UDC 004.382.2; 004.312 **doi: 10.26906/SUNZ.2026.2.243**

Dmytro Salnikov, Oleg Vasylchenkov

National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine

AREA-EFFICIENT HARDWARE MODULES FOR FP16/FP8/FP32 FORMAT CONVERSION IN EMBEDDED SYSTEMS

Abstract. The rapid proliferation of neural networks in embedded and edge computing systems has led to an increasing demand for efficient hardware implementations that can support precision-scalable arithmetic. Applications such as autonomous vehicles, intelligent sensors, and industrial automation require high computational performance, low latency, and strict energy constraints. Floating-point arithmetic, defined by the IEEE 754 standard, remains the dominant numerical representation in such systems due to its versatility and broad dynamic range. However, deploying modern deep learning models on resource-limited platforms poses significant challenges in balancing accuracy, throughput, and hardware footprint. To address these challenges, emerging reduced-precision formats such as FP16, BF16, and FP8 (E4M3, E5M2) have gained popularity for both inference and training, enabling decreased memory bandwidth and improved energy efficiency with minimal accuracy degradation. Despite their growing prevalence, many microcontrollers and FPGAs lack native hardware support for these low-precision formats, motivating the need for compact and reconfigurable conversion modules capable of bridging compatibility with conventional FP32 processing units. This work presents the design, implementation, and hardware evaluation of fully synthesizable VHDL modules for converting between FP8, FP16, BF16, and standard IEEE-754 single-precision (FP32) formats. The proposed architecture leverages FPGA Look-Up Tables (LUTs) to perform exponent and mantissa field manipulation, bias adjustment, and classification of special numerical cases such as Infinity and NaN, ensuring full standard compliance. The converters were synthesized using a commercial design flow targeting an Intel Cyclone V device. Experimental results demonstrate exceptionally low resource utilization and high operating frequency, with the FP8E4M3 and FP8E5M2 converters each requiring only 14 ALMs while achieving frequencies exceeding 500 MHz. These outcomes confirm the suitability of the proposed modules for deployment in mixed-precision computing systems and embedded neural network accelerators, providing an efficient hardware foundation for energy-aware and high-performance AI workloads on constrained platforms.

Keywords: floating-point formats, reduced-precision number representation, embedded systems, edge computing, FPGA, VHDL, embedded neural network acceleration, area-efficient architecture.

Introduction

Nowadays, neural networks are increasingly integrated into modern embedded systems, enabling intelligent features such as real-time object detection, speech recognition, and sensor data analysis directly on edge devices. These capabilities underpin a wide range of applications, including autonomous driving, smart cameras, wearables, and industrial IoT, where low latency and energy efficiency are critical. Deploying such models on resource-constrained hardware requires optimized computation and compact data formats making precision-scalable representations like FP16, FP8, and FP32 essential for balancing accuracy, performance, and footprint.

Floating-point arithmetic, standardized by IEEE 754, remains the most versatile and widely used method for numerical computation in modern systems. Among the available formats, “single precision” offers an effective balance between precision, dynamic range, and implementation cost. It serves as the standard choice FP type for most embedded system ICs manufacturers. At the same time, many applications still rely on fixed-point math usage [1].

Background and Related Work

Although FP8 formats are relatively recent (introduced in 2022 in [2]), they have rapidly gained adoption in contemporary neural network architectures. The use of FP8 and FP16 numerical formats has become increasingly prevalent in contemporary neural network workloads.

These low-precision formats allow models to be quantized to a smaller bit-width without a dramatic loss in accuracy, reducing both memory bandwidth and storage requirements.

FP8, in particular, offers a favorable trade-off between dynamic range and computational efficiency, enabling faster inference and lower energy consumption on hardware that supports SIMD or specialized multiply-accumulate units. FP16 remains a common choice for training and fine-tuning tasks because of its wider range and compatibility with existing accelerators. Tradeoffs of INT8 vs FP8 usage are described in [3]. Research in [4] presents a comparison of U-Net performance and memory efficiency across various data representation formats, including FP32, FP16, and INT8 quantization.

Despite their advantages, many microcontroller devices lack native FP8 or even FP16 support, relying instead on integer data types. This limitation motivates the need for dedicated, hardware-optimized conversion modules that can bridge the gap between the neural-network-friendly FP8/FP16 formats and the native FP32 or integer representations found on embedded processors.

A mixed-precision ALU unit and related architectures were introduced in [5], [6] and [7]. Despite demonstrating strong performance characteristics, the use of low-precision computations remains rather limited. Moreover, it leads to increased resource consumption. Most low-precision floating-point operations used in neural network frameworks use FP32 calculations internally.

The main goal of this work is to design and evaluate compact, fully synthesizable "cast-only" hardware modules for efficient conversion between compact floating-point data formats and IEEE-754 single-precision format, targeting resource-constrained embedded and FPGA-based systems.

