Analog Integr Circ Sig Process (2016) 89:619–629
DOI 10.1007/s10470-016-0803-4

Implementing delay insensitive oscillatory neural networks using CMOS and emerging technology

Thomas C. Jackson[1] · Rongye Shi[1] · Abhishek A. Sharma[1] · James A. Bain[1] · Jeffrey A. Weldon[1] · Lawrence Pileggi[1]

Received: 8 April 2016 / Revised: 28 June 2016 / Accepted: 14 July 2016 / Published online: 21 July 2016

Abstract One major challenge in efficiently implementing neuromorphic networks is the need for a large number of variable synaptic connections. Networks that use emerging resistive memories as synapses have been proposed to tackle this problem, but interfacing with these devices is still inefficient in deeply-scaled CMOS. Oscillatory Neural Networks (ONNs) use a different paradigm than most analog hardware implementations, and may be able to interface more efficiently with RRAM neurons. Previous work on ONNs, however, has not considered the effects of actual hardware implementation realities, such as delay in the network. In this work, the first reported IC implementation of an oscillatory neural network is designed and fabricated. Modifications are made to the ONN architecture based on theoretical analysis to allow for proper operation in real-world conditions. One modification is changing the PLL-type, giving the system a different dynamic trajectory which is robust to global delays. Additionally, circuitry is added to control the transport delay of the neuron output signals. A chip with the modified ONN architecture is designed and tested in 28 nm CMOS and estimated power and area figures are reported.

Keywords Neuromorphic computing · Oscillatory neural network · RRAM crossbar synapses

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA, and the National Science Foundation under Contract CCF1318160.

✉ Thomas C. Jackson
thomasjackson@cmu.edu

[1] Carnegie Mellon University, 5000 Forbes Avenue, 2135 Hamerschlag Hall, Pittsburgh, PA, USA

1 Introduction

For many years, neuromorphic networks have been proposed as a way to build efficient systems that can perform a variety of tasks, from classification, to audio and image processing. These networks have shown massive success in software implementation, and are found in many products today [3, 6]. When considering hardware acceleration of these systems, however, one large challenge is implementing the amount of connectivity required. This connectivity requirement is one of the main limiting factors keeping hardware systems from being as power and area efficient as the brain. Ideally, such connections are implemented with variable weights, but efficiently building a large number of variable weights has proven challenging with conventional technology.

One solution to this problem that has been proposed is the use of emerging resistive memories to act as the connection between devices, and this has been partially demonstrated in works such as [9]. Resistive memories, briefly explained in Sect. 3.2, are capable of storing an analog value in a very compact area. Significant strides have been made to enable crossbar implementation of these devices, and they are being successfully used in digital memories. Unfortunately, there have not yet been any fully-integrated hardware neural networks that use RRAM crossbars as analog connection elements. This is primarily because making analog measurements of the RRAM memory devices in deeply-scaled CMOS is challenging. If voltage and current are the state variables, the reduced dynamic range and increased variability in small devices must be corrected through expenditure of additional power and area. Instead, this work proposes using RRAM crossbars in a different neural networking paradigm to extract the information contained in them efficiently.

Specifically, this work considers recurrent neural networks that use phase to compute, rather than current or voltage. These networks are called oscillatory neural networks (ONNs) and were initially proposed in [4]. In an ONN, each neuron contains an oscillator, and the phase of that oscillator represents the state of the neuron. These oscillatory outputs are combined through a synaptic network, and each neuron settles to a value that is the consensus of its neighbors. The RRAM crossbar array scales the amplitude of the neuron outputs according to their analog value. When summing the neuron outputs, the larger amplitude signals contribute more strongly to the overall phase. By using time to represent analog values, the problem of reduced supply voltage can be avoided, and mixed signal techniques can be used to reduce error due to variation with minimal power and area overhead. Therefore, by using ONNs, robust and efficient neural systems with RRAM crossbar synapses are feasible in deeply scaled technologies.

With the resistive crossbar providing the memory of the system, the rest of the components in the network can be designed in CMOS. When designing a CMOS system, it is essential to consider effects such as process variation and transport delay of signals. Section 4 discusses the impact of variation and delay on ONNs, and section 5 details circuitry developed to account of this impact. The paper culminates in a presentation of the simulation results of a chip designed in 28 nm CMOS, and a discussion of the scalability of this architecture in future designs.

2 Oscillatory neural networks

Oscillatory neural networks are systems composed of oscillatory neurons that use either frequency or phase of periodic signals to perform computation. The particular ONN being considered in this work is composed of connected phase-locked loops (PLLs), as shown in Fig. 1. This network has a few important features that are essential for efficient and correct operation.

2.1 Frequency synchronization

One critical result from the analysis of these networks is that, given a symmetric weight matrix, all of the PLLs in the network will synchronize to the same frequency, regardless of their initial conditions. A complete mathematical analysis of the system can be found in [4], and the key results will be replicated here as they are essential to understand phenomena that arise when building the system in hardware.

The dynamics of the PLL system can be described by the following dynamical system:

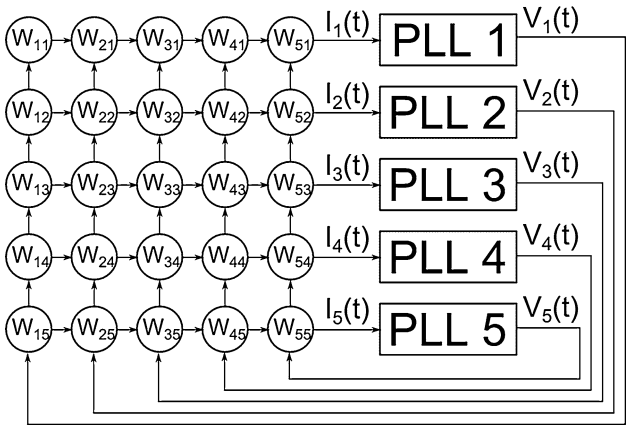


Fig. 1 A conceptual representation of the PLL ONN. The output waveforms $V_i(t)$ are passed through the synaptic network of weights, and the PLLs work to match the inputs signals, $I_i(t)$. This figure adapted from [4]

$$\dot{\phi}_i = V(\Omega t + \phi_i) \sum_{j=1}^{n} w_{ij} V(\Omega t + \phi_j - \frac{\pi}{2}). \tag{1}$$

In this equation, ϕ_i represents the phase of the output PLL i, while $V(\phi_i)$ represents the voltage output corresponding to a given phase. Additionally, Ω represents the natural frequency of the oscillator in the PLL (assumed to be $>>1$), and w_{ij} represents the connection weight between PLL i and j. The term of $\frac{\pi}{2}$ is a necessary artifact due to using a multiplier as the phase detector in the PLL.

