ICCAD: G: ABCD: Accurate Booleanization of Continuous Dynamical Systems for Analog/Mixed-signal Design

Aadithya V. Karthik

PhD student at The University of Califonia, Berkeley, ACM student member 6727949
Email: aadithya@berkeley.edu, ***Address:*** 1322 Shattuck Ave Apt 205, Berkeley, CA 94709
PhD advisor: Jaijeet Roychowdhury ⟨jr@berkeley.edu⟩

We present ABCD[1], a suite of tools for modelling continuous dynamical systems (*e.g.*, SPICE-level circuit netlists) using *purely Boolean models* (*e.g.*, FSMs, BDDs, or AIGs), in an accurate and scalable manner, for high-speed simulation, test-pattern generation, and formal analysis/verification of analog/mixed-signal designs.

I. PROBLEM AND MOTIVATION

In today's advanced process technologies (32nm and below), analog/mixed-signal (AMS) components (such as A/D and D/A data converters, I/O and equalization circuitry, phase-locked and delay-locked loops, *etc.*) have become key performance bottlenecks. They not only limit the performance of the overall design, but also account for a significant proportion of overall design bugs and debugging costs.

For example, Fig. 1 depicts a successive approximation D/A converter, or a SAR-ADC. As the figure shows, this circuit has several AMS components (such as a track and hold unit, a D/A converter, and an analog comparator). In addition, the circuit also includes a fair amount of digital logic (*e.g.*, the registers and combinational circuitry). In cutting-edge designs, the AMS components play a significant role in limiting performance, and they can also introduce a variety of bugs and subtle design flaws that compromise functionality. Therefore, to effectively design the overall system, it is necessary to develop the capability to model, simulate, and analyze the underlying AMS components at or near SPICE-level accuracy. Furthermore, to guarantee performance and correctness of operation of the overall design, it is important to develop formal analysis/verification capabilities involving AMS systems. That is, given a set of specifications (*e.g.*, bounds on settling time, power consumption, eye diagram parameters, *etc.*), we need a method to *formally prove* that the design meets the specifications; if not, the method should produce a valid counter-example illustrating *why* the design fails.

The challenge is that we have one set of (continuous) models for the analog components (*e.g.*, SPICE models, systems of differential/algebraic equations, *etc.*), and another set of (discrete/Boolean) models for the digital components (*e.g.*, truth tables, BDDs, AIGs, *etc.*). These models do not mix well from the standpoint of verification, and the most powerful and capable verification tools that we have today (*e.g.*, ABC [1]) can only work with Boolean models, not continuous ones.

The goal of ABCD, therefore, is to *approximate* the AMS components in the design using *purely Boolean models* (Fig. 1, right), so as to capture the analog dynamics of the underlying components in an accurate and scalable manner. This results in an "all boolean" system, which opens up possibilities for high-speed simulation, test-pattern generation, and formal analysis/verification of designs involving AMS components.

II. BACKGROUND AND RELATED WORK

The formal analysis and verification of AMS designs is a long-standing problem that has received considerable attention from the design automation community. A number of techniques have been proposed for this problem; together, these fall under the umbrella of "hybrid systems verification". Such techniques offer the capability to represent and reason about continuous (analog) quantities, where continuous variables are used to model analog signals in the design, and Boolean variables are used for digital signals. Some prominent methods based on this approach include the linear hybrid automata used by HyTech [2] and Phaver [3], the polyhedral invariant hybrid automata employed by tools such as Checkmate [4], the labelled hybrid petri-nets verified using difference bound matrices as part of the LEMA toolkit developed by Myers et. al. [5], *etc.*

Unfortunately, however, the techniques above suffer from *severe scalability limitations*, because the formal analysis of continuous quantities is an inherently difficult problem. Consequently, such methods are unable to handle systems with more than a few (*e.g.*, 5-10) continuous signals. Moreover, due to these scalability constraints, these approaches are often forced to resort to *a priori modelling simplifications* that sacrifice SPICE-level accuracy; without this accuracy, the "guarantees" obtained by verification can be untrustworthy and dangerous to rely on.

In contrast, we have proposed a new family of techniques [6]–[9] that model AMS components without employing *any* continuous variables, *i.e.*, all continuous signals in the system are *discretized* and represented as Boolean variables (*i.e.*, bit vectors). This transformation preserves accuracy and simultaneously enables us to leverage cutting-edge Boolean analysis techniques (*e.g.*, ABC [1]) to carry out model checking/formal verification of AMS designs entirely in the Boolean domain, which is very scalable and efficient in practice.

