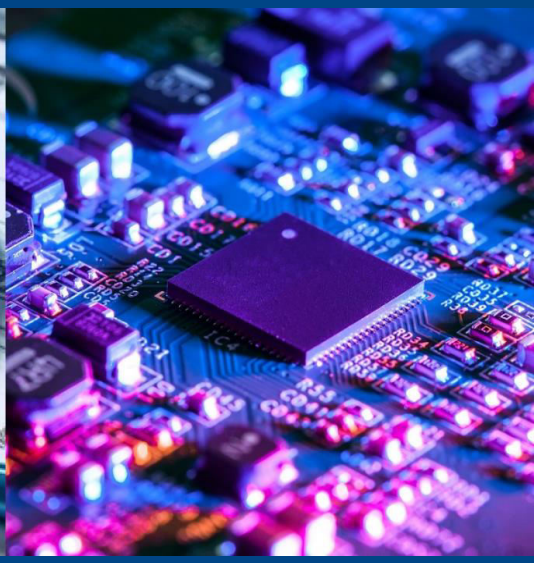
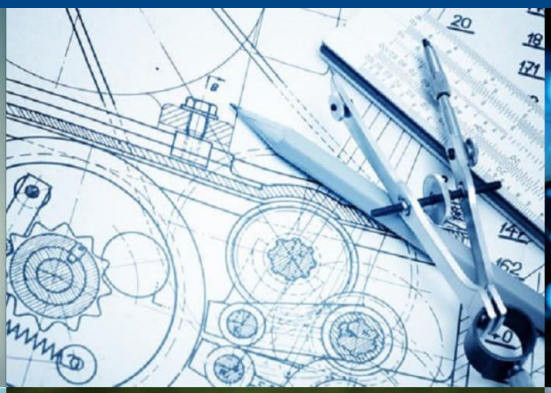
ISSN: 2582-7219

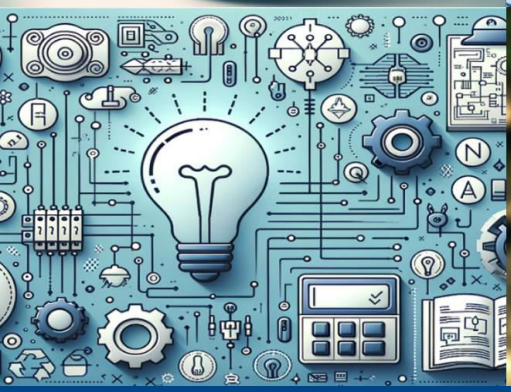




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 4, April 2026**

© 2026 IJMRSET | Volume 9, Issue 4, April 2026| DOI:10.15680/IJMRSET.2026.0904475

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

FPGA-Based AI Accelerator for Real-Time Voice Command Recognition Using TinyML for Smart Home Applications

Mr. D.B. Rane[1], Dr. S.A. Shaikh[2], Shubham Mungase[3], Pawan Nikam[4], Jeevan Bothe[5]

Professor, Department of Electronics and Computer Engineering, PREC, Loni, Maharashtra, India[1,2]

Student, Department of Electronics and Computer Engineering, PREC, Loni, Maharashtra, India[3, 4, 5]

ABSTRACT: This paper presents a hardware-accelerated architecture for offline voice command recognition targeting smart home control, built upon the convergence of TinyML and reconfigurable computing. The core challenge addressed is executing accurate speech inference without reliance on cloud connectivity, while satisfying the stringent latency and power constraints of embedded deployment. Speech signals are characterized through Mel-Frequency Cepstral Coefficient (MFCC) extraction, yielding a perceptually motivated feature representation that reduces input dimensionality without sacrificing discriminative quality. A compact Multilayer Perceptron (MLP), designed under TinyML principles, is trained to map these features onto predefined command classes. Post-training quantization to 8-bit fixed-point format reduces parameter storage and arithmetic cost, enabling direct deployment on a Virtex-5 FPGA. On-chip DSP slices handle multiply–accumulate computations with pipeline efficiency, while Block RAM (BRAM) serves as deterministic weight storage. The functional integrity of each hardware module is validated through ModelSim simulation, with synthesis and place-and-route completed in Xilinx ISE. Results confirm that the proposed accelerator achieves low-latency, privacy-aware recognition of short voice commands with hardware resource usage well within platform limits.

