
Lab Module 5: Layout Versus Schematic (LVS) Verification and Post-Layout Simulation

Discipline: Digital VLSI Design

Module Title: Layout Versus Schematic (LVS) Verification and Post-Layout Simulation

Duration: 4 Hours (or two 120-minute sessions)

---

1. Aim:

The primary aim of this comprehensive lab is to thoroughly understand and proficiently execute the critical post-layout verification steps within the VLSI design flow. Students will gain expertise in the meticulous extraction of parasitic components (resistance, capacitance) directly from a designed physical layout. They will then perform a rigorous Layout Versus Schematic (LVS) verification to unequivocally confirm the faithful representation of the schematic in the physical design. Finally, students will conduct advanced post-layout transient simulations to precisely quantify the impact of these extracted parasitics on crucial circuit performance metrics, specifically propagation delay and power dissipation, contrasting these findings with pre-layout simulation results to highlight the significance of physical effects.

---

2. Theory:

The journey of an integrated circuit design progresses from abstract conceptualization to tangible physical realization. After the functional verification of a circuit's schematic through pre-layout simulations, the next pivotal stage involves translating this logical design into a physical layout – a precise geometric blueprint defining the actual structures and interconnections on a silicon wafer. This transformation, while essential for manufacturing, inherently introduces non-ideal physical phenomena known as **parasitics**. These are unwanted, yet unavoidable, resistances and capacitances that arise from the materials, dimensions, and proximity of the interconnect wires, active device regions (diffusion), and even the inherent capacitance of the transistor terminals themselves. These seemingly minor parasitic effects can, in aggregate, profoundly alter the circuit's intended electrical behavior and performance.

Parasitic Extraction: Quantifying the Unintended:

Parasitic extraction is the analytical process of quantifying these unintended resistive and capacitive elements directly from the detailed geometric information within the physical layout. Specialized algorithms in extraction tools meticulously analyze the dimensions (length, width, thickness), spacing, and material properties of every metal trace, via, diffusion area, and contact.

- **Capacitance:** Arises from electric fields between conductors. Key types include:

    - **Area Capacitance:** Between a conductor and the substrate, or between parallel plates of different metal layers.
    - **Fringe Capacitance:** From the edges of conductors to adjacent conductors or the substrate. More significant for narrower lines.
    - **Coupling Capacitance:** Between adjacent interconnects on the same or different layers, leading to crosstalk effects.
    - **Device Parasitics:** Gate-to-source capacitance (CGS), gate-to-drain capacitance (CGD), drain-to-bulk capacitance (CDB), source-to-bulk capacitance (CSB) are inherent to the MOSFET structure and are also considered during extraction.
- Resistance: Arises from the finite resistivity of the interconnect materials (e.g., copper, aluminum) and the contact/via resistances. Longer and narrower wires have higher resistance.
  The output of the extraction process is an augmented netlist. This netlist describes the original active devices (transistors) alongside an intricate network of explicitly modeled parasitic resistors and capacitors, providing a more comprehensive electrical model of the physical layout.

Layout Versus Schematic (LVS) Verification: The Fidelity Check:

LVS is a cornerstone of IC design sign-off, acting as an indispensable gatekeeper before fabrication. Its core mission is to rigorously verify that the physical layout (what you built) is an exact, one-to-one correspondence with the original schematic (what you intended). The LVS tool performs a sophisticated topological and structural comparison using two primary inputs:

1. **Schematic Netlist:** Derived directly from the logical circuit design captured in the schematic editor.
2. Extracted Netlist: Generated from the physical layout, detailing all identified active devices (transistors with their W/L values) and their interconnections.
   The LVS engine systematically checks for:

- **Device Matching:** Are all transistors (nMOS, pMOS) present in both netlists? Do their types and critical parameters (like W/L ratios, multiplier factors) perfectly match?
- **Net Connectivity Equivalence:** Is every wire and connection in the layout identically mapped to the corresponding net in the schematic? This catches common errors like:
    - **Opens:** A physical break in a connection.
    - **Shorts:** Unintended connection between two nets.
    - **Missing/Extra Devices:** Devices present in one but not the other.
    - Pin Mismatches/Swaps: Input/output pins or internal device terminals connected incorrectly.
      LVS success (a "clean LVS") is non-negotiable for tape-out. Debugging LVS failures cultivates crucial problem-solving skills in physical design.

