**Project Requirement Specification (PRS)**

**Project Title:** 1 TH/s Cryptocurrency Mining ASIC Design  
**Process Technology:** SKY130 Open Source PDK (130nm CMOS)  
**Die Size:** 28 mm × 28 mm  
**Target Throughput:** 1 Terahash per second (1 TH/s)  
**Power Efficiency:** High (Target optimized power per hash)  
**Design Paradigm:** Hierarchical pipelined job distribution (Grand → Child → Grandchild units)

**Functional Requirements**

* Implement SHA-256 hash computation pipeline for cryptocurrency mining.
* Support hierarchical job dispatch:
  + **Grand:** Top-level job manager distributing hashing jobs downstream.
  + **Child:** Intermediate processing units receiving jobs and forwarding subtasks.
  + **Grandchild:** Hash computation cores executing SHA-256 calculations.
* Pipeline job flow top-down; results flow bottom-up with minimal latency.
* Provide control and status registers accessible via on-chip interface.

**Performance Requirements**

* Achieve aggregate throughput of at least 1 TH/s at target frequency.
* Maintain timing closure at chosen clock frequency with pipelined stages.
* Ensure power efficiency through clock gating and optimized datapaths.

**Physical Constraints**

* Total die area capped at 28 mm × 28 mm to fit foundry constraints.
* Use SKY130 PDK design rules and standard cell libraries.
* Implement hierarchical floorplanning for modular block placement.

**Verification & Testing**

* Full RTL simulation of functional blocks and pipeline integration.
* Post-synthesis and post-layout timing verification.
* DRC, LVS, and parasitic extraction per SKY130 foundry rules.

**Design Implementation Plan**

| **Phase** | **Description** | **Tools/Methods** | **Deliverables** |
| --- | --- | --- | --- |
| 1. Specification & Architecture | Define hashing algorithm pipeline, job distribution hierarchy, and power goals | Documentation, architecture diagrams | Detailed design specification document |
| 2. RTL Design | Develop modular Verilog RTL for grand, child, grandchild pipeline units | Verilog, simulation testbenches | Verified RTL code with testbenches |
| 3. Functional Verification | Simulate RTL for correctness and throughput | Icarus Verilog, ModelSim, Verilator | Passing test results and coverage reports |
| 4. Synthesis | Convert RTL to gate-level netlist with timing and power constraints | Yosys | Synthesized netlist, reports |
| 5. Floorplanning & Placement | Create hierarchical floorplan; place modules in 28mm×28mm die area | OpenROAD, custom floorplan scripts | Floorplan, placement DEF files |
| 6. Clock Tree Synthesis | Generate balanced clock network with gating for power efficiency | OpenROAD CTS | Clock tree implementation |
| 7. Routing | Route signal and power nets; optimize for signal integrity | OpenROAD router | Routed layout database |
| 8. DRC & LVS | Verify design rules and layout matches netlist | Magic, OpenROAD DRC/LVS | DRC/LVS clean reports |
| 9. Parasitic Extraction & Timing Signoff | Extract parasitic data, run static timing analysis | OpenSTA, StarRC (if available) | Timing closure reports |
| 10. Power Analysis & Optimization | Analyze power, refine gating and layout to reduce consumption | Power analysis tools, iterative optimization | Final power report |
| 11. GDSII Generation | Generate final GDSII layout file for tapeout | OpenROAD, Magic | GDSII file ready for fabrication |
| 12. Silicon Validation | Develop testbench and silicon test plan | Lab equipment, on-chip debug | Validation reports, silicon debug data |