Post Silicon Validation of Analog/Mixed Signal/RF Circuits and Systems: Recent Advances

(Invited Paper)

Abhijit Chatterjee, Sabyasachi Deyati and Barry J. Muldrey

School of Electrical & Computer Engineering, Georgia Institute of Technology, Atlanta Georgia 30332

e-mail: sdeyati3@gatech.edu, bmuldrey3@gatech.edu, abhijit.chatterjee@ece.gatech.edu

Abstract— Technology scaling along with unprecedented levels of device integration has led to increasing numbers of analog/mixed-signal/RF design bugs escaping into silicon. Such bugs are manifested under specific system-on-chip (SoC) operating conditions and their effects are difficult to predict a-priori. This paper describes recent advances in detecting and diagnosing such bugs using "guided" stochastic test stimulus generation algorithms. A key challenge is that unlike traditional test generation for manufacturing test that is predicated on *known* failure mechanisms, the nature of design bugs is generally *unknown* and must be discovered on-the-fly. Classes of design errors from undesired capacitive coupling and incorrect biasing conditions to incorrect guard-banding of designs are considered. It is shown that high design bug coverage can be obtained over a range of test cases.

I. INTRODUCTION

The last decade has seen tremendous advances in the transformation of desktop systems to ubiquitous and mobile computing and communication platforms that are "anytime, anywhere". With increasing human dependence on machines, as described above, it is becoming increasingly imperative that these machines and their underlying computing and communication platforms be *completely bug-free*. However, recent data shows that it is becoming increasingly difficult to verify all aspects of the correctness of a design pre-silicon. Current methods for post-manufacture bug localization and root cause determination rely on system level simulation/emulation based failure reproduction methods. This involves returning the system to an error free state and resimulating/emulating the system with the exact input stimuli (i.e., program instructions under identical voltage, temperature and frequency conditions). However, simulation is orders of magnitude slower than actual silicon and most state of the art emulation methods are *unable to reproduce the on-chip electrical conditions that activate electrical bugs*. Failure reproduction in mixed-signal/RF subsystems is limited primarily by simulation capability. The fastest simulators use behavioral models that trade off simulation accuracy for speed and the most accurate simulators are too slow to be useful for simulating mixed-signal/RF devices across diverse process conditions and input stimuli.

Scan chains are popular in digital design for providing state observability and controllability for system debug. There has been research on analog scan based design for debug. Although the analog scan standard IEEE 1149.1/1149.4 [1, 2] is in practice for board level debugging, it is not popular for circuit level testing of analog/RF IPs in SoCs. Current based analog

scan chain was proposed in [3, 4] by Soma et.al. Popular analog scan methods are based on: (i) voltage to frequency conversion and (ii) voltage to delay conversion [5]. In [5], the authors propose a voltage to delay conversion method based on voltage measurement. Two in-phase clock signals are delayed by a sampling voltage and a reference voltage respectively. A digital delay measurement unit measures the delay as a proxy of the measured voltage. Scan techniques such as above suffer from large layout area, data acquisition time and capacitive loading of the node that is monitored for test purposes. In [6] the authors propose an analog DFT technique that relies on Vdd ramping. The captured current signatures from various IP blocks are compared against a fixed threshold and multiple digital bits are generated as a signature of the test response. In contrast, analytical model based analog/RF post silicon validation has been proposed in [7, 8] and model learning based diagnosis is proposed in [9]. The problem with analytical model based validation is that accurate model development that captures physical electrical phenomena is practically impossible.

In the past, there has been research on test stimulus generation based validation in the digital space [10, 11]. However, there is not much parallel work in the mixed-signal/RF domain. In [12, 13], the authors use *diverse programs* with the *same functionality* to detect processor design bugs. A hardware design bug is detected if there is any inconsistency in the results obtained from the two functionally equivalent program streams. The test procedure does not make any assumption about the nature of design bugs and the extent of design bugs uncovered is limited only by the diversity of test programs deployed.

