
Lab Module 10: ASIC Design Flow - Floorplanning, Placement, and Routing (Conceptual/Tool Demonstration)

Duration: 3 Hours

1. Objective(s):

Upon successful completion of this laboratory module, which will primarily involve a guided demonstration and conceptual understanding, students will be able to:

- **Grasp ASIC Physical Implementation Flow:** Develop a clear conceptual understanding of the crucial physical implementation stages within the Application-Specific Integrated Circuit (ASIC) design flow, following logical (RTL/gate-level) design.
- **Understand Floorplanning Principles:** Comprehend the objectives of floorplanning, including defining chip boundaries, managing I/O pin placement, and strategic power distribution planning.
- **Explain Automatic Placement:** Understand the process and goals of automatic standard cell placement within the defined floorplan.
- **Describe Automatic Routing:** Explain how automated routing tools connect placed standard cells using various metal layers according to the design's netlist.
- **Visualize Physical Design Outputs:** Recognize and interpret the visual outputs of floorplanning, placement, and routing stages within advanced EDA tools.
- **Appreciate Post-Layout Extraction:** Understand the critical importance of post-layout parasitic extraction as the final step before sign-off for accurate timing and power analysis.

2. Theory and Background:

In the modern ASIC design flow, after a digital circuit's Register Transfer Level (RTL) description is verified and then synthesized into a gate-level netlist (a description of interconnected standard cells), the next major phase is **physical implementation**, also known as backend design. This phase transforms the logical gate-level netlist into a manufacturable physical layout of the chip. This highly complex process is largely automated by sophisticated EDA tools.

2.1 Transition from Logical to Physical Design:

Prior labs focused on designing and verifying individual gates and custom layouts. However, for large, complex chips, manual layout is impractical. The ASIC design flow leverages pre-designed and characterized standard cells (like inverters, NANDs, NORs, flip-flops, etc.). These cells have fixed dimensions, characterized timing, and pre-verified layouts. Physical implementation tools arrange and connect these standard cells automatically.

2.2 Floorplanning: The Chip's Blueprint:

Floorplanning is the initial and arguably most critical step in physical implementation. It lays out the overall structure of the chip before detailed components are placed. It is like designing the blueprint of a building before placing furniture.

- **Objectives:**
    - **Define Chip/Core Boundaries:** Establish the total physical area the design will occupy on the silicon.
    - **I/O Pin Placement:** Decide where the input/output signals will enter and leave the chip, considering package requirements and signal integrity.
    - **Block Partitioning:** For very large designs, divide the chip into major functional blocks (e.g., CPU core, memory, peripherals) and determine their approximate shapes and locations.
    - **Power Planning:** Design the power delivery network, including VDD and GND rings and meshes using wide metal layers, to ensure stable power supply to all parts of the chip and minimize IR drop (voltage loss).
    - **Macro Placement:** Strategically place large, pre-designed blocks (e.g., embedded memories like SRAM, or custom analog IP blocks) that cannot be automatically placed by the tool. Their fixed size and often specific port locations heavily influence the rest of the floorplan.
- **Challenges:** Balancing area utilization, power distribution efficiency, signal integrity, and routability. A poor floorplan can lead to routing congestion, longer critical paths, and power integrity issues, delaying the project significantly.

2.3 Placement: Positioning the Standard Cells:

After floorplanning defines the general areas, the placement step precisely positions all the standard cells (e.g., inverters, NAND gates, flip-flops) from the synthesized netlist within the core area.

- **Automatic Process:** Placement tools use complex algorithms to determine the optimal location for each standard cell.
- **Objectives:**
    - **Minimize Wirelength:** Placing connected cells close together to reduce interconnect length, which in turn reduces parasitic capacitance and resistance, leading to faster circuits and lower power consumption.
    - **Minimize Congestion:** Avoiding areas where too many wires are needed, which could make routing impossible or inefficient.
    - **Meet Timing Constraints:** Placing cells to satisfy timing requirements for critical paths, ensuring signals arrive within specified deadlines.
    - **Power/Ground Connection:** Ensuring that each placed cell can easily connect to the power and ground rails established during floorplanning.
- **Output:** A layout where all standard cells are placed, but not yet connected by wires (except for internal connections within the cells).

