Circuit Partitioning for Behavioral Full Chip Simulation Modeling of Analog and Mixed Signal Circuits

Samantha Alt, Malgorzata Marek-Sadowska, and Li. C. Wang

***Abstract*—This paper presents a new method for automatically creating transient voltage behavioral models of analog and mixed signal circuits based on functionally independent partitions. The models are obtained using Support Vector Machines (SVM), a data dependent black box modeling technique. Larger circuits are partitioned based on their structure and intermediate behavioral models are built for high sensitivity nets. The models are implemented in System Verilog for use in full chip system validation. We demonstrate the soundness of this approach by modeling large and highly non-linear circuits such as Sigma-Delta Analog-to-Digital (ADC) and jitter of a Phase Lock Loop. Experimental results show 95% accuracy behavior predictions and three orders of magnitude speedup over SPICE simulation time.**

***Index Terms*—Circuit partition, support vector machines, channel connected component graph, behavioral models.**

I. Introduction

Full chip verification of complex mixed-signal system on chip (SoC) is necessary to cover possible complex analog and digital interactions which may lead to functional errors [1]. It was reported in [2] that some design bugs occur due to system level PVT, noise, and un-modeled system level interactions. Due to the complex interaction of these circuits, independently verifying analog components is no longer sufficient for full system verification. Unfortunately performing circuit-level simulation of a whole system with interacting analog and digital components is not possible due to excessive run times. Utilizing the real number system inherent in System Verilog, it is now possible to create analog behavioral models that can be simulated with a digital simulator. To do so, it is necessary to replace the analog circuits with accurate behavioral models extracted directly from their transistor level netlists.

Many behavioral techniques model the circuit frequency domain response and a comprehensive overview can be found in [3]. Here, we focus on time-domain transient modeling for non-linear analog and mixed signal circuits which can be accomplished by equation fitting or black box modeling. Time domain modeling is important for system level validation since digital components are simulated sequentially in discrete time steps. Any analog behavioral model needs to operate under the same discrete time steps as the rest of the system.

Black box models describe a circuit or system in terms of its inputs and outputs without any knowledge of its internal workings. Such models can be derived strictly from data, measured or simulated, with no in-depth circuit understanding. The models can be applied to known or unknown circuits whose equations may not have been developed.

Time-domain black box behavioral modeling methods have been proposed in the past, but they suffer either from lack of automation, provide minimal speed-up over traditional methods, cannot be applied to larger circuits, or cannot be implemented within digital simulation environments. In [4], the authors propose to extract behavioral models directly from the netlists by processing simulation results into transfer function trajectories. The method was demonstrated on a small circuit with minimal speedup and the approach is not always automatable. The work in [5] presents a real-time neural-network-based approach for microwave RF devices. The approach is demonstrated on small circuits and it is unclear if it can be applied to large and complex designs.

The methods proposed in this paper build upon the work described in [6]. There, the authors use Gaussian Process Regression to build non-parametric models thus eliminating the need to use and understand complex circuit models and equations. Non-parametric behavior modeling is extremely attractive because there is no need to learn equations or set parameters since the device behavior is predicted based on similar known behaviors. Unfortunately, applying [6] directly to large, more complex circuits produces poor results. Inspired by this work, we propose a partitioning method along with support vector machine (SVM)-based approach to create a hierarchy of behavioral models for large, strongly non-linear analog circuits. The partitions are derived from a channel connected component graph and hierarchy graph which determine the input/output relationships and hierarchy of partitions. Simulations are run with SPICE on the full un-partitioned netlist in order to ensure continuity between partitions. Once the data has been extracted from the simulations, the black box models are created based on derived partitions. The models are connected to form a series of predictions that create a transient output waveform. The models are real number based, and can be converted easily into System Verilog. Though we use SVM as the primary modeling algorithm, many other black box approaches, such as response surface modeling, can benefit from the reduced complexity associated with the proposed partitioning scheme.

This paper is organized as follows. Section II motivates this work. Section III provides background on SVM modeling and data capture techniques. Section IV describes the circuit graph representation, net sensitivity calculations, and modeling-resistant circuit components. Section V describes circuits used in experiments. Section VI shows experimental results. Section VII concludes the paper.

Manuscript received September 29, 2013; revised November 23, 2013.

The authors are with the University of California, Santa Barbara, CA 93107 USA (e-mail: {salt,mms,licwang}@ ece.ucsb.edu).

II. Motivation

Analog circuits vary vastly in their complexity and behavior. For this reason a universal solution has not been developed for automating behavioral models. The existing modeling methods cannot be easily automated because they either 1) require design or designer knowledge to develop sub-circuit equations, 2) sub-circuits are used to speed up simulation of the model development thus the continuity between sub-blocks is difficult to maintain, and 3) input-output relationships of the functional blocks are often too complex or have one-to-many relationship making black box modeling difficult or unachievable. Analog design is usually partitioned into well-known sub-block components like quantizers, VCOs, or charge pumps which can all be modeled by equations or have data fitted to equations. Unfortunately many of these methods fail to capture block to block interactions or they cannot be expanded to include process variation due to the increased complexity of the equations. Such equations will need to be developed for every type of a new circuit or new behavior type encountered.

