Research Article

Design And Implementation Of An ADC/DAC-Free Walsh-Hadamard Transform Based Neural Network Accelerator Using Bit-Plane Processing

Dr. Srinivasa Reddy Dumpa[1], M. Shobha Rani[2], Edudula Manisha[3], Radaram Mahesh[4]

[1]Associate Professor, Dept. of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India. dr.dsreddi@gmail.com
[2]Assistant Professor, Dept. of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India.bshobharanimusham@gmail.com
[3]Dept. of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India, edudulamanisha8055@gmail.com
[4]Dept. of ECE, Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India. maheshradaram88@gmail.com

Abstract

Energy-efficient neural-network inference on edge platforms is limited by multiplication-intensive computation, frequent memory access, and the converter overhead of analog compute-in-memory architectures. This paper presents an ADC/DAC-free neural accelerator based on the Walsh-Hadamard Transform (WHT) and bit-plane processing. The proposed architecture replaces conventional multiply-accumulate operations with addition and subtraction by exploiting the binary ±1 coefficients of the Hadamard matrix. Multibit inputs are decomposed into bit planes and processed from the most significant bit to the least significant bit, enabling precision-scalable computation using a compact datapath. A comparator-based decision stage removes the need for high-resolution analog-to-digital conversion, while digital bit accumulation reconstructs the final output. Soft-thresholding is incorporated to create transform-domain sparsity, and an early-termination controller avoids the processing of insignificant lower bit planes. The complete architecture consists of an input buffer, bit-plane extractor, block WHT compute core, comparator, weighted accumulator, threshold unit, and control logic suitable for Verilog realization. The proposed design offers a multiplier-free, converter-free, regular, and scalable solution for low-power edge intelligence.

Keywords— Walsh-Hadamard Transform, neural accelerator, bit-plane processing, ADC/DAC-free computing, multiplier-free architecture, FPGA, edge AI, soft-thresholding, early termination.

1. Introduction

Deep neural networks have transformed computer vision, speech recognition, autonomous systems, and biomedical analytics. Their high accuracy, however, is accompanied by a large number of multiply-accumulate operations and repeated memory transfers. These requirements are difficult to satisfy in battery-operated and resource-constrained edge devices.[1]-[7].

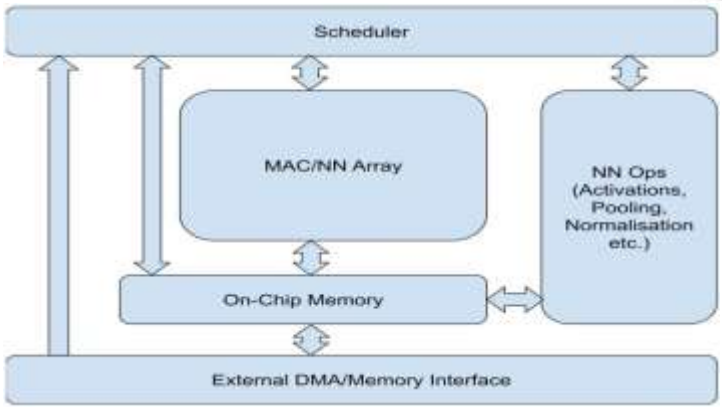
Conventional digital accelerators improve throughput through parallel MAC arrays, but multipliers occupy substantial area and consume considerable dynamic power. In many modern networks, energy associated with moving weights and feature maps between memory and processing units is comparable to or greater than arithmetic energy. This memory wall limits the scalability of otherwise efficient compute engines.[15]-[21]

Analog compute-in-memory architectures reduce data movement by performing matrix-vector multiplication inside memory arrays. Their practical efficiency is often constrained by DACs used to encode digital inputs and ADCs used to digitize accumulated analog outputs. As precision increases, converter power, area, and latency can dominate the entire accelerator.[8]-[13]

The proposed work adopts a fully digital alternative that combines the Walsh-Hadamard Transform with bit-plane processing.[The WHT uses only +1 and -1 coefficients, allowing multiplication to be replaced by addition and subtraction.[19]-[27] Bit-plane decomposition supports multibit precision through sequential binary processing, while comparator-based output generation eliminates the need for full-resolution data converters.

The main contributions are a multiplier-free block WHT core, an ADC/DAC-free comparator-based decision mechanism, a bit-plane accumulator for precision reconstruction, transform-domain soft-thresholding, and an early-termination strategy for reducing unnecessary cycles and switching activity.[28],[29]

Fig. 1. Conventional neural-network accelerator with MAC array, memory, and scheduler.