Architecture of LUT-Based Floating-Point Conversion Modules

Currently, the number of parameters in modern neural network architectures may vary significantly across different model families and configurations (Table 1).

This variability has a direct impact on the overall memory footprint of the system.

Table 1 – **Number of parameters for common neural network topologies**

Model	Number of parameters
wav2vec 2.0 (base)	95 million [8]
wav2vec 2.0 (large)	317 million [8]
Whisper (tiny)	39 million
Whisper (base)	74 million
Whisper (small)	244 million
Whisper (medium)	769 million
Whisper (large-v2 / large-v3)	1.55 billion
YOLOv5 (small)	7.2 million
YOLOv5 (large)	46.5 million
LLaMA 3	from 8M up to 405 billions
Mixtral 8x7B	from 8M up to 140 billions [9]

Moreover, storage requirements are determined not only by the parameters themselves but also by the intermediate activation values generated during computation, both of which contribute substantially to the total memory demand. As a result, understanding parameter count and activation behavior is essential when designing compute- and memory-efficient hardware modules for embedded deployments.

Quantization is often employed to reduce the memory footprint of neural network models by representing parameters and activations with lower-precision numerical formats. While this approach can substantially decrease storage requirements and improve computational efficiency (as was shown in [10]), it is not universally achievable.

Certain models or layers exhibit sensitivity to reduced precision, leading to significant degradation in accuracy or instability during inference. Moreover, some operations require higher numerical fidelity to preserve convergence or maintain representational capacity. As a result, although quantization is a powerful technique for shrinking memory usage, its applicability depends on the model architecture, task constraints, and tolerance for accuracy loss.

At the moment, several floating-point formats are in use across modern machine-learning and hardware platforms (Table 2). Beyond the conventional FP32 format, reduced-precision types such as FP16, BF16, and emerging FP8 variants (E4M3 and E5M2) have gained widespread adoption in both training and inference workloads. This diversity of numerical representations enables significant improvements in performance and energy efficiency, but also introduces new considerations for model stability and hardware support.

Table 2 – **Parameters of floating point types encoding**

Format	Sign	Exponent Bits	Mantissa Bits	Exponent Bias
float32 (FP32)	1	8	23	127
float16 (FP16)	1	5	10	15
bfloat16 (BF16)	1	8	7	127
fp8_e4m3 (143)	1	4	3	7
fp8-e5m2 (152)	1	5	2	15

A representative example is provided in [11], which introduces an adaptive quantization methodology for FP8-based deep neural network training.

Modern FPGAs contain fundamental building blocks for implementing combinational and sequential logic — Look-Up Tables (LUTs). Depending on the vendor, they are organized within slices or adaptive logic modules. Xilinx FPGAs typically use slices containing four 6-input LUTs, along with associated flip-flops and carry-chain logic, while Intel FPGAs employ Adaptive Logic Modules with fracturable LUTs supporting up to 8 inputs.

LUTs are highly flexible, allowing complex logic functions to be implemented efficiently, or partitioned into smaller functions to maximize resource utilization.

For our floating-point type conversion blocks, LUTs provide an ideal substrate for implementing the combinational logic required for exponent and mantissa manipulation, as well as offset adjustments. By leveraging the fracturable or multi-LUT capabilities, these blocks can achieve high throughput and area efficiency, minimizing the consumption of registers and logic resources while supporting multiple precision formats such as FP32, FP16, BF16, and FP8.

All modules share a similar structural organization. To convert a low-precision floating-point number into the IEEE-754 single-precision FP32 format, the sign, exponent, and mantissa fields are first extracted from the input representation. The bit widths of these fields are then adjusted to match the FP32 format, including the appropriate exponent bias transformation and mantissa expansion.

Special numerical cases, such as positive and negative infinity and Not-a-Number (NaN), are explicitly detected and handled to ensure full compliance with the IEEE-754 standard.

The overall architecture of the proposed modules is illustrated in Fig. 1.

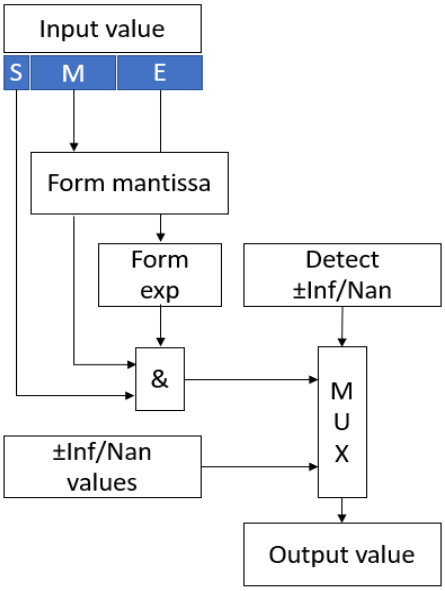

Fig. 1. Conversion block diagram

The diagram presents the structural organization and data flow between the main functional blocks, highlighting the bits extraction, classification, conversion, and value form stages implemented in the design.