Post-Layout Simulation (Extracted Simulation): Performance with Reality:

Once the layout has successfully passed LVS, its extracted netlist (which now encompasses the parasitic components) becomes the foundation for "post-layout" or "extracted"

simulations. This simulation is vastly superior to pre-layout simulation because it incorporates the real-world parasitic effects, providing a far more accurate prediction of the circuit's actual performance on silicon.

Impact of Parasitics on Performance Metrics:

- **Propagation Delay:** This is arguably the most critical performance metric affected by parasitics. Every parasitic capacitance on a signal path must be charged or discharged by the transistor's limited current, which takes time. Similarly, parasitic resistances in series with current paths create RC time constants that slow down signal propagation. Consequently, post-layout delays are almost always *greater* than pre-layout (ideal) delays.
- **Power Dissipation:**
    - **Dynamic Power:** This component of power dissipation arises from the charging and discharging of capacitances during switching events. The formula $P_{dynamic}=0.5\times C_{load}\times V_{DD}^2\times f_{switch}$ clearly shows that increased load capacitance (C_{load}, which now includes parasitic capacitances) directly leads to higher dynamic power dissipation.
    - Static Power: While ideally zero in CMOS inverters when quiescent, factors like subthreshold leakage current, gate leakage current, and reverse-bias junction leakage (all minor but present in advanced nodes) can contribute to non-zero static power. Parasitic resistance can also slightly increase static power by creating voltage drops.
      The analysis of post-layout simulation results is vital for verifying that the fabricated circuit will meet its specifications and for identifying potential performance bottlenecks that might necessitate further layout optimization or even schematic modifications.

---

3. Pre-lab Questions:

Students must answer these questions thoroughly before beginning the lab session.

1. Beyond just transistors, what other electrical components are typically present in a netlist generated from a parasitic extraction of a standard cell layout? Explain their physical origin.
2. Differentiate clearly between "connectivity check" and "device parameter check" as performed by an LVS tool. Provide an example of an error that would be caught by each type of check.
3. Why can a circuit pass pre-layout simulations with flying colors but fail to meet performance specifications after fabrication? How does post-layout simulation address this potential discrepancy?
4. Describe the relationship between interconnect length and its associated parasitic resistance and capacitance. How does this relationship influence the decision-making process for routing critical signals in a layout?
5. If you observe that your post-layout propagation delay (t_{pd}) is significantly higher than your pre-layout t_{pd}, what are the primary physical reasons for this increase?

6. Explain the concept of "capacitive loading." How does capacitive loading, including parasitic capacitance, impact the current drive capability required from a driving gate?
7. Outline the complete sequence of steps from schematic entry to final post-layout simulation. Why is the order of these steps important?

---

4. Procedure:

Assumptions: For this lab, you are assumed to have successfully completed the schematic design (Lab 2) and the physical layout design (e.g., Lab 4, ensuring it's DRC-clean) of a CMOS inverter. If you have not, use a provided, DRC-clean CMOS inverter layout.

Part A: Comprehensive Parasitic Extraction

1. **Open Layout View:** Launch your circuit design environment (e.g., Cadence Virtuoso) and open the *layout view* of your CMOS inverter. Ensure it is the top-level view or the cell that you intend to extract.
2. **Launch Extraction Tool:** Navigate to the specific menu or command to initiate the parasitic extraction. This is often found under a "Verification" or "Tools" menu (e.g., `Calibre -> Run PEX`, `Assura -> RCX`, `QRC -> Run`).
3. **Configure Detailed Extraction Settings:**
    - **Extraction Scope:** Confirm that you are extracting from the current cell view.
    - **Output Netlist Format:** Select a common simulation-ready format, typically **SPICE** or **Spectre** (often selected by default). This netlist will include the extracted R and C components.
    - **Extraction Type:** Choose **'RC' extraction**. This ensures that both resistive and capacitive parasitics are calculated. Avoid 'C only' or 'R only' unless specifically instructed for specialized analysis.
    - **Transistor Models:** Verify that the correct technology file and model libraries are linked. The extraction tool needs this to correctly characterize the intrinsic device parasitics (e.g., gate capacitance).
    - **Substrate Connection:** Ensure the substrate/bulk connection is accurately defined for extraction. Typically, the substrate is considered grounded or tied to VDD/GND as per design rules. This affects how substrate capacitances are modeled.
    - **Output File Naming:** Specify a clear name for your extracted netlist (e.g., `inverter_pex.spi` or `inverter_qrc.scs`).
    - **Flat vs. Hierarchical Extraction (for larger designs):** For this single inverter, a flat extraction is sufficient. Understand that for complex designs, hierarchical extraction can be more efficient.
4. **Run Extraction:** Execute the extraction process. This may take a few moments depending on the complexity of the layout and the tool's settings. Monitor the tool's log window for any warnings or errors during extraction.
5. **Examine Extracted Netlist (Crucial Step for Understanding):**
    - Navigate to the directory where the extracted netlist file was saved.
    - Open the generated `.spi` or `.scs` file using a text editor.
    - **Analyze the Contents:**

