8. Verifying Analog/Mixed-Signal Systems

Jacob Abraham

Department of Electrical and Computer Engineering
The University of Texas at Austin

Verification of Digital Systems
Spring 2020

February 13, 2020

Acknowledgements

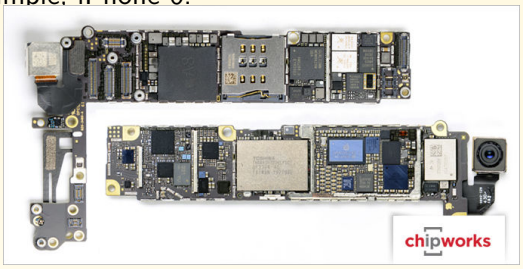
- Graduate students: A. Balivada, N. Nagi, Y. Hoskote
- U. C. Berkeley: K. Aadithya, R. Brayton, A. Mishchenko, P. Nuzzo, S. Ray, J. Roychowdhury, B. Sterin
- Univ. of Illinois: S. Ahmadyan; S. Vasudevan
- Georgia Tech: A. Chatterjee, S. Deyati, B. Muldrey

ECE Department, University of Texas at Austin Lecture 8. Verifying Analog/Mixed-Signal Systems Jacob Abraham, February 13, 2020 1 / 27

Motivation

Though we talk about living in a "digital world", there are many analog/mixed-signal subsystems that verification tasks need to address

Example, iPhone 6:



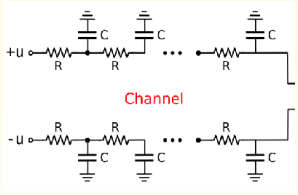
Some of the non-digital blocks in the iPhone 6

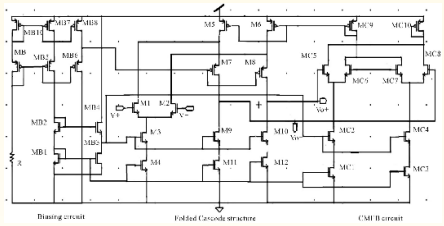
- In CPU Core: PLL, voltage regulators, ...
- SDRAM controller
- Power management, tracking
- Communication: RF transceivers, WiFi, Bluetooth, etc.
- Sensors: Accelerometer, Gyroscope, Barometor
- NFC controller, Touchscreen controller
- ...

What is Analog/Mixed-Signal?

Circuits dealing with non-binary, or a mixture of binary and non-binary values

- interconnects (on chip, I/O channels, power supplies, ...)
- op-amps, comparators, mixers, charge pumps, ...
- digital circuits driven at high speeds → analog effects
- when timing delays are important

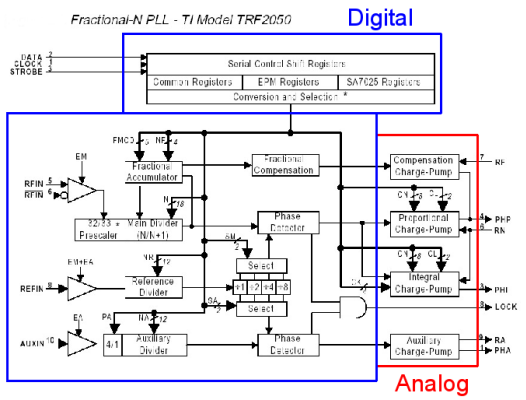


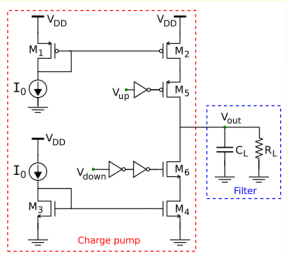


Examples of Analog/Mixed Signal (AMS)

