ISCLEs: Importance Sampled Circuit Learning Ensembles for Trustworthy Analog Circuit Topology Synthesis

Peng Gao[1], Trent McConaghy[1,2], and Georges Gielen[1]

[1] ESAT-MICAS, K.U. Leuven, Leuven, Belgium
[2] Solido Design Automation Inc., Saskatoon, Canada
{Peng.Gao,Trent.McConaghy,Georges.Gielen}@esat.kuleuven.be

Abstract. Importance Sampled Circuit Learning Ensembles (ISCLEs) is a novel analog circuit topology synthesis method that returns designer-trustworthy circuits yet can apply to a broad range of circuit design problems including novel functionality. ISCLEs uses the machine learning technique of boosting, which does importance sampling of "weak learners" to create an overall circuit ensemble. In ISCLEs, the weak learners are circuit topologies with near-minimal transistor sizes. In each boosting round, first a new weak learner topology and sizings are found via genetic programming-based "MOJITO" multi-topology optimization, then it is combined with previous learners into an ensemble, and finally the weak-learning target is updated. Results are shown for the trustworthy synthesis of a sinusoidal function generator, and a 3-bit A/D converter.

1 Introduction

The design / choice of a cell-level analog circuit topology can have a giant impact on the performance of a system. Currently, industrial topology design is done almost exclusively by hand. A longtime goal has been to automate the design or choice of topology, and there has been significant progress towards the goal via the fields of evolvable hardware (EH) and analog computer-aided design (CAD), but it has not been fully realized because either the synthesized topology has not been sufficently trustworthy (McConaghy and Gielen, 2005), or the approach does not allow novel functionality and topologies.

Table 1 shows synthesis approaches, by capability. Earlier EH research (row 1) focused on fully open-ended structural synthesis[1,2], but the CPU effort was prohibitive[3] and the results were not only not trustworthy[4], they often

[1] "Novel functionality" here means that the approach can be set to a new problem just by changing testbenches, which allows for new types of analog circuit functionality.
[2] "Novel structures" means that the approach may invent new structures.
[3] "Reasonable CPU effort" is for the context of industrial use by a tool user (semiconductor company).
[4] "Trustworthy" means that the results are either designer-trusted by construction, or the new structural novelty is easily identifiable by a designer.

G.S. Hornby et al. (Eds.): ICES 2008, LNCS 5216, pp. 11–21, 2008.

Table 1. Topology Synthesis Approaches

Approach	Novel functionality?	Novel structures?	Trustworthy?	Topology variety?	Reasonable CPU effort?
Open-ended (Koza et al, 2003, Shibata et al, 2002, ...)	yes	yes	no	yes	no
Open-ended + domain-specific constraints (Sripramong et al, 2002, Dastidar et al, 2005, Mattiuissi et al, 2007)	yes	yes	no	yes	borderline
Flat pre-specified blocks (Kruiskamp et al, 1995, Maulik et al, 1995)	no	no	yes	no	yes
Hier. pre-specified blocks: MOJITO (McConaghy et al, 2007a)	no	no	yes	yes	yes
Hier. pre-specified blocks + novelty: MOJITO-N (McConaghy et al, 2007b)	no	yes	yes	yes	yes
Boosting tiny pre-specified blocks: ISCLEs (this work)	**yes**	**yes**	**yes**	**yes**	**yes**

looked strange. More recent efforts (row 2) added domain knowledge to improve efficiency and trustworthiness, but there is still no guarantee of trustworthy results or trackable novelty. Early CAD research (row 3) focused on searching through sets of known topologies, which gave both speed and trustworthy results; unfortunately the number of possible topologies was extremely limited[5] and there was no clear way to generalize the approaches to more problem types. More recent research has attempted to merge ideas from both fields: MOJITO searches through combinations of hierarchically-organized designer-specified analog building blocks, thus giving a large set of topologies that can be readily applied to common analog design problems. MOJITO-N allows for more open-ended structural novelty, but tracking the novelty explicitly and only rewarding novel individuals that actually improve performance. But both MOJITO and MOJITO-N are constrained to problems that analog designers have attacked, they do not address problems with novel functionality.

