Multiple-base Logarithmic Quantization and Application in Reduced Precision AI Computations

Vassil Dimitrov[1,2] Richard Ford[1] Laurent Imbert[1,3]
Arjuna Madanayake[1] Nilan Udayanga[1] Will Wray[1]

[1]Lemurian Labs, Oakville, Canada

[2]University of Calgary, Canada

[3]CNRS, LIRMM, University of Montpellier, France

ARITH 2024

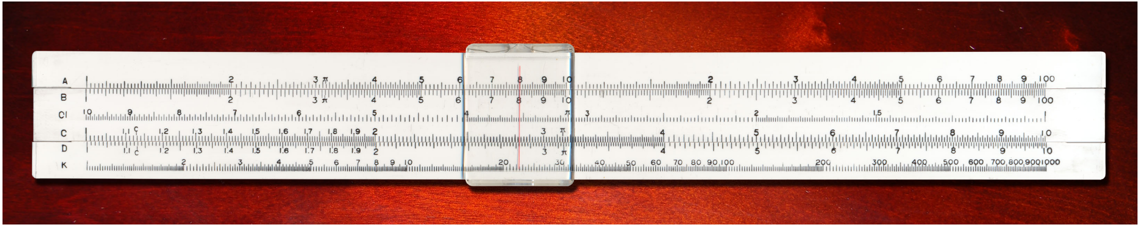
Málaga, June 10-12

What the paper is all about

- A brief history of the logarithmic representations
- Generalization of the logarithmic number systems into multidimensional representation
- The usefulness of logarithmic quantizations in reduced precision computations
- QSNR experimental results
- FPGA designs for dot-product engine
- Conclusions

A brief history of logarithmic representations

- Logarithmic number system (LNS) is just a digital version of the slide rule!
- Initially used to simplify multiplications and divisions
- A large body of literature on the use of LNS in DSP in mid 70s and 80s
- Since 2016 - many articles and patents on the use of LNS for reduced precision ML computations



LNS representations

In LNS, real numbers are represented by the logarithm in base 2 of their absolute values

$$x \in \mathbb{R} \longrightarrow \begin{array}{c} (s, e) \\ s = \pm 1,\ e = \log_2 |x| \\ \Rightarrow x = s \cdot 2^e \end{array}$$

In the original definition, e is written in signed fixed-point representation.

LNS in DSP (1970s – 2010s)

- LNS for digital filtering
- DSP transforms
- some other applications

New era: LNS in machine learning applications (2026 – today)

- Mayashita et al. (2016): LNS for reduced precision ML computations.

  point out that LNS with base $\sqrt{2}$ seems more appropriate than LNS with base 2.

MDLNS: Multi-Dimensional Logarithmic Number Systems

Defined using 3 finite sequences:

- $R = (\beta_1, \ldots, \beta_k) \in (\mathbb{R}_{>0})^k$: **MDLNS bases** (rationally independent real numbers)
- $W = (w_1, \ldots, w_k) \in \mathbb{N}^k$ **exponent bit-lengths**
- $B = (b_1, \ldots, b_k) \in \mathbb{Z}^k$ **exponent biases**

The total bit-length of the MDLNS reprensentation is $n = 1 + \sum_{i=1}^{k} w_i$

Then, $\text{MDLNS}_n(R, W, B)$ is the finite set of real numbers of size 2^n given by:

$$\text{MDLNS}_n = \left\{ \pm \prod_{i=1}^{k} \beta_i^{e_i};\ 0 \leq e_i + b_i < 2^{w_i} \right\}$$

The exponents $e_i \in \mathbb{Z}$ have bit-length w_i respectively and are biased with bias b_i, i.e. the unsigned binary encoded value $\hat{e}_i$ corresponds to the integer $e_i = \hat{e}_i - b_i$.