2.4 Routing: Connecting the Placed Cells:

Routing is the final and often most computationally intensive step in physical implementation. It involves drawing the actual metal interconnects (wires) to connect the terminals of the placed standard cells according to the netlist.

- **Multi-Layer Process:** Modern processes have many metal layers (e.g., 6 to 12 or more). Routing tools utilize these layers, typically running wires horizontally on one layer (e.g., Metal1, Metal3, Metal5) and vertically on an adjacent layer (e.g., Metal2, Metal4, Metal6), using vias to connect between layers.
- **Automatic Process:** Routers are highly sophisticated algorithms that find paths for thousands or millions of connections without violating design rules.
- **Objectives:**
    - **Complete All Connections:** Route every single net defined in the netlist.
    - **Adhere to Design Rules:** Ensure all drawn wires and vias comply with minimum width, spacing, and other rules.
    - **Minimize Wirelength:** As with placement, shorter wires are better for performance and power.
    - **Minimize Crosstalk:** Keeping sensitive signals separated to prevent unwanted interference.
    - **Meet Timing Constraints:** Route critical paths optimally to meet timing targets.
- **Output:** A complete, DRC-clean layout of the entire chip with all cells placed and interconnected.

2.5 Viewing the Routed Design:

After routing, the entire design can be viewed in a layout viewer. This provides a detailed, full-chip visual representation, showing all the placed standard cells, the VDD/GND power networks, and the intricate network of metal interconnects spanning multiple layers. This is the closest representation to what will be physically manufactured.

2.6 Post-Layout Extraction and its Importance for Accurate Timing:

Even after routing, the physical design process isn't complete for final verification.

- **Parasitic Extraction:** As introduced in Lab 7, this step analyzes the fully routed layout to identify and calculate all the parasitic capacitances (from wires, contacts, transistors) and resistances (from wires, contacts) that are inherent to the physical geometry. These are unintended but unavoidable electrical components created by the physical layout.
- **Impact on Timing:** These extracted parasitics significantly impact the actual circuit performance.
    - **Capacitance:** Increases the time required to charge/discharge nodes, leading to longer delays.
    - **Resistance:** Causes voltage drops along interconnects and contributes to delays.
- **Accurate Timing Analysis (Timing Closure):** The extracted parasitic information is then used in a final, highly accurate **post-layout timing analysis** (often Static Timing Analysis, STA). This analysis determines if the design still meets all its timing

requirements after considering the real-world parasitic effects. If timing violations occur, the design must go through iterative refinement (e.g., optimizing critical nets, re-placement, re-routing). This iterative process to meet all timing constraints is known as "timing closure." This final parasitic-aware timing analysis is crucial before the chip layout is sent for fabrication ("tape-out").

3. Pre-Lab Questions and Preparation:

Students are expected to complete the following thoroughly before the lab demonstration/conceptual session.

1. **ASIC Design Flow Context:** Briefly describe where floorplanning, placement, and routing fit into the overall ASIC design flow, specifically what input they take (e.g., from synthesis) and what output they produce.
2. **Objective of Floorplanning:** What are the primary goals of the floorplanning stage? List at least three key decisions made during floorplanning.
3. **Role of Standard Cells:** What is a "standard cell" in the context of ASIC design? Why are standard cells used instead of full-custom layout for entire chips?
4. **Placement vs. Routing:** Differentiate between the "placement" and "routing" steps in the physical implementation flow. What is the main objective of each?
5. **Multi-Layer Routing:** Explain why modern ASIC designs utilize multiple metal layers for routing. How are connections typically made between different metal layers?
6. **Post-Layout Significance:** Why is "post-layout parasitic extraction" and subsequent "post-layout timing analysis" so critical, even after extensive simulation in earlier stages?

