OPEN Compute Project

OCP 8-bit Floating Point Specification (OFP8)

Revision 1.0

Date Submitted: May 26, 2023
Date Approved: June 20, 2023

Author: Paulius Micikevicius and Stuart Oberman, NVIDIA
Author: Pradeep Dubey, Marius Cornea and Andres Rodriguez, Intel
Author: Ian Bratt and Richard Grisenthwaite, Arm
Author: Norm Jouppi, Chiachen Chou and Amber Huffman, Google
Author: Michael Schulte and Ralph Wittig, AMD
Author: Dharmesh Jani and Summer Deng, Meta

Table of Contents

1. License 3

1.2 Acknowledgements 4

2. Compliance with OCP Tenets 5

2.1. Openness 5

2.2. Efficiency 5

2.3. Impact 6

2.4. Scale 7

2.5. Sustainability 7

3. Version Table 9

4. Scope 10

4.1.Definitions 10

4.2.Abbreviations and acronyms 10

5. Overview 12

5.1. OFP8 Binary Interchange Formats 12

5.2 Conversion Behavior 14

5.2.1. Conversion Arithmetic 14

6. Software Support (recommended) 15

7. References (recommended) 15

1. License

Contributions to this Specification are made under the terms and conditions set forth in Open Web Foundation Modified Contributor License Agreement (“OWF CLA 1.0”) (“Contribution License”) by:

AMD, Arm, Google, Intel, Meta, and NVIDIA

Usage of this Specification is governed by the terms and conditions set forth in **Open Web Foundation Modified Final Specification Agreement (“OWFa 1.0.2”) (“Specification License”).**

You can review the applicable OWFa1.0.2 Specification License(s) referenced above by the contributors to this Specification on the OCP website at http://www.opencompute.org/participate/legal-documents/. For actual executed copies of either agreement, please contact OCP directly.

Notes:

1) The above license does not apply to the Appendix or Appendices. The information in the Appendix or Appendices is for reference only and non-normative in nature.

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING

NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2 Acknowledgements

The Contributors of this Specification would like to acknowledge the following companies for their feedback:

AWS

2. Compliance with OCP Tenets

2.1. Openness

The specification for 8-bit floating point (OFP8) complies with the tenet of Openness in the following ways.

The specification recognizes the heterogeneity of data centers where specialized silicon and disaggregated solutions are becoming more common. This also leads to increasing importance of software infrastructure in managing this complexity. By providing a standardized interchange format for FP8 encodings, it enables seamless interoperability between different hardware architectures and software stacks, which can reduce the complexity of managing a diverse set of formats.

The contribution provides two binary interchange formats for floating point encodings that can be implemented in hardware, software, or a combination of both. The openness is demonstrated through collaboration, willingness to share, and, seek feedback to design and specification contributions under consideration. The OFP8 format is being adopted in software infrastructure and it is highly desirable that any standard preserves that business investment.

To address the tenet of Openness, the standardization approach proposed is to define a version 1 that embodies the format being used today. As learnings mature and if enough business value is demonstrated, a version 2 can be defined that advances the state of FP8 while striving for backward compatibility with version 1 to preserve software investment.

The measure of openness is the ability of third parties to build, modify, or personalize the device or platform from the contribution. The OFP8 specification contributes interchange formats only, including conversion operations and behaviors, to enable diversity of chips. Openness can be achieved by removing barriers that prevent an open platform. In this case, the OFP8 specification aims to remove barriers by providing clear interchange formats and conversion operations given the diversity of chips in the data center.

2.2. Efficiency

The OCP specification for the binary interchange formats for 8-bit floating point (OFP8) encodings aligns with the Open Compute Project's tenet of efficiency in several ways.

Firstly, it contributes to the reduction of infrastructure costs by providing a standardized format for floating point encodings that can be implemented in hardware, software, or a combination of both. This reduces the need for customized solutions, which can be costly to develop and maintain.

Secondly, the specification promotes energy efficiency by reducing the power consumption of the hardware that implements the OFP8 format due to reduced computational load. This also improves platform performance by providing a performance-to-watt ratio that is optimized for the FP8 format. This enables more efficient use of computational resources, which is particularly important as training models become increasingly large and costly to train.

