Analog Implementation of Ontogenic Neural Networks for RF Built-In Self-Test

Dzmitry Maliuk* and Yiorgos Makris†
*Electrical Engineering Department, Yale University, New Haven, CT, 06520-8267
†Electrical Engineering Department, The University of Texas at Dallas, Richardson, TX, 75080-3021

***Abstract*— We introduce an analog implementation of an ontogenic neural network (ONN) model and investigate its applicability to the field of built-in self-test (BIST) of RF circuits. Our chip consists of a reconfigurable array of synapses and neurons operating below threshold and featuring sub-μW power consumption. The synapse circuits employ dynamic weight storage for fast bidirectional weight updates during training. The learned weights are copied onto the analog floating gate (FG) memory for permanent storage. Training is performed using a chip-in-the-loop strategy by the cascade-correlation learning algorithm. A benchmark XOR task is employed to evaluate the system performance. Finally, the network is trained to distinguish faulty from functional LNA chips using a set of low-cost sensor measurements.**

I. Introduction

Machine learning-based testing of RF circuits has suggested the use of non-linear classifiers to predict pass/fail test labels from low-cost measurements. If a circuit-under-test (CUT) is integrated on the same die alongside stimuli generators, on-chip sensors and a non-linear classifier, then a complete stand-alone BIST architecture can be achieved [1]. The readings of on-chip sensors, usually provided in the form of DC voltages, are presented to the neural classifier which produces a binary output indicating whether the CUT passes or fails its specifications. The accuracy of such prediction depends both on the ability of the neural network to learn the underlying mapping (its learning ability) and on the separability of classes as such (sensor measurement quality). The latter has been addressed in recent studies showing great promise in predicting performances of CUTs with minimum overhead (e.g. [2]). Instead, we focus on the hardware implementation of the non-linear classifier — an ontogenic neural network. The software version of the ONN has proven to be an adequate learning model to predict circuit health from low-cost measurements [3]. However, the processing resources required to run the software model, such as an external computer or a built-in DSP, are not always available to a stand-alone IC, thus calling for a custom hardware network that can be integrated on-chip.

In applications such as BIST the additional power and area overhead incurred by test circuits is crucial. We selected an analog implementation for its superior power efficiency during run time, compact size and the possibility of permanent weight storage using analog floating gate memory. Indeed, when high precision is not required, analog computation can be as much as 1000x more energy efficient than digital [4]. In addition, analog circuits can be directly interfaced with sensor outputs, thereby eliminating the use of analog-to-digital converters. Finally, the use of the FG technology in standard CMOS offers efficient non-volatile storage of weight values with high accuracy [5]. Following the mentioned guidelines we have designed and fabricated a reconfigurable experimentation platform in a 0.35-μm CMOS process available from TSMC. The remainder of the paper is structured as follows. Section II introduces the ONN learning model. Section III describes the implementation details including the overall architecture, the weight storage mechanism, the synapse and neuron circuits. Section IV presents the measurement data and evaluates the system's learning ability on XOR and LNA classification tasks.

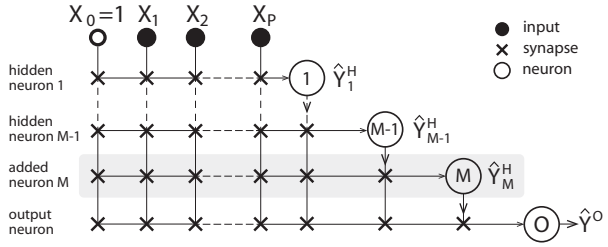

Fig. 1. The ontogenic neural network topology. The hidden neurons receive connections from both the network inputs X_i and the outputs of the previous hidden neurons $\widehat{Y}_j^H$. The bottom neuron serves as a network output $\widehat{Y}^O$.

