Rubio, A., Moll, F., Escudero, M., Zuin, S., Vourkas, I., Sirakoulis, G. Experience on material implication computing with an electromechanical memristor emulator. A: IEEE Symposium Series on Computational Intelligence. "Proceedings SSCI 2016". Atenes: IEEE Press, 2016, p. 170-175. ISBN978-1-5090-4240-1

DOI 10.1109/SSCI.2016.7850154

Experience on Material Implication Computing With an Electromechanical Memristor Emulator

S. Zuin[1], M. Escudero-López[1], F. Moll[1], A. Rubio[1]
(1) Department of Electronic Engineering
Universitat Politècnica de Catalunya (UPC), BarcelonaTech
Barcelona, Spain
antonio.rubio@upc.edu

I. Vourkas[2], G. Ch. Sirakoulis[3]
(2) Department of Electrical Engineering
Pontificia Universidad Católica de Chile, Santiago, Chile
(3) Department of Electrical Engineering
Democritus University of Thrace, Greece
iovourkas@uc.cl, gsirak@ee.duth.gr

Abstract—**Memristors are being considered as a promising emerging device able to introduce new paradigms in both data storage and computing. In this paper the authors introduce the concept of a quasi-ideal experimental device that emulates the fundamental behavior of a memristor based on an electro-mechanical organization. By using this emulator, results about the experimental implementation of an unconventional material implication-based data-path equivalent to the i4004 are presented and experimentally demonstrated. The use of the proposed quasi-ideal device allows the evaluation of this new computing paradigm, based on the resistance domain, without incorporating the disturbance of process and cycle to cycle variabilities observed in real nowadays devices that cause a limit in yield and behavior.**

Keywords—Memristor devices, Imply, Material Implication function, Unconventional Computing.

I. Introduction

The memristor (M) is a new circuit element postulated by Prof. Leon Chua in 1971 [1] extending the conventional set of well-known resistance (R), inductance (L) and Capacitance (C) elements. Conceptually, it is a passive element that holds a nonlinear relationship between flux linkage and electric charge. This implies that memristor resistance evolves with the previous history of charge to which it has been exposed exhibiting an interesting nonvolatile characteristic. Memristor-based research activity was boosted in 2008 when Hewlett Packard connected experimental nanodevices with this, for the time being, theoretical principle [2], opening a fascinating new field of research and applications. Memristor devices are a promising alternative for storage devices because they inherently behave as nonvolatile memory elements, with the corresponding impact on power consumption reduction and with the unconventional characteristic of storing data (logic states) in form of electrical resistance. It is BEOL [3] compatible, scalable, and can achieve higher density [3] levels of stored data than with conventional technology (MOS devices, 6T and dynamic cells). It is also expected that memristive devices speed could match in a near future the one of conventional CMOS devices, focusing on a scenario of hybrid technology. Together with the clear application as storage device both academy and industry are focusing efforts to orient memristors towards unconventional ways to process data, exploring non-von Neumann architectures [4bis]. Among all computing techniques with memristors that are being explored, material implication-based [5] is especially interesting because it exploits the store/process capability in a straightforward way. Material implication (in many cases indicated as logic implication and notated as $p \rightarrow q$) is based on the IMPLY function, a Boolean function declared in Table I.

TABLE I. TRUTH TABLE OF THE IMPLY FUNCTION

Case	p	q	$q' = p \rightarrow q$
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

The interest of this function when memristive circuits are considered comes from the fact that when the two logical states (1, 0) are represented by the resistance of a memristor the IMPLY function can be implemented with the simple circuit shown in Fig. 1, where p and q (Table I) correspond to the logic states of the two memristors of the circuit.

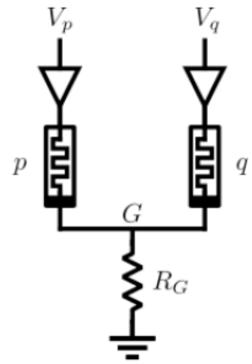

Fig. 1. Implementation of the IMPLY function, where logical states p and q correspond to the resistance state of the two indicated memristors.

