Cellular Memristive Dynamical Systems (CMDS)

Mohammad Bavandpour∗‡, Hamid Soleimani†, Saeed Bagheri-Shouraki∗ and Arash Ahmadi†
∗Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-11155, Iran
†Department of Electrical Engineering, Razi University, Kermanshah 67149-67346, Iran
‡mbavandpour388@gmail.com
Derek Abbott
School of Electrical and Electronic Engineering, The University of Adelaide, SA 5005, Australia
Leon O. Chua
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770, USA
Received January 22, 2014

This study presents a cellular-based mapping for a special class of dynamical systems for embedding neuron models, by exploiting an efficient memristor crossbar-based circuit for its implementation. The resultant reconfigurable memristive dynamical circuit exhibits various bifurcation phenomena, and responses that are characteristic of dynamical systems. High programmability of the circuit enables it to be applied to real-time applications, learning systems, and analytically indescribable dynamical systems. Moreover, its efficient implementation platform makes it an appropriate choice for on-chip applications and prostheses. We apply this method to the Izhikevich, and FitzHugh–Nagumo neuron models as case studies, and investigate the dynamical behaviors of these circuits.

Keywords: Cellular mapping; memristor; Izhikevich neuron model; FitzHugh–Nagumo neuron model; bifurcation analysis.

1. Introduction

A dynamical system consists of a set of variables indicating its state, and a deterministic law describing the evolution of the state variables with time. In other words, the dynamical laws determine how the state of the system in the next moment of time depends on the inputs, and its state at a previous time-step in terms of mathematical functions. The overall qualitative description of dynamics can be obtained through the study of the phase portrait of the system, which depicts the direction, and velocity of motion in space, and also a geometric representation of certain special trajectories that determine the topological behaviors of all the other trajectories in phase space.

Using dynamical systems, we can model the behavior of a system without knowing all the details that govern the system evolution. This fact is remarkably compatible with the nature of neuroscience, as well as many other sciences, and can be
Digital platforms are used to realize spiking neural networks. There exist three major approaches to this challenge: a) Analog implementations are considered to become a strong choice for direct implementation of neuro-inspired systems [Massoud & Horriuchi, 2001; Basu et al., 2010; Arthur & Boahen, 2011; Aghdasi et al., 2013]. In this approach, nonlinear circuit elements typically determine the nonlinear dynamics, and thus, the variability of the circuit parameters significantly influences the circuit performance. On the other hand, they are comparatively inflexible, and parameter adjustment is difficult to some extent in these circuits. b) Digital platforms are used to realize spiking neurons [Weinstein et al., 2007; Soleimani et al., 2012; La Rosa et al., 2005; Mokhtar et al., 2008; Cassidy et al., 2007]. This approach uses digital computation to emulate individual neural behaviors in parallel, and distributed network architecture to implement a system level dynamic. This approach achieves low development time, high reconfigurability, and immunity to device mismatch — however, due to significantly large computational units such as multipliers, adders, and comparators, it consumes considerably more silicon area, and power. c) Recently, a new sequential-logic-based neuron modeling approach that implements a cellular automaton (CA) in reconfigurable digital hardware has been proposed [Hashimoto & Torikai, 2010; Hishiki & Torikai, 2011; Matsubara & Torikai, 2013]. The CA-based neuron model consists of registers, logic gates, and reconfigurable wires, where the pattern of the wires determines the nonlinear dynamics of the system. Although this approach consumes lower silicon area in comparison with the other digital approaches while achieving high reconfigurability, it suffers from a number of problems. First, the wiring pattern of this structure is a large complicated wiring network that occupies a large switching area, and makes the implementation process cumbersome. Second, the detailed dynamics of the spike shapes such as refractory phase, and exact trajectory of signal, which are among the most important characteristics in biological neural dynamics, are ignored in the first generation of this approach [Hashimoto & Torikai, 2010; Hishiki & Torikai, 2011]. Although this feature can be added to the generalized version through velocity counters [Matsubara & Torikai, 2013], the final approach is complicated, its implementation is cumbersome, and the accuracy of the signal shape strongly depends on the size of the counters, and state registers. Third, one of the most time consuming and complicated aspects of this approach concerns determining the borders of states (i.e. due to numerous parameters), that strongly influence the performance of the circuit.

The implementation constraints strongly limit the power of dynamical systems in modeling, and neuroscientists have been unable to develop an accurate model. Moreover, a number of behaviors such as a special signal shape in the output signal have no elegant analytical description. Our cellular approach presents no limitation on the computational effort of the dynamical functions, and even has the capability of implementing analytically indescribable dynamical systems.

For hardware implementation of our approach, we use a memristor nanoscale crossbar platform to add programmability to the system while obtaining an efficient hardware implementation in terms of area and implementation cost. The memristor is a passive two-terminal device whose resistance changes depending on the polarity, and magnitude of a voltage applied to the device’s terminals, and the duration of this voltage’s application. Its existence was first theoretically predicted by...
The physical implementation of the memristor was realized in late 2008 [Strukov et al., 2008]. Consequently, significant interest in the memristor emerged, and several practical applications have been proposed. A number of applications in which memristors can be used are: resistive memories (RRAM) [Yenpo et al., 2011], synapses in SNN [Serrano-Gotarredona et al., 2013], digital logic [Raja & Mourad, 2009], and so on.

Our hardware approach is a mixed analog-digital memristive system that implements special dynamical systems aimed at emulating neuron models. This approach uses cellular mapping to store the phase plane on the memristor crossbar structure, and shift registers to store the current state of the dynamical system.

The proposed approach is fully configurable due to the memory property of the memristor. It implies that different compatible dynamical systems such as neuron models with different parameter values can be easily programmed on the system. Moreover, in the case of neuron models, the circuit can mimic dynamic behaviors accurately, and produce different biological-like spike shapes at the output.

The rest of this paper is organized as follows: Sec. 2 discusses the dynamical systems, and cellular mapping. The hardware structure of the memristive cellular system is introduced in Sec. 3. Section 4 includes the implementation of the well-known Izhikevich neuron model on the proposed platform, and the bifurcation analysis of the resultant circuit as a case study. A second case study that is an implementation of FitzHugh–Nagumo neuron model on the Cellular Memristive Dynamical System (CMDS) platform is investigated in Sec. 5. Two important remarks about the CMDS are discussed in Sec. 6, and the work is concluded in Sec. 7.

2. Cellular Mapping of Dynamical Systems

A dynamical system is a concept in mathematics where a fixed rule describes the time dependence of a point in a geometrical space. At any given time, a dynamical system has a state given by a set of real numbers (a vector) that can be represented by a point in an appropriate state space (a geometrical manifold). Small changes in the state of the system create small changes in the numbers. The evolution rule of the dynamical system is a fixed rule that describes what future states follow from the current state. The rule is deterministic; in other words, for a given time interval only one future state follows from the current state.

In this section, we transform the dynamical behavior of a system to a cellular concept that is easily implementable on the memristor crossbar structure. The target dynamical system in this study is a two-dimensional system with a number of auxiliary functions. The overall relation of such systems is:

\[
\begin{align*}
\frac{dx}{dt} &= F(x, y) + b \\
\frac{dy}{dt} &= G(x, y) + c
\end{align*}
\]

where $F$ and $G$ are smooth functions, $b$, and $c$ are input variable parameters, and a number of auxiliary functions on $x$, $y$ may be attached. We assume the solution exists for all $t \geq 0$, and is unique when initial data is provided. Typically, explicit solutions to Eq. (1) cannot be found. However, the phase portrait for the system can be achieved without finding the solutions. In this case, the phase plane consists of a 2-D space of $(x, y) \in \mathbb{Z}^2$ determining the velocity, and direction of the moving state point at every point of the space by $(\dot{x}, \dot{y})$ vector.