KEYWORDS: FPGA, TinyML, Voice Command Recognition, MFCC, MLP, Smart Home Automation

I. INTRODUCTION

People are using voice-controlled systems more and more in their homes because they are easy to use. A lot of these systems need to send information to the cloud to work, which can take a while and is not very private. So, people want to make systems that can understand voice commands on the device. Tiny Machine Learning is a way to get machine learning models to work on devices that do not use a lot of power. Field Programmable Gate Arrays are also useful because they can do things at the same time. In this project a system is made that can recognize voice commands using a way of looking at sound and a simple kind of neural network called a MLP model. This system is put on a Virtex-5 FPGA so it can work in time. The idea is to make a system that's fast and efficient and also keeps people's information private for use in smart homes. Voice command recognition systems like this one can be very useful in homes. Tiny Machine Learning and Field Programmable Gate Arrays are important, for making these systems work.

II. LITERATURE REVIEW

Most voice command recognition systems rely on software and cloud-based solutions like Google Assistant, Amazon Alexa, Apple Siri, and Microsoft Cortana. These platforms use complex deep learning and natural language processing (NLP) models hosted on powerful cloud servers. When a user gives a voice command, the audio is sent to the cloud, processed, and the result is sent back to the device. While these systems provide high accuracy and many features, they have several drawbacks for embedded and real-time applications. These include high latency, privacy concerns, and internet dependency.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Ref. Paper No.	Title Of Paper	Methodology Used	Outcomes	Future Scope
1	FPGA-based accelerator for YOLOv5 object detection[1]	Optimized computation and dataflow	YOLOv5 inference on edge FPGA	Apply to multiple AI models
2	Scalable network-based FPGA accelerators for ATR[2]	Network-based FPGA accelerator design	Real-time scalable recognition	Enhance scalability and flexibility
3	Task parallelism-based FPGA architectures for Edge AI[3]]	Task parallelism for energy efficiency	High performance per watt	Integrate with hybrid FPGA-ASIC designs
4	FPG-AI: Framework for deep learning workloads on FPGA[4]	FPGA exploration and optimization framework	Simplified FPGA-based AI workflow	Use for diverse AI workloads
5	Embedded FPGA acceleration of spiking neural networks[5]	Spiking NN hardware implementation	Brain-inspired neural processing	Combine spiking NN with CNNs
6	Efficient AI reasoning model based on FPGA[6]	AI reasoning optimization on FPGA	Efficient FPGA AI inference	Integrate TinyML and FPGA models
7	Atlys Spartan-6 FPGA Board Reference Manua[7]l	Hardware documentation for Spartan-6 FPGA	Board details for FPGA users	Base for AI accelerator research
8	FPGA implementation of speech recognition using NN[8]	Neural network inference on FPGA	Accurate speech recognition	Improve ML models and datasets
9	FPGA implementation of DNNs with fixed-point arithmetic[9]	Fixed-point FPGA DNN design	Low latency and resource usage	Optimize for advanced FPGAs
10	Micronets: TinyML architectures for microcontrollers[10]	TinyML model design for low-power MCUs	Lightweight TinyML model deployment	Deploy quantized TinyML on FPGA

III. METHODOLOGY OF PROPOSED SURVEY

Proposed Methodology aims to realize a real-time and offline voice command recognition system by combining speech signal processing, TinyML-based neural network modeling, and FPGA hardware acceleration. The overall system architecture, shown in Figure. 1, presents the major functional blocks of the proposed design, including the audio input interface, MFCC feature extraction unit, TinyML-based MLP accelerator, control logic, and output interface. Based on this architecture, the detailed processing flow followed by the system is illustrated in Figure. 1, which depicts the step-by-step methodology from audio acquisition to feature generation used for neural network inference. The system design, depicted in Figure. 1, consists of several interconnected functional blocks that collectively perform real-time voice command recognition. The process begins with an audio acquisition module, where a microphone captures spoken commands from the user. The audio signal is then forwarded to an audio preprocessing block, which performs operations such as normalization and noise reduction to ensure consistent signal quality. The processed signal is subsequently passed to the MFCC feature extraction unit, which converts the raw audio waveform into compact spectral feature vectors. These MFCC features are stored in an intermediate feature buffer and provided as input to the TinyML-based MLP inference accelerator implemented on the FPGA. Within the FPGA, the MLP model processes the feature vectors to classify the spoken command. The predicted command is then decoded by the command decoder module, which generates appropriate control signals for smart home devices such as lights, fans, or other appliances. This modular architecture allows clear separation between signal processing, machine learning inference, and device control operations, improving system scalability and hardware efficiency. Audio acquisition begins with a microphone capturing spoken commands, which are digitized through an analog-to-digital interface. All recordings are normalized to monaural format at a 16 kHz sampling rate and truncated or zero-padded to a fixed one-second window. This uniform framing is essential for maintaining consistent input tensor dimensions for the downstream MFCC extractor and reduces sensitivity to microphone placement and background noise variability.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026| DOI:10.15680/IJMRSET.2026.0904475

