IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

FlexXNOR-OD: A Channel-Grouped XNOR-Based Variable-Precision Accelerator for Real-Time Edge Object Detection

Budidha Sriman[1] and D. Sateesh[2]
PG Scholar, Dept. of ECE[1]
Assistant Professor, Dept. of ECE[2]
Sree Dattha Institute of Engineering and Science, Hyderabad, Telangana, India.
srimanbudidha99@gmail.com[1], dhudhigamsatheesh@gmail.com[2]

Abstract: *Object detection at the edge requires a difficult balance among detection accuracy, deterministic latency, memory bandwidth, and energy consumption. Existing binarized accelerators replace multipliers with XNOR and population-count logic, but many designs use a fixed binary datapath or select precision only at the layer level, limiting their ability to protect accuracy-sensitive channels while fully exploiting low-bit parallelism. This paper proposes FlexXNOR-OD, a new FPGA-oriented accelerator for real-time object detection that combines channel-group precision assignment with a lane-fusible XNOR processing element. The architecture supports 1-, 2-, 4-, and 8-bit weight/activation groups using four independently clock-gated XNOR-popcount slices per processing element. The slices operate independently in binary mode and are fused through signed bit-plane recombination for higher-precision groups. A 64-PE output-stationary array, dual-bank feature and weight buffers, a precision-and-tile scheduler, and an integrated batch-normalization, RPReLU, and requantization pipeline minimize data movement and control overhead. An analytical model for a 200-MHz design point predicts a peak rate of 4096 binary dot-product bit pairs per cycle, equivalent to 0.819 Tbinary-MAC/s or 1.638 TOPS when a multiply and accumulation are counted separately. For an illustrative channel-group precision map with an average weight width of 2.12 bits, parameter storage is reduced by 3.77× relative to uniform INT8 and 15.1× relative to FP32. The proposed design therefore offers a practical path toward precision-scalable, multiplier-light object detection on resource-constrained edge platforms. A complete RTL synthesis and dataset-level evaluation protocol is specified to support reproducible implementation.*

Keywords: binary neural network, edge AI, FPGA accelerator, object detection, variable precision, XNOR-popcount, channel-group quantization.

I. INTRODUCTION

Object detection extends image classification by jointly predicting object categories and spatial locations. This capability supports autonomous navigation, surveillance, traffic analysis, robotics, agriculture, and smart sensing, but it also introduces a much larger computational burden than classification because feature extraction, multi-scale fusion, classification, and bounding-box regression must be executed for every frame. One-stage detectors reduce algorithmic latency by producing predictions in a single forward pass, yet their convolutional backbones and detection heads still require substantial arithmetic and data movement [1]–[4].

General-purpose GPUs deliver high throughput, but their energy, thermal, and platform requirements are often unsuitable for small edge devices. FPGAs provide spatial parallelism, custom memory hierarchies, deterministic timing, and reconfigurability, making them attractive for embedded inference. The efficiency of an FPGA accelerator,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

however, depends less on nominal operation count than on how effectively the design reduces multiplier cost, keeps processing elements busy, and avoids external-memory traffic [5], [6].

Quantization attacks both arithmetic and storage cost by reducing the numerical width of weights and activations. At the extreme, binarized neural networks represent operands with one bit and replace conventional multiplication with XNOR followed by population count [7], [8]. Binary networks are highly attractive for logic-based accelerators, but uniform binarization can degrade object-detection accuracy because the stem, transition layers, feature-fusion paths, and prediction heads have different sensitivity to quantization. Mixed-precision methods therefore preserve more bits only where they are needed [9]–[12].

A recent XNOR-based variable-precision object-detection processor demonstrated the value of combining a compact binarized network, a DenseToRes feature-preservation structure, and an FPGA datapath that supports more than one precision [13]. That work motivates the present study but also exposes an opportunity: precision can be scheduled at a finer channel-group granularity, and the binary hardware can be organized so that unused slices become independent parallel lanes instead of remaining idle. The uploaded project document similarly identifies multiplier cost, fixed precision, memory bandwidth, and limited scalability as the central design constraints for edge deployment.

The principal contributions of this paper are:

- A channel-group precision scheduler that selects 1, 2, 4, or 8 bits for groups of output channels rather than enforcing one precision across an entire layer.
- A lane-fusible XNOR processing element containing four 16-bit XNOR-popcount slices. The slices execute independent binary dot products or fuse for multi-bit signed bit-plane recombination.
- A 64-PE output-stationary accelerator with double-buffered feature and weight memories, precision-aware clock gating, and integrated normalization, activation, and requantization.
- An analytical performance and storage model, together with a complete RTL, synthesis, and detection-accuracy evaluation methodology that avoids unsupported empirical claims.

II. RELATED WORK AND RESEARCH GAP

2.1 Object Detection and Low-Precision Networks

Two-stage detectors such as Faster R-CNN generate candidate regions before classification, while one-stage detectors such as YOLO and SSD directly predict classes and bounding boxes [2]–[4]. One-stage models are commonly selected for real-time systems, but their computational cost remains dominated by convolution. Fully quantized detectors have shown that low-bit inference can be applied to complex detection workloads when training instability and sensitive layers are handled carefully [10]. BiDet further demonstrated that binary object detection requires detector-specific treatment rather than direct replacement of full-precision layers [11].

BinaryConnect and binarized neural networks established the feasibility of training networks with binary parameters and activations [7]. XNOR-Net formalized binary convolution as XNOR plus bit count [8]. Bi-Real Net, BinaryDenseNet, and ReActNet improved information flow and activation modeling, reducing the accuracy gap between binary and real-valued networks [14]–[16]. More recent partial-binarization approaches show that selective retention of higher precision can outperform indiscriminate binarization under a fixed computational budget [17].

2.2 FPGA Accelerators

Eyeriss established the importance of data reuse and locality in energy-efficient neural accelerators [5]. FINN and FINN-R mapped quantized and binary networks to deeply pipelined FPGA datapaths [6], [18]. NN2CAM extended multi-precision streaming inference to FPGA-based cameras, while mixed-precision design-space studies showed that the PE structure, array dimensions, and dataflow must be optimized together [19], [20]. These works confirm that numerical precision should influence not only memory format but also spatial parallelism and scheduling.

