RAMTERA

PRODUCT DOCUMENTATION · PHYSICAL IMPLEMENTATION (PRE)

RAMTERA-PRE-ARCH

Platform Architecture Overview

One engine behind GUI, batch, and Tcl — with stated concurrency properties

August 2026 · Ramtera PRE engineering

PRE Platform Architecture Overview

PRE is a single C++ engine with three faces: a Tcl interpreter carrying the full command surface, a batch mode for farms, and a Qt GUI for interactive debug — all three run the same engine code, so a script behaves identically in every mode. This document describes the design model, the engine stages, and the concurrency properties.

1	2	worker-pool	multi-host
engine behind GUI, batch, and Tcl	global placers, one command surface	parallelism in analysis stages	distributed multi-corner timing

Design model

Technology (LEF), design (DEF or Verilog), timing libraries (liberty), constraints (SDC), power intent (UPF), and parasitics (SPEF) load into one in-memory model shared by every stage. Published routing geometry is tagged by owning net, which is what lets antenna checking, clock non-default rules, and stream-out consume routing without re-deriving connectivity.

Engine stages and their contracts

Stage	Architectural statement
Placement	two engines — quadratic with density spreading, and a nonlinear analytical solver of the electrostatic family — selected by mode behind one command; legalization and refinement are common back-ends
Timing	graph-based STA over NLDM tables with slew propagation; per-net wire model is RC-tree where SPEF annotates and geometric otherwise; exceptions resolve by the documented priority model
MMMC	views are first-class (library set × RC corner × constraint mode); analysis runs per active view and merges into one report
Routing	global routing allocates GCell capacity; detailed routing checks spacing in-loop against committed geometry, so published wires are clean by construction
Power	UPF domains project into placement as fences; PG geometry feeds a nodal conductance solve for static IR with current-density checks

Concurrency and distribution

Analysis and data-preparation stages run on a process-wide worker pool with results merged in deterministic order — the same design and thread setting produce the same bytes, and analysis results are independent of thread count. Multi-corner timing additionally distributes across hosts (secure-shell or batch-queue launchers); the acceptance criterion is exact equality with local analysis, and the published validation shows it (WP-21). The nonlinear placement engine is intentionally excluded from the determinism guarantee, as analytical placers of its family generally are; its output is judged by the checker and the timer.

SCOPE OF THIS DOCUMENT — Properties, not mechanisms: the scheduling, result-merge, and distribution methods behind these guarantees are the subject of patent filings.

About Ramtera

Ramtera Inc. (Delaware, USA), with R&D at Ramtera Design Automation India Pvt Ltd (Bangalore), builds electronic design automation software for physical verification and physical implementation. LVR is Ramtera's physical verification engine (DRC, LVS, ERC, XOR, parasitic extraction); PRE is Ramtera's physical implementation environment (floorplanning, placement, clock tree synthesis, routing, timing analysis and optimization), built on industry-standard command conventions so existing flow scripts and operator skills transfer directly. The company holds granted patents and registered copyrights covering its engines.

Contact

Web	www.ramtera.com
Inquiries	sameer@ramtera.com
Engineering	Bangalore, Karnataka, India

This document reports measurements from Ramtera's engineering environment (Ubuntu 24.04, GCC 13.3, Tcl 8.6) unless stated otherwise. Product behavior and figures are subject to change; no compatibility, certification, or endorsement by any third party is stated or implied.