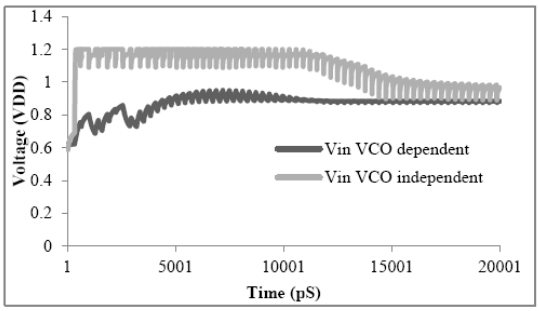

Fig. 1. Charge pump transient simulation results for independent and dependent simulations until lock.

In Fig. 1 we show how a circuit block differs in its generated output when it is simulated within the full PLL system and when simulated independently. Both simulations have the same initial conditions for the charge pump block and the inputs to the charge pump are extracted from the dependent simulation so that each block is simulated with the same inputs. The figure shows the voltage value generated by the charge pump. This value controls how much to increase or decrease the generated frequency to match the reference frequency. The difference between the two waveforms is caused the block to block interactions such as current drawn from other blocks.

Black box-based models can easily capture strongly correlated relationships of simple sub-circuits automatically, but fail when complex or weak relationships are encountered. To illustrate this point we analyzed a quantizer circuit defined as a single sub-block within a large sigma-delta analog-to-digital converter (ΣΔ-ADC) [7]. This circuit contains 40 transistors and has three inputs and a differential output. Taking two common black box approaches, the response surface modeling (RSM) and support vector machines (SVM), we build transient behavioral models based on 1000 points of transient simulation data for a single period of the input frequency and test them with another set of 1000 points. Both methods predict output transient waveforms with low accuracy: RSM - 49.6% and SVM-75.82%. This indicates that the input-output relationships are not highly correlated.

To address weakly correlated relationships, we break the netlist into functionally-independent, strongly correlated sub-circuits. Data capture for model building is done from simulations performed on the un-partitioned netlist in order to preserve the continuity between sub-circuits. The graph partitioning methodology and high sensitivity net detection removes the requirement for designer knowledge and allows for modeling unknown, flat netlists, or new circuit structures.

III. Modeling and Data Capture

Within the domain of behaviors studied here, RSM and SVM models are comparable in speed and accuracy. We choose SVM-based models because of their ability to handle a large number of model inputs and to discover trends based on small or irregular sample sets. We note that any black box modeling methodology would benefit from our partitioning algorithm due to reduced complexity of the sub-circuits.

Our models apply supervised learning algorithms, *Support Vector Machine Regression* (SVM-R) for real values (analog outputs) and Support Vector Machine Classification (SVM-C) for digital outputs [8]. The objective of supervised learning is to derive a function from a set of input samples (*training set*) and their associated outputs. SVMs are well studied and well understood and will not be discussed in detail here. A comprehensive review of SVMs can be found in [8].

SVM utilizes kernels to compare vectors in the original feature space. In this work, we use the Gaussian kernel expressed as

$$k(x,x') = \exp\left(-\frac{\left\|x-x'\right\|^2}{2\sigma^2}\right), \text{where } \sigma > 0 \qquad (1)$$

where x and x' are vectors and σ^2 is the variance. Other kernels, such as linear, polynomial, or sigmoid can be used, but experimentally, we found that the Gaussian kernel provides the best results for the applications considered here.

The SVM algorithm utilizes data obtained from low level models such as SPICE or differential equations to create a high level model abstraction that captures a specific behavior. A circuit can be completely described by differential equations

$$\frac{d}{dt}\vec{q}\left(\vec{u}(t)\right) + \vec{f}\left(\vec{u}(t)\right) + \vec{B}\left(\vec{x}(t)\right) = 0, \qquad (2)$$
$$\vec{y}(t) = \vec{l}^T\vec{u}(t) + \vec{d}^T\vec{x}(t),$$

where $\vec{u} \in R^n$ are state variables, i.e. capacitor currents, $\vec{x}$ are top level inputs, and $\vec{y}$ are the top level outputs. Obtaining a unique output $\vec{y}$ requires access to the state and input variables. At a high level of abstraction, black box, the internal state variables are no longer accessible. Therefore unless the inputs $\vec{x}$ are highly correlated to the outputs $\vec{y}$ the models will produce very low accuracy results. Access to $\vec{u}$ be required to produce high accuracy models. To determine which state variables are required we propose a heuristic which partitions the circuit and determines highly

correlated intermediate behaviors to the top level output. Simulations are performed to capture the behavior relationships between inputs $\vec{x}$ and the intermediate behaviors captured in SVM models. The output $\vec{y}$ can be determined based on top level input $\vec{x}$ and intermediate models.