        - Identify the original transistors (nMOS, pMOS) and their connections.
        - Observe the newly added parasitic elements:
            - Look for capacitor instances (e.g., `C1`, `C_int_VOUT`) connected to various nodes. Note their values (e.g., in fF).
            - Look for resistor instances (e.g., `R1`, `R_VOUT_TRACE`) in series with interconnects or at contacts/vias. Note their values (e.g., in Ohms).
        - Pay particular attention to the output node (`Vout`) and input node (`Vin`) and identify the parasitic capacitances loading these nodes.
        - Note how the tool assigns unique names to each parasitic element.

Part B: Rigorous Layout Versus Schematic (LVS) Verification

1. **Launch LVS Tool:** From your design environment, open the LVS verification tool (e.g., `Calibre -> Run LVS`, `Assura -> LVS`, `PVS -> LVS`).
2. **Specify Input Files for Comparison:**
    - **Layout Input:** Point to the *top-level cell view* of your inverter layout.
    - **Schematic Input:** Point to the *top-level cell view* of your inverter schematic.
    - **LVS Rule Deck/Technology File:** Load the specific LVS rule deck (`.lvs` file) provided by your technology foundry. This file contains the rules for device recognition (how to identify a transistor from layout layers) and connectivity comparison.
3. **Configure LVS Options (Detailed):**
    - **Comparison Mode:** Ensure the tool is set to perform a full device and net comparison.
    - **Power/Ground Recognition:** Explicitly define the names of your power (VDD) and ground (GND) nets. Many tools allow you to specify these as "power" or "ground" nets to avoid unnecessary mismatch reports related to power distribution.
    - **Ignore Parameters/Nets:** For this basic lab, avoid ignoring any parameters or nets. For complex designs, selective ignoring might be used for specific debug scenarios.
    - **Connectivity Extraction:** Verify that the tool is set to extract connectivity from both the layout and schematic.
4. **Run LVS:** Execute the LVS comparison. This process involves the tool extracting a netlist from the layout internally (if not already done) and comparing it against the schematic netlist.
5. **In-depth Analysis of the LVS Report (Critical Debugging Skill):**
    - Locate and open the LVS report file (e.g., `lvs.report`, `_LVS.rpt`).
    - **Successful LVS (Ideal Case):** If LVS passes, the report will display a clear message indicating "Layout and Schematic Match," "Netlists are Equivalent," or "LVS Clean." It will also typically summarize the number of devices and nets found in both.
    - **LVS Mismatches (Common Scenarios & Debugging):**

        - **"Mismatch in Number of Devices":** Means you have more or fewer transistors in one view than the other. Check for accidental deletion, extra instantiation, or incorrect device recognition in layout.
        - **"Device Type Mismatch":** An nMOS in schematic recognized as a pMOS in layout, or vice-versa. Check layer definitions in layout.
        - **"Parameter Mismatch (W/L)":** The W/L of a transistor in layout doesn't match the schematic. Double-check your layout dimensions.
        - **"Net Mismatches":** This is the most common and detailed category.
            - **"Open Circuits":** A net in the schematic is not fully connected in the layout. Visually inspect the routing.
            - **"Short Circuits":** Two distinct nets in the schematic are physically connected in the layout. This is severe. Use the LVS results viewer (if available) to highlight the shorted nets.
            - **"Missing Nets/Extra Nets":** A net exists in one view but not the other, or is improperly recognized.
            - **"Pin Mismatch/Swap":** Input/output pins or internal device terminals are connected to the wrong places.
        - **Debugging Strategy:**
            - Use the LVS report to pinpoint the exact nature and location of the error.
            - Utilize the LVS results viewer (if your tool provides one), which graphically highlights mismatches directly on the layout.
            - Systematically re-check connections in both schematic and layout for the reported errors.
            - Correct the errors in your layout, save, and **re-run LVS** until a clean report is obtained. This iterative process is fundamental to physical design.

