We need to talk about arithmetic

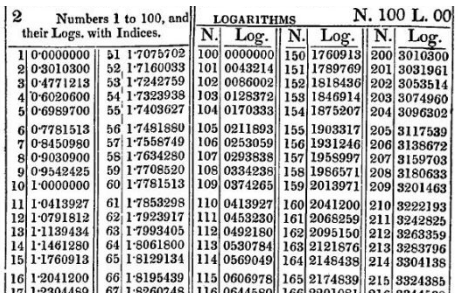
Tim Fernandez-Hart

Brunel University of London

Katherine Johnson

1918 - 2020

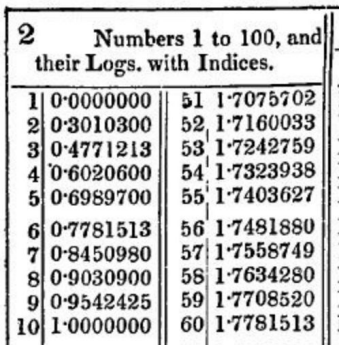
How best to represent a range of continuous values using a finite set of discrete numbers?



2 Numbers 1 to 100, and their Logs. with Indices.

LOGARITHMS N. 100 L. 00

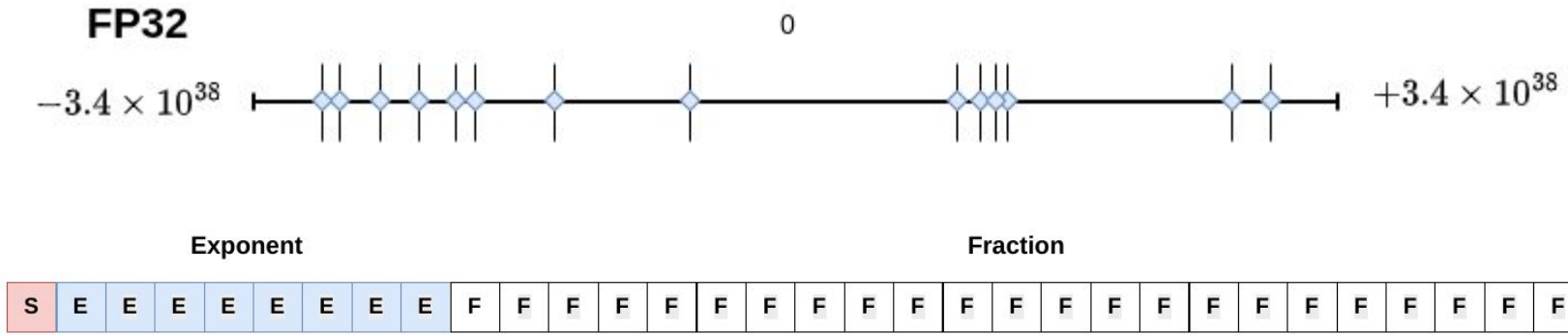
N.	Log.	N.	Log.	N.	Log.
100	0000000	150	1760913	200	3010300
101	0043214	151	1789769	201	3031961
102	0086002	152	1818436	202	3053514
103	0128372	153	1846914	203	3074960
104	0170333	154	1875207	204	3096302
105	0211893	155	1903317	205	3117539
106	0253059	156	1931246	206	3138672
107	0293838	157	1958997	207	3159703
108	0334238	158	1986571	208	3180633
109	0374265	159	2013971	209	3201463
110	0413927	160	2041200	210	3222193
111	0453230	161	2068259	211	3242825
112	0492180	162	2095150	212	3263359
113	0530784	163	2121876	213	3283796
114	0569049	164	2148438	214	3304138
115	0606978	165	2174839	215	3324385



2 Numbers 1 to 100, and their Logs. with Indices.

1	0·0000000	51	1·7075702
2	0·3010300	52	1·7160033
3	0·4771213	53	1·7242759
4	0·6020600	54	1·7323938
5	0·6989700	55	1·7403627
6	0·7781513	56	1·7481880
7	0·8450980	57	1·7558749
8	0·9030900	58	1·7634280
9	0·9542425	59	1·7708520
10	1·0000000	60	1·7781513

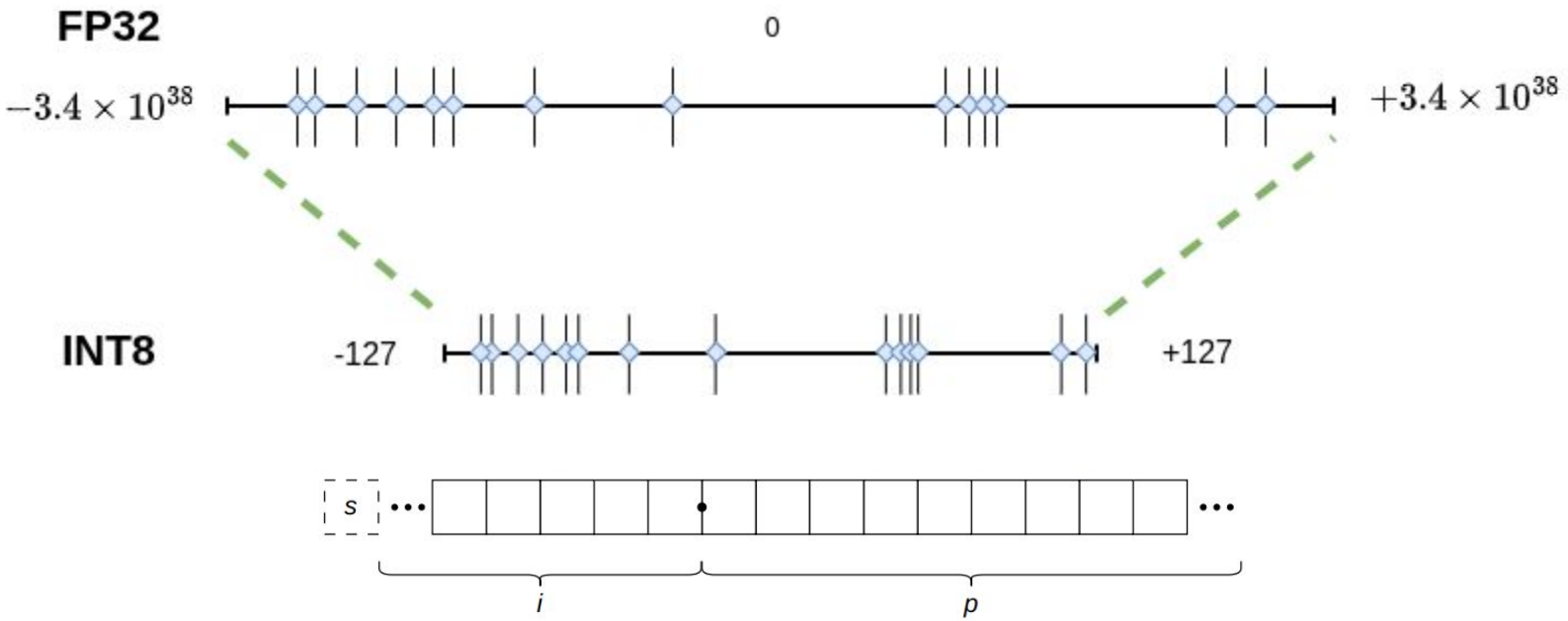
How best to represent a range of continuous values using a finite set of discrete numbers?