To complete the analysis, this system is averaged in time (over many cycles of the VCO) to get a system in terms of only phases and connection weights:

$$\dot{\phi}_i = \sum_{j=1}^{n} w_{ij} H(\phi_j - \phi_i). \tag{2}$$

The function H is the time averaged product of $V(\Omega t + \phi_i)$ and $V(\Omega t + \phi_j - \frac{\pi}{2})$. If the VCO output is a 2π periodic odd-even function (e.g. square wave or sinusoid), the function $H(\chi)$ is zero at $\chi = \{0, \pm\pi\}$. This means that one equilibrium point of the system is when all oscillators are either in phase with one another, or π out of phase.

Using stability theory, [4] shows that the stable states of an ONN with real symmetric weights have the property $(\phi_i - \phi_j) = \{0, \pm\pi\}$. Since each oscillator has a constant phase offset to the rest of the oscillators, they are also the same frequency.

Furthermore, the precise pattern of phases the PLLs of the system settle to are a function of the weights that connect them, and the initial phases of the oscillators. This is one way that such a system can be used for computation, for example, the aforementioned ONN can be used as an associative memory, as shown in Sect. 6.

2.2 PLL type and hardware implementation

The first important modification proposed and tested in this work is the use of a "Type-II" PLL in the ONN as opposed to the "Type-I" PLL that was initially proposed in [4]. A Type-I PLL, although simpler to analyze and implement, has some inherent issues that make them less appealing to implement in CMOS. First, there is an unavoidable trade-off between loop stability and ripple in the output phase [10]. Both of these deviations from the ideal are not considered in the preceding analysis, and could cause issues in the settling of the system.

Another non-ideality is that the acquisition range of a Type-I PLL is low. This means that if the center frequency of the VCO is too far from the reference signal, the PLL may fail to lock, or may lock to a harmonic of the reference. This is particularly a problem when initializing the proposed system, as the theoretical analysis starts from an initialized point.

The solution to these problems is to design the PLL to lock in both frequency and phase. Doing this requires the addition of an additional integrator, which comes in the form of a phase frequency detector combined with a charge pump, as shown in Fig. 2. This removes the problem of harmonic locking, as well as providing additional parameters that allow the designer to decouple loop stability and phase ripple. We have shown in a previous work that an ONN using Type-II PLL satisfies the same properties as the original Type-I ONN [13].

3 Emerging technology for ONNs

Oscillatory networks are relatively simple to describe mathematically, but when considering hardware implementation there are a few factors that make them difficult to build efficiently. This work specifically considers building synaptic connections and the voltage controlled oscillators (VCOs) with emerging technology. The overall architecture includes several novel features, including the VCOs, as depicted in Fig. 3. The details of this architecture will be further described in Sect. 5.

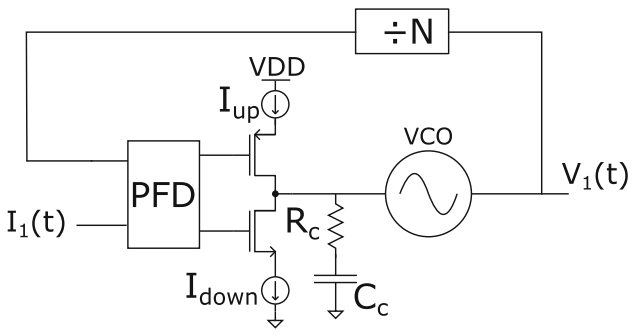


Fig. 2 A Type-II PLL. Given enough time, the output $V_1(t)$ will match the input $I_i(t)$ in frequency and phase. The key difference between this and a Type-I PLL is the Phase-Frequency Detector (PFD) and charge pump between the input and the VCO

3.1 Synaptic connections

The number of connections between the neurons in a fully-connected network scales quadratically with the number of neurons. This polynomial scaling is not a significant impact for a small number of neurons, but would be a significant problem for future systems consisting of thousands if not millions of neurons. This quadratic scaling factor means that the synapses must be implemented as efficiently as possible—taking up comparatively little area and power. To further complicate matters, these synapses should be adjustable post-fabrication to allow for general computing, and to enable learning in the system.

Previous systems in silicon have used transistors or small analog circuits as synaptic devices. Transistors in traditional CMOS, however, cannot store their own state. Therefore, using a transistor as a synapse requires an explicit storage of the analog value of that synapse. To allow for interface with digital circuits, this weight is often stored digitally and converted via a digital to analog converter (DAC). This translates to synaptic hardware that is relatively large, and designers are forced to use techniques such as time multiplexing [12] or weight sharing [7], which introduces additional overhead or limit computational power.

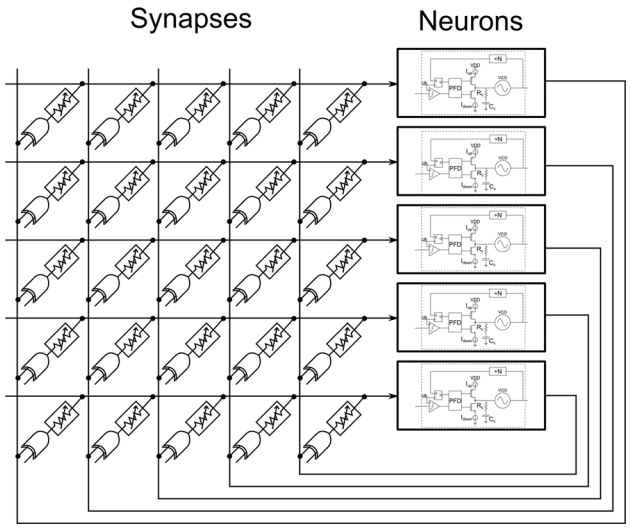


Fig. 3 The proposed ONN architecture. The divided output of each PLL is passed through a synaptic network of variable resistors. These signals are summed on a common node, and fed to the reference input of the PLL. Efficient implementation of these networks is enabled by resistive crossbar memories in the weight array, and low-power nano-oscillators in the VCOs

Alternatively, some recent systems floating-gate transistors as synaptic connections. This solves the problem of needing explicit storage and many DACs, but these devices suffer from a high amount of non-linearity, making them difficult to use in some applications. For example, in the case of [8], multiple devices are needed to make the synaptic impact on the system linear, reducing the scalability of the synapse. In addition, although they are capable of dense integration, manufacturers are already reaching the scaling limits of flash devices, and have started to resort to exotic topologies that require special processing [2].