III. APPROACH AND UNIQUENESS

As mentioned above, the unique feature of our approach, ABCD, is that we "Booleanize" all analog components/signals in the given design, and we do so without sacrificing either accuracy or scalability.

ABCD takes as input an analog/mixed-signal system, specified, for example, as a SPICE model or a set of differential equations. Given this input, ABCD produces as output a purely Boolean circuit representation (consisting only of Boolean constructs such as registers, counters, combinational logic blocks, *etc.*) that accurately captures the end-to-end I/O behaviour of the given system, in a completely scalable fashion.

This approach carries with it many advantages relative to existing techniques:

1. **No *a priori* modelling simplifications:** ABCD-L accepts arbitrary SPICE models/differential equations as input, without requiring crude *a priori* over-simplifications. Therefore, in principle, any effect that SPICE can predict can be modelled by ABCD in Boolean form. This enables reliable and trustworthy formal analysis/verification of AMS systems.
2. **High accuracy that is configurable:** Even though ABCD-L produces models that are purely Boolean, these models can capture the dynamics of analog systems to high accuracy. This accuracy is determined by the number of bits used to discretize the underlying analog signals, which is a configurable parameter that can be passed to ABCD at runtime.
3. **High-speed simulation:** Because ABCD models are purely Boolean, they can be simulated very efficiently, entirely in the logical domain, using constant-time memory lookup operations (as opposed to having to solve complicated differential equations). This translates to orders of magnitude speedup over SPICE, while simultaneously retaining much of SPICE's accuracy.
4. **Scalable verification:** ABCD produces models that can be directly imported into state-of-the-art model checking/verification tools like

[1] **A**ccurate **B**ooleanization of **C**ontinuous **D**ynamics

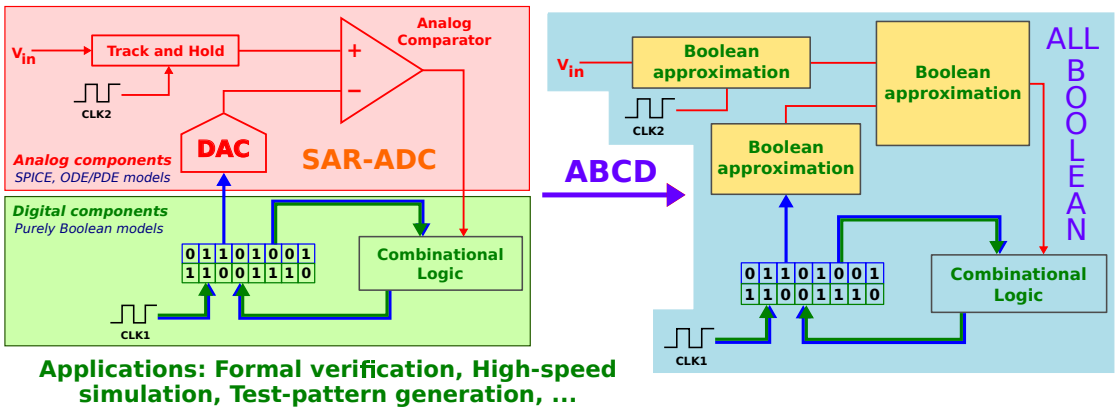

Fig. 1. A successive approximation A/D converter (SAR-ADC) modelled using ABCD.

ABC [1]. Since ABCD models do not employ any continuous quantities, they can leverage the power of such cutting-edge Boolean analysis tools to enable scalable formal analysis of AMS systems. Furthermore, ABCD models can also be used in conjunction with existing hybrid systems methodologies, where they can model AMS components using "cheap" Boolean variables, *i.e.*, without incurring the penalty of using up expensive continuous variables.

5. **Extensive applications:** The above characteristics make possible a diverse range of applications for ABCD in circuit design, including high-speed simulation, test-pattern generation, formal analysis/verification involving AMS components, *etc.* In addition, ABCD opens up interesting possibilities for analysing and verifying systems that arise in other domains, *e.g.*, systems of chemical reactions, neural/biological systems, *etc.*

IV. RESULTS AND CONTRIBUTIONS

We have demonstrated the accuracy and scalability of ABCD using a variety of linear and non-linear circuit examples (for example, see our publications describing ABCD-L [8] and ABCD-NL [9]) such as equalizers, I/O links, filters, charge pumps, D/A and A/D converters, comparators, *etc.* Due to space constraints, we are unable to present all these examples here; instead, we confine our discussion to just a few examples that are of foremost relevance to AMS designers, including (1) a charge pump/filter system that is relevant to PLL design, (2) a digital to analog (D/A) converter or DAC, embedded within a SAR-ADC as shown in Fig. 1, and, (3) a linear dispersive channel followed by a continuous time differential equalizer. In each case, we show that the Boolean model produced by ABCD-NL is able to accurately reproduce the SPICE-level analog dynamics of the underlying circuit, including important performance-limiting non-ideal phenomena. Finally, we also show an example where we use ABCD's Boolean model in conjunction with a state-of-the-art Boolean analysis engine (ABC) to formally verify an AMS design.