For object detection, low-power heterogeneous implementations have demonstrated that quantized detector execution can be partitioned between embedded processors and programmable logic [21]. Lee et al. proposed a real-time XNOR-based variable-precision processor and reported 64.51 frames/s with 64.92 mAP on its evaluation setup [13]. Its

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

DenseToRes model and variable-precision unit are important prior art; therefore, the present work does not claim those concepts as new. FlexXNOR-OD instead focuses on channel-group precision, four-level precision support, lane fusion, and clock-gated sub-lanes as the differentiating architectural contributions.

2.3 Identified Gap

The literature reveals four unresolved issues. First, fixed binary accelerators maximize throughput but often sacrifice detector accuracy. Second, layer-wise mixed precision may still waste computation because channels within the same layer can have different sensitivity. Third, many precision-scalable units reduce arithmetic width without proportionally increasing useful parallelism. Fourth, memory transfers can dominate latency unless tile fetch, compute, and write-back are overlapped. FlexXNOR-OD addresses these issues through channel-group scheduling, lane fusion, precision-aware clock gating, and ping-pong buffering.

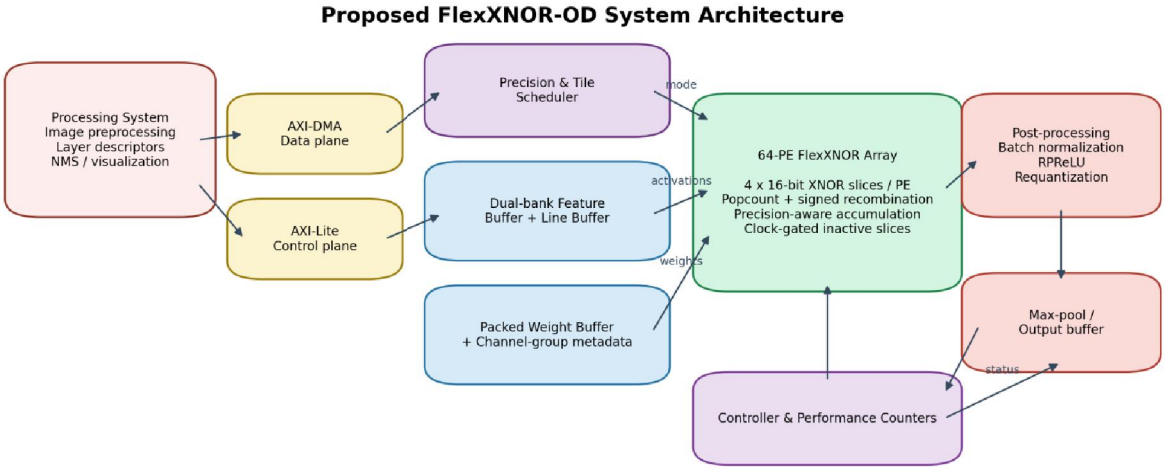

Figure 1. Overall architecture of the proposed FlexXNOR-OD accelerator.

III. PROPOSED FLEXXNOR-OD ARCHITECTURE

3.1 Design Objectives

The proposed accelerator is designed for an embedded FPGA system-on-chip in which an application processor performs image resizing, normalization, detection decoding, non-maximum suppression, and display, while programmable logic executes convolution-dominated feature extraction. The architecture targets five objectives: multiplier-light computation, fine-grained precision control, scalable parallelism, high on-chip reuse, and deterministic layer execution.

As shown in Figure 1, the processing system transfers input tiles and layer descriptors through AXI interfaces. The programmable-logic subsystem contains a precision-and-tile scheduler, dual-bank feature and weight memories, a 64-PE compute array, integrated post-processing, an optional max-pool unit, and performance counters. All data-path widths are multiples of 512 bits so that packed low-precision operands can be streamed without format conversion in external memory.

3.2 Channel-Grouped Precision Assignment

Uniform precision is simple but inefficient. FlexXNOR-OD partitions the output channels of each convolution into groups of 16. Each group stores a two-bit precision code selecting 1, 2, 4, or 8 bits. During calibration, the quantization controller evaluates the detection-loss increase caused by lowering the precision of each group. Groups with the smallest sensitivity are assigned binary execution first, followed by 2-bit and 4-bit modes until an accuracy or resource budget is reached. The stem, final feature-fusion layers, and detection heads are retained at 8 bits by default.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

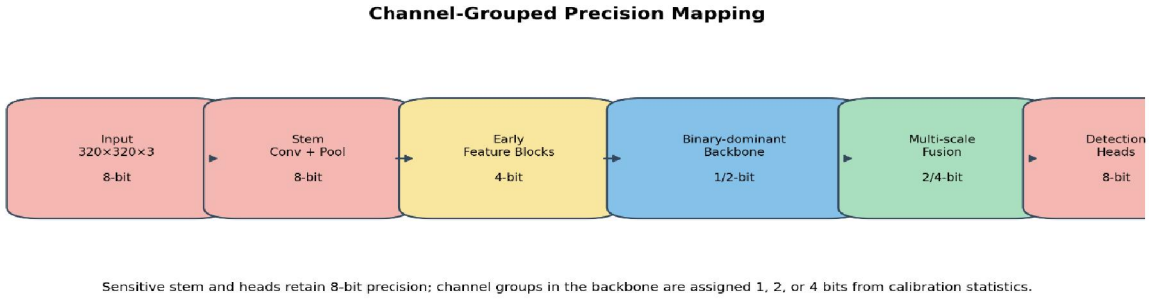

Figure 2. Illustrative precision map used by the analytical model.

This channel-group policy provides a finer trade-off than layer-wise quantization while keeping the metadata small. For a group size of 16 channels, only two metadata bits are required per group, corresponding to 0.125 bits per output channel. The metadata is stored beside the packed weights and is consumed by the scheduler before each group is issued to the PE array.

3.3 Lane-Fusible XNOR Processing Element

Each PE contains four 16-bit XNOR slices, four local population-count units, a lane-fusion network, a signed recombination tree, a precision-aware accumulator, and post-processing support. In 1-bit mode, the slices are independent and can process four binary channel groups concurrently. In 2-bit mode, two slices are fused per logical lane. In 4-bit and 8-bit modes, all slices cooperate over multiple bit-plane cycles. Inactive slices and unused adder-tree levels are clock-gated to reduce switching activity.