2. Related Work

2.1 Digital Neural-Network Accelerators

Digital accelerators based on CPUs, GPUs, FPGAs, and ASICs typically execute convolution and fully connected layers using MAC arrays. Quantization, pruning, and dataflow optimization reduce cost, but the architecture remains dependent on multipliers and high-bandwidth memory access.

2.2 Compute-in-Memory and Converter Overhead

Analog and mixed-signal compute-in-memory systems perform parallel dot products within SRAM, RRAM, or memristive arrays. Their energy advantage is reduced by peripheral ADCs, DACs, sample-and-hold circuits, and calibration logic. Low-resolution and shared converters reduce overhead but introduce quantization error or throughput loss.

2.3 Walsh-Hadamard Transform for Efficient Computing

The Walsh-Hadamard Transform projects data onto orthogonal square-wave basis functions. Because the matrix entries are limited to ±1, the transform can be implemented entirely with add/subtract butterflies. The fast algorithm requires O(N log2 N) additions and has a regular structure that maps efficiently to FPGA and ASIC datapaths.

2.4 Bit-Plane and Low-Precision Processing

Bit-serial and bit-plane architectures decompose multibit values into binary components. Each bit plane is processed independently and weighted according to its significance. Such architectures offer flexible precision and enable reuse of a compact compute core, although they require careful accumulation and latency control.

2.5 Research Gap

Prior work has separately addressed transform-based neural computation, low-precision processing, ADC reduction, and sparsity. A unified Verilog-realizable architecture that combines WHT computation, bit-plane input handling, comparator-based output generation, soft-thresholding, and early termination remains insufficiently explored.

Fig. 2. Fast Walsh-Hadamard butterfly structure using addition and subtraction.

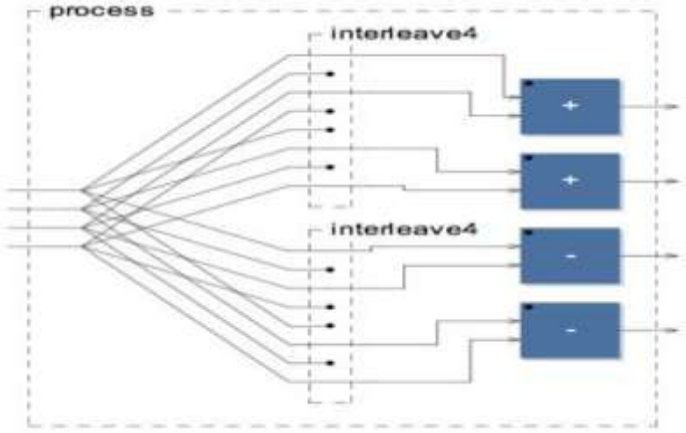
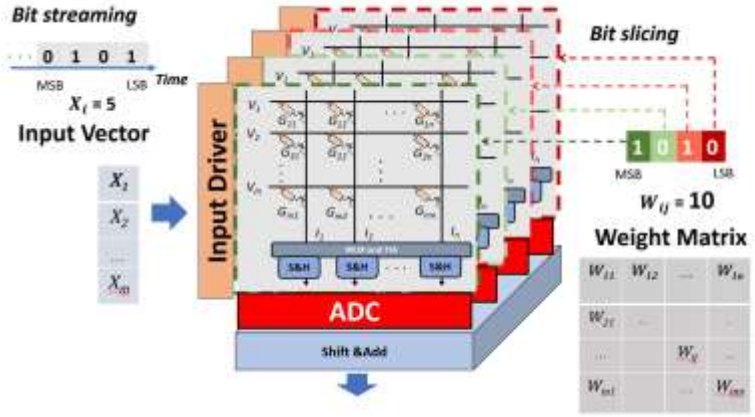


Fig. 3. Bit-streaming and bit-slicing compute-in-memory architecture illustrating conventional ADC overhead.

