Caravel Functional-vs-GLS Verification Task

Objective:
Verify that Caravel's RTL simulation and Gate-Level Simulation (GLS) produce identical functional results for the hkspi test and subsequently for all other DV tests inside efabless/caravel/verilog/dv.

Tools Allowed:
Icarus Verilog or Verilator, OpenLane + OpenROAD, Magic/KLayout, Sky130 PDK, GitHub Codespaces or local system.

Task List

1. Environment Setup

   a. Clone the official Caravel repository - https://github.com/efabless/caravel
   b. Install Sky130 PDK.
   c. Confirm that the following tools run correctly:
      - iverilog
      - vvp
      - verilator
2. Understand Caravel's hkspi Test
   a. Open and study the following files:
      - verilog/dv/hkspi/hkspi_tb.v
      - verilog/rtl/housekeeping_spi.v
      - hksip.hex

      b. Understand the I/O behavior, register expectations, and test logic.
      c. Write a short note (max 1 page) on how hkspi interacts with the management SoC and user project.
3. Run RTL Simulation for hkspi
   a. Compile hkspi testbench using Icarus Verilog.
   b. Run vvp and capture the console output.
   c. Verify that the output ends with:
   Monitor: Test HK SPI (RTL) Passed
   d. Save the full RTL log as rtl_hkspi.log.
4. Generate Gate-Level Netlist
   a. Run synthesis for Caravel top-level using yosys
   b. Extract the corresponding gate-level netlist for the hkspi test (typically inside verilog/gl or openlane/caravel/results).
   c. Verify that the synthesized modules match the expected hierarchy.
   d. Save the synthesized netlist and attach it as part of deliverables.
5. Run Gate-Level Simulation (GLS) for hkspi
   a. Compile the gate-level netlist with the hkspi testbench and Sky130 standard cell models.
   b. Run vvp with timing annotation if required.
   c. Verify that GLS also ends with:

Monitor: Test HK SPI (GL) Passed
d. Save the log as gls_hkspi.log.

6. Compare RTL vs GLS Output
   a. Perform a line-by-line comparison for all register reads printed by hkspi.
   b. Confirm that every value in GLS matches RTL exactly.
   c. Write a short summary explaining the match (or mismatch if any).
7. Extend the Same Flow to All Other DV Tests - https://github.com/efabless/caravel/tree/main/verilog/dv
   For each DV test in verilog/dv:
   a. Study its corresponding testbench.
   b. Run RTL simulation and record the logs.
   c. Run GLS after synthesis and record the logs.
   d. Verify that functional outputs match and that final monitor messages end with "Passed" for both RTL and GLS.
   e. Produce a table with:
      - Test name
      - RTL result
      - GLS result
      - Functional match (Yes/No)
8. Documentation and Submission
   Submit the following materials in a clean directory structure:
   a. RTL logs for every test
   b. GLS logs for every test
   c. Synthesized gate-level netlists for caravel
   d. A 2–3 page summary describing:
      - Environment setup
      - hkspi verification results
      - Challenges faced in GLS
      - Final RTL vs GLS comparison table

      e. Screenshots of at least one successful waveform (RTL or GLS) for hkspi.

Final Deliverable

A complete verification package proving that Caravel's RTL and GLS produce identical behavior for hkspi and all other DV tests using only open-source tools and Sky130 PDK.

Deadline: 9th December, 2025