The Future is Analog: Energy-Efficient Cognitive Network Functions over Memristor-Based Analog Computations

Saad Saleh
University of Groningen
Groningen, Netherlands
s.saleh@rug.nl

Boris Koldehofe
Technische Universität Ilmenau
Ilmenau, Germany
boris.koldehofe@tu-ilmenau.de

Abstract

Current network functions build heavily on fixed programmed rules and lack capacity to support more expressive learning models, e.g. brain-inspired *Cognitive* computational models using neuromorphic computations. The major reason for this shortcoming is the huge energy consumption and limitation in expressiveness by the underlying TCAM-based digital packet processors. In this research, we show that recent emerging technologies from the analog domain have a high potential in supporting network functions with energy efficiency and more expressiveness, so called cognitive functions. We propose an analog packet processing architecture building on a novel technology named Memristors. We develop a novel analog match-action memory called Probabilistic Content-Addressable Memory (pCAM) for supporting deterministic and probabilistic match functions. We develop the programming abstractions and show the support of pCAM for an active queue management-based analog network function. The analysis over an experimental dataset of a memristor chip showed only 0.01 fJ/bit/cell of energy consumption for corresponding analog computations which is 50 times less than digital computations.

CCS Concepts

• **Networks** → **In-network processing**; **Network protocol design**; • **Hardware** → **Emerging architectures**; **Networking hardware**; **Impact on the environment**.

Keywords

Network functions, Energy efficiency, Memristors

HotNets'23, November 28-29, 2023, Cambridge, Massachusetts

ACM ISBN 979-8-4007-0415-4/23/11.
https://doi.org/10.1145/3626111.3628192

ACM Reference Format:
Saad Saleh and Boris Koldehofe. 2023. The Future is Analog: Energy-Efficient Cognitive Network Functions over Memristor-Based Analog Computations. In *Proceedings of The 22nd ACM Workshop on Hot Topics in Networks (HotNets'23).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3626111.3628192

1 Introduction

The Internet relies heavily on programmable network functions, like congestion control [6], load balancing [32], traffic analysis [45, 47, 51] or packet scheduling [61] in order to establish communication links and deliver services across the network. Despite line-rate performance, the programming models of network functions are still based upon fixed programmed rules and cannot support more expressive learning models, like brain-inspired Cognitive models using neuromorphic computations [9, 49]. The major reasons for this shortcoming are the huge energy consumption and limited match-action possibilities in the underlying Ternary Content-Addressable Memory (TCAM)-based packet processors [20, 24]. The continuous data movements between the storage and computational units consume significant amount of energy, e.g., upto 90% for TCAM [23, 41](Figure 1). Moreover, TCAM supports only digital outputs (match or mismatch) without any possibility of computing an analog output (partial match) required for cognitive models. These shortcomings require the use of novel technologies that can support *analog* computations with colocalized computation and storage, and one such technology is the *Memristor* [48, 57].

A memristor is a non-volatile, nanoscale and programmable component with colocalized computation and storage. In this research, we show that memristor-based components allow for a transformation of the traditional TCAM-based digital match-action process to an analog match-action process through the design of a Probabilistic Content-Addressable Memory (pCAM). The analog match-action process can be programmed for both digital (deterministic) and analog (probabilistic) outputs based upon the closeness of the match process for incoming search query against the locally stored policies. The analog match-action process can use the analog

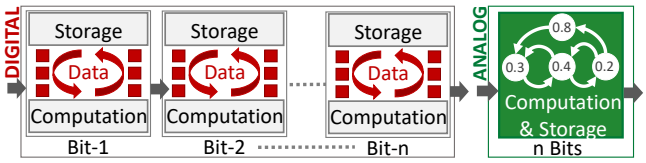


Figure 1: Energy savings by colocalizing computation and storage in analog computations vs digital computations.

features for network functions building on cognitive models, referred as *cognitive network functions*, to compute an analog output. For example, the analog Active Queue Management (AQM) network function can incorporate the higher-order derivatives of sojourn times and buffer sizes in order to compute an analog Packet Drop Probability (PDP).

Enabling the transition from the digital to analog in-network computations requires an understanding of the underlying memristor-based analog match-action process. It motivates the first research question, "How can analog in-network computations be used for supporting cognitive network functions at packet processors?". It requires an understanding of the packet processing pipeline for supporting analog computations considering the precision requirements of various network functions. This leads to the next research question, "How can the memristor-based analog components be integrated in the current digital packet processing architectures?". Moreover, the programming abstractions and analysis of energy consumption are still limited to the network functions building on digital computations. It motivates the last research question, "What would be the programming abstractions and energy efficiency gains for analog network functions?". It requires the proof of concept for a baseline network function, like AQM, by using a real-world memristor dataset.

Contributions and Research Findings. In this research, we make the pioneer effort in supporting cognitive network functions through memristor-based analog in-network computations. Our major contributions are as follows; (1) Proposition of a novel analog match-action process over pCAM to support analog computations in packet processors, (2) Development of an analog packet processing architecture for supporting energy-efficient cognitive network functions, (3) Development of the programming abstractions for a base-line analog network function, i.e., AQM, in packet processors, (4) Proof-of-concept (i.e., AQM) for packet processors by using the dataset of a Nb-doped $SrTiO_3$ memristor chip, and estimation of the energy efficiency and performance gains for analog packet processors. The Nb-doped $SrTiO_3$ memristor lowered energy consumption by a factor of 50 compared to digital packet processing. The analysis over an AQM-based network function showed an efficient queue management by keeping the packet delays within the programmed latency bounds due to the use of analog higher-order derivatives of sojourn times and buffer sizes.

Paper Organization. Sec-2 presents the limitations of TCAMs and introduces the memristors. The research questions have been discussed in Sec-3. The proposed analog packet processing is presented in Sec-4. The proof-of-concept for an AQM-based function and performance analysis have been shown in Sec-5 and Sec-6, respectively. Sec-7 summarizes the related work, and Sec-8 concludes the paper.