Neuromorphic arithmetic...

Year	Device	Arithmeic	Width
2011	SpiNNaker - 1	Integer-based	32-bit
2014	TrueNorth	Integer-based	
2018	Loihi - 1	Integer-based	Configurable weight 1, 2, 4, or 8-bits State Vars. 24-bits
2020	Tianjic	Integer-based	Weights: 8-bit integer State Vars. : 24 bits
2021	Akida	Integer-based	Weights: 1, 2, 4, 8-bits
2021	Loihi - 2	Integer-based	32-bit fixed-point Weights: variable 1 to 32-bit
2022	Speck	Integer-based	Weights: 8-bit integer State Vars. : 24 bits
2024	SpiNNaker - 2	Integer-based FPU	32-bit FPU
2025	PAICORE	Integer-based	configurable weight 1, 2, 4, or 8-bits input: 8-bits State Vars. 30-bits
2025	FeNN	Integer-based	16-bit

Quantisation



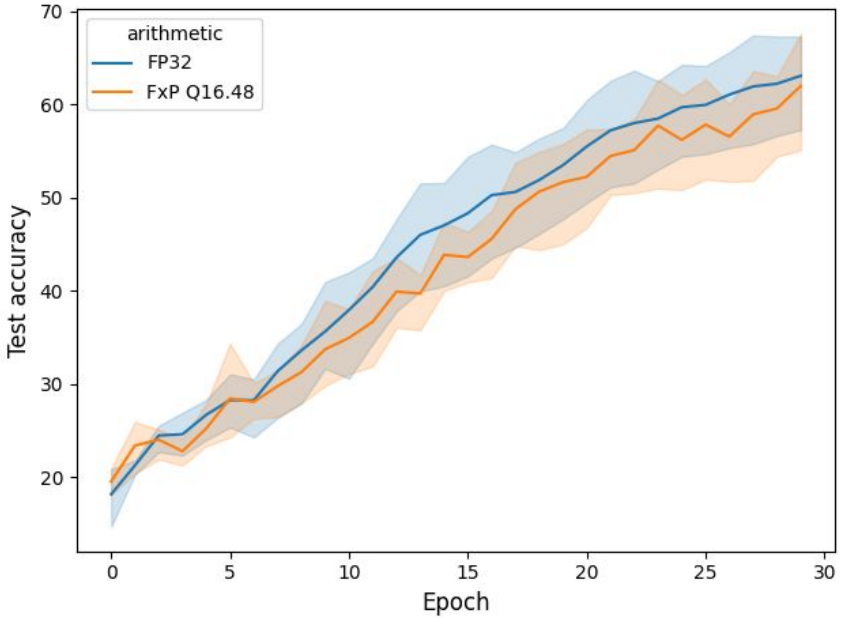
Why have they done that?

Operation	Integer		Floating Point	
	16-bit	32-bit	16-bit	32-bit
Addition	0.05 pJ	0.1 pJ	0.4 pJ	0.9 pJ
	67 μm^2	137μm^2	1360μm^2	4184 μm^2
Multiplication	-	3.1 pJ	1.1 pJ	3.7 pJ
	-	3495 μm^2	1640 μm^2	7700 μm^2

9x energy
30x area
2x area

Table 2: Energy and area cost of addition and multiplication for FxP and FP for 16, and 32-bit adapted from Horowitz [5] and Gholami et al. [6]

But FP32 far more capable



- E-prop online-training SHD
- Recurrent network:
  - 700-256-20
- Functional equivalence:
  - FP32 = 64-bit Fixed-point Q(16.48)

Is Fixed-Point *really* less efficient?

Operation	Integer		Floating Point	
	16-bit	32-bit	16-bit	32-bit
Addition	0.05 pJ	0.1 pJ	0.4 pJ	0.9 pJ
	67 μm^2	137μm^2	1360μm^2	4184 μm^2
Multiplication	-	3.1 pJ	1.1 pJ	3.7 pJ
	-	3495 μm^2	1640 μm^2	7700 μm^2

4x energy
10x area
1/3 energy
1/2 area

Table 2: Energy and area cost of addition and multiplication for FxP and FP for 16, and 32-bit adapted from Horowitz [5] and Gholami et al. [6]

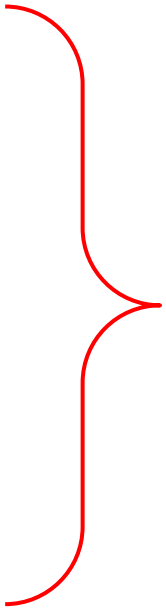
1. M. Horowitz, "**1.1 computing's energy problem (and what we can do about it),**" in 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), Feb. 2014, pp. 10–14. doi: 10.1109/ISSCC.2014.6757323.
2. A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, **A survey of quantization methods for efficient neural network inference,** Jun. 21, 2021. doi: 10.48550/arXiv.2103.13630. arXiv: 2103.13630[cs].

