A Vectorized Exact Quire for B-Posit Arithmetic on AI Accelerators

Ry Bruscoe (Anomly) Laslo Hunhold (Openchip)

CoNGA 2026, Chicago



Curta II. Photo: Oldsoft, CC BY-SA 4.0, Wikimedia Commons

github.com/anomly-labs/open-bposit

Zenodo 10.5281/zenodo.22651391

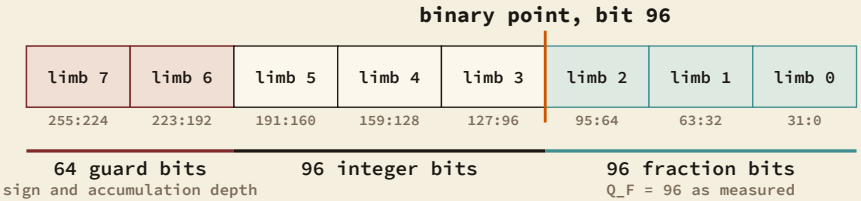
The harm

Same model. Same prompt. Two runs.

- Reward verifier: BF16 flips **52** of 4,096 verdicts. Quire: **0**.
- Logit GEMM: BF16 changes **30,077** of 32,768 logits. Quire, full size on FPGA: **131,072 / 131,072** identical.
- $2^{40} + 62 - 2^{40}$: fp32 says **0**. Quire says **62**.

The answer should be determined by the question, not by the machine that added the numbers.

The quire in one picture



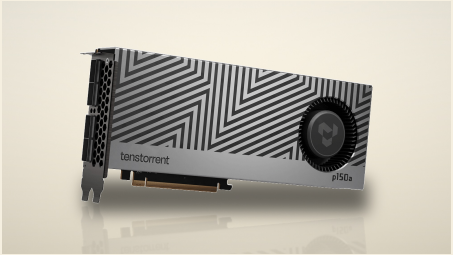
256 = 192-bit product span + 64 guard bits.

Set by the format's range, not chosen.

BF16, exponents -126 to 127: over 500 bits.

What was built

- A 256-bit quire per lane
- Blackhole, all 130 cores
- Any M, N, K
- Native ttnn op

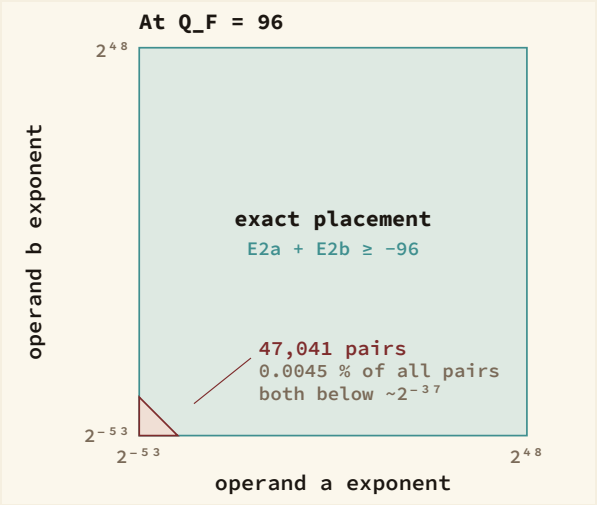


quires in flight	4,160
exact vs oracle	4,160 / 4,160
any shape	4,480 / 4,480
	8,192 / 8,192

How we know it is right

- **Oracle:** bit-for-bit vs arbitrary precision (ExBLAS method)
- **Codec:** all 65,534 codes, 0 mismatches
- **3 SAT proofs:** compressor, reduction, addend
- **Carry identity,** proved for all inputs:
  $\mathsf{c}_{\text{out}} = \big((\mathsf{q} \,\&\, \mathsf{x}) \mid ((\mathsf{q} \mid \mathsf{x}) \;\&\; \neg \mathsf{s}_0)\big) \gg 31$
- **Independent host:** AIFoundry core-et (ET-SoC1 descendant), RV64 build: 132,811 / 132,811 identical; caught a real bug

Exactly which products are exact



Exact everywhere but one corner.

0.0045% of pairs, both tiny.

Fix: Q_F = 106.
Same 256 bits, same adder.

Where it changes the answer

case	float	quire
cancellation	0	**62**
attention outliers	wrong arg-max	**right**
biased BF16 sum, z	+154	**0.0**
depth k = 5120, error	1.13×10^{-3}	**7.87×10^{-5}**

Distilled triggers: they show it can happen, not how often.

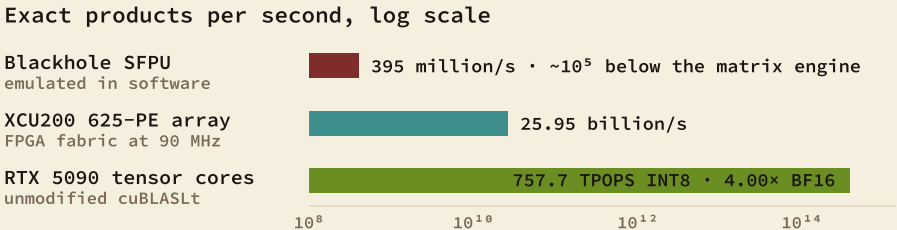
Order independence on hardware



The Alveo U200 the quire array ran on. Image: Xilinx

- **131,072 / 131,072** identical across two orders (U200)
- Llama-3.2-1B, b-posit W8A8: cosine **0.9924** (matmul on the host)
- W8A8 on the Tensix matrix engine: **24 / 24** exact

What it costs



XCU200	LUT	Fmax	power
exact (quire)	**4,179**	**341.6 MHz**	16 mW
rounded	9,673	32.6 MHz	38 mW
fp16 MAC			**7 mW**

Smaller and faster than rounding. Dearer than fp16.

After submission (not in the proceedings)

Measured 20 October 2026, n = 5.

measured since submission	vs BF16
fused GEMM + encode, 4096^3	**4.04×**
Qwen2.5-Coder-1.5B, prefill	1.46–1.62×
Qwen2.5-Coder-7B, prefill	1.94–2.06×
Qwen2.5-Coder-7B, decode	1.56× graphed



One consumer RTX 5090, cuBLASLt.
Photo: TechPowerUp

Same arg-max. Perplexity +0.9 %.

Alveo U200: 1.46 trillion exact/s (paper: 48 billion)

5-bit and b-posit8 modes, 4096^3 GEMM, 291 MHz, 0 mismatches in 348,306, 1 October

Takum and b-posit

- Takum: general purpose. Wide range: **takum16 wins** 2.66×10^{-3} vs 7.61×10^{-2}
- b-posit: a bounded range, sized for AI
- Near 1: a tie at eS = 2; deployed eS = 3 is one bit coarser
- The quire is **independent of the format**

b-posit16 arm uses the eS = 2 profile (Limitation 7).

Limitations

- Llama: matmul on the host
- SFPU: 10^5 slower. A proof, not a deployment
- FPGA only. No standard-cell numbers
- 16-bit encoder: tested, not proved
- 0.0045% of pairs excluded. No 800-bit quire
- Selective precision: a proposal
- One device each

What to take away

Cost is set by **where the accumulator sits,**
not by the format.

vector unit, emulated	10^5 slower
dedicated datapath	**1.46 trillion exact/s**
tensor cores	**≥ BF16 rate**

Built. Proven. Measured.

Artifact

github.com/anomly-labs/open-bposit

Zenodo 10.5281/zenodo.22651391

Every number in the paper, mapped to what produced it.

GPU kernels: anomly-labs/mosyne-bposit

Thank you.