In the cellular concept, discrete points on the state variables, and accordingly discrete mesh-like points on the phase plane are considered. Thus, for continuous variable $x$:

\[
X(t) = i, \quad i \in \{1, 2, \ldots, M\} \equiv M
\]

where $X$ is the counterpart of $M$ discrete variable values of $x$ achieved as

\[
X(t) = \left\lfloor \frac{M \cdot (x(t) - x_{\min})}{x_{\max} - x_{\min}} \right\rfloor + 1
\]

where $x(t) \in [x_{\min}, x_{\max})$. (3)

Accordingly for a continuous variable $y$:

\[
Y(t) = i, \quad i \in \{1, 2, \ldots, N\} \equiv N
\]

where $Y$ is the counterpart of $N$ discrete variable values of $y$ achieved as

\[
Y(t) = \left\lfloor \frac{N \cdot (y(t) - y_{\min})}{y_{\max} - y_{\min}} \right\rfloor + 1
\]

where $y(t) \in [y_{\min}, y_{\max})$. (5)

Therefore, the discrete phase plane is a $M \times N$ space that consists of discrete velocity and direction values. To evaluate each component of the velocity vector in the cellular space, the amount of...
displacement is constant, and can be achieved as:
\[
\begin{align*}
\Delta x &= \frac{x_{\text{max}} - x_{\text{min}}}{M} \\
\Delta y &= \frac{y_{\text{max}} - y_{\text{min}}}{N}
\end{align*}
\]  
(6)

The velocity vector at a given point \((i, j)\) of the discrete phase plane can be obtained by:
\[
\begin{align*}
V_x^{(i,j)} &= F(x_{\text{min}} + (i-1) \cdot \Delta x) \\
y_{\text{min}} + (j-1) \cdot \Delta y + b \\
V_y^{(i,j)} &= G(x_{\text{min}} + (i-1) \cdot \Delta x) \\
y_{\text{min}} + (j-1) \cdot \Delta y + c
\end{align*}
\]
(7)

where \(i\) is the value of state variable \(X\), \(j\) is the value of state variable \(Y\), \(V_x^{(i,j)}\) is the analog value of velocity for moving from \((i,j)\) to \((i\pm 1,j)\), and \(V_y^{(i,j)}\) is the analog value of velocity for moving from \((i,j)\) to \((i,j\pm 1)\) in the discrete phase plane. Accordingly, the cellular motion times at a given point \((i,j)\) of the discrete phase plane can be obtained by:
\[
\begin{align*}
\Delta t_x^{(i,j)} &= \frac{\Delta x}{V_x^{(i,j)}} \\
\Delta t_y^{(i,j)} &= \frac{\Delta y}{V_y^{(i,j)}}
\end{align*}
\]
(8)

where \(\Delta t_x^{(i,j)}\) is the analog value of required time for moving from \((i,j)\) to \((i\pm 1,j)\), and \(\Delta t_y^{(i,j)}\) is the analog value of required time for moving from \((i,j)\) to \((i,j\pm 1)\) in the discrete phase plane. Note that the variables \(\Delta t_x^{(i,j)}\) and \(\Delta t_y^{(i,j)}\) are the real-valued variables such that their negative value means backward motion.

Clearly, the cellular approach is asynchronous. This implies that the value of the state variables change asynchronously. In this condition, when one state variable is changed, this is critical to correctly handle the timing of change in another state variable in the new location. Here, we clarify the procedure with a special condition, and then generalize it to draw an overall rule to handle the timing.

Assume that the initial state \((t = 0)\) of the system is \((i,j)\), and the corresponding cellular motion times are \(\Delta t_x^{(i,j)}\) and \(\Delta t_y^{(i,j)}\) where \(|\Delta t_x^{(i,j)}| < |\Delta t_y^{(i,j)}|\).

At \(t = |\Delta t_x^{(i,j)}|\) the location is changed to \((i+1,j)\), and the new motion time of variable \(X\) is \(|\Delta t_x^{(i+1,j)}|\), but it is not logical to consider \(|\Delta t_y^{(i+1,j)}|\) as the new motion time of variable \(Y\), and ignore the resting time in point \((i,j)\) with no change in \(Y\). Hence, we consider the elapsed time through subtracting it as a portion of the new motion time:
\[
\Delta t_x^{(i+1,j)} = \frac{1 - \Delta t_x^{(i,j)}}{\Delta t_x^{(i,j)}} \cdot \Delta t_x^{(i+1,j)}.
\]
(9)

According to the above explanation, we can draw an overall recursive rule to handle the motion times for moving from location \(P \in M \times N\) to the neighbor location \(Q \in M \times N\):
\[
\begin{align*}
t_x^P(Q) &= \begin{cases} 
1 \cdot \frac{|\Delta t_x^{(i,j)}|}{|T_x^{(i,j)}|} \cdot |T_x^{(i,j)}| & (X(Q) = X(P)) \\
|T_x^{(i,j)}| & (X(Q) \neq X(P))
\end{cases} \\
t_y^P(Q) &= \begin{cases} 
1 \cdot \frac{|\Delta t_y^{(i,j)}|}{|T_y^{(i,j)}|} \cdot |T_y^{(i,j)}| & (Y(Q) = Y(P)) \\
|T_y^{(i,j)}| & (Y(Q) \neq Y(P))
\end{cases}
\end{align*}
\]
(10)

where \(t_x^P(Q)\), and \(t_y^P(Q)\) are respectively the remaining time to change for variables \(X\) and \(Y\) at the location \(Q\), and are decreased by passing the operation time of the system. The cell change policy is given by:
\[
\begin{align*}
X(t^+) &= \begin{cases} 
X(t) + \text{sign}(T_x^{(i,j)}(Y(t))) & (t_x^P = 0) \\
X(t) & \text{otherwise}
\end{cases} \\
Y(t^+) &= \begin{cases} 
Y(t) + \text{sign}(T_y^{(i,j)}(X(t))) & (t_y^P = 0) \\
Y(t) & \text{otherwise}
\end{cases}
\end{align*}
\]
(12)

Thus, when the remaining time for a given variable is completed, the variable is changed by one step, and the remaining times for all variables are updated according to the new state. In the next sections, it is shown that the dynamical behavior of a given dynamical system described generally by Eq. (1) can be mimicked properly using the velocity buffers \(V_x\) and \(V_y\) given by Eq. (7) (where \(b = c = 0\)), and motion policy given by Eqs. (10)–(13).
3. Cellular Memristive Implementation

In this section, we first briefly introduce the memristor, and then propose a memristor-CMOS structure capable of implementing the cellular dynamical systems introduced in the previous section. This work presents the system-level structure of the circuit to form the basis of this work, and also a devices-level circuit to demonstrate the implementation capability of the proposed approach. The detailed practical device-level analysis of the system considering the detailed circuit-level model of the blocks, side effects, and implementation constraints such as mismatch is out of the present scope, and will be presented in a future publication.