2 From Digital to Analog Technology

Traditional in-network functions build on the TCAM architecture for enabling line-rate packet processing. In this section, we refer to limitations and explain how memristor-based analog technology can help alleviate these shortcomings.

Limitations of TCAMs. The high-end packet processors rely on TCAMs for matching packet headers against rules defining network policies in a single clock cycle. Despite line-rate performance, TCAM consumes huge amount of energy due to the continuous data movements between the computational and storage units. Moreover, TCAM offers limited amount of space due to its digital storage and processing. TCAM is programmable in the digital domain only and there is no way to express probable matches, e.g., a match with a given probability. This, however, is crucial when dealing with analog functions. The major reason for these limitations is the strong reliance on the traditional transistor-based technology in TCAM. This technology gives remarkable precision, but it is volatile, large scale in size and requires separate computational and storage units for in-network computations.

Memristors. Memristors are non-volatile and nanoscale energy-efficient components which can be programmed to store analog data (i.e., network policies) in form of a state S, mostly represented as a physical property *Resistance* [7, 8]. Built upon the principles of in-memory computing [25, 50], the read/search operation can supply inputs (i.e., incoming packet header fields) to these memristors in order to receive an output which is a function of S. Unlike the transistor-based components, memristor is the only component which can provide different states against the same analog input depending upon the programmed initial state, as shown in Figure 2. The application of an analog input (in *Computation-1*) can yield either $S^1_{h_1}$ or $S^1_{l_1}$ depending upon the programmed initial state S^1_1 or S^1_m. Moreover, reprogramming the initial states to new analog states (S^n_1 or S^n_m) can generate a new state machine as shown in *Computation-n*. The input/output response of the memristor is shown in the function $AnalogCompute()$.

```
1 function AnalogCompute(){
2     Output^Analog = S_x^y × Input^Analog
3     ∀y ∈ [1,n]      \\"n" state machines,
4     ∀x ∈ [1,m]      \\"m" states inside a state machine}
```

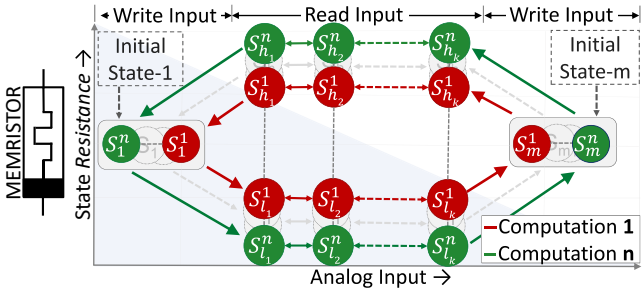


Figure 2: The analog state machine of the memristor.

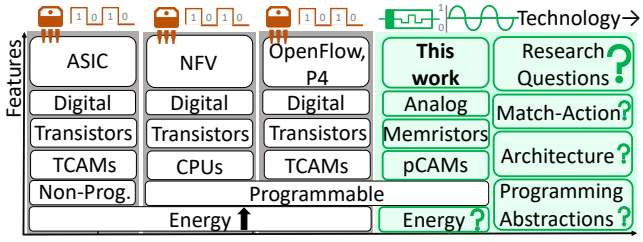


Figure 3: Taxonomy of packet processing architectures.

3 Problem Statement & Research Questions

Our research focuses on the given research problem;

"Given the huge energy requirement in using brain-inspired cognitive models inside traditional network functions, how can analog computations be integrated and support packet processors to become more energy efficient?"

The use of memristor-based analog computations requires an understanding of the match-action process, and the integration and programmability inside packet processors since it resembles a fundamentally new and different technology (Figure 3) [5, 18, 34].

Research Question-1

How can analog in-network computations support cognitive network functions at the packet processors?

In this research, we study the use of analog computations for modeling the *match-action* process in the analog domain. A typical match process takes the packet header fields of the input packet and calculates the difference between the input and the stored contents, called as Hamming distance. The use of digital computations severely limits the Hamming distance calculation because the TCAMs round the match results to the nearest logic level. The TCAM output is always discrete, i.e. a match or mismatch. There is no possibility to express a partial match. Contrary, an analog match-action process can enable programmability of the digital logic levels and support an additional range of analog logic levels. For example, for a stored policy of 2.5 V, the programmer can specify the range of deterministic matches i.e., Match(logic-1): [2.4-2.6] V, Mismatch(logic-0): [0-1.5] V, and probable matches i.e., analog (0-1): (1.5-2.4) V. The analog match-action process supports cognitive functions by providing diverse analog outputs (probable matches) in addition to the digital outputs for identifying the closely matching stored policies for an incoming query with zero matches.

Research Question-2

How can memristor-based analog components be integrated in the current packet processing architectures?

The incorporation of memristor-based components in the current packet processing architectures is a two step process; (1) Development of an analog match-action memory, (2) Integration of the analog match-action memory into the packet processing architecture. The traditional TCAM memory supports only digital inputs and outputs. Building on prior findings [30, 40], we propose the development of a programmable memristor-based pCAM memory for supporting the digital and analog outputs at the packet processors. In the next step, pCAM can be integrated into the current packet processing architecture. However, the match output can lose its precision depending upon the line losses, signal strength and interference from the neighboring components. It requires an understanding of the network functions depending upon their precision requirements. For example, network functions like IP lookup and IP firewall have high thresholds for precision than the network functions like AQM, traffic analysis, etc. Hence, an understanding of the packet processing pipeline is required in order to integrate the digital and analog components (TCAMs and pCAMs) for various network functions.

Research Question-3

What are the programming abstractions and energy efficiency gains for analog network functions, like AQM?

