IEEE Working Group P3109 Interim Report on Binary Floating-point Formats for Machine Learning

Initial release: 18 September 2023
Second interim release: 29 October 2024

Version 0.9.0: 2024-10-29
(compiled 2024-10-29)

DRAFT DOCUMENT

To cite this report please see the CITATION.cff file
found at https://github.com/P3109/Public.

Contents

1 Contributors **4**

2 Introduction **5**
2.1 Typographical conventions and notation . . . 5
2.2 Scope . . . 5

3 Formats **6**
3.1 Format parameters and value sets . . . 6
3.1.1 Computation of bias . . . 6
3.1.2 Computation of bias, $\mathsf{SE} = 1$ . . . 7
3.1.3 Computation of bias, $\mathsf{SE} = 0$ . . . 7
3.1.4 Observations on P3109 value sets . . . 7
3.2 Encodings and special values . . . 8
3.3 Subnormals . . . 8
3.4 Not a number (NaN) . . . 9
3.5 Zero . . . 9
3.6 Infinities . . . 10
3.7 Extremal values for $\boldsymbol{K = 8}$ . . . 10
3.8 Equivalences . . . 10

4 Operations **11**
4.1 Operation variants and superset specifications . . . 12
4.2 Notation and definitions . . . 12
4.2.1 Mathematical notations . . . 12
4.2.2 The set of extended reals . . . 12
4.3 Projection specifications . . . 13
4.4 Non-requirement for operator side effects . . . 13
4.5 Operator definition template . . . 14
4.6 Functions . . . 15
4.6.1 Decode . . . 15
4.6.2 Project . . . 16
4.6.3 RoundToPrecision . . . 17
4.6.4 Saturate . . . 18
4.6.5 Encode . . . 19
4.7 Conversion operations . . . 20
4.7.1 Specification of IEEE Std 754 formats . . . 20
4.7.2 Conversion from IEEE Std 754 formats to P3109 . . . 21
4.7.3 Conversion from P3109 to IEEE Std 754 . . . 22
4.7.4 Conversion from P3109 to P3109 . . . 23
4.8 Arithmetic operations . . . 24
4.8.1 Unary sign operations . . . 24
4.8.2 Binary sign operations . . . 25
4.8.3 Binary arithmetic operations . . . 26
4.8.4 Unary mathematical operations . . . 27

4.8.5 Scaled addition . . . 28
4.8.6 Scaled multiplication . . . 29
4.9 Mixed IEEE Std 754 and P3109 operations . . . 30
4.9.1 Scaled Fused Multiply Add . . . 31
4.10 Comparisons, predicates, and classification . . . 32
4.10.1 Minimum and Maximum . . . 32
4.10.2 Comparisons . . . 33
4.10.3 Predicates and classification . . . 34
4.10.4 Classifier operation . . . 35
4.10.5 Total order predicate . . . 36
4.10.6 Comparison predicates . . . 37

Appendices **38**

A Rationales **38**
A.1 Use of infinity in computation of attention masks . . . 38
A.2 Eight Bit Formats . . . 38
A.3 Essential Formats . . . 39
A.3.1 binary8p3 . . . 39
A.3.2 binary8p4 . . . 39
A.3.3 binary8p5 . . . 39
A.3.4 binary8p6 . . . 39

B External Formats **40**

C 8-bit Value Tables **40**
C.1 Value Table: P1, $\mathsf{emin} = -62, \mathsf{emax} = 63$ `(nonsymmetric)` . . . 41
C.2 Value Table: P2, $\mathsf{emin} = -31, \mathsf{emax} = 31$ . . . 42
C.3 Value Table: P3, $\mathsf{emin} = -15, \mathsf{emax} = 15$ . . . 43
C.4 Value Table: P4, $\mathsf{emin} = -7, \mathsf{emax} = 7$ . . . 44
C.5 Value Table: P5, $\mathsf{emin} = -3, \mathsf{emax} = 3$ . . . 45
C.6 Value Table: P6, $\mathsf{emin} = -1, \mathsf{emax} = 1$ . . . 46
C.7 Value Table: P7, $\mathsf{emin} = 0, \mathsf{emax} = 0$ `(linear)` . . . 47

1 Contributors

Here are the active members of the Arithmetic Formats for Machine Learning working group.

Kiran Gunnam, *Chair*
Leonard Tsai, *Vice Chair*
Jeffrey Sarnoff, *Secretary*

Tom Thompson, *IEEE Standards Board Liaison*

Editors
Jeffrey Sarnoff, Andrew Fitzgibbon, Amos Omondi, Guy Lemieux

Paul Balanca	Michel Hack	Nathalie Revol
Luca Bertaccini	Simon Knowles	Jason Riedy
Javier Diaz Bruguera	Seokbum Ko	Ali Sazegari
David H. C. Chen	Carlo Luschi	Eric Schwarz
Marco Cococcioni	David Lutz	Oliver Sentieys
Mike Cowlishaw	Tue Ly	Michael Siu
Marius Cornea	Al Martin	Gil Tabak
Debjit Das Sarma	Paulis Micikevicius	Julio Villalba-Moreno
James Davenport	Mantas Mikaitis	Christop Wintersteiger
Jim Demmel	Aaftab Munshi	Kristopher Wong
Ken Dockser	Santosh Nagarakatte	Tom Yeh
Massimiliano Fasi	Badareddine Noune	Chao Yu
Silviu Filip	Stuart Oberman	Aleksandr Zakharchenko
Jeff Gonion	Michael Overton	Hao Zhang
John Gustafson	Valentina Popescu	

Contacting The Working Group

To reach us, email `FP-FOR-ML-STUDY-GROUP@listserv.ieee.org`.

2 Introduction

This document represents ongoing discussions and current matters of consensus from IEEE Working Group P3109, "Standard for Arithmetic Formats for Machine Learning". The Project Authorization Request (PAR) for P3109 defines the scope, need, and stakeholders as follows:

> **Scope of proposed standard**: This standard defines a binary arithmetic and data format for machine learning-optimized domains. It also specifies the default handling of exceptions that occur in this arithmetic. This standard provides a consistent and flexible arithmetic framework optimized for Machine Learning Systems (MLS) in hardware and/or software implementations to minimize the work required to make MLS interoperable with each other, as well as other dependent systems. This standard is aligned with IEEE Std 754-2019 for Floating-Point Arithmetic.
>
> **Need for this Work**: Machine Learning Systems have different arithmetic requirements from most other domains. Precisions tend to be lower, and accuracy is measured in dimensions other than just numerical (e.g. inference accuracy). Furthermore, machine learning systems are often integrated into mission-critical and safety-critical systems. With no standards specifically addressing these needs, Machine Learning Systems are built with inconsistent expectations and assumptions that hinder the compatibility and reuse of machine learning hardware, software, and training data.
>
> **Stakeholders for the Standard**: System developers, vendors, and users of machine learning applications across many industries and interests including but not limited to computation, storage, medical, telecommunications, e-commerce, fleet management, automotive, robotics, and security.

2.1 Typographical conventions and notation

Bold text describes the decisions and specifications of this document.

Text that is not in boldface is background material, typically providing rationale and arguments that represent discussions of the working group leading to a decision and specification.

2.2 Scope

This document specifies compact floating-point interchange formats (binary formats) and associated operations.

Binary formats are parameterized by their width in bits and their precision—the number of bits in the true significand (which is one more than the number of bits in the trailing significand).

The 8-bit formats defined herein shall be referred to as "binary8" formats, and further qualified by precision yielding names "binary8pP" for values $1 \leq P \leq 7$.

For example, "binary8p3" is an 8-bit format with 1 sign bit, 5 exponent bits and 3 bits of precision (of which only two need to be explicitly represented).

This version of the interim report covers interchange formats and scalar operations including rounding with saturation modes and conversion between P3109 and IEEE Std 754. vector operations, block formats, stochastic rounding, and conversion between P3109 and BFloat16 are not discussed here. These topics may be included in a later version.

3 Formats

This section describes P3109 formats, and the set of values that such formats shall represent.

The universe of values in existing floating-point usage is the finite reals, the non-finite values positive and negative infinity (Inf, -Inf), the value negative zero (-0), and not-a-number values (NaN, NaN_1, ...), subscripted by *NaN payload*.

A K-bit binary floating-point format $\mathcal{F}$ comprises: the *value set*: a subset $\mathcal{V}_\mathcal{F}$ of the universe of values; and an *encoding*: a mapping from integers $0 \ldots 2^\mathsf{K} - 1$ to $\mathcal{V}_\mathcal{F}$.

3.1 Format parameters and value sets

The finite floating-point numbers representable with a binary format are determined by two *format-defining* parameters:

- Storage width K, the total size of the format in bits
- Precision P, the number of bits in the significand including the implicit leading bit.

All other parameters, such as the exponent of the largest finite value emax, are derived from the format-defining parameters.

IEEE Std 754-2019 includes the radix B and the minimum exponent emin in a list of format-defining parameters, this document excludes both of them for these two reasons:

- This document covers binary (radix 2) formats only, so B is not a format parameter.
- The quantity emin is determined by P and emax; it cannot be varied independently, so it cannot be a format-defining parameter.

P3109 formats shall be defined by the parameters of storage width K, and precision P, where $1 \leq \mathsf{P} < \mathsf{K}$

In IEEE Std 754, emax was set for all defined formats to be $2^{\mathsf{W}-1} - 1$, where W is the exponent field width in bits. In this document, this convention is formalized: emax is a fixed function of P, written $\mathsf{emax}(\mathsf{K}, \mathsf{P})$, with the formula following IEEE Std 754, noting that $\mathsf{W} = \mathsf{K} - \mathsf{P}$.

P3109 formats shall define $\mathsf{emax}(\mathsf{K}, \mathsf{P})$ to be $2^{\mathsf{K}-\mathsf{P}-1} - 1$

3.1.1 Computation of bias

The choice of emax for a given format then determines the exponent bias for that format. The bias is chosen so that the exponent of the largest finite value is emax.

For IEEE Std 754 formats, the largest finite value corresponds to an exponent field which has all but the zeroth bit set (e.g. 11110 for Binary16), because all of the values with all-bits-one exponents are occupied by non-finite values (Not-a-Numbers or Infinities). This derived parameter of a format is termed *all-special exponent*, and denoted by the symbol SE, where $\mathsf{SE} = 1$ indicates that all-bits-one exponent fields are entirely occupied by special values. For P3109 values, the all-bits-one exponent contains only one special value ($\pm$Inf) and hence $\mathsf{SE} = 0$ unless $\mathsf{P} = 1$. Hence, for P3109, $\mathsf{SE} = (\mathsf{P} = 1)$.

Table 1: Parameters for binary formats

Symbol	Parameter Description	Derived Value	binary8pP, K = 8, P = 7	6	5	4	3	2	1	IEEE754-2019, K = 16	32	64
K	**storage (bits)**	**K**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**16**	**32**	**64**
P	**precision (bits)**	**P**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**11**	**24**	**53**
S	sign (bits)	1	1	1	1	1	1	1	1	1	1	1
W	exponent (bits)	$K - P$	1	2	3	4	5	6	7	5	8	11
T	trailing significand (bits)	$P - 1$	6	5	4	3	2	1	0	10	23	52
SE	all-special exponent	SE	0	0	0	0	0	0	1	1	1	1
emax	maximum exponent	$2^{W-1} - 1$	0	1	3	7	15	31	63	15	127	1023
emin	minimum exponent	SE − emax	0	-1	-3	-7	-15	-31	-62	-14	-126	-1022
bias	exponent bias	1 − emin	1	2	4	8	16	32	63	15	127	1023

Format-defining parameters in bold, derived parameters in normal font.
Adapted from Table 3.5 of IEEE Std 754-2019, and extended to include
the binary8pP formats. Concepts are explained in detail in this section.

3.1.2 Computation of bias, SE $= 1$

With P3109 formats where $P = 1$, in common with IEEE Std 754, the biased exponent of the largest finite value is $2^W - 2$, from which bias should be defined so that

$$(2^W - 2) - \text{bias} = \text{emax}$$

Rearranging, we obtain the following

$$\text{bias} = (2^W - 2) - (2^{W-1} - 1) = 2 \cdot (2^{W-1} - 1) - (2^{W-1} - 1) = 2^{W-1} - 1 = \text{emax}$$

Hence bias $=$ emax.

3.1.3 Computation of bias, SE $= 0$

For the binary8 formats in this document where $P > 1$, only one of the values that has exponents with all-bits-one is non-finite ($\pm$Inf), so the biased exponent of the largest finite value is $2^W - 1$. Hence the bias calculation becomes

$$\text{bias} = (2^W - 1) - (2^{W-1} - 1) = 2^{W-1} - 1 + 1 = \text{emax} + 1$$

As, by formula, emax is odd, the bias term in the $P > 1$ formats is typically even, yielding a more symmetrical range, where emin $= -$emax.

3.1.4 Observations on P3109 value sets

Table 1 shows some properties of the P3109 value sets for $K = 8$.

For $P \geq 1$, the value sets are subsets of the IEEE Std 754 Binary32 value set. For $P \geq 3$, the binary8pP value sets are also subsets of the IEEE Std 754 Binary16 value set.

The precision P is to be strictly less than K, hence formats where $\mathsf{P} = \mathsf{K}$ are not defined in this document. Rationale: strictly following Table 1 would yield $\mathsf{emax} = -\frac{1}{2}$ which means all ordinary values are irrational. Rounding this computation upward yields $\mathsf{emax} = 0$ and $\mathsf{bias} = 1$, where all representations within the format are subnormal, with the consequence that the value sets and encodings for $\mathsf{P} = \mathsf{K} - 1$ and $\mathsf{P} = \mathsf{K}$ are identical except for a scaling factor of 1/2.

The following notes from IEEE Std 754-2019 have not been found to apply to the definitions in this document:

- "For binary formats, the precision [P] should be at least 3, as some numerical properties do not hold for lower precisions."
- "Similarly, emax should be at least 2 to support the operations listed in 9.2. [sin/cos/exp/log/etc]"

3.2 Encodings and special values

Encodings are mappings from integers $0 \ldots 2^{\mathsf{K}} - 1$ to the value set, and are defined in §4.6.5 below.

Values are considered either "special" or "ordinary". Encodings of the special values, shared by all P3109 formats, are shown in Table 2. P3109 formats have only a single NaN, located at the position that -0 would occupy in an analogous IEEE-754 encoding, providing an increased range. The ordinary values consist of the normal and subnormal values and zero.

Table 2: Special value encodings

Special Value	Symbol	Code point	Code point ($K = 8$)
Positive Infinity	Inf	$2^{\mathsf{K}-1} - 1$	`0x7F`
Negative Infinity	-Inf	$2^{\mathsf{K}} - 1$	`0xFF`
Not a Number	NaN	$2^{\mathsf{K}-1}$	`0x80`

These mappings are shared by all P3109 formats with bit-width K.

3.3 Subnormals

P3109 value sets shall include subnormals.

IEEE Std 754 value sets include subnormals. A value with trailing significand field T and exponent field E is interpreted as $1.T \times 2^{E-\mathsf{bias}}$ except when all bits of the exponent bitfield are 0, in which case the value is $0.T \times 2^{\mathsf{emin}}$.

Subnormal numbers extend the dynamic range of floating-point values and induce equal quantization steps close to zero. They can be useful when training models, where it is common to represent near-zero values for gradients. Subnormals can also be useful to represent random values drawn from certain distributions. For example, model weights are initialized to small random values at training. Subnormals are uniformly spaced around zero, and values near zero are more probable values drawn from Gaussian-like distributions. Finally, formats with narrow exponent widths necessarily have a limited range; subnormals extend this range by a power of 2 for every bit in the trailing significand.

3.4 Not a number (NaN)

P3109 value sets shall include exactly one NaN, encoded as 2^{K-1}, which shall not signal.

Many other floating-point formats define several NaN values which are returned from operations with results outside the set of values, e.g., $\mathsf{DIV}(0,0)$, or $\mathsf{ADD}(+\mathsf{Inf},-\mathsf{Inf})$. Multiple NaN encodings are used in other formats to allow different exceptional conditions to be distinguished.

In the context of machine learning systems, uses of NaN include:

- Debugging of code running on accelerator hardware. In A.I. accelerators, exceptions may be difficult or expensive to convey back to user code, so it is common practice to allow NaN values to propagate through calculations to indicate that an error has occurred.
- Use as a sentinel value. In some datasets, for example, where individual element values may be missing or out of range, a sentinel may be used to record the position of these values. In many cases, this will require less memory than storing such information out-of-band, such as in a coordinate-list (COO) format array. In some cases, $\pm$Inf can be used as a missing value, but given the restricted range of P3109 formats, it is likely that infinity shall be used as a separate indicator of rounding from values outside of the finite range.
- The use of multiple NaN payloads is known in statistical code (e.g. the R system has NaN and N/A), but it is not widely used. In the context of P3109, supporting multiple NaN swould reduce the already limited encoding space (e.g., occupying all code points where the exponent field is all ones, thereby reducing dynamic range) and would likely add additional hardware complexity.

3.5 Zero

P3109 formats shall have exactly one zero, encoded as the integer 0. This zero value is non-negative.

The inclusion of negative zero (-0) would incur the cost of an additional code point. Given the decision to encode only a single NaN, placing that NaN at the negative zero code point enables the strictly positive and strictly negative number ranges to be symmetric.

A key rationale for including -0 in IEEE Std 754 was the consistent implementation of branch cuts in the atan2 function [1, 2]. Although the atan function is common in deep learning, it is generally used as an activation function, rather than a trigonometric operation, and the atan2 function is rare, if not unknown, in deep learning applications. Hence, it is not expected that this standard shall define either atan or atan2.

A secondary reason for providing -0 is the hardware simplification offered by its presence in the implementation of sign/magnitude arithmetic. However, the existence of in-market implementations is evidence that the small hardware simplification has not been sufficient to balance the loss of one code point.

It might be considered that the use of integer comparisons in sorting would argue against placing NaN at the negative zero code point. For example, the JAX machine learning framework is known to sort using integer comparison [3]. However, such sorting still requires $O(n)$ prepossessing and postprocessing steps to enable the use of twos-complement integer comparison, and already has special treatment of NaN and -0, so eliminating -0 and placing NaN in the -0 position imposes negligible additional burden. Although sorting using comparison operations is undefined in the presence of NaN s, existing practice is to sort NaN susing totalOrder.

3.6 Infinities

P3109 formats shall include positive and negative infinities, encoded as $2^{K-1} - 1$ and $2^K - 1$, respectively.

This decision causes a reduction in dynamic range (252 values rather than 254 for binary8, for example), while offering improved numerical robustness in important machine learning use cases.

Examples of such usage are:

- Mask values, for example, in Transformer models in machine learning. See §A.1 for more detailed discussion on this usage.
- Representation of overflow, for example, to adjust dynamic loss scaling factors [4]. Existing implementations offer several behaviors on overflow: overflow to infinity, saturation to MaxFloat, and overflow to NaN. The existence of a code point for infinity allows any of these options to be implemented in a given instantiation, while removing the code point removes the possibility of implementing the first.

3.7 Extremal values for $K = 8$

Table 3: Extremal values

Format	minSubnormal	maxSubnormal	minNormal	maxNormal
binary8p1	N/A	N/A	1×2^{-62}	1×2^{63}
binary8p2	1×2^{-32}	1×2^{-32}	1×2^{-31}	1×2^{31}
binary8p3	1×2^{-17}	$3/2 \times 2^{-16}$	1×2^{-15}	$3/2 \times 2^{15}$
binary8p4	1×2^{-10}	$7/4 \times 2^{-8}$	1×2^{-7}	$7/4 \times 2^{7}$
binary8p5	1×2^{-7}	$15/8 \times 2^{-4}$	1×2^{-3}	$15/8 \times 2^{3}$
binary8p6	1×2^{-6}	$31/16 \times 2^{-2}$	1×2^{-1}	$31/16 \times 2^{1}$
binary8p7	1×2^{-6}	$63/32 \times 2^{-1}$	1×2^{0}	$63/32 \times 2^{0}$

It is practical to list extremal finite values defined by the binary8 formats. Following IEEE Std 754-2019 naming patterns, we adopt: $\mathsf{maxNormal}(\tau), \mathsf{minNormal}(\tau), \mathsf{minSubnormal}(\tau)$ where τ is a binary8 format. For example: $\mathsf{maxNormal}(\mathsf{binary8p4}) = 7/4 \times 2^7$ and $\mathsf{minNormal}(\mathsf{binary8p5}) = 1 \times 2^{-3}$.

Table 3 shows the extremal values for $1 \leq \mathsf{P} \leq 7$. For reference, section C provides complete tables of 8-bit values.

3.8 Equivalences

The $P = K - 1$ formats are equivalent to a signed-magnitude fixed-point integer, except there are code points reserved for NaN and $\pm$Inf.

The $P = 1$ formats are "purely exponential" formats, where all values are of the form 2^n, excepting 0, NaN, $\pm$Inf.

4 Operations

This section defines the *behavior* (with no constraint as to the implementation) of operations which P3109 systems may provide. Such operations include:

- conversion between P3109 formats and between P3109 formats and IEEE-754 formats.
- comparison, classification, and informational operations
- addition, multiplication and fused multiply add

In the definitions of the above operations, certain mathematical *functions* are also defined. We emphasize that the definitions in this document are specifications of behavior, not implementation. All arithmetic represented as operating on extended real values is to be interpreted in the usual mathematical sense.

An implementation of any of the operations defined herein is conforming if it computes the same output as the defined operation, for all possible inputs. This may be attested by any proof method, including direct computation.

As defined above, a K-bit P3109 floating-point value (referred to in general as a *P3109 value*) is encoded by an integer in the range 0 to $2^{\mathsf{K}} - 1$. Operation specifications operate on such values using integer arithmetic (e.g., divide and modulo) operations. Implementations may perform these operations in any equivalent manner, for example bit shifting and masking.

For arithmetic operations, an implementation may provide an α-*approximate* implementation. Such an implementation will compute values whose maximum difference from the defined outputs, over all inputs producing finite outputs, does not exceed α units in the last place. This may be attested by any proof method, including direct computation.

4.1 Operation variants and superset specifications

Each operation definition in this document is *parameterized*. Example parameters might include input and output formats, rounding modes and saturation modes. In addition operand types may be specified as if to infinite precision (e.g. s in the template definition below is defined as an integer). The set of variants so defined for a given operation is called a *superset specification*.

No implementation is required to provide all variants of a given operation, but rather should clearly define the implemented variants.

It is required that the use of an operation name defined in this document, in the context of declaration of P3109 compliance, specify precisely the variants implemented, and that the name be used only for variants which exactly follow these specifications.

For example, an implementation might declare as follows. (See later sections for definitions of terms in this example.)

> "This implementation conforms to P3109, providing the following operation variants:
>
> ConvertToP3109, ϕ in {Binary16, Binary32}, f in {binary8p3, binary8p4, binary8p5}
>
> AddScaled, with $s_x = 0$, $s_y \in \{-128 \dots 127\}$, $\{f_x, f_y, f_z\} \subset$ {binary8p3, binary8p4, binary8p5}
>
> MultiplyScaled, with $s \in \{-32 \dots 32\}$
>
> In all cases, provided projection modes π obey $\mathsf{Rnd}_\pi \in \{\mathsf{NearestTiesToEven},\ \mathsf{NearestTiesToAway}\}$ and $\mathsf{Sat}_\pi \in \{\mathsf{SatFinite}, \mathsf{OvfInf}\}$"

This example is truncated—realistic compliance declarations might run to many pages. Such declarations may be compressed by defining common *profiles*, outside of the scope of this current document.

4.2 Notation and definitions

4.2.1 Mathematical notations

We denote by $\mathbb{1}[b]$ the value 1 if $b = \mathsf{True}$, or 0 if $b = \mathsf{False}$.

$\mathsf{IsEven}(I)$ is true if integer I is even, false otherwise.

Integer division is denoted by $x \div y$.

Comments are introduced with an em-dash, and are right-justified —An example comment

4.2.2 The set of extended reals

The set of *extended reals* is the set $\mathbb{R} \cup \{-\infty, \infty\}$ of real numbers augmented with positive and negative infinity, also known as the "affinely extended real numbers". In common with existing mathematical treatments, there is no negative zero in this set. In the extended reals, certain operations, such as $\infty - \infty$, are *undefined*. The definitions in this specification are constructed such that no operation on extended real quantities produces an undefined result.

Operation specifications will, in general, convert floating-point operands to extended real values in order to define their behaviors.

4.3 Projection specifications

Operations on P3109 values are defined via conversion to extended real values, on which the mathematical operation is performed, before conversion back to the appropriate P3109 range. In general, operation results will not be exact P3109 values, and hence will be *projected* into the P3109 range via rounding and overflow handling. A *projection specification* is a pair (rounding mode, saturation mode). For a given projection specification π, these are written $\mathsf{Rnd}_\pi, \mathsf{Sat}_\pi$.

The defined rounding modes are as follows. The precise specifications of these modes are in the function RoundToPrecision (§4.6.2):

NearestTiesToEven	Round to nearest, ties to even
NearestTiesToAway	Round to nearest, ties away from zero
TowardPositive	Round toward positive
TowardNegative	Round toward negative
TowardZero	Round toward zero

Values are first rounded to the target precision, with exponent unbounded above. Those which are then outside the maximum value in the target format are then treated according to the saturation mode.

The defined saturation modes are as follows. The precise specifications of these modes are in the function Saturate (§4.6.4):

SatMax	All return values are clamped to the representable finite range.
SatFinite	Finite out-of-range values are clamped to the representable finite range, infinities are preserved.
OvfInf	Out-of-range values are replaced with extreme value, positive or negative infinity as indicated by the rounding mode.

4.4 Non-requirement for operator side effects

The operator definitions herein describe no side effects, such as the setting of flags, or the triggering of interrupts, and hence do not return values other than the defined return value. Typically NaN is returned when input values are out of range (e.g. $\log(-1.0)$). When saturation is specified, there is no direct mechanism to distinguish overflowed values from values which round to the format's maximum value. An implementation which has side effects will still conform to this specification providing its return value on all inputs matches the definitions herein.

4.5 Operator definition template

Operations are defined according to the following template:

Signature

$\mathsf{Operator}_f(x, s) \rightarrow Z$ —Naming the operator, its parameters, operands and result.

Parameters —Parameters specify a family of related operations

f : format, precision P_f

Operands

x : P3109 value, in format f
s : Integer scale

Output

Z : extended real value or NaN —Result value and type

Behavior —An ordered sequence of pattern-matching declarations.

$\mathsf{Operator}(\mathsf{NaN}, 0) \rightarrow \mathsf{NaN}$ —Exact pattern: only the provided operands match.
$\mathsf{Operator}(x \in \{\text{-}\mathsf{Inf}, \mathsf{Inf}\}, 0) \rightarrow x$ —Match for all x values in the given set.
$\mathsf{Operator}(*, 0) \rightarrow 0$ —The $*$ symbol matches any value.
$\mathsf{Operator}(x, s < 0) \rightarrow -\mathsf{Operator}_f(x, -s)$ —Operators in pattern RHS have explicit parameters.