3.1. Memristor

Chua [1971] predicted that a fourth missing element must exist in addition to the three well-known basic circuit elements, namely the resistor, capacitor and inductor. He called this basic element the memristor, a portmanteau of “memory + resistor” because it possesses a variable resistance that changes according to the history of the current passing through it, and remains fixed when the current disappears. The memristor equation creates the missing link between flux and charge as \(\frac{d\phi}{dq} = M(q) \cdot dq\). Charge-controlled memristor characteristics can therefore be described [Chua, 1971] as:

\[
M(q) = \frac{d\phi}{dq} = \frac{d\phi}{dq} = \frac{d\phi}{dq} = \frac{d\phi}{dq}\]  

where \((\tau = M(q) \cdot i)\). Similarly, when the \(\varphi - q\) relationship is flux-controlled, the element is known as a memductance [Chua, 1971]:

\[
W(\varphi) = \frac{dq}{d\varphi} = \frac{d\varphi}{dq} = \frac{d\varphi}{dq} = \frac{d\varphi}{dq} 
\]

where \((I = W(\varphi) \cdot V)\). In the simplest condition, when \(M\) or \(W\) is a constant, the memristor acts as the resistance, and if the \(\varphi - q\) relationship is nonlinear, the device’s behavior is more complex.

Hewlett Packard (HP) presented a memristor [Yang et al., 2008] with a definition based on memristive systems. It contains two different resistance portions that are connected together in serial form: one is high resistance (undoped), and the other is low resistance (doped). The internal structure of the HP device is depicted in Fig. 1(a), and its simple equivalent circuit is shown in Fig. 1(b). The internal state variable, \(w\), is also the variable length of the doped region, and \(D\) is the device length. When \(w \rightarrow 0\), the memristor turns into a high resistance channel, and when \(w \rightarrow D\) it turns into a low-resistance channel. Using [Strukov et al., 2008], the mathematical model for memristive device resistance can be described as:

\[
v(t) = \left(\frac{R_{ON}}{D} \cdot w(t) + R_{OFF} \cdot \left(1 - \frac{w(t)}{D}\right)\right) \cdot i(t).
\]

One of the most important features of the memristor is the threshold condition. It means that small voltages across the nano-device, below its threshold \((V_{th})\), do not induce considerable change in the memristance, while larger voltages induce much greater changes [Jo et al., 2010].

3.2. System-level Structure

Figure 2 shows the overall structure of the cellular dynamical system. In the proposed system, the user determines the dynamical functions, and the desired intervals \([\tau_{min}, \tau_{max}]\), and \([\tau_{min}, \tau_{max}]\) on

![Fig. 1. (a) Memristor device structure containing doped (low resistance), and undoped (high resistance) regions, (b) equivalent circuit model representing the two regions depicted in (a), and (c) circuit symbol for memristor.](image-url)
the input variables \( x \) and \( y \) through a user interface. The cellular velocity matrices \( V_x \) and \( V_y \), excluding \( M \times N \) analog values given by Eq. (7) are calculated in a host computer (where \( b = c = 0 \)), and programmed on the stand-alone target chip through the chip programmer supporting a special communication protocol. In this protocol, all cells of the buffers are selected, and programmed sequentially through a special writing pulse shape. Clearly, the velocity buffers of the chip must be updated every time the parameters of the dynamical system functions \( F(x, y) \) are changed. Then, the circuit starts tracking the state point trajectory using the motion policy given by Eqs. (10)–(13) implemented on the chip. In the memristive chip, \( V_x \) and \( V_y \) buffers are the crossbar memristive analog storages to store the \( V_x \) and \( V_y \) matrices. The XvCO, and YvCO are the Voltage Controlled Oscillators (VCOs) satisfying the timing rules given by Eqs. (10) and (11). The XR and YR are respectively the \( M \) bit and \( N \) bit bidirectional shift registers that determine the location of the state point in the phase plane using the following one-hot coding scheme:

\[
XR(i) = \begin{cases} 
1 & i = 1, 2, \ldots, M \\
0 & \text{otherwise}, 
\end{cases}
\]

\[
YR(i) = \begin{cases} 
1 & i = 1, 2, \ldots, N \\
0 & \text{otherwise}, 
\end{cases}
\]

Moreover, the auxiliary function block is a logical block for implementing the auxiliary functions such as threshold conditions, and resetting functions. There are a number of controlling wires between the blocks of the memristive chip as follows:

BSX, BSY are Buffer Select signals from programmer to the buffers for selecting the timing buffers during the programming phase.

PS is Point Select signal from the programmer to the XR, and YR registers for selecting one state point through setting the state registers during the programming phase.

PP is Programming Pulse from the pulse generator in the chip programmer block to the buffers containing the memristors. This signal produces a special signal shape with specific amplitude and duration according to the analog timing value of the selected location in the phase plane to program this value in the proportional memristor.

AS is the Auxiliary Signal wires between auxiliary function block and state registers to apply the auxiliary function to the dynamical system.

SS is the state point location from state registers to the timing buffers for providing the state change timing for VCOs.

VX, VY are the analog values proportional to the state change timing of \( X \) and \( Y \) variables from velocity buffers to the VCOs.

XC, YC are the shift commands from VCOs to the state registers which determine the timing and direction of the shifts.

\( b, c \) are the input parameters of the dynamical system that directly change the motion speed of the state point.
3.3. **Cellular memristive circuit**

At the architectural level, a crossbar-based structure appears to be the most frequently proposed nanotechnology architecture [Bahar et al., 2007], and compatibility with the crossbar structure is a major factor in memristive circuit design. This type of architecture, with a memristor at each cross point, offers simplicity, flexibility, and scalability, and also provides maximum density. According to the crossbar structure of the cellular phase plane, it is compatible with crossbar structure of the memristor, and it can easily be implemented on this structure. Nonetheless, there are also other options for replacing the memristor crossbar as a memory that we ought to care about — a number of performance parameters including density (that relate to area utilization), cost, speed (both access time, and bandwidth), retention time, and persistence; read/write endurance, active power dissipation, standby power, robustness such as reliability, and temperature related issues characterize memories [Eshraghian et al., 2011]. Recently, a number of memory technologies have emerged such as phase-change random access memory (PCRAM), magnetic RAM (MRAM), ferroelectric RAM (FeRAM), resistive RAM (RRAM), showing promise, and some are already being considered for implementation of memories. Table 1 compares a range of performance parameters, and remarkable features of each of the technologies across memristor-based memories.

As shown in Table 1, the memristor crossbar-based architecture is highly scalable, and shows promise for ultra-high density memories (such as large scale neural networks). For example, a memristor with minimum feature sizes of 10 nm, and 3 nm yield 250 Gb/cm, and 2.5 Tb/cm, respectively. In spite of the high density, zero standby power dissipation, and long life time that have been pointed out for the emerging memory technologies, their long written latency has a large negative impact on memory bandwidth, power consumption, and the general performance of a memory system. Moreover, note that in this paper, we have saved each analog value of the velocities in a memristor through crossbar structure, while this value ought to be discretized in CMOS-based memories into digital bits, despite the aforementioned size efficiency of memristor across other memory technologies, the number of each memory cell is increased more than four times in digital CMOS-based memories, while they do not ever achieve maximum accuracy. So, memristor crossbar structure is the most efficient, and accurate option for saving analog values in our proposed cellular structure.

Figure 3 shows the process of obtaining the cellular phase plane that is storable on the memristor crossbar structure for a given 2-D dynamical system. In this process, two velocity buffers are created that contain the crossbar analog memory cells. These cells store the direction and speed of the state point motion in the corresponding location.

