Rationale Report

Microwatt Secure Boot

Summary: This report outlines the key design and build approaches explored during the Microwatt Secure Boot implementation, along with the reasoning behind each decision. It captures the iterative process of environment setup, simulation, integration, and debugging, providing a clear view of how and why the final working configuration was reached.

1. Environment and Build Setup Approaches

Approaches Tried:

- **Direct Local Build using GHDL and Yosys** Attempted a native build on Ubuntu 24.04 using system-installed tools. **Why:** To quickly prototype without Docker and verify dependencies. **Result:** Led to library path mismatches and LLVM plugin issues (ghdl-yosys plugin missing).
- **Docker-based Controlled Build Environment** Created a Dockerfile with all required tools (GHDL, Verilator, Yosys, LLVM-18). **Why:** Ensure reproducible builds and dependency isolation. **Result:** Successfully built the image; used as the main simulation environment thereafter.

2. Microwatt Integration Attempts

Approaches Tried:

- **Initial Workdir Compilation (Flat)** Compiled VHDL sources directly from the root directory. **Why:** Simplicity for debugging early integration. **Result:** "unit not found in library 'work'" errors appeared due to missing compilation order.
- **Hierarchical Make-based Compilation** Invoked Microwatt's internal Makefile via our own build script. **Why:** Preserve the source's internal dependency structure. **Result:** Clean compilation of Microwatt submodules and predictable output files.

3. Top-level Design Integration

Approaches Tried:

- **Direct Wishbone Wiring (Simplified Bus)** Used a minimal subset of Microwatt's data wishbone ports to connect to a stub SHA-256 unit. **Why:** Reduce integration complexity while still demonstrating secure-boot behavior. **Result:** Simulation ran successfully; UART signals were observable in waveform.
- **SOC-style Connection using Microwatt Core Entity** Integrated full `core.vhdl` ports (wishbone_insn, wishbone_data, etc.) to mirror Microwatt's SoC design. **Why:** To ensure realistic system-level simulation. **Result:** Working build but longer runtime; future integration with firmware pending.

4. Simulation Strategy Evolution

Approaches Tried:

- **Manual Run Script (`run.sh`) Why:** Early stage manual control for quick testing. **Result:** Functional, but required repeated chmod and GHDL cleanup steps.
- **Unified Makefile-driven Flow Why:** Automate build, elaborate, and simulate in a single command. **Result:** Simplified the workflow; made `make sim` the one-step test command.

5. Final Stable Configuration

Outcome:

- Stable build and simulation under Docker (Ubuntu 24.04 + LLVM 18).
- Microwatt core and SHA-256 stub integrated and compiled cleanly.
- Simulation outputs waveforms and UART activity logs successfully.
- Remaining tasks: firmware payload integration and SHA verification logic.