Storing and retrieving wavefronts with resistive temporal memory

Advait Madhavan*, Mark D. Stiles†
Physical Measurement Laboratory, National Institute of Standards and Technology*†
Institute for Research in Electronics and Applied Physics, University of Maryland, College Park*
Email: *advait.madhavan@nist.gov, †mark.stiles@nist.gov

Abstract—We extend the reach of temporal computing schemes by developing a memory for multi-channel temporal patterns or “wavefronts.” This temporal memory re-purposes conventional one-transistor-one-resistor (1T1R) memristor crossbars for use in an arrival-time coded, single-event-per-wire temporal computing environment. The memristor resistances and the associated circuit capacitances provide the necessary time constants, enabling the memory array to store and retrieve wavefronts. The retrieval operation of such a memory is naturally in the temporal domain and the resulting wavefronts can be used to trigger time-domain computations. While recording the wavefronts can be done using standard digital techniques, that approach has substantial translation costs between temporal and digital domains. To avoid these costs, we propose a spike timing dependent plasticity (STDP) inspired wavefront recording scheme to capture incoming wavefronts. We simulate these designs with experimentally validated memristor models and analyze the effects of memristor non-idealities on the operation of such a memory.

I. THE NEED FOR TEMPORAL MEMORY

The three pillars that form the foundation of any computing system are computation (for processing), input/output (I/O) (for sensing and feedback), and memory (for storage). In the context of single-spike-per-wire, arrival-time coded computation, circuits that allow sensing and processing natively in the temporal domain are already being researched. Dynamic vision sensor (DVS) cameras [1], time-to-first-spike (TTFS) vision sensors [2], and address event representation (AER) ears [3] are a few examples of the sensing systems that natively encode information in the temporal domain. On the computational side, a space-time computing approach [4] has been proposed as a novel paradigm that encodes information in the relative arrival time between input events. A direct implementation of such a paradigm with off the shelf complementary-metal-oxide-semiconductor (CMOS) technology has also been proposed and demonstrated [5], [6]. The active research on two of the three pillars increases the urgency to develop a memory that natively operates in the time domain. As of now, no such memory exists.

The biological motivation behind single-spike-per-wire temporal computation can be traced back to Thorpe and Imbert’s work [7] arguing that the speed of processing of the visual system is too fast for a rate-coded interpretation of neural computation to be feasible [8]. Instead, they proposed a wavefront based computing approach that encodes information in the relative arrival time between a volley of spikes [9] as is shown in Fig. [1(a)]. This information representation is radically different from the conventional Boolean one and opens up a vastly different trade-off space of possible computing architectures [4]–[6], [10]–[12].

The temporal domain allows for different encoding schemes, among which two have been well studied [13]–[15]. One is a timing code, where the exact delays between the spikes carry information [14], [16]. A more relaxed version of such an approach, though less dense in coding space, is a rank order code, in which only the relative orderings of the spikes carry information [15], [16]. Though sparser than their exact timing counterparts, order codes have been shown to contain appreciable information capacity while still maintaining robustness to variations in individual spike timing [15]. Though the first temporal networks were studied as early as 1990 [13], more recently, rank order codes have been studied in the context of deep spiking neural networks [17]–[19]. Such single-event-per-wire based (also known as non-leaky integrate and fire) models have been trained with modified STDP algorithms. Early results [18], [20] report comparable accuracies to deep learning networks but with small network sizes.

Building hardware implementations of such systems requires physical realizations of events. Race logic [5] is a temporally coded logic family that takes advantage of the simplicity of the digital domain and represents events with
An ideal temporal memory would be one that could be natively capture wavefronts directly into memristors. We then describe a proposed solution to from analog to digital and back, has significant overhead corresponding captured digital values. This domain shifting, specific write circuitry to tune the memristor resistances to the through standard digital circuit techniques which then invokes such a circuit, the values of the memristive states can be directly read out at as a waveform of digital rising edges, also known as waveform recalling. This is shown in Fig. 2 where a linear variation in memristive values leads to a linearly spaced output wavefront.

Though the structure of the crossbar remains the same, the way it is used in this work differs in some important ways from conventional approaches. When used in a multilevel memory or feed-forward inference context, a static read voltage is applied across the device (pinned with a sense/measurement amplifier) while the resultant current is summed and measured. Hence, the energy efficiency in these approaches improves the larger the $R_{on}$ and $R_{off}$ resistances become. In contrast, in this RC charging based recall mode of operation, the voltage drop across the device is not static, because the voltage on the output capacitor changes during a read operation (Fig. 2(b)(iii)).