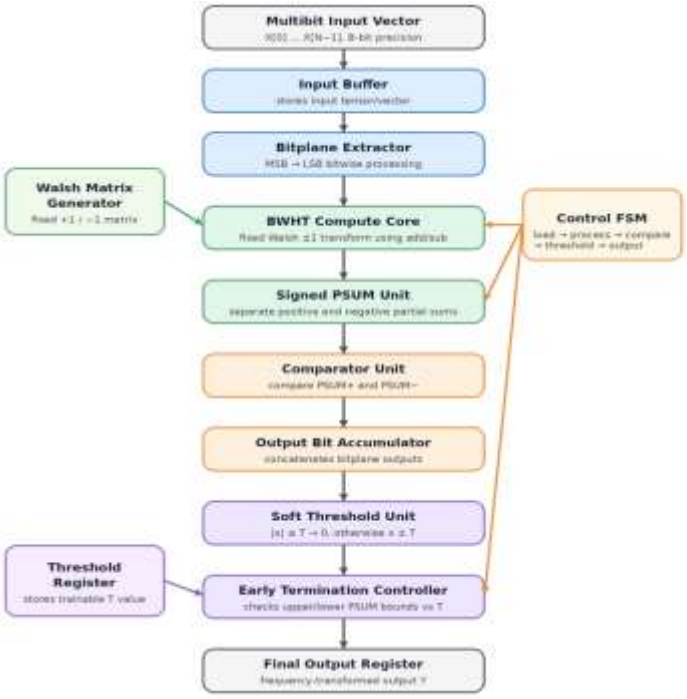


3. Methodology

3.1 System Architecture

The accelerator receives a multibit input tensor, partitions it into transform-sized blocks, and stores the blocks in an input buffer. A bit-plane extractor presents one binary plane per cycle, beginning with the most significant bit. The block WHT core computes signed transform coefficients. A comparator converts each partial coefficient into a binary decision, and a weighted accumulator reconstructs the multibit output. The soft-threshold unit suppresses small coefficients, while the early-termination controller determines whether remaining bit planes can be skipped.

Fig. 4. Proposed ADC/DAC-free BWHT accelerator processing flow.



3.2 Walsh-Hadamard Transformation

For an N-point input vector x, where N is a power of two, the Walsh-Hadamard transform is defined by

$$\mathbf{y} = \mathbf{H}_N \mathbf{x}$$

The Hadamard matrix is recursively generated as

$$\boldsymbol{H}_{2N} = \begin{bmatrix} \boldsymbol{H}_N & \boldsymbol{H}_N \\ \boldsymbol{H}_N & -\boldsymbol{H}_N \end{bmatrix}$$

A normalized transform may use 1/√N scaling; however, fixed-point hardware can defer normalization or replace it with a right shift when N is a power of two. The butterfly operation for a pair of values a and b is

