Alexandria University
Faculty of Engineering
Division of Communications & Electronics

EE 432 VLSI Modeling and Design
Lab#3: Hierarchial Design and Layout of Complex Logic Circuits

---

Objectives

Upon the completion of this Lab, you should be able to:

1. Design CMOS logic gates respecting the design rules and transistor sizing guidelines.
2. Use hierarchical design to layout complex CMOS circuits from primitive cells.
3. Analyze delay, area, and power parameters in terms of transistor sizing.

Requirements

Lab 3 will expose students to the design of more complex logic gates using hierarchical methods. A 1-bit full adder circuit will be developed using INVERTER, NAND, NOR, and XOR gate. You can use either 2-input or 3-input primitive gates of your design to draw an optimized layout of the full adder circuit. You might need to layout and simulate several implementations of the full-adder circuit to establish an optimum design in terms of area, delay, and power consumption.

You are required to draw an optimized layout of the primitive cells with minimum transistor width on a MOSIS/Orbit n-well 2.0 micron process. You should respect the design rules and transistor sizing regulations to keep the gate delay as minimum as the gate delay of a basic inverter. Use the developed primitives to draw the layout of the full adder circuit.

Use Tspice to verify the functionality of the circuits through transient analysis (check all possible input/output combinations). Also use Tspice to evaluate and compare the delay and power consumption for different implementations of the full adder.

Optional Requirements:

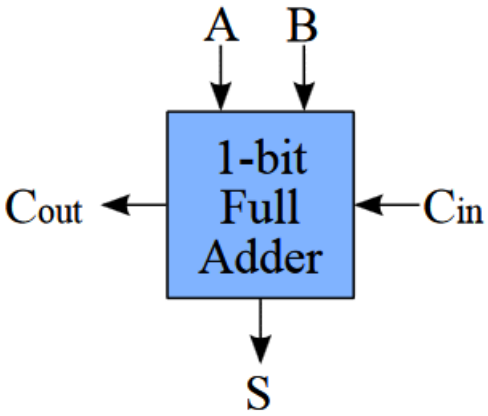
To compare between hierarchical design using cell primitives and flat design using only polygons, you are required to draw the layout of the CMOS implementation of the full adder directly drawn from the logical

equations and truth table. You are only allowed to use the inverter primitive if needed. Use Tspice to verify the functionality and evaluate the delay and power consumption of the developed circuit.

Full Adder Description:

A **full adder** adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and C_{in}; A and B are the operands, and C_{in} is a bit carried in from the next less significant stage. The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. binary numbers. The circuit produces a two-bit output, output carry and sum typically represented by the signals C_{out} and S, where

$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = A.B + A.C_{in} + B.C_{in}$$



A	B	C_{in}	C_{out}	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

Procedures:

1. You need to read the hierarchical design using L-Edit tutorial provided with this Lab.
2. Create the schematic for basic cells including INVERTER, NAND2, NOR2, XOR2, NAND3, NOR3, XOR3 gates using CMOS logic with minimum-sized nMOS (L=2 lambda, W= 5 Lambda) and pMOS transistors. You should consider the ratio of 2.5 between the hole and electron mobility for sizing pMOS transistors. When drawing the schematic, consider how the transistors might be arranged in the layout (step 3).
3. The power and ground rails will be made 10 lambda wide using the Metal-1 layer and the standard cell pitch (height from bottom of the GND rail to top of the VDD rail) will be 70 lambda.
4. Complete the cell layout for the cells and pass DRC and LVS. These cells must be designed as primitive cells. Do not use Metal2. You should optimize the size of the cell by minimizing the cell

width. Do not forget to provide contacts to the pMOS substrate and the n-well.

5. Pass DRC and LVS on the primitive cells and extract the spice file.
6. Use Tspice to verify the functionality of the primitive cells through simulations (check all possible input/output combinations).
7. Instantiate, place, and route primitive cells as needed to form the full adder circuit.
8. Flatten all cells to create one level of polygons, pass DRC and LVS, and extract the spice file of the full adder cell.
9. Use Tspice to verify the functionality of the developed full adder through simulations (check all possible input/output combinations).
10. Use Tspice to evaluate the delay, and power consumption of the developed full adder.
11. Repeat steps 7-9 for other implementations of the full adder and compare between different layouts in terms of area, power consumption, and delay.
12. Use the full adder logical equations and truth table to establish the pull-up and pull-down parts of the CMOS implementation of the full adder outputs.
13. Use the stick diagram and Euler path methods to optimize the layout of the full adder.
14. Complete the cell layout for the full adder and pass DRC and LVS. Do not use Metal2. You should optimize the size of the cell by minimizing the cell width. Do not forget to provide contacts to the pMOS substrate and the n-well.
15. Pass DRC and LVS on the primitive cells and extract the spice file.
16. Use Tspice to verify the functionality of the developed full adder through simulations (check all possible input/output combinations).
17. Use Tspice to evaluate the delay, and power consumption of the developed full adder.
18. Compare between the flat and hierarchical designs of the full adder in terms of area, power consumption, and delay.