Self-Testing Analog Spiking Neuron Circuit

Sarah Ali El-Sayed, Luis Camuñas-Mesa, Bernabe Linares-Barranco, Haralampos-G. Stratigopoulos

Self-Testing Analog Spiking Neuron Circuit

Sarah A. El-Sayed*, Luis A. Camuñas-Mesa†, Bernabé Linares-Barranco†, Haralampos-G. Stratigopoulos*
*Sorbonne Université, CNRS, LIP6, Paris, France
†Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC y Universidad de Sevilla, Sevilla, Spain

***Abstract*—Hardware-implemented neural networks are foreseen to play an increasing role in numerous applications. In this paper, we address the problem of post-manufacturing test and self-test of hardware-implemented neural networks. In particular, we propose a self-testable version of a spiking neuron circuit. The self-test wrapper is a compact circuit composed of a low-precision ramp generator and a small digital block. The self-test principle is demonstrated on a spiking neuron circuit design in 0.35μm CMOS technology.**

I. Introduction

Artificial intelligence (AI) and deep learning [1] algorithms are increasingly dominating the computer industry, with applications ranging from speech and object recognition all the way to robotics, Internet-of-Things (IoTs), autonomous vehicles, smart healthcare, etc.

To this day, the actual processing typically runs on giant servers in the cloud using general-purpose central processing unit (CPU) clusters. This hardware is too large to fit inside a portable device and it needs far more power than a device battery can provide. Compared to the human brain that has a neuron density of around 10,000/mm^2 and consumes approximately 10^{-11} Joules per spike [2] -which means it runs on 20 Watts-, CPUs are estimated to be about 10^8 less efficient in terms of size and energy consumption. In addition, several AI applications require a real-time response, in which case performing inference or on-line learning with a neural network running in software on a CPU is not an option. Similar limitations are observed when graphic processing units (GPUs) and field-programmable gate arrays (FPGAs) are used instead. All of these restrictions, along with other more technical challenges, such as the von Neumann bottleneck and the approaching end of Moore's law [3], have made it crucial to find alternative architectures.

The answer comes with neuromorphic computing, a term introduced by Carver Mead in the late 1980s [4] referring to special purpose very large-scale integration (VLSI) artificial neural network (ANN) implementations that resemble -or are inspired from- biology. The work that followed has led to the advent of VLSI implementations of neural networks that can work as customized hardware accelerators or can offer a much smaller form factor and better energy efficiency, such that they can be used in resource-constrained IoT nodes for near-sensor computation and near-sensor intelligence.

Today, there exist many hardware neural network architectures [5]–[7] that can be digital, mixed analog-digital, or purely analog. Analog implementations are more efficient in terms of power consumption and form factor [8]. However, they are less robust than digital implementations since digitally stored synaptic weights and digital arithmetic operations carried out by neurons are not affected by process variations. On the other hand, there are emerging architectures that use nonvolatile memory devices for implementing efficiently the synapses [9].

With the foreseen industrialization and high-volume production of hardware neural networks in the coming years, testing strategies specific to hardware neural networks is an emerging topic that is largely unexplored [10].

Post-manufacturing testing aims at detecting manufacturing errors and is done per manufactured chip. For a high-volume production, testing is performed on automated test equipment (ATE) and needs to be completed in a few seconds. For safety- and mission-critical applications, testing also needs to be performed in the field concurrently with the operation or in idle times, in order to detect latent defects, aging, etc. For this purpose, built-in self-test (BIST) capabilities need to be added into the design that allow stand-alone evaluation of the health status of the chip without relying on external test instruments. BIST is a critical block for putting in place a self-healing and self-repair methodology towards fault-tolerance and dependable design, where the chip is capable of detecting and correcting errors in its operation on its own.

Testing strategies vary depending on the type of the integrated circuit (IC), i.e. digital or analog. Techniques for post-manufacturing testing and BIST for digital ICs are considered quite mature nowadays [11], [12]. For analog ICs, post-manufacturing testing still relies mostly on measuring sequentially the performances that are promised in the data sheet and comparing them to their specifications [13]. BIST solutions are specific to the IC class (i.e. ADC, DAC, PLL, filter, op-amp, RF, etc.) and specific to different architectures within each IC class (i.e. SAR, pipeline, $\Sigma\Delta$, etc., architectures for the ADC class). BIST for analog ICs is not very widespread because of the many practical challenges. For example, analog signal paths are sensitive and BIST circuitry tapping into them loads the IC and degrades the performance.