<table>
<thead>
<tr>
<th>Knowledge Level</th>
<th>DRAM</th>
<th>SRAM</th>
<th>NOR</th>
<th>NAND</th>
<th>FeRAM</th>
<th>MRAM</th>
<th>PCRAM</th>
<th>Memristor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Level</td>
<td>mature</td>
<td>advanced</td>
<td>product</td>
<td>advanced</td>
<td>early stage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell Elements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half pitch ((F)) ((\mu)m)</td>
<td>50</td>
<td>65</td>
<td>90</td>
<td>90</td>
<td>180</td>
<td>130</td>
<td>65</td>
<td>3-10</td>
</tr>
<tr>
<td>Smallest cell area ((F^2))</td>
<td>6</td>
<td>140</td>
<td>10</td>
<td>5</td>
<td>22</td>
<td>45</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>Road time ((ms))</td>
<td></td>
<td>&lt; 1</td>
<td>&lt; 0.3</td>
<td>&lt; 10</td>
<td>&lt; 50</td>
<td>&lt; 45</td>
<td>&lt; 20</td>
<td>&lt; 60</td>
</tr>
<tr>
<td>Write/Erase time ((ms))</td>
<td>&lt; 0.5</td>
<td>&lt; 0.3</td>
<td>10^4</td>
<td>10^6</td>
<td>10^4</td>
<td>10^4</td>
<td>10^4</td>
<td>10^4</td>
</tr>
<tr>
<td>Retention time ((ms))</td>
<td></td>
<td></td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
<td>&gt; 10</td>
</tr>
<tr>
<td>Write op. voltage (V)</td>
<td>2.5</td>
<td>1</td>
<td>12</td>
<td>15</td>
<td>0.9-3.3</td>
<td>1.5</td>
<td>1</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>Road op. voltage (V)</td>
<td></td>
<td>1.8</td>
<td>1</td>
<td>2</td>
<td>0.9-3.3</td>
<td>1.5</td>
<td>3</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>Write endurance</td>
<td>10^9</td>
<td>10^14</td>
<td>10^5</td>
<td>10^5</td>
<td>10^2</td>
<td>10^10</td>
<td>10^9</td>
<td>10^9</td>
</tr>
<tr>
<td>Write energy ((J/\text{bit}))</td>
<td>5</td>
<td>0.7</td>
<td>10</td>
<td>10</td>
<td>30</td>
<td>1.5 x 10^5</td>
<td>6 x 10^5</td>
<td>&lt; 50</td>
</tr>
<tr>
<td>Density (Gbit/cm^2)</td>
<td>6.67</td>
<td>0.17</td>
<td>1.23</td>
<td>2.47</td>
<td>0.14</td>
<td>0.13</td>
<td>1.48</td>
<td>250</td>
</tr>
<tr>
<td>Voltage scaling</td>
<td>fairly scalable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highly scalable</td>
<td>major technological barriers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Comparison of traditional and emerging memory technologies.
The minimum speed motion corresponding to the
minimum frequency of the VCO is shown in white
color, and the maximum speed motion correspond-
ing to the maximum frequency of the VCO is shown
in blue or red colors, and the intensity of the colors
is proportional to the speed of the motion. The red
color shows the forward motion or increasing direc-
tion, and the blue color shows the backward motion
or decreasing direction of a variable. Note that in
circuit realization, at running stage, for a long pro-
cessing time, readjustment of the velocity values in
the memristor cells would be necessary during idling
periods. There are three specified samples in Fig. 3
labeled by 1, 2 and 3 to explain how the mapping
is obtained:

- In sample 1 on the phase portrait, the location
  of the state point increases in X vector direc-
tion, and remains constant in Y vector direction.
  Hence, the color of mapped cell is red in the X
  velocity buffer, and white in the Y velocity buffer.

- In sample 2 on the phase portrait, the location
  of the state point is constant in X and Y vector
directions. Therefore, the color of mapped cell is
  white in X and Y velocity buffers. Sample 2 is
  an unstable equilibrium point of the system so
  that a small motion in this location results in
  the divergence of the state point. So, we applied
  a condition to disable the motion even with the
  minimum frequency to mimic this behavior where
  the cell color is very close to the absolute white.

- In sample 3 on the phase portrait, the location
  of the state point decreases in X vector direction,
  and increases in Y vector direction. Accordingly,
  the color of mapped cell is blue in the X velocity
  buffer, and red in the Y velocity buffer.

The detailed internal structure of the mem-
ristic dynamical circuit is shown in Fig. 4. The
BSX and BSY signals switch the circuit between
programming mode (applying programming pulse
to the memristor crossbar, and bypass the \( R_C \))
and run mode (applying VCC to the memris-
tor crossbar). Using the switch transistors on
the crossbar terminals, and according to the one-hot
coding of the state variables, one specific memris-
tor is selected at each moment. In the programming
mode, the programmer applies a signal shape to set
the memristor to the target value. Figure 5 shows
a simple programming signal shape. As shown in
the figure, the programming signal shape contains
a negative rectangular pulse followed by a positive
one. Amplitude of the pulses ought to be greater
than the threshold voltage of the memristor to
change the memristance to the desired value. Neg-
avive pulse width ought to be enough to reset the
memristor to maximum resistance from any initial
resistance. As derived from [Yonpo et al., 2011],
and in the critical case when the initial resistance
of memristor is \( R_{ON} \), the minimum required pulse
width is evaluated by:

\[
T_{min}^N = \frac{V_A \cdot \varphi \cdot D}{|V_A| \cdot R_{OFF}^2 - R_{ON}^2} \quad (19)
\]

where \( V_A \) is the amplitude of pulse, and \( \varphi D = \frac{(c \cdot D)^2}{2 \cdot \mu \cdot (c \cdot D)} \cdot \frac{R_{OFF}}{R_{ON}} \) and \( \mu \) is the mobility coeffi-
cient of the channel. Positive pulse ought to bring
the resistance of the memristor from resetting resistance ($R_{\text{OFF}}$) to desired resistance ($R_{\text{desired}}$) according to the velocity value. Required pulse width for this purpose is given by Yenpo et al. [2011]:

$$T_P = \frac{\varphi D_{\text{in}}}{|V_A| \cdot R_{\text{OFF}}^2 - [R_{\text{OFF}}^2 - R_{\text{desired}}^2]}.$$  \hspace{1cm} (20)

In this circuit, the maximum positive, and negative velocities are respectively represented by $V_{\text{ON}}$ and $R_{\text{OFF}}$ on the memristors. So, the output voltages of the velocity buffers (TX and TY) corresponding to the maximum positive, maximum negative, and minimum velocities are given by:

$$v_{\text{max}}^0 = \frac{R_e}{R_{\text{ON}} + R_e} \cdot V_c$$  \hspace{1cm} (21)

$$v_{\text{max}}^{-1} = \frac{R_e}{R_{\text{OFF}} + R_e} \cdot V_c$$  \hspace{1cm} (22)

$$v_{\text{min}} = \frac{1}{2} \left( v_{\text{max}}^0 + v_{\text{max}}^{-1} \right)$$  \hspace{1cm} (23)

and

$$v_{\text{min}} = \frac{1}{2} \left( v_{\text{max}}^0 + v_{\text{max}}^{-1} \right)$$  \hspace{1cm} (24)

where $R_T = \frac{R_{\text{ON}} + R_{\text{OFF}} + R_e}{(R_{\text{ON}} + R_e) \cdot (R_{\text{OFF}} + R_e)}$. So, the corresponding resistance of the minimum velocity ($v_{\text{min}}$) is derived by:

$$v_{\text{min}} = \frac{R_e}{R_{\text{MIN}} + R_e} \cdot V_c = \frac{R_e \cdot V_c}{R_T}$$  \hspace{1cm} (25)

$$R_{\text{MIN}} = \frac{2}{R_T} - R_e.$$  \hspace{1cm} (26)