In this work, we consider using emerging RRAM technology as synapses, since they are capable of dense, monolithic CMOS integration and multi-level storage. There has been recent success in using resistive memories as synapses in neural networks, for example in [1, 9]. In these systems, however, off-chip analog devices are required to read the memory devices, and therefore systems must either be small [9] or time-multiplexed [1], which reduces overall power efficiency. For truly efficient systems, RRAM devices should interface with the CMOS used to read them on-chip.

The specific devices presented here are composed of a metal oxide between two metal electrodes. They are bipolar devices, and depending on the polarity of the voltage applied to them their resistance either increases or decreases. Plots showing an example IV curve from such a device, and showing the multi-level capability of these devices, are shown in Fig. 4. More details regarding these devices and their operation can be found in our earlier work [5]. The key concept is that they behave as non-volatile, variable resistors that are capable of dense CMOS integration.

3.2 Nano-oscillator based VCOs

The same materials used to make RRAM memory devices can be used to design efficient VCOs that can also be used in oscillatory neural networks. One such device is shown in Fig. 5 along with its output waveforms. By including a transistor in series with a transition metal-oxide device, a low-power relaxation oscillator can be built with a high degree of tunability. The detailed operation of these nano-oscillators is given in [5].

These nano-oscillators can scale to tens of nanometers, and are capable of low power operation over a large operating range. Although VCOs are not difficult to build in CMOS, it is challenging to build a VCO with as small a form factor and power budget while still operating at high frequency. Given the correct CMOS design, these devices, along with resistive crossbar synapses, will enable highly efficient oscillatory networks.

4 Desynchronization of ONNs in hardware

To effectively use the emerging technologies described in the previous section, CMOS circuitry must be designed to interface with it. As part of implementing ONNs in a physical system, it is essential to consider physical effects that may change the behavior of the system. In particular, the existence of delay in the feedback of the system may affect its stability and operation. In the case of the PLL ONN, delay in the system causes PLLs to settle at different frequencies.

4.1 Theoretical basis of desynchronization

As noted in Sect. 2, one of the key attributes of the PLL ONN is that the oscillators naturally synchronize to the same frequency. Synchronization allows for comparison of phase, as comparing phases between two signals of different frequency is meaningless. Unfortunately, the system proposed in [4] becomes desynchronized in the presence of delay in the network. A theoretical analysis of this desynchronization effect is provided in [13], but the key result is shown here.

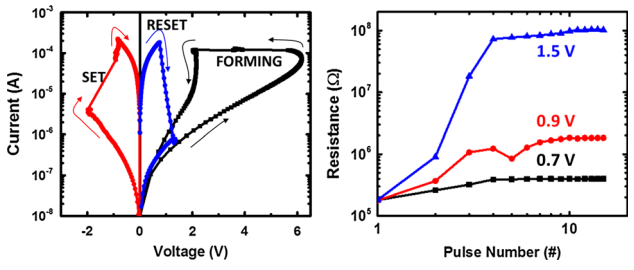


Fig. 4 (*Left*) An IV curve of a typical RRAM device. After a one-time forming process (shown in *black*), the device changes resistance when a threshold is exceeded in one direction or the other. (*Right*) An example of how devices can be consistently programmed to multiple resistance values by using multiple pulses of the same amplitude

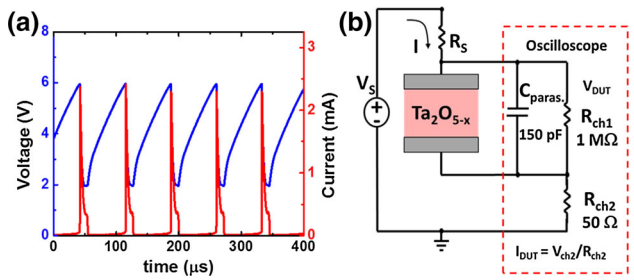


Fig. 5 (*Left*) The voltage and current outputs of the RRAM VCO as a function of time. (*Right*) A schematic showing an RRAM nano-oscillator. By replacing the resistor with a transistor, the device can have variable output frequency with voltage

For analysis, instead of introducing delay at every point in the system, the delay is lumped at the input to each PLL. This means that the input to the ith PLL is taken to be $I_i(t + \delta\phi_i)$, where the $\delta\phi_i$ term represents some amount of delay accrued around the loop. Our analysis in [13] shows that, in this case, the dynamics of the system at the stored pattern points becomes

$$\dot{\phi}_i = \sum_{j=1}^{n} c_{ij} H(\delta\phi_i). \tag{3}$$

The important takeaway from this equation is that, in the originally proposed architecture described in [4], a non-zero $\delta\phi_i$ means a non-zero $\dot{\phi}_i$. This indicates that the system does not synchronize to a constant frequency at the predicted pattern points, since $\delta\phi_i$ is a function of both random variation, and synaptic weight pattern. Furthermore, this phenomenon occurs in systems with both Type-I and Type-II PLLs. This non-synchronization of the ONN can be observed in transistor level SPICE simulations, as shown below.

4.2 Desynchronization simulations

Although desynchronization was shown as a possibility in theoretical analysis, little insight is given in into the magnitude of the desynchronization in a real architecture. An ONN was designed in the 28 nm Samsung PDK using Type-II PLLs and a resistive crossbar consisting of silicon resistors. These components are detailed in Sect. 5 along with the corrections to the architecture. The network consists of 20 PLL neurons, fully connected. The weight pattern in the network was designed to act as an associative memories, storing three different binary patterns shown in Fig. 6.