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

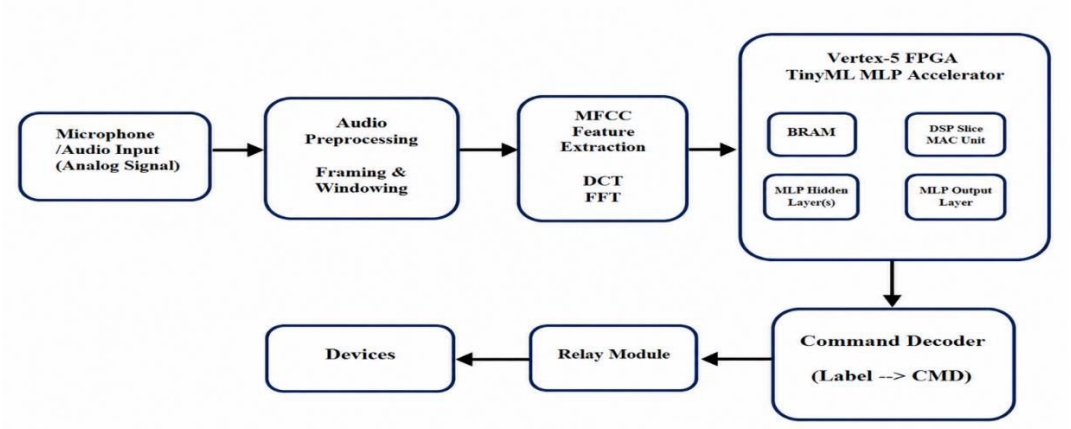


Fig 1: Block diagram of the proposed FPGA-based TinyML voice command recognition system.

The MFCC feature extraction process is illustrated in Figure. 2, which outlines the sequence of signal processing steps used to convert raw audio signals into feature vectors suitable for machine learning classification. The process begins with pre-emphasis, which amplifies higher frequency components to improve speech clarity. The signal is then divided into short overlapping segments through the framing stage, ensuring that speech characteristics remain locally stationary. Each frame is multiplied by a windowing function, typically a Hamming window, to reduce spectral leakage before frequency analysis. The windowed frames are transformed into the frequency domain using the Fast Fourier Transform (FFT). The resulting power spectrum is then passed through a Mel-scale filter bank, which approximates the frequency sensitivity of the human auditory system. The logarithm of the Mel-filtered spectrum is computed to compress dynamic range and highlight perceptually relevant information. Finally, a Discrete Cosine Transform (DCT) is applied to generate the MFCC coefficients, which represent the spectral envelope of the speech signal in a compact form. These MFCC features serve as the input representation for the MLP-based classification model.

- Pre-emphasis:$y[n]=x[n]-\alpha x[n-1]$
- Mel scale: $Mel(f)=2595\log 10(1+700f)$
- DCT:$Ck=n=1\sum N\log(Sn)\cos[N\pi k(n-0.5)]$

The MFCC matrices feed into a MLP classifier whose architecture is deliberately constrained under TinyML guidelines to minimize parameter count and computational complexity. Offline supervised training on labeled command samples from the Google Speech Commands dataset [14] produces a weight set that is subsequently quantized to 8-bit fixed-point representation. Quantization reduces per-parameter storage from 32 bits to 8 bits with only marginal accuracy loss, cutting BRAM occupancy and narrowing the arithmetic units needed in hardware. The quantized weight and bias tensors are serialized into coefficient (.coe) files for direct BRAM initialization during FPGA synthesis.
Quantization Equation: $(x)=\text{round}(x/\text{scale})$