Notes

Notes on the operation.

In a sequence of pattern-matching declarations, the first matching pattern in the order presented in this document defines the behavior for a given operand sequence.

In pattern matching, certain shorthand notations are used, as follows. Consider an operation which takes a P3109 value x in format f_x, and returns a P3109 value in format f_y. One matching pattern might be $\mathsf{Operator}(\mathsf{NaN}) \rightarrow \mathsf{NaN}$, where the input and output NaN values are in different formats, although this is not explicitly marked. A more explicit presentation, such as $\mathsf{Operator}(\mathsf{NaN}_{f_x}) \rightarrow \mathsf{NaN}_{f_y}$ was considered to increase difficulty of comprehension without a compensatory reduction in ambiguity.

Similarly, parameter subscripts are elided on the left-hand-side of patterns where they are common to all, but are written explicitly on the right.

Ancillary information, typically parameters, may include information such as an operand's format or rounding behaviors. An implementation may choose to provide that information using any appropriate mechanism. For example, available mechanisms in a hardware implementation might include passing the information in a hardware register, or as additional bits in an operation's opcode.

4.6 Functions

This section defines mathematical functions which are referenced in the later definitions of operations, but which themselves need not (and, in cases where the inputs or outputs are real, cannot) be provided in a conforming implementation.

Behavior that is specified through cases should be read from top to bottom.

4.6.1 Decode

Decode P3109 value to extended real or NaN.

Signature

$\mathsf{Decode}_f(x) \to X$

Parameters

f : format, width K_f, precision P_f, exponent bias b_f

Operands

x : P3109 value, in format f

Output

X : extended real value or NaN

Behavior

$\mathsf{Decode}(2^{\mathsf{K}_f-1}) \to \mathsf{NaN}$
$\mathsf{Decode}(2^{\mathsf{K}_f-1}-1) \to \infty$
$\mathsf{Decode}(2^{\mathsf{K}_f-1} < x < 2^{\mathsf{K}_f}) \to -\mathsf{Decode}_f(x - 2^{\mathsf{K}_f-1})$
$\mathsf{Decode}(0 \le x < 2^{\mathsf{K}_f-1}-1) \to X$

where

$$X = \begin{cases} (0 + T \times 2^{1-\mathsf{P}_f}) \times 2^{E+1} & \textbf{if } E = -b_f \quad \text{—Subnormal} \\ (1 + T \times 2^{1-\mathsf{P}_f}) \times 2^{E} & \text{Otherwise} \quad \text{—Normal} \end{cases}$$

$T = x \bmod 2^{\mathsf{P}_f-1}$

$E = x \div 2^{\mathsf{P}_f-1} - b_f$

4.6.2 Project

Project extended real value to P3109 format f, applying specified rounding and saturation.

Signature

$\mathsf{Project}_{f,\pi}(X) \to x$

Parameters

f : target format, precision P_f, exponent bias b_f, maximum finite value M_f
π : projection specification: rounding mode Rnd_π, saturation Sat_π

Operands

X : extended real value

Output

x : P3109 value, format f

Behavior

$\mathsf{Project}_{f,\pi}(X) \to x$

where

$R = \mathsf{RoundToPrecision}_{\mathsf{P}_f, b_f, \mathsf{Rnd}_\pi}(X)$ —Round to precision P_f with exponent unbounded from above
$S = \mathsf{Saturate}_{M_f}(\mathsf{Sat}_\pi, \mathsf{Rnd}_\pi, R)$
$x = \mathsf{Encode}_f(S)$

4.6.3 RoundToPrecision

Convert extended real value to extended real value representable with a given precision. The exponent is bounded below by $2 - \mathsf{P} - b$ and unbounded above.

Signature

$\mathsf{RoundToPrecision}_{\mathsf{P},b,\mathsf{Rnd}}(X) \to Z$

Parameters

P : integer precision
b : exponent bias
Rnd : rounding mode

Operands

X : extended real value

Output

Z : extended real value, of the form $N \times 2^E$, where $N \in \mathbb{Z}$, $0 \leq |N| < 2^{\mathsf{P}}$, and $2 - \mathsf{P} - b \leq E$.

Behavior

$\mathsf{RoundToPrecision}(X \in \{0, -\infty, \infty\}) \to X$
$\mathsf{RoundToPrecision}(X) \to Z$

where

$E = \max(\lfloor \log_2(|X|) \rfloor, 1 - b) - \mathsf{P} + 1$ —Subnormals handled by $\max(\cdot, 1 - b)$

$S = |X| \times 2^{-E}$ —Real-valued significand, to be rounded to integer

$\Delta = S - \lfloor S \rfloor$

$$\mathsf{CodeIsEven} = \begin{cases} \mathsf{IsEven}(\lfloor S \rfloor) & \textbf{if } \mathsf{P} > 1 \\ (\lfloor S \rfloor = 0) \vee \mathsf{IsEven}(E + b) & \text{Otherwise} \end{cases}$$

$I = \lfloor S \rfloor + \mathbb{1}[\mathsf{RoundAway}(\mathsf{Rnd})]$

$Z = \mathrm{sign}(X) \times I \times 2^E$

and

$\mathsf{RoundAway}(\mathsf{NearestTiesToEven}) = \Delta > 0.5 \vee \big(\Delta = 0.5 \wedge \neg \mathsf{CodeIsEven}\big)$

$\mathsf{RoundAway}(\mathsf{NearestTiesToAway}) = \Delta \geq 0.5$

$\mathsf{RoundAway}(\mathsf{TowardPositive}) = \Delta > 0 \wedge X > 0$

$\mathsf{RoundAway}(\mathsf{TowardNegative}) = \Delta > 0 \wedge X < 0$

$\mathsf{RoundAway}(\mathsf{TowardZero}) = \mathsf{False}$

Notes

The $\mathsf{P} = 1$ logic for $\mathsf{CodeIsEven}$ uses the observation that $I = 0 \implies Z = 0$, hence the code is 0, hence even.

Only precision and bias are needed for this definition, not format width.

Note that I may be set to 2^{P}, which might appear to preclude its representation in $\mathsf{P} - 1$ bits of explicit significand, but the computed real value is then $(1 + 0) \times 2^{E+1+\mathsf{P}}$, representable as the first number in the next binade.

4.6.4 Saturate

Saturate extended real to $\pm\infty$, or to maximum value, according to projection specification parameters Sat, Rnd.

Signature

$\mathsf{Saturate}_M(\mathsf{Sat}, \mathsf{Rnd}, X) \rightarrow Z$

Parameters

M : real maximum value

Operands

Sat : saturation mode
Rnd : rounding mode
X : extended real value

Output

Z : extended real value

Behavior

$\mathsf{Saturate}(*, *, X \in [-M, M]) \rightarrow X$

$\mathsf{Saturate}(\mathsf{SatMax}, *, X \notin [-M, M]) \rightarrow \mathrm{sign}(X) \times M$

$\mathsf{Saturate}(\mathsf{SatFinite}, *, X \in \{\pm\infty\}) \rightarrow X$
$\mathsf{Saturate}(\mathsf{SatFinite}, *, |X| \in [M, \infty)) \rightarrow \mathrm{sign}(X) \times M$

$\mathsf{Saturate}(\mathsf{OvfInf}, *, X \in \{\pm\infty\}) \rightarrow X$
$\mathsf{Saturate}(\mathsf{OvfInf}, \mathsf{TowardZero}, |X| \in [M, \infty)) \rightarrow \mathrm{sign}(X) \times M$
$\mathsf{Saturate}(\mathsf{OvfInf}, \mathsf{TowardPositive}, X \in (-\infty, -M)) \rightarrow -M$
$\mathsf{Saturate}(\mathsf{OvfInf}, \mathsf{TowardNegative}, X \in (M, \infty)) \rightarrow M$
$\mathsf{Saturate}(\mathsf{OvfInf}, *, X) \rightarrow \mathrm{sign}(X) \times \infty$

Note

In Saturate, all values above M are sent to either M or ∞. Nevertheless, Project (§4.6.2) has the property that values below $M + \frac{1}{2}\,\mathrm{ulp}(M)$ will round to M because RoundToPrecision precedes saturation in the definition of Project.

4.6.5 Encode

Encode extended real value or NaN to P3109 format f. Encode must be applied only to a value which is in the value set of format f. Such a value may be produced by, for example RoundToPrecision and Saturate, or the input may be known to be in the value set, for example, in the negation of a value already in the set.

Signature

$\mathsf{Encode}_f(X) \to z$

Parameters

f : target format, width K_f, precision P_f, exponent bias b_f

Operands

X : extended real value or NaN, in the value set of format f

Output

z : P3109 value, format f

Behavior

$\mathsf{Encode}(\mathsf{NaN}) \to 2^{\mathsf{K}_f-1}$
$\mathsf{Encode}(\infty) \to 2^{\mathsf{K}_f-1}-1$
$\mathsf{Encode}(X<0) \to \mathsf{Encode}_f(-X)+2^{\mathsf{K}_f-1}$
$\mathsf{Encode}(0) \to 0$
$\mathsf{Encode}(X>0) \to z$

where

$$z = \begin{cases} T & \textbf{if } S < 2^{\mathsf{P}_f-1} \\ T + (E+b_f)\times 2^{\mathsf{P}_f-1} & \text{Otherwise} \end{cases}$$ —Subnormals

and

$E = \max(\lfloor \log_2(X) \rfloor, 1-b_f)$
$S = X \times 2^{-E} \times 2^{\mathsf{P}_f-1}$ —S is the significand
$T = S \mod 2^{\mathsf{P}_f-1}$

Note

Because of the precondition that X is in the value set of format f, it follows that $S \in \mathbb{N}$.

4.7 Conversion operations

4.7.1 Specification of IEEE Std 754 formats

In specifying IEEE Std 754 formats, the symbol ϕ is used, from which format parameters are extracted as needed as follows:

M_ϕ : maximum finite value (e.g., 65504.0 for Binary16)
P_ϕ : precision (e.g., 11 for Binary16)
B_ϕ : exponent bias (e.g., 15 for Binary16)

We will make use of the function Encode754, defined as follows:

Signature

$\mathsf{Encode754}_\phi(X) \rightarrow z$

Parameters

ϕ : target IEEE Std 754 format

Operands

X : NaN or extended real value, in the value set of format ϕ

Output

z : IEEE Std 754 value, format ϕ

Behavior

$\mathsf{Encode754}(\mathsf{NaN}) \rightarrow$ Any quiet IEEE Std 754 NaN
$\mathsf{Encode754}(X) \rightarrow$ The code in ϕ that decodes to X

Notes

In this document, the Encode754 function is only called with arguments in the value set of format ϕ, therefore encoding is unambiguous and independent of rounding mode. Conversely, it is an error in this document if this condition is not guaranteed.

An implementation may return any quiet NaN. It is recommended that the quiet NaN with zero payload is returned.

4.7.2 Conversion from IEEE Std 754 formats to P3109

Convert a value in an IEEE Std 754 format to the corresponding value in a given P3109 format, considering rounding and saturation.

Signature

$\mathsf{ConvertToP3109}_{\phi,f,\pi}(X) \to z$

Parameters

ϕ : source IEEE Std 754 format
f : target format
π : projection specification

Operands

X : IEEE-754 value, format ϕ

Output

z : P3109 value, format f

Behavior

$\mathsf{ConvertToP3109}(\text{Any IEEE Std 754 NaN}) \to \mathsf{NaN}$
$\mathsf{ConvertToP3109}(X) \to \mathsf{Project}_{f,\pi}(X') \quad \textbf{where} \quad X' = \mathsf{AsExtendedReal}_{\phi}(X)$

Note

$\mathsf{AsExtendedReal}_{\phi}$ is a function which converts a non-NaN encoded IEEE Std 754 value to a value in the extended reals. $\mathsf{AsExtendedReal}(\text{-}0) \to 0$.

4.7.3 Conversion from P3109 to IEEE Std 754

Convert a P3109 value to a value in an IEEE Std 754 format. We note that for IEEE Std 754 binary{K} formats for $\mathsf{K} > 16$, all P3109 values are values in the target format, hence the conversion of non-NaN values is unambiguous. For Binary16 outputs, some P3109 values will be out of the target range; it is necessary to be precise about rounding and saturation.

Signature

$\mathsf{ConvertToIEEE754}_{f,\pi,\phi}(x) \to X$

Parameters

f : input format
π : projection specification
ϕ : IEEE Std 754 result format, precision P_ϕ, bias B_ϕ, maximum value M_ϕ

Operands

x : P3109 value, format f

Output

X : IEEE-754 value, format ϕ

Behavior

$\mathsf{ConvertToIEEE754}(\mathsf{NaN}) \to \mathsf{Encode754}_\phi(\mathsf{NaN})$
$\mathsf{ConvertToIEEE754}(x) \to \mathsf{Encode754}_\phi(X)$
where
$Y = \mathsf{Decode}_f(x)$
$R = \mathsf{RoundToPrecision}_{P_\phi, B_\phi, \mathsf{Rnd}_\pi}(Y)$
$X = \mathsf{Saturate}_{M_\phi}(\mathsf{Sat}_\pi, \mathsf{Rnd}_\pi, R)$

4.7.4 Conversion from P3109 to P3109

Convert a P3109 value to another P3109 format.

Signature

$\mathsf{ConvertP3109ToP3109}_{f_x, f_z, \pi}(x) \to z$

Parameters

f_x : input format
f_z : target format
π : projection specification

Operands

x : P3109 value, format f_x

Output

z : P3109 value, format f_z

Behavior

$\mathsf{ConvertP3109ToP3109}(\mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{ConvertP3109ToP3109}(x) \to \mathsf{Project}_{f_z, \pi}(X) \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x)$

4.8 Arithmetic operations

Arithmetic operations which take one or more P3109 values as arguments, and return one or more P3109 values are defined in this section. Arithmetic operations which mix IEEE Std 754 and P3109 values are described in §4.9.

4.8.1 Unary sign operations

Operations which take P3109 values as input, and whose outputs are guaranteed to be exact values in the same P3109 format. These operations do not permit the output format to be different from the input format.

Signature

$\mathsf{Operation}_f(x) \to z$

Parameters

f : input and output format

Operands

x : P3109 value, format f

Output

z : P3109 value, format f

Behavior

$$\mathsf{Abs}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{Encode}_f(|X|) & \mathsf{Otherwise} \end{cases}$$
$$\textbf{where} \quad X = \mathsf{Decode}_f(x)$$

$$\mathsf{Negate}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{Encode}_f(-X) & \mathsf{Otherwise} \end{cases}$$
$$\textbf{where} \quad X = \mathsf{Decode}_f(x)$$

Note

The extended reals do not have negative zero, hence $-0 = 0$ and $\mathsf{Abs}(0) = 0$.

4.8.2 Binary sign operations

Signature

$\mathsf{Operation}_f(x, y) \to z$

Parameters

f : input and output format

Operands

x : P3109 value, format f
y : P3109 value, format f

Output

z : P3109 value, format f

Behavior

$$\mathsf{CopySign}(x, y) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \vee \mathsf{isNaN}(y) \\ \mathsf{Abs}(x) & Y \geq 0 \\ \mathsf{Negate}(\mathsf{Abs}(x)) & Y < 0 \end{cases}$$
$$\textbf{where} \quad Y = \mathsf{Decode}_f(y)$$

4.8.3 Binary arithmetic operations

These operations take two P3109 values as input, and return a P3109 value. The definitions allow for all input and output formats to differ, a given implementation may supply any subset of the defined operations.

Signature

$\mathsf{Operation}_{f_x,f_y,f_z,\pi}(x, y) \to z$

Parameters

f_x : format of x
f_y : format of y
f_z : format of z
π : projection specification

Operands

x : P3109 value, format f_x
y : P3109 value, format f_y

Output

z : P3109 value, format f_z

Behavior

$\mathsf{Add}(*, \mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{Add}(\mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{Add}(\text{-}\mathsf{Inf}, \mathsf{Inf}) \to \mathsf{NaN}$
$\mathsf{Add}(\mathsf{Inf}, \text{-}\mathsf{Inf}) \to \mathsf{NaN}$
$\mathsf{Add}(x, y) \to \mathsf{Project}_{f_z,\pi}(X + Y) \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x), Y = \mathsf{Decode}_{f_y}(y)$

$\mathsf{Subtract}(*, \mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{Subtract}(\mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{Subtract}(\mathsf{Inf}, \mathsf{Inf}) \to \mathsf{NaN}$
$\mathsf{Subtract}(\text{-}\mathsf{Inf}, \text{-}\mathsf{Inf}) \to \mathsf{NaN}$
$\mathsf{Subtract}(x, y) \to \mathsf{Project}_{f_z,\pi}(X - Y) \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x), Y = \mathsf{Decode}_{f_y}(y)$

$\mathsf{Multiply}(*, \mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{Multiply}(\mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{Multiply}(0, \pm\mathsf{Inf}) \to \mathsf{NaN}$
$\mathsf{Multiply}(\pm\mathsf{Inf}, 0) \to \mathsf{NaN}$
$\mathsf{Multiply}(x, y) \to \mathsf{Project}_{f_z,\pi}(X \times Y) \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x), Y = \mathsf{Decode}_{f_y}(y)$

$\mathsf{Divide}(*, \mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{Divide}(\mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{Divide}(*, 0) \to \mathsf{NaN}$
$\mathsf{Divide}(x, y) \to \mathsf{Project}_{f_z,\pi}(X/Y) \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x), Y = \mathsf{Decode}_{f_y}(y)$

Notes

Divide(x,±Inf) where x is finite yields 0.
Divide(x,0) yields NaN. With x finite, it would be inconsistent to return Inf: $1/(1/\text{-}\mathsf{Inf}) \to \mathsf{Inf}$.

4.8.4 Unary mathematical operations

Signature

$\mathsf{Operation}_{f_x, f_z, \pi}(x) \to z$

Parameters

f_x : input format
f_z : output format
π : projection specification

Operands

x : P3109 value, format f_x

Output

z : P3109 value, format f_z

Behavior

$$\mathsf{Sqrt}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{NaN} & X < 0 \\ \mathsf{Project}_{f_z,\pi}(\sqrt{X}) & X \geq 0 \end{cases} \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x)$$

$$\mathsf{Exp}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{Project}_{f_z,\pi}(e^X) & \mathsf{Otherwise} \end{cases} \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x)$$

$$\mathsf{Exp2}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{Project}_{f_z,\pi}(2^X) & \mathsf{Otherwise} \end{cases} \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x)$$

$$\mathsf{Log}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{NaN} & X < 0 \\ \mathsf{Project}_{f_z,\pi}(\log(X)) & X \geq 0 \end{cases} \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x)$$

$$\mathsf{Log2}(x) \to \begin{cases} \mathsf{NaN} & \mathsf{isNaN}(x) \\ \mathsf{NaN} & X < 0 \\ \mathsf{Project}_{f_z,\pi}(\log_2(X)) & X \geq 0 \end{cases} \quad \textbf{where} \quad X = \mathsf{Decode}_{f_x}(x)$$

4.8.5 Scaled addition

Compute $X \times 2^{s_x} + Y \times 2^{s_y}$, and return a P3109 value. Scaling is applied in the extended reals, before projection to the target format.

Signature

$\mathsf{AddScaled}_{f_x, f_y, f_z, \pi}(x, s_x, y, s_y) \to z$

Parameters

f_x : format of x
f_y : format of y
f_z : format of z
π : projection specification

Operands

x : P3109 value, format f_x
s_x : integer log-scale factor for x
y : P3109 value, format f_y
s_y : integer log-scale factor for y

Output

z : P3109 value, format f_z

Behavior

$\mathsf{AddScaled}(\mathsf{NaN}, *, *, *) \to \mathsf{NaN}$
$\mathsf{AddScaled}(*, *, \mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{AddScaled}(\text{-}\mathsf{Inf}, *, \mathsf{Inf}, *) \to \mathsf{NaN}$
$\mathsf{AddScaled}(\mathsf{Inf}, *, \text{-}\mathsf{Inf}, *) \to \mathsf{NaN}$
$\mathsf{AddScaled}(x, s_x, y, s_y) \to \mathsf{Project}_{f_z, \pi}(Z)$

$$\textbf{where}$$
$$Z = X \times 2^{s_x} + Y \times 2^{s_y}$$
$$X = \mathsf{Decode}_{f_x}(x)$$
$$Y = \mathsf{Decode}_{f_y}(y)$$

Note

The valid range of the log-scale factors s_x and s_y is not specified. Implementers shall declare the provision of AddScaled with specified constraints on legal values for these operands.

Example declarations might include

"AddScaled, with log-scale factors in $\{-32 \dots 31\}$"

"AddScaled, with $s_x = 0$, $s_y \in \{-128 \dots 127\}$"

"AddScaled, with $s_x \in \{-128 \dots 127\}$, $s_y = s_x$"

See §4.1 for further discussion.

4.8.6 Scaled multiplication

Compute $X \times Y \times 2^s$, and return a P3109 value. Scaling is applied in the extended reals, before projection to the target format.

Signature

$\mathsf{MultiplyScaled}_{f_x, f_y, f_z, \pi}(x, y, s) \to z$

Parameters

f_x : format of x
f_y : format of y
f_z : format of z
π : projection specification

Operands

x : P3109 value, format f_x
y : P3109 value, format f_y
s : integer log-scale factor

Output

z : P3109 value, format f_z

Behavior

$\mathsf{MultiplyScaled}(*, \mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{MultiplyScaled}(\mathsf{NaN}, *, *) \to \mathsf{NaN}$
$\mathsf{MultiplyScaled}(0, \pm\mathsf{Inf}, *) \to \mathsf{NaN}$
$\mathsf{MultiplyScaled}(\pm\mathsf{Inf}, 0, *) \to \mathsf{NaN}$
$\mathsf{MultiplyScaled}(x, y, s) \to \mathsf{Project}_{f_z, \pi}(Z)$

$$\textbf{where}$$
$$Z = X \times Y \times 2^s$$
$$X = \mathsf{Decode}_{f_x}(x)$$
$$Y = \mathsf{Decode}_{f_y}(y)$$

Note

The valid range of the log-scale factor s is not specified. Implementations shall declare the provision of MultiplyScaled with specified constraints on legal values for these operands.

An example such declaration might be

> "MultiplyScaled, with log-scale factor in $\{-32 \ldots 31\}$"

See §4.1 for further discussion.

4.9 Mixed IEEE Std 754 and P3109 operations

This section describes operations which take a mix of IEEE Std 754 and P3109 operands and return IEEE Std 754 values. An implementation is free to define additional fused operations in which P3109 operands are upconverted to IEEE Std 754 before operating.

This section addresses operations that cannot be expressed as a fused sequence of operations. For instance, it is not possible to upconvert all values from binary8p1 to Binary16. Moreover, the operations described here may include saturation and rounding modes not available in definitions based solely on upconversion.

Like the operations defined previously, these are superset specifications. An implementation might provide any subset of the parameterized set of operations, accompanied by a statement specifying which subset is implemented.

These definitions make use of the functions AsExtendedReal and Encode754 defined above.

4.9.1 Scaled Fused Multiply Add

Compute $Z = A \times 2^{s_a} + X \times Y \times 2^s$, with A and Z in the same format. Scaling is applied in the extended reals, before projection to the target format.

Signature

$\mathsf{ScaledFMA}_{\phi,f_x,f_y,\pi}(a, s_a, x, y, s) \to z$

Parameters

ϕ : IEEE Std 754 format of a and target, precision P_ϕ, bias B_ϕ, maximum value M_ϕ
f_x : format of x
f_y : format of y
π : projection specification

Operands

a : IEEE Std 754 value, format ϕ
s_a : integer log-scale factor —or NaN?
x : P3109 value, format f_x
y : P3109 value, format f_y
s : integer log-scale factor

Output

z : IEEE Std 754 value, format ϕ

Behavior

$\mathsf{ScaledFMA}(\mathsf{NaN}, *, *, *, *) \to \mathsf{NaN}$
$\mathsf{ScaledFMA}(*, *, \mathsf{NaN}, *, *) \to \mathsf{NaN}$
$\mathsf{ScaledFMA}(*, *, *, \mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{ScaledFMA}(a, s_a, x, y, s) \to z$

where

$A = \mathsf{AsExtendedReal}_\phi(a)$
$X = \mathsf{Decode}_{f_x}(x)$
$Y = \mathsf{Decode}_{f_y}(y)$
$Z = A \times 2^{s_a} + X \times Y \times 2^s$
$Z' = \mathsf{RoundToPrecision}_{P_\phi,B_\phi,\mathsf{Rnd}_\pi}(Z)$
$Z'' = \mathsf{Saturate}_{M_\phi}(\mathsf{Sat}_\pi, \mathsf{Rnd}_\pi, Z')$
$z = \mathsf{Encode754}_\phi(Z'')$ —Z'' guaranteed representable in format ϕ

Note

The valid range of the log-scale factors are not specified. Implementations shall declare the provision of ScaledFMA with specified constraints on legal values for these operands.

An example such declaration might be:
"ScaledFMA, with log-scale factors in $\{-32 \ldots 31\}$, $f_x = f_y$ in $\{\mathsf{binary8p3}, \mathsf{binary8p4}\}$, ϕ in $\{\mathsf{Binary16}, \mathsf{Binary32}\}$"

4.10 Comparisons, predicates, and classification

4.10.1 Minimum and Maximum

These operations take two P3109 values as input, and return a P3109 value. Operands and return value are in the same format.

Signature

$\mathsf{Operation}_f(x, y) \to z$

Parameters

f : format

Operands

x : P3109 value, format f
y : P3109 value, format f

Output

z : P3109 value, format f

Behavior

$\mathsf{Minimum}(*, \mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{Minimum}(\mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{Minimum}(x, y) \to \mathsf{Encode}_f(\min(X, Y)) \quad \textbf{where} \quad X = \mathsf{Decode}_f(x), Y = \mathsf{Decode}_f(y)$

$\mathsf{Maximum}(*, \mathsf{NaN}) \to \mathsf{NaN}$
$\mathsf{Maximum}(\mathsf{NaN}, *) \to \mathsf{NaN}$
$\mathsf{Maximum}(x, y) \to \mathsf{Encode}_f(\max(X, Y)) \quad \textbf{where} \quad X = \mathsf{Decode}_f(x), Y = \mathsf{Decode}_f(y)$

Note

The operations **minimumNumber**, **maximumNumber**, defined by IEEE Std 754-2019 are not defined in this document.

4.10.2 Comparisons

Comparison operators take two P3109 operands and return a boolean. Any NaN operand yields the result False.