The programming of analog network functions requires a novel programming abstraction due to the use of analog hardware technology i.e., memristors. All prior network devices like switches, FPGAs, etc. allow the programmability of the network function at the application layer and leave the mapping of hardware resources to the underlying compiler resulting in resource mapping and energy efficiency issues [22, 37, 55]. However, the analog hardware can allow the programmer to specify the hardware function from the application layer for efficient mapping of network resources, and making colocalized algorithms with limited data movements between the different computational units. It requires an elaborate study on the energy consumption of these computations for real-world memristors in order to verify the energy efficiency claims (shown below) for network functions [3].

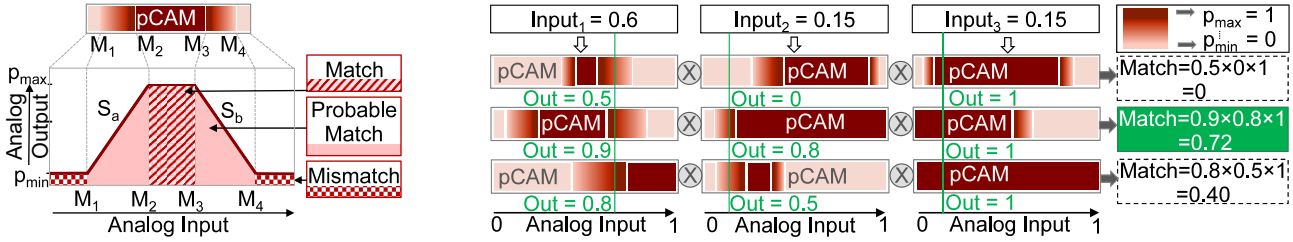


Figure 4: Abstract working operation of (a) pCAM, and (b) pCAM-based analog match-action process.

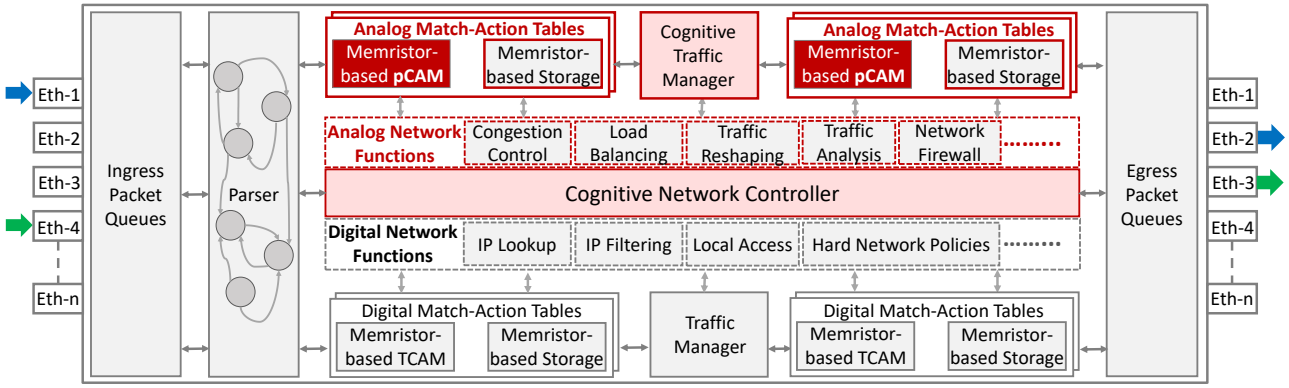


Figure 5: The proposed memristor-based cognitive packet processing architecture.

> "Analog systems.. use 10,000 times less power than comparable digital systems." C. Mead (1990)[35]
> "Energy dissipation.. The factor-of-1000 opportunity requires us to make algorithms more local, so that we do not have to ship the data all over the place." C. Mead (2022)[36]

4 Proposed Analog Match-Action Processing

In this section, we present the proposed analog packet processing architecture building upon a novel pCAM memory.

Proposed Memristor-based pCAM. An analog computation is characterized by enabling the use of continuous logic levels instead of the discrete logic levels. Memristors have shown the support of analog computations in the Content-Addressable Memory (CAM) at a circuit level [14, 28, 29, 31, 56]. Building on [30, 40, 44], we propose an analog match-action process on top of an analog pCAM memory. The role of pCAM is to take the input queries in form of analog signals and compute the probability of a match between stored and supplied contents. A typical match process inside a single pCAM cell maps the analog input to a maximum output (p_{max}) for a match, minimum output (p_{min}) for a mismatch and in between the maximum and minimum outputs for a probable match based upon the programmed parameters, as shown in Figure 4(a). The programmable parameters M_1-M_4 specify five different regions with deterministic and probabilistic matches, and output is defined by the slope function S_a and S_b for probabilistic matches. For multistage match-action process, multiple pCAM cells can be combined in series to obtain the product of deterministic and probabilistic matches at the output, as shown in Figure 4(b).

Proposed Packet Processing Architecture. The proposed memristor-based analog packet processing architecture for supporting cognitive network functions is shown in Figure 5. It uses the pCAM-based match-action memory for providing both deterministic and probabilistic matches. The digital domain enables high precision, however, lacks expressiveness, while the analog domain enables energy-efficient analog computations at the cost of precision. In both domains, memristors play a significant role to reduce the energy footprint. For example, prior researches [42, 43, 46] demonstrated high energy savings for memristor-based TCAMs.