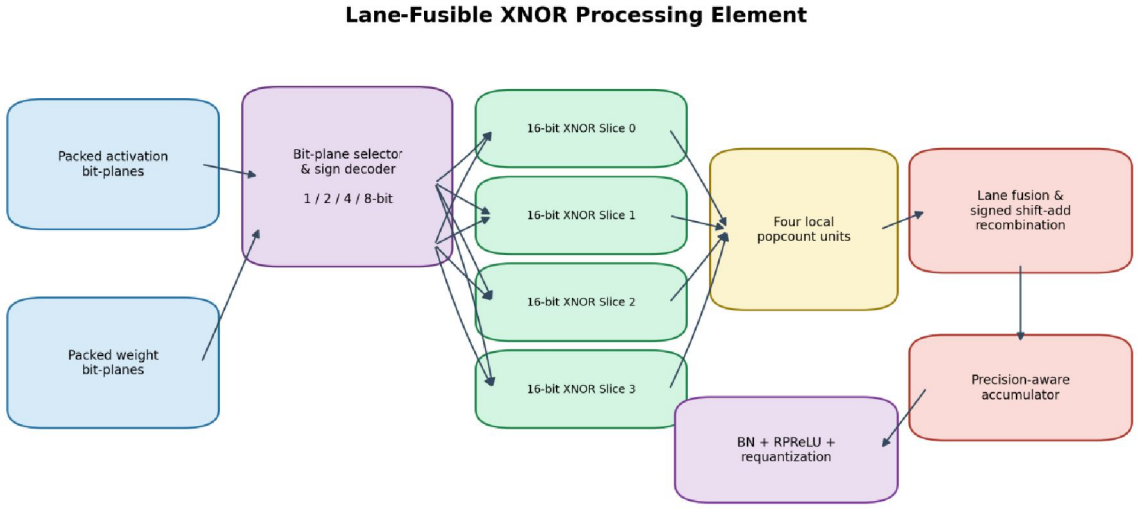

Figure 3. Internal organization of a lane-fusible XNOR processing element.

The PE does not instantiate a general-purpose 8 × 8 multiplier for every lane. Instead, quantized operands are represented as weighted bipolar bit planes. XNOR-popcount produces the signed correlation for each pair of planes, and a shift-add network recombines the partial terms. The same physical XNOR slices therefore support binary and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

multi-bit groups. This approach is most attractive when low-bit groups dominate the backbone, while the retained 8-bit groups protect accuracy-sensitive operations.

3.4 Output-Stationary Dataflow and Memory System

The PE array uses an output-stationary dataflow. Each PE retains a partial sum for one output channel and spatial location until all kernel positions and input-channel groups have been accumulated. Activations are broadcast across PEs, while weights are distributed according to output-channel assignment. This reduces repeated partial-sum traffic and allows the accumulation width to be adapted to the selected precision and kernel depth.

Two feature banks and two weight banks implement ping-pong buffering. While the PE array computes tile k, DMA and unpack logic load tile k+1 into the alternate bank. A line buffer reuses overlapping rows for 3 × 3 and 5 × 5 kernels. The scheduler changes banks only after both transfer and computation complete, preventing pipeline hazards. The same mechanism overlaps output write-back with the next tile computation.

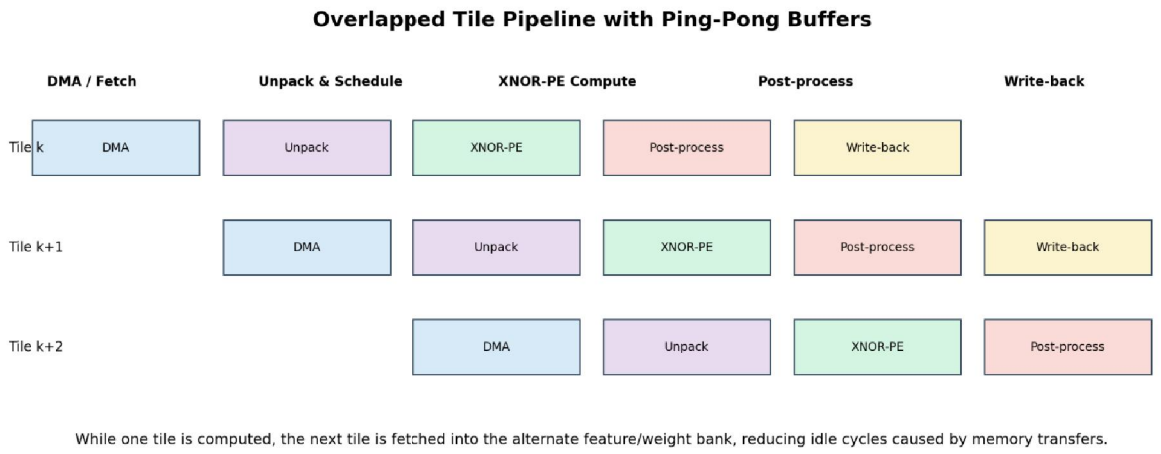

Figure 4. Overlapped tile execution enabled by ping-pong buffers.

3.5 Post-Processing and Control

Batch-normalization parameters are folded into scale and bias constants whenever possible. The post-processing path supports affine normalization, RPReLU activation, saturation, and requantization to the precision required by the next channel group. Max pooling is optional and bypassed for layers that use strided convolution. The controller executes a compact layer descriptor containing feature dimensions, kernel size, stride, channel count, memory addresses, precision metadata address, activation mode, and output scaling parameters.

Table 1. Principal architectural parameters of the proposed design.

Parameter	Proposed value	Purpose
Processing elements	64	Parallel output-channel computation
XNOR width per PE	4 × 16 bits	Independent binary lanes or fused multi-bit execution
Supported precision	1, 2, 4, and 8 bits	Channel-group accuracy/efficiency control
Channel-group size	16 output channels	Low metadata overhead and simple scheduling
Feature/weight interface	512 bits	Efficient packed transfers

DOI: 10.48175/IJARSCT-38402

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

Parameter	Proposed value	Purpose
Dataflow	Output stationary	Local partial-sum reuse
Buffering	Dual-bank feature and weight SRAM/BRAM	Overlap transfer and compute
Post-processing	BN, RPReLU, requantization, optional max pool	Complete convolution block in programmable logic
Control interface	AXI-Lite; AXI-Stream/DMA for data	Heterogeneous PS-PL integration

IV. COMPUTATIONAL FORMULATION

4.1 Binary Dot Product

For bipolar binary vectors b_x, b_w $\in \{-1,+1\}^N$, equality contributes +1 and inequality contributes −1. The signed dot product is therefore obtained from XNOR and population count:

$$s_{\text{bin}} = 2\ \text{popcount}\big(\text{XNOR}\ (b_x, b_w)\big) - N \qquad (1)$$

where:

Equation (1) replaces N multiplications with N one-bit comparisons and a population count. In the proposed design, N = 16 per slice and four slices are available in every PE.