The system is tested by initializing it to a distorted pattern (see Fig. 6). In this case, the network should settle to the closest stored figure, '4'. When testing the unmodified architecture, however, the system does not successfully recover the pattern, and the phase relationships between the neurons do not stabilize. The results from the unmodified architecture test are shown in Fig. 7. Instead of synchronizing in frequency (and therefore having a constant phase relationship), some of the neurons fail to synchronize, continually accruing negative phase relative to the reference neuron.

An interesting fact to note is that the desynchronization occurs at a different time scale than the phase shift induced by the correct network operation. This is highlighted in Fig. 7 with a blue circle, which identifies the two neurons that are supposed to change from 180° to 0° (or equivalently, 360°) change faster than the rest of the system. Unfortunately, the process of frequency

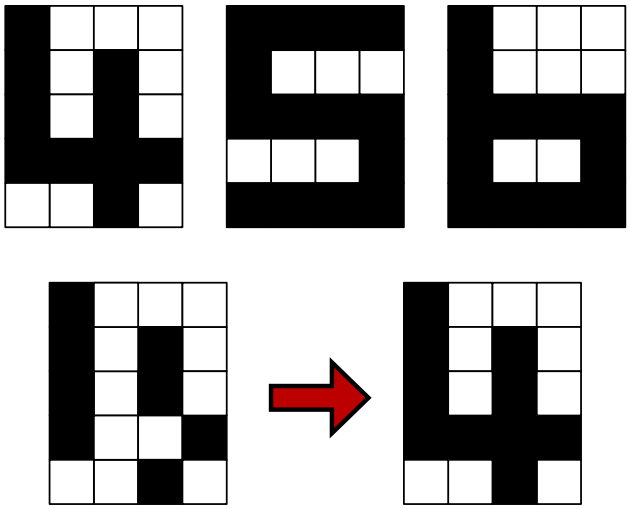


Fig. 6 (*Top*) The three patterns stored in the weights of the ONN. Given an input that is not one of these patterns, the system should settle to the pattern closest to the input. (*Bottom*) The test input/output pair

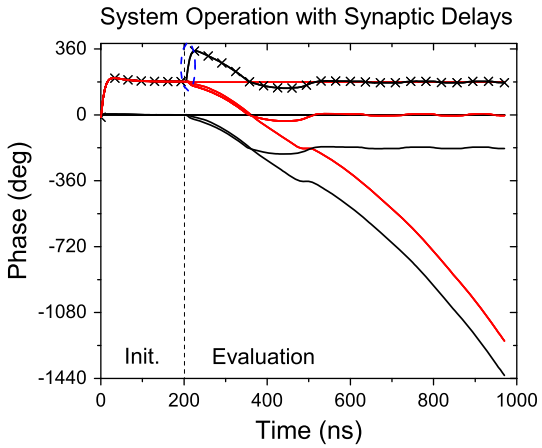


Fig. 7 The phase of the neuron outputs as a function of time. Before 200 ns, the PLLs are initialized to either 0° or 180° phase. The *red lines* represent neurons that should settle to 180°, while the *black lines* represent neurons that should settle to 0°. The *lines* with the '*x*' marked on them are the two neurons that should flip from 180° to 0° in correct operation (they do not in this test, due to desynchronization). The *blue circle* highlights the "high speed" transition associated with correct operation (Color figure online)

desynchronization happens on a similar timescale, so the correct solution is quickly lost. Another important fact gleaned from the original system is that the neurons tend to synchronize in groups. Although there is no global synchronization, the neurons in this example separate into two distinct groups. These groups correlate to the delay caused by the weight matrix. Specifically, neurons that experience the same delay due to the weight matrix end up in the same group. This is an important fact that inspired the corrected design.

5 Corrected ONN architecture

The key to building a working IC version of this ONN is correcting the frequency desynchronization phenomenon. Through the theoretical analysis provided in [13], it was discovered that the PLLs synchronize even in the face of delay as long as the delay experienced by each neuron is identical. This is the basis for the solution in hardware. By ensuring the delay from output to input of each PLL is identical, the system synchronizes successfully.

5.1 Re-timing for synchronization

To ensure each neuron sees the same delay, the signals into each PLL are delayed by a constant amount larger than the largest intrinsic system delay. This is shown in Fig. 8. As long as the intrinsic delay of each path is less than the delay introduced by re-timing, the system works correctly. This method does introduce some quantization error, but this can be minimized by careful design of the re-timing delay to be only slightly larger than the largest delay in the system.

This re-timing is implemented through the use of a clocked comparator (see the full neuron schematic in figure 9). The specific architecture used provides both clocking and significant insensitivity to variation, while consuming very little static power. Details of this latch design, referred to as a StrongARM latch, can be found in [11]. Since there is no external reference for the system while it is in evaluation mode, the comparators are all clocked from the VCO of one of the neurons. The VCO output is sent through a frequency divider before it becomes the neuron output, therefore the comparator clock runs significantly faster than the neuron outputs, reducing quantization. Since the PLL detects phase rather than multiplying signals, information is not lost when the summed input passes through a comparator (the zero crossings are preserved).

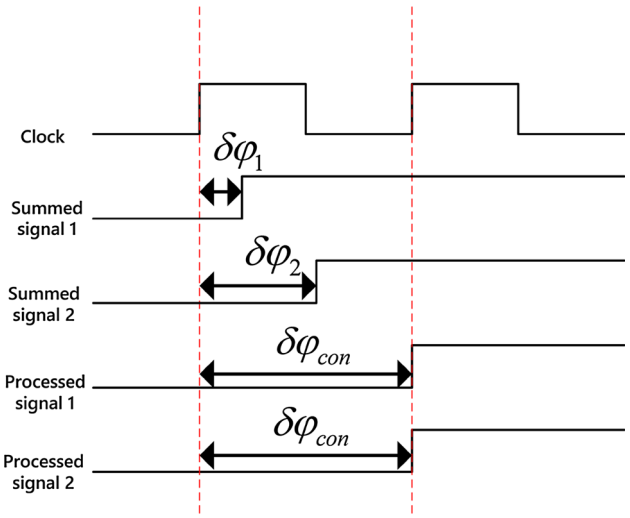


Fig. 8 The re-timing technique used to ensure synchronization of the neurons in the system. The value ϕ_{con} must be designed to be larger than the intrinsic delay in the loop. This method causes all neurons to see the same delay at their input. This figure adapted from [13]

