

An Oscillatory Neural Network with Programmable Resistive Synapses in 28 nm CMOS

Thomas Jackson
ECE Department
Carnegie Mellon University
Pittsburgh, USA
thomasjackson@cmu.edu

Samuel Pagliarini
ECE Department
Carnegie Mellon University
Pittsburgh, USA
pagliarini@cmu.edu

Lawrence Pileggi
ECE Department
Carnegie Mellon University
Pittsburgh, USA
pileggi@cmu.edu

Abstract—Implementing scalable and effective synaptic networks will enable neuromorphic computing to deliver on its promise of revolutionizing computing. RRAM represents the most promising technology for realizing the fully connected synapse network: By using programmable resistive elements as weights, RRAM can modulate the strength of synapses in a neural network architecture. Oscillatory Neural Networks (ONNs) that are based on phase-locked loop (PLL) neurons are compatible with the resistive synapses but otherwise rather impractical. In this paper, A PLL-free ONN is implemented in 28 nm CMOS and compared to its PLL-based counterpart. Our silicon results show that the PLL-free architecture is compatible with resistive synapses, addresses practical implementation issues for improved robustness, and demonstrates favorable energy consumption compared to state-of-the-art NNs.

Keywords—oscillatory neural network, neuromorphic computing, RRAM, PLL

I. INTRODUCTION

Neural networks (NNs) are systems inspired by the brain. The brain is extremely power efficient; it is able to solve complex image and audio processing problems while consuming less than 20W [1]. In fact, biological systems outperform any modern computer by orders of magnitude in terms of power efficiency. One of the main reasons for the performance gap between the brain and contemporary silicon-based neural networks is that the silicon systems do not take full advantage of the properties of neural networks which can save power. This work seeks to take advantage of these properties to build efficient neural networks in hardware.

NNs are intriguing architectures capable of solving many interesting big data problems, such as image & video classification [2], natural language processing [3], etc. NNs have many different classes and types, but are generally built around two basic units: Neurons and synapses. Neurons are the effective processing element while synapses act as some form of weighted connection between neurons.

As neuromorphic computing advances, the network sizes scale in the number of neurons and in the number of synapses. However, the number of synapses scales significantly faster – even quadratically in the worst case. Therefore, an efficient implementation of the synaptic network is imperative to make

such systems realizable and an effective alternative computing paradigm.

Resistive memory technology, such as RRAM, is a promising artificial synapse [4] with its non-volatile storage capability and ability to program a continuum of (analog) resistance values as weights. Oscillatory Neural Networks (ONNs) [5] can, in theory, utilize a resistive synaptic network to control time or phase state variables. Therefore, the use of RRAM would alleviate the concerning scaling trends of number of synapses in neuromorphic circuits. The basic principle behind ONNs is that neurons will interact with each other through weighted synaptic connections until the entire system achieves global consensus.

To date, however, no functional large-scale ONNs have been demonstrated in silicon. The most hindering difficulty comes from the impractical requirement of having a phase-locked loop (PLL) for emulating each and every artificial neuron. Even implementing a simplistic all-digital PLL may require hundreds of CMOS gates [6]. Analog PLLs are smaller in size, but related practical implementation issues are also of great concern (e.g., non-idealities and process variation). In this paper we address some of the aforementioned challenges by proposing an ONN architecture that is free of PLLs.

The remainder of this paper is organized as follows: In Section II we describe PLL-based ONNs and the difficulties to implement these system. In Section III we describe the alternative PLL-free architecture and circuit which does not suffer from the same difficulties. Our silicon results for an ONN with 100 neurons and 10,000 synapses are described in Section IV. Finally, we draw conclusions in Section V.

II. PLL-BASED OSCILLATORY NEURAL NETWORKS

In one of the earliest works to consider PLL-based ONNs [5], the authors built a small-sized network using discrete and off-the-shelf PLL components. They have shown that the network can memorize and reproduce complex oscillatory patterns in which all neurons oscillate with the same frequency, but present different phase relations. The analysis made by the authors of [5] also proved that all PLLs in an ONN will synchronize to the same frequency and to a relative phase of either 0° or 180° without a global reference signal.