4.2 Multi-Bit Bipolar Decomposition

A quantized scalar or vector can be approximated by a weighted sum of bipolar bases. For B_x-bit activations and B_w-bit weights:

$$x \approx \sum_{i=0}^{B_x-1} \alpha_i\, b_i,\ w \approx \sum_{j=0}^{B_w-1} \beta_j\, c_j,\ b_i, c_j \in \{-, 1+1\} \qquad (2)$$

$$x \cdot w \approx \sum_{i=0}^{B_x-1} \sum_{j=0}^{B_w-1} \alpha_i \beta_j \left[2\ \text{popcount}\Big(\text{XNOR}\ \big(b_i, c_j\big)\Big) - N\right] \qquad (3)$$

The coefficients α_i and β_j are powers of two for a hardware-friendly uniform quantizer, so recombination is realized with shifts and additions. A 1 × 1-bit group requires one plane pair, a 2 × 2-bit group requires four pairs, and an 8 × 8-bit group requires 64 pairs. Lane fusion and temporal reuse allow the same XNOR slices to execute every case.

4.3 Cycle and Throughput Model

Let C_in be the number of input channels, K the kernel dimension, L the number of binary operands processed per PE cycle, and B_x and B_w the activation and weight precisions. Ignoring pipeline fill and memory stalls, the cycles required for one output-channel group and one output pixel are:

$$C_{\text{group}} = \left\lceil \frac{C_{\text{in}} K^2}{L} \right\rceil \frac{B_x B_w}{F_{\text{lane}}} \qquad (4)$$

where F_lane is the effective lane-fusion parallelism. For the four-slice PE, F_lane = 4 in fully independent binary mode, 2 for paired 2-bit groups, and 1 when all slices cooperate. The effective binary-pair throughput is:

$$T_{\text{bit}} = f_{\text{clk}} \times N_{\text{PE}} \times L \qquad (5)$$

For N_PE = 64, L = 64 aggregate XNOR bits per PE, and f_clk = 200 MHz, T_bit equals 819.2 Gbinary-MAC/s. Counting the binary product and accumulation as two operations gives 1.638 TOPS.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

4.4 Storage Model

If p_b is the fraction of weights assigned precision b, the average stored width is:

$$B_{\text{avg}} = \sum_{b\in\{1,2,4,8\}} p_b \; b \qquad (6)$$

$$\text{Compression}_{\text{INT8}} = \frac{8}{B_{\text{avg}}},\ \text{Compression}_{\text{FP3 2}} = \frac{32}{B_{\text{avg}}} \qquad (7)$$

The illustrative map used in this paper assigns 60% of parameters to 1 bit, 20% to 2 bits, 12% to 4 bits, and 8% to 8 bits. The resulting B_avg is 2.12 bits, corresponding to 3.77× compression relative to INT8 and 15.1× relative to FP32, excluding the small precision metadata.

V. IMPLEMENTATION AND VERIFICATION METHODOLOGY

5.1 RTL Organization

The proposed accelerator can be described in synthesizable SystemVerilog as seven independently verifiable modules: (1) AXI input/output adapters, (2) feature and weight buffers, (3) precision-and-tile scheduler, (4) PE array, (5) post-processing pipeline, (6) layer controller, and (7) performance counters. Parameterized generate blocks define the PE count, slice width, accumulator width, and memory depth. The precision code is registered at group boundaries so that all control signals remain static during a dot-product sequence.

5.2 Functional Verification

A bit-accurate Python reference model should quantize activations and weights using the same bipolar decomposition and rounding rules as the RTL. Randomized tests compare the output of every precision combination, including 1 × 1, 2 × 2, 4 × 4, 8 × 8, and asymmetric combinations. Directed tests cover accumulator overflow, negative values, saturation, precision transitions, padded channels, bank switching, and back-to-back layer descriptors. Assertions verify valid/ready handshakes, buffer ownership, and the absence of writes to an active compute bank.

5.3 FPGA Evaluation Protocol

The recommended implementation target is a Xilinx Zynq UltraScale+ MPSoC or a comparable FPGA with sufficient BRAM/URAM capacity. Vivado synthesis and implementation should report LUTs, flip-flops, BRAM/URAM, DSP blocks, achieved clock frequency, and estimated dynamic power. Throughput must be measured from DMA start to final output availability, with separate reporting for accelerator-only and end-to-end frame rate. Power should be sampled under continuous video input after thermal stabilization.

5.4 Detection Evaluation Protocol

The detector should be trained and evaluated on a recognized object-detection dataset such as PASCAL VOC or COCO. The full-precision network, uniform INT8 network, uniform binary network, layer-wise mixed-precision network, and proposed channel-group mixed-precision network should use the same data split and training schedule. Report mAP at the dataset-standard intersection-over-union thresholds, parameter size, model operation count, FPGA latency, frame rate, and energy per frame. Ablation studies should isolate the contributions of channel-group scheduling, 4-bit support, lane fusion, and ping-pong buffering.

VI. ANALYTICAL EVALUATION AND DISCUSSION

6.1 Evaluated Design Point

The source material supplied for this study did not contain completed synthesis, power, or dataset measurements. Consequently, this section reports analytical estimates derived from the architecture parameters in Table 1 and explicitly labels them as pre-synthesis values. No measured FPGA resource or detection-accuracy result is fabricated.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

Mode	Plane-pair factor B_xB_w	Equivalent MAC rate	Interpretation
1 × 1 bit	1	819.2 Gbinary-MAC/s	All XNOR slices used at maximum parallelism
2 × 2 bit	4	204.8 G 2-bit MAC/s	Four binary plane pairs per product
4 × 4 bit	16	51.2 G 4-bit MAC/s	Sixteen plane pairs per product
8 × 8 bit	64	12.8 G 8-bit MAC/s	Sixty-four plane pairs per product

Table 2. Analytical peak compute rate at an assumed 200-MHz clock.