This changing voltage has two advantages. First, and more important, it decouples the energy cost per read operation from the value stored in the memristor. Independent of the state of the device, a single read operation consumes $CV_{\text{read}}^2(\approx 600 \text{ fJ})$ of energy per line, with $CV_{\text{read}}^2/2$ lost due to joule heating across the memristor and $CV_{\text{read}}^2/2$ stored on the capacitor. This data independent energy cost allows memristors to be used in the high conductance regime, without incurring the increased energy cost. Circuit and architectural designs can then take advantage of the high conductance regime, where the behaviour of the device is more linear, repeatable and less susceptible to variation. Recently, for very low resistance states, the device to device variation has shown to be $\leq 1\%$ [21]. The second advantage is that the degree of read disturb on the device is reduced as the full read voltage is applied across the device for a relatively short period of time.

To enable easy interface with digital signal levels, level-shifters are required to translate between the memristor read voltages ($V_{\text{read}}$) and digital voltage levels ($V_{\text{dd}}$). This shifting down process can be implemented with regular inverters but the shifting up process requires either current mirror based level-shifters or cross coupled level-shifters. The current mirror based designs have a smoother response, and consume static power while the cross coupled versions are more power efficient, but have a more complicated response.

The cross coupled topology is representative of a positive feedback loop between transistors M1-M4 (Fig. 3). This positive feedback loop itself has a time constant that varies with the current charging the input node. This variable time constant can add timing uncertainties that are data dependent and could cause errors. One way to avoid this problem is to take advantage of the one sided nature of this information encoding. Using rising edges only determines the transistor that is responsible for the pull-down so it can be sized.
The circuit then applies backward pulses of variable lengths. The first arriving edge is conceptually treated as the “post” edge. Encode tunable delays, STDP-like behaviour can be used to create a regime where the Taylor series expansion of its output timings for recalling the wavefront is linear, arising from this first-arriving “post” event and the later-arriving events, which can be thought of as “pre” events. The device with the largest difference between “pre” and “post” events, has the maximum conductance change, and hence the highest resistance. When a wavefront is then recalled, the highest conductance device responds first and the most resistive one responds last, preserving the wavefront shape.

Simulation results for such a procedure are shown in Fig. 4(b). These simulations are performed in a 180 nm process node, with a 1.8 V power supply. The memristor models used are from [24], and are modelled based on experimental measurements reported in [24]. The wavefront recording operation proceeds by first initializing the column in question, (column 1, shown in figure 4(a)), with all memristors set to the ON state (≈ 10 kΩ) and the enable line (E_1) activated. This can be seen in the first 100 ns of figure 4(b)(v) with all devices having the same impedance. The write path through the multiplexers, as shown in figure 4(a), is also activated, such that the OR gate controls the source line (SL).

The wavefront (having a dynamic range of 40 ns) to be recorded is presented at the digital bit lines (DBLs), which behave like the input in this phase of operation. Similarly to the digital case, the first arriving rising edge is detected by an OR gate, which triggers the application of an appropriate write voltage ($V_{\text{write}} \approx 1.4$ V), through the multiplexer, to the source line (SL). The bit-lines (BLs) of the array are operated in the write voltage regime with rising edges level shifted down from $V_{\text{id}}$ to $V_{\text{write}}$. Each device sees the difference in voltage between the source line (figure 4(b)(iv)) and corresponding bit lines (figure 4(b)(iii)) applied across it. For the device corresponding to the first arriving edge, both its source line and bit line go high at the same time, so there is no change in the memristive state. Meanwhile, the other devices experience a reverse $V_{\text{write}}$ voltage across them, since their edges haven’t arrived yet. They experience this voltage for the difference in time between their corresponding edge and the first arriving edge, hence causing a change in the memristive state proportional to the relative times between the inputs.

Once appropriate pulse lengths have been successfully applied across the devices, a copy of the input wavefront should be captured into the memristive device states. The last arriving edge signals the end of the recording operation and the circuit is reset with an external reset pulse. The reset pulse (not shown in the figures) discharges the crossbar without affecting the device state. The array is now ready for playback.

In order to get linear behavior out of such a device, it must operate in a regime where the Taylor series expansion of its nature of Boltzmann statistics and energy barriers.