Moreover, the specification also contributes to the reduction of code weight and latencies. By defining a standardized interchange format for floating point encodings, it reduces the need for complex conversion operations that can increase the size of the codebase and introduce latency.

Finally, the specification also recognizes the need to preserve existing software investments while advancing the state of FP8. This aligns with the Open Compute Project's tenet of efficiency by avoiding unnecessary duplication of effort and reducing the costs associated with upgrading legacy software.

2.3. Impact

The OFP8 specification proposed by our team aligns with the Impact tenet of the Open Compute Project by introducing a new technology that has the potential to transform the industry. With the increasing size of training models and heterogeneity in the data center, there is a pressing need for clear interchange formats and conversion operations for floating point encodings, particularly as chips become more specialized and disaggregated solutions are adopted.

OFP8 is already being adopted in software infrastructure, and our proposed standardization approach aims to preserve this investment while advancing the state of FP8 in the future, demonstrating business value before defining additional formats. By defining interchange formats and behaviors, we aim to enable OFP8 through a global silicon supply chain, achieving the transformative impact that the Open Compute Project seeks.

Further, the potential impact of the OFP8 specification extends beyond just the data center ecosystem. As a new technology, it has the potential to offer time-to-market advantages and enable new applications that were previously not possible or feasible. Therefore, by introducing and advancing the OFP8 specification, we aim to have a positive impact on the industry, enabling new opportunities and possibilities for many customers in many regions of the world.

2.4. Scale

This specification for binary interchange formats of 8-bit floating-point encodings adheres to the Scale tenet published by Open Compute Project, which requires that contributions must have sufficient reach to cater to Fortune 100 customers and large hyperscale customers.

With the exponential growth in AI systems and servers globally in the next five years, it's crucial to have binary interchange formats that support the increasing diversity of chips in the industry at a vast scale, resulting in a transformative impact. Our contribution meets this scale tenet by defining interchange formats, conversion operations, and behaviors that enable adoption of new technology through a global supply channel, accelerates time-to-market advantage of technology, and supports ongoing innovation as well as legacy infrastructure.

The OFP8 format is being adopted today in software infrastructure, providing a foundation for this innovation. Our proposed standardization approach accommodates the legacy infrastructure both at the hardware and software levels, and only additional formats defined if there is significant business value demonstrated. This scaling approach ensures backward compatibility with previous versions, preserving software investment and facilitating cross-compatibility. As data centers continue to grow in size and complexity, the proposed standardization approach of binary interchange formats for floating point encodings will significantly reduce the cost and complexity of software development, and enable critical and transformative changes in the industry.

The transformative impact of standard binary interchange formats for FP8 encodings is significant. With the projected growth in servers and AI systems globally, the proposed standardized OFP8 format provides a much-needed framework supporting innovation and driving adoption at scale in the data center. This will undoubtedly propel the industry forward and enable developers to build solutions to meet global demand while ensuring their long-term success.

2.5. Sustainability

This specification for binary interchange formats for 8-bit floating point (OFP8) encodings complies with the Open Compute Project's Sustainability tenet in several ways.

First, the specification minimizes the compute intensity by enabling more efficient communication and computation between the increasing array of accelerators that are now common in data centers. By ensuring that there are clear interchange formats and conversion operations that can be implemented across a wide variety of accelerators, the OFP8 specification can help reduce the need for additional hardware and thereby limit the environmental impact of data centers.

Secondly, by providing a robust and flexible specification for 8-bit floating point encodings, the specification facilitates interoperability between hardware devices that enables training on one type of infrastructure and deployment on another type of infrastructure. This combination of flexibility and interoperability allows the industry to keep the entire data center better utilized and optimize each step (training, inference) to run on the most efficient infrastructure.

Overall, the OFP8 specification for binary interchange formats offers a clear path towards achieving sustainable and responsible data center operations, while also driving innovation and progress in the broader world of technology.

3. Version Table

Date	Version #	Author	Description
Jun 20, 2023	1.0	Paulius Micikevicius	Defines FP8 interchange format.