With regard to analog circuits, there are two classes of designs bugs: (a) those for which the design model encapsulates all circuit performance non-idealities but the model parameter values do not correspond to manufactured silicon and (b) those for which hardware non-idealities are unknown and therefore not included in the simulation model used for circuit design. In case (a), the circuit model is defined to be *complete* and in case (b) the model is defined to be *incomplete*. The post-silicon validation procedure for analog/mixed-signal circuits is to first determine if the model is complete, i.e. can the model parameters be tuned in such a way as to explain the observed device under test (DUT) behavior? If not, then the model is incomplete and must be amended to include model artifacts that subsume the non-ideal behavior observed in the hardware that cannot be explained on the basis of its simulation model. Determining whether a model is complete or incomplete

involves the use of directed (stochastic) test stimulus generation algorithms.

In Section II, we first present a model based validation technique that aims to determine if the simulation model for the DUT is complete. Sections III and IV present two techniques that make no assumption about model completeness but merely check for inconsistencies between observed and expected behavior using mostly on-chip vs. external tester based instrumentation, respectively. In Section V emerging techniques for learning about unknown hardware behavior that cannot be explained on the basis of simulation models is discussed, followed by a discussion of how validation algorithms can be speeded up and conclusions.

II. MODEL BASED VALIDATION

In earlier work [7, 8] model based validation of analog/RF circuits was introduced. The methodology aims at determining if there are behaviors in the DUT that cannot be explained by the behavioral model of the system. The objective of design validation is to first prove equivalence between the DUT model and the observed DUT behavior in silicon and then find the source of the design error in case it is determined that the DUT behavior is different from that predicted by its model.

A specially designed test derived from consideration of the DUT model is used to *simultaneously* stimulate the DUT and its model (running on an emulator/simulator). Any difference between the DUT response and the model response is treated as a *design anomaly signature*. If this difference cannot be brought down to a value below a specified threshold (defined by simulation accuracy) by perturbing the model parameters using known optimization methods for minimizing this error, then the obtained DUT response cannot be explained by its model and we conclude that the model is *incomplete*. We then invoke *model update* procedures and repeat the validation step until the model is determined to be complete. Under the assumption that the design anomaly is due to a single embedded DUT module, the approach proposed in this paper determines if the model is complete and also the specific embedded module that caused the anomaly in a *single step*.

Figure 1: IQ mismatch model building

In Figure 1 we show an example where an RF DUT exhibits IQ gain and IQ phase mismatch but this IQ mismatch was not included in the corresponding behavioral simulation model. In Figure 1, the order of amplitude-to-phase (AM to PM) and amplitude-to- amplitude (AM to AM) is increased across a set of experiments as shown, but is unable to eliminate the residual

error in the output signal obtained from test stimulus application. In Figure 2 we cite another example where a RF DUT exhibits AM-to-AM , AM-to-PM , IQ gain mismatch, IQ phase mismatch and DC offset effects. Figure 2 shows the residual error obtained from test stimulus application for different cases where the behavioral model includes only a subset of the device non-idealities that are manifested in hardware. It is seen that the residual error is very small only for the case where the model includes all of the non-idealities exhibited by the hardware implemented in silicon.

Figure 2: Model convergence

III. BUILT IN STATE CONSISTENCY CHECKING

Table 1: Formal Definitions

State Variable (SV)	Any observed node voltage or branch current is a state variable. $SV \in \mathbb{R}$ and $R_{i1} < SV_i < R_{i2}$ Where (R_{i1}, R_{i2}) is the dynamic range of the observed current/voltage corresponding to state variable i .
State Consistency	Temporal Consistency: Two state values SV_i^1 and SV_i^2 of state variable SV_i will be consistent with each other if $ SV_i^1 - SV_i^2 < \epsilon$ where ϵ is comparator offset voltage Spatial Consistency: Two state values SV_i^1 and SV_j^1 of state variables SV_i and SV_j will be consistent with each other if $ SV_i^1 - SV_j^1 < \epsilon$ where ϵ is comparator offset voltage

A formal definition of state variable and state consistency is given in Table 1. Intrinsic as well as parasitic device/interconnect resistance, capacitance and inductance make any RF/Analog circuit a state machine. Number of possible states in such state machine is infinite as analog currents/voltages can take any values within the dynamic range. Here any branch current or node voltage can be thought of as state variable of the system. Exciting the above said state machine with an input stimulus, takes the state machine from one state to another through various intermediate states. If two implementations of the same functional circuit are different, then under the same input stimulation the state space traversal (going from initial state to final state) would be different. If the state variable value changes appreciably then it is possible to discern between the above two implementations of the same functional circuit, by observing the state values. This work is based on the above said state machine representation of any RF/analog circuit.