Assume that the maximum absolute values of the velocity (where $b = c = 0$) in a given phase plane are $V_{\text{max}}^0$ and $V_{\text{max}}^{-1}$. The corresponding output voltages of the velocity buffers (TX and TY) for a given
velocity of $V_{y}^{(i,j)}$ and $V_{x}^{(i,j)}$ are:

$$
\begin{align*}
V_{x}^{(i,j)} &= \frac{V_{x}^{(i,j)}_{\text{shift}} - v_{\min}}{v_{\max} - v_{\min}} \\
V_{y}^{(i,j)} &= \frac{V_{y}^{(i,j)}_{\text{shift}} - v_{\min}}{v_{\max} - v_{\min}} \\
V_{x}^{(i,j)}_{\text{shift}} &= \frac{V_{x}^{(i,j)}_{\text{pos}} - v_{\min}}{v_{\max} - v_{\min}} + v_{\min} \\
V_{y}^{(i,j)}_{\text{shift}} &= \frac{V_{y}^{(i,j)}_{\text{pos}} - v_{\min}}{v_{\max} - v_{\min}} + v_{\min}
\end{align*}
$$

The corresponding resistances of the memristive velocity buffers are given by:

$$
\begin{align*}
R_{x}^{(i,j)} &= \frac{R_{e}}{R_{e} + R_{\text{shift}}^{(i,j)}} \cdot V_{\text{cc}} \\
R_{y}^{(i,j)} &= \frac{R_{e}}{R_{e} + R_{\text{shift}}^{(i,j)}} \cdot V_{\text{cc}} \\
R_{\text{shift}}^{(i,j)} &= \frac{R_{e} \cdot V_{\text{in}}}{V_{\text{pos}} - v_{\min}} - R_{e} \\
R_{\text{shift}}^{(i,j)} &= \frac{R_{e} \cdot V_{\text{cc}}}{V_{\text{pos}} - v_{\min}} - R_{e}
\end{align*}
$$

According to the mapping of the velocity values to the voltage level values given by Eq. (25), the input parameters $b$ and $c$ ought to be mapped using this equation as a constant part of velocity vector given by Eq. (7):

$$
\begin{align*}
b &= \frac{b \cdot (v_{\max} - v_{\min})}{v_{\max}} + v_{\min} \\
c &= \frac{c \cdot (v_{\max} - v_{\min})}{v_{\max}} + v_{\min}
\end{align*}
$$

Therefore, in the run mode, the analog voltages received by the VCOs are:

$$
\begin{align*}
v_{\text{in}}^{(i,j)} &= \frac{V_{x}^{(i,j)}_{\text{shift}} + hb}{v_{\max}} \\
v_{\text{in}}^{(i,j)} &= \frac{V_{y}^{(i,j)}_{\text{shift}} + cc}{v_{\max}}
\end{align*}
$$

and the functional equation of the VCOs is given by:

$$
\begin{align*}
j_{\text{out}}(t^{+}) &= \begin{cases} 
\text{f}_{\text{max}} & |v_{\text{in}}^{\text{X, Y}} - v_{\min}| > \gamma \\
\beta \cdot |v_{\text{in}}^{\text{X, Y}} - v_{\min}| & \alpha < |v_{\text{in}}^{\text{X, Y}} - v_{\min}| \leq \gamma \\
0 & |v_{\text{in}}^{\text{X, Y}} - v_{\min}| \leq \alpha
\end{cases}
\end{align*}
$$

where $\gamma$ is the threshold value for maximum frequency, $\beta$ is the VCO coefficient determined according to the hardware specification, and the cellular parameters $\Delta x$ and $\Delta y$, $\alpha$ is the threshold value for minimum frequency satisfying the behavior of the system at the equilibrium points. Noted that VCOs inherently satisfy Eqs. (10) and (11). The VCOs produce a clock pulse signal according to the derived equation on the pins $X_{\text{SD}}$ and $Y_{\text{SD}}$. Two other essential signals that determine the direction of shift in the state registers are $X_{\text{SD}}$ and $Y_{\text{SD}}$. The value of zero on these wires means the backward shift, and the value of one means the forward shift in the state registers:

$$
X_{\text{SD}} = \begin{cases} 
1 & v_{\text{in}}^{\text{X, Y}} > v_{\min} \\
0 & \text{otherwise}
\end{cases}
$$

$$
Y_{\text{SD}} = \begin{cases} 
1 & v_{\text{in}}^{\text{X, Y}} > v_{\min} \\
0 & \text{otherwise}
\end{cases}
$$

Finally, the state registers are changed at the positive edge of the $X_{\text{SD}}$ and $Y_{\text{SD}}$ signals as:

$$
\begin{align*}
X(t+^{\text{posedge}}) &= \begin{cases} 
X(t) + 1 & X(t) < M, \ X_{\text{SD}} = 1 \\
X(t) - 1 & X(t) > 1, \ X_{\text{SD}} = 0
\end{cases} \\
Y(t+^{\text{posedge}}) &= \begin{cases} 
Y(t) + 1 & Y(t) < N, \ Y_{\text{SD}} = 1 \\
Y(t) - 1 & Y(t) > 1, \ Y_{\text{SD}} = 0
\end{cases}
\end{align*}
$$
where $t_{\text{reset}}$ is the time of appearance of a positive edge on the input clock pulse of the state registers. The multiplexers shown in Fig. 4 allow the programmer to take control of the state registers by the PS signals to select the cells, and write into them in programming mode, and return it to the VCO units to track the state point in the run mode.

4. Cellular Memristive Izhikevich Neuron Model

This section investigates the cellular implementation of the well-known dynamical system proposed by Izhikevich that describes biological neuron responses. The bifurcation analysis of the cellular approach shows that it can properly mimic the bifurcation scenarios of the original model.

4.1. Cellular Izhikevich neuron model

Izhikevich [2003] introduced one of the widely accepted neuron models, which can reproduce a variety of neuron firing patterns. This model is claimed to be one of the simplest possible 2-D models that can exhibit all the firing patterns. This neuron model has been commonly accepted as an accurate, and computationally affordable model yet producing a wide range of cortical pulse coding behaviors. This model contains two coupled differential equations as:

\[
\begin{align*}
\dot{v} &= 0.04v^2 + 5v + 140 - u + I \\
\dot{u} &= a'(v' - u)
\end{align*}
\]  

(34)

with the auxiliary after-spike resetting equations:

\[
\text{If } v > v_{th} \text{ then } \begin{cases} v \leftarrow c' \\ u \leftarrow u + d' \end{cases}
\]

(35)

where $u$ represents the membrane potential of the neuron, and $v$ represents a membrane recovery variable, which accounts for the activation of $\text{K}^+$ ionic currents, and inactivation of $\text{Na}^+$ ionic currents, and it provides negative feedback to $u$, and $a, b, c, d$ are dimensionless parameters. After the spike reaches its apex ($v_{\text{th}}$), the membrane voltage, and the recovery variable are reset according to the equations above. If $v$ skips over $v_{\text{th}}$, it is first reset to $v_{\text{th}}$, and then to $c'$ so that all spikes have equal magnitudes. According to the overall Eq. (1) of dynamical systems, for $x = v$, and $y = u$ this model will be:

\[
\text{If } v > v_{\text{th}} \\
\begin{align*}
F(x, y) &= 0.04x^2 + 5x + 140 - y \\
G(x, y) &= a'(v' - y) \\
\end{align*}
\]

then

\[
\begin{align*}
b &= I \\
c &= 0
\end{align*}
\]

(36)