Most likely, post-manufacturing testing for digital neural network implementations will not be any different than testing of regular digital ICs. Developing BIST solutions, however, can benefit from the modularity observed in neural network architectures. On the other hand, for purely analog neural network implementations or implementations with analog sub-blocks, i.e. analog neurons, new post-manufacturing and BIST test strategies will need to be developed since, in the first place, it is not clear what specifications we should test for.

While the "inverse" problem of applying machine learning for solving test-related tasks is extensively studied [14], a few papers have been published recently on testing of hardware neural networks. In [15], a neuromorphic BIST architecture was proposed for analog ICs that has as its central block an on-chip neural network that can classify simple measurements directly to 1-bit pass or fail test decisions. However, the test of the neuromorphic BIST wrapper was not studied. In

978-1-7281-1201-5/19/$31.00

[16], it is proposed to use a checker neuron for real-time resilience against soft errors in feed-forward neural network architectures. The checker neuron estimates the output of a layer, compares it against the output of the actual neuron layer, and when the difference is outside a specified limit, it triggers an error signal. In this case, the error can be corrected by bypassing the whole layer and subsequently re-training. In [17], a fault-tolerant design of Google's Tensor Processing Unit (TPU) hardware accelerator is proposed. The TPU is composed of a grid of Multiply-And-Accumulate (MAC) units. If a MAC unit is detected to be faulty, then it is pruned, i.e. bypassed, using multiplexers that are added into the design. Fault-tolerance design of Resistive Random Access Memory (RRAM)-based neuromorphic architectures is discussed in [18]. A fault-tolerant design of a spiking neural network used for controlling the motion of a robotic car and implemented in an FPGA is proposed in [19]. The fault-tolerance is based on employing redundancy, i.e. additional synapses, and is application-specific.

In this paper, we focus on spiking neural networks and we propose a self-test approach for an analog spiking neuron circuit. The self-test approach is based on applying a low-precision ramp to the biases of the neuron and using a digital block at the output to register the output firing patterns and match them to all expected firing patterns that the neuron is capable of producing. If one or more firing patterns are missing, then the neuron is declared faulty. A single compact BIST wrapper can be used to sequentially test all neurons.

The rest of the paper is structured as follows. In Section II, we provide a brief description of the neuron employed in our study and the principle of its operation. In Section III, we describe the BIST architecture. In Section IV, we provide test coverage results based on the proposed BIST architecture. Finally, Section V concludes the paper.

II. Analog Spiking Neuron Circuit

According to their computational paradigm, ANNs can be classified into three generations [20]. The first generation is based on the McCulloch-Pitts neurons and designed to give digital outputs, while the second generation is based on neurons that implement an "activation function", such as the sigmoid function. Structure-wise, these two generation models are very different from biological neural networks and they cannot match their efficiency. The third generation of neural networks employs spiking neurons that can code information in a spacio-temporal manner, making them more biologically realistic compared to the previous two models.

The spiking neuron is the basic building block of neuromorphic systems. Spiking neurons communicate -much like their biological counterparts- through discrete electrical pulses called spikes or action potentials. This means that instead of constantly firing, as in the case of first and second generation neurons, a spiking neuron accumulates inputs from preceding neurons, and if a certain number of spikes occur within a specific time frame, it generates a spike of its own. This event-driven operation is the basis of the huge power savings offered by neuromorphic computing.

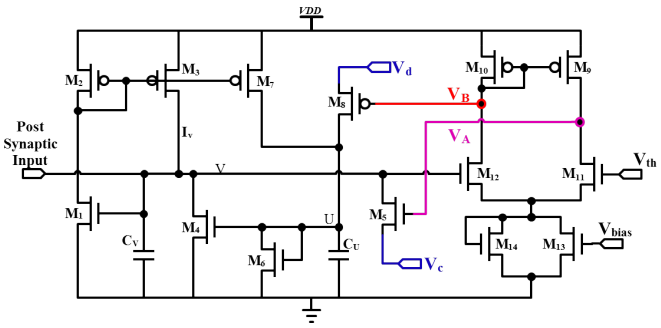

Fig. 1. Transistor-level design of spiking neuron circuit.

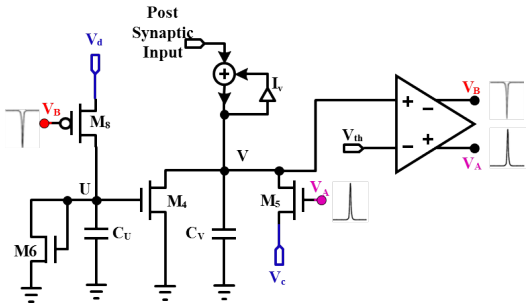

Fig. 2. High-level model of spiking neuron circuit.