SVMs may require long training time for complex data sets, but there exist sampling and data partitioning methods that can be used to create smaller, more compact models which significantly reduces the training time. The work in [9] explores various methods for data sampling to achieve high SVM model accuracy with fewer simulation runs.

A. Data Capture

Given the design specifications, input operation ranges for the desired performance, and the circuit netlist, the required data for building behavioral models can be obtained. In this work, we only consider transient voltage behaviors. Before any partitioning is done the entire netlist is simulated as a single entity and the transient voltage behaviors are captured for each net at uniform intervals. We perform uniform sampling within the range of the design specifications and simulate to capture the behavior.

The uniformity of samples may cause model inconsistencies around highly volatile areas which traditionally require increased sampling. It is not known beforehand whether more data is needed for specific areas or for intermediate behavior models. To address this problem, data is captured at finer intervals then required in the design specifications, which does not increase simulation time. The models are initially built based on the original intervals. If testing step discovers inaccuracies in a specific model, the models are rebuilt based on the finer interval data. This data capture method is rarely required; it may be invoked when the output behavior is state dependent and highly volatile. When capturing smaller intervals, the waveform changes become less extreme, making the behavior relationships easier for the SVM algorithm to discover.

Probes are inserted at each edge defined by the partitions in Section III for voltage capture. This way the system level input and output relationships of each partition are maintained without any extra modeling for block coupling. Our models are specific to the simulated design and cannot be utilized in a different design model unless the input and output relationships between the designs are maintained.

B. Model Creation

The regression function can be expressed as

$$y = w \cdot \Phi(x) + b \tag{3}$$

where b is the bias, w is the weight, and $\Phi(x)$ denotes the feature of the inputs. The weight vector is the vector orthogonal to the optimal hyperplane representing the decision boundary [8] found by minimizing the risk function:

$$\begin{gathered} \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{Subject to} \quad y_i - [(w, x_i) + b] \leq \varepsilon + \xi_i \\ (w, x_i) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i \geq 0, \xi_i^* \geq 0 \end{gathered} \tag{4}$$

where C is the regularization constant, ε denotes the ε-insensitive coefficient, ξ_i, ξ_i^* denote positive slack variations. This vector and its direction provide the prediction point of any new vector x by performing the dot product, Equation 3. The importance of features (elements in vector x) correspond to the entry values of the weight vector.

Due to the decomposition of the circuit each partition has a strong input-output correlation resulting in a weight vector having a non-zero value for each feature. The number of partitions per circuit is generally high for large analog circuits. Classification models for digital partitions remove most of the real number errors in the regression models by predicting the digital values 0 or 1.

Circuit state and feedback characteristics can easily be incorporated into the vectors. The previous output value of the circuit being modeled is included as a feature in the vector describing the current output value. States are incorporated in the same way, as features within the vector. When applying test vectors to the model, the previous predicted output value is included within the current test vector.

Parameter selection is a difficult problem for many learning algorithms and is crucial for creating a good model. In this work we use the υ-SVM-R algorithm which automatically fits the user-defined values ε and C based on the provided data. The only value the user defines is the variable υ which determines the slack variables. The larger the υ value the more vectors with slack ξ may become support vectors. In this work we use υ=0.1 for all regression models.

IV. Partitions and Intermediate Behaviors

A common approach to addressing slow transistor-level simulation times is to partition the circuit into its digital and analog components. A major challenge in this type of partitioning is to determine when the digital components are in analog modes of operation, which include detection of the internal and negative feedback. The work in [10] provides an algorithm for such partitioning using channel-connected sub-circuits. In [11], the analog sub-circuits are simulated using nonlinear macromodels. The macromodels approximate the charging and discharging behavior of the circuit output nodes and provide one order of magnitude speedup over SPICE for analog subcircuits. The macromodeling using circuit partitioning proposed in [12], [13] decompose the circuit into building blocks based on the netlist structure. Methods that use subcircuit or building block-based partitions assume that either the user declares the partitions or that predefined building blocks is given. It is not always the case with complex or custom built circuits which makes modeling using these methods difficult.

In our approach, the graph model of the circuit and sensitivity analysis of captured data are used for establishing the signal flow, hierarchy, and determining which signals are essential to predict intermediate behaviors. In this Section we discuss the two types of graphs used in partitioning and explain how the intermediate nodes are determined based on net sensitivity. We then discuss circuits which are resistant to this type of partitioning and propose a method of handling them. We use the quantizer

circuit as an example throughout this explanation.

