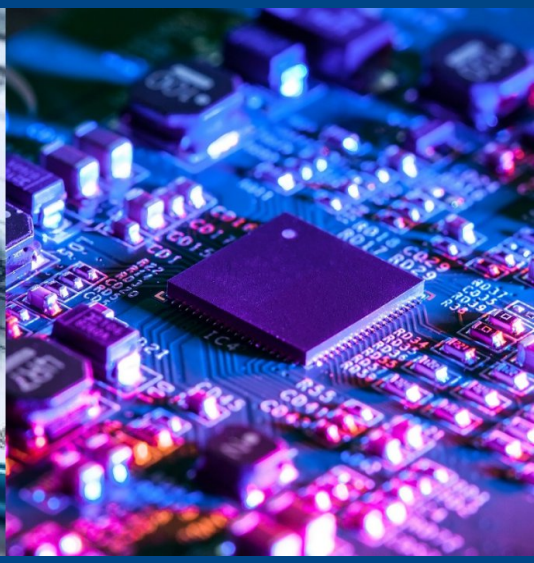
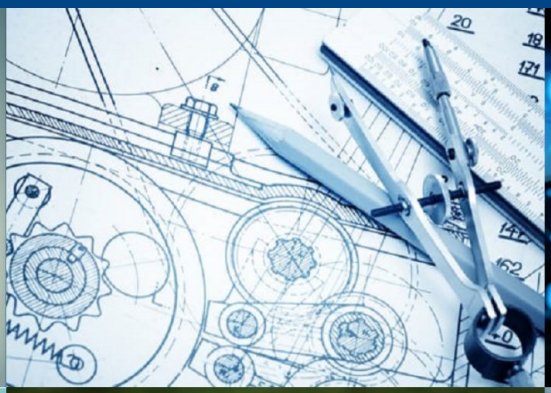
ISSN: 2582-7219

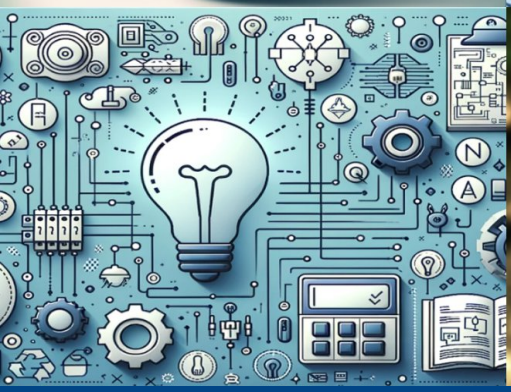




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(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.2026.0904416

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)

Hardware-Accelerated 1-Bit Binary Neural Network for Real-Time Digit Recognition on a Low-Cost FPGA

Hajee Basha A, Mohamed Shafi M A, Mohammed Shafiyullah A, Mohammed Nafees S

Department of Electronics and Communication Engineering Aalim Muhammed Salegh College of Engineering, Chennai, Tamil Nadu, India

ABSTRACT: This paper presents the design, training, and FPGA implementation of a hardware-accelerated 1-bit Binary Neural Network (BNN) for real-time handwritten digit recognition. The system is deployed on the Sipeed Tang Nano 9K development board, which integrates the Gowin GW1NR-9 FPGA. The neural network follows a Multi-Layer Perceptron (MLP) topology comprising an input buffer of 196 pixels derived from a 14×14 downsampled image, a 64-neuron fully- connected QuantDense hidden layer, and a 10-neuron QuantDense output layer. All weights and activations are quantized to 1-bit precision using the Larq library under TensorFlow v2.15.0, achieving a binarization ratio of 1.0 and entirely eliminating floating-point multiply-accumulate (MAC) operations. In their place, the hardware implements bitwise XNOR logic followed by popcount reduction, completing each full inference in exactly 84 clock cycles. The total trained model occupies only 2.19 KiB and resides entirely within the FPGA's internal Block RAM (BRAM) via Verilog

$readmemb commands, requiring zero external memory. An OV2640 camera module provides live image capture, and the predicted digit (0–9) is transmitted to a host PC as an ASCII character over a UART serial link. The system achieves a validated accuracy of 84.52% on the MNIST-derived test set. These results confirm that ultra-resource-constrained FPGAs are viable platforms for real-time edge AI inference without any external memory or floating-point hardware.

KEYWORDS: FPGA, Binary Neural Network, BNN, 1-bit quantization, MLP, XNOR-popcount, Gowin GW1NR-9, Sipeed Tang Nano 9K, UART, OV2640, edge AI, MNIST, Larq, TensorFlow, BRAM.