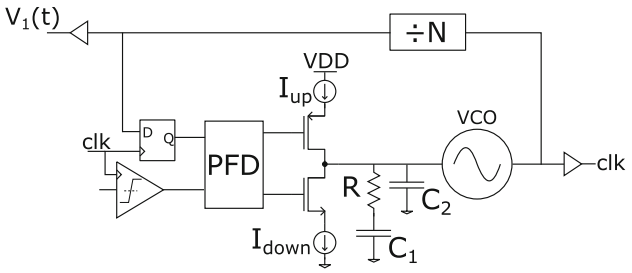


Fig. 9 The schematic of a neuron in the ONN. It is a Type-II PLL with additional circuitry to ensure the system will synchronize. The output of the neuron is taken after the divider, and is buffered before being sent to the synapse network. This figure represents the reference neuron, whose VCO output is used as the clock for the re-timing of the entire network

5.2 PLL-based neuron

The most complex component of the system is the PLL neuron, and it is shown in Fig. 9. The neuron consists of a Type II PLL, with additional circuitry to get the neurons to synchronize correctly.

The summed signal from the synaptic network enters at the left. This signal is passed through a clocked comparator into the phase frequency detector (PFD), which outputs pulses to the charge pump proportional to the phase difference between the two signals. The PFD is shown in Fig. 10, and consists of standard digital circuits.

The charge pump uses current pulses to change the voltage on the VCO input. To ensure stability of the PLL, it is necessary to filter this input with a low-pass network. This prototype uses a silicon resistor and gate capacitors to build the analog filter. The use of gate capacitance causes some nonlinearity in the filter, but this was not found to impact the overall operation of the system, since these PLLs do not need to be extremely accurate.

The VCO used in this design is a five stage, current-starved inverter-based ring oscillator. It was designed to have a nominal center frequency of 1 GHz, and sufficient gain for high parametric yield at this frequency. The ring oscillator was chosen due to its relatively small area and efficient implementation in any CMOS process. Furthermore, its frequency and input range were selected to be comparable to fabricated RRAM nano-oscillators for a proof-of-concept design.

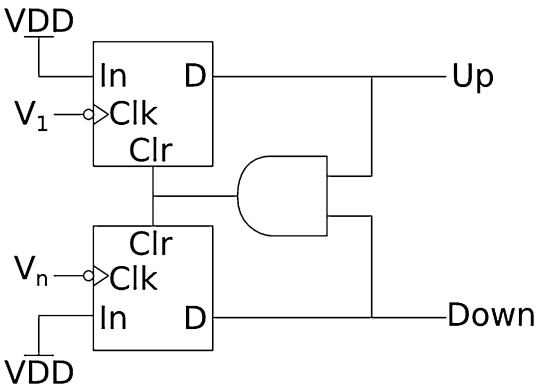


Fig. 10 The phase-frequency detector. When V_1 (the feedback signal) arrives before V_n (the input signal) the VCO frequency is slightly increased, advancing the phase of the PLL output signal. The amount of increase is proportional to the difference in phase between the two signals. The opposite occurs if V_n arrives first, delaying the VCO output

The VCO output is passed through a frequency divider, and the output of the divider is fed back into the PFD. The divider allows the VCO output to be used as the re-timing clock. Furthermore, by reducing the frequency of the information-carrying waveform, the signals passed through the synaptic network are lower frequency and therefore less impacted by skew. Before the feedback signal is input to the PFD, it passes through a flip-flop clocked by the re-timing signal used by the comparator. This ensures the inner loop of the PLL sees the same amount of delay as the reference signal, preventing the VCO from rapidly running to its limit.

The neuron shown in Fig. 9 is the "reference" neuron. This is the neuron that all the other phases are measured against, and it further provides the re-timing clock for the rest of the network. The clock is to all of the flip-flops and comparators via balanced clock tree routing.

5.3 Resistive synaptic crossbar

The resistive crossbar is shown in Fig. 3. The output of each PLL is fed to each synapse, which consists of an XOR gate and a resistive memory element. The XOR gate is required to enable negative weights in the system. With resistive devices, only positive weights are possible, the XOR can shift the digital signal by 180°, which is equivalent to negating in the phase domain.

As noted in Sect. 3, the network is designed with resistive RRAM elements in mind. Since, however, this technology is still under development, the prototype in this work uses silicon resistors to emulate RRAM devices. The synapses used are shown in Fig. 11, and consist of three binary-weighted silicon resistors that are switched via transmission gates. The binary weighting allows for 8 possible weights, plus sign. Although RRAM devices are capable of arbitrary analog storage, storing a precise value can be difficult, making a quantized neuron an appropriate emulation.

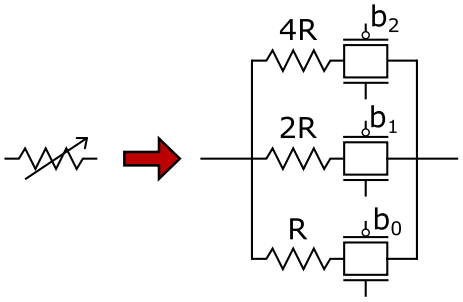


Fig. 11 The CMOS approximation of a multi-level resistive device. Binary-weighted silicon resistors are used to provide eight linearly-spaced conductance values

5.4 Digital interface

For most applications, neural networks will likely be used in conjunction with traditional computing architectures to solve problems that are more efficiently tackled in the massively parallel domain. As a result, these neural co-processors will need to interface with traditional digital processors, meaning that they will need to support digital inputs and outputs. This prototype supports both. The inputs to the network are the weights and the initial condition of the neurons. The weights are binary, as discussed in the previous section. The initial conditions are similarly chosen in a digital manner.

To initialize the system, an off-chip reference is used to generate a high-speed clock signal. This signal is then divided, and multiple phases of the divided signal are isolated and distributed to each neuron. For this proof of concept, four initialization phases are possible (0°, 90°, 180°, and 270°). A digital multiplexer is used in each neuron to select the desired initial phase.