Network functions building over cognitive models, like AQM, load balancing, etc., require probabilistic and deterministic matches, and can be offloaded to the pCAM-based analog computational components. The splitting of network functions into the digital and analog domains requires a cognitive network controller. The controller programs the memristor-based pCAMs and TCAMs based upon the requirements of the network functions. The proposed architecture contains the ingress and egress queues for acting as packet buffers, and

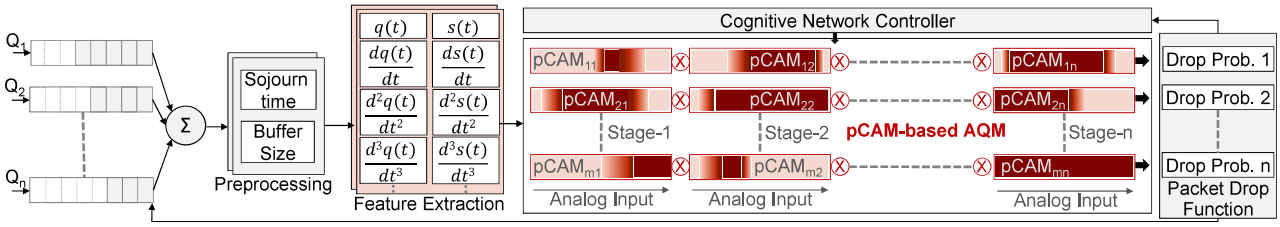


Figure 6: pCAM-based analog AQM for the memristor-based cognitive traffic manager.

memristor-based storage for storing actions. It also contains a parser to extract the required packet header fields and forward them to the respective analog and digital computational units.

5 Proof of Concept: Analog AQM

Network systems use AQM algorithms, like CODEL [38], RED [10] or PIE [39] in order to keep an optimal queue size by selectively dropping packets. This allows counteracting problems like Bufferbloat, congestion, buffer overflows and unfairness [1, 60]. AQM algorithms can be implemented inside match-action tables [21]. This, however, comes at a significant cost of energy and system resources [26, 27].

pCAM-based AQM. The analog match-action process makes it possible to support line-rate queue management at lower energy cost in packet processors, as shown in Figure 6. The proposed AQM collects the statistics of sojourn time and buffer size. Later, it computes additional features, like first, second and third-order derivatives of sojourn time and buffer size, in-order to estimate the network congestion. The additional features are computed by the analog components [52, 63]. The first-order derivative gives an insight into the rate of increase of sojourn time and buffer size. Based upon the increase, high priority traffic gets lower drop probability as compared to the low priority traffic. The second-order derivative provides an insight into the change of first-order derivative for accurate PDP estimation and adaptation of AQM parameters. The third-order derivative provides information about the bursty periods of the network traffic. The collected features are passed through a series of pCAM-based processing stages which contain the programmed feature ranges. The final output of pCAMs is the PDP for AQM.

Programming Abstractions. pCAM-based AQM can be programmed by specification of the eight pCAM programmable parameters (*prog_pCAM()*, Figure 4(a)). It's possible to specify the I/O response, and controller can map it to *prog_pCAM()* by using the function *pCAM()*. The processing pipeline is enlisted in the function *AQM()*. The analog match-action table, *analogAQM()*, incorporates the *read*, *action* and *output*. The *output* is the raw analog voltage, and it can be used directly (like PDP for AQM) or indirectly by fetching the stored actions related to the given output. For AQM, *action* updates the pCAM parameters M_1-M_4, S_a, S_b, p_{max} and p_{min} through function *update_pCAM()*.

```
function prog_pCAM(){
    program(M1, M2, M3, M4, Sa, Sb, pmax, pmin);}
```

```
function pCAM(input, output){
    if (input ≤ M1 || input ≥ M4)
        output=pmin;
    elseif input > M3
        output=Sb(input) + (M4pmax − M3pmin)/(M4 − M3);
    elseif input < M2
        output=Sa(input) + (M2pmin − M1pmax)/(M2 − M1);
    else
        output=pmax;}
```

```
function AQM(){
    drop = pipeline{
            pCAM(sojourn_time),          // Stage-1
            pCAM(d/dt(sojourn_time)),    // Stage-2
            ...
            pCAM(d³/dt³(buffer_size))}} // Stage-n
```

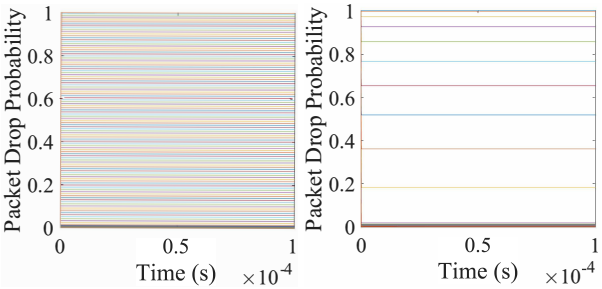
```
table analogAQM{
    read{
        sojourn_time;
        d/dt(sojourn_time);
        ...
        d³/dt³(buffer_size);}
    output{
        AQM();}
    action{
        update_pCAM();}}
```

```
action update_pCAM(id, parameter[1:8]){
    set_field(prog_pCAM.sojourn_time, M[1:8]);
    set_field(prog_pCAM.d/dt(sojourn_time), M[1:8]);
    ...
    set_field(prog_pCAM.d³/dt³(buffer_size), M[1:8]);}
```

6 Preliminary Results

In this section, we analyze the energy consumption and queue management of the analog AQM network function.

Energy Consumption. The energy analysis of the pCAM-based AQM was conducted by using real world dataset of Nb-doped $SrTiO_3$ memristor chip [12, 13]. The analysis showed that pCAM has maximum power consumption of 0.16 nJ/bit/-cell. However, pCAM also provides a range of states which show very low energy consumption. The lowest energy consumption states require only about 0.01 fJ/bit/cell of energy.



(a) Input ∈ [1,4] V. **(b) Input ∈ [-2,1] V.**

Figure 7: Analog AQM outputs for the memristor dataset.