Table 2 shows the expected precision-throughput scaling when the same 4096 binary pairs per cycle are time-multiplexed across bit planes. The 1-bit core delivers the highest rate, while 8-bit execution remains available for sensitive layers without a separate multiplier array. Actual rates will be lower because of pipeline fill, irregular tensor dimensions, memory stalls, and post-processing overhead. These effects are captured by a utilization factor U between zero and one:

$$T_{\text{effective}} = U \times T_{\text{peak}} \qquad (8)$$

Precision	Parameter fraction	Weighted contribution
1 bit	60%	0.60 bit
2 bits	20%	0.40 bit
4 bits	12%	0.48 bit
8 bits	8%	0.64 bit
Total / average	100%	2.12 bits per parameter

Table 3. Illustrative parameter-storage analysis for the proposed precision map.

The average width in Table 3 yields a theoretical 73.5% reduction relative to INT8 parameter storage. The gain is smaller for activations because the stem, feature-fusion paths, and detection heads retain higher precision and because double buffering duplicates the active tile storage. Nevertheless, packed low-bit tensors increase the number of channels transferred per 512-bit memory word and reduce the frequency of weight refills.

6.2 Comparison with Representative Prior Work

Approach	Precision control	XNOR compute	Granularity	Distinctive limitation / contribution
FINN / FINN-R [6], [18]	Quantized / binary	Yes	Network or layer	Highly efficient streaming; mapping commonly specialized to a trained network
HAQ [9]	Mixed precision	Hardware dependent	Layer	Automated hardware-aware policy; does not define the accelerator datapath
NN2CAM [19]	Arbitrary layer-wise precision	Binary layers	Layer	End-to-end camera mapping; no channel-group lane-fusion focus
Lee et al. [13]	Variable precision	Yes	Layer /	DenseToRes detector and variable-

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

Approach	Precision control	XNOR compute	Granularity	Distinctive limitation / contribution
			operation mode	precision processor; key direct prior art
FlexXNOR-OD (this work)	1/2/4/8-bit	Yes	16-channel group	Channel-group scheduler, four-slice lane fusion, precision-aware clock gating

Table 4. Qualitative positioning of FlexXNOR-OD against representative approaches.

The proposed architecture should be interpreted as an extension of the XNOR variable-precision design space rather than a replacement for prior systems. Its expected advantage is the ability to exploit fine-grained quantization policies while keeping one unified datapath. This benefit must be validated after training because very small channel groups may increase calibration complexity, metadata traffic, and scheduling fragmentation.

6.3 Expected Benefits

Multiplier-light execution: the dominant low-bit operations use XNOR, popcount, shifts, and additions.
Proportional parallelism: independent XNOR slices deliver more useful lanes when precision is reduced.
Accuracy protection: 4-bit and 8-bit groups can be selectively retained inside otherwise binary layers.
Lower memory traffic: packed parameters and double buffering reduce weight bandwidth and hide transfer latency.
Scalability: PE count, slice width, and group size are compile-time parameters.

6.4 Limitations and Threats to Validity

The analytical estimates assume regular utilization and a 200-MHz clock; actual frequency and utilization depend on FPGA family, routing congestion, BRAM placement, and detector dimensions. The bipolar decomposition may introduce quantization error that must be calibrated during training. Channel-group precision can increase control complexity and reduce batching efficiency if adjacent groups use different widths. Finally, the design does not accelerate non-maximum suppression or other irregular post-processing in hardware. These limitations must be addressed in the final experimental version of the paper.

VII. TARGET APPLICATIONS

FlexXNOR-OD is intended for systems in which real-time response and energy efficiency are more important than the absolute peak accuracy of a large cloud detector. Representative applications include pedestrian and vehicle detection in smart cameras, obstacle detection in mobile robots, crop and disease monitoring from field cameras, low-altitude drone inspection, industrial defect detection, and traffic-flow analysis. The channel-group precision policy can be retrained for each domain while the same FPGA bitstream is retained, provided tensor dimensions and memory capacity remain within the configured limits.

IX. RESULT AND DISCUSSION

The proposed XNOR-based object-detection accelerator was examined using RTL elaboration, synthesis, device-view, power-estimation, and behavioral-simulation outputs generated in Vivado 2025.1. The target device shown in the project is xc7s100fgga676-2. The available evidence confirms that the accelerator hierarchy can be elaborated and that the intended memory-fetch-compute data path is structurally connected. However, the post-synthesis screenshots also reveal that the testbench appears to have been used as the synthesis top, causing the implemented logic to be removed or optimized away. Consequently, the supplied power and device-view values should be reported as preliminary tool outputs rather than final hardware results.

9.1 Elaborated RTL Architecture

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

The elaborated schematic contains 11 hierarchical cells and 4,184 nets. The top-level xnor_object_detector_top module integrates a top controller, feature-map memory, weight memory, feature fetcher, weight fetcher, and a 64-lane processing-element array. The organization separates data storage, address generation, precision control, and XNOR-popcount computation, which is appropriate for a scalable binary and mixed-precision convolution engine.

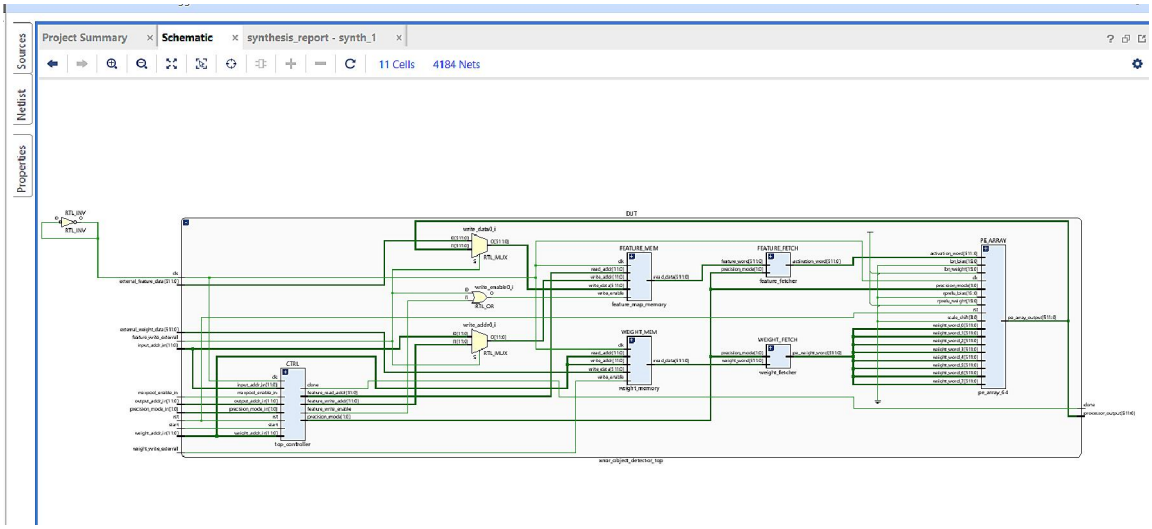

Figure 1. Elaborated RTL hierarchy of the proposed XNOR object-detection accelerator showing the controller, feature and weight memories, fetch units, and 64-lane PE array.