These equations, from a geometrical viewpoint, represent a parabolic curve, and a line, as depicted in Fig. 6. The crossing points of these curves give the equilibrium points of the system (neuron). Different spiking patterns are produced by changing these crossing points and the threshold action potential. Here, the cellular auxiliary after-spike resetting equation is given by:

![Cellular Memristive Dynamical Systems (CMDs)](image_url)

Fig. 6. Phase portrait, and equilibrium lines of the Izhikevich neuron model described by Eq. (34), and its cellular mapping on the X and Y timing buffers.
\[ X(t^+) = \begin{cases} X(t) & t \neq t_{\text{posedge}} \\ X(t) \equiv \{ (t, Y(t)) \mid (X(t), Y(t)) \in L \} \\ Y(t) = Y(t) \mid (X(t), Y(t)) \in L & t = t_{\text{posedge}} \end{cases} \]

\[ Y(t^+) = \begin{cases} Y(t) + dd' & (X(t), Y(t)) \in L \\ Y(t) \equiv \{ (t, Y(t)) \mid (X(t), Y(t)) \in L \} \\ Y(t) \mid (X(t), Y(t)) \in L & t = t_{\text{posedge}} \end{cases} \]

where \( L \) is the threshold line defined as:

\[ L \equiv \{ (X, Y) \mid X = M, Y \in N \}. \]

Assuming \( x_{\text{max}} = v_{\text{th}} = 30, cc', dd' \) are calculated by:

\[
\begin{align*}
cc' &= \frac{v' - v_{\text{min}}}{\Delta x} + 1 \\
\text{dd}' &= \frac{v' - v_{\text{min}}}{\Delta y} + 1
\end{align*}
\]

Although this is known as one of the most practical, yet accurate, available models, still there are several challenges in realizing the model on the digital or analog circuits. The difficulty of implementation arises from the quadratic part of the model which is shown by the parabolic curve in Fig. 6. Our cellular approach applies no limitation on the computational effort of the velocity functions, and every complex function such as the parabolic curve can be programmed on the memristive analog memory plates. Based on the simulation analysis, in this section, we show that the memristive cellular approach can exhibit various neuron-like responses of the original Izhikevich neuron model. Figure 7 shows time waveforms produced by the 64-bit memristive cellular approach as the various neuron-like responses of the neuron model by applying various input currents \( I \), and transitions from resting state to the spiking states based on the underlying bifurcation scenarios. Clearly, increasing the bit number increases the accuracy of the system to track the bifurcation scenarios, and produce the output signal shapes. Figure 8 shows the effect of bit number on the membrane potential \( v \), and recovery variable \( u \) signal shapes. The bit number selection depends on the required accuracy, dynamical system complexity, area limitations, and power limitations of the specific applications.

4.2. Cellular bifurcation analysis

From the dynamical systems point of view, the transition in the overall behavior of the system corresponds to a qualitative change of phase portrait of the system. This phenomenon, which determines the most fundamental computational properties of the system, is called bifurcation [Izhikevich, 2003]. In this section, we investigate the qualitative change of the cellular memristive phase plane corresponding to different bifurcation scenarios of the Izhikevich neuron model. At the first step, a number of concepts related to the different bifurcations ought to be defined:

Bistability: Coexisting of two stable phenomena (e.g. Stable Resting State and Stable Tonic State) that the neuron model exhibits one of them depending on the initial state, is called a bistability [Izhikevich, 2003]. Many neural models are bistable or can be made bistable when the parameters have appropriate values. Often bistability results from the coexistence of an equilibrium attractor corresponding to the resting state and a limit cycle attractor corresponding to the repetitive firing state.

Subthreshold Oscillations: In a number of neuronal responses, the firing is followed by damped subthreshold oscillations of the membrane potential under the spiking threshold. From a neuron model viewpoint, such a phenomenon is called a periodic subthreshold oscillation [Izhikevich, 2003]. If the periodic subthreshold oscillation attracts any nearby point, it is called a stable periodic subthreshold oscillation. If the periodic subthreshold oscillation repels any nearby point, it is called an unstable periodic subthreshold oscillation.

As shown in Fig. 9, four basic bifurcations, which we investigate in this section, are categorized based on these two phenomena. At the second step, we redefine a number of dynamical concepts according to our memristive cellular phase plane.

Cellular Nullclines: X Nullcline (XN), and Y Nullcline (YN) are two successive subsets of the phase plane \( PP \equiv \{ (X, Y) \mid X \in M, Y \in N \} \) defined as:

\[
\begin{align*}
\text{XN} \equiv \{ (X, Y) \mid X \in M, Y \in N, v_{\text{min}}^{\text{XN}} - v_{\text{min}} < \alpha \} \\
\text{YN} \equiv \{ (X, Y) \mid X \in M, Y \in N, v_{\text{min}}^{\text{YN}} - v_{\text{min}} < \alpha \}
\end{align*}
\]
Fig. 7. Time waveforms produced by the 64-bit memristive cellular approach as the various neuron-like responses of the neuron model by applying various input currents $I$. (a) Tonic spiking, (b) phasic spiking, (c) tonic bursting, (d) phasic bursting, (e) mixed mode, (f) spike frequency adaptation, (g) class I excitable, (h) class II excitable, (i) spike latency, (j) sub-threshold oscillation, (k) resonator, (l) integrator, (m) rebound spike, (n) rebound burst, (o) threshold variability, (p) bistability, (q) Depolarized After-Potential (DAP), (r) accommodation, (s) inhibitory induced spiking and (t) inhibitory induced bursting.
The subsets $X_N$ and $Y_N$ are the equivalents of $x$ and $y$ nullclines in the cellular space, if the proper $\alpha$ value is used.

**Equilibrium Cells (EC)** is a subset of the PP that satisfies the following condition:

$$EC \equiv \{(X,Y) \mid (X,Y) \in X_N, (X,Y) \in Y_N\}. \quad (42)$$

The subset $EC$ is the equivalent of the equilibrium point in the cellular space, if the proper $\alpha$ value is used. In the study of each bifurcation, some other concepts related to specific bifurcations are defined.

### 4.2.1. Saddle-node off-invariant bifurcation

A neuron with behavior based on this type of bifurcation exhibits bistability, and no subthreshold oscillations. For investigating dynamical behavior of this bifurcation, a number of basic concepts ought to be redefined in the cellular space:

**Sink Cells (SIC)** is a successive subset of the EC by the following condition:

$$SIC \equiv \{EC \mid \exists (X,Y) \in EC :$$

$$\begin{cases} |x_{\min} + (X - 1) \cdot \Delta x - x_{si}| < \frac{\Delta x}{2} \\ |y_{\min} + (Y - 1) \cdot \Delta y - y_{si}| < \frac{\Delta y}{2} \end{cases} \} \quad (43)$$

where $(x_{si}, y_{si})$ is the nodal sink of the system according to the continuous dynamic system definition [Izhikevich, 2003], and $\land$ is an AND operator.

**Source Cells (SOC)** is a successive subset of the EC by the following condition:
SOC \equiv \left\{ EC \mid \exists (X,Y) \in EC : \right. \\
\left. \begin{align*}
|x_{\min} + X \cdot \Delta x - x_a| &< \frac{\Delta x}{2} \\
\land |y_{\min} + Y \cdot \Delta y - y_a| &< \frac{\Delta y}{2}
\end{align*}\right\}
\tag{44}

where \((x_a, y_a)\) is the nodal source of the system according to the continuous dynamic system definition [Izhikevich, 2003].