A. Channel Connected Graph and Channel Connected Components

A Channel Connected Graph (CCG) [11] describes the source-drain dependencies within a circuit. It is defined as $CCG= (V, E)$ where V is vertices corresponding to transistors, V_{dd} or GND, and E are edges capturing the source-drain connections. There is an edge between vertices v_1 and v_2 if their corresponding transistors have drains or sources connected or if one of them represents V_{dd} or GND and the other transistor connected to it. Once the V_{DD} and GND nodes are removed, CCG is fractured into a set of smaller graphs. Each such a sub-graph with V_{dd}/GND and connections to them restored corresponds to a channel connected component graph (CCCG).

Fig. 2 shows the schematic of a 1-bit quantizer used in the ΣΔ-ADC described in Section V(a) and used as a motivational example in Section II. Fig. 3 shows the CCG graph of this circuit. The schematic diagram is not drawn to mimic the graph partitioning, but to show the components and the connections. The graph of the circuit is partitioned into six groups which are labeled based on the signal flow from inputs V_p and V_n (Group 1) to outputs V_{op} and V_{on} (Groups 5 and 6).

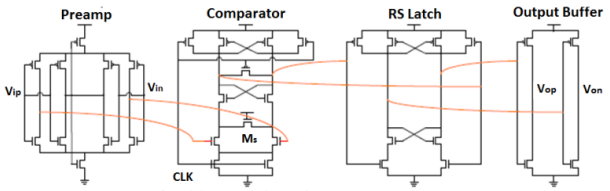

Fig. 2. Schematic of 1-bit quantizer for ΣΔ-ADC. The labeled sub-blocks are connected from left to right by the output of the previous block.

In analog circuits, it is possible that some resulting components do not contain V_{DD} or GND. Such sub-graphs are not CCCGs and may occur when a circuit contains floating nets, resistors, capacitors, or inductors that supply connections to the source or drain terminals. Partitions which are not CCCGs are discussed in part *D* of this Section.

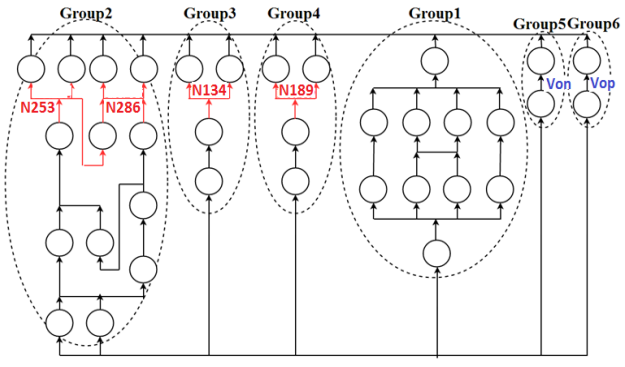

Fig. 3. Channel connected compont graph of the quantizer circuit. Six group partitions with intermediate behaviors.

B. Group Hierarchy and Feedback Detection

Each partition within a *CCG* has an associated set of driving gate inputs which can be the top level inputs to the circuit, internal nets, or nets from other partitions. The relationships between partitions can be described by a directed graph $HG= (N, L)$ where N is the set of partitions, top level inputs t, primary outputs o; and L are edges. An edge l (t, b) exists in *HG*, if there is an input to b from the top level t. An edge l (a, b) exists if the output of a partition a is an input to the partition b, referred to as an *external input*. Self-loops, edges with the same head and tail nodes, l (a, a), indicate *internal inputs*. The quantizer contains inputs t, Vn and Vp, outputs o, Vop and Von, and no internal nodes.

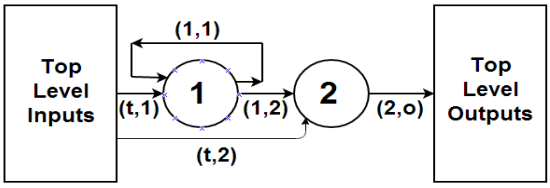

Fig. 4. Hierarchy graph of a circuit with feedback, internal, exteral, and top level inputs and outputs.

The graph *HG* determines the circuit hierarchy. The first level of the hierarchy contains t and those nodes with no in-coming edges other than self-loop. The remaining hierarchical levels can be determined by performing breadth-first traversal of the graph *HG*. The partitions containing self-loops indicate state dependency due to the feedback cycle between the gate-drain or gate-source connection.

Fig. 4 depicts an example hierarchy graph for a circuit containing two groups, one with feedback. The *internal input*, feedback, is shown as edge l (1, 1). *External input* is indicated *by* edge l (1, 2) which connects the two groups. Node 1 contains no external edges, only top and internal inputs, making it the top of the hierarchy. Node 2 is on the second level due to external edge l (1, 2). Each node in *HG* is translated into a SVM model where the node inputs represent the vector features, or dimentions, and the node output represents the output value. The model for Node 1 is solved first using the top level inputs and feedback, and then the predicted output is used as an input to the model for Node 2 which predicts the top level output value.