Temporal State Consistency Checking (Type I Test): In our approach a piecewise linear stimulus of duration T is crafted across a time grid of spacing δ and N grid points where $T =$

$N\delta$ (see Figure 3). At time $t=T$ the final value of the state variable SV in response to stimulus S1 is sampled and held using a sample and hold (S/H) circuit for additional time T . Between time $t=T$ to $t=2T$, a different stimulus S2 is applied to the DUT starting with the same initial condition as in S1. Final value of the state variable SV in response to S2 is acquired at $t=2T$ using a S/H circuit. Subsequently the sampled values of SV at $t=T$ and $t=2T$ in response to the applied stimuli S1 and S2 respectively, are compared. If they are consistent then a logic “0” is generated by the error triggering circuit else a logic “1” is generated.

The core idea behind temporal state consistency checking is to design the stimuli S1 and S2 in such a way that they are diverse (exercise the analog/RF circuit through diverse state trajectories) but result in consistent final states sampled at $t=T$ and $t=2T$. This leads to the hypothesis that an arbitrary design bug or fault is unlikely to affect the state trajectories of SV in response to S1 and S2 identically and thereby results in inconsistent final state values.

Figure 3: Temporal State Consistency

Spatial State Consistency Checking (Type 2 Test): Type II tests are designed to check consistency across state variables at same sampling instant as opposed to type I test where state consistency of a state variable is checked between two different sampling instants. A stimulus will be generated so that the two observed state variables are consistent for nominal circuit at sampling instant $t=T$ (see Figure 4). It is to be noted that two observed state variable pair may be having different dynamic ranges, so proper level shifting and gain compression are required before comparison. There are some pathological cases such as gain compression, DC offset where the faulty circuit may show state consistency under type I stimuli test. To catch these faults we introduce type II tests, where state consistency across state variables are checked.

The above technique has been applied to several circuits including an RF transceiver. For the RF receiver, individual transistor widths, threshold voltages, resistance, capacitance, inductance values and bias voltages (total 30 parameters) were randomly varied to create “good” vs. “bad” circuits (as defined by a set of given RF specifications). Random capacitive and resistive opens and shorts were introduced into the LNA and mixer netlists to create design defects circuits (the method also detects manufacturing failures). Over the complete set of Type 1 and Type 2 tests applied, 100% failure coverage was obtained with minimal yield loss.

Figure 4: Spatial Consistency Checking

IV. TEST STIMULUS GENERATION: RAVAGE

Since the exact nature of behavioral discrepancies between fabricated silicon and a given simulation model of the DUT is not known a-priori, tests are generated by *stimulating the DUT model as well as its hardware implementation concurrently* as shown in Figure 5. The RAVAGE algorithm [14] begins with a population of arbitrary band-limited stimuli and employs a genetic algorithm to evolve the population (within several constraints) in such a way that it best excites behaviors in the DUT that the DUT’s model does not exhibit. Whether due to structural or topological inadequacy of the model (equivalent to unexpected circuitry in the DUT), lack of complexity in the model, or process variation in the DUT (equivalent to incorrect model parameters), RAVAGE will favor those stimuli which best expose differences, regardless of their origin. If stimulus performance improvement becomes static, RAVAGE attempts to tune its model to such that the observed discrepancy is minimized. The stimuli used for tuning are then put aside, the entire population is reinitialized with random seeds, and the process repeats. If RAVAGE can tune the model sufficiently, such that error power never exceeds some error-noise floor; then the model can be treated as complete, both behaviorally and parametrically. The overview of the RAVAGE algorithm is shown in Figure 5.