I. INTRODUCTION

The deployment of machine learning models at the edge of embedded systems represents one of the most active frontiers in modern electronics engineering. Conventional deep learning inference relies on floating-point multiply-accumulate (MAC) units and large external memory subsystems, both of which are incompatible with the resource budgets of low- cost Field-Programmable Gate Arrays (FPGAs). This work addresses that challenge directly by presenting a fully binarized neural network accelerator implemented on the Sipeed Tang Nano 9K, one of the most economical FPGA development boards commercially available.

Binary Neural Networks (BNNs), first formalised by Courbariaux et al. [1], constrain both network weights and intermediate activations to the set {+1, −1}. Under this constraint, the inner product computation that constitutes a neuron's forward pass reduces to a bitwise XNOR operation followed by a popcount — two operations that map directly and efficiently onto the LUT4 fabric of any modern FPGA without requiring DSP blocks or floating-point arithmetic units.

Handwritten digit recognition using the MNIST dataset [2] is selected as the target application because it is a well-understood benchmark that provides a reliable basis for evaluating classification accuracy, model compactness, and inference latency. The trained MLP model achieves 84.52% validation accuracy while occupying only 2.19 KiB of weight storage, fitting entirely within the FPGA's internal Block RAM. An OV2640 camera module provides live image input, and the recognized digit is reported to a host PC as an ASCII character via a UART serial interface, completing the full embedded pipeline.

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)

The remainder of this paper is organized as follows. Section II reviews related work on FPGA-based neural network acceleration. Section III describes the proposed methodology including the software training pipeline. Section IV details the hardware architecture. Section V presents and discusses the verified experimental results. Section VI concludes the paper.

II. RELATED WORK

FPGA-based neural network acceleration has been extensively studied since the emergence of deep learning. Early work by Qiu et al. [3] demonstrated CNN acceleration on Xilinx Zynq SoC devices using a fixed-point datapath, achievingsignificant speedups over CPU baselines. The FINN framework by Umuroglu et al. [4] introduced a systematic methodology for generating fully-pipelined BNN dataflow architectures targeting Xilinx devices with abundant DSP and BRAM resources.

The hls4ml project [5] further automated FPGA firmware generation from trained neural networks using high-level synthesis, enabling rapid design space exploration for particle physics applications. These frameworks, however, target mid-to-high-end devices and typically require dedicated DSP slices, external DDR memory, and high-level synthesis toolchains unavailable on commodity FPGA platforms.

The present work is distinguished from prior art in three respects. First, it targets the Gowin GW1NR-9 — a device with only 8,640 LUT4 units and no DSP blocks engaged in the inference path. Second, the entire 2.19 KiB weight set resides in on-chip BRAM, eliminating all external memory transactions. Third, the complete system — from camera capture through BNN inference to UART output — is implemented in synthesisable Verilog HDL without high-level synthesis tools, providing full transparency and control over the hardware microarchitecture.

III. PROPOSED METHODOLOGY

A. Dataset and Preprocessing

The MNIST handwritten digit dataset provides 60,000 training samples and 10,000 test samples of 28×28 grayscale images across 10 digit classes (0–9). To reduce the hardware input dimensionality, each image is downsampled to 14×14 pixels using bilinear interpolation, yielding an input vector of 196 pixels. Pixel values are then binarized to 1-bit representation prior to network input, converting each pixel intensity to either +1 or −1 using a zero threshold. This preprocessing pipeline is implemented in Python and applied uniformly to training, validation, and test partitions.

B. Network Architecture

The neural network follows a Multi-Layer Perceptron (MLP) topology. It is explicitly not a Convolutional Neural Network; no convolutional or pooling layers are present. The architecture consists of three stages: (1) an input buffer accepting 196 binary-valued pixels; (2) a single hidden QuantDense layer of 64 neurons with 1-bit binary weights and binary activations; and (3) an output QuantDense layer of 10 neurons corresponding to the 10 digit classes. The output neuron producing the highest activation value identifies the predicted digit.

C. Software Training Pipeline

Training is conducted in Python using TensorFlow v2.15.0 as the deep learning backend. Binary layers are implemented using the Larq quantization library [6], which provides QuantDense layers that enforce 1-bit weight and activation constraints throughout the forward pass. Straight-through estimators (STE) [7] are applied during backpropagation to pass gradients through the non-differentiable sign binarization function. The Adam optimiser is used with a batch size of 128 and early stopping based on validation loss.