C. Intermediate Net Detection

Intermediate behavioral nodes are determined based on CCCGs, hierarchy graph, and sensitivity analysis performed on initial simulation data. We define net sensitivity as a change in voltage of the net to the change in voltage of the output, $NS=\Delta V_{net}/\Delta V_{out}$. We find the nets that have the greatest effect on the output behavior. The nets are sorted based on their sensitivities. The most sensitive nets are considered for use as intermediate nodes. If there are no high sensitivity nets within a specific CCCG, then that group is combined with the next group in the hierarchy. The exception is if the group contains an external edge to the top level output l (b, o), i.e. Group 5 and 6 in Fig. 5. If a partition contains a high sensitivity net then the intermediate behaviors are the nets, edges l (n, n+1), correspond to *external inputs* in the hierarchy graph. The quantizer design contains five high sensitivity nets based on the simulation data but only four, highlighted in Fig. 4, are utilized because they are external inputs. Since Group 1 does not contain any high sensitivity nets, it is combined with Group 2 which is next in the hierarchy. Fig. 5(a) shows the initial *HG* of Fig. 3. Fig. 5(b) shows the reduced graph which combines groups without high sensitivity nets.

The model of each partition is based on simulation data

which encompasses the behavior changes occurring due to internal variables $\vec{u}$. Each model represents functions dependent on only previous model outputs or top level inputs. The top level output behavior is then predicted based on few highly correlated internal nodes and the top level inputs.

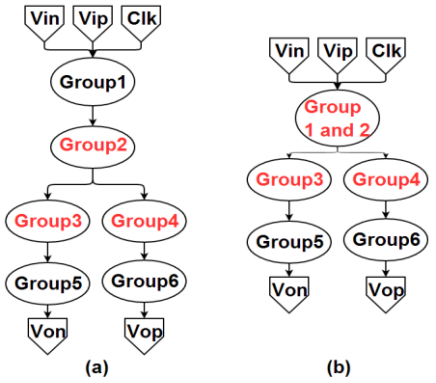

Fig. 5. (a) The initial hierarchy graph of the quantizer. (b) Hierarchy graph with combined non-sensitive groups Group 5 and 6 contain output edges.

D. Partition Resistant Components

Six models were created for the quantizer circuit, trained with 1000 vectors and tested with 1000 different vectors. For the intermediate nodes or final outputs producing real values, support vector regression was used. The input vector consists of circuit inputs and previous group outputs applied as shown in Fig. 5(a) and Fig. 5(b). The outputs have 100% prediction rate which is significantly better than the results prior to partitioning at 75.82%.

In certain instances a circuit or a sub circuit may not form a CCCG; such a partition may be missing the V_{DD} or GND node. Such a circuit is considered resistant to partitioning and must be modeled based on the circuit inputs/outputs without any intermediate nodes. For example, a switch capacitor network does not form a CCCG because the source or drain terminals of the nodes are connected only via capacitors which are excluded from the construction of the CCG.

The structures resistant to partitioning are detected at the netlist partitioning stage and are not decomposed. They are contained within a single node in the hierarchy graph.

V. Circuits

For experiments, we use compact low-power low-noise neural recording wireless channel circuits for high density neural implants [7]. We discuss the second order ∑Δ-ADC and a phase lock loop (PLL).

A. Sigma-Delta

A low-power, high-resolution ADC converter is required to digitize neural data. A 2nd order ∑Δ-ADC, shown in Fig. 6, can provide the required 6-bit resolution without consuming much power. The design consists of two fully-differential self-biased amplifiers, two switched capacitor networks, a 1-bit quantizer (used as an example in Section IV), and a non-overlapping clock generator. This circuit contains 647 components and takes 64 minutes to simulate two clock periods using HSPICE.

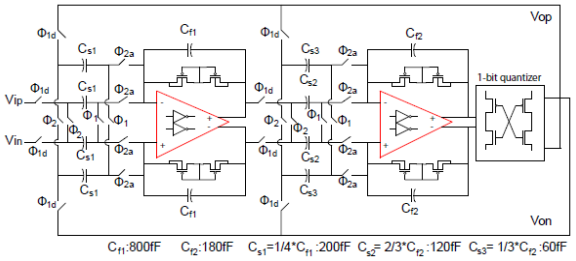

Fig. 6. Schematic of the low power high-resolution second order ∑Δ-ADC for digitizing neural data.

