SCIENCE CHINA
Information Sciences

• LETTER •

January 2026, Vol. 69, Iss. 1, 119404:1–119404:2
https://doi.org/10.1007/s11432-025-4521-4

Revisiting a classic form of memory-centric computing—lookup table

Weibang DAI[1,2], Xiaogang CHEN[1*], Sannian SONG[1], Houpeng CHEN[1], Shunfen LI[1], Tao HONG[1,2], Zhenhao JIAO[1] & Zhitang SONG[1]

[1]*Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China*
[2]*University of Chinese Academy of Sciences, Beijing 100080, China*

Received 27 February 2025/Revised 20 June 2025/Accepted 16 July 2025/Published online 21 October 2025

Citation Dai W B, Chen X G, Song S N, et al. Revisiting a classic form of memory-centric computing—lookup table. Sci China Inf Sci, 2026, 69(1): 119404, https://doi.org/10.1007/s11432-025-4521-4

Memory-centric computing (MCC) has emerged as a promising solution for the memory wall problem, which is related to notable bottlenecks in modern computing systems. By performing computations close to the memory (e.g., near-memory computing (NMC)) or directly within the memory (e.g., compute-in-memory (CIM)), MCC reduces data movement, leading to reduced latency and energy consumption. MCC has demonstrated its potential in various applications such as artificial intelligence accelerators, database processing, and high-performance computing. However, it faces significant challenges that hinder its widespread adoption. For example, CIM requires substantial modifications to the existing memory devices and is typically limited to specific operations, such as matrix-vector multiplication, which restricts its flexibility and programmability [1]. Additionally, NMC faces architecture-related issues, such as interoperability with host processor caches and virtual memory, because of the need for data sharing [2].

In this study, we revisit a classic computing method, specifically, the lookup table (LUT) computation, which is a well-established approach for accelerating computations by storing precomputed results in memory. This approach is particularly advantageous in computationally intensive or repetitive operations. Different from using LUTs as accelerators in processor-centric computing, the proposed LUT-based MCC platform allocates dedicated memory for diverse LUT algorithms. Although this approach has been increasingly applied, it has not been sufficiently investigated. The LUT-based streaming data processor (LSDP) proposed by Yuemaier et al. has emerged as a *prototype* [3]. However, considering that this architecture uses a field programmable gate array to control the memory array, there is still room for reducing power consumption and latency. A preliminary schematic diagram of a *future* version is shown in Figure 1(a). In this architecture, each node, which consists of a fixed-size LUT and a simple write/read controller, serves as the basic computing unit. A rectangular region composed of one or more adjacent nodes can be configured as a processing block (PB), where the bus width can be expanded horizontally, whereas the capacity can be increased vertically. Various PBs can be designed to execute different algorithms, as denoted by the various colors in the figure. Additionally, preconfigured dedicated pathways establish direct physical connections between PBs before execution, enabling a pipelined dataflow with deterministic latency. This architecture is more flexible than CIM; furthermore, by avoiding data migration, it overcomes the data-sharing bottleneck problem encountered in NMC. In this study, we outline the potential advantages and applications of the LUT-based MCC platform and investigate the directions toward its development.

Potential of the LUT-based MCC platform.

• Rich algorithmic ecosystem: The LUT algorithms are well-established, highly optimized, and easily integrated into existing workflows. In contrast, CIM and NMC still lack a robust library of algorithms and programming tools.

• Flexibility: Different from the operation-specific nature of CIM, LUTs are inherently flexible and reconfigurable; these advantages enable them to implement diverse algorithms. In NMC, data mapping is used to determine the optimal memory allocation for different processing elements (PEs), constituting a longstanding research problem [2]. The LUT-based MCC platform avoids the interaction between PEs and memory, allowing for flexible setting of LUT sizes according to the requirements of specific applications.

• Energy efficiency: By eliminating the need for repetitive computations, LUTs achieve a significant reduction in energy consumption, making them ideal for ultra-low-power applications.

• Scalability: With the emergence of non-volatile memory (NVM) technologies, such as phase-change memories (PCMs) and resistive random-access memories, large-capacity RAM arrays can be used to implement LUT-based computation at scale, further enhancing the applicability of the LUT-based platform.

Applications specifically suitable for LUT-based computing.