Figure 5: RAVAGE System Overview

Using a maximum stimulus bandwidth of 5 MHz, and an error bandwidth of interest of 18 MHz, RAVAGE was allowed to run for 623 generations on a simulation model (Model in Figure 5) of a mixer and Maxim MAX2039 high linearity up/down-conversion mixers (DUT in Figure 5). Computation time was approximately 1s per generation. Fig.6 shows the test

generation setup. Figure 7. shows the fitness and total error power of the population through 60 generations of the experiment and Fig.8 shows a frequency-domain plot of the most effective stimulus. Maximum total error power rose within 30 generations to over 95% of its final value of 0.6. This gives an example of the sensitivity of the system.

Figure 6: Test Generation Setup for RF System

Figure 7: Error and Fitness Function vs. No of Generations

Figure 8: Spectrum of Generated Test Stimulus

V. LEARNING BASED VALIDATION

It is possible to discover and learn about unknown hardware behavior using a collaborative test generation vs. learning approach. We propose an approach in which the DUT (hardware) and its design model (augmented model in Figure 9) are stimulated with input signals generated using a random process. Initially, the learning kernel of Figure 9 is an all-pass unity gain transfer function. The process starts off with random stimulus and iteratively refines the stimulus using a stochastic optimization algorithm (genetic algorithm in Figure 9) to *amplify observed differences* between the DUT and the model response. Such differences are analyzed to determine the existence of bugs in the mixed-signal design. A key problem that happens with this kind of approach is that many designs have multiple design nonidealities and in general, specific types of design nonidealities can make others hard to find. For

example, the presence of gain mismatch in a design can make small phase mismatch effects hard to detect and debug. To alleviate this problem, the strategy adopted is to perform test generation in several passes. In each pass, a supervised learning algorithm (neural net learning kernel in Figure 9) inserted into the DUT design model is trained to minimize or cancel out the observed differences between the DUT and its augmented model. A key benefit of this approach is that at the end of the validation procedure, the learning kernel encapsulates all the discovered nonidealities of the mixed-signal DUT. This facilitates subsequent design debug.

Figure 9: Collaborative Test Generation and Behavior Learning

Figure 10: Model Error Vs. Neurons Added Under Test Generation Iterations

Figure 11: PA Transfer Function: Device Under Validation, Augmented Model and Ideal (Unaugmented) Model

An RF transmitter system similar to that depicted in Fig.6 was setup in hardware using Maxim Semiconductor MAX 2039 up- and down-converting mixers with a Maxim MAX2242 power

amplifier in the RF portion of the chain. The system model was initialized using the manufacturer's supplied conversion gain for the mixers and the PA's nominal linear gain. Iterative stimulus was begun, and error measurements were taken throughout (See Figure 10). In total, 8 stimuli were found and 34 neurons were added and trained within the model. Figure 11 presents the pre-silicon assumption of behavior, the Device Under Validation (also referred to as DUT earlier, DUV) behavior and the augmented model's behavior, and Fig.12 shows the concatenated stimuli after the 8 iterations.

Figure 12: Cumulative Validation Test Stimulus After 8 Iterations

VI. SPEEDING UP VALIDATION: BOOLEANIZED MODELS

Karthik and Roychowdhury have recently developed an approach called ABCD for synthesizing FSM descriptions of linear and nonlinear analog circuits [16,17]. This approach allows simulation of *mixed-signal/analog* circuits up to 2 orders of magnitude faster than direct Spice simulation although there is an up-front cost of generating the models from Spice transient simulation runs. *By replacing the simulation models in Figures 5 and 9 with booleanized models, validation can be speeded up by similar degrees.* We have adapted ABCD to allow FSM model generation (booleanization) of RF circuits. Further, we are able to generate such FSM models from *direct stimulation of hardware* rather than from Spice simulation runs, *speeding up the model generation process by at least 3 orders of magnitude.* While a detailed description of the modified version of ABCD called ABCD-RFH is beyond the scope of this paper, key details are given below.

