EE 577A SPRING 2025

LAB 2: TIMING ANALYSIS

Assigned: 02/06/2025 **Due: 02/13/2025**

In this assignment, the goal is to develop a comprehensive understanding of how to define timing parameters effectively. The findings of exhaustive characterization for a logic library are core to the development of new logic cells. These are also crucial for optimizing the design and operation of logic blocks, ensuring they meet the stringent timing requirements of modern high-speed applications.

PART I: (30 Points) TIMING PARAMETERS CHARACTERIZATION

In this part, we will try to complete the simulations of timing parameters of the "*cell under study*" (CUS), where CUS is a simple NAND gate. **You should build the circuit in transistor level before the simulations.** (Reasoning is given in the 4th question)

1. You will build the circuit as shown below, except NAND gate should be in transistor level. The NAND gate is the cell-under study (CUS), the inverters are CUS input drivers, CapOut is CUS output load.

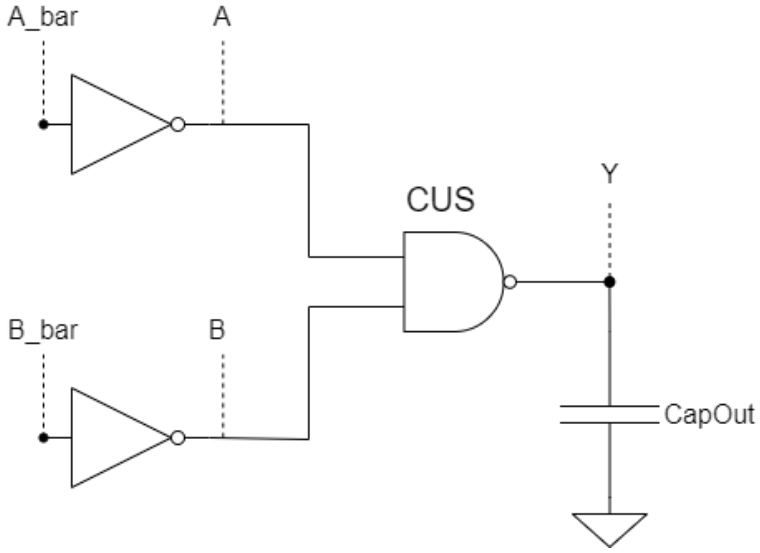

Figure 1

2. Simulation Template Directions
   a. For simulation of each two-pattern test sequence, apply signals using simulation commands at A and B. Let R, F, $S0$ (steady-0), and $S1$ (steady-1) denote two-pattern test sequences 0 → 1, 1 → 0, 0 → 0, and 1 → 1, respectively. Hence, a non-exhaustive set of two-pattern tests sequences for our two-input circuit will require application of the following combinations of values at A and B. We do not plan to investigate "simultaneous switching" scenarios (R-R and F-F). **Summary: Your input pattern should encapsulate four test sequences shown below.**

Table 1

A	R	$S1$	F	$S1$
B	$S1$	R	$S1$	F

b. The above set of two-pattern sequences will be applied for different values of CapOut at the CUS output Y, to enumerate the effects on delay scenarios. You may sweep this value between 0.1f and maximumValue (such as 5f). Choose the number of steps and choose the maximum value yourself.

c. To create different rise/fall times at A and B inputs, you can create a simulation variable called "transition" as we applied in Prelab B before. You can also try another method if you want to use a vector file instead of the "Stimuli". In this method you need to add input capacitances to control rise/fall time at A and B pins. This method is shown below in Figure 2. You need to vary the capacitance value from 0 fF to 1 fF to see the effects of input rise/fall time. **Please either choose the blue method or red method. This change is applied only to observe CellRise and CelFall delays.** You may sweep this value between 1p and maximumValue (such as 100p). Choose the number of steps and choose the maximum value yourself.

