Dmitrii Vasilev

FPGA / RTL & Verification Engineer · Hardware-AI · Custom Number Formats

Ternary ML · GoldenFloat / GF-T · spec→RTL→silicon · functional & formal verification

📍 Ko Samui, Thailand · Remote worldwide (UTC+7) admin@t27.ai Telegram @t27_dev
github.com/gHashTag arXiv 2606.05017 2606.09686

PROFILE

Creator of GF-T — a ternary number format measured best-in-class on benchmarks (≈3–5.5× vs comparable formats such as tekum16) and of the whole **GoldenFloat** family (GF-T8/16/32; φ-derived GF4→GF1024). Full-stack hardware engineer who owns the chain **spec → RTL → verification → silicon**, with a focus on bit-exact conformance and correctness. I build golden-reference verification where every RTL node is checked bit-for-bit against an independent model, catching spec/RTL divergence **before synthesis**. Took a novel numeric format from an arXiv paper to a live Artix-7 and a SKY130 ASIC — on a fully open-source flow, no Vivado. 10+ years building and teaching software; latest cycle entirely RTL / FPGA / verification.

Grade: Senior / Lead · **Open to:** remote hardware-AI · ML-systems · FPGA/RTL/verification — full-time or contract.

KEY ACHIEVEMENTS (ON HARDWARE, 2026)

- **100% held-out accuracy** — verified neural primitives that **train themselves on FPGA**: binary & 3-class classification 100% held-out, a 2-layer ReLU net solves XOR (4/4), whole spec→silicon path **bit-exact**.
- **Per-node bit-exact regression** — pipeline where every RTL node is checked with KAT vectors against a Python golden model (iverilog); regression catches spec/RTL divergence pre-synthesis.
- **Published conformance catalog for 83 numeric formats** — bit-exact test vectors for FP8/BF16/MXFP4/microscaling (arXiv:2606.09686) — a vendor-neutral reference for arithmetic verification.
- **323 MHz · 41.2 GOPS · 0 latches** — designed & verified a GF16 4×4 matmul on Artix-7 (0 DSP48, clean timing sign-off).
- **Silicon confirmed** — TinyTapeout SKY130 ASIC: GDS ✅ · gate-level test ✅ · precheck ✅.
- **tri-net — full ternary network stack (OSI / TCP-IP analog)** — 133 `.t27` specs: ternary GF16 PHY, BPSK modem over AD9361, ETX mesh routing, AEAD crypto (ChaCha20-Poly1305 / X25519); every layer formally specified & FPGA-synthesizable, proven device-to-device over the air.

PUBLICATIONS & COMPETITIONS

GoldenFloat: A Phi-Derived Static-Split Floating-Point Family (GF4→GF1024) with a Lucas-Exact Integer Identity — arXiv:2606.05017

An 83-Format Numeric Catalog with Bit-Exact Conformance Vectors: FP8, BF16, MXFP4, Microscaling — arXiv:2606.09686

Competitions: DARPA CLARA (PA-25-07-02) submission — Trinity Cognitive Stack, 10 CLARA reasoning gaps realized in open-RTL silicon (TinyTapeout SKY130) · OpenAI Parameter Golf — entry with novel GF16 quantization · Google DeepMind AGI Hackathon 2026 · The Open League (TON).

EXPERIENCE

FPGA / RTL / Verification Engineer — Trinity S³AI 2026 — present

- Designed & verified RTL for ternary neural primitives and a GF16 matmul; introduced per-node bit-exact checking vs an independent Python golden model (iverilog + KAT). Built an end-to-end open-source RTL→bitstream flow (Yosys · nextpnr-xilinx · prjxray · openFPGALoader · iverilog) on macOS arm64, no vendor tools; taped out to SKY130 (TinyTapeout).

AI Consultant / Agent-systems Developer — Академия Вайбкодинга 2025 — 2026

- Multi-agent AI solutions (Gleam, Eliza OS, Claude Code, Cursor, SWE-Agent); mentored 50+ developers; delivered 5+ enterprise-grade projects.

Founder, AI Agents Developer — NeuroBlogger 2024 — present

- Founded and built an AI content-generation platform (Claude / GPT-4 / Gemini, multi-agent, Telegram bots); scaled the product from zero.

Full Stack Developer — NeuroCalls Jan 2024 — Aug 2024

- Full-stack build of a meeting-analysis product that transcribes and summarizes conversations and extracts action items (Telegram, LLMs). TON Open League Hackathon participant.

Senior React Native Developer — HAQQ (UAE) Sep 2022 — Jan 2023

- Blockchain wallet product (Web3, React Native, Redux) in an international team.

React Native Developer — Leela Chakra AI 2022 — 2023

- React Native apps for an AI product, published to the App Store and Google Play.

Founder — JS CAMP 2016 — 2023

- Founded a programming school (React Native / GraphQL / AWS Amplify); created the first React Native + AWS Amplify course in RuNet; mentored 1000+ developers.

Senior React Native Developer — Muffins / Rightway Healthcare (US) 2021 — 2022

- Mobile apps for a US health-tech leader (React Native, GraphQL).

React Native Developer — Business Generation 2019

- React Native development for mobile products (Moscow).

React Native Developer — Easypay Indonesia (National Payment Systems) 2018 — 2019

- React Native app for Indonesia's national payment system, serving millions of users (Bali, Indonesia).

SKILLS

Verification: functional verification · golden-reference / scoreboard modeling · directed & KAT vectors · regression · RTL simulation (iverilog) · formal verification (Coq/Rocq) · bit-exact conformance · latch-free / timing sign-off · CDC

RTL / FPGA: Verilog · Yosys · nextpnr-xilinx · prjxray · openXC7 · openFPGALoader · JTAG/XVC · Xilinx Artix-7 · SKY130 PDK · TinyTapeout

Languages / tools: Rust · Zig · Python · Tcl · Git · LaTeX · compilers (spec→HDL) · ternary logic · custom float formats

Ramping (industry stack): SystemVerilog · UVM · SVA assertions · Questa/VCS — fast transfer from strong reference-model-checking & formal background

AI / ML: quantization · on-device training · BitNet-class · MCP · RAG · multi-agent systems

EDUCATION

- **SADK** — Technician-technologist (higher), 1988 · **State University of Management**, Moscow — Management, 2006
- **AWS Community Builders** — member · English: **B2** (write English arXiv papers)

SELECTED OPEN-SOURCE — GITHUB.COM/GHASHTAG (186 REPOS)

trinity (ternary compute stack) · **t27** (spec-first `.t27`→Verilog, bit-exact) · **trinity-fpga** (GF16 matmul, 323 MHz) · **zig-golden-float** (GF-T) · **trinity-s3ai** (Coq/Rocq machine-checked proofs) · **tri-net** (ternary mesh, AEAD crypto).

$\phi^2 + \phi^{-2} = 3$ · Building hardware that thinks — and proving it computes correctly