Figure 13: (a) Input and output transition (b) State transition table

RF FSM Model Generation: In ABCD [16,17], the FSM states correspond to specified transient as well as DC steady state values of the analog circuit inputs with circuit output values associated with every state. Transitions between states are defined by input transitions between quantized levels of the input dynamic range. Since RF circuits are DC-blocking, the above definition of the states of analog circuits needs to be

modified to allow RF FSM model generation. In [18] it is shown how the non-idealities of an RF device map onto the baseband signal obtained by RF demodulation. In this work, we assume that an envelope detector is used for this purpose. In our algorithm, *each DC state of the FSM represents the magnitude (quantized) of the baseband signal and is associated with the corresponding output value of the demodulated (baseband) signal* (see Figure 13).

Incorporation of memory effects: Given the above descriptions (transient, DC) of the states of the FSM model, ABCD constructs transitions between the states of the FSM model by directed Spice (transient) simulation. While a transition is in progress from one DC state to another, it is possible that the input experiences a transition before the circuit output has settled to its new value. This is handled in ABCD [16,17] using transient “jump” states that define the sequence of transitions the circuit output goes through while “moving” from one DC state to another. To capture such jump states *the complete transition between the respective DC states needs to be simulated.*

In ABCD-RFH, the FSM model and the DC/transient states are created by replacing Spice with direct hardware stimulation. However, because the baseband data rate is known, the DUT need not be stimulated beyond the duty cycle of the input data rate while transitioning from one DC state to another. However, this creates a problem. As opposed to ABCD in which all transitions from other states to a known DC state result in the *same* output value, in ABCD-RFH, because the DUT is stimulated only within the duty cycle of the input data rate, transitions from other states to a known DC state can result in *different* output values. To accommodate this, we *split the corresponding DC state into multiple DC states* depending on how that state is reached from other FSM (DC) states.

Extraction of FSM Models from Hardware: By replacing Spice simulation with directed hardware stimulation, the cost of model generation incurred by ABCD is virtually avoided. Moreover, these models can be used to simulate how devices behave *even after the latter have been shipped to customers*, a capability that *does not exist with any other tool today.*

The hardware setup for this experiment is shown in Figure 14. In this experiment, the Device under Test (DUT) is a LNA (X3533 7AZCYH3) mounted on a load board. A high frequency signal generator is used to generate LO signal (2.4GHz -10 dBm). Low frequency envelop signal is generated by Labview and imparted into the circuit by NI data cards. The modulated signal coming from up conversion mixer is used to stimulate the LNA. Output envelope of the LNA is captured by an envelope detector (ADL 5511). Envelope detector output voltage and rms power output constitute signature of the device in this work. Envelope detector output is feed to NI digitizer and data is captured. From extracted data, envelope models are generated by Matlab. For model generation the input slope and the minimum time required to hold it at the current level to stabilize the output response is critical (shown in Figure 14). If the model is generated for input slope T_{slope} then in test generation no transition steeper than T_{slope} is permissible. This input slope in hardware is constrained by DAC capability.

Hold time (T_{stay}) cannot be predicted a priori so some repeated trials are required. In this hardware experiment T_{slope} and T_{hold} are taken as $1e-5$ and $4e-4$ second respectively. The accuracy of the FSM model for a random baseband input is shown in Figure 15..

Figure 14: Hardware experiment setup

Figure 15: Extracted FSM model accuracy

VII. CONCLUSIONS AND FUTURE WORK

In this research, we have presented a framework for post-silicon validation of mixed-signal systems. Collaborative test generation and learning plays a significant role in our proposed approach and has been shown to yield great benefits both in validation speed as well as accuracy. Future research is focused on validation of complex mixed-signal SoCs with a plethora of circuit types and sensors and co-validation of mixed-signal systems with digital in a unified validation framework.