Tightly coupled analog and digital

Example: Modern PLL





Further Motivation

20% of design bugs seen in AMS portion

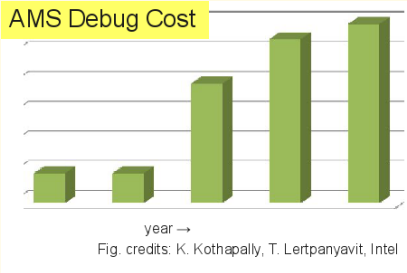


Fig. credits: K. Kothapally, T. Lertpanyavit, Intel

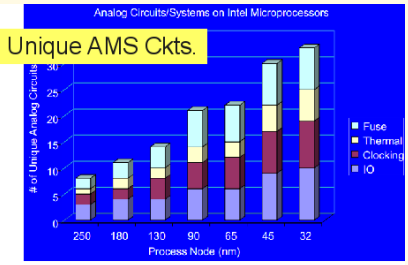
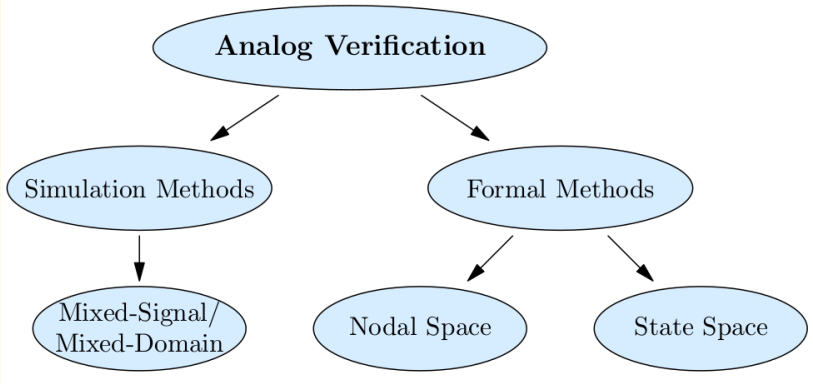


Fig. credits: Rachel Parker, Intel

Input from Intel Corp.

- "Simulation speed significantly limits [bug] coverage"
- "Greater interactions (between Analog/Digital) – can't validate in isolation"
- "Automation to generate and verify analog behavioral models"
- "Seamless integration of analog and digital verification tools"

Approaches to Verifying Analog/Mixed-Signal Designs



Barke *et al.*, DATE 2009

Correctness in Analog Verification

Digital designs

- Results are direct for logic – output values should be correct during every clock cycle
- Timing or power-related bugs also result in either the correct or incorrect logic values in storage elements

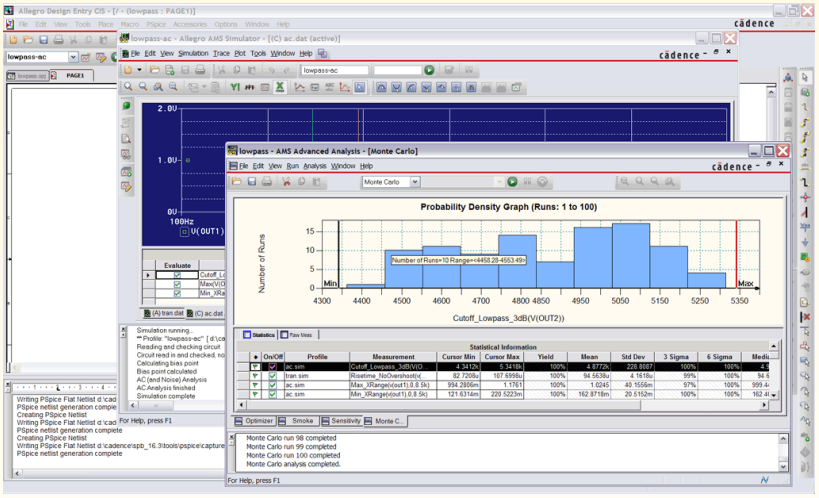
Analog subsystems

- Analog value at some output may be different from expected, but it is not so simple to determine whether it is correct
- Analog subsystems must meet **specifications**
  - For example, signal to noise ratio, linearity, etc.
- **This problem has not been solved in research to date in analog verification**

Simulation-Based Verification

Monte-Carlo simulation to explore process corners

Example: Cadence Allegro



Issue: Analog circuit simulation is very slow

Directed Stimulus Generation

Goal-oriented stimulus generation

- Directed toward a goal region
- Triggers specific behavior or output in the circuit

Coverage-driven stimulus generation

- Attempt to cover the state space
- Samples are uniformly distributed over the reachable state space

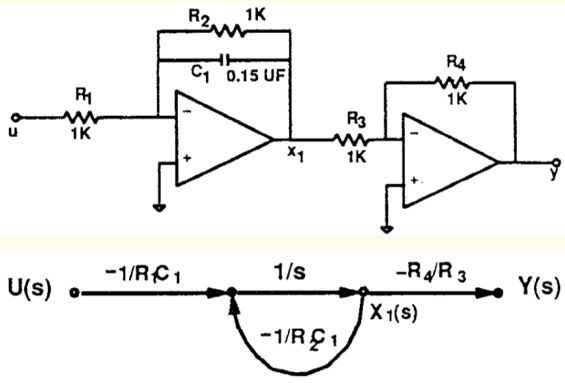
Multi-Objective Rapidly-exploring Random Trees (MORRT)