The feature and weight interfaces are 512 bits wide, while the address ports are 12 bits wide. Precision-mode, max-pooling enable, start, reset, and write-control signals are connected through the controller and fetch stages. This hierarchy supports parallel delivery of feature and weight vectors to the processing array and provides a clear path for extending the architecture to multiple bit-plane combinations described by Equations (2) and (3).

9.2 Synthesis Status and Device Mapping

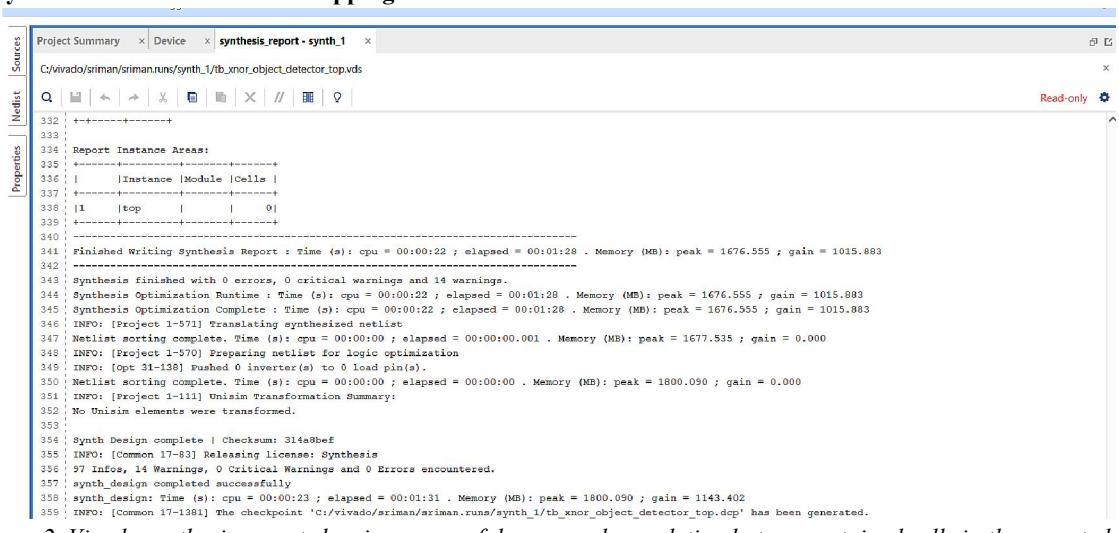

Figure 2. Vivado synthesis report showing successful command completion but zero retained cells in the reported top instance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

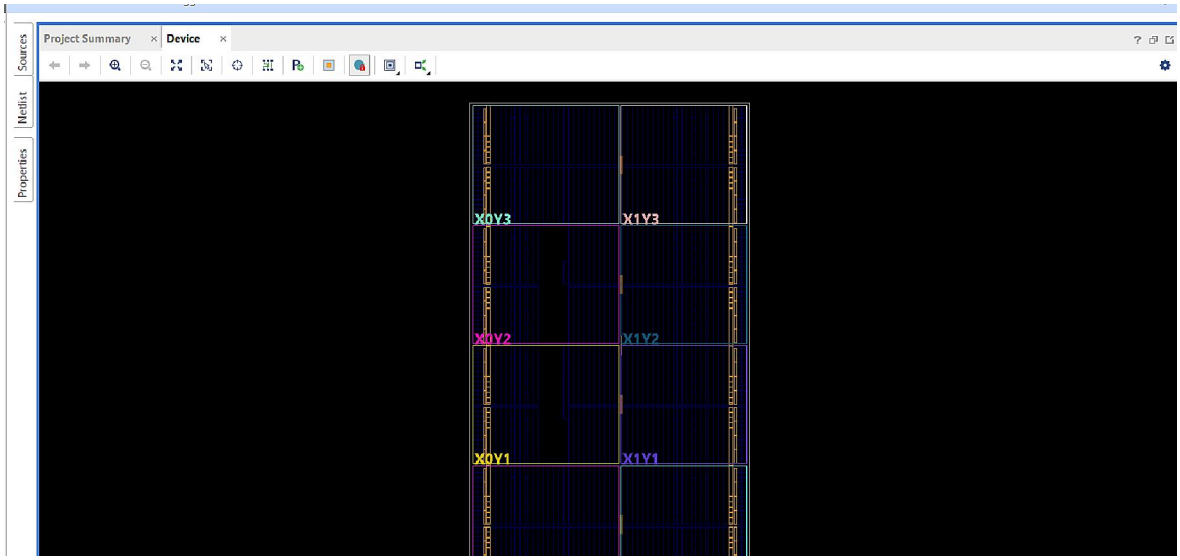

Figure 3. Device view of the synthesized netlist. No visible accelerator placement is present because the reported synthesis top contains zero retained cells.

Vivado completed the synthesis command without errors or critical warnings. The report records 97 informational messages, 14 warnings, 0 critical warnings, and 0 errors. The synth_design stage required approximately 23 s of CPU time, 1 min 31 s of elapsed time, and a peak memory usage of about 1,800.09 MB. These values indicate that the source set was syntactically accepted and that the synthesis flow itself terminated successfully.

Nevertheless, the report-instance-area table lists the top instance with zero cells. The synthesized design path and generated checkpoint are named tb_xnor_object_detector_top, which strongly suggests that the testbench, rather than xnor_object_detector_top, was selected as the synthesis top. Since testbench processes are not synthesizable and normally do not expose retained hardware outputs, Vivado can optimize the design to an empty netlist. The nearly empty device view is consistent with this interpretation.