This work was supported in part by the National Science Foundation under contract CCF-1714334.

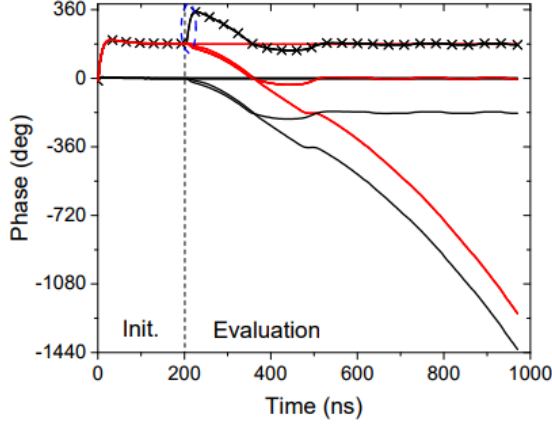


Fig. 1 Phase of the neuron outputs as a function of time. The PLL-based implementation loses synchronism after a short initial period (circled in blue).

However, this theoretical analysis does not consider real-world non-idealities such as propagation delay of signals in the network. In [7] we postulated that hardware implementation of ONNs will only synchronize if their input signals are retimed at the input of each neuron to ensure each neuron sees the same effective delay. The desynchronization of an ONN with random delays on the inputs is shown in Fig. 1 – our simulations show that frequency synchronization is not achieved other than during a very short initial period. After this initial period, the neuron phase keeps changing and the system therefore drifts continuously, resulting in an ONN operation that fails to reach consensus.

The figure depicts the phase of the neuron outputs as a function of time in nanoseconds. The phase of the output of neuron #1 is adopted as the reference phase. Before 200 ns, the PLL phases are initialized to either 0 or 180 degrees. The red lines represent neurons that should settle to 180 degrees while the black lines represent neurons that should settle to 0 degrees. The lines labeled with the ‘X’ are from two neurons that should shift from 180 to 0 degrees when in correct operation. Instead of synchronizing in frequency, some of the neurons fail to synchronize, accruing negative phase relative to the reference neuron. When the system starts the evaluation phase at approximately 200ns, synchronization is briefly achieved. Unfortunately, the process of frequency desynchronization happens on a fairly similar timescale, so the correct solution is quickly lost.

A delay correction mechanism can be sought to correct desynchronization issues. We have implemented a PLL-based system with delay correction in 28 nm CMOS using a clock-based delay circuitry to confirm this hypothesis for an analog ONN. Our delay correction scheme uses a technique borrowed from classic PLL design known as re-timing [8]. Although it demonstrated synchronization, the use of PLLs led to other issues that ultimately led to a system that was not sufficiently robust to noise.

Re-timing is implemented in hardware with a clocked-comparator, which must be designed such that it operates at a higher frequency than the neurons. However, the clock period

should be sufficiently long to accommodate any of the individual neuron delays (one can think of this constraint as being akin to setup and hold requirements in digital circuits). To avoid the use of an external clock for the system when it is in evaluation mode, the comparators are all directly clocked from the voltage controlled oscillator (VCO) output of one of the neurons. The VCO output is a signal with a higher frequency than the neuron outputs because it is sent through a frequency divider. With a re-timing comparator in the neuron, the system synchronizes as predicted.

However, when asked to store and recover trained patterns, the system did not display the expected behavior. Although the system intent is to recover the closest stored pattern, the system instead returned an arbitrary stored pattern. Even worse, the pattern it returned is nondeterministic. That is, even with the same initial conditions and initial training, the network settles to a stored pattern randomly. The fact that the system returns one of the stored patterns indicates that some amount of the system functionality is preserved. Unfortunately it is not possible to measure the system state directly as such degree of observability is typically not available in real hardware unless specifically built-in for this purpose.