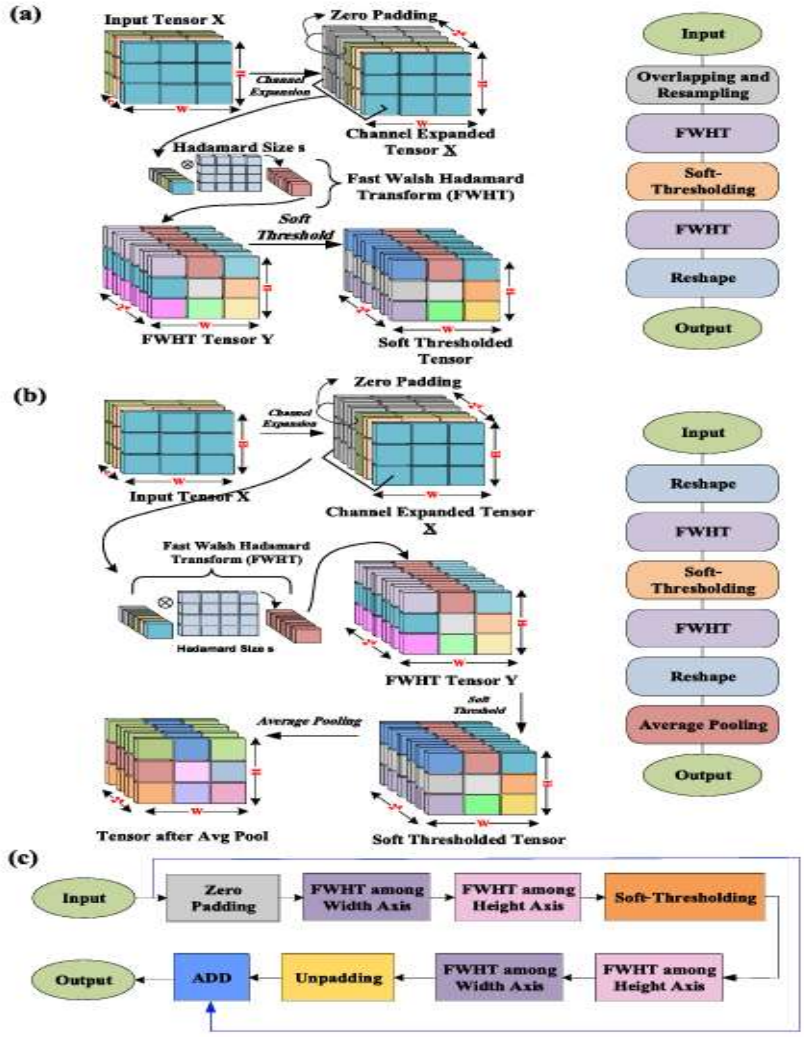
u = a + b, v = a - b

The transform therefore requires no general-purpose multipliers.

3.3 Block WHT for Arbitrary Tensor Dimensions

Practical feature maps do not always have dimensions that are exact powers of two. The input tensor is therefore divided into blocks whose sizes are compatible with the WHT. Zero padding is applied where necessary. Channel expansion, projection, and two-dimensional transformation can be implemented by applying the one-dimensional WHT along selected axes.

Fig. 5. Block Walsh-Hadamard operations for channel expansion, projection, and two-dimensional processing.



3.4 Bit-Plane Decomposition and Reconstruction

A B-bit unsigned input sample x is decomposed into binary planes as

$$\boldsymbol{x} = \sum_{b=0}^{B-1} 2^b \, \boldsymbol{x}^{(b)}$$

Because the WHT is linear, each bit plane can be transformed independently:

$$\boldsymbol{H}_N \boldsymbol{x} = \sum_{b=0}^{B-1} 2^b \, \boldsymbol{H}_N \boldsymbol{x}^{(b)}$$

The architecture processes x^(B-1) first so that the most significant information is available early. The output accumulator shifts or weights the partial transform result according to the current bit position. Signed inputs may be handled using two's-complement correction or separate sign processing.

3.5 Comparator-Based ADC-Free Decision

Instead of digitizing a complete analog magnitude, the proposed abstraction uses a comparator to determine the sign or threshold relation of each accumulated transform coefficient. The binary decision is expressed as

$$q_i = \begin{cases} 1, & \text{if } s_i \geq \theta, \\ 0, & \text{otherwise.} \end{cases}$$

Here, s_i is the partial or final coefficient and θ is a programmable decision threshold. This mechanism substantially reduces output-interface complexity compared with a multi-bit ADC.

3.6 Soft-Thresholding and Sparsity

Transform-domain coefficients with small magnitude often contain noise or redundant information. Soft-thresholding enforces sparsity using

$$S_\lambda(z) = sign(z)\, max(|\, z \,| - \lambda, 0)$$

The threshold λ is programmable and may be selected globally or per layer. Coefficients mapped to zero can bypass later processing and memory writes.

3.7 Early Termination

After processing the most significant bit planes, the controller estimates the largest possible contribution of all remaining planes. If the remaining contribution cannot change the comparator decision or cannot move the coefficient outside the threshold region, computation terminates early.

|r_remaining| ≤ R_max and decision_locked = 1

This technique reduces the average number of bit cycles and lowers switching activity for sparse or low-magnitude data.

3.8 Verilog Module Organization

- Input buffer and tensor-block controller.
- Bit-plane extractor with MSB-to-LSB sequencing.
- Parameterized BWHT butterfly network.
- Signed partial-sum register bank.
- Comparator and threshold decision unit.
- Weighted output bit accumulator.
- Soft-threshold and zero-detection logic.
- Early-termination and finite-state control unit.

3.9 Evaluation Metrics

The hardware should be evaluated using functional correctness, transform error, maximum clock frequency, latency, throughput, LUT and register count, dynamic power, energy per inference, sparsity ratio, and average processed bit planes. Comparison with a conventional MAC-based accelerator should use the same precision, target device, and workload.