Implementation and Results Analysis

In this work, fully synthesizable VHDL modules supporting the FP16, FP8E5M2, FP8E4M3, and BF16 floating-point formats were designed and implemented.

All modules comply with standard hardware design flows, enabling straightforward deployment on both FPGA- and ASIC-based platforms. Functional correctness was verified at the register-transfer level, after which the designs were synthesized using a commercial synthesis tool targeting an Intel Cyclone V FPGA.

The resulting hardware characteristics, including logic utilization and maximum achievable operating frequency, are summarized in Table 3.

These results provide a quantitative comparison of the hardware cost associated with each supported floating-point format and highlight the efficiency of the proposed cast-only conversion architecture.

Table 3 – **Synthesis results for Intel Cyclone V FPGA**

Module	Logic ALMs	Maximum frequency
fp16_to_fp32_bits	42	254
fp8e5m2_to_fp32	14	494
fp8e4m3fn_to_fp32	14	507
bf16_to_fp32_bits	5	497

Overall, the proposed designs demonstrate exceptionally low hardware resource utilization while maintaining high operating frequencies. These characteristics make the modules well suited for integration into modern System-on-Chip (SoC) platforms, including resource-constrained embedded processors and neural network accelerators, where area efficiency and performance are critical.

Conclusions

The proposed conversion modules can be integrated into various parts of an embedded processing pipeline, including the data access stage, ALU pipeline, or as standalone memory-mapped units.

Such flexibility enables efficient adaptation to different system architectures and applications. By offloading precision conversion tasks to dedicated hardware, these modules reduce the overall computational burden, minimize resource duplication, and lower hardware costs.

At the same time, they provide a convenient and scalable mechanism for utilizing low-precision arithmetic, thereby enhancing perfor-mance and energy efficiency in embedded systems without compromising design simplicity.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

REFERENCES

1. Zoni, D., & Galimberti, A. (2022). *Cost-effective fixed-point hardware support for RISC-V embedded systems*. J. Syst. Archit., 126, 102476. https://doi.org/10.1016/j.sysarc.2022.102476.
2. Micikevicius, P., Stosic, D., Burgess, N., Cornea, M., Dubey, P., Grisenthwaite, R., Ha, S., Heinecke, A., Judd, P., Kamalu, J., Mellempudi, N., Oberman, S. F., Shoeybi, M., Siu, M., & Wu, H. (2022). *FP8 formats for deep learning*. arXiv:2209.05433. Machine Learning (cs.LG). https://doi.org/10.48550/arXiv.2209.05433.
3. van Baalen, M., Kuzmin, A., Nair, S. S., Ren, Y., Mahurin, E., Patel, C., Subramanian, S., Lee, S., Nagel, M., Soriaga, J., & Blankevoort, T. (2023). *FP8 versus INT8 for efficient deep learning inference*. arXiv:2303.17951. Machine Learning (cs.LG) https://doi.org/10.48550/arXiv.2303.17951.
4. Tedja, H. A., & Onno W. Purbo. (2024). *Performance and Efficiency Comparison of U-Net and Ghost U-Net in Road Crack Segmentation with Floating Point and Quantization Optimization*. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 8(6), 779-787. https://doi.org/10.29207/resti.v8i6.6089.
5. Chen, J., Hao, H., Wang, S., Li, L., Zhao, X., Yu, F., Wang, J., Xu, G., Sun, Z., & Jiang, K. (2024). *A multiple precision floating-point arithmetic unit based on the RISC-V instruction set*. In Proceedings of the 2024 4th International Conference on