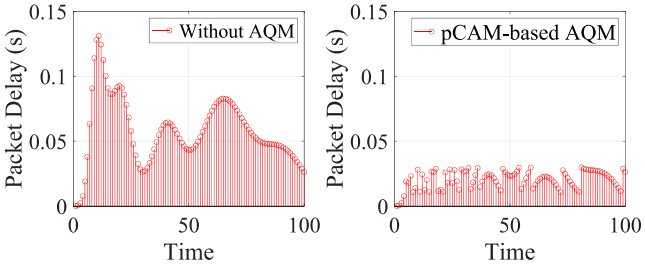


Figure 8: Queue management by using the analog AQM.

The major reason for this low energy consumption is the use of analog memristive states and colocalization of the computation and storage inside the memristor. In comparison to state-of-the-art digital computations, the analog computations proved to be at least 50 times more energy efficient (Table 1).

Queue Management. The analog output of the pCAM-based AQM for the memristor dataset is shown in Figure 7. The PDP ranges from 0 to 1 depending upon the analog input (sojourn time and buffer size) mapped to hardware voltages (DACs [58])). The performance of pCAM-based AQM was analyzed by simulating the network queues with the Poisson distributed network flows, as shown in Figure 8. pCAM has been programmed to maintain an average delay of 20 ms with a maximum deviation of 10 ms. The results show that the packet delays keep increasing sharply without AQM. However, the use of pCAM-based AQM can manage the congestion by observing the rate of change of packet delays and selectively dropping the packets based upon the congestion.

7 Literature Review

The support of cognitive computational models is a fundamental requirement of packet processors. Saleh et al. [46] have shown the self-learning capabilities and energy savings for network functions by using memristor-based cognitive models in packet processors. In [64], Zulfiqar et al. highlight the throughput and latency compromises by continuous data movements between the data plane and control plane. The authors suggest the development of a match-compute abstraction for line-rate network functions in the data plane. Shrivastav [53, 54] showed the limited match-action possibilities in packet processors. The author proposed multi-dimensional match-action tables and stateful multi-pipeline packet processors for supporting more expressive network functions.

Table 1: Performance comparison of Transistors(T)/ Memristors(M)-based Digital(D)/Analog(A) computations.

Researches	[2]	[19]	[42]	[33]	[11]	[4]	[62]	[59]	pCAM
Computation (D/A)	D	D	D	D	D	D	D	D	**A**
Technology (T/M)	T	T	M	M	M	M	M	M	**M**
Latency (ns)	1	1.9	1	0.29	0.18	1	2.3	8	**1**
Energy (fJ/bit)	0.58	1.98	1-16	1.04	1.2	2.15	3	7.4	**0.01**

Memristors have shown improvements in energy savings, space and throughput for the digital packet processing due to the non-volatility and nanoscale size [42, 43]. The network functions, like regular expression matching, showed an improvement in throughput by 12 times (upto 47.2 Gbps) by using memristor-based TCAMs instead of the FPGAs [15–17]. Considering the resource scarcity issues, recent researches [14, 28–31, 40, 56] have focused on the development of analog CAMs and differential CAMs to support deterministic matches for functions like decision trees. These researches have shown huge savings in space (upto 18 times) and energy (upto 10 times) by moving to the memristor-based analog computations. However, memristors have not been used for deterministic and probabilistic matches at packet processors.

8 Conclusion and Future Work

In this paper, we presented the use of a novel memristor-based analog technology for supporting cognitive network functions inside packet processors. We proposed an analog packet processing architecture built upon a novel memristor-based analog pCAM memory, and developed the programming abstractions for a baseline AQM-based network function. The energy analysis of the analog computations based upon the experimental dataset of Nb-doped $SrTiO_3$ memristor showed only 0.01 fJ/bit/cell of energy consumption. In future, we would focus on the understanding of (1) precision and diversity of the analog match-action process including modeling of non-linear match functions in the data plane; (2) cognitive models deployment, e.g., neuromorphic computations, for self-learning line-rate network functions in the data plane.

Acknowledgments

The authors would like to acknowledge the financial support of the CogniGron research center and the Ubbo Emmius Funds (University of Groningen).

References

[1] Vamsi Addanki, Maria Apostolaki, Manya Ghobadi, Stefan Schmid, and Laurent Vanbever. 2022. ABM: Active Buffer Management in Datacenters. In Proceedings of the SIGCOMM Conference. ACM, 36–52. https://doi.org/10.1145/3544216.3544252

[2] Igor Arsovski, Travis Hebig, Daniel Dobson, and Reid Wistort. 2013. A 32 nm 0.58-fJ/bit/Search 1-GHz Ternary Content Addressable Memory Compiler Using Silicon-Aware Early-Predict Late-Correct Sensing with Embedded Deep-Trench Capacitor Noise Mitigation. IEEE Journal of Solid-State Circuits 48, 4 (2013), 932–939. https://doi.org/10.1109/jssc.2013.2239092

[3] Peter Blouw, Xuan Choo, Eric Hunsberger, and Chris Eliasmith. 2019. Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware. In Proceedings of the Annual Neuro-Inspired Computational Elements Workshop. ACM, 1–8. https://doi.org/10.1145/3320288.3320304

[4] Venkataramesh Bontupalli, Chris Yakopcic, Raqibul Hasan, and Tarek M Taha. 2018. Efficient Memristor-Based Architecture for Intrusion Detection and High-Speed Packet Classification. ACM Journal on Emerging Technologies in Computing Systems 14, 4 (2018), 1–27. https://doi.org/10.1145/3264819

[5] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: Programming Protocol-Independent Packet Processors. ACM SIGCOMM Computer Communication Review 44, 3 (2014), 87–95. https://doi.org/10.1145/2656877.2656890

[6] Xiaoqi Chen, Shir Landau Feibish, Yaron Koral, Jennifer Rexford, Ori Rottenstreich, Steven A Monetti, and Tzuu-Yi Wang. 2019. Fine-grained Queue Measurement in the Data Plane. In Proceedings of the International Conference on Emerging Networking Experiments And Technologies. ACM, 15–29. https://doi.org/10.1145/3359989.3365408