III. Capturing Wavefronts: Digital vs Native

A functionally correct digital timing measurement approach to record wavefronts is shown in Fig. 2(b). High speed up-counters can be used for time scales on the order of 1 ns to 50 ns, while vernier delay lines, which extend the precision to the order of a single-inverter-delay, can be used for more precise measurements [22]. Using race logic principles, the first arriving edge is detected with an OR gate, which signals the beginning of the timing measurement system (counter or vernier delay line). With each subsequently arriving rising edge, the corresponding count value is captured in a register bank. An AND gate signals the last arriving input, at which time the recording process comes to an end with a digital representation of the relative arrival times stored in a temporary register bank. These values can be used as targets for a closed loop feedback programming approach [23] that writes the corresponding values into the correct memory column.

To increase the energy efficiency of wavefront recording we eliminate the need to translate between encoding domains by using the ability to change memristor resistances with applied voltage pulses. This native approach to capturing wavefronts, results in a more natural and energy efficient, albeit more error prone implementation. In a time coded information representation, in which the plastic memristor resistances explicitly encode tunable delays, STDP-like behaviour can be used to record wavefronts as shown in Fig. 4. In this approach, the first arriving edge is conceptually treated as the “post” edge. The circuit then applies backward pulses of variable lengths across the memristors proportional to the difference in timing between this first-arriving “post” event and the later-arriving events, which can be thought of as “pre” events. The device with the largest difference between “pre” and “post” events, has the maximum conductance change, and hence the highest resistance. When a wavefront is then recalled, the highest conductance device responds first and the most resistive one responds last, preserving the wavefront shape.

Fig. 3. Asymmetric rising edge level shifter: Here transistor M2 is sized larger than its counterpart M1 such that node “b” is pulled down faster with little competition from M1 via node “a”. The inverter with a “T” inside represents a tri-state buffer.

III. Capturing Wavefronts: Digital vs Native

A functionally correct digital timing measurement approach to record wavefronts is shown in Fig. 2(b). High speed up-counters can be used for time scales on the order of 1 ns to 50 ns, while vernier delay lines, which extend the precision to the order of a single-inverter-delay, can be used for more precise measurements [22]. Using race logic principles, the first arriving edge is detected with an OR gate, which signals the beginning of the timing measurement system (counter or vernier delay line). With each subsequently arriving rising edge, the corresponding count value is captured in a register bank. An AND gate signals the last arriving input, at which time the recording process comes to an end with a digital representation of the relative arrival times stored in a temporary register bank. These values can be used as targets for a closed loop feedback programming approach [23] that writes the corresponding values into the correct memory column.

To increase the energy efficiency of wavefront recording we eliminate the need to translate between encoding domains by using the ability to change memristor resistances with applied voltage pulses. This native approach to capturing wavefronts, results in a more natural and energy efficient, albeit more error prone implementation. In a time coded information representation, in which the plastic memristor resistances explicitly encode tunable delays, STDP-like behaviour can be used to record wavefronts as shown in Fig. 4. In this approach, the first arriving edge is conceptually treated as the “post” edge. The circuit then applies backward pulses of variable lengths across the memristors proportional to the difference in timing between this first-arriving “post” event and the later-arriving events, which can be thought of as “pre” events. The device with the largest difference between “pre” and “post” events, has the maximum conductance change, and hence the highest resistance. When a wavefront is then recalled, the highest conductance device responds first and the most resistive one responds last, preserving the wavefront shape.

Simulation results for such a procedure are shown in Fig. 4(b). These simulations are performed in a 180 nm process node, with a 1.8 V power supply. The memristor models used are from [24], and are modelled based on experimental measurements reported in [24]. The wavefront recording operation proceeds by first initializing the column in question, (column 1, shown in figure 4(a)), with all memristors set to the ON state (≈ 10 kΩ) and the enable line (E_1) activated. This can be seen in the first 100 ns of figure 4(b)(v) with all devices having the same impedance. The write path through the multiplexers, as shown in figure 4(a), is also activated, such that the OR gate controls the source line (SL).