A. Charge pump driving an analog filter

Fig. 2 shows a charge pump driving an analog filter. The system works as follows: the transistors M1 and M2 form a current mirror that can pump a current I_0 into the load capacitor C_L, whereas transistors M3 and M4 form an opposing current mirror that can withdraw current I_0 from C_L. The circuit has two inputs, V_{up} and V_{down}. During charging/discharging, exactly one of these inputs is high; if V_{up} (V_{down}) is high, current is pumped in (out), driving the output voltage V_{out} higher (lower); this is called the charging (discharging) mode of the charge pump, and the system responds most quickly when in one of these modes (*e.g.*, using a 90nm process, response times are typically of the order of tens of nanoseconds).

If both inputs are high, the charge pump enters an imbalance-driven mode, which can either charge or discharge the load, depending on transistor sizing and operating conditions. This is typically much slower (*e.g.*, hundreds of nanoseconds response time). Finally, if both inputs are low (cutoff mode), the output voltage is chiefly determined by leakage currents – the slowest mode of operation of the system, with response time in the microsecond range.

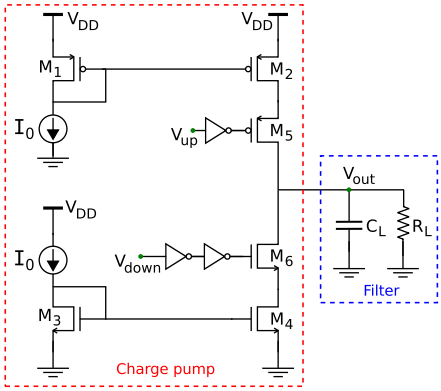

Fig. 2. Schematic of a charge pump driving an analog filter.

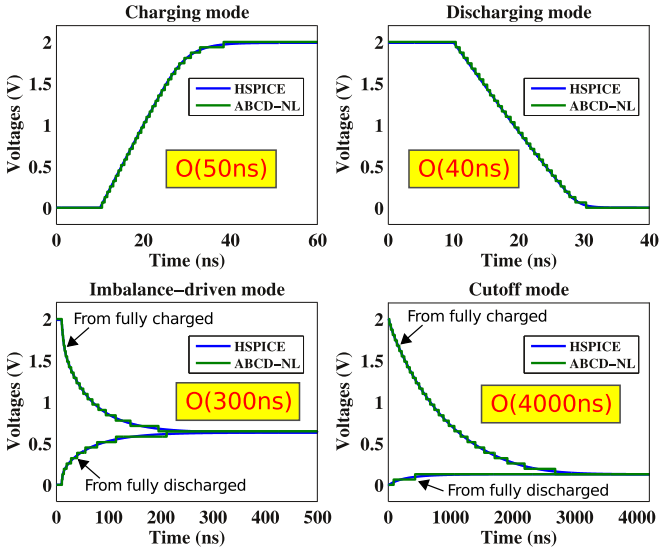

Fig. 3. ABCD-NL accurately captures the behaviour of the charge pump under all four modes of operation, in spite of the widely differing time scales involved.

Our goal is to use ABCD-NL to accurately model the behaviour of the above system under all possible operating conditions. So we designed the above circuit in 90nm CMOS, using BSIM4 device models. We then applied ABCD to Booleanize this system (using 5 bits to encode the output waveform), and we simulated the resulting Boolean model on a range of inputs that covered all four modes of operation. In each case, we compared the output predicted by ABCD-NL's Boolean model, against that predicted by HSPICE. Fig. 3 shows the results, where HSPICE waveforms are shown in blue, and ABCD-NL waveforms are in green. As the figure shows, in spite of the widely differing time scales involved in the four modes, the Boolean model produced by ABCD-NL closely matches the SPICE-level

