have at least one sample in the “likely” regions?
The N initial samples

Following parameter distribution can cause redundant effort

Warning: Artificial test case designed to excite complex interactions
The N initial samples

Spread out samples using prior information (only if it exists)
Warning: Artificial test case designed to excite complex interactions
Where does uncertainty arise?

Part 3

How accurate are our boundaries?

Are inaccurate boundaries for unlikely events really a problem?
Observation: Highest uncertainty around failure boundaries
(Warning: Chosen model may introduce additional spurs)
Reducing uncertainty

Can be reduced by Active learning !!!

( Run SPICE simulations for low confidence regions )
Adaptive Sampling

Model accuracy $\propto$ How well failure boundaries have been captured
( Discussed previously )
Adaptive Sampling

Model accuracy $\propto$ How well failure boundaries have been captured
Adaptive sampling produces new samples close to boundary
Bounding the failure probability

\[ P(\text{failure}) \leq \hat{P}(\text{failure}) + \text{Area (green)} \]
\[ P(\text{failure}) \geq \hat{P}(\text{failure}) - \text{Area (red)} \]

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
High level overview

- Introducing the “interval learner”
- A bias compensation stage wrapped around it

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
The interval learner

- Instead of a mapping $X \rightarrow Y$, the model returns,

$$\forall x, \quad P(F|x) = \int P(F|y) p(y|x) \, dy$$

This captures the model's belief about a prediction.
The interval learner

- Purpose of model is to make a prediction. Thus, $\forall x$:
  - The prediction
    $$\hat{i}(x) = P(F \mid E[y \mid x])$$
  - Probability of misprediction
    $$P(\epsilon \mid x) = \begin{cases} P(F \mid x) & \hat{i}(x) = 0 \\ 1 - P(F \mid x) & \hat{i}(x) = 1 \end{cases}$$

$$\forall x, \quad P(F \mid x) = \int_{y} P(F \mid y) p(y \mid x)\, dy$$
The interval learner

- Purpose of model is to make a prediction. Thus, \( \forall x : \)
  
  - The prediction
    \[
    \hat{i}(x) = P(F \mid E[y \mid x])
    \]
  
  - Probability of misprediction
    \[
    P(\varepsilon \mid x) = \begin{cases} 
    P(F \mid x) & \hat{i}(x) = 0 \\
    1 - P(F \mid x) & \hat{i}(x) = 1 
    \end{cases}
    \]

  May still suffer from data / model bias

\[
\hat{P}(F) = \int \hat{i}(x) p(x) \, dx \\
[\min P, \max P] = \\
\begin{bmatrix} 
\hat{P}(F) - \int_{x \mid \hat{i}(x) = 1} P(\varepsilon \mid x) p(x) \, dx , \quad \hat{P}(F) + \int_{x \mid \hat{i}(x) = 0} P(\varepsilon \mid x) p(x) \, dx 
\end{bmatrix}
\]
• K-fold cross validation on original training data
• At each iteration, delete $m = N/K$ samples
Bias reduction: Ensemble learning

- At each iteration, delete $m = N/K$ samples

Original Training Data

Delete -m

Model 2

Approximate property checking of mixed-signal circuits - Prijat Mukherjee, TAU '14
• At each iteration, delete $m = N/K$ samples.
### Bias reduction: Data and Model

- Consolidate estimates from last $K$ models

<table>
<thead>
<tr>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(failure) 1</td>
<td>P(failure) 2</td>
<td>P(failure) 3</td>
<td>P(failure) 4</td>
<td>P(failure) 5</td>
<td>← average</td>
</tr>
<tr>
<td>Lower bound 1</td>
<td>Lower bound 2</td>
<td>Lower bound 3</td>
<td>Lower bound 4</td>
<td>Lower bound 5</td>
<td>← min</td>
</tr>
<tr>
<td>Upper bound 1</td>
<td>Upper bound 2</td>
<td>Upper bound 3</td>
<td>Upper bound 4</td>
<td>Upper bound 5</td>
<td>← max</td>
</tr>
</tbody>
</table>

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Bias reduction: Training Data

• Adaptive sampling points added at each step
  – Slowly reduces bias due to original training set
Bias reduction: Training Data

- Adaptive sampling points added at each step
  - Slowly reduces bias due to original training set
Bias reduction: Training Data

- Nature of ensemble changes very little
  - K-fold subsampling only on original seed data
When to stop?

Target parameter space

Resampled Model

$K = 5$

Model’s predictive accuracy will grow with increase in SPICE data
When to stop?

Target parameter space

Resampled Model

K = 5

Reflected by reduction in bounds

P(failure)

Original Bounds

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Summary

Bayesian Additive Regression Trees

Latin Hypercube sampling

Run loop at least $K$ times

$N$ Initial samples

Simulate (SPICE)

K-fold subsampling

Build failure model

Latin Hypercube sampling

Pick $q$ low confidence points

Check failure probability and bounds

Resample failure model

Tournament selection

Average / bound over last $K$ runs

Lots of intricate details

Exit if met!

Approximate property checking of mixed-signal circuits - Prijat Mukherjee, TAU '14
Experimental setup

Integer-N Phase-locked loop (PLL)
- 9 Input parameters and 31 design uncertainties (40 total)
- 8 output properties
- SPICE Simulation time : 0.5 hrs
- Two versions with varying error rates
Results: Adaptive sampling

- Cost of SPICE simulation dominant
- Cost of model building and evaluation negligible

Evolution of $P(F)$ estimate with number of simulations for PLL2

Approximate property checking of mixed-signal circuits - Parijat Mukherjee, TAU '14
Results: Bias compensated interval learner

- Interval grows smaller or less biased over time
- Better off using bootstrap if sampling using MC + LHS

<table>
<thead>
<tr>
<th></th>
<th>PLL1 99% Bootstrap CI (MC + LHS)</th>
<th>Bias compensated interval learner (MC + LHS)</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P(F)$</td>
<td>$\text{min}P, \text{max}P$</td>
<td>$P(F)$</td>
</tr>
<tr>
<td>200</td>
<td>9.50</td>
<td>4.50, 15.00</td>
<td>12.07</td>
</tr>
<tr>
<td>400</td>
<td>7.00</td>
<td>4.00, 10.00</td>
<td>10.90</td>
</tr>
<tr>
<td>600</td>
<td>6.33</td>
<td>3.83, 9.00</td>
<td>9.54</td>
</tr>
<tr>
<td>800</td>
<td>8.13</td>
<td>5.75, 10.63</td>
<td>9.88</td>
</tr>
<tr>
<td>1000</td>
<td>8.70</td>
<td>6.50, 11.10</td>
<td>9.64</td>
</tr>
</tbody>
</table>

- Works better for lower failure probabilities (Previous slide)
Outline

• Mixed-signal design
• Property checking
• Why approximate?
• Interpreting “failure probability”
• Approximate property checking
• Implementation details

• Conclusion: Challenges and future work
Summary of preliminary work

• Failure probability estimation
  – Accuracy comes at a (high) cost
  – Simulation budgets are limited
  – Focus on doing the best job within the limited budget
  – Allow user to exercise accuracy vs. turn-around-time trade-offs
Future work

Short Term
- Optimal choice of model?
- Compensate for model bias?

Long Term
- Diagnosis using discovered failures [DAC ’14]
- Extend to equivalence checking
- Failure model applied to system level analysis?
Lots of questions answered BUT
Most are yet to be answered

Your questions !!!