

2nd International Through-life Engineering Services Conference

Using Memristors to Handle Cell Failures in Flexible Networks: From Programmed Cell Death to Zombies.

Oliver Matthews^a, Ioannis Georgilas^a, Ella Gale^{a*}

^a*Bristol Robotics Laboratory, T Building, University of the West of England, Frenchay Campus, Coldhardbour Lane, Bristol, BS16 1QY, UK*

* Corresponding author. Tel.: +44-117-3286-175; E-mail address: ella.gale@uwe.ac.uk

Abstract

Within a network, cells may fail in multiple ways. It is important that such cells be both identified and isolated from the network as a whole to both allow the failing cell's tasks to be reallocated and for the prevention of cascade failures where a malfunctioning cell causes other cells to fail. We present a novel, low-power approach using simple memristor-based electrical circuits that provides autonomic handling of clock sync issues as well as supporting cell isolation by either the cell or the network and cell resurrection by the network.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Selection and peer-review under responsibility of the International Scientific Committee of the “2nd International Through-life Engineering Services Conference” and the Programme Chair – Ashutosh Tiwari

Memristors; Flexible networks

1. Background

A fault-tolerant electronic architecture system will typically have a number of tasks it must handle. For instance, an aircraft's systems will need to handle everything from keeping track of the aircraft's location and the attitudes required on its various control surfaces through to handling the air conditioning within the craft and the in-flight entertainment (on a passenger plane). This is typically handled by individual, specialized systems dedicated to each task. Although this approach allows for a number of advantages, it suffers from one main drawback; as each component can only perform a single task, there needs to be multiple instances of each component to provide redundancy in the case of component failure.

An alternative route is via a flexible network. In this approach the system consists of a number of discrete flexible cells along with some form of control mechanism, all connected via a communications channel. These cells, are typically implemented using Field Programmable Gate Arrays (FPGAs)[1] for performance reasons, although a low power risc-based System On Chip would also be feasible.

The cells are inspired by prokaryote biology and attempt to implement artificial biofilm organisation and resilience (see for example[2] for an implementation). In this paper, we focus on an architecture where each cell can potentially perform any of the required tasks of the system and is allocated a single task from the list. This leads to two main benefits over the traditional approach:

- A spare cell can replace any failure, which, as the probability of a failure happening is low, means that only a few spare cells are required rather than duplicates of all components.
- If all the spare cells are in use, a cell performing a non-essential task can be reassigned to take over a more important one.

This approach can be complicated by having cells of varying capability (such as in the SABRE architecture[2]), by having cells handle multiple tasks or by allocating multiple cells to a single task. In this paper we will concentrate only on the simple model outlined above. We shall suggest a novel memristor-based synchronisation approach which has advantages over software-based synchronisation by being very low power, more robust to

electromagnetic field and cosmic ray state-changing and an inherent part of the hardware which means that it can't be turned off and does not rely on functioning software.

2. Problem Specifics

2.1. Heartbeats and the control mechanism

For a flexible system to function there needs to be a controller system, either explicit or implicit, that handles task allocation. This may be implemented as a separate system to the cells, a task run on a cell, or it may be built into the cells using some form of consensus algorithm (e.g. Paxos[1]). Either way the cells must be able to communicate freely with each other.

A particular issue with such networks is maintaining synchronisation of state between both the cells and the control system and/or between the cells. This can be solved with some form of a 'heartbeat': a regular signal sent between cells that is used as a clock for the network.

2.2. Cell failure modes

Cells can fail in numerous ways, and the network needs to be able to handle all of them. We can classify the failure modes as falling within one of three basic categories:

- Dead cell - The cell stops working and no longer responds to outside stimulus. When this occurs the network needs to detect the failure and allocate a replacement.
- Dying cell - The cell experiences a fatal fault but is able to detect the situation and can perform some form of orderly shutdown, similar to apoptosis in cell biology. These failures are the easiest to deal with, but some caution is required in the hardware design to ensure that the dead cells do not act as unintentional grounds once shutdown. Also, the network itself should take responsibility for disconnecting the cell in case the failure also prevents the cell from shutting itself off properly (turning into a zombie, as below).
- Zombie cell - The cell is functionally broken and should be dead, however it is still in some (incorrect) manner active on the network, in comparison to biology this is functionally equivalent to cancer. Failures falling into this category are the trickiest to handle, as they require detection and until they are detected not only will that cell's task(s) not get performed, but the cell may be transmitting random noise into the network, disrupting the activities of other nodes (potentially causing them to be diagnosed incorrectly as malfunctioning).