Since the system did return a stored pattern, we have tried to identify the source of the nondeterministic behavior. A reasonable hypothesis is that some noise source causes a disturbance to the state of the system that causes it to lose its initial condition. One potential noise source is the phase noise of the PLLs, specifically the phase disturbance caused by noise on the power supply. It was observed in our test environment that output switching caused an unexpectedly large noise on the analog supply, even with the addition of decoupling capacitors on the test board. We were able to quantify the noise as being capable of producing 35 degrees of phase disturbance, which is rather significant when two given neurons are separated by 180 degrees.

More details about the implemented circuit can be found in [9], as well as an in-depth discussion of its shortcomings. We proceed now by presenting the main contribution of this work,

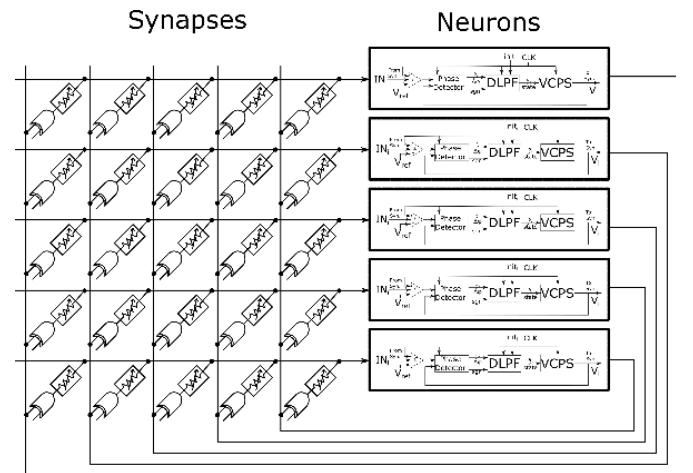


Fig. 2 A subset of the full PLL-free ONN consisting of 5 neurons and 25 synapses. The full network has 100 neurons and 10,000 synapses. Some initialization and control signals are removed for clarity.

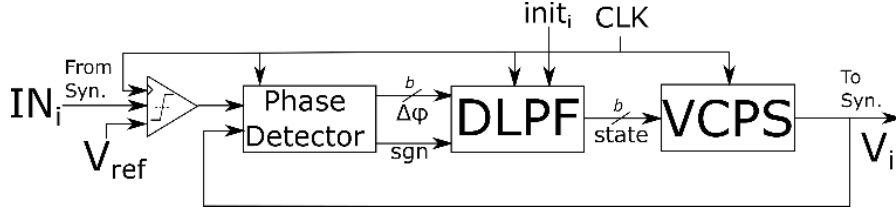


Fig. 3 The neuron for the PLL-free ONN. It consists of an input comparator, a phase detector, a digital low pass filter, and a voltage controlled phase shifter.

i.e., a circuit that does not rely on PLLs and is largely capable of tolerating noise.

III. PLL-FREE OSCILLATORY NEURAL NETWORK – CIRCUIT AND ARCHITECTURE

A scalable and PLL-free ONN architecture is depicted in Fig. 2. Only five neurons and 25 synapses are shown in the image, while the whole network consists of 100 fully-connected neurons (i.e., 10,000 synapses). The network is designed to operate at a clock frequency of 1 GHz. The PLL-free ONN architecture described in this section has a similar synapse as the PLL-based ONN previously described (we found that the emulated RRAM synapse network worked as expected and therefore could be reused). However, this design has an improved neuron design that merits a deeper discussion (i.e., we have heavily modified our baseline neuron design to achieve robustness).

The neuron in this system was designed to be primarily digital. A digital implementation allows the use of synthesis tools and ensures scalability to deeper technology nodes. A schematic of the neuron is shown in Fig. 3.