4. Procedure/Conceptual Hands-On Experience (Guided Tool Demonstration):

This lab will primarily involve a guided demonstration by the instructor using a commercial ASIC physical implementation tool (e.g., Synopsys Innovus, Cadence Innovus, or a similar platform). Students will observe the steps, the tool's capabilities, and the impact of each stage. There may be opportunities for simplified hands-on exercises if the lab environment permits.

4.1 Task 1: Loading the Synthesized Netlist and Initial Setup

1. **Instructor Demonstration:** The instructor will launch the ASIC physical implementation tool.
2. **Loading Input Files:** Observe the instructor loading the input files for the design, which typically include:
    - The **gate-level netlist** (the structural description of the circuit, composed of standard cells and their connections, often in Verilog or EDIF format).
    - The **technology library files** (containing physical and timing characteristics of standard cells, design rules, layer stack-up information from the foundry PDK).
    - **Timing constraints** (SDC file, specifying clock frequencies, input/output delays, setup/hold times).
3. **Design Initialization:** Observe the tool's console output as it initializes the design, reads in all the data, and prepares the environment for physical design.

4.2 Task 2: Floorplanning the Design

1. **Core Area Definition:** Observe the instructor defining the overall physical dimensions of the chip's "core area" where standard cells will be placed. This might involve specifying the aspect ratio or a fixed area.
2. **I/O Pin Placement:** Witness the process of placing the primary input/output (I/O) pins around the periphery of the chip. Discuss how their placement is influenced by packaging requirements or external connectivity.
3. **Power Planning:** Observe the instructor setting up the power delivery network. This typically involves:
    - Creating thick metal rings (VDD and GND) around the core area.
    - Generating a power mesh (interdigitated VDD and GND stripes) over the core area using higher metal layers (e.g., Metal3, Metal4) to distribute power evenly and reduce IR drop.
    - Connecting the standard cell rows to these power rails.
4. **Macro Placement (if applicable):** If the demonstration design includes large IP blocks (e.g., a small SRAM), observe how these are manually placed first, as they often have fixed dimensions and interface points that constrain subsequent placement.
5. **Visualization:** Observe the resulting floorplan in the layout viewer, noting the defined core area, I/O pin locations, and the prominent power grid.

4.3 Task 3: Automatic Standard Cell Placement

1. **Placement Command:** The instructor will initiate the automatic placement engine of the tool.
2. **Observation of Placement:** Observe the tool's progress as it automatically positions thousands or millions of standard cells within the defined core area. The display may update dynamically, showing cells being moved and optimized.
3. **Placement Goals:** Discuss how the tool tries to minimize wirelength and congestion while meeting timing constraints during this process.
4. **Visualization:** Examine the placed design in the layout viewer. You will see individual standard cells (represented by their abstract bounding boxes or detailed layouts) neatly arranged in rows, ready for routing.

4.4 Task 4: Automatic Routing

1. **Routing Command:** The instructor will initiate the automatic routing engine.
2. **Observation of Routing Layers:** Observe how the tool utilizes different metal layers (e.g., Metal1, Metal2, Metal3, etc.) for routing. Notice how wires run predominantly horizontally on some layers and vertically on others, connected by vias.
3. **Routing Progress:** Witness the routing process, which might involve multiple stages (e.g., global routing, detailed routing). The tool will attempt to connect all the pins of the placed standard cells according to the netlist.
4. **Routing Rules Check:** Understand that the router continuously checks for design rule violations (min width, min spacing) during this process.
5. **Visualization:** View the fully routed design in the layout viewer. This will be a dense, intricate pattern of metal wires and vias, representing the complete interconnect fabric of the chip.