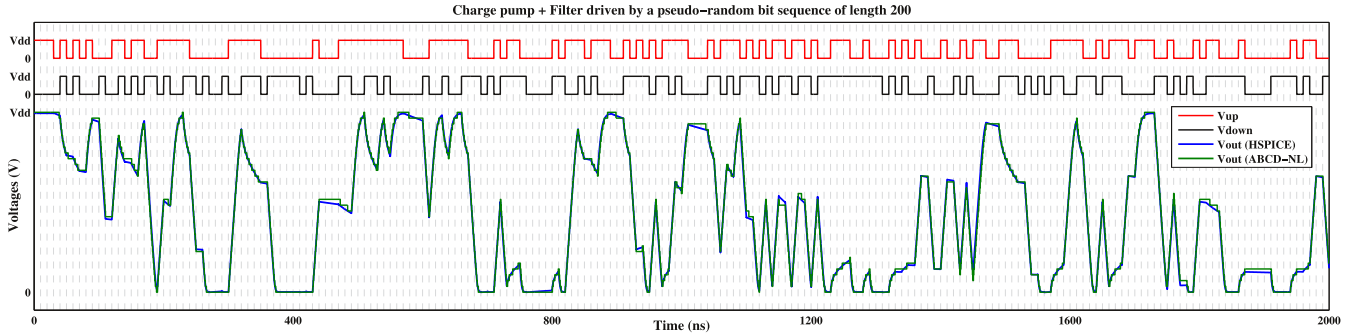

Fig. 4. ABCD-NL can predict the response of the charge pump/filter system, almost to SPICE-level accuracy, even over long time frames that involve rapid switching between all 4 modes of operation.

dynamics of the system in all its modes.

Fig. 4 further demonstrates the accuracy and robustness of the Boolean model produced by ABCD-NL. The figure shows a long pseudo-random bit sequence applied as input to the circuit, which switches the circuit in and out of all 4 modes of operation over a long time frame. Throughout this time, it is seen that the Boolean model produced by ABCD-NL (the green waveform) closely tracks the SPICE-simulated output (the blue waveform) of the system. This indicates that ABCD-NL is indeed a powerful and accurate modelling technique, and one that can conceivably be used as a much faster, almost-as-accurate, drop-in replacement for SPICE over long transient runs.

B. A D/A converter (or DAC) within a SAR-ADC

For this experiment, we designed a D/A converter using off-the-shelf components developed by Analog Devices, Inc. As Fig. 5 shows, we used four AD8079A buffers (SPICE models available from [10]), and an R/2R ladder that feeds a voltage follower, to design this (4-bit) DAC.

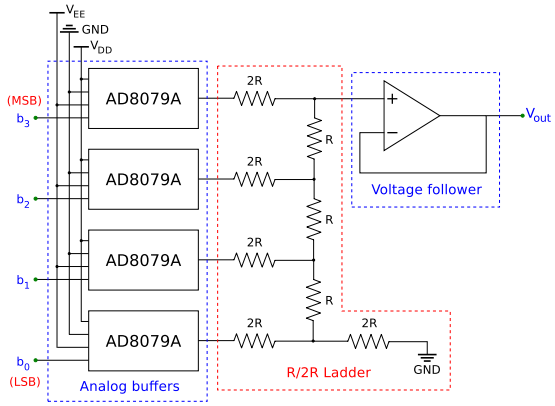

Fig. 5. Schematic of a D/A converter used within a SAR-ADC.

We then produced a purely Boolean model for the D/A converter using ABCD, and compared its predictions against SPICE. For example, a key figure of merit of a D/A converter is speed; so it is important to accurately capture the delay of the system for all possible bit transitions at the input. Fig. 6 shows many of these transitions (due to space constraints, we are unable to show all the transitions), and it can be seen from the figure that ABCD-NL accurately captures the system's delay for all these inputs (using 6 bits to encode the D/A output).

Furthermore, because this DAC is embedded within a SAR-ADC, it is important to have our Boolean model reproduce the system's dynamics for input patterns that are typical to the SAR-ADC environment. Indeed, Fig. 7 shows that ABCD's Boolean model is able to accurately reproduce the D/A converter's dynamics when it is embedded within a SAR-ADC, as shown in Fig. 1. The red waveform in the above figure shows a 150kHz sine wave, which is the ADC input. The ADC operates at about 8MHz, so each period of the input generates about 52 ADC samples. Over these samples, the input bits b_0 to b_3 of the D/A converter switch as shown in the top half of Fig. 7. The blue waveform at the bottom of the figure depicts the

