Memristive Implementation of Fuzzy Logic for Cognitive Computing

Martin Klimo, Ondrej Šuch
Department of InfoComm Networks
University of Zilina
Zilina, Slovakia
e-mail: {martin.klimo, ondrej.such}@fri.uniza.sk

***Abstract*—Today's digital computers are based on three cornerstones: von Neumann architecture, Boolean algebra, and transistor as the basic element. With a history of approximately 70 years, this concept has demonstrated to be a success for algorithmic computing. However, at present, its disadvantages can be seen in real-time cognitive computing. This contribution presents the concept in which cognitive computing acts as a support for algorithmic computing, and the cognitive part is based on non-von Neumann architecture, Zadeh fuzzy logic, and resistive switch as the basic element.**

***Keywords*—resistive switch; memristive circuits; fuzzy logic; non-von Neumann architecture; cognitive computing.**

I. INTRODUCTION

Computers, as we know them, are based on three cornerstones: von Neumann architecture, Boolean logic, and transistor as a switching element. All these aspects limit computer performance to energy consumption ratio compared to the human brain. The man is not so fast in algorithmic thinking using exact terms, but he can think intuitively with the fuzzy meaning of words. Boolean values for True and False are implemented as analog values, while the brain uses spike trains. Von Neumann computer architecture has liberated programs from hardware processing circuits and stored them as data to the memory, but this transfer from the memory into the processing unit takes some time. On the other hand, the neural network in the brain is naturally a massive parallel structure, where processing and memory are both located in the same place. Since 1947, the transistor has been a principal component in computing implementation. The volatility of the transistor leads to enormous energy consumption when compared to the human brain. Non-volatile elements and spike-like computing give some promise to achieve brain efficiency in the future. However, the future does not mean replacing old architectures with new ones. It is about the coexistence of the algorithmic computing with the cognitive computing. These general trends are mirrored in recent experiments with a Central Processing Unit (CPU) support by hardware accelerators, and, in this paper, we present several examples in this regard.

Since 2008, when the HP Lab realised a memristor [1], nanotechnology has offered new ways to overcome traditional computer limits. Energy savings and higher densities can be obtained and memristor crossbar can unify the memory and the processing unit together.

Elementary circuits with resistive switches can give results for Min, Max, Avg functions in the voltage domain [2]. This idea has a significant impact on the fuzzy computer architecture. Comparing memristor-based computing with

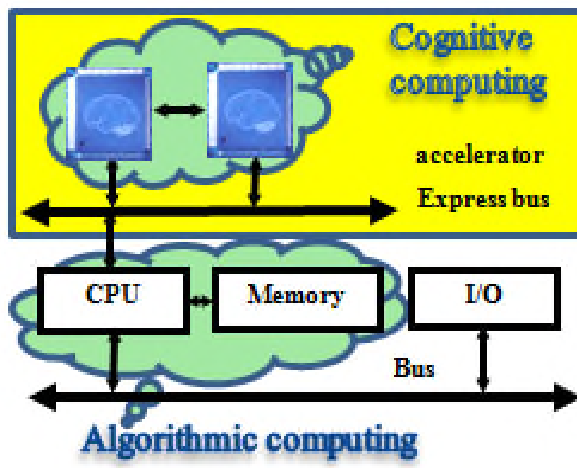

Figure 1. von Neumann architecture accelerated by non-von Neumann architecture

quantum or bio-cellular computing, the memristor technology is more mature, and several approaches (metal-oxide, ferromagnetic, grapheme-oxide) can be compared. The memristor crossbar, unlike synapse emulation and stateful Boolean implication, can provide the full range of the Zadeh logic functions in voltage input-output space. Fuzzy computing maintains continuity with classical computing where humans write programmes following the given logic requirements (implications). This is the basic approach used in fuzzy logic programming and Fuzzy Prolog-like programming languages [3], or specialised languages based on Haskell [18] e.g., Bluespec SystemVerilog [19].

The rest of the paper is structured as follows.

Section II gives the motivation for supporting algorithmic computing in von Neumann architecture by cognitive computing.

Section III supports the idea of memristive and fuzzy logic based cognitive computing [2], [10] with an experimental proof of concept by implementing elementary Zadeh fuzzy logic functions [17].

Section IV extends the proof of concept also to minterms (and analogically maxterms) in the fuzzy logic approximation by finite normal forms.

This short paper presents the progress in the memristive implementation of fuzzy cognitive computing.