- Grow a tree in the search space of the analog circuit
- Bias the growth of the tree toward the goal region
- Simultaneously, bias the growth of the tree to increase coverage

Verifying the Transient Response of Linear Analog Circuits

Linear Analog Circuit Transfer Function (example, filters):

$$H(s) = \frac{Y(s)}{X(s)} = \frac{a_0 + a_1 s + \ldots + a_m s^m}{1 + b_1 s + \ldots + b_n s^n}, \qquad m \leq n$$

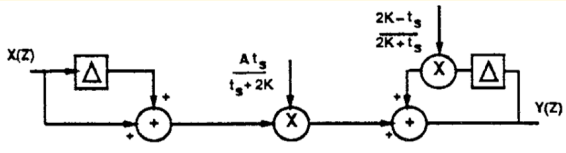


Low pass filter and its signal flow graph

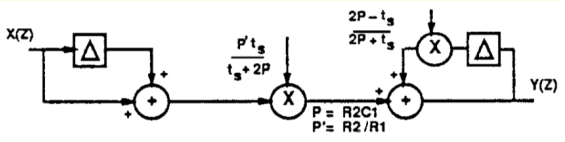
Balivada, VTS 1995

Verification Process

Both specification and implementation are discretized using a bilinear transformation



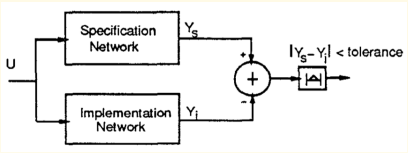
Discrete realization of transfer function (specification)



Discrete realization of filter implementation

Checking Equivalence

Two networks are implemented as digital circuits with some word size (i.e., as FSMs), and can be checked for equivalence for a given range of inputs and a tolerance value for the outputs



Setup for state machine comparison

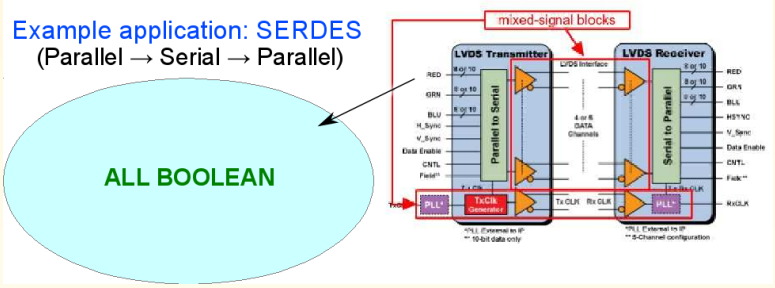
Results with capacitor value changes in implementation

- $C_1 = 0.1\mu F$: circuit implements transfer function accurately
- $C_1 = 0.15\mu F$: mismatch after 3 and 7 steps for ± 0.015 and ± 0.05 tolerances respectively
  - For a tolerance of ± 0.24, no mismatch for over 20 steps

Model checking can also be used on the discretized models

AMS Validation/Debug by Booleanizing Analog Dynamics

(U.C. Berkeley project)



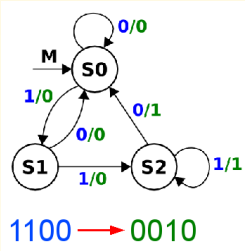
Problem and challenge

- Problem: analyze, simulate, verify, debug complete system (digital + AMS)
- Challenge: Boolean and analog models don't mix

Solution

- Approximate AMS parts using purely Boolean models
- Analyze, verify all-Boolean system using digital design tools

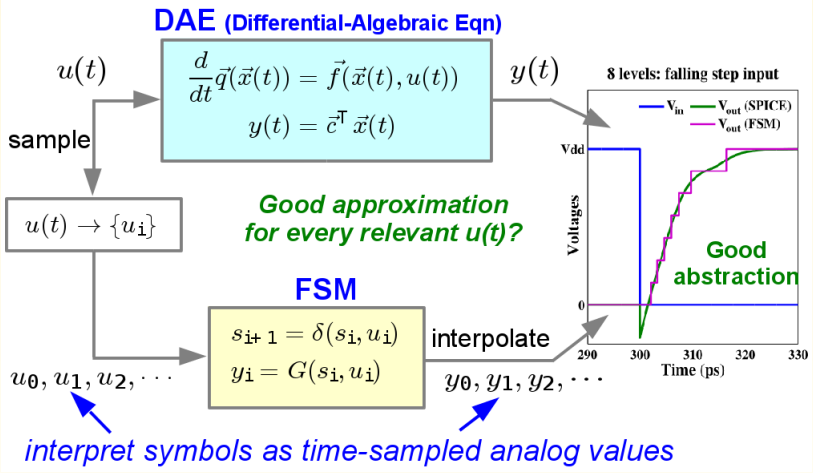
Approach: Use Finite-State Machines (FSMs)