A digital output interface is possible due to the dynamical properties of the ONN discussed in Sect. 2. Each neuron settles to be either 0° or 180° relative to one another. Therefore, the outputs are measured by comparing each neuron with the reference neuron through an XOR gate. If the signals are in phase, the output remains low, while if they are 180° out of phase it remains high. This provides an efficient digital interface to the output of the network.

6 Hardware ONN simulation results

The network described in Sect. 5 was implemented in the Samsung 28 nm process. The simulations confirm the network operates as expected, all of the neurons synchronizing and their phases taking on the desired stored pattern.

6.1 Pattern recovery example

To demonstrate the operation of the system, the three patterns shown in Fig. 6 were stored in the network using the Hebbian learning algorithm. The weights and initialization of the network are input digitally via scan chain, and the system is initialized using an input reference signal to synchronize the PLLs. This initialization is done for 200 ns before the reference input is disconnected and the neurons are all connected to one another. The results of this simulation are shown in Fig. 12.

With the addition of the re-timing circuitry, the system synchronizes and the neurons all settle to stable phase relationships as originally predicted by the theoretical analysis of the PLL ONN. The settling happens over the course of a few nanoseconds, which corresponds to tens of cycles of the neuron outputs, given the frequency of the system operation. The PLLs all show a critically-damped behavior in the time response of their phase, which is the ideal point for trade-off between stability and speed.

6.2 System power and area cost

This neural system was designed in Samsung 28 nm process. The relative sizes of the components are illustrated by the layouts shown in Figs. 14, 15, 16.

Looking at the first layout, it is apparent that, even with 20 neurons, the synapses take up significantly more area in the network than any other component. This is the main reason for looking to alternative technologies for synapse implementation - as the network scales up the number of synapses scales quadratically. Most of the synaptic area is consumed by the memory for the transmission gates, and by the resistors. The XOR is only a small portion of the digital circuitry shown. Therefore, moving to an RRAM technology would significantly reduce the area of the

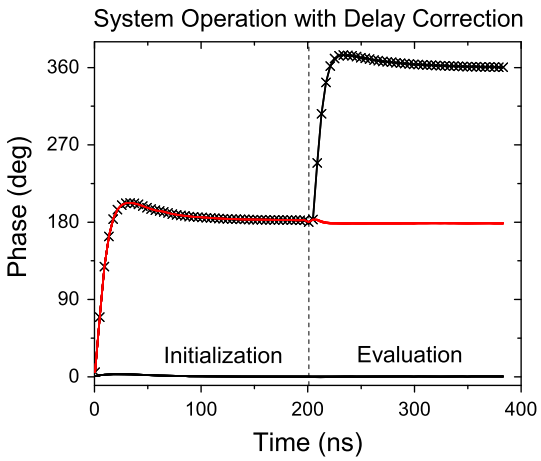


Fig. 12 The phase of each neuron in the corrected architecture. The *red lines* represent neurons that should settle to 180° while the *black lines* represent neurons that should settle to 0°. The marked lines are the two neurons that will change from 180° to 0° during evaluation. These lines settle to 360°, which is equivalent to 0° in phase (Color figure online)

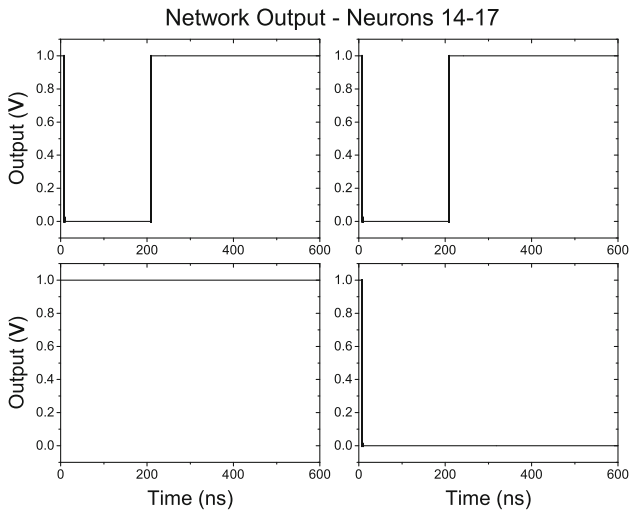


Fig. 13 The digital output of four of the neurons in the network. The *top* two *plots* are the neurons that change from being out of phase to in-phase with the reference neuron. All neurons start with an output of "1" since all of the oscillators are initially in-phase. The neurons are initialized between zero and 200 ns, when the evaluation begins. At the evaluation, only the neurons that are incorrect flip in value

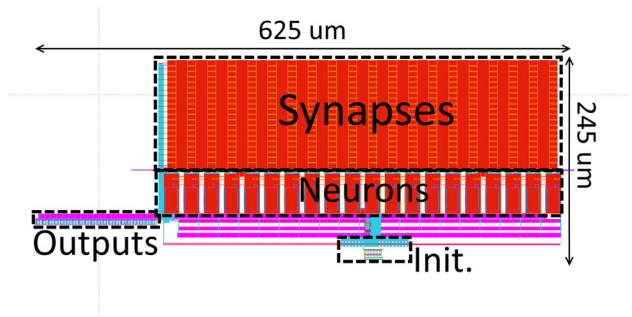


Fig. 14 Layout of the full neuromorphic network

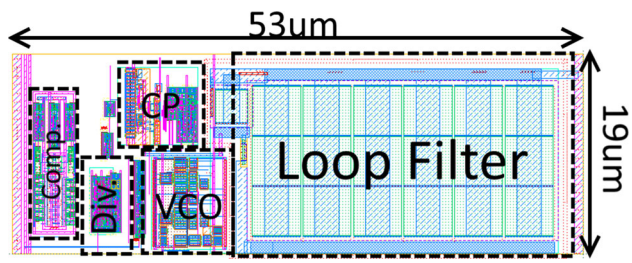


Fig. 15 Layout of one neuron, showing the loop filter, voltage controlled oscillator, charge pump, frequency divider, and clocked comparator