B. PLL

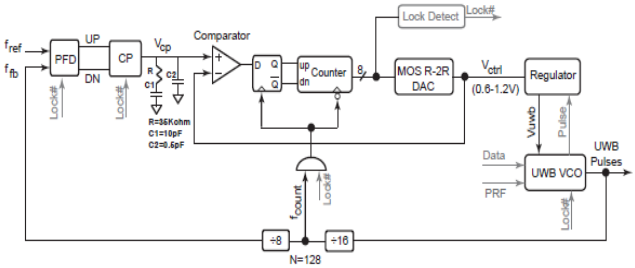

Fig. 7. Block diagram of the UWB-PLL with embedded digital tracking used to tune oscillation frequency of IR-UWB transmitter

The UWB-PLL in Fig. 7 is used to tune the frequency of the impulse radio ultra wideband transmitter. The UWB-PLL is designed with an 8-bit Up/Down/Hold counter and an 8-bit MOD-only digital-to-analog converter to general V_{ctrl} instead of directly connecting the loop-filtered charge-pump output voltage V_{cp}. A comparator forces V_{ctrl} to track V_{cp} and thus the PLL operates normally but has a digital tracking loop. If the counter does not change its output code for 6 consecutive F_{ref} cycles, the PLL locks.

This circuit contains 1892 circuit elements and takes 117 minutes to simulate in HSPICE. With parasitics incorporated into the netlist file the circuit contains 60830 circuit elements and takes 7 hours, and 8 minutes time to simulate.

VI. Experiments

The partitioning, data capture, and net sensitivity calculations described in this paper have been implemented as a fully automated tool and demonstrated on the circuits described in Section V. The data capture is done via HSPICE [14] and the SVM models are created using LIB-SVM [15]. Each circuit has an associated set of design specifications which indicate the input parameters correlated with the desired operational behavior. In each experiment, the transient voltage behavior is modeled. The model accuracy is determined by the predicted values compared to the actual simulated values.

Each netlist is modified to include the duration of time for the transient simulation and voltage capture for each net. In parallel with simulation, the hierarchy of the defined sub-circuits is determined and partition graphs are created. The data file generated from the simulation run is parsed and a subset of data is used to quickly determine if the sub-circuits require further partitioning starting from the topmost level of the hierarchy until the desired accuracy is achieved. Once the level is determined then the models are created using

full simulation data. The models are then chained together to predict the final circuit outputs. Those models within the same level of hierarchy can be determined in parallel.

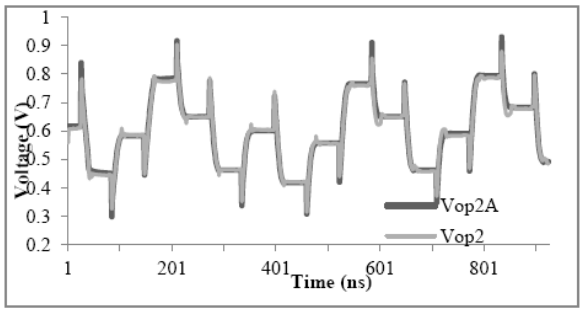

Fig. 8. Waveform comparision between simulation waveform, *Vop*2A, and predicted waveform, *Vop*2.

The models for the ∑Δ-ADC were trained with 50,000 vectors, one input period and tested with 50,000 different vectors and each model maintains 99% prediction accuracy. The accuracy is slightly lower, 98% when the models are combined to form the full system. The final differential output waveform is predicted at 98.82% accuracy. The total amount of time to perform 500,000 predictions, 50,000 per model, is 184 seconds. The speedup over HSPICE is only 21_x because of the high number of features in the amplifier models, with each model having 15 dimensions. Fig. 8 shows a subset of dependent predictions for the Amp_V_{op2} model. V_{op2A} is the simulated waveform and V_{op2} is the predicted waveform.

TABLE I: BEHAVIOR MODELS FOR THE UWB-PLL

Model Name	*# Support Vectors*		*Training Time (S)*		*Prediction Accuracy (%)*	
	nom	para	nom	para	nom	para
PFD_UP	115	88	.21	.266	99.64	98.98
PFD_UPi	111	94	.28	.192	99.63	99.01
PFD_DN	106	94	.18	.186	99.19	98.78
PFD_DNi	107	90	.24	.185	99.11	98.78
CP	489	518	2.65	4.41	95.31	96.8
DIVN_S1	180	180	.18	.18	98.55	98.66
DIVN_S2	162	162	.17	.162	99.3	99.23
DIVN_S3	1176	886	.81	.637	98.86	99.33
DIVN_S4	1154	366	.78	.294	99.42	99.63
DIVN_S5	490	230	.36	.203	99.11	99.47
DIVN_S6	245	123	.21	.137	99.12	99.48
DIVN_S7	131	71	.14	.103	99.32	99.45
REG_VUWB	1003	1018	6.99	6.7	94.95	98.7
VCO_NET41	639	1027	2.7	6.81	96.17	98.66
VCO_Vn	1511	1519	3.4	4.33	96.68	96.2
VCO_Vp	1512	1523	3.3	4.32	96.68	96.2
VCO_Von	1512	1511	3.3	4.1	97.56	97.12
VCO_Vop	1509	1516	3.4	4.2	97.58	97.24
VCO_fout	**1008**	**1012**	**2.5**	**.81**	**97.1**	**95.1**