Example of MDLNS representation

Let:

- $R = (2, 3)$
- $W = (2, 2)$ bit-length of the representation: $n = 1 + 2 + 2 = 5$
- $B = (2, 2)$ i.e. the exponents are encoded using two's complement notation

Then $\text{MDLNS}_5 = \{\pm 2^a 3^b, -2 \leq a, b \leq 1\}$

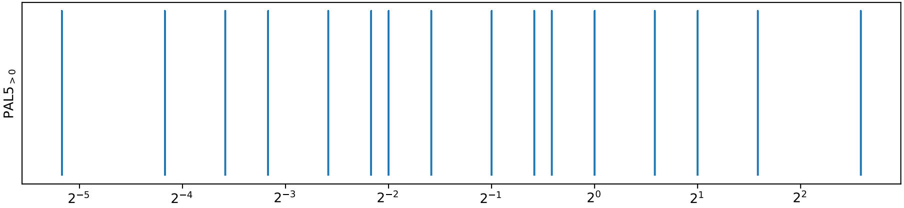
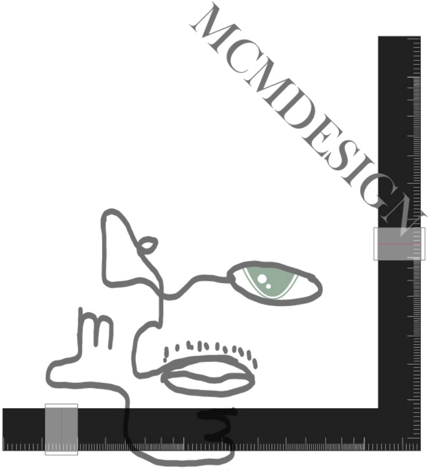
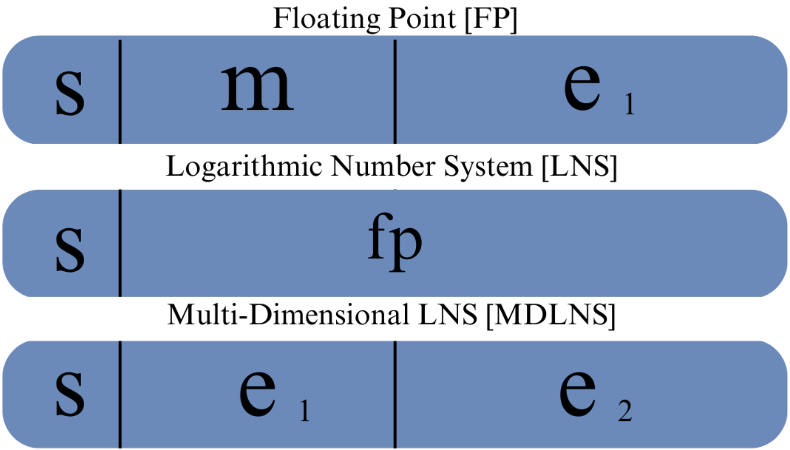


Figure: The 16 positive real values from MDLNS_5 (on a log scale)

Multi-Dimensional Slide Rule



Pictorial representation - floating point, logarithmic and MDLNS



Connections with DBNS (for ARITH aficionados)

- ARITH-1997: "Theory and Applications of the Double-Base Number System"
- ARITH-2001: "The use of multidimensional logarithmic number system in DSP applications"
- ARITH 2007: "Multiplication by a constant is sublinear" (main theorem uses DBNS)

Alternative MDLNS in the literature

- Dual-logarithmic (Jeff Johnson ARITH 2020)
- multi-base LNS NVIDIA (IEEE Trans on Computers 2023)
- Logarithmic Posits (`https://arxiv.org/abs/2403.05465`)

MDLNS as a quantization engine

Quantization: the process of mapping an infinite set of continuous values to a finite set of discrete values.