The goal of this paper is in the final row: to determine topologies that can be novel in both functionality and topology, yet trustworthy, in reasonable CPU effort. This paper shows an approach that hits the goal on two circuit design problems, with promise for a broader set of applications.

How might we hit the aims of novel structural synthesis? Moore's Law (Moore, 1965, ITRS, 2008) presents us with a possible opportunity: IC transistor geometries have exponentially shrunk so much that each individual minimally-sized transistor has become virtually free. This means that in design, as predicted

[5] "Topology variety" means that the set of possible topologies is sufficiently rich that it contains appropriate solution(s) to the target functionality, including problem variants with different objectives and constraint settings.

by decades ago, we can *waste transistors* (Mead and Conway, 1980). However, this only holds if the transistors are near-minimal for the process. Digital circuits obey this, but not currently analog: designers have kept analog circuits larger as a key way to reduce the effects of process variation-induced mismatch (Sansen, 2006).

This paper shows how digitally-sized transistors can be used for analog design, by stacking together dozens or hundreds of minimally-sized topologies using the machine learning framework of boosting. ISCLEs returns novel topologies that are trustworthy by construction and robust to mismatch.

The rest of this paper is organized as follows. Section 2, discusses machine learning and ISCLEs. Section 3 describes weak learner topologies that we designed. Section 4 has experimental results on the synthesis of a sinusoidal function generator, and a 3-bit A/D converter. Section 5 concludes.

2 Machine Learning and ISCLEs

This section starts with a discussion on machine learning, and how its evolution as a field can be emulated in circuit design (1). Then, ISCLEs is detailed.

Two major sub-problems in machine learning (Hastie, 2001) are regression and classification; the key challenge for each is to find an input-output mapping that predicts well on unseen data. For decades, the prevailing approach was to come up with some single well-performing model, which almost always had the issue of overfitting, in which the model performed well on training data but generalized poorly to unseen data. However, a new approach has emerged in the last decade: ensembles of models (Polikar, 2006), which combine the output of many learners. It inherently overfits less because the errors made by sub-learners can be averaged out (assuming the sub-learners' outputs are not correlated). In "bagging", each sub-learner learns the full input-output mapping. Alternatively, a series of "weak learners" can be "boosted" into a "strong learner" that captures the overall mapping (Freund and Schapire, 1997). Weak

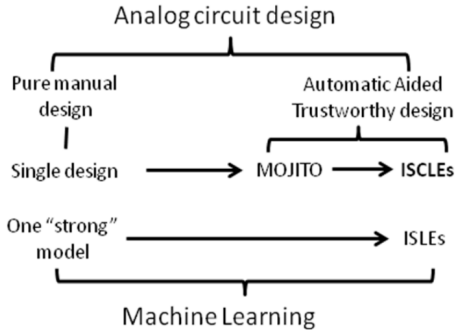

Fig. 1. ISCLEs shows how analog circuit design can shift from one "strong" model to ensembles of "weak" models, just as machine learning has

learning is much easier to do than strong learning of one model: each learner only needs to do better than random, rather than fully capture the mapping. The outer boosting algorithm takes care of combining the many weak learners together in order to get the target mapping. Boosting does importance sampling in model space, hence the label Importance Sampled Learning Ensembles (ISLEs)(Friedman and Popescu, 2003).