**Cellular Attraction Domain (CAD)** is a successive subset of PP with the following condition:

\[
\text{CAD} \equiv \{(X,Y) \mid (X,Y) \in PP, \quad \exists n < M \cdot N : (X^{(n)}, Y^{(n)}) \in \text{SIC}\}
\tag{45}
\]

where \((X^{(n)}, Y^{(n)})\) is the location of point \((X,Y)\) after \(n\) one-cell motion in the PP. Note that trapping the point in the cellular phase plane corners, and its attempt to move out from the plane each time is counted as a cell change, despite no change in the point location.

Figures 10(a1)–10(a3) show the saddle-node on-invariant bifurcation in three captured steps of the process. In the first capture (a1), Izhikevich dynamical system creates SIC and SOC subsets in the PP. As shown in the figure, the SIC and SOC subsets cause a limited CAD subset in the cellular space. This limited CAD separates the cellular space to two different spaces. Any initial state point in the CAD subset is attracted to the SIC subset without any oscillation. Initial state point in the other cells causes a firing limit cycle. This phenomenon is known as bistability. In the second capture (a2), two SIC and SOC subsets are annihilated, and the CAD subset is consequently vanished. This process is completed in the third capture (a3), where any initial point in the cellular space results in a firing limit cycle.

### 4.2.2 Saddle-node on-invariant bifurcation

A neuron with behavior based on this type of bifurcation exhibits no bistability, and no subthreshold oscillations. Figures 10(b1)–10(b3) show the saddle-node on-invariant bifurcation in three captured steps of the process. In the first capture (b1), the Izhikevich dynamical system creates SIC and SOC subsets in the PP. As shown in the figure, the SIC causes a CAD subset all over the cellular space. In this condition, any initial state point in the cellular plane is attracted to the SIC subset without any oscillation. In the second capture (b2), two SIC and SOC subsets are annihilated, the CAD subset vanishes consequently, and the firing limit cycle is established in the location of annihilation. This process is completed in the third capture (b3), where any initial point in the cellular space results in a firing limit cycle.

### 4.2.3 Subcritical Andronov–Hopf bifurcation

A neuron with behavior based on this type of bifurcation exhibits bistability, and subthreshold oscillations. For investigating dynamical behavior of this bifurcation, a number of basic concepts ought to be redefined in the cellular space:

**Stable Focus Cells** (SFC) is a successive subset of the EC with the following condition:

\[
\text{SFC} \equiv \left\{ EC \mid \exists (X,Y) \in EC : \right. \\
\left. \begin{align*}
|x_{\min} + X \cdot \Delta x - x_d| &< \frac{\Delta x}{2} \\
\land |y_{\min} + Y \cdot \Delta y - y_d| &< \frac{\Delta y}{2}
\end{align*}\right\}
\tag{46}
\]

where \((x_d,y_d)\) is the stable focus of the system according to the continuous dynamic system definition [Izhikevich, 2003].

**Unstable Limit Cycle Cells** (ULCC) is a successive subset of the PP with the following condition:

\[
\text{ULCC} \equiv \{(X,Y) \mid ((X+1, Y) \in \text{CAD}) \\
\lor ((X-1, Y) \in \text{CAD}) \\
\lor ((X, Y+1) \in \text{CAD}) \\
\lor ((X, Y-1) \in \text{CAD}) \\
\lor ((X+1, Y-1) \in \text{CAD}) \\
\lor ((X-1, Y-1) \in \text{CAD}) \\
\lor ((X+1, Y+1) \in \text{CAD}) \\
\lor ((X-1, Y+1) \in \text{CAD})\}
\tag{47}
\]
Fig. 10. Theoretical analysis of (a1)–(a3) Saddle-node off-invariant bifurcation, (b1)–(b3) Saddle-node on-invariant bifurcation, (c1)–(c3) Subcritical Andronov–Hopf bifurcation and (d1)–(d3) Supercritical Andronov–Hopf bifurcation on the 64 × 64 cellular non-axial phase plane.
where CAD is the attractor area of the SFC, and ∨ is an OR operator. In other words, ULCC contains the border cells of the stable focus attractor area.

**Unstable Focus Cells (UFC)** is a successive subset of the EC with the following condition:

\[
\text{UFC} \equiv \{ (X, Y) \in \text{EC} : \\
(x_{\text{uf}} + X \cdot \Delta x - x_{\text{uf}}) \frac{\Delta x}{2} \\
\land (y_{\text{uf}} + Y \cdot \Delta y - y_{\text{uf}}) \frac{\Delta y}{2} \}
\]

(48)

where \((x_{\text{uf}}, y_{\text{uf}})\) is the unstable focus of the system according to the continuous dynamic system definition [Izhikevich, 2003].

Figures 10(c1)–10(c3) show the subcritical Andronov–Hopf bifurcation in three captured steps of the process. In the first capture (c1), Izhikevich dynamical system creates SFC subset in the PP which results in a CAD subset and ULCC border subset. As shown in the figure, ULCC separates the cellular space into two different spaces. Any initial state point in the CAD subset is attracted to the SFC on a damped oscillatory track in the cellular space. The initial state point in the other cells causes a firing limit cycle. This phenomenon is known as bistability. In the second capture (c2), the CAD area of the SFC is decreased, vanished, and the SFC converts to a UFC subset consequently. This process is completed in the third capture (c3), where any initial point in the cellular space results in a permanent limit cycle in the SLCC.

4.2.4. **Supercritical Andronov–Hopf bifurcation**

A neuron with behavior based on this type of bifurcation exhibits no bistability and subthreshold oscillations. For investigating the dynamical behavior of this bifurcation, a number of basic concepts ought to be redefined in the cellular space:

**Stable Limit Cycle Cells (SLCC)** is a successive subset of the PP with the following condition:

\[
\text{SLCC} \equiv \{ (X, Y) \mid \exists n < N \cdot M, u \neq 1 : \\
(X^{(n)}, Y^{(n)}) = (X, Y) \}
\]

(49)

Figures 10(d1)–10(d3) show the supercritical Andronov–Hopf bifurcation in three captured steps of the process. In the first capture (d1), the Izhikevich dynamical system creates SFC subset in the PP which results in a CAD subset all over PP. As shown in the figure, any initial state point in the PP subset is attracted to the SFC on a damped oscillatory track in the cellular space. In the second capture (d2), the CAD area of the SFC vanishes, and SFC converts to a UFC subset with a small SLCC subset. This process is completed in the third capture (d3), where any initial point in the cellular space results in a permanent limit cycle in the SLCC.

5. **Cellular Memristive FitzHugh–Nagumo Neuron Model**

In this section, we investigate the implementation of FitzHugh–Nagumo neuron model (FHN Model) [FitzHugh, 1961] on the CMDS platform as the second case study. The final goal of this section is to investigate the independence of CMDS to the computational effort of the target dynamical system, and its capability to implement various applicable dynamical systems. The FHN model is a two-dimensional simplification of the Hodgkin–Huxley model (HH model) of spike generation in squid giant axons which is mathematically more complex than Izhikevich neuron model because of a third power element in its equations. The simplified equations of this model are given by:

\[
\begin{aligned}
\dot{v} &= v - \frac{v^3}{3} - u + I \\
u &= 0.08(v + 0.7 - 0.8u)
\end{aligned}
\]

(50)

where \(v\) is the membrane potential variable, \(u\) is the recovery variable and \(I\) is the stimulus input current. Note that FHN model has no auxiliary resetting function, and it eliminates auxiliary logical hardware in the CMDS implementation. From the dynamical systems point of view, this system is an example of a relaxation oscillator because, if the external stimulus exceeds a certain threshold value, the system will exhibit a characteristic excursion in phase space, before the variables, and relax back to their rest values. This behavior is typical for spike generation (i.e. short elevation of membrane
Fig. 11. Phase-plane trajectory and time-domain representation of (a) absence of all-or-none spikes phenomenon, (b) excitation block phenomenon, (c) post-inhibitory rebound spike phenomenon and (d) spike accommodation phenomenon.
voltage) in a neuron after stimulation by an external input current.