Very popular for accelerating inference and for reducing memory/power consumption in DNN

MDLNS is particularly suited for quantization since:

- We can choose any bases
  1. Adapt to any model/layer distribution
  2. Can likely find base to beat out FP, LNS, ...
- Can easily scale from 16bit to 4,6,8...
- Multiplication maps to addition
- Great dynamic range with precision around 0

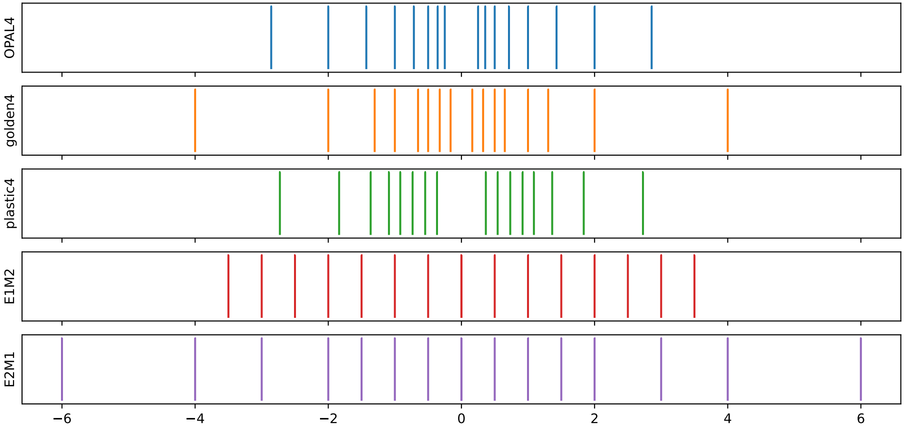


Figure: Distributions of values for various $MDLNS_4$ and FP4 formats

Numerical fidelity of a quantization scheme

Quantization signal to noise ratio [Rouhani et. al. 2023]

Ratio of the power of the non-quantized signal $X = (x_1, x_2, \ldots, x_k) \in \mathbb{R}^k$ to the power of the quantization noise expressed in decibels

$$\mathrm{QSNR} := -10 \log_{10} \left(\frac{E\left[\|Q(X) - X\|^2\right]}{E\left[\|X\|^2\right]} \right)$$

$\|.\|$ denotes the L_2 norm

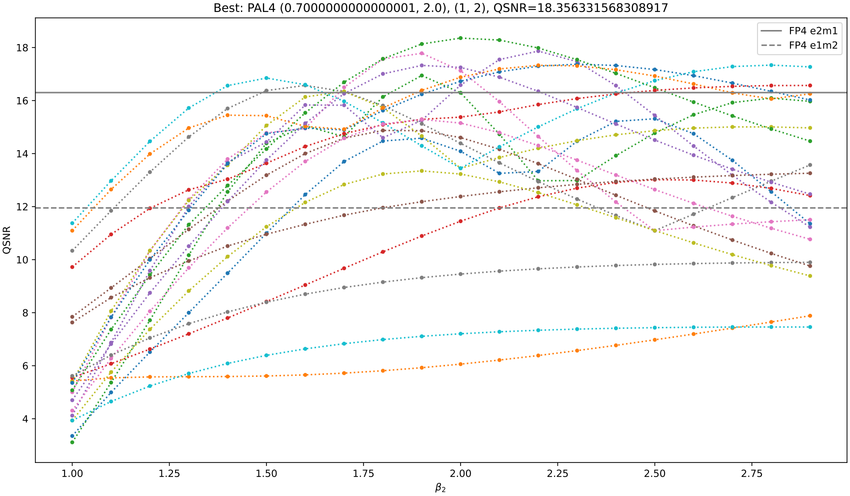
Parameters of our MDLNS_5 example

	$\mathrm{MDLNS}_5((2,3),(2,2))$
Bases	[2, 3]
Exponent sizes	[2, 2]
Exponent biases	[2, 2]
Min. pos. value	0.02777778
Max. pos. value	6.00000000
DNR(*)	7.75488750
QSNR	19.67874706