In Fig. 1 the function IMPLY is performed when synchronized pulses are applied to V_p (of amplitude V_{cond1}) and

V_q (of amplitude Vcond2), when the conditions

$V_{cond1} < V_{set}$
$V_{cond2} > V_{set}$
$R_{off} > R_G > R_{on}$

are verified (R_{off}/R_{on} stand for high, low resistance state).

It can be shown that any function can be performed using the same topology (universal topology), applying an appropriate sequence of voltage pulses V_p and V_q. The IMPLY function together with the FALSE function (the reset of memristors to high resistance state) form an universal logic set.

In this work section II introduces the electrical behavior of a memristor, first from a theoretical point of view and later in the case of a modern real device, showing the deviations of that last one and commenting the problems that causes when the IMPLY function is implemented with it Section III introduces our memristor emulator, a physical device, that based on an electromechanical principle behaves as an ideal memristor, practically without any deviation and easily parameterizable. Section IV deals with details of the physical construction of the emulator and shows a demonstration assembling composed by seven emulators, an extension of the IMPLY circuit shown in Fig. 1. In this circuit the pulsed voltages of the seven devices are generated with a FPGA board and a specific level shifter circuit. Section V shows how the basic IMPLY circuit can be used to implement a generic computing data path (including the memristors memory and processing data thanks to the IMPLY function). It is shown how the 10 key instructions of the i-4004 processor [6] can be implemented with this structure showing with detail the required pulses sequence for one of them (INC A). Finally in section VI the conclusion and discussion of the work is presented.

II. Electrical Behavior of an Ideal and a Real Memristor

A. *Electrical Behavior of a Memristor.*

A memristor is a polarised (nonreversible) dipole element with a resistance between its terminals dependable of the past. Fig. 2 shows the ideal behaviour of a memristor. The figure shows the I-V characteristic of a memristor, clearly two different resistive states can be observed, R_{on} (low resistance high conductance state) and R_{off} (high resistance low conductnace state). In the example of Fig. 2 $R_{on} \approx 1100\Omega$ and $R_{off} \approx 120.000\Omega$. The transition from one resistance level to the other is given by V_{set} (transition from off to on, in the example around 7V) and V_{reset} (transition from on to off, in the example -2.5V). In the margin between these two voltage levels the device keeps at a constant nonvolatile resistance state (on or off depending on the history of voltage).

B. *Electrical behavior of a modern real memristor device*

Fig. 3 shows the corresponding I-V characteristic of a real device. In the case shown we are considering a chalcogenide memirstor manufactured by KNOWM company [7]. The different colored curves correspond just to repetitive cycles of set and reset, where it can be clearly observed that the experimental device exhibits an important time-varying variability in the four parameters of the device (R_{on}, R_{off}, V_{set} and V_{reset}).

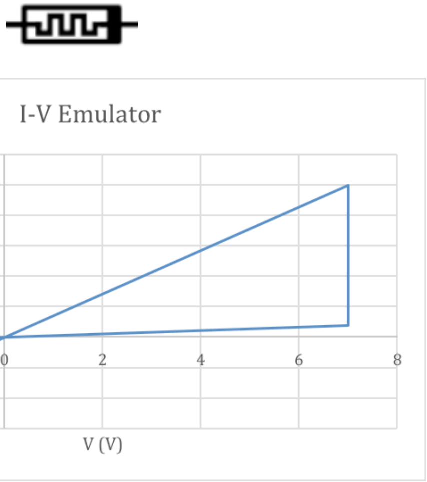

Fig. 2. Symbol and electrical characteristic of an ideal memristor. The characteristic shows the two resistance states (on and off) of the device as well as the voltages at which the state changes occur.