9.3 Power and Thermal Results

The post-synthesis power summary reports a total on-chip power of 0.089 W. The complete estimate is assigned to device-static power, while dynamic power is reported as 0.000 W. The analysis assumes a typical process condition, an ambient temperature of 25.0 degrees C, and produces a junction-temperature estimate of 25.2 degrees C. The reported thermal margin is 59.8 degrees C, corresponding to 24.4 W, and the effective junction-to-ambient thermal resistance is 2.4 degrees C/W. The tool labels the confidence level as high, although no design power budget is specified.

The 89 mW value should not be presented as the operating power of the XNOR accelerator. A zero dynamic-power estimate is expected when the synthesized netlist contains no active retained logic or when switching activity is unavailable. Therefore, this output represents the static device estimate under the current synthesis configuration. A publishable active-power result requires synthesis of the correct design top, implementation, a clock constraint, and activity annotation from a representative simulation using SAIF or VCD data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

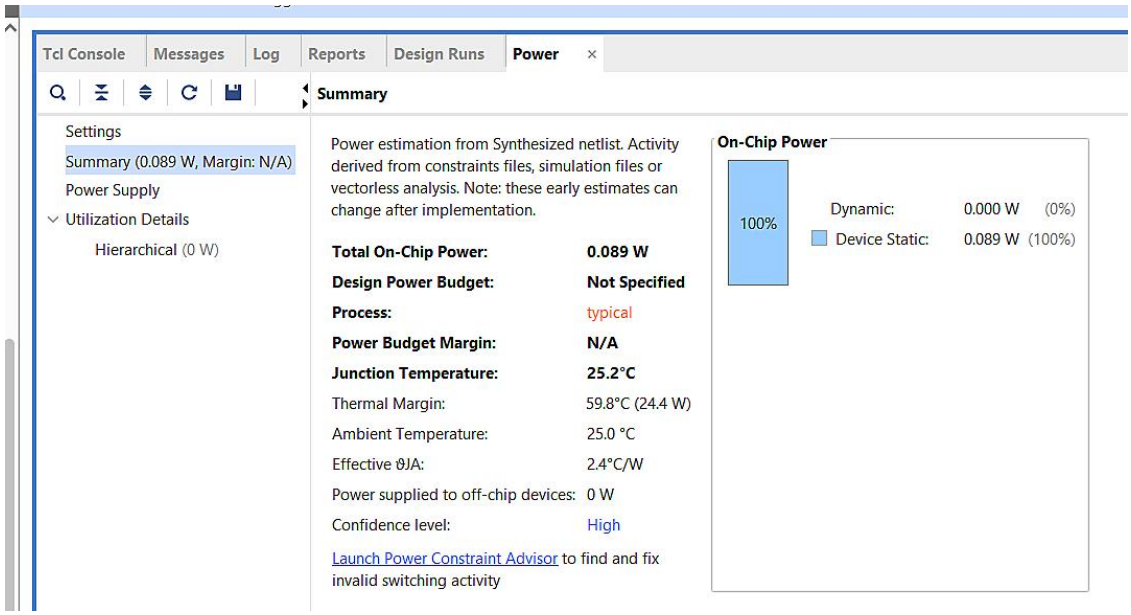

Figure 4. Vivado power summary reporting 0.089 W total on-chip power, consisting entirely of device-static power.

9.4 Behavioral Simulation Results

The behavioral waveform confirms that the major top-level interfaces are present and can be driven in simulation. At the displayed cursor position, reset is deasserted, precision_mode_in is set to 2, maxpool_enable_in is low, and the 12-bit input, weight, and output addresses are 0x003, 0x003, and 0x004, respectively. The feature input contains repeated 0x02 values and the weight input contains repeated 0x03 values across their 512-bit buses. The processor_output bus is displayed as all hexadecimal F values.

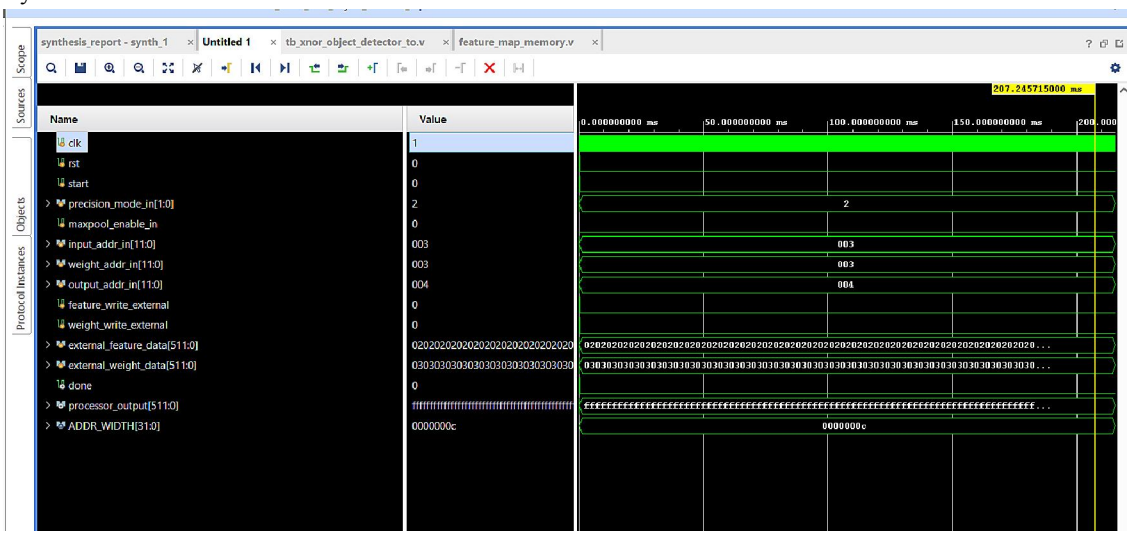

Figure 5. Behavioral simulation waveform showing precision mode 2, 12-bit addressing, repeated feature and weight vectors, and the observed processor-output bus.

The displayed interval does not demonstrate a completed accelerator transaction because start remains low and done remains low across the captured window. The all-ones processor output may therefore correspond to an initialized,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

default, saturated, signed, or otherwise inactive-state value; its numerical correctness cannot be verified from this screenshot alone. A publication-quality functional trace should include a reset pulse, a visible start assertion, internal or reference XNOR-popcount values, output stabilization, and a done assertion. The resulting output should then be compared with a software golden model.