In analog circuit design and in analog synthesis, all existing approaches do the equivalent of designing a single "strong" circuit realizing the target functionality. In contrast, ISCLEs boosts many "weak" circuits. Crucially, these weak circuits each have small area (via near minimally-sized transistors) so that overall area is not prohibitive. The overall architecture is trustworthy: it merely does weighted addition of the weak learners' outputs plus an offset voltage. Each weak learner's topology is also trustworthy, as all the possible weak-learner topologies are in the set of hierarchically-organized designer-specified building blocks. At each ISCLEs boosting iteration, a weak learner is built, then added with a weighting factor α in parallel with the other weak learners, and the difference between the current ensemble's output waveform(s) and overall target output waveform(s) will be calculated. These waveforms can be from dc sweeps, ac sweeps, transient simulation, etc. This difference is used as the target for the next weak learner. The process repeats until the difference is sufficiently low to stop, at which point the overall ensemble circuit is returned. The whole process is automatic.

The sole boosting parameter was α (learning rate), which we set to 0.10, meaning that on each iteration, 10% of the newest weak learner's output is used to update the overall target waveform. This setting strikes a compromise between risk of overfitting (higher α), and slower convergence (lower α).

Each weak learner is found with MOJITO (McConaghy et al, 2007a) searching the possible sized topologies of section 3. MOJITO views these possible sized topologies as a parameterized grammar, then searches them with grammatical genetic programming (Koza, 1992, Whigham, 1995). MOJITO's objective is to maximize the correlation between the current target waveform(s) (as specified by the boosting loop) and its candidate circuit's waveform(s). MOJITO's constraints are to keep each transistor width and length in the range from minimal size to 10x minimal size according to the process node (thereby keeping area in check), and to meet device operating constraints (e.g. "keep transistor in saturation"). By optimizing on correlation rather than squared error, MOJITO's problem is easier because correlation ignores the difference in offset between waveforms; the outer boosting loop takes care of this with its offset voltage.

MOJITO was configured to maximize search efficiency yet avoid getting stuck, using the following setup. At a given weak learner target, the population size was set to 10, and 50 generations were run. If the resulting circuit reduced the ensemble's overall error, then that weak learner was considered complete, and added to the ensemble. But if overall error did not improve, then the population size was doubled and MOJITO was re-run. In practice, we found that no doubling occurred in early iterations, but a few rounds of doubling occurred in later iterations. All other MOJITO settings were the same as (McConaghy et al, 2007a).

3 Weak Learner Topologies

A central challenge in this work was to design a competent library of possible weak learners. Some applications may only need a simple inverter, and others may need more complex topologies. We designed three weak learners: an inverter, an inverter with I-V amplifier, and an inverter cross-coupled differential pair. Together, these form the library of possible topologies that MOJITO searches through. We now describe each.

1. **Inverter Weak Learner.** This is the simplest weak learner. Its top-level block (figure 2) can turn into one of four possible sub-blocks (figure 3), and has 4 sizing parameters.

2. **Inverter With I-V Amplifier.** The core idea of this weak learner leverages the fact that current flow in an inverter is not a monotonic function of the input voltage. While the input sweeps from 0 to Vdd, the current will increase because the NMOS is gradually turned on, but after a certain threshold point, the PMOS becomes off and current will reduce to 0 again. The position of that peak is determined by the sizing of the two transistors. If the aspect ratio of the NMOS is increased, the peak position will be lower, and vice versa. This means that the position and the width of the peak are controllable. Then we use a current to voltage conversion, an I-V amplifier, to convert this current peak information into a voltage peak. So by proper sizing the I-V amplifier, we can make controllable voltage peak waveforms,