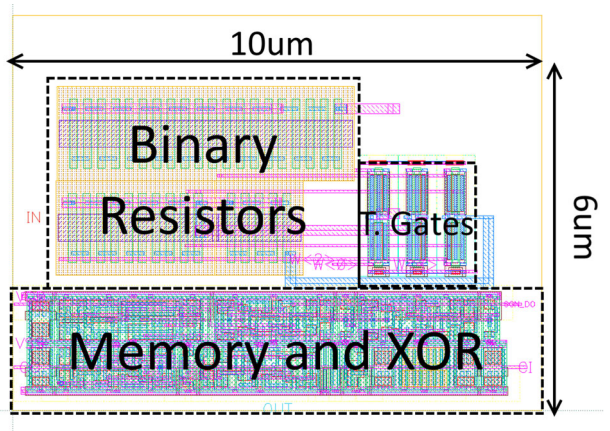


Fig. 16 Layout of one synapse. The use of emerging RRAM technology could replace the resistors, transmission gate, and memory with a single device scalable to 10 nm

synapse, as the memory, transmission gates, and resistors would all be replaced by a single device scalable to 10 nm.

The area of the neuron in this design is dominated by the loop filter. The primary reason for this is the size of the capacitor needed to stabilize the PLL. At the target frequency, this capacitor is about 7.2 pF, a very large capacitance for the target technology. Future designs will explore replacing this analog loop filter with a digital filter, as the system is already quantizing at the input with a local clock. The use of a digital loop filter will reduce the size of the neuron, as well as increasing its flexibility and allowing it to be scaled more easily to other technology nodes.

Using RRAM nano-oscillator VCOs helps reduce the power of the system in two ways. First, each neuron in the system uses approximately 226.5 μW of power during evaluation, but 155.9 μW of the power is used in the VCO. Therefore, replacing the VCO with a more efficient nano-oscillator would significantly reduce the power drawn per neuron. Additionally, the RRAM VCO is capable of operating at much lower frequencies than the CMOS VCO. While the CMOS VCO stops working at hundreds of megahertz, the nano-oscillator VCO can operate into the hundred kilohertz range. Since there is very little static power burned in the neuron, reducing the VCO frequency can directly cut power, allowing the system to flexibly trade off power for speed.

6.3 System scalability

This system is designed to scale up in number of neurons and synapses, while being capable of easy scaling to smaller technology nodes. Scaling in both dimensions is necessary for a future system to be large enough to be useful in a real application.

In terms of increasing the number of neurons and synapses, the neurons must be able to drive many synapses effectively. This is known by many as "fan-out" in neural networks. Since the outputs of the neurons are square waves, they are straightforward to buffer with digital circuits accurately—despite containing analog information there is no need for an analog buffer. Therefore, a single neuron can easily be scaled to drive many synapses at the relatively small cost of additional buffering.

A scaling challenge specific to oscillatory networks is skew across a large chip. The farther a signal has to travel, the more its phase information will be distorted. Clock skew is a problem that also exists in traditional synchronous digital design, and therefore many tools exist to analyze and control the skew of a signal as it travels long distances.

To fit even more neurons on a chip, a clear path is to scale the CMOS components to smaller technology nodes. One of the biggest challenges in scaling a mixed-signal system is that often the analog components need to be redesigned, and they do not scale well. Many components of this system are already digital, the only analog components are the charge pump, loop filter, and VCO. Furthermore, as discussed previously, the input to the PLL is already quantized, so for future systems the loop filter and charge pump can also be made digital without a loss in performance. This "mostly-digital" design allows the system to easily benefit from technology scaling.

7 Conclusion

To approach brain-scale neuromorphic computing in power efficiency and connectivity, it will be necessary to use emerging technologies such as resistive memories. These devices are capable of dense integration and analog storage, both key properties for effective synapses. To date, there have been no integrated solutions that can utilize these devices in a scalable fashion. In this work, we propose using oscillatory neural networks to interface with a crossbar array of RRAM devices.

As part of implementing an ONN in an integrated circuit, theoretical analysis was done to expand the original work on PLL ONNs to include delay effects. This work found that some of the theoretical properties are no longer guaranteed in the face of random variation, specifically the property of guaranteed network synchronization. The impact of the desynchronization was observed in transistor-level simulations, and these observations helped inform changes to the design to combat these effects. By changing the PLLs in the system from Type-I to Type-II PLLs, the system synchronizes under a constant global delay. Then, re-timing circuitry is added to change the random delays at the input to each neuron to

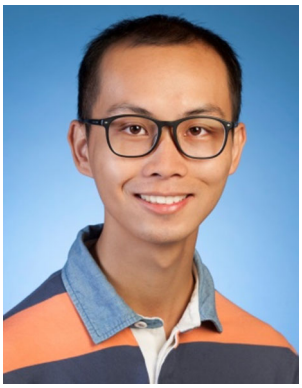
a constant, identical delay, allowing the system to synchronize. A scalable 20 neuron proof-of-concept network was demonstrated in 28 nm CMOS and was shown in simulation to operate correctly.

Acknowledgments Thanks to George Bocchetti and Lily Zhang for assistance in the design and simulation of various PLL components.