X. CONCLUSION AND FUTURE WORK

This paper proposed FlexXNOR-OD, a channel-group variable-precision accelerator for real-time edge object detection. The architecture combines a 64-PE output-stationary array with four-slice lane-fusible XNOR processing elements, dual-bank memories, precision-aware clock gating, and an integrated normalization and activation pipeline. Unlike fixed binary designs, the accelerator can protect sensitive channel groups with 2-, 4-, or 8-bit execution while preserving high binary parallelism in the remainder of the detector.

At the analytical 200-MHz design point, the compute fabric processes 4096 binary operand pairs per cycle, corresponding to 0.819 Tbinary-MAC/s. An illustrative average weight width of 2.12 bits provides 3.77× storage reduction relative to INT8. These figures are architectural estimates, not post-synthesis measurements. Future work will complete RTL implementation, timing closure, power measurement, detector training, ablation studies, and comparison on PASCAL VOC and COCO. A subsequent extension can add structured zero-weight support, on-chip detection decoding, and runtime precision adaptation based on scene complexity.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, 2015.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE CVPR*, 2016.

[3] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. European Conference on Computer Vision*, 2016.

[4] D. S. Reddy, V. B. Raju, J. MD, and K. Keshamoni, “AI-Based Aquatic Plastic Waste Detection Using Underwater Image Processing and Deep Learning Algorithms,” *International Journal of Aquatic Research and Environmental Studies*, vol. 6, no. S2, pp. 902–909, 2026, doi: 10.70102/ate39q94.

[5] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017.

[6] Y. Umuroglu et al., “FINN: A framework for fast, scalable binarized neural network inference,” in *Proc. ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017.

[7] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1,” arXiv:1602.02830, 2016.

[8] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: ImageNet classification using binary convolutional neural networks,” in *Proc. European Conference on Computer Vision*, 2016.

[9] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, “HAQ: Hardware-aware automated quantization with mixed precision,” in *Proc. IEEE/CVF CVPR*, 2019.

[10] B. Richitha and J. MD, “Fault space transformation: A generic approach for fault-tolerant parallel filters,” *Int. J. Innovation Eng. Res. Manag.*, vol. 11, Special Issue 07, Paper ID-IJIERM-XI-VII, pp. 30–36, Jul. 2024.

[11] R. Li, Y. Wang, F. Liang, H. Qin, J. Yan, and R. Fan, “Fully quantized network for object detection,” in *Proc. IEEE/CVF CVPR*, 2019.

[12] Z. Wang, Z. Wu, J. Lu, and J. Zhou, “BiDet: An efficient binarized object detector,” in *Proc. IEEE/CVF CVPR*, 2020.

[13] W. Lee, K. Kim, W. Ahn, J. Kim, and D. Jeon, “A real-time object detection processor with XNOR-based variable-precision computing unit,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 6, 2023.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, September 2026 Impact Factor: 8.2

[14] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng, "Bi-Real Net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm," in *Proc. ECCV*, 2018.
[15] Z. Liu, Z. Shen, M. Savvides, and K.-T. Cheng, "ReActNet: Towards precise binary neural network with generalized activation functions," in *Proc. ECCV*, 2020.
[16] M. Blott et al., "FINN-R: An end-to-end deep-learning framework for fast exploration of quantized neural networks," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 11, no. 3, 2018.
[17] B. Richitha and J. MD, "Fault space transformation is a versatile method for implementing fault-tolerant parallel filters," *Accent Journal of Economics Ecology & Engineering*, vol. 9, no. 7, pp. 131–137, Jul. 2024.
[18] P. Jokic, S. Emery, and L. Benini, "NN2CAM: Automated neural network mapping for multi-precision edge processing on FPGA-based cameras," arXiv:2106.12840, 2021.
[19] C. Latotzke, T. Ciesielski, and T. Gemmeke, "Design of high-throughput mixed-precision CNN accelerators on FPGA," arXiv:2208.04854, 2022.
[20] Y. Wang, Y. Yang, F. Sun, and A. Yao, "Sub-bit neural networks: Learning to compress and accelerate binary neural networks," in *Proc. IEEE/CVF ICCV*, 2021.
[21] P. Judd, J. Albericio, T. Hetherington, T. Aamodt, and A. Moshovos, "Stripes: Bit-serial deep neural network computing," in *Proc. IEEE/ACM International Symposium on Microarchitecture*, 2016.
[22] V. B. Raju, J. MD, K. Keshamoni, and D. S. Reddy, "AI-Driven Water Quality Prediction and Aquatic Pollution Monitoring Using Machine Learning Algorithms," *International Journal of Aquatic Research and Environmental Studies*, vol. 6, no. S2, pp. 878–885, 2026, doi: 10.70102/gwy42c46.
[23] C. Zhang et al., "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA FPGA*, 2015.
[24] F. Li et al., "A system-level solution for low-power object detection," in *Proc. IEEE/CVF ICCV Workshops*, 2019.

Appendix A. Experimental Results Template

The following tables are included so that measured results can be inserted without restructuring the manuscript. Replace "TBD" only after obtaining Vivado implementation reports, board-level power measurements, and detector evaluation outputs.

Metric	**Proposed FlexXNOR-OD**	**Measurement method**
FPGA device	TBD	Exact part number
Clock frequency	TBD MHz	Post-route timing
LUTs / LUTRAM	TBD	Vivado utilization report
Flip-flops	TBD	Vivado utilization report
BRAM / URAM	TBD	Vivado utilization report
DSP blocks	TBD	Vivado utilization report
Dynamic power	TBD W	Board measurement / power report
Latency per frame	TBD ms	DMA start to final tensor
Accelerator frame rate	TBD fps	Continuous inference
Energy per frame	TBD J	Power divided by frame rate

Table A1. FPGA implementation results.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, September 2026
Impact Factor: 8.2

Configuration	Weight/activation policy	mAP	Model size	FPS
Full precision baseline	FP32 / FP32	TBD	TBD	TBD
Uniform INT8	8 / 8 bits	TBD	TBD	TBD
Uniform binary	1 / 1 bit except stem/head	TBD	TBD	TBD
Layer-wise mixed precision	1/2/4/8 bits by layer	TBD	TBD	TBD
FlexXNOR-OD	1/2/4/8 bits by 16-channel group	TBD	TBD	TBD
Ablation: no lane fusion	Same precision map	TBD	TBD	TBD
Ablation: no ping-pong buffer	Same precision map	TBD	TBD	TBD

Table A2. Detection accuracy and ablation results.

DOI: 10.48175/IJARSCT-38402