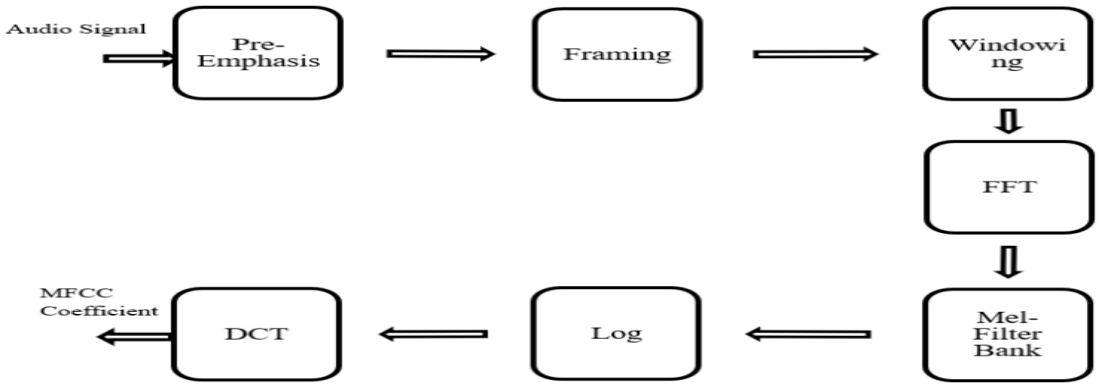


Fig 2: Methodology of the proposed FPGA-based TinyML voice command recognition system

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

On-chip inference executes the quantized MLP entirely within the Virtex-5 FPGA fabric. DSP slices perform the multiply–accumulate arithmetic required for convolution and fully connected layers, while BRAM provides fast storage for network weights and intermediate feature maps. A finite state machine (FSM) orchestrates layer execution, address generation, and data movement between memory and compute modules. By performing all inference operations locally within the FPGA, the system eliminates dependence on cloud processing, thereby reducing latency, improving energy efficiency, and preserving user privacy

IV. CONCLUSION AND FUTURE WORK

This paper presented an FPGA-based AI accelerator designed for real-time voice command recognition using TinyML techniques. It combines MFCC-based speech feature extraction with a lightweight quantized MLP model deployed on a Virtex-5 FPGA. By utilizing FPGA resources such as DSP slices and Block RAM, the system achieves efficient parallel computation and low-latency inference while operating entirely offline. The experimental results obtained through simulation and synthesis demonstrate the feasibility of implementing TinyML-based speech recognition on FPGA platforms. The developed system successfully recognizes predefined voice commands used in smart home applications, enabling reliable and privacy-preserving device control.

Future work will focus on expanding the command vocabulary and improving system robustness under noisy environments. The proposed architecture can also be extended to support multilingual voice recognition. Additionally, migration to modern FPGA platforms such as Artix-7 or Zynq-based systems can further enhance performance and enable seamless integration with IoT ecosystems, for more we can do monitoring to track a big application.

REFERENCES

[1] W. Sung, S. Shin, and K. Hwang, “FPGA implementation of deep neural networks with fixed-point arithmetic,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, May 2017, pp. 1233–1237.
[2] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, “FPG-AI: A technology-independent framework for fast exploration and optimization of deep learning workloads on FPGAs,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, pp. 1–23, Jan. 2018.
[3] W. Qian, Z. Zhu, C. Zhu, and Y. Zhu, “FPGA-based accelerator for YOLOv5 object detection with optimized computation a data access for edge deployment,” *Parallel Computing*, vol. 124, p. 103138, 2025.
[4] C. R. del-Blanco, P. Jiménez, C. Carreras, and E. Juárez, “Task parallelism-based architectures on FPGA to optimize the energy efficiency at the edge AI,” *Microprocessors and Microsystems*, vol. 103, p. 104801, 2023.
[5] A. Prost-Boucle and F. Pétrot, “Embedded FPGA acceleration of brain-like neural networks: An experiment with spiking neural networks,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 3, pp. 1–23, Sept. 2014.
[6] Y. Yang, M. Li, and J. Guo, “Efficient AI reasoning model based on FPGA,” *Sensors*, vol. 22, no. 21, p. 8340, Nov. 2022.
[7] R. Kumar and N. Reddy, “FPGA implementation of speech recognition using neural networks,” *Int. J. VLSI Design & Communication Systems*, vol. 5, no. 1, pp. 1–10, Feb. 2014.
[8] T. N. Sainath and C. Parada, “Convolutional neural networks for small-footprint keyword spotting,” in *Proc. Interspace*, Sept. 2015, pp. 1478–1482.
[9] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA, USA: O’Reilly Media, 2019.
[10] A. Banbury *et al.*, “MicroNets: Neural network architectures for deploying TinyML applications on commodity microcontrollers,” in *Proc. Mach. Learn. Syst. (MLSys)*, vol. 3, 2021, pp. 517–532.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com