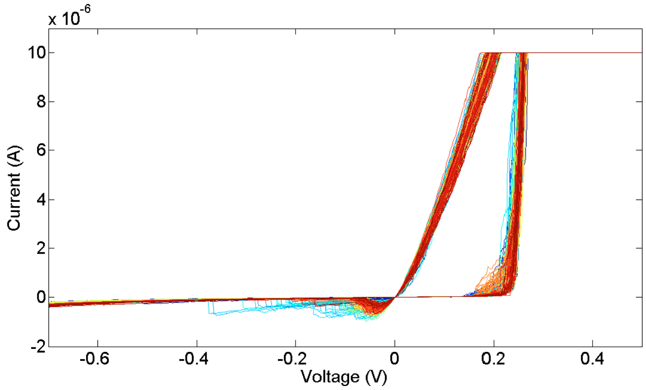

Fig. 3. Characteristic I-V for an experimenatl memristor, shwowing variability from cycle to cycle.

For the device shown in Fig. 3 (that has different parameters than the one shown in Fig. 2) the R_{off}, keeps quite stable but R_{on} exhibits a significant random variability (with a drift between 15.000Ω and 10.000Ω). Additionally V_{set} shows a significant variability from 1.7V to 2.2V) as well as V_{reset} (from -0.1V to -0.01V approximatelly).

It has been earlier analysed and shown that these state and voltages variabilities cause a drop in the yield and behavior of the IMPLY circuit in [8].

III. An Electromechanical Memristor Emulator Circuit

In order to perform an experimental evaluation of the application of the material implication principle in computing a physical quasi-ideal emulator circuitry has to verify the following conditions:

- Able to exhibit a binary (without intermediate states) and perfectly lineal resistance without variability (that is R_{on} or R_{off}).
- The voltages at which the resistance state changes (V_{set} and V_{reset}) has to be clearly defined exhibiting a neglectible cycle to cycle variability.
- It has to exhibit an inherent nonvolatility property.
- It is desiderable that the device be passive.

Fig. 4 shows the proposed emulator circuit:

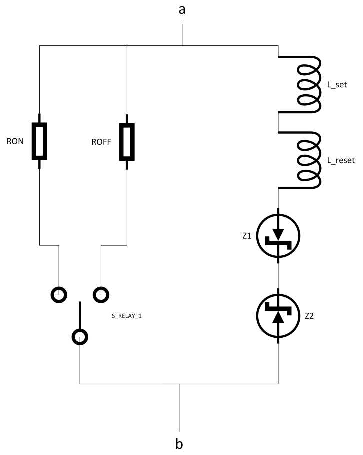

Fig. 4. Proposed emulator circuitry.

In Fig. 4 a and b represent the two terminals of the emulator circuit, so of the memristor. The basis of the proposal is an electromechanical device TQ latching (bistable) relay. The relay keeps the position of the contacts without the need of any external source of energy. The connection is changed when a pulse of current passes through the set and reset coils in one or other sense. The main circuit is the branch b-relay-resistors-a. The two resistors are conventional resistors with resistances R_{on} and R_{off}. The other branch, b-zeners-coils-a, corresponds to the set and reset functions of the memristor. The two anti-series zeners cause a potential barrier that defines the set and rest voltages. When V_{ab} is in-between the set and reset voltages (zeners are off) the branch does not conduct current. The values of the R_{on} and R_{off} of the emulated memristor are given by the two resistors and the transition voltages by (V_a-V_b criteria):

$$V_{set} = V_{z2} + V_\gamma$$
$$V_{reset} = -V_{z1} - V_\gamma$$

where V_γ is the forward voltage of diodes. It is possible to adapt the design to any set and reset voltages selecting the appropiated zener diodes. In cases where the V_{reset} is near to zero (like in the physical memristor shown in Fig. 3) it is possible to use just one zener, being in this case:

$$V_{set} = V_{z2}$$

$$V_{reset} = - V_\gamma.$$

Taking advantage of the fact that the relay device (Panasonic TQ 2 coils 2 circuits relay) has two independent circuits the second one can be used to optionally display the state of the memristor with a green led and with the inclusion of a switch and a push button and an external power supply the functionality to pre-set the emulator of any of the two resistance states (on, off). These two complementary circuits do not affect the basic circuit shown in Fig. 4. Fig. 5 shows the scheme of the complete emulator including display and pre-setting.

