Boreal Neuro-Core v2.4: Final Build Documentation

The Boreal Neuro-Core v2.4 is a specialized FPGA architecture designed to close the loop between the human nervous system and high-speed robotics. It bypasses the latency of software-based neural processing by executing **Active Inference** directly in the silicon fabric.

Architecture Highlights

- **Inference Latency:** < 500ns (Input to Internal State Update).
- **Core Principle:** Variational Free Energy Minimization (FEP).
- **Signal Integrity:** Integrated 24-bit DC-blocking high-pass filters and saturation arithmetic to prevent overflows.
- **Hardware Placity:** Hebbian Learning Module for real-time synaptic weight updates in BRAM.
- **Scaling:** Daisy-chain support for up to 32 EEG channels (4x ADS1299).

File Manifest

1. lut_gen.py: Pre-calculates Sigmoid (\sigma) and Derivative (\sigma') tables.
2. boreal_apex_core.v: The mathematical heart (Filtering + Inference).
3. boreal_memory.v: Dual-Port BRAM for weight/LUT storage.
4. boreal_pll_tracker.v: Frequency-adaptive zero-crossing phase lock.
5. boreal_peripherals.v: Learning engine, PWM, and expansion logic.
6. boreal_system_top.v: Integrated top-level RTL.
7. DigitalTwin.jsx: React-based simulation and verification dashboard.
8. boreal_constraints.xdc: Physical FPGA pin mapping.

Operational Instructions

1. **Initialize:** Generate boreal_lut.mem using the Python script.
2. **Synthesize:** Load all .v files and the .xdc into your FPGA toolchain (e.g., Vivado).
3. **Deploy:** Connect the ADS1299 SPI pins and the physical bite_switch_n for safety.
4. **Verify:** Use the Digital Twin dashboard to tune the alpha (filter) and learning_rate parameters.

Note: Ensure the emergency bite-switch is physically accessible during all flight HUD or prosthetic testing phases.