The general operation of the neuron is described as follows: The summed signal from the synaptic network enters the neuron from the left side. The summation occurs at the input node, with no need for additional circuitry. A clocked comparator rectifies this signal to full rail and retimes it to eliminate any delay difference seen by the neuron signals around the loop. This resulting signal is passed to a phase detector that measures the arrival time difference between the input signal and the neuron state. This value is recorded in number of clock cycles, and the information along with a signal indicating the order of arrival is output to the next stage. The next stage is a digital low-pass filter (DLPF) that holds the state of the neuron and provides the desired damped behavior to ensure noise does not adversely affect the system, as was the case with the PLL-based ONN. This signal is passed to a voltage controlled phase shifter (VCPS), which sets the phase of the neuron output signal according to the state of the neuron. Finally, the output is connected to the synaptic network to close the loop.

The step by step operation of this mostly-digital system is more observable than the previous design (i.e., easier to debug). The system state is stored in registers that can be shifted out after each clock cycle to inspect the system behavior at any given time. The clock signal can be externally driven one time step at a time for debugging, or connected to a high speed 1 GHz source to test the system at full speed.

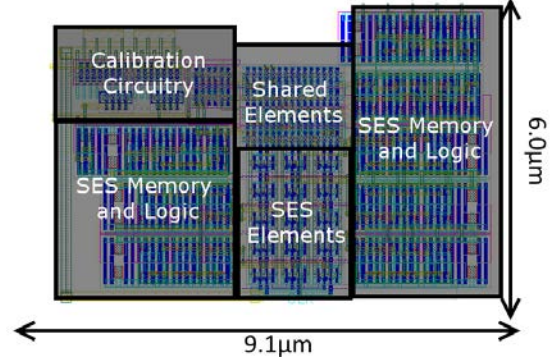


Fig. 4 Layout of the strongARM comparator that uses Statistical Element Selection [10] for post-manufacturing tuning of offset voltage. The configuration bits are stored in DFFs. It was determined from testing that the comparator offset had negligible impact on the quality of results for this chip.

The input comparator is the only analog component in the neuron. The waveform at the input of the neuron is typically not a full-rail signal, and therefore, it does not interface well with digital circuitry. The comparator is used to measure that signal relative to the mid-rail voltage, and rectify it to the full-rail voltage for the next stage in the network. Since the comparators are all clocked, they also eliminate any delay differences caused by weight patterns in the synapse network.

The comparator area is $54.6 \mu\text{m}^2$; even with calibration to ensure a small input offset voltage, the comparator only consumes a small portion of the overall neuron area. In Section IV we discuss the need for calibration circuitry and detail some of the results for the comparator since it is the only fully-custom designed block in the system. The neuron comparator circuit layout is shown in Fig. 4. All the remaining neuron components are built using digital standard cells and for this reason we show no layouts – digital circuits built with standard cells can be heavily automated by synthesis tools.

The neuron phase detector uses the global clock and a counter to track the arrival order and phase difference of the input and feedback signals. The phase detector is implemented with a counter with a small state machine to track the order of signal arrival. Moreover, the state of the phase detector can be read out via scan chain for debugging purposes.

The phase difference is passed into a low pass filter that holds the system state and provides the filtering that ensures stability and robustness to impulse noise (DLPF block in Fig. 3).

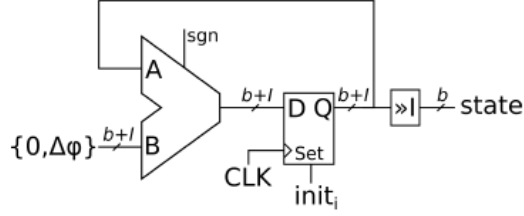


Fig. 5 Schematic of the digital low pass filter.

This filter consists of an adder/subtractor, registers to hold the system state, and a bit shifter that provides damping for the dynamics of the system. A schematic of the digital low pass filter is shown in Fig. 5. The parameters for this design are $b = 4$ and $I = 3$.