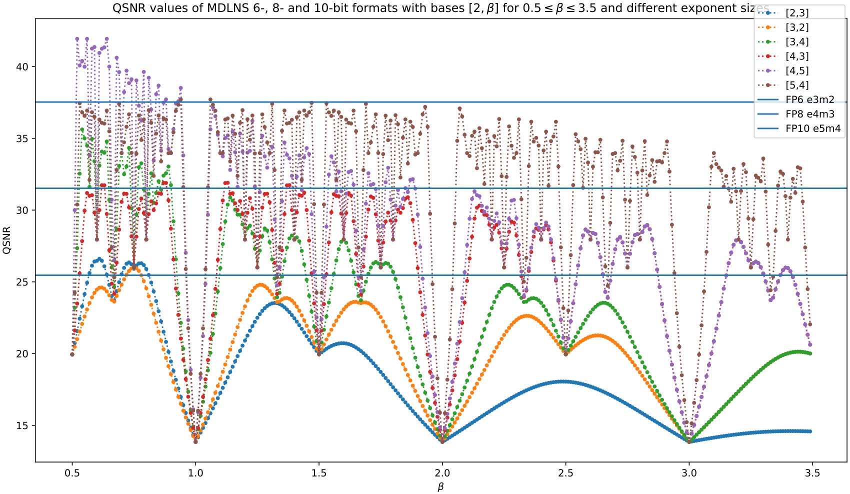
Table: $\mathrm{MDLNS}_5((2,3),(2,2))$ parameters

(*) the dynamic range (DNR) of a finite set of strictly positive real numbers is defined as the logarithm in base 2 of the ratio between the largest and the smallest values from that set.

Comparisons with floating point in terms of QSNR (MDLNS_4)

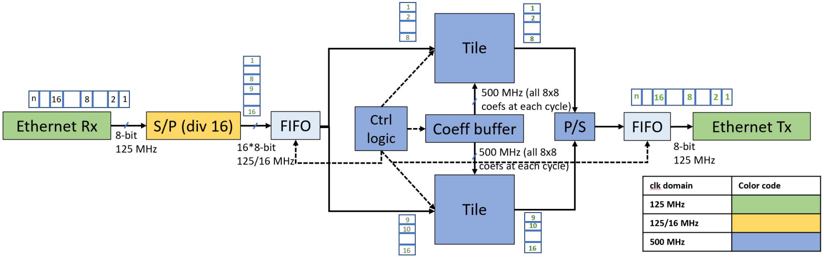


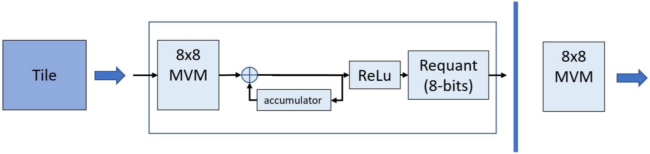
Comparisons with floating point in terms of QSNR ($\text{MDLNS}_{6,8,10}$)



MDLNS matrix-vector multipliers (FPGA)

Demo Architecture for 8x8 Matrix Vector Multiplier (MVM)





*No of channels accumulate inside the tile = tile_clk*noOfTiles*8/ethernet_clk = Coeff buffer depth = number of coefs applied per 8x1 input

Comparisons between fixed-point and MDLNS implementation results

	Fixed-point	MDLNS
Configurable logic blocks	35 659 CLBs	28 813 CLBs
Static power	3.53 W	3.22 W
Dynamic power	3.2 W	4.41 W
Maximum clock rate	312 MHz	555 MHz
Throughput	47.4 Gops/W	74.4 Gops/W
AT^2	0.37	0.09

Possible generalizations

- Non-integers exponents
- MDLNS with more than 2 bases
- Complex and hypercomplex MDLNS

Open problems and directions for future research

- MDLNS arithmetic: The biggest challenge - MDLNS addition and subtraction
- Efficient conversion from float to MDLNS and back
- MDLNS for complex arithmetic
- MDLNS for image processing
- Theoretical problems

Thank you for your attention!

`https://www.lemurianlabs.com/`

MDLNS can be seen as a LNS

A MDLNS is a LNS where the encoding of the exponent can be written as linear forms of logarithms.

For all $\beta_i \in B$, it is always possible to write $\beta_i = \exp(\log(\beta_i))$ so that:

$$\mathrm{MDLNS}_n = \left\{ \pm \exp\left(\sum_{i=1}^{k} e_i \log(\beta_i) \right) \; ; \; 0 \leq e_i + b_i < 2^{w_i} \right\}$$

MDLNS block quantization