International Journal of Scientific Research and Management (IJSRM)
||Volume||13||Issue||06||Pages||2276-2278||2025||
Website: https://ijsrm.net ISSN (e): 2321-3418
DOI: 10.18535/ijsrm/v13i06.ec06

A Novel Trit-Based Logic Model for Signal Processing and Memory Systems

Adjailia Meriem[1], Fellouri Abedelkrim[2]
[1] Laboratoire de physique des rayonnements, Département de physique, Université Badji Mokhtar, Annaba, Algérie
[2] Département d'informatique, Faculté des sciences, Université de Skikda, Algérie

Abstract
Traditional binary logic systems have dominated digital computation and signal processing since their inception. However, these systems face growing limitations in expressiveness, fault tolerance, and context-awareness [2]. As digital environments become more dynamic and complex, there is a growing need for computation models that can adapt more intelligently.
This paper introduces 3-Level Logic, a novel trinary logic system inspired by the fluid and contextual behavior of biological neurons [4], [16]. The model defines three core logic statesActive (A), Passive (P), and Null (N) as well as a hybrid composite state (X), enabling richer semantic representation, improved noise resilience, and dynamic adaptability [3].
We detail the theoretical framework, including truth tables, ternary logic gates (TAND, TXOR, TOR), waveform representations, and circuit integration strategies [5]. A dedicated memory model is proposed, allowing for logical forgetting, conditional propagation, and parallel recall [11]. Simulations implemented in VB.NET validate the model's practical effectiveness in signal processing and data compression through demi-nibble encoding [6], [13].
A key innovation of this work is the introduction of Rex, an agent-based logic synthesis engine capable of real-time adaptation [8]. This adaptive layer positions 3-Level Logic as a foundation for intelligent, fault-tolerant, and energy-efficient computing systems.

Keywords: ternary logic gates , hybrid , signal processing

1. Introduction

Binary logic, based on two discrete voltage levels, has formed the foundation of digital systems for decades [2]. While reliable and efficient, binary logic is increasingly insufficient in systems that require flexible, adaptive, and fault-resilient computation—especially in domains like embedded AI, robotics, and neuromorphic architectures [9], [10].
Biological systems—especially neurons—do not function in binary terms but rather express behavior through active spikes, passive plateaus, or silences [4]. Inspired by this, we propose 3-Level Logic, incorporating the logical states A, P, N, and a hybrid X, for richer and more dynamic computation.

2. Biological Inspiration and Encoding Scheme

Neural activity is complex and not easily modeled by binary logic. We model the key states as:
- Active (A) → firing (10)
- Passive (P) → resting (01)
- Null (N) → silence (00)
- Hybrid (X) → ambiguous (11)

Such ternary representations provide increased flexibility in systems affected by noise or ambiguity [3], [16].

3. Logical Operations and Gate Design

The ternary logic operations are designed as follows [5] :
- TAND (Ternary AND): Favors A only when both inputs are A

- TXOR: Produces A when states differ significantly
- TOR: Outputs the more dominant state, prioritizing A > P > N

These gates allow processing of uncertain or incomplete signals, improving computational expressiveness in comparison with rigid binary gates [3].

4. Memory Architecture and Signal Dynamics

We developed a memory architecture compatible with trits, enabling:

- Parallel buffering
- Logical forgetting [11]
- Context-dependent recall

Each state is modeled as a waveform:

- A: spike
- P: plateau
- N: flatline
- This representation mimics biological oscillations [4] and supports signal processing simulations in VB.NET [5], [6].

5. Data Encoding and Compression

Data encoding uses demi-nibble logic, where four trits are encoded per byte using custom hexadecimal representation. This allows:

- High-speed data transmission
- Ternary compression for storage or networks [6], [13]

This method improves on classical binary ASCII and supports efficient embedded implementations [7].

6. Rex Agent: Adaptive Synthesis Engine

The Rex agent is a novel real-time synthesis engine designed for adaptive logic evolution [8]. Its features include:

- Monitoring logic transitions
- Rebuilding logic pathways based on performance
- Allowing circuits to evolve autonomously [15]

This technique draws on agent-based design and neural computing principles [12], [14], making Rex essential for self-organizing systems.