Regarding Fig. 5, the signals $\Delta\phi$ and sgn are supplied by the phase detector. The output of the phase detector is added (or subtracted, depending on signal arrival order) from the neuron's state. This is divided by a factor of 2^I before being passed to the next stage, giving the neuron an over-damped characteristic. The low pass filter is also used to set the initial condition of the network when using the network as an associative memory. The state of the system can be set directly using $init_i$ to initialize the network.

The final unit in the neuron is the VCPS. This block takes the state of the neuron and translates it into an output signal with a phase linearly related to the state of the neuron. It is implemented with a set of shift registers with the output of the last register feeding the input of the first register, shown in Fig. 6. The output phase is selected by using the neuron state as the selection bits of a MUX that selects the signal from one of the shift register outputs.

The width of the shift registers in the VCPS determines the resolution of the neurons in the system. In this system, sixteen flip-flops are used to generate the waveform, providing 4 bits of resolution. The waveform can be arbitrarily set, so higher frequency waveforms can be achieved at the expense of resolution.

Now that we have covered all blocks of the neuron, let us introduce the circuitry used for the synapses. We remind the reader that our system was designed using muxed silicon resistors to emulate RRAM synapses and therefore is designed to be compatible with emerging resistive memory technologies. The synapses in our ONN are implemented with silicon resistors combined with an XOR gate on every synapse, as shown in Fig. 7.

Each synapse contains 5 bits of memory for storing the weights (therefore providing 32 synapse discrete weights) and an additional sign bit. The total resistance of the synapse can range from about 645Ω to $20\text{ k}\Omega$ (i.e., conductances of $50\text{ }\mu\text{S}$ to 1.55 mS). While larger resistances are possible and would make the resistance range wider, the required silicon area for emulating these RRAM synapses would be above our budgeted area for the chip. We must note that emulating RRAM incurs considerable area that otherwise would not be required.

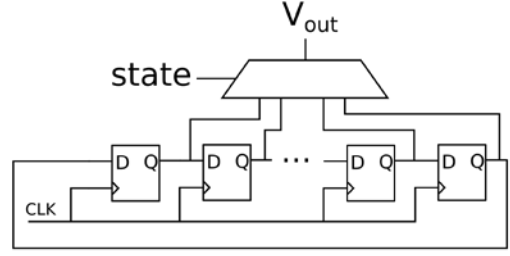


Fig. 6 The voltage controlled phase shifter.

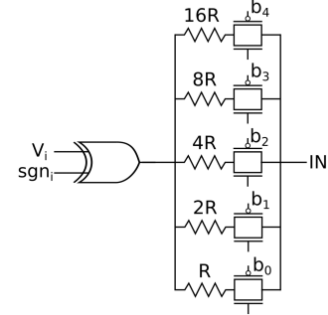


Fig. 7 Synapses used in the ONN chip to emulate programmable RRAM synapses [4].

The digital values of the synapses are stored in a scan chain which drives the transmission gates for each synapse ($b_0, b_1, \dots, b_4$). The synapse weights were designed to be values that can be achieved by demonstrated resistive memories. This is a fairly high range, however, so secondarily they were selected considering a trade-off between power and speed of the network. More power is drawn by smaller synapse resistors, since there is a constant voltage across them. On the other hand, if the resistors are too large then the RC delay created at the input node of the neuron will affect the network dynamics. Although some delay can be tolerated due to the re-timing technique, if the delay is too close to the length of a clock cycle then random variations can cause different delays for different neurons.

IV. ANALYSIS AND SILICON RESULTS

This section presents the simulation and post-fabrication results of the PLL-free ONN. We show that the PLL-free ONN is able to complete the associative memory task successfully, and does so consuming less power and area than the PLL-based ONN while mitigating the issues previously identified in Section II.