 ISBN: 978-1-61208-530-2

II. Algorithmic Computing Accelerated by Cognitive Computing

Problem-solving is as old as life itself, and nature has created, by evolution, structures like the brain to do that. At the time prior to Homo sapiens, problems were less complex, but solutions had to be found in real time to save human life. Homo sapiens solved more complex problems that could be decomposed to a sequence of well-defined steps, but the process was not time critical. Among the first examples of such problem-solving would be dividing spoils in a group, growing crops and building houses. The sequence of steps that solves a problem in finite time is nowadays called an algorithm. Mainly, a history of mathematics gives an abundant supply of algorithms in numeric operations, geometry, algebra, and computer science (even for evolution). To conclude, the algorithmic solution of problems, from our perspective, is a result of human culture. Although algorithms have been implemented in the past by specific instruments (e.g., abacus, straightedge and compass, Antikythera mechanism), Turing found a universal algorithmic machine that von Neumann implemented as a digital computer. While the brain has the ability to solve problems through the topology of the neuronal network, this ability in the von Neumann computer is ascribed to the program (data). Until recently, it seemed that the von Neumann computer could solve all problems solvable by the brain. Today, we can see bottlenecks in the von Neumann architecture (transfer of data between the memory and CPU) and the advantages of programming by topology (naturally massive parallelism). As Amdahl's law [20] pointed out, even a small part of a program serially executed can suppress the advantage of the parallel connection of CPUs. Therefore, naturally massive parallel computation is needed for time-critical applications. As mentioned before, nature has found such a structure through evolution in which the circuit topology gives the program. We call it cognitive computing in this paper. This approach was applied in the ENIAC and analog computers, but it was forgotten due to the low flexibility of programming compared to the program stored as data. Even if today there is a large demand for massively parallel computing, we do not think that cognitive computing will replace algorithmic computing. From our perspective, algorithmic computing has the same importance as cognitive computing, but they have different missions. They have to be combined: algorithmic computing with von Neumann architecture should be supported by cognitive computing executed by non-von Neumann architecture (see Fig. 1). This is not a new approach, and CPUs supported by accelerators (e.g., Intel Xeon Phi coprocessor) contain a well-known architecture.

While the von Neumann architecture is a well-established concept for algorithmic computing that has been improving over the past sixty years, it is still an open question as to which concept and which inorganic technology is the best for non-von Neumann architecture. Game applications showed graphics bottlenecks, and Graphical Processing Units (GPU) have been developed to overcome this. But the potential of a GPU is much broader, and GPUs are used as accelerators for CPUs. Dell added NVIDIA GPU coprocessors and Tesla K80 to accelerate Intel Xeon CPUs in PowerEdge servers [4]; SGI has done the same in SGI Rackable Servers [5]. If general purpose GPU (GPGPU) has its roots in graphics processing, the Field-Programmable Gate Array (FPGA) has its roots in the Digital Signal Processor (DSP). As the name FPGA indicates, programming means creating digital circuit topology within a set of gates that performs the goal of the program. All gates work in a naturally massively parallel way and can provide cognitive computing in real time. The main goal of the von Neumann part of this hybrid architecture is to configure/reconfigure FPGA digital circuit topology to perform cognitive computing. Microsoft has studied FPGAs as accelerators under the project Catapult since 2010 [6], and the last results in "Configurable Cloud" architecture were published [7]. The acquisitions of Altera by Intel or Xilinx by IBM have shown movement in the same direction [8]. IBM has also developed a different kind of accelerator. They introduced TrueNorth as an accelerator [9] in which pulses are switched over the crossbar. Pulses run over the crossbar in parallel, which allows building a naturally massive parallel system from crossbars.

This paper presents the concept of a coprocessor built on fuzzy logical circuits implemented by the memristive structure. As an idea, it was published at this conference in 2012 [10]. According to Fig. 1, it can be redrawn as presented in Fig. 2.

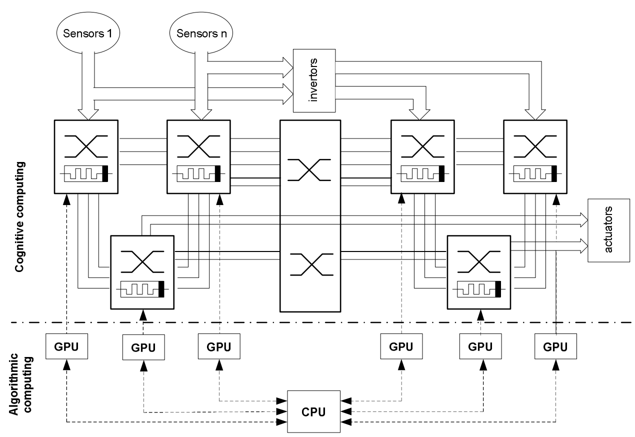

Figure 2. von Neumann architecture accelerated by memristive based cognitive computation

This paper presents an experimental proof of this concept on the level of the elementary fuzzy logic functions: minimum (Min) and maximum (Max).