According to the explanations in the previous section, the model can be mapped easily on the CMDS platform based on the similar equations for Izhikevich model. This model can present a number of important qualitative phenomena in different conditions based on the proportional bifurcations. Here, we investigate four significant phenomena on our CMDS approach of FHN model individually:

Absence of All-or-None Spike The FitzHugh–Nagumo model explained the absence of all-or-none spikes in the HH model in response to stimuli \( I \). Weak stimuli (small pulses of \( I \)) result in small-amplitude trajectories that correspond to subthreshold responses; stronger stimuli result in intermediate-amplitude trajectories that correspond to partial-amplitude spikes; and strong stimuli result in large-amplitude trajectories that correspond to suprathreshold response (firing). Figure 11(a) shows the phase-plane representation, and time-domain representation of this phenomenon. The size of cycles in the phase plane corresponds to the amplitude of pulses in the time domain.

Excitation Block The FHN model explains the cessation of repetitive spiking as the amplitude of the stimulus current increases. This phenomenon is called the excitation block phenomenon. When \( I \) is weak or zero, the equilibrium (intersection of nullclines) is on the left (stable) branch of \( V \)-nullcline, and the model is resting. Increasing \( I \) shifts the nullcline upward, and the equilibrium slides onto the middle (unstable) branch of the nullcline. The model exhibits periodic spiking activity in this case. Increasing the stimulus further shifts the equilibrium to the right (stable) branch of the \( V \)-shaped nullcline, and the oscillations are blocked. Figure 11(b) shows the phase-plane, and time-domain representations of the CMDS response trajectory from resting state to the oscillation, and then blocking states when a ramp input is injected into the system.

Post-inhibitory Rebound Spike The FHN model explained the phenomenon of post-inhibitory rebound spikes, called anodal break excitation. Figure 11(c) shows the trajectory of this phenomenon in the CMDS when a negative short pulse is injected in the system. As the stimulus \( I \) becomes negative (hyperpolarization), the resting state shifts to the left. As the system is released from hyperpolarization (anodal break), the trajectory starts from a point far below the resting state, makes a large-amplitude excursion (fires a transient spike, and then returns to the resting state).

Spike Accommodation The FHN model explained the dynamical mechanism of spike accommodation in HH-type models. When stimulation strength \( I \) increases slowly, the neuron remains quiescent. The resting equilibrium of the FHN model shifts slowly to the right, and the state of the system follows it smoothly without firing spikes. In contrast, when the stimulation is increased abruptly, even by a smaller amount, the trajectory cannot potentially go directly to the new resting state, but fires a transient spike. As shown in Fig. 11(d), this phenomenon occurs in the CMDS system when ramp and step signals are injected to the system.

6. CMDS Remarks

In this section, two important remarks about CMDS are discussed.

6.1. CMDS learning

One of the most important benefits of CMDS is its learning capability due to its flexibility brought by the memristors. Using this specification, different learning schemes can be applied to the CMDS. The block diagram of a sample learning system is shown in Fig. 12 in order to clarify the procedure of the learning concept in this subsection.

In this figure, the basic concept of a learning system with specific application is depicted. In this

![Fig. 12. Sample learning system structure for extracting and imitating behavior of an unknown system (black box system).](image-url)
structure, there is a black box system with unknown behavior. We assume that the state variables of the system are observable, we can apply different input stimulating currents to the system, and force it into different states, and the behavior of the black box system can be mimicked by the CMDS system. This black box system plays the role of teacher in the structure, and its output signals are the teacher signals for CMDS to follow its behavior. Learning controller puts the black box, and CMDS in equal states, compares their behaviors based on the changes in state variables, and then tries to modify the CMDS behavior according to the black box behavior. During the learning phase, learning signals change the memristance of the memristors to modify the CMDS behavior.

6.2. **CMDS networks**

The capability of networking dynamical systems with different variable weights is one of the most crucial demands in different applications from network oscillators to bio-inspired neural networks. In this subsection, we present the basic concept of a networking scheme to prove the networking capability of CMDS.

As explained, the CMDS uses a digital one-hot coding approach to present the state variables and analog voltage to inject the input to the system. On the other hand, we have to provide a variable weight to connect CMDS cells, so the resultant system is flexible, and capable of applying different network learning schemes. There are two steps to provide a networking platform for CMDS. First, a circuit to convert the digital value of the state variables to the proportional analog voltage, and second, a circuit to add analog input voltages from CMDS cells with different weights, and apply the results to the VCOs. Figure 13 shows the overall circuit for these two steps.

As shown in the figure, the digital value of the state variables is converted into the analog voltage through one-hot to Analog Converter Units (OAC Units). The \( v_{e(i)} \) inputs are the proportional voltage for each discrete value of the state variable. Using Eqs. (6) and (27), these voltage values can be calculated by:

\[
  v_{e(i)} = \left( x_{\text{min}} + (i - 1) \cdot \Delta x \right) \\
  \frac{\Delta V_{max}}{V_{x_{\text{max}}}} + v_{\text{min}}.
\]

The analog voltages are applied to an analog voltage adder with OPAMP. The output of the unit can be obtained by:

\[
  v_{\text{out}} = v_{e(1)} \cdot \frac{R_f}{R_{m1}} + v_{e(2)} \cdot \frac{R_f}{R_{m2}} + \ldots + v_{e(K)} \cdot \frac{R_f}{R_{mK}}.
\]

where \( v_{e(i)} \) is the \( i \)th analog input voltage, and \( R_{m} \) is the input memristance of the \( i \)th input. According to the equation, each input value is multiplied by a weight given by \( \frac{R_f}{R_{m}} \). So the weights can be modified by changing the input memristances. Memristance modification is performed by applying different signal shapes to the plasticity ports of the circuit (free terminals of the crossbar) shown in Fig. 12. Note that the structure of memristors in the Network Unit is based on the crossbar structure that is easy to implement.

7. **Conclusion**

In this study, a general cellular-based mapping for two-dimensional dynamical systems, and its
implementation on an efficient nano-scale memristive circuit were presented. We investigated the fidelity of our approach on well-known two-dimensional dynamical systems — the Izhikevich neuron model and the FitzHugh-Nagumo model — describing biological neuron behaviors and responses. We stimulated our cellular memristive neuron with various signal shapes, and showed that it can mimic different dynamical behaviors. In addition to investigating neuron behavior, we analyzed the four main types of bifurcations using our approach to confirm that it can reproduce not only the response characteristics but also the bifurcation mechanisms of the dynamical system. The benefits of our approach include:

• High accuracy
• Independence to computational effort
• High programmability
• Capability of implementing mathematically indescribable dynamical systems, and learnable dynamical systems
• Nanoscale efficient implementation will be the key to the development of future applications of Cellular Memristive Dynamical Systems (CMDS).

Future problems include the following.

• A more detailed hardware design, and optimization for different blocks of the CMDS system.
• Development of powerful learning algorithms for cellular dynamical systems.

Acknowledgments

The authors would like to thank Prof. Hyongsuk Kim for his fruitful comments. This research is supported in part by an AFOSR grant number FA9550-13-1-0136.

References


M. Bavandpour et al.