4. Scope

This document specifies binary interchange formats for 8-bit floating point (FP8) encodings, as well as conversion to these formats from wider formats (IEEE single and half precision, also known as IEEE binary32 and binary16 [1], as well as bfloat16 [2] formats).

Not in scope are:

- methods for conversion between the specified OFP8 formats and alternative 8-bit floating point formats
- methods for conversion between the specified OFP8 formats and integer formats
- methods for performing arithmetic operations on the specified OFP8 formats
- details of applications' use of specified OFP8 formats in general
- scaling factor selection, representation, or maintenance

4.1.Definitions

For definitions or descriptions of the following terms see the IEEE-754 Floating-Point Specification [1]: biased exponent, binary16, binary32, destination, exception, exponent, floating-point number, format, implementation-defined, interchange format, integer format, NaN, normal number, status flag, subnormal number, trailing significant field, and wider format.

Definitions for other terms are given below.

binade: a set of numbers in a binary floating-point format that all have the same exponent.

bfloat16: a binary format with 1 sign bit, 8 exponent bits, 7 mantissa bits, and an exponent bias of 127. [2]

exponent: the part of a finite floating-point number that signifies the integer power of two by which the significand is multiplied to determine the value of the number. This is consistent with IEEE definition of exponent, but we consider only powers of 2 (i.e. radix 2).

mantissa: the part of a finite floating-point number that contains the significant bits (a.k.a. trailing significand).

sign bit: the bit in a floating-point number that indicates if the number is positive or negative.

4.2.Abbreviations and acronyms

Inf: Infinity

OFP8: An 8-bit floating-point number with the formats and encodings specified in this document.

E4M3: An OFP8 format with 1 sign bit, 4 biased exponent bits, 3 mantissa bits, and an exponent bias of 15. See below for further details.

E5M2: An OFP8 format with 1 sign bit, 5 biased exponent bits, 2 mantissa bits, and an exponent bias of 7. See below for further details.

emax: the maximum exponent in a specified format

emin: the minimum exponent in a specified format

max_E4M3: The maximum magnitude of a number in the E4M3 format. It has the value 448.

max_E5M2: The maximum magnitude of a number in the E5M2 format. It has the value 57,344.

5. Overview

This specification contributes two binary interchange formats for floating point encodings. These formats can be implemented in hardware, software, or a combination of the two.

5.1. OFP8 Binary Interchange Formats

OFP8 representation consists of sign, exponent, and mantissa fields. In this specification we use the term *mantissa* to refer to the trailing significand bits. Two encodings are defined - E4M3 and E5M2, where the name explicitly states the number of bits in the exponent (E) and mantissa (M) fields. Encodings are illustrated in Figure 1 and consist of:

- 1 sign bit: the most significant bit
- e-bit biased exponent: 4 bits for E4M3, 5 bits for E5M2
- *m* mantissa (trailing significand) bits: 3 bits for E4M3, 2 bits for E5M2

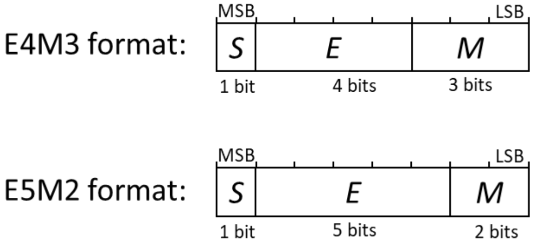

Figure 1. Binary interchange OFP8 formats

The value, *v*, of a normal OFP8 number is

$$v = (-1)^S \times 2^{E-\text{bias}} \times (1 + 2^{-m} \times M)$$

The value, *v*, of a subnormal OFP8 number (subnormals have $E = 0$ and $M > 0$) is

$$v = (-1)^S \times 2^{1-\text{bias}} \times (0 + 2^{-m} \times M)$$

Exponent parameters and min/max values for both OFP8 formats are specified in Table 1.