3. The Memristor

The memristor is the recently discovered [4] 4th fundamental circuit element[5] which relates magnetic flux, φ (the time intergral of voltage), to charge, q , (the time integral of current) via the constitutive relation:

$$d\varphi = M(q) dq , \tag{1}$$

where M is the memristance and is dependent on the charge that has passed through the circuit. The linear relation between the intragrals of current and voltage mean that the device is nonlinear in V - I measurements. As the charge is time dependent, $q(t)$, the memristance also varies with time, but at each point in time, t , the voltage, V , and current, I , through the memristor are linearly related

$$V(t) = M(q(t))I(t) , \tag{2}$$

demonstrating that the memristors is a non-linear resistor whose resistance is dependent on some memory property q . It is this memory property[6] that stores the state of the memristor and allows the memristor to have a memory. This memory gives rise to hysteresis, see Fig. 1, the distinctive pinched shape is due to the fact that a perfect theoretical memristor is a passive element and thus should go through the origin.

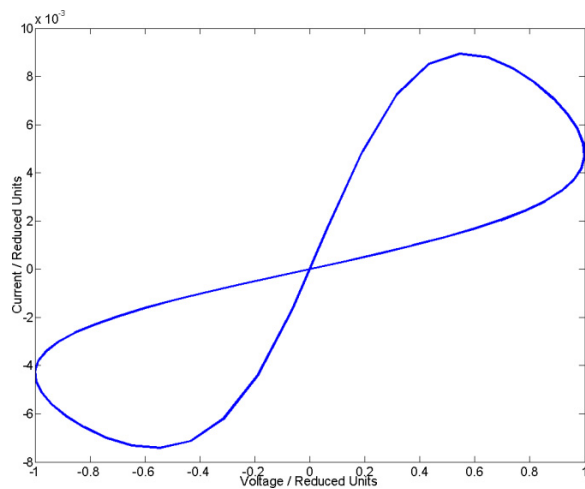


Figure 1. An example memristor I - V curve as simulated under A.C.

3.1. Memristor Spike Addition and Subtraction

Under D.C. voltage the memristor responds to changes in voltage, ΔV , with a current spike, ΔI , as shown in Fig. 2 for a voltage square wave between +0.1V and -0.1V, this data is taken from experimental tests with TiO_2 sol-gel memristors as made in our laboratory[7]. The memristors take 3.3s to equilibrate to this change or equivalently, we can say that the memristor has a short-term memory of the spike for 3.3s[8].

If a second ΔV is input whilst the memristor still holds this short-term memory (i.e. before the device equilibrates), the second spike ΔI is smaller than would be expected, see, for example Fig. 3. In this experiment the memristor was subject to the same square wave as for Fig. 2, but memristor was given only 1 timestep ($\sim 0.2s$) to equilibrate after the positive spike at 20s before a negative voltage was applied. The short-term memory of the positive spike has interacted with the negative spike response, causing a reproducible and repeatable subtraction of the spike current. This effect

can also be used to do addition and has been used to make Boolean logic gates[9].

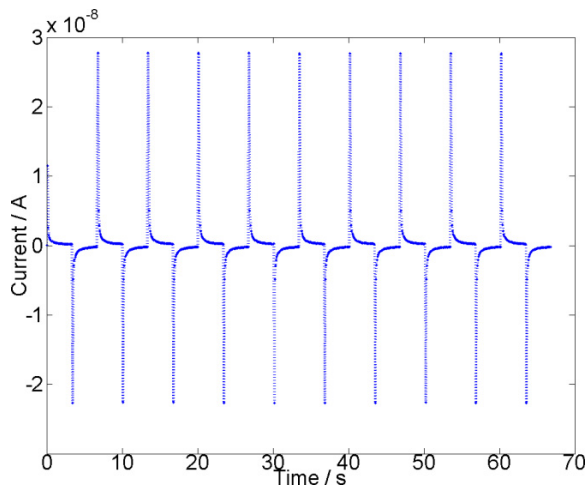


Figure 2. D.C. response of a memristor subjected to a voltage square wave.

