EchoPulse – Document 17: Hardware Mapping Concepts

This document outlines potential hardware integration targets and design considerations for implementing the EchoPulse symbolic Key Encapsulation Mechanism (KEM) in real-world embedded environments.

1. Target Devices and Architectures

- IoT Microcontrollers: ARM Cortex-M0+, M4F, RISC-V embedded cores
- Smartcards / Secure Elements: JavaCard, ISO7816-compliant chips
- FPGAs: Xilinx Artix, Intel Cyclone – with SHA3 module and symbol-path LUT
- Custom ASICs: Minimalistic crypto cores for EchoPulse-specific path/state logic

2. Requirements and Constraints

- RAM requirement: ~8–10 KB for full EchoPulse operation
- ROM/Flash: ~32–64 KB for code and static tables
- Timing: < 500 µs for encapsulation (target M4F class)
- Entropy source: required for r-generation and mutation trigger

3. Implementation Notes

- Use lookup tables for transition function δ(state, symbol) in embedded
- Mutation µ(t) can be triggered via hardware counters or watchdog events
- SHA3 acceleration optional but preferred (software fallback viable)