The below specifications are expressed with the following substitutions for compareOp and $CompareExpr$:

compareOp	$CompareExpr$
compareLess	$X < Y$
compareLessEqual	$X \leq Y$
compareEqual	$X = Y$
compareGreaterEqual	$X > Y$
compareGreater	$X \geq Y$

Signature

compare$Op_{f_x, f_y}(x, y) \rightarrow$ Boolean

Parameters

f_x : format of x
f_y : format of y

Operands

x : P3109 value, format f_x
y : P3109 value, format f_y

Output

b : boolean value

Behavior

compareOp(NaN, NaN) $\rightarrow$ False
compareOp(NaN, y) $\rightarrow$ False
compareOp(x, NaN) $\rightarrow$ False
compare$Op(x, y) \rightarrow$ $CompareExpr$

where

$X = \text{Decode}_{f_x}(x)$
$Y = \text{Decode}_{f_y}(y)$

4.10.3 Predicates and classification

Conforming implementations shall provide the classification predicates and the classifier operation defined below.

The classification operations comprise: 1) a set of predicate functions with a boolean return value, taking a single P3109 value as input; 2) a classifier operation $\mathrm{class}(x)$ that returns a single value of enumeration type, describing the input value's properties.

Signature

$\mathsf{is}Class_f(x) \to b$

Parameters

f : format of x

Operands

x : P3109 value, format f

Output

b : boolean value

Behavior

$\mathsf{isZero}(\mathsf{NaN}) \to \mathsf{False}$
$\mathsf{isZero}(x) \to \mathsf{Decode}_f(x) = 0$

$\mathsf{isOne}(\mathsf{NaN}) \to \mathsf{False}$
$\mathsf{isOne}(x) \to \mathsf{Decode}_f(x) = 1$

$\mathsf{isNaN}(\mathsf{NaN}) \to \mathsf{True}$
$\mathsf{isNaN}(x) \to \mathsf{False}$

$\mathsf{isSignMinus}(\mathsf{NaN}) \to \mathsf{True}$
$\mathsf{isSignMinus}(x) \to \mathsf{Decode}_f(x) < 0$

$\mathsf{isNormal}(x \in \{0, \text{-}\mathsf{Inf}, \mathsf{Inf}, \mathsf{NaN}\}) \to \mathsf{False}$
$\mathsf{isNormal}(x) \to (x \mod 2^{\mathsf{K}_f - 1}) \div 2^{\mathsf{P}_f - 1} > 0$

$\mathsf{isSubnormal}(x \in \{0, \text{-}\mathsf{Inf}, \mathsf{Inf}, \mathsf{NaN}\}) \to \mathsf{False}$
$\mathsf{isSubnormal}(x) \to (x \mod 2^{\mathsf{K}_f - 1}) \div 2^{\mathsf{P}_f - 1} = 0$

$\mathsf{isFinite}(x) \to x \notin \{\text{-}\mathsf{Inf}, \mathsf{Inf}, \mathsf{NaN}\}$

$\mathsf{isInfinite}(x) \to x \in \{\text{-}\mathsf{Inf}, \mathsf{Inf}\}$

$\mathsf{isSignaling}(x) \to \mathsf{False}$ —All P3109 formats have one NaN, which does not signal

$\mathsf{isCanonical}(x) \to \mathsf{True}$ —There are no non-canonical P3109 interchange formats

4.10.4 Classifier operation

The Classifier operation $\mathsf{class}(x)$ tells which of the eight classes x falls into as defined by Table 4.

Signature

$\mathsf{class}_f(x) \rightarrow c$

Parameters

f : format of x

Operands

x : P3109 value, format f

Output

c : enumeration

Behavior

$\mathsf{class}(x) \rightarrow \mathrm{ClassEnum}$

Table 4: Classifier operation

ClassEnum	*Condition*
clsNaN	isNaN(x)
clsNegativeInfinity	isInfinite(x) $\wedge$ isSignMinus(x)
clsNegativeNormal	isNormal(x) $\wedge$ isSignMinus(x)
clsNegativeSubnormal	isSubnormal(x) $\wedge$ isSignMinus(x)
clsZero	isZero(x)
clsPositiveSubnormal	isSubnormal(x) $\wedge \neg$isSignMinus(x)
clsPositiveNormal	isNormal(x) $\wedge \neg$isSignMinus(x)
clsPositiveInfinity	isInfinite(x) $\wedge \neg$isSignMinus(x)

4.10.5 Total order predicate

The $\mathsf{totalOrder}(x, y)$ predicate provides a total ordering over each P3109 format's value set.

Signature

$\mathsf{totalOrder}_{f_x, f_y}(x, y) \to b$

Operands

f_x : format of x
f_y : format of y

Operands

x : P3109 value, format f_x
y : P3109 value, format f_y

Output

b : boolean value

Behavior

$\mathsf{totalOrder}(\mathsf{NaN}, x) \to \mathsf{True}$
$\mathsf{totalOrder}(x, \mathsf{NaN}) \to \mathsf{False}$
$\mathsf{totalOrder}(x, y) \to \mathsf{compareLessEqual}_{f_x, f_y}(x, y)$

Note

The above definition is consistent with the IEEE Std 754-2019 definition of totalOrder. In particular, among P3109 formats, there is a single NaN and it always compares as the most-negative value.

4.10.6 Comparison predicates

Conforming implementations shall provide the comparison predicates defined by Table 5 and the totalOrder(x, y) predicate.

Comparison operations are two-argument predicates, and their negations, that return True or False. Comparisons may be ordered or unordered. A comparison is considered unordered iff either argument is NaN. All other comparisons are ordered.

For $\{=, >, \geq, <, \leq, \lessgtr\}$, if any argument is NaN, the result is False.

For $\{\neq, \not>, \not\geq, \not<, \not\leq, \not\lessgtr\}$, if any argument is NaN, the result is True.

Otherwise, the result of a comparison shall match the mathematical result.

Table 5: Comparison predicates and negations

Math symbol	Predicate *true relations*	Math symbol	Negation of predicate *true relations*
$=$	compareEqual *equal*	$\neq$, NOT $=$	compareNotEqual *less than, greater than, unordered*
$>$	compareGreater *greater than*	$\not>$, NOT $>$	compareNotGreater *less than, equal, unordered*
$\geq$	compareGreaterEqual *greater than, equal*	$\not\geq$, NOT $\geq$	compareLessUnordered *less than, unordered*
$<$	compareLess *less than*	$\not<$, NOT $<$	compareNotLess *greater than, equal, unordered*
$\leq$	compareLessEqual *less than, equal*	$\not\leq$, NOT $\leq$	compareGreaterUnordered *greater than, unordered*
$\lessgtr$	compareOrdered *less than, equal, greater than*	$\not\lessgtr$, NOT $\lessgtr$	compareUnordered *unordered*

Appendices

A Rationales

A.1 Use of infinity in computation of attention masks

This section expands the rationale in §3.6. A common use for ∞ is to create masks, for example, in Transformer models in machine learning [5].

These values, assembled in mask matrix M with values $M_{ij} \in \{0, -\infty\}$ are typically added to computed values A, in a computation such as:

$$\log\left(\sum \exp(\tau \times (A + M))\right)$$

where τ is a "temperature" or "base" parameter [6]. This calculation depends on the property $\exp(\tau \times (A_{ij} - \infty)) = 0$.

If a floating-point encoding does not provide infinity, then instead M_{ij} will be replaced by a large float (e.g. 224 is the largest finite binary8p4 value). This is not in itself a difficulty: if all the A values are bounded (e.g. the results of a softmax operation are bounded above by 1.0), then $\exp(1.0 - 224.0)$ is an extremely small number, which will certainly round to zero. Therefore, an explicit representation of infinity is *not* needed in order for this computation to yield its desired value.

However, careful implementations do not execute the calculation as written, and instead fuse the $\log(\sum_i \exp(v_i))$ operation into a single operation $\mathsf{logsumexp}(v)$, whose implementation makes use of the identity transformation

$$\mathsf{logsumexp}(v) \to \mathsf{logsumexp}(v - \max(v)) + \max(v)$$

Without the "sticky" properties of Inf, this would produce incorrect answers.

For example, in a format where maxFinite=240 without Inf, and maxFinite=224 with Inf:

$$\mathsf{logsumexp}(t * [-224, -\infty]) \to \mathsf{logsumexp}(t * [0, -\infty])$$

while

$$\mathsf{logsumexp}(t * [-224, -240]) \to \mathsf{logsumexp}(t * [0, -16])$$

If $t = 1$ and all calculations are done in 8-bit floating-point, then the answer will be the same, because $\exp(-16) \approx 1.1 \times 10^{-7}$, which will round to zero in all precisions $\mathsf{P} > 2$; but if t is small, or calculations are done in mixed precision, as is common with 8-bit floating-point, the loss of "stickiness" will silently yield unexpected answers. It is not expected that the full calculation shall be done in 8-bit floating-point, but the subtraction of the maximum value (and computation of the maximum) might reasonably be in 8-bit floating-point.

A.2 Eight Bit Formats

Eight bit floating-point representations have received much attention for their usefulness and efficacy in machine learning, especially deep learning. Various 8-bit floating-point formats have been proposed, investigated in research papers, and some have been modeled in software. Four precisions [3, 4, 5, and 6 bit precisions] have become generally accepted as providing greater operational benefits than the others. Overall [there are exceptions], current efforts focus

more on precisions of 3 and 4 bits (exponent fields of 5 and 4 bits, respectively). The specifics of third-party proposals for 8-bit floating-point representations vary and can differ from one precision to another. Regardless, the precisions emphasized in current research and other third-party work do share the same focus.

A.3 Essential Formats

A.3.1 binary8p3

This format has 3 subnormal and 123 normal magnitudes. The subnormal magnitudes cover $[1.0 * 2^{-17}, 3.0 * 2^{-17}]$. The normal magnitudes cover $[1.0 * 2^{-15}, 4195.0]$, There are 31 normal binades with 4 magnitudes per complete binade [binade 2^{15} has 3 magnitudes, the 4^{th} is used for Inf].

A.3.2 binary8p4

This format has 7 subnormal and 119 normal magnitudes. The subnormal magnitudes cover $[1.0 * 2^{-10}, 7.0 * 2^{-10}]$. The normal magnitudes cover $[1.0 * 2^{-7}, 224.0]$. There are 15 normal binades with 8 magnitudes per complete binade [binade 2^{7} has 7 magnitudes, the 8^{th} is used for Inf],

A.3.3 binary8p5

This format has 15 subnormal and 111 normal magnitudes. The subnormal magnitudes cover $[1.0 * 2^{-7}, 15.0 * 2^{-7}]$. The normal magnitudes cover $[0.125, 15.0]$. There are 7 normal binades with 16 magnitudes per complete binade [binade 2^{3} has 15 magnitudes, the 16^{th} is used for Inf],

A.3.4 binary8p6

This format has 31 subnormal and 95 normal magnitudes. The subnormal magnitudes cover $[1.0 * 2^{-6}, 31.0 * 2^{-6}]$. The normal magnitudes cover $[0.5, 3.875]$. There are 3 normal binades with 32 magnitudes per complete binade [binade 2^{1} has 31 magnitudes, the 32^{nd} is used for Inf],

B External Formats

This table summarizes the points of agreement and of difference between the formats proposed in this document and a number of existing formats, some of which have hardware implementations.

OCP: Open Compute Platform [7], describing hardware implementations including nVidia, Intel, and ARM.

AGQ: AMD, Graphcore, Qualcomm[8], implemented in Graphcore's C600 product, and AMD's gfx940.

TSL: Tesla Dojo Technology [9], A Guide to Tesla's Configurable floating-point Formats & Arithmetic

Format	P3109			OCP		AGQ		TSL	
Subformat	P3	P4	P5	E5	E4	E5	E4	E4	E5
Special values shared by all subformats	Y			N		Y		N	
Exactly one NaN	Y			N		Y		Y	
Positive and negative infinity	Y			Y	N	N		N	
Include negative zero	N			Y		N		N	
Max exponent emax	15	7	3	15	8	15	7	N/A	N/A

C 8-bit Value Tables

Value tables mapping 8-bit strings to value sets are provided in this section.

A typical entry is of the form:

```
HEX  = BINARY     = BINARY_FLOAT   = DECIMAL
0x0b = 0_0001_011 = +0b1.011×2^-7 = 0.0107421875
```

Where the fields are interpreted as follows:

`HEX`	Hexadecimal encoding of the code point
`BINARY`	Binary expansion of the code point, underscores separate < `sign` >_< `exponent` >_< `significand` >
`BINARY_FLOAT`	The precise float value as a binary fraction followed by `2^`e with exponent e
`DECIMAL`	A decimal expansion of the value. If the expansion is not an exact representation of the precise float value, the equals sign is replaced by "approximately equals" (≈).

In addition, entries for subnormal and special values are rendered in color as follows:

`0x05 = 0_0000_101 = +0b0.101×2^-7 = 0.0048828125`	Subnormal value
`0x80 = 1_0000_000 = NaN`	Special value (NaN, +Inf, -Inf)

C.1 Value Table: P1, emin $= -62$, emax $= 63$ (nonsymmetric)

0x00 = 0_0000000_ = 0.0
0x01 = 0_0000001_ = +0b1.0×2^-62 ≈ 2.1684043E-19
0x02 = 0_0000010_ = +0b1.0×2^-61 ≈ 4.3368087E-19
0x03 = 0_0000011_ = +0b1.0×2^-60 ≈ 8.6736174E-19
0x04 = 0_0000100_ = +0b1.0×2^-59 ≈ 1.7347235E-18
0x05 = 0_0000101_ = +0b1.0×2^-58 ≈ 3.469447E-18
0x06 = 0_0000110_ = +0b1.0×2^-57 ≈ 6.9388939E-18
0x07 = 0_0000111_ = +0b1.0×2^-56 ≈ 1.3877788E-17
0x08 = 0_0001000_ = +0b1.0×2^-55 ≈ 2.7755576E-17
0x09 = 0_0001001_ = +0b1.0×2^-54 ≈ 5.5511151E-17
0x0a = 0_0001010_ = +0b1.0×2^-53 ≈ 1.110223E-16
0x0b = 0_0001011_ = +0b1.0×2^-52 ≈ 2.220446E-16
0x0c = 0_0001100_ = +0b1.0×2^-51 ≈ 4.4408921E-16
0x0d = 0_0001101_ = +0b1.0×2^-50 ≈ 8.8817842E-16
0x0e = 0_0001110_ = +0b1.0×2^-49 ≈ 1.7763568E-15
0x0f = 0_0001111_ = +0b1.0×2^-48 ≈ 3.5527137E-15
0x10 = 0_0010000_ = +0b1.0×2^-47 ≈ 7.1054274E-15
0x11 = 0_0010001_ = +0b1.0×2^-46 ≈ 1.4210855E-14
0x12 = 0_0010010_ = +0b1.0×2^-45 ≈ 2.8421709E-14
0x13 = 0_0010011_ = +0b1.0×2^-44 ≈ 5.6843419E-14
0x14 = 0_0010100_ = +0b1.0×2^-43 ≈ 1.1368684E-13
0x15 = 0_0010101_ = +0b1.0×2^-42 ≈ 2.2737368E-13
0x16 = 0_0010110_ = +0b1.0×2^-41 ≈ 4.5474735E-13
0x17 = 0_0010111_ = +0b1.0×2^-40 ≈ 9.094947E-13
0x18 = 0_0011000_ = +0b1.0×2^-39 ≈ 1.8189894E-12
0x19 = 0_0011001_ = +0b1.0×2^-38 ≈ 3.6379788E-12
0x1a = 0_0011010_ = +0b1.0×2^-37 ≈ 7.2759576E-12
0x1b = 0_0011011_ = +0b1.0×2^-36 ≈ 1.4551915E-11
0x1c = 0_0011100_ = +0b1.0×2^-35 ≈ 2.910383E-11
0x1d = 0_0011101_ = +0b1.0×2^-34 ≈ 5.8207661E-11
0x1e = 0_0011110_ = +0b1.0×2^-33 ≈ 1.1641532E-10
0x1f = 0_0011111_ = +0b1.0×2^-32 ≈ 2.3283064E-10
0x20 = 0_0100000_ = +0b1.0×2^-31 ≈ 4.6566129E-10
0x21 = 0_0100001_ = +0b1.0×2^-30 ≈ 9.3132257E-10
0x22 = 0_0100010_ = +0b1.0×2^-29 ≈ 1.8626451E-09
0x23 = 0_0100011_ = +0b1.0×2^-28 ≈ 3.7252903E-09
0x24 = 0_0100100_ = +0b1.0×2^-27 ≈ 7.4505806E-09
0x25 = 0_0100101_ = +0b1.0×2^-26 ≈ 1.4901161E-08
0x26 = 0_0100110_ = +0b1.0×2^-25 ≈ 2.9802322E-08
0x27 = 0_0100111_ = +0b1.0×2^-24 ≈ 5.9604645E-08
0x28 = 0_0101000_ = +0b1.0×2^-23 ≈ 1.1920929E-07
0x29 = 0_0101001_ = +0b1.0×2^-22 ≈ 2.3841858E-07
0x2a = 0_0101010_ = +0b1.0×2^-21 ≈ 4.7683716E-07
0x2b = 0_0101011_ = +0b1.0×2^-20 ≈ 9.5367432E-07
0x2c = 0_0101100_ = +0b1.0×2^-19 ≈ 1.9073486E-06
0x2d = 0_0101101_ = +0b1.0×2^-18 ≈ 3.8146973E-06
0x2e = 0_0101110_ = +0b1.0×2^-17 ≈ 7.6293945E-06
0x2f = 0_0101111_ = +0b1.0×2^-16 ≈ 1.5258789E-05
0x30 = 0_0110000_ = +0b1.0×2^-15 ≈ 3.0517578E-05
0x31 = 0_0110001_ = +0b1.0×2^-14 ≈ 6.1035156E-05
0x32 = 0_0110010_ = +0b1.0×2^-13 ≈ 0.00012207031
0x33 = 0_0110011_ = +0b1.0×2^-12 = 0.000244140625
0x34 = 0_0110100_ = +0b1.0×2^-11 = 0.00048828125
0x35 = 0_0110101_ = +0b1.0×2^-10 = 0.0009765625
0x36 = 0_0110110_ = +0b1.0×2^-9 = 0.001953125
0x37 = 0_0110111_ = +0b1.0×2^-8 = 0.00390625
0x38 = 0_0111000_ = +0b1.0×2^-7 = 0.0078125
0x39 = 0_0111001_ = +0b1.0×2^-6 = 0.015625
0x3a = 0_0111010_ = +0b1.0×2^-5 = 0.03125
0x3b = 0_0111011_ = +0b1.0×2^-4 = 0.0625
0x3c = 0_0111100_ = +0b1.0×2^-3 = 0.125
0x3d = 0_0111101_ = +0b1.0×2^-2 = 0.25
0x3e = 0_0111110_ = +0b1.0×2^-1 = 0.5
0x3f = 0_0111111_ = +0b1.0×2^0 = 1.0
0x40 = 0_1000000_ = +0b1.0×2^1 = 2.0
0x41 = 0_1000001_ = +0b1.0×2^2 = 4.0
0x42 = 0_1000010_ = +0b1.0×2^3 = 8.0
0x43 = 0_1000011_ = +0b1.0×2^4 = 16.0
0x44 = 0_1000100_ = +0b1.0×2^5 = 32.0
0x45 = 0_1000101_ = +0b1.0×2^6 = 64.0
0x46 = 0_1000110_ = +0b1.0×2^7 = 128.0
0x47 = 0_1000111_ = +0b1.0×2^8 = 256.0
0x48 = 0_1001000_ = +0b1.0×2^9 = 512.0
0x49 = 0_1001001_ = +0b1.0×2^10 = 1024.0
0x4a = 0_1001010_ = +0b1.0×2^11 = 2048.0
0x4b = 0_1001011_ = +0b1.0×2^12 = 4096.0
0x4c = 0_1001100_ = +0b1.0×2^13 = 8192.0
0x4d = 0_1001101_ = +0b1.0×2^14 = 16384.0
0x4e = 0_1001110_ = +0b1.0×2^15 = 32768.0
0x4f = 0_1001111_ = +0b1.0×2^16 = 65536.0
0x50 = 0_1010000_ = +0b1.0×2^17 = 131072.0
0x51 = 0_1010001_ = +0b1.0×2^18 = 262144.0
0x52 = 0_1010010_ = +0b1.0×2^19 = 524288.0
0x53 = 0_1010011_ = +0b1.0×2^20 = 1048576.0
0x54 = 0_1010100_ = +0b1.0×2^21 = 2097152.0
0x55 = 0_1010101_ = +0b1.0×2^22 = 4194304.0
0x56 = 0_1010110_ = +0b1.0×2^23 = 8388608.0
0x57 = 0_1010111_ = +0b1.0×2^24 = 16777216.0
0x58 = 0_1011000_ = +0b1.0×2^25 = 33554432.0
0x59 = 0_1011001_ = +0b1.0×2^26 = 67108864.0
0x5a = 0_1011010_ = +0b1.0×2^27 = 134217728.0
0x5b = 0_1011011_ = +0b1.0×2^28 = 268435456.0
0x5c = 0_1011100_ = +0b1.0×2^29 = 536870912.0
0x5d = 0_1011101_ = +0b1.0×2^30 = 1073741824.0
0x5e = 0_1011110_ = +0b1.0×2^31 = 2147483648.0
0x5f = 0_1011111_ = +0b1.0×2^32 = 4294967296.0
0x60 = 0_1100000_ = +0b1.0×2^33 = 8589934592.0
0x61 = 0_1100001_ = +0b1.0×2^34 = 17179869184.0
0x62 = 0_1100010_ = +0b1.0×2^35 = 34359738368.0
0x63 = 0_1100011_ = +0b1.0×2^36 = 68719476736.0
0x64 = 0_1100100_ = +0b1.0×2^37 = 137438953472.0
0x65 = 0_1100101_ = +0b1.0×2^38 = 274877906944.0
0x66 = 0_1100110_ = +0b1.0×2^39 = 549755813888.0
0x67 = 0_1100111_ = +0b1.0×2^40 ≈ 1.0995116e+12
0x68 = 0_1101000_ = +0b1.0×2^41 ≈ 2.1990233e+12
0x69 = 0_1101001_ = +0b1.0×2^42 ≈ 4.3980465e+12
0x6a = 0_1101010_ = +0b1.0×2^43 ≈ 8.796093e+12
0x6b = 0_1101011_ = +0b1.0×2^44 ≈ 1.7592186e+13
0x6c = 0_1101100_ = +0b1.0×2^45 ≈ 3.5184372e+13
0x6d = 0_1101101_ = +0b1.0×2^46 ≈ 7.0368744e+13
0x6e = 0_1101110_ = +0b1.0×2^47 ≈ 1.4073749e+14
0x6f = 0_1101111_ = +0b1.0×2^48 ≈ 2.8147498e+14
0x70 = 0_1110000_ = +0b1.0×2^49 ≈ 5.6294995e+14
0x71 = 0_1110001_ = +0b1.0×2^50 ≈ 1.1258999e+15
0x72 = 0_1110010_ = +0b1.0×2^51 ≈ 2.2517998e+15
0x73 = 0_1110011_ = +0b1.0×2^52 ≈ 4.5035996e+15
0x74 = 0_1110100_ = +0b1.0×2^53 ≈ 9.0071993e+15
0x75 = 0_1110101_ = +0b1.0×2^54 ≈ 1.8014399e+16
0x76 = 0_1110110_ = +0b1.0×2^55 ≈ 3.6028797e+16
0x77 = 0_1110111_ = +0b1.0×2^56 ≈ 7.2057594e+16
0x78 = 0_1111000_ = +0b1.0×2^57 ≈ 1.4411519e+17
0x79 = 0_1111001_ = +0b1.0×2^58 ≈ 2.8823038e+17
0x7a = 0_1111010_ = +0b1.0×2^59 ≈ 5.7646075e+17
0x7b = 0_1111011_ = +0b1.0×2^60 ≈ 1.1529215e+18
0x7c = 0_1111100_ = +0b1.0×2^61 ≈ 2.305843e+18
0x7d = 0_1111101_ = +0b1.0×2^62 ≈ 4.611686e+18
0x7e = 0_1111110_ = +0b1.0×2^63 ≈ 9.223372e+18
0x7f = 0_1111111_ = +Inf
0x80 = 1_0000000_ = NaN
0x81 = 1_0000001_ = −0b1.0×2^-62 ≈ −2.1684043E-19
0x82 = 1_0000010_ = −0b1.0×2^-61 ≈ −4.3368087E-19
0x83 = 1_0000011_ = −0b1.0×2^-60 ≈ −8.6736174E-19
0x84 = 1_0000100_ = −0b1.0×2^-59 ≈ −1.7347235E-18
0x85 = 1_0000101_ = −0b1.0×2^-58 ≈ −3.469447E-18
0x86 = 1_0000110_ = −0b1.0×2^-57 ≈ −6.9388939E-18
0x87 = 1_0000111_ = −0b1.0×2^-56 ≈ −1.3877788E-17
0x88 = 1_0001000_ = −0b1.0×2^-55 ≈ −2.7755576E-17
0x89 = 1_0001001_ = −0b1.0×2^-54 ≈ −5.5511151E-17
0x8a = 1_0001010_ = −0b1.0×2^-53 ≈ −1.110223E-16
0x8b = 1_0001011_ = −0b1.0×2^-52 ≈ −2.220446E-16
0x8c = 1_0001100_ = −0b1.0×2^-51 ≈ −4.4408921E-16
0x8d = 1_0001101_ = −0b1.0×2^-50 ≈ −8.8817842E-16
0x8e = 1_0001110_ = −0b1.0×2^-49 ≈ −1.7763568E-15
0x8f = 1_0001111_ = −0b1.0×2^-48 ≈ −3.5527137E-15
0x90 = 1_0010000_ = −0b1.0×2^-47 ≈ −7.1054274E-15
0x91 = 1_0010001_ = −0b1.0×2^-46 ≈ −1.4210855E-14
0x92 = 1_0010010_ = −0b1.0×2^-45 ≈ −2.8421709E-14
0x93 = 1_0010011_ = −0b1.0×2^-44 ≈ −5.6843419E-14
0x94 = 1_0010100_ = −0b1.0×2^-43 ≈ −1.1368684E-13
0x95 = 1_0010101_ = −0b1.0×2^-42 ≈ −2.2737368E-13
0x96 = 1_0010110_ = −0b1.0×2^-41 ≈ −4.5474735E-13
0x97 = 1_0010111_ = −0b1.0×2^-40 ≈ −9.094947E-13
0x98 = 1_0011000_ = −0b1.0×2^-39 ≈ −1.8189894E-12
0x99 = 1_0011001_ = −0b1.0×2^-38 ≈ −3.6379788E-12
0x9a = 1_0011010_ = −0b1.0×2^-37 ≈ −7.2759576E-12
0x9b = 1_0011011_ = −0b1.0×2^-36 ≈ −1.4551915E-11
0x9c = 1_0011100_ = −0b1.0×2^-35 ≈ −2.910383E-11
0x9d = 1_0011101_ = −0b1.0×2^-34 ≈ −5.8207661E-11
0x9e = 1_0011110_ = −0b1.0×2^-33 ≈ −1.1641532E-10
0x9f = 1_0011111_ = −0b1.0×2^-32 ≈ −2.3283064E-10
0xa0 = 1_0100000_ = −0b1.0×2^-31 ≈ −4.6566129E-10
0xa1 = 1_0100001_ = −0b1.0×2^-30 ≈ −9.3132257E-10
0xa2 = 1_0100010_ = −0b1.0×2^-29 ≈ −1.8626451E-09
0xa3 = 1_0100011_ = −0b1.0×2^-28 ≈ −3.7252903E-09
0xa4 = 1_0100100_ = −0b1.0×2^-27 ≈ −7.4505806E-09
0xa5 = 1_0100101_ = −0b1.0×2^-26 ≈ −1.4901161E-08
0xa6 = 1_0100110_ = −0b1.0×2^-25 ≈ −2.9802322E-08
0xa7 = 1_0100111_ = −0b1.0×2^-24 ≈ −5.9604645E-08
0xa8 = 1_0101000_ = −0b1.0×2^-23 ≈ −1.1920929E-07
0xa9 = 1_0101001_ = −0b1.0×2^-22 ≈ −2.3841858E-07
0xaa = 1_0101010_ = −0b1.0×2^-21 ≈ −4.7683716E-07
0xab = 1_0101011_ = −0b1.0×2^-20 ≈ −9.5367432E-07
0xac = 1_0101100_ = −0b1.0×2^-19 ≈ −1.9073486E-06
0xad = 1_0101101_ = −0b1.0×2^-18 ≈ −3.8146973E-06
0xae = 1_0101110_ = −0b1.0×2^-17 ≈ −7.6293945E-06
0xaf = 1_0101111_ = −0b1.0×2^-16 ≈ −1.5258789E-05
0xb0 = 1_0110000_ = −0b1.0×2^-15 ≈ −3.0517578E-05
0xb1 = 1_0110001_ = −0b1.0×2^-14 ≈ −6.1035156E-05
0xb2 = 1_0110010_ = −0b1.0×2^-13 ≈ -0.00012207031
0xb3 = 1_0110011_ = −0b1.0×2^-12 ≈ -0.00024414062
0xb4 = 1_0110100_ = −0b1.0×2^-11 = -0.00048828125
0xb5 = 1_0110101_ = −0b1.0×2^-10 = -0.0009765625
0xb6 = 1_0110110_ = −0b1.0×2^-9 = -0.001953125
0xb7 = 1_0110111_ = −0b1.0×2^-8 = -0.00390625
0xb8 = 1_0111000_ = −0b1.0×2^-7 = -0.0078125
0xb9 = 1_0111001_ = −0b1.0×2^-6 = -0.015625
0xba = 1_0111010_ = −0b1.0×2^-5 = -0.03125
0xbb = 1_0111011_ = −0b1.0×2^-4 = -0.0625
0xbc = 1_0111100_ = −0b1.0×2^-3 = -0.125
0xbd = 1_0111101_ = −0b1.0×2^-2 = -0.25
0xbe = 1_0111110_ = −0b1.0×2^-1 = -0.5
0xbf = 1_0111111_ = −0b1.0×2^0 = -1.0
0xc0 = 1_1000000_ = −0b1.0×2^1 = -2.0
0xc1 = 1_1000001_ = −0b1.0×2^2 = -4.0
0xc2 = 1_1000010_ = −0b1.0×2^3 = -8.0
0xc3 = 1_1000011_ = −0b1.0×2^4 = -16.0
0xc4 = 1_1000100_ = −0b1.0×2^5 = -32.0
0xc5 = 1_1000101_ = −0b1.0×2^6 = -64.0
0xc6 = 1_1000110_ = −0b1.0×2^7 = -128.0
0xc7 = 1_1000111_ = −0b1.0×2^8 = -256.0
0xc8 = 1_1001000_ = −0b1.0×2^9 = -512.0
0xc9 = 1_1001001_ = −0b1.0×2^10 = -1024.0
0xca = 1_1001010_ = −0b1.0×2^11 = -2048.0
0xcb = 1_1001011_ = −0b1.0×2^12 = -4096.0
0xcc = 1_1001100_ = −0b1.0×2^13 = -8192.0
0xcd = 1_1001101_ = −0b1.0×2^14 = -16384.0
0xce = 1_1001110_ = −0b1.0×2^15 = -32768.0
0xcf = 1_1001111_ = −0b1.0×2^16 = -65536.0
0xd0 = 1_1010000_ = −0b1.0×2^17 = -131072.0
0xd1 = 1_1010001_ = −0b1.0×2^18 = -262144.0
0xd2 = 1_1010010_ = −0b1.0×2^19 = -524288.0
0xd3 = 1_1010011_ = −0b1.0×2^20 = -1048576.0
0xd4 = 1_1010100_ = −0b1.0×2^21 = -2097152.0
0xd5 = 1_1010101_ = −0b1.0×2^22 = -4194304.0
0xd6 = 1_1010110_ = −0b1.0×2^23 = -8388608.0
0xd7 = 1_1010111_ = −0b1.0×2^24 = -16777216.0
0xd8 = 1_1011000_ = −0b1.0×2^25 = -33554432.0
0xd9 = 1_1011001_ = −0b1.0×2^26 = -67108864.0
0xda = 1_1011010_ = −0b1.0×2^27 = -134217728.0
0xdb = 1_1011011_ = −0b1.0×2^28 = -268435456.0
0xdc = 1_1011100_ = −0b1.0×2^29 = -536870912.0
0xdd = 1_1011101_ = −0b1.0×2^30 = -1073741824.0
0xde = 1_1011110_ = −0b1.0×2^31 = -2147483648.0
0xdf = 1_1011111_ = −0b1.0×2^32 = -4294967296.0
0xe0 = 1_1100000_ = −0b1.0×2^33 = -8589934592.0
0xe1 = 1_1100001_ = −0b1.0×2^34 = -17179869184.0
0xe2 = 1_1100010_ = −0b1.0×2^35 = -34359738368.0
0xe3 = 1_1100011_ = −0b1.0×2^36 = -68719476736.0
0xe4 = 1_1100100_ = −0b1.0×2^37 ≈ −1.3743895e+11
0xe5 = 1_1100101_ = −0b1.0×2^38 ≈ −2.7487791e+11
0xe6 = 1_1100110_ = −0b1.0×2^39 ≈ −5.4975581e+11
0xe7 = 1_1100111_ = −0b1.0×2^40 ≈ −1.0995116e+12
0xe8 = 1_1101000_ = −0b1.0×2^41 ≈ −2.1990233e+12
0xe9 = 1_1101001_ = −0b1.0×2^42 ≈ −4.3980465e+12
0xea = 1_1101010_ = −0b1.0×2^43 ≈ −8.796093e+12
0xeb = 1_1101011_ = −0b1.0×2^44 ≈ −1.7592186e+13
0xec = 1_1101100_ = −0b1.0×2^45 ≈ −3.5184372e+13
0xed = 1_1101101_ = −0b1.0×2^46 ≈ −7.0368744e+13
0xee = 1_1101110_ = −0b1.0×2^47 ≈ −1.4073749e+14
0xef = 1_1101111_ = −0b1.0×2^48 ≈ −2.8147498e+14
0xf0 = 1_1110000_ = −0b1.0×2^49 ≈ −5.6294995e+14
0xf1 = 1_1110001_ = −0b1.0×2^50 ≈ −1.1258999e+15
0xf2 = 1_1110010_ = −0b1.0×2^51 ≈ −2.2517998e+15
0xf3 = 1_1110011_ = −0b1.0×2^52 ≈ −4.5035996e+15
0xf4 = 1_1110100_ = −0b1.0×2^53 ≈ −9.0071993e+15
0xf5 = 1_1110101_ = −0b1.0×2^54 ≈ −1.8014399e+16
0xf6 = 1_1110110_ = −0b1.0×2^55 ≈ −3.6028797e+16
0xf7 = 1_1110111_ = −0b1.0×2^56 ≈ −7.2057594e+16
0xf8 = 1_1111000_ = −0b1.0×2^57 ≈ −1.4411519e+17
0xf9 = 1_1111001_ = −0b1.0×2^58 ≈ −2.8823038e+17
0xfa = 1_1111010_ = −0b1.0×2^59 ≈ −5.7646075e+17
0xfb = 1_1111011_ = −0b1.0×2^60 ≈ −1.1529215e+18
0xfc = 1_1111100_ = −0b1.0×2^61 ≈ −2.305843e+18
0xfd = 1_1111101_ = −0b1.0×2^62 ≈ −4.611686e+18
0xfe = 1_1111110_ = −0b1.0×2^63 ≈ −9.223372e+18
0xff = 1_1111111_ = −Inf