To demonstrate the functionality of this system, we performed simulations using three distinct patterns that were stored in the synapses using Hebbian learning. A behavioral model of the synapses and comparators was implemented in Verilog to allow for efficient simulation (electrical simulation of the entire chip is not feasible). The behavioral model takes the output of each neuron, sums them with weighted factors based on the synapse network, and then outputs either high or low based on whether the overall sum is positive or negative. Our

TABLE I
NEURAL NETWORK PERFORMANCE COMPARISON

Design	This work	PLL-based ONN	TrueNorth [11]
Tech.	28 nm	28 nm	28 nm
Neurons	100	20	871936
Area/Neur	275 μm^2	1000 μm^2	14.3 μm^2
Pow/Neur	303 μW	550 μW	72.3 nW
Time/Op	4 ns	4 ns	1 ms
Energy/Op	1.21 pJ	2.20 pJ	72.3 pJ

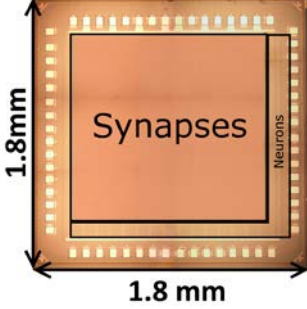


Fig. 8 Chip microphotograph.

simulations shown that the output of a given neuron starts at an incorrect phase, but over the course of a few cycles it is corrected by interacting with other the neighboring neurons.

A system with 100 neurons fully-connected through 10,000 synapses was implemented in TSMC 28 nm CMOS on a 1.8 mm by 1.8 mm die. A microphotograph of the chip is shown in Fig. 8. The chip contains approximately 6.3 million transistors and was designed to operate at 1 GHz. The synapses consume a total area of 1.69mm², while the neurons consume a total area of 0.0275mm². Since there are 100 neurons, each neuron has an approximate area of 275 μm^2 .

As highlighted in Fig. 8, the emulated RRAM synapses take the majority of the core area of the chip. The ratio of neuron vs synapse area would be considerably different if actual RRAM was used, therefore enabling the proposed architecture to scale very favorably.

The fabricated chip was tested as an associative memory to confirm its functionality. For example, the system can be used to store an “A” pattern and a “B” pattern, whereby each neuron represents one pixel. As expected, the chip returns the stored pattern that is closest to the distorted input (see Fig. 9). The implemented system can serially output the value of each neuron, line by line and column by column, as a one-bit digital output from which we can reconstruct the image pattern as shown in Fig. 9. Other patterns were tested similarly, and the behavior matched what was expected from theoretical analyses concerning training methods and spurious stored patterns.

The power and area metrics for this system are given in Table I, and shown to compare favorably to our PLL-based ONN that would not scale, and to the TrueNorth neural network [11].

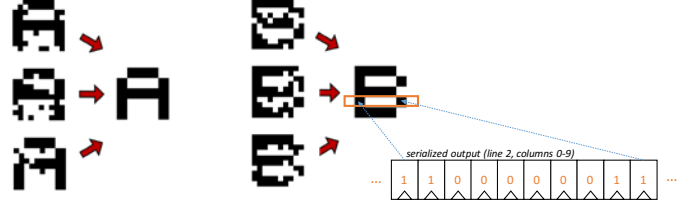


Fig. 9 PLL-free ONN acts as an associative memory. The distorted inputs are corrected to undistorted outputs. The outputs are stored in a scan chain, which is scanned out from the chip after the internal state has settled.

Compared to the TrueNorth chip, there are a few interesting points to consider. First, as with the PLL-based ONN, the neuron power is higher in the PLL-free ONN than in TrueNorth. This is balanced by the fact that the system is operating at a much higher frequency (250,000 times higher). Therefore, the neuron power is normalized by the operating speed to estimate the energy per operation, and using this metric the PLL-free ONN is significantly better than the TrueNorth results. It makes up for higher power by having much higher throughput.