[7] Leon Chua. 1971. Memristor-The Missing Circuit Element. IEEE Transactions on Circuit Theory 18, 5 (1971), 507–519. https://doi.org/10.1109/tct.1971.1083337

[8] Leon O Chua, Ronald Tetzlaff, and Angela Slavova. 2022. Memristor Computing Systems. Springer. 298 pages. https://doi.org/10.1007/978-3-030-90582-8

[9] Mike Davies. 2019. Benchmarks for Progress in Neuromorphic Computing. Nature Machine Intelligence 1, 9 (2019), 386–388. https://doi.org/10.1038/s42256-019-0097-1

[10] Sally Floyd and Van Jacobson. 1993. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking 1, 4 (1993), 397–413. https://doi.org/10.1109/90.251892

[11] Krishna Gnawali and Spyros Tragoudas. 2021. High-Speed Memristive Ternary Content Addressable Memory. IEEE Transactions on Emerging Topics in Computing 10, 3 (2021), 1349–1360. https://doi.org/10.1109/TETC.2021.3085252

[12] Anouk S. Goossens and Tamalika Banerjee. 2023. Tunability of Voltage Pulse Mediated Memristive Functionality by Varying Doping Concentration in $SrTiO_3$. Applied Physics Letters 122, 3 (2023), 034101. https://doi.org/10.1063/5.0124135

[13] Anouk S. Goossens, A Das, and T Banerjee. 2018. Electric Field Driven Memristive Behavior at the Schottky Interface of Nb-doped $SrTiO_3$. Journal of Applied Physics 124, 15 (2018), 152102. https://doi.org/10.1063/1.5037965

[14] Catherine Graves, Can Li, Kivanc Ozonat, and John Paul Strachan. 2022. Hardware Accelerator with Analog-Content Addressable Memory (a-CAM) for Decision Tree Computation. US Patent App. 17/071,924.

[15] Catherine E. Graves, Can Li, Xia Sheng, Wen Ma, Sai Rahul Chalamalasetti, Darrin Miller, James S. Ignowski, Brent Buchanan, Le Zheng, Si-Ty Lam, Xuema Li, Lennie Kiyama, Martin Foltin, Matthew P. Hardy, and John Paul Strachan. 2019. Memristor TCAMs Accelerate Regular Expression Matching for Network Intrusion Detection. IEEE Transactions on Nanotechnology 18 (2019), 963–970. https://doi.org/10.1109/tnano.2019.2936239

[16] Catherine E. Graves, Wen Ma, Xia Sheng, Brent Buchanan, Le Zheng, Si-Ty Lam, Xuema Li, Sai Rahul Chalamalasetti, Lennie Kiyama, Martin Foltin, John Paul Strachan, and Matthew P. Hardy. 2018. Regular Expression Matching with Memristor TCAMs. In Proceedings of the International Conference on Rebooting Computing (2018). IEEE, 1–11. https://doi.org/10.1109/icrc.2018.8638603

[17] Catherine E. Graves, Wen Ma, Xia Sheng, Brent Buchanan, Le Zheng, Si-Ty Lam, Xuema Li, Sai Rahul Chalamalasetti, Lennie Kiyama, Martin Foltin, John Paul Strachan, and Matthew P. Hardy. 2018. Regular Expression Matching with Memristor TCAMs for Network Security. In Proceedings of the International Symposium on Nanoscale Architectures (2018). IEEE/ACM, 65–71. https://doi.org/10.1145/3232195.3232201

[18] Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. 2015. Network Function Virtualization: Challenges and Opportunities for Innovations. IEEE Communications Magazine 53, 2 (2015), 90–97. https://doi.org/10.1109/mcom.2015.7045396

[19] Isamu Hayashi, Teruhiko Amano, Naoya Watanabe, Yuji Yano, Yasuto Kuroda, Masaya Shirata, Katsumi Dosaka, Koji Nii, Hideyuki Noda, and Hiroyuki Kawai. 2013. A 250-MHz 18-Mb Full Ternary CAM With Low-Voltage Matchline Sensing Scheme in 65-nm CMOS. IEEE Journal of Solid-State Circuits 48, 11 (2013), 2671–2680. https://doi.org/10.1109/jssc.2013.2274888

[20] IEA. 2022. Data Centres and Data Transmission Networks. IEA. Retrieved Oct 20, 2023 from https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

[21] Intel®. 2023. Tofino 2 12.8 Tbps, Reduced Stage, 4 Pipelines. Intel. Retrieved Oct 20, 2023 from https://www.intel.com.au/content/www/au/en/products/details/network-io/intelligent-fabric-processors.html

[22] Lavanya Jose, Lisa Yan, George Varghese, and Nick McKeown. 2015. Compiling Packet Programs to Reconfigurable Switches. In Symposium on Networked Systems Design and Implementation. USENIX Association, 103–115.

[23] Norman P. Jouppi, Doe Hyun Yoon, Matthew Ashcraft, Mark Gottscho, Thomas B. Jablin, George Kurian, James Laudon, Sheng Li, Peter Ma, Xiaoyu Ma, Thomas Norrie, Nishant Patil, Sushma Prasad, Cliff Young, Zongwei Zhou, and David Patterson. 2021. Ten Lessons From Three Generations Shaped Google's TPUv4i: Industrial Product. In Proceedings of the International Symposium on Computer Architecture. IEEE/ACM, 1–14. https://doi.org/10.1109/isca52012.2021.00010

[24] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon.