II. Overview of Ontogenic Neural Networks

Fig. 1 illustrates a block diagram of the ONN learning model. A decision boundary is constructed by successively adding hidden neurons; each hidden neuron augments the feature space of the original inputs with the intention of making the derived space linearly separable. This strategy is guided by the cascade-correlation algorithm [6], which repeats the following steps for each added neuron. Suppose that our current stage has $M-1$ hidden neurons, as shown in Fig. 1. Let $\widehat{Y}_i^H$ be the output of the i-th hidden neuron and $\widehat{Y}_i^O$ be the output of the network when it has i hidden neurons. The M-th hidden neuron is added at the bottom so that it sees the primary inputs $X_0, ..., X_P$ as well as all the outputs of the preceding neurons $\widehat{Y}_1^H, ..., \widehat{Y}_{M-1}^H$. Next, we train this neuron to maximize the correlation between its output $\widehat{Y}_M^H$ and the training error of the previous stage $\widehat{E}_{M-1} = \sum (Y_T - \widehat{Y}_{M-1}^O)^2$, where Y_T represents the target class labels and the summation is done over the entire training set. Once the correlation is maximized, the weights of this neuron become permanent and the output layer is retrained to minimize the error on the training set, i.e. $\widehat{E}_M = \sum (Y_T - \widehat{Y}_M^O)^2$. Note that in each step, only the weights of the neuron being added or of the output neuron undergo modification, while the other weights are kept unchanged. This feature greatly simplifies the gradient estimation by the hardware and leads to stable performance even for large-sized topologies. Hidden neurons

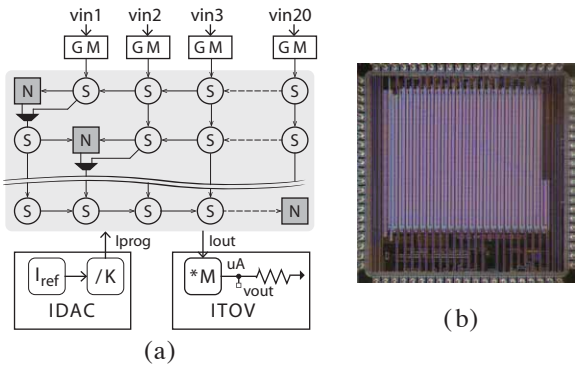

Fig. 2. (a) System architecture of the ontogenic neural network chip. The reconfigurable array of synapses (S) and neurons (N) is shown in the shaded box. (b) Die photograph of ONN chip measuring 3×3 mm^2.

are added until a stopping criterion is reached, which in our case is a classification error on a validation set. The correlation maximization and the error minimization are done by the resilient back propagation algorithm (iRPROP+), which can be efficiently customized for hardware networks.

III. Chip Design

A. System Description

Fig. 2 shows a block diagram of the neural network chip. A 30×20 array of synapses and neurons is arranged so that the neurons are aligned along the main diagonal of the upper matrix and along the right edge for the bottom part. The reconfigurable fabric for the ONN topologies is a subset of synapses lying below the main diagonal (similar to the layout in Fig. 1). Global connectivity is programmable by means of multiplexors inserted between rows. The core operates in the analog domain with weights and signals represented by differential currents. A single weight value requires two current sources for differential current storage. A current source is implemented as a current storage cell (CSC) circuit that combines two modes of operation: the dynamic, for fast bidirectional weight updates, and the non-volatile, for long-term storage of learned weights. The dynamic mode is engaged during training, when the weight values undergo multiple updates. Upon the completion of training, the learned weights are copied onto the FG transistors for permanent storage. The peripheral circuits provide support for fast programing and interfacing with the external world. The GM blocks convert voltage-encoded input signals (sensor readings) into balanced differential currents required by the core. The digitally-controlled current source IDAC generates target currents from an on-chip reference for dynamic programming of the CSCs. Finally, the current-to-voltage converter ITOV facilitates the reading of internal currents by converting them to voltages that can be sampled by an external ADC. Each of the blocks undergoes extensive characterization to provide the reading/sourcing accuracy of at least 8 bits.

B. Weight Storage

The principle of weight storage is illustrated in Fig. 3. We use a multiple-input FG transistor (FGT) P1 to store the drain

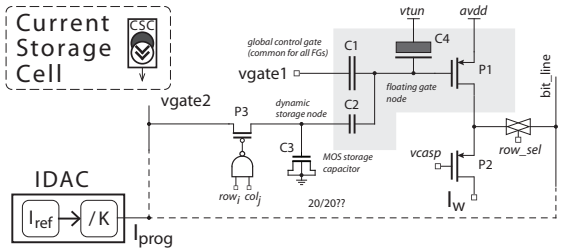

Fig. 3. Schematic of the current storage cell along with the dynamic programming loop. The sizes of key components are as follows: P1 = 2×2 μm^2, P2 = 2×1 μm^2, P3 = 0.4×0.35 μm^2; C1 = 40 fF, C2 = 15 fF, C3 = 1.35 pF, C4 = 0.6 fF.