C.2 Value Table: P2, emin $= -31$, emax $= 31$

0x00 = 0_000000_0 = 0.0
0x01 = 0_000000_1 = +0b0.1×2^-31 ≈ 2.3283064E-10
0x02 = 0_000001_0 = +0b1.0×2^-31 ≈ 4.6566129E-10
0x03 = 0_000001_1 = +0b1.1×2^-31 ≈ 6.9849193E-10
0x04 = 0_000010_0 = +0b1.0×2^-30 ≈ 9.3132257E-10
0x05 = 0_000010_1 = +0b1.1×2^-30 ≈ 1.3969839E-09
0x06 = 0_000011_0 = +0b1.0×2^-29 ≈ 1.8626451E-09
0x07 = 0_000011_1 = +0b1.1×2^-29 ≈ 2.7939677E-09
0x08 = 0_000100_0 = +0b1.0×2^-28 ≈ 3.7252903E-09
0x09 = 0_000100_1 = +0b1.1×2^-28 ≈ 5.5879354E-09
0x0a = 0_000101_0 = +0b1.0×2^-27 ≈ 7.4505806E-09
0x0b = 0_000101_1 = +0b1.1×2^-27 ≈ 1.1175871E-08
0x0c = 0_000110_0 = +0b1.0×2^-26 ≈ 1.4901161E-08
0x0d = 0_000110_1 = +0b1.1×2^-26 ≈ 2.2351742E-08
0x0e = 0_000111_0 = +0b1.0×2^-25 ≈ 2.9802322E-08
0x0f = 0_000111_1 = +0b1.1×2^-25 ≈ 4.4703484E-08
0x10 = 0_001000_0 = +0b1.0×2^-24 ≈ 5.9604645E-08
0x11 = 0_001000_1 = +0b1.1×2^-24 ≈ 8.9406967E-08
0x12 = 0_001001_0 = +0b1.0×2^-23 ≈ 1.1920929E-07
0x13 = 0_001001_1 = +0b1.1×2^-23 ≈ 1.7881393E-07
0x14 = 0_001010_0 = +0b1.0×2^-22 ≈ 2.3841858E-07
0x15 = 0_001010_1 = +0b1.1×2^-22 ≈ 3.5762787E-07
0x16 = 0_001011_0 = +0b1.0×2^-21 ≈ 4.7683716E-07
0x17 = 0_001011_1 = +0b1.1×2^-21 ≈ 7.1525574E-07
0x18 = 0_001100_0 = +0b1.0×2^-20 ≈ 9.5367432E-07
0x19 = 0_001100_1 = +0b1.1×2^-20 ≈ 1.4305115E-06
0x1a = 0_001101_0 = +0b1.0×2^-19 ≈ 1.9073486E-06
0x1b = 0_001101_1 = +0b1.1×2^-19 ≈ 2.8610229E-06
0x1c = 0_001110_0 = +0b1.0×2^-18 ≈ 3.8146973E-06
0x1d = 0_001110_1 = +0b1.1×2^-18 ≈ 5.7220459E-06
0x1e = 0_001111_0 = +0b1.0×2^-17 ≈ 7.6293945E-06
0x1f = 0_001111_1 = +0b1.1×2^-17 ≈ 1.1444092E-05
0x20 = 0_010000_0 = +0b1.0×2^-16 ≈ 1.5258789E-05
0x21 = 0_010000_1 = +0b1.1×2^-16 ≈ 2.2888184E-05
0x22 = 0_010001_0 = +0b1.0×2^-15 ≈ 3.0517578E-05
0x23 = 0_010001_1 = +0b1.1×2^-15 ≈ 4.5776367E-05
0x24 = 0_010010_0 = +0b1.0×2^-14 ≈ 6.1035156E-05
0x25 = 0_010010_1 = +0b1.1×2^-14 ≈ 9.1552734E-05
0x26 = 0_010011_0 = +0b1.0×2^-13 ≈ 0.00012207031
0x27 = 0_010011_1 = +0b1.1×2^-13 ≈ 0.00018310547
0x28 = 0_010100_0 = +0b1.0×2^-12 = 0.000244140625
0x29 = 0_010100_1 = +0b1.1×2^-12 ≈ 0.00036621094
0x2a = 0_010101_0 = +0b1.0×2^-11 = 0.00048828125
0x2b = 0_010101_1 = +0b1.1×2^-11 = 0.000732421875
0x2c = 0_010110_0 = +0b1.0×2^-10 = 0.0009765625
0x2d = 0_010110_1 = +0b1.1×2^-10 = 0.00146484375
0x2e = 0_010111_0 = +0b1.0×2^-9 = 0.001953125
0x2f = 0_010111_1 = +0b1.1×2^-9 = 0.0029296875
0x30 = 0_011000_0 = +0b1.0×2^-8 = 0.00390625
0x31 = 0_011000_1 = +0b1.1×2^-8 = 0.005859375
0x32 = 0_011001_0 = +0b1.0×2^-7 = 0.0078125
0x33 = 0_011001_1 = +0b1.1×2^-7 = 0.01171875
0x34 = 0_011010_0 = +0b1.0×2^-6 = 0.015625
0x35 = 0_011010_1 = +0b1.1×2^-6 = 0.0234375
0x36 = 0_011011_0 = +0b1.0×2^-5 = 0.03125
0x37 = 0_011011_1 = +0b1.1×2^-5 = 0.046875
0x38 = 0_011100_0 = +0b1.0×2^-4 = 0.0625
0x39 = 0_011100_1 = +0b1.1×2^-4 = 0.09375
0x3a = 0_011101_0 = +0b1.0×2^-3 = 0.125
0x3b = 0_011101_1 = +0b1.1×2^-3 = 0.1875
0x3c = 0_011110_0 = +0b1.0×2^-2 = 0.25
0x3d = 0_011110_1 = +0b1.1×2^-2 = 0.375
0x3e = 0_011111_0 = +0b1.0×2^-1 = 0.5
0x3f = 0_011111_1 = +0b1.1×2^-1 = 0.75
0x40 = 0_100000_0 = +0b1.0×2^0 = 1.0
0x41 = 0_100000_1 = +0b1.1×2^0 = 1.5
0x42 = 0_100001_0 = +0b1.0×2^1 = 2.0
0x43 = 0_100001_1 = +0b1.1×2^1 = 3.0
0x44 = 0_100010_0 = +0b1.0×2^2 = 4.0
0x45 = 0_100010_1 = +0b1.1×2^2 = 6.0
0x46 = 0_100011_0 = +0b1.0×2^3 = 8.0
0x47 = 0_100011_1 = +0b1.1×2^3 = 12.0
0x48 = 0_100100_0 = +0b1.0×2^4 = 16.0
0x49 = 0_100100_1 = +0b1.1×2^4 = 24.0
0x4a = 0_100101_0 = +0b1.0×2^5 = 32.0
0x4b = 0_100101_1 = +0b1.1×2^5 = 48.0
0x4c = 0_100110_0 = +0b1.0×2^6 = 64.0
0x4d = 0_100110_1 = +0b1.1×2^6 = 96.0
0x4e = 0_100111_0 = +0b1.0×2^7 = 128.0
0x4f = 0_100111_1 = +0b1.1×2^7 = 192.0
0x50 = 0_101000_0 = +0b1.0×2^8 = 256.0
0x51 = 0_101000_1 = +0b1.1×2^8 = 384.0
0x52 = 0_101001_0 = +0b1.0×2^9 = 512.0
0x53 = 0_101001_1 = +0b1.1×2^9 = 768.0
0x54 = 0_101010_0 = +0b1.0×2^10 = 1024.0
0x55 = 0_101010_1 = +0b1.1×2^10 = 1536.0
0x56 = 0_101011_0 = +0b1.0×2^11 = 2048.0
0x57 = 0_101011_1 = +0b1.1×2^11 = 3072.0
0x58 = 0_101100_0 = +0b1.0×2^12 = 4096.0
0x59 = 0_101100_1 = +0b1.1×2^12 = 6144.0
0x5a = 0_101101_0 = +0b1.0×2^13 = 8192.0
0x5b = 0_101101_1 = +0b1.1×2^13 = 12288.0
0x5c = 0_101110_0 = +0b1.0×2^14 = 16384.0
0x5d = 0_101110_1 = +0b1.1×2^14 = 24576.0
0x5e = 0_101111_0 = +0b1.0×2^15 = 32768.0
0x5f = 0_101111_1 = +0b1.1×2^15 = 49152.0
0x60 = 0_110000_0 = +0b1.0×2^16 = 65536.0
0x61 = 0_110000_1 = +0b1.1×2^16 = 98304.0
0x62 = 0_110001_0 = +0b1.0×2^17 = 131072.0
0x63 = 0_110001_1 = +0b1.1×2^17 = 196608.0
0x64 = 0_110010_0 = +0b1.0×2^18 = 262144.0
0x65 = 0_110010_1 = +0b1.1×2^18 = 393216.0
0x66 = 0_110011_0 = +0b1.0×2^19 = 524288.0
0x67 = 0_110011_1 = +0b1.1×2^19 = 786432.0
0x68 = 0_110100_0 = +0b1.0×2^20 = 1048576.0
0x69 = 0_110100_1 = +0b1.1×2^20 = 1572864.0
0x6a = 0_110101_0 = +0b1.0×2^21 = 2097152.0
0x6b = 0_110101_1 = +0b1.1×2^21 = 3145728.0
0x6c = 0_110110_0 = +0b1.0×2^22 = 4194304.0
0x6d = 0_110110_1 = +0b1.1×2^22 = 6291456.0
0x6e = 0_110111_0 = +0b1.0×2^23 = 8388608.0
0x6f = 0_110111_1 = +0b1.1×2^23 = 12582912.0
0x70 = 0_111000_0 = +0b1.0×2^24 = 16777216.0
0x71 = 0_111000_1 = +0b1.1×2^24 = 25165824.0
0x72 = 0_111001_0 = +0b1.0×2^25 = 33554432.0
0x73 = 0_111001_1 = +0b1.1×2^25 = 50331648.0
0x74 = 0_111010_0 = +0b1.0×2^26 = 67108864.0
0x75 = 0_111010_1 = +0b1.1×2^26 = 100663296.0
0x76 = 0_111011_0 = +0b1.0×2^27 = 134217728.0
0x77 = 0_111011_1 = +0b1.1×2^27 = 201326592.0
0x78 = 0_111100_0 = +0b1.0×2^28 = 268435456.0
0x79 = 0_111100_1 = +0b1.1×2^28 = 402653184.0
0x7a = 0_111101_0 = +0b1.0×2^29 = 536870912.0
0x7b = 0_111101_1 = +0b1.1×2^29 = 805306368.0
0x7c = 0_111110_0 = +0b1.0×2^30 = 1073741824.0
0x7d = 0_111110_1 = +0b1.1×2^30 = 1610612736.0
0x7e = 0_111111_0 = +0b1.0×2^31 = 2147483648.0
0x7f = 0_111111_1 = +Inf
0x80 = 1_000000_0 = NaN
0x81 = 1_000000_1 = −0b0.1×2^-31 ≈ −2.3283064E-10
0x82 = 1_000001_0 = −0b1.0×2^-31 ≈ −4.6566129E-10
0x83 = 1_000001_1 = −0b1.1×2^-31 ≈ −6.9849193E-10
0x84 = 1_000010_0 = −0b1.0×2^-30 ≈ −9.3132257E-10
0x85 = 1_000010_1 = −0b1.1×2^-30 ≈ −1.3969839E-09
0x86 = 1_000011_0 = −0b1.0×2^-29 ≈ −1.8626451E-09
0x87 = 1_000011_1 = −0b1.1×2^-29 ≈ −2.7939677E-09
0x88 = 1_000100_0 = −0b1.0×2^-28 ≈ −3.7252903E-09
0x89 = 1_000100_1 = −0b1.1×2^-28 ≈ −5.5879354E-09
0x8a = 1_000101_0 = −0b1.0×2^-27 ≈ −7.4505806E-09
0x8b = 1_000101_1 = −0b1.1×2^-27 ≈ −1.1175871E-08
0x8c = 1_000110_0 = −0b1.0×2^-26 ≈ −1.4901161E-08
0x8d = 1_000110_1 = −0b1.1×2^-26 ≈ −2.2351742E-08
0x8e = 1_000111_0 = −0b1.0×2^-25 ≈ −2.9802322E-08
0x8f = 1_000111_1 = −0b1.1×2^-25 ≈ −4.4703484E-08
0x90 = 1_001000_0 = −0b1.0×2^-24 ≈ −5.9604645E-08
0x91 = 1_001000_1 = −0b1.1×2^-24 ≈ −8.9406967E-08
0x92 = 1_001001_0 = −0b1.0×2^-23 ≈ −1.1920929E-07
0x93 = 1_001001_1 = −0b1.1×2^-23 ≈ −1.7881393E-07
0x94 = 1_001010_0 = −0b1.0×2^-22 ≈ −2.3841858E-07
0x95 = 1_001010_1 = −0b1.1×2^-22 ≈ −3.5762787E-07
0x96 = 1_001011_0 = −0b1.0×2^-21 ≈ −4.7683716E-07
0x97 = 1_001011_1 = −0b1.1×2^-21 ≈ −7.1525574E-07
0x98 = 1_001100_0 = −0b1.0×2^-20 ≈ −9.5367432E-07
0x99 = 1_001100_1 = −0b1.1×2^-20 ≈ −1.4305115E-06
0x9a = 1_001101_0 = −0b1.0×2^-19 ≈ −1.9073486E-06
0x9b = 1_001101_1 = −0b1.1×2^-19 ≈ −2.8610229E-06
0x9c = 1_001110_0 = −0b1.0×2^-18 ≈ −3.8146973E-06
0x9d = 1_001110_1 = −0b1.1×2^-18 ≈ −5.7220459E-06
0x9e = 1_001111_0 = −0b1.0×2^-17 ≈ −7.6293945E-06
0x9f = 1_001111_1 = −0b1.1×2^-17 ≈ −1.1444092E-05
0xa0 = 1_010000_0 = −0b1.0×2^-16 ≈ −1.5258789E-05
0xa1 = 1_010000_1 = −0b1.1×2^-16 ≈ −2.2888184E-05
0xa2 = 1_010001_0 = −0b1.0×2^-15 ≈ −3.0517578E-05
0xa3 = 1_010001_1 = −0b1.1×2^-15 ≈ −4.5776367E-05
0xa4 = 1_010010_0 = −0b1.0×2^-14 ≈ −6.1035156E-05
0xa5 = 1_010010_1 = −0b1.1×2^-14 ≈ −9.1552734E-05
0xa6 = 1_010011_0 = −0b1.0×2^-13 ≈ -0.00012207031
0xa7 = 1_010011_1 = −0b1.1×2^-13 ≈ -0.00018310547
0xa8 = 1_010100_0 = −0b1.0×2^-12 ≈ -0.00024414062
0xa9 = 1_010100_1 = −0b1.1×2^-12 ≈ -0.00036621094
0xaa = 1_010101_0 = −0b1.0×2^-11 = -0.00048828125
0xab = 1_010101_1 = −0b1.1×2^-11 ≈ -0.00073242188
0xac = 1_010110_0 = −0b1.0×2^-10 = -0.0009765625
0xad = 1_010110_1 = −0b1.1×2^-10 = -0.00146484375
0xae = 1_010111_0 = −0b1.0×2^-9 = -0.001953125
0xaf = 1_010111_1 = −0b1.1×2^-9 = -0.0029296875
0xb0 = 1_011000_0 = −0b1.0×2^-8 = -0.00390625
0xb1 = 1_011000_1 = −0b1.1×2^-8 = -0.005859375
0xb2 = 1_011001_0 = −0b1.0×2^-7 = -0.0078125
0xb3 = 1_011001_1 = −0b1.1×2^-7 = -0.01171875
0xb4 = 1_011010_0 = −0b1.0×2^-6 = -0.015625
0xb5 = 1_011010_1 = −0b1.1×2^-6 = -0.0234375
0xb6 = 1_011011_0 = −0b1.0×2^-5 = -0.03125
0xb7 = 1_011011_1 = −0b1.1×2^-5 = -0.046875
0xb8 = 1_011100_0 = −0b1.0×2^-4 = -0.0625
0xb9 = 1_011100_1 = −0b1.1×2^-4 = -0.09375
0xba = 1_011101_0 = −0b1.0×2^-3 = -0.125
0xbb = 1_011101_1 = −0b1.1×2^-3 = -0.1875
0xbc = 1_011110_0 = −0b1.0×2^-2 = -0.25
0xbd = 1_011110_1 = −0b1.1×2^-2 = -0.375
0xbe = 1_011111_0 = −0b1.0×2^-1 = -0.5
0xbf = 1_011111_1 = −0b1.1×2^-1 = -0.75
0xc0 = 1_100000_0 = −0b1.0×2^0 = -1.0
0xc1 = 1_100000_1 = −0b1.1×2^0 = -1.5
0xc2 = 1_100001_0 = −0b1.0×2^1 = -2.0
0xc3 = 1_100001_1 = −0b1.1×2^1 = -3.0
0xc4 = 1_100010_0 = −0b1.0×2^2 = -4.0
0xc5 = 1_100010_1 = −0b1.1×2^2 = -6.0
0xc6 = 1_100011_0 = −0b1.0×2^3 = -8.0
0xc7 = 1_100011_1 = −0b1.1×2^3 = -12.0
0xc8 = 1_100100_0 = −0b1.0×2^4 = -16.0
0xc9 = 1_100100_1 = −0b1.1×2^4 = -24.0
0xca = 1_100101_0 = −0b1.0×2^5 = -32.0
0xcb = 1_100101_1 = −0b1.1×2^5 = -48.0
0xcc = 1_100110_0 = −0b1.0×2^6 = -64.0
0xcd = 1_100110_1 = −0b1.1×2^6 = -96.0
0xce = 1_100111_0 = −0b1.0×2^7 = -128.0
0xcf = 1_100111_1 = −0b1.1×2^7 = -192.0
0xd0 = 1_101000_0 = −0b1.0×2^8 = -256.0
0xd1 = 1_101000_1 = −0b1.1×2^8 = -384.0
0xd2 = 1_101001_0 = −0b1.0×2^9 = -512.0
0xd3 = 1_101001_1 = −0b1.1×2^9 = -768.0
0xd4 = 1_101010_0 = −0b1.0×2^10 = -1024.0
0xd5 = 1_101010_1 = −0b1.1×2^10 = -1536.0
0xd6 = 1_101011_0 = −0b1.0×2^11 = -2048.0
0xd7 = 1_101011_1 = −0b1.1×2^11 = -3072.0
0xd8 = 1_101100_0 = −0b1.0×2^12 = -4096.0
0xd9 = 1_101100_1 = −0b1.1×2^12 = -6144.0
0xda = 1_101101_0 = −0b1.0×2^13 = -8192.0
0xdb = 1_101101_1 = −0b1.1×2^13 = -12288.0
0xdc = 1_101110_0 = −0b1.0×2^14 = -16384.0
0xdd = 1_101110_1 = −0b1.1×2^14 = -24576.0
0xde = 1_101111_0 = −0b1.0×2^15 = -32768.0
0xdf = 1_101111_1 = −0b1.1×2^15 = -49152.0
0xe0 = 1_110000_0 = −0b1.0×2^16 = -65536.0
0xe1 = 1_110000_1 = −0b1.1×2^16 = -98304.0
0xe2 = 1_110001_0 = −0b1.0×2^17 = -131072.0
0xe3 = 1_110001_1 = −0b1.1×2^17 = -196608.0
0xe4 = 1_110010_0 = −0b1.0×2^18 = -262144.0
0xe5 = 1_110010_1 = −0b1.1×2^18 = -393216.0
0xe6 = 1_110011_0 = −0b1.0×2^19 = -524288.0
0xe7 = 1_110011_1 = −0b1.1×2^19 = -786432.0
0xe8 = 1_110100_0 = −0b1.0×2^20 = -1048576.0
0xe9 = 1_110100_1 = −0b1.1×2^20 = -1572864.0
0xea = 1_110101_0 = −0b1.0×2^21 = -2097152.0
0xeb = 1_110101_1 = −0b1.1×2^21 = -3145728.0
0xec = 1_110110_0 = −0b1.0×2^22 = -4194304.0
0xed = 1_110110_1 = −0b1.1×2^22 = -6291456.0
0xee = 1_110111_0 = −0b1.0×2^23 = -8388608.0
0xef = 1_110111_1 = −0b1.1×2^23 = -12582912.0
0xf0 = 1_111000_0 = −0b1.0×2^24 = -16777216.0
0xf1 = 1_111000_1 = −0b1.1×2^24 = -25165824.0
0xf2 = 1_111001_0 = −0b1.0×2^25 = -33554432.0
0xf3 = 1_111001_1 = −0b1.1×2^25 = -50331648.0
0xf4 = 1_111010_0 = −0b1.0×2^26 = -67108864.0
0xf5 = 1_111010_1 = −0b1.1×2^26 = -100663296.0
0xf6 = 1_111011_0 = −0b1.0×2^27 = -134217728.0
0xf7 = 1_111011_1 = −0b1.1×2^27 = -201326592.0
0xf8 = 1_111100_0 = −0b1.0×2^28 = -268435456.0
0xf9 = 1_111100_1 = −0b1.1×2^28 = -402653184.0
0xfa = 1_111101_0 = −0b1.0×2^29 = -536870912.0
0xfb = 1_111101_1 = −0b1.1×2^29 = -805306368.0
0xfc = 1_111110_0 = −0b1.0×2^30 = -1073741824.0
0xfd = 1_111110_1 = −0b1.1×2^30 = -1610612736.0
0xfe = 1_111111_0 = −0b1.0×2^31 = -2147483648.0
0xff = 1_111111_1 = −Inf