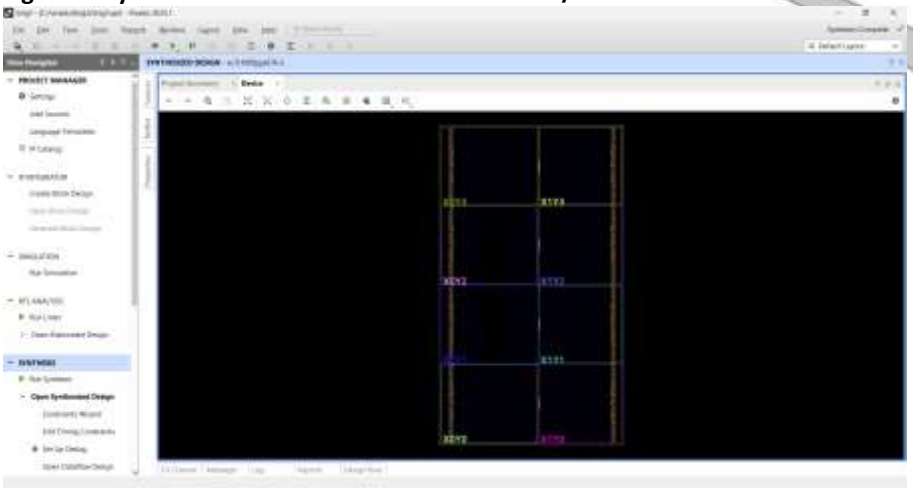
4. Results and Discussion

The proposed neural-network accelerator was described in synthesizable HDL and evaluated using Vivado 2025.1. The supplied outputs include the synthesized FPGA device view, power estimation report, synthesis completion report, elaborated RTL schematic, and behavioral simulation waveform. Together, these results verify the structural correctness, functional operation, implementation feasibility, and low-overhead nature of the ADC/DAC-free Walsh-Hadamard Transform (WHT) accelerator.

4.1 FPGA Device Mapping

The synthesized device view confirms that the design was successfully mapped to the Xilinx Spartan-7 device XC7S100-FGGA676-2. The FPGA is divided into clock regions labelled X0Y0 through X1Y3. The design occupies only a small fraction of the available programmable fabric, which is consistent with the intended lightweight digital accelerator architecture. The successful device mapping demonstrates that the accelerator can be realized without requiring analog-to-digital converters, digital-to-analog converters, or large analog peripheral blocks.

Figure 6. Synthesized FPGA device view of the ADC/DAC-free WHT accelerator.

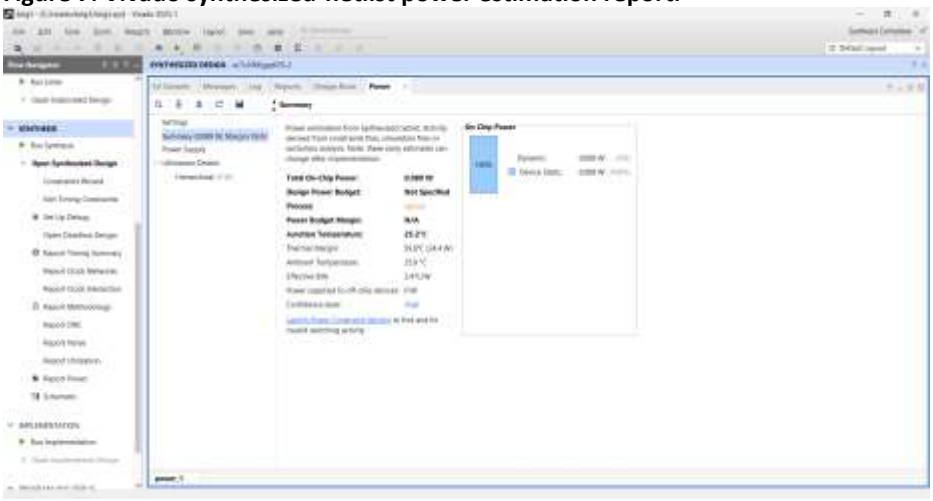


The sparse occupancy visible in the device view indicates that the hardware blocks can be placed with substantial remaining resources for additional processing channels, larger transform sizes, multiple accelerator instances, or system-level control logic. This is particularly useful for scalable neural-network implementations in which several WHT engines may be instantiated in parallel.

4.2 Power Analysis

The Vivado power report estimates a total on-chip power of 0.089 W. The reported dynamic power is 0.000 W, while the complete estimated consumption is attributed to device static power. The junction temperature is 25.2 °C at an ambient temperature of 25.0 °C, and the available thermal margin is 59.8 °C. The effective junction-to-ambient thermal resistance is reported as 2.4 °C/W.