It is also important to consider the scale of the networks in question. The TrueNorth system is significantly larger than the PLL-free ONN in terms of number of neurons. The power is normalized by this neuron count to get a power per neuron, but it is important to note that this does not take into account the inevitable overhead of system scaling. The larger the system, the more overhead will be needed for long distance communications and skew correction. Therefore, although the energy per operation numbers are very promising, they should be taken as optimistic. Despite this, since the energy per operation is an order of magnitude lower in the PLL-free ONN, it is still a promising path to consider.

Finally, the area of the neurons in TrueNorth is less than those in the PLL-free ONN, but this is primarily due to the silicon resistance synapses that were used to emulate the RRAM. With access to scalable RRAM, this area comparison would be much more favorable.

A. Statistical Element Selection

As we previously mentioned, the comparator is a full custom analog block with additional calibration circuitry. The calibration technique used in this design is referred to as statistical element selection (SES) [10].

Each comparator is designed and calibrated to achieve an offset voltage with a standard deviation of less than 1 mV. The comparator at the input to the neuron was designed based on the comparators described in [10]. These comparators use differential pairs with inputs IN+ and IN-. To apply SES, many copies of the differential pair are made. While all pairs are connected to the same input pair, only a subset of these pairs are selected such that each other's input offset voltages are cancelled out, therefore creating a comparator with low input offset. For each comparator, k input pairs are selected of N possible input pairs.

To build a comparator using SES, the number of selected and possible input pairs must be selected to meet the specifications.

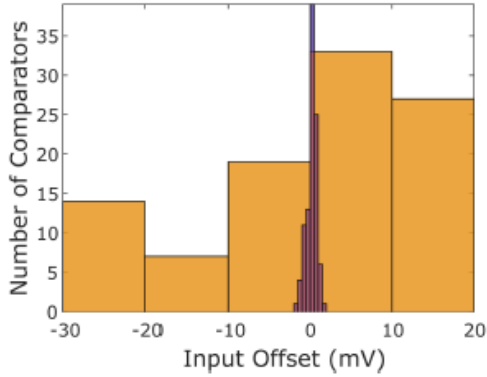


Fig. 10 Comparison of the SES-picked comparators versus typical configuration.

The first step is to find the standard deviation of the input offset voltage of an individual input pair in this technology. For this design, 500 Monte Carlo trials were run on the entire comparator, and the input offset voltage was found to have a normal distribution centered around 0V with a value of 37.60mV. Next, the same number of trials was run varying only the parameters on the input pair, and this yielded an input offset voltage of 36.34mV. This result indicates that the majority of input offset variation comes from the input pair.

Once the standard deviation of the input offset voltage is found, that information can be used to select the number of input pairs to build (N) and the number of those devices to select (k). The target for this design was an input offset voltage of 1mV with a yield of 70%. This is a smaller yield than would be desirable for a production chip, but it was targeted since 1mV is a particularly aggressive specification that was selected as a reasonable margin of safety.

The optimal point in this case was $N = 15$; $k = 5$. This chosen configuration was simulated in SPICE to ensure it would function correctly at 1 GHz. Our chip measurements confirmed that the SES technique was successful in achieving a small offset voltage on all of the comparators. This result is shown in Fig. 10. The orange bars represent the universe of possible offsets that can be measured if comparators are selected at random. However, by using SES the input offset can be restricted to a narrow band of approximately 1mV (purple bars).

B. Potential Design improvements

The detailed analysis of the behavior of this chip reveals that there is still some room for improvement in neuron design to provide a faster system with a less noisy neuron state. There are two areas in this design where improvements can be made with additional research, the digital low pass filter and the input comparator. For the digital low pass filter, the results of measuring this system indicate that it may be possible that a better filter could be used to more aggressively filter the high frequency noise seen in the neuron state. There is also a potential to improve the filtering by including more complex logic in the phase detector so that it is insensitive to high frequency changes. A more complex filter, while requiring more area, may allow the network to settle accurately in fewer cycles.