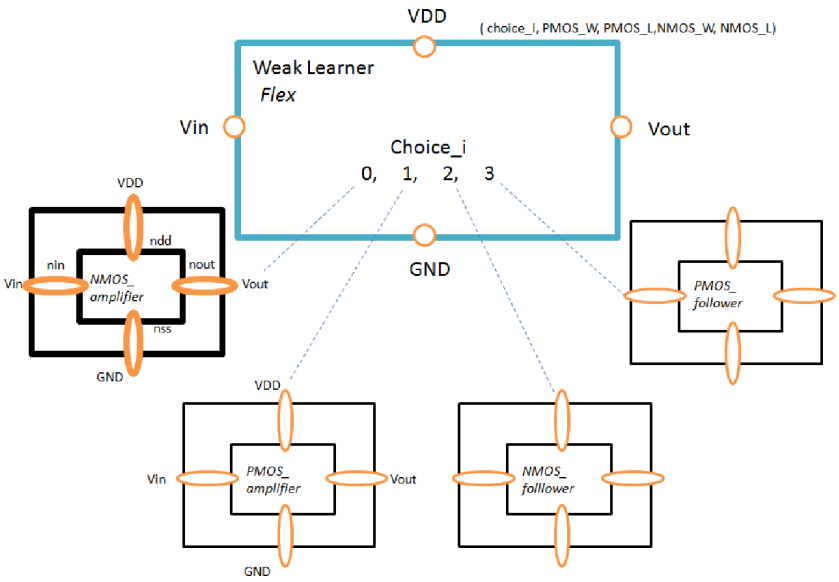

Fig. 2. Schematic for inverter weak learner. It can expand into one of four possible topologies.

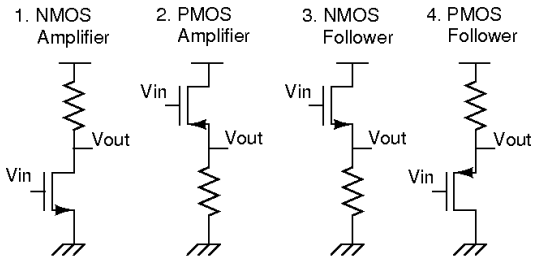

Fig. 3. The four possible sub-blocks for the inverter: NMOS single-stage amplifier, NMOS source follower, PMOS single-stage amplifier, or PMOS source follower

where we control both the position and the width of the peak. Of course the width cannot be too small, due to the finite gain and sensitivity of the I-V amplifier. A peak simulation result is shown in figure 4, which shows how different waveforms between any the transition point and the higher transition point are realizable by different transistor sizes.

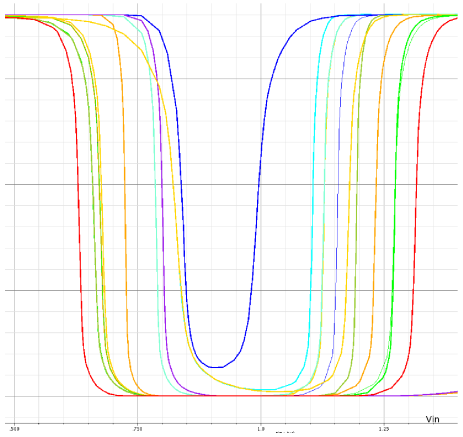

Fig. 4. Negative voltage peaks (Generated by inverter with I-V amplifier with parameter sweep simulations). The x-axis is V_{in}, and y-axis is V_{out}.

3. **Inverter Cross-Coupled Differential Pair.** This weak learner circuit (figure 5) is composed of a cross coupled differential pair and several current mirrors. The input signal is connected to one of the input pins of each differential pair. The other input pins are connected to different bias voltages V_{b1} and V_{b2}. These two bias voltages will build two fixed threshold points (Johns and Martin, 1997); together with varying the size of input transistor pairs, the threshold point will be controlled to go up or down. Therefore the output transfer curve will be similar to figure 4.