Mitigation techniques

- Model simplifications:
  - E.g. swap ALIF for LIF
  - No recurrent connections
- Training simplifications
  - Avoid softmax (lose competitive winner-takes-all dynamic)
- Mixed precision
  - Weights can often be smaller bit-widths,
  - Eligibility traces/ gradients need to be longer bit-widths
- QAT:
  - Inefficient
  - May not have dataset

Mitigation techniques

- Model simplifications:
  - E.g. swap ALIF for LIF
  - No recurrent connections
- Training simplifications
  - Avoid softmax (lose competitive winner-takes-all dynamic)
- Mixed precision
  - Weights can often be smaller bit-widths,
  - Eligibility traces/ gradients need to be longer bit-widths
- QAT:
  - Inefficient
  - May not have dataset



Researcher time & effort

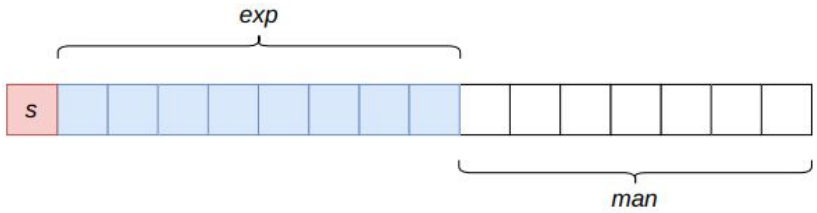
FP…… It’s not the bee’s knees

- 2 versions of zero
- Different permitted rounding modes mean results can differ between machines
- Many bit-patterns reserved for NaN

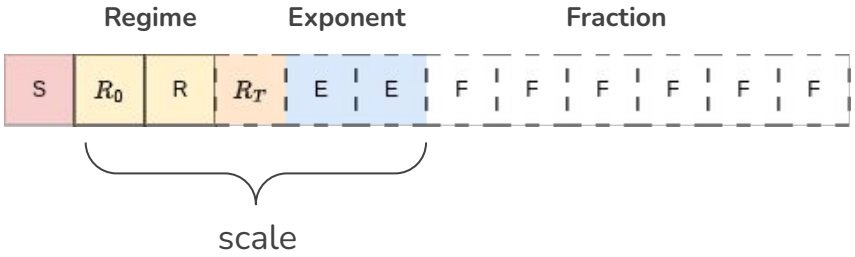
Format	Smallest positive	Range	# NaN
FP8 (e=4, m=3)	*0.015625*	*±240*	*7*
FP8 (e=3, m=4)	*0.25*	*±15.5*	*15*
FP8 (e=5, m=2)	6.1×10^{-5}	*±57344*	*3*
FP16 - Half-precision (e=5, m=10)	6.1×10^{-5}	±65504	1023
FP32 - Single-precision (e=8, m=23)	1.18×10^{-38}	3.4×10^{38}	8,388,607
FP64 - Double-precision (e=11, m=53)	2.23×10^{-308}	1.798×10^{308}	9,007,199,254,740,991

Arithmetic innovation

bfloat16



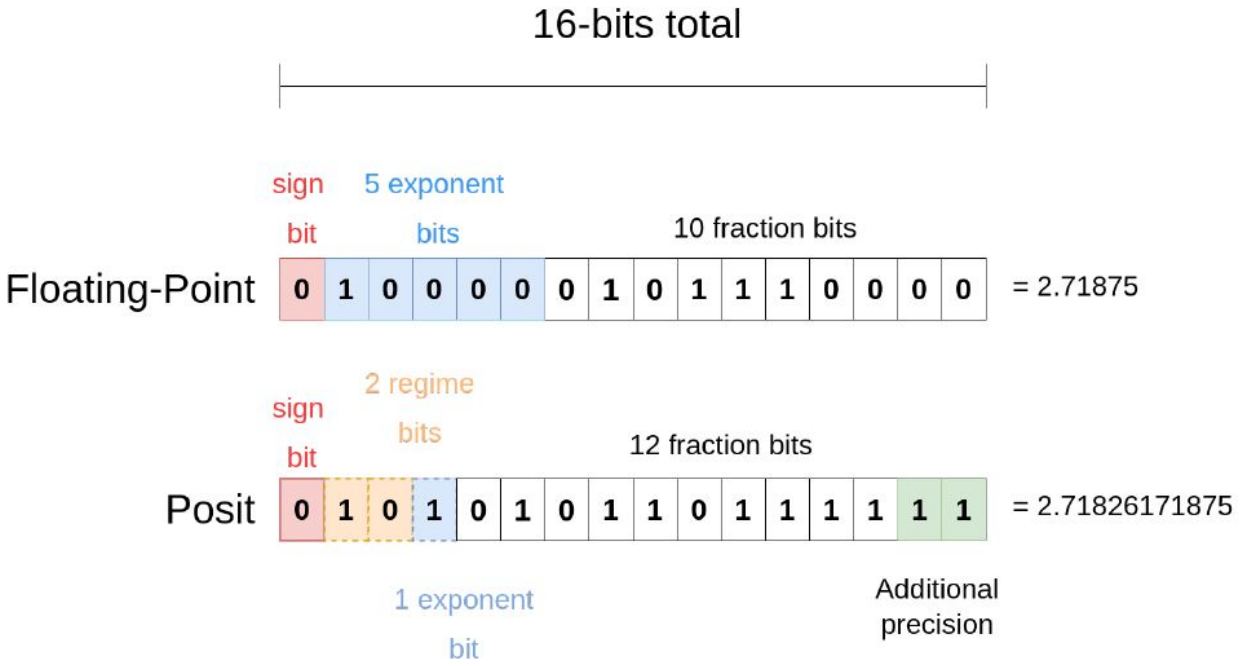
posit<n, es>



Many others:

- TensorFloat-32 (Nvidia)
- Takum arithmetic
- Logarithmic arithmetic
- Morris floats
- Flexpoint (Intel)
- Valids (interval arithmetic)