2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. In Proceedings of the Annual International Symposium on Computer Architecture. ACM, 1–12. https://doi.org/10.1145/3079856.3080246

[25] Seungchul Jung, Hyungwoo Lee, Sungmeen Myung, Hyunsoo Kim, Seung Keun Yoon, Soon-Wan Kwon, Yongmin Ju, Minje Kim, Wooseok Yi, Shinhee Han, Baeseong Kwon, Boyoung Seo, Kilho Lee, Gwan-Hyeob Koh, Kangho Lee, Yoonjong Song, Changkyu Choi, Donhee Ham, and Sang Joon Kim. 2022. A Crossbar Array of Magnetoresistive Memory Devices for In-Memory Computing. Nature 601, 7892 (2022), 211–216. https://doi.org/10.1038/s41586-021-04196-6

[26] Ralf Kundel, Amr Rizk, Jeremias Blendin, Boris Koldehofe, Rhaban Hark, and Ralf Steinmetz. 2021. P4-CoDel: Experiences on Programmable Data Plane Hardware. In Proceedings of the International Conference on Communications. IEEE, 1–6. https://doi.org/10.1109/icc42927.2021.9500943

[27] Ike Kunze, Moritz Gunz, David Saam, Klaus Wehrle, and Jan Rüth. 2021. Tofino+ P4: A Strong Compound for AQM on High-Speed Networks?. In International Symposium on Integrated Network Management. IFIP/IEEE, 72–80.

[28] Can Li, Catherine Graves, and John Paul Strachan. 2020. Analog, Non-volatile, Content Addressable Memory. US Patent 10,847,238.

[29] Can Li, Catherine Graves, and John Paul Strachan. 2021. Methods and Systems for an Analog CAM with Fuzzy Search. US Patent 10,998,047.

[30] Can Li, Catherine E Graves, Xia Sheng, Darrin Miller, Martin Foltin, Giacomo Pedretti, and John Paul Strachan. 2020. Analog Content-Addressable Memories with Memristors. Nature Communications 11, 1 (2020), 1–8. https://doi.org/10.1038/s41467-020-15254-4

[31] Shih-Chii Liu, John Paul Strachan, and Arindam Basu. 2021. Prospects for Analog Circuits in Deep Networks. In Analog Circuits for Machine Learning, Current/Voltage/Temperature Sensors, and High-speed Communication: Advances in Analog Circuit Design. Springer, 49–61. https://doi.org/10.1007/978-3-030-91741-8_4

[32] Robert MacDavid, Xiaoqi Chen, and Jennifer Rexford. 2023. Scalable Real-Time Bandwidth Fairness in Switches. In Proceedings of the INFOCOM. IEEE, 1–10. https://doi.org/10.1109/infocom53939.2023.10228997

[33] Shoun Matsunaga, Akira Katsumata, Masanori Natsui, Shunsuke Fukami, Tetsuo Endoh, Hideo Ohno, and Takahiro Hanyu. 2011. Fully Parallel 6T-2MTJ Nonvolatile TCAM with Single-Transistor-based Self Match-line Discharge Control. In Symposium on VLSI Circuits-Digest of Technical Papers. IEEE, 298–299.

[34] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Computer Communication Review 38, 2 (2008), 69–74. https://doi.org/10.1145/1355734.1355746

[35] Carver Mead. 1990. Neuromorphic Electronic Systems. Proc. IEEE 78, 10 (1990), 1629–1636. https://doi.org/10.1109/5.58356

[36] Carver Mead. 2023. Neuromorphic Engineering: In Memory of Misha Mahowald. Neural Computation 35, 3 (2023), 343–383. https://doi.org/10.1162/neco_a_01553

[37] Christopher Monsanto, Nate Foster, Rob Harrison, and David Walker. 2012. A Compiler and Run-time System for Network Programming Languages. ACM SIGPLAN Notices 47, 1 (2012), 217–230. https://doi.org/10.1145/2103656.2103685

[38] Kathleen Nichols, Van Jacobson, Andrew McGregor, and Jana Iyengar. 2018. Controlled Delay Active Queue Management. RFC 8289. https://doi.org/10.17487/RFC8289

[39] Rong Pan, Preethi Natarajan, Fred Baker, and Greg White. 2017. Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem. RFC 8033. https://doi.org/10.17487/RFC8033

[40] Giacomo Pedretti, Catherine E Graves, Thomas Van Vaerenbergh, Sergey Serebryakov, Martin Foltin, Xia Sheng, Ruibin Mao, Can Li, and John Paul Strachan. 2022. Differentiable Content Addressable Memory with Memristors. Advanced Electronic Materials 8, 8 (2022), 2101198. https://doi.org/10.1002/aelm.202101198

[41] Akshay Krishna Ramanathan, Gurpreet S Kalsi, Srivatsa Srinivasa, Tarun Makesh Chandran, Kamlesh R Pillai, Om J Omer, Vijaykrishnan Narayanan, and Sreenivas Subramoney. 2020. Look-Up Table based Energy Efficient Processing in Cache Support for Neural Network Acceleration. In Annual International Symposium on Microarchitecture. IEEE/ACM, 88–101. https://doi.org/10.1109/micro50266.2020.00020

[42] Saad Saleh, Anouk S. Goossens, Tamalika Banerjee, and Boris Koldehofe. 2022. TCAmM$^{\text{CogniGron}}$: Energy Efficient Memristor-Based TCAM for Match-Action Processing. In Proceedings of the International Conference on Rebooting Computing. IEEE, 89–99. https://doi.org/10.1109/ICRC57508.2022.00013

[43] Saad Saleh, Anouk S. Goossens, Tamalika Banerjee, and Boris Koldehofe. 2022. Towards Energy Efficient Memristor-based TCAM for Match-Action Processing. In Proceedings of the International Green and Sustainable Computing Conference. IEEE, 1–4. https://doi.org/10.1109/IGSC55832.2022.9969354

[44] Saad Saleh, Anouk S. Goossens, Tamalika Banerjee, and Boris Koldehofe. 2023. PAmM: Memristor-based Probabilistic Associative Memory for Neuromorphic Network Functions. In Proceedings of the Non-Volatile Memory Technology Symposium. IEEE, 1–5. In Press.