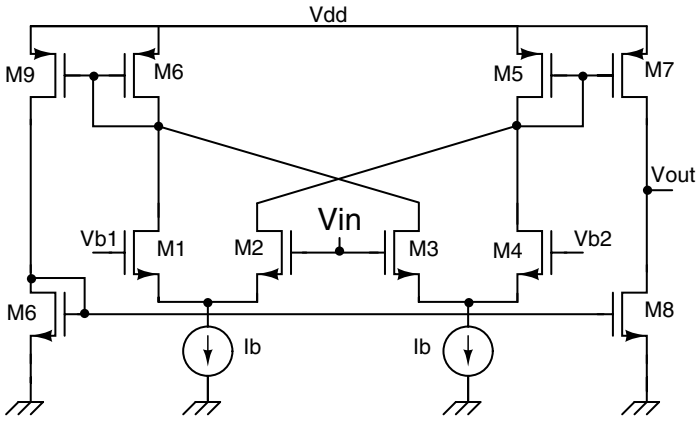

Fig. 5. Cross-Coupled Differential Amplifier

4 Experimental Results

We applied ISCLEs to three problems: a DC-DC sinusoidal function generator, a 3-bit A/D converter, and a 3-bit A/D converter with process variation. The circuit simulator was Hspice, using a 0.18μm CMOS process technology; other settings were provided in section 2. Runtime for each was under 8 hours on a Linux machine with a single-core 2.0 GHz Intel processor.

4.1 Sinusoidal Function Generator

In this example, ISCLEs is applied to generate a DC-in DC-out sinusoidal function generator. Specifically, the aim is to minimize the squared error difference between target DC response and synthesized circuit's DC response, for several different input DC values.

Figure 6 shows the result of 40 boosting iterations, resulting in an ensemble of 40 weak learners. Sub-figures 6(a) to (e) show ensemble's output response waveform (lower waveform / green) and the target waveform (upper waveform / red), where input voltage is the x-axis and output voltage is the y-axis. We see that as the boosting iterations progress, the ensemble's output response waveform converges to match the target waveform. Sub-figure 6(f) shows the evaluation parameter NMSE (normalized mean squared error) vs. boosting iteration. We see that after 40 iterations, only a few percent error is remains between the target and ensemble circuit's responses. This example demonstrates that the core idea of ISCLEs – applying boosting to structural circuit synthesis – is sound.

4.2 3-Bit "Flash" A/D Converter

The aim of this example is to target a more widespread type of circuit functionality, namely A/D conversion. Here, we specifically aim for a 3-bit "Flash" A/D