Posit decoding



scale

$$fp = (-1)^s \times 2^{e-bias} \times \left(1 + \frac{fraction}{2^{10}}\right)$$

$$fp = (-1)^0 \times 2^{16-15} \times \left(1 + \frac{368}{2^{10}}\right)$$

$$= 2.71875$$

scale

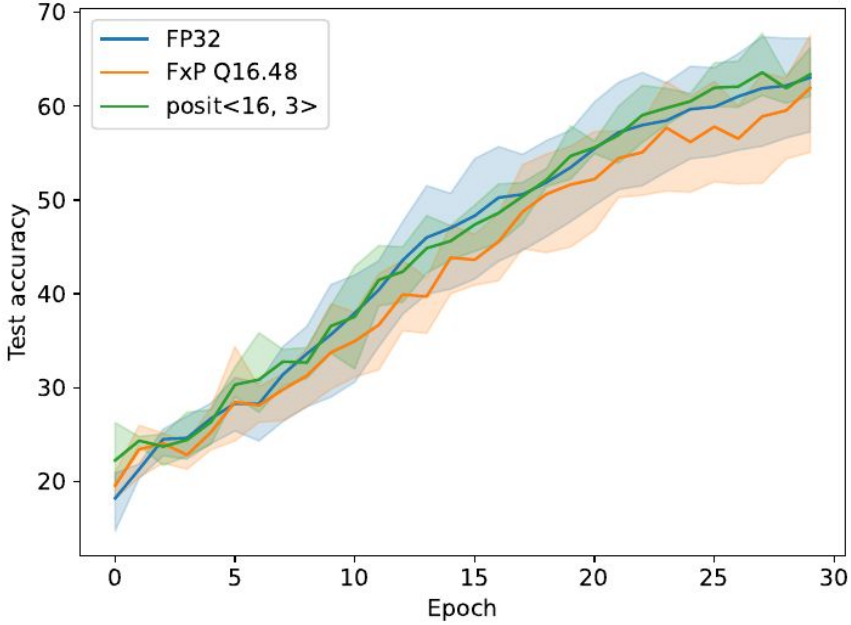
$$p = (-1)^s \text{useed}^r 2^e \left(1 + \frac{f}{2^{\text{fs}}}\right)$$

$$p = (-1)^0 \times 4^0 \times 2^1 \times \left(1 + \frac{1471}{2^{12}}\right)$$

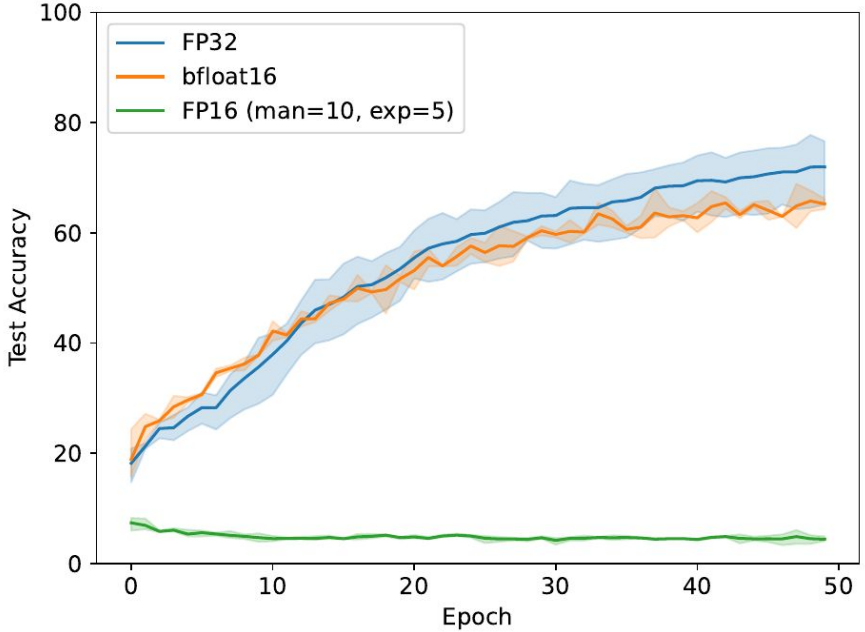
$$= 2.71826171875$$

How about posits?

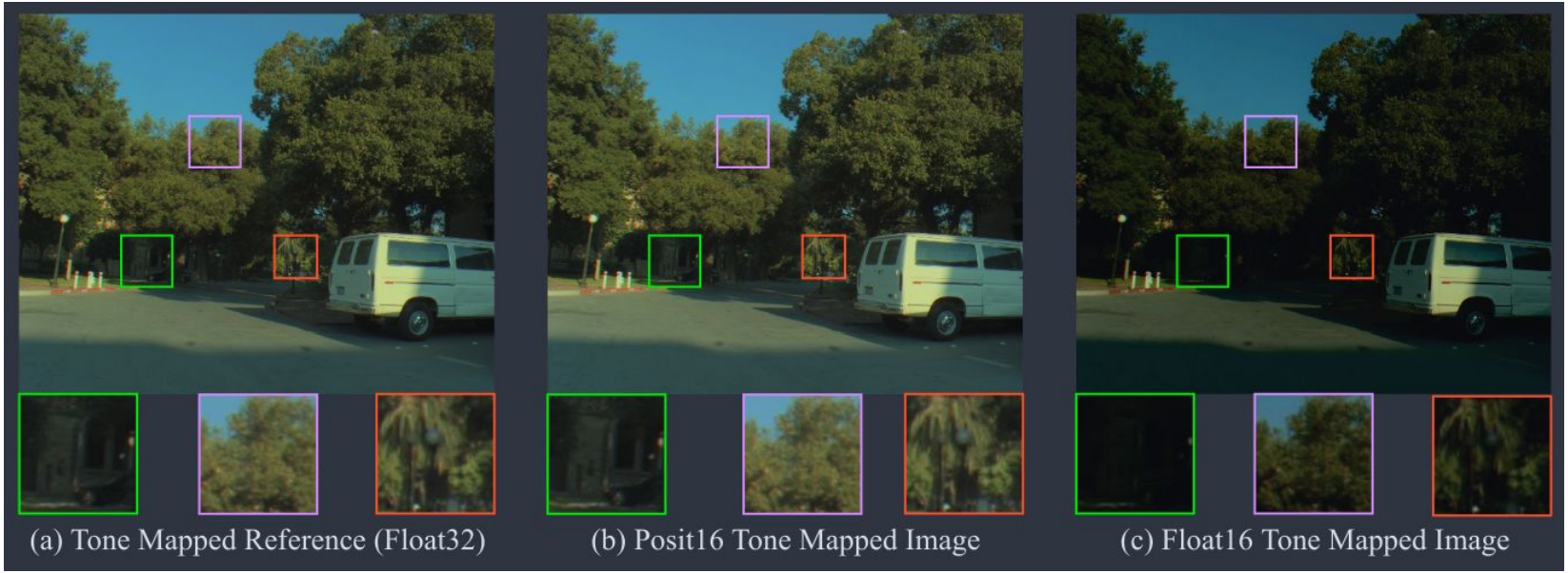
Posit eProp:



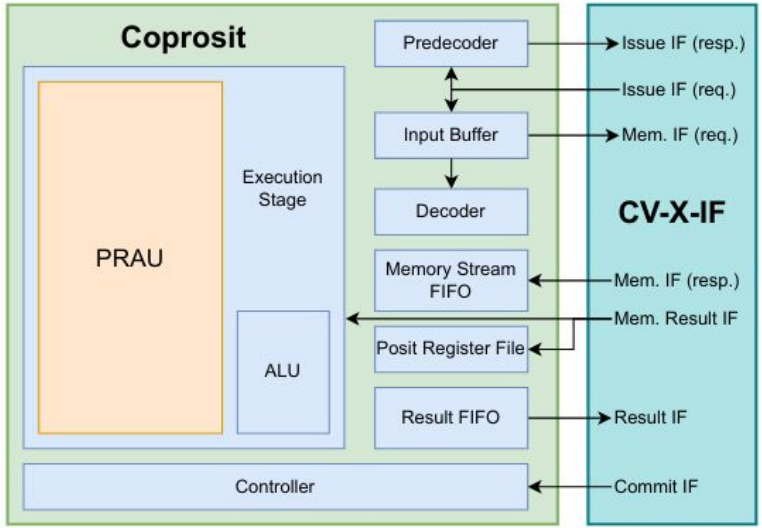
Bfloat16 eProp:

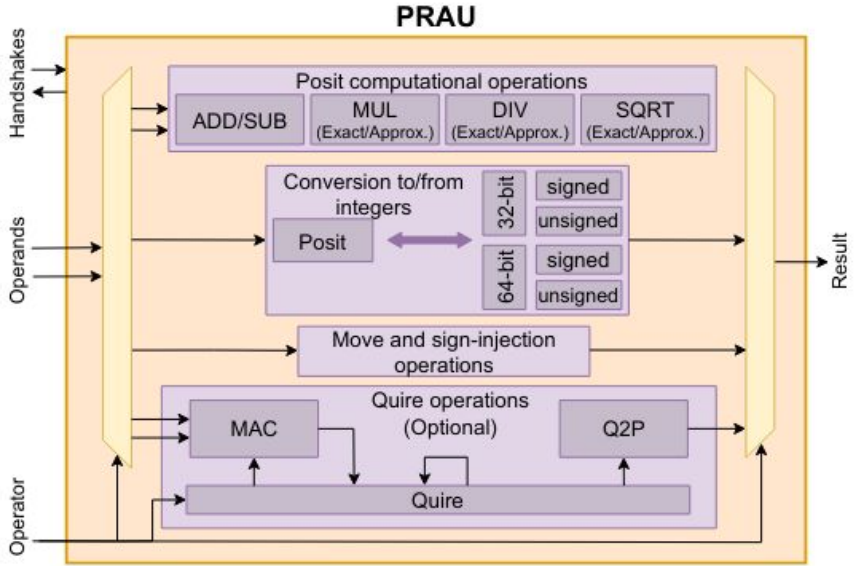


PositIV: HDR


(a) Tone Mapped Reference (Float32)
(b) Posit16 Tone Mapped Image
(c) Float16 Tone Mapped Image

16-bit posit Vs. 32-bit FP

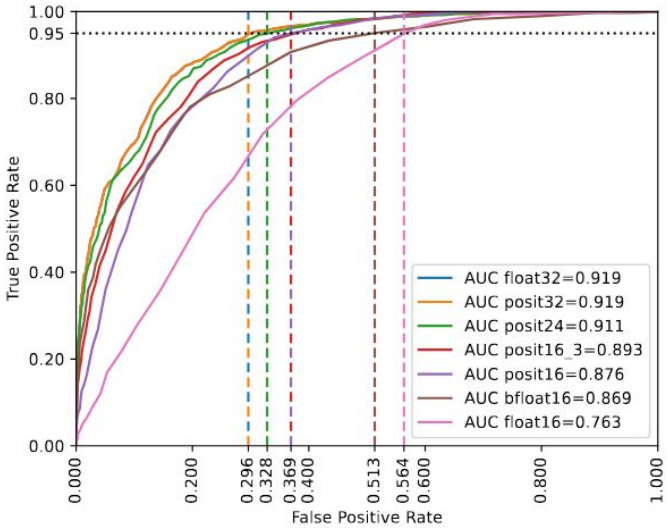




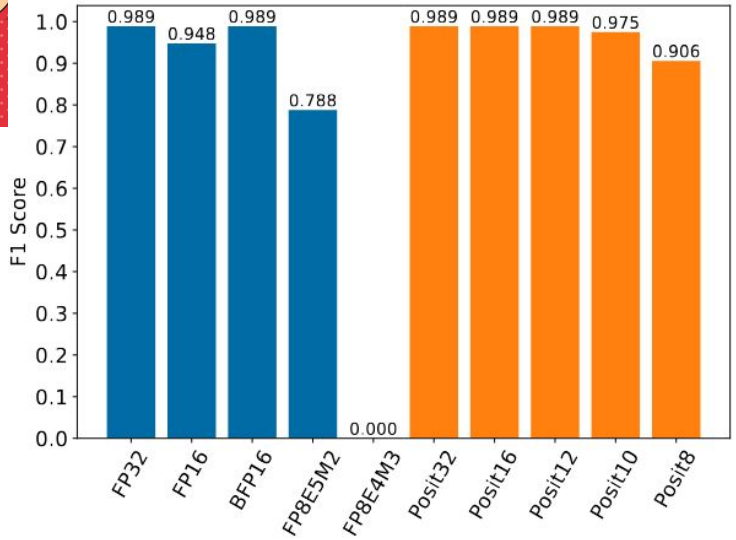
1. D. Mallasén, P. D. Schiavone, A. A. D. Barrio, M. Prieto-Matias, and D. Atienza, '**Increasing the Energy-Efficiency of Wearables Using Low-Precision Posit Arithmetic with PHEE'**, Jan. 30, 2025, arXiv: arXiv:2501.18253. doi: 10.48550/arXiv.2501.18253.