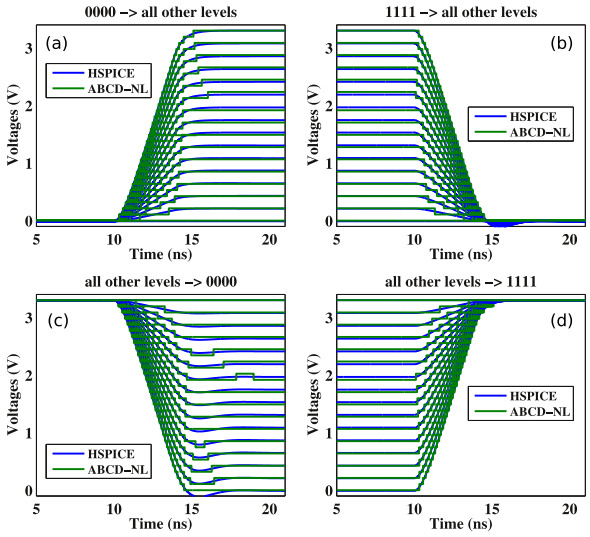

Fig. 6. ABCD-NL closely matches SPICE-level simulation of the D/A converter, for several input bit transitions.

D/A converter's output, as predicted by HSPICE. And the green waveform is the prediction made by ABCD's Boolean model (after mapping the bits back into the analog domain). Clearly, ABCD is able to reproduce the system's response accurately, showing that ABCD can be a powerful way to analyse mixed-signal designs. Moreover, we have also imported the resulting Boolean model into the formal verification engine ABC [1], thereby illustrating ABCD's suitability for model checking/verification as well. Indeed, as illustrated in [9], we have been able to use ABCD to formally verify the throughput of a non-linear AMS signaling system – modelled in SPICE using 22nm BSIM4 transistors, Booleanized with high accuracy using ABCD, and property-checked using ABC.

C. Linear channel + continuous time differential equalizer

We now apply ABCD to an LTI channel followed by a (linearized) equalization circuit, as illustrated in Fig. 8. We note that, in this circuit, both the channel (modelled as an RC chain) and the equalizer use *differential signalling*, *i.e.*, the circuit's inputs and outputs are represented by the *difference* between two voltages (instead of a single voltage)[2]. The equalizer plays a critical role in this circuit: it *partially reverses* the distortion (ISI) produced by the channel, so that one can transmit bits across the channel at much higher speeds than would be possible otherwise. For example, if the channel's cut-off frequency is 1 GHz, then reliable transmission can happen only at bitrates at or below 1Gbps. However, if the combined "channel plus equalizer" system has an effective cut-off frequency at 3 GHz, then one can triple the throughput without suffering distortion.

Fig. 9 illustrates the application of 6-bit ABCD-L to the small-signal linearized "channel plus equalizer" circuit (henceforth simply referred to as

[2]Differential signalling has important advantages over single-ended signalling, including better noise resilience, improved resistance to external interference, *etc.*

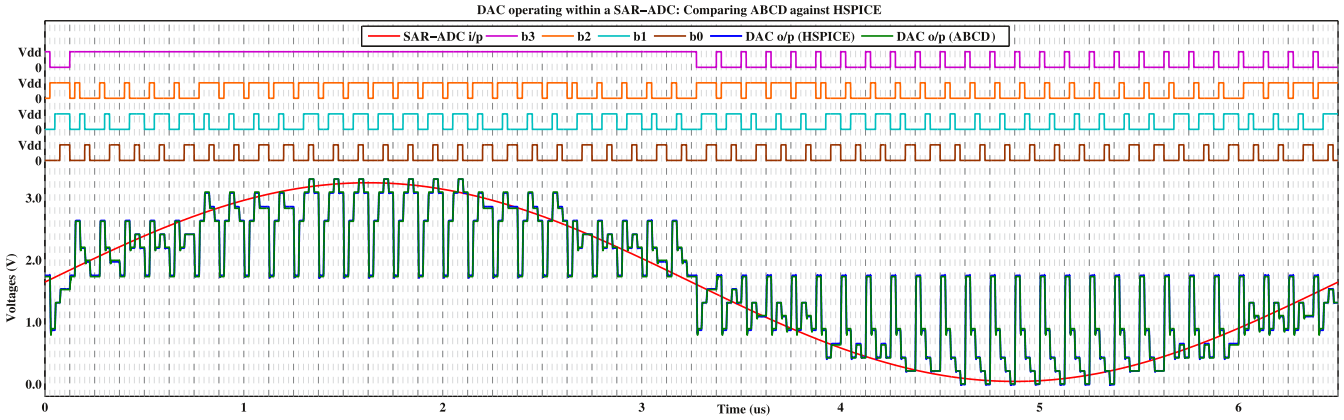

Fig. 7. ABCD-NL, using a purely Boolean model, is able to accurately capture the dynamics of a D/A converter embedded within a SAR-ADC, across a long time frame encompassing several ADC samples.