There are numerous models of spiking neurons used in neuromorphic systems [21], and most of them perform this accumulation and firing function, albeit with different mechanisms. Of these models, the integrate-and-fire (I&F) neuron is perhaps the simplest model that is still complex enough to be considered biologically inspired [7].

As known from neurobiology, neurons in the brain respond to post-synaptic stimuli in one of four basic patterns [22]. The first pattern, called *Regular Spiking (RS)*, is when the neuron fires repeatedly with a gradually decreasing frequency, i.e. it adapts, until it settles to a stable frequency. The second, called *Fast Spiking (FS)*, is when the neuron fires at a high frequency with little or no adaptation. The third, called *Intrinsic Bursting (IB)*, is when the neuron starts with a cluster of spikes followed by repetitive smaller clusters or single spikes. And the fourth pattern, called *Chattering (CH)*, is when the neuron response is in the form of long clusters of spikes occurring at regular intervals.

The transistor-level design of the neuron chosen for this work is shown in Fig. 1. It is a compact CMOS implementation of an I&F neuron [23] based on the mathematical model proposed in [24]. While most I&F models are incapable of generating both spiking and bursting behaviors in a single circuit, this neuron uses only 14 transistors and 2 capacitors to produce all four basic firing patterns of a real neuron by varying two control voltages, namely V_c and V_d.

Fig. 2 explains the most important aspects of the circuit's operation. The spiking behavior of the circuit is represented by two variables, V and U, which are the voltages accumulated on capacitors C_V and C_U, respectively. Capacitor C_V integrates the post-synaptic input current in addition to a positive feedback current I_V. When V reaches the threshold value, the comparator generates two short-duration pulses, namely V_A and V_B, and the circuit spikes. Pulse V_A activates transistor M_5, discharging capacitor C_V and hyperpolarizing V to a

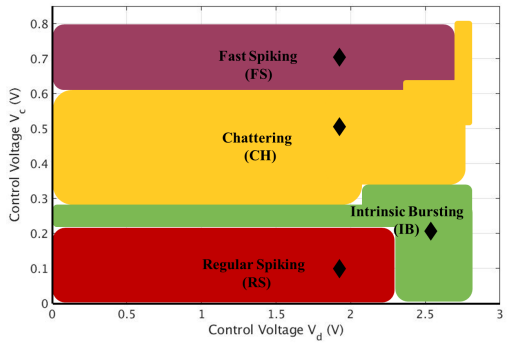

Fig. 3. Approximate areas in the control voltages space V_c-V_d that produce the different firing patterns. The diamond points correspond to the nominal control voltages combinations used to produce each firing pattern.

predetermined value V_c. Pulse V_B turns on transistor M_8, which has a narrow channel, so that only a small amount of charge from V_d passes to capacitor C_U. The two capacitors are sized so that C_U charges more slowly than C_V. With every spike, voltage on C_U increases a little, thus increasing the leakage current through M_4 and the current through M_6, which is diode-connected to act like a non-linear resistor that discharges C_U. Leakage slows down the charging of C_V, allowing the refractory period between spikes. By varying control voltages V_c and V_d, the relative speeds of V and U can be controlled and the four basic firing patterns can be obtained.

The neuron circuit is designed in the ams 0.35um HV CMOS H35B4D3 technology. Fig. 3 shows the four nominal combinations of control voltages V_c and V_d that produce the four distinct firing patterns. Fig. 3 also shows approximate areas in the V_c-V_d space that produce each firing pattern.

III. BIST ARCHITECTURE

The proposed BIST architecture for the spiking analog neuron is illustrated in Fig. 4. The BIST wrapper includes a ramp generator block that applies ramps at the two control voltages of the neuron aiming to excite the neuron across its different operation modes and produce the four firing patterns. In particular, V_d is ramped from 1.9V to 2.8V, and during this duration V_c is ramped once from 0.1 to full scale and then ramped again from 0.1 to half the full scale, as shown with the saw-tooth stimulus in Fig. 4. The rationale behind this ramping strategy can be made clear by looking in the approximated areas in Fig. 3. During the first full V_c ramp, the neuron should produce the RS, CH, and FS firing patterns. During the second half V_c ramp, the neuron should produce the IB firing pattern. The ramps do not have any stringent requirements. Low-precision step-wise ramp generators can be used in this context.