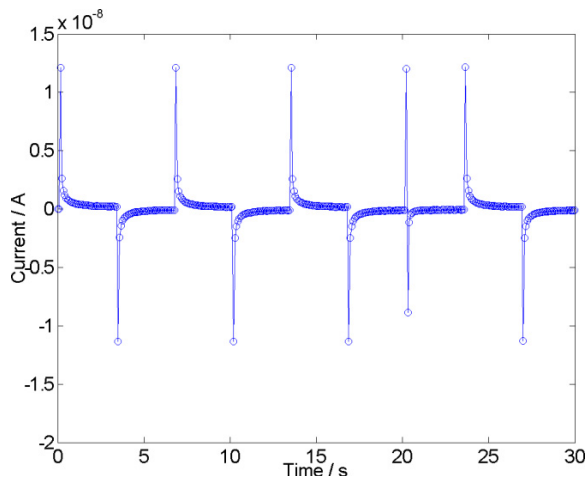


Figure 3. Memristor spike 'addition'.

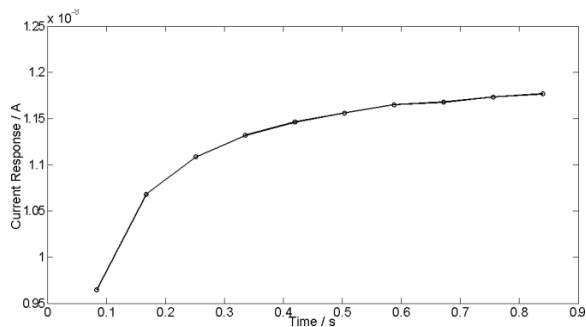


Figure 4. Second current spike size depends on the time between spikes.

The size of this response spike is highly dependent on when the second spike is input. Figure 4 shows the spike addition for spikes separated by the increasing time. We see that

when the two spikes are 0.1s apart the current response of the second spike is the smaller than when they are separated by 0.2s.

As the spike size is highly dependent on the time gap between spike events, we can use this effect (and its compliment spike addition) to synchronise cells in a network. If spiking cells fall out of synchronisation, the size of the current response will rise above a threshold. We will now go through an example of a watchdog circuit to respond to this effect.

4. The watchdog circuit

The watchdog circuit is as shown in Fig 5. It consists of a low-pass filter in the form of a resistor and a capacitor, a voltage comparator implemented with an op-amplifier (as in [10]) and a D-type flip-flop with reset signal. The basic idea of the implementation is that the RC circuit will filter the spikes of current from the memristor and generate a slowly degrading voltage for the comparator. The capacitor will be charged by the current of the memristor, minus the current through the resistor, and then slowly discharge through the resistor. Selecting the appropriate values for the capacitor and resistor is based on the discharging equations for capacitors given in Eqn 3:

$$I_C = \frac{V_0}{R} e^{-\frac{t}{RC}} \text{ and } V_C = V_0 e^{-\frac{t}{RC}}, \quad (3)$$

Where V_0 is the voltage that the capacitor has been charged to by the memristor's current, defined in Eqn 4:

$$V_0 = f(I_M) = \frac{Q}{C}, \quad (4)$$

where $Q = I_M t$. The comparator, which is an operation amplifier with positive voltage set as the power supply and negative voltage the ground, is comparing the output voltage of the capacitor with a pre-defined threshold value, V_{th} . If $V_C \leq V_{th}$, then the amplifier's output is the power supply value which is also the logical high for the circuit. As soon as a new pulse from the memristor arrives V_C increases again changing the output of the amplifier to logical low. The transition from logical low to logical high and back is sensed by the D-type flip-flop as a CLK signal and as a result the output Q is being raised driving the latch to disengage the cell from the bus. For the cell to be reconnected to the bus the controller can reset the flip-flop by sending a logical high pulse in the reset line.

The benefit of the circuit is that is very simple and energy efficient, if the respective values are selected. All the components are widely used and standardised allowing for very small tolerances, thus very predictable behaviour. As a result the difference between V_{th} and V_C , with the latter reflecting a change in I_M , can be selected low enough so as to trace small variations of the synchronization of the heart beat signals.

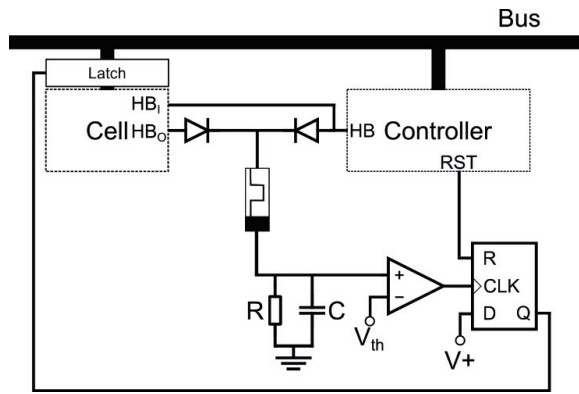


Figure 5. The watchdog circuit

4.1. Dead cells

The watchdog's handling of dead cells is trivial – when a cell dies and stops sending the heartbeat the latch trips and the cell is isolated.