4.5 Task 5: Brief Discussion of Post-Layout Extraction and Final Timing

1. **Conceptual Overview:** The instructor will discuss how, after routing is complete, the EDA tool performs a **parasitic extraction** step.
2. **Extracted Information:** Explain that this step calculates the exact parasitic resistances and capacitances from all the wires, vias, and transistor junctions in the *actual physical layout*.
3. **Input for Final Timing:** Discuss that this highly accurate parasitic information is then back-annotated into the netlist and used for the crucial **post-layout static timing analysis (STA)**.
4. **Timing Closure Importance:** Emphasize that this final timing analysis determines if the chip meets all its performance specifications, considering the real-world impact of the physical layout. If timing violations exist, the design cycle must iterate back to placement or routing for optimization ("timing closure").
5. **Tool Output (Demonstration):** The instructor might briefly show a parasitic extraction setup or a final timing report summary to illustrate the outputs of these crucial final backend steps.

5. Post-Lab Questions and Analysis:

Answer the following questions comprehensively in your lab report, based on your observations from the demonstration and your understanding of the ASIC design flow.

1. **Floorplanning Decisions:** What are the three most critical decisions made during the floorplanning stage, and how does each decision significantly impact the subsequent placement and routing stages?
2. **Standard Cell Placement Optimization:** Describe the primary objectives of the automatic placement tool. Explain how minimizing wirelength and avoiding congestion are often conflicting goals, and how the tool tries to balance them.
3. **Routing Challenges:** Based on your observation of the routed design, describe two challenges that an automatic router must overcome to successfully connect millions of gates without errors. How does the use of multiple metal layers help in this regard?
4. **Power Delivery Network:** Explain the purpose of the VDD and GND power rings/meshes observed during floorplanning. Why are these wide metal structures essential for a functional chip, and what problems do they aim to prevent?
5. **From Logic to Silicon:** Reflect on the entire process you observed from a gate-level netlist to a fully routed physical layout. What is the most significant conceptual leap a designer must make when moving from a logical schematic/RTL view to the physical layout view?
6. **Importance of Post-Layout Extraction:** Why is it insufficient to rely solely on pre-layout simulations for final timing sign-off of an ASIC? How does the information from post-layout parasitic extraction contribute to a more accurate prediction of chip performance?

6. Deliverables:

Your lab report must be a professional document, including the following sections and content. Since this is a conceptual/demo lab, focus on detailed descriptions and conceptual understanding.

1. **Title Page:** Your full name, student ID, course name, lab number, date of submission, instructor's name.
2. **Objectives:** Copy the objectives from this lab module.
3. **Pre-Lab Questions:** Your complete and well-reasoned answers to all pre-lab questions.
4. **Procedure Summary:** A detailed narrative summary of the steps demonstrated by the instructor during the lab. Describe what was shown at each stage (Floorplanning, Placement, Routing, Post-Layout Discussion) and your key observations. Include descriptions of any specific tool windows or displays that were highlighted.
5. **Visual Observations:**
    - If permitted, capture or sketch conceptual diagrams/screenshots from the demonstration tool at key stages:
        - **Initial Floorplan:** Showing chip/core boundaries, I/O pin placement, and initial power grid (VDD/GND rings/meshes).
        - **Placed Design:** Showing standard cells arranged in rows within the floorplan (their abstract representations are sufficient).
        - **Fully Routed Design:** Showing the dense network of metal interconnects over the placed cells, illustrating the use of multiple layers.
    - Provide clear descriptive captions for each image/sketch.
6. **Post-Lab Questions:** Your comprehensive, well-articulated answers to all post-lab analysis questions. Support your answers by referencing the observed demonstration and theoretical concepts discussed.
7. **Conclusion:** A concise yet impactful summary of your key learnings from this lab module. Emphasize your understanding of the ASIC physical implementation flow, the purpose of each stage, and why these automated steps are indispensable for modern chip design.