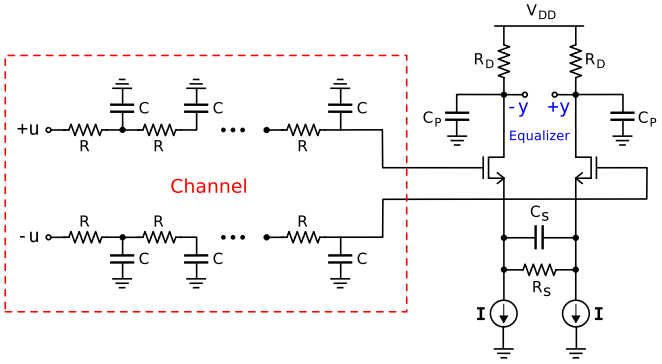

Fig. 8. LTI channel followed by a differential equalizer.

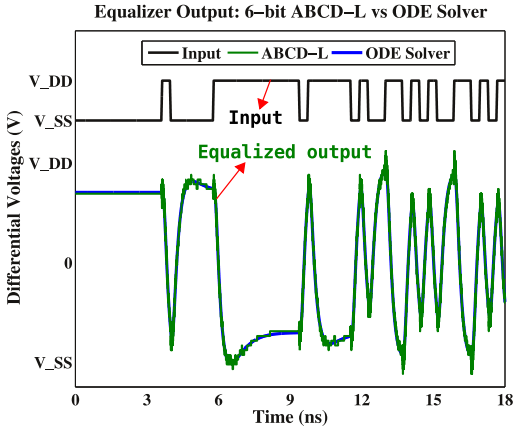

Fig. 9. ABCD-L accurately reproduces the time-domain continuous dynamics of the equalizer.

"the system") above (with the channel being a 5-unit RC chain). The blue waveform of Fig. 9 was obtained by using an ODE solver to simulate the system, for a random bit pattern applied at the input (the black waveform). As before, we see from the figure that the Boolean model produced by ABCD-L is able to accurately duplicate the time-domain behaviour of the system.

For equalizers, an important AMS-relevant design metric is the eye diagram correction produced by the circuit. In typical AMS applications, the eye diagram at the input to the equalizer (*i.e.*, the channel output) has a very small or even non-existent eye opening (Fig. 10(a)). The equalizer offsets some of the ISI produced by the channel, which can considerably widen the eye opening; for example, Fig. 10(b) shows the eye diagram produced by a small-signal SPICE simulation of the above system, using SpiceOPUS [11]. Parts (c) and (d) of Fig. 10 depict the eye diagrams produced by applying 6-bit and 8-bit ABCD-L to the above "channel plus equalizer" system, overlaid on top of the (blue) SPICE eye diagram. As the figures show, the eye diagrams obtained from ABCD's Boolean models are able to accurately reproduce the eye diagram correction achieved by the equalizer. This demonstrates that ABCD is indeed a viable technique to Booleanize the continuous dynamics of AMS systems.

D. Formally verifying an ABCD-NL model

As we remarked before, the main focus of this paper is the *accurate modelling of AMS components for verification*, as opposed to the verification itself. However, for completeness, we now present an example where we Booleanize an AMS system using ABCD-NL, import the resulting Boolean model into a verification engine (ABC, [1]), and carry out formal property checking of the model against an AMS-design relevant specification.

Fig. 11. Schematic of a system that was formally verified using a combination of ABCD-NL and ABC. The inverters were designed in a 22nm CMOS process, using BSIM4 models. The channel was modelled as a long RC chain.

Fig. 11 depicts the system that we formally verified. It follows the same pattern as the systems shown in Figs. **??** and **??**. As we mentioned earlier, such systems play an important role in SI applications, where it is important to determine, and *formally verify*, the throughput of the system. We have used 22nm BSIM4 models for each transistor in the system, and an analog channel that consists of several RC units chained together.