4.2. Dying cells

Should a cell detect some form of internal error within itself that doesn't (yet) result in it being unable to send the heartbeat, then it can mimic programmed cell death (apoptosis) mechanisms by voluntarily stopping the clock such as by switching off the memristor and effectively killing itself. An example of such a situation would be the detection of a parity error in a memory bank.

4.3. Zombie cells

If a cell fails but in such a way that the heart-beat continues to be sent, this causes the spike to be above a threshold and will be spotted by the controller which can then cut the signal from its side. This isolates the cell and prevents its malfunctioning signals from causing interference elsewhere.

4.4. Heart-beat synchronization loss

As the network requires that the cells all operate in time with each other, a cell which is unable keep time (for instance if overloaded to the point where it is unable to process its workload quickly enough) is effectively a zombie cell in that it's signals will be offset from the rest of the network and thus be unintelligible. Specifically, the memristor subtraction will now be small and one of the spikes will be above the threshold. The design of the watchdog circuit means that this will be handled automatically and any cells that lose time will simply get cut off.

4.5. Resurrection

It is possible for the network to bring a cell back to life

by re-instating its access via the reset line on the flip-flop. Beyond allowing a reset of the state of the system, there are several situations in which this might be a useful action:

- When a cell has killed itself (due to a detected internal fault), it may report the nature of the fault to the network before doing so. In extreme circumstances it may be more desirable to reactivate this cell (perhaps assigning it lesser tasks) rather than not having sufficient cells in the event of multiple cell failure.
- When a cell has been cut off due to loss of synchronization, the cell can potentially realize this, reset itself and start sending the heart-beat again. The network can, at its option, pulse the reset to see if this is the case (it is not advisable to allow the cell to reset the latch as a pathological case can easily be imagined where a zombie repeatedly resets the latch).

5. Conclusions

Memristor spike subtraction is a useful approach for a cell synchronisation heartbeat. If current spikes from the controller and the cell arrive within the correct time window, the spikes up and appear below a threshold. If they fall out of sync or one of the spike signals fails, the spikes will be above the threshold. As this is a result of the hardware, it is got for 'free' in a system with a changing voltage and will operate at a level below the software, making it resilient to software faults.

References

- [1] Hauck, S. "The roles of FPGAs in reprogrammable systems", *Proceedings of the IEEE*, 86, 615–638 (1998).
- [2] P. Bremner, Y.Liu, M. Samie, G. Dragffy, A. Pipe, G. Tempesti, J. Timmis and A. Tyrell, "SABRE: A bio-inspired fault-tolerant electronic architecture", *Bioinspiration and Biomimetics*, 8, 1–30 (2013).
- [3] L. Lamport, "The Part-Time Parliament", *ACM Transactions on Computer Systems*, 16, 133-169 (1998).
- [4] D.B. Strukov, G.S. Snider, D.R. Stewart and R.S. Williams, "The Missing Memristor Found", *Nature*, 453, 80–84 (2008)
- [5] L.O. Chua, "Memristor – the missing circuit element", *IEEE Transactions on Circuit Theory*, CT-18, 507 (1971).
- [6] E. Gale, "The Memory-Conservation Model of Memristance", [Forthcoming]
- [7] E. Gale, D. Pearson, S. Kitson, A. Adamatzky and B. de Lacy Costello, "Different behaviour seen in flexible titanium dioxide sol-gel memristors dependent on the choice of material", *Technical Digest of Frontiers in Electronic Materials*, 577-578 (2012).
- [8] E. Gale, B. de Lacy Costello and A. Adamatzky, "Observation, characterisation and modelling of memristor current spikes", *Applied Mathematics and Information Sciences*, 7, 4, 1395-1403 (2013)
- [9] E. Gale, B. de Lacy Costello and A. Adamatzky, "Boolean logic gates from a single memristor via low-level sequential logic", *Proceedings of the 12th International Conference in Unconventional Computation and Natural Computation, Lecture Notes in Computer Science*, 7956, 78–89 (2013).
- [10] Georgilas, I., Gale, E., Adamatzky, A. and Melhuish, C. UAV horizon tracking using memristors and cellular automata visual processing. *Lecture notes in Computer Science 2013*; [in press]