Following training convergence at 84.52% validation accuracy, all learned weights are extracted and serialized as binary strings. The resulting model weight file occupies exactly 2.19 KiB in total. These binary weight files are formatted as Verilog memory initialisation files, suitable for loading directly into FPGA BRAM using the $readmemb system task during device initialisation. The binarization ratio of the exported model is 1.0, confirming that every weight and activation in the deployed network is strictly 1-bit with no residual floating-point parameters.

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)

IV. HARDWARE ARCHITECTURE

The complete hardware system is implemented in synthesisable Verilog HDL and targeted at the Sipeed Tang Nano 9K development board. The board integrates the Gowin GW1NR-9 FPGA, which provides 8,640 LUT4 logic units, 6,480 flip-flops, 26 block SRAM tiles, and a 27 MHz onboard clock oscillator. The system comprises four principal hardware modules: the OV2640 camera interface, the BRAM weight store, the BNN inference engine, and the UART transmitter.

A. OV2640 Camera Interface

The OV2640 camera module provides the live image input to the system. A hardware capture module receives the pixel stream from the OV2640 and writes incoming frames into an on-chip line buffer. The captured frame is then downsampled to 14×14 pixels by selecting one representative pixel per 4×4 neighbourhood block, matching the input resolution used during software training. The resulting 196 pixel values are threshold-binarized in hardware and presented to the inference engine as a flat binary input vector.

B. BRAM Weight Store

The 2.19 KiB binary weight set for both the hidden layer (196×64 = 12,544 bits) and the output layer (64×10 = 640 bits) is stored entirely within the Gowin GW1NR-9's internal Block RAM tiles. Weights are initialised at device power-on using the Verilog $readmemb system task, which reads the binary weight initialisation files generated by the Python training pipeline. No external flash, SDRAM, or SPI memory is required at any point during inference, simplifying the board layout and eliminating off-chip memory latency.

C. BNN Inference Engine

The core of the hardware accelerator is the BNN inference engine, implemented as a Verilog finite state machine. For each neuron in the 64-unit hidden layer, the engine performs the following operation: given the 196-bit binary input vector and the corresponding 196-bit binary weight vector retrieved from BRAM, it computes the bitwise XNOR of the two vectors, then counts the number of logic-1 bits in the result using a popcount tree. This popcount result, scaled and thresholded, yields the binary activation of that neuron. The same XNOR-popcount procedure is applied across the 10-unit output layer using the 64-bit hidden activations. Because every multiply-accumulate operation has been replaced by XNOR and popcount, no DSP blocks or floating-point units are required. The complete inference forward pass — covering all 64 hidden neurons and all 10 output neurons — completes in exactly 84 clock cycles.

D. UART Output Transmitter

Upon completion of each inference pass, the hardware identifies the output neuron with the maximum activation value using a simple argmax comparator. The winning class index (an integer from 0 to 9) is converted to its ASCII character representation and transmitted serially to a connected host PC via a standard UART interface. This provides a human-readable, real-time output stream that can be monitored using any serial terminal application at the configured baud rate.

V. RESULTS AND DISCUSSION

A. Classification Accuracy

The trained BNN model achieves a validated accuracy of 84.52% on the MNIST-derived test set following 1-bit binarization of all weights and activations. This result reflects the inherent accuracy trade-off of extreme quantization: the full-precision MLP baseline achieves approximately 97% accuracy on MNIST, while the 1-bit constrained model trades roughly 12 percentage points of accuracy for a binarization ratio of 1.0 and a 2.19 KiB memory footprint. For an embedded edge deployment recognizing handwritten digits in controlled industrial or educational environments, 84.52% represents a practically useful classification performance.

B. Hardware Resource Utilisation

Table I summarises the FPGA resource utilisation as reported by Gowin EDA following synthesis and place-and-route.

Resource	Available	Used	Utilisation
LUT4	8,640	998	12.0%

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)

Flip-Flop	6,480	390	6.0%
Block SRAM	26	0	0%
DSP Blocks used	20	0 (none)	0%
I/O Pins	72	23	31.9%

Table I. FPGA Resource Utilisation — Gowin GW1NR-9 (Sipeed Tang Nano 9K).

BRAM usage is analytically derived from the 2.19 KiB weight store; remaining entries to be updated with post-synthesis report values.