REFERENCES

- [1] B. Suparjo, A. Ley, A. Cron, and H. Ehrenberg, "Analog Boundary-Scan Description Language (ABSDL) for Mixed-Signal Board Test," in *Test Conference, 2006. ITC '06. IEEE International*, 2006, pp. 1-9.
- [2] "IEEE Standard for a Mixed-Signal Test Bus," *IEEE Std 1149.4-2010 (Revision of IEEE Std 1149.4-1999)*, pp. 1-116, 2011.
- [3] M. Soma, "Structure and concepts for current-based analog scan," in *Custom Integrated Circuits Conference, 1995., Proceedings of the IEEE 1995*, 1995, pp. 517-520.
- [4] M. Soma, T. M. Bocek, T. D. Vu, and J. D. Moffatt, "Experimental results for current-based analog scan," in *Test Conference, 1997. Proceedings., International*, 1997, pp. 768-775.
- [5] R. Vasudevamurthy, P. K. Das, and B. Amrutur, "A mostly-digital analog scan-out chain for low bandwidth voltage measurement for analog IP test," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, 2011, pp. 2035-2038.
- [6] A. Zjajo, H. J. Bergveld, R. Schuttert, and J. P. d. Gyvez, "Power-scan chain: design for analog testability," in *Test Conference, 2005. Proceedings. ITC 2005. IEEE International*, 2005, pp. 8 pp.-83.
- [7] A. Chatterjee, S. Deyati, B. Muldrey, S. Devarakond, and A. Banerjee, "Validation signature testing: A methodology for post-silicon validation of analog/mixed-signal circuits," in *Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on*, 2012, pp. 553-556.
- [8] S. Deyati, A. Banerjee, B. J. Muldrey, and A. Chatterjee, "VAST: Post-Silicon Validation and Diagnosis of RF/Mixed-Signal Circuits Using Signature Tests," in *VLSI Design and 2013 12th International Conference on Embedded Systems (VLSID), 2013 26th International Conference on*, 2013, pp. 314-319.
- [9] S. Deyati, B. J. Muldrey, A. Banerjee, and A. Chatterjee, "Atomic model learning: A machine learning paradigm for post silicon debug of RF/analog circuits," in *VLSI Test Symposium (VTS), 2014 IEEE 32nd*, 2014, pp. 1-6.
- [10] B. Querbach, R. Khanna, D. Blankenbeckler, Y. Zhang, R. T. Anderson, D. G. Ellis, et al., "A reusable BIST with software assisted repair technology for improved memory and IO debug, validation and test time," in *Test Conference (ITC), 2014 IEEE International*, 2014, pp. 1-10.
- [11] X. Shi and N. Nicolici, "On-chip generation of uniformly distributed constrained-random stimuli for post-silicon validation," in *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on*, 2015, pp. 808-815.
- [12] N. Oh, P. P. Shirvani, and E. J. McCluskey, "Error detection by duplicated instructions in super-scalar processors," *IEEE Transactions on Reliability*, vol. 51, pp. 63-75, 2002.
- [13] D. Lin, T. Hong, Y. Li, S. Eswaran, S. Kumar, F. Fallah, N. Hakim, D. Gardner and S. Mitra, "Effective Post-Silicon Validation of SoCs Using Quick Error Detection," *IEEE Transactions on Computer-Aided Design*, Vol. 33 Issue 10., 2014, pp. 1573-1590.
- [14] B. Muldrey, S. Deyati and A. Chatterjee, "RAVAGE: Post-silicon validation of mixed-signal systems using genetic stimulus evolution and model-tuning
- [15] B. Muldrey, S. Deyati and A. Chatterjee, "Post-Silicon Validation of Analog/RF Designs Using Cooperative Test Generation and Behavior Learning Using Sparse Wiener Neural Kernels," *DAC Poster Presentation*, 2015.
- [16] Karthik and J. Roychowdhury, "ABCD-L: Approximating continuous linear systems using Boolean models," in *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, 2013, pp. 1-9.
- [17] A. V. Karthik, S. Ray, P. Nuzzo, A. Mishchenko, R. Brayton, and J. Roychowdhury, "ABCD-NL: Approximating continuous non-linear dynamical systems using purely Boolean models for analog/mixed-signal verification," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, 2014, pp. 250-255.
- [18] R. Voorakaranam, S. S. Akbay, S. Bhattacharya, S. Cherubal, and A. Chatterjee, "Signature Testing of Analog and RF Circuits: Algorithms and Methodology," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, pp. 1018-1031, 2007.