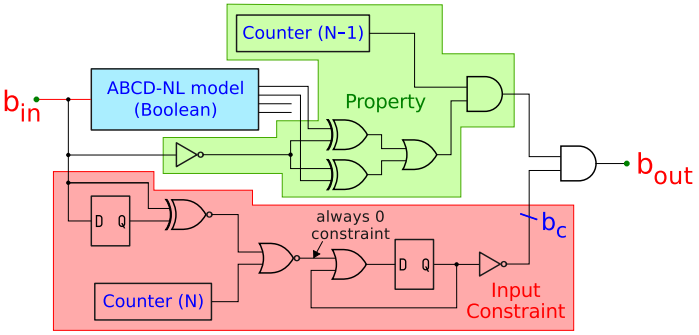

Fig. 12. Encoding the throughput property, along with constraints on the input, in a Boolean form so as to formally verify the ABCD-NL model using ABC [1].

Fig. 12 shows the verification flow that we used. As the figure shows, the Boolean circuit that is verified consists of three parts, (1) the ABCD-NL Boolean model, (2) the Boolean logic that encodes the property to be checked, and (3) some Boolean logic to encode constraints on the inputs that can be applied to the AMS system. The circuit of Fig. 12 is constructed in such a way that the given AMS design fails to meet its throughput specification if and only if the bit b_{out} can somehow be asserted to 1 by choosing an appropriate sequence of bits applied at b_{in}.

The constraint on the input is that it can change only once per N clock periods of the ABCD-NL model. The time period of ABCD-NL's sequential Boolean model is 10ps. The input constraint is modelled using a counter that outputs a 1 every N clock cycles (Fig. 12). If this constraint is violated, the bit marked b_c immediately becomes 0 and stays there forever, which

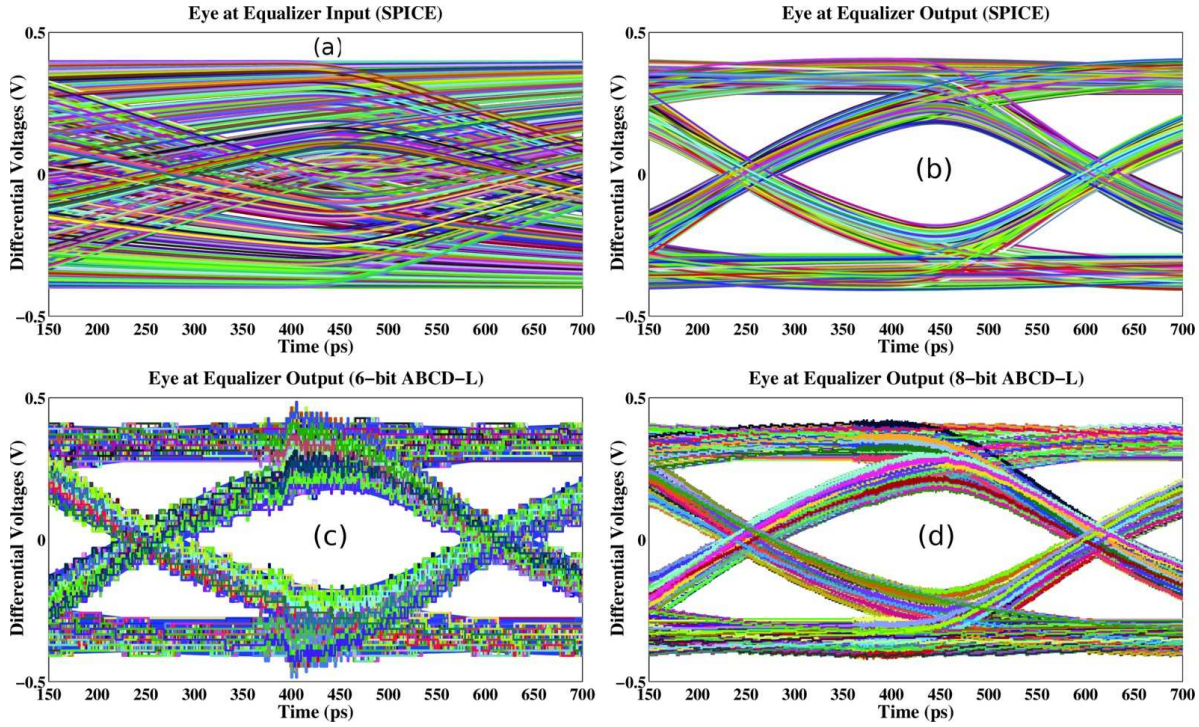

Fig. 10. ABCD accurately reproduces the entire shape of the eye diagram at the equalizer's output.

makes it impossible to assert b_{out} to 1. This ensures that any counter-example returned by ABC would satisfy the input constraint.