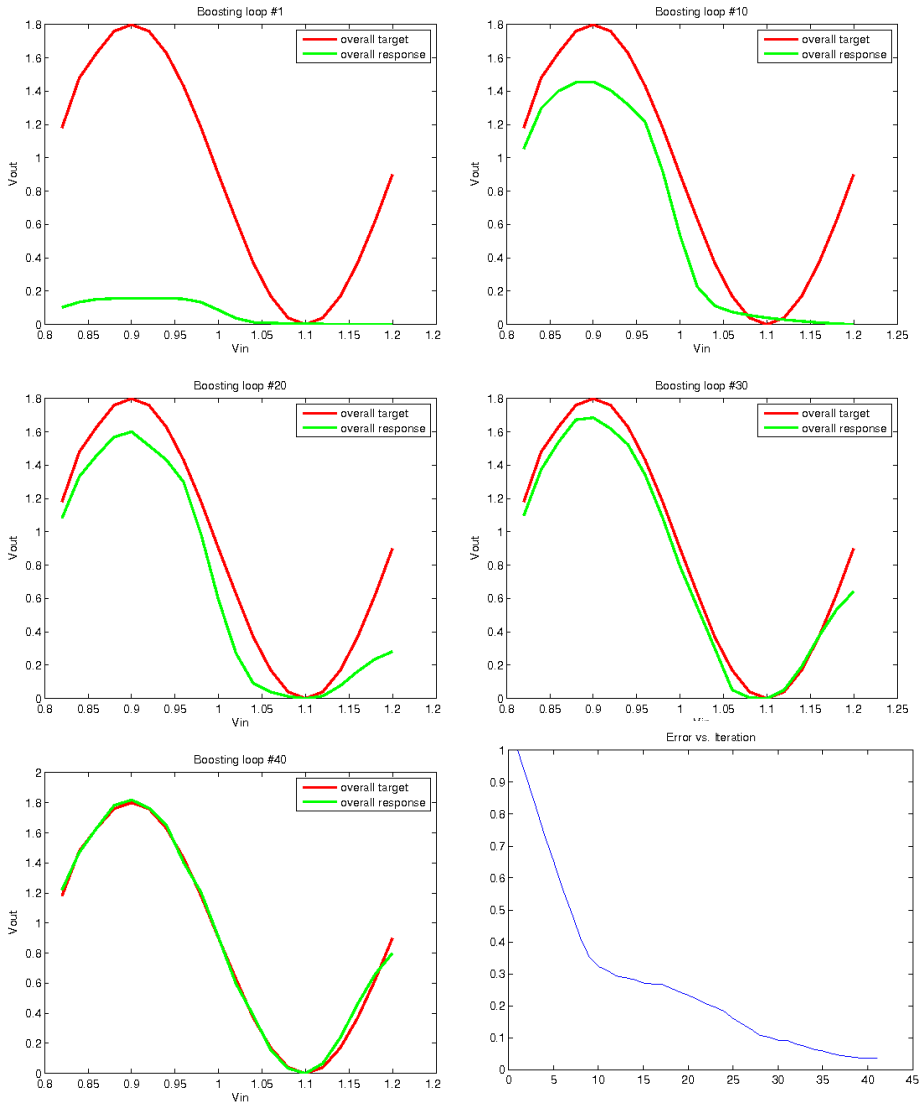

Fig. 6. ISCLEs Sinusoidal Waveform Learning Response

architecture. "Flash" type A/Ds are quite sensitive to process variations, due to the matching property of the resistor ladder and comparator (Sansen, 2006). We approach this problem by synthesizing for one bit at a time. For each bit, the aim is to minimize the squared error difference between target DC response and synthesized circuit's DC response, for several different input DC values.

Figure 7 shows results. The top row of sub-figures is LSB, middle row is 2nd LSB, and bottom row is MSB. For each row (bit), the left figure shows the output vs. input DC voltage, for both target and synthesized output response; and the