The wavefront (having a dynamic range of 40 ns) to be recorded is presented at the digital bit lines (DBLs), which behave like the input in this phase of operation. Similarly to the digital case, the first arriving rising edge is detected by an OR gate, which triggers the application of an appropriate write voltage ($V_{\text{write}} \approx 1.4$ V), through the multiplexer, to the source line (SL). The bit-lines (BLs) of the array are operated in the write voltage regime with rising edges level shifted down from $V_{\text{id}}$ to $V_{\text{write}}$. Each device sees the difference in voltage between the source line (figure 4(b)(iv)) and corresponding bit lines (figure 4(b)(iii)) applied across it. For the device corresponding to the first arriving edge, both its source line and bit line go high at the same time, so there is no change in the memristive state. Meanwhile, the other devices experience a reverse $V_{\text{write}}$ voltage across them, since their edges haven’t arrived yet. They experience this voltage for the difference in time between their corresponding edge and the first arriving edge, hence causing a change in the memristive state proportional to the relative times between the inputs.

Once appropriate pulse lengths have been successfully applied across the devices, a copy of the input wavefront should be captured into the memristive device states. The last arriving edge signals the end of the recording operation and the circuit is reset with an external reset pulse. The reset pulse (not shown in the figures) discharges the crossbar without affecting the device state. The array is now ready for playback.

IV. Discussion

While such a wavefront recording approach seems feasible, some problems arise in the context of exact timing codes. First, the relationship between memristor resistance and relative output timings for recalling the wavefront is linear, arising directly from the $t \propto RC$ relationship. On the other hand, for recording the wavefront, the relationship between memristor conductance and voltage pulse duration is not linear, and depends on material properties. Since most memristive state change dynamics are governed by transitions over energy barriers, the effectiveness of a fixed voltage to change the device state drops logarithmically. In the wavefront recording process, a linearly spaced input wavefront will end up creating a logarithmically spaced resistive change, which when recalled would create a logarithmically spaced output wavefront. This problem is fundamental, being governed by the exponential nature of Boltzmann statistics and energy barriers.
behavior has small higher order coefficients, so that it can be approximated as linear. Such behavior can be seen for a range of voltages where a sharp pulse (≤ 40 ns) across the device creates a linear change in the devices state (from 10 kΩ to 40 kΩ), which tapers off if the pulse is applied for a longer time. Here, as shown in figure 4(e)(v), our pulse duration is calibrated to access that specific linear region of the memristor IV characteristics, and therefore does not access the complete available dynamic range.

The reduced range is not detrimental and depends on the quality of the memristive devices being used. Multiple groups have shown 5 bit or more resolution in limited resistance ranges with the low resistance state incurring, programming cycle to cycle variations as low as 4.2% and device to device variation as low as 4.5% [26]. For very low resistances (between 1 kΩ and 10 kΩ), even lower variation numbers have been reported (≤ 1% [21]). Such technological improvements allow us to extract 4 to 5 bits of precision, even from a reduced dynamic range.

A second difficulty for exact timing codes is that the time scales of recording and of recalling need to match. For example, the resistance change created by 10 ns pulses in the recording process, should create 10 ns spaced edges when recalled. While the former is a material property and cannot be changed by circuit techniques, the latter can be addressed by adding a digitally programmable capacitances (∼1 pF, in the current simulation) on the output line to correctly scale the timescale. For small array sizes such a capacitance can take up extra area, but as the array is scaled to more representative sizes, the crossbar, transistor-drain and driver capacitances will contribute significantly to this capacitance. Array scaling will also require scaling of the array drive circuits, especially with the high conductance regime operation. Though the write drivers on the bit line do not need to be adjusted, but the source line write driver will have to be designed to support \( N \) memristors in parallel during the capture phase. Future work includes a more detailed scaling analysis accounting for crossbar wire resistances and capacitances.

An important point to note is that rank order codes are more tolerant to the aforementioned concerns than exact timing codes. Logarithmic compression preserves order, and variable capacitances can be used with order codes to stretch the logarithmically compressed stored values. This allows enough write pulse duration to still change state on the next write operation. This makes rank order codes a more robust and error tolerant encoding for this kind of a temporal memory.

V. CONCLUSION

In this work we have proposed and validated through simulation, a single-event-per-wire temporal memory that operates in the sub 50 ns timing range while utilizing the low variability, low resistance states (10 kΩ to 40 kΩ) of memristive devices. We show how our recalling/playback operation has an energy cost of about 600 fJ per line, whose magnitude is independent of the device conductance. Rank order coded architectures seem to be the more promising encoding schemes for such an approach due to their error tolerance. Though many challenges remain, we believe that this is a first step towards realizing temporal memories that can work synergistically with tomorrow’s temporally coded computing architectures.