16-bit posit Vs. 32-bit FP

R peak detection



Cough detection



[1] D. Mallasén, P. D. Schiavone, A. A. D. Barrio, M. Prieto-Matias, and D. Atienza, '**Increasing the Energy-Efficiency of Wearables Using Low-Precision Posit Arithmetic with PHEE**', Jan. 30, 2025, arXiv: arXiv:2501.18253. doi: 10.48550/arXiv.2501.18253.

16-bit posit Vs. 32-bit FP

TABLE I
COPROSIT AND FPU_SS AREA RESULTS.

Area (μm^2)	Coprosit	FPU_ss
PRAU / FPU	2353.85	3726.26
Register File	878.79	1896.31
Controller	190.56	211.25
Input Buffer	178.33	231.41
Result FIFO	80.66	—
ALU	79.11	—
Mem Stream FIFO	63.82	63.82
Decoder	31.52	25.87
Predecoder	9.07	11.20
CSR	—	112.39
Compressed Predecoder	—	9.38
Total	4076.23	6565.43

PAU 35% smaller

Coprosit 38% smaller

TABLE III
COPROSIT AND FPU_SS DETAILED POWER VALUES.

Power (μW)	Coprosit	FPU_ss
PRAU / FPU	21.4	46.5
Input Buffer	24.7	31.7
Regfile	19.1	29.9
Controller	16.3	16.6
Result FIFO	10.8	—
Mem Stream FIFO	6.2	6.2
ALU	5.4	—
Decoder	1.1	1
Predecoder	0.3	0.4
CSR	—	14.6
Compressed Predecoder	—	0.2
Total	115	159

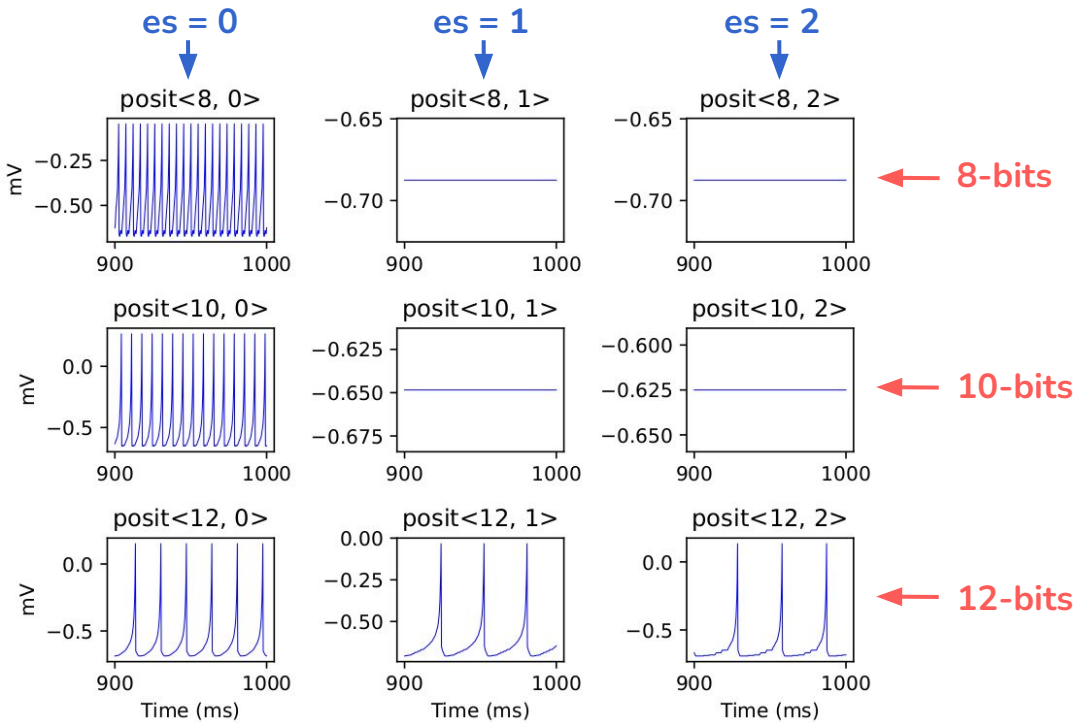
PAU >50% less power

Coprosit 28% less power

[1] D. Mallasén, P. D. Schiavone, A. A. D. Barrio, M. Prieto-Matias, and D. Atienza, '**Increasing the Energy-Efficiency of Wearables Using Low-Precision Posit Arithmetic with PHEE**', Jan. 30, 2025, arXiv: arXiv:2501.18253. doi: 10.48550/arXiv.2501.18253.