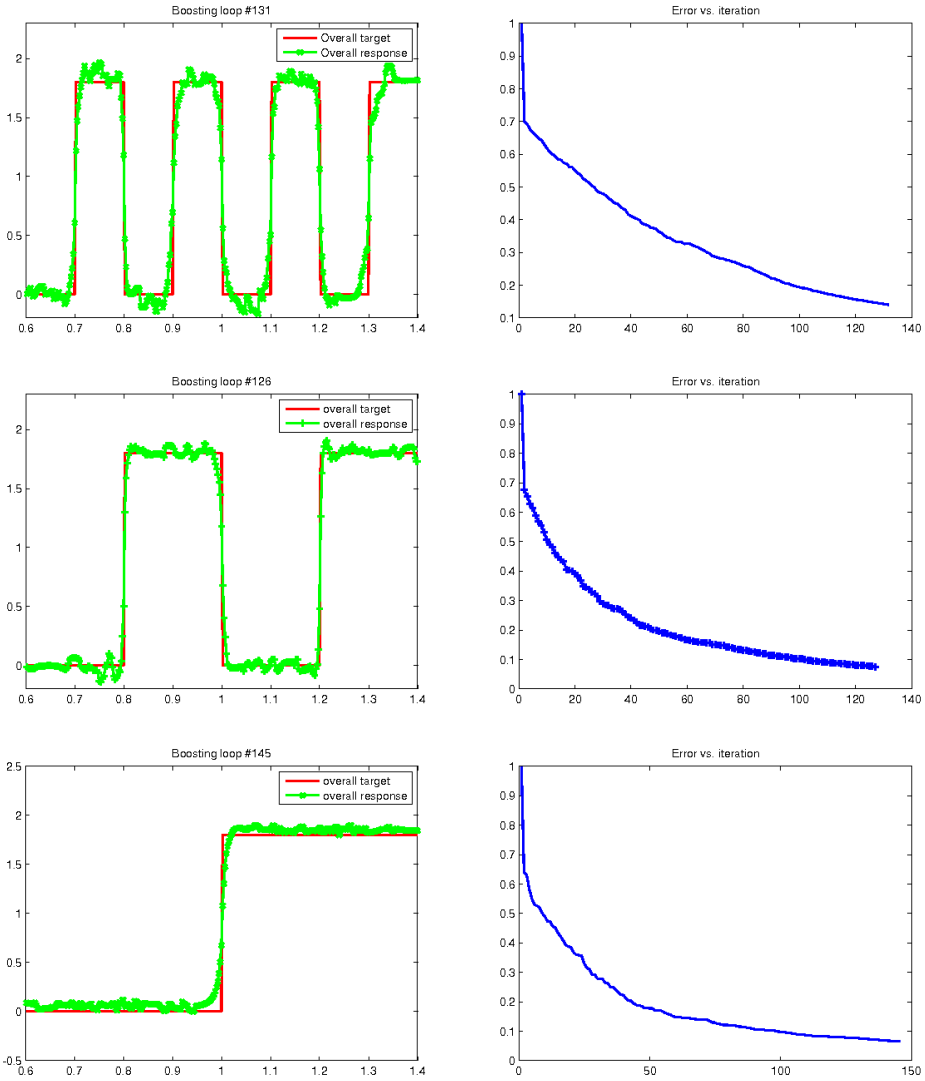

Fig. 7. ISCLEs 3-bit A/D converter output

right figure shows the convergence of NMSE vs. boosting iteration. The LSB has the most complex input/output mapping, but ISCLEs still achieved 13% NMSE, having 131 weak learners. To our knowledge, no prior ("strong learner") synthesis approaches have ever successfully synthesized a DC-DC mapping as complex as this. The 2nd LSB reached 9% NMSE with 126 weak learners. The MSB also reached 9% NMSE with 145 weak learners. Note that for actual implementation, the bits' outputs are usually passed through an inverter that would rail the outputs to the high or low voltage value (i.e. Vdd and ground), thus making the DC-DC mapping tighter yet.

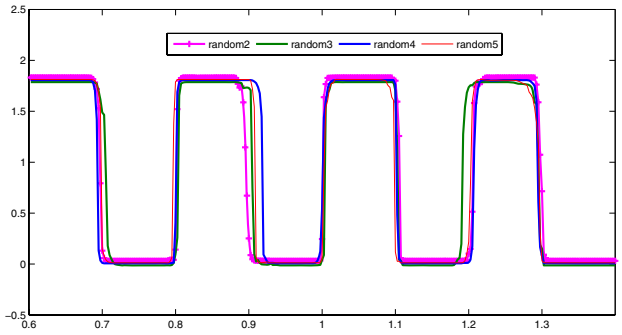

Fig. 8. 3-Bit ADC's LSB with smaller process variation injection

4.3 A/D Converter Simulation with Process Variation

Recall that the key issue of using (near) minimally-sized transistors for analog circuits was sensitivity to process variation. So, we investigate its effect here, with the hypothesis that the importance-sampling nature of ISCLEs might have some natural resilience to process variation. So, in order to test its tolerance to process variation, we inject some variations in the transistor model parameter V_{th} (threshold voltage) into the already-synthesized A/D circuits, and measure the response.

Figure 8 shows, for four Monte Carlo samples, the A/D's LSB (railed) simulations with $A_{vth} = 6\text{mV}\mu\text{m}$. The overall response changes only slightly from sample to sample; that is, our ISCLEs-synthesized circuits have graceful tolerance to process variations. We acknowledge that it is likely safer to account for variation more directly by incorporating process variations into the boosting loop iself; we leave that to future work.