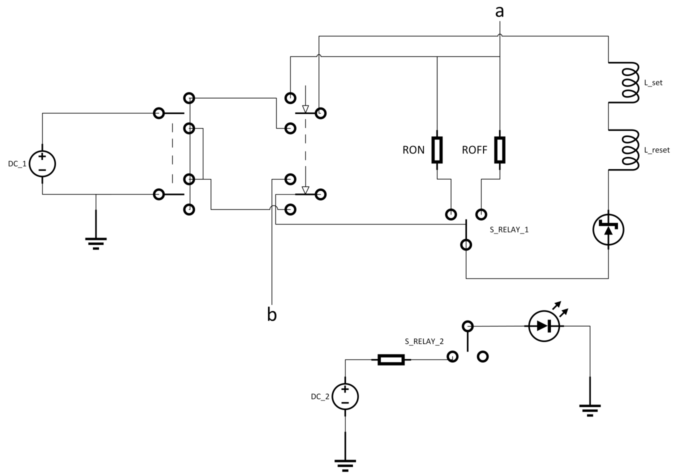

Fig. 5. Complete emulator circuitry including display and pre-setting facilities.

IV. Physical Construction of the Emulator and Description of a 7-Memristors Imply Circuit Set-Up

Fig. 6 shows the physical aspect of the emulator implementation (25×25 mm) on a printed circuit board. Detail 1 (circled) shows the two R_{on} and R_{off} resistors, 2 the zener diode, 3 the relay, 4 the on/off selector switch, 5 the setting button and 6 the state led. Fig. 7 shows I-V characteristic of the emulator.

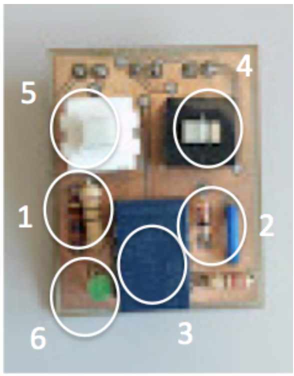

Fig. 6. Memristor emulator implementation.

We next introduce our scheme for the computation between two registers in an unconventional data path unit by using the IMPLY principle. Let’s consider two registers A and B (in order to simplify the physical implementation we will consider

2-bit registers). The structure of the circuit is universal for any set of instructions between the registers when including the necessary auxiliary memristors. Our approach is straightforward, we will consider an extended IMPLY circuit as the one shown in Fig. 1 but with 7 meristors (2 per register, RA1/RA0 for register A, RB1/RB0 for register B and 3 auxiliary (A0, A1, A2), enough for the set of instruction of section V) for storage of intermediate results (see Fig. 8).

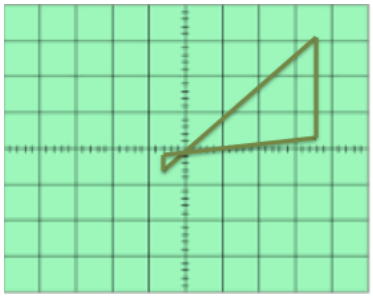

Fig. 7. I-V characteristic of the emulator (vertical axis: 25 mA/div., horizontal axis: 2 volts/div.).

The seven pulsed signals are independent and are generated with a sequencer implemneted with a FPGA board with the appropiated level shifter buffers. Fig. 9 shows the complete set-up implementation.