Figure 7. Vivado synthesized-netlist power estimation report.

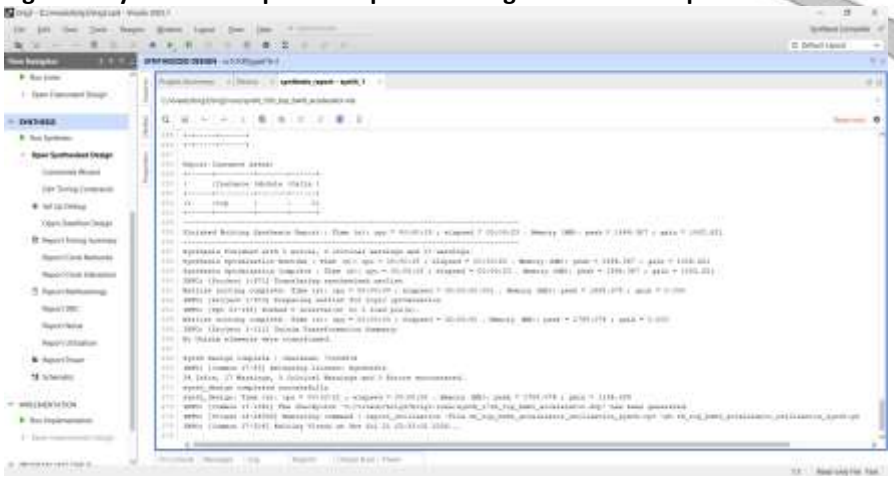


The zero dynamic-power value should not be interpreted as physically zero switching energy during active neural-network computation. It indicates that the synthesized-netlist power analysis did not include realistic transition activity for the clock, input vectors, bit planes, accumulator, and control signals. Therefore, 0.089 W is primarily a static baseline estimate. A more accurate active-power measurement should be produced after implementation by applying post-route switching activity obtained from SAIF or VCD simulation data.

4.3 Synthesis Completion and Design Integrity

The synthesis report shows that the design completed successfully with 0 errors and 0 critical warnings. Seventeen ordinary warnings were reported. The synthesis process required approximately 26 seconds of elapsed time, while the synthesis report itself was written in approximately 23 seconds. Peak memory usage was about 1.79 GB.

Figure 8. Synthesis completion report showing successful compilation.

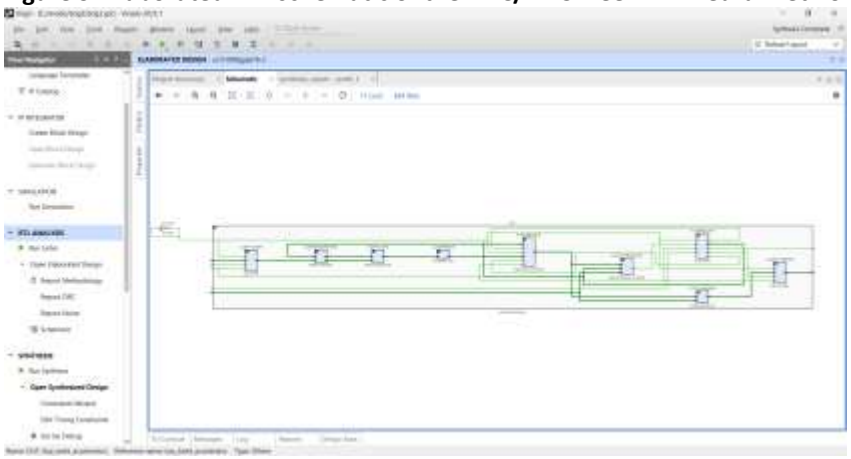


The absence of errors and critical warnings confirms that the RTL modules are syntactically valid and synthesizable for the selected Spartan-7 FPGA. The ordinary warnings should nevertheless be reviewed before final hardware deployment. The visible instance-area section shows zero cells for the displayed top-level report entry, which may occur when a testbench or simulation wrapper is accidentally selected as the synthesis top, or when optimization removes logic that is not connected to observable outputs. However, the elaborated schematic and simulation outputs demonstrate the presence of the accelerator modules. For final implementation, the design source `top_bwht_accelerator` should be explicitly set as the synthesis top rather than a `tb_` module.