C.3 Value Table: P3, emin $= -15$, emax $= 15$

0x00 = 0_00000_00 = 0.0
0x01 = 0_00000_01 = +0b0.01×2^-15 ≈ 7.6293945E-06
0x02 = 0_00000_10 = +0b0.10×2^-15 ≈ 1.5258789E-05
0x03 = 0_00000_11 = +0b0.11×2^-15 ≈ 2.2888184E-05
0x04 = 0_00001_00 = +0b1.00×2^-15 ≈ 3.0517578E-05
0x05 = 0_00001_01 = +0b1.01×2^-15 ≈ 3.8146973E-05
0x06 = 0_00001_10 = +0b1.10×2^-15 ≈ 4.5776367E-05
0x07 = 0_00001_11 = +0b1.11×2^-15 ≈ 5.3405762E-05
0x08 = 0_00010_00 = +0b1.00×2^-14 ≈ 6.1035156E-05
0x09 = 0_00010_01 = +0b1.01×2^-14 ≈ 7.6293945E-05
0x0a = 0_00010_10 = +0b1.10×2^-14 ≈ 9.1552734E-05
0x0b = 0_00010_11 = +0b1.11×2^-14 ≈ 0.00010681152
0x0c = 0_00011_00 = +0b1.00×2^-13 ≈ 0.00012207031
0x0d = 0_00011_01 = +0b1.01×2^-13 ≈ 0.00015258789
0x0e = 0_00011_10 = +0b1.10×2^-13 ≈ 0.00018310547
0x0f = 0_00011_11 = +0b1.11×2^-13 ≈ 0.00021362305
0x10 = 0_00100_00 = +0b1.00×2^-12 = 0.000244140625
0x11 = 0_00100_01 = +0b1.01×2^-12 ≈ 0.00030517578
0x12 = 0_00100_10 = +0b1.10×2^-12 ≈ 0.00036621094
0x13 = 0_00100_11 = +0b1.11×2^-12 ≈ 0.00042724609
0x14 = 0_00101_00 = +0b1.00×2^-11 = 0.00048828125
0x15 = 0_00101_01 = +0b1.01×2^-11 ≈ 0.00061035156
0x16 = 0_00101_10 = +0b1.10×2^-11 = 0.000732421875
0x17 = 0_00101_11 = +0b1.11×2^-11 ≈ 0.00085449219
0x18 = 0_00110_00 = +0b1.00×2^-10 = 0.0009765625
0x19 = 0_00110_01 = +0b1.01×2^-10 = 0.001220703125
0x1a = 0_00110_10 = +0b1.10×2^-10 = 0.00146484375
0x1b = 0_00110_11 = +0b1.11×2^-10 = 0.001708984375
0x1c = 0_00111_00 = +0b1.00×2^-9 = 0.001953125
0x1d = 0_00111_01 = +0b1.01×2^-9 = 0.00244140625
0x1e = 0_00111_10 = +0b1.10×2^-9 = 0.0029296875
0x1f = 0_00111_11 = +0b1.11×2^-9 = 0.00341796875
0x20 = 0_01000_00 = +0b1.00×2^-8 = 0.00390625
0x21 = 0_01000_01 = +0b1.01×2^-8 = 0.0048828125
0x22 = 0_01000_10 = +0b1.10×2^-8 = 0.005859375
0x23 = 0_01000_11 = +0b1.11×2^-8 = 0.0068359375
0x24 = 0_01001_00 = +0b1.00×2^-7 = 0.0078125
0x25 = 0_01001_01 = +0b1.01×2^-7 = 0.009765625
0x26 = 0_01001_10 = +0b1.10×2^-7 = 0.01171875
0x27 = 0_01001_11 = +0b1.11×2^-7 = 0.013671875
0x28 = 0_01010_00 = +0b1.00×2^-6 = 0.015625
0x29 = 0_01010_01 = +0b1.01×2^-6 = 0.01953125
0x2a = 0_01010_10 = +0b1.10×2^-6 = 0.0234375
0x2b = 0_01010_11 = +0b1.11×2^-6 = 0.02734375
0x2c = 0_01011_00 = +0b1.00×2^-5 = 0.03125
0x2d = 0_01011_01 = +0b1.01×2^-5 = 0.0390625
0x2e = 0_01011_10 = +0b1.10×2^-5 = 0.046875
0x2f = 0_01011_11 = +0b1.11×2^-5 = 0.0546875
0x30 = 0_01100_00 = +0b1.00×2^-4 = 0.0625
0x31 = 0_01100_01 = +0b1.01×2^-4 = 0.078125
0x32 = 0_01100_10 = +0b1.10×2^-4 = 0.09375
0x33 = 0_01100_11 = +0b1.11×2^-4 = 0.109375
0x34 = 0_01101_00 = +0b1.00×2^-3 = 0.125
0x35 = 0_01101_01 = +0b1.01×2^-3 = 0.15625
0x36 = 0_01101_10 = +0b1.10×2^-3 = 0.1875
0x37 = 0_01101_11 = +0b1.11×2^-3 = 0.21875
0x38 = 0_01110_00 = +0b1.00×2^-2 = 0.25
0x39 = 0_01110_01 = +0b1.01×2^-2 = 0.3125
0x3a = 0_01110_10 = +0b1.10×2^-2 = 0.375
0x3b = 0_01110_11 = +0b1.11×2^-2 = 0.4375
0x3c = 0_01111_00 = +0b1.00×2^-1 = 0.5
0x3d = 0_01111_01 = +0b1.01×2^-1 = 0.625
0x3e = 0_01111_10 = +0b1.10×2^-1 = 0.75
0x3f = 0_01111_11 = +0b1.11×2^-1 = 0.875
0x40 = 0_10000_00 = +0b1.00×2^0 = 1.0
0x41 = 0_10000_01 = +0b1.01×2^0 = 1.25
0x42 = 0_10000_10 = +0b1.10×2^0 = 1.5
0x43 = 0_10000_11 = +0b1.11×2^0 = 1.75
0x44 = 0_10001_00 = +0b1.00×2^1 = 2.0
0x45 = 0_10001_01 = +0b1.01×2^1 = 2.5
0x46 = 0_10001_10 = +0b1.10×2^1 = 3.0
0x47 = 0_10001_11 = +0b1.11×2^1 = 3.5
0x48 = 0_10010_00 = +0b1.00×2^2 = 4.0
0x49 = 0_10010_01 = +0b1.01×2^2 = 5.0
0x4a = 0_10010_10 = +0b1.10×2^2 = 6.0
0x4b = 0_10010_11 = +0b1.11×2^2 = 7.0
0x4c = 0_10011_00 = +0b1.00×2^3 = 8.0
0x4d = 0_10011_01 = +0b1.01×2^3 = 10.0
0x4e = 0_10011_10 = +0b1.10×2^3 = 12.0
0x4f = 0_10011_11 = +0b1.11×2^3 = 14.0
0x50 = 0_10100_00 = +0b1.00×2^4 = 16.0
0x51 = 0_10100_01 = +0b1.01×2^4 = 20.0
0x52 = 0_10100_10 = +0b1.10×2^4 = 24.0
0x53 = 0_10100_11 = +0b1.11×2^4 = 28.0
0x54 = 0_10101_00 = +0b1.00×2^5 = 32.0
0x55 = 0_10101_01 = +0b1.01×2^5 = 40.0
0x56 = 0_10101_10 = +0b1.10×2^5 = 48.0
0x57 = 0_10101_11 = +0b1.11×2^5 = 56.0
0x58 = 0_10110_00 = +0b1.00×2^6 = 64.0
0x59 = 0_10110_01 = +0b1.01×2^6 = 80.0
0x5a = 0_10110_10 = +0b1.10×2^6 = 96.0
0x5b = 0_10110_11 = +0b1.11×2^6 = 112.0
0x5c = 0_10111_00 = +0b1.00×2^7 = 128.0
0x5d = 0_10111_01 = +0b1.01×2^7 = 160.0
0x5e = 0_10111_10 = +0b1.10×2^7 = 192.0
0x5f = 0_10111_11 = +0b1.11×2^7 = 224.0
0x60 = 0_11000_00 = +0b1.00×2^8 = 256.0
0x61 = 0_11000_01 = +0b1.01×2^8 = 320.0
0x62 = 0_11000_10 = +0b1.10×2^8 = 384.0
0x63 = 0_11000_11 = +0b1.11×2^8 = 448.0
0x64 = 0_11001_00 = +0b1.00×2^9 = 512.0
0x65 = 0_11001_01 = +0b1.01×2^9 = 640.0
0x66 = 0_11001_10 = +0b1.10×2^9 = 768.0
0x67 = 0_11001_11 = +0b1.11×2^9 = 896.0
0x68 = 0_11010_00 = +0b1.00×2^10 = 1024.0
0x69 = 0_11010_01 = +0b1.01×2^10 = 1280.0
0x6a = 0_11010_10 = +0b1.10×2^10 = 1536.0
0x6b = 0_11010_11 = +0b1.11×2^10 = 1792.0
0x6c = 0_11011_00 = +0b1.00×2^11 = 2048.0
0x6d = 0_11011_01 = +0b1.01×2^11 = 2560.0
0x6e = 0_11011_10 = +0b1.10×2^11 = 3072.0
0x6f = 0_11011_11 = +0b1.11×2^11 = 3584.0
0x70 = 0_11100_00 = +0b1.00×2^12 = 4096.0
0x71 = 0_11100_01 = +0b1.01×2^12 = 5120.0
0x72 = 0_11100_10 = +0b1.10×2^12 = 6144.0
0x73 = 0_11100_11 = +0b1.11×2^12 = 7168.0
0x74 = 0_11101_00 = +0b1.00×2^13 = 8192.0
0x75 = 0_11101_01 = +0b1.01×2^13 = 10240.0
0x76 = 0_11101_10 = +0b1.10×2^13 = 12288.0
0x77 = 0_11101_11 = +0b1.11×2^13 = 14336.0
0x78 = 0_11110_00 = +0b1.00×2^14 = 16384.0
0x79 = 0_11110_01 = +0b1.01×2^14 = 20480.0
0x7a = 0_11110_10 = +0b1.10×2^14 = 24576.0
0x7b = 0_11110_11 = +0b1.11×2^14 = 28672.0
0x7c = 0_11111_00 = +0b1.00×2^15 = 32768.0
0x7d = 0_11111_01 = +0b1.01×2^15 = 40960.0
0x7e = 0_11111_10 = +0b1.10×2^15 = 49152.0
0x7f = 0_11111_11 = +Inf
0x80 = 1_00000_00 = NaN
0x81 = 1_00000_01 = −0b0.01×2^-15 ≈ −7.6293945E-06
0x82 = 1_00000_10 = −0b0.10×2^-15 ≈ −1.5258789E-05
0x83 = 1_00000_11 = −0b0.11×2^-15 ≈ −2.2888184E-05
0x84 = 1_00001_00 = −0b1.00×2^-15 ≈ −3.0517578E-05
0x85 = 1_00001_01 = −0b1.01×2^-15 ≈ −3.8146973E-05
0x86 = 1_00001_10 = −0b1.10×2^-15 ≈ −4.5776367E-05
0x87 = 1_00001_11 = −0b1.11×2^-15 ≈ −5.3405762E-05
0x88 = 1_00010_00 = −0b1.00×2^-14 ≈ −6.1035156E-05
0x89 = 1_00010_01 = −0b1.01×2^-14 ≈ −7.6293945E-05
0x8a = 1_00010_10 = −0b1.10×2^-14 ≈ −9.1552734E-05
0x8b = 1_00010_11 = −0b1.11×2^-14 ≈ -0.00010681152
0x8c = 1_00011_00 = −0b1.00×2^-13 ≈ -0.00012207031
0x8d = 1_00011_01 = −0b1.01×2^-13 ≈ -0.00015258789
0x8e = 1_00011_10 = −0b1.10×2^-13 ≈ -0.00018310547
0x8f = 1_00011_11 = −0b1.11×2^-13 ≈ -0.00021362305
0x90 = 1_00100_00 = −0b1.00×2^-12 ≈ -0.00024414062
0x91 = 1_00100_01 = −0b1.01×2^-12 ≈ -0.00030517578
0x92 = 1_00100_10 = −0b1.10×2^-12 ≈ -0.00036621094
0x93 = 1_00100_11 = −0b1.11×2^-12 ≈ -0.00042724609
0x94 = 1_00101_00 = −0b1.00×2^-11 = -0.00048828125
0x95 = 1_00101_01 = −0b1.01×2^-11 ≈ -0.00061035156
0x96 = 1_00101_10 = −0b1.10×2^-11 ≈ -0.00073242188
0x97 = 1_00101_11 = −0b1.11×2^-11 ≈ -0.00085449219
0x98 = 1_00110_00 = −0b1.00×2^-10 = -0.0009765625
0x99 = 1_00110_01 = −0b1.01×2^-10 ≈ -0.0012207031
0x9a = 1_00110_10 = −0b1.10×2^-10 = -0.00146484375
0x9b = 1_00110_11 = −0b1.11×2^-10 ≈ -0.0017089844
0x9c = 1_00111_00 = −0b1.00×2^-9 = -0.001953125
0x9d = 1_00111_01 = −0b1.01×2^-9 = -0.00244140625
0x9e = 1_00111_10 = −0b1.10×2^-9 = -0.0029296875
0x9f = 1_00111_11 = −0b1.11×2^-9 = -0.00341796875
0xa0 = 1_01000_00 = −0b1.00×2^-8 = -0.00390625
0xa1 = 1_01000_01 = −0b1.01×2^-8 = -0.0048828125
0xa2 = 1_01000_10 = −0b1.10×2^-8 = -0.005859375
0xa3 = 1_01000_11 = −0b1.11×2^-8 = -0.0068359375
0xa4 = 1_01001_00 = −0b1.00×2^-7 = -0.0078125
0xa5 = 1_01001_01 = −0b1.01×2^-7 = -0.009765625
0xa6 = 1_01001_10 = −0b1.10×2^-7 = -0.01171875
0xa7 = 1_01001_11 = −0b1.11×2^-7 = -0.013671875
0xa8 = 1_01010_00 = −0b1.00×2^-6 = -0.015625
0xa9 = 1_01010_01 = −0b1.01×2^-6 = -0.01953125
0xaa = 1_01010_10 = −0b1.10×2^-6 = -0.0234375
0xab = 1_01010_11 = −0b1.11×2^-6 = -0.02734375
0xac = 1_01011_00 = −0b1.00×2^-5 = -0.03125
0xad = 1_01011_01 = −0b1.01×2^-5 = -0.0390625
0xae = 1_01011_10 = −0b1.10×2^-5 = -0.046875
0xaf = 1_01011_11 = −0b1.11×2^-5 = -0.0546875
0xb0 = 1_01100_00 = −0b1.00×2^-4 = -0.0625
0xb1 = 1_01100_01 = −0b1.01×2^-4 = -0.078125
0xb2 = 1_01100_10 = −0b1.10×2^-4 = -0.09375
0xb3 = 1_01100_11 = −0b1.11×2^-4 = -0.109375
0xb4 = 1_01101_00 = −0b1.00×2^-3 = -0.125
0xb5 = 1_01101_01 = −0b1.01×2^-3 = -0.15625
0xb6 = 1_01101_10 = −0b1.10×2^-3 = -0.1875
0xb7 = 1_01101_11 = −0b1.11×2^-3 = -0.21875
0xb8 = 1_01110_00 = −0b1.00×2^-2 = -0.25
0xb9 = 1_01110_01 = −0b1.01×2^-2 = -0.3125
0xba = 1_01110_10 = −0b1.10×2^-2 = -0.375
0xbb = 1_01110_11 = −0b1.11×2^-2 = -0.4375
0xbc = 1_01111_00 = −0b1.00×2^-1 = -0.5
0xbd = 1_01111_01 = −0b1.01×2^-1 = -0.625
0xbe = 1_01111_10 = −0b1.10×2^-1 = -0.75
0xbf = 1_01111_11 = −0b1.11×2^-1 = -0.875
0xc0 = 1_10000_00 = −0b1.00×2^0 = -1.0
0xc1 = 1_10000_01 = −0b1.01×2^0 = -1.25
0xc2 = 1_10000_10 = −0b1.10×2^0 = -1.5
0xc3 = 1_10000_11 = −0b1.11×2^0 = -1.75
0xc4 = 1_10001_00 = −0b1.00×2^1 = -2.0
0xc5 = 1_10001_01 = −0b1.01×2^1 = -2.5
0xc6 = 1_10001_10 = −0b1.10×2^1 = -3.0
0xc7 = 1_10001_11 = −0b1.11×2^1 = -3.5
0xc8 = 1_10010_00 = −0b1.00×2^2 = -4.0
0xc9 = 1_10010_01 = −0b1.01×2^2 = -5.0
0xca = 1_10010_10 = −0b1.10×2^2 = -6.0
0xcb = 1_10010_11 = −0b1.11×2^2 = -7.0
0xcc = 1_10011_00 = −0b1.00×2^3 = -8.0
0xcd = 1_10011_01 = −0b1.01×2^3 = -10.0
0xce = 1_10011_10 = −0b1.10×2^3 = -12.0
0xcf = 1_10011_11 = −0b1.11×2^3 = -14.0
0xd0 = 1_10100_00 = −0b1.00×2^4 = -16.0
0xd1 = 1_10100_01 = −0b1.01×2^4 = -20.0
0xd2 = 1_10100_10 = −0b1.10×2^4 = -24.0
0xd3 = 1_10100_11 = −0b1.11×2^4 = -28.0
0xd4 = 1_10101_00 = −0b1.00×2^5 = -32.0
0xd5 = 1_10101_01 = −0b1.01×2^5 = -40.0
0xd6 = 1_10101_10 = −0b1.10×2^5 = -48.0
0xd7 = 1_10101_11 = −0b1.11×2^5 = -56.0
0xd8 = 1_10110_00 = −0b1.00×2^6 = -64.0
0xd9 = 1_10110_01 = −0b1.01×2^6 = -80.0
0xda = 1_10110_10 = −0b1.10×2^6 = -96.0
0xdb = 1_10110_11 = −0b1.11×2^6 = -112.0
0xdc = 1_10111_00 = −0b1.00×2^7 = -128.0
0xdd = 1_10111_01 = −0b1.01×2^7 = -160.0
0xde = 1_10111_10 = −0b1.10×2^7 = -192.0
0xdf = 1_10111_11 = −0b1.11×2^7 = -224.0
0xe0 = 1_11000_00 = −0b1.00×2^8 = -256.0
0xe1 = 1_11000_01 = −0b1.01×2^8 = -320.0
0xe2 = 1_11000_10 = −0b1.10×2^8 = -384.0
0xe3 = 1_11000_11 = −0b1.11×2^8 = -448.0
0xe4 = 1_11001_00 = −0b1.00×2^9 = -512.0
0xe5 = 1_11001_01 = −0b1.01×2^9 = -640.0
0xe6 = 1_11001_10 = −0b1.10×2^9 = -768.0
0xe7 = 1_11001_11 = −0b1.11×2^9 = -896.0
0xe8 = 1_11010_00 = −0b1.00×2^10 = -1024.0
0xe9 = 1_11010_01 = −0b1.01×2^10 = -1280.0
0xea = 1_11010_10 = −0b1.10×2^10 = -1536.0
0xeb = 1_11010_11 = −0b1.11×2^10 = -1792.0
0xec = 1_11011_00 = −0b1.00×2^11 = -2048.0
0xed = 1_11011_01 = −0b1.01×2^11 = -2560.0
0xee = 1_11011_10 = −0b1.10×2^11 = -3072.0
0xef = 1_11011_11 = −0b1.11×2^11 = -3584.0
0xf0 = 1_11100_00 = −0b1.00×2^12 = -4096.0
0xf1 = 1_11100_01 = −0b1.01×2^12 = -5120.0
0xf2 = 1_11100_10 = −0b1.10×2^12 = -6144.0
0xf3 = 1_11100_11 = −0b1.11×2^12 = -7168.0
0xf4 = 1_11101_00 = −0b1.00×2^13 = -8192.0
0xf5 = 1_11101_01 = −0b1.01×2^13 = -10240.0
0xf6 = 1_11101_10 = −0b1.10×2^13 = -12288.0
0xf7 = 1_11101_11 = −0b1.11×2^13 = -14336.0
0xf8 = 1_11110_00 = −0b1.00×2^14 = -16384.0
0xf9 = 1_11110_01 = −0b1.01×2^14 = -20480.0
0xfa = 1_11110_10 = −0b1.10×2^14 = -24576.0
0xfb = 1_11110_11 = −0b1.11×2^14 = -28672.0
0xfc = 1_11111_00 = −0b1.00×2^15 = -32768.0
0xfd = 1_11111_01 = −0b1.01×2^15 = -40960.0
0xfe = 1_11111_10 = −0b1.10×2^15 = -49152.0
0xff = 1_11111_11 = −Inf