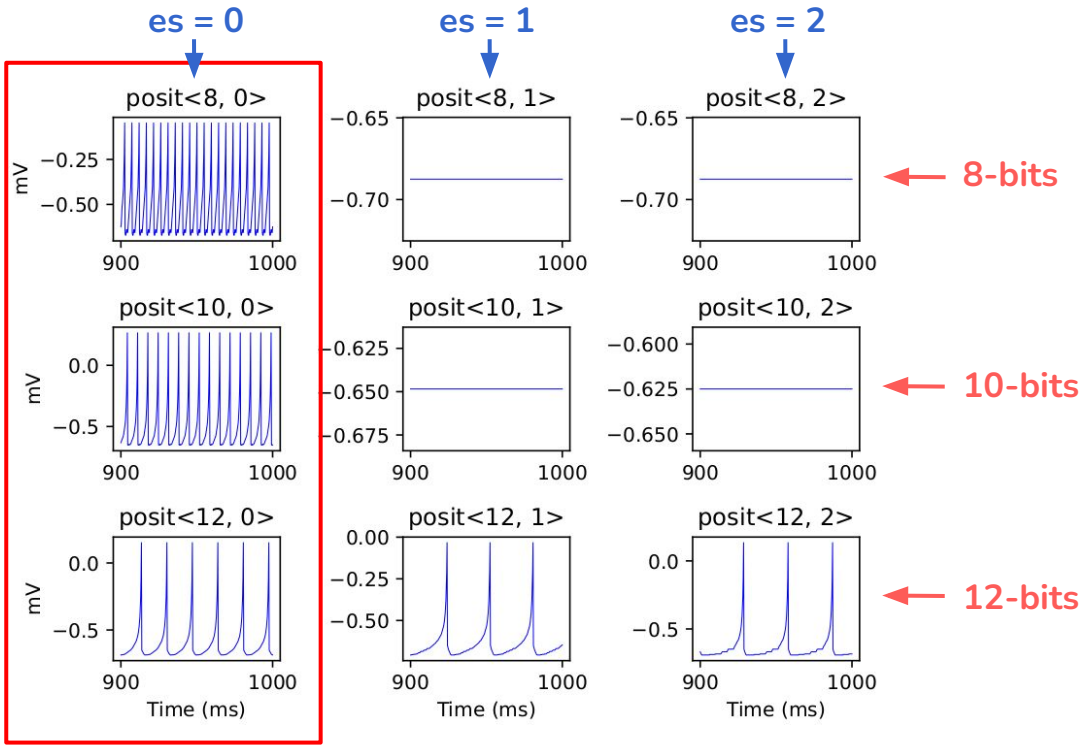
Curious dynamics

posit<n, es>



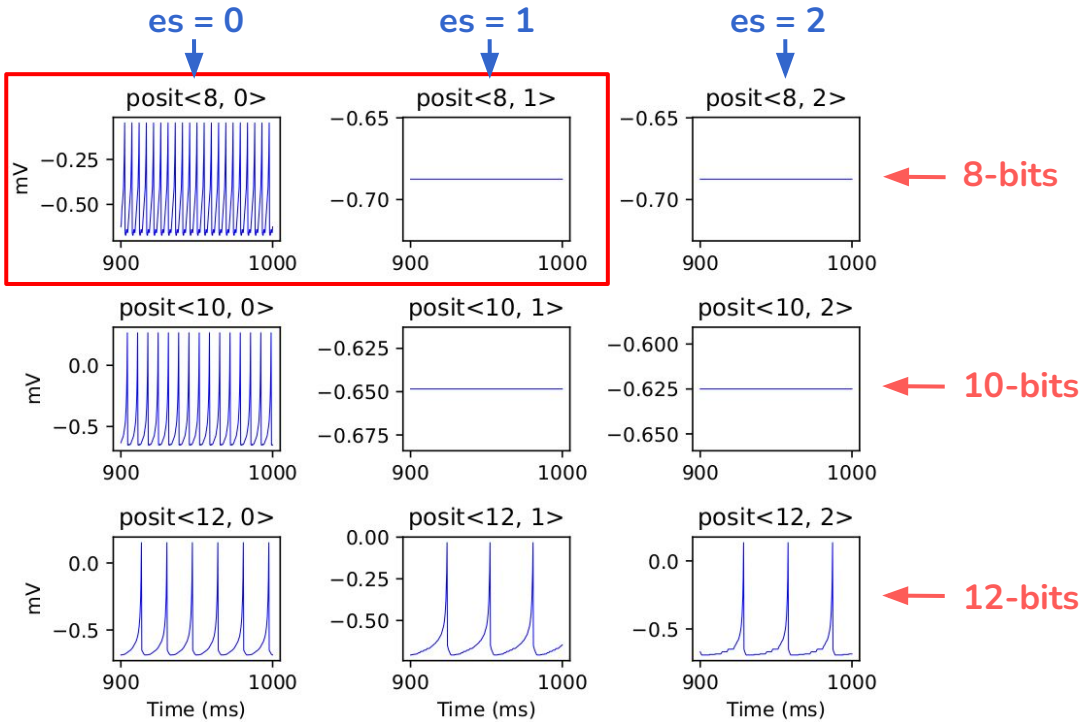
T. Fernandez-Hart, J. C. Knight, and T. Kalganova, '**Posit and floating-point based Izhikevich neuron: A Comparison of arithmetic**', *Neurocomputing*, vol. 597, p. 127903, Sep. 2024, doi: 10.1016/j.neucom.2024.127903.

Curious dynamics



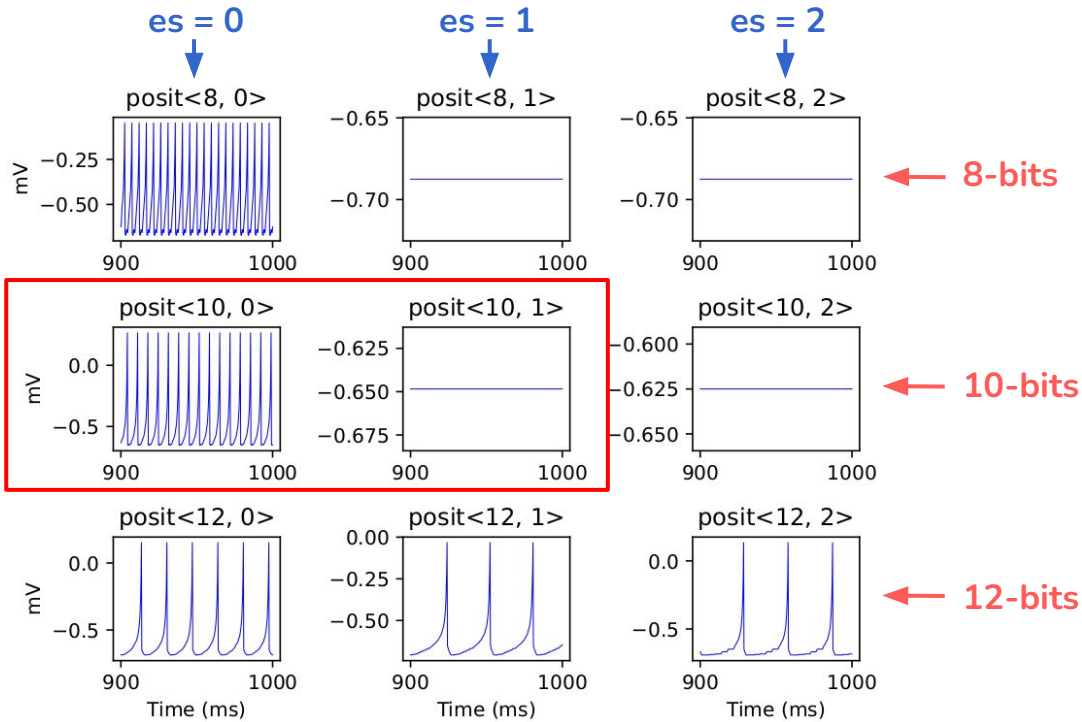
T. Fernandez-Hart, J. C. Knight, and T. Kalganova, '**Posit and floating-point based Izhikevich neuron: A Comparison of arithmetic**', *Neurocomputing*, vol. 597, p. 127903, Sep. 2024, doi: 10.1016/j.neucom.2024.127903.