[45] Saad Saleh, Muhammad U Ilyas, Khawar Khurshid, Alex X Liu, and Hayder Radha. 2015. IM Session Identification by Outlier Detection in Cross-correlation Functions. In Proceedings of the Annual Conference on Information Sciences and Systems. IEEE, 1–5. https://doi.org/10.1109/ciss.2015.7086851

[46] Saad Saleh and Boris Koldehofe. 2022. On Memristors for Enabling Energy Efficient and Enhanced Cognitive Network Functions. IEEE Access 10 (2022), 129279–129312. https://doi.org/10.1109/access.2022.3226447

[47] Saad Saleh, Mamoon Raja, Muhammad Shahnawaz, Muhammad U Ilyas, Khawar Khurshid, M Zubair Shafiq, Alex X Liu, Hayder Radha, and Shirish S Karande. 2014. Breaching IM Session Privacy Using Causality. In Proceedings of the Global Communications Conference. IEEE, 686–691. https://doi.org/10.1109/glocom.2014.7036887

[48] Zarin Tasnim Sandhie, Jill Arvindbhai Patel, Farid Uddin Ahmed, and Masud H. Chowdhury. 2021. Investigation of Multiple-Valued Logic Technologies for Beyond-Binary Era. Comput. Surveys 54, 1, Article 16 (2021), 30 pages. https://doi.org/10.1145/3431230

[49] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Prasanna Date, and Bill Kay. 2022. Opportunities for Neuromorphic Computing Algorithms and Applications. Nature Computational Science 2, 1 (2022), 10–19. https://doi.org/10.1038/s43588-021-00184-y

[50] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. 2020. Memory Devices and Applications for In-Memory Computing. Nature Nanotechnology 15, 7 (2020), 529–544. https://doi.org/10.1038/s41565-020-0655-z

[51] Satadal Sengupta, Hyojoon Kim, and Jennifer Rexford. 2022. Continuous In-Network Round-Trip Time Monitoring. In Proceedings of the SIGCOMM Conference. ACM, 473–485. https://doi.org/10.1145/3544216.3544222

[52] Sangho Shin, Kyungmin Kim, and Sung-Mo Kang. 2010. Memristor Applications for Programmable Analog ICs. IEEE Transactions on Nanotechnology 10, 2 (2010), 266–274. https://doi.org/10.1109/tnano.2009.2038610

[53] Vishal Shrivastav. 2022. Programmable Multi-Dimensional Table Filters for Line Rate Network Functions. In Proceedings of the SIGCOMM Conference. ACM, 649–662. https://doi.org/10.1145/3544216.3544266

[54] Vishal Shrivastav. 2022. Stateful Multi-Pipelined Programmable Switches. In Proceedings of the SIGCOMM Conference. ACM, 663–676. https://doi.org/10.1145/3544216.3544269

[55] Hardik Soni, Myriana Rifai, Praveen Kumar, Ryan Doenges, and Nate Foster. 2020. Composing Dataplane Programs with μP4. In Proceedings of the SIGCOMM Conference. ACM, 329–343. https://doi.org/10.1145/3387514.3405872

[56] John Paul Strachan, Catherine Graves, and Can Li. 2022. Analog Content Addressable Memory Utilizing Three Terminal Memory Devices. US Patent 11,289,162.

[57] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. 2008. The Missing Memristor Found. Nature 453, 7191 (2008), 80–83. https://doi.org/10.1038/nature06932

[58] Rudy J Van de Plassche. 2012. Integrated Analog-to-Digital and Digital-to-Analog Converters. Vol. 264. Springer. https://doi.org/10.1007/978-1-4615-2748-0

[59] Wei Xu, Tong Zhang, and Yiran Chen. 2009. Design of Spin-Torque Transfer Magnetoresistive RAM and CAM/TCAM with High Sensing and Search Speed. IEEE Transactions on Very Large Scale Integration Systems 18, 1 (2009), 66–74. https://doi.org/10.1109/tvlsi.2008.2007735

[60] Jiancheng Ye, Ka-Cheong Leung, and Steven H Low. 2021. Combating Bufferbloat in Multi-Bottleneck Networks: Theory and Algorithms. IEEE/ACM Transactions on Networking 29, 4 (2021), 1477–1493. https://doi.org/10.1109/tnet.2021.3066505

[61] Zhuolong Yu, Chuheng Hu, Jingfeng Wu, Xiao Sun, Vladimir Braverman, Mosharaf Chowdhury, Zhenhua Liu, and Xin Jin. 2021. Programmable Packet Scheduling with a Single Queue. In Proceedings of the SIGCOMM Conference. ACM, 179–193. https://doi.org/10.1145/3452296.3472887

[62] Le Zheng, Sangho Shin, Scott Lloyd, Maya Gokhale, Kyungmin Kim, and Sung-Mo Kang. 2016. RRAM-based TCAMs for Pattern Search. In International Symposium on Circuits and Systems. IEEE, 1382–1385. https://doi.org/10.1109/iscas.2016.7527507

[63] Mohammed A Zidan, YeonJoo Jeong, Jihang Lee, Bing Chen, Shuo Huang, Mark J Kushner, and Wei D Lu. 2018. A General Memristor-based Partial Differential Equation Solver. Nature Electronics 1, 7 (2018), 411–420. https://doi.org/10.1038/s41928-018-0100-6

[64] Annus Zulfiqar, Ben Pfaff, William Tu, Gianni Antichi, and Muhammad Shahbaz. 2023. The Slow Path Needs an Accelerator Too!. ACM SIGCOMM Computer Communication Review 53, 1 (2023), 38–47. https://doi.org/10.1145/3594255.3594259