A digital block is connected to the output of the neuron and its role is to digitize the analog output, store the digital signature, and try to identify within this digital signature the four firing patterns, i.e. to match excerpts of the digital signature to the desired firing patterns. The rationale of our approach is that a functional neuron should be able to generate all intended firing patterns. If one or more firing patterns are missing, then the neuron is declared to be faulty. In essence, the proposed BIST architecture targets verifying one of the

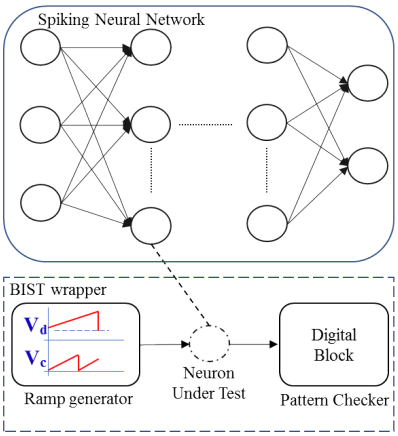

Fig. 4. BIST architecture.

functional specifications of the neuron, which is its ability to provide all firing patterns.

A single BIST wrapper is required to test the complete spiking neural network. In this case, the neurons can be connected sequentially to the BIST wrapper.

The proposed BIST architecture can be used for both post-manufacturing testing and self-test in the field in idle times or before on-line re-learning is attempted. Neurons identified as faulty can be neutralized by setting their input synapses to zero. In this way, having a neural network with only functional neurons will enable a faster and more robust learning.

Finally, the proposed BIST architecture tests the neurons themselves independently of the application and of the data that is processed through the neural network for training or inference. This dissociates the test procedure from the underlying training algorithm and the cognitive task that the neural network is performing. In other words, this BIST architecture is suitable for versatile use as it looks solely at the hardware.

IV. RESULTS

Fig. 5 shows the simulation of the nominal neuron by applying the BIST stimuli, i.e. by ramping the two control voltages. As it can be seen, the four firing patterns appear at the output. Specifically, RS appears first between 5 and 10 μs, CH follows between 10 and 20 μs, FS comes next between 20 and 25 μs, and IB appears last between 25 and 30 μs. A few μs is enough to excite the neuron in all four operation modes, thus the test is very fast.

The BIST principle is that a faulty neuron looses its ability to produce one or more firing patterns. A neuron could be faulty due to process variations or due to defects, i.e. random spots or voids on the die surface that may occur due to errors during the manufacturing steps and that translate into short- and open-circuits or extreme variation.

Neurons with process variations are generated by performing a Monte Carlo (MC) analysis with mismatch and inter-die variations using the statistical process design kit (PDK) of the technology. Specifically, we performed a MC analysis with 100 runs. Fig. 6 shows the response of the neurons in the first 5 runs. As it can be seen, the second and fifth neuron are not producing all four firing patterns, thus they are

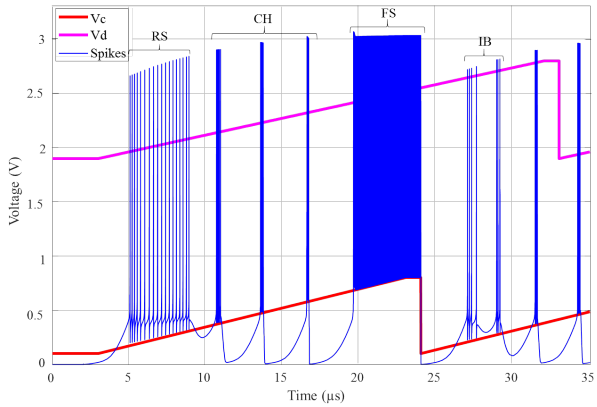

Fig. 5. Neuron output response to BIST stimuli.

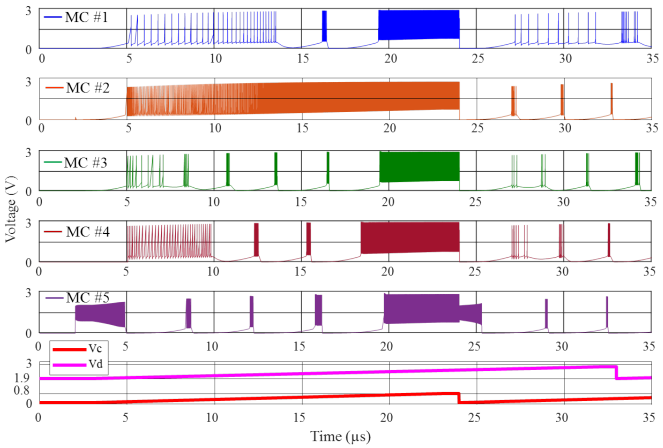

Fig. 6. Monte Carlo analysis showing the neuron output response to BIST stimuli for 5 runs.

detected by the BIST. In total, the test yield by the BIST is around 70%, showing that analog neuron circuits can suffer a lot from process variations, thus requiring a thorough and comprehensive post-manufacturing test procedure.