Table I shows the models created for the PLL with the amount of training time, number of support vectors, and the prediction accuracy with dependent testing. The columns are broken into nominal circuit (nom) and including extracted layout parasitics (para). Parasitics do not affect the CCG partitioning therefore the sub-graphs are the same. The models are trained with 10,000 vectors and tested with 20,000 different vectors. The training time is significantly shorter for the PLL due to its continuous nature and smaller dimensionality. Each partition in the PLL contains fewer features making the intermediate and input-output relationships less complex. The total prediction time for 380,000 vectors is 15.76 seconds which is a 445X speedup over HSPICE simulation for the nominal case. For the case with parasitics, to perform all 380,000 predictions, 20,000 test vectors per model, the total prediction time is 14.54 seconds, a 1766X speedup over HSPICE. The simulation time in HSPICE for this netlist with 60830 components took 7 hours and 8 minutes. Fig. 9 shows the simulated F_{out} and predicted F_{out_P} waveforms without and with parasitics.

Our models can be used for multiple purposes. Performance characteristics can be extracted directly from the waveform, i.e. lock time for the PLL. SVM models are language independent and can be translated directly into System Verilog modules. The waveforms generated from the models allow for testing of analog and mixed signal components within much larger and complex systems.

It is well known that power supply noise may cause jitter in the PLL system. With the tight interaction between analog and digital circuits in SoCs, it is difficult or impossible to create separate power and ground planes for the analog components. In order to demonstrate the robustness of our models, we apply random power supply fluctuations to the system that are within 10% of nominal V_{DD}. The waveforms generated from the models determine the amount of jitter associated with changes in V_{DD}.

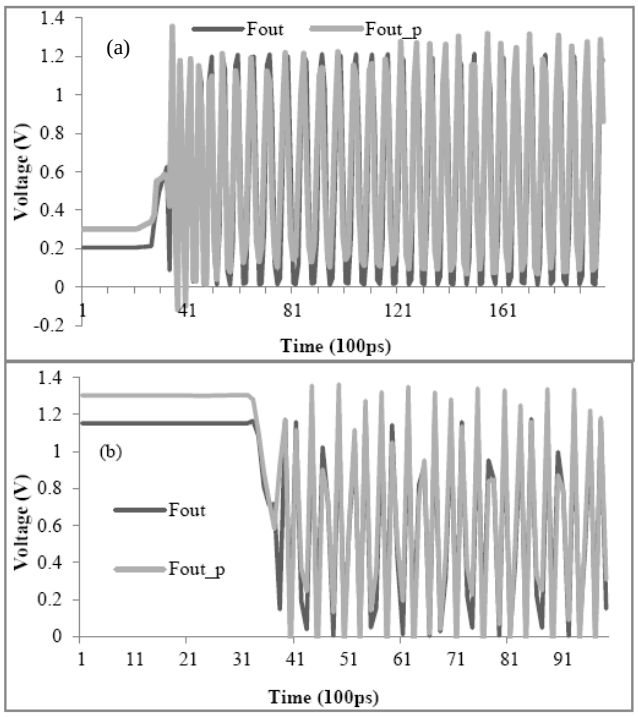

Fig. 9. Simulated waveform, fout; and predicted waveform, fout_*P* for the PLL-UWB (a) nominal circuit, (b) with parasitics.

The models from the above experiment are enhanced to include simulation data of the PLL at ten different V_{DD} levels and are created and compiled in the same fashion. The V_{DD} perturbations occur for a fraction of each clock period. From the transient waveform we extract the jitter associated with changing V_{DD}. Fig. 10 shows the frequency with jitter produced by the PLL under various changes in the power supply. The nominal operating conditions are at 1.2V and 2.5GHz. The graph compares the PLL frequency

from the simulated waveforms and predicted from the SVM models.

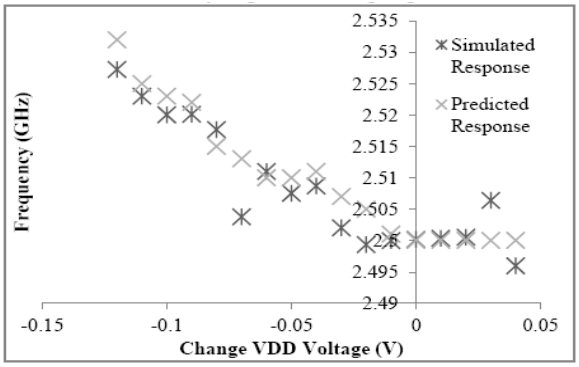

Fig. 10. Frequency of the PLL during the periods of V_{DD} variation.