5 Conclusion

This paper presented ISCLEs, a method that synthesizes circuit topologies which are novel in both functionality and topology, yet trustworthy, within reasonable CPU effort (on at least the two applications shown). ISCLEs extends the machine learning method of boosting to circuit design: boosting's "weak learners" are designer-trusted topologies that are sized and chosen by a genetic programming-based approach (MOJITO), and the overall boosting ensemble ties together all the weak learner circuits with a weighted adder circuit. We designed a library of trusted weak learner topology choices for MOJITO to search. ISCLEs' effectiveness was demonstrated on two problems: a sinusoidal function generator, and 3-bit A/D converter learning. By demonstrating promising resilience to process variations yet using minimally-sized devices, ISCLEs has promise as a way for analog circuits to scale with process technology.

References

McConaghy, T., Gielen, G.: Genetic Programming in Industrial Analog CAD: Applications and Challenges. GP Theory and Practice III, pp. 291–306. Springer, Heidelberg (2005)

McConaghy, T., Palmers, P., Gielen, G., Steyaert, M.: Simultaneous multi-topology multi-objective sizing across thousands of analog circuit topologies. In: Proc. DAC, pp. 944–947 (2007)

McConaghy, T., Palmers, P., Gielen, G., Steyaert, M.: Genetic programming with design reuse for industrially scalable, novel circuit design. GP Theory and Practice V, pp. 159–184. Springer, Heidelberg (2007)

Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)

Whigham, P.A.: Grammatically-based Genetic Programming. In: Proc. Workshop on GP: from Theory to Real-World Applications (1995)

Moore, G.E.: Cramming more components onto integrated circuits. Electronics Mag. 38(8) April 19 (1965)

ITRS: International technology roadmap for semiconductors (last accessed March, 2008), `http://public.itrs.net`

Mead, C., Conway, L.: Introduction to VLSI Systems. Addison-Wesley, Reading (1980)

Sansen, W.: Analog Design Essentials. Springer, Heidelberg (2006)

Gielen, G., et al.: Analog and digital circuit design in 65 nm cmos: End of the road? In: Proc. DATE, pp. 36–42 (2005)

Johns, D., Martin, K.: Analog Integrated Circuit Design. Wiley, Chichester (1997)

Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journ. Computer and System Sci. 55(1), 119–139 (1997)

Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer, Heidelberg (2007)

Polikar, R.: Ensemble Based Systems in Decision Making. IEEE CAS Mag. (2006, 3rd quarter)

Friedman, J.H., Popescu, B.E.: Importance sampled learning ensembles. Technical Report, Department of Statistics, Stanford University (2003)

Koza, J.R., et al.: Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer, Dordrecht (2003)

Lohn, J.D., Colombano, S.P.: Automated Analog Circuit Synthesis using a Linear Representation. In: Sipper, M., Mange, D., Pérez-Uribe, A. (eds.) ICES 1998. LNCS, vol. 1478, pp. 125–133. Springer, Heidelberg (1998)

Sripramong, T., Toumazou, C.: The Invention of CMOS Amplifiers Using Genetic Programming and Current-Flow Analysis. IEEE Trans. CAD 21(11), 1237–1252 (2002)

Dastidar, T.R., Chakrabarti, P.P., Ray, P.: A Synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse. IEEE Trans. EC 9(2), 211–224 (2005)

Mattiussi, C., Floreano, D.: Analog Genetic Encoding for the Evolution of Circuits and Networks. IEEE Trans. EC 11(5), 596–607 (2007)

Kruiskamp, W., Leenaerts, D.: DARWIN: CMOS Opamp Synthesis by Means of a Genetic Algorithm. In: Proc. DAC, pp. 433–438 (1995)

Maulik, P., Carley, L., Rutenbar, R.A.: Integer Programming Based Topology Selection of Cell Level Analog Circuits. IEEE Trans. CAD 14(4), 401–412 (1995)