C.4 Value Table: P4, emin $= -7$, emax $= 7$

0x00 = 0_0000_000 = 0.0
0x01 = 0_0000_001 = +0b0.001×2^-7 = 0.0009765625
0x02 = 0_0000_010 = +0b0.010×2^-7 = 0.001953125
0x03 = 0_0000_011 = +0b0.011×2^-7 = 0.0029296875
0x04 = 0_0000_100 = +0b0.100×2^-7 = 0.00390625
0x05 = 0_0000_101 = +0b0.101×2^-7 = 0.0048828125
0x06 = 0_0000_110 = +0b0.110×2^-7 = 0.005859375
0x07 = 0_0000_111 = +0b0.111×2^-7 = 0.0068359375
0x08 = 0_0001_000 = +0b1.000×2^-7 = 0.0078125
0x09 = 0_0001_001 = +0b1.001×2^-7 = 0.0087890625
0x0a = 0_0001_010 = +0b1.010×2^-7 = 0.009765625
0x0b = 0_0001_011 = +0b1.011×2^-7 = 0.0107421875
0x0c = 0_0001_100 = +0b1.100×2^-7 = 0.01171875
0x0d = 0_0001_101 = +0b1.101×2^-7 = 0.0126953125
0x0e = 0_0001_110 = +0b1.110×2^-7 = 0.013671875
0x0f = 0_0001_111 = +0b1.111×2^-7 = 0.0146484375
0x10 = 0_0010_000 = +0b1.000×2^-6 = 0.015625
0x11 = 0_0010_001 = +0b1.001×2^-6 = 0.017578125
0x12 = 0_0010_010 = +0b1.010×2^-6 = 0.01953125
0x13 = 0_0010_011 = +0b1.011×2^-6 = 0.021484375
0x14 = 0_0010_100 = +0b1.100×2^-6 = 0.0234375
0x15 = 0_0010_101 = +0b1.101×2^-6 = 0.025390625
0x16 = 0_0010_110 = +0b1.110×2^-6 = 0.02734375
0x17 = 0_0010_111 = +0b1.111×2^-6 = 0.029296875
0x18 = 0_0011_000 = +0b1.000×2^-5 = 0.03125
0x19 = 0_0011_001 = +0b1.001×2^-5 = 0.03515625
0x1a = 0_0011_010 = +0b1.010×2^-5 = 0.0390625
0x1b = 0_0011_011 = +0b1.011×2^-5 = 0.04296875
0x1c = 0_0011_100 = +0b1.100×2^-5 = 0.046875
0x1d = 0_0011_101 = +0b1.101×2^-5 = 0.05078125
0x1e = 0_0011_110 = +0b1.110×2^-5 = 0.0546875
0x1f = 0_0011_111 = +0b1.111×2^-5 = 0.05859375
0x20 = 0_0100_000 = +0b1.000×2^-4 = 0.0625
0x21 = 0_0100_001 = +0b1.001×2^-4 = 0.0703125
0x22 = 0_0100_010 = +0b1.010×2^-4 = 0.078125
0x23 = 0_0100_011 = +0b1.011×2^-4 = 0.0859375
0x24 = 0_0100_100 = +0b1.100×2^-4 = 0.09375
0x25 = 0_0100_101 = +0b1.101×2^-4 = 0.1015625
0x26 = 0_0100_110 = +0b1.110×2^-4 = 0.109375
0x27 = 0_0100_111 = +0b1.111×2^-4 = 0.1171875
0x28 = 0_0101_000 = +0b1.000×2^-3 = 0.125
0x29 = 0_0101_001 = +0b1.001×2^-3 = 0.140625
0x2a = 0_0101_010 = +0b1.010×2^-3 = 0.15625
0x2b = 0_0101_011 = +0b1.011×2^-3 = 0.171875
0x2c = 0_0101_100 = +0b1.100×2^-3 = 0.1875
0x2d = 0_0101_101 = +0b1.101×2^-3 = 0.203125
0x2e = 0_0101_110 = +0b1.110×2^-3 = 0.21875
0x2f = 0_0101_111 = +0b1.111×2^-3 = 0.234375
0x30 = 0_0110_000 = +0b1.000×2^-2 = 0.25
0x31 = 0_0110_001 = +0b1.001×2^-2 = 0.28125
0x32 = 0_0110_010 = +0b1.010×2^-2 = 0.3125
0x33 = 0_0110_011 = +0b1.011×2^-2 = 0.34375
0x34 = 0_0110_100 = +0b1.100×2^-2 = 0.375
0x35 = 0_0110_101 = +0b1.101×2^-2 = 0.40625
0x36 = 0_0110_110 = +0b1.110×2^-2 = 0.4375
0x37 = 0_0110_111 = +0b1.111×2^-2 = 0.46875
0x38 = 0_0111_000 = +0b1.000×2^-1 = 0.5
0x39 = 0_0111_001 = +0b1.001×2^-1 = 0.5625
0x3a = 0_0111_010 = +0b1.010×2^-1 = 0.625
0x3b = 0_0111_011 = +0b1.011×2^-1 = 0.6875
0x3c = 0_0111_100 = +0b1.100×2^-1 = 0.75
0x3d = 0_0111_101 = +0b1.101×2^-1 = 0.8125
0x3e = 0_0111_110 = +0b1.110×2^-1 = 0.875
0x3f = 0_0111_111 = +0b1.111×2^-1 = 0.9375
0x40 = 0_1000_000 = +0b1.000×2^0 = 1.0
0x41 = 0_1000_001 = +0b1.001×2^0 = 1.125
0x42 = 0_1000_010 = +0b1.010×2^0 = 1.25
0x43 = 0_1000_011 = +0b1.011×2^0 = 1.375
0x44 = 0_1000_100 = +0b1.100×2^0 = 1.5
0x45 = 0_1000_101 = +0b1.101×2^0 = 1.625
0x46 = 0_1000_110 = +0b1.110×2^0 = 1.75
0x47 = 0_1000_111 = +0b1.111×2^0 = 1.875
0x48 = 0_1001_000 = +0b1.000×2^1 = 2.0
0x49 = 0_1001_001 = +0b1.001×2^1 = 2.25
0x4a = 0_1001_010 = +0b1.010×2^1 = 2.5
0x4b = 0_1001_011 = +0b1.011×2^1 = 2.75
0x4c = 0_1001_100 = +0b1.100×2^1 = 3.0
0x4d = 0_1001_101 = +0b1.101×2^1 = 3.25
0x4e = 0_1001_110 = +0b1.110×2^1 = 3.5
0x4f = 0_1001_111 = +0b1.111×2^1 = 3.75
0x50 = 0_1010_000 = +0b1.000×2^2 = 4.0
0x51 = 0_1010_001 = +0b1.001×2^2 = 4.5
0x52 = 0_1010_010 = +0b1.010×2^2 = 5.0
0x53 = 0_1010_011 = +0b1.011×2^2 = 5.5
0x54 = 0_1010_100 = +0b1.100×2^2 = 6.0
0x55 = 0_1010_101 = +0b1.101×2^2 = 6.5
0x56 = 0_1010_110 = +0b1.110×2^2 = 7.0
0x57 = 0_1010_111 = +0b1.111×2^2 = 7.5
0x58 = 0_1011_000 = +0b1.000×2^3 = 8.0
0x59 = 0_1011_001 = +0b1.001×2^3 = 9.0
0x5a = 0_1011_010 = +0b1.010×2^3 = 10.0
0x5b = 0_1011_011 = +0b1.011×2^3 = 11.0
0x5c = 0_1011_100 = +0b1.100×2^3 = 12.0
0x5d = 0_1011_101 = +0b1.101×2^3 = 13.0
0x5e = 0_1011_110 = +0b1.110×2^3 = 14.0
0x5f = 0_1011_111 = +0b1.111×2^3 = 15.0
0x60 = 0_1100_000 = +0b1.000×2^4 = 16.0
0x61 = 0_1100_001 = +0b1.001×2^4 = 18.0
0x62 = 0_1100_010 = +0b1.010×2^4 = 20.0
0x63 = 0_1100_011 = +0b1.011×2^4 = 22.0
0x64 = 0_1100_100 = +0b1.100×2^4 = 24.0
0x65 = 0_1100_101 = +0b1.101×2^4 = 26.0
0x66 = 0_1100_110 = +0b1.110×2^4 = 28.0
0x67 = 0_1100_111 = +0b1.111×2^4 = 30.0
0x68 = 0_1101_000 = +0b1.000×2^5 = 32.0
0x69 = 0_1101_001 = +0b1.001×2^5 = 36.0
0x6a = 0_1101_010 = +0b1.010×2^5 = 40.0
0x6b = 0_1101_011 = +0b1.011×2^5 = 44.0
0x6c = 0_1101_100 = +0b1.100×2^5 = 48.0
0x6d = 0_1101_101 = +0b1.101×2^5 = 52.0
0x6e = 0_1101_110 = +0b1.110×2^5 = 56.0
0x6f = 0_1101_111 = +0b1.111×2^5 = 60.0
0x70 = 0_1110_000 = +0b1.000×2^6 = 64.0
0x71 = 0_1110_001 = +0b1.001×2^6 = 72.0
0x72 = 0_1110_010 = +0b1.010×2^6 = 80.0
0x73 = 0_1110_011 = +0b1.011×2^6 = 88.0
0x74 = 0_1110_100 = +0b1.100×2^6 = 96.0
0x75 = 0_1110_101 = +0b1.101×2^6 = 104.0
0x76 = 0_1110_110 = +0b1.110×2^6 = 112.0
0x77 = 0_1110_111 = +0b1.111×2^6 = 120.0
0x78 = 0_1111_000 = +0b1.000×2^7 = 128.0
0x79 = 0_1111_001 = +0b1.001×2^7 = 144.0
0x7a = 0_1111_010 = +0b1.010×2^7 = 160.0
0x7b = 0_1111_011 = +0b1.011×2^7 = 176.0
0x7c = 0_1111_100 = +0b1.100×2^7 = 192.0
0x7d = 0_1111_101 = +0b1.101×2^7 = 208.0
0x7e = 0_1111_110 = +0b1.110×2^7 = 224.0
0x7f = 0_1111_111 = +Inf
0x80 = 1_0000_000 = NaN
0x81 = 1_0000_001 = −0b0.001×2^-7 = -0.0009765625
0x82 = 1_0000_010 = −0b0.010×2^-7 = -0.001953125
0x83 = 1_0000_011 = −0b0.011×2^-7 = -0.0029296875
0x84 = 1_0000_100 = −0b0.100×2^-7 = -0.00390625
0x85 = 1_0000_101 = −0b0.101×2^-7 = -0.0048828125
0x86 = 1_0000_110 = −0b0.110×2^-7 = -0.005859375
0x87 = 1_0000_111 = −0b0.111×2^-7 = -0.0068359375
0x88 = 1_0001_000 = −0b1.000×2^-7 = -0.0078125
0x89 = 1_0001_001 = −0b1.001×2^-7 = -0.0087890625
0x8a = 1_0001_010 = −0b1.010×2^-7 = -0.009765625
0x8b = 1_0001_011 = −0b1.011×2^-7 = -0.0107421875
0x8c = 1_0001_100 = −0b1.100×2^-7 = -0.01171875
0x8d = 1_0001_101 = −0b1.101×2^-7 = -0.0126953125
0x8e = 1_0001_110 = −0b1.110×2^-7 = -0.013671875
0x8f = 1_0001_111 = −0b1.111×2^-7 = -0.0146484375
0x90 = 1_0010_000 = −0b1.000×2^-6 = -0.015625
0x91 = 1_0010_001 = −0b1.001×2^-6 = -0.017578125
0x92 = 1_0010_010 = −0b1.010×2^-6 = -0.01953125
0x93 = 1_0010_011 = −0b1.011×2^-6 = -0.021484375
0x94 = 1_0010_100 = −0b1.100×2^-6 = -0.0234375
0x95 = 1_0010_101 = −0b1.101×2^-6 = -0.025390625
0x96 = 1_0010_110 = −0b1.110×2^-6 = -0.02734375
0x97 = 1_0010_111 = −0b1.111×2^-6 = -0.029296875
0x98 = 1_0011_000 = −0b1.000×2^-5 = -0.03125
0x99 = 1_0011_001 = −0b1.001×2^-5 = -0.03515625
0x9a = 1_0011_010 = −0b1.010×2^-5 = -0.0390625
0x9b = 1_0011_011 = −0b1.011×2^-5 = -0.04296875
0x9c = 1_0011_100 = −0b1.100×2^-5 = -0.046875
0x9d = 1_0011_101 = −0b1.101×2^-5 = -0.05078125
0x9e = 1_0011_110 = −0b1.110×2^-5 = -0.0546875
0x9f = 1_0011_111 = −0b1.111×2^-5 = -0.05859375
0xa0 = 1_0100_000 = −0b1.000×2^-4 = -0.0625
0xa1 = 1_0100_001 = −0b1.001×2^-4 = -0.0703125
0xa2 = 1_0100_010 = −0b1.010×2^-4 = -0.078125
0xa3 = 1_0100_011 = −0b1.011×2^-4 = -0.0859375
0xa4 = 1_0100_100 = −0b1.100×2^-4 = -0.09375
0xa5 = 1_0100_101 = −0b1.101×2^-4 = -0.1015625
0xa6 = 1_0100_110 = −0b1.110×2^-4 = -0.109375
0xa7 = 1_0100_111 = −0b1.111×2^-4 = -0.1171875
0xa8 = 1_0101_000 = −0b1.000×2^-3 = -0.125
0xa9 = 1_0101_001 = −0b1.001×2^-3 = -0.140625
0xaa = 1_0101_010 = −0b1.010×2^-3 = -0.15625
0xab = 1_0101_011 = −0b1.011×2^-3 = -0.171875
0xac = 1_0101_100 = −0b1.100×2^-3 = -0.1875
0xad = 1_0101_101 = −0b1.101×2^-3 = -0.203125
0xae = 1_0101_110 = −0b1.110×2^-3 = -0.21875
0xaf = 1_0101_111 = −0b1.111×2^-3 = -0.234375
0xb0 = 1_0110_000 = −0b1.000×2^-2 = -0.25
0xb1 = 1_0110_001 = −0b1.001×2^-2 = -0.28125
0xb2 = 1_0110_010 = −0b1.010×2^-2 = -0.3125
0xb3 = 1_0110_011 = −0b1.011×2^-2 = -0.34375
0xb4 = 1_0110_100 = −0b1.100×2^-2 = -0.375
0xb5 = 1_0110_101 = −0b1.101×2^-2 = -0.40625
0xb6 = 1_0110_110 = −0b1.110×2^-2 = -0.4375
0xb7 = 1_0110_111 = −0b1.111×2^-2 = -0.46875
0xb8 = 1_0111_000 = −0b1.000×2^-1 = -0.5
0xb9 = 1_0111_001 = −0b1.001×2^-1 = -0.5625
0xba = 1_0111_010 = −0b1.010×2^-1 = -0.625
0xbb = 1_0111_011 = −0b1.011×2^-1 = -0.6875
0xbc = 1_0111_100 = −0b1.100×2^-1 = -0.75
0xbd = 1_0111_101 = −0b1.101×2^-1 = -0.8125
0xbe = 1_0111_110 = −0b1.110×2^-1 = -0.875
0xbf = 1_0111_111 = −0b1.111×2^-1 = -0.9375
0xc0 = 1_1000_000 = −0b1.000×2^0 = -1.0
0xc1 = 1_1000_001 = −0b1.001×2^0 = -1.125
0xc2 = 1_1000_010 = −0b1.010×2^0 = -1.25
0xc3 = 1_1000_011 = −0b1.011×2^0 = -1.375
0xc4 = 1_1000_100 = −0b1.100×2^0 = -1.5
0xc5 = 1_1000_101 = −0b1.101×2^0 = -1.625
0xc6 = 1_1000_110 = −0b1.110×2^0 = -1.75
0xc7 = 1_1000_111 = −0b1.111×2^0 = -1.875
0xc8 = 1_1001_000 = −0b1.000×2^1 = -2.0
0xc9 = 1_1001_001 = −0b1.001×2^1 = -2.25
0xca = 1_1001_010 = −0b1.010×2^1 = -2.5
0xcb = 1_1001_011 = −0b1.011×2^1 = -2.75
0xcc = 1_1001_100 = −0b1.100×2^1 = -3.0
0xcd = 1_1001_101 = −0b1.101×2^1 = -3.25
0xce = 1_1001_110 = −0b1.110×2^1 = -3.5
0xcf = 1_1001_111 = −0b1.111×2^1 = -3.75
0xd0 = 1_1010_000 = −0b1.000×2^2 = -4.0
0xd1 = 1_1010_001 = −0b1.001×2^2 = -4.5
0xd2 = 1_1010_010 = −0b1.010×2^2 = -5.0
0xd3 = 1_1010_011 = −0b1.011×2^2 = -5.5
0xd4 = 1_1010_100 = −0b1.100×2^2 = -6.0
0xd5 = 1_1010_101 = −0b1.101×2^2 = -6.5
0xd6 = 1_1010_110 = −0b1.110×2^2 = -7.0
0xd7 = 1_1010_111 = −0b1.111×2^2 = -7.5
0xd8 = 1_1011_000 = −0b1.000×2^3 = -8.0
0xd9 = 1_1011_001 = −0b1.001×2^3 = -9.0
0xda = 1_1011_010 = −0b1.010×2^3 = -10.0
0xdb = 1_1011_011 = −0b1.011×2^3 = -11.0
0xdc = 1_1011_100 = −0b1.100×2^3 = -12.0
0xdd = 1_1011_101 = −0b1.101×2^3 = -13.0
0xde = 1_1011_110 = −0b1.110×2^3 = -14.0
0xdf = 1_1011_111 = −0b1.111×2^3 = -15.0
0xe0 = 1_1100_000 = −0b1.000×2^4 = -16.0
0xe1 = 1_1100_001 = −0b1.001×2^4 = -18.0
0xe2 = 1_1100_010 = −0b1.010×2^4 = -20.0
0xe3 = 1_1100_011 = −0b1.011×2^4 = -22.0
0xe4 = 1_1100_100 = −0b1.100×2^4 = -24.0
0xe5 = 1_1100_101 = −0b1.101×2^4 = -26.0
0xe6 = 1_1100_110 = −0b1.110×2^4 = -28.0
0xe7 = 1_1100_111 = −0b1.111×2^4 = -30.0
0xe8 = 1_1101_000 = −0b1.000×2^5 = -32.0
0xe9 = 1_1101_001 = −0b1.001×2^5 = -36.0
0xea = 1_1101_010 = −0b1.010×2^5 = -40.0
0xeb = 1_1101_011 = −0b1.011×2^5 = -44.0
0xec = 1_1101_100 = −0b1.100×2^5 = -48.0
0xed = 1_1101_101 = −0b1.101×2^5 = -52.0
0xee = 1_1101_110 = −0b1.110×2^5 = -56.0
0xef = 1_1101_111 = −0b1.111×2^5 = -60.0
0xf0 = 1_1110_000 = −0b1.000×2^6 = -64.0
0xf1 = 1_1110_001 = −0b1.001×2^6 = -72.0
0xf2 = 1_1110_010 = −0b1.010×2^6 = -80.0
0xf3 = 1_1110_011 = −0b1.011×2^6 = -88.0
0xf4 = 1_1110_100 = −0b1.100×2^6 = -96.0
0xf5 = 1_1110_101 = −0b1.101×2^6 = -104.0
0xf6 = 1_1110_110 = −0b1.110×2^6 = -112.0
0xf7 = 1_1110_111 = −0b1.111×2^6 = -120.0
0xf8 = 1_1111_000 = −0b1.000×2^7 = -128.0
0xf9 = 1_1111_001 = −0b1.001×2^7 = -144.0
0xfa = 1_1111_010 = −0b1.010×2^7 = -160.0
0xfb = 1_1111_011 = −0b1.011×2^7 = -176.0
0xfc = 1_1111_100 = −0b1.100×2^7 = -192.0
0xfd = 1_1111_101 = −0b1.101×2^7 = -208.0
0xfe = 1_1111_110 = −0b1.110×2^7 = -224.0
0xff = 1_1111_111 = −Inf