Part C: Post-Layout (Extracted) Transient Simulation

1. **Create/Modify Simulation Testbench:**
    - Open the *schematic* of your inverter testbench.
    - **Crucial Step:** Instead of placing the schematic symbol of your inverter, you will now instantiate its **extracted view**. In most tools, you can right-click on the inverter instance and select "View" or "Reference" to choose "extracted" or "pex" view. This tells the simulator to use the netlist with parasitics for this instance.
    - Ensure your input stimulus (Vin pulse source) and output probes (Vout measurement) are configured identically to your pre-layout transient simulation setup from Lab 2.
    - Set the transient simulation stop time to observe multiple switching cycles (e.g., 5-10 times the expected gate delay).
2. **Run Post-Layout Simulation:** Execute the transient simulation using the extracted view.
3. **Plot Waveforms:** Display the Vin and Vout waveforms on the waveform viewer.

Part D: Comprehensive Comparison and Analysis

1. **Retrieve Pre-layout Simulation Results:** Load the Vin and Vout waveforms from your *previous* pre-layout transient simulation (from Lab 2). Ensure the input stimulus is identical.
2. **Overlay and Visual Comparison:**
    - On a single plot, display the following waveforms:
        - Input waveform (Vin).
        - Pre-layout output waveform (Vout_pre).
        - Post-layout output waveform (Vout_post).
    - Carefully observe the differences in the rise and fall times, and particularly the shift in the switching points (delays).
3. **Precise Delay Measurements:**
    - Using the measurement tools (e.g., cursors, built-in delay functions) of your waveform viewer:
        - **Propagation Delay Low-to-High (t_PLH):** Measure the time difference from 50% of Vin (rising edge) to 50% of Vout (falling edge).
        - **Propagation Delay High-to-Low (t_PHL):** Measure the time difference from 50% of Vin (falling edge) to 50% of Vout (rising edge).
        - **Average Propagation Delay (t_PD):** Calculate (t_PLH + t_PHL) / 2.
    - Perform these measurements for *both* the pre-layout and post-layout Vout waveforms.
4. **Detailed Power Dissipation Analysis (Post-Layout):**
    - **Dynamic Power Calculation:**
        - Measure the instantaneous current flowing from the VDD supply (I(VDD)) during the simulation.
        - Calculate instantaneous power: P_inst = VDD * I(VDD).
        - Use the waveform calculator to compute the *average power* over several stable switching cycles (e.g., from the start of the second cycle to the end of the second-to-last cycle to avoid transient effects). Integrate instantaneous power over time and divide by the time duration.
    - If you have calculated pre-layout power (or estimated it using the formula 0.5 * C_load * VDD^2 * f), make a direct comparison.

---

5. Observation/Results:

Document all your findings in a clear, highly organized, and professional manner, using tables, plots, and screenshots as indicated.

1. **Summary of Extracted Parasitics:**
    - Provide a qualitative description of the types of parasitic elements you observed (e.g., "significant interconnect capacitance on Vout," "small series resistance on input trace").
    - Quantify at least one or two dominant parasitic values (e.g., "The total parasitic capacitance on the output node (Vout) was extracted as X fF," "The series resistance of the output metal trace was Y Ohms").

