VLSI Design : Course Project

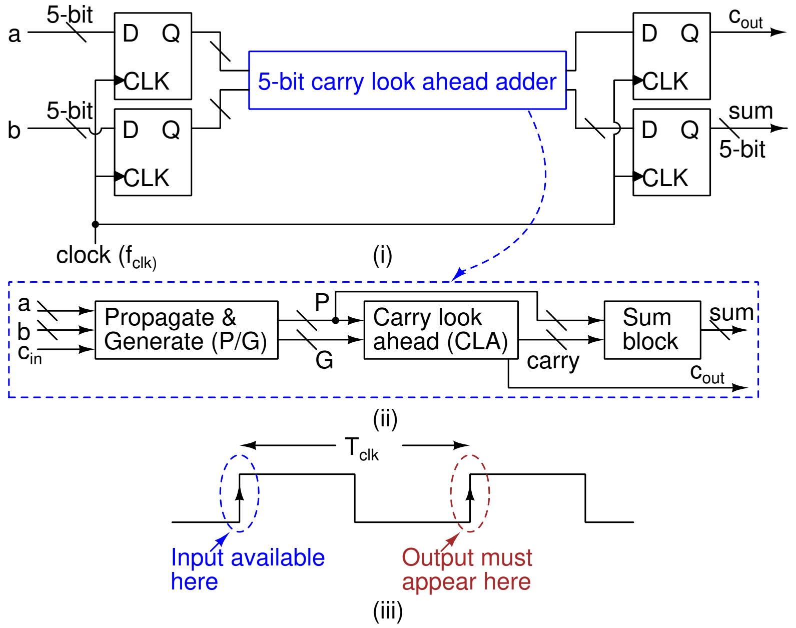
Monsoon 2025, IIIT Hyderabad (Instructor: Prof. Abhishek Srivastava, CVEST)

Release Date: 1-Nov-25, Due date : 3-Dec-25 (18:00 hrs), Demo/Viva date: 4^{th} & 5^{th} Dec 2025

Instructions:

1. Submit your assignment as a single file in pdf format (Name_RollNo.pdf)
2. Use the given 180 nm technology file for the NGSPICE simulations and $SCN6M\_DEEP.09.tech27$ for MAGIC layout
3. Consider lengths of NMOS and PMOS to be equal ($L_n = L_p$), and $V_{DD} = 1.8V$ until stated otherwise
4. Use ‘set curplottitle= Your-name-roll-question-number-part’ for every plot in your report so that it is printed on the top of each plot
5. Answers should be complete and must be presented in a systematic way with explanation, plots, annotations net-lists and HDL description
6. **More than 60% weightage is for the layout & post-layout simulation results and FPGA implementation & demonstration**
7. Utilize moodle platform to discuss and clear your questions. Discussion is highly encouraged.

You are asked to design a 5-bit carry look ahead (CLA) adder as shown in Fig. 1(i). Different modules of the CLA-adder are shown in Fig. 1(ii). Each output sum bit needs to drive an inverter of size W_p/W_n = $20\lambda/10\lambda$, where $\lambda = 0.09\mu$m. As shown in Fig. 1(iii), consider that input bits are available before the rising edge of the clock and the output should be computed and present at the next rising edge of the clock. You can choose any logic style (static, dynamic, mix) to implement the circuit.



(i)

(ii)

(iii)

Figure 1

CLA-Adder: If the numbers to be added are $a_5a_4a_3a_2a_1$ and $b_5b_4b_3b_2b_1$, then the propagate (p_i) and generate (g_i) signals for each bit position can be defined as (for i = 1, 2, 3, 4, 5)

$$p_i = a_i \oplus b_i$$

$$g_i = a_i.b_i$$

and the carry out ($c_{(i+1)}$) of the i^{th} bit position can be written as (assuming c_0 = 0) follows:

$$c_{(i+1)} = (p_i.c_i) + g_i, \quad i = 1, 2, 3, 4, 5$$

Thus, $c_{(i+1)}$ can be expressed entirely in terms of the p_i and g_i functions and sum can be represented as follows:

$$sum_i = p_i \oplus c_{i-1}$$

1. Briefly discuss your proposed structure for the adder.
2. Give design details (topology and sizing) of each block (D-flip-flop, adder modules).
3. Simulate each block and verify its functionality using NGSPICE.
4. For D-flip-flop, find its setup time, hold time and clock to Q delay from NGSPICE simulations.
5. Give stick diagrams of all unique gates in your design.
6. Layout each block using MAGIC layout editor and previously given technology file. Perform post layout extraction and compare the results with schematic simulation.
7. Integrate different block designed in previous steps and write the netlist for the full circuit shown in figure 1(i). Use NGSPICE and verify the functionality of the circuit. Attach the properly annotated waveforms. Report the worst case delay of your adder and maximum clock speed for which your design operates correctly.
8. Give the floor plan of the layout for the complete circuit. Identify the horizontal and vertical pitches in the regular structures.
9. Make layout of the complete circuit and extract the spice netlist. Repeat the simulations discussed above on the extracted netlist. Give a table comparing the schematic and postlayout simulation results.
10. Report the delay of the CLA-adder and maximum clock frequency at which your circuit operates reliably.
11. Using Verilog HDL write the structural description of your circuit and show the correctness of the functionality using simulations. Attach the required wave forms.
12. Implement the given circuit on FPGA and submit the hardware experimental results as oscilloscope waveforms. Same FPGA allocation should be followed for this.
13. Demo and viva will be conducted on 4^{th} & 5^{th} Dec 2025. Please book your tickets for 6^{th} Dec onwards.

Suggested references:

1. Digital Logic and Computer Design by Morris Mano.
2. Computer Architecture and Organization by John P. Hayes.
3. CMOS VLSI Design (fourth edition) by Weste and Harris.
4. Fundamentals of Digital Logic with Verilog Design by Brown and Vranesic.