In the experiment the following settings have been used: R_{off} = 1kΩ, R_{on} = 100Ω, R_G = 220Ω, V_{set} = 7V, V_{cond1}= 9V, V_{cond2} = 6.5V, V_{clear} (to reset the meristors) = -2V. The FPGA generates for each of the 7 pulsed voltages, a 2-bit code of the applied voltage (V_{cond1}, V_{cond2}, V_{clear}, high impedance), decoded by the level shifter buffer board.

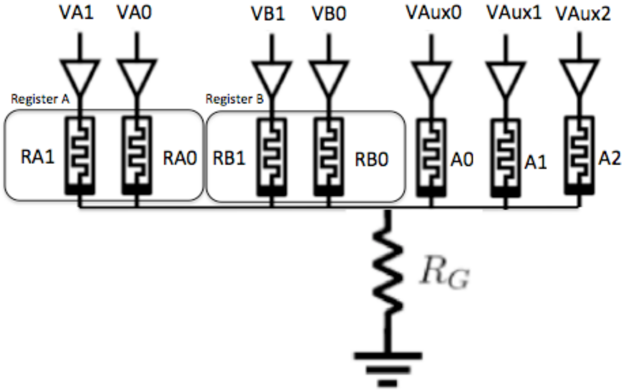

Fig. 8. Universal data path implemantation for two registers data path by using the Imply principle and the scheme of Fig. 1.

V. Application of the Universal Data Path Circuctry to an I-4004-like Instruction Set

We will consider the following set of instructions: AND (A,B), OR (A,B), NAND (A,B), NOR (A,B), XOR (A,B), R (A), R (B) INC (A), INC (B), DEC (A), DEC (B), ADD (A,B) and SUB (A,B). It has to be observed that in this structure there is no privileged register (as was the case in the i-4004 with the register A or accumulator), any register (A and B) can be source and or destinity of the instructions, R corresponds to rotate, so we consider 10 instructions.

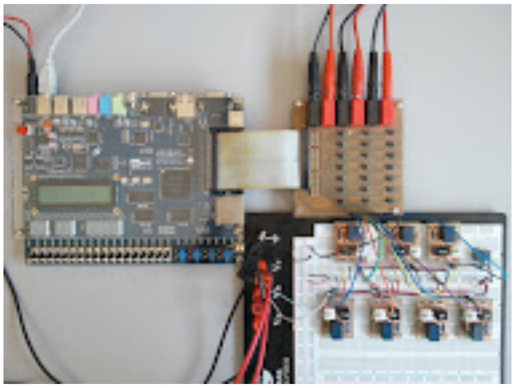

Fig. 9. Global implementation of a universal data path for two 2-bit registers. It can be identified the protoboard with 7 emulators, the drivers specific board and ALTERA D-2 FPGA board as sequence generator.

A. *Application of sequences and performances*

The electromechanical relays allow to work with pulses at a cadence of 250Hz. Table II shows the time performances of execution of each one of the 10 instructions.

TABLE II. Summary of the Instructions Requirement (@250Hz)

instruction	**Time of execution (ms)**	**# of pulses applied**	**# of internal operations**
AND	160	40	10
OR	96	24	6
NAND	192	48	12
NOR	256	64	16
XOR	352	88	22
R	112	28	7
INC	320	80	20
DEC	320	80	20
ADD	560	140	35
SUB	832	208	52

B. *The Increment instruction (INC).*

In this subsection we show in detail as a matter of example the sequences of actions and pulses corresponding to the instruction INC. In the example we apply INC to register A, not being used consequently register B, and auxiliary memristors A0, A1 and A2 the latter being where the carry is stored at the end of the operation.

The sequence of 20 required operations are given by:

False A2, A0

RA0 → A0

RA1 → A0

A0 → A2

False A0, A1

RA0 → A0

RA1 → A1

RA0 → RA1

A0 → A1

False A0

A1 → A0

RA1 → A0

False A1, RA1

A0 → A1

A1 → RA1

False A1, A0

RA0 → A0

A0 → A1

False RA0

A1 → RA0

Fig. 10 shows the 20 operations expressed as pulsed voltages on the respective memristors. The different operations are performed just modifying the sequence of pulses and not the architecture. Able to work at a pulsed frequency of 250Hz the execution time of the instructions goes from 96ms in the case of the OR to 208ms in the case of the SUB. IMPLY can be considered an interesting approach to unconventioal processors.

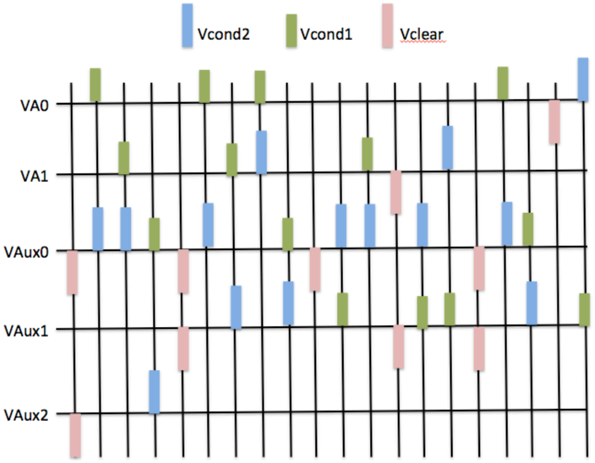

Fig. 10. Sequence of voltage levels applied to the five memristors used in the INC instruction.

VI. Conclusion

This paper has presented a laboratory implementation of a two-register data path unit by using seven physical memristor emulators in a strainghtforward implementation of the material implication or IMPLY function. The result is that it is a satisfactory and universal way to implement a set of instructions that in the case of the present article have been inspired from the basic set of instructions of the i-4004 microprocessor. The ten basic considered instructions can be implemented just modifying the sequence of pulsed voltages sequence. Evidently the sequence is independent of the data and also the architecture is simple and unique for the ten instructions and for any other that could be selected. The memristor used in the experience is an electromechanical emulator, a quasi-ideal easily customizable device based on a bistable relay. The use of such quasi-ideal memristor emulator allows to experiment without taking into account process and cycle to cycle variability limitations present in modern nanodevices. This variability introduces yield limitations and requiring in many cases memristor state refreshing. The experimentation shows the benefits of the IMPLY as a unconventional universal proccessing unit.

Acknowledgment

This work has been funded by the Spanish MINECO and ERDF through project Maragda (TEC2013-45638-C3-2-R).

References

[1] L. Chua, “Memristor: The missing Circuit Element”, IEEE Trans. Circuit Theory, vol. CT-18, pp. 507-519, September 1971.

[2] D.B. Strunov, G.S. Snider, D.R. Stewart, S. Williams, “The missing memristor found”, Letters, Nature, vol. 453, pp. 80-84, May 2008.

[3] Ed Korczynski, “Knowm first to deliver configurable artificial neural networks using bi-directional learning memristors” Semiconductor Manufacturing and Design, Solid State Technology, semimd.com.

[4] I. Vourkas and G. Sirakoulis, Memristor-Based Nanoelectronic Computing Circuits and Architectures: Foreword by Leon Chua, ser. Emergence, Complexity and Computation, New York, NY, USA: Springer-Verlag, 2015

[5] J. Borguhetti, G.S. Snider, P.J. Kuekes, J.J. Yang, D.R. Stewart, R. S. Williams, “Memristive switches enable ‘stateful’ logic operations via material implication”, Letters, Nature, vol. 464, pp. 873-876, April 2010.

[6] W. Aspray, “The Intel 4004 microprocessor”, IEEE Annals of the History of Computing, vol. 19, issue 3, pp. 4-15, August 2002.

[7] R. Colin Johnson, “Memristor computer emulates brain functions”, EE Times, 1/20/2016, knowm.org.

[8] X. Fang, Y. Tang, “Circuit analysis of the memristor stateful implication gate”, Electronic Letters, vol. 49, no. 20, pp. 1282-1283, September 2013.