Techniques for Booleanization

- Models for latches and flip-flops
  - Very inefficient for circuits with deep memories
- ABCD-L: Linear analog systems
  - Interconnect, equalizers, filters, etc.
- ABCD-NL: Non-linear AMS circuits and systems
  - Charge pumps, ADC/DAC, comparators, etc.

Equivalent FSMs for AMS systems



Problem: How to come up with a good FSM

Finding a Finite-State Machine Representation

Using a system of nonlinear differential equations, generate an FSM abstraction which would model the behavior accurately

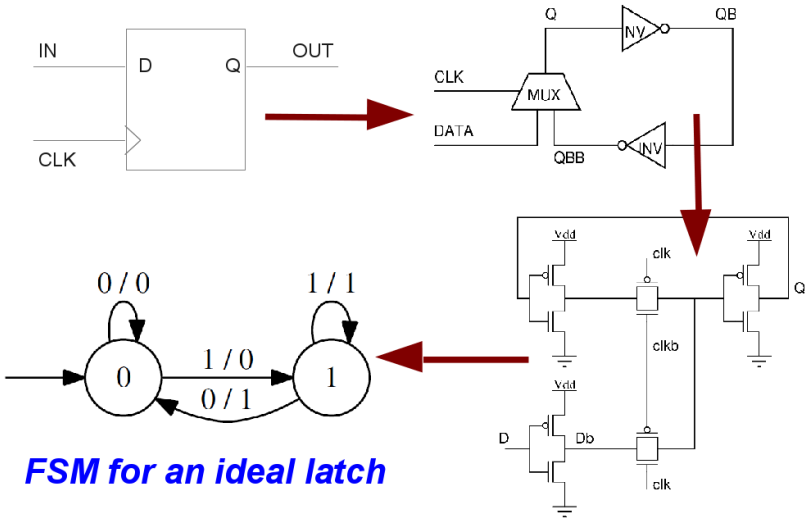
Adapt Angluin's algorithm

- D. Angluin, "Learning regular sets from queries and counterexamples", Information and Computation, vol. 75, 1987.
- Learn internal structure of black-box deterministic finite automaton (DFA) by,
  - simply querying the DFA with strings of inputs,
  - and observing the outputs

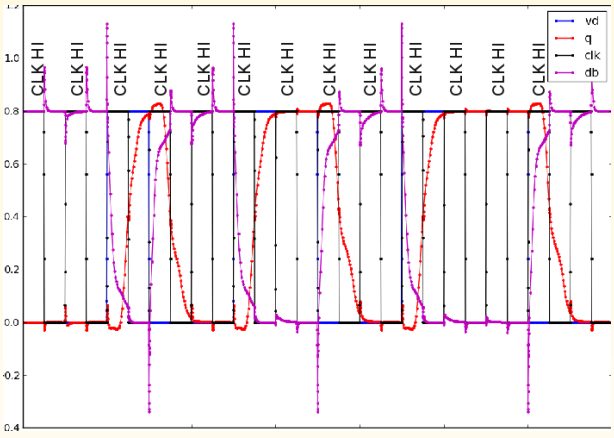
Approach

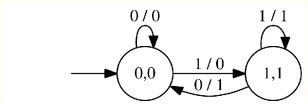
- Query the AMS system as if it were a Boolean FSM
  - Boolean input strings interpolated to analog u(t)
  - SPICE simulation of AMS gets y(t)
  - y(t) sampled to get Boolean outputs