Defective neurons are generated by assuming a classical defect modeling approach [25]. Specifically, for transistors we consider open-gate defects following the simulation method in [26], and drain-to-source shorts; considering additional shorts across other terminals and opens in other terminals are shown to be redundant in practice and only increase defect simulation time. For capacitors, we consider short- and open-circuits and $\pm 50\%$ variation. One defect is injected in the neuron at a time manually. The proposed BIST approach was capable of achieving 90.6% defect coverage. Only three defects are not detected, namely $\pm 50\%$ variation in C_U and $+50\%$ variation in C_V. It turns out that the neuron can be tuned to accommodate these variations.

V. Conclusions

We proposed a compact BIST architecture for an analog spiking neuron that can be shared among all neurons in the neural network. This architecture consists of a ramp generator block that controls two bias voltages to produce all four different firing patterns at the output in the fault-free case, and a digital block that checks if any of these patterns is missing, in which case it flags a fault detection. The BIST achieves a test yield of 70% in the case of process variations and over 90% defect coverage.

Acknowledgments

This work has been carried out in the framework of the Penta HADES project. Luis A. Camuñas-Mesa was funded by the VI PPIT through the Universidad de Sevilla.

References

[1] I. Goodfellow et al., *Deep Learning*, MIT Press, 2016.
[2] C.-S. Poon and K. Zhou, "Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities," *Frontiers in Neuroscience*, vol. 5, 2011, Article 108.
[3] T. N. Theis and H.-S. P. Wong, "The end of moore's law: A new beginning for information technology," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.
[4] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.
[5] R. Serrano-Gotarredona et al., "CAVIAR: A 45k neuron, 5m synapse, 12g connects/s AER hardware sensory–processing– learning–actuating system for high-speed visual object recognition and tracking," *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1417–1438, 2009.
[6] G. Volanis et al., "Toward silicon-based cognitive neuromorphic ICs—a survey," *IEEE Design & Test*, vol. 33, no. 3, pp. 91–102, 2016.
[7] C. D. Schuman et al., "A survey of neuromorphic computing and neural networks in hardware," *arXiv:1705.06963v1*, 2017.
[8] M. Valle, "Analog VLSI implementation of artificial neural networks with supervised on-chip learning," *Analog Integrated Circuits and Signal Processing*, vol. 33, pp. 263–287, 2002.
[9] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, 2018.
[10] L. Anghel et al., "Neuromorphic computing - from robust hardware architectures to testing strategies," in *Proc. IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2018, pp. 176–179.
[11] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Kluwer Academic Publishers, 2000.
[12] S. Mitra and E. J. McCluskey, "Which concurrent error detection scheme to choose ?," in *Proc. IEEE International Test Conference*, 2000, pp. 985–994.
[13] G. Roberts et al., *An Introduction to Mixed-Signal IC Test and Measurement*, Oxford University Press, 2011.
[14] H.-G. Stratigopoulos, "Machine learning applications in IC testing," in *Proc. IEEE European Test Symposium*, 2018.
[15] D. Maliuk et al., "Analog neural network design for RF built-in self-test," in *Proc. IEEE International Test Conference*, 2010, Paper 23.2.
[16] S. Pandey et al., "Error resilient neuromorphic networks using checker neurons," in *Proc. IEEE International Symposium on On-Line Testing And Robust System Design*, 2018, pp. 135–138.
[17] J. J. Zhang et al., "Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator," in *Proc. IEEE VLSI Test Symposium*, 2018.
[18] M. Liu et al., "Design of fault-tolerant neuromorphic computing systems," in *Proc. IEEE European Test Symposium*, 2018.
[19] A. P. Johnson et al., "Homeostatic fault tolerance in spiking neural networks: A dynamic hardware perspective," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 687–699, 2018.
[20] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
[21] G. Indiveri et al., "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, 2011, Article 73.
[22] B. W. Connors and M. J. Gutnik, "Intrinsic firing patterns of diverse neocortical neurons," *Trends Neurosciences*, vol. 13, no. 3, pp. 99–104, 1990.
[23] J. H. B. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and burtsing behaviour," *Neural Networks*, vol. 21, no. 4, pp. 524–534, 2016.
[24] E.M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.
[25] S. Sunter et al., "Using mixed-signal defect simulation to close the loop between design and test," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2313–2322, 2016.
[26] B. Esen et al., "Effective DC fault models and testing approach for open defects in analog circuits," in *Proc. IEEE International Test Conference*, 2016, Paper 3.2.