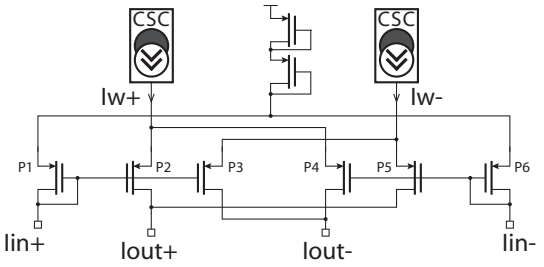

Fig. 4. Schematic of the synapse circuit (P1 = ... = P6 = 4×2 μm^2).

current I_w representing one of the weight value components. The drain current is modulated by the voltage on the FG node, which is itself determined by the FG node charge and the voltages on two control gates. The global voltage *vgate1* of the first control gate is shared among all FGTs, while *vgate2* is stored locally in the dynamic sample-and-hold (S/H) circuit which consists of the switch transistor P3 and the MOS capacitor C3. The low-coupling capacitor C2 makes I_w much less sensitive to charge leakage and other parasitic effects of the S/H circuit. The tunneling capacitor C4 is implemented as a minimum size PMOS transistor with its source, drain and well terminals connected to *vtun*. The details of non-volatile and dynamic programming are described in [7].

C. Synapse and Neuron Circuits

The synapse circuit, illustrated in Fig. 4, implements a four-quadrant multiplication of a differential input current $\{I_{in}^+, I_{in}^-\}$ by a differential weight current $\{I_w^+, I_w^-\}$. The circuit features two CSC cells for differential weight components storage and a six-transistor core P1-P6. The neuron circuit, illustrated in Fig. 5, applies a nonlinear activation function to the sum of the outputs of the connected synapses. The nonlinear transformation is completed in two stages. The first stage, represented by the bottom part of the circuit, controls the slope of the activation function. The slope is adjusted by programming the I_{gain} current, which is stored in a local CSC. The top part (P1-P6) performs nonlinear transformation of the normalized input current. The common-mode signal I_{neur} of the output current is set by a separate FGT.

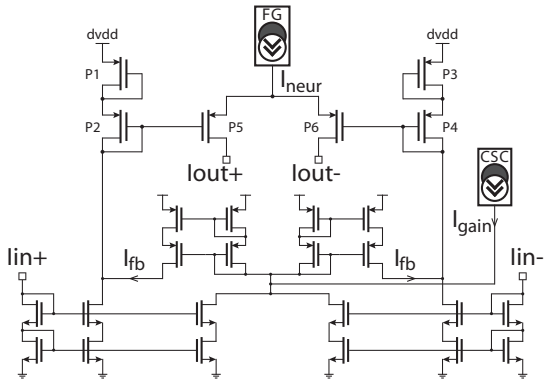

Fig. 5. Schematic of the neuron circuit. All PMOS and NMOS transistors have size 4×1 μm^2.

IV. Experimental Results

A. Weight Decay

Errors in dynamic programming directly affect the learning ability of the analog neural network. Reverse-bias leakage current of the switch transistor P3 (Fig. 3) is particularly important since it determines the time window during which the weight change remains insignificant. In order to characterize this effect we measured weight decays for a randomly selected set of CSCs. The weight leakage rate r_w can be defined as a relative change of the weight value per unit time, or

$$r_w = \Delta I_w/(\Delta t \cdot I_w) \quad (1)$$

where I_w is the original weight current, ΔI_w is the weight change over time Δt. Fig. 6 shows a histogram of the leakage rate estimates collected for 35 randomly selected CSCs. Each cell was programmed to a number of initial currents from 0.1 nA to 30 nA. The normal distribution of the process variation parameters suggests that the leakage rate belongs to the family of lognormal distributions (leakage current affects the exponential part of the output drain current). Indeed, Kolmogorov-Smirnov test does not reject the null hypothesis (p-value=0.8). The leakage rate corresponding to the 90% percentile is 0.00443 sec^{-1}. If we assume that the weight value is represented by an 8-bit word, a 1-bit change occurs in 0.88 sec. It has been observed that the change of 1-bit results in virtually no change in the network output. To limit the exposure to weight decay, large datasets (>200 observations) are split into smaller chunks during the forward pass and separated by reprogramming the dynamic memory.