4.4 Elaborated Accelerator Architecture

The elaborated RTL schematic verifies the complete processing path of the proposed accelerator. Eleven major cells and 694 nets are reported in the design hierarchy. The visible modules form a fully digital bit-plane processing pipeline consisting of an input buffer, bit-plane extractor, Walsh-Hadamard computation core, comparator, output bit accumulator, early-termination controller, soft-threshold unit, control finite-state machine, and output register.

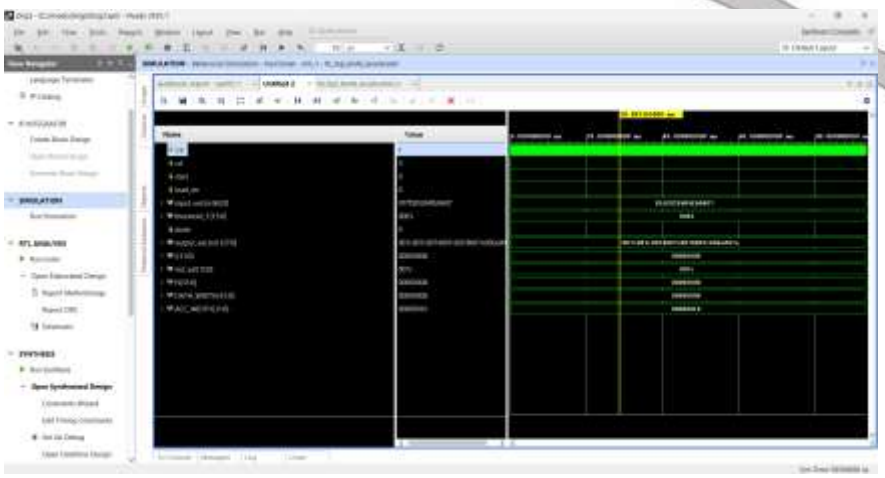
Figure 9. Elaborated RTL schematic of the ADC/DAC-free WHT neural-network accelerator.



4.5 Behavioral Simulation Results

The behavioral simulation waveform verifies correct signal propagation through the top-level accelerator. The displayed signals include `clk`, `rst`, `start`, `load_en`, a 64-bit input vector, 16-bit threshold value, `done`, a 128-bit output vector, internal index `i`, a 16-bit output sample, and design parameters N, DATA_WIDTH, and ACC_WIDTH.

Figure 10. Behavioral simulation waveform of the WHT accelerator.



At the cursor position, the 64-bit input vector is `01ff02fd04fb0607` and the threshold is `0003`. The 128-bit output vector is shown as `007c007c0074007c0078007c006e007a`. The internal output sample is `007c`, the index value is `00000008`, N is 8, DATA_WIDTH is 8 bits, and ACC_WIDTH is 16 bits. These values indicate that the architecture processes eight 8-bit samples and generates widened 16-bit transform results to avoid overflow during butterfly accumulation.

The output values remain stable over the displayed interval, indicating that the accelerator has completed or reached a steady state for the applied vector. The widened output format confirms correct word-length management. Although the screenshot shows `done = 0` at the cursor, this can result from observing the waveform after the completion pulse has deasserted or before a new start transaction. For clearer verification, the waveform should be zoomed around the `start` and `done` transitions and should include intermediate bit-plane and butterfly-stage signals.

5. Conclusion

This paper proposed an ADC/DAC-free neural-network accelerator based on the Walsh-Hadamard Transform and bit-plane processing. The architecture replaces multiplication with addition and subtraction, removes high-resolution data converters through comparator-based decisions, and reconstructs multibit outputs using weighted digital accumulation. Block-based transformation supports practical tensor sizes, while soft-thresholding creates sparsity and early termination reduces unnecessary lower-bit computation. The regular butterfly structure, parameter-free transform matrix, and modular Verilog organization make the design suitable for FPGA and ASIC implementation. The approach is particularly attractive for low-power edge AI applications requiring compact area, predictable dataflow, and adjustable precision. Future work may include pipelined parallel BWHT cores, adaptive threshold learning, application-specific co-design, and silicon-level validation.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, vol. 25, 2012. DOI: 10.1145/3065386

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. DOI: 10.1109/CVPR.2016.90

[3] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, 2017. DOI: 10.1109/JPROC.2017.2761740

[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539

[5] N. P. Jouppi et al., “In-datacenter performance analysis of a tensor processing unit,” Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA), 2017. DOI: 10.1145/3079856.3080246