Curious dynamics



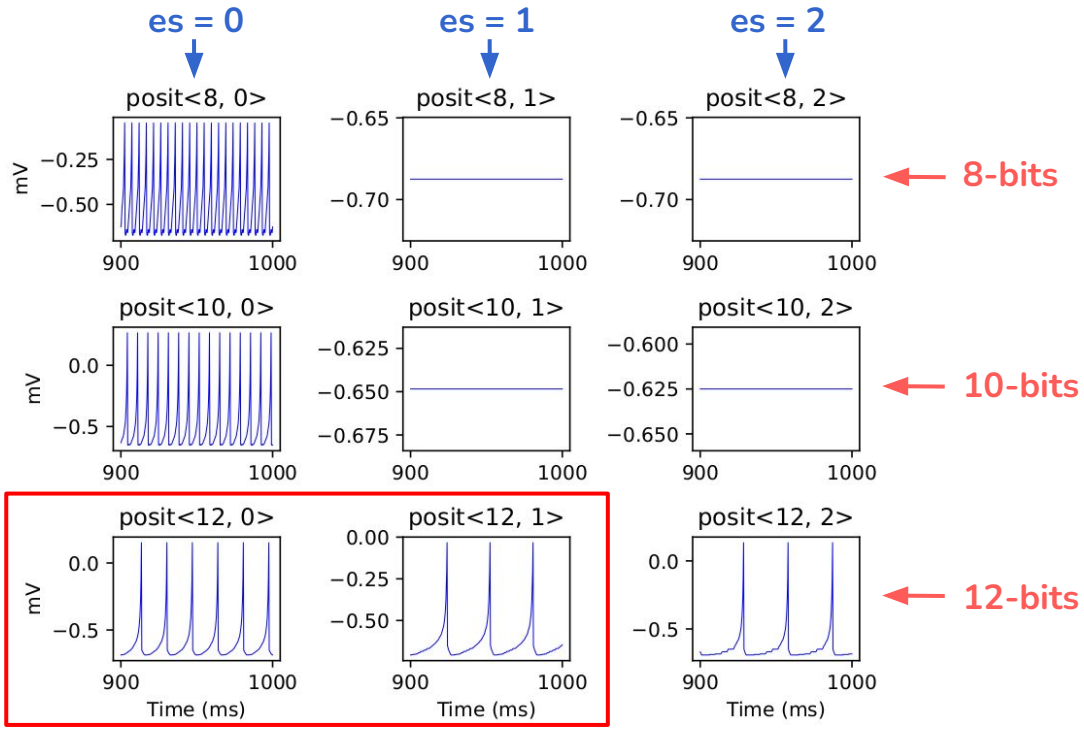
T. Fernandez-Hart, J. C. Knight, and T. Kalganova, '**Posit and floating-point based Izhikevich neuron: A Comparison of arithmetic**', *Neurocomputing*, vol. 597, p. 127903, Sep. 2024, doi: 10.1016/j.neucom.2024.127903.

Curious dynamics



T. Fernandez-Hart, J. C. Knight, and T. Kalganova, '**Posit and floating-point based Izhikevich neuron: A Comparison of arithmetic**', *Neurocomputing*, vol. 597, p. 127903, Sep. 2024, doi: 10.1016/j.neucom.2024.127903.

Curious dynamics



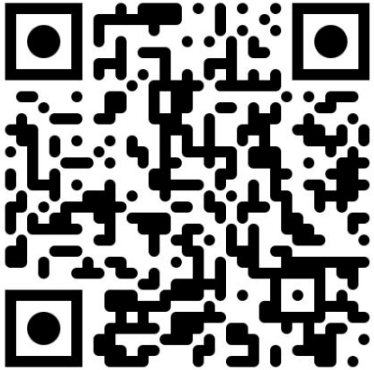
T. Fernandez-Hart, J. C. Knight, and T. Kalganova, '**Posit and floating-point based Izhikevich neuron: A Comparison of arithmetic**', *Neurocomputing*, vol. 597, p. 127903, Sep. 2024, doi: 10.1016/j.neucom.2024.127903.

Conclusions

- **‘Efficiency’ is multifaceted**
  - Not just bit-width, range and precision
  - Arithmetic format’s numerical capabilities
  - Researcher time & effort
- **Performance matched comparison:**
  - **16**-bit posit ⇔ **32**-bit FP ⇔ **64**-bit Fixed-point

- **A more complex number format may paradoxically be *MORE* efficient than simpler one.**

End



T. Fernandez-Hart, J. C. Knight, and T. Kalganova, '**Posit and floating-point based Izhikevich neuron: A Comparison of arithmetic**', *Neurocomputing*, vol. 597, p. 127903, Sep. 2024, doi: 10.1016/j.neucom.2024.127903.

T. Fernandez-Hart, T. Kalganova and J. C. Knight, "**Exploring 8-Bit Arithmetic for Training Spiking Neural Networks**," 2024 *IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, London, UK, 2024, pp. 1-6, doi: 10.1109/COINS61597.2024.10622154.

#OPENTOWORK