C.5 Value Table: P5, emin $= -3$, emax $= 3$

0x00 = 0_000_0000 = 0.0
0x01 = 0_000_0001 = +0b0.0001×2^-3 = 0.0078125
0x02 = 0_000_0010 = +0b0.0010×2^-3 = 0.015625
0x03 = 0_000_0011 = +0b0.0011×2^-3 = 0.0234375
0x04 = 0_000_0100 = +0b0.0100×2^-3 = 0.03125
0x05 = 0_000_0101 = +0b0.0101×2^-3 = 0.0390625
0x06 = 0_000_0110 = +0b0.0110×2^-3 = 0.046875
0x07 = 0_000_0111 = +0b0.0111×2^-3 = 0.0546875
0x08 = 0_000_1000 = +0b0.1000×2^-3 = 0.0625
0x09 = 0_000_1001 = +0b0.1001×2^-3 = 0.0703125
0x0a = 0_000_1010 = +0b0.1010×2^-3 = 0.078125
0x0b = 0_000_1011 = +0b0.1011×2^-3 = 0.0859375
0x0c = 0_000_1100 = +0b0.1100×2^-3 = 0.09375
0x0d = 0_000_1101 = +0b0.1101×2^-3 = 0.1015625
0x0e = 0_000_1110 = +0b0.1110×2^-3 = 0.109375
0x0f = 0_000_1111 = +0b0.1111×2^-3 = 0.1171875
0x10 = 0_001_0000 = +0b1.0000×2^-3 = 0.125
0x11 = 0_001_0001 = +0b1.0001×2^-3 = 0.1328125
0x12 = 0_001_0010 = +0b1.0010×2^-3 = 0.140625
0x13 = 0_001_0011 = +0b1.0011×2^-3 = 0.1484375
0x14 = 0_001_0100 = +0b1.0100×2^-3 = 0.15625
0x15 = 0_001_0101 = +0b1.0101×2^-3 = 0.1640625
0x16 = 0_001_0110 = +0b1.0110×2^-3 = 0.171875
0x17 = 0_001_0111 = +0b1.0111×2^-3 = 0.1796875
0x18 = 0_001_1000 = +0b1.1000×2^-3 = 0.1875
0x19 = 0_001_1001 = +0b1.1001×2^-3 = 0.1953125
0x1a = 0_001_1010 = +0b1.1010×2^-3 = 0.203125
0x1b = 0_001_1011 = +0b1.1011×2^-3 = 0.2109375
0x1c = 0_001_1100 = +0b1.1100×2^-3 = 0.21875
0x1d = 0_001_1101 = +0b1.1101×2^-3 = 0.2265625
0x1e = 0_001_1110 = +0b1.1110×2^-3 = 0.234375
0x1f = 0_001_1111 = +0b1.1111×2^-3 = 0.2421875
0x20 = 0_010_0000 = +0b1.0000×2^-2 = 0.25
0x21 = 0_010_0001 = +0b1.0001×2^-2 = 0.265625
0x22 = 0_010_0010 = +0b1.0010×2^-2 = 0.28125
0x23 = 0_010_0011 = +0b1.0011×2^-2 = 0.296875
0x24 = 0_010_0100 = +0b1.0100×2^-2 = 0.3125
0x25 = 0_010_0101 = +0b1.0101×2^-2 = 0.328125
0x26 = 0_010_0110 = +0b1.0110×2^-2 = 0.34375
0x27 = 0_010_0111 = +0b1.0111×2^-2 = 0.359375
0x28 = 0_010_1000 = +0b1.1000×2^-2 = 0.375
0x29 = 0_010_1001 = +0b1.1001×2^-2 = 0.390625
0x2a = 0_010_1010 = +0b1.1010×2^-2 = 0.40625
0x2b = 0_010_1011 = +0b1.1011×2^-2 = 0.421875
0x2c = 0_010_1100 = +0b1.1100×2^-2 = 0.4375
0x2d = 0_010_1101 = +0b1.1101×2^-2 = 0.453125
0x2e = 0_010_1110 = +0b1.1110×2^-2 = 0.46875
0x2f = 0_010_1111 = +0b1.1111×2^-2 = 0.484375
0x30 = 0_011_0000 = +0b1.0000×2^-1 = 0.5
0x31 = 0_011_0001 = +0b1.0001×2^-1 = 0.53125
0x32 = 0_011_0010 = +0b1.0010×2^-1 = 0.5625
0x33 = 0_011_0011 = +0b1.0011×2^-1 = 0.59375
0x34 = 0_011_0100 = +0b1.0100×2^-1 = 0.625
0x35 = 0_011_0101 = +0b1.0101×2^-1 = 0.65625
0x36 = 0_011_0110 = +0b1.0110×2^-1 = 0.6875
0x37 = 0_011_0111 = +0b1.0111×2^-1 = 0.71875
0x38 = 0_011_1000 = +0b1.1000×2^-1 = 0.75
0x39 = 0_011_1001 = +0b1.1001×2^-1 = 0.78125
0x3a = 0_011_1010 = +0b1.1010×2^-1 = 0.8125
0x3b = 0_011_1011 = +0b1.1011×2^-1 = 0.84375
0x3c = 0_011_1100 = +0b1.1100×2^-1 = 0.875
0x3d = 0_011_1101 = +0b1.1101×2^-1 = 0.90625
0x3e = 0_011_1110 = +0b1.1110×2^-1 = 0.9375
0x3f = 0_011_1111 = +0b1.1111×2^-1 = 0.96875
0x40 = 0_100_0000 = +0b1.0000×2^0 = 1.0
0x41 = 0_100_0001 = +0b1.0001×2^0 = 1.0625
0x42 = 0_100_0010 = +0b1.0010×2^0 = 1.125
0x43 = 0_100_0011 = +0b1.0011×2^0 = 1.1875
0x44 = 0_100_0100 = +0b1.0100×2^0 = 1.25
0x45 = 0_100_0101 = +0b1.0101×2^0 = 1.3125
0x46 = 0_100_0110 = +0b1.0110×2^0 = 1.375
0x47 = 0_100_0111 = +0b1.0111×2^0 = 1.4375
0x48 = 0_100_1000 = +0b1.1000×2^0 = 1.5
0x49 = 0_100_1001 = +0b1.1001×2^0 = 1.5625
0x4a = 0_100_1010 = +0b1.1010×2^0 = 1.625
0x4b = 0_100_1011 = +0b1.1011×2^0 = 1.6875
0x4c = 0_100_1100 = +0b1.1100×2^0 = 1.75
0x4d = 0_100_1101 = +0b1.1101×2^0 = 1.8125
0x4e = 0_100_1110 = +0b1.1110×2^0 = 1.875
0x4f = 0_100_1111 = +0b1.1111×2^0 = 1.9375
0x50 = 0_101_0000 = +0b1.0000×2^1 = 2.0
0x51 = 0_101_0001 = +0b1.0001×2^1 = 2.125
0x52 = 0_101_0010 = +0b1.0010×2^1 = 2.25
0x53 = 0_101_0011 = +0b1.0011×2^1 = 2.375
0x54 = 0_101_0100 = +0b1.0100×2^1 = 2.5
0x55 = 0_101_0101 = +0b1.0101×2^1 = 2.625
0x56 = 0_101_0110 = +0b1.0110×2^1 = 2.75
0x57 = 0_101_0111 = +0b1.0111×2^1 = 2.875
0x58 = 0_101_1000 = +0b1.1000×2^1 = 3.0
0x59 = 0_101_1001 = +0b1.1001×2^1 = 3.125
0x5a = 0_101_1010 = +0b1.1010×2^1 = 3.25
0x5b = 0_101_1011 = +0b1.1011×2^1 = 3.375
0x5c = 0_101_1100 = +0b1.1100×2^1 = 3.5
0x5d = 0_101_1101 = +0b1.1101×2^1 = 3.625
0x5e = 0_101_1110 = +0b1.1110×2^1 = 3.75
0x5f = 0_101_1111 = +0b1.1111×2^1 = 3.875
0x60 = 0_110_0000 = +0b1.0000×2^2 = 4.0
0x61 = 0_110_0001 = +0b1.0001×2^2 = 4.25
0x62 = 0_110_0010 = +0b1.0010×2^2 = 4.5
0x63 = 0_110_0011 = +0b1.0011×2^2 = 4.75
0x64 = 0_110_0100 = +0b1.0100×2^2 = 5.0
0x65 = 0_110_0101 = +0b1.0101×2^2 = 5.25
0x66 = 0_110_0110 = +0b1.0110×2^2 = 5.5
0x67 = 0_110_0111 = +0b1.0111×2^2 = 5.75
0x68 = 0_110_1000 = +0b1.1000×2^2 = 6.0
0x69 = 0_110_1001 = +0b1.1001×2^2 = 6.25
0x6a = 0_110_1010 = +0b1.1010×2^2 = 6.5
0x6b = 0_110_1011 = +0b1.1011×2^2 = 6.75
0x6c = 0_110_1100 = +0b1.1100×2^2 = 7.0
0x6d = 0_110_1101 = +0b1.1101×2^2 = 7.25
0x6e = 0_110_1110 = +0b1.1110×2^2 = 7.5
0x6f = 0_110_1111 = +0b1.1111×2^2 = 7.75
0x70 = 0_111_0000 = +0b1.0000×2^3 = 8.0
0x71 = 0_111_0001 = +0b1.0001×2^3 = 8.5
0x72 = 0_111_0010 = +0b1.0010×2^3 = 9.0
0x73 = 0_111_0011 = +0b1.0011×2^3 = 9.5
0x74 = 0_111_0100 = +0b1.0100×2^3 = 10.0
0x75 = 0_111_0101 = +0b1.0101×2^3 = 10.5
0x76 = 0_111_0110 = +0b1.0110×2^3 = 11.0
0x77 = 0_111_0111 = +0b1.0111×2^3 = 11.5
0x78 = 0_111_1000 = +0b1.1000×2^3 = 12.0
0x79 = 0_111_1001 = +0b1.1001×2^3 = 12.5
0x7a = 0_111_1010 = +0b1.1010×2^3 = 13.0
0x7b = 0_111_1011 = +0b1.1011×2^3 = 13.5
0x7c = 0_111_1100 = +0b1.1100×2^3 = 14.0
0x7d = 0_111_1101 = +0b1.1101×2^3 = 14.5
0x7e = 0_111_1110 = +0b1.1110×2^3 = 15.0
0x7f = 0_111_1111 = +Inf
0x80 = 1_000_0000 = NaN
0x81 = 1_000_0001 = −0b0.0001×2^-3 = -0.0078125
0x82 = 1_000_0010 = −0b0.0010×2^-3 = -0.015625
0x83 = 1_000_0011 = −0b0.0011×2^-3 = -0.0234375
0x84 = 1_000_0100 = −0b0.0100×2^-3 = -0.03125
0x85 = 1_000_0101 = −0b0.0101×2^-3 = -0.0390625
0x86 = 1_000_0110 = −0b0.0110×2^-3 = -0.046875
0x87 = 1_000_0111 = −0b0.0111×2^-3 = -0.0546875
0x88 = 1_000_1000 = −0b0.1000×2^-3 = -0.0625
0x89 = 1_000_1001 = −0b0.1001×2^-3 = -0.0703125
0x8a = 1_000_1010 = −0b0.1010×2^-3 = -0.078125
0x8b = 1_000_1011 = −0b0.1011×2^-3 = -0.0859375
0x8c = 1_000_1100 = −0b0.1100×2^-3 = -0.09375
0x8d = 1_000_1101 = −0b0.1101×2^-3 = -0.1015625
0x8e = 1_000_1110 = −0b0.1110×2^-3 = -0.109375
0x8f = 1_000_1111 = −0b0.1111×2^-3 = -0.1171875
0x90 = 1_001_0000 = −0b1.0000×2^-3 = -0.125
0x91 = 1_001_0001 = −0b1.0001×2^-3 = -0.1328125
0x92 = 1_001_0010 = −0b1.0010×2^-3 = -0.140625
0x93 = 1_001_0011 = −0b1.0011×2^-3 = -0.1484375
0x94 = 1_001_0100 = −0b1.0100×2^-3 = -0.15625
0x95 = 1_001_0101 = −0b1.0101×2^-3 = -0.1640625
0x96 = 1_001_0110 = −0b1.0110×2^-3 = -0.171875
0x97 = 1_001_0111 = −0b1.0111×2^-3 = -0.1796875
0x98 = 1_001_1000 = −0b1.1000×2^-3 = -0.1875
0x99 = 1_001_1001 = −0b1.1001×2^-3 = -0.1953125
0x9a = 1_001_1010 = −0b1.1010×2^-3 = -0.203125
0x9b = 1_001_1011 = −0b1.1011×2^-3 = -0.2109375
0x9c = 1_001_1100 = −0b1.1100×2^-3 = -0.21875
0x9d = 1_001_1101 = −0b1.1101×2^-3 = -0.2265625
0x9e = 1_001_1110 = −0b1.1110×2^-3 = -0.234375
0x9f = 1_001_1111 = −0b1.1111×2^-3 = -0.2421875
0xa0 = 1_010_0000 = −0b1.0000×2^-2 = -0.25
0xa1 = 1_010_0001 = −0b1.0001×2^-2 = -0.265625
0xa2 = 1_010_0010 = −0b1.0010×2^-2 = -0.28125
0xa3 = 1_010_0011 = −0b1.0011×2^-2 = -0.296875
0xa4 = 1_010_0100 = −0b1.0100×2^-2 = -0.3125
0xa5 = 1_010_0101 = −0b1.0101×2^-2 = -0.328125
0xa6 = 1_010_0110 = −0b1.0110×2^-2 = -0.34375
0xa7 = 1_010_0111 = −0b1.0111×2^-2 = -0.359375
0xa8 = 1_010_1000 = −0b1.1000×2^-2 = -0.375
0xa9 = 1_010_1001 = −0b1.1001×2^-2 = -0.390625
0xaa = 1_010_1010 = −0b1.1010×2^-2 = -0.40625
0xab = 1_010_1011 = −0b1.1011×2^-2 = -0.421875
0xac = 1_010_1100 = −0b1.1100×2^-2 = -0.4375
0xad = 1_010_1101 = −0b1.1101×2^-2 = -0.453125
0xae = 1_010_1110 = −0b1.1110×2^-2 = -0.46875
0xaf = 1_010_1111 = −0b1.1111×2^-2 = -0.484375
0xb0 = 1_011_0000 = −0b1.0000×2^-1 = -0.5
0xb1 = 1_011_0001 = −0b1.0001×2^-1 = -0.53125
0xb2 = 1_011_0010 = −0b1.0010×2^-1 = -0.5625
0xb3 = 1_011_0011 = −0b1.0011×2^-1 = -0.59375
0xb4 = 1_011_0100 = −0b1.0100×2^-1 = -0.625
0xb5 = 1_011_0101 = −0b1.0101×2^-1 = -0.65625
0xb6 = 1_011_0110 = −0b1.0110×2^-1 = -0.6875
0xb7 = 1_011_0111 = −0b1.0111×2^-1 = -0.71875
0xb8 = 1_011_1000 = −0b1.1000×2^-1 = -0.75
0xb9 = 1_011_1001 = −0b1.1001×2^-1 = -0.78125
0xba = 1_011_1010 = −0b1.1010×2^-1 = -0.8125
0xbb = 1_011_1011 = −0b1.1011×2^-1 = -0.84375
0xbc = 1_011_1100 = −0b1.1100×2^-1 = -0.875
0xbd = 1_011_1101 = −0b1.1101×2^-1 = -0.90625
0xbe = 1_011_1110 = −0b1.1110×2^-1 = -0.9375
0xbf = 1_011_1111 = −0b1.1111×2^-1 = -0.96875
0xc0 = 1_100_0000 = −0b1.0000×2^0 = -1.0
0xc1 = 1_100_0001 = −0b1.0001×2^0 = -1.0625
0xc2 = 1_100_0010 = −0b1.0010×2^0 = -1.125
0xc3 = 1_100_0011 = −0b1.0011×2^0 = -1.1875
0xc4 = 1_100_0100 = −0b1.0100×2^0 = -1.25
0xc5 = 1_100_0101 = −0b1.0101×2^0 = -1.3125
0xc6 = 1_100_0110 = −0b1.0110×2^0 = -1.375
0xc7 = 1_100_0111 = −0b1.0111×2^0 = -1.4375
0xc8 = 1_100_1000 = −0b1.1000×2^0 = -1.5
0xc9 = 1_100_1001 = −0b1.1001×2^0 = -1.5625
0xca = 1_100_1010 = −0b1.1010×2^0 = -1.625
0xcb = 1_100_1011 = −0b1.1011×2^0 = -1.6875
0xcc = 1_100_1100 = −0b1.1100×2^0 = -1.75
0xcd = 1_100_1101 = −0b1.1101×2^0 = -1.8125
0xce = 1_100_1110 = −0b1.1110×2^0 = -1.875
0xcf = 1_100_1111 = −0b1.1111×2^0 = -1.9375
0xd0 = 1_101_0000 = −0b1.0000×2^1 = -2.0
0xd1 = 1_101_0001 = −0b1.0001×2^1 = -2.125
0xd2 = 1_101_0010 = −0b1.0010×2^1 = -2.25
0xd3 = 1_101_0011 = −0b1.0011×2^1 = -2.375
0xd4 = 1_101_0100 = −0b1.0100×2^1 = -2.5
0xd5 = 1_101_0101 = −0b1.0101×2^1 = -2.625
0xd6 = 1_101_0110 = −0b1.0110×2^1 = -2.75
0xd7 = 1_101_0111 = −0b1.0111×2^1 = -2.875
0xd8 = 1_101_1000 = −0b1.1000×2^1 = -3.0
0xd9 = 1_101_1001 = −0b1.1001×2^1 = -3.125
0xda = 1_101_1010 = −0b1.1010×2^1 = -3.25
0xdb = 1_101_1011 = −0b1.1011×2^1 = -3.375
0xdc = 1_101_1100 = −0b1.1100×2^1 = -3.5
0xdd = 1_101_1101 = −0b1.1101×2^1 = -3.625
0xde = 1_101_1110 = −0b1.1110×2^1 = -3.75
0xdf = 1_101_1111 = −0b1.1111×2^1 = -3.875
0xe0 = 1_110_0000 = −0b1.0000×2^2 = -4.0
0xe1 = 1_110_0001 = −0b1.0001×2^2 = -4.25
0xe2 = 1_110_0010 = −0b1.0010×2^2 = -4.5
0xe3 = 1_110_0011 = −0b1.0011×2^2 = -4.75
0xe4 = 1_110_0100 = −0b1.0100×2^2 = -5.0
0xe5 = 1_110_0101 = −0b1.0101×2^2 = -5.25
0xe6 = 1_110_0110 = −0b1.0110×2^2 = -5.5
0xe7 = 1_110_0111 = −0b1.0111×2^2 = -5.75
0xe8 = 1_110_1000 = −0b1.1000×2^2 = -6.0
0xe9 = 1_110_1001 = −0b1.1001×2^2 = -6.25
0xea = 1_110_1010 = −0b1.1010×2^2 = -6.5
0xeb = 1_110_1011 = −0b1.1011×2^2 = -6.75
0xec = 1_110_1100 = −0b1.1100×2^2 = -7.0
0xed = 1_110_1101 = −0b1.1101×2^2 = -7.25
0xee = 1_110_1110 = −0b1.1110×2^2 = -7.5
0xef = 1_110_1111 = −0b1.1111×2^2 = -7.75
0xf0 = 1_111_0000 = −0b1.0000×2^3 = -8.0
0xf1 = 1_111_0001 = −0b1.0001×2^3 = -8.5
0xf2 = 1_111_0010 = −0b1.0010×2^3 = -9.0
0xf3 = 1_111_0011 = −0b1.0011×2^3 = -9.5
0xf4 = 1_111_0100 = −0b1.0100×2^3 = -10.0
0xf5 = 1_111_0101 = −0b1.0101×2^3 = -10.5
0xf6 = 1_111_0110 = −0b1.0110×2^3 = -11.0
0xf7 = 1_111_0111 = −0b1.0111×2^3 = -11.5
0xf8 = 1_111_1000 = −0b1.1000×2^3 = -12.0
0xf9 = 1_111_1001 = −0b1.1001×2^3 = -12.5
0xfa = 1_111_1010 = −0b1.1010×2^3 = -13.0
0xfb = 1_111_1011 = −0b1.1011×2^3 = -13.5
0xfc = 1_111_1100 = −0b1.1100×2^3 = -14.0
0xfd = 1_111_1101 = −0b1.1101×2^3 = -14.5
0xfe = 1_111_1110 = −0b1.1110×2^3 = -15.0
0xff = 1_111_1111 = −Inf

C.6 Value Table: P6, emin $= -1$, emax $= 1$

0x00 = 0_00_00000 = 0.0
0x01 = 0_00_00001 = +0b0.00001×2^-1 = 0.015625
0x02 = 0_00_00010 = +0b0.00010×2^-1 = 0.03125
0x03 = 0_00_00011 = +0b0.00011×2^-1 = 0.046875
0x04 = 0_00_00100 = +0b0.00100×2^-1 = 0.0625
0x05 = 0_00_00101 = +0b0.00101×2^-1 = 0.078125
0x06 = 0_00_00110 = +0b0.00110×2^-1 = 0.09375
0x07 = 0_00_00111 = +0b0.00111×2^-1 = 0.109375
0x08 = 0_00_01000 = +0b0.01000×2^-1 = 0.125
0x09 = 0_00_01001 = +0b0.01001×2^-1 = 0.140625
0x0a = 0_00_01010 = +0b0.01010×2^-1 = 0.15625
0x0b = 0_00_01011 = +0b0.01011×2^-1 = 0.171875
0x0c = 0_00_01100 = +0b0.01100×2^-1 = 0.1875
0x0d = 0_00_01101 = +0b0.01101×2^-1 = 0.203125
0x0e = 0_00_01110 = +0b0.01110×2^-1 = 0.21875
0x0f = 0_00_01111 = +0b0.01111×2^-1 = 0.234375
0x10 = 0_00_10000 = +0b0.10000×2^-1 = 0.25
0x11 = 0_00_10001 = +0b0.10001×2^-1 = 0.265625
0x12 = 0_00_10010 = +0b0.10010×2^-1 = 0.28125
0x13 = 0_00_10011 = +0b0.10011×2^-1 = 0.296875
0x14 = 0_00_10100 = +0b0.10100×2^-1 = 0.3125
0x15 = 0_00_10101 = +0b0.10101×2^-1 = 0.328125
0x16 = 0_00_10110 = +0b0.10110×2^-1 = 0.34375
0x17 = 0_00_10111 = +0b0.10111×2^-1 = 0.359375
0x18 = 0_00_11000 = +0b0.11000×2^-1 = 0.375
0x19 = 0_00_11001 = +0b0.11001×2^-1 = 0.390625
0x1a = 0_00_11010 = +0b0.11010×2^-1 = 0.40625
0x1b = 0_00_11011 = +0b0.11011×2^-1 = 0.421875
0x1c = 0_00_11100 = +0b0.11100×2^-1 = 0.4375
0x1d = 0_00_11101 = +0b0.11101×2^-1 = 0.453125
0x1e = 0_00_11110 = +0b0.11110×2^-1 = 0.46875
0x1f = 0_00_11111 = +0b0.11111×2^-1 = 0.484375
0x20 = 0_01_00000 = +0b1.00000×2^-1 = 0.5
0x21 = 0_01_00001 = +0b1.00001×2^-1 = 0.515625
0x22 = 0_01_00010 = +0b1.00010×2^-1 = 0.53125
0x23 = 0_01_00011 = +0b1.00011×2^-1 = 0.546875
0x24 = 0_01_00100 = +0b1.00100×2^-1 = 0.5625
0x25 = 0_01_00101 = +0b1.00101×2^-1 = 0.578125
0x26 = 0_01_00110 = +0b1.00110×2^-1 = 0.59375
0x27 = 0_01_00111 = +0b1.00111×2^-1 = 0.609375
0x28 = 0_01_01000 = +0b1.01000×2^-1 = 0.625
0x29 = 0_01_01001 = +0b1.01001×2^-1 = 0.640625
0x2a = 0_01_01010 = +0b1.01010×2^-1 = 0.65625
0x2b = 0_01_01011 = +0b1.01011×2^-1 = 0.671875
0x2c = 0_01_01100 = +0b1.01100×2^-1 = 0.6875
0x2d = 0_01_01101 = +0b1.01101×2^-1 = 0.703125
0x2e = 0_01_01110 = +0b1.01110×2^-1 = 0.71875
0x2f = 0_01_01111 = +0b1.01111×2^-1 = 0.734375
0x30 = 0_01_10000 = +0b1.10000×2^-1 = 0.75
0x31 = 0_01_10001 = +0b1.10001×2^-1 = 0.765625
0x32 = 0_01_10010 = +0b1.10010×2^-1 = 0.78125
0x33 = 0_01_10011 = +0b1.10011×2^-1 = 0.796875
0x34 = 0_01_10100 = +0b1.10100×2^-1 = 0.8125
0x35 = 0_01_10101 = +0b1.10101×2^-1 = 0.828125
0x36 = 0_01_10110 = +0b1.10110×2^-1 = 0.84375
0x37 = 0_01_10111 = +0b1.10111×2^-1 = 0.859375
0x38 = 0_01_11000 = +0b1.11000×2^-1 = 0.875
0x39 = 0_01_11001 = +0b1.11001×2^-1 = 0.890625
0x3a = 0_01_11010 = +0b1.11010×2^-1 = 0.90625
0x3b = 0_01_11011 = +0b1.11011×2^-1 = 0.921875
0x3c = 0_01_11100 = +0b1.11100×2^-1 = 0.9375
0x3d = 0_01_11101 = +0b1.11101×2^-1 = 0.953125
0x3e = 0_01_11110 = +0b1.11110×2^-1 = 0.96875
0x3f = 0_01_11111 = +0b1.11111×2^-1 = 0.984375

0x40 = 0_10_00000 = +0b1.00000×2^0 = 1.0
0x41 = 0_10_00001 = +0b1.00001×2^0 = 1.03125
0x42 = 0_10_00010 = +0b1.00010×2^0 = 1.0625
0x43 = 0_10_00011 = +0b1.00011×2^0 = 1.09375
0x44 = 0_10_00100 = +0b1.00100×2^0 = 1.125
0x45 = 0_10_00101 = +0b1.00101×2^0 = 1.15625
0x46 = 0_10_00110 = +0b1.00110×2^0 = 1.1875
0x47 = 0_10_00111 = +0b1.00111×2^0 = 1.21875
0x48 = 0_10_01000 = +0b1.01000×2^0 = 1.25
0x49 = 0_10_01001 = +0b1.01001×2^0 = 1.28125
0x4a = 0_10_01010 = +0b1.01010×2^0 = 1.3125
0x4b = 0_10_01011 = +0b1.01011×2^0 = 1.34375
0x4c = 0_10_01100 = +0b1.01100×2^0 = 1.375
0x4d = 0_10_01101 = +0b1.01101×2^0 = 1.40625
0x4e = 0_10_01110 = +0b1.01110×2^0 = 1.4375
0x4f = 0_10_01111 = +0b1.01111×2^0 = 1.46875
0x50 = 0_10_10000 = +0b1.10000×2^0 = 1.5
0x51 = 0_10_10001 = +0b1.10001×2^0 = 1.53125
0x52 = 0_10_10010 = +0b1.10010×2^0 = 1.5625
0x53 = 0_10_10011 = +0b1.10011×2^0 = 1.59375
0x54 = 0_10_10100 = +0b1.10100×2^0 = 1.625
0x55 = 0_10_10101 = +0b1.10101×2^0 = 1.65625
0x56 = 0_10_10110 = +0b1.10110×2^0 = 1.6875
0x57 = 0_10_10111 = +0b1.10111×2^0 = 1.71875
0x58 = 0_10_11000 = +0b1.11000×2^0 = 1.75
0x59 = 0_10_11001 = +0b1.11001×2^0 = 1.78125
0x5a = 0_10_11010 = +0b1.11010×2^0 = 1.8125
0x5b = 0_10_11011 = +0b1.11011×2^0 = 1.84375
0x5c = 0_10_11100 = +0b1.11100×2^0 = 1.875
0x5d = 0_10_11101 = +0b1.11101×2^0 = 1.90625
0x5e = 0_10_11110 = +0b1.11110×2^0 = 1.9375
0x5f = 0_10_11111 = +0b1.11111×2^0 = 1.96875
0x60 = 0_11_00000 = +0b1.00000×2^1 = 2.0
0x61 = 0_11_00001 = +0b1.00001×2^1 = 2.0625
0x62 = 0_11_00010 = +0b1.00010×2^1 = 2.125
0x63 = 0_11_00011 = +0b1.00011×2^1 = 2.1875
0x64 = 0_11_00100 = +0b1.00100×2^1 = 2.25
0x65 = 0_11_00101 = +0b1.00101×2^1 = 2.3125
0x66 = 0_11_00110 = +0b1.00110×2^1 = 2.375
0x67 = 0_11_00111 = +0b1.00111×2^1 = 2.4375
0x68 = 0_11_01000 = +0b1.01000×2^1 = 2.5
0x69 = 0_11_01001 = +0b1.01001×2^1 = 2.5625
0x6a = 0_11_01010 = +0b1.01010×2^1 = 2.625
0x6b = 0_11_01011 = +0b1.01011×2^1 = 2.6875
0x6c = 0_11_01100 = +0b1.01100×2^1 = 2.75
0x6d = 0_11_01101 = +0b1.01101×2^1 = 2.8125
0x6e = 0_11_01110 = +0b1.01110×2^1 = 2.875
0x6f = 0_11_01111 = +0b1.01111×2^1 = 2.9375
0x70 = 0_11_10000 = +0b1.10000×2^1 = 3.0
0x71 = 0_11_10001 = +0b1.10001×2^1 = 3.0625
0x72 = 0_11_10010 = +0b1.10010×2^1 = 3.125
0x73 = 0_11_10011 = +0b1.10011×2^1 = 3.1875
0x74 = 0_11_10100 = +0b1.10100×2^1 = 3.25
0x75 = 0_11_10101 = +0b1.10101×2^1 = 3.3125
0x76 = 0_11_10110 = +0b1.10110×2^1 = 3.375
0x77 = 0_11_10111 = +0b1.10111×2^1 = 3.4375
0x78 = 0_11_11000 = +0b1.11000×2^1 = 3.5
0x79 = 0_11_11001 = +0b1.11001×2^1 = 3.5625
0x7a = 0_11_11010 = +0b1.11010×2^1 = 3.625
0x7b = 0_11_11011 = +0b1.11011×2^1 = 3.6875
0x7c = 0_11_11100 = +0b1.11100×2^1 = 3.75
0x7d = 0_11_11101 = +0b1.11101×2^1 = 3.8125
0x7e = 0_11_11110 = +0b1.11110×2^1 = 3.875
0x7f = 0_11_11111 = +Inf

0x80 = 1_00_00000 = NaN
0x81 = 1_00_00001 = −0b0.00001×2^-1 = -0.015625
0x82 = 1_00_00010 = −0b0.00010×2^-1 = -0.03125
0x83 = 1_00_00011 = −0b0.00011×2^-1 = -0.046875
0x84 = 1_00_00100 = −0b0.00100×2^-1 = -0.0625
0x85 = 1_00_00101 = −0b0.00101×2^-1 = -0.078125
0x86 = 1_00_00110 = −0b0.00110×2^-1 = -0.09375
0x87 = 1_00_00111 = −0b0.00111×2^-1 = -0.109375
0x88 = 1_00_01000 = −0b0.01000×2^-1 = -0.125
0x89 = 1_00_01001 = −0b0.01001×2^-1 = -0.140625
0x8a = 1_00_01010 = −0b0.01010×2^-1 = -0.15625
0x8b = 1_00_01011 = −0b0.01011×2^-1 = -0.171875
0x8c = 1_00_01100 = −0b0.01100×2^-1 = -0.1875
0x8d = 1_00_01101 = −0b0.01101×2^-1 = -0.203125
0x8e = 1_00_01110 = −0b0.01110×2^-1 = -0.21875
0x8f = 1_00_01111 = −0b0.01111×2^-1 = -0.234375
0x90 = 1_00_10000 = −0b0.10000×2^-1 = -0.25
0x91 = 1_00_10001 = −0b0.10001×2^-1 = -0.265625
0x92 = 1_00_10010 = −0b0.10010×2^-1 = -0.28125
0x93 = 1_00_10011 = −0b0.10011×2^-1 = -0.296875
0x94 = 1_00_10100 = −0b0.10100×2^-1 = -0.3125
0x95 = 1_00_10101 = −0b0.10101×2^-1 = -0.328125
0x96 = 1_00_10110 = −0b0.10110×2^-1 = -0.34375
0x97 = 1_00_10111 = −0b0.10111×2^-1 = -0.359375
0x98 = 1_00_11000 = −0b0.11000×2^-1 = -0.375
0x99 = 1_00_11001 = −0b0.11001×2^-1 = -0.390625
0x9a = 1_00_11010 = −0b0.11010×2^-1 = -0.40625
0x9b = 1_00_11011 = −0b0.11011×2^-1 = -0.421875
0x9c = 1_00_11100 = −0b0.11100×2^-1 = -0.4375
0x9d = 1_00_11101 = −0b0.11101×2^-1 = -0.453125
0x9e = 1_00_11110 = −0b0.11110×2^-1 = -0.46875
0x9f = 1_00_11111 = −0b0.11111×2^-1 = -0.484375
0xa0 = 1_01_00000 = −0b1.00000×2^-1 = -0.5
0xa1 = 1_01_00001 = −0b1.00001×2^-1 = -0.515625
0xa2 = 1_01_00010 = −0b1.00010×2^-1 = -0.53125
0xa3 = 1_01_00011 = −0b1.00011×2^-1 = -0.546875
0xa4 = 1_01_00100 = −0b1.00100×2^-1 = -0.5625
0xa5 = 1_01_00101 = −0b1.00101×2^-1 = -0.578125
0xa6 = 1_01_00110 = −0b1.00110×2^-1 = -0.59375
0xa7 = 1_01_00111 = −0b1.00111×2^-1 = -0.609375
0xa8 = 1_01_01000 = −0b1.01000×2^-1 = -0.625
0xa9 = 1_01_01001 = −0b1.01001×2^-1 = -0.640625
0xaa = 1_01_01010 = −0b1.01010×2^-1 = -0.65625
0xab = 1_01_01011 = −0b1.01011×2^-1 = -0.671875
0xac = 1_01_01100 = −0b1.01100×2^-1 = -0.6875
0xad = 1_01_01101 = −0b1.01101×2^-1 = -0.703125
0xae = 1_01_01110 = −0b1.01110×2^-1 = -0.71875
0xaf = 1_01_01111 = −0b1.01111×2^-1 = -0.734375
0xb0 = 1_01_10000 = −0b1.10000×2^-1 = -0.75
0xb1 = 1_01_10001 = −0b1.10001×2^-1 = -0.765625
0xb2 = 1_01_10010 = −0b1.10010×2^-1 = -0.78125
0xb3 = 1_01_10011 = −0b1.10011×2^-1 = -0.796875
0xb4 = 1_01_10100 = −0b1.10100×2^-1 = -0.8125
0xb5 = 1_01_10101 = −0b1.10101×2^-1 = -0.828125
0xb6 = 1_01_10110 = −0b1.10110×2^-1 = -0.84375
0xb7 = 1_01_10111 = −0b1.10111×2^-1 = -0.859375
0xb8 = 1_01_11000 = −0b1.11000×2^-1 = -0.875
0xb9 = 1_01_11001 = −0b1.11001×2^-1 = -0.890625
0xba = 1_01_11010 = −0b1.11010×2^-1 = -0.90625
0xbb = 1_01_11011 = −0b1.11011×2^-1 = -0.921875
0xbc = 1_01_11100 = −0b1.11100×2^-1 = -0.9375
0xbd = 1_01_11101 = −0b1.11101×2^-1 = -0.953125
0xbe = 1_01_11110 = −0b1.11110×2^-1 = -0.96875
0xbf = 1_01_11111 = −0b1.11111×2^-1 = -0.984375