Electronic Information Engineering and Computer (EIECT) (pp. 573–578). IEEE. https://doi.org/10.1109/EIECT64462.2024.10867213.
6. Mach, S., Schuiki, F., Zaruba, F., & Benini, L. (2020). *FPnew: An open-source multi-format floating-point unit architecture for energy-proportional transprecision computing*. arXiv:2007.01530. Hardware Architecture (cs.AR). https://doi.org/10.48550/arXiv.2007.01530.
7. Brand, M., Hannig, F., Keszocze, O., & Teich, J. (2022). *Precision- and Accuracy-Reconfigurable Processor Architectures — An Overview*. IEEE Transactions on Circuits and Systems II: Express Briefs, 69, 2661-2666. https://doi.org/10.1109/TCSII.2022.3173753.
8. Kunešová, M., Zajíc, Z., Šmídl, L. & Karafiát M. (2024) *Comparison of wav2vec 2.0 models on three speech processing tasks*. International Journal of Speech Technology. 27, 847–859. https://doi.org/10.1007/s10772-024-10140-6.
9. Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., & El Sayed, W. (2024). *Mixtral of Experts*. arXiv:2401.04088. Machine Learning (cs.LG). https://doi.org/10.48550/arXiv.2401.04088.
10. Peng, Z., Budhkar, A., Tuil, I., Levy, J., Sobhani, P., Cohen, R., & Nassour, J. (2021). *Shrinking Bigfoot: Reducing wav2vec 2.0 footprint*. arXiv:2103.15760. Computation and Language (cs.CL). https://doi.org/10.48550/arXiv.2103.15760.
11. Hassani Sadi, M., Sudarshan, C. & Wehn, N. (2024) *Novel adaptive quantization methodology for 8-bit floating-point DNN training*. Design Automation for Embedded Systems, 28, 91–110. https://doi.org/10.1007/s10617-024-09282-2.

Received (Надійшла) 27.01.2026
Accepted for publication (Прийнята до друку) 29.04.2026
Publication date (Дата публікації) 22.05.2026

ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

Сальніков Дмитро Валентинович – кандидат технічних наук, старший викладач кафедри автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Dmytro Salnikov – Candidate of Technical Sciences, Senior Lecturer at the Department of automation and control in technical systems, head of the department, National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine
e-mail: dmytro.salnikov@khpi.edu.ua; ORCID Author ID: https://orcid.org/0009-0007-6201-5370;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57225126629.

Васильченков Олег Георгійович – кандидат технічних наук, доцент кафедри автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Oleg Vasylchenkov – Candidate of Technical Sciences, Associate Professor at the Department of Automation and Control in Technical Systems, National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine
e-mail: oleh.vasylchenkov@khpi.edu.ua; ORCID Author ID: https://orcid.org/0000-0002-0969-2248);
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57225129501.

Компактні апаратні модулі для перетворення форматів FP16/FP8/FP32 у вбудованих системах

Д. В. Сальніков, О. Г. Васильченков

Анотація. Стрімке поширення нейронних мереж у вбудованих та периферійних обчислювальних системах зумовило зростання попиту на ефективні апаратні рішення, що здатні підтримувати арифметику зі змінною точністю. В таких сферах, як автономні транспортні засоби, інтелектуальні сенсори та промислова автоматизація, вимагається висока обчислювальна продуктивність, мала затримка та суворі обмеження електроживлення. Арифметика з плаваючою комою, визначена стандартом IEEE 754, залишається домінуючим методом числового представлення у подібних системах завдяки своїй універсальності та широкому динамічному діапазону. Водночас розгортання сучасних моделей глибокого навчання на платформах з обмеженими ресурсами створює значні труднощі, пов'язані з досягненням балансу між точністю, пропускною здатністю та апаратними витратами. Для подолання цих обмежень дедалі більшої популярності набувають формати зменшеної точності, такі як FP16, BF16 та FP8 (E4M3, E5M2), які використовуються як під час інференсу, так і під час навчання, забезпечуючи зниження пропускної здатності пам'яті та підвищення енергоефективності одночасно з мінімальною втратою точності. Попри їх зростаюче поширення, багато мікроконтролерів і FPGA не мають нативної апаратної підтримки таких форматів, що зумовлює необхідність розробки компактних і придатних до реконфігурації модулів перетворення для забезпечення сумісності з традиційними обчислювальними блоками FP32. У цій роботі показані результати проєктування, реалізації та оцінки апаратних модулів для перетворення між форматами FP8, FP16, BF16 та стандартним форматом IEEE-754 (FP32). Запропонована архітектура використовує логічні можливості FPGA для виконання операцій перетворення в полях експоненти та мантиси, корекції зсуву та класифікації спеціальних числових випадків, таких як нескінченність і NaN, що забезпечує повну відповідність до стандарту IEEE-754. Синтез перетворювачів виконано з використанням засобів розробки та реалізації на FPGA Intel Cyclone V. Експериментальні результати демонструють надзвичайно низьке використання апаратних ресурсів і високу робочу частоту: перетворювачі FP8E4M3 та FP8E5M2 потребують лише 14 адаптивних логічних модулів (ALM) кожен, досягаючи частот понад 500 МГц. Отримані результати підтверджують придатність запропонованих модулів для застосування в системах змішаної точності та вбудованих прискорювачах нейронних мереж, забезпечуючи ефективну апаратну основу для енергоефективних і високопродуктивних моделей ШІ на платформах з обмеженими ресурсами.

Ключові слова: формати з плаваючою комою, числове представлення зі зменшеною точністю, вбудовані системи, периферійні обчислення, FPGA, VHDL, прискорювачі вбудованих нейронних мереж, компактна архітектура.