2. **LVS Verification Report:**
    - **Crucial:** Include a screenshot of the *successful* LVS report, clearly showing the "matched" status.
    - If you encountered and resolved LVS mismatches, briefly describe one specific error you found (e.g., "Initially, I had a 'short circuit' error between Vout and VDD due to overlapping metal layers; fixed by...") and how you debugged and fixed it.
3. **Comparative Transient Waveforms:**
    - Present a single, high-resolution plot containing three traces:
        - `Vin` (input stimulus).
        - `Vout_pre` (output from pre-layout simulation).
        - `Vout_post` (output from post-layout simulation).
    - Ensure axes are clearly labeled (Time, Voltage). Use different line styles or colors for clarity.
    - **Annotate the plot:** Use arrows or text to visually highlight the difference in delay between `Vout_pre` and `Vout_post` for both rising and falling transitions.
4. **Detailed Delay Comparison Table:**

Delay Parameter	Pre-Layout Value (ps/ns)	Post-Layout Value (ps/ns)	Absolute Difference (ps/ns)	Percentage Increase
t_PLH				
t_PHL				
t_PD (Average)				

5. **Average Power Dissipation:**
    - State the calculated average power dissipation of your inverter from the post-layout simulation.
    - If you have pre-layout power data, present it alongside the post-layout value and calculate the percentage increase.

---

6. Analysis and Discussion:

Provide a comprehensive, analytical discussion of your experimental results. Go beyond just stating observations; explain the "why" behind them.

1. **The Indispensable Role of Parasitic Extraction:**
    - Elaborate on why including parasitic components is critical for accurate circuit modeling, especially in modern deep sub-micron technologies where interconnect delays can dominate gate delays.
    - Discuss how the complexity of the extracted netlist (the sheer number of R and C elements) increases with layout complexity.
2. **The Uncompromising Nature of LVS Verification:**
    - Explain in detail why LVS is considered a "sign-off" step. What are the severe consequences (e.g., costly re-spins, non-functional chips) of manufacturing a design that fails LVS?
    - Reflect on your personal experience with LVS. How did debugging LVS mismatches enhance your understanding of the relationship between schematic and layout? What debugging strategies proved most effective?
3. **Quantifying the Impact of Parasitics on Delay:**
    - Provide a detailed explanation for the observed differences in `t_PLH` and `t_PHL` between pre-layout and post-layout simulations. Clearly link these increases to the specific parasitic components you identified (e.g., "The output node capacitance directly increased the time required to charge/discharge, leading to a X% increase in t_PLH").
    - Discuss if there was a disproportionate increase in either rise or fall delay. If so, provide a reasoned explanation based on the inverter's pull-up/pull-down strengths and the parasitic loading.
    - Conclude on the significance of these delay changes for overall circuit timing and clock frequency in larger digital systems.
4. **Analyzing Post-Layout Power Dissipation:**
    - Explain how the presence of extracted parasitic capacitances directly contributes to increased dynamic power dissipation. Refer back to the dynamic power formula.
    - Discuss the practical implications of higher power dissipation in terms of heat generation, battery life for portable devices, and the design of power delivery networks.
5. **Iterative Design Flow and Design Margin:**
    - Explain why the VLSI design process is inherently iterative, especially concerning post-layout verification. How does this cycle (layout -> extract -> LVS -> simulate -> optimize) lead to a more robust and manufacturable design?
    - Discuss the concept of "design margin." How do designers account for the inevitable impact of parasitics when setting initial performance targets in the schematic phase?

---

7. Post-lab Questions:

Answer these questions comprehensively after completing the lab and thoroughly analyzing your results.

1. In a deep sub-micron technology (e.g., 28nm), why might interconnect delays (due to parasitic R and C) become more dominant than gate delays in determining the overall propagation delay of a signal path?
2. If an LVS report shows a "short circuit" between two power domains (e.g., VDD and an isolated analog VDD_A), what are the immediate and severe consequences for the chip if this error goes undetected into fabrication?
3. Besides increasing delay and power, what other potential electrical issues can arise from excessive parasitic coupling capacitance between adjacent signal lines in a layout? How can a designer mitigate these?
4. Modern VLSI designs often use various extraction complexities (e.g., 'C only', 'RC', 'RC with coupling'). When would a designer choose a simpler extraction (e.g., 'C only') over a more complex one, and what are the trade-offs?
5. You performed post-layout simulation at nominal process corners. Explain why it is crucial to perform additional post-layout simulations at various process corners (e.g., SS, FF, SF, FS) and temperature variations. How would these affect delay and power?
6. Describe the concept of "design for manufacturability" (DFM) in relation to layout and verification. How do tools like LVS and parasitic extraction contribute to DFM?