For the input comparator, the offset voltage value of 1mV was selected to be very conservative in case offset voltage was the limiting factor in the efficacy of the design. However, with many work vectors the swing on the input of a neuron will be much larger than 1 mV, as this is a corner case where the consensus is very weak among the neighbors. With further research, it may be possible to avoid these situations with careful weight training, or feedback-based training. This would allow for a more relaxed input offset voltage specification for the comparator. A simpler comparator design would save neuron area, and more importantly significant design and calibration time.

V. CONCLUSIONS

We have designed and fabricated a PLL-free ONN in 28nm CMOS based on the lessons learned from PLL-based ONNs and their difficulty to implement. Combining a PLL-free neuron design, which is smaller can be optimized using digital synthesis tools, with the SES technique, we were able to design a larger network containing 100 neurons and 10,000 synapses. The neurons in the network are more area and power efficient than those in the PLL-based ONN, and more importantly the overall system can perform pattern recovery successfully.

This chip demonstrates that the phase-based paradigm is useful for efficient neural networks in hardware that can take advantage of resistive memory for synaptic networks. By continuing to focus on primarily digital neurons, it is possible to take advantage of digital signal processing techniques to even further improve the performance of a phase-based ONN. This neural paradigm provides a path forward for building neurons that can scale along with emerging synapses to build efficient ONNs in hardware.

In conclusion, PLL-free ONNs are a promising architecture for large scale implementations that utilizes the benefits of scalable non-volatile RRAM technology. Overall, the use of emerging memory technology and mixed signal techniques provides a promising path forward for deeply scaled, efficient neural networks.

REFERENCES

- [1] Merolla, Paul, John Arthur, Filipp Akopyan, Nabil Imam, Rajit Manohar, and Dharmendra S. Modha. "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm." In Custom Integrated Circuits Conference (CICC), 2011 IEEE, pp. 1-4. IEEE, 2011.
- [2] Ng, Joe Yue-Hei, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, pp. 4694-4702. IEEE, 2015.
- [3] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." In Proceedings of the 25th international conference on Machine learning, pp. 160-167. ACM, 2008.
- [4] Prezioso, Mirko, Farnood Merrikh-Bayat, B. D. Hoskins, G. C. Adam, Konstantin K. Likharev, and Dmitri B. Strukov. "Training and operation of an integrated neuromorphic network based on metal-oxide memristors." *Nature* 521, no. 7550 (2015): 61.
- [5] Hoppensteadt, Frank C., and Eugene M. Izhikevich. "Pattern recognition via synchronization in phase-locked loop neural networks." *IEEE Transactions on Neural Networks* 11, no. 3 (2000): 734-738.

- [6] Dunning, Jim, Gerald Garcia, Jim Lundberg, and Ed Nuckolls. "An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors." *IEEE Journal of Solid-State Circuits*, vol. 30, no. 4, pp. 412-422, Apr 1995.
- [7] Shi, Rongye, Thomas C. Jackson, Brian Swenson, Soumya Kar, and Lawrence Pileggi. "On the design of phase locked loop oscillatory neural networks: mitigation of transmission delay effects." In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 2039-2046. IEEE, 2016.
- [8] Razavi, Behzad. *RF microelectronics*. Vol. 2. New Jersey: Prentice Hall, 1998.
- [9] Jackson, Thomas C., Rongye Shi, Abhishek A. Sharma, James A. Bain, Jeffrey A. Weldon, and Lawrence Pileggi. "Implementing delay insensitive oscillatory neural networks using CMOS and emerging technology." *Analog Integrated Circuits and Signal Processing* 89, no. 3 (2016): 619-629.
- [10] Keskin, Gokce, Jonathan Proesel, and Larry Pileggi. "Statistical modeling and post manufacturing configuration for scaled analog CMOS." In *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, pp. 1-4. IEEE, 2010.
- [11] Merolla, Paul A., John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson et al. "A million spiking-neuron integrated circuit with a scalable communication network and interface." *Science* 345, no. 6197 (2014): 668-6