

## Week 7 Task – BabySoC Physical Design & Post-Route SPEF Generation

### Objective

To perform **complete Physical Design** for the BabySoC using OpenROAD — covering **floorplanning, placement, routing, and post-route SPEF generation** — and to understand how all stages come together in a real SoC design flow.

---

### Why This Task Is Important

Up to now, you’ve explored each step of VLSI design in isolation — from RTL and SPICE to synthesis and early floorplanning.

This week’s task integrates everything into a **complete physical design flow on a real chip (BabySoC)**.

By executing this task, you will:

- Experience how a full **RTL-to-layout** process works.
- Understand how **floorplan constraints, placement density, and routing topology** impact timing.
- Learn the significance of **post-route parasitic extraction (SPEF)** and how it feeds into **STA for accurate delay analysis**.

This task connects the **theory of STA, the layout process, and real-world SoC implementation** — the essential foundation for professional ASIC engineers.

---

### Task Reference

Refer to **Task 13** from the below repository for detailed commands and flow setup:

 [OpenROAD Physical Design Reference – Task 13](#)

---

### Task Components

#### 1. BabySoC Floorplanning

- Import the synthesized BabySoC netlist into OpenROAD.
- Define floorplan dimensions, core utilization, and power ring/straps.
- Ensure all macros and standard cells are placed within the defined boundary.

## 2. Placement

- Perform **global and detailed placement**.
- Verify placement density and legality.
- Capture snapshots of placement layout.

## 3. Routing

- Run **global routing** followed by **detailed routing**.
- Ensure there are **no DRC violations** after routing.
- Capture final routed layout images.

## 4. Post-Route SPEF Generation (Critical Step)

- Extract **Standard Parasitic Exchange Format (SPEF)** using OpenROAD.
  - Verify that net parasitics (resistance, capacitance) are generated successfully.
  - Understand how the generated SPEF is used for **post-route STA** in real design flows.
- 

## Deliverables

### 1. Screenshots / Images

- Floorplan view
- Placement view
- Routed design
- Terminal outputs showing SPEF generation

### 2. Documentation (GitHub Repo)

- Step-by-step summary of commands used
- Observations and challenges at each stage
- Explanation of what SPEF represents and why it's vital for accurate STA

### 3. Verification Note

- Confirm that BabySoC layout completed successfully

- Mention total cell count, utilization, and number of routed nets (from OpenROAD reports)
- 

 **By the end of Week 7**, you will:

- Execute the **complete physical design flow for BabySoC** using OpenROAD.
- Generate and interpret **post-route SPEF files**.
- Develop an end-to-end understanding of how **digital design data becomes a physical chip** ready for timing closure and sign-off.