B. XOR2 Problem

We begin with the classical 2-input XOR task. It is well known that linear classifiers fail to allocate a boundary in this case. In fact, a multilayer perceptron (MLP) requires a minimum of two hidden neurons for this task. For power efficiency demonstration, we limited the operating currents to 1 nA (i.e. the output currents of the GM blocks, neurons and the maximum weight currents). The training started with just an output layer and successively added hidden layers

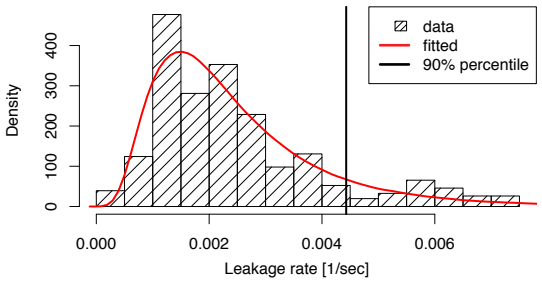

Fig. 6. A histogram of weight leakage rate estimates with a corresponding fitted lognormal density function.

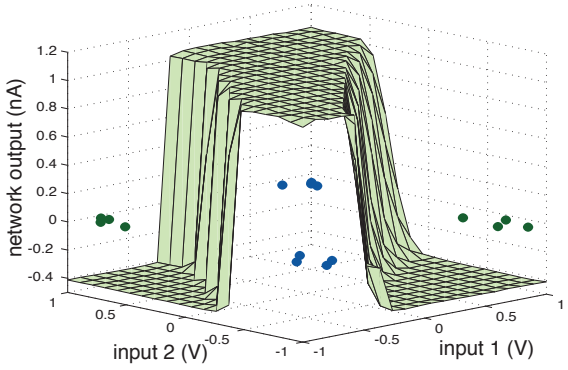

Fig. 7. Decision surface for XOR2 task obtained by measuring network output on a fine grid in input space. Also shown are the input patterns with added noise.

(neurons) until all 4 patterns were classified correctly. The training consistently converged with a single hidden neuron. Fig. 7 illustrates the output produced by the trained ONN with one hidden neuron. The output neuron is programmed for high gain, which explains the rail-to-rail response. The transient characteristic of the neural network is presented in Fig. 8. The response time is 4 ms, which also includes the propagation delay due to the GM and ITOV converters. The system performance and comparison data are summarized in Table I.

C. LNA chips

We proceed by training the hardware neural network to separate faulty from functional LNA chips using on-chip sensor measurements. The LNA chip is integrated with two RF amplitude detectors placed at its input and output ports (for detailed description see [1]). Two test stimuli are applied to the LNA and the outputs of the amplitude detectors comprise a 4-dimensional input pattern presented to the neural classifier. The LNA chip is labeled as either functional or faulty according to the set of specifications. The training and test sets consist of 900 and 1,000,000 instances, respectively. This large number of instances is obtained by generating a synthetic population of LNA chips from the original post-layout Monte-Carlo simulations [1].

The training was repeated for a total of 10 times for network topologies of up to 6 hidden neurons. A portion of training instances was set aside for calculation of validation errors which are shown in Fig. 9. The fact that the error does not

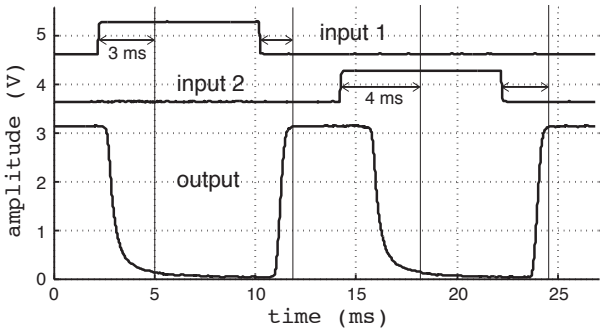

Fig. 8. Transient characteristic of the ONN trained to classify XOR2 patterns. The output is recorded from the voltage output pin of the ITOV converter.

TABLE I

SYSTEM PERFORMANCE AND COMPARISON

	ETANN [8]	[9]	this work
Technology	1 um CMOS	0.35 um, DP	0.35 um, DP
Weight storage	floating gate	floating gate	FG + dynamic
Learning models	MLP	VMM+WTA	MLP+ONN
Learning strategy	off-chip	off-chip	chip-in-the-loop
Synapse current	20 um @5V	10 nA @2.4V	2 nA @3.3V
Response time	5 us	NA	4 ms
Total power (XOR2 taks)	NA	700 nW	66 nW
Computation efficiency	1.3 GMAC/s/W	11–14 TMAC/s/W (theoretical)	57 GMAC/s/W (measured)

improve after a few hidden neurons have been added to the topology suggests that the optimal boundary is fairly simple. The test error evaluated for the best topology having 2 hidden neurons is 0.48%. This result agrees well with our previous hardware neural network implementation [1]. For the same classification problem, the test error achieved by the hardware perceptron with 2 and 4 hidden neurons was 0.727% and 0.435%, respectively. The software training resulted in 0.566% and 0.548% test errors.