0xc0 = 1_10_00000 = −0b1.00000×2^0 = -1.0
0xc1 = 1_10_00001 = −0b1.00001×2^0 = -1.03125
0xc2 = 1_10_00010 = −0b1.00010×2^0 = -1.0625
0xc3 = 1_10_00011 = −0b1.00011×2^0 = -1.09375
0xc4 = 1_10_00100 = −0b1.00100×2^0 = -1.125
0xc5 = 1_10_00101 = −0b1.00101×2^0 = -1.15625
0xc6 = 1_10_00110 = −0b1.00110×2^0 = -1.1875
0xc7 = 1_10_00111 = −0b1.00111×2^0 = -1.21875
0xc8 = 1_10_01000 = −0b1.01000×2^0 = -1.25
0xc9 = 1_10_01001 = −0b1.01001×2^0 = -1.28125
0xca = 1_10_01010 = −0b1.01010×2^0 = -1.3125
0xcb = 1_10_01011 = −0b1.01011×2^0 = -1.34375
0xcc = 1_10_01100 = −0b1.01100×2^0 = -1.375
0xcd = 1_10_01101 = −0b1.01101×2^0 = -1.40625
0xce = 1_10_01110 = −0b1.01110×2^0 = -1.4375
0xcf = 1_10_01111 = −0b1.01111×2^0 = -1.46875
0xd0 = 1_10_10000 = −0b1.10000×2^0 = -1.5
0xd1 = 1_10_10001 = −0b1.10001×2^0 = -1.53125
0xd2 = 1_10_10010 = −0b1.10010×2^0 = -1.5625
0xd3 = 1_10_10011 = −0b1.10011×2^0 = -1.59375
0xd4 = 1_10_10100 = −0b1.10100×2^0 = -1.625
0xd5 = 1_10_10101 = −0b1.10101×2^0 = -1.65625
0xd6 = 1_10_10110 = −0b1.10110×2^0 = -1.6875
0xd7 = 1_10_10111 = −0b1.10111×2^0 = -1.71875
0xd8 = 1_10_11000 = −0b1.11000×2^0 = -1.75
0xd9 = 1_10_11001 = −0b1.11001×2^0 = -1.78125
0xda = 1_10_11010 = −0b1.11010×2^0 = -1.8125
0xdb = 1_10_11011 = −0b1.11011×2^0 = -1.84375
0xdc = 1_10_11100 = −0b1.11100×2^0 = -1.875
0xdd = 1_10_11101 = −0b1.11101×2^0 = -1.90625
0xde = 1_10_11110 = −0b1.11110×2^0 = -1.9375
0xdf = 1_10_11111 = −0b1.11111×2^0 = -1.96875
0xe0 = 1_11_00000 = −0b1.00000×2^1 = -2.0
0xe1 = 1_11_00001 = −0b1.00001×2^1 = -2.0625
0xe2 = 1_11_00010 = −0b1.00010×2^1 = -2.125
0xe3 = 1_11_00011 = −0b1.00011×2^1 = -2.1875
0xe4 = 1_11_00100 = −0b1.00100×2^1 = -2.25
0xe5 = 1_11_00101 = −0b1.00101×2^1 = -2.3125
0xe6 = 1_11_00110 = −0b1.00110×2^1 = -2.375
0xe7 = 1_11_00111 = −0b1.00111×2^1 = -2.4375
0xe8 = 1_11_01000 = −0b1.01000×2^1 = -2.5
0xe9 = 1_11_01001 = −0b1.01001×2^1 = -2.5625
0xea = 1_11_01010 = −0b1.01010×2^1 = -2.625
0xeb = 1_11_01011 = −0b1.01011×2^1 = -2.6875
0xec = 1_11_01100 = −0b1.01100×2^1 = -2.75
0xed = 1_11_01101 = −0b1.01101×2^1 = -2.8125
0xee = 1_11_01110 = −0b1.01110×2^1 = -2.875
0xef = 1_11_01111 = −0b1.01111×2^1 = -2.9375
0xf0 = 1_11_10000 = −0b1.10000×2^1 = -3.0
0xf1 = 1_11_10001 = −0b1.10001×2^1 = -3.0625
0xf2 = 1_11_10010 = −0b1.10010×2^1 = -3.125
0xf3 = 1_11_10011 = −0b1.10011×2^1 = -3.1875
0xf4 = 1_11_10100 = −0b1.10100×2^1 = -3.25
0xf5 = 1_11_10101 = −0b1.10101×2^1 = -3.3125
0xf6 = 1_11_10110 = −0b1.10110×2^1 = -3.375
0xf7 = 1_11_10111 = −0b1.10111×2^1 = -3.4375
0xf8 = 1_11_11000 = −0b1.11000×2^1 = -3.5
0xf9 = 1_11_11001 = −0b1.11001×2^1 = -3.5625
0xfa = 1_11_11010 = −0b1.11010×2^1 = -3.625
0xfb = 1_11_11011 = −0b1.11011×2^1 = -3.6875
0xfc = 1_11_11100 = −0b1.11100×2^1 = -3.75
0xfd = 1_11_11101 = −0b1.11101×2^1 = -3.8125
0xfe = 1_11_11110 = −0b1.11110×2^1 = -3.875
0xff = 1_11_11111 = −Inf

C.7 Value Table: P7, emin $= 0$, emax $= 0$ (linear)

0x00 = 0_0_000000 = 0.0	0x40 = 0_1_000000 = +0b1.000000×2^0 = 1.0	0x80 = 1_0_000000 = NaN	0xc0 = 1_1_000000 = −0b1.000000×2^0 = -1.0
0x01 = 0_0_000001 = +0b0.000001×2^0 = 0.015625	0x41 = 0_1_000001 = +0b1.000001×2^0 = 1.015625	0x81 = 1_0_000001 = −0b0.000001×2^0 = -0.015625	0xc1 = 1_1_000001 = −0b1.000001×2^0 = -1.015625
0x02 = 0_0_000010 = +0b0.000010×2^0 = 0.03125	0x42 = 0_1_000010 = +0b1.000010×2^0 = 1.03125	0x82 = 1_0_000010 = −0b0.000010×2^0 = -0.03125	0xc2 = 1_1_000010 = −0b1.000010×2^0 = -1.03125
0x03 = 0_0_000011 = +0b0.000011×2^0 = 0.046875	0x43 = 0_1_000011 = +0b1.000011×2^0 = 1.046875	0x83 = 1_0_000011 = −0b0.000011×2^0 = -0.046875	0xc3 = 1_1_000011 = −0b1.000011×2^0 = -1.046875
0x04 = 0_0_000100 = +0b0.000100×2^0 = 0.0625	0x44 = 0_1_000100 = +0b1.000100×2^0 = 1.0625	0x84 = 1_0_000100 = −0b0.000100×2^0 = -0.0625	0xc4 = 1_1_000100 = −0b1.000100×2^0 = -1.0625
0x05 = 0_0_000101 = +0b0.000101×2^0 = 0.078125	0x45 = 0_1_000101 = +0b1.000101×2^0 = 1.078125	0x85 = 1_0_000101 = −0b0.000101×2^0 = -0.078125	0xc5 = 1_1_000101 = −0b1.000101×2^0 = -1.078125
0x06 = 0_0_000110 = +0b0.000110×2^0 = 0.09375	0x46 = 0_1_000110 = +0b1.000110×2^0 = 1.09375	0x86 = 1_0_000110 = −0b0.000110×2^0 = -0.09375	0xc6 = 1_1_000110 = −0b1.000110×2^0 = -1.09375
0x07 = 0_0_000111 = +0b0.000111×2^0 = 0.109375	0x47 = 0_1_000111 = +0b1.000111×2^0 = 1.109375	0x87 = 1_0_000111 = −0b0.000111×2^0 = -0.109375	0xc7 = 1_1_000111 = −0b1.000111×2^0 = -1.109375
0x08 = 0_0_001000 = +0b0.001000×2^0 = 0.125	0x48 = 0_1_001000 = +0b1.001000×2^0 = 1.125	0x88 = 1_0_001000 = −0b0.001000×2^0 = -0.125	0xc8 = 1_1_001000 = −0b1.001000×2^0 = -1.125
0x09 = 0_0_001001 = +0b0.001001×2^0 = 0.140625	0x49 = 0_1_001001 = +0b1.001001×2^0 = 1.140625	0x89 = 1_0_001001 = −0b0.001001×2^0 = -0.140625	0xc9 = 1_1_001001 = −0b1.001001×2^0 = -1.140625
0x0a = 0_0_001010 = +0b0.001010×2^0 = 0.15625	0x4a = 0_1_001010 = +0b1.001010×2^0 = 1.15625	0x8a = 1_0_001010 = −0b0.001010×2^0 = -0.15625	0xca = 1_1_001010 = −0b1.001010×2^0 = -1.15625
0x0b = 0_0_001011 = +0b0.001011×2^0 = 0.171875	0x4b = 0_1_001011 = +0b1.001011×2^0 = 1.171875	0x8b = 1_0_001011 = −0b0.001011×2^0 = -0.171875	0xcb = 1_1_001011 = −0b1.001011×2^0 = -1.171875
0x0c = 0_0_001100 = +0b0.001100×2^0 = 0.1875	0x4c = 0_1_001100 = +0b1.001100×2^0 = 1.1875	0x8c = 1_0_001100 = −0b0.001100×2^0 = -0.1875	0xcc = 1_1_001100 = −0b1.001100×2^0 = -1.1875
0x0d = 0_0_001101 = +0b0.001101×2^0 = 0.203125	0x4d = 0_1_001101 = +0b1.001101×2^0 = 1.203125	0x8d = 1_0_001101 = −0b0.001101×2^0 = -0.203125	0xcd = 1_1_001101 = −0b1.001101×2^0 = -1.203125
0x0e = 0_0_001110 = +0b0.001110×2^0 = 0.21875	0x4e = 0_1_001110 = +0b1.001110×2^0 = 1.21875	0x8e = 1_0_001110 = −0b0.001110×2^0 = -0.21875	0xce = 1_1_001110 = −0b1.001110×2^0 = -1.21875
0x0f = 0_0_001111 = +0b0.001111×2^0 = 0.234375	0x4f = 0_1_001111 = +0b1.001111×2^0 = 1.234375	0x8f = 1_0_001111 = −0b0.001111×2^0 = -0.234375	0xcf = 1_1_001111 = −0b1.001111×2^0 = -1.234375
0x10 = 0_0_010000 = +0b0.010000×2^0 = 0.25	0x50 = 0_1_010000 = +0b1.010000×2^0 = 1.25	0x90 = 1_0_010000 = −0b0.010000×2^0 = -0.25	0xd0 = 1_1_010000 = −0b1.010000×2^0 = -1.25
0x11 = 0_0_010001 = +0b0.010001×2^0 = 0.265625	0x51 = 0_1_010001 = +0b1.010001×2^0 = 1.265625	0x91 = 1_0_010001 = −0b0.010001×2^0 = -0.265625	0xd1 = 1_1_010001 = −0b1.010001×2^0 = -1.265625
0x12 = 0_0_010010 = +0b0.010010×2^0 = 0.28125	0x52 = 0_1_010010 = +0b1.010010×2^0 = 1.28125	0x92 = 1_0_010010 = −0b0.010010×2^0 = -0.28125	0xd2 = 1_1_010010 = −0b1.010010×2^0 = -1.28125
0x13 = 0_0_010011 = +0b0.010011×2^0 = 0.296875	0x53 = 0_1_010011 = +0b1.010011×2^0 = 1.296875	0x93 = 1_0_010011 = −0b0.010011×2^0 = -0.296875	0xd3 = 1_1_010011 = −0b1.010011×2^0 = -1.296875
0x14 = 0_0_010100 = +0b0.010100×2^0 = 0.3125	0x54 = 0_1_010100 = +0b1.010100×2^0 = 1.3125	0x94 = 1_0_010100 = −0b0.010100×2^0 = -0.3125	0xd4 = 1_1_010100 = −0b1.010100×2^0 = -1.3125
0x15 = 0_0_010101 = +0b0.010101×2^0 = 0.328125	0x55 = 0_1_010101 = +0b1.010101×2^0 = 1.328125	0x95 = 1_0_010101 = −0b0.010101×2^0 = -0.328125	0xd5 = 1_1_010101 = −0b1.010101×2^0 = -1.328125
0x16 = 0_0_010110 = +0b0.010110×2^0 = 0.34375	0x56 = 0_1_010110 = +0b1.010110×2^0 = 1.34375	0x96 = 1_0_010110 = −0b0.010110×2^0 = -0.34375	0xd6 = 1_1_010110 = −0b1.010110×2^0 = -1.34375
0x17 = 0_0_010111 = +0b0.010111×2^0 = 0.359375	0x57 = 0_1_010111 = +0b1.010111×2^0 = 1.359375	0x97 = 1_0_010111 = −0b0.010111×2^0 = -0.359375	0xd7 = 1_1_010111 = −0b1.010111×2^0 = -1.359375
0x18 = 0_0_011000 = +0b0.011000×2^0 = 0.375	0x58 = 0_1_011000 = +0b1.011000×2^0 = 1.375	0x98 = 1_0_011000 = −0b0.011000×2^0 = -0.375	0xd8 = 1_1_011000 = −0b1.011000×2^0 = -1.375
0x19 = 0_0_011001 = +0b0.011001×2^0 = 0.390625	0x59 = 0_1_011001 = +0b1.011001×2^0 = 1.390625	0x99 = 1_0_011001 = −0b0.011001×2^0 = -0.390625	0xd9 = 1_1_011001 = −0b1.011001×2^0 = -1.390625
0x1a = 0_0_011010 = +0b0.011010×2^0 = 0.40625	0x5a = 0_1_011010 = +0b1.011010×2^0 = 1.40625	0x9a = 1_0_011010 = −0b0.011010×2^0 = -0.40625	0xda = 1_1_011010 = −0b1.011010×2^0 = -1.40625
0x1b = 0_0_011011 = +0b0.011011×2^0 = 0.421875	0x5b = 0_1_011011 = +0b1.011011×2^0 = 1.421875	0x9b = 1_0_011011 = −0b0.011011×2^0 = -0.421875	0xdb = 1_1_011011 = −0b1.011011×2^0 = -1.421875
0x1c = 0_0_011100 = +0b0.011100×2^0 = 0.4375	0x5c = 0_1_011100 = +0b1.011100×2^0 = 1.4375	0x9c = 1_0_011100 = −0b0.011100×2^0 = -0.4375	0xdc = 1_1_011100 = −0b1.011100×2^0 = -1.4375
0x1d = 0_0_011101 = +0b0.011101×2^0 = 0.453125	0x5d = 0_1_011101 = +0b1.011101×2^0 = 1.453125	0x9d = 1_0_011101 = −0b0.011101×2^0 = -0.453125	0xdd = 1_1_011101 = −0b1.011101×2^0 = -1.453125
0x1e = 0_0_011110 = +0b0.011110×2^0 = 0.46875	0x5e = 0_1_011110 = +0b1.011110×2^0 = 1.46875	0x9e = 1_0_011110 = −0b0.011110×2^0 = -0.46875	0xde = 1_1_011110 = −0b1.011110×2^0 = -1.46875
0x1f = 0_0_011111 = +0b0.011111×2^0 = 0.484375	0x5f = 0_1_011111 = +0b1.011111×2^0 = 1.484375	0x9f = 1_0_011111 = −0b0.011111×2^0 = -0.484375	0xdf = 1_1_011111 = −0b1.011111×2^0 = -1.484375
0x20 = 0_0_100000 = +0b0.100000×2^0 = 0.5	0x60 = 0_1_100000 = +0b1.100000×2^0 = 1.5	0xa0 = 1_0_100000 = −0b0.100000×2^0 = -0.5	0xe0 = 1_1_100000 = −0b1.100000×2^0 = -1.5
0x21 = 0_0_100001 = +0b0.100001×2^0 = 0.515625	0x61 = 0_1_100001 = +0b1.100001×2^0 = 1.515625	0xa1 = 1_0_100001 = −0b0.100001×2^0 = -0.515625	0xe1 = 1_1_100001 = −0b1.100001×2^0 = -1.515625
0x22 = 0_0_100010 = +0b0.100010×2^0 = 0.53125	0x62 = 0_1_100010 = +0b1.100010×2^0 = 1.53125	0xa2 = 1_0_100010 = −0b0.100010×2^0 = -0.53125	0xe2 = 1_1_100010 = −0b1.100010×2^0 = -1.53125
0x23 = 0_0_100011 = +0b0.100011×2^0 = 0.546875	0x63 = 0_1_100011 = +0b1.100011×2^0 = 1.546875	0xa3 = 1_0_100011 = −0b0.100011×2^0 = -0.546875	0xe3 = 1_1_100011 = −0b1.100011×2^0 = -1.546875
0x24 = 0_0_100100 = +0b0.100100×2^0 = 0.5625	0x64 = 0_1_100100 = +0b1.100100×2^0 = 1.5625	0xa4 = 1_0_100100 = −0b0.100100×2^0 = -0.5625	0xe4 = 1_1_100100 = −0b1.100100×2^0 = -1.5625
0x25 = 0_0_100101 = +0b0.100101×2^0 = 0.578125	0x65 = 0_1_100101 = +0b1.100101×2^0 = 1.578125	0xa5 = 1_0_100101 = −0b0.100101×2^0 = -0.578125	0xe5 = 1_1_100101 = −0b1.100101×2^0 = -1.578125
0x26 = 0_0_100110 = +0b0.100110×2^0 = 0.59375	0x66 = 0_1_100110 = +0b1.100110×2^0 = 1.59375	0xa6 = 1_0_100110 = −0b0.100110×2^0 = -0.59375	0xe6 = 1_1_100110 = −0b1.100110×2^0 = -1.59375
0x27 = 0_0_100111 = +0b0.100111×2^0 = 0.609375	0x67 = 0_1_100111 = +0b1.100111×2^0 = 1.609375	0xa7 = 1_0_100111 = −0b0.100111×2^0 = -0.609375	0xe7 = 1_1_100111 = −0b1.100111×2^0 = -1.609375
0x28 = 0_0_101000 = +0b0.101000×2^0 = 0.625	0x68 = 0_1_101000 = +0b1.101000×2^0 = 1.625	0xa8 = 1_0_101000 = −0b0.101000×2^0 = -0.625	0xe8 = 1_1_101000 = −0b1.101000×2^0 = -1.625
0x29 = 0_0_101001 = +0b0.101001×2^0 = 0.640625	0x69 = 0_1_101001 = +0b1.101001×2^0 = 1.640625	0xa9 = 1_0_101001 = −0b0.101001×2^0 = -0.640625	0xe9 = 1_1_101001 = −0b1.101001×2^0 = -1.640625
0x2a = 0_0_101010 = +0b0.101010×2^0 = 0.65625	0x6a = 0_1_101010 = +0b1.101010×2^0 = 1.65625	0xaa = 1_0_101010 = −0b0.101010×2^0 = -0.65625	0xea = 1_1_101010 = −0b1.101010×2^0 = -1.65625
0x2b = 0_0_101011 = +0b0.101011×2^0 = 0.671875	0x6b = 0_1_101011 = +0b1.101011×2^0 = 1.671875	0xab = 1_0_101011 = −0b0.101011×2^0 = -0.671875	0xeb = 1_1_101011 = −0b1.101011×2^0 = -1.671875
0x2c = 0_0_101100 = +0b0.101100×2^0 = 0.6875	0x6c = 0_1_101100 = +0b1.101100×2^0 = 1.6875	0xac = 1_0_101100 = −0b0.101100×2^0 = -0.6875	0xec = 1_1_101100 = −0b1.101100×2^0 = -1.6875
0x2d = 0_0_101101 = +0b0.101101×2^0 = 0.703125	0x6d = 0_1_101101 = +0b1.101101×2^0 = 1.703125	0xad = 1_0_101101 = −0b0.101101×2^0 = -0.703125	0xed = 1_1_101101 = −0b1.101101×2^0 = -1.703125
0x2e = 0_0_101110 = +0b0.101110×2^0 = 0.71875	0x6e = 0_1_101110 = +0b1.101110×2^0 = 1.71875	0xae = 1_0_101110 = −0b0.101110×2^0 = -0.71875	0xee = 1_1_101110 = −0b1.101110×2^0 = -1.71875
0x2f = 0_0_101111 = +0b0.101111×2^0 = 0.734375	0x6f = 0_1_101111 = +0b1.101111×2^0 = 1.734375	0xaf = 1_0_101111 = −0b0.101111×2^0 = -0.734375	0xef = 1_1_101111 = −0b1.101111×2^0 = -1.734375
0x30 = 0_0_110000 = +0b0.110000×2^0 = 0.75	0x70 = 0_1_110000 = +0b1.110000×2^0 = 1.75	0xb0 = 1_0_110000 = −0b0.110000×2^0 = -0.75	0xf0 = 1_1_110000 = −0b1.110000×2^0 = -1.75
0x31 = 0_0_110001 = +0b0.110001×2^0 = 0.765625	0x71 = 0_1_110001 = +0b1.110001×2^0 = 1.765625	0xb1 = 1_0_110001 = −0b0.110001×2^0 = -0.765625	0xf1 = 1_1_110001 = −0b1.110001×2^0 = -1.765625
0x32 = 0_0_110010 = +0b0.110010×2^0 = 0.78125	0x72 = 0_1_110010 = +0b1.110010×2^0 = 1.78125	0xb2 = 1_0_110010 = −0b0.110010×2^0 = -0.78125	0xf2 = 1_1_110010 = −0b1.110010×2^0 = -1.78125
0x33 = 0_0_110011 = +0b0.110011×2^0 = 0.796875	0x73 = 0_1_110011 = +0b1.110011×2^0 = 1.796875	0xb3 = 1_0_110011 = −0b0.110011×2^0 = -0.796875	0xf3 = 1_1_110011 = −0b1.110011×2^0 = -1.796875
0x34 = 0_0_110100 = +0b0.110100×2^0 = 0.8125	0x74 = 0_1_110100 = +0b1.110100×2^0 = 1.8125	0xb4 = 1_0_110100 = −0b0.110100×2^0 = -0.8125	0xf4 = 1_1_110100 = −0b1.110100×2^0 = -1.8125
0x35 = 0_0_110101 = +0b0.110101×2^0 = 0.828125	0x75 = 0_1_110101 = +0b1.110101×2^0 = 1.828125	0xb5 = 1_0_110101 = −0b0.110101×2^0 = -0.828125	0xf5 = 1_1_110101 = −0b1.110101×2^0 = -1.828125
0x36 = 0_0_110110 = +0b0.110110×2^0 = 0.84375	0x76 = 0_1_110110 = +0b1.110110×2^0 = 1.84375	0xb6 = 1_0_110110 = −0b0.110110×2^0 = -0.84375	0xf6 = 1_1_110110 = −0b1.110110×2^0 = -1.84375
0x37 = 0_0_110111 = +0b0.110111×2^0 = 0.859375	0x77 = 0_1_110111 = +0b1.110111×2^0 = 1.859375	0xb7 = 1_0_110111 = −0b0.110111×2^0 = -0.859375	0xf7 = 1_1_110111 = −0b1.110111×2^0 = -1.859375
0x38 = 0_0_111000 = +0b0.111000×2^0 = 0.875	0x78 = 0_1_111000 = +0b1.111000×2^0 = 1.875	0xb8 = 1_0_111000 = −0b0.111000×2^0 = -0.875	0xf8 = 1_1_111000 = −0b1.111000×2^0 = -1.875
0x39 = 0_0_111001 = +0b0.111001×2^0 = 0.890625	0x79 = 0_1_111001 = +0b1.111001×2^0 = 1.890625	0xb9 = 1_0_111001 = −0b0.111001×2^0 = -0.890625	0xf9 = 1_1_111001 = −0b1.111001×2^0 = -1.890625
0x3a = 0_0_111010 = +0b0.111010×2^0 = 0.90625	0x7a = 0_1_111010 = +0b1.111010×2^0 = 1.90625	0xba = 1_0_111010 = −0b0.111010×2^0 = -0.90625	0xfa = 1_1_111010 = −0b1.111010×2^0 = -1.90625
0x3b = 0_0_111011 = +0b0.111011×2^0 = 0.921875	0x7b = 0_1_111011 = +0b1.111011×2^0 = 1.921875	0xbb = 1_0_111011 = −0b0.111011×2^0 = -0.921875	0xfb = 1_1_111011 = −0b1.111011×2^0 = -1.921875
0x3c = 0_0_111100 = +0b0.111100×2^0 = 0.9375	0x7c = 0_1_111100 = +0b1.111100×2^0 = 1.9375	0xbc = 1_0_111100 = −0b0.111100×2^0 = -0.9375	0xfc = 1_1_111100 = −0b1.111100×2^0 = -1.9375
0x3d = 0_0_111101 = +0b0.111101×2^0 = 0.953125	0x7d = 0_1_111101 = +0b1.111101×2^0 = 1.953125	0xbd = 1_0_111101 = −0b0.111101×2^0 = -0.953125	0xfd = 1_1_111101 = −0b1.111101×2^0 = -1.953125
0x3e = 0_0_111110 = +0b0.111110×2^0 = 0.96875	0x7e = 0_1_111110 = +0b1.111110×2^0 = 1.96875	0xbe = 1_0_111110 = −0b0.111110×2^0 = -0.96875	0xfe = 1_1_111110 = −0b1.111110×2^0 = -1.96875
0x3f = 0_0_111111 = +0b0.111111×2^0 = 0.984375	0x7f = 0_1_111111 = +Inf	0xbf = 1_0_111111 = −0b0.111111×2^0 = -0.984375	0xff = 1_1_111111 = −Inf

References

[1] W. Kahan, "Branch cuts for complex elementary functions or much ado about nothing's sign bit," *Institute of Mathematics and its Applications Conference*, 1987.
https://people.freebsd.org/~das/kahan86branch.pdf.

[2] W. Kahan and J. W. Thomas, "Augmenting a programming language with complex arithmetic," tech. rep., EECS Department, University of California, Berkeley, 1991.

[3] Google, "Jax lax package: _float_to_int_for_sort ."
https://github.com/google/jax/blob/fc5960f2b8b7a0ef74dbae4e27c5c08ff1564cff/jax/_src/lax/lax.py#L3934.

[4] B. Noune, P. Jones, D. Justus, D. Masters, and C. Luschi, "Adaptive loss scaling for mixed precision training," tech. rep., arXiv cs.LG, 2019.
https://arxiv.org/abs/1910.12385.

[5] PyTorch authors, "Pytorch torchtext package: _t5_multi_head_attention_forward ."
https://github.com/pytorch/text/blob/a933cbe5a008bc2cb61d985cf5864069194157eb/torchtext/prototype/models/t5/modules.py#L236.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ch. 6.2.2.3 Softmax Units for Multinoulli Output Distributions, pp. 180–184. MIT Press, 2016.

[7] P. Micikevicius, S. Oberman, P. Dubey, M. Cornea, A. Rodriguez, I. Bratt, R. Grisenthwaite, N. Jouppi, C. Chou, A. Huffman, M. Schulte, R. Wittig, D. Jani, and S. Deng, "OCP 8-bit floating point specification (OFP8)," tech. rep., opencompute.org, 2023.
https://www.opencompute.org/documents/ocp-8-bit-floating-point-specification-ofp8-revision-1-0-2(

[8] B. Noune, P. Jones, D. Justus, D. Masters, and C. Luschi, "8-bit numerical formats for deep neural networks," tech. rep., arXiv cs.LG, 2022.
https://arxiv.org/abs/2206.02915.

[9] Tesla, Inc., "Tesla Dojo Technology: A guide to Tesla's configurable floating point formats and arithmetic," 2023.
https://web.archive.org/web/20230503235751/https://tesla-cdn.thron.com/static/MXMU3S_tesla-dojo-technology_1WDVZN.pdf.