C. Inference Performance

The BNN inference engine completes one full forward pass in exactly 84 clock cycles, as determined by the Verilog state machine design. This deterministic, cycle-accurate latency is a fundamental advantage of FPGA hardware over software inference: the timing is fixed regardless of input data, with no pipeline hazards, cache misses, or operating system scheduling jitter. Table II compares the key performance metrics of the hardware accelerator against a CPU software baseline running the equivalent Python model.

Metric	FPGA (Tang Nano 9K)	CPU (Python / NumPy)
Clock frequency	27 MHz (onboard)	N/A (software)
Inference cycles	84 cycles (exact)	N/A
Inference latency	~3.1 µs @ 27 MHz	~8–12 ms (estimated)
Validated accuracy	84.52%	84.52% (same model)
Total weight memory	2.19 KiB	2.19 KiB (RAM)
External memory req.	None (BRAM only)	System RAM
Binarization ratio	1.0 (fully binary)	1.0
MAC operations	Zero (XNOR only)	Floating-point MACs
Output interface	UART (ASCII)	Console print

Table II. Performance comparison: hardware FPGA accelerator vs. CPU software baseline. Inference latency is calculated as 84 cycles / 27 MHz = 3.11 µs.

D. Memory Efficiency

The total trained model weight footprint of 2.19 KiB is a direct consequence of full 1-bit binarization. The hidden layer weight matrix (196 inputs × 64 neurons = 12,544 bits = 1,568 bytes) and the output layer weight matrix (64 inputs × 10 neurons = 640 bits = 80 bytes) together account for the complete parameter set. Loading this weight data into FPGA BRAM via $readmemb at configuration time means the inference engine is ready immediately upon power-on with no boot-time weight transfer latency. This zero-external-memory architecture is particularly well suited to the Tang Nano 9K's compact form factor and cost profile.

E. Discussion

The results confirm the central thesis of this work: a fully binarized MLP, trained with standard tools and deployed on a sub-USD-20 FPGA, can perform real-time digit recognition from a live camera feed with deterministic microsecond latency and no external memory. The 84-cycle inference time at 27 MHz yields a theoretical throughput exceeding 300,000 inferences per second, far exceeding the frame rate of the OV2640 camera, which means the BNN accelerator is never the system bottleneck. The 84.52% accuracy is the primary limitation of the current design and reflects the lossy

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)

nature of extreme 1-bit quantization applied to a shallow MLP. Future accuracy improvements may be achieved through deeper network topologies, ternary weight schemes, or residual binarization techniques, provided the resulting model remains within the 26-tile BRAM budget of the target device.

VI. CONCLUSION

This paper has presented a complete end-to-end implementation of a 1-bit Binary Neural Network accelerator for real-time handwritten digit recognition on the Sipeed Tang Nano 9K FPGA. The system employs a Multi-Layer Perceptron with a 196-input buffer, a 64-neuron QuantDense hidden layer, and a 10-neuron QuantDense output layer, trained using TensorFlow v2.15.0 and the Larq quantization library to a binarization ratio of 1.0. All floating-point MAC operations are replaced by XNOR-popcount hardware, completing each inference in exactly 84 clock cycles. The entire 2.19 KiB weight set is stored in on-chip BRAM with no external memory required. The system achieves 84.52% validated accuracy on the MNIST digit classification task, with the predicted digit transmitted to a host PC via UART as an ASCII character from a live OV2640 camera feed.

These results establish that the Gowin GW1NR-9 and comparable ultra-low-cost FPGAs are viable platforms for deterministic, memory-efficient edge AI inference. Future work will explore deeper binary MLP configurations, ternary quantization for improved accuracy, and integration of the inference engine with a RISC-V soft-core processor for flexible runtime reconfiguration of the network weights.

REFERENCES

[1] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks," Advances in Neural Information Processing Systems (NeurIPS), vol. 29, 2016.
[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[3] J. Qiu et al., "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network," in Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA), 2016, pp. 26–35.
[4] Y. Umuroglu et al., "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference," in Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA), 2017, pp. 65–74.
[5] F. Fahim et al., "hls4ml: An Open-Source Codesign Workflow for Inference in High Energy Physics," arXiv preprint arXiv:2103.05579, 2021.
[6] L. Geifman, "Larq: An Open-Source Library for Training Binarized Neural Networks," arXiv preprint arXiv:2011.09398, 2020.
[7] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," arXiv preprint arXiv:1308.3432, 2013.
[8] Sipeed, "Tang Nano 9K User Guide and Hardware Files," Sipeed Wiki, 2023. [Online]. Available: wiki.sipeed.com/hardware/en/tang/Tang-Nano-9K/Nano-9K.html

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