• Function approximation. LUTs excel in approximating complex mathematical functions that are computationally intensive to implement using traditional methods. For example, trigonometric and logarithmic functions, which are often used in graphics processing, signal processing, and scientific computing, can be efficiently handled using precomputed LUTs. CIM and NMC face difficulties in performing these operations because of their hardware limitations. In such cases, LUTs can provide a straightforward

* Corresponding author (email: chenxg@mail.sim.ac.cn)

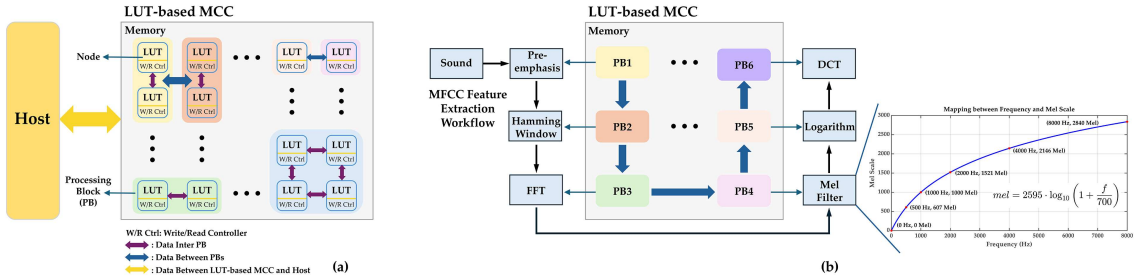

Figure 1 (Color online) (a) Schematic diagram of the preliminary concept of the LUT-based MCC platform; (b) block diagram showing the extraction of MFCCs using the proposed LUT-based MCC platform.

and effective solution.

• Neural networks. LUTs can directly store precomputed weight values and activation function outputs, enabling fast and energy-efficient computations. Neural network weights, which are often fixed during inference, can be stored in LUTs for direct retrieval, thus eliminating the need for repeated multiplications and additions. Additionally, LUTs can accelerate nonlinear activation functions, such as the rectified linear unit, sigmoid, and softmax functions, by mapping input values to their corresponding outputs. By incorporating LUTs into memory-centric architectures, the computational bottleneck in neural network inference can be alleviated, leading to improved performance and reduced power consumption.

• High-complexity computations. Certain applications, such as encryption and scientific simulations, involve high-computational-complexity simulations. In such operations, LUTs can directly map inputs to outputs, thus avoiding the need for intensive calculations. Our approach is especially suited to scenarios involving repeatedly performed computations.

• Ultra-low-power applications. In energy-constrained applications, such as wearable devices and IoT sensors, LUT-based computation provides an effective solution. By precomputing results and storing them in memory, LUTs can minimize power consumption while maintaining computational accuracy.

Case study. To provide insights into the operation of the proposed LUT-based MCC platform, we consider the mel-frequency cepstrum coefficients (MFCCs) extraction process as a simple case study. MFCCs are features representing the short-term power spectrum of sound in a way that mimics human auditory perception; MFCCs are widely used in acoustic processing operations such as speech and speaker recognition [4]. The steps required for extracting MFCCs are presented in Figure 1(b); each step is separately assigned to a PB. The windowing step involves complex cosine operations. Additionally, the fast Fourier transform (FFT), logarithmic operations, and the discrete cosine transform in the other steps are relatively complex computational tasks. Therefore, function approximation based on LUTs is well-suited in this case. Furthermore, the mel-frequency scale mapping is described by a complex equation; in this case, LUT-based computation would be a resource-efficient approach. The dataflow direction is between PBs that undertake the LUT tasks before execution, and the entire process adopts the pipeline mode. The number of nodes is different in each PB (although in Figure 1(b), their sizes appear to be the same). LUT-based computing is advantageous in the extraction of MFCCs. For example, during FFT processing, our *prototype* LSDP exhibits significant advantages over the existing MCC solutions [3]; for example, it achieves 70.6% lower power consumption (3.53 mW vs. 12 mW) compared with that reported in [5]. In addition, emerging NVMs can be used to improve the performance of LUT-based MCC. For example, considering the use of PCMs in LSDPs as an example, the power consumption can be reduced to 0.66 nJ/KB at a read voltage of 3.3 V, and the read time can be reduced to only 20 ns [3].

Conclusion. LUT-based computing is a classic but underinvestigated method, providing a promising avenue for advancing MCC. By exploiting the strengths of LUTs, i.e., availability of algorithms, flexibility, and energy efficiency, many of the limitations of the current MCC architectures, such as CIM and NMC, can be overcome. With the rise in emerging NVM technologies, LUT-based computing should be revisited, and its potential applications should be investigated.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2023YFB4502903).

References

1 Verma N, Jia H, Valavi H, et al. In-memory computing: advances and prospects. IEEE Solid-State Circuits Mag, 2019, 11: 43–55
2 Singh G, Chelini L, Corda S, et al. Near-memory computing: past, present, and future. Microprocessors Microsyst, 2019, 71: 102868
3 Yuemaier A, Chen X, Qian X, et al. A streaming data processing architecture based on lookup tables. Electronics, 2023, 12: 2725
4 Davis S, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process, 1980, 28: 357–366
5 Yantir H E, Guo W, Eltawil A M, et al. An ultra-area-efficient 1024-point in-memory FFT processor. Micromachines, 2019, 10: 509