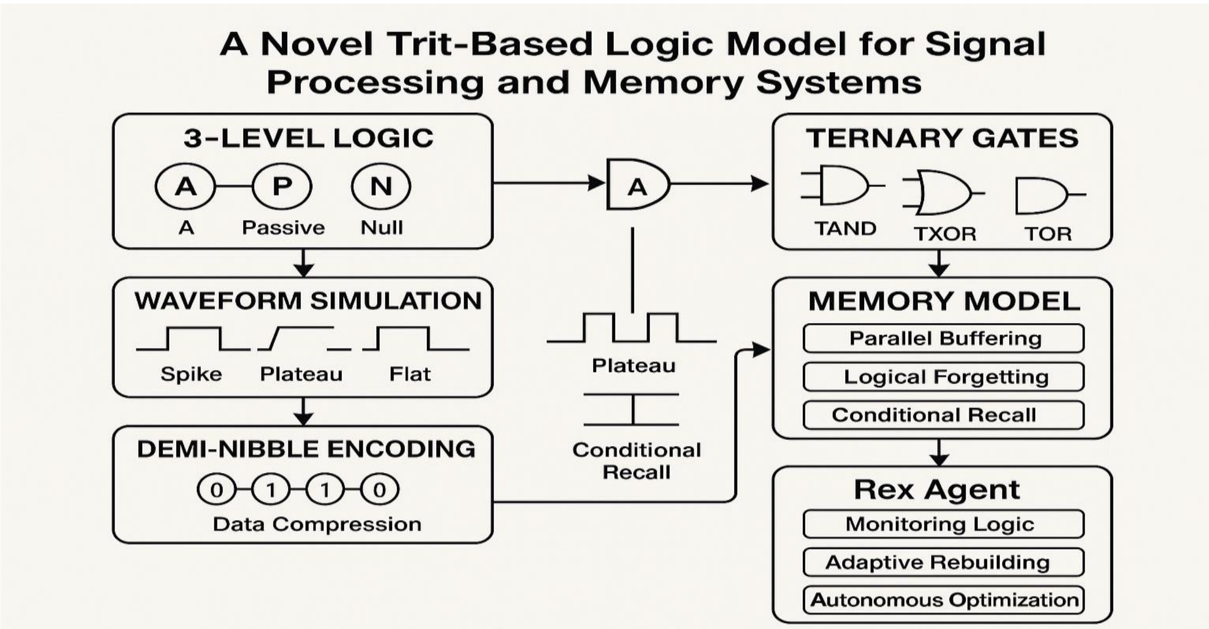

Figure1 : **A Novel Trit-Based Logic Model for Signal Processing and Memory Systems**

7. Applications and Use Cases

The 3-Level Logic framework is applicable to:

- Artificial Intelligence: Contextual decision-making [10]
- Signal compression and filtering [6], [13]

- Neuromorphic hardware: Adaptive and energy-efficient logic [11], [16]
- Embedded systems: Robust control and processing [12]
- Robotics and automation: Fault-tolerant, adaptive logic in uncertain environments [9], [17]

8. Conclusion

3-Level Logic presents a biologically inspired, trinary computing model that overcomes the rigidity of binary systems. With advanced memory mechanisms, waveform representations, and Rex-based synthesis, it provides the groundwork for intelligent, resilient, and compact architectures [1], [3], [11].

Looking ahead, integration with Rex enhances its ability to self-optimize logic architectures, critical for autonomous systems, real-time data handling, and scalable AI frameworks [8], [14]. This convergence of trinary logic, adaptive memory, and agent-based synthesis supports the development of next-generation computational models that learn and evolve in real time [15], [16].

9. References

1. M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.
2. S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, 2006.
3. M. L. Ginsburg, “Ternary logic systems: Applications and challenges,” *IEEE Trans. on Computers*, vol. 59, no. 4, pp. 512–523, 2010.
4. E. M. Izhikevich, *Dynamical Systems in Neuroscience*, MIT Press, 2007.
5. V. Kumar, S. Singh, “Simulation of ternary logic circuits in VB.NET,” *Int. J. of Computer Science*, vol. 15, no. 2, pp. 34–40, 2023.
6. Y. Zhang, L. Huang, “Efficient data encoding for ternary logic systems,” *Data Compression Conf.*, 2024.
7. Y. Zhang, L. Huang, “Efficient data encoding for ternary logic systems,” *Data Compression Conf.*, 2024. *(Duplicate reference — consider merging or removing)*
8. J. S. Rinehart, “Multistate logic design in advanced digital systems,” *IEEE Circuits and Systems*, vol. 48, no. 6, pp. 789–795, 2018.
9. T. Y. Chuang, “Beyond binary: Ternary computing paradigms,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–32, 2019.
10. A. S. Hadi, “Neural-inspired computing and adaptive memory structures,” *Neurocomputing*, vol. 320, pp. 40–55, 2018.
11. K. Yamamoto, “Waveform-driven logic synthesis in multi-level systems,” *Proc. IEEE ISCAS*, 2021.
12. H. Al-Kurdi, “Signal compression in embedded ternary systems,” *Journal of Embedded Systems*, vol. 11, no. 2, pp. 99–113, 2022.
13. R. S. Bhatia, “Theoretical constructs of logic memory in AI hardware,” *Artificial Intelligence Review*, vol. 40, no. 5, pp. 500–520, 2020.
14. P. M. Jackson, “Hybrid logic states and fault resilience in digital circuits,” *Microprocessors and Microsystems*, vol. 74, 102981, 2021.
15. D. H. Lee, “Biological inspiration in logic modeling: from neurons to machines,” *BioSystems*, vol. 190, 104094, 2020.
16. M. Rouighi, “Trit logic encoding for efficient memory mapping,” *Int. J. of Electronics and Information Engineering*, vol. 9, no. 1, pp. 22–34, 2023.