The E5M2 format represents infinities and NaNs. Interpretation of the three mantissa values for NaNs is not defined. The E4M3 format does not represent infinities and uses only two bit patterns for NaN (a single mantissa-exponent bit pattern but allowing both values of the sign bit) in order to increase *emax* to 8 and thus to increase the dynamic range by one binade. Various values for OFP8 formats are detailed in Table 2.

Table 1: OFP8 exponent parameters

	E4M3	E5M2
Exponent bias	7	15
emax (unbiased)	8	15
emin (unbiased)	-6	-14

Table 2: OFP8 value encoding details

	E4M3	E5M2
Infinities	N/A	$S.11111.00_2$
NaN	$S.1111.111_2$	$S.11111.\{01, 10, 11\}_2$
Zeros	$S.0000.000_2$	$S.00000.00_2$
Max normal number	$S.1111.110_2 = \pm 448$	$S.11110.11_2 = \pm 57{,}344$
Min normal number	$S.0001.000_2 = \pm 2^{-6}$	$S.00001.00_2 = \pm 2^{-14}$
Max subnormal number	$S.0000.111_2 = \pm 0.875 * 2^{-6}$	$S.00000.11_2 = \pm 0.75 * 2^{-14}$
Min subnormal number	$S.0000.001_2 = \pm 2^{-9}$	$S.00000.01_2 = \pm 2^{-16}$
Dynamic range	18 binades	32 binades

5.2 Conversion Behavior

5.2.1. Conversion Arithmetic

This specification defines conversion from wider floating point types (IEEE binary32, binary16 [1], colloquially known as single- and half-precision; bfloat16 [2]) to OFP8 formats. Conversion can be implemented in software, hardware, or combination of the two. The specification does not require any exception handling during these conversions, nor does it require using status flags for exceptions.

There are two aspects to consider when converting the wider formats to OFP8:

- Saturation mode: this specification requires implementation of saturating and non-saturating modes.
- Rounding mode: this specification requires implementation of round to nearest even mode. Implementations may implement additional rounding modes that are not required by this specification.

Conversion from a NaN produces an implementation-defined OFP8 NaN. Conversion from a wider format infinity depends on the saturation mode as shown in Table 3. Conversion from all other values first applies rounding to reduce the mantissa bit count to that of the destination OFP8 format. After that, if the rounded magnitude:

- is above the maximum destination magnitude:
  - if using saturating mode: generate the max normal OFP8 magnitude
  - if using non-saturating mode:
    - E4M3 destination: generate a NaN
    - E5M2 destination: generate an infinity
- is in the representable range of OFP8 (including subnormal numbers): corresponding OFP8 exponent bits are generated.
- is below the minimum subnormal number magnitude: generate a corresponding zero.

The sign bit of the source value is preserved except in the cases where a NaN is produced, where sign bit handling is left up to the implementation. Conversion behavior is summarized in Table 3.

Table 3: Summary of cases for conversion to OFP8 from wider formats

Source value (after rounding)	**Destination value**			
	E5M2		**E4M3**	
	SAT	**NONSAT**	**SAT**	**NONSAT**
NaN	NaN	NaN	NaN	NaN
±Inf	±max_E5M2	±Inf	±max_E4M3	NaN
greater than max OFP8 magnitude	±max_E5M2	±Inf	±max_E4M3	NaN
in OFP8 range	Rounded value	Rounded value	Rounded value	Rounded value
smaller than min OFP8 subnormal magnitude	±0	±0	±0	±0
±0	±0	±0	±0	±0

6. Software Support (recommended)

Please document any software tools used to validate the hardware design and include test and validation using virtual simulation, design decisions based upon digital models, or proof of manufacturability via 3-D tools.

This is an interchange format that may be implemented in hardware, software, or a combination of both. Please refer to the OFP8 Software and Hardware References document located on the OCP FP8 GitHub for more information about software support.

7. References (recommended)

[1] "IEEE Standard for Floating-Point Arithmetic," in *IEEE Std 754-2019 (Revision of IEEE 754-2008)* , vol., no., pp.1-84, 22 July 2019, doi: 10.1109/IEEESTD.2019.8766229.

[2] Google, “The Bfloat16 Numerical Format”, https://cloud.google.com/tpu/docs/bfloat16.