VII. Conclusion

We have presented a new automated approach for creating behavioral models of analog and mixed signal circuits based on partitioning. The methodology addresses the need for system level behavioral modeling of any type of analog and mixed signal circuits without the need for in-depth designer knowledge and utilizes full system simulation data which incorporates block level interactions into the models. The SVM models used to predict the intermediate behaviors and final output behavior are language independent and are implemented in System Verilog using the real number system. We have demonstrated our method on large low-power circuits such as a PLL and ∑Δ-ADC. Our results indicate that we can obtain three orders of magnitude speedup over transistor level simulations while maintaining over 95% accuracy. We were able to illustrate the usefulness of the approach as it pertains to full chip simulations by showing the accuracy of the models are maintained when under the effects of power supply fluctuations induced by high switching digital circuits.

References

[1] Solutions for Mixed-Signal Soc Verification. [Online]. Available: http://www.cadence.com/rl/Resources/white_papers/ms_soc_verification_wp.pdf

[2] N. Z. Hakim, A. Bhaduri, K. Donepudi, and S. Bodapati, "A hybrid electrical-behavioral modeling approach for pre-and post-silicon electrical validation," *CICC, 2012 IEEE*, pp.1-5, Sept. 2012.

[3] R. A. Rutenbar *et al.*, "Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 640-669, 2007.

[4] D. De Jonghe *et al.*, "Extracting analytical nonlinear models from analog circuits by recursive vector fitting of transfer function trajectories," *DATE*, 2013, pp.1448-1453.

[5] Y. Cao *et al.*, "Dynamic behavioral modeling of nonlinear microwave devices using real-time recurrent neural network," *IEEE Trans. El. Dev.*, vol. 56, no. 5, pp.1020-1026, 2009

[6] D. Drmanac *et al.*, "A non-parametric approach to behavioral device modeling," *ISQED*, pp. 284-290, 2010.

[7] M. Elzeftawi, "Compact low-power low-noise neural recording wireless channel for high density neural implants (HDNIs)," Ph.D. thesis, Dept. ECE, UCSB, 2012.

[8] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA USA, 2001.

[9] H. Li *et al*., "Analog behavioral modeling flow using statistical learning method," *ISQED*, pp. 872-878, 2010.

[10] D. Overhauser *et al.*, "Automatic mixed-mode timing simulation," in *Proc. ICCAD*, pp. 84, no. 87, 1989.

[11] L. Yang and C. J. R. Shi, "FROSTY: a fast hierarchy extractor for industrial CMOS circuits," *ICCAD*, pp. 741- 746, 2003.

[12] S. Basu *et al*., "Variation-aware macro modeling and synthesis of analog circuits using spline center and range method and dynamically reduced design space," in *Proc. Int. Conf. on VLSI Design*, 2009, pp. 433-438.

[13] Y. Wei and A. Doboli, "Structural macro modeling of analog circuits through model decoupling and transformation," *IEEE Trans. on CAD*, vol. 27, no. 4, pp. 712-725, 2008

[14] HSPICE User Guid. [Online]. Available: http:// www.synopsys.com

[15] C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. in Int. Syst. and Tech.*, vol. 2, no. 3, 2011.

Samantha Alt received her B.S. and M.S. degrees in electrical and computer engineering from the University of California Santa Barbara, in 2007 and 2011, respectively.

During her study at University of California Santa Barbara, she hold internships at Mentor Graphics and Intel, where she worked on transition fault testing, and virtual design under test generation, RTL test generation using SMT solvers, test point insertion, and mixed signal simulation analysis. Her research interests include machine learning applications in design and verification of mixed-signal circuits, diagnosis, and test.

Malgorzata Marek-Sadowska received her M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from Politechnika Warszawska (Technical university of Warsaw), Poland.

From 1976 to 1982 she was an assistant professor at the institute of electron technology at the Technical University of Warsaw. She became a research engineer at the U.C. Berkley Electronics Research Laboratory in 1982 and continued there until 1990, when she joined the Department of Electrical and Computer Engineering at the University of California, Santa Barbara as a professor.

Marek-Sadowskawas an associate editor from 1991 to 1993, and the editor-in-chief from 1993 to 1995 of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. She is an IEEE Fellow.

Li. C. Wang received his B.S. degree in computer engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1986, and the M.S. degree in computer science and the Ph.D. degree in electrical and computer engineering from the university of Texas, Austin, in 1991 and 1996 respectively.

From 1996 to 2000, he was with the IBM/Motorola Somerset PowerPC Design Center, Austin, TX, as a senior software engineer. Since 2001, he has been with the Department of Electrical and Computer Engineering, University of California Santa Barbara as a Faculty Member His current research interests include microprocessor tests and verification, performance validation, and applications of data learning in design and testing.

Wang has twice received the Best Paper Award from the Design Automation and Test in Europe Conference, in 1998 and 2003, and also from the IEEE Very Large Scale Integration Test Symposium in 1999. He co-founded the IEEE Microprocessor Test and Verification Workshop and currently serves as a Program Committee Member for the International Symposium on Quality of Electronic Design, Design, Automation, and Test in Europe Conference, the International Test Conference, the International Workshop on High-Level Design Validation and Test, and the international Test Syntheses Workshop.