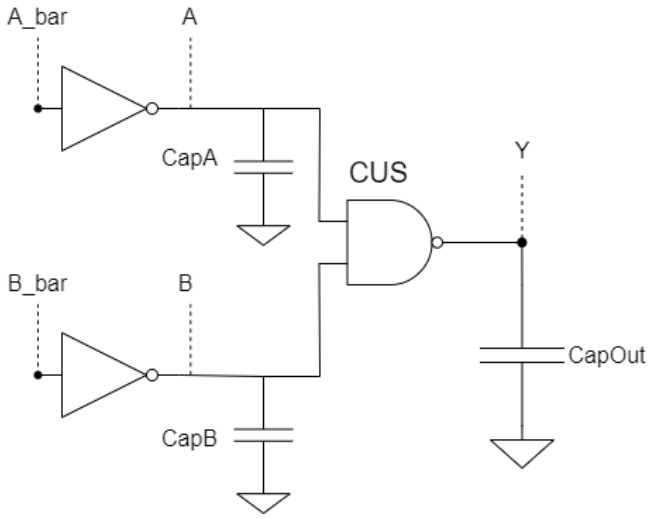

Figure 2

3. All delay measurements will be made at CUS inputs and outputs, i.e., at A, B, and Y. Specifically, input rise/fall times (measured from 10% V_{dd} to 90% V_{dd}) and arrival times (50% of V_{dd}) will be measured at A and B. The output rise/fall times and arrival times will be measured at Y. The skew which is another timing parameter is not considered in this assignment since simultaneous switching scenarios are eliminated. The rise/fall delays for the CUS will be computed using the difference between the arrival times at Y and the arrival time either at A or B.

4. Finally, from all these delay results from simulation, compute the parameters for an extremely simple delay model: **One delay function that captures the worst-case rise delay as a function of load capacitance and another delay function that captures the worst-case fall delay as a function of load capacitance.**

a. Choose a pin that gives worst-case delay scenarios for each of the rising and falling output transitions. Stop simulating other delay expressions and focus on these two values.

b. For the chosen pins, choose the worst input transition time values that gave worst case delay and change your input transition design variable to this value.

c. Sweep output load values for more than 10 steps and extract the results in .csv format.

d. Take this file to your computer and use the provided MATLAB file named "LinearRegression.m" that uses fitting function/linear regression to report the delay model: d(= a + bC)). (a and b are the fitting coefficients).

e. Before using this function, please make sure you modify some of the lines to fit your own excel file format. (Watch Week4 Discussion Session) and report the outputs. The outputs are: (1) a and b: the fitting coefficients, (2) plots

PART II: (70 Points) DECODER: DESIGN EXPLORATION AND OPTIMIZATION

In this problem, you will use a library of primitive complementary logic cells: INV, NAND2, NAND3, …, NAND8, NOR2, NOR3, …, and NOR8.

1. Design a decoder with **eight inputs** $(a_0, a_1, a_2, a_3, \overline{a_0}, \overline{a_1}, \overline{a_2}, \overline{a_3})$ and **16 outputs** $(w_0, w_1, \dots, w_{15})$.

a. Gate-level logic circuit design and exploration: Explore a wide range of design options, **including different implementations** of the four-input function that drives each of the 16 outputs.

b. Shortlisting two designs: Select one design, D1, that requires the smallest number of transistors. Also, select another design, D2, that you think may provide lower delay than D1. **Show gate-level logic circuit diagrams for D1 and D2.** Also, explain your reasons for selecting D2.

c. Getting ready for simulations: Using only the cells available in your 45nm technology's library, create schematics of D1 and D2. For both of your designs:

- At each output, add an external load capacitance that is approximately equal to 64C_INV1x, where C_INV1x is the total gate capacitance of 1x INV in the above library (described in Prelab B).
- Also, add two INVs in series before each input.

d. **Simulations to verify correctness:** Perform simulations for an exhaustive set of input patterns (0000, 0001, 0010, …, 1111) and verify the logical correctness of your designs D1 and D2. *Remember you can use the stimuli in the Maestro tool or use a vector file.*

e. **Simulations to estimate the worst-case delays:** Try to describe the meaningful vectors for the decoder. Identify two-pattern sequence(s) that are likely to excite the highest delay at the output w0 of D1. Repeat for D2. Perform as many simulations as you need to measure the worst-case for D1 as well as D2. According to your assumptions and simulations, define a worst-case a worst-case scenario for each of your designs. Explain your thought process to prove your answer.

Please deliver the files below

- **YourAccountName_LAB2Report.pdf**
- **Cadence Cell file of your simulation.**