This result suggests that the classification accuracy is similar across various implementations. The major improvement of the ONN chip is achieved with respect to power efficiency and non-volatility. Table II lists the mean and standard error of power consumption for all 10 runs of the training algorithm. It should be noted that we raised the minimum level of currents to 20 nA as compared to the XOR training. Yet, the power consumption is significantly lower than that required by our previous implementation ($\approx$ 0.5 mW). Finally, we copied the learned weights onto the FGTs and verified that the test error remained consistent within several days. The long term study of non-volatile weight retention is yet to be conducted.

V. CONCLUSIONS

The system characteristics of the presented analog implementation of the ONN make it a great candidate for on-chip inclusion as a part of RF BIST. Low power is achieved by biasing all circuits below threshold and employing the translinear principle for analog computation. Efficient dynamic weight programming is achieved by storing dynamic voltage on a MOS capacitor which drives a low-coupling gate of a mutliple-input FGT. Non-volatile weight storage is made possible through the FG technology available in double-poly CMOS. Finally, the ONN chip demonstrated excellent results in predicting pass/fail labels for LNA circuits using low-cost sensor measurements.

TABLE II

POWER CONSUMPTION FOR TRAINED LNA CLASSIFIERS

No. of hidden layers	0	1	2	3	4	5	6
Mean power consumption (μW)	0.551	1.54	2.59	3.65	4.71	5.89	7.15
Stderr power consumption (μW)	0.068	0.126	0.846	1.20	1.54	2.31	2.73

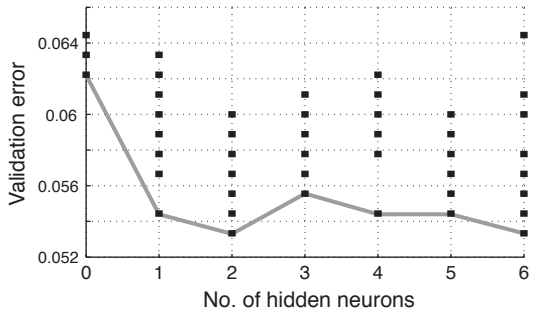

Fig. 9. The results of 10 runs of the training algorithm on the LNA task. The best validation error achieved for each number of hidden neurons is highlighted by the solid line.

REFERENCES

[1] D. Maliuk, H.-G. Stratigopoulos, H. He, and Y. Makris, "Analog neural network design for RF built-in self-test," in *Proceedings of the IEEE International Test Conference (ITC)*, 2010, pp. 23.2.1–23.2.10.

[2] L. Abdallah, H.G. Stratigopoulos, S. Mir, and C. Kelma, "RF front-end test using built-in sensors," *IEEE Design and Test of Computers*, vol. 28, no. 6, pp. 76–84, 2011.

[3] H.-G. Stratigopoulos and Y. Makris, "Error moderation in low-cost machine learning-based analog/RF testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 339–351, 2008.

[4] C.R. Schlottmann and P.E. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 403–411, 2011.

[5] A. Bandyopadhyay, G. J. Serrano, and P. Hasler, "Adaptive algorithm using hot-electron injection for programming analog computational memory elements within 0.2% of accuracy over 3.5 decades," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 9, pp. 2107–2114, 2006.

[6] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Proc. Advances Neural Inform. Process. Syst*, 1990, vol. 2, pp. 524–532.

[7] D. Maliuk and Y. Makris, "A dual-mode weight storage analog neural network platform for on-chip applications," in *IEEE International Symposium on Circuits and Systems*, 2012, pp. 2889–2892.

[8] H. A. Castro, S. M. Tam, and M. A. Holler, "Implementation and performance of an analog nonvolatile neural network," *Analog Integrated Circuits and Signal Processing*, vol. 4, no. 2, pp. 97–113, 1993.

[9] S. Ramakrishnan and P. Hasler, "The VMM and WTA as an analog classifier," *IEEE Transactions on Very Large Scale Integration Systems*, in press.