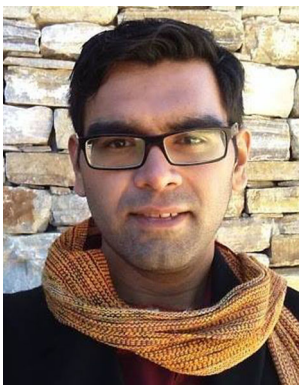
References

1. Burr, G. W., Shelby, R. M., Sidler, S., Di Nolfo, C., Jang, J., Boybat, I., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element. *Electron Devices, IEEE Transactions on*, *62*(11), 3498–3507.
2. Goda, A.: Recent progress and future directions in nand flash scaling. In: Non-Volatile Memory Technology Symposium (NVMTS), 2013 13th, pp. 1–4. IEEE (2013)
3. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1764–1772 (2014)
4. Hoppensteadt, F. C., & Izhikevich, E. M. (2000). Pattern recognition via synchronization in phase-locked loop neural networks. *IEEE Transactions on Neural Networks*, *11*(3), 734–738.
5. Jackson, T. C., Sharma, A. A., Bain, J. A., Weldon, J. A., & Pileggi, L. (2015). Oscillatory neural networks based on TMO nano-oscillators and multi-level RRAM cells. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, *5*(2), 230–241.
6. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1725–1732. IEEE (2014)
7. Linan, G., Espejo, S., Domínguez-Castro, R., & Rodríguez-Vázquez, A. (2002). ACE4k: An analog I/O 64 × 64 visual microprocessor chip with 7-bit analog accuracy. *International Journal of Circuit Theory and Applications*, *30*(2–3), 89–116.
8. Lu, J., Young, S., Arel, I., Holleman, J. (2015). A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 μm CMOS. Solid-state circuits. *IEEE Journal of 50*(1), 270–281
9. Prezioso, M., Merrikh-Bayat, F., Chakrabarti, B., Strukov, D. (2016). RRAM-based hardware implementations of artificial neural networks: progress update and challenges ahead. In: *SPIE OPTO* (pp. 974,918–974,918). International Society for Optics and Photonics
10. Razavi, B. (1998). RF Microelectronics, vol. 1. Prentice Hall New Jersey
11. Razavi, B. (2015). The StrongARM latch [a circuit for all seasons]. *Solid-State Circuits Magazine, IEEE*, *7*(2), 12–17.
12. Schemmel, J., Fieres, J., Meier, K. (2008). Wafer-scale integration of analog neural networks. In: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pp. 431–438. IEEE
13. Shi, R., Jackson, T. C., Swenson, B., Kar, S., Pileggi, L. (2016). On the design of phase locked loop oscillatory neural networks: mitigation of transmission delay effects. In: Neural Networks, 2016. IJCNN 2016. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. IEEE

Thomas C. Jackson is currently pursuing a Ph.D. in Electrical and Computer Engineering at Carnegie Mellon University. He received a Bachelor of Science degree in ECE at Cornell University in 2012, and a Master of Science from Carnegie Mellon University in 2016. His previous work includes electronics systems and sensor design for robotics applications. His current interests are in building neuromorphic and massively parallel systems efficiently by using emerging device technology in conjunction with CMOS circuits. His specific focus has been on the use of Phase Change Materials and Transition Metal Oxide RRAM for the use in analog neural networks and oscillatory neural networks. He was named a Merril Presidential Scholar at Cornell University in 2012, and received the ARCS Fellowship from 2012–2015. He received Best in Session at SRC TECHCON 2014, and the Bardeen Award for Excellence in Nanodevices Research in 2014.



Rongye Shi received the M.S. degree in Electronics and communication Engineering from Peking University, Beijing, China in 2014. He is currently pursuin graduate degree in electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA, USA. His previous work includes atomic clocks, quantum magnetometers and corresponding servo system designs. His current research is in the area of neuromorphic computing system with the focus on theoretical understanding of phase-locked loop based oscillatory neural networks for pattern recognition purpose. Mr. Shi received the Google Excellence Scholarship in 2014 and the John Bardeen Award for Excellence in Nanodevices Research in 2015 SONIC Annual Review Meeting.



Abhishek A. Sharma (M'07) completed his doctoral studies at Electrical & Computer Engineering, Carnegie Mellon University in 2015. He now works at Components Research, Intel Corporation. His doctoral research themes include understanding physical modeling of oxide-based resistive random access memories and using nonlinearities in devices to explore beyond-CMOS functions – specifically using S-type negative differential resistance for oscillators and latches. Previously, he has worked on VLSI process modeling, device modeling, hardware accelerators for image

processing. His interests also include graphical image processing algorithms and neural networks.

James A. Bain received the B.S. degree in material science and engineering from the University of Pennsylvania, Philadelphia and the M.S. and Ph.D. degrees in material science and engineering from Stanford University, Stanford, CA, in 1988, 1991 and 1993, respectively. He is a professor with the Electrical and Computer Engineering Department, Carnegie Mellon University (CMU), Pittsburgh, PA. He also holds a courtesy appointment with the Department of Materials Science and Engineering and is an Associate Director of the Data Storage Systems Center (DSSC), College of Engineering, CMU. Prof. Bain has co-authored more than 225 papers in the field of magnetic, optical, electrical, thermal, and mechanical devices and materials for information storage. He is currently has active research programs in heat assisted magnetic recording (HAMR), and resistive switches for memory (RRAM) and beyond CMOS- architectures; and reconfigurable electronics using phase-change materials. Dr. Bain is a member of the Materials Research Society and a senior member of the IEEE Magnetics Society.

Jeffrey A. Weldon received B.S. degree in engineering physics from the University of California, Berkeley and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 2005. From 2006 to 2010 he was a postdoctoral scholar at the Center for Integrated Nanomechanical Systems. Since 2011 he has been an Assistant Professor in the Electrical and Computer Engineering Department at Carnegie Mellon University. His doctoral research in the area of RF CMOS integrated circuits has been widely adopted by industry and is frequently cited in journals and conferences. He has consulted for a number of semiconductor companies. His current research interests include nanoscale device design in emerging technologies, heterogeneous integration with CMOS and bio-medical devices. He is a member of the ISSCC Student Research Preview committee. Dr. Weldon received the 2001 ISSCC Lewis Winner Award for Outstanding Paper and was the recipient of the 1998 ISSCC Jack Kilby Award for Outstanding Student Paper.

Lawrence Pileggi is the Tanoto professor of electrical and computer engineering at Carnegie Mellon University, and has previously held positions at Westinghouse Research and Development and the University of Texas at Austin. He received his Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University in 1989. He has consulted for various semiconductor and EDA companies, and was co-founder of Fabbrix Inc. (acquired by PDF Solutions) and Extreme DA (acquired by Synopsys). His research interests include various aspects of digital and analog integrated circuit design and design methodologies. He has received various awards, including the 2010 IEEE Circuits and Systems Society Mac Van Valkenburg Award, Inline image 1 the Carnegie Institute of Technology B.R. Teare Teaching Award for 2013, and the 2015 Semiconductor Industry Association (SIA) University Researcher Award. He is a co-author of "Electronic Circuit and System Simulation Methods", McGraw-Hill, 1995 and "IC Interconnect Analysis," Kluwer, 2002. He has published over 300 conference and journal papers and holds 35 U.S. patents. He is a fellow of IEEE.