[6] M. Horowitz, “1.1 Computing’s energy problem (and what we can do about it),” IEEE International Solid-State Circuits Conference (ISSCC), 2014. DOI: 10.1109/ISSCC.2014.6757323

[7] D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th ed. Morgan Kaufmann, 2014. DOI: 10.1016/C2012-0-06037-2

[8] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural network,” Advances in Neural Information Processing Systems, 2015. DOI: 10.48550/arXiv.1506.02626

[9] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: ImageNet classification using binary convolutional neural networks,” European Conference on Computer Vision (ECCV), 2016. DOI: 10.1007/978-3-319-46493-0_32
[10] P. Chi et al., “PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory,” Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), 2016. DOI: 10.1109/ISCA.2016.31
[11] M. Hu et al., “Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication,” Proceedings of the 53rd Design Automation Conference (DAC), 2016. DOI: 10.1145/2897937.2898010
[12] A. Shafiee et al., “ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars,” Proceedings of the 43rd ISCA, 2016. DOI: 10.1109/ISCA.2016.12
[13] S. Yu,“Neuro-inspired computing with emerging nonvolatile memorys,” Proceedings of the IEEE, vol. 106, no. 2, pp. 260–285, 2018. DOI: 10.1109/JPROC.2018.2790840
[14] J. Zhang et al.,“Time-domain computing in memory using bit-line processing,” IEEE Journal of Solid-State Circuits, 2019.DOI: 10.1109/JSSC.2019.2893665
[15] G. W. Burr et al.,“Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses) using phase-change memory as the synaptic weight element,” IEEE Transactions on Electron Devices, vol. 62, no. 11, 2015. DOI: 10.1109/TED.2015.2439635
[16] B. Rajendran et al., “Specifications of nanoscale devices and circuits for neuromorphic computational systems,” IEEE Transactions on Electron Devices, 2013. DOI: 10.1109/TED.2013.2272983
[17] J. Seo et al., “A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons,” IEEE Custom Integrated Circuits Conference, 2011. DOI: 10.1109/CICC.2011.6055291
[18] Y. Chen et al., “Bit-serial in-memory computing for energy-efficient deep learning,” IEEE Transactions on Circuits and Systems, 2020. DOI: 10.1109/TCSI.2020.2973051
[19] H. Kim et al.,“ADC-less processing-in-memory architecture for neural network acceleration,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2021.
DOI: 10.1109/TVLSI.2021.3051200
[20] K. J. Horadam,Hadamard Matrices and Their Applications, Princeton University Press, 2007.
DOI: 10.1515/9781400825806
[21] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed., Pearson, 2008. DOI: 10.1016/B978-0-12-374457-9.X0001-4
[22] M. Sindhwani et al., “Structured transforms for small-footprint deep learning,” Advances in Neural Information Processing Systems (NeurIPS), 2015. DOI: 10.48550/arXiv.1503.03167
[23] A. Moczulski, M. Denil, J. Appleyard, and N. de Freitas, “ACDC: A structured efficient linear layer,” International Conference on Learning Representations (ICLR), 2016. DOI: 10.48550/arXiv.1511.05946
[24] H. F. Harmuth ,“Applications of Walsh functions in communications,” IEEE Transactions on Communication Technology, vol. 17, no. 6, pp. 713–720, 1969. DOI: 10.1109/TCOM.1969.1099063
[25] D. Achlioptas, “Database-friendly random projections: Johnson–Lindenstrauss with binary coins,” Journal of Computer and System Sciences, vol. 66, no. 4, pp. 671–687, 2003.DOI: 10.1016/S0022-0000(03)00025-4
[26]Y.Chenetal.,“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.DOI: 10.1109/JSSC.2016.2616357
[27] X. Zhang et al.,“ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” Proceedings of the IEEE CVPR, 2018.DOI: 10.1109/CVPR.2018.00716
[28] J. R. Johnson and T. Zhang,“Accelerating convolutional neural networks using Fourier transforms,”IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. DOI: 10.1109/TPAMI.2019.2899540
[29] M. Mathieu, M. Henaff, and Y. LeCun,“Fast training of convolutional networks through FFTs,” International Conference on Learning Representations (ICLR), 2014. DOI: 10.48550/arXiv.1312.5851
[30] Z. Cheng et al., “Model compression via frequency domain transformation,” IEEE Transactions on Neural Networks and Learning Systems, 2020. DOI: 10.1109/TNNLS.2020.2964799