Example: Ideal CMOS Latch



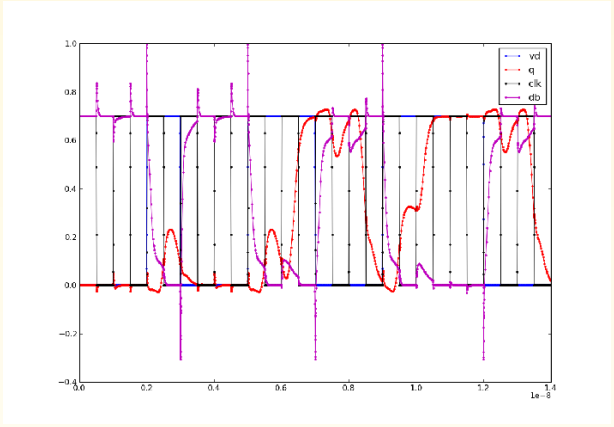
Example Latch: Vdd=0.8V, T=1ns

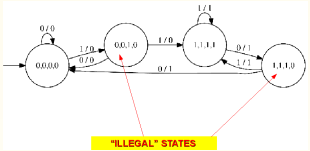




Waveforms show latch works correctly
Ideal latch automatically generated

Example Latch: Vdd=0.7V, T=1ns

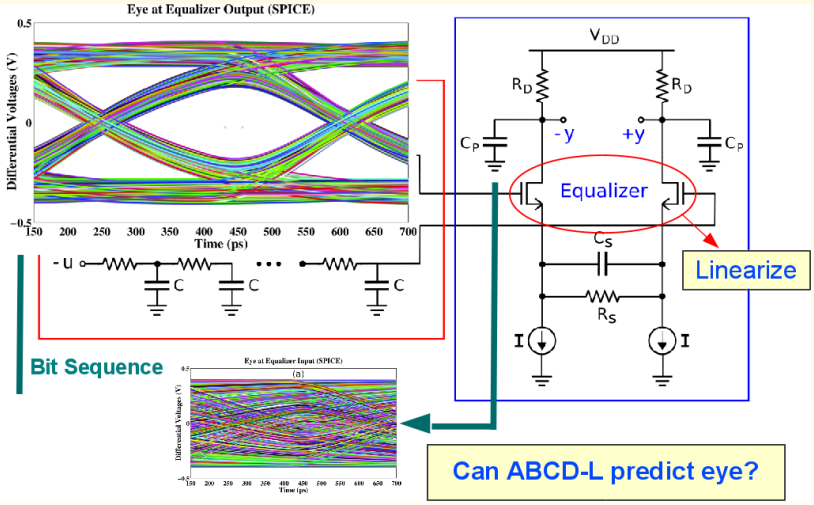




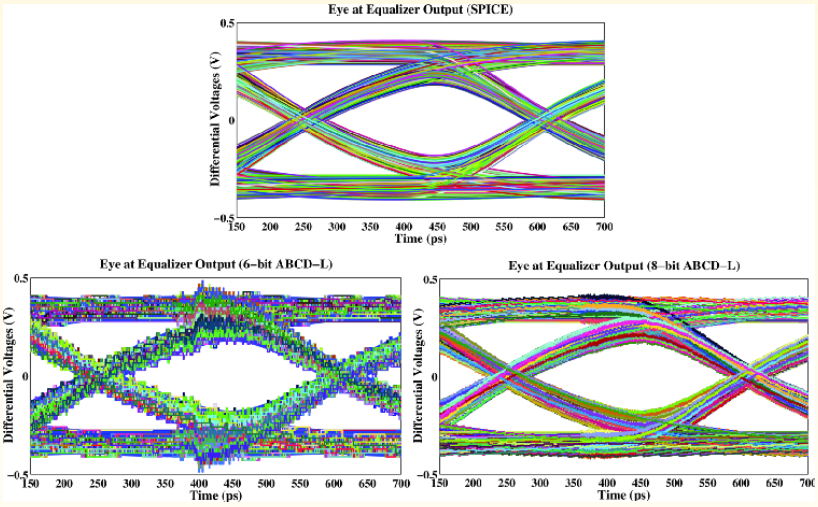
Data needs to be stable for 2 or more cycles; if not, latch fails
Non-idealities lead to more states

ABCD-L: LTI Channel + Equalizer

Tool from UC Berkeley for linear systems

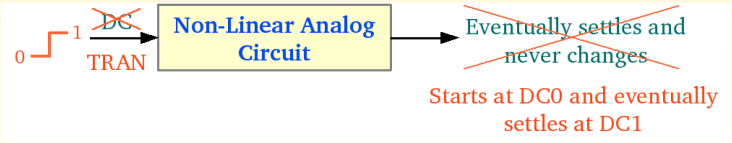


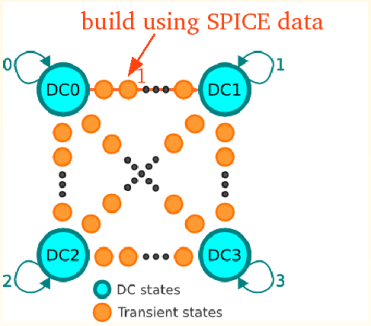
ABCD-L: LTI Channel + Equalizer, Cont'd



Nonlinear Blocks: ABCD-NL

Problem:





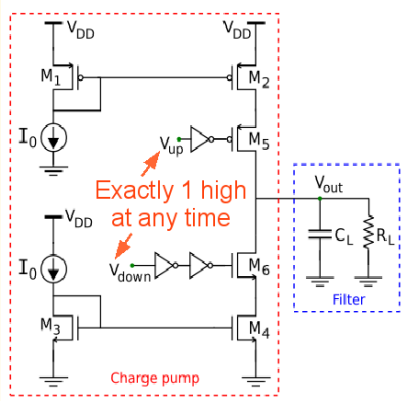
Key characteristics

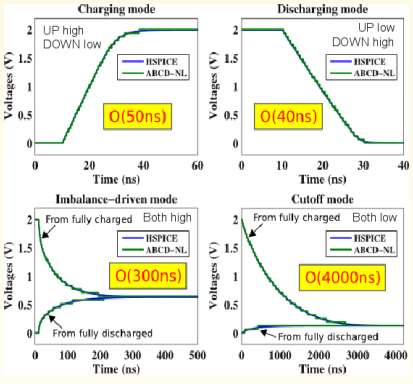
- DC FSM states with loops
- Multiple such DC states
- Capture different DC Operating Points – don't change instantly
- Transient FSM states between each pair of DC states
- Purely Boolean Model

Example: Charge Pump + Filter

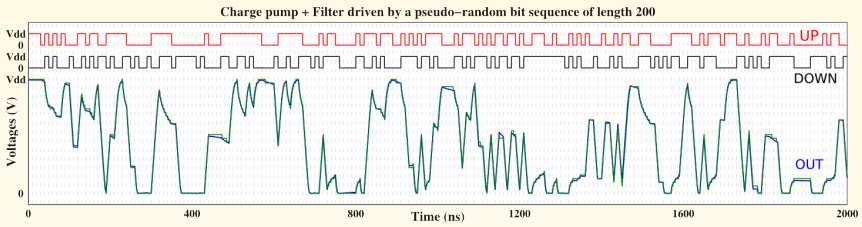
Discretize Vup, Vdown using 1 bit each, Vout using 5 bits

Booleanize using ABCD, Simulate Boolean model





Charge Pump – Long Pseudo-Random Binary Sequence



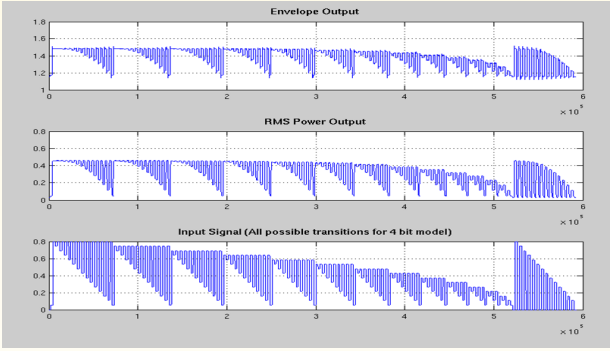
Non-linear analog dynamics accurately captured over long time-frame

10X speedup in Python

FSM Models for RF Circuits

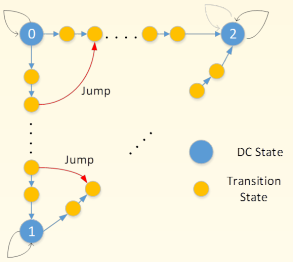
Issues

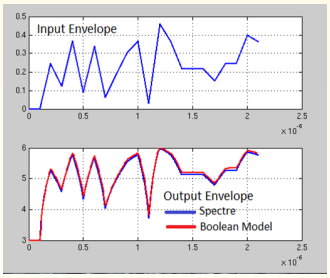
- RF circuits are DC-blocking
- Need to model carrier envelopes
- Memory effects in RF circuits must be incorporated
- Direct model extraction from hardware?



Model Accuracy and Effectiveness

Transiting to another state before completing present transition to maintain analog waveform continuity





Cadence SpectreRF Transient	4,000 ms
HP ADS Transient	3,000 ms
Cadence Harmonic Balance	500 ms
Boolean model	1 ms

Model Extraction from Hardware