The throughput property to be checked is that, given the above constraint on the input, the output should always reach an acceptable state before N clock cycles (*i.e.*, before the input can change). This acceptable state is defined as being ≥ 0.8V for a 1, and being ≤ 0.2V for a 0.

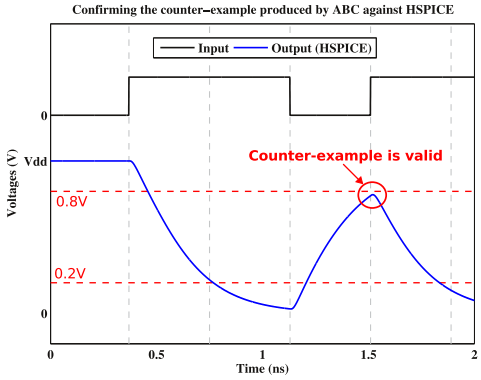

Fig. 13. Checking that the counter-example returned by ABCD-NL + ABC is valid in the analog domain.

Clearly, there is an N_0 such that the above throughput property will fail for all $N \leq N_0$. We can use ABC to quickly zero in on N_0, by incorporating ABC-based verification within a binary search loop. In this way, we were able to determine that $N_0 = 38$. This translates to a throughput of approximately 2.56Gbps. Furthermore, we were also able to confirm, using HSPICE, that for $N = 38$, the counter-example returned by ABC is a valid one in the analog domain (Fig. 13). Therefore, the throughput bound obtained is tight and meaningful. Thus, ABCD-NL is a powerful and capable modelling technique that can be used for AMS verification.

V. SUMMARY AND CONCLUSIONS

We have presented ABCD, a suite of techniques and algorithms for deriving purely Boolean approximations of continuous dynamical systems in an accurate and scalable manner. We have shown how ABCD can be used to approximate a number of non-linear analog/mixed-signal (AMS) systems (including equalizers, charge pumps, A/D and D/A converters, *etc.*) to near-SPICE accuracy, without making any *apriori* simplifications. The Boolean models produced by ABCD-NL can be used to solve a number of challenging problems in cutting-edge AMS design, including high-speed simulation, test generation, and formal verification of AMS designs, by leveraging existing tools developed for Boolean/hybrid systems analysis (*e.g.*, ABC [1]). Also, we have demonstrated how ABCD can be used in a formal verification flow for a high-speed signaling system – modelled in SPICE using 22nm BSIM4 transistors, Booleanized with high accuracy using ABCD, and property-checked using ABC.

REFERENCES

[1] R. K. Brayton and A. Mishchenko. ABC: An academic industrial-strength verification tool. In *CAV '10: Proceedings of the 22nd International Conference on Computer Aided Verification*, pages 24–40, 2010.

[2] T. A. Henzinger, P. H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1):110–122, 1997.

[3] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. *International Journal on Software Tools for Technology Transfer*, 10(3):263–279, 2008.

[4] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 2003.

[5] S. Little. *Efficient Modeling and Verification of Analog/Mixed-Signal Circuits Using Labeled Hybrid Petri Nets*. PhD thesis, University of Utah, 2008.

[6] C. Gu and J. Roychowdhury. FSM model abstraction for analog/mixed-signal circuits by learning from I/O trajectories. In *ASP-DAC '11: Proceedings of the 16th Asia and South Pacific Design Automation Conference*, pages 7–12, 2011.

[7] K. V. Aadithya and J. Roychowdhury. DAE2FSM: Automatic generation of accurate discrete-time logical abstractions for continuous-time circuit dynamics. In *DAC '12: Proceedings of the 49th Design Automation Conference*, pages 311–316, 2012.

[8] A. V. Karthik and J. Roychowdhury. ABCD-L: Approximating continuous linear systems using Boolean models. In *DAC '13: Proceedings of the 50th Design Automation Conference*, pages 63:1–63:9, 2013. **(Best paper nomination)**.

[9] A. V. Karthik, S. Ray, P. Nuzzo, A. Mishchenko, R. K. Brayton, and J. Roychowdhury. ABCD-NL: Approximating continuous non-linear dynamical systems using purely Boolean models for analog/mixed-signal verification. In *ASPDAC '14: Proceedings of the 19th Asia and South Pacific Design Automation Conference*, 2014.

[10] http://www.analog.com/Analog_Root/static/techSupport/designTools/spiceModels/license/spice_general.html?cir=AD8079A.cir.

[11] http://www.spiceopus.si/.