III. Implementation of Fuzzy Logic by the Memristive Circuit

Hardware implementation of Min, Max functions is not new and fuzzy logic circuits based on Complementary Metal–Oxide–Semiconductor (CMOS), Field-Effect Transistor (FET) or FPGA implementations have been used before. However, there are two principal advantages of memristor-based implementations:

1. Energy supplies the circuit only over inputs, and no extra source of energy is needed to be compared with the transistor based implementations mentioned above.
2. The memristive implementation also introduces a memory function to these elementary functions. This property needs further research.

Fig. 3 shows an example of the implementation of $Y = Max(0, X)$ function using electrochemical metallization memory (ECM) resistive switches NEURO-BIT BT10001B14 [11]. To interpret this figure with respect to fuzzy logic, the input X after normalisation from <-1.5V, 1.5V> interval into <-1, 1> interval represents the difference $X = a - b$ in $y = Max(a, b);\ a, b \in$ <0, 1> function. The accuracy of the mathematical function implementation depends mainly on the switching threshold (approximately 0.2V for the measured resistive switches), and measurement repeatability.

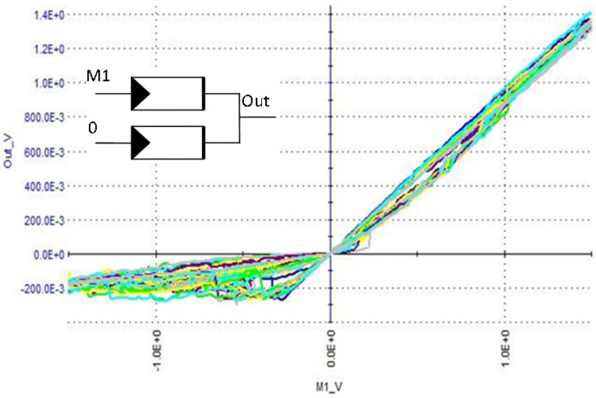

Figure 3. The input/output characteristic of the Max circuit implemented by ECM memristors NEURO-BIT BT10001B14

On the one hand, non-volatility is useful, but on the other hand, the preservation of the switch state causes a memorylessness in the input – output relationship in the Max circuit. More precisely, fuzzy logic circuits have to be assumed as state automata. Everything mentioned above regarding the implementation of Max functions is also valid for Min functions.

However, Min, Max functions allow only the building of a monotone fuzzy logic system [12]. In general, inverters are needed, but they cannot be implemented as a passive element by resistive switches. Here, we present an architecture using the property of the de Morgan´s law that a dual logic function with inverted inputs results in an inverted function. This dual system approach allows for simpler implementation because inverters are located only in the first stage of the Min/Max based circuit, and can be implemented by active elements.

IV. Implementation of the Fuzzy LOGIC function in a normal form

Our experience with memristive-based implementation of fuzzy logic functions shows [13] that in deep memristive networks, there pairs of states with no direct transition between them may occur. This is caused mainly by switching thresholds, and this phenomenon should be studied in the future. For the moment, flat memristive networks can be instrumental in fuzzy logic implementation. This flat topology means a structure corresponding to the normal form within the classical Boolean logic (BL) algebra. But disjunctive or conjunctive finite normal forms are universal approximation formulas for any BL-algebra [14].

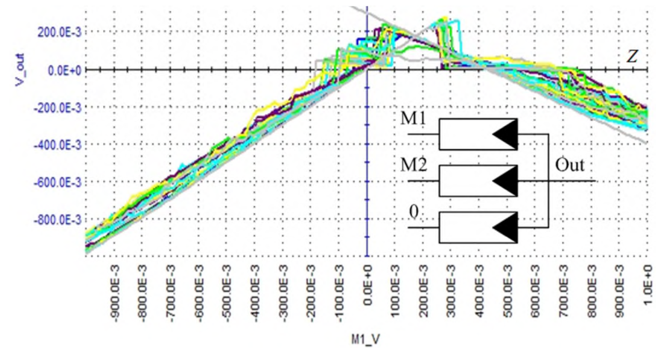

Figure 4. The input/output characteristic of the 3-input Min circuit implemented by ECM memristors NEURO-BIT BT10001B14

Max and Min functions implement disjunctions and conjunctions in the Zadeh fuzzy logic with more inputs. It is assumed again that input variables and their negations are available as inputs.

Fig. 4 shows an example of the implementation of the function

$$Y = Min(0, X1, X2).$$

As the independent input is taken $M1 = X1$, $X1 \in$ <-1, 1>, the second input is set into

$$M2 = 1 - 0.7(1 + X1),$$

and the third one is a zero reference input. The impact of the switching threshold is visible here even more so than in Figure 2.

V. Conclusion and the Future Scope

The paper sets forth the idea of how cognitive massively parallel computing based on the Zadeh fuzzy logic can be implemented using memristive circuits. These results support an idea of cognitive computing based on fuzzy logic accelerators implemented via memristive circuits.

Looking at Fig. 3 and Fig. 4, we can see the importance of the non-deterministic behaviour of memristors and the influence of the switching threshold. The impact of these properties on the accuracy of fuzzy memristive computing needs further research. Another extension of the field of research would be the large area of non-fuzzy memristive computing [15], and applications of memristive circuits [16].

Acknowledgment

APVV-14-0560 grant funded this research.

REFERENCES

[1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Wiliams, “The missing memristor found,” Nature 453, pp. 80-83, May 2008, doi:10.1038/nature06932.

[2] M. Klimo and O. Such, “Memristors can implement fuzzy logic,” http://arxiv.org/abs/1110.2074.

[3] S. Guadarrama, S. Muñoz, and C. Vaucheret, **“**Fuzzy Prolog: a new approach using soft constraints propagation,” Fuzzy Sets and Systems,” vol. 144, Issue 1, pp. 127-150, May 2004.

[4] S. Igbal, M. Deshmukh, and N. Dandapanthula, “Game-changing Extreme GPU computing with The Dell PowerEdge C4130,” A Dell Technical White Paper, Dell HPC Engineering, May 2015.

[5] “SGI with NVIDIA GPU Accelerators Complete Solutions”, Data Sheet, www.sgi.com, Accessed September 30, 2016.

[6] E. S. Chung, J. C. Hoe, and K. Mai, “CoRAM: An In-Fabric Memory Architecture for FPGA-based Computing,” International Symposium on Field-Programmable Gate Arrays (FPGA), Monterey, CA, pp. 97-106, Feb. 2011, doi:10.1145/ 1950413.1950435

[7] Project Catapult, https://www.microsoft.com/en-us/research/ project/project-catapult/, Accessed October 17, 2016.

[8] G. Lacey, G. Taylor, and S. Areibi, “Deep Learning on FPGAs: Past, Present, and Future”, arXiv:1602.04283 [cs.DC].

[9] P. A. Merolla et al., “A million spiking-neuron integrated circuit with a scalable communication network and interface,” Science, vol. 345, Issue 6197, pp. 668-673, Aug. 2014, doi:10.1126/ science.1254642.

[10] M. Klimo and O. Such, “Fuzzy Computer Architecture Based on Memristor Circuits,” Proc. FUTURE COMPUTING 2012, pp. 84-87, July 2012, Nice, France, ISBN: 978-1-61208-217-2.

[11] “Neuro-bit user manual,” http://www.bioinspired.net/, Accessed April 19, 2016.

[12] M. Blum, R. Rue, and K. Yang, “On the complexity of MAX/MIN/AVRG circuit,” Carnegie Mellon Technical Report CMU-CS-02-110, 2002, http://repository.cmu.edu/compsci/150

[13] L. Nielen et al., “Memristive Sorting Networks Enabled by Electrochemical Metallization Cells,“ International Journal of Unconventional Computing, vol. 12, No. 4, pp. 303-317, 2016, ISSN: 1548-7199 (print), ISSN: 1548-7202 (online).

[14] I. Perfilieva, “Normal forms for fuzzy logic functions,” Proc. 33rd Int. Symp. Multi-valued logic, ISMVL´03, pp. 59–64, 2003, doi:10.1109/ISMVL.2003.1201385.

[15] I. Vourkas and G. Ch. Sirakoulis, “Memristor-Based Nanoelectronic Computing Circuits and Architectures”, Springer, 2016, ISBN 978-3-319-22646-0.

[16] A. Chens, J. Hutchby, V. Zhirnov, and G. Bourianoff, editors, “Emerging Nanoelectronic Devices”, Wiley, 2015, ISBN 978-1-118-44774-1.

[17] L. A. Zadeh, “Fuzzy logic and its application to approximate reasoning,” In Information Processing 74, Proc. IFIP Congr. 1974 (3), pp. 591–594.

[18] S. Thompson, “Haskell: The Craft of Functional Programming,” Addison-Wesley, 1996.

[19] Y. S. Shao and D. Brooks, “Research infrastructures for hardware accelerators,” Synthesis Lectures on Computer Architecture, 2015, Vol. 10, No. 4, pp. 1-99, Morgan & Claypool Pub, doi: 10.2200/S00677ED1V01Y201511CAC034.

[20] G. M. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," AFIPS Conf. Proc